MULTIPLE-TARGET SINGLE CYCLE INSTRUMENT PLACEMENT

Liam Pedersen(1), Matthew Deans(1), David Lees(1), Srikanth Rajagoplan(1*), David E. Smith(2)

(1)QSS Group, Inc at NASA Ames, NASA ARC, MS 269-3, Moffett Field, CA 94035, USA)
{pedersen, mdeans, lees}@email.arc.nasa.gov
** Currently at Google, Mountain View, CA 94035, USA*
srikanth@google.com
(2)NASA Ames Research Center, Moffett Field, CA 94035, USA
David.E.Smith@nasa.gov

ABSTRACT

Approaching targets and placing instruments on them is fundamental to planetary exploration. Because of communications, power and operational limitations, it currently takes 3 full sol command cycles to accomplish this on Mars with the MER rovers.

To accomplish single cycle instrument placement (SCIP) on multiple targets, we developed and integrated precision visual tracking, off-board contingency planning, robust execution, autonomous instrument placement, round trip data tracking, and a photorealistic virtual reality system to visualize the robot's environment and returned data products, and request further measurements.

Our system has demonstrated a tenfold improvement in robotic capability, as measured by number of samples measured in a single command uplink, by getting 3um/pixel microscopic images from 3 targets designated with 1cm accuracy and up to 10m distant from the rover start position in a single command cycle, executed in under 3 hours.

1. INTRODUCTION

Because of bandwidth and power limitations, the 20-40 minute Earth-Mars signal latency, strict flight rules, and the length of time required to assimilate returned data products and generate a verified activity plan for the day's operations, the MER vehicles currently on Mars require up to three full sol command cycles to approach a distant target and accurately place an instrument against it. One cycle is required to drive the rover up to the vicinity of the target, another for a correction maneuver to bring the target within reach, and a final uplink to command placement of the rover manipulator on the target feature itself.

During the primary mission, amortized MER rover operations cost $4M-$4.5M per day and required 240 operators working 24/7. Speeding this up and reducing operator workload can greatly increase science productivity and reduce cost.

Our goal is to autonomously approach and place an instrument on multiple features of scientific interest up to 10m distant with 1 cm precision in a single command sequence uplink. This is inspired by early design requirements for the 2009 Mars Science Laboratory rover [1], but goes beyond it in the pursuit of multiple targets per command cycle.

Achieving these goals requires broad advances in robotics, autonomy and user interfaces:

Visual Target Tracking – the rover's localization error from deduced reckoning after a 10m traverse is too large to guide it to target with the required accuracy. Targets features must be explicitly tracked as the rover navigates around the worksite. This is complicated by the fact that features are selected for scientific relevance, not ease of visual tracking; and that going to multiple targets means longer duration traverses that may go completely around targets, and significant lighting changes due to changing sun angles or rover shadowing.

Figure 1 : *K9 planetary exploration rover test bed.*

Automated instrument placement – the chosen placement point on a target feature may harm the instrument. For example, putting a microscopic camera up against a protruding rock edge could damage the lens. Even assuming that users can be certain from 10m away that a target point is safe, because of designation, tracking and arm placement errors, there is no certainty that final placement will be on that safe location. The rover must therefore autonomously confirm the safety of a presumed target point, and find alternatives if it is not.

Activity planning and execution – Going to multiple targets implies significant time and energy expenditure

Proc. 'ISAIRAS 2005 Conference', Munich, Germany,
5-8 September 2005 (ESA SP-603, September 2005)

coupled with greater uncertainty about their use. Planetary rovers face tight limits on these resources. In addition, target tracking imposes constraints on the paths a rover can take, which targets it can go to and in what order. Violating these imposes a risk that the targets will be lost (**Figure 5**).

Rapid activity specification and data interpretation tools -- Ground data systems are necessary for users to rapidly identify, prioritize and specify many potential targets, evaluate the plan of action, and understand the data returned from the multiple sites the rover actually visited.

The next sections of this paper describe the underlying assumptions and mission scenario our system is designed for; the technology components addressing the functions identified in the mission scenario; detailed technical descriptions of the activity planning and target tracking technologies; followed by system accomplishments, performance results and conclusions.

2. MISSION SCENARIO

Our mission scenario begins with the rover at the site to be explored and a detailed panorama of stereo images of the area obtained and downloaded to mission control (**Figure 2**) to create a 3D photo-realistic virtual model of the environment.

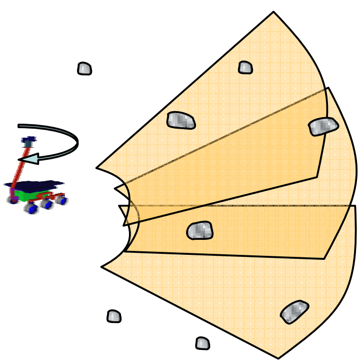

Figure 2 : *Rover has stereo image panorama of worksite at start of mission scenario.*

Observations are requested for each interesting target point, and prioritized with an assigned numerical value. In addition to instrument parameters, users specify *observation points*, locations that the rover must go to in order to do an observation. These will be right up in front of a target if the observation requires putting an instrument in contact with the target.

Additional observation constraints that users may specify include precedence, time of day, and whether a target should be visually tracked (required for 1cm precision microscopic camera placements).

A network of path segments connecting the rover start location and all the observation points and avoiding hazardous zones (specified by users) is generated. Redundant paths are consolidated, and paths more likely to cause tracking failures removed. The rover's onboard obstacle avoidance capability is sufficient to compensate for inaccurate or incomplete specification of obstacles.

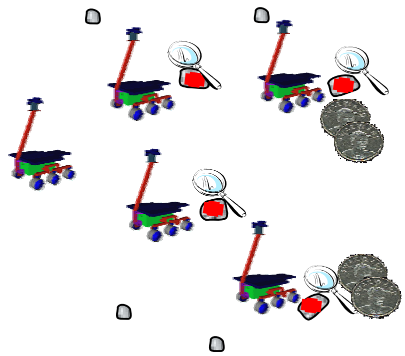

Figure 3 : *Observation requests at start of mission. For each observation, users specify a target point, the instrument (eg microscopic imager), where the rover must be, and a subjective numerical value for the observation.*

An activity plan to get the highest value measurements, subject constraints on the total time and energy, is generated. It is standard mission practice to plan activity sequences on the ground, both to take advantage of greater computing resources available and so that mission operators may modify and verify the sequence prior to execution.

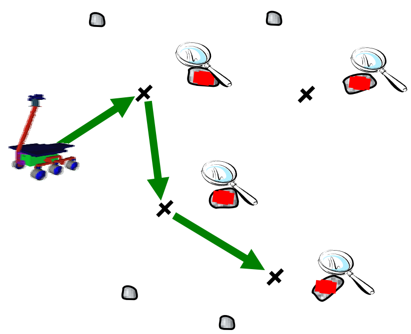

Figure 4 : *Nominal main-line activity plan to get microscopic measurements from highest value targets, subject to constraints on energy, time and target visibility.*

Potential failure points in the plan must therefore be identified *a priori* and contingency actions determined. Pertinent failures include losing track of a target (which will prevent accurate instrument placement on that target) or consuming excessive resources (time and energy) to get to a point, thus putting future measurements at risk.

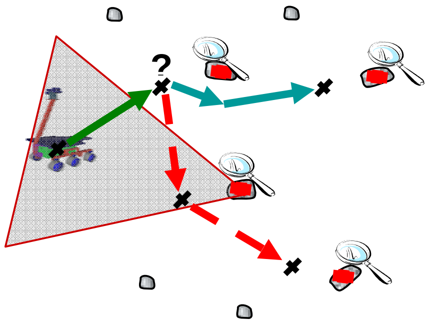

Figure 5 : *Potential target tracking failure identified when rover violates a tracking constraint, conservatively modeled here as a cone shaped region extending from the target to the rover's position when it first imaged the target. Leaving this "tracking region" implies a greater chance of losing track of the target.*

In the example above, it can be seen that in going to the first target, the rover might lose track of a subsequent target and break the remaining plan. Contingency actions to deal with this, short of requesting help from mission control, are to acquire observations from other, still tracked, targets.

Similarly, contingency actions in the event of excessive time or energy consumption include skipping low value targets to conserve time and energy to get to higher value targets.

Users review the plans and make adjustments through changing constraints and observation values, adding or removing targets and editing the path networks. This iterative plan generation requires both a rapid planner and intuitive interfaces to quickly understand the plan generated.

Upon receipt of the sequence, the rover commences navigating to observation points, using stereo vision based tracking of the target features to update their apparent location. Once in front of targets, tracking is handed off from the main science cameras on the rover to the hazard cameras, directly over the manipulator workspace. 3D stereo models of the target area are analyzed to ensure it is safe to place an instrument against the requested target points. If not, the closest safe points on the rock are located and adjustments in the rover base position made, so that the arm may be autonomously deployed and microscopic images of the target obtained. Once done, the rover moves on to the next target until the activity sequence is completed.

Data products obtained by the rover are tagged so that they may be associated with the correct observation requests. Upon being down-linked back to mission control, users can access them through hyperlinks attached to each target in the map based UI used initially to specify targets.

3. TECHNOLOGY COMPONENT OVERVIEW

The end-to-end workflow outlined in the mission scenario description requires integrating multiple components for the users – mission planners and science teams – to assess the rover's environment, plan the day's activities, monitor rover telemetry and view data products in proper context.

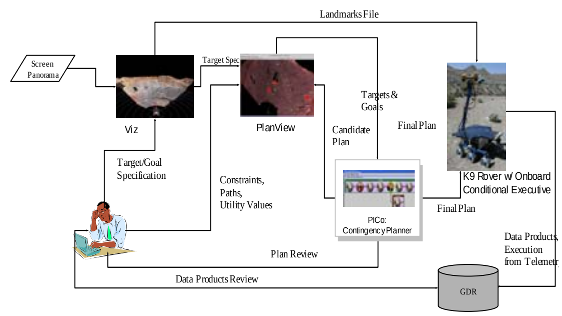

Figure 6: *System overview and information flow*

Viz [2] is a 3D photo-realistic immersive display program for visualizing 3D terrain models of the area around the rover, generated from rover stereo camera panoramas. Viz was originally developed for the 1997 Mars Pathfinder mission, and successfully used by MER for a variety of geo-morphological measurements and virtual exploration of the area surrounding the rovers. Our version of Viz has been significantly enhanced for users to rapidly specify many science targets and observations. An automatic Base Placement algorithm automatically computes the optimal observation point and orientation for the rover that brings the target point within the arm workspace for contact measurements, where necessary. Doing this manually has proven tedious and subject to error.

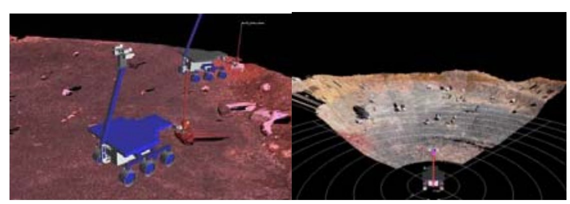

Figure 7 : *Viz, showing target selection and rover base placements.*

PlanView is a Java based program running on a large format, touch sensitive 2D display for users to review requested observations and targets and associated utility values, specify additional daily mission constraints (paths, obstacles, time-of-day constraints), visualize plans and execution traces returned from the planner and rover respectively, and access returned data products. Automatic path generation, as an aid to the mission planner, allows rapid computation of reasonable traverse paths for planning purposes that takes into account tracking and navigation constraints.

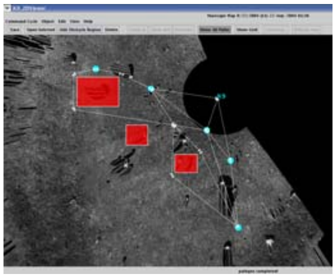

Figure 8 : *PlanView display showing observation targets, rover observation points, user defined obstacle regions and PathGen computed network of paths.*

PICo Planner is the planner that generates rover execution sequences with contingencies to deal with anticipated failures. Inputs are the observation & path specifications from Viz & PlanView. Generated plans are reviewed using the **PICoGUI** interface to understand the temporal characteristics, and PlanView to understand the spatial characteristics of the plan developed.

The **CRL Executive** runs the actual plan on the rover. It is capable of handling concurrent threads of activity, temporal constraints, and contingency branch selection (including so called "floating contingencies" that may occur at any point in execution)

All data transfer between ground based components, including returned data products, is maintained in the Ground Data Repository (**GDR**) – a PostgreSQL database and AFS file system.

The **K9 Rover** is a prototype planetary rover, comparable in size, capability and sensors to the MER rovers. K9 has MER equivalent drive and steering mechanisms, a 5 DOF manipulator arm for deploying a microscopic camera, a suite of mast mounted steerable cameras, hazard cameras overlooking the arm workspace, and standard odometry, inclination and compass sensors that provide deduced reckoning pose estimation with approximately 5% of distance traveled error.

K9's avionics [3] are based around a 1.2 GHz Pentium M laptop, running the Linux operating system and supporting the Coupled Layered Architecture for Robotic Autonomy (CLARAty)[4].

K9 has previously demonstrated automated instrument placement [11].

4. CONTINGENCY PLANNING

Given a set of objectives and their associated values, the PICo planning system selects the subset of objectives to pursue and the detailed commands necessary to achieve them. In addition, it also inserts contingency branches into the plan to cover anticipated potential failures in the plan. This contingency planning is done using an incremental Just-In-Case approach [8] (**Figure 9**). First a "seed" plan is generated using a conventional planning system, assuming that actions have their expected outcomes. This plan is then evaluated to determine where it might fail, given a information about the probability of failure for the different actions, and information about the uncertainty in time and resource consumption of the actions. A branch point is then chosen using heuristics that estimate where a branch is likely to most improve the overall expected utility of the plan. An alternative or *contingency* plan is then constructed for this branch, and incorporated into the primary plan. The resulting conditional plan is again evaluated, and additional branches can be added as needed, either to the original mainline plan, or to already existing contingency branches.

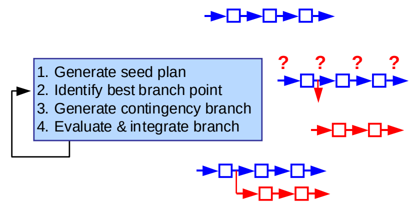

Figure 9 : *PICo algorithm*

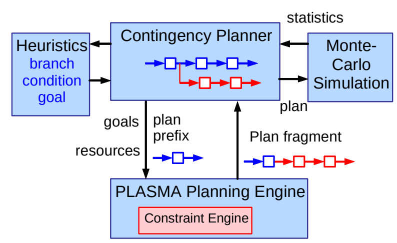

Figure 10 : *Architecture of the contingency planner*

PICO uses of the EUROPA II planning engine to generate both seed plans and contingency branches. EUROPA II is a constraint-based temporal planning engine developed at ARC, building on an earlier version, EUROPA I [9] that is currently in use by the MAPGEN software that generates daily command sequences for the two MER rovers, Spirit and Opportunity. To construct a seed plan, the contingency planner gives EUROPA II a subset of the possible goals, expected resource availability, and expected resource consumption of actions. When the plan comes back, the contingency planner evaluates it using a Monte Carlo simulation to determine the impact of possible tracking failures, and uncertainty in both time and resource usage. To build the branch, the planner again passes an appropriate subset of the goals, the state of the rover at the branch point, and resource availability to EUROPA II. The state of the rover and the resource availability is based on the branch condition and includes the amount of resources (time and energy) available, and the tracking status of the different targets.

The problem of automatically choosing good branch points and good branch conditions is quite hard in general [10]. Intuitively, a good place to put a contingency branch is the place where the plan is most likely to fail. This works well for discrete events like tracking failure, but not for failures associated with consumption of resources like time and resources that are typically exhausted near the end of a plan, at which point few useful options remain. Impending failures must be anticipated earlier in the plan when useful alternatives remain. In other words, the planner must find points in the plan where a contingency branch would significantly increase the overall utility of the plan. Unfortunately, examining and constructing plans for every possible contingency branch is computationally intractable, hence we use heuristics. Among the many different heuristics we have implemented and evaluated, the most effective is to select points where there is high variance in the utility of the plan. For example, suppose there is considerable uncertainty in the amount of time or energy consumed by a particular drive operation. This could impact whether or not science measurements at the end of the plan are possible. As a result, there is considerable variance in the utility of the plan following this drive action. This variance for a given point in the plan is computed from i) the utility of the remainder of the plan as a function of the resources available and ii) the probability distributions for resource availability at that point. Both of these distributions are estimated using Monte Carlo Simulation.

Given a particular branch point, we must also select the branch condition. For tracking failure, this is trivial; we branch if tracking has failed. For continuous resources it is more difficult. In general, the decision depends on and influences the plan that we construct for the contingency branch. If we have a branch plan with high utility, we could be more aggressive about branching, but if the branch plan has low utility, we would like to be more conservative. Unfortunately, we need to first choose a condition in order to generate a good branch plan. Again, we resort to heuristics for doing this. A simple but effective heuristic is to branch when the resources remaining are less than 80% of the expected consumption for the remainder of the plan. A somewhat better heuristic would be to branch when the remaining resources are such that the expected utility of the remainder of the plan drops below some threshold.

In addition to branch and condition selection, the contingency planner must also decide which goals to pursue, both for the original seed plan, and for each contingency branch. This is because the fundamental problem is an *over-subscription* problem – there are more objectives than possible, given the time and resources available. Few planning systems are designed to deal with over-subscription, and those that do, typically adopt a greedy approach to selecting goals. For rover problems, we have found that goal selection is critical to obtaining high quality plans. As a result, we use a sophisticated method for selecting goals, based on the solution of a deterministic *orienteering problem* [12], a variant of the traveling salesman problem in which there are rewards in various cities, the salesman has a finite amount of gas, and the objective is to collect as much reward as possible before running out of gas. **Figure 11** shows a rover problem recast as a deterministic orienteering problem. Note though that the orienteering graph is only an abstraction of the real planning problem, since it leaves out time constraints and other details involved in performing the science experiments. As a result, the solution is only an approximation and may not be optimal or even possible.

Solving this orienteering problem results in an ordered set of goals that can likely be achieved. This information is used by the contingency planner to determine appropriate goals for the EUROPA II planning engine.

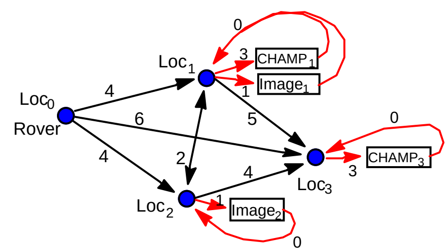

Figure 11 : *Orienteering graph for a rover problem. Edges represent resource usage, rewards are associated with the scientific measurements.*

To summarize, the contingency planner first constructs and solves an orienteering problem to determine an appropriate subset of the goals. This goal set is then fed to the deterministic EUROPA II planning engine to produce a seed plan. The seed plan is evaluated using Monte-Carlo Simulation, and the resulting statistics are used by the branch selection and branch condition heuristics to propose an appropriate branch. A new orienteering problem is constructed and solved to determine an appropriate goal set for the branch, and EUROPA II is invoked again to produce the branch plan. The resulting branch plan is integrated into the mainline plan, evaluated using Monte-Carlo, and the process continues.

Floating contingencies are impractical for the planner to consider because of the large number of possible branch points that they would add to a plan, enlarged continuous space, and additional modeling complexity.

5. VISUAL TRACKING

We have developed a combined feature based and shape based visual tracking system that leverages the benefits of each method in a complementary manner.

We use the SIFT feature detector [14] to find large populations of visual features in successive images of the target of interest. By matching features across a stereo pair, as well as matching pairs before and after robot motion, the tracker can quickly compute a 6-DOF motion, and in a static environment this 6-DOF transformation describes the motion of the tracked point. RANSAC [15] is used to provide robustness to errors during feature detection and matching. RANSAC finds the largest set of putative matches that can be aligned with a single rigid body transformation, rejecting as outliers those points that cannot be aligned.

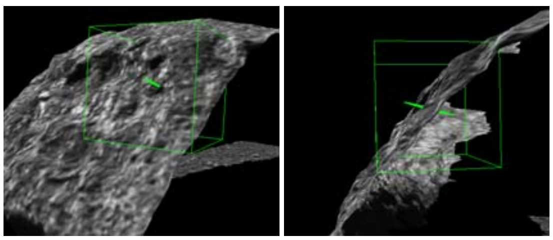

Figure 12 : *Initial target location uncertainty due to subpixel errors in target selection and stereo matching.*

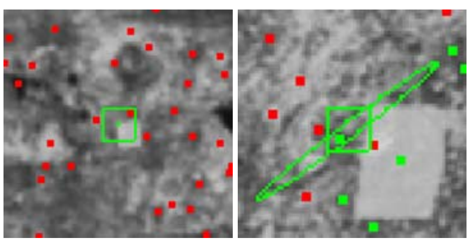

Figure 13 : *(Left) Initial target specification from 10 meters away. The upper left corner of the rectangular marker is chosen as the target of interest. (Right) Final image after robot navigation, the tracked point is only a few pixels away and the correct point is well within the covariance ellipse.*

In addition to tracking the nominal location of the target, the tracker also estimates the uncertainty in the tracker estimate. The initial target selection uncertainty is found using the unscented transform, assuming a half-pixel error in the specification of the pixel coordinates in the initial science camera views to get a covariance matrix over the XYZ position of the specified target, shown in **Figure 12** as an ellipsoid rendered with a stereo model of the rock of interest. Bootstrap is used to recover the uncertainty in each incremental update by recovering the optimal transformation under many random subsets of the inliers found with RANSAC, and the uncertainty in the target location is compounded with the uncertainty in the transform to estimate the uncertainty in the target location after each tracker update.

The tracker also maintains a measure of confidence that the target is still being tracked at all. This confidence measure is a function of the number of points that match between successive views. If RANSAC finds a large number of inliers, then the reported confidence is high. If RANSAC cannot find a solution or only finds a small number of inliers, then the confidence drops. Several low confidence updates in a row will lead to very low confidence and below a threshold the tracker will simply report a failure to the rover executive which may then abort the approach or switch to a different target as described earlier.

If no consistent matches can be found in two consecutive views, the tracker saves the last set of valid interest points and attempts to match them with subsequent incoming images until either a match is found or enough updates have occurred without success for the tracker to abort tracking. Typically the tracker will try 4 or 5 times before the confidence falls below the threshold, and in a few tests the tracker has actually recovered a lost target.

Because the motion recovered by the feature based tracker is incremental, compounding the transformations leads to target drift over time. Once the rover is in front of the target, we match a 3D model of the target taken with the rover science cameras prior to motion with a model obtained from close up with the hazard cameras overlooking the arm workspace, aligning the current view with the original view and eliminating drift incurred by the SIFT feature tracker.

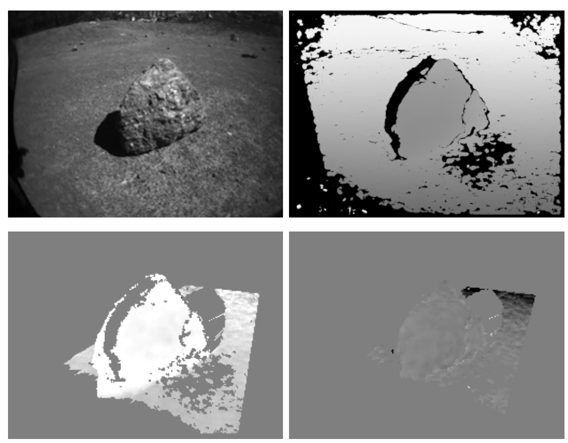

Figure 14 : *Registration result. (top-left) Hazcam images taken when rover arrives at target. (top-right) Depth map from stereo. (bottom-left) Minimum depth error from correlation search. (bottom-right) Depth error after Nelder-Mead optimization.*

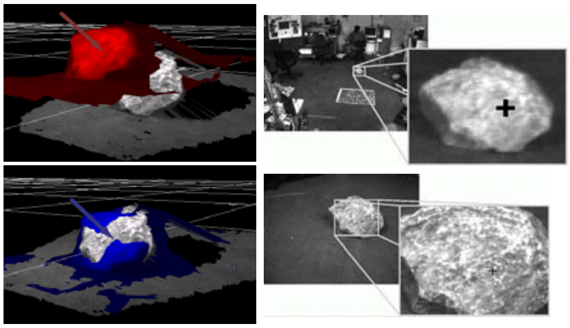

Figure 15 : *The combined tracking system is capable of tracking user specified points for robotic navigation with centimeter accuracy over tens of meters.*

The 3D registration method [16] is based on matching computed range images under different hypotheses for the transformation between views until the transformation that best aligns the models is found. By minimizing the difference between the rendered depths at each point, we can extract a rigid transformation that aligns the two models, thereby allowing us to determine the coordinate transformation between views. The rendering step is fast and eliminates solving a separate correspondence problem using nearest neighbor heuristics such as ICP.

6. PERFORMANCE AND ACCOMPLISMENTS

6.1 ACCURACY AND DISTANCE

We conducted tests on 9/22/2004 – 9/23/2004 at NASA ARC's Marscape test site to gauge the accuracy, reliability and distance limitations of the target tracking, navigation and instrument placement systems.

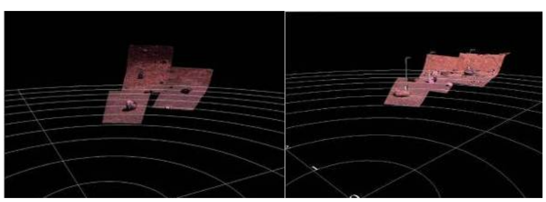

Figure 16 : *Test target arrangements on 9/22/2004 and 9/23/2004. Rover started at origin and was commanded to navigate sequentially to each rock target and place the microscopic imager at operator designated target points (artificially marked). The target rocks are approximately 5m, 7.5m and 10m distant from the origin.*

The only failure in tracking occurred in the feature-based tracker for the second rock on September 23rd when the rover cast a large shadow into the scene. Target position updates reverted to dead reckoning when the tracker failed to find a transformation between subsequent images. After the navigation was finished, the shape-based tracker was able to recover the target with accuracy comparable to the other experiments.

Table 1 *Performance Characteristics from Test Runs (9/22/04 – 9/23/04)*

Target	1 (5m)	2 (7.5m)	3 (10m)
Time to reach target	21 mins	+42 mins	+17 mins
	25 mins	+27 mins	+23 mins
Tracker accuracy	0.68 cm	0.29 cm	1.3 cm
	~0.3 cm	Failed	1.7 cm
Hand-off accuracy	0.5 cm	2.7 cm	1.7 cm
	1.5 cm	~1.6 cm	2.7 cm

6.2 OVERALL SYSTEM PERFORMANCE

On October 28, 2004 we conducted a live integrated demonstration of this system and mission scenario before an audience of Mars Science Laboratory (MSL) and MER mission managers, engineers, and scientists.

The audience spent the morning in the Mission Ops room, reviewing a 3D panorama of the rover worksite, choosing targets and observations, and generating an activity plan with contingencies in case of insufficient time or inability to track targets. In the afternoon, K9 executed the sequence, traversing 28m over 2 hours, 38 minutes to get close up microscopic images of three targets (a 4th target was autonomously rejected by the rover on the grounds that insufficient stereo data was available to guarantee a safe placement)

Up to and including the aforementioned demonstration, our system has achieved the following:

- Instrument placement on 4 targets (1:23 hrs execution).
- Targets up to 10m distant.
- Total traverse distances exceeding 28m per command cycle.
- Up to 1cm tracking and hand-off accuracy.
- Autonomous tool placement safety analysis
- Tracking failure avoidance and recovery.
- Time and resource monitoring and recovery successfully anticipated and avoided future failures.

Qualitatively, once we have the plan running on the rover, we have yet to encounter a situation in which the exec failed because of an unplanned contingency, nor has instrument placement failed once we reached a rock (though our system does occasionally decide not attempt instrument placement if no safe locations on the rock are found).

The keypoint tracker often recovers targets that have been temporarily lost. If it doesn't, and the rover has been driven up to a target, the 3D model registration algorithm can still recover the target.

7. CONCLUSIONS

We have successfully demonstrated a complete integrated system for multi-target single cycle instrument placement, meeting or exceeding early MSL requirements and representing a tenfold increase in capability, as measured by the number of targets investigated per unit time, over the current flown state-of-the-art.

The current MSL mission scenario calls for intensive, long duration analyses of each rock it encounters, thus eroding the relative value of getting to them quickly. Nevertheless, even with a presumed 5 sol dwell time per rock, this single cycle placement capability implies a 30% increase in productivity.

The SCIP system makes possible a strategy of aggressive sample triage to quickly identify promising targets using both remote, close up and contact measurements from multiple features in an area, and returning to select samples for in depth analysis.

This strategy would enable science missions looking for rare phenomena, such as signs of past or present life on Mars. In general, life in extreme environments is both scarce and heterogeneously distributed, finding it requires investigating many locations, diversely distributed at both macroscopic and microscopic scales. Thorough analysis of candidate features is necessary to unambiguously detect life and draw meaningful conclusions. In regions where small fractions (i.e. 0.1% as has been shown in the Atacama Desert) of potential microhabitats actually harbor life, the amount of activity required to carry out a meaningful tele-robotic search could only be accomplished with autonomous multi-target instrument placement.

8. REFERENCES

[1] Krasner, S.M., Tamppari, L., Steve Peters, S., Limonadi, D. (2002), "MSL Scenarios and Autonomy Requirements", MPSET meeting 4/11/2002.

[2] Stoker, C., E. Zbinden, T. Blackmon, B. Kanefsky, J. Hagen, C. Neveu, D. Rasmussen, K. Schwehr, M. Sims (1999), "Analyzing Pathfinder Data Using Virtual Reality and Super-resolved Imaging," Journal of Geophysical Research, vol 104, no E4, pp. 8889-8906, April 25, 1999.

[3] Eric Park, Linda Kobayashi, and Susan Y. Lee. Extensible hardware architecture for mobile robots. In IEEE International Conference on Robotics and Automation, 2005, under submission.

[4] R. Volpe and et al. The claraty architecture for robotic autonomy. In Proceedings of the 2001 IEEE Aerospace Conference, 2001.

[5] J. Bresina, R. Dearden, N. Meuleau, S. Ramakrishnan, D. Smith and R. Washington, "Planning under continuous time and resource uncertainty: a challenge for AI," 19th Conference on Uncertainty in AI, 2002.

[6] J. Bresina and R. Washington. "Robustness via run-time adaptation of contingent plans," AAAI Spring Symposium: Robust Autonomy, 2001.

[7] R. Dearden, N. Meuleau, S. Ramakrishnan, D. Smith and R. Washington, "Incremental contingency planning," ICAPS-03 Workshop on Planning under Uncertainty, 2003.

[8] M. Drummond, J. Bresina and K. Swanson, "Just-In-Case Scheduling," 12th National Conf. on Artificial Intelligence (AAAI-94), 1994.

[9] J. Frank and A. Jónsson, "Constraint-based attribute and interval planning," Constraints, 2003.

[10] N. Meuleau and D. Smith, "Optimal limited contingency planning," 19th Conf. on Uncertainty in AI, 2003.

[11] L. Pedersen, M. Bualat, D. Smith and R. Washington, "Integrated Demonstration of Instrument Placement, Robust Execution and Contingent Planning," 7th International Symposium on Artificial Intelligence and Robotics in Space (iSAIRAS-03), 2003.

[12] C. Keller, "Algorithms to solve the orienteering problem: a comparison," European J. Operational Research 41, 224–231, 1989.

[13] D. Smith, "Choosing objectives in over-subscription planning," 14th Intl. Conf. on Automated Planning and Scheduling (ICAPS-04), 2004.

[14] D. G. Lowe. Distinctive image features from scale invariant keypoints. International Journal of ComputerVision, 60(2):91.110, 2004.

[15] M. A. Fischler and R. C. Bolles. Random sample consensus: a paradigm for model _tting with applications to image analysis and automated cartography. Communications of the ACM, 24(6):381.395, June 1981.

[16] Deans, M., C. Kunz, R. Sargent, L. Pedersen, "Terrain Model Registration for Single Cycle Instrument Placement," in International Conference on Intelligent Robots and Systems (IROS), Las Vegas, USA, 2003.