Proceedings of the 2001 IEEE
International Conference on Robotics & Automation
Seoul, Korea • May 21-26, 2001

The Science Autonomy System of the Nomad Robot

Michael D. Wagner, Dimitrios Apostolopoulos, Kimberly Shillcutt, Benjamin Shamah, Reid Simmons,
William “Red” Whittaker
email: {mwagner, da1v, bshamah, kimberly, reids, red}@ri.cmu.edu
Field Robotics Center, The Robotics Institute
Carnegie Mellon University
Pittsburgh, PA 15213
Phone: (412) 268-5354; Fax: (412) 268-5895

Abstract:

The Science Autonomy System (SAS) is a hierarchical control architecture for exploration and in situ science that integrates sensing, navigation, classification and mission planning. The Nomad robot demonstrated the capabilities of the SAS during a January 2000 expedition to Elephant Moraine, Antarctica where it accomplished the first meteorite discoveries made by a robot. In this paper, the structure and functionality of the three-tiered SAS are detailed. Results and lessons learned are presented with a focus on important future research.

Keywords: robotic meteorite search, science autonomy system, robot control architecture, planning

1. Introduction

The Science Autonomy System (SAS) was developed for the Nomad robot (Fig. 1) in response to the Robotic Search for Antarctic Meteorites project, which has developed technologies and strategies for autonomous meteorite search [1]. The SAS is a control architecture for exploration and *in situ* science that integrates sensing, navigation, classification and mission planning, enabling a meteorite searching robot to autonomously find and study surface samples while performing an exhaustive patterned search. Once an interesting new sample has been found, the SAS handles the deployment of sensors capable of discriminating terrestrial rocks from meteorites. Bayesian network classification software then calculates its confidence in the sample being a meteorite as well as metrics estimating the benefit of taking additional sensor data.

As the project’s final demonstration, Nomad was sent to Elephant Moraine, Antarctica to perform autonomous meteorite searches. Nomad operated between January 10 to 30, 2000. Over this time, ten individual demonstrations were performed along with many experiments and data gathering efforts. During the demonstrations, Nomad classified 42 samples with spectrometry. Of these samples, three meteorites were correctly classified. An additional two meteorites were correctly classified during tests performed without patterned searches. The SAS autonomously acquired new targets with a 79% success rate and deployed Nomad’s manipulator arm with a 72% success rate. These results prove the concept that the SAS enables autonomous exploration robots.

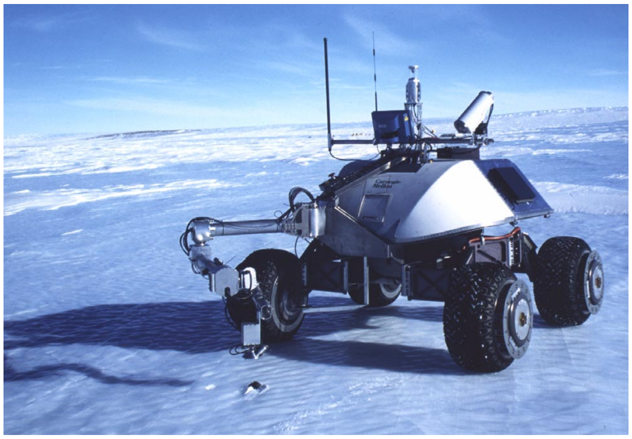

Figure 1. The Nomad robot searching for meteorites in Elephant Moraine, Antarctica

2. Science Autonomy System Design

2.1. Approach

Although the SAS was developed specifically for autonomous meteorite search, it is designed as a control architecture for a more general class of autonomous scientific exploration missions. Scientific exploration tasks may require vastly different types of sensors, actuators and data understanding algorithms. Scientific goals in the SAS are framed as classification problems. Given a set of sensor readings, objects are classified as belonging to certain known exemplar classes. Note that the details of how classification is performed should be abstracted from the rest of the system. The goal of the SAS is therefore to classify as many objects in the world as possible. This brings about the need for directed, exhaustive searches.

Different sensors on a robot are deployed with varying time and energy costs. For instance, imagery generally requires less time and energy to collect than reflection

0-7803-6475-9/01/$10.00© 2001 IEEE

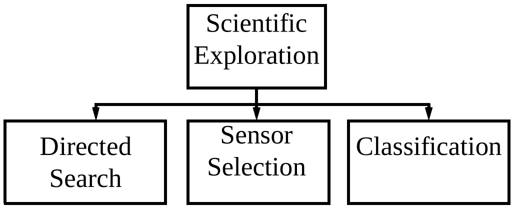

Figure 2. Scientific exploration task decomposition

spectrometry. It is therefore beneficial to intelligently select sensor usage because most field robots operate with limited time and energy resources. Different objects can often be classified with varying degrees of difficulty. For instance, a majority of meteorites are small and dark-colored. Chances are good that a costly spectrometer reading will only reinforce the fact that a large, white rock is terrestrial. To achieve intelligent sensor selection, the SAS utilizes the following approaches:

- Deploying sensors must be associated with some kind of a cost such as energy, time or digital storage space.
- The target classifier must provide an estimate of the information that would be gained by deploying all sensors in the system. This quantity, referred to as *information gain*, is weighed against sensor deployment costs to create action plans.

With this framework, the SAS decomposes scientific exploration tasks into directed search, sensor selection and classification tasks (Fig. 2).

2.2. System Architecture

Three-tiered architectures typically distinguish control, sequencing and planning in a hierarchical structure [2]. Similarly, the SAS is comprised of the *control layer*, the *sequencing layer* and the *planning layer*; but outside each layer lies the *scientific knowledge base*, which contains the system's scientific capability. Figure 3 shows the SAS architecture.

The modules of the SAS primarily use client / server interprocess communication. A client / server model is appropriate because science-driven autonomous functions require a sequential passing of information to create exploration plans. Interprocess communication has been implemented on Nomad using Network Data Delivery Service (NDDS) [3]. Each layer of the architecture makes requests to the layer directly below it and responds to requests from the layer above. Figure 4 describes the flow of information from sensor hardware up to the planning layer. In contrast Figure 5 indicates how commands created with this information are passed back down, eventually affecting the real world through the robot's actuators.

All interfacing with the mobility platform is accomplished through the autonomous navigation system, a self-encapsulated architecture described in [4].

2.2.1. The Control Layer

The control layer is the lowest layer of the SAS, containing the robot's sensors and actuators. It allows the SAS to interact with the world. Control loops between software, sensors and actuators create *primitive behaviors*. In the SAS, primitive behaviors include:

- Sensor calibration
- Sensor deployment
- Acquisition of sensor readings
- Sensor diagnostics
- Stowing the sensor
- Returning sensor cost and workspace parameters

The *sensor driver* interface defines available primitive behaviors and abstracts sensor hardware specifics from the

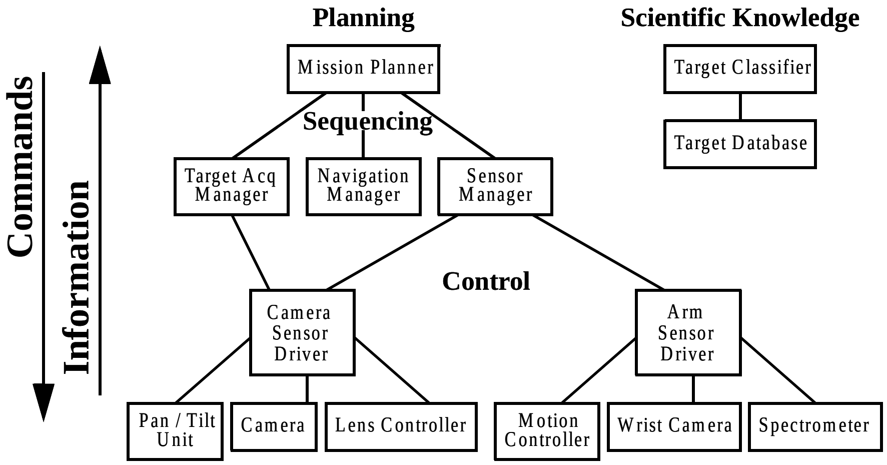

Figure 3. Science autonomy system as a modified three-tiered architecture

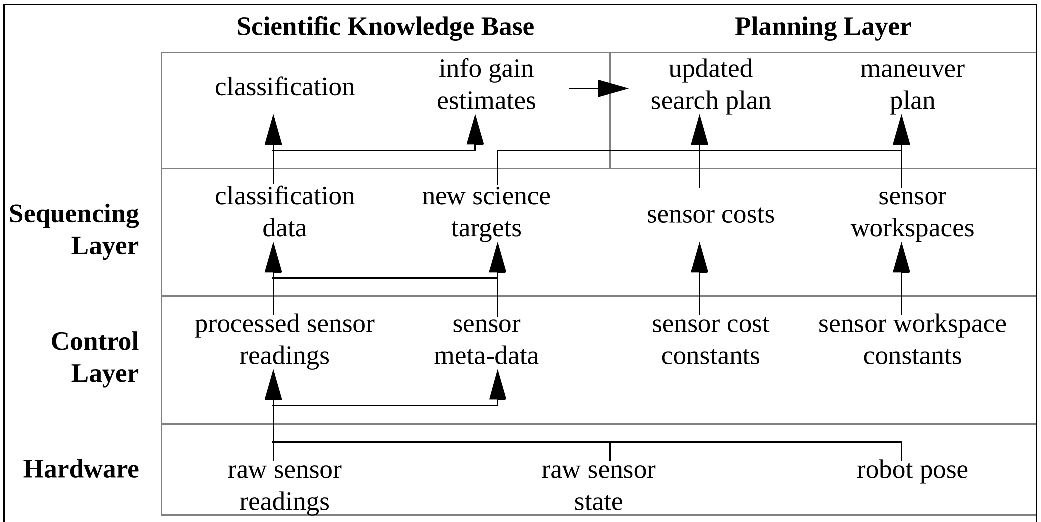

Figure 4. Flow of information in the SAS

Planning Layer	start / stop target acquisition	take sensor reading	start / resume search pattern	execute pre-planned maneuver	
Sequencing Layer	calibrate sensor	acquire data	deploy sensor	calculate sensor costs and workspace	command navigation system
Control Layer	read frame grabbers	command arm motors	read A/D converters	use spectrometer lamps	

Figure 5. Flow of commands in the SAS

sequencing layer. The primitive behaviors return descriptive status values to the sequencing layer so appropriate actions can be taken to handle error situations, such as calibrating an uncalibrated sensor [5]. They are also responsible for resolving device usage conflicts if multiple modules request simultaneous action. Finally, sensor drivers maintain constant values describing sensor deployment costs and workspaces.

The control layer waits for the completion of hardware operations as the primitive behaviors are executed. Whether these operations are blocking is dependent on device and operating system specifics. The raw data from hardware devices is converted into formats that the sequencing layer can understand. Additional sensor meta-data are created such as rock size estimates. The sequencing layer awaits completion of the control layer's behaviors before it can finish executing a command sequence. Requests to the control layer are blocking to simplify implementation.

2.2.2. The Sequencing Layer

Raw data sensed by the control layer are utilized for two operational modes: *acquisition* and *identification*. In acquisition mode, the SAS searches for new science targets in the world. In identification mode, study and classification of a target takes place. Multiple sensors may be coordinated to carry out both types objectives. The *sequencing layer* realizes this type of coordination by creating sequences of sensor or mobility commands. For instance, sensor calibration, deployment and data acquisition may be performed in response to a single "deploy sensor" command from the planning layer.

This layer is comprised of three modules: the *target acquisition manager*, the *sensor manager* and the *navigation manager*. The target acquisition manager uses *target acquisition drivers* to discover new targets while abstracting sensor-specific data processing methods from the rest of the system. Similarly, the sensor manager uses *sensor manager drivers* to successfully collect data on a given target while encapsulating knowledge of sensor

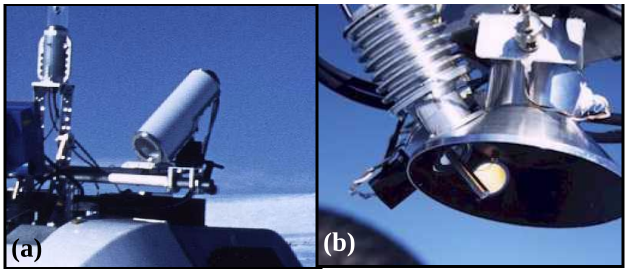

Figure 6. (a) Nomad's high-resolution camera, mounted on a sensor mast, is used both to acquire new targets and investigate them. (b) The spectrometer mounted on Nomad's manipulator arm (the small probe in the picture near the lamp) is critical to differentiate rocks from meteorites.

specifics. The navigation manager is responsible for executing search patterns and performing maneuvers by commanding the robot's autonomous navigation system.

New targets found, sensor data processed and calculated sensor deployment costs are sent to the planning layer. Although the sequencing layer should calculate real sensor deployment costs based on the current robot pose and target position, the present implementation on Nomad simply passes the constant values maintained by the control layer (see Section 4 for more discussion).

2.2.3. The Planning Layer

The highest level, the *planning layer*, considers the results of the sequencing layer such as newly acquired targets, sensor deployment costs and processed sensor data, and creates a plan that will optimize mission variables such as energy costs and scientific information gains [6]. Commands are sent to the sequencing layer to deploy sensors and construct navigation plans. The planning layer is therefore responsible for merging navigation and science within the SAS. The *mission planner* module comprises the planning layer.

Unlike typical three-tiered architectures, the planning layer of the SAS interfaces with both the sequencing layer and another layer, the *scientific knowledge base*. Many interactions are blocking, similar to those between the sequencing and control layers. Other non-blocking interactions exist that enable re-planning while the sequencing layer carries out commands. For instance, the mission planner does not block until the navigation manager finishes its pattern. Furthermore, the mission planner must listen for published notification messages from the target database that announce the existence of new targets, sensor data or classifier results.

2.2.4. The Scientific Knowledge Base

Outside each layer lies the *scientific knowledge base*, composed of the target database and target classifier. New targets and their sensor data are input into the scientific knowledge base, which stores them into its database. When new sensor data appear for a target, the target's classification is updated in the database along with new information gain estimates. Whenever the database state is altered, it publishes a notification message that reflects the changes. Many other modules in the SAS listen to this message, providing system-wide data synchronization without polling. The planning layer uses these messages to adjust its plans to deploy sensors and perform patterned searches.

2.3. Technologies for Autonomous Meteorite Search

2.3.1. The Nomad Robot

Autonomous search and in situ classification was made possible through the use of the Nomad robot, an autonomous planetary-rover prototype with specialized mechatronic and cognitive systems appropriate for this class of polar exploration missions (Fig. 1). Nomad's unique combination of in-wheel propulsion, deployable chassis and four-wheel rocker bogie suspension are major contributors to the robot's superior terrainability and robust autonomous navigation [7]. Nomad's navigational autonomy utilizes laser range finding and robot pose measurements to detect hazards and assess the quality of its state. The execution of autonomous science functions is carried out by a high-resolution camera mounted on the rover's sensor mast (Fig. 6a), and a manipulator arm that carries a reflection spectrometer (Fig. 6b).

2.3.2. Sensors

The SAS on Nomad contains two sensor drivers: a high-resolution camera sensor driver and a manipulator arm sensor driver. The camera sensor driver controls a 3-CCD color camera, lens and pan / tilt unit. The driver's deployment method converts estimated differential GPS (DGPS) coordinates of a target to pan and tilt angles. The data acquisition method stores the image from the CCD along with the pixel coordinates of any rocks in the image. The hardware has no real need to perform calibration.

Nomad's manipulator arm sensor driver contains multiple sensors and actuators, all of which are encompassed by the arm sensor driver. Deployment involves coordination between three axes of a motion control board and a color CCD camera mounted on the wrist of the arm to visually servo the wrist down to the potential meteorite target [8]. Again, estimated DGPS target coordinates are passed into

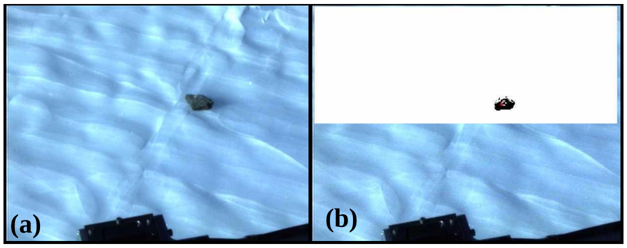

Figure 7. (a) Raw and (b) segmented images processed by the high-resolution camera acquisition driver.

the deployment method and errors in these estimates are overcome by visual servoing. Data acquisition involves storing spectrometer data and images from the wrist-mounted camera. Calibration of the spectrometer is performed after every fourth spectrum taken. This involves placing the wrist so the spectrometer is directly above a calibration target.

2.3.3. Target Acquisition

The high-resolution camera was ths basis for Nomad's target acquisition driver. Images from this camera are processed to segment rocks from background ice (Fig. 7). A linear combination of blue and green color ratios is calculated for each pixel in an image window. Pixels with a low blue-green ratio are designated as rock. Shadows in parts of an image generally create areas of noticeably different blue-green color ratios. A high standard deviation of these ratios therefore initiates an intensity-based shadow compensation routine. Coefficients used in this segmentation method were experimentally determined using images obtained from the project's 1998 expedition to Patriot Hills, Antarctica. Additional validation came from testing near McMurdo Station and Elephant Moraine before Nomad's meteorite searches officially began. Rocks as small as 1 cm in diameter can be detected by Nomad's high-resolution camera using this approach. This requirement is important to detect meteorites of this size, which are commonly encountered.

Using the pose of the robot and an assumption that the ground is a flat plane near the robot, the DGPS coordinates of each target are estimated and placed in the target database. By representing target locations in world coordinates, the SAS need not track an object to later identify it. Instead, the planning layer can consider other actions and possibly revisit the target at a later time, perhaps when rover resources are not as strictly constrained. This flexibility comes at a cost however, since the terrain near the robot always deviates slightly from a flat ground plane. Therefore when the planning layer decides to deploy a sensor, the sensor manager drivers must be robust to uncertainty in target position.

2.3.4. Sensor Manager Drivers

Similarly to its sensor drivers, Nomad uses two sensor manager drivers: one for its high-resolution camera and one for its manipulator arm. The camera sensor manager driver only communicates directly with the sensor driver of the same name. It is a fairly simple object; sensor manager commands are basically passed directly to the camera sensor driver.

The manipulator arm sensor manager driver is more complex. It communicates with both the camera and the arm sensor drivers. Before the arm is deployed, the camera takes a new image of the target to provide an improved location estimate. This method compensates for inaccuracies in the robot's pose that introduce errors in the initial transformation of target location to world coordinates. The arm then visually servos to and gathers spectral information about the target.

2.3.5. Navigation Manager

The navigation manager converts high-level mobility plans from the planning layer to steering arcs passed to the autonomous navigation system every second. A separate obstacle detection module simultaneously sends desired steering commands to the navigation system, which arbitrates between the two inputs and sends a command to the robot designating a steering direction and a speed, which is currently set at 15 cm/s. The navigation manager follows a search path chosen by the planning layer using the pure pursuit path tracking algorithm [9]. When executing patterns, it dynamically updates the robot's next waypoint. This lookahead distance enables the robot to quickly return to the path after avoiding obstacles or examining targets without creating oscillations around the path.

The navigation manager also enacts pre-planned maneuvers provided by the planning layer, sending steering commands one-by-one to the navigation system. Such maneuvers may be necessary to put desired targets into the workspace of the robot's sensors. In this mode, any obstacle avoidance module input is ignored, as the maneuvers are generally performed in limited locations that are already known to be clear of obstacles. This prevents the obstacle avoidance module from interfering with the pre-planned sequence of maneuvering steps.

2.3.6. Mission Planner

The mission planner initiates a search pattern by notifying the navigation manager and the target acquisition manager. New targets found by the target acquisition manager are added to a list. Considering each target in this list in combination with each available sensor, the mission planner requests an estimate of the information that could be gained by additional sensor data. If the estimated information gain for a particular target / sensor pair is below a threshold, then that pair is ignored and no further cost calculations are performed.

If at least one target / sensor pair passes this first test, sensor deployment costs are calculated. The distance to the target from the robot, the time cost of using the sensor and the need for maneuvering the robot into the sensor's workspace are all considered. These costs are compared for each target / sensor pair, and the lowest cost target is selected for investigation.

Upon a decision to investigate, the mission planner may request a maneuver to move the robot's sensors into range of the target. Maneuver planning is performed using the A* algorithm [10] and a model of how the robot responds to a discrete set of steering commands to speed the planning process. The model also contains state information about the previously commanded move. Although A* ensures an optimal plan, the mission planner augments it with heuristics to decrease average execution time. It checks if the robot will need to back up first if another approach could more accurately place the robot. This maneuver plan is then passed to the navigation manager to execute.

When complete, the mission planner activates the sensor manager, requesting the desired data on the chosen target. New sensor data are placed in the target database and then classified. The mission planner re-analyzes its target list to determine if additional target / sensor pairs should be selected at the current time.

2.3.7. Target Classifier

Once the robot has acquired a new target sample, maneuvered into proper sensing position and gathered sensor data, the target classifier is invoked. The target classifier is responsible for deciding, based on sensor data, the likelihood that a sample is either a meteorite or terrestrial rock. To do so, it uses a Bayes network to classify science targets as belonging to predefined classes such as sedimentary, metamorphic, igneous, extraterrestrial or "other," generally meaning ice or snow. Based on its current assumed knowledge of a target, it also calculates the information gain that would result from readings from each remaining sensor in the system.

Nomad has multiple sensors which are not deployed all at once. The classifier must therefore accept incomplete data and compound evidence as more sensor data become available. Moreover, the classifier should accept prior evidence from other sources, including expert knowledge on what to expect in a particular location. Rock classes are often ambiguous, and the distinctions between certain types fuzzy at best [11]. The classifier must handle this ambiguity and indicate several likely hypotheses if a definite classification cannot be achieved.

Nomad performs classification using imagery and spectral data. A Bayes network, which encodes the statistical distribution of image and spectral features for each rock type along with their assumed prior probabilities, computes the posterior probability of the rock type being examined, given the current sensor data [12]. Image features used include color and size. A fixed set of spectral features is matched against Gaussian templates defined throughout the range of spectra wavelengths. A detailed discussion of the issues and implementation of the target classifier can be found in [13].

The classifier works asynchronously whenever new sensor data enter the database. Once receiving notice of this event, it classifies the science target based on both the new data received and data previously recorded in the database. Therefore each new sensor reading enhances previous classifications rather than replacing them.

3. Field Demonstration Results

During January 2000, Nomad's autonomous exploration and in situ science capabilities were put to test in the extreme environment of Elephant Moraine, Antarctica (76 deg 16' S, 157 deg 12' E). After a short period of subsystem tests, such as target acquisition, arm servoing and classification of planted samples, Nomad was set on its own to pursue the discovery of new meteorites, the first of which was found January 22, 2000 (Fig. 8). Through the course of ten demonstrations that featured autonomous patterned searches and persistent examination of targets in the robot's course, Nomad found and correctly classified three meteorites and more than forty terrestrial rocks. An additional two meteorites were correctly classified during tests performed without patterned searches. Sensor deployment was performed with a high degree of autonomy; the SAS autonomously acquired new targets with a 79% success rate and deployed Nomad's manipulator arm with a 72% success rate. Image segmentation required for target acquisition and manipulator visual servoing proved capable in many conditions, although its parameters had to be hand-tuned for different lighting conditions that ranged from bright direct sunlight to diffuse overcast conditions. Not surprisingly, autonomously deployed sensor data quality did not match that of human gathered training set data, but it allowed effective discrimination of meteorites from rocks. However, classification was systematically poor for hydro-thermally altered dolerite and basalt rocks upon which the robot had not been trained but were common at

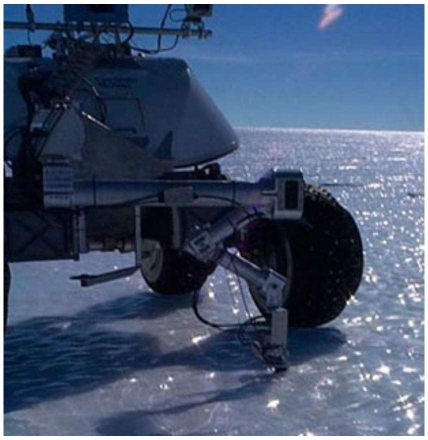

Figure 8. Nomad studying its first autonomously found meteorite on January 22, 2000.

Elephant Moraine. During its autonomous searches Nomad covered 2500 m^2 of blue ice and snow, which translates to about 1.25 km of linear distance. Nomad's discoveries were made in 16 hours of productive searches out of 10 full days of field operations. These two metrics define the standard for autonomous search for meteorites.

Demonstrations were performed in two terrain types. Eight demonstrations took place in open ice fields and involved sample densities of about one sample every ten square meters. Although sample densities were low, many samples found were meteorites. Nomad spent 50.7% of the time driving and performing target acquisition in this type of area. 18 out of 23 arm visual servoing attempts were successful; this 78% success rate shows that the study of individual rock samples in this environment is practical. Here high robot velocity is key to effective search with the SAS.

Two demonstrations took place near the moraine proper. Here Nomad saw sample densities of one to two samples per square meter. However, far fewer samples were meteorites. Therefore Nomad spent less time driving and more time deploying its sensors. In fact, 48.7% of Nomad’s time was spent deploying its manipulator arm. However, only 69% of the 36 deployment attempts were successful. The additional failures generally occurred when multiple targets were found in close proximity. Errors in the targets’ initial position estimates would cause uncertainty in target select during visual servoing, resulting in a failed arm deployment attempt. Therefore, modifications such as more discriminating target acquisition drivers would increase the speed and robustness of the searches in moraine environments.

Complete expedition results and discussion of outcomes can be found in [14].

4. Critique and Future Work

The modularity of the SAS resulted in robustness and manual adaptability to unforeseen problems. Individual software modules experiencing problems could be individually restarted without disabling the entire system, implying that fault tolerance could be implemented in the future. Alterations and bug fixes made to modules during the mission did not require substantial changes to the rest of the SAS.

In its present state, the SAS has no means to calculate the quality of sensor data. It has no measure of confidence that the data being classified are valid. This is especially critical for spectrometer data that are very sensitive, not only to sensor head placement but to random natural features such as rock face angle. While these values are difficult to sense, a confidence metric should be placed into the system to be made available to the classifier. A low confidence would suggest to the mission planner that additional sensor readings are beneficial. New data quality metrics created by the sequencing layer would enable this improvement.

Further work could also be done in the realm of target acquisition. Algorithms developed by the Onboard Science Understanding Project at NASA Ames Research Center could be used to autonomously detect interesting geologic features and individual rocks using texture segmentation [14]. Furthermore, there is only one target acquisition driver currently used and there is no method in place to fuse their data together. For instance, if an image showed a rock on the ice and a metal detector being swept in front of the robot found a signal, the system could only recognize two new targets even if they were really the same object seen by different sensors. A good way to address this problem could be through the use of evidence grids [15]. Not only would this allow fusion of data from different sensors, but also multiple readings from a single sensor could be combined to give a more accurate target location.

Similarly, several times during its demonstrations, Nomad re-examined or ran over rocks it had already seen. While this was mostly due to a very small search row width, the robot often did have knowledge of the rock’s location as it ran it over. These rocks could be designated as obstacles in the mission planner’s global map and therefore not repeatedly studied or even damaged.

Finally, the calculation of sensor deployment costs and information gains still must be investigated to achieve the efficiency gains that sensor selection may provide. The deployment cost of a sensor is a combination of energy and time costs. However, other cost metrics could be used such as data storage requirements. Costs are currently defined as constants for each sensor in the system; cost has no dependence on distance to the target, as it should. In the case of some sensors, such as those on Nomad's manipulator, a cost could not be reliably calculated prior to deployment because an unknown number of visual servoing steps may be taken. Therefore statistical techniques or fuzzy logic may be useful for cost estimation and calculation, in both the sensors and the mission planner.

5. Conclusions

Nomad's unprecedented discovery and in situ classification of Antarctic meteorites is primarily attributed to the effectiveness of the SAS. Built on a paradigm of hierarchical control driven by science goals and intelligent apportion of sensor management, sequencing, and mission oversight, SAS is a prototypical architecture for autonomous science robots.

Although human scientists will always be preeminent, we envision that robots with advanced SAS architectures will transform exploration through the ability to search, classify and make discoveries, especially in the context of missions that prohibit frequent human oversight. Examples include missions to the far side of planets, extremely remote polar regions and hydrothermal springs.

The SAS implementation on Nomad has yielded useful technical lessons. It is evident that autonomous search strategies should take into account terrain and target distribution information to dynamically update the pace of search. Moreover, SAS must incorporate intelligence to evaluate sensor data quality without any human input. This observation implies the need for automatic assessment of sensor placement quality. Finally, the incorporation of metrics such as information gain should have profound implications on the effectiveness of SAS architectures. Our current work focuses on the implementation of SAS on life seeking robots.

6. Acknowledgements

This program has been supported by the National Aeronautics and Space Administration (NASA) under grants NAG5-7707 and NAG9-1090. The authors would like to thank Liam Pedersen, Stewart Moorehead, William Cassidy, James Teza and the rest of the Robotic Antarctic Meteorite Search Project team for their crucial contributions to this effort.

7. References

[1] Apostolopoulos, D. S., Wagner, M. D., Shamah, B. N., Pedersen, L., Shillcutt, K. and Whittaker, W. L., "Technology and Field Demonstration of Robotic Search for Antarctic Meteorites", International Journal of Robotics Research, Vol. 19, Number 11, pp. 1015-1032, November 2000.

[2] Gat, E., "On Three-Layer Architectures", AI and Mobile Robots, D. Kortenkamp, P. Bonasso and R. Murphy eds., MIT/AAAI Press, Cambridge, MA, 1998.

[3] RTI web page, http://www.rti.com.

[4] Moorehead, S., Simmons, R., Apostolopoulos, D. and Whittaker, W. L., "Autonomous Navigation Field Results of a Planetary Analog Robot in Antarctica", Proceedings of International Symposium on Artificial Intelligence, Robotics and Automation in Space, pp. 237 - 242, Noordwijk, The Netherlands, June 1999.

[5] Noreils, F., "Integrating Error Recovery in a Mobile Robot Control System", IEEE International Conference on Robotics and Automation, pp. 396 - 401, Cincinnati, OH, May 1990.

[6] Shillcutt, K. and Whittaker, W. L., "Modular Optimization for Robotic Explorers", AAAI Fall Symposium on Integrated Planning for Autonomous Agent Architectures, Orlando, FL, 1998.

[7] Shamah, B., Apostolopoulos, D., Rollins, E. and Whittaker, W. L., "Field validation of Nomad's robotic locomotion", Proceedings of the SPIE International Conference on Mobile Robots and Intelligent Transportation Systems, pp. 214 - 222, Boston, MA, 1998.

[8] Wagner, M. "Experimenter's Notebook: Robotic Search for Antarctic Meteorites 2000 Expedition", Technical Report CMU-RI-TR-00-13, Robotics Institute, Carnegie Mellon University, June 2000.

[9] Coulter, R. C., "Implementation of the Pure Pursuit Path Tracking Algorithm", Technical Report CMU-RI-TR-92-01, Robotics Institute, Carnegie Mellon University, January 1992.

[10] Hart, P., Nilsson, N. and Raphael, B., "A Formal Basis for the Heuristic Determination of Minimum Cost Paths," IEEE Transactions on Systems Science and Cybernetics, SSC-4(2), pp.100 - 107, 1968.

[11] Dietrich, R. and Skinner, B., Rocks and Minerals, New York, J. Wiley & Sons, 1979.

[12] Pedersen, L., Apostolopoulos, D. and Whittaker, W. L., "Bayes Networks on Ice: Robotic Search for Antarctic Meteorites", Proceedings of Neural Information Processing Symposium, Denver, Colorado, November 27 - December 2 2000.

[13] Pedersen, L., "Autonomous Robotic Characterization of Unknown Environments", accepted to the International Conference on Robotics and Automation, Seoul, Korea, May 2001.

[14] Apostolopoulos, D., Pedersen, L., Shamah, B., Wagner, M. D., Whittaker, W., "Robotic Antarctic Meteorite Search: Outcomes", accepted to the International Conference on Robotics and Automation, Seoul, Korea, May 2001.

[15] Gulick, V. C., Morris, R. L., Ruzon, M. A. and Roush, T. L., "Autonomous Image Analyses During the 1999 Marsokhod Rover Field Test", JGR-Planets, in press.

[16] Moravec, H., "Certainty Grids for Sensor Fusion in Mobile Robots," Sensor Devices and Systems for Robotics, Springer-Verlag, Berlin, 1989, pp. 243 - 276.