Safeguarded Teleoperation for Lunar Rovers: From Human Factors to Field Trials

Eric Krotkov Reid Simmons Fabio Cozman Sven Koenig
{epk,reids,fgcozman,skoenig}@cs.cmu.edu
School of Computer Science
Carnegie Mellon University
Pittsburgh, Pennsylvania 15213

Abstract

In this paper we present recent advances in developing and validating the safeguarded teleoperation approach to time-delayed remote driving. This approach shares control of the rover using a command fusion strategy: In benign situations, users remotely drive the rover; in hazardous situations, a safeguarding system running on-board the rover overwrites user commands to ensure vehicle safety. This strategy satisfies users, because it allows them to drive (except in hazardous situations), while maintaining the integrity of the rover and mission.

We present results from experiments on untrained teleoperators with and without safeguarding, which reveal needs to be met by future user interfaces. We describe three technical advances in safeguarding: improving the accuracy of dead reckoning by a factor of 2, speeding up the controller by a factor of 18, and developing an area-based rather than a path-based obstacle avoidance planner in order to circumvent map merging problems. Finally, we discuss a field trial validating the approach in a 10 km traverse, demonstrating the effectiveness of safeguarding, even with malicious drivers.

1 Introduction

In our previous work on Mars rovers [Krotkov and Simmons, 1996], we aimed for a high degree of autonomy, because teleoperation with long (up to 45-minute) time delays severely limits rover capabilities. In our current work on lunar rovers, we aim for a mix of teleoperation and autonomous safeguarding, because the Moon is close enough that teleoperation is possible, yet far enough away that time delays prove troublesome.

Our work belongs to a larger Lunar Rover Initiative, which aims to conduct a lunar mission sponsored not by government agencies, but by private ventures dedicated to public participation in space exploration [Krotkov *et al.*, 1994]. A central goal of the initiative is to enable untrained people to experience, and at times directly control, a rover visiting historic sites and unexplored areas on the Moon.

This goal is ambitious, because remotely driving a lunar rover presents a number of technical challenges, with time-delay foremost among them. The largest distance between Earth and the Moon is 407,000 kilometers, leading to a communication delay of about 2.5 seconds in each direction (1.4 seconds between Earth and the Moon, and an estimated 1.1 second communication delay on Earth). Thus, if the rover is about to hit an obstacle on the Moon, the teleoperator on Earth notices this situation 2.5 seconds later, and (assuming instant response) their evasive motion commands reach the rover another 2.5 seconds after that – five seconds after hitting the obstacle, potentially crippling the rover and ending the mission.

A classical approach to time-delayed teleoperation requires the user to specify *waypoints* which the ground station uplinks to the rover, which drives itself to them. Although effective, this approach does not meet the needs of untrained users—whether they are customers at entertainment parks, visitors at museums, students at school, or family members in living rooms—who require intuitive interfaces that permit them to keep their hands on the "steering wheel" as much as possible. These users need a degree of control greater than clicking waypoints on a screen.

Responding to these needs, we have developed the *safeguarded teleoperation* approach, which *shares* control of the rover using a command fusion strategy. In benign situations, users remotely drive the rover. In hazardous situations, a safeguarding system running on-board the rover overwrites user commands to ensure vehicle safety. This strategy satisfies users, because it allows them to drive (except in hazardous situations), while maintaining the integrity of the rover and mission.

While operating in safeguarding mode, the rover must exhibit many of the characteristics of autonomy: perception and understanding of its surroundings, monitoring its own status and health, knowing where it is and where it is going, and taking appropriate actions to maintain its own safety. We observe that the safeguarding system must be on-board in order to react to hazards without communication delay.

In previous work, we implemented a complete safeguarded teleoperation system and validated it by traversing one kilometer of lunar-like terrain [Krotkov *et al.*, 1995; Simmons *et al.*, 1994; 1995a]. The mobile base for the traverse was Ratler (Figure 1), developed by Sandia National Laboratory. This paper addresses a simplified version (Figure 2) of the complete system. The key simplification is to omit goal achievement modules, which in the complete system were implemented as a route planner and a position and heading goal-seeking module. The simplification permits us to concentrate directly on the interaction between stereo-based safeguarding and user control.

This work makes two key contributions. First, it has begun studying the effects of safeguarding on untrained operators in remote driving tasks. The methodology and results of such studies may have profound consequences for all forms of time-delayed teleoperation. Second, it has achieved an unprecedented ten-kilometer traverse under safeguarded teleoperation control, which was enabled by advances in dead reckoning, task synchronization, and stereo-based obstacle avoidance planning.

The paper is organized as follows. In the two next sections, we survey related literature and present a study of human performance in time-delayed remote driving tasks, both with and without safeguarding. In the two following sections, we describe key technical advances over our previous efforts, including improvements in dead reckoning and real-time control, and an innovative area-based approach to obstacle avoidance planning. In the final two sections, we present the results of

Figure 1: Ratler mobile robot

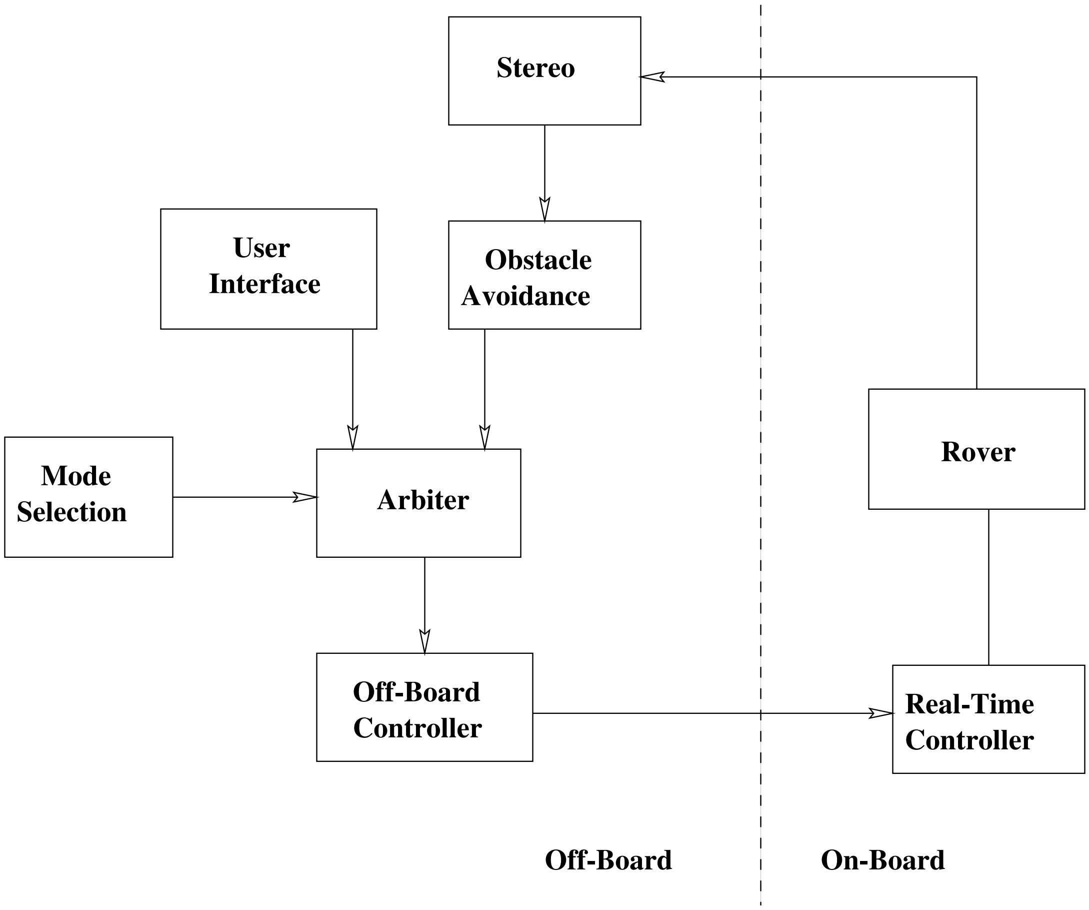

Figure 2: Safeguarded teleoperation system architecture

field trials, and discuss key issues and future work.

2 Related Work

A number of researchers have addressed the problem of stereo-based remote driving under planetary constraints. Some achieve near real-time performance by virtue of special-purpose hardware, such as Datacube systems [Matthies, 1992], digital signal processors [Proy *et al.*, 1993], and transputer arrays [Balazs *et al.*, 1993; Parkes *et al.*, 1993]. Unlike these approaches, we have restricted our attention to general-purpose processors (in this case, a Sparc 10), with computing power comparable to flight-qualifiable computers such as the R4400 and R4600.

Other researchers have demonstrated stereo-based remote driving using modest computing resources [Pagnot and Grandjean, 1995; Lamboley *et al.*, 1995]. [Boissier and Marechal, 1995]. The distinction of our approach is that it has been validated by more thorough testing.

Figure 3: Time-delay

strategy of wizard	runs		score of strategy			compared to strategy of wizard — change in score								
						no intervention (N)			wizard vetos (V)			wizard alters (A)		
			Fat	Fru	Enj	Fat	Fru	Enj	Fat	Fru	Enj	Fat	Fru	Enj
N	5	mean	3.0	2.6	6.4				-0.3	+0.0	-1.0	-0.5	+1.0	+0.0
		variance	2.0	1.0	1.8				0.9	0.7	0.7	0.3	0.0	0.0
V	9	mean	3.3	2.9	6.1	+0.3	+0.0	+1.0				-0.3	+0.5	-0.8
		variance	5.1	1.8	4.1	0.9	0.7	0.7				3.0	1.8	0.8
A	8	mean	2.8	3.4	6.0	+0.5	-1.0	+0.0	+0.3	-0.5	+0.8			
		variance	1.9	4.0	3.6	0.3	0.0	0.0	3.0	1.8	0.8			

Table 1: Teleoperation trial results

We tested our command fusion strategy on eleven subjects, mostly undergraduate and graduate students. Their initial training was minimal: they were given only two two-minute test drives on a small training course, one test drive without time-delay and one with a five-second time-delay. The subjects were then instructed to drive as quickly as possible through an obstacle course to a brightly marked finish line that was visible from the start position of the rover. The obstacle course (Figure 4, right) was 54 meters long and contained 32 obstacles, mostly old tires. The obstacle spacing varied from 3 to 4.5 meters.

We compared trials with no wizard (N) and with a wizard that vetos shortly before the rover would hit an obstacle (V). This control strategy has the disadvantage that, when the wizard vetos a user command, the user can react to it only with a time-delay of five seconds. Thus, after a veto command, the rover usually sits idle for this period of time, which is a potential cause of frustration. We therefore also tested a control strategy in which the wizard alters the user commands slightly to avoid collisions with obstacles and vetos only when absolutely necessary (A). Each subject, knowing the current control strategy, performed two trials with two different control strategies of the wizard.

Even if the wizard did not intervene, all subjects but one were able to complete the obstacle course successfully and hit only approximately one obstacle per trial. However, nine of the eleven subjects reported that the hardest aspect of their driving experience was the time-delay. After each trial, they rated their fatigue (Fat), frustration (Fru), and overall enjoyment (Enj) on a scale from 1 (lowest) to 10 (highest).

Table 1 tabulates the results. Given the small sample size, we expected the data to be informative, but not definitive.

We expected to observe a tendency for the presence of the wizard to decrease the fatigue and frustration of the subjects and, as a result, increase their overall enjoyment and performance. Unexpectedly, this did not happen. For example, there were nine subjects who had one trial where the wizard vetoed (V). They rated their fatigue as 3.3 on average. The subjects during whose other trial the wizard did not intervene at all (N) rated their fatigue in this other trial 0.3 points higher, for an average score of 3.6.

Post-driving interviews revealed two reasons for this unexpected finding. First, many subjects did not take advantage of the presence of the wizard. Eight subjects did not adapt their driving strategy to the strategy of the wizard. They wanted to be in control of their driving and did not rely at all on the presence of the wizard, for example by increasing the speed of the rover. (There was

Figure 6: Success

rover, maintaining heading towards the goal line was a problem for only one subject. However, if an obstacle left the field of view, then its presence was no longer remembered by the subjects. This proved to be a problem, because there was a blind spot of 64 cm directly in front of the rover in which obstacles could not be seen. Once an obstacle entered this blind spot, the subjects forgot about it. They were then surprised when the wizard vetoed their forward commands soon afterwards. Three subjects therefore explicitly asked us to either eliminate the blind spot or provide a global map that shows the rover and the obstacles that surround it.

Most subjects maintained the maximum speed of the rover in the middle of the offered velocity range, even if the largest velocity allowed was drastically reduced from 42 cm/sec to 21 cm/sec. We attribute this behavior to the limited training time; half of the maximum speed appears to be a choice that the user, intuitively, accepts as safe and not too bold, given that they do not have the experience yet to make their own judgements.

Three subjects, two of whom were completely unfamiliar with graphical user interfaces, initially had problems understanding the function of the input device. They believed that pointing into the eastern region of the "clock face" would automatically turn the rover until it faced east. These subjects would have preferred either an actual joystick or a steering wheel. Two subjects did not like that the mouse button had to be pressed continuously to make the rover move ("gas pedal control"). They would have preferred an active stop command ("brake pedal control"). Two subjects mentioned that it was hard to watch two monitors, the one that provided the video from the rover and the one that displayed the input device.

Our conclusion from these observations is that more sophisticated approaches to safeguarded teleoperation are needed. We have begun work on a new user interface with the following properties:

First, the user is given more control over the safeguarding software. Although the safeguarding software always stops the rover before it hits an obstacle, the user is able to decide when the system is allowed to alter their commands and when it is allowed to take over control completely, for example to get the rover out of a tricky situation. We expect the user to rely more on the safeguarding mechanisms when he feels more in control of them.

Second, the user uses a steering wheel to control the rover, together with gas and brake pedals – input devices that most users can be expected to be familiar with. The speed control is non-linear, with smaller speeds given a larger range of the input.

Third, to reduce the cognitive load, we provide users with more information. We eliminate the blind spot in front of the rover and provide the user with a predictive display that, based on the commands of the user, estimates the future rover position and thus predicts for them the future movement of the rover. The advantage is that the user need not predict the situation in which their current command will apply; this situation is now predicted and visualized by the user interface. We are also considering installing a second camera on the rover that points backwards and whether to let the user interface display a local map of known obstacles around the rover, using an elevation map created by the on-board stereo perception system and then transmitted to the user control station.

Although the new user interface is still under development, we [Mathan *et al.*, 1996] have simulated the predictive display (Figure 7), which consists of two lines, projected onto the video display, that trace the predicted path of the rover in the next five seconds. It also shows an arrow

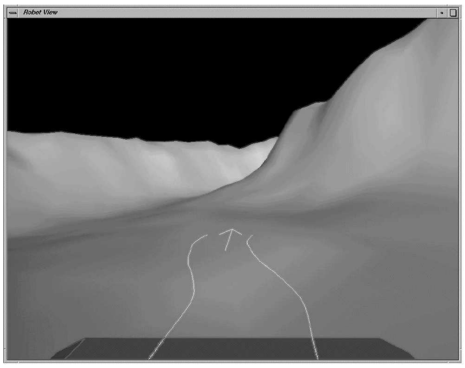

Figure 7: Predictive display

that describes the position and heading of the rover in five seconds. Initial experiments with this predictive display have been encouraging: users produced more continuous command streams, drove faster, oversteered less, and hit fewer obstacles when no wizard was present.

4 Real-Time Controller

In this section we discuss two efforts to enhance the real-time controller. The first improves the accuracy of dead reckoning by a factor of 2. The second implements feedback control that increases response time by a factor of 18. These enhancements contributed substantially to the success of the field trials.

4.1 Dead Reckoning

Eight different sensor measurements are available inside Ratler: four encoders (one for each wheel), three inclinometers (roll, left pitch, and right pitch), and a compass. We have presented one approach to using these measurements for dead reckoning [Krotkov *et al.*, 1995]. We have also presented an alternative approach using an inertial measurement unit [Fuke and Krotkov, 1996]. In this section, we present an evolution of the first approach, and the results of a rigorous test program.

The signals from the three inclinometers are quite stable and noiseless; on the other hand, the compass signal presents high levels of noise during motion. Further tests with the compass and similar devices from other robots available in our group pointed to the same general conclusion: the existence of large metallic structures causes sudden and unexpected changes in orientation measurements. Since we conducted our tests in the vicinity of large buildings, we were forced to abandon the compass altogether, and rely solely on encoder and inclinometer integration.

Take (x, y, z) for position, θ for yaw (orientation) and ϕ for average pitch from the filtered inclinometer signals. We computed the coordinates with respect to a global coordinate system:

$$x_{n+1} = x_n + \frac{e_r + e_l}{2} \cos\theta_n \cos\phi$$

$$
\begin{aligned}
y_{n+1} &= y_n + \frac{e_r + e_l}{2} \sin\theta_n \sin\phi \\
z_{n+1} &= z_n + \frac{e_r + e_l}{2} \sin\phi \\
\theta_{n+1} &= \frac{e_r - e_l}{B},
\end{aligned}
$$

where B is the wheelbase and e_r and e_l are respectively the encoder displacements expressed in metric units.

To quantify the performance of this approach, we built a test site in a 10×20 meter area with rough but uniform terrain, in which we drew a grid with intervals of 0.25 meter. We attached a piece of chalk to the center of Ratler, so that motion of the robot produced a trail of chalk on the ground.

We commanded Ratler to follow 10 different arcs, varying from a sharp right turn (radius of 3 meters) to a sharp left turn (same radius). We repeated each arc 3 times. We repeated straight motions approximately 15 times. Misalignment of one of the wheels to the right by a mere 2° caused the robot to drift 0.6 meter every 20 meters (3% error). Skid steering contributed an additional 0.75 meter drift (4% error), on average, for commanded linear trajectories. We compensated these errors by introducing correction factors in the encoder readings. This substantially reduced the mean error. For straight lines, average error fell to 0.3 meter for every 20 meter trajectory (1.5% error). However, for abrupt turns performance is poor; for instance, a 3 meter radius turn leads to 10% mean error. This poor performance during turns appears to be an inevitable consequence of the slippage caused by skid steering.

4.2 Task Synchronization

In an attempt to achieve modularization, the original design for the low-level control system divides dead reckoning and control into five tasks, all executing concurrently and sharing memory. The task responsible for reading the encoders executes every 10 ms and the compass-reading task executes every 100 ms. The task responsible for wheel speed control executes every 100 ms, and occurs asynchronously with respect to all other tasks. The original design does not include feedback control for the wheels. During tests reported in [Krotkov *et al.*, 1995], it became obvious that closed-loop control was necessary.

The modular architecture, in which different tasks behave independently, creates artificial sources of noise. In some cycles, the 100 ms-time-based low-level control task obtains the result of ten readings from the 10 ms-time-based encoder task; in other cycles, it obtains nine or eleven readings, due to minor delays caused by task swapping. As a result, the integration of encoder signals received values corrupted by $\pm 10\%$. Figure 8 illustrates this phenomenon by graphing measured speed against time in a long run (225 seconds). Speed gradually increased and then gradually decreased. Notice that noise is *not* additive; instead, it depends on speed, because the lack of synchronization induces speed-relative errors rather than absolute errors. With such noisy signals, feedback control tends to lead the system into unstable behavior, causing the rover to shake, fight itself, and behave in schizophrenic bursts. The best possible gains are stable only for time

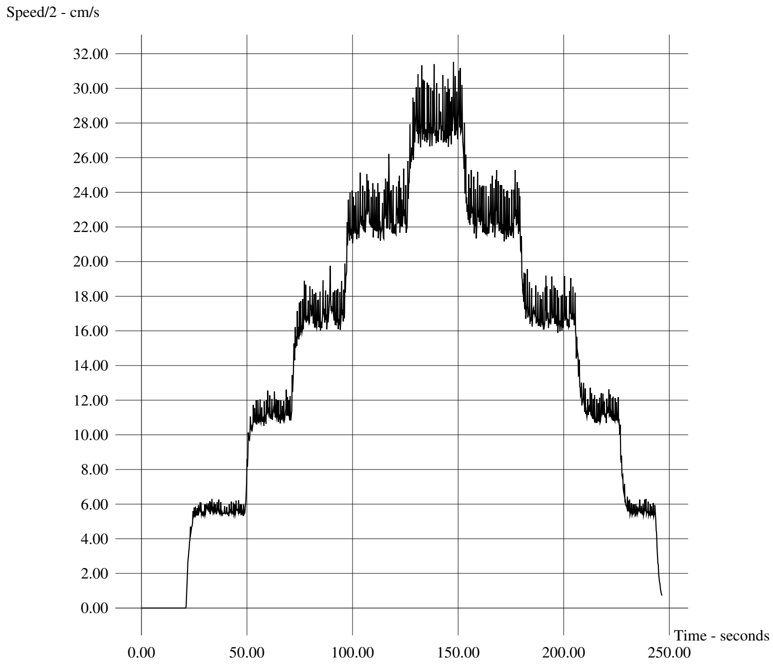

Figure 8: Relationship between noise and speed in modular design

constants of 12 seconds, meaning that the robot would take approximately 30 seconds to obey an arc command.

We redesigned the system to synchronize the critical tasks of data acquisition, dead reckoning, and control. This removes the independence among functional units, but improves performance dramatically, suppressing noise significantly, and reducing time constants to 0.67 second (18 times faster response). We now describe the controller tuning.

Tests with the four wheel motors indicated that their parameters were virtually identical; the electrical constants were practically negligible. The motors were analyzed with various load conditions: Ratler suspended, Ratler on the ground, Ratler loaded with stones. In all cases, there was little variation in the mechanical constant of the motors. This was due to the reduction gears (100:1) coupled to the motors, which consume a substantial portion of the power produced by the motors, irrespective of load. Thus, we used a first-order model for the motors:

$$\dot{v} = \frac{K}{T}v - \frac{1}{T}u \ ,$$

where u is the input, v is the measured speed, K is the motor's steady state gain, and T is the motor's mechanical time constant. The time constant of all motors, with full load was determined to be $T = 0.15$ second. The steady-state gain was more complicated; the motors had a dead zone from 0.0 to 0.03m/s, and then exhibited highly linear behavior until speeds of 0.91m/s (Figure 9 shows the relationship between commanded and enforced steady speed). The gain $K = 1.248$ reasonably approximates this behavior for all commanded speeds. We implemented a proportional controller,

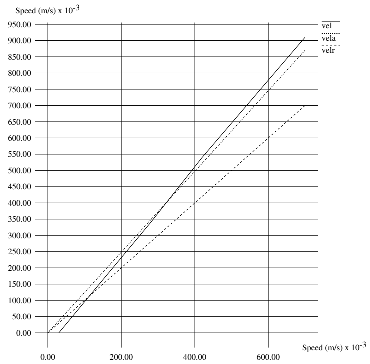

Figure 9: Steady state gain tests: the line *vel* is the empirically tested relationship; *vela* is the approximation obtained with $K = 1.248$; *velr* is the (theoretically correct) gain of 1.

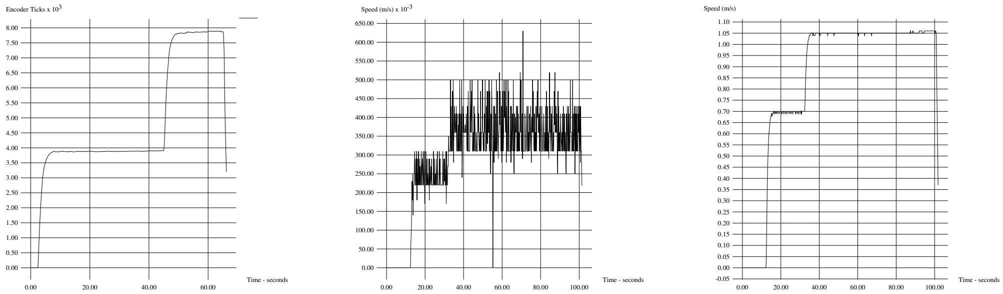

Figure 10: Results: encoder signals for synchronized system (left); speed signal for modular system (middle); speed signal for synchronized system (right)

with constant gain set to unity. This gain fixed the dominant pole of the motors at (approximately) -1.5, guaranteeing fast response (0.67 second time constant).

The three graphs in Figure 10 illustrate the results. The first graph shows the encoder signals in a typical run; compare this with Figure 8. The second and third graphs compare the average speed signal computed by the dead reckoning task. In the second graph, speed was computed by taking the noisy encoder signals and dividing them by a noisy time-scale. The third graph shows the average speed with the tasks synchronized in a unique module.

5 Obstacle Avoidance Planning

A key to our approach is that the rover should, in real-time, evaluate where it is safe for it to be heading, based on local sensing (e.g., stereo vision). These evaluations can then be used either to drive the rover autonomously, or to safeguard the rover by over-riding (or even by modifying) user commands (Section 3).

In our previously reported experiments with Ratler [Krotkov *et al.*, 1995; Simmons *et al.*, 1995b; 1995a], we adapted a local obstacle avoidance planner called Ranger, which was developed at CMU for ARPA's Unmanned Ground Vehicle (UGV) program for cross-country navigation [Kelly, 1995]. Ranger works by analyzing the paths the vehicle would traverse along the terrain for a number of different steering angles, and choosing the one that evaluates as the safest.

Terrain data for Ranger is obtained from stereo vision. To make the stereo computation tractable, only a small segment of the stereo image (about 2%) is requested, at reduced resolution (every fifth row and column). Experiments show that this is sufficient for reliably detecting features on the order of 20 cm high. The local obstacle avoidance algorithm dynamically chooses which portion of the image should be processed, based on the current vehicle speed, stopping distance, and expected cycle time of the perception/planning/control loop. The Ranger algorithm merges individual stereo-produced elevation maps to create a 25 cm resolution grid map up to seven meters in front of the rover. Map merging is necessary because the limited fields of view of the cameras do not allow a single image to view sufficient terrain.

Ranger works by projecting the rover's state (position, roll, pitch, yaw) as it travels along a path. The projection is based on a desired steering angle, the vehicle dynamics, and the underlying terrain. The vehicle's current pose, its dynamics and the steering angle are used to determine the position and yaw of the vehicle at the next time step. The height of the terrain under the wheels is then used to determine the roll and pitch of the vehicle at that point.

Once a projection of the vehicle along a path has been computed, the vehicle state at each point in time is evaluated. Four criteria are used to determine the "goodness" of a path: roll, pitches of the left and right body segments of Ratler, and number of known terrain points the vehicle crosses along the path. If any of the criteria exceed a given threshold of safety (such as excessive roll or pitch), the whole path is given a very low evaluation. Otherwise, the criteria are normalized to the range [0..1] and are combined using a linear weighted function. This determines the overall merit of choosing that steering angle for the rover. These evaluations are then combined with the user's preferences (Section 3) to determine the overall best command, which is then sent to the rover to be executed. The cycle time for this process is about 2 seconds, with the stereo computations taking up about 75% of the time.

While the Ranger algorithm has worked well for high-speed navigation of Humvees, it is not entirely well-suited to the much smaller, and slower, lunar rover.

As is often the case in robotics, the problems are mainly attributable to an abundance of noise: noise in the the stereo-produced terrain maps and noise in the dead-reckoning (particularly, the roll, pitch and yaw angles). The main effect of the noise was to make it difficult to cleanly merge terrain maps acquired from separate images, which is required by the Ranger algorithm since it uses only a small percentage of each image. Map merging often produced artifacts in the map, such as crevasses and ridges, which the rover would refuse to cross. This is less of an issue with the Humvees, since they can cross much taller obstacles. We tried several merging techniques in an attempt to minimize the artifacts, but none was robust enough to yield consistent driving results.

Another effect of noisy terrain data is that, because of the rover's relatively short wheelbase, small deviations in perceived terrain elevation under the wheels produced relatively large changes in estimated roll and pitch. For example, a 20 cm "spike" in the terrain map (not uncommon) produces a 13 degree change in pitch, given a 90 cm wheelbase. Thus, it was often difficult to distinguish noise from steep bumps. This same problem also made it difficult to reliably determine

effective than trying to combine overlapping evaluations in some way).

Our navigation experiments (Section 6) have demonstrated the superiority of the Morphin algorithm for our rover. Morphin addresses the problem of noisy data by aggregating independent data points into an overall statistic, thus dramatically lessening the impact of any single point. While this aggregation can sometimes cause the rover to behave more conservatively than would otherwise be warranted, in our application it is better to be too conservative than to allow the rover to head into danger.

The remaining weakness of the algorithm is that the stereo vision often cannot see craters, and reports them as unknown areas on the map. To handle such situations intelligently, Morphin needs to have a better understanding of spatial continuity: it needs to distinguish disjoint patches of truly unknown areas from contiguous unknown patches that may represent craters and other drop-offs.

6 Field Trials

To validate the safeguarded teleoperation approach, and gain insight into its strengths and weaknesses, we performed an extensive field trial. The test site (Figure 12) is an open field of area 2,000 square meters, bounded by a river and the Robotics Engineering Consortium building in Pittsburgh. The ground surface consists of soil, crumbled asphalt, loose gravel, scree, and some grassy vegetation. Obstacles to rover passage include soil mounds, depressions, cliffs at the river bank, building walls, metal pipes, cement blocks, railroad ties, trees, and bushes.

In previous outdoor experiments and field trials, the test method was to identify a goal (position and heading), and evaluate performance based on the accuracy with which the rover reached the goal. Unlike previous efforts, this field trial was formulated to test the performance of safeguarding given a careless and occasionally malicious driver. The trial called for the driver to explore the test site, at times covering ground systematically, at other times driving aimlessly, and at still other times driving maliciously, intentionally trying to drive into the river or at an obstacle. The performance criteria is the number of safeguarding failures.

On-board, we employed one 486 and one 386 processor. Off-board, we used a Sparc 10/30 (SuperSparc processor) with 32 MB memory for stereo, and a Sparc IPX (SunW 4/50 processor) with 48 MB memory for obstacle avoidance planning.

We conducted the trial using a video tether, because the microwave video link performed poorly, probably because of the proximity to the river and interference from broadcasting by local businesses. This required tether management, but did not limit the scope of the trial.

Figure 13 summarizes the statistics of the field trial. The rover traveled over 10 km in one period of 3 days, during which the same problems arose again and again. In practice, the rover could have covered greater distances, but given the repetitiveness of the problems, further testing would not be likely to cause new types of problems to appear.

One set of problems was due to obstacle detection. Stereo performed poorly at depressions, producing a significant number of unknown measurements, and a larger than usual number of outlier measurements. These problems accounted for virtually all of the operator interventions. Stereo performed with competence at discrete obstacles such as rocks and soil mounds, and functioned well despite varying lighting conditions

Figure 12: Test site

Traverse (meters), total	10,205
Traverse (meters), first day	1,163
Traverse (meters), second day	4,528
Traverse (meters), third day	4.514
Typical speed (cm/sec)	15–20
Operator interventions per 100 obstacles (est)	5
Operator drive time percentage (estimated)	1
Number of stereo pairs matched (projected)	9,135
Overall cycle time (seconds), mean	4.477
Stereo cycle time (seconds), mean	3.414
Stereo cycle time (seconds), standard deviation	0.440
Obstacle avoidance cycle time (seconds), mean	1.063
Obstacle avoidance cycle time (seconds), st. dev.	0.104
Number of 3D pixels processed per cycle, mean	1,612
Number of 3D pixels processed per cycle, st. dev.	359
Total number of 3D pixels processed (projected)	3,279,465

Figure 13: Field trial statistics

Another set of problems involved system timing and synchronization. An example is spastic behavior after switching between safeguarded and pure teleoperation. Another example is unreasonably slow recovery from situations in which forward progress is blocked (the rover should back up or begin rotating, but does nothing for some time).

A third set of problems was caused by component limitations. Problems included communication dropouts, epoxy debonding, and electronics overheating. These are not fundamental, in the sense that buying higher-quality devices is the solution.

Acting maliciously, we attempted to drive the rover into hazardous situations. In one scenario, we drove as fast as possible at an obstacle, trying to ram it. The safeguarding system naturally resists operation at ramming speed, because the rover rapidly outruns the known data, and Morphin treats unknown data as if it were a known obstacle. In another scenario, we aimed at a discrete obstacle. In these cases, Morphin prevented contact by planning and executing an alternative trajectory around the obstacle. In a third scenario, we aimed at an extended obstacle. In these cases, Morphin would begin evasive maneuvers as in the case of the discrete obstacle. However, no amount of looking left or right would reveal a traversable path. So Morphin would halt the vehicle. Exception monitors then kicked in, causing the rover to back up and/or turn aggressively until it could find a new path. In summary, the safeguarding system performed competently at detecting and avoiding hazardous situations, even when drivers acted maliciously.

We observed interesting behavior when traveling in areas with sharp, well-defined shadows. Under such conditions, the rover appears to pick either the shadowed or unshadowed region, and to avoid the shadow boundary. The reason for this unintended behavior is that the auto-iris lens adapts to either the brighter or the darker scene, whichever happens to occupy a larger portion of the image. After adaptation, the image portions with poor contrast begin to correlate less and less, causing the number of matches to fall, in turn causing the number of unknown areas to climb, causing the rover to steer away from that area.

We observed another interesting behavior when driving down a dirt road. Vegetation on either side of the road caused those areas to appear as untraversable, producing the unintended behavior of remarkably robust road-following.

7 Discussion

In this paper, we have presented recent advances in developing and validating the safeguarded teleoperation approach to time-delayed remote driving. We presented results from experiments on untrained teleoperators with and without safeguarding. The results were inconclusive, but indicated that developing and evaluating the user interface raised a number of inter-related issues. We described three technical achievements in the safeguarding system: improving the accuracy of dead reckoning by a factor of 2, speeding the controller up by a factor of 18, and developing an area-based rather than a path-based obstacle avoidance planner in order to circumvent map merging problems. Finally, we discussed the 10 km field trial, which demonstrated the effectiveness of safeguarding, even with malicious drivers.

Future work will continue to push safeguarded teleoperation to new performance peaks. We will add a proximity sensor to detect the depressions and dropoffs that stereo does not see well enough. We anticipate that this will enable the system to traverse a wide variety of terrains without any form of intervention. We will mount all computing resources on-board, moving from Sparcs running SunOS to Pentiums running Linux. We expect that this will reduce cycle time by a factor of two or more. We will upgrade communication componentry, including a radio-frequency link for commands and data and a video link for camera imagery. Finally, we will achieve still higher levels of reliability and performance, continuing to refine the safeguarded teleoperation approach until time-delayed remote driving becomes routine, and we elevate our efforts to the next frontier: the Moon.

Acknowledgments

This research was partially sponsored by NASA under grants NAGW-1175 and NAGW-3863. We gratefully acknowledge significant contributions to this work by Ben R. Brown, Lonnie Chrisman, Diana Gant, Richard Goodwin, Martial Hebert, Lalitesh Katragadda, Paul Klarer, Patrick Muir, Yoshiakazu Shinoda, Gregory Whelan, and Red Whittaker.

References

[Balazs *et al.*, 1993] A. Balazs, J. Biro, and S. Szalai. Transputer-based onboard computer. In *Proc. Workshop on Computer Vision for Space Applications*, pages 396–411, Antibes, France, September 1993.

[Boissier and Marechal, 1995] L. Boissier and L Marechal. Rover demonstrator for Moon exploration. *Autonomous Robots*, 2(4):353–362, December 1995.

[Brooks *et al.*, 1990] R. Brooks, P. Maes, M. Mataric, and G. More. Lunar base construction robots. In *Proc. IEEE Intl. Workshop on Intelligent Robots and Systems*, pages 389–392, Tsuchiura, Japan, July 1990.

[Chatila *et al.*, 1995] R. Chatila, S. Lacroix, T. Simeon, and M. Herrb. Planetary exploration by a mobile robot: Mission teleprogramming and autonomous navigation. *Autonomous Robots*, 2(4):333–344, December 1995.

[Fuke and Krotkov, 1996] Y. Fuke and E. Krotkov. Dead reckoning for a lunar rover on uneven terrain. In *Proc. IEEE Intl. Conf. Robotics and Automation*, Minneapolis, To appear, April 1996.

[Kay and Thorpe, 1993] J. Kay and C. Thorpe. STRIPE: Supervised telerobotics using incremental polygonal-earth geometry. In *Proc. Intl. Conf. Intelligent Autonomous Systems*, Pittsburgh, February 1993.

[Kelly, 1995] Alonzo Kelly. An intelligent predictive control approach to the high speed cross country autonomous navigation problem. Technical Report CMU-CS-TR-95-33, Carnegie Mellon University, 1995.

[Krotkov and Simmons, 1996] E. Krotkov and R. Simmons. Perception, planning, and control for autonomous walking with the Ambler planetary rover. *Intl. J. Robotics Research*, 15(2):?, To appear, April 1996.

[Krotkov *et al.*, 1994] E. Krotkov, J. Bares, L. Katragadda, R. Simmons, and R. Whittaker. Lunar rover technology demonstrations with Dante and Ratler. In *Proc. Intl. Symp. Artificial Intelligence, Robotics, and Automation for Space*, Jet Propulsion Laboratory, Pasadena, California, October 1994.

[Krotkov *et al.*, 1995] E. Krotkov, M. Hebert, and R. Simmons. Stereo perception and dead reckoning for a prototype lunar rover. *Autonomous Robots*, 2(4):313–331, December 1995.

[Lamboley *et al.*, 1995] M. Lamboley, C. Proy, L. Rastel, T. Trong, A. Zashchirinski, and S. Buslaiev. Marsokhod: Autonomous navigation tests on a Mars-like terrain. *Autonomous Robots*, 2(4):345–352, December 1995.

[Lescoe *et al.*, 1991] P. Lescoe, D. Lavery, and R. Bedard. Navigation of military and space unmanned ground vehicles in unstructured terrains. In *Proc. Conf. on Military Robotic Applications*, September 1991.

[Mathan *et al.*, 1996] S. Mathan, A. Hyndman, K. Fischer, J. Blatz, and D. Brams. Efficacy of a predictive display, steering device, and vehicle body representation in the operation of a lunar vehicle. In *Human Factors in Computing Systems: CHI 96 Conference Companion*, page ?, Vancouver, To appear, April 1996.

[Matthies *et al.*, 1995] L. Matthies, E. Gat, R. Harrison, B. Wilcox, R. Volpe, and T. Litwin. Mars microrover navigation: Performance evaluation and enhancement. *Autonomous Robots*, 2(4):291–312, December 1995.

[Matthies, 1992] L. Matthies. Stereo vision for planetary rovers: Stochastic modeling to near real-time implementation. *Intl. J. Computer VIsion*, 8(1):71–91, July 1992.

[Newman, 1962] R. Newman. Time lag considerations in operator control of lunar vehicles from Earth. In C. Cummings and H. Lawrence, editors, *Technology of Lunar Exploration*. Academic Press, 1962.

[Pagnot and Grandjean, 1995] R. Pagnot and P. Grandjean. Fast cross-country navigation on fair terrains. In *Proc. IEEE Intl. Conf. Robotics and Automation*, pages 2593–2598, Nagoya, Japan, May 1995.

[Parkes *et al.*, 1993] S. Parkes, A. Gillions, and S. Price. A processing architecture for space-based computer vision systems. In *Proc. Workshop on Computer Vision for Space Applications*, pages 322–334, Antibes, France, September 1993.

[Proy *et al.*, 1993] C. Proy, B. Hotz, O. Faugeras, P. Gernesson, and M. Berhod. Onboard vision system for a mobile planetary exploration robot. In *Proc. Workshop on Computer Vision for Space Applications*, pages 2–12, Antibes, France, September 1993.

[Shirley and Matijevic, 1995] D. Shirley and J. Matijevic. Mars Pathfinder microrover. *Autonomous Robots*, 2(4):283–290, December 1995.

[Simmons *et al.*, 1994] R. Simmons, E. Krotkov, L. Katragadda, and M. Hebert. Experience with rover navigation for lunar-like terrains. In *Proc. Intl. Lunar Exploration Conf.*, San Diego, California, November 1994.

[Simmons *et al.*, 1995a] R. Simmons, E. Krotkov, L. Chrisman, F. Cozman, R. Goodwin, M. Hebert, L. Katragadda, S. Koenig, G. Krishnaswamy, Y. Shinoda, W. Whittaker, and P. Klarer. Experience with rover navigation for lunar-like terrains. In *Proc. Intl. Conf. Intelligent Robotos and Systems (IROS)*, pages 441–446, Pittsburgh, Pennsylvania, August 1995.

[Simmons *et al.*, 1995b] R. Simmons, E. Krotkov, L. Chrisman, F. Cozman, R. Goodwin, M. Hebert, G. Heredia, S. Koenig, P. Muir, Y. Shinoda, and W. Whittaker. Mixed-mode control of navigation for a lunar rover. In *Proc. SSI/Princeton Space Manufacturing Conference*, Princeton, New Jersey, May 1995.

[Stentz, 1989] A. Stentz. *The NAVLAB System for Mobile Robot Navigation*. PhD thesis, Carnegie Mellon University, 1989.