14th Symposium on Advanced Space Technologies in Robotics and Automation, Jun 2017, Leiden, Netherlands. Proceedings of the14th Symposium on Advanced Space Technologies in Robotics and Automation, 2017.

INFUSE: A COMPREHENSIVE FRAMEWORK FOR DATA FUSION IN SPACE ROBOTICS

Shashank Govindaraj(1), Jeremi Gancet(1), Mark Post(2), Raul Dominguez(3), Fabrice Souvannavong(4), Simon Lacroix(5), Michal Smisek(6), Javier Hidalgo-Carrio(3), Bilal Wehbe(3), Alexander Fabisch(3), Andrea De Maio(5), Nassir Oumer(6), Vincent Bissonnette(4), Zoltan-Csaba Marton(6), Sandeep Kottath(5), Christian Nissler(6), Xiu Yan(2), Rudolph Triebel(6), Francesco Nuzzolo(1)

(1) Space Applications Services NV, Leuvensesteenweg 325, 1932 Zaventem, Belgium,
Email: jeremi.gancet@spaceapplications.com
(2) Strathclyde University, Richmond Street 16, G1 1XQ, Glasgow, UK,
Email: mark.post@strath.ac.uk
(3) DFKI, Robert-Hooke-Straße 1, 28359 Bremen, Germany,
Email: raul.dominguez@dfki.de
(4) Magellium SAS, 24 rue Hermes, 31521, Ramonville St Agne, France,
Email: fabrice.souvannavong@magellium.fr
(5) LAAS-CNRS, 7, avenue du Colonel Roche, 31031 Toulouse, France
Email: simon.lacroix@laas.fr
(6) DLR, Münchener Str. 20, 82234 Weßling, Germany,
Email: michal.smisek@dlr.de

ABSTRACT

Fused sensory data provides decision-making processes with exploitable information about the external environment and a robot's internal state. This paper describes some preliminary work on the InFuse project to create a modular and portable data fusion system funded by European Commission's Horizon 2020 Strategic Research Cluster on Space Robotics Technologies. In space robotics, a wide range of data fusion techniques are required to accomplish challenging objectives for exploration, science and commercial purposes. This includes navigation for planetary and orbital robotics, scientific data gathering, and on-orbit spacecraft servicing applications. InFuse aims to develop a comprehensive open-source data fusion toolset to combine and interpret sensory data from multiple robotic sensors, referred as a Common Data Fusion Framework (CDFF).

1. MOTIVATIONS

The process of data fusion is at the heart of autonomous robotic systems, and plays a key role in comprehensive automated perception systems. In this paper, we present the initial work on InFuse: a modular, portable and robust data fusion system for robotics applications in space environments, both for orbital servicing and planetary exploration. Sensor fusion frameworks [1] have been developed for selecting suitable algorithms for a specific set of sensors. The InFuse Common Data Fusion Framework (CDFF) is being developed with expertise from a consortium of partners that have substantial experience with sensory data handling and processing techniques for perception and navigation in a wide range of space and terrestrial robotic applications. The approach taken in InFuse to handle data fusion algorithms and data products is intended to make their adoption easier and more effective by a wide range of users, both among the H2020 Space Robotics SRC stakeholders and in the wider space robotics community.

Outline: after a review of the objectives and design drivers for the CDFF, section 3 provides a brief overview of the main data fusion techniques that are relevant for the space robotics context. Section 4 then depicts the architecture of the CDFF. Finally, section 5 sketches the exploitation perspectives of the CDFF.

2. OVERALL APPROACH AND KEY CDFF DESIGN DRIVERS

2.1. Prime Objective and Overview

InFuse aims to develop a comprehensive data fusion toolset for robot sensors that will serve in the context of future space robotics and possibly ground based applications. The InFuse CDFF provides access to an extensive set of robust data fusion capabilities that are relevant both for on-orbit servicing and planetary exploration scenarios. These allow a user to control the data fusion processes and to conveniently retrieve (on-demand) products such as maps, models of the environment and objects, and relevant science data. Furthermore, the InFuse CDFF is being designed with the objective of remaining independent (agnostic) of any particular robotic middleware. Deploying InFuse towards a particular robotics middleware such as ESROCOS, ROS, ROCK or GenoM will nonetheless be supported by dedicated InFuse tools. It also includes data fusion orchestration and product management tools.

InFuse provides navigation and perception functionalities. Navigation pertains to positioning in general: of a rover, of particular elements in the environment, of a servicing satellite with respect to an object in space to service. Perception pertains on the one hand to the detection and modelling of the environment (be it the whole environment surrounding the rover or some specific elements in the environment), and on the other hand to track specific environment features during motions. Mature state of the art perception and localization libraries will be integrated into the framework to cover these objectives, as well as more commonly used libraries within the robotics domain.

From a research perspective, InFuse will develop a formal model of perception activities, aiming to allow their autonomous optimization and configuration depending on the task and the context. The goal is to improve the quality of the output through optimization and planning techniques, by implementing active perception schemes [2]. An essential part of the model of the data processing and fusion functions is the definition of figures of merit associated to the produced data types.

2.2. Reference Scenarios

InFuse addresses perception and data fusion within the context of two reference scenarios of space robotic: planetary exploration and on-orbit servicing, as depicted in Fig. 1. Perception and data fusion capabilities are selected for answering the need to perceive, analyse and interpret the environment as well as satisfying constraints related to reliability, memory footprint and computation complexity, inherent to space vehicles.

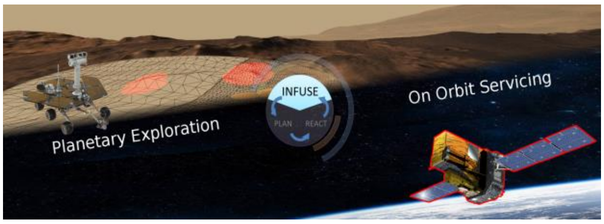

Figure 1. Illustration of reference scenarios addressed by InFuse.

The planetary track focuses on surface exploration, autonomous navigation [11] and science, and rendezvous. A nominal sequence of tasks done by a rover is : a) Creation of an initial panorama, b) Production of navigation maps, c) Path planning, d) Path execution and hazard detection / avoidance, e) Rover self-localization (odometry / SLAM), f) Updating of navigation maps, g) Back to d. or detection of a point of interest - POI, h) Visual servoing towards the POI i) Soil sample acquisition, j) Global localisation using orbiter data, k) Going back to lander, l) Visual servoing towards the lander, m) Visual servoing of the robotics arm for sample transfer.

The orbital track focuses on rendezvous for on-orbit servicing operations such as refueling, re-configuration of hardware modules and repair of parts. A nominal sequence of tasks performed by a chaser spacecraft is: a) Detection of a target satellite far away (it might be done manually by ground observation to determine its orbit), b) Bearing tracking, c) Initial approach, d) 3D modeling of the satellite, e) 3D tracking, f) Final approach, g) Visual servoing of the robotics arm, h) Docking and berthing.

Within these scenarios, InFuse deals with perception and data fusion. This translates into data production for the Autonomy Framework. Three data classes are identified: localization (self or w.r.t a target), landmark / object management (detection and tracking), environment modeling (DEM, navigation maps, structured 3D models). From this point of view, we identify 9 data production use cases covering both reference scenarios:

Planetary exploration:
a) Rover localization in its environment [10, 12],
b) Relative localization w.r.t a fixed or moving asset,
c) Production of DEM and navigation maps [13],
d) Production of a panorama, Detection of point of interest,
e) Rover collaboration.

One possible illustration of the use case “Rover localization in its environment” is provided in Fig. 2.

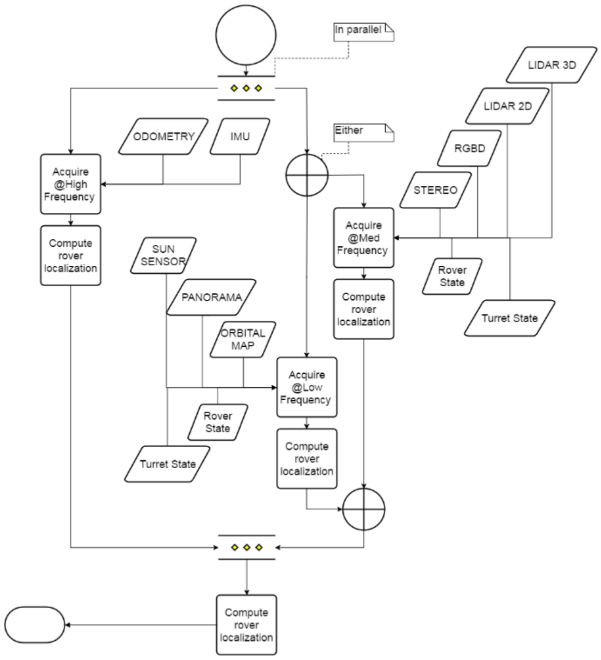

Figure 2. Functional description of the use case “Rover localization in its environment”

On-orbit servicing:

a) Bearing only localization for approach (target position and direction estimation w.r.t. chaser / Long range),
b) Localization w.r.t satellite for rendezvous and orbital parameter estimation (target pose estimation w.r.t. chaser close perception) [14],
c) Target pose estimation w.r.t. chaser / Docking (Visual servoing),
d) 3D reconstruction of a target.

2.3. Sensors and data types

The CDFF primarily supports sensory input from the standard suite of sensors currently used in orbital and planetary applications (Tab. 1). In addition to that, CDFF is designed to allow for including additional sensors - sun sensor, star tracker, radar, ultrasonic sensor, magnetometer, multi/hyperspectral camera. The possibility to further extend the CDFF to include support of other types of sensors in the future is available by design.

Table 1. Sensor data and associated sensors types

Type of Data	Name of Sensor
Image Data	Close-up High-Resolution Camera
Image Data	Light field camera (2-D images, ext. DoF)
Extended Image Data	Hyper / Multispectral Camera
Depth Image Data	TOF Camera
Depth Image Data	Structured Light Vision
Depth Image Data	Visible Stereo Camera
Depth Image Data	IR Stereo Camera
Depth Data	3D LIDAR
Planar Depth Data	2D LIDAR
Angular Range Data	Multi Angle Radar
Range Data	Narrow Angle Radar
Range Data	Ultrasonic Sensor
Force Data	Contact Sensor
Torque Data	Force/Torque Sensor
Angular Data	Joint Position Encoder
Angular Pose Data	Star Tracker
Angular Pose Data	Sun Sensor
Angular Magnetic Data	Magnetic Field Sensor (Magnetometer)
Angular Velocity Data	Angular Velocity Sensor (Gyroscope)
Angle of Acceleration Data	Linear Acceleration Sensor (Accelerometer)
Arbitrary Data	"Alternative" or General Sensors

Data types for raw and pre-processed sensor data, and fused data products will predominantly be based on the Rock middleware base types [7, 8]. The ESROCOS [3] robotics software framework uses ASN 1.0 to encode the Rock base types. In specific cases, the data types need to be enhanced to include metadata relevant for data fusion processing steps and propagated to other higher-level components within the system.

2.4. Requirements highlights

CDFF primarily aims to satisfy a core set of requirements as per the SRC Compendium [4] that are relevant for space robotics applications. Additionally, InFuse will address requirements from the scientific community making it feasible for quality and performance analysis among data fusion algorithms, provide a comprehensive repository data fusion methods at different levels of granularity and facilitate porting solutions to target robotics middlewares. The main CDFF requirements are as follows:

1. Facilitate relative and absolute localization with respect to structured and unstructured objects.
2. Support reconstruction of terrain and 3D models to represent environment geometry and scene interpretation.
3. Allow detection, reconstruction and tracking of objects and landmarks in the environment.
4. Data fusion processes can deal with complementary, redundant and cooperative data.
5. Contain algorithms devoted to robot state estimation and outlier removal.
6. Designed to be compatible with space qualified processors running RTEMS (Real-Time Executive for Multiprocessor Systems) and satisfy processing and memory constraints at runtime.
7. Support the fusion of data from a wide range of proprioceptive and exteroceptive sensors.
8. Provide a data product management tool to store and retrieve fused sensor data with associated metadata.
9. Development environment tools to run data fusion methods from data logs, visualisations and evaluation of camera calibration and automatic (re-) calibration of sensors to each-other.

3. REVIEW OF SPACE ROBOTICS RELEVANT DATA FUSION TECHNIQUES

3.1. Taxonomy of Data Fusion Techniques

The purpose of establishing a taxonomy of DF processes is to have a deep and complete understanding (and thus representation) of the nature of the entire perception layer. This grants the possibility to adopt a formal approach to the composition of complex data fusion processes based on the desired data product. On a purely functional level, the taxonomy may serve as guide during the reconfiguration of a data fusion process. The formalization of data fusion techniques allows to quickly look-up for potential candidates that may perform better under the current situation compared to the currently used one. In other words, the CDFF will know in advance what are the applicable techniques capable to satisfy the needs for a given data product.

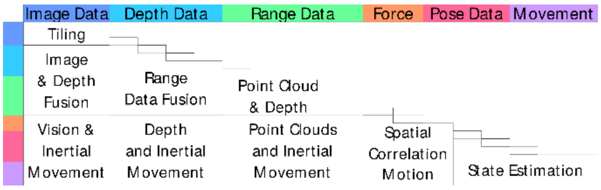

Figure 3. Abbreviated Representative Matrix of Sensors & Data Combination Methods (see https://www.h2020-infuse.eu for complete matrix)

With the goal of ensuring that as many fusion combinations could be represented as possible, a lower-triangular matrix was constructed with the list of sensors across the horizontal and vertical axes intended to represent a “continuum” of similar data types. In this way, fusion methods more appropriate for similar data types are clustered together within the matrix. The matrix itself with annotations to facilitate understanding of these “clusters” is shown in Fig. 3. As a broad introduction to the data fusion methods included in this matrix, we have identified the following core groups of methods that will be required for data fusion.

1. For fusing image data, image stitching methods using RANSAC feature matching and Bayesian filtering are available for Multiple View-Point reconstruction, detection of a pattern of keypoints for calibration can be used, as well as plenoptic odometry (3D+2D from single camera) and 3D map generation with 2D-texture from same sensor.
2. To fuse images and depth data, 3D based odometry with image based orientation determination can be used on image data, and data fusion accomplished by Extended Kalman Filter or directly in a 3D based odometry algorithm. Texture mapping can be used in a 3D Reconstruction with coloured textures, and monocular localization can use the range provided by another sensor in order to measure the scale.
3. Depth data can be fused with depth data by merging reconstructions either as raster or point cloud, or by constraining a disparity search thanks to the use of a different depth sensor. Visual point clouds can be merged with radar point clouds after transformation of range scans with consideration of the different resolutions and range by sensors. Classification of materials is also possible by radar.
4. Visual data can be combined with angular pose data, such as from an IMU, by monocular localization and the use of localization information provided by the pose sensor by means of a Kalman Filter variant or directly in the 3D based odometry algorithm.
5. Depth data can also be combined with angular pose data by means of 3D based odometry with localization information provided by the pose sensor by means of a Kalman Filter variant or directly in the 3D based odometry algorithm.
6. Contact sensors and force/torque sensors can be fused by measurement of forces and torques at different contact points/joints to build a coarse map by just sensing surfaces and provide pose corrections (Ex. collisions causing wheel slippage or while overcoming obstacles).
7. Pose velocity and acceleration data can be fused using a multi-sensor fusion approach combining 6D force/torque, 6D acceleration, angular velocity, and joint angle data to estimate set of inertial parameters (i.e. the mass, the coordinates of centre of mass, and the elements of the inertia matrix) of an object gripped by or attached to a manipulator.
8. Fusion of inertial sensors is commonly and easily done by means of Kalman Filter variants, for noise reduction, sensor redundancy and failure detection.

3.2. Method used for the analysis and selection of relevant DF techniques

All partners in the project contributed to assembling a survey of methods for data fusion and merging of sensor data that represents the current state of the art and state of play in the field of robotics and autonomous systems. This survey included summaries of the strengths and weaknesses of these methods, and an estimate of the maturity of each method. The availability of open source implementations and the usefulness of each method as a part of a Data Fusion Processing Compound (DFPC) to accomplish the core CDFF functional requirements was also evaluated. To select the most appropriate and effective set of algorithms, test implementations will be created from existing code where available, and from scratch where there is no available code. Test scenarios will create data for each type of algorithm. The most suitable methods for each application will be selected for C++ implementation as a data fusion node. In the ideal case, one or two algorithms will be included as the primary fusion in each category and others can be added optionally.

3.3. Resulting (preliminary) selection of DF

techniques/algorithms

Table 2: Preliminary selection of data fusion algorithms for implementation

Category of Algorithm	**Selected Algorithms**	**Optional Algorithms**
Optical Feature Detection	Hough Transform; Harris Detector; ORB Descriptor	SIFT; SURF
Point Cloud Feature Detection	Harris 3D; SHOT Descriptor	PFH/FPFH; SIFT 3D; SURF 3D
Recognition & Registration	ICP; RANSAC	Levenberg-Marquardt
General Data Association	K-Nearest Neighbours; Linear Classifier; Bayesian Classifier	Dense Registration
Non-Probabilistic State Estimation	Flow-Based Estimation; Fuzzy Logic; Dempster-Shafer	Dezert-Smarandache
Probabilistic State Estimation	Extended Kalman Filter; Unscented Kalman Filter	Particle Filter/SMC
Data Filtering and Pre-Processing	FFT Band Filters Variance Filter; Decimation; Normalization	-
Outlier Removal Methods	Interquart. Range; Mahalanobis Dist.; One Class SVM	GMMs; K-Means; Min. Vol. Ellipsoid

While the final list of data fusion algorithms to be included in InFuse will be defined after precise analyses and trade-offs, a preliminary set of algorithms has been defined that includes the most likely candidates to be selected. Tab. 2 shows this list of algorithms, with the highest-priority algorithms for implementation shown as "selected" and others that may be implemented if needed as "optional".

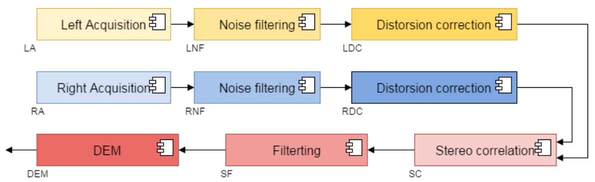

Figure 4. Assembly of data fusion node in a data fusion processing compound, here the production of a DEM.

These data fusion algorithms will finally be assembled to build a DFPC, as illustrated in Fig. 4 for the use case of DEM production. One of InFuse objectives is to ease this integration operation on development and space platforms.

4. CDFF ARCHITECTURE

At the core of the CDFF architecture lie the Data Fusion Nodes (DFN) - core libraries each one performing a specific subtask. These are connected via a Common Interface forming Data Fusion Processing Compounds DFPCs. The DFPCs are regulated by the main management component, the Orchestrator, which is also in charge of handling any external data product requests -sent by the Autonomy module-. The last main component is the Data Product Management Tool which provides data products storage and retrieval services. These services can be requested either by the DFNs or by the Orchestrator.

4.1. Main components and terminology

CDFF-Core

The CDFF-Core provides the "core" data fusion set of state of the art algorithms and techniques currently in use and applicable to the Planetary and Orbital RIs within the scope of Infuse, as a collection of ready-to-use libraries or packages. These libraries represent the CDFF's processing methods necessary to fuse sensory data. The CDFF-Core is to be implemented in a modular fashion with a common interface to allow high flexibility in configuration as well as in operation as a distributed system on multiple platforms. The core libraries are to be deployed on the target system (robotic platform) as well as on the designer's environment.

Examples of core libraries include low-level functions such as feature detection, registration and recognition, data association, state estimation, outlier removal, and filtering which can be assembled into building environmental representations, achieving 3D object reconstruction or SLAM. The preliminary set of algorithms implementable in core libraries is shown in Tab. 2.

CDFF-Support

The CDFF-Support provides the necessary tools for the instantiation and execution of Data Fusion Processing Compounds. Under the CDFF-Support functionalities can be found the DFPC, the Orchestrator, and the Data Product Management Tool (DPM) as shown in Fig. 5.

1. The **Data Fusion Processing Compound** (DFPC) is a combination of several DFN designed to provide a certain data product (e.g. pose estimation, map...). The DFPC defines the connections between input and output ports of different DFNs.
2. The **Orchestrator** is the component that deals with

the activation and deactivation of DFPCs. It is also the component that receives and answers the requests from the Autonomy Module – Operational Grant (OG2). The selection of one or another DFPC is done based on availability and quality of data sources and on the data product required.

3. The **Data Product Management** (DPM) Tool acts as a long-term memory for the data fusion products generated by the DFPCs to be used by OG2 or other DFPCs.

Along with the CDFF-Core entities, all components of CDFF-Support will be deployed in the Target System. They will also be available for programming and testing in the Developer Environment.

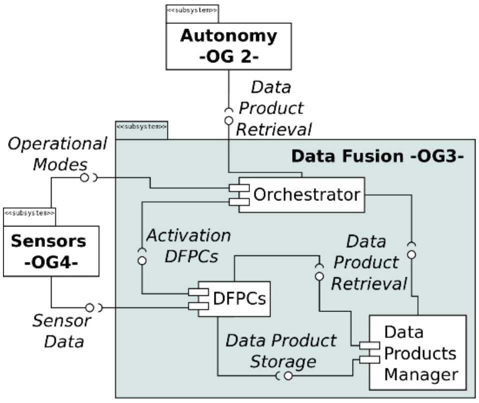

Figure 5. Interaction among CDFF-Support components and interfaces with OG2 and OG4.

4.2. Technical Baseline

InFuse will be designed and developed primarily as the CDFF for space robotics. This implies that the software solutions should be compatible or easily portable to space related (i) computing architectures (SPARC-Leon II/III) (ii) real time operating systems (RTEMS) (iii) programming methodologies and paradigms (ECSS standards). These challenges impose constraints during the design and in development such as following specific coding standards, static and fixed memory allocations, limited usage of OOPS… in order to facilitate the development of such data fusion solutions in an affordable manner. InFuse has application in two separated environments: The Target System and the Developers Environment. The first one is controlled by the RCOS and the software restrictions above mentioned apply, while the second environment will not be deployed in the robotic system and thus can make use of less limited software and hardware resources for the development process.

The CDFF-Core and CDFF-Support components will be implemented in C and C++. While on the Developer's Environment side, Python bindings will be utilized to expose the C/C++ interface of libraries that need to be tested independently. By using Python Pandas or any similar utility that can replay recorded or logged sensor data, any CDFF-Core library can be tested independently, and evaluated for performance and quality of fused outputs. Note that this approach facilitates RCOS independence, as the Developers Environment is independent of the Target System.

Environment Representation (EnviRe) is a package for representing arbitrary information on the environment for robotics [6]. The purpose is to have a common way of holding any information of the environment and how these information pieces relate to each other in a consistent representation across different applications. In the context of InFuse, the EnviRe graph will be used as symbolic representation and structuring of the spatio-temporal information, as well as to facilitate the attribution of metadata. EnviRe provides a visualizer that will be used to analyse data in the Developer's Environment.

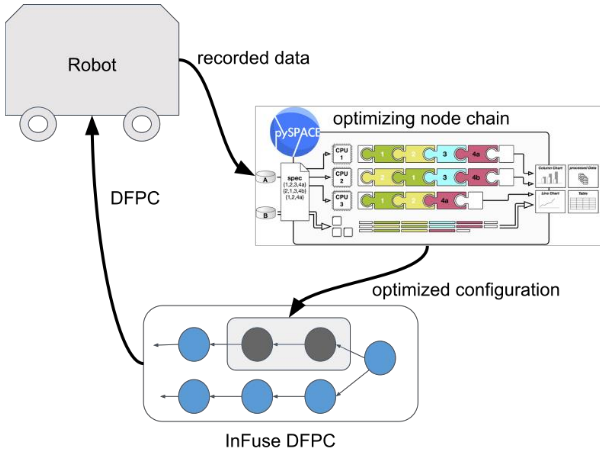

Figure 6: Development flow with pySPACE.

The framework pySPACE [5] will be available to evaluate and compare signal processing and machine learning algorithms. The main purpose of pySPACE is on comparing methods and configurations in terms of performance before the designer of a DFPC decides to implement or integrate them in a DFPC. pySPACE has the concept of node chains. These can be either one DFN or a sequence of multiple DFNs. Individual nodes of the pySPACE chain can be deployed as DFNs on the target system (Fig. 6).

4.3. Creating and Deploying an InFuse Instance

As mentioned above, the process of generating a Reference Implementation (RI) data fusion solution solutions consist of two stages each one taking place in a different environment as shown in Fig. 7: (1) development and initial evaluation on the Developer's Environment and (2) deployment and testing on the Target System. These two stages can be reiterated: after evaluating the initial solution on the Target System, the generated logs can be used in the Developer's Environment to analyse and improve the solution. On the developer's environment, a set of utilities for design and evaluation are available. The developed perception solution is designed to be easy to integrate in any Target System, independently of the RCOS and the specific hardware.

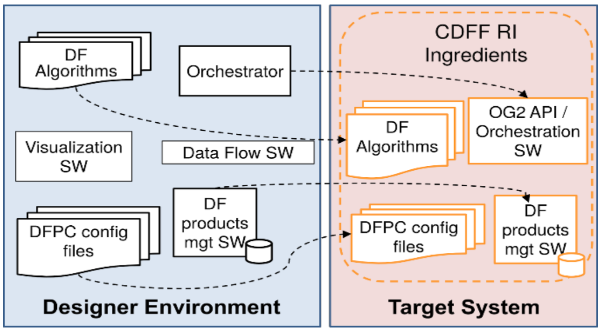

Figure 7. Migration of software components from the developer's environment a target RCOS

CDFF-Dev

The CDFF-Dev provides the tools to develop, test, visualize, and perform analysis on data fusion products. None of the components of the CDFF-Dev are deployed on the target system.

1. The CDFF-Dev **Common Interface (Python level)** are python bindings provided for the DFN common interface.
2. The **Middleware Facilitator** provides the CDFF the capability to partially convert a DFPC from the designer's environment in the corresponding DFPC on the target RCOS.
3. The **Logs and Data Flow Management** allows injecting logged data to a data fusion process in a chronological fashion. It is envisioned to allow loading of log data produced by different RCOS. The output of the nodes in the DFPC are also stored in logs which are used as input for posterior nodes in the processing compound.
4. The **Data Analysis and Performance Tools** are comprised of statistical analysis tools and graphical representations necessary for comparison of data fusion products resulting from different processing chains. The goal is to provide the designer methods to assess the quality of the data products and of the DFPC.
5. The **Visualizer** is responsible for presenting graphical representations of the different data products (e.g. 2D/3D plots, maps, camera images).
6. The **Calibration Tools** provide a framework for automatic (re-)calibration by using a manipulator. The goal is to automatically assess the quality of both intrinsic and extrinsic/hand-eye calibration and trigger an automatic calibration if needed [9].

During the design of the architecture other features were identified with potential interest, which could be implemented in the future. For instance, the **Data Fusion Processing Compound Configurator** that enables the designer to define intuitively, DFPCs. CDFF-Dev could provide in the future monitoring of the execution time and memory consumption of individual components in order facilitate the design of an embedded DFPC. CDFF Software components are classified in Tab. 3.

Table 3. Classification of CDFF Software components

CDFF-Components	**Subsystem**	**On Target System**
Core Libraries	CDFF-Core	Yes
Common Interface (C++ Level)	CDFF-Core	Yes
Data Fusion Processing Compound	CDFF-Support	Yes
Orchestrator	CDFF-Support	Yes
Data Product Management Tool	CDFF-Support	Yes
Common Interface (Python Level)	CDFF-Dev	No
Middleware Facilitator	CDFF-Dev	No
Logs and Data Flow Management	CDFF-Dev	No
Visualizer	CDFF-Dev	No
Data Analysis and Performance Tools	CDFF-Dev	No
Calibration Tools	CDFF-Dev	No

5. CDFF RELEASE AND EXPLOITATION PLAN

5.1. Open Source Strategy

On reaching the end of the project, a final version of the CDFF will be made publicly available as open source. Before its release, the quality of the code and the documentation will be reviewed to facilitate its use in future projects. A Git server will be used to allocate the final software. A second package will be generated as a legacy for upcoming Space Robotics Challenge projects with potentially some license restricted subparts which will be worked out under the Post-Project Follow up guidance.

5.2. Exploitation Perspectives

InFuse main outputs are of three sorts: software components, hardware support and contribution to the robotics community as summarized in Fig. 8 and Fig. 9. These outputs offer several exploitation perspectives targeting the robotics community, either for space or civilian applications. Exploitation perspectives will be driven by the open source and freeware releases of the software. A commercial support will also be provided to help companies to use or integrate CDFF components in their products with an appropriate licence. Regarding space applications, exploitation activities will first of all take place in the future projects of the SRC Space Robotics. CDFF components will be used and extended to implement targeted scenarios, and address additional scenarios such as precision landing and autonomous navigation on comets. CDFF components also offer the opportunity to develop new "ready to use" space products. Regarding civilian applications, exploitation activities will be driven by the technological transfer to answer the needs of the market. It includes for instance self-localisation for autonomous vehicles (aerial, terrestrial, underwater, humanoid) or high-precision pose estimation and tracking, e.g. for industrial manipulation tasks.

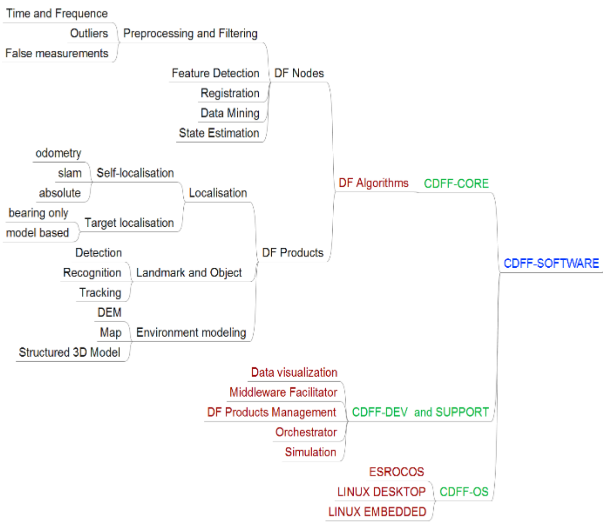

Figure 8. Software components developed by InFuse.

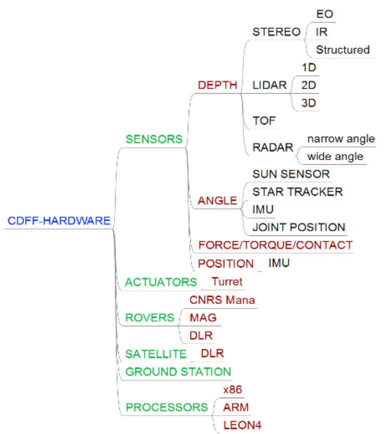

Figure 9. Hardware supported by InFuse (partial support through data types or full support with drivers).

6. CONCLUSION

InFuse aims to provide a comprehensive, modular data fusion system for future space robotic missions. InFuse will explore optimization and perception techniques to estimate location and model the surroundings of a robotic platform in order to support the planning and execution. In addition, InFuse aims to increase the overall performances of the offered DFPCs by providing tools to perform fast prototyping with core data fusion libraries before their final integration on the target platform. Ideally, this approach will bring improvements in the parameterisation of DFNs and will enable a rapid and probably an autonomous reconfiguration of DFPCs.

7. ACKNOWLEDGEMENTS

InFuse is funded under the European Commission Horizon 2020 Space Strategic Research Clusters - Operational Grants, grant number 730014.

REFERENCES

1. Ofir, C., Yael, E. (2008). A sensor fusion framework for online sensor and algorithm selection. RAS, Volume 56, Issue 9, Pages 762-776.

2. Bajcsy, R. (1988). Active perception. Proceedings of the IEEE, 76(8), 966-1005

3. European Space Robotics Control and Operating System (ESROCOS) http://www.h2020-esrocos.eu/

4. Gianfranco, V., Daniel, N., Michel D., Raffele, M., Javier R., Daniel J. (2015). D3.1-Compendium of SRC activities. EC H2020 SRC in Space Robotics Technologies.

5. Krell, M. M., Straube, S., Seeland, A., Wöhrle, H., Teiwes, J., Kirchner, F. (2013). pySPACE - a signal processing and classification environment in Python. In Frontiers in Neuroinformatics 7(40).

6. Carrió, J.H., Arnold, S., Böckmann, A., Born, A., Domínguez, R., Kirchner, F. (2016). EnviRe - Environment Representation for Long-term Autonomy. In AI for Long-term Autonomy Workshop of the Int. Conf. on Robotics and Automation (ICRA).

7. Rock: The Robot Construction Kit (http://rock-robotics.org).

8. Sylvain, J., Jan A. (2011). Robot development: from components to systems. 6th Nat. Conference on Control Architectures of Robots, Grenoble, France. 15 p., Inria.

9. Nissler, C., Marton, Z. C. (2017). Robot-to-Camera Calibration: A Generic Approach Using 6D Detections. First IEEE International Conference on Robotic Computing *(IRC)*, Taiwan, pp. 299-302.

10. Ron, L., Kaichang D., Jue, W., Sanchit, A., Larry, M., Andrew, H., Reg, W. (2005). Incremental bundle adjustment techniques using networked overhead and ground imagery for long range autonomous mars rover localization. In Proceedings of ISAIRAS.

11. Maurette, M. (2003). Mars rover autonomous navigation. Autonomous Robots, 14(2-3):199-208.

12. Souvannavong, F., Lemarechal, C., Rastel, L., Maurette, M. (2010). Vision-Based Motion Estimation for the ExoMars Rover, I-SAIRAS

13. Issa, A., Nesnas, D., Max, B., Richard, Madison. (2004). Visual Target Tracking for Rover-based Planetary Exploration, Proceedings of the IEEE Aerospace Conference.

14. Nassir, W.O., Giorgio, P., Quirin, M., Anastasia, T. (2015), Vision-based localization for on-orbit servicing of a partially cooperative satellite, Acta Astronautica, Vol. 117.