STR4SD - EXPLORING THE CONCEPT OF OPPORTUNISTICALLY USING STAR-TRACKERS FOR SPACE DEBRIS OBSERVATIONS

J. Feiteirinha(1)**, V. Kairiss**(2)**, V. Reggestad**(3)**, T. Flohrer**(3)**, L. Maleville**(4)**, J. Siminski**(5)**, and E. Maestroni**(6)

(1)*SERCO GmbH, 64293 Darmstadt, Germany, Email: Jose.Luis.Feiteirinha@esa.int*
(2)*RHEA GmbH, 64293 Darmstadt, Germany, Email: Vadims.Kairiss@esa.int*
(3)*ESA, 64293 Darmstadt, Germany, Email: {Vemund.Reggestad, Tim.Flohrer}@esa.int*
(4)*LSE Space GmbH, 64293 Darmstadt, Germany, Email: Laurent.Maleville@esa.int*
(5)*IMS Space Consultancy GmbH, 64293 Darmstadt, Germany, Email: Jan.Siminski@esa.int*
(6)*ESA, 64293 Darmstadt, Germany, Email: Elia.Maestroni@esa.int*

ABSTRACT

The objective of this study was to exploit currently unused downlink capacities in Earth Observation missions to perform opportunistic star tracker imagery and validate it for space debris observations. Approximately two thousand star-tracker observations have been collected and processed, several of these images contain unidentified moving bodies. Done in close cooperation between the Earth Observation Ground Data Systems section, Space Debris office and SWARM Flight Control team, this reconnaissance study provides a preliminary proof-of-concept end-to-end ground segment prototype containing: observation intentions, triage, uplink, downlink and data processing elements. The study provides also future considerations regarding the flight operation concept with multiple Earth Observation missions being analyzed and compared regarding their applicability for these objectives. There were three main deliverables produced: Mission Planning concept with a prototype; Data Processing concept with a prototype; and the final report. The study has also identified conceptual limitations and provides recommendations for future work.

Keywords: Star Tracker, STR4SD, MPS, Mission Planning, SWARM, Space Debris.

1. INTRODUCTION

It has been proven that images taken by star trackers can detect the presence (and some orbit details) of resident space objects [1]. Furthermore ESA Earth Observation satellites contain star trackers and there are occasional unused downlink capacities. ESA infrastructure is currently migrating to a more flexible mission planning infrastructure[2] which shall enable higher complexity planning rules and modelling.

The proof-of-concept idea is to offer Space Debris office either planned or opportunistic observations made with star tracker cameras, when resources like downlink, mass memory and star tracker cameras themselves are available.

This solution involves using the new Mission Planning System (EGOS MPS[1]). A few XSLT[2] templates were developed in order to ingest all the products and files from Space Debris and Flight Dynamics into the MPS. Output templates to produce products like command stacks are already available. Rules, which are meant to process inputs into outputs are still to be developed.

Although the SWARM mission was selected for the probatory implementation, a mission survey was done during the study to evaluate how star trackers on each of the satellites work and whether it is possible to easily acquire images, information about star tracker cameras hardware, available memory, etc.

Future work would be to write the rules for the Mission Planning System, which will provide the logic behind scheduling and election of observations requests and provide some analytic information like mass memory usage.

2. USAGE CONCEPT

The Space Debris office produces a Star tracker image request file proposing a wish-list of multiple image acquisition opportunities. The requests are classified as: opportunistic, when no exact timing necessary; or planned (desired) when exact timing is required.

This file, together with other relevant flight dynamics products and inputs, is ingested by the EGOS MPS, which processes them to determine possible acquisition opportunities calculated, so that they don't impact mission operations negatively (i.e. use bandwidth only when

[1] EGOS MPS - EGOS Mission Planning System (where `EGOS` is ESA Ground Operations System)

[2] Extensible Stylesheet Language Transformations

it's not used by the mission) and generate respective command sequences. EGOS MPS then produces output files (e.g. command stack files), which will contain all commands to be sent to the spacecraft, including those requested by the Space Debris team.

The prototype planning rules were developed as part of this activity and would need to be improved and complemented in the future. These rules will take into account all of the accrued information in order to produce the command sequences to be sent to the spacecraft. Rules will perform multiple checks like star tracker availability at certain times of requests, commanding availability and mass memory usage. As a result of these checks, the rules will either generate or skip generating respective command sequences for star trackers. In case of opportunistic requests, time slots will be used. In order for an opportunistic request to be created, all of the aforementioned conditions must be met star tracker is available, mass-memory is available, no other important activities are happening at the same time.

Further development of the planning rules will be the major part of the work done for the MPS because all the triage logic is going to be defined on top of these rules. This will be done in the future, after SWARM's migration to EGOS MPS is mature and used operationally. Mass Memory resource modelling might get eventually implemented by the Flight Control Team (FCT) in the MPS, which would greatly help fine-tuning the rules.

After running these rules, Mission Planning then produces two products:

- an effective requests XML file, presenting the times and properties of the requests to be sent to the spacecraft,
- a command stack with time-tagged commands, which will be the actual commands sent to the spacecraft

Flight Operations Segment (FOS) will then send the commands and collect telemetry in the Mission Control System (MCS). Automatic, nightly requests collect the necessary telemetry to generate images. These requests overlap slightly so that images that spillover mid-night can also be properly collected. The requests shall be always done for a window of about 24 hours, starting 48 hours in the past as to ensure that all data has been replayed when in nominal operations.

Space Debris will then be responsible for periodic collection of data.

3. PROOF-OF-CONCEPT IMPLEMENTATION (SWARM)

Figure-1 is a schematic representation of which inputs are to be used by the Mission Planning System and what results will be produced.

The following existing input products will be used:

- `SUMO file`[3] is a list of spacecraft star trackers and their respective stance to Earth, Moon and Sun including information about whether they are blinded by any of these three celestial bodies. After receiving feedback from Space Debris this file is not required in the MPS for any checks, but still might be useful in the future for additional validation,
- `ORBEVT`[4] a file which lists all orbits of a spacecraft and is used to calculate times for the orbital angle tagged commands and sequences,
- `SPF`[5] - lists all the possible ground station passes, includes time-tagged `AOS`[6], `LOS`[7], TC and TM masks, eclipses and other events.

Some of the following input templates for the EGOS MPS were developed fully as a part of this activity (SUMO and SD file templates), others were modified to accommodate to needs:

- `xPFtoEVF.xsl` is a template to convert SPF and any other Planning Files, which share the same format into the Event facts of the EGOS MPS. This will be used for the SPF conversion and ingestion,
- `SUMOtoEVF.xsl` is a template developed specifically for this activity. It allows converting SUMO files into a set of EGOS MPS facts of the Event type. These facts contain information about each of the spacecraft's star trackers orientation and blinding from the Sun, Moon or Earth at any certain point in time.
- `SDtoEVF.xsl` is a template to parse the Space Debris proposed input file. It parses all the relevant request information, which would be used further on in the rules.
- Other templates like `xPFtoCRF.xsl`, which convert any planning files (OPF, PPF, SPF, DPPF,) into command sequences already exist as part of the configuration of the missions and shall be reused.

Output template for the Command Stack File already exists within the current configuration, but will most probably also be modified and updated as a result of the migration to EGOS MPS. An extra template to produce a report XML file with all the requests to star trackers will also need to be developed.

[3]Sun/Moon to star tracker boresight report
[4]ORBEVT - Orbital Event File
[5]SPF Skeleton Planning file
[6]AOS - Acquisition of Signal
[7]LOS - Loss of Signal

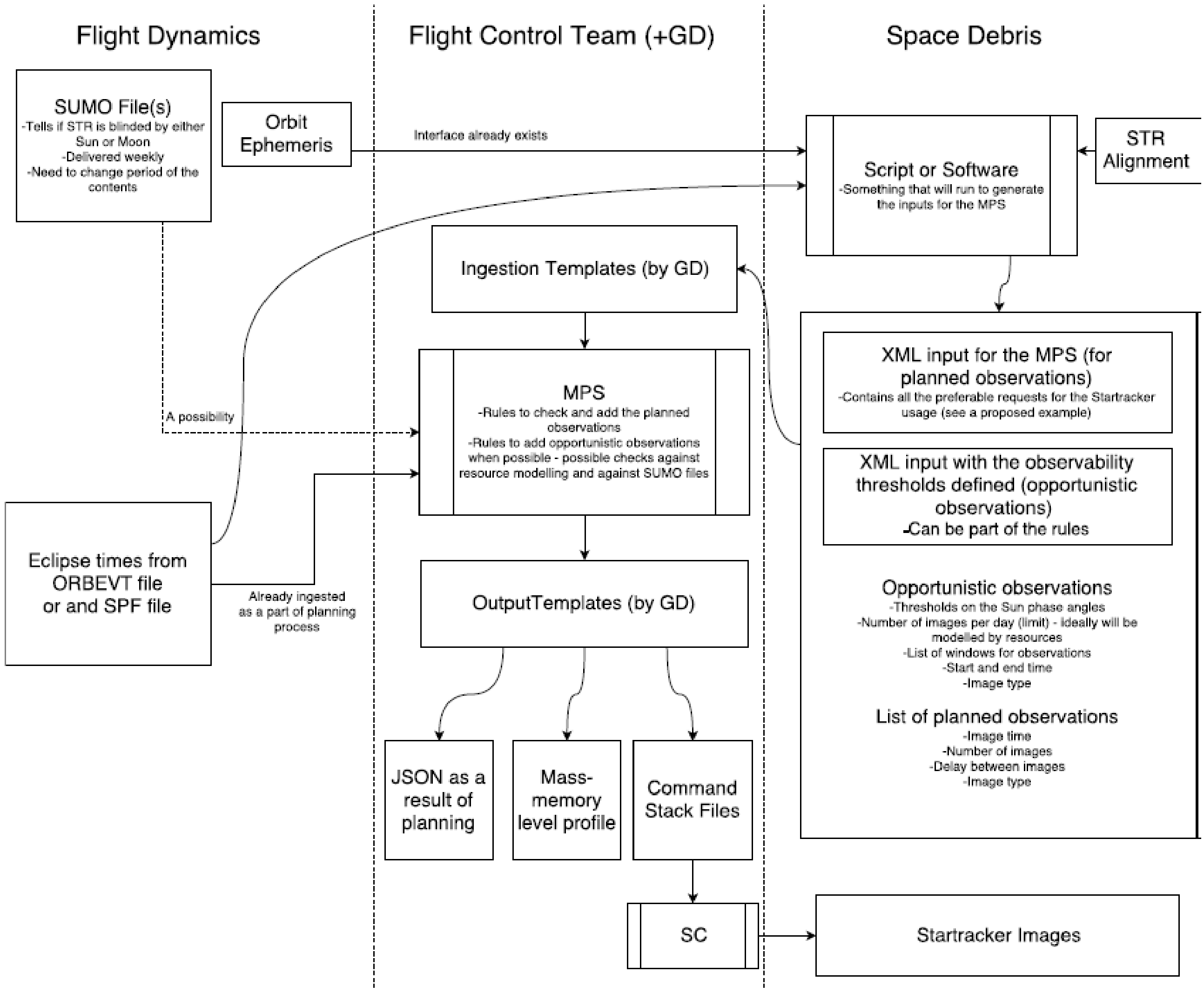

Figure 1. Diagram showing MPS inputs and outputs expected.

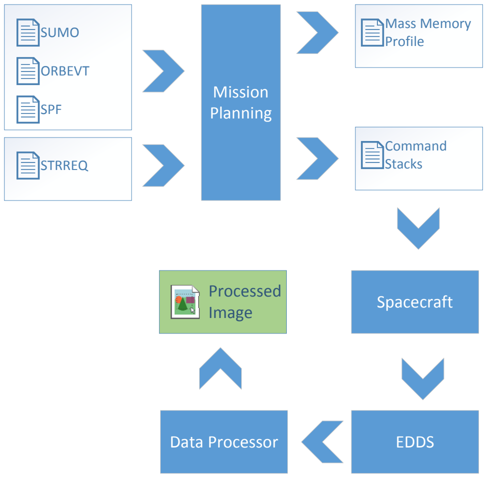

Figure 2. An overview of the implemented prototype.

3.1. Star tracker acquisition details

The star trackers deployed on SWARM have two fields per image. Each integrates sequentially for 0.500 seconds leading to a total integration period of 1.000 seconds for the entire image. The "center-of-integration" of each of the two image fields are therefore separated 0.500 seconds. This enables time-resolution within a single acquisition and further opportunities such as the generated animated GIFs, as well as differential images like the one highlighted in Figure-6.

4. PROOF-OF-CONCEPT DEPLOYMENT

The Mission Planning input and output files are described in detail in section 3, the application and data flow are abbreviated in Figure-2. The Spacecraft to the EDDS Data Dissemination System connection omits `SCOS`[8]/`PARC`[9] which are responsible for acquiring and distributing data to EDDS respectively. A working prototype of the Data Processor tool was implemented in Node.js for SWARM in the scope of this activity. The current implementation produces PNG files with the images and companion JSON files containing the metadata. In the future, the FITS format should enable a single container for image and meta-data information. In order to facilitate the viewing of the images a prototype data browser was implemented, it can be consulted here: `https://esa.github.io/str4sd/`.

[8]SCOS - SpaceCraft Operations Control System
[9]PARC - Packet ARChive

5. LIMITATIONS AND FUTURE WORK

5.1. Support FITS file format

Currently only PNG and GIF file formats are supported. All meta-data is delivered in the format of JSON files. It is understood that most benefit will be taken if the meta-data is embedded directly onto the image format and delivered using the FITS file format.

This can potentially be done by parsing the JSON files for the meta-data and extracting the matrix point information from the lossless PNG images. Another alternative is to generate the FITS files during processing, like it is done for PNG. At the moment of writing this, we are not aware of any Node.js library capable of supporting FITS file format generation.

5.2. Timestamping accuracy

During this study it was understood that the time accuracy of the acquired images will be mission dependent. For SWARM, the star tracker clock is independent from the on-board computer clock and resets with each reboot of the star tracker. The mission response will be evaluated, currently it is only known that there is an on-board software feature to trigger the generation of TM [9,131] datation (i.e. timestamps) packets from all instruments. But before introducing any change impacting all payloads (and not only the startracker) there will first be required an evaluation of the existing images to proof that they are really suitable for study. Furthermore, such changes may need to be agreed with other parties in the mission, as it is good and common practice to involve the relevant parties before changing something of such high impact.

As best effort, the timestamp of the images was taken from the on-board time of the telemetry packets containing the image. This is by procedure only - about 10 seconds after that the image acquisition.

5.3. Parameter Correlation

Other periodic telemetry parameters may be of interest and should be added as meta-data to the FITS image files. This can be, for example, attitude related parameters. The periodicity of the parameters, and future requirements should decide if interpolation or before/after value pairs (with timestamping) should be provided as part of the meta-data. For this study there was no effort done in merging such information.

5.4. Developing Mission Planning System rules

Some rules were developed as a result of this activity:

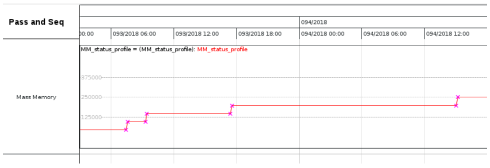

Figure 3. An example of how the modelled Mass Memory would look like on a Jaret chart

- `Generate_STR4SD_Requests` rule is generating the actual events (later command sequences) for taking star tracker pictures out of the ingested events representing planned and opportunistic observation requests from the Space Debris Office.
- `Generate_STR4SD_Profiles` rule is generating the profiles for the Mass Memory resource modelling, which is going to keep track of how full Mass Memory is and later to be used in different rules for validation of the requests. For instance, check that there is still space in Mass Memory for an extra picture request to decide whether to generate or not to generate a command sequence.

Further work will have to be done when implementing the remaining EGOS MPS rules responsible for selecting the available windows for taking the star tracker images. There is room for further improvements and writing new rules. This will need to be done after SWARM has a working operational set of configuration available, because now this configuration is still under development and lacks the required Mass Memory modelling.

For the first prototype, a starting point could be using very limited rules (see Figure 3), which would simply create respective command sequences at any point, where there is a planned request. Later these would be improved with every iteration/delivery.

5.5. Automation

For the proof-of-concept no automation was put in place. It should however be possible to do so re-using existing ESA software once this concept becomes operational. ESA's Generic File Transfer System (GFTS) can be used to automate the file transfer and execution of scripts (e.g. to call the data processor). EDDS requests can be scheduled to run every night to collect the acquired images.

5.6. Data Visualisation

The website currently is semi-dynamic (dynamically generated from static XML and JSON files created at processing time) and will most certainly not scale. If this tool is considered valuable, then some effort should be spent in improving scalability (e.g. adding a back-end with a database).

6. COLLECTED DATA

A total of 2504 images were collected since the beginning of the mission until the end of 2018. These images have been acquired since the beginning of the mission as part of routine operations and were kept in the mission archive as RAW data.

6.1. Sample Raw Image

Figure-4, shows a processed RAW matrix data. This includes both integration fields interleaved.

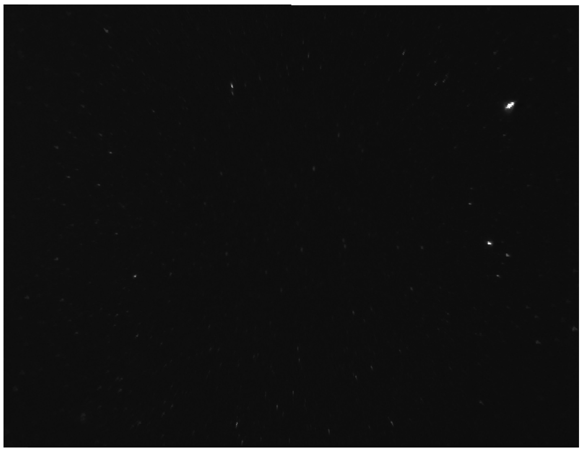

Figure 4. Raw image taken from SWARM-B's star tracker B on 2015-01-13 around 14:30 UTC

6.2. Sample Processed image

In order to facilitate triage of the most relevant acquisitions, a crude differential processing algorithm was implemented. See Figure-5 and Figure-6.

This algorithm de-interlaces both fields and compares them. The difference between the first and the second field are then highlighted using the formulas:

`Red channel = (ΔP << 2 ) & 0xFF`, if difference is positive

`Green channel = (-ΔP << 2 ) & 0xFF`, if difference is negative

`Blue channel =` *Original pixel intensity (gray scale)*

Where `ΔP` is the pixel's difference for that particular channel.

Red/Green channels will default to the original pixel intensity (gray scale) if the conditions are not met.

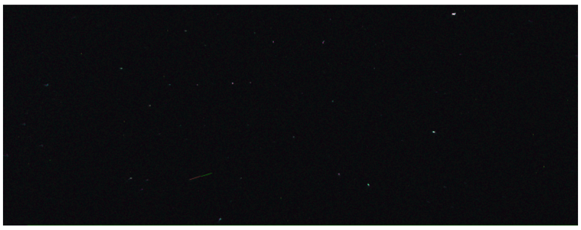

Figure 5. Generated by processing image taken from SWARM-B's star tracker B on 2015-01-13 around 14:30 UTC

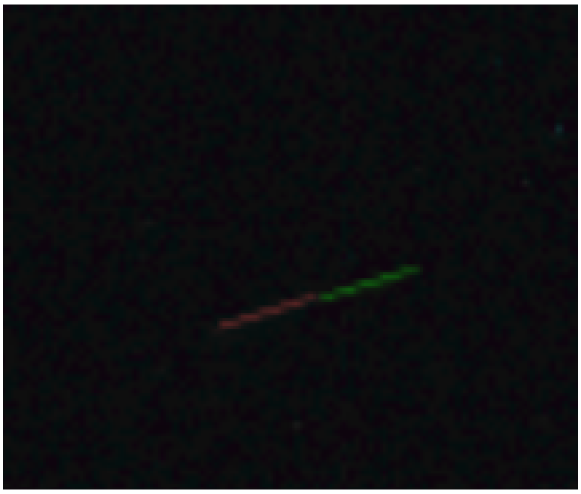

Figure 6. Zoom-in on an unidentified body from the image above

7. PRODUCT NAMING CONVENTIONS

Existing and planned product file naming conventions can be found in Table 1. Acronym codes used in the table can be understood as follows:

- `SAT` Satellite (SWA, SWB,)
- `DDD` Day of Year
- `VV` - version
- `YYYY_DDD_HH_MM_SS_MSEC` - timestamp of an image

8. MISSION SURVEY

To acquire an overview of the resources available in ESA, a small mission survey was performed. Four missions responded: Cryosat-2, SWARM, Sentinel-2 (A and B) and Sentinel-5p.

Figure-7 summarizes the results of the survey in table containing the processed responses.

9. EXAMPLE FILES

Two new file formats were created during the proof-of-concept. One, was the request file to be processed by the mission planning service. An example of this file can be found in sub-section 9.1. This file should contain all information needed for the final implementation. The second was in JSON format and contains the meta-data of the acquired image. This file may change depending on the characteristics of the camera and interface of the star tracker. It may eventually be superseded by embedded data in a FITS formatted image. An example of this file can be found in subsection 9.2.

9.1. Star Tracker Request File (XML)

Below is an example star tracker observation requests in the proposed XML format. This file would be generated by the Space Debris team and passed to the Mission Planning System.


```xml
<startracker_request_file>
  <file_header>
    <source>Space Debris</source>
    <purpose>Operational</purpose>
    <validity_start>2018-04-02T00:00:00.000Z</validity_start>
    <validity_end>2018-04-08T23:59:59.000Z</validity_end>
    <mission_name>SWARM</mission_name>
    <spacecraft_name>SWA3</spacecraft_name>
    <generation_time>2018-04-01T06:56:16.000Z</generation_time>
    <version>1.0</version>
  </file_header>
  <startracker_request_list>
    <str_request type="planned" satellite_id="SWA3"
startracker_id="2">
      <start_time>2018-04-03T07:22:12.700Z</start_time>
      <end_time>2018-04-03T08:22:12.700Z</end_time>
      <plan_param_list>
        <plan_param_item>
          <parameter_name>images</parameter_name>
          <parameter_value>5</parameter_value>
        </plan_param_item>
        <plan_param_item>
          <parameter_name>delay</parameter_name>
          <parameter_value>20</parameter_value>
        </plan_param_item>
        <plan_param_item>
          <parameter_name>image_type</parameter_name>
          <parameter_value>bitmap</parameter_value>
        </plan_param_item>
      </plan_param_list>
    </str_request>
    <str_request type="opportunistic" satellite_id="SWA3"
startracker_id="2">
      <start_time>2018-04-02T00:00:00.000Z</start_time>
      <end_time>2018-04-08T23:59:59.000Z</end_time>
      <plan_param_list>
        <plan_param_item>
          <parameter_name>sun_phase_low</parameter_name>
          <parameter_value>45</parameter_value>
        </plan_param_item>
        <plan_param_item>
          <parameter_name>sun_phase_high</parameter_name>
          <parameter_value>90</parameter_value>
        </plan_param_item>
        <plan_param_item>
          <parameter_name>images_per_day</parameter_name>
          <parameter_value>40</parameter_value>
        </plan_param_item>
        <plan_param_item>
          <parameter_name>image_type</parameter_name>
          <parameter_value>bitmap</parameter_value>
        </plan_param_item>
      </plan_param_list>
```

	Question\Mission	Cryosat-2	SWARM	Sentinel-2A/2B	Sentinel-5p
Operations	**Frequency of acquisition of startracker images**	Other	Monthly	Never	Month / On request
	Procedure to acquire startracket images	Available	Available	Available	Available
	Process complecity (0 low, 5 high)	4	1	3	3
	Startracker provider	TERMA	DTU	Jena Optronik	SODERN
IMAGE FORMATS	**Raw TM Packet**	X	X	X	X
	Region Of Interest (Partial image)		X		
	JPEG		X		
	PNG	X			
	FITS				
	Centroids (list of the captured centroided objects)		X		
	Non-Stellar Object List (list of unrecognized centroids)		X		
COLOR DEPTH	**8 bits, gray scale**	X	X		X
	12 bits gray scale			X (TBC)	
	16 bits gray scale				
	24 bits gray scale				
	24 bits RGB				
SENSOR	**Field of View**	22 deg x 22 deg	13.7 x 18.3 deg	20 deg circular	21.5 deg
	Image Size (Raw)	~ 1 MB	~ 0.5 MB	~ 1.5 MB	~ 0.5 MB
	Sensor Resolution	1024 x 1024	580 x 752	1020 x 1020	600 x 699
	Pixel Size (x / y directions)	13 µm / 13 µm	8.6µm / 8.3µm	15µm x 15µm	18µm x 18µm
	Readout noise	$40e^-$ @ 4Hz	$11e^-$ @ 4Hz	35e-10 @ ???	*n/a*
	Integration Times	300 ms (configurable)	500 ms	1 ms	1 - 8000ms
	Timestamp information	Available	Uncorrelated	Available	Available (1/8s acc)
	Minimum time between two acquisitions	22.5 min	~ 8 min (RAW)	Unknown	Low / TBD
	Comments		Images are interlaced		
Attitude & Position	**Roll angle error**	< 0.1 deg	< 5.0 deg	< 0.0012 deg	< 0.0001 deg
	Pitch angle error	< 0.12 deg	< 5.0 deg	< 0.0012 deg	< 0.0001 deg
	Yaw angle error	< 0.15 deg	< 5.0 deg	< 0.0012 deg	< 0.0001 deg
	Position Error	0.4 - 2m (typically)			< 10 m
	Available Memory	Yes	10 - 100 images per day	Yes	Yes
Opportunity	**Count**	Acquisition disables AOCS	Up to 2	1	1
	Schedulability	No	Yes	Yes (TBD)	Yes

Figure 7. Overview of the results of the mission survey.

Filename	Description	Flow
SD_STR_REQ_SAT_YYYY_DDD_VV.xml	Acquisition request	SD $\rightarrow$ MPS
STR_REQ_OPS_SAT_YYYY_DDD_VV.STACK	Generated command stack	MPS $\rightarrow$ MCS
EDDS_SAT_STR_YYYY_DDD_VV.dat	RAW Packet Data	EDDS $\rightarrow$ *tbd*
SAT_STR_YYYY_DDD_VV.JSON	Camera specific metadata	*tbd* $\rightarrow$ SD
SAT_STRID_YYYY_DDD_HH_MM_SS_MSEC.PNG	Image in PNG format	*tbd* $\rightarrow$ SD
SAT_STRID_YYYY_DDD_HH_MM_SS_MSEC.FITS (tbc)	Image in FITS format	*tbd* $\rightarrow$ SD
STR_EFREQ_SAT_YYYY_DDD_VV.xml	Effective requests *(optional/tbd)*	MPS $\rightarrow$ SD

Table 1. A table declaring the products.

```
    </str_request>
  </startracker_request_list>
</startracker_request_file>
```

There are offered two types of requests. One with planned star tracker images and another for opportunistic takes. It will probably be good practice to include the whole planning period for the opportunistic pictures.

There are the following attributes for each of the requests:

- `type` defines whether its going to be a planned or an opportunistic request
- `satellite_id` defines the satellite identification code
- `startracker_id` defines the target star tracker for the request

And the following are possible parameters:

- `start_time` request start time
- `end_time` request end time
- `image_type` type of an image to be requested (if there is a choice)
- `images` (for planned) how many images to take
- `delay` (for planned) delay between images taken
- `sun_phase_low` (for opportunistic) lowest sun phase in relation to a star tracker. Take images only between low and high.
- `sun_phase_high` (for opportunistic) highest sun phase in relation to a star tracker. Take images only between low and high.
- `images_per_day` (for opportunistic) max number of images to take per day.

9.2. Image Meta-Data File (JSON)

Following is an example of an image metadata in JSON format below. The fields naming follows the convention that any field directly derived from the Star tracker Interface Control Document (ICD) shall be in full capitals. Other fields shall be in lower-case.

```
{
  "packet_count": 440,
  "Image Header Hex": "99089 ... 01002201f0020100",
  "DAC_OFFSET": 2201,
  "DAC_GAIN": 3481,
  "INTEGRATION_TIME": "1s",
  "COMPRESSION": 0,
  "ROI": 8,
  "JPEG_QUALITY": 75,
  "THRESHOLD": 16,
  "INFO": 10,
  "INFO:CAMERA_ID": "StrC",
  "VALID": "5055",
  "STATUS": 0,
  "CODE_START": 34,
  "CODE_END": 436194,
  "SUB_TIMESTAMP": 0.00001740553147790368,
  "TIMESTAMP": 79288,
  "H": 290,
  "W": 752,
  "IMOD": 1,
  "uid": 0,
  "Data Length (bytes)": 436162,
  "first pus packet header": "0D09C1B803F9",
  "filepath": "./imgs/SWA__0_StrC_0006217967_Rujhge.png"
}
```

Description of the available fields:

- `packet_count` Total number of packets[10]
- `Image Header Hex` Hexadecimal display of the RAW image header
- `DAC_OFFSET` The AGC floor value used for acquiring the image
- `DAC_GAIN` The AGC ceiling value used for acquiring the image
- `INTEGRATION_TIME` The integration time of the image

[10] All SWARM packets are in PUS (Packet Utilization Standard) format.

- `COMPRESSION` Compression code:
  - 0 - Uncompressed,
  - 1 - Centroids,
  - 2 - Region of Interest (ROI),
  - 3 - JPEG,
  - 4 - Non-Stellar Object List
- `ROI` The region of interest used for the ROI compression
- `JPEG_QUALITY` The quality of the JPEG compression.
- `THRESHOLD` The threshold used for the ROI compression
- `INFO` Various information about the image
- `INFO:CAMERA_ID` Sub-field of INFO with the camera identifier.
- `VALID` Image validity
- `STATUS` Status of the compression
- `CODE_START` Offset of start of image data within the total image
- `CODE_END` Offset of end of image data within the total image
- `SUB_TIMESTAMP` subsecond part of the timestamp with accuracy of $\frac{1}{2^{16}}$ seconds
- `TIMESTAMP` timestamp in seconds
- `H` height of the image
- `W` width of the image
- `IMOD` Interlace modifier
- `uid` image identifier within the processing batch
- `Data Length (bytes)` total data length of the image in bytes
- `first pus packet header` hexadecimal display of the RAW PUS header of the first packet of the image
- `filepath` path to the processed file

10. FUTURE WORK

We've proved that SWARM star trackers can observe resident space objects, and conceived a concept to automatically request, download and process the optimal amount of acquisitions from a running mission, without impacting nominal operations. The acquisitions collected for this project have been made publicly available by ESA with the hope it can help future research. Future work will include moving from the concept to the design and implementation phases, and may involve considering other missions and star trackers.

ACKNOWLEDGMENTS

The authors would like to thank: John L. Joergensen and Jose M. G. Merayo from the Technical University of Denmark for the advice during the implementation of the image processing software, to Rune Floberghagen, the SWARM Mission Manager for his support towards public distribution of acquired images and the SWARM Flight Control Team for their continuous involvement and interest.

REFERENCES

1. Karl Bengtsson Bernander,(2014), *A Method for Detecting Resident Space Objects and Orbit Determination Based on Star Trackers and Image Analysis*, Uppsala University
2. Dietz A., Werner D., Montagnon E., et al., (2016). *New MPS*, Rosetta / BepiColombo Mission Planning System: from mission to infrastructure, Spaceops
3. Node.js is a JavaScript runtime built on Chrome's V8 JavaScript engine, `https://nodejs.org/`, 14 January 2019.