A Personal User Interface for Collaborative Human-Robot Exploration

Terrence Fong[1,2], Nathalie Cabrol[3], Charles Thorpe[1] and Charles Baur[2]

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

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

[3]SETI Institute
NASA Ames Research Center
Moffett Field, CA 94035 USA

Keywords. Vehicle teleoperation, human-robot interaction, human-robot collaboration, user interface, planetary exploration, planetary rover

Abstract

Human-robot collaboration has significant potential to improve planetary missions. Specifically, by enabling humans and planetary rovers to work together in the field we can greatly increase mission productivity while reducing cost, particularly for surface operations such as material transport, survey, sampling, and in-situ site characterization. Thus, we are developing a personal user interface to enable EVA crew members and mobile robots to collaborate and jointly perform tasks in the field. In this paper we describe the motivation for our work, present the design and implementation of our user interface, and discuss our planned field testing methodology.

1 Introduction

1.1 Motivation

Within the next thirty years, a human crew is expected to explore the surface of Mars. Although much preparatory work has already been performed, it would be a serious misjudgment to believe that we will be able to design a productive Mars mission simply by assembling current technology and by employing the same exploration strategy as used for the Moon.

Quite simply, Mars is not the Moon. In many ways, the Martian world more closely resembles the Earth than does the Moon. Mars has a varied geology of channels, canyons, volcanoes, and ancient lakes. It has a variety of terrain and steep slopes. Wind accumulation regions and dune fields of various sizes and composition are omnipresent. Exploration of Mars, therefore, will require that equipment and procedures be adapted to such conditions. In particular, the success of Mars missions will strongly depend on solutions that need to be identified, developed and tested[3].

Moreover, in considering human exploration and exploitation of Mars, a number of critical issues must be resolved for both surface system support and science operations. Long-term human presence on Mars will require significant infrastructure for habitat construction, material transport, etc. Characterization of Martian geology will demand long-distance and long-duration excursions. Search for water or life will require systems which can cope with challenging geography (e.g., cliff walls).

Considerable focus has already been given to human and to robot systems for planetary surfaces. Scant attention, however, has been paid to the development of joint human-robot systems. Yet, such systems are attractive for numerous reasons. Rovers could be used to carry support equipment, samples, and instruments. An autonomous rover could follow a scientist in geological sampling sorties. Rovers could also provide contingency life support or emergency transport. More advanced rovers could be used to perform sampling and site documentation[5].

Figure 1. Astronaut-Rover Interaction Field Test (1999)

Given the potential of human-robot systems to improve the safety and performance of surface operations while reducing cost, it is clear that we need to better understand how to build and use such systems. A first step in this direction was taken in February 1999 with the "Astronaut-Rover Interaction Field Test" (ASRO). ASRO was conducted at Silver Lake, Mojave Desert and involved a space suited test subject and a rover (Figure 1) [3][17]. ASRO was designed as an exploratory effort to identify issues, requirements, scenarios, and needed technologies for future human-robot surface exploration.

1.2 Approach

Our long-term goal is to create the tools and techniques necessary for human-robot exploration. We believe that enabling direct collaboration in the field between humans and planetary rovers will fundamentally advance NASA's planetary mission capabilities. In particular, we are convinced that human-robot collaboration will significantly improve mission productivity, particularly for surface operations such as material transport, survey, sampling, and in-situ site characterization.

We recognize, however, that the development of robotic systems capable of working side-by-side with humans will not occur overnight. Full human-robot collaboration will require, at a minimum, the following:

- human-robot interfaces designed for field use
- mixed-initiative and adjustable autonomy techniques for human-robot systems
- procedures for human-robot surface operations
- short-term autonomy for rovers operating in natural, unstructured environments

Thus, during the past year, we have begun to develop a personal user interface for EVA crew members. We designed this interface to enable humans and robots to communicate and to collaborate with each other in the field. Specifically, the human is able to supervise the robot (generating commands, reviewing sensor/performance data, etc.) and the robot is able to take advantage of human capabilities (perception, planning, etc.).

Figure 2. PdaDriver: a personal user interface for collaborative human-robot exploration

Our interface, the *PdaDriver*, is shown in Figure 2. It incorporates a variety of tools and techniques we have been developing to make vehicle teleoperation easier for all users, novices and experts alike. In particular, PdaDriver supports *collaborative control* and runs on WindowsCE[1] Palm-size PC hardware[6].

_____

1. WindowsCE is a trademark of Microsoft, Inc.

2 Related Work

2.1 Robotic Planetary Exploration

To date, there have been three planetary exploration missions (Lunokhod I, Lunokhod II, and Sojourner) involving science rovers[4][19]. In these missions, the rover was remotely driven via rate or waypoint control by a team of earth-based operators.

Numerous terrestrial field tests have been conducted with science rovers in planetary (Mars and lunar) surface analog environments[2][11][16]. In these tests, trained operators as well as planetary science teams remotely controlled rovers via high-bandwidth communication links and multimodal/multisensor interfaces.

2.2 Human-Robot Exploration

The ASRO field test was the first time that an astronaut and a rover explored a planetary surface together[3][17]. During the test, four human-robot interaction scenarios were studied. However, only a single suited subject was tested, operator command and communication protocols were not rigorously defined, and test time was limited. Nevertheless, ASRO did provide valuable insight into human-robot exploration. Most significantly, it underscored the importance of grounding rover technology development in real-world tests.

Vandapel et al. conducted a series of experiments during the 1999 Haughton-Mars Project expedition (Devon Island, Canada) to evaluate robotic needs for supporting the field activities of biologists and geologists[18]. This study examined four robot functions: scout, equipment transport, caretaker (monitor scientist during EVA), and co-worker (deploy sensor, register data, explore region). Vandapel et al. claim that the major challenges for robotic EVA support are the integration of geological knowledge, autonomous exploration, and human-robot interaction.

2.3 Supervisory control interfaces

Supervisory control interfaces are designed for high-level command generation. To effect supervisory control, the operator divides a problem into a sequence of sub-tasks which the robot then executes on its own. Thus, supervisory control requires that the robot be able to achieve goals (even limited ones) while keeping itself safe.

Supervisory control interfaces are well-suited for applications which must operate with low-bandwidth communication links or in the presence of high delay. To date, there have been few supervisory control interfaces for planetary exploration and all have required significant display and computational resources[1][4][15].

3 Design

3.1 Design Methods

There are many factors which shape the way interfaces are designed: psychological, economic, organizational, and even political[13]. Interfaces may be designed to meet detailed requirements or to fulfill a perceived need. Thus, interface design is context sensitive: designers must understand the task they are designing for and that an interface well designed for one task may be inappropriate for another task that seems superficially similar.

When an interface is well-designed and easy-to-use, it becomes transparent: we take it for granted and can use it with minimal effort. On the other hand, if an interface is poorly crafted and difficult to understand, it becomes burdensome: limiting performance and making work tiresome. Hence, the value of an interface is largely determined by its how easily and effectively it allows the user to perform tasks.

However, interface design is not simply about ease of learning. Making user interfaces "easy to use" is a good goal, but a different emphasis is sometimes needed to support the skilled user. Moreover, breakdowns in human performance can be caused by a variety of problems. Some breakdowns are caused by physical or technological issues (e.g., display glare). Others breakdowns are psychological in nature and user inefficiency can be linked to cognitive limitations including memory capacity, learning, and attention failures[13].

Interface design has traditionally followed a sequential approach. Task analysis is performed to identify design objectives, which lead to detailed requirements. These specifications are used to select a design method, to guide implementation and as a basis for evaluation. Among the more widely used design methods are heuristic design, guidelines (standards, style-guides, etc.) and iterative design (rapid prototyping).

HCI methods

Most modern computer interfaces are designed with user-centered methods. In user-centered design, the basic goal is to support human activity: to enable humans to do things faster, with fewer errors, and with greater quality[14]. Thus, user studies are often conducted (prior to interface development) in order to understand the user's activities, how they perform them and what they need in support. The three primary methods for studying users are interviews, observation, and questionnaires.

Once a design is completed, some form of evaluation is generally needed to assess the usability of the interface. Qualitatively, usability can mean that an interface is "easy-to-use", "easy-to-learn", or that it is "well-suited" for a task. Quantitatively, usability may be specified as in terms of learning (training time, skill retention), performance (speed, accuracy, throughput), error (incidence, ability to recover), or even psychological experience (satisfaction, aggravation).

The purpose of evaluation, however, is not merely to compute a single measure of the interface (although this is sometimes necessary to satisfy project requirements). Rather, the true value of evaluation is to provide an informed critique of an interface, identifying its strengths and its weaknesses so that it can be improved[14].

HCI methods for teleoperation

In spite of the proven success of HCI methods at increasing performance and reducing error, there has been little use of these methods in robotics. One hypothesis is that mainstream HCI design and evaluation techniques are ill-suited for human-robot interfaces[9]. Cognitive walkthrough, for example, is generally performed for multi-dialogue interfaces and from the viewpoint of novice users, both of which are rare in teleoperation systems.

Green, Huttenrauch, and Norman describe the development of a personal service robot using methods from user centered system design[10]. The authors performed a questionnaire study to assess attitudes towards personal service robots and then performed task analysis to ascertain user needs and work tasks. The interface design was performed via a "thinking aloud" walkthrough, followed by heuristic evaluation and guidelines checking.

Graves proposes a set of heuristics for the design of teleoperator user interfaces[9]. The primary heuristics are: be consistent, make it simple, support human decision making, keep the user in control, and assist error recovery. Graves also stresses the importance of contextual design, i.e., interface input/output should be appropriate for the working environment.

3.2 Requirements

We believe there are clear benefits to be gained from humans and robots working together in the field. In particular, if we treat a robot not as tool, but rather as a partner, we find that we can accomplish more meaningful work and achieve better results. Thus, we have developed a new system model for teleoperation called collaborative control. In this model, a human and a robot work as partners (if not peers), collaborating to perform tasks and to achieve common goals[6].

For collaborative control to be effective, however, the human-robot interface must facilitate communication. That is, the interface should enable the human to express intent, to provide information, and to understand and interpret what the robot has accomplished. At the same

time, the user interface allows the robot to make requests, to ask for help, and to better interact with the human.

For example, a geologist should be able to say (not necessarily using natural language) to a robot,

> "Drive along this path and explore the region. If you find an interesting rock, tell me."

Similarly, as the robot is executing the task, it should be able to use the same interface to query the human or to convey information: "How should I get around this obstacle?", "Is this an interesting rock?", or "Come help me! I'm stuck!"

Based on the ASRO field test findings and on the needs of collaborative control, we defined the following interface requirements:

- field-portable (lightweight, rugged)
- require minimal infrastructure
- enable rover motion control
- provide interaction with rover autonomy
- support human-robot dialogue
- usable by non-roboticists
- short training period (less than 1 hour)
- rapid command generation
- maintain high situational awareness

3.3 Approach

We designed the PdaDriver using a combination of heuristic design, heuristic evaluation, and cognitive walkthrough. We chose these methods because they are efficient (low-cost, rapid, easy to perform), can be used throughout the design process, and have been proven to produce high quality interfaces in a variety of domains.

Heuristic evaluation, or "usability inspection", is an informal review method[14]. With this method, a team of reviewers use a set of heuristics to critique an interface. The result is a list of problems (design flaws, implementation errors, etc.) which could reduce usability. The primary weakness of heuristic evaluation is that problem identification (types, coverage, etc.) is strongly correlated to the reviewers' experience and fastidiousness.

Cognitive walkthrough is an evaluation method which simulates the way users explore and gain familiarity with an interface[14]. During cognitive walkthrough, evaluators perform a structured examination by working step-by-step through the performance of a task, trying to predict the actions a user will take. This produces considerable insight into the design, especially regarding ease of use and likelihood of user errors.

Table 1 lists the heuristics we used during interface design and evaluation. We found that the PDA-related heuristics ("single-click interaction" and "design for small screen") were the two hardest to follow. This is not surprising, given the difficulty of creating effective PDA interfaces. In addition, we debated considerably over the use of multiple modes. Although we would have preferred a modeless interface (to speed learning and habituation), we do not believe that there is a single, optimal method for all vehicle teleoperation tasks. Thus, we chose to create task-specific modes, each with distinctive appearance and function (to minimize modal errors).

Table 1. Design and evaluation heuristics

heuristic	meaning / effect
single-click interaction	facilitate PDA input, avoid text entry where possible
design for small screen	present only task-relevant information
effectively use color	avoid excessive hue, use familiar color mapping
consistency	handle input and display feedback consistently
simplicity	make tasks easy to perform and displays easy to understand
task-specific modes	create separate modes for each distinct task

4 Implementation

4.1 Architecture

PdaDriver provides a variety of command modes, enables human-to-robot dialogue, and supports human-to-human interaction (audio and video). The current version supports simultaneous (independent) control of multiple mobile robots and runs on WindowsCE-based PDA's such as the Casio Cassiopeia[1]. We use WindowsCE-based PDA's because they provide high quality color display (12-bit or 16-bit) and support high-bandwidth communication (wired and wireless ethernet).

The PdaDriver architecture is shown in Figure 3. When operating, PdaDriver connects to a collaborative control system on each mobile robot being operated[6]. This system uses network messaging to connect a variety of modules including human-robot dialogue management, image capture, robot controller, sensor-based map making, and task-specific behaviors (e.g., rock finder for geologic survey). We use a safeguarded teleoperation controller to keep the robot safe regardless of environmental conditions and operator input[7].

We implemented PdaDriver using Personal Java[2], a Java application environment designed for personal con-

---

1. Cassiopeia is a trademark of Casio, Inc.
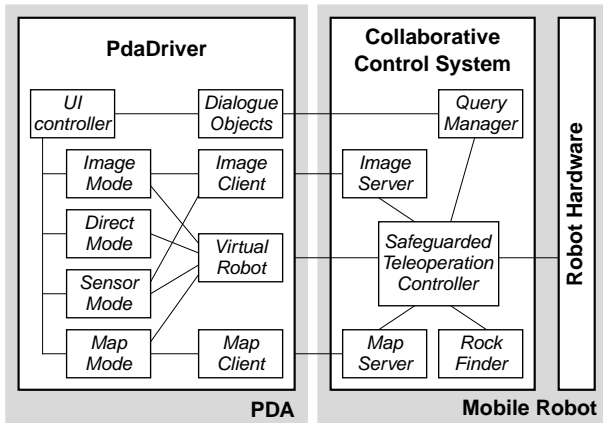2. PersonalJava and Java are trademarks of Sun Microsystems, Inc.

Figure 3. PdaDriver architecture

sumer devices. The top-level Java object is the UI controller, which operates the user interface. The UI controller performs a variety of tasks including interface initialization/shutdown, object activation, and communication link monitoring. The other primary interface objects are the control and dialogue modes (described below) and the Virtual Robot.

The VirtualRobot encapsulates robot-specific parameters and maintains a local copy of robot state (pose, sensor readings, etc.). This enables interface objects to have continuous access to robot data. The VirtualRobot also provides coordinate frame conversion (sensor, world, local) and a set of standard command methods. When the user issues a command, the object receiving the input invokes a method in the Virtual Robot, which then outputs a command to the teleoperation controller. This level of indirection facilitates integration and use of different mobile robots

4.2 Control Modes

Vehicle teleoperation 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, we designed the PdaDriver with a variety of control modes (see Figure 4).

Direct Mode
Direct mode provides rate and position control of robot pose. This mode is most appropriate for line-of-sight precision driving tasks such as fine worksite positioning. It is also useful for commanding constant rates (e.g., forward translation) when long-distance or long-duration motion must be performed.

In direct mode, a graduated cross is shown (Figure 4, left). Clicking on the vertical axis commands translation and on the horizontal axis rotation (all motions are performed in the robot's coordinate frame). The *rate* and *pose* buttons allow the user to switch between rate and relative position control. A scale bar is used to change command magnitude. The center circle indicates the size of the robot (only shown with position control).

Image Mode
Remote driving is an inherently visual task. Thus, image mode is designed to facilitate image-based, waypoint driving. Our method was inspired by Kay [12], but has two significant differences. First, instead of continuous groundplane reprojection, we use a flat-earth model. This simplifies computation, yet works well over short distances. Second, we use a camera model which corrects for first-order radial distortion. This allows us to use wide-angle lenses (70˚ FOV).

Image mode (Figure 4, middle left) displays images from the robot-mounted camera. Images are retrieved using an event-driven model to minimize bandwidth usage[7]. Horizontal lines overlaid on the image indicate the projected horizon line and the robot width at different
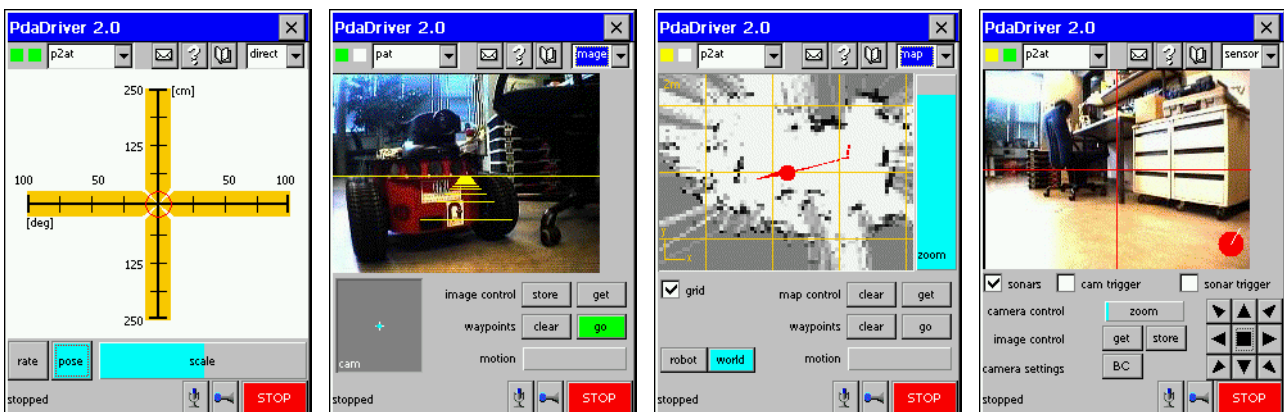

Figure 4. PdaDriver control modes. *left to right*: direct, image, map, sensor

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. As the robot moves, a status bar displays the robot's progress.

Map Mode

Although image-based driving is an efficient command mechanism, it may not provide sufficient contextual cues for good situational awareness. Map mode remedies this by providing reference to environmental features and explored regions.

Map mode complements image mode: waypoints entered in map mode appear on the image mode display and vice versa. In addition, map mode is useful for entering waypoints which are problematic in image mode (outside the field-of-view, near the image horizon, etc.).

Map mode (Figure 4, middle right) displays a map (a histogram-based occupancy grid built from range data) in either robot (local) or global (world) coordinates. As with image mode, the user drives the robot by clicking a series of waypoints and then pressing the *go* button. As the robot moves, the traversed path is shown in the display.

Sensor Mode

Sensor mode (Figure 4, right) is used to control the robot's on-board sensors. In our current system, the user can command the robot's camera (position and imaging settings), configure sonars and activate movement detection triggers.

4.3 Dialogue Displays

In our collaborative control system, dialogue results from messages exchanged between the human and the robot. PdaDriver supports this type of human-robot interaction through dialogue displays. In particular, it enables both the human and robot to ask questions of the other and to receive the responses.

Robot to user

Our current collaborative control system supports a variety of *query-to-user* messages related to robot health, configuration, and task performance. These questions are categorized into two types based on the expected response: y/n (requires y or n response) and value (requires a decimal value response).

Whenever the robot has a question to ask the human, it sends a message to the QueryManager (see Figure 3). Since the multiple robot modules may ask questions of the human at the same time, the QueryManager performs query arbitration: a mechanism for choosing which questions to ask based on both immediate (local) needs and overall (global) strategy.
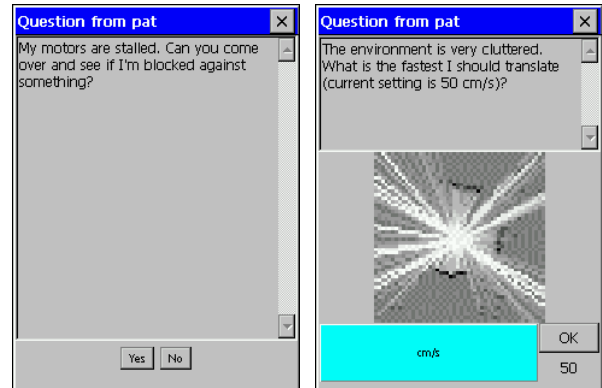

Figure 5. Queries to the user

Figure 5 shows two queries to the user as displayed by PdaDriver. In the y/n query on the left, the robot is asking if the human can provide assistance. If the human answers 'n', the robot must then try to resolve the problem by itself. In the value query on the right, the robot is asking for guidance in setting a safeguard threshold. Because the answer to this question requires expertise (some level of teleoperation experience), the collaborative controller is designed not to present the query to novices[6]. In other words, collaborative control adapts human-robot dialogue to the user.

User to robot

In PdaDriver, most human to robot dialogue is expressed via the control modes. This is because, under normal conditions, the human gives commands to the robot and receives feedback through the control displays. There are times, however, when the human may need to query the robot. In particular, if the robot has been operating autonomously without monitoring or supervision, the human may want to quickly know how the robot is doing.
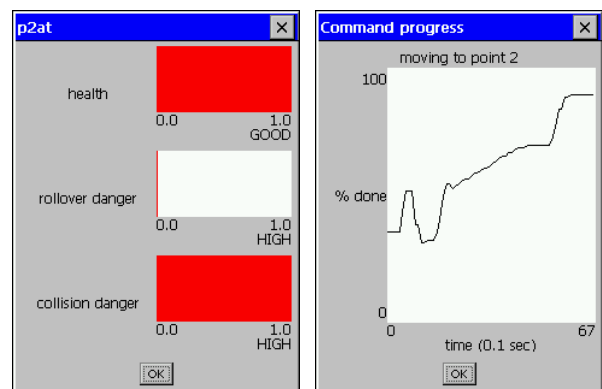

Figure 6. Responses from the robot

Figure 6 shows the robot's response to two user questions. The left image shows the response to "How are you?": the robot's current health and safeguard metrics. The right image presents the response to "Command progress?": it shows a snapshot of current task execution.

5 Preliminary Results

To study human-robot interaction during collaborative exploration, we have developed a robot module which locates "interesting rocks". *RockFinder* is not intended for field use (i.e., it does not examine geologic properties), but merely serves as an example of the type of work a rover could perform when assisting an EVA scientist.

5.1 RockFinder

RockFinder searches camera images for contiguous regions having a certain color signature. Color segmentation is an efficient method for object detection because it can be performed via simple pixel filtering. To reduce the effect of varying illumination, shadows, and scene geometry, we transform image pixels into a normalized color space before filtering[8]. We use morphological filters to reduce image noise and to merge scattered pixels.
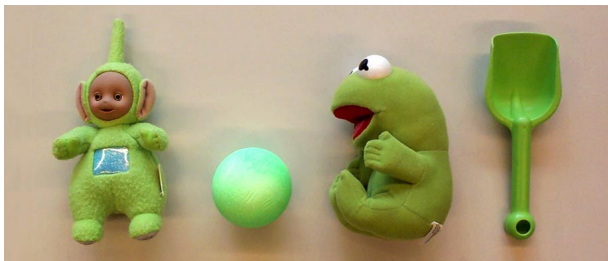

Figure 7. Some interesting "rocks"

Once an image has segmented, RockFinder labels each region exceeding certain extents as a potential "rock". Figure 7 shows several of the "rocks" that Rock-Finder is designed to locate.
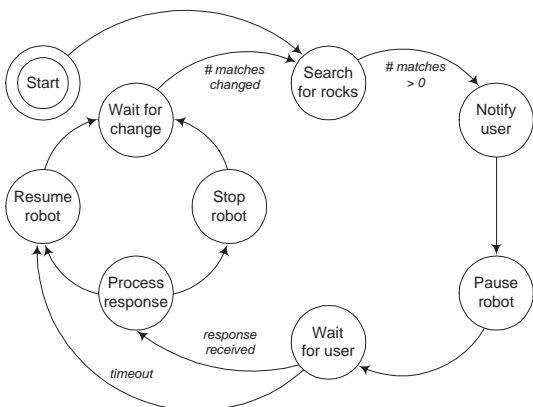

Figure 8. RockFinder state machine

RockFinder operates via a simple state machine (Figure 8). Whenever RockFinder finds a "rock", it pauses the robot and asks the human for confirmation. If the human says that the object is not a rock, or if he does not respond, RockFinder releases the robot and restarts its search.

5.2 Human-Robot Interaction

We are now studying how human-robot interaction influences collaborative work practice. In particular, we are trying to understand at what point (and to what extent) does dialogue allow the human and robot to work together as partners. In addition, we are interested in identifying promising human-robot exploration scenarios and finding quantitative ways of measuring human-robot task performance.

As a first step, we recently conducted a test using RockFinder in a cluttered, indoor environment. For this test, the central task was to search for interesting "rocks". We chose this task to represent part of the work normally performed by a geologist during site characterization. To perform the task, the human and robot collaborate: the human assigns the robot a region to examine; the robot performs the search autonomously and notifies the human whenever it finds potential samples.
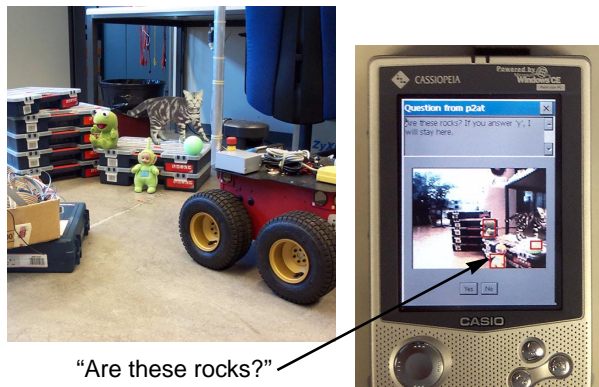

"Are these rocks?"

Figure 9. Collaborative exploration

Figure 9 shows an example of this interaction occurring during the test. RockFinder has identified three "rocks" immediately in front of the robot and has generated a question for the human: "Are these rocks? If you answer 'y', I will stay here." PdaDriver presents this question to the user and displays an image showing the objects (each marked with a bounding box). At this point, the human has the opportunity to decide whether or not to examine the "rocks" in more detail.

In this particular case, human-robot interaction strongly resembles the relationship between a research assistant and a scientist. Specifically, we have partitioned the task based on expertise: the research assistant (the robot) knows enough to identify potential samples and the scientist (the human) has the skill to decide if the samples are worth keeping. In this way, with human and robot collaborating, exploration becomes very efficient: the human is freed from performing a tedious task and the robot is able to search even though its on-board autonomy (e.g. RockFinder) is limited.

6 Future Work

6.1 Rover Integration

To date, we have conducted our development with Pioneer™ mobile robots. Although these robots are designed for outdoor use, they have limited capability for rough terrain and are not well suited for use in Mars analog environments such as Silver Lake or Haughton Crater.

Thus, we would like to integrate the PdaDriver with a planetary exploration relevant rover such as K9 (NASA ARC). K9 is a FIDO class rover and has a six-wheel rocker-bogie for multi-kilometer traverses over medium sized (30 cm) rocks. In addition, K9 has a broad sensor suite including stereo imagers, an IMU, and a range of field science instruments (e.g., spectrometer).

6.2 Field Testing

We plan to conduct a series of field tests to evaluate the performance of our collaborative exploration system and to identify areas for improvement. Specifically, we plan to examine in greater detail the human-robot interaction scenarios studied in the ASRO field test:

Rover as a Scout. Prior to the surface EVA, the rover examines the traverse area, establishes favorable sites for geology and biology.

Rover as Video Coverage Assistant. The rover provides video coverage of the suited activity.

Rover as Science Field Assistant. When the EVA crew member identifies a feature of interest, the rover documents the site.

Rover as Technical Field Assistant. The rover assists the EVA crewmember by carrying tools and returning samples to a lander or base station.

In addition to the ASRO scenarios, we are considering two other scenarios:

Rover as Infrastructure Assistant. The rover performs basic site preparation tasks such as cable spooling, antenna deployment, etc.

Rover as Science Site Assistant. The rover performs scientific site preparation tasks such as drilling or trenching.

For all six scenarios, we plan to develop work models and metrics to evaluate human-robot collaboration. Specifically, we intend to study collaborative work practice (who does what, when, and how), to measure performance (task achievement, completion time, rover utilization, etc.), and to analyze failure modes (to guide future rover autonomy and interface development).

References

[1] P. Backes, et al., "The Web Interface for Telescience", in *Proceedings of IEEE ICRA* (Albuquerque, NM), 1997.

[2] N. Cabrol et al., "Atacama I: Science Results of the 1997 Nomad Rover Field Test in the Atacama Desert, Chile", in *Proceedings of LPS Conference* (Houston, TX), 1998.

[3] N. Cabrol (ed.), *Proc. of the Astronaut-Rover Interaction Post Silver Lake Field Test Debriefing*. NASA Ames, 1999.

[4] B. Cooper, "Driving on the Surface of Mars Using the Rover Control Workstation", in *Proc. of SpaceOps* (Tokyo, Japan), 1998.

[5] M. Duke, *Mars Surface Mission Workshop*, LPI Contribution 934, Lunar Planetary Institute, Houston, TX, 1998.

[6] T. Fong, et al. "Advanced Interfaces for Vehicle Teleoperation: Collaborative Control, Sensor Fusion Displays, and Remote Driving Tools", *Autonomous Robots* 11(1), 2001.

[7] T. Fong, C. Thorpe, and C. Baur, "A Safeguarded Teleoperation Controller", in *Proceedings of IEEE ICAR* (Budapest, Hungary), August 2001.

[8] T. Gevers and W. Smeulders, "Object Recognition Based on Photometric Color Invariants", in *Proceedings of SCIA* (Lappeenranta, Finland), 1997.

[9] A. Graves, "User Interface Issues in Teleoperation", De Montfort University, Leicester, United Kingdom, 1998.

[10] A. Green, H. Huttenrauch, and M. Norman, "User Centered Design for Intelligent Service Robots", in *Proceedings of IEEE RO-MAN* (Osaka, Japan), 2000.

[11] B. Jolliff et al., "Remotely-sensed Geology from Lander-based to Orbital Perspectives: Results of FIDO Rover Field Tests", Lunar and Planetary Institute, Houston, TX, 2000.

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

[13] M. Lansdale and T. Ormerod, *Understanding Interfaces*, Academic Press, London, 1994.

[14] W. Newman and M. Lamming, *Interactive System Design*, Addison-Wesley, New York, 1995.

[15] L. Nguyen et al., "VR Interfaces for Visualization and Control of Remote Vehicles", *Auton. Robots* 11(1), 2001.

[16] C. Stoker et al., "The 1999 Silver Lake Marsokhod Field Test: Mission Overview, Data Sets, and Summary of Results", *JGR-Planets* (in press).

[17] R. Trevino et al., "First Astronaut - Rover Interaction Field Test", in *Proc. of SAE ICES* (Toulouse, France), July 2000.

[18] N. Vandapel et al. "The EVA Robotic Assistant Experiment on the Haughton-Mars 99 Expedition", in *Proc.of Mars Society Convention* (Boulder, Colorado), 1999.

[19] A. Vinogradjv (Ed.), *Mobile laboratory on the Moon Lunokhod-1* (1), Nauka, Moscow, 1971.