In *Proceedings IEEE Aerospace Conference*
Big Sky Montana, March 5-12, 2005

Automated Rover Positioning and Instrument Placement

Paul Backes Antonio Diaz-Calderon Matthew Robinson Max Bajracharya Daniel Helmick
Jet Propulsion Laboratory
California Institute of Technology
4800 Oak Grove Drive, Pasadena, CA 91109-8099

Abstract—Technologies to enable automated placement of a rover arm mounted instrument with the rover starting about three meters from the specified instrument target on a terrain feature have been developed and demonstrated on a prototype mars rover. The technologies are automated rover base placement, collision-free arm path planning, and vision guided manipulation. These technologies were integrated with rover visual tracking to provide a complete capability for automated rover approach and instrument placement.

TABLE OF CONTENTS

1. INTRODUCTION

NASA research agenda calls for studies of robots that can substantially increase science return compared to the baseline mission approach. Furthermore, future rover designs are expected to have one or more rover-mounted manipulator arms that place instruments on surface targets. The instruments will be mounted at the end of a manipulator and may include imagers, contact science instruments, surface preparation tools, and sampling tools. Examples of sampling tools are a scoop and corer which will be used to acquire surface samples and deposit them in a sample processing and analysis system located on the rover body. One command cycle (a command sequence uplinked to the rover) is expected to be completed each day. By moving perception and decision functions from human operators on earth to rover on-board functions, the number of required command cycles can be reduced. This has been previously attempted by other authors [1], [2]. In these studies the authors report on systems that achieved one command cycle to a science target from a distance from 1 to 6m from the target. In [1] the authors present a target grasping and instrument placement system for the Rocky 7 Mars Rover prototype [3]. In this work, the authors demonstrated the autonomous acquisition of small rocks (3-5 cm) located over 1 meter in front of the rover. In [2] the authors demonstrated an algorithm for a single command sequence in which feature points are derived on board the rover using the image of the target taken from up to 6 meters away. These features were tracked for docking and instrument arm deployment.

Leveraging from these ideas we have developed algorithms that accurately place an instrument on a target designated from ten meters away. Furthermore, these algorithms take into account the geometry of the environment in which the rover is operating and integrate new technologies such as vision-guided manipulation. Such a system is expected to dramatically improve science return by reducing the number of sols (Martian days) required to approach a science target. For future missions where minimizing the time on the surface of the planet becomes a priority (e.g., the 2013 Mars Sample Return mission), single cycle instrument placement will be a significant mission component.

Currently, three sols are required to drive to a science target and place instruments on the target due to required human operator inputs, which follows the practice of the Mars Exploration Rover (MER) mission. The objective of the work reported in this paper is to achieve autonomous instrument placement to reduce the current baseline to one sol.

Achieving this objective involves the development of the following new technologies: 1) automated rover base placement, 2) arm collision prediction, 3) collision-free arm path planning, 4) vision-guided manipulation, and 5) navigation and tracking. Automated rover base placement computes the rover position to make a target reachable by instruments on a rover-mounted arm. Different poses optimize different instrument task criteria such as positioning accuracy, maximum force application, and magnitude of twist along the approach vector. Arm collision prediction checks whether a given arm configuration will cause collisions between the arm and the rover or terrain. Collision-free arm path planning generates end effector trajectories guaranteed to have no collisions with the rover or the environment. Vision-guided manipulation utilizes camera image feedback to improve manipulator positioning accuracy. The Navigation framework provides a

0-7803-8870-4/05/$20.00/©2005 IEEE
IEEEAC paper # 1486

generic "sense-think-act" architecture, which navigation algorithms implement. This framework provides an implementation of the Morphin Navigation algorithm [9]. The Morphin navigator's navigation cycle consists of using both front and rear stereo pair cameras of the rover to generate a traversability map, then using both a local cost function and global cost function to determine what action to take, and then executing this action. The tracking algorithm [22] enables the rover to autonomously approach a user designated target, approximately 10m away, to within 1m away, while maintaining visual correspondence of the target to within several pixels accuracy. This algorithm attempts to achieve the maintenance of visual correspondence of a target to within several pixels for motions that the rover may undergo during its approach.

The technologies described in this paper will enable future NASA missions (e.g., Mars Science Laboratory and Mars Sample Return) to require only one uplink command sequence to place an instrument on a target with the rover starting many meters from the target.

The contribution of the technologies to single cycle instrument placement is shown by how they would be used in the following example single cycle instrument placement sequence:

1. Specify target from many, e.g., 10, meters away.
2. Drive rover to within 3 meters of target while tracking it.
3. Do camera handoff from navigation cameras to hazard cameras.
4. Base placement to generate rover position to put target within arm workspace.
5. Drive rover to specified rover position.
6. Track target in hazard cameras and generate target information.
7. Plan collision-free arm path to target.
8. Control arm along specified path.
9. Do vision-guided manipulation to accurately place instrument on target.
10. Acquire instrument data.
11. Retract arm.

The work presented in this paper covers steps 4, 7, 8, and 9. Tracking and camera handoff (steps 2, 3, and 6) are described in a companion paper [22].

The remainder of this paper is organized as follows, Section 2 describes the automated rover base placement technology, Section 3 describes arm collision prediction technology, Section 4 describes collision-free arm path planning, Section 5 describes vision-guided manipulation, Section 6 describes the approach followed to implement and integrate single cycle instrument placement, Section 7 presents an execution of the integrated system demonstrating steps 4 through 9.

2. AUTOMATED ROVER BASE PLACEMENT

The purpose of automated rover base placement is to autonomously position a rover so that an instrument on a rover-mounted manipulator can be placed on a specified science target.

For the MER mission, manual approaches are used to specify the rover position and orientation (pose) so that an arm-mounted manipulator can reach a science target. A communication cycle with earth is therefore required before each rover motion. This results in extra sols being used for the communication with earth operators and the final rover pose is acceptable rather than optimal. The reason is that there is always some uncertainty in where the rover will actually end up after a move and earth-based operators stop adjusting the rover pose when the rover pose is good enough to perform the task. If automated rover base placement is used, the rover can move multiple times to get to a better pose than would be achieved via reliance on earth-based operators.

Several researchers have studied the kinematic and dynamic interactions between the mobile platform and the manipulator arm, and have proposed methods for coordinating the base mobility with the arm manipulation [11]–[18]. In particular, Carriker, Khosla, and Krogh [11], [12] formulate the coordination of mobility and manipulation as a nonlinear optimization problem. A general cost function for point-to-point motion in Cartesian space is defined and is minimized using a simulated annealing method. Pin and Culioli [13] [14] define a weighted multi-criteria cost function, which is then optimized using Newton's algorithm. Liu and Lewis [15] describe a decentralized robust controller for a mobile robot by considering the base and the robot as two separate subsystems. Seraji develops a simple on-line approach for coordinated motion control of the manipulator arm and the mobile base [18].

Most of the methods address coordination of mobility and manipulation in which the base and the arm move simultaneously to accomplish user-defined tasks. For rover-based manipulation, we choose to not allow simultaneous motion of the rover and arm. The rover is therefore only a positioner for the arm base. This is operationally simpler, more reliable, and safer.

Based on this we define the general base placement problem as follows: "find a rover location that provides the arm configuration that maximizes a suitably defined manipulability metric." This metric will quantitatively evaluate the ability of the manipulator to arbitrarily change the position and orientation of the end effector at the tip of the manipulator.

The approach used here first defines the preferred task function to optimize, e.g., application of forces, minimization of twist along the approach axis, or positioning accuracy. Using the manipulability metric defined in this section we can optimize for task functions that relate to an ability of manipulation.

Inputs to base placement are the target 3D position, and surface normal. There will be an optimal rover pose to place the manipulator base for each activity type and target. Constraints of actual terrain can be added, e.g., elevation map and obstacle maps. These contraints can be incorporated in the algorithm to find the best rover pose for a given activity type. As a result, the best rover pose that places the rover on the terrain and puts the target within the arm's reachable workspace is computed.

Manipulability Metric

Consider a manipulator with n degrees of freedom. The joint variables are denoted by an *n*-dimensional vector q. an *m*-dimensional vector $r = [r_1 r_2 \cdots r_m]$ describes the position and orientation of the end effector. The kinematic relation between q and r is assumed to be $r = f(q)$. Considering the set of all end effector velocities $v = J(q)\dot{q}$ (where $J(q)$ is the Jacobian of the manipulator), which are realizable by joint velocities such that the Euclidean norm of $\dot{q}$ satisfies $\|\dot{q}\| \leq 1$, we define an ellipsoid in the *m*-dimensional Euclidean space [20]. The major and minor axes of the ellipsoid represent the directions in which the end effector can move at different speeds. For example in the direction of the major axis the end effector can move at higher speeds than in the direction of the minor axis. In the particular case when the ellipsoid is a sphere the end effector can move in all directions uniformly. Such ellipsoid is called manipulability ellipsoid [20] and it represents an ability of manipulation.

A number of manipulability metrics based on the manipulability ellipsoid have been proposed, including the volume of the manipulability ellipsoid, $w_1 = \sqrt{det(J(q)J^T(q))}$, directional uniformity, $w_2 = \sigma_m/\sigma_1$ (i.e., the ratio of the minimum and maximum radii of the ellipsoid), and the upper bound of the magnitude of velocity at which the end effector can move in any direction, $w_3 = \sigma_m$ (minimum radius of the ellipsoid).

Since we are interested in quantifying the ability of manipulation we opted for the volume of the manipulability ellipsoid as the metric for manipulability. This metric has been implemented in the base placement algorithm to compute rover poses that maximize the ability of manipulation at the desired target.

Finding Rover Pose to Maximize Ability of Manipulation

Rover base placement is executed in two steps. First a target is acquired from an initial rover pose (Figure 1), which is at a distance larger than maximum arm reach (typically 10 meters from the target). In this step the algorithm computes an approach pose (location and heading) relative to the current rover frame. Typically the approach pose is defined within 3 meters from the target. Second, once the rover is at the approach pose the algorithm performs a search along the projection of the rover *x* axis on the *x*-*y* plane that maximizes the manipulability metric.

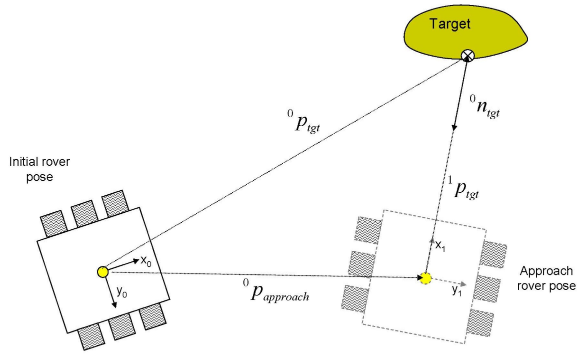


Figure 1. Base placement. The target is a acquired from a distance (typically 10 meters) and a candidate rover pose is computed to initiate target approach.

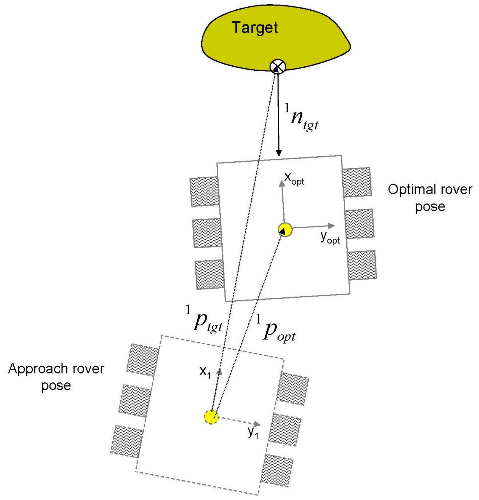


Figure 2. Base placement–within 3 meters of target. The algorithm iterates along the approach vector to compute an updated rover pose. The rover drives to that updated pose and the process repeats until a convergence criteria has been achieved.

With the rover at the approach pose (Figure 2), the high level control loop of the algorithm proceeds as follows: the target is re-acquired and a refined target location and desired target approach vectors are input to the algorithm. A mapping (Algorithm A) from target location to rover pose is computed and the rover drives to the updated rover pose while tracking the target in the hazard cameras. This process repeats until the change of the mapped rover pose is within some predefined tolerance.

Algorithm A *(Mapping of target description to rover pose)*. Given a target location p_{tgt} and surface normal n_{tgt}) compute an optimal rover pose $\mathcal{P}$ (e.g., location and heading) and corresponding joint angles q that maximize the manipulability

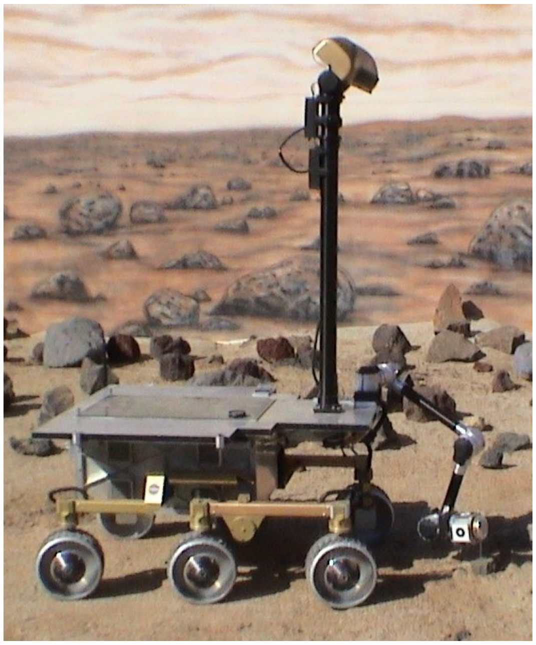


Figure 3. The JPL Rocky 8 rover.

metric w_1.

A1. Compute starting point. Set $n_\perp \leftarrow n_{tgt} - (n_{tgt} \cdot n)n, p_s \leftarrow p_{tgt} + \Delta\, d n_\perp$ where n is the normal vector to the world *x-y* plane. $\Delta\, d$ defines the maximum distance from the target (typically maximum manipulator reach).

A2. Compute starting rover heading. Set $a_\perp \leftarrow -n_\perp, h_s \leftarrow \tan\left(\frac{a_{\perp y}}{a_{\perp x}}\right)$.

A3. Compute tool approach vector. Set $a_g \leftarrow -n_{tgt}$.

A4. Iterate on $a_\perp$. Define points $p^i, i = 0, \ldots, k$ along $a_\perp$. The value of k is the number of discrete points along $a_\perp$. Set $p^0 \leftarrow p_s$.

A5. Compute rover attitude. If a range map exists, get *(x, y, z)* location for each wheel from current rover pose. Compute rover inverse kinematics to get rover attitude relative to ground. If no range map exists, use the assumed rover position with flat and level ground assumption.

A6. Compute arm configuration. Do inverse kinematics of arm for the current rover pose and goal tool approach vector. Set $q_i \leftarrow g(^i\ p_{tgt}, \mathcal{O}(a_g))$, where $^i\ p_{tgt}$ is the location of the target relative to the current rover pose and $\mathcal{O}(a_g)$ is the desired orientation of the tool. If $q_i = \emptyset$ go to step A8.

A7. Compute volume of manipulability ellipsoid. Set $w_1^i \leftarrow \sqrt{det(J(q_i)J^T(q_i))}$.

A8. Compute next trial point. Set $p^{i+1} \leftarrow p^i + \delta\, d$. If $p^{i+1} \leq D_{max}$ go to step A5.

A9. Select best rover pose. If $w_1^i \neq \emptyset$, set $j \leftarrow ndx(\max(w_1^i)), \mathcal{P} \leftarrow [p^j, h_s]^T, q \leftarrow q^j$.

A10. No valid arm positions were found. If $w_1^i = \emptyset$, then return that the target is not reachable. ■

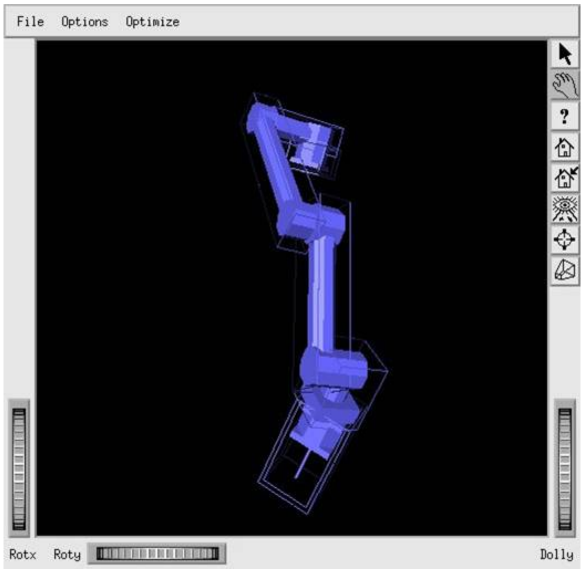


Figure 4. Oriented bounding box representation of the Rocky 8 arm.

Rover base placement has been integrated and demonstrated on Rocky8 (Figure 3) a JPL rover.

3. Arm Collision Prediction

Allowing an arm motion to follow a rover motion as part of one uplink command sequence requires that the arm motion be tested for potential collisions with the rover and terrain before executing the arm motion. In the current state-of-the-art represented by the MER mission, on-board arm collision prediction is only done between the arm and rover but not between the arm and terrain. For MER, collisions between the arm and terrain are tested by human operators on earth thereby imposing the operational constraint in the MER mission that arm motions cannot follow rover motions as part of one uplink command sequence. The single cycle instrument placement and single cycle sample acquisition capabilities desired by future NASA missions both require arm motions to follow rover motions as part of one uplink command sequence, so both require the automated arm collision prediction with the rover and terrain that is reported in this paper.

The approach being described here is a hybrid model-based (using a model of the arm and rover to check for collisions) and sensor-based (using stereo cameras to build an obstacle model) approach that is being used in the MER mission for on-board collision prediction between the arm and rover and by earth operators to predict collisions between the arm and terrain [19]. In this approach each part of the rover or arm is modeled using Oriented Bounding Boxes (OBBs) objects (Figure 4). Cylindrical parts are efficiently modeled using Oriented Bounding Prisms (OBPs) consisting of a series of OBBs. The terrain is bounded using a series of OBBs.

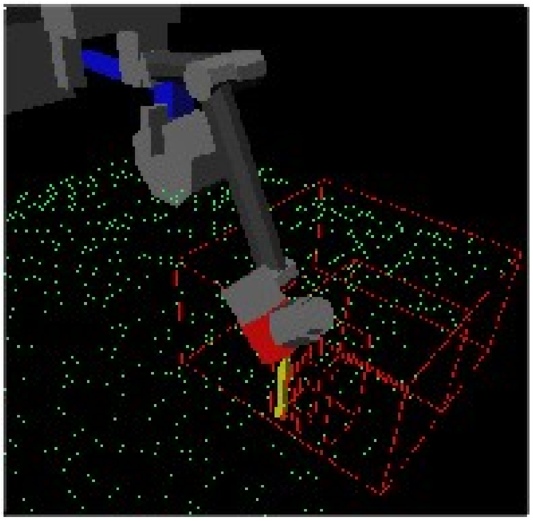


Figure 5. Collision checking with the terrain.

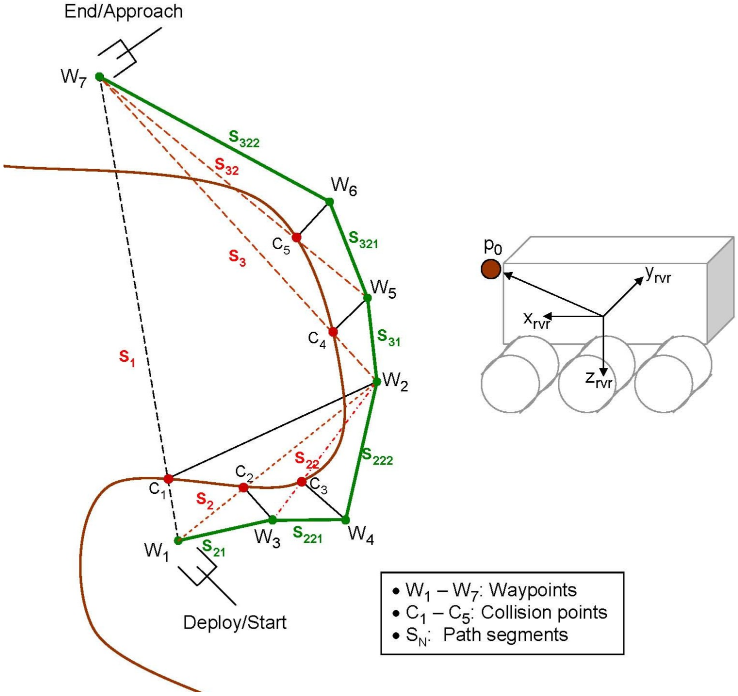


Figure 6. Planning a collision-free arm trajectory.

The basic approach of collision prediction is to determine whether two OBBs overlap. This is done efficiently and accurately by representing the manipulator arm, rover, and terrain obstacles in a volumetric octree. For example each link of the manipulator arm is represented by a high level OBB object that encompasses the entire link geometry. At lower levels of the octree the link geometry is more tightly bound by a series of OBBs and OBPs. The software first checks for collisions between the coarse high-level OBB objects. In most cases this is sufficient since no collision will be detected. If a collision is found between two high-level OBB objects, the next lower level of objects that more tightly bound the geometry is checked. The detected collision may be a result of the coarseness of the high-level OBB objects versus an actual collision. Therefore checking the next lower level of objects may rule out a collision. Each subsequent level of the tree is checked until either the collision is ruled out or the lowest level of objects, which most tightly bound the geometry, is checked and an actual collision is detected. The alternative of checking collisions between each OBB or OBP object at the lowest level of the tree is computationally inefficient.

The terrain is represented in a volumetric octree as well. To do so the raw stereo data is converted into an elevation map. The height of each grid cell in the elevation map is the maximum height of all stereo data points lying in the grid cell. Initially a single grid cell represents the entire range map. If a collision is detected between the manipulator arm and the cell the grid cell is divided into two cells of equal size, which more tightly bound the local terrain. Each new grid cell is checked for a collision with the arm. If a collision is detected with one of the two new cells, that cell is divided into two cells of equal size. This process continues until either a collision is ruled out or a collision is detected with sufficient resolution. This process is shown in Figure 5.

4. Collision-free Arm Path Planning

Collision-free arm path planning algorithm is intended to be used for single cycle instrument placement on the Rocky8 rover. The algorithm operates on Cartesian paths generated from the given start and end points and utilizes the collision prediction algorithm described in this paper to test whether there is a collision along the path.

When designing this algorithm care was taken to accommodate the limited computing resources available on-board the rover. The goal was to design an algorithm that would utilize minimal set of resources. As a result we have designed a geometric-based approach that reasons about the geometric state of the arm and the environment. The paths result in a Cartesian motion of the end effector that follows the obstacle profile at a preset distance from the obstacle (Figure 6).

To describe the algorithm consider the example shown in Figure 6. In this case the end-effector of the arm is at the starting point (W_1) and it is commanded to go to point W_7. The straight line motion from W_1 to W_7 passes through the obstacle so that path is not feasible. At this point we proceed as follows: generate a straight line path (S_1) between the start (W_1) and end (W_7) Cartesian points. Test for collisions along the path and if there is a collision *split* this path into two subpaths (S_2 and S_3) at the collision point (C_1). Independently solve each of the sub-paths (i.e., test for collisions and split if needed) and then combine the sub-path solutions so as to yield a solution for the original path (S_1) from start to end.

Splitting computes a new waypoint that is guaranteed to be

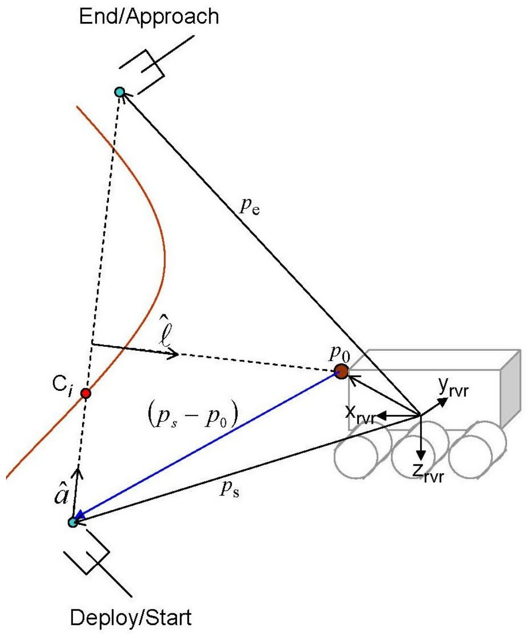


Figure 7. Geometry to compute a candidate waypoint along the arm trajectory.

collision free (e.g., $W_i, i = 2, \ldots, 6$) in Figure 6). The waypoint is derived from the geometric configuration at which the collision occurred. This is formally described in the following algorithm.

Algorithm B *(Split)*. Given a path segment $S = <\wp_s, \wp_e>$, collision point $C_i = (x_c, y_c, z_c)$ along the line connecting start $\wp_s$ and end $\wp_e$ poses ($\wp = [\wp_p, \wp_o]^T$, $\wp_p = [x, y, z]^T$ and $\wp_o = [az, el]^T$ i.e., position and orientation of the end-effector), and arm mount frame p_0 all relative to a defined rover frame F_{rvr} (Figure 6), compute a collision-free waypoint.

B1. Compute the vector that defines the line joining the starting $\wp_s$ and ending pose $\wp_e$. Set $\vec{a} \leftarrow \wp_{p_e} - \wp_{p_s}, \hat{a} \leftarrow \vec{a}/\|\vec{a}\|$.

B2. Compute the vector $\vec{\ell}$ normal to $\hat{a}$ that passes through p_0. Set $\vec{\ell} \leftarrow (\wp_{p_s} - p_0) - ((\wp_{p_s} - p_0)\boldsymbol{.}\hat{a})\hat{a}, \hat{\ell} \leftarrow \vec{\ell}/\|\vec{\ell}\|$ (Figure 7).

B3. Compute a candidate waypoint. Set $W_i \leftarrow C_i + \Delta\, d \times \hat{\ell}$ (Figure 8). The scalar $\Delta\, d$ is the step size to take (in meters) so that the waypoint results in a pose with no collisions.

B4. Compute candidate arm pose. Set $\wp_{W_i} \leftarrow [W_i, \wp_{o_s}]$. Compute inverse kinematics to find arm configuration q_{W_i} at candidate arm pose $\wp_{W_i}$. Set $q_{W_i} \leftarrow g(\wp_{W_i})$.

B5. Test for arm collisions at candidate arm configuration, q_{W_i}. If arm configuration is collision free go to step B6. Otherwise set $C_i \leftarrow W_i$, go to step B3.

B6. Return W_i. ■

The geometric-based collision-free arm path planning is described in Algorithm C.

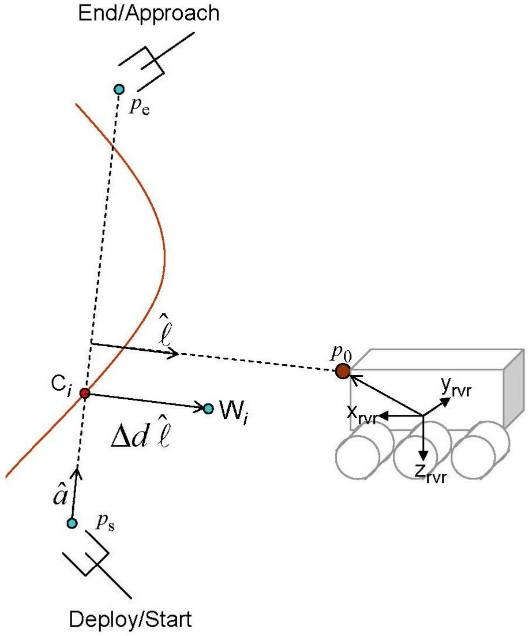


Figure 8. Computing a candidate waypoint along the arm trajectory. The algorithm iterates along the line defined by $\hat{\ell}$ taking small $\Delta\, d$ steps until the waypoint results in no arm collisions.

Algorithm C *(Collision free arm path planning)*. Given an initial arm configuration, q_s, and a target location, p_{tgt}, and target normal, $\hat{n}_{tgt}$ (both in rover frame F_{rvr}, compute a collision-free path from the starting configuration to the target arm configuration. Assume: 1) arm is assumed to be in a deployed configuration, 2) the current configuration of the arm is the starting point of the path, 3) the starting and ending configurations are the same.

C1. Compute forward kinematics to obtain current arm orientation. Set $\wp_s \leftarrow f(q_s)$.

C2. Compute target pose. Set $\wp_{tgt} \leftarrow [p_{tgt}, \wp_o(\hat{n}_{tgt})]^T$.

C3. Do inverse kinematics to compute target arm configuration. Set $q_{tgt} \leftarrow g(\wp_{tgt})$

C4. Compute approach pose. Set $\wp_a \leftarrow [p_{tgt} + \Delta\, d \times \hat{n}_{tgt}, \wp_{s_o}], \wp_e \leftarrow \wp_a$

C5. Do inverse kinematics to compute approach configuration. Set $q_a \leftarrow g(\wp_a)$.

C6. Confirm that approach configuration (q_a) is the same as target configuration q_{tgt}. If $q_a \neq q_{tgt}$, go to step C8.

C7. Compute collision-free arm path for $<\wp_s, \wp_e>$.

C7a. Compute straight line path from start to end pose. Set $\mathcal{S} \leftarrow path(\wp_s, \wp_e)$.

C7b. Test for collisions. Set $\mathcal{C} \leftarrow collision\_check(S)$ ($\mathcal{C}$ is a list of points along the path that collide with the environment).

If $\mathcal{C} \neq \emptyset$ go to step C7c, otherwise go to C7d.

C7c. There is a collision. Split the segment, set $W_i \leftarrow split(\mathcal{S}, \mathcal{C}_0), \wp_{W_i} \leftarrow [W_i, \wp_{o_s}], S_1 \leftarrow <\wp_s, \wp_{W_i}>, S_2 \leftarrow <\wp_{W_i}, \wp_e>$. Call C7 with S_1. Call C7 with S_2. Go to C8.

C7d. There are no collisions. Merge the straight line path $\mathcal{S}$ into the collision-free path. The collision-free path includes the start and end points and the new waypoints that have been generated. Set $\mathcal{S}_{free} \leftarrow merge(\mathcal{S}, \mathcal{S}_{free})$. Go to C8.

C8. Return. ■

In the example in Figure 6, collision point C_1 spawns sub-paths S_2 and S_3. Solving sub-path S_2 independently results in a new collision point C_2, which spawns sub-paths S_{21} and S_{22}. Solving sup-path S_{21} results in S_{21} itself since there are no collisions along this path. Sub-path S_{22} is split into sub-path S_{221} and S_{222}. Since there are no collisions along these paths the solution for sub-path S_2 is complete and it is given by combining sub-paths $<S_{21}, S_{221}, S_{222}>$. Sub-path S_3 is solved in a similar manner and the solution for the starting path S_1 is complete.

5. Vision-Guided Manipulation

Vision-guided manipulation utilizes camera image feedback to improve manipulator positioning accuracy. To this end, we make use of a new calibration-free approach to robotic manipulation known as Hybrid Image-Plane/Stereo (HIPS) to improve arm-mounted instrument placement accuracy beyond the limits of typical calibrated stereo methods [4]. The HIPS technique can be separated into two approaches: the off-line model approach and the real-time approach.

In the off-line approach a single camera model is generated for the entire workspace. This is done by acquiring samples of both the image-plane location of a fiducial marker on the end-effector and the 3D location of the marker (as given by the joint angles and the manipulator kinematic model) at a series of pre-defined poses throughout the workspace, as shown in Figure 9. Upon generating the model the 3D location of the target with respect to the manipulator coordinate frame is determined. The joint angles that achieve this goal then are computed with the same manipulator kinematic model used to generate the camera model.

The off-line model approach accounts for kinematic uncertainties such as link lengths and separate camera and manipulator frames. However, stochastic uncertainties such as fiducial marker detection resolution, joint angle knowledge, etc., remain unaccounted for. Therefore, in the real-time approach the camera model is updated with additional joint and image samples during the approach to the target to locally refine the model. Thus, the camera model is made locally accurate leading to an improvement in positioning accuracy.

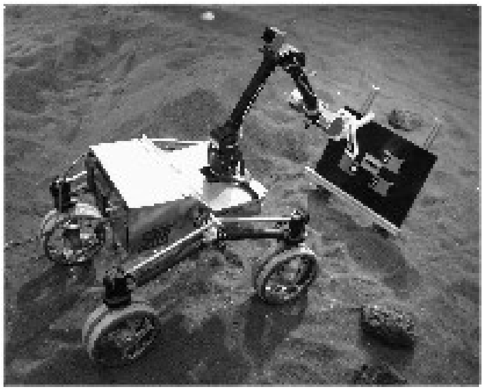


Figure 9. Sample Return Rover (SRR) with fiducial marker on manipulator and target board for validation testing.

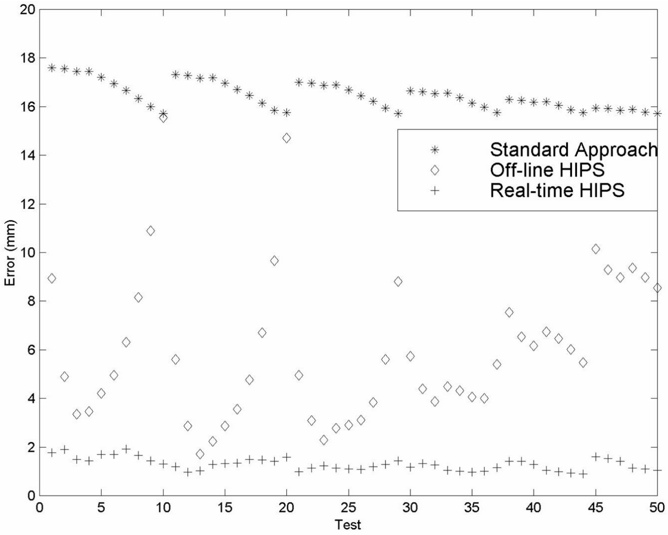


Figure 10. Results of simulated positioning tests using the standard flight approach (calibrated stereo), using the HIPS off-line model approach, and using the real-time HIPS approach.

Benefits of HIPS

The benefits in using the HIPS approach are apparent in the results of a simulation study shown in Figure 10. The simulation study involved a series of positioning exercises of a four degree-of-freedom manipulator relative to a target. For these positioning simulations a significant error was introduced into the arm kinematics (a combined total of over 2.0 cm change in the link lengths) without providing knowledge or modifying the nominal kinematic model of the manipulator. In addition a “truth” camera model was introduced to map target locations from three-dimensional space into each image plane of a simulated stereo camera pair.

For the study three sets of fifty positioning exercises were performed. In the first set of tests, depicted with a *, the standard flight approach (i.e., MER mission) was used to position the manipulator. The mean terminal error using the standard approach was 16.5 mm. In the final two series of simulated tests HIPS was used to control the manipulator to the target. The off-line HIPS approach, depicted by a $\diamond$, achieved an average terminal error of 6.4 mm, a 60% improvement from the standard flight approach. Finally, the real-time HIPS approach achieved an average error of 1.30 mm. This represents an order of magnitude increase in accuracy from the standard flight approach.

Comparison to Vision-guided Manipulation State-of-the-Art

Significant effort has been devoted to image-based visual servoing, primarily in the laboratory environment [5], [6], [7]. In image-based visual servoing a feedback control loop comprising the difference between the current and goal manipulator states as measured in the camera image-plane is used to drive the manipulator to a zero image-plane error state.

While visual servoing has achieved some success in unstructured environments outside the laboratory there are several major advantages for using HIPS in space applications [8]. By definition image-based visual servoing relies on continuous updates of the manipulator-target image error [5]. In space-based applications this is impractical due to the limited processor speed and available camera frame rate. The resulting long delays could create controller instability and final positioning error [5]. In addition, the target is often obscured in the image-plane near the manipulator terminus due to the limited choice of camera and manipulator configurations. Conversely, the HIPS approach is limited by neither frame rate nor constant access to image-plane error. As new samples become available the camera models are updated and the goal position is refined. However, the manipulator can still be controlled to the target in the absence of new information.

Visual servoing utilizes only the most recent image to compute the image error, i.e. the control variable. A fundamental limit on terminal precision then is the error associated with target extraction from a single, possibly noisy, image. Alternatively, HIPS yields improved precision by estimation of the manipulator-generated camera models based on a history of image-plane appearances and internal joint angles of the manipulator.

HIPS is most similar to an alternative technique for hand-eye coordination known as Camera-Space Manipulation (CSM) [10]. The CSM method has been shown to achieve excellent terminal precision (less than 1 mm position and 1.0 orientation) when the participating cameras are widely separated with a vergence of greater than 60. Unfortunately, the placement of widely spaced cameras on a rover platform is difficult to achieve due to the finite size of a rover and the use of existing platform cameras configured as stereo pairs for rover navigation. HIPS is designed to achieve the precision of CSM using stereo cameras.

6. IMPLEMENTATION APPROACH

The automated rover positioning and instrument placement system was implemented in the Coupled Layer Architecture for Robotic Autonomy (CLARAty) [21]. CLARAty is a two layer software architecture that provides robotic functionality and simplifies the integration of new technologies on robotic platforms: the Functional and Decision Layer. The Functional Layer provides both low- and mid-level autonomy capabilities. The Decision Layer integrates AI software. It globally reasons about the intended goals, system resources, and state of the system and its environment. The Decision Layer uses a declarative-based model while the Functional Layer uses a procedural-based model.

Through abstraction of the hardware layers, CLARAty enables software components available in the architecture to be transparently used on 4 custom research rovers (Rocky 7, Rocky 8, K9, and FIDO), one commercial platform (ATRV Jr.), and benchtop duplicates of these systems' avionics. Leveraging from the use of hardware layers the system described in this paper ran on the Rocky 8 rover.

The Rocky 8 rover is a six wheel drive/six wheel steer mars rover research prototype that includes a 5 degree of freedom arm (DOF), and a 2 DOF mast (Figure 3). There are 2 stereo camera pairs mounted on the mast head at 1024x768. One stereo pair includes 6mm lenses at 20cm baseline while the other stereo pair includes 16mm lenses at 30cm baseline. There are two more stereo pairs mounted on the body below the deck at a 45 deg angle with respect to the deck. These hazard cameras (hazcams) are 640x480 both stereo pairs with 2.3mm lenses at 8.5cm baseline. All cameras are dragonflys (1394, Sony CCDs).

The 5 DOF arm is mounted on top of the deck as shown in Figure 11. This arm provides two yaw joints (first and last joint) and three pitch joints. Hardware level motion control is implemented via JPL's distributed widget board architecture. Software motion control is implemented in the CLARAty architecture and includes Cartesian and joint space motion and collision-free arm path planning (described in Section 4).

The 2 DOF mast is a fixed mast that allows for pan and tilt of the mast head. It is primarily used for tracking and panoramas. Tracking, however, uses only the stereo pair with 6mm lenses [22]. Hazard cameras are used for navigation and tracking during base placement/instrument placement and HIPS. Hazcams are also used for position estimation via visual odometry to improve estimation of the vehicle position.

7. INTEGRATED SYSTEM DEMONSTRATION

This section presents initial results of an integrated system, which includes the technologies described in this paper. The technologies integrated include:

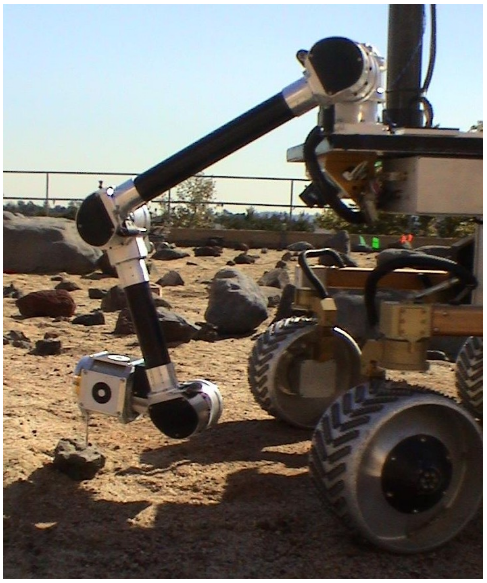


Figure 11. Rocky 8 5 DOF arm. Arm is deployed to place an instrument on a science target. Fiducial markers are mounted on the turret. These markers are used by HIPS to accurately position the instrument on the target. Front hazcams are shown mounted below the deck.

- Kinematic and image-based target “handoff” from the pan/tilt mast navigation cameras to the fixed body mounted hazard cameras [22].
- Iterative rover base placement to maximize arm manipulability.
- Target tracking in hazard cameras during base placement approach [22].
- Instrument placement
  - Collision-free arm path planning
  - Image-based instrument placement using HIPS (Hybrid Image-Plane Stereo) manipulation.

In the current scenario the rover takes a number of panoramic images with the cameras mounted on the mast. These images are sent back to the ground operator to be analyzed to determine specific targets of interest. These panoramic images are taken at a distance of 10m from the targets. Once the images are analyzed the operator selects the target of interest on the images and sends a command to the rover to place an instrument on the selected target.

Once the rover receives the command, the autonomous (single command) instrument placement system initiates execution.

Approach Rover triangulates target and issues navigation command to a goal location (computed by base placement) 3m away from target (between starting location and target) with 0.25m (radius) tolerance. While rover navigates to the goal location the rover analyzes the terrain using the images captured through the hazcams and issues arc drive command. At intervals of 30cm/10 deg visual odometry provides an updated estimate of the current rover location. At the completion of the arc command tracking re-orients the mast and re-triangulates the target position. This process is repeated until the goal is reached.

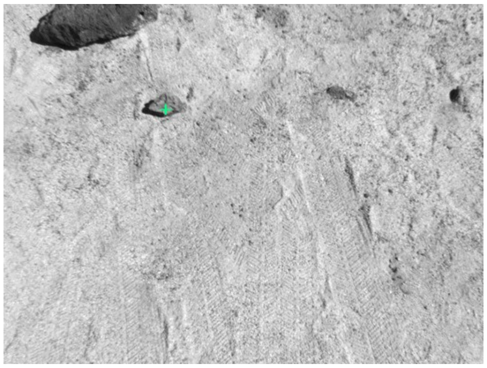


Figure 12. Target in the mast cameras at a distance of about 3 meters.

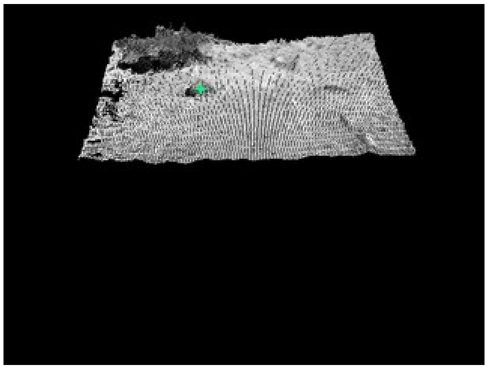


Figure 13. Projection of the mast camera image onto the hazcams.

Handoff At the goal location (e.g., 3m away from target), handoff takes place by re-projecting tracked target from mast cams to hazcams (Figure 12, Figure 13, Figure 14).

Base placement The last approach to the target is executed by calling base placement algorithm to compute updates on the desired rover pose. After computing an updated rover pose, the rover drives to the computed pose and in doing so it runs visual odometry to update rover position and tracks/retriangulates the target position. Once an updated target position is computed, the process starts again until some number of iterations have been completed or the change on updated rover poses is negligible (Figure 15).

Instrument placement This is the last step of the execution of the automated instrument placement command. At this point the rover is placed at the optimal pose and the arm is unstowed. The arm executes a collision-free arm motion to above target and then the control is handed over to HIPS to execute high accuracy motion to reach the target (Figure 16).

At the end of the command the instrument is deployed on the target (Figure 17, Figure 18) and science tasks can begin.

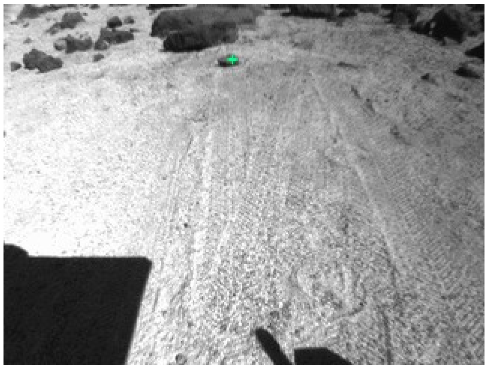


Figure 14. Target matched in the hazcam image.

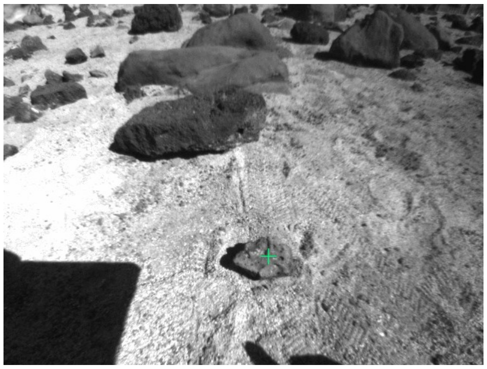


Figure 15. Target is retriangulated after achieving the location that maximizes arm manipulability.

Based on the above scenario, initial experiments of the integrated end-to-end system were executed on the Rocky 8 rover in the JPL Mars Yard starting from about 10 meters from the target. A task is currently validating and determining the accuracy and repeatability of the entire system. The potential sources of error (approach tracking, handoff, base placement tracking, and instrument placement) have all been identified but not all rigorously tested. Using the mast cameras, tracking between frames is generally accurate to less than a pixel, and less than several pixels over the entire 10m traverse. The 2D tracking algorithm has gone through a validation process ([23]) and shown to be as accurate as 4cm (3σ), but has been augmented in as part of the overall system to be more robust to failure.

8. Conclusions

We have presented collision-free arm path planning, base placement, and vision-guided manipulation technologies that enable automated placement of an instrument mounted on a rover arm with the rover starting at about ten meters away from the target. Critical to achieving the goal is the availability of other supporting technologies, which include tracking, camera handoff, navigation and kinematic mast pointing. Combining all these technologies results in a fully automated system capable of placing an instrument on a target with a single command cycle. This system was demonstrated on Rocky 8, a prototype mars rover. It is expected that the technologies described herein will enable single command instrument placement that will reduce the current 3 sol MER baseline to one sol.

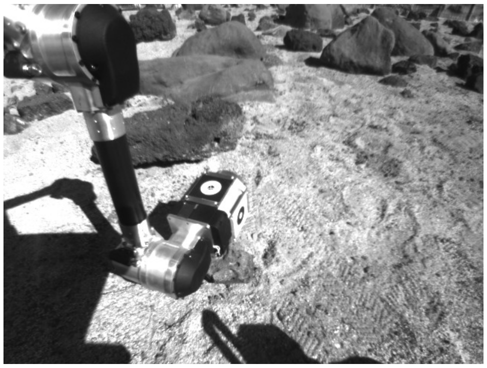


Figure 16. Final approach to target using image-based instrument placement. Tracking of the fiducial markers is done through the hazcams.

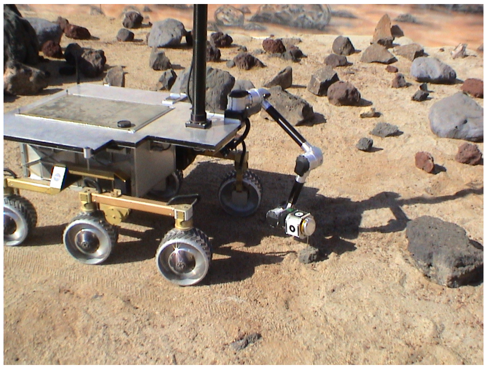


Figure 17. End result of single command instrument placement.

9. 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.

References

[1] I. A. Nesnas, M. Maimone, and H. Das, “Autonomous Vision-Based Manipulation from a Rover Platform,” *IEEE Symposium on Computational Intelligence in Robotics & Automation*. Monterey, CA, November 1999.

[2] T. Huntsberger, H. Aghazarian, Y. Cheng, E. T. Baumgartner, E. Tunstel, C. Leger, A. Trebi-Ollennu, and P. S. Schenker, “Rover autonomy for long range navigation and science data acquisition on planetary surfaces.” *IEEE International Conference on Robotics and Automation*. Washington DC, May 2002.

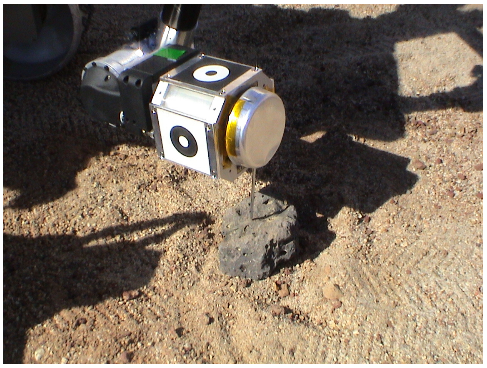


Figure 18. End result of single command instrument placement.

[3] R. Volpe, J. Balaram, T. Ohm, R. Ivlev. "Rocky 7: A Next Generation Mars Rover Prototype." *Journal of Advanced Robotics*., 11(4), December 1997.

[4] T. Huntsberger, Y. Cheng, E. T. Baumgartner, M. Robinson, P. Schenker, "Sensory fusion for planetary surface robotic navigation, rendevouz, and manipulation operations." *Proc. International Conference on Advanced Robotics (ICAR'03)*, University of Coimba, Portugal, June 30-July 3, 2003.

[5] S. Hutchinson, G. Hager, and P. Corke, "A tutorial on visual servo control", *IEEE Trans. Robotics and Automation*, vol. 12, no. 5, pp 651–670, Oct. 1996.

[6] N. R. Gans, P. I. Corke, and S. A. Hutchinson, "Performance tests of partitioned approaches to visual servo control," *IEEE International Conference on Robotics and Automation*, Washington DC, May 2002.

[7] D. Kriegman, G. Hager, and A. Morse, eds, *The confluence of vision and control*, Springer Verlag, 1998.

[8] C. Urmson, B. Shamah, J. Teza, M. D. Wagner, D. Apostolopoulos, and W. L. Whittaker, "A sensor arm for robotic antarctic meteorite search," in *Proceedings of the 3rd International Conference on Field and Service Robotics*, Helsinki, Finland, July 2001.

[9] C. Urmson, R. Simmons and I. Nesnas, "A Generic Framework for Robotic Navigation," in *Proceedings of the 2003 IEEE Aerospace Conference*, Big Sky Montana, March 8-15 2003.

[10] S. B. Skaar, W. H. Brockman, and R. Hanson, "Camera space manipulation," *Int. Journal of Robotics Research*, vol. 6, no. 4, pp. 20-32, Winter 1987.

[11] W. F. Carriker, P. K. Khosla, and B. H. Krogh, "Path planning for mobile manipulators for multiple task execution," *IEEE Trans. Robotics and Automation*, vol. 7, no. 3, pp. 403–408, 1991.

[12] W. F. Carriker, P. K. Khosla, and B. H. Krogh, "The use of simulated annealing to solve the mobile manipulator planning problem," *Proceedings IEEE International Conference on Robotics and Automation*, pp. 204–209, 1990.

[13] F. G. Pin and Culioli J. C., "Optimal positioning of redundant manipulator–platform systems for maximum task efficiency," *Proc. Int. Symp. Rob. Manuf.* 1990, pp. 489–495.

[14] F. G. Pin and Culioli J. C., "Multi-criteria position and configuration optimization for redundant platform/manipulator systems," *Proceedings IEEE Workshop on Intelligent Robots and Systems*, 1990, pp. 103–107.

[15] K. Liu and Lewis F. L., "Decentralized continuous robust controller for mobile robots," *Proceedings IEEE International Conference on Robotics and Automation*, 1990, pp. 1822–1827.

[16] N. A. M. Hootsmans and Dubowski S., "Large motion control of mobile manipulators including vehicle suspension characteristics," *Proceedings IEEE International Conference on Robotics and Automation*, 1991, pp. 2336–2341.

[17] G. L. Long and Paul R. R. "Minimum energy consumption and the optimal placement of robot manipulators on mobile platforms," *Proc. Second National Conf. on Applied Mechanisms and Robotics*, 1991, pp. III B1-1–III B1-5.

[18] H. Seraji, "An on-line approach to coordinated mobility and manipulation," *Proceedings IEEE International Conference on Robotics and Automation*, 1993, pp. 28–35.

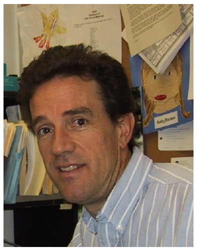
[19] C. Leger, "Efficient sensor/model based on-line collision detection for planetary manipulators," *Proceedings IEEE International Conference on Robotics and Automation*, May 2002.

[20] T. Yoshikawa, *Foundations of robotics analysis and control*, MIT Press, Cambridge Massachusetts, 1990.

[21] R. Volpe, I.A.D. Nesnas, T. Estlin, D. Mutz, R. Petras, and H. Das, "The CLARAty Architecture for Robotic Autonomy." *Proceedings of the 2001 IEEE Aerospace Conference*, Big Sky Montana, March 10-17 2001.

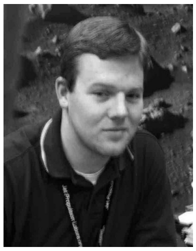
[22] M. Bajracharya, A. Diaz-Calderon, M. Robinson, and M. Powell, "Target tracking, approach and camera handoff for automated instrument placement," *Proceedings of the 2005 IEEE Aerospace Conference*, Big Sky Montana, March 5-12 2005.

[23] W. Kim, R. Steinke, R. Steele, "2-D Target Tracking Technology Validation Report", JPL Technical Report D-28523, Apr. 2004.



Paul Backes, Ph.D. *Paul Backes is a technical group leader in the Autonomy and Control section at Jet Propulsion Laboratory, Pasadena, CA, where he has been since 1987. He received the BSME degree from U.C. Berkeley in 1982 and Ph.D. in 1987 in Mechanical Engineering from Purdue University. Dr. Backes received the 1993 NASA Exceptional Engineering Achievement Medal for his contributions to space telerobotics, 1993 Space Station Award of Merit, Best Paper Award at the 1994 World Automation Congress, 1995 JPL Technology and Applications Program Exceptional Service Award, 1998 JPL Award for Excellence, 1998 NASA Software of the Year Award Sole Runner-up, and 2004 NASA Software of the Year Award. He has served as an Associate Editor of the IEEE Robotics and Automation Society Magazine.*

Dr. Antonio Diaz-Calderon *Antonio Diaz-Calderon is a member of Information Systems and Computer Science Staff with the Telerobotics Research and Applications Group at the Jet Propulsion Laboratory, Pasadena, CA. His research work at JPL involves manipulation kinematics, optimal estimation and motion control. He received the BSEE in 1987 and the M.S. in Computer Science in 1990 from Monterrey Institute of Technology (Mexico City, Mexico), and M.S. in 1993 in Engineering and Ph.D. in 2000 in Electrical and Computer Engineering from Carnegie Mellon University. He held a Post-Doctoral Fellowship appointment with The Robotics Institute at Carnegie Mellon University (2000-2001), and a faculty appointment as Commercialization Specialist with The Robotics Institute (2001-2003). While at the Robotics Institute Dr. Diaz-Calderon developed an estimation-based tipover sensing system to enhance dynamic stability of heavy articulated machines, robust position estimation algorithms for the PerceptOR program and vehicle dynamic models used for mobility analyses on prototype vehicles for Future Combat Systems' Unmanned Ground Combat Systems (UGCV), and Gladiator. His research interests include: sensor fusion and state estimation, and rover navigation and mobility.*



Dr. Matthew Robinson *Matthew L. Robinson received the B.S. in 1996 and the Ph.D. in 2001, both in mechanical engineering from the University of Notre Dame. Currently, he is a member of the Engineering Staff in the Mechanical and Robotic Technologies Group at the Jet Propulsion Laboratory in Pasadena, CA. At JPL he has both flight mission and research experience. Currently, he contributes to the 2007 Phoenix mission (a lander equipped with a back-hoe style robotic arm and various scientific instruments destined for the Martian pole) as the flight software lead for the robotic arm. His research efforts have included the development of calibration-free, stereo-based manipulation techniques (HIPS) and control systems for holonomic and hybrid nonholonomic/holonomic. His research interests include machine vision, simulation, manipulation, mechatronic systems and controls.*

Max Bajracharya *Max Bajracharya is a member of the Telerobotics Research and Applications Group at the Jet Propulsion Laboratory, Pasadena, CA. His research includes vision-based tracking, navigation, and pose estimation for rovers and he is currently a lead systems and software engineer for the CLARAty software project and the Rocky 8 research rover. He received his Bachelors and Masters degrees in computer science and electrical engineering from MIT in 2001.*

Daniel Helmick *Daniel Helmick received his B.S degree in Mechanical Engineering from Virginia Polytechnic Institute and State University and his M.S. in Mechanical Engineering with a specialization in controls from Georgia Institute of Technology in 1996 and 1999 respectively. Since June 1999 he has been working at the Jet Propulsion Laboratory in the Telerobotics Research and Applications Group on robotics research projects involving vision/sensor based control of robots, state estimation, and navigation and mobility algorithms. He has worked on robotic vehicles covering a wide range of functionality, including: Mars research rovers for rough terrain mobility; small, tracked robots for urban mobility; a cryobot for ice penetration; and reconfigurable wheeled robots for Mars exploration. His research interests include: sensor-based control of robots, sensor fusion and state estimation, and rover navigation and mobility.*