18th International Conference on Space Operations, Montreal, Canada, 26 - 30 May 2025.

SpaceOps-2025, ID # 121

Finding asteroids in a fly's component eye: The challenge of developing a data processing pipeline for the Flyeye telescope

Johannes Klug [a], **Daniel Fischer** [b], **Ernesto Doelling** [c], **Jiri Doubek** [d]

[a] *European Space Agency*, *Johannes.Klug@esa.int*
[b] *European Space Agency*, *Daniel.Fischer@esa.int*
[c] *European Space Agency*, *Ernesto.Doelling@esa.int*
[d] *Iguassu Software Systems, jiri.doubek@iguassu.cz*

Abstract

ESA is currently building the novel Flyeye telescope, to be used in surveys and follow-up observations of asteroids potentially threatening Earth. A key component of the overall Flyeye system is its image processing pipeline, called "Flyeye Data Processing Chain (DPC)". Flyeye DPC processes raw images captured by the telescope's 16 high-resolution cameras and extracts accurate detections of asteroids. Flyeye DPC is built from several subsystems with (pre-)operational heritage, but also new functionality. In this paper, we describe the overall development and integration process, including meeting the demanding performance requirements. Furthermore, we discuss the operational concept and give a first glimpse into the detection performance based on several simulated datasets. Finally, we outline the next steps for Flyeye DPC, and share some ideas for future developments.

Keywords: (data processing, asteroids, detection, FlyEye, telescope)

Acronyms/Abbreviations

Asteroid and Satellite Automatic Processor (ASAP)
Continuous Deployment (CD)
Continuous Integration (CI)
Data Processing Chain (DPC)
European Space Agency (ESA)
File Transfer Protocol (FTP)
Flexible Image Transport System (FITS)
Intellectual Property Rights (IPR)
Italian Space Agency (ASI)
Minor Planet Center (MPC)
Simple Network Management Protocol (SNMP)
Site Acceptance Test (SAT)
Site Equipment Controller (SEC)

1. Introduction

Asteroids pose a constant threat to Earth. Within ESA's Space Safety Programme [1], the Planetary Defence Office is in charge of asteroid observation and threat evaluation. Depending on the relative positions, asteroids are visible to terrestrial telescopes. In a single image, they appear as a faint dot, looking like a star. When the same general area is observed again a while later, an asteroid will have moved with respect to the fixed stars around it. If this process is repeated a few times, one can begin to see the characteristic movement of an asteroid within the night sky. Its movement's arc can be used to calculate its orbital trajectory. Of course, many observations are needed to succeed in such a detection. The Flyeye system is currently under construction to provide such high-quality observations. A first Flyeye telescope, Flyeye-1, will be situated atop Mt Mufara on the island of Sicily in Italy. Flyeye-1 has a impressively wide field of view of 45 sq. degrees, utilising 16 cameras simultaneously. It is capable of surveying the entire night sky approximately every 3 days, detecting faint objects down to 21 mag. The resulting images are then fed into a bespoke processing pipeline, called "Flyeye Data Processing Chain (DPC)". The Flyeye DPC is developed by a

European consortium led by Iguassu Software Systems (ISS) [2] as Prime Contractor. Development of this software system is in its final stages, but already now we can share some of the challenges overcome on its way to operational usage.

2. Flyeye Data Processing Chain (DPC) Architecture

This section discusses the overall DPC architecture and its constituent components. All of the bespoke software falls under ESA's Operational Software clause, implying Intellectual Property Rights (IPR) belonging to ESA. Of course, like in many contemporary software projects, open-source software is used wherever possible.

2.1. Overview

The Flyeye DPC runs across three dedicated physical servers, which in itself posed a challenge – more on that later. During a typical night in routine operations, all 16 cameras of Flyeye-1 are active, taking images with exposure times of around 40 to 60 seconds, followed by roughly 10 seconds of image transmission and re-pointing. The cameras actively transmit their images in FITS (Flexible Image Transport System) format into the DPC in-tray via file transfer. The below figure gives an overview, with the following sections describing the individual subsystems.

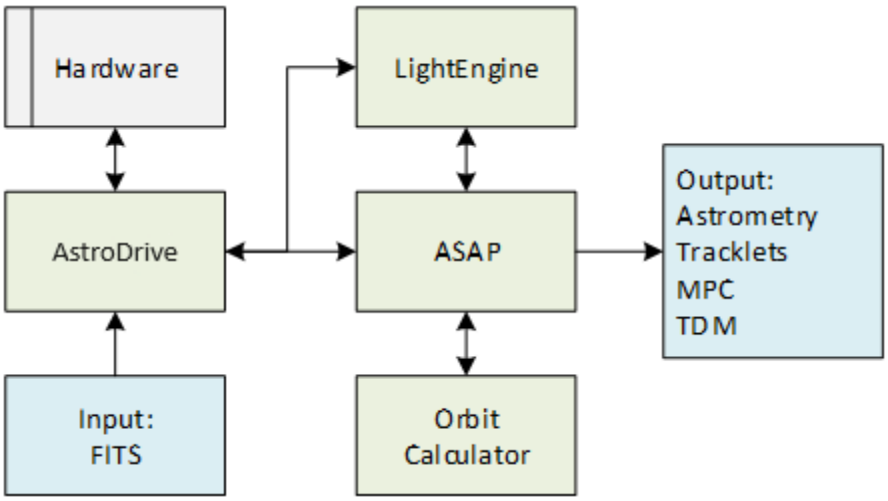

Figure 1 Flyeye DPC architecture overview

This paper focuses on three components, namely AstroDrive, LightEngine, and Asteroid and Satellite Automatic Processor (ASAP). All three subsystems have years of operational heritage and are developed by European industry. They were adapted for use in the Flyeye DPC.

2.2. AstroDrive

From the DPC in-tray, each FITS image is picked up and archived in a custom component called AstroDrive, acting as the short-term archive for the overall system, as well as the repository for storing intermediate image versions during pipeline runs.

Images are notably treated as immutable, meaning that modifications always result in a new version of that image, preserving the original.

AstroDrive uses a relational database to store image meta-data and offers a web-based user interface to browse and inspect image files, including a FITS viewer.

2.3. LightEngine

After reception of a given FITS file, the next steps in the processing pipeline are performed by LightEngine. It applies basic photometric correction using dark, flat, and bias fields. These are re-created once per processing night for calibration purposes.

Additionally, known bad pixels are masked, and cosmic rays removed. LightEngine then establishes which exact part of the night sky is visible in a given image (plate solving). Known objects from the Gaia catalogue are discarded, resulting in a set of objects to be further processed. This information is passed on in textual format to the next step.

2.4. ASAP

The Asteroid and Satellite Automatic Processor (ASAP) is the subsystem tasked with finding needles in a proverbial haystack. As the name suggests, it is also capable of satellite and space debris tracking. For usage in Flyeye DPC, it was tuned specifically for asteroid workloads. The exact configuration is still under evaluation and will be fine-tuned once the system is in operation with real telescope images coming in.

ASAP uses LightEngine astrometric output and it matches asteroid candidates to known ones. If a match is found, that is good news, as it provides a refined observation of a known object, leading to a more accurate and up-to-date orbit determination. If a candidate has no match, that could be arguably even better news, as it could indicate the discovery of a previously unknown asteroid.

Those two detection cases map onto the two basic modes of Flyeye operation, which are follow-ups of known objects, and a general survey mode. ASAP of course will perform its processing thoroughly regardless of operations mode, meaning it can find completely new objects also in frames taken for the purpose of following up a known object.

ASAP comes with a web-based user interface, which is the main workbench for astronomers supervising the data processing chain. After a night of observations and processing, astronomers will check the results, and issue good observations to the Minor Planet Center (MPC).

3. Challenges

This section points out some of the challenges met and overcome during the development of the Flyeye DPC. This is a subset of challenges, necessarily, and should be complemented by a report once the Flyeye DPC has successfully entered routine operations.

3.1. Adaptation of heritage software

The three core subsystems described in section 2 each have operational heritage, both within European industry, and even within ESA: an earlier version of those systems is used in the processing system of ESA's robotic "test-bed telescopes" in Cebreros, Spain, as well as La Silla in Chile.

The decision to use heritage software was taken early in the Flyeye DPC procurement process. It was estimated that re-using and adapting existing European software would result in a quicker and cheaper procurement overall. The drawbacks are less freedom in software design and a potentially more challenging integration. In fact, all three subsystems have seen substantial changes and additional developments to adapt them for the Flyeye DPC.

We would like to point out two particular challenges, one legal, another one technical.

On the legal front, ESA requires the possibility to maintain and evolve the software making up its operational systems, to ensure smooth operations throughout mission, or in this case, observatory lifetime. Should an industrial supplier go out of business or be otherwise no longer interested in maintaining a system, ESA has the option of procuring support from other capable companies within its member states.

In the case of the Flyeye DPC, a key component's Intellectual Property Rights (IPR) were owned by a member of the implementing consortium. It was agreed that this IPR was transferred to ESA, satisfying the Operational Software requirements. The result is a system made up of ESA IPR source code for key components, plus permissible open-source licensed source code, plus software licensed from European industry. This is in line with ESA's legal and operational requirements.

The technical challenge in adopting heritage software in this case stems from ESA's preference for Linux-based operational systems. Some key components are written in the C# language for Microsoft's Dotnet runtime. Our industrial partners develop, and in some cases, operate this software on a Microsoft Windows platform. For ESA's operational infrastructure teams, this is not possible, however. As a mitigation, the Flyeye DPC development team early on targeted Dotnet Core as the runtime, which would allow deployment on Linux. Some code changes were required; however, they were luckily not too dramatic in scope or depth. Now, there is no longer a dependency on the Microsoft Windows platform for build or deployment. We will touch on this point later.

3.2. Operational Concept

During the development time of the Flyeye DPC, the overall Flyeye operational concept has become much clearer. When DPC development begun, there hadn't been a decision yet on where the Flyeye telescope was even going to be

located. Key information about the site was therefore unknown, most importantly accessibility by operators, and type and performance of its internet uplink. As a worst case, it was assumed that the telescope site might be inaccessible due to snowy conditions for weeks at a time. Furthermore, the worst-case assumption was that only very limited internet connectivity would be available – enough bandwidth for basic monitoring and control, but far too little for transmission of bulky image data.

As a consequence of these operational constraints, it was initially foreseen that the Flyeye DPC would host a short-term archive containing up to 90 days of image data. Before the local disk space was exhausted, an engineer would travel to the observatory and move dozens of terabytes of data to a portable storage. This would then be physically taken to an ESA site and moved into the long-term archive.

Luckily, in the case of Flyeye-1, the Mt Mufara observatory site will not have the aforementioned constraints and will be equipped with a comms line capable of transferring image data to ESA's data centre daily. The rather challenging "worst-case" operational concept can therefore be adapted to a mode of operations where the Flyeye DPC still maintains a short-term archive but transmits its data to the long-term archive at the earliest opportunity. Local storage is still required to cover comms line outages, although it is debatable whether 90 days of storage capability is still required. Reducing this would result in server hardware cost savings.

3.3. Hardware Envelope

On the topic of server hardware, a fixed layout of three physical servers was defined during the procurement of Flyeye DPC. One server is dedicated to an archiving role. Additionally, two servers were foreseen to carry out processing activities. A shared storage system is used by all three. Ultimately, all three servers were uniquely equipped, which may have been sensible in the original operational concept. From a parallel processing and especially high-availability operations perspective however, this resource layout is not optimal.

As a first step, the Flyeye DPC development team have distributed the servers' main memory allocation more evenly, to allow for better resource usage during parallel processing. Of which there is a lot: 16 cameras produce 32 megabytes of raw data every 40 to 60 seconds, resulting in many processing steps, some of which can again be carried out in parallel. A lot of the tasks are high in memory usage and having one machine with far less memory than the others places an unnecessary limit on how much work it could perform. By distributing the memory more evenly across the machines, processing time was reduced overall.

For the final deployment at Mt Mufara, we are planning on procuring new server hardware and will take into consideration the lessons learned so far. We will align the hardware specifications to the real-world needs and size the servers in such a way that clustered high-availability operations become possible.

3.4. Runtime Environment

From the start of development activities, it was clear that a containerised runtime environment should be targeted. Across ESA's Space Safety programme, practically all software systems have been moved from "bare-metal" deployments to containerised runtimes. The advantages are plenty: operating system maintenance and upgrade cycles are effectively detached from application-level release cycles. Strict access segregation can be enforced, such as no root-level access by software engineers or operators. Software is pre-packaged during the build process (more on that later), including sensible default configuration. This is enriched with per-environment configuration during the deployment process, putting the overall system under tight version control, including software, runtime layout, internal comms, configuration. This allows for predictable, repeatable deployments.

There are several competing technologies implementing this paradigm, two of which were analysed in the design phase of the activity: Docker in Swarm mode, and Kubernetes. Kubernetes offered more scalability potential, at the cost of more complexity at infrastructure level. The scalability aspect was analysed in detail but considered not to be mission critical in this application. The workload profile is very well known, and defined by the number of cameras, their image sizes, and numbers of observations per night. Fluctuations are possible, but only towards lower workload, for example in cases of cloudy skies or dome closures due to bad weather. With the upper bound of data volume known, and the processing hardware capabilities known, scalability becomes much less of a design driver.

The simpler Docker setup was therefore chosen. It binds the three participating servers into a Swarm cluster. Via private networking within the cluster, all internal communications are facilitated, with only a few external endpoints

exposed. Shared storage is available to all nodes, enabling shifting of workloads and even allows fault-tolerance in operations in case of single server failure.

3.5. Continuous Integration / Continuous Deployment (CI/CD)

For ESA's Space Safety Programme, fully automated build pipelines are mandatory for practically all software systems. Industry delivers source code to ESA's Gitlab instance, complete with machine-readable build instructions for Gitlab-CI, a built-in component that performs builds and deployments following the CI/CD paradigm (continuous integration / continuous deployment).

While the Continuous Integration (CI) concept is more relevant at the developers' side, it is still followed at ESA, too: every delivery to ESA automatically triggers a build process. Source code is compiled to executable units and packaged into Docker runtime images along with configuration or other required data. The resulting Docker images are stored in an artifact repository, from which they can be deployed to a runtime system.

The Flyeye DPC's development team were already practicing this workflow. With a development environment tied closely to the Microsoft ecosystem, they primarily use Azure's CI/CD offerings. This was not desired for the Flyeye DPC, so that the industry team reworked its build and deployment instructions to match ESA's standards. Here, the aforementioned adoption of Dotnet Core paid off, allowing build and deployment natively on ESA's operational infrastructure.

Now, as the teams are approaching Site Acceptance Test (SAT) at the validation site in Matera, Italy, three distinct deployment environments are being defined: (1) the operational ("production") environment at the telescope site, (2) a testing deployment at ESA premises for acceptance testing before triggering operational deployments, and (3) a development and testing deployment at industry side, where the industry team prepares and tests new versions of the software before delivering to ESA. By SAT, those three systems will be fully up and running, to support smooth maintenance and evolution beyond acceptance testing.

3.6. Integration with Observatory Systems

At the observatory, one central piece of equipment is in charge of monitoring and control of all observatory subsystems: the Site Equipment Controller (SEC). This includes monitoring the weather, monitoring and controlling physical elements like the dome, telescope mount, cameras, and software elements such as the Flyeye DPC.

The baseline assumption at the start of the activity was a Simple Network Management Protocol (SNMP) implementation at the Flyeye DPC. Meanwhile however, the SEC adopted more tailored ways of monitoring subsystems, including a new HTTP-based Basic Monitoring and Control Protocol.

This protocol is implemented as a server-side component on both SEC and Flyeye DPC. In this configuration, it allows the SEC to query the Flyeye DPC for status and health information periodically. Additionally, it allows the Flyeye DPC to actively inform the SEC of changes, for instance the reception of new imagery from cameras. SEC then retrieves scaled-down versions of camera images from DPC's AstroDrive system and displays it to an operator. Furthermore, rolling image counters are transmitted to SEC, allowing for monitoring of processing progress, and, together with error reporting, results in a tight monitoring and control integration with the Flyeye system's observatory controller.

This integration has successfully been tested with the SEC's testing harness and will be confirmed during Site Acceptance Test with the real systems in the loop.

4. Preliminary Results

As of the time of writing this paper, Factory Acceptance Test at industry has been completed successfully. The full battery of tests takes a whole week to execute and contain an end-to-end test exercising all routine facilities of the system. This end-to-end test utilises a synthetic data set, containing stars, asteroids, and all manner of distortions. This data set is sufficient to determine the overall processing performance and robustness of the system.

In the last twelve months of development, the system underwent thorough performance testing and improvements. Figure 2 shows a plot of degraded processing performance, which prevented acceptance testing from passing. Astrometry processing delay is plotted against observation time. Orange and red bars denote the acceptable bounds of delays. If processing occurs too late, i.e., exceeding the red threshold, the associated astrometry is disregarded, resulting in a loss of observation of the object in question. If an object is not observed at least three time during a night, no useful orbital information can be derived. Timely processing is therefore of the essence.

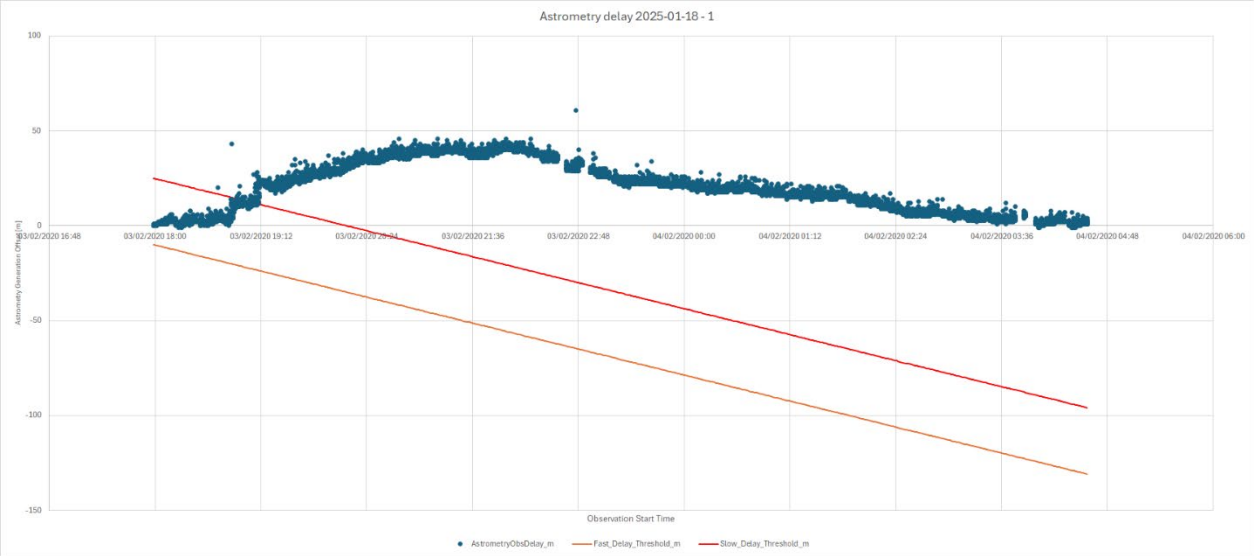

Figure 2 Processing performance - degraded

After careful analysis, several bottlenecks for performance were identified and removed. Some were in software; others were actual hardware problems which had to be tackled with the storage manufacturer. Lately, processing run plots look like Figure 3.

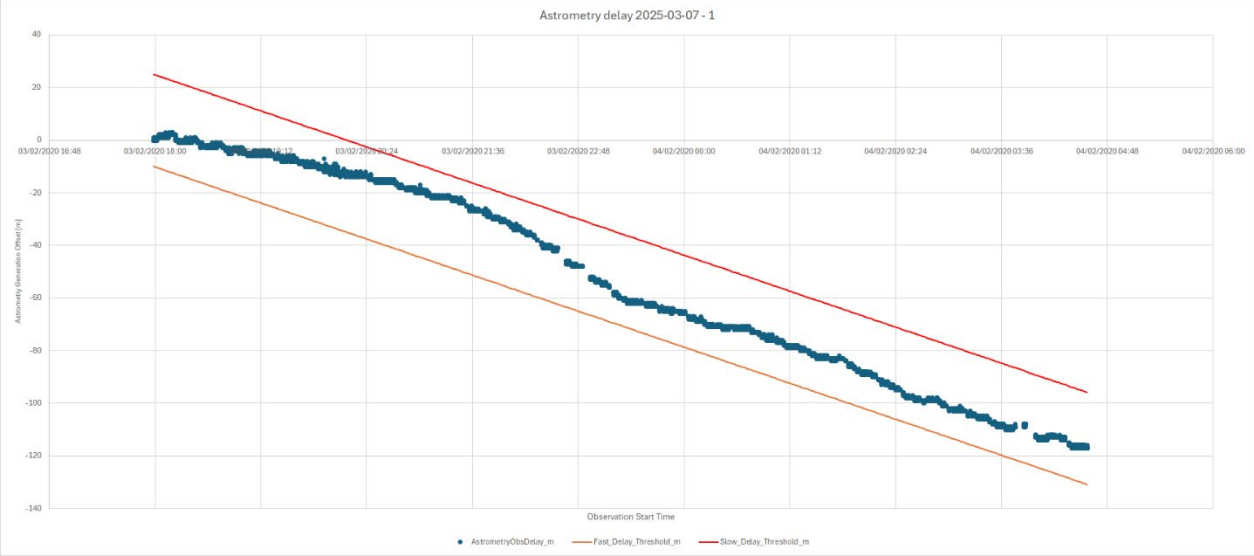

Figure 3 Processing performance - nominal

Ultimately, the system is now consistently performing within the time envelope, with no frames or astrometry dropped even during the taxing two-night processing test. This resulted in successful completion of Factory Acceptance Test at industry.

5. Next Steps and Outlook

Flyeye-1 telescope is currently being assembled at an intermediate observatory site in Matera, Italy. This observatory has all of the operational subsystems intended for the final Flyeye-1 location at Mt Mufara. As part of the overall Flyeye-1 acceptance campaign, a dedicated Flyeye DPC Site Acceptance Test campaign is being prepared. This will prove full integration with its surrounding systems: tasking software, cameras, SEC, and observatory infrastructure such as networking. Additionally, SAT and the ensuing operations campaign will yield valuable performance metrics which will feed into the hardware procurement of the Mt Mufara processing hardware.

The dataset used in FAT was sufficient to prove basic operations and timeliness but was found to be not too realistic with regards to the true Flyeye-1 images. In parallel to the integration activities, a dedicated activity is underway to fine-tune detection performance against another, more realistic data set provided by ESA.

Finally, the next telescope, Flyeye-2, is already under preparation. It will feature a different optical system, and likely a smaller number of cameras. Individual camera data rate is likely higher, so total performance requirements could surpass those for the Flyeye-1 DPC. During development of the DPC, great care was taken to retain flexibility in processing chain design, down to the number of cameras. Adaptation to a number of cameras other than 16 should be almost trivial, and in fact the DPC already today supports operations with individual cameras being offline and not delivering images. The bigger challenge will be fine-tuning of detection parameters, much like the process currently ongoing for Flyeye-1. The lessons learned during Flyeye-1 preparation will therefore also pay off for Flyeye-2 and beyond.

6. References

[1] https://www.esa.int/Space_Safety
[2] http://iguassu.cz/