Visual Target Tracking on the Mars Exploration Rovers

Won S. Kim, Jeffrey J. Biesiadecki, and Khaled S. Ali
Jet Propulsion Laboratory, California Institute of Technology
Won.S.Kim@jpl.nasa.gov

Abstract

Visual Target Tracking (VTT) has been implemented in the new Mars Exploration Rover (MER) Flight Software (FSW) R9.2 release, which is now running on both Spirit and Opportunity rovers. Applying the normalized cross-correlation (NCC) algorithm with template image magnification and roll compensation on MER Navcam images, VTT tracks the target and enables the rover to approach the target within a few cm over a 10 m traverse. Each VTT update takes 1/2 to 1 minute on the rovers, 2-3 times faster than one Visual Odometry (Visodom) update. VTT is a key element to achieve a target approach and instrument placement over a 10-m run in a single sol in contrast to the original baseline of 3 sols. VTT has been integrated into the MER FSW so that it can operate with any combination of blind driving, Autonomous Navigation (Autonav) with hazard avoidance, and Visodom. VTT can either guide the rover towards the target or simply image the target as the rover drives by. Three recent VTT operational checkouts on Opportunity were all successful, tracking the selected target reliably within a few pixels.

1. Introduction

The Mars Exploration Rovers, Spirit and Opportunity, have been exploring Martian surface more than three years since they landed in January 2004, well exceeding the 90-day primary mission lifetime. After the deployment and extensive validation of Visual Target Tracking (VTT) on the research rovers, we proposed this algorithm as part of extended MER missions, and VTT was selected as one of the four technologies to be demonstrated on Mars. Since then, VTT has been successfully incorporated into the new MER FSW R9.2 and is currently being tested through operational checkouts on Mars.

Visual target tracking is directly related to feature tracking and visual servoing [1], [2], which is a well-established field in computer vision. Visual servoing uses visual feedback such as feature tracking to control a robot. Nowadays, many implementations can run in real-time or near real-time frame rate. For the Mars Exploration Rovers, however, the image capture and transfer rate with a 20-MHz CPU are rather slow. It takes tens of seconds to capture and transfer a pair of full-resolution (1024×1024 pixels) stereo images. Thus, it is impossible to implement real-time visual servoing on the Mars Exploration Rovers. An alternative approach is to drive the rover in short steps and perform visual target tracking with active mast camera pointing at only these discrete stops. In MER applications, the term “visual target tracking” seems appropriate since the target is selected by scientists and rover planners on Earth, and the update rate is rather slow, from 40 s to 1 minute per update.

Some of the very first versions of visual target tracking were demonstrated on the Marsokhod rover at Ames Research [3] and on the Rocky 7 rover at the Jet Propulsion Laboratory [4]. The Marsokhod tracker used the sign of the difference of Gaussian (SDOG) to match the target templates to new images. The Rocky 7 tracker used three-dimensional information from stereo images combined with intensity information for target tracking, which was used for both sample acquisition and instrument placement. A follow-on effort by both teams resulted in the first visual target tracker [5], [6] that was developed within the CLARAty (Coupled Layer Architecture for Robotic Autonomy) framework [7], [8] and delivered on the Rocky 8 rover for formal validation. The first version of the tracker included affine trackers at multiple image resolutions. Extensive validation [9], [10], [11], [12] indicated shortcomings of the affine tracker for applications where image changes are large between frames and initial knowledge of the pose has large uncertainties. This eventually led to the development of the normalized cross-correlation (NCC) with template image magnification, which was added in the second version of the visual target tracker [13]. Validation experiments indicated that the affine tracker had a limited matching range of up to only about 30 pixels at most and was less reliable than NCC. The affine tracker does iterative search by successive approximations to the solution, which could converge to a local minimum. By contrast, the NCC matcher does a brute-force search, increasing the search range virtually to the entire image area. NCC with template image magnification was most reliable, and was able to track any target points on natural rock surfaces.

During the infusion of VTT into the MER FSW, further improvements were made such as 1) template image roll compensation (the MER mast does not provide roll control), 2) auto exposure over a subframe image, 3) specification of target position in rover and site (world) frames [14] as well as target position update in site frame, 4) target loss detection and fault protection, 5) 10× speedup of NCC and point stereo, 6) integration with MER Autonav and Visodom, and 7) VTT command interface.

After presenting the concept of VTT operations in Section 2, we explain the detailed functional description in Section 3. Section 4 describes the VTT integration into MER FSW, and Section 5 presents the recent two VTT operational checkouts performed successfully on Opportunity.

2. Concept of operations

Visual target tracking (VTT) enables the rover to approach the designated target accurately within a few centimeters where the target could be initially up to 10 to 20 m away from the rover. In a typical operational scenario, scientists and rover planners first select a target from the downlinked Navcam images of the previous day, and thereafter generate and uplink a command sequence that specifies the VTT target position and subsequent VTT operations. The initial VTT target position can be specified by 1) 2D image coordinates on the left Navcam image with the PMA (Pancam Mast Assembly) azimuth/elevation angles, 2) 3D coordinates in rover frame, or 3) 3D coordinates in site frame. When the rover is commanded to autonomously go to a specified goal with VTT, the rover drives in small steps (10% of the target distance from the rover) until it reaches the goal, while a VTT update is performed at each stop, updating the drive goal position. The VTT update can also be issued as an individual command that does not move the rover, allowing target tracking during non-autonomous driving. The VTT update does the following.

1) After a rover move, estimate the target position relative to the new rover pose.
2) Compute the azimuth and elevation angles of the PMA mast so that the left Navcam points to the estimated target position at the center of the image.
3) Point the mast and capture Navcam stereo images.
4) Magnify and roll the target template image based on the target distance change and the camera model.
5) Find the target image point in the left Navcam image by using NCC with template magnification and roll.
6) Perform point stereo in the right Navcam image and update the target 3D position.
7) Save the new target template image for next iteration.

An example of a VTT test over a rough terrain using the MER Surface System Test Ted (SSTB) rover is illustrated in Figures 1 and 2. A target is selected from a full-frame (1024×1024 pixels) 1-bpp ICER-compressed image. During VTT updates, subframe images of 161×161 pixels are used. Note that the target image is magnified by a factor of 4 as the rover approaches the target from 6 m away to 1.5 m.

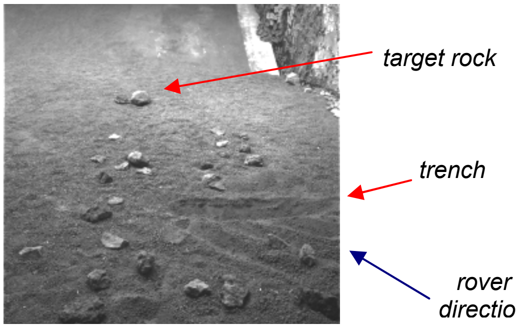

Figure 1. A target is selected at the center of the target rock from this initial full-frame image taken in the MER test bed.

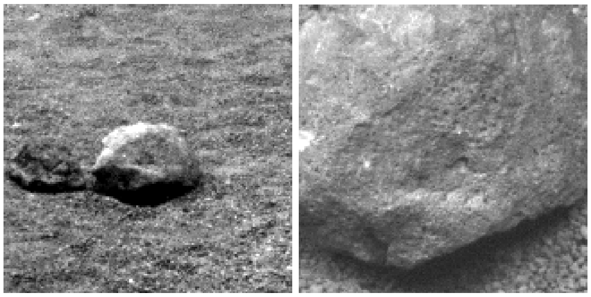

Figure 2. Initial and final 161×161 pixels subframe images for visual tracking over rough terrain: initially at 6 m away from target (left) and finally at 1.5 m away (right).

2.1. Functional diagram of visual tracker

Figure 3 shows a functional diagram that implements the concept of VTT operations above.

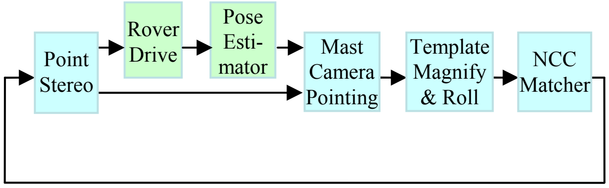

Figure 3. Functional diagram of the visual target tracker. Existing MER FSW provided rover drive and rover pose estimator.

2.2. Rover drive step size

Earlier VTT validation experiments [10], [11] indicated that NCC with template image magnification tracked 100% of the images tested over an 8-m straightforward run even when the rover step size is 20% of the target distance (a 20% change in target distance causes a 20% change in target image size). On the other hand, pure NCC without template image magnification tracked only 80% of the images tested even at smaller step size change of 5% the target distance. Further, the target azimuth angle change relative to the rover (rover path perpendicular to the target direction) was limited to 5 to 10° to maintain good tracking. Based on this experimental result, the rover step size for VTT is limited to 10% of the target distance. The 10% limit is sufficient to cover all directions of rover motion since this limits the viewing yaw angle change to 0.1 radian or 5.7°. The step size is 1 m at 10 m away and 20 cm at 2 m away.

2.3. Rover pose estimator

MER provides the rover pose (position and attitude) estimator [14] based on wheel odometry and IMU. This estimator is accurate on flat terrain but does not take into account rover slippage. MER also provides Visual Odometry (Visodom) [15] for more accurate rover pose estimation regardless of rover slippage. In normal VTT operations on low-slippage terrain, Visodom is not needed. If the rover is expected to experience large slippage, for example, on a steep slope, it is recommended to enable Visodom during VTT operations.

2.4. Mast camera pointing

In order to do mast camera pointing, we need to compute the target position relative to the new rover pose. The new rover pose after the rover move is provided by the MER rover pose estimator. The target position is the same as the selected initial target seed for the first VTT update. For subsequent VTT updates, the updated target position is given by the previous VTT update. Hence, the target position relative to the new rover pose can be determined by coordinate transformation. Once this is known, the PMA azimuth and elevation angles to point the left Navcam to the target can be computed based on PMA inverse kinematics and the left Navcam camera model. Existing MER FSW already provides such a function.

2.5. Subframe image capture

After the mast camera pointing, the Navcam stereo images are captured in the subframe mode [16]. The subframe (default 161×161 pixels for VTT) option is used since it does auto-exposure adjustment over a small subframe region rather than over the entire image area. This, for example, prevents the bright sky at the top of the image from darkening the terrain portion of the image inadvertently. Since the mast pointing centers the target in the left Navcam image, the target image on the right Navcam image is offset by the stereo disparity. Thus, to specify the subframe region for the right Navcam image, the stereo disparity in pixels is computed for a given target distance.

2.6. Template image magnification and roll

Based on the perspective projection imaging geometry, the magnification factor for the template image is computed as the ratio of the previous target distance from the camera to the current target distance after the rover move.

$$mag\_factor = \frac{target\_dist\_old}{target\_dist\_new}, \tag{1}$$

where the target distance is the vector norm from the left Navcam lens center **C** to the target 3D position **p**

$$target\_dist = |\mathbf{p} - \mathbf{C}|. \tag{2}$$

The previous VTT update provides the previous target distance by point stereo, whereas the new estimated target distance relative to the current rover frame is computed by using the rover pose estimator output after rover motion. When the rover moves closer to the target, the template image gets enlarged with a *mag_factor* greater than 1. When the rover moves away from the target, the template image gets reduced with a *mag_factor* less than 1.

Since the PMA provides only azimuth and elevation control, the roll compensation can only be done by software. To handle the image roll by software, an image roll algorithm is added prior to NCC so that the template image can be rotated by an estimated roll angle in addition to image magnification or shrinking. In the MER CAHVOR camera model [17], [18], **H**, **V**, and **A** axes are not mutually orthogonal. Let the mutually orthogonal unit vectors be denoted by $\mathbf{H}_x$ along x-axis, $\mathbf{V}_y$ along y-axis, and **A**.

$$h_s = |\mathbf{A} \times \mathbf{H}| , \tag{3}$$

$$h_c = \mathbf{A} \bullet \mathbf{H} , \tag{4}$$

$$\mathbf{H}_x = \frac{\mathbf{H} - h_c \mathbf{A}}{h_s} . \tag{5}$$

Given the target 3D position in previous update $\mathbf{p}_0$, we define an adjacent point $\mathbf{p}_{0\text{-adj}}$ along the $\mathbf{H}_x$ axis as

$$\mathbf{p}_{0\text{-}adj} = \mathbf{p}_0 + k\,\mathbf{H}_x , \tag{6}$$

where k is set to an arbitrarily small length of 0.05 m. Since $\mathbf{H}_x$ is along the image plane x-axis, the projection of the vector from $\mathbf{p}_0$ to $\mathbf{p}_{0\text{-adj}}$ on the image plane makes the zero roll angle. After the rover's move, using the rover pose estimator output, we can estimate the target and its adjacent positions in the new rover frame, $\mathbf{p}_1$ and $\mathbf{p}_{1\text{-adj}}$. The projection of the vector from these two points determines the roll angle of the new image relative to the previous image. Since the image y-axis is defined downward, a clockwise rotation yields a positive roll angle.

A large change in image roll angle can be observed during a VTT run when the rover traverses over a slope or when the rover goes by the target because the goal position is specified to be offset from the target position. The rover path and initial and final VTT images of an example run with a goal offset of (1 m, 0 m) from the target are shown in Figures 4 and 5, respectively. The template roll compensation made this run successful.

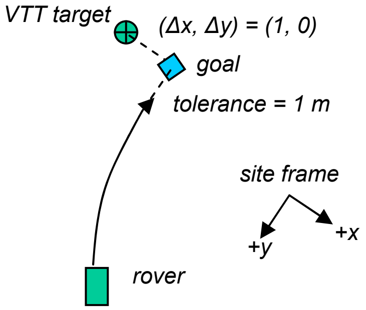

Figure 4. The rover approaches the goal within 1 m (tolerance) while tracking the target, where the goal is (1 m, 0 m) offset from the target.

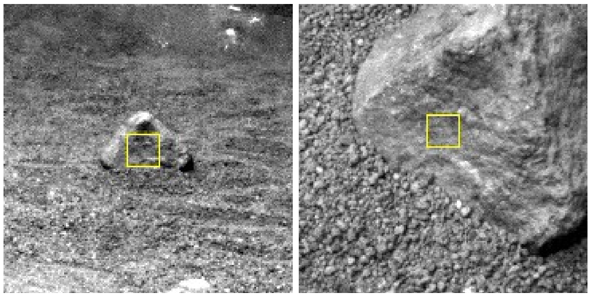

Figure 5. Subframe images of 161×161 pixels, taken at full resolution during the VTT run of Figure 4, starting from 6.8 m away from target (left) down to 1.8 m (right). The total roll change between the initial and final images was about 30º.

2.7. Template Image Size

Since earlier validation results [10], [11] indicated a template window size of 21×21 pixels performed best, this size is used as the default value. The VTT module defines the template image storage size to be 41×41 pixels, from which the next template image is extracted. The storage size must be larger than the actual template image size to handle 1) template image roll and 2) template image shrinking when the rover moves away from the target.

2.8. Normalized cross-correlation with pyramidal image reduction

The normalized cross-correlation is computed as

$$N = \frac{\sum (I - \bar{I})(J - \bar{J})}{\sqrt{\sum (I - \bar{I})^2 \sum (J - \bar{J})^2}} \tag{7}$$

where $\bar{I}$ and $\bar{J}$ are the pixel intensity averages of the template image and the matching image window of the template image size. The range of N is $-1 \le N \le 1$. The normalized cross-correlation value is computed for every possible matching window within the image search area, and the image window location that yields the maximum correlation is selected as the best match. Since it is brute-force search, the search range can be as large as desired unlike the affine matching by iterative search. NCC is a normalized score and thus less sensitive to lighting changes.

A pyramidal image reduction by 2×2 image down-sampling reduces the NCC computational time dramatically. NCC matching is first performed at half resolution between the 2×2 down-sampled template and image. Thereafter, NCC matching is performed at full resolution over a very small search area by using the best match at half resolution as the initial seed. Table 1 compares the execution times and memory usages of the two implementations, running on the 20-MHz RAD6000 processor of the MER SSTB rover. The execution time was measured in finding the best match of a 21×21 template in a 161×161 image. Coarse NCC with 2×2 down-sampling followed by fine NCC at full resolution improved the computational speed by a factor of 10.

Table 1. Execution time and memory usage.

Algorithm	Execution time (sec)	Memory usage (Kbytes)
NCC full res.	45	5
NCC 2×2 followed by full res	4	37

2.9. Point stereo and target position update

After the target image point in the left Navcam is determined by using NCC matching, we need to find the corresponding point in the right Navcam image. Since the full-frame stereo vision takes a long time, we use a point stereo that searches the corresponding right image point only along the epipolar line with a few pixels (default 2 pixels) of vertical offset tolerance [6]. Instead of searching along the entire 1024 pixels, we further limit the stereo search segment range from half the subframe image width to zero. For instance, if the subframe image width is 161 pixels, the epipolar search segment is only about 80 pixels long. Once the corresponding target image point in the right image is found, the 3D position of the target is computed by stereo triangulation. The target position is updated both in rover frame and in site frame. This point stereo computation took about 1 to 2 seconds on the 20-MHz RAD6000 processor.

2.10. Target position update accuracy

When scientists and rover planners designate a 10-m-away target on Navcam images, the target designation error due to the stereo range 3-σ error is as much as 40 cm by considering that the subpixel-interpolated stereo disparity 3-σ error is about 1 pixel. This implies that even if the rover has a perfect rover pose estimator, without VTT the final target position could be as much as 40 cm off (3-σ) when the rover reaches the target rock from about 10 m away. By contrast, with VTT the final target position will be within a few centimeters error when the rover reaches the target within 2 m. As the rover gets closer to the target, the updated target position becomes more accurate since VTT tracks the target and the point stereo yields a more accurate target position with a smaller stereo range error. Another major source of the rover approach positioning error without VTT is the rover pose estimator error. For instance, a 1-2% visual odometry (Visodom) estimation error over a 10 m traverse corresponds to a 10-20 cm error. A wheel odometry and IMU-based rover pose estimator is in general less accurate and does not take into account the soil slippage. Unlike the rover pose estimators including Visodom, VTT employs a closed loop control around the designated target enabling the rover to compensate for pose error and track the target within a few pixels, resulting in only 0.1-0.5% error over 10 m.

2.11. Target loss detection and fault protection

Although it is very rare, it is possible that the target can be lost during tracking due to conditions such as occlusion, shadow, and no features. Four main criteria for the declaration of target loss during VTT update are listed here.

1) NCC score < 0.7.
2) point stereo match score < 0.7.
3) target distance error > target distance tolerance.
4) target position change > target position change tolerance.

The first and second criteria are to avoid potential false matches during image-based feature matching by eliminating the cases when its correlation score is too low. The third criterion is to check the target distance error primarily along the range direction of the camera, and its computation is done in rover frame. The fourth criterion is to check the target position change relative to the initial target position, and its computation is done in site frame. As the rover approached closer to the target, the target distance tolerance used in the third criterion decreases, while the target position change tolerance used in the fourth criterion increases. When the VTT target is lost during tracking, the system tries to recover the lost target up to N times, where the default value is currently set to 2. Once the VTT target is lost N consecutive times, VTT declares FAILURE and remains in that state until the target is cleared. The drive can optionally continue towards the target position with a larger goal tolerance.

3. Integration with MER FSW

3.1. VTT module interface to MER FSW

The MER flight software [19] consists of many modules. Many of these modules start their own independent task (or tasks) and monitor message queues to communicate with other tasks through message passing. Some modules, however, simply provide library functions that are called and managed by other modules. The Autonomous Navigation (NAV) module [20] is such an example. Likewise, the VTT module also contains a collection of visual tracking functions but does not have its own task context. The Mobility Manager (MOBM) task [21], while communicating with other tasks such as Image (IMG), Pancam Mast Assembly (PMA), Rover Drive (DRIVE), Command (CMD) tasks via message passing, handles queue messages including commands to perform visual tracking and calls appropriate VTT and NAV module functions when needed. VTT has its own queue for handling parameter setting commands. This queue is managed by the Background Command Processing (BCP) task. Similar to the NAV module, the VTT module ported from CLARAty is mostly written in C++. Global variables and public functions are defined with a C interface, so that MER C programs can call VTT public functions.

3.2. VTT integrated with MER Autonav and Visodom

VTT is fully integrated into the existing MER FSW. VTT can run in any combinations of rover driving: blind driving (with wheel-odometry and IMU-based estimator), Visodom (visual odometry), and Autonav (autonomous navigation with hazard avoidance). Figure 6 illustrates the details of one cycle of the MOBM loop. If visual odometry is enabled, this is done first. Then, if not blind drive mode, the hazard avoidance map is updated and path selection for autonomous navigation is performed. Finally, the rover is driven a single step. When VTT is enabled, the VTT routine is inserted right after Visodom and right before Autonav. The VTT routine specifies the camera pointing and performs the VTT update. The Autonav goal position may optionally be assigned to the new VTT target position. The actual image capture is done by the IMG module.

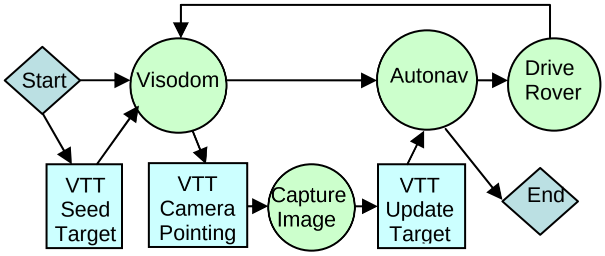

Figure 6. Functional flow of VTT integrated into MER MOBM/NAV FSW. The upper portion shows the existing MOBM/NAV functional flow for one cycle of rover motion with Autonav and Visodom when enabled. In the lower portion, VTT functions are inserted right after Visodom and right before Autonav. VTT module functions (square boxes) are additions to the existing MER FSW (circles).

3.3. Timing

Each VTT update takes about 0.5 to 1 minute on the Mars Exploration Rovers. The NCC with image magnification and roll and the point stereo take only 4 s and 1 s, respectively. The majority of the time is spent on Navcam stereo image subframe capture with auto exposure and post processing. Since each Visodom update takes about 1.5 to 2 minutes, the VTT update is about 2 to 3 times faster. So VTT is not only accurate in target approach but also relatively fast.

3.4. Regression test

A total of 14 regression tests including VTT parameter setting, VTT with combinations of blind driving/Visodom/Autonav, VTT on slope, and fault protection were performed. All tests completed successfully.

3.5. Operational guideline

There are a few operational guidelines. First, the rover should not be moved after taking Navcam images for VTT target selection the next day. The accuracy of the VTT target position will deteriorate as the rover moves without VTT. Second, the operator should avoid selecting a target on the rock boundary (within 10 pixels). A target at the rock boundary is susceptible to background change, particularly when there are other rocks adjacent. It is fine if the adjacent background is just a flat surface with no conspicuous texture. Third, the operator should avoid potential rover shadow cast over the target. Thus, around noon is best. The operator should avoid targets in the west direction early morning, and in the east direction late afternoon. In general the rovers are not moved in early or late hours by considering the battery energy.

4. Operational checkouts on Mars

Three operational checkouts for visual tracking were performed on Opportunity roaming along the rim of the Victoria Crater, and all three were successful. In the first checkout performed on sol-992 (November 8, 2006), VTT was instructed to only track the target without controlling the rover's movement. A small 7-cm wide rock located about 4 m away from the rover was chosen for the target (Figure 7). The rover path on that day happened to go around the target in 7 steps of 42.5 or 55 cm each with the target viewing azimuth angle change of a total 53°, while keeping the target distance at about 4 m. Despite the target azimuth angle change exceeding the usual 0.1 radian limit (10% of the target distance in any direction), VTT tracked over all 8 images successfully (Figure 8) with healthy NCC scores from 0.79 to 0.94.

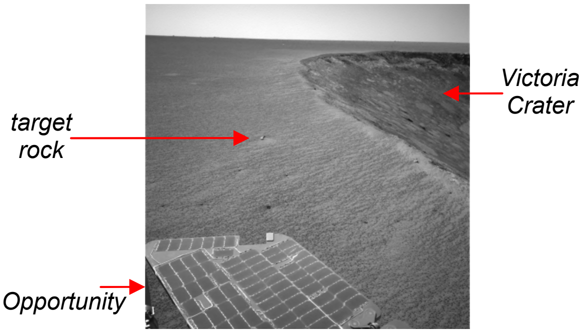

Figure 7. A target was selected at the center of the target rock from this initial full-frame Navcam image (45°×45° field of view).

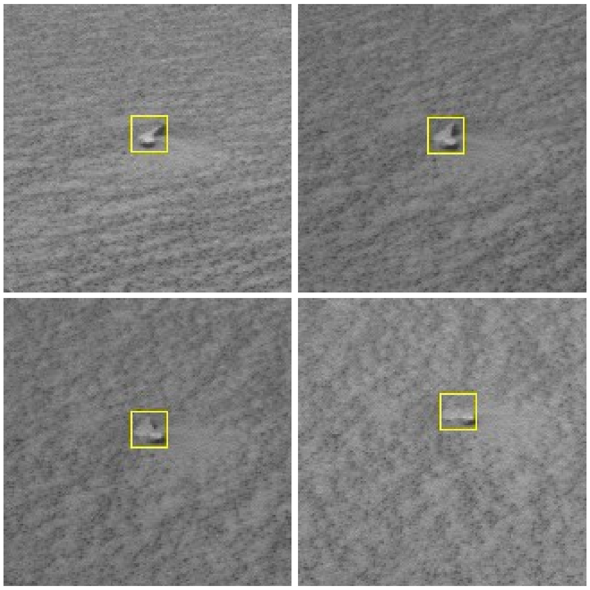

Figure 8. Operational checkout #1. Subframe images of 161×161 pixels (7°×7° field of view), taken during the VTT run, with the target window overlay: (top left) dist=4.00 m, az=-3.4°, (top right) dist=3.67 m, az=17.5°, (bottom left) dist=3.73 m, az=34.7°, (bottom right) dist=3.68 m, az=50.1°.

In the second checkout of sol-1100 (February 26, 2007), the Opportunity drove autonomously from about 10 m to within 2 m of a target rock in 15 steps, while doing VTT using Navcam subframe images at each stop to update the drive goal. Pancam images of the target were also taken at a couple of locations during the drive. A 12-cm wide rock located about 10 m away from the rover was chosen for the target. VTT tracked well over all 18 images (Figure 9). Note that the target rock image is magnified by 5.

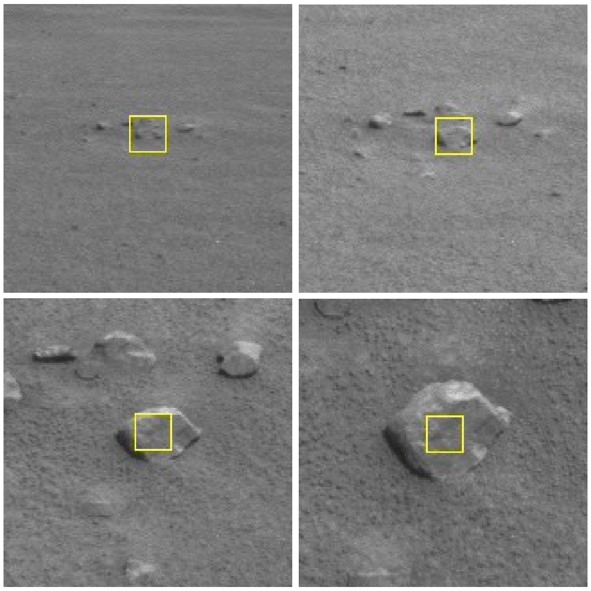

Figure 9. Operational checkout #2. Subframe images (161×161 pixels) with the target window overlay: (top left) dist=9.65 m, (top right) dist=6.11 m, (bottom left) dist=3.05 m, and (bottom right) dist=1.92 m.

The third and final VTT checkout ran successfully on Opportunity on Sol 1194 (June 2, 2007). This checkout involved using VTT with Visodom over 2.1 m and thereafter using VTT with Autonav over 1.4 m. Both combinations ran successfully. For Autonav, two pairs of Navcam stereo images (left and right 45° wedges) were used to build a sufficiently wide terrain map. It took 1/2 to 1 minute for a VTT update, 1 to 2 minutes for a Visodom update, and about 5 minutes for a 2-wedge Autonav update.

VTT was also used in the final AutoPlace checkout, which ran successfully on Opportunity on Sol 1216 (June 23, 2007). This combined VTT and Autoplace checkout demonstrated an exciting "go-and-touch" or single-sol instrument placement capability on Mars.

5. Conclusion

As part of the MER extended missions, we integrated Visual Target Tracking (VTT) into the new MER FSW release, running on both Spirit and Opportunity. VTT is shown to be accurate and relatively fast. All planned operational checkouts were conducted successfully, and we expect VTT will be used more often on Mars.

6. Acknowledgments

This work was performed at the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration. This work is supported by NASA Mars Technology Program (MTP) and Mars Exploration Rover (MER) Program. Prior to mission infusion, Issa Nesnas, Max Bajracharya, Richard Madison, and Won Kim were involved with CLARAty VTT development and deliveries, and Won Kim, Robert Steele and Adnan Ansar performed VTT validation. The authors would like to thank Mark Maimone for valuable suggestions during the VTT integration into MER and Richard Volpe for suggesting the template image roll.

10. References

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

[2] D. Lowe, "Distinctive image features from scale-invariant keypoints," International Journal of Computer Vision, vol. 60, no. 2, pp. 91–110, 2004.

[3] D. Wettergreen, H. Thomas, M. Bualat, "Initial Results from Vision-based control of the Marsokhod Rover," IEEE International Conference on Intelligent Systems, p1377-1382, Grenoble, France, Sep.1997.

[4] I. A. Nesnas, M. Maimone, H. Das, "Autonomous Vision-Based Manipulation from a Rover Platform," IEEE Symposium on Computational Intelligence in Robotics & Automation, Monterey, CA, Nov. 1999.

[5] I. Nesnas, M. Bajracharya, R. Madison, E. Bandari, C. Kunz, M. Deans, M. Bualat, "Visual Target Tracking for Rover-based Planetary Exploration," IEEE Aerospace Conference, Big Sky, Montana, 2004.

[6] M. Bajracharya, A. Diaz-Calderon, M. Robinson, M. Powell, "Target Tracking, Approach, and Camera Handoff for Automated Instrument Placement," IEEE Aerospace Conference, Big Sky, Montana, Mar. 2005.

[7] I.A. Nesnas, R. Simmons, D. Gaines, C. Kunz, A. Diaz-Calderon, T. Estlin, R. Madison, J. Guineau, M. McHenry, I. Shu, and D. Apfelbaum, "CLARAty: Challenges and Steps Toward Reusable Robotic Software," International Journal of Advanced Robotic Systems, Vol. 3, No. 1, pp. 023-030, 2006.

[8] R. Volpe, I.A.D. Nesnas, T. Estlin, D. Mutz, R. Petras, H. Das, "CLARAty: Coupled Layer Architecture for Robotic Autonomy," Technical Report D-19975, December 2000.

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

[10] W. S. Kim, R. D. Steele, Adnan I. Ansar, Danielle E. Ator, Shawn S. Catron, Visual Target Tracking Technology Validation Report [JPL Internal Only], JPL D-33416, Jan. 2006.

[11] W. S. Kim, R. D. Steele, A. I. Ansar, K. Ali, I. Nesnas, "Rover-Based Visual Target Tracking Validation and Mission Infusion," AIAA Conf., Space 2005, Aug. 2005.

[12] W. S. Kim, A. Ansar, R. Steele, "Rover Mast Calibration, Exact Camera Pointing, and Camera Handoff for Visual Target Tracking," Int. Conf. on Automation and Robotics (ICAR), Jul. 2005.

[13] R. Madison, I. Nesnas, "CLARAty Delivery of the initial 2D/3D Visual Tracking," Jun. 2004.

[14] K. Ali, C. Vanelli, J. Biesiadecki, M. Maimone, Y. Cheng, A. San Martin, J. Alexander, "Attitude and Position Estimation on the Mars Exploration Rovers", IEEE Conference on Systems, Man and Cybernetics, Waikoloa, Hawaii, Oct. 2005.

[15] Y. Cheng, M. Maimone, L. Matthies, "Visual Odometry on the Mars Exploration Rovers", IEEE Conference on Systems, Man and Cybernetics, Waikoloa, Hawaii, Oct. 2005.

[16] T. Litwin, J. Maki, "Imaging Services Flight Software on the Mars Exploration Rovers", IEEE Conference on Systems, Man and Cybernetics, Waikoloa, Hawaii, Oct. 2005.

[17] D. Gennery, Camera calibration including lens distortion, JPL D-8580, Jet Propulsion Laboratory, 1991.

[18] D. Gennery, "Generalized Camera Calibration Including Fish-Eye Lenses," Int. J. Computer Vision, vol. 68, no. 3, pp. 239-266, Jul. 2006.

[19] G. Reeves, "Mars Exploration Rovers Flight Software", IEEE Conference on Systems, Man and Cybernetics, Waikoloa, Hawaii, Oct. 2005.

[20] Mark W. Maimone, P. Chris Leger, Jeffrey J. Biesiadecki, "Overview of the Mars Exploration Rovers' Autonomous Mobility and Vision Capabilities," IEEE Int. Conf. on Robotics and Automation (ICRA) Space Robotics Workshop, Roma, Italy, Apr.2007.

[21] J. J. Biesiadecki M. W. Maimone, "The Mars Exploration Rover Surface Mobility Flight Software: Driving Ambition," IEEE Aerospace Conference, Big Sky, Montana, Mar. 2006.