Two Years of Visual Odometry on the Mars Exploration Rovers

Mark Maimone, Yang Cheng, and Larry Matthies
Jet Propulsion Laboratory
California Institute of Technology
Pasadena, CA USA
mark.maimone@jpl.nasa.gov

Abstract

NASA's two Mars Exploration Rovers (MER) have successfully demonstrated a robotic Visual Odometry capability on another world for the first time. This provides each rover with accurate knowledge of its position, which allows it to autonomously detect and compensate for any unforeseen slip encountered during a drive. It has enabled the rovers to drive safely and more effectively in highly-sloped and sandy terrains, and has resulted in increased mission science return by reducing the number of days required to drive into interesting areas.

The MER Visual Odometry system comprises onboard software for comparing stereo pairs taken by the pointable mast-mounted 45 degree FOV Navigation cameras (NAVCAMs). The system computes an update to the 6 Degree Of Freedom rover pose (x, y, z, roll, pitch, yaw) by tracking the motion of autonomously-selected terrain features between two pairs of 256x256 stereo images. It has demonstrated good performance with high rates of successful convergence (97% on Spirit, 95% on Opportunity), successfully detected slip ratios as high as 125%, and measured changes as small as 2 mm, even while driving on slopes as high as 31 degrees.

During the first two years of operations, Visual Odometry evolved from an "extra credit" capability into a critical vehicle safety system. In this paper we describe our Visual Odometry algorithm, discuss several driving strategies that rely on it (including Slip Checks, Keep-out Zones, and Wheel Dragging), and summarize its results from the first two years of operations on Mars.

1 Background

Keeping track of a vehicle's location is one of the most challenging aspects of planetary rover operations. NASA's Mars Exploration Rovers (MERs) typically have been commanded only once per Martian solar day (or "sol") using a pre-scheduled sequence of precise metrically specified commands (e.g., "drive forward 2.34 meters, turn in place 0.3567 radians to the right, drive to location X,Y, take color pictures of the terrain at location X,Y,Z" (Biesiadecki et al., 2005)), so having an accurate position estimate onboard during the execution of *all* terrain-based commands has been of critical importance.

The design goal for MER was to maintain a position estimate that drifted no more than 10% during a 100 meter drive. MER rover onboard position and attitude estimates were updated at 8 Hz nearly every time the wheels or rover arm (Instrument Deployment Device, or IDD) were actuated. Changes in attitude (roll, pitch, yaw) were measured using a Litton LN-200 Inertial Measurement Unit (IMU) that has 3-axis accelerometers and 3-axis angular rate sensors, and changes in position were estimated by combining attitude measurements with encoder readings of how much the wheels turned (wheel odometry). Position estimates derived solely from those sensors easily achieved the desired accuracy in benign terrains (Li et al., 2005), but not on steep slopes or sandy terrain.

After moving a small amount on a slippery surface, the rovers were often commanded to use camera-based *Visual Odometry* to correct any errors in the initial wheel odometry-based estimate that occur when the wheels lose traction on large rocks and steep slopes. Our Visual Odometry system computes an update to the 6-DOF rover pose (x, y, z, roll, pitch, yaw) by tracking the motion of "interesting" terrain features between two pairs of stereo images in both 2D pixel coordinates and 3D world coordinates. A maximum likelihood estimator applied to the computed 3D offsets produces the final motion estimate. However, if any internal consistency check fails, too few feature points are tracked, or the estimation fails to converge, then no motion estimate update will be produced and the initial estimate (nominally based on wheel odometry and the IMU) will be maintained.

NASA's twin Mars Exploration Rovers Spirit and Opportunity landed on the surface of Mars in January 2004. As shown in the blue lines of the traverse plots in Figures 14 and 15, human rover drivers have commanded extensive use of the Visual Odometry software, especially during high-tilt operations: driving Opportunity inside Eagle and Endurance craters, and climbing Spirit through the Columbia Hills. Visual Odometry was not used on every drive step, however. Initially the reason was that the operations team was unfamiliar with the capability; this was an "extra credit" capability not originally baselined for the mission, and therefore had not been included in operational readiness tests (although it did go through the verification and validation program). But even after being shown to work well on Mars, the time required to perform vision processing on the 20 MHz CPU reduced the overall effective drive speed by an order of magnitude, so the utility of the better position estimate had to be weighed against the desire to cover longer distances (Biesiadecki et al., 2007).

In the first two years since landing, the rovers have driven over terrain with as much as 31 degrees of tilt, and over textures comprised of slippery sandy material, hard-packed rocky material, and mixtures of both. Engineering models of vehicle slip in sandy terrain developed during Earth-based testing correlated remarkably well with the sand-only terrain inside Eagle crater during the first two months. However, slip was extremely difficult to predict when the rover was driven over nonhomogeneous terrains (e.g., climbing over rock for one part of a drive and loose soil for another, or climbing over low-lying ripples of sandy material). Early on, the uncertainty in the amount of slip resulting from drives on high slopes or loose soils forced the operations team to spend several days driving toward some targets, even those just a few meters away. But through the rest of the mission, Visual Odometry software has enabled precision drives (i.e., ending with the science target being directly reachable by the IDD) over distances as long as 8 meters on slopes greater than 20 degrees (see Figure 3), and made it possible to safely traverse the loose sandy plains of Meridiani.

2 Algorithm

Work on estimating robot motion with stereo cameras can be traced back to Moravec's work (Moravec, 1980). Following Moravec's work, Matthies et al. treated motion estimation as a statistical estimation problem and developed sequential methods for estimating the vehicle motion and updating the landmark models. This system achieved an accuracy of 2% of distance over 5.5 meters and 55 stereo image pairs (Matthies and Shafer, 1987; Matthies, 1989), with a consistent level of accuracy reported more recently (Olson et al., 2003). Similar work has been reported elsewhere (Zhang et al., 1988; Lacroix et al., 1999; Nistér et al., 2004). Recently, Nistér et al. have reported a successful real-time Visual Odometry implementation (Nistér et al., 2006; Nistér, 2004). This implementation contains two motion estimation schemes: the stereo scheme, which is an iterative pose refinement scheme, and the monocular scheme, which was based on the framework of 5-point algorithm (Nistér, 2004) and an outlier rejection scheme (Nistér, 2005) and good results have been reported on a very long image sequences. Other approaches to the subject of Visual Odometry schemes have been reported as well. For example, McCarthy and Barnes have reported the performance of optical flow based motion estimation (McCarthy and Barnes, 2004) and Vassallo and Gluckman and others have developed ego-motion estimation with omnidirectional images (Vassallo et al., 2002; Gluckman and Nayar, 1998; Corke et al., 2004).

Our approach to position estimation is to find features in a stereo image pair and track them from one frame to the next. The key idea of the present method is to determine the change in position and attitude for two pairs of stereo images by propagating uncertainty in a 3D to 3D pose estimation formulation using maximum likelihood estimation. The basic steps of this method are described as follows.

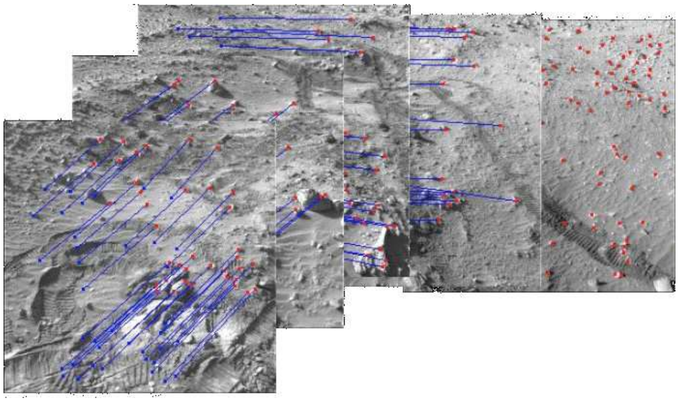

Figure 1: Feature tracking occurs between every pair of images. In this view, several images from Spirit's Sol 178 drive and their tracked features have been superimposed.

Feature Detection First, features that can be easily matched between stereo pairs and tracked across a single motion step are selected. An interest operator tuned for corner detection (e.g. Forstner or Harris) is applied to an image pair, and pixels with the highest interest values are selected. To reduce the computational cost, a grid with cells smaller than a preset minimum distance between features is superimposed on the left image. The feature with strongest corner response in each grid cell is selected as a viable candidate. A fixed number of features having the highest interest operator responses is selected, subject to a minimum distance constraint to ensure that features span the image.

Feature-based Stereo Matching Each selected feature's 3D position is computed by stereo matching. Because the stereo cameras are well calibrated, the stereo matching is done strictly along the epipolar line with only a few pixels of offset buffer above and below it. We use pseudo-normalized correlation to determine the best match. In order to obtain subpixel accuracy, a biquadratic polynomial is fit to a 3x3 neighborhood of correlation scores, and the peak of this polynomial is chosen as the correlation peak.

The 3D positions of these selected features are determined by intersecting rays projected through the camera models. In perfect conditions, the rays of the same feature in the left and right images would intersect at a point in space. However, due to image noise, camera model uncertainty and matching error, they do not always intersect. The shortest distance "gap" between the two rays indicates the goodness of the stereo match: features with large gaps are thrown out.

Next we compute the covariance associated with each feature using methods described in (Matthies and Shafer, 1987) and detailed in (Matthies, 1989).

Note that the covariance of each point $\mathbf{P}$ is

$$\textstyle\sum_{\mathbf{P}} = \mathbf{P}' \begin{bmatrix} \sum_l & 0 \\ 0 & \sum_r \end{bmatrix} \mathbf{P}'^T \tag{1}$$

where $\mathbf{P}'$ is the Jacobian matrix or the first partial derivative of $\mathbf{P}$ with respect to the 2D feature locations in the left and right images, and $\sum_l$ and $\sum_r$ are 2x2 matrices whose elements are the curvatures of the biquadratic polynomial along the vertical, horizontal and diagonal directions, which can be obtained directly from subpixel interpolation.

The quality of a 3D feature is a function of its relative location, the gap between the two stereo rays, and the sharpness of the correlation peak. This covariance computation fully reflects these three factors.

Feature Tracking After the rover moves a short distance, a second pair of stereo images is acquired. The features selected from the previous image are projected into the second pair using the approximate motion provided by onboard wheel odometry (see Figure 1 for some examples). Then a correlation-based search reestablishes the 2D positions precisely in the second image pair. Stereo matching of these tracked features determines their new 3D positions. Because the 3D positions of those tracked features are already known from the previous step, the stereo matching search range can be greatly reduced. A rigidity test (comparing the relative 3D distances of corresponding points) is performed as an initial outlier rejection step.

Robust Motion Estimation If the initial motion is accurate, the difference between two estimated 3D feature positions should be within the error ellipse. However, when the initial motion is off, the difference between the two estimated positions of the 3D points reflects the error of the initial motion and it can be used to determine the change of rover position.

Motion estimation is done in two steps. First, a less accurate motion is estimated by least squares estimation. The error residual between current position $\mathbf{P}_{Cj}$ and previous position $\mathbf{P}_{Pj}$ of the j^{th} feature is

$$\mathbf{e}_j = \mathbf{P}_{Cj} - \mathbf{R}\mathbf{P}_{Pj} - \mathbf{T} \tag{2}$$

and the cost expression is

$$M(\mathbf{R}, \mathbf{T}) = \sum w_j \mathbf{e}_j^T \mathbf{e}_j \tag{3}$$

$$w_j = (det(\sum{}_{Pj}) + det(\sum{}_{Cj}))^{-1} \tag{4}$$

There is a closed form solution for this least squares estimation (Schonemann and Carroll, 1970; Matthies, 1989). Let

$$w = \sum_j w_j \tag{5}$$

$$\mathbf{Q}_c = \sum_j w_j \mathbf{P}_{Cj} \tag{6}$$

$$\mathbf{Q}_p = \sum_j w_j \mathbf{P}_{Pj} \tag{7}$$

$$\mathbf{A} = \sum_j w_j \mathbf{P}_{Pj} \mathbf{P}_{Cj} \tag{8}$$

$$\mathbf{E} = \mathbf{A} - \frac{1}{w} \mathbf{Q}_c \mathbf{Q}_p^T \tag{9}$$

Let $\mathbf{E} = \mathbf{U}\mathbf{S}\mathbf{V}^T$ be the singular value decomposition of $\mathbf{E}$. Then

$$\hat{\mathbf{R}} = \mathbf{U}\mathbf{V}^T \tag{10}$$

$$\hat{\mathbf{T}} = \frac{1}{w} \left[\mathbf{Q}_c - \hat{\mathbf{R}} \mathbf{Q}_p \right] \tag{11}$$

The advantage of this method is that it is simple, fast and robust. Its disadvantage is that its results can be substantially inferior to those derived with a full error model because this only takes the quality of the observations (the volume of the error ellipsoid or determinant of the covariance matrix) as a weight factor (Matthies, 1989).

Because it is an inexpensive operation, we embed it within a RANSAC (Random Sample Consensus) process to do outlier removal:

1. A small set of features (e.g., six) is randomly selected and the motion is then estimated using the least squares estimation method.
2. All features from the previous step are projected into the current image frame using the newly estimated motion. If the gap between a reprojected feature and its correspondent is less than a threshold (e.g. 0.5 pixels), the score of this iteration will be incremented once for each viable feature.

3. Steps 1 and 2 repeat for a fixed number of iterations and the motion with the highest score is selected. All features that pass this iteration will be used in the following more accurate estimation — the maximum likelihood motion estimation.

Maximum Likelihood Estimation The maximum likelihood motion estimation considers the 3D feature position difference and associated error models when estimating rover position (Matthies and Shafer, 1987; Matthies, 1989). As above, let $\mathbf{P}_{Pj}$ and $\mathbf{P}_{Cj}$ be the observed positions of feature j prior to and after the current robot motion. Then

$$\mathbf{P}_{Cj} = \mathbf{R}\mathbf{P}_{Pj} + \mathbf{T} + \mathbf{e}_j \tag{12}$$

where $\mathbf{R}$ and $\mathbf{T}$ are the rotation and translation of the robot and $\mathbf{e}_j$ is the combined error in the observed positions of jth features. In this estimation, three axis rotations $\theta_{\mathbf{R}}$ and translation $\mathbf{T}$ are directly determined by minimizing the summation:

$$\sum \mathbf{e}_j^T \mathbf{W}_j \mathbf{e}_j \tag{13}$$

where $\mathbf{W}_j = (\mathbf{R}\sum_{Pj}\mathbf{R}^{\mathbf{T}} + \sum_{Cj})^{-1}$ is the inverse covariance matrix of $\mathbf{e}_j$. The minimization of this nonlinear problem is done by linearization and an iterative process (Matthies, 1989). The linearization is obtained by taking a first-order expansion of Equation 12 with respect to rotation angles $\mathbf{\Theta}_0(\theta_x, \theta_y, \theta_z)$, and applying the least-squares method:

$$\begin{aligned} \mathbf{P}_{Cj} &= \mathbf{R}\mathbf{P}_{Pj} + \mathbf{T} + \mathbf{e}_j && (14) \\ &\approx \mathbf{R}_0\mathbf{P}_{Pj} + \mathbf{J}_j\left(\mathbf{\Theta} - \mathbf{\Theta}_0\right) + \mathbf{T} + \mathbf{e}_j && (15) \end{aligned}$$

where the 3x3 Jacobian matrix $\mathbf{J}_j$ is given by:

$$\mathbf{J}_j = [\mathbf{R}_x\mathbf{P}_{Pj}\ \mathbf{R}_y\mathbf{P}_{Pj}\ \mathbf{R}_z\mathbf{P}_{Pj}] \tag{16}$$

Where $\mathbf{R}_x, \mathbf{R}_y, \mathbf{R}_z$ are the partial derivatives of the rotation matrix with respect to the rotation angles $\theta_x, \theta_y, \theta_z$, respectively. We obtain a solution for the optimal rotation $\mathbf{\Theta}$ in a least-squared sense, using an iterative method:

$$\begin{aligned} \mathbf{P}_{ij} &= \mathbf{P}_{Cj} - \mathbf{R}_{i-1}\mathbf{P}_{Pj} + \mathbf{J}_{ij-1}\mathbf{\Theta}_{i-1} && (17) \\ \hat{\mathbf{\Theta}}_i &= \left[\sum_j \mathbf{J}_{ij}^T\mathbf{W}_j\mathbf{P}_{ij} - \sum_j \mathbf{J}_{ij}^T\mathbf{W}_j(\sum_j \mathbf{W}_j)^{-1}\sum_j \mathbf{W}_j\mathbf{P}_{ij}\right]^{-1} && (18) \\ &\quad \left[\sum_j \mathbf{J}_{ij}^T\mathbf{W}_j\mathbf{J}_{ij} - \sum_j \mathbf{J}_{ij}^T\mathbf{W}_j(\sum_j \mathbf{W}_j)^{-1}\sum_j \mathbf{W}_j\mathbf{J}_{ij}\right] \end{aligned}$$

Iteration continues until $\mid \hat{\mathbf{\Theta}}_i - \hat{\mathbf{\Theta}}_{i-1} \mid < \sigma$, where typically $\sigma = 0.000006$. Then the optimal translation is obtained from:

Table 1: Optional Constraints on 3D Updates

Constraint	Units
Allowed nonconvergences	
World Coordinate delta X component	meters
World Coordinate delta Y component	meters
3D Update Vector Magnitude	meters
Max Change in Roll	radians
Max Change in Pitch	radians
Max Change in Yaw	radians
Max Angle from Downslope (when tilt is high)	radians
Min Tilt Angle at which to check Downslope	radians

$$\hat{\mathbf{T}} \; = \; (\sum_j \mathbf{W}_j)^{-1} \sum_j \mathbf{W}_j \left(\mathbf{P}_{Cj} - \hat{\mathbf{R}} \mathbf{P}_{Pj} \right) \tag{19}$$

The key advantage of this approach, compared to the scalar-weighted approach (Equation 3), is that incorporating the full 3D covariance matrices of the 3D features in the matrices **W** properly weights the triangulation error, leading to much better motion estimates (Matthies and Shafer, 1987; Matthies, 1989).

The MER Visual Odometry implementation improves on earlier implementations (Olson et al., 2003) in two main areas. First, we used a more accurate feature covariance calculation. In the previous implementation, the feature stereo matching error in image space is a constant for all features, which cannot faithfully reflect the error model. For example, the correlation error in an area of high image texture area would be less than the low texture area, which should be incorporated into the covariance computation. In this MER implementation we used the curvature of the biquadratic polynomial along the vertical, horizontal and diagonal directions to quantify the quality of the correlation matching. The second improvement is the use of RANSAC with the least-squares estimator for outlier rejection based on image reprojection error of the features, which is similar to the procedure suggested by Nistér (Nistér, 2005; Nistér et al., 2006).

As of the February 2005 version of MER flight software, optional constraints can also be placed on the final motion estimate to provide additional sanity checking. The magnitude of the 3D update vector, its X and Y World Frame components, the magnitude of the roll, pitch and yaw updates, and the angular deviation from a purely downslope vector can all be restricted (see Table 1). Any update violating the active set of constraints is treated as an "update failure". Other types of "update failures" are initialization steps (there is nothing to track, hence no update) and failures to converge. The total number of acceptable update failures can also be constrained.

3 Ground-based Validation

Precursor implementations of our Visual Odometry software have been tested on numerous rover platforms, typically resulting in position estimates accurate to within 2% of the distance traveled (Matthies and Shafer, 1987; Olson et al., 2003). Details of those tests can be found

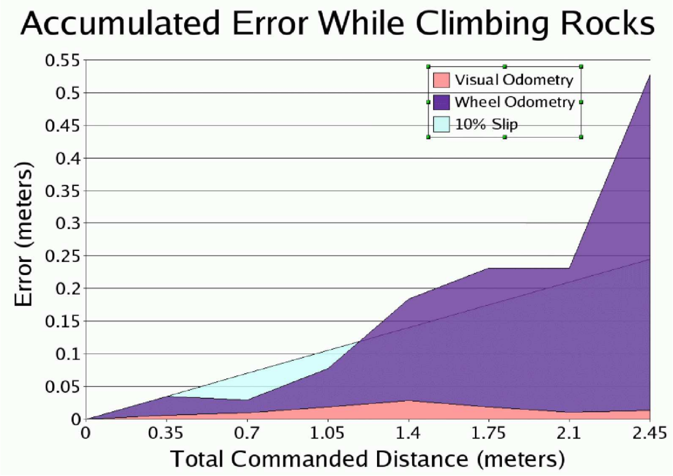

Figure 2: Visual Odometry Error measured during a 2.45 meter drive using HAZCAMs on the MER Surface System Testbed Lite rover. The rover was driven over several large non-obstacle rocks, each less than 20 cm tall, in 35 cm steps. The vehicle was held in place during the final step, so the wheel odometry error for that step is artificially large, yet the Visual Odometry error remains small.

in the cited publications, here we will just summarize a recent test on a research rover and a quick validation performed on a MER engineering model vehicle.

The latest tests were conducted on JPL's Rocky 8 rover at the JPL Marsyard and in Johnson Valley, California (Helmick et al., 2004). Rocky 8 has two pairs of hazard avoidance stereo cameras mounted on the front and rear of the rover body about 50 cm above the ground. The image resolution is 640 by 480, horizontal and vertical fields of view are 80 degrees horizontal by 64 degrees vertical and the baseline is about 8.4 cm. The Johnson Valley site had slopes of loose granular sand where the rover experienced substantial slip, tilt, and roll during the test.

In order to evaluate Visual Odometry performance, high precision ground-truth data (position and attitude) was also collected using a total station (like a surveyor's theodolite with a laser range sensor). By tracking four prisms on top of the rover, the rover's position and attitude were measured with high precision ($<$ 2 mm in position and $<$ 0.2 degree in attitude). The absolute position errors were less than 2.5% over the 24 meter Marsyard course, and less than 1.5% over the 29 meter Johnson Valley course. The rotation error was less than 5.0 degrees in each case (Helmick et al., 2004).

Simple confirmation tests were also run on the MER Surface System Testbed Lite rover in an indoor sandbox test area. Ground truth was acquired using a total station to measure the vehicle's 6-DOF motion estimate by tracking three points at each step. During these tests Visual Odometry processing took place using images from the 120-degree FOV HAZCAM sensors (but on Mars only the 45-degree FOV NAVCAMs are commanded). Several tests were run in which Visual Odometry was found to be as good as wheel odometry on simple terrain (within the design spec), and much better in complex terrain.

As an example, Figure 2 shows the position estimation error that resulted from the most slip-inducing test run on the MER engineering model: a 2.45 meter rock-laden course driven in 35 cm steps. All 6 rocks that were climbed were small enough (less than 20cm in height)

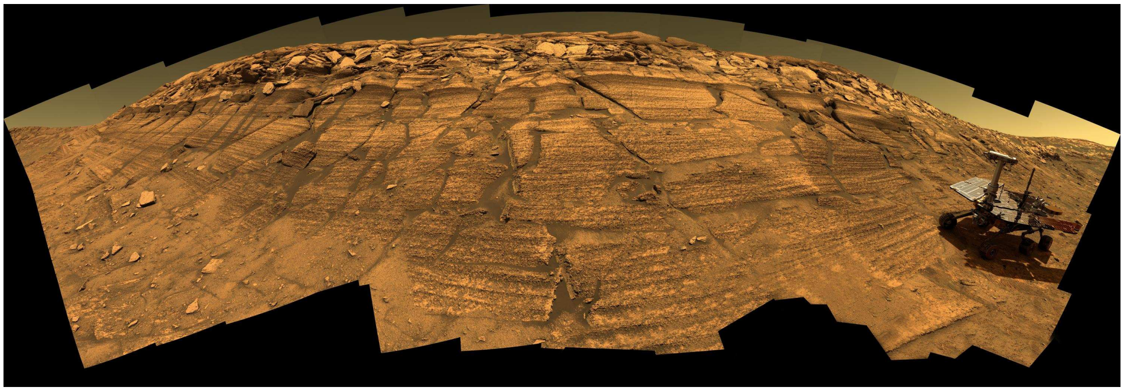

Figure 3: CGI rendering of Opportunity on the side of Burns Cliff in Endurance Crater. Opportunity perched here on slopes ranging from 22 to 31 degrees.

that they would not be considered obstacles. The straight sloped line above the light blue background in the figure represents the design goal of at most 10% error in the position estimate. The dark curve represents the error that accrued when the position was estimated using only the IMU and wheel odometry; after 1.4 meters of driving, the accumulated error had already gone beyond the desired 10% curve. Finally, the light curve at the bottom represents the error remaining after Visual Odometry processing has completed. Even after 2.45 meters of driving over rough obstacles with as much as 85% slip, the Visual Odometry error remained small, less than 1% of the total traverse distance.

Other work on quantitative evaluation of a different Visual Odometry algorithm has been reported by Nistér et al. (Nistér et al., 2006). Their algorithm is formulated to minimize image reprojection error instead of error between 3D feature coordinates. These authors measure performance in terms of errors in integrated path length, compared to differential GPS as ground truth, in order to factor out effects of attitude error. Note that we have used total position error as the performance measure, which does include the effects of attitude error. Nistér et al. reported errors of 1-2% on data sets covering up to 365 meters. It is difficult to draw any conclusions from this comparison, since besides using a different formulation and a different performance metric, this work also had very different computational constraints that allowed tracking two orders of magnitude more features per frame.

4 Operational Constraints

The requirement that Visual Odometry operate on another world leads to operational constraints that make traditional terrestrial solutions inadequate.

For example, consider the feature tracker. Some terrestrial visual odometry systems follow the "update often, search less" mantra: they update so frequently that the motion of each feature between frames is guaranteed to be small (e.g., (Amidi et al., 1999)). This reduces the amount of computation needed to guarantee that features can be successfully tracked. In contrast, the MER vehicles' 20 MHz CPU and low-throughput camera bus result in each Visual Odometry update taking a nontrivial amount of time; up to three minutes for a single tracking step using the nominal mission software. Hence we were driven to the opposite extreme; take images as infrequently as possible, and make the tracker robust to large changes in feature location.

Consider also the feature detector. Ensuring that the images used for Visual Odometry processing contain enough features is still the job of the human rover driver. While planning a drive that uses Visual odometry, rover drivers need to carefully consider which way to point the cameras. Much of Spirit's driving took place in terrain with so many features that pointing was rarely an issue. In contrast, year two saw the introduction of the Slip Check safety constraint on Opportunity to ensure that it would not get bogged down in piles of sand. The feature detector was optimized for feature-laden natural terrain, and often failed to find sufficient unique features in the piles of sand, so another approach had to be taken (happily, the featureless terrain was also very pliable, so the tracks left by the vehicle were not only visible, but also provided enough features to enable convergence).

In the False Positives discussion in Section 5.4 below, we note that Visual Odometry did sometimes produce incorrect updates. So we added additional sanity checks during the second year of operations, to allow the rover drivers to constrain the set of acceptable updates, according to their expectations of the terrain ahead. These optional constraints are summarized in Table 1.

Robustness to nonconvergence is another difficult issue. A system that updates frequently can better absorb the impact of a few frames that fail to track. But MER rover drivers need to write command sequences that closely monitor whether the Visual Odometry processing has succeeded at each step.

5 Using Visual Odometry on Mars

Visual Odometry processing was performed on both MER rovers using mast-mounted NAVCAM imagery. NAVCAMs have a 45-degree field of view and sit 1.5 meters above the ground plane (Maki et al., 2003), so all Visual Odometry drives were split into steps small enough to ensure at least 60% overlap between adjacent images. During each step the rover was typically commanded to drive no more than 75 cm in a straight line or curved arc, and when turning in place was commanded to change heading by no more than 18 degrees per step. Motions outside these bounds forced the process to forego any update for that step.

Although Visual Odometry processing could have been beneficial during all rover motion, each step required an average of nearly three minutes of computation time on MER's 20 MHz RAD6000 CPU, and thus it was only commanded during specific types of motion: relatively short drives (typically less than 15 meters) that occurred either on steep slopes (typically more than 10 degrees), in situations where a wheel was being dragged (digging a trench, or conserving drive motor lifetime on Spirit's right front wheel), or driving through piles of sand. The onboard IMU exhibited a very small drift rate (usually less than 3 degrees per hour of operation) and therefore maintained attitude knowledge very well: therefore during the first two years of operations from January 2004 through January 2006, Visual Odometry was typically used to update rover position only.

There were some instances in which Visual Odometry did not converge to a solution. These are mostly attributable to either too large a motion (e.g. commanding a 40 degree turn in place, resulting in too little image overlap) or to a lack of features in the imaged terrain; but see the description of False Positives in Section 5.4 too. It has successfully measured slips as high as 125% on Sol 206 when Spirit tried to drive up a more than 25 degree slope.

Several benefits were realized from Visual Odometry. Vehicle safety was maintained by having the rover terminate a planned drive early, if it realized via Visual Odometry that it was making insufficient progress toward its goal vis a Slip Check, or was nearing the prespecified “keep-out” location of an obstacle. The improved drive accuracy in new or mixed-soil terrains also yielded a greater number of science observations, by reducing the number of sols needed to make targets reachable by the instrument arm (IDD). PANCAM (Panoramic Camera) and MiniTES science observations requiring precision pointing at a particular target were often scheduled in the middle of a drive using Visual Odometry, which eliminated the need for human confirmation of the pointing angle and therefore saved an extra sol.

5.1 Meridiani Planum: Opportunity Rover

The terrain at Meridiani Planum is a challenging one for Visual Odometry. It is often difficult or impossible to find a patch of nearby terrain that has enough texture for Visual Odometry processing to successfully find and track features, because much terrain is covered by a thick layer of extremely fine particles. Fortunately, areas that have this smooth, featureless appearance tend to be very flat, and in those areas the IMU and encoder-based position estimation has performed well enough that Visual Odometry was not needed. Terrain that exhibits higher slope (and consequently more position uncertainty) almost always has a distinctive appearance (e.g., bedrock outcrop), or is near enough to interesting features that Visual Odometry can be employed successfully.

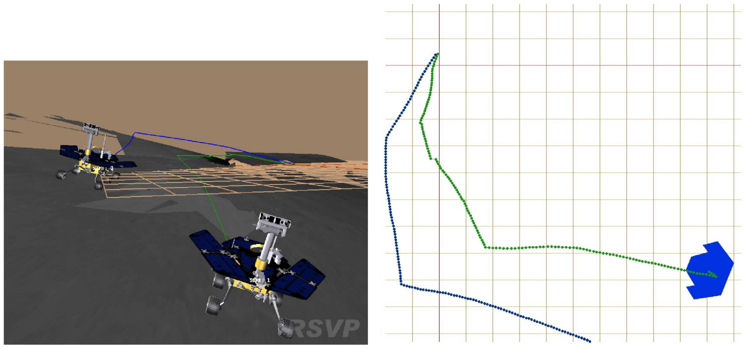

Figure 4: Views of Opportunity’s 19 meter drive from Sol 188 through Sol 191. The inside path shows the correct, Visual Odometry updated location. The outside path shows how its path *would* have been estimated from the IMU and wheel encoders alone. Each cell represents one square meter.

The path predicted by wheel odometry alone can be quite different from the actual path. Figure 4 shows two views of the trajectory taken by Opportunity during Sols 188-191. The rover was driven uphill and across slope over a real distance of 19 meters, but wheel odometry alone would have underestimated it by 1.6 meters and failed to measure the slip-induced elevation change. The outside path indicates the course as it would have been estimated solely by wheel odometry (ignoring the translation and additional distance that resulted from sideslip), and the inside path shows the Visual Odometry-corrected course plot that was generated onboard and more accurately models actual motion. The final positions differ by nearly 5 meters.

The earliest benefit from Visual Odometry came inside 20 meter-diameter Eagle Crater,

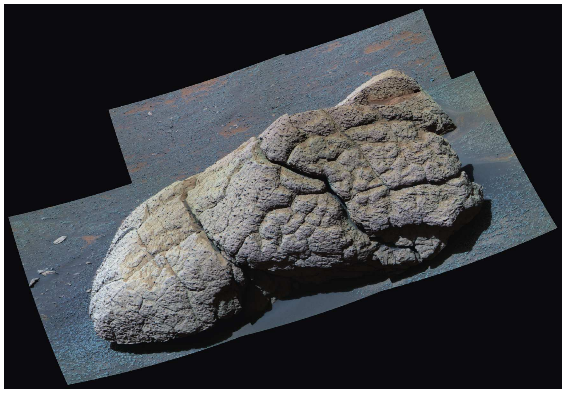

Figure 5: Wopmay, an obstacle in Endurance Crater 60 cm tall, 90 cm wide, and 150 cm long.

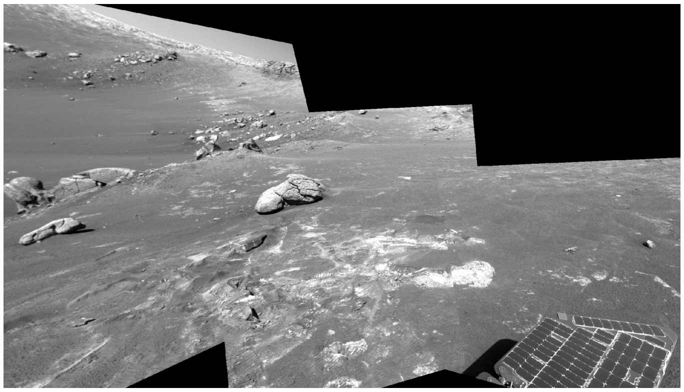

Figure 6: Looking back at Opportunity's tracks near Wopmay on sol 268 after weeks of trying to drive away. The slope of the sandy terrain varied from 17 to 23 degrees.

Opportunity's landing site (upper left corner of Figure 15). Most driving inside Eagle Crater was meticulously planned by human drivers, predicting slip using tables generated by the mechanical team from Earth-based tests of a rover driving in sand (Lindemann and Voorhees, 2005). But while those tables worked well for predicting purely upslope and cross-slope slips on pure sand, no model was available for driving on pure bedrock outcrop, mixtures of bedrock and loose sand, or at angles other than 0, 45 and 90 degrees from the gradient. In those circumstances Visual Odometry was sometimes used to drive to the proper target, or ensure that high resolution PANCAM images of science targets taken after a drive would be pointed right on target.

The most extensive use of Visual Odometry was made by Opportunity inside 130 meter diameter Endurance Crater from Sol 133 to Sol 312 (see the upper right corner of Figure 15). Except for a 12 meter approach and return at the lowest point (with lowest rover tilt) on Sols 201 and 203 and a 17 meter drive on Sol 249, Visual Odometry was used virtually continuously throughout. Had it not been available onboard, many more sols would have been needed to approach targets, and fewer targets would have been achieved.

Visual Odometry not only improved target approach efficiency, it also proved crucial to maintaining vehicle safety. From Sols 249 to 265 Opportunity kept finding itself near a 1.5 meter long rock called Wopmay (see Figures 5 and 7). Although Wopmay was originally considered a science target, it also proved to be a most difficult obstacle to avoid. It was

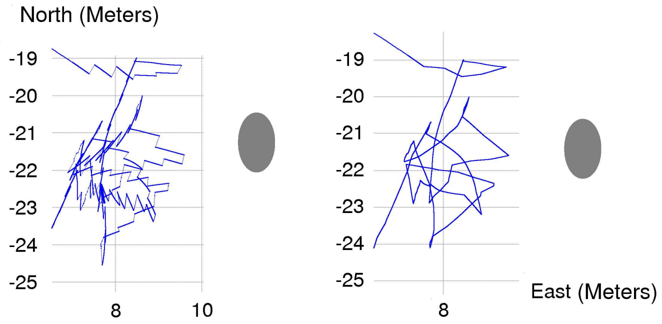

Figure 7: Opportunity's 15 sol trajectory near Wopmay (approximately indicated by the grey ellipse), first driving toward and then trying to get around or away from it. Downslope is up to the right. In the left plot, the "jumps" that point up to the right are the result of Visual Odometry adjusting the vehicle's position downslope. Visual Odometry only corrects the rover's position at the *end* of each step of less than 1 meter. The right plot shows the same course with the Visual Odometry jumps removed.

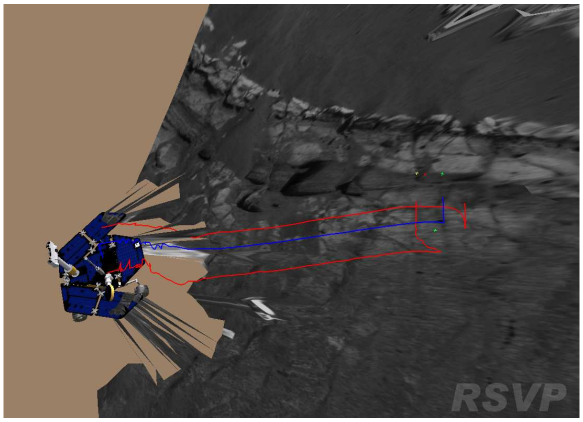

Figure 8: Opportunity's planned 8.7 meter drive along a 20-24 degree slope on Sol 304.

located downhill from a 17-20 degree downslope area comprised of loose sand and buried rocks. Several attempts to drive around it were thwarted not only by very high slip, but also by the unseen rocks buried just beneath the surface (one of which caused a stall on sol 265). Fortunately, the human-commanded sequences took into account the possibility that the rover might slip, and so Opportunity halted several of its planned drives prematurely (and *correctly*) when it realized that it was moving too close to Wopmay. The churned up tracks that resulted from multiple attempts to leave Wopmay can be seen in Figure 6.

Visual Odometry also enabled more precise approaches to difficult targets. On Sol 304, a drive of over 8 meters was planned on an outcrop whose slope varied from 20 to 24 degrees. Because the drive plan took a wide range of potential slips into account, Opportunity was able to drive just far enough across slope, then turn and drive just far enough upslope, to perfectly position the desired target within the IDD work volume in a single sol. Figure 8 illustrates the planned drive, and Figure 9 shows the final image from the body-mounted front Hazard cameras (HAZCAMs) showing the target area perfectly located between and just ahead of the front wheels.

After four months of long distance drives (including 15 sols in which 100 – 224 meters were covered in a single sol (Biesiadecki and Maimone, 2006)), during which Visual Odometry

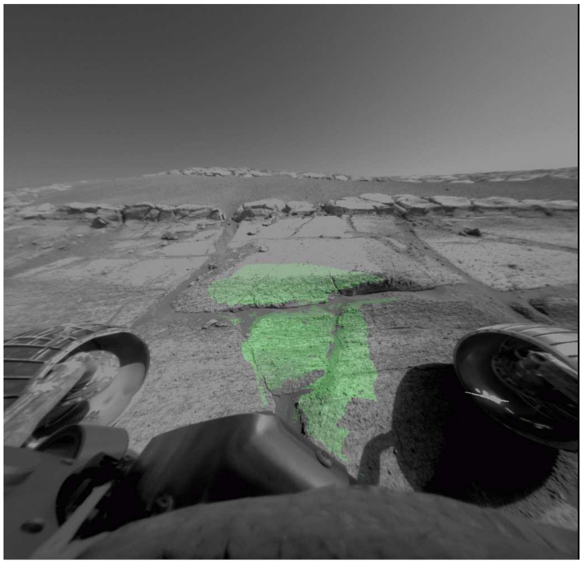

Figure 9: After Opportunity's 8.7 meter slope drive on Sol 304, the goal area is perfectly reachable inside the IDD work volume, indicated in the green shaded area in the middle of the image.

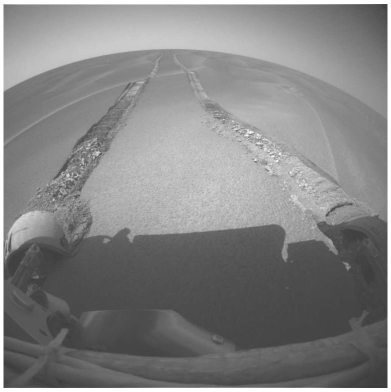

Figure 10: Our first close-up first view of Purgatory ripple from sol 446. The churned up tracks near the rover are only 2 meters long, but the wheels had rotated enough to have traveled 50 meters on open terrain. This event led us to use the Slip Check drive mode for all future drives across sandy terrain.

was rarely used, it suddenly came to the fore again. On sol 446, Opportunity nearly buried its wheels in Purgatory ripple, a nondescript pile of sand, by executing 50 meters of blind driving while failing to climb over it (see Figure 10). How could we get out of it, and how could we stop ourselves from doing the same thing in the next pile of sand when we did finally break free? After much deliberation, a solution was proposed that hinged on Visual Odometry. After each commanded 2 meters of driving, Visual Odometry would be used to estimate the vehicle's motion. If Visual Odometry confirmed that little or no motion had taken place, more commands would be sent. But once Visual Odometry measured a nontrivial motion or failed to converge, all drive commands would stop so we could take a close look back at the pile that had caused so much trouble (see Figure 11).

Experiments verified that although the Visual Odometry feature detector often failed to find enough features when looking at piles of sand, there was a set of features rich enough to enable

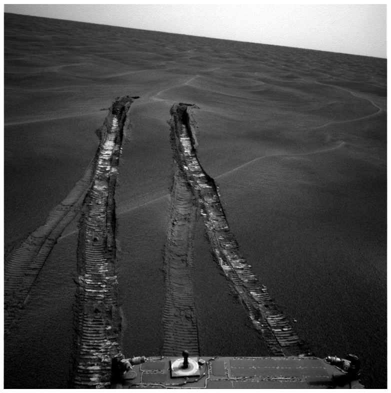

Figure 11: Looking back on sol 491 at Opportunity's tracks on and near Purgatory ripple, after having spent a month escaping it.

Table 2: Assessment of 256x256 NAVCAM-based Onboard Visual Odometry during Purgatory extraction. For each sol, we considered three points in 3D: start position, end position as estimated onboard, and end position as estimated on Earth (Ground) using Hi-Resolution PANCAM images. Columns indicate activity Sol, number of NAVCAM Visual Odometry updates computed onboard, start to Onboard end position magnitude, start to Ground end position magnitude, vector between end positions magnitude, vector between end positions magnitude divided by number of updates.

Sol	**Num. of Onboard Updates**	**Onboard Distance Estimate**	**Ground Distance Estimate**	**Hi-res Pancam Correction**	**Average Error per Step**
463	10	0.029 m	0.030 m	0.0084 m	0.0008 m
465	10	0.020 m	0.020 m	0.0090 m	0.0009 m
466	10	0.033 m	0.026 m	0.0121 m	0.0012 m
467	10	0.029 m	0.021 m	0.0196 m	0.0020 m
468	20	0.037 m	0.039 m	0.0125 m	0.0006 m
469	5	0.010 m	0.011 m	0.0026 m	0.0005 m
470	6	0.059 m	0.057 m	0.0053 m	0.0009 m
471	6	0.063 m	0.061 m	0.0071 m	0.0012 m

successful estimation: the rover's own tracks. This strategy proved highly successful. From sol 463 through sol 483, onboard Visual Odometry measured progress in the millimeters; slip rates varied from 98.9% to 99.5%. These numbers were confirmed both by manual inspection of the images, and also by running Visual Odometry processing on Earth using images from Opportunity's high-resolution PANCAM stereo pair taken before and after each sol's drive; progress on the order of 1 millimeter of actual motion vs 2 meters commanded (with wheels steered to perform a 23 degree heading change) was seen for many sols. Finally on sol 484, all driving commands were terminated when Visual Odometry failed to converge three times indicating that it had driven so far that features could not be tracked.

From this point on, a new driving strategy was adopted for Opportunity. Blind drives over sandy terrain would be limited to some maximum distance, initially 5 meters. After each 5 meter drive segment, Visual Odometry would be used to perform a Slip Check (running Visual Odometry over a short, 20 cm drive step) to ensure that the rover was still capable of forward motion. If the amount of forward progress was less than some minimum fraction, all driving would be stopped. The Slip Check has stopped several drives from digging in (e.g., sols 501, 603), and thus enabled driving to resume one sol later instead of the 39 sols required to exit Purgatory.

Visual Odometry results are summarized in Table 3. As of February 2006, Opportunity has thrived for 720 sols. Visual Odometry was used more here than on Spirit, because Opportunity spent more of its first year on slippery surfaces. It has converged to a solution 95% (828/875) of the time.

5.2 Performance Assessment

We will not attempt to fully characterize the accuracy of the updates computed by Visual Odometry over multiple long drives on Mars, because our means of measuring what actually happened is extremely limited. Publications such as those cited in Section 3 provide the best performance measurements for the algorithm, albeit in Earth-based testbeds and without all the additional constraint checks. However, the extraction from Purgatory does provide a limited means of evaluating the performance of onboard Visual Odometry.

In general we have no ground truth against which to compare the position estimates calculated from the 256x256 NAVCAMs. But during the drives that were performed to escape from the Purgatory ripple, not only were NAVCAM images of the tracks 1.3 – 4 meters away taken between each drive step, high-resolution 1024x1024 PANCAMs of a single track 2.5 – 4 meters away were also taken at the beginning and end of each sol's multi-step drive. The actual daily distance driven during the first eight sols was never more than 7 cm, so while these drives are not representative of the mission as a whole, they do give us a means of estimating the average error introduced by running Visual Odometry onboard. By treating the results from high-resolution PANCAM images as an independent estimate of the distance driven each sol ("ground truth"), we can estimate the precision with which the onboard estimates are computed.[1] We performed Visual Odometry processing on Earth using the PANCAM images, and then compared the delta computed from each sol's before-and-after high-resolution 1024x1024 images against the cumulative onboard estimate computed from multiple low-resolution 256x256 NAVCAM pairs. The magnitude of the "correction" computed from the high-resolution PANCAM pair divided by the number of updates generated onboard gives us an average error associated with each Visual Odometry update that was run onboard during a particular sol. Table 2 summarizes the data, which demonstrates that the average error from a single update was less than 2 millimeters during each of the eight sols in the table, which is within the measurement error of the "ground truth".

In a qualitative sense, we know that our algorithm performs well because the rover's behavior matches our predictions better when Visual Odometry is enabled. To truly measure the performance of our algorithm in nominal Martian terrain, we would need a good source of ground truth and substantial data reduction efforts. Possible sources of this information include MER stereo views of the tracks left by Visual Odometry-enabled drives (with potential precision at the centimeter level, but difficult to correlate with each step), post-drive bundle-adjusted localization estimates such as those summarized in (Li et al., 2005; Li et al., 2006) (of various precisions), and orbital images (e.g., 1.5 meter resolution images

[1] We estimate the stability of the high-resolution PANCAM updates themselves by comparing PANCAM images taken while the rover was motionless (comparing each end-of-sol PANCAM against the next sol's beginning-of-sol PANCAM). Except for lighting differences such as shadows, these views should have been identical. Over seven of the same eight sols, the mean computed drift was 1.3 mm +/- 0.5 mm, with a maximum drift of 2 mm.

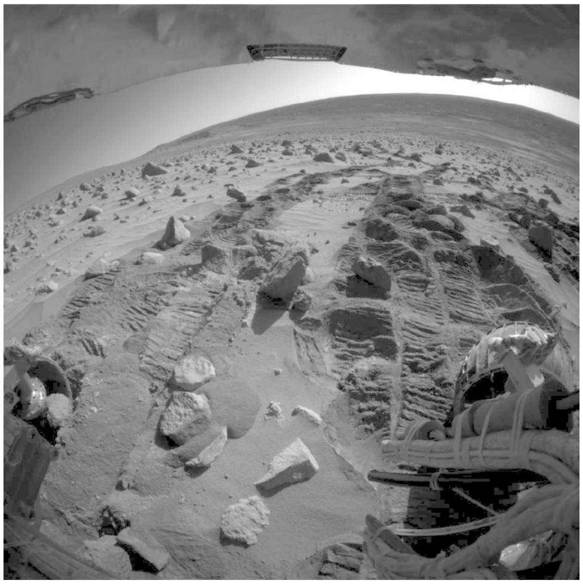

Figure 12: Spirit spent sols 339 through 345 dealing with a potato-sized rock that got stuck in the right rear wheel (viewed here on the left side of this image from the rear hazard camera).

from Mars Global Surveyor, and 0.25 meter resolution images from Mars Reconnaissance Orbiter). Good candidate locations for extensive use of Visual Odometry on Opportunity would be Endurance Crater and the area around Purgatory ripple, and on Spirit the initial exploration of the Columbia Hills (in general, any dense blue area in Figures 14 and 15). Additional performance quantification may be achievable from these resources, but that is left to future work. Raw images from the rovers are available on the Mars Rovers web site.

5.3 Gusev Crater: Spirit Rover

The terrain at Gusev crater is well suited for Visual Odometry processing. The rock abundances there matched predicted distributions (Golombek and Rapp, 1997), resulting in a generally feature-rich landscape with detailed textures comprised of rocks of different sizes and brightnesses. When planning for drives using Visual Odometry, rover drivers typically only had to bear in mind the restriction that adjacent frames should have at least 60% image overlap, only occasionally having to avoid pointing the cameras at infrequently occurring sand dunes. As a result, Spirit's Visual Odometry software has performed well.

One unique driving mode that benefited a great deal from Visual Odometry on Spirit was wheel-dragging. The right front wheel was found to draw more current while driving than any of the other wheels starting on Sol 125. This concern led to the development of a driving strategy to conserve motor lifetime, during which that wheel would be dragged while all the others were driven. Although this was found to enable reasonable progress on relatively flat terrain, error in the position estimate grew substantially in this mode. The Visual Odometry capability meant that not only could progress be made, but also the error added to the onboard position estimate could be bounded as well.

Another invention by the rover drivers was a Visual Odometry-based Obstacle Check style of driving (Biesiadecki et al., 2007). Human rover drivers could create a set of keep-out zones

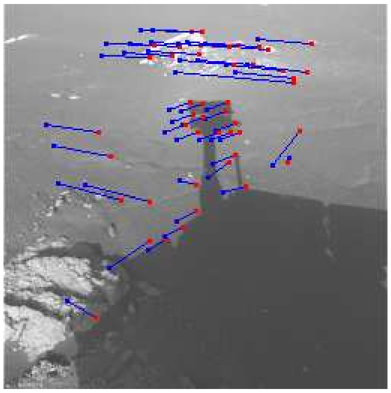

Figure 13: Image of Opportunity's shadow on Sol 235, with tracked features overlaid. Red dots indicate the features in this current image, blue dots and vectors show the locations of corresponding features found in the previous image,

around whatever was perceived to be a nearby obstacle. By driving with Visual Odometry enabled all the time, we could ensure that the rover would stop driving if it ever slipped too close to any obstacle by checking the rover's corrected position against the preset list of obstacle locations. This style of driving was used whenever commanded drives took the rover uphill of any obvious obstacles. Although the rover was capable of detecting rocks and geometric hazards on its own (Biesiadecki and Maimone, 2006), this human-guided obstacle detection was used to allow drives to execute more quickly and hence cover larger distances. Also, the MER vehicles have no means of detecting non-geometric hazards (like slippery sandy areas) onboard at all. Whenever there was a concern about such non-geometric hazards, human drivers had to manually create a keep-out zone around them. The mobility sequence of commands would then poll the rover's location and compare it to the list of keep-out areas, halting the drive if it got too close.

On sol 339 Spirit's right rear wheel stalled when a potato-sized rock that had been buried under soil got wedged in that wheel (see Figure 12). Up to this point, such small rocks had posed no hazards to mobility, and Visual Odometry had been used only as a tool for improving overall position knowledge. But the experience of having to spend a week extricating the potato-sized rock led the rover drivers to create a new use for Visual Odometry: as a Slip Checking device. Rather than worry about the overall position, Visual Odometry could be used infrequently to measure how much *relative* slip the rover had just encountered; for example, if a 40cm arc had been commanded but fewer than 20cm of progress was measured, then it would be clear that the rover was slipping more than 50%. The sol 339 event led to Slip Checks becoming a standard speed optimization; instead of driving with Visual Odometry running all the time (with a max drive rate of only around 10 meters/hour), Slip Checks could be used to place an upper bound how deeply we might get dug in (e.g., by limiting blind drive segments to 5 meters).

Relatively little slip was seen during the first six months of Spirit's mission. But once the base of the Columbia Hills was reached, drives up into the hills were found to exhibit much more unpredictable slip. Thus Visual Odometry has been used during most of the drives in the Columbia Hills, especially to ensure that Spirit stays far enough away from nearby rock obstacles. The average tilt of the rover during those times that Visual Odometry was commanded was 14.4 degrees +/- 4.4 degrees, counting 625 samples spanning an absolute range from 2 - 30 degrees.

Visual Odometry results are summarized in Table 3. As of March 2005, Spirit has thrived for 414 sols. Visual Odometry was only used on Spirit after it had reached the Columbia Hills, nearly six months into its mission. But since then it has converged to a solution 97% (590/609) of the time.

Table 3: Results of running Visual Odometry (Visodom) on Mars. Expressions $m \pm s$ indicate mean m and standard deviation s.

	Spirit		**Opportunity**	
Lifetime as of 4 March 2005	414	sols	394	sols
Total Drive Distance	4161	meters	3158	meters
Days Spent Driving	184	sols	172	sols
Days Using Visual Odometry	52	sols	75	sols
Nominal Evaluation Steps	609	pairs	875	pairs
Nominal Initialization Steps	57	pairs	75	pairs
Forced Init by Large Turn	10	pairs	11	pairs
Forced Init by Planned Repointing	5	pairs	9	pairs
Forced Init by Driving Too Far	1	pairs	2	pairs
Total Visodom Image Pairs	682	pairs	972	pairs
Successful (non-initial) Convergences	590	pairs	828	pairs
Iterations (assume Convergence)	6.4 ± 1.7	iterations	8.4 ± 5.2	iterations
Features Tracked at each Step	73.4 ± 29.3	features	87.4 ± 34.1	features
Non-convergences	19	pairs	47	pairs
Mean Updates per Drive Sol	12.0 ± 8.5	pairs	12.9 ± 10.2	pairs
Max Updates per Drive Sol	33	pairs	59	pairs
Mean Rover Tilt During Visodom	14.6 ± 4.4	degrees	18.0 ± 4.6	degrees
Rover Tilt Range During Visodom	2 - 30	degrees	0.8 - 31	degrees

5.4 False Positives

Although we had never seen Visual Odometry converge to an inaccurate solution during testing, on Opportunity Sols 137 and 141 several unreasonable position updates were computed onboard. These are attributable to an improper parameter setting; at that time, the minimum separation between features was too small. As a result, the set of detected features was allowed to cluster tightly around a small planar but feature-rich area. Increasing that parameter was all that was needed to allow the software to find additional features with better spatial distribution and then converge to a reasonable solution.

The only other instance of a false positive solution in this part of the mission was on Sol 235. During that sol the NAVCAMs were pointed at two small and widely separated rocks. Although features were found on those rocks, more usable features were found on the largest shape in the image; the rover's shadow (see Figure 13). Even though forward drive progress was made, the onboard estimator assumed that the shadow (having more consistently tracked features spread throughout the image) better reflected actual motion, and therefore produced an incorrect estimate. This problem would not have arisen had there been more interesting texture around and under the shadow, so since that sol human drivers have had to take the rover shadow into account whenever planning Visual Odometry drives.

5.5 Suggested Improvements

The MER mission has demonstrated the usefulness of Visual Odometry technology for planetary exploration. But several open issues remain to be resolved for future missions.

Runtime The MER system architecture was designed for robustness, not speed. Image acquisition alone could take tens of seconds, including the 2.5 second acquisition of a 256x1024 binned pair of images (multiplied by however many images are needed for auto-exposure processing), plus tens of seconds more to prepare the image for potential transmission as a separate data product to Earth. The Visual Odometry processing itself took place on a single 20 MHz CPU that shared system resources between the intensive image processing operations and dozens of realtime tasks. One complete processing cycle from image acquisition through the position update took 3 minutes on average, limiting overall drive speed to approximately 10 meters/hour (this is expected to improve somewhat in the September 2006 flight software update). Faster processing would allow more frequent use of Visual Odometry, improving vehicle safety, predictability, and autonomous capability.

Finding Good Terrain The implicit assumption in any Visual Odometry algorithm is that the terrain exhibits a rich visual texture, but MER encountered many areas with little or no easily-tracked texture, on both rovers. Spirit's terrain at Gusev crater typically had an interesting appearance that made tracking easy, but occasional sandy areas caused problems. Opportunity's terrain at Meridiani was the opposite, in that *most* of it lacked trackable features. To become a truly autonomous capability, Visual Odometry would need to be extended to autonomously (and efficiently) search for appropriately textured terrain, instead of requiring human operators to select the viewing direction. The search space might include physical changes like re-pointing the cameras or switching lens filters, or algorithmic improvements to detect or eliminate shadows, dynamically update the imaging parameters to adapt to the terrain, and switch from feature-based to area-based homography or shape registration methods autonomously (e.g., (Olson and Matthies, 1998)) (NAVCAM-based dense stereo processing works well even on the sandy terrain that exhibits no obvious features).

Wide-angle lenses Our feature tracker was not able to track features that underwent large scale changes between images. This prevented us from using wide-angle HAZCAM images for position estimation on Mars, even though they seemed to work well on Earth; although HAZCAM tests worked in our indoor test area, we later determined that the indoor success was largely due to having tracked features on the walls, which did not change scale between motions. With no walls on Mars, and a desire to maximize the distance between images to enable fast driving, the HAZCAMs were eliminated as a useful sensor for Visual Odometry on MER. A future ability to use the wide-angle HAZCAMs could save time if HAZCAMs had already been acquired for autonomous obstacle assessment, and would also make it possible to see more of the terrain (and hence find more trackable features without having to repoint cameras). Lowe's work on scale invariant features is one possible approach (Se et al., 2002).

Flexibility Maintaining program flexibility is critical in adapting technology to novel domains, and Mars is full of surprises. For instance, although originally tested on 128x128 HAZCAM images, Visual Odometry was found to work best using 256x256 NAVCAMs. Also, the rover often had to be driven backward to give the NAVCAMs an unobstructed view of the tracks, since rover tracks provided the most viable features in the sands of Meridiani Planum. Most spaceflight control software is written with an eye toward reducing complexity, but flexibility is paramount when new terrains and new surprises are found every sol.

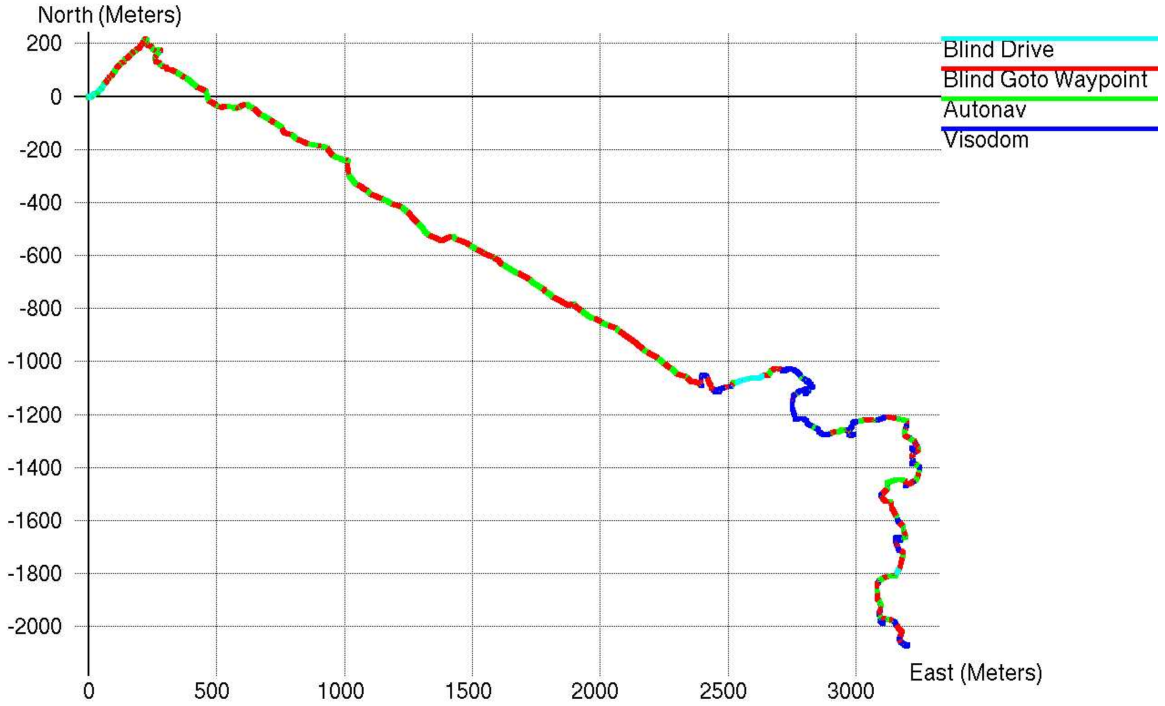

Figure 14: Plot of Spirit's traverse history using Visual Odometry in the Columbia Hills from sols 1 – 850. Units are in meters from the landing site origin, as measured onboard the rover. Cyan lines indicate directly commanded "blind" drives, red lines indicate blind drives with autonomous heading compensation, green lines indicate autonomous hazard detection, and blue lines indicate Visual Odometry. Spirit only used Visual Odometry within the Columbia Hills, not during its three kilometer trek to reach them.

6 Conclusion

Visual Odometry has been a highly effective tool for maintaining vehicle safety while driving near obstacles on slopes, achieving difficult drive approaches in fewer sols, performing Slip Checks to ensure the vehicle is still making progress, and ensuring accurate science imaging. Although it requires active pointing by human drivers in feature-poor terrain, the improved position knowledge enables more autonomous capability and better science return during planetary operations.

For planetary exploration applications, the primary performance limitation for Visual Odometry at present is the runtime of the image processing part of the task given the highly constrained flight computers available, not accuracy or precision of the motion estimates. Therefore, addressing this runtime bottleneck is the overriding immediate priority for future work in this domain, through any combination of algorithmic improvements and advances in flight computers. In the longer term, more robust feature tracking and autonomous terrain selection would be needed to make it a truly robust technology.

7 Acknowledgments

Thanks to Jeff Biesiadecki for integrating this capability into the MER flight software Mobility Manager state machine, the Surface Navigation and Mobility testing team for testing and

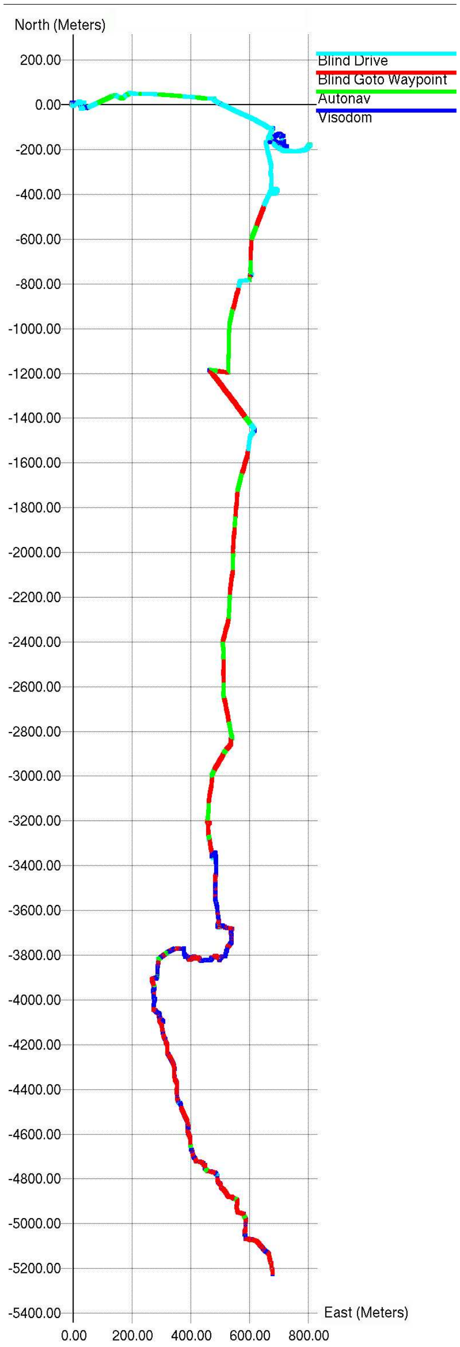

Figure 15: Plot of Opportunity's traverse history using Visual Odometry from sols 1 – 830. Opportunity landed in 20 meter diameter Eagle crater in the upper left, drove in and around Endurance crater (upper right) from sols 133 – 312, and continued to drive south toward Victoria crater as of sol 830. Units are in meters from the landing site origin, as measured onboard the rover. Cyan lines indicate directly commanded "blind" drives, red lines indicate blind drives with autonomous heading compensation, green lines indicate autonomous hazard detection, and blue lines indicate Visual Odometry.

validating it in time to be included the mission, flight software lead Glenn Reeves for supporting its inclusion, Steve Goldberg, Dan Helmick, Clark Olson and Marcel Schoppers for their contributions to algorithm development and implementation, the RSVP team (Brian Cooper, Frank Hartman, Scott Maxwell, John Wright and Jeng Yen) for the 3D rover course visualizations, and Robert Liebersbach for the course plot and corrected pose plotting tools.

The work described in this paper was carried out at the Jet Propulsion Laboratory, California Institute of Technology, under a contract to the National Aeronautics and Space Administration.

References

Amidi, O., Kanade, T., and Fujita, K. (1999). A visual odometer for autonomous helicopter flight. *Journal of Robotics and Autonomous Systems*, 28:185 – 193. http://www.ri.cmu.edu/pub_ files/pub4/amidi_ omead_ 1999_ 1/amidi_ omead_ 1999_ 1.pdf.

Biesiadecki, J. J., Baumgartner, E. T., Bonitz, R. G., Cooper, B. K., Hartman, F. R., Leger, P. C., Maimone, M. W., Maxwell, S. A., Trebi-Ollenu, A., Tunstel, E. W., and Wright, J. R. (2005). Mars Exploration Rover surface operations: Driving Opportunity at Meridiani Planum. In *IEEE Conference on Systems, Man and Cybernetics*, The Big Island, Hawaii, USA.

Biesiadecki, J. J., Leger, P. C., and Maimone, M. W. (2007). Tradeoffs between directed and autonomous driving on the Mars Exploration Rovers. *International Journal of Robotics Research*. Special Issue on ISRR 2005, to appear.

Biesiadecki, J. J. and Maimone, M. W. (2006). The Mars Exploration Rover surface mobility flight software: Driving ambition. In *IEEE Aerospace Conference*, volume 5, Big Sky, Montana, USA.

Corke, P. I., Strelow, D., and Singh, S. (2004). Omnidirectional visual odometry for a planetary rover. In *Proceedings of IROS 2004*. http://www.ri.cmu.edu/pubs/pub_ 4913.html.

Gluckman, J. and Nayar, S. K. (1998). Ego-motion and omnidirectional cameras. In *ICCV*, pages 999–1005.

Golombek, M. and Rapp, D. (1997). Size-frequency distributions of rocks on Mars and Earth analog sites: Implications for future landed missions. *Journal of Geophysical Research - Planets*, 102(E2):4117–4129.

Helmick, D., Cheng, Y., Roumeliotis, S., Clouse, D., and Matthies, L. (2004). Path following using visual odometry for a Mars rover in high-slip environments. In *IEEE Aerospace Conference*, Big Sky, Montana, USA.

Lacroix, S., Mallet, A., Chatila, R., and Gallo, L. (1999). Rover self localization in planetary-like environments. In *International Symposium on Artificial Intelligence, Robotics, and Automation for Space (i-SAIRAS)*, pages 433–440, Noordwijk, Netherlands. ftp://ftp.laas.fr/pub/ria/simon/isairas99.ps.gz.

Li, R., Archinal, B. A., Arvidson, R. E., Bell, J., Christensen, P., Crumpler, L., Marais, D. J. D., Di, K., Duxbury, T., Golombek, M., Grant, J., Greeley, R., Guinn, J., Johnson, A., Kirk, R. L., Maimone, M., Matthies, L. H., Malin, M., Parker, T., Sims, M., Thompson, S., Squyres, S. W., and Soderblom, L. A. (2005). Spirit rover localization and topographic mapping at the landing site of Gusev Crater, Mars. *JGR-Planets, Special Issue on Spirit Rover*, 111(E02S06, doi:10.1029/2005JE002483). http://www.agu.org/journals/ss/SPIRIT1/.

Li, R., Arvidson, R. E., Di, K., Golombek, M., Guinn, J., Johnson, A., Maimone, M., Matthies, L. H., Malin, M., Parker, T., and Squyres, S. W. (Submitted 2006). Opportunity rover localization and topographic mapping at the landing site of Meridiani Planum, Mars. *Journal of Geophysical Research - Planets.*

Lindemann, R. A. and Voorhees, C. J. (2005). Mars Exploration Rover mobility assembly design, test and performance. In *IEEE Conference on Systems, Man and Cybernetics*, The Big Island, Hawaii, USA.

Maki, J. N., III, J. F. B., Herkenhoff, K. E., Squyres, S. W., Kiely, A., Klimesh, M., Schwochert, M., Litwin, T., Willson, R., Johnson, A., Maimone, M., Baumgartner, E., Collins, A., Wadsworth, M., Elliot, S. T., Dingizian, A., Brown, D., Hagerott, E. C., Scherr, L., Deen, R., Alexander, D., and Lorre, J. (2003). Mars Exploration Rover engineering cameras. *Journal of Geophysical Research*, 108(E12):12–1–12–24. http://www.agu.org/pubs/crossref/2003/2003JE002077.shtml.

Matthies, L. (1989). *Dynamic Stereo Vision*. PhD thesis, Carnegie Mellon University Computer Science Department. CMU-CS-89-195.

Matthies, L. and Shafer, S. (1987). Error modelling in stereo navigation. *IEEE Journal of Robotics and Automation*, RA-3(3).

McCarthy, C. and Barnes, N. (2004). Performance of optical flow techniques for indoor navigation with a mobile robot. In *International Conference on Robotics and Automation*, pages 5093–5098, New Orleans, USA.

Moravec, H. (1980). *Obstacle Avoidance and Navigation in the Real World by a Seeing Robot Rover*. PhD thesis, Stanford University, Stanford, CA.

Nistér, D. (2004). An efficient solution to the five-point relative pose problem. *IEEE Trans. Pattern Anal. Mach. Intell.*, 26(6):756–777.

Nistér, D. (2005). Preemptive ransac for live structure and motion estimation. *Machine Vision and Applications*, 16(5):321–329.

Nistér, D., Naroditsky, O., and Bergen, J. (2004). Visual odometry. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 652 – 659. IEEE Computer Society Press.

Nistér, D., Naroditsky, O., and Bergen, J. (2006). Visual odometry for ground vehicle applications. *Journal of Field Robotics*, 23(1):3–20.

Olson, C. F. and Matthies, L. H. (1998). Maximum likelihood rover localization by matching range maps. In *International Conference on Robotics and Automation*, pages 272–277.

Olson, C. F., Matthies, L. H., Schoppers, M., and Maimone, M. W. (2003). Rover navigation using stereo ego-motion. *Robotics and Autonomous Systems*, 43(4):215–229.

Schonemann, P. H. and Carroll, R. M. (1970). Fitting one matrix to another under choice of a central dilation and a rigid motion. *Psychometrika*, pages 245–255.

Se, S., Lowe, D., and Little, J. (2002). Mobile robot localization and mapping with uncertainty using scale-invariant visual landmarks. *International Journal of Robotics Research*, 21(8):735–758.

Vassallo, R. F., Santos-Victor, J., and Schneebeli, J. (2002). A general approach for egomotion estimation with omnidirectional images. In *OMNIVIS*, pages 97–103.

Zhang, Z., Faugeras, O., and Ayache, N. (1988). Analysis of a sequence of stereo scenes containing multiple moving objects using rigidity constraints. In *International Conference on Computer Vision*, pages 177–186, Tampa, Florida. Computer Society of the IEEE, IEEE Computer Society Press.