REMOTE CONTROL OF MOBILE ROBOTS IN LOW BANDWIDTH ENVIRONMENTS

Markus Sauer, Florian Zeiger, Frauke Driewer, Klaus Schilling
Informatics VII: Robotics and Telematics, Julius-Maximilians-Universität Würzburg,
Am Hubland, 97074 Würzburg, Germany

Keywords: Telerobotics, telelearning, teleoperation, low bandwidth connections, mixed reality.

Abstract: Tele-learning experiments with hardware require information about the working environment and the equipment status as a base. Scenarios with limited bandwidth are of interest for mobile devices as well as for users in areas with a poor telecommunication infrastructure. While camera images provide a realistic view on the remote scene, they need a high bandwidth for quality pictures. In this context an approach to replace transmission of video images is presented. At the example application of tele-learning experiments with mobile robots, data about vehicle position and orientation are essential. This input is to be determined by external tracking systems. The preprocessed sensor information can be sent via internet link even under very low bandwidth conditions. On the students side the robot is visualized in its work space in two- or three-dimensional virtual environments depending on the performance of the used computer. The paper describes the external tracking as well as the remote interface enabling access to the experiments under different conditions and reports about experiences in using this infrastructure.

1 INTRODUCTION

Remote laboratories enable students to perform experiments with hardware equipments physically located at distant locations via internet. Telematics techniques (Halme, 2004, Schilling, Roth, 2001) offer appropriate methods for remote sensor data acquisition and tele-operation access. In tele-education precursor experiments related to web-robots (Goldberg, Siegwart, 2001) started in the mid-90ies. Nowadays they are developed into complete units for selfguided learning, including tutorials, feedback on learning progress, integrated simulation models of the experiment, and remote access to the hardware equipment (Dormido, 2001, Weinberg et al., 2003).

Advantages of tele-learning include cost reduction, better utilization and permanent availability of expensive hardware. The field of mobile robots is with respect to industrial transport robots a field of growing economic relevance, while only in recent years textbooks with a more consolidated theoretical basis emerged (Dudek, Jenkin, 2000, Siegwart, Nourbakhsh, 2004). Thus learning units including hardware experiments in this field address a growing demand and are used in this paper as an application example.

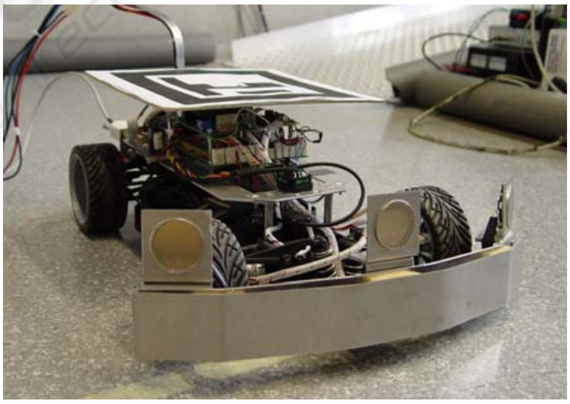

Figure 1: MERLIN Robot with marker.

The laboratory at University of Würzburg operates a remote-laboratory providing experiments controlling real mobile robot hardware (Zysko et al., 2004a / Zysko et al., 2004b). These experiments are fully integrated into the curriculum and familiarize the students with problems, which occur during the operation of real hardware instead of dealing with idealized models. In many countries, the expansion of the internet achieved a stage, where the available upload and download bandwidth, even for home connections, is high enough to run these experiments. Nevertheless, providing access to these

Sauer M., Zeiger F., Driewer F. and Schilling K. (2005).
REMOTE CONTROL OF MOBILE ROBOTS IN LOW BANDWIDTH ENVIRONMENTS.
In *Proceedings of the Second International Conference on Informatics in Control, Automation and Robotics - Robotics and Automation*, pages 163-168
DOI: 10.5220/0001184601630168

remote-laboratories in regions with a poor infrastructure or on mobile devices like PDAs or cellular phones requires an economically use of the available bandwidth. These environments do not allow the transmission of a good quality video stream due to the lack of the required connection performance.

2 SYSTEM ARCHITECTURE

The presented approach enables the user to adapt the display of experiment data according to the capabilities offered by the telecommunication link. Thus, it supports real video streams if sufficient bandwidth is available and provides a virtual experiment area if the available bandwidth is too low. The pose (position + orientation data) of the robot in its work space is determined by an external tracking system, which combines sensor data from different sources to reliable pose information with a sufficient accuracy. The use of real mobile robot hardware is fully supported and has to guarantee a real behavior of the mobile robot and the collected sensor data even for low bandwidth.

2.1 Remote Laboratory

The remote laboratory is tele-operated via a JAVA applet and communicates with the robot control server over a socket connection. The control applet provides the user all available sensor data like odometry or gyroscope angles. In addition, the user can send different experiment specific control commands for the mobile robot to the control server.

Table 1: Required bandwidth for a colored video stream

quality	downlink	uplink
very high	342 KB/s	9 KB/s
high	52 KB/s	4 KB/s
medium	29 KB/s	3 KB/s
low	16 KB/s	2 KB/s
minimum	14 KB/s	2 KB/s

Table 2: Required bandwidth for a grayscale video stream

quality	downlink	uplink
very high	248 KB/s	7 KB/s
high	47 KB/s	3 KB/s
medium	25 KB/s	2 KB/s
low	13 KB/s	2 KB/s
minimum	11 KB/s	2 KB/s

Depending on the available bandwidth, the applet can provide a video stream of the experiment area. The tables 1 and 2 show the required up and downlink bandwidth for a colored and a grayscale video with four pictures per second. If the available bandwidth is too low, a virtual experiment area can be shown in the applet, which is described later.

The remote laboratory itself has four main components: the robot control server, a camera server with camera, an external localization system, and the mobile robot.

The experiment area is a square with a side length of 3m. The localization systems provide an intelligent environment for the robot, where it can localize itself and move. They are installed in the configuration presented in Figure 2.

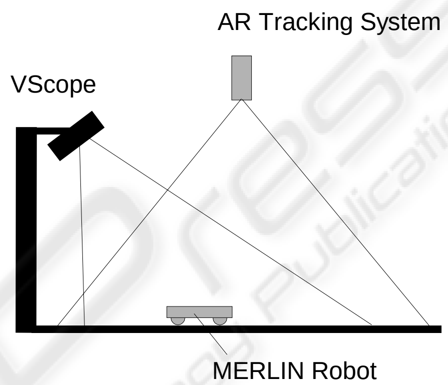

Figure 2: VScope and ARTS configuration

Using this VScope configuration, it is possible to cover almost 75% of the area. Near the borders, the VScope system has to be supported by other localization methods. If the experiment area is enlarged, two VScopes are used to provide a sufficient coverage for position determination.

The visual tracking system is mounted in the center above the experiment area and covers the complete area as the camera can be moved.

The robot control server is responsible for different activities. Besides the authentication of the remote users the server processes the sensor data from the robot and provides it to the control applet. Furthermore, control commands received from the applet are sent to the robot.

The control server includes also the module for computing the robot's. Here, the data from the different localization systems are combined in order to determine the exact position of the robot inside the experiment area.

2.2 Mobile Robot

The remote-laboratory uses the mobile robot MERLIN (Mobile Experimental Rover for Locomotion and Intelligent Navigation, cf. Figure 1). MERLIN was developed first as sensor test vehicle in the European Mars rover development.

Later, it was transferred into the educational framework (Schilling, Meng, 2002 / Schilling et al. 2003). It is a car-like mobile robot equipped with an Ackerman steering and two propelled rear wheels. MERLIN is equipped with several sensors for indoor and outdoor navigation: hall sensors, wheel encoders, gyroscope, ultrasonic sensors, VScope buttons, and a marker for visual tracking. All on board computations like sensor data acquisition and preprocessing are done with a C167 microcontroller board. The communication between the microcontroller board and the control server is done via serial port.

3 ROBOT LOCALIZATION

3.1 VScope

The VScope system is capable of tracking objects in 2D or 3D environments. It consists of three components: the VScope buttons, the VScope towers, and the VScope microcomputer.

A VScope button has an infrared receiver and an ultrasonic transmitter. Each button has a unique ID and each button's position can be determined separately. In order to track MERLINs position and orientation two buttons are needed.

The VScope towers have an infrared transmitter and an ultrasonic receiver. Starting the VScope system sends an infrared signal from each tower to the buttons. The VScope buttons will be activated by these signals and they start transmitting a synchronized ultrasonic signal. The VScope towers receive the signal of each button and send the data to the VScope microcomputer.

The VScope microcomputer processes the data sent by the towers and respectively the buttons, and calculates the absolute position of each VScope button in Cartesian coordinates. This data is provided to a PC via serial connection.

The advantage of the VScope system is its high accuracy. During experiments, an accuracy of the mobile robot position of about 2mm was achieved if both markers were in range of the VScope towers. Unfortunately, the VScope system also has some disadvantages. The infrared signal from the towers to activate the VScope buttons is heavily disturbed by sun light or even the lighting of the room. If MERLIN is too far away from the VScope towers, the buttons cannot be activated and determining the position is impossible. The second disadvantage of this localization system is the covered area. In the 3m x 3m experiment area, the VScope cannot cover the border areas due to a limited opening angle for transmitting and receiving the ultrasonic/infrared signal at the buttons and towers (cf. Figure 2).

To cope with these restrictions, a visual tracking and positioning system is installed.

3.2 Visual Tracking

The visual tracking of the MERLIN robot is realized with the help of two passive markers. One is the reference marker with known position and the other one is mounted on the robot and will be tracked. With the position and orientation estimation of these two markers the relative position and orientation of the tracked marker to the reference marker can be calculated. For the tracking of the reference marker and the tracked marker the well-known ARToolKit (Billinghurst et al., 2001) in combination with a modified version of the Java binding jARToolKit (Geiger et al., 2002) adjusted for the presented system is used. This toolkit is designed for video-based augmented reality systems. To realize an augmented reality system the six parameters for position and orientation of the camera or respectively the position of the eyes of the viewer relatively to the environment must be determined continuously. These must be done in addition to the initial calibration of the camera, which delivers the intrinsic parameters. The passive markers used by the ARToolkit have a black frame and some special patterns within this frame to identify the marker as shown in Figure 1.

The workflow of the ARToolkit can be divided into two parts. Before running the system, the markers are taught once to the system and an initial camera calibration is done. This information is used to run an ARToolkit based system. During runtime at first all black frames eligible in the camera images and the four corresponding edges for each black frame are detected with image processing methods. With the intrinsic, physical camera parameters, the defined marker size and the four detected edges of the frame, position and orientation of the marker in the world coordinate system relative to the camera capturing the images is estimated. With this estimated values the inner part of the detected marker is normalized and the resulting data is used to identify the marker.

The design of the ARToolkit for augmented reality purposes, results in a number of advantages and disadvantages. The most significant disadvantage of the ARToolkit is that the provided camera calibration tool of this library delivers quite poor results for the intrinsic parameters. For augmented reality this errors are not further important, because the secondary use of this parameters for the projection matrix of the virtual objects compensates this errors almost completely. Nevertheless, for absolute localization of markers, the results from the camera calibration tool without any adjustments are not good enough. Therefore, some mechanisms and possibilities were implemented in the presented system to adjust these parameters during runtime of the system.

The advantages leading to the decision of this system are the capability to estimate position and orientation, identify multiple markers in real-time and the easy setup of a tracking system with the ARToolkit and its corresponding passive markers.

In the presented work the ARToolkit is used with a Pan-Tilt-Zoom (PTZ) camera in combination with a TV-card from the consumer market to capture images. The pan, tilt and zoom functionality of the camera allows to cover a much larger area compared to a static camera, but results also in new tasks. At first a robust camera control module must be implemented, that the system always knows what are the pan, tilt and zoom values, when the camera is moving and when the camera has an undefined state. These pan, tilt and zoom values must be considered when position and orientation of markers are calculated.

The procedure of the AR tracking system (ARTS) has two basic states. In the init state the position and orientation matrix of the reference matrix is determined. The best position and orientation values are achieved, when the marker is in the center of the camera image, because errors in distortion correction have least influence in the center of the image. Therefore, the ARTS always moves the camera head to center the marker in the camera image.

After storing the position and orientation matrix of the reference marker, the ARTS switches to the robot tracking mode. This matrix is used to transform the position and orientation information of the robot marker to position and orientation values in the coordinate system of the experimental area. In the robot tracking mode the reference marker is no longer needed. The ARTS works with the predefined parameters completely autonomous and controls the camera head so that it automatically follows the moving robot.

3.3 Integration of Sensor Data

For determining the pose of the robot in the experimental area, three localization systems are available, which have very different properties for the generate information quality: the VScope, ARTS and the odometry calculations of the onboard microcontroller. Table 3 gives an overview of the relevant properties here.

Table 3: Properties available localization systems

	VScope	ARTS	onboard odometry
covered area	small	large	large
precision	high	mid	low (abs. values) high (small relative values)
error accumulation	no	no	yes
activation	yes	no	no

An intelligent combination of the three systems allows eliminating the specific disadvantages of each individual system. Therefore, at first all raw pose information from the localization systems are transformed into the experimental field coordinate system, with the help of default offsets depending on the physical setup of the system. Next the most probable pose of the mobile robot is calculated. This is realized by using the VScope as the reference system. As long as VScope data is available, the calculated pose of the VScope is used and stored. Additionally, the offset to the ARTS pose data is calculated and stored. As soon as VScope loses contact to the buttons on the MERLIN, the ARTS data corrected with the last stored offset are used as pose of the MERLIN. This allows compensating misalignments of the experimental setup and the offsets of the different coordinate systems, which are inevitable. On the other hand the calculated odometry information from the microcontroller can be used to generate a probability based filter selecting values from the other localization systems. The sensor data integration presented here, allows estimating the pose of the robot at a quality level accomplishing the requirements for the remote control task in the experiments.

4 REMOTE INTERFACE

The students receive access to the telelab through a Java applet. From this applet the student needs to be able to send control commands to the robot, change parameters on the robot and receive sensor data, e.g. odometry. The observation of the experimental area, i.e. the robots real movements, in real-time and good quality is also essential for satisfying performance of the experiment.

As long as the available bandwidth allows receiving video streams with sufficient quality a user interface with camera images from the scene and the numerical display of sensor data is adequate. As this high bandwidth connection cannot always be expected, the robust localization system based on sensor data fusion of VScope, ARTS and odometer is used to have a representation of the data comparable to video image.

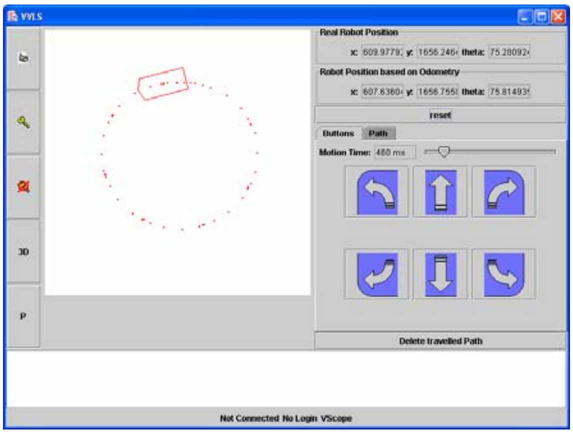

Figure 3: Applet for remote control with navigation buttons and two-dimensional map

Figure 3 shows the remote control interface consisting of all features of the former user interface (Zysko et al., 2004b) and additionally a two-dimensional map showing the position of MERLIN in the experimental area. For those, who are able to download the JAVA 3D library and have a computer able to deal with 3d calculation, a three-dimensional view on the scene is provided (cf. Figure 4). Additionally to the solution for the problem of low bandwidth connections, this mixed reality representation has several other benefits.

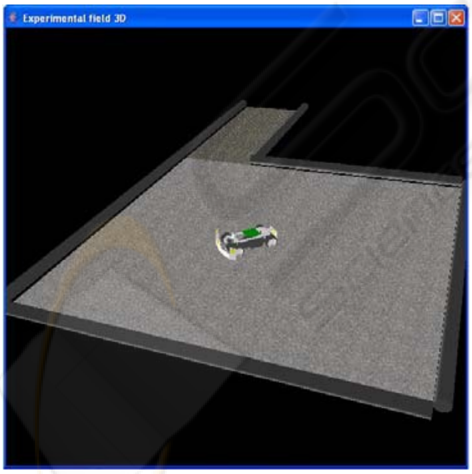

Figure 4: 3D visualization of the robot in the test field.

The camera could not cover the complete experimental area without using the pan and zoom function. If the robot was moved through the complete area, the student needed to move the camera behind the robot. This laborious task slowed down the experiment and decreased the motivation of the students. The newly implemented remote control interface based on localization data shows the complete area in 2D. Moreover, the 3D view allows the observation of the experiment from all sides (viewpoints). For the camera view the automatically following of the ARTS relieves the student from this task.

Furthermore, the virtual views cannot only represent the real pose of the robot. It can also show the data received from the onboard sensors in an intuitive way. As described in the last chapter the odometry calculations accumulate errors and are therefore inaccurate for longer distances. Understanding this kind of problems with real hardware and imperfect sensors is one goal of the experiments. The 2D view can visualize the difference between real pose and odometry based pose straightforward. The 2D view displays also the traveled path (each measured location as a dot) until the students deletes it. This feature allows better documentation of the experiment and helps the student to prepare the report, e.g. by submitting screenshots.

Moreover, the navigation of the robot, which was previously done by six buttons for the directions, can be improved. In the 2D view the student can enter a path by clicking in front of the robot.

5 EVALUATION AND TEST

The localization module of the robot control server estimates the pose of the robot. The usage of the different tracking systems within the experimental area is shown in Figure 5.

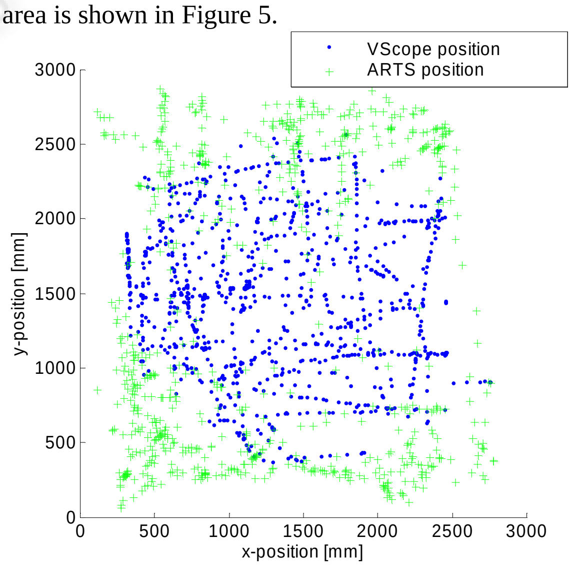

Figure 5: Tracking system usage in the experimental field

The ARTS covers the border regions of the experimental field and is used as backup when the VScope-Button activation fails, as it was planed in the system design.

Our first system for the tele-laboratory was only based on video streams as feedback from the virtual laboratory. With this system we examined the possibility to perform the remote-experiments from Tianjin University in China. The available bandwidth was about 14 Kbytes/s to 21 Kbytes/s. This bandwidth only allows a low picture quality. These tests and other performance tests showed that at least medium picture quality is necessary to provide certain usability of the experiments for the students with the video-based system. As presented in Table 2 a bandwidth of least 25 Kbytes/s (downlink) and 2 Kbytes/s (uplink) for a grayscale video stream is required.

Bandwidth tests of the system described here with the virtual representation of the experimental setup and the external tracking results in required bandwidth of about 1,3 Kbytes/s for the downlink and 0,1 Kbytes/s for the uplink. This strong reduction of required bandwidth makes it possible for users with even very low bandwidth internet connection to perform the experiments.

6 CONCLUSION

The presented work demonstrates an approach to enable tele-experiments via the internet for limited link capabilities. This offers possibilities to perform experiments with the equipment in our university for remote students from all over the world.

The described system is applied and will be further optimized in projects with Chinese and Indian universities, but also for the local students using modem connections from their homes. The flexible user interface allowing operation of the robot in two- or three-dimensional space enables the student users to choose the optimal visualization depending on the performance of their computer and internet connection.

Future work will include investigations on improved external tracking systems to cover a larger experiment area and provide a higher precision. The application potential of such telematics methods extends beyond tele-learning to industrial fields like tele-maintenance, home automation, space exploration and service robotics.

ACKNOWLEDGEMENTS

We appreciated the financial support provided for part of this research within the “EU-India” program of the European Union and the PPP China program by the German Academic Exchange Service DAAD.

REFERENCES

Billinghurst M., Kato H., Poupyrev I.,. 2001. “The MagicBook: A Transitional AR Interface”, *Computers and Graphics*, November 2001, pp. 745-753.

Dormido, S. (ed.), 2001. *Proceedings IFAC Workshop on Internet Based Control Education*, Madrid 2001, Pergamon Press

Dudek, G., Jenkin, M., 2000. *Computational Principles of Mobile Robotics*, Cambridge University Press.

Geiger Ch., Paelke V., Reimann Ch., Stöcklein J. JARToolKit - A Java Binding for ARToolKit. In *Proc. of First IEEE Workshop on AR-Toolkit*, Darmstadt, September 2002.

Goldberg, K., Siegwart R. (eds.) 2001. *Beyond Webcams: An Introduction to On-line Robots*, MIT Press 2001.

Halme, A. (ed.), Proceedings IFAC Symposium on Telematics Applications in Automation and Robotics, Helsinki 2004, Pergamon Press 2004.

Schilling K., Popescu D., Meng Q., Roth H., 2003. Mobile Roboter, In *D. Schmid, G. Gruhler, A. Fearns (eds.), eLearning - Experimente und Laborübungen zur Automatisierungstechnik über das Internet*, Verlag Europa Lehrmittel, p. 137 - 146

Schilling K., Meng Q., 2002. “The MERLIN vehicles for outdoor applications”, *In SPIE conference proc. „Unmanned Ground Vehicle Technology IV“*, Orlando 2002.

Schilling, K., Roth H. (eds.), 2001. *Telematics Applications in Automation and Robotics*, Proceedings IFAC Conference, Weingarten 2001 Pergamon / Elsevier Science.

Schilling, K., Pérez Vernet M., 2002. Field Vehicle Teleoperations Support by Virtual Reality Interfaces. In *Proceedings of 15th IFAC World Congress, Barcelona*.

Siegwart, R. Nourbakhsh, I. R., 2004. *Introduction to Autonomous Mobile Robots*, Bradford Books.

Weinberg, J. B., Yu X., 2003. Special Issue „Robotics in Education“, *IEEE Robotics & Automation Magazine, Vol. 10, No. 2 and 3.*

Zysko G., Barza R., Schilling K., 2004. Tele Lab Using Non-holonomic Car-Like Mobile Robot, *IFAC Workshop Grenoble 2004.*

Zysko G., Barza R., Schilling K., Lei Ma, Driewer F., 2004. Remote Experiments on Kinematics and Control of Mobile Robots, In *Proceedings 5th IFAC Symposium on Intelligent Autonomous Vehicles, IAV 2004*, Lisbon.