IEEE International Workshop on Robot-Human Interactive Communication, September 2001, Bordeaux and Paris, France

Multi-robot remote driving with collaborative control

Terrence Fong[1,2], Sébastien Grange[2], Charles Thorpe[1] and Charles Baur[2]

[1]The Robotics Institute
Carnegie Mellon University
Pittsburgh, Pennsylvania 15213 USA

[2]Institut de Systèmes Robotiques
Ecole Polytechnique Fédérale de Lausanne
CH-1015 Lausanne EPFL, Switzerland

Abstract

Multi-robot remote driving has traditionally been a difficult problem. Whenever an operator is often forced to divide his limited resources (attention, decision making, etc.) among multiple robots, control becomes complicated and performance quickly deteriorates as a result. To remedy this, we need to find ways to make command generation and coordination efficient, so that human-robot interaction is transparent and tasks are easy to perform. In this paper, we discuss the use of collaboration, human-robot dialogue and waypoint-based driving for vehicle teleoperation. We then describe how these techniques can enable a single operator to effectively control multiple mobile robots.

1 Introduction

1.1 Multi-robot remote driving

The American military is currently developing mobile robots to support future combat systems. These robots will be used to perform reconnaissance, surveillance and target acquisition. Because this work has traditionally required significant human resources and risk taking, one of the primary areas of interest is determining how a small number of operators can use multiple mobile robots to perform these tasks.

Vehicle teleoperation, however, is not easy to perform. With manual control, performance is limited by the operator's motor skills and his ability to maintain situational awareness. Fatigue, lack of concentration, and poor displays all contribute to reduced performance. Additionally, humans have difficulty building mental models of remote environments. Distance estimation, obstacle detection and attitude judgement can also be difficult [11].

Moreover, task and environmental factors can further complicate the problem. If multiple robots must be controlled, the operator must divide his limited resources among them. If a robot operates in an unfamiliar setting or in the presence of hazards, the operator has to dedicate significant attention to assess the remote environment. If the operator and robot are widely separated, communications may be affected by noise or transmission delay, both of which can make direct control impractical or impossible.

In short, vehicle teleoperation is problematic. At a minimum, poor performance (imprecise control, slow driving, etc.) will occur. In the worst case, vehicle failure (rollover or collision) will result. Thus, to make multi-robot remote driving effective and productive, we need to make it easier for the operator to understand the remote environment, to assess the situation, and to generate commands.

Our approach is to create techniques and tools which improve human-robot interaction in vehicle teleoperation [4]. Thus, we are investigating a new system model for teleoperation and are developing techniques for efficient waypoint-based driving. Additionally, we are building operator interfaces which are highly portable, easy to deploy and easy to use.

1.2 Collaborative control

We believe there are clear benefits to be gained from humans and robots working together. In particular, if we can treat a robot not as tool, but rather as a partner, we will be able to accomplish more meaningful work and to achieve better results[5].

To this end, we have developed *collaborative control*, a system model in which a human and a robot collaborate to perform tasks and to achieve common goals. Instead of a supervisor dictating to a subordinate, the human and the robot engage in *dialogue* to exchange ideas and resolve differences. Hence, the robot is more equal and can treat the human as a limited source of planning and information.

An important consequence is that the robot can decide how to use human advice: to follow it when available and to modify it when unsafe. This is not to say that the robot becomes "master": it still follows high-level strategy set by the human. However, with collaborative control, the robot has more freedom in execution. As a result, teleoperation is better able to accommodate varying levels of autonomy.

Perhaps the most significant benefit of collaborative control is that it preserves the best aspects of supervisory control (use of human perception and cognition) without requiring time-critical or situation-critical response from the human. If the human is available, then he can provide direction or problem solving assistance. But, if the human is unavailable, the system can still function.

2 Related Work

2.1 Human-Robot Collaboration

Humans and robots have been working together since the 1940. At first, human-robot interaction was primarily uni-directional: simple switches or controls for operating manipulator joints and remote vehicles. However, as robots have become more autonomous, this relationship has changed to be more like the relationship between two human beings. As a result, humans and robots now communicate and collaborate in a multitude of ways[14].

Personal service robots, for example, directly assist people in daily living activities. Baltus et al. discuss the development of mobile robots that provide a range of caretaking services such as patient monitoring and medical data collection[1]. Green et al. present a fetch-and-carry robot which assists physically impaired office workers[7]. Nourbakhsh et al. describe Sage, an educational mobile robot that gives museum tours [12].

Additionally, some researchers have begun studying how humans and robots can function as a unit, jointly participating in planning and problem solving. Laengle, Hoeniger, and Zhu discuss human and robot working in teams[9]. Bonasso addresses the use of mixed-initiative and adjustable autonomy between humans and robots[2].

2.2 Waypoint driving

Waypoint driving is one of the oldest methods of vehicle navigation. In waypoint driving, the operator specifies a series of intermediate points which must be passed *en route* to a target position. A waypoint may be chosen for a variety of reasons. It may refer to a well-known or easily identified location. It may designate a safe area or place of interest. Or it may provide a position fix to bound localization error.

Waypoint driving has numerous advantages over direct (rate or position) control. In particular, it requires less motor skill, uses less bandwidth, and can tolerate significant delay. Waypoint driving can be performed using either maps or images. Map-based driving, however, requires accurate localization and maps. Thus, for unexplored environments, most remote-driving systems are image-based.

Wilcox, Cooper, and Sato (1986) describe “Computer Aided Remote Driving (CARD)”, a stereo image based method for interplanetary teleoperation of planetary rovers[16]. Rahim discusses the “Feedback Limited Control System (FELICS)”, a video system for real-time remote driving[13]. Kay describes STRIPE, which uses still images and continuous groundplane reprojection for low-bandwidth driving over uneven terrain[8]. Matijevic discusses the “Go to waypoint” command used to operate the Sojourner rover on Mars[10].

3 Approach

During the past year, we have developed a collaborative control system which includes a safeguarded teleoperation controller, human-robot dialogue management, and a personal user interface [5][6]. We are using our collaborative control system to remotely drive Pioneer mobile robots in unknown, unstructured terrain. At present, we are using a Pioneer-AT and a Pioneer2-AT, both of which are skid-steered vehicles equipped with microprocessor-based servo controller, on-board computing and a variety of sensors.

3.1 Dialogue

Dialogue is the process of communication between two or more parties. Dialogue is a joint process: it requires sharing of information (data, symbols, context) and of control. Depending on the situation (task, environment, etc.), the form or style of dialogue will vary. However, studies of human conversation have revealed that many properties of dialogue (e.g., initiative taking) are always present[5].

In our system, dialogue arises from an exchange of messages between human and robot. Effective dialogue does not require a full language, merely one which is pertinent to the task at hand and which efficiently conveys information. Thus, we do not use natural language and we limit message content to vehicle mobility (navigation, obstacle avoidance, etc) and task specific issues.

At present, we are using approximately thirty messages to support vehicle teleoperation (Table 1). Robot commands and information statements are uni-directional. A query (to either the human or the robot) is expected to elicit a response, though the response is not guaranteed and may be delayed.

3.2 User Interface

Our current user interface (shown in Figure 1) is the PdaDriver[4]. We designed PdaDriver to enable collaborative control dialogue (i.e., the robot can query the user through the interface) and human-to-human interaction (audio and video). The current version supports simultaneous (independent) control of multiple mobile robots and runs on WindowsCE Palm-size PC’s.

Remote driving in unstructured, unknown environments requires flexible control. Because both the task and the environment may vary (depending on situation, over time, etc.), no single command-generation method is optimal for all conditions. For example, cross-country navigation and precision maneuvering have considerably different characteristics. Thus, PdaDriver provides a variety of control modes including image-based waypoint, rate/position control, and map-based waypoint.

Table 1. Vehicle teleoperation dialogue

	Category	Messages
user $\longrightarrow$ robot	robot command (*command for the robot*)	position (*pose*, *path*) rate (*translate*, *rotate*) stop camera pose (*pan*, *tilt*, *zoom*) camera config (*exposure*, *iris*) sonar config (*polling sequence*)
	query-to-robot (*question from the user*)	How are you? Command progress?
	response-from-user (*query-to-user response*)	y/n value
robot $\longrightarrow$ user	info statement (*information for the user*)	pose (*x*, *y*, *z*, *roll*, *pitch*, *yaw*) rates (*translate*, *rotate*) message (*event*, *status*, *query*) camera state (*pan*, *tilt*, *zoom*) get new image
	query-to-user (*question from the robot*)	Can I drive through (*image*)? Is this a rock (*image*)? If you answer 'y', I will stay here. [exploration] The environment is very cluttered (*map*). What is the fastest I should translate? My motors are stalled. Can you come help? Motion detected. Is this an intruder? If you answer 'y', I will follow him [surveillance] Motion control is currently turned off. Shall I enable it? Safeguards are currently turned off. Shall I enable it? Stopped due to collision danger. Disable safeguards? Stopped due to high temperature. What should the safety level be? Stopped due to low power. What should the safety level be? Stopped due to rollover danger. Can you come over and help?
	response-from-robot (*query-to-robot response*)	How are you? $\longrightarrow$ bargraphs (*health*, *rollover*, *collision*) Command progress? $\longrightarrow$ stripchart (*progress over time*)

Figure 1. PdaDriver and control modes: image (*top center*), direct (*top right*), map (*bottom center*), sensor (*bottom right*)

4 Waypoint Driving

4.1 Image-based

Remote driving is an inherently visual task, especially for unstructured or unknown terrain. Thus, we have developed a method for waypoint driving using still images. Our method was inspired by [8], but has two significant differences. First, we use a camera model which corrects for first-order radial distortion. This allows us to use wide-angle lenses. Second, instead of continuous groundplane reprojection, we use a flat-earth projection model. This simplifies computation, yet works well over short distances.

PdaDriver's "image mode" (Figure 1, top center) displays images from a robot camera. Horizontal lines overlaid on the image indicate the projected horizon line and the robot width at different depths. The user is able to position (pan and tilt) the camera by clicking in the lower-left control area. The user drives the robot by clicking a series of waypoints on the image and then pressing the go button.

Camera model

To aid the operator's perception of the remote environment, we are using color CCD cameras with wide-angle lenses. To correct for the optical distortion inherent with these lenses and to obtain a precise estimate of focal length, we use the camera model and calibration technique described by Tsai [15]. Tsai's model is based on pinhole perspective projection and incorporates five intrinsic and six extrinsic camera parameters.

Our Pioneer-AT is equipped with a forward-mounted Supercircuits PC17 (2.8 mm focal length, 60 deg HFOV). Our Pioneer2-AT has a top-mounted Sony EVI-D30 pan-tilt-zoom (3.2-38.9 mm focal length, 6.7-70 deg HFOV). Since both units have the same size CCD and because we digitize the video signal (Square NTSC, 640x480 pixels) using identical framegrabbers, the only camera parameters which differ between the two robots are focal length and first-order radial distortion coefficient.

Flat-earth projection model

To transform image points to world points (and vice versa), we use perspective projection based on a pinhole camera model. We assume that the ground plane is locally flat and that it is parallel to the camera central axis (for zero camera tilt). We perform the forward projection as:

1. compute undistorted coordinates (Tsai dewarp)
2. transform from image to CCD sensor plane
3. project from sensor plane to camera frame
4. transform from camera frame to world frame

Although this procedure computes 3D world points, we only use 2D coordinates (i.e., ground points) for driving.

Designation error

There are many factors that affect the precision of waypoint designation, and consequently, driving accuracy. Some of these factors, such as camera calibration, have relatively minor influences on the resulting projection and will not be discussed here. See [8] for a detailed discussion.

For PdaDriver, stylus input has a considerable impact. Unlike mouse-based interfaces, PDA's do not show a cursor to provide position feedback to the operator. Point selection (screen tap) is, therefore, essentially open loop. Additionally, stylus calibration can only partially compensate for touchscreen misalignment and irregularities. Thus, pointing precision may vary considerably across the display.

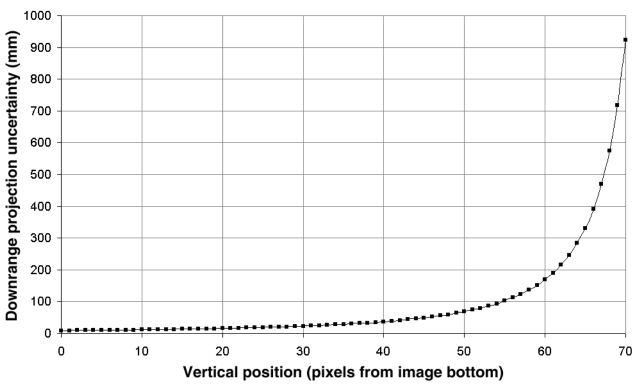

Figure 2. Projection uncertainty for zero camera tilt (downrange error due to pixel position)

The most significant factor, however, is the projection uncertainty caused by limited image resolution. Because PdaDriver is constrained by display hardware to low-resolution (208H x 156V) images, each pixel projects to a large ground-plane area. Moreover, with perspective projection and low-mounted cameras with low tilt, image points may be transformed to 3D points with high uncertainty. For example, Figure 2 shows how downrange projection uncertainty varies with vertical pixel position when there is no camera tilt. We can see from this graph that pixels near the image center may cause large driving errors.

4.1 Map-based

Although image-based driving is an efficient command mechanism, it may fail to provide sufficient contextual cues for good situational awareness. Maps can remedy this by providing reference to environmental features, explored regions and traversed path.

In PdaDriver's "map mode" (Figure 1, bottom center), the operator defines a series of waypoints by clicking points on a map. We build maps from range data (sonar, stereo, or ladar) using a 2D histogram occupancy grid. Our method is inspired by Borenstein and Koren's *HIMM* method, but differs in several respects[3].

As in HIMM, we use a 2D Cartesian histogram grid to map the environment. Each grid cell contains a certainty value *cv* that indicates the confidence of an obstacle (or free space) in the cell. Unlike HIMM, we use a signed 8-bit integer to represent certainty values (-127=*clear*, 0=*unknown*, 127=*obstacle*). This wider range improves map appearance.

Instead of HIMM's large, world-fixed grid, we use a small grid (200x200 with 10 cm x 10 cm cells) which is periodically relocated in response to robot motion. Specifically, whenever approaches a border, we perform a grid shift operation (discarding cells which are pushed over the grid boundary) to keep the robot on the map. In this way, we are able to construct useful "global" maps (i.e., up to 20x20 m) while bounding computation and memory usage.

The major difference between HIMM and our approach is how we update the histogram grid. In HIMM, and its successors (*VFH* and *VFH+*), sonar ranges are used only while the robot is moving. This reduces the impact of spurious readings due to multiple reflections and sensor noise. However, this also makes HIMM perform poorly when dynamic obstacles are present: if the robot is stopped, the map does not reflect moving objects. To address this shortcoming, we update the grid whenever a range reading is available. However, if the robot is stopped, we only use the reading if it indicates clear (i.e., no return or a large range).

In addition to range processing, we update the grid to account for localization error. We do this so that the map reflects how certain we are about the robot's pose, especially with respect to mapped features of the environment. Thus, whenever localization error increases, we globally increment/decrement all certainty values towards zero. As a consequence, local (recently mapped) areas appear "crisp" and distant (long-ago mapped) regions become fuzzy.

As an example, consider the localization of a skid-steered vehicle using only odometry. With skid-steering, rotation (e.g., in-place turning) produces larger dead-reckoning errors than translation. We can use this fact to compute confidence value change Δcv due to vehicle motion:

$$\Delta cv = (K_t \Delta t) + (K_r \Delta r) \qquad (1)$$

where Δt and Δr are position and orientation changes since the last grid update. The two constants, K_t and K_r, provide an estimate of error growth. The grid update is then:

for *each cell in grid* **do**
  if *cv > 0* **then** *cv = cv* - Δcv
  else if *cv < 0* **then** *cv = cv* + Δcv
end

Because this update is computationally expensive (i.e., it requires modifying every cell), we only perform the operation when there is considerable change in localization error.

5 Remote driving tests

We are now studying how collaborative control, human-robot dialogue, and waypoint-based driving can improve multi-robot remote driving. Our goal is to understand how to create effective human-robot teams for performing tasks in unknown and unstructured environments.

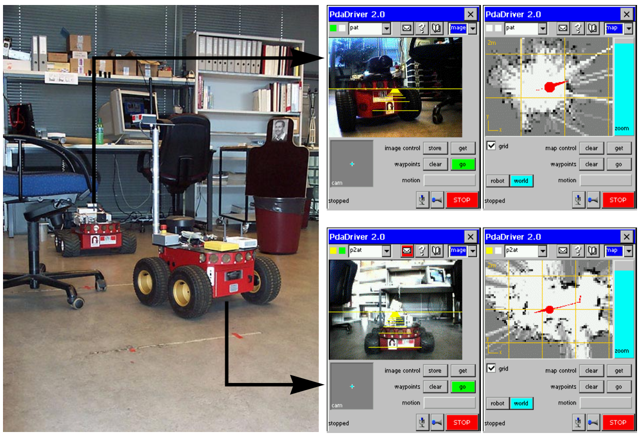

Figure 3. Indoor surveillance with two robots
left: PioneerAT (far) and Pioneer2-AT (near)
right: PdaDriver PioneerAT (top), Pioneer2-AT (bottom)

We recently conducted two tests involving a single operator and two mobile robots. In the first test, the operator used both robots to conduct surveillance in an unknown indoor environment (Figure 3). The primary tasks were to map the environment and to track intruders. To assist the human in these tasks, each robot was equipped with a motion detection module. This module detects motion by acquiring camera images and computing interframe differences whenever the robot is stationary. If the robot detects a moving object, it notifies the human and asks what to do.

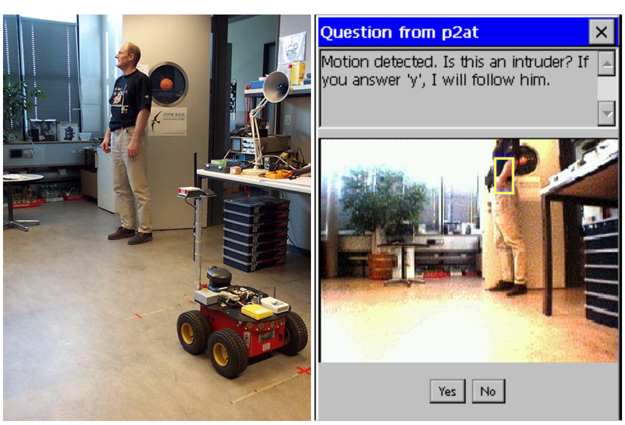

Figure 4. Human-robot interaction during surveillance

Figure 4 shows an example of this interaction occurring during the test. One of the robots has detected motion and has generated a question for the human: "*Motion detected. Is this an intruder? If you answer 'y', I will follow him.*" PdaDriver presents this question to the user and displays an image showing the motion area (marked with a bounding box). At this point, the human has the opportunity to decide whether or not an intruder is present.

In the second test, the operator remotely drove the two robots through an unfamiliar outdoor environment. The objective for this test was to perform reconnaissance in the presence of dynamic (moving) and static hazards. Because the environment had not been previously explored, the operator was forced to rely on waypoint driving and on-robot safeguarding to conduct the task.

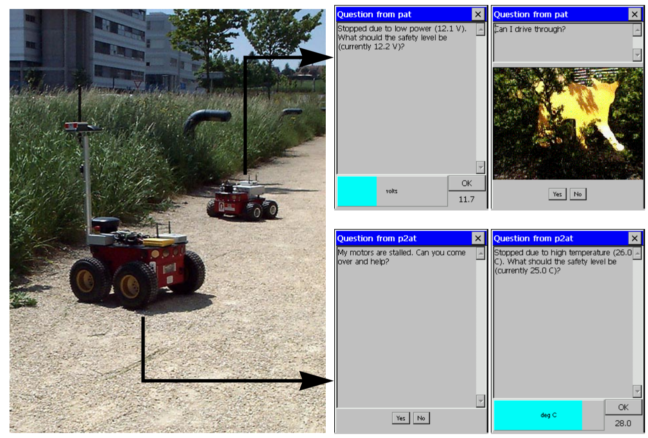

Figure 5. Cross-country driving with two robots

Figure 5 shows human-robot interaction during the second test. Since the human can only focus his attention on one robot at a time, we use collaborative control to unify and coordinate the dialogue. Specifically, we arbitrate among the questions from the robot so that the human is always presented with the one which is most urgent (in terms of safety, task priority, etc.) This allows us to maximize the human's effectiveness at performing simultaneous, parallel control. In addition, because each robot is aware that the human may not be able to respond (i.e., because he is busy or unavailable), it is free to attempt to resolve the problem on its own.

6 Discussion

6.1 Human-robot collaboration

In our testing, we found that are two key factors for achieving effective human-robot collaboration. First, roles and responsibilities must be assigned according to the capabilities of both the human and the robot. In other words, for any given task, the work needs to be partitioned and given to whomever is best equipped to handle it. Although this might seem easy to do, in practice it is not. In particular, vehicle teleoperation tasks, such as identifying obstacles in an unfamiliar environment, can be highly situation dependent. Thus, even if the robot has previously accomplished a task by itself, it may not be able to the next time without some amount of human assistance.

In the case of multi-robot remote driving by a single operator, the human usually constrains performance because he has limited sensorimotor and cognitive resources to share. Thus, we need to reduce, as much as possible, the level of attention and control the operator must dedicate to each robot. This is true whether the human controls the robots individually or as a group (in formation, as a task team, etc.). Moreover, even if one or more robots work together (i.e., robot-robot collaboration), we must still find ways to direct the human's attention to where it is needed, so that he can help solve problems.

One way to achieve this is for the human to focus on global strategy and task planning (e.g., where to go) and to allow the robots to handle the low-level details (i.e., how to get there safely). Then, whenever a robot completes a task or encounters a problem, it notifies the operator. If multiple robots, working individually or as a team, encounter problems at the same time, we arbitrate among the requests to identify the most urgent one for the human to address.

Given this approach, the second factor is clear: we must make it easy for the human to effect control and to rapidly assess the situation. In other words, we need to make the human-robot interface as efficient and as capable as possible. In our system, therefore, we designed PdaDriver to facilitate quick (single-touch) command generation, situational awareness, and human-robot dialogue.

Dialogue is particularly important when the human is operating multiple robots. Dialogue allows the operator to review what has happened, to understand problems each robot has encountered, and to be notified when his assistance is needed. Dialogue also improves context switching: enabling the human to quickly change his attention from robot to robot, directing and answering questions as needed.

6.2 Benefits of collaborative control

By enabling humans and robots to work as partners, we have found that collaborative control provides significant benefits to multi-robot remote driving. First, it allows task allocation to adapt to the situation at hand. Unlike other forms of teleoperation, in which the division of labor is defined a priori, collaborative control allows human-robot interaction and autonomy to vary as needed. If the robot is capable of handling a task autonomously, it can do so. But, if it cannot, the human can help.

Second, we have observed that collaborative control reduces the impact of operator limitations and variation on system performance. Because allows the robot to treat the operator as a limited source of planning and information, collaborative control allows use of human perception and cognition without requiring continuous or time-critical response. Hence, if the human is unavailable because he is performing other tasks, the system will still function.

Finally, we have seen that dialogue allows the human to be highly effective. By focusing attention on where it is most needed, dialogue helps to coordinate and direct problem solving. In particular, we have found that in situations where the robot does not know what to do, or in which it is working poorly, a simple human answer (a single bit of information) is often all that is required to get the robot out of trouble.

Acknowledgements

This work was partially supported by grants from the DARPA ITO Mobile Autonomous Robot Software (MARS) program, the National Science Foundation, and SAIC.

References

[1] Baltus, G. et al., "Towards personal service robots for the elderly", *WIRE*, Pittsburgh, 2000.

[2] Bonasso, P., "Issues in providing adjustable autonomy in the 3T architecture", *AAAI Spring Symposium*, Stanford, 1999.

[3] Borenstein, J., and Koren, Y., "Histogramic In-motion Mapping for mobile robot obstacle avoidance", *IEEE Journal of Robotics and Automation* 7(4), 1991.

[4] Fong, T., Thorpe, C., and Baur, C., "Advanced interfaces for vehicle teleoperation: collaborative control, sensor fusion displays, and remote driving tools", *Auton. Robots* 11:1, 2001.

[5] Fong, T., Thorpe, C., and Baur, C., "Collaboration, dialogue and human-robot interaction", *International Symposium on Robotics Research*, Lorne, Australia, 2001. (in submission)

[6] Fong, T., Thorpe, C., and Baur, C., "A safeguarded teleoperation controller", *IEEE ICAR*, Budapest, 2001.

[7] Green, A. et al., "User centered design for intelligent service robots", *IEEE RO-MAN*, Osaka, 2000.

[8] Kay, J., *STRIPE: Remote driving using limited image data*. Ph.D. dissertation, Carnegie Mellon University, 1997.

[9] Laengle, T., Hoeniger, T., and Zhu, L., "Cooperation in Human-Robot-Teams", *ICI*, St. Petersburg, 1997.

[10] Matijevic, J., "Autonomous navigation and the Sojourner microrover", *Science* 276, 1998.

[11] McGovern, D, "Experiences and results in teleoperation of land vehicles", SAND87-0646, Sandia National Lab., 1990.

[12] Nourbakhsh, I. et al., "An affective mobile robot educator with a full-time job", *Artificial Intelligence* 114(1-2), 1999.

[13] Rahim, W., "Feedback limited control system on a skid-steer vehicle", *Robotics and Remote Systems 5*, Knoxville, 1993.

[14] Sheridan, T., "Eight ultimate challenges of human-robot communication", *IEEE RO-MAN*, 1997.

[15] Tsai, R., "An efficient and accurate camera calibration technique for 3D machine vision", *IEEE CVPR*, 1986.

[16] Wilcox, B., Cooper, B., and Salo, R., "Computer Aided Remote Driving", *AUVS*, Boston, 1986.