

# Visual Odometry on the Mars Exploration Rovers

Yang Cheng and Mark Maimone  
Jet Propulsion Laboratory  
Pasadena, CA USA  
yang.cheng@jpl.nasa.gov  
mark.maimone@jpl.nasa.gov

**Abstract**— NASA’s Mars Exploration Rovers (MER) were designed to traverse in Viking Lander-I style terrains: mostly flat, with many small non-obstacle rocks and occasional obstacles. During actual operations in such terrains, onboard position estimates derived solely from the onboard Inertial Measurement Unit (IMU) and wheel encoder-based odometry achieved well within the design goal of at most 10% error in the onboard position estimate. However, the MER vehicles were also driven along slippery slopes tilted as high as 31 degrees. Under those conditions, an additional sensing mode was employed to maintain a sufficiently accurate onboard position estimate: *Visual Odometry*.

The MER Visual Odometry system comprises onboard software for comparing stereo pairs taken by the pointable mast-mounted 45 degree FOV Navigation stereo cameras (NAVCAMs). The system computes an update to the 6-DOF rover pose ( $x, y, z$ , roll, pitch, yaw) by tracking the motion of autonomously-selected “interesting” terrain features between two pairs of stereo images, in both 2D pixel coordinates and 3D world coordinates. A maximum likelihood estimator is applied to the computed 3D offsets to produce a final motion estimate.

In this paper we describe the Visual Odometry algorithm used on the Mars Exploration Rovers, and summarize the results obtained by running this processing onboard during the first year of operations on Mars.

**Index Terms**—MER, Mars Exploration Rover, Visual Odometry, Motion Estimation, Egomotion

## I. BACKGROUND

Keeping track of a vehicle’s location is one of the most challenging aspects of planetary rover operations. The Mars Exploration Rovers (MER) are typically 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”), so the maintenance of the rover’s position estimate onboard is of critical importance.

During each MER mission, the 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 based on encoder readings of how much the wheels turned (wheel odometry).

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 is applied to the computed 3D offsets to produce the final motion estimate. However, if any of the many internal consistency checks fails, too few feature points are detected, or the estimation procedure 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 in Figures 1 and 2, human rover drivers have made extensive use of the Visual Odometry software during high-tilt operations: driving Opportunity inside Eagle and Endurance craters, and climbing Spirit through the Columbia Hills.

In the first year 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 between the two. Engineering models of vehicle slip in sandy terrain were developed during Earth-based testing, and they correlated remarkably well with some Meridiani terrains. However, slip behavior was extremely difficult to predict in areas where the rover was driven over nonhomogeneous terrains (e.g., when climbing over rock for one part of the drive and loose soil for another). 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 that were just a few meters away. But throughout most of the first year of operations, use of Visual Odometry software has enabled precision drives (e.g., ending with the target of interest being immediately reachable by the IDD) over distances as long as 8 meters, on slopes greater than 20 degrees.

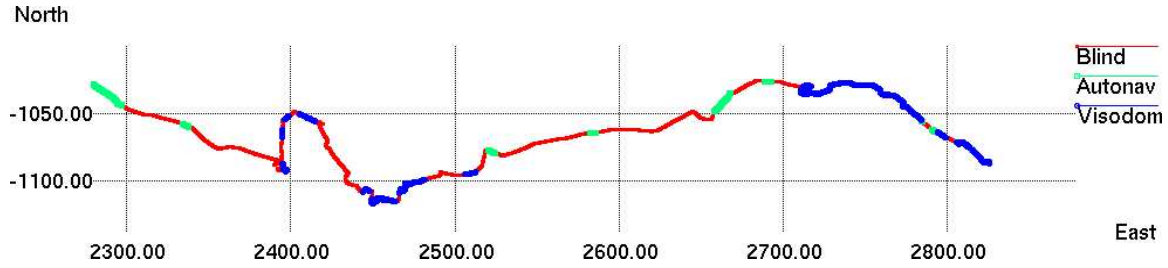


Fig. 1. Plot of Spirit's traverse history using Visual Odometry in the Columbia Hills from sols 178 – 418. Units are in meters from the landing site origin, as measured onboard the rovers. Red lines indicate directly commanded "blind" drives, green lines indicate autonomous hazard detection, and blue lines indicate visual odometry.

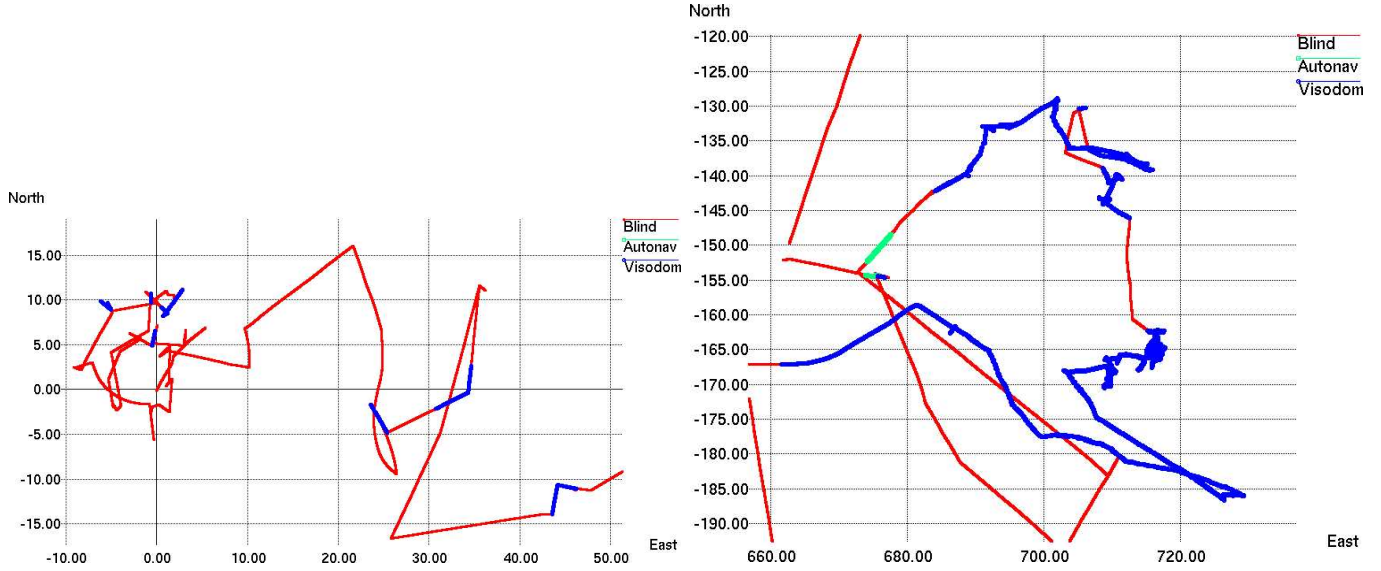


Fig. 2. Plot of Opportunity's traverse history using Visual Odometry. On the left, the drive in and around Eagle crater from sols 1 – 70. On the right, the drive in and around Endurance crater from sols 133 – 312. Units are in meters from the landing site origin, as measured onboard the rovers. Red lines indicate directly commanded "blind" drives, green lines indicate autonomous hazard detection, and blue lines indicate visual odometry.

## II. ALGORITHM

Our approach to position estimation is to find features in a stereo image pair and track them from one frame to the next. This approach, known as Visual Odometry or ego-motion estimation, was originally developed by Matthies [6]. Following his work, some minor variations and modifications helped improve its robustness and accuracy [8]. Related work using stereo images for localization can be found in [4], [7], and using a single omnidirectional camera in [1]. The key idea of the present method is to determine the change in position and attitude for two or more pairs of stereo images using maximum likelihood estimation. The basic steps of this method are described as follows.

*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 one of the image pairs, and pixels with the highest interest values are selected. To reduce the computational cost, a grid with cells smaller than the minimum distance between

features is superimposed on the left image. In each grid cell, only the one feature with strongest corner response 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 are selected evenly across 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. Under perfect conditions, the rays of the same feature from the left and right images should 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 eliminated from further processing.

Next we compute the **covariance** associated with each feature. Assume the stereo cameras are located at  $C_1(X_1, Y_1, Z_1)$  and  $C_2(X_2, Y_2, Z_2)$ . Let  $r_1$  and  $r_2$  be two unit vectors connecting  $C_1$  and  $C_2$  to the same feature in both images. Because of noise,  $r_1$  and  $r_2$  do not always intersect precisely at a point. But as long as the rays converge in front of the lens and are not exactly parallel, there will always be a shortest line segment connecting these two rays. Letting  $P_1$  and  $P_2$  be the endpoints of that line segment, we have the following constraints on the distance from each camera to that feature (where  $m_1 = |P_1 C_1|$ ,  $m_2 = |P_2 C_2|$ ):

$$P_1 = C_1 + r_1 m_1 \quad (1)$$

$$P_2 = C_2 + r_2 m_2 \quad (2)$$

Therefore we have

$$(P_2 - P_1)r_1 = (C_2 - C_1 + r_2 m_2 - r_1 m_1) \cdot r_1 \quad (3)$$

$$= 0$$

$$(P_2 - P_1)r_2 = (C_2 - C_1 + r_2 m_2 - r_1 m_1) \cdot r_2 \quad (4)$$

$$= 0$$

Then we have

$$m_1 = \frac{Br_1 - (B \cdot r_2)(r_1 \cdot r_2)}{1 - (r_1 \cdot r_2)^2} \quad (5)$$

$$m_2 = (r_1 \cdot r_2)m_1 - Br_2 \quad (6)$$

$$P = (P_1 + P_2)/2 \quad (7)$$

where  $B = C_2 - C_1$  is the stereo baseline, and  $m_1$  and  $m_2$  are functions of feature locations in both images whose partial derivatives are:

$$m'_1 = \quad (8)$$

$$\begin{aligned} & \{[B \cdot r'_1 - (B \cdot r'_2)(r_1 \cdot r_2) - \\ & (B \cdot r_2)(r'_1 \cdot r_2 + r_1 \cdot r'_2)] \\ & [1 - (r_1 \cdot r_2)^2] + \\ & 2[B \cdot r_1 - (B \cdot r_2)(r_1 \cdot r_2)] \\ & [(r_1 \cdot r_2)(r'_1 \cdot r_2 + r_1 \cdot r'_2)]\} \\ & / [1 - (r_1 \cdot r_2)^2]^2 \end{aligned}$$

$$m'_2 = (r_1 \cdot r_2)m'_1 + (r'_1 \cdot r_2 + r_1 \cdot r'_2)m_1 - Br'_2 \quad (9)$$

$$P' = (r'_1 m_1 + r_1 m'_1 + r'_2 m_2 + r_2 m'_2)/2 \quad (10)$$

Further note that the covariance of  $P$  is

$$\sum_P = P' \begin{bmatrix} \sum_l & 0 \\ 0 & \sum_r \end{bmatrix} P'^T \quad (11)$$

where  $P'$  is the Jacobian matrix or the first partial derivative of  $P$  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.

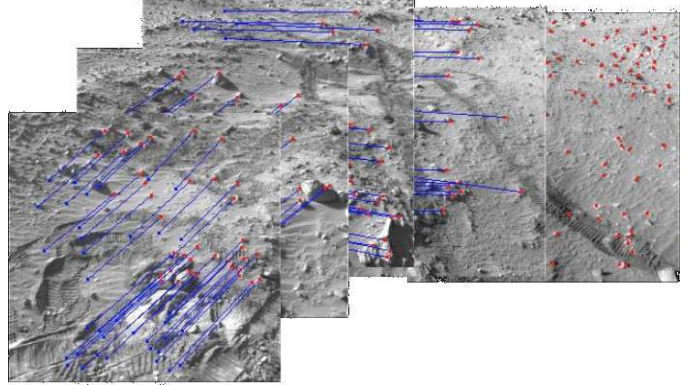


Fig. 3. Feature tracking occurs between every pair of images. In this view, several images from Spirit's Sol 178 drive have been positioned using Visual Odometry updates, and the features that were tracked have been overlaid.

*Feature Tracking* After the rover moves a certain distance, a second pair of stereo images is acquired. The features selected from the previous image can be projected into the second pair using the prior knowledge of the approximate motion provided by the onboard wheel odometry (see Figure 3). Then a correlation-based search reestablishes the 2D positions precisely in the second image pair. Stereo matching is then performed on these tracked features on the second pair to determine 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. Features whose initial and final 3D positions differ by too large an amount are filtered out.

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

The motion estimation is done in two steps. First, a less accurate motion is estimated by **Least-squares estimation**. The error residual is

$$e_j = P_{cj} - RP_{pj} - T \quad (12)$$

and the cost expression is

$$M(R, T) = \sum w_j e_j^T e_j \quad (13)$$

$$w_j = (\det(\sum p_j) + \det(\sum c_j))^{-1} \quad (14)$$

There is a closed form solution for this least squares estimation [6]. The advantage of this least squares method is that it is simple, fast and robust. Its disadvantage is that it is less accurate because it only takes the quality (the volume of the error ellipsoid) of the observations as a weight factor.

Because it is an inexpensive operation, we embed it within a **RANSAC** process to do **outlier removal**:

- 1) A small set of features (e.g. 6) is randomly selected and the motion is then estimated using the least squares estimation method.
- 2) All features from previous step are projected to the current image frame by the newly estimated motion. If the gap between a reprojected feature and its correspondent is less than a threshold (e.g. 0.5), 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.

The maximum likelihood motion estimation takes into account the 3D position difference and associated error models when estimating position. Let  $P_{pj}$  and  $P_{cj}$  be the observed feature positions prior to and after the current robot motion. Then we have

$$P_{cj} = RP_{pj} + T + e_j \quad (15)$$

where  $R$  and  $T$  are the rotation and translation of the robot and  $e_j$  is the combined error in the observed positions of  $j$ th features. In this estimation, 3 axis rotations  $\theta_R$  and translation  $T$  are directly determined by minimizing the summation in the exponents

$$\sum r_j^T W_j r_j \quad (16)$$

$$r_j = P_{cj} - RP_{pj} - T \quad (17)$$

where  $W_j$  is the inverse covariance matrix of  $e_j$ . The minimization of this nonlinear problem is done by linearization and an iterative process [6]. Two nice properties of the maximum-likelihood estimation make the algorithm powerful. First, it estimates the 3 axis rotations  $\theta_R$  directly so that it eliminates the error caused by rotation matrix estimation done by the least-squares estimation. Secondly, it

fully incorporates error models (the shape of the ellipsoid) into the estimation, which greatly improves the accuracy of the final motion estimate.

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 Site Frame components, the magnitude of the Roll, Pitch and Yaw updates, and the angular deviation from a purely downslope vector can all be restricted. Any update violating the active set of constraints is treated as a failure to converge; the number of acceptable failures is another optional constraint.

### III. GROUND-BASED VALIDATION

This visual odometry software has been tested on numerous rover platforms. The latest test was conducted on JPL's FIDO rover at the JPL Arroyo [3]. The FIDO rover 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 field of views (FOV) are 112 degrees horizontal by 84 degrees vertical and the baseline is about 12 cm. The test site was a rugged dry riverbed where the FIDO rover experienced a lot of slip, tilt, and roll during the test. The rover traversed approximately 8 meters, taking images about every 20 cm.

In order to evaluate Visual Odometry performance, high precision ground-truth data (position and attitude) was also collected using a Leica totalstation. By tracking a prism on the top of a rotating fixture, the rover's position and attitude were measured with high precision ( $< 3$  mm in position and  $< 0.1$  degree in attitude). The absolute position errors at the end of this run are 2.0, 5.3 and 10.1 cm in the X, Y, and Z directions respectively. The rotation error was less than 1.0 degrees.

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 (surveyor's theodolite) to measure the vehicle's 6-DOF motion estimate by tracking 3 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 on par with wheel odometry (within the design spec) on simple terrain, and better than plain wheel odometry in complex terrain.

Figure 4 shows the position estimation error that resulted from the most slip-inducing test run: a 2.45 meter rock-laden course driven in 35 cm steps. The straight line and light blue background represent the design goal of at most 10% error allowed in the position estimate. The dark curve represents the error that accrued onboard 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

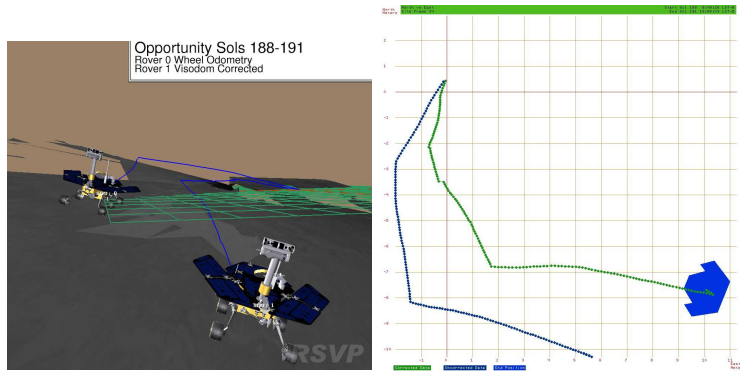


Fig. 5. Views of Opportunity's 19 meter drive from Sol 188 through Sol 191. The inside path shows the correct, Visual Odometry updated location. The outside path shows how its path *would* have been estimated from the IMU and wheel encoders alone.

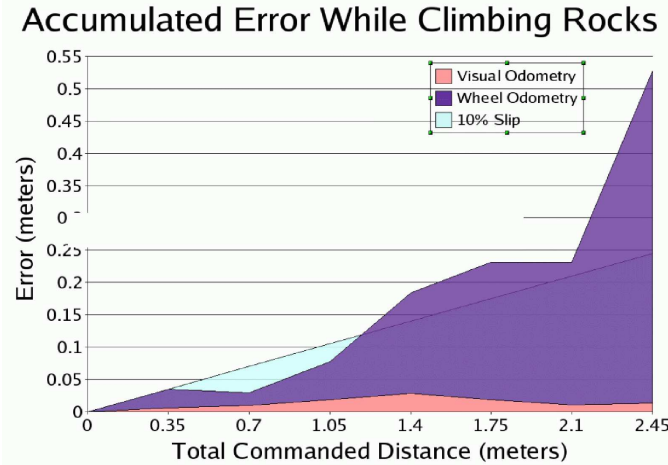


Fig. 4. Visual Odometry Error measured during a 2.45 meter drive using HAZCAMs on the MER Surface System Testbed Lite rover. The rover was driven over several large non-obstacle rocks, each less than 20 cm tall, in 35 cm steps. The vehicle was held in place during the final step, so the wheel odometry error for that step is artificially large, yet the Visual Odometry error remains small.

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.

#### IV. COMMANDING 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 [5], so all Visual Odometry drives were split into several small steps to ensure at least 60% overlap between adjacent images. During each step the rover was typically commanded to drive no more than 75cm 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.

Although Visual Odometry processing could have been beneficial during all rover motion, each step required two to three minutes of computation time on MER's 20 MHz RAD6000 CPU, and thus it was only commanded during relatively short drives (typically less than 15 meters) that occurred either on steep slopes (typically more than 10 degrees), or in situations where a wheel was being dragged (digging a trench, or conserving drive motor lifetime on Spirit's right front wheel). 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: so during the first year of operations from January 2004 through January 2005, 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 primarily attributable to either too large a motion (e.g. commanding a 40 degree turn in place which resulted in too little image overlap) or to a lack of features in the imaged terrain (but see *False Positives* below). 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 by the use of Visual Odometry. Science observations requiring precision pointing were often scheduled in the middle of drives, and Visual Odometry helped to ensure accurate data collection. The accuracy of driving in new or mixed-soil terrains was improved by re-pointing to the drive goal or recomputing the distance remaining to the goal after each step along the way. And sometimes vehicle safety was maintained by having the rover terminate a planned drive early, when it realized via Visual Odometry that it was making insufficient progress toward its goal, or away from an obstacle.

##### A. 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 so much of the 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.

The path predicted by wheel odometry alone can be quite different from the path actually taken. Figure 5 shows two views of the trajectory taken by Opportunity during Sols 188-191. The rover was driven uphill and across slope over a total actual distance of 19 meters, but wheel odometry underestimated the distance by 1.6 meters and failed to take the slip-induced elevation change into account. The outside path indicates the course as estimated solely by the wheel odometry subsystem, the inside path shows the Visual Odometry-corrected course plot that was actually generated onboard. The final positions differ by nearly 5 meters.

The earliest benefit from Visual Odometry came inside the 20 meter-diameter Eagle Crater, Opportunity's landing site. Much of the 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. But while those tables were found to work well for predicting purely upslope and crossslope 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 Panoramic camera (PANCAM) images of science targets taken after a drive would be pointed right on target (see the left-hand side of Figure 2).

But by far the most extensive use of Visual Odometry was made inside the 130 meter diameter Endurance Crater, where Opportunity lived from Sol 133 to Sol 312 (see the right-hand side of Figure 2). Except for a 12 meter approach and return at the lowest point (and hence lowest tilt) on Sols 201 and 203 and a 17 meter drive on sol 249, Visual Odometry was employed virtually continuously throughout. Had Visual Odometry not been available onboard, many more Sols would have been needed to approach targets, and fewer targets would have been achieved. In addition to improving target approach efficiency, Visual Odometry also proved crucial to maintaining vehicle safety.

From Sols 249 to 265 Opportunity kept finding itself near a 1 meter wide rock obstacle called Wopmay (see Figures 6 and 7). Although Wopmay was originally considered a primary science target, it also proved to be one of the most difficult obstacles to avoid. Wopmay was located downhill from a 17-20 degree downslope area comprised of loose sand and buried rocks. Several attempts to drive



Fig. 6. Wopmay, a 1 meter wide obstacle inside Endurance Crater

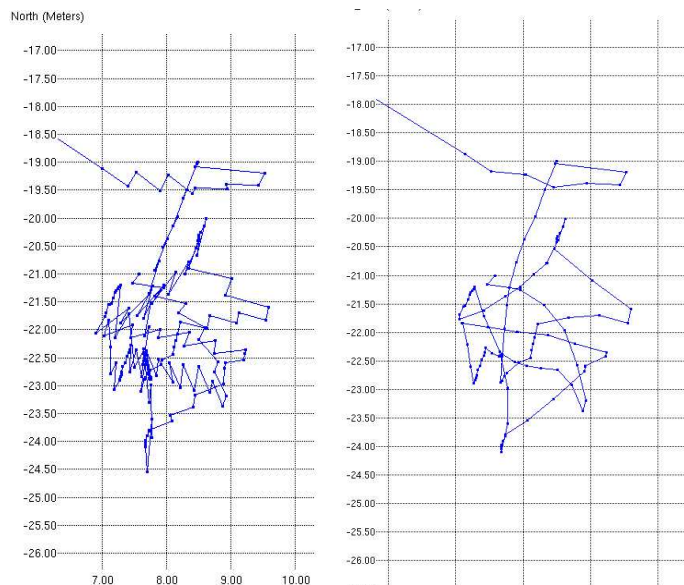


Fig. 7. Opportunity's 15 sol trajectory, first driving toward and then trying to get around or away from Wopmay. In the left plot, the "jumps" that point up to the right are the result of Visual Odometry processing adjusting the vehicle's position downslope. Visual Odometry only corrects the rover's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

around it were thwarted not only by higher-than-usual slip, but also by the unseen rocks buried just beneath the surface. Fortunately, the human-commanded sequences took into account the possibility that the rover might slip, and so Opportunity halted its planned drives prematurely (and *correctly*) when it realized that it was moving too close to Wopmay.

Visual Odometry enabled more precise approaches even 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

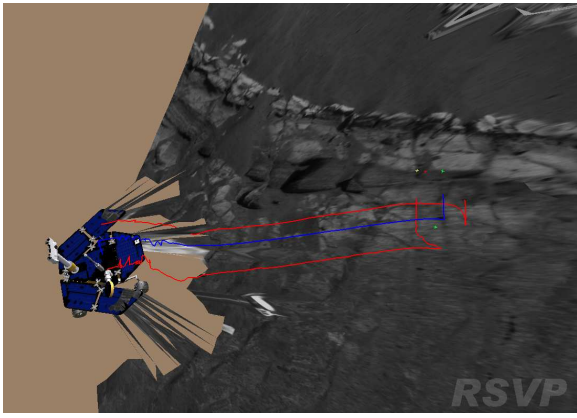


Fig. 8. Opportunity's planned 8.7 meter drive along a 20-24 degree slope.

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 perfectly located between the front wheels.

Visual Odometry results are summarized in Table I. As of March 2005, Opportunity has thrived for 394 sols. Visual Odometry was used more on Opportunity than Spirit, because Opportunity spent more of its first year on slippery surfaces. It has converged to a solution 95% (828/875) of the time it has been used.

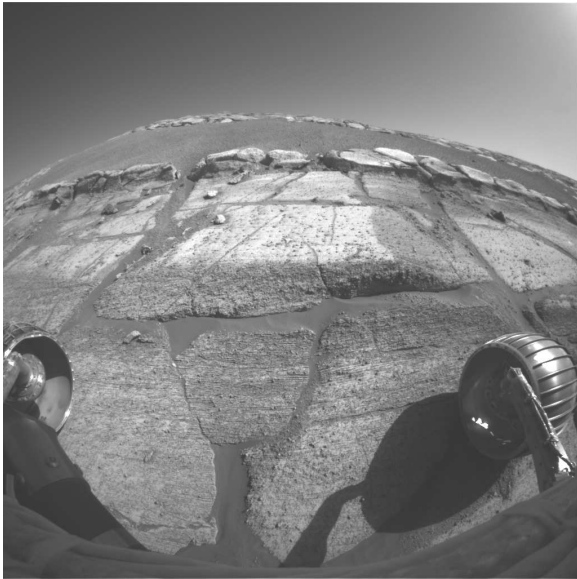


Fig. 9. After Opportunity's 8.7 meter slope drive, the goal is perfectly inside the IDD work volume.

### B. Gusev Crater: Spirit Rover

The terrain at Gusev crater is well suited for Visual Odometry processing. The rock abundances there matched

predicted distributions [2], 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, though they sometimes also had to avoid pointing the cameras at (the relatively infrequently occurring) sand dunes. As a result, Spirit's Visual Odometry software has performed admirably.

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.

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  $\pm$  4.4 degrees, counting 625 samples spanning an absolute range from 2 - 30 degrees.

Visual Odometry results are summarized in Table I. As of March 2005, Spirit has thrived for 414 sols. Visual Odometry was not used on Spirit until it had reached the Columbia Hills, nearly six months into its mission. It has converged to a solution 97% (590/609) of the time it has been used.

### C. False Positives

Although we had never seen Visual Odometry converge to an inaccurate solution during pre-launch 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. That meant that the overall 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 out-of-plane features and converge to a reasonable solution.

The only other instance of a false positive solution 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, many more features were found on the largest shape in the image; the rover's

TABLE I

RESULTS OF RUNNING VISUAL ODOMETRY ON MARS. EXPRESSIONS SUCH AS  $m \pm s$  INDICATE MEAN  $m$  AND STANDARD DEVIATION  $s$ .

	<b>Spirit</b>		<b>Opportunity</b>	
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 Processed	682	pairs	972	pairs
Successful (non-initial) Convergences	590	pairs	828	pairs
Iterations Needed (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	degrees	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
Absolute Tilt Range During Visodom	2 - 30	degrees	0.8 - 31	degrees

shadow. So even though forward drive progress was made, the onboard estimator assumed that the shadow (having more distinct 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, but since then human drivers have had to take the rover shadow into account whenever planning Visual Odometry drives.

## V. CONCLUSION

Visual Odometry has proven to be a highly effective tool for ensuring accurate science imaging, achieving difficult drive approaches, and maintaining vehicle safety while driving near obstacles on slopes. 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.

## VI. ACKNOWLEDGMENTS

Thanks to Jeff Biesiadecki for integrating this capability into the MER flight software Mobility Manager state machine, the entire Surface Navigation and Mobility testing team for testing and validating it in time to be included the mission, flight software lead Glenn Reeves for supporting the inclusion of this capability, 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 color-coded 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

- [1] Peter Ian Corke, Dennis Strelow, and Sanjiv Singh. Omnidirectional visual odometry for a planetary rover. In *Proceedings of IROS 2004*, 2004. [http://www.ri.cmu.edu/pubs/pub\\_4913.html](http://www.ri.cmu.edu/pubs/pub_4913.html).
- [2] M. Golombek and D. Rapp. 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, February 1997.
- [3] D.M. Helmick, Y. Cheng, S.I. Roumeliotis, D. Clouse, and L. Matthies. Path following using visual odometry for a mars rover in high-slip environments. In *IEEE Aerospace Conference*, Big Sky, Montana, USA, March 2004.
- [4] Simon Lacroix, Anthony Mallet, Raja Chatila, and Laurent Gallo. 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, June 1999. <ftp://ftp.laas.fr/pub/ria/simon/isairas99.ps.gz>.
- [5] J. N. Maki, J. F. Bell III, K. E. Herkenhoff, S. W. Squyres, A. Kiely, M. Klimesh, M. Schwochert, T. Litwin, R. Willson, A. Johnson, M. Maimone, E. Baumgartner, A. Collins, M. Wadsworth, S. T. Elliot, A. Dingizian, D. Brown, E. C. Hagerott, L. Scherr, R. Deen, D. Alexander, and J. Lorre. Mars exploration rover engineering cameras. *Journal of Geophysical Research*, 108(E12):12–1–12–24, December 2003. <http://www.agu.org/pubs/crossref/2003/2003JE002077.shtml>.
- [6] Larry Matthies. *Dynamic Stereo Vision*. PhD thesis, Carnegie Mellon University Computer Science Department, October 1989. CMU-CS-89-195.



- [7] David Nister, Oleg Naroditsky, and James Bergen. Visual odometry. In *Computer Vision and Pattern Recognition*, pages 652 – 659. IEEE Computer Society Press, June 2004.
- [8] Clark F. Olson, Larry H. Matthies, Marcel Schoppers, and Mark W. Maimone. Rover navigation using stereo ego-motion. *Robotics and Autonomous Systems*, 43(4):215–229, June 2003.