

Autonomous Planetary Exploration using LIDAR data

Ioannis Rekleitis, Jean-Luc Bedwani, and Erick Dupuis

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 autonomously navigate 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 flatter terrain and safely away from obstacles. Experiments performed in CSA's Mars emulation terrain that validate our approach are also presented.

I. INTRODUCTION

Mobile robotics has enabled scientific breakthroughs in planetary exploration [1]. 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 would take them along a safe trajectory. MER’s have reached traverses of 300m/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) [2] and ESA’s ExoMars [3]. 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 via-points 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.

In the next Section we discuss the state-of-the-art in robotic planetary exploration. Section II presents the overall process for planetary exploration together with a short description of our test-bed. Next we present a summary of our approach to environmental modelling, Section IV

I. Rekleitis, is an Adjunct Professor with the School of Computer Science, McGill University yiannis@cim.mcgill.ca
J.-L. Bedwani, and E. Dupuis are with the Canadian Space Agency.
FirstName.LastName@space.gc.ca
© Canadian Space Agency 2009.



Fig. 1. The Mars terrain with our modified P2AT

provides a summary of our approach on terrain modelling using LIDAR data. Section V presents a new algorithm for planning an optimal path for the rover using the Irregular Triangular Mesh (ITM) while keeping it a safe distance from the detected obstacles. Next, the results from a variety of experiments are presented in Section VI.

II. RELATED WORK

The work on planetary exploration can be divided according to the sensing modality used and also according to the environment representation used. Both vision [4] and LIDAR [5] technologies have been proposed, each one having different advantages and disadvantages. Early work on planetary exploration using LIDAR [5], though promising, was not compatible with the flight weight constraints. The Mars Exploration Rovers are currently performing long traverses using vision [6]. Vision although lightweight, requires more computing power, has limited range and accuracy. Currently, LIDAR based systems¹ 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.

The problem of autonomous long range navigation is also very important in terrestrial settings. The DARPA grand challenge in 2005 resulted in several vehicles travelling 132 miles over desert terrain [7]. The majority of the contestants used a combination of multiple LIDAR, vision, and RADAR sensors. Similar work involved traverses on the order to 30Km in the Atacama desert [8] using vision. See also [9] for a discussion of the many challenges and additional related work.

Currently, the most advanced exploration robots that have been deployed for planetary exploration are the Mars Exploration Rovers (MER) “Spirit” and “Opportunity”. These rovers have successfully demonstrated, on Mars, concepts such as visual odometry and autonomous path selection from

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

a terrain model acquired from sensor data [10]. The main sensor suite used for terrain assessment for the MER has been passive stereo vision [11]. The models obtained through stereo imagery are used for both automatic terrain assessment and visual odometry. 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 planned for 2010 [12].

In the case of automatic terrain assessment, the 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 [13]. LIDAR sensors have also been used successfully for 3D mapping of underground mines [14].

For our work, we have been using a laser range sensor (LIDAR) as the main sensing modality. Several factors have motivated the choice of a LIDAR sensor: among others, our mobility platform has very low ground clearance. A LIDAR sensor is capable of providing range data to build terrain models with 1-2cm accuracy. Such an 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. 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.

III. OVERVIEW

The goal of our work is to navigate autonomously from the current position to an operator-specified location which lies 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 descent [15]. 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 relocating between successive scans. Preliminary localization test results, though promising, have not yet proven to be robust enough. As such, scan to scan localization is in the immediate future plans, but outside the scope of this paper.

Our approach is implemented on a modified Pioneer P2-AT robot; cf. Fig. 1. The robot is equipped with wheel

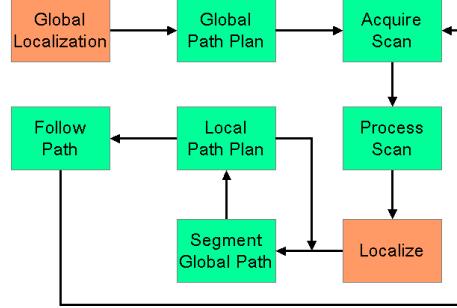


Fig. 2. The main components of Autonomous Over-the-Horizon Navigation.

encoders for odometry, a six-axis Inertial Measurement Unit (IMU), and a digital compass for azimuth. Two different LIDAR sensors have been used in the experiments. During 2006 an ILRIS 3D unit from OPTECH was used with a range of over 1 km but with a field of view (*fov*) limited to 40°. During 2007 and 2008 a scanning LIDAR sensor is mounted on top of the robot to obtain detailed terrain models for localization and path planning. It is using a SICK laser mounted vertically on a pan-unit, which provides a 360° field of view (*fov*). As the pan-unit rotates, the SICK scans vertical slices, thus producing complete coverage all around the rover. The LIDAR sensor is mounted at a low grazing angle, as evidenced by the long shadows (lack of data) in the scan; cf. Fig. 3a. The height at which the sensor is mounted, on several occasions, did not clear above obstacles or steeper slopes, thus requiring frequent scans. On the other hand this design choice introduced realistic challenges.

IV. TERRAIN MODELLING

As mentioned above, the main sensor of the robotic platform is a 2D LIDAR sensor mounted on a pan-unit thus producing full 2.5D data with a 360° *fov*. The sensor returns a set of points in polar coordinates $[\rho, \phi, \theta]$ which represent the distance to the obstacle (ρ), the elevation angle ($\phi \in [-90^\circ, 90^\circ]$), and the azimuth ($\theta \in [-180^\circ, 180^\circ]$). The angular resolution of our sensor can be set to 0.5° or 1°. The main advantage of the LIDAR is that it directly provides a 2.5D point cloud giving the coordinates of the terrain in its *fov*, with no need for post-processing. As a point cloud can not be used for robotic operations such as path-planning and navigation, a different representation was chosen. The main requirements to be fulfilled are: the new representation had to be compact, a single scan consists of tens of thousand of points; it had to be compatible with navigation algorithms; and, it must also preserve the scientific data contained in the terrain topography. To fulfil these requirements, the ITM terrain representation was chosen [16]. Our terrain modelling approach has been presented before [17] together with an analysis of the different terrain properties encountered in the Mars emulation terrain used for our experiments. It is worth noting that each individual scan results in a 2.5D surface, however, the ITM formalism is capable of combining several scans to model 3D structures such as overhangs. Next, we

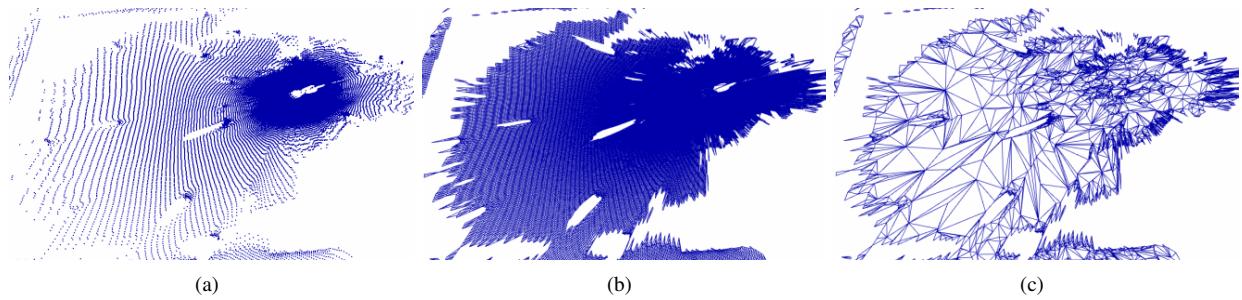


Fig. 3. (a) The raw point cloud. (b) Delaunay triangulation in polar coordinates. (c) The decimated irregular triangular mesh

are going to review the key aspects of the terrain modelling process to provide sufficient background for the discussion of the autonomous navigation process.

The 2.5D points in polar coordinates are used as the input to the Delaunay triangulation algorithm, which creates triangles based on the adjacency of the data points. It is worth noting that performing the triangulation in polar coordinates allows us to eliminate shadow triangles and outliers in a systematic manner. Figure 3a presents the 2.5D point cloud, it is clear that even small obstacles create long shadows with no measurements in them. The Delaunay triangulation in polar coordinates is used which preserves the shadows; cf. Fig. 3b. 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 original points are at a distance less than two centimeters from the resulting mesh; cf. Fig. 3c. The implementation of the terrain modelling and decimation using a triangular mesh is done using the Visualisation Toolkit [18] libraries. Two different LIDAR sensors have been used by CSA, with different range, *fov*, 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 modelling maintained the high levels of accuracy while at the same time reducing the data volume by 90%-95% [19].

The ITM preserves the science content from the topographical data, while capable of modelling concave geological structures like overhangs and caverns. Please refer to [17] for a discussion of various LIDAR modelling methods. It is worth mentioning though the recent approach which combines digital elevation maps with multiple layers [20] as it models concave structures. However, the method in [20] does not lead easily to path-planning.

V. PATH-PLANNING ON ITM

One of the advantages of the ITM representation is that it is amenable to path planning. Indeed, 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 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 using Dijkstra's graph search methods from the *jgraph* java library with a variety of cost functions taking into account distance travelled, terrain slope, and terrain roughness. 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 an unacceptable trajectory that zigzags unnecessarily between cell centers.

It is therefore necessary to smooth out the resulting trajectory by removing superfluous via-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 via-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 via-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 via-point that could not be removed and stops when reaching the final destination.

In this context, the usage of ITM introduces additional challenges. First, on flat terrain, the cells are relatively large. Therefore, although large cells are preferable for safety reasons, a cost function taking only distance travelled 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

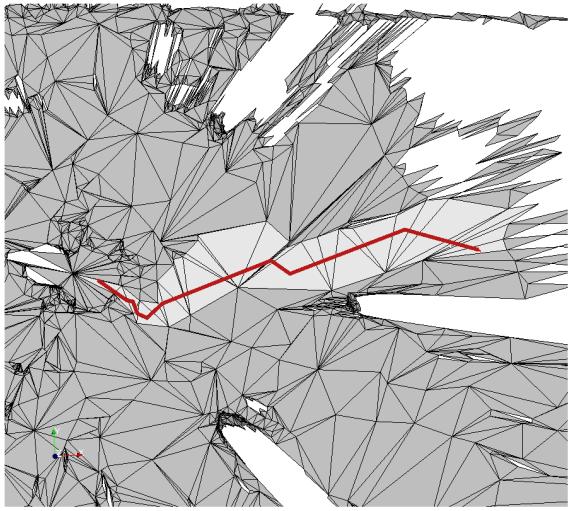


Fig. 4. Results of Path Planner on Typical Irregular Triangular Mesh the resulting safety corridor can be very narrow, hence more difficult to navigate.

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 the environment.

Figure 4 shows a path that was planned in a typical terrain scan obtained using a LIDAR range scanner in CSA's Mars emulation terrain. 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 travelling from cell i to cell j :

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

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. Parameters α and β 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:

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

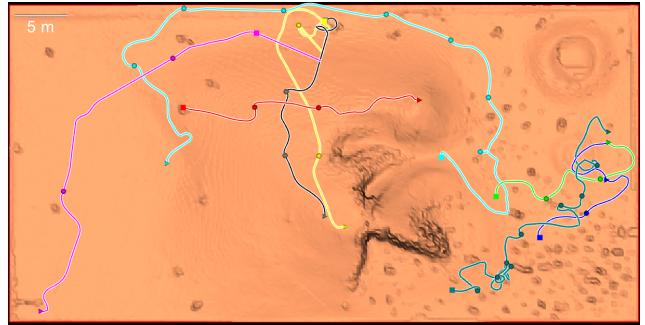


Fig. 5. A model of the Mars emulation terrain together with the recorded trajectories for all the different experiments performed in the 2007 testing season.

and

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

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

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

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 equation 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.

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. Preliminary results indicate that, using typical terrain models, the computation time can be accelerated by a factor ranging between 3 and 6. The path-planner using A* was also tested off-line using the collected scans of the 2007 testing season. 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 [19]. 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 [21].

VI. EXPERIMENTAL RESULTS

The experiments were run at the Canadian Space Agency's Mars emulation terrain: a 60 meter by 30 meter rover test area that emulates the topography of a broad variety of Martian landscapes. The terrain includes plains, a hill, a canyon and rock fields of varying density; cf. Fig. 1 for a photograph of the terrain and Fig. 5 for a model of the complete terrain. It is worth noting, that the model is four years old and as the terrain is outdoors, wind, rain, and snow have somewhat modify the topography, as such the model is used mainly for illustrative purposes. A large number of experiments were conducted in the 2006 testing session ² using the ILRIS 3D LIDAR sensor with limited *fov* sensor. In the 2007 testing session, 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 filter. A large number of closed trajectories was executed varying in size and in location. Consequently, we were able to quantify the error exhibited in terms of both the length of the travelled 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.

Next a series of over-the-horizon trajectories were executed to test the integrated system. Figure 5 presents the trajectories of several experiments of autonomous over-the-horizon navigation over a model of the Mars emulation terrain. As can be seen, the experiments covered all the terrain types. 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. 5.

Figure 6 presents the results from a representative fully autonomous navigation experiment. First the model of the Mars emulation terrain was used as the global map and the operator entered the global destination, then a simple planner was used to calculate the global path. Figure 6a shows all the paths; the global path is presented as a dashed (blue) line, and the planned local paths together with the odometric estimates are drawn as green and red lines respectively. The start, the two way points, and the global destination are also marked. Figure 6b 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 local destination.

²CSA's location allows outside testing only during the summer and early fall months, thus, the 2006 and the 2007 testing seasons were from June to early November.

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. Way-Point-1 in Fig. 6a is the first selected local destination. The rover planned and executed a successful traversal and reached the first way point, at which step it took the second scan; cf. Fig. 6c. 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 final trajectory; cf. Fig. 6d.

VII. 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 include further research on localisation and scan matching. This will enable the rover to re-localise 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.

REFERENCES

- [1] M. Maimone, J. Biesiadecki, E. Tunstel, Y. Cheng, and C. Leger, *Int. for Space Robotics*. TSI press, 2006, ch. Surface Navigation and Mobility Intelligence on the Mars Exploration Rovers, pp. 45–69.
- [2] R. Volpe, “Rover functional autonomy development for the mars mobile science laboratory,” in *IEEE Aerospace Conf.*, 2006.
- [3] J. Vago, “Overview of exomars mission preparation,” in *8th ESA Workshop on Advanced Space Technologies for Robotics & Automation*, Noordwijk, The Netherlands, Nov. 2004.
- [4] L. Matthies and S. Shafer, “Error modeling in stereo navigation,” *IEEE J. of Robotics and Automation*, vol. 3, no. 3, pp. 239 – 250, 1987.
- [5] M. Hebert et al., “Terrain mapping for a roving planetary explorer,” in *Proc. of the IEEE Int. Conf. on Robotics and Automation (ICRA '89)*, vol. 2, May 1989, pp. 997–1002.
- [6] S. Goldberg, M. Maimone, and L. Matthies, “Stereo vision and rover navigation software for planetary exploration,” in *Proc. of IEEE Aerospace Conf.*, vol. 5, Mar. 2002, pp. 2025–2036.
- [7] M. Montemerlo, S. Thrun, H. Dahlkamp, D. Stavens, and S. Strohband, “Winning the darpa grand challenge with an ai robot,” in *Proc. of the AAAI Nat. Conf. on Artificial Intelligence*, 2006.
- [8] D. Wettergreen and et al., “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*, Sep. 2005.

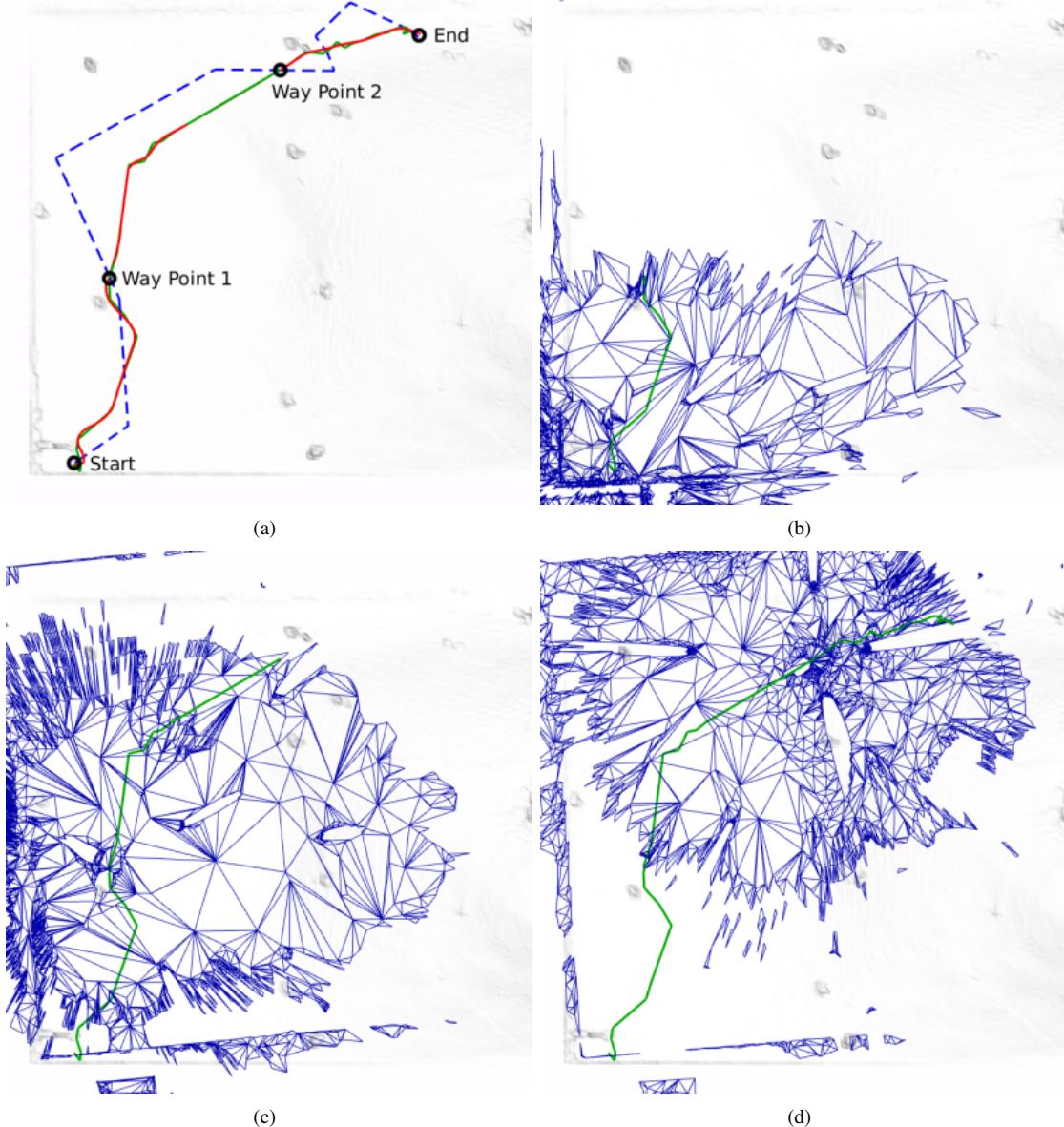


Fig. 6. (a) The global path, the start and final destination, the two way-points and the local-paths. (b) The first scan together with the first local path. (c) The second scan together with the first and second local paths. (d) the final scan together with all the local-paths

- [9] A. Kelly and et al., "Toward reliable off-road autonomous vehicles operating in challenging environments," *The Int. Journal of Robotics Research*, vol. 25, no. 5-6, pp. 449–483, Jun. 2006.
- [10] J. Biesiadecki, C. Leger, and M. Maimone, "Tradeoffs between directed and autonomous on the mars exploration rovers," in *Proc. of Int. Symposium of Robotics Research*, San Francisco, 2005.
- [11] J. Wright, A. Trebi-Ollennu, F. Hartman, B. Cooper, S. Maxwell, J. Yen, and J. Morrison, "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, 2005, pp. 1372 – 1377.
- [12] A. Johnson, S. Goldberg, C. Yang, and L. Matthies, "Robust and efficient stereo feature tracking for visual odometry," in *IEEE Int. Conf. on Robotics and Automation*, May 2008, pp. 39 – 46.
- [13] A. M. Howard and E. W. Tunstel, Eds. *Intelligence for Space Robotics*. TSI Press, 2006, ch. MER Surface Navigation and Mobility.
- [14] D. Silver, D. Ferguson, A. C. Morris, and S. Thayer, "Topological exploration of subterranean environments," *Journal of Field Robotics*, vol. 23, no. 1, pp. 395–415, Jul. 2006.
- [15] A. Mourikis, N. Trawny, S. Roumeliotis, A. Johnson, and L. Matthies, "Vision-aided inertial navigation for precise planetary landing: Analysis and experiments," in *Proc. Robotics: Science and Systems Conf. (RSS'07)*, Atlanta, GA, June 27-30 2007.
- [16] R. J. Fowler and J. J. Little, "Automatic extraction of irregular network digital terrain models," in *SIGGRAPH '79: Proc. of the 6th annual conference on Computer graphics & interactive techniques*, 1979, pp. 199–207.
- [17] I. Rekleitis, J.-L. Bedwani, S. Gemme, and E. Dupuis, "Terrain modelling for planetary exploration," in *Computer and Robot Vision (CRV)*, Montreal, QC, Canada, May 2007, pp. 243–249.
- [18] "Kitware inc. visualization toolkits." <http://www.vtk.org>, Website (accessed: September 2005), 2005.
- [19] I. Rekleitis, J.-L. Bedwani, D. Gingras, and E. Dupuis, "Experimental results for over-the-horizon planetary exploration using a lidar sensor," in *11th Int. Symp. on Experimental Robotics (ISER '08)*, Jul. 2008.
- [20] R. Triebel, P. Pfaff, and W. Burgard, "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, 2006, pp. 2276–2282.
- [21] I. Rekleitis, J.-L. Bedwani, E. Dupuis, and P. Allard, "Path planning for planetary exploration," in *5th Canadian Conf. on Computer and Robot Vision (CRV)*, May 2008, pp. 61–68.