International Journal of Computer Vision

DOI: 10.1007/s11263-007-0046-z

Computer Vision on Mars

LARRY MATTHIES, MARK MAIMONE, ANDREW JOHNSON, YANG CHENG, REG WILLSON, CARLOS VILLALPANDO, STEVE GOLDBERG, AND ANDRES HUERTAS
Jet Propulsion Laboratory, California Institute of Technology, 4800 Oak Grove Drive, Pasadena, CA 91109, USA
larry.matthies@jpl.nasa.gov
mark.maimone@jpl.nasa.gov
andrew.johnson@jpl.nasa.gov
yang.cheng@jpl.nasa.gov
reg.willson@jpl.nasa.gov
carlos.villalpando@jpl.nasa.gov
steve.goldberg@jpl.nasa.gov
andres.huertas@jpl.nasa.gov

ANDREW STEIN
Carnegie Mellon University, 5000 Forbes Avenue, Pittsburgh, PA 15213, USA
anstein@cmu.edu

ANELIA ANGELOVA
California Institute of Technology, 1200 East California Boulevard, Pasadena, CA 91125, USA
anelia@caltech.edu

Received May 16, 2006; Accepted February 20, 2007

Abstract. Increasing the level of spacecraft autonomy is essential for broadening the reach of solar system exploration. Computer vision has and will continue to play an important role in increasing autonomy of both spacecraft and Earth-based robotic vehicles. This article addresses progress on computer vision for planetary rovers and landers and has four main parts. First, we review major milestones in the development of computer vision for robotic vehicles over the last four decades. Since research on applications for Earth and space has often been closely intertwined, the review includes elements of both. Second, we summarize the design and performance of computer vision algorithms used on Mars in the NASA/JPL Mars Exploration Rover (MER) mission, which was a major step forward in the use of computer vision in space. These algorithms did stereo vision and visual odometry for rover navigation and feature tracking for horizontal velocity estimation for the landers. Third, we summarize ongoing research to improve vision systems for planetary rovers, which includes various aspects of noise reduction, FPGA implementation, and vision-based slip perception. Finally, we briefly survey other opportunities for computer vision to impact rovers, landers, and orbiters in future solar system exploration missions.

Keywords: stereo vision, obstacle detection, visual odometry, visual velocity estimation, slip prediction, planetary exploration

1. Introduction

Both on Earth and in space, a key motivation for developing computer vision-based, autonomous navigation systems is that communication latency and bandwidth limitations severely constrain the ability of humans to control robot functions remotely. In space, onboard computer vision enables rovers to explore planetary surfaces more quickly and safely, landers to land more safely and precisely, and orbiters to better maintain safe orbits

in the weak and uneven gravity fields of small asteroids, comets, and moons. The performance limitations of space-qualified computers strongly constrain the complexity of onboard vision algorithms. Nevertheless, the MER mission, which landed two rovers on Mars in 2004, very successfully used stereo vision, visual odometry, and feature tracking for rover navigation and for estimating the horizontal velocity of the landers before touchdown. This was the first use of such algorithms in a planetary exploration mission. More advanced capabilities, using more advanced spaceflight computers, are now of interest for future missions.

This paper starts with a historical perspective on the four decades of research that led up to the autonomous navigation capabilities in MER (Section 2), then describes the design and performance of the MER vision systems (Section 3), ongoing research to improve them (Section 4), and other opportunities for computer vision to impact rover, lander, and orbiter missions (Section 5). The history of planetary rover research is tightly interwined with research on robots for Earth-bound applications, since this work has proceeded in parallel, served related requirements, often been done by the same people, and experienced much cross-fertilization. Therefore, the historical perspective touches on highlights of both.

In discussing the MER mission itself, we summarize pertinent characteristics of the mission, the rover, and the lander, then we briefly describe the algorithms, present main results of their operation in the mission, and note their key limitations. Rover navigation used stereo cameras for 3-D perception and visual odometry. Computing was performed by a 20 MHz "RAD6000" flight computer, which is a space-qualified version of an early PowerPC architecture. The lander used one descent camera and the same flight computer to track features over three frames in the last 2 km of descent to the surface, in order to estimate terrain-relative horizontal velocity so that retro-rockets could reduce that velocity, if necessary, to avoid tearing the airbags on impact. Since these algorithms and results are discussed in detail elsewhere, we keep the discussion to an overview and provide references for more detail.

Ongoing research on vision for rover navigation addresses both engineering and research-oriented ends of the spectrum. At the engineering end of the spectrum, work on stereo includes FPGA implementation, to increase speed in a space-qualifiable architecture, and improved algorithms for rectification, prefiltering, and correlation to reduce noise, improve performance at occluding boundaries (edges of rocks), and reduce pixel-locking artifacts. At the research-oriented end of the spectrum, one of the major navigation safety and performance issues in MER was slippage on sloping terrain. Visual odometry addressed this to some degree for MER, but we are now applying learning algorithms to attempt to predict the amount of slip to expect from the appearance and slope angle of hills immediately in front of the rover.

The most significant possibilities to impact future missions are to improve position estimation for precision landing, to detect landing hazards, and to improve station-keeping and orbit estimation around low-gravity bodies, including small asteroids, comets, and moons of the outer planets. Sensors and algorithms are in development for all of these functions at JPL and elsewhere.

2. Historical Perspective

Planetary rover research began in the early 1960s with analysis and prototype development of a robotic lunar rover for NASA's Surveyor program (Bekker, 1964) (Table 1, Fig. 1). The U.S. never sent an unmanned rover to the moon, but the Soviet Union sent two teleoperated "Lunokhod" rovers in the early 1970s (A Scientific Rationale for Mobility in Planetary Environments, 1999). Research on more automated navigation of rovers for Mars continued through the 1970s at JPL, Stanford University, and elsewhere, using onboard stereo cameras and scanning laser rangefinders and off-board computers (O'Handley, 1973; Levine et al., 1973; Thompson, 1977; Lewis and Johnston, 1977; Gennery, 1980; Moravec, 1980).

In the early 1980s, there was a hiatus in planetary rover funding from NASA, but mobile robot research continued under funding from other agencies at various research centers. At Carnegie Mellon University (CMU), Moravec developed a series of mobile robots using stereo vision for perception (Moravec, 1983). One product of this line of work was a stereo vision-based visual odometry algorithm that produced the first quantitatively accurate stereo vision-based egomotion estimation results (Matthies and Shafer, 1987) and led to the visual odometry algorithm now in use on Mars.

Computer vision for mobile robots got a major boost in this period with the start of the DARPA Strategic Computing (SC) and Autonomous Land Vehicle (ALV) programs. One goal of SC was to demonstrate vision and advanced computing results from SC on robotic vehicles developed under ALV. At CMU, Takeo Kanade was a Principal Investigator in both of these programs. Under SC, Kanade initiated a long-running effort to develop fast stereo vision implementations on special-purpose computing hardware, commencing with an implementation on the "Warp" systolic array computer at CMU (Guerra and Kanade, 1985).

Robot vehicles built under the ALV program, including the CMU "Navlab", a converted Chevy van, were possibly the first to have enough onboard computing power to host substantial terrain perception, terrain mapping, and path planning algorithms onboard. ALV-related research

Table 1. Chronology of sample ground robot systems and programs

Period	Robot system/program
Mid 1960s	Robotic lunar rover prototypes for NASA Surveyor program
Early 1970s	Russian Lunokhod lunar rovers
1970s	Stanford Cart
Mid-late 1980s	DARPA Autonomous Land Vehicle (ALV) Program; first CMU Navlab
1995	"No-Hands Across America" road-following demo by CMU Navlab 5
Late 1980s-early 1990s	CMU Ambler
Late 1980s-early 1990s	JPL Robby
1992–1996	DARPA "Demo II" Unmanned Ground Vehicle (UGV) Program
1997	Mars Pathfinder mission with Sojourner rover
1997–2002	DARPA Tactical Mobile Robotics (TMR) Program
1998–2001	Demo III Experimental Unmanned Vehicle (XUV)
2001-present	Robotics Collaborative Technology Alliance
2000–2003	DARPA Perception for Off-road Robotics (PerceptOR) Program
2003-present	Mars Exploration Rover mission with Spirit and Opportunity
2004–2007	DARPA Learning Applied to Ground Robotics (LAGR) Program
2004–2005	DARPA Grand Challenge (1 and 2) desert robot race

Figure 1. Sample ground robots. Top: Surveyor lunar rover prototype, Lunokhod 1, CMU Navlab 1. Middle: JPL Robby Mars rover testbed, CMU Ambler Mars rover testbed, CMU Dante. Bottom: Demo II UGV, Demo III XUV, DARPA TMR testbed.

under Kanade and colleagues focused on road-following with monocular, color imagery and on terrain mapping and obstacle avoidance with a two-axis scanning laser rangefinder (ladar) built by the Environmental Research Institute of Michigan (ERIM). By the end of the decade, the Navlab was able to follow a variety of structured and unstructured roads and avoid obstacles in modestly rough off-road terrain at speeds from a few kilometers per hour (kph) off-road to 28 kph on structured roads (Thorpe et al., 1991a,b). This was the beginning of an extensive, ongoing body of work at CMU on both road-following and terrain mapping for off-road navigation by a series of students and more junior faculty. Highlights of this body of work include such achievements as the "No-Hands Across America" trip by Navlab 5 from Pittsburgh to San Diego, which covered 2849 miles with 98.2% of it autonomous (Pomerleau and Jochem, 1996). Work with the ERIM ladar produced a number of techniques for registering and fusing sequences of range images into aggregate terrain maps and for doing obstacle detection and avoidance with such maps (Hebert et al., 1988). Terrain mapping and analysis for off-road obstacle avoidance remains an open, active area of research 20 years later.

In the mid-to-late 1980s, NASA resumed funding research on autonomous navigation for Mars rovers, with JPL and CMU as the primary participants. The initial co-Principal Investigators at CMU were Kanade, Tom Mitchell, and Red Whittaker. CMU carried over its ladar-based work from the ALV program into the planetary rover domain (Hebert et al., 1989), while JPL explored the use of stereo vision as an all-solid-state approach that might be more easily space qualifiable. CMU built a six-legged robot over 4 m tall called Ambler, to be able to step over 1 m tall obstacles, and developed perception, planning, and control algorithms for statically stable legged locomotion (Krotkov and Simmons, 1996). This was followed at CMU by a series of NASA-funded projects led by Whittaker to develop mobile robots (Dante I and Dante II) for major field campaigns on Earth, including descending into volcanoes in Antarctica and Alaska (Wettergreen et al., 1993; Bares and Wettergreen, 1999). Dante II included a novel ladar mounted on a central mast with a 360 degree, spiral scan pattern to do 360 degree mapping around the robot. JPL's effort achieved a breakthrough in real-time, area-based stereo vision algorithms that enabled the first stereo vision-guided autonomous, off-road traverse (Matthies, 1992). This algorithm used SSD correlation, implemented with efficient sliding sums and applied at low resolution to bandpass filtered imagery. Implemented on a Motorola 68020 CPU and Datacube convolution hardware, the system produced 64×60 pixel range imagery at 0.5 Hz. This success shifted the focus of stereo vision research from edge-based methods to area-based methods and inspired other robotic vehicle projects to experiment more with stereo.

The 1990s was a period of tremendous progress, enabled by more powerful computing and better 3-D sensors. The DARPA Unmanned Ground Vehicle (UGV) program built robotic HMMWVs relying on stereo vision for 3-D perception (Mettala, 1992). Autonomous off-road runs of up to 2 km at 8 kph were achieved with a stereo system that generated range data in a 256×45 pixel region of interest at about 1.3 Hz (Matthies et al., 1996). This program also experimented for the first time with stereo vision at night using thermal infrared cameras (Matthies et al., 1996; Hebert et al., 1996). Concurrently, Okutomi and Kanade developed an influential, SAD-based, multi-baseline stereo algorithm (Okutomi and Kanade, 1993), which Kanade and co-workers extended in a custom hardware implementation as the CMU Video-Rate Stereo Machine (Kanade et al., 1996). This produced 256×240 disparity maps at 30 Hz. A software version of this algorithm was evaluated for obstacle detection on highways at speeds up to 25 mph (Williamson, 1998). When Konolige showed that SAD-based stereo algorithms could run at up to 30 Hz for 320×240 imagery using only current DSPs or microprocessors (Konolige, 1997), emphasis shifted away from special-purpose hardware implementations.

The 1990s also saw relatively sophisticated sensors and autonomous navigation functions migrate into small robots. In the DARPA Tactical Mobile Robotics (TMR) program, tracked robots less than one meter long were equipped with stereo vision and/or SICK single-axis scanning laser rangefinders and programmed to do obstacle mapping and avoidance, vision-guided stair climbing, and indoor mapping of hallway networks (Krotkov and Blitch, 1999; Matthies et al., 2002; Thrun, 2001). In this period, NASA refocused on small rovers for affordability reasons and landed the Sojourner rover on Mars in the 1997 Mars Pathfinder mission (Wilcox and Nguyen, 1998). Since Sojourner's computer was only an Intel 8085 clocked at 2 MHz, its 3-D perception system was a simple light-stripe sensor that measured about 25 elevation points in front of the rover (Matthies et al., 1995). The lander had a multispectral stereo camera pair on a pan/tilt mast about 1.5 m high. Processing this stereo imagery on Earth with JPL's real-time stereo algorithm produced excellent maps of the terrain around the lander for rover operators to use in planning the mission. This was validation of the performance of the stereo algorithm with real Mars imagery.

Major outdoor autonomous robot research programs in the 2000s to date include the Demo III Experimental Unmanned Vehicle (Demo III XUV) and Robotics Collaborative Technology Alliance (RCTA) programs, both funded by the Army Research Lab (ARL), the DARPA Perception for Off-Road Robotics (PerceptOR) and Learning Applied to Ground Robotics (LAGR) programs, and NASA's Mars Exploration Rover (MER)

mission and supporting technology development. Demo III, RCTA, and PerceptOR addressed off-road navigation in more complex terrain and, to some degree, day/night, all-weather, and all-season operation. A Demo III follow-on activity, PerceptOR, and LAGR also involved systematic, quantitative field testing. For results of DemoIII, RCTA, and PerceptOR, see (Shoemaker and Bornstein, 2000; Technology Development for Army Unmanned Ground Vehicles, 2002; Bornstein and Shoemaker, 2003; Bodt and Camden, 2004; Krotkov et al., 2007) and references therein. LAGR focused on applying learning methods to autonomous navigation. The DARPA Grand Challenge (DGC), though not a government-funded research program, stressed high speed and reliability over a constrained, 131 mile long, desert course. Both LAGR and DGC are too recent for citations to be available here. We review MER in the next section.

With rover navigation reaching a significant level of maturity, the problems of autonomous safe and precise landing in planetary missions are rising in priority. Feature tracking with a downlooking camera during descent can contribute to terrain-relative velocity estimation and to landing hazard detection via structure from motion (SFM) and related algorithms. Robotic helicopters have a role to play in developing and demonstrating such capabilities. Kanade has made many contributions to structure from motion, notably the thread of factorization-based algorithms initiated with Tomasi and Kanade (1992). He also created one of the largest robotic helicopter research efforts in the world (Amidi et al., 1998), which has addressed issues including visual odometry (Amidi et al., 1999), mapping (Miller and Amidi, 1998; Kanade et al., 2004), and system identification modeling (Mettler et al., 2001). For safe and precise landing research *per se*, JPL began developing a robotic helicopter testbed in the late 1990s that ultimately integrated inertial navigation, SFM, and a laser altimeter to resolve scale in SFM. This achieved the first fully autonomous landing hazard avoidance demonstration using SFM in September of 2003 (Johnson et al., 2005a,b; Montgomery et al., to appear).

Finally, Kanade guided in early work in the area that became known as physics-based vision (Klinker et al., 1990; Nayar et al., 1991; Kanade and Ikeuchi, 1991), which exploits models of the physics of reflection to achieve deeper image understanding in a variety of ways. This outlook is reflected in our later work that exploits physical models from remote sensing to improve outdoor scene interpretation for autonomous navigation, including terrain classification with multispectral visible/near-infrared imagery (Matthies et al., 1996), negative obstacle detection with thermal imagery (Matthies and Rankin, 2003), detection of water bodies, snow, and ice by exploiting reflection, thermal emission, and ladar propagation characteristics (Matthies et al., 2003), and modeling the opposition effect to avoid false feature tracking in Mars descent imagery (Cheng et al., 2005).

3. Computer Vision in the MER Mission

The MER mission landed two identical rovers, Spirit and Opportunity, on Mars in January of 2004 to search for geological clues to whether parts of Mars formerly had environments wet enough to be hospitable to life. Spirit landed in the 160 km diameter Gusev Crater, which intersects the end of one of the largest branching valleys on Mars (Ma'adim Vallis) and was thought to have possibly held an ancient lake. Opportunity landed in a smooth plain called Meridiani Planum, halfway around the planet from Gusev Crater. This site was targeted because orbital remote sensing showed that it is rich in a mineral called gray hematite, which on Earth is often, but not always, formed in association with liquid water. Scientific results from the mission have confirmed the presence of water at both sites, and the existence of water-derived alteration of the rocks at both sites, but evidence has not been discovered yet for large lakes (Squyres and Knoll, 2005).

Details of the rover and lander design, mission operation procedures, and the individual computer vision algorithms used in the mission are covered in separate papers. In this section, we give a brief overview of the pertinent aspects of the rover and lander hardware, briefly review the vision algorithms, and show experimental results illustrating qualitative behavior of the algorithms in operation on Mars. Section 4 addresses more quantitative performance evaluation issues and work in progress to improve performance.

3.1. Overview of the MER Spacecraft and Rover Operations

Figure 2 shows a photo of one of the MER rovers in a JPL clean room, together with the flight spare copy of the Sojourner rover from the 1997 Mars Pathfinder mission for comparison. The MER rovers weigh about 174 kg, are 1.6 m long, have a wheelbase of 1.1 m, and are 1.5 m tall to the top of the camera mast. Locomotion is achieved with a rocker bogie system very similar to Sojourner, with six driven wheels that are all kept in contact with the ground by passive pivot joints in the rocker bogey suspension. The outer four wheels are steerable.

The rovers are solar powered, with a rechargeable lithium ion battery for nighttime science and communication operations. The onboard computer is a 20 MHz RAD6000, which has an early PowerPC instruction set, with no floating point, a very small L1 cache, no L2 cache, 128 MB of RAM, and 256 MB flash memory. Navigation is done with three sets of stereo camera pairs: one pair of "hazcams" (hazard cameras) looking forward

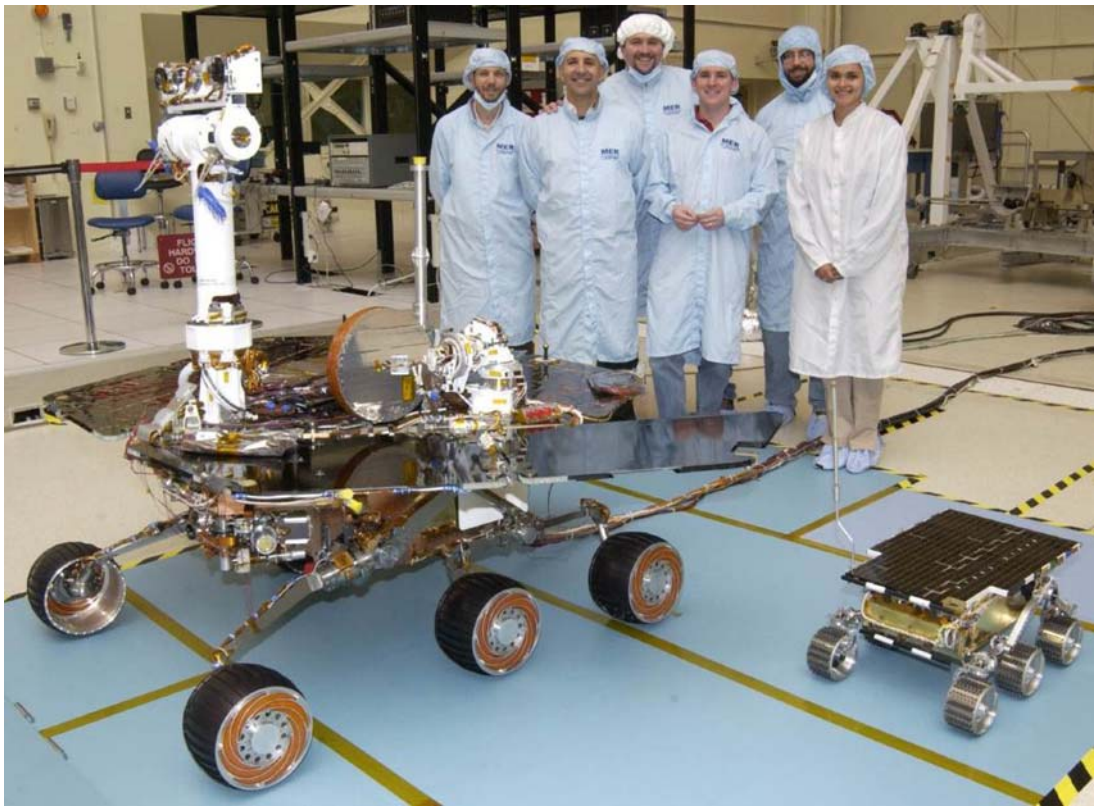


Figure 2. MER rover (left) with Sojourner rover from the 1997 Mars Pathfinder mission (right), shown in a JPL clean room.

under the solar panel in front, another pair of hazcams looking backward under the solar panel in the back, and a pair of "navcams" (navigation cameras) on the mast. All cameras have 1024×1024 pixel CCD arrays that create 12 bit greyscale images. The hazcams have a 126 degree field of view (FOV) and baseline of 10 cm; the navcams have a 45 degree FOV and baseline of 20 cm (Maki et al., 2003). Each rover has a five degree of freedom arm in front which carries a science instrument payload with a microscopic imager, Mossbauer spectrometer, alpha/proton/x-ray backscatter spectrometer (APXS), and a rock abrasion tool (RAT). The camera mast has two additional science instruments: a stereo pair of "pancams" (panoramic cameras) and the "mini-TES" (thermal emission spectrometer). The pancams have filter wheels for multispectral visible and near-infrared imaging for mineral classification. They have the highest angular and range resolution of all cameras on the rover, with a 16 degree field of view and 30 cm baseline. The mini-TES acquires 167 spectral bands between 5 and 29 μm in a single pixel. All instruments on the mast are pointable by one set of pan/tilt motors.

Because of constraints on solar power, the rovers drive for up to 3 hours per sol,[1] followed by a downlink telemetry session of up to 2 hours per sol. A large team of people plans the next sol or several sols' mission in the remaining hours per sol. The rovers' top driving speed is 5 cm/sec, but they are typically driven at 3.75 cm/sec to limit motor heating. The basic traverse cycle involves acquiring hazcam stereo images and planning a short drive segment while standing still, then driving 0.5 to 1.5 m, then stopping and repeating the process. With computing delays, this results in a net driving speed on the order of 1 cm/sec. Because the *a priori* 3σ landing uncertainty ellipse was about 80×10 km, exact targets for exploration could not be identified before landing. After landing, the science team concluded that the desirable investigation sites required the rovers to travel more quickly than planned in order to reach them within tolerable time limits. This led to a new operational mode for long distance drives in which navcam or pancam stereo pairs acquired at the end of each sol are used by human operators to identify hazard-free paths up to 100 m ahead for the next sol's traverse. The rovers drive these initial segments with little or no obstacle detection and avoidance processing, then switch to "autonav" mode with complete obstacle detection and avoidance. This has enabled drives of up to 370 m/sol in the most flat, safe terrain.

Additional details about the rover hardware, software architecture, and operations are given in Maimone et al. (2006) and references therein.

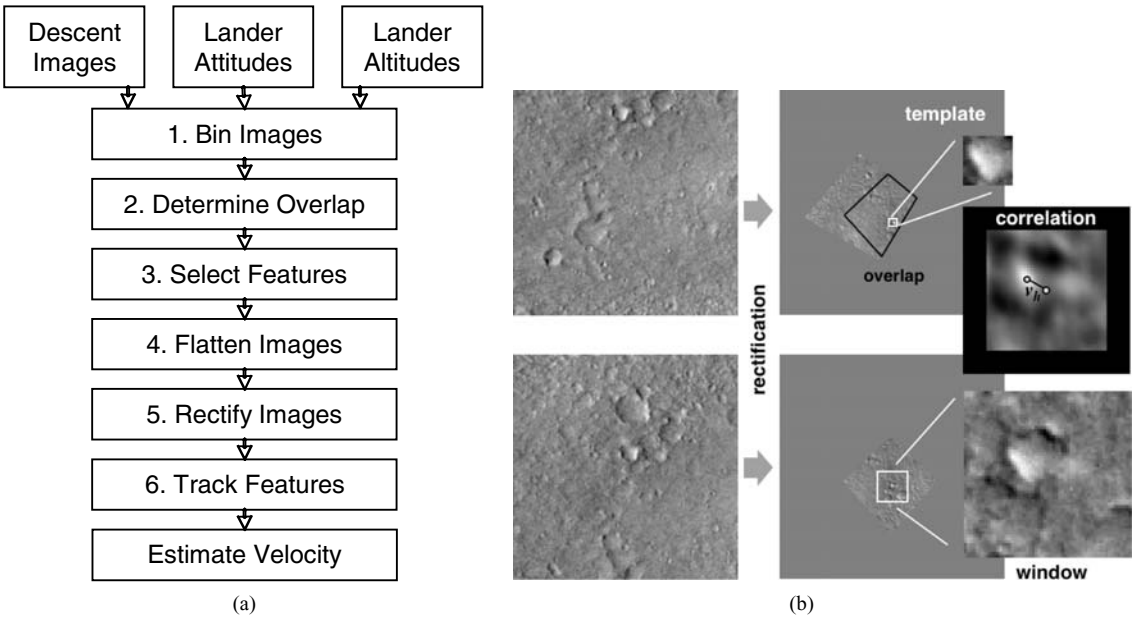


Figure 3. Basic elements of DIMES: (a) algorithm flow, (b) pictorial illustration.

3.2. Descent Image Motion Estimation System

The first vision algorithm to operate in the MER mission was the Descent Image Motion Estimation System (DIMES), though it was by far the last to be developed. A little more than two years before launch, MER staff realized that statistical models that had been used for Mars near-surface wind velocities were wrong, and that an improved model predicted higher steady-state winds, with a consequently higher horizontal velocity at impact and higher probability of catastrophic tearing of the lander airbags (Cheng et al., 2004). The lander system had lateral rockets ("TIRS", for Transverse Impulse Rocket System) that were needed to orient the lander vertically before firing the main retrorockets. In principle, TIRS could be used to reduce horizontal velocity, but there was no horizontal velocity sensor in the system to guide such a maneuver. Cost and schedule constraints prohibited adding a doppler radar velocity sensor, which is the usual approach to velocity sensing. By coincidence, a sun sensing camera had been built for MER but deleted from the system earlier in development. The only velocity sensing solution that did fit in the cost and schedule constraints was to reinsert this camera as a descent camera and to develop software to use it to estimate velocity. With an inertial measurement unit (IMU) in the lander to sense angular velocity and an altimeter to sense vertical velocity, the entire velocity vector could be estimated by tracking a single surface feature. More features are desirable for reliability and precision, but limitations of the onboard computer allowed tracking only two features per frame in real-time. A set of redundant measurements and error checks made this robust and an extensive testing protocol with elaborate simulations and field testing validated performance of the system at the required levels of precision and reliability.

The basic elements of the DIMES algorithm are illustrated in Fig. 3 and consist of the following; details are given in Cheng et al. (2005). Many of the details were motivated by the need to fit within the very limited computing power and time available.

1. The raw 1024×1024, 12 bit descent imagery was reduced to 256×256 pixels by a combination of binning in the CCD for one axis and software averaging in the other axis, then truncated to 8 bits/pixel. To avoid tracking the shadow of the parachute, knowledge of the lander attitude and sun direction was used to identify where the shadow would occur in the image. A radiometric effect called the "opposition effect" causes a broad peak in image brightness around that point, which could also interfere with tracking (Hapke, 1986). A "zero phase mask" was computed to eliminate a pre-determined part of the image to avoid both of these problems.
2. For each pair of images, knowledge of the altitude, an upper bound on horizontal velocity, and bounds on attitude measurement errors were used to determine the maximum possible area of overlap between the images and the extent of search windows to use for feature tracking.
3. Two features were selected by applying a Harris interest operator on a coarse grid in one image, within the area of overlap and avoiding the zero phase mask.

4. Radiometric corrections ("flattening") were applied to the selected feature templates and search windows to reduce the effects of (1) smearing because the CCD camera had a frame transfer architecture without a shutter, (2) pixel-to-pixel response variations, and (3) vignetting due to optical transfer roll-off.
5. The feature templates and search windows were rectified to take out orientation and scale differences by using knowledge of lander altitude, attitude, and orientation relative to north to project the imagery into a camera frame parallel to the ground with the same scale and orientation for both images.
6. Features were matched between images by applying Moravec's pseudo-normalized correlator (Moravec, 1980) in a two-level image pyramid, with subpixel peak detection at the highest resolution. Validity checks applied to screen false matches were correlation value, peak width, and the ratio between the two best correlation peaks.

Three images were acquired in total. Two features were tracked between the first pair and combined with the IMU and altimetry to produce one velocity estimate. Two features were used in case one failed to track, and two was the most that would fit in the time budget. Two more features were tracked between the second and third image to produce a second velocity estimate. Differencing these produced a rough acceleration estimate for the total interval, which was compared with accelerations measured with the IMU for a final error check. The total runtime of this algorithm on the flight computer was just under 14 sec, using about 40% of the CPU. To amplify the runtime constraint, in 14 sec the landers fell over 1000 m, which was more than half the distance to the ground from where the first image was acquired.

The Harris interest operator embodied a generic feature definition that was applicable to any kind of terrain and could be computed quickly. Tracking features by multiresolution correlation search, instead of by a gradient descent tracker or other means of estimating optical flow, allowed features to be tracked despite the large camera motion between frames. The various optimizations described above for each stage of the algorithm allowed it to complete in the time available despite the slow clock rate, lack of cache, and lack of floating point in the processor.

This algorithm was tested first with a simulator, called MOC2DIMES, that used real Mars orbital imagery to generate triples of synthetic descent images, based on an elaborate model of the descent camera optical and noise effects, simulated descent trajectories from a model of lander dynamics, a model of the opposition effect, and a sampling of orbital imagery representative of the terrain variation within the 80×10 km landing targeting ellipse (Willson et al., 2005a). Results of Monte Carlo trials with this simulator predicted 3σ horizontal velocity estimation errors of <4 m/s for both landing sites, using imagery acquired at 2, 1.7, and 1.4 km above the ground. Because of the potentially severe consequences of a gross error, parameters were tuned so that no gross errors occurred in testing. This resulted in some cases where no velocity estimate was reported (1% of cases for Gusev Crater simulations and 29% for Meridiani Planum simulations), in which case the landing system would perform its default actions without input from the vision system. This was deemed acceptable by the mission.

The algorithm was then tested in the field by mounting an engineering model of the real descent camera on a manned helicopter, within a gimbal to emulate attitude disturbances from parachute oscillation, together with an inertial measurement unit, data collection hardware, and ground truthing sensors (Johnson et al., 2005b). This was flown at three different altitudes over several Mars analog sites in the Mojave desert, and triples of images selected from the three altitudes were used to test the algorithm against the measured ground truth. The distribution of the velocity errors over 1913 test cases was consistent with the MOC2DIMES simulation results.

DIMES was employed for both MER landings. For Spirit, it estimated that the horizontal velocity at impact would be about 23.5 m/s without firing TIRS, which was on the edge of the airbag limits. Therefore, TIRS was fired, producing a safe horizontal velocity of 11 m/s. For Opportunity, near-surface winds were lower and DIMES determined that TIRS firing was unnecessary. Both landings were successful; DIMES may have been instrumental in the successful landing of Spirit. Figure 4 shows sample images and processing results from the Spirit landing; the entire DIMES algorithm, testing, and flight results are described in detail in Johnson et al. (2006).

3.3. Stereo Vision

In the evolution from the 1997 Mars Pathfinder (MPF) mission to MER, traverse goals for the rovers expanded from staying within about 10 m of the lander for MPF to traveling up to 100 m/day and at least 600 m in 90 days for MER. To make obstacle detection more reliable, JPL chose to improve the quality of 3-D terrain perception beyond the very sparse measurements of the Sojourner light-stripe system. At the time, the main alternatives for doing so were scanning laser rangefinders (ladar) and stereo vision. In missions like this, it is always desirable to have imagery for human consumption, even if a ladar is used. Ladar has an advantage of greater range, but this was unnecessary given the slow driving speed and relatively low obstacle frequency. Stereo vision has the advantage of being all solid-state, hence having greater mechanical reliability and longer life. Based on experience

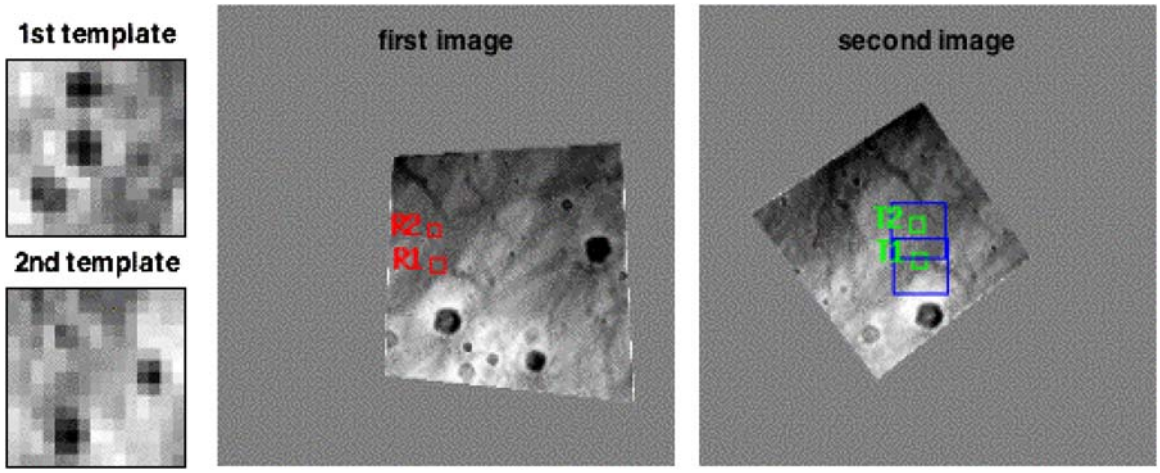


Figure 4. First pair of descent images for Spirit landing, shown after rectification. Selected templates are in red, search windows in blue, and match locations in green.

from the Mars Pathfinder mission, it was believed that Mars terrain would have enough texture for stereo vision to work nearly everywhere. It was already established that simple stereo algorithms based on area correlation could produce sufficiently dense, sufficiently accurate range imagery with sufficient speed within the available computing power (Goldberg et al., 2002; Matthies, 1992; Matthies et al., 1995). Therefore, stereo vision was selected as the best trade-off between cost, risk, and performance for 3-D sensing for this mission. The entire autonomous driving software architecture, including the stereo vision and obstacle avoidance algorithms, is discussed in Biesiasdecki and Maimone (2006). Here we summarize the main elements of the stereo algorithm, in order to discuss its limitations and compare it to modifications addressed in Section 4, then illustrate performance of the algorithm on Mars imagery.

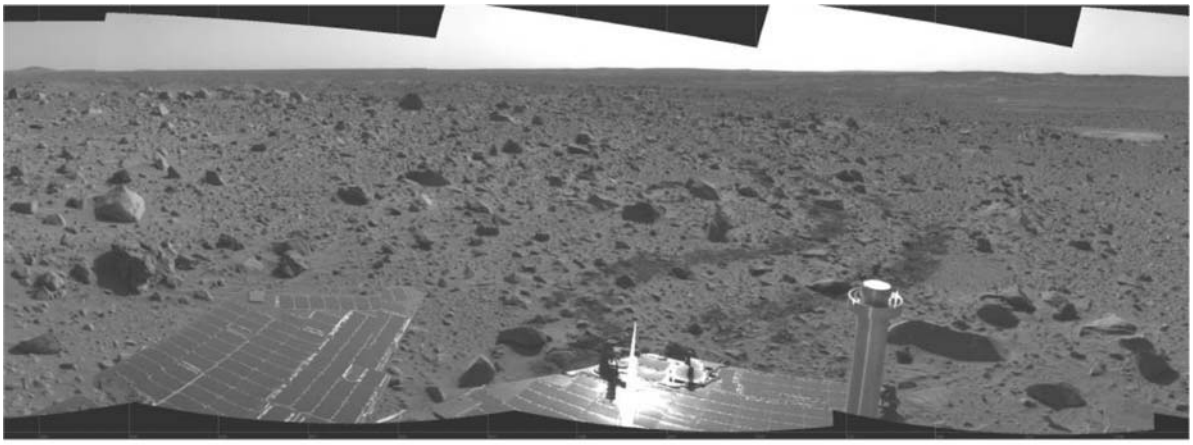
Main steps of the algorithm are as follows:

- The raw 1024×1024 images are binned down to 256×256 and rectified with bilinear interpolation.
- The images are highpass filtered by subtracting out local block averages.
- 7×7 SAD scores are computed for the disparity search range and minima are computed independently for each pixel.
- The left-right check is applied to eliminate ambiguous matches.
- Subpixel disparity is computed by fitting parabolas to the three SAD scores around the minima.
- Small blobs of unreliable disparity estimates are removed by applying a modified blob filter for which the connectivity criterion is a threshold on the disparity gradient.
- A few other local reliability tests are applied, such as thresholding the curvature of the parabolic fit, masking out parts of the image known to contain parts of the rover, and eliminating any pixels that appear to represent overhangs, which on Mars are invariably due to disparity artifacts caused by noise or nearly horizontal intensity edges in the imagery (e.g. shadows and the skyline).

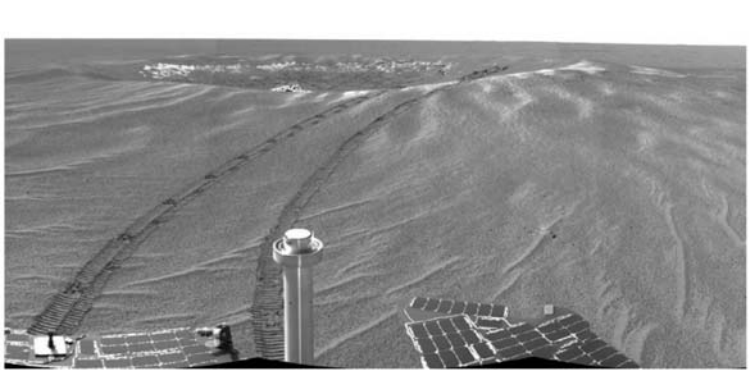
On the MER flight processor, with rest of the flight software system running at the same time, 256×256 disparity maps take about 30 seconds to compute. For comparison, the same algorithm runs at 30 Hz for 320×240 disparity maps on a 1.4 GHz Pentium M in other applications.

Gusev Crater has scattered rocks and is highly textured, but much of Meridiani Planum turned out to be a smooth, featureless plain with fine soil that produces relatively low texture imagery (Fig. 5). Spirit could get adequate range data at Gusev from the wide angle hazcams, but Opportunity had to use the higher resolution, narrower angle navcams to perceive the soil texture well enough get adequate range data.

Figure 6 shows a typical stereo result for navcams and Fig. 7 for hazcams, both from Gusev Crater. The small holes in the range image result mostly from the left-right check; we do not interpolate over such holes. The ragged disparity border around the edge of Humphrey rock is typical of this class of stereo algorithm; this does not noticeably impact obstacle avoidance performance. The rippling visible in the elevation plot is the result of pixel-locking artifacts in the subpixel disparity estimation process; there is insufficient computing power onboard to address this with acceptable runtime. While this raises the noise floor in elevation maps, at short range it is still far below the size of rock that is a hazard. This is illustrated by the fact that the green bumps corresponding to small rocks in the height image in the upper right of Fig. 5



(a)



(b)

(c)

Figure 5. (a) Navcam mosaic from Spirit in Gusev Crater at the end of a traverse; wheel tracks show some obstacle avoidance maneuvering. (b) Navcam mosaic from Opportunity in Meridiani Planum. Opportunity landed in the 20 m diameter Eagle Crater in the middle of this mosaic. (c) Textureless hazcam image from inside Eagle Crater. Additional imagery can be seen at **http://marsrovers.jpl.nasa.gov/gallery/images.html**.

are distinct from the pixel-locking ripples. Algorithms for converting range images into elevation maps, assessing elevation maps to produce traversability "goodness" maps, accumulating goodness maps over time, and using goodness maps to plan driving trajectories are discussed at length in Maimone et al. (2006). We summarize overall navigation experience in Section 3.5. Potential for dust accumulation on the lenses to negatively impact stereo was analyzed in Willson et al. (2005b) and determined to be tolerable.

3.4. Visual Odometry

During mission development, the default approach to position estimation for the rovers was to initialize attitude while stationary, using sun sensing and accelerometers (to sense the gravity vector), then to propagate attitude and position while driving using gyros and wheel encoders (Ali et al., 2005). It was recognized that this would be vulnerable to error due to wheel slip. On Mars, there are very few options for other methods to estimate position; there is no global positioning system (GPS) and visual landmarks with known absolute positions cannot be guaranteed to be available. Incremental motion estimation by visual feature tracking ("visual odometry", or VO) with the existing cameras is the only solution that doesn't require adding more hardware to the vehicle. This capability was included in the flight software as an option.

Experience early in the mission showed that slippage on slopes could be severe, could significantly compromise rover safety by causing it to slide into rocks, and could reduce the efficiency of operation by causing the rover to end up far off target at the end of a day's traverse. This was confirmed in testing with Mars analog soil on a tilt table at JPL, which showed nonlinear behavior of slip as a function of slope, slippage rates of 20% on slopes of 10 degrees, and slippage rates of 95% on slopes of 20 degrees (Lindemann and Voorhees, 2005). This was exacerbated when one wheel motor on Spirit began to fail and operators chose to drag it except when absolutely necessary to cross difficult terrain. Opportunity also had one serious incident of getting stuck in a sand drift, in which

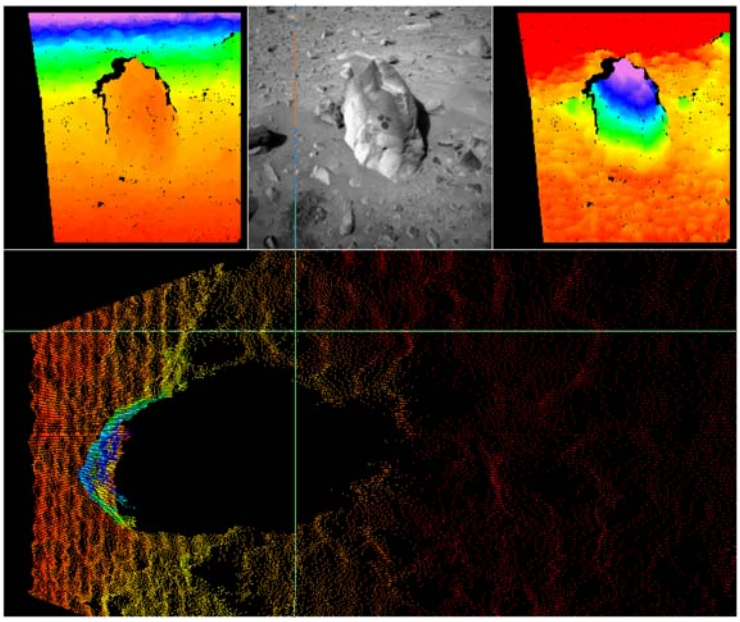


Figure 6. Stereo results from the Spirit navcams, looking at Humphrey rock in Gusev Crater. The rock is about 0.5 m tall. Upper left: false color range image (red is closest, magenta is furthest). Upper right: false color height image (red is lowest, magenta is highest). Bottom: elevation plot, seen from above, where the cameras are at the left looking right (same color coding as the height image). Green cross-hairs are not significant in this image.

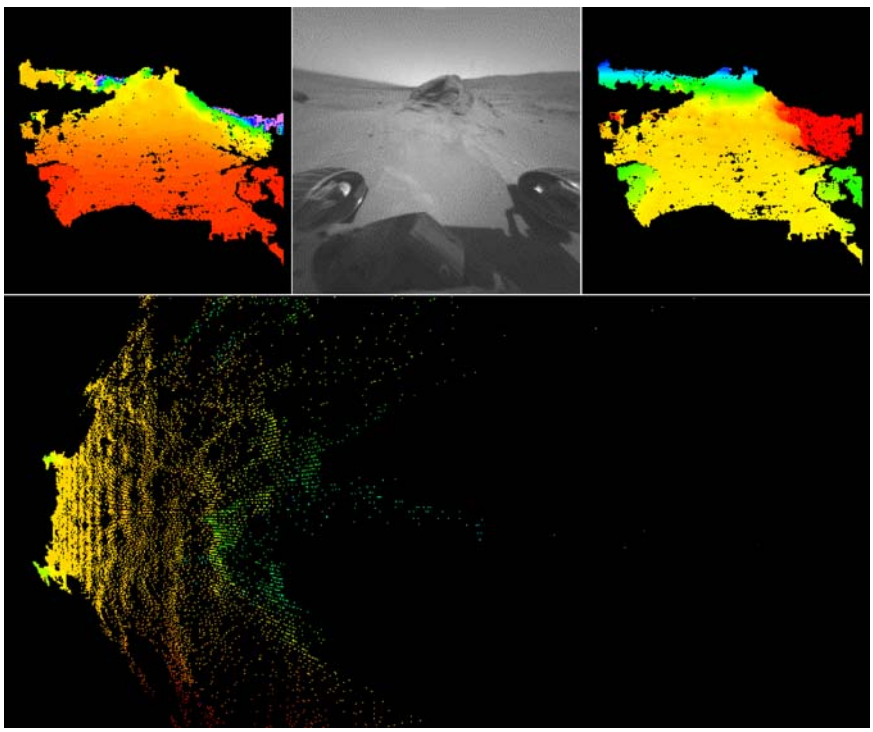


Figure 7. Stereo results from Spirit hazcams, looking at Comanche Spur on top of Husband Hill in the Columbia Hills. The base of the spur is 2.6 m away.

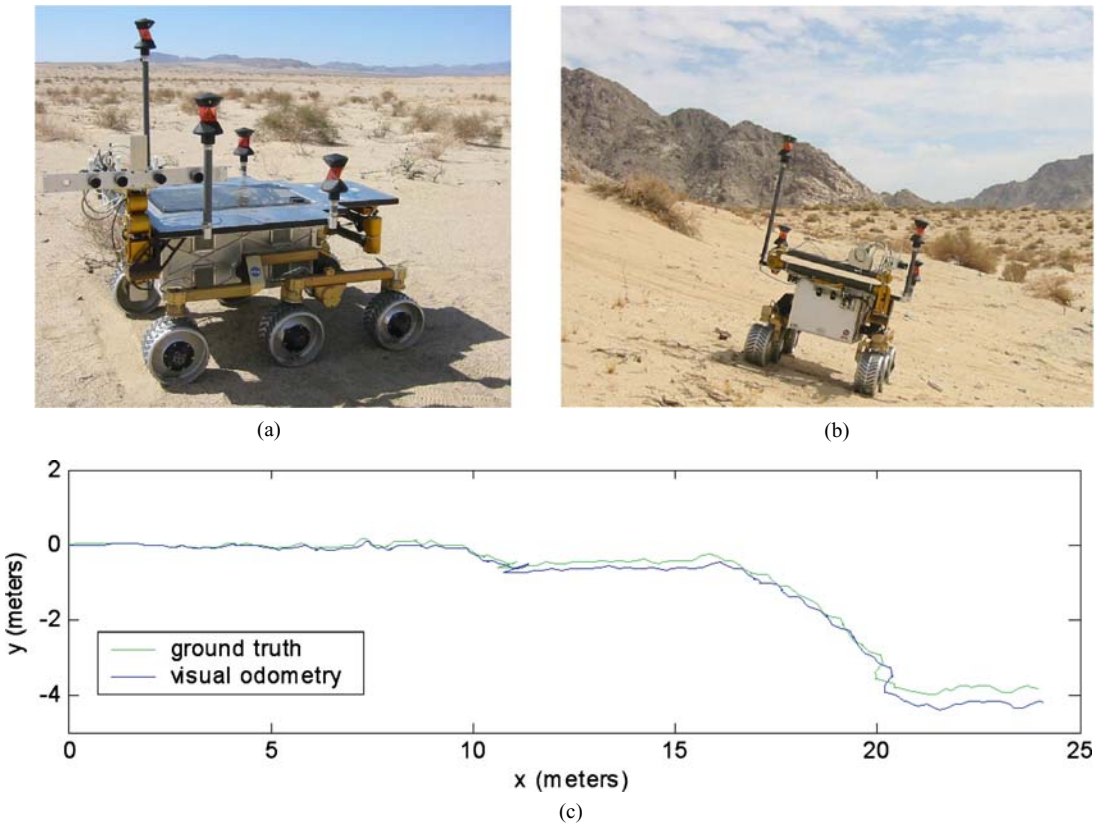


Figure 8. (a) Rocky 8 research rover, showing retro-reflectors used with a total station survey instrument to measure ground truth pose. (b) Rocky 8 on a sandy test site in the Mojave Desert with slopes up to 25°. (c) Plot of ground truth and VO position estimates for a 29 m run on this hillside. The final VO position differs from the true final position by 0.37 m, or 1.3% of the distance traveled.

the wheels spun in place for the equivalent of nearly 50 m of driving while the rover dug itself in. Thereafter, VO was used whenever there was potential for non-trivial slippage or sinkage.

The VO algorithm selects point features, matches these in stereo with multi-resolution correlation to establish their 3-D coordinates, tracks the features in stereo with multi-resolution correlation, and solves for the motion between successive stereo pairs as a 3-D to 3-D pose estimation problem. To make this precise, 3-D Gaussian error models are propagated into the pose estimation formulation. Elements of this algorithm originated with Moravec (1980) and were improved by Matthies and co-workers (Matthies, 1989; Olson, 2003; Helmick et al., 2004; Cheng et al., 2006); we refer readers to these references for details and focus here on discussing performance of the algorithm. Due largely to lack of development time, the MER navigation system does not use a Kalman filter or related algorithm to combine motion information from the IMU, wheel encoders, and visual odometry. Instead, when VO is used, it either completely replaces the motion estimate coming from IMU+wheel odometry or it replaces just the position degrees of freedom—because the gyros in the IMU have very low drift rates and produce good incremental attitude estimates. Future missions are likely to employ a more sophisticated sensor fusion algorithm; they will also include a processor with floating point, which will make such algorithms easier to implement.

This algorithm has been evaluated by Monte Carlo simulation, by outdoor testing on rovers with accurately surveyed ground truth, and by operation on Mars. Olson used Monte Carlo simulations to examine position error growth with time, to show that integrating absolute heading sensors (e.g. sun sensing) with VO achieved linear position error growth, and to find an optimal baseline for VO (Olson, 2003). In outdoor testing with an urban robot vehicle using differential GPS as ground truth, he observed a position error of 1.2% of the distance traveled over 210 stereo pairs covering 20 m. This was in good agreement with the predictions of his Monte Carlo simulations. More recent outdoor testing with Mars rover research vehicles has obtained ground truth position and attitude data with a laser-based survey instrument called a "total station," which measures range to retro-reflectors on the rover with sub-centimeter accuracy (Fig. 8(a)). The total station is motorized, automatically tracks one retro-reflector as the rover moves, and automatically finds

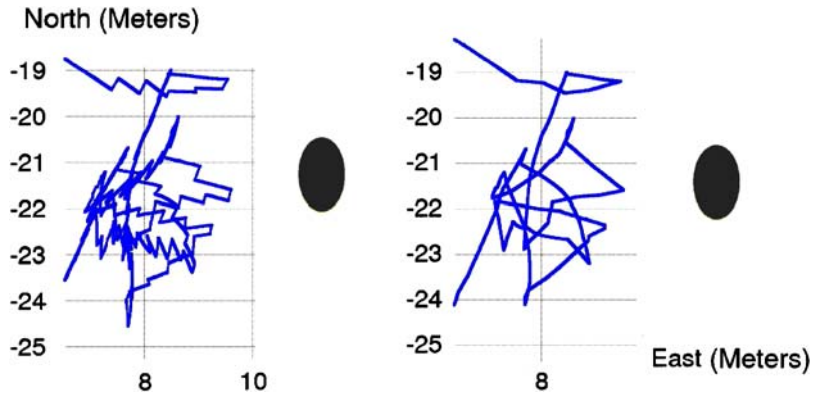


Figure 9. Opportunity's 15 sol trajectory near Wopmay Rock in Endurance Crater. The rock is approximately at the grey ellipse. The vehicle started in the upper left, first driving toward the rock and then trying to get around or away from the rock. The downslope direction is at about 1 o'clock. In the left plot, after each step, wheel odometry puts the rover further upslope than it really is; the "jumps" pointing toward 1 o'clock are the result of VO adjusting the rovers' position downslope to reflect slip during that step. The right plot shows the same course using the VO corrections to approximate the rover's true path.

the other retro-reflectors when the rover stops to determine the full 6-DOF pose of the vehicle. We estimate that this gives ground truth with a precision of 2 mm for position and 0.2° for attitude. In multiple outdoor tests of 25 m to 29 m in length, final position errors were between 1.3% and 2.5% of distance traveled (Helmick et al., 2004). Figure 8(b) shows the rover on one of the outdoor test sites. For this test, Fig. 8(c) plots the position trajectory from the ground truth system and from VO; the final VO position was accurate to 1.3% of the total distance traveled. VO provided position updates approximately every 20–30 cm in this test. A closely related algorithm developed at LAAS-CNRS in the late 1990s is described in Mallet et al. (2000), where they report position errors of 1 m over 25 m (4%) in outdoor experiments.

Results of using this algorithm on Mars are discussed in Cheng et al. (2006). It has been used on both rovers with navcam imagery. Vehicle motion between images was specified to ensure at least 60% overlap between adjacent images, which led to forward steps of no more than 75 cm and turns-in-place of no more than 18 degrees per step. Runtime of the algorithm on the flight processor with all other flight software operating averaged 150 sec/step, with a large standard deviation, while tracking on the order of 100 features. With no other flight software running, this comes down to 21 sec/step. For perspective, on a 3.6 GHz Pentium 4 workstation we are able to run essentially the same algorithm, tracking 266 features to an image resolution of 640 × 480 pixels, simultaneous with SAD stereo at 320 × 240 resolution with 32 disparities, and achieve 15 frames/sec. As of March 4, 2005, in use on both rovers on Mars, VO had converged to a solution successfully1418 times in 1484 cases, or 95.5% of the time, tracking an average of about 80 features per frame. Failures to converge appear to be mostly cases where vehicle motion was too large for effective feature tracking. Except for a few cases of improper parameter settings, only one case had been observed where the algorithm reported success but Earth-based checking showed a clearly bad estimate. This occurred when the tracked feature set was dominated by features on the rover shadow.

VO was used on Mars in situations involving potential slip on slopes, potential for sinking in sand drifts, and to ensure accurate approach to rocks against which the science team wished to place instruments. One of the more striking instances of its use happened when Opportunity was in Endurance Crater, operating on a 17–20° slope with loose sand and buried rocks near a large rock ("Wopmay") of interest to the science team. As Opportunity first tried to approach, then struggled over several days to escape this rock, VO was used to make sure the rover did not slip onto the rock. Figure 9 shows rover trajectories estimated by wheel odometry and VO for this sequence of drives. In the left plot, the vehicle started in the upper left corner and drove toward the rock. The first line segment shows the first step as estimated by wheel odometry; the second line segment shows a position correction applied by VO reflecting downslope slip. The rest of path similarly shows steps as estimated by wheel odometry and corrected by VO. The right plot uses the VO corrections to approximate the true vehicle path continuously, instead of applying discrete corrections after each step.

Of course, visual odometry drifts over time, with position error growing faster as heading error grows. For MER, heading is updated periodically by sun sensing (Ali et al., 2005), since there is not enough magnetic field for a compass. The whole history of the rovers' trajectories are also estimated on Earth by bundle adjustment, using overlapping navcam images acquired periodically and using manually-assisted tie-point matching (Di et al., 2005; Li et al., 2006).

Algorithms for missions must reach a high degree of maturity several years before launch, so MER's VO represents fairly mature, tested software from the late 1990s. More recent research on VO has used other error functions and feature tracking methods with an emphasis on real-time performance on Pentium architectures (Nister et al., 2006). Motion estimation performance appears to be comparable, though a direct comparison on common data sets has not been made. Another more recent trend in the literature has been the use of monocular or stereo image sequences in simultaneous localization and mapping (SLAM) problems, using Harris or SIFT and sometimes integrating inertial sensors (Eustice et al., 2005; Se et al., 2002). Formulations for integration of multiple sensors for localization are definitely germane to rover navigation. However, rover navigation is typically a one-way traverse, whereas much research on SLAM and SIFT landmarks addresses loop closure and contexts where landmarks might be revisited many times, so feature invariance and path topology are less important for rovers than computational efficiency and the ability to achieve order of 1% error in a linear traverse.

3.5. Overall Mars Navigation Results

Driving mode statistics through August 15, 2005, are given in Maimone et al. (2006). At that point, Spirit had been on Mars for 573 sols and had driven 4798 m; Opportunity had been on Mars for 555 sols and had driven 5947 m. Spirit used stereo vision for obstacle detection for 28% of its total traverse and VO for 16%; obstacle detection and VO were only used together for 3 m, because that was too slow. Opportunity used stereo vision for obstacle detection for 21% of its traverse and VO for 11%; it never used obstacle detection and VO together. The rest of the driving was commanded as "blind" drives, because operators on Earth could see in navcam or pancam imagery that the way was clear and chose to travel faster by not incurring the runtime delay for the autonomous navigation functions. Spirit drove roughly 3 km on the floor of Gusev Crater to the base of the Columbia Hills, which are about 100 m high, and has been in the hills ever since. Most of its obstacle detection was done on the floor of the crater, as part of drives in which an initial segment that operators could see was blind and beyond which obstacle detection was enabled. Most of its VO was done in the hills. Opportunity used obstacle detection during very long daily drives on the plains, mostly to avoid driving into small craters. It used VO on sloping terrain in Eagle Crater, Endurance Crater, and on the plains as it entered terrain with sand drifts, as a precaution against getting stuck with wheels spinning deeply in sand. As of March 15, 2006, Spirit had covered 6797 m and Opportunity 6645 m; driving mode statistics up to that point are not yet available. Detailed narrations of driving experience for both rovers are given in Leger et al. (2005) and Biesiadeck et al. (2005).

Since the goal for the primary mission was to survive 90 days and to drive a minimum of 600 m, the rovers have far surpassed that goal. Onboard computer vision algorithms have been instrumental in that achievement. Moreover, because long drives were necessary to reach scientifically interesting terrain, MER has firmly established the importance of long range mobility for Mars surface exploration.

3.6. Lessons Learned

Before moving to a discussion of ongoing development, we will briefly recap key requirements for rover vision systems and lessons learned from the MER mission. Rovers drive slowly and the terrain typically is not very complex, so lookahead distance requirements for obstacle detection are short. Power and energy are limited, which puts a premium on low-power electronics. Rover driving is generally confined to a few hours around midday for operational reasons largely unrelated to the vision system. Simple, area-based feature tracking and stereo correlation algorithms generally have proved sufficient for velocity estimation for landers and for obstacle detection and visual odometry for rovers.

While MER has been a very successful mission despite its slow computer, future missions will want to drive a lot farther in a given amount of time, so faster implementations of vision and planning algorithms are important. Several low-level data quality issues need to be addressed in stereo vision, including maximizing ability to match in low-texture terrain such as that encountered at Meridiani Planum and reducing pixel-locking artifacts. One of the key performance limitations in this mission has been slip on slopes. Coping with this consumed a great deal of time when Opportunity was in Endurance Crater and prevented Opportunity from making a closer approach to scientifically interesting cliff outcrops in the crater. This could be improved by making visual odometry fast enough to use in a continuous steering control algorithm that would compensate for slip.

4. Current Development

JPL is designing a new rover, called the Mars Science Laboratory (MSL), for the 2009 Mars launch window. This design is powered by a radio-isotope thermal generator (RTG) instead of solar panels, is much larger than MER, and is intended to be able to drive 20 km in a primary mission of nearly two Earth years. Stereo cameras are again the sole 3-D perception sensors onboard. The flight computer is a space-qualified PowerPC 750 with a clock rate of about 100 MHz; this brings floating point,

but still no L2 cache. Budget priorities dictate that MSL's autonomous navigation capabilities will be little changed from MER. Nevertheless, vision system R&D is continuing along several lines in parallel to MSL development. This includes vastly increasing computational throughput by exploiting flight qualifiable field programmable gate arrays (FPGAs) as computing elements, improving the quality of disparity data by reducing noise, improving performance at occluding boundaries, and reducing pixel locking, and attempting to learn to predict slippage ahead of the vehicle by regressing past slippage experience against the visual appearance of the terrain.

4.1. FPGA Computing

JPL is developing a new flight computing system, called the Mobility Avionics Module (MAM), around the Xylinx Virtex-II Pro FPGA as the main computing element. This FPGA includes PowerPC 405 (PPC405) processor hard cores that can be clocked at up to 300 MHz; the rest of the FPGA logic can be clocked at approximately 100 MHz. A prototype MAM exists as a PMC form factor ($\sim 6 \times 3$ inch) board that weighs 150 g, draws 3 W, and includes a Virtex II Pro with two embedded PPC405 cores running Linux, with Ethernet, PCI bus, Compact Flash, 1394, and serial I/O. This board has been installed in a research rover and the entire rover onboard software has been ported to one PPC405 in the FPGA. Benchmarks show that stereo vision and VO run together in 6.4 sec/frame for 640×480 input imagery, 320×240 disparity maps with 32 disparities, and VO tracking 266 features in a four-level image pyramid. This is a vast speed-up over the MER flight processor.

To increase speed further, we are moving the most time-consuming vision functions into the FPGA logic. Our initial step has been to move the SAD computation for stereo vision into the logic, since that greatly dominates the runtime of the stereo algorithm. The logic implementation has been designed to be highly parallel and pipelined, so that in steady state it produces one subpixel disparity estimate per clock cycle at a 100 MHz clock rate (Villalpando, 2006). The current implementation is hard-coded for 320×240 pixel imagery with 32 disparities. SAD scores for all 32 disparities are computed in parallel, then flowed into a comparator tree that determines the minimum SAD at each pixel and performs a left-right consistency check. Finally, subpixel disparity is computed by parabolic fitting and output as 8 bits of integer and 8 bits of subpixel data. The implementation uses roughly 40% of the resources in a Virtex II Pro model 2VP4 FPGA. At one output pixel per clock cycle, the pipeline throughput rate is equivalent to roughly 1 ms/frame. The current I/O implementation does not use DMA, so with I/O one frame takes 50 ms. Together with running rectification, prefiltering, and triangulation on the PPC405, the entire stereo process takes 250 ms/frame.

Related work on FPGA and other hardware-based stereo implementations includes an FPGA implementation of another SAD-based algorithm similar to ours ([STOC] Stereo Head User Manual 1.1, 2006), a custom ASIC implementation of the Census stereo algorithm (Woodfill et al., 2004), and an implementation of a multiscale, multi-orientation, phase-correlation algorithm in multiple FPGAs (Masrani and MacLean, 2006). The first two are commercial products; all three claim speeds of 30 Hz operation or greater on images of 512×480 or 640×480 pixels. All of these are purely stereo machines and do not include a general purpose processor core with an operating system that can host other application functions coded in C, as the FPGA in our mobility avionics module does, so they were designed for more narrow purposes. The key points to note about our work are that (1) it is becoming possible to fly high capacity FPGAs in space, including embedded microprocessors and large amounts of reconfigurable logic in the same chip, and (2) such chips support area-based stereo in a few milliseconds/frame. A detailed, engineering-oriented comparison of the chip real estate versus speed versus disparity map quality of different real-time stereo algorithms would be valuable, but is beyond our current scope.

4.2. Noise Reduction and Improved Occluding Boundaries

As engineering improvements, we are working to improve disparity estimation at rock boundaries and to improve the signal-to-noise ratio (SNR) of the disparity maps, within the limits of real-time performance.

For better boundary estimation, we have had good success with using five overlapping SAD windows ("SAD5") (Hirschmuller et al., 2002). In initial work on a real-time implementation, we have a SAD5 correlator running at 30 Hz on a 1.4 GHz Pentium M for 320×240 disparity maps and 32 disparities. We got further improvement at occluding boundaries by changing the image prefiltering used to compensate for radiometric differences between left and right images. SAD is normally run on images that have been bandpass or highpass filtered to reduce brightness differences between the stereo images, which arise from a number of causes. This filtering introduces overshoot or ringing artifacts at strong intensity edges. When such edges correspond to occluding boundaries, the artifacts affect different parts of the background surface in the left and right images, which contributes to poor matching at occluding boundaries. We have addressed this by replacing linear filters with a fast, separable approximation to the edge-preserving bilateral filter; that is, the bilateral-filtered image is subtracted

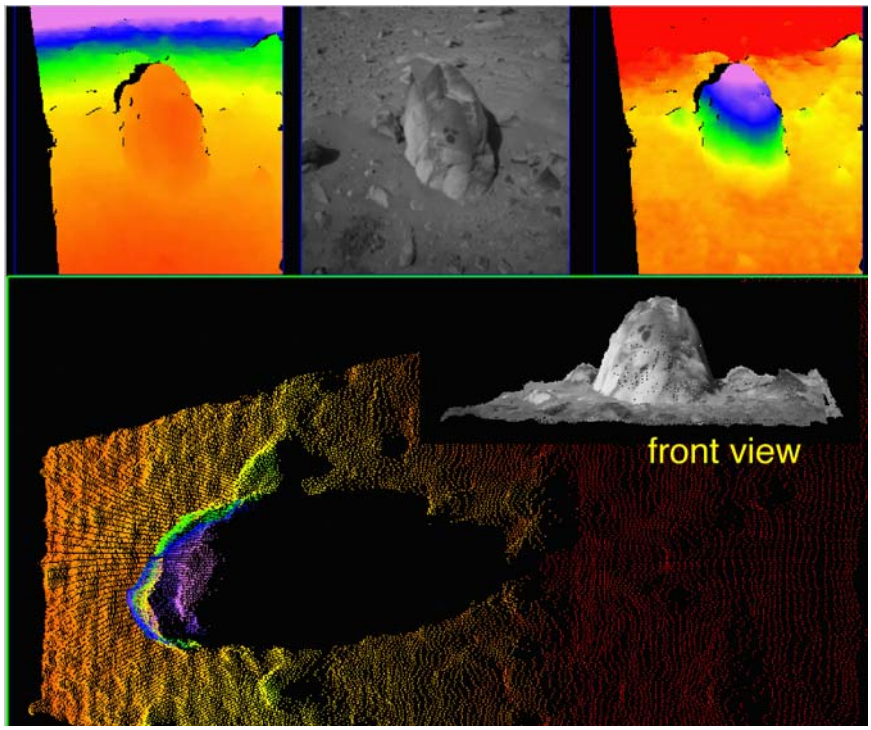


Figure 10. Stereo results on the same Spirit navcam imagery as Fig. 6, with SAD5 and improved low-level signal processing (see text). The occluding boundary of the rock is cleaner, there are fewer holes in the disparity map, the small rocks are better delineated, the shadow to the right of the large rock gets the disparity of the ground instead of the rock, and pixel-locking is slightly reduced.

from the original to reduce radiometric differences between left and right images. Our experiments show that this produces better depth maps for a variety of outdoor scenes (Ansar et al., 2004).

In most of the research community, stereo evaluation has focused on estimating integer disparity and scoring the fraction of pixels with correct integer disparity. However, for autonomous navigation applications, subpixel estimation is essential, so accuracy and precision of the subpixel estimates is a key performance issue. This is affected by such factors as aliasing caused by simplifications introduced to low-level filtering for real-time performance, pixel-locking, and other errors caused by correlation windows overlapping strong intensity edges or occluding boundaries. In the MER stereo algorithm, aliasing arises from the use of bilinear interpolation in rectification and the use of block averaging in the high-pass filter used to compensate for radiometric differences between images. We have not undertaken a detailed frequency domain analysis and redesign to address this, but we have experimented with bicubic interpolation in rectification and better highpass filter kernels.

Figure 10 illustrates the effect on the MER Humphrey Rock stereo pair of using bicubic interpolation in rectification, bilateral filtering for preprocessing, and SAD5 for correlation (see Fig. 5 for comparison). The occluding boundary of the rock is cleaner, there are fewer small holes, the small rocks are better defined, and the shadow on the right edge of the rock gets the disparity of the ground instead of the rock. Pixel-locking is also slightly reduced, though this is hard to see in this figure; we address this in more detail in Section 4.3. To begin to quantify the difference, we have taken 10 left navcam images from Meridiani Planum, with appearances similar to portions of Fig. 4(b), shifted them to create synthetic right images with disparity gradients corresponding to a flat ground plane seen from the navcam, and collected RMS disparity error statistics for the stock MER algorithm and the modified version described above. This yielded RMS errors of 0.16 pixels and 0.09 pixels, respectively. While this is just a start for quantitative evaluation, reducing the noise in the disparity map by nearly a factor of two is significant for practical applications and indicates that this issue is worth further examination.

4.3. Reduced Pixel-Locking

We have recently conducted a short study of pixel-locking (Stein et al., 2006), the phenomenon in which disparities estimated by window-based stereo algorithms are biased toward integer disparities. It is desirable to reduce this effect in order to increase the feasible lookahead distance for obstacle detection for a given stereo camera set-up.

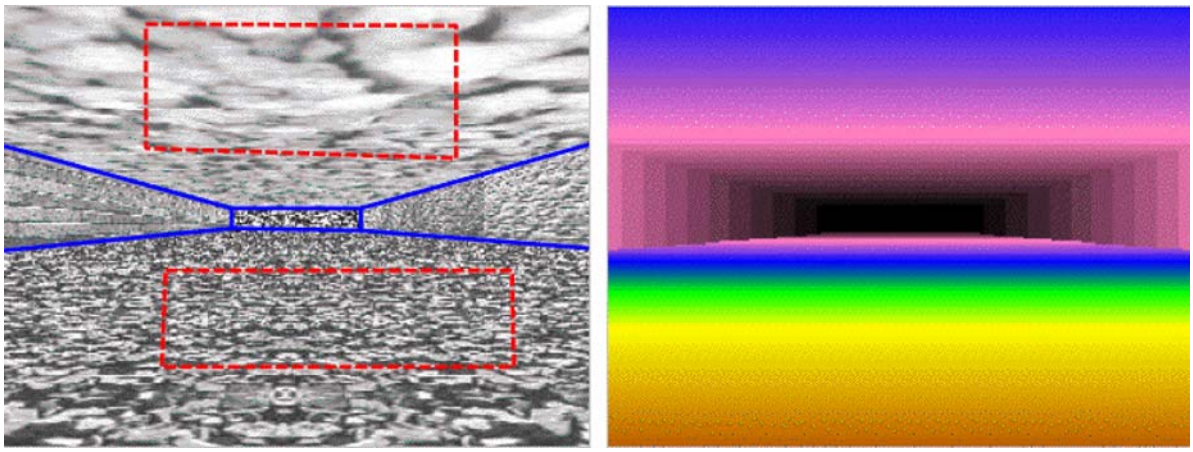


Figure 11. Left image of a texture-mapped synthetic room and the corresponding ground truth disparity map. The blue lines are for visual aid only, to help distinguish the surfaces in the room. The red dashed rectangles indicate the ceiling/floor regions used for analysis (see text). Also note that any stair-stepped pattern visible in the disparity map is due to color quantization; the disparities do indeed vary smoothly.

This effect was noted at least as far back as Gennery (1980). Xiong explained it for SSD matching through a frequency domain analysis and showed modest improvement by replacing parabolic fits with quartic fits to estimate subpixel disparity (Xiong and Matthies, 1997). Shimizu and Okutomi gave an analysis for both SSD and SAD matching and showed that SSD suffers less from this effect (Shimizu and Okutomi, 2001). Since the bias is approximately anti-symmetric around integer disparity values, they proposed to reduce the effect by creating new image pairs sampled halfway between the original pixels, doing subpixel stereo matching again with the resampled imagery, and averaging the results with disparities obtained from the original imagery. In their experiments, that reduced errors to 1/6 to 1/8 of the standard method. Szeliski and Scharstein also analyzed SSD matching in the frequency domain, though not with an explicit focus on pixel-locking, and proposed stereo matching approaches based on interpolating imagery to higher resolution (Szeliski and Scharstein, 2002). Their algorithm had a number of complexities and was not targeted at real-time systems. The main conclusion of our study is that iterative affine matching reduces the RMS disparity error on planar surfaces and attenuates the rippling artifacts from pixel-locking more than the Shimizu-Okutomi method. The difference appears to increase with increasing surface tilt, which is both intuitive and significant for ground vehicle navigation, where a steeply tilted ground surface (relative to the camera frame) is the norm. To date, we have not addressed runtime issues in this study, so prospects for real-time performance of iterative affine matching for subpixel stereo disparity estimation remain to be evaluated.

Our experiments used standard SAD stereo to estimate integer disparities, then used affine matching constrained to 1-D (along scanlines) to estimate subpixel disparity, allowing X translation, scale, and shear deformations of the template and iterating to convergence. This allows initial disparity errors of greater than one pixel to be corrected, which are quite possible in highly foreshortened parts of the image. Standard parabola fitting cannot correct errors greater than 0.5 pixels. The use of discrete windows for summing matching error required a few additional practical considerations. First, to avoid artifacts due to sharp changes in matching cost as the window's size or position change, we used a Gaussian-shaped weighting function. In addition, we ignored any pixels (by zero-weighting them) which are marked as occluded or invalid in the initial integer disparity result. Finally, if the window straddles an occlusion boundary, the center pixel's computed offset will be influenced by pixels from a different physical surface in the image. Since we have an initial estimate of disparity (and thus occlusion boundaries in the scene), we also ignore any pixels in the window whose initial integer disparity differs radically from that of the central pixel whose offset we are computing. To cope with local brightness differences between the images, the matcher was applied after the difference of bilateral filtering discussed earlier. For direct comparison to Shimizu-Okutomi, we used squared differences instead of absolute differences in the affine matcher. In Stein et al. (2006), we referred to this method as "affine LK," for affine Lucas-Kanade.

We compared this approach to simple parabola fitting and to our implementation of the error compensation scheme of Shimizu-Okutomi. To allow quantitative analysis, we first evaluated results on synthetic imagery. The left image of a synthetic stereo pair along with the corresponding ground truth disparity map are shown in Fig. 11, depicting a rectangular room with various

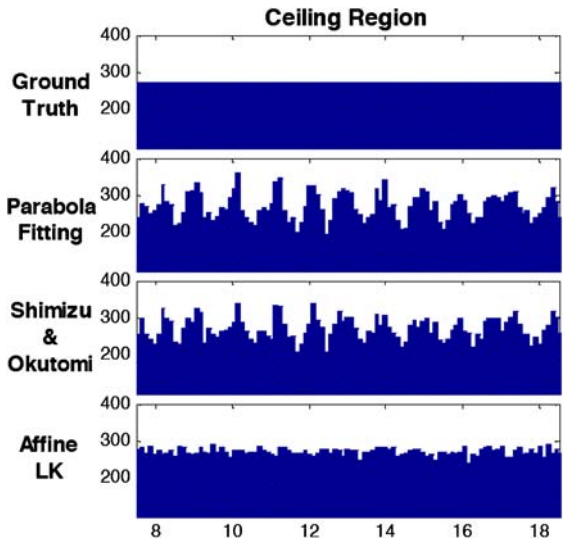


Figure 12. Histograms of subpixel disparity from the outlined region on the ceiling of the synthetic room.

textures mapped to the planar ceiling, floor, and walls. Because the four surfaces in the room are planar, the true distribution of subpixel disparities along each should be uniform. A histogram of ground truth subpixel disparities for a region on the room's ceiling is shown at the top of Fig. 12.

Initial integer disparities were computed for the pair as described above. Windows for all results provided were 7×7 pixels. After applying parabola fitting to estimate subpixel disparity, we see in Fig. 12 the typical pixel-locking effect as peaks of subpixel disparity. Applying the approach of Shimizu and Okutomi (2001) improves the "peakiness" somewhat, and using Lucas-Kanade with with constant windows yields similar results. But once we enable affine warping of the windows, we see the more dramatic flattening of the histogram shown at the bottom of the figure, which more closely resembles the ground truth distribution. The floor region produces similar results.

In addition to comparing disparity distributions, which shed light on the pixel-locking performance, it is important to check the actual disparity errors as well. For the initial integer disparity map shown in Fig. 13(a), consider its absolute error versus the ground truth disparity map, shown in Fig. 13(b). The errors have been capped at 0.5 to de-emphasize outliers and highlight initial smaller errors. Note that most of the errors on the floor start larger than 0.5, which is most likely due to significant foreshortening since the synthetic camera is placed fairly close to the ground plane. The error maps after subpixel disparity estimation by parabola fitting, by the approach in Shimizu and Okutomi (2001), and by the affine LK method are shown in Fig. 13(c)–(e), respectively. Note that errors on the floor are significantly reduced only when using the affine matcher.

Figure 14 compares the RMS error over the ceiling and floor regions for the various approaches. To suppress the influence of outliers, we ignore all pixels whose initial integer disparity error was greater than 3 when computing the following RMS values (note that the choice of this threshold does not radically alter the results). For the ceiling, both parabola fitting and the method in Shimizu and Okutomi (2001) do reduce the error from the initial integer estimates by about 65%. The affine LK approach reduces the error by 78%. On the floor, where the errors are much higher initially, both methods based on parabolas yield only about 6% error reduction over the integer estimates, whereas the affine LK approach achieves 86%.

Figure 15 shows results on a real stereo image pair, taken from outdoor experiments in a field. We do not have ground truth for this pair, but because the ground is roughly planar, we can expect a smooth distribution of disparity. The pixel-locking effect is clearly visible for simple parabola fitting. The Shimizu-Okutomi method shows a marked improvement for this pair, but the affine LK approach still produces the smoothest histogram. Figure 16 shows similar results on a difference image pair.

Figure 17 shows oblique overhead views of 3-D reconstructions for portions of the scenes in Figs. 15 and 16. Pixel-locking ripples are clear in the parabola fitting results and reduced in the Shimizu-Okutomi and affine LK results. The 3-D reconstruction for the imagery in Fig. 16 includes part of the nearest tree and the fallen log in the lower left corner of the image.

Overall, the affine LK approach produced results that varied from modestly to significantly better than the Shimizu-Okutomi method, depending on surface tilt and other factors that were not controlled in these experiments. The affine LK approach adapts to the local slant of the underlying surface, which makes it appealing compared to simpler schemes like using a fixed, tilted horopter to rectify imagery before matching (Burt et al., 1995). It is likely to reduce potential for false alarm obstacle detections on the ground. We have not yet tested the method on uneven ground to ensure that true relief is preserved. Given that we still require a real-time implementation, the next phase of this work must weigh implementation costs against the noise reduction realized for the various approaches to ameliorating pixel-locking.

4.4. Vision-Based Slip Prediction

One of the most significant mobility limitations to emerge from MER is slippage on slopes. Using VO to improve state estimation has already helped control steering to offset slippage, but it would be valuable to be able to predict in advance when a slope might be slippery, so as to make more intelligent path planning decisions about

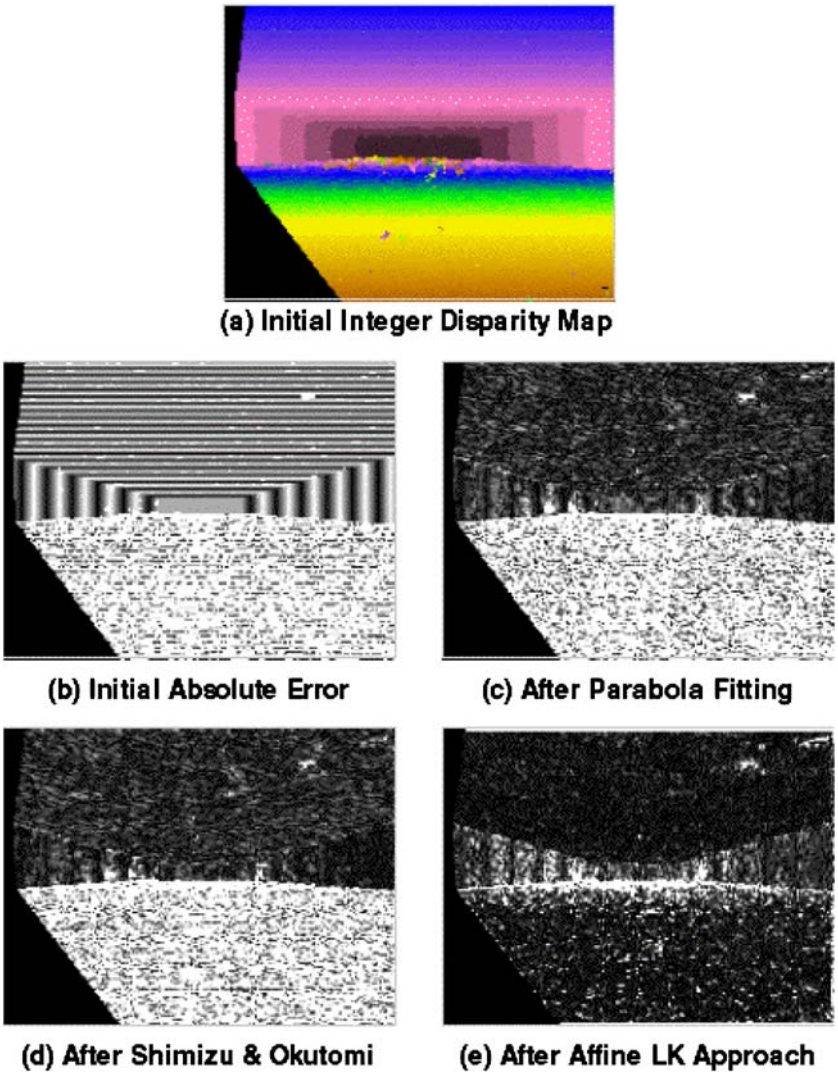


Figure 13. Absolute disparity errors for the synthetic room imagery, capped at 0.5. The large initial errors on the floor are corrected only using the affine LK approach.

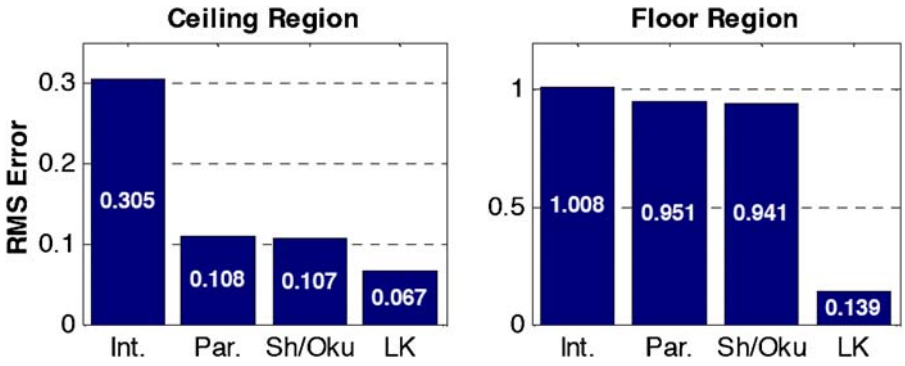


Figure 14. RMS subpixel disparity errors from the floor and ceiling regions of the synthetic room imagery.

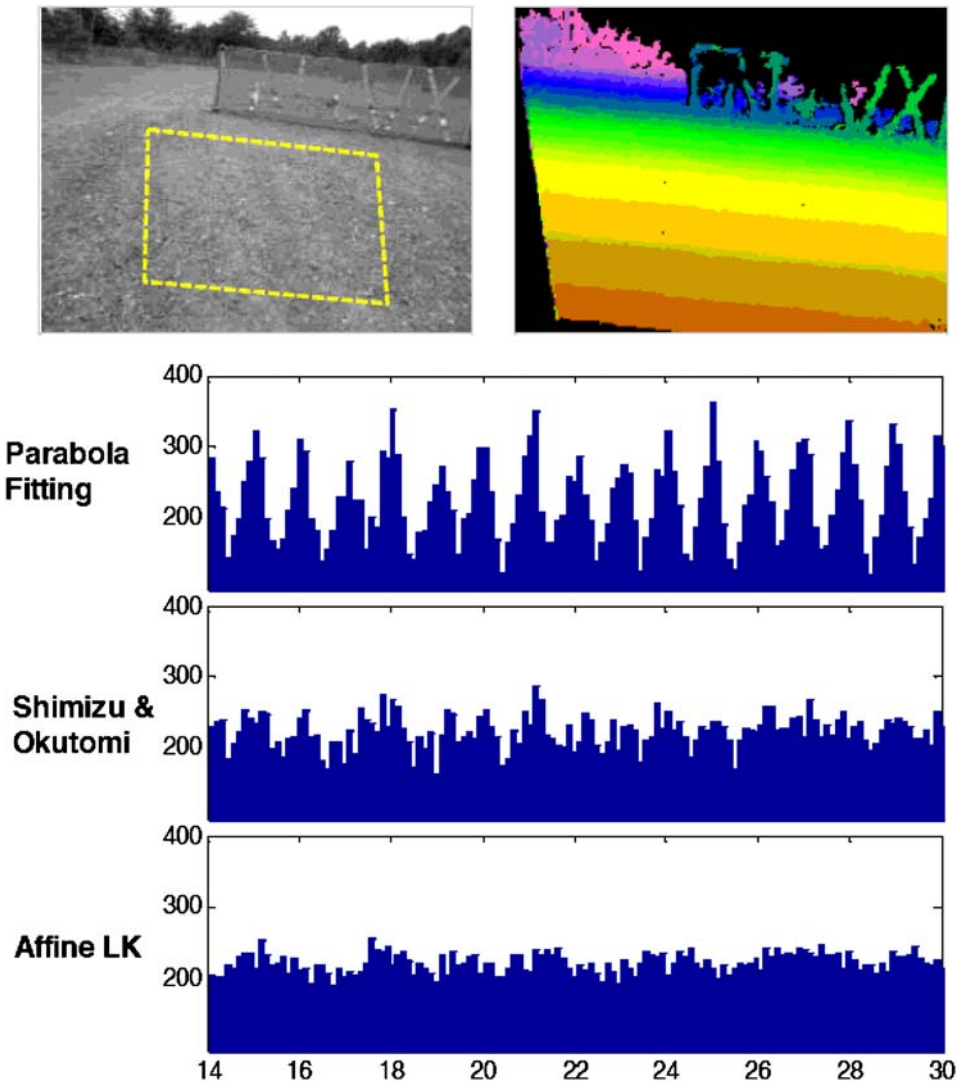


Figure 15. The left image of a real stereo pair, initial integer disparity map, and subpixel disparity distributions for the ground region (yellow dashed lines). The Shimizu-Okutomi method improves the results considerably, but affine LK does slightly better.

it. Attempting to predict slip from physical models of wheel/soil interaction would be complex, particularly given the heterogeneity and spatial variation of real terrain surfaces and the unknown parameters of such models that would have to be estimated first. Existing work on modeling wheel/soil interaction typically assumes homogeneous terrain and simplifies the physical model in addition (Andrade et al., 1998; Iagnemma et al., 2004). Therefore, we have begun to tackle slip prediction as a visual learning problem that, for the time being, avoids modeling wheel/soil interaction (Angelova, 2006a, b).

Our approach starts by building gridded elevation maps of the terrain in front of the rover that record geometric and visual appearance features of the terrain. As the rover drives over mapped areas, visual odometry is compared to wheel odometry to estimate slip of the whole rover as a unit. Pairing the geometric and visual appearance features (e.g. texture) of map cells with the measured slip in each cell enables learning a local, nonlinear regression function that maps from the features to predicted rover slip. Since the input space of terrain features could be high dimensional, to limit the complexity of the problem we currently partition it as follows. We assume that terrain consists of a number of distinct classes, where each class can be recognized by visual features such as color and texture, and we assume that slippage for each class is only a function of slope. From labeled training data, we train a classifier to recognize the terrain class in each map cell; then for each class, we learn a regression function that maps from slope to slip.

To evaluate the feasibility of this approach, we first experimented with predicting slip from slope for a single, known terrain class (Angelova et al., 2006a), then extended our experiments to situations where the set of

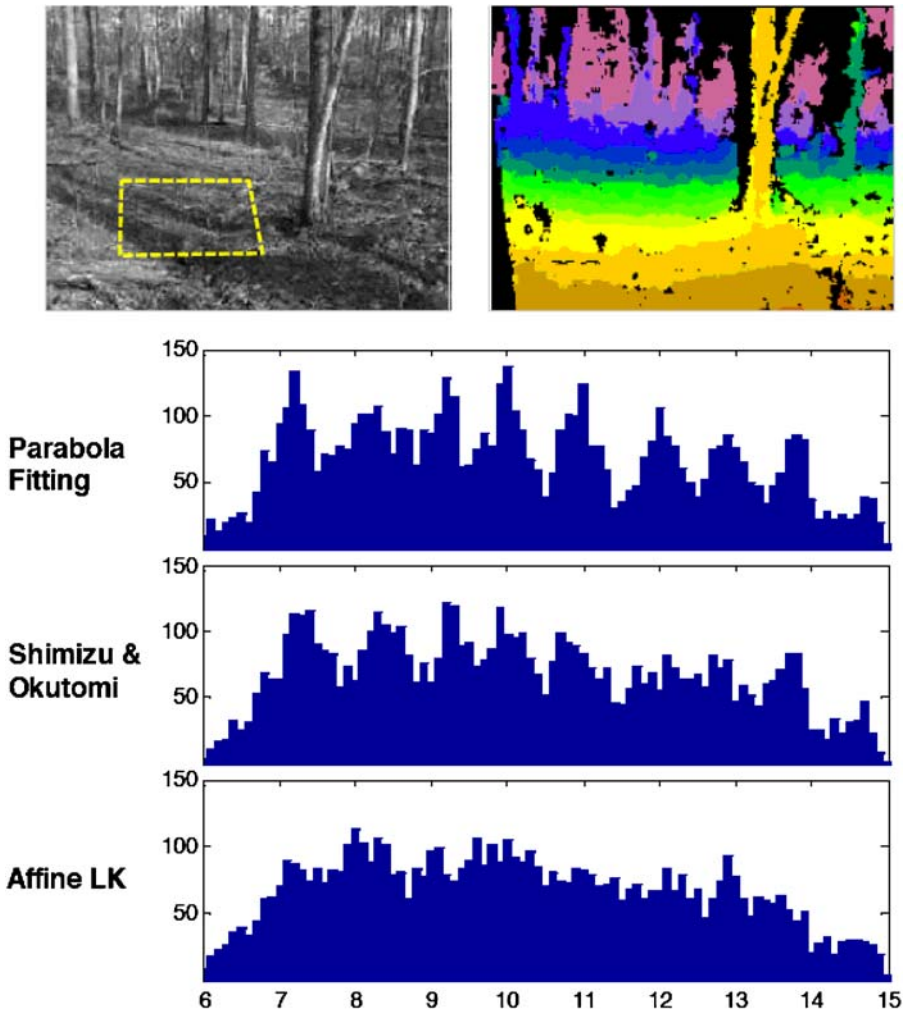


Figure 16. The left image of a real stereo pair, initial integer disparity map, and subpixel disparity distributions for the ground region (yellow dashed lines). Here the Shimizu-Okutomi method makes modest improvement and affine LK is again the smoothest.

classes and labeled training data were given (Angelova et al., 2006b). Ultimately, our goal is to learn the set of classes from experience. However, *a priori* it was unclear how well slip can be predicted from slope for even just one, known terrain class, so we tested this first, as follows.

We estimate slope by fitting a plane to the average elevation of map cells in neighborhoods in front of the rover, and decompose slope into pitch and roll angles aligned with the rover reference frame. We consider slip learning as nonlinear function approximation. While many learning algorithms can be applied to this problem, our choice is motivated by our goal to eventually have the algorithm running onboard a rover with the ability to do fast, online updates. We selected a type of Receptive Field Regression algorithm (Schaal and Atkeson, 1998) called Locally Weighted Projection Regression (Vijayakumar et al., 2005). The main idea is to split the input domains into sub-regions (called receptive fields), apply local linear fits in each receptive field, and do weighted blending of these to approximate a globally nonlinear function.

Slip S (as a function of terrain slopes $\mathbf{x}$) can be written in the following form:

$$\hat{S}(\mathbf{x}) = \sum_{c=1}^{C} e^{-\|\mathbf{x}-\mathbf{x_c}\|^2/\lambda} \left(b_0^c + \sum_{r=1}^{R} b_r^c \left\langle \mathbf{d}_r^c,\ \mathbf{x} \right\rangle \right)$$

where $e^{-\|\mathbf{x}-\mathbf{x_c}\|^2/\lambda}$ is a weighting function that depends on the distance from the query example $\mathbf{x}$ to the receptive field center $\mathbf{x_c}$, C is the number of receptive fields for this dataset, $b_0^c, b_r^c, \mathbf{d}_r^c$ are the parameters of the linear regression in each receptive field, and λ controls the receptive field size and in our case is estimated using a validation set (to avoid overfitting). Given the training data $\mathbf{D} = \{\mathbf{x}_i,\ S_i\}_{i=1}^N$, where the vector $\mathbf{x}_i$ contains the

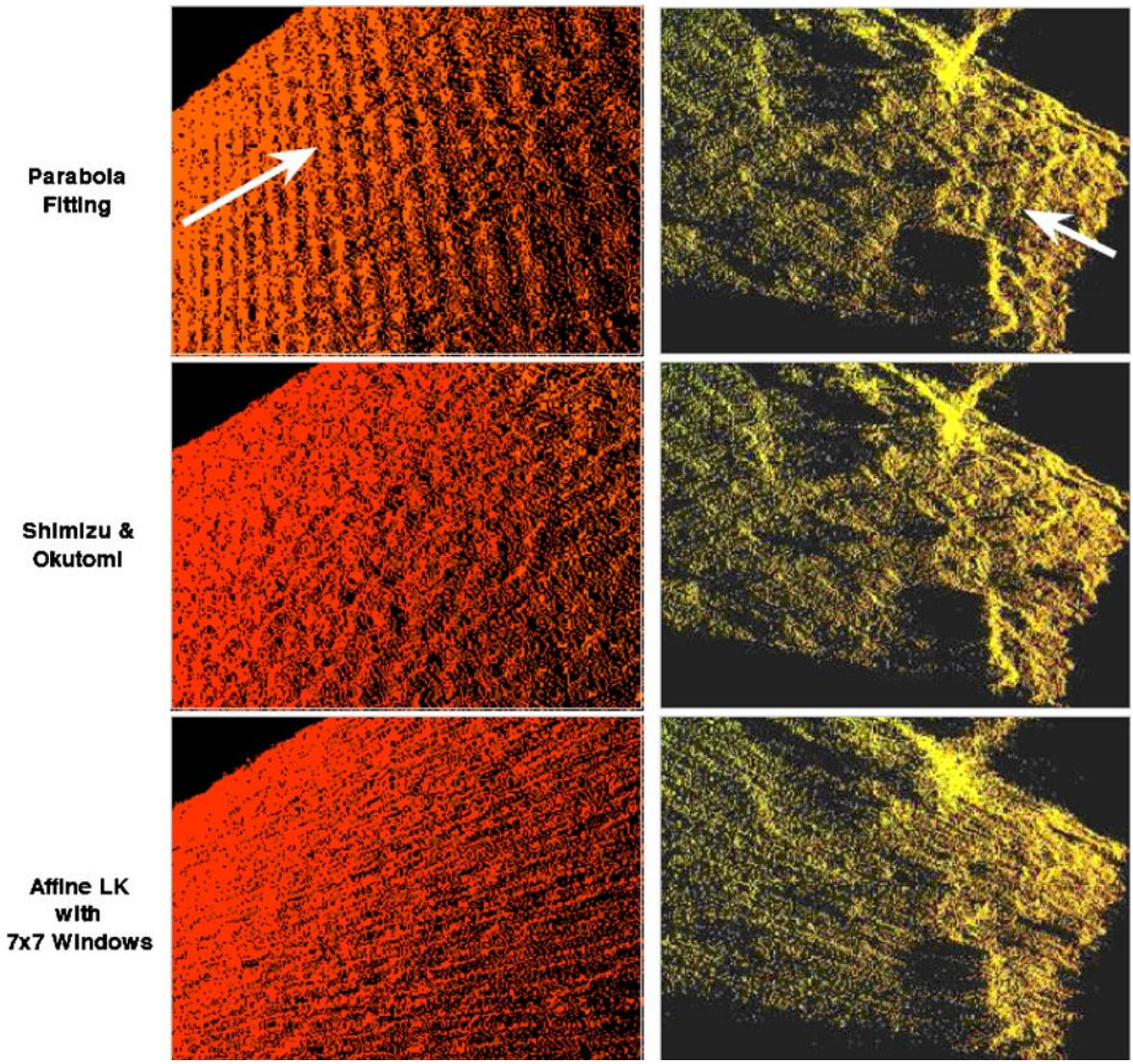


Figure 17. Oblique overhead views of the 3-D reconstructions from subpixel disparity maps for the imagery in Fig. 14 (left column) and Fig. 15 (right column), with camera viewing directions indicated by white arrorws.

estimated slopes from range imagery and S_i are the corresponding slip measurements at this particular location, the task of the learning procedure is to estimate the unknown parameters $\mathbf{x}_c$, b_0^c, b_r^c, $\mathbf{d}_r^c$, λ so that they fit the training data $\mathbf{D}$ well (see Angelova et al., 2006a for details). The receptive fields are distributed to cover the input space so that all training data belong to at least one receptive field. This is done by adding new receptive fields whenever an incoming training example is not covered by other receptive fields. The linear fit in each receptive field is done using the Partial Least Squares algorithm (Wold, 1966), but other linear regression algorithms could be used (Schaal and Atkeson, 1998). When a new point arrives, only the parameters of the receptive fields in the vicinity of this point are to be re-evaluated. This allows for fast update in online learning. Note that the assumed model for the function is fairly nonrestrictive and varieties of nonlinear behaviors could be learned from the training data.

To test how well this works when the terrain type is known, we acquired data with a testbed robotic vehicle in five different terrain types in a park next to JPL (soil, sand, gravel, asphalt, and woodchips). This represents a total of about 5000 frames covering a total distance of about 1 km, of which approximately 3000 frames were used for training and 2000 for testing. Figure 18 shows results for one, representative terrain type (soil); results for the other types are discussed in Angelova et al. (2006a). The vehicle was driven several times up and down the dirt hill shown in the inset photo; roughly half of this data was used for training, the rest for testing. Comparing plots of pitch (lower graphs) to slip (upper graphs), there is a reasonable correlation between the two. RMS prediction errors were 11.8% for the test data in this case. Results

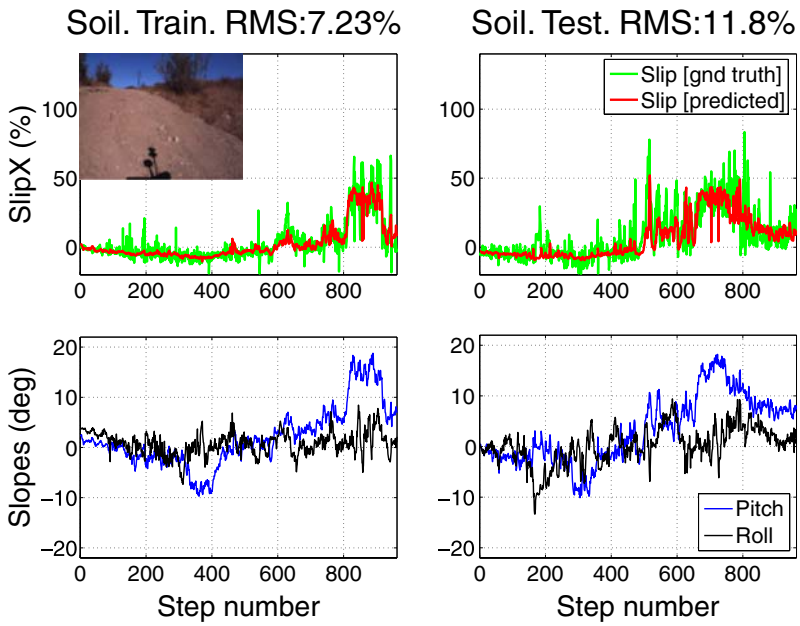


Figure 18. Learning to predict forward slip as a function of terrain slope: results for a single, known terrain type (soil). Top row shows predicted (red) and ground truth (green) slip as a function of step number; training data is on the left, test data on the right. The bottom graphs show the corresponding pitch and roll angles for each step number. Slip does correlate with pitch. Average RMS prediction error is given at the top of each plot.

for other terrain types include RMS test errors of 3.6% on asphalt and 27.5% on gravel. Given the stochastic nature of slip and the presence of noise in various measurements that go into estimating slip, we consider these levels of error to be encouraging. Note that Spirit has experienced slippage has high as 125%; that is, it has actually slipped backwards in the process of trying to climb.

In subsequent work with labeled training data for multiple terrain types, we have used texton-based texture analysis (Varma and Zisserman, 2005) for terrain classification, then used slope as above for slip prediction once the terrain class had been identified. Details and experimental results are reported in Angelova et al. (2006b). The results to date are encouraging. As this still represents early work on a new problem area, a number of issues remain to be addressed, such as determining how well the assumption of a discrete set of classes works in Mars terrain, automatically learning the classes, and reducing this to a real-time, onboard system. Nevertheless, we believe this foray into visual learning of terrain traversability properties is bearing fruit for Mars rovers and will have impact on terrestrial applications of ground robots as well.

5. Discussion and Future Directions

The MER mission has succeeded far beyond its primary mission objectives. The three computer vision algorithms in the mission—stereo vision and visual odometry for the rover and feature tracking for horizontal velocity estimation for the lander—have performed reliably and made material contributions to mission success. Limitations of the 20 MHz onboard processor kept these algorithms very simple, but since this is the first time algorithms like this have ever been used in a mission, this represents an important starting point and opens the door to more advanced algorithms in future missions as onboard computing ability improves.

The 100 MHz PowerPC 750 onboard computer planned for a 2009 Mars rover now in design will improve the computing situation modestly. A much bigger boost may come from the use of FPGA-based computing in subsequent missions. JPL has prototyped a new computing system based on the Xylinx Virtex II Pro FPGA, with two embedded PowerPC 405 (PPC405) processor hard cores. Rover navigation software running on the PPC405s already shows dramatic speed improvements over the MER computer. The stereo vision correlator has been moved into the FPGA logic, with a throughput of one pixel of disparity per clock cycle at 100 MHz, or the equivalent of about 1 ms for a 320×240 disparity map with 32 disparities. Additional functionality in the FPGA logic will follow in the future.

We have discussed some efforts to improve the SNR of the disparity maps, improve disparity at occluding boundaries, and reduce pixel-locking artifacts. The SNR improvement and occluding boundary work is relatively mature and largely implementation-oriented in nature, though important for practical performance. Our

examination of pixel-locking raises interesting questions about the potential value of refining SAD-based disparity maps with affine iterations, but leaves a number of issues open, particularly whether this can be embodied in a real-time system in the near future.

What's next for rover vision? In the vein of more optimization, multi-resolution stereo algorithms should be able to significantly improve the efficiency of stereo for terrain like Mars; doing so has simply not been a top priority so far. More recent work on visual odometry-like algorithms could improve localization, but VO performance is already good enough that this would offer marginal return. There is still much work to be done to enable onboard algorithms that provide absolute localization relative to landmarks or orbital imagery; however, this is not a strong priority in the funding arena. We briefly described a new research thrust on attempting to learn to predict percent slip from experience, by using visual appearance (texture) to classify terrain and nonlinear regression to relate measured slope and slip in each terrain class.

By far the biggest impact for computer vision in future missions will not be new capabilities for rovers, but rather new capabilities to enable precision landing and landing hazard avoidance. We have made a start on these by demonstrating real-time, onboard structure from motion for autonomous landing hazard avoidance by a robotic helicopter (Johnson, 2005a) and by demonstrating algorithms for recognizing craters as landmarks for precision navigation during orbiting and descent operations (Cheng et al., 2003; Cheng and Miller, 2003). Additional techniques are being actively pursued at JPL and elsewhere.

Note

1. A *sol* is one Mars day, which is about 24 hours and 40 minutes.

References

Ali, K. et al. 2005. Attitude and position estimation on the Mars Exploration Rovers. In *IEEE Conference on Systems, Man, and Cybernetics*.

Amidi, O., Kanade, T., and Miller, J.R. 1998. Vision-Based Autonomous Helicopter Research at Carnegie Mellon Robotics Institute 1991–1997. In *American Helicopter Society International Conference*, Heli, Japan.

Amidi, O., Kanade, T., and Fujita, K. 1999. A visual odometer for autonomous helicopter flight. *Journal of Robotics and Autonomous Systems*, 28:185–193.

Andrade, C., Ben Amar, F., Bidaud, P., and Chatila, R. 1998. Modeling robot-soil interaction for planetary rover motion control. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 576–581.

Angelova, A., Matthies, L., and Perona, P. 2006. Slip prediction using visual information. *Robotics Science and Systems Conference*.

Angelova, A., Matthies, L., Sibley, G., and Perona, P. 2006. Learning to predict slip for ground robots. In *IEEE International Conference on Robotics and Automation*.

Ansar, A., Castano, A., and Matthies, L. 2004. Enhanced real-time stereo using bilateral filtering. In *2nd International Symposium on 3D Data Processing, Visualization, and Transmission*, Thessaloniki, Greece.

Bares, J., Hebert, M., Kanade, T., Krotkov, E., Mitchell, T., Simmons, R., and Whittaker, W.L. 1989. Ambler: An autonomous rover for planetary exploration. *IEEE Computer*, 22(6):18–26.

Bares, J. and Wettergreen, D. 1999. Dante II: Technical description, results and lessons learned. *International Journal of Robotics Research*, 18(7):621–649.

Bekker, M.G. 1964. Mechanics of locomotion and lunar surface vehicle concepts. *Transactions of the Society of Automotive Engineers*, 72:549–569.

Biesiadecki, J. et al. 2005. Mars Exploration Rover surface operations: Driving opportunity at meridiani planum. In *IEEE Conference on Systems, Man, and Cybernetics*.

Biesiasdecki, J. and Maimone, M. 2006. The Mars Exploration Rover surface mobility flight software: Driving ambition. In *IEEE Aerospace Conference*.

Bodt, B.A. and Camden, R.S. 2004. Technology readiness level six and autonomous mobility. In *Proc. SPIE Vol. 5422: Unmanned Ground Vehicle Technology VI*, pp. 302–313.

Bornstein, J.A. and Shoemaker, C.M. 2003. Army ground robotics research program. In *Proc. SPIE Vol. 5083: Unmanned Ground Vehicle Technolgy V*, pp. 303–310.

Burt, P., Wixson, L., and Salgian, G. 1995. Electronically directed 'focal' stereo. In *International Conference on Computer Vision*.

Cheng, Y., Johnson, A., Olson, C., and Matthies, L. 2003. Optical landmark detection for spacecraft navigation. In *13th Annual AAS/AIAA Space Flight Mechanics Meeting*.

Cheng, Y. and Miller, J. 2003. Autonomous landmark based spacecraft navigation system. In *13th Annual AAS/AIAA Space Flight Mechanics Meeting*.

Cheng, Y., Goguen, J., Johnson, A., Leger, C., Matthies, L., San Martin, M., and Willson, R. 2004. The Mars Exploration Rovers Descent Image Motion Estimation System. *IEEE Intelligent Systems Magazine*, May/June, pp. 13–21.

Cheng, Y., Johnson, A., and Matthies, L. 2005. MER-DIMES: A planetary landing application of computer vision. In *IEEE Conf. on Computer Vision and Pattern Recognition*, San Diego, pp. 806–813.

Cheng, Y., Maimone, M., and Matthies, L. 2006. Visual odometry on the Mars Exploration Rovers. *IEEE Robotics and Automation Magazine*, Vol. 13, No. 2, pp. 54–62.

Di, K., Xu, F., Wang, J., Niu, C., Serafy, F., Zhou, R., Li, R., and Matthies, L. 2005. Surface imagery based mapping and rover localization for the 2003 Mars Exploration Rover mission. In *Proc. ASPRS Annual Conference*, MD, Baltimore.

Eustice, R., Singh, H., Leonard, J., Walter, M., and Ballard, R. 2005. Visually navigation the RMS Titanic with SLAM information filters. In *Proc. Robotics Science and Systems Conference*.

Gennery, D.B. 1980. *Modelling the Environment of an Exploring Vehicle by Means of Stereo Vision*, PhD thesis, Computer Science Department, Stanford University.

Goldberg, S., Maimone, M., and Matthies, L. 2002 Stereo vision and rover navigation software for planetary exploration. *IEEE Aerospace Conference*, Big Sky, MO, Vol. 5, pp. 2025–2036.

Guerra, C. and Kanade, T. 1985. A systolic algorithm for stereo matching. In *VLSI: Algorithms and Architectures*, P. Bertolazzi and F. Luccio (Eds.), Elsevier Science Publishers B.V. (North-Holland), pp. 103–112.

Hapke, B. 1986. Bidirectional reflectance spectroscopy. IV. The extinction coefficient and the opposition effect. *Icarus*, 67:264–280.

Hebert, M., Kanade, T., and Kweon, I. 1988. 3-D Vision techniques for autonomous vehicles. Technical report CMU-RI-TR-88-12, Robotics Institute, Carnegie Mellon University.

Hebert, M., Krotkov, E., and Kanade, T. 1989. A perception system for a planetary rover. In *Proc. 28th IEEE Conference on Decision and Control*, pp. 1151–1156.

Hebert, M., Bolles, B., Gothard, B., Matthies, L., and Rosenbloom, M. 1996. Mobility for the unmanned ground vehicle. In *Reconnaissance, Surveillance, and Target Recognition (RSTA) for the Unmanned Ground Vehicle*, O. Firschein (Ed.), PRC Inc.

Helmick, D., Cheng, Y., Clouse, D., and Matthies, L. 2004. Path following using visual odometry for a Mars Rover in high-slip environments. In *IEEE Aerospace Conference*, Big Sky, MO.

Hirschmuller, H., Innocent, P.R., and Garibaldi, J. 2002. Real-time correlation-based stereo vision with reduced border errors. *International Journal of Computer Vision*, 47:229–246.

Iagnemma, K., Kang, S., Shibly, H., and Dubowsky, S. 2004. On-line terrain parameter estimation for wheeled mobile robots with application to planetary rovers. *IEEE Transactions on Robotics*, 20(5):921–927.

Johnson, A., Montgomery, J., and Matthies, L. 2005. Vision guided landing of an autonomous helicopter in hazardous terrain. In *IEEE International Conference on Robotics and Automation*, Barcelona, Spain.

Johnson, A., Willson, R., Goguen, J., Alexander, J., and Meller, D. 2005. Field testing of the Mars Exploration Rovers Descent Image Motion Estimation System. In *IEEE International Conference on Robotics and Automation*, Barcelona, Spain, pp. 4463–4469.

Johnson, A., Willson, R., Cheng, Y., Goguen, J., Leger, C., SanMartin, M., and Matthies, L. Design through operation of an image-based velocity estimation system for Mars landing. Joint issue of the *International Journal of Computer Vision* and the *International Journal of Robotics Research*, Special Issue on Vision and Robotics, to appear.

Kanade, T. and Ikeuchi, K. 1991. Introduction to the special issue on physical modeling in computer vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(7):609–610.

Kanade, T., Yoshida, A., Oda, K., Kano, H., and Tanaka, M. 1996. A stereo machine for video-rate dense depth mapping and its new applications. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 196–202.

Kanade, T., Amidi, O., and Ke, Q. 2004. Real-time and 3D vision for autonomous small and micro air vehicles. In *IEEE Conference on Decision and Control*.

Klinker, G., Shafer, S., and Kanade, T. 1990. A physical approach to color image understanding. *International Journal of Computer Vision*, 4:7–38.

Konolige, K. 1997. Small vision systems: Hardware and implementation. In *Proc. 8th International Symposium on Robotics Research*, Hayama, Japan.

Krotkov, E. and Simmons, R. 1996. Perception, planning, and control for autonomous walking with the ambler planetary rover. *International Journal of Robotics Research*, 15(2):155–180.

Krotkov, E. and Blitch, J. 1999. The DARPA tactical mobile robotics program. *International Journal of Robotics Research*, 18(7):769–776.

Krotkov, E., Fish, S., Jackel, L., McBride, B., Pershbacher, M., and Pippine, J. 2007. The DARPA PerceptOR evaluation experiments. *Autonomous Robots*, to appear.

Leger, C. et al. 2005. Mars Exploration Rover surface operations: driving Spirit at Gusev Crater. In *IEEE Conference on Systems, Man, and Cybernetics*.

Levine, M.D., O'Handley, D.A., and Yagi, G.M. 1973. Computer determination of depth maps. *Computer Graphics and Image Processing*, 2:131–150.

Lewis, R.A. and Johnston, A.R. 1977. A scanning laser rangefinder for a robotic vehicle. In *Proc. 5th IJCAI*, Cambridge, MA, pp. 762–768.

Li, R., Archinal, B., Arvidson, R., Bell, J., Christensen, P., Crumpler, L., Des Marais, D., Di, K., Duxbury, T., Golombek, M., Grant, J., Greeley, R., Gunn, J., Johnson, A., Kirk, R., Maimone, M., Matthies, L, Malin, M., Parker, T., Sims, M., Thompson, S., Squyres, S., and Soderblom, L. 2006. Rover localization and topographic mapping at the landing site of Gusev Crater, Mars. *Journal of Geophysical Research—Planets*, 111(E1).

Lindemann, R. and Voorhees, C. 2005. Mars Exploration Rover mobility assembly design, test, and performance. In *IEEE Conference on Systems, Man, and Cybernetics*.

Maimone, M., Biesadecki, J., Tunstel, E., Cheng, Y., and Leger, C. 2006. Surface navigation and mobility intelligence on the Mars Exploration Rovers. In *Intelligence for Space Robotics*, TSI Press, Albuquerque, NM.

Maki, J. et al. 2003. Mars Exploration Rover engineering cameras. *Journal of Geophysical Research*, 108(E12).

Mallet, A., Lacroix, S., and Gallo, L. 2000. Position estimation in outdoor environments using pixel tracking and stereovision. In *IEEE International Conference on Robotics and Automation*, pp. 3519–2524.

Masrani, D.K. and MacLean, W.J. 2006. A real-time large disparity range stereo-system using FPGAs. In *Asian Conference on Computer Vision*, Hyderabad, India.

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

Matthies, L. 1992. Stereo vision for planetary rovers: Stochastic modeling to near real-time implementation. *International Journal of Computer Vision*, 8(1).

Matthies, L., Gat, E., Harrison, R., Wilcox, B., Volpe, R., and Litwin, T. 1995. Mars microrover navigation: Performance evaluation and enhancement. *Autonomous Robots*, 2(4).

Matthies, L., Kelly, A., Litwin, T., and Tharp, G. 1996. Obstacle detection for unmanned ground vehicles: A progress report. In *Robotics Research: The Seventh International Symposium*, G. Giralt and G. Hirzinger (Eds.), Springer, pp. 475–486.

Matthies, L., Xiong, Y., Hogg, R., Zhu, D., Rankin, A., Kennedy, B., Hebert, M., Maclachlan, R., Won, C., and Frost, T. 2002. A portable, autonomous, urban reconnaissance robot. *Robotics and Autonomous Systems*, 40(2).

Matthies, L. and Rankin, A. 2003. Negative obstacle detection by thermal signature. In *IEEE Conf. on Intelligent Robots and Systems*, Las Vegas, NV, pp. 906–913.

Matthies, L., Bellutta, P., and McHenry, M. 2003. Detecting water hazards for autonomous off-road navigation. In *SPIE Conf. on Unmanned Ground Vehicles*, Orlando, FL.

Mettala, E.G. 1992. The OSD Tactical Unmanned Ground Vehicle program. In *Proc. DARPA Image Understanding Workshop*, MorganKaufmann Publishers, pp. 159–172.

Mettler, B., Tischler, M.B., and Kanade, T. 2001. System identification modeling of a small-scale unmanned helicopter. *Journal of the American Helicopter Society*.

Miller, J.R. and Amidi, O. 1998. 3-D Site mapping with the CMU autonomous helicopter. In *Proc. 5th International Conf. on Intelligent Autonomous Systems (IAS-5)*.

Montgomery, J., Johnson, A., Roumeliotis, S.I., and Matthies, L. The JPL autonomous helicopter testbed: A platform for planetary exploration technology research and development. *Journal of Field Robotics*, to appear.

Moravec, H.P. 1980. *Obstacle Avoidance and Navigation in the Real World by a Seeing Robot Rover*, PhD thesis, Computer Science Department, Stanford University.

Moravec, H.P. 1983. The Stanford Cart and the CMU Rover. *Proceedings of the IEEE*, 71(7):872–884.

Nayar, S., Ikeuchi, K., and Kanade, T. 1991. Surface reflection: Physical and geomerical perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(7):611–634.

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

20.

A Scientific Rationale for Mobility in Planetary Environments. 1999. Committee on Planetary and Lunar Exploration, National Research Council. The National Academies Press.

Technology Development for Army Unmanned Ground Vehicles, 2002. Committee on Army Unmanned Ground Vehicle Technology, Board on Army Science and Technology, Division of Engineering and Physical Sciences, National Research Council, The National Academies Press, Washington, D.C.

O'Handley, D.A. 1973. Scene analysis in support of a Mars Rover. *Computer Graphics and Image Processing*, 2:281–297.

Okutomi, M. and Kanade, T. 1993. A multiple-baseline stereo. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(4):353–363.

Pomerleau, D. and Jochem, T. 1996. Rapidly adapting machine vision for automated vehicle steering. *IEEE Expert: Special Issue on Intelligent Systems and their Applications*, 11(2):19–27.

Schaal, S. and Atkeson, C. 1998. Constructive incremental learning from only local information. *Neural Computation*, 10(8):2047–2084.

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

Shimizu, M. and Okutomi, M. 2001. Precise sub-pixel estimation on area-based matching. In *International Conference on Computer Vision*.

Shoemaker, C.M. and Bornstein, J.A. 2000. Overview and update of the Demo III Experimental Unmanned Vehicle program. In *Proc. SPIE Vol. 4024: Unmanned Ground Vehicle Technology II*, pp. 212–220.

Squyres, S. and Knoll, A.H. 2005. Sedimentary rocks at Meridiani Planum: Origin, diagenesis, and implications for life on Mars. *Earth and Planetary Science Letters,* Special Issue on Sedimentary Geology at Meridiani Planum, Mars, Vol. 240, No. 1, pp. 1–10.

Stein, A., Huertas, A., and Matthies, L. 2006. Attenuating stereo pixel-locking via affine window adaptation. In *IEEE International Conference on Robotics and Automation*.

Szeliski, R., and Scharstein, D. 2002. Symmetric sub-pixel stereo matching. In *European Conference on Computer Vision*, pp. 525–540.

Thompson, A.M. 1977. The navigation system of the JPL robot. In *Proc. 5th IJCAI*, Cambridge, MA, pp. 749–757.

Thorpe, C., Hebert, M., Kanade, T., and Shafer, S. 1991a. Toward autonomous driving: The CMU Navlab. Part I: Perception. *IEEE Expert*, 6(4):31–42.

Thorpe, C., Hebert, M., Kanade, T., and Shafer, S. 1991b. Toward autonomous driving: The CMU Navlab. Part II: System and architecture. *IEEE Expert*, 6(4):44–52.

Thrun, S. 2001. A probabilistic online mapping algorithm for teams of mobile robots. *International Journal of Robotics Research*, 20(5):335–363.

Tomasi, C. and Kanade, T. 1992. Shape and motion from image streams under orthography: A factorization method. *International Journal of Computer Vision*, 9(2):137–154.

Varma, M. and Zisserman, A. 2005. A statistical approach to texture classification from single images. *International Journal of Computer Vision*, 62.

Stereo-on-a-Chip [STOC] Stereo Head User Manual 1.1, 2006. Videre Design.

Vijayakumar, S., D'Souza, A., and Schaal, S. 2005. Incremental online learning in high dimensions. *Neural Computation*, 17(12):2602–2634.

Villalpando, C. 2006. Acceleration of stereo correlation in verilog. In *9th Military and Aerospace Programmable Logic Devices (MAPLD) International Conference,* Washington, D.C.

Wettergreen, D., Thorpe, C., and Whittaker, W.L. 1993. Exploring Mount Erebus by walking robot. *Robotics and Autonomous Systems*.

Wilcox, B.H. and Gennery, D.B. 1987. A Mars rover for the 1990s. *Journal of the British Interplanetary Society*, pp. 483–488.

Wilcox, B. and Nguyen, T. 1998. Sojourner on Mars and lessons learned for future planetary rovers. In *Proc. 28th International Conf. on Environmental Systems (ICES'98)*, Danvers, MA, Society of Automotive Engineers.

Williamson, T. 1998. *A High-Performance Stereo Vision System for Obstacle Detection*, technical report CMU-RI-TR-98-24, Robotics Institute, Carnegie Mellon University.

Willson, R., Johnson, A., and Goguen, D. 2005. MOC2DIMES: A camera simulator for the Mars Exploration Rover Descent Image Motion Estimation System. In *8th International Symposium on Artificial Intelligence, Robotics and Automation in Space (iSAIRAS)*, Munich, Germany.

Willson, R., Maimone, M., Johnson, A., and Scherr, L. 2005. An optical model for image artifacts produced by dust particles on lenses. In *8th International Symposium on Artificial Intelligence, Robotics, and Automation in Space (i-SAIRAS)*.

Wold, H. 1966. Estimation of principal components and related models by iterative least squares. *Multivariate Analysis*, P.R. Krishnaiah (Ed.), Academic Press, New York, pp. 391–420.

Woodfill, J., Gordon, G., and Buck, R. 2004. Tyzx DeepSea high speed stereo vision system. In *IEEE Workshop on Real Time 3-D Sensors and their Use*, Washington, D.C.

Xiong, Y. and Matthies, L. 1997. Error analysis of a real-time stereo system. In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1087–1093.