Smart On-board Processing for Next Generation SAR Payloads

Oskar Flordal[c], Nicola Gollin[b], Marc Jäger[b], Vangelis Kollias[d], Michele Martone[b], Jamin Naghmouchi[e], Mathias Persson[c], Søren Pedersen[c], Nikos Pogkas[d], Rolf Scheiber[b], Daniel Smith[a], Maike Taddiken[a], Ole Bischoff[a]
[a] DSI Aerospace GmbH, Otto-Lilienthal-Str. 1, D-28199 Bremen, Germany
[b] German Aerospace Center (DLR), Microwaves and Radar Institute, D-82234 Oberpfaffenhofen, Germany
[c] Unibap AB, Västra Agatan 16 5FL, SE-75309 Uppsala, Sweden
[d] Teletel S.A, 124 Kifissias Avenue, GR-11526 Athens, Greece
[e] Universität zu Lübeck (UZL), Ratzeburger Allee 160, D-23562 Lübeck, Germany

Abstract

Smart on-board processing for Earth observation systems (SOPHOS) is a 3-year Horizon Europe project. SOPHOS will design and implement enabling technology for high-end data products produced on board spacecraft via the implementation of more power efficient high-performance space processing chains for various Low-Earth Orbit (LEO) missions, with a focus on Synthetic Aperture Radar (SAR), which is one of the most data intensive space applications currently used. This paper describes the adopted technology and the selected SAR use cases.

1 Introduction

The number of satellites being launched into orbit is increasing rapidly every year, and with it the complexity and capabilities of each satellite continues to grow dramatically. Today, near full Earth coverage by optical sensors is achieved daily by civilian spacecraft, and soon civilian SAR (Synthetic Aperture Radar) will achieve a similar daily coverage. The ever-growing amount of spaceborne data will need new solutions to get that data to the ground, because the available downlink is always a significant bottleneck in space system design. More sophisticated on-board data processing and storage will allow future iterations of spacecraft to achieve higher performance in smaller packages. Current solutions present limitations in computational performance, memory capacity and performance, and data reliability in very small form factors. SOPHOS will design and implement enabling technology up to TRL level 6 for high-end data products produced on-board spacecraft via the implementation of more power efficient high-performance space processing chains for various Low-Earth Orbit (LEO) missions, with a focus on SAR, which is one of the most data intensive space applications currently used. The list of objectives includes enhanced data access to on-board mass storage, miniaturization of HW modules within the processing chain, improvement in on-board SAR raw data compression, improvement in/feasibility of on-board SAR image formation, as well as optimization of portable processing SW routines to enable enhanced on-board SAR processing.

In this paper we present the overall SOPHOS concept and the adopted SAR use cases together with details of the individual payload processing and memory sub-systems, all based on COTS (Commercial Off-The-Shelf) components, their interconnections, as well as the employed SAR algorithms and attempted optimizations.

2 Demonstrator Concept

The SOPHOS demonstrator foresees the physical representation of the Payload Processing and Mass Memory Modules whereas the SAR sensor, downlink and the ground segment functionalities are emulated and implemented as part of the Electrical Ground Support Equipment (EGSE). The concept is depicted in **Figure 1**. High rate payload data communication is implemented with 10GB Ethernet whereas the control and command go through SpaceWire protocol.

The SOPHOS demonstrator implements a flexible data path, based primarily around the router function built into the Mass Memory Module. This allows the payload data to be selectively stored in the Mass Memory Module, processed in the Payload Processing Module, and/or transmitted to the Ground Control EGSE, bypassing or modifying the order of the data path based on the needs of the demonstration.

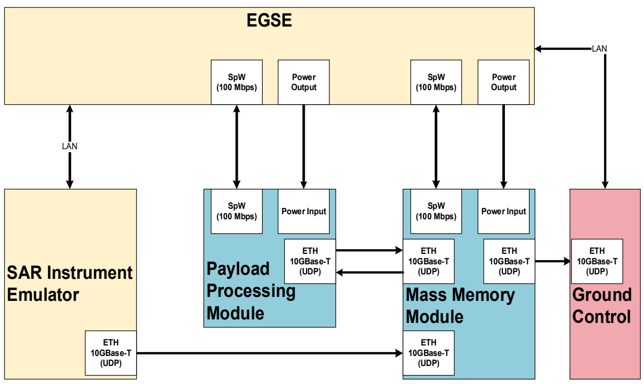

Figure 1 Preliminary design of the SOPHOS demonstrator.

3 SAR Use Cases & Requirements

The on-board processing algorithms to be implemented within the frame of SOPHOS are illustrated in **Figure 2**.

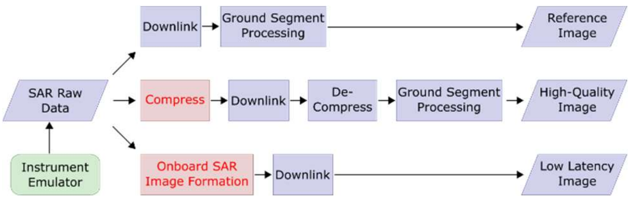

Figure 2 The proposed onboard processing algorithms, highlighted in red, in the context of the use cases considered in the project. The upper line represents the state-of-the-art handling of present-day SAR mission data.

They address two important use cases in a multi-objective, multi-user SAR mission: the derivation of High-Quality as well as Low-Latency SAR information products.

Table 1 summarizes the main parameters of the mission scenarios considered. These parameters were derived taking into consideration the parameters and technical characteristics of commercial reference missions such as TerraSAR-X, ICEYE, Capella Space, Umbra and StriX.

SAR System Parameters	SOPHOS High End	SOPHOS Light
Center frequency	9.65 GHz	=
Bandwidths	600 MHz	150 MHz
Range sampl. freq.	660 MHz	165 MHz
Antenna length	3.2 m	=
Antenna width	0.4 m	=
Off-Nadir	30 deg	=
Pulse length	50 us	25us
Orbit height	560 km	=
Swath width & length	30 km / 50 km	50 km / 50 km
Ground resolution @ 30 deg incidence angle (rg/az)	0.5m / 3 m	2m / 6m

Table 1 SAR stripmap mode parameters for the two mission scenarios considered in SOPHOS.

Based on the reference radar parameters, we derive scene-level requirements regarding data volume and data rate. On this basis, we also look into a typical, traditional mission scenario to derive onboard storage requirements, by considering typical direct data downlink capabilities and sensor operation duty cycles.

In general, and especially for the SOPHOS project, the aim is to ensure that onboard processing does not represent a mission constraint. We make the following assumptions to derive additional requirements for the SOPHOS demonstrator:

- Single-orbit timeline with 2 ground station contacts (e.g. one on the descending and ascending part of the orbit). The effective downlink time is 2 x 10 min, during which no SAR data recording and processing is possible, mainly to conserve instrument resources.
- For the same reason, simultaneous onboard processing and active sensor operation is also excluded.
- Orbit duration is 90 minutes.
- Orbit duty cycle is 1 % to 10 %.

The duty cycle between 1 % and 10 % gives us 1-9 minutes for data acquisition, corresponding to 6-36 reference data takes per orbit (see Table 1). Subtracting the downlink and data acquisition time from the orbit period, we have 61-69 minutes available for onboard processing. This results in the following processing time requirement of one single scene: 100s (61 min / 36 scenes) for the 10% duty cycle scenario and 11 minutes (69 min / 6 scenes) for the 1% duty cycle case.

4 Hardware Components

Figure 1 illustrates the main components of the envisioned SOPHOS demonstrator system. The EGSE / SAR Instrument Emulator will allow to command the payload processing module and mass-memory unit and to monitor all the necessary parameters for their functional testing.

4.1 EGSE

The EGSE shall ensure the telemetry (TM) and telecommands (TC) communication with the payload sub-systems through various links simulating the S/C platform (e.g. Onboard Computer).

The SAR Instrument Emulator will transmit synthetic science data (i.e. SAR raw data) via Ethernet to the Mass Memory Module simulating a SAR Instrument.

The EGSE and SAR Instrument Emulator will be controlled from the Ground Control System via TM/TC communication through LAN connection receiving telecommands to control their operation, and sending telemetry for monitoring their status. The EGSE will also forward S/C telecommands to the Payload Processing and Mass-Memory Modules and receive and forward S/C telemetry from these modules to the ground segment for processing and analysis, simulating the role of the platform.

The Ground Control System will simulate the ground segment transmitting telecommands and receiving telemetry to/from the EGSE to command and control the Payload Processing Module, the Mass Memory Module and the instrument emulator. The Ground Control System will also include a unit receiving the science data from the Mass Memory (simulating a ground telemetry receiver) and will perform decompression, decryption and storage for later offline analysis and visualisation. The Ground Control System will include the appropriate control and data software (Central Checkout System or CCS), that will provide the required control and monitoring during end-to-end tests and measurements.

4.2 Mass Memory Module

The limiting factors of state-of-the art mass memory technologies for use in small satellites are the low storage capacity of standard SDRAMs and the relatively low data rate and complex handling of NAND flash devices. In addition, the scalability is low due to the inflexible internal data routing and legacy EEE component design.

While legacy Mass Memory Modules for space can reach high data rates and capacities, the use of older components and the restrictions inherent in ECSS placement and routing limit the ability for previous generation rad hard or rad-tolerant MMMs to be small enough to feasibly fit in small satellite designs. The SOPHOS system aims to overcome these limitations through the use of new FPGA technology and high-speed interfaces to allow the design to take full advantage of the high capacity and speed of current generation NAND Flash devices. These Flash devices allow the SOPHOS MMM to reach data capacities of more than 32 Tbit in a small form factor unit. Newer SerDes interfaces built into the MMM FPGA allow for data rates of greater than 10 Gbps over a single 2-pin interface lane, allowing for much higher total throughput to support the very high data rate operations required by present and next-generation SAR payloads. The advances in SerDes interface speeds are supported by the use of much faster synchronous NAND Flash interfaces and DDR memory interfaces, which allow for the MMM to support the full data rate of a potential SAR instrument envisioned in SOPHOS, in addition to the routing of data to the independent payload processing module. To enable this, 10G Ethernet as a baseline high speed is used, to allow for data routing between the payload, processing module, and downlink.

The external links are flexible, and the design will allow a wide variety of external interfaces to payloads or other units including Ethernet, SpaceFibre, SerDes, SpaceWire, and others. In the SOPHOS system, a bidirectional connection to the Payload Processing Module is also implemented using high-speed data links to facilitate reading of SAR raw data and writing processed data for download.

4.3 Payload Processing Module

In the SOPHOS project the SpaceCloud solution designed and manufactured by Unibap will be used as a beyond-state-of-the-art solution that carries the needed processing architectures to perform on-orbit processing of high data volumes by carrying a combination of different computing architectures as shown in **Figure 3**.

This heterogeneous compute solution contains an AMD GPU V1000 SoC as the main compute node. This is complimented by different augmenting compute offerings like PolarFire FPGA from Microchip and, Intel Movidius MyriadX to make up the full compute solution [4]. For interfaces 10 GB Ethernet native to the AMD processor is used to source data together with SpaceWire connected via the PolarFire FPGA. Given the system is standard Linux, the system provides a familiar development environment that allows for the use of modern cloud enabled development tools as demonstrated on previous generations of SpaceCloud hardware when used for optical workloads [5].

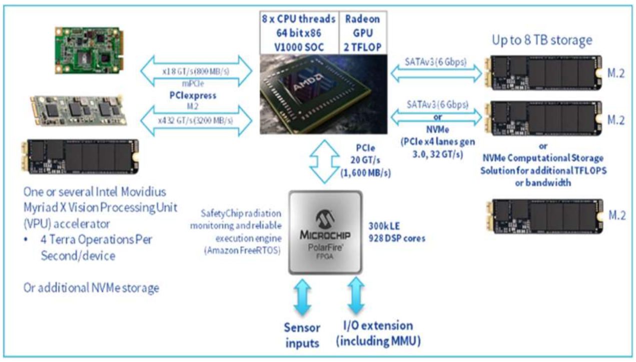

Figure 3 Block diagram with SpaceCloud processing architectures available on iX10.

5 SAR Algorithms & Optimisations

As shown in **Figure 2**, the algorithms selected for implementation on-board cover two central mission scenarios: the derivation of High-Quality as well as Low-Latency SAR information products. High quality SAR images are assumed to be single-look at full resolution to serve advanced applications (e.g. SAR interferometry). They are usually processed on ground w/o any approximations. For this purpose, a high-quality BAQ compression on raw data level is implemented on board. On the other hand, low-latency products might be processed up to a multi-looked image on board making use of the time between data acquisition and available downlink. At the same time, the data size for downlink can be reduced compared to raw data.

5.1 Onboard SAR Raw Data Compression

SOPHOS introduces a novel performance-optimized block-adaptive quantization (PO-BAQ), which extends the concept of the state-of-the-art BAQ and allows for an optimization of the resource allocation by controlling the resulting SAR image degradation [1].

Since quantization errors are significantly influenced by the local distribution of the SAR intensity, such an optimization is achieved by exploiting the a priori knowledge about the SAR backscatter statistics of the imaged scene. The flowchart illustrating the processing for on-board SAR raw data compression is depicted in **Figure 4**: the radar echoes (raw data) are received by the SAR sensor and then quantized using variable bit-rate according to a pre-defined bitrate map (BRM). For this, a state-of-the-art BAQ scheme is considered. After compression the data are as usual stored in the on-board memory and, at some point, downlinked to the ground for further processing and elaboration. The BRM is derived on ground in order to fulfil specific performance requirements (i.e. degradation due to quantization) and it is uploaded to the sensor prior to data-take commanding. An example is shown **in Figure 5**(a): the bitrate map (BRM) on the left-hand side is designed to achieve a constant signal-to-quantization noise ratio

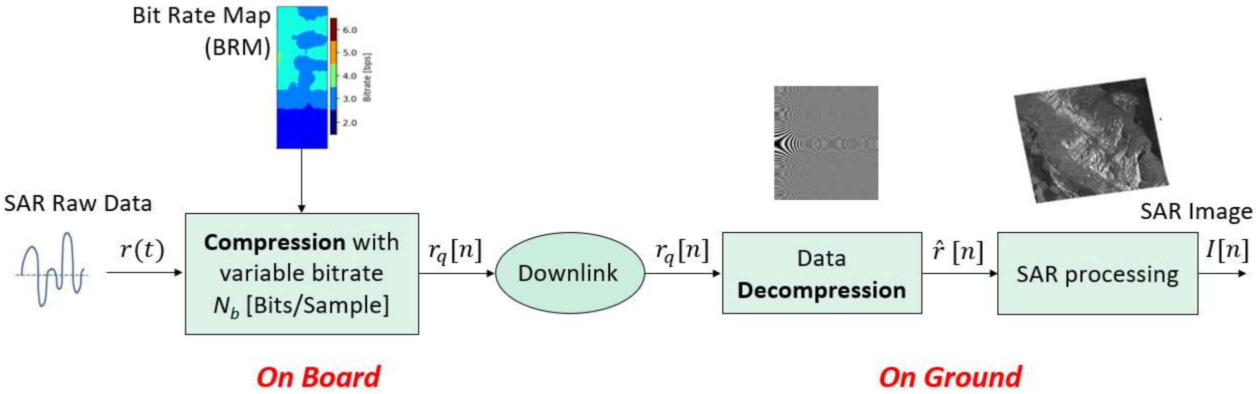

Figure 4 Flowchart illustrating the processing for on-board SAR raw data compression.

(SQNR) of 10 dB, the acquired raw data are quantized according to the given BRM, and afterwards SAR processing is carried out on ground. The resulting SAR image degradation shows a homogeneous distribution (SQNR map on the right and histogram in **Figure 5**(b)), which is effectively concentrated around the target value. If, on the other hand, a constant bitrate is applied, this results in an inhomogeneous image degradation, as depicted in the SQNR histogram in **Figure 5**(c), resulting from a compression using 3-bit BAQ.

Given the severe constraints imposed by the downlink capacity, the approach and analyses adopted in SOPHOS will help to optimize the overall global performance for a given downlink budget and, in this way, to increase the acquisition capability of the system allowing for more continuous observation.

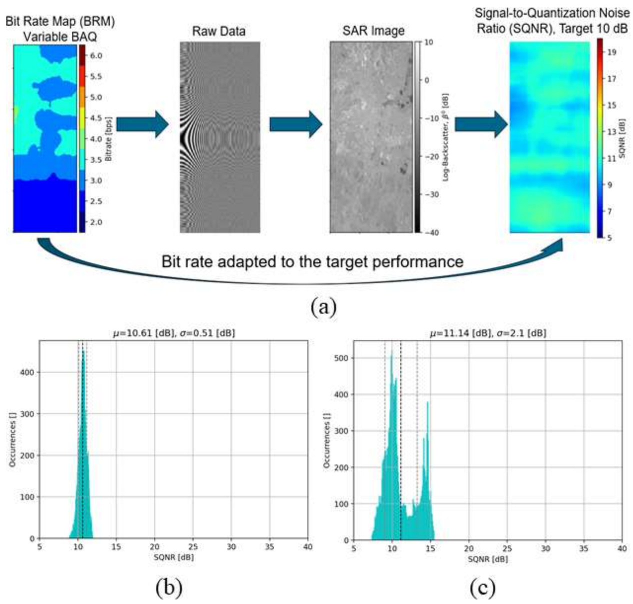

Figure 5 (a) On the left, example of BRM applied to SAR raw data for the area shown in the middle to achieve a target SQNR of 10 dB, on the right the resulting SQNR map is depicted; (b) Distribution of the SQNR values using the BRM in (a), and (c) distribution of the SQNR values in case a constant 3-bit BAQ is used for raw data compression.

5.2 Onboard SAR Image Formation

As shown in **Figure 6**, an extended wavenumber domain algorithm is selected for implementing the processing of monostatic SAR data on board.

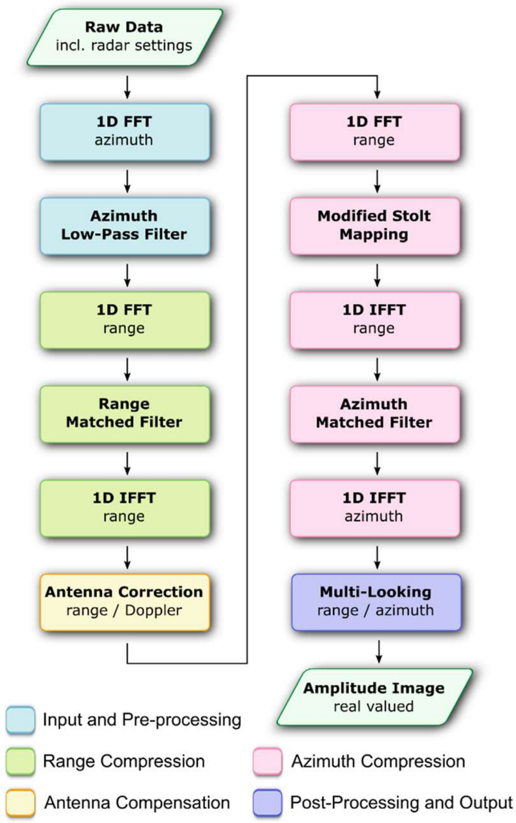

Figure 6 Flowchart illustrating the processing for on-board SAR image formation.

In particular an adaptation to the spaceborne case of the extended omega-K algorithm is selected for the following reasons [2, 3]:

- It allows updating the effective velocity for azimuth compression with range,
- It is sufficiently general to support also high-resolution SAR sensors (0.5m and better),
- It can be extended by additional blocks to support advanced SAR modes like spotlight, ScanSAR, TOPS or even HRWS.

In case of oversampled data, an azimuth low-pass filter in frequency domain is used to reduce the data volume as a first stage of data reduction prior to processing, as a function of desired azimuth resolution.

	SOPHOS High End	**SOPHOS Light**	**Sentinel-1**
Rows (range)	98946	31590	21838
Colums (azimuth)	38906	38880	48125
Low-Pass Dec. Factor	2	4	1
No of blocks	3	1	2
Acq. Time	7.5 sec	7.5 sec	25 sec
Proc. Time (Prototype)	410 sec	100 sec	207 sec

Table 2 Data dimensions for the different use cases in the SOPHOS project. For reference the numbers for a Sentinel-1 stripmap dataset are listed as well [6].

Table 2 shows the typical data set dimensions and the execution time for the developed Python prototype, which includes I/O, deformatting, SAR image formation in a monochromatic algorithm setup, multi-looking and storage, all performed on a single CPU core. The SOPHOS Light use case can be processed within one block on the SpaceCloud system, as sufficient RAM is available. However, the High-End case requires block-processing in azimuth. This is applicable also for larger azimuth scene dimensions, like for the Sentinel-1 (S-1) reference.
Within SOPHOS, only the basic algorithm for the conventional stripmap SAR mode will be optimized and benchmarked on the target SpaceCloud platform.

5.3 Parallelisation & Optimisation

The development effort in the project is focused providing highly optimized implementations of both data compression and image formation algorithms for CPU and GPU architectures. Both CPU and GPU are targeted to maximize portability and flexibility in terms of supporting a diverse set of space-borne platform hardware. In the interest of portability, the GPU development work will prioritize hardware vendor independent frameworks and interfaces such as OpenCL and/or high-level toolboxes, such as JAX. For both CPU and GPU, performance is to be maximized by comprehensive vectorization, such that the algorithms become near real-time capable and, in principle, applicable in persistent-observation scenarios. Algorithm development will combine approaches for increased robustness and reliability and avoid the pitfalls of hand-crafted low-level code optimization by leveraging techniques such as self-optimizing code.
The accuracy of the optimized SAR image formation and SAR raw data compression algorithms will be validated with respect to reference results obtained from the existing Python/C prototype implementations (corresponding to the Reference Image processing workflow of **Figure 2**). This validation process covers the use cases illustrated, as it effectively compares the high-quality and low-latency image results to the reference image. In addition, the benefit of algorithm vectorization is to be demonstrated by comparing runtime performance to the baseline Python/C implementation in application benchmarks.

6 Summary & Future Work

Whilst the concept and design work for the SOPHOS demonstrator has been concluded, much of the interfacing and implementation work between EGSE, Mass Memory and Payload Processing Module is ongoing. This is also the case for the algorithmic optimisation work.

Acknowledgment

This work receives funding from the EU's Horizon Europe research and innovation programme under grant agreement No 101082745.

7 Literature

[1] M. Martone, N. Gollin, P. Rizzoli, and G. Krieger: "Performance-Optimized Quantization for SAR and InSAR Applications" IEEE Trans. Geosc. and Rem. Sensing, doi: 10.1109/TGRS.2022.3181237. ISSN 0196-2892

[2] A. Reigber, E. Alivizatos, A. Potsis, and A. Moreira, "Extended wavenumber domain SAR focusing with integrated motion compensation," IEE Proc. Radar Sonar Navig., vol. 153, no. 3, pp. 301–310, Jun. 2006

[3] P. Prats-Iraola, R. Scheiber, M. Rodriguez-Cassola, J. Mittermayer, S. Wollstadt, F. De Zan, B. Bräutigam, M. Schwerdt, A. Reigber, and A. Moreira (2014) On the Processing of Very High Resolution Spaceborne SAR Data. IEEE Trans. Geosc. and Rem. Sens, 52 (10), pp. 6003-6016. doi: 10.1109/TGRS.2013.2294353.

[4] Unibap Space Cloud Products: https://unibap.com/space/spacecloud-products/ix10100/

[5] G. Mateo-Garcia, J. Veitch-Michaelis, C. Purcell, N. Longepe, S. Reid, A. Anlind, F. Bruhn, J. Parr & P. Philippe Mathieu (2023) In-orbit demonstration of a re-trainable machine learning payload for processing optical imagery Sci Rep 13, 10391. https://doi.org/10.1038/s41598-023-34436-w

[6] Sentinel-Online: https://sentinel.esa.int/web/sentinel/user-guides/sentinel-1-sar/acquisition-modes/stripmap