Reinventing Space Conference
BIS-RS-2017-5

System Framework for Autonomous Data Processing Onboard Next Generation of Nanosatellite

Steve Greenland, Chisato Kobayashi, Murray Ireland
Craft Prospect Ltd

Peter Mendham
Bright Ascension Ltd

David White
CREST, University College London

Mark Post
DMEM, University of Strathclyde

Bill Crowther, Khris Kabbabe
MACE, University of Manchester

15th Reinventing Space Conference
24-26 October 2017
Glasgow, UK

BIS-RS-2017-5

System Framework for Autonomous Data Processing Onboard Next Generation of Nanosatellite

Steve Greenland, Murray Ireland, Chisato Kobayashi
Craft Prospect Ltd
Tontine Building, Glasgow G1 5NA; +44 (0) 75 88 888 146
steve.greenland@craftprospect.com

Peter Mendham
Bright Ascension Ltd
River Court, Dundee DD1 3JT

David White
CREST, University College London
Gower Street, London WC1E 6BT

Bill Crowther, Khris Kabbabe
MACE, University of Manchester
Oxford Road, Manchester M13 9PL

Mark Post
DMEM, University of Strathclyde
Richmond Street, Glasgow G1 1XQ

ABSTRACT

Progress within nanosatellite systems development makes niche commercial Earth observing missions feasible; however, despite advances in demonstrated data rates, these systems will remain downlink limited able to capture more data than can be returned to the ground cost-effectively in traditional raw or near-raw forms. The embedding of existing ground-based image processing algorithms into onboard systems is non-trivial especially in limited resource nanosatellites, necessitating new approaches. In addition, mission opportunities for systems beyond Earth orbit present additional challenges around relay availability and bandwidth, and delay-tolerance, leading to more autonomous approaches. This paper describes a framework for implementing autonomous data processing onboard resource-constrained nanosatellites, covering data selection, reduction, prioritization and distribution. The framework is based on high level requirements and aligned to existing off-the-shelf software and international standards. It is intended to target low-resource algorithms developed in other sectors including autonomous vehicles and commercial machine learning. Techniques such as deep learning and heuristic code optimization have been identified as both value-adding to the use cases studied and technically feasible. With the framework in place, work is now progressing within the consortium under UKSA Centre for Earth Observation and Instrument funding to deliver an initial prototype data chain implemented within a representative FPGA-based flight computer system.

KEYWORDS: CubeSat, Nanosatellite, Autonomy, Onboard Data Processing, NewSpace, Earth Observation

INTRODUCTION

The number of small satellites, and in particular CubeSats, is increasing yearly and forecast to reach 2,400 in orbit by 2020 [1]. The recurrent engineering cost for building and launching standard platforms is benefiting from this increased demand. This has led to an off-the-shelf cost for a 3 U CubeSat platform and launch of less than 500 kGBP, excluding any mission specific developments [2]. Projects are underway to further reduce these costs and the trend might be expected to continue as new launch systems and large-scale nanosatellite constellations begin to be deployed. With cost reductions in these areas achieved, techniques to maximise cost-effective mission return during operations will become of increasing importance.

The incoming generation of multi-, hyper- spectral and high resolution CubeSat imagers are capable of generating data rates in excess of 100 Mbps [3]. This presents a challenge to the ground station infrastructure, solved to some extent by shared outsourced facilities and technological enhancements in downlink bandwidth. Despite this, the return from many missions remain downlink limited. The current state of the art is the reported as 120 Mbps X-band downlink achieved by Planet on their satellites using an extensive ground network sized for ~5 TB of downlink per day [4]. This allows the company to provide a monthly updated 5 m resolution landmass, and remains a significant USP. Performance of off-the-shelf systems available to those without Planet levels of investment are lower, with < 10 Mbps more common.

Within Earth observation, the increase in availability of data through deployment of new systems such as the ESA Sentinel satellites, and the growth in commoditised machine learning, is leading to greater on-ground automation of data product pipeline. In alternate sectors, machine learning is being applied to a diverse problem set, including autonomous vehicles, medical devices, and data centres. It is the low resource implementations on initially phones and now wearables which is of particular interest here given the resource limitations of nanosatellites.

FRAMEWORK SYNTHESIS

To evaluate the potential, a framework for enabling onboard data autonomy is proposed [5]. It was considered that alignment of this framework to existing space standards such as ECSS and CCSDS was highly desirable to allow scaling up of the solution. The framework was generated by first identifying elements of the autonomous system contrasted to more traditional EO systems in terms of typical processing chains and standards to support more autonomous behaviours. A review of previous missions utilising behaviours which may be considered within the data autonomy framework and CubeSat state of the art provided further background. Based on engagement with end users, and informed by the background, a set of high level driving requirements have been defined. This led to the definition of a proposed architecture to be assessed (a) with respect to currently available off-the-shelf software for CubeSats and (b) against the CCSDS Mission Operations Service standard.

EO Pipeline Model

The delivery of downstream products, driving typically human led responses, is a sequence of image and data processing activities within an EO Pipeline. In traditional approaches, the vast majority of these activities are conducted on the ground, as described in Figure 1. There are a number of potential barriers for the which should be considered within this work as a transition to onboarding capabilities is considered,

1. Assurance, that end users of the information have visibility of the processing chain
2. Comprehension, that users can interrogate the data at different product levels
3. Ease of access, that data may be easily returned to as improved processes emerge
4. Cost of resources, that ground-based power and processing is significantly cheaper
5. Maintenance, that data can be easily backed-up and restored in event of an anomaly

To develop a system model and evaluate the potential of integrating algorithms within the onboard segment of the EO Pipeline chain, a standardised view of the pipeline is first required. Candidates for onboarded elements may be classified as: (a) elements currently on the ground which may be incorporated on board, (b) elements adding value to the chain if included on board, (c) elements mitigating drivers identified for ground-based systems.

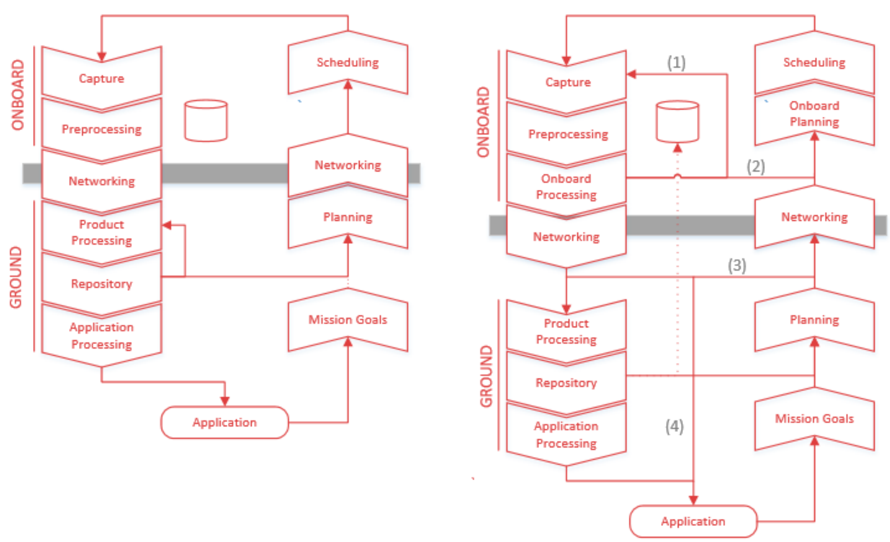

Figure 1; contrast between the more traditional processing chain and the modified pipeline integrating the autonomy-enabling processing and planning

Standardised views of the two chains may be seen in Figure 1. The traditional chain is typified by a single primary feedback loop. As might be expected for the data autonomy chain, greater feedback and feedforward loops are identified that enable adaptation of the system based on the data captured. The autonomy chain puts the application using the information within the feedback loop.

The primary feedback mechanisms within the chain are,

- In-situ analysis adapting sensor selection or reconfiguration (labelled 1 in Figure 1)
- Data capture adapting onboard planning and scheduling operations (2)
- Data capture feeding back into tasking a constellation or system of systems (3)
- Data capture informing distribution of data to a range of different targets (4)
- Repository data onboard the satellite enhancing onboard processing

Applicable Standards

As a starting point in considering autonomy high-level definitions were baselined. As such, approaches which, when incorporated within an overall system, facilitate the highest two levels of behaviour within the scale, from 'human supervised' through to 'fully autonomous' were agreed as the targets for data autonomy. As a result onboard operations would be expected to go beyond low level automation to activity and goal-based functions.

Next, ECSS and CCSDS standards were surveyed. With reference to the mission execution and data management autonomy levels defined within ECSS-E-ST-70-11, these would represent implementing behaviours described in Table 1**Error! Reference source not found.**. The functionality required by this work around the onboarding of mission data processing now extends or breaks down further these ECSS descriptions.

From other standards, particular elements were identified, such as the definition of Onboard Control Procedures in ECSS-E-ST-70-01 and the compatibility with the industry driven Packet Utilisation Standard in ECSS-E-ST-70-41. Overall, the CCSDS Mission Operations Service Standard (MOSS) framework (CCSDS-520.0-G-3) was considered most applicable as a starting point for the data autonomy architecture.

As illustrated in Figure 2, the MOSS framework is based on a Service-Oriented Architecture (SOA). SOA is defined to facilitate transition from monolithic architecture to provide a service-driven networked system for new applications. It enables a modular approach to the operational system design through the identification of components which interact through open and published service interfaces.

	ECSS Description	Functionality Anticipated
E3	Execution of adaptive mission operations onboard	Event-based autonomous operations Execution of onboard operation control procedures
E4	Execution of goal-oriented mission operations onboard	Goal-oriented mission re-planning
D2	Storage onboard of all mission data i.e. space segment independent from the ground segment [for the autonomy duration required]	Storage and retrieval of all mission data Storage management

Table 1; relevant ECSS autonomy levels for mission execution and data management extracted from ECSS-E-ST-70-11

In Fig 2, the architecture defines a set of standard services, which constitutes a framework that enables many similar systems to be assembled from compliant 'plug-in' components. These components may be located anywhere, provided they are connected via a common infrastructure. This allows components to be re-used in different mission-specific deployments: between agencies, between missions, and between systems [6]. In the data autonomy context, the service approach in principle allows modules to be exchanged between on-ground and on-board implementations.

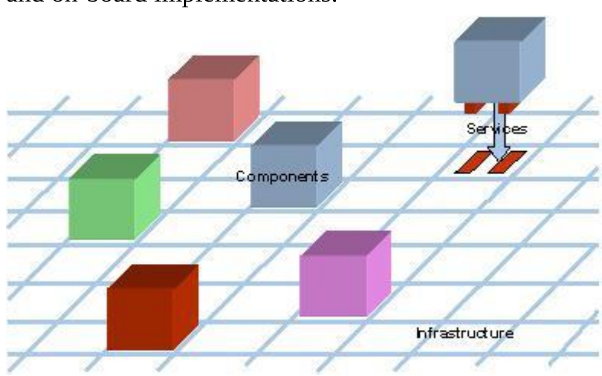

Figure 2; CCSDS-520.0-G-3 MOSS [6] definition of a Service Oriented Architecture

Requirement Generation

Based on engagement with end users, applications and mission scenarios have been consolidated into use cases for techniques to enable data autonomy. These have included near term opportunities such as onboard retasking for cloud avoidance through to more challenging opportunities like real-time reporting of targets from bistatic radar techniques. From consideration of the user and functional needs, an initial set of high level requirements for the architecture have been defined covering,

- Autonomy and Autonomy Management
- Development and Maintainability
- Data Product Management & Metadata
- Operations and Protection Systems
- Onboard Data Processing

Typical functional needs for the algorithms within this architecture as enablers for the autonomy include,

1. Onboard data processing to deliver data to a user in-the-field in near-real time for a specific targeted application
2. Reporting which nanosatellite within a constellation has captured information most suitable for a specific application for an end user
3. Providing a contextual summary of the information within images stored onboard a nanosatellite and determining which image is of highest value
4. Retasking of imaging payloads or satellites within a constellation based on data acquired either onboard or within the constellation through intersatellite links
5. Removing features from an image to minimise file size in response to a request for specific information content
6. Detecting and potentially modifying the pre-processing at the sensor readout to correct for anomalies in the data set as captured

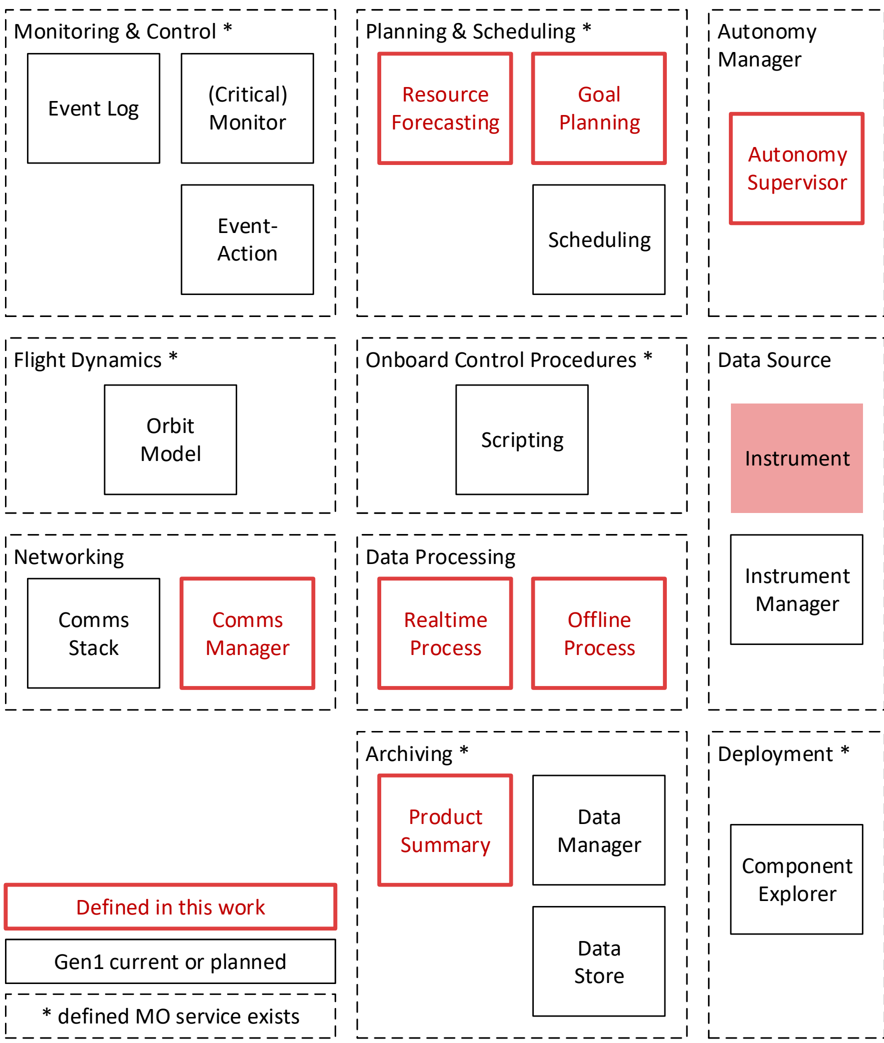

Figure 3; Data autonomy architecture, defined with respect to MOSS, identifying new services, and functional groupings based on existing standard off-the-shelf software for CubeSats

Data Autonomy Architecture

Figure 3 describes the resulting data autonomy architecture proposed, leading to the definition of three new services for the MOSS framework. It frther identifies additional functionality required within existing services based on GenerationOne, a leading off-the-shelf software package for CubeSats [7]. The new services are described below,

- The **Communications Management Service** permits the management of communications functions, controlling aspects such as link utilisation and data routing.
- The **Data Source Service** captures the ability to generate data, either from the original source, in the case of an instrument, or as an output of a data processing chain. Data can either be sourced for real-time applications ("pushed" to other onboard functions) or archived for later use.
- The **Data Processing Service** provides the ability to query and configure data processing chains which can be used with a data source or archived data.

ALGORITHM CASE STUDY

Cloud Detection Specification

The cloud detection case represents a near-term opportunity for responsive Earth observation imaging, and an opportunity to consider the implementation of the framework against a use case. Real-time detection of clouds using forwards looking wide field of view imagers allow a second near-Nadir pointing payload to be targeted at gaps in the cloud, such as for high resolution Earth imaging or to enable optical laser communications. In all avoidance scenarios, a rapid reacquisition of the payload is necessary, such that the overhead is significantly less than the active operations, < 10%. In the Earth imaging case, in subsequent passes retasking the satellite or satellites within the same network can fill the gaps, provided information on previous acquisitions can be efficiently exchanged.

A preliminary specification based on a nanosatellite system has been defined. Assuming a forwards-looking wide field of view imager, deployed on a 400 km altitude satellite, a total response time of 10 min or less will be adequate after consideration of along-track acquisition and foreshortening. This includes the capture, processing, decision, and action time to respond to the cloud avoidance input. A field of view for this imager is selected of 4 deg width pitched forwards at ~50 deg, creating a 800 km swath. A 2000 pixel width detector will provide 400 m coarse imagery of clouds. This is matched with a corresponding 0.2 deg narrow field imager, providing 1 m resolution at Nadir, at 2 km swath.

Either the satellite itself, in the case of a simple CubeSat, or payload pointing might then be altered to maximise utility of the narrow field imager. Although in the former case there may be some savings in complexity and mechanisms, the allowable motion across-track is halved against the forwards looking field of view and therefore payload actuation is preferable in most circumstances. It is noted that a benefit of creating a wider field of view 'fisheye' beyond the horizon with the forwards looking imager might also provide some attitude knowledge data. To achieve a < 10% overhead for acquisition, a retargeting rate of > 0.1 deg.s-1 is required, based on the instantaneous targetable range between -4 and +4 deg (payload repointing).

A 3 min processing time requirement for the 400 m resolved cloud images is baselined to provide sufficient margin for auctioning and acquisition. Following each 3 min update, 1 min decision time is provided to allow the current schedule to update, this will evaluate the time cost (loss of imaging) against the value of image data captured. A repointing operation lasting up to a further minute, will retarget the narrow field of view imager. The narrow field imager will then focus and begin acquisition.

Algorithmic Approaches

The problem of onboarding further capabilities such as image processing or classification can be divided between the *optimisation* of existing software to target an onboard platform and separately as the *development* of new algorithms that are more suited to onboard deployment.

Conventional approaches to cloud detection, such as Fmask developed for LandSat [7] are very processor intensive and reliant on specific wavelengths within the imager. An example output is shown in Figure 4, used for benchmarking the relative performances.

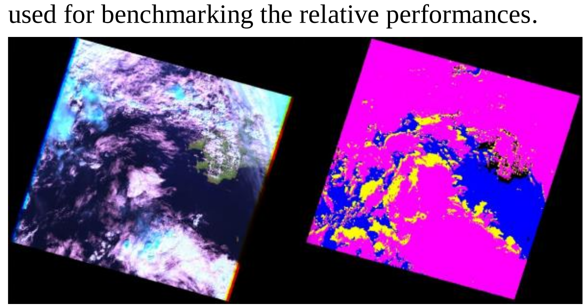

Figure 4; Fmask benchmark for evaluation of relative and absolute processing requirements

TextureCam, based on NASA machine learning developments for the Mars rovers has more recently been successfully demonstrated in orbit for cloud detection [8]. In this case, in addition to utilizing these techniques, we will be considering further approaches: heuristic code optimization acting on traditional code techniques, and deep learning.

- Heuristic methods guide optimization towards more fruitful areas of a design space with transformations not guaranteed to preserve the semantics of existing code.
- Deep learning refers to a group of learning methods and associated models used for machine learning tasks such as prediction and classification; the models are composed of connected layers of simple nonlinear processing units often described as "neurons" in a manner somewhat analogous to the brain.

INITIAL PROTOTYPING

Prototyping Approach

The challenge facing embedded algorithm implementations in FPGA logic is that procedural image processing code is very difficult to convert into VHDL or Verilog code for implementation into pure logic. Essentially, each operation performed on data must be converted into a logical operation or hard-wired arithmetic logic unit in sequence. This makes even simple floating point vector algorithms very complex in logic implementation and is the main reason that vector processing engines on accelerated graphics cards are most commonly used for computer vision. To overcome this challenge, we make use of High-Level Synthesis (HLS) methods that automate the conversion of procedural code into logical constructs. The toolchain we use for this is the recently released Xilinx SDSoC environment, an Eclipse-based software suite designed to write complete software systems, then move specific algorithms into the Programmable Logic (PL) area of a hybrid System-on-a-Chip (SoC) device with FPGA built in such as the Zynq-7000 series processors, which combine an FPGA with a dual-core ARM Cortex-A9 hard microcontroller. We use the AVNet MicroZed Z-7020 board for prototyping and testing algorithms at present due to its small size, low cost, and accessibility for code development having been in production for several years. A range of space-targeted computers using the Zynq-7000 series are available, such as the Xiphos Q7 [9].

The Xilinx SDSoC environment is used to build boot images on SD card that contain a first-stage bootloader, a Linux kernel, a complete Linux filesystem, ELF-format binaries that implement the software side (un-accelerated) of the application, and a bitstream that represents the hardware (accelerated through HLS) side of the application and is uploaded to the PL automatically on boot-up. After the HLS process, the resulting logic design is synthesized, placed, routed, optimized, and connected to the internal AXI bus for communication by the Vivado software suite. The Linux kernel and filesystem are derived from Xilinx' PetaLinux distribution, which can be easily customized for use on SoC and FPGA based processors. This process is illustrated in Figure 5. As HLS methods are limited in the complexity of code that can be converted to pure logic, the OpenVX framework for vision acceleration and a small set of OpenCV functions alone are available for implementation in logic.

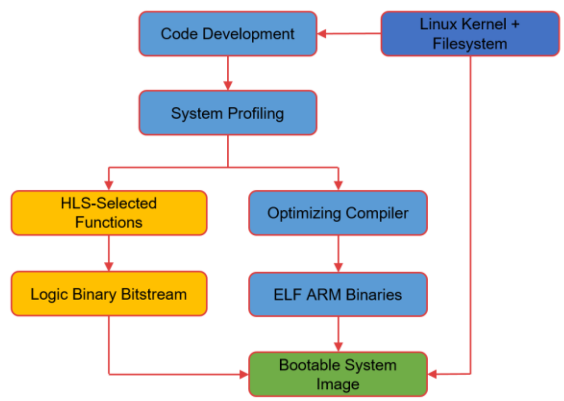

Figure 5; Implementation process for prototyping algorithms onto the FPGA

The focus of work at this stage is to 1) validate that a functioning vision system can be produced using HLS on the Zynq platform, to 2) estimate the degree of acceleration of core vision functions that can be accomplished using HLS on the Zynq platform, and to 3) verify that comparable results are obtained to a purely software based vision system. Ultimately, both stereo vision reconstruction and neural network based machine learning is targeted for acceleration, but as the toolchain is quite new, only dense non-pyramidal optical flow, Harris corner detection, and bilateral filtering are available for the Z-7020 devices (neural network implementations in the Caffe framework currently require the use of a larger UltraScale-series device with a Cortex-A53 multi-microcontroller core). Consequently, our benchmarking currently makes use of only dense non-pyramidal optical flow and Harris corner detection algorithms.

Performance Validation

Validation of a functioning system can be done both using file I/O for sending image files to the PL processing area, and "live" using a stream of images from HDMI or camera input. As the MicroZed board has neither HDMI input nor camera interface, we instead provide a stream of stored image files that are processed on board the device, with timing information gathered on a per-frame basis. To estimate the degree of acceleration for the logic-based processing of frames, we run two tests on identical image sets. The first test ("software") is run using optical flow and Harris corner detection algorithms implemented in traditional software functions using OpenCV on ARM9. This is the baseline for performance and produces good results but at a slow speed. The second test ("hardware") is run using software functions for reading and sending the images, but hardware functions for the actual processing of the images to produce optical flow vectors and feature descriptors respectively. The resulting image processing timing and outputs from both software and hardware tests are then compared.

CONCLUSIONS & FUTURE WORK

A framework for facilitating onboard data autonomy has been presented, based on user needs, assessment of state-of-the-art, and applicable standards. Contrasted against these standards and off-the-shelf software, opportunities to further develop and facilitate autonomy have been identified in terms of new services and functionality.

Application use cases, such as the cloud detection scenario described in this work, will drive work to prototype and benchmark both the onboard systems and the enabling algorithms. This will facilitate more detailed mission-specific trades as to where these techniques can provide the most value, and provide insight into differing techniques.

Follow-on opportunities to demonstrate the selected solutions have been identified, in particular through the GAMMA program at University of Manchester, and the UK In-Orbit Demonstration program. As such, a CubeSat for OnBoard Realisation of Autonomy (COBRA) mission is in development, with the goal of forming a collaboration amongst a wide base for onboard autonomy activities. For further information please contact the authors.

ACKNOWLEDGEMENTS

This work is being supported by the UK Space Agency and the Centre for Earth Observation as a Pathfinder RP10G0435B04.

For further information and reports from the outputs and application of this work, please contact Craft Prospect Ltd., steve.greenland@craftprospect.com.

The authors would like to thank Adam Taylor of Ardiuvo Engineering, for helpful conversations and support during this work.

REFERENCES

[1] SpaceWorks Enterprises Inc, "Nano/micro satellite market forecast 2017," 2017.

[2] OpenCosmos, www.opencosmos.com.

[3] Greenland, S, et al. "NANOBED-MX: international collaboration for nanosatellite real time surveillance missions," International Astronautical Congress 2016, Mexico, IAC-16-B4-3-10.

[4] Henry, C, "Planet Preps Ground Network to Handle 5 Terabytes of Data Per Day," www.satellitetoday.com, August 2016.

[5] Greenland, S, et al. "Onboard EO Autonomy: Architecture & Requirements," Craft Prospect Ltd., August 2017.

[6] CCSDS, "Mission Operations Service Standard" Greenbook, CCSDS-520.0-G-3.

[7] Bright Ascension Ltd, www.brightascension.com

[8] Thompson, D, et al. " TextureCam: A Smart Camera for Microscale, Mesoscale, and Deep Space Applications," Jet Propulsion Laboratory, 2013.

[9] Xiphos Ltd., www.xiphos.com