Deliverable Reference	:	D4.1
Title	:	Technical trade-offs analysis
Confidentiality Level	:	PU
Lead Partner	:	CNRS
Abstract	:	This deliverable presents the choices of the data processing and data fusion functions that will be developed within the InFuse CDFF.
EC Grant N°	:	730014
Project Officer EC	:	Christos Ampatzis (REA)

InFuse is co-funded by the Horizon 2020
Framework programme of the European Union

D4.1: Technical trade-offs analysis

DOCUMENT APPROVAL SHEET			
	Name	**Organization**	**Date**
Prepared and cross-reviewed by:	Simon Lacroix Sandeep Kottath Fabrice Souvannavong Vincent Bissonnette Zoltan Marton Michal Smisek Mark Post Xiu Yan Shashank Govindaraj	CNRS-LAAS CNRS-LAAS MAG MAG DLR DLR USTRATH USTRATH SPACEAPPS	28/07/2017

DOCUMENT CHANGE RECORD

Version	Date	Author	Changed sections	Reason for change
0.1	09/03/2017	Simon Lacroix	All	Define the document structure
0.2	09/05/2017	Fabrice Souvannavong Vincent Bissonnette Michal Smisek	2	Orbital RI scenario
0.3	16/05/2017	Simon Lacroix	All	Refactored the document structure
0.8	22/06/2017	Mark Post, Zoltan Marton, Simon Lacroix	All	Filled all the document sections
0.9	23/06/2017	All	All	Document review
1.0	28/07/2017	All	All	Post-PDR revision
1.1	14/09/2018	All	All	Modifications, to answer the RIDs

Using InFuse Preliminary Design deliverables in Perspective of the H2020 Space Robotics Technologies SRC Call 2

Overview of the Preliminary Design WP deliverables' content and purpose/logics within InFuse

D4.1 (this deliverable): The goal of D4.1 is to present the choices of the data processing and data fusion functions that will be developed within InFuse.
After an analysis of operational scenarios which exhibit the required InFuse outputs, the list of these outputs ("data products") is defined. For each data product, a brief analysis of possible solutions is made, and a solution is selected. Each solution is defined by a "Data Fusion Process Compound" (DFPC), and is described with a coarse level of precision. The refinement of the DFPCs into a series of elementary "Data Fusion Nodes" (DFNs, i.e. elementary data processing functions), and the definition of the organisation of the DFNs to compose a DFPC is matter for further work for the InFuse project.
The DFPCs are defined for both the planetary and orbital reference implementations. While both implementations mostly require different DFPCs, many DFNs will be shared between the two targeted contexts.
The document also presents a sets of operational scenarios for each reference implementation. They have the purpose of putting in context the proposed DFPCs.

D4.2: This deliverable focuses on the one hand on software architecture considerations, with the preliminary specification of all the key components that the InFuse CDFF consists of, and on the other hand on the identification of all relevant interfaces, both internal to InFuse and the external ones with respect to other OGs (OG2-ERGO and OG4-I3DS) in particular.
Note that the architecture and ICD material introduced in this document are essentially application independent – application specific considerations will be introduced at a later stage, during the detailed design of the InFuse CDFF (each of the orbital track and plenatery track will then be tailored, accordingly).
This deliverable will serve as a starting point for the detailed design work, in the following work package.

D4.3: The purpose of this document is to define the strategy and overall approach for the testing activities to be carried out internally (OG3), so that to ensure that the developed software is sound while meeting the requirements expressed in the earlier phases of the project. The content is orbital / planetary independent: it is not in the scope of this deliverable to specify the testing approach with facilities and EGSEs specific to either the orbital or planetary scenarios. These aspects will instead be addressed in the detailed design activities that are just starting as this document is being written, and will be released by the CDR milestone.

Relation between WP4 deliverable's content and other "common building block" OGs

The scope of OG3-InFuse lies essentially between OG4-I3DS, that produces raw sensor data, and OG2-ERGO, which controls all the rover activities. The interfaces with these two OGs are defined in the deliverable D4.2:

- The interfaces with OG4-I3DS are defined by sensor data types
- The interfaces with OG2-ERGO are defined on the one hand by the types of data products (mainly terrain maps and localization information related to the rover and the terrain, and to the chaser and target satellites), and on the other hand by requests made by OG2-ERGO to OG-InFuse.

OG2-ERGO is interfaced with OG3-InFuse at the granularity of the DFPCs: OG2-ERGO has no view of the inner mechanisms of OG3-InFuse that assembles DFNs into a DFPC. An OG2-ERGO request triggers the activation of a DFPC (which can either be synchronous or not), along with given parameters, and the DFPC returns the requested data products with an execution report. Within a DFPC, the DFNs are assembled, sequenced and triggered via pre-defined scripts, which are configured according to the parameters associated to the OG2-ERGO requests.

The definition of the DFPCs provided in D4.1 is generic, and makes no hypothesis regarding the integration middleware within which they will be developed. The way the developed functions will be integrated within the OG1-ESROCOS framework (and potentially other target middlewares) is depicted in the document D4.2.

Finally the content of D4.3 is largely centered on InFuse internal testing and validation activities – in that context, "integrated test plans" deal with the joint testing of several sub-parts of the InFuse framework, not InFuse and other OGs. Still, several components of the InFuse CDFF have interfaces with other OGs – mainly OG2-ERGO and OG4-I3DS. For these ones, it is foreseen in the test plan to develop specific components as placeholders of ERGO and I3DS, exposing the interfaces that are assumed to be the ones with which InFuse should integrate in the upcoming Space Robotics SRC projects. We call them M-OG2 and M-OG4 (M standing for Mock). Their purpose again is only to make it possible, internally, to carry out end-to-end tests and ensuring the soundness of the CDFF interfaces.

Applicability to the H2020 Space Robotics Technologies SRC upcoming calls (i.e. OG7 to OG11a/b).

Environment perception, be it to model the environment or to localize the controlled robotic platform within this environment, is at the core of autonomous operations, and is therefore required in all the applications defined by the upcoming calls.

For each of the future Operational Grants, a set of relevant DFPCs identified in D4.1 document is identified below as having particular relevance (note however that this list is based on very preliminary, and still limited knowledge of the content and scope of each upcoming OG).

OG7: (Orbital Support Services): Long/Mid/Close-range Tracking and Detection, 3D Target reconstruction, Point Cloud based Localisation
OG8: (Modular Robotized Assembly): Mid/Close-range Tracking and Detection, Point Cloud based Localisation
OG9: (Satellite Re-configuration): Mid/Close-range Tracking, Point Cloud based Localisation
OG10: (Advanced Autonomy): DEM Building + Soil Type Map, and all the rover localisation DFPCs: Visual Odometry, Visual/LIDAR based SLAM, Scientific Area Localisation, Visual Map-based Localisation, Absolute Localisation.
OG11a: (Advanced Mobility): DEM Building + Soil Type Map and Visual Odometry for extreme terrain mobility, plus all the localisation DFPC for the coordination of multiple platforms.
OG11b: (Robotized Construction): The required DFPCs for this OG are certainly similar to some of the ones required for the orbital OGs: Long/Mid/Close-range Tracking and Detection, 3D Target reconstruction, Point Cloud based Localisation – though in planetary context. DEM building and rover localization DFPCs remain relevant.

Deliverable D4.2 provides the latest baseline about the InFuse CDFF architecture and ICD. In perspective of the next OGs, it is essential to understand the proposed architecture and mechanisms to handle data, and the proposed approach to generate middleware specific reference implementations from the vanilla (middleware independent) CDFF environment.

In perspective of the upcoming OGs, D4.3 has limited relevance as its purpose is essentially to define the InFuse internal strategy to test the various components of the CDFF. It is therefore of little use for what concerns the preparation of bids for the next call, and similarly would have limited relevance for the future implementation of the next OGs.

Table of contents

List of figures

List of tables

1.1 Introduction

1.1 Purpose and scope

The goal of the document is to present the choices of the data processing and data fusion functions that will be developed within InFuse. The methodology that lead to the choices is the following:

- Assess the list of data products the CDFF is meant to deliver. This is done by defining and analysing operational scenarios, which exhibit the required InFuse outputs.
- Define a baseline solution for each data product. A brief analysis of possible solutions is made, and a selection of the set of data processing and fusion functions is proposed. The selection is made (more or less explicitly) according to a series of criteria defined in in section 4.3.

1.2 Document structure

The document is essentially organized in two parts:

- Section 2 "Operational scenarios" first assesses the list of InFuse data products, on the basis of operational scenarios briefly defined for this purpose. The section ends with a complete list of data products to be delivered.
- The core of the document is section 3 "Baseline solutions", which defines the DFPCs (and associated DFNs) that generate the data products. This definition is made following (more or less explicitly) a functional trade-off analysis of the possible solutions.

1.3 Applicable documents

[Grant_Agreement] InFuse Grant Agreement (CO document)
[Consortium_Agreement] Infuse Consortium Agreement (CO document)

1.4 Reference documents

[Compendium] Peraspera, "D3.1 Compendium of SRC activities (for call 1)", 05/01/2015
[InFuse_D3.1] InFuse Consortium, "Technological Review", Jan 2017 (PU document, available on demand)
[InFuse_D3.2] Infuse Consortium, "System Requirements", Jan 2017 (PU document, available on demand)
[InFuse_D3.3] InFuse Consortium, "Early Architecture and ICD", Jan 2017 (PU document, available on demand)
[ERGO_D1_1] Ergo Consortium, "Technology review", Jan 2017
[ERGO_D1_2] Ergo Consortium, "System Requirements", Jan 2017
[FACILITATORS-D1.2] System Requirements Document, Jan 2017

1.5 Acronyms

CDFF: Common Data Fusion Framework
DEM: Digital Elevation Map
DFPC: Data Fusion Processing Compound
DFN: Data Fusion Node
DTM: Digital Terrain Map
ICP: Iterative Closest Point
SLAM: Simultaneous Localization And Mapping
RI: Reference Implementation

2 Operational scenarios

Elements of scenarios can be found in the SRC Compendium of activities [Compendium]. For the planetary track, the following operational requirements are defined:

- Depart from a landing spacecraft,
- Reach towards another planetary asset several kilometers away,
- Perform autonomous science while traversing,
- Rendezvous with another planetary asset.

For the orbital track, the defined operational requirements are:

- Rendez-vous and capturing, which encompasses close perception, capturing and docking
- Exchange of a payload module between the chaser and the target.

From these operational requirements, the Infuse Consortium has proposed five scenarios in the systems requirements document [InFuse_D3.2], aiming at allowing the verification that the identified requirements are suitably implemented.
Besides, two reference scenarios for planetary exploration and in-orbit servicing have been defined by the OG2-ERGO consortium in the system requirements document [ERGO_D1.2]. These two complete scenarios integrate all the required functionalities for the two reference implementations, and depict the products that OG2-ERGO requires from InFuse (note that other elements of scenarios are defined in [FACILITATORS_D1.2], mostly for validation purposes).

Here, we complement the description of the scenarios introduced in [InFuse_D3.2], aiming at enumerating all the data products that InFuse has to deliver. This list of data products is the basis upon which the technical trade-offs are made, in section 3, “Baseline solutions".

2.1 Planetary track

The planetary track reference implementation encompasses a large variety of functionalities. For the ease of the analysis, instead of defining a general scenario that requires all these functionalities, we have chosen in [InFuse_D3.2] to define 5 different scenarios, each exhibiting a specific set of functionalities. The first scenario (“Long traverse”) is naturally the most important one, the four others defining extensions of this one:

- Long traverse: the objective of the mission for the rover is to autonomously reach a target located about 1km away, defined by its absolute coordinates;
- Autonomous science: while navigating, the rover processes acquired data so as to detect and localize potential scientific targets;
- Rendez-vous: the goal is here to precisely position the rover with respect to an existing asset (*e.g.* the lander);

- Getting back: this consists in getting back to the initial position after a long range traverse;
- Cooperation: this scenario involves the use of a second mobile robot (*e.g.* a scout-rover, or a UAV) to assist the rover during long traverses.

2.1.1 Long traverse

This scenario is the base one upon which all the others are defined.

2.1.1.1 Mission definition

The objective of the mission for the rover is to autonomously reach a target located about 1km away, defined by its absolute coordinates.

The mission is defined by an path specified for the traverse, with constraints to satisfy:

- Localisation constraints: each part of the path (or each waypoint to reach) is associated to a requirement on the precision of the localisation;
- Time constraints: the path is specified with bounds on time to satisfy

The path is specified either by a corridor or a series of waypoints.

2.1.1.2 Mission preparation

The mission is defined by an operator on the ground (possibly assisted with planning tools), who considers a series of constraints to satisfy and criteria to optimize. For this purpose, various information have been considered, such as terrain traversability, sun illumination, possibility to absolutely localize the robot. These information come from orbiter data and a priori knowledge and models, but can also encompass information gathered by the rover in previous missions.

2.1.1.3 Mission execution

The rover follows a nominal navigation cycle, that is sequences of short term path planning and trajectory following phases.

- Short term path planning. This requires the "Fused Rover Map" (Digital Elevation Map structure);
- Trajectory following:
  - Localisation at 10Hz to ensure satisfactory path following;
  - Hazard avoidance. This requires the production of the "Rover Map".

In addition to the nominal navigation cycle, the following phases are also planned:

- Upon request from OG2, or automatically triggered according to some criteria (e.g. precision of the current global position estimate, length of the executed path): absolute localization is made avaiable;
- End of the mission: If specified in the mission, build a Fused Total Map. This map may serve as initial information for further mission planning, *e.g.* for the getting back scenario.

2.1.1.4 Associated data products

The data products associated to this scenario are the following:

- Three kinds of Digital Elevation Maps, that differ from the amount of fused data, their spatial extent, their reference frame, and the frequency of their update:
  - ”Rover Map”;
  - “Fused Rover Map”;
  - “Fused Total Map”.
- Two localization services, that differ for the frame in which they estimate the rover pose and the frequency at which the output pose estimates:
  - @ 10Hz when driving;
  - Absolute localization (using the orbiter map).

Notes:

- each data that has to be integrated within the DTM must be precisely localized, so that the produced DTM is spatially consistent.
- If required by OG2-ERGO, in order to help the production of Navigation Maps, the DTM can be complemented with a Soil Type Map (mostly derived from luminance information).
- the “getting back” scenario (section 2.1.4) introduces the need to build a specific data structure (a “localization map”) during the execution of the long range traverse. This data structure (which can be defined upon the built DTMs) is not exported to OG2-ERGO.

Section 2.3.2 discusses these points.

2.1.2 Autonomous science exploration

Rather than an actual operational self-contained scenario, here “Autonomous science exploration” is considered as an additional functionality that can be triggered during the “Long traverse” scenario.

2.1.2.1 Mission definition

The mission consists in activating a process of scientifically relevant target detections, in given areas during the achievement of an autonomous traverse scenario.

2.1.2.2 Mission preparation

The details of the mission are defined by an operator (possibly assisted by mission planning support tools), along with the definition of the long traverse scenario. On the basis of similar information that are used to prepare the long traverse, the operator defines:

- The areas within which the scientific target detection process must be activated;
- The specification of the kind of scientific target to be detected.

The specification of the kind of target to be detected are limited to the sensorry information available and corresponds to the selection of the sensor(s) and detection process(es) to activate, because there may co-exist a variety of detection processes within the system. Note that the data acquisition may require pointing the sensor(s) or making a panorama: the priori planning and on-line adaptation of these options, if retained, does not pertain to OG3-InFuse, but to OG2-ERGO.

2.1.2.3 Mission execution

The mission execution simply consists in activating the selected target detection process and corresponding modules in the defined corresponding areas.

2.1.2.4 Associated data products

- The single data product associated to this scenario is the localization of the detected scientifically relevant target(s). A target is localized with respect to the rover, its spatial extent is provided (bounding box).

2.1.3 Rendezvous

2.1.3.1 Mission definition

The mission consists in reaching a precise position and orientation with respect to a man-made asset, e.g. the sample analysis module, for instance so that a sample can be transferred.

2.1.3.2 Mission preparation

The rover is supposed to be in the close vicinity to the asset. In the general case the terrain to traverse to achieve the Rendezvous with the asset is not known - yet it could have been previously modeled, and hence may be known, this does not change the gist of this scenario. The mission preparation consists in specifying a given position and orientation relatively to the asset to reach, as well as the trajectory the rover must follow to reach it. The preferred rendezvous feature on the asset will also need to be specified.

2.1.3.3 Mission execution

The rover executes the planned trajectory (using the basic nominal navigation cycle if the terrain is known). To insist on the specificity of this scenario, we considered that the rover is in a close range situation with respect to the asset to reach, and so that it must localize itself with respect to the asset, using specified features that are detected on the asset.

2.1.3.4 Associated data products

- The single data product associated to this scenario is the relative pose of the robot with respect to the asset to approach, produced at a frequency allowing fine trajectory control (10 Hz, to be confirmed).

Note: this localization is produced by a dedicated localisation service, which exploits features detected on the asset to reach. This service is analogous to the one required by the Orbital RI scenarios, and hence shares common processes.

2.1.4 Getting back

This scenario is the reverse traverse that can be defined after the execution of a long traverse mission, or any other trajectory formerly executed.

2.1.4.1 Mission definition

The objective of the mission for the rover is to autonomously execute rearward a trajectory that has been executed beforehand (*e.g.* after a long traverse, or after having fetched a sample or performed a scientific analysis). The benefit of this ability is to achieve traverses without resorting to path planning, nor absolute localization: in theory only the trajectory following process, supported by the knowledge of potential hazards identified previously in the Rover Maps, drives the rover, thus enabling faster motions (yet, for safety reason the hazard detection process can be required, thus requiring the production of Rover Maps).

The mission is defined by:

- The trajectory to follow;
- Constraints on the localisation precision defined along the trajectory;

2.1.4.2 Mission preparation

The mission is prepared on the basis of data collected during the forward traverse, upon which the trajectory to execute is defined, and with which localization can be ensured with the required accuracy during the execution of the trajectory. The later is encoded within a *Localization Map*, a specific data product that is not exported to OG2-ERGO.

Note: the trajectory to execute might be slightly different from the one that has been executed: it can indeed be replanned by OG2-ERGO using the Total Rover Map.

2.1.4.3 Mission execution

The execution of the mission consists in following the defined trajectory exploiting the Localisation Map and the Fused Total Map from the previous trip before going back. Possibly, a local Rover Map can be produced at a given spatial frequency.

2.1.4.4 Associated data products

The single data product associated to this scenario is:

- Localisation with respect to the “localization map”;

Note: Additionally, if required local Rover Maps can be produced and exported to OG2-ERGO at a given spatial frequency.

2.1.5 Cooperation

This scenario is the most complex one, and also the most prospective. Its definition remains quite open (it has not been depicted in any other OG deliverables), and instead of a complete scenario definition, we briefly depict some cooperation schemes, aiming at defining the associated InFuse products.
The consideration of such schemes is here mainly to assess that the actual definition and achievement of such scenarios can be supported by InFuse, without requiring to update the InFuse architecture or to develop brand new functionalities. The structure of this section is therefore less precise that for the other scenarios.

2.1.5.1 Perception and navigation functionalities for cooperation

Cooperative schemes can be defined to achieve the whole variety of missions that a single rover can be tasked with, such as exploring / mapping an area, long range traverse, or a declination of the getting back scenario in which one robot executes a trajectory, formerly executed by another. The cooperating robots may be heterogeneous, *e.g.* a UAV can act as a scout for a rover, and the cooperation schemes are of course not limited to two robots. Without going into further details, the functionalities related to perception (terrain modelling) and navigation (localization) that can be used to build cooperation scenarios are the following:

- Fusion of data gathered by the different robots into a Digital Elevation Map. Here the fusion can be made along a tight scheme (incorporating point-clouds), or a loose scheme (integrating DTMs independently produced by the robots with their own-acquired data). Note that the data / DTMs need to be registered beforehand (see below)
- Inter-robot localization. This can be achieved by two different means:

- ○ Direct localisation of one robot with respect to the other (when both robots are in sight);
- ○ Indirect localisation, by registering point-cloud data (of DTMs, or point-cloud data with a DTM). This can be achieved even when the robots are not in direct sight.

These inter-robot localization information are used to refine each robot position estimates.

2.1.5.2 Associated data products

The data products associated to cooperation scenarios are exactly the same than for the former scenarios: DTMs (Fused Rover and Total Rover Maps), and localization estimates at low frequency (akin to Absolute Localization). The way to generate these data products will not be further detailed, yet they will be slight derivatives from the DFNs and DFPCs retained for the basic scenarios.

2.2 Orbital track

Here, instead of associating a scenario to the various operational concepts, a single global scenario is depicted. Also, since the validation of the developments will require a dedicated setup (robot manipulators), the use of this setup is mentioned in the various scenario phases.

The scenarios in orbital track can be found in the SRC Compendium of activities [Compendium], where the following operational requirements are defined:

- Detection of a target spacecraft;
- Far-range rendezvous ;
- 3D mapping (3D model);
- Close-range approach.

2.2.1 Mission definition

The orbital track reference implementation consists in approaching a target spacecraft (mainly at close range), which can be simulated with a hardware in the loop simulator. The chaser spacecraft needs to identify, track and approach a target satellite to perform in orbital servicing operations under the following constraints:

- localization constraints: each part of the path is associated to a requirement on the precision of the localisation. Thus, the localization constraints are defined at various ranges, which may require bearing only tracking, 3D tracking and detection;
- range constraints: the localization is specified with bounds on range to satisfy approach requirements;

- time constraints: the range constraint is specified with bounds on time.

2.2.2 Mission preparation

With pre-conditions described in [InFuse_D3.2] such as preparation of dynamic test bench, calibration and pre-configured OBC with CDFF software, the target satellite will be first detected, knowing its geometric model and using calibrated cameras. The satellite model will be pre-processed offline to reduce computational complexity of the tracking software. In order to achieve the defined scenarios, robot manipulators in the dynamic test bench will be powered up along with all the supporting motion control system, real time simulator, monitoring and debugging system and the OBC running the CDFF software stack. The sensors are activated at the start of the test run and the CDFF software is started to perform a sanity check on the flow of sensor data to the CDFF software and forward it to a pre-defined set of data fusion methods that begins to acquire sensor data for processing.

2.2.3 Mission execution

The robot manipulators of the dynamic test bench perform a series of pre-defined trajectories and goes to a pre-defined starting position. The ground truth which is obtained from robot forward kinematics and if available by an additional tracker system provides the absolute pose of the spacecraft mockups.

The mockup of a chaser spacecraft will be positioned initially at the maximum possible distance from that of a target spacecraft. The illumination of the setup is activated and if applicable together with the multiple-axis free rotations of the mockup to replicate a freely floating satellite in orbit. The demonstration of the CDFF will begin by evaluating a baseline implementation of localization and target satellite state estimation data processing chains with respect to the target spacecraft. This task would involve obtaining a dynamic model of the target spacecraft using LIDAR complemented by stereo/optical cameras, detecting and classifying designated landmarks for grasping and stabilization. Relative localization of the target satellite or a designated target point provides inputs for planning trajectories of the robot manipulator.

The final phase of operation requires vision-aiding markers or a designated target structure for accurate pose estimation under simulated in-orbit illumination conditions. The localization chain can be switched to a slower rate but higher accuracy of output to enable a close and safe approach to the target. Once the two spacecrafts are close enough (< 1m), the manipulator onboard the servicer spacecraft will be commanded most likely in open loop wrt the CDFF software. The quantitative evaluation of the CDFF will be primarily offline on a recorded dataset obtained from the on-orbit servicing simulator to allow multiple parameterizations to be tested under the same conditions in reasonable time, and to minimize the effect of possible synchronization issues that may be insufficiently addressed by the hardware setup. However, the CDFF will be demonstrated for qualitative analysis as integrated on the OG6 platform. The data fusion processing chains will be evaluated based on the localization precision w.r.t to ground truth measurements.

2.2.4 Associated data products

Which state information is estimated depends on the range to the target and the same applies for the possible and required accuracy.
At far range, the state contains bearing only information, i.e. the position of the target relative to the servicer.
At mid- and close range the state of the target satellite consists of its position and orientation, i.e. the 6 DOF pose, relative to the servicer. By detecting the pose over a sequence of inputs it might be possible to determine the relative speed between the target and the servicer spacecraft. State estimation over periods of time assumes that no external forces are applied to the target (constant speed model). It is possible to determine whether sudden changes have occurred (e.g. a collision) by comparing the estimated state with the sensed state at each step and determining a residual.

The data products associated to this scenario are the following:

- Target satellite detection (far-range range and bearing localization);
- Target satellite state estimation (full localisation and speeds wrt. the chaser);
  - For the mid-range rendezvous;
  - For the close range approach;
- 3D reconstruction of the target.

2.3 Synthesis: list of InFuse Data products

The set of data products generated by InFuse and delivered to OG2-ERGO is split along the two planetary and orbital reference implementations.

The products delivered in the planetary track are:

- Digital Elevation Maps (DTMs):
  - Rover map;
  - Fused Rover Map;
  - Total Rover Map;

- Scientific target localisation;
- Localization;
  - At 10Hz while driving;
  - Absolute localization;
    - With respect to the orbital map (case of the long traverse scenario)
    - With respect to a localization map formerly built by the robot (case of the getting back scenario)
  - Relative localisation with respect to a man-made object.

The products delivered in the orbital track are:

- Target satellite detection (far-range range and bearing localization)
- Target satellite state estimation (relative localisation and speeds wrt. the chaser)
  - For the mid-range rendezvous
  - For the close-range approach
- 3D reconstruction of the target

Notes:

- This list associates a name for each of the data products, that are used throughout the rest of the document - yet these names may not match the ones defined in the ICD document, the final product naming convention will be adopted when this ICD document is finalized.
- Besides these data products exported to OG2-ERGO, various data products are built and maintained *within* InFuse for its own purposes (*e.g.* the Localisation Map for the getting back scenario). These private products, which are managed by the Data Product Manager, are introduced in Section 3 "Baseline solutions", along with the processes that generate and/or exploits them.

3 Baseline solutions

This section is organized according to the list of data products established in section 2.3: for each data product, a subsection analyses the main possibilities to define a DFPC that generates it, and selects a baseline solution that is retained for development and integration within the CDFF.

The structure of each subsection is the following:

1. Reminder of the definition of the data product (and associated variations) and of their use.
2. Definition of the DFPC that generates this data product, list of the DFNs that constitute it. This definition is made after a more or less explicit trade-off analysis, based on a very brief review of the principle of existing possibilities.

The analysis pertains to the DFNs and their assembly that constitutes a DFPC. A thorough trade-off analysis would require the evaluation of the following series of criteria:

- Performance: to what extent does the approach fulfill the need? (in some cases, the performance can be assessed by a quantitative criteria)
- Maturity: has the approach been successfully tested and validated in realistic conditions?
- Portability: is the approach compatible with the currently foreseen Leon target processors? Can the performance be assessed? (in both terms of quality and required CPU resources)
- Open-source availability: are there open-source implementations available?
- Expertise of the InFuse partners: is one (or more) partner(s) of the consortium acquainted to the approach and its implementation?

Notes:

- These criteria rather apply to DFNs, which are atomic well defined processes: they are harder to assess for DFPCs.
- The estimation of some of these criteria is rather qualitative: maturity of a solution and expertise of the partners are for instance hardly assessed by numbers.
- Other specific criteria not mentioned in this list may pertain to only a single data product or DFN (*e.g.* re-use by a DFPC of data products generated by other DFPCs). In such cases, they are explicitly mentioned, and this is in particular the case for the analysis for the absolute localization DFPC (section 4).
- The compatibility with respect to space-grade CPUs is of course very important. Yet, the choices of the DFNs will be very limited if one targets Leon CPUs for instance, and one can not exclude effective solutions without having assessed precisely their portability to such CPUs, which can only be done during the development phase.

3.1 Digital Elevation Maps

3.1.1 Definition of the data product

Digital Terrain Maps are the core maps of the rover environment, that can be used for a variety of purposes. The primary need they fulfill is for OG2-ERGO to plan short term paths, to detect obstacles while navigating, and to define navigation maps upon which long term itineraries can be planned. Additional uses can be locomotion monitoring, or to build Localization Maps used in the “Getting Back” scenario.

The core structure is a regular Cartesian grid to each cell of which elevations are assessed, built upon a point cloud. This representation can be complemented by information related to the soil (Soil Type Map), extracted from luminance information associated to the point cloud.

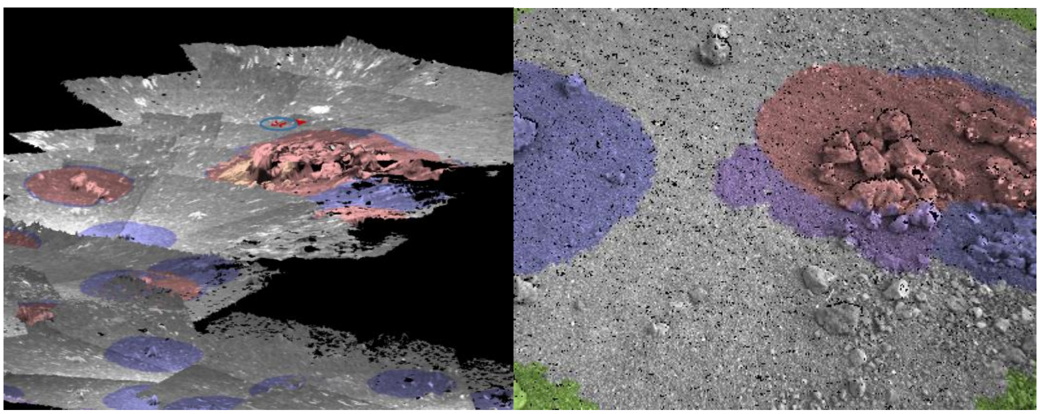

Figure 1: Illustration of Digital Terrain Maps built with a sequence of stereovision data. A luminance value has been associated to each cell (colors encode a notion of traversability, out of the scope of the DFPC)

3.1.2 Definition of the associated DFPC

Figure 2 shows the flow chart of the DFPC that generates the various Digital Terrain Maps. Two “threads” of processes are interleaved: the ones that generate the DTMs, and the ones that generate the Soil Type Map. Note that while DTMs can be produced without Soil Type Maps, Soil Type Maps can not be produced without DTMs: the production of these latter maps is an option of the production of the Rover Maps.

3.1.2.1 Building the Rover Maps (DTMs)

The overall process that builds a Rover Map takes as input a point cloud registered in the rover reference frame, either produced by stereovision or Lidar, and sets the elevations (and associated uncertainties if required) of the Rover Map. The function that sets the DTM elevations comes to a re-sampling of the point cloud along a regular Cartesian grid. Its complexity is linear in the size (number of points) of the point cloud.

The building of a Fused Rover Map is made by the integration of the Rover Map in the current Fused Rover Map, after having properly registered it in the Fused Rover Map frame. The complexity of this function is linear in the size of the Rover Map.

Notes:

- In case no Rover Map is required by OG2-ERGO, an alternate solution is to *directly update* the information in the current Fused Rover Map with the data encoded in the point cloud. This solution is nearly equivalent to fuse a Rover Map, and saves the computation required to build the (intermediate) Rover Map.
- Depending on the nature (quality) of the input point-cloud, some filtering processes may be applied prior to their integration, *e.g.* to remove outliers.

These processes require a fixed number of instructions and a fixed (small) memory footprint that are fully defined by the sizes of the DTM and point cloud data: they run within a known predictable bounded (and small) time, and can easily be ported on space-grade CPUs.

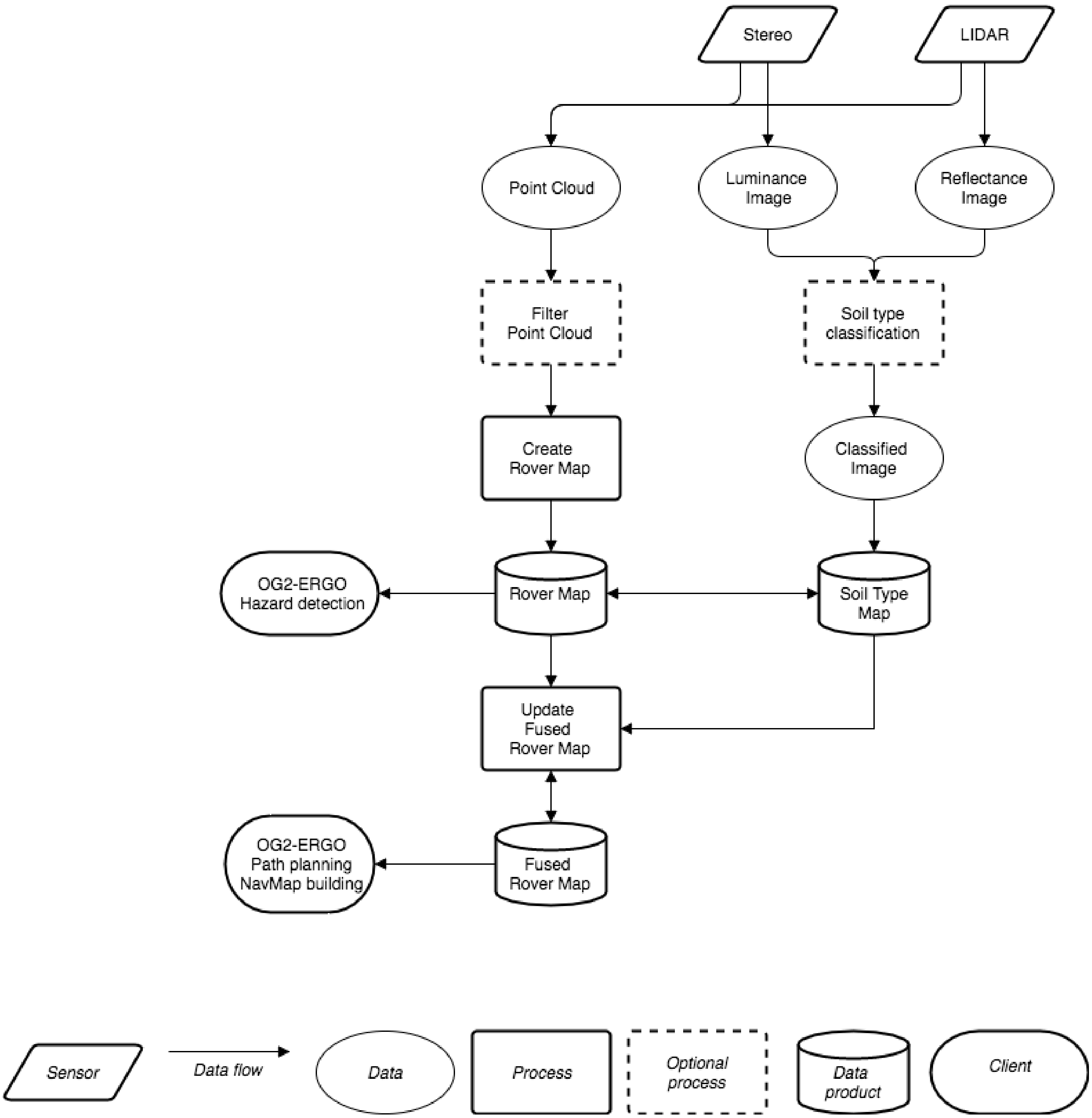

Figure 2: Flow chart of the processes that generate the Rover Maps (DTMs) and the Soil Type Map.

3.1.2.2 Building the Soil Type Maps

To produce the Soil Type Map, luminance information (in the case of stereovision) can be exploited. The literature abounds with classification processes that exploit texture / colorimetric attributes, the most classic ones (supervised classification schemes) requiring bounded processing time and memory, and hence being portable to space-grade CPUs.
The choice of the attributes to compute and the selection of the classification approach to apply (naive Bayes, adaboost, SVM, random trees…) will be made after a more detailed definition of the classes of terrain relevant for navigation will be assessed (*e.g.* sand, gravel, small rocks, rock outcrops…). The classification problem at hand is a rather easy one, and there is no need to resort to the latest state of the art contributions, which would both require mass amount of data and massive computation abilities. More relevant is the exploitation of the geometric information of the point clouds to compute additional attributes that will yield more precise soil types estimates, allowing in particular the use of prior knowledge that relate the soil geometry and type: the classic classification processes enable such an extension.

The situation is different regarding terrain classification using Lidar remittance/reflectance values, as it is a problem that has not been thoroughly studied (mainly due to the lack of Lidar delivering calibrated remittance information in the roboticist community, such Lidars have only been available recently). While on a varied earth scene this information has shown to allow the identification of *e.g.* vegetated vs. mineral soils, the benefits of the use of such information in a planetary context remains to be assessed. This will be made by evaluating the application of classic classification processes on data acquired on representative sites.

3.2 Scientific area localization

3.2.1 Definition of the data product

Scientific area localization consists in providing the target position with respect to the rover at the time of detection, as well as its spatial extent (bounding box).

3.2.2 Definition of the associated DFPCs

The detection of scientific targets is a visual process applied on camera images, which segments out image areas assessed as scientifically relevant. The localization of these image areas in the rover reference frame basically comes to estimating the position and geometric extent of the 3D point cloud associated to the pixels they encompass. This is achieved depending on the additional data acquired jointly with the camera image on which they are detected:

- If the image is one of a stereoscopic image pair, the extraction of the point cloud associated to the detected areas is simply achieved by extracting the associated 3D points in the point cloud produced by the stereovision process.
- If the image is acquired by a camera which is not of a stereoscopic bench (*e.g.* the TIR camera), the point cloud of the scene is produced by another sensor (stereovision or Lidar). Extracting the point cloud associated to the detected areas is simply achieved by reprojecting the image areas on the point cloud, using the inter-sensor 3D transform.

3.3 Rover Localization

3.3.1 Foreword: on localization in mobile robotics

Localization is a key functionality in mobile robotics, whatever the context is, as it is required to ensure the spatial consistency of the built environment models (the DTMs in the case of InFuse), to achieve the execution of the planned path (“localization at 10 Hz” in our case), ensure the monitoring of the missions executions (“absolute localization” and “map-based localization”, respectively for the long traverse and getting-back scenarios).

Localization is defined by the estimation of the robot pose (in 6D, 3 translations and 3 rotations) and the associated uncertainties. But a localization service is defined by additional concerns:

- The frequency of the localization estimate, which ranges from tens of Hertz to “from time to time” (*e.g.* once a day, or at the end of a mission)
- The frame in which the pose estimation is provided (local frame for a planned motion, global frame for a whole mission, or absolute “geo-referenced” frame),
- The precision of the localization estimate, defined by an uncertainty model (most often a probabilistic one, but alternate models can be used, such as the set-membership one),
- Integrity: is the pose guaranteed to lie within the uncertainties of the estimate? (either probabilistically in the case of probabilistic uncertainty model, or within bounds in the case of set-membership uncertainty model)
- Availability: is the localization service available everywhere and every time? (and if not, where and when?)

From a general point of view, the development of one (or of a set of) localization solution calls the integration of three kinds of processes:

- Signal processing, which is defined by the nature of the exploited data, and generates the inputs of the estimation process,
- Estimation process, which produces the localization information. Two families of processes are mainly exploited: Bayesian filtering (Kalman filter and its derivatives,

particle filters, ensemblist processes) and optimization processes (systematically non-linear),
- Management of the data structures and pose estimates produced over time.

In mobile robotics, localization solutions can be taxonomized in three families:

- Dead-reckoning: these solutions estimate the pose by integrating the estimation of elementary motions. They only require low level data processing, and hence can run at high frequencies. The position is estimated in a local frame (most often the start position before executing a path or a whole mission), and its uncertainty grows indefinitely with the length of the executed motions. Such solutions do not exhibit integrity (or only over short distances), but are nearly always available.
  - SLAM solutions: by estimating concurrently the robot pose and a specific map of the environment, such solutions require more processing time and hence run at a lower frequency than dead-reckoning solutions[1]. The position is estimated in a local frame (most often the start position before executing a path or a whole mission), and its uncertainty can be reduced when the robot traverses areas previously mapper. Such a solution does not exhibit a full integrity, and their availability inherits the specificities of the user sensors and the environment.
- Map-based solutions: these solutions estimate the robot pose by matching acquired data (or environment models built on purpose) with an a priori known map. They require the processing of exteroceptive data, and hence run at a lower frequency than dead-reckoning solutions. The position is estimated in the frame associated to the used map (and can hence be coined as “absolute”). Such solutions do not exhibit a full integrity, and their availability inherits the specificities of the used sensors, the environment and the associated map.

The definition of a wholesome approach to localization that would ensure full integrity and availability is still a matter for research (and one can actually doubt such an approach will ever be proposed), yet the state of art is now proposing a full corpus of formalisms and method to achieve localisation in robotics. A sound approach is to develop *a series of localization solutions*, adapted to given situations and requirements. Yet, the solutions should not be mutually exclusive, on the contrary they should be *integrated and flexibly selected*, so as to deliver high quality localisation in any defined circumstance: this is the approach retained for InFuse.

[1] This is actually not really true, as most SLAM solution integrate a dead reckoning solution, and hence produce pose estimates at high frequencies. Only estimates that exploit the built map are produced at lower frequencies.

3.3.2 Definition of the data products

The need for four localization services have been exhibited:

- Localization while driving: this is mainly to ensure the execution of the planned path, but also to ensure the spatial consistency of the built environment model. It can be provided by a dead-reckoning or a SLAM approach.
- Absolute localisation: this is required only from time to time, to ensure the proper execution of long traverses. This is typically a map-based localization approach, that uses an initial environment model derived from orbiter data.
- Localization while driving back: this localisation service is specific to the “driving back” scenario. It is map-based localisation service, which exploits a map and possibly the hazard information built during the former execution of a traverse (*e.g.* by a SLAM approach).
- Localization wrt. a man made asset: this is also a map-based localization service, the map being here a model of the known asset (one should rather refer to this localization service as a model-based localization).

3.3.3 Definition of the associated DFPCs

Following our ambition to deliver an integrated solution, a series of localization DFPCs have been defined. Figures 3 and 4 presents these DFPCs and their relations, which are briefly depicted in the following subsections defined for each data product.

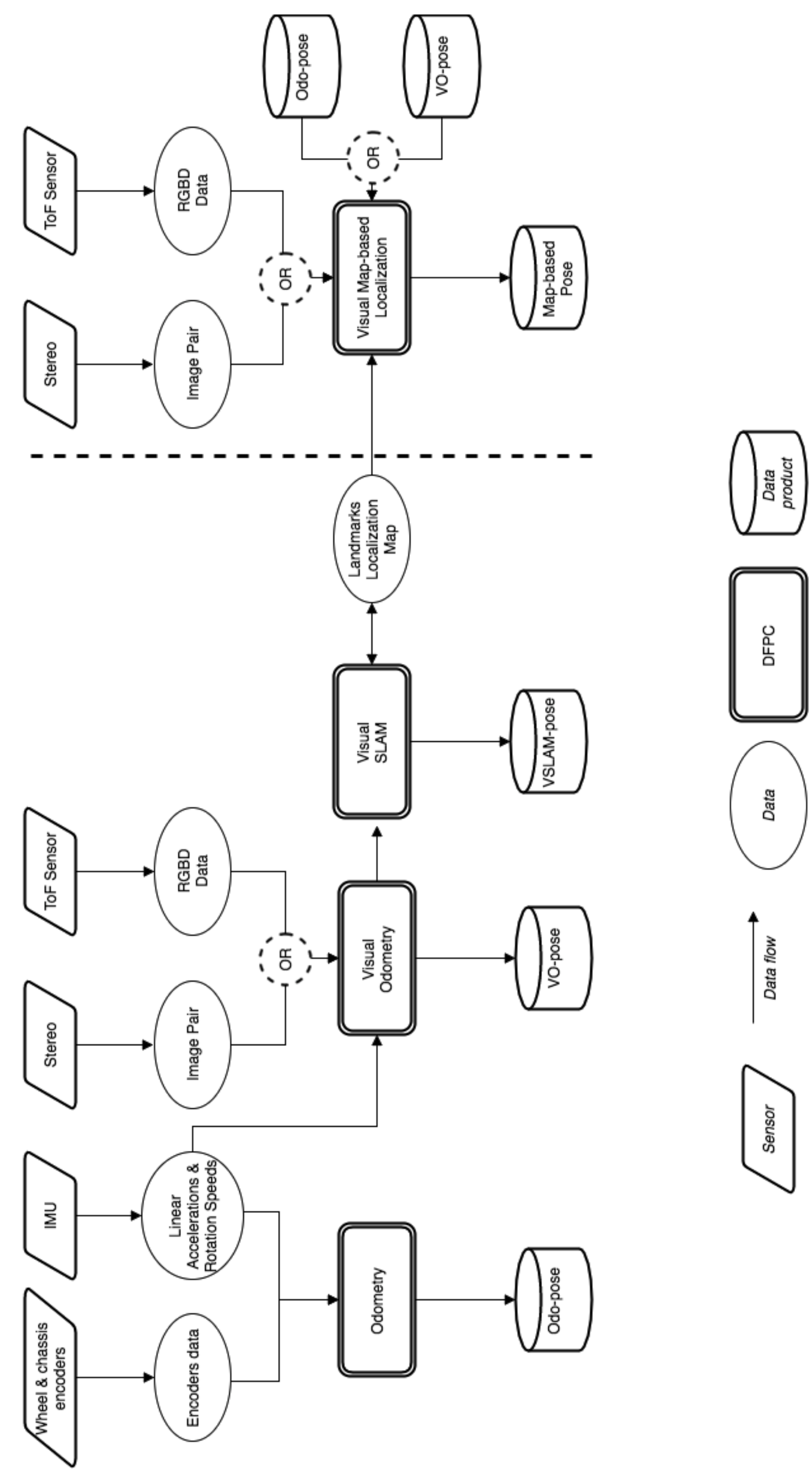

Figure 3: Overview of the localization DFPCs that exploit visual sensors

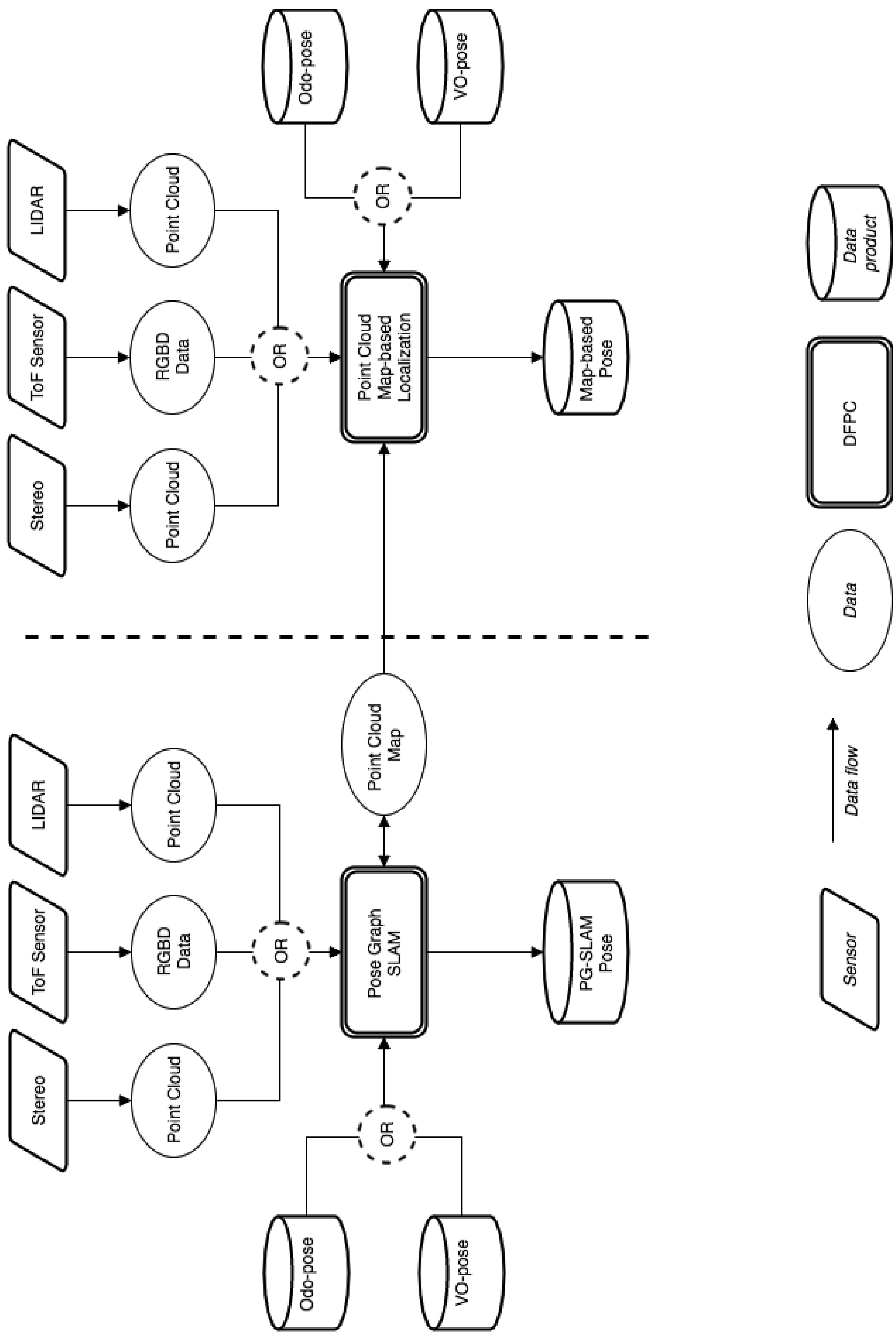

Figure 4: Overview of the localization DFPCs that exploit point cloud sensors

3.3.3.1 10Hz localization while driving

This is mainly ensured by two DFPCs: Odometry and Visual Odometry. Optionally, a visual SLAM or a Pose-Graph SLAM approach can be used, either to ensure more precise localization estimates or to produce a localization map (respectively a Landmark Map and a Point Cloud Map), that can be exploited during a “getting back” mission.

- Odometry simply processes all the wheels and chassis encoders. A special care should be taken on slippages detection, which can be ensured by monitoring the consistency of the various inputs.
- Visual odometry has shown to significantly enhance the precision over odometry (or even to detect odometry or locomotion faults). The InFuse approach to visual odometry is a classic one, that matches key points between two consecutive stereoscopic image pairs (this matching being possibly focused by inertial measurements), and then applies a least-square estimate to compute the displacement between the two acquisitions, applying the usual RANSAC outlier rejection scheme.
- Visual SLAM can enhance the precision of the pose estimates produced by Visual Odometry, and has the benefit to produce a Landmark Map that can be used for localization. The InFuse approach to Visual SLAM is to build upon Visual Odometry, applying an extended Kalman filter over a fixed number of landmarks to comply with computation constraints and keep the process time bounded.
- Pose-Graph SLAM is an approach that can substitute for both the Visual Odometry and Visual SLAM DFPCs, using only point cloud data as exteroceptive inputs (*e.g.* to allow localization while navigating in low lights, using a Lidar). Pose estimates are produced at the reception of each point cloud, using an Iterative Closest Point approach (ICP), and the satisfaction of some criteria (such as coverage) is assessed to select the point clouds that are memorized in the SLAM map.

3.3.3.2 Localization while driving back

This localization service exploits the Localization Map built during the forward traverse.

There are several alternatives for the definition of the Localization Map: it can either be the Landmark Map or the Point Cloud Map, depending on the SLAM solution that has been activated. In both cases, the process consists in refining the position estimated either by Odometry or Visual Odometry by matching detected visual features with the landmarks stored (or by applying an ICP approach between the current point cloud and the closest one that has been memorized). The localization map can also be the Rover Total Map (maybe re-processed), in which case the localization solution is very similar the absolute localisation process.

3.3.3.3 Absolute localization

Absolute Localization exploits matches established between data gathered by the rover and an initial orbiter map. The appendix 4 details the trade-off analysis that defined the solution to be developed within InFuse: the built DTMs, along with the associated luminance layer will be reused to extract geometric and luminance features to be matched with similar features extracted from the orbital map. Note that this solution does not exploit any sun sensor, which can be used to recover an aboslute heading. In case such a sensor is available, its output (absolute heading) can be fused with the proposed solution.

3.3.3.4 Localization wrt. a man-made asset

This localization scheme is strictly equivalent to the one that is used to localize the target with respect to the chaser in the Orbital RI: the very same DFNs will be used to defined the associated DFPC (see section 3.5). Only unique object related features will be considered here, which include sample canisters and other associated tools to be defined.

3.4 Target satellite detection

Figure 5: Illustration of the visual detection (and bearings estimation) of a satellite at far range

3.4.1 Definition of the data product

Based on the camera resolution and focal length specifications expressed in the Compendium [Compendium], a wide field of view camera is the most appropriate sensor to be used for far range target detection, and does not require precise pointing before initialization. However, at great distances, a target the size of a typical satellite would only register as a few pixels on the sensor (e.g. 10 pixels wide. for a 5m wide target at 120m, with a 65°, 1024x1024 pixel camera).

In these conditions, it is not expected that the localization DFPC will be able to resolve enough geometric features on the target to estimate its full pose. This function therefore ensures the production of a range and bearing localization of the target with respect to the chaser spacecraft at the necessary accuracy and frequency to initiate and servo an approach phase and allow for the preparation of the rendezvous phase. The baseline input data are monocular images and rangefinder measurements, which can be augmented by the inertial measurements of the chaser spacecraft for a more robust tracking.

This function effectively serves as the long-range version of the state estimation function presented in section 3.5.

3.4.2 Definition of the associated DFPCs

As presented in Figure 6, the core of this DFPC consists in a visual tracking function capable of producing an estimation (and associated uncertainties) of the range and bearing of the target spacecraft relative to the chaser. The relative simplicity and low computational complexity of typical filtering techniques make them well-suited for this type of application. As inputs, this tracking function takes 3 possible types of measurements:

- The position of the detected target in a monocular image, providing bearing information. At long range, this detection is performed with a simple image filtering and thresholding function, as geometric features cannot be resolved yet.
- A lidar or radar rangefinder distance and bearing measurement. The lidar and radar also have the advantage of ensuring detection in adverse lighting or eclipse conditions.
- Chaser spacecraft inertial data or Attitude and Orbit Control System (AOCS) rotational states (angular position and velocities), enabling a more robust tracking of the target through prediction of the chaser motion.

The tracking filter is initialized with a selection by the user of the target spacecraft in one of the acquired monocular images. The selected target position and visual appearance can then be used to facilitate detection in subsequent images.

It should be noted that the output of the DFPC shall somehow be included in the chaser control loop in order to ensure constant visibility of the target in the camera and rangefinder FOVs as the chaser progresses along its path.

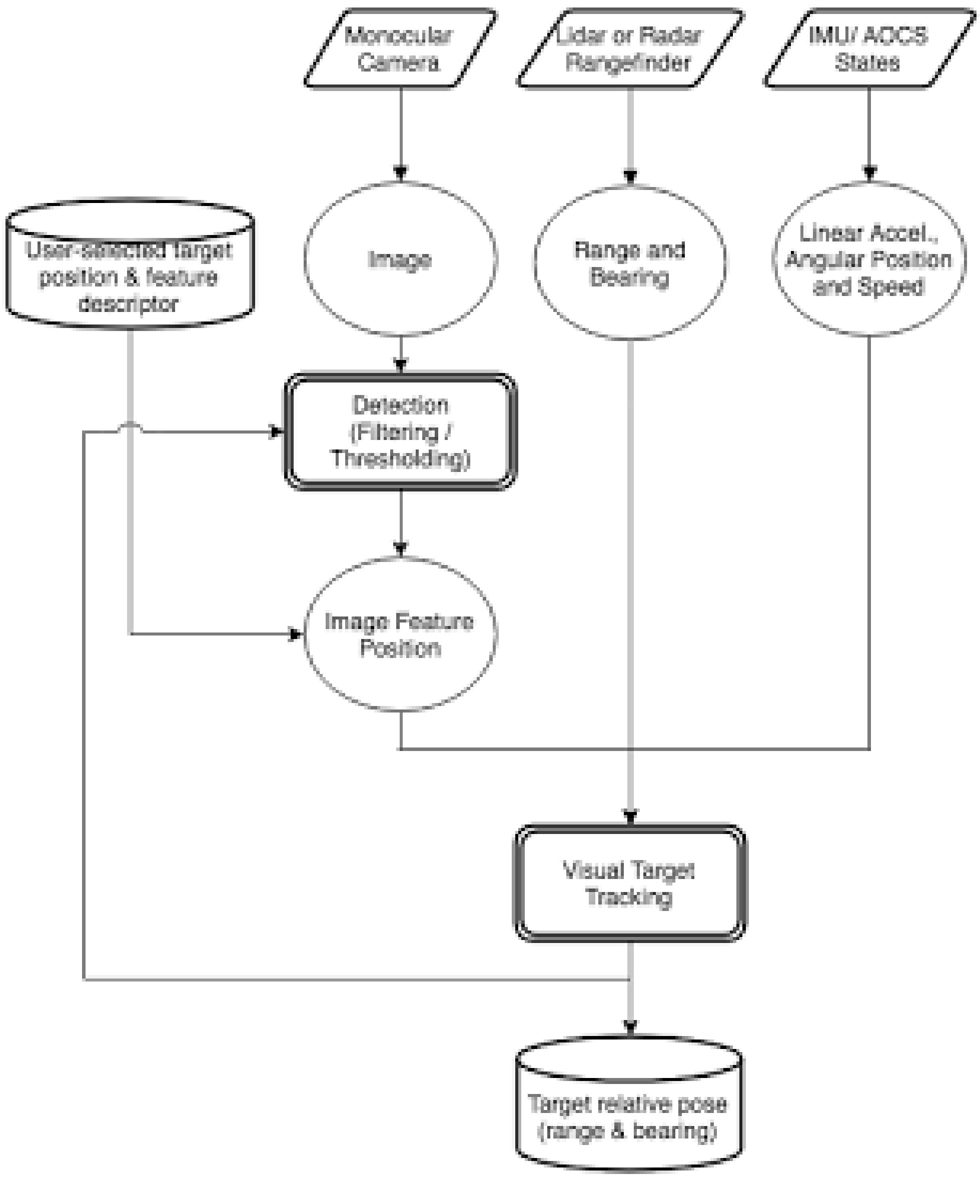

Figure 6: Overview of the detection, range and bearings estimate of the target satellite at far range

3.5 Target satellite state estimation

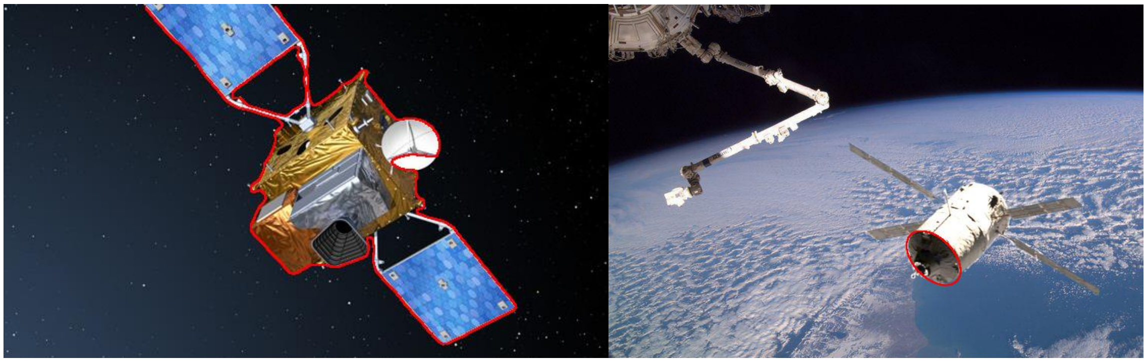

Figure 8: Illustration of the target satellite localization at mid-range (left, the full target model is used to localize it) and short range (right, only parts of the target model are used to localize it).

3.5.1 Definition of the data product

The function to ensure is the production of the position and orientation and associated derivatives (speed) of the target satellite with respect to the chasing servicer, with the necessary accuracy and frequency to plan and execute a rendezvous maneuver.

The required states for rendezvous of a resident space object such as a client satellite depend on the range from the servicer. The employed localization methods to estimate the states also depend on those states. Over time, the states will be improved as more data is gathered. Two cases are considered:

- At mid-range (approx. below 100m), it is most likely to detect the 6 DOF pose of the *complete* target satellite or of the part facing the servicer,
- At close-range (approx. below 5m) the pose of one or multiple target points on the surface of the target satellite are of interest, e.g. for performing a grasping maneuver. At close range, more points are also available for state estimation, improving overall accuracy.

The accuracy of the state estimation must allow to avoid collision and damage of the target, given a certain inaccuracy of a positioning approach, e.g. for rendezvous or for the positioning of a robotic tool. The required accuracy depends on the actual geometry and the characteristics of the target (points). Note that the state estimation is based on a geometric model of a target and does not rely on any assumption of the servicer or target dynamics. However, the estimated poses can be further used with a such assumption for specific mission, for example a GNC navigation function may incorporate servicer and client dynamics using filtering techniques.

3.5.2 Definition of the associated DFPCs

The state estimation requires visual information, e.g. in the form of passive camera images or active range measurements. Due to the complexity of active range sensors for space, passive cameras are a valid option, that require little mass and power. For far and mid-range a simple, single point range measurement might be an option. The correct choice of the sensor suite depends very much on the actual target and the specific mission.

In the case of camera-based operation, well illuminated images with good level of contrast are required as input in order to allow the detection of certain features or keypoints on the target. Additionally, a guess of the target state from previous measurements, e.g. from ground based tracking for the far-range case, is important for the initialization of the DFPC. A good initialization can increase the success and accuracy of the target pose estimation.

At close range, either two cameras operating in a stereo vision configuration (more reliable) or a single camera taking multiple images (less reliable) can be used to build up a point cloud over time of a moving target which can then be compared with a model shape to determine the identification and orientation of a target.

Figure 9 shows the generic flowchart of the target state estimation.

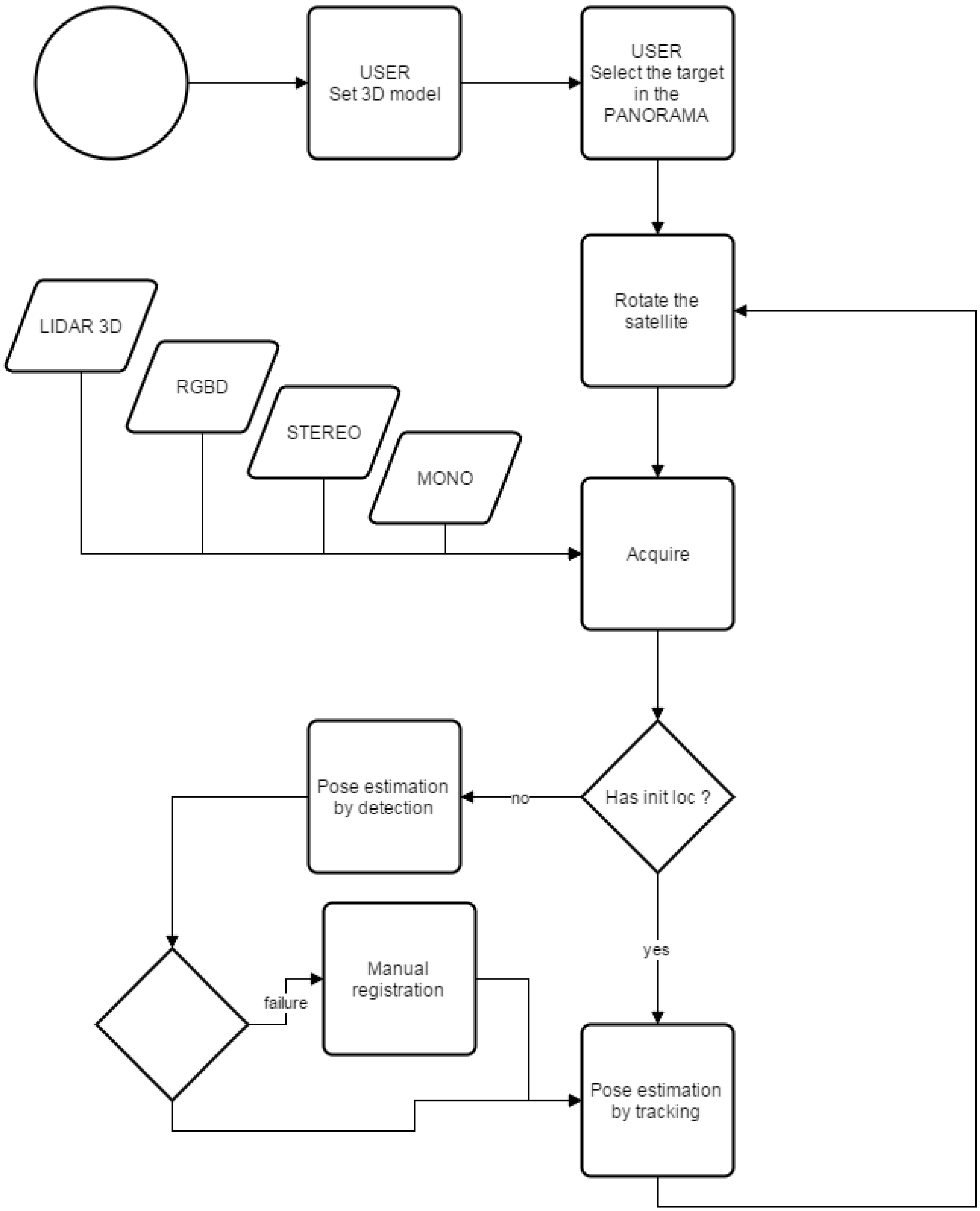

Figure 9: Generic flow chart of the satellite state estimation

Table 1 shows the states of a target satellite according to ranges of the target from the servicer and appropriate image features suitable for tracking.

Range	States	Features	Remark
Mid-range	Translation, orientation	Edges, contour, keypoints	Resolution of camera determines range and accuracy
Close-range	Translation, orientation	Edges, keypoints, colors/shades	Requires a higher accuracy and processing capacity than mid-range. Active illumination unit on robotic end-effector might enhance the illumination conditions.

Table 1: The states and image features of a target satellite at various ranges

Camera-based tracking of a resident satellite relies on image features such as edges and keypoints. The model of the satellite is often assumed to exist, and used for a model-based tracking. The model-based tracking can exploit edges or keypoints visible in an image. Table 2 shows benefits and disadvantages of edge and keypoint based tracking in the context of rendezvous for an on-orbit servicing.

Features	**Advantage**	**Disadvantage**	**Tracking method**	**Remark**
Keypoints	Easy to detect	Few in number and unstable due to reflections Hard describe invariantly	Model and 3D sensor data from stereo registration	Can require significant processing time depending on size and precision
Edges	Available abundantly on spacecrafts Easier to match model and image Faster	Ambiguous for symmetrical structures	Model-image edge matching along edge normals	Most spacecrafts consist of smooth structures and are poorly textured objects
Hybrid features	Exploit both edge and keypoint features Robust	Computationally expensive for tracking	Integrating color statistics and intensity edges	Requires high quality camera

Table 2: Comparison of image features to estimate the target satellite state

There exist several methods of tracking which exploit 3D model and image features or pixel intensities. Based on a comparative analysis of Table 2, and the stringent constraint of resources (power, speed, memory) for space applications, the edge-based tracking approach is the most suitable technique for on-orbit servicing. Hence, the reference implementation focuses mainly on the edge-based methods due to the fact that space objects are man-made textureless and provide abundant image edges features. Feature-based tracking is kept as an option for more high performance and high memory processors to produce high resolution point clouds of objects.

3.5.2.1 For the mid-range approach

Target detection is achieved by first performing movement or parallax estimation on the target scene to isolate the target, and then performing shape detection on the isolated target in the scene to estimate pose and relative motion over several frames.

- Performance: Dependent on range but can be accurate to several meters;
- Maturity: Tested in simulation, not in realistic sensing scenarios;
- Portability: Based on OpenCV functions and reproducible on other platforms;
- Expertise in InFuse: STRATH.

Target pose is estimated using template tracking methods such as the ones implemented by the ViSP and LINEMOD libraries. By using image data as well as a point cloud generated by a ToF or stereo camera, these methods attempt to estimate the pose of the target by matching the detected edges on a pre-trained template (LINEMOD) or on a simplified 3D model (ViSP). These methods are valid as long as the whole target can be seen in the field of view of the sensors.

- Performance: Dependent on range but can be accurate to several centimeters;
- Maturity: Tested in simulation, not in realistic sensing scenarios;
- Portability: Based on open libraries and reproducible on other platforms;
- Expertise in InFuse: MAG.

It could be used for close range approach with the right sensors and an appropriate 3D model.

Target pose could also be estimated using an Iterative Closest Point optimisation between the a priori 3D model of the target and a 3D point clouds provided by a LIDAR or TOF camera.

- Performance: Could be very accurate if the 3D model is discriminative (shape sufficiently rich and compatible with the LIDAR sub-sampling);
- Maturity: Tested on few laser scans;
- Portability: Based on open libraries and reproducible on other platforms;
- Expertise in InFuse: MAG.

It could be used for close range approach with the right sensors and an appropriate 3D model.

3.5.2.2 For the close range approach

Pose estimation by detecting edges in passive camera images and matching them to edges of a simplified 3D model of the target area. The approach benefits from using camera images taken from different vantage points, e.g. with a stereo camera system. It requires a valid guess of the target’s relative pose for initialization purposes.

- Performance: Possible accuracy can be in the range of one to a few centimeters but it depends very much on the target geometry and the illumination conditions;
- Maturity: The approach has been successfully tested using real image data from an OOS simulation facility;
- Portability: Based on C++ and partly opencv library
- Expertise in InFuse: DLR.

3.6 3D target reconstruction

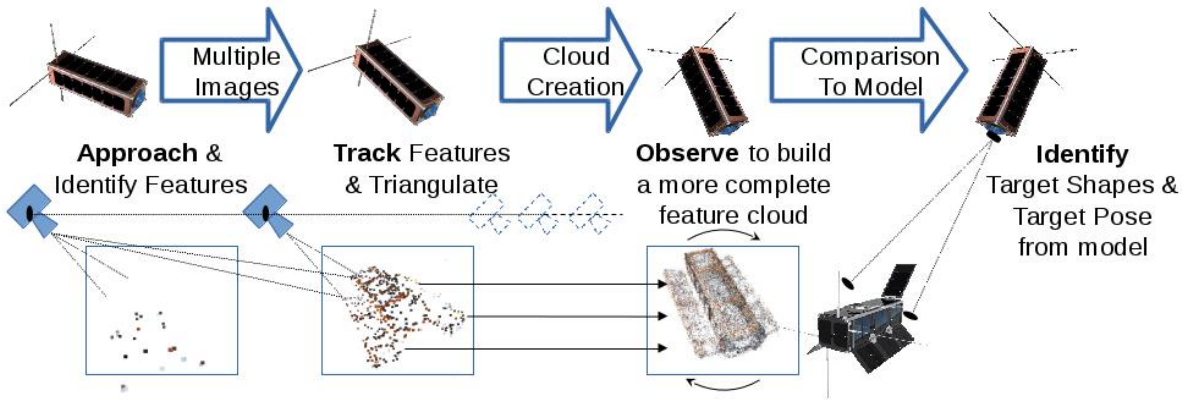

Figure 10: Illustration of sequential reconstruction of a target satellite from multiple perspectives of a chaser satellite.

3.6.1 Definition of the data products

The product to deliver is a 3D model of the target (possibly textured), with the necessary accuracy for model based tracking. A functioning approach is to exploit a short-range multiple-view vision system. By means of visual feature detection and tracking across successive frames, features detected in two-dimensional images are matched and triangulated to provide three-dimensional feature maps using structure-from-motion techniques. Alternately, LIDAR or time-of-flight point clouds may be used. Triangulated points are organized by means of orientation histogram descriptors and used to identify and track targets over time. The state variables with respect to the camera system are extracted as a relative rotation quaternion and relative translation vector. Partially occluded targets can be identified as long as enough of the structure is seen to produce an unambiguous point cloud shape. The resulting products will include the tracked feature map of the target and the point cloud itself.

3.6.2 Definition of the associated DFPCs

3.6.2.1 Feature descriptors from 2-D images

The current approach to target reconstruction makes use of ORB, SIFT or SURF feature descriptors to track optical features across several image frames. A single camera can be used in this process if necessary, or multiple cameras with a known baseline difference can be used to increase the accuracy of the depth triangulation process.

3.6.2.2 Feature Matching and triangulation

Features are matched between images and the fundamental matrix for each image set is obtained through multiple-view geometry solution. Using the fundamental matrix, the features are triangulated in three dimensions and inserted into a point cloud space. In addition, bundle adjustment can be used after the triangulation process to increase the accuracy of the resulting local point cloud.

3.6.2.3 Ego-Motion Estimation

A perspective-and-point (PnP) solution from the fundamental matrix allows the ego-motion of the camera to be obtained during solution as well, which provides a means to localize the point cloud in space. As more features are obtained in sequence, they are added to this point cloud and increase the accuracy of the shape obtained.

3.6.2.4 Object Recognition and Tracking

Once a detailed point cloud is obtained, it can be used to identify shapes or features in three dimensions. A reference point cloud of an object or part of an object in space can be provided, and localized with six degrees of freedom within the reconstructed point cloud. The point cloud does not necessarily have to be a visual reconstruction, as it is possible to use LIDAR or time-of-flight camera generated point clouds as long as a similar density of points is obtained. To perform this operation, the SHOT histogram descriptor is used to characterize several keypoints within the point cloud and recognize shapes from comparing the points in relative orientations and distances from each keypoint. The more keypoints are used, the more accurate the identification but the more processing power is needed. The advantage of using multiple keypoint histograms is that small parts of larger point clouds can be recognized from models without necessarily having a model of the entire point cloud to match. Figure 10 shows the process of building a point cloud and identifying a target in three dimensions from a pre-defined point cloud shape reference.

3.6.2.5 Kalman Filtering

A Kalman filter can also be used to improve the accuracy of relative object tracking by making use of ego-motion estimate from the camera as well as the movement of the tracked point cloud. Both Extended Kalman Filters (EKF) and Unscented Kalman Filters (UKF) can be used, though the UKF provides higher model flexibility and better performance in exchange for higher computational load. The Cubature Kalman Filter (CKF) is a variant of the UKF with better performance in higher-order systems.

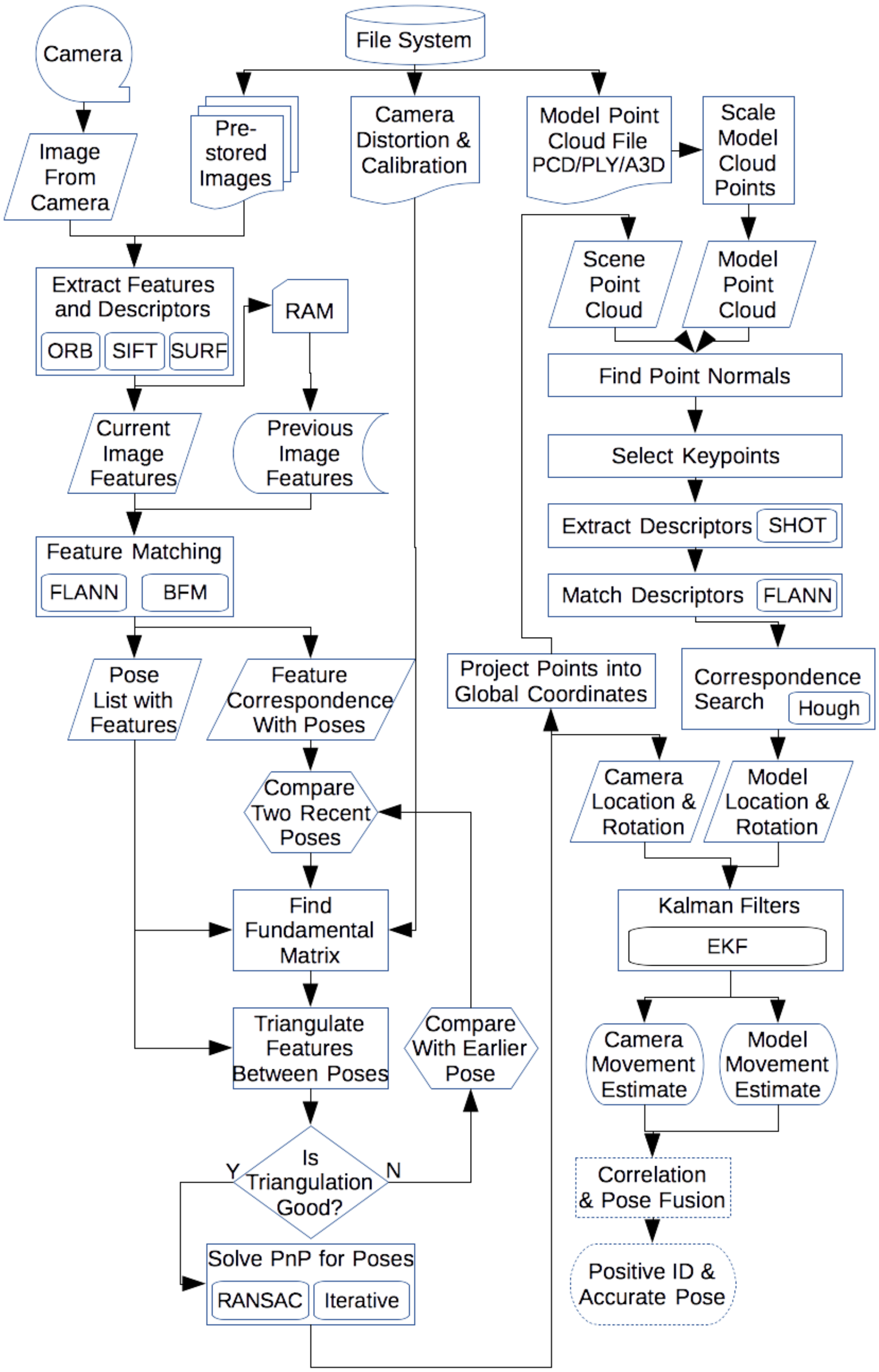

Figure 11: Generic flow chart of 3D target reconstruction and tracking using 3D model-based point cloud recognition

The reconstruction and identification method is highly processing intensive due to the amount of matching required to triangulate points and identify three dimensional point clouds (usually RANSAC is used in both cases) but can produce high fidelity point clouds of a target as long as enough visual features are available. As several algorithms are available at each stage of feature detection, triangulation, reconstruction, and identification, trade-offs can be made in terms of the computational time required versus the quality of results obtained. Table 3 provides a tradeoff comparison of several of these options available.

Operation	**Advantages**	**Disadvantages**
Feature detection using ORB	Open-source, fast, invariant to rotation	Lower accuracy and scale invariance than SIFT/SURF
Feature detection using SIFT	Good quality and commonly used	Patent-encumbered, high computational load
Feature detection using SURF	Faster than SIFT and almost as good quality	Patent-encumbered, not as fast as ORB on fixed-point
Single-camera comparison of images	Less hardware needed	Triangulation harder and less reliable than multi-camera
Dual or Multiple-camera comparison of images	Better triangulations, more reliable reconstruction	More hardware with known baselines needed
Iterative (brute-force) matching of features, clouds, and PnP	Reliable and simple	Very high computational power needed
FLANN (Fast Approximate Nearest Neighbor) matching	Much faster for features and very accurate	Slightly less accurate but not noticeably
RANSAC (random sample) matching	Much more efficient than brute force, almost as good	More complex method, very occasionally incorrect
Extended Kalman Filter estimation	Well known, efficient, and robust estimator	More complex to design and not as flexible or high order
Unscented/Cubature Kalman Filter estimation	More flexible and performs for higher order systems	Higher computational load

Table 3: Tradeoffs between different algorithms for the point cloud reconstruction process

In summary of the point cloud reconstruction method described here:

- Performance: Centimeter level accuracy at close range with high resolution hardware. Accuracy decreases with triangulation distance and resolution, and is least accurate in the Z (away from camera) direction.
- Maturity: This method has been tested using laboratory images of spacecraft movement with a simulated spacecraft movement model
- Portability: Based on OpenCV functions and reproducible on other platforms as long as memory and processing power are sufficient for near real time operation.

4 Appendix: analysis of the Absolute Localization state of the art

4.1 Introduction

Mars Explorations have achieved significant milestones in the last two decades. But still there are enormous challenges ahead. For example, through March 2017, the ongoing Mars Exploration Rover(MER) Curiosity has driven only around 16 kilometres since the mission's August 2012 landing on Mars. The future explorations of Mars will aim to have Rovers that can explore at least few hundreds of kilometres away from its landers site. This requires autonomous localisation of the rover over a long range.

Given the initial rover location (or its estimate), the Rover's pose (x; y) (sometimes also orientation) at any given time can be determined by various Dead Reckoning techniques such as Wheel Odometry, Visual Odometry(VO) and local Bundle Adjustment(BA). Among these, VO and BA are used mostly to complement the wheel odometry whenever needed. Most of these dead reckoning approaches exhibits unbounded error growth with distance traversed. Absolute (Global) Localisation techniques can be used to correct dead-reckoning pose estimates once these become unreliable. On earth, the GPS is used for this purpose, but the satellite infrastructure needed for such a system is not available for Mars applications.

Map based Absolute localisation is the process of finding the Rover's position in a global map (for eg. map acquired by the cameras in the orbiter, such as HiRISE Camera in the Mars Reconnaissance Orbiter (MRO) Orbiter) by matching features in the global map with the features in the local map obtained by the Rover's sensors (Camera, Lidar etc). As the resolutions of the orbiter maps and rover maps are not the same, this can be a challenge. The absolute localization approach is illustrated graphically in Figure 9. We assume that the rover's position is known with a maximum error of 100m (initial Pose estimate). This initial estimate, the orbital data around these location, and the rover's extroceptive sensor data are combined in order to find the current pose of the rover in the orbital (absolute) reference frame.

HiRISE stereo pair images has a resolution of 25cm/pixel and the corresponding DTM and Orthoimage has a resolution of 1m/pixel. Some objects having size less than the resolution of the image can also be detected due to the presence of shades.
Various previous works have tried to address the main challenges of the absolute localisation problem in either planetary context (Lunar or Martian) or in urban scenarios. We will go through the most relevant of those works in the following section.

4.2 Categorization of approaches

We will analyse the previous works in the absolute localisation of planetary rover and the major techniques used by them. The approaches can be classified according to the data type used from the rover and orbiter, and how the data association is performed to localize the rover. We classify them as follows:

A. Skyline Based Approaches.
B. Localisation using DTM Registration.
C. Localization using Feature Matching between Point-Cloud and DTM.
D. Localisation using Orbiter-Rover Image Feature Matching.

The following section describes each of these types of approaches and also analyses the previous works utilizing similar approaches.

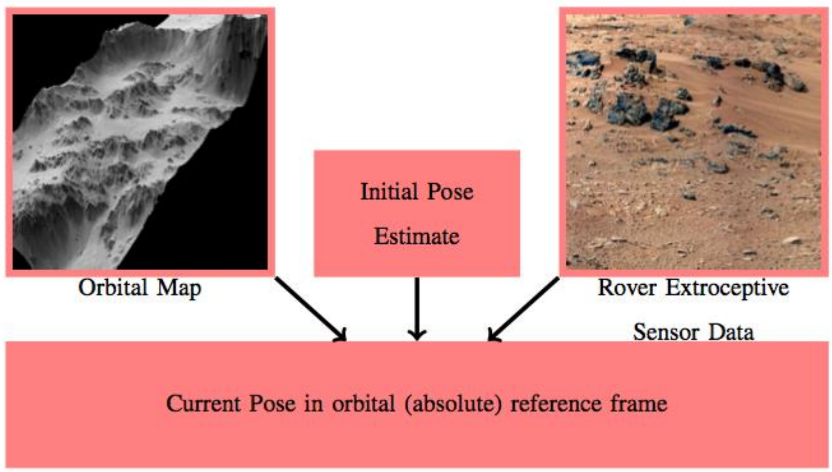

Figure 12: A Graphical Illustration of the Pose Estimation Problem

4.2.1 Skyline Based Matching Approach

Description of the Approach:

A Skyline is defined as the Sky-Ground boundary perceived by the rover camera as it performs a full 360^0 turn in place. The main drawback of the Skyline based approaches are that they do not consider the topography below the Skyline, which sometimes has many important features that might aid the localisation.

Some of the initial approaches developed in the rover localization relied on Skyline matching. Here, a skyline extracted from the Rover DTM (generated from Stereo images) is matched to the skyline computed from the Orbiter DTM in order to localize the Rover in the orbital DTM.

Contrary to other types of approaches, the skyline matching does not aim to localize a rover accurately, but rather to find an initial estimate of the location, as it gives only a minimal estimation accuracy. Two prominent skyline based approaches are proposed by Stein and Medioni [1] and Cozman and Krotkov [2]. The main parts of both approaches can be summarized as follows :

Offline Tasks (Precomputed and stored on the Rover)

– Skyline Rendering : Computation of Skylines for each point in the DTM.
– Feature Extraction on Skylines (Optional)

Online Tasks (Computed on the Rover)

– Skyline Detection from the Rover Images.
– Feature Extraction on Skylines (Optional)
– Search and Locate the Rover Skyline in the rendered DTM Skylines.
– Pose Estimation, (x; y) or (x; y;), based on the Skyline location.

The main approach consists localising a rover at an unknown position, also known as ”lost in space” or ”drop off problem”, laying inside an area that has been mapped, by means of orbital imaging. The robot is assumed to have precise knowledge about its orientation, based on for example sun sensor as the one presented in [3]. The rover is equipped with panoramic cameras, or normal cameras on pan units, capable of capturing full circle images of the horizon. Then, the Skyline is detected employing image segmentation techniques. Assuming that an orbital DTM is available covering the area of uncertainty, within which the rover lays, a ”simulated” skyline is rendered at each and every point of the DTM. A search follows to match the rover's skyline to the rendered ones.

Stein and Medioni [1] proposed a table based matching strategy for the rover localisation. As an initial step, the panoramic views for each location is computed using the orbital DTM, encoded (by polygonal or line segments approximation, which captures some curvature information in the form of angle between consecutive segments) and stored and indexed in a table (database). In order to find the location of the rover, the panoramic view at the current location is computed, encoded and compared with the database to retrieve the candidate locations. The best candidate is found by applying further geometrical constraints on the candidates. Here, the robot orientation is not estimated, and is assumed to be perfectly known using other sensor measurements, only the (x; y) coordinates are estimated. Stein and Medioni [1] reports estimation accuracy between 300 and 600 meters on a global map of resolution 300m. The approach developed in this paper was for any general applications. Cozman and Krotkov [2] used this approach and applied to the planetary context.

Cozman and Krotkov [2] presented VIsual Position Estimation for Rover(VIPER) algorithm for localising a rover ”lost in space” or ”dropped off”. They uses the same idea of Stein and Medioni [1] but applies to the context of planetary rover localisation. They also presents extensive analysis of their algorithm. The VIPER system was tested in five completely different terrain types, and the best position estimation accuracy achieved is from 84 to 200 meters for a rendered Skyline covering 154km^2 (in a 30m grid). Overall, the VIPER System determines position with an acccuracy from 2:5 to 6:5 global map cells. Hence, with an improved global DTM resolution (around 1m for HiRISE), the system is expected to give a localisation accuracy of around 2:5m to 6:5m.

The idea of Skyline based user localisation was also explored in [4], where the authors applied it to localize a vehicle, fitted with a fish eye camera on top, in urban canyons, where the GPS signals are week. They estimate the global position by matching skylines extracted from the omni-directional camera images to skyline segments obtained from coarse 3D models. They use a skysegmentation algorithm based on graph cuts. This method works well in case of complex skylines as in the case of the urban canyons, but has less scope in the case of planetary rover localisation.

Selection of the best Skyline Matching Approach:

Out of the three skyline based approaches mentioned before, the VIPER algorithm presented by Cozman and Krotkov [2] is the most promising.

With the advances in machine learning, the manual tasks in the VIPER system can be automated. We will compare this algorithm with the other types of approaches.

4.2.2 Localization using DTM Registration

In these type of approaches, the DTM obtained from the stereo pair images of the HIRISE camera is matched with the DTM obtained from the Rover Stereo Pairs to localize the rover in the orbiter DTM. DTM resolution can change drastically between these two types and that can be a challenge.

Van Pham et al.[5] proposed an absolute rover localisation algorithm based on modified particle filter, which is able to retrieve the location of the rover on the orbiter DTM. In this method, the Bayesian posterior probability density function corresponding to the global position of the rover is approximated by a set of samples, the so-called particles. In this way, the algorithm can take into account multiple possible rover locations that arise in the lost-in-space problem. The initial particles are selected based on the prior informations available about the rover location. Then, during the prediction step, the Gaussian distribution of each particle is modified according to new motion estimate (from visual odometry). The sampling phase adds new particles if the variance of any particle is greater than a threshold. This is followed by an update phase, where the particle position and the likelihood are updated, using the local and global DTM matching, as soon as the rover moves significantly, and then the particles are resampled into a discrete grid with the same spatial resolution of the global DTM. The experiments were carried out in a sand quarry and the global DTM was created by means of a UAV, while ground control points were used for georeference. Zero mean Normalized Cross-Correlation is used to compare the local and predicted (from global DTM) DTMs.

Experimental results show that this method can provide a localization estimate with 2m accuracy on a 1m resolution global DTM. However this is achieved by the computational complexity of the algorithm. But considering the fact that the rover initial position can be known by at least a few hundreds of meters, the computational complexity might be reduced.

4.2.3 Localization using Feature Matching between Point-Cloud and DTM

In this type of approaches, the features of the point-cloud obtained from the rover sensors (for example using LIDAR) is matched with that of the orbiter DTM.

Carle et al. [6] localizes a rover by matching features from an orbital (global) DTM to features from the rover (local) based point cloud (PC) of 3D LIDAR scans. The PC and DTM just differ in the way they represent the data. The approach follows feature based approach, with features being the prominent peaks that appear on Global DTM and Local Point-Cloud. The relative distribution of the prominent peaks are used to localize the Rover in the orbiter DTM. The extraction of the features, peak detection in particular, is performed on the DTM and PC by filtering them with the external morphological gradient, i.e. the difference between the dilated image and the original image. The correspondence between the rover and orbital features are computed utilizing a method named "Data-Aligned Rigidity-Constrained Exhaustive Search" (DARCES) [7].

In the MOGA (Multi-frame Odometry-compensated Global Alignment) framework developed in the same paper [6], the feature correspondences obtained from feature matching are combined with visual odometry and orientation measurements into a simultaneous localisation and mapping (SLAM) problem to refine pose estimates. Outlier feature correspondences are also rejected with an enveloping RANSAC (Random Sample Consensus) algorithm. This is essentially a batch SLAM algorithm that fuses relative and absolute pose measurements over an entire rover traverse. This is particularly useful to localise the rover on the traverse, in case if a particular frame does not have a DARCES feature matching solution (for eg. a frame with only one or two extracted features). In such a case, the DARCES solution is estimated using VO to the next-closest, solved frame. This is a good approach to combine the absolute localisation and visual odometry.

When the LIDAR scan contains sufficient number of good topographic features, the localisation produced position errors no more than 100m and as low as few meters in many cases, using a global orbital map of x,y resolution 13m and 24m respectively. Since we are expecting to exploit a global map of 1m resolution, one can expect that the estimation accuracy will be improved significantly. Currently, LIDAR based systems have been successfully used in space missions on-Earth-orbit and thus are space qualified. The major advantage of LIDAR systems is their superior resolution and range [8]. The most significant issue with the LIDAR based systems are their heavy-ness and the high power consumption.

4.2.4 Localisation using Orbiter-Rover Image Feature Matching

In this type of approach, the orbital and rover images are used directly to extract the features, instead of DTM generated from them. Hwangbo et al. [10] presents an approach to rover localisation based on feature (rocks) extraction and distribution pattern matching between ground (rover) and orbiter imagery. The distinctiveness of this approach is that it employs a two-step approach utilising both terrain and rock matching to hierarchically trace correspondences between orbital and rover images. Firstly, the 3D terrain matching is utilized to locate the region of interest in the global DTM that correspond to the rover position. The rover DTM is created from stereo imagery. For the purpose of having the scale in elevation, the mean elevation is subtracted from both DTMs. Then, the matching is performed by identifying the region of the maximum weighted correlation of values and slopes within the global DTM.

As soon as the region of interest is located, the rock extraction and pattern matching is performed.

The rocks on orbital imagery are identified via an intensity threshold technique. The rock detection on rover imagery includes the removal of ground points and the identification of peaks, i.e. highest points in an area having elevation of more than 25cm. Finally morphology of the rocks is examined to eliminate false positives. The experimental analysis has been performed using a stereo pair of HiRISE images that covered Columbia Hills area of the MER Spirit rover landing site.

An improved version of the same approach has been proposed in [9] by Di et al. The major difference is in the feature extraction phase, where the new method not only extracts rocks but also SIFT keypoints (which are invariant to image translation, scaling and rotation). This enables the new algorithm applicable to both rocky areas such as those in Spirit Rover Landing site and outcrop areas such as those in Opportunity Rover landing site. The algorithm can be described in a four step process. In the first preprocessing step, 3D mapping is performed on the rover stereo images taken at one position to generate 3D point clouds, a DTM and an orthophoto. At the same time, the HiRISE Imagery is preprocessed using histogram stretching and a gaussian filter to enhance the image and remove any noise effects. In the second step (feature extraction), large rocks and keypoints are extracted from ground-image-derived 3D point clouds and orthophoto. The same points are also extracted from the HiRISE imagery. In the third step, feature matching is performed for rocks (rocky areas) or keypoints (outcrop areas). A RANSAC (RANdom SAmple Consensus) like procedure is applied in this feature matching step to eliminate incorrectly matched feature points (outliers). In the final step, the rovers are localized within the HiRISE imagery through a similarity transformation.

4.3 Analysis of the approaches

In this section, we will assess the different approaches listed in the previous section based on various criteria. The different approaches evaluated here are that of Cozman and Krotkov (VIPER Algorithm) [2], Van Pham et al. [5], Carle et al. [6] and Di et al. [9]. These approaches will be analysed following the criteria below and summarized in Table I.

A. Is the Scheme Instantaneous?
This criteria analyses if the scheme is instantaneous or not, i.e. whether the rover needs to move to different locations to gather data before localising itself. [2], [6] and [9] are instantaneous approaches, while [5] is not, as it needs to update the particles using the motion estimate.

B. Applicability in Different Terrain Types
The type of terrain encountered by the rover can influence the accuracy of the localisation algorithms. Although not all types of terrain can be named, the most common terrain types includes : ”rock-cluttered”, ”bedrock outcrop”, ”layered-rock”, ”sand dunes”, ”gullies”, ”mesas”, ”flat terrains”, etc. Some of the terrain types encountered in Mars are shown in Fig.4, which includes some orbital images and also onerover image (fig.4f).

C. Prediction of Localisability from the Orbiter Map
This criteria analyses if the localisability of the Rover can be predicted from the Orbiter Map charac-teristics, like the resolution of the global map or some visual features of it.

D. Precision obtained in Localisation

Precision obtained in the localization as a function of the orbiter map resolution.

E. Uncertainty on the position estimate

This criteria checks if the approach provides an uncertainty on the position estimate or not.

F. Computation Time and Memory Requirement

Computation time required and memory usage are two important criteria for choosing the best approach. As there are offline and online computations in most of the schemes discussed, we are considering the online computation time and memory usage to store offline computation result.

G. Re-usability of Computational Blocks / Data Products

As there are more functional blocks in the Rover other than the absolute localisation block, it is important to see if some of the computations performed by other blocks (for eg. the computations that has already been done for the autonomous navigation task) can be reused in order to reduce the overall computational complexity of the Rover.

H. Validation on Realistic Data

This evaluates the approaches for the validation on the realistic data from Mars terrain or similar terrains.

	Skyline Based Approach [2] (VIPER)	Point-Cloud/DEM Registration [6] (LIDAR based)	DEM Registration [5] (using Particle Filter)	Orbiter-Rover Image Feature Matching[9]
Terrain Types	Any terrain with visible skylines	Terrains with prominent peaks	any terrain with features that can be extracted in both orbiter and rover images	Rocky and Outcrop areas
Prediction of Localisability from Orbiter Map	Yes, but is not very obvious	Yes	Yes	Yes
Localisation Precision Obtained	2.5 to 6.5 times DEM resolution	$50m$ precision on a global map of $13m$ resolution	$2m$ precision on a $1m$ global DEM	0.23m accuracy on a global map of resolution 0.25m
Provide an uncertainty on the position estimate	?	Yes	Yes	?
Localisability from a Single Position	Yes	Yes, for single frame case	No, it needs motion estimates to update particle weights	Yes
Computational Time / Memory Usage	$15s$ for position estimation. Huge ($31MB$) memory usage for Skyline Rendering	$15min$	around 5 sec, very high memory usage for initiating all the particles	1 to 3 min
Reuse of previous Data Blocks	No	No	Yes (DEM Map built for avoiding obstacles during navigation)	Partly Yes (DEM)
Validation on realistic data	++, Tested on Apollo-17 landing site	++ (Tested on Mars like Terrains)	++ (Tested on Mars like Terrains)	+++ (Tested on real data from MER Spirit and Opportunity Rover and HiRISE data)

Table I: Comparison Table for the best scheme in each of the four types of approaches presented here

4.4 Conclusion

After comparing the different approaches for the purpose of absolute localisation using Table I, the scheme presented by Di et al. [9] has been selected for the final implementation due to its advantages found over the other methods. It will be implemented with (possibly using DARCES method for feature matching) or without additional modifications.

4.5 References

[1] F. Stein and G. Medioni, “Map-based Localization Using the Panoramic Horizon,” in Proceedings 1992 IEEE International Conference on Robotics and Automation, May 1992, pp. 2631–2637 vol.3.

[2] F. Cozman, E. Krotkov, and C. Guestrin, “Outdoor visual position estimation for planetary rovers,” Autonomous Robots, 2000.

[3] P. Furgale, J. Enright, and T. Barfoot, “Sun sensor navigation for planetary rovers: Theory and field testing,” IEEE Transactions on Aerospace and Electronic Systems, vol. 47, no. 3, pp. 1631–1647, July 2011.

[4] S. Ramalingam, S. Bouaziz, P. Sturm, and M. Brand, “SKYLINE2GPS: Localization in Urban Canyons using Omni-skylines,” in 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems, Oct 2010.

[5] B. Van Pham, A. Maligo, and S. Lacroix, “Absolute Map-Based Localization for a Planetary Rover,” in 12th Sympusium on Advanced Space Technologies and Automation in Robotics, Noordwijk, Netherlands, May 2013, pp. 1–8.

[6] P. J. Carle, P. T. Furgale, and T. D. Barfoot, “Long-range rover localization by matching lidar scans to orbital elevation maps,” Journal of Field Robotics, vol. 27, no. 3, pp. 344–370, 2010.

[7] C.-S. Chen, Y.-P. Hung, and J.-B. Cheng, “Ransac-based darces: a new approach to fast automatic registration of partially overlapping range images,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 21, no. 11, pp. 1229– 1234, Nov 1999.

[8] I. Rekleitis, J. L. Bedwani, and E. Dupuis, “Autonomous planetary exploration using lidar data,” in 2009 IEEE International Conference on Robotics and Automation, May 2009, pp. 3025–3030.

[9] K. Di, Z. Liu, and Z. Yue, “Mars Rover Localization based on Feature Matching between Ground and Orbital Imagery,” Photogrammetric engineering and remote sensing, vol. 77, no. 8, 2011.

[10] J. W. Hwangbo, K. Di, and R. Li, “Integration of Orbital and Ground Image Networks for the Automation of Rover Localisation,” 2009.

[11] M. Maimone, Y. Cheng, and L. Matthies, “Two years of visual odometry on the mars exploration rovers,” Journal of Field Robotics, vol. 24, no. 3, pp. 169–186, 2007.