VIMOS - MODULAR COMMANDING AND EXECUTION FRAMEWORK FOR ONBOARD REMOTE SENSING APPLICATIONS

Katharina Götz, Kurt Schwenk, Felix Huber

German Space Operations Center (GSOC), German Aerospace Center, 82234 Weßling, GERMANY

ABSTRACT

For decades, field help in case of disasters on the Earth's surface - like floods, fires or earthquakes - is supported by the analysis of remotely sensed data. In recent years, the monitoring of vehicles, buildings or areas fraught with risk has become another major task for satellite-based crisis intervention. Since these scenarios are unforeseen and time-critical, they require a fast and well coordinated reaction.
If useful information is extracted out of image data in real-time directly on board a spacecraft, the timespan between image acquisition and an appropriate reaction can be shortened significantly. Furthermore, on board image analysis allows data of minor interest, e.g. cloud-contaminated scenes, to be discarded and/or treated with lower priority, which leads to an optimized usage of storage and downlink capacity.
This paper describes the modular application framework of VIMOS, an on board image processing experiment for remote sensing applications. Special focus will be on resource management, safety and modular commandability.

Index Terms— Remote Sensing, Onboard, Modular Commanding, Image Analysis, Image Processing

1. INTRODUCTION

Images acquired by Earth observing satellites have to be stored on board before they can be sent down to Earth when within view of a ground station. Due to the high data volume, several passes may be needed to downlink a complete acquisition (cf. [1]), hence the whole procedure from image acquisition to image analysis can take hours to days.
In case of disasters it is often not necessary to view an acquisition, but in the first instance to simply extract information, such as coordinates and extent of a flood. Apart from this, an acquisition may not be of interest for the actual purpose, and hence, could be discarded directly on board or treated with minor priority. This is, for example, useful in case of cloud-contaminated scenes. The extraction of valuable information out of image data in real-time directly on board a spacecraft can shorten the timespan between acquisition and reaction drastically as well as optimize the usage of storage and downlink resources significantly (cf. [2]). Altogether, on board image analysis provides considerable benefit for conservative remote sensing processing chains accomplished on ground.

2. VIMOS - AN EXPERIMENT ONBOARD OF BIROS

As successors of the German BIRD satellite, TET-1 and BIROS are dedicated to the detection of wildfires and hotspots. The constellation of both will be referred to as the FireBird mission (see [3]). While TET-1 has been brought to its orbit in August 2012, the launch of BIROS is currently scheduled to the midyear of 2015. Both satellites carry the same primary payload. It is composed of three cameras, one covering the visible spectral range (VIS), i.e. green, red and near-infrared with a spatial resolution of about $42\,\mathrm{m}$. The other two sensors operate in the mid-wave infra-red and the thermal infra-red region, respectively, and have a spatial resolution of about $178\,\mathrm{m}$. The main purpose of FireBird is the detection of hotspots on the Earths surface.

VIMOS (Verification of Image Analysis Onboard a Spacecraft) is a first demonstration of an image evaluation system based on a modular framework, specially designed to perform remote sensing applications on board a satellite.
It will run as a software experiment on board of BIROS (Berlin InfraRed Optical System). VIMOS consists of algorithms applied to data of optical sensors, dedicated to an operation on board a satellite, like e.g. cloud detection, change detection and object recognition. On board of BIROS, images acquired by the VIS camera will be used.
VIMOS will be able to send trigger messages to the autonomous on board mission planning unit, which can check possibly modified conditions on board, and react accordingly e.g. by discarding an acquisition or planning a new timeline. In a later stage of the experiment it is intended to forward the output of VIMOS (e.g. warnings) to the onboard Orbcomm modem (http://www.orbcomm.com/), which is capable to downlink small information packages without direct contact to a ground station. In the future it is aspired to have the results of image analysis available on ground within minutes after acquisition.
The highly modular application and commanding concept of VIMOS provides the adaptability to basically every other

optical sensor. This was tested already with data from TET-1, Landsat-5, Landsat-7 and RapidEye. Furthermore, the modularity allows an easy expansion and the porting to other architectures with comparably little effort, such that dedicated processing steps can be implemented directly in hardware. This is currently done using the hybrid board XILINX Zync-7000.

2.1. Onboard application framework

VIMOS will be an application on the Payload Processing Unit of BIROS, which is based on a Xilinx Virtex4 FPGA (Field Programmable Gate Array) with to embedded PowerPC cores. On top of them cores runs an adapted version of the satellite operating system RODOS (Real Time Object Oriented Dependable Operating System). RODOS supplies quasi-parallel processing and multi-threading and permits programming in C/C++ (see www.dlr.de/rodos). A BIROS-specific service-layer provides interfaces to mass memory, AOCS-data, logging partition etc.
VIMOS is divided into two separate threads, VIMOSmain and VIMOSprocessing. While VIMOSmain is responsible for controlling, timing and command reception, VIMOSprocessing performs the actual image processing.
VIMOSprocessing mainly consists of commandable methods, called *c-modules*. Most of the *c-modules* correspond to algorithms commonly used in the field of remote sensing and image analysis, among them: data conversion (e.g radiance/reflectance at top-of-atmosphere), morphological filters, edge detection filters, feature detection and feature matching, spectral thresholding and spectral indices, segmentation, connected component labelling etc.. Additionally, there exist *c-modules* for reading AOCS (Attitude and Orbit Control System) data, target-finding (given by geographic coordinates), subset selection, RAM (de-)allocation, reading/writing data and parameter adjustment.
Since modularity is often accompanied by complexity, the outer structure of all *c-modules* has been kept as simple as possible: Each *c-module* can be addressed by a unique ID (*c-ID*). One single variable (*c-params*) is passed to each *c-module*, containing all necessary parameters for the *c-module* to run. A *c-module* returns always an acknowledge. The worst case is a negative acknowledge, which leads to an immediate, but proper termination of the processing chain. VIMOS will then return to standby mode. System crashs or hang-ups are hereby precluded, even if commanding is defective.
To process an image with respect to a desired result, e.g. cloud detection, *c-modules* are executed in the commanded user-defined order of *c-IDs* and with commanded user-defined *c-params*. Choosing another composition of *c-IDs* with other *c-params* results in another usecase.
When applying VIMOS to other missions, changes may have to be made in satellite-depending interfaces and operations like reading/writing data or RAM (de-)allocation. However, the overall framework of VIMOS and the major part of the algorithms is kept independent of e.g. the number of channels or the spatial and spectral resolution, and hence can be deployed to other sensors without any modifications.

2.2. Commanding and controlling VIMOS

The controlling of VIMOS is based on three types of lists and three corresponding managers: preparation lists (*prepLists*), processing lists (*procLists*) and one unique coordination list (*coordList*). A simplified scheme is depicted in figure 1.
Every list can be filled independently by commands. Each instance of the first two types is addressable by an identifier and initially set to *inactive*. Commands for populating such a list must contain the instance identifier followed by an ordered alternating sequence of *c-IDs* and *c-params*. Commands, which are meant for filling the *coordList*, have to contain pairs of *prep-/procList* instance identifiers.
The activation manager uses the *coordList* to activate the next pair of *prep-/procLists* and transfers them to their managers as soon as a data-ready-signal is received.
PrepLists contain all acquisition-dependent *c-modules*, e.g. target finding with geographic coordinates, opening and reading the desired acquisition.

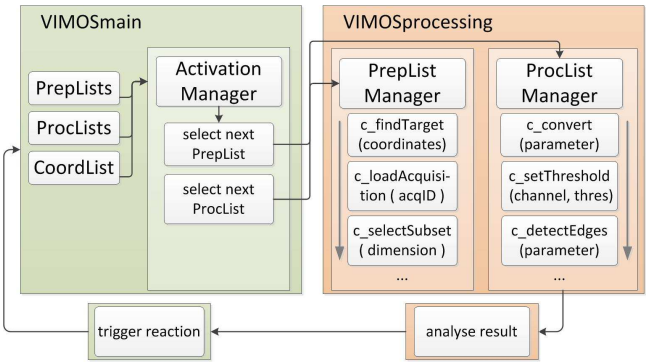

Fig. 1. VIMOS - Overall framework scheme (simplified)

The entries of the active *prepList* are consecutively interpreted by the *prepList* manager, which calls the *c-module* corresponding to the next *c-ID* and hands over the respective *c-params*. The output of a *prepList* is usually an acquisition subset of commanded width and height with the target (if commanded) approximately in the center. A subset can involve one or more spectral channels.
Subsets serve as input for the *procList* manager. *ProcLists* consist of all *c-modules* which are independent of the selected scene. For example, the order of *c-IDs* and *c-params* in *procList1* may result in the detection of clouds, water and vegetation, and *procList2* the recognition of an object. The reaction depends on the outcome of the image analysis; a warning may be generated and sent to Earth or the deletion

of an acquisition may be triggered. Visual results like masks can be saved and downlinked optionally.
PrepLists and *procLists* as well as their managers have the same structure and functioning and could be technically combined. However, splitting acquisition-dependent and -independent *c-modules* facilitates the definition of default *procLists*, which can be reused for arbitrarily many scenes.

2.3. Data interface

All data, i.e. images as well as header and AOCS information, will be fetched by VIMOS via service-layer interfaces from the mass memory, where they are stored as source-packets in the Packet Utilization Standard (PUS). Every packet can be addressed by the application ID of its source device and the time stamp of its creation. A source-packet consists of a descriptive header followed by the actual data. Images are stored channel-wise and line by line and have, depending on the camera operation mode, one ore two bytes per pixel.
During an acquisition, AOCS packets, containing i.a. attitude, satellite and solar position, will be created every 500 ms. From these together with the camera's calibration data, VIMOS will calculate repeatedly the region of sight on the Earth's surface for one spectral channel (set by command) until the commanded target is found or a predefined loop limit is reached. If the target is found, this channel will be opened at the data line, which corresponds to the time stamp of the target's AOCS packet. Using the service-layer's reverse- and forward functions, further data lines will be read consecutively into a previously allocated image frame, such that the target lies (if possible) approximately in the middle.
Due to the design of the VIS camera, the three spectral channels will be recorded with a time offset with respect to each other and hence, are not congruent. The amount of the offset depends on the camera's operation mode and due the Earth's curvature and the satellite's movement the resulting distortion is non-linear. If the desired usecase involves more than the selected channel, VIMOS will compute about the appropriate subsets of the remaining channels and warp them to each other.

2.4. Resource management

Although VIMOS will be written in C++, several comfortable programming practices will be banned for safety reasons, in particular dynamic allocation. There will be 16MByte RAM exclusively and statically reserved for VIMOS, which is comparably poor in the context of image processing. But, since VIMOS shall be able to execute various modularly defined processing chains, each of them having different memory requirements, the RAM usage must be dynamically customizable. Therefore, a commandable intra-allocation concept was designed to control RAM operations exclusively on the statically reserved block. Whenever a *c-module* needs memory (e.g. for an image frame) as an input, output or temporarily, the corresponding RAM will be intra-allocated in advance by a command, i.e. intra-(de-)allocation operations are *c-modules* themselves. In addition, this approach has the advantage, that the image frame size is kept adjustable, while at the same time the system is prevented from overflows, since the peak usage of RAM capacity will be known already on ground when defining the processing chain.
VIMOS will run on the Payload Processing Unit (PPU), which is also responsible for handling the acquisition stream and storing it in packets on the mass memory. Naturally, while an image is recorded, this procedure is given the highest priority regarding processor usage, which is distributed by the service layer. Technically VIMOS could be carried out online, i.e. while image acquisition is still ongoing. But, since VIMOS will have a lower priority, it may be repeatedly interrupted and resumed. Whether on board of BIROS online processing is reasonable, will be known after integrating all applications into the engineering model of the PPU device. The same applies to speed and run time evaluation.

2.5. Application Examples

As mentioned before, the modular processing framework supports a variety of applications for a variety of sensors.

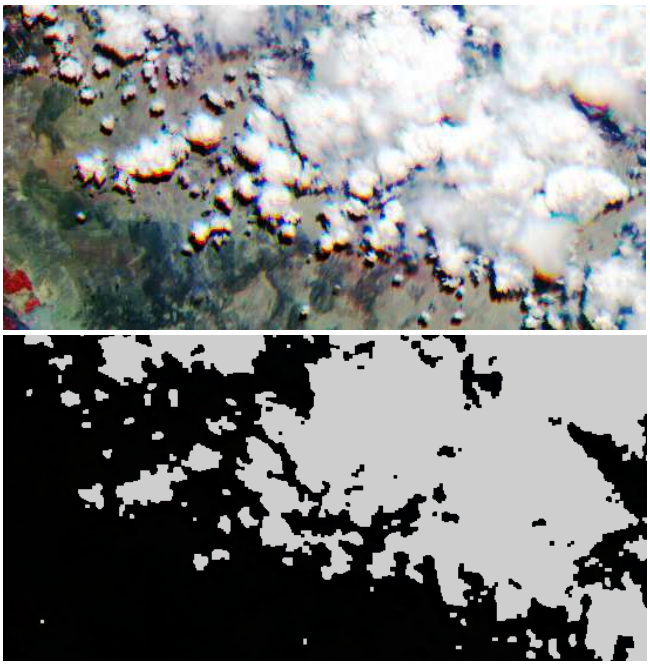

Fig. 2. VIMOS cloud cover evaluation
(t) TET 1 - 2014/07/02 - Central China (source: DLR e.V.)
(b) Underestimated cloudmask derived from TET raw data

In the following two examples are presented, which are

among others planned to be performed on board of BIROS. The choice of the usecases is induced by the spectral and spatial resolution of the sensor. Data from TET-1 and Landsat 5/7 are used for development. For a better understanding the results are visualized. Apart from the actual detection modules, all examples include also pre-processing steps like spectral conversion, band-coreferencing or georeferencing. Further information can be found in [4].

Cloud cover evaluation

Whether cloud detection is used as a stand-alone application or preliminary to another usecase, it will always be an essential step when processing data from optical sensors. The cloud cover percentage can be calculated e.g. by applying threshold criteria to spectrally and radiometrically corrected images.
In the case of targeting a possible deletion of the scene directly on board, an underestimation of the cloud cover is desirable, in order to not discard scenes actually useful. In contrary, if clouds shall be masked out for proceeding with a subsequent application, overestimation is preferred to reduce the probability of misinterpretation. For both intentions morphological filters (erosion / dilatation) can be applied additionally, yielding a shrinkage or expansion of the detected cloud areas. An example result for an underestimated cloud detection can be seen in figure 2. It was derived from TET-1 raw image data by the VIMOS on ground development version.

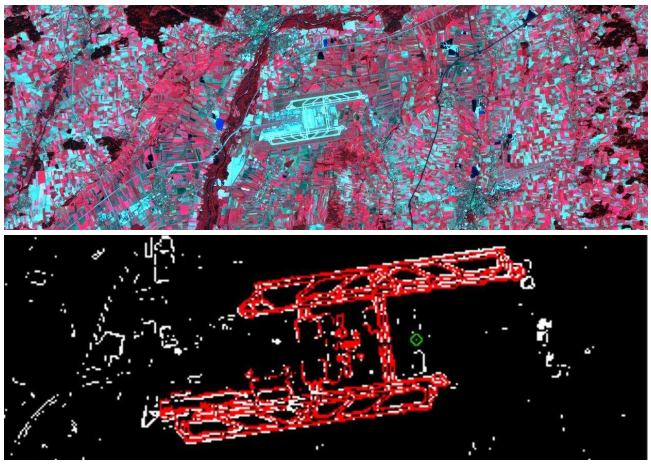

Fig. 3. VIMOS object recognition (hough transformation)
(t) False colour composite Landsat 7 ETM+
(b) Munich Airport recognized by VIMOS

Object recognition

Object recognition from space has become an important factor for security management. Locations, which are supposed to be fraught with risk, are surveyed regularly e.g. on behalf of governments. For instance, the German Center for Satellite Based Crisis Information (DLR-ZKI, see www.zki.dlr.de), observes the ship traffic in selected regions and monitors certain buildings like for example embassies, government buildings and football stadiums.
On board of BIROS, VIMOS will concentrate on larger objects like lakes or airports due to the spatial resolution of 40 m in the VIS-range. Currently there are two object recognition algorithms implemented: the SURF feature detection and matching algorithm (see [5]) and the general Hough transformation (see [6]). Figure 3 depicts how the Munich airport is recognized by the VIMOS on ground development version using the latter method.

3. CONCLUSION

VIMOS will contribute to the vision of smarter satellites by demonstrating that the potential of observing the Earth from space has not been exhaustively utilized by now. The aim of VIMOS is to validate, that it is possible to extract time-critical information out of image data directly on board of satellites and initiate appropriate reactions autonomously. It has already left the stage of a coarse conception. Since it is a running project, the state of VIMOS changes permanently.

4. REFERENCES

[1] A.S. Dawood, S.J. Visser, and J.A. Williams, "Reconfigurable fpgas for real time image processing in space," in *14th International Conference on Digital Signal Processing (DSP) 2002*. IEEE, 2002, vol. 2, pp. 845–848.

[2] S. Yuhaniz, T. Vladimirova, and M. Sweeting, "Embedded intelligent imaging on-board small satellites," *Advances in Computer Systems Architecture*, pp. 90–103, 2005.

[3] G. Ruecker, E. Lorenz, A.A. Hoffmann, D. Oertel, J. Tiemann, and W. Halle, "High resolution active fire monitoring for global change analysis: The upcoming firebird satellite mission," *The 5th International Wildland Fire Conference*, 2011.

[4] K.A.M. Goetz, F. Huber, and M. Schoenermark, "Vimos-autonomous image analysis on board of biros," in *Automatic Control in Aerospace*, 2013, vol. 19, pp. 423–428.

[5] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, "Speeded-up robust features (surf)," *Computer vision and image understanding*, vol. 110, no. 3, pp. 346–359, 2008.

[6] D.H. Ballard, "Generalizing the hough transform to detect arbitrary shapes," *Pattern recognition*, vol. 13, no. 2, pp. 111–122, 1981.

[7] M.A. Fischler and R.C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981.