

Visual End-Effector Position Error Compensation for Planetary Robotics

Max Bajracharya, Matthew DiCicco,
Paul Backes, and Kevin Nickels
*Mobility and Manipulation Group 3473
Jet Propulsion Laboratory
Pasadena, California 91109
e-mail: maxb@robotics.jpl.nasa.gov*

Received 2 June 2006; accepted 29 January 2007

This paper describes a vision-guided manipulation algorithm that improves arm end-effector positioning to subpixel accuracy and meets the highly restrictive imaging and computational constraints of a planetary robotic flight system. Analytical, simulation-based, and experimental analyses of the algorithm's effectiveness and sensitivity to camera and arm model error is presented along with results on several prototype research systems and "ground-in-the-loop" technology experiments on the Mars Exploration Rover (MER) vehicles. A computationally efficient and robust subpixel end-effector fiducial detector that is instrumental to the algorithm's ability to achieve high accuracy is also described along with its validation results on MER data. © 2007 Wiley Periodicals, Inc.

1. INTRODUCTION

The ability to autonomously accurately place an arm end-effector on a target selected from a camera image is becoming a common requirement in many landed planetary exploration missions. All foreseeable future and many past landed missions to Mars include a robotic arm and stereo camera pairs. The 1999 Mars Polar Lander (MPL) carried a four degree-of-freedom (DOF) arm (Bonitz et al., 2001) to be used for soil trenching and digging, as well as placement of the Robotic Arm Camera (RAC) (Keller et al., 2001). The 2003 Mars Exploration Rover (MER) vehicles carry a 5-DOF arm that can place one of three *in situ* instruments or a rock abrasion tool (Trebi-Ollennu, Baumgartner, Leger, & Bonitz, 2005). Targets are designated

via a stereo camera pair mounted on the rover body (Tunstel et al., 2005). The 2007 Phoenix Lander will carry a robotic arm of MPL descent and use it to scrape samples from the surface and transport them to its on-board instruments (Smith, 2004). The 2009 Mars Science Laboratory (MSL) vehicle will carry a 5-DOF arm that will be used both to place *in situ* instruments as well as obtain samples to deposit in its on-board instruments (Savage & Cook-Anderson, 2004). Like MER, targets will be designated from stereo camera pairs (either body or mast mounted). Future Mars missions, like the Astrobiology Field Laboratory (AFL), the mid-rovers, or Mars Sample Return, will also all include a robotic arm and stereo camera pairs.

Furthermore, with each successive mission, the demands for increased science return and the use of more sophisticated instruments result in a requirement for higher placement accuracy. The need of MSL and future missions to draw a sample core, deposit it into a sample processor, and then return to the same core hole for additional science all require an accuracy better than the 1 cm requirement imposed on MER.

While vision-guided manipulation is routinely used in industrial applications to accurately place an arm end-effector, planetary exploration missions have many unique constraints that limit the implementation of standard approaches. Because electronics must be radiation hardened, designed to handle extreme thermal cycling, and use minimal power, processors, memory, and camera electronics tend to be several generations behind those available commercially. This severely limits both the available computation capability and camera frame rate. For example, MER flew a RAD6000 20 MHz processor running approximately 100 tasks (Reeves & Snyder, 2005) and required several seconds to obtain a full image (Maki et al., 2003). Furthermore, tele-operation is limited because of communication time delays and many times a highly accurate calibration cannot be relied on due to the process of launch, landing, and thermal cycling.

The End-effector Position Error Compensation (EPEC) algorithm described here was developed in response to the need for increased placement accuracy under the constraints of a flight system. Unlike many visual servoing methods, EPEC only requires a single iteration of imaging and corrective motion. The algorithm's novelty is its reliance on calculating a local error between the arm and camera model, and then applying it to future local motions. As a consequence, the algorithm does not rely on the convergence properties of a servo loop. Additionally, while the algorithm is designed to be used online, it can also be used offline by building a 3D table of corrections throughout the workspace and applying the closest one, or with the "ground-in-the-loop," in which images are downlinked, a correction calculated by a ground system, and a new command uplinked.

After prior and related work is reviewed in Section 2, the algorithm is described in Section 3, with an analysis given in Section 4, and experimental results provided in Section 5.

2. PRIOR AND RELATED WORK

Currently, landed Mars missions only use open-loop arm positioning based on well-calibrated arm and camera models. Targets are selected and triangulated in a stereo image pair that has been calibrated in the vehicle's frame of reference. Inverse kinematics with gravity sag compensation is then used to calculate the arm joint angles, to which the joints are servoed. On the MER rovers, this technique results in a positioning error of approximately 1 cm (Baumgartner et al., 2005) (also see Figure 16).

During the extended mission of MER, a vision-based calibration technique was used to update the camera models used to designate the target. The Hybrid Image-Plane/Stereo (HIPS) algorithm [(Baumgartner & Schenker, 1996; Seeling, Baumgartner, & Skaar, 2002), based on the Camera Space Manipulation (CSM) (Skaar, Brockman, & Hanson, 1987) concept, but modified to work for very wide field of view cameras with a target close to the camera], detects the end-effector position in several locations throughout the workspace and then augments the existing camera model calibration data taken before launch and recalibrates the cameras using a nonlinear least-squares adjustment. This calibrates the camera models in an arm-centric frame of reference, local to the arm's workspace, and accounts for arm model errors or uncompensated gravity sag. However, because this approach is expensive (requiring imaging at many arm positions and the computation of a camera recalibration), it is best used only periodically throughout the mission to update the camera models used for manipulation. With these models, the EPEC algorithm can be used online to improve the placement with a single image and correction, rather than using an iterative recalibration as the arm approaches the target. Using an online method also guards against model error in novel poses, as models generated by HIPS may not generalize well outside of the region where the data was taken.

Another space related application is Robonaut, a robotic astronaut's assistant being developed at Johnson Space Center, for which a hand-eye calibration method has been proposed for more accurate grasping (Nickels, 2003). This technique uses an end-effector mounted spherical target that is detected in the robot's head mounted stereo cameras to calibrate its 7DOF arm's kinematic model.

In addition, many other standard visual servoing techniques also exist (Hutchinson, Hager, & Corke,

1996) and have been applied to industrial robots. Many of the approaches handle the more complex task of servoing an uncalibrated or weakly calibrated end-effector mounted camera system with no 3D knowledge of the target (Malis, Chaumette, & Boudet, 1999; Smith, Brandt, & Papanikolopoulos, 1997). Other approaches use a static stereo camera system to observe the end-effector, but rely on high bandwidth imagery to close a control loop with the camera images providing feedback (Allen, Timenko, Yoshimi, & Michelman, 1992; Nelson, Papanikolopoulos, & Khosla, 1996). With respect to previous implementations, the algorithm being presented is an endpoint closed-loop (ECL) system, with a stereo camera observation of a fiducial that is offset from the end-effector. It combines standard visual servoing techniques while taking advantage of a reasonably well-calibrated stereo camera pair and arm to reduce computation time and operate with only periodic image acquisition. The major difference between EPEC and traditional visual servo systems is that because EPEC models the local error between the commanded and observed position in stereo, it requires only one iteration of the correction to be effective. This allows for either the use of newly calculated corrections or a lookup table to apply an appropriate correction, calculated earlier, without additional use of the stereo cameras. A critical advantage of this approach for *in situ* rover operations is the inherent safety. All commands from EPEC can be observed by ground based controllers before being uploaded to the spacecraft. The arm motion need not be left solely to an on-board closed-loop visual servo system, and, while most visual servo systems improve positioning accuracy with a single iteration, the analysis described here shows that EPEC can achieve an accuracy close to the minimum observable error while being computationally efficient and robust enough to be used on a Mars rover in highly variable environmental conditions.

3. ALGORITHM DESCRIPTION

The End-effector Position Error Compensation (EPEC) algorithm locates and triangulates the position of a single fiducial on an arm end-effector and servos the end-effector to a position designated in a stereo image pair. In doing so, the EPEC algorithm maintains an error correction vector between the fiducial's position as predicted by the forward kinematics of the arm and as found by the stereo camera

pair. It then applies this correction vector as a position offset to the commanded arm position to compensate for kinematic and camera model inaccuracies. If the fiducial is occluded or imaging is no longer available, a previously calculated correction vector can be used locally to reduce positioning error in the final placement. The algorithm requires that the arm be calibrated well enough to execute accurate small cartesian motions and the stereo cameras be calibrated well enough to triangulate the position of a point found in the two cameras. In fact, because metric reconstruction is not necessary, weakly calibrated cameras would be sufficient; however, the transformation between the stereo camera frame and the arm base frame must be approximately known in order to facilitate predicting the fiducial position.

The input to the EPEC algorithm is a target selected in an image of a stereo camera pair, providing the position of the target in the camera frame. The arm is then placed in a configuration that allows the fiducial to be seen by the camera; the fiducial is detected, and its position triangulated in the stereo camera frame. The difference between the arm's position as determined by forward kinematics (in the arm frame) and by stereo (in the camera frame) results in a "correction vector". This correction vector is then added to the desired tool pose and the end-effector is repositioned to this corrected pose in the camera frame. If the arm and cameras are calibrated perfectly, the correction vector would be zero and kinematic placement would be sufficient. Obviously, since there will be errors in the arm and camera models, the correction vector will be non-zero; however, because it is a function of the arm configuration and location in the image, the error can be applied locally. That is, a correction vector calculated in one location can be applied to correct for errors in the surrounding region of the workspace. While the correction vector only needs to be calculated once near the target, recalculating it as the arm approaches its final position will further reduce the position error. A general block diagram of the algorithm is shown in Figure 1, and a detailed block diagram showing how it is used in a complete arm control system is shown in Figure 2.

3.1. Target Selection

The desired end-effector target is selected in one of the images of the stereo pair and standard subpixel correlation-based stereo is used to determine the 3D location of the target. For the camera configuration

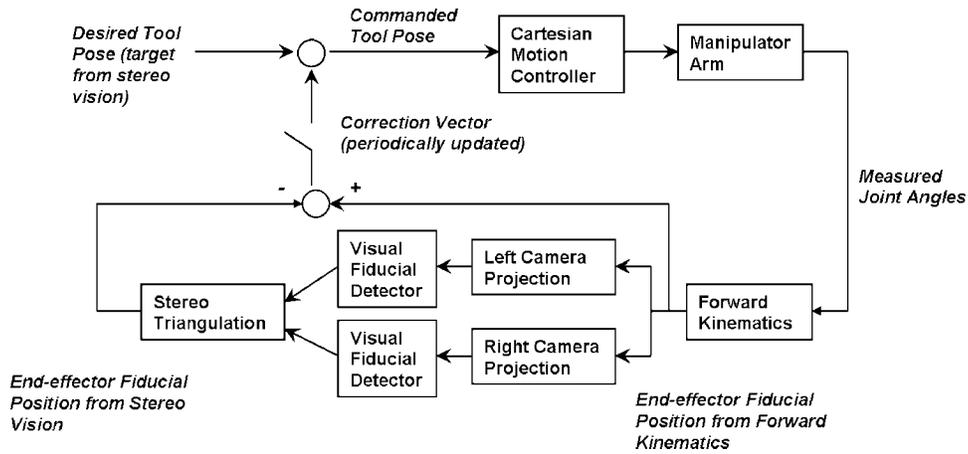


Figure 1. EPEC algorithm block diagram.

of the MER vehicles (Maki et al., 2003), the range error associated with the selection due to correlation error (generally around 0.3 pixels) leads to less than 1 cm of range error at a range of 1 m. This error is compensated for by moving to an offset position and then approaching the target along its surface normal until contact. Lateral error due to selection is better than 1 mm for the camera configurations generally available on Mars rovers.

3.2. Sub-pixel Fiducial Detection

In order to servo the end-effector to the target, the algorithm must be able to localize the end-effector in the stereo camera pair. This is done by detecting the position of a fiducial on the arm's end-effector (see, for example, Figure 3). To achieve maximum robustness and accuracy, the approach taken was to match a directed edge template of the fiducial by shifting

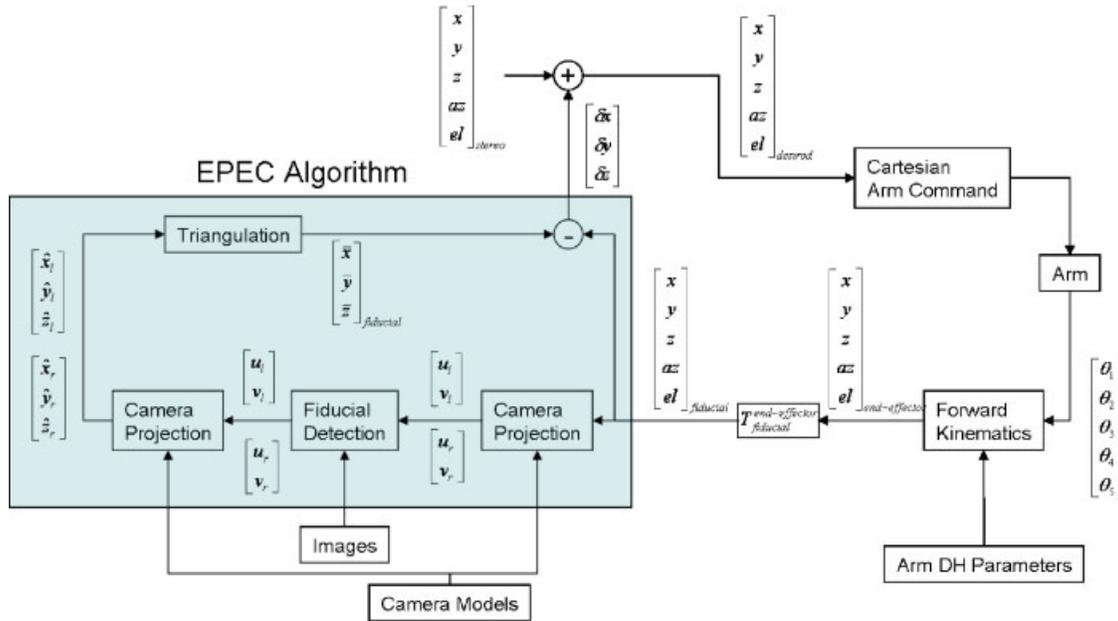


Figure 2. EPEC algorithm detailed block diagram.

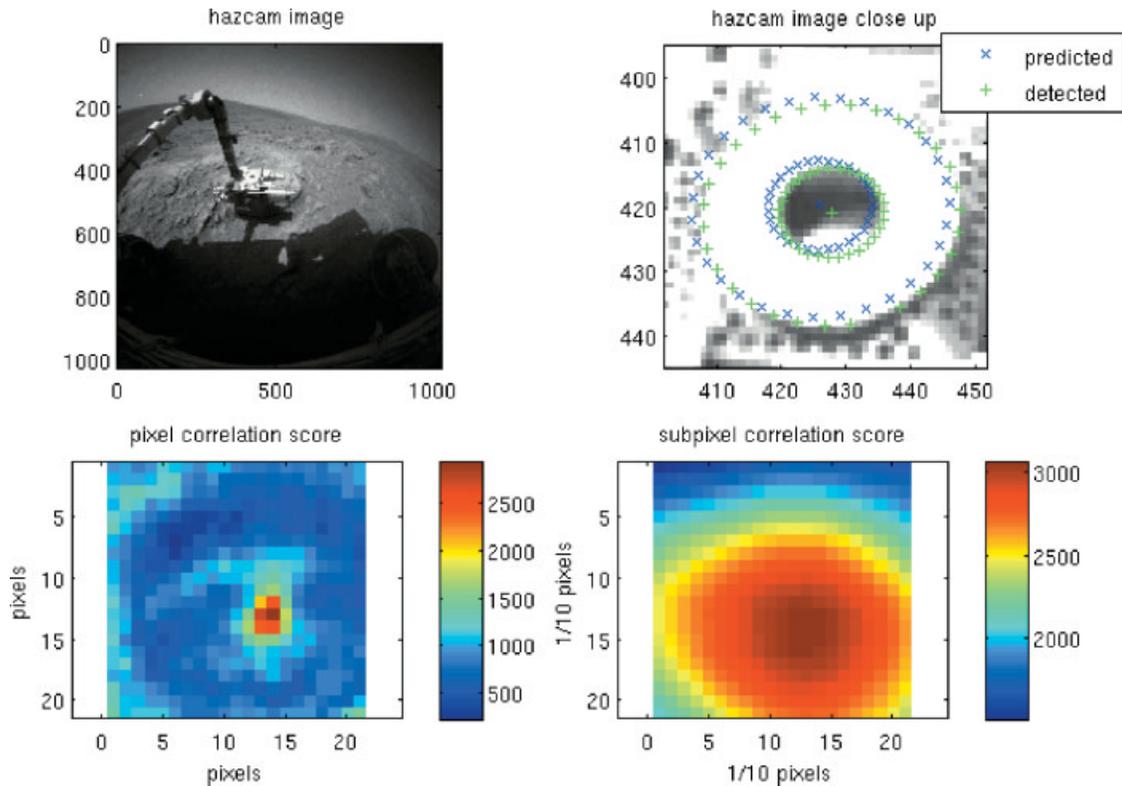


Figure 3. Pixel and sub-pixel correlation scores for the predicted and detected sampled template in an MER hazcam image.

the template with increasing step resolution in a local area in the image. Although using a feature detector [a Harris corner detector (Shi & Tomasi, 1994) for example] is fast and invariant to affine transformations, it is prone to false positives, particularly with direct sunlight on the specular metal end-effector. With a good prediction, a standard intensity patch template correlation method [using normalized cross correlation (Lewis, 1995)] would be robust to false matches like spurious points or edges, but is computationally expensive, requiring every pixel in the template window to be shifted and compared rather than only a fixed number of edge points. A further motivation for using a correlation approach is that, while fiducials were not explicitly placed on the MER arm, portions of the instruments on the end-effector can serve the purpose but are difficult to detect with a feature detector. The contact ring of the Mössbauer Spectrometer appears as two concentric circles and provides a unique target with respect

to the environment and spacecraft visible by the cameras. However, unlike a standard corner fiducial, a model-based approach has the disadvantage of requiring foreshortening compensation to accurately localize the fiducial center.

The algorithm implemented is a correlation-based approach using only step features to improve robustness and speed. Using the camera, arm, and fiducial models and the current arm position as determined by kinematics, a template of sampled fiducial edges is created and then correlated across the image. For a given arm pose, its fiducial template is created by sampling the predicted position of the rings' edges (of radius r_{inner} and r_{outer}) at angles θ sampled between 0 and 2π and projecting these positions to 2D image coordinates (p_s , the sampled edge coordinates) via camera and arm models (M_{world}^{camera} and $T_{fiducial}^{world}$). Conceptually, this can be written as

$$p_s = M_{world}^{camera} T_{fiducial}^{world} \begin{bmatrix} r \cos \theta \\ r \sin \theta \\ 0 \\ 1 \end{bmatrix} \quad \forall \theta \in [0, 2\pi],$$

$$r \in \{r_{inner}, r_{outer}\}, \quad (1)$$

and the center of the fiducial in the image (c) is then simply

$$c = M_{world}^{camera} T_{fiducial}^{world} \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}. \quad (2)$$

However, camera projection is actually computed using the CAHVOR model and method in (Gennery, 2001). The camera model accounts for all image parameters (including distortion) and consequently works for very wide field of view images without requiring the expense of image rectification. While the parametric template implemented is specific to the MER fiducial, any template with edges or step functions could be used.

To find the fiducial, the template is correlated across a local area (of window size w) in the image with incrementally smaller shift steps (Δp). Initially, the template is correlated at only the pixel level. Once the maximum score has been found (at $\Delta \bar{p}$) the template is correlated at subpixel steps to localize the template further. The window size w is determined *a priori* based on the expected error in the arm and camera models. A smaller search window will result in the algorithm running faster and being less prone to false matches, but will be less robust to unexpected large model errors.

The correlation score is the sum of directional derivatives (tangent to the edge) at each of the image (I) coordinates (p) in the direction of the fiducial center (c), with the maximum over all the shifts representing the best match (with offset $\Delta \bar{p}$),

$$\Delta \bar{p} = \arg \max_{\Delta p} \sum_{p=p_s+\Delta p} \left[I \left(p - \frac{p-c}{\|p-c\|} \right) - I \left(p + \frac{p-c}{\|p-c\|} \right) \right]^2 \quad \forall \Delta p \in \left[-\frac{w}{2}, \frac{w}{2} \right]. \quad (3)$$

An example of the initial sub-sampled template

points and the locations that maximize the correlation score are shown as the dark blue \times and light green $+$ marks, respectively, in the top-right of Figure 3. The corresponding correlation scores using a 1 pixel and 0.1 pixel step size are shown in the bottom row of Figure 3. Note that the curvature of the peak of the subpixel step size is an indication of the accuracy that can be achieved.

Fiducial detection in ideal lab conditions on a mock-up arm is shown in Figure 4. However, detecting the fiducial in outdoor environment conditions is significantly more difficult (as can be seen by the images in Figures 5 and 6). As a result, to increase robustness, several detection metrics are used to reduce false positives. These include the fiducial aspect ratio, which is directly related to the end-effector pose, the intensity difference between the inner and outer ring, the stereo ray gap (see Section 3.3), and the correlation score. The combination of threshold parameters was optimized for the MER rovers over a large set of previously taken images of the Mössbauer ring. Using a training set of images randomly selected from all MER data, the fiducial detection algorithm was run and then manually tagged as correct or incorrect by several independent viewers. An exhaustive search of all parameter thresholds was then performed to find the thresholds that maximized true positives while keeping the false positive rate below 1%. This analysis showed that thresholding on only the ray gap and ring contrast resulted in the best performance. The detection and false positive rate of the fiducial detector on imagery with the fiducial in view available for the two MER mission vehicles is summarized in Table I. Example images of correct fiducial detection and rejection of false positives are shown in Figures 5 and 6.

The approach taken is faster than a standard template correlation approach because it samples the model to edge points and only uses image values at these points. It is also lighting invariant because it is only summing directional derivatives (a step function in intensity). Like a pyramidal correlation approach, it also only looks for a maximum correlation score at the pixel level initially, and then refines this by shifting with sub-pixel steps. The method exploits the highly accurate arm and camera models available on most flight vehicles to generate a good model and only correlate in a very local area. The main motivation for using a prediction and correlation approach rather than an ellipse edge detecting

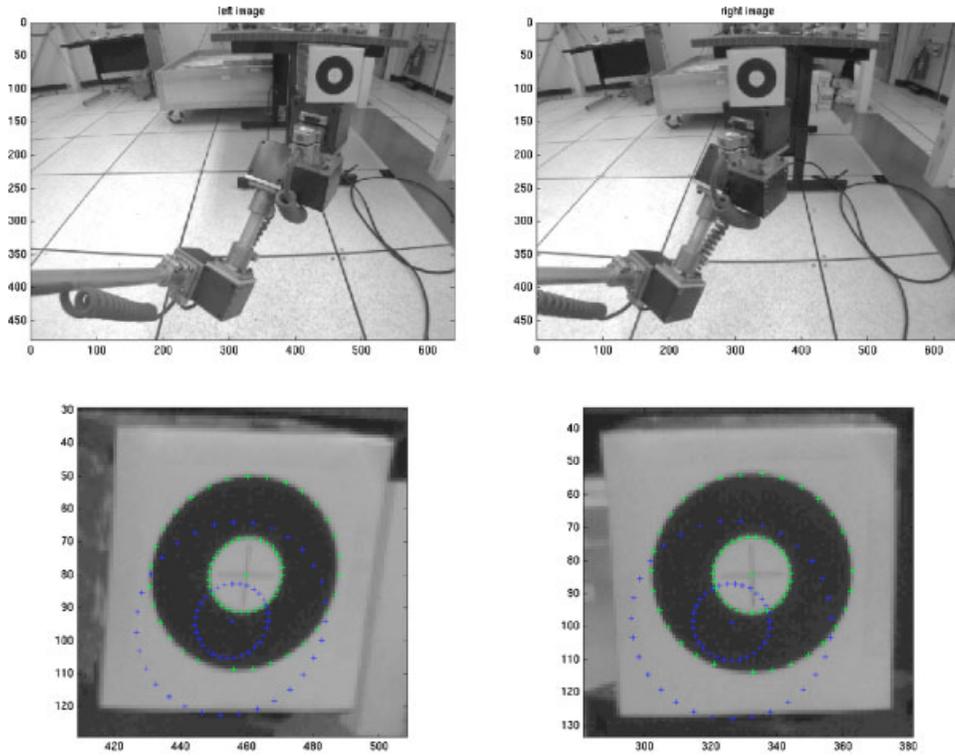


Figure 4. Fiducial prediction (dark+'s) and detection (light+'s) in the left and right images of the mock-up arm.

algorithm is the ability to account for foreshortening. Because the camera lenses are extremely wide field of view, the center of an ellipse fitted to the projected circles would result in several tenths of pixels of localization error, in addition to any matching and fitting error. The approach used also does not introduce an edge bias like many naive edge detection algorithms.

3.3. Stereo Triangulation

With the fiducial located in each of the stereo images, the 3D position of the fiducial is triangulated by finding the point of closest intersection between the two rays projected from the left and right cameras. The pixel at which the fiducial was detected corresponds to a ray, $r_{\{left, right\}}$, passing through the focal point of the camera lens. For a baseline vector b between the two cameras focal points, the distance, $d_{\{left, right\}}$, of the fiducial along each ray can be calculated with

$$\begin{bmatrix} d_{left} & d_{right} \end{bmatrix} \begin{bmatrix} -r_{left} \\ r_{right} \end{bmatrix} + b = 0. \quad (4)$$

The ray gap—the distance between the closest intersection points of the rays—is used as a metric to detect incorrect matches by the fiducial detector.

The camera models are the JPL standard CAHVOR model (Gennery, 2001) and account for the CCD and lens parameters, including distortion. For lab experiments, the cameras were calibrated offline using an unsurveyed calibration technique (Ansar, 2005). The calibration provides the relative position of the two cameras, so their location in the vehicle frame are based only on the vehicle’s CAD model. As a result, this is the largest source of error in the triangulation of the fiducial. For the flight experiments on the MER vehicles, the cameras and arm were precisely calibrated using laser tracker surveying equipment before launch (Maki et al., 2003;

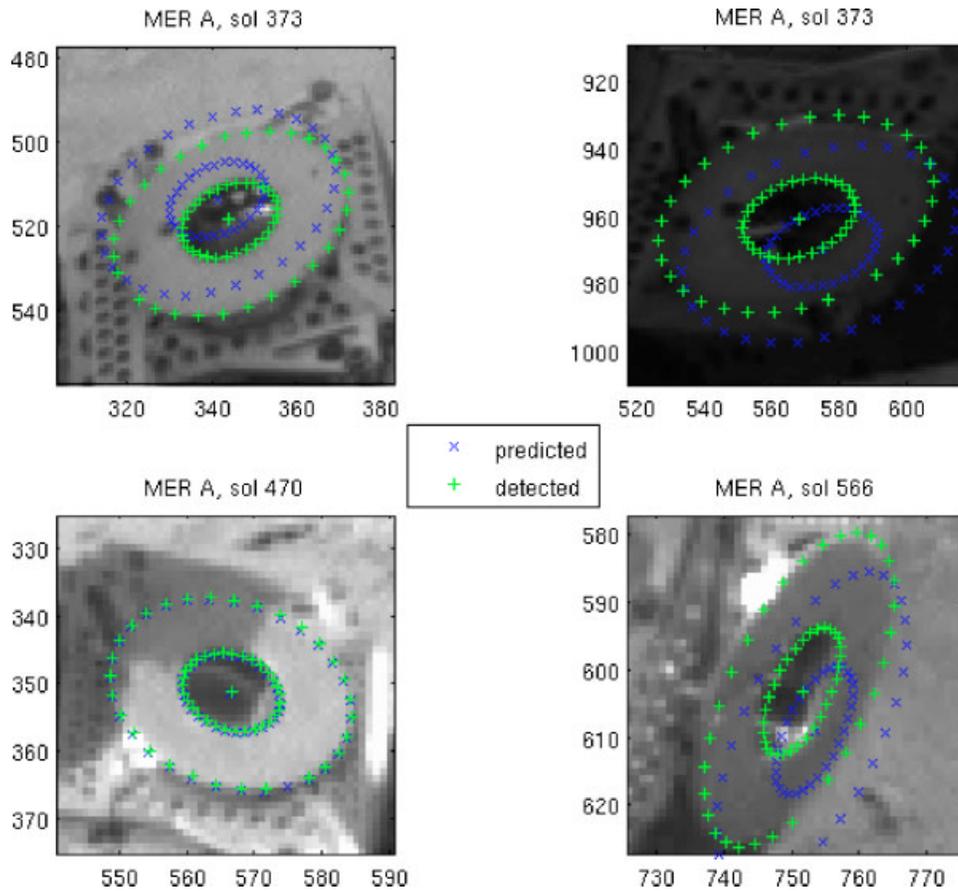


Figure 5. Examples of predicted and correct detections of fiducials on MER data.

Trebi-Ollennu et al., 2005), and the largest source of error contributing to the 1 cm of placement error is not obvious.

3.4. Motion Correction

The difference between the fiducial's position based on the arm's forward kinematics and the triangulation from the stereo cameras provides a correction vector used to compensate for inaccuracies in the arm and camera models during end-effector positioning. The correction vector calculated is subtracted from the commanded position to produce a new command. The correction vector is a function of the fiducial location in the images and the joint angles of the arm, which makes it valid only in a local region. However, with good initial models, placing the end-effector relatively close to the target

using purely kinematics is practical and, consequently, only a single iteration of applying the correction vector is required.

The algorithm can also be used in a closed loop control system that allows the arm to start at any location and updates the correction vector as it approaches the target. In this case, as the end-effector is moved closer to the target location, the correction vector is recomputed. If the fiducial is ever occluded, the previous correction vector can be used and would be valid for small motions.

4. ALGORITHM ANALYSIS

4.1. Algorithm Locality and Sensitivity

The effectiveness of the EPEC algorithm relies on the assumption that arm positioning errors introduced

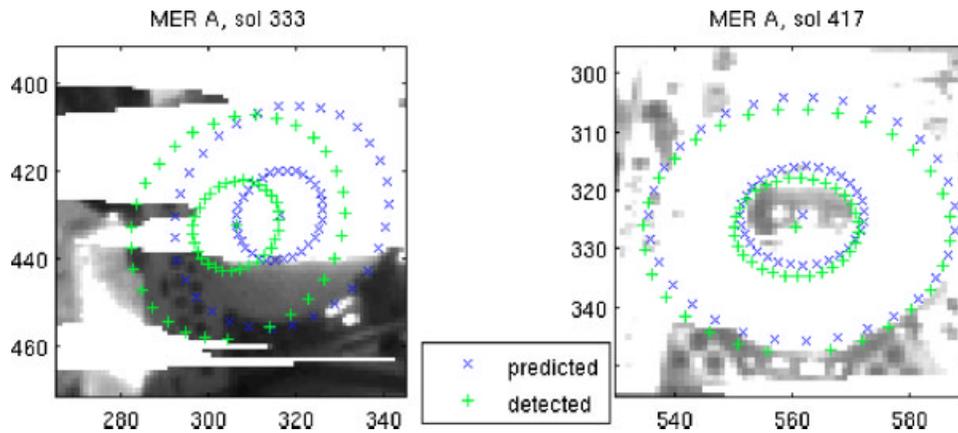


Figure 6. Examples of correct rejections of incorrectly detected fiducials on MER data.

by the camera and arm models are locally constant. This allows a correction vector computed in one location to be applied to a nearby location to compensate for positioning error. The deviation from this assumption and its effect on the algorithm can be shown for most camera/arm systems by using local perturbations of the model parameters.

For instance, an error in a link length of the arm model will result in a correction vector parallel to the direction of that link for the given pose. A small motion θ of a joint before this link will then result in an angle change of the link, resulting in the applicability of the correction vector proportional to $\sin(\theta)$, or θ for small angles. Similarly, a small error ϕ in an arm offset angle will result in a correction vector pointing in the direction tangent to ϕ , and a small change of motion θ of a joint before this joint will result in an angle change of the correction vector proportional to $\sin(\theta)$, or θ for small angles. The effect of an error between the transformation of the arm and camera will result in either a constant translation offset or an error equivalent to an extra arm link with a constant angle error.

For a camera in which a world point is projected to an image and using rays projected from these image plane points by a camera model and stereo triangulation as described in Section 3.3 to determine the world position, the correction vector is simply the difference between the original point and the projected and triangulated point. If a small change is applied to the original world point, the reprojected point will differ by the reprojection of the change, which represents the applicability of the correction vector. That is, the applicability of the correction vector decreases with an increase in the camera re-projection and stereo triangulation error.

Generally, the further the correction vector is computed from where it is applied, the less it will be applicable. The growth of this error is a function of the specific parameter and can result in worse placement than nominal if the correction vector calculated is in the opposite direction of the actual error (for instance, a yaw error on one side of the workspace applied on the other side of the workspace).

4.2. Simulation Analysis

While the locality of error due to small perturbations of independent model parameters is easy to show, the combination of all the errors does not lend itself well to an analytical analysis. As a result, a statistical simulation method was employed to determine the locality of the algorithm and the sensitivity to model parameters by injecting distributions of model errors into the system.

Table I. Fiducial detection rates on MER data.

Data set	Image pairs	Correct detections	False detections
Training data	114	87	4
Testing data	134	113	0

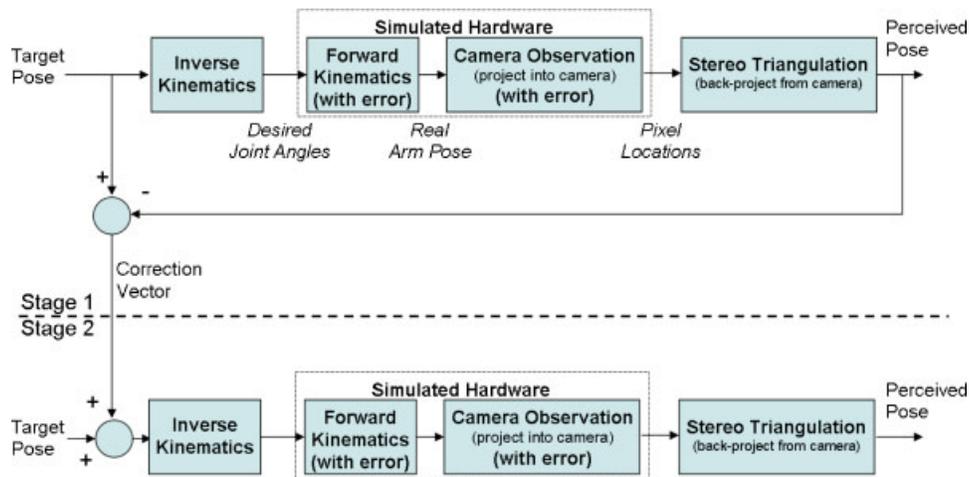


Figure 7. Simulation environment block diagram.

4.2.1. Simulation Environment

To facilitate the experiments, a simulation environment was created where arbitrary errors could be set and the performance of the algorithm measured was created. With the assumption that the closed loop joint control system produces zero steady state error, discrepancies in the final positioning of the arm can be attributed to errors between the actual hardware (arm and cameras) and the associated models. To simulate this, two copies of each model are used: a nominal set to represent the models used in the control software, and a perturbed set with controlled errors to represent real hardware. This simulation environment is illustrated in Figure 7.

Inverse kinematics are performed on a target pose using a nominal arm model, which provides a set of desired joint angles. To simulate the robot moving to these joint angles, forward kinematics is performed on a second arm model which has some amount of error introduced. The result of this process is the actual cartesian pose of the arm which is not known in the real system since there is no ground truth data. The camera observation is another two stage process which involves projecting the 3D arm location into camera pixel space (a 2D pixel location on each camera CCD). This is done first with a camera model that represents the real camera parameters. This location is then projected as one ray from each camera through a second set of camera models, representing imperfectly-calibrated

camera models. The intersection of these two rays (or its closest approximation) is the perceived arm location. The difference between the commanded pose and the perceived pose is the correction vector.

In this setup, a similar set of forward and inverse operations is performed on both sets of models. For validation, the same camera and arm models can be used in each stage of the process and would result in no final positioning error. The observed location of the arm would be exactly where it was commanded to move.

To assess the performance of the EPEC algorithm, the process described above is performed in two stages, as illustrated in Figure 7. In stage 1 of the simulation, the correction vector with a target pose is calculated. In stage 2, a target is designated, to which the correction vector from stage 1 is added, and the process is repeated to calculate the final error. The poses designated in stage 1 and 2 do not need to be the same. The experiments outlined below have a goal of observing the performance degradation as these two poses move further apart.

4.2.2. Parameter Space

The baseline arm and camera models used in the simulation studies mimic the MER configuration. For the stereo cameras, CAHVOR models (Gennery, 2001) were used and each model has 18 possible parameter values. For simplicity, the radial lens distort-

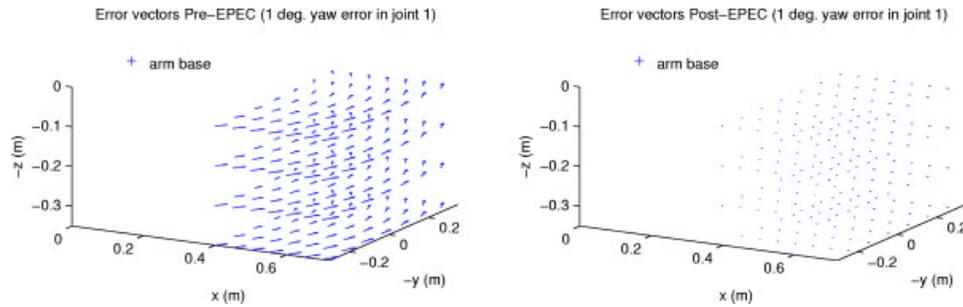


Figure 8. Example parameter analysis plot, error in first yaw joint is shown.

tion parameters (R) were ignored in simulation and the CCD pointing vector was assumed to be aligned with the lens vector ($O=A$), reducing the number of parameters to 12. Although errors in the radial distortion and optical axis parameters would introduce further error in stereo triangulation, as with the other parameters, it is expected to be locally linear. The analysis could be easily extended to account for these parameters, but the results and conclusion are expected to be the same. Similarly, to reduce the parameter space, only the errors between the two cameras (not all of the individual parameters) were considered. Errors were introduced into only one of the two cameras while the other was kept fixed. Again, while introducing error in the parameters for both cameras would degrade the performance further, the error in a single camera is sufficient to show that the effect is locally linear.

For the arm kinematics, the standard Denavit-Hartenberg convention (Denavit & Hartenberg, 1955) was used on the 5 degree of freedom arm with a yaw-pitch-pitch-pitch-yaw (YPPPY) configuration. This produces four parameters for each link for a total of 20. An additional transformation was used to relate the fiducial to the end effector. No errors were introduced into this transform.

With this set of parameters defined, how each parameter individually affects the total positioning error and how they are compensated with the EPEC algorithm can be examined to verify the analysis. Furthermore, it can be used to identify key parameters which may require additional consideration during the calibration process. A representative plot of one parameter of this analysis is shown in Figure 8. On the left is a plot of correction vectors caused by error in the first yaw angle parameter. All errors ra-

diate perpendicular to the arm direction and grow as the distance from the first yaw joint increases. This is to be expected with a rotational error. The error magnitude from this parameter error is manageable and sufficiently compensated with the EPEC algorithm. The right side of Figure 8 shows positioning errors when compensated with EPEC. Performing the same procedure on the other 31 parameters produced similar results.

To perform a full analysis of this large number of parameters, a statistical approach was employed. Three levels of error were chosen for each of the parameter types (lengths, angles, etc.) and these levels were used as the standard deviation in a random distribution. Arm parameter errors and extrinsic camera parameter errors were chosen to mimic the expected errors of a surveyed calibration of the arm and cameras (assuming various surveying equipment, with accuracy of approximately 0.5–2 mm). Intrinsic camera errors were exaggerated in these experiments to gauge how well the algorithm would do with large errors and are clearly the largest source of error. The arm and camera parameter errors were analyzed both independently and jointly, resulting in three groups (arm only, camera only, and arm and camera). Table II outlines these three groups with the standard deviations for their three levels of intensity. The three groups at three levels of error then results in nine parameter groups, each with 100 members whose values are a random distribution which adheres to the standard deviations in Table II.

The intent of this analysis is to observe the combined influence of all parameter errors in a combined manner on the arm motion and EPEC correction. For each of the tests outlined below, all 100

Table II. Parameter set error magnitudes.

Set name	Link length error	Joint angle error	Camera position error	Camera rotation error	Focal length error	Pixel center error
Arm 1	1.0 mm	0.5 deg	X	X	X	X
Arm 2	1.0 mm	1.0 deg	X	X	X	X
Arm 3	2.5 mm	1.0 deg	X	X	X	X
Camera 1	X	X	1.0 mm	0.05 deg	1 pixel	5 pixels
Camera 2	X	X	1.0 mm	0.15 deg	2.5 pixel	12.5 pixels
Camera 3	X	X	1.0 mm	0.30 deg	5 pixel	25 pixels
Combined 1	1.0 mm	0.5 deg	1.0 mm	0.05 deg	1 pixel	5 pixels
Combined 2	1.0 mm	1.0 deg	1.0 mm	0.15 deg	2.5 pixel	12.5 pixels
Combined 3	2.5 mm	1.0 deg	1.0 mm	0.30 deg	5 pixel	25 pixels

parameter sets in each group are tested and the resulting correction vector magnitudes are averaged.

4.2.3. Simulated Algorithm Performance

The first goal of the simulation analysis was to determine the effectiveness of the EPEC algorithm. To this effect, the first test was designed to test how the algorithm would perform when stereo camera images of the arm at the target location are available. In this test, the arm was commanded to a target and the initial error was calculated. The error corresponds to the correction vector at that target. To implement EPEC, the arm was commanded again, this time with the correction vector added, and the final positioning error was calculated. This case, in which the same target pose is used in both stages, represents

the best performance possible with only one iteration. Table III outlines the results of this test on the nine elements of the parameter space.

The results indicate that even with significant error in the models (several millimeters in link lengths and camera position, a degree in joint offsets, 0.3 degrees in camera rotation, and several pixels in focal length and image center), 2 mm positioning accuracy can be achieved. With reasonable parameter errors, sub-millimeter accuracy can be easily achieved.

4.2.4. Simulated Algorithm Locality

The second goal of this analysis was to determine the effectiveness of a correction vector when it is applied away from the place where it was calculated.

Table III. Error magnitudes for EPEC at target.

Set name	Uncorrected mean error	Uncorrected std.dev.	Corrected mean error	Corrected std. dev.
Arm 1	12.2 mm	0.7 mm	0.3 mm	0.01 mm
Arm 2	23.0 mm	1.2 mm	0.9 mm	0.05 mm
Arm 3	25.7 mm	1.5 mm	1.2 mm	0.07 mm
Camera 1	3.3 mm	2.0 mm	0.06 mm	0.07 mm
Camera 2	8.4 mm	5.4 mm	0.40 mm	0.45 mm
Camera 3	15.9 mm	9.7 mm	1.50 mm	1.70 mm
Combined 1	4.9 mm	1.0 mm	0.09 mm	0.04 mm
Combined 2	10.8 mm	3.2 mm	0.50 mm	0.35 mm
Combined 3	19.0 mm	9.4 mm	1.90 mm	1.80 mm

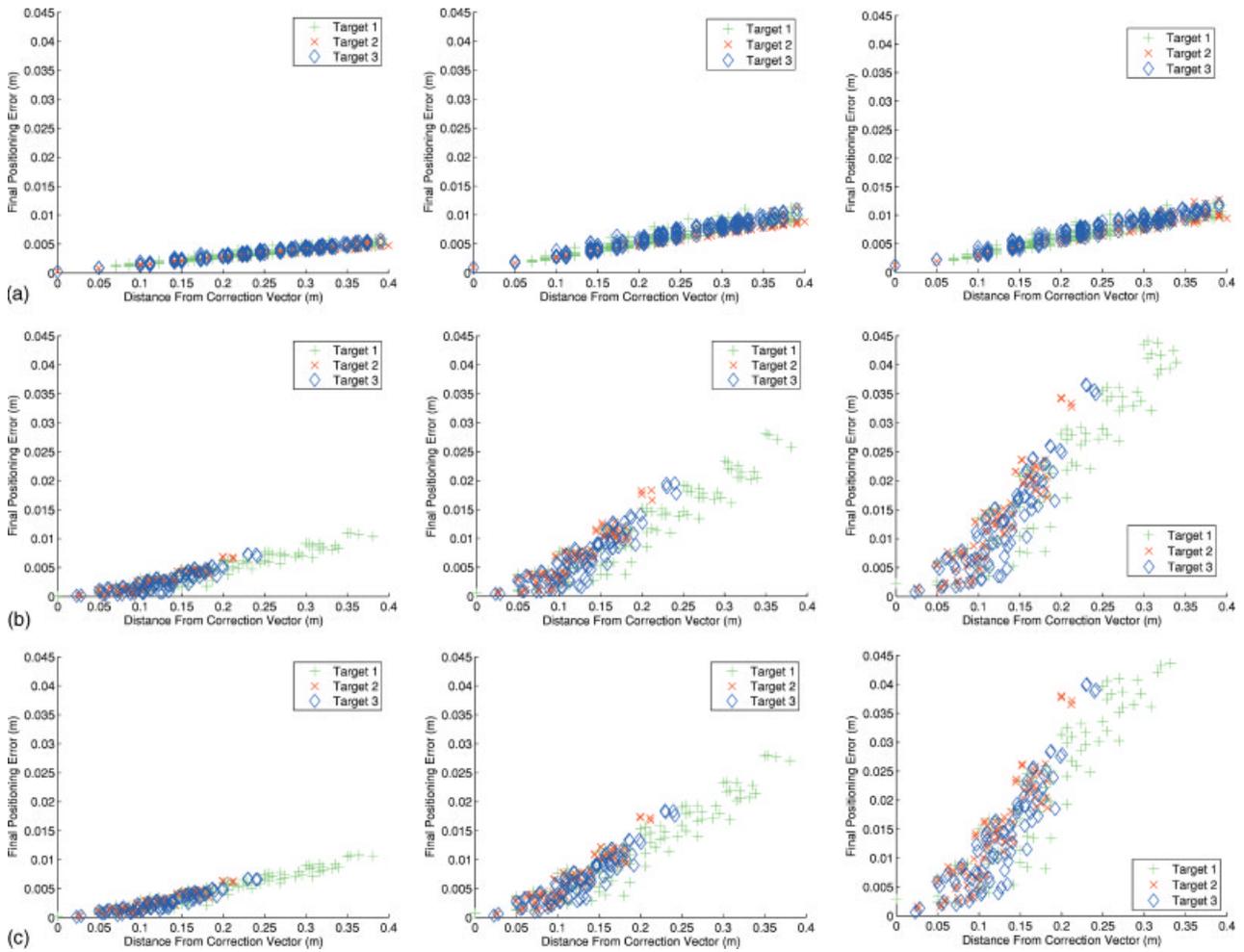


Figure 9. EPEC locality for the parameter error sets in Table II.

To test this, a correction vector was calculated at one location and then applied to targets throughout the workspace. Of interest now is the final positioning error at a target along with the distance between the target and the place where the correction vector was calculated.

For each of the nine parameter sets, three starting locations were selected to calculate the correction vector. This vector was then applied to each of the targets throughout the workspace. The mean error was calculated at each target and plotted versus distance from the location where the correction vector was calculated. The result of this test for the parameter sets is shown in Figure 9. The three markers in this plot represent the three targets: target 1 in the

center of the arm workspace, target 3 in the center of the camera field of view, and target 2 on the extreme edge of the workspace.

Figure 9(a) illustrates the three cases of arm kinematic errors only. In all three cases, the final positioning error grows linearly with distance from the correction vector location. The three sets in this group represent three levels of increasing error and likewise the three trend lines on these sets have increasing slope. As errors in the arm model increase, the area where a correction vector can produce a desired level of positioning performance will decrease linearly.

For the case of only camera errors, the results are shown in Figure 9(b). In these cases, there is a qua-

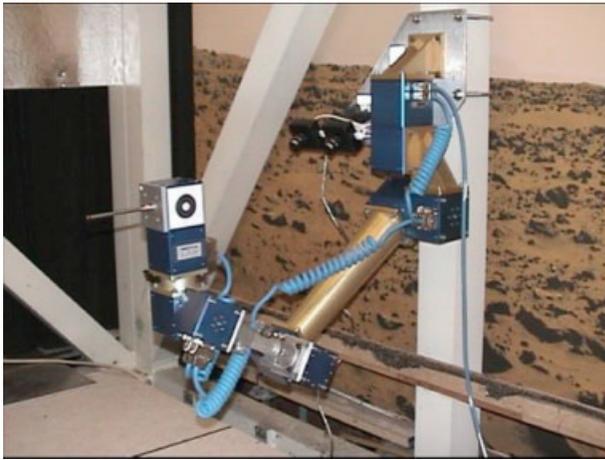


Figure 10. MER mock-up arm and camera system used for experiments.

dratic increase in the positioning error as the correction vector is applied away from where it was calculated. As expected, again the slope of this quadratic trend increases as the error introduced into the camera models is increased.

Figure 9(c) shows the results of the three combined error sets. For the error magnitudes chosen, the camera model errors dominate the algorithm performance. The magnitude of only the camera errors is substantially higher than the kinematic errors. Combining the two produces a locality plot that is only slightly worse than camera errors only.

The trends in the locality plots shows that even for models with relatively larger errors (resulting in a mean or approximately 2.5 cm kinematic placement error), a correction vector calculated at 10 cm away from the final end-effector position will result in a mean of at worst 1 cm placement accuracy.

5. EXPERIMENTAL RESULTS

5.1. MER Mock-up Experiments

5.1.1. Experimental Setup

To validate the analytical and simulated algorithm performance and sensitivity, EPEC was tested on a hardware arm/camera system representing the MER vehicle. The test arm, shown in Figure 10, is a wall

mounted 5 degree of freedom manipulator with identical configuration and similar sizing to the MER IDD (approximately 80 cm at full extension). The cameras are rigidly mounted to the robot's back plate, have a 10 cm baseline, and are angled at 30 deg down. They have 2.8 mm lenses, a 640×480 pixel CCD with $4.65 \mu\text{m}$ pixel size and a field of view of about 90 deg. For this configuration, at 1 m away, a 1 pixel error in the image corresponds to approximately 1 mm of lateral error in the workspace. Similarly, a 0.3 pixel error in stereo disparity matching results in 10 mm of range error. However, because the algorithm uses a target designated in the same stereo pair that it uses to compute the correction vector, it can achieve much better placement performance in practice. Unlike the MER vehicles, this system was not calibrated to high fidelity. The arm model and location of the cameras is based on CAD drawings and the cameras were calibrated using an unsurveyed technique (Ansar, 2005).

5.1.2. Correction Vector Measurements

The first experiment performed was simply a measurement of correction vectors throughout the workspace. The arm was commanded to 100 target points evenly spaced at 5 cm throughout the workspace. At each target, the arm location was calculated through kinematics and through vision and their difference, the correction vector, is plotted in Figure 11. The vectors are anchored at their location calculated by vision and they point towards the commanded target, showing the direction of the correction. The camera fields of view and approximate arm workspace are superimposed with the vector field in the left figure to show the vector field's spread throughout the usable space.

This experiment serves as a baseline for positioning performance and also gives a visualization of the modeling error through the workspace. The mean size of these 100 correction vectors is 1.5 cm. A comparison of these results to the individual parameter error measurements discussed in Section 4.2.2 may reveal particular parameters of concern. For instance, if all correction vectors radiate inwards to the camera centers (as these do) there is most likely an error in a parameter that affects stereo ranging, such as the baseline measurement.

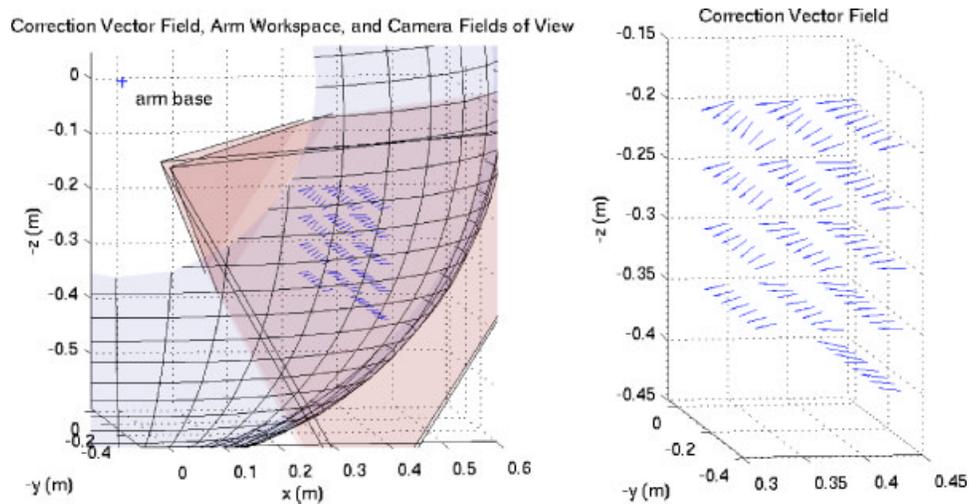


Figure 11. Experimentally measured correction vector field with and without camera fields of view and the approximate arm workspace superimposed.

5.1.3. Correction Vector Locality

The second experiment performed was an exact copy of the simulated locality experiment described in Section 4.2.3. The arm was commanded to three different start positions and a correction vector is calculated associated with that point. This correction vector is then applied to each of a large number of targets spread throughout the workspace. Using a laser rangefinder total station (Leica Geosystems model TCRA1103+), the target location is measured along with position of the arm before and after the application of the correction vectors. The final positioning error (distance from the target location) is plotted against distance from the three correction vectors. Figure 12 shows the result.

Unfortunately, because the ground truthing system has a ± 2 mm error, there is a wide spread on all the data collected and the actual accuracy that can theoretically be achieved is difficult to measure. As expected, there is no relationship between distance to target and positioning error when EPEC is not used. The mean of these targets is also identical to the 1.5 cm calculated in the previous experiment. With EPEC, however, there is a clear trend in better performance for targets close to the correction vector. An exact fit with noisy data would require more data points, but a simple linear fit gives a good idea of EPEC performance as both a linear and quadratic fit would be very close at the small scales (10 cm or

less) in question. As with simulation results, the EPEC algorithm is effective in compensating for system with significant model error (such as this prototype arm).

5.2. Experiments on MER

In order to demonstrate the effectiveness and feasibility of using EPEC to improve placement accuracy of a flight system manipulator, a set of “ground-in-the-loop” experiments were accommodated by the MER mission and performed with the MER B (Opportunity) vehicle on Mars. For the experiments, the rover arm was commanded to several 3D locations throughout the workspace and imaged. These images along with the corresponding arm joint angles were then downlinked and the EPEC algorithm run on each set. The output corrected arm commands were then uplinked to the rover and executed, and images acquired of the new arm positions.

The target positions are specified as 3D locations in the camera frame of reference. Inverse kinematics with gravity compensation is then used to calculate the joint angles to which the arm joints were servoed. After servoing to the position, the joint encoders were read to provide the actual joint angles of the arm and stereo images acquired as input for the EPEC algorithm. This data was then processed on the ground by the EPEC algorithm to detect the ac-

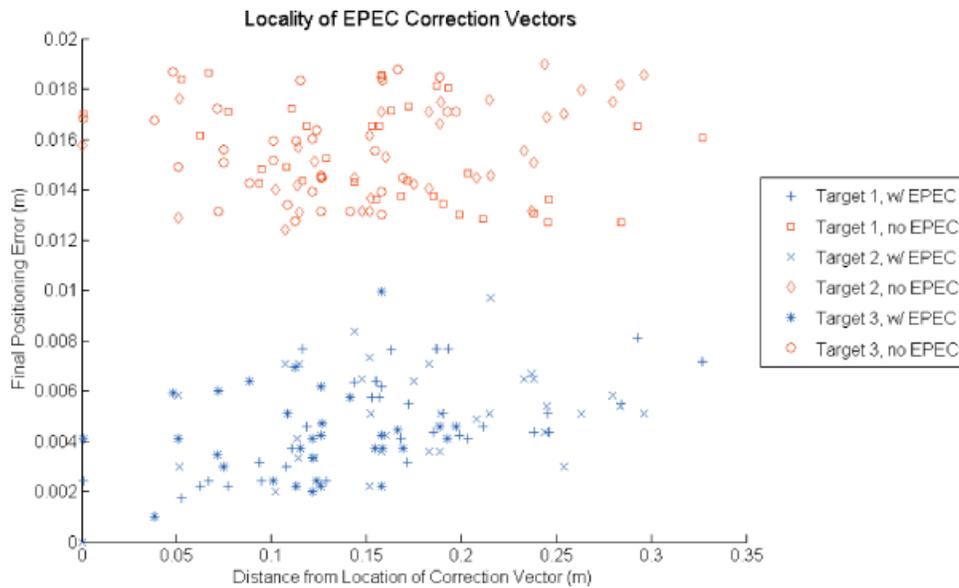


Figure 12. Locality experiment results, with and without EPEC.

tual end-effector position in the camera frame of reference and provide a corrected end-effector position. This position was then fed through the same inverse kinematics to determine the joint angles, which were then uplinked to the rover. As previously, the arm was servoed to these joint angles and joint angles and images downlinked. To determine the effectiveness of the correction, the end-effector was detected in the final images and compared to the originally designated locations (both in the image plane and in 3D with stereo triangulation). While redetecting the fiducial in the final image does not provide a ground-truth comparison (any bias in the fiducial detector would be present in both the initial and final images), it does provide an indication of how well the end-effector can be visually replaced to a designated location.

The difference between the designated locations and the uncorrected locations as detected in the image for the “ground-in-the-loop” experiments were approximately 1 cm, resulting in a corresponding correction vector magnitude. The difference between the designated and corrected locations of two of the three tests is less than 1 mm; the difference between the last test is about the same as the uncorrected due to a failure to accurately detect the fiducial in the final image. Although the data set is small, it corre-

sponds to what is expected based on previous analytical and experimental results. The fiducial detection results can be seen in Figures 13–15 and a summary of the correction vectors and final localization errors can be seen in Table IV.

While “ground-in-the-loop” experiments are difficult to perform due to their overhead, the EPEC algorithm has been run on all of the MER data collected so far in the mission where the end-effector fiducial (Mossbauer contact plate) is visible. Monitoring the correction vector over the mission duration provides insight into the performance of the arm and camera models over time. Figure 16 shows the correction vector of the two MER vehicles for each sol (Martian day) of the mission (where the fiducial was visible) using the camera models that were generated using a surveyed calibration during vehicle assembly and testing (ATLO) before launch. The mean error shows an overall camera and arm model with the variation resulting mainly from the different locations in the workspace.

The algorithm run on the ground was exactly the same as the one that was used online during previous experiments, but was run on a standard desktop machine rather than an MER testbed system. Because the computation costs are small com-

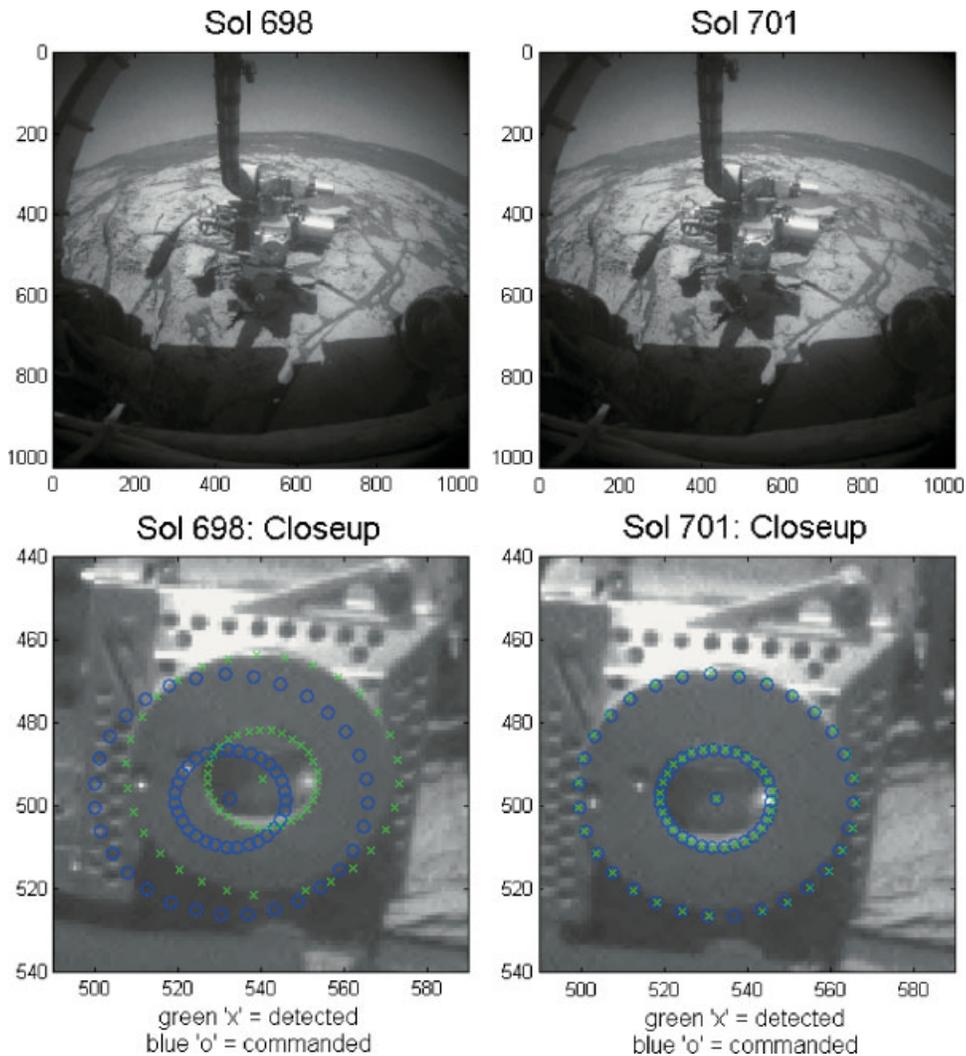


Figure 13. Application of EPEC correction on the MER B vehicle, showing the commanded (blue circles) and detected (green x's) fiducial locations before (sol 698) and after (sol 701) correction for test position 1.

pared to other rover operations, little effort was made to optimize the code or profile its performance on the actual MER processor.

6. CURRENT AND FUTURE WORK

Currently the EPEC algorithm is being compared to and integrated with other techniques that could be used on a flight system. This includes using HIPS to periodically generate an initial camera model and then use EPEC during the actual placement to im-

prove final accuracy. A DH-Tune algorithm (Nickels, 2003) is also being investigated for improving the arm model in a similar way as HIPS updates the camera model. Combining EPEC with an *in situ* calibration technique would benefit from the better kinematic placement and image projection, resulting in a more accurate fiducial detection, and consequently, improved compensation.

The algorithm has also been integrated in the MER set of operations tools for downlink analysis to monitor the performance of the vehicles' arms. Any new stereo image pair that views the Mossbauer con-

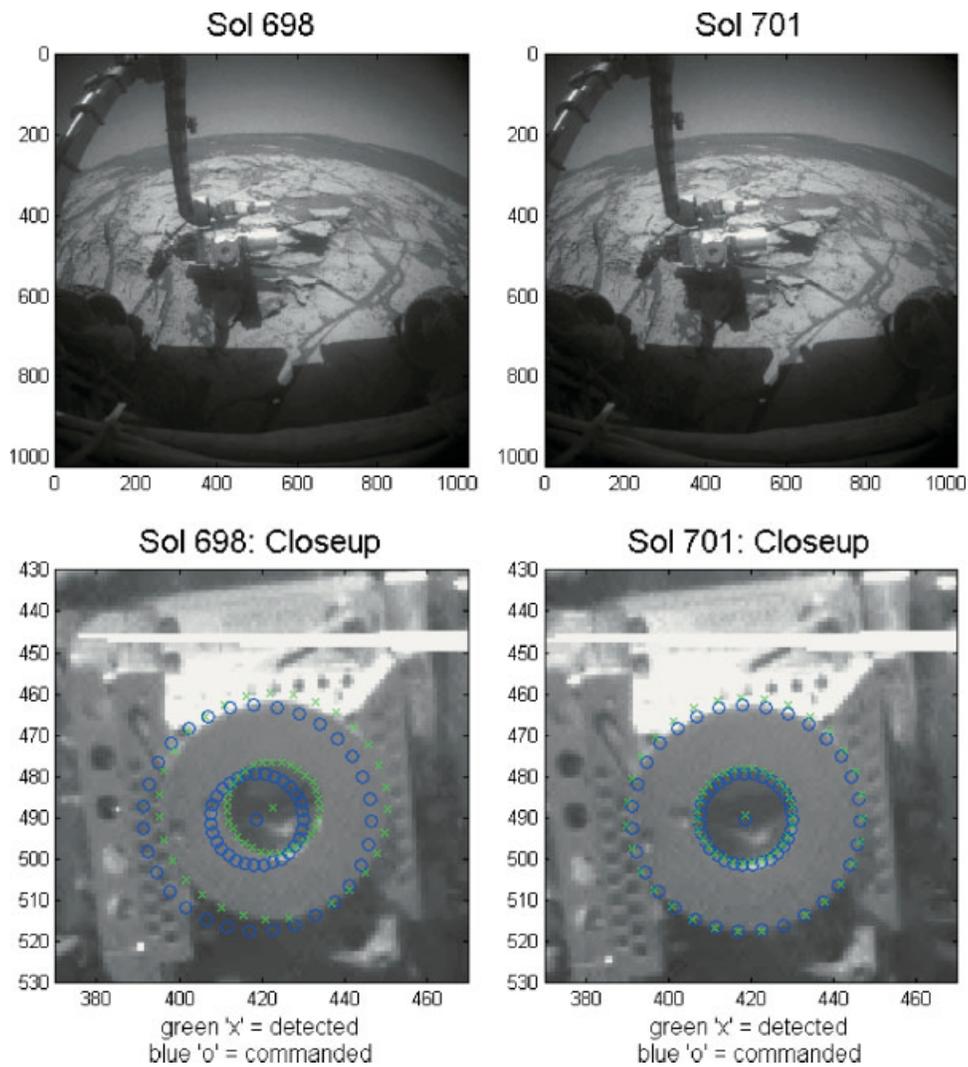


Figure 14. Application of EPEC correction on the MER B vehicle, showing the commanded (blue circles) and detected (green x's) fiducial locations before (sol 698) and after (sol 701) correction for test position 3.

tact ring is provided to the offline tool, which then detects the ring and outputs a correction vector. This correction vector is then added to the plot of all correction vectors over the mission lifetime for offline viewing.

Finally, the algorithm has been integrated into a complete end-to-end system that demonstrates “single cycle instrument placement” for a planetary rover (Backes, Diaz-Calderon, Robinson, Bajracharya, & Helnick, 2005; Bajracharya, Diaz-Calderon, Robinson, & Powell, 2005; Huntsberger, Cheng, Stoupe, & Aghazarian, 2005; Pedersen et al., 2003). In this sce-

nario, the rover takes an image of a scene and a scientist selects a target on which the arm’s end-effector instrument will be placed. The rover then autonomously approaches the target, places itself so that the target is in the arm workspace, deploys the arm and places the instrument on the target (Figure 17).

The next step in the development of the algorithm itself is to account for the complete end-effector pose, including its orientation. Currently, determining orientation from the MER fiducial is extremely difficult due to its small size and the lack of other fiducials on the end-effector. While the center of the fi-

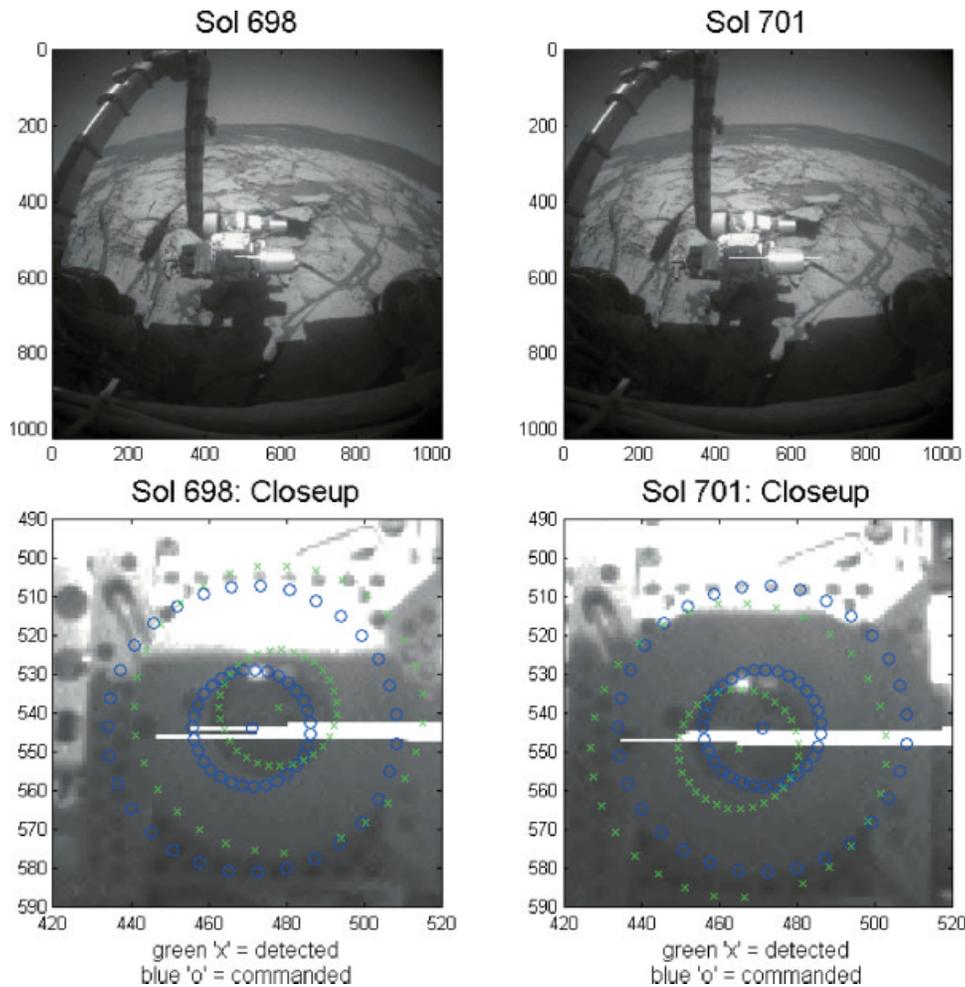


Figure 15. Application of EPEC correction on the MER B vehicle, showing the failure of fiducial detection (green x's) on sol 701 for test position 2, which prevents determining the final positioning accuracy.

fiducial is easy to detect, the orientation, which is a function of the detected shape, would be very sensitive to small errors in shape and, consequently, difficult to localize precisely. Furthermore, the most important orientation is the end-effector yaw, which is the most sensitive even if a more appropriate single fiducial could be created and used. If multiple fiducials were available, this would simplify the problem by allowing orientation to be determined from stereo triangulation. While triangulation error would be the dominate cause of absolute orientation error, if the same EPEC concept used for positioning is used to position multiple fiducials on the end-effector (resulting in orientation control), the triangulation error would be less of an issue.

Lastly, in order to have the EPEC algorithm implemented for and used on the 2009 Mars Sample Laboratory (MSL) mission, the algorithm will be tested on a mockup of this system. While the arm and camera configuration is similar to MER (a 5DOF, YPPY arm with a wide field of view stereo camera pair viewing the workspace), it is a significantly larger rover and the arms extent is approximately 2 m and the camera baseline is approximately 20 cm. Intuitively, the EPEC algorithm and the analysis presented should scale for any arm configuration and stereo pair location, but experimentation on various configurations could be used to verify this. For use on board a future mission vehicle, checks on the correction vector would no doubt need to be implemented

Table IV. Positioning error with EPEC on Opportunity IDD, sols 698 and 701.

Trial	Error before correction	Correction vector (x, y, z) (m)	Error after correction
Position 1	6.9 mm	(-.00499736, -0.00470137, -0.00100508)	0.50 mm
Position 2	6.8 mm	(-.00523693, -0.00310702, -0.0030674)	7.2 mm
Position 3	11.9 mm	(-.00956444, -0.000652192, -0.00700548)	0.39 mm

to prevent incorrectly detected fiducials from commanding a large, incorrect motion. Although a simple magnitude check might be sufficient, comparing cor-

rection vector determined online to vectors previously calculated in the same vicinity would provide a good indication of incorrect corrections.

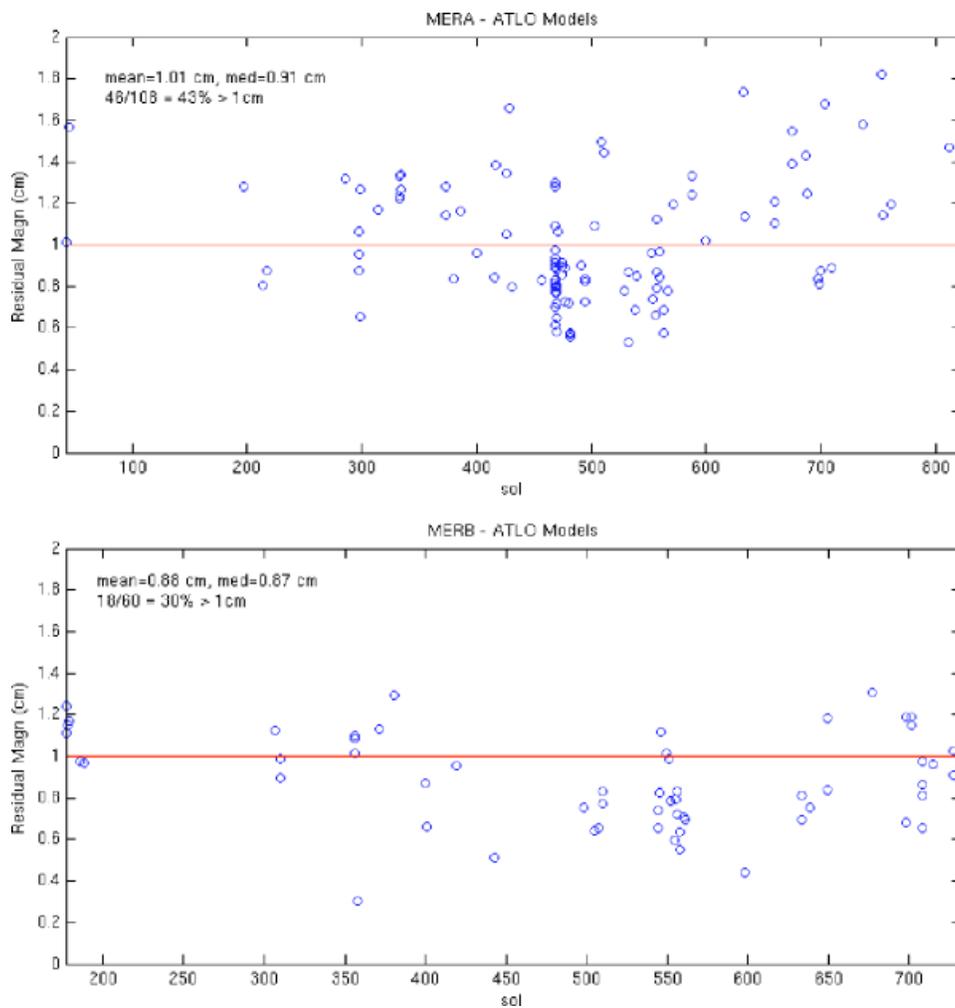


Figure 16. Correction vectors (the difference between the fiducial position based on kinematics vs vision) for the two MER vehicles.



Figure 17. Rocky 8 research rover completing a single cycle instrument placement.

7. CONCLUSION

All foreseeable future landed Mars missions and many other space robotics applications will need to precisely place an arm end-effector on a target that is designated in a camera image. However, the imaging and computation constraints of space applications are generally too prohibitive for standard visual servoing methods.

The EPEC algorithm provides a simple but effective method for compensating for end-effector positioning error. Because of its simplicity, the algorithm is not only computationally efficient, but also easy to implement, test, and validate. It is also easy to analyze and predict performance and can be run with the “ground-in-the-loop,” making it a low risk technology for flight missions. Its analysis, simulation, and flight experiments show that very accurate placement can be achieved if the end-effector is imaged near the designated position and final position error grows approximately linearly with the distance it is imaged from the target. It also shows that the algorithm degrades gracefully with decreasing arm and camera model accuracy.

The fiducial detection algorithm developed to facilitate the EPEC algorithm is robust to lighting change and viewpoint and accurate to less than a pixel. It makes use of relatively well-calibrated arm and camera models and a pyramidal resolution scheme to improve its computational efficiency. Using various metrics for rejection of false matches, it

achieves less than a 1% false positive rate on all of the existing MER images of the Mossbauer contact ring.

The EPEC algorithm has been analyzed in simulation and on prototype arm and camera systems, and has been tested with “ground-in-the-loop” experiments on the MER vehicles. It is currently being used by tasks in the Mars Technology Program to implement the placement of an arm end-effector on research rover using a single command, and has the potential to be used on future landed Mars missions to enable mission critical capabilities such as instrument placement and sample acquisition and handling.

ACKNOWLEDGMENTS

The research described in this publication was carried out at the Jet Propulsion Laboratory, California Institute of Technology under contract from the National Aeronautics and Space Administration (NASA), with funding from the Mars Technology Program, NASA Science Mission Directorate.

REFERENCES

- Allen, P. K., Timcenko, A., Yoshimi, B., & Michelman, P. (1992). Real-time visual servoing. In *Proceedings 1992 International Conference on Robotics and Automation*, pp. 1850–1856.
- Ansar, A. (2005). Survey free camera calibration. (JPL Technical Report, NTR 42336).
- Backes, P., Diaz-Calderon, A., Robinson, M., Bajracharya, M., & Helmick, D. (2005). Automated rover positioning and instrument placement. In *IEEE Aerospace Conference Proceedings*, pp. 60–71.
- Bajracharya, M., Diaz-Calderon, A., Robinson, M., & Powell, M. (2005). Target tracking, approach and camera handoff for automated instrument placement. In *IEEE Aerospace Conference Proceedings*, pp. 52–59.
- Baumgartner, E. T., Bonitz, R. G., Melko, J. P., Shiraishi, L. R., & Leger, C. P. (2005). The mars exploration rover instrument positioning system. In *IEEE Aerospace Conference Proceedings*, pp. 1–19.
- Baumgartner, E. T., & Schenker, P. S. (1996). Autonomous image-plane robot control for martian lander operations. In *Proceedings of the IEEE International Conference on Robotics and Automation*, Volume 1, pp. 726–731.
- Bonitz, R. G., Slostad, J., Bon, B., Braun, D., Brill, R., Buck, C., et al. (2001). Mars volatile and climate surveyor robotic arm. *Journal of Geophysical Research* 106(E8), 17623–17634.
- Denavit, J., & Hartenberg, R. S. (1955). A kinematic notation for lower-pair mechanisms based on matrices.

- ASME Journal of Applied Mechanics 77(2), 215–221.
- Gennery, D. B. (2001). Least-squares camera calibration including lens distortion and automatic edition of calibration points. In Grun, A. and Huang, T., editors, *Calibration and Orientation of Cameras in Computer Vision*, pp. 123–136. New York: Springer-Verlag.
- Huntsberger, T., Cheng, Y., Stroupe, A., & Aghazarian, H. (2005). Closed loop control for autonomous approach and placement of science instruments by planetary rovers. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 3783–3790.
- Hutchinson, S., Hager, G., & Corke, P. (1996). A tutorial on visual servo control. *IEEE Transactions on Robotics and Automation* 12(5), 651–670.
- Keller, H. U., Hartwig, H., Kramm, R., Koschnyn, D., Markiewicz, W. J., Thomas, N., et al. (2001). The MVACS robotic arm camera. *Journal of Geophysical Research* 106(E8), 17609–17622.
- Lewis, J. P. (1995). Fast normalized cross-correlation. In *Vision Interface*, pp. 120–123. Montreal, Canada: Canadian Image Processing and Pattern Recognition Society.
- Maki, J. N., Bell, J.F., III, Herkenhoff, K. E., Squyres, S. W., Kiely, A., Klimesh, M., et al. (2003). Mars Exploration Rover engineering cameras. *Journal of Geophysical Research* 108(E12), 8071.
- Malis, E., Chaumette, F., & Boudet, S. (1999). 2-1/2-d visual servoing. *IEEE Journal of Robotics and Automation* 15, 238–250.
- Nelson, B., Papanikolopoulos, N. P., & Khosla, P. (1996). Robotic visual servoing and robotic assembly tasks. *IEEE Robotics & Automation Magazine* 3(2), 23–31.
- Nickels, K. (2003). Hand-eye calibration for robonaut. NASA Summer Faculty Fellowship Program Final Report, Johnson Space Center.
- Pedersen, L., Sargent, R., Bualat, M., Deans, M., Kunz, C., Lee, S., & Wright, A. (2003). Single cycle instrument deployment. In *International Symposium on Artificial Intelligence, Robotics and Automation in Space (iSAIRAS)*.
- Reeves, G. E., & Snyder, J. F. (2005). An overview of the Mars Exploration Rovers' flight software. In *2005 IEEE International Conference on Systems, Man, and Cybernetics*, Volume 1, pp. 1–7.
- Savage, D., & Cook-Anderson, G. (2004). NASA selects investigations for the Mars Science Laboratory. Jet Propulsion Laboratory. Press Release 04-398.
- Seelinger, M., Yoder, J.-D., Baumgartner, E. T., & Skaar, S. B. (2002). High-precision visual control of mobile manipulators. *IEEE Transactions on Robotics and Automation* 18(6), 957–965.
- Shi, J., & Tomasi, C. (1994). Good features to track. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR'94)*, pp. 593–600.
- Skaar, S. B., Brockman, W. H., & Hanson, R. (1987). Camera space manipulation. *International Journal of Robotics Research* 6(4), 20–32.
- Smith, C., Brandt, S., & Papanikolopoulos, N. P. (1997). Eye-in-hand robotics tasks in uncalibrated environments. *IEEE Journal of Robotics and Automation* 13(6), 903–914.
- Smith, P. (2004). The Phoenix mission to Mars. In *Proceedings 2004 Aerospace Conference*, Vol. 1, p. 342.
- Trebi-Ollennu, A., Baumgartner, E. T., Leger, P. C., & Bonitz, R. G. (2005). Robotic arm in-situ operation for the Mars Exploration Rovers surface mission. In *2005 IEEE International Conference on Systems, Man, and Cybernetics*, Volume 2, 1799–1806.
- Tunstel, E., Maimone, M., Trebi-Ollennu, A., Yen, J., Petras, R., & Willson, R. (2005). Mars exploration rover mobility and robotic arm operational performance. In *2005 IEEE International Conference on Systems, Man, and Cybernetics*, pp. 1807–1814.