DEVELOPING A FRAMEWORK FOR RELIABLE AUTONOMOUS SURFACE MOBILITY

David Wettergreen[(1)] and Michael Wagner[(2)]
[(1)]The Robotics Institute, Carnegie Mellon University, Pittsburgh, PA 15213 USA, dsw@ri.cmu.edu
[(2)] ProtoInnovations, LLC, 100 43rd Street, Pittsburgh, PA 15201 USA

ABSTRACT

The Reliable Autonomous Surface Mobility (RASM) architecture and algorithms model terrains, plan paths, and control motion to enable rovers to navigate autonomously through kilometers of barren terrain. We have developed requirements for rovers from the needs of field investigations and mission scenarios of the last decade. Thus far the RASM architecture has been ported to five rovers and tested in field sites in Hawaii, California and Canada. We report on the architectural framework, the underlying algorithms, and representative results.

1. INTRODUCTION

As the capability and durability of planetary rovers increases, the ambitions of their missions of exploration also grow. To accomplish more challenging goals these rovers must navigate autonomously for kilometers at a time. Our research focuses on a comprehensive method to ensure safe guidance for rovers (minimize risk) with reasonable performance (minimize time, energy, and uncertainty); the aim is reliable autonomous surface mobility.

The Reliable Autonomous Surface Mobility (RASM) architecture and algorithms map terrain and plan paths that enable rovers to navigate in unknown, planetary terrain. RASM is intended for a broad range of operating scenarios including scouting, survey, and in support of activities like excavation and construction. We have applied RASM to five different rover prototypes (Figure 1, Juno rover not pictured) and continue to collect and analyze data on its performance in field tests.

The idea underlying RASM is that an exploration rover will move steadily and continuously through terrain while making frequent measurements. RASM continuously aggregates terrain surface points, from rover sensing or prior registered maps, and manages this integrated model in a database. It generates, and regenerates, a map for determining the best path to the next goal. This map is tailored to the demands of navigation; its sparse, non-uniform mesh structure and spatially-associated terrain properties are the essential information for exploration planning. RASM evaluates potential paths using metrics for time, energy, and information gain. We say that RASM is optimistic because it will try routes across areas of sparse data assuming that it will be free of obstacles. We find in practice that if an obstacles is present that it will become observable as the rover approaches it and then it can be deliberately avoided. This is in contrast to many

Figure 1. Rovers operated by RASM software include Zoë, Artemis Jr., K-REX, LATUV (from upper left).

navigation systems that pessimistically assume that a lack of data, or unknown terrain, will be impassible. We find that in barren terrain, it works to be optimistic.

In this paper we describe work to formalize the RASM architecture and verify the reliability of the method. The RASM software has been made portable and has been specialized to a variety of terrain sensors and robotic platforms. We describe the key algorithms and how they have been adapted to different applications and we report on performance in field experiments.

2. REQUIREMENTS

The purpose of RASM is to ensure safe and efficient mobility with a reasonable balance of speed, energy and risk; thus reliable autonomous surface mobility. The objectives of RASM are to:

- Model terrain geometrically
- Support multipurpose traversability evaluation
- Provide wide sensor and rover compatibility
- Accommodate weak localization
- Achieve real-time performance
- Command continuous motion

We have derived requirements from the likely scenarios that a planetary rover might employ and the specific tasks that must be performed. Scenarios include: landing, scouting, survey, excavation, construction, infrastructure maintenance, and deployment. Specific tasks in each scenario relate to rover navigation—in some cases indirectly—and influence the RASM requirements. Requirements are also derived from desired performance and other qualities of how the tasks are to be executed.

Landing scenarios drive initialization and calibration issues, as well as the need to make special-purpose maneuvers. The scouting scenario implies navigation in unknown terrain and the various activities for navigating autonomously in such an environment. The survey scenario is for large-area coverage, often with prior terrain knowledge or infrastructure, but needing less communication. Excavation scenarios impose situations with substantial disturbance to motion execution. As well, sensor noise becomes significant. Construction scenarios introduce varying payloads, built environment, and increased reliance on complex infrastructure. Similarly, maintenance scenarios involve operating in close proximity or contact to the built environment. Localization may become crucial and the opportunity of artificial features is valuable. Within each scenario specific activities and actions have been identified and correlated to generate requirements. Requirements for a hypothetical rover performing scenarios including landing, scouting, survey, excavation, construction, maintenance as well as terrestrial aspects of deployment and testing are summarized in Table 1.

Table 1 RASM Derived Requirements

Communication	Localization	Mapping	Navigation	Operation	Mobility	Environmental
1.1. Ensure eventual delivery (reliability)	2.1. No artificial infra-structure	3.1. Produce timely maps	4.1. Command continuous driving	5.1. Support autonomous traverse to goal	6.1. Ascend 15° slope	7.1. Operate for 6 hours
1.2. Robust to variable bandwidth (reliability)	2.2. Contin-uous rover position and velocity	3.2. Incorporate geometric and non-geometric information	4.2. Go fast (greater than 5 KPH)	5.2. Support supervisory teleoperation (waypoint guidance)	6.2. Descend 20° slope	7.2. Operate in low light and darkness
1.3. Robust to variable latency (persistency)	2.3. Estimate relative pose, 1% error	3.3. Enable 3D features	4.3. Perform multi-resolution evaluation	5.3. Support tele-operation	6.3. Surmount 25-cm obstacle	7.3. Resist dust (increased sensor noise)
1.4. Bridge arbitrary link loss/store and forward (durability)	2.4. Estimate absolute pose, 5% error	3.4. Support one or more navigators	4.4. Avoid-dynamic obstacles (less than rover speed)	5.4. Detect faults	6.4. Straddle 25-cm obstacle	
1.5. Synchron-ized data messaging (modular)	2.5. Operate high-rate, 30 Hz	3.5. Export maps	4.5. Straddle obstacles	5.5. Diagnose and recover non-fatal faults		
1.6. Priority-based and time-critical delivery			4.6. Consider non-geometric attributes			

The requirements, which reach across these multiple scenarios, have been organized into areas of communication, localization, mapping, navigation, operation, mobility and environmental factors. Metric values of the requirements, for example cycle at 5 Hz, drive navigation at 5 KPH, and ascend 15° slopes are attuned to recent lunar mission concepts and to current research applications. We have used these requirements to guide the design of RASM and perform tests to validate our navigation framework.

3. ARCHITECTURE

A planetary rover's architecture and algorithms for modeling terrain, planning paths, and controlling motion, enable it to navigate autonomously. Our rover navigation software has evolved from an early emphasis on obstacle avoidance and resource monitoring to accommodate the needs of long-distance autonomous traverse.[1] Improvements for more deliberate exploration organized the architecture into three functional groups: mission planning, navigation, and science.[2][3] In the present work we have formalized an architecture in which an integrated model of the environment serves all aspects of exploration.

A planetary rover engaged in exploration is most effective if it moves swiftly to allow for making frequent observations of its surroundings. The rover captures new observations after appreciable driving, or time, has elapsed. The architecture must support this continuous influx of new data and organize it in a coherent manner. The collection of observations provides measurements to model the environment and the estimate position, determine efficient and safe paths, and choose exploration actions.

The RASM architecture encompasses three core modules: Mapper, Localizer and Planner. (Figure 2) Additional functionality is provided by an Executive that modifies behavior and goals and a Visualizer that displays terrain and positions. Central to the RASM architecture is a topological model of terrain observations, cached maps, and ancillary information.

Integrated Model

RASM organizes information efficiently in a database to support of a variety of functions needed for mapping, localization and planning. It begins with individual sensor observations of terrain geometry, as unordered sets of points in three-dimensional space, called *point clouds*. These point clouds are associated with the growing topological model of the explored environment and added to the database along with their relationship, a homogeneous transform, to neighboring observations.

Point clouds can source from calibrated stereo cameras that transform disparity into points in the coordinate frame of the sensor. They can also come from direct range-sensors such as LIDAR or RADAR. Relevant Digital Elevation Models (DEMs), from orbital sensors or prior ground surveys, can also be simplified to a collection of geometric surface points. These too can be added to the database as a point cloud, albeit of a larger scale than rover-based measurements. RASM is able to navigate without any prior information but it also seamlessly incorporates a DEM if it is available. By design the RASM architecture is agnostic as to the source of the three-dimensional data that can be incorporated into its model of the environment.

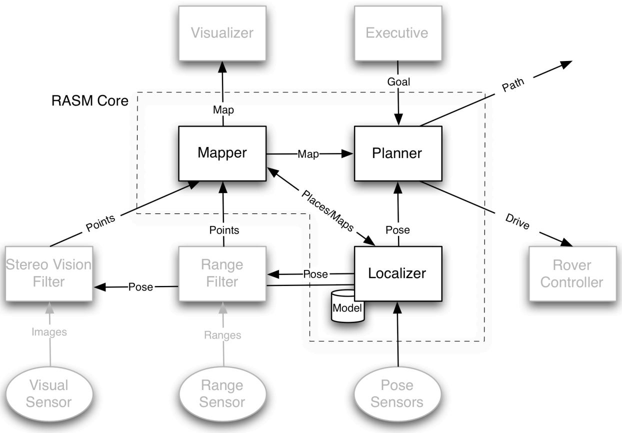

Figure 2. RASM Architecture. The Mapper, Localizer and Planner contain the key algorithms for autonomous navigation. A topologic model of terrain and ancillary information is a shared repository for maps and places.

Points from a sensor observation may also have associated ancillary information. In a relational database this information can take any form, to suggest a few: the uncertainty in the measurement, the variance on the value, material properties of the terrain, the velocity of the point, or a measurement from a scientific instrument. As example, points can be linked to corresponding visual features, which can be useful for localization when it is necessary to recognize a place. Points with ancillary data are preserved through any data reduction operations that may be performed on the database. All functions that reduce, decimate or transform points are implemented to check for ancillary data, maintain those points and then either bypass them or reinsert them after data reduction is complete. (Figure 3)

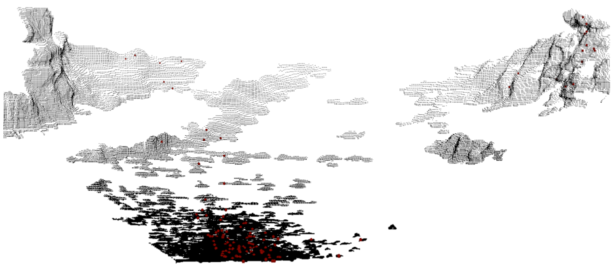

Figure 3. Points in a cloud include tagged (red points) with features extracted by visual feature detector.

Source imagery remains associated with points in the database so that properties can be extracted and matched later. This has shown to aid in the alignment of otherwise geometrically indistinct point clouds. Each piece of data is also tagged with the time and pose at which its was captured by sensors.

The database continually accepts new point clouds but it also grows as secondary information is generated from these primary sensor observations. Eventually for multi-day missions it will be necessary to eliminate data from the database to remain within some storage limit.

3.1. Mapping

The RASM mapping algorithms operate on point clouds to first reduce the number of points, then link nearby points to create a mesh of vertices and edges. These meshes can then be further reduced so that they are sparse where is there is little complexity in the terrain and dense where details are significant. A variety of mesh reduction algorithms applying edge and triangle contraction using volume, slope and other metrics have been evaluated and can be applied. Typically the Mapper applies edge contraction, replacing edges with a point, that minimizes volumetric change in the mesh.

Decimation

Some sensors, like high-resolution stereo cameras, produce dense point clouds. Meshing all of these points into a vertex graph is computationally expensive and mesh reduction is tedious when tens or hundreds of thousands of points are involved. RASM decimates—drastically subsamples—points before meshing occurs.

The Mapper employs a covariance decimation algorithm that splits the point cloud along planes of greatest variance.[4] Each new point cloud is characterized using principal component analysis and each point is projected onto the principal components subspace creating two smaller point clouds each with lower variance. The decimation splitting step then recurs on each of the resulting point clouds until the total variance or the number of points in the cloud is below a specified threshold. Then each of the final groups of points are replaced by a representative point at its centroid. In experiments covariance decimation produced smaller distances (67% reduction in average distance and 92% reduction in maximum distance) between a full resolution cloud and the decimated cloud when compared to naïve methods that uniformly subsample the original point cloud. (Figure 4)

Meshing

The decimated point cloud is meshed into an graph with each point as a vertex and edges determined so as to form non-uniform triangles of maximum area. [5] It is important to note that occluded regions, for which there are no surface points, are triangulated as well. No vertices are present in occluded regions, as can be seen in the low density of edges in occlusions. If the area is observed eventually, then vertices will be introduced. As part of the merging process edges that capture the newly observed structure are established with no ambiguity.

To ensure that the new mesh is below the target number of points a final step uses quadric error metrics to characterize the surface shape local to a point in the cloud, and then performs iterative vertex pair contraction to reduce the size of the mesh.[6] Based on the error in surface approximation (change in surface shape), the algorithm calculates the error in removing a pair of adjacent points and replacing them with a representative central point.

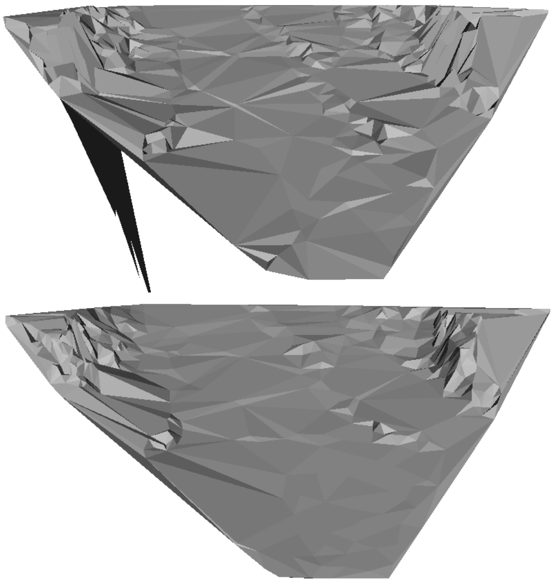

Figure 4. A point-cloud decimated by uniform sampling (top) and by covariance decimation (bottom) in which edge outliers are eliminated and resolution is focused on areas of complex geometry (data from Figure 3.)

The pair of points that results in the lowest increase in error to the overall mesh is removed, and the algorithm iterates until the point cloud has reached a target size.

Triangulated meshes have been applied to modeling of natural terrain to reduce the available geometric information to its essential form.[7] An advantage of triangulated meshes, as illustrated by the dense data modeled by Bamako, et al. is that resolution of the mesh can be adjusted with the amount of information, specifically the complexity of terrain.[8] A variety of metrics can be applied to reduce the size of the mesh. By aggregating neighboring points of low variance the result is a uniform distribution in level terrain, as seen in the center of terrain in Figure 4.

Representing terrain geometry as a set of triangulated meshes has advantages to memory efficiency; meshes are reduced to the minimum number of mesh-faces needed to represent a given terrain. Faces can be iteratively removed until a desired level of fidelity for navigation is reached. Due to the many applications of meshes these algorithms have been well optimized.

The Mapper must continually generate new meshes as it encounters new terrain. This requires a cycle of mesh generation and then merging with prior meshes. RASM uses the rover pose estimate for initial alignment and then the iterative closest point (ICP) algorithm [9] for fine correction of position and orientation of the mesh. ICP is only used to make relative corrections in mesh merging. At times ICP does not converge to refine position but if no solution is found, then the initial estimate of pose can be used. Similarly if the error the ICP surface fit is large, the erroneous refinement can be discarded without significant impact to navigation.

The navigation process proceeds with the newly reduced mesh being merged by the Mapper with previously generated, and merged meshes, to produce a map of the local terrain. (Figure 5, next page) Meshes can be merged continuously as needed. Places that are revisited can be relocated from the topological model and will have prior meshes that can be retrieved from the database. The mapping process can be initialized by pulling all the meshes near any location and merging them into a best map of the location. As they aggregate a detailed map of the local terrain emerges.

This process of point generation, decimation, meshing, reduction and merging has been shown to execute at approximately 100 ms on a 2 GHz processor core. When computation is required to correlate stereo image pairs, generate disparity and then transform ranges into a point cloud and this requires an 150 ms (for 250 ms total), while with wide-area laser scanning, point cloud processing is as little as 20ms (120 ms total). For vehicle speeds up to 2 m/s the RASM cycle time of 4 Hz is sufficient to support continuous motion.

3.2. Localization

The localization algorithms employed by RASM incorporate both directly sensed properties, like wheel odometry and inertially sensed velocities and accelerations, and information derived from integrated topological model. The Localizer employs an extended Kalman filter to continuously estimation position and orientation. The Mapper requires the location of each terrain observation and the Planner needs to know the rover location so that it can evaluate paths.

As each new sensor observation is made, the point cloud is placed in a topological model by the Localizer. (Figure 6) Design of the Localizer incorporates components to organize places and to perform pose refinements. The pose refinements alter the relationships between sensor observations and are maintained in a topologic model that structures the RASM database.

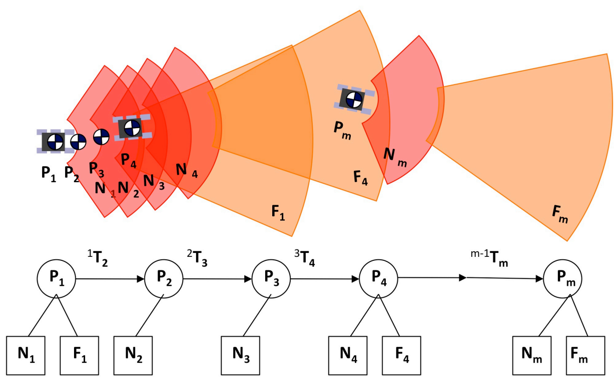

Figure 6. Observations occur periodically with various sensors and the topologic structure of these places is maintained by RASM.

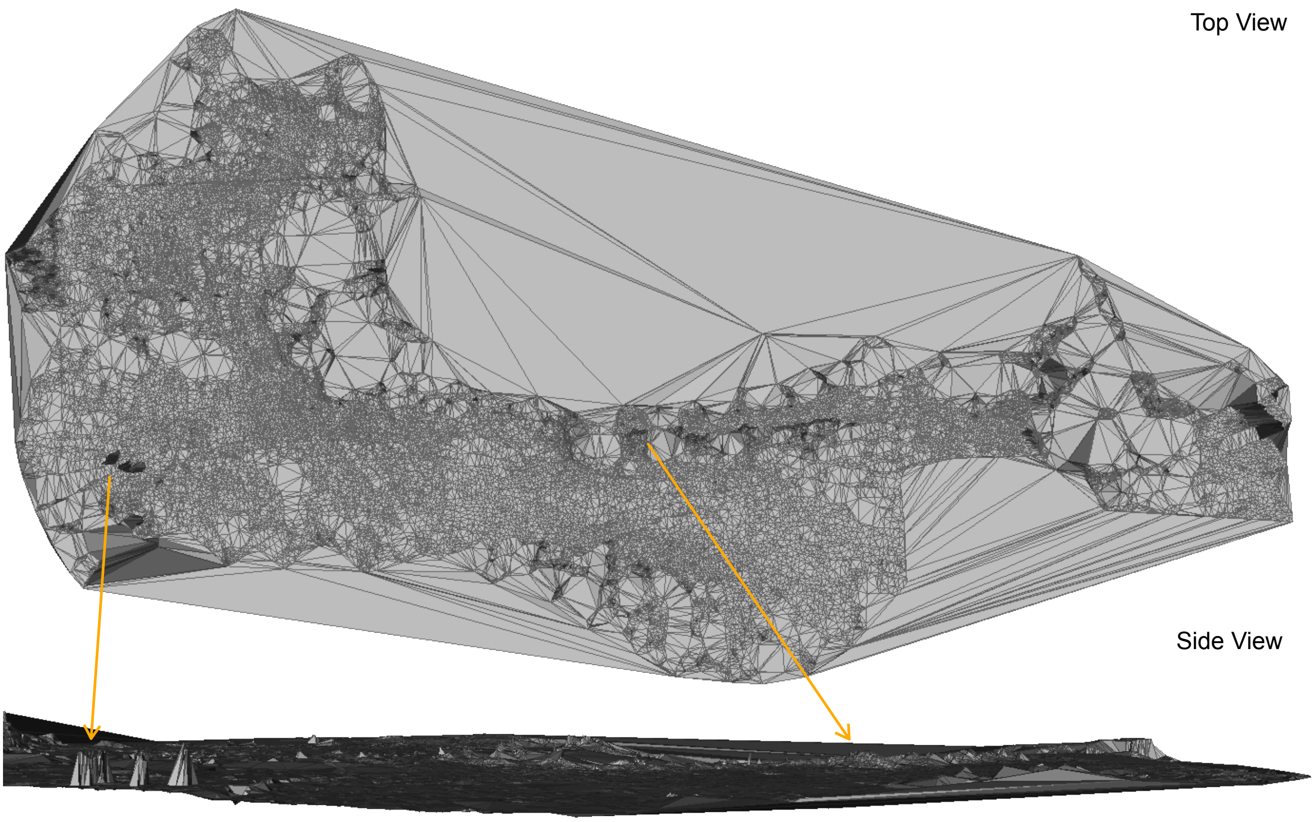

Figure 5. Terrain mesh (top) assembled from 1353 sensor observations using stereo cameras. Model includes people (at left) and row of dirt mounds. Mesh reduced to 34,306 vertices for navigation. Maximum dimension is 108 meters.

The Planner and Localizer work closely, sharing information through the database, to create meshes of sensor observations and associate them with visual features that define a recognizable place. The Localizer creates a graph of these places and continually refines the spatial relationships among places. Places are thus organized in a topological model. In parallel the Mapper can obtain the most accurate information and build maps while the Localizer organizes places and continually refines the topologic structure. Within the Localizer there are two functional aspects, organizing places and refining poses. (Figure 7)

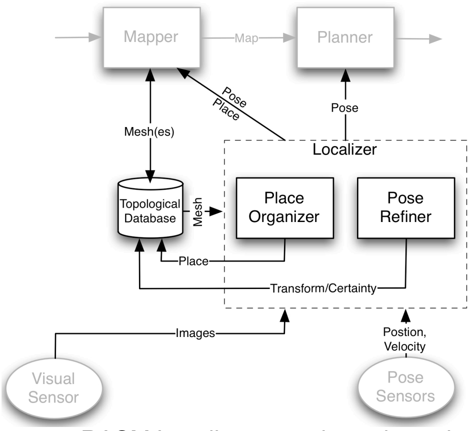

Figure 7. RASM Localizer organizes places in the topological model and acts to refine estimate of rover pose and the transforms between places.

Organizing Places

The relationships between adjacent places form a topological graph. A place can be defined as a location from which a set of distinct features can be observed. In RASM, database queries are used to determine adjacency in the topological model. Given a place, it can determine nearby places, which are needed for mapping and to refine pose. The topological model aggregates regions by linking references to similar places.

With a metric for similarity, either visual [10] or geometric [11], it is possible to recognize when the rover has returned to previously visited places (loop closure). It is also possible to compute the relative novelty of a given scene indicating when new places should be added. Large-scale relationships are learned by analyzing which local places are associated in time. By increasing the associating time and distance between observations the area covered by a place is increased.

Knowledge of pose can restrict which vertices in the topological graph warrant detailed investigation or updating in the face of new evidence. The topological model reduces confusion that can be caused by entering new, spatially distinct but visually similar places. This is an important aspect when navigating large areas of natural terrain like deserts.

Refining Pose

The Localizer also refines pose. It traverses places in the topological model and uses both geometric and visual features to simultaneously refine the alignment between places and correct the history of rover poses. Constraints, such as multiple observations of the same place, form loop closures and allow back propagated corrections by minimizing error between the alignment of places.

Simultaneous localization and mapping draws estimation of location, including pose, into the core structure of RASM. In recent experiments, the Localizer can refine terrain meshes from the database along with relative pose between observations. As these poses are updated the Mapper remeshes the terrain data into a current best map. Loop closure can be achieved from geometric and visual features to propagate pose and map correction through the topologic model. (Figure 8)

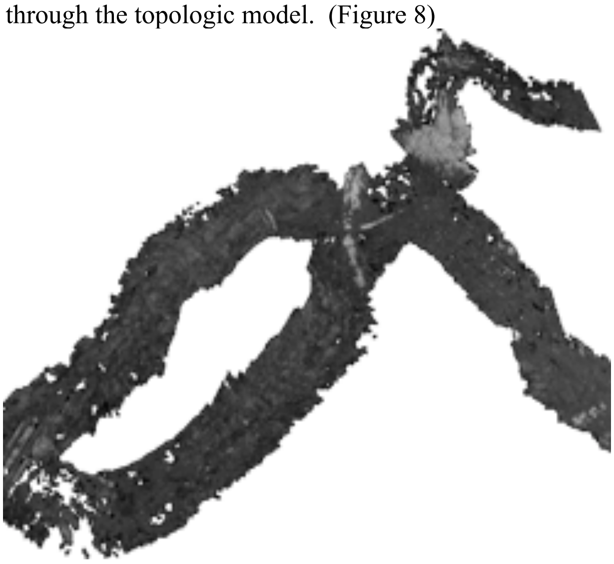

Figure 8. Loop closure using geometric and visual cues on 100m path in natural terrain.

The use of a topological structure gives the Localizer flexibility in how it optimizes transforms. For example, it can focus first on the region immediately surrounding the vehicle to produce local refinements and then when compute cycles are available, further refinement of larger areas can occur.

In this architectural design, the Mapper queries the Localizer for references to places in the vicinity of the rover. The Mapper retrieves the meshes that correspond to these places and can then merge the meshes to a single map for more efficient path planning, for example, by further simplifying mesh regions that are currently distant from the vehicle. Working together to estimate and refine rover and place locations, the best relative terrain maps are produced upon which navigation planning can proceed. (Figure 9)

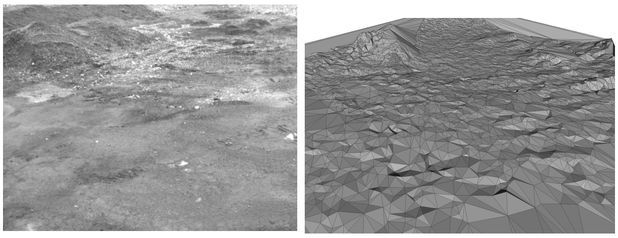

Figure 9. Rover image of terrain with mounds several meters ahead (left) and terrain map (right).

3.3. Planning

The RASM planning algorithms evaluate possible immediate motions and the subsequent path to the goal to determine the next best move. A small increment of this move is actually executed before the process repeats, with new information and possibly corrected positions. The planning cycle has two steps: first the rover's possible motions are evaluated and then the remaining path to the goal is considered.

Goals are defined as a location with some tolerance—a radius that defines the minimum acceptable distance of arrival—but without orientation. Fine motions and orientation at the goal will require a motion planner different from what RASM is designed to achieve.

A variety of metrics can be used to compute cost and then to combine cost between motion and path evaluation. The most basic distance metric and weighted combination can be effective in path selection, although more complex metrics, incorporating expected power or estimated risk have been employed. [12] Using only the simple distance minimizing metric, RASM has been able to choose paths that avoid obstacles and terrain extremes.

Motion Evaluation

The first part of navigation planning is to evaluation the actions that the rover can take to drive the next few meters. For each unique rover, as constrained by its kinematics and, potentially, dynamics, a set of candidate actions will be evaluated. (Figure 10) These actions can be predetermined or they can be modified or even generated on the fly. These driving actions are evaluated relative to the terrain by simulating the contact in small increments. At each increment configurations that exceed capability limits are disallowed, for example from collision or extremes of chassis motion.

This type of forward-simulation of vehicle-terrain interaction is used to evaluate motion, as conceptually described in Kelly et al. [13], although RASM applies different metrics and evaluation. The idea is to simulate rover actions with vehicle and terrain models to determine possible failures. Motion evaluation must incorporate exact rover kinematics to correctly predict collision, including correctly determining straddling or surmounting obstacles.

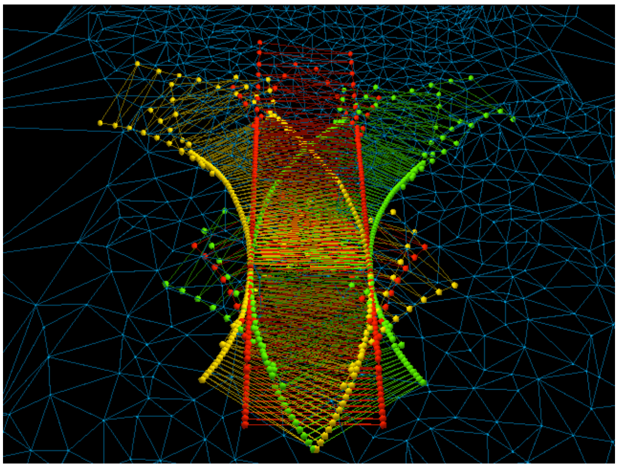

Figure 10. Driving actions (arcs, point turns) are specific to rover kinematics and then applied by the planner in evaluating maneuvers. In this image, arcs of different radius and direction are evaluated for the Zoë rover.

The relative merit of allowable motions, in terms of distance/time, risk or other factors is computed. This is used in selection and generation of drive commands that avoid obstacles but make the most progress to the goal. (Figure 11)

Path Planning

Path planning algorithms generate and evaluate the cost to reaching the next goal traveling over intermediate terrain. The Planner applies search on the terrain mesh from the endpoint of each of the motions being considered to the goal. This search may be across detailed meshes in the near field and more sparsely observed terrain in the far field. If the goal is beyond the currently observed terrain (and no prior models are available) then the direct distance to the goal is used—in effect this distance is all that is really known. In practice sufficient far-field observations are made so that reliable obstacle avoiding behaviors are exhibited.

For planning paths, Pimenta, et al. used a triangulated planar map of traversal cost to achieve variable resolution, primarily for path planning efficiency [14]. RASM considers 3D meshes and applies A* search, as does Dupuis et al. [15] to find the optimal path. In applications where frequent mesh updates or limited computing make incremental search advantageous, it is also possible to use D* [16]. As an alternative Gringras et al. applied potential fields on a triangular mesh to determine a path to the goal. [17] Unrefined search on a mesh usually results in a discontinuous path with many point turns required. The fluid-flow approach provides continuous paths. In comparison, RASM controls the rover in a continuous-motion, continuous-steering manner sending drive commands to the rover Motion Controller [18].

RASM performs action selection in the near field and then graph search over the far range to the goal. Its arcs and paths are can be smooth for power and time efficiency. The frequent cycling of the Planner also closes a loop on driving actions and tends to smooth the resulting rover path.

RASM navigates continuously to the goal, rapidly updating the driving direction and speed of the rover. It does not perform path tracking in the sense of closing a servo loop to minimize error between desired and achieved path, and specifically it does not rely on the commands it generates to be faithfully executed either individually or in sequence. Instead, RASM evaluates paths all the way to the goal at each cycle and commands one action toward that goal before repeating the process. RASM does close a servo loop on the rover path to the goal. Its open loop execution is constrained by rapid re-evaluation of action. At approximately 4 Hz the Planner generates a drive action for the rover to execute.

4. SUMMARY

The Reliable Autonomous Surface Mobility (RASM) architecture and algorithms have evolved from research and application of planetary rover navigation over the past decade. This architecture and specific algorithms in

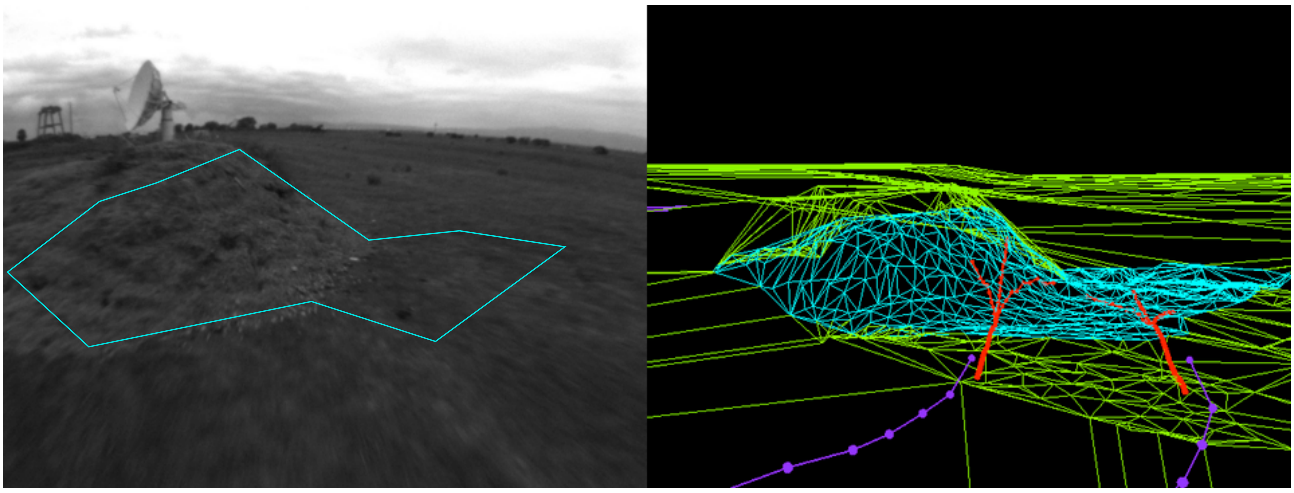

Figure 11. Rover image of mound at left. Corresponding mesh (green) with recent observation (cyan) and path showing past motions in purple and next commanded motion in red. From ~500m traverse at NASA Ames Research Center.

RASM has been selected with a broad set of rover requirements for guidance and constraint. We intend to evaluate RASM against these requirements.

In order to achieve continuous navigation, RASM associates terrain observations in a topological structure maintained in a database. These sensor observations are efficiently decimated, meshed, reduced and then merged to produce a geometric model of the terrain. Various refinements to surface of the terrain (using ICP) and the position of the rover and terrain (using SLAM) are possible. With model of terrain and model of the rover, a planner can determine admissible actions over the short term while evaluating which of those actions best advances the rover toward is goal in the long term. The process cycles at high rate to continually correct and smooth motion and to drive the rover toward its goal

Technology has advanced to the point where autonomous systems can reliably navigate over long distances in barren terrain. In our work we are striving toward a framework that can be applied to different rovers with different sensors for missions of exploration. There is more effort required to validate our reliability requirements quantitively but we believe the results thus far give confidence.

5. ACKNOWLEDGMENTS

We acknowledge and appreciate the technical assistance of Dimitrios Apostolopoulos at ProtoInnovations LLC; Greydon Foil, Michael Furlong, Dominic Jonak and James Lee at Carnegie Mellon University; and Peter Iles and Kirk MacTavish at Neptec Design Group.

Artemis Jr. And Juno rovers were created by Neptec. LATUV and K-REX rovers designed and built by ProtoInnovations. Images of Artemis Jr. courtesy Neptec and image of LATUV courtesy ProtoInnovations.

This research was supported by NASA under grants NNX11-CC51C (through ProtoInnovations LLC) STTR program and NNX11-AJ87G, ASTEP program, Mary Voytek, Program Executive and in collaboration with the NASA Ames Research Center's Intelligent Robotics Group, Terry Fong, Group Lead.

6. REFERENCES

[1] Wettergreen, D., et. al., "Sun Synchronous Robotic Exploration: Technical Description and Field Experimentation," *Intl. Journal of Robotics Research*, January 2005.

[2] Wettergreen, D., et al., "Second Experiments in the Robotic Investigation of Life in the Atacama Desert of Chile" iSAIRAS, Munich, September 2005.

[3] Wettergreen, D., et al., "Long-distance Autonomous Survey and Mapping in Robotic Investigation of Life in the Atacama," iSAIRAS, Los Angeles, February 2008.

[4]Pauly, M. Gross, M., and Kobbelt, L.P., "Efficient Simplification of Point-Sampled Surfaces," http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.19.5512

[5] Chew, L., "Constrained Delaunay Triangulation," Algorithmica, Volume 4, Numbers 1-4, Springer, New York, June 1989.

[6] M. Garland, "Quadric-Based Polygonal Surface Simplification", Ph.D. thesis available online at http://mgarland.org/files/papers/thesis-onscreen.pdf

[7] Fayek, R., Wong, A., "Using hypergraph knowledge representation for natural terrain robot navigation and path planning," International Conference on Robotics and Automation, Minneapolis, April 1996.

[8] Bakambu, J.N., Allard, P., Dupuis, E., "3D Terrain Modeling for Rover Localization and Navigation," 3rd Canadian Conference on Computer and Robot Vision, 2006.

[9] Zhang, Z., "Iterative Point Matching for Registration of Free-Form Curves", IRA Research Reports, No. 1658, Program 4: Robotics, Image and Vision, 1992.

[10] Cummins, M., Newman, P. "FAB-MAP: Probabilistic Localization and Mapping in the Space of Appearance", The International Journal of Robotics, Volume 27, Number 6, June 2008

[11] Brunskill, E., Kollar, T., Roy, N., "Topolgical Mapping Using Spectral Clustering and Classification", Intelligent Robots and Systems, IROS, 2007

[12] Tompkins, P., et al., "Mission-Level Path Planning and Re-Planning for Rover Exploration," *Robotics and Autonomous Systems*, January 2006

[13] Kelly, A., and Stentz, A., "An Approach to Rough Terrain Autonomous Mobility," *Autonomous Robots*, April, 1998

[14] Pimenta, L., Pereira, G., and Mesquita, R. "Fully continuous vector fields for mobile robot navigation on sequences of discrete triangular regions," IEEE International Conference on Robotics and Automation, Rome, April 2007.

[15] Dupuis, E., Rekleitis, I., et al., "Over-The-Horizon Autonomous Rover Navigation: Experimental Results," iSAIRAS, Los Angeles, February 2008.

[16] Stentz, A., "The Focused D* Algorithm for Real-Time Replanning," 14th Intl. Joint Conference on Artificial Intelligence, Montreal, August 1995.

[17] Gingras, D., Dupuis, E., Payre, G., de Lafontaine, J., "Path Planning Based on Fluid Mechanics for Mobile Robots Using Unstructured Terrain Models," IEEE International Conference on Robotics and Automation, Anchorage, May 2010.

[18] Wettergreen, D., et al., "Design and Field Experimentation of a Lunar Prospecting Rover," *Intl. Journal of Robotics Research*, October 2010.