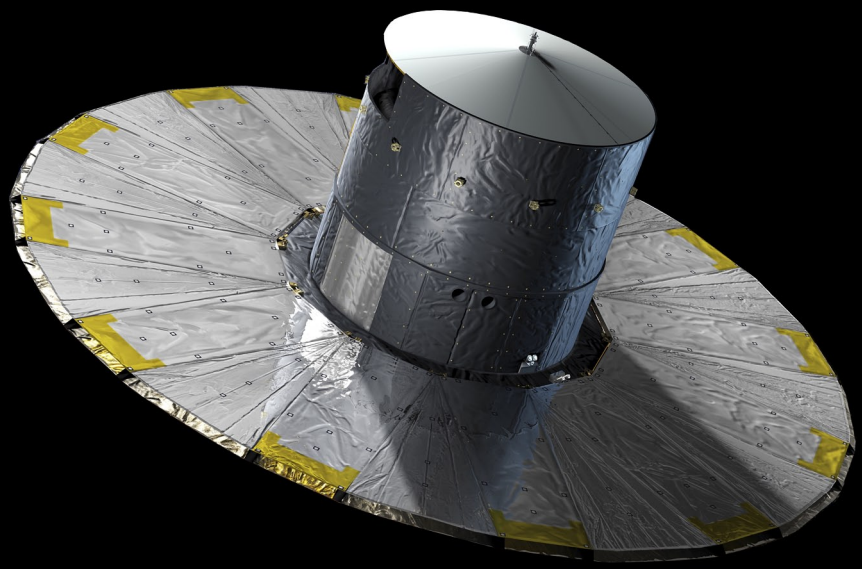
UNIVERSITAT DE BARCELONA

High performance computing of massive Astrometry and Photometry data from Gaia

Javier Bernardo Castañeda Pons

High performance computing of massive Astrometry and Photometry data from Gaia

PhD Thesis



Javier Bernardo Castañeda Pons

September 2015

Programa de doctorat en Física

Línia de Recerca en Astronomia i Astrofísica

High performance computing of massive Astrometry and Photometry data from Gaia

Memòria presentada per

Javier Bernardo Castañeda Pons

per optar al grau de doctor per la Universitat de Barcelona

Directors:

Dr. Claus Vilhelm Fabricius

Dr. Jordi Torra Roca

Tutor:

Dr. Alberto Manrique Oliva

Barcelona, Setembre 2015

The illustrations of the covers are:

- an author's figure of the object detections on board Gaia
- Artist's impression of Gaia (Copyright ESA–D. Ducros, 2013)
  `http://www.esa.int/spaceinimages/Images/2013/08/Artist_s_impression_of_Gaia`

All the figures shown in this thesis have been made by the author unless otherwise specified.

Declaration of Authorship

I, **Javier Bernardo Castañeda Pons**, declare that this thesis titled, **High performance computing of massive Astrometry and Photometry data from Gaia** and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed: _____

Date: _____

UNIVERSITAT DE BARCELONA

Resumen

Facultat de Física
Departament d'Astronomia i Meteorologia

High performance computing of massive Astrometry and Photometry data from Gaia

por Javier Bernardo Castañeda Pons

El trabajo realizado en esta tesis se ha centrado en el diseño y desarrollo de IDU, Intermediate Data Updating. IDU es una de las principales etapas de calibración instrumental y de procesado de los datos astrométricos de la misión espacial Gaia.

En los siguientes apartados se resumen los aspectos principales que han sido cubiertos en esta tesis para la implementación de la primera versión operacional del sistema IDU.

La misión Gaia y su procesado de datos

Gaia es la misión espacial astrométrica más ambiciosa de la Agencia Espacial Europea (ESA). El satélite fue lanzado el 19 de Diciembre de 2013 y su objetivo principal es la producción de un catálogo, con una resolución y precisión sin precedentes, de las posiciones, distancias y velocidades de más de mil millones de estrellas de nuestra galaxia. Este catálogo también incluirá información fotométrica y espectroscópica para la mayor parte de los objetos observados durante toda la misión.

El satélite Gaia está compuesto principalmente de dos telescopios y un centenar de detectores CCD encargados de tomar las medidas

astrométricas, potométricas y espectroscópicas de las estrellas observadas. Estos datos, junto con la información sobre la posición y estado del satélite, son procesados y combinados para calcular los principales parámetros que se incluirán en el catálogo final de Gaia.

La operación y la calibración de los telescopios e instrumentos montados en Gaia son muy complejas. Por este motivo, el estudio del funcionamiento interno del satélite ha sido fundamental para poder afrontar con éxito el diseño de IDU. Este conocimiento se ha ido adquiriendo progresivamente durante esta tesis y ha sido determinante en muchas de las decisiones de diseño adoptadas para IDU.

El procesado de los datos de Gaia es un gran reto científico y tecnológico. En particular, el gran volumen de datos a procesar y el elevado número de procesos involucrados ha implicado la adopción de un sistema de distribución y procesado de datos muy complejo. Este procesado comienza con la recepción diaria de los datos del satélite. Estos datos son procesados produciendo unos resultados intermedios preliminares que se utilizan para monitorear diariamente el funcionamiento del satélite. Tras este procesado preliminar, los resutados intermedios entran de forma continuada en un sistema iterativo de reducción de datos donde diferentes procesos se encargan de resolver los diferentes aspectos científicos de la reducción de datos de Gaia. Estos procesos, además, se ejecutan en diferentes centros de procesado debido a los elevados requisitos computacionales que necesitan.

De entre estos procesos iterativos, esta tesis se centra en IDU. En este sistema, todos los datos en bruto recibidos del satélite se reprocesan y se calibran de nuevo usando los resultados más recientes obtenidos de los otros procesos iterativos. En este sentido, IDU es esencial para conseguir la convergencia del proceso iterativo global ya que es el encargado de regenerar y mejorar los datos intermedios que constituyen, a su vez, el punto de partida para el resto de sistemas de procesado.

Diseño e Implementación de IDU

El diseño e implementación de IDU ha presentado una gran variedad de retos; incluyendo los problemas puramente científicos pero también las dificultades técnicas que aparecen en el procesado del gran volumen de datos de Gaia. Adicionalmente, la gestión de todas las tareas de desarrollo, test y coordinación de los equipos que contribuyen a este sistema también se han cubierto en esta tesis.

IDU se compone de siete tareas de procesado diferentes. Cada una de estas tareas presenta características peculiares tanto en su implementación científica como técnica. Estas características han sido descritas y analizadas en detalle en esta tesis incluyendo los principales motivos por los que cada tarea es imprescindible y como se integran sus resultados en las demás tareas o sistemas.

Las tareas de procesado de IDU se ejecutan en el supercomputador Marenostrum, gestionado por el Barcelona Supercomputing Center (BSC). La integración de IDU en Marenostrum ha sido una de las tareas más complejas de esta tesis. Esta integración ha consistido básicamente en el desarrollo de una infraestructura para la implementación y distribución de las tareas de IDU en los nodos de computación que ofrece Marenostrum. Esta infraestructura se ha diseñado para ser lo más flexible y versátil posible de manera que pueda ser fácilmente adaptada a cualquier entorno de procesado y ofrezca las máximas facilidades para poder introducir nuevas funcionalidades según sea necesario. Como parte de esta infraestructura, también se ha desarrollado una capa de acceso a datos con el objetivo de ofrecer un acceso lo más eficiente posible a los datos de Gaia.

Por otra parte, una parte importante del diseño de IDU se ha centrado en el análisis de los datos de entrada y en la caracterización del perfil de procesado de cada tarea. Esta información es fundamental para la ecualización del sistema de procesado. El conocimiento detallado de los datos de entrada (volumen, distribución temporal/espacial, etc.) y un buen modelo de la respuesta de las tareas a esos

datos es fundamental a la hora de diseñar un sistema de distribución eficiente en un superordenador como Marenostrum. Para este propósito se han llevado a cabo numerosos ensayos que también han servido para la mejora y la identificación de problemas en las tareas de IDU.

Por supuesto, el rendimiento tanto científico como computacional de IDU ha sido también analizado en detalle. El conocimiento detallado de este rendimiento es esencial para obtener resultados de calidad y lograr un uso eficiente de los recursos. Por este motivo hemos desarrollado numerosas herramientas para la monitorización del correcto rendimiento del sistema y para la validación y visualización de los resultados científicos de las diferentes tareas. Estas mismas herramientas son las que se han utlizado para generar la totalidad de las figuras de resultados incluidas en esta tesis.

Primeros resultados operacionales de IDU

Esta tesis incluye los resultados de la primera ejecución operacional de dos de las tareas principales de IDU en el Marenostrum. La primera ejecución de estas tareas no estaba prevista hasta finales de 2015 pero su ejecución fue adelantada seis meses para poder corregir algunos de los problemas del procesado diario de datos. Esta ejecución ha constituido el primer procesado oficial a nivel iterativo sobre datos reales de Gaia y constituye uno de los mayores logros conseguidos gracias a todo el trabajo realizado en esta tesis.

Los resultados científicos y el rendimiento computacional han sido analizados en detalle enfatizando aquellos aspectos directamente relacionados con las funcionalidades más relevantes desarrolladas específicamente para IDU en esta tesis. Además este ejercicio ha servido también para presentar los diagnósticos que producen las herramientas desarrolladas para el análisis del rendimiento de IDU. Estas herramientas son fundamentales para monitorizar y detectar cualquier

problema durante el procesado y además aportan información esencial para la mejora del sistema.

Por último, con esta ejecución hemos podido demostrar como el sistema que hemos diseñado y desarrollado para IDU es completamente capaz de gestionar y procesar el gran volumen de datos de Gaia haciendo un uso eficiente de los recursos del supercomputador Marenostrum.

Conclusiones

IDU (Intermediate Data Updating) constituye una parte fundamental en el procesamiento de datos de Gaia. Este sistema se encarga de varios procesos, de calibración del instrumento y de procesado de datos, que son esenciales para poder lograr los objetivos finales de precisión en astrometría de la misión. Además, el procesado en IDU es muy exigente tanto en el volumen de datos que gestiona como en los recursos de computación que necesita.

Esta tesis se ha dedicado al diseño e implementación de este sistema de procesado. El trabajo desarrollado ha contribuido amplia y positivamente a la definición, evolución y operación del sistema de procesado iterativo de Gaia. Concretamente, dos de las tareas y parte de la infraestructura que hemos desarrollado ya han sido ejecutadas con éxito en el procesado de datos reales de Gaia aportando antes de lo previsto beneficios en los resultados obtenidos hasta el momento.

Finalmente, los ensayos llevados a cabo indican además que, en gran medida, IDU está preparado para poder gestionar el gran volumen de datos de Gaia y que podrá afrontar sin problemas el exigente reto de procesar los datos reales de Gaia durante los próximos años de misión.

UNIVERSITAT DE BARCELONA

Abstract

Facultat de Física
Departament d'Astronomia i Meteorologia

High performance computing of massive Astrometry and Photometry data from Gaia

by Javier Bernardo Castañeda Pons

Gaia is an extremely ambitious astrometric space mission adopted within the scientific programme of the European Space Agency (ESA) in October 2000. It aims to measure with very high accuracy the positions, motions and parallaxes of a large number of stars and galactic objects, including also for almost all the objects information about their brightness, colour, radial velocity, orbits and astrophysical parameters. Gaia requires a demanding data processing system on both data volume and processing power. The treatment of the Gaia data has been designed as an iterative process between several systems each one solving different aspects of the data reduction system.

In this thesis we have addressed the design and implementation of the Intermediate Data Updating (IDU) system. The Intermediate Data Updating (IDU) is the instrument calibration and astrometric data processing system more demanding in data volume and processing power of the data processing system of the Gaia satellite data. Without this system, Gaia would not be able to provide the envisaged accuracies and its presence is fundamental to get the optimum convergence of the iterative process on which all the data processing of the spacecraft is based.

The design and implementation of an efficient IDU system is not a simple task and a good knowledge of the Gaia mission is fundamental. This design and implementation work is not only referring to the actual design and

coding of the system but also to the management and scheduling of all the related development tasks, system tests and in addition the coordination of the teams contributing to this system. The developed system is very flexible and modular so it can be easily adapted and extended to cope with the changes on the operational processing requirements.

In addition, the design and implementation of IDU presents a variety of interesting challenges; covering not only the purely scientific problems that appear in any data reduction but also the technical issues for the processing of the huge amount of data that Gaia is providing. The design has also been driven by the characteristics and restrictions of the execution environment and resources – Marenostrum supercomputer hosted by the Barcelona Supercomputing Center (BSC) (Spain). Furthermore, we have developed several tools to make the handling of the data easier; including tailored data access routines, efficient data formats and an autonomous application in charge of handling and checking the correctness of all the input data entering and produced by IDU.

Finally, we have been able to test and demonstrate how all the work done in the design and implementation of IDU is more than capable of dealing with the real Gaia data processing. We have basically executed two of the IDU tasks over the first ten months of routine operational Gaia data. This execution has been the very first cyclic data processing level run over real Gaia data so far. Executing IDU at Marenostrum over that amount of data for the first time has been a challenging task and from the results obtained we are confident that the system, we have designed and that constitutes the bulk of this thesis, is ready to cope with the Gaia data according to the requirements sets. Furthermore, the presented design provides a solid IDU system foundation for the demanding task of processing the Gaia data during the forthcoming years.

Acknowledgements

The present work is the result of a long and fruitful collaboration between many teams and individuals within the Departament d'Astronomia i Meteorologia, Institut de Ciències del Cosmos (ICC), Institut d'Estudis Espacials de Catalunya (IEEC), Universitat de Barcelona (UB), Gaia Data Analysis and Processing Consortium (DPAC) and the Barcelona Supercomputing Center (BSC) teams.

Jordi P. you are the first I would like to acknowledge; without you I would not have been working in Gaia and without your help I would not have been able to succeed in the completion of this thesis. Your personality and work are endless sources of inspiration ;-).

Secondly, I would like to thank my supervisors, Claus F. and Jordi T., it is always a pleasure to work with you and your guidance and expertise have contributed substantially to the work presented here – this thesis is of course also yours. Claus, I learned a lot working along your side and I could not have imagined having a better advisor and mentor for my PhD study. I am also very grateful, and I have also learned a lot, from Cesca F., Carme J., Xavi L. and Eduard M.. All of you have always been willing to help me and your expertise have been always very helpful and fruitful.

I want also to thank our colleagues from Edinburgh who provided insight and expertise that greatly assisted this research. The fruitful stay in Edinburgh was the perfect break–out from the normal work in Barcelona. In particular, Nigel H. and Michael D., you make me feel right at home and it is always a great pleasure meeting you.

Thanks to everyone working within DPAC. It is an honour and a privilege to be involved with this project, and I wish every success in the future.

Thanks also to everyone working at the Barcelona Supercomputing Center for the facilities and support provided during all these years. Your expertise have also contributed substantially to the work presented here and in general to the work of our group at the UB within Gaia.

Thanks my office mates Nora G., Marcial C. and Juanjo G. for the numerous digressive philosophical chats! To all the current members of the Gaia team in Barcelona; Erika A., Raul B., Dani M., M. Romero, Josep Manel C., Lola B., F. Julbe, Holger V. – always hard working – thanks a lot for your support and for all the fun we have had in the last years. Also thanks to all the people no longer working with us; Eva G., Toni S., Nadia B., Pau V., Marwan G., Aidan F., Maria C., Dafnis B., Teresa and Oscar. It was a pleasure getting to know you all.

A big thanks to all the people at the Department of the University of Barcelona for making it a unique environment. It was an honour to work with researchers and students always at the forefront of their discipline and so involved in outreach activities, essential for connecting and infecting others with your enthusiasm. I started my PhD journey together with most of you, and it has been a pleasure to have shared some experiences, thanks for helping me along at times and congratulations on your achievements. I take also this opportunity to express gratitude to all of the Department faculty members for their help and support, specially to J.R., Jordi V. and Gabi P..

Igualmente, también agradecer a todos mis amigos en Palma y Barcelona que me han dado ánimos para poder acabar esta tesis y que por fín tendran una prueba palpable de que realmente estaba realizando un doctorado.

Por último, los agradecimientos finales de esta tesis van para mi familia; que siempre han mostrado una gran fe en mi éxito y me han apoyado incluso desde la distancia en la escritura de esta tesis y en mi vida en general. Ya hace prácticamente 18 años que me marché de la isla, casi ya media vida, pero gracias a dios todavia me reconocen y aceptan cuando vuelvo... Mallorca siempre ha sido y será mi hogar. Y por último, a Irene, que ha experimentado probablemente más que nadie mis momentos de estrés y desconexiones durante los momentos de más trabajo. Desde que nos conocimos en Edimburgo, hace ya 5 años, siempre me has apoyado y probablemente seas la persona que más contenta estarás cuando por fin se

acabe esta etapa de mi vida y podamos entonces plantearnos y empezar una nueva aventura, esta vez juntos.

Thanks all! Muchas gracias! Moltes gràcies!

Acknowledgement of funding support

This work has been partially funded by the European Union FEDER funds and the *Ministerio de Educación y Ciencia* through grants: ESP2005--24356-E, ESP2006-26356-E, ESP2006-13855-C02-01 and CSD2007-00050. It has also been supported by the *Ministerio de Ciencia e Innovación* through grants AYA2009-14648-C02-01 and CONSOLIDER CSD2007-00050 and lately by the *Ministerio de Economía y Competitividad*; grants AYA2012--39551-C02-01, ESP2013-48318-C2-1-R and AYA2014-53365-REDT.

Finally, the Gestió d'Ajuts Universitaris i de Recerca (AGAUR) has also contributed through grants 2009SGR217 and 2014SGR86.

Contents

List of Figures

Chapter 1

Chapter 2

Chapter 3

Chapter 4

Chapter 5

Chapter 6

List of Tables

Chapter 1

Chapter 2

Chapter 3

Chapter 4

Chapter 5

Chapter 6

1 Gaia Mission Overview

1.1 Introduction

Gaia is an extremely ambitious astrometric space mission [Perryman et al., 2001] adopted within the scientific programme of the European Space Agency (ESA) in October 2000.

Gaia (Figure 1.1) aims to measure with very high accuracy the positions, motions and parallaxes of a large number of stars and galactic objects [ESA, 2014e]. Consequently, a detailed three-dimensional map of more

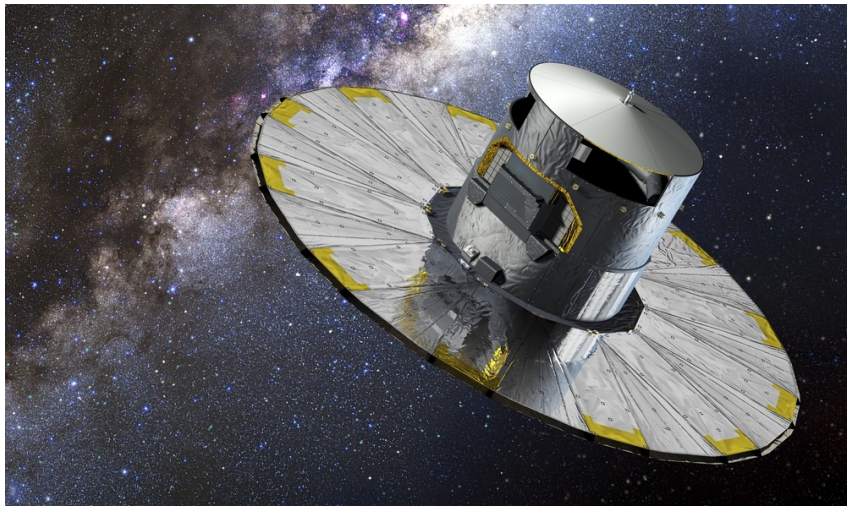


Figure 1.1: Gaia spacecraft in space – Concept Art (Credit: ESA/-Gaia)

FIGURE 1.2: Milky Way concept art on top and the expected Gaia coverage on bottom (Credit: Gaia/DPAC/CU2)

than 1 billion stars of our Galaxy up to the 20^{th} magnitude will be obtained (approximately 1% of the stars populating the Milky Way). This map will also include for almost all the objects information about their brightness, colour, radial velocity, orbits and astrophysical parameters. Gaia will also reveal and classify thousands of extra-solar planetary systems, minor bodies within our solar system and millions of extragalactic objects, including some 500.000 quasars. Figure 1.2 shows the expected Milky Way coverage Gaia will achieve at the end of the mission lifetime.

Gaia scans the sky using the proven principles of its precursor ESA mission Hipparcos [Perryman, 2010]. Hipparcos was the first satellite devoted to precision astronomy, launched in 1989 and operated until 1993. The resulting Hipparcos Catalogue was published in 1997 and contains the positions, distances and movements, 200 times more accurate than any previous measurement, for almost 120.000 stars.

Gaia will significantly improve Hipparcos not only in the number of objects mapped but also in the precision of the angular measurements obtained.

Gaia will be able to measure the position and motions more accurately than Hipparcos, reaching about 25 μas at 15^{th} magnitude [ESA, 2014f].

The data produced during the mission lifetime will require over a Petabyte of disk storage. The raw data will be originally transmitted and compressed from the spacecraft to Earth (around 100 Terabytes) and then it will be processed to turn it into a calibrated set of measurements for the astronomical community. Gaia is one clear example on how mission data sets are increasing in size and complexity which, additionally to the design and building of the spacecraft itself, requires also the development of new computer software for its efficient processing. With Gaia, engineers and astronomers have a very big challenge for dealing with a massive flood of data. To deal with this very large amount of data it is currently estimated that Gaia will need a processing power of 10^{21} flops [Mignard et al., 2007].

Gaia will be acquiring astrometric and spectroscopic measurements over a nominal period of five years and its science results will be released in the form of progressive catalogues at several points in the mission starting in 2016. First releases will mainly include the main astrometric parameters and they will be progressively extended until the completion of the science mission, when the final catalogue will be released.

Next sections summarize some additional topics about the Gaia mission which were considered relevant for the scope of this work, more extended information can be found in ESA's Gaia Brochure [Clark and EJR-Quartz, 2012] and ESA's Gaia Portal [ESA, 2014a].

1.2 Spacecraft

Gaia, is a very complex space observatory developed and built by Airbus Defence and Space (ADS).

Figure 1.3 shows the breakdown of the spacecraft in its more essential parts. All these parts can be classified in three main modules: a Payload Module, a Mechanical Service Module and an Electrical Service Module.

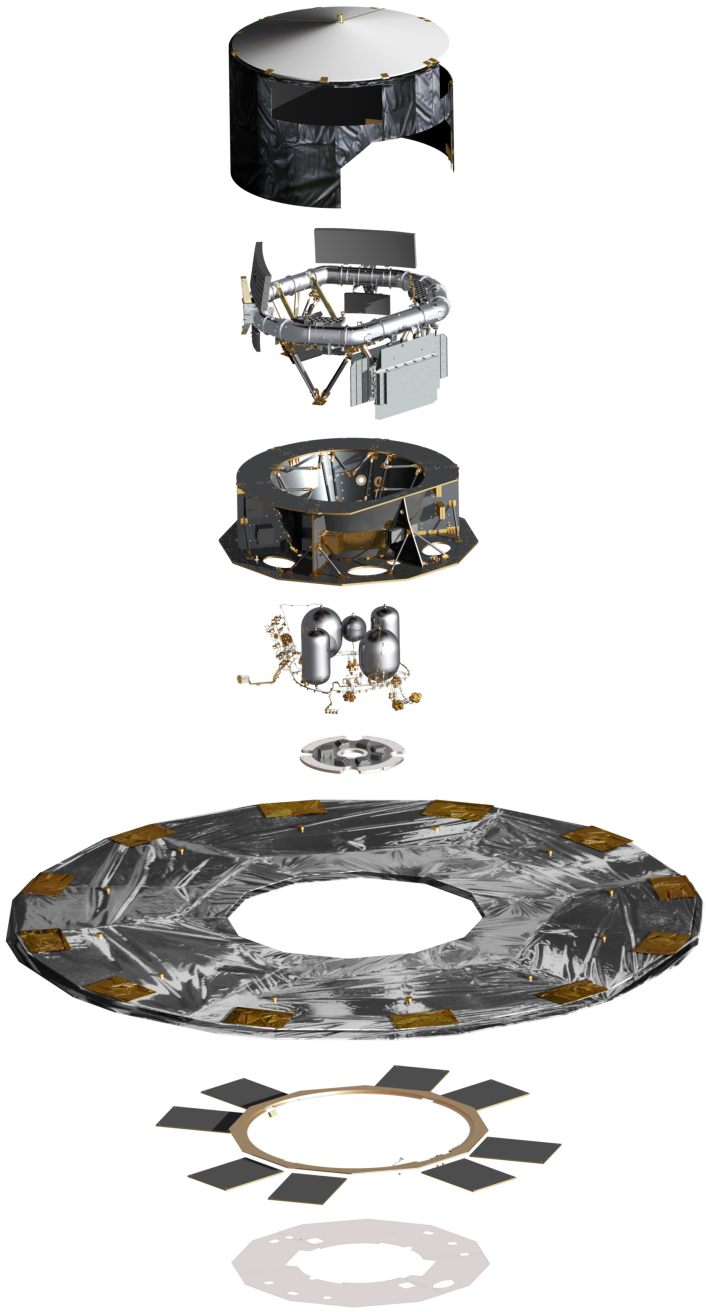


FIGURE 1.3: Gaia spacecraft decomposition illustration - from top to bottom: Thermal Tent, Payload Module, Service Module, Propellant and Pressurant Tanks, Equipped Phase Array Antenna Panel, Deployable Sun shield, Deployable Solar Panels and Fixed Solar Panels (Credit: ESA/Gaia)

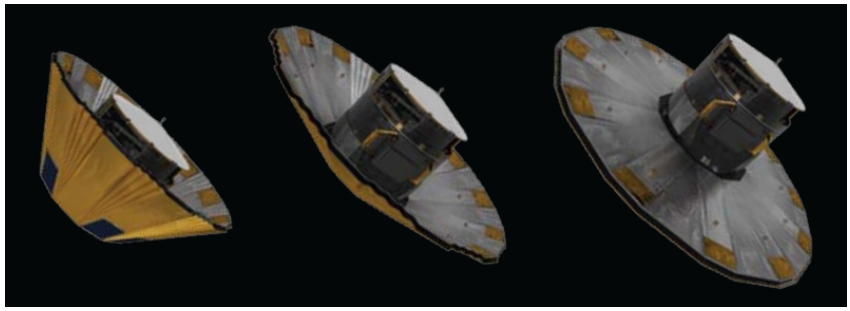


FIGURE 1.4: Gaia Sunshield deployment illustration (Credit: ESA/-Gaia)

The Payload Module is housed inside a protective Thermal Tent and contains the two optical telescopes and the three science instruments further described in Section 1.3. They are all mounted on a torus made of a ceramic material presenting a very strong mechanical and thermal stability performance which is essential for the measurement accuracy required for Gaia.

The Electrical Service Module, located underneath the Payload Module, contains the electronic units to run the instruments, the Payload Data Handling Unit (PDHU), the power module, communications units and other electronic subsystems. The Attitude and Orbit Control sub-System (AOCS) and Star Trackers are also part of the Electrical Service Module of Gaia and are in charge of determining the spacecraft orientation and position in space. The Mechanical Service Module includes all mechanical, structural and thermal elements that support the instruments and electronics. It also includes the Propulsion Systems and the deployable Sun shield. Gaia Sun shield is an essential component of the spacecraft. It keeps Gaia in shadow, maintaining the Payload Module at an almost constant temperature of around -110°C. This Sun shield measures about 10 meters across and was build with folding panels to be deployed after launch as shown in Figure 1.4.

1.3 Instrument

Gaia has two optical telescopes which are combined into a single focal plane. These two telescopes involve a total of ten mirrors of various sizes and surface shapes to collect, focus and direct light to the Focal Plane of Gaia.

The Focal Plane is composed of 106 Charge-Coupled Devices (CCDs) to record the light coming from both telescopes. Figure 1.5 shows the actual focal plane built and installed in Gaia by ADS. Added together, the Gaia CCDs make the largest focal plane ever flown in space, a total of almost one billion pixels (one Gigapixel).

The Focal Plane integrates the detectors of three instruments:

- The astrometric instrument is devoted to measuring the stellar positions, object flux and provide data to track their motion and parallax.
- The photometric instrument provides colour information through two low-resolution spectra, one in the blue and one in the red range of the optical spectrum which will be used for the determination of the

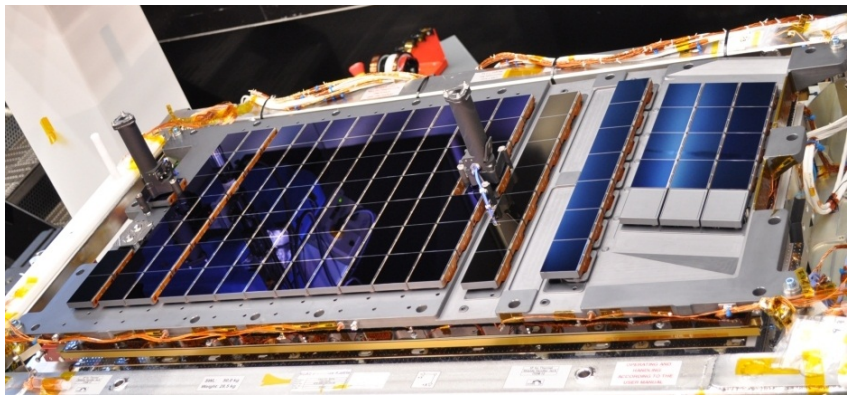


FIGURE 1.5: Gaia Focal Plane picture showing the 106 Gaia CCDs integrated onto the support structure. The different instruments of the focal plane can be recognized: (from left to right) the two BAM and the first WFS, then the two SM, the nine AF (plus the second WFS), the BP and RP and finally the RVS (last three columns) (Credit: ESA)

stellar properties such as temperature, mass and chemical composition.

- The Radial Velocity Spectrometer determines the velocity of the brighter objects along the line of sight of Gaia by measuring the Doppler shift of absorption lines.

In this Thesis, we will mainly focus on the calibration of the astrometric instrument and the processing of its related data which is the core of the data reduction system of Gaia.

1.4 Launch, Orbit and Comissioning Phase

Gaia launch was originally planned in December 2011 but Payload Module production complications, related to the focal plane and the telescope mirrors, caused a notable delay of about 21 months. On October 2013, a technical issue was identified in the transponders of another satellite already in orbit, which are also installed on Gaia. As precautionary measure, the potentially faulty components were verified and replaced introducing an additional delay of one month in the final verification test campaign.

At last, after two additional months to fit the launch into the overall schedule at Kourou (French Guiana), Gaia was successfully launched on 19 December 2013 at 09:12 UTC from Europe's Space port in French Guiana by Arianespace [ESA, 2014a]. The spacecraft was carried into space by a Soyuz-STB launch vehicle with a Fregat-MT upper stage. This three-stage version Soyuz has been launched more than 850 times and is one of the most-used and reliable launch vehicles. Figure 1.6 shows a detailed overview of the launch sequence of Gaia until the burn to inject Gaia into its L_2 transfer trajectory.

The spacecraft reached its working orbit three weeks after launch, on 8 January 2014. Gaia follows a Lissajous Orbit around the Sun-Earth L_2 point, one of the gravitationally stable Lagrangian Points. L_2 offers a

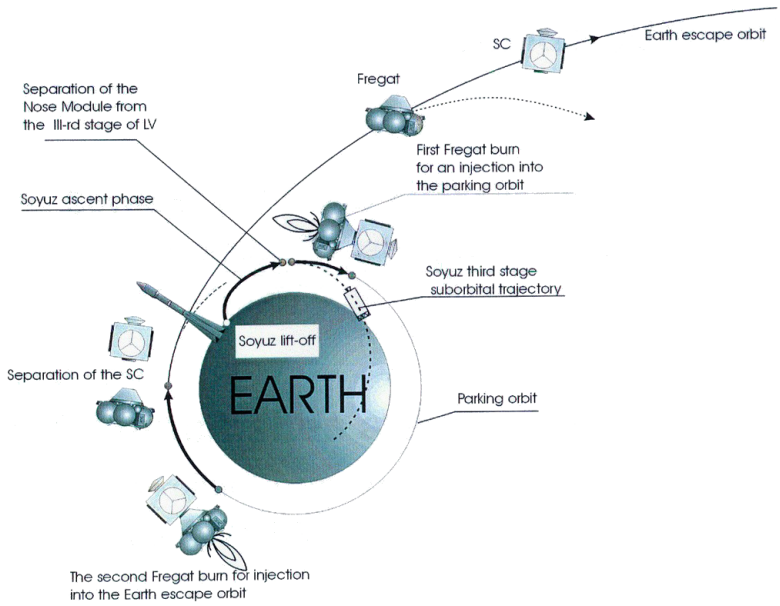


FIGURE 1.6: Overview of Gaia's launch sequence (Credit: ESA/Gaia)

stable thermal environment with moderate radiation, which benefits the longevity of the instrument detectors. Several other satellites have already taken advantage of this location, including Herschel and Planck, also from ESA.

Full sky coverage is accomplished thanks to the spin of the satellite around its own axis, which itself precesses at a fixed angle of 45 degrees with respect to the Sun-Satellite line. The spacecraft rotates at a constant angular rate of 60 arcsecond per second around an axis perpendicular to the two fields of view, thus describing a full great circle in 6 hours.

Thanks to the adopted scanning principle, Gaia will observe each star 75 times on average during the five years of duration of the mission. Figure 1.7 shows a Hammer-Aitoff Projection of the sky in terms of observation per square degree already observed by Gaia during the first year of the mission. In this figure it can be seen how Gaia has already scanned the full sky and it can also be appreciated how the telescopes have scanned some regions more than others, result of the observing scanning law adopted described later in Chapter 2.

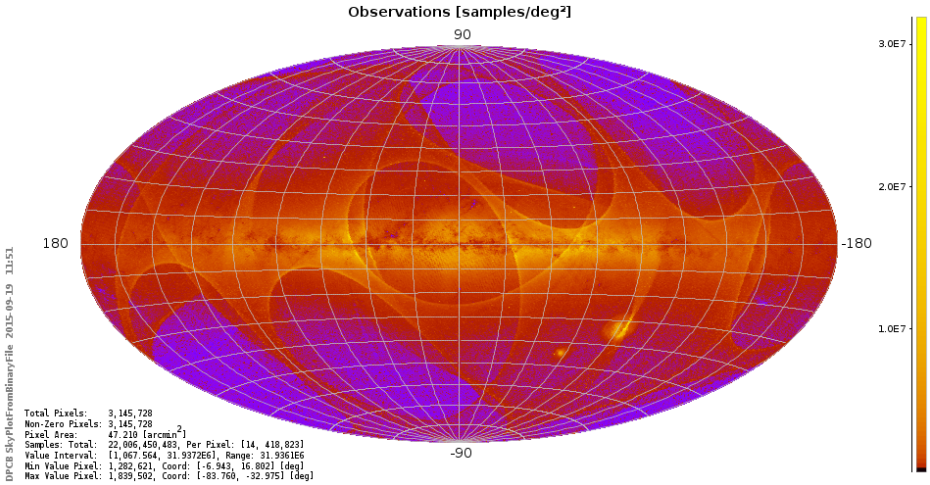


FIGURE 1.7: Gaia observation density obtained during the first ten month of routine operations, in Galactic coordinates

The *commissioning phase* to test and calibrate the spacecraft started while Gaia was on route to the L_2 point and continued until the end of July 2014. During this phase, the Gaia team on ground sent commands to the spacecraft to adjust the spin rate, mirrors focus and many other configuration parameters to bring the spacecraft up to full performance.

This phase, originally planned for four months, took three additional months due to unforeseen issues [ESA, 2014c]. The most relevant issues were those related to:

- Ice deposits on the mirrors, water was likely trapped in the spacecraft before launch and emerged once it was in a vacuum causing a steady drop in the transmission of the telescopes.
- High level *stray light* entering the detector from the astronomical sky and the Sun, obtaining very high background levels, variable in time and across the focal plane.

The ice deposits have been largely removed by repeated decontamination activities on-board based on the selective heating of the affected optics and most likely further decontaminations will be exercised during the mission

as long as the degradation shows up. Regarding the stray light issue, the main source was initially thought to be related with the ice deposits, but later its source has been identified as the fibres of the Sun shield edges [ESA, 2014g].

Fortunately, it has been concluded that the degradation in science performance due to both issues for the astrometric instrument will be relatively modest and mostly restricted to objects with low flux levels [ESA, 2015d].

The first science data products were released by the mission in June. The measurements, acquired in the form of images centered in the detected observations, with the astrometric CCDs (Figure 1.8) and the star spectra acquired with the photometers (Figure 1.9) and the Radial Velocity Spectrometer (Figure 1.10) showed the expected quality and matched the ground-based references.

Since 18 of July 2014 [ESA, 2014b], the *commissioning phase* was considered accomplished and the operations team on-ground focused their efforts on the complex task of processing the large amount of science data received on a daily basis from the spacecraft.

At this point Gaia has been working for more than a year already observing more than 25 billion transits (around 18 Terabytes of science data), seeing objects fainter than required and even discovering its first Supernova [ESA, 2015a].

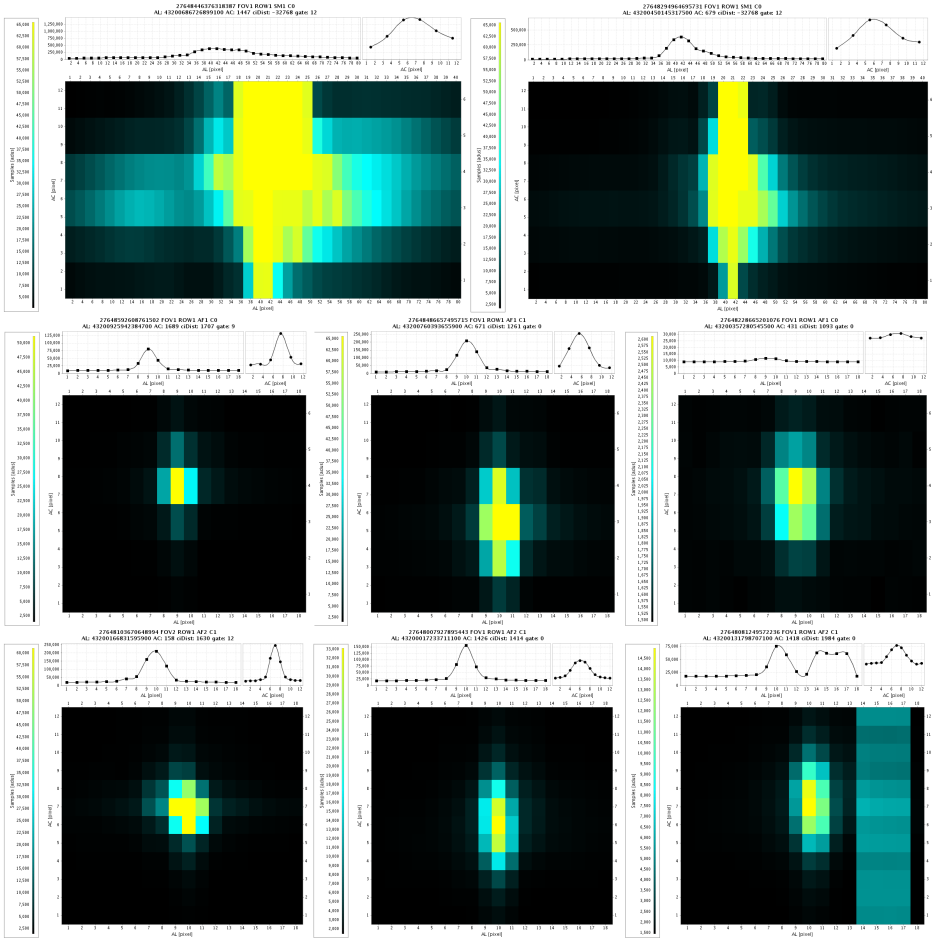


FIGURE 1.8: Astrometric observation examples for SM, AF1 and AF2 CCDs (top, middle and bottom pane line respectively). Total window flux decreases from left to right pane in each CCD cases - note that first examples for AF1 and AF2 are gated observations. Additionally, a CI of 4 TDIs can be clearly seen in the bottom right pane

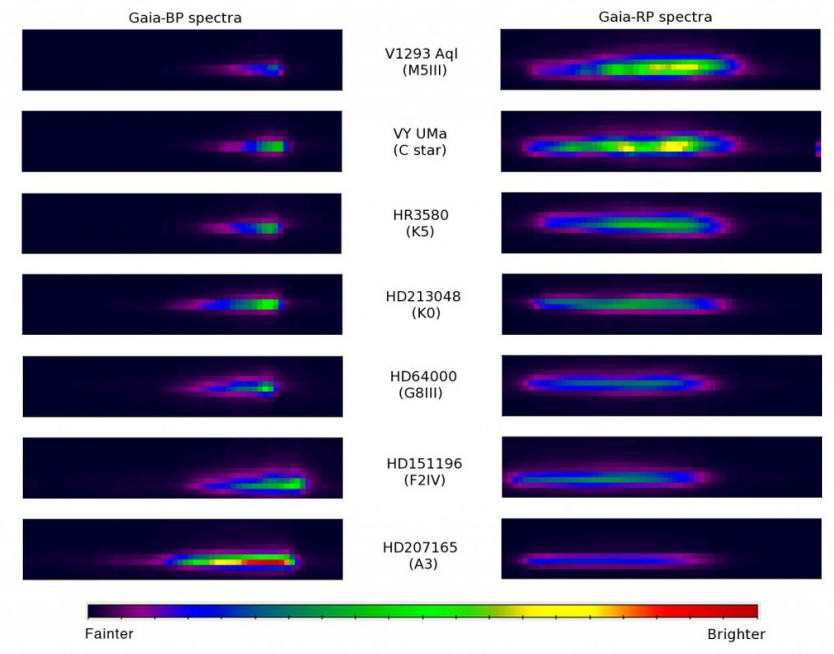


FIGURE 1.9: Photometer observations examples for different source types; BP examples on left pane and RP examples on right pane (Credit: ESA/Gaia/DPAC/Airbus DS)

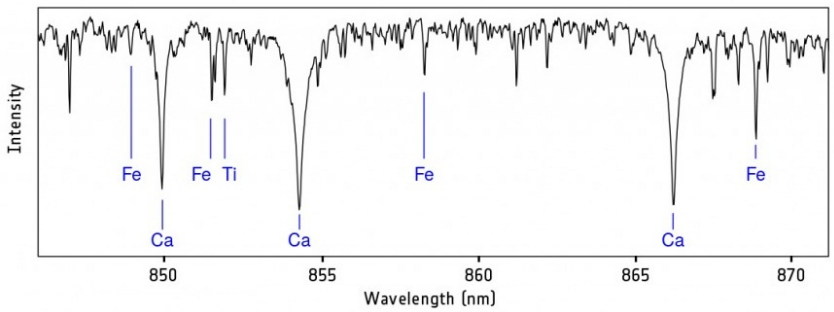


FIGURE 1.10: Radial Velocity Spectrometer observation example from a RVS CCD already with the corresponding wavelength determined for each sample. Additionally, different spectral lines have been identified (Credit: ESA/Gaia/DPAC/Airbus DS)

1.5 Motivation of this Thesis

Most of the work described in this thesis was conducted at the University of Barcelona in Spain. Since the beginning my work at the university has been devoted to the Gaia data processing. After two years, when I finished my Master's degree on *Computational and Applied Physics* I started this thesis for the fulfilment of the degree of *Doctor of Physics*.

In this thesis we address the implementation of the Intermediate Data Updating (IDU) system. This implementation is not only referring to the actual design and coding of the system but also to the management, scheduling and coordination of all the related development and the teams contributing to IDU. Since the very beginning of this thesis, a lot of work has been devoted to chase the continuous changes in the instrument and the processing algorithms affecting ultimately the final design of IDU. This circumstance is clearly evident in the contents of this work.

In order to gain a better understanding of IDU goals, it is mandatory to understand most of the Gaia instrument operation but also the basis of the general astrometric data reduction processing. In this sense, the IDU implementation presents a variety of interesting challenges; covering the purely scientific problems that appear in any data reduction but also the technical issues for the processing of the huge amount of data that Gaia will provide. The design of IDU is additionally exciting from the point of view of an engineer like me. IDU will run in the most powerful supercomputer in Spain, Marenostrum, hosted in the Barcelona Supercomputing Center (BSC).

During this thesis, we have actively participated in most of the major decisions that have been taken in relation to the astrometric processing – core of the data reduction system of Gaia – which have provided us with a very good knowledge of the calibration of the astrometric instrument and the processing of its related data. We have tried to include in the present work most of this knowledge and we expect you find this information useful for a better understanding of IDU and in general the Gaia data processing.

1.6 Structure of this thesis

This thesis has been divided in seven chapters. The current one just aiming to provide a general overview of the Gaia mission and to introduce the main goals of this thesis.

In Chapter 2, we give an overview of the design of the Gaia spacecraft, telescopes and instruments. Specific sections are devoted to the features of the Gaia CCD detectors as well as the main issues identified during the test campaign carried out while the instruments were build. Later, the principle of astrometric measurement with a scanning satellite is described and discussed, including a brief description of the Gaia telemetry stream. The scanning law is then explained together with the spacecraft attitude representation. Finally, we include the recipe for the determination of the final sky coordinates from the observation measurements through the several reference systems defined as part of the Gaia processing.

After this more detailed description of the Gaia instrument operation, Chapter 3 is devoted to the data reduction processing. In this chapter we explain how the complex data processing required for Gaia has been divided and distributed in several systems and data processing centres. We make a clear distinction between the daily and the cyclic processes – processes involved in the iterative data reduction system. Then, we focus on the systems directly involved in the astrometric core processing: IDU, AGIS and PhotPipe. Finally, we explain the data reduction system from the point of view of the data handling – data model, central data repository (Main Database (MDB)), etc. We also include details on how the data is identified, versioned and retracted if needed. The chapter is closed listing the main contributions of this work to some of the topics covered during the chapter.

Chapter 4 provides a full overview of the several IDU tasks. This overview is focused mainly in the scientific topics affecting the design and implementation of the tasks. In the different sections the main data dependencies are identified as well as the role of each task product within the astrometric

reduction solution. This chapter additionally covers the several validation and monitoring tools available for the analysis of IDU outputs.

After the scientific description, the foremost technical requirements and constraints for the IDU integration at the Marenostrum supercomputer are covered in Chapter 5. In this chapter, we also summarise the most relevant standards and guidelines (to be followed by all the Gaia data processing systems) and how they have been implemented in the case of IDU. Additionally, specific sections are devoted to the processing decomposition approach, the execution framework and the data access layer.

At this point, most of the work originally planned for this thesis has been already covered. However, a remaining Chapter 6 has been included describing the very first operational execution of one of the main IDU task, the Cross–Match (IDU-XM). This operational execution was not originally scheduled but due to several issues in the daily processing, this early execution was requested. This chapter summarises all these issues and presents the processing strategy approach adopted for serving the results in a very tight schedule. The results are also reviewed, including a summary of the main findings and issues found.

Finally, in Chapter 7, a summary of the contribution of this thesis is given, including the foremost conclusions and presenting the most relevant pending features and tests that should be addressed in the near future.

2 Instruments Overview and Spacecraft Operation

Generally speaking, Gaia is an spacecraft holding two combined space telescopes, including several mirrors, a hundred of CCDs and a lot of hardware and software pieces which must work in perfect harmony to achieve the mission goal. In this sense, Gaia is a really complex auto-calibrated instrument [Lindegren and Bastian, 2010] where it is mandatory to know perfectly the status of each spacecraft part to be able to obtain the envisaged precision and accuracy on the final catalogue parameters.

This chapter provides specific details on the operation of the several spacecraft parts including their inter–dependencies. It also introduces the main issues encountered during the design and building of the instrument which have played an essential role in the main calibration and processing systems design.

Firstly, the design and operation of the spacecraft and the instruments are reviewed, with the requirements of scientific applications in mind. In particular, the telescope and instrument assembly is described in high detail. Later, the principle of astrometric measurement with a scanning satellite is described and discussed. Finally, we cover the main issues identified during the instrument build and the test campaigns, namely:

- CCD Image Distortion; caused by the Charge Transfer Inefficiency (CTI) or Charge Distortion as explained in Section 2.3.

- CCD Bias Non-Uniformity; introduced during the image read–out as described in Section 2.4.

These two issues are treated in detail due to their relevance when determining the astrometric Image Parameters, key parameters for the determination of the position, proper motion and parallax of the observed objects by the on ground processing systems.

All the topics covered in this chapter are essential to understand how Gaia works but also to understand the requirements and constraints of the data reduction system described in Chapter 4.

2.1 Telescopes & Instrument Assembly

Gaia is composed of two telescopes providing two fields of view, 1 and 2 as illustrated in Figure 2.1. The images of the two fields of views are combined into a single focal plane holding the detectors of the three science instruments (already introduced in Section 1.3) as well as other hardware elements for calibration purposes.

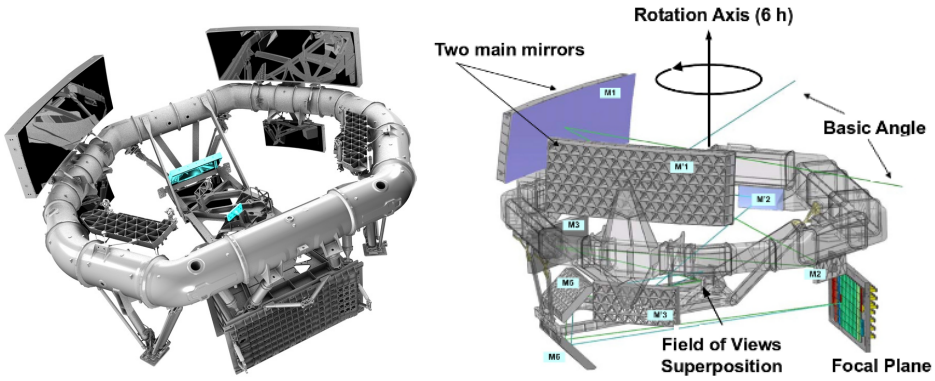


FIGURE 2.1: Gaia instrument illustrations showing the layout of the two telescopes, with their main mirrors on top, and the shared Focal Plane on the bottom right of the torus (Credit: ESA/Gaia)

2.1.1 Basic Angle

The Basic Angle (BA) refers to the angular distance between the two fields of view, 106.5 degrees for Gaia. The stability of the angle between the two viewing directions is a critical calibration concern for the global astrometry, mainly to the parallax measurements. The parallax determination requires a rigid system of reference which in Gaia is obtained thanks to the wide angle between the two telescopes and by precisely measuring the relative positions of objects from both observing directions. Although the thermodynamic mechanical design of the satellite assures the stability of this BA on large time scales, a more accurate determination is done as part of the data processing.

For the determination of the short-term variations on the BA Gaia incorporates two independent devices, Basic-Angle Monitors (BAMs). BAMs are two optical devices, each directing a pair of low-intensity laser beams towards its respective primary mirror. The beams produce two sets of interference fringes that are detected by dedicated CCDs on the focal plane (Figure 2.2). The beams form in principle a stable reference against which any mechanical fluctuation can be detected as a relative displacement of the two sets of interference fringes. By including these measurements in the data processing chain, the astrometric effects of short-term variations can be mitigated. From the two BAMs, only one is used an the other is a spare.

The in–orbit performance and early results are described in Mora et al. [2014]. These results indicates that the BA may not be as rigid as expected which has implied the implementation of more sophisticated calibration models than initially anticipated. This BA issue is further discussed in Section 3.2.1.

2.1.2 Focal Plane

The Gaia focal plane holds the detectors of the three science instruments as shown in Figure 2.2 and is operated by seven Video Processing Units

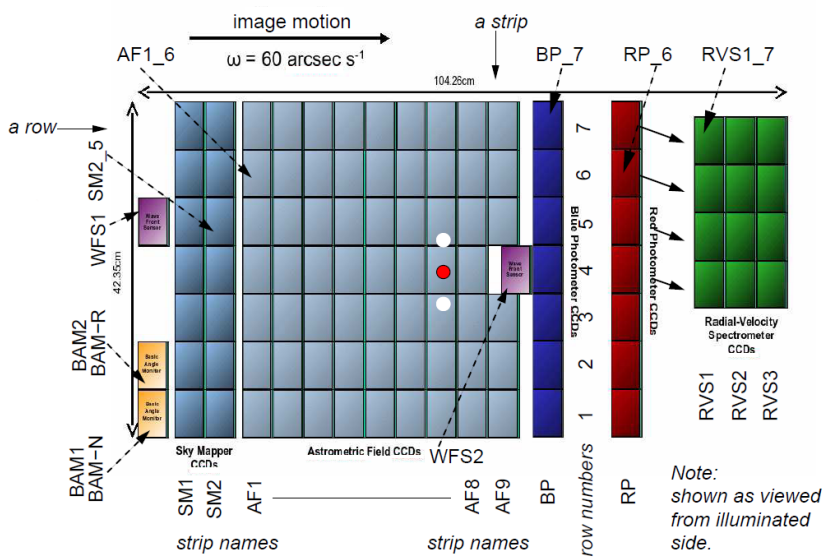


FIGURE 2.2: Layout of CCDs in the focal plane of Gaia. Images travel from left to right, crossing in turn the SMs, AFs, BPs, RPs, and (some of them) the RVSs. Also shown are the CCDs for the BAMs and WFSs (Credit: Gaia/DPAC)

(VPUs). Each VPU operates a single row of CCDs of the Focal Plane and is in charge of the processing of the measurements of the science instruments.

We can distinguish five groups of CCDs according to their technical and scientific functionalities [ESA, 2015c]:

- The WFS sensor and BAM, covering 2 plus 2 CCDs. WFS CCDs are used for re-aligning the telescopes in orbit to cancel errors due to mirror micro-settings and gravity release and BAM CCDs for continuously measuring fluctuations in the BA between the two telescopes.
- The SMs, containing 14 CCDs (7 for each telescope/field of view), which autonomously detect objects entering the fields of view. These object transits are then tracked in the subsequent CCDs.
- The main AF, covering 62 CCDs, devoted to angular–position and flux measurements. These measurements are essential to derive the five astrometric parameters: position, proper motions, and parallax

of the observed objects. The first strip of seven detectors (AF1) also serves the purpose of object confirmation.

- The BPs, RPs, provide low-resolution spectrophotometric measurements for each object over the wavelength ranges 330-680 and 640-1050 nanometers, respectively. In addition these measurements enable the on-ground calibration of telescope–induced chromatic image shifts in the astrometry measurements and the astrophysical classification of the objects.

- The RVSs, covering 12 CCDs, collect high-resolution spectra of the brighter objects allowing the derivation of radial velocities and stellar atmospheric parameters.

2.1.3 CCD Detectors

One of the most important hardware components that allows Gaia to reach the desired performance is the CCD. The CCDs are well-known and widely used in optical astronomy for 2D imaging. More details on CCD imaging is available at Williams et al. [1998].

Gaia will observe objects over a very wide range of apparent magnitude and the CCDs must therefore be capable of handling a wide dynamic signal range. Due to this requirement on the object brightness, an Anti-Blooming Drain (ABD) and Gates features were implemented in the Gaia CCDs.

- The ABD allows the CCDs to observe very bright stars at the same time as very faint ones, preventing the charge bleeding to adjacent pixels.

- The Gates allow to adapt the integration time to the brightness of the object. There are different Gates with different effective exposure times available to cover the huge magnitude (brightness) range that can be observed. These Gates are activated depending on the magnitude of the objects and this magnitude is configurable individually

for each CCD section. While a Gate is activated, the integration time of all objects crossing that Gate is affected which may degrade significantly the observations of the fainter objects. The configuration of the Gates is important to get usable measurements for all magnitudes but it must be chosen carefully to guarantee that enough observations are available for the proper calibration on ground.

Additionally, the CCDs comprise two hardware tools to deal with the CTI (later described in Section 2.3). This hardware tools are: a Charge Injection (CI) structure and a Supplementary Buried Channel (SBC).

- The CI structure is located all along the first CCD pixel line; and is capable of generating artificial charges and a gate that controls the number of electrons to be injected in the first pixel line and subsequently transferred across the whole CCD [Kohley, 2012]. The CI were included to be able to temporarily fill a large fraction of the traps present in the CCD and effectively prevent the charge trapping of the following photoelectrons generated and transferred through the CCD.
- The SBC is a second and narrower doping implant on top of the buried channel which creates a deeper potential minimizing the electron–trap interactions in the rest of the pixel volume. The SBC was introduced mainly to improve the measurement of fainter objects [Seabroke et al., 2013].

All these hardware additions make the Gaia CCD one of the most complicated ever manufactured.

A Gaia CCD, due to its large format (40x60 millimetres), has been manufactured by assembling smaller units with slightly different parameters called stitch blocks. CCDs are thus composed by 9 stitch blocks in Across-Scan (AC) direction and 2 in Along-Scan (AL) direction. This block-based manufacturing with the assembly of the CCD in the focal plane, the BA

and the optical path of the light through all the mirrors are another main calibration issues. This calibration, referred as *Geometric Calibration*, enables the possibility to do the transformations between the measurement coordinates to the celestial coordinates which is essential for deriving the astrometric parameters of the observed objects. Chapter 3 describes the usages of the *Geometric Calibration* within the main processing systems and in Section 2.7 an overview of the different Reference System (RS) defined for Gaia is given.

The Gaia CCD features 4500 pixels in the AL direction and 1966 pixels in AC, which means it nearly contains 9 million pixels. Figure 2.3 illustrates the main parameters of the Gaia CCD as well as some of the above commented hardware additions. Here is a very short description of the individual terms used when referring to CCD parameters (extracted from Bastian [2007]):

- A *pixel* is the elementary charge generation and storage element in the light-sensitive area of the CCD.
- A *column* is the set of all pixels having the same across-scan coordinate (i.e. a one-dimensional pixel array extending along scan).
- A *line* is the set of all pixels having the same along-scan coordinate (i.e. a one-dimensional pixel array extending across scan).
- The *read–out register*, also called *serial register* is the special pixel line which is used to dump the accumulated charges to the digitisation electronics.

2.2 Telescopes & Instrument Operation

2.2.1 Observation strategy

The spacecraft operates in a continuously scanning motion and during the scan the CCDs integrate continuously the star images crossing both

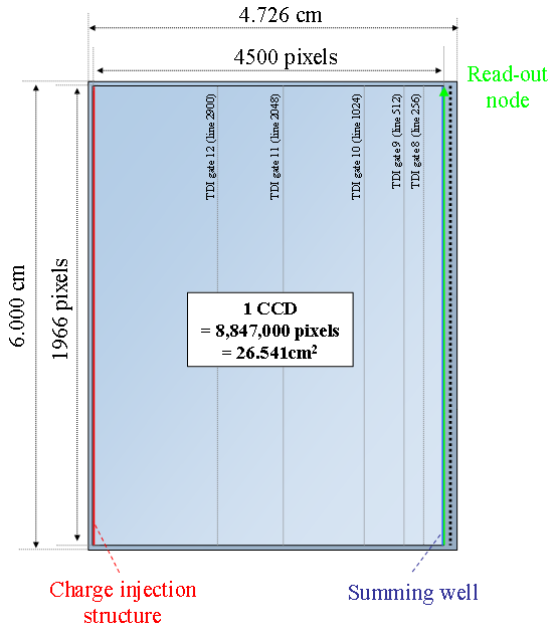


FIGURE 2.3: Schematic view of CCDs structure with physical and pixel dimensions. It also shows the location of the different Gates, CI and read–out line (Credit: Gaia/DPAC)

fields of view. This image integration is achieved by operating the CCDs in Time Delayed Integration (TDI) mode. The TDI mode is commonly used for imaging applications where an object is tracked by synchronising the charge transfer rate between the CCD pixel lines with the speed of motion of the object. This method allows for the integration of charge as the image moves across the focal plane producing a final image with better statistics. This improvement in the measurement is particularly relevant for faint objects where the Signal-to-noise ratio (SNR) is highly increased. Additionally, as each imaged object is sampled by every pixel in the column, it is essentially detected with the mean efficiency of all the pixels in the line reducing then the non-uniformities between pixels.

As the spacecraft sweeps the sky, the on board processing system is able to detect any object brighter than 21^{st} magnitude as it enters the focal plane. Every object crossing the focal plane is detected either by SM1 or SM2. These CCDs identify and record respectively, the objects coming from each

telescope separately. This is achieved by a physical mask that is placed in each telescope intermediate image in one of the beam–combiner mirrors.

Next, the acquisition of the object must be confirmed. This confirmation can be:

- Conditional; depending on the object detection in the corresponding AF1 observation.
- Unconditional; where AF1 observation is not verified. This is needed for the virtual objects (Section 2.2.3) and to enable the acquisition of those cases where the AF1 observation may not be reliable; saturated objects, gated observations, observations containing a charge injection, close to edge, etc.

This confirmation step eliminates false detections such as cosmic rays and avoids the tracking of Solar System Object (SSO) with very high proper motion.

Once the detection is confirmed, a window or set of CCD pixels is allocated to the object, which is propagated through the following CCDs of the CCD row as the imaged object crosses the focal plane. The actual propagation is based on the spacecraft attitude available on board, from which the position of each object in the focal plane can be predicted.

The object then progressively crosses the eight next CCDs strips in AF, followed by the BPs, RPs, and RVSs detectors (the latter are present only for four of the seven CCD rows).

All CCDs, except SMs, are operated in windowing mode so only selected image regions of the CCD data stream containing objects of interest are read out. The use of windowing mode reduces the computational resources on board, the telemetry data stream volume and also the read–out noise of the measurements.

Depending on the magnitude determined in the detection, an object will be observed with larger or smaller windows, with (2D) or without AC resolution (1D) and with or without binned samples/pixels. These different window strategies are used to increase the SNR and reduce the data volume. Finally, each object gets assigned a priority inherited by the individual windows and samples, which is used when resolving conflicts between overlapping windows.

At the beginning of the TDI period, charge is transferred in parallel from one pixel line to the next one. Once the signal reaches the read–out Serial Register, we have one TDI period available to flush all the integrated pixels. This is done, except for SM by flushing at high speed all non selected pixels before reading the samples of each window. In AF and BP/RP braking samples are inserted before the first sample of each window, except of course where we have contiguous window samples. These braking samples mitigate the effects of the Bias Non-Uniformity or Proximity Electronics Module (PEM) Non-Uniformity (PEM-NU) anomaly.

2.2.2 Pre-Scan Measurements

Each CCD is operated by a dedicated PEM. The PEMs feature the main functions to operate the CCD: CCD biases generation, clock level translation and perform CCD image signal digitisation. The Pre-Scan pixels are a number of additional pixels in the CCDs (columns 0 to 13) which are not fed with photoelectric charges (outside the CCD illuminated area) but nevertheless are always read out.

These Pre-Scan pixels are sampled in AC direction following in general the same pattern of the image section pixels. Only two samples from each TDI are acquired and downloaded in blocks of 1024 consecutive TDIs. The sampling pattern, the samples to select and the period between the blocks of 1024 pairs of Pre-Scan samples are configured separately for each individual CCD although it is in general shared for all CCDs of the same type (SM, AF, BP/RP, RVS).

These measurements are fundamental to determine the so-called bias (offset voltage), dark current and the read–out noise introduced by the PEM of each individual CCD as explained in more details in Section 4.4.

2.2.3 Virtual Objects

The Gaia VPUs can be configured to do some custom fictitious observations producing the so-called Virtual Objects (VOs). The VOs are very useful and they were introduced to ease the determination and calibration of:

- The astrophysical background (see Section 4.5).
- The radiation damage of the CCD exposed through the CTI profile in combination with CI (see Section 2.3).
- The CCD Bias Non-Uniformity response (see Section 2.4).

The number, frequency and properties of VOs are individually configurable for each VPU. In general, the VOs are acquired together with the real star observations but the VPUs can also be configured to only acquire VOs which is done periodically in dedicated campaigns for the study of the radiation damage and the calibration of the Bias Non-Uniformity model parameters. The VO strategy for routine and special observation modes is described in Davidson et al. [2013].

2.3 CCD Charge Transfer Inefficiency

Gaia CCDs suffer from CTI that progressively degrade the image quality and may also degrade the astrometric performance of Gaia if not properly addressed (Prod'homme et al. [2012] and Holl et al. [2012]).

Generally speaking, the CTI is the fraction of charge lost during the transfer from one pixel to the next. During the charge movement process, a little part of the charges are for example captured by traps and re-emitted

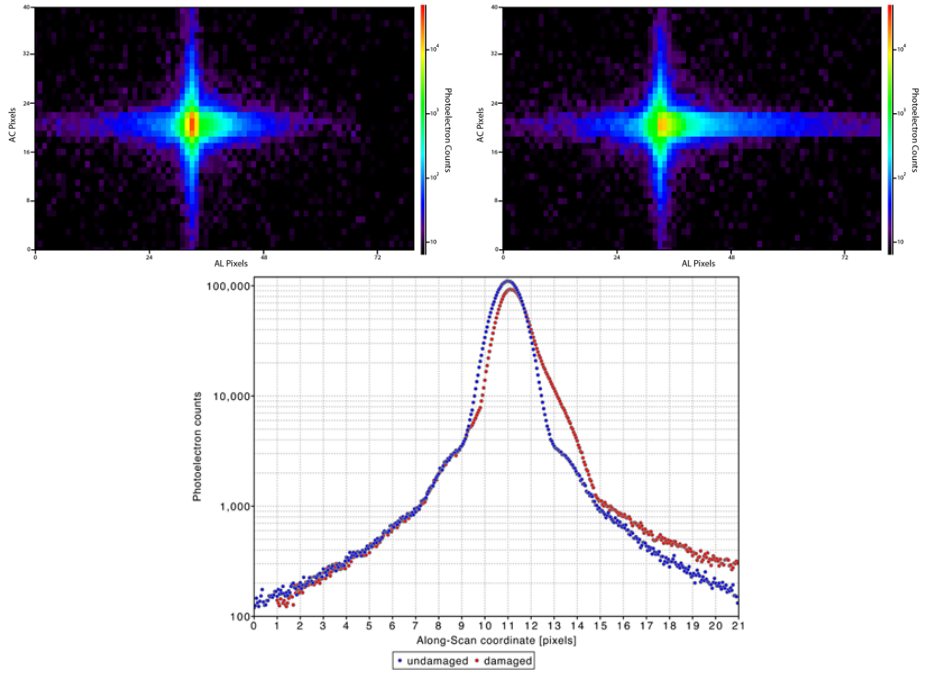


FIGURE 2.4: Example of a simulated Gaia image with and without the CTI effects included. In the right panel (the CTI damaged image) and the bottom panel (AL image profile), it can be clearly seen how the charge profile is distorted: electrons are trapped in the leading edge and released later to form a charge tail (Credit: Gaia/DPAC/CU2)

at a later time, thus distorting the final image acquired (image smearing) introducing systematic bias in the image location estimation and charge loss compared to that of an undamaged/ideal CCD.

The CTI is present in both directions; in the AL direction or TDI direction during image integration and in the AC direction or during image read–out in the Serial Register.

The CTI will progressively increase by the radiation damage dose received by the CCDs during the mission. The radiation may introduce defects on the detector materials creating new traps which will degrade the CCDs response. At L_2, the radiation environment is dominated by energetic protons emitted during solar flares, which are governed by the cyclic activity of the Sun.

In the AL direction and taking into account the expected end–of–mission radiation dose, the systematic centroid shifts of the astrometric images may amount to several milliarcsecond, depending in a complex way on many factors including the recent illumination history of the pixel column [Lindegren et al., 2008].

In the Serial Register, the reports of the test performed on ground (EADS Astrium [2009b] and EADS Astrium [2009a]) confirmed the presence of a non-negligible CTI effect in that direction. These tests also revealed that the response was almost independent of the radiation and the main parameters were the brightness of the star and the distance to the Serial Register output: the fainter is the star and the farther from the Serial Register output, the larger is the AC distortion.

The calibration of the radiation damage of the CCDs is one of the most difficult calibration issues. To ease the calibration and mitigation of the CTI in the AL direction, the possibility of artificially filling charge traps prior to the transfer of signals in TDI mode was implemented in the CCD in the form of CI. These injections have indeed a beneficial effect on the signal distortion due to radiation damage, but they give rise to a charge release signal that inflates the general background signal. This charge release signal must be modelled as part of the background processing, and since charge release is some function of the injected signal at a given column and line of the CCDs, it is also essential to characterise the injection levels as part of the processing [Cross and Hambly, 2010]. Thus, injection levels have to be routinely monitored during the mission to enable injection characterisation as part of the routine background processing.

In principle, all CCDs support the activation of periodic CIs but in SM, due to the need to activate a permanent Gate to reduce the image smearing, the CI is not included in the default baseline [EADS Astrium, 2008]. This fact will imply a worse quality of the parameters derived from SM images but this is not considered dramatical because the final solution will be completely dominated by the AF measurements.

The initial analysis carried out during the commissioning phase shows that the charge trapping and release effects of the CCDs was extremely small and stable. This analysis also confirmed that the current calibration models match the effects seen on the real data received [Cross and Hambly, 2014]. See Chapter 4 for a more detailed information on how the CTI effects are treated and corrected in the data processing.

2.4 CCD Bias Non-Uniformity

The calibration of the gross offsets introduced during the read–out by means of Pre-Scan measurements processing was always planned for and included in the ground segment design [Hambly and Fabricius, 2010]. Unfortunately as reported in EADS Astrium [2009c], it turned out that the implemented read–out strategy introduces signal spikes on the electronic bias correlated with the sequencing of the object being read–out. These fluctuations in the bias are what are referred as the Bias Non-Uniformity or PEM-NU anomaly.

During the read–out of the AF, BP/RP, and RVS CCDs, only the pixels around the target windows are actually read, while the remaining pixels are flushed out rapidly [Fabricius, 2012]. This strategy provides more time for reading the pixels of interest which should in principle improve the measurements quality.

There are two main effects involved in this offset instability. There is an offset depending on the number of flushes carried out immediately before a given sample, and there is an offset after each glitch, i.e. the periods where read–out is interrupted or frozen. The level of the offset fluctuations then decrease gradually while reading the regular samples – increasing again when new flushes or glitches happen.

In order to mitigate this fluctuation, braking samples have been introduced immediately before the target windows in AF and generally also in BP/RP, but not in RVS. The braking samples are read like the window samples and

therefore absorb most of the fluctuation. In AF the price has been that more samples (resources) must be read in the little time available and the read noise is therefore slightly increased.

Fortunately, the fluctuations are predictable, being based on the timing of the samples during the read–out and can be calibrated. The corrections can then be modelled as an unwanted instability of the gross bias offset which is independent of the signal of the observed objects. In order to be able to calibrate the offset non-uniformity, it is necessary to know the pre-history of each sample: especially its situation with respect to the last glitch; if it was contiguous with braking samples or with other science samples; and how many pixels were flushed. These pieces of information are not directly available in the telemetry packages, but must be reconstructed from the list of confirmed observed objects provided in auxiliary telemetry file Object Log (ASD7). For AF2-9, BP/RP and RVS, the effects can be easily corrected [Hambly and Fabricius, 2010]. For SM, it is even easier since the CCD is always fully read out and the effects of the glitches are present always in the same columns. For AF1, the necessary information regarding the read–out history is not available (the non-confirmed objects are not logged in ASD7) and for this reason the read–out in this CCD introduces an additional braking sample to better protect the samples of interest.

2.5 Telemetry Stream

Each confirmed object will produce a telemetry Star Packet (SP1) with the results from AF and BP/RP measurements, and a record in the object log (ASD7). A minor part will also be selected for observation in the RVS and produce a telemetry Star Packet (SP2) with the RVS observation, as well as an additional record in ASD7. Additionally to the SP1, SP2 and ASD7 telemetry packets, Gaia also produces more packet types containing Auxiliary Science Data (ASD), measurements from BAM and WFS and

also additional measurements from SM, AF and BP/RP in special VPU operation modes used for calibration purposes.

Here is the list of the main data produced by Gaia and received on ground [Airbus DS, 2015]:

- Housekeeping data, e.g. attitude and actuators data; with the attitude quaternions, the spacecraft AL/AC rates, the commanded MicroPropulsion Sub-system (MPS) force, the Phased-Array Antenna (PAA) direction and power among others [Bastian and van Leeuwen, 2007].
- SP1 star packets, with science data from SM, AF and BP/RP.
- SP2 star packets, with science data from RVS.
- SP3 star packets, with science data from BP/RP of Suspected Moving Object (SMO).
- SP4 packets, with science data from the BAM.
- SP5 packets, with data from the WFS.
- SP6 packets, with data generated by the VPU in its *Zoom+Gate* mode.
- SP7 packets, with data generated by the VPU in its *Gate* mode.
- SP8 packets, with snapshot data from AF1 in *Service* mode.
- SP9 packets, with snapshot data from SM in *Service* mode.
- ASD1 packets, with the AC shift correction applied to the coordinate of the windows propagated along the focal plane due to the attitude drift.
- ASD2 packets, with Pre-Scan data from all CCDs.
- ASD3 packets, indicating the window resolution changes in RVS.

- ASD4 packets, reporting statistical data (counters) of objects processing.
- ASD5 packets, logging the CI commands for each CCD.
- ASD6 packets, logging the Gate commands for each CCD.
- ASD7 packets, logging the confirmed detections which have in principle lead to the creation of a SP1/SP2.

Some Star Packets (SPs) may never reach ground, either because of the limited resource on board (computational or storage capacity) or because of the limited capacity of the data transmission or due to transmission errors. The ASD packets, on the other hand, are much smaller and have dedicated storage resources and strategy which assures that no data is lost and should therefore reach ground with high certainty. Additionally the ASD packets may be retransmitted in case of a transmission failure. This particular treatment of the ASD was introduced because the data in ASD packets are essential for the full reconstruction of the SPs measurements and for the proper parametrization of some calibration models.

To achieve its scientific goals, Gaia will have to detect, select and measure hundreds of stars per second almost non-stop for the mission life time, producing a prodigious volume of data. Each day, some 50 Gigabytes of data will be generated and these must be sent to Earth. Appendix D summarizes the expected data volume for the full data processing, including the raw data, intermediate data and the final catalogue products.

This extraordinary data volume is achievable thanks to on board data processing and compression, combined with a fast downlink speed (transmitter on Gaia can maintain a rate of around 5 Megabit/sec). However, collecting the faint signal requires the use of the most powerful ground stations from ESA, the 35 meter diameter radio dishes in Cebreros (Spain), New Norcia (Australia) and Malargüe (Argentina).

2.6 Spacecraft Attitude

Gaia builds on the proven principles of Hipparcos to determine the astrometric parameters by combining a large number of one-dimensional (along-scan) angular measurements in its focal plane. A continuous scanning motion ensures that every object is observed in several epochs per year which is essential for the resolution of the astrometric parameters. The capability to make accurate differential measurements over long arcs is the key to obtaining absolute parallaxes as well as a globally consistent reference system for the positions and proper motions [Lindegren et al., 2008].

The nominal scanning law of Gaia (illustrated in Figure 2.5) ensures that each sky region is observed ~ 75 times on average during the whole mission with a nearly isotropic distribution of the orientations of the scanning directions. It also maximizes the uniformity of the sky coverage during the mission operation life.

To obtain accurate astrometry of the sources observed by Gaia a precise determination of the attitude of the spacecraft is required [Hobbs and Lindegren, 2010]. The attitude of Gaia is initially obtained from the AOCS module in the spacecraft and is represented by Quaternions. Quaternions provide a convenient and elegant mathematical notation for representing orientations and rotations of objects in three dimensions. This raw attitude data is received from the spacecraft every second and then a refined attitude is reconstructed on-ground using improved algorithms.

The first attitude processing is just adjusting a B-spline to the received quaternions, called Initial On–Ground Attitude (IOGA). With this IOGA representation we can then interpolate the attitude for any desired time in the data processing tasks on ground. From this initial attitude then more sophisticated algorithms are executed where the raw attitude is adjusted applying the necessary corrections derived from the observations of sources from a dedicated reference catalogue of well known bright sources (Attitude Source Catalogue (ASC)). For this, an initial Cross-Match of the observations is required. This second processing aims to achieve an

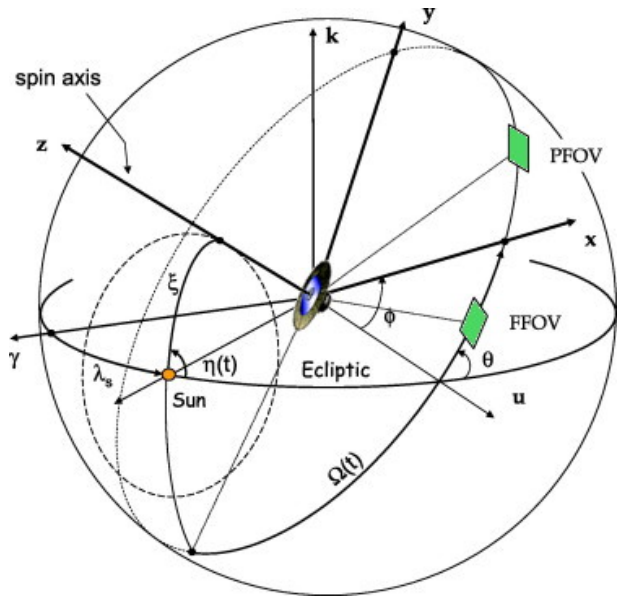


FIGURE 2.5: The nominal scanning law of Gaia. The spacecraft rotates around the z-axis in 6 h so that the fields of view scan approximately a great circle on the sky. The z-axis is constrained to move on a sun-centred cone of 45° half-aperture with a period of 62 days, forcing the plane of scan to sway back and forth with inclinations to the ecliptic between 45° and 135°. The axis of the cone follows the annual solar motion leading to a full sky coverage after six months (Credit: ESA/Gaia)

accuracy of about 50 milliarcsecond. Subsequently, further refinements are performed this time using the huge number of observations available to reach the desired accuracy of less than 20 microarseconds [Hobbs, 2009].

Additionally, Gaia is exposed to *micro–meteoroid* impacts which may cause a change in the angular velocity of Gaia and thus may introduce high frequency noise in the attitude determination. These impacts must be calibrated and corrected in the processes in charge of the attitude reconstruction. During the first year of Gaia operation, small rotation rate changes of the spacecraft were discovered at a frequency of about one per minute which were generally interpreted as micro-meteorite hits. Recently, based on more and better data, they were identified to be mostly due to sudden small structural changes within the spacecraft, called *micro–clanks* Bastian [2015]. At the time of writing this thesis, the treatment of these attitude

jumps due to the *micro–meteoroid* and *micro–clanks* is being studied and a first preliminary approach is outlined in Lindegren [2015].

2.7 Reference Systems

The data processing of Gaia requires a method to convert observations on the focal plane (with plane coordinates of the centroid of the selected images) to suitable celestial coordinates, that is, right ascension and declination, (α and δ). An overview of the several steps and reference systems involved in the process of converting the observed positions on the focal plane to physically sound coordinates is shown in Figure 2.6. Bastian [2007] includes a detailed description of these reference systems but for the scope of this work, it is useful to briefly describe the most significant properties of the several conversion steps.

The top and target reference system for the final catalogue of Gaia is the so-called Barycentric International Celestial Reference System (BCRS/ICRS), which is a quasi-inertial and rotation-free reference system with respect to distant extragalactic objects. To obtain the coordinates of every source in this RS, several additional instrumental reference systems are introduced. The first of these (see Figure 2.6) is the Center-of-Masses Reference System (CoMRS). This RS moves with the Gaia spacecraft and is defined to be kinematically non-rotating with respect to the BCRS/ICRS. This RS orbits around the Sun and causes variable aberration of light, varying coordinate velocities of the observed celestial bodies and other undesired effects. The transformation of the BCRS/ICRS coordinate direction towards a source into the CoMRS coordinate direction of a light ray coming from that source (and vice versa) is one of the central issues of the Gaia astrometric data reduction system. From the CoMRS, it is useful to introduce a new RS co-moving and co-rotating with the body of the Gaia spacecraft, namely the Scanning Reference System (SRS), which is mainly used to define the satellite attitude. Celestial coordinates in the

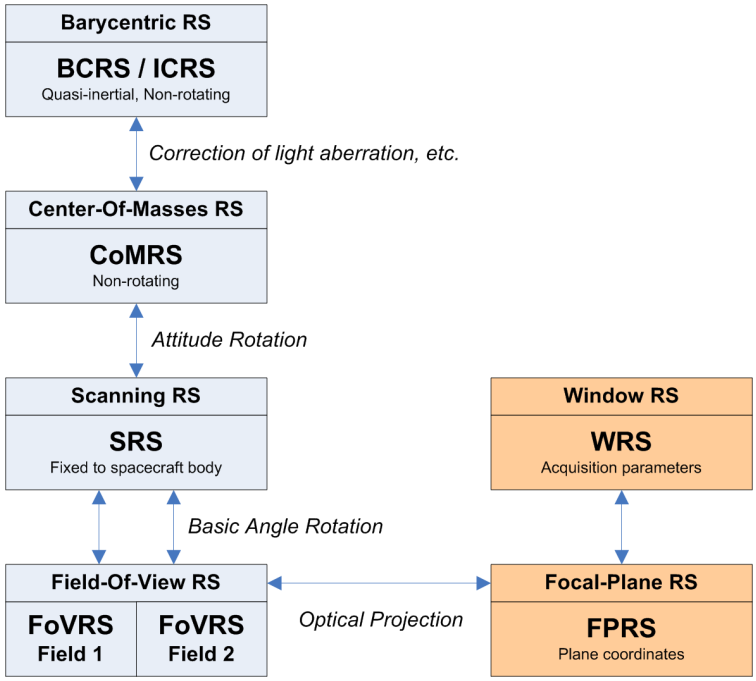


FIGURE 2.6: Overview of the several reference systems (RSs) adopted in Gaia. From the RS for the final catalogue of Gaia to the RS used for the acquisition parameters of the observations within each CCD of the focal plane. RSs on the left can be seen as astronomical RS while the ones in the right refers to purely mechanical RSs

SRS differ from those in the CoMRS only by an euclidean rotation given by the attitude quaternions.

From this point, we need to distinguish two Field-of-View Reference Systems (FoVRS), one for each sky field seen by Gaia, with their origins at the centre of masses of the spacecraft and with the abscissa axis pointing to the optical centre of each of the fields of view (f_1 and f_2 in Figure 2.7). The coordinates in this reference system, called field angles (η and ζ), are defined for convenience of the modelling of the observations and instruments. Celestial coordinates in each of the FoVRS differ from those in the SRS only by a fixed nominal euclidean rotation around the Z axis. These rotation angles are defined by the two viewing directions of the two telescopes of Gaia in the form of the BA, already described in Section 2.1.

Finally, and through the optical projections of each instrument, we reach

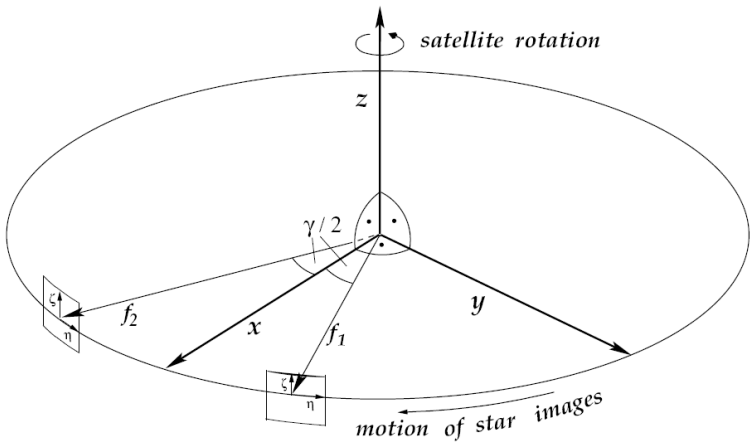


FIGURE 2.7: The x, y and z axes of the Scanning Reference System (SRS), the two Gaia viewing directions (f_1 and f_2), and the Field-Of-View Reference Systems (FoVRS) (η and ζ) for both viewing directions (Credit: Gaia/DPAC)

the Focal Plane Reference System (FPRS). FPRS provides the location of the individual CCDs on the focal plane but also the plane coordinates of the observed image centroids: x and y. From FPRS only one conversion is left to get the corresponding parameters of each individual acquired image in the so-called Window Reference System (WRS) (introduced in Castañeda and Fabricius [2012]). This RS is defined at CCD level and describes the locations of the centroid of the images within each CCD. The coordinates of the WRS are: the *Field of View (FoV)*, the *CCD row*, the *gate*, the *AL pixel coordinate (κ) or observation time*, and the *AC pixel coordinate (μ)*.

The conversion of coordinates between the different reference systems is a critical process within the data reduction systems in Gaia. This conversion process is essential to link the Gaia observations to the source entries in the catalogue and to calibrate the instrument response. During this thesis, we have actively participated in the description and design of most of the developed routines, always focusing in the needs of IDU processing (described in more details in Chapter 4).

2.8 Conclusions & Contributions of this thesis

For a better understanding of forthcoming chapters in this thesis, we have described the most relevant concepts related to Gaia instruments and its operation. We have focused on the astrometric instrument and the foremost issues discovered during the testing of the instrument on ground and during the early mission. We will show you, how these issues have motivated some of the most challenging calibration tasks included in IDU.

Regarding the contribution of this thesis, it is limited to the topics covered in the last three sections: the telemetry stream, the attitude and the reference systems.

During the first year of this thesis, the core processing routines for reading the telemetry were implemented (see Castañeda et al. [2011b], Castañeda and Portell [2009] and Castañeda et al. [2009b]). The undertaking of this task provided an invaluable knowledge on the low level operations of the Gaia VPUs which has been very useful for understanding several functionalities as the CI and Gates and the more fundamental issues related to the CTI.

Additionally, the reference systems were studied in high detail being the major contribution the introduction of a new reference system, WRS, representing the very basic coordinates defined at individual CCD level [Castañeda and Fabricius, 2012]. This reference system definition and the corresponding conversion procedure are essential for the proper interface between the AGIS solution and the LSF/PSF calibration – in the form of the source location within the observed window. This interface is detailed in Section 4.6.2.

3 Data Reduction Approach

The goal of the data reduction is to transform the raw telemetry data into the final science data, consisting of an astrometric and spectrophotometric catalogue based on all the measurements made of each observed object, meeting the final accuracies of mission goals.

The core data reduction consists of several processing systems dealing with raw astrometric, photometric and spectrometric measurements downloaded since the beginning of the mission from the spacecraft. This data reduction starts with a preliminary treatment on daily basis of the most recent data received and continues with the execution of several processing chains included in a cyclic reduction system (shown in Figure 3.1). The cyclic processing chains are reprocessing all the accumulated data again in each iteration or Data Reduction Cycle (DRC), thus adding the latest measurements and recomputing the outputs to obtain better quality on their results [Mercier and Hoar, 2013]. This cyclic processing lasts until the convergence of the results is achieved.

The catalogue releases then will consist on one or more DRCs exercises which will be subsequently published and distributed to the international scientific community [Prusti, 2012]. Updated data release scenario can be checked online on ESA Web Portal [ESA, 2015b].

The Gaia mission requires a Ground Segment (GS) that shall be operative beyond the end–of–life of the satellite. Therefore, the DRC exercises will

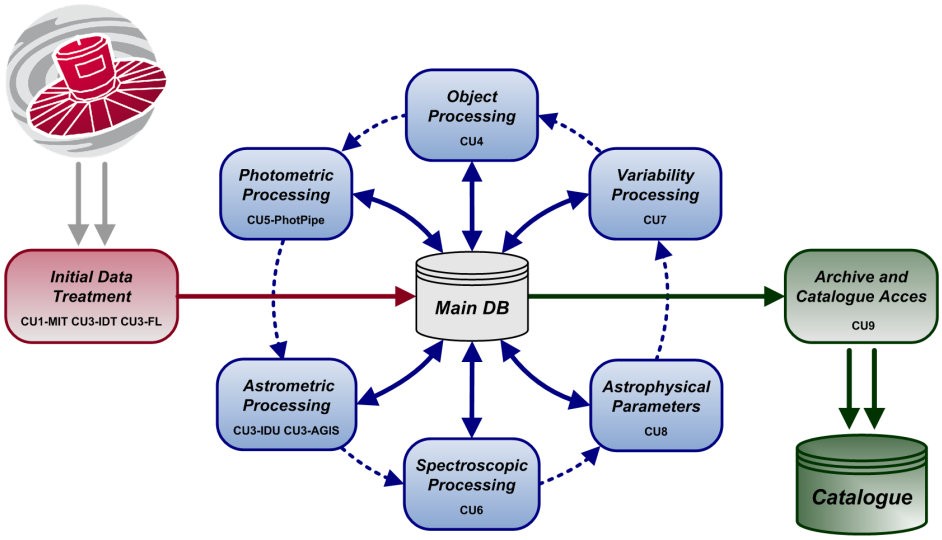


FIGURE 3.1: Gaia data reduction overview; identifying the main systems and differentiating those systems involved in the cyclic processing

continue after routine satellite operations have finished. These DRCs will finish when the mission accuracy and precision goals for the final publication are reached.

The GS is formed by six Data Processing Centers (DPCs), managed by the Data Analysis and Processing Consortium (DPAC). DPAC is the European Consortium responsible for the processing of the Gaia mission data. These responsibilities are specifically:

- preparation of the data analysis algorithms to reduce the astrometric, photometric, and spectroscopic data within a coherent and integrated processing framework, including special objects such as multiple stars and minor planets.
- generation and supply of simulated data to support the design, development and testing of the entire data processing system.
- design, development, procurement and operation of all aspects of the hardware and software processing environment necessary to process the mission data throughout the simulation, mission operations and final catalogue production phases.

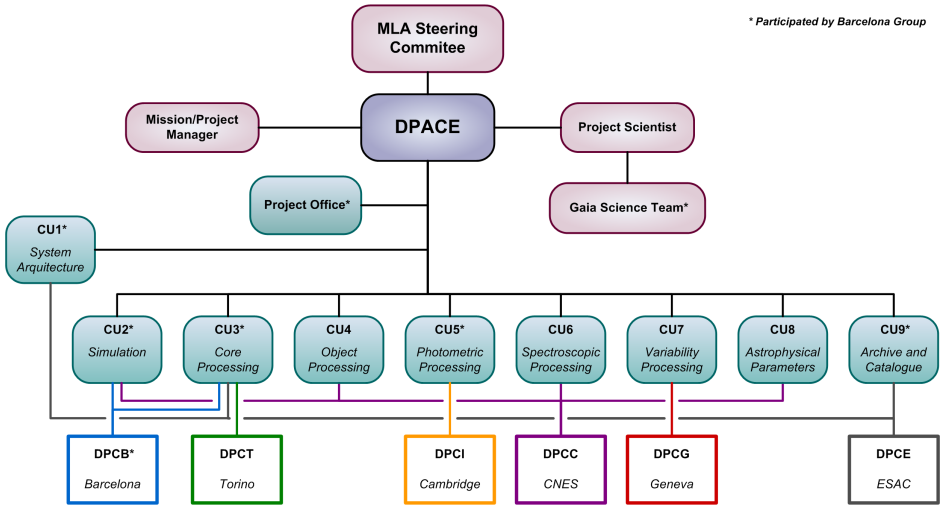


FIGURE 3.2: CUs and DPCs within DPAC

- design, development and operation of the Gaia database and archive, which will contain the intermediate and final mission products of interest to the scientific community at large.

The DPAC is funded through national funding agencies of the participating ESA member states. These funding agencies have signed a Multi-Lateral Agreement (MLA) with ESA which commits all parties to fund the DPAC effort up to the completion of the final Gaia catalogue, expected around 2022. The Data Analysis and Processing Consortium Executive (DPACE) supervises the formation and coordination of the different parts of the Consortium and coordinates the work with ESA. The DPACE additionally reports to the MLA Steering Committee with representatives of ESA and the partner funding agencies. In parallel, ESA selected a senior body, the Gaia Science Team (GST), representing the scientific community.

The complexity of the data reduction has implied the development of different modules or processing systems in charge of specific parts of the processing (described in Section 3.2). These processing systems are managed by nine Coordination Units (CUs), each one responsible for a particular aspect of data processing. Figure 3.2 depicts the structure of DPAC, including the connection between the several DPCs and the CUs.

In general each DPC is dedicated to a specific part of the processing and thus to a given CU. These are the main task of each CU (see LL: [2007] for the full and more detailed task list):

- CU1, dedicated to define the overall system processing philosophy, architecture and strategy. This CU develops the Interface Control Documents (ICDs), coding guidelines, product assurance plans, configuration management guides and additionally some central/common software libraries. Attached to DPCE.
- CU2, in charge of developing the Gaia simulators that gives simulated data to allow the development and validation of the data reduction of the mission, also generating big simulated samples to the common usage of the other CUs. Attached to DPCB.
- CU3, in charge of the core processing, covering the entire processing chain going from the raw telemetry to the astrometric core solution. Attached to DPCB, DPCE and DPCT.
- CU4, responsible for Objects Processing: its tasks include the processing of the astrometric and photometric data of more complex objects not handled by the astrometric core processing, and specifically: (a) non-single stars (binary and multiple stars); (b) SSOs (asteroids, near-Earth objects, etc.); (c) extended objects. Attached to DPCC.
- CU5, in charge of the photometric processing derived from AF and BP/RP instrument measurements. Attached to DPCI.
- CU6, responsible for all aspects of the spectroscopic processing derived from RVS instrument measurements. Attached to DPCC.
- CU7, responsible for the data processing and analysis of the variable sources observed by Gaia. Attached to DPCG.
- CU8, focus on the extraction of Astrophysical Parameters of the Sources that Gaia will observed. Attached to DPCC.

- CU9, covers the management, documentation, archive architecture and development, data validation, operation and services, education and outreach, science enabling applications and visualisation of the Gaia Catalogue releases [Luri et al., 2013]. Attached to DPCE.

The CU with the most relevant role in the present work are CU3 and CU5, and - on a second level - CU1 and CU4.

More than 450 people across Europe are contributing to the development of the huge Gaia data processing effort, including specialist astronomers, engineers, programmers, etc. The participation of my group in DPAC takes place in the CU3, CU5, CU9, CU2 and CU1. This multidisciplinary participation has been of invaluable help during the elaboration of this thesis.

Following sections in this chapter provide an overall description of the different data processing steps involved in the data reduction, describing the main processing systems, data products and the iterative loop governing this reduction. These topics have been grouped in three sections. The first section describes the Daily or near-real-time processing - systems running on a time scale of approximately a couple of days since the reception of the data from the spacecraft. Whereas, the second section covers the Iterative Reduction Processing, or DRCs processing. Finally, a last section has been included to describe how all the received and produced data is managed within DPAC.

3.1 Daily Processing

The downlinked science data along with the relevant housekeeping data from spacecraft is received by Mission Operation Centre (MOC) which is located at the European Space Operations Centre (ESOC) in Darmstadt, Germany. This data is then relayed to the Science Operation Centre (SOC) located at Data Processing Center ESAC (DPCE). When data arrives at DPCE, it is first reconstructed by composing measurement packets from

the raw telemetry packets, decompressed and backed up. Then it enters the first of the scientific data processing systems, which is called Intermediate Data Treatment (IDT) [Portell et al., 2014a]. The IDT processes all newly arrived telemetry and auxiliary data. The most important operations executed by IDT are the following:

- Rearranging and reformatting the data (SP and ASD packets) to create raw astrometric, photometric and spectrometric information ready for storage in the MDB. This part is not strictly data processing since there is no change or addition to the information content, it only reconstructs and transforms the measurements performed on board to a more friendly Data Model (DM).
- Refining the raw attitude quaternions of the spacecraft by combining them with the astrometric observations of a subset of bright reference sources, called On–Ground Attitude (OGA1).
- Astrometric and photometric image parameter: transit times, centroids, fluxes, colour wavelength, etc. In this stage, also the electronic bias and the astrophysical background is determined.
- Approximate celestial coordinates for source identification; linking the observations to sources present in the Gaia catalogue or else to new ones if necessary.

All the IDT processing is done in near-real time as soon as data reach DPCE. The processing is based on a data–driven approach; IDT has to detect and identify the incoming data and trigger the associated processing module - which, in turn, can trigger other processing modules. This operation allows to promptly examine the data to diagnose the health of all its systems and instruments. A quick examination is possible thanks to several integrated monitoring tools developed during this thesis described in more details in Section 4.8. IDT software design and development is managed by the CU3-UB group and integrates developments from several teams; CU5-DU10, CU3-Torino, etc.

Together with IDT it also runs First Look (FL). The primary objective of the FL is to ensure the scientific health of Gaia. It produces a lot of diagnostics indicating if there are anomalies in the scientific output of the satellite which can be corrected on-board. FL processing carry out a restricted astrometric solution on a dataset from a small number of great-circle scans. This solution generates a new refined attitude (OGA2), and a geometric calibration and allows to detect as soon as possible inconsistencies on the operation of the spacecraft instrument or in the processing performed by IDT. Additionally, FL performs some calibration tasks regarding:

- Charge Injection (CI) and CI release profile (see Section 2.3).
- PEM Non-Uniformity (see Section 2.4).
- Instrument Line/Point Spread Function (LSF/PSF) response.
- CCD cosmetics: column defects, saturation levels, Column Response Non Uniformity (CRNU), dark current signal, dead columns, etc.

These calibrations are just initial solutions for prompt feedback and usage in IDT and most of them are refined and improved in the iterative reduction processing. This improvement comes from having all the data available (non-real time processing) and by means of more complex models.

The outputs of the daily systems are made available to all DPCs and ingested into the MDB. In that sense, DPCE acts as a hub for the GS, being the central interface between MOC and the other DPCs. The daily data volume (from 40 to 100 Gigabytes, see Appendix D), may not look very high for nowadays computing standards, but the complexity lies in the millions of star transits contained in such data and the complex and tight processing dependencies. See Appendix C to get an overall summary of the Gaia Transfer System (GTS).

Another major daily task is the production of Science Alerts [Burgon et al., 2010]. This task run in near-real time as soon as data is made available at Data Processing Center Cambridge (DPCI) and aims to detect:

- unexpected and rapid changes in the flux, spectrum or position.
- appearance of new objects.
- trigger ground-based follow-up to the community.

The first confirmed Gaia science alert, corresponding to a supernova, was discovered September 2014 [ESA, 2014d].

3.2 Iterative Reduction Processing

As commented above, the complexity of the data reduction has implied its decomposition into different tasks or processing systems in charge of specific parts of the processing. The system break down approach is very common for large processing systems to allow a distributed development. The decomposition has been based on the identification of those major processing tasks which may operate in a relatively independent manner limiting the data interdependencies. Practically, all tasks are in fact interdependent from the point of view of the data but from a development point of view, a well defined ICD allows completely decoupled components to be developed and even operated in different locations. This approach was also taken due to the fact that Gaia processing system was developed in many countries and by teams of varying competence [O'Mullane et al., 2006b].

Gaia processing must be fully self–calibrating with very reduced external connections. It must solve together the attitude, instrument parameters and the object catalogue parameters (positions, proper motion, parallaxes, etc.) mainly from the observations and the housekeeping data coming from the spacecraft. Therefore, the same data that are ultimately forming the astrometric catalogue are also used to reconstruct the attitude and to determine the instrument calibration parameters. Similar considerations, but with less entanglement, apply to the spectroscopic and photometric data, where reference stars will be used to make the initial calibrations and

determine the photometric system. For the work presented in the thesis, the photometric and spectroscopic reduction operation is not relevant and therefore it will not be covered.

The main iterative steps are:

Step 0

Reconstruct observational and housekeeping measurements from the raw telemetry packets.
This is done only once by IDT as the data is received by MOC.

Step 1

Treat raw observational measurements to calibrate the Bias and the Astrophysical Background.
Done for the first time in the daily pipeline at DPCE by IDT and improved as part of the data reduction by IDU as later described in Sections 4.4 and 4.5).

Step 2

Apply calibrations obtained from **Step 1** to reconstruct the estimated photo counts of the observed objects from the raw observational measurements. This step basically revert the CCD image signal digitisation, subtract the Bias and Astrophysical Background and correct the PEM-NU of the readout.
In this step external calibrations for the PEM-NU and CCD cosmetics are also applied. This operation is performed in IDT and IDU and it is a prerequisite for any further processing of the observed images.

Step 3

From the reconstructed photo counts of the observed objects (see **Step 2**) the LSF/PSF response of the instrument is determined. The LSF/PSF calibration is one of the more complex tasks. It depends not only on the observations but also in the astrometric solution from Astrometic Global Iterative Solution (AGIS), the photometric solution from Photometric Pipeline (PhotPipe) and the Cross-Match. AGIS and PhotPipe solutions provide essential parameters required

for the proper modelling of the image shape response of the Gaia instrument; such as the AC motion, magnitude and colour of the source, zero-point reference for the model, etc. The full set of parameters and dependencies for the LSF/PSF task are provided in Section 4.6.

A preliminary calibration is done in FL (or by offline processes) but it is IDU the one in charge of computing and progressively improving the final LSF/PSF calibration in each DRC.

Step 4

Using the calibrated LSF/PSF library (see **Step 3**), the Image Parameters are obtained.

This is also done for the first time in IDT for the newly received data but they are subsequently improved by IDU on a DRC basis.

Step 5

Finally from the Image Parameters and the Cross-Match, the attitude, the *Geometric Calibration* and the main astrometric parameters of all objects are recomputed together providing a global update of the Gaia catalogue.

The system in charge of this global update is called AGIS and runs on a DRC basis.

The subsequent execution of all steps from **Step 1** to **Step 5** is what ultimately will make it possible to reach the final mission goals on astrometry. The time scale for the described iteration loop is much longer than that of the near-real-time processing, of the order of six months. Six months is approximately the time required to assure that new observations for almost the full sky are made available for each DRC.

Next sections describe the most relevant DRC systems involved in the core data reduction, namely AGIS, PhotPipe and IDU.

3.2.1 Astrometric Global Iterative Solution

The AGIS or astrometric core solution is the cornerstone of the data processing [Lindegren et al., 2012] and [O'Mullane, 2012]. It provides the focal plane calibration, the improved attitude solution and updates the main astrometric parameters of the Gaia catalogue. AGIS determines six astrometric parameters:

- Position on celestial sphere: α and δ.
- Parallax (distance): ϖ.
- Proper motion: $\mu_{\alpha*}$ and μ_{δ}.
- Radial proper motion: μ_R (only computed for a small subset of sources).

These parameters are determined using (in theory) no a priori knowledge of these quantities but deriving them from observation data alone in a self–consistent manner. The solution constitutes an internally consistent celestial reference frame, but does not coincide with International Celestial Reference System (ICRS). The solution is then transformed into ICRS using a uniform rotation computed using a subset of primary stars and quasars. Recent findings indicate that the BA is not as rigid as expected and that fluctuations observed may not be completely corrected Mora et al. [2014]. This issue may require the inclusion of some additional external references to get rid of the systematic bias on the parallax parameters [Hobbs and Lindegren, 2011].

Resolving the resulting system of equations combining the six astrometric parameters of all sources, the attitude and the geometric calibration parameters is intractable in a direct way with today's computational capabilities [Bombrun et al., 2010]. Basically one would obtain a system with 10^{12} measurements for determining 5×10^9 unknowns in a globally and self-consistent manner. Instead of that, the adopted approach is to resolve each block of parameters separately and iterate globally until reaching a

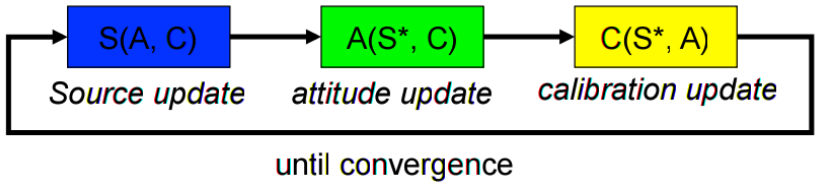


FIGURE 3.3: Simple AGIS iteration diagram where A stands for Attitude, S for source parameters and C for the Focal Plane geometry calibration parameters (Credit: Gaia/DPAC/AGIS)

convergence in the parameters updates. Figure 3.3 illustrates this very basic AGIS internal iteration. For more details on AGIS design and implementation details see O'Mullane et al. [2011]

The geometric calibration model developed for AGIS is in charge of providing the transformation matrix between the focal plane coordinates and the celestial coordinates [Lindegren, 2008a]. This transformation works in both spatial (position in the field of view) and temporal scale (time of the observation). It also tries to disentangle the purely geometric response (GEO-CAL) with respect to the offsets introduced by the colour and magnitude response of the instrument and the other image distortion terms i.e. the CTI effects. The non-purely geometric terms (COMA-CAL) obtained in the AGIS calibration should eventually be removed (set to zero) being completely absorbed by the LSF/PSF model calibration.

AGIS is operated at DPCE.

3.2.2 Photometric Pipeline

Broad–band photometry will result from the fluxes measured in the AF CCDs during the star transits, while the dedicated BP/RP CCDs will provide dispersed images in blue and red bands. The photometric calibrations will ultimately provide the fluxes (magnitudes) and basic spectral information for all observed stars. Combined with the parallax measurements this will provide stellar luminosities, the actual fluxes produced by stars, for stars in a wide range of temperatures, composition and stages of evolution.

The system in charge of the photometric calibrations is called PhotPipe [van Leeuwen et al., 2011]. The PhotPipe data processing has been designed splitting the process into three steps:

Pre-processing:

In this first phase, the raw observations are processed in order to "clean" them from several effects. At this stage Bias and Astrophysical Background is subtracted and possible contamination and blending effects are considered. Also the effects of the CTI are also accounted for in this pre-processing stage.

Internal calibration

The internal calibration will combine all different transits of a given source to a common reference internal system producing a *mean* Gaia observation. This internal calibration accounts for the differential instrumental effects (in CCD sensitivity, flux losses by aperture effect, variations of the LSF/PSF, spectral dispersion and geometry, etc.) and depend on the colour and type of the source. It is worth mentioning that the selection of calibration sources ensuring a good representation of all kind of observed sources is fundamental [Carrasco et al., 2015].

External calibration

Once the *mean* Gaia observations are produced, a final step, the external calibration, transforms them to absolute fluxes and wavelengths. In principle, few calibration sources are needed but they need to have accurate determinations of their absolute fluxes and their non-variability. For this purpose, a big international observational effort has been done using on ground telescopes.

For the astrometric reduction, the photometric processing basically provides the source colour information which is used for colour dependent calibration and for the chromaticity correction of the astrometry. The photometry result must, therefore, be available for the processing of the next astrometric solution in each DRC.

PhotPipe runs at DPCI in Cambridge. Within DPCI, the Source Environment Analysis (SEA) is also run. The SEA software is used to check for the presence of faint companions to sources, which can disturb, or even distort, the astrometric and photometric measurements. This software is to be implemented later during the mission, as SEA requires good data coverage to be successful and eventually could also be used in the IDU processing.

3.2.3 Intermediate Data Updating

Intermediate Data Updating (IDU) is the instrument calibration and data reduction system more demanding in data volume and processing power of DPAC. Intermediate Data Updating (IDU) aims to provide:

- An updated Cross-Match table using the latest attitude, geometric calibration and source catalogue available
- Updated calibrations for Bias, Astrophysical Background and instrument LSF/PSF model
- Updated Image Parameters; location and fluxes.

The design and development of the IDU system has been the main objective of this thesis and thus the next two chapters have been devoted to its detailed description and operation. This section, on the other hand, only covers the role of IDU in the iterative data reduction.

As presented in Section 3.2, the successive iterations between IDU, AGIS and PhotPipe (as shown in Figure 3.4) are what will make possible to achieve the high accuracies envisaged for the final Gaia catalogue.

Basically, IDU incorporates the astrometric solution from AGIS resulting in an improved Cross-Match but also incorporates the photometric solution from PhotPipe within the LSF/PSF calibration obtaining also improved Image Parameters. These improved results turns out to be the starting

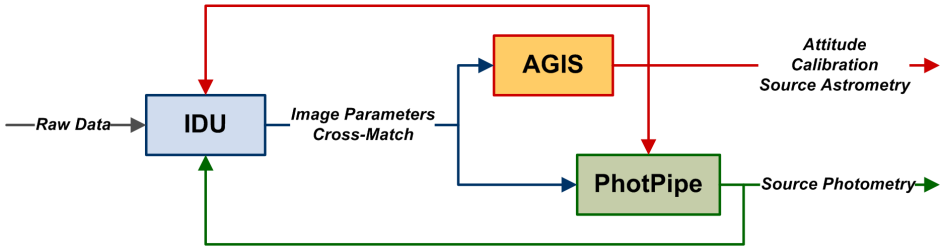


FIGURE 3.4: Diagram of the IDU, AGIS and PhotPipe system operation and interdependency; from raw data until the updated source catalogue updates

point for the next iterative reduction loop. Without IDU, Gaia would not be able to provide the envisaged accuracies and its presence is key to get the optimum convergence of the iterative process on which all the data processing of the spacecraft is based.

IDU software design and development is also managed by CU3-UB group and integrates developments from CU5-DU10 and CU3-Torino.

3.3 Data Management

As commented in this chapter, DPAC has enforced the adoption of a well defined Interface Control Document (ICD) to ensure the intercommunication of all the processing systems participating in the data reduction effort [Hernandez, 2014]. This ICD basically defines the interface or interfaces between all the system through a well defined and controlled Data Model (DM). All the data produced following this DM ends up in the Gaia MDB. The MDB can be understood as a hub of all data produced by the Gaia data processing systems as shown in Figure 3.1.

As mentioned several times, new data is received on a daily basis and enters in the data processing pipeline continuously. It seem obvious that the data – which will reach really huge volumes - need to be partitioned. This data partition is actually useful for the definition and coordination of the data entering the iterative processing. The data partition, in the form of Data Segments (DSs), is defined according to the On Board Mission Time

(OBMT) an nominally will cover periods of approximately six months. The length of the DS will ultimately be adapted to the Gaia release schedule. Finally, the DRC will always be phased with the definition of the DS.

The plan is to version the MDB at regular intervals in phase with the DS and DRC [Hernandez, 2013]. Due to the iterative nature of the science processing, new versions of the MDB will be always derived from the data of the previous version and therefore this previous version will be completely superseded by the new one. For this purpose all the data products are uniquely tagged with a solution identifier - coding at least the DRC and the software that generate the data [Hernandez, 2012]. This solution identifier is also of great help for tracking the input data used to generate subsequent data products or when data qualification is required, allowing for example the retraction or deletion of invalid data.

In general, each DPC stores a fraction of the current and past MDB versions depending on the needs of their processing systems. Only DPCC stores a full version of the MDB, including the raw data from the spacecraft. Once a MDB is closed, a new MDB is initialised, in general with a new DM adjusted to the most recent updates in the calibration models and data product changes.

It will be a normal scenario to have to deal with data from an earlier version of the MDB and DM, typically we need to read the input data with a newer version of the DM. In general, if no breaking changes have been introduced in the DM, the DPAC software is capable of reading old versions of the data. In case of breaking changes, tools are available for the conversion which will be responsibility of each DPC.

All the operations related to the MDB and DM handling are fully described in Els [2014]. This technical note also summarises all the details related to data transfers, data naming conventions and the set of procedures for data conversion and data recovery.

As described in previous sections, during the DRC the different processing systems produce specific data products which are sent to DPCE for its

addition to the MDB. In general, integrating these data products does not imply any additional task except for the data contributing or updating the source parameters of the Gaia catalogue. In practice, it is required the execution of a last process in charge of the integration of the partial source parameter solutions. This system is called MDB Integrator [Hutton, 2014] and is in charge of merging the updates and resolving any conflict found in the data to be integrated. More details are provided in Section 4.3 devoted to the IDU Cross-Match task.

It is only when the MDB Integrator runs, at the end of the DRC, that the current MDB can be effectively closed. DPCE is both, the MDB hosting and MDB Integrator. The expected data volume of the MDB has been summarised in Appendix D whereas Appendix C describes the data exchange scheme and technology adopted for the GTS.

Regarding the Gaia data releases, they will be created from the MDB when a given DRCs is closed, initially only fraction of the catalogue will be extracted but ultimately the full MDB contents will be made public to the scientific community as described in Luri et al. [2013].

3.4 Conclusions & Contributions of this thesis

We have summarised the data reduction approach adopted for Gaia. This summary has covered the most relevant systems starting from the daily processing; IDT and FL and finishing with the main system involved in the astrometric core solution: IDU, AGIS and PhotPipe. We have also described in detail the main steps involved in the astrometric data iterative reduction and the essential role of IDU and AGIS systems.

As remarked on Section 3.2, IDT and IDU have similar features. A large fraction of the algorithms developed for IDU during this thesis has been also integrated in IDT and are actively used in the daily pipeline at DPCE. We must highlight the contributions done for the raw data handling (see Section 2.8), the Cross-Match and the Image Parameters Determination

(IPD) tasks. Also the work done for this thesis has contributed to the improvement of the algorithms provided by other CUs, mainly CU1 and CU5. Additionally, a lot of monitoring tools have been developed during this thesis which have been integrated in both systems. A detailed list of these tools is available in Section 4.8.

It is worth pointing out that the definition of this data reduction approach has been achieved thanks to the efforts of all the CU and DPC teams. We have participated as CU3 members but also as DPCB members, thus assuring the fulfilment of the scientific requirements of IDU but also the technical topics related to the data provision and processing scheduling.

Due to the iterative nature and the interdependencies between all the systems the definition of a detailed operations schedule is fundamental. Also the adoption of a common DM and ICD plays an important role for the success of the integration of all the systems. We have participated actively on the definition of this DM which has required several updates to fulfil the main interface requirements between the data reduction systems, in our case AGIS and IDU.

At the time of writing, the first AGIS run is conducted using already the Cross–Match (IDU-XM) results (see Chapter 6). In a few months, when this data is distributed to all DPCs, the very first iterative loop will start. This is a very relevant milestone and achievement, that should demonstrate the correct interface of all systems developed during the last years.

4 IDU Scientific Overview

The Intermediate Data Updating (IDU) has two main objectives: to refine the Image Parameters for the SM and AF measurements, and to refine the Cross-Match for all Gaia detections using the more recent and thus most accurate calibrations and source catalogues available [Castañeda et al., 2011a] and [Fabricius et al., 2009]. For the achievement of these objectives, IDU also includes some of the major Gaia calibrations tasks which run in the same environment due to the strong relation between them, this symbiosis will facilitate the delivery of suitable observations to the calibrations, and of calibration data to IDU tasks. Therefore, IDU software product does not only refer to the updating of the intermediate data, it also includes the calibration processes and all the processing framework required to make everything work together.

IDU tasks can be logically grouped in three main blocks according to the nature of their processing:

IDU-SDM

This first task group aims to provide information about the connection between Gaia observations and the actual observed sources [1]. It includes three tasks: Scene (IDU-SCN), Detection Classifier (IDU-DC) and Cross–Match (IDU-XM).

[1] The term source refers to both stellar sources and Solar System Objects (SSOs)

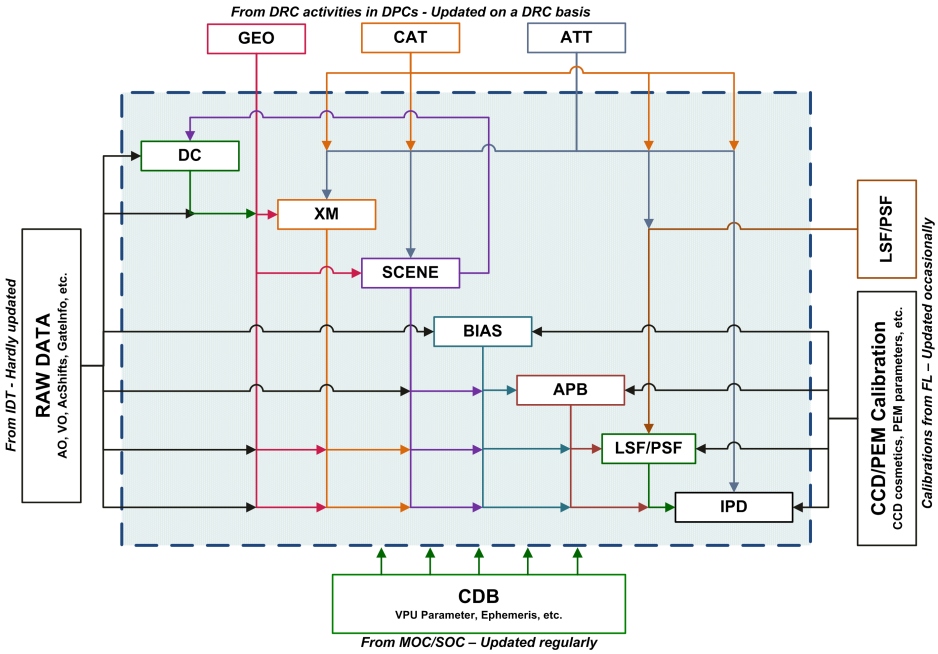


FIGURE 4.1: High level diagram of the data interfaces and tasks involved in a standard IDU execution flow

IDU-CAL

Task group including all the calibration tasks: Bias (IDU-BIAS), Astrophysical Background (IDU-APB) and LSF/PSF (IDU-LSF/PSF).

IDU-IPD

Task group responsible of the determination of the Image Parameters carried out by a single task, namely Image Parameters Determination (IDU-IPD).

Figure 4.1 presents a high level diagram of the tasks involved in a standard IDU execution flow. It also shows the main input interfaces and the interdependencies between each task.

Additionally to the scientific tasks, IDU (through DPCB software facilities, specifically DpcbTools) also implements a number of auxiliary processes. In general, these processes are run just before the IDU main tasks start and once the task results are made available. These processes particularly covers:

- Ingestion of new data in IDU environment.
- Determination of the initial job scheduling according to the requirement and constraints of each processing task.
- Collection and monitoring of the task outputs.
- Ingestion of the task outputs for the other IDU processing tasks.
- Backup of all task output results and configurations.
- Arrangement of output data for its transfer to the other DPCs.

In this chapter, we will cover the core processing tasks while the framework and auxiliary processes will be treated in Chapter 5.

4.1 Scene Determination

The Scene (IDU-SCN) is in charge of providing a prediction of the objects scanned by the two fields of view of Gaia according to the spacecraft attitude and orbit, the SSO ephemeris and the source catalogue [Castañeda and Fabricius, 2012]. It was originally introduced to track the illumination history of the CCD columns for the parametrization of the CTI mitigation. However, this information is also valuable to identify the nearby sources that may be affecting the Astrophysical Background and LSF/PSF profile of a given observation - the IDU-SCN can easily tell us if the transit is disturbed or polluted by a parasitic source.

The Scene will not only include the sources actually scanned by both fields of view but it will also identify:

- Sources without the corresponding Gaia observations. This can happen in case of:
  - Very Bright Sources (VBSs) and SSOs transits not detected in the SM or not finally confirmed in AF1.

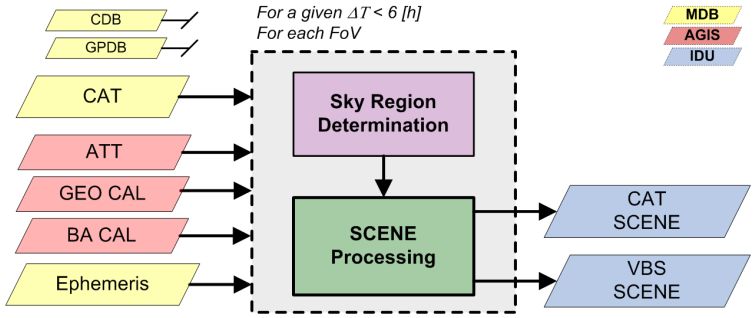


FIGURE 4.2: Schematic data flow for the IDU-SCN task for catalogue sources, showing the main inputs and outputs. In this case, the sky region corresponding to the requested time interval is firstly determined and then the subset of sources is treated by the scene processing core

  - Very high proper motion SSOs, detected in SM but not successfully confirmed in AF1.
  - High density regions where the on board resources are not able to cope with all the crossing objects.
  - Very close sources where the detection and acquisition of two separate observations is not feasible due to the capacity of the VPUs processing.
  - Data losses due to: on board storage overflow, data transfer issues or processing errors.
- Sources falling into the edges and between the CCD rows.
- Sources falling out of both fields of view but so bright that they may disturb or pollute nearby observations.

It must be specially noted that the IDU-SCN is established not from the individual observations, but from the catalogue sources and SSOs. Figure 4.2 and 4.3 present the schematic processing diagrams and data dependencies for each case: catalogue sources and SSOs.

For each object selected, this task provides the reference time when the object has an AL field angle, η, corresponding to the un-gated AF1 Fiducial Line - this time reference is equivalent to the one provided in the raw data from the spacecraft but earlier by half the CCD integration time.

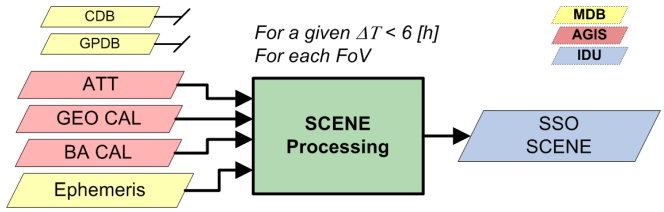


FIGURE 4.3: Schematic data flow for the IDU-SCN task for SSOs, showing the main inputs and outputs

The reference time is obtained by an iterative process as shown in Figure 4.4. The prediction algorithm, starting from a initial observation time (t_{obs}), computes the field angles of the given object. Then it compares the obtained η value with the expected AF1 reference and estimates a new observation time, $t_{obs} + \delta\ t$ taking into account the scanning velocity of Gaia until it reaches the desired convergence (AL/time distance). The time needs not be accurate to better than about one TDI period, AL pixel. It is meant to facilitate sorting and to facilitate direct comparison with the transit identifiers of the actual observations. Finally, the predicted field angles, η and ζ, for additional time offsets covering the full focal plane are included in the scene record.

In general, the main criteria to select an object transit for the IDU-SCN depends on its magnitude and the AC distance to the focal plane centre at the AF1 Fiducial Line. Information about the expected AC size for different Gaia magnitudes can be extracted from Mora et al. [2010]. However, recent findings on the instrument response for bright objects (more described in Section 4.2) indicates that these initial estimations must be largely increased to cope with the diffraction spikes.

In practice, the task splits the full DRC time interval in smaller subintervals of less than 6 hours, expected time to complete a scan of one great circle (6 hours x 60 arcsec/sec = 360 degrees). That way we make sure there will be only one transit of each source in each interval and each FoV.

From the scene data, we can find out which objects contribute to a given transit. The accurate trail across individual CCDs at WRS level can be

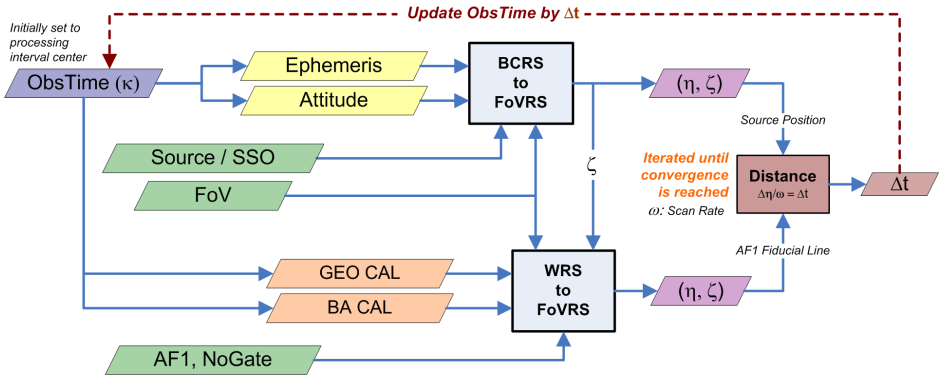


FIGURE 4.4: Diagram of the IDU-SCN core processing

interpolated from the estimated field angles points included in the same record as illustrated in Figure 4.5.

The design and implementation of the IDU-SCN task as well as the corresponding core routines have been responsibility of the CU3-UB, and more specifically of the author of this thesis. Some examples of the results obtained by IDU-SCN task have been included in Chapter 6.

4.2 Detection Classifier

The Gaia on board detection software was build to detect point like sources and it is in principle capable of autonomously discriminating stars from false detections i.e. cosmic rays. For this, parametrised criteria of the shape of the LSF/PSF are used, which need to be calibrated and tuned. Furthermore, this criteria of point like sources was relaxed as otherwise Gaia would not get moving asteroids, little bit extended galaxies or peculiar double star configurations. A study of the detection capability, in

FIGURE 4.5: A scene record parametrizes the object transit providing three predicted knots over the trajectory of the transit over the focal plane in the SRS - from SM (left) to AF9 CCD

particular non-saturated stars, double stars, unresolved external galaxies, and asteroids is provided in de Bruijne et al. [2015].

However, during Gaia commissioning, we detected several kinds of spurious detections and related issues, and in much larger quantities than we expected [Fabricius, 2014b]. In fact, the number of spurious detections was largely increased when it was decided to update the detection parameters on board to make possible the observation of sources fainter than magnitude 20 with few rejections of real sources. Some mitigation procedures were also introduced in the on board detection software [Fabricius, 2014a] beginning 2015 but they are not enough when such faint observations are pursued.

The main problem with the spurious detections arises from the fact that each of them may lead to the creation of a new source in the Cross-Match. Therefore, the goal of the Detection Classifier (IDU-DC) task is precisely to avoid that these detections result in new sources in the catalogue, classifying detections in genuine and spurious and by maintaining a list of blacklisted detections. In other words, IDU-DC results will prevent that spurious sources are created in the Cross-Match and consequently that spurious sources enter other calibration pipelines from other downstream processes.

Here is a brief description of the several categories of spurious detections found in the data so far:

- Spurious detections due to cosmic rays. These are relatively harmless because they happen randomly across the sky.
- Spurious detections due to background noise or CCD cosmetics defects (i.e. CCD bad columns). These are also relatively harmless and normally rare, but depending on the detection criteria, they can lead to a huge number of spurious detections. A study of the probability of this kind of detections is available in Azaz [2014].
- Duplicated detections (essentially double detections) produced from slightly asymmetric SM images where more that one local maximum

is detected. In this case the acquired windows are basically containing the same samples.

- Spurious detections around and along the diffraction spikes of bright sources. Bright sources may easily lead to numerous (from hundreds to thousands) of spurious detections in each transit, especially near the source centre and along the diffraction spikes in the AL direction (example included in Figure 4.7).

- Spurious detections appearing on the other FoV originated from unexpected light paths and reflections within the spacecraft for very bright sources and very close planets. This group of spurious can be seen as ghost detections from those on the original FoV (Figure 4.7).

- Spurious detections from major SSO, mainly planets. These transits can easily pollute arbitrary sky regions with thousands spurious detections (Figure 4.8).

- Spurious detections from extended and diffuse objects. One clear example is the Cat's Eye Planetary Nebula or NGC 6543 shown in Figure 4.6. This case was detected during an IDU test campaign at Data Processing Center Barcelona (DPCB) and it was published as image of the week on December 2014 showing that Gaia is actually detecting not only stars but also high surface brightness filamentary structures.

The detections in the filamentary structures looks point like enough, then Gaia SHOULD detect it

As commented above, the big impact of the spurious detections is an issue recently identified. The current mitigation measures and software modules implemented are still under active development in IDT and IDU. However, it seems clear that in the long run, an effective mitigation scheme should form part of the iteration from DRC to DRC with input from several downstream processes, mainly the SEA from CU5 and the results from CU4.

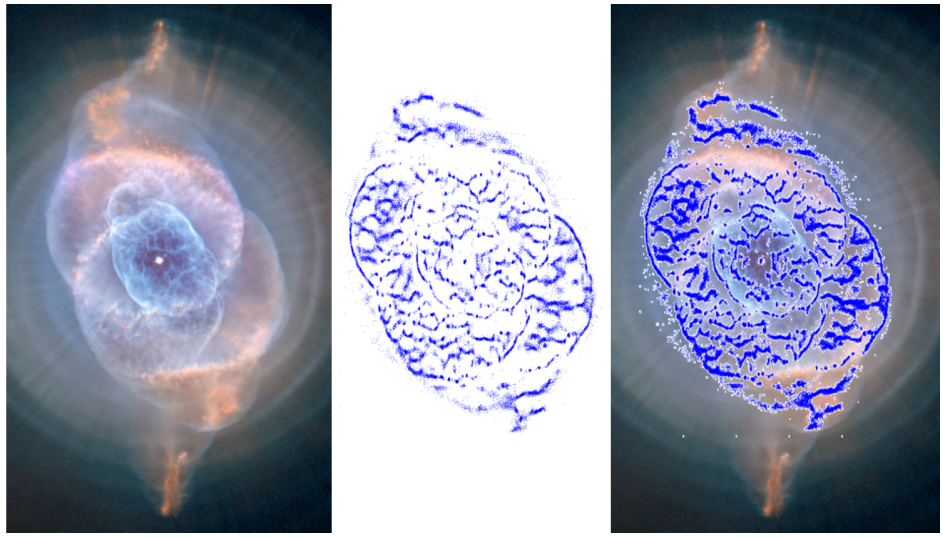


FIGURE 4.6: The Cat's Eye Planetary Nebula or NGC 6543 observed with the Hubble Space Telescope (left image) and as Gaia detections (the 84,000 blue points on middle and right images) (Credit: Photo: NASA/ESA/HEIC/The Hubble Heritage Team/STScI/AURA; Gaia Observation Plot: Gaia/DPAC/DPCB)

Currently the baseline is that each CU will provide its own list of blacklisted or whitelisted detections (list reverting the blacklisted detections) which will have to be combined for the ultimate filtering of the detections.

Each spurious detection case listed above has its own complications and particularities and most of them are still not being treated. Currently only IDT and IDU implement the classification and the filtering of the spurious detections in the Cross-Match.

The current implementation in IDT is just identifying the spurious detections in predefined regions or boxes around the actual observed bright stars [Bestard, 2015]. The process basically looks for the brighter observations in the object log (provided by the ASD7 packets) and select all the observations falling in a predefined set of boxes centred in the parent observation coordinates. The selected observations are then analysed and classified as spurious detections if given distance and magnitude decay conditions are satisfied. These predefined boxes have been parametrized with the features and patterns seen in the real data according to the parent magnitude. This

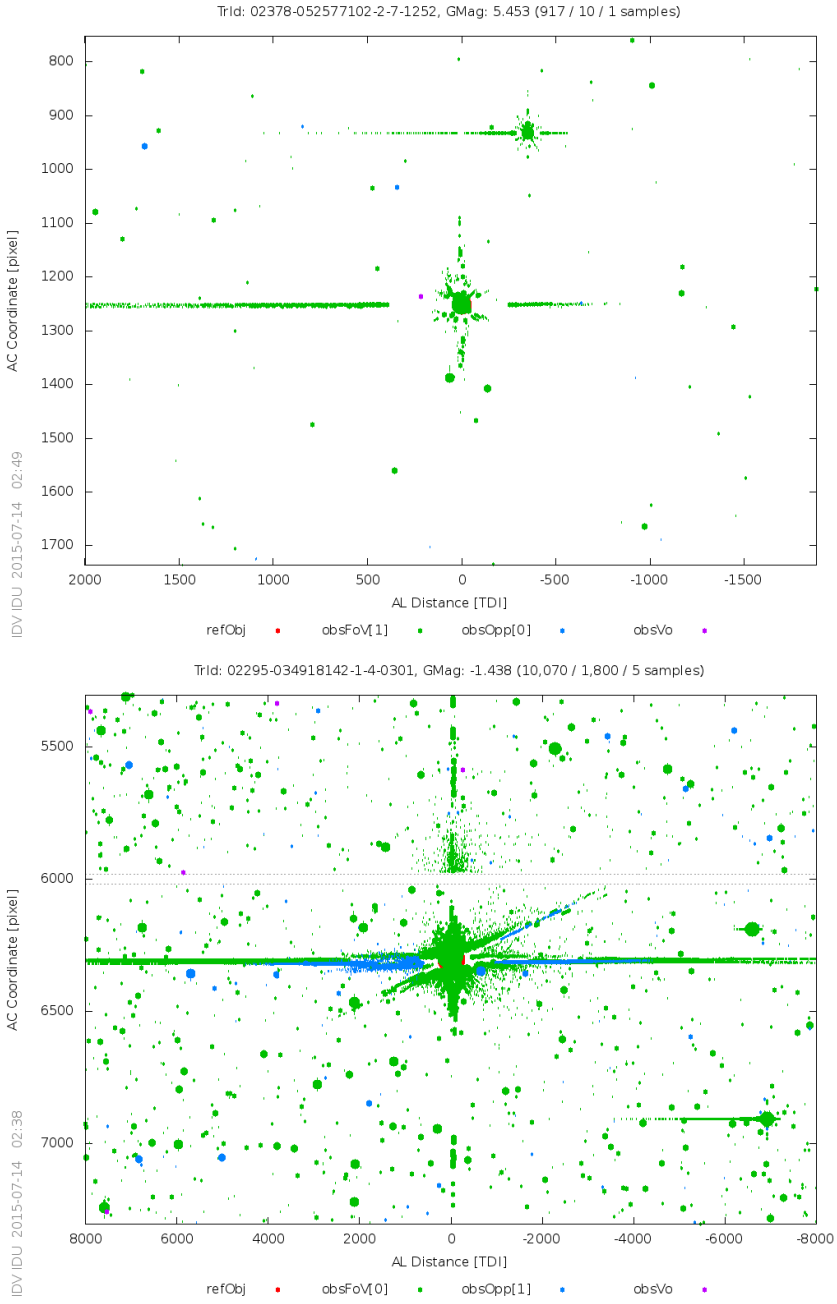


FIGURE 4.7: Spurious detections around a bright source of magnitude 5.4 on the top panel. Very bright source of magnitude -1.4 on bottom panel where it can be seen in blue the spurious detection structures (ghost detections) created on the other FoV

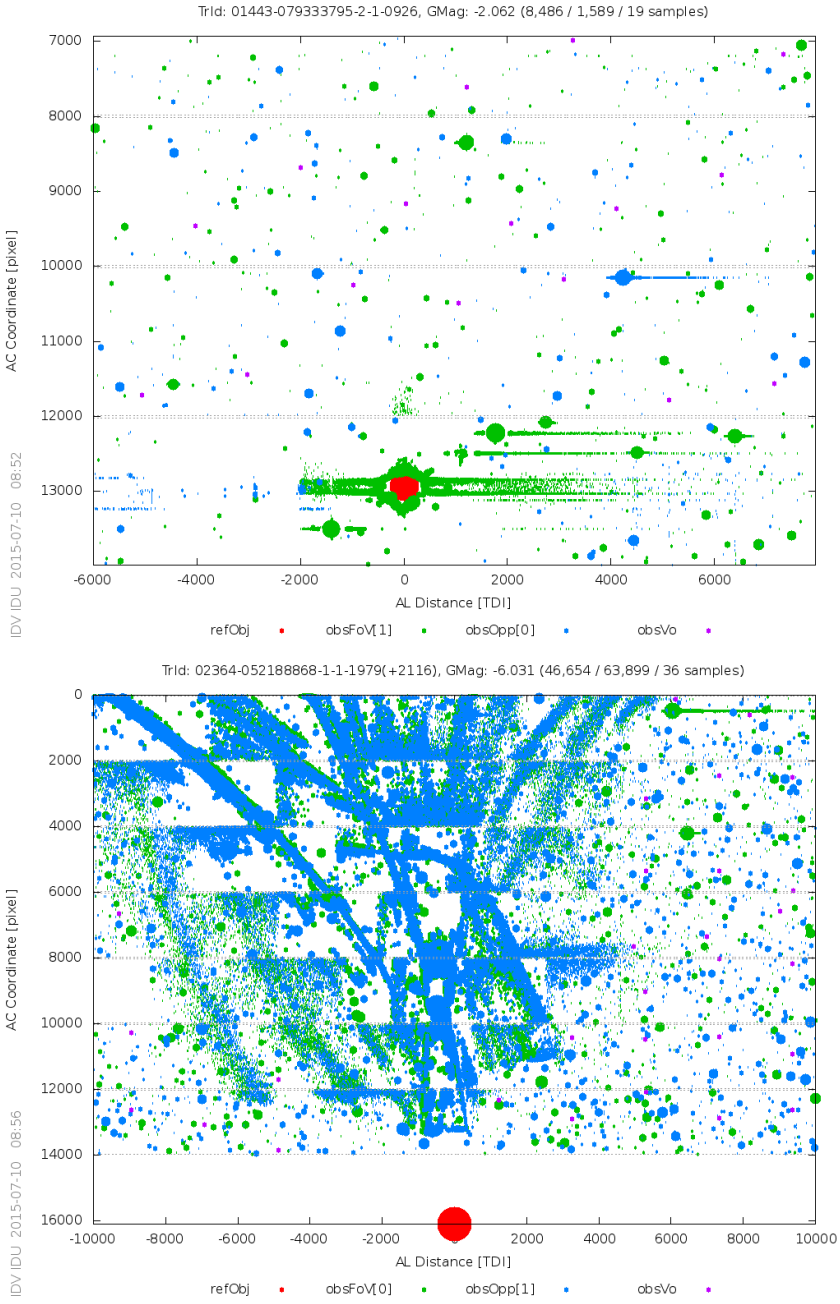


FIGURE 4.8: Spurious detections around Jupiter and Venus (red dots on top and bottom panel respectively). For Jupiter, some of its satellites are also recognizable. In the case of Venus, although the planet is not directly observed by Gaia (being located far below the bottom CCD row) it is producing a large amount of spurious detections in both FoVs

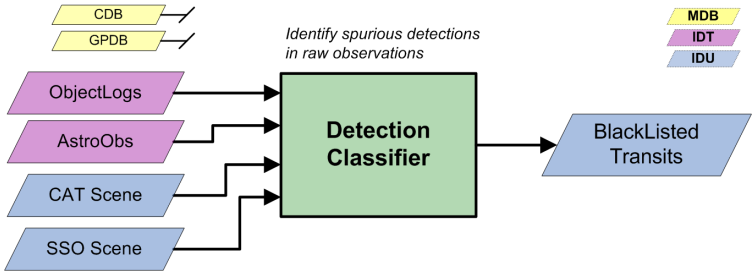


FIGURE 4.9: Schematic data flow for the IDU-DC task, showing the main inputs and outputs

implementation is quite limited and fails to identify quite large numbers of spurious detections. This implementation will hardly be improved in IDT due to processing restrictions, in both resources and introduction of additional dependencies such as the prior computation of some kind of transit predictions as done in the IDU-SCN task.

For IDU the situation is quite different, and a more ambitious solution is being implemented (see Figure 4.9 for a schematic diagram of the processing and data dependencies of this task). First of all, the IDU-SCN results are available which will enable the possibility of identifying more spurious detections cases. The IDU-SCN removes the limitation of only treating spikes of actually observed bright sources, even adding information of sources transiting the CCD edges, which may produce orphan spikes (without a parent observation to trigger the classification as in IDT).

Additionally, the IDU-SCN provides information of the far too bright sources, the SSOs transits and the diffuse objects. These scene records will trigger the corresponding tailored classifications. As an example, for the VBSs and major SSOs we currently plan to filter all faint observations around the predicted transits – even from both FoVs for the transits of Venus.

The treatment of spurious created by faint sources is more tricky since no observable structures like the spikes shown in Figure 4.7 can be detected. In these cases, a multi–epoch treatment might be required to know if they are genuine or spurious detections - i.e. checking if more transits are compatible or resolve to the same new source entry. Additionally, some kind

of feed back from downstream processes such as SEA from CU5 and CU4 solution could be of great help to resolve such cases.

For the remaining cases; cosmic rays, background noise and CCD bad columns, the damage caused is quite limited and we may consider a process that check the observed samples for the presence of any useful signal. This is in any case a low priority task, which will be for sure revisited as the DRCs progress.

In addition, spurious new sources can also be caused by attitude large excursions leading to misplaced detections. These detections are not strictly spurious detections but they are considered as well since they may cause similar problems in the Cross-Match. Consequently, it is highly desirable to identify and clean up these detections during the on ground data processing.

The design and implementation of the IDU-DC task is lead by the CU3-UB, and preliminary results have been also included in Chapter 6.

4.3 Cross–Match

The IDU-XM task is in charge of providing the links of the individual Gaia detections with the entries in the Gaia catalogue (Castañeda et al. [2011a] and Fabricius et al. [2011]). A first Cross-Match is carried out by IDT for the newly arrived observations [Castañeda et al., 2011e]. However, this Cross-Match will need to be updated due to the improvements on the Gaia catalogue, the calibrations and the attitude coming from each DRC exercise solution. Additionally, when IDT resolves the Cross-Match, the data is processed separately in time batches and it is not necessarily complete due to the downlink priority scheme. Therefore, the resolution of dense sky regions or complex cases may be deficient. In fact, detections of high proper motion sources may create duplicated sources that must be merged, while occasionally resolved multiples may need the creation of supplementary source entries. Furthermore, we can obtain much better

results in IDU, not only because we have data completion, and the more recent and accurate calibrations and source parameters available, but also because the spurious detections list may have improved significantly by using the more sophisticated IDU-DC implementation (see Section 4.2).

As already commented, the Gaia catalogue has an astrometric ambition level corresponding to a very small fraction of a pixel in terms of accuracy, and also a fraction of a pixel in terms of resolution. However, the starting catalogue used for the daily processing has been initialised with sources of quite heterogeneous provenance and this provenance must be taken into account when determining the proper source match.

This Initial Gaia Source List (IGSL) has been compiled from the best optical astrometry and photometry information on celestial objects available before Gaia launch: GEPC, GSC2.3, LQRF, OGLE, PPMXL, SDSS, UCAC4, Tycho-2, Sky2000 and HIPPARCOS [Smart, 2013]. This catalogue is expected to be progressively updated and cleaned up during the DRC exercises. Figure 4.10 plots the density of objects included in the IGSL in galactic coordinates. The IGSL has more than 1.2 billion entries with positions, proper motions (if known) and a blue, red, G and G_{rvs} magnitude estimation. Full report of the IGSL is provided in Antiche et al. [2014]. All IGSL sources have been given unique source identifiers, sourceId. This sourceId is basically a numeric field assigned to each Gaia source to ease its identification and spatial arrangement. This numeric field basically codes a spatial HEALPix index, the DPC producer and a running number [Bastian, 2013].

The Cross-Match provides as results a single source link for each detection, and consequently a list of linked detections for each source. This is given in the *Match* Table. Additionally, when a detection has more than one source candidate fulfilling the selected match criterion, we also provide an additional table listing all these candidate sources, called *AmbiguousMatch* Table. In these cases the *Match* Table will still provide a single source match, which we will refer to as the principal match.

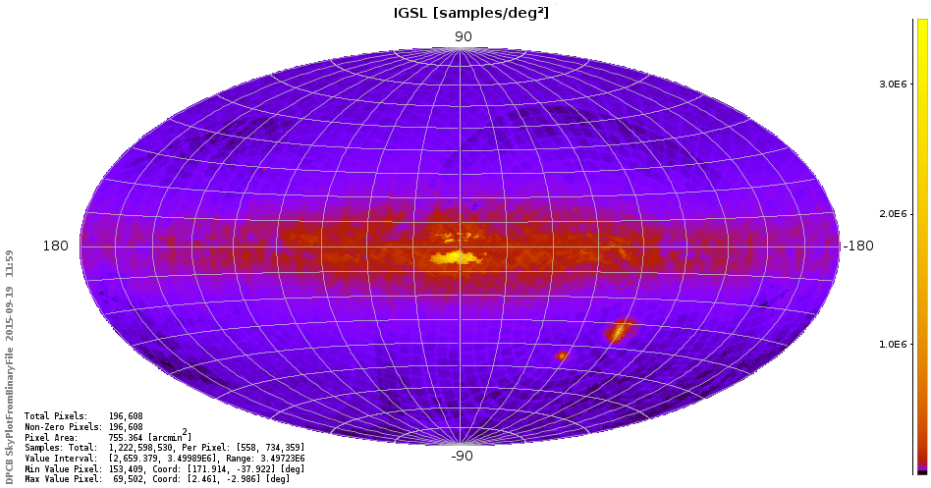


FIGURE 4.10: Density of objects included in the IGSL (galactic coordinates). The square grid visible near the plane is due to differing photometric information and completeness in the overlap of the Schmidt plates that were used for the PPMXL and GSC2.3 input catalogues. The bands that traverse the plane are due to extra objects and photometric information from the SDSS surveys

The resolution of the Cross-Match could require the creation of new sources. These new sources could be derived from already existing sources in the catalogue or created directly from unmatched detections. The newly created sources are logged in the *Track* Table, where the relation to previous sources, if any, is persisted. Section 4.3.3 lists all the currently foreseen cases, including source deletion and source creation in splitting and merging scenarios.

Although the data volume entering the IDU-XM task is small, the number of detections will be huge at the end of mission, reaching $\sim 10^{11}$. Ideally the Cross-Match should handle all these detections in a single process, which is clearly not an efficient approach, especially when deploying the software in the DPCB computer cluster (Appendix A). The first solution that comes to mind is to arrange the detections by an spatial index, such as HEALPix (Gorski et al. [2005], Castañeda [2008] and Castañeda and Fabricius [2010]), and then distribute and treat the arranged group of detections separately. However, this solution presents some disadvantages:

- Treatment of detections close to the region boundaries of the adopted arrangement approach.
- Handling of detections of high proper motion stars which can not be easily bounded to any fixed region.
- Repeated accessing to time-based data such as attitude and geometric calibration from spatially distributed jobs.

These issues could in principle be solved but we have preferred to follow a procedure best adapted to the Gaia operation. The processing approach developed during this thesis has consisted in the splitting of the task in three steps. Figure 4.11 shows an schematic diagram of the three steps, including the main input and output products.

Detection Processor

In this first step, we process the input observations in time order to compute the detection sky coordinates and obtain the preliminary source candidates for each individual detection. Covered in Section 4.3.1.

Sky Partitioner

This second step is in charge of grouping the results from the previous step according to the source candidates provided for each individual detection. The objective is to determine isolated groups of detections, all located in a rather small and confined sky region. Therefore, this step does not perform any scientific processing but simply provides an efficient spatial data arrangement by solving region boundary issues and high proper motion scenarios. In this sense this stage only acts as a bridge between the core time-based and the final spatial-based processing. See Section 4.3.2.

Match Resolver

Final step where the Cross-Match is resolved and the final data products are produced. This step is ultimately a spatial-based processing where all detections from a given isolated sky region are resolved

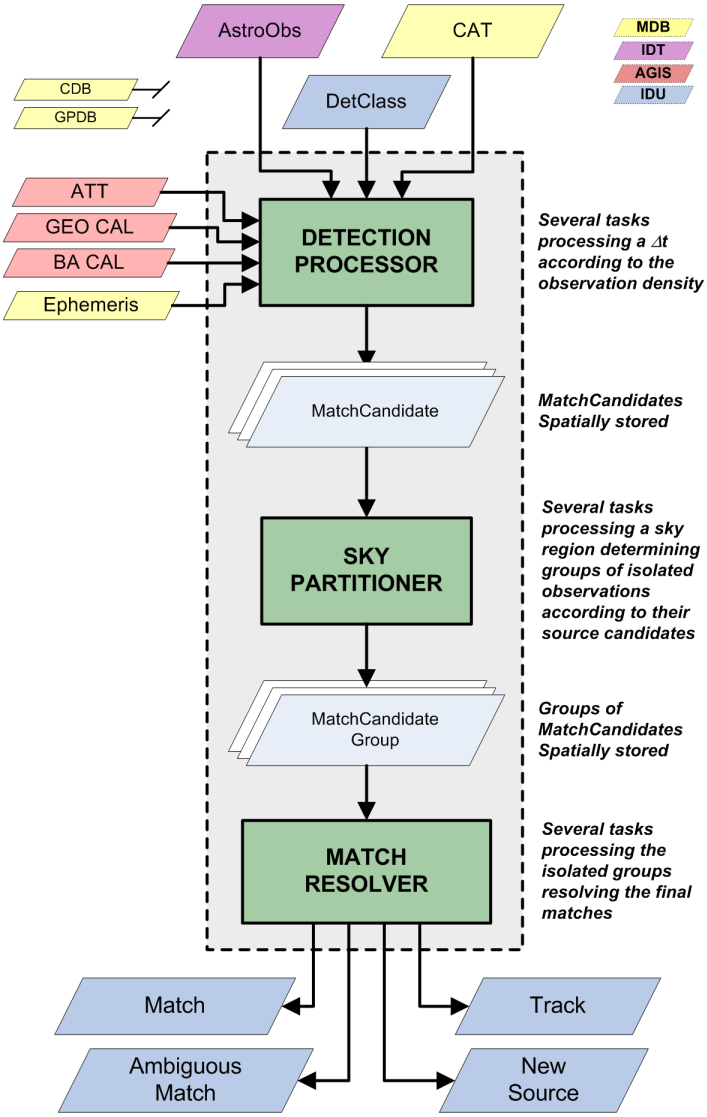


FIGURE 4.11: Schematic data flow for the IDU-XM task, showing the main inputs and outputs

together, thus taking into account all observations of the sources of that region from the different scans. See Section 4.3.3.

The design and implementation of the IDU-XM task is lead by the CU3-UB in collaboration with CU3-Torino. Next sections describe with more details all the operations involved in each one of these processing steps

whereas Chapter 6 covers the results of an early execution of the current IDU-XM implementation with real Gaia data.

To fulfil the task requirements and to be able to implement - in an efficient way - the described modules, we have developed a specific data access layer as part of the IDU framework. The detailed features of this framework will be covered in Chapter 5.

4.3.1 Detection Processor

This processing step is in charge of providing an initial list of source candidates for each individual observation. For the accomplishment of this objective, we have implemented two separated modules: the *Obs–Src Match* and the *Unmatches Processor*.

The purpose of the *Obs–Src Match* module (depicted in Figure 4.12) is to identify for each detection, all the possible matching sources from the Gaia catalogue, producing the so-called *MatchCandidates*. In practice, we split the mission data into small intervals or batches according to the observation reference time. These batches are then processed independently.

The first step over these observation batches, is the determination of the sky coordinates.In principle all Gaia observations enter the IDU-XM, with the exception of VO, and data from dedicated calibration campaigns. Furthermore, all the observations positively classified as spurious detections are filtered out. From the selected observations, the sky coordinates are computed using the reference AF1 acquisition time determined by the detection algorithm on board. This reference time is thus limited by the pixel resolution and the detection error. Eventually, better precision could be obtained using the Image Parameters derived from SM and AF image processing but the pixel binning and the lack of AC resolution for most of the observations (1D measurements) may introduce undesired side effects. Using the reference AF1 acquisition time is simple and robust, and gives a fully sufficient accuracy [Fabricius et al., 2011].

Once we have the observation sky coordinates, we compare the detections with a list of sources. These sources are extracted from the Gaia catalogue such that they cover the band of the sky seen by Gaia in that time interval, and propagated with respect to parallax, proper motion, orbital motion, etc. to the relevant epoch. The capability of predicting the sky seen by Gaia from the spacecraft attitude is an essential functionality to access the data more efficiently and to equalise properly the jobs for their distribution in a computer cluster. We developed this tool, called *AttitudeToHealpix*, as part of the master thesis prior to this thesis [Castañeda, 2008].

The candidate sources are selected based on a pure distance criterion. The decision of only using the distance was taken because the position of a source changes slowly and predictably, whereas other parameters as the magnitude may change in an unpredictable way. Additionally, as commented at the beginning of this chapter, the Gaia catalogue is quite heterogeneous, exhibiting different accuracies and errors which suggest the need of a match criterion subjected to the provenance of the source data. In the later stages of the mission, when the source catalogue is dominated by Gaia astrometry, this dependency could be removed but then the criterion should be updated to take advantage of the better accuracy of the detection in the along scan direction. We can then use separate AL and AC criteria, or use an ellipse with the major axis oriented AC which will benefit the resolution of the most complex cases.

A special case is the treatment of SSO observations. In principle, its processing is responsibility of CU4 and for this reason no special considerations have been implemented in IDU-XM. SSOs will have Gaia Catalogue entries created by IDT and those entries will remain, so the corresponding observations will be matched again and again to their respective sources without any major impact on the other observations.

The *Unmatches Processor* module is only required when we find observations with no source candidates at all after the first *Obs–Src Match* run. In principle this situation should be rare as IDT has already treated all observations before the IDU-XM run. However, unmatched observations

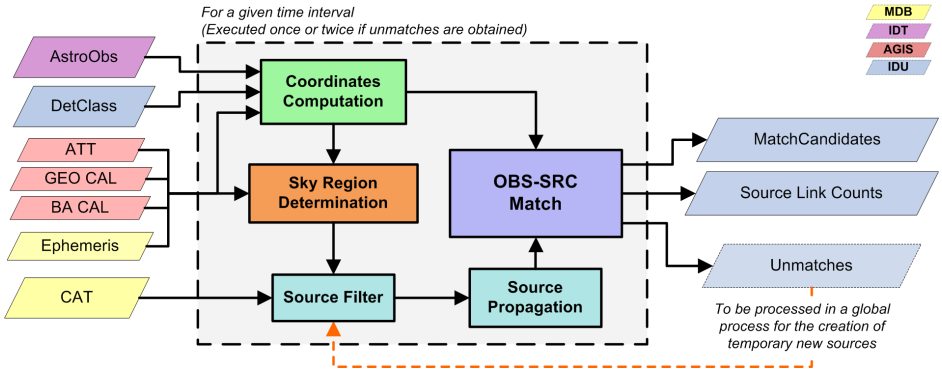


FIGURE 4.12: Schematic data flow of the task in charge of the determination of the *MatchCandidates*, showing the main inputs and outputs

may arise because of IDT processing failures, updates in the detection classification, updates in the source catalogue or simply the usage of a more strict match criterion in IDU-XM. Thus, this module (see Figure 4.12) is basically in charge of processing the unmatched observations and creating temporary sources as needed just to remove all the unmatched observations in a second run of the *Obs–Src Match*. The current implementation is using a very simplistic strategy based on the following recursive recipe:

1. Load all observations for a given sky region.
2. Take the first observation and create and store a new source located at the same position.
3. Take the next observation and check if it can be linked to any of the already stored sources. If not a new source is created and stored.
4. Repeat previous step until no more observations are available.

It is clear that the matching solution provided by this process is far from being optimal but it is more than enough for the goals of this module. These new sources will only be used for obtaining a new set of *MatchCandidates*, free from unmatched observations but still providing reliable and valuable information of the spatial dependencies between the observations.

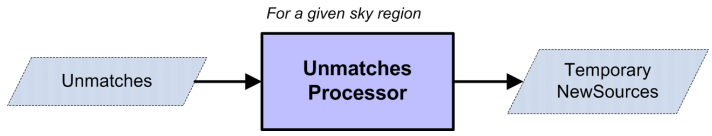


FIGURE 4.13: Schematic data flow of the task in charge of creating temporary sources from the possible unmatched observations obtained after the initial run of the *Obs–SrcMatch* task

The new sources created by this tasks will ultimately be resolved (by confirmation or deletion) in the last IDU-XM step.

Summarising, the result of this first step is a set of *MatchCandidates* for the whole accumulated mission data. Together with the *MatchCandidates*, an auxiliary table is also produced for the tracking of the links created for each individual source. Each job generates its own *SourceLinksCount* table and the absolute link counts can be obtained easily merging all the partial results stored. Results are stored in a space based structure using HEALPix for convenience of the next processing steps.

4.3.2 Sky Partitioner

The *Sky Partitioner* is in charge of grouping the results from the last *Obs–Src Match* run according to the source candidates provided for each individual detection. The purpose of this process is to create self contained groups of *MatchCandidates*. The process starts loading all *MatchCandidates* for a given sky region. From the loaded entries, the unique list of matched sources is identified and the corresponding *SourceLinksCount* information is loaded. Once loaded, a recursive process is followed to find the isolated and self contained groups of detections and sources. In a simplified way the algorithm, illustrated in Figure 4.14, does the following:

1. Take the first *MatchCandidate* and initialise a new group from it, called *MatchCandidateGroup*.

2. For each source listed in the latest *MatchCandidate* added:

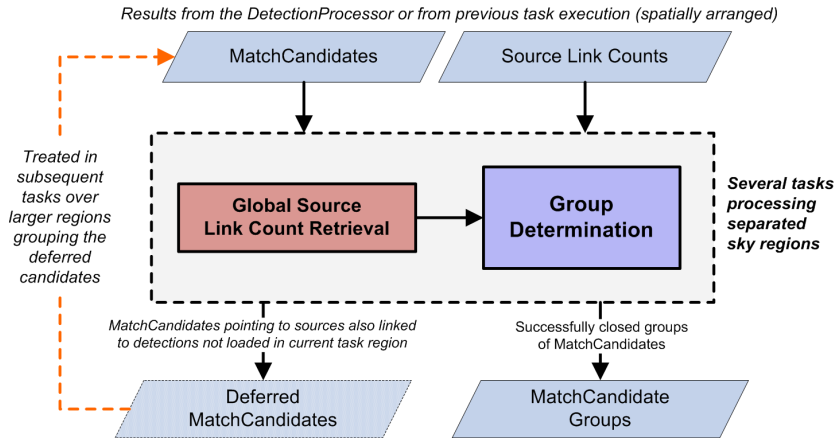


FIGURE 4.14: Schematic data flow of the Sky Partitioner task

- Select detections linked to the current source not already contained in the *MatchCandidateGroup*.
- Check if all detections have been loaded by checking if the absolute source link count equals the count retrieved only from the loaded *MatchCandidates*.
  If *MatchCandidates* are missing – not loaded due to the task region filtering – we abort the current *MatchCandidateGroup* (jump to step 3) storing the already collected *MatchCandidates* for its later processing with a reduced task granularity (larger task regions).
- For each loaded *MatchCandidate*:
  - Add the *MatchCandidate* to the *MatchCandidateGroup* and run again the process from step 2.
- Continue with the next source in step 2.

3. Take the next remaining *MatchCandidate* (not yet processed in previous step 2 run), initialise a new *MatchCandidateGroup* and replay the processing from step 2 onwards.

As it can be deduced from the described algorithm logic, this recursive process ends when we have classified all the input *MatchCandidates* in two groups:

- Resolved *MatchCandidateGroups*; set of *MatchCandidates* grouped according to their links to the same group of sources. This implies that it is not possible to find any observation from a certain group linked to the same source of an observation belonging to another group. In this sense we obtain self-contained and isolated group of detections and sources.

- Deferred *MatchCandidates*; set of *MatchCandidates* which could not be grouped because their companions are in neighbour region.

This process is initially run in parallel jobs following an equalisation of the sky according to the observation density. In the first run, *MatchCandidates* are always deferred mainly because they form part of an agglomeration of observations exceeding the boundaries of the initial task regions. This situation is easily solved by running a new equalisation over all the deferred *MatchCandidates* obtaining a new distribution which should solve the previous boundary issue. Sooner or later, a last run consisting in a single task covering all the sky is always needed. A practical case is described in Chapter 6.

The final result of this process is the set of *MatchCandidateGroups* where all the input observations are included. Figure 4.15 includes two examples of the groups obtained from the execution of this task. This figure has been produced by a tailored tool developed in the frame of this thesis, called *SkyExplorer* (for more detail see Section 4.8).

In early runs, there is a certain risk to end with unmanageably big groups. For those cases we have introduced a limit in the number of sources per group so the processing is not stopped. The adopted approach may create spurious or duplicated sources in the *overlapped* area of these groups. However, as the DRC processing progress, these cases should disappear (groups will be reduced) due to better precision in the catalogue, improved attitude and calibration and the adoption of smaller match radius. During the testing of this task, we have not encountered any of these cases and therefore we can not provided any assessment on the practical limit for the number of sources per group and the amount of cases that could be found.

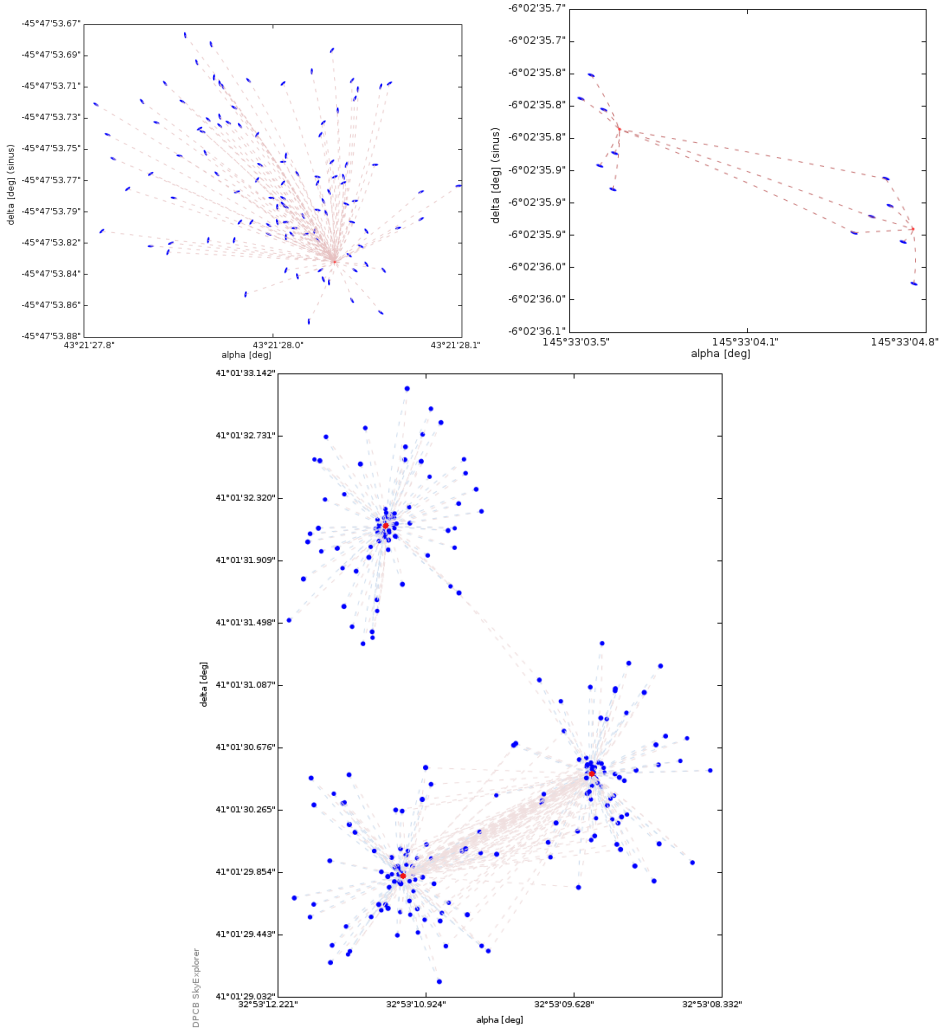


FIGURE 4.15: Three examples of the groups obtained from the execution of the Sky Partitioner task. From left to rigth, top to bottom, the groups contain one, two and three sources respectively. Blue dots corresponds to observations, Red dots are sources and the dashed lines are the links provided in the *MatchCandidates*

4.3.3 Match Resolver

The final step of the IDU-XM is the most complex and it is responsible of resolving the final matches and consolidating the new sources. In that sense, the previous stages are mere preparation steps. Its main inputs are:

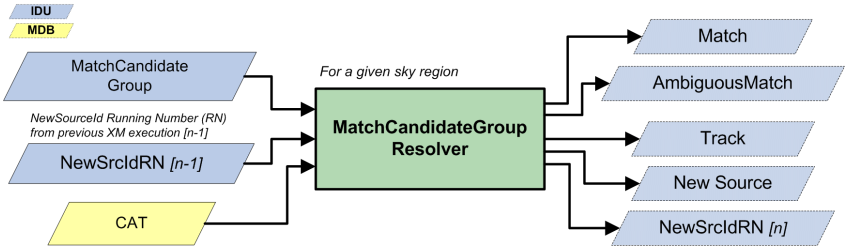


FIGURE 4.16: Schematic and simple data flow of the Match Resolver task

- The *MatchCandidateGroups*, for convenience already stored in a space based structure using HEALPix.
- The Gaia source catalogue, also spatially arranged.
- The *NewSourceIdRunningNumber*, table keeping track of a running number or counter used when creating new sources in a given sky region. The baseline is basically to take advantage of the HEALPix tessellation and its hierarchical numbering scheme, by assigning a separate counter for each pixel at level 6. By using this level, we obtain 49.152 separate counters, each one covering 0.839 square degrees, which taking into account the maximum running number that can be coded and the expected new source density should be more than enough. The sourceId where this running number is coded, also includes a HEALPix index at level 12 and therefore the corresponding counter can be determined using a simple bit shifting of this index.

Two tasks are required to accomplish a final and integral Cross-Match solution: *MatchResolver* and *NewSourceIdConsolidator*. The first one or *MatchResolver* (Figure 4.16) is in charge of resolving separately each input *MatchCandidateGroup* by detecting and resolving the conflicts present among the candidate matches. We distinguish three main conflict cases:

- Duplicate matches: when more than one detection close in time and observed by the same FoV are matched to the same source. In this case we have to deprecate the existing source and create two new

sources, we refer to this as source splitting. After the creation of the new sources, the original *MatchCandidates* must be revised and updated.

- Duplicate sources: when a pair of sources from the catalogue have never been observed together, thus never identifying two detections within the same time frame, but having the same matches. In this case, we would want to merge those duplicates sources. As for the splitting, this involves deprecating existing sources, creating a new one and revising the *MatchCandidates*.

- Unmatched observations: strictly speaking no unmatched observations reach this part of the processing because the first step has already created new sources when necessary. However, those new sources are temporary and they have to be revised. In practice, once a *MatchCandidateGroup* has been created the references to these temporary new sources are removed so the *MatchResolver* is forced to resolve them from scratch, this time having all relevant observations and sources together.

All three cases will imply the creation of an entry in the *Track* table, logging the new source created or the splitting and merging operation of the parent sources. As already commented in Section 3.3, these Track entries will be used by the MDB Integrator for the consolidation of the new catalogue version. The *Track* table allows the migration of fields from the parent sources to the new ones created by IDU. The migration is desirable since IDU will only populate the very basic fields such us the source position and magnitude.

It is not within the scope of the present thesis to document all the details of the different algorithms available for resolving the conflicts and deciding the best matches. Three implementations are available:

- Nearest neighbour solution: provided by CU3-Torino [Spagna and Messineo, 2011].

- 2–by–2 nearest neighbour solution: based on IDT implementation [Castañeda et al., 2011e].
- Clustering solution: still in the very first design and development stage and led by the CU3-UB team [Clotet et al., 2015] (in preparation).

The first two are basically a classification based solution while the new one, in order not to lose generality, aims for the use of cluster analysis. At the time of writing, only the second one is completely operational and has been already successfully executed as part of the first DRC as described in Chapter 6.

The main *MatchResolver* products are the *Match*, *AmbiguousMatch*, *NewSource*, *Track* and the updated *NewSourceIdRunningNumber* tables. The *MatchResolver* execution strategy is based on jobs equalised following the HEALPix-based sky density of the input observations. These jobs are executed in the DPCB computer cluster, being each job completely independent from the rest. This execution strategy is by far the more robust but it may introduce inconsistencies in the form of sourceId collisions. The job equalisation is not restricted to any HEALPix level and therefore different jobs may need to create new sources from the same *NewSourceIdRunningNumber* entry. Other software systems (as IDT) implement a central server in charge of serving the running numbers but this approach was discarded to conform to the general IDU batch processing baseline (see Chapter 5). Instead of the server, we have implemented a final consolidation process, the *NewSourceIdConsolidator*.

The *NewSourceIdConsolidator* (Figure 4.17), is in charge of fixing the duplicated sourceIds present in the data generated by the *MatchResolver* jobs. The sourceId collisions can be easily detected by comparing the entries of the updated *NewSourceIdRunningNumber* generated in each job execution. In practice, this task loads all the *NewSourceIdRunningNumber* for the full sky. The size of these tables is really small and does not introduce any performance complication. Once loaded, it looks for the HEALPix indexes

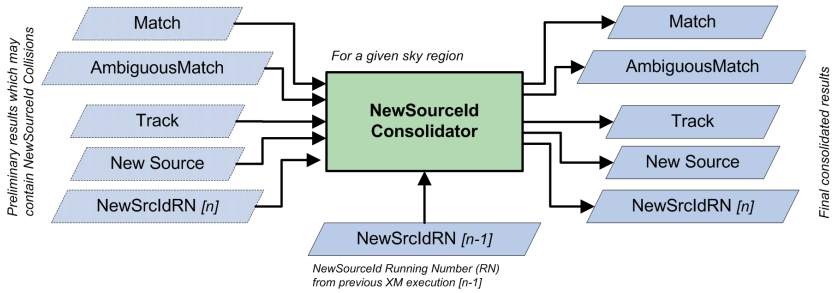


FIGURE 4.17: Schematic data flow of the NewSource Consolidation task

updated by more than one job and determines the offset to apply in the running number of the duplicated sourceIds of each individual job. Tracking the *NewSourceIdRunningNumber* and the data of each job is possible because each job gets on execution an unique solutionId (see Section 3.3).

Finally, and taking into account that the correction (offset) computation is deterministic and can be computed unequivocally, the consolidation can again be executed efficiently in parallel following the same job equalisation as the *MatchResolver*. With the completion of all *NewSourceIdConsolidator* jobs, the IDU-XM can be considered finished and the results are ready to be distributed to the rest of the DPCs.

A full report, including both scientific results and task performance reports has been included in Chapter 6.

4.4 Bias Determination

The Bias (IDU-BIAS) task is responsible of the characterisation of the Pre-Scan fluctuations in SM and AF CCDs to determine the gross bias offset signal introduced on all samples by the PEM [Hambly and Fabricius, 2010]. The Bias processing is completely separable from the rest of the signals added to the observed samples because the Bias is measured on a dedicated set of Pre-Scan measurements (ASD2 packets).

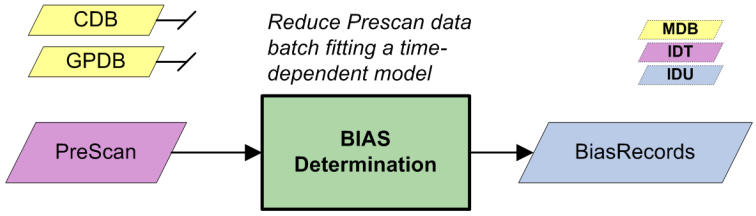


FIGURE 4.18: Schematic data flow for Bias treatment, showing the main inputs and outputs

This task specifically recomputes the predictions of the time dependent electronic bias in each detector and needs to run before any other task requiring to de–bias the data [CU5 DU10, 2010]. This task re-analyses the Pre-Scan telemetry packets in order to take more accurately into account any anomalies present in the data, and re–analyses the read noise properties of the PEMs. This objective is possible since IDU processes the Pre-Scan over larger datasets reducing the possible non-linearities that systems like IDT may introduce by its near–real time and time–chunked processing approach. In principle, it is assumed that over short time scales (of the order of seconds) there is no significant drift in the electronic Bias. The only recommendation is to run the task over overlapping batches such that the model fits are adequately constrained at the end points.

Figure 4.18 represents the schematic processing diagram for the Bias task for a given data time interval. The core algorithm uses a configurable functional fit (constant, single piece polynomial or spline of arbitrary order) over the Pre-Scan samples to determine the median bias value and scatter without mitigating the PEM-NU anomaly.

These predictions are mainly used in the Astrophysical Background processing, in the LSF/PSF response calibration and in the determination of the Image Parameters and in any other processes requiring the removal of the Bias offset. The recomputing of the offsets is carried out to improve the rough IDT estimations and to make use of the most recent calibrations of the PEM-NU calibration from other DPAC systems (FL mainly).

Although this task is included in the standard iterative data reduction cycle, it is envisaged that it will only be run a few times over each DRC

data segment since gross Bias signal related calibrations should be quite stable.

Regarding the low-level corrections to the gross Bias signal, mainly for AF2-9 due to the PEM-NU as already discussed in Section 2.4, the current baseline is simply to recompute the offset corrections given the most recent PEM-NU calibration (in principle from FL). In this manner, the correction is basically restricted to a more detailed bias determination issue in the consumer tasks. The application of this additional correction for the Bias is completely configurable, being necessary the full reconstruction of the read out sequence for each observation if the more accurate mitigation model is selected. The full reconstruction is always feasible thanks to the ASD data, mainly the object log from ASD7 packets (see Section 2.5). Finally, it is worth mentioning that the full mitigation of the PEM-NU has a significant impact on processing performance and although it is disabled by default in IDT, it is envisaged to be enabled in all IDU tasks requiring Bias treatment.

The implementation of the core algorithm for the IDU-BIAS task is the responsibility of the Institute for Astronomy - Royal Observatory of Edinburgh (IfA-ROE) team while its integration in IDU is done by CU3-UB.

4.5 Astrophysical Background Determination

The aim of the Astrophysical Background (IDU-APB) task is to produce functional fits to the astrophysical background so that it can be derived for any SM and AF window sample [CU5 DU10, 2010]. The effective background for Gaia observations can be modelled as the combination of the following components:

- Regularly spaced Charge Injections (CIs). See Section 2.1.3.
- Charge Release (CR) from these injections. See Section 2.3.
- Astrophysical Background coming from the two FoV.

The Astrophysical Background and the Charge Release (CR) are a tiny fraction of the charge level of an injection, and therefore the injection levels can be characterised separately. The other two components are not so easily separable; CR dominates at very short distances to previous charge injections but is almost negligible at long distances. In any case, and for simplicity, they are also processed separately. This breaks the background determination problem into three main processes: CI characterisation, CR characterisation and Astrophysical Background determination.

Initially, the CI and CR characterisation were also considered IDU tasks but during the latest test campaigns before launch it was decided to drop both tasks from IDU and integrate these processes within FL [Hauser et al., 2014]. The CI and CR calibration processes are considered stable enough so its reprocessing in each DRC is not necessary. However, we have included a brief description of these calibration for the sake of completeness.

The CI characterisation is the first step in the overall background calibration process. Because of the CTI, CIs introduce a charge release signal (or CR profile) that must be modelled as part of the background signal. The level of CR depends on the precise injection levels, so it is mandatory to determine accurately any variations of the injection levels in time and in AC position before attempting to model the other combined background signal components (see Cross and Hambly [2010] for the report on the AL and AC stability study for the CIs). The CI is nominally a series of four injections over the full height of the CCD, the first three of which aim to fill up all the traps (Figure 4.19). Taking into account that the injection level is many orders of magnitude greater than the astrophysical background and that the last sample in the injection is barely affected by CTI effects, the injection level can be determined by measuring the charge level in this fourth sample. The CI calibration processes the VOs – adequately configured to contain the four CI lines and covering the full AC extent – and fits two 1D model functions for each CCD. One to fit the AC variation and one to fit the time/AL dependent variation (although this last is envisaged to be very little). The results are stored in the so-called *CiAcProfile* library.

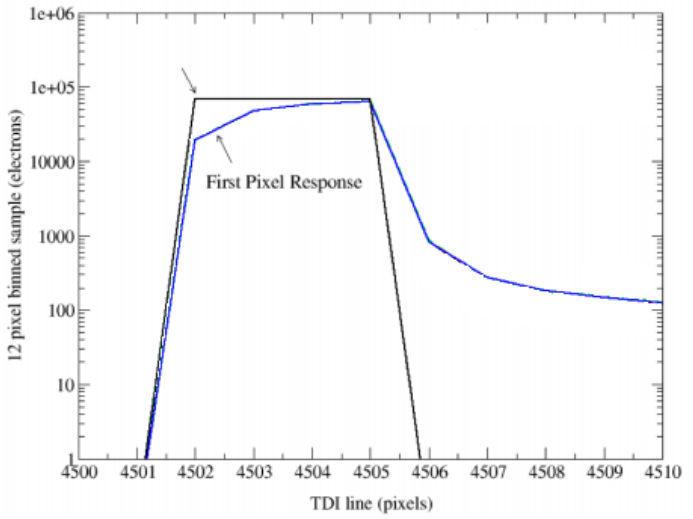


FIGURE 4.19: Charge Injection and Release simulation with zero (black) and mean (blue) trapping levels (Credit: Gaia/DPAC/CU5/IfA-ROE)

After the *CiAcProfile* library has been determined, the CR is calibrated. This second calibration process also uses the VOs and is based in the same fitting model, using two 1D model functions for each direction. Initially the CR curve model was modelled as a sum-of-exponentials charge release curve model, but this model was replaced with an empirical release signature [Hambly and Davidson, 2012]. This new model is more stable and provides a better adjustment to the real data. The successful run of this calibration process generates the *CrBackground* library.

Once the CI and CR characterisation is finished, the determination of the Astrophysical Background is done to achieve the full characterisation of the combined background. As for the other calibration processes, the model is based on a two 1D functions fitting. Before launch, the Astrophysical Background fitting was computed separately only for SM CCDs, whereas the AF were computed as the combination of the two SM solutions. This implementation was based on the assumption that the Astrophysical Background varies in both time and spatial position but does not vary for all detections of the same object in the CCDs along the row. However, due to the high level *stray light* (see Section 1.4), this approach was changed and currently the Astrophysical Background fitting is performed for each

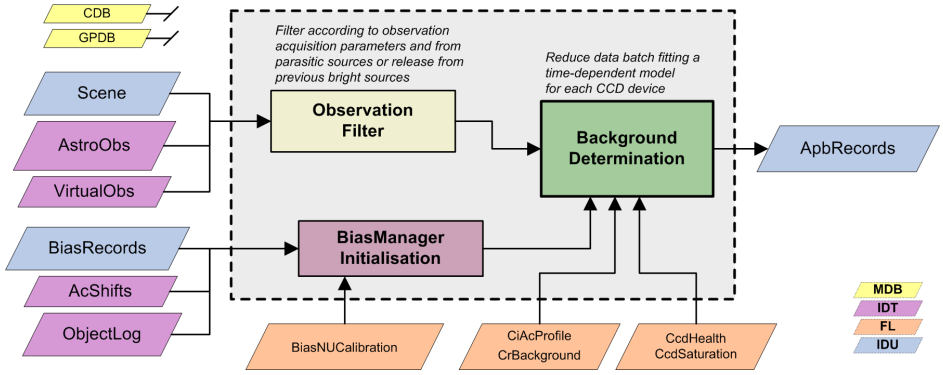


FIGURE 4.20: Schematic data flow for SM and AF Astrophysical Background treatment, showing the main inputs and outputs.

CCD separately. With the new approach and algorithm improvements, the fluctuations introduced by the *stray light* are successfully accounted for [Brown and Jordan, 2014].

The fitting of the Astrophysical Background is not only done over the VOs but also from a subset of normal observations. These observations are included to improve the stability of the solution – more samples to the fitting algorithm. In practice, only the samples on the edges of faint observations windows are used, where the observed source signal is supposed to be negligible. It is possible that the selected samples, or even the VOs, are contaminated by nearby bright sources but these can be easily identified using the transit prediction provided by the IDU-SCN task.

In practice, the IDU-APB task can be divided in 3 main blocks as shown in Figure 4.20. A first one in charge of the selection of the best calibration measurements from both the VOs and observations. A second block in charge of the computation of the Bias and PEM-NU corrections for each selected sample. And the last one in charge of the actual fitting after performing the corresponding Bias, the CI and CR treatment.

Note that for SM CCDs, since no CI is performed the CTI effects are not mitigated. However, due to the permanent CCD gate the damage on the acquired image is smaller.

Finally, we want to emphasize the benefits obtained by running the IDU-APB compared to those obtained in the analogous tasks in IDT:

- more accurate treatment of the bias; new gross offset derived in IDU-BIAS and the capability of a full PEM-NU mitigation configuration.
- more accurate treatment of the CI and CR components in both directions introduced by the improved calibrations coming from FL.
- more suitable calibration data – samples available for the background determination – mainly because IDU is not affected by the telemetry downlink priority.
- prior clean up of the calibration data; filtering any sample contaminated by sources as predicted from the IDU-SCN.
- better constraints at the beginning/end of the time dependent solutions accomplished by processing overlapping time batches.

The implementation of the core algorithm for the IDU-APB task is also the responsibility of the IfA-ROE team while CU3-UB is in charge of its integration in IDU.

4.6 LSF/PSF Calibration

The LSF/PSF calibration task in IDU is in charge of determining the response of the SM and AF instruments in the form of an LSF/PSF library for the 1D windows and 2D windows respectively. The LSF/PSF calibration process must use the latest calibrations to process the window samples and also the latest astrometric and cross–match solution to disentangle the colour/chromatic dependency between observations of different sources and to meet the formal requirements on the location estimation performance [de Bruijne, 2009].

The very first strategy, known as *forward modelling*, was based in the application of a Charge Distortion Model (CDM) over an LSF/PSF library

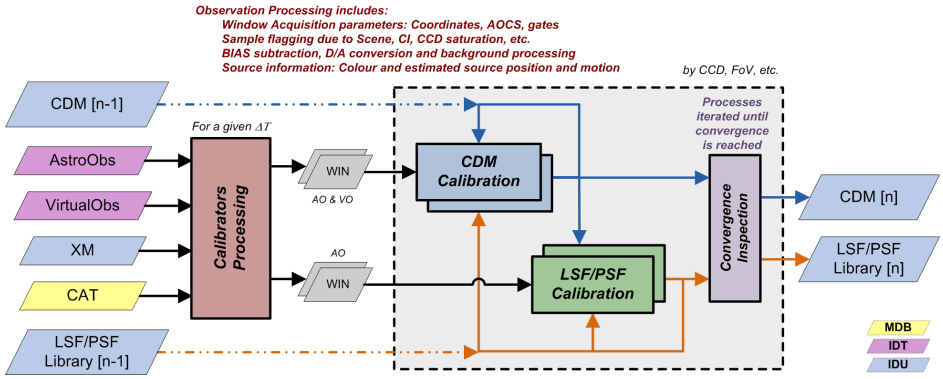


FIGURE 4.21: Schematic data flow for the iterative CDM and LSF/PSF calibration required in the *forward modelling* solution

for undamaged image profiles only accounting for optical projection distortion. The CDM must therefore describe the distortion of the charge image resulting from radiation-induced CTI damage to the CCD [Lindegren, 2008b]. In this sense the *forward modelling* is trying to decouple the optical response of the Gaia instrument from all other distortions coming from the CCD detectors. Importantly, in the *forward modelling* no attempt is made to correct the images for the effects of radiation damage and then fit a undamaged LSF/PSF model but instead the CDM model is applied to undamaged image profiles to arrive at a prediction of the observed image which is then compared to the real Gaia observations.

In the *forward modelling*, the CDM calibration is closely coupled to the LSF/PSF calibration, and this is only possible if both calibration processes are performed within the same iterative process as shown in Figure 4.21.

Several CDMs have been developed: CDM-01 [Lindegren, 2008b], CDM-02 [Short, 2009], CDM-03 [Short, 2011] and CDM-02/03 [Weiler, 2013]. Each of them moves towards a more complete and realistic physical model, as for example the inclusion of the gate handling in CDM-03.

The CTI mitigation tests done with all available models before launch raised fundamental issues blocking the proper development of the overall LSF/PSF calibration [Hambly et al., 2011]. The main issues identified were:

- The complexity of the CDM calibration; including the track of complex illumination histories and thus requiring a lot of computational resources.
- The processing of 1D windows; where large systematic biases were obtained in the along scan image location estimation.

Several options were then discussed, describing simplified CDM versions for 1D windows and light CDM calibration processes (see Fabricius and Torra [2011] and Davidson et al. [2011]). However, due to the lack of available time to develop an improved and lighter CDM version on time before launch, it was decided to proceed with the *Empirical LSF/PSF (ELSF)* model [Davidson and Hambly, 2013]. The ELSF model is based on the direct parametrisation of the final damaged image shape in a unique library of LSF/PSF profiles, thus accounting for all the distortion effects. A feasibility study of this approach is described in Brown and Crowley [2012].

The ELSF was initially introduced only to cover the processing needs and to meet the resource limitations of IDT/FL whereas the *forward modelling* was maintained as the IDU baseline. IDU schedule was not so tight as it was entering operations much later than IDT/FL and thus more time was available for further CDM developments. Additionally, the computational resources devoted to IDU allow more ambitious and complex processing solutions.

Unfortunately, the research of new CDMs was discontinued and no major progress has been made since the adoption of the ELSF model. Instead, an extended version of the ELSF was suggested for use in IDU. This model, by the name of eXtended Empirical LSF/PSF (XELSF) is still only a concept for the inclusion of some parametrisation of the illumination history in the formal ELSF.

The implementation of the ELSF based calibrations is much more simple than the *forward modelling* removing the need of two separated processes for the decoupling of the CDM from the undamaged LSF/PSF profile.

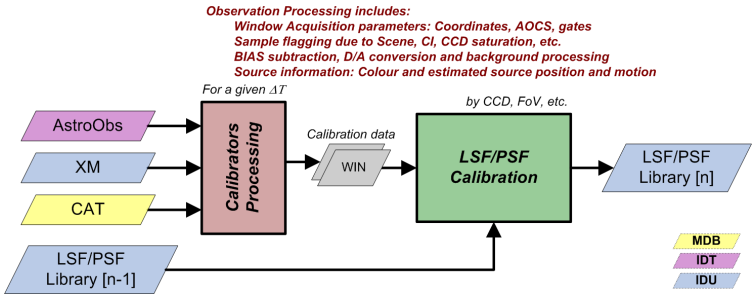


FIGURE 4.22: Schematic data flow for ELSF calibration, showing the main inputs and outputs.

A schematic implementation diagram of the ELSF task is shown in Figure 4.22.

Both LSF/PSF calibration strategies (Figures 4.21 and 4.22) basically split the full mission data set in time chunks. They start from a similar preparatory step in charge of the selection and preparation of the calibration data set. This step involves the combination of various data and the calculation (prediction) of the source location for the transits involved, etc. merely to provide a flexible framework where iterations can be carried out at the lowest cost. After this preprocessing of the observations, the core calibration starts which sooner or later will fit the modelled windows with the observed ones to construct the LSF/PSF library. Another point in common is the assumption that the LSF/PSF, regardless of the model, is expected to change slowly over the course of the mission, with the possibility of a more rapid change due to increased radiation damage after solar flares, for example. Consequently, there should be a running solution entering the calibration process to keep the calibration up to date and to guarantee the continuity of produced library solutions.

As for the IDU-BIAS and IDU-APB, the implementation of the core algorithm is responsibility of the IfA-ROE team. At the time of writing this thesis, the complete integration of the LSF/PSF calibration task in IDU is still ongoing but latest activities indicate that most probably a first working version based in the ELSF could be integrated on time for its first execution in the next DRC, at the beginning of 2016.

Following sections describe in more detail the LSF/PSF concept and implementation and how the astrometric solution is integrated in the LSF/PSF calibration.

4.6.1 LSF/PSF Model

The Point Spread Function (PSF) describes the response of an imaging system to a point source. It represents an essential tool for describing the optical response of the Gaia instrument, mainly including the response functions of the mirror, the optical projection and the detectors (CCDs). The Line Spread Function (LSF) is in general derived by integrating the PSF perpendicular to the direction of interest – for Gaia, in the AC direction which is required for the processing of the 1D windows.

The optical distortion across the Gaia instrument makes the shape of the PSF vary with the CCD, the source brightness and colour, the gate and the AC position. Additionally, the synchronization of the TDI with respect the scan rate, the AC smearing and the CTI may enhance the variations in the AL and AC directions.

The modelling of the PSF is quite complicated as it must consider a wide range of parameters presenting both linear and non-linear responses. This modelling is in general performed by means of:

1. Analytical basis functions decomposition.
2. Spline fitting over detailed (oversampled) 1D/2D numerical maps.

From the very beginning, the baseline for the PSF modelling in Gaia was the adoption of the first method. This method is based on the parametrisation of the PSF as a linear combination of analytical basis functions. The basis functions are such that the linear combination can represent any observed profile with sufficient accuracy and precision. This requires a certain minimum number of basis functions to be defined, spanning the relevant subspace of possible profiles. Once the basis functions are defined,

the function coefficients or model parameters are determined by a fitting process over the observed data.

In practice, these basis functions are derived by Principal Component Analysis (PCA) over a set of minimally–damaged observations or over a large ensemble of randomly generated, but physically plausible clean PSFs. This procedure offers the possibility to derive models with a minimum number of free parameters for given accuracy. It also implies that the first basis should ideally be close to the expected (mean) PSF of the real instrument, which may be used in the absence of any other information. With this formulation the number of parameters and thus the accuracy of the model can be adapted to the amount and quality of the data available. With more and better data available, the number of fitted parameters can be increased successively.

The full study on the LSF/PSF model formulation is described in several technical notes: Lindegren [2003], Lindegren [2009a], Lindegren [2009b], Lindegren [2010a] and Lindegren [2010b], which additionally describes a general 2D PSF model, considering the integration of the AC smearing. This LSF/PSF model defines a set of basis functions such that:

- their linear combination can represent any observed LSF to sufficient accuracy.
- basis functions of increasing order only represent finer details of the LSF, so that a truncated expansion provides a useful approximation of the full expansion.
- the linear combination is normalised independently of the number of functions combined, thus providing unbiased flux estimations.
- they allow the accurate interpolation and even a reasonably safe extrapolation beyond the original fitted data.
- they clearly distinguish between the model origin and the geometric model centroid, further described in Section 4.6.2.

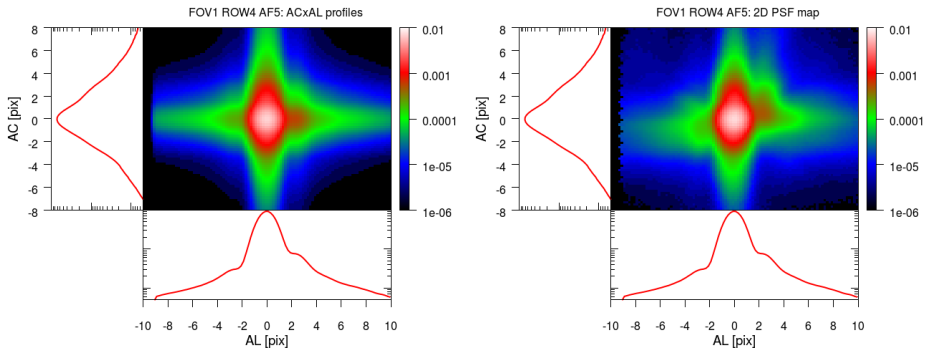


FIGURE 4.23: Example of the limitation of the PSF modelling using the outer product of the ALxAC LSFs compared to the PSF obtained using a 2D numerical mapping, left and right respectively (Credit: IfA-ROE)

- they minimize the number of free parameters as much as possible.
- they provide numerical stability of the fitting process.

The decomposition in basis functions is in general more flexible and versatile than the oversampled numerical maps which have a large number of free parameters coming from the need of increasing the oversampling factor for reducing the interpolation errors.

A practical and rather simple implementation of the general PSF model is provided in [Fabricius, 2011]. This implementation is based on the approximation of the 2D PSF as the product of the AL LSF and an analytic smeared version of the ideal AC LSF. However, recent analysis of the 2D PSF fitting performance has revealed that this approach may not be enough since it is forcing a symmetrical profile not representative of the actual data as shown in Figure 4.23.

For the 2D PSF case, the mapping solution could be the best approach and new developments led by the IfA-ROE group have already started. With this technique, the mean observed PSF for low AC rates could be represented directly with a 2D spline fit whereas the AC smearing could be modelled applying a rectangular convolution over the reconstructed image.

As commented at the beginning of this chapter, the complexity of a direct CTI modelling in the form of a CDM has led to the adoption of the ELSF

model. This model basically accounts for the image damage caused by the CTI by adding additional parameters in the computation of the coefficients of the LSF model. These parameters are the time since last injection and the illumination history prior to the observation. This modelling is accomplished in practice by modelling the coefficients of the different basis functions with a multidimensional spline [Davidson and Hambly, 2013]. As for the 2D PSF, it is constructed as the cross product of the AL LSF with the AC LSF, this last one directly accounting for the AC smearing.

At the time of writing this thesis, the ELSF calibration is performed in three steps for the provision of:

Mean LSF

This first step is in charge of the computation of the *Mean LSF* from a set of minimally-damaged observations, i.e. those close to charge injections. This calibration resolves the most basic parameter dependencies; CCD, FoV, gate and more. These parameters are discrete and they can therefore be treated as completely separate calibration dimensions. Furthermore, this calibration does not need to be performed often; perhaps every few months or as a result of a sudden LSF change [Davidson and Hambly, 2013].

Optical Corrections

This second process takes the *Mean LSF* and a selection of suitable minimally-damaged observations (with their corresponding image parameters and backgrounds) and minimises the residuals between the model and data. The results are basically the updated optical correction coefficients in the form of multidimensional splines.

Electronic Corrections

This calibration is done in almost exactly the same way as the *Optical Corrections* described above just modifying the inputs. In addition to the *Mean LSF* and the *Optical Corrections* this step requires observations covering the full range of the electronic correction parameter values along with their image parameter estimates for all CCDs.

The general response to CTI as a function of source magnitude and distance to last injection is expected to be common for all devices and therefore this calibration is done globally for all CCD observations.

It is anticipated that an additional step will be added for accounting the actual difference in response caused by the variation in radiation damage across the focal plane. This difference would be modelled as a CCD-dependent CTI scaling factor applied to the general *Electronic Corrections* computed in the last step. This factor would be calibrated in an analogous way to both corrections.

With this approach we ultimately separate the detector effects from the optical effects; so we first fit the *Optical Corrections* to obtain an undamaged LSF from the *Mean LSF* and then we apply the *Electronic Corrections* to obtain a damaged LSF from the final undamaged LSF.

The calibrations of both corrections are done in almost exactly the same way, using Householder [Householder, 1958] least squares to solve for the updated correction coefficients over a calibration dataset. The main difference lies in the characteristics of the datasets used in the calibration.

The main parameters used to determined the along–scan LSF are:

- For the *Optical Corrections*:
  - The source colour; represented by the wave number and causing charge diffusion.
  - AC position; accounting for the distortion of the image projection and the variation on the response of each AC column.
- For *Electronic Corrections* :
  - Distance to the last CI to incorporate parallel CTI.
  - Magnitude of the observed source.

Analogously, the main parameters defining the across–scan LSF remain as:

- For the optical corrections:
  - The source colour.
  - AC motion during integration along the CCD; accounting for the AC smearing.
- For electronic corrections:
  - AC position; which is known to be a good parametrisation of the serial CTI.
  - Source magnitude.

Early in the mission the *Optical Corrections* are likely to be much greater than the *Electronic Corrections*. Later, when damage has accumulated, the *Electronic Corrections* will become more important and the *Optical Corrections* should be relatively stable and well-characterised.

It must be noted, that since the ELSF is magnitude-dependent it can no longer be freely scaled and combined as the general LSF model. For the purposes of Image Parameters Determination (IPD), it is important to obtain good initial estimates of the source flux to ensure that a reasonable LSF model is returned.

The most challenging part of the LSF/PSF calibration process is the mitigation of the CTI effects according to the illumination history of the CCD. In general, determining the illumination is quite easy and in fact is basically provided by the IDU-SCN task. However, finding a feasible formulation based on the scene in function of the distance and flux of the sources contributing in the illumination history is one of the more challenging aspects for the instrument response modelling.

Figure 4.24 represents one practical example of the illumination history that needs to be solved when processing an individual observation. In the figure it can be easily identified the last CI and also the several sources observed in between.

Additionally, the scene information is again used for filtering the observations entering each calibration step as it is done for the IDU-APB task.

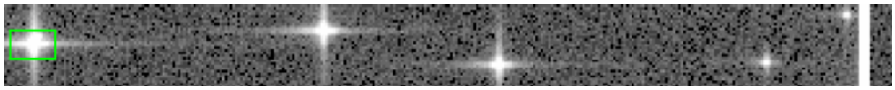


FIGURE 4.24: Example of the illumination history for a given observation (green boxed) starting from the latest CI (Credit: IfA-ROE)

4.6.2 Astrometric Solution Integration

As commented above, the astrometric and cross–match solutions must be included in the LSF/PSF calibration process to be able to untangle the chromatic dependency between observation of different sources.

The chromatic shift, i.e. between blue and red stars, can be handled by the proper definition of the LSF/PSF origin with respect to the LSF/PSF geometric centroid. The LSF/PSF origin is conceptually the reference point from which the observation time and AC location are determined after the fitting whereas the centroid can be understood as an average (e.g. arithmetic mean) of the coordinates of all the points of the shape. Figure 4.25 illustrates the difference between these two parameters and how the chromatic shift can be easily accounted by a proper definition of the LSF/PSF origin. This origin should be chosen to be equivalent to the geometric centroid of an achromatic instrument response which in practice is achieved using the astrometric solution from AGIS.

Following the overall philosophy of the IDU-AGIS iterative loop introduced in Section 3.2.3, AGIS will be in charge of fixing the location of the LSF-/PSF origin. Letting AGIS make the decision of where to put the origin relieves the calibration from taking a rather arbitrary decision, but first of all assures that the origin is achromatic. The chromatic shift between a blue and a red star should be fully compensated with this definition of the LSF/PSF origin during the fitting, and there should be no colour terms left for later AGIS runs.

In practical terms, the LSF/PSF calibration task is in charge of computing the expected location of the source image centroid within each calibration window in the form of the expected observation time and AC position at

the expected fiducial line (gate configuration). This expected location is computed from the attitude, the *geometric calibration* (only accounting for purely geometric terms GEO-CAL) and the corresponding astrometric solution in the Gaia catalogue retrieved by means of the IDU-XM results. These AL and AC coordinates could also be obtained from the corresponding *Scene* record for that observation, interpolating the field angles and from them the centroids at the fiducial lines of each CCD window.

The computation of these predicted coordinates can not be done directly since the *geometric calibration* can not be inverted. Instead an iterative process, depicted in Figure 4.26, must be followed where an initial guess is progressively updated until the desired convergence threshold is reached. During this process, the absorption of the chromatic shifts in the LSF-/PSF calibration is enforced by using only the purely geometric term of the *geometric calibration* done by AGIS, the GEO-CAL.

Following this approach, the errors in the LSF/PSF fitting will come from the errors in the source parameters, attitude, and *geometric calibration*. The errors will also be present in the corresponding derived Image Parameters which will then be used to improve the AGIS solution in the next run. In the early stages of the mission as well as for IDT, it is much preferable

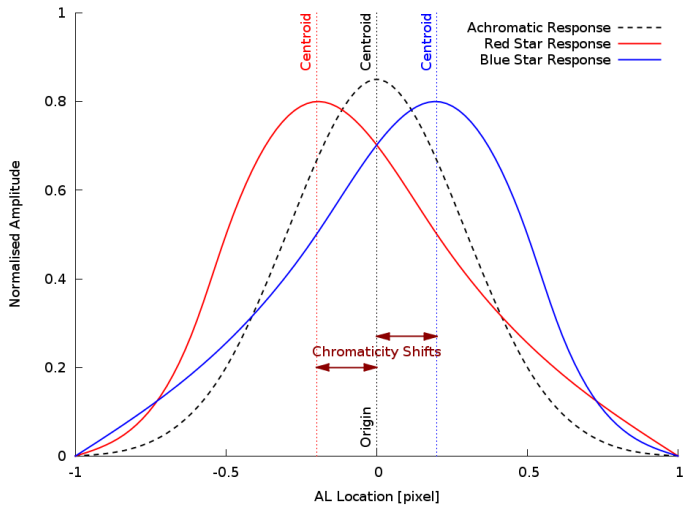


FIGURE 4.25: Definition of LSF origin with respect the image geometric centroid adopted applied for chromatic shift correction

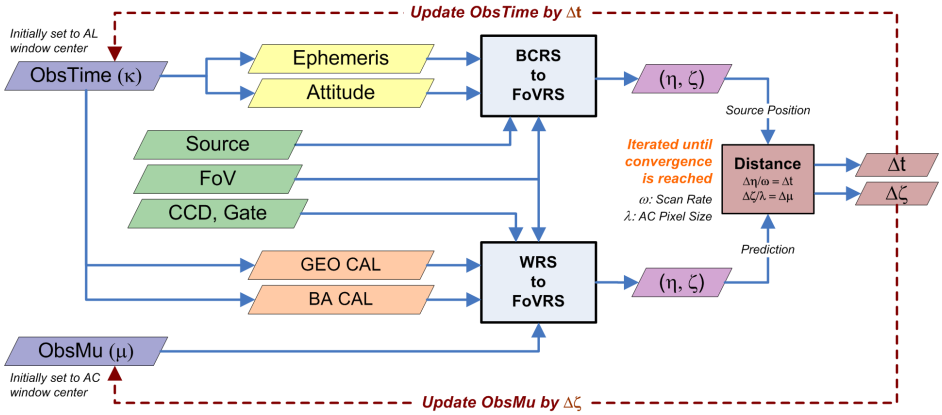


FIGURE 4.26: Diagram of the iterative processing required for the estimation of the expected source location within each calibration window

to adopt an LSF/PSF origin equal to the centroid, as being less susceptible, e.g., to attitude errors. However, it is clear that IDU should use the approach described above since it will determine the LSF/PSF shape more accurately – once the source and attitude parameters are accurate enough [Lindegren, 2010b].

4.6.3 Input Calibration Data

The LSF/PSF calibration will be run on each DRC and whenever a new astrometric solution is made available. This task, as the rest of IDU tasks, will gradually receive and accumulate a huge amount of observational data to be processed. However, not every transit is considered useful for LSF-/PSF calibration, for instance due to acquisition problems or if its background has not been well-determined.

It is clear, that for the purpose of this calibration, processing the expected $\sim 10^{11}$ transits observed during the 5 years mission is not actually required. In fact, for most of the IDU tasks only the well-behaved observations will enter in the processing. Understanding as well-behaved the minimally-damaged observations, windows free from the CI signature, without contaminating flux from other sources, with the nominal window geometry,

with its bias and background well-determined, etc. However, in the case of the LSF/PSF calibration a representation of almost all possible observations is required to get a complete and unbiased calibration – preferably observations having also a good astrometry from AGIS.

The huge number of observations and the variety of parameters and effects to be calibrated requires a full pre-processing and selection of the data before running the calibrations. This selection must take into account the following aspects:

- The calibration dataset must cover the complete magnitude range and window sampling classes.
- Red and Blue sources do not have the same sky distribution and if the sources are taken randomly, the attitude AC smearing and the chromatic dependency will be mixed.
- Any information on the close source environment, to avoid the use of binaries which may lead to inaccuracies in the calibrations.
- All major events of the spacecraft.

For the new data arriving to IDU, the selection will be based on the basic information available from:

- The observation acquisition flags.
- The scene, cross–match and catalogue analysis.
- The quality indicators of the calibrations involved in the raw samples processing; including Bias, CI, CR and Astrophysical Background.

4.7 Image Parameters Determination

The main purpose of the Image Parameters Determination (IPD) is to use the latest calibrations of the Bias, the CI/CR, the Astrophysical Background, LSF/PSF, etc. to compute improved Image Parameters which

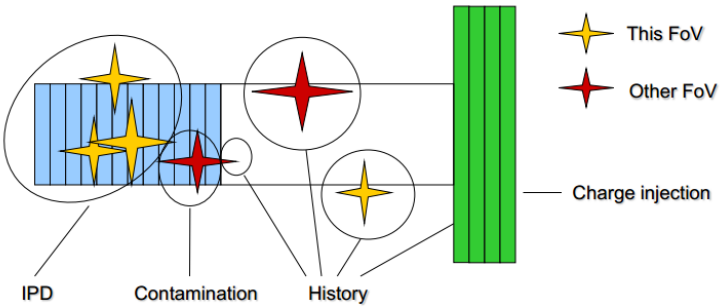


FIGURE 4.27: Schematic example of the processing classification applied to the sources (yellow and red) according to their position with respect to the observed window (blue), the CI (green) and their FoV CI (Credit: Gaia/DPAC/IDT/IDU)

were initially determined by IDT or previous DRC runs of IDU. The precision and accuracy of the AL location of a source within each individual window observation is of the foremost importance for the astrometric reduction system, specifically for the loop between IDU and AGIS. The level of error in this measurement affects the astrometric performance of the mission and, as anticipated, this level of performance can only be achieved after iterative processing.

To the Image Parameters Determination (IDU-IPD), all observations are assumed to be of fixed point sources. Non-single sources and moving sources are processed by CU4. The IPD may additionally consider known sources from both FoVs (see Figure 4.27) for:

- Working out the illumination history prior to the window, considering all the sources in between the last CI and the window.
- Treatment of contaminated pixels due to parasitic sources from the other FoV.

The contribution of the sources from the given FoV within the window are deferred to more sophisticated Image Parameters processes from other DPAC processes as e.g. the SEA done by CU5. The complete list of sources is provided by the IDU-SCN task.

The Gaia observations present a wide range of effects and not all of them can be easily treated. The main effects which must be considered when computing the Image Parameters are:

- **CTI effects:** The treatment of CTI effects is highly non–trivial, as we are dealing with not fully understood, non–linear effects, interacting with other instrument deficiencies, and without the full CCD data. With the ELSF adoption, these effects are accounted directly in the LSF/PSF calibration.
- **CCD Sensitivity:** All columns in a window are assumed to have the same sensitivity, and no attempt is made to include photometric calibration parameters which will be handled by other DPAC systems.
- **Saturation and non-linearity:** These effects must be taken into account when processing 2D windows but they can be neglected when processing 1D windows in AF where the saturation level is far enough from the estimated signal levels.
- **Window geometry:** Windows may suffer all kinds of complexities, as a result of CCD gating, window truncation, AOCS updates (window AC shifts), CIs, etc. To avoid unnecessary complex processing of damaged observation, the IPD will only process the set of samples obtained with the same gate, having the same start column, and the same length (binning) as described in Fabricius et al. [2012].
- **CCD Cosmetics:** Column defects may be taken into account when processing 2D windows, by simply ignoring the affected column, while this is not possible for 1D windows due to the AC binning.

The error on the estimated AL location can be described in general as a function of the observation magnitude and the CCD strip. However, there are major systematic errors introduced by the radiation damage and this residual error can be an order of magnitude above the mission requirements if no mitigation is applied.

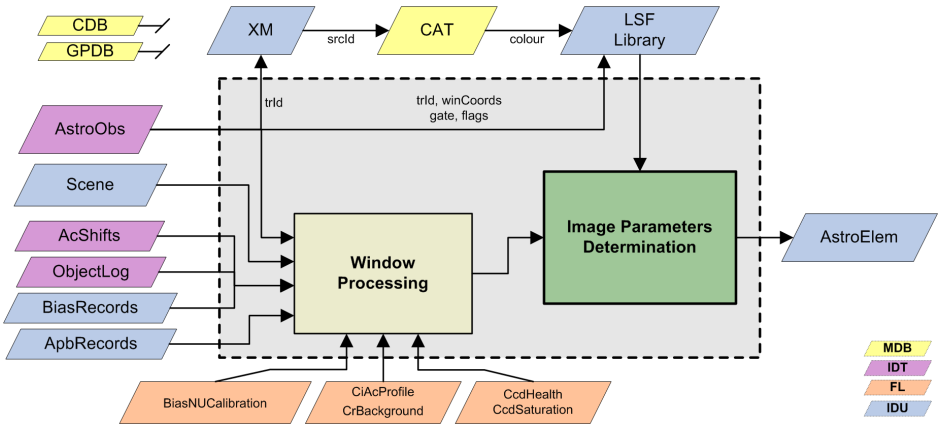


FIGURE 4.28: Schematic data flow for Image Parameters Determination (IPD), showing the main inputs and outputs.

Figure 4.28 shows the principal components of the data processing flow for the IDU-IPD task. The main differences of this data processing flow from the one used in IDT are:

- The analysis of the scene, used for knowing the detailed illumination history of each CCD, and in case we wish to treat parasitic signals in the modelling of the observation.
- The retrieval of the photometric information from the Gaia catalogue instead of the direct measurement obtained from the initial processing of the BP/RP data.

As for the previously described tasks (IDU-APB and LSF/PSF (IDU-LSF/PSF)) the raw observations are first preprocessed. This preprocessing consists in the determination of the effective samples derived from the raw observation and in the retrieval of all the suitable window-level calibrations, such us Bias, CI/CR, Astrophysical Background, etc. Over this data, a preliminary estimation of the Image Parameters (see Castañeda and Fabricius [2010]) is carried out to provide good initial location and flux values for the late fitting which may help when analysing the scene. Thereafter, we look up the relevant source information for the observation accessing the Cross-Match table. This information is required for the retrieval of

the more suitable LSF/PSF solution, dependent on the source colour and observation flux as described in Section 4.6.1.

The last step then consists in the fitting of the effective samples with the LSF/PSF, giving as results the updated parameters. This fitting is based on a Maximum-Likelihood Estimation (MLE), Lindegren [2008c] and [Castañeda and Fabricius, 2009]. Additionally to the Image Parameters, the fitting provides several indicators, as formal errors and Goodness of Fit (GoF), which are of great help for the monitoring and validation of the processing itself and for the selection and compilation of the calibration datasets for future iterations. See Section 4.8 for a more detailed description of the currently available validation and monitoring tools.

As described in previous sections, IDU includes several tasks, most of them essential for the successful reduction of the astrometric solution. However, all of them would be worthless without the execution of the IDU-IPD, where the actual integration of all the partial solutions – including non IDU products as FL and AGIS calibration – is done. Thanks to IDU-IPD all these intermediate products are consolidated for the generation of the most fundamental data required for the core astrometric reduction.

On the other hand, the processing and storage needs of this task represents one of the most challenging design issues of IDU system. The IDU-IPD not only has to process all the transits that will be accumulated during the mission, but it has to process them repeatedly every data reduction cycle. IDU will receive every cycle up to 20×10^9 new observations (200×10^9 windows in $\sim$ 4 Terabytes) which will be accumulated and reprocessed again and again. Chapter 5 covers all the technical difficulties and design issues found for the successful implementation of the IDU-IPD task.

The design and implementation of the IDU-IPD task is lead by the CU3-UB in collaboration with IfA-ROE.

4.8 Validation & Monitoring

The IDU scientific performance is assured by specific test campaigns following the approved DPAC standards described in leaders CU1 [2012] and Guerra and leaders CU leaders [2013]. These tests are carried out regularly by the DPCB team in close collaboration with all the IDU contributors. For these tests, detailed analysis over the obtained results are done – even including the execution of reduced iterations with other systems.

As already commented in previous sections, IDU processes a huge amount of data and produces similarly a huge amount of output results. The continuous and progressive check on the quality of these results is more than a desirable feature. However, the analysis of every calibration and parameter produced by IDU (as it is done for the test campaigns) is not affordable – it would have almost the same computational cost than the processing itself. For this reason, we aim for the design and implementation of a modular system able to assure the quality of the results up to a reasonable limit.

First of all, we have implemented in each IDU task several built–in consistency checks over the input and output data. These are really basic checks for:

- verifying the consistency of the configuration parameters including its tracking along the full processing pipeline.
- verifying the consistency of the input data, so corrupted data or inconsistent input data combinations do not enter the pipeline and are not propagated to subsequent tasks.
- accounting for the number of outputs with respect to the inputs, so data lost is detected and properly handled – in general forcing a task failure.

Additionally, all IDU tasks integrates the Intermediate Data Validation (IDV) framework [Valles et al., 2012]. This framework provides several tools for the generation of statistical plots of different kinds. IDV provides:

Bar Histograms

Histograms for the characterisation of the frequency of parameters with limited number of values.

For example IPD processing status, counts of observations per row, etc.

1D Histograms

Histograms for the computation of the frequency of non discrete parameters. This implementation supports the computation of several percentiles and the Robust Scatter Estimator (RSE).

2D Histograms

Histograms showing the distribution of values in a data set across the range of two parameters. They support static dimensions or abscissae dynamic allocation – in general for analysis of the evolution of a given parameter in function of a non restricted increasing parameter as the observation time. They can be normalised globally or locally for each abscissae bin. Percentiles and RSE as well as contours are supported.

Mainly used for the analysis of 2D dependencies or 2D density distributions of two given parameters – usually the abscissae parameter is the magnitude, or some kind of distance; to last CI, to a reference observation/source, etc.

Sky Maps

Plots generated from a histogram based in the HEALPix tessellation and implementing the Hammer-Aitoff Projection. It can represent the pixel count, pixel density or the pixel mean value for a given measured parameter.

Mainly used to obtain the sky distribution of some particular object (sources, observations, etc.) or to analyse the alpha and delta dependency of some parameter mean value, i.e. the Astrophysical Background, proper motion, etc.

Sky Region Maps

Tool for plotting sources and detections in small ICRS-based sky

regions.

Mainly used for the analysis of the IDU-XM and IDU-DC results.

Focal Plane Region Maps

For the representation of the observations according to their AL and AC plane coordinates.

Mainly used for the analysis of the IDU-SCN and IDU-DC results.

Round-Robin Database (RRD)

RRD are used to handle and plot time-series data like network bandwidth, temperatures, CPU load, etc. but also usable to handle Quaternion evolution, observation density, match distance evolution, etc.

Range Validators

Implement very basic range validation against the expected nominal parameters defined in the MDB ICD

TableStats

Collector of statistics for miscellaneous MDB table fields. Basically provides counters for the discrete values of predefined fields or for boolean flags/fields.

All **Histograms** and the **SkyMaps**, share a common framework allowing the split of the collected statistical data according to the FoV, CCD row, CCD strip, window class, source type, etc. This functionality is quite useful for restricting the origin of any features visible in the general plots – in that sense the user can see if some plot peculiarity or feature are present only in one of the FoV, rows or for a given source type.

It is worth mentioning that the **Sky Region Plot** can be generated directly from the code but also we have implemented a graphical tool for the interactive generation of this kind of plots. This tool is called **Sky-Explorer**. This tool allows the loading and visualisation of all kinds of Cross-Match related data. It also implements the functionalities for sky navigation, zoom, distance measurement, animation of Gaia scans and

much more. The **SkyExplorer** has become an essential tool for the validation of the IDU-XM but also for the IDU-DC. Several examples of this tool outputs have been already included in Section 4.3 in this chapter and more can be found in Chapter 6.

All these tools have also been integrated in IDT and they are used for its monitoring on a daily basis. A handful of examples of the plots obtained using these statistics tools have been included in Chapter 6.

With all the listed IDV features, the monitoring of the IDU scientific results should be easy. The only thing pending is the definition of the best diagnostics for each specific task. Some examples could be:

- For all task in general:
  - Range validation against expected nominal parameters.
- For IDU-XM and IDU-DC:
  - Monitoring of the amount of new sources created compared with previous executions of the IDU-XM.
  - Monitoring of the time evolution of the Cross-Match AL/AC distance to the primary matched source per FoV.
  - Check the evolution of matches to a predefined set of reference sources, to check if the overall transits have been assigned differently now, as compared to the previous cycle.
  - Monitoring of the evolution of the number of spurious detection density for very bright sources.
  - etc.
- For IDU-IPD:
  - Monitoring on the goodness–of–fit obtained.
  - Comparison of the derived Image Parameters against the AGIS solution over a pre–selection of well–behaved sources.

- Cross—check of the residual from the previous statistic against the chromatic calibration residuals from AGIS.
- etc.

Most of these checks are still done manually by the operator but their progressive integration in the processing framework is envisaged so they can be performed automatically and summarising reports are provided to the user for inspection. At the time of writing this thesis, many efforts have been devoted on identifying the best diagnostics – the ones assuring the best control on the quality of the produced data – and some examples are provided in Chapter 7.

Finally, it is worth pointing out that the computational performance and the correct progress of the processing is also monitored. The detailed list of functionalities designed and implemented so far for monitoring the job performance and handling the jobs outcome is presented in Chapter 5

4.9 Conclusions

After describing in Chapter 3 the basis of the data reduction system, we have described in more detail some of the most important tasks involved in the astrometric reduction loop. Particular attention has been paid to the IDU-XM, the IDU-SCN and IDU-IPD where most of the work done is attributable to this thesis.

The LSF/PSF calibration has also been covered exhaustively. During years, it has been discussed and studied the possibility of producing a clean LSF/PSF library – free from the CTI effects – in conjunction with a Charge Distortion Model (CDM). In this scenario, the IPD would have been in charge of predicting the distorted image from both: the clean LSF-/PSF library and the CDM and then performing the Maximum-Likelihood Estimation (MLE) against the observed image. This direct modelling of the charge distortion would be more transparent, versatile and will be able

to cope more rigorously the illumination history. Unfortunately, this approach has been deferred since no suitable CDM has been achieved yet and because the estimated development cost of this full direct modelling have been considered not affordable at the current stage of the LSF/PSF calibration developments. Instead an empirical modelling approach has been followed, implementing an Empirical LSF/PSF (ELSF) library directly accounting for all possible image distortion factors.

We have also described the IDU-IPD task, reasoning how this task brings together all the partial solutions of the several processing and calibration systems coming from different CUs, consolidating the starting of the astrometric reduction iteration loops. This consolidation is essential for the improvement of the astrometric solution produced by AGIS.

It is worth pointing out the close cooperation with IfA-ROE team during this thesis. The four months stay with this team during 2010 and 2011 procured a solid basis for the design and implementation of most of the IDU tasks. This close cooperation continues and is fundamental for the progressive improvement of most of the IDU tasks.

It must be noted, that the current design of IDU tasks, and their implementation later described in Chapter 4, fulfil to big extent all the requirements for the Gaia data exploitation.

Finally, an overview of the several monitoring and validation tools implemented in the frame of this thesis has been included. The autonomous validation and monitoring of the outputs is still an ongoing task but this fact can not be considered a major or stopping issue for the execution of any of the developed IDU tasks.

5 IDU Operation and Implementation

The design and implementation of IDU has been one of the main goals of the work done within the frame of this thesis. This design has not been only driven by the scientific goals and requirements [Castañeda et al., 2011a] but also by the characteristics and restrictions of the execution environment and resources.

The execution of IDU is done at DPCB, in particular at the Marenostrum supercomputer hosted by Barcelona Supercomputing Center (BSC). This supercomputer offers a peak performance of 1.1 Petaflops and 100.8 Terabytes of main memory and is composed of more than three thousand computing nodes. Because of the machine design, the use of databases hosted in a dedicated hardware would be complicated and very costly and it is discouraged by BSC. This implies consequently that all the processing is based exclusively in files. The technical specifications of this supercomputer are given in Appendix A.1.1. More information about the DPCB responsibilities and resources has also been included in Appendix A.

Additionally, hardware upgrades of Marenostrum are envisaged during the mission – the latest one was done beginning of 2013. This circumstance, covered in more detail in Appendix A.1.1, has also been taken into account in the design of IDU to obtain a very modular and flexible implementation as we described later on in Section 5.2.

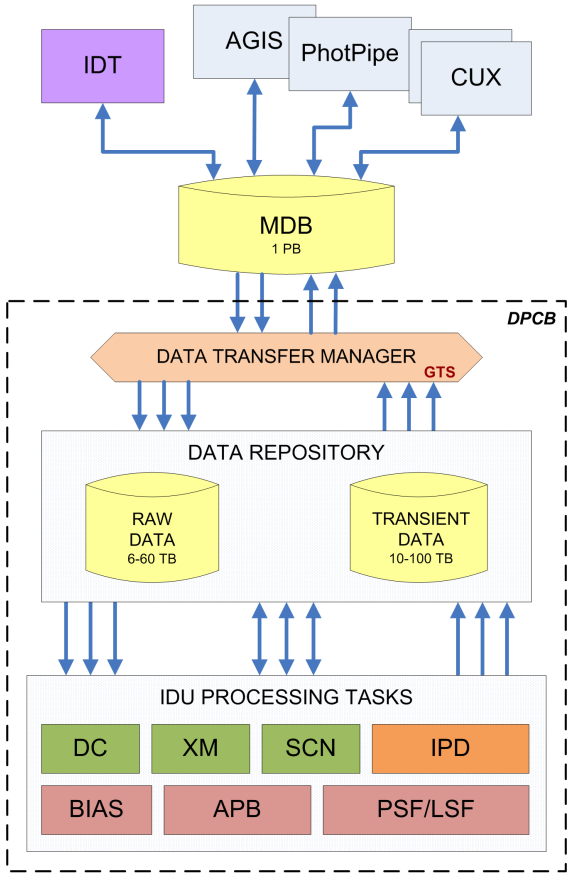


FIGURE 5.1: Schematic context of the IDU tasks within DPCB, including some of its main interfaces with other DPAC products

Figure 5.1 illustrates the context in which IDU tasks operate, including the main interfaces with DPCB resources and with the data products from other DPAC software systems.

IDU itself is composed of seven sub–processes, which we call tasks, that perform the operations on the Gaia data already described in Chapter 4.

In practice, IDU could be described as a *batch processing* system, where series of processes or jobs are executed on a computer without external communication or intervention. Jobs are set up with all the inputs predefined so they can be run to completion in isolation. This is in contrast to interactive or *stream processing* systems which expect the input from user or other processes to trigger the processing. IDT would be a clear example

of this *stream processing* where messages about the newly available data are received and trigger the processing.

Batch processing is very efficient in processing high volume data, where data is collected, entered to the system, processed and then results are produced in batches. Besides, using partitioning allows multiple jobs to run concurrently thus reducing the elapsed time required to process the full data volume. Special care must be taken in the partitioning of jobs, so they only process their assigned data set.

However, due to data dependencies between the tasks, the IDU task execution must be coordinated, and considerable data transfers and arrangement are required. There are data dependencies between some of these tasks, and the tasks must be executed in a particular order (see Figure 4.1).

For this reason the execution framework must permit the launching of the tasks, the management of I/O data of each task, and additionally provide monitoring and overall management of the task flow. Issues of efficiency are especially important for this work, such as providing efficient data access, and ensuring the efficient job partitioning.

In this chapter, we firstly cover the practicalities for a nominal IDU execution – data preparation, job definition and the execution plan according to the DRC and DS definitions. Next section, is focused on the details of our implementation of IDU, describing the developed interfaces and execution framework. This section also presents the guidelines adopted for the software development within DPAC, including the additional constraints due to the DPCB environment and the strategy we have followed for testing IDU system. Afterwards, we describe the main monitoring and profiling tools available within IDU, fundamental to get the best performance at DPCB. Finally, we summarise the main topics covered, highlighting the work directly attributable to this thesis.

5.1 IDU Operation

IDU only runs on DRC basis, which means it will be executed once or twice a year over the accumulated Data Segments (DSs). The first DS, covering the first ten months of Gaia routine operation, reaches already a total amount of 7 TeraBytes only for the raw observations. Forthcoming DSs will imply subsequent data volume increases of 3-6 TeraBytes.

Handling this volume of data is not an easy task and requires the adoption of very strict procedures covering the data preparation, the task definition and the detailed execution plan. The definition of the execution plan is critical and must take into account the following:

- The time slot assigned for the IDU execution within the overall DRC schedule, preventing any undesired delay which may affect the other downstream systems.
- The feasibility to get enough resources for the processing. DPCB resources are shared with other projects and the resources must be requested in advance.

Furthermore, any official execution of software must follow strict procedures so all inputs, software releases, algorithm configurations, etc. are perfectly accounted for. In other words, any execution must be completely deterministic and reproducible. To accomplish this premise, we have defined three different stages (outlined in Figure 5.2):

Release Stage

Well before the IDU tasks execution, its functionalities must be fixed according the latest developments carried out by the main contributors. This involves several CUs and software libraries (already introduced in Section 5.2) from teams located at different places which in general have their own software development schedules. Arranging all these schedules is not an easy task since most of these software

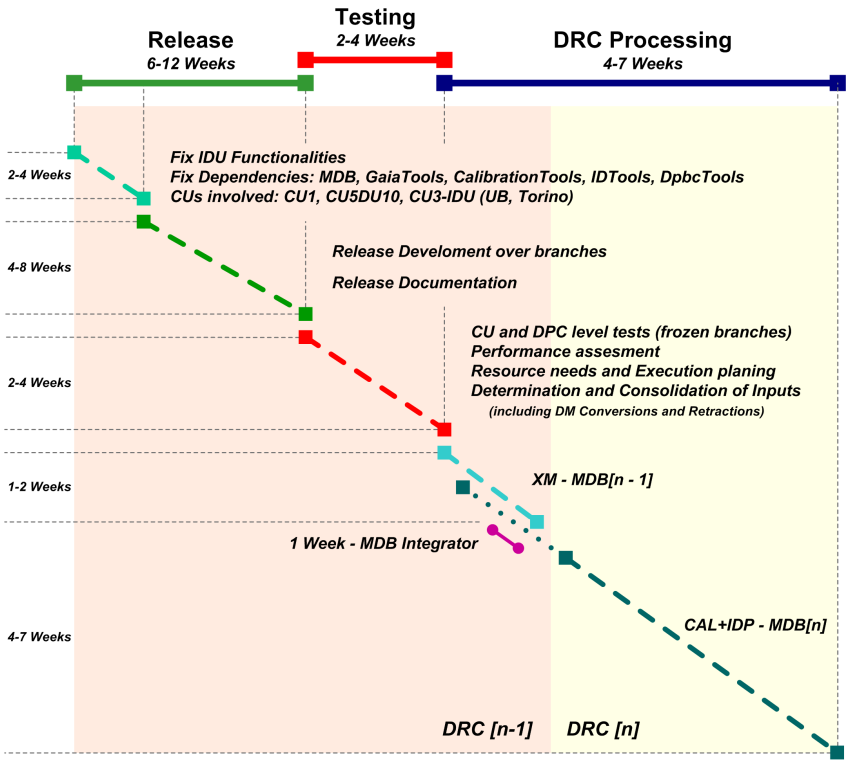


FIGURE 5.2: Schematic outline of the main steps (horizontal axis) that must be followed for the execution of IDU tasks, also accounting for the wall clock time requirements (vertical axis)

libraries must also fit in with other software releases not strictly tied with the DRCs as IDT.

Once the main functionalities have been established, they have to be implemented and integrated in IDU. During this stage, major implementation issues or lack of man power may compromise the original release functionalities but in these cases corrective actions will be taken so the final scientific impact is minimised. Next releases will eventually fix these issues as part of the nominal development activities.

This stage could extend over periods of one or two months depending on the new functionalities introduced or the major issues identified in previous executions.

Testing Stage

This stage starts when the IDU release is considered already stable. At this point, all software changes must be approved by the DPCB Configuration Control Board (CCB). During this stage, the release is deeply tested at CU level and also at DPC level (see Section 5.2.5). From the testing, essential information will be retrieved for the estimation of the resource requirements and for the elaboration of an initial operation plan. New releases may include substantial performance updates or new dependencies forcing changes on the already existing operation plans consolidated in previous executions.

During this stage, preliminary input data arrangements and data statistics are also carried out. From the results, fundamental information is retrieved for the estimation of the resources required for the processing.

Typically, this stage should last for one month allowing several executions and occasional small-scale iterations with external systems as AGIS. This stage finishes once the release of IDU is done.

DRC Processing Stage

This is the stage where the actual execution of the IDU tasks is performed. The time available for the processing will extend approximately up to two months and during this period the following steps will be followed:

1. Execution of IDU-DC and IDU-XM and distribution of the results to DPCE.
2. Reception of the new catalogue produced by the MDB Integrator at DPCE.
3. Conversion of all the accumulated data to the new MDB DM.
4. Execution of the remaining task: IDU-SCN, IDU-BIAS, IDU-APB, IDU-LSF/PSF and IDU-IPD.

Step 2 is only possible when the previous step has been completed. This fact introduces a processing barrier for the execution of the

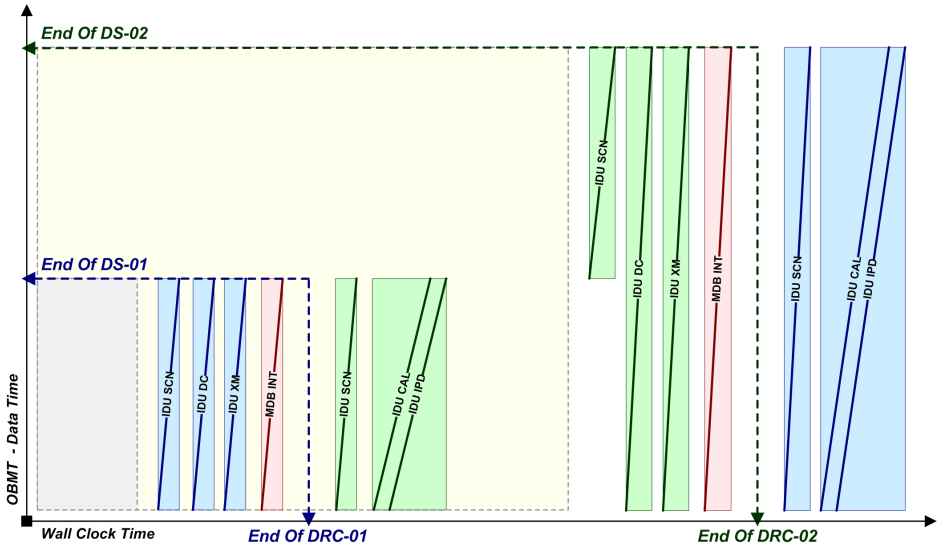


FIGURE 5.3: IDU task flow with respect to the time of the data stream (vertical axis) and the wall clock time (horizontal axis) – the DSs and DRCs are defined over these same axes respectively. The height of the boxes surrounding the tasks indicates the data segment length processed whereas the width the corresponding start and end time of the processing. Note that the DRC-02 scenario can be extrapolated to all subsequent DRCs

remaining IDU tasks included in step 4. However, the DM conversion of step 3 is feasible during this period when the MDB Integrator runs although not completely since the new catalogue has to be converted as well.

As explained in Section 3.3, a given DRC is considered closed when the MDB Integrator is run and the new catalogue is distributed to all DPCs. In that sense, during this stage DPCB would be executing tasks within two DRCs. In practice, this DRC transition only introduces the just commented barrier in the processing and the need of executing the DM conversion (which in case of DPCB is not a major issue since the changes in raw data are very restricted).

Figure 5.3 illustrates the IDU task flow with respect to the DS and DRC definitions.

The next sections will summarise the most relevant procedures from the

last two stages – basically those procedures directly related with the execution of IDU tasks as well as the derived activities of the DPCB operators. We refer as DPCB operators to the people in charge of executing and monitoring the IDU tasks but also responsible of all those activities related to the data and resources handling within DPCB. We will also describe some of the more relevant functionalities – at the time of writing still under development – that will be of great help to cope with the larger data volumes in next cycles.

5.1.1 Data Preparation

Data arrives to DPCB on a daily basis following the processing flow of IDT. DPCB receives the data in several transfers – from few to hundreds of transfers of few to hundreds GigaBytes every day. These data then present a large fragmentation and its sorting is not guaranteed due to the downlink priority scheme on board. Data from a relative short time interval on board – i.e. observations of faint sources from a period of one hour from a crowded region of the sky – could take more than one week to be downloaded and processed by IDT and be finally received at DPCB. More details relative to the GTS between DPCs is included in Appendix C.

Most of the DPCs solve this data fragmentation and additionally sort the data transparently by using database solutions. However, as commented before, DPCB operates exclusively with files what has implied the development of additional auxiliary processes to prepare the data before the processing. Operating directly over the received data files is not desirable for several reasons:

- A large number of files and folders is more prone to data losses.
- The data volume get increased, due to the file fragmentation.
- Basic operations like data filtering are more complex, even requiring additional meta data or supplementary indexes to access only to relevant files.

- Large I/O overheads are introduced in the processing.

As it can be seen from the points raised, it is clear that a more intelligent and sophisticated data arrangement is needed. The GTS is used anyway (as it is the standard transfer system for DPAC), but an adequate layer shall be added to it. This improved data handler will not only have to arrange the data in a suitable structure but also has to process the IDU outputs before they can be transferred to DPCE. Additionally, statistics over the input data need to be performed to detect possible data gaps and to ultimately report the results of all the DRC processing.

As soon as the data reach DPCB, the data is preprocessed. This preprocessing is in charge of arranging the input data according to a predefined time or HEALPix file structure. This preprocessing also gathers time and the HEALPix statistics for each separate data type. The time statistics are in the form of number of records per time bin (nominally bins of 1 second) while the HEALPix statistics provide the counter for all pixels in a given HEALPix level, 7 by default (196 608 pixels). This first arrangement is still presenting the same base structure than the original transfers.

The next step is the consolidation of the data. This is done when no more delayed data is expected (approximately two weeks since the first reception of a given time bin) or when a current DS is closed. The first step is the determination of the best file partition. This partition is done according to the file system recommendations described in Section 5.2.4 and the information on time or HEALPix object densities.

The idea is to get equalised time intervals and HEALPix groups to reduce as much as possible the number of files but avoiding also too small or too large files. Too small files would imply more operations with the file system increasing the I/O time and too large files would increase the memory footprint to load the files degrading processing performance.

The time based equalisation is basically taking the time bins and grouping them until a given limit of objects is reached. When this limit is reached the resulting time interval is stored. These time intervals then are used

to configure the *stores* which will arrange all the input data following that partition and producing a new set of data sorted and without the original fragmentation. This time partition can also be done independently for each CCD row if available. Figure 5.4 includes one example of the statistics obtained for the raw astrometric observations and the resulting partition.

On the other hand, the HEALPix equalisation is a bit more complex being a two dimensional process. The goal is again the partition of the data in files with the same number of records but this time according to the location of the sky, practically its HEALPix pixel. The solution implemented is taking advantage of the hierarchical scheme used for the pixel numbering following the following recursive processing:

1. Load the region counters from the initial statistics.
   These initial regions represent the partition limit, the smallest regions that can be produced by this process.
2. Compute the total amount of records, summing all loaded regions. If the resulting count is less than the configured limit no partition is required and the process finishes.
3. Move to the first HEALPix level, which is basically the split of the sphere in 12 equal-area subregions.
4. For each subregion; determine the corresponding amount of records and store those fulfilling the configured limit.
5. Move to the next HEALPix level. This basically means that each of the current subregions is split in four subregions (see Figure 5.5).
6. Replay the step 4 and 5 until we reach the level of the initial regions.

The described process is the default implementation used for the data arrangement. It produces equalised regions which can be named with the corresponding pixel name at the given level (see Figure 5.5). This naming is really convenient and simple for limiting the number of files to load according to a given sky region.

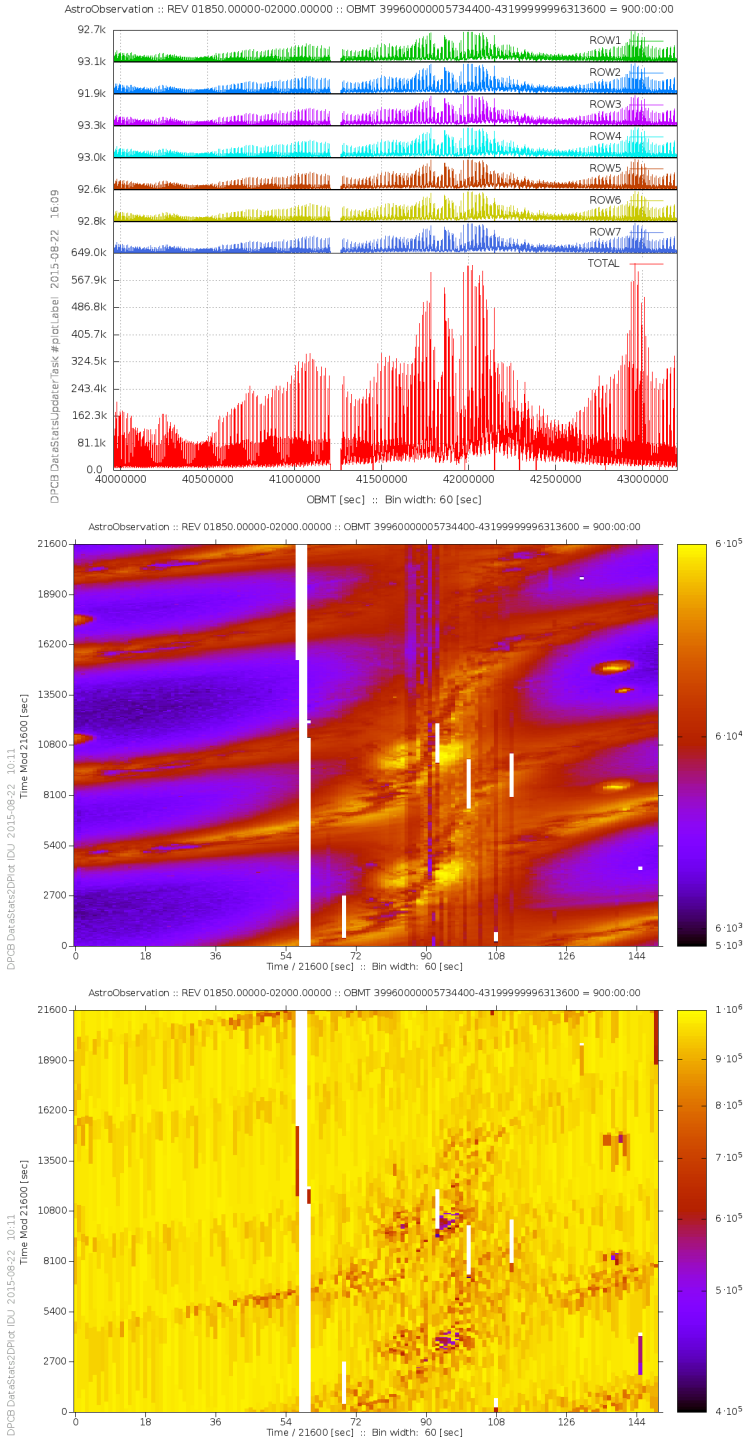


FIGURE 5.4: Example of the time statistics for raw astrometric observations represented in 1D and 2D and the resulting time interval equalised partition (bottom). In the 2D plots the time advances from bottom to top and left to right so each vertical column represent 6 hours of consecutive observations

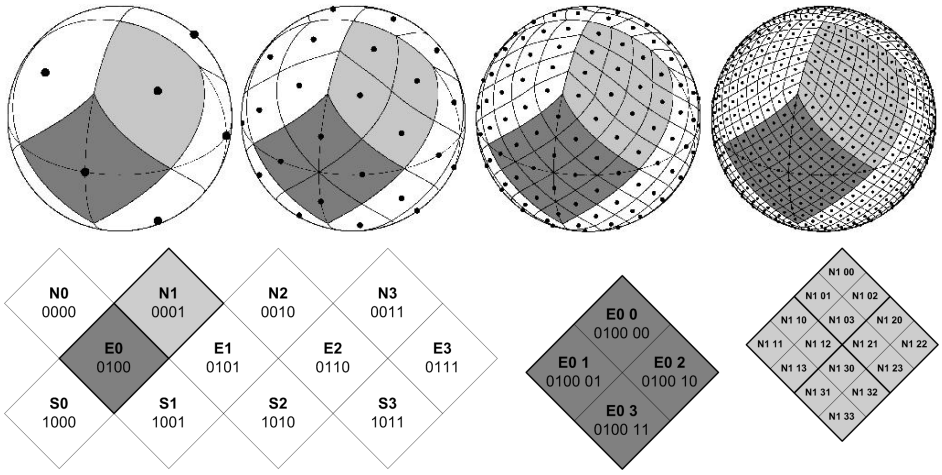


FIGURE 5.5: On top the orthographic view of HEALPix partition of the sphere. From left to right the grid is hierarchically subdivided in subsequent level starting from 0 (Credit: Gorski et al. [2005]). On bottom, the pixel naming convention (in bold) and its binary representation adopted for Gaia for level 0 and 2

Alternative solutions have been implemented where neighbour regions at the same level can be grouped for the creation of the final equalised regions. This gives more flexibility to generate final regions having a more similar number of records but on the other hand naming these multi–pixel regions is more complex. In Section 5.1.2 we will see how these multi–pixel regions are more useful for the job load balancing.

Figure 5.6 shows the equalisation obtained for the IGSL catalogue fixing a limit of 10^6 sources per region for both implementations. In this figure, it can be clearly seen that both solutions follow the sky object density but also that the second one create more equalised group counts.

One may argue that this file arrangement is not actually required and that by simply sorting and arranging the data in multiple files together with some kind of meta data with the corresponding HEALPix information would provide the same benefits. However, if we take into account that the jobs will be based almost in the same equalisation then the arrangement described before should considerably reduce file system access collisions.

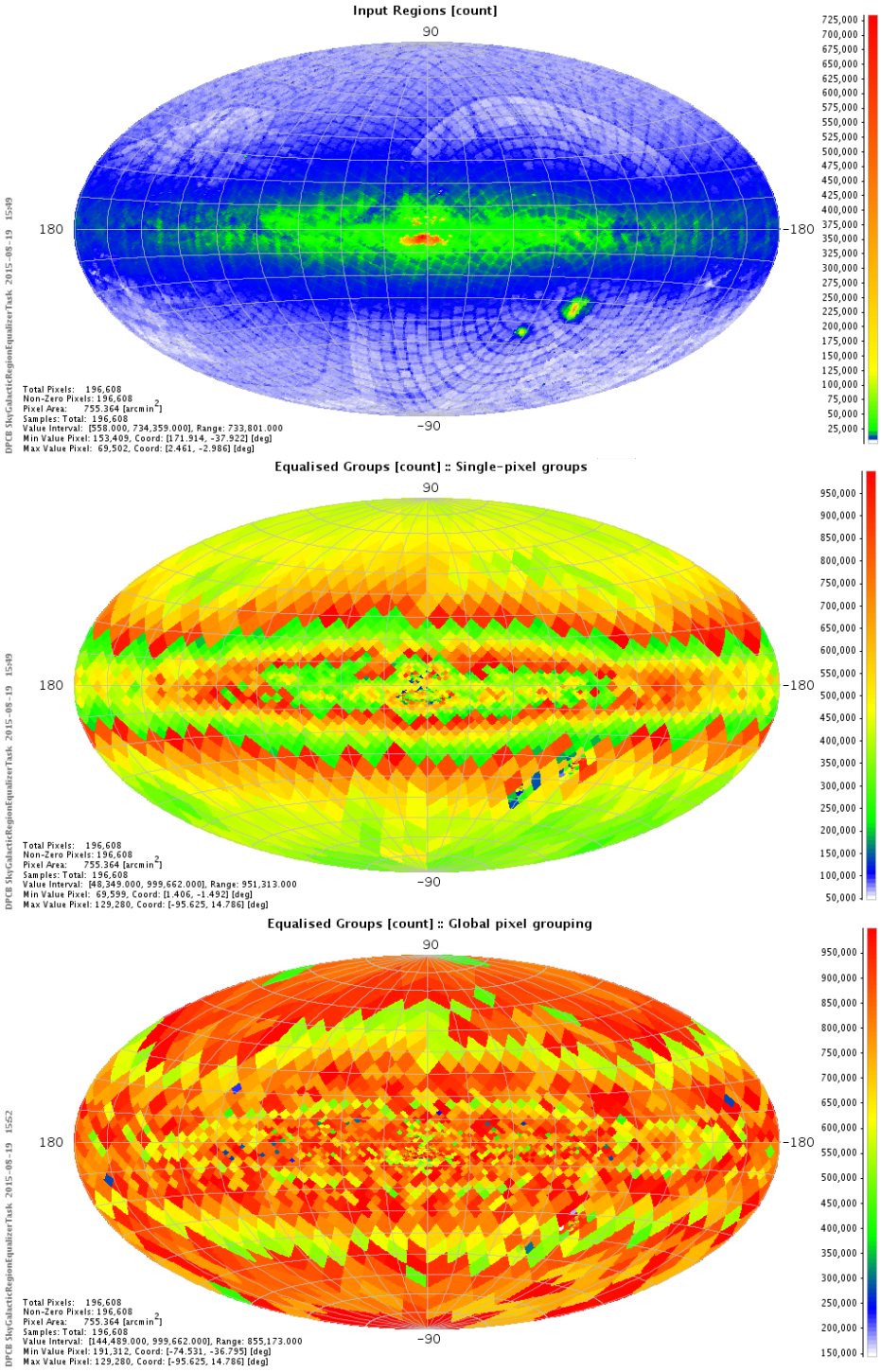


FIGURE 5.6: Sky region equalisation examples. On top the original input regions counts at HEALPix level 7. Middle panel shows the resulting region counts for an equalisation based on multi–level single pixels while the bottom panel is using the unrestricted/global neighbour pixel grouping which results on a more equalised group counts

Finally, it is worth pointing out that this kind of data arrangement – with a physical meaning — provide some additional advantages compared to other strategies. With this arrangement, the extraction and distribution of selective partial data to IDU contributors for the analysis of general job failures or performance issues is much easier.

5.1.2 Job Definition

As anticipated, the equalisation of the IDU tasks in the form of a set of jobs is also a fundamental topic to get an efficient batch processing at DPCB. In this case, the equalisation is not simply referring to the number of objects or data volume to be processed in each job but to the resources required to process those inputs.

In this context, the job definition refers to how a given processing task is split in the form of distinct jobs for their batch processing in the computing cluster. This job definition is mainly based on two inputs: the detailed statistics on the input data but also the task performance metrics. In Section 5.3, we describe the main task profiling metrics that IDU framework produces when a given task is executed. These metrics characterize the task processing time and memory usage and together with the statistics on the input data will allow determining the best task definition to optimize DPCB resource use.

We distinguish two kinds of job definition strategies:

Time–Based

This strategy is applied to most of the IDU tasks: the IDU-SCN, the IDU-DC, the *Obs–Src Match* step in IDU-XM, the IDU-BIAS, the IDU-APB, the IDU-LSF/PSF and the IDU-IPD.

All these tasks, except the IDU-BIAS and IDU-SCN for SSOs, use the statistics on the raw observation time density for dividing the entire time interval to process (time covered by all the DSs entering the current DRC).

Regarding the IDU-BIAS task, the only input data used are the Pre-Scan data. The Pre-Scan data volume is really small and the time density profile is almost flat so no equalisation is really needed. The IDU-BIAS task is then simply divided in regular time intervals according to the available wall clock time and the available processing resources.

Similar is the case of the IDU-SCN for SSOs, the time dependent input data volume is negligible and the output data as well. In this case, we simply split the full interval in regular chunks to fit the resources and the execution wall clock constraints.

Finally, it is worth explaining why the jobs for the IDU-SCN for the catalogue sources are also defined according to the raw observation time density profile. Although this task does not use the raw observations (see Figure 4.2), its outputs – considered also a good performance indicator – are expected to present the same time density profile.

Another option for the equalisation of this task (also applicable for the *Obs–Src Match*) would be to combine the sky region density information with the estimated scanned region provided by the *AttitudeToHealpix* tool (see Section 4.3). The resulting equalisation would be for sure more accurate since it will take into account the real density of the region covered during the scan, but probably the effort would not be worthwhile.

Spatial–Based

This strategy is used mainly by the IDU-XM task, specifically the *Sky Partitioner* and the *Match Resolver*. These tasks could use directly the spatial density profile from the observations but a more accurate equalisation (with respect to the memory usage) can be obtained if the spatial density profile of the source candidates is retrieved from the *MatchCandidates* produced in the *Obs–Src Match*. In that sense, we would be accounting for both the time and the spatial object density.

Practically, the jobs are defined in the form of XML files as described in Section 5.2.2. These job files are stored and in most cases will define statically the processing of each DS for the current but also the future DRCs. However it is expected that the initial job definition may eventually fail in some cases – incomplete input statistics or biased performance metrics for example. In these few cases, the Extensible Markup Language (XML) file of these jobs can be redefined (applying further splitting or merging) or specific parameters added or overridden. These updates can then be also persisted for proper tracking in forthcoming DRCs. The job parameters will be further discussed in Section 5.2.2.

Finally, as for the data arrangement solution, this job definition can be also exported and used outside the DPC. In other words, these job files can be distributed to IDU contributors who will be able to reproduce the job execution in almost the same conditions. This is extremely useful for finding the underlying issues of job failures or job performance degradation.

5.1.3 Job Execution

The overall task flow of IDU has already been described in Section 5.1 as well as how each task is divided in several jobs for its execution in Section 5.1.2. Therefore, the only pending topic to be described is how the jobs are executed within DPCB.

The DPCB processing resources (the computing nodes) can only be accessed through job submission to a shared job queue (see Figure 5.7). The queue system integrated is the IBM Platform Load Sharing Facility (IBM-LSF): a powerful workload management platform for demanding, distributed High Performance Computing (HPC) environments. It provides a comprehensive set of intelligent, policy-driven scheduling features that enable the prioritization of jobs according to the requested resources but also ensure optimal exploitation of the computing resources. It must be noted that no direct access is granted to any computing node which adds additional complications when deploying, executing and monitoring the IDU task jobs.

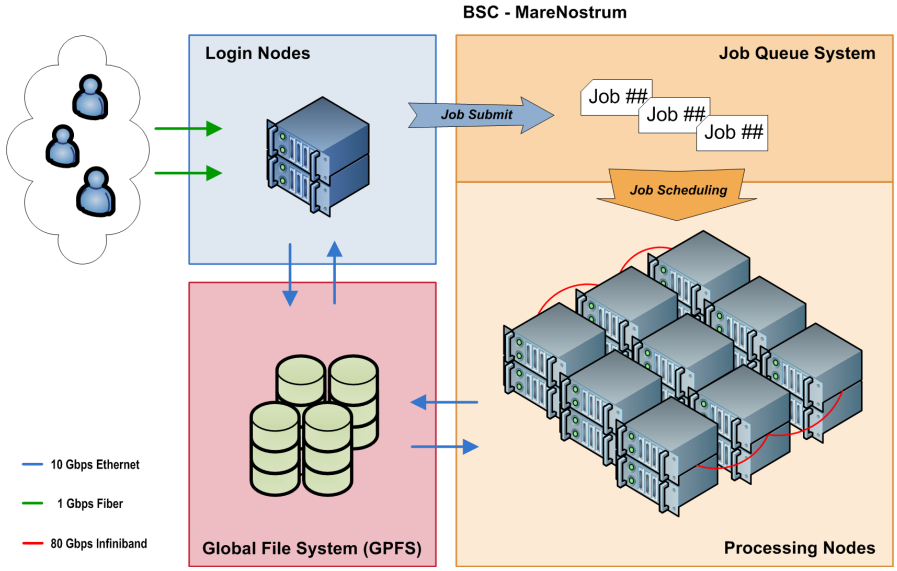


FIGURE 5.7: Diagram of Marenostrum computing resources and access policies

First of all, it is necessary to clarify the different scope of the IDU task jobs against the *queue jobs*. The first one represent the processing of an IDU task over a given chunk of data whereas the *queue jobs* represent the resource allocation request to the Marenostrum queue system and the subsequently granted resources. The *queue job* could hold the execution of one or more jobs from one or several IDU tasks without any kind of restriction.

For a *queue job*, the operator can request from one single CPU to thousand CPUs for a maximum wall clock time of 72 hours. Then, this *queue job* will be scheduled according to corresponding user queue priority, the requested resources and the already submitted jobs. Jobs limited to a single node (less than 16 CPUs) are automatically redirected to an special queue. This queue guarantees a quick response in time (matter of hours) but at the same time does not restrict the node access to a single user. This queue is really useful for executing light processes or for testing software deployments. In all other cases, the job will always get the nodes in exclusive and will be assigned a priority which will increase according to the job size (larger resources) and job wall clock time. In that sense, bigger jobs are considered

better because they reduce the job fragmentation on the queue and thus the dead times of new job allocations. Furthermore, the resources available for the single–node jobs are quite limited which means that you will obtain a very low job concurrency – only few jobs will be running at the same time.

The IBM-LSF provides a complete toolbox for the interaction with the job queue [IBM, 2015]. Jobs can be submitted, paused or killed but also grouped. Additionally, in IBM-LSF, whether a job should start can be dependent on other jobs, usually based on the job states of preceding jobs. Finally, the allocated resources for each individual job can be monitored as explained in Section 5.3.

IDU tasks are implemented and executed as stand–alone applications and can be distributed as follows:

- Independently; so one *queue job* is allocated for each separated IDU job, thus requesting resources from a single node.
- Grouped; one *queue job* for the execution of several IDU jobs in multiples nodes.

In general, as already commented, the preferred option is the second one, thus reducing the number of jobs in the queue. However, this implies the need of some framework in charge of handling the granted resources — in other words in charge of distributing and managing the jobs among the nodes allocated. To solve this issue we have two solutions available:

BSC Greasy

BSC offers for the HPC applications a tool called Greasy. This tool is able to run in parallel a list of different commands, schedule them and run them using the allocated resources of a given *queue job*.

This tool has been developed for those applications requiring the execution of hundreds/thousands/millions of jobs without any inter–communication among them.

One of the main principles of Greasy is to keep it simple for the user, the list of tasks is just a text file listing the commands corresponding to each job to be executed. Then, each line in the file becomes a job to be run by Greasy. Furthermore, it is able to manage dependencies between tasks, or to rerun a task in case of failure (retry mechanism).

During the execution, Greasy generates helpful execution reports where all greasy actions are recorded to keep track of what is the progress of the run. From these reports, the user can check the proper execution of the job chain and analyse the performance of the jobs. Additionally, the corresponding lists of failed tasks and pending tasks (not even allocated due to *queue job* time limit) can be extracted for their subsequent execution.

Internally, Greasy is based on the *Master/Slave* load sharing model, where one *Master* process has unidirectional control over one or more *Workers*. These workers are in charge of running the jobs served by the master. Greasy workers are configurable to get different configuration over a given set of nodes. Basically, user can fix the number of workers to be executed in each node, which practically result in a uniform distribution of the resources with the same number of CPUs, memory and local hard disk allocated for each worker. This uniform partitioning could be considered as a limitation of this tool, not allowing the execution of jobs with different resource needs. Figure 5.8 shows a schematic view of a Greasy based job.

Another issue of this tool implementation is the resource usage when Greasy is processing the last jobs. During this last phase a lot of resources may become idle if the resource arrangement (number of nodes and wall clock time) of the *queue job* do not match with the jobs processing profile. This can be easily mitigated by means of a good job equalisation and the corresponding *queue job* dimension and scaling but it is almost impossible to have all jobs using all the resources up to the very end of the processing.

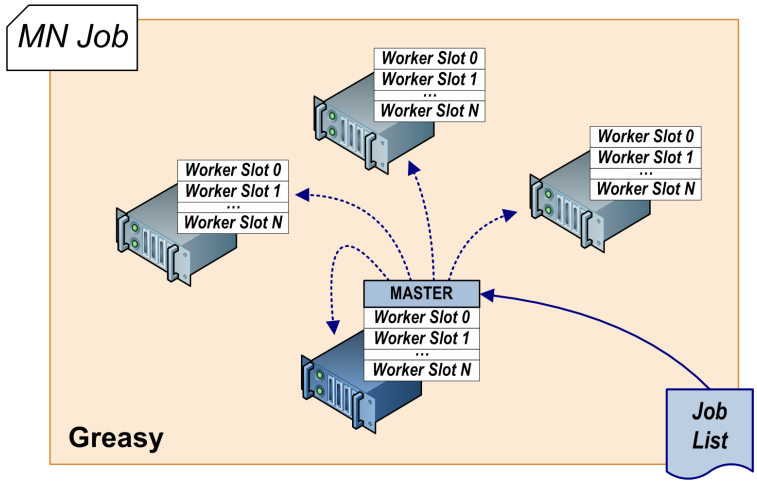


FIGURE 5.8: Diagram of the Greasy execution framework for a given Marenostrum job, where worker slots are assigned statically to each computing node and a master process is in charge of distributing the user jobs

BSC is working in the possibility of getting dynamic allocation of resources so the *queue job* can release nodes when idle and no job allocations are pending.

DpcbTools NodeCoordinators

We have developed within DpcbTools an execution framework to manage, as Greasy does, the parallel execution of a list of different jobs. This framework implements a similar *Master/Slave* load sharing model but in this case local to each processing node in the form of a *NodeCoordinator* which provides additional functionalities not available in a pure Greasy environment. In practice, this framework is deployed to the computing nodes using Greasy but the job scheduling and execution is in charge of the these deployed *NodeCoordinators*.

The full description of this framework is provided in Section 5.2.3 as it has been one of the main contributions of this thesis to the IDU execution framework.

Greasy has been and is still used a lot for the execution of the CU2 simulator. For these simulations, Greasy has been used to execute more than 50 thousand jobs in 200 nodes, consuming more than 200 thousand CPU hours

in a single *queue job* for the generation of approximately a half terabyte of data corresponding to a catalogue simulation.

For the first executions of IDU tasks, Greasy will also be the preferred tool. However, as data volume increases and tasks gets more complex, the more sophisticated *NodeCoordinator* solution will be used. Exhaustive tests are ongoing at the time of writing, already showing the benefits of the dynamic and interactive functionalities of the *NodeCoordinator* framework.

5.1.4 Results Handling

As for the data input, the volume of the results will also be considerable (see Appendix D). Additionally, the results will also present a high fragmentation due to the job partitioning used during the batch processing. Therefore, new data arrangements will be required.

In general this arrangement will only imply the data merging in a more efficient data structure for the next task or for its tranfers to DPCE. However, in some specific cases it may involve the swapping from time–based to spatial–based structures and vice versa. The main interface requiring this swapping operation is the *Match* table produced by the IDU-XM task (see Section 4.3). This arrangement is done similarly to the ones done for the input data coming from DPCE so the same tools and equalisation parameters can be reused.

The results from each individual IDU task job are tagged with a unique solutionId, described in more details in Section 5.2.3. Together with the solutionId an auxiliary table is also produced logging the solutionIds of the input data involved in the processing of the corresponding output solutionId. This table is called *InputDataUsed* and is essential to be able to reconstruct the processing history providing a clear link between the input data features and the characteristics of its derived outputs. In that sense, the granularity of the DPCB solutionId is more than desirable and provides a valuable information to analyse the IDU results. Finally, it is worth mentioning that some of the solutionIds produced at DPCB will be completely

internal to the DPC and will not be distributed to the MDB. One clear example is the intermediate data produced in the first and second steps of the IDU-XM. For these cases, a solutionId consolidator has been implemented to post-process the *InputDataUsed* table removing these intermediate solutionIds. The final *InputDataUsed* table will then only contain solutionIds present in the MDB.

Occasionally, some IDU outputs can be considered incorrect or even unusable. Although, the retraction of already transferred solutionIds to DPCE will rarely occur, DPCB will eventually retract some of its outputs. The main causes for a retraction would be:

- A retraction commanded by DPCE on the raw data accumulated at DPCB. This may happen well before the start of the IDU processing or during the actual processing. In the first case DPCB will apply the retraction before starting the processing whereas in the second case, which should rarely occur, DPCB will be forced to hold the current processing and resolve the conflicts, notifying the retraction of the data already transferred if necessary.
- A software bug in any of the IDU libraries, leading to a new software release.
- A configuration error in the execution of a given task.
- An inconsistency in the execution environment or resources used, as for example incorrect data repositories, wrong Java version, use of an old deployment, etc.

All these retractions may imply the removal of a reduced or rather huge set of solutionIds. The amount of data to be retracted will depend in the primary affected inputs and outputs and their relation to subsequent tasks, which can trigger the retraction of more outputs in cascade.

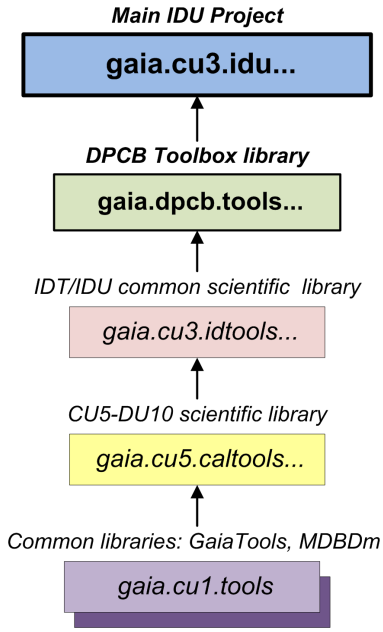


FIGURE 5.9: Java library dependency tree of IDU project

5.2 IDU Implementation

During the design stage of IDU, the requirements and constraints of all IDU tasks were compiled together with the main processing capabilities of the DPCB resources. For this activity, the stay in the Institute for Astronomy - Royal Observatory of Edinburgh (IfA-ROE) in 2010 and 2011 was of great help to settle and collect information for IDU-BIAS, IDU-APB and IDU-LSF/PSF tasks. Additionally, the experience obtained from CU2 simulations activities at DPCB provided a very good basis for the design and development of the IDU software.

As commented in Chapter 4, IDU integrates seven tasks in charge of different groups within DPAC. The processing core of these tasks are implemented in two libraries; *CalibrationTools* and *IDTools*. The first one is a library maintained by the IfA-ROE team whereas the second one is responsibility of the CU3-UB team with some contributions from the CU3-Torino team. Figure 5.9 illustrates the tree of library dependencies of IDU. Each library contributes to different IDU functionalities as follow:

- CU1 common libraries (including the MDB DM); shared by all DPAC systems. These libraries define the main interfaces and provide general implementations for the more basic processing tasks: spacecraft configuration handling, attitude and geometric calibration servers, coordinate transformations, solutionId tracking, telemetry and Gaia Binary File Format (GBIN) reading and writing routines, etc.

- *CalibrationTools*: CU5-DU10 software library (developed by the IfA-ROE team) including all the routines for the treatment and calibration of the Bias, Astrophysical Background, CCD calibrations and LSF/PSF. This library also includes the new developments for the IDU-IPD related to the 2D mapping (see Section 4.6.1).

- *IDTools*: common CU3 library shared by IDT and IDU mainly developed and maintained by the CU3-UB team. As already commented, both systems implement similar functionalities and the creation of this intermediate library reduces a lot the code duplication but also allows the direct migration of any fix or improvement between both projects. This library implements the processing core of the Cross-Match, the Detection Classifier (DC), the IDU-SCN and the IDU-IPD. Additionally it implements most of monitoring and validation functionalities already described in Section 4.8.

- DpcbTools: library to assist the execution of IDU tasks at DPCB. This library, developed also by the CU3-UB team, contains utilities for performing I/O; data manipulation; communication; task creation, job scheduling and launching; data visualisation; and monitoring. This library implements additional functionalities related to the GTS which have been described in Appendix A.2.1.

- IDU: main project where the IDU tasks are ultimately implemented and integrated. Strictly speaking this project does not contain any relevant scientific routine. Scientific implementation is completely entrusted to *IDTools*, *CalibrationTools* and *GaiaTools* libraries. In that sense, this project basically integrates the scientific routines from these libraries with the framework provided in DpcbTools. The

design and implementation of this project is responsibility of the CU3-UB team.

In this section we will cover the most relevant design issues adopted for the current IDU implementation. Firstly, we will present the general software development guidelines that all DPAC software must follow, covering the adopted standards, the programming language, etc. Afterwards, we will focus in the more technical details of the IDU task framework design. A specific section is devoted to the data access layer, fundamental concern since it could potentially become a major bottleneck in the processing at DPCB. Finally, a last section is devoted to summarising the overall testing strategy that is followed for the software being developed.

5.2.1 Development Guidelines

The data processing system for Gaia can be seen as the combination of machines, people, and software processes that for a set of inputs produces a defined set of outputs. For the achievement of this goal, it is mandatory that all the parts follow a common set of guidelines to guarantee the proper communication and compatibility. This section summarised the more relevant guidelines that all DPAC processing system must follow.

DPAC has adopted the European Cooperation for Space Standardization (ECSS) system, specially the software engineering standards for science ground segment development. This ECSS system covers the management, engineering and product assurance standards. These standards have been tailored for Gaia and summarised in Lock [2007]. In practice, this implies that any piece of software must be attached to the corresponding documentation describing the purpose and requirements of the software, including the traceability of these requirements with the actual software code. Additionally, software manuals, test reports, code quality reports, etc. must be also provided. For IDU, we have followed rigorously these standards, producing a lot of documentation within the frame of this thesis.

As the data processing at the DPCB may continue until around the year 2022 or a bit later, the hardware at the DPCB will go through a number of major upgrades, and may completely change before the mission is complete (see Appendix A). The other DPCs will also go through several changes to their hardware, and at the time of writing, it is impossible to know the hardware architecture that will be available at the end of the mission. For this reason, it is very important that DPAC systems are developed to be portable, and not dependent on any particular hardware features.

For this reason, the majority of DPAC systems are entirely written in Java. Java is one of the most widely used computer programming languages in the world [Tiobe, 2012]. It is a general purpose, Object-Oriented (OO), high-level language, with a syntax quite similar to C++. Java is used for creating software for all kinds of uses, and it runs on a vast range of hardware environments from hand-held devices to supercomputers [Fries, 2012]. A fundamental strength of the Java platform is its portability. This has huge advantages when distributing software amongst a group of developers who may be using different operating systems or different hardware environments. Additionally, Java is generally considered safer and easier to master than other programming languages, implying higher developer productivity thanks to the number of lightweight built-in features designed to allow for its monitoring and debugging.

For systems like IDU – integrating several software routines from external groups – choosing Java was one of the best decisions DPAC has ever made. This decision is also reinforced by the extensive set of tools that are available for the development and maintenance of Java software. Furthermore, DPAC has compiled in O'Mullane et al. [2006a] a summary of the more relevant Java development guidelines, based on common practice in the Java community.

Finally, it is also relevant to mention that DPAC has enforced the creation of CCBs at DPC and CU level. These CCBs are in charge of the management and approval of any relevant change affecting the software and documentation. In general, the majority of the CCB members must

be external to the software development teams and are chosen according to their expertise on the software characteristics. The CCBs play a very important role at the time of creating the software releases as well as when software updates are required to solve any software bugs.

5.2.2 Job Interface

Adopting a flexible job interface is essential to be able to cope with the particularities of all IDU tasks. In order to define this job interface we need to identify the input and output requirements and dependencies of each individual task. For this task, we have created a document template with the guidelines to define the IDU tasks [Castañeda, 2009]. This template not only requests the proper definition of all the input and output interfaces but also requests specific information about:

- Expected data flow: data access policy, access frequency, etc.
- Estimation of the input and output data volumes for the overall task
- Minimum and maximum job dimension. In practice, this information refers to the minimum amount of data required for a meaningful execution of the task and the maximum amount of data the task could handle.
- Expected scalability of the processing; whether it will be linear with the data volume or not.
- Identify data dependencies with jobs from the same task and also from the other tasks.

Once we have all the task well characterized, it is also fundamental to identify which additional job parameters would be required to extend the functionalities of the execution framework. The most relevant functionalities would be the following:

- Capability to detect and kill stuck jobs. This can be achieved for example by simply defining a parametrised processing time limit so the job manager kills the job when this time limit is reached.
- General procedures to improve the fault tolerance even recovering completely from processing failures. This functionality would imply a resubmission of the failed jobs according to some predefined procedure. These procedures could include updates on the JVM parameters (to solve resource issues) or the use of alternative task configurations. Also the basic retry mechanism is more than desirable to be able to cope with occasional file system failures.

The flexibility of the job interface has been assured using XML for the definition of all job parameters. At the time of writing, each job includes the following information:

- Task, in the form of fully qualified name of the implementing Java class.
- Time To Live (TTL); to limit the maximum execution time of the job.
- JVM arguments; in general to set the JVM memory parameters.
- Java Properties files; fixing the overall task configuration.
- Time Interval; defining the mission time extent to be processed in the job. This time interval is mainly used to filter the input data and to restrict the extent of the outputs.
- HEALPix region: defining job spatial extent which is also used to filter the input data and to restrict the output extent.
- Data Repositories; defining the base location of the input, output and working/local repositories.
- Data Stores; for the specific definition of each input data types. It can also define bounding data types, additional file filters, file pattern, etc.

Most of these parameters can also be fixed globally for each task type, simplifying the XML and reducing its size. This XML configuration can be also persisted to disk and then be reused in next executions or distributed to the IDU developers for debugging purposes.

5.2.3 Execution Framework

Since DPCB resources are not exclusively dedicated to Gaia and the hardware and administration is not under DPCB management, not all the CU1 infrastructure already developed for other DPAC system is suitable to be used in this DPC. This infrastructure is mainly designed to work against a central database and the execution framework deployment on the processing nodes requires direct access to the processing nodes.

Adapting the CU1 infrastructure to the DPCB constraints, although possible, was not considered the best option, mainly because it was actively used and updated according to the needs of the daily systems at DPCE. This circumstance could have caused undesired delays and even the implementation of dubious solutions to be able to cope with the requirements of both DPCs. Therefore, it was decided that a specific execution framework should be developed. This framework should be able to handle the launching of IDU tasks in the assigned computing nodes and managing the huge input and output data volumes, while efficiently exploiting the available computing resources. IDU is quite data-intensive, in fact, it is the most data–intensive of all the DPAC systems.

As anticipated at the beginning of this chapter, IDU framework has been developed with the batch processing in mind. This decision was largely based on the nature of the IDU tasks where a lot of data has to be processed by lousily coupled tasks. Besides, the high partitioning of all the tasks is also fundamental to understand the implemented solution. Taking into account all these properties, the problem is then reduced to finding the best way of distributing the task jobs efficiently among the computational resources.

As mentioned in Section 5.1.3, BSC provides a *Master/Slave* based framework for this kind of applications, called Greasy. This framework however has some limitations:

- The total resources are uniformly distributed among all the workers.
- Static job execution sequence (the one defined in the task list file). This job sequence is static but job dependencies can be defined.
- Task list limited to hundreds of thousands of jobs.
- No dynamic job prioritization.
- No dynamic job creation.
- Limited fault tolerance, just job retry mechanism and logging.
- Manual assignation of solutionIds to each job

None of these limitations disqualifies Greasy for its usage for IDU but in practice a more sophisticated and integral solution is more than desirable. For this reason, we have developed an alternative framework which can be seen as a Java alternative for Greasy tailored according to IDU needs.

The main difference between the two frameworks is the *Master/Slave* topology. Greasy creates statically a single *master* which distributes the resources among a fixed number of *workers*. In our framework, on the other hand, an additional control layer is added by creating dedicated *masters* (called *NodeCoordinators*) in each computing node. Each *NodeCoodinator* will be then in charge of executing the jobs by dynamically creating the necessary workers according to the job resource request. Figure 5.10 shows a schematic view of a *NodeCoordinators* based job. In fact, these *NodeCoordinators* are currently deployed using Greasy, just configuring a single worker per node.

Each *NodeCoordinator* implements an internal local queue where the jobs are prioritized according to their configured parameters. This prioritization is based on:

- The individual requested resources. The jobs with higher resource needs and thus probably the longest ones will be the first to be executed. Running the longest jobs first in general benefits the overall load balancing since we are avoiding that these long jobs are executed just at the end when several nodes may already be idle.

- Job dependencies; independently of the original input sequence the jobs having more consumers are executed first.

- Proper job sequence; basically the jobs will be sorted by its natural ordering, by time or sky region. This behaviour will basically benefit the progressive completeness of the output stream.

- Job status; re–queued jobs will also get the highest priority to provide as soon as possible indications about any error in the software deployment, task configuration or any failure on the allocated resources. This quick feedback is fundamental to be able to apply the necessary corrective actions to avoid wasting resources.

The assignation of jobs to the *NodeCoordinators* can be:

- Static; defined at deploy time, in this case the input jobs are uniformly distributed among the allocated nodes. This method isolates the *NodeCoordinators* not being possible the load balancing between nodes.

- Dynamic; jobs are initially distributed as in the previous method but the local job queue becomes public allowing the transfer of jobs from one *NodeCoordinator* to another. This strategy minimises the communication between the *NodeCoordinator* but enables the inter–node load balancing as soon as all local queue jobs have been processed or are already running.

The communication between *NodeCoordinators* is based on the establishment of point–to–point channels using Java Message Service (JMS) or Java

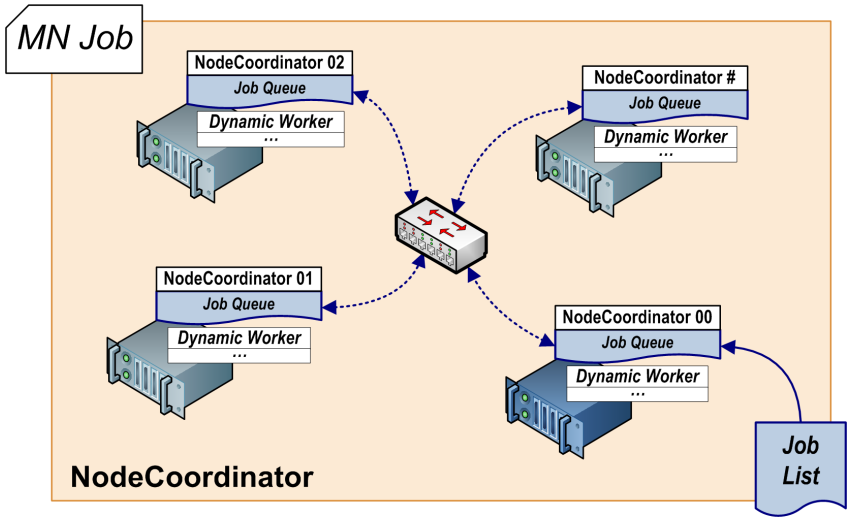


FIGURE 5.10: Diagram of the *NodeCoordinator* execution framework for a given Marenostrum job, where a *NodeCoordinator* and local job queue are assigned statically to each computing node. Each *NodeCoordinator* is in charge of launching workers for executing its own jobs. Jobs are initially homogeneously distributed among all the *NodeCoordinators* but their queues are shared allowing job redistribution

Remote Method Invocation (RMI) Java APIs. Furthermore, an additional *service* channel is also available. This *service* channel can be used to check the *NodeCoordinator* status, change some configurations, stopping and pausing the queue and eventually queuing more tasks. This *service* channel thus removes the limitation of having static job lists but also provides on demand real time information about the node status.

Another implemented feature is the possibility of linking and unlinking *NodeCoordinators* to a *hot* group. This feature permits the connection and disconnection of *NodeCoordinators* to any group of interconnected *NodeCoordinators* already running without affecting the existing instances. The new *NodeCoordinators* would basically connect to all the already available instances and the jobs from each local queue would be then available to the new instances. These new *NodeCoordinators* could be launched to the Marenostrum queue in a new *queue job* as a processing resource reinforcement or simply to migrate all the pending IDU jobs to a new *queue job* extending the runtime hard limit of 72 hours of the *queue jobs*.

Although this framework is already operative, its exhaustive testing is still pending in an official test campaign. Once this campaign is done and the corresponding CCB approves the release it will replace the current execution baseline which is still based on Greasy. This testing has not been carried out yet mainly due to all the effort diverted to other operational activities as the early execution of the IDU-XM included in Chapter 6.

Additionally to the job management, we also envisage the implementation of the following functionalities:

- *NodeCoordinator* Data Exchanges: adding the capability to collect and redistribute data between *NodeCoordinators*. This feature could be useful to reduce the direct access to the global data repository (see Section 5.2.4) from each individual job. In practice the *NodeCoordinator* would determine and retrieve part of the input data (replicating partially the contents of the global data repository in the local hard disk) before launching the job. In the same way, the job would store the final outputs in the local hard disk and the *NodeCoordinator* would be in charge of storing the results in the global data repository. These feature would be only necessary in later DRCs when the data volume and job number could be too high to allow the direct access from each individual job.

- NodeDataCaches and NodeDataServers: following the same goal than the previous point, reduce the direct access to the global data repository but also aiming to exploit the high performance network connecting the nodes (see Appendix A).

- TaskMaster: a master process in charge of the full automatisation of the execution of all IDU tasks. Currently the decision of starting the execution of the next task is still a manual operator action.

- Dynamic job definition: so job partitioning of a given task is created according to the results from previous tasks. This functionality could be very useful for the second and last steps in the IDU-XM where

the subsequent job partitioning depends on the produced *MatchCandidate* and *MatchCandidateGroups*.

- Job dependency determination according to additional consumer information of each output data product defined at task/job level

Another fundamental and mandatory feature for the IDU execution framework is the automatic assignation of unique solutionIds to each job. This is required to tag and track specific information related to the data processing; input data, configuration used, etc. A centralized management for the assignation of these solutionIds is again not desirable in IDU environment. To avoid this, the DPCB/IDU solutionId follows the scheme shown in Figure 5.11 and it is currently coding the following information [Portell et al., 2013]:

- Software identifier and version (10+11 bits), as defined in Hernandez [2012].
- A composite execution identifier (42 bits) coding:
  - The DRC identifier (5 bits, 0-31), to easily distinguish data from the different DRC exercises.
  - The task identifier (7 bits, 0-127), for coding the different task types including arranging, processing and validation tasks.
  - The execution node (12 bits, 0-4095), for the pool of ∼3000 nodes currently available in Marenostrum.
  - Spare bits (3 bits, 0-7) for future extensions.

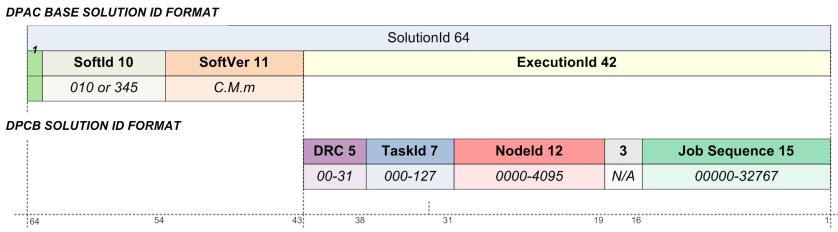


FIGURE 5.11: Decomposition of the unique DPCB/IDU solutionId assigned to each task job

- The job sequence number (15 bits, 0-32767), indicating the job sequence number of each executed job per node.

This solutionId codification has been fixed according to our preliminary estimations on the job partitioning, the current task computational performance and the wall clock time allocated for the execution of IDU tasks within each DRC which ultimately is fixing the number of nodes used in each task run.

5.2.4 Data Access Layer

Contrary to most of the DPCs, DPCB will not use databases but operate exclusively with files. Fortunately, BSC file system is based on General Parallel File System (GPFS) which provides a very good performance and scalability [Schmuck and Haskin, 2002]. Additionally, each processing node has a local hard disk up to 500 gigabytes which is used as temporary working repository and may be used to hold data in the node for subsequent task jobs.

The importance of efficient I/O increases with the number of concurrent processes, and if the data is accessed repeatedly. In the case of I/O-intensive applications as IDU, I/O can become a significant factor affecting the overall performance of the application, and has the potential to become a bottleneck in processing, if not implemented correctly. Furthermore, the DPCB file system is shared with other users which introduce additionally I/O limitations.

The input data used for IDU will reach about ∼100 Terabytes coming from the mission raw data; this will be received from DPCE and will need to be arranged as previously discussed in Section 5.1.1.

When building the final data repository structure we must consider some constraints imposed by the current Marenostrum and GPFS capabilities, namely:

- Not exceeding more than 512 entries in each directory (either as files or subdirectories).
- Not exceeding a folder tree with more than 256 levels.
- Using a file size much larger than the repository disk sector, which is 512 kilobytes for the GPFS.

Fulfilling these constraints [BSC, 2015] will lead to a better performance when accessing the data. These constrains will have to be also applied to the output data – of the order of ∼75 Terabytes. A more detailed information about the overall data volume for the Gaia mission – and IDU in particular – is summarised in Appendix D.

For all these reasons, we have developed a data access layer within DpcbTools providing all kinds of tools for accessing and storing the data. These specific tools extend the functionalities from the CU1 tool box, although some of them have been reimplemented and new ones have been created. These tools provide a transparent data access for all IDU developers, giving a level of independence from the physical storage mechanism and format. For this, it is normal to use some abstraction. Java interfaces provide an excellent approach to provide such insulation with no overheads since they only bind the using class and the providing class with no translation of any kind.

The first concern for the design of this data access layer was the file format. The large data volume required an efficient storage data format, including parallel access capabilities, data filtering and efficient indexing. The CU1 file format standard, called GBIN, was designed as an archiving file format, to backup and transfer the data between the DPCs. This circumstance brings in several drawbacks for the usage of GBIN format during the processing. Consequently, we decided to adopt a more sophisticated file format based on the Hierarchical Data Format (HDF) [Portell et al., 2011]. A summary of the DPCB file format has been included in Appendix E where preliminary performance results and comparison with the GBIN format have been also included.

For the data access, the main tools implemented are the data *stores*. These *stores* handle the data in the form of Java Objects (which is generally termed the Data Model (DM) for the system) and are in charge of:

- Loading the data from the configured repositories.
- DM conversions and the application of data qualification/retractions.
- Serving the data to the processing tasks; some of them even integrating final user functionalities as for example the attitude determination.
- Arranging and storing the output data.

The *stores* provide transparent access to the DPCB data repositories. Taking into account the IDU tasks descriptions provided in Chapter 4, we have devised two main approaches for the structure of these repositories depending on the kind of data. Observational data will be distributed according to a time criterion. On the other hand, source related data will be arranged following a spatial criterion. The *stores*, then, are able to perform selective an automatic file filtering for the data loading according to the time and spatial job parametrisations.

In general, each individual IDU job retrieves the data from a global data repository. However, it is envisaged that most of the jobs running in one particular node may reuse the same input data. This often happens for some calibrations, spacecraft configuration and attitude, etc. This data is in general small in size and retrieving it from the global data repository could introduce performance degradation. In these cases, adopting some shared resource in the nodes could be advantageous to optimize the access to this kind of data. Thus, the *store* interface has been defined also taking into account this possibility.

These shared *stores* would act as local data caches or servers in charge of serving this data to the jobs minimising the number of accesses made to

the shared disk. This operation would be in any case completely transparent to the job and will be managed by the IDU execution framework. Furthermore, these cached *stores* could also be expanded to more than one nodes, thus serving the data locally generated by jobs in other nodes. This functionality is still under development but a deep feasibility study is available in Fries [2012]. This study was focused in the usage of the Message Passing Interface (MPI) for transferring data among the *NodeCoordinators* and thus be able to exploit the high performance network interconnecting the Marenostrum nodes.

Finally, it is worth mentioning that all these tools have been implemented with monitoring built-in features so very detailed I/O statistics are provided for the profiling of the IDU tasks. Thanks to this functionalities, the developer or operator can easily identify any possible bottleneck or performance degradation in any operation against the DPCB data repositories.

5.2.5 Testing Strategy

Following ECSS standards, IDU software is rigorously and periodically tested to guarantee its quality and with the intent of finding software bugs (errors or other defects). Designing these tests is not an easy task since the number of possible tests for even simple software components is practically infinite. For IDU we can distinguish two kinds of test Castañeda et al. [2011c]:

Scientific Tests

To evaluate the overall scientific results of all integrated processing tasks. No deep scientific test is carried on, instead we only check the correctness and consistency of the results obtained and the absence of processing error and exceptions. The specific scientific tests are responsibility of the developers/contributors of each individual task which are covered in other dedicated tests.

Computational Performance and Scalability Tests

To evaluate the overall performance of all tasks and the complete

system. Also for profiling the system in order to find possible bottlenecks and characterize the scaling profiles of each IDU task.
These tests contribute to a better understanding of the software and eventually to a better integration in the execution environment.

All the test designs defined follow the *DPAC system validation and test plan* [Guerra and leaders CU leaders, 2013] and the agreed performance analysis and metrics tools from Hoar [2010].

Besides these tests, which are referred to as *CU level tests*, additional test are included in DPCB documentation. These *DPC level tests* aim to check the proper integration of the software at the DPC so they are focused in the assessment of the correct execution of the tasks and their performance in the DPC hardware [Gonzalez et al., 2015].

5.3 Profiling and Monitoring

As described in Chapter 4, IDU integrates seven different tasks. Each task, presenting different I/O and computational requirements. A good balancing of the task jobs is essential to exploit the DPCB resources and to be able to meet the wall clock constraints of IDU DRC time slots. This balancing is only feasible when we have a good knowledge of the processing performance profile of each task in terms of CPU time, memory and I/O load.

These performance metrics are a built-in feature of IDU framework. Each task provide measurements for:

- Number of elements processed: sources, observations, time intervals, etc.
- Total time elapsed for data loading, data writing and for each data processing algorithm.
- File system timing on file access, copy and deletion.

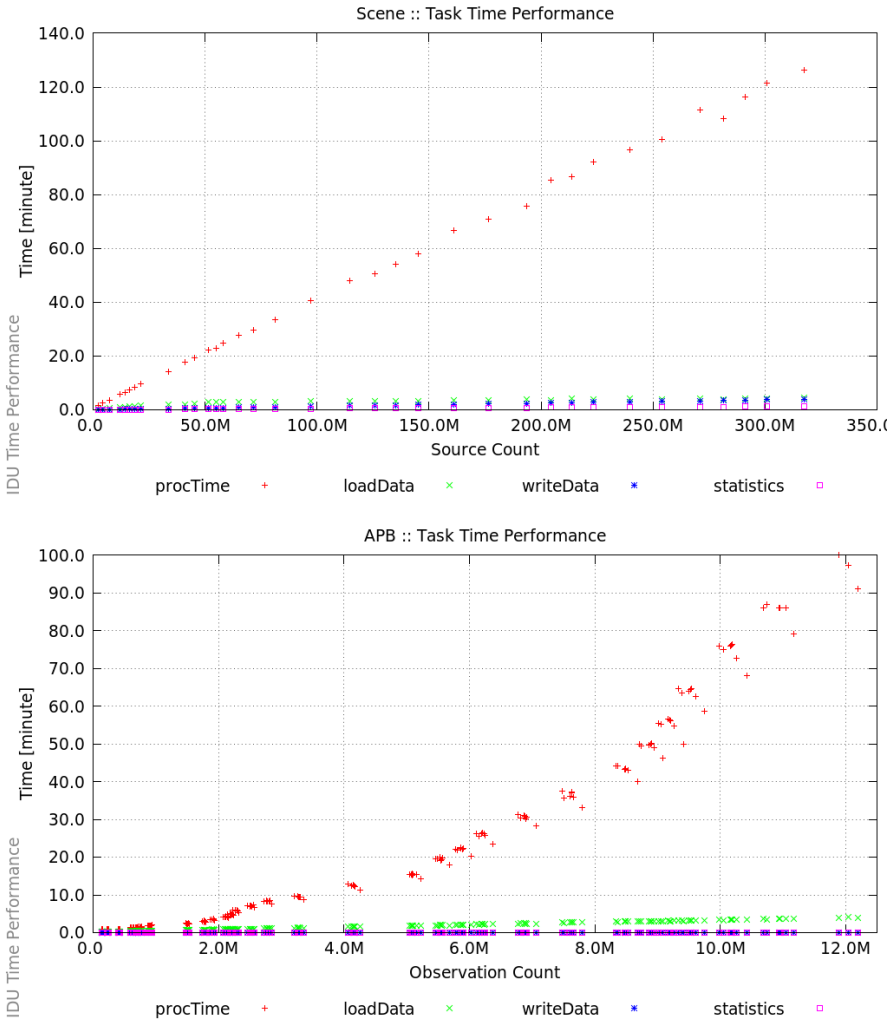


FIGURE 5.12: Task scalability profiles for IDU-SCN (top) and IDU-APB (bottom). The N^2 profile of the IDU-APB processing time as a function of the input observations is noticeable whereas the IDU-SCN present a linear profile as a function of the number of input sources. Both plots include the timing for the different task stages; data loading, pure processing, statistics and data writing

- Total CPU and I/O time accounted for each processing thread.

With all this information several diagnostics can be generated to obtain the scalability of each task to different parameters. These diagnostics provide very valuable information on how the tasks scale when the data inputs are increased; linearly or exponentially as shown in Figure 5.12. These plots

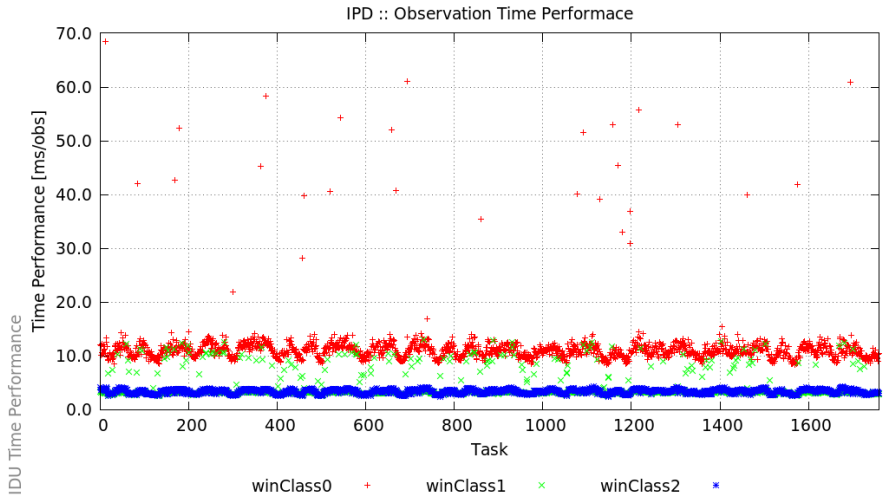


FIGURE 5.13: IDU-IPD processing performance for each observation window sampling class. The time required for the processing of 2D windows (in red) increases by a factor of 3 with respect to 1D windows (in blue and green). Very few outliers can be identified which mainly comes from occasional saturations of the node CPUs

are obtained by executing each task in isolation (one single process per node) with different configurations, mainly covering larger time intervals or sky regions.

Besides this overall task profiling, more detailed information can also be obtained for the profiling of some specific parts of the processing. One clear example has been included in Figure 5.13 where we have analysed the IPD processing performance for each individual window class. These diagnostics are very useful to detect possible bottlenecks or unexpected performance degradation for specific parts of the processing.

Additionally to the task level metrics, the IDU and DpcbTools frameworks provides more diagnostics as:

- Overall node CPU and memory usage (see Figures 5.14 and 5.15)
- Network and storage usage (see Figures 5.16 and 5.17)
- Job queue status (see Figure 5.18)

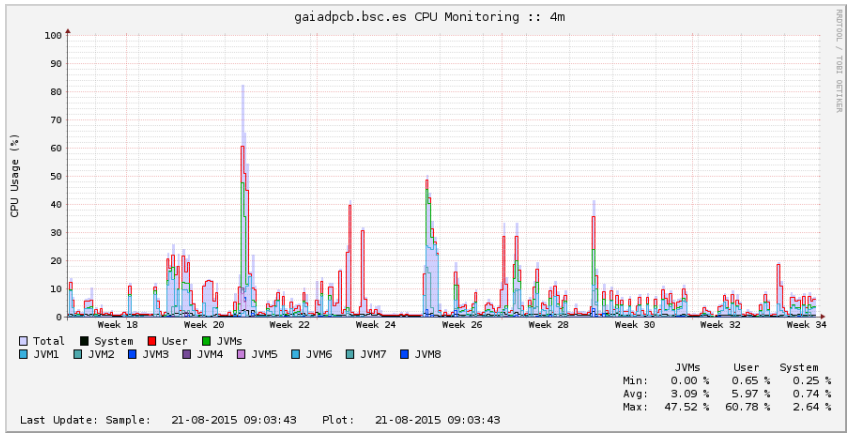


FIGURE 5.14: CPU monitoring plot example

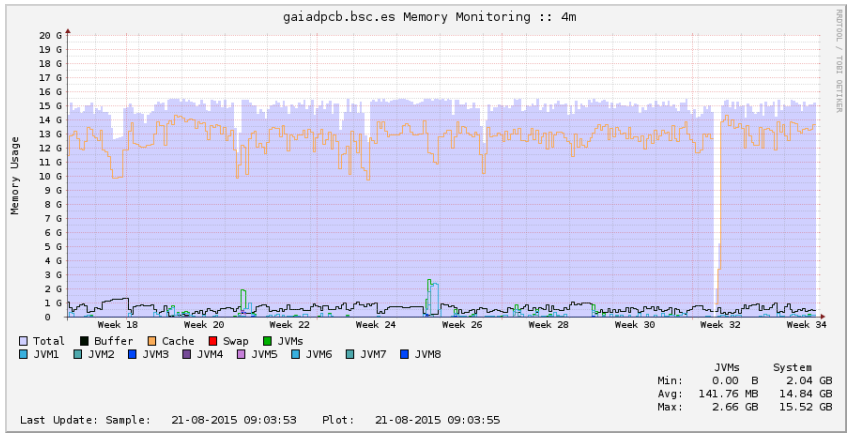


FIGURE 5.15: Memory monitoring plot example

- Job resource usage (see Figure 5.19)

The first two diagnostics are obtained directly from the Operation System (OS) running in each node while the other two are basically obtained from the Marenostrum job queue. All these diagnostics are published in the DPCB Interface Server (described in Appendix A.1.3) so IDU operators can monitor remotely the progress of the IDU task processing from a more friendly interface.

The current operator interface (Figure 5.20) at the time of writing does not implement yet the job management functionalities. The operator has

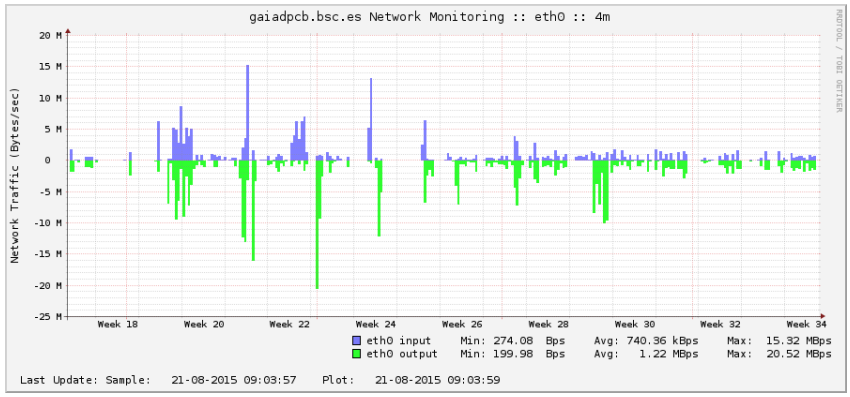


FIGURE 5.16: Network usage monitoring plot example

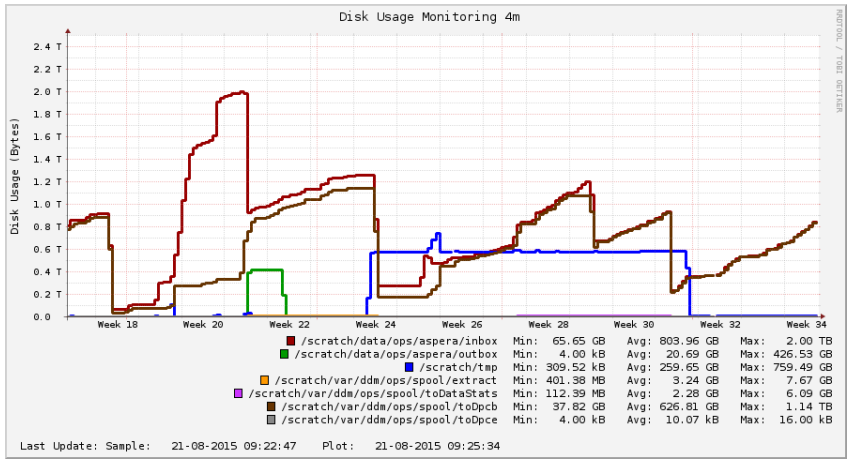
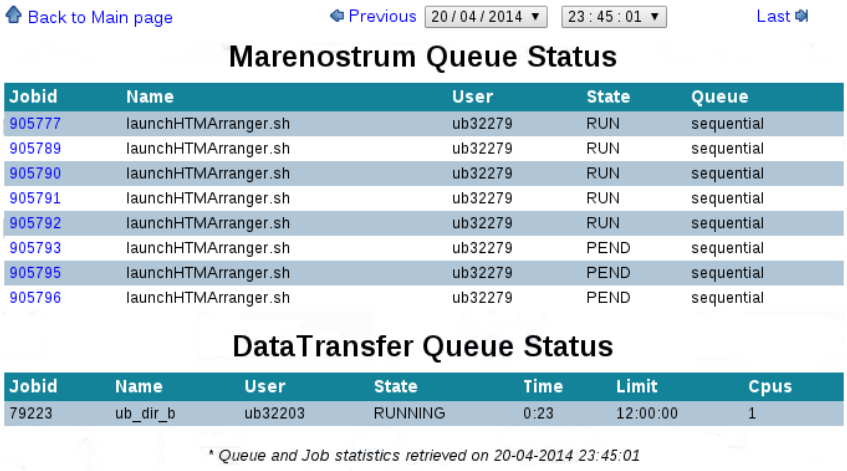


FIGURE 5.17: Disk usage monitoring plot example

to connect to Marenostrum login nodes for interacting with the queue and be able to launch new jobs or cancel the current ones.



Back to Main page Previous 20/04/2014 23:45:01 Last

Marenostrum Queue Status

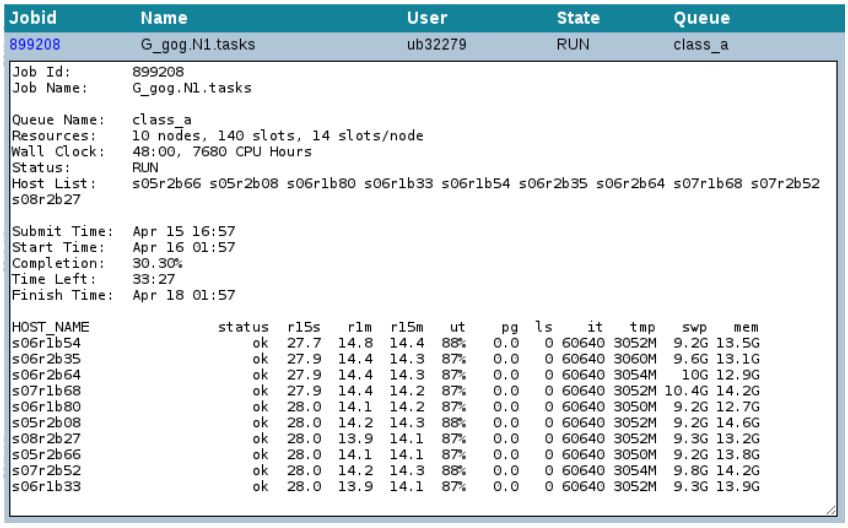
Jobid	Name	User	State	Queue
905777	launchHTMArranger.sh	ub32279	RUN	sequential
905789	launchHTMArranger.sh	ub32279	RUN	sequential
905790	launchHTMArranger.sh	ub32279	RUN	sequential
905791	launchHTMArranger.sh	ub32279	RUN	sequential
905792	launchHTMArranger.sh	ub32279	RUN	sequential
905793	launchHTMArranger.sh	ub32279	PEND	sequential
905795	launchHTMArranger.sh	ub32279	PEND	sequential
905796	launchHTMArranger.sh	ub32279	PEND	sequential

DataTransfer Queue Status

Jobid	Name	User	State	Time	Limit	Cpus
79223	ub_dir_b	ub32203	RUNNING	0:23	12:00:00	1

** Queue and Job statistics retrieved on 20-04-2014 23:45:01*

FIGURE 5.18: Marenostrum queue monitoring, listing all jobs in the queue (including user and queue name) and showing the current job status



Jobid	Name	User	State	Queue
899208	G_gog.N1.tasks	ub32279	RUN	class_a

```
Job Id:       899208
Job Name:     G_gog.N1.tasks

Queue Name:   class_a
Resources:    10 nodes, 140 slots, 14 slots/node
Wall Clock:   48:00, 7680 CPU Hours
Status:       RUN
Host List:    s05r2b66 s05r2b08 s06r1b80 s06r1b33 s06r1b54 s06r2b35 s06r2b64 s07r1b68 s07r2b52
s08r2b27

Submit Time:  Apr 15 16:57
Start Time:   Apr 16 01:57
Completion:   30.30%
Time Left:    33:27
Finish Time:  Apr 18 01:57

HOST_NAME               status  r15s   r1m  r15m   ut    pg  ls    it   tmp   swp   mem
s06r1b54                    ok  27.7  14.8  14.4  88%   0.0   0 60640 3052M  9.2G 13.5G
s06r2b35                    ok  27.9  14.4  14.3  87%   0.0   0 60640 3060M  9.6G 13.1G
s06r2b64                    ok  27.9  14.4  14.3  87%   0.0   0 60640 3054M   10G 12.9G
s07r1b68                    ok  27.9  14.4  14.2  87%   0.0   0 60640 3052M 10.4G 14.2G
s06r1b80                    ok  28.0  14.1  14.2  87%   0.0   0 60640 3050M  9.2G 12.7G
s05r2b08                    ok  28.0  14.2  14.3  88%   0.0   0 60640 3052M  9.2G 14.6G
s08r2b27                    ok  28.0  13.9  14.1  87%   0.0   0 60640 3052M  9.3G 13.2G
s05r2b66                    ok  28.0  14.1  14.1  87%   0.0   0 60640 3050M  9.2G 13.8G
s07r2b52                    ok  28.0  14.2  14.3  88%   0.0   0 60640 3054M  9.8G 14.2G
s06r1b33                    ok  28.0  13.9  14.1  87%   0.0   0 60640 3052M  9.3G 13.9G
```

FIGURE 5.19: Detailed Marenostrum job information, including resources requested, job times and the status of each individual node assigned to the job

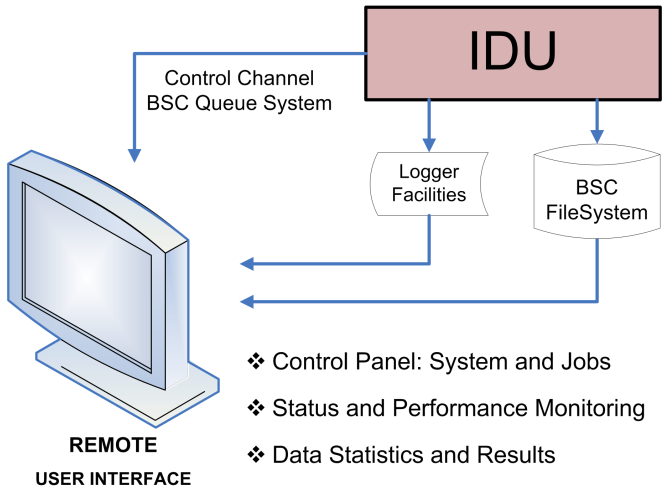


FIGURE 5.20: Schematic diagram of the user interface implemented for IDU monitoring and management

5.4 Conclusions

This chapter is the second and closing part focused on the work done for the design and implementation of an efficient IDU processing system.

We have described the IDU operation strategy in high detail, including all the phases; release definition, testing and execution. This description has also included the detailed timing with respect to the DS and DRC definitions and dependencies with other DPAC systems. This timing has been discussed several times within DPAC and our work has been essential to reach the current definition. The presented timing should provide robust solutions as well as introduce fast improvements in the results of the processing iterations.

Along this chapter, we have shown how the design has not been only driven by the scientific requirements but also by the characteristics and restrictions of the execution environment and resources – Marenostrum supercomputer. The DPCB environment and the huge amount of data to process leads us to the implementation of a batch processing system, where a lot of jobs are executed on the computing resources without external communication or intervention. However, the implementation of this batch processing system is not so straightforward and trivial. Jobs needs to be prepared and equalised properly and the resulting jobs must be distributed and executed among the supercomputer processing nodes efficiently. We have been working closely with BSC team and performing a huge amount of tests in Marenostrum to be able to understand and characterise the full potential and limitations of the DPCB environment. This knowledge has helped us a lot to design an efficient IDU but also has helped to improve and add new functionalities to some of the BSC services and applications. As a result of this huge effort, we have designed and developed a processing framework which should be capable of withstanding the high processing demand of future IDU tasks.

For any batch processing, knowing in very high detail the processing requirements and the performance profiles of the tasks is fundamental. This

information has been collected initially from the IDU contributors but also from the exhaustive tests carried out and the tools developed during this thesis. These tools have not only provided the task performance profiles but also have helped in the detection and fix of several tasks issues. Additionally, we have implemented tools for the equalisation of the data and the jobs; for its storing and processing respectively. This equalisation is what allows us to define the best job flow and job distribution to take full advantage of the Marenostrum computational resources.

For IDU, an efficient data access is also essential. We have worked on a tailored file format based on HDF5 and also in the development on an integral data access layer to provide all the necessary functionalities to the final IDU tasks regardless the physical storage mechanism and file format. This data access layer has been implemented to be portable and usable in other environments so the developers are not forced to use DPCB resources for their activities.

On the other hand, we have also discussed why the advanced features of the Java language have been employed to ease the development of the software, making this system work well and remain very portable. Additionally, the DPAC guidelines have contributed positively to the achievement of a well documented and easy to maintain IDU software.

Although, further work for the coming years has been already identified, the implemented system is already capable of satisfying the performance demands for the execution of IDU in the first DRCs.

6 IDU Early Execution

The first execution of an IDU task was initially scheduled after the first two DSs, DS-00 and DS-01 (Mercier and Hoar [2013] and Castañeda et al. [2013]). These two DSs would include all the accumulated data since the start of nominal/routine operation of Gaia on 25th of July 2014 until approximately end of September 2015. Unfortunately, the daily pipeline at DPCE accumulated some inconsistencies and produced poor cross match results during this period, namely:

- Double detections on board. Identified end commissioning in June 2014 as already mentioned in Section 4.2. These detections caused the duplication of some catalogue sources during the first four months of the DS-00. The proper treatment of these cases was implemented and activated in IDT around November 2014.
- Diffraction spikes of bright stars causing spurious on-board detections (see also Section 4.2). Identified in August 2014 and mitigated at IDT level from November 2014 onwards.
- Occasional excursions/spikes of the refined attitude.
- Inconsistencies introduced due to the clean ups performed over the IDT new source table used in the DPCE pipeline. These clean ups where requested by FL because some of its diagnostics were not able to handle properly the large ambiguity present in the IDT cross match results.

- Unmatched sky areas of mostly triangular shapes caused by an issue in IDT software when accessing the catalogue sources. Fixed in IDT release end of September 2014.

The reprocessing of non–raw IDT products is not a nominal DPAC activity but the need of a consistent cross match data and a clean catalogue was considered essential for all downstream systems, mainly to assure a consistent input data for the AGIS and PhotPipe systems. Therefore, this activity was discussed and was finally approved in the DPACE #28 meeting on 28th of October 2014 and assigned to DPCE. It should have been completed during November 2014.

This reprocessing activity at DPCE however was highly delayed, as of end of March 2015:

- DPCE Hardware/Software performance issues during last two months increased the data backlog and thus the processing load of the daily pipeline.
- Increase of the processing load at DPCE due to the data volume increase from the first galactic plane scan from end February and beginning of March 2015.
- Priority clash with the upgrade activities of the software on board Gaia in terms of manpower and hardware resources at DPCE.

At that time, the estimations indicated that the cross match reprocessing completion at DPCE before end of the first DS (around first of July 2015) was not feasible and estimations would point to mid August 2015. On top of that, DPCI needed this data well before the DS end – approximately one month before – to be able to restart the PhotPipe processing on time and thus deliver its results according to the original schedule. This of course added even more pressure to the already tight schedule.

In view of this situation, during the Operations Workshop #03 (Centre National d'Études Spatiales (CNES), 25-27 of March 2015) the possibility

to move to DPCB this reprocessing activity was discussed between DPCB and DPCE. This discussion was then extended in a dedicated teleconference on 30th of March 2015 with the CU3 Leader and Technical Manager, CU3-IDU/DPCB Manager and DPCB QA Manager. After this teleconference, we prepared the technical note [Castañeda et al., 2015] with the assessment on the possibility of the execution of this reprocessing at DPCB before the end of DS-00 using IDU . This note includes not only a detailed execution proposal but also the impact that this activity would have for the DPCB and IDU manpower and hardware resources. This proposal was then presented to all the DPCs and was finally approved on 10th of April.

In this chapter, we firstly describe the main terms of the original execution proposal and summarise how it was finally carried out. Then, we characterise the input data of the DS-00 including a summary of the main issues affecting the data. Afterwards, we present the scientific and computational performances of the executed IDU tasks at DPCB. Finally, we summarize the main findings and conclusions reached thanks to this first execution of these IDU tasks over real Gaia data.

6.1 Execution Plan

IDU tasks run nominally just after a given DS is closed and DPCB has received all its data (see Figure 5.3). The first tasks (IDU-DC, IDU-SCN and IDU-XM) are then executed in one single run over the full data set extent requiring a period of approximately three weeks to complete. Adopting this execution plan would have implied that the reprocessed data would only be available not sooner than approximately four weeks after the DS end – after the full data was processed at DPCB and delivered first to DPCE and then distributed to the remaining DPCs. This delay would have ruined completely the operations schedule of the other processing systems and for this reason the following alternative execution plan was proposed as of beginning of April 2015:

1. First bulk execution of all observations already received at DPCB until mid March 2015, just covering the first eight months of Gaia routine operations. This first stage would process 85% of the expected data entering the DS-00 and the processing and distribution of the data could then be completed before end of April 2015.

2. Second execution covering all the newly accumulated observations up to mid May 2015. The results would be obtained and distributed downstream before end of DS-00.

3. Last execution including the remaining data up to the DS-00 end beginning June 2015, approximately two additional weeks of data.

As it can be seen, the proposed plan was to split the processing in three stages. In each execution stage the amount of data to be processed decreases considerably thus reducing also the time required to complete the processing and to distribute the final data after the DS-00 end. Additionally, following this procedure, DPCB was able to provide progressively most of the reprocessed data downstream before the end of the DS-00. This situation was beneficial for the other DPCs since they could start preparing the new data for their imminent processing activities avoiding undesired delays on the schedule. This execution plan was presented and discussed with all the DPCs and it was considered the best solution taking into account the tight schedule.

For the accomplishment of this reprocessing using IDU, the following tasks were executed at DPCB:

- Data arrangements over the received data, including the computation of the time and spatial statistics for the job and task processing definition. This data includes the attitude (from IDT and FL), FL geometric calibration, catalogue sources and the raw observations. None of the IDT new sources produced by the daily pipeline at DPCE were used so a completely new, independent and consistent solution would be produced.

- IDU-DC: for the redetermination of the spurious detections to be excluded during the Cross-Match. The execution of the IDU-SCN was not necessary because the IDU-DC implementation at that time was not yet capable of treating the IDU-SCN outputs.
- IDU-XM: using a similar algorithm than IDT featuring:
  - Spurious detection filtering.
  - Double detection handling.
  - New source creation only from unmatched observations or duplicated matches resolution. Source merging/splitting functionality was still not functional at that time.

Additionally, to avoid DM interface issues downstream, IDU software was updated to produce the same DM interfaces than IDT. As a result, the other DPCs would not need to do any change on their systems – still seeing the data as being generated by the daily DPCE pipeline. This was also fundamental to keep the DRC tight schedule. These data however was tagged with a specific solutionId so its origin could be clearly identified. The only caveat was that since no photometric data is available at DPCB, the new sources created in this reprocessing could not have any colour information.

It is worth pointing out that while the reprocessing at DPCB was done, IDT continued producing data which would eventually clash with the new reprocessed data. For this reason, the devised plan also included the progressive retraction of the old data results together with the delivery of the new DPCB reprocessed data.

At the end this progressive delivery of the results was not done and the DPCs received first the information for the full retraction of the previous IDT results and afterwards the new data reprocessed at DPCB. This deviation of the original plan was possible because the initial constraint for providing the reprocessed data before the DS end was dropped – basically PhotPipe run finally was deferred to the next DS – relaxing in this way the requirements on the processing activities at DPCB.

To accomplish the final goal of cleaning up the inconsistencies in IDT products, an additional corrective operation on the DPCE pipeline was necessary. Basically, before restarting the daily pipeline for DS-01, the working source catalogue in IDT was updated using the new sources regenerated at DPCB. This operation implied then the removal of $\sim$ 2.5 billion new sources created by IDT during the DS-00 followed by the ingestion of the new sources from DPCB, almost 2 billion sources as reported in Section 6.3.5. With this last corrective operation, IDT could be started again avoiding further inconsistencies in the new cross match data.

This execution plan for the reprocessing was clearly very tight and demanding for DPCB and IDU teams at the Universitat de Barcelona (UB) and the following consequences were stated and accepted by DPACE:

- All the activities would be executed on best-effort basis and that minor deviations on input data and algorithm configurations were envisaged depending of the results obtained during the reprocessing activities.
- DPCB would be entering operations four months in advance with respect to what was indicated in its Operations Plan [Castañeda et al., 2013].
- The rest of DPCB and IDU preparatory activities for operations would be delayed by at least one month.
- Non-negligible additional BSC resources would be required and accounted for.

The reprocessing at DPCB was completed beginning of June 2015. However, after the completion of this first operational activity, we processed again the full DS-00 following the nominal procedure – running the IDU tasks in a single run.

This additional reprocessing was done to get a more reliable reference solution removing any unnecessary effects introduced by the data split. The

data split might affect the final quality and consistency of the results mainly because we are solving some cases without all the available observations. The data split and consequently data incompleteness may:

- leave some spurious detections undetected leading to the creation of unnecessary new sources.
- fail in creating new sources for actual resolved objects having separate observations but treated in different processing stages.

In addition, few issues were detected in the IDU software during the execution. These issues were progressively fixed including also some minor improvements in the model parameters of the IDU-DC. As a result of these updates some minor deviations are observable on the results of each processing stage.

With this second run, we have consequently removed all these undesired effects providing an improved solution more appropriate for analysis in this thesis. Hence, we have preferred to present the scientific and computational performances of the second run of the tasks at DPCB because it provides a better and more compact overview of the obtained results. However, we have also included the most relevant details of the original results obtained in the first execution for each category in a separate section.

6.2 Data Segment 00

The DS-00 covers the first ten months of data after the commissioning phase when Gaia started routine operations. The DSs are measured in OBMT coordinates which basically are derived from the actual reading in units of 50 nanoseconds of the master clock controlling all Gaia operations [García-Berro et al., 2006]. More specifically the time range of the DS-00 is:

- **OBMT Range:** 23 292 998 211 919 880 to 44 001 099 999 999 999 [nanosec]

- **OBMT Length:** 6285.48 [hour] = 305.40 [day] = 10.18 [month]

Table 6.1 shows a short summary of the relevant incoming data volume and counts of this first data segment. The volume of data to be processed was approximately 7 Terabytes mainly coming from the 22 billion observations received. Figure 6.1 and Figure 6.2 show the observation time density with respect to the OBMT time – the density of the confirmed detected objects and the successfully received observations on ground respectively. In these figures it can be identified the increase of observations when scanning the galactic plane. During these periods Gaia is acquiring more than 600 thousand observations per minute, which approximately translates to an observation density of $\sim$ 400 thousand observations per square degree taking into account the focal plane AC size ($\sim$0.7 degrees) and the scan rate (of 60 arc second per second). Figure 6.3 on the other hand shows the magnitude distribution of the objects detected by Gaia. In this figure a bulge around 12^{th} magnitude and a peak at 13^{th} magnitude are evident. This behaviour probably comes from the detections of saturated sources and the on board detection algorithm limitations. Also the detections of bright cosmic rays might be contributing to this behaviour.

Data type	**Record count**	**Disk Size**
IGSL Sources	1 222 598 530	122.00 GBytes
Object Logs (ASD7)	24 681 840 626	239.50 GBytes
Observations (SP1)	22 097 513 352	6,489.34 GBytes
IDT Attitude	15 014	0.07 GBytes

TABLE 6.1: Incoming data summary of the Data Segment 00

While preparing the input data several issues were identified:

- A $\sim$3,52% of the DS-00 did not have attitude information or its quality was too bad for its usage in the processing. The resulting gaps are shown in Figure 6.4 for which no Cross-Match results could be computed.
- A $\sim$4.8% of the attitude corresponding to the beginning of September 2014 was wrongly computed in the DPCE daily pipeline using

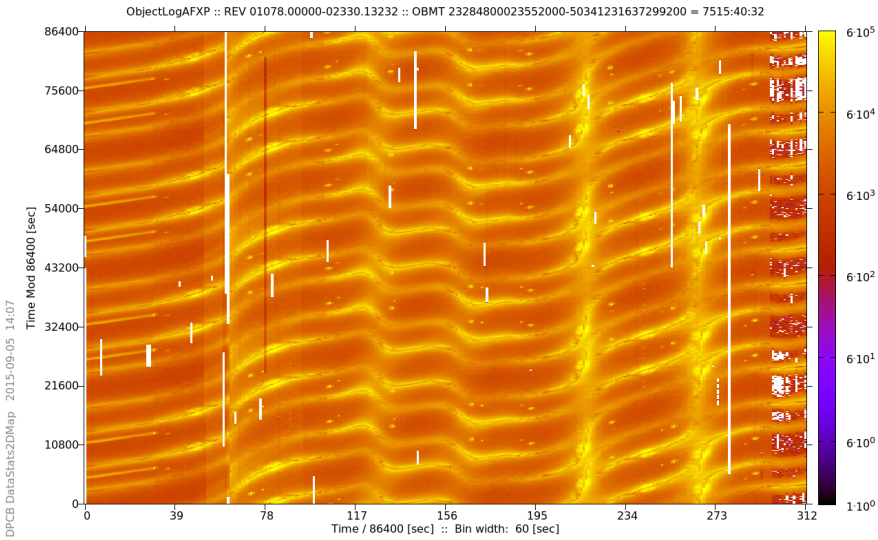


FIGURE 6.1: Data Segment 00 object log density with respect the on board time (time increasing from bottom to top and left to right). Regions in white corresponds to empty time intervals (data gaps) and on the right it can be identified the time intervals with missing data due to the processing issues on the daily processing systems

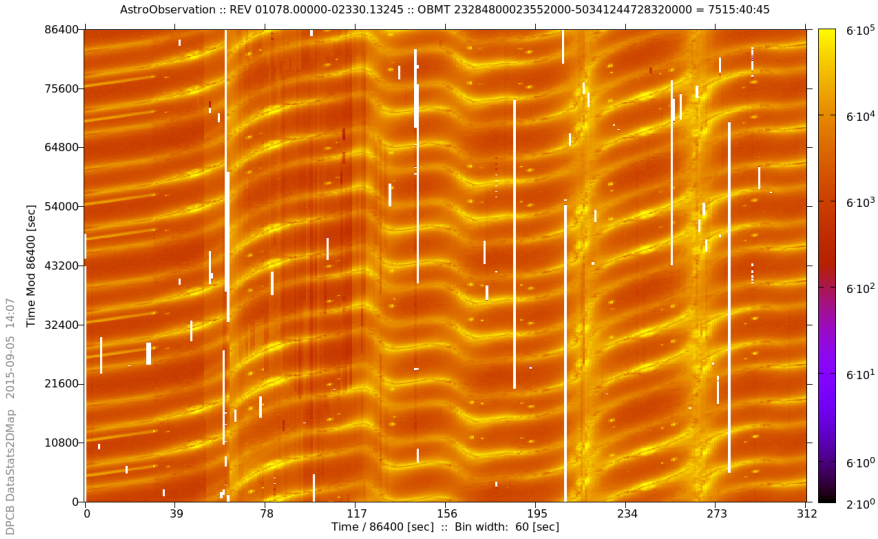


FIGURE 6.2: Data Segment 00 acquired observation density. In this plot it can be clearly identified the set of four consecutive scans (of both FoVs) of the galactic plane done around days +215 and +270 (on abscissa axis) from the start of the Data Segment

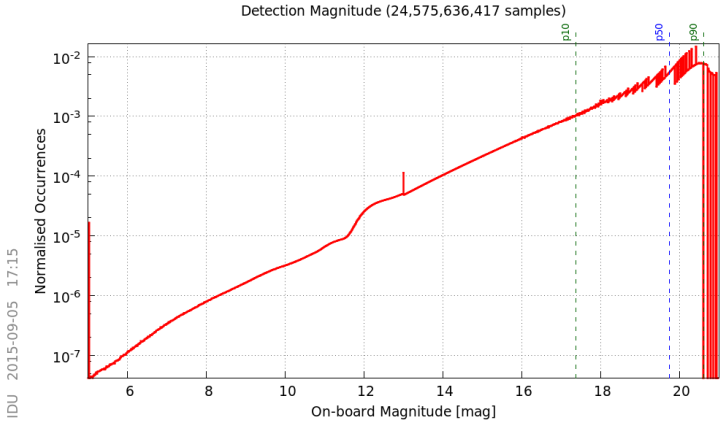


FIGURE 6.3: Magnitude distribution in logarithmic scale of the detected objects on board showing how the 90% of the observations are fainter than 17^{th} magnitude. The bin peak at 13^{th} magnitude is caused by the magnitude coding of saturated images while the ones at the faint end are due to the rounding limitation of the magnitude coding on board

the nominal geometric calibration. This issue has a major impact on the quality of the IDU-XM results since it implies attitude errors (i.e excursions) of up to 800 milliarcseconds. These errors lead to observations being misplaced which consequently may imply the creation of unnecessary new sources polluting the resulting catalogue.

- A ∼15% of the input observations were affected by several IDT processing issues. Fortunately these issues did not affect any of the fields used by IDU-XM (the reference acquisition time and the magnitude field) and therefore this data was considered usable and was not filtered out. There is a period of approximately one month of data from December 2014 where some of these IDT issues affected the attitude. For this period the attitude has lower quality in the AC direction however this was also considered not a major issue for the cross match reprocessing.

- A ∼1,12% of object logs were missing in the DS interval tail due to configuration errors on board and DPCE pipeline data handling issues. These gaps can be clearly identified in Figure 6.1.

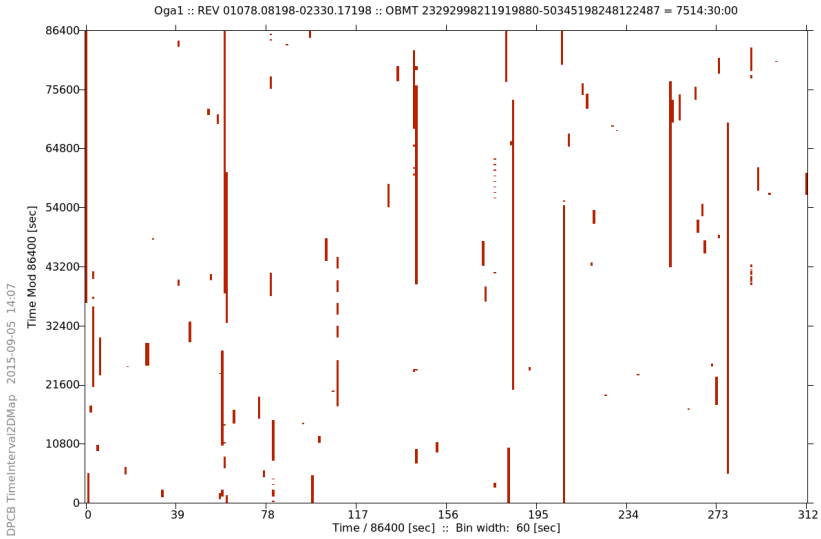


FIGURE 6.4: Attitude gaps for Data Segment 00 (in dark red) obtained after filtering invalid/unusable data

All these issues in the input data could have been easily solved by means of data reprocessing and data recovery activities at DPCE but the tight schedule did not allow to include these additional processing dependencies. These issues have been already scheduled at DPCE and in principle should be all resolved for the next operational execution in October 2015.

Finally, it is worth pointing out that during the DS-00 the IDT new source table was cleaned several times at DPCE. Consequently IDT may have created again the same new sources when scanning again some sky regions. This basically implies that the new sources created by IDT were inconsistent at the time of the start of this activity which has prevented any kind of scientific results comparison and/or regression checks against the IDT data.

6.3 Scientific Performance

This section provides the overall scientific assessment over the configuration, the processing steps and the data results obtained running the IDU

tasks involved in this reprocessing activity. For the execution of these tasks, the following scientifically relevant configuration was used:

- IDU-DC was configured to identify spurious detections from observations up to the 16^{th} magnitude. This value was increased by one magnitude with respect to the IDT configuration. This update was chosen according to the results obtained in the daily pipeline which indicated that spurious detections were also present around the extended magnitude range.
- The radius used for matching the observation to the IGSL sources was fixed to 1.5 arcsecconds. This value was chosen according to the IGSL errors and the results obtained in IDT; in both the attitude and the Cross-Match.
- For the unmatched observations, a more restrictive radius was used, of 1.0 arcsecond. This radius was reduced to be more in agreement with the performance of the Gaia detection errors.

In next sections, we present the more relevant scientific diagnostics and findings encountered in each one of the executed tasks; starting with the IDU-DC and finishing with the results of the IDU-XM processing stages.

6.3.1 Detection Classifier

This section describes the scientific results obtained running the IDU-DC task. A total amount of 3 204 227 611 spurious detections were identified, $\sim 14.50\%$ of total amount of input observations. This percentage is similar to the one obtained in IDT as expected since both systems used the same core algorithm with minor improvements in the model parameters.

Figure 6.5 shows the time distribution of the identified spurious detections whereas the Figure 6.6 shows its distribution on the sky.

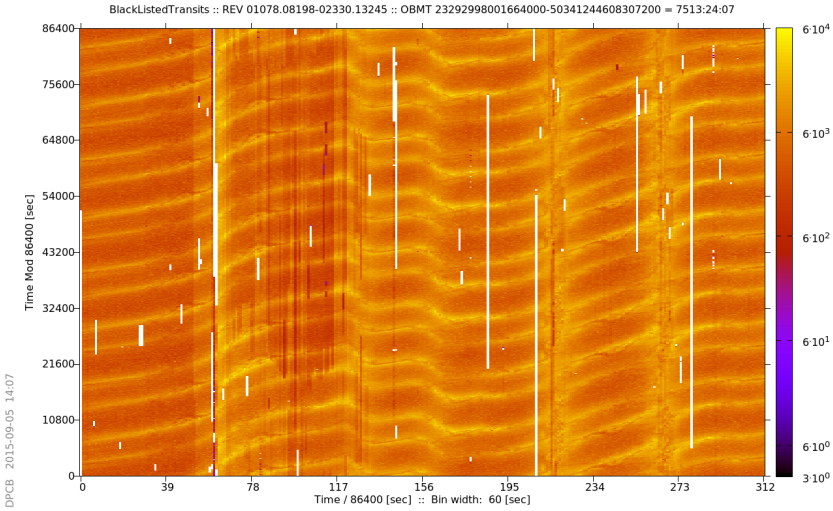


FIGURE 6.5: Spurious detections density with respect the on board time identified in the Data Segment 00. The amount of spurious is proportional to the real observation density (about 15%) as expected

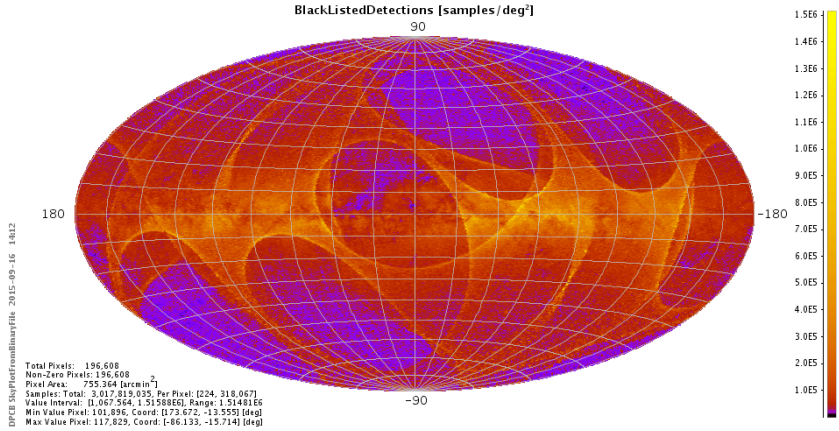


FIGURE 6.6: Sky distribution of the spurious detections. Note that not all the detections could be included due to missing attitude for some periods

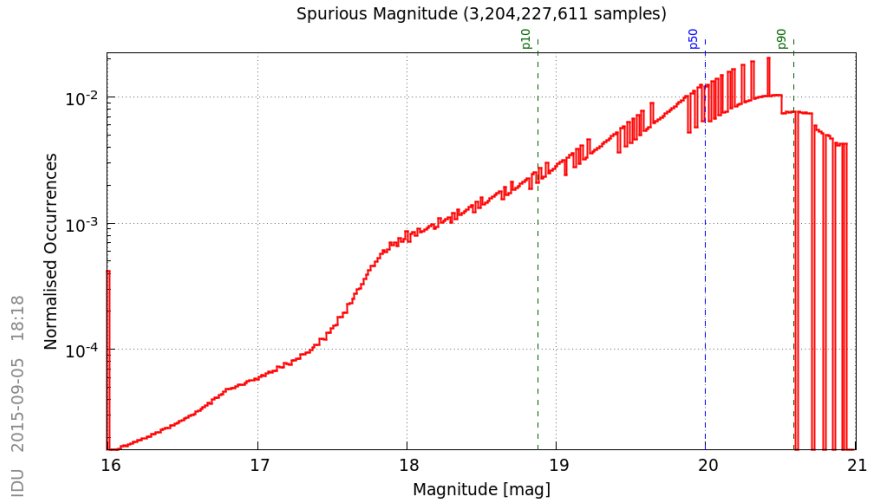


FIGURE 6.7: Magnitude distribution of the identified spurious detections in the Data Segment 00 where ∼90% of the detections are fainter than 19^{th} magnitude

Figure 6.7 shows the magnitude distribution of the spurious detections. The figure shows that the majority of detections have magnitudes fainter than 19^{th} magnitude as expected.

Figure 6.8 shows the distribution of the distances between the spurious detections and the corresponding parent object position (in both AL and AC directions) as a function of the parent magnitude. The AL distance tends to increase for brighter parents in the following tail mainly because of the CTI. This figure also shows a symmetric distribution for AC distance which means the spurious are equally distributed around the parent in the AC direction. These results are consistent with the cases described in Bestard [2015].

As already commented, the current algorithm is under heavy development, therefore some spurious detections are still not identified correctly. Only further executions and testing will clarify if the obtained distribution of spurious detections are really representing the actual response of the instrument. Figure 6.9 shows the comparison of a sky region before and after cleaning the detections classified as spurious. In this figure we can see how

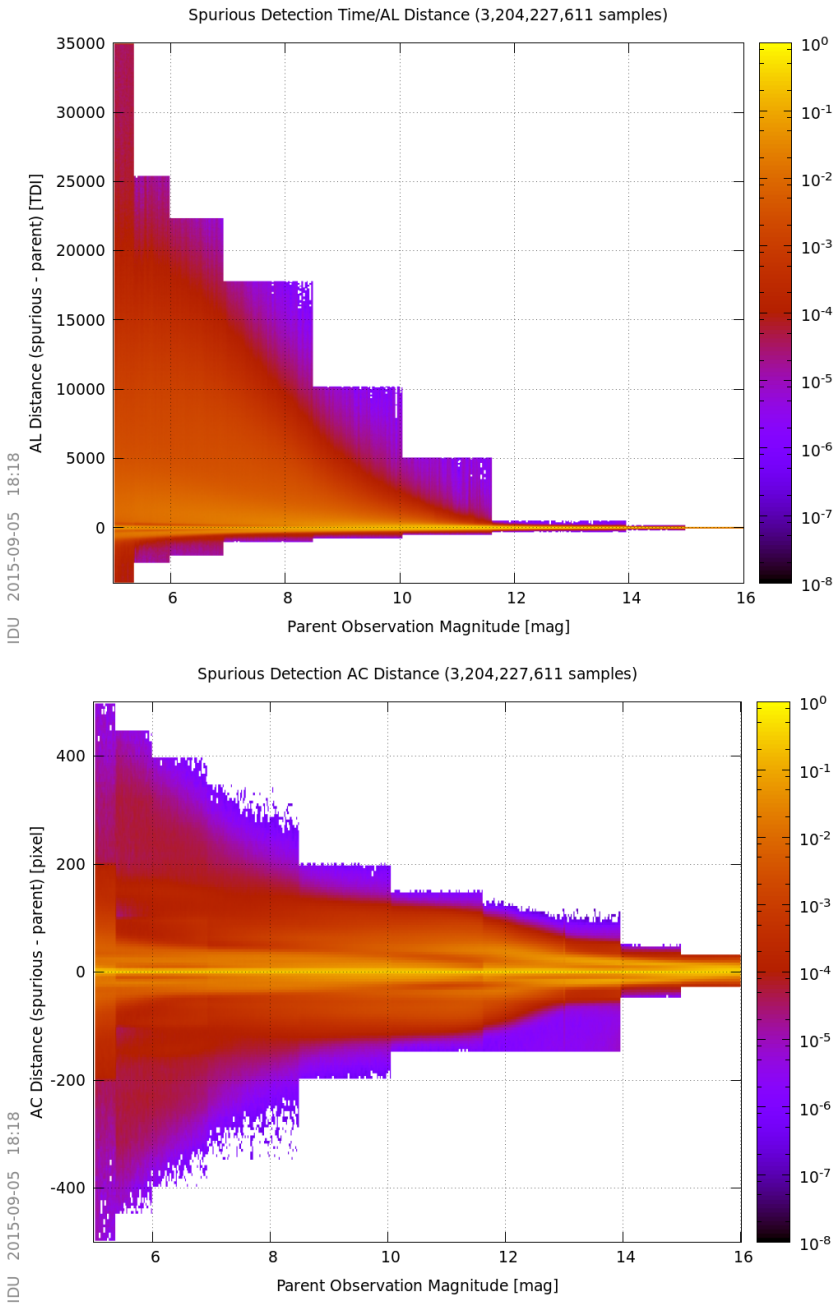


FIGURE 6.8: Distribution of the AL and AC distance (top and bottom panels) with respect to the brightness of the parent object magnitude. On top, the positive distances correspond to spurious detections identified after the parent object and it covers larger distances as expected due to the CTI effects. On the bottom panel, the AC direction, this dependency is not present. The stepped profile is due to the current magnitude bin parametrisation

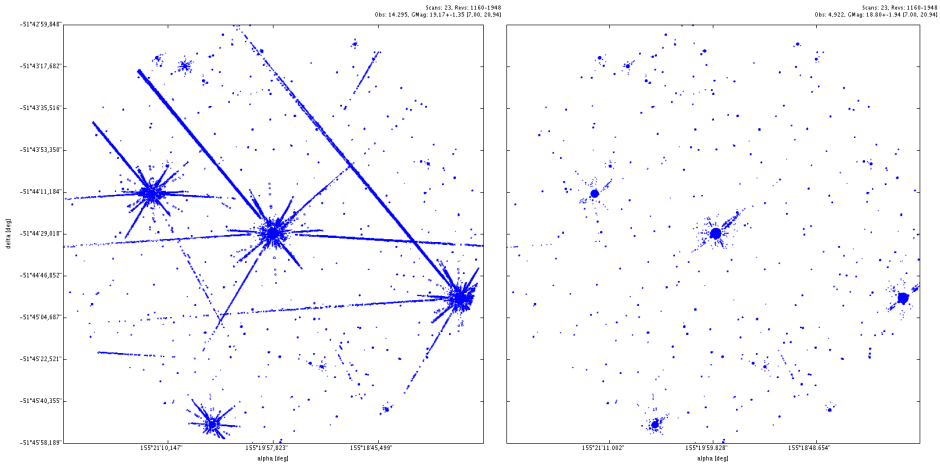


FIGURE 6.9: Example of the spurious detection performance. The left figure represent the original input observations whereas the right figure presents the same region after the clean up of more than 10 thousand detections successfully classified as spurious. Unfortunately some detections still remain which ultimately will pollute the final cross match

most of the spurious structures are successfully removed but unfortunately some detections still remain nearby the brighter sources.

During the task execution, we also identified several not foreseen cases. These new cases are being analysed and will probably imply the development of additional models and algorithms for their proper handling within the IDU-DC task. Below, a brief summary of these cases are presented.

Uncontrolled Spurious Detections

Figures 6.10, 6.11 and 6.12 show some examples of cases where the current implementation fails to identify the spurious detections for regular objects. These failures mainly come from:

- The simplistic model being used described in Section 4.2. This model is clearly not appropriate for the treatment of the spurious detections coming from very bright objects as well as for the fainter ones.

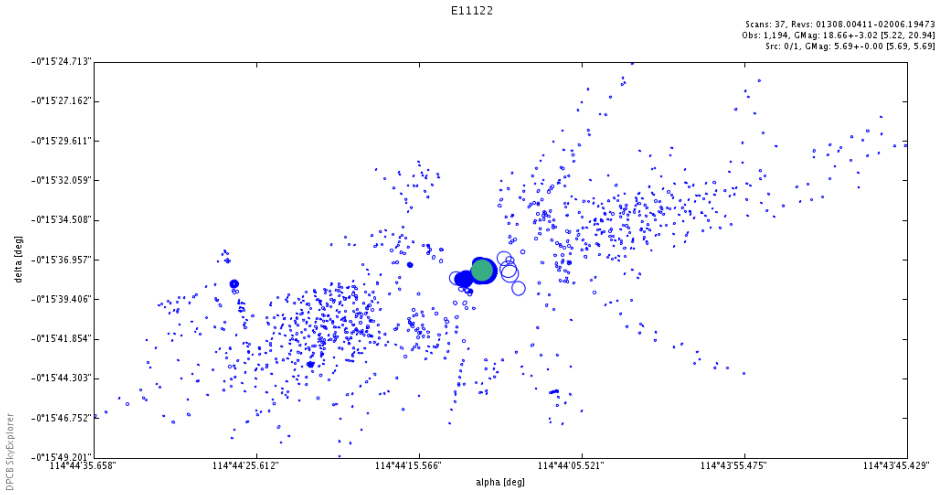


FIGURE 6.10: Remaining detections around Hipparcos source HIP37243 with magnitude ~ 5.6. The source is plotted in green whereas the detections are plotted in filled blue dots if they have at least one source candidate or empty dots otherwise. The dot size is for both cases proportional to the brightness of the object. Unmatched observations nearby the source (~ 2 arcseconds) with similar brightness can be identified

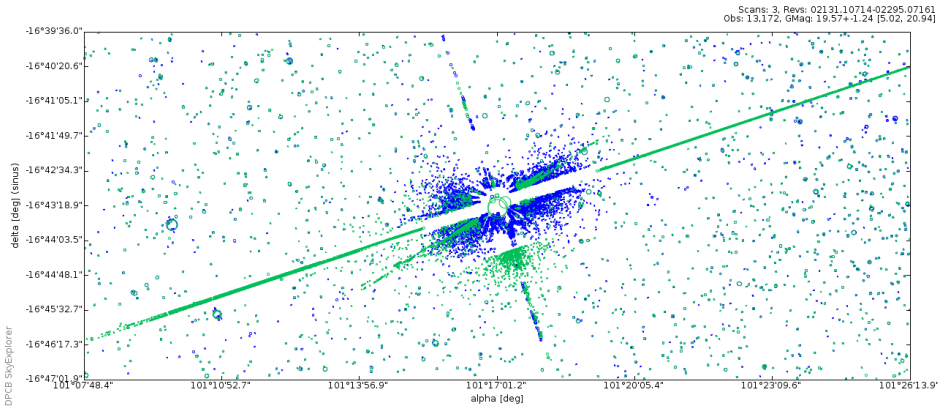


FIGURE 6.11: Remaining spurious detections from two scans of Sirius. In the blue scan the source fell in between two CCD rows

- The need of the parent observation in the algorithm inputs for the triggering of the clean up process. Requirement which will be fulfilled in the next executions at DPCB by means of the IDU-SCN outputs.

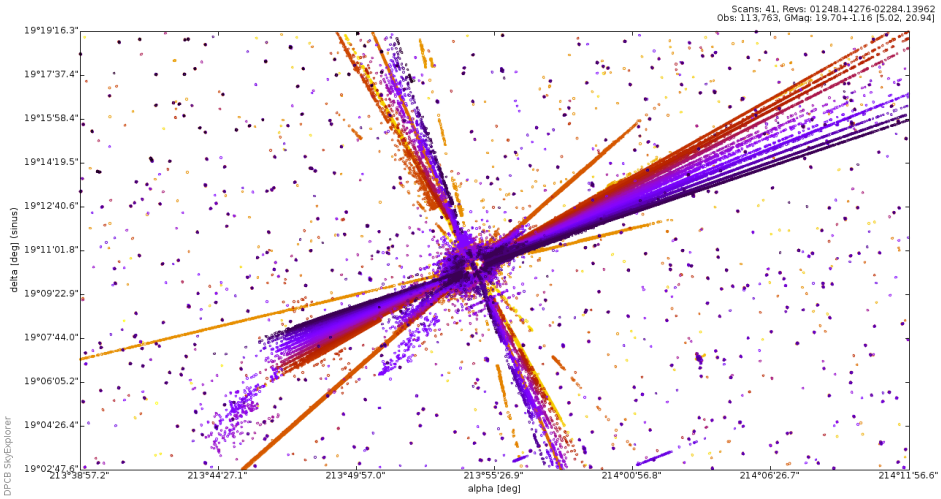


FIGURE 6.12: Remaining spurious detections from the 41 scans (colored in gradient) of Arcturus done during the DS-00. The larger spike has an angular length of more than 25 arcminutes with respect to the source location. In the plot more than 110 thousand spurious detections are shown with an average magnitude of 19.7 ±1.16

SSO Spurious Detections

Other undesired spurious detection sources, not treated by the executed *DetectionClassifier* algorithm, are the SSOs transits. The major planets create a huge number of spurious detections following completely different profiles than the regular objects as shown in Figure 6.13 corresponding to a single Jupiter transit in end April 2015. Besides, these spurious detections are not created at fixed sky positions but depend on the position of the SSO at the scan time thus polluting different sky regions each time as shown in Figure 6.14. An overwhelming situation appears for some Venus transits (see Figure 4.8) where the full focal plane and both FoVs are highly contaminated by spurious detections.

Extended Objects

A lot of extended objects have been identified during this reprocessing activity, including:

- Galaxies, as the ones shown in Figures 6.15, 6.16 and 6.17

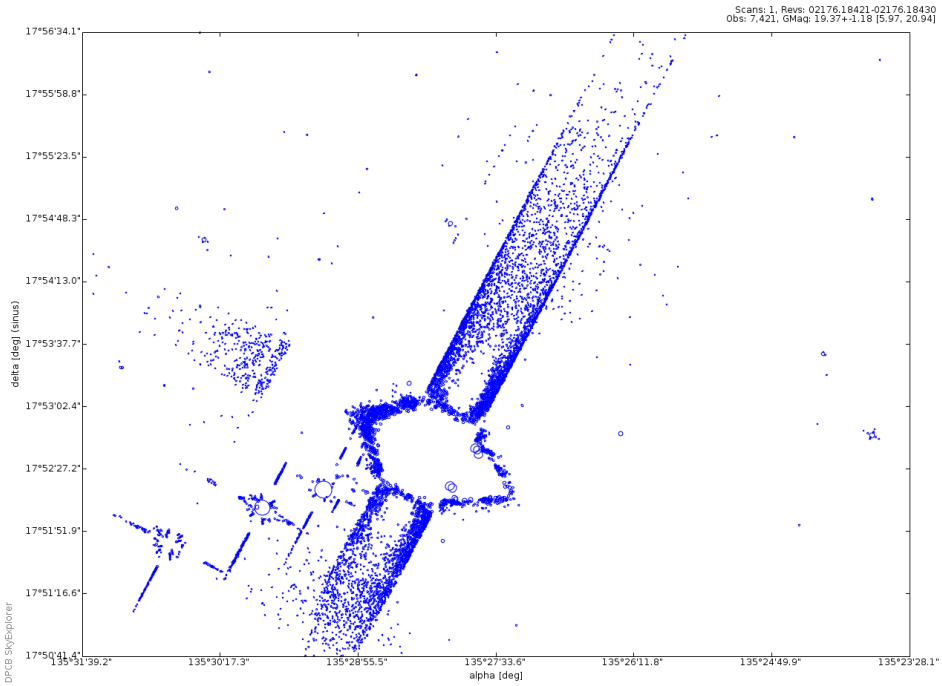


FIGURE 6.13: Spurious detections around a Jupiter transit. The plot corresponds to a single scan of the planet and includes more than 7 thousand observations. Also the spurious detections around four of its satellites can be identified

- Planetary Nebulas, as in Figures 6.18, 6.19, 6.20 and 6.21

For both cases, Gaia is producing spurious detections over the brightness filamentary structures of the ionized gas. It must be noted that each scan of these extended objects produces a huge amount of spurious detections which consequently lead to the creation of a considerable number of new sources. Eventually, the detections from diffuse objects should be also filtered out but no suitable algorithm has been yet developed although its treatment can be easily triggered through to the IDU-SCN predictions. Fortunately, these objects are always located in the same locations and therefore do not pollute other sky regions as for the SSOs transit case.

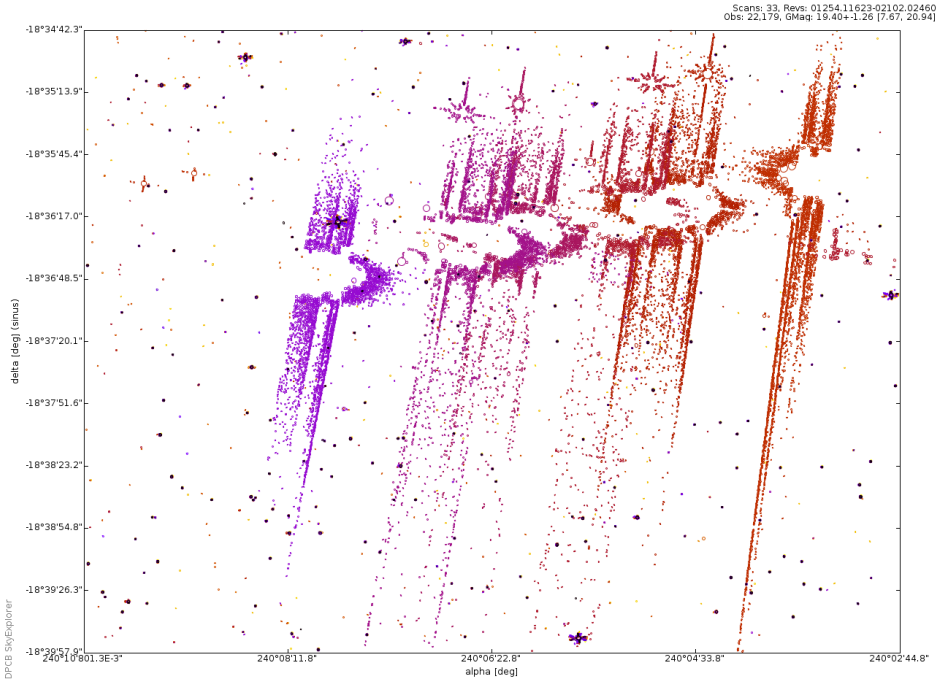


FIGURE 6.14: Spurious detections from several consecutive Saturn transits. The plot shows more than 22 thousand observations and how the planet transit is polluting different sky regions

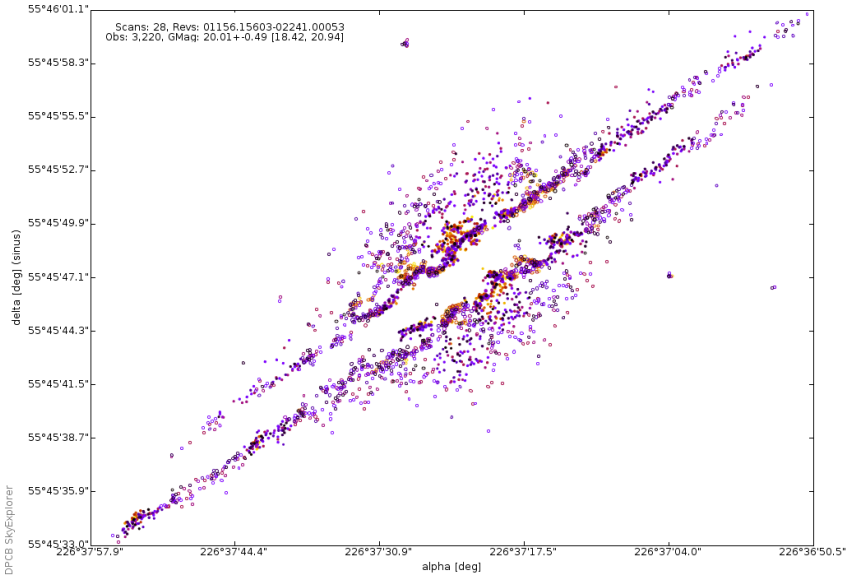


FIGURE 6.15: NGC 5866 (also called the Spindle Galaxy or Messier 102) characteristic because of its thin disk where almost no detections are produced due to its dust

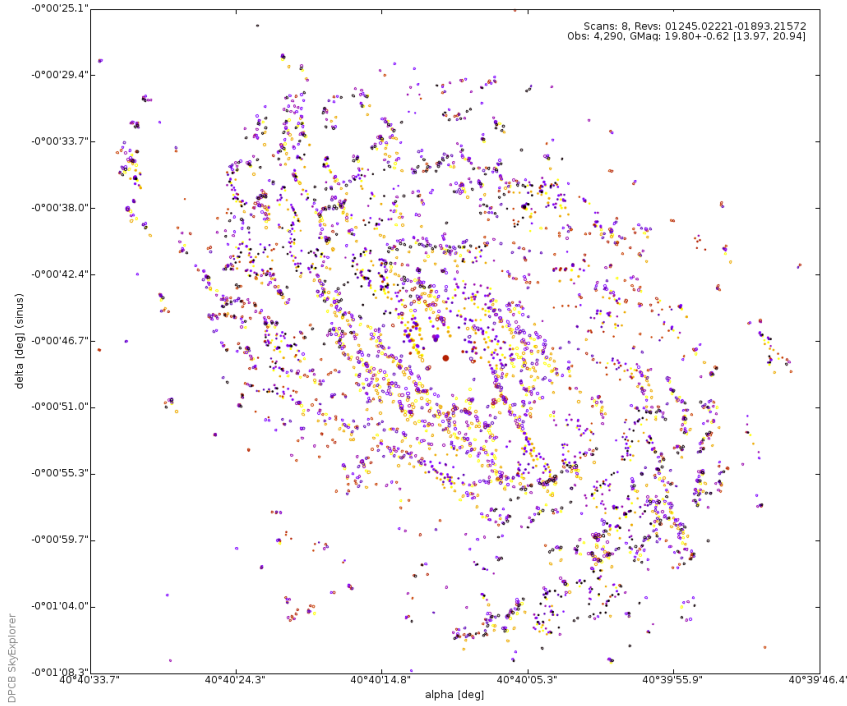


FIGURE 6.16: Messier 77 (also known as NGC 1068) is a barred spiral galaxy

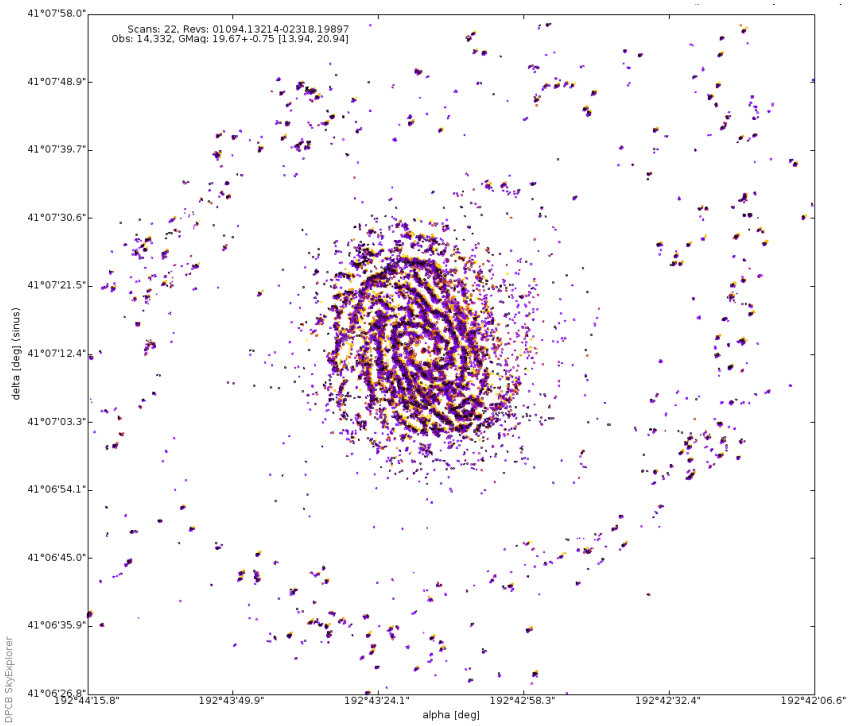


FIGURE 6.17: Messier 94 (also known as NGC 4736) spiral galaxy. The two characteristics ring structures can be identified

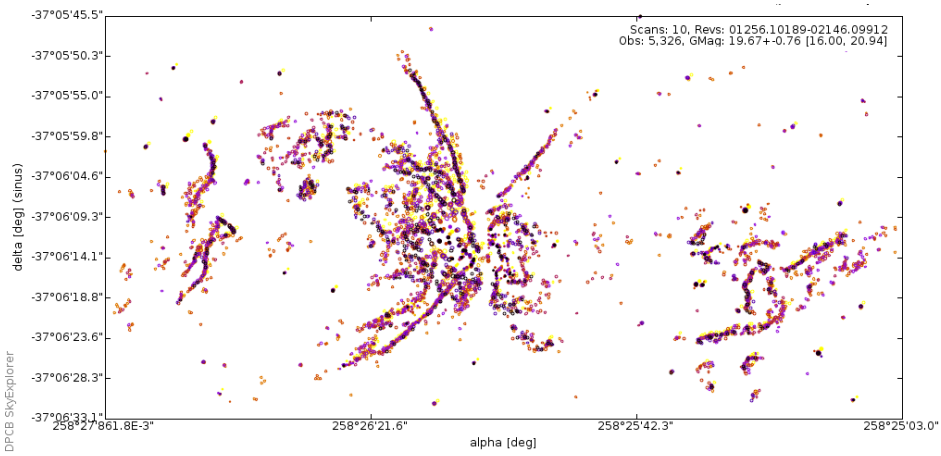


FIGURE 6.18: NGC 6302, also called the Bug Nebula, Butterfly Nebula, is a bipolar planetary nebula. The central star, a white dwarf, is never observed by Gaia

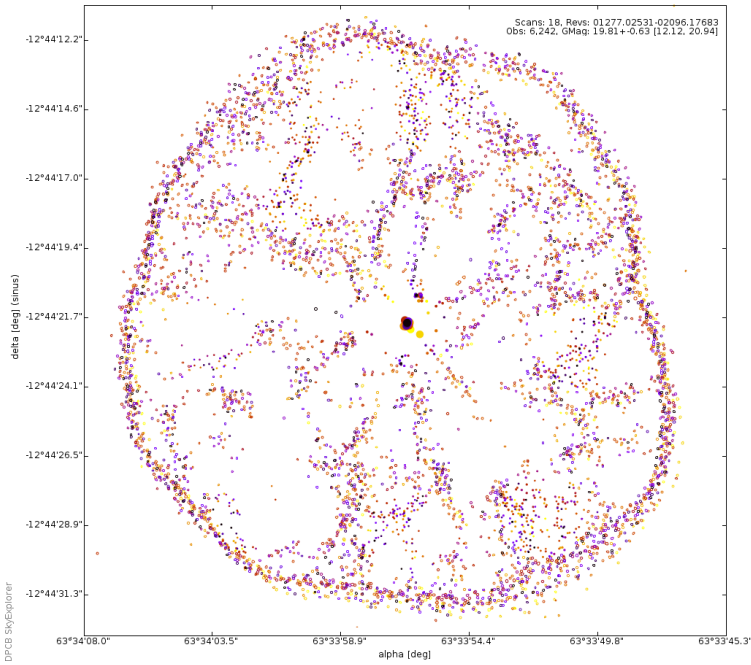


FIGURE 6.19: NGC 1535 planetary nebula. In this case the central star of $12.12th$ magnitude can be identified

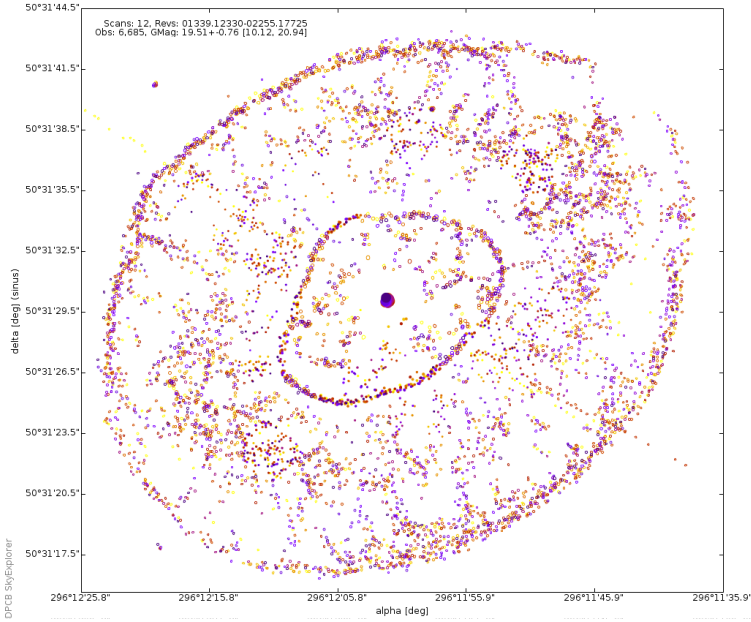


FIGURE 6.20: NGC 6826 planetary nebula around a bright star of 10^{th} magnitude. In this case some spurious detections coming from the bright star spikes were not completely identified

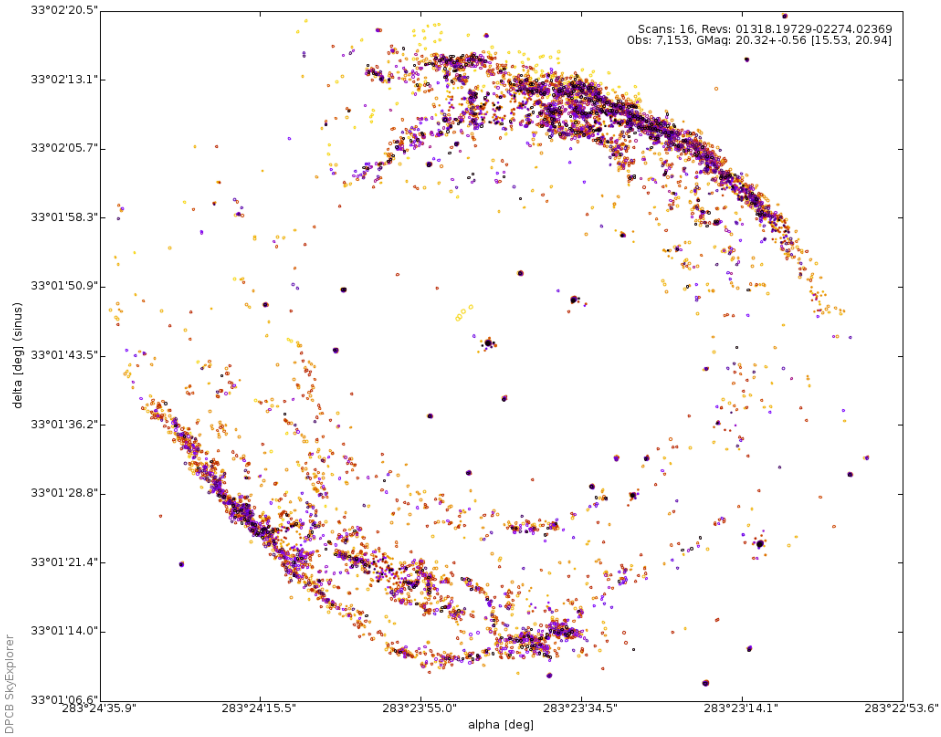


FIGURE 6.21: The Ring Nebula (also catalogued as Messier 57 or NGC 6720). Planetary nebula around a red giant star

6.3.2 Cross–Match: Detection Processor

This section describes the scientific results for the first step of the IDU-XM task, the *Detection Processor*. As described in Section 4.3.1, in this step we basically compute the preliminary source candidates for each input observation. In the first execution of this step we ended with more than 7 billion observations without any source candidate in the IGSL, Figures 6.22 and 6.23 show the time and spatial distributions of these unmatched observations. Additionally, a ∼0.44% of the input observations could not be processed because of the absence of suitable attitude data.

The magnitude distribution of the unmatched observations is shown in Figure 6.24. The distribution basically follows the same profile as the input observations previously shown in Figure 6.3). In this figure we can also check that the ∼90% of the unmatched observation are fainter than 19^{th} magnitude which was in principle expected due to the incompleteness of the IGSL catalogue for the faint magnitude end. Ideally no unmatched

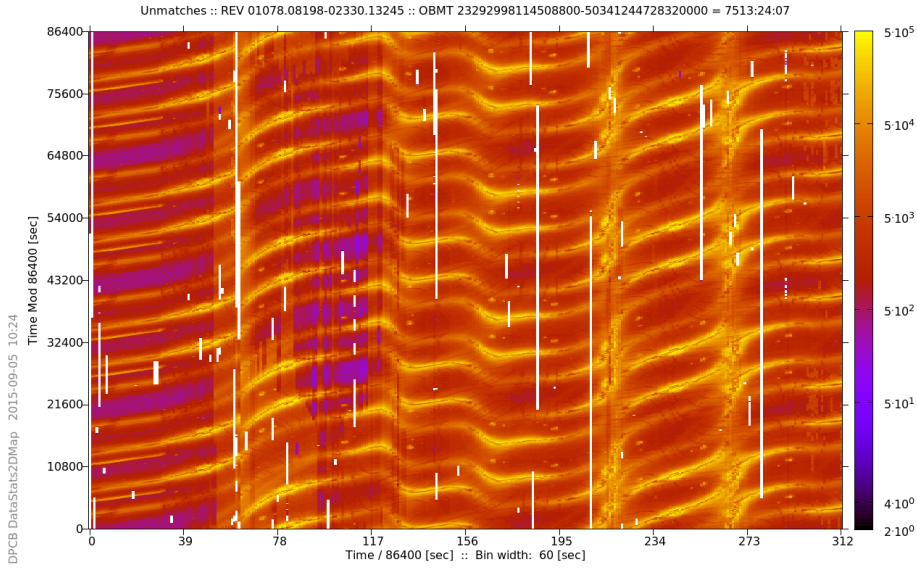


FIGURE 6.22: Time density distribution of the unmatched observations. In the plot it can be identified the effects produced by the scanning law and the updates on the detection magnitude limit at the beginning of the mission

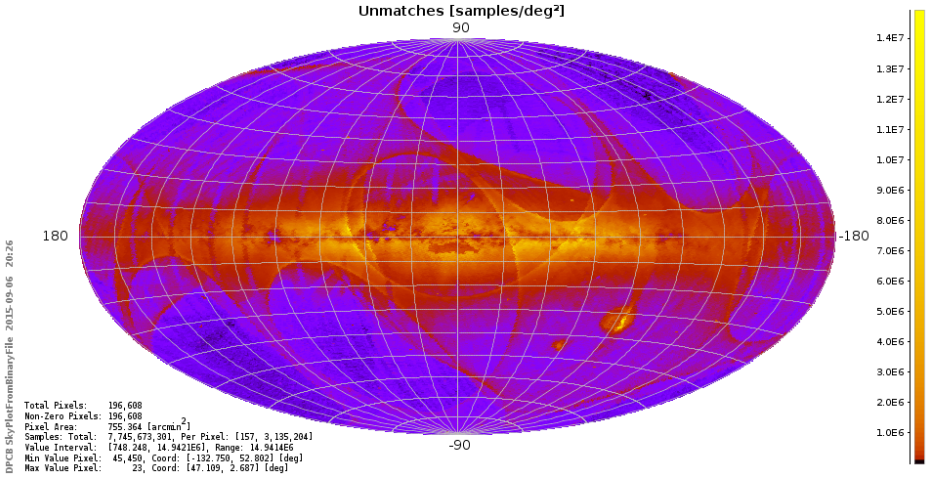


FIGURE 6.23: Sky distribution of the unmatched observations. In the plot it can be also identified the different source densities of the input IGSL catalogue as well as the over densities produced by the scanning law

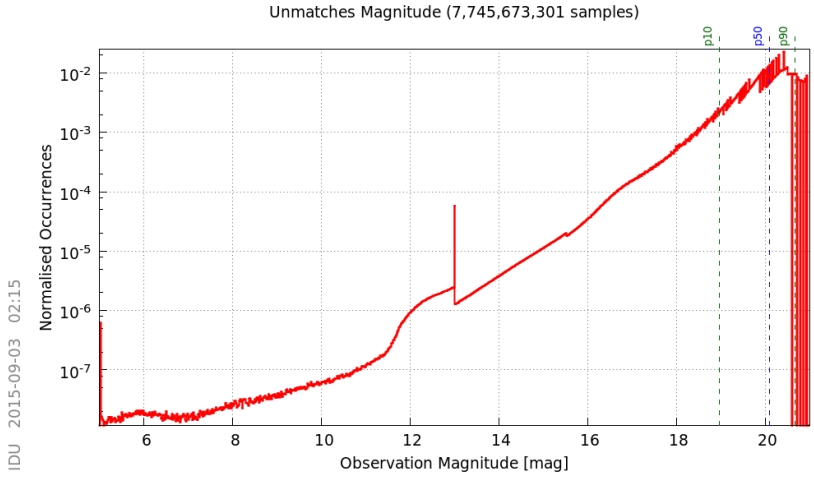


FIGURE 6.24: Magnitude distribution of the unmatched observations

observations brighter than 12^{th} magnitude should be obtained but in this case they are probably caused by the quality of the attitude in some specific periods and some cases could be caused by the ghost detections mentioned already in Section 4.2.

The same figures are available for the matched observations: 6.25, 6.26 and 6.27. However, Figures 6.28, 6.29, 6.30 and 6.31 and are probably more

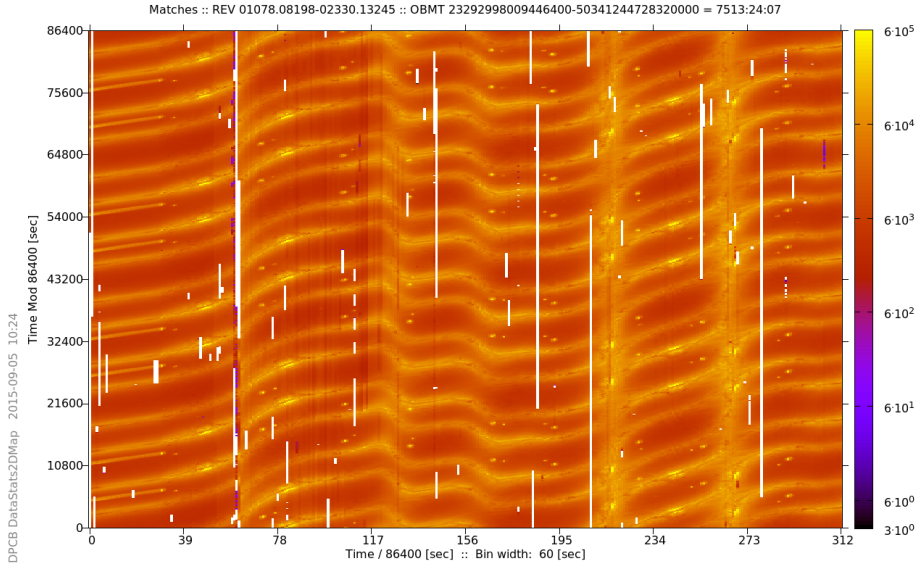


FIGURE 6.25: Time density distribution of the matched observations where the final attitude gaps can be clearly identified

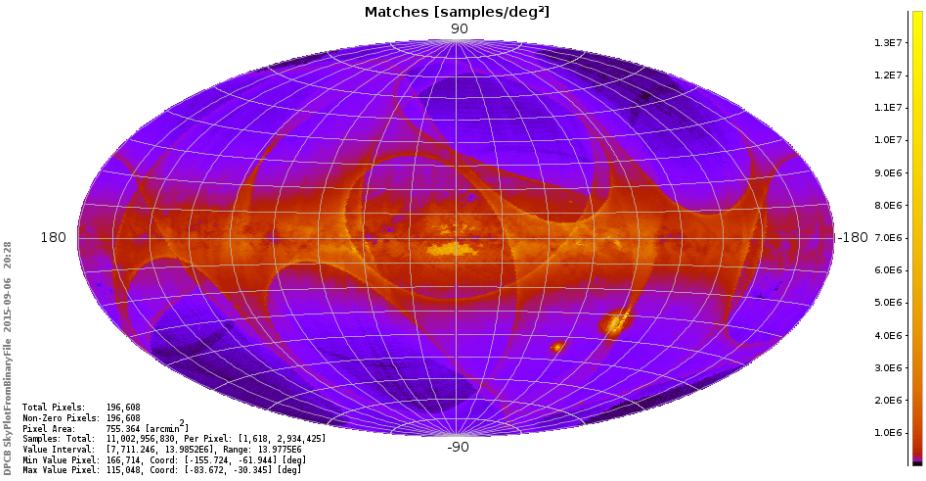


FIGURE 6.26: Sky distribution of the matched observations. In the plot, the different source densities of the input IGSL catalogue can be identified as well as the over densities produced by the scanning law

relevant in this case.

Figure 6.28 presents the match performance in terms of AL and AC distance to the closest source candidate of each matched observation. The

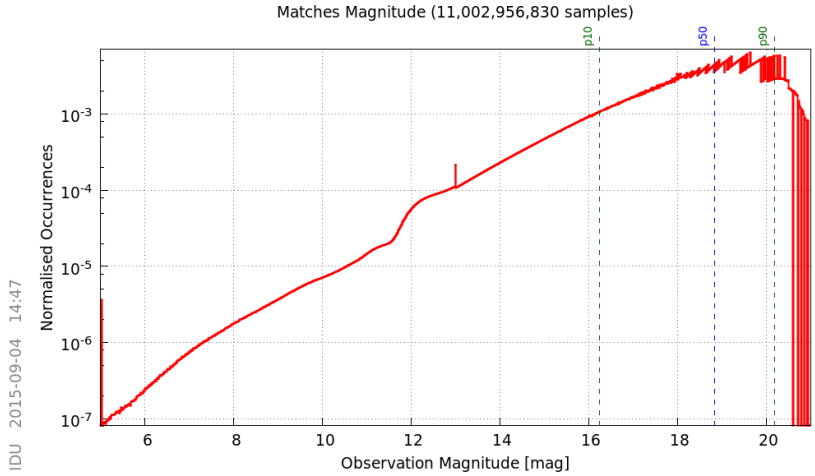


FIGURE 6.27: Magnitude distribution of the matched observations

plot indicates that ∼90% of the source candidates are closer than 1 arcsecond partly justifying the decision regarding the match radius adopted. Also, the distance in the negative AL direction (increasing TDI) presents a slightly higher spread which probably is due to the larger spikes on that direction which consequently produce a larger number of spurious detections than the other directions.

The second one, Figure 6.29, also characterizes the match performance but this time as a function of the observation magnitude. The increasing spread for fainter observations is probably due to the error on the IGSL positions for the faint sources. The bulge starting at 16^{th} magnitude is not yet completely understood although it could be related to the magnitude cut applied in the *Detection Classifier* parameters.

Finally, the last two figures present the distribution of the number of source candidates per observation. Figure 6.30 presents the overall distribution of the source candidate counts whereas Figure 6.31 presents the distribution as a function of the magnitude of the observation. In this plot each magnitude bin is normalized individually and indicates that ∼90% of the observations have only one candidate regardless of the magnitude.

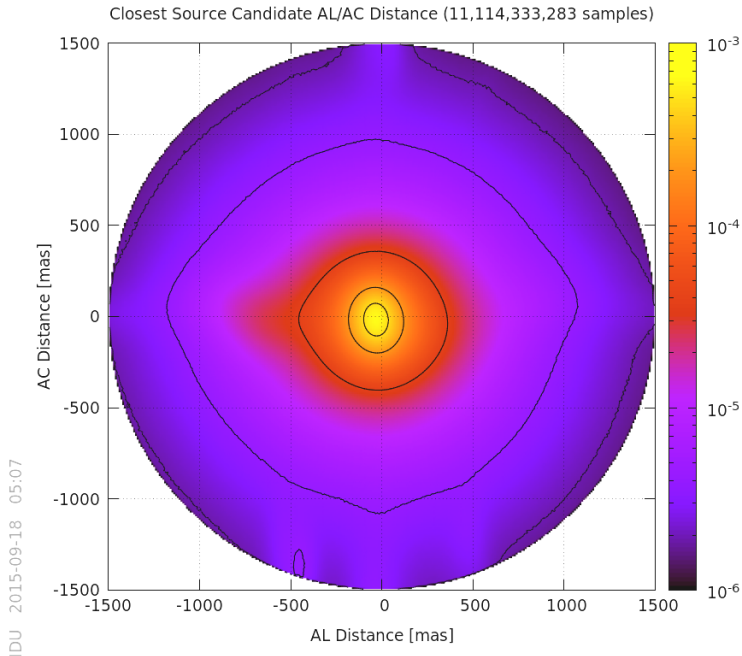


FIGURE 6.28: Match performance in terms of AL and AC distance of the closest source candidate of each matched observation. The plot includes the contours for the 99, 90, 60, 30 and 10 percentiles, this last one corresponding to the smaller region

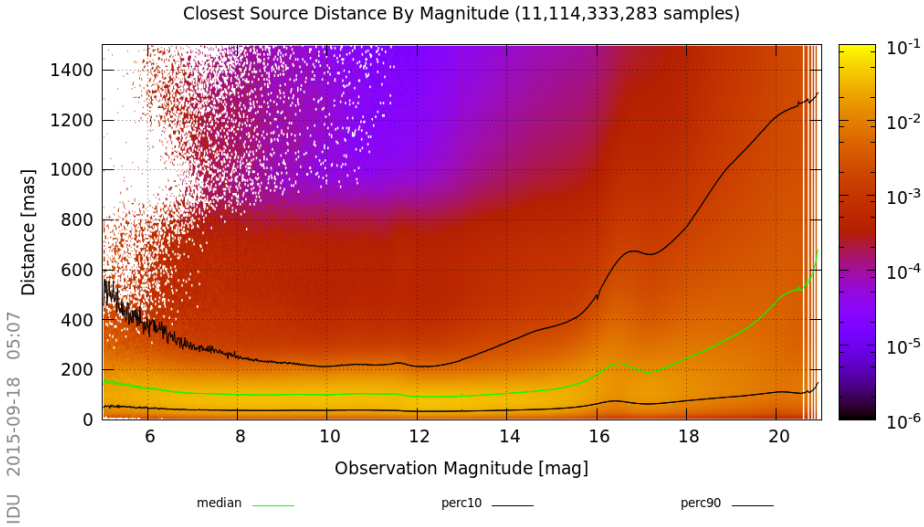


FIGURE 6.29: Number of source candidates as a function of the magnitude and distance to the matched observation. The ~90% of the observations have less than two candidates within a match radius of 1.5 arcseconds

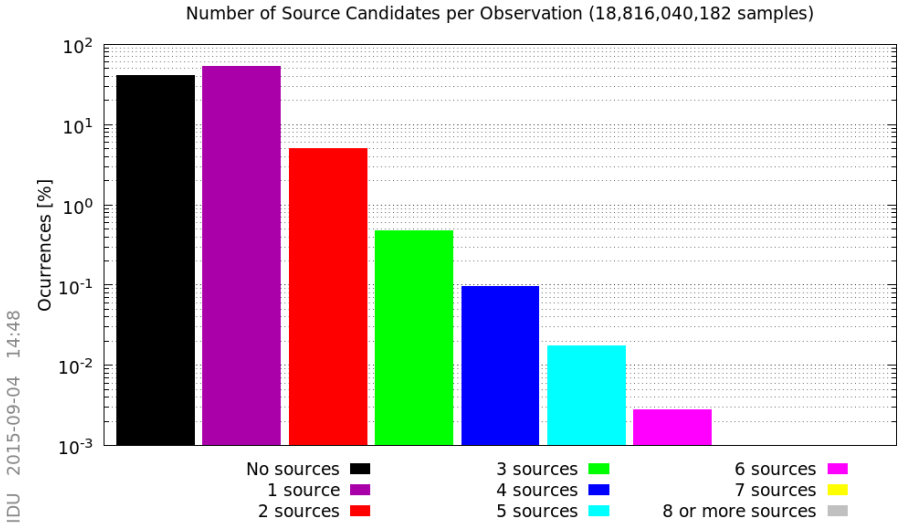


FIGURE 6.30: Distribution of the number of source candidates per observation. The ∼41% of the observations do not have any match in the IGSL catalogue and only a ∼5.5% have more than one candidate

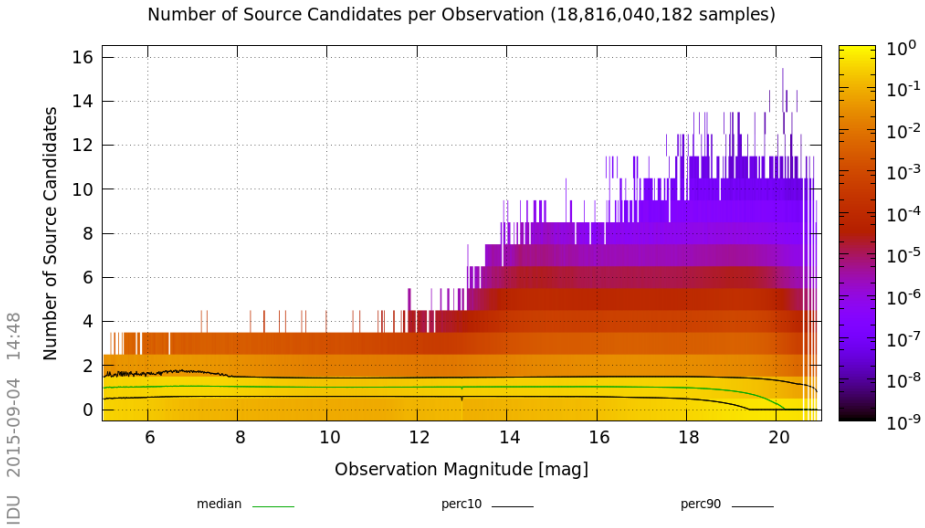


FIGURE 6.31: Distribution of the number of source candidates as a function of the magnitude of the observation. In this plot each magnitude bin is normalized individually and indicates that ∼90% of the observations have only one candidate regardless of the magnitude

Figure 6.32 shows the sky distribution of the IGSL sources that have been matched to at least one Gaia observations whereas Figure 6.35 shows the distribution of the number of matched observations for each individual source. About 11 billion of the 18 billion input observations ($\sim 60\%$) have been matched to the IGSL. The percentage of matched IGSL sources is $\sim 80.5\%$ and the underlying causes for the unmatched sources could be among others:

- The quality of the IGSL itself which might contain due to the quality of the original catalogues:

  - sources incorrectly placed

  - duplicated entries mainly due to the overlap of the Schmidt plates for the PPMXL and GSC2.3 catalogues

  - dubious proper motions (as shown in Figure 6.34)

  - etc.

- The reported attitude issues (bad quality, excursions, spacecraft decontamination, etc.) and poor geometric calibration might cause observations to be misplaced. Figure 6.36 shows an example of the observation misplacement for the extended object NGC7009. In this unusual case, we can see an offset on the location of the observations of one of the scans of ~ 30 arcseconds which was caused by a decontamination activity performed in September 2014.

- IGSL sources brighter than 5^{th} magnitude may not be detected by Gaia.

- Gaia may resolve in separated observations some blended IGSL sources. These resolved observations could present a significant offset with respect the mean position defined in the corresponding IGSL entry being this one left unmatched.

After this first execution, the unmatched observations are processed to create intermediate new sources (see Section 4.3.1). With these temporary new sources, the source candidates for each input observation are

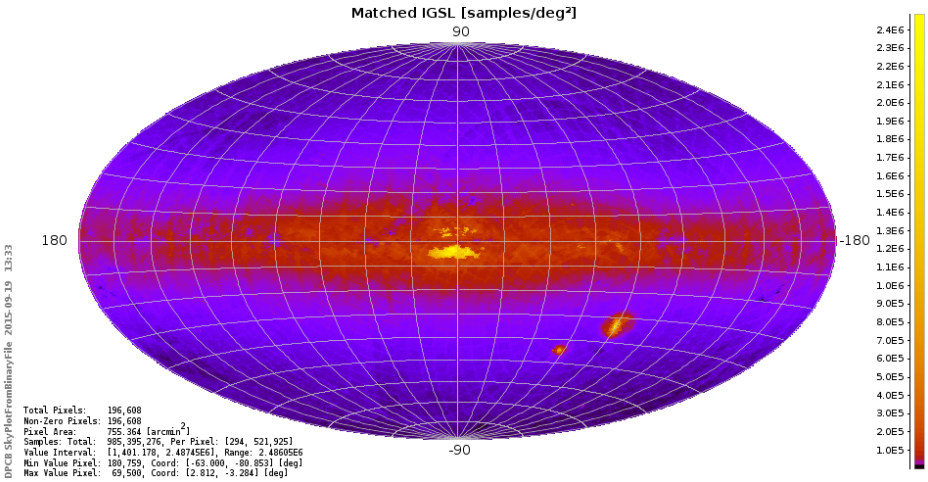


FIGURE 6.32: Sky distribution of the 9 855 395 276 matched IGSL sources

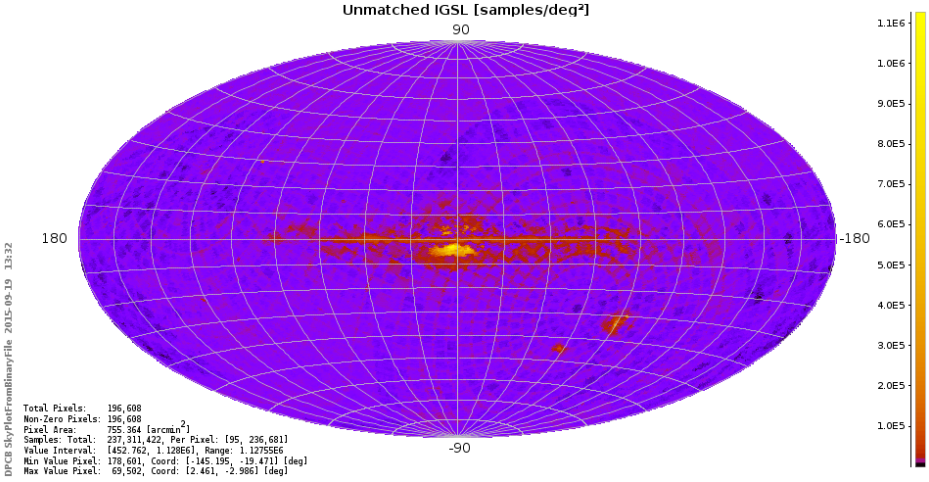


FIGURE 6.33: Sky distribution of 237 311 422 unmatched IGSL sources. The unmatched sources are concentrated in the galactic plane, the Magellanic clouds and in the overlap of the Schmidt plates for the PPMXL and GSC2.3 input catalogues

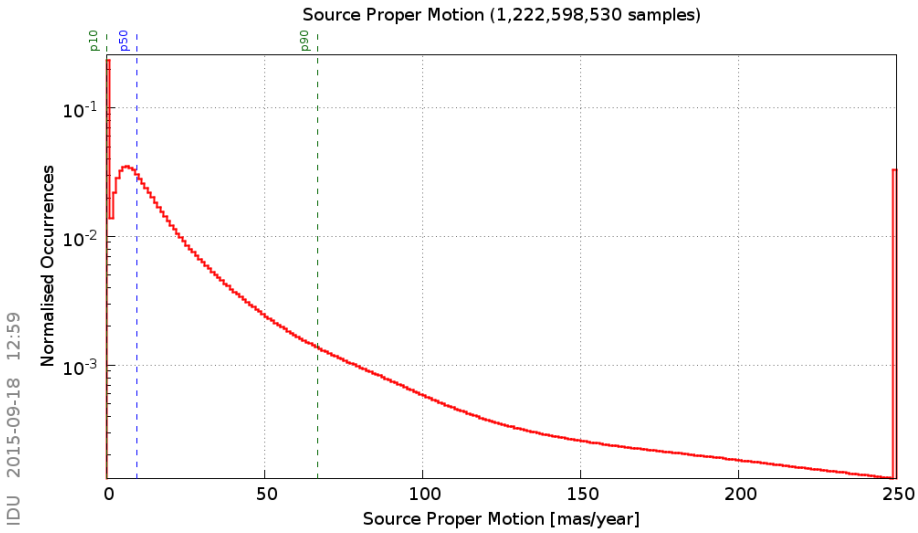


FIGURE 6.34: Distribution of the absolute proper motion of the IGSL sources. We can see that $\sim 24\%$ of the sources have zero proper motion

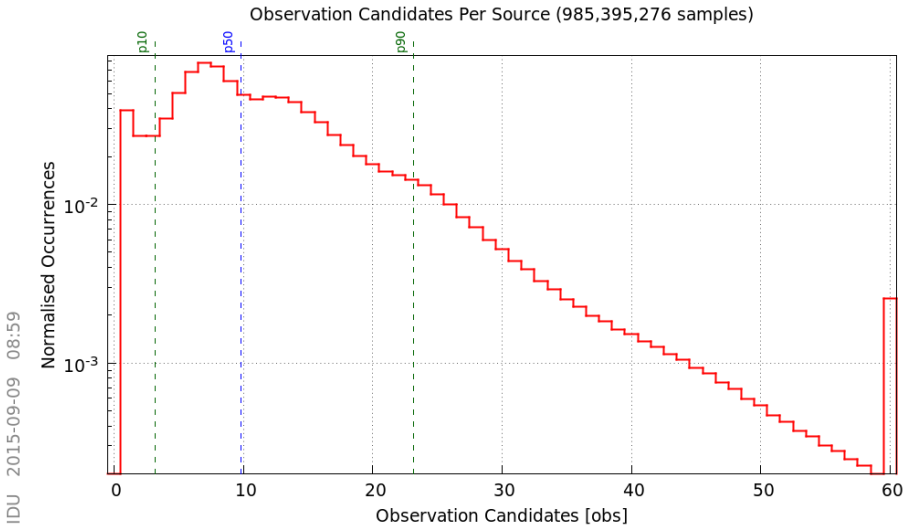


FIGURE 6.35: Distribution of the number of observations matched to an IGSL source

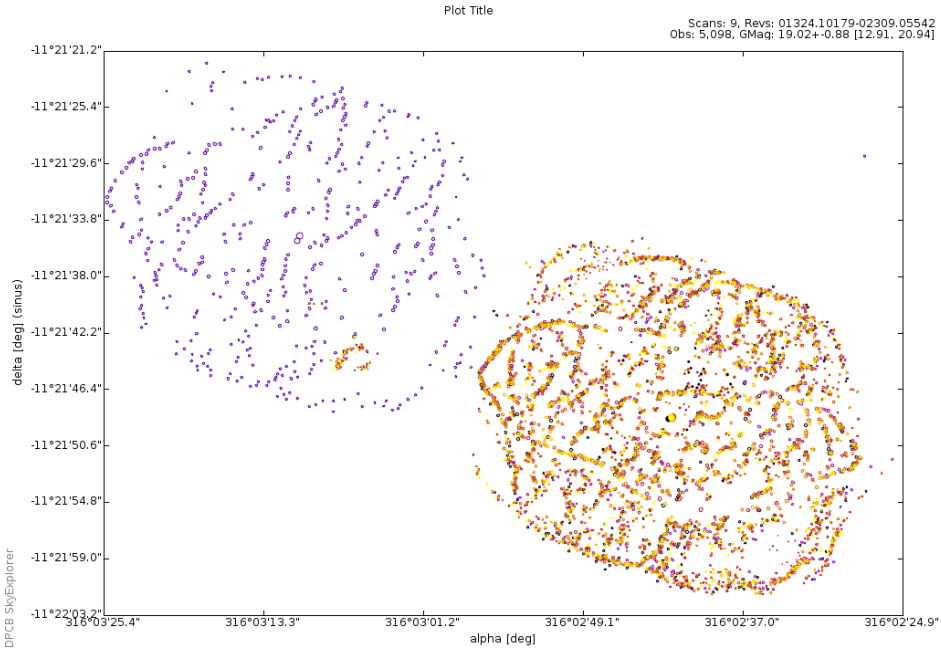


FIGURE 6.36: Example of the observation misplacement for one scan of NGC7009 during the decontamination activity in September 2014 where the attitude was not properly determined. The blue scan, with 522 observations, present an offset of 30 arcsecond with respect to the other 8 scans of the same extend object. Each one of these misplaced observations produces a new spurious source in the final catalogue

determined again. In the new results, all observations have at least one source candidate, not leaving any unmatched observation. The statistics of this second run are not scientifically relevant because these temporary new sources will be removed in the *Sky Partitioner* run and ultimately will be superseded in the final resolving step.

6.3.3 Cross–Match: Sky Partitioner

This section includes the main diagnostics and findings over the groups of *MatchCandidates* created by the *Sky Partitioner* processing. From this processing a total amount of 2 018 275 715 *MatchCandidateGroups* were identified. Figure 6.37 shows the sky distribution of these obtained groups which still present the characteristic distribution of the input observations – where the Gaia scans are still visible. Ideally, this distribution should

be more similar to the distribution of the sources without any trace of the Gaia scans which should be completely blended by the grouping of the observations. In this case, some scans are still visible probably due to the remaining spurious detections, detection limit configuration on board and the fact that some regions have been observed more times than the others.

Figure 6.38 and Figure 6.39 provide the distribution of the number of observations and sources in the generated *MatchCandidateGroups*. From these figures we can see how the majority of the groups have been created from isolated single sources with very few observations in average. This fact is even more evident in Figure 6.40 where both parameters are represented in a single 2D plot.

Although these are relevant statistics to describe the generated *MatchCandidateGroups*, they might be biased due to the large amount of groups created from the spurious detections.

It is also interesting to analyse the *MatchCandidateGroup* composition regarding the number of unmatched and matched observations to the input IGSL. Figure 6.41 and Figure 6.42 show the distribution of these two sets whereas Figure 6.43 presents their ratio. In this last figure we can see how $\sim$ 50% of the groups have been created from observations without

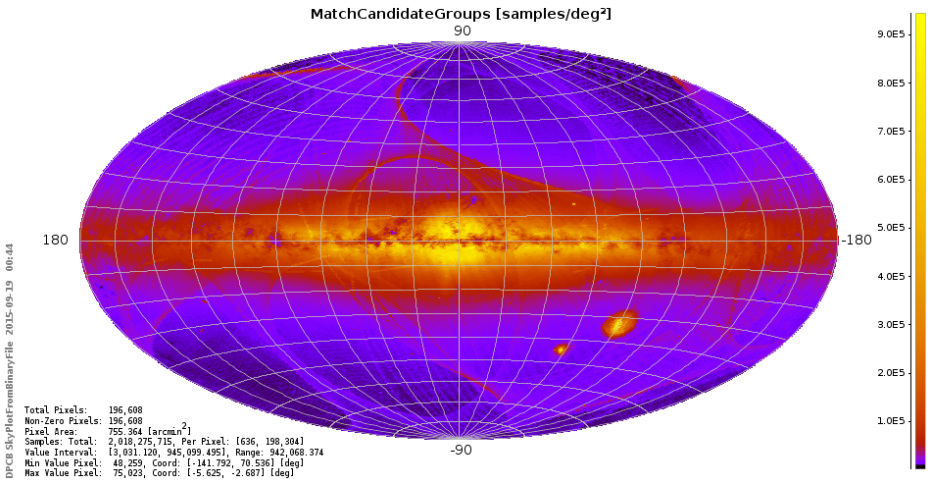


FIGURE 6.37: Sky distribution of the 2 018 275 517 *MatchCandidateGroups*

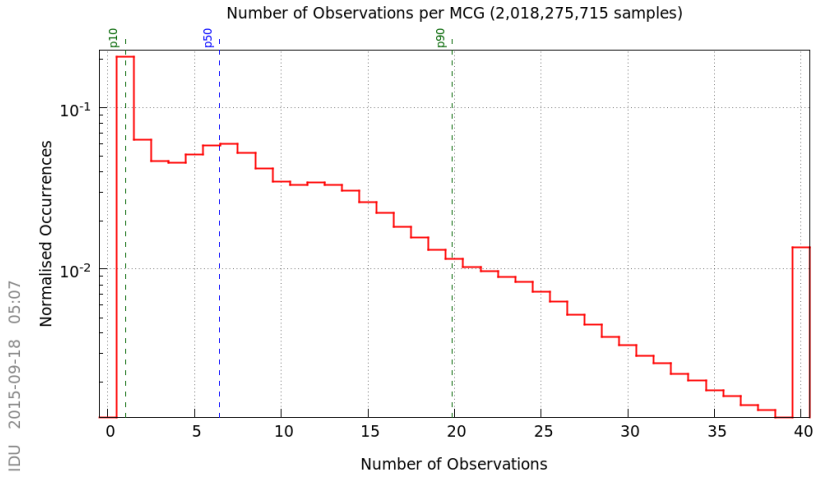


FIGURE 6.38: Number of observations per *MatchCandidateGroup* indicating that 90% of the groups have less than 12 observations, and 50% less than 3

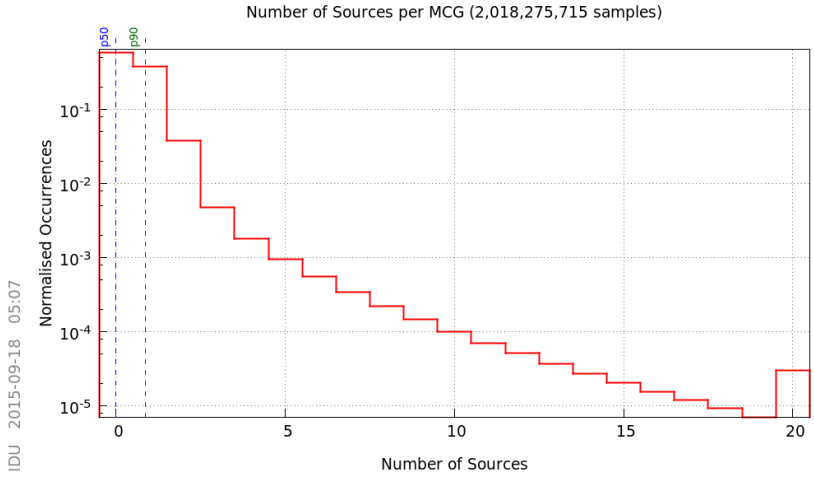


FIGURE 6.39: Number of sources per *MatchCandidateGroup* indicating that 50% of the groups does not have any IGSL source included and that 90% only have one single source

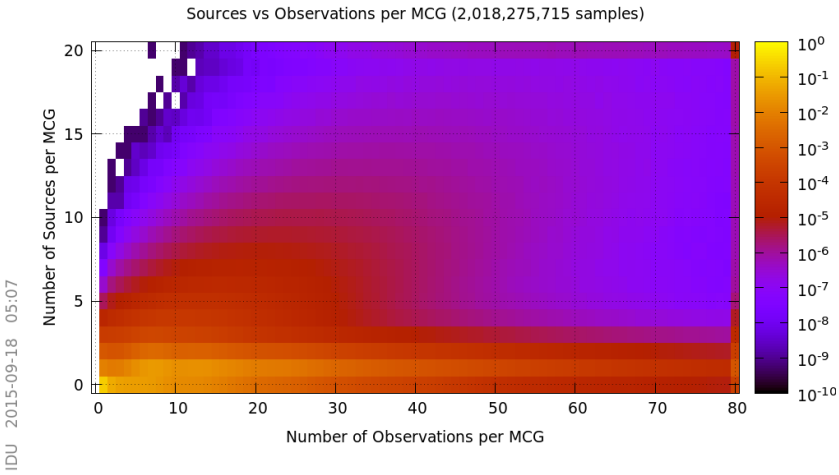


FIGURE 6.40: 2D map providing the *MatchCandidateGroups* distribution according to the number of observations and sources grouped

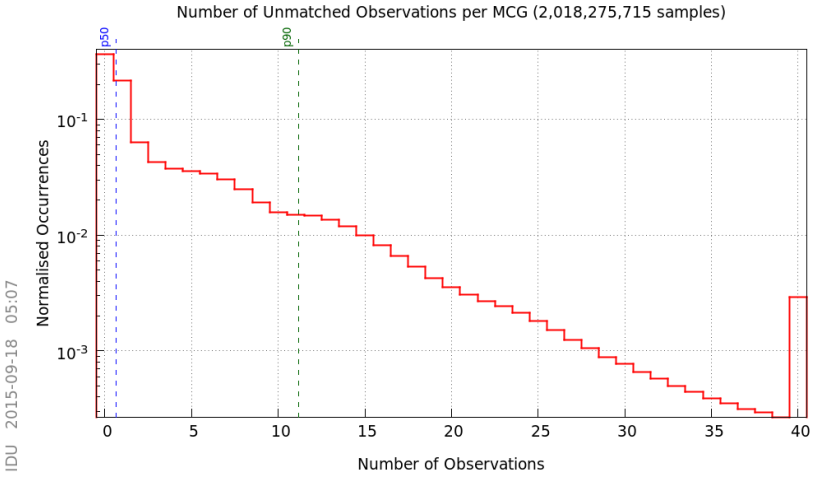


FIGURE 6.41: Distribution of the number of unmatched observations per *MatchCandidateGroup*

any match to an IGSL source whereas only $\sim 10\%$ have been created completely from matched observations.

Figure 6.44 shows the *MatchCandidateGroups* distribution according to the number of scans of the grouped observations. This is specially significant as it shows how many times a group has been observed. This provides a hint on the number of observations per source expected. For DS-00, 90% of groups have been scanned less than nine times.

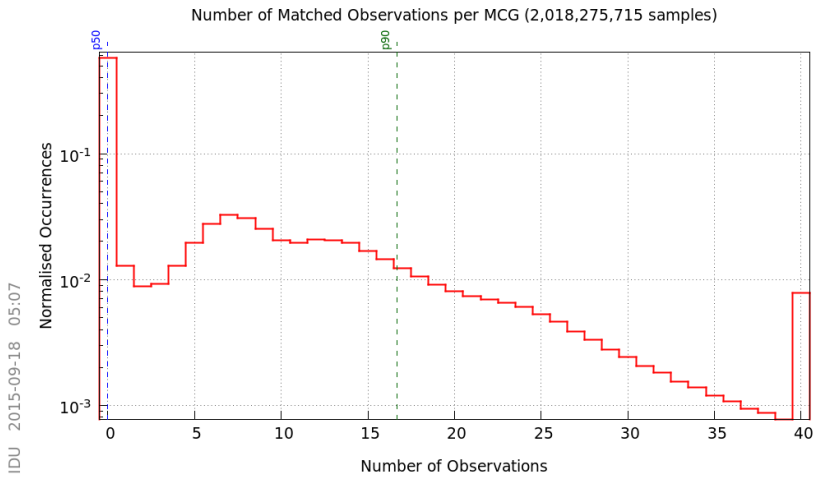


FIGURE 6.42: Distribution of the number of matched observations per *MatchCandidateGroup*

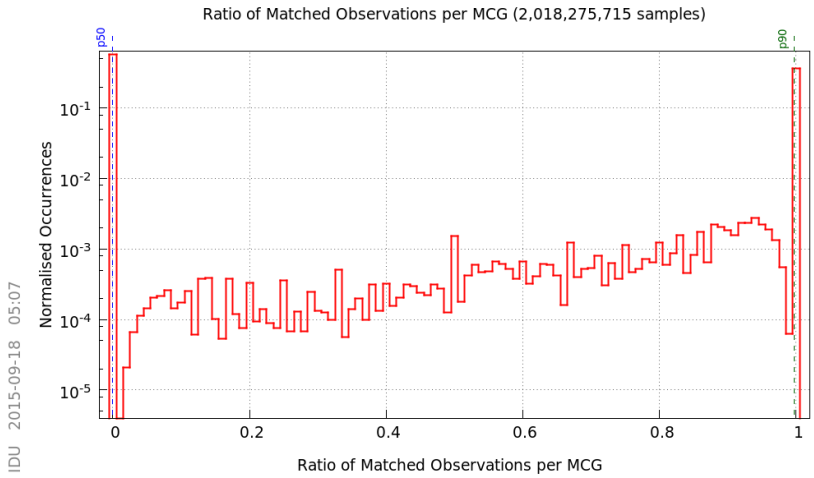


FIGURE 6.43: Distribution of the ratio of the matched and unmatched observations per *MatchCandidateGroup*. In this plot we can see how $\sim 50\%$ of the groups have been created from observations without any match to an IGSL source whereas a $\sim 10\%$ have been created only from matched observations

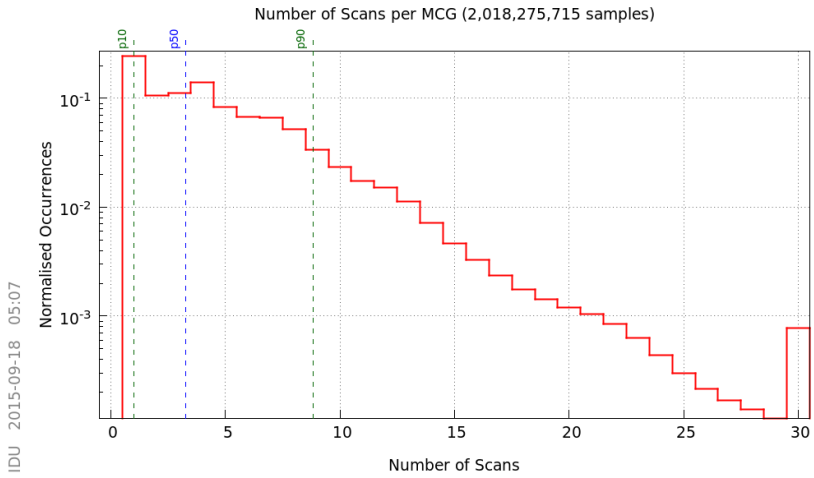


FIGURE 6.44: *MatchCandidateGroups* distribution according to the number of scans of the grouped observations

Finally, we have also analysed the dispersion of the observations within the created *MatchCandidateGroup*. Figure 6.45 shows the distribution of the distance from the observations to the *MatchCandidateGroup* center. From the plot we can see that the 90% of the observations are very close to their respective group center thus indicating that most of the groups are quite small and concentrated in an area with a radius smaller than 800 milliarcseconds. In fact more than 50% of the observations are closer than 100 milliarsecond which probably means that groups have been created from a single source (which is again consistent with the previous plots commented).

It is worth pointing out that this processing step is not only helpful for grouping and isolating the detections for their later resolution but it can also be used to identify and reveal complex structures of detections. One example is the extended objects discussed previously in Section 6.3.1. In practice, we could play with the match radius configuration in the *Detection Processor* step to be able to reveal in the *Sky Partitioner* processing structures at quite different scales and sizes.

Finally, after analysing the results it is worth noting the following:

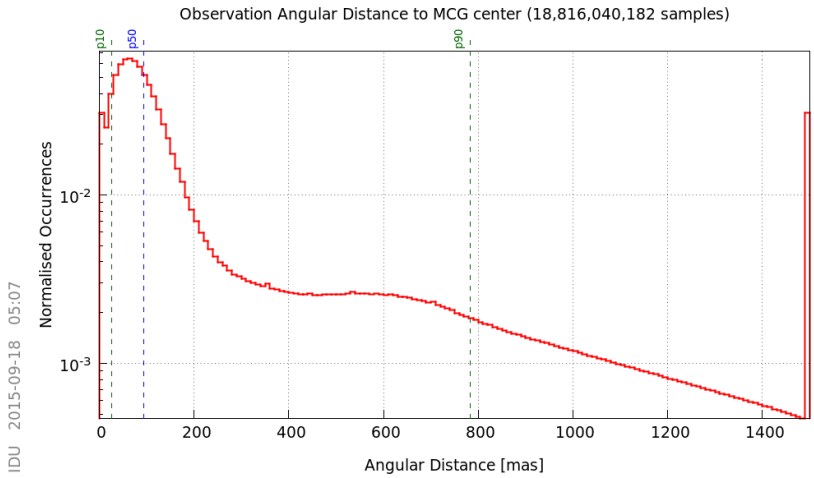


FIGURE 6.45: Distribution of the distance between the *MatchCandidateGroup* center and the grouped observations

- Identified groups are affected by the remaining spurious detections resulting in unexpected groups covering large and elongated sky regions and with a high amount of inter–linked observations and sources.
- The match radius used in the *Detection Processor* step must be chosen carefully to avoid creating too big groups or on the contrary separate groups of observations that should be resolved together. Consequently, this match radius should be calibrated probably as a function of the errors of the source catalogue parameters.
- The *SkyPartitioner* task can identify not only isolated groups of observations but also the observations in high surface brightness filamentary structures as the ones corresponding to the extended objects.

6.3.4 Cross–Match: Match Resolver

This section includes all the diagnostics over the final products resulting of the resolution of the *MatchCandidateGroups*.

The final step of the IDU-XM produced a total amount of 1 995 652 409 new sources. This number basically duplicates the initial input source count

of the IGSL catalogue. Figure 6.46 shows the sky distribution of the new sources. The new sources are concentrated in the galactic plane but we can clearly distinguish also the effects of the scanning law probably accentuated by the remaining spurious detections. Additionally some bands present a higher density that might also be caused, in part, by the original IGSL catalogue incompleteness – issue to be studied further.

Figure 6.47 shows the distribution of the magnitude of the new sources. This figure is again demonstrating that most of the new sources are very faint (above 20.3). There is a peak at 5^{th} magnitude that is suspected to be the result of double detections and spurious detections around very bright sources which have been detected by Gaia as 5^{th} magnitude. The peak around 13^{th} magnitude is a reproduction of the one also seen in the input observations (see Figure 6.3) which is then carried on through the different processing steps until its final resolution. Therefore it is likely that this is due to detections of cosmic rays and the on board detection limitations as commented in previous sections and it might be perfectly normal.

Figure 6.48 shows the distribution of the number of matched observations to the new sources where it can be seen how 50% of the new sources has less than 2 observations. Similarly, Figure 6.49 shows the same distribution

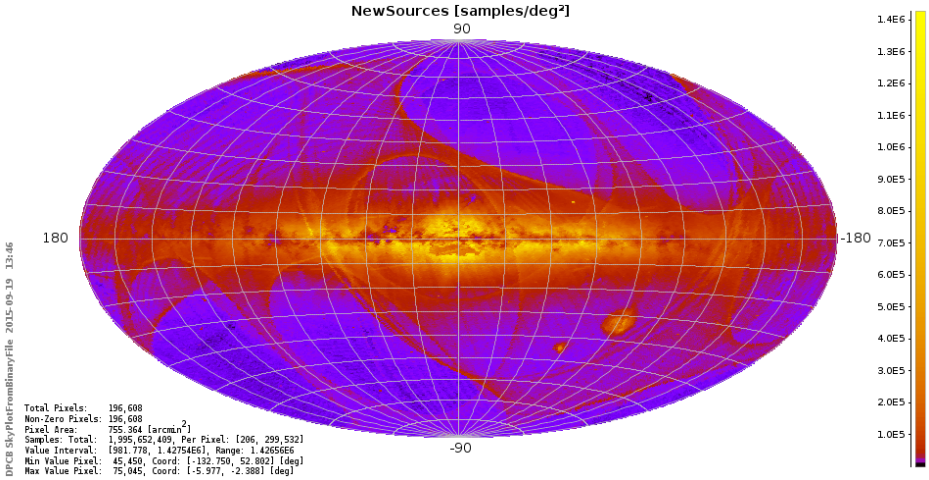


FIGURE 6.46: Spatial distribution of the 1 995 652 405 new sources

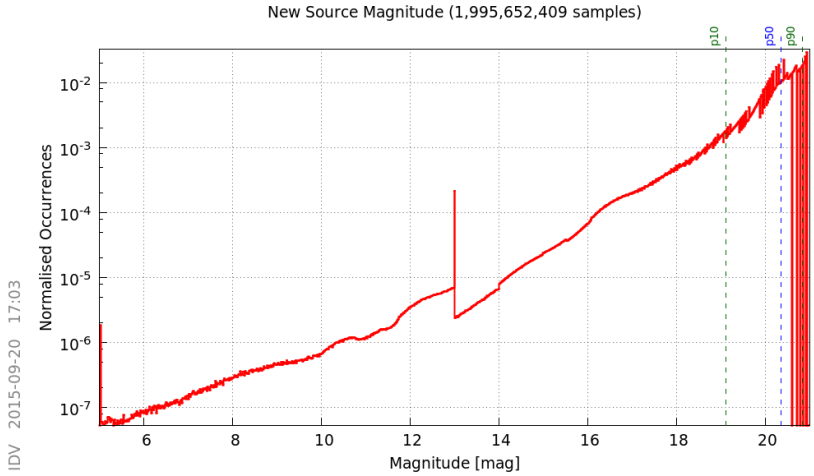


FIGURE 6.47: New source distribution by magnitude (same bin size than the figures with the magnitude of the observations)

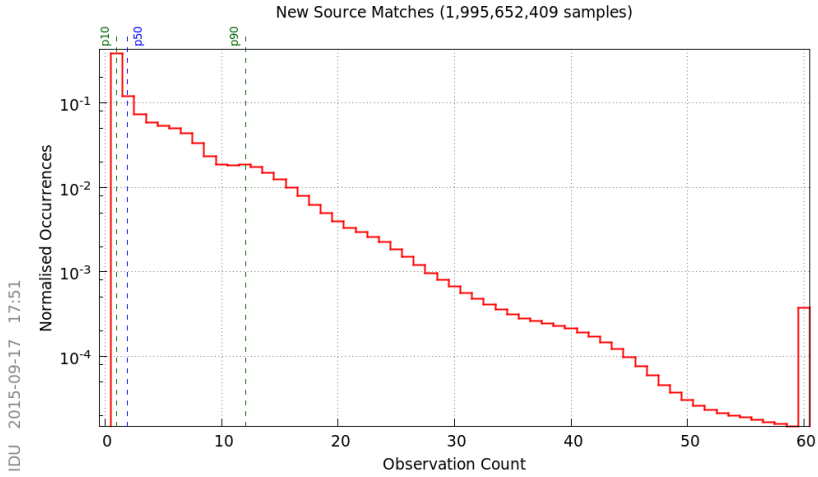


FIGURE 6.48: Distribution of the number of observations matched to the new sources

this time as a function of the magnitude of the new sources. In this last plot we have indicated the 60, 90 and 99 percentiles.

Figures 6.50 and 6.51 show the same information but this time limited with the magnitude range limited from 18 to 21. These figures have been included to demonstrate that half of the created new sources are very faint (above 20.3 magnitude). In the last figure, it can also be seen that $\sim 60\%$ of the new sources have very few observations compared to the mean number

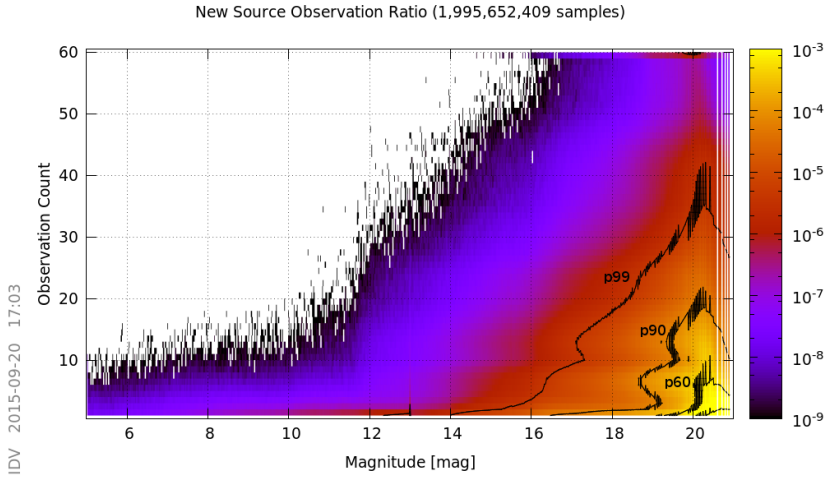


FIGURE 6.49: Distribution of the number of observations matched as a function of the new source magnitude

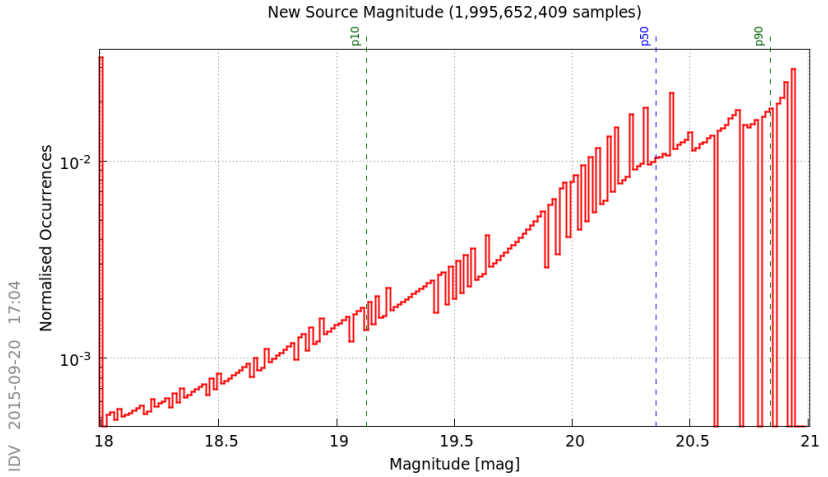


FIGURE 6.50: New source distribution by magnitude – zoom from 18^{th} to 21^{th} magnitudes

of scans and they are all concentrated in the spurious magnitude range. Probably these new sources have been created from spurious detections which will explain the few matched observations. Basically these new sources are matched to other spurious detections from consecutive scan with similar scan direction angles. This issue will require further investigation from the IDT and IDU teams to understand better the situation.

Regarding the *Match* and *AmbiguousMatch* tables, Figure 6.52 shows the

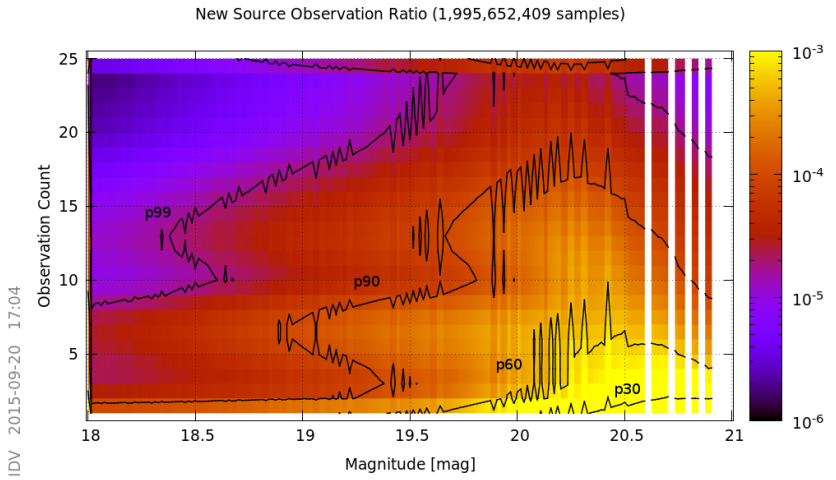


FIGURE 6.51: Distribution of the number of observations matched as a function of the new source magnitude – zoom from 18^{th} to 21^{th} magnitudes

distribution of the distances of the primary source match. In this figure we can see a peak at zero corresponding to $\sim$ 12% of the observations. This peak reflects basically the features of the resolution algorithm used. This algorithm basically creates the new sources directly using the position of the first unmatched observation, thus having a zero distance. The step at 1 arcsecond is also related to the unmatched observations processing which was configured to used a maximum match radius of 1 arcsecond. Finally, the bulges at 100 and 700 milliarcseconds probably are related to the IGSL errors and the attitude issues commented in Section 6.2 respectively.

Figure 6.53 shows the distribution of the number of ambiguous matches for each observation. As can be seen, roughly 77% of the cases have only one source candidate and only $\sim$ 5% of the observations have more than two candidates.

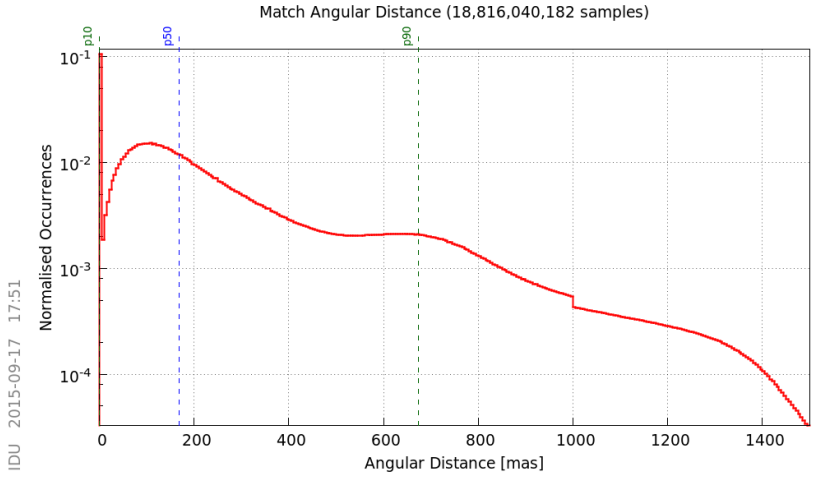


FIGURE 6.52: Distribution of distances to the primary matched source with a bin size of 5 milliarcseconds

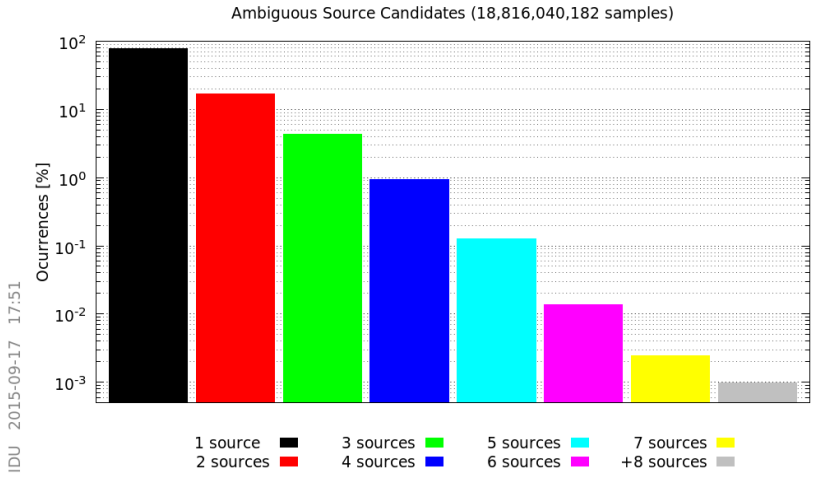


FIGURE 6.53: Number of source candidates per observation, also referred as ambiguous matches

6.3.5 Summary of the Operational Execution

As anticipated, the scientific results obtained in the operational run present exactly the same behaviour as the results just shown corresponding to the second reprocessing. Only small differences in some counts and percentages are appreciable, being the profiles in all diagnostics exactly the same.

Table 6.2 provides a list of the produced outputs during this exercise as well as the comparison with the initial expected outputs. The expected number of *Match* and *AmbiguousMatch* records shown in this table corresponds to the number of observations after applying the filtering according to the results from the IDU-DC. The missing records correspond to the observations that could not be processed due to missing or corrupted attitude as reported in the task logs. We can conclude that the data entering the processing and produced were properly accounted for. The differences detected correspond to data that could not be processed due to the already noted issues at the beginning of this chapter.

MDB Table	**Expected**	**Output**	**Difference**
BlackListedTransit	N/A	3 241 067 614	N/A
Match	18 837 883 989	18 760 735 128	77 148 861
AmbiguousMatch	18 837 883 989	18 760 735 128	77 148 861
NewSource	N/A	1 960 186 877	N/A
Track	N/A	0	N/A

TABLE 6.2: Output data counts for the full DS-00

The first stage covered roughly 85% of the data entering the DS-00, including the Ecliptic Pole scanning law data period acquired from 25th of July to 27th of August 2014 and the first galactic plane scan end of February 2015. Table 6.3 and Table 6.4 present the most relevant counts for both the input and output data involved during this stage. The total volume of the output data produced was of 426.52 GBytes.

The second stage of the reprocessing was carried out over all the accumulated observations from the end of the first stage up to the end of April 2015. This stage covers about a month of data including the second galactic plane scan. The execution was initially scheduled to start by the 4th of

Data type	Record count	Data Gap
IGSL Source	1 222 598 530	
Attitude	11 414	2.28%
Object Log	18 033 996 583	0.86%
Observation	15 842 952 901	1.71%

TABLE 6.3: Summary of the main inputs involved in the first stage

Data type	Record count	Data Size
BlackListedTransit	2 383 736 073	17.69 GBytes
Valid Observation	13 382 127 410	
Unmatched Observation	5 419 468 826	
IGSL source hits	968 445 175	
MatchCandidateGroup	1 946 673 321	
Match	13 382 127 410	147.04 GBytes
AmbiguousMatch	13 382 127 410	209.85 GBytes
NewSource	1 747 699 573	51.94 GBytes

TABLE 6.4: Summary of the main outputs produced in the first stage

May but it was delayed due to inconsistencies in the input data received at DPCB. Table 6.5 and Table 6.6 present the main counts for the input and output data of this stage. The total volume of the output data produced was of 152.28 GBytes.

Data type	Record count	Data gap
Object Log	6 316 525 432	4.37%
Observation	5 817 756 335	2.42%
Attitude	3217	2.42%
Stage 0 NewSource	1 747 699 573	

TABLE 6.5: Summary of the main inputs involved in the second stage

Data type	Record count	Data Size
BlackListedTransit	780 365 735	5.84 GBytes
Valid Observation	5 037 331 604	
Unmatched Observation	159 029 735	
IGSL source hits	848 991 497	
MatchCandidateGroup	1 441 243 602	
Match	5 037 331 604	55.88 GBytes
AmbiguousMatch	5 037 331 604	84.51 GBytes
NewSource	201 147 620	6.06 GBytes

TABLE 6.6: Summary of the main outputs produced in the second stage

Last stage processed the remaining observations from the end of the previous stage to the end of DS-00. This stage covers about a week of data.

Table 6.7 and Table 6.8 present the counts for the input and output data. The total volume of the output data produced was of 10.63 GBytes.

Data type	Record count	Data Gap
Object Log	350 922 831	17.66%
Observation	418 242 367	0.60%
Attitude	403	0.60%
Stage 0 NewSource	1 747 699 573	
Stage 1 NewSource	201 147 620	

TABLE 6.7: Summary of the main inputs involved in the last stage

Data type	Record count	Data Size
BlackListedTransit	76 965 806	557.36 MBytes
Valid Observation	341 276 114	
Unmatched Observation	9 274 928	
IGSL source hits	55 602 934	
MatchCandidateGroup	86 726 992	
Match	341 276 114	3.89 GBytes
AmbiguousMatch	341 276 114	5.84 GBytes
NewSource	11 339 684	360.39 MBytes

TABLE 6.8: Summary of the main outputs produced in the last stage

Unfortunately, after the first trial run of the MDB Integrator, in late July 2015, it was discovered that some *Match* records were referring to non-existing sourceIds. We investigated this issue and isolated the problem in the new sourceId consolidation step (after the IDU-XM resolution). The problem is quite complex to detail and it could happen only because of the non-nominal splitting strategy followed for this activity – i.e. running the IDU-XM in three different stages in the same DRC. In a nominal DRC execution this issue would never have shown up.

A software implementation error in this consolidation process accidentally allowed *Match* records from the last two stages to be wrongly updated. As a result, the matches of these two stages referring to the new sources created in previous stages were corrupted and this corruption led to the following possible cases:

- *Matches* pointing to sourceIds that do not exist.
- *Matches* pointing to a wrong sourceId. This happens when a Match pointed to a sourceId from a previous stage and after incorrectly

applying the sourceId update in the *Match* the new sourceId has clashed with a sourceId used in the new sources of the current stage.

From the analysis of all the data, we can ensure that:

- *Matches* to IGSL sources are not affected.
- All new sources created are valid. Consequently, there is no problem in integrating them in the MDB Integrator run and IDT can also safely use them.
- The *Matches* and *AmbiguousMatches* from the first stage (first 8 months of data) are not affected.
- Only a fraction of the *Match* and *AmbiguousMatch* from the second and third stages are corrupted. This fraction is limited to the *Matches* and *AmbiguousMatches* pointing to the new sources created during the previous stages.

The number of *Match* entries affected by stage are:

- Second Stage: 2 670 530 955 out of a total of 5 037 331 604 observations (53.01%)
- Last Stage: 169 642 321 out of a total of 341 276 114 observations (49.70%)

This issue may affect the downstream systems but the reprocessing of the affected data was not considered a priority taking into account the following facts:

- The affected data can be easily identified by means of the solutionId of the new sources and the *Match* records.
- Most of the affected observations are from the faint magnitude end, probably from spurious detections

- A new IDU-XM will be performed which will supersede all the results. This new execution is currently scheduled for October 2015, after a very short DS covering only 3 months of new data.

6.4 Computational Performance

The early IDU execution was carried out in the nominal operational DPCB hardware described in Appendix A. The common software used in the Marenostrum processing nodes for this activity was the following:

- IBM J9 VM build 2.6, JRE 1.7.0 Linux amd64-64 (JIT enabled, AOT enabled)
- Linux version 3.0.101-0.35-default (SUSE Linux)

The task jobs have been defined following the procedures described in Section 5.1.2 and its distribution has been done using Greasy (introduced in Section 5.1.3).

The following sections summarise the task execution plan followed for each IDU task including the computational performance obtained. We focus not only on the consumed CPU hours but also in all the details related to the node usage and I/O load, searching for any kind of performance issues during the processing. As already commented we only detail the results of the second run of the reprocessing although specific statistics and metrics for each one of the operational execution stages have been summarised in Section 6.4.5.

6.4.1 Detection Classifier

For the IDU-DC execution, we split the processing of the full DS-00 extent in time intervals of approximately one million observations. These jobs

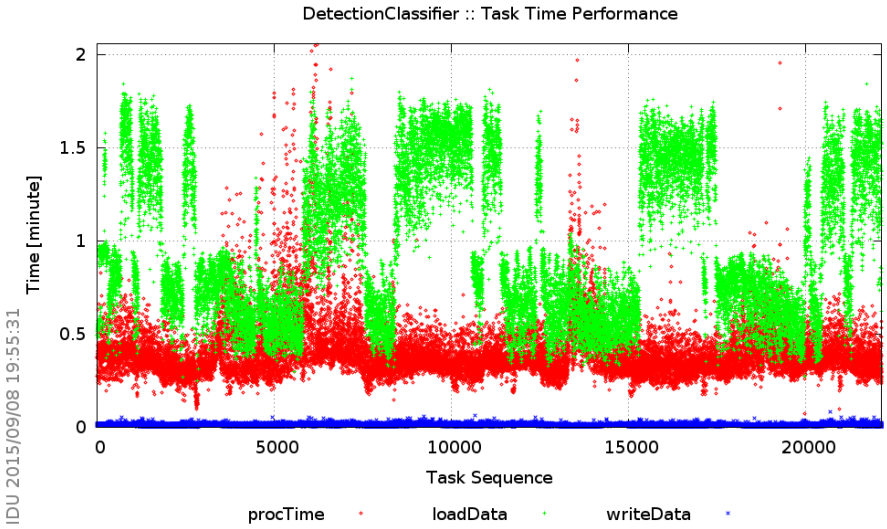


FIGURE 6.54: Job processing and I/O times for the *Detection Classifier* execution

then were processed in Marenostrum using Greasy in several nodes. Table 6.9 summarises the main execution parameters and metrics obtained in the execution of the IDU-DC task.

Total Jobs		22 149
Resources		30 Nodes
		10 Jobs/Node
		1.6 CPUs/Job
Total Wall Clock		1h 58m 19s
Task CPU Time		946h 32m 00s
Job Time	Total	558h 48m 53s
	Avg	1m 30s
	Min	29s
	Max	10m 18s
Output Size	Data	34.2 GBytes
	Logs	2.8 GBytes

TABLE 6.9: Performance metrics of the preliminary *Obs–Src Match* step

From the statistics from Figures 6.54 and 6.55, it is clear that the current IDU-DC implementation is dominated by the data loading. The jobs are currently loading both the object logs and the observations (including the samples). In both figures, a bimodal distribution of the data loading time can be appreciated, this double distribution comes from the fact that half of the jobs have loaded data from twice the files than the rest. The input

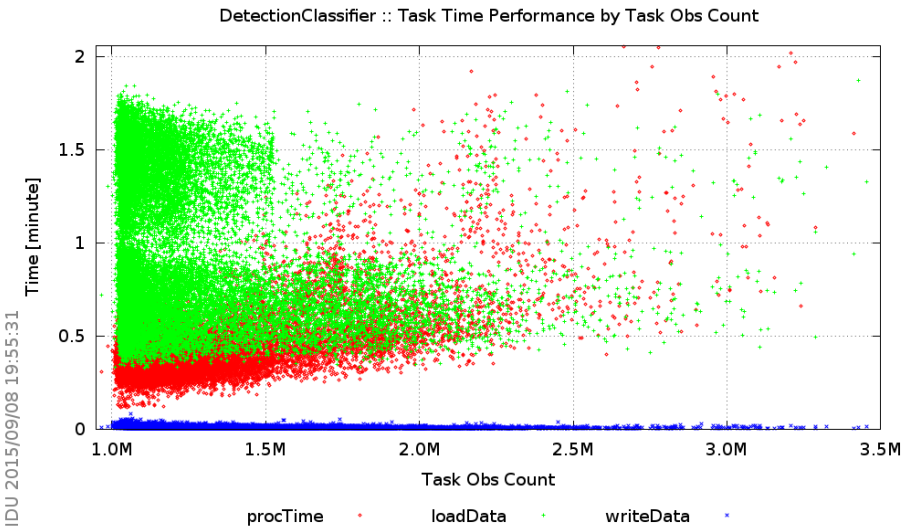


FIGURE 6.55: Job processing and I/O times for the *Detection Classifier* execution with respect the input object logs

data files were partitioned using the same equalisation record limit but due to the gaps the job intervals present small offsets with respect to the file partitioning in some periods where the loaded data have been increased.

Furthermore, the original equalisation of the jobs regarding the input data is somehow lost by the fact that each job is loading extra margins at both ends of the interval which depending on the density of the scanned region could imply big changes in the estimated amount of input data finally loaded. Figure 6.56 shows the final distribution of the observations entering each job including the observations loaded due to the extra margins. In this plot, we can see how in some cases we load more than 1.5 million observations, a factor of 1.5 with respect to the initial limit.

Finally, it must be taken into account that these are only preliminary results and that the computational performance could be highly modified when more sophisticated algorithms are integrated. However, the change in the task performance should not become a major concern compared with other IDU tasks.

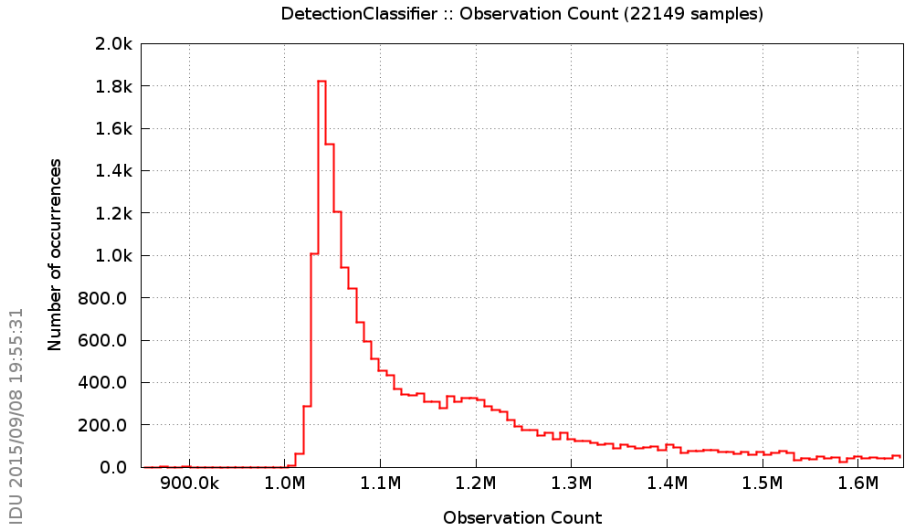


FIGURE 6.56: Job processing and I/O times for the *Detection Classifier* execution

6.4.2 Cross–Match: Detection Processor

The first IDU-XM stage requires three processing steps:

1. Preliminary *Obs–Src Match* processing, with jobs split by equalised time intervals. This step identifies the unmatched observations and provide a first indication of the matching performance for the selected configuration.

2. Processing of unmatched observations for the creation of temporary new sources.

3. Final *Obs–Src Match* processing, where no observation is left unmatched and thus we are ready for the next processing IDU-XM stage.

For the first and last processing steps, we have split again the processing of the full DS-00 extent in time intervals, this time with no more than five hundred thousand observations per job. These jobs then have been also processed in Marenostrum using Greasy in several nodes. The *Obs–Src*

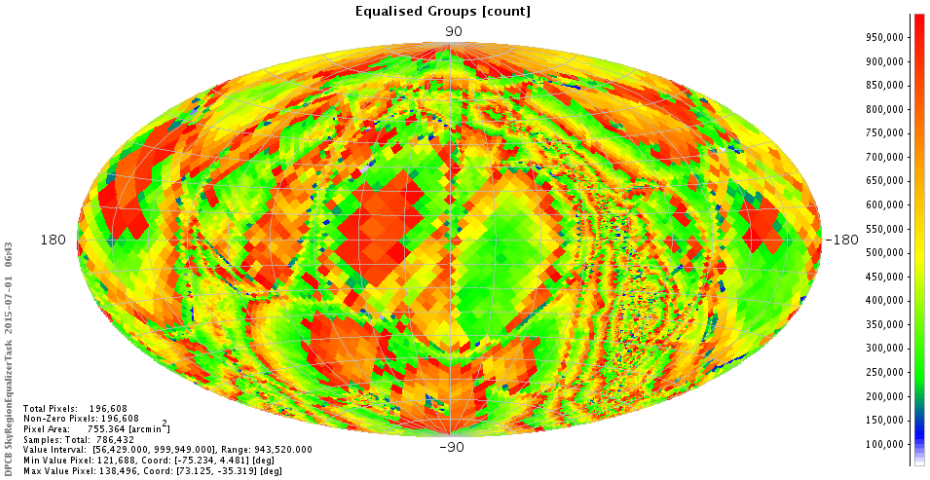


FIGURE 6.57: Groups obtained after the spatial equalisation of the unmatched observations. Note that this figure shows the distribution in the Equatorial coordinate system instead of the Galactic one just because the data HEALPix identifier is computed using that system

Match jobs require more memory since they have to load also the source catalogue and for this reason we could not reuse the time equalisation from the IDU-DC task.

For the processing of unmatched observations, on the other hand, we have defined the jobs according to the spatial density of the resulting unmatched observations obtained in the first *Obs–Src Match*. Figure 6.57 shows the resulting equalised sky regions for the unmatched observations (see Figure 6.23 for the distribution of the unmatched observations).

Tables 6.10, 6.11 and 6.12 summarise the main execution parameters and metrics obtained in the execution of the three processing steps.

Figures 6.58 and 6.59 present the job processing and I/O times for the two *Obs–Src Match* executions with respect the ratio between the loaded input observations and sources. Comparing both figures we can clearly see the following:

- The processing time presents a non linear dependency with respect to the ratio of observations and sources. This behaviour is completely expected taking into account the internal operation of the algorithm.

Task Jobs		44 356
Resources		60 Nodes
		9 Jobs/Node
		1.7 CPUs/Job
Task Wall Clock		7h 21m 25s
Task CPU Time		7062h 40m 00s
Job Wall Clock	Total	3919h 20m 13s
	Avg	5m 19s
	Min	54s
	Max	32m 01s
Output Size	Data	144.4 GBytes
	Logs	5.6 GBytes

TABLE 6.10: Performance metrics of the preliminary *Obs–Src Match*

Task Jobs		22 800
Resources		30 Nodes
		13 Jobs/Node
		1.2 CPUs/Job
Task Wall Clock		9h 37m 30s
Task CPU Time		4620h 01m 00s
Job Wall Clock	Total	3692h 35m 58s
	Avg	9m 43s
	Min	34s
	Max	37m 08s
Output Size	Data	147.0 GBytes
	Logs	15.1 GBytes

TABLE 6.11: Performance metrics of the unmatched observations processing

The *Obs–Src Match* basically takes each observation and computes the distances against a subset of selected sources. This subset of sources is obtained by extracting from the input catalogue a small region around the observation proportional to the configured match radius. Therefore as more sources enters the job, more time is needed for the processing.

- In the first run we can see ratios higher than one (abscissa axis) which basically indicates the presence of unmatched observations – we basically have less source candidates than observations.

- The observation to source ratio is always below one after the processing of the unmatched observations.

Task Jobs		44 356
Resources		80 Nodes 9 Jobs/Node 1.7 CPUs/Job
Task Wall Clock		16h 24m 58s
Task CPU Time		19782h 37m 20s
Job Wall Clock	Total	10874h 07m 48s
	Avg	14m 45s
	Min	01m 31s
	Max	1h 13m 39s
Output Size	Data	584.0 GBytes
	Logs	20.2 GBytes

TABLE 6.12: Performance metrics of the final *Obs–Src Match*

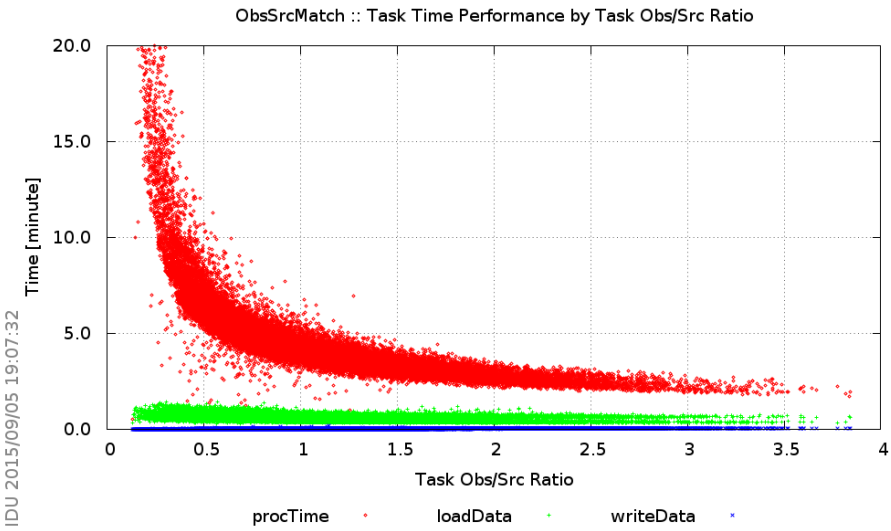


FIGURE 6.58: Job processing and I/O times for the first *Obs–Src Match* processing with respect the ratio between the loaded input observations and sources

- In the second *Obs–Src Match* execution we see an increase of the input data load. This increase comes from the loading of the temporary new sources.
- The processing time, excluding the input data loading, decreases with the observation to source ratio as expected.

From the obtained results, it seems clear that it could be worth equalising this task not only according to the time density but also taking into account

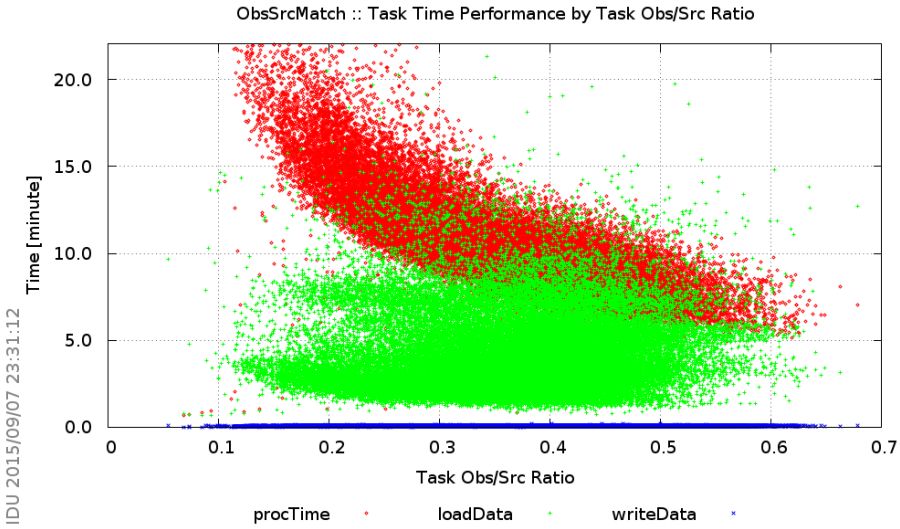


FIGURE 6.59: Job processing and I/O times for the second *Obs–Src Match* processing with respect the ratio between the loaded input observations and sources. This time including the temporary new sources created by the unmatched observations processing which limit the abscissa maximum value below one, indicating that we are now loading more sources than observations

the sky density of the region loaded. This equalisation could be tricky having to process the observation time density and the catalogue spatial density together. However, a very good approximation of this equalisation could be extracted from the results of the IDU-SCN task, which will provide directly the combined density profile.

Regarding the unmatched observations processing, Figure 6.60 shows the time response with respect the input observations. This figure shows that the time used for loading the data is dominating the processing but in this case the execution is affected by the large number of files (about 200 thousand files) resulting from the first *Obs–Src* Match. This time could be reduced by means of a previous file arrangement but it was not considered necessary since this arrangement would have in any case consumed a similar amount of time. Finally, it is also worth noting that the processing time presents a non linear response with respect to the input observations. The non linear behaviour was expected but the task profile has not been yet completely assessed although it should be close to N^2.

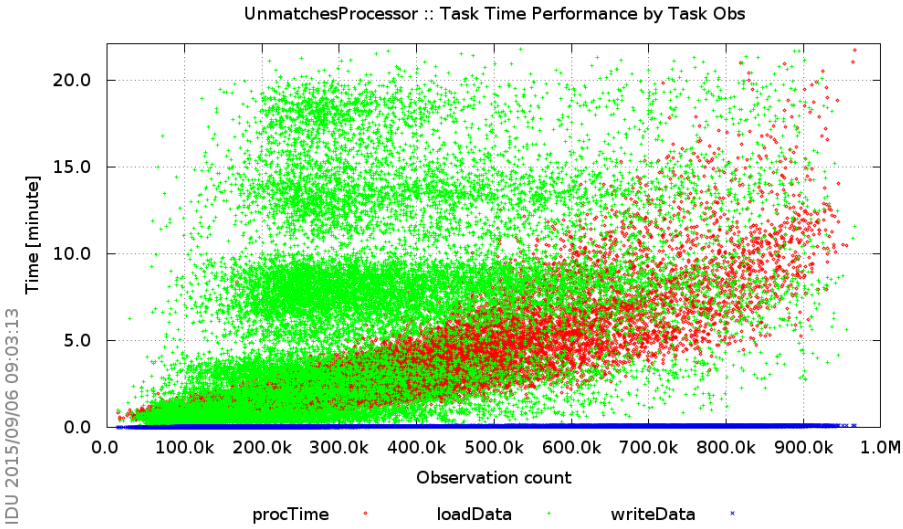


FIGURE 6.60: Job processing and I/O times for the unmatched observations processing with respect the number of input observations

6.4.3 Cross–Match: Sky Partitioner

The *Sky Partitioner* task processes the *MatchCandidates* in HEALPix batches according to their spatial density distribution. This task will rather be performed in a single execution and in general will require two or three iterations for its completion. Figure 6.61 shows an example of the sky distribution of the deferred *MatchCandidates* produced in the first *Sky Partitioner* run. These *MatchCandidates* could not be grouped mainly in the first run because not all observations were contained in the HEALPix region of the job.

Tables 6.13, 6.14 and 6.15 summarise the main execution parameters and metrics obtained in the execution of the three *Sky Partitioner* runs. As it can be seen from the numbers, this task does not demand major computational resources. The I/O is reduced thanks to the adoption of the light *MatchCandidate* interface design and the processing load is also small as no mathematical computations are done. Figures 6.62 and 6.63 show the processing and I/O times for all the *Sky Partitioner* jobs with respect to the task sequence and the number of input observations respectively. In the second figure, it can be seen that the time increases linearly with the

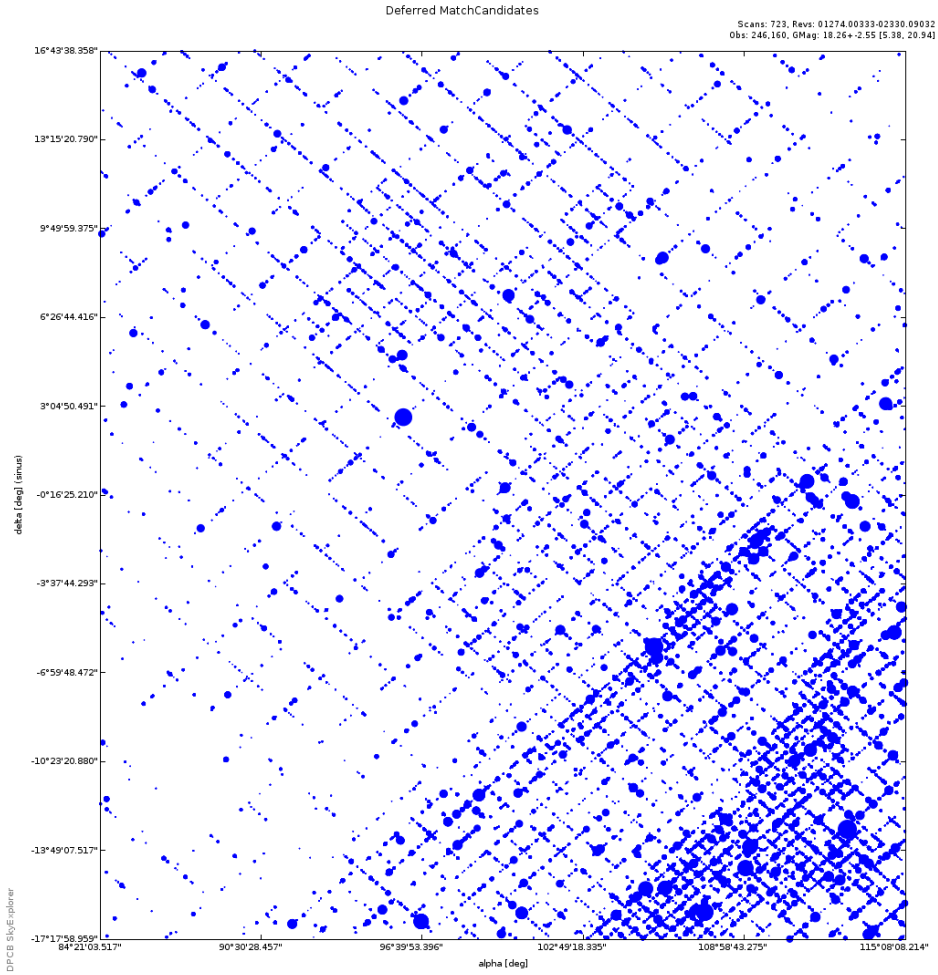


FIGURE 6.61: Deferred *MatchCandidates* in the first *Sky Partitioner* run where the HEALPix boundaries can be clearly seen

number of observations as expected. Also the time devoted for reading and writing are almost the same as expected, since this task is basically grouping the input data without generating additional outputs.

Considering the total number of *MatchCandidates* processed, ∼18 billion, and the total CPU time, we obtain an average performance of 3986 *MatchCandidates* per second (0.251 ms/AO). Performance can also be computed by *MatchCandidateGroup* taking into consideration a total amount of 3 387 925 964 groups: 716.88 groups per second (1.395 ms/group).

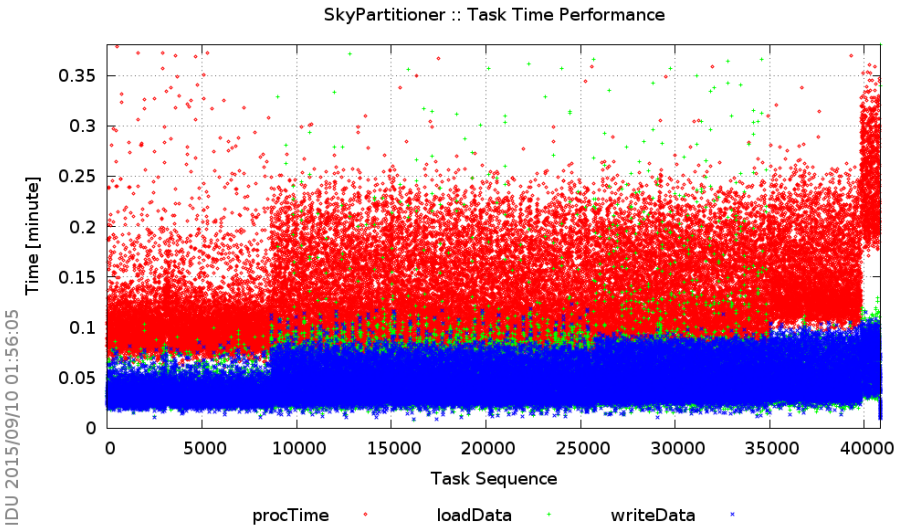


FIGURE 6.62: Processing and I/O times for all the *Sky Partitioner* jobs. The times increase from left to right basically because the job sequence identifier has been assigned according to the volume of input data; that is the equalised region counts

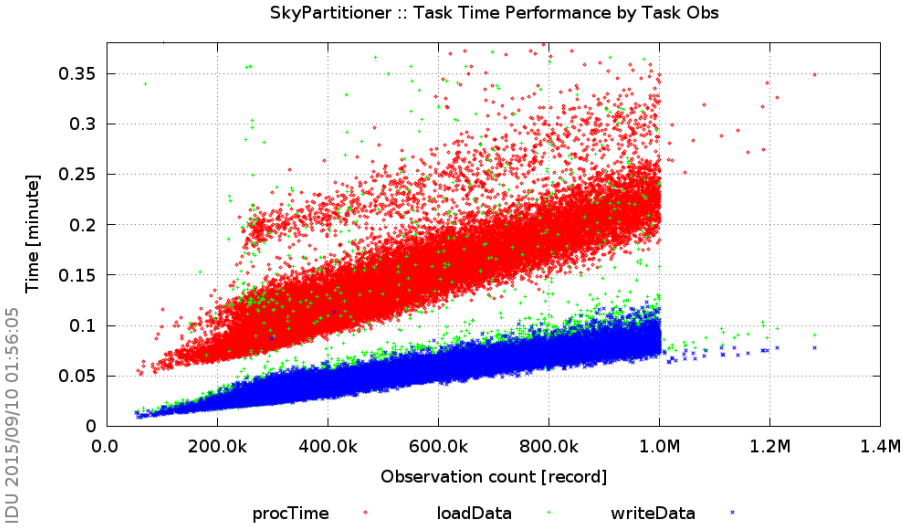


FIGURE 6.63: Processing and I/O times for all the *Sky Partitioner* jobs with respect to the observations input count

Task Jobs		40 821
Resources		40 Nodes
		9 Jobs/Node
		1.7 CPUs/Job
Task Wall Clock		0h 59m 52s
Task CPU Time		638h 34m 40s
Job Wall Clock	Total	181h 12m 47s
	Avg	1m 19s
	Min	6s
	Max	14m 45s
Output Size	Data	406.9 GBytes
	Logs	5.1 GBytes

TABLE 6.13: Performance metrics of the first *Sky Partitioner* run

Task Jobs		48
Resources		1 Nodes
		8 Jobs/Node
		2 CPUs/Job
Task Wall Clock		27m 48s
Task CPU Time		7h 243m 42s
Job Wall Clock	Total	3h 35m 07s
	Avg	4m 28s
	Min	1m 05s
	Max	13m 02s
Output Size	Data	1.0 GBytes
	Logs	59 MBytes

TABLE 6.14: Performance metrics of the second *Sky Partitioner* run performed over the deferred *MatchCandidateGroups* previously grouped according to spatial distribution

6.4.4 Cross–Match: Match Resolver

The *Match Resolver* task is the final resolution step of the IDU-XM. It processes independently each *MatchCandidateGroup* and produces the *Match*, *AmbiguousMatch*, *NewSource* and *Track* tables. Although the current implementation could be distributed by any arbitrary condition we have used the nominal distribution based on a spatial density criterion. This spatial distribution is recommendable for those algorithm implementations that require loading again the input sources for the final resolution of the *MatchCandidateGroups*.

Table 6.16 summarises the main execution parameters and metrics obtained for this task.

Task Jobs		1
Resources		1 Nodes
		4 Jobs/Node
		4 CPUs/Job
Task Wall Clock		17m 57s
Task CPU Time		1h 58m 48s
Job Wall Clock	Total	17m 57s
	Avg	-
	Min	-
	Max	-
Output Size	Data	120.3 MBytes
	Logs	2.9 MBytes

TABLE 6.15: Performance metrics of the last *Sky Partitioner* run performed over the deferred *MatchCandidateGroups*. This time in a single job covering the full sky

Task Jobs		28 977
Resources		30 Nodes
		9 Jobs/Node
		1.7 CPUs/Job
Task Wall Clock		1h 10m 04s
Task CPU Time		560h 32m 00s
Job Wall Clock	Total	292h 49m 38s
	Avg	36s
	Min	19s
	Max	07m 49s
Output Size	Data	450.0 GBytes
	Logs	550.0 GBytes

TABLE 6.16: Performance metrics of the *Match Resolver*

Taking into consideration the total number of *MatchCandidateGroups* generated, a bit more than 2 billions, and the total CPU time required shown in Table 6.16, we obtain an average performance of 963.66 groups per second (1.038 ms/group).

Finally, the only remaining processing task was the execution of the *NewSourceIdConsolidator*. This task is also distributed according to the spatial density of the observations. Its time response is completely dominated by the I/O as it is basically loading data and storing again the same data with the minor updates in the corresponding sourceIds.

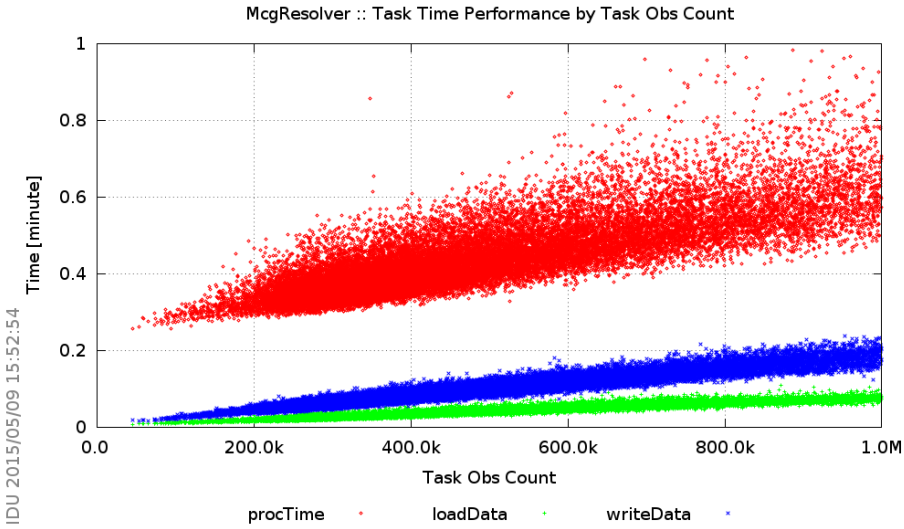


FIGURE 6.64: Job processing and I/O times for the *Match Resolver* task with respect the input observations processed

6.4.5 Summary of the Operational Execution

The computational performance obtained in the operational run presents exactly the same behaviour as the results shown for the second reprocessing. Only minor differences in some total numbers are appreciable, being the profiles in all diagnostics almost the same. Furthermore, the computing resources used in each stage follow a linear distribution with respect to the data volume entering each stage.

Table 6.17 includes the total amount of inputs processed including their corresponding size on DPCB repository.

Data type	**Record count**	**Data Size**
Spacecraft Configurations	16 684	128 MBytes
Object Log	24 681 840 626	221 GBytes
Observation	22 097 513 352	6.7 TBytes
Attitude	15 034	66 MBytes
IGSL Source	1 222 598 530	122 GBytes

TABLE 6.17: Summary of the main inputs involved in the operational reprocessing

The total computing hours consumed for the operational reprocessing, only including the successful run of the final tasks is detailed in Table 6.18.

Task	CPU hours
Detection Classifier	1163
Detection Processor	31 168
Sky Partitioner	1313
Match Resolver	977
Total	34 621

TABLE 6.18: Total computing hours consumed for operational reprocessing, only including the successful run of the final tasks

The total amount of 34 621 CPU hours for the three stages is consistent with the 33 616 CPU hours used for the second reprocessing as expected. In both cases, however the actual CPU hours consumed approaches 100 000 hours, approximately a factor of 3. This factor accounts for the resources used in the preprocessing and arrangement tasks as well as the testing and validation activities performed before each processing stage to check the correctness of the deployment and the software patch releases.

The processing started on 14th of April 2015 and finished on the 9th of June 2015. These were the execution period for each stage:

- First Stage: started on 14th of April 2015 and finished on 11th of May 2015. Almost one month to process the 85% of the data (more than eight months of data).
- Second Stage: started on 31st of May 2015 and finished on the 5th of June 2015. Therefore one week to process the accumulated data from end March to end May, approximately two month of data including the second galactic plane scan. The start of this stage was delayed almost three weeks due to the output transfer backlog accumulated at DPCE and the time spent to clarify and resolve several inconsistencies detected in the received data.
- Third Stage: started on the 6th of June of 2015 and finished on the 9th of June of 2015. The DS-00 was closed on the 3rd of June and we started two days later just waiting for the data to arrive at DPCB. For the last stage we consumed three days for processing the last two weeks of data.

As it can be noted, the processing performance improved from one stage to the next. This was expected, and we already knew that the processing of the first stage would be more costly and tricky. We were dealing for the first time with a very large interval of real Gaia data. Also, several issues were affecting this data period; including IDT and DPCE processing issues, and also some particular Gaia spacecraft operations as the decontamination activity in September 2014.

The data results were sent to DPCE in three separate transfers at an average rate of 255.3 Mbps. Tables 6.19, 6.20 and 6.21 summarize the main statistics of these transfers. For more detailed information of the data transfer system see Appendix C.

Start Time	2015-06-10 15:38:02
End Time	2015-06-10 19:41:57
Elapsed Time	4h 3m
Data Transferred	426.4 GB
Average rate	250.3 Mbps

TABLE 6.19: Statistics of the first stage transfer to DPCE

Start Time	2015-06-11 07:54:27
End Time	2015-06-11 09:18:38
Elapsed Time	1h 24m
Data Transferred	152.3 GB
Average rate	259.0 Mbps

TABLE 6.20: Statistics of the second stage transfer to DPCE

Start Time	2015-06-11 11:47:02
End Time	2015-06-11 11:52:40
Elapsed Time	5m 38s
Data Transferred 10.6 GB	
Average rate	270.1 Mbps

TABLE 6.21: Statistics of the last stage transfer to DPCE

Regarding the data volume, we consumed a total amount of 5.9 TBytes, more specifically:

- First Stage: up to 3.5 TBytes of intermediate data and logs for producing a final output of 426.52 GBytes.

- Second Stage: 1.5 TBytes of intermediate data and logs. 152.28 GBytes for the output data.
- Third Stage: 35.67 GBytes of intermediate data and logs. 10.63 GBytes for the output data.

The intermediate data corresponds mainly to all the temporary data produced by the IDU-XM; *MatchCandidates*, *MatchCandidateGroups*. Also a 20% of this intermediate data corresponds to information related to the solutionId and the input data used of each task job. From this activity we have learned that this intermediate data could increase very quickly and we have already taken some corrective actions to reduce and compact all this data.

6.5 Conclusions

We have recomputed the Cross-Match for the full DS-00 extent. This execution has been the very first DRC level run over real data of DPAC. Executing IDU at DPCB over that amount of data for the first time has been a challenging task. The activity has been completed, with minor deviations, following the envisaged schedule and fulfilling the resource estimations. No major/critical issues have been identified during the execution of the software nor in the obtained results.

The successfully accomplishment of this early execution of the IDU tasks are one of the main achievements resulting of all the work done within this thesis. In addition, it must be remarked that in the second execution we have been able to process the full DS-00 extent in less than two weeks – excluding the data arrangements and without the dependency on the data availability from DPCE suffered in the first run. This achievement is very important since we have demonstrated that our system, IDU, is already capable of handling the processing of the real Gaia data. This has been possible thanks to all our work in the execution framework, data analysis and equalisation of the jobs described in this thesis in Chapter 5.

Below, a brief analysis of each task is presented.

DetectionClassifier

The *DetectionClassifier* task has behaved as expected. The classification has improved compared to the previous results provided by IDT as a newer version of the algorithm has been used. Even between each stage, there have been some additional improvements on the algorithm and configuration changes driven by the results of the previous stages.

Even though the *DetectionClassifier* task has removed more spurious than previously done in the IDT daily processing still a significant number are left undetected and have entered the IDU-XM. This can be seen in the results presented in Figures 6.10, 6.11 and 6.12. Additionally Figures 6.13 and 6.14 show the spurious detections from SSOs transits which were not anticipated and no mitigation was implemented yet. The causes for the remaining spurious detections are known and the algorithm is still under development to improve the spurious identification efficiency.

The analysis of the IDU-XM results has allowed the identification of new types of spurious detections previously unknown or not yet seen. These include spurious detections caused by SSOs, spurious detections on the other FoV for very bright objects and several other causes.

The task did not show any performance issue. Job equalization has allowed the task to run in very short wall clock times. CPU consumption per detection is also within acceptable ranges. Section 6.4.1 provides a detailed analysis and further information.

Detection Processor

This task has been executed as expected. It has required a significant amount of operator time for the executions. This is because it is the first task in the sequence that has to deal with all the issues of the input data for the first time, mainly related to the attitude.

The process of dealing with the features in the data (gaps, different attitude quality, processing inconsistencies upstream, satellite activities, etc.) has

been slow. Mainly because these events are not yet consistently reported in a centralized way. Also, there is currently no way to input this information to the processing software. Once tasks were adjusted to the input data problems and features (filtering and discarding some ranges) the jobs run correctly.

The task has produced consistent results, as reported in Section 6.3.2. This task has also been affected by the presence of spurious detections. In particular, Figure 6.28 shows that the match distances are affected by tails and diagonal detection lines produced by the remaining spurious.

Further analysis is being performed to understand the IGSL match ratio and distribution reported and presented in Figure 6.32.

The task has no performance problems. Job equalization has allowed the task to be distributed without difficulties in the computing nodes. However, this is the task that has required far more CPU hours than the rest of the tasks combined. In part, this is due to the fact that it is loading all the observations (including the samples) and the catalogue sources. In subsequent tasks only a minimum set of fields are kept, therefore reducing the access time and memory footprint

CPU hours consumption per observation is within the expected ranges. Section 6.4.2 provides a detailed analysis and further information.

SkyPartitioner

The *SkyPartitioner* task has run without any deviation and its execution has been easy as all the problems with the input data have been absorbed in the previous task.

A detailed study of the *MatchCandidateGroup* created by the task has been performed. This included distributions by number of observations and sources as well as the sky distribution. After the execution of the task several off line tasks were executed to extract singular cases from all the data. This has allowed the detection and analysis of the Gaia detection response for extended objects and Hipparcos bright sources. Also, it

has been useful to identify new spurious detection configurations. Some examples are shown in Section 6.3.3.

The task has no computational performance bottlenecks. Jobs are equalized by HEALPix batches according to the sky density and incomplete groups (close to region boundaries) have been resolved in subsequent jobs at higher HEALPix levels as envisaged. Therefore, several runs of the task are usually required to complete all the isolated groups. This equalization has allowed the task to be distributed without difficulties in the computing nodes.

CPU hours consumption is within the expected ranges. Section 6.3.3 provides a detailed analysis and further information.

Match Resolver

The *Match Resolver* task is the final resolution stage of the IDU-XM. This task provides the final resolution of the *MatchCandidateGroups* creating the *Match*, *AmbiguousMatch*, *NewSource* and *Track* tables. As noted in Castañeda et al. [2015], in this particular case no merge or split operations can be performed by the algorithm. Therefore, no Track table was generated.

The algorithm used for this stage is comparable to the one used in IDT. Section 6.3.4 provides an analysis of the task results. In summary, the resolution is highly affected by the remaining spurious detections that dominate the creation of new sources. From the investigations carried out it is clear that roughly 70% of the new sources have been created from observations fainter than 20^{th} magnitude with less than seven observations. The small number of observations is relevant as it is below the 90 percentile of the number of scans in each *MatchCandidateGroup* (nine scans as shown in Figure 6.44). This fact could be an indication that these sources could have been created from spurious detections not properly identified.

The large number of spurious sources created may have strong implications for the final catalogue. Mainly, the created catalogue could be polluted

with a high number of sources coming from spurious detections and unfortunately these sources would remain in the catalogue forever according to the baseline defined in Bastian [2013]. This baseline specifies that no sources are deleted from the catalogue but just left with zero matches. It is clear that this baseline did not anticipate the large number of this kind of sources and from our point of view the baseline should be updated to allow their deletion in IDU.

In terms of performance the task has required very few resources. Jobs are equalized by batches of *MatchCandidateGroups* which can be resolved independently one from another. This equalization has allowed the task to be distributed without difficulties in the computing nodes. As the used algorithm does not require the input catalogue again, the I/O load is small and the resolution is quick. Memory requirements are also low as no sources have to be loaded. Therefore, the task has consumed less resources than what is expected for a nominal IDU-XM resolution stage. Section 6.4.4 provides a detailed analysis and further information.

6.5.1 Recommendations

The aim of this activity was to consolidate and improve the IDT Cross-Match results for DS-00. The activity has indeed provided an improved Cross-Match solution despite the issue described in Section 6.3.5. However, the amount of new sources created due to spurious detections is still a problem. For this reason, the IDT, IDU and DPCB teams proposed the adoption of the following mitigation actions in June 2015:

1. Filter the faintest new sources in the forthcoming MDB Integrator run.
   Specifically, new sources with magnitude fainter than 20^{th} magnitude having less than seven observations linked. This is about 70% of the newly created sources.

2. For the next IDU-XM and MDB Integrator run after DS-01, do one of the following:

- Remove all new sources no longer matched, mainly due to better spurious detection filtering.

- Start again with only the sources from the original IGSL catalogue, discarding again all the new sources created so far.

Additionally, the following actions are proposed to reduce the catalogue pollution in the daily pipeline:

- Update IDT to avoid creation of new sources for unmatched observations fainter than 20.3 (about 50%)
- Include all the resulting unmatched observations as *BlackListed* transits (flagged accordingly). These observations will be reintegrated if considered appropriate by IDU.

These recommendations were forwarded to DPACE for consideration one week after the DS-00 but unfortunately none of the recommended actions were taken into account so all the new sources have been integrated and IDT has continued polluting the catalogue with these spurious new sources. Hence, IDU will have to deal with a very polluted catalogue in the next run.

7 Conclusions and Future Work

Gaia requires a demanding data processing system on both data volume and processing power. The treatment of the Gaia data has been designed as an iterative process between several systems each one solving different aspects of the data reduction system.

In this thesis we addressed the design and implementation of the Intermediate Data Updating (IDU) system. The IDU is the instrument calibration and data processing system more demanding in data volume and processing power of DPAC. Without this system, Gaia would not be able to provide the envisaged accuracies and its presence is key to get the optimum convergence of the iterative process on which all the data processing of the spacecraft is based.

The design and implementation of an efficient IDU system is not a simple task and a good knowledge of the Gaia mission is fundamental. The initial chapters of this thesis have described the essential aspects of the spacecraft, the instrument and the overall data reduction system of Gaia. These chapters have shown consequently the depth of understanding acquired during this thesis. This knowledge has been fundamental when implementing the scientific algorithms and building an efficient IDU system. The main tasks included in IDU were presented in Chapter 4 whereas the implementation details were deeply discussed in Chapter 5.

This design and implementation work is not only referring to the actual design and coding of the system but also to the management and scheduling of all the related development tasks, system tests and in addition the coordination of the teams contributing to this system. Since the very beginning of this thesis, a lot of work has been devoted to chase the continuous changes in the instrument and the processing algorithms affecting ultimately the final design of IDU. This circumstance is clearly evident in the contents of this work and it has led to the implementation of a modular and very flexible system. Thanks to this flexibility and modularity, the system can be easily adapted and extended to cope with the changes on the operational requirements. A clear example is the implementation and integration of the IDU-DC task described in Section 4.2.

In addition, the IDU implementation presents a variety of interesting challenges; covering not only the purely scientific problems that appear in any data reduction but also the technical issues for the processing of the huge amount of data that Gaia is providing. The handling of this data volume has also been one of the main topics covered in the present work. Within DPCB, we have developed several tools to make this task easier; including tailored data access routines, efficient data formats and an autonomous application in charge of handling and checking the correctness of all the input data entering to this DPC as well as for checking and transferring of the IDU outputs to DPCE. These additional topics have been included as appendixes to this thesis.

Finally, we have had the chance during this last year to test and demonstrate how all the work done in the design and implementation of IDU is more than capable of dealing with the real Gaia data processing. We have basically executed the IDU-DC and IDU-XM over the first Data Segment (DS) of the mission. This execution has been the very first DRC level run over real data of DPAC. Executing IDU at DPCB over that amount of data for the first time has been a challenging task and has additionally provided valuable information for the improvement of the current IDU implementation.

Below, a brief summary on the foremost contributions and findings of this thesis is presented. We also identify in each case the current work being done and the future work envisaged.

Execution Framework

In Chapter 5 we have described in detail the design and implementation of the IDU system which constitutes the main goal of this work. The implementation of an efficient and versatile execution framework for the IDU tasks has been one of the most challenging tasks accomplished. The IDU design has not been only driven by the scientific requirements but also by the characteristics and restrictions of the execution environment and resources – Marenostrum supercomputer hosted by the Barcelona Supercomputing Center (BSC).

The framework we have developed can be run either in a common personal workstation or in hundreds of nodes of any computer clusters. For the second case, the framework relies on a BSC tool to distribute the task jobs in the computer nodes. This tool solves in a quite simple manner the issue of distributing the jobs but we have added additional functionalities on top to have more control of several distribution related topics as the job prioritization and the capability of assigning the computing resources individually for each job in runtime. These functionalities are very useful for being able to exploit efficiently the BSC resources. Due to the characteristics of the Gaia data and the IDU processing the perfect equalisation of the jobs is not always feasible and therefore this capability of assigning the resources dynamically is essential.

These extended functionalities are already operational but they have not yet been used for any operational activity. Trial runs are scheduled for end 2015 and as soon as the stability and robustness is confirmed, this extended framework will be used for all operational activities. In that sense, this migration is one of the foremost improvements and steps forwards for the nearest future.

Detection Classifier

As commented in Section 4.2, the impact of the spurious detections on the Cross-Match is an issue recently identified. A lot of effort has been dedicated for the development of models to properly separate the spurious detections from the real ones. However, the results obtained so far indicate that much work is needed to obtain a data set free from these detections.

From the results obtained in the first run of IDU-DC presented in Chapter 7, new kinds of spurious have been identified e.g. the spurious detections around bright SSOs and the detections of diffuse objects. Additionally, this exercise has provided valuable information about the statistics of all these kinds of spurious detections and this better knowledge of the problem has already implied major improvements on the current model. The more recent updates in IDU at the time of writing these conclusions has been the treatment of the major planets surroundings and the tracking of the VBSs. Only with these two additions, only possible in IDU because of the availability of the IDU-SCN results, millions of detections will be blacklisted thus contributing to a less polluted source catalogue.

However, the combination of the detection classifications done by other CUs and the implementation of a 2D treatment of the detections will remain pending. This last one may introduce new dependencies on the IDU-DC task namely the attitude, having to compute the sky coordinates of all the detections although we are also studying the possibility of doing this analysis as an intermediate processing step before the resolution of the Cross-Match where the sky coordinates are already available.

LSF/PSF integration

The LSF/PSF calibration is one of the more complex processes included within IDU. This task together with the Image Parameters Determination (IDU-IPD) task and with their successive iterations with AGIS and PhotPipe is what will make it possible to achieve the high astrometric accuracies envisaged for the final Gaia catalogue.

The calibration of the instrument LSF/PSF response has been since the beginning one of the more challenging issues of the Gaia processing system. This calibration has to deal not only with the variation with time of the optical properties of the mirrors but also with the image distortion introduced by the detectors. The correction of these electronic distortions, dominated by the CTI, has not been yet completely addressed. It is envisaged that more sophisticated models will be introduced in IDU in the coming years when this distortion will progressively increase and degrade the overall astrometric results. All these facts are evident in Section 4.6 where we have summarised the development history of the IDU-LSF/PSF task. At the time of writing, the latest Empirical LSF/PSF (ELSF) implementation was integrated in IDU and the corresponding scientific and performance tests were started.

IDU-AGIS Loop

To assure the readiness of all the systems involved in the cyclic data processing, DPAC has defined and exercised several testing campaigns during the last years. These tests have been exercised over simulated datasets and currently over the real data from the spacecraft. We can distinguish two kinds of tests:

- Global Operation Rehearsals focused on testing the interfaces and enforcing a common development road map for all systems and DPCs.
- Test campaigns of specific processing concepts and systems.

We have participated actively on the definition and execution of these test campaigns covering the DPCB operation and the execution of the IDU system. We have also provided support to the daily activities at DPCE, mainly for IDT monitoring and software maintenance.

A special test campaign is the IDU-AGIS test, focused on the proof of the astrometric iterative reduction concept by running several times both systems and analysing the convergence and stability of the resulting astrometric parameters. Unfortunately, the operation of the daily systems and

the handling and analysis of the results obtained during this first year of mission has required more effort and manpower than expected and these test campaign are highly delayed. We have already completed the first test stage running IDU and shortly AGIS data should be made available. Ideally, this test should be finished before the beginning of 2016 when the first IDU-LSF/PSF and IDU-IPD task run is scheduled.

The successful completion of this test is currently one of the main priorities in the IDU roadmap for 2016.

Autonomous Monitoring and Validation

The autonomous validation and monitoring of the IDU outputs is still an ongoing task. We have developed and implemented several tools for this specific purpose which have been essential for the analysis and understanding of the Gaia data and for the improvement of the IDU task algorithms. These tools have been described in Section 4.8 and they have been used extensively for the presentation of most of the IDU results included in this thesis. A quite complete set of these diagnostics have been included in Chapter 6.

Although these tools are already integrated in all IDU tasks, a centralised system must be implemented to provide an overall and quick view of the system performance. We have already contributed to the implementation of a similar system for the daily pipeline running at DPCE, specifically in the IDT system. However for IDU we have a more ambitious goal. Contrary to IDT where the results can be easily monitored on a daily basis, the high parallelism of the IDU execution requires a more sophisticated solution to handle the large amount of results and at the same time guarantee the quality of the results. This system should also produce summary reports to the operator for inspection.

The first approach will be based on the definition of several reference data sets. These sets will be produced before the real task execution and they will be deeply analysed by the corresponding scientists. Afterwards, these reference sets will be compared with the operational outputs during the

IDU execution. This comparison will then reveal any deviation on the task execution or configuration thus assuring the proper deployment of the task. These reference sets, in some cases will be extracted directly from previous DRC exercises thus obtaining information regarding the evolution of the task results. These kinds of evolution diagnostics are quite useful for the analysis of tasks like the IDU-DC and the IDU-XM where better and clearer results are expected due to the increase of the number of observations.

In addition to these static predefined references, external consistency checks will also need to be performed as for the case of the IDU-LSF/PSF and IDU-IPD where the residuals of the estimated locations will be compared against the AGIS astrometric solution.

Finally, we think that the work presented has largely contributed to the evolution and definition of the Gaia data reduction system and its operation. We are also confident that the system, we have designed and that constitutes the bulk of this thesis, is ready to cope with the Gaia data according to the requirements set. Furthermore, the presented design provides a solid IDU system foundation for the challenging task of processing the Gaia data during the forthcoming years.

A DPCB Overview

Data Processing Center Barcelona (DPCB) is embedded in the Gaia DPAC group at the Universitat de Barcelona (UB), in close cooperation with the Barcelona Supercomputing Center (BSC) and the Consorci de Serveis Universitaris de Catalunya (CSUC), also in Barcelona, Spain. More specifically, the DPCB hardware used in operations is provided by BSC, whereas the team at the UB carries out the management, operations, development and tests of the software.

The main DPCB responsibilities are the execution of:

- CU1-GTS, described in Appendix C.
- CU3-IDU, covered in Chapter 4 and Chapter 5.
- CU2 Simulations: CU2-GASS and CU2-GOG.

Additionally, CU3-IDT and other related products are also developed and tested within DPCB, mainly in CSUC resources. The complete list of DPCB responsibilities are document in Portell et al. [2014b] and Castañeda et al. [2013].

Focusing on the operational activities, DPCB is part of the DPAC cyclic processing and receives data on a daily basis from DPCE (as described in Appendix C.1) and runs the several stages of IDU (IDU-SDM, IDU-CAL and IDU-IPD) every DRC (see Chapter 4). Depending on the inputs

available, particularly at the beginning of operations, only some IDU subsystems might run. On later stages of the mission, repeated executions of these subsystems may be needed during a given cycle.

Regarding the simulation activities, DPCB has generated most of the CU2 simulation datasets for development and testing of the whole DPAC products. CU2 simulations have been essential prior to Gaia launch to test DPAC daily processing software and will still be used, even after launch, to test the cyclic processing chains. Currently simulations are still being generated for the CU9 software validation and testing. These simulations are essential for the preparation for the first Gaia catalogue release.

A.1 Hardware Resources

The DPCB has access to computing resources at the BSC, and at CSUC, for operational and testing activities respectively.

A.1.1 Barcelona Supercomputing Center (BSC)

BSC is a public research center located in Barcelona, Catalonia, Spain. It is managed by a consortium composed of the Spanish Government, the Generalitat de Catalunya and the Universitat Politècnica de Catalunya (UPC). It hosts one of the most powerful supercomputers in Europe, called Marenostrum.

The main resources provided by BSC are [BSC, 2015]:

Marenostrum

At the time of writing, Marenostrum offers a peak performance of 1,1 Petaflops and 100.8 Terabytes of main memory and is composed of 3056 computing nodes. Each node offers a peak performance of 332.8 Gigaflops, with 16 cores of Intel SandyBridge–EP E5–2670 processors (2.6 Ghz), 32 GB of RAM and 500 Gigabytes (6Gbps) local disk.

They are interconnected using a point–to–point fiber optic network (Infiniband 10 Gigabit).

The last full upgrade took place beginning 2013. However, during 2014 and 2015, a memory upgrade was done over 256 nodes: 128 nodes to 64 GB and 128 nodes to 128 GB.

Storage

The main file system of Marenostrum, build upon GPFS, provides a total capacity of 2 PB, offered globally to all the nodes and providing a parallel access through a 10 Gigabit Ethernet.

Besides, a long-term storage is available, offering about 4 PB, which will be used for long-term storage of IDU input data and also to store the output data from each DRC.

Next full upgrade may take place around 2016-2017 but an agreement with the Spanish Government and BSC guarantees the computing resources for Gaia so an adequate portion of the current resources may be kept (Figure A.1) in the unlikely scenario that BSC stops offering a general purpose CPU architecture (i.e. migrate to Cell or GPU processors) where no JVM implementation is available which is mandatory for running DPAC software.

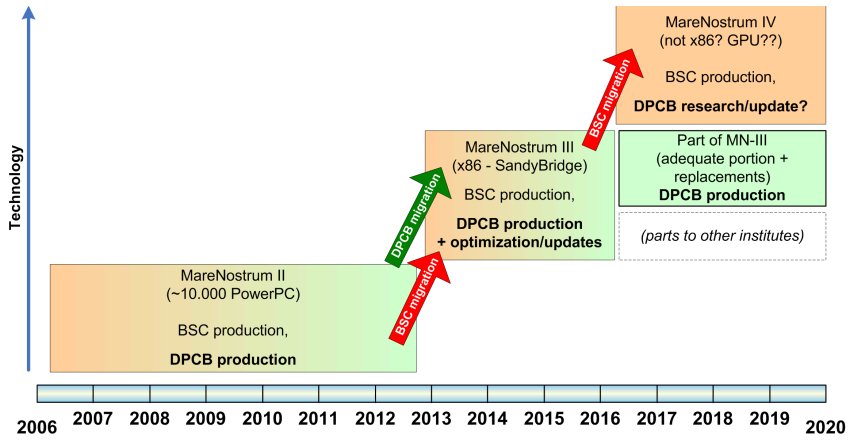


FIGURE A.1: BSC resources upgrade planning and service procurement and migration strategy for DPCB operations

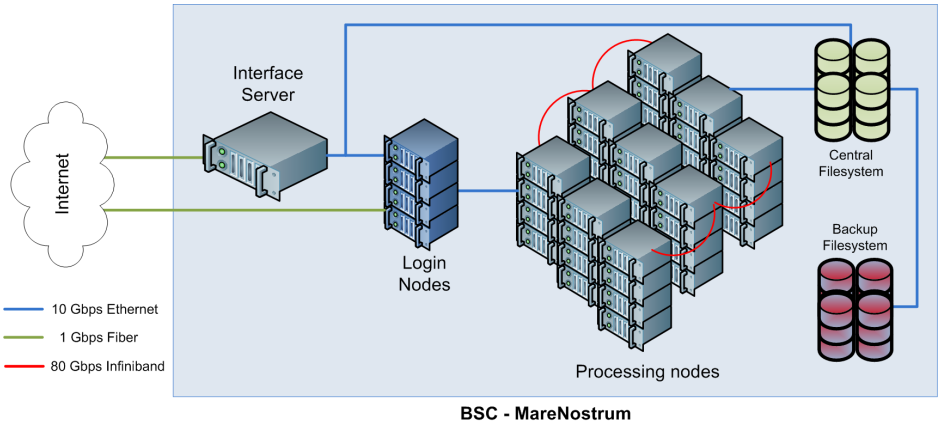


FIGURE A.2: Overview of hardware resources at BSC

Figure A.2 provides an overview of the hardware resources available at BSC.

Marenostrum has been largely used for the generation of CU2 simulation and it will be running IDU for the Gaia data reduction processing.

It is important to note that this machine is not under exclusive DPAC control, we do not have permanent dedicated resources assigned. Instead DPAC systems running on Marenostrum share the computing resources with other users. To access the system, users log into one of the login nodes, from where they can submit jobs to the job management system. The system used to manage jobs is the commercially IBM-LSF.

DPCB also owns a dedicated server integrated into the BSC for the integration of the GTS. This server is also used for the monitoring of the operational activities. It basically collects and provides reports on the jobs and system status. The Gaia team at the UB provides the hardware (described in Appendix A.1.3) and is in charge of the implementation of the necessary software to cover the previous commented functionalities.

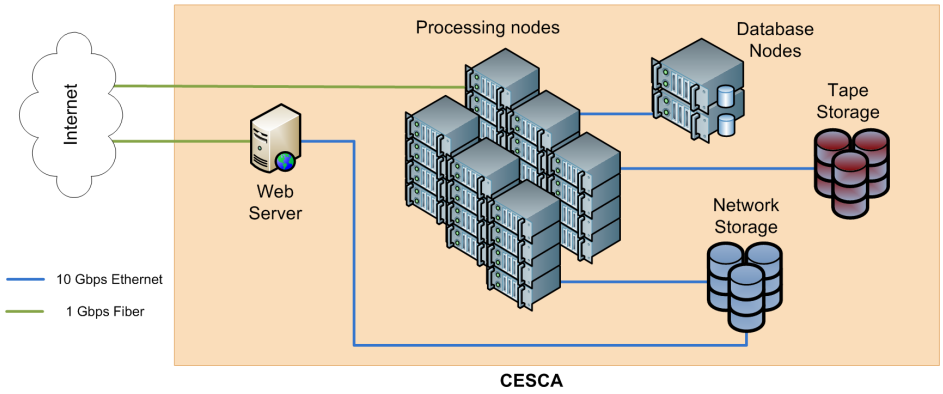


FIGURE A.3: Overview of CSUC hardware resources

A.1.2 Consorci de Serveis Universitaris de Catalunya (CSUC)

The Consorci de Serveis Universitaris de Catalunya (CSUC) provides services to public and private universities, research centers and institutes, libraries, and other entities which participate in R&D and innovation projects. It is integrated by the Generalitat de Catalunya and ten Catalan universities (UB, UAB, UPC, UPF, UdL, UdG, URV, UOC, URL and UVic-UCC).

CSUC provides several hardware resources as shown in Figure A.3, being the following ones the most relevant:

Prades

Xeon 8-core cluster with 45 nodes, each with 2 quad-core Intel Xeon processors. 29 of the nodes have Xeon E5472 at 3 GHz and 32 GB RAM, while the other 16 have Xeon X5550 at 2.66 GHz (identical to those used for IDT operations at DPCE) and 48 GB RAM. The total processing performance is 2.68 TFLOP.

Pirineus

Altix UV cluster, a shared-memory system with 1344 Intel Xeon X7542 cores (2.66 GHz) and 6056 GB RAM, offering about 14 TFLOP.

It is used for CU2-GOG simulations for CU9 and specific scientific simulations related to Gaia.

Empedrat

It is a single node, but with a total of 48 cores and 64 GB RAM. Currently this node hosts the Cache DB system for IDT tests.

Graftonita

This server is based on a Linux Virtual Machine. It offers public HTTP/FTP services for the distribution of data generated at DPCB (either CU2 simulations or CU3 test data) to other DPAC teams.

A.1.3 Interface Server

The current server specifications are:

- OS: SUSE 12.2 *Mantis* - Kernel 3.4.28 x86 64
- RAM: 16GB
- Processors: 8 Intel(R) Xeon(R) CPU E5620 @ 2.40GHz
- HDD: 4.5 TB RAID 5
- Gigabit Ethernet network card

A.2 Software

A.2.1 DpcbTools

The DpcbTools is intended to provide a common software toolbox in Java to be used by the Gaia DPAC community for the implementation of data processing tasks in the DPCB-BSC environment. This toolbox is being developed to be able to exploit at the maximum level the resources of the BSC environment and to provide a common processing infrastructure to

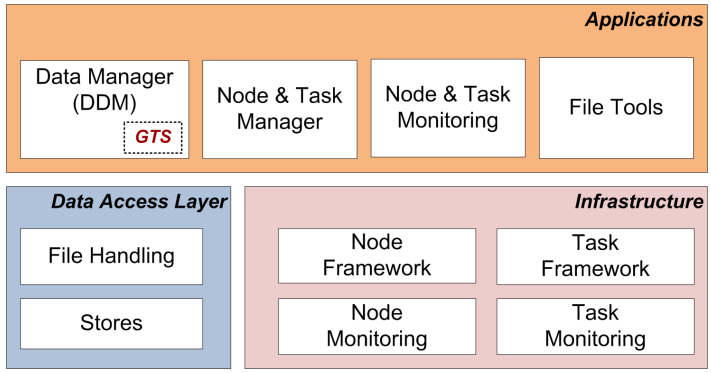


FIGURE A.4: Decomposition of DpcbTools

the DPAC software engineering teams (Castañeda et al. [2011d] and Fries et al. [2011]).

This toolbox implements (as shown in Figure A.4):

Data Access Layer

Set of routines providing simplified access to data stored in BSC file systems. It also provides the tailored HDF5 file format, and a *stores* interface for the implementation of data servers and data caches. These *stores* exploit the Infiniband network for inter-node connections and the Gigabit Ethernet for the communication with remote file systems by the implementation of optimal and simplified communication interfaces.

Infrastructure

Collection of frameworks for the handling of the computing nodes and their tasks.

- Job/Task scheduling and management; interacting with the Marenostrum queue system (IBM-LSF [IBM, 2015]).
- Monitoring Tools for Job/Task/Node operation and status.

Applications

Additionally, DpcbTools implements several end-user applications:

- DPCB Data Manager (DDM), main interface between GTS and the DPCB storage resources (see Appendix C).

- Web based front-ends for the monitoring and management of nodes and tasks.
- File tools for the extraction of data statistics, data filtering, data conversions, etc.

B DPCB File Format

Contrary to most DPAC data processing centers, DPCB will not use databases but operate exclusively with files. Furthermore, the DPCB file system is shared with other users which additionally introduce I/O constraints and performance limitations.

The current DPAC file standard GBIN was designed as a deployment format. The GBIN file consists basically of serialized Java classes compressed using a ZIP algorithm. This implies that decompression is mandatory before the program is able to access any information from the file contents. Although, the GBIN has some meta–data information and data is stored in chunks, this full chunk has to be loaded for the proper de-serialization of the Java classes. In other words, file contents has to be 'fully loaded', without regard for which part of the information is required. Additionally, ZIP is very CPU intensive.

To tackle some of these issues, an HDF-based file format has been developed for DPCB. Hierarchical Data Format (HDF) is a set of file general binary formats designed to store and organize large amounts of numerical data. Originally developed at the National Center for Supercomputing Applications, it is supported by the non-profit HDF Group, whose mission is to ensure continued development of HDF technologies, and the continued accessibility of data stored in HDF. HDF is used for very large datasets, fast access requirements and very complex datasets. Additionally HDF

provides useful built-in features like *chunking* and several data compression solutions.

Another inherent limitation in the GBIN files not commented before is that you require a JVM to read them. The HDF data format can be accessed through multiple interfaces: C, Java, Python, Matlab, etc. Also, the HDF data format is architecture independent as HDF library already handles endianess problems and conversions transparently.

The HDF format is similar to XML documents in the sense that the HDF files are self-describing and allow users to specify complex data relationships and dependencies. However, HDF files can contain binary data and allow direct access to parts of the file without first parsing the entire contents.

B.1 HDF5 Implementation

The HDF Group already provides some HDF-Java wrappers, including a Java browser for HDF files. The first integration of HDF5 in DpcbTools was based on this Java library [Portell et al., 2011]. However, this implementation was insufficient because these Java wrappers do not provide all the functionalities of the native library. The latest implementation uses directly the native HDF5 routines through a Java Native Interface (JNI) layer.

The most significant part of the data volume handled by IDU are in form of raw astrometric observations. These data are almost accessed by all IDU tasks and thus the efficient storage and access to this data is more than desirable. For this reason, the HDF implementation was initially only supporting this DM interface. The corresponding HDF5 data definition follows the Java DM interface, defining the same fields but grouping them in separated data sets – basically separating the raw window samples from the rest of the fields. This grouping is fundamental to take full advantage

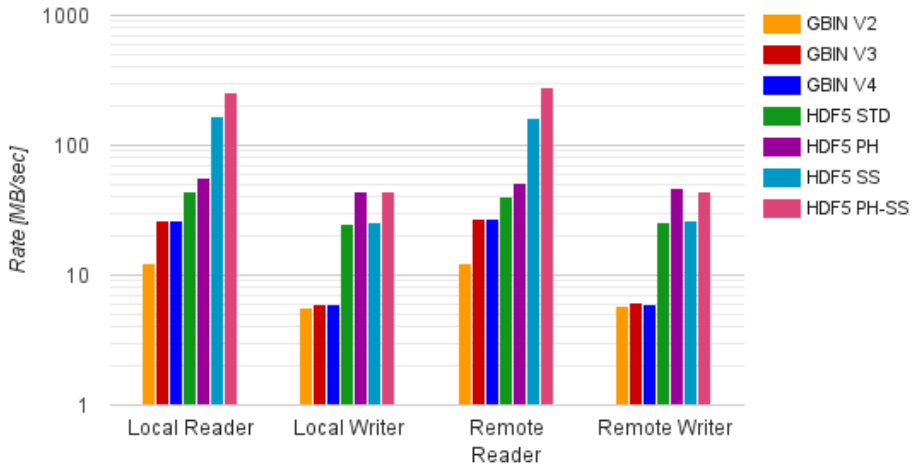


FIGURE B.1: HDF5-GBIN File format performance comparison.

of the HDF5 available features; selective read of the file contents and the application of specific compressors for each type of data.

For this specific implementation, we are getting a better compression ratio and better I/O performance than GBIN as shown in Figure B.1 and Figure B.2. In the figures, we show the results of four different configurations of our HDF5 format:

- Standard (STD): standard configuration.
- Primary Header (PH): where only the samples are compressed.
- Skip Samples: (SS): only reading the header fields, ignoring the observations samples.
- PH-SS: combining the previous two configurations.

For other data types a more generic HDF5 data format has been implemented where the HDF implementation is simply storing the binary stream of the Java serialisation. In this case, the only benefit may come from the possibility of using better compression solutions.

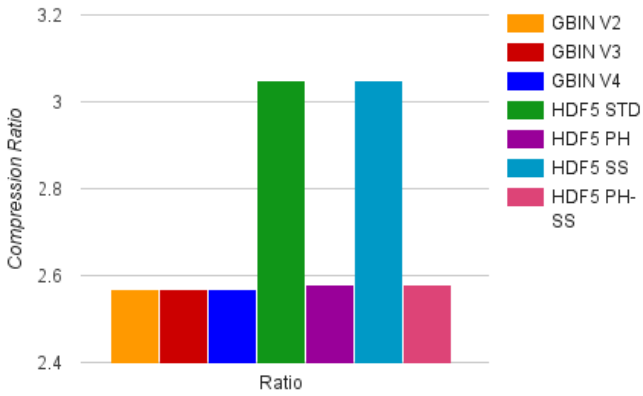


FIGURE B.2: HDF5-GBIN File format compression ratio comparison

C Gaia Transfer System

This appendix describes the GTS which is the service that permits data exchange between the DPCs involved in the Gaia data reduction processing. The data exchange plays an important role in ensuring compatibility amongst all groups involved in the Gaia data processing. The data transferred between groups follow rigid procedures based on the MDB ICD

The GTS system is composed of Aspera and the DTSTool [Valette and Dufourg, 2011]. Aspera is a commercial tool for fast data transfer which has been customized and deployed to meet all GTS requirements (for more details see Section C.2). The DTSTool (Data Transfer Subsystem Tool) is responsible for building a simplified transfer service interface for Aspera, common to all DPCs.

DTSTool can distinguish between 3 different transfer types or channels:

Daily Transfers

For the transfers of data produced be the daily processing system at DPCE. This category also includes the transfer of offline calibrations not assignable to a DRC activity and the spacecraft configuration updates.

DRC Transfers

For the transfers of data generated by the DRC processing systems

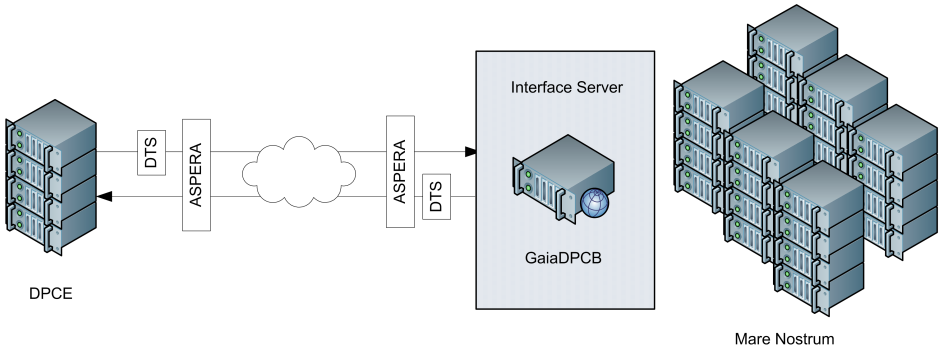


FIGURE C.1: DPCB Data Manager Input/Output Overview

Untagged Transfers

For any other type of transfers not fulfilling the previous description.

Each transfer type is in general associated to a predefined table mapping. Only a limited number of tables can be sent through the first two categories, being in general table representing data qualification or tables logging processing related events. All transfers must start or end at DPCE which is acting a the central hub for all the data transfers. This is done to assure that all the data is tracked in the MDB central repository.

Finally, each DPC is responsibility of the implementation of any additional software tools that may be required for a proper integration of the GTS on their side. In case of DPCB, this tool is the DPCB Data Manager (DDM) which is described in Section C.1

C.1 DPCB Data Manager

As introduced in Appendix A, DPCB has an *Interface Server* to provide the link between DPCE and DPCB as shown in Figure C.1. The server is physically located in the BSC premises and runs the Aspera software and the DPCB Data Manager (DDM). The DPCB Data Manager (DDM) [Clotet, 2015] is the system service integrating the GTS and also in charge of handling the data at DPCB.

DDM is included in DpcbTools and has been implemented using Python which allows higher flexibility to quickly adapt to possible changes in the BSC architecture. The main functionalities of the DDM are:

- to process the newly data arrived from DPCE, arrange and store it in DPCB repositories.
- to detect the output data coming from IDU, arrange it and send it to DPCE.

Furthermore, DDM interacts with the BSC systems to ensure data is properly backed up. The backup policy is described in Portell and Clotet [2015]. More actions might be required in future versions of the software, either because of design changes or because of the underlying architecture upon which the system works.

Logical environments are implemented by mean of several DDM service instances for:

- Operations (OPS)
- Validation (NO-OPS)
- Testing (TEST)

The purpose of these environments is evident. They are configured by means of their own configuration parameters (directory structure, processes, data flow, bandwidth, etc.) separating the data streams and guaranteeing the quality of service.

The design of the DDM follows a data driven approach where several actions (within a job) are triggered when a given set of conditions over the available data is fulfilled. These actions in general are basic file system operations; copy, move, remove, rename, etc. although it is also possible to execute Java based action. These Java actions are basically statistical,

reporting or arrangement processes which can not be done using the system built-in commands due to the impossibility to read and write the Gaia data file format directly.

The BSC environment has two shared repositories amongst all the computing nodes (GPFS Project and Scratch), and a long term storage system (GPFS Archive). Due to BSC security policies the access to these storage systems is strictly monitored and limited to queue jobs. The DDM has to take these limitations into account, interacting with the job queue and still deliver the expected functionality.

The DDM implements a monitoring utility, the so-called *DDM-watch*, which is in charge of performing a specific set of checks in order to ensure the proper operation of the DDM in the DPCB environment.

This utility is launched through a Unix system *crontab* service configured at the server and generates several reports with the result of all the checks performed in both the IFS and the BSC. Once the checks are finished an email is generated with a summary of the issues found, if any, and the generated reports attached, so the DPCB operators can analyse them and take action if needed.

C.2 Aspera

Aspera Point–to–Point is a complete file transfer application built upon Aspera's patented FASP^{TM} file transfer technology. Aspera is the data transfer tool used to transfer data between the different DPCs. The current licence provides up to 500 Mbps secure data transfers and includes data integrity checks.

Aspera's file transfer protocol dramatically speeds transfers over IP networks by eliminating the fundamental bottlenecks in conventional technologies. FASP^{TM} features include bandwidth control, transfer resuming and encryption, content protection, data integrity and validation.

D Gaia Data Volume

Table D.1 table shows the MDB theoretical size at different DRCs assuming a total of 10 DRCs in a 5 year mission. For simplicity in the first two DRCs only CU1 and CU3 data is taken into account, in DRC 3 also CU5 data is added. All data which is consumed continuously or whose number of estimated records is 8×10^{10} or has a field called transitId has been assumed to grow linearly with the DRCs.

Table D.2 and Table D.3 show the DPCB theoretical data size that will be accumulated after the last DRC exercise.

Table D.4 table shows the DPCB theoretical output data size for the last DRC.

DRC	MDB	DPCB
1	80.20 TBytes	20.89 TBytes
2	143.53 TBytes	27.82 TBytes
3	217.15 TBytes	34.74 TBytes
4	703.17 TBytes	41.67 TBytes
5	872.83 TBytes	48.60 TBytes
6	1.02 PBytes	55.52 TBytes
7	1.18 PBytes	62.45 TBytes
8	1.35 PBytes	69.38 TBytes
9	1.52 PBytes	76.30 TBytes
10	1.68 PBytes	83.23 TBytes

Table D.1: MDB and DPCB theoretical accumulated data size at different DRCs

Table	Record Size	Records	Total Size
CU3/IDT/Raw/AstroObservation	892 Bytes	8×10^{10}	64.9 TBytes
CU3/IDT/Raw/AstroObservationVo	892 Bytes	10^9	830.74 GBytes
CU3/IDT/Raw/ChargeInjection	1.21 KBytes	1.1×10^8	124.12 GBytes
CU3/IDT/Raw/ObjectLogAFXP	25 Bytes	10^9	23.28 GBytes
CU3/IDT/Raw/PreScan	8.03 KBytes	3×10^6	22.99 GBytes
CU3/IDT/Raw/GateInfoAstro	171 Bytes	3×10^6	489.23 MBytes
CU3/IDT/Raw/AcShifts	89 Bytes	3×10^6	254.63 MBytes
CU3/IDT/Raw/AcShiftsSmearing	27 Bytes	3×10^6	77.25 MBytes
CU3/IDT/Raw/GateInfoRaw	22 Bytes	3×10^6	62.94 MBytes
CU3/IDT/Xm/NewSource	471 Bytes	10^9	438.65 GBytes
CU3/IDT/Xm/Match	31 Bytes	8×10^{10}	2.26 TBytes
CU3/IDT/Xm/BlackListedTransits	51 Bytes	1.5×10^{10}	712.46 GBytes
CU3/IDT/Interm/Oga1	601.91 MBytes	1	601.91 MBytes
CU3/FL/Oga2	601.91 MBytes	1	601.91 MBytes

TABLE D.2: Size of MDB extract sent to DPCB coming from the daily processing pipeline

Table	Record Size	Records	Total Size
CU1/Integrated/CompleteSource	15 KBytes	10^9	13.97 TBytes
CU3/AGIS/Oga3	198.97 MBytes	1	198.97 MBytes
CU3/AGIS/AstroCalibration	55 Bytes	1	55 Bytes

TABLE D.3: Size of MDB extract sent to DPCB coming from the DRC processing

Table	Record Size	Records	Total Size
CU3/IDU/Scene	57 Bytes	8×10^{10}	4.15 TBytes
CU3/IDU/XM/WhiteListedTransits	51 Bytes	2×10^9	94.99 GBytes
CU3/IDU/XM/BlackListedTransits	51 Bytes	1.5×10^{10}	712.46 GBytes
CU3/IDU/XM/NewSource	471 Bytes	10^9	438.65 GBytes
CU3/IDU/XM/Track	828 Bytes	10^9	771.14 GBytes
CU3/IDU/XM/Match	31 Bytes	8×10^{10}	2.26 TBytes
CU3/IDU/XM/AmbiguousMatch	357 Bytes	8×10^{10}	25.98 TBytes
CU3/IDU/BiasRecordDt	105 Bytes	5×10^6	500.68 MBytes
CU3/IDU/ApBackgroundRecordDt	90 Bytes	5×10^6	429.15 MBytes
CU3/IDU/EmpiricalLsf/OpticalCorrections	91 Bytes	2.5×10^7	2.12 GBytes
CU3/IDU/EmpiricalLsf/ElectronicCorrections	83 Bytes	2.5×10^7	1.93 GBytes
CU3/IDU/EmpiricalLsf/EmpiricalLsfLibrary	26 Bytes	5×10^5	1.24 MBytes
CU3/IDU/AstroElementary	751 Bytes	8×10^{10}	54.64 TBytes
CU3/IDU/GsConfigParam	102 Bytes	100	9.96 KBytes
CU3/IDU/SolutionId/InputDataUsed	71 Bytes	2×10^9	132.25 GBytes
CU3/IDU/SolutionId/SolutionIdMetaData	73 Bytes	7×10^6	487.33 MBytes
CU3/IDU/SolutionId/SolutionIdQualification	41 Bytes	7×10^6	273.7 MBytes

TABLE D.4: Size of DPCB data sent to MDB

E IDU Task Templates

These are the main IDU task templates:

```
<?xml version="1.0" encoding="UTF-8" ?>

<TASK name="detectionClassifierTask" class="gaia.cu3.idu.xm.infra.DetectionClassifierTask"
     priority="5" ttl="0">
  <PROPERTIES>conf/idu.xm.properties</PROPERTIES>
  <!-- TIMEINTERVAL -->
  <!-- CCDROW -->
  <DATASTORES>
    <!-- Configuration data -->
    <FILE dataType="gaia.cu1.mdb.cu3.idu.dm.GsConfigParam" url="${COMMONINPUT}/gscp"
     filePattern="FREE"/>
    <!-- FL Qualification, required for Scene processing -->
    <FILE dataType="gaia.cu1.mdb.cu3.fl.dm.FlQualificationInfo" url="${COMMONINPUT}/qi_fl"
     filePattern="FREE"/>
    <!-- AstroCalibration, required for Scene processing -->
    <FILE dataType="gaia.cu1.mdb.cu1.dm.AstroCalibration" url="${COMMONINPUT}/ac"/>
    <!-- IDT/IDU Calibrations -->
    <FILE dataType="gaia.cu1.mdb.cu3.idu.dm.Scene" url="${COMMONINPUT}/scn"/>
    <!-- Raw data -->
    <FILE dataType="gaia.cu1.mdb.cu3.idt.raw.dm.ObjectLogAFXP" url="${COMMONINPUT}/ol"/>
    <FILE dataType="gaia.cu1.mdb.cu3.idt.raw.dm.AstroObservation" url="${COMMONINPUT}/ao"/>
  </DATASTORES>
  <PATHS>
    <!-- COMMONINPUT -->
    <!-- TASKINPUT -->
    <!-- TASKOUTPUT -->
    <!-- LOCAL -->
  </PATHS>
</TASK>
```

Listing E.1: DetectionClassifier task

```
<?xml version="1.0" encoding="UTF-8" ?>

<TASK name="scnTask" class="gaia.cu3.idu.scene.infra.SceneTask" priority="5" ttl="0">
  <PROPERTIES>conf/idu.scene.properties</PROPERTIES>
  <!-- TIMEINTERVAL -->
  <!-- CCDROW -->
  <!-- HEALPIXSET -->
  <DATASTORES>
    <!-- Configuration data -->
    <FILE dataType="gaia.cu1.mdb.cu3.idu.dm.GsConfigParam" url="${COMMONINPUT}/gscp"
     filePattern="FREE"/>
    <!-- FL Qualification -->
    <FILE dataType="gaia.cu1.mdb.cu3.fl.dm.FlQualificationInfo" url="${COMMONINPUT}/qi_fl"
     filePattern="FREE"/>
    <!-- AstroCalibration -->
    <FILE dataType="gaia.cu1.mdb.cu1.dm.AstroCalibration" url="${COMMONINPUT}/ac"/>
    <!-- Attitude -->
    <FILE dataType="gaia.cu1.mdb.cu1.basictypes.dm.BSplineFittedAttitudeData" url="${
     COMMONINPUT}/att"/>
    <!-- Source interfaces -->
```

```
    <FILE dataType="gaia.cu1.mdb.cu3.idt.xm.dm.NewSource" url="${COMMONINPUT}/nsrc_idt"
     filePattern="healpix"/>
    <FILE dataType="gaia.cu1.mdb.cu1.integrated.dm.CompleteSource" url="${COMMONINPUT}/
     src_mdb" filePattern="healpix"/>
  </DATASTORES>
  <PATHS>
    <!-- COMMONINPUT -->
    <!-- TASKINPUT -->
    <!-- TASKOUTPUT -->
    <!-- LOCAL -->
  </PATHS>
</TASK>
```

Listing E.2: Scene task

```
<?xml version="1.0" encoding="UTF-8" ?>

<TASK name="obssrcMatchTask" class="gaia.cu3.idu.xm.infra.ObsSrcMatchTask" priority="5" ttl
     ="0">
  <PROPERTIES>conf/idu.xm.properties</PROPERTIES>
  <!-- TIMEINTERVAL -->
  <!-- CCDROW -->
  <DATASTORES>
    <!-- Configuration data -->
    <FILE dataType="gaia.cu1.mdb.cu3.idu.dm.GsConfigParam" url="${COMMONINPUT}/gscp"
     filePattern="FREE"/>
    <!-- FL Qualification -->
    <FILE dataType="gaia.cu1.mdb.cu3.fl.dm.FlQualificationInfo" url="${COMMONINPUT}/qi_fl"
     filePattern="FREE"/>
    <!-- Attitude -->
    <FILE dataType="gaia.cu1.mdb.cu1.basictypes.dm.BSplineFittedAttitudeData" url="${
     COMMONINPUT}/att"/>
    <!-- AstroCalibration -->
    <FILE dataType="gaia.cu1.mdb.cu1.dm.AstroCalibration" url="${COMMONINPUT}/ac"/>
    <!-- TransitId interfaces -->
    <FILE dataType="gaia.cu1.mdb.cu3.idt.raw.dm.AstroObservation" url="${COMMONINPUT}/ao"/>
    <!-- DetectionClassification interfaces -->
    <FILE dataType="gaia.cu1.mdb.cu3.idu.xm.dm.BlackListedTransits" url="${COMMONINPUT}/bl"
     />
    <!-- Source interfaces -->
    <FILE dataType="gaia.cu1.mdb.cu3.idt.xm.dm.NewSource" url="${COMMONINPUT}/nsrc_idt"
     filePattern="healpix"/>
    <FILE dataType="gaia.cu1.mdb.cu1.integrated.dm.CompleteSource" url="${COMMONINPUT}/
     src_mdb" filePattern="healpix"/>
  </DATASTORES>
  <PATHS>
    <!-- COMMONINPUT -->
    <!-- TASKINPUT -->
    <!-- TASKOUTPUT -->
    <!-- LOCAL -->
  </PATHS>
</TASK>
```

Listing E.3: ObsSrcMatch task

```
<?xml version="1.0" encoding="UTF-8" ?>

<TASK name="unmatchProcessorTask" class="gaia.cu3.idu.xm.infra.UnmatchProcessorTask"
     priority="5" ttl="0">
  <PROPERTIES>conf/idu.xm.properties</PROPERTIES>
  <!-- HEALPIXSET -->
  <DATASTORES>
    <!-- Configuration data -->
    <FILE dataType="gaia.cu1.mdb.cu3.idu.dm.GsConfigParam" url="${COMMONINPUT}/gscp"
     filePattern="FREE"/>
    <!-- MatchCandidate interfaces -->
    <FILE dataType="gaia.cu3.idtools.dm.MatchCandidate" url="${TASKINPUT}/mc" filePattern="
     healpix"/>
  </DATASTORES>
  <PATHS>
    <!-- COMMONINPUT -->
    <!-- TASKINPUT -->
    <!-- TASKOUTPUT -->
    <!-- LOCAL -->
  </PATHS>
</TASK>
```

Listing E.4: Unmatch Processor task

```
<?xml version="1.0" encoding="UTF-8" ?>

<TASK name="skyPartitionerTask" class="gaia.cu3.idu.xm.infra.SkyPartitionerTask" priority="
     5" ttl="0">
  <PROPERTIES>conf/idu.xm.properties</PROPERTIES>
  <!-- HEALPIXSET -->
  <DATASTORES>
    <!-- Configuration data -->
    <FILE dataType="gaia.cu1.mdb.cu3.idu.dm.GsConfigParam" url="${COMMONINPUT}/gscp"
     filePattern="FREE"/>
    <!-- MatchCandidate interfaces -->
    <FILE dataType="gaia.cu3.idtools.dm.MatchCandidate" url="${COMMONINPUT}/mc" filePattern
     ="healpix"/>
    <!-- SourceIdInfo interfaces -->
    <FILE dataType="gaia.dpcb.dm.SourceIdInfo" url="${COMMONINPUT}/si" filePattern="healpix
     "/>
  </DATASTORES>
  <PATHS>
    <!-- COMMONINPUT -->
    <!-- TASKINPUT -->
    <!-- TASKOUTPUT -->
    <!-- LOCAL -->
  </PATHS>
</TASK>
```

Listing E.5: Sky Partitioner task

```
<?xml version="1.0" encoding="UTF-8" ?>

<TASK name="mcgCatResolverTask" class="gaia.cu3.idu.xm.infra.McgResolverTask" priority="5"
     ttl="0">
  <PROPERTIES>conf/idu.xm.properties</PROPERTIES>
  <!-- HEALPIXSET -->
  <DATASTORES>
    <!-- Configuration data -->
    <FILE dataType="gaia.cu1.mdb.cu3.idu.dm.GsConfigParam" url="${COMMONINPUT}/gscp"
     filePattern="FREE"/>
    <!-- Catalogue -->
    <FILE dataType="gaia.cu1.mdb.cu1.dm.AstrometricSource" url="${COMMONINPUT}/src"
     filePattern="healpix"/>
    <!-- MatchCandidateGroup -->
    <FILE dataType="gaia.cu3.idtools.dm.MatchCandidateGroup" url="${COMMONINPUT}/mcg"
     filePattern="healpix"/>
    <!-- Reference XM -->
    <!--FILE dataType="gaia.cu1.mdb.cu3.id.dm.Match" url="${COMMONINPUT}/xm" filePattern="
     healpix"/-->
  </DATASTORES>
  <PATHS>
    <!-- COMMONINPUT -->
    <!-- TASKINPUT -->
    <!-- TASKOUTPUT -->
    <!-- LOCAL -->
  </PATHS>
</TASK>
```

Listing E.6: MatchCandidateGroup Resolver task

```
<?xml version="1.0" encoding="UTF-8" ?>

<TASK name="newSourceConsolidatorTask" class="gaia.cu3.idu.xm.infra.
     NewSourceConsolidatorTask" priority="5" ttl="0">
  <PROPERTIES>conf/idu.xm.properties</PROPERTIES>
  <!-- HEALPIXSET -->
  <DATASTORES>
    <!-- Configuration data -->
    <FILE dataType="gaia.cu1.mdb.cu3.idu.dm.GsConfigParam" url="${COMMONINPUT}/gscp"
     filePattern="FREE"/>
    <!-- SrcRunningNumber -->
    <FILE dataType="gaia.dpcb.dm.SrcRunningNumber" url="${COMMONINPUT}/srn" filePattern="
     FREE"/>
    <!-- XM -->
    <FILE dataType="gaia.cu1.mdb.cu3.id.dm.NewSource" url="${COMMONINPUT}/nsrc" filePattern
     ="healpix"/>
    <FILE dataType="gaia.cu1.mdb.cu3.id.dm.Match" url="${COMMONINPUT}/xm" filePattern="
     healpix"/>
    <FILE dataType="gaia.cu1.mdb.cu3.id.dm.AmbiguousMatch" url="${COMMONINPUT}/xam"
     filePattern="healpix"/>
    <FILE dataType="gaia.cu1.mdb.cu1.dm.Track" url="${COMMONINPUT}/track" filePattern="
     healpix"/>
```

```
  </DATASTORES>
  <PATHS>
    <!-- COMMONINPUT -->
    <!-- TASKINPUT -->
    <!-- TASKOUTPUT -->
    <!-- LOCAL -->
  </PATHS>
</TASK>
```

Listing E.7: NewSourceId Consolidator task

```
<?xml version="1.0" encoding="UTF-8" ?>

<TASK name="preScanTask" class="gaia.cu3.idu.bias.infra.PreScanTask" priority="5" ttl="0">
  <PROPERTIES>conf/idu.bias.properties</PROPERTIES>
  <!-- TIMEINTERVAL -->
  <!-- CCDROW -->
  <DATASTORES>
    <!-- Configuration data -->
    <FILE dataType="gaia.cu1.mdb.cu3.idu.dm.GsConfigParam" url="${COMMONINPUT}/gscp"
     filePattern="FREE"/>
    <!-- Raw data -->
    <FILE dataType="gaia.cu1.mdb.cu3.idt.raw.dm.PreScan" url="${COMMONINPUT}/ps"/>
  </DATASTORES>
  <PATHS>
    <!-- COMMONINPUT -->
    <!-- TASKINPUT -->
    <!-- TASKOUTPUT -->
    <!-- LOCAL -->
  </PATHS>
</TASK>
```

Listing E.8: Bias task

```
<?xml version="1.0" encoding="UTF-8" ?>

<TASK name="apbTask" class="gaia.cu3.idu.cicrb.infra.ApbTask" priority="5" ttl="0">
  <PROPERTIES>conf/idu.cicrb.properties</PROPERTIES>
  <!-- TIMEINTERVAL -->
  <!-- CCDROW -->
  <DATASTORES>
    <!-- Configuration data -->
    <FILE dataType="gaia.cu1.mdb.cu3.idu.dm.GsConfigParam" url="${COMMONINPUT}/gscp"
     filePattern="FREE"/>
    <!-- FL Calibrations -->
    <FILE dataType="gaia.cu1.mdb.cu3.fl.dm.FlQualificationInfo" url="${COMMONINPUT}/qi_fl"
     filePattern="FREE"/>
    <FILE dataType="gaia.cu1.mdb.cu3.fl.dm.BiasNUCalibrationLibrary" url="${COMMONINPUT}/
     nu_fl" filePattern="FREE"/>
    <FILE dataType="gaia.cu1.mdb.cu3.fl.dm.CiAcProfileLibrary" url="${COMMONINPUT}/ci_fl"
     filePattern="FREE"/>
    <FILE dataType="gaia.cu1.mdb.cu3.fl.dm.CrBackgroundLibrary" url="${COMMONINPUT}/cr_fl"
     filePattern="FREE"/>
    <!-- IDT/IDU Calibrations -->
    <FILE dataType="gaia.cu1.mdb.cu3.id.dm.BiasRecordDt" url="${COMMONINPUT}/bias"/>
    <FILE dataType="gaia.cu1.mdb.cu3.idu.dm.Scene" url="${COMMONINPUT}/scn"/>
    <!-- Raw data -->
    <FILE dataType="gaia.cu1.mdb.cu3.idt.raw.dm.AcShifts" url="${COMMONINPUT}/as"/>
    <FILE dataType="gaia.cu1.mdb.cu3.idt.raw.dm.ObjectLogAFXP" url="${COMMONINPUT}/ol"/>
    <FILE dataType="gaia.cu1.mdb.cu3.idt.raw.dm.AstroObservation" url="${COMMONINPUT}/ao"/>
    <FILE dataType="gaia.cu1.mdb.cu3.idt.raw.dm.AstroObservationVo" url="${COMMONINPUT}/vo"
     />
  </DATASTORES>
  <PATHS>
    <!-- COMMONINPUT -->
    <!-- TASKINPUT -->
    <!-- TASKOUTPUT -->
    <!-- LOCAL -->
  </PATHS>
</TASK>
```

Listing E.9: Astrophysical Background task

```
<?xml version="1.0" encoding="UTF-8" ?>

<TASK name="elsfCalibratorTask" class="gaia.cu3.idu.lsfpsf.infra.ElsfCalibratorTask"
     priority="5" ttl="0">
  <PROPERTIES>conf/idu.lsfpsf.properties</PROPERTIES>
```

```
  <!-- TIMEINTERVAL -->
  <!-- CCDROW -->
  <DATASTORES>
    <!-- Configuration data -->
    <FILE dataType="gaia.cu1.mdb.cu3.idu.dm.GsConfigParam" url="${COMMONINPUT}/gscp"
     filePattern="FREE"/>
    <!-- FL Qualification -->
    <FILE dataType="gaia.cu1.mdb.cu3.fl.dm.FlQualificationInfo" url="${COMMONINPUT}/qi_fl"
     filePattern="FREE"/>
    <!-- Attitude -->
    <FILE dataType="gaia.cu1.mdb.cu1.basictypes.dm.BSplineFittedAttitudeData" url="${
     COMMONINPUT}/att"/>
    <!-- AstroCalibration -->
    <FILE dataType="gaia.cu1.mdb.cu1.dm.AstroCalibration" url="${COMMONINPUT}/ac"/>
    <!-- FL Calibrations -->
    <FILE dataType="gaia.cu1.mdb.cu3.fl.dm.BiasNUCalibrationLibrary" url="${COMMONINPUT}/
     nu_fl" filePattern="FREE"/>
    <FILE dataType="gaia.cu1.mdb.cu3.fl.dm.CiAcProfileLibrary" url="${COMMONINPUT}/ci_fl"
     filePattern="FREE"/>
    <FILE dataType="gaia.cu1.mdb.cu3.fl.dm.CrBackgroundLibrary" url="${COMMONINPUT}/cr_fl"
     filePattern="FREE"/>
    <FILE dataType="gaia.cu1.mdb.cu3.fl.dm.CcdHealthLibrary" url="${COMMONINPUT}/ch_fl"
     filePattern="FREE"/>
    <FILE dataType="gaia.cu1.mdb.cu3.fl.dm.CcdSaturationLibrary" url="${COMMONINPUT}/cs_fl"
      filePattern="FREE"/>
    <!-- Raw data -->
    <FILE dataType="gaia.cu1.mdb.cu3.idt.raw.dm.AcShifts" url="${COMMONINPUT}/as"/>
    <FILE dataType="gaia.cu1.mdb.cu3.idt.raw.dm.ObjectLogAFXP" url="${COMMONINPUT}/ol"/>
    <FILE dataType="gaia.cu1.mdb.cu3.idt.raw.dm.AstroObservation" url="${COMMONINPUT}/ao"/>
    <!-- IDT/IDU Calibrations -->
    <FILE dataType="gaia.cu1.mdb.cu3.id.dm.BiasRecordDt" url="${COMMONINPUT}/bias"/>
    <FILE dataType="gaia.cu1.mdb.cu3.id.dm.ApBackgroundRecordDt" url="${COMMONINPUT}/apb"/>
    <FILE dataType="gaia.cu1.mdb.cu3.idu.dm.Scene" url="${COMMONINPUT}/scn" filePattern="
     HEALPIX"/>
    <!-- Match interfaces -->
    <FILE dataType="gaia.cu1.mdb.cu3.id.dm.Match" url="${COMMONINPUT}/xm"/>
    <!-- Source interfaces -->
    <FILE dataType="gaia.dpcb.dm.DpcbSource" url="${COMMONINPUT}/src" filePattern="healpix"
     />
  </DATASTORES>
  <PATHS>
    <!-- COMMONINPUT -->
    <!-- TASKINPUT -->
    <!-- TASKOUTPUT -->
    <!-- LOCAL -->
  </PATHS>
</TASK>
```

Listing E.10: ELSF Calibrator task

```
<?xml version="1.0" encoding="UTF-8" ?>

<TASK name="elsfOptCorTask" class="gaia.cu3.idu.lsfpsf.infra.ElsfOptCorTask" priority="5"
     ttl="0">
  <PROPERTIES>conf/idu.lsfpsf.properties</PROPERTIES>
  <!-- TIMEINTERVAL -->
  <!-- CCDROW -->
  <DATASTORES>
    <!-- Configuration data -->
    <FILE dataType="gaia.cu1.mdb.cu3.idu.dm.GsConfigParam" url="${COMMONINPUT}/gscp"
     filePattern="FREE"/>
    <!-- Raw data -->
    <FILE dataType="gaia.cu3.idtools.dm.CompleteWindow" url="${COMMONINPUT}/cwin"/>
    <!-- Other data -->
    <FILE dataType="gaia.cu1.mdb.cu3.empiricallsf.dm.BasisComponentSet" url="${COMMONINPUT
     }/elsf_bc" filePattern="FREE"/>
    <FILE dataType="gaia.cu1.mdb.cu3.empiricallsf.dm.MeanLsf" url="${COMMONINPUT}/elsf_mean
     " filePattern="FREE"/>
    <!-- CorrectionsHH interfaces -->
    <FILE dataType="gaia.cu5.du10.empiricallsfv2.householder.dm.ElectronicCorrectionsHH"
     url="${COMMONINPUT}/elsf_echh"/>
    <FILE dataType="gaia.cu5.du10.empiricallsfv2.householder.dm.OpticalCorrectionsHH" url="
     ${COMMONINPUT}/elsf_ochh"/>
  </DATASTORES>
  <PATHS>
    <!-- COMMONINPUT -->
    <!-- TASKINPUT -->
    <!-- TASKOUTPUT -->
    <!-- LOCAL -->
  </PATHS>
</TASK>
```

Listing E.11: ELS Optical Corrections task

```
<?xml version="1.0" encoding="UTF-8" ?>

<TASK name="elsfLibSnapshotTask" class="gaia.cu3.idu.lsfpsf.infra.ElsfLibSnapshotTask"
     priority="5" ttl="0">
  <PROPERTIES>conf/idu.lsfpsf.properties</PROPERTIES>
  <!-- TIMEINTERVAL -->
  <!-- CCDROW -->
  <DATASTORES>
    <!-- Configuration data -->
    <FILE dataType="gaia.cu1.mdb.cu3.idu.dm.GsConfigParam" url="${COMMONINPUT}/gscp"
     filePattern="FREE"/>
    <!-- Other data -->
    <FILE dataType="gaia.cu1.mdb.cu3.empiricallsf.dm.BasisComponentSet" url="${COMMONINPUT
     }/elsf_bc" filePattern="FREE"/>
    <FILE dataType="gaia.cu1.mdb.cu3.empiricallsf.dm.MeanLsf" url="${COMMONINPUT}/elsf_mean
     " filePattern="FREE"/>
    <!-- Corrections interfaces -->
    <FILE dataType="gaia.cu1.mdb.cu3.empiricallsf.dm.ElectronicCorrections" url="${
     COMMONINPUT}/elsf_ec"/>
    <FILE dataType="gaia.cu1.mdb.cu3.empiricallsf.dm.OpticalCorrections" url="${COMMONINPUT
     }/elsf_oc"/>
    <!-- EmpiricalLsfLibrary interfaces -->
    <FILE dataType="gaia.cu1.mdb.cu3.idu.empiricallsf.dm.EmpiricalLsfLibrary" url="${
     COMMONINPUT}/elsf"/>
  </DATASTORES>
  <PATHS>
    <!-- COMMONINPUT -->
    <!-- TASKINPUT -->
    <!-- TASKOUTPUT -->
    <!-- LOCAL -->
  </PATHS>
</TASK>
```

Listing E.12: ELSF Library Snapshot task

```
<?xml version="1.0" encoding="UTF-8" ?>

<TASK name="ipdTask" class="gaia.cu3.idu.ipd.infra.IpdTask" priority="5" ttl="0">
  <PROPERTIES>conf/idu.ipd.properties</PROPERTIES>
  <!-- TIMEINTERVAL -->
  <!-- CCDROW -->
  <DATASTORES>
    <!-- Configuration data -->
    <FILE dataType="gaia.cu1.mdb.cu3.idu.dm.GsConfigParam" url="${COMMONINPUT}/gscp"
     filePattern="FREE"/>
    <!-- FL Qualification -->
    <FILE dataType="gaia.cu1.mdb.cu3.fl.dm.FlQualificationInfo" url="${COMMONINPUT}/qi_fl"
     filePattern="FREE"/>
    <!-- Attitude -->
    <FILE dataType="gaia.cu1.mdb.cu1.basictypes.dm.BSplineFittedAttitudeData" url="${
     COMMONINPUT}/att"/>
    <!-- AstroCalibration -->
    <FILE dataType="gaia.cu1.mdb.cu1.dm.AstroCalibration" url="${COMMONINPUT}/ac"/>
    <!-- FL Calibrations -->
    <FILE dataType="gaia.cu1.mdb.cu3.fl.dm.BiasNUCalibrationLibrary" url="${COMMONINPUT}/
     nu_fl" filePattern="FREE"/>
    <FILE dataType="gaia.cu1.mdb.cu3.fl.dm.CiAcProfileLibrary" url="${COMMONINPUT}/ci_fl"
     filePattern="FREE"/>
    <FILE dataType="gaia.cu1.mdb.cu3.fl.dm.CrBackgroundLibrary" url="${COMMONINPUT}/cr_fl"
     filePattern="FREE"/>
    <FILE dataType="gaia.cu1.mdb.cu3.fl.dm.CcdHealthLibrary" url="${COMMONINPUT}/ch_fl"
     filePattern="FREE"/>
    <FILE dataType="gaia.cu1.mdb.cu3.fl.dm.CcdSaturationLibrary" url="${COMMONINPUT}/cs_fl"
      filePattern="FREE"/>
    <!-- Raw data -->
    <FILE dataType="gaia.cu1.mdb.cu3.idt.raw.dm.AcShifts" url="${COMMONINPUT}/as"/>
    <FILE dataType="gaia.cu1.mdb.cu3.idt.raw.dm.ObjectLogAFXP" url="${COMMONINPUT}/ol"/>
    <FILE dataType="gaia.cu1.mdb.cu3.idt.raw.dm.GateInfoAstro" url="${COMMONINPUT}/ga"/>
    <FILE dataType="gaia.cu1.mdb.cu3.idt.raw.dm.AstroObservation" url="${COMMONINPUT}/ao"/>
    <!-- IDT/IDU Calibrations -->
    <FILE dataType="gaia.cu1.mdb.cu3.id.dm.BiasRecordDt" url="${COMMONINPUT}/bias"/>
    <FILE dataType="gaia.cu1.mdb.cu3.id.dm.ApBackgroundRecordDt" url="${COMMONINPUT}/apb"/>
    <FILE dataType="gaia.cu1.mdb.cu3.idu.empiricallsf.dm.EmpiricalLsfLibrary" url="${
     COMMONINPUT}/elsf" filePattern="FREE"/>
    <FILE dataType="gaia.cu1.mdb.cu3.idu.dm.Scene" url="${COMMONINPUT}/scn" filePattern="
     HEALPIX"/>
```

```
    <!-- Match interfaces -->
    <FILE dataType="gaia.cu1.mdb.cu3.id.dm.Match" url="${COMMONINPUT}/xm"/>
    <!-- Source/Color interfaces -->
    <FILE dataType="gaia.dpcb.dm.DpcbSource" url="${COMMONINPUT}/src" filePattern="healpix"
     />
  </DATASTORES>
  <PATHS>
    <!-- COMMONINPUT -->
    <!-- TASKINPUT -->
    <!-- TASKOUTPUT -->
    <!-- LOCAL -->
  </PATHS>
</TASK>
```

Listing E.13: Image Parameters Determination task

Bibliography

The following is a complete list of references used in this document.

DPAC: Proposal for the Gaia Data Processing, April 2007. URL http://www.rssd.esa.int/cs/livelink/open/2720336. 44

Airbus DS. VPU to PDHU Telemetry Interface Control Document, February 2015. URL http://www.rssd.esa.int/cs/livelink/open/2777572. 32

M. Altmann and U. Bastian. Ecliptic Poles Catalogue Version 1.1, March 2009. URL http://www.rssd.esa.int/cs/livelink/open/2885828. 286

A. H. Andrei, J. Souchay, N. Zacharias, R. L. Smart, R. Vieira Martins, D. N. da Silva Neto, J. I. B. Camargo, M. Assafin, C. Barache, S. Bouquillon, J. L. Penna, and F. Taris. The large quasar reference frame (LQRF). An optical representation of the ICRS. *A&A*, 505:385–404, October 2009. doi: 10.1051/0004-6361/200912041. 288

E. Antiche, R. Borrachero, M. Clotet, et al. Gaia Operational Complete Source Statistics, IGSL, April 2014. URL http://www.rssd.esa.int/cs/livelink/open/3254677. 72

S. Azaz. Sky-background-induced false detections, December 2014. URL http://www.rssd.esa.int/cs/livelink/open/3296747. 65

U. Bastian. Reference systems, conventions and notations for Gaia, July 2007. URL http://www.rssd.esa.int/cs/livelink/open/358698. 23, 36

U. Bastian. Source Identifiers - Assignment and Usage throughout DPAC, June 2013. URL http://www.rssd.esa.int/cs/livelink/open/2779219. 72, 234

U. Bastian. Micro-meteorite hits and micro-clanks in the Gaia attitude, August 2015. URL http://www.rssd.esa.int/cs/livelink/open/3365598. 35

U. Bastian and F. van Leeuwen. The necessity of extensive attitude actuator data in the Gaia telemetry, January 2007. URL http://www.rssd.esa.int/cs/livelink/open/2091040. 32

J. J. Bestard. Study and treatment of spurious detections in IDT. Technical report, University of Barcelona (Dept. Astronomia i Meteorologia), July 2015. URL http://gaia.esac.esa.int/dpacsvn/DPAC/CU3/docs/IDT/nonECSS/algorithms/SpuriousModel/GAIA-C3-TN-UB-JJ-001.pdf. 67, 178

A. Bombrun, L. Lindegren, B. Holl, and S. Jordan. Complexity of the Gaia astrometric least-squares problem and the (non-)feasibility of a direct solution method. *A&A*, 516:A77, June-July 2010. 51

A. Brown and S. Jordan. Stray light modulations caused by diffracted sun light, April 2014. URL http://www.rssd.esa.int/cs/livelink/open/3256532. 91

S. Brown and C. Crowley. Image Parameter Determination for 1D Windows with a parameterisation of PCA component amplitude changes, January 2012. URL http://www.rssd.esa.int/cs/livelink/open/3109976. 94

BSC. *Barcelona Supercomputing Center*. http://www.bsc.es, December 2014. 284

BSC. MareNostrum III User's Guide . http://www.bsc.es/support/MareNostrum3-ug.pdf, August 2015. 152, 246

R. Burgon, S. Hodgkin, and L. Wyrzykowski. Proposed Alert Dissemination and Format for the Gaia Science Alerts - Publication System, September 2010. URL http://www.rssd.esa.int/cs/livelink/open/3045166. 47

J. M. Carrasco, H. Voss, C. Jordi, C. Fabricius, E. Pancino, and G. Altavilla. Selection of stars to calibrate gaia. In *Proceedings of the XI Scientific Meeting of the Spanish Astronomical Society*, 2015. 53

J. Castañeda. High-precision space astrometry and on-ground data management. Master's thesis, Universitat Politècnica de Catalunya, 2008. 73, 77, 287

J. Castañeda and C. Fabricius. Maximum Likelihood Fitting algorithms - Test Report, July 2009. URL http://www.rssd.esa.int/cs/livelink/open/3079105. 109

J. Castañeda and C. Fabricius. AttitudeToHealpix in CU3-IDA, June 2010. URL http://www.rssd.esa.int/cs/livelink/open/3070403. 73, 108

J. Castañeda and C. Fabricius. Coordinate Reference Systems operations in IDT/IDU algorithms, February 2012. URL http://www.rssd.esa.int/cs/livelink/open/3125232. 38, 39, 61

J. Castañeda and J. Portell. TmTools Interface Control Document, December 2009. URL http://www.rssd.esa.int/cs/livelink/open/2964363. 39

J. Castañeda, A. Fries, and J. Portell. DPCB Tools Software Product Overall Description, December 2009a. URL http://www.rssd.esa.int/cs/livelink/open/2961199. 285

J. Castañeda, O. Martinez, C. Fabricius, et al. TmTools Implementation of Astrium TM Model for Cycle 6, April 2009b. URL http://www.rssd.esa.int/cs/livelink/open/2894737. 39

J. Castañeda, C. Fabricius, J. Torra, et al. Intermediate Data Updating Software Requirements Specification, July 2011a. URL http://www.rssd.esa.int/cs/livelink/open/3086451. 59, 71, 117

J. Castañeda, O. Martinez, and J. Portell. TmTools Software Design Description, July 2011b. URL http://www.rssd.esa.int/cs/livelink/open/2894956. 39

J. Castañeda, J. Portell, and N. Blagorodnova. Intermediate Data Updating Software Test Specification, September 2011c. URL http://www.rssd.esa.int/cs/livelink/open/2892965. 154

J. Castañeda, J. Portell, A. Fries, et al. DpcbTools Software Requirements Specification, June 2011d. URL http://www.rssd.esa.int/cs/livelink/open/3081149. 251

J. Castañeda, A. Spagna, C. Fabricius, J. Portell, J. Torra, and N. Blagoródnova. Cross match for CU3-IDT. Technical report, University of Barcelona (Dept. Astronomia i Meteorologia), April 2011e. URL http://gaia.esac.esa.int/dpacsvn/DPAC/CU3/docs/IDT/nonECSS/algorithms/XM/Xm/GAIA-C3-TN-UB-JC-043.pdf. 71, 85

J. Castañeda, M. Clotet, and J. Portell. DPCB Operations Plan, December 2013. URL http://www.rssd.esa.int/cs/livelink/open/3097651. 165, 170, 245

J. Castañeda, M. Clotet, and al. IDT crossmatch reprocessing at DPCB. Technical report, University of Barcelona (Dept. Astronomia i Meteorologia), April 2015. URL http://gaia.esac.esa.int/dpacsvn/DPAC/

DPCB/docs/nonECSS/IdtRxmDs0/GAIA-DB-TN-UB-JC-072.pdf. 167, 233

J. P. Castañeda. Preparation and description of tasks for CU3-IDU, March 2009. URL http://www.rssd.esa.int/cs/livelink/open/2889884. 143

S. Clark and EJR-Quartz. *ESA BR-296 Gaia: ESA's galactic census*. ESA Communications Production, 2012. 3

M. Clotet, J.J.Gonzalez, J. Castañeda, and C. Fabricius. Idu-xm clustering resolver. Technical report, University of Barcelona (Dept. Astronomia i Meteorologia), September 2015. 85

M. G. Clotet. DPCB Data Manager, July 2015. URL http://www.rssd.esa.int/cs/livelink/open/3233997. 258

N. Cross and N. Hambly. Initial In-Orbit Characterisation of radiation damage in AF and XP (IIOC–150), October 2014. URL http://www.rssd.esa.int/cs/livelink/open/3278759. 30

N. J. G. Cross and N. C. Hambly. Charge injection stability in Astrium Radiation Campaign test data, October 2010. URL http://www.rssd.esa.int/cs/livelink/open/3046960. 29, 89

CU5 DU10. Early CU5DU10 dataflows for CTI, PSF/LSF, background and bias processing, March 2010. URL http://www.rssd.esa.int/cs/livelink/open/3011180. 87, 88

M. Davidson and N. Hambly. Empirical LSF Representation, Usage and Calibration in IDT/FL, March 2013. URL http://www.rssd.esa.int/cs/livelink/open/3177987. 94, 99

M. Davidson, C. Fabricius, and N. Hambly. Image Parameter Determination for 1D Windows with a Simplified CDM, December 2011. URL http://www.rssd.esa.int/cs/livelink/open/3105851. 94

M. Davidson, G. Busso, and G. Seabroke. Proposal for a Coordinated Virtual Object Strategy, November 2013. URL http://www.rssd.esa.int/cs/livelink/open/3055981. 27

J. de Bruijne. ALong- and ACross-scan location-estimation performance, July 2009. URL http://www.rssd.esa.int/cs/livelink/open/2913726. 92

J. H. J. de Bruijne, M. Allen, S. Azaz, A. Krone-Martins, T. Prod'homme, and D. Hestroffer. Detecting stars, galaxies, and asteroids with gaia. *A&A*, 576:A74, April 2015. doi: 10.1051/0004-6361/201424018. 65

EADS Astrium. Tests report of the radiation campaign #2, January 2008. URL http://www.rssd.esa.int/cs/livelink/open/2823366. 29

EADS Astrium. Radiation campaign #3. Serial register additional tests data delivery, May 2009a. URL http://www.rssd.esa.int/cs/livelink/open/2907680. 29

EADS Astrium. Radiation Campaign #3. Astrium report of the serial register tests, May 2009b. URL http://www.rssd.esa.int/cs/livelink/open/2911255. 29

EADS Astrium. RVS offset non-uniformity. Answers to action items from GAIA.ASF.MN.PLM.01553, June 2009c. URL http://www.rssd.esa.int/cs/livelink/open/2907569. 30

S. Els. DPAC Operations Interface Control Document, September 2014. URL http://www.rssd.esa.int/cs/livelink/open/3168420. 56

ESA. *European Space Agency Gaia Portal.* http://sci.esa.int/gaia, December 2014a. 3, 7

ESA. *Commissioning review: Gaia ready to start routine operations.* http://www.cosmos.esa.int/web/gaia/news_20140729, July 2014b. 10

ESA. *Commissioning update.* http://blogs.esa.int/gaia/2014/04/24/commissioning-update/, April 2014c. 9

ESA. *Gaia Discovers its first Supernova.* http://www.esa.int/Our_Activities/Space_Science/Gaia/Gaia_discovers_its_first_supernova/, December 2014d. 48

ESA. *Science Objectives.* http://www.cosmos.esa.int/web/gaia/science-objectives, January 2014e. 1

ESA. *Science Performance.* http://www.cosmos.esa.int/web/gaia/science-performance, January 2014f. 3

ESA. *Status of the Gaia strayligth analysis and mitigation actions.* http://www.cosmos.esa.int/web/gaia/news_20141217, December 2014g. 10

ESA. *A year on-station for Gaia.* http://blogs.esa.int/rocketscience/2015/01/14/a-year-on-station-for-gaia/, January 2015a. 10

ESA. *Data release scenario.* http://www.cosmos.esa.int/web/gaia/release/, August 2015b. 41

ESA. *Gaia Focal Plane.* http://www.cosmos.esa.int/web/gaia/focal-plane, January 2015c. 20

ESA. *Gaia Science Performance*. http://www.cosmos.esa.int/web/gaia/science-performance, January 2015d. 10

C. Fabricius. PSF model implementation for IDT, February 2011. URL http://www.rssd.esa.int/cs/livelink/open/3059410. 98

C. Fabricius. Reconstruction of the readout processes, May 2012. URL http://www.rssd.esa.int/cs/livelink/open/3028457. 30

C. Fabricius. Proposed changes to the VPA detection, July 2014a. URL http://www.rssd.esa.int/cs/livelink/open/3294855. 65

C. Fabricius. Spurious detections - A proposal for their treatment in IDT and IDU, October 2014b. URL http://www.rssd.esa.int/cs/livelink/open/3294850. 65

C. Fabricius and J. Torra. Proposal for the implementation of CTI mitigation in IDT and IDU, September 2011. URL http://www.rssd.esa.int/cs/livelink/open/3096570. 94

C. Fabricius, J. Castañeda, J. Torra, et al. Intermediate Data Updating, Definition, March 2009. URL http://www.rssd.esa.int/cs/livelink/open/2889494. 59

C. Fabricius, J. Castañeda, J. Portell, et al. IDU Cross Match, Definition, April 2011. URL http://www.rssd.esa.int/cs/livelink/open/3072606. 71, 76

C. Fabricius, J. Torra, J. Portell, et al. Treatment of non-nominal windows in IDT and IDU, January 2012. URL http://www.rssd.esa.int/cs/livelink/open/3108068. 107

A. Fries. *The use of Java in large scientific applications in HPC environments*. PhD thesis, UB, 2012. URL http://hdl.handle.net/10803/98405. 142, 154

A. Fries, J. Castañeda, Y. Isasi, G. L. Taboada, R. Sirvent, and J. Portell. An efficient framework for Java data processing systems in HPC environments. In *High-Performance Computing in Remote Sensing (SPIE 8183)*. American Institute of Physics, 2011. 251

E. García-Berro, J. Portell, J. Castañeda, and J. Gordo. Assessing the clock of gaia: Design and implementation of a clock framework simulator. *Exp. Astron.*, 18:133–158, 2006. 171

J. Gonzalez, M. Clotet, J. Castañeda, et al. DPCB test specification, April 2015. URL http://www.rssd.esa.int/cs/livelink/open/2914096. 155

K. M. Gorski, E. Hivon, A. J. Banday, B. D. Wandelt, F. K. Hansen, M. Reinecke, and M. Bartelman. HEALPix – a Framework for High Resolution Discretization, and Fast Analysis of Data Distributed on the Sphere. *ApJ*, 622:759–771, 2005. URL `http://arXiv:astro-ph/0409513`. xxviii, 73, 128, 287

R. Guerra and D. leaders CU leaders. DPAC System Validation and Test Plan, March 2013. URL `http://www.rssd.esa.int/cs/livelink/open/2898933`. 110, 155

N. Hambly and M. Davidson. Revision of the charge release profiling. Technical report, IfA, University of Edinburgh, January 2012. URL `http://gaia.esac.esa.int/dpacsvn/DPAC/CU5/docs/GAIA-C5-TN-IFA-NCH-022/GAIA-C5-TN-IFA-NCH-022.pdf`. 90

N. Hambly and C. Fabricius. Proposed processing design for handling the CCD–PEM offset non–uniformity in AF, XP and RVS, November 2010. URL `http://www.rssd.esa.int/cs/livelink/open/3051410`. 30, 31, 86

N. C. Hambly, M. Davidson, and R. S. Collins. CTI mitigation - implementation and development issues, August 2011. URL `http://www.rssd.esa.int/cs/livelink/open/3093085`. 93

M. Hauser, M. Davidson, and N. Hambly. On the data products of the First Look system, November 2014. URL `http://www.rssd.esa.int/cs/livelink/open/3294195`. 89

J. Hernandez. Solution Identifier–Assignment and Usage throughout DPAC, July 2012. URL `http://www.rssd.esa.int/cs/livelink/open/2848774`. 56, 150

J. Hernandez. Concepts related to MDB versioning and evolution. Technical report, ESAC, September 2013. URL `http://gaia.esac.esa.int/dpacsvn/DPAC/CU1/docs/TechNotes/Mdb-Evolution-JH-14/GAIA-C1-TN-ESAC-JH-014.pdf`. 56

J. Hernandez. Main Database Interface Control Document, March 2014. URL `http://www.rssd.esa.int/cs/livelink/open/2786145`. 55

J. Hoar. Performance Analysis and Metrics within DPAC, June 2010. URL `http://www.rssd.esa.int/cs/livelink/open/3034710`. 155

D. Hobbs. Implementation of M-order B-spline fitting for IOGA, May 2009. URL `http://www.rssd.esa.int/cs/livelink/open/2896701`. 35

D. Hobbs and L. Lindegren. Attitude Processing for Gaia, April 2010. URL `http://www.rssd.esa.int/cs/livelink/open/2843088`. 34

D. Hobbs and L. Lindegren. Impact of systematic basic angle variations on the parallax zero point, July 2011. URL `http://www.rssd.esa.int/cs/livelink/open/3010458`. 51

E. Høg, C. Fabricius, V. V. Makarov, S. Urban, T. Corbin, G. Wycoff, U. Bastian, P. Schwekendiek, and A. Wicenec. The Tycho-2 catalogue of the 2.5 million brightest stars. *A&A*, 355:L27–L30, March 2000. 291

B. Holl, T. Prod'homme, L. Lindegren, and A. G. A. Brown. The impact of CCD radiation damage on Gaia astrometry - II. Effect of image location errors on the astrometric solution. *MNRAS*, 442:2786–2807, 2012. doi: 10.1111/j.1365-2966.2012.20429.x. 27

A. S. Householder. Unitary triangularization of a nonsymmetric matrix. *J. ACM*, 5(4):339–342, 1958. doi: 10.1145/320941.320947. URL `http://doi.acm.org/10.1145/320941.320947`. 100

A. Hutton. Mdb integrator operation, December 2014. URL `http://gaia.esac.esa.int/dpacsvn/DPAC/CU1/docs/TechNotes/MDBIntegrator-Operation/GAIA-C1-TN-ESAC-AH-029.pdf`. 57

IBM. IBM Platform LSF . `http://www-03.ibm.com/systems/platformcomputing/products/lsf/`, August 2015. 134, 251, 288

R. Kohley. The Charge Injection Structure in Gaia CCDs, October 2012. URL `http://www.rssd.esa.int/cs/livelink/open/3143669`. 22

B. Lasker, M. Lattanzi, B. McLean, B. Bucciarelli, R. Drimmel, J. Garcia, G. Greene, F. Guglielmetti, C. Hanley, G. Hawkins, V. G. Laidler, C. Loomis, M. Meakes, R. Mignani, R. Morbidelli, J. Morrison, R. Pannunzio, A. Rosenberg, M. Sarasso, R. L. Smart, A. Spagna, C. R. Sturch, A. Volpicelli, R. L. White, D. Wolfe, and A. Zacchei. THE SECOND-GENERATION GUIDE STAR CATALOG: DESCRIPTION AND PROPERTIES. *The Astronomical Journal*, 136(2), Aug. 2008. doi: 10.1088/0004-6256/136/2/735. 287

D. C. leaders CU1. DPAC Software and System Specification, July 2012. URL `http://www.rssd.esa.int/cs/livelink/open/2786798`. 110

L. Lindegren. Representationof LSF and PSF for GDAAS-2, May 2003. URL `http://www.rssd.esa.int/cs/livelink/open/357835`. 97

L. Lindegren. Geometric calibration model for the 1M astrometric GIS demonstration, July 2008a. URL `http://www.rssd.esa.int/cs/livelink/open/510843`. 52

L. Lindegren. Modelling radiation damage effects for Gaia - A first-order Charge Distortion Model (CDM-01), April 2008b. URL `http://www.rssd.esa.int/cs/livelink/open/2824544`. 93

L. Lindegren. A general Maximum-Likelihood algorithm for model fitting to CCD sample data, November 2008c. URL `http://www.rssd.esa.int/cs/livelink/open/2861864`. 109

L. Lindegren. A framework for consistent definition and use of LSFs in AF/BP/RP/RVS, June 2009a. URL `http://www.rssd.esa.int/cs/livelink/open/2906236`. 97

L. Lindegren. Minimum-dimension LSF modelling, August 2009b. URL `http://www.rssd.esa.int/cs/livelink/open/2915742`. 97

L. Lindegren. LSF modelling with zero to many parameters, September 2010a. URL `http://www.rssd.esa.int/cs/livelink/open/3042524`. 97

L. Lindegren. A generic LSF/PSF model and its role in IDT, FL, IDU and AGIS, November 2010b. URL `http://www.rssd.esa.int/cs/livelink/open/3049410`. 97, 104

L. Lindegren. Detecting and handling micro-clanks in AGIS, September 2015. URL `http://www.rssd.esa.int/cs/livelink/open/3366779`. 36

L. Lindegren and U. Bastian. Basic principles of scanning space astrometry. In *GAIA: At the Frontiers of Astrometry*, volume 45, pages 109–114, January 2010. 17

L. Lindegren, C. Babusiaux, C. Bailer-Jones, U. Bastian, A. G. A. Brown, M. Cropper, E. Høg, C. Jordi, D. Katz, F. van Leeuwen, X. Luri, F. Mignard, J. H. J. de Bruijne, and T. Prusti. The Gaia mission: science, organization and present status. In *IAU Symposium*, volume 248 of *IAU Symposium*, pages 217–223, 2008. doi: 10.1017/S1743921308019133. URL `http://dx.doi.org/10.1017/S1743921308019133`. 29, 34

L. Lindegren, U. Lammers, D. Hobbs, W. O'Mullane, U. Bastian, and J. Hernández. The astrometric core solution for the Gaia mission. Overview of models, algorithms and software implementation. *A&A*, 538, 2012. URL `http://arxiv.org/abs/1112.4139`. 51

T. Lock. Software Engineering Standards (ECSS-E-40B) - Tailored for Gaia Science Ground Segment, October 2007. URL `http://www.rssd.esa.int/cs/livelink/open/2786522`. 141

X. Luri, W. O'Mullane, J. Alves, et al. Delivering the promise of Gaia, response to ESA's Announcement of Opportunity, February 2013. URL `http://www.rssd.esa.int/cs/livelink/open/3165853`. 45, 57

E. Mercier and J. Hoar. Gaia Science Ground Segment Operations Plan, July 2013. URL http://www.rssd.esa.int/cs/livelink/open/2892601. 41, 165

F. Mignard, C. Bailer-Jones, U. Bastian, R. Drimmel, L. Eyer, D. Katz, F. van Leeuwen, X. Luri, W. O'Mullane, X. Passot, D. Pourbaix, and T. Prusti. Gaia: organisation and challenges for the data processing. In *IAU Symposium*, volume 248 of *IAU Symposium*, pages 224–230, 2007. doi: 10.1017/S1743921308019145. URL http://dx.doi.org/10.1017/S1743921308019145. 3

A. Mora, F. Raison, and R. Kohley. Gaia PSF. Modelisation and detectability of faint sources near bright extended objects, June 2010. URL http://www.rssd.esa.int/cs/livelink/open/3010526. 63

A. Mora, U. Bastian, M. Biermann, F. Chassat, L. Lindegren, I. Serraller, and W. Serpell, E. van Reeven. The Gaia Basic angle: measurement and variations. In *Proceedings of The Milky Way Unravelled by Gaia. GREAT Science from the Gaia Data Releases*, December 2014. URL http://arxiv.org/abs/1503.02614v1. 19, 51

J. R. Myers, C. B. Sande, A. C. Miller, W. H. Warren, Jr., and D. A. Tracewell. SKY2000 Catalog, Version 4 (Myers+ 2002). *VizieR Online Data Catalog*, 5109:0, September 2001. 290

W. O'Mullane. *Implementing the Gaia Astrometric Solution.* PhD thesis, UB, 2012. URL http://hdl.handle.net/10803/83861. 51

W. O'Mullane, J. Hernandez, C. Huc, et al. Java coding standard and guidelines for DPAC, July 2006a. URL http://www.rssd.esa.int/cs/livelink/open/542649. 142

W. O'Mullane, L. U., C. Bailer-Jones, U. Bastian, A. Brown, R. Drimmel, L. Eyer, C. Huc, F. Jansen, D. Katz, L. Lindegren, D. Pourbaix, X. Luri, F. Mignard, J. Torra, and F. van Leeuwen. Gaia data processing architecture. In *ADASS XVI Proceedings*, 2006b. URL http://arxiv.org/abs/astro-ph/0611885. 48

W. O'Mullane, U. Lammers, L. Lindegren, J. Hernández, and D. Hobbs. Implementing the Gaia Astrometric Global Iterative Solution (AGIS) in Java. *Exper.Astron.*, 31:215–241, 2011. doi: 10.1007/s10686-011-9248-z. 52

M. Perryman. *The Making of History's Greatest Star Map.* Springer-Verlag, 2010. doi: 10.1007/978-3-642-11602-5. 2

M. A. C. Perryman, L. Lindegren, J. Kovalevsky, E. Hoeg, U. Bastian, P. L. Bernacca, M. Crézé, F. Donati, M. Grenon, M. Grewing, F. van Leeuwen, H. van der Marel, F. Mignard, C. A. Murray, R. S. Le Poole, H. Schrijver, C. Turon, F. Arenou, M. Froeschlé, and C. S. Petersen. The HIPPARCOS Catalogue. *A&A*, 323:L49–L52, July 1997. 287

M. A. C. Perryman, K. S. de Boer, G. Gilmore, E. Høg, M. G. Lattanzi, L. Lindegren, X. Luri, F. Mignard, O. Pace, and P. T. de Zeeuw. Gaia: Composition, formation and evolution of the galaxy. *A&A*, 369:339–363, April 2001. doi: 10.1051/0004-6361:20010085. 1

J. Portell and M. Clotet. DPCB Backup Policy, July 2015. URL `http://www.rssd.esa.int/cs/livelink/open/3057234`. 259

J. Portell, E. García-Berro, C. Estepa, J. Castañeda, and M. Clotet. Efficient data storage of astronomical data using HDF5 and PEC compression. In *High-Performance Computing in Remote Sensing (SPIE 8183)*, volume 8183, October 2011. doi: 10.1117/12.898203. URL `http://dx.doi.org/10.1117/12.898203`. 152, 254

J. Portell, J. Castañeda, and M. Clotet. DPCB interface control document, March 2013. URL `http://www.rssd.esa.int/cs/livelink/open/3086251`. 150

J. Portell, F. C, J. Torra, N. Garralda, J. González, and J. Castañeda. Daily processing of gaia data. In *Proceedings of the XI Scientific Meeting of the Spanish Astronomical Society*, 2014a. 46

J. Portell, M. Clotet, and N. Blagorodnova. DPCB Requirements Specification, October 2014b. URL `http://www.rssd.esa.int/cs/livelink/open/3079671`. 245

T. Prod'homme, B. Holl, L. Lindegren, and A. G. A. Brown. The impact of CCD radiation damage on Gaia astrometry - I. Effect of image location errors on the astrometric solution. *MNRAS*, 419:2995–3017, 2012. doi: 10.1111/j.1365-2966.2011.19934.x. 27

T. Prusti. Gaia Intermediate Data Release Scenario, October 2012. URL `http://www.rssd.esa.int/cs/livelink/open/3145458`. 41

S. Roeser, M. Demleitner, and E. Schilbach. The PPMXL Catalog of Positions and Proper Motions on the ICRS. Combining USNO-B1.0 and the Two Micron All Sky Survey (2MASS). *AJ*, 139:2440–2447, June 2010. doi: 10.1088/0004-6256/139/6/2440. 290

F. Schmuck and R. Haskin. GPFS: A Shared-Disk File System for Large Computing Clusters. In *FAST 2002 Conference on File and Storage*

Technologies, pages 231–244, January 2002. ISBN 1-880446-03-0. 151, 287

G. Seabroke, T. Prod'homme, N. Murray, C. Crowley, G. Hopkinson, A. Brown, R. Kohley, and A. Holland. Digging supplementary buried channels: investigating the notch architecture within the CCD pixels on ESA's Gaia satellite. *MNRAS*, 430:3155–3170, 2013. doi: 10.1093/mnras/stt121. 22

A. Short. Charge Distortion Model 02 (CDM02), February 2009. URL `http://www.rssd.esa.int/cs/livelink/open/2882372`. 93

A. Short. CDM03 Fortran 90 implementation, February 2011. URL `http://www.rssd.esa.int/cs/livelink/open/3058525`. 93

R. Smart. The Initial Gaia Source List and the Attitude Star Catalog, October 2013. URL `http://www.rssd.esa.int/cs/livelink/open/3223578`. 72

J. Snyder. *Map Projections: A Working Manual*. 1987. 287

A. Spagna and R. Messineo. Cleaning XM. Algorithm description and requirements, April 2011. URL `http://www.rssd.esa.int/cs/livelink/open/3072864`. 84

M. K. Szymański, A. Udalski, I. Soszyński, M. Kubiak, G. Pietrzyński, R. Poleski, Ł. Wyrzykowski, and K. Ulaczyk. The Optical Gravitational Lensing Experiment. OGLE-III Photometric Maps of the Galactic Bulge Fields. *Acta Astronomica*, 61:83–102, June 2011. 289

Tiobe. Tiobe list. `http://www.tiobe.com`, November 2012. 142

V. Valette and N. Dufourg. GTS Software Design Description, July 2011. URL `http://www.rssd.esa.int/cs/livelink/open/2890229`. 257

P. Valles, J. Portell, and J. Castañeda. IDV software user manual, January 2012. URL `http://www.rssd.esa.int/cs/livelink/open/3114354`. 110

F. van Leeuwen, A. Brown, C. Cacciari, et al. PhotPipe Software Requirement Specifications, September 2011. URL `http://www.rssd.esa.int/cs/livelink/open/3073809`. 53

M. Weiler. Modifications to CDM02/03, June 2013. URL `http://www.rssd.esa.int/cs/livelink/open/3212937`. 93

G. M. Williams, H. H. Marsh, and M. Hinds. Back-illuminated ccd imagers for high information content digital photography. In *Proceedings of SPIE 3302*, 1998. 21

N. Zacharias, S. E. Urban, M. I. Zacharias, G. L. Wycoff, D. M. Hall, D. G. Monet, and T. J. Rafferty. The Second US Naval Observatory CCD Astrograph Catalog (UCAC2). *AJ*, 127:3043–3059, May 2004. doi: 10.1086/386353. 291

Glossary

AC smearing AC image spread caused by the object motion in the AC direction. 96–99

MDB Integrator MDB Integrator is in charge of the integration of the source related results from the different processing systems involved in the Gaia data reduction. 57, 84, 122, 123, 212, 213, 234

ADS Airbus Defence and Space is a division of Airbus Group responsible for defence and aerospace products and services. Airbus Defence and Space was formed in January 2014 from the former EADS divisions Airbus Military, Astrium, and Cassidian. 3, 6

Anti-Blooming Drain Blooming dram occurs when the charge in a pixel exceeds the saturation level and the charge starts to fill adjacent pixels. Anti-blooming structures bleed off any excess charge before they can overflow the pixel and thereby stop blooming. 21

API An Application Programming Interface (API) is a set of routines, protocols, and tools for building software applications. An API specifies how software components should interact by expressing the software component in terms of its operations, inputs, outputs, and underlying types. 148

Arianespace Arianespace SA is a French-based multinational company founded in 1980 as the world's first commercial launch service provider. It undertakes the production, operation, and marketing of the Ariane 5 launch vehicle but it also operates the Soyuz-2 as a medium-lift alternative to Ariane 5. 7

Astrophysical Background The astrophysical background refers to the background signal observable in the Gaia images coming from the near objects, the zodiacal light, etc.. xxvii, 49, 53, 54, 61, 87–91, 105, 108, 111, 140, 285, 288

B-spline A B-spline is a piecewise polynomial function (generalization of a Bèzier curve) constructed so as to pass through a given set of points and have a certain number of continuous derivatives. These functions can then be used to interpolate intermediate values reducing the oscillation that can occur between points when interpolating using high degree polynomials. 34

Bias The bias is the electronic signal offset introduced on all samples during the image read–out by the PEM. xxvi, 49, 53, 54, 86–88, 91, 105, 108, 140, 285, 288, 290

BSC The Barcelona Supercomputing Center (BSC) is a public research center located in Barcelona, Catalonia, Spain. It hosts the Marenostrum supercomputer [BSC, 2014]. xiv, xv, xxxiv, 13, 117, 134, 136, 146, 151, 162, 170, 239, 245–248, 250, 251, 258–260, 285, 287

C++ A programming language based on the C language, offering an object-orientated programming model, but also supporting the procedural and functional models. The syntax of Java, and later, the C# language, were heavily influenced by the syntax of C++. There are many important differences between C++ and Java, and in many cases Java simplifies, and reduces the possibility of bugs. For example, C++ supports class multiple inheritance, as well as operator overloading, neither of which are currently available in Java. 142

CCD Readout In order to obtain a digital signal that is appropriate for doing quantitative analysis, it is necessary to convert the analog signal to a digital format. When light is gathered on a CCD and is ready to be read out, a series of serial shifts and parallel shifts occurs. First, the rows are shifted in the serial direction towards the serial register. Once in the serial register, the data is shifted in the parallel direction out of the serial register, into the output node, and then into the A/D converter where the analog data is converted into a digital signal. 290

centroid The centroid designates the point at the centre of any shape, sometimes called centre of area or centre of volume. The coordinates of the centroid are the average (arithmetic mean) of the coordinates of all the points of the shape. 97, 102, 104

CI The charge injections are artificial charges injected in the first pixel line of the Gaia CCDs which are subsequently transferred across the whole CCD. They are used to temporarily fill a large fraction of the traps present in the CCD and prevent the charge trapping of the following photoelectrons generated and transferred through the

CCD. xxiii, xxiv, xxvii, 11, 22, 24, 27, 29, 33, 39, 47, 88–92, 104–108, 111

COMA-CAL The COlour-MAgnitude terms of the AGIS geometric calibration solution. 52

Cross-Match The cross-match provides the link information between the observation and the Gaia source catalogue. This information is produced initially by IDT and updated by IDU. 34, 49, 50, 54, 57, 59, 65, 67, 71–74, 83, 108, 112, 113, 140, 169, 172, 176, 230, 234, 240

CU3-Torino CU3 team from OATo-INAF contributing in the design and development of the cross-match tasks integrated in IDT and IDU. 46, 55, 75, 84, 139

CU3-UB CU3 team from DAM-ICC-IEEC-UB contributing in the design and development of IDT and IDU systems. 46, 55, 64, 71, 75, 85, 88, 92, 109, 139–141

CU5-DU10 CU5 team from IfA-ROE contributing in the design and development of several calibration tasks integrated in IDT and IDU, including Bias, Astrophysical Background, PEM-NU and LSF/PSF. 46, 55, 140

DPCB DPCB is the Data Processing Center in Barcelona (Spain). DPCB is partipated by the University of Barcelona (UB) and is composed of two different institutions, namely, the BSC and the Consorci de Serveis Universitaris de Catalunya (CSUC). xxvii, xxviii, xxxiv, 44, 58, 60, 66, 73, 85, 110, 117–120, 122–125, 130, 132, 137–142, 145, 150–155, 158, 162, 163, 167–171, 181, 211, 214, 227, 228, 230, 234, 238, 241, 245–248, 250, 253, 258, 259, 285

DpcbTools One of the DPAC systems developed to assist the execution of other DPAC systems at the Data Processing Centre Barcelona (see DPCB). DpcbTools contains utilities for performing I/O; data manipulation; communication; task creation, scheduling and launching; data visualisation; and monitoring operations [Castañeda et al., 2009a]. xxxiv, 60, 136, 140, 152, 157, 250, 251, 254, 259

DPCC DPCC is the Data Processing Center provided by CNES (French space agency) hosting the processing center for the CU4 (Objects Processing), CU6 (Spectroscopic processing) and CU8 (Astrophysical Parameters). 44, 56

DPCE DPCE is the Data Processing Center at the European Space Astronomy Center (ESAC) in Madrid belonging to the European Space

Agency (ESA). 44–47, 49, 52, 56, 57, 122, 125, 137, 138, 145, 151, 165–170, 172, 174, 175, 228–230, 238, 241, 242, 245, 257–259, 291

DPCG DPCG is the Data Processing Center in Geneva (Switzerland) dedicated to the detection and characterization of variable sources (CU7). 44

DPCI DPCI is the Data Processing Center in Cambridge (United Kingdom) and is responsible for the operation of the main photometric pipeline. 44, 47, 54, 166

DPCT DPCT is the Data Processing Center in Torino (Italy) and participates in the core processing of the Gaia data (CU3). DPCT is partipatec by the Astronomical Observatory of Torino and the ALTEC industry team. 44

DS Gaia mission Data Segments refer to a well-defined continuous block of data. drcs are adjusted to this data block typically of six months duration. xxvii, 55, 56, 119, 120, 123, 125, 130, 132, 162, 165–172, 174, 175, 201, 210, 211, 214, 217, 228, 230, 234, 235, 238

Fiducial Line The fiducial line for a particular CCD can be thought of as the central line of pixels, half-way between the observation Gate and the read out lines. It can also be understood as the line over the CCD corresponding to the half integrated AL area. In reality the definition of the fiducial observation line is a bit more complex, as some of the pixel lines are blocked out by an aluminium mask. 62, 63

Gate Gates are special lines within the light-sensitive area of the CCDs. When activated they act like summing registers, holding up charges, i.e. preventing them from moving/accumulating along scan in spite of the TDI clocking. This causes a collapse of the already accumulated TDI images into a single line, and the start of the charge from scratch from this line onwards. xxiv, 21, 22, 24, 29, 33, 39, 286

GEO-CAL The purely geometric terms of the AGIS geometric calibration solution. 52, 103

GEPC The Gaia Ecliptic Pole Catalogue, version 3.0 [Altmann and Bastian, 2009], 6.1×10^6 entries in approximately 1 square degree around the North and Southern Ecliptic Poles produced specifically for the calibration of Gaia. 72

GoF The Goodness of Fit (GoF) of a statistical model describes how well it fits a set of observations. Measures of goodness of fit typically

summarize the discrepancy between observed values and the values expected under the model in question. 109

GPFS The General Parallel File System (GPFS) is a high-performance clustered file system developed by IBM. It is designed to allow many concurrent accesses, while maintaining high performance. It also provides fault tolerance, and optimal use of the storages devices. The technology can involve splitting files into small blocks, and then storing these blocks across several disks, thus obtaining the total combined bandwidth of all the disks [Schmuck and Haskin, 2002]. 151, 152, 247, 260

Greasy Greasy is a tool offered by BSC to the HPC applications for the execution of parallel jobs in a computer cluster. xxviii, 134–137, 146, 149, 214, 215, 217

GS The ground segment refers to an ensemble of facilities responsible for the acquisition, processing, distribution and archiving of the satellite data and of the derived products. 41, 42, 47

GSC2.3 The Second Guide Star Catalogue version 2.3 [Lasker et al., 2008], 9.4x10^8 objects all sky, magnitude limit R_F 21.5. This catalogue forms the bulk of the photometry and defines the red and blue magnitudes as this is the sky survey with the largest number of objects on a homogeneous system. xxvi, xxxii, 72, 73, 195, 196

Hammer-Aitoff Projection The Hammer-Aitoff equal-area projection, also called the Hammer projection, is a map projection that is a modification of the Lambert azimuthal equal-area projection. It consists of halving the vertical coordinates of the equatorial aspect of one hemisphere and doubling the values of the meridians from the centre [Snyder, 1987]. Like the Lambert azimuthal equal-area projection, it is equal area, but it is no longer azimuthal. 8, 111

HEALPix Hierarchical Equal-Area iso-Latitude Pixelisation [Gorski et al., 2005]. HEALPix largely improves the performance of other tessellation techniques, such as HTM, in terms of computing time but perhaps the most important one if the fact that the pixels obtained using it all have the same area. More details on the usage of this pixelisation in Gaia is available in Castañeda [2008]. xxviii, xxxiii, xxxiv, 72, 73, 79, 83, 85, 111, 125, 126, 128, 129, 144, 218, 222, 223, 233, 291

HIPPARCOS Hipparcos catalogue [Perryman et al., 1997], 117x10^3 entries. 72

IBM-LSF IBM Platform Load Sharing Facility is a powerful workload management platform for demanding, distributed HPC environments. It provides a comprehensive set of intelligent, policy-driven scheduling features that enable you to utilize all of your compute infrastructure resources and ensure optimal application performance [IBM, 2015]. 132, 134, 248, 251

IDU-CAL IDU tasks providing Gaia instrument calibrations: Bias, Astrophysical Background and LSF/PSF. 60, 245

IDU-IPD IDU tasks involved in the redetermination of the Image Parameters. 60, 245

IDU-SDM IDU tasks providing the connection between Gaia observations and the actual sources or SSOs in the sky. 59, 245

Image Parameters The image parameters refers to the parameters that can be derived from the processing of the Gaia images. These parameters are mainly the estimated image centroid (AL and AC centroid location) and the estimated object flux obtained after a fitting process of a LSF/PSF model. 18, 50, 54, 59, 60, 76, 87, 103, 105–109, 113, 288

JMS The Java Message Service (JMS) is a Java API that allows applications to create, send, receive, and read messages using reliable, asynchronous, loosely coupled communication. . 147

JNI Java Native Interface (JNI) is a programming framework that enables Java code running in a Java Virtual Machine (JVM) to call and be called by native applications (programs specific to a hardware and operating system platform) and libraries written in other languages such as C, C++ and assembly. 254

Lissajous Orbit Lissajous Orbit is a quasi-periodic orbital trajectory that an object can follow around a Lagrangian point of a three-body system without requiring any propulsion. 7

LQRF Large Quasar Reference Frame [Andrei et al., 2009], 1.7x10^5 QSOs, magnitude limit R_F 22 and mostly fainter than R_F 18. This is a compilation of Quasi-Stellar Object (QSO)s with precise positions produced as part of the Gaia auxiliary catalogue development. 72

Marenostrum MareNostrum is the most powerful supercomputer in Spain and one of seven supercomputers in the Spanish Supercomputing Network. ix–xi, xiv, xxviii, xxix, 13, 15, 117, 133, 136, 148, 150, 151, 154, 158–160, 162, 163, 214, 215, 217, 239, 246–248, 251, 284

MDB Gaia Main Database holding both the raw and intermediate data produced in the data processing pipeline of Gaia. in that sense, the MDB is central repository for all Gaia mission data. 14, 46, 47, 55–57, 84, 112, 122, 123, 138, 140, 212, 213, 234, 257, 258, 283

Message Passing Interface A message passing system widely used in computing for developing parallel software applications. Its is particularly used in distributed memory systems. MPI bindings are available for a number of languages including Fortran, C, C++ and Java. 154

MLE The Maximum-Likelihood Estimation (MLE) is a method of estimating the parameters of a statistical model. When applied to a data set and given a statistical model, maximum-likelihood estimation provides estimates for all the parameters of the model. 109, 114

MOC The Mission Operation Centre (MOC) is located at the European Space Operations Centre (ESOC) in Darmstadt, Germany. It is responsible for spacecraft operations and associated ground segment development. It is also in charge of providing the downlinked science data along with the relevant housekeeping data to the Science Operation Centre (SOC). 45, 47, 49, 291

OBMT The On Board Mission Timeline is a time coordinate system derived from the actual reading in units of nanoseconds of the master clock controlling all operations in Gaia spacecraft. 55, 171, 172

OGLE Optical Gravitational Lensing Experiment version III [Szymański et al., 2011], $2.2\text{x}10^8$ objects in the bulge, LMC, SMC and Southern-EPC. This catalogue was included at the request of the Gaia science alerts team to improve the large incompleteness of the IGSL in the very crowded regions. 72

Payload Module Payload is the carrying capacity of an aircraft or launch vehicle, usually measured in terms of weight. Depending on the nature of the flight or mission, the payload of a vehicle may include cargo, passengers, flight crew, munitions, scientific instruments or experiments, or other equipment. xxiii, 3–5, 7

PCA Principal Component Analysis is a statistical procedure that uses an orthogonal transformation to convert a set of observations of possibly correlated variables into a set of values of linearly uncorrelated variables called principal components. 97

PPMXL Positions and Proper Motions "Extra Large" Catalogue, [Roeser et al., 2010], 9.1x10^8 entries. The positions and proper motions should be the most precise available for the objects fainter than the UCAC4 limit. xxvi, xxxii, 72, 73, 195, 196

Pre-Scan Additional pixels in the Gaia CCDs (columns 0 to 13) which are not fed with photoelectric charges (outside the CCD illuminated area) but nevertheless are always read out. They are used for the calibration of the Bias. 26, 30, 32, 86, 87, 131

Quaternion In mathematics, the quaternions are a number system that extends the complex numbers. A quaternion is a four-element vector that can be used to encode any rotation in a 3D coordinate system. Technically, a quaternion is composed of one real element and three complex elements. 34, 112

RMI The Java Remote Method Invocation (Java RMI) is a Java API that allows invoking methods from a remote Java virtual machine supporting the direct transfer of serialized Java classes. 147

RRD Round-Robin Databases are aimed to handle time–series data. The data are stored in a circular buffer based database, thus the system storage footprint remains constant over time. 112

RSE The Robust Scatter Estimator (RSE) is defined as $RSE = 0.390152\times(P_{90} - P_{10})$, where P_{10} and P_{90} are the 10^{th} and 90^{th} percentiles, and the numerical constant is chosen to make RSE equal to the standard deviation for a gaussian variable. 111

SDSS Sloan Digital Sky Survey data release 9 (`http://www.sdss.org`), 4.7x10^8 entries, magnitude limit $R_F < 22$. This catalogue provides precise astrometry, photometry and classification for one quarter of the sky. xxvi, 72, 73

SEA Source Environment Analysis. 54, 66, 71, 106

Serial Register A row of pixels adjacent to the parallel register. When the CCD is exposed to light, the serial register receives charge from the parallel register and shifts it to the output node to form an image. Also called a horizontal register. See CCD Readout. 26, 28, 29

Sky2000 The SKYMAP Master Catalogue, Version 4 [Myers et al., 2001], 300x10^3 entries brighter than 8.0 magnitude. 72

SNR Signal-to-noise ratio is a measure used in science and engineering that compares the level of a desired signal to the level of background

noise. It is defined as the ratio of signal power to the noise power. 24, 26

SOC The Science Operation Centre (SOC) is at Data Processing Center ESAC (DPCE). It coordinates the distribution of the data received from the Mission Operation Centre (MOC) to the relevant Coordination Units and the six Data Processing Centres. . 45, 289

solutionId The solution identifier is a numeric field in order to uniquely tag the data produced by each processing systems. This field can be used to track the software version and the DRC when a specific data set was generated. xxviii, 86, 137, 138, 140, 146, 150, 151, 169, 213, 230

sourceId The source identifier is a numeric field in order to uniquely tag the entries from the Gaia catalogue. This numeric field codes a HEALPix spatial index, the DPC producer and a running number which is increased on new source creation. 72, 83, 85, 86, 212, 213, 226

spline In mathematics, a spline is a numeric function that is piecewise-defined by polynomial functions, and which possesses a sufficiently high degree of smoothness at the places where the polynomial pieces connect (which are known as knots). 96, 99

Star Tracker Navigational device which measures the angular separation of stars with reference to a known time and place in order to achieve precise navigation. 5

Tycho-2 Tycho-2 Catalogue [Høg et al., 2000], 2.4x10^6 stars, magnitude limit $R_F < 12$. This catalogue forms the backbone of all the major ground based catalogues currently available. The astrometric information is mostly superseded by UCAC4 however this catalogue provides the photometric information for most objects of this brightness. 72

UCAC4 USNO CCD Astrograph Catalogue version 4 [Zacharias et al., 2004], 1.1x10^8 entries mostly stars, magnitude limit $R_F < 17$. This is the most precise astrometric catalogue in the range V=11-16 currently available that is all sky. 72, 290, 291

XML Extensible Markup Language (XML) is a markup language that defines a set of rules for encoding documents in a format which is both human-readable and machine-readable. 132, 144, 145

Acronyms

AC Across-Scan. xxvii, 22, 23, 26, 28, 29, 32, 38, 50, 63, 76, 77, 89, 96–99, 102, 103, 105, 107, 112, 113, 172, 178, 191, 283, 288

AF Astrometric Field. xxiii–xxv, xxvii, 11, 20, 25, 26, 29–32, 44, 52, 59, 61–64, 76, 86, 88, 90–92, 107

AGIS Astrometic Global Iterative Solution. xxv, 14, 39, 49–52, 54, 55, 57, 58, 102, 103, 105, 106, 109, 113–115, 122, 166, 240–243, 285, 286

AL Along-Scan. xxiv, xxvii, 22, 23, 28, 29, 32, 38, 62, 63, 66, 77, 89, 96, 98, 99, 103, 106, 107, 112, 113, 178, 191, 192, 286, 288

AOCS Attitude and Orbit Control sub-System. 5, 34, 107

ASC Attitude Source Catalogue. 34

ASD Auxiliary Science Data. 31, 33, 46, 86, 88

ASD7 Object Log. 31, 33, 67, 88

BA Basic Angle. 19, 20, 22, 37, 51

BAM Basic-Angle Monitor. xxiv, 19, 20, 31, 32

BCRS/ICRS Barycentric International Celestial Reference System. 36

BP Blue Photometer. xxiii, xxiv, 12, 20, 21, 25

BP/RP Blue and Red Photometers. 26, 30–32, 44, 52, 108

CCB Configuration Control Board. 122, 142, 143, 149

CCD Charge-Coupled Device see. xxiii–xxvi, 6, 10–12, 14, 17–33, 37–39, 47, 49, 52, 53, 61–65, 69–71, 86, 89–91, 93, 96, 99–101, 103, 107, 108, 112, 126, 140, 284–286, 290

CDM Charge Distortion Model. xxvii, 92–94, 98, 114, 115

CNES Centre National d'Études Spatiales. 166

CoMRS Center-of-Masses Reference System. 36, 37

CR Charge Release. 88–92, 105, 108

CRNU Column Response Non Uniformity. 47

CSUC Consorci de Serveis Universitaris de Catalunya. xxxiv, 245, 246, 249, 285

CTI Charge Transfer Inefficiency. xxiv, 17, 22, 27–30, 39, 52, 53, 61, 89, 91, 93, 96, 98–101, 107, 114, 178, 241

CU Coordination Unit. xxv, 43–45, 58, 67, 115, 120, 122, 142, 155, 240

CU1 Coordination Unit 1. 44, 45, 58, 140, 145, 152, 245

CU2 Coordination Unit 2. 44, 45, 136, 139, 245, 246, 248, 250

CU3 Coordination Unit 3. 44, 45, 140, 167, 245, 250, 285

CU4 Coordination Unit 4. 44, 45, 66, 71, 77, 106

CU5 Coordination Unit 5. 44, 45, 58, 66, 71, 106, 285

CU6 Coordination Unit 6. 44

CU7 Coordination Unit 7. 44

CU8 Coordination Unit 8. 44

CU9 Coordination Unit 9. 45, 246, 250

DAM-ICC-IEEC-UB Departament d'Astronomia i Meteorologia, Institut de Ciències del Cosmos (ICC), Institut d'Estudis Espacials de Catalunya (IEEC), Universitat de Barcelona (UB). xv, 285

DC Detection Classifier. 140

DM Data Model. 46, 55, 56, 58, 122, 123, 140, 153, 169, 254

DPAC Data Analysis and Processing Consortium. xv, xxv, xxvii, 42, 43, 45, 54–56, 87, 106, 107, 110, 118, 119, 125, 139–142, 145, 155, 162, 163, 166, 230, 237, 238, 241, 245–248, 250, 251, 253,
Glossary: Data Analysis and Processing Consortium (DPAC)

DPACE Data Analysis and Processing Consortium Executive. 43, 166, 170, 235

DPC Data Processing Center. xxv, 42–44, 47, 56, 58, 61, 72, 86, 122–124, 132, 138, 142, 145, 151, 152, 155, 167–169, 238, 241, 257, 258, 260, 291

DRC Data Reduction Cycle. xxvii, 41, 42, 45, 50, 53, 56, 57, 63, 66, 71, 72, 81, 85, 87, 89, 95, 104, 106, 119–123, 125, 130, 132, 149–151, 155, 162, 163, 169, 212, 230, 238, 243, 245, 247, 257, 291

ECSS European Cooperation for Space Standardization. 141, 154

ELSF Empirical LSF/PSF. xxvii, 94, 95, 98, 99, 101, 107, 115, 241

EPC Ecliptic Pole Catalogue. 289

ESA European Space Agency. xiii, 1–3, 8, 33, 41, 43

ESOC European Space Operations Centre. 45

FL First Look. 47, 50, 57, 87–89, 92, 94, 109, 165, 168

FoV Field of View. xxvi, xxvii, 38, 63, 66, 68, 70, 83, 88, 99, 106, 112, 113, 182, 231

FoVRS Field-of-View Reference Systems. 37

FPRS Focal Plane Reference System. 38

GBIN Gaia Binary File Format. 140, 152, 253–255

GST Gaia Science Team . 43,
Glossary:

GTS Gaia Transfer System. 47, 57, 124, 125, 140, 248, 257, 258

HDF Hierarchical Data Format. 152, 163, 251, 253–255

Hipparcos HIgh Precision PARallax COllecting Satellite. xxx, 2, 3, 34, 181, 232, 287

HPC High Performance Computing. 132, 134, 287

HTM Hierarchical Triangular Mesh. 287

ICD Interface Control Document. 44, 48, 55, 58, 112, 257

ICRS International Celestial Reference System. 51, 111

IDT Intermediate Data Treatment. 46, 47, 49, 50, 57, 66, 67, 70, 71, 77, 78, 85, 87, 88, 92, 94, 103, 106, 108, 113, 118, 121, 124, 140, 165, 166, 168–170, 174–176, 207, 213, 229, 231, 233–235, 241, 242, 245, 249, 250, 285

IDU Intermediate Data Updating. xiii, xiv, xxv, xxvii–xxix, 13–15, 38, 39, 49, 50, 54, 55, 57–61, 66, 67, 70, 72, 76, 84, 85, 87–89, 92, 94, 95, 102, 104, 106, 109, 110, 113, 115, 117–125, 130, 132–134, 136–143, 145, 146, 148–155, 157, 158, 161–163, 165, 167–171, 175, 207, 214, 216, 230, 234, 235, 237–243, 245–248, 254, 259, 263, 285, 288

IDU-APB IDU Astrophysical Background Calibration task. xxviii, 60, 88, 91, 92, 95, 101, 108, 122, 130, 139, 156

IDU-BIAS IDU Bias Calibration task. 60, 86, 88, 92, 95, 122, 130, 131, 139

IDU-DC IDU Detection Classifier. xxvi, 59, 65, 70–72, 112, 113, 122, 130, 167, 169, 171, 176, 180, 210, 214, 215, 218, 238, 240, 243

IDU-IPD IDU Image Parameters Determination. xxix, 60, 106, 108, 109, 113–115, 122, 130, 140, 157, 240, 242, 243

IDU-LSF/PSF IDU LSF/PSF Calibration task. 60, 108, 122, 130, 139, 241–243

IDU-SCN IDU Scene. xxv, xxviii, 59, 61–64, 70, 91, 92, 101, 106, 112, 114, 122, 130, 131, 140, 156, 167, 169, 181, 183, 221, 240

IDU-XM IDU Cross–Match. xxvi, 15, 58, 59, 71, 73, 75–79, 82, 86, 103, 112–114, 122, 130, 131, 137, 138, 149, 167, 169, 174, 176, 189, 204, 212, 214, 217, 225, 230, 231, 233, 234, 238, 243

IDV Intermediate Data Validation. 110, 113

IfA-ROE Institute for Astronomy - Royal Observatory of Edinburgh. xxvi, xxvii, 88, 90, 92, 95, 98, 102, 109, 115, 139, 140, 285

IGSL Initial Gaia Source List. xxvi, 72, 73, 128, 176, 189, 192, 195, 199, 205, 208, 213, 232, 235, 289

IOGA Initial On–Ground Attitude. 34

IPD Image Parameters Determination. xxvii, 57, 101, 105–108, 111, 114, 157

LMC Large Magellanic Cloud (special, high-density area on the sky). 289

LSF Line Spread Function. xxvii, 96–101, 103

LSF/PSF Line/Point Spread Function. xxvii, 39, 47, 49, 50, 52–54, 61, 64, 87, 92–97, 101–105, 107, 109, 114, 115, 140, 240, 241, 285, 288, 295, 298

MLA Multi-Lateral Agreement. 43

MPS MicroPropulsion Sub-system. 32

OATo-INAF Astronomical Observatory of Torino - Istituto Nazionale di Astrofisica. 285

OGA On–Ground Attitude. 47

OGA1 On–Ground Attitude. 46

OO Object-Oriented. 142, *Glossary:* Object-Oriented (OO)

OS Operation System. 158

PAA Phased-Array Antenna. 32

PDHU Payload Data Handling Unit. 5

PEM Proximity Electronics Module. 26, 27, 47, 86, 87, 284, 297

PEM-NU PEM Non-Uniformity. 26, 30, 49, 87, 88, 91, 92, 285

PhotPipe Photometric Pipeline. xxv, 14, 49, 50, 53–55, 57, 166, 169, 240

PSF Point Spread Function. xxvii, 96–99

QSO Quasi-Stellar Object. 288

RP Red Photometer. xxiii, xxiv, 12, 20, 21, 25

RS Reference System. xxiv, 23, 36–38

RVS Radial-Velocity Spectrometer. xxiv, 12, 20, 21, 25, 26, 30–32, 44

SBC Supplementary Buried Channel. 22

SM Sky Mapper. xxiii–xxv, xxvii, 11, 20, 24–26, 29, 31, 32, 59, 61, 62, 64, 65, 76, 86, 88, 90–92

SMC Small Magellanic Cloud (special, high-density area on the sky). 289

SMO Suspected Moving Object. 32

SP Star Packets. 33, 46

SP1 SM/AF/BP/RP Star Packet. 31–33

SP2 RVS Star Packet. 31–33

SRS Scanning Reference System. xxv, 36, 37, 64

SSO Solar System Object. xxv, 25, 44, 59, 61–63, 66, 70, 77, 130, 131, 182, 183, 231, 240, 288

TDI Time Delayed Integration. xxiii, 11, 24, 26, 28, 29, 63, 96, 192

UB Universitat de Barcelona. 170, 245

VBS Very Bright Source. 61, 70, 240

VO Virtual Object. 27, 76, 89–91

VPU Video Processing Unit. 19, 20, 27, 32, 39, 62

WFS WaveFront Sensor. xxiv, 20, 31, 32

WRS Window Reference System. 38, 39, 63

XELSF eXtended Empirical LSF/PSF. 94

Notes

Gaia is an extremely ambitious astrometric space mission adopted within the scientific programme of the European Space Agency (ESA) in October 2000. It aims to measure with very high accuracy the positions, motions and parallaxes of a large number of stars and galactic objects, including also for almost all the objects information about their brightness, colour, radial velocity, orbits and astrophysical parameters. Gaia requires a demanding data processing system on both data volume and processing power. The treatment of the Gaia data has been designed as an iterative process between several systems each one solving different aspects of the data reduction system.

In this thesis we address the design and implementation of the Intermediate Data Updating (IDU) system. The IDU is the instrument calibration and data processing system more demanding in data volume and processing power of Gaia. Without this system, Gaia would not be able to provide the envisaged accuracies and its presence is fundamental to achieve the optimum convergence of the iterative process on which all the data processing of the spacecraft is based.

Universitat de Barcelona

Gaia DPAC Data Processing & Analysis Consortium