

Autonomous over-the-horizon navigation using LIDAR data

Ioannis Rekleitis · Jean-Luc Bedwani · Erick Dupuis ·
Tom Lamarche · Pierre Allard

Received: 4 February 2011 / Accepted: 8 September 2012
© Her Majesty the Queen in Right of Canada 2012

Abstract In this paper we present the approach for autonomous planetary exploration developed at the Canadian Space Agency. The goal of this work is to enable autonomous navigation to remote locations, well beyond the sensing horizon of the rover, with minimal interaction with a human operator. We employ LIDAR range sensors due to their accuracy, long range and robustness in the harsh lighting conditions of space. Irregular Triangular Meshes (ITMs) are used for representing the environment, providing an accurate, yet compact, spatial representation. In this paper a novel path-planning technique through the ITM is introduced, which guides the rover through flat terrain and safely away from obstacles. Experiments performed in CSA's Mars emulation terrain, validating our approach, are also presented.

Keywords Space robotics · Planetary exploration · Mapping · Path planning

I. Rekleitis (✉)
School of Computer Science, McGill University, Montreal,
QC, Canada
e-mail: yiannis@cim.mcgill.ca

J.-L. Bedwani
Institut de recherche d'Hydro-Québec (IREQ), Varennes,
QC, Canada
e-mail: bedwani.jean-luc@ireq.ca

E. Dupuis · T. Lamarche · P. Allard
Canadian Space Agency, Saint-Hubert, QC, Canada

E. Dupuis
e-mail: erick.dupuis@asc-csa.gc.ca

T. Lamarche
e-mail: tom.lamarche@asc-csa.gc.ca

P. Allard
e-mail: pierre.allard@asc-csa.gc.ca

1 Introduction

Mobile robotics have enabled scientific breakthroughs in planetary exploration (Maimone et al. 2006). Recent accomplishments have demonstrated beyond doubt the necessity and feasibility of semi-autonomous rovers for conducting scientific exploration on other planets. Both Mars Exploration Rovers (MERs) “Spirit” and “Opportunity” have the ability to detect and avoid obstacles, picking a path that takes them along a safe trajectory. MER's have traveled up to 300 m/sol. On occasion, the rovers have had to travel to locations that were at the fringe of the horizon of their sensors or even slightly beyond.

The next rover missions to Mars are the “Mars Science Laboratory” (MSL) (Hayati et al. 2004; Volpe 2006) and ESA's ExoMars (Vago 2004). Both of these missions have set target traverse distances on the order of one kilometer per day. Both the MSL and ExoMars rovers are therefore expected to drive regularly a significant distance beyond the horizon of their environment sensors. Earth-based operators will therefore not know a-priori the detailed geometry of the environment and will thus not be able to select waypoints for the rovers throughout their traverses. One of the key technologies that will be required is the ability to sense and model the 3D environment in which the rover has to navigate. To address the above mentioned issues, the Canadian Space Agency is developing a suite of technologies for long-range rover navigation. For the purposes of this paper, “long-range” is defined as a traverse that takes the rover beyond the horizon of the rover's environment sensors.

The main contribution of this paper is the use of ITMs for constructing accurate models of the environment and also for facilitating the efficient planing of trajectories that are optimal for a set combination of constraints, such as distance traveled, terrain accessibility, and ruggedness. LIDAR



Fig. 1 The Mars emulation terrain (MET) with our modified P2AT robot

sensing as proposed in the presented work is also a departure from the traditional vision-based planetary exploration. Extensive experimental results from autonomous navigation at the CSA's Mars emulation terrain (MET) validate the proposed framework.

In the next section we discuss the state-of-the-art in robotic planetary exploration. Section 3 presents the overall process for planetary exploration together with a short description of our test-bed. Section 4 provides a summary of our approach on terrain modeling using LIDAR data. Section 5 presents the algorithm used for planning an optimal path for the rover using the Irregular Triangular Mesh (ITM) while keeping it a safe distance away from the detected obstacles. Global path planning and a brief overview of the navigation procedure are discussed next. Section 8 outlines CSA's approach to data management. Subsequently, experimental validation performed at MET is presented in Sect. 9. The paper concludes with lessons learned and a description of future work.

2 Related work

Work on planetary exploration can be divided according to the sensing modality used and also according to the environment representation used. Both vision (Matthies and Shafer 1987) and LIDAR (Hebert et al. 1989) technologies have been proposed, each one having different advantages and disadvantages. Early work on planetary exploration using LIDAR (Hebert et al. 1989), though promising, was not compatible with the space-flight weight-constraints. Simmons et al. (1995) proposed a stereo-vision based perception and navigation scheme which was tested on trajectories over a kilometer long; the sensing used provided a limited field of view and a modeling accuracy of 10 cm. The Mars Exploration Rovers are currently performing long traverses using stereo vision (Goldberg et al. 2002; Johnson

et al. 2008). This approach, although lightweight, requires more computing power and has limited range and accuracy. Currently, LIDAR based systems¹ have been successfully used in space missions on-Earth-orbit and thus have become space qualified, enabling their future use in planetary exploration missions. The major advantage of LIDAR systems over vision-based systems is their superior accuracy, resolution, and range.

Planetary exploration with a mobile robot was discussed as early as the late eighties. Among the earliest sensing modalities used was a laser based sensor (Hebert et al. 1989; Bares et al. 1989) for use with a walking robot. The terrain model used was an elevation map and the experiments were limited to a 25 m² terrain and a single leg. Terrain mapping was proposed to assist a walking robot as early as 1994 (Krotkov and Hoffman 1994), including classification based on the safety of traversal (Kubota et al. 2006). Planning over an elevation map was very time consuming but necessary for analyzing foot placement. More recently, the Morphin planner (Singh et al. 2000) uses stereo vision data with a resolution of 10 cm and selects the best of a number of alternative paths based on the terrain characterization as it is mapped on a grid representation. In contrast the A* based algorithm proposed in this paper returns a single path that is optimal in terms of the constraints specified. Vision in the form of stereo was also used by several different research groups (Matthies and Shafer 1987; Giralt and Boissier 1992; Kunii et al. 2003; Goldberg et al. 2002). Early work by Chatila et al. (1993, 1995) and Alami et al. (1998) proposed different path-planners depending on the condition of the environment. An early approach that has strong similarities to our approach was the work by Liegeois and Moignard (1993) where they used a Delaunay triangulation (Preparata and Shamos 1985) as the underlying structure for path planning. The main input for the triangulation algorithm was a topographical map and not any type of sensor data. As such, none of the challenges of long shadows and variable resolution were addressed. In addition the use of a topographical map does not reflect current conditions on an ever-changing terrain. More recent work using a triangular mesh was presented by Wettergreen et al. (2009) during field tests of a lunar rover prototype. The focus of their work was to investigate the mobility capabilities of the rover in lunar analog terrains. Grid representations in general are quite popular, both in 2D and in 3D, in the form of digital elevation maps (DEM). The accuracy of the model is directly linked to the resolution of the grid which is inversely proportional to storage requirements. In contrast irregular triangulation meshes (ITMs) maintain the same order of accuracy as the sensor while maintaining low storage requirements.

¹<http://www.neptec.com>; <http://www.optech.ca/>; <http://sm.mdacorporation.com/>.

Furthermore, different research groups have proposed a variety of schemes for planetary exploration addressing different problems (Giralt and Boissier 1992). One important aspect is the control architecture used. JPL for example introduced two software architectures: CLARAty (Nenas et al. 2003) which has focused on module reusability and CAMPOUT (Huntsberger et al. 2003) with a focus on multi-rover applications. Regardless of the sensing modality sensor data preprocessing, terrain modeling, and path planning (global and local) are major building blocks in most approaches, sometimes formulated as behaviors (Gat et al. 1994).

More recently, Mora et al. (2008, 2009) proposed a similar functionality to the one proposed in this paper, using digital elevation maps (DEM) for lunar rovers. Global and local DEM's were proposed, to be constructed from on-orbit imagery, and from LIDAR scans from the rover. In addition a Dijkstra based algorithm was presented taking into account ruggedness, slope, and distance. In this paper we present an A* based planning algorithm which is much more efficient than the Dijkstra based one, because the search is biased towards the goal and no calculations are consumed for cells in irrelevant directions. Further work (Ishigami et al. 2011) used the previous approach to calculate and evaluate several alternative paths off-line considering different cost functions.

Currently, the most advanced exploration robots that have been deployed for planetary exploration are the Mars Exploration Rovers (MERs) “Spirit” and “Opportunity”. These rovers have successfully demonstrated concepts such as visual odometry and autonomous path selection from a terrain model acquired from sensor data (Biesiadecki et al. 2005). The main sensor suite used for terrain assessment for the MERs has been passive stereo vision (Wright et al. 2005). The models obtained through stereo imagery are used for both automatic terrain assessment and visual odometry. Path planning is based on a variant of D* (Carsten et al. 2008) that facilitates efficient replanning. Due to high computation load visual odometry is rarely used on the MERs; a more efficient algorithm was proposed for the Mars Science Laboratory mission that launched on November of 2011 (Johnson et al. 2008).

In the case of automatic terrain assessment, the raw data in the form of a cloud of 3D points is used to evaluate the traversability of the terrain immediately in front of the rover, defined as a regular grid of square patches. In the case of visual odometry, the model is used to identify and track features of the terrain to mitigate the effect of slip (Howard and Tunstel 2006).

Field trials on a Mars analog site were presented by Barfoot et al. (2011) comparing different guidance, navigation, and control (GN&C) approaches for a ground ice prospecting mission to Mars. The main sensing modality was again

stereo vision, and a teach and repeat path planning strategy was applied.

The problem of autonomous long range navigation is also very important in terrestrial settings. The DARPA grand challenge in 2005 resulted in several vehicles traveling 132 miles over desert terrain (Montemerlo et al. 2006). The majority of the contestants used a combination of multiple LIDAR, vision, and RADAR sensors. Similar work involved traverses on the order to 30 km in the Atacama desert (Wettersgreen et al. 2005) using vision. More recently the “Leaving Flatland” project (Rusu et al. 2009) presented a vision based scheme for a hexapod walking robot. LIDAR sensors have also been used successfully for 3D mapping of underground mines (Silver et al. 2006). In addition to digital elevation maps, semantic labels were attached to the different areas which facilitated gait selection. Terrestrial navigation has vast literature which is beyond the scope of this paper; please refer to (Kelly et al. 2006) for a discussion of the many challenges and additional related work.

For our work, we have been using a laser range sensor (LIDAR) as the main sensing modality for many reasons: among others, our mobility platform has very low ground clearance. A LIDAR sensor is capable of providing range data to build terrain models with 1–2 cm accuracy. Such accuracy would be difficult to attain with most stereo vision systems over the full range of measurement. Such accuracy is also very important for the scientific return of the mission i.e. identifying rock formations. In addition, LIDAR sensors return accurate geometric information in three dimensions in the form of a 3D point cloud without requiring additional processing. Finally, since they do not rely on ambient lighting, we do not have to address the problems arising from adverse lighting conditions.

3 Overview

The goal of our work is to navigate autonomously from the current position to an operator-specified location which lies

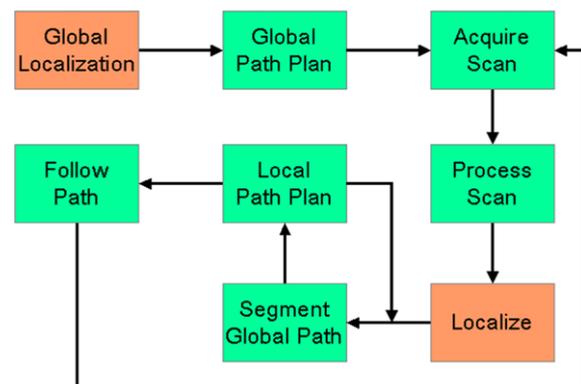


Fig. 2 The main components of the autonomous over-the-horizon navigation framework



Fig. 3 (a) The ILRIS 3D sensor on top of the modified P2-AT robot. (b) The SICK sensor on a pan-unit mounted on the same robot

beyond the sensing horizon of the rover. In order to achieve this goal several components need to be developed, tested and integrated. Figure 2 presents a schematic diagram of the different components. We operate under the assumption that a global map is available from satellite imagery, previous missions, or from data collected during entry descent and landing (Mourikis et al. 2007). For all the experiments a global map with one meter resolution was used. At top level, the rover uses the global map to plan a path from its current position to an operator-specified location. The rover collects the first local scan using its LIDAR sensor, then the global path is segmented successively using the locally collected scans; each time an optimal trajectory is planned through the ITM representation of the local scan. Finally, the rover uses the local path to navigate to the next way-point. At the current state, the pose estimation from the IMU and the odometer, combined with the trajectory length in the order of ten meters allows to safely navigate in open loop without relocalizing between successive scans.

3.1 Experimental setup

Two different LIDAR sensors have been used for our experiments. Initially, we used an ILRIS 3D unit from OPTECH² with a range of over 1 km but with a field of view (*fov*) limited to 40°; see Fig. 3a. Afterwards, a new sensor developed in-house was used. The new sensor of the robotic platform is a LIDAR system based on the popular SICK³ LMS-200/291. Our implementation has the SICK sensor installed sideways (scanning vertically, 180° from bottom to top) on a rotary table panning horizontally over the full 360°

range (Lamarche 2009). A particular feature of the implemented architecture is its modularity. A Rabbit Semiconductor micro-controller is used to coordinate the low level control of the SICK sensor and the rotary table via their respective serial ports. An Ethernet port from the same micro-controller is then used as interface to the rover's on-board network/computer. Such interface makes the sensor integration to any system using a standard Ethernet network fairly straightforward. The range of the new sensor is maximum 30 m. Figure 3b shows the new sensor mounted on our mobile platform during an experiment.

At the time the experiments described in this paper were executed, CSA's MET was a 60 m by 30 m testing area designed to emulate a broad variety of Martian topographies. The terrain includes plains, a hill, a canyon and rock fields of varying density. Figures 1 and 3 show the robot in different locations on the terrain, while Fig. 6 presents a complete model of the terrain. The terrain is covered with sand and rocks. It emulates the topography of some areas on Mars but not the geotechnical properties of the soil.

The mobile robot base used to conduct the experiments is a P2-AT mobile robot from Adept MobileRobots;⁴ the robot is shown in Fig. 1. The P2-AT is a skid-steered four-wheeled robot. The robot comes equipped with motor encoders for odometry. It is also equipped with a 6-axis Inertial Measurement Unit (IMU) from Crossbow.⁵ The IMU provides angular velocity readings through three solid-state gyroscopes, and linear acceleration readings through three accelerometers. The gyroscopes are used to correct the odometry readings, which are very sensitive to slip during rotations. The accelerometers are used to reset the roll and pitch components of the robot's attitude by measuring the components

²<http://www.optech.ca/>.

³<http://www.sick.com>.

⁴<http://www.mobilerobots.com>.

⁵<http://www.xbow.com/>.

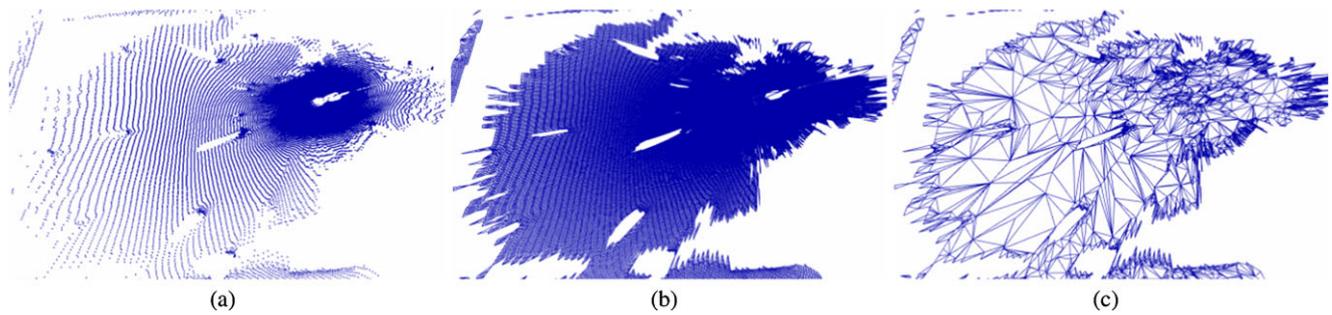


Fig. 4 Sample sensor data from the 360° LIDAR sensor: (a) The raw point cloud. (b) Delaunay triangulation in polar coordinates. (c) The decimated irregular triangular mesh (ITM)

of the gravitational acceleration vector. The robot is also equipped with a digital compass (TCM2 from PNI Corporation⁶). The compass is used to reset the yaw component of the robot's attitude. The TCM2 is only used at rest since the motors induce magnetic fields that corrupt the sensor's readings while running. On Mars the compass would be replaced with a different orientation sensor, such as a Sun-sensor because there is no planet wide magnetic field. The data from the wheel odometry, the IMU and the digital compass are fused together to provide a six degrees of freedom (DOF) state estimate.

4 Terrain modeling

With both LIDAR sensors, the scans are acquired at constant angular steps. Therefore, the sensor's raw data output is structured as a triplet (θ, ϕ, ρ) , representing azimuth, elevation and range respectively. Figure 4 shows a point cloud resulting from a single scan at CSA's MET, converted to Cartesian coordinates for visualization. Because the scans are taken at constant angular resolutions, the resulting point clouds, once viewed in their Cartesian projection, have a very uneven point density distribution. As seen in Fig. 4, the point density is high near the sensor and decreases rapidly along the radius distance from the sensor.

In order to perform path planning, the terrain surface must be reconstructed from the acquired point clouds. Representing the surfaces as irregular triangular meshes (ITM) (Fowler and Little 1979) is a natural choice since data are collected from unstructured outdoor scenes. Irregular triangular mesh representation inherently supports concave geological structures like overhangs and caverns, unlike the well known digital elevation map representation. While many approaches were developed over the years, one fact remains: correctly meshing arbitrary 3D point clouds is never trivial. There does not seem to exist a generic meshing technique

that will properly work for any arbitrary 3D data set. One of the main problems typically resides in identifying the proper neighboring relationships among the points. Failure to do so typically results in triangles intersecting each other in the mesh, therefore not corresponding to a "real" surface. When the point density is not constant, as in our case, this issue becomes even more difficult since many meshing algorithms make assumptions about the point distribution at some level, in order to determine these neighboring relations.

While the acquired laser points are in three dimensions, the raw data from a single scan is, in reality, a 2.5D data source. That is, for any angular position set (θ, ϕ) , there can only be a single range (ρ) measurement. This is a key feature used to generate meshes that properly implement the 3D point neighboring relationships. It turns out that in the raw data, points that are neighbors in the (θ, ϕ) plane are inevitably neighbors in the "real world" when the scan is taken. We can therefore mesh the raw data by simply applying a 2D Delaunay triangulation algorithm to the (θ, ϕ) coordinates, initially ignoring the range readings. Converting the points of the resulting mesh to Cartesian coordinates gives us a proper 3D mesh with accurate neighboring relationships. That is, a mesh free from any intersecting triangles.

However, we now have a mesh that does not contain any holes, even though the real surface sensed by the LIDAR usually has some discontinuities due to the sensor's low angle of incidence; see Fig. 5a. These shadow regions exist whenever there is an object in front of another. Triangles inside these shadow regions must be identified and removed from the mesh because they do not model an existing surface. A second type of undesired triangles is also created by the Delaunay triangulation algorithm producing a convex mesh when the acquired point cloud has concave sections in its (θ, ϕ) planar projection; the meshing algorithm will fill these areas. This happens for example when there are two contiguous hills at the horizon. The concave portion between the hills is filled, resulting in oversized triangles linking both summits. These visual artifacts, referred to as frontier triangles, must also be removed.

⁶<http://www.pnicorp.com/>.

In order to deal with the above mentioned artifacts, three filters were implemented. Experiments showed that while all filters are relatively efficient, none of them is sufficient by itself to remove all the undesired triangles from any given mesh. Combining the three filters provides the best performance, each filter picking up where the others fail. For every triangle in the mesh, the first filter applies a threshold on the ratio between its closest and farthest vertices, relative to the sensor's origin. Since raw data directly provide the distance (range) for every point, this is quickly computed. This filter mainly removes the typically elongated shadow triangles. The second filter applies a threshold on the triangle's perimeter, targeting the frontier triangles, which have huge perimeters in most cases. Finally, the last filter removes any triangle having an incident angle, relative to the sensor's line of sight, smaller than a specified threshold. This targets some problematic shadow triangles that might escape the first filter. Finally, in order to decrease the large number of triangles, in the order of 150,000 triangles, a mesh decima-

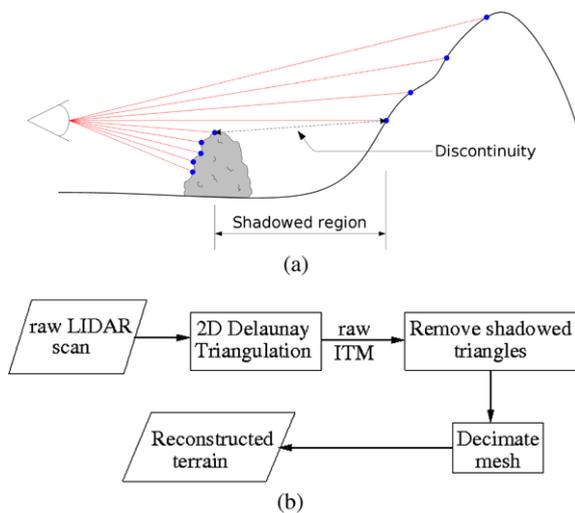


Fig. 5 (a) Side view of a LIDAR scan and its neighboring relationships presenting a discontinuity. (b) Flow diagram of surface reconstruction process

tion algorithm is applied on the reconstructed surface. Figure 5b outlines the meshing algorithm. Figure 4a presents the 2.5D point cloud, it is clear that even small obstacles create long shadows with no measurements inside. The Delaunay triangulation in polar coordinates is used which preserves the shadows; see Fig. 4b. As a result there are no surface representations in the areas for which there are no measurements. Finally the full ITM is decimated by removing triangles that are nearly co-planar, while ensuring that the distance from the original points to the resulting mesh never exceeds a selected threshold; see Fig. 4c. The implementation of the terrain modeling and decimation using a triangular mesh is done using the Visualisation Toolkit (Kitware inc 2005) libraries. Two different LIDAR sensors have been used by CSA, with different range, field-of-view, and accuracy characteristics. More than two hundred scans from the two LIDARS have been collected during our experiments. Each scan contains 111,000 (SICK based LIDAR) or 31,200 (ILRIS 3D) points on average depending on the sensor. The employment of ITMs for terrain modeling maintained the high levels of accuracy while at the same time reducing the data volume by 90–95 % (Rekleitis et al. 2008b).

During testing using the ILRIS sensor, 96 scans were collected from many different locations; see Fig. 6a. Due to the limited field-of-view *fov* of the sensor, often, from the same position several scans in different orientations were collected. Later on 105 scans were collected using the SICK-based sensor, shown in Fig. 6b. Even though the two sets of scans were collected from different LIDAR sensors, the decimation ratios achieved were comparable. Table 1 presents the cumulative results for decimation ratios of 80 %, 90 %, and 95 %. With acceptable error as low as 1.5 cm, decimation ratios of up to 94.9 % were achieved on average. It is worth noting that the SICK-based scans had a 360° *fov*, as such there were on average 111,000 data-points per scan compared to 31,200 points, on average per scan, for the 40° *fov* scans collected with the ILRIS 3D LIDAR. High decimation ratios for fixed error translate to great savings in on-board storage and bandwidth usage.

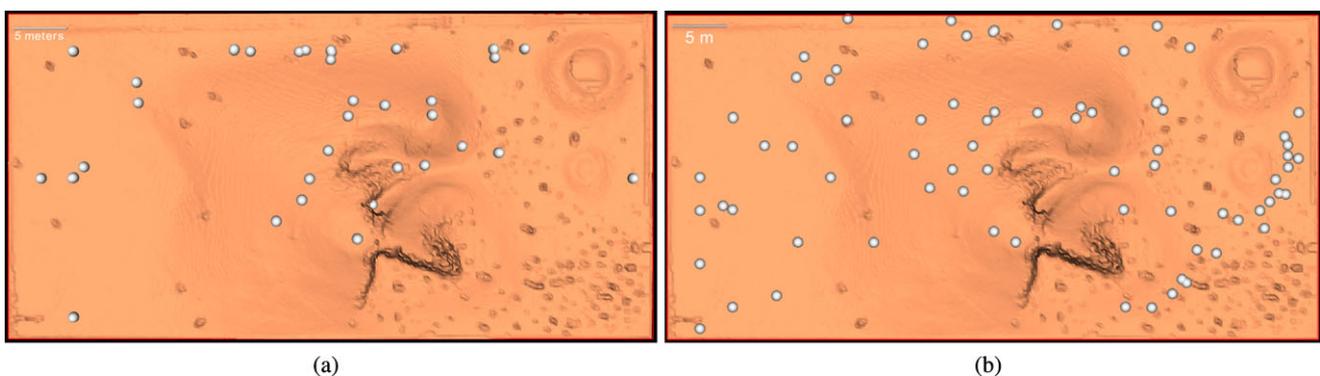


Fig. 6 (a) The locations from where 96 scans with a 40° *fov* were collected. (b) The location of the 105 scans collected using the 360° *fov* sensor

Table 1 Properties of decimated terrain scans for the testing using the ILRIS and the SICK based LIDARs. Acceptable error 1.5 cm

	Original scans	Target decimation ratio					
		80 %		90 %		95 %	
		Number	Real %	Number	Real %	Number	Real %
ILRIS Points (mean)	31200	6530	79.00 %	3440	88.86 %	2090	93.09 %
ILRIS Points (std)	7840		0.74 %		1.16 %		2.31 %
ILRIS Triangles (mean)	61700	12300	80.00 %	6190	89.91 %	3590	94.01 %
ILRIS Triangles (std)	15800		0.00 %		0.75 %		1.90 %
SICK Points (mean)	111000	23400	78.91 %	12500	88.72 %	6700	93.69 %
SICK Points (std)	10300		0.23 %		0.27 %		0.28 %
SICK Triangles (mean)	216000	43300	80.00 %	21600	90.00 %	10900	94.98 %
SICK Triangles (std)	20100		0.00 %		0.00 %		0.14 %

The ITM preserves the science content from the topographical data, while capable of modeling concave geological structures like overhangs and caverns. Please refer to Rekleitis et al. (2007), Gingras et al. (2010) for a discussion of various LIDAR modeling methods. It is worth mentioning though the recent approach which combines digital elevation maps with multiple layers (Triebel et al. 2006) as it models concave structures. However, it does not lead easily to path-planning. More recent approaches use techniques from machine learning (Vasudevan et al. 2010) or from fuzzy logic (Mandow et al. 2011) for terrain modeling in order to deal with contradictory readings.

4.1 Comparing ITMs to grid-based representations

A popular approach to terrain modeling and path planning is grid-based representation, due to its simplicity. However, grid-based approaches suffer from either low accuracy or excessive memory requirements. Next we present a qualitative comparison between the ITM approach and the basic grid-based approach.

Let us consider a scan like the one presented in Fig. 4, with a sensor range of 15 m; a rectangular grid generated from this scan would cover a square of 30 m × 30 m. Using the average number of points for the 360° sensor as in Table 1, with 95 % decimation ratio (6700 points), the resulting grid cell size would be 0.36 m. When mapping terrain features of varying slope, between 30° to 85°, a grid-cell size of 0.36 m would result in a discretization error between 0.1 m and 1 m, compared to the 0.015 m reported in Table 1. More formally, for an obstacle with slope θ and grid cell size d , the elevation h can be calculated as follows: $h = d \tan(\theta)$; the discretization error is $h/2$.

In addition, it is worth noting that any regular grid would generate fictitious data points inside shadow areas, where no real information is available, as well as at the corners of the grid square which are out of sensing range; see Fig. 7. In

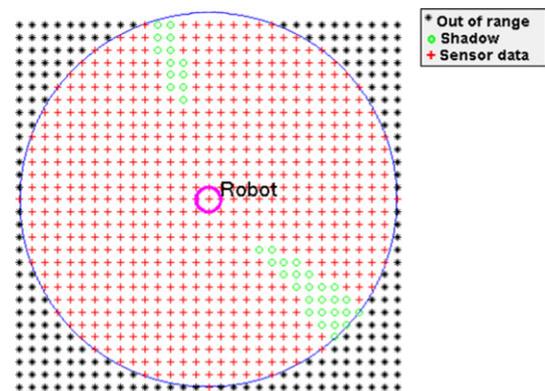


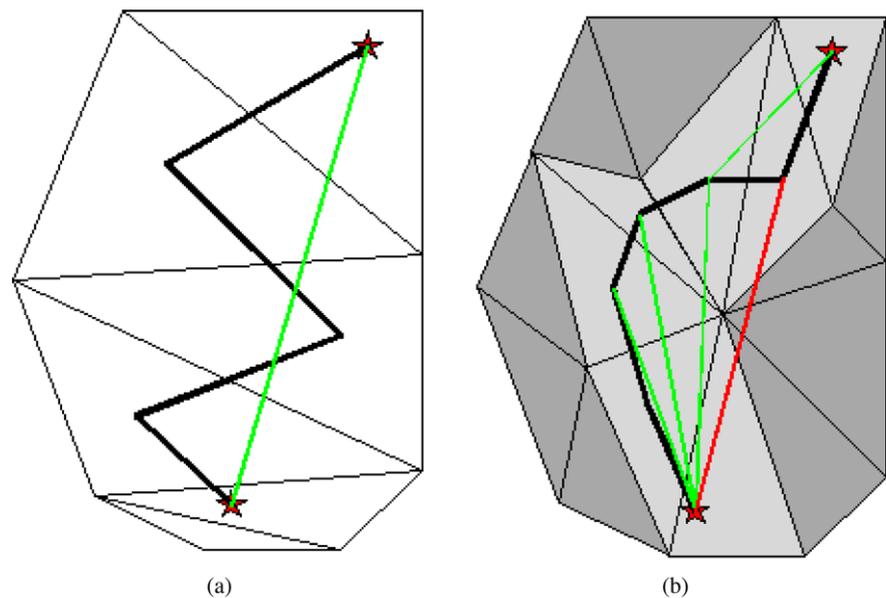
Fig. 7 Grid representation for a single scan, with out-of-range and shadow data points marked

an application domain such as planetary exploration, where resources are scarce and accurate models of the environment are scientifically valuable, grid-based approaches are worse choices than ITMs.

5 Path planning on ITM

One of the advantages of the ITM representation is that it is well suited to path planning. ITM contrary to grid like structures contain all the geometric information available from the sensor in a single data structure; extracting distance and slope information in an as-needed basis, is trivial. In contrast the same information can be encoded in a grid like structure, but it will have to be precomputed for the whole grid regardless if it would be needed. In particular, the triangles in the mesh form individual cells. While traversing the terrain, the robot moves from one cell to another by crossing their common edge. The ITM representation can therefore easily be transformed into a graph structure where the cells are the graph nodes and the common edges between cells are the

Fig. 8 (a) Example of path planned through an irregular triangular mesh. (b) Example of trajectory simplification on an irregular triangular mesh



edges between the nodes of the graph. The path-planning problem is then formulated as a graph search problem.

The results described in this paper were obtained initially using Dijkstra's graph search algorithm (Dijkstra 1959) with several different cost functions taking into account distance traveled, terrain slope, and terrain roughness. With the transition to the 360° *fov* scans the breadth first nature of the Dijkstra's algorithm resulted in excessive computations, and an A* based search algorithm was used. One of the main advantages of graph search techniques is that they do not get stuck in local minima: if a feasible path exists between any two locations, graph search algorithms will find it. In addition, given any cost function, Dijkstra's algorithm always returns the lowest cost solution between any two locations.

It should be noted that the output of the graph search algorithm is a series of cell identifiers. When traversed in the given order, the cells will lead the robot from start to destination along a path that is deemed safe and optimal according to the given cost function. The robot's guidance and motion control algorithms, however, require a trajectory composed of a series of points in 3D space. The easiest way to convert cell ID's to 3D points is to use the geometric centers of the cells as trajectory points. The trajectory is then the list of the center points of all cells in the list generated by the graph search algorithm. This results in a trajectory that zigzags unnecessarily between cell centers; see Fig. 8a. It is therefore necessary to smooth out the resulting trajectory by removing superfluous way-points in the trajectory. The trajectory simplification algorithm first defines a safety corridor as the set of all cells in the path generated by the graph search algorithm. Each of these cells has been identified by the planner as a safe area on which the robot can tread. The trajectory generation algorithm then assigns a way-point to the geometric center of every cell in the path. The simplification

algorithm removes intermediate points in the trajectory and verifies whether the straight-line segment joining the two points on either side of the removed way-point stays on the safety corridor. This procedure is applied iteratively starting from the initial location of the robot. Points are removed as long as the safety corridor constraint is not violated. At this point, the algorithm is re-started from the location of the way-point that could not be removed and stops when reaching the final destination. Figure 8b shows an example of the trajectory simplification algorithm. The light grey cells are the safety corridor, the thick black line is the raw path joining the geometric centers of all cells in the path. The green lines show the successive steps in the simplification of the trajectory by the elimination of superfluous way-points. The red line is a case of a simplification that would lead to the violation of the safety corridor constraint.

In this context, the usage of ITM introduces additional challenges. First, on flat terrain, the cells are relatively large, as can be observed by comparing Figs. 8a and 8b. Therefore, although large cells are preferable for safety reasons, a cost function taking only distance traveled into account would unduly penalize traversal through large cells because the raw path zigzags between cell centers. On the other hand, on rough terrain, the cells are much smaller and the resulting safety corridor can be very narrow, hence more difficult to navigate.

In addition, the trajectory simplification algorithm, by design, simplifies the trajectory until it skims the boundaries of the safety corridor: the resulting trajectory can therefore skim obstacles. If the width of the robot is not considered, the planned trajectory will result in a collision between the robot and obstacles in the environment.

Figure 9 shows a path that was planned in a typical terrain scan obtained using a LIDAR range scanner in CSA's MET.

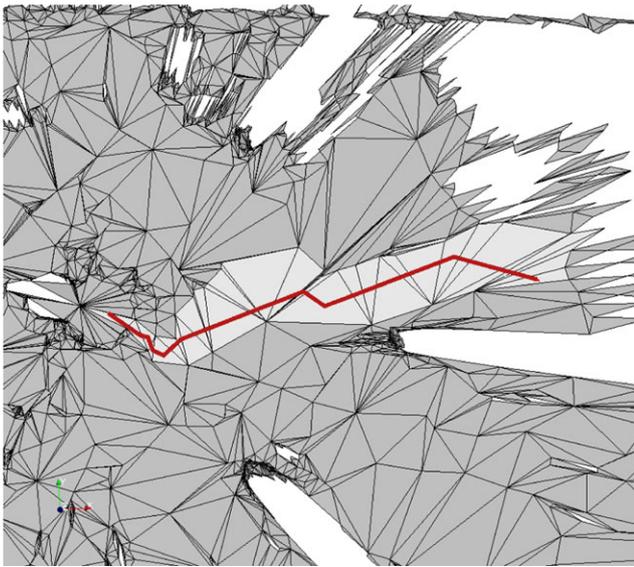


Fig. 9 Results of path planner on typical irregular triangular mesh

The scan was acquired from the start point located at the left end of the planned trajectory (the red multi-segmented line). The figure clearly shows that the trajectory remains within the bounds of a safety corridor without going through the center points of every cell in the path.

These results were obtained using Dijkstra’s graph search algorithm with the following cost function to compute the cost of traveling from cell i to cell j :

$$Q = \|\mathbf{x}_j - \mathbf{x}_i\| \alpha \beta \gamma e^{\frac{\|\mathbf{x}_j - \mathbf{x}_i\|}{A_i + A_j}} \quad (1)$$

where \mathbf{x}_i and \mathbf{x}_j are the geometric centers, and A_i and A_j are the areas of cells i and j respectively. The exponential term is used to encourage the path to cross wide cells instead of long thin cells. The parameters α and β are penalty multipliers to take into account the slope of the terrain, and γ is a penalty multiplier for the terrain roughness. These parameters are computed taking into account the footprint of the robot.

The footprint of the robot is defined as $\mathcal{C} = \{c_1, c_2, \dots, c_m\}$, the set of all cells with at least one vertex within a distance r from \mathbf{x}_j ; where r is a safety parameter. The average normal of the terrain within the footprint is defined as:

$$\bar{\mathbf{n}} = \frac{\sum_{k=1}^m A_k \mathbf{n}_k}{\sum_{k=1}^m A_k} \quad (2)$$

where A_k and \mathbf{n}_k are the area and the unit normal of cell k . The cross-track vector and along-track vector are then computed as:

$$\bar{\mathbf{c}} = \bar{\mathbf{n}} \times (\mathbf{x}_j - \mathbf{x}_i) \quad (3)$$

and

$$\bar{\mathbf{a}} = \bar{\mathbf{c}} \times \bar{\mathbf{n}} \quad (4)$$

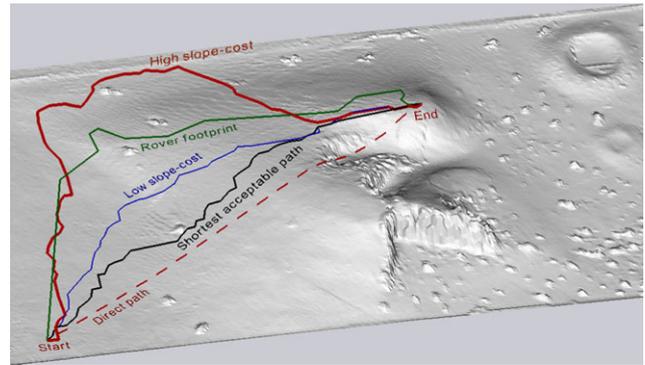


Fig. 10 The model of the Mars terrain and several global paths planned using different cost functions

The cross-track slope angle and the along track slope angles are then computed as:

$$\phi = \left| \text{atan2}(\bar{\mathbf{c}}_z, \sqrt{\bar{\mathbf{c}}_x^2 + \bar{\mathbf{c}}_y^2}) \right| \quad (5)$$

and

$$\theta = \text{atan2}(\bar{\mathbf{a}}_z, \sqrt{\bar{\mathbf{a}}_x^2 + \bar{\mathbf{a}}_y^2}) \quad (6)$$

The values of ϕ and θ are then used to compute the slope penalty parameters in (1) as follows:

$$\alpha = \begin{cases} k_a \frac{\theta}{\theta_{\max}} & \text{if } \theta_{\min} < \theta \leq \theta_{\max} \\ \infty & \text{if } \theta < \theta_{\min} \text{ or } \theta > \theta_{\max} \end{cases} \quad (7)$$

$$\beta = \begin{cases} 1 & \text{if } \phi \leq \phi_{\max} \\ \infty & \text{if } \phi > \phi_{\max} \end{cases} \quad (8)$$

where k_a is a scaling parameter and θ_{\min} , θ_{\max} , and ϕ_{\max} are platform specific threshold values.

The roughness penalty factor γ is computed by evaluating the distance between every vertex of the *ITM* contained in the rover footprint and the plane defined by the average normal $\bar{\mathbf{n}}$ and \mathbf{x}_j , the center point of cell j . The maximum distance between the vertices in the footprint and the plane is computed as:

$$\delta = \max(|\bar{\mathbf{n}} \cdot (\mathbf{p}_k - \mathbf{x}_j)|) \quad \forall k / \|\mathbf{p}_k - \mathbf{x}_j\| < r \quad (9)$$

where \mathbf{p}_k is any vertex of cell k . The roughness penalty factor γ is then computed from the maximum deviation from the average plane. Experimentation using realistic terrain models acquired using the 3D sensor in the CSA’s MET has shown that it is sufficient to compute γ in the following manner:

$$\gamma = \begin{cases} 1 & \text{if } \delta \leq \delta_{\max} \\ \infty & \text{if } \delta > \delta_{\max} \end{cases} \quad (10)$$

Figure 10 demonstrates the effect of different cost functions on the planned path. The paths were planned using the coarse model of the Mars terrain. The dashed (red) line shows the direct path from start to end without any concern

about the slope or the roughness of the terrain. For the rest of the paths only acceptable triangles were considered, that is, triangles with an absolute slope of less than 35 degrees. The (green) line labeled “Rover footprint” plans a path taking into account the rover footprint avoiding high-slope areas and showing a slight preference for flatter terrain. Finally, three paths are planned without taking into account the rover’s footprint. The shortest acceptable path planned (black line) is close to the direct path while avoiding the cliff due to the infinite cost of high slope triangles. The second path (blue line) is planned with low slope-cost for acceptable slopes, and the third path (red line) weights flat terrain much higher than distance traveled, thus traveling around the terrain over the most flat areas. While parameters selected by an expert operator have been used in most rough-terrain navigation approaches, newer research shows promising results using parameters learned by demonstration (Silver et al. 2010). We expect future work to evolve towards automatic cost function learning.

As mentioned earlier, one of the issues encountered during our field-testing was due to the fact that the environment sensor has a 360° *fov* and the Dijkstra’s graph search algorithm is a breadth-first algorithm: it grows the search space from the start location irrespective of the target destination. The planner ends up spending much precious time searching in the direction away from the destination. The A* graph search algorithm was used in an attempt to reduce the computation time. Experimental results indicate that, using typical terrain models, the computation time can be accelerated by a factor of three up to six times on the same computer. The path-planner using A* was also tested off-line using the collected scans from the SICK based sensor. Random destination points were selected at five and ten meters from the location the scan was originated for all 107 scans. The computation time was on average 14 seconds for the destinations at five meters, and 25 seconds for ten meters. The proposed planning method was very efficient, the paths were computed in seconds using ITMs with several thousand triangles, and the computed paths were on average 25 % longer than a straight line between start and destination (Rekleitis et al. 2008b). As noted earlier, a path was always found if a feasible path, for a given cost function, existed. For an in-depth discussion of the CSA’s path-planning approach, including the implementation of different cost functions, please refer to Rekleitis et al. (2008a).

6 Path-segmentation

The path planning approach described above can be used in several different circumstances. In the proposed framework, the fast distance-and-slope cost function is used for global path planning over the low resolution global map; see

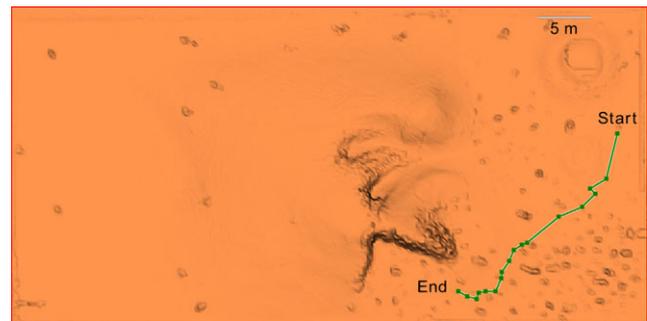


Fig. 11 Global path planned using the low resolution map of the Mars-like terrain at CSA

Fig. 11. The generated path while avoiding large slopes and big obstacles leads over smaller obstacles, not discernible at that resolution. This global path provides a general guideline for traveling from the starting position to the operator-selected destination, beyond the robots sensing horizon. The global path is then divided into smaller segments by selecting a series of way-points. These way-points act as intermediate destinations, each of them inside sensing distance from the previous one. The more accurate rover-footprint cost function in conjunction with the A* cost function is used to plan a path between way-points, using the high-resolution map constructed using a local scan obtained from the LIDAR.

The first step in segmenting the global path is to consider what is the appropriate effective range of the local scan.⁷ In the current setup, the effective range of our sensor is between ten to fifteen meters; further than that the angular resolution is too coarse for safe operations. When the sensed terrain is flat and sparsely populated with obstacles a range of ten meters is used. When a large number of obstacles surrounds the rover, the sensed scan is full of shadows, areas with no data located behind obstacles. In such cases, the effective range is usually set to four meters; see Fig. 12a for such a terrain where the four meter range is indicated by a yellow circle. The intersection point between the global path and a sphere centered at the current position of the rover with a radius of four meters is calculated; see Fig. 12b. The intersection point is used as a starting point in search of a suitable triangle in the area, at most one meter from the starting point; see blue circle in Fig. 12b. All triangles that have at least one corner less than a meter away from the intersection point are considered and the bigger one that has an acceptable slope is selected; highlighted in dark red in Fig. 12b. When the destination way-point is selected, the A* and the rover-footprint cost function are used to plan a local path,

⁷When the local scan spans 360° , as with the current sensor, only range is considered. When the scan has a limited field of view, see Rekleitis et al. (2007), then the global path has to be segmented against the field of view of the sensor.

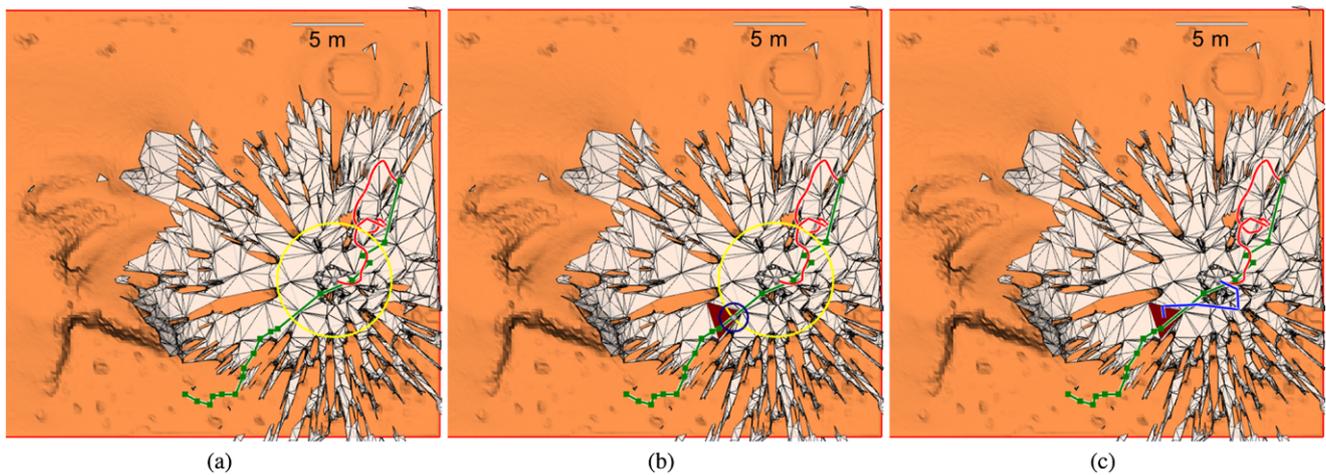


Fig. 12 (a) The global path (*green*), the local scan, the trajectory up to this point (*red*), the current position, and the four meter range the local path planner would use (*yellow circle*). (b) The intersection point

between the global path and the four meter range; the local area from where the destination triangle (*red*) is selected (*blue circle*). (c) The local path planned (*blue*)

through the decimated ITM, between the current pose and the destination way-point; Fig. 12c.

7 Navigation

The algorithm presented in the previous section produces a set of straight-line segments representing the path. The piecewise linear path is then interpolated using a Catmull-Rom spline curve,⁸ with one way-point every half a meter, in order to produce a locally smooth trajectory. The resulting trajectory is given as input to a low-level motion controller based on a discontinuous state feedback control law, initially proposed by Astolfi (1999). There were a few improvements introduced in order to increase the capabilities of our platform. When the IMU detects that the rover is moving uphill, the guidance controller increases the speed in order to gain momentum. Furthermore, the guidance controller was tuned to reduce the turning in favor of forward motions.

8 Data management

During planetary exploration, an autonomous rover is required to store and handle huge amounts of sensor data, results from scientific experiments and other relevant information. This information is typically geo-referenced to maps describing the world. As the number of maps increases, simple planning operations become more complex because different map combinations can be used to plan the path to the goal. Data from different sensors, such as LIDAR range

finders, thermal cameras, images from monocular and stereo cameras, etc., produce different maps. Moreover, data from the same sensor collected at different times and from different locations produce maps of varying resolution and fidelity. Effective management of the different maps is addressed by CSA by the introduction of an Atlas framework.

The implemented Atlas framework supports typical mapping, localization and planning operations performed by a mobile robot. Central to the development is the ability to provide a generic infrastructure to manage maps from multiple sources. Changes in the map processing algorithms are transparent to the Atlas implementation. In the same way, using maps from different sensor sources or types becomes transparent to the planning algorithm.

The main features of the Atlas management system are the capability to dynamically manage a variety of data formats, the handling of uncertainty in the spatial relationship between the maps, the capability to provide series of maps linking two locations in planning operations, and the functionality to correlate maps in localization operations (Nsasi Bakambu et al. 2006). Figure 13 presents representative data sets stored in the Atlas used at CSA for the experiments leading to the Avatar Explore mission (Martin et al. 2008). Figure 13a contains a single infrared image which is used to detect geologically interesting locations for closer inspection by the rover. For this experiment, a heat source was buried in the terrain to provide a clearly identifiable target. A 360° LIDAR scan from a single location is displayed in Fig. 13b; such scans are used for safe path-planning by the rover. Finally, Fig. 13c shows a uniform mesh representation of the testing terrain. This representation is the result of an off-line integration process that combines several data sets and can be uploaded on the rover by remote operators.

⁸See <http://www.mvps.org/directx/articles/catmull/>.

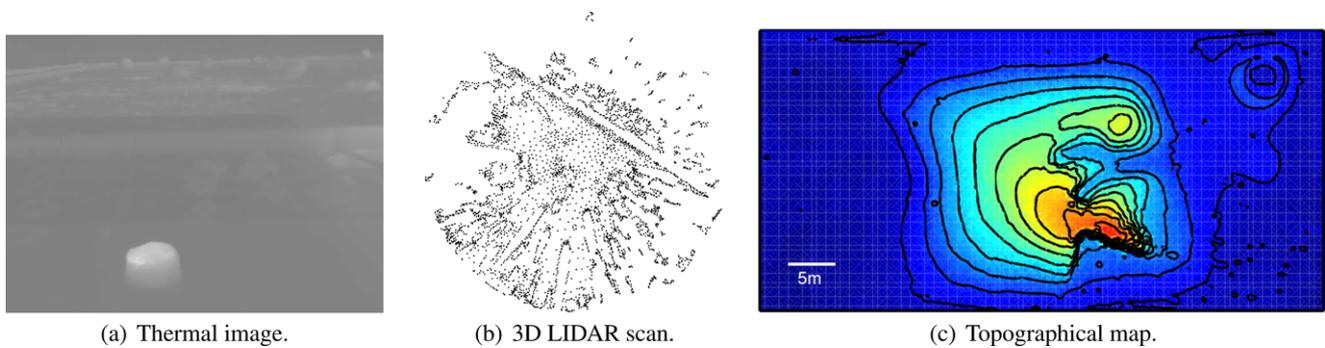


Fig. 13 Maps from different sensors are stored in the Atlas framework: (a) Image from a thermal camera. (b) A single scan from a 360° LIDAR sensor. (c) Topographical map of fixed resolution containing information of a larger region

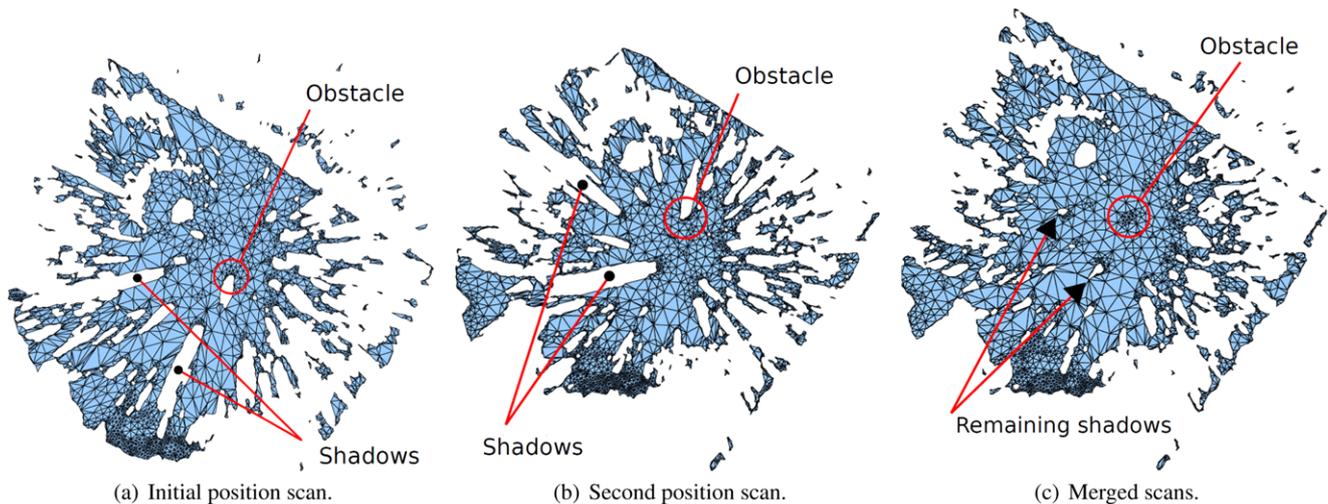


Fig. 14 (a, b) Two LIDAR scans of an area with several obstacles. (c) The Irregular Triangular Mesh resulting from localizing with Kd-ICP and merging the two scans, eliminating most of the shadows

This data management tool opens new opportunities in the development of autonomous navigation schemes. For instance, maps are never fused by the system, thus permitting registration of the maps at any time. The system makes it possible for two given maps to have multiple mutual spatial relationships such as an estimation of their relative pose from the wheel odometry and visual registration. The Atlas system provides the mechanism to select/merge these multiple estimation relations in order to get the best estimation possible. The system being modular, it is possible to provide different cost functions in order to obtain more accurate use of these multiple relations. Finally, by using relative relations instead of global relations, the Atlas system automatically propagates to neighboring maps any improvements/changes deriving from the updated relation between two maps. One of the important uses of the Atlas framework is to provide all available information of a specific area of interest. In particular, in presence of obstacles, most LIDAR scans are plagued by long shadows that make path planning challenging; see Figs. 14a, b. By requesting one or more ad-

ditional data sets from the Atlas for the vicinity of the area of interest, the rover is capable of merging the retrieved scans and obtaining a new mesh more suitable for path-planning, see Fig. 14c, where most of the shadows are eliminated. The search query of scans covering a specific point is utilized for retrieving relevant scans.

9 Experimental results

The experiments were run at the CSA's Mars emulation terrain (MET). A large number of experiments were conducted using the ILRIS 3D LIDAR sensor with limited *fov* sensor. Subsequently, the 360° *fov* sensor enabled us to perform several experiments validating CSA's approach to fully autonomous over-the-horizon navigation. The first set of tests conducted was for validating the accuracy of the 3D odometry state estimation algorithm. A large number of closed trajectories was executed varying in size and in location. Consequently, we were able to quantify the error exhibited in

Table 2 Path planning results using all the scans collected using the SICK-based LIDAR. Destinations selected randomly at 5 m and 10 m from the origin of the scan

	Time (sec)	Path length	Euclidean distance	# of path points	Total # of triangles	Max slope	Weighted slope (WS)
5 m (mean)	13.8	6.3 m	5.0 m	10	7744	27.2°	1.8°
5 m (std)	17.8	1.06 m	0.1 m	5.5	2330	22.8°	2.0°
10 m (mean)	25.4	12.70 m	10.0 m	14.5	7550	34.8°	1.3°
10 m (std)	29.8	2.23 m	0.0 m	6.5	1598	22.8°	3.2°

terms of both the length of the traveled trajectory as well as the morphology of the terrain.

A statistical error analysis has revealed that the actual error on position for the closed loop trajectories is on the order of 2.19 % with a standard deviation of 2.25 %, of which approximately 0.5 % is due to the 3D odometry. The high value of the standard deviation is due to the fact that three experimental runs (out of 29) resulted in errors on the order of 7–8 % due to excessive wheel slip.

9.1 Path planning verification

The scans collected during the two testing seasons were also used to verify the quality of our path-planning algorithm by off-line batch testing. For these experiments of path-planning on a single scan, the start position was always assumed to be the position from which the scan was acquired. The end-location was randomly selected to be at five or ten meters from the origin, and inside the boundaries of CSA's MET. If the destination point was unreachable, that is in an area of forbidding slope or with no data, then another point was selected randomly, up to three times. Out of 94 scans with recorded origin, acceptable destinations were found randomly for 82 scans at a distance of five meters, and 68 destinations at a distance of ten meters. Please note, that for every scan only 3 attempts were made to find a destination point randomly. The results are an indication of the challenging terrain where obstacles created shadows with no data and several areas had non-traversable slopes. Every time a destination point was inside the mesh and at a navigable slope, the path planner found a smooth path. Table 2 presents average results for the computed paths for destinations at five and ten meters, respectively. The computation time was on average 14 seconds for the destinations at five meters, and 25 seconds for ten meters. The proposed planning method was very efficient, the paths were computed in seconds using ITMs with several thousand triangles; see Table 2. Furthermore, the ITMs used were created from LIDAR scans of more than a hundred thousand points each; see Table 1 for the decimation rates. The computed path was on average 25 % longer than a straight line between start and destination. It is worth noting that the estimated paths had to negotiate maximum slopes of 30° on average, but the triangles with high slope were always very small.

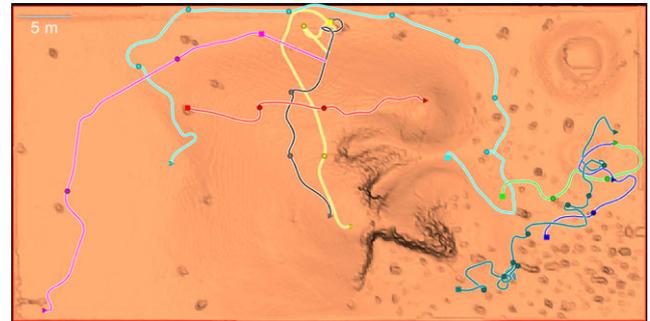


Fig. 15 A model of the CSA's MET together with the recorded trajectories for all the different experiments performed using the SICK-based sensor and the proposed framework

A weighted average slope (*WS*) was calculated by multiplying the slope of each triangle on the path with the area that triangle and then normalized by dividing by the sum of all the areas:

$$WS = \frac{\sum S_i A_i}{\sum A_i} \quad (11)$$

where S_i and A_i is the slope and the area of triangle i respectively. As can be seen by the last column in Table 2, the average slope (*WS*) is in the order of two degrees, as the path planner favored the more level ground.

9.2 Over-the-horizon navigation

The next, and final, phase of testing was comprised by a series of over-the-horizon autonomous navigation traverses that tested the integrated system. Figure 15 presents the trajectories of several experiments of autonomous over-the-horizon navigation over a model of the CSA's MET. As can be seen, the experiments covered all the terrain types represented in MET. In particular long trajectories over flatter terrain with sparse obstacles were traversed first, while the climbing abilities of the mobile platform were tested during the trajectories that appear in the middle of the terrain, where two small hills are located. Finally, the robot was able to navigate autonomously through an area littered with obstacles depicted in the right side of Fig. 15.

Figure 16 presents the results from a representative fully autonomous navigation experiment. First the model of the CSA's MET was used as the global map and the operator entered the global destination, then a simple planner was used

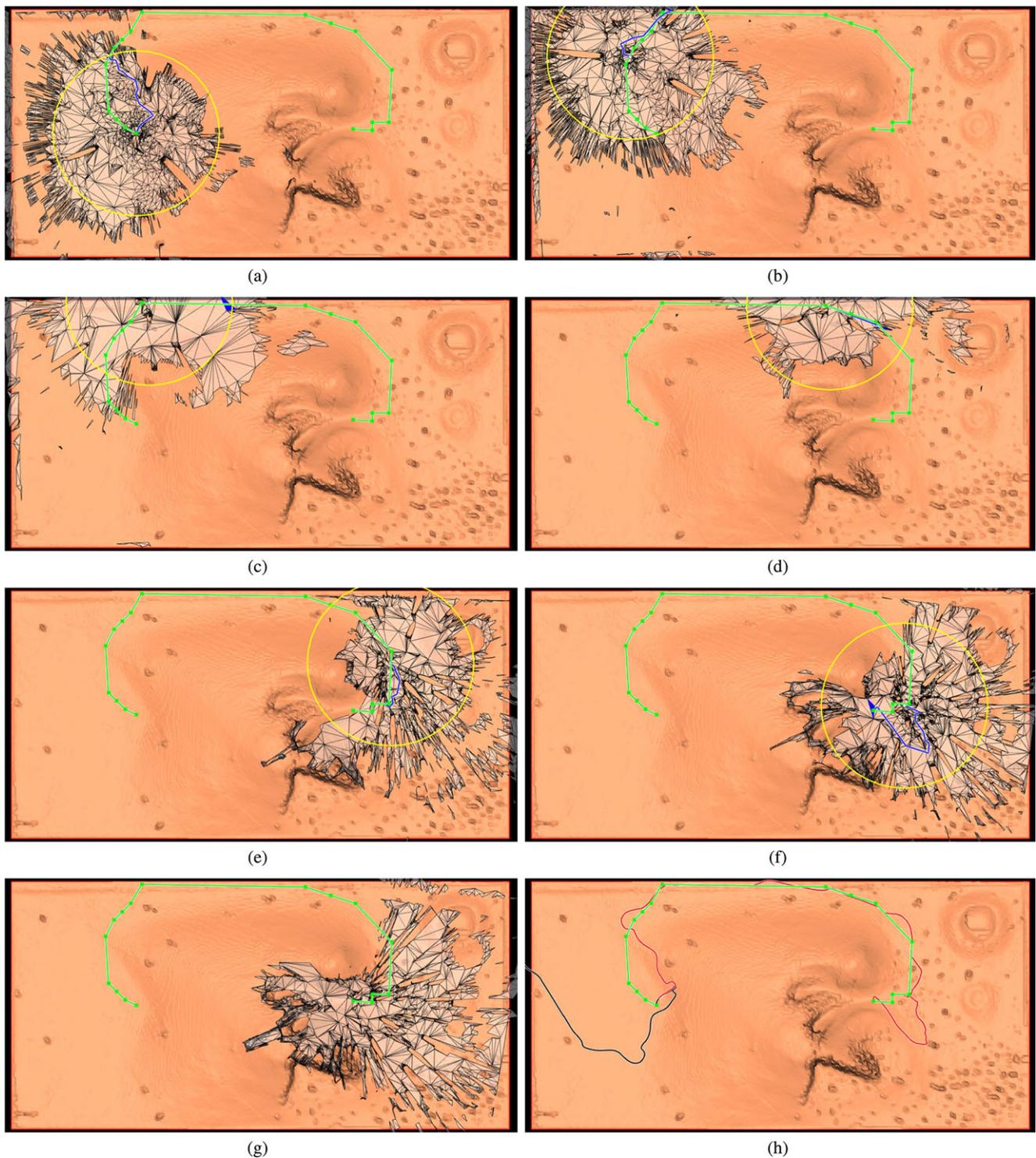


Fig. 16 A sequence of consecutive scans together with the trajectory of the robot. A global model of the Mars emulation terrain (MET) is set as background for reference purposes. **(a)** The global path is displayed (*green line*) together with a single scan, **(b)–(g)** the consecutive scans

can be seen together with the planned path. **(h)** The global path (*green*), the traversed path (*red*) and the odometry (*black*) are displayed on the CSA's MET

to calculate the global path. Figure 16h shows all the paths; the global path is presented as a (*green*) line, and the planned local paths together with the odometric estimates are drawn

as red and black lines respectively. Figure 16a presents the first scan and the first local path. The scan was used to determine the last point in the global path that resides inside it

and is accessible from the start position; the selected point is then designated as the first way-point. It is worth noting that due to the shadows, a point in the global path could reside inside an isolated triangle in which case it would not be reachable when used as a destination for the local-path-planner. The rover planned and executed a successful traversal and reached the first way point, at which step it took the second scan; see Fig. 16b. The global path was used again to determine the next local destination, second way point, and the local path planner was used to plan the second collision free path. Finally when the robot reached the second way point the final destination from the global path plan was reachable and the robot planned and executed the next trajectories; see Fig. 16c–g.

10 Conclusions and future work

In this paper we presented successful autonomous over-the-horizon navigation experiments in a Mars-like terrain. The operator selected a destination way beyond the sensing horizon of our rover and then monitored as the robot selected intermediate destinations, planned a safe path and traversed to the next way-point. The Irregular Triangular Mesh representation was used which enabled us to have a compact yet accurate model of the environment. Path planning is conducted in the ITM terrain model using the A* graph search algorithm using a cost function that takes into account the physical dimensions of the rover and its limitations to traverse rough terrain. A new factor was introduced in the cost function to handle the conditions of ITM where safe terrain cells are typically large in size. Experimental results demonstrating the feasibility of our approach are presented.

Upcoming work includes further research on localization and scan matching. This will enable the rover to re-localize by matching features in successive environment scans. Such an approach has the potential to be computationally less expensive than on-line visual odometry based on stereo camera views. Current work includes a re-formulation of the ITM terrain models to render them more amenable to scan matching algorithms such as the Iterative Closest Point algorithm (Rusinkiewicz and Levoy 2001).

The realization of autonomous navigation on a Mars-like environment, led to the recent mission Avatar-2. In the Avatar-2 mission (Martin et al. 2008; Dupuis et al. 2010), that completed successfully recently, a Canadian astronaut on-orbit in the International Space Station (ISS) communicates and sends high level commands to a rover operating at CSA's MET. The rover collects data from different sensors and sends them back to ISS. This scenario emulates the situation of a human operator on orbit around Mars, controlling a rover on the Martian surface.

References

- Kitware inc. (2005). Visualization toolkits. <http://www.vtk.org>, Website (accessed: September 2005).
- Alami, R., Chatila, R., Fleury, S., Ghallab, M., & Ingrand, F. (1998). An architecture for autonomy. *The International Journal of Robotics Research*, 17(4), 315–337.
- Astolfi, A. (1999). Exponential stabilization of wheeled mobile robot via discontinuous control. *Journal of Dynamic Systems Measurement and Control*, 121–126.
- Bares, J., Hebert, M., Kanade, T., Krotkov, E., Mitchell, T., Simmons, R., & Whittaker, W. (1989). Ambler: an autonomous rover for planetary exploration. *IEEE Computer*, 22(6), 18–26.
- Barfoot, T., Furgale, P., Stenning, B., Carle, P., Thomson, L., Osinski, G., Daly, M., & Ghafoor, N. (2011). Field testing of a rover guidance, navigation, and control architecture to support a ground-ice prospecting mission to Mars. *Robotics and Autonomous Systems*, 59(6), 472–488. doi:10.1016/j.robot.2011.03.004.
- Biesiadecki, J., Leger, C., & Maimone, M. (2005). Tradeoffs between directed and autonomous on the mars exploration rovers. In *Proc. of int. symposium of robotics research*, San Francisco.
- Carsten, J., Rankin, A., Ferguson, D., & Stentz, A. (2008). Global planning on the mars exploration rovers: software integration and surface testing. *Journal of Field Robotics*, 26(4), 337–357.
- Chatila, R., Alami, R., Lacroix, S., Perret, J., & Proust, C. (1993). Planet exploration by robots: from mission planning to autonomous navigation. In *1993 international conference on advanced robotics*, Tokyo (Japan).
- Chatila, R., Lacroix, S., Siméon, T., & Herrb, M. (1995). Planetary exploration by a mobile robot: mission teleprogramming and autonomous navigation. *Autonomous Robots Journal*, 2(4), 333–344.
- Dijkstra, E. W. (1959). A note on two problems in connexion with graphs. *Numerische Mathematik*, 1, 269–271.
- Dupuis, E., Langlois, P., Bedwani, J. L., Gingras, D., Salerno, A., Allard, P., Gemme, S., L'Archevêque, R., & Lamarche, T. (2010). The avatar explore experiments: results and lessons learned. In *Tenth international symposium on artificial intelligence, robotics and automation in space (i-SAIRAS)*, Hokkaido, Japan.
- Fowler, R. J., & Little, J. J. (1979). Automatic extraction of irregular network digital terrain models. In *SIGGRAPH'79: proc. of the 6th annual conference on computer graphics & interactive techniques* (pp. 199–207).
- Gat, E., Desai, R., Ivlev, R., Loch, J., & Miller, D. (1994). Behavior control for robotic exploration of planetary surfaces. *IEEE Transactions on Robotics and Automation*, 10(4), 490–503.
- Gingras, D., Lamarche, T., Bedwani, J. L., & Dupuis, E. (2010). Rough terrain reconstruction for rover motion planning. In *Seventh Canadian conference computer and robot vision*, Ottawa, Canada (pp. 191–198).
- Giralt, G., & Boissier, L. (1992). The French planetary rover VAP: concept and current developments. In *Proc. of the 1992 IEEE/RSJ int. conf. on intelligent robots and systems* (Vol. 2, pp. 1391–1398).
- Goldberg, S., Maimone, M., & Matthies, L. (2002). Stereo vision and rover navigation software for planetary exploration. In *Proc. of IEEE aerospace conf.* (Vol. 5, pp. 2025–2036).
- Hayati, S., Udomkesmalee, G., & Caffrey, R. (2004). Initiating the 2002 mars science laboratory (MSL) focused technology program. In *IEEE aerospace conf.*, Golden, CO, USA.
- Hebert, M., et al. (1989). Terrain mapping for a roving planetary explorer. In *Proc. of the IEEE int. conf. on robotics and automation (ICRA'89)* (Vol. 2, pp. 997–1002).
- Howard, A. M. & Tunstel, E. W. (Eds.) (2006). *Intelligence for space robotics*. San Antonio: TSI Press. Chapter: MER surface navigation and mobility.

- Huntsberger, T., Pirjanian, P., Trebi-Ollennu, A., Nayar, H. D., Ag-hazarian, H., Ganino, A., Garrett, M., Joshi, S., & Schenker, P. (2003). Campout: a control architecture for tightly coupled coordination of multirobot systems for planetary surface exploration. *IEEE Transactions on Systems, Man and Cybernetics. Part A. Systems and Humans*, 33(5), 550–559.
- Ishigami, G., Nagatani, K., & Yoshida, K. (2011). Path planning and evaluation for planetary rovers based on dynamic mobility index. In *IEEE/RSJ international conference on intelligent robots and systems* (pp. 601–606).
- Johnson, A., Goldberg, S., Yang, C., & Matthies, L. (2008). Robust and efficient stereo feature tracking for visual odometry. In *IEEE int. conf. on robotics and automation* (pp. 39–46).
- Kelly, A., et al. (2006). Toward reliable off-road autonomous vehicles operating in challenging environments. *The International Journal of Robotics Research*, 25(5–6), 449–483.
- Krotkov, E., & Hoffman, R. (1994). Terrain mapping for a walking planetary rover. *IEEE Trans. on Robotics & Automation*, 10(6).
- Kubota, T., Ejiri, R., Kunii, Y., & Nakatani, I. (2006). Autonomous behavior planning scheme for exploration rover. In *Second IEEE int. conf. on space mission challenges for information technology (SMC-IT 2006)* (p. 7).
- Kunii, Y., Tsuji, S., & Watari, M. (2003). Accuracy improvement of shadow range finder: SRF for 3D surface measurement. In *Proc. of IEEE/RSJ int. conf. on intelligent robots and systems (IROS 2003)* (Vol. 3, pp. 3041–3046).
- Lamarche, T. (2009). *Conception et prototypage d'un capteur LIDAR 3D pour la robotique mobile*. Master's thesis, Ecole Polytechnique de Montréal.
- Liegeois, A., & Moignard, C. (1993). *Lecture notes in computer science: Vol. 708. Geometric reasoning for perception and action* (pp. 51–65). Berlin: Springer. Chapter: Optimal motion planning of a mobile robot on a triangulated terrain model.
- Maimone, M., Biesiadecki, J., Tunstel, E., Cheng, Y., & Leger, C. (2006). *Int. for space robotics* (pp. 45–69). San Antonio: TSI Press. Chapter: Surface navigation and mobility intelligence on the Mars exploration rovers.
- Mandow, A., Cantador, T., Garcia-Cerezo, A., Reina, A., Martinez, J., & Morales, J. (2011). Fuzzy modeling of natural terrain elevation from a 3D scanner point cloud. In *IEEE 7th international symposium on intelligent signal processing* (pp. 1–5).
- Martin, E., L'Archevêque, R., Gemme, S., Rekleitis, I., & Dupuis, E. (2008). The avatar project: remote robotic operations conducted from the international space station. *IEEE Robotics & Automation Magazine*, 14(4), 20–27.
- Matthies, L., & Shafer, S. (1987). Error modeling in stereo navigation. *IEEE Journal of Robotics and Automation*, 3(3), 239–250.
- Montemerlo, M., Thrun, S., Dahlkamp, H., Stavens, D., & Strohband, S. (2006). Winning the DARPA grand challenge with an AI robot. In *Proc. of the AAAI nat. conf. on artificial intelligence*.
- Mora, A., Ishigami, G., Nagatani, K., & Yoshida, K. (2008). Sensing position planning for lunar exploration rovers. In *IEEE international conference on mechatronics and automation* (pp. 732–737).
- Mora, A., Nagatani, K., Yoshida, K., & Chacin, M. (2009). A path planning system based on 3D occlusion detection for lunar exploration rovers. In *Third IEEE international conference on space mission challenges for information technology*.
- Mourikis, A., Trawny, N., Roumeliotis, S., Johnson, A., & Matthies, L. (2007). Vision-aided inertial navigation for precise planetary landing: analysis and experiments. In *Proc. robotics: science and systems conf. (RSS'07)*, Atlanta, GA.
- Nesnas, I., Wright, A., Bajracharya, M., Simmons, R., & Estlin, T. (2003). Clarity and challenges of developing interoperable robotic software. In *Int. conf. on intelligent robots and systems (IROS)*, Nevada (pp. 2428–2435).
- Nsasi Bakambu, J., Gemme, S., & Dupuis, E. (2006). Rover localization through 3D terrain registration. In *IEEE/RSJ international conference on intelligent robots and systems*, Beijing, China.
- Preparata, F. P., & Shamos, M. I. (1985). *Computational geometry: an introduction*. New York: Springer.
- Rekleitis, I., Bedwani, J. L., Gemme, S., & Dupuis, E. (2007). Terrain modelling for planetary exploration. In *Computer and robot vision (CRV)*, Montreal, QC, Canada (pp. 243–249).
- Rekleitis, I., Bedwani, J. L., Dupuis, E., & Allard, P. (2008a). Path planning for planetary exploration. In *5th Canadian conf. on computer and robot vision (CRV)* (pp. 61–68).
- Rekleitis, I., Bedwani, J. L., Gingras, D., & Dupuis, E. (2008b). Experimental results for over-the-horizon planetary exploration using a Lidar sensor. In *11th int. symp. on experimental robotics (ISER'08)*.
- Rusinkiewicz, S., & Levoy, M. (2001). Efficient variants of the ICP algorithm. In *Proc. of the 3rd intl. conf. on 3D digital imaging and modeling* (pp. 145–152).
- Rusu, R. B., Sundareshan, A., Morisset, B., Hauser, K., Agrawal, M., Latombe, J. C., & Beetz, M. (2009). Leaving flatland: efficient real-time 3D perception and motion planning. *Journal of Field Robotics* (Special Issue on 3D Mapping). <http://files.rbrusu.com/publications/Rusu09JFR-LFlatland.pdf>.
- Silver, D., Bagnell, J. A. D., & Stentz, A. T. (2010). Learning from demonstration for autonomous navigation in complex unstructured terrain. *The International Journal of Robotics Research*, 29(12), 1565–1592.
- Silver, D., Ferguson, D., Morris, A. C., & Thayer, S. (2006). Topological exploration of subterranean environments. *Journal of Field Robotics*, 23(1), 395–415.
- Simmons, R., Krotkov, E., Chrisman, L., Cozman, F., Goodwin, R., Hebert, M., Katragadda, L., Koenig, S., Krishnaswamy, G., Shinoda, Y., Whittaker, W., & Klarer, P. (1995). Experience with rover navigation for lunar-like terrains. In *Proceedings of IEEE/RSJ international conference on intelligent robots and systems* (Vol. 1, pp. 441–446).
- Singh, S., Simmons, R., Smith, T., Stentz, A., Verma, V., Yahja, A., & Schwehr, K. (2000). Recent progress in local and global traversability for planetary rovers. In *Proceedings of the IEEE international conference on robotics and automation* (Vol. 2, pp. 1194–1200).
- Triebel, R., Pfaff, P., & Burgard, W. (2006). Multi-level surface maps for outdoor terrain mapping and loop closing. In *Proc. of the IEEE/RSJ int. conf. on intelligent robots and systems (IROS)*, Beijing, China (pp. 2276–2282).
- Vago, J. (2004). Overview of exomars mission preparation. In *8th ESA workshop on advanced space technologies for robotics & automation*, Noordwijk, The Netherlands.
- Vasudevan, S., Ramos, F., Nettleton, E., & Durrant-Whyte, H. (2010). A mine on its own: fully autonomous, remotely operated mine. *IEEE Robotics and Automation Magazine*, 63–73.
- Volpe, R. (2006). Rover functional autonomy development for the mars mobile science laboratory. In *IEEE aerospace conf.*
- Wettergreen, D., et al. (2005). Second experiments in the robotic investigation of life in the Atacama desert of Chile. In *8th int. symp. on artificial intelligence, robotics and automation in space*.
- Wettergreen, D., Jonak, D., Kohanbash, D., Moreland, S. J., Spiker, S., & Teza, J. (2009). Field experiments in mobility and navigation with a lunar rover prototype. In *Field and service robotics*.
- Wright, J., Trebi-Ollennu, A., Hartman, F., Cooper, B., Maxwell, S., Yen, J., & Morrison, J. (2005). Terrain modelling for in-situ activity planning and rehearsal for the mars exploration rovers. In *IEEE int. conf. on systems, man and cybernetics* (Vol. 2, pp. 1372–1377).



Ioannis Rekleitis is currently an Adjunct Professor at the School of Computer Science, McGill University. Between 2004 and 2007 he was a visiting fellow at the Canadian Space Agency. Between 2002 and 2003, he was a Postdoctoral Fellow at the Carnegie Mellon University in the Sensor Based Planning Lab with Professor Howie Choset. His research has focused on mobile robotics and in particular in the area of cooperating intelligent agents with application to multi-robot cooperative localization, map-

ping, exploration and coverage. He has work with underwater, terrestrial, aerial, and space robots. His interests extend to computer vision and sensor networks. Ioannis Rekleitis has published more than fifty journal and conference papers. He has an H-Index of 23 (using Google Scholar). He has served as Program Chair, Associate Editor, and Program Committee member in several journal and conferences. He is the Principal Investigator for a five year Natural Sciences and Engineering Research Council of Canada (NSERC) discovery grant and a Co-PI on Microsoft Research and NSERC grants. Ioannis Rekleitis was granted his Ph.D. from the School of Computer Science, McGill University, Montreal, Quebec, Canada in 2002 under the supervision of Professors Gregory Dudek and Evangelos Miliotis. Thesis title: "Co-operative Localization and Multi-Robot Exploration". He finished his M.Sc. in McGill University in the field of Computer Vision in 1995. He was granted his B.Sc. in 1991 from the Department of Informatics, University of Athens, Greece.



Jean-Luc Bedwani has joined the Institut de recherche of Hydro-Québec (IREQ) in 2009 as a researcher in robotic vision. He is involved in the development and integration of vision systems required to improve the autonomy and capabilities of robots. His current research interests are in the use of laser line cameras by flexible robotic arm for measuring and reconstructing surfaces in 3D. Prior to his position at Hydro-Québec, he fulfilled the role of scientific authority in the Avatar Explore experiment

conducted by the Canadian Space Agency. He also worked as a software engineer to develop a novel digital X-ray system for ImaSight Inc. He also worked for Impath Networks where he developed a Video-Over-IP hardware device. Jean-Luc Bedwani holds a B.Sc.A. in computer engineering from Université de Sherbrooke and an M.Sc.A. in electrical engineering from Université de Sherbrooke.



Erick Dupuis has joined the Canadian Space Agency in 1992 and is currently the manager, of Integration and Planning, at Space Science & Technology. In his previous role he was the head of the robotics group in the Space Exploration Branch. He has been a research engineer in robotics, the lead systems engineer for the SPDM Task Verification Facility and for several projects in planetary exploration. He was also a member of the Jet Propulsion Lab's Mars Program Systems Engineering Team. He is currently a member of the NASA/ESA Joint Mars Exploration Program Analysis and Review Team. His research interests have concentrated on the remote operation of space robots, starting with ground control of Canada's Mobile Servicing System on the International Space Station. He is now directing R&D on autonomous rover navigation for planetary exploration missions. Erick Dupuis holds a B.Sc.A. from the University of Ottawa, a Master's degree from the Massachusetts Institute of Technology and a Ph.D. from McGill University. All of his degrees are in mechanical engineering with specialization in robotics.



Tom Lamarche has joined the Canadian Space Agency in 2001 and is currently a robotics engineer in the Space Exploration branch. As a research engineer, he has been involved in the development, integration, test and demonstration of numerous robotic systems and subsystems, as well as multiple related test beds. Covering both robotic manipulators and rovers, his research interests have concentrated on sensing, actuation, motion control, autonomous navigation and electronics. In the recent years, he has also been fulfilling the role of technical authority on various technology development contracts put in place by CSA, including one to develop the JUNO rover platform. Tom Lamarche holds a technical degree in industrial electronics, a B.Eng. in electrical engineering from École de Technologie Supérieure and an M.Sc.A. in electrical engineering from École Polytechnique de Montréal.



Pierre Allard has joined the Canadian Space Agency in 1998 and works as a robotics specialist in the field of mobile robotics for planetary exploration. He was part of the Mars Phoenix Lander Mission where he participated in the development of the ground data system for the Meteorological Station and worked in mission operations as a telemetry analyst. He was also a robotic instructor for Canada's Mobile Servicing System on the International Space Station. His research interests have concentrated on the

development and integration of autonomy software for mobile robots operating in natural environments. Pierre Allard holds a Bachelor in Space Sciences from Collège militaire royal de Saint-Jean, and a Master's degree in Electrical Engineering from McGill University.