2008 IEEE International Conference on
Robotics and Automation
Pasadena, CA, USA, May 19-23, 2008

Spatial Coverage Planning for a Planetary Rover

Daniel M. Gaines   Tara Estlin   Caroline Chouinard

Abstract— **We are developing onboard planning and execution technologies to support the exploration and characterization of geological features by autonomous rovers. In order to generate high quality mission plans, an autonomous rover must reason about the relative importance of the observations it can perform. In this paper we look at the scientific criteria of selecting observations that improve the quality of the area covered by samples. Our approach makes use of a priori information, if available, and allows scientists to mark sub-regions of the area with relative priorities for exploration. We use an efficient algorithm for prioritizing observations based on spatial coverage that allows the system to update observation rankings as new information is gained during execution.**

I. Introduction

Our goal is to increase the onboard decision-making capabilities of planetary exploration rovers. Currently, each morning of the Mars Exploration Rover (MER) mission the scientists and engineers meet to discuss the observations they would like the rover to perform. A subset of these observations are selected that are predicted to fit within the time and resource (e.g. energy, onboard memory) constraints of the rover. The engineering team spends the rest of the day preparing the specific sequences that the rover will perform to collect these observations and modeling the plan to ensure it fits within resource constraints.

While the MER mission has been highly successful at exploring Mars, mission operations are manually intensive and time consuming. And, in some cases, the sequences that are uplinked do not always take full advantage of available opportunities. For example, if the rover receives more solar array input than expected, it may have energy to preform more science observations than what was uplinked.

By enabling rovers to perform onboard planning and scheduling, we anticipate greatly reducing the time and effort required to perform mission operations while increasing the science that is acquired. The science and engineering teams will be able to uplink observation requests that potentially over-subscribe the rover's resources. The rover will use observation priorities and its current assessment of available resources to make decision about which observations to perform and when to perform them.

Jet Propulsion Laboratory, California Institute of Technology, 4800 Oak Grove Drive, Pasadena, California 91109, {*firstname.lastname*}@jpl.nasa.gov

In order to make effective decisions about which observations to perform, the rover must reason about science priorities. In our current work, we are focusing on situations in which the rover is exploring large geological features such as craters, channels or a boundary between two different regions. In these cases, an important factor in assessing the quality of a plan is how the set of chosen observations spatially cover the area of interest. Thus, one of the considerations a rover should make when evaluating which observations should be included in a plan is how well the candidate observations will increase the spatial coverage of the plan.

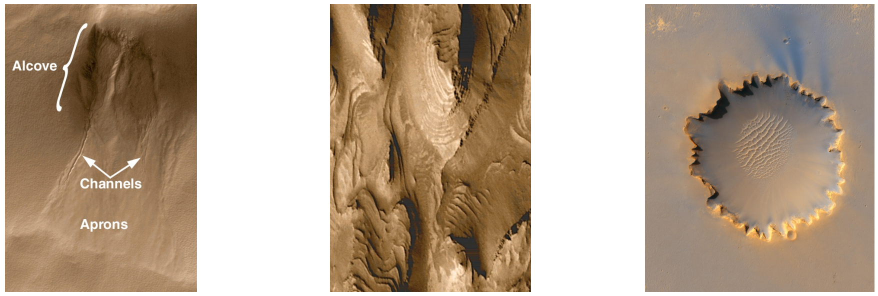
The overall goal of this technology is to enable the rover to generate and execute plans that makes an appropriate balance between detailed study and broad coverage of a region. In this paper we describe a technique that allows a rover to evaluate the spatial coverage quality of a plan. It is also important that the rover consider the cost of acquiring observations and ensure that the plan respect mission and resource constraints. For example, some observations may be more time intensive while others are more power or memory intensive. The rover should also consider ordering observations to reduce traverse distance and time. Therefore, we have integrated the spatial coverage metric into a planning system that reasons about observations costs along with time and resource constraints. This enables the rover to generate high-quality, efficient plans that take into account spatial coverage quality while respecting mission and resource constraints.

II. Exploring Geological Features

We are developing onboard planning and scheduling technology to enable rovers to more effectively assist scientists in exploring geological features. Figure 1 shows examples of geological features on Mars illustrating the types of features rovers may be directed to explore.

A scientific campaign for exploring a geological feature will employ a variety of rover instruments for collecting data about the region. For example, each Mars Exploration Rover is equipped with remote sensing instruments including high-resolution panoramic stereo cameras with a variety of filters (Pancam), navigational (Navcam) and hazard avoidance (Hazcam) stereo cameras and a Mini Thermal Emissions Spectrometer (Mini-TES). Each rover also has an arm with a suite of instruments for close contact measurements: a microscopic imager (MI), two spectrometers and a rock abrasion tool (RAT) able to remove a few millimeters of a rock's surface.

978-1-4244-1647-9/08/$25.00 ©2008 IEEE.



(a) Channels (MGS MOC Image) (b) Layers (MGS MOC Image) (c) Craters (MRO HiRISE Image)

Fig. 1. Example geological features on Mars.

When humans perform mission planning, there may be a variety of reasons why a particular observation is selected in a given plan such as its benefit to different science considerations (geology, atmospheric studies, . . .). In this paper, we are focusing on assessing the contribution observations make to the spatial coverage of a plan.

Of course, the mission planning team must also take into account the limited set of resources that the rovers have to perform observations. The rovers are constrained by limited energy, onboard data storage, downlink opportunities and bandwidth and time to complete observations. Each observation places a different set of demands on these resources. Some are very time consuming, such as long-term spectrometer integrations, while others are memory intensive, such as Pancam acquisitions. And some activities are constrained to occur at certain periods of the day due to sun angle or temperature.

With respect to spatial coverage, some observations have a wide field of view, such as Navcams, while others, such as Mini-TES and Pancam have a narrow field of view. The instruments also vary in the quality of their coverage with respect to the distance of an observation target from the rover. Terrain features may obstruct the areas covered by an observation. Finally, for a given geological feature, scientists may be more interested in certain sub-regions of that feature than in others. Thus, observations should also be evaluated based on the relative importance of the area for which they provide coverage.

III. CASPER Continuous Planning and Optimization Framework

Our objective is to enable onboard planning software to reason about the scientific quality of a plan so that it can make more informed decisions about which observations to perform. This will enable the ground team to uplink a larger set of observations and let the rover dynamically select among them based on the scientific and engineering merit of the resulting plan and the rover's assessment of available resources. During execution, the rover will modify the plan based on the current estimate of its resources.

Our approach is implemented within the CASPER system [1], [2]. CASPER employs a continuous planning technique where the planner continually evaluates the current plan and modifies it when necessary based on new state and resource information. Rather than consider planning a batch process, where planning is performed once for a certain time period and set of goals, the planner has a current goal set, a current rover state, and state projections into the future for that plan. At any time an incremental update to the goals or current state may update the current plan. This update may be an unexpected event (such as a new science target) or a current reading for a particular resource level (such as battery charge). The planner is then responsible for maintaining a plan consistent with the most current information.

A plan consists of a set of grounded (i.e., time-tagged) activities that represent different rover actions and behaviors. Rover state in CASPER is modeled by a set of plan timelines, which contain information on states, such as rover position, and resources, such as energy. Timelines are calculated by reasoning about activity effects and represent the past, current and expected state of the rover over time. As time progresses, the actual state of the rover drifts from the state expected by the timelines, reflecting changes in the world. If an update results in a problem, such as an activity consuming more memory than expected and thereby over-subscribing RAM, CASPER re-plans, using iterative repair [3], to address conflicts.

CASPER includes an optimization framework for reasoning about soft constraints such as reducing the distance traversed by the rover and increasing the value of science data collected. User-defined preferences are used to compute plan quality based on how well the plan satisfies these constraints. Optimization proceeds similar to iterative repair. For each preference, an optimization heuristic generates modifications that could potentially improve the plan score.

Figure 2 provides a high level description of the control algorithm used for the rover application of CASPER. The algorithm takes as input a set of goals with associated science priorities and a set of time and resource constraints. CASPER's optimization framework supports a wide-range of user-defined preferences. For the purpose of this paper, we focus only on a spatial coverage preference which will be explained in the next section.. The main loop of the algorithm interleaves iterative repair and iterative optimization to search for a conflict-free plan of high quality. The loop begins by processing any updates on state and resource timelines or on activity status. It then enters a loop in which it attempts to improve the plan by repairing conflicts or performing optimization steps.

```
Input
    A set of science observations (oversubscribed)
    Time & resource constraints
Repeat:
    Process updates from Executive
    Optimize for n iterations
        If no conflicts, satisfy observation
            with largest spatial coverage contri-
            bution
        Else, resolve conflicts
            if delete, delete observation with
            smallest spatial coverage contribu-
            tion
    Commit and/or Rescind activities
    If idle, attempt to move up future activities
```

Fig. 2. CASPER control algorithm for rover domain.

If there are no conflicts, CASPER attempts to improve the plan by satisfying an observation, in this case, the observation with the largest spatial coverage contribution. If there are conflicts, it will perform an iteration of repair, selecting one of the available repair methods (e.g. move an activity, add an activity, ...). If deletion of an observation is selected, it will select the observation providing the smallest spatial coverage contribution.

Note that satisfying an observation will likely introduce conflicts as this is where CASPER will evaluate the resource and temporal requirements of an observation. CASPER will use subsequent iterations to try to resolve these conflicts. For example, if the rover is not currently at the appropriate location to take an observation, this will introduce a state conflict which CASPER will attempt to resolve. One option for fixing this conflict is to add an activity that can move the rover from one location to another, i.e. a traverse activity. This is also where CASPER selects an ordering of observations in an attempt to minimize traverse distance. We use a simple traveling salesman heuristic to pick start times for activities to reduce traverse distance.

Figure 3 illustrates the "lifetime" of observations in the system. New observations are placed in a requested bin. When an observation is selected to be satisfied, it moves from requested to pending in which it awaits execution. In the meantime, it may be deleted to resolve conflicts in the plan, in which case it moves back to requested. As it nears time for a pending observation to be executed, it is committed and sent to an executive process for execution. If a problem occurs in the plan before the actual execution time of the activity, the planner has the ability to request a rescind of the observation from the executive. If the executive is able to honor the rescind request, it is as if the observation had been deleted from the plan and it returns to the requested bin.

The next section provides details on how the spatial coverage quality of a plan is computed and how observations are selected to improve this score.

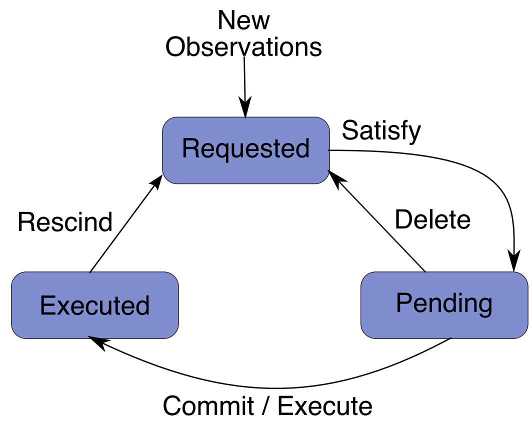


Fig. 3. The lifetime of an observation.

IV. Spatial Coverage Preference

Figure 4 provides an example region of terrain that we want a rover to explore and along with an example set of observations that are under consideration for the plan.

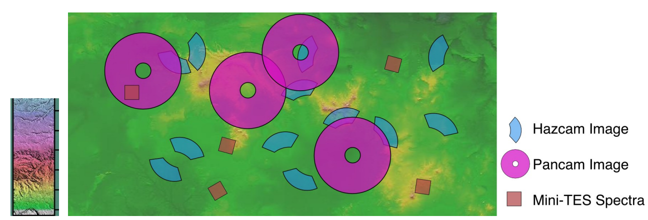


Fig. 4. Digital elevation map of an example terrain to be explored along with a set of observations to perform.

With limited available resources, it is unlikely that the rover will be able to perform all of these observations. As discussed previously, there are many considerations for determining which subset of observations should be included in a plan. The objective of this work is to develop a preference to encourage spatial coverage to be one of the considerations during plan generation and modification.

In this section we describe our approach to representing and reasoning about the spatial coverage quality of a plan. We begin by describing how we represent a priori information about the terrain to be explored along with scientists' priorities indicating the relative importance of various sub-regions. We then describe how we model the coverage quality afforded by a given observation. These observation models are used to track the spatial coverage quality of the plan, taking into account those observations that have already been executed and those that are scheduled to execute in the future. When resources and plan space is available, all of this information is then used to select which observations to add to the plan in an attempt to optimize the spatial coverage of the plan. Conversely, when resources are over-subscribed and observations must be shed, this information is used to select an observation that will make the smallest impact on the spatial coverage of the plan if it were removed.

A. Terrain and Terrain Priority Representation

Knowledge of terrain will enable the system to make better predictions about the coverage of observations as it will know about occlusions from terrain features such as rocks or hills. Scientists typically have a variety of a priori information that is used to identify candidate observations that can contribute to the initial terrain map. Images from previous observations, such as Navcam and Pancam observations, are the primary source of information for selecting new targets. In addition, images from orbiting spacecraft as well as images taken during the spacecraft's descent, provide a coarse view of the geological features.

We represent a priori knowledge of the terrain to be explored as a digital elevation map where each pixel represents the height of the terrain at that point. Figure 4 shows an example terrain map. The resolution of the map has a direct impact on the space and time complexity of the algorithm. It is not critical that we compute a highly accurate score for the amount of terrain covered by a given observation. Rather, it is important that the relative scores of different observations be correctly assessed. Thus, we convert the input terrain map into a coarser resolution such as the one in Figure 5. The resolution of the terrain map is a parameter that can be tuned to make a trade-off between accuracy of coverage quality predictions and computational complexity of the system. We have considered using an octree representation of the terrain. This would improve the efficiency of computing observation visibility. However, as we will see, to compute the spatial coverage contribution of a given observation, we would still need to query individual cells of the terrain representation. Because in our algorithm, the latter computation is performed far more frequently than the visibility check, we decided to go with the simpler matrix representation of the terrain.

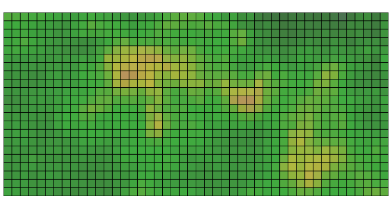


Fig. 5. Lower resolution version of terrain in Figure 4.

It is also important to note that the approach does not require that a priori knowledge be complete or accurate. Missing or incorrect data in the terrain map will result in incorrect estimates of the spatial coverage that an observation will afford which, in turn, could result in lower quality plans. However, as observations are performed the terrain map will be updated and the coverage quality of upcoming observations will be re-assessed.

The system also takes as input a matrix of weights that define the relative scientific importance of sub-regions of the terrain map. The matrix is the same size and dimensions as the input terrain matrix with each cell containing a value between 0 (least important) and 1 (most important). Figure 6 shows an example terrain weight matrix.

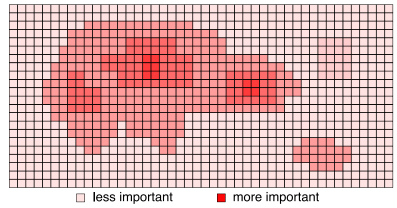
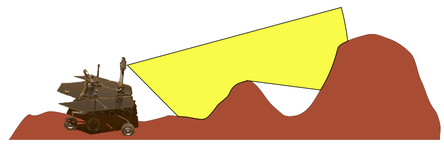


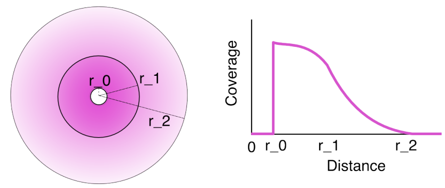
Fig. 6. Example of scientist priorities for terrain coverage.

B. Modeling Observations

In order to evaluate the coverage quality of a plan, it is necessary to compute the coverage afforded by a given observation. Figure 7 illustrates the key steps in this computation.



(a) compute observation visibility



(b) compute observation coverage quality

Fig. 7. Modeling observation coverage quality.

The first step is to determine which cells of the terrain are visible from the location of the observation. For each cell within range of the observation (as determined by the range value for the instrument) we perform an intersection test between the terrain and a line from the location of the instrument to the cell using the intersection test [4].

Once it has been determined that a cell is visible to an observation, we compute the coverage quality score which represents how well that cell is covered by the observation. A score of 0 indicates that the cell is not covered at all by this observation (e.g. it is occluded by the rover body). A score of 1 represents "perfect" coverage, in the sense that another observation of the cell would not improve the cell's coverage.

Computation of the coverage quality score varies based on the type of instrument used. In general, cells further from the origin of the observation are not covered as well as those closer in. Figure 7 (b) illustrates the coverage quality model for a panoramic image observation. Let d be the distance of the visible cell from the origin of the observation. The radius r_0 represents the diameter of the rover body. If $d \leq r_0$ then the cell is occluded by the rover body and not covered by this observation. Cells that lie between r_0 and r_1 are within the primary range of this instrument. Coverage is high for cells close to r_0 but drops off moving out toward r_1.

The range between r_1 and r_2 is used to encourage "spreading out" of observations. The idea is that, all else being equal, one might prefer to have observations spread out across the terrain rather than clustered in a small region. The extend to which spreading out is encouraged can be tuned by increasing or decreasing r_2.

Finally, the coverage of the cell is multiplied by the weight in the scientific priority matrix resulting in the cell's spatial coverage quality provided by this observation.

Note that the above computation is performed once per observation per run. An *observation overlay matrix* is produced which represents this observation's individual contribution independent of other observations. Later, this overlay matrix will be queried to compute the observation's contribution in combination with other observations. The cost of computing the observation overlay depends on the size of the area covered by the observation and the resolution of the terrain matrix as this determines the number of cells that must be checked for coverage. If an observation has a diameter of n cells, then there are n^2 cells to check and, in the worst case, the check to see if a cell is visible from the observation's location is $\mathcal{O}(n)$. Thus the cost for computing the overlay matrix is $\mathcal{O}(n^3)$. If there are o observations, then the cost of initialization is $\mathcal{O}(on^3)$.

C. Tracking Coverage Quality

Now that we have a way of computing the spatial coverage for a given observation, the next step is to keep track of the spatial coverage provided by a set of observations. We do this by recording the spatial coverage quality afforded by the observations into a coverage quality matrix. A coverage quality matrix is the same dimension and resolution as our terrain matrix with each cell containing a coverage quality value. If multiple observations cover the same cell in the coverage quality matrix we record the max coverage quality score afforded by these observations. Figure 8 shows an example coverage quality matrix reflecting the coverage quality for a set of observations.

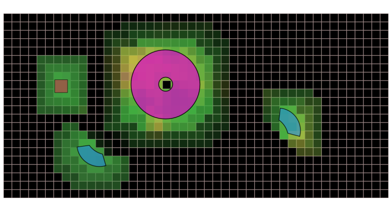


Fig. 8. Example coverage quality for a set of observations.

We maintain two separate coverage qualities matrices. An Executed Coverage Quality Matrix tracks the coverage quality afforded by the observations that have already executed. The Pending Coverage Quality Matrix tracks the coverage quality from the executed observations and the predicted coverage quality that will be obtained after the pending observations in the plan have been executed. Each coverage quality matrix has a score which is equal to the sum of the coverage quality of each of its cells.

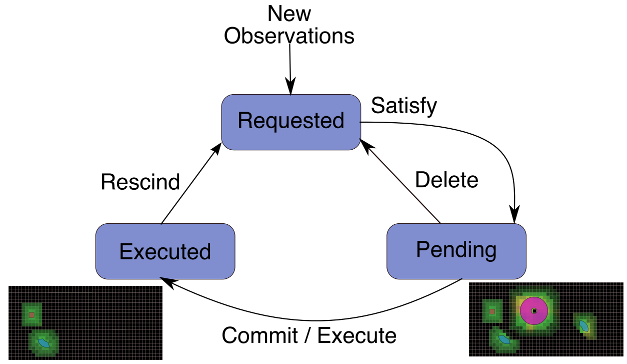


Fig. 9. Maintaining two coverage quality matrices.

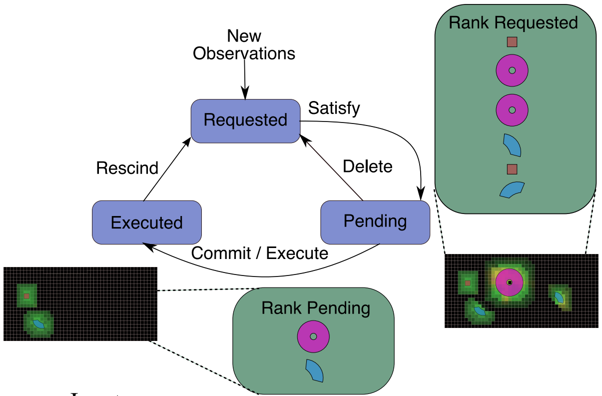
D. Ranking Observations

We rank observations with respect to how well they are expected to improve the coverage quality of the plan. We maintain two rankings, one for the requested observations, those that are not yet in the plan, and the pending observations, those that are in the plan but have not yet executed. When selecting a requested observation to add we select the highest ranked observation from the requested observations ranking. If we must shed a pending observation to resolve a conflict, we select the lowest observation from the pending observations rankings. In actuality, rather than always selecting the highest observation to add (or lowest when deleting) we perform a probabilistic selection from the ranked list of observations with a probability of selecting a particular observation proportional to the coverage quality it is expected to contribute to the plan. This helps the system to avoid getting stuck trying to satisfy an observation for which there are insufficient resources to perform.

Because the coverage afforded by observations may overlap, the coverage quality an observation will contribute to a plan depends in part on the other observation already in the plan. Thus, when we rank the requested observations we do so relative to the pending coverage quality matrix. Similarly, the coverage quality contribution of a pending observation depends on the observations that have already been executed. Therefore, the pending observations are ranked relative to the executed coverage quality matrix. Figure 10 shows the algorithm used to rank a set of observation relative to a coverage quality matrix. A contribution score is computed for each observation which is equal to how much the score of the coverage quality matrix would increase if this observation were added.

Note that the algorithm in Figure 10 ranks only the single next observation to add (or remove) and does not indicate which order the remaining observations should be added (or removed). Instead, we use an iterative approach since, given the iterative nature of CASPER's repair and optimization loop, we will add or remove activities one at a time.

The cost of computing a given observations contribution score is $\mathcal{O}(n^2)$ where n is the diameter of the observation's coverage in number of cells. If there are o observations, then the cost of ranking the observations is $\mathcal{O}(on^2)$. However,



Input
    Unranked Observations
    Initial Coverage Quality Matrix
For each observation in Unranked Observations
    Observation's contribution score = how much the score of the coverage quality matrix would improve if this observation were included
Sort observations based on contribution score

Fig. 10. Ranking a set of observations relative to a coverage quality matrix.

since an observation's contribution only changes if there are observations with which it overlaps, this algorithm can be sped up by only re-computing an observation's contribution if another overlapping observation is selected. A simple bounding box intersection check can be used to quickly check for overlapping observations. But still, the worst case cost of ranking observations will be $\mathcal{O}(on^2)$.

This iterative approach represents a greedy algorithm for selecting observations to add and remove and does not guarantee an optimal solution except in the case where the observations do not overlap. We have chosen not to attempt an optimal solution for three main reasons. First, observations are selected for a variety of reasons, not just spatial coverage. Thus, we cannot count on the spatial coverage ranking being honored when observations are added and removed. Second, because we will have to re-rank observations during execution (when observations are added and removed or when the terrain map is updated) we want a fast computation. Finally, we expect that observations will not overlap significantly and thus the greedy approach will not be far from optimal.

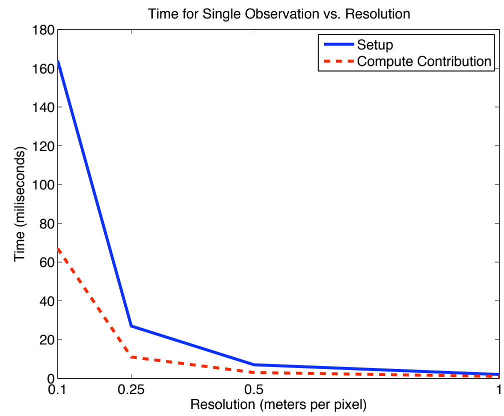
E. Updating Spatial Coverage During Execution

During the course of executing the plan, the system will need to update its rankings. While performing observations, we will be collecting new information about the terrain being explored. We can update the terrain map when this happens. Doing so will improve the accuracy of the coverage quality predictions. However, when the terrain is updated we will need to re-compute the coverage quality afforded by each of the observations and re-compute our rankings.

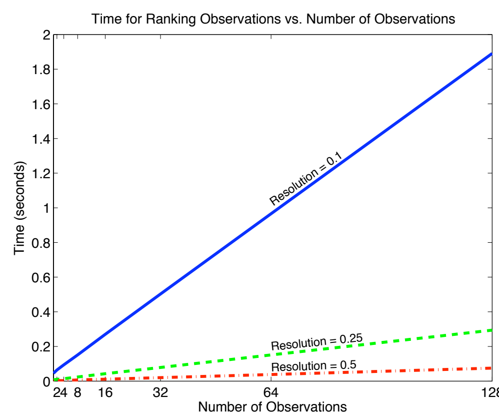
We must also re-compute rankings when observations are added to or removed from the plan since the contribution score of an observation depends on the order in which it is added to the plan.

V. Empirical Performance Results

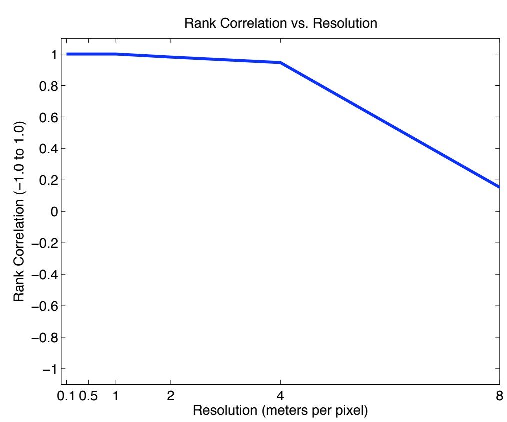
We performed a set of experiments to evaluate the performance of the spatial coverage preference algorithm. Figure 11 summarizes these results. The experiments were performed in simulation on a 2.6 GHz Pentium. This is of course a much faster processor than would be used for a rover mission in the foreseeable future. However, at this stage of development we are more concerned with the order of growth in practice than in the exact computation time.



(a) Observation computation time



(b) Ranking time



(c) Ranking quality

Fig. 11. Empirical performance results.

Figure 11 (a) evaluates the time it takes to initialize an observation (compute the overlay matrix in Section IV-B) and the time it takes to compute the spatial coverage contribution of an observation relative to a coverage matrix. We recorded the times for an observation represented with different resolutions, from high resolution (0.1m per pixel) to coarse resolution (1m per pixel). As can be seen, the time required decreases rapidly when the resolution becomes more coarse.

Figure 11 (b) shows the time it takes to rank a set of observations at different resolutions. Again, the time required decreases rapidly when the resolution decreases. Finally, Figure 11 (c) shows the impact resolution has on the quality of the ranking. Each data point is an average over 10 randomly generated problem sets of 10 observations each. We ranked the observations at several resolution levels with the highest being 0.1m per pixel and lowest 8m per pixel. For each run, we correlated the ranking of an observation set with the ranking that was produced for that same observation set at the highest resolution. The results show that we achieve high correlation even when the resolution gets very coarse. Thus, the algorithm can perform efficiently at coarse resolutions while still providing high quality rankings.

We have also performed runs of the integrated planning system using a high-fidelity rover simulator [5]. In future work we will evaluate the overall performance of the integrated system.

VI. Related Work

The spatial coverage problem we are solving is similar to the Art Gallery Problem [6] from computational geometry. However efficient solutions to the the 2D Art Gallery Problem do not scale well to 3D. The ROPE (Rank and Overlap Elimination) system selects locations for video cameras for visual surveillance of large 3D open spaces [7]. ROPE uses a greedy algorithm similar to the one used in this paper. However, ROPE does not model the quality of coverage (e.g. observations from a distance may not be as good as close-up observations) nor does it consider the cost of observations.

Dhillon and Chakrabarty present an approach for selecting locations for placing sensors in terrain to provide efficient coverage and surveillance [8]. The objective is to place sensors to provide a given probability that targets will be detected at a given set of grid points. The algorithm takes into account the probability that a target will be detected. This is analogous to our objective of increasing coverage quality of a geographical area. Both approaches allow users to specify priorities on the areas that are covered. Our approach differs in that it does not assume uniform cost for observations but instead uses the planner to assess the cost of performing an observation. Also, when applied to the spatial coverage problem that we are addressing this approach would be too space and time intensive.

The swath coverage problem for orbital satellites is similar to the spatial coverage problem addressed in this paper [9], [10]. While these systems reason about observations costs, planning for surface operations involves distinct types of constraints, choice points and observations modeling. However, our approach does use a similar greedy algorithm for selection observations as used in ASTER [10].

VII. Conclusions

We have presented a set of algorithms that enable a rover to compute the spatial coverage quality of a plan and to rank candidate observations by how well they are expected to improve coverage quality. Using this technique, a rover is better able to assist in the exploration of geological features by generating high quality operations sequences that take into account spatial coverage along with other science considerations. We have currently implemented and tested these algorithms in stand-alone mode as well as integrated into an execution system with a high-fidelity rover simulator. In future work, we will focus on evaluations of the integrated system and on techniques for combining multiple preferences functions so that the system can more effectively trade-off science and engineering objectives when generating and executing plans.

Acknowledgments

The research described in this paper was carried out at the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration. This work was funded by the Jet Propulsion Laboratory Research and Technology Development program.

References

[1] T. Estlin, F. Fisher, D. Gaines, C. Chouinard, S. Schaffer, and I. Nesnas, "Continuous planning and execution for a Mars rover," in *Third International NASA Workshop on Planning and Scheduling for Space*, Houston, TX, October 2002.

[2] S. Chien, R. Knight, A. Stechert, R. Sherwood, and G. Rabideau, "Using iterative repair to improve the responsiveness of planning and scheduling," in *Fifth International Conference on Artificial Intelligence Planning and Scheduling*, Breckenridge, CO, April 2000.

[3] M. Zweben, B. Daun, E. Davis, and M. Deale, "Scheduling and rescheduling with iterative repair," in *Intelligent Scheduling*, M. Fox and M. Zweben, Eds. Morgan Kaufmann Publishers Inc., 1994, pp. 241–256.

[4] A. Shapira, *Fast line-edge intersections on a uniform grid*. Morgan Kaufmann, 1990, ch. 6, pp. 29–37.

[5] A. Jain, J. Guineau, C. Lim, W. Lincoln, M. Pomerantz, G. Sohl, and R. Steele, "Roams: Planetary surface rover simulation environment," in *Proceedings of the International Symposium on Artificial Intelligence, Robotics and Automation in Space*, Nara, Japan, May 2003.

[6] J. O'Rourke, *Art Gallery Theorems and Algorithms*. New York: Oxford University Press, 1987.

[7] S. Rana, "Use of gis for planning visual surveillance installations," in *ESRI Homeland Security GIS Summit*, Denver, 2005.

[8] S. S. Dhillon and K. Chakrabarty, "Sensor placement for effective coverage and surveillance in distributed sensor networks," in *Proceedings of the IEEE Wireless Communications and Networking Conference*, 2003, pp. 1609–1614.

[9] R. Knight and B. Smith, "Optimally solving nadir observation scheduling problems," in *International Symposium on Artificial Intelligence, Robotics, and Automation in Space (i-SAIRAS 2005)*, Germany, September 2005.

[10] H. Muraoka, R. H. Cohen, T. Ohno, and N.Doi, "Aster observation scheduling algorithm," in *SpaceOps 98*, Tokyo, Japan, June 1998.