Novel interfaces for remote driving: gesture, haptic and PDA

Terrence Fong[a*], François Conti[b], Sébastien Grange[b], Charles Baur[b]

[a]The Robotics Institute, Carnegie Mellon University, Pittsburgh, Pennsylvania 15213 USA
[b]L'Ecole Polytechnique Fédérale de Lausanne, CH-1015 Lausanne EPFL, Switzerland

ABSTRACT

Remote driving is a difficult task. Not only do operators have problems perceiving and evaluating the remote environment, but they frequently make incorrect or sub-optimal control decisions. Thus, there is a need to develop alternative approaches which make remote driving easier and more productive. To address this need, we have developed three novel user interfaces: *GestureDriver*, *HapticDriver* and *PdaDriver*. In this paper, we present the motivation for and design of each interface. We also discuss research issues related to the use of gesture, haptics, and palm-size computers for remote driving. Finally, we describe lessons learned, potential applications and planned extensions for each interface.

Keywords: Vehicle teleoperation, remote driving, visual gesturing, haptic interface, personal digital assistant, PDA

1. INTRODUCTION

User interfaces for remote driving have remained largely unchanged during the past fifty years. To this day, almost all remote driving systems are rate-controlled: a trained operator adjusts a remote vehicle's translation and rotation rates via hand-controllers while receiving feedback from multiple video and data displays. Such systems can be difficult to use (especially in unknown or unstructured environments), expensive to build, time consuming to deploy, and require significant training[1].

Moreover, remote driving is often problematic. Loss of situational awareness, poor attitude and depth judgement, inadequate perception of the remote environment, and failure to detect obstacles are all common occurrences. Even if a vehicle has autonomous capabilities, poor communications (low-bandwidth and/or high-delay), operator workload, and dynamic factors (e.g., changing environment) may still reduce performance[2].

Our objective is to make remote driving easier and more productive. Thus, we have developed three novel user interfaces: GestureDriver, HapticDriver and PdaDriver. *GestureDriver* uses visual gesture recognition for flexible, operator-adaptive vehicle control. *HapticDriver* facilitates precision driving tasks (docking, maneuvering in cluttered spaces, etc.) via haptic feedback. *PdaDriver* enables remote driving anywhere and anytime using a Palm-size computer and low-bandwidth communications.

2. RELATED RESEARCH

2.1. Vehicle teleoperation

In vehicle teleoperation, there are three basic problems: determining where the vehicle is currently located, deciding where it should be, and moving it. These problems can be difficult to solve, particularly when the vehicle is operating in an unknown or unstructured environment. Furthermore, humans in continuous control directly limit performance. Specifically, poor performance (task achievement) and vehicle failure (collision, roll over, etc.) are often attributable to operator error.

The majority of vehicle teleoperation research has centered on rate-control systems for hazardous environments (air, ground, and underwater). It has been shown that even with *a priori* knowledge (e.g., a map), operators can quickly become disoriented when forced to rely on rate-control and video feedback[3]. Yet, for applications such as unmanned aerial reconnaissance, this approach continues to be state-of-the-art. Recently, however, multi-modal operator interfaces and supervisory (or "semi-autonomous") control have begun to appear. These methods have proven useful for a number of exploration robots[4].

*Correspondence: e-mail: terry@cs.cmu.edu; WWW: http://www.cs.cmu.edu/~terry

2.2. Visual gesturing

Gesture recognition, or the identification and classification of human movements, is being widely studied as a mechanism for computer input. While there are many human gesture analysis tools based on invasive techniques[5], few robust visual gesturing systems exist. Reliable, real-time, three-dimensional tracking of humans (head, hands, body, etc.) is a difficult task, and many vision-based gesture recognition systems restrict their application to a constrained environment.

Pfinder[6] and *Spfinder* use adaptive background subtraction and pixel classification techniques to track humans in a static environment. A blob model of the body is used to extract and analyzes gestures for interaction with a playful virtual pet and American Sign Language recognition[7]. Other systems have used visual gesturing to command devices or to perform simple interaction with virtual world[8,9]. In general, a short gesture grammar is coupled with a simple tracker to identify commands, thus limiting interaction to the tracker's capabilities.

2.3. Haptics

Kinesthetic displays have long been used for teleoperation, particularly for telemanipulation. These displays allow perception of a remote environment by providing force and position feedback to the operator[10,11]. Depending on the application, force feedback can significantly contribute to the ease or success of performing a task[12]. There are a large number of existing kinesthetic displays (e.g., force-reflecting joysticks), most of which were developed for telemanipulation tasks.

More recently, considerable attention has been given to the development of haptic displays. These displays provide both kinesthetic and tactile feedback, allowing operators to sense contact, texture, and other mechanical attributes. Haptic displays are being increasingly used for applications including art, medical training, and virtual environments[13,14].

2.4. Personal Digital Assistants

Once limited to the academic community, Personal Digital Assistants (PDA)'s are now commonplace. PDA's are attractive interface devices because they are lightweight, extremely portable, and feature touch-sensitive displays. At the same time, however, current PDA's (especially the Palm Pilot*) have slow processors, limited memory/storage, and small displays. Thus, PDA applications have generally be limited to personal information tasks such as address books, schedulers, and note taking.

Recently, however, researchers have started applying PDA's to less mundane tasks. The *Pebbles* project is using multiple PDA's connected to a single PC to enable shared information access and device control[17]. Tools developed by the Pebbles project allow multiple people to simultaneously control the PC's mouse, keyboard and applications (e.g., presentation software). Perzanowski et. al. are using a PDA as part of a multi-modal interface for interacting with an autonomous robot[18]. In this system, the PDA is used in command and control situations to direct the robot and to disambiguate natural language inputs.

3. GESTURE DRIVER

3.1. Overview

GestureDriver is a remote driving interface based on visual gesturing. Hand motions are tracked with a color and stereo vision system and classified into gestures using a simple geometric model. The gestures are then mapped into motion commands which are transmitted to the remote vehicle.

Visual gesturing offers several advantages over traditional methods. To start, the interface is passive: it does not require the user to master special hardware or to wear tracking tags. Thus, the interface is easy to deploy and can be used anywhere in the field of view of the visual tracker. This flexibility is difficult to achieve with hand controllers such as 3-axis joysticks.

Vision also allows a variety gesture interpretations to be used. Since the interpretation is software based, it is possible to customize the interpretation to minimize sensorimotor workload, to accommodate operator preferences and to fit the task being performed. Furthermore, the mapping can be varied in real-time, automatically adapting to the operator. This offers the potential to reduce learning time and to improve system efficiency.

*Palm Pilot is a trademark of 3Com, Inc.

3.2. Design

3.2.1. Human Oriented Tracking (HOT)

The GestureDriver is an application of the Human Oriented Tracking (HOT) library[19]. HOT is a layered architecture for active interfaces and provides a robust feature tracker, geometric and dynamic feature modeling, and parametric activity monitoring. The HOT architecture is presented in Figure 1.

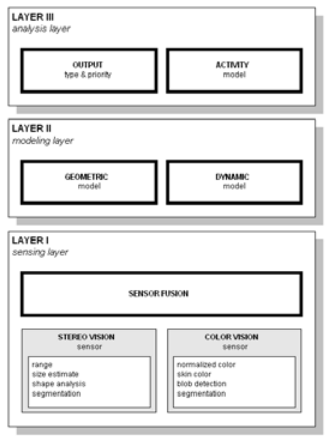

Figure 1. HOT architecture

HOT's feature tracker combines normalized color filtering with stereo vision. Normalized color provides fast 2D object localization, while stereo provides shape, size and range measurements (see Figure 2). HOT's modeling layer processes the tracker output using a Kalman filter to build a geometric model and to segment feature motion.

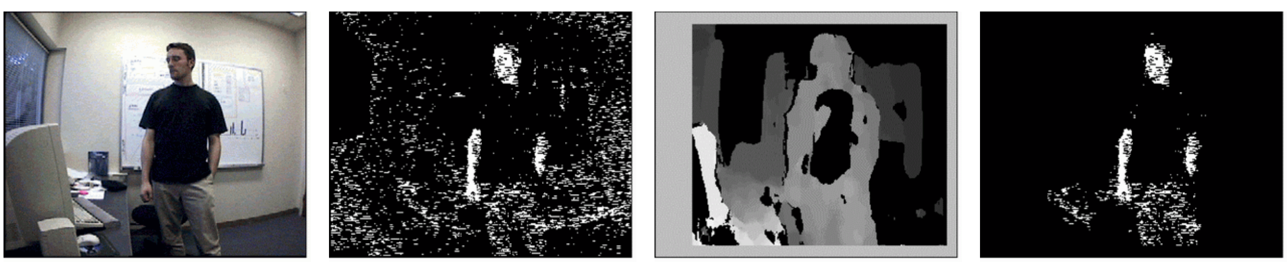

Figure 2. Color and range image filtering (left to right: original image, normalized color filter, range filter, combined filter)

3.2.2. Gesture-based driving

GestureDriver provides several interpretations for mapping gestures to commands. For example, the *virtual joystick* interprets operator hand motion as a two-axis joystick (see Figure 3). To start, the operator raises his left hand to activate the gesture system. The operator then uses his right hand to specify direction and command magnitude.

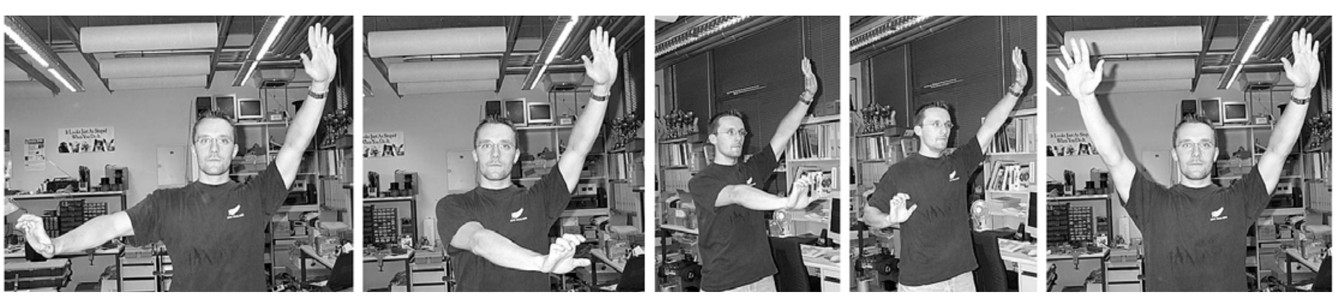

Figure 3. Virtual joystick mode. The right hand position indicates (left to right) right, left, forward, reverse, stop.

We designed two driving modes, direct and indirect, using the virtual joystick. In *direct* mode, hand positions are used as direct input to the robot. Distance from a reference point (defined by the user) sets the vehicle's speed, while orientation controls the vehicle's heading. This mode is easy to learn and is well-suited for continuous control. In *indirect* mode, vectors are extracted from user gestures. Vector magnitude sets the vehicle's speed, while vector orientation controls the vehicle's heading. This mode allows a user to position a robot as if he is "pushing" it in a specific direction.

In both modes, the user is able to switch between absolute (world frame) and relative (robot frame) positioning. Also, if the vision system is unable to locate the user's hands, or if the user does not provide control input for an extended period of time, the robot is halted.

3.3. Results

When we initially tested GestureDriver, we found that users had difficulty controlling the robot. Analysis revealed that this was due to the localization accuracy of the HOT tracker. Specifically, the stereo method provides fairly coarse range maps and is somewhat noisy (even under constant illumination). Once users were familiarized with the tracker's limitations, they quickly learned to accurately position the robot. Figure 4 shows a user driving a robot using virtual joystick gestures.

Figure 4. Visual gesturing for vehicle teleoperation

GestureDriver's strength lies in its flexibility. The system works well almost anywhere within the vision system's field of view. Thus, users were free to move about when they were not directly commanding the robot. Additionally, GestureDriver was able to easily accommodate users of different sizes and with different control preferences.

However, giving commands with visual gestures is not as easy as one might believe. Although humans routinely give commands using visual gestures, these gestures are often ambiguous, imprecise, and irregularly formed (i.e., gestures may be identical in information content but vary tremendously in spatial structure). We found that using visual gesturing for precision driving can be both difficult and tiring. Thus, to improve the GestureDriver's usability we are considering adding additional interface modalities (e.g., speech) to help classify and disambiguate visual gestures.

4. HAPTIC DRIVER

4.1. Overview

The most difficult aspect of remote driving, as with all teleoperation, is that the operator is separated from the point of action. As a result, he must rely on information from sensors (mediated by communication links and displays) to perceive the remote environment. Consequently, the operator often fails to understand the remote environment and makes judgement errors. This problem is most acute when precise motion is required, such as maneuvering in cluttered spaces or approaching a target.

The *HapticDriver* addresses this problem by providing force feedback to the operator. Range sensor information is transformed to spatial forces using a linear model and then displayed to the operator using a large-workspace haptic device. Thus, the Haptic Driver enables the operator to feel the remote environment and to better performance precise driving tasks.

4.2. System configuration

The HapticDriver system consists of a mobile robot equipped with range sensors, a graphical interface and the *Delta Haptic Device*[20]. Radio and network communication links connect the system modules, as shown in Figure 5.

Figure 5. HapticDriver architecture (left to right: mobile robot, graphical interface, Delta Haptic Device)

4.2.1. Mobile robot

We designed the HapticDriver to teleoperate a Koala mobile robot. The Koala is a small, six-wheeled, skid-steered vehicle manufactured by K-Team SA and is shown in Figure 6. Low-level motion and hardware control is performed by an on-board microprocessor (16 MHz Motorola 68331). The Koala is equipped with a ring of 16 infrared proximity/ambient light sensors, wheel encoders and a forward-looking CCD camera. The IR sensors provide range measurements (1.5 to 10 cm) to nearby objects. High-frequency (1.2 and 2.4 GHz) analog transmitters are used for wireless video and full-duplex data communication.

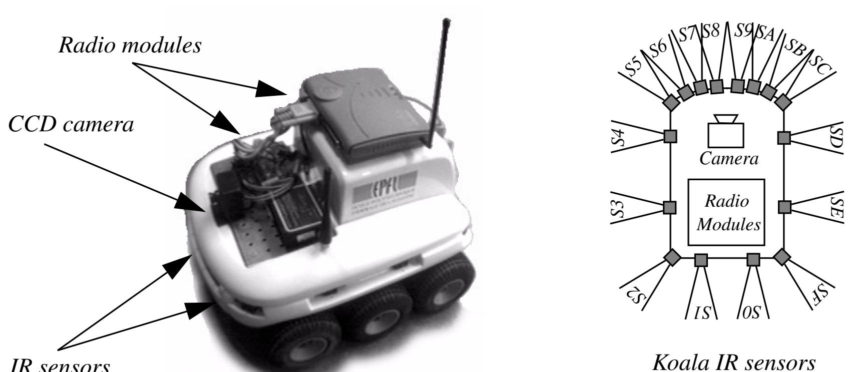

Figure 6. Koala mobile robot

4.2.2. Graphical interface

A graphical interface running under WindowsNT displays video images and range sensor information to the operator at a rate of 10Hz. The interface is shown in Figure 7 below. The interface also transforms the range data into forces, which are then sent to the haptic device. Details of the force computation method are presented below.

Figure 7. HapticDriver user interface

4.2.3. Delta Haptic Device

The Delta Haptic Device, shown in Figure 8 is a high-performance haptic device developed at the Ecole Polytechnique Fédérale de Lausanne (Swiss Federal Institute of Technology)[20]. The device is based on the Delta manipulator[21] and has 6 degrees-of-freedom: 3 translations from the parallel Delta structure and 3 rotations from a wrist module (actively controlled or passively driven) mounted on the Delta end-plate.

Unlike other haptic mechanisms (which have either limited force capability or small workspace), the Delta Haptic Device is capable of providing large forces (up to 25N) throughout a large cylindrical working volume (30 cm diameter, 30 cm length). In addition, because of it's design and it's base mounted actuators, the device has high stiffness, decoupled translation and rotation, and very low inertia. These characteristics allow the Delta Haptic Device display high-fidelity, high-quality kinesthetic and tactile information.

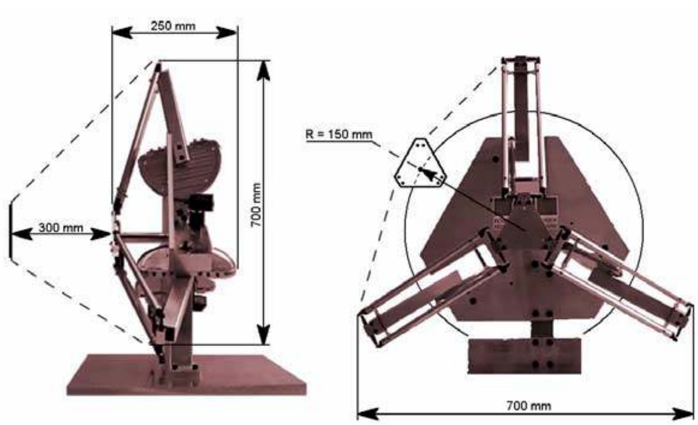

Figure 8. Delta Haptic Device

The Delta Haptic Device is driven by a real-time controller (a 500 MHz Pentium II running the RealLink* operating system) and supporting electronics. The controller executes at 1 KHz and functions as a server, exchanging messages with networked client applications via UDP/IP. Client messages describe three-dimensional force structures built from primitive elements (spheres, cubes, cylinders, planes, line attractors, point attractors, and polygonal solids). Each primitive is defined by its dimensions, pose and material hardness.

*RealLink is a trademark of RealTech AG

4.3. Force computation

With the HapticDriver, the Delta Haptic Device is used both as a control input and as a force display. For control, the 2D displacement of the haptic device from it's origin specifies the robot's translation and rotation rates. In other words, the haptic device functions as a 2D joystick: forward/backward controls translation, left/right controls rotation.

For force display, we compute force feedback in two-steps. First, we assume that robot operates purely in a 2D plane (we felt that this was a reasonable assumption given the Koala's performance characteristics). This means that the operator should only be able to move the haptic device on the control plane. To do this, we use a plane attractor (which applies forces to constrain the haptic end-point to a plane) with a point attractor (see Figure 9). The point attractor is used so that the operator is able to feel the origin of the haptic device (i.e., this serves as a virtual detent).

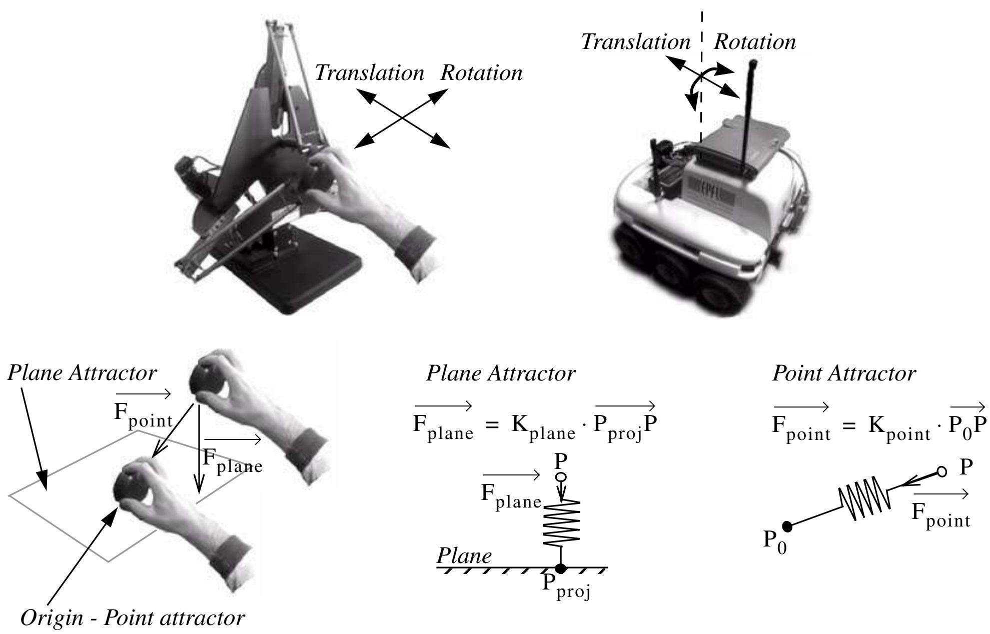

Figure 9. Force computation: control plane

Second, we model the range information returned from each of the Koala's infrared sensors by a force cube primitive, as shown in Figure 10. We compute the hardness of each cube using a linear function of range (the scaling factor was experimentally determined). Then, whenever the haptic device's end-point intersects one of these cubes, a repulsive force (proportional to the penetration distance) is generated. Consequently, whenever the robot approaches an obstacle, the operator feels an increasing force: the closer to the object, the stronger the opposing force.

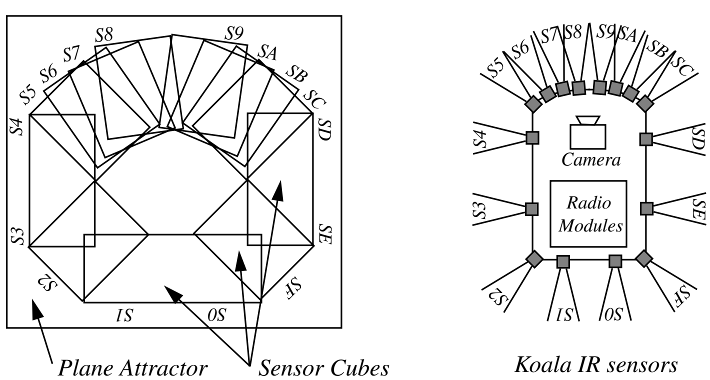

Figure 10. Force computation: range data transformation

4.4. Results

The HapticDriver is an effective interface for navigating cluttered environments and for performing docking maneuvers. Although, the interface does not completely prevent the operator from driving into an obstacle, we found that it greatly improves obstacle detection and avoidance.

We informally tested the HapticDriver at a consumer trade show by having attendees (i.e., untrained, novice operators) teleoperate the Koala through a maze. The operators were instructed to attempt the task with and without haptic feedback. Camera video and range sensor information was always available through the graphical interface. We found that with haptic feedback, almost all operators were able to precisely drive through the maze. Without haptic feedback, many of the operators were unable to complete the task and had numerous collisions.

We also performed a “drive-by-feel” evaluation in which the robot’s camera was disconnected. We found that the limited number of range sensors did not provide sufficient coverage of the remote environment under all conditions. However, operators were still able to successfully achieve limited tasks such as parking the robot in a cul-de-sac.

A significant limitation of the HapticDriver is that it only provides 2D force information. While this is sufficient for obstacle detection, it does not allow the operator to sense other environment characteristics such as terrain roughness or “viscosity”. Thus, we are considering mapping other robot sensor data (wheel torques, 3D orientation, accelerations) into 3D forces. This would enable the operator to perceive when the robot is driving across an uneven surface, through bushes, etc.

5. PDA DRIVER

5.1. Overview

For some remote driving applications, installing operator stations with multiple displays, bulky control devices and high-bandwidth/low-latency communication links is infeasible (or even impossible) due to monetary, technical or environmental constraints. For other applications, the vehicle is driven by a range of operators having diverse backgrounds and for whom extensive training is impractical. In these situations, we need a remote driving system which requires minimal infrastructure, which can function with poor communications, and which does not tightly couple performance to training.

PdaDriver is a Personal Digital Assistant (PDA) interface for vehicle teleoperation and is shown in Figure 11. PdaDriver is easy-to-deploy, is designed for low-bandwidth/high-latency communications links, and is easy-to-use. PdaDriver uses multiple control modes, sensor fusion displays, and safeguarded teleoperation to make remote driving fast and efficient. PdaDriver is intended to enable any user (novice and expert alike) to teleoperate a mobile robot from any location and at any time.

Figure 11. PdaDriver: user interface (left), remote driving a mobile robot (right)

5.2. Architecture

The PdaDriver architecture is shown in Figure 12 and contains three basic elements: the user interface, a mobile robot, and a safeguarded teleoperation controller. The user interface is described in greater detail in the following section.

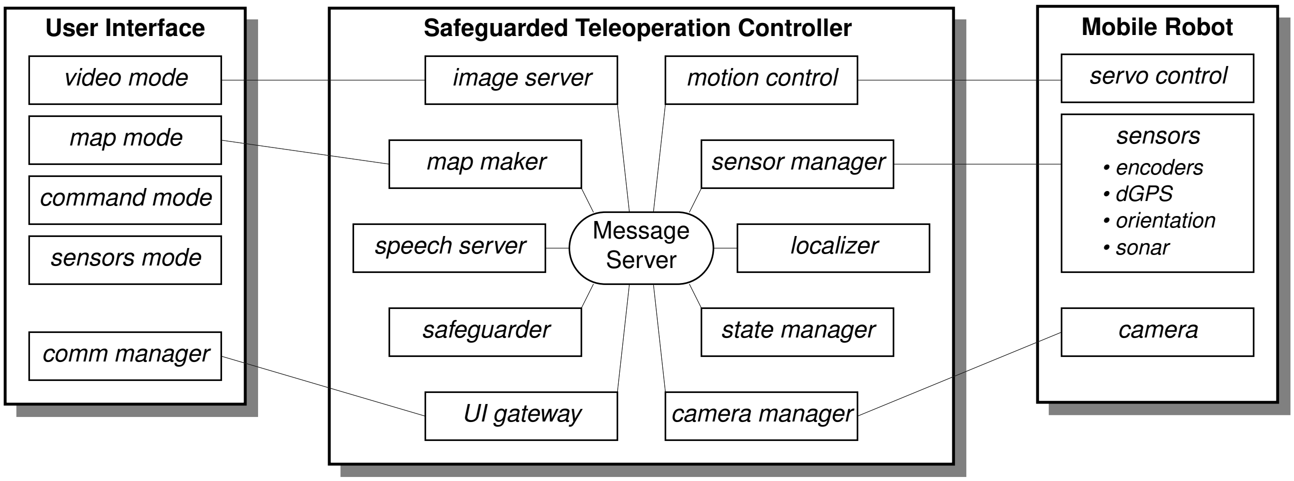

Figure 12. PdaDriver system architecture

We currently use the PdaDriver with the Tarbot ("target robot"), shown in Figure 14. The Tarbot is a Pioneer2-AT* mobile robot equipped with on-board computing (233 MHz Pentium MMX), a variety of sensors (a pan/tilt/zoom color CCD camera, wheel encoders, dGPS, 3-axis orientation module, an ultrasonic sonar ring), wireless ethernet and a pivoting training target.

We have developed a safeguarded teleoperation controller which runs on the Tarbot. The controller uses a network messaging layer to connect a variety of system modules including sensor management, map making (histogram-based occupancy grid), obstacle detection and avoidance, motion control, and localization. The controller processes user interface commands (e.g., waypoint-based trajectories) and outputs robot state (pose, health, status, etc.) at 10 Hz.

5.3. User interface

We designed the PdaDriver user interface to minimize the need for training, to enable rapid command generation, and to improve situational awareness. We implemented the interface using PersonalJava† and run it on a Casio Cassiopeia E-105 Palm-size PC. The interface currently has four primary interaction modes: video, map, command, and sensors (see Figure 13).

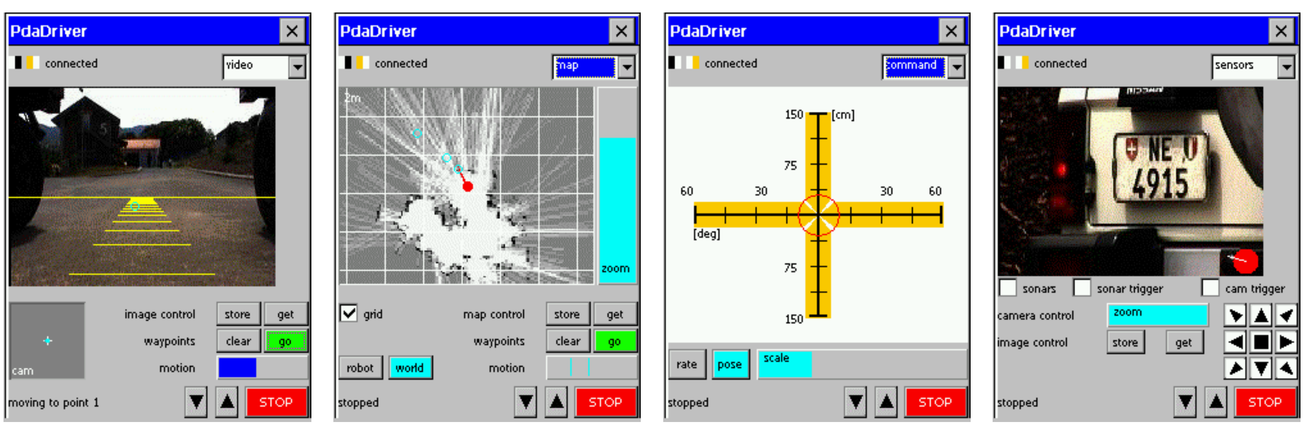

Figure 13. PdaDriver user interface modes (left to right: video, map, command, sensors)

*Pioneer2 is a trademark of ActivMedia Robotics, LLC.
†PersonalJava is a trademark of Sun Microsystems, Inc. and provides a subset of the Java 1.1 API.

Video mode displays images from the robot-mounted camera. Images are retrieved using an event-driven model to minimize bandwidth usage[1]. 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. As the robot moves from point to point, the motion status bar displays the robot's progress. This image-based waypoint driving method was inspired by STRIPE[22].

Map mode displays a map (a histogram-based occupancy grid constructed with sonar range data) registered to either robot (local) or global (world) coordinates. As in video mode, the user drives the robot by clicking a series of waypoints and then pressing the go button. As the robot moves, the motion status bar displays the robot's progress.

Command mode provides direct control (relative position or rate) of robot translation and rotation. The user commands translation by clicking on the vertical axis. Similarly, the user command rotation by clicking on the horizontal axis. A scale bar (located to the right of the pose button) is used to change command magnitude. The centered circle indicates the size of the robot and is scaled appropriately.

Sensors mode provides direct control of the robot's on-board sensors. The user is able to directly command the robot's camera (pan, tilt, zoom), enable/disable sonars and to activate movement detection triggers.

5.4. Results

We conducted a number of field tests with the PdaDriver and the Tarbot. During the tests, we used the PdaDriver to remotely drive the Tarbot in a variety of environments ranging from structured outdoor (paved roads in an urban setting), unstructured outdoor (off-road benign terrain), and uncluttered indoor. Figure 14 shows the results of an indoor test.

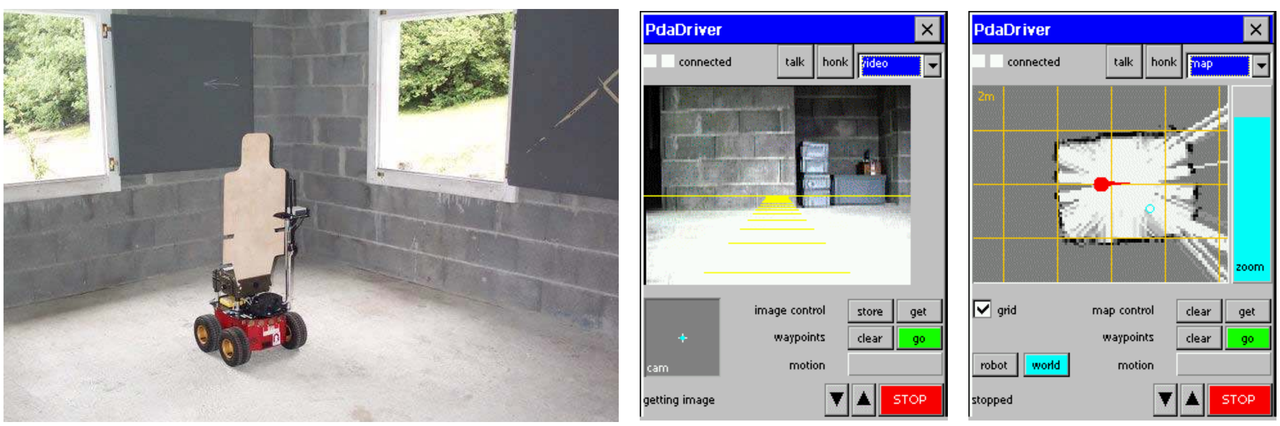

Figure 14. Indoor remote driving with the PdaDriver
Tarbot (left), video mode (center), map mode (right)

Anecdotal evidence from a range of operators (novices to experts) suggests that the PdaDriver has high usability, robustness, and performance. Since remote driving is performed in a safeguarded, semi-autonomous manner, continuous operator attention is not required and the robot moves as fast as it deems safe. Users reported that the PdaDriver interface enabled them to have good situational awareness (being able to rapidly switch between image and map displays was judged invaluable), to quickly generate motion commands (waypoint trajectories are very efficient for both short and long range motion), and to understand at a glance what the robot was doing.

We feel, however, that the PdaDriver can be improved in two specific ways. First, we need to make it easier for the user to understand the remote environment, to better identify obstacles and areas to avoid. To do this, we plan to combine information from multiple, complementary sensors and data sources to create *sensor fusion displays*[2]. Second, for semi-autonomous remote driving to be effective, we must have true dialogue between the operator and the robot: the human should be able to express intent and interpret what the robot has done, the robot should be able to provide contextual information and to ask the human for help when needed. Our approach will be to add *collaborative control*, a teleoperation model in which humans and robots work as peers to perform tasks[23].

6. CONCLUSION

We have developed three novel user interfaces for remote driving. GestureDriver uses visual gesture recognition for operator-adaptive vehicle control. The interface is extremely flexible and allows different gestures to be used, depending on the operator's preferences and the task to be performed. HapticDriver facilitates precision driving via haptic feedback. It improves perception of the remote environment, making it easier to better perform tasks such as docking and maneuvering in cluttered spaces. PdaDriver enables remote driving anywhere and anytime using a Palm-size computer and low-bandwidth communications. The interface requires minimal infrastructure, provides efficient command generation tools, and requires minimal training.

ACKNOWLEDGEMENTS

We would like to thank Microsoft Research for generously contributing WindowsCE hardware and software. This work was partially supported by the DARPA ITO Mobile Autonomous Robot Software program (SAIC/CMU task) and the Swiss Commission for Technology and Innovation (CTI project 4327.1).

REFERENCES

1. S. Grange, T. Fong, and C. Baur, "Effective vehicle teleoperation on the World Wide Web", *IEEE International Conference on Robotics and Automation*, San Francisco, CA, 2000.

2. T. Fong, C. Thorpe, and C. Baur, "Advanced interfaces for vehicle teleoperation: collaborative control, sensor fusion displays, and web-based tools", *Autonomous Robots*, July 2001.

3. D. McGovern, "Experiences and results in teleoperation of land vehicles", *Technical Report SAND90-0299*, Sandia National Laboratories, Albuquerque, NM, 1990

4. T. Fong and C. Thorpe, "Vehicle teleoperation interfaces", *Autonomous Robots*, July 2001.

5. L. Emering, R. Boulic, S. Balcisoy, and D. Thalmann, "Real-time interactions with virtual agents driven by human action identification", *First ACM Conference on Autonomous Agents'97*, Marina Del Rey, 1997.

6. C. Wren, et. al, "Pfinder: real-time tracking of the human body", *IEEE Trans. Pat. Anal. and Mach. Intelligence*, July 1997.

7. T. Starner and A. Pentland, "Real-time American Sign Language recognition from video using hidden markov models", *International Symposium on Computer Vision*, Coral Gables, FL, USA, 1995.

8. L. Conway and C. Cohen, "Video mirroring and iconic gestures: enhancing basic videophones to provide visual coaching and visual control", *IEEE Transactions on Consumer Electronics*, May 1998.

9. J. Machate, E. Bekiaris, and S. Nikoalaou, "Control it by gestures: results from a Wizard of Oz experiment in a smart home environment", *5th European Conference for the Advancement of Assistive Technology*, Düsseldorf, Germany, 1999.

10. W. Barfield and T. Furness, *Virtual Environments and Advanced Interface Design*, Oxford University Press, Oxford, 1995.

11. K. Shimoga, "Survey of perceptual feedback issues in dexterous telemanipulation: Part I. finger force feedback", *IEEE VRAIS-93*, Seattle, WA, 1993.

12. A. Koivo and D. Repperger, "Effects of force feedback on operator's skills in tele-operated systems", *American Control Conference*, 1998.

13. T. Massie and J. Salisbury, "The PHANTOM haptic interface: a device for probing virtual objects", *ASME Winter Annual Meeting: Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*, Chicago, IL, 1994.

14. M. Kurze, “Rendering drawings for interactive haptic perception”, *Computer Human Interaction,* Atlanta, GA, 1997.

15. M. Bergamasco, *Haptic interfaces: the study of force and tactile feedback systems*, IEEE International Workshop on Robot and Human Communication, 1995.

16. D. Ruspini, K. Kolarov, and O. Khatib, “The haptic display of complex graphical environments”, *Compute Graphics Proceedings*, Annual Conference Series, 1997.

17. B. Myers, H. Stiel, and R. Gargiulo, “Collaboration using multiple PDA’s connected to a PC”, *ACM Conference on Computer-Supported Cooperative Work*, Seattle, WA 1998.

18. D. Perzanowski, W. Adams, A. Schultz, and E. Marsh, “Towards seamless integration in a multi-modal interface”, *Workshop on Interactive Robotics and Entertainment (WIRE 2000)*, Carnegie Mellon University, AAAI Press, 2000.

19. S. Grange, *Vision-based sensor fusion for active interfaces*, Diploma report, Microengineering Department, Swiss Federal Institute of Technology (EPFL), Lausanne, Switzerland, 2000.

20. L. Flückiger, *Interface pour le pilotage et l’analyse des robots basée sur un generateur de cinematiques*, Ph. D. Thesis 1897, Swiss Federal Institute of Technology (EPFL), Lausanne, Switzerland, 1998.

21. R. Clavel, Conception d’un robot parallel rapide à quatre degrés de liberté, Ph. D. Thesis 925, Swiss Federal Institute of Technology (EPFL), Lausanne, Switzerland, 1991.

22. J. Kay and C. Thorpe, “STRIPE: low-bandwidth and high-latency teleoperation”, in M. Hebert, C. Thorpe, and A. Stentz, *Intelligent Unmanned Ground Vehicles*, Kluwer, 1997.

23. T. Fong, C. Thorpe, and C. Baur, “Collaborative control: a robot-centric model for vehicle teleoperation”, *AAAI Spring Symposium: Agents with adjustable autonomy*, Stanford, CA, 1999.