Multi-Layer Atlas System for Map Management

Jean-Luc Bedwani
Institut de recherche d'Hydro-Québec,
bedwani.jean-luc@ireq.ca

Ioannis Rekleitis
School of Computer Science, McGill University,
yiannis@cim.mcgill.ca

François Michaud
Faculté de génie, Université de Sherbrooke,
Francois.Michaud@USherbrooke.ca

Érick Dupuis
Canadian Space Agency,
Erick.Dupuis@asc-csa.gc.ca

Abstract

Next generation planetary rovers will require greater autonomous navigation capabilities. Such requirements imply the management of potentially large and rich geo-referenced data sets stored in the form of maps. This paper presents the design of a data management system that can be used in the implementation of autonomous navigation schemes for planetary rovers. It also outlines an approach that dynamically manages a variety of data content and the uncertainty of the spatial relationship between two maps; in addition the proposed framework provides basic path planning operations through maps, and the correlation of maps in localization operations. Timing results from a rich data set demonstrate the efficiency of the proposed framework. In addition, experimental results on the usage of our Atlas management system by a rover performing autonomous navigation operations are also presented.

1 Introduction

In December 2003 and January 2004, the Mars Exploration Rovers (MERs) "Spirit" and "Opportunity" landed on Mars and conducted, for the first time, semi-autonomous exploration of another planet [1]. This was a major step in robotics as all previous robotic missions (Lunokhod [4], Viking [8], Sojourner [11]) had not implemented any significant on-board autonomy. The next missions planned to Mars are NASA's Mars Science Laboratory (MSL) in 2011 [22] and the European Space Agency's (ESA) ExoMars in 2016 [21]. Both MSL and ExoMars have set requirements to travel up to one kilometer per day.

Given the extremely high cost of these missions, great efforts are made by the scientific teams to make efficient use of the robots lifetime, capabilities, and resources. One of the main problems faced by these robotic probes is the limitations of the communication link with Earth: the round trip

Figure 1. The Mars emulation terrain with our modified P2AT rover.

communication delays range between 8 and 40 minutes; the communication windows last approximately one hour and are spaced 12 hours apart. Finally, the bandwidth is very narrow [16]. Traditional concepts of operation require intense human interaction with operators on Earth before accomplishing an operation. Historically, three communication windows were typically required to reach a rock once it has been identified by the scientific team. This is one of the main incentives for increasing the navigation autonomy of planetary robots. Providing planetary exploration rovers with the capability to navigate in an unknown, or partially known, terrain autonomously would provide more efficient operations than the current operational mode. While disconnected from Earth, the rover must be capable of traveling autonomously towards one or more locations using high-level commands such as: map this region, analyze this set of features, etc. This would maximize the bandwidth during the communication windows for returning scientific data.

Autonomous long-range navigation for a rover implies the ability to map the environment, localize itself and plan trajectories to reach different goals. During these opera-

978-0-7695-4040-5/10 $26.00 © 2010 Crown Copyright
DOI 10.1109/CRV.2010.34

tions, 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. Figure 1 shows our robot equipped with a 360° LIDAR sensor and a thermal camera exploring the Mars emulation terrain located at the Canadian Space Agency (CSA).

Our approach addresses the problem of data management by maintaining each individual data set together with a network of spatial relations between the different data sets. Depending on the uncertainty accumulated between different sensing data, the spatial relationships are treated differently. For example, data collected from different sensors from the same location and at the same time are affected only by the sensor uncertainty, while data collected from different locations contain errors due to the robot's localization error in addition to the sensor noise. Each individual data set represents a local map. Operations combining several local maps to produce an integrated map for planning purposes must be supported by a framework. During operations, as the number of local maps increases an efficient management system is required to maintain world consistency.

The objective of our work is to design such a data management system that can be used in the development of autonomous navigation schemes for planetary rovers. The presented solution is capable of dynamically managing a large set of maps while considering the problems related to cartographic operations and uses. The issues considered are: the broad variety of data content, uncertainty in the spatial relationship between maps, trajectory planning through multiple maps, and the correlation of maps in localization operations.

Related work on map management systems is presented in Section 2. Section 3 outlines the structure of the map management system developed at CSA. Experimental results are presented in Section 4, and a discussion of the implementation and future work is given in Section 5.

2 Mapping a World

Map management systems appear in the literature under different names: atlases, modeling environments, databases. Sometimes such systems are merged into the localization and mapping process.

Bose et al. [2] proposed a map management system based on the concept of a conventional atlas for large-scale cyclic environments. Their framework is based on a hybrid, hierarchical metric/topological map system. At a higher level, a graph is used to represent the world, with graph vertices representing locations, and edges representing the transformation between locations. At a lower level, each location is described as a local 2D metric map. Uncertainty therefore is modeled in each map, and not with respect to a global reference frame. The uncertainty model is based on a Gaussian representation of the error. The proposed atlas framework is used to transform state uncertainty between local maps, create new maps, handle loop closing situations, and to generate and evaluate competing hypotheses of the local map for representing the current system state.

A similar atlas-like system was proposed by Lisien et al. [9]. At the higher level, a topological representation is traced on the Generalized Voronoi Graph (GVG) of the mapped environment. However, the graph edges are represented at the lower level as a collection of features. The GVG has the advantage to be embedded in the environment, and can be easily calculated during exploration. Moreover, the collection of features of an edge-map represents the local area from the perspective of one GVG node, towards a neighboring node. Each edge-map represents discontinuities in the environment as a relative pose and a Gaussian uncertainty. Such a representation has the advantage of being scalable in memory and computation, but is highly sensitive to changes in the environment and is constrained to non-wide, feature-rich, indoor spaces.

Hierarchical arrangements of maps are explored differently in Kelly et al. [7], where three levels of maps are maintained, providing decreasing periods of data accumulation and increasing levels of details accordingly. The lower level maps are centered on the robot, while the higher-level map is a global representation of the world. The three maps used are a volumetric grid, a local elevation map and a global traversability map. The creation/update of the map contents flows from the highest detailed map towards the global map. Moreover, the global map is composed of multiple channels. Therefore, traversability maps from different sensors and resolutions can be presented on the same map: for example, data from the robot, from a remote unmanned air vehicle (UAV), and from available satellite information can be presented in one map. The hierarchical map arrangement has the advantage of permitting relocalization of the global map channels without affecting the lower-level maps. It also bounds the memory requirement of maintaining a complete 3D map of the world. On the other hand, this system fuses the data during the update process of the maps, which may result in corruption of their content via the permanent introduction of noise.

The SimScape terrain modeling toolkit [6] from the Jet

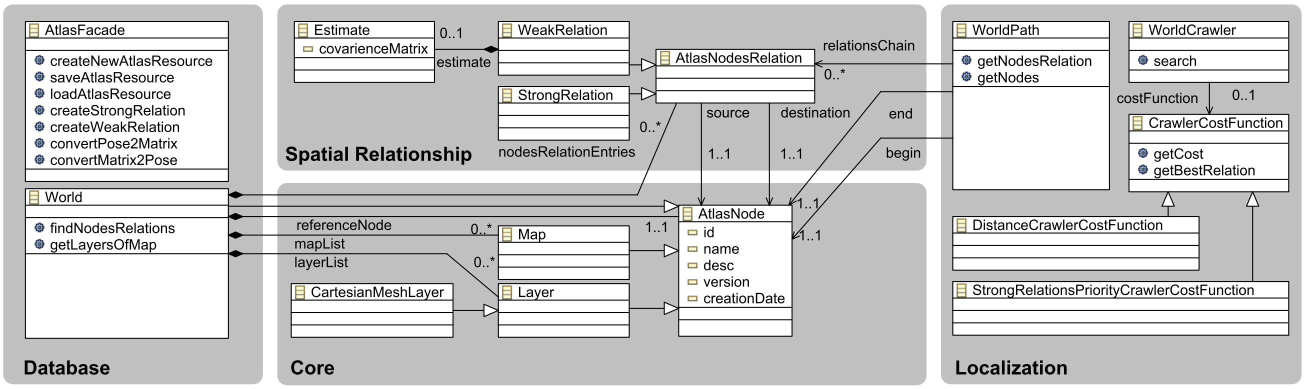


Figure 2. UML diagram of our multi-layer atlas management system.

Propulsion Laboratory (JPL) provides a common infrastructure to represent terrain model data from multiple data sources and make them available to simulation applications. This toolkit supports multiple representations of the terrain geometry such as 2.5D digital elevation maps, point clouds, 3D meshes, and 2.5D irregular triangular mesh (ITM). It also provides transformations between different terrain model representations, and generation of composite terrain models from heterogeneous models. Despite the many capabilities of this toolkit, it is oriented toward simulation applications and does not provide any management capabilities of map registration error.

Hybrid Maps were used in an approach termed DenseSLAM [14] that combined the sensor information in several layers with focus on facilitating simultaneous localization and mapping of a mobile robot.

Finally, the System for Unifying Multiresolution Models and Integrating Three-dimensional Terrains (SUMMITT) is a suite of software tools providing terrain modeling functionality for the MERs operators [15]. This immersive environment handles images of multiple resolutions and scales available from a variety of sources: descent images, surface images from the lander and the rover, as well as orbital images. For a given exploration site, SUMMITT manages the image registration process, the conversion of stereo images to volumetric primitives (voxels), and the storage/merging of the voxels as an octree structure. Once the site is modeled in the octree, SUMMITT is capable of producing polygonal representation of the site (for visualization and collision avoidance) as well as an elevation map. Unfortunately, this system is not dynamic regarding the evolution of its data content. Once registered, the data is merged and therefore, cannot be changed unless the complete database is re-registered and re-built. Finally, the intent for this suite of tools is to be used by human operators and is not integrated to be used by an autonomous robotic system.

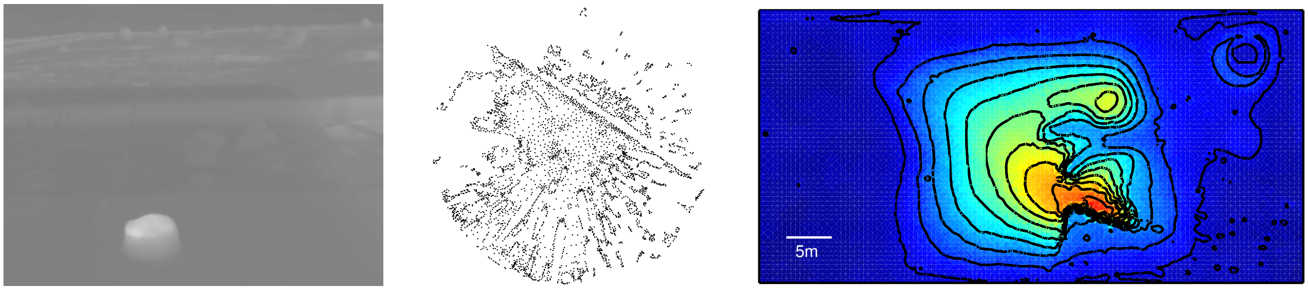
3 Atlas Structure

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 current implementation of the Atlas is divided into four key components: the *Core* component defines the atomic elements of the system; the *Database* component is used to store and access the different data within the system; the *Spatial Relationship* component incorporates the concept of weak/strong geographic relations between elements of the Atlas; and the functionality of estimating the location of an arbitrary data set in relation to the coordinate frame of another one is implemented in the *Localization* component. Figure 2 presents a UML diagram of the Atlas structure.

Figure 3 presents representative data sets stored in the Atlas used at CSA for the experiments leading to the Avatar Explore mission [10] [1]. Fig. 3a contains a single infrared image which is used to detect geological interesting locations for closer inspection by the rover. For this experiment, a heat source was placed in the terrain to provide a clearly identifiable target. A 360° LIDAR scan from a single location is displayed in Fig. 3b; such scans are used for safe path-planning by the rover. Finally, Fig. 3c 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.

[1] In Avatar Explore, a Canadian astronaut on orbit in the International Space Station communicates and sends high level commands to a rover operating at CSA's Mars emulation terrain. 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.



(a) Thermal image. (b) 3D LIDAR scan. (c) Topographical map.

Figure 3. 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.

3.1 A hierarchy of data abstraction

Each one of the geo-referenced data sets represents a specific area of the environment, and it is encoded in the Layer component. For a specific region, several different Layers can be available: some originate from different sensors, some are taken from different vantage points, and some are collected at different times. All the different Layers that describe a specific region are associated with the Map component, which encodes all the available information about a specific area. Finally, the collection of all relevant Maps for the operating environment of the robot are connected with the World component that belongs to the *Database*. All three components (Layer, Map, and World) inherit from a common component, i.e., the AtlasNode, which contains fundamental information such as name, description, and creation timestamp of when the data was collected. It is through the AtlasNode data structure that the network of spatial relationships is realized. In particular, the World coordinate system represents the global reference frame to which the Map coordinate systems are relative. It is worth noting that the current implementation of the Atlas system supports multiple data formats, differences in resolution, precision, uncertainty, and the temporal properties of each data set, all of which are encoded in the Layer component.

3.2 Keeping track of the data

The *Database* component handles storage and access operations of the Atlas system. It provides loading and saving of different collections of data sets organized as Worlds, each one describing coherently the operating environment of the rover. In addition, the *Database* is responsible for transforming the online data into operator-readable XML files for saving and also for uploading to a remote location.

3.3 Linking the data as a graph

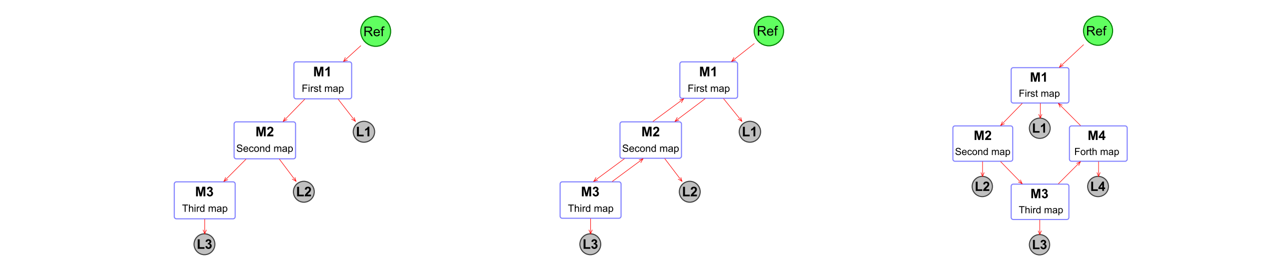
Central to the Atlas framework is the organization of the different data sets based on their spatial relationship. Each AtlasNode element can serve as a vertex in a graph where edges represent spatial relations. These relationships can be classified as strong or weak depending on the circumstances of acquiring the data. A StrongRelation exists for data resulting from a single sensor reading. Such a relationship is affected only by the sensing error. Moreover, when different sensor readings are acquired at the same time and location, their spatial relationship is affected only by alignment error [13].

WeakRelation represents transformations with greater localization uncertainty, such as sensor reading taken from different locations. It is represented as a transformation associated with a covariance matrix that encodes the pose uncertainty.

The *Localization* component of the Atlas system provides the functionality of combining a series of relative relations, referenced above, into a simple transformation. The pose (position and orientation) of an arbitrary AtlasNodes can be transformed to an arbitrary reference frame by following a sequence of local coordinate frame transformations. The recorded uncertainty can be used to plan paths that offer the maximum benefit in relocalization as in [12]. It is worth noting that spatial relations between AtlasNodes can be the results of odometry information, scan to scan matching, or a combination of the two. Furthermore, by using the WorldCrawler functionality, the graph of spatial relations can be traversed using a CrawlerCostFunction that favors paths that reduce uncertainty.

3.4 Atlas compared to others

Table 1 outlines the features present in the different data management systems discussed in section 2. Most of the systems simply lack some properties that are required for autonomous planetary exploration.



(a) Simple maps chain. (b) Double maps chain. (c) Loop of maps.

Figure 4. Visualization of different maps configuration topology. The green circle tagged "Ref" represents the WORLD reference element, blue rounded rectangles represent MAP elements, while grayed circles represent LAYER elements. The red arrows correspond to an ATLASNODESRELATION element.

System	Database	Multiple data	Lossless	Uncertainty	Dynamic	3D	Extension	Multi platform	Operator	Autonomous
[2]	•	•	•	•	•					•
[9]					•					•
[7]		•			•	•			•	•
[6]	•	•	•			•	•		•	
[15]	•	•	•			•			•	
Atlas	•	•	•	•	•	•	•	•	•	•

Table 1. Comparison of different systems with our Atlas. Presence of a characteristic is marked by a • symbol.

4 Experimental Results

The implementation and testing of the Atlas structure was done using the *Eclipse Modeling Framework* [3] (EMF). An Atlas database was created using this framework to verify that each element could be added and accessed appropriately. As presented by Fig. 4, a simple database viewer was generated using the *Graphical Modeling Framework*[2] (GMF) in order to view and manipulate such EMF data structures. The complete implementation was deployed on the Canadian Space Agency's mobility platform shown in Figure 1.

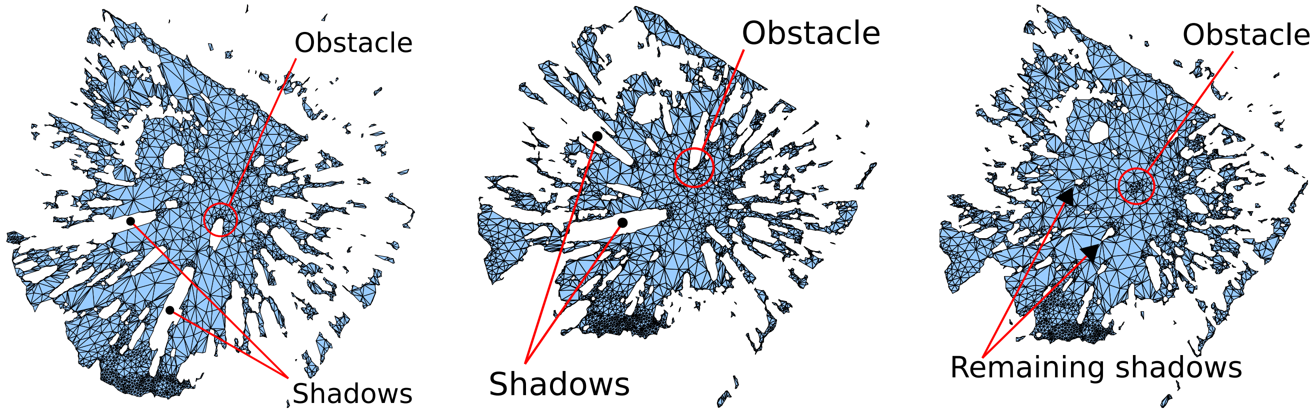
In order to validate the usability of the Atlas system, a set of use cases were defined: loading an existing Atlas, creating a new Atlas with a navigation scheme, updating the Atlas with visual odometry, etc. These use cases were performed using the 2.5D laser range data and 3D odometry information collected during previous robotic experiments [18]. The objective of the off-line experiments was to ensure that a large set of geo-referenced data (94 scans) can be easily handled by this system and to verify that the registration of the different data sets are preserved when using relative instead of global registration. The use of relative registration implies additional computational effort in order to obtain a map pose. This effort is therefore quantified in terms of time.

Finally, the system was tested on its information retrieval capabilities. A set of random points was generated, and for each random point the Atlas was queried to return all the scans that contain data near the selected point. The Atlas query consists of retrieving all the data sets origins in the proper frame of reference, sorting them by distance to the specified random point, and retrieving the corresponding scans data for maps within a 15 meters range. As shown in table 2, the query time was on average 53 seconds with a standard deviation of 13 seconds when using meshes of 50,000 triangles. However, if only the first match is required, this process is done within few seconds. The results in table 2 follow a non-Gaussian distribution, this is due to different factors. For small times a non-neglectable systematic error is present due to the use of Java, an interpreted language that manages its own memory. For processes such as search or the Kd-ICP, the timing is a function of the map's density in a region and of the number of iterations required to converge.

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 Fig. 5a,b). By requesting one or more additional 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. 5c), where most of the shadows are eliminated. The search query of scans covering a specific point is utilized for retrieving relevant scans.

Experimental validation of the Atlas system was performed on a Pioneer P2AT platform enhanced with an IMU, a 360° fov LIDAR, and a Intel® Core^{TM}2 Duo processing

[2] See official web site http://www.eclipse.org/gmf



(a) Initial position scan. (b) Second position scan. (c) Merged scans.

Figure 5. (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.

unit (1.6 GHz, 3 GB of RAM). They were run on the Canadian Space Agency's testing grounds. The terrain is 60 m by 30 m and emulates the topography of a broad variety of Martian landscapes [20]. The goal is to perform an autonomous navigation [18, 19] by employing the Atlas system to store and register maps (using 3D odometry for the pose estimation) while traversing unknown terrain. When visual matching for two neighbors maps is available (using a variant of the ICP algorithm [5] called Kd-ICP [17]), the Atlas updates its content by adding a relation between these two maps, thus creating a double linked chain of maps as shown in Fig. 4b.

Figure 6 presents an illustrative example of the Atlas framework in practice. The rover started at a known location and traversed to a final destination beyond its sensing horizon. During the experiment, there was no operator intervention after the selection of the final destination. The Atlas was used to store the different scans and to also provide updated spatial relationships when the Kd-ICP algorithm was employed. Figure 6a presents the raw data from the nine scans that were acquired during the experiment by using the robot odometry as registration information. As can be seen by the discrepancies in the shades of colors, the scans do not intersect properly on a common surface. Therefore, planning a transition from one scan to the next is a difficult process. Figure 6b presents the same nine scan extracted from the Atlas system while using the Kd-ICP corrected registration. They were transformed to be in the same coordinate frame using all available information. The different surfaces are well aligned and transitioning from one data set to the next is trivial.

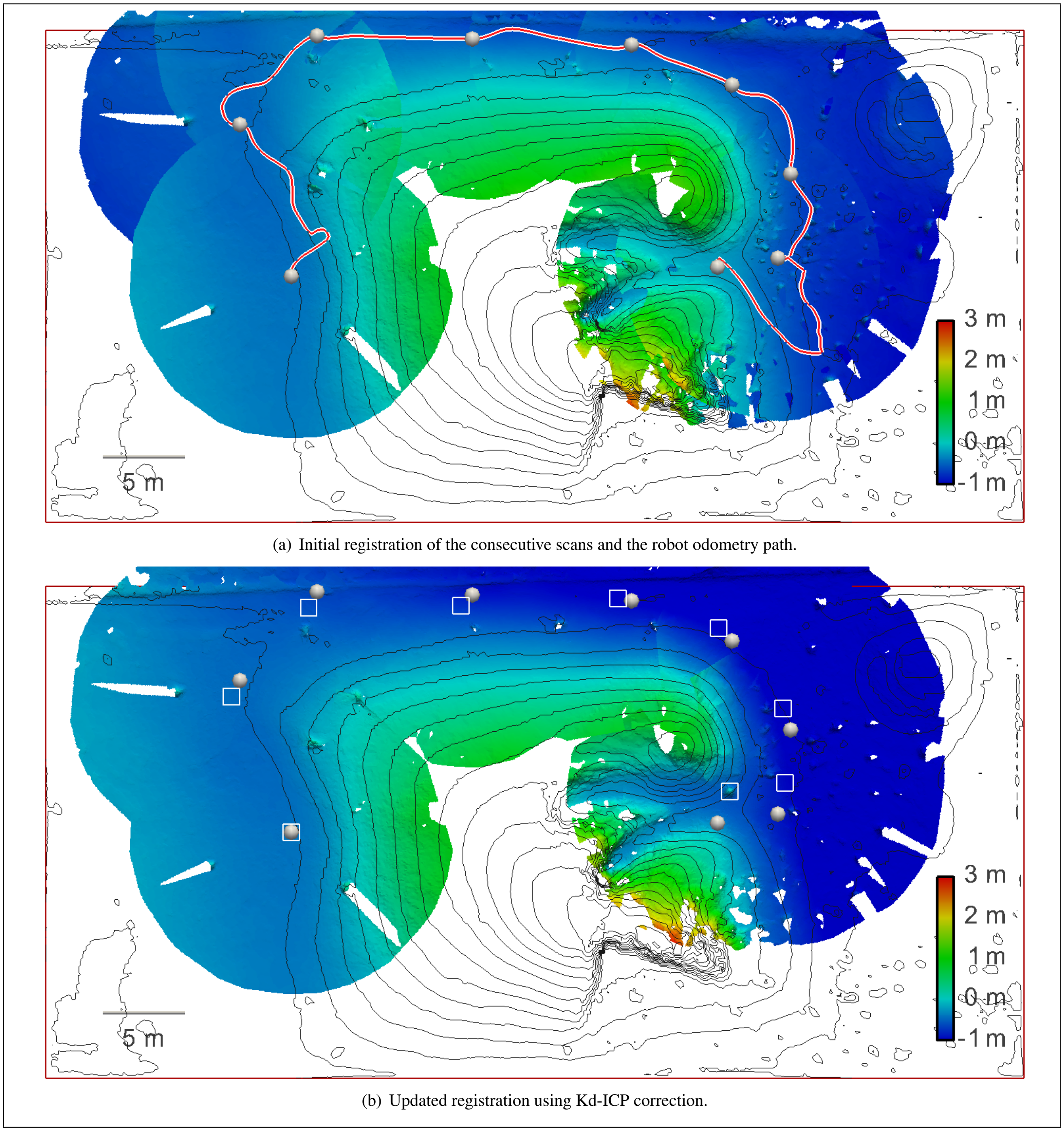
Table 2. Mean values±standard deviation of timing in milliseconds for different Atlas operations using meshes of three different resolutions. Results using a database of 94 different meshes.

Operations	10k cells (ms)	25k cells (ms)	50k cells (ms)
Read	306±80	733±146	1,430±311
Write	304±157	1,196±398	3,318±1,130
Relation	0.35±0.14	0.45±0.9	0.48±1.37
Search first	887±1,150	2,040±2,860	3,770±5,230
Search all	9,880±2,600	26,200±7,020	53,000±13,400
Kd-ICP	6,630±4,280	22,500±17,900	65,400±56,600

5 Conclusion and Future Work

This work demonstrates the feasibility of a data management system suitable for robotic planetary exploration. The main features of our approach 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 provides series of maps linking two locations in planning operations, and the functionality to correlate maps in localization operations.

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 rel-



(a) Initial registration of the consecutive scans and the robot odometry path.

(b) Updated registration using Kd-ICP correction.

Figure 6. Atlas management system used to present an elevation map composed of nine scans collected during an autonomous navigation experiment. Scans registration is done using robot odometry and corrected using visual correlation with Kd-ICP algorithm. Circles (spheres) represent the odometry estimate for the scan origin, squares represent the updated origin from the scan matching. For the first scan (lower left) the two estimates coincide, then, the odometry estimate increasingly deviates.

ative relations instead of global relations, the Atlas system automatically propagates to neighboring maps any improvements/changes deriving from the updated relation between two maps.

Experimental results of the Atlas management system showed its capability to provide the required information in a timely fashion. In addition, the robust handling of the uncertainty relations between sub-maps led to more accurate and efficient navigation behavior.

The Atlas system introduced in this paper has several interesting additional features which are scheduled for future implementations. One example is the capability to automatically convert data from one type to another, i.e., converting a mesh layer to an occupancy grid layer. Another improvement is to use a binary database instead of XML to minimize communication bandwidth with a planetary rover. Future robotic experiments using the Atlas system are scheduled for human supervised autonomous exploration experiments. The Atlas system will provide a common framework to store, use, exchange, and visualize a broad variety of georeferenced data (thermal, visual, and 2.5D range data) to both human and robotic clients.

Acknowledgment

We would like to thank Régent l'Archevêque and Pierre Allard for their many suggestions during the design and implementation of the Atlas management system, Tom Lamarche and Sébastien Gemme for their support in providing a reliable LIDAR system and tools, David Gingras and Alessio Salerno for their support during field testings. Finally, thanks to François Pomerleau who kindly provided us with a reliable, efficient and robust implementation of the Kd-ICP algorithm. F. Michaud holds the Canada Research Chair in Mobile Robotics and Autonomous Intelligent Systems.

References

[1] J. J. Biesiadecki, P. C. Leger, and M. W. Maimone. Tradeoffs between directed and autonomous driving on the mars exploration rovers. *The Int. Journal of Robotics Research*, 26(1):91–104, 2007.

[2] M. C. Bosse, et. al. SLAM in large-scale cyclic environments using the atlas framework. *The Int. Journal of Robotics Research*, 23(12):1113–1139, 2004.

[3] F. Budinsky, S. Brodsky, and E. Merks. *Eclipse modeling framework*. Pearson Education, 2003.

[4] W. D. Carrier. Soviet rover systems. In *Space Programs and Technologies Conf.*, page 9. American Institute of Aeronautics and Astronautics, 1992.

[5] D. Chetverikov, D. Svirko, D. Stepanov, and P. Krsek. The trimmed iterative closest point algorithm. In *Proc. of the Int. Conf. on Pattern Recognition*, vol. 3, pg 545–548, 2002.

[6] A. Jain, et. al. SimScape terrain modeling toolkit. In *Proc. of the 2nd IEEE Int. Conf. on Space Mission Challenges for Information Technology*, pp 8, 2006.

[7] A. Kelly, et. al. Toward reliable off road autonomous vehicles operating in challenging environments. *The Int. Journal of Robotics Research*, 25(5-6):449–483, 2006.

[8] H. P. Klein, J. Lederberg, A. Rich, V. I. Oyama, and G. V. Levin. The viking mission search for life on Mars. *Nature*, 262:24–27, 1976.

[9] B. Lisien, D. Morales, D. Silver, G. Kantor, I. M. Rekleitis, and H. Choset. The hierarchical atlas. *IEEE Transactions on Robotics*, 21(3):473–481, 2005.

[10] E. Martin, R. L'Archevêque, S. Gemme, I. Rekleitis, and E. Dupuis. The avatar project: Remote robotic operations conducted from the international space station. *IEEE Robotics and Automation Magazine*, 14(4):20–27, 2008.

[11] J. Matijevic and D. Shirley. The mission and operation of the Mars pathfinder microrover. *Control Engineering Practice*, 5(6):827–835, 1997.

[12] D. Meger, I. Rekleitis, and G. Dudek. Heuristic search planning to reduce exploration uncertainty. In Proc. of *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems,* pg 3382 – 3399, 2008.

[13] F. Mirzaei, A. Mourikis, and S. Roumeliotis. On the performance of multi-robot target tracking. In *Proc. of the IEEE Int. Conf. on Robotics and Automation*, pg 3482–3489, 2007.

[14] J. Nieto, J. Guivant, and E. Nebot. DenseSLAM: Simultaneous localization and dense mapping. *The Int. Journal of Robotics Research*, 25(8):711–744, 2006.

[15] C. F. Olson, L. H. Matthies, J. R. Wright, R. Li, and K. Di. Visual terrain mapping for Mars exploration. *Computer Vision and Image Understranding*, 105(1):73–85, Jan. 2007.

[16] L. Pedersen, R. Sargent, M. Bualat, C. Kunz, S. Lee, and A. Wright. Single-cycle instrument deployment for Mars rovers. In *Proc. of the 7th Int. Symposium on Artificial Intelligence, Robotics and Automation in Space*, 2003.

[17] F. Pomerleau. Registration algorithm optimized for simultaneous localization and mapping. Master's thesis, Department of Electrical Engineering and Computer Engineering, Université de Sherbrooke, 2008.

[18] I. Rekleitis, J.-L. Bedwani, and E. Dupuis. Experimental results for over-the-horizon planetary exploration using a lidar sensor. In *Proc. of the 11th Int. Symposium on Experimental Robotics*, pg 65–77, Athens, Greece, 14-17 July 2008.

[19] I. Rekleitis, J.-L. Bedwani, and E. Dupuis. Autonomous planetary exploration using lidar data. In *Proc. of the IEEE Int. Conf. on Robotics and Automation*, pg 3025–3030, Kobe, Japan, 12-17 May 2009.

[20] I. Rekleitis, J.-L. Bedwani, E. Dupuis, and P. Allard. Path planning for planetary exploration. In *Proc. of the Conf. on Computer and Robot Vision*, pg 61–68, May 2008.

[21] J. Vago. Overview of ExoMars mission preparation. In *Proc. of the 8th ESA Workshop on Advanced Space Technologies in Robotics and Automation*, The Netherlands, 2004.

[22] R. Volpe. Rover functional autonomy development for the Mars mobile science laboratory. In *Proc. of the IEEE Aerospace Conf.*, volume 2, pg 643–652, 2006.