IEEE International Conference on Intelligent Robots
and Systems, Grenoble, September 1997

Initial Results from Vision-based Control of the Ames Marsokhod Rover

David Wettergreen, Hans Thomas, and Maria Bualat
Intelligent Mechanisms Group
NASA Ames Research Center, MS 269-3
Moffett Field, CA 94035-1000 USA

Abstract

A terrestrial geologist investigates an area by systematically moving among and inspecting surface features, such as outcrops, boulders, contacts, and faults. A planetary geologist must explore remotely and use a robot to approach and image surface features. To date, position-based control has been developed to accomplish this task. This method requires an accurate estimate of the feature position, and frequent update of the robot's position. In practice this is error prone, since it relies on interpolation and continuous integration of data from inertial or odometric sensors or other position determination techniques.

The development of vision-based control of robot manipulators suggests an alternative approach for mobile robots. We have developed a vision-based control system that enables our Marsokhod mobile robot to drive autonomously to within sampling distance of a visually designated natural feature. This system utilizes a robust correlation technique based on matching the sign of the difference of the Gaussian of images. We will describe our system and our initial results using it during a field experiment in the Painted Desert of Arizona.

1 Introduction

Local inspection of remote planetary surfaces is a key part of understanding the geological processes at work in our Solar System. Upcoming missions, such as the Mars Pathfinder mission, will offer planetary scientists their first opportunities to use a mobile robot to make close observations of surface features, and to help answer long-standing questions regarding planetary development. In order to make these observations, scientists must have the capability to maneuver the rover to within centimeters of the feature of interest. Many of the control solutions pursued to date are position-based in nature: the position of the feature must be estimated, the position of the rover must be continuously estimated as it moves, and some control process must act to close the distance between the rover and the feature. This technique is in practice error-prone, since it usually involves some form of dead-reckoning, or integration of motion along various headings over time, to calculate rover position. Even with good odometric and inertial sensors and sophisticated filtering, errors grow rapidly.

A more direct approach to rover navigation is suggested by emerging work in vision-based control of fixed-base manipulators. (For survey see [1].) Since the given geologic feature is of primary interest, not its exact location, closing the rover's control loop by visually servoing offers a more direct means of navigating the vehicle. This technique avoids the problems of estimating the feature position accurately, as well as the complexity of maintaining an accurate running estimate of the rover's motion.

We have developed a robust image correlator based on binary correlation of the sign of the difference of Gaussian of an image. This correlator allows us to track a feature from frame to frame as the robot moves, as well as perform stereo correlation to estimate feature range. The correlator has been integrated into the control loop of our Marsokhod robot (Figure 1) to control both the motion of the vehicle and its pan-tilt camera head. In this paper, we will describe the implementation of the correlation algorithm and how it integrates into the control of both the pan-tilt head and mobile robot. We will also describe our initial results using the correlator to autonomously navigate to geologic features in an unstructured outdoor testsite in the Painted Desert of Arizona.

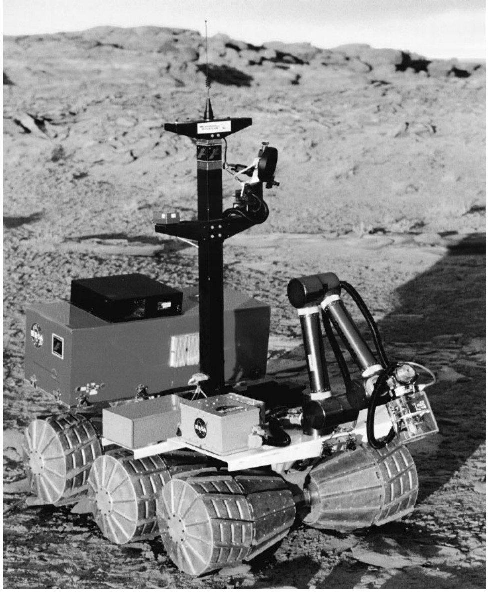


Figure 1: Ames Marsokhod rover at the Painted Desert field site

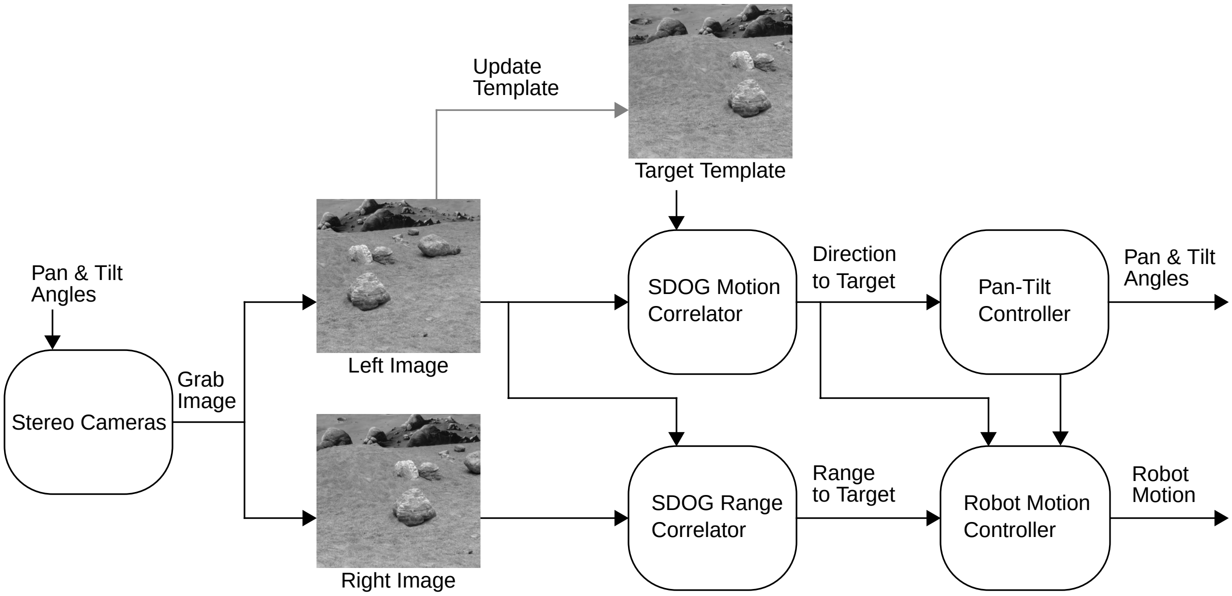


Figure 2: Overview of vision-based control of the Marsokhod

2 Marsokhod rover

The Ames Marsokhod rover is a prototype planetary rover being developed jointly by Russia and America for future Mars exploration. The rover is approximately 100 cm wide, 150 cm long, and 30 kg unloaded mass. The chassis consists of three pairs of independently driven titanium wheels, joined together by a three degree-of-freedom passively articulated chassis. Two of these degrees-of-freedom allow the segments to roll relative to one another, while the third allows the segments to pitch relative to one another. This design allows the rover to conform passively to very rugged terrain. Terrainability is further increased by mounting the rover's battery packs inside the wheels, lowering the center of mass of the vehicle to within 30cm of the ground.

The imaging hardware used in this work consists of a monochrome stereo pair of cameras mounted on a pan-tilt base, giving the cameras a 360° pan range and a 120° tilt range. The pan-tilt base is mounted on a mast about 130 cm above the center axle of the rover, and is offset to the left-hand side of the vehicle by about 30 cm. The mast is supported by an averaging linkage such that it always bisects the angle formed between the front and rear segments; this provides passive pitch stability to the imaging hardware.

The pan-tilt device, which can be seen on the lower crosspiece of the mast in Figure 1, is capable of velocities as high as 200° per second, is accurate to 3 arc minutes in both pan and tilt, with minimal backlash.

The stereo pair are very inexpensive “pinhole” cameras with a 28° horizontal field-of-view, 26° vertical field-of-view and a baseline of approximately 25 cm. The images from these cameras are digitized using a frame grabber, and all subsequent processing is performed by an on-board 68060-based processor board.

3 Vision-based control of Marsokhod

The vision-based control scheme we have developed for the Marsokhod is diagrammed in Figure 2. Input imagery comes from a stereo pair of cameras mounted atop a computer controlled pan-tilt head, which is itself mounted on the rover mast. Outputs from the control system consist of angle commands to the pan-tilt head, and steering and velocity commands to the base.

After a human operator designates a target feature, two control loops are activated to drive the robot to the feature. A gaze fixation loop correlates between previous and current images from one of the cameras, controlling the pan-tilt head to keep the target feature centered in the camera's field-of-view. A robot motion control loop correlates between left and right images, and uses the stereo range data in conjunction with the bearing data from the fixation loop to keep the vehicle driving to the feature. This control loop also halts the vehicle when the feature is within the desired distance for geologic examination.

3.1 Correlating with a sign-based operator

Both the gaze fixation loop and the vehicle control loop use the same binary correlation algorithm, which was first described in [2]. This algorithm exploits the invariance of the sign of the zero crossing in the Laplacian of the Gaussian (LOG) of an image. Even in the presence of noise and image intensity shifts, this sign information is stable.[2] Binary correlation also offers simpler implementation over other schemes, such as sum-of-squared differences (SSD) [3][4][5] and frequency domain matching[6], since it uses logical rather than arithmetic operations to match the binary sign information, and hence operates on several pixels at once. This allows adequate performance on relatively slow, general purpose computing hardware found on spacecraft.

Input images are subsampled to a resolution of 128x120, then processed using a difference of Gaussian (DOG) operator. This operator offers many of the same stability properties of the LOG operator, but is far simpler to implement on our hardware. We maintain sixteen bit precision in computing this Gaussian, which allows us to use up to a 16 pixel wide Gaussian in computing the DOG image. By selecting different Gaussian sizes and differences, the overall sharpness of the filter can be tuned to match to input images. In our experiments, 10x10 and 14x14 windows were empirically selected to achieve best correlation for natural terrain features.

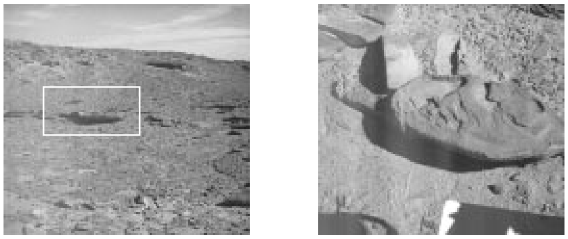
The DOG image is then binarized based on its sign information, and the resulting single bit pixels are packed into 32 bit words (the native word size of our CPU). This packed, binary image is then processed by the correlator, which matches a small window of another image (called the template), either from a previous frame in the case of tracking, or a stereo frame in the case of ranging. Because we need to maintain 32 bit alignment on our processor, we actually store 32 separate copies of the template, each shifted by one bit relative to the other; this allows us to perform shifting and masking once per image.

A logical exclusive OR (XOR) operation is used to correlate the template with the input image; matching pixels will always give a value of zero, while non-matching pixels will give a value of one. A lookup table is then used to count the number of matched pixels. Our current correlator operates with a fixed, 32x32 pixel template and can perform a correlation in about 40 msecs. By performing correlation of the template over the entire image, the correlator locates the peak where the best match occurred. The distance from center indicates pixel disparity and thus either heading or range to the target.

3.2 Correlating images to track a feature

We use a sequence of images from the left camera of the mast-mounted stereo pair to track the feature. The target is centered within the image from this camera and a target template is extracted. The target, and specifically the template, must exhibit sufficient texture to be distinctive from its surroundings[7]. We have found that even though there are no structured targets, natural terrain has sufficient features of an appropriate scale. The motion correlator matches this template with subsequent images taken as the robot advances. Using a 66 MHz 68060 processor, tracking can be performed at 2 Hz.

The appearance of the target can change drastically as the rover drives toward it; for example, examine the initial and final appearance of this rock ledge (Figure 3) which was successfully reached. The greatest change in appearance occurs when the robot nears the target, within two or three meters. At longer distances, the image to image change in appearance, and pixel correlation, is slight. Simply updating the template every correlation cycle would seem to solve this aspect change problem, but leads to the problem of small tracking errors being integrated each time the template is updated. In the worst case, this would cause the correlator to “slide” off the feature of interest. By using the same template for several correlation cycles, the effect of accumulated error can be reduced. We have found empirically that updating the template every five correlation cycles, or about every 15cm of vehicle motion, is sufficient to handle appearance change without suffering from excessive accumulated correlation error.



Initial image

Final image

Figure 3: Initial image with final field-of-view boxed

3.3 Correlating images to estimate range

Range to a feature is estimated by correlating between left and right stereo images. The range correlator extracts a template containing the feature from the left image, and correlates it against the right image to determine the disparity. This disparity is then converted to an absolute range based on a function determined in a calibration procedure.

An advantage of our approach is that a rough calibration is sufficient. The stereo cameras are mounted on a beam of imprecisely known baseline (approximately 25cm) and pointed so that distant (farther than 50m) targets have zero pixel disparity. Relative roll about the optical axis is minimized by maximizing the distant target correlation. We calibrated the range estimate by measuring the pixel disparity for twelve targets placed at various distances from the cameras. In general, disparity is linearly related to the inverse range.[4] If we fit such a function to the disparity and target distance data, we obtain a function which converts between pixel disparity and range. (Figure 4)

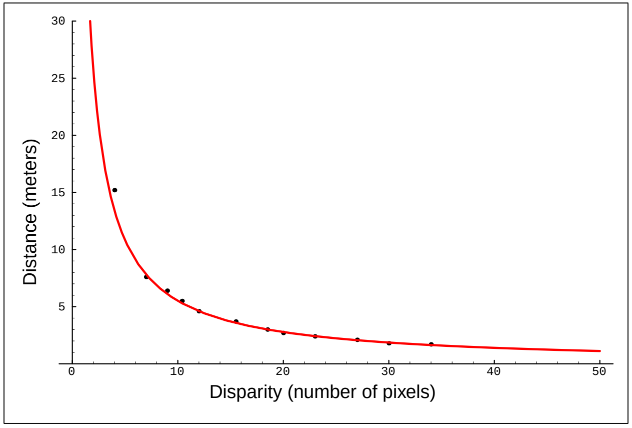


Figure 4: Plot of pixel disparity versus distance from camera

In practice, the calibration data sometimes does not follow such a linear relationship, owing to image distortions and noise. We have also used a fourth degree polynomial to calculate range from disparity; this polynomial was fitted numerically to the data.

3.4 Controlling the pan-tilt and robot motion

The correlator determines the pan and tilt offsets of the feature relative to the center of the image. These offsets are converted to absolute angles and used by the pan-tilt controller to fix the gaze of the cameras at the feature, regardless of base motion. This gaze fixing is important because the Marsokhod, like many outdoor mobile robots, has a rigid suspension and will pitch, yaw, and roll in response to terrain disturbances. In practice, we have had to add a small amount of deadband to this gaze-holding loop, since single-pixel errors make the pan-tilt head move unnecessarily.

The pan angle calculated in the pan-tilt controller is combined with the range estimates from the range correlator to command robot motion. The robot is steered left or right to maintain a desired pan offset angle; this offset allows the rover to closely approach a feature without self-occluding. As pan angle increases, the robot increases rotation and slows, at the extreme, turning in place.

The tilt angle and range estimate are used as cues to tell the rover when it has reached the feature; since most features are well below the cameras, tilt angle will drop as the feature is neared. Range data at long ranges is rather noisy, owing to the aggressive sub-sampling done on the images, but at short ranges is reasonably accurate. Range data is calculated every 2.5 seconds, or about every 15 cm of motion. Thresholds on range values are used to determine rotation and translation rates and when to stop.

4 Performance in the Painted Desert

In an ongoing series of field experiments, the Ames Marsokhod rover is deployed to remote locations and operated by scientists in mock planetary explorations. These experiments provide insight for both scientists preparing for real planetary exploration and for robotics researchers. During the latest experiment, our vision-based control system was used as the primary means of navigation. We intended to demonstrate that it would simplify navigation and make robot control more directly accessible to scientists.

4.1 Description of the field experiment

In early November 1996, a field experiment using the Marsokhod took place on the Navajo reservation in the Painted Desert of Northern Arizona. The site was chosen for its sparse vegetation and Mars-like geology. Three separate science teams with different mission objectives took part in the 6-day test, each team having 2 days to characterize the geology of the site using the Marsokhod.

The first team, which simulated the upcoming Mars Pathfinder mission, proceeded without any foreknowledge of the site since the Pathfinder lander does not have descent cameras. This limited the team to selecting features directly visible to the Marsokhod's cameras. All visual features selected were of scientific interest.

The second and third teams were given simulated descent imagery at the start of operations. This allowed the team to select not only nearby science sites, but areas outside the "landing area" that looked significant. With these teams, visual targets alternated between features of scientific interest and navigation waypoints chosen by the operator to traverse from one site to the next.

An important lesson is that some method of position determination is still helpful. The technique of using visual navigation waypoints was successful but did require careful inspection to determine whether the location identified in the descent image corresponded to the features observed on the ground. Each 8-hour day of the experiment involved about 2 hours of robot driving; the remaining time was spent collecting and analyzing science imagery.

4.2 Operator interface for vision-based control

The operator interface for vision-based control of the Marsokhod is simply the window (implemented in the TCL/Tk scripting language) shown in Figure 5. This window provides an obvious and intuitive method of selecting a visual target from an image: the operator designates a target simply by clicking the cursor (indicated with a small cross) on the feature. The interface sends a command to the pan-tilt head, centering the feature in the field-of-view. In this manner the operator can also look around until she has found a target. She may then command the robot to begin driving to the target. The interface has additional features that allow the operator to set various parameters, for example, to set the stopping distance from the target or the driving speed, and also the parameters to the correlator, like the averaging window size or the minimum correlation threshold.

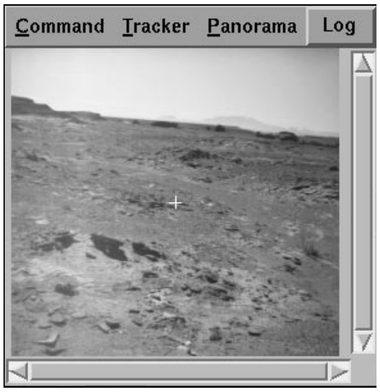


Figure 5: Operator interface for designating the visual target

4.3 Performance of vision-based control

During the field experiment the vision-based control preformed so well as to be transparent to the planetary scientists. They selected interesting features and the rover drove to them, sometimes in a few steps. We were successful in demonstrating the utility of the control scheme.

In total the Marsokhod drove over 400 meters with two long traverses exceeding 45 meters. To characterize our initial examination of the performance, almost all targets were tracked at first, but very few were tracked all the way to the stopping position. The majority were lost during the intermediate traverse as the vehicle drove over obstacles. In some cases, a single wheel on a rock induced a roll of 2° and caused the correlation to fail; for example, see Figure 6.

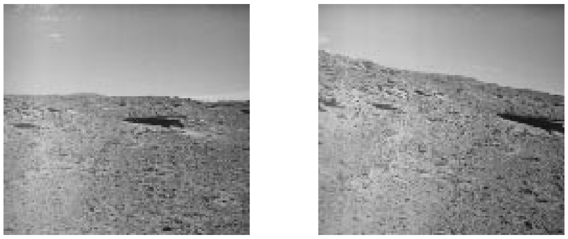

Target locked
Target lost

Figure 6: Roll about the chassis causes rotation and translation in the image and leads to correlation failures

Looking at a traverse in detail, in this case between the images in Figure 3, as the robot approaches the target the range decreases and the variation in range estimate also decreases, as shown in Figure 7.

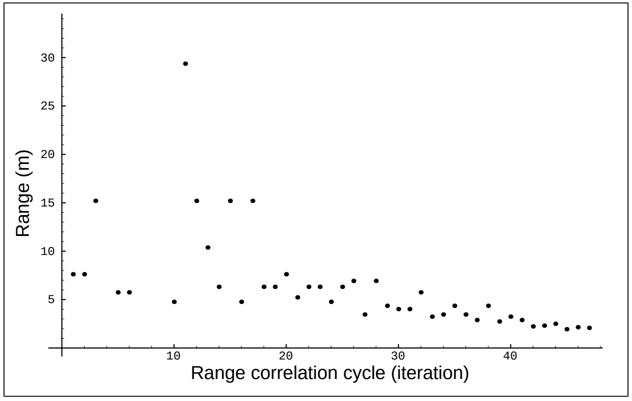


Figure 7: Correlation range versus range as the robot drives toward the target

Paralleling the trend in Figure 7, in Figure 8 the value of the motion correlation peak trends upward toward a perfect correlation of 1024. This indicates that the feature is becoming more visibly distinctive, the correlation is more closely matched, and the pixel disparity is more stable. This fits with our observation that, for our scale of target, at ranges of eight to three meters visual servoing is stable. This time the Marsokhod drove 8.5m in 2. 4 minutes to stop fixated exactly where the geologist had indicated.

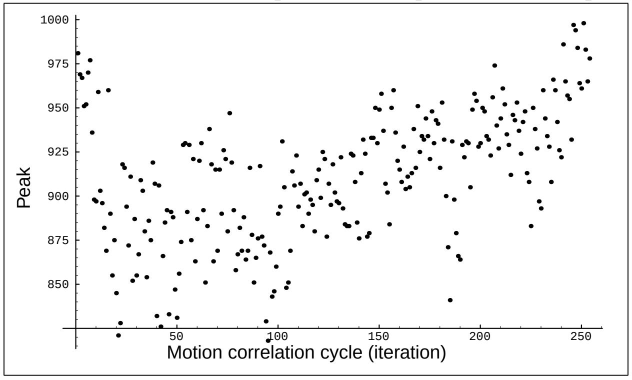
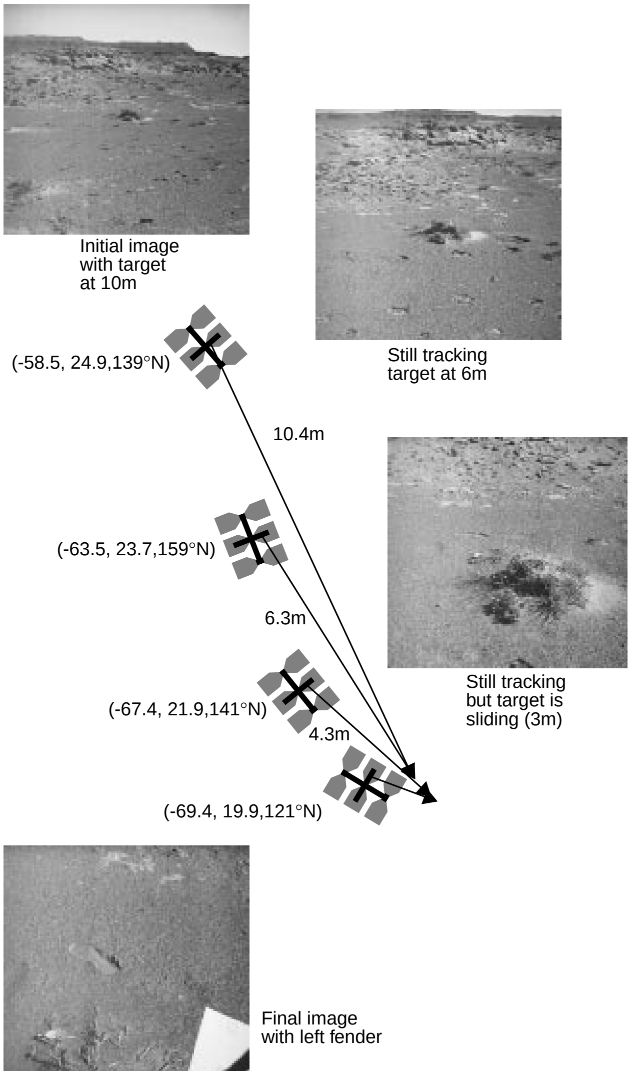


Figure 8: Iteration of the motion correlation versus peak intensities as the robot drives toward the target

In another successful traverse, the target, initially thought to be a rock outcrop, turned out to be some infrequent vegetation. Figure 9 overviews the traverse in which distance traveled was 12.3m at average velocity 7cm/sec.



Initial image with target at 10m
Still tracking target at 6m
(-58.5, 24.9,139°N)
10.4m
(-63.5, 23.7,159°N)
Still tracking but target is sliding (3m)
6.3m
(-67.4, 21.9,141°N)
4.3m
(-69.4, 19.9,121°N)
Final image with left fender

Figure 9: Overview of a vision-guided traverse

Visual-servoing, cycling at 2Hz, was able to keep the Marsokhod driving at its maximum safe speed and the target closely centered in the field-of-view (Figure 10).

The Marsokhod stopped itself when it correlated the target within its stopping distance (<2m); however without higher resolution subsampling (for target designation at long range) the target can be miscorrelated at short range, resulting in gradually sliding off the target

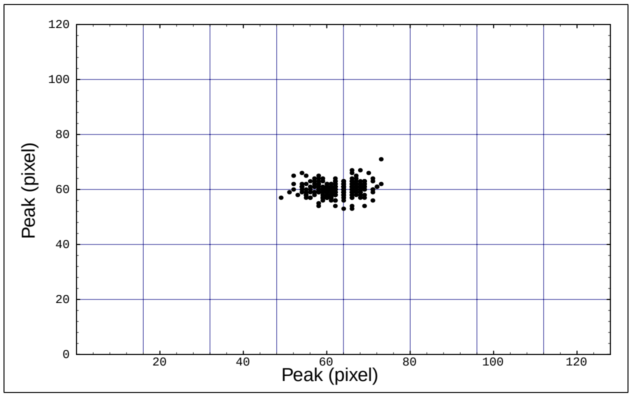


Figure 10: Location of the correlation peak during motion

The vision-based control method is particularly advantageous when operating with a communication time-delay,. The operator uploads a target designation, and in the next download cycle receive images from the robot, already at that target, perhaps even with its sample collection equipment deployed on the feature. We introduced a 6-minute delay (typical for Mars) in our communication system and drove the robot; this run is shown in Table 1, The rover first makes a long traverse, losing its target after 40 meters and then spends time taking images in various directions (a process that should be performed in one command cycle), before it drives to targets of interest.

Time	X	Y	Heading	Distance
15:15	-76.2	10.1	27.5°	
15:27	-31.8	22.0	339.7°	39.4m
15:48	-24.2	23.4	336.1°	6.2m
16:06	-13.9	28.4	1.3°	11.4m

Table 1: Time-delayed performance

In the final drive of the field experiment we targeted the distinctive bumper of the command truck. The Marsokhod drove continuously for 45.3m, in one area climbing a 20° slope, to stop less than 2m from its destination.

5 Discussion

There are three significant failure modes for this vision-based control scheme: target loss due to robot motion and subsequent appearance change, above-threshold correlation for an erroneous target (false positive), and correlation that slowly drifts off the target due to weak feature texture.

The most frequent cause of target loss is excessive vehicle pitch or roll. We have already implemented a target loss strategy in which the robot stops after a series of below-threshold correlations, and executes an expanding search pattern. On occasion, the robot has reacquired the target, but stronger strategies would anticipate target motion to constrain search.

By far the most frequent reason why the vision-based control method fails is that roll about the optical axis (which the pan-tilt head cannot compensate) leads to correlation failure. An improvement may be to compensate for some vehicle motion in the pan-tilt control. Affine transformation of the correlation template has been used by others [5] to mitigate camera roll.

To reduce the occurrence of false positive correlations, an improvement suggested by related work is to incorporate more robust correlation peak detection—complexity here must be traded with processing speed.

Our current correlator derives subsamples by averaging over the entire image. Given that our cameras fixate on the target feature, high-resolution foveal processing suggested in [6] could be more effective. A hybrid approach that uses high and low resolution templates may ease the mid-range transition as features change appearance.

6 Summary

We have developed a vision-based control system for coordinated motion control of an actively pointed pan-tilt device on a mobile robot. Our system enables a mobile robot to visually guide itself to a designated natural feature. This capability provides a valuable degree of autonomy to remote planetary rovers.

Our implementation runs onboard a general purpose processor and is able to drive the Marsokhod prototype rover at its maximum speed. Improvements being pursued include foveal processing and vehicle motion compensation. Our initial results indicate promising performance in visually-servoing a mobile robot in natural terrain.

7 Acknowledgments

This work was supported by the NASA Space Telerobotics Program. We thank members of Intelligent Mechanisms Group including Mike Sims, Dan Christian, Cesar Mina, Kurt Schwer, Deanne Tucker, and Eric Zbinden for their assistance. We also thank the participants in the Arizona field experiment and the Navajo nation for their support.

8 References

[1] Hutchinson, S., Hager, G., Corke, P., “A Tutorial on Visual Servo Control,” IEEE Transactions on Robotics and Automation, Vol. 12, No. 5, 1996, pp. 651-671.

[2] Nishihara, K., “Practical Real-Time Imaging Stereo Matcher”, Optical Engineering, Vol. 23, 1984, pp. 536-545

[3] Okutomi, M. and Kanade, K. “A Locally Adaptive Window for Signal Matching”, CMU-CS-90-178, Carnegie Mellon School of Computer Science,1990

[4] Matthies, L., “Dynamic Stereo Vision”, CMU-CS-89-195, Carnegie Mellon Computer Science, 1989

[5] Omead Amidi, “Autonomous Vision-Guided Helicopter”, Ph.D. Thesis, Dept. of Elec. and Comp. Engineering., Carnegie Mellon University, 1996

[6] Coombs, D.J., “Real-Time Gaze Holding in Binocular Robot VIsion”, Technical Report 415, University of Rochester School of Computer Science, 1992

[7] Shi, J., Tomasi, C., “Good features to track”, Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 1994, pp. 593-600.