Interacting Mobile Robots on the Web

Roland Siegwart, Patrick Saucy
Swiss Federal Institute of Technology Lausanne (EPFL)
EPFL, DMT-ISR, CH-1015 Lausanne, Switzerland
roland.siegwart@epfl.ch, patrick.saucy@epfl.ch
http://dmtwww.epfl.ch/imt/asl/
http://diwww.epfl.ch/lami/

Abstract

The internet has become the most important network for communication and the biggest data storage. However, real interaction through this fast growing network is only possible through robots connected to it.

This paper discusses the approach and preliminary results of interdisciplinary projects in Internet-robotics. After a discussion of the basic issues, our modular framework for mobile robots on the Web is described. Running and future applications at EPFL, ranging from mini-robots up to experimental robot connected to the Web, are presented.

1 Introduction

The Internet connects millions of computers all over the world, giving access to communication, data, pictures, videos and even real images of distant environments. However, only a few examples of real physical interaction with distant places is available at the moment.

The goal of the described projects at EPFL is to add a new dimension to the Internet by combining network technology with the capabilities of mobile robots. This way, Internet users could discover and physically interact with places far away from their home or test their control algorithms on a real mobile robot platform.

Within different interdisciplinary projects at EPFL the following main research issues are addressed:

- *Interactive Web interfaces for mobile robots*
- *Modular Framework for Web robotics*
- Control Architectures for human guided mobile robot
- Mobile Robotics Modeling

Within this paper we mainly focus on the first two issues. After discussing the motivation of the work, our framework for mobile robots on the Web will be presented an various setups and results will be discussed.

2 Mobile Robots on Networks

Before establishing a framework for mobile robots on the Web, it might be appropriate to discuss the following two questions:

- Why should one connect mobile robots to the Web?
- How to guide mobile robots through the Web?

2.1 Why connecting *mobile* robots to the web?

Since the first robots have appeared on the Internet in 1994 [1] [2] an enormous effort has been undertaken by hundreds of research labs to push this technology. However, most of the first robots on the Web have been industrial type robots operating within a very structured and limited environment. Similarly, the first *mobile* robots on the Web have been mainly teleoperated systems [3] [4] with very limited or even without autonomy working in highly structured environments.

To get real unbound interaction with a distant environment, a mobile platform is most adequate. It has no workspace limitations and allows thus for movement to places of interest and for real exploration of distant locations. Possible applications are visits of labs [6] or museums [4] [5], sojourning exciting cities or scenic natural parks and exploring the desert, the poles or even other planets[7].

2.2 How to guide mobile robots through the Web?

Mobile robot systems connected to the Web or to any other network are facing three major problems:

- The network (Web) can introduce large time delay for which no upper bound can be guar-

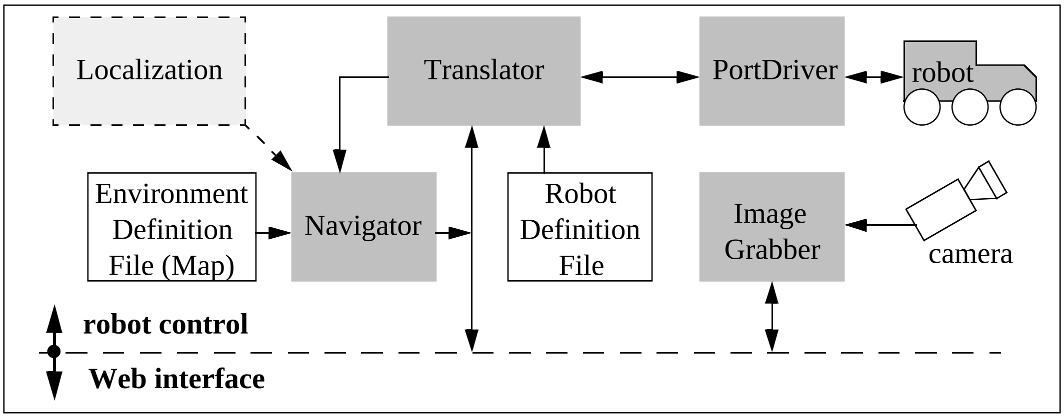

Figure 1.Schematics of the robot control framework

anteed [7].

- The network enables unexperienced peoples without any sense for technology to guide the robots.
- The Web interface has to be easy to understand and to use in order to attract as many people as possible.

Systems specifications

In consequence to the above mentioned issues, one can list the crucial points and the resulting specifications for mobile robots on the Web.

- The system should never be harmed or crash due to large time delay.
  -> the robot system requires a high degree of autonomy and a robust installation
- Internet customers should also be served appropriately during heavy network traffic.
  -> a minimal data transfer should always indicate the instant status, events and robots position. e.g. update of the robots' current position on an environment map requires only the transmission of three coordinates (x, y, θ).
- The update rate of the transmitted video images should be as fluent as possible to provide a good feeling for reality.
  -> data compression, optimal choice of resolution, focus the video feedback on *area of interest*, e.g. display only the part of a full frame image where the robot actually is.
- The Web interface should be designed for "*connect and play*". Large introduction pages will scare most of the customers.
  -> For complex environments an training phase on a *virtual reality simulation* [3] of the same setup might help.
- The control strategy of the robot should be as intuitive as possible, e.g. follow the doorway until the second intersection.

3 Modular Framework for Mobile Robots on the Web

One of the main goals of the work described here is to establish an modular framework for mobile robots on the Web. It should allow guiding mobile robots through the Internet by using a easy to use interface and simple and robust commands. It should also easily enable connecting different robots operating in various environments to the Internet. The basic idea is to build a modular programming environment with multiple processes running in parallel.

The developed programming environment enables to generate new processes and to connect them easily. This facilitates and accelerates the creation of complex systems and architectures. Changing from one robot to the other is therefore possible in a "plug and play" manner, only by changing the definition file of the robot. The approach also allows having different robots running in parallel within the same environment.

In the following sections, the two main blocks, the *robot control framework* and the *Web interface* are described.

3.1 Robot control framework

The main goal of this work is to easily connect different mobile robots to the Web. This is only possible

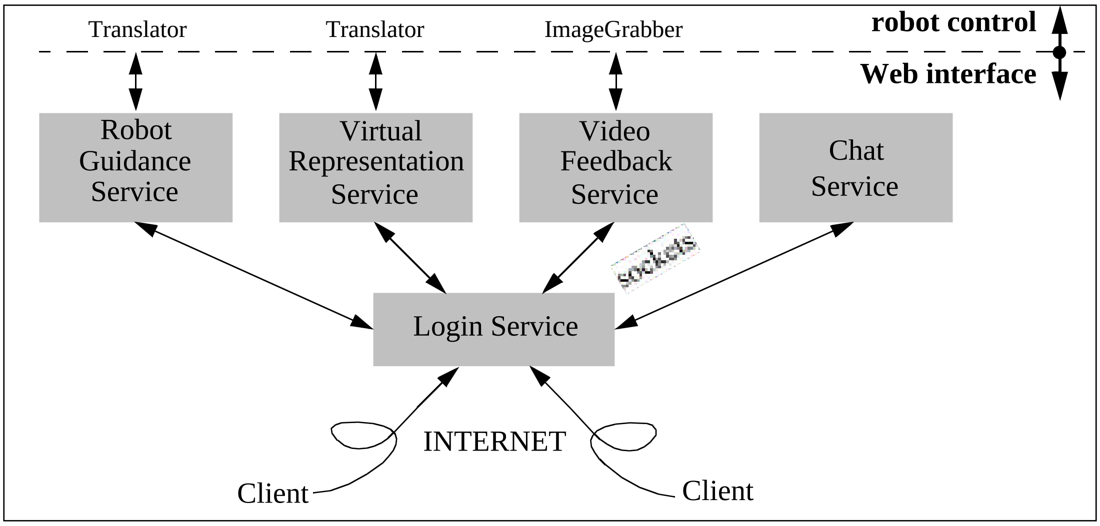

Figure 2.Schematics of the Web interface

if the main control modules are as general as possible and the adaptation to a new robot can be done by changing only very specific parts of the software. By creating two robot and environment specific files, the *robot definition file* and the *environment definition file*, most adaptations can be made by only changing these files. It works very nicely for some of the examples described in chapter 4. However, adaptations on the ***Localization*** module are still necessary and some extension of the framework have to be made for more sophisticated robot systems.

The robot control framework basically consists of four main processes (module) written in C and running on a PC with LINUX.

The ***Translation*** module translates the unified high level commands to robot and application specific low level commands. The translation is done based on a definition file which describes the robot and its capabilities.

After translation, the ***PortDriver*** gets the robot commands and sends them through a serial or parallel port to the robots.

The precisely defined standards that define the relation between the different processes (modules) allows easy exchange of processes or connections without interfering with the other processes.

The ***Navigator*** module is in charge of the path planing and navigation. Primary inputs are coming from a *environment definition file* (map file), from the *Translator* (robot definition) and from a robot localization system. Based on this information the *Navigator* generates high level control commands which will then be translated by the *Translator* module. The *Navigator* module is highly dependent on the application (see chapter 4).

The last module called ***ImageGrabber*** continuously grabs images and transmits them on request in JPEG format to the Java server.

For most navigation tasks an additional module for localization is required. As already mentioned, this module is less general then the others. It depends very much on the type of sensors used for localization and is still under development. However, for the second, third and fourth examples in chapter 4, localization is based on an external sensor (camera, laser system).

3.2 Web interface

The web interface consists of five independent modules for custom service. Each of them includes a server-side program and a client-side applet. These five services are: a *chat service*, a *video feedback service*, a *robot guidance service*, a *virtual representation service*, and a *login service*. These five services are completely independent, and written in 100% pure Java.

- The ***chat service*** allows users to send written messages to each other, private or broadcast.
- The ***video feedback service*** allows feedback from different video cameras. Users can switch from different image sources.
- The ***robot guidance service*** allows the user to

send high-level commands to a socket. This commands are then handled by the Translator described above.

- The ***virtual representation service*** draws an icon representing each robot at specific coordinates and angle. This enables to update the robot's position on an environment map.

A fifth service, the ***login service***, provides communication with the other services, and allows the system manager to get information about established connections; a log file is maintained. Other informations, such as the client's OS and browser names, are also available. The maximum simultaneous connections tolerated for each service can be specified independently (for example, one person controls a robot with video feedback, and four other persons have video feedback only).

About video feedback

The implementation of the video feedback is somewhat original and certainly effective. Where other programs use server-push systems, here the image is sent by a Java (server) program to a Java applet, via a socket, and interpreted by the applet (in either GIF or JPEG format). With this system, which allows feedback from any number of sources, we achieve 10 to 15 200x150 medium quality frames per second in a local network.

4 Examples and Results

The aim of our activities in Internet-robotics is to connect mobile robots operating in real indoor and outdoor environments. However, the first robots work in somewhat artificial environments. They serve to investigate different other aspects of mobile robots on the Web, e.g. multi robot cooperation or human-robot interaction.

A first attempt was done with KhepOnTheWeb ***(http://KhepOnTheWeb.epfl.ch)***. Its purpose is only to demonstrate some possibilities of remote control of a Khepera mobile robot. Then, another projects have been started, with the goal to add more autonomy to the robot and to allow for scientific experiments on a robot on the Web. In the following the different setups are shortly described.

4.1 First attempt, KhepOnTheWeb

In this first project, our goal was to understand the possibilities available with the current technology, the reactions of users and the requirements for the equipment of such a setup.

Our setup consists of a mobile robot Khepera accessible via the Internet and moving in a maze (fig. 3).

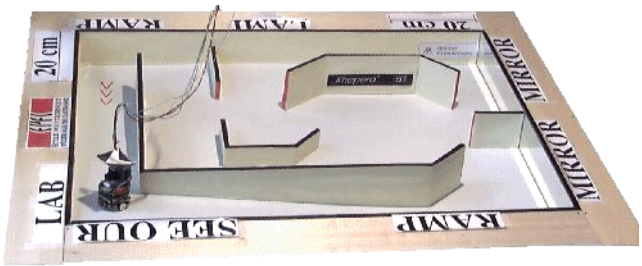

Figure 3.Khepera with its on-board video camera watching out of the maze, whose size is 65 x 90 cm

Khepera is equipped with an on-board CCD camera. An external camera is mounted on the ceiling above the maze in order to give the user a global view. The robot is purely teleoperated without an intelligence on the robot. The user can access this installation via the WEB browser Netscape. Internet Explorer^{TM} is not yet supported. This setup is accessible daily to everybody.

After one year of use we performed an analysis of the log files in order to understand the behavior of the users. People mainly visit our site once and the few who returned did it rapidly. Therefore, you have to catch the netsurfer's attention the first time he comes. It was also interesting to observe that the behavior of the users is generally independent of the country or the time of the visit, which shows that there is a global and stable Internet culture.

4.2 The TeleRoboLab

In a follow up project, our goal is to provide a benchmark to test algorithms in mobile robotics control. Thus it is necessary to provide full access to the robot and a mean to control the remote environment.

The user can register in a timetable in order to plan his experiment and have the robot available during a certain period of time. This reservation is available through the WEB for all people who have been registered as official users. This restriction is necessary because the user has the possibility to download code on the robot to control it. Through a WEB interface, the user has also the possibility to control devices (sliding door, lights) in the remote environment. The control of the robot and the video feedback can be made through Internet or ISDN depending the requirements of the user. Internet has the advantage of being widely available and ISDN of having little constant delays in the range of real time for robotics.

The setup consists of a mobile robot Koala moving in a L shape maze, see figure 4.

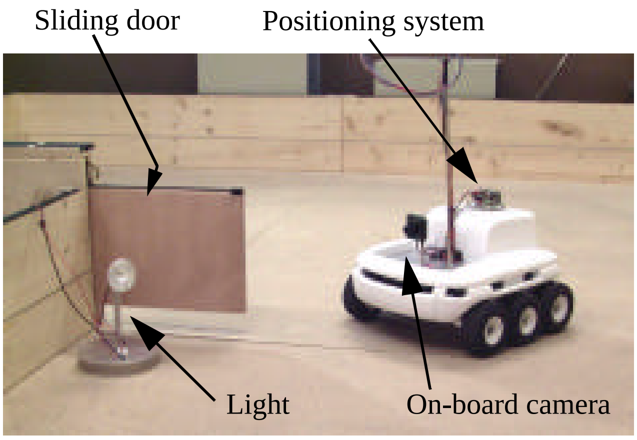

Figure 4.Koala with its on-board video camera in its dynamic environment

Like Khepera, Koala is equipped with an on-board camera. There are three other cameras, one mounted on the ceiling and two round the maze. The external Position System named KPS[1] allows the user to monitor robot trajectories or to simulate virtual events, walls or devices such as compass. A video conference system can send the selected camera view through ISDN and a frame grabber can send the image through Internet. The devices such lights and sliding door are controlled by a card named IWRE[2].

At this time the setup is built but and tests are in progress.

4.3 *Alice* on the Web

Alice (fig. 5) is a mini mobile robot which size is only about 2cm^3 [8]. It is deferentially driven by two watch motors, has on board sensors and a PIC µ-controller. Low level algorithms running on the µ-controller allow wall following, primitive obstacle avoidance and the detection of intersections. Alice has an energy autonomy of a full day, can communicate through an IR or radio link and is very cheap. It is therefore an excellent candidate to investigate a multi-robot system on the Web, allowing multiple users moving in the same environment.

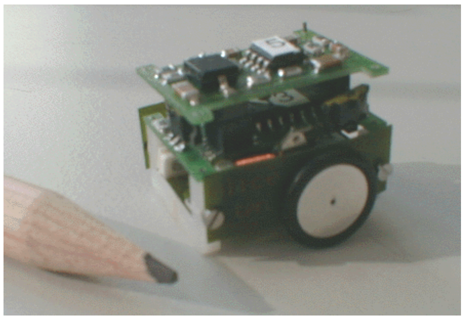

Figure 5. The sugar cube robot ***Alice****, the smallest mobile robot with real autonomy*

The AliceOnWeb setup will finally consists of 5 Alice running in the same environment. Each of the five mini robots will be operated by an different Internet user within a city-like maze (fig. 6). Three different view of cameras installed around the "playground" will be available to watch the robots movements.

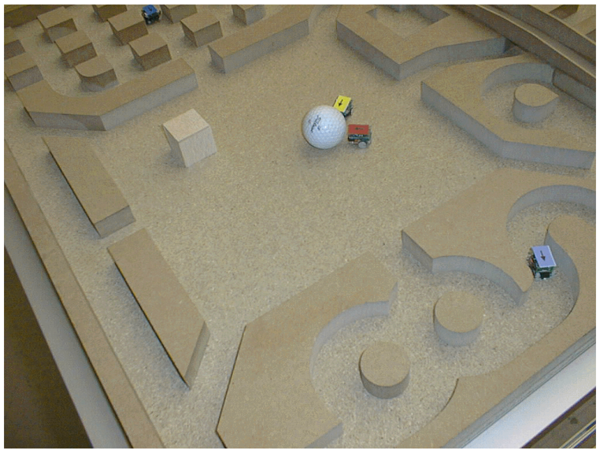

Figure 6.The city-like environment with two Alices pushing a golf ball and one Alice moving along a curved street.

For precise localization, each robot is equipped with a different color patch on its top. A camera digitizes a top view of the field, and the robots are detected by their color. The vision algorithm is capable of extracting various data about the color patches, like position, orientation and area, and makes this information available through a socket server. It can also transmit the full input image or parts of it in JPEG format, either by specifying absolute coordinates or by

1. KPS means K-Position System. It was developed for the K-family robot designed by K-Team S.A.
2. IWRE means Interact With Robot Environment.

following a robots.

To control the robots movement, the user only has to select points of interest on the map of the maze. The requested new position is then processed by the *Navigator,* which generates segments of movements executable by the robots. A typical series of segments might be: *follow left wall until first intersection, then take a left turn and follow right wall until a wall is detected in front.*

The main goal of this setup is to gain experience on multi robot control combined with multiple users. It serves also to verify the robustness of the soft- and hardware system. A preliminary setup can be found on (http://dmtwww.epfl.ch/isr/asl/projects.html). It will be permanently installed at a technical museum in summer 1999.

4.4 *Koala* on the Web

Figure 7. ***Koala***, the big brother of the famous *Khepera* robot, developed by the swiss company K-Team S.A. Its size is about 30x30 cm.

Within the second setup, which is planed to go permanently on-line also in spring 1999, a *Koala* robot can be operated through the Web in a moon-like environment. Precise localization of the robot will also be done by a overhead camera and a vision system. Obstacle avoidance and all the required low level control is installed on the on board Motorola 68332 µ-controller. This allows for safe navigation even if the vision system is covered by people around the setup.

With this setup interaction between a realistic (not real) environment and mainly with peoples at the robot's site is explored. The robot is guided by selecting points of interest on the view of the overhead camera. The robot autonomously moves to the requested position using vision feedback and obstacle avoidance. The people at the site are able to interfere with the robots task by adding and moving obstacles within the robot's environments. Speech transmission will allow the Internet user to communicate directly with the visitors at the robot's site.

4.5 *Pygmalion* on the Web

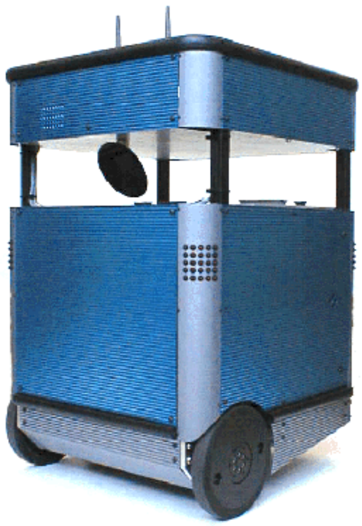

Figure 8. ***Pygmalion*** *is a sophisticated and very powerful mobile robot for indoor applications. It sizes about 40x40 cm and has a height of about 50 cm.*

Pygmalion is our high end mobile robot platform with a PowerPC, a frame grabber and a laser range finder on board. It is connected, via a radio ethernet link to the network and has its own IP number. Most of our main research in localization, navigation and map building is performed on this robot [9].

Pygmalion is our first mobile robot on the Web working in real, unchanged indoor environment. At the moment only very simple open loop commands can be controlled through the Web, e.g. move 5 m forward, turn 90°, then move 2 m backwards. In a next step a more sophisticated control interface will be established and a non-permanent installation at the computer exhibition 2000 in Lausanne is in preparation.

5 Conclusions and Outlook

Our concept and approach in Internet-robotics has been discussed. Various experimental setups have been presented. Our first applications and the feedback from Internet users allows us to gain the required experience for further work.

There are still many open problems to solve, until exploration of unknown or very complex environments through the Web becomes reality.

References

[1] K. Goldberg et al.: Desktop Teleoperation via the World Wide Web, IEEE International Conference on Robotics and Autonmation, Nagoya, Japan, 1995

[2] K. Taylor, J. Trevelyan: Australia's Telerobot on the Web, 26th International Symposium on Industrial Robots, Singapore, October 1995

[3] O. Michel, P. Saucy, and F. Mondada: KhepOnTheWeb: An Experimental Demonstrator in Telerobotics and Virtual Reality, Virtual Reality and Multimedia Conference, IEEE Computer Society Press, Switzerland, Sept. 10-12, 1997.

[4] J. A. Fryer: Remote-control experiment using a networked robot, Robotics and Machine Perception, Special Issue: Networked Robotics, Vol. 5, No. 1, March 1996, page 12.

[5] W. Burgard et al.: The Interactive Museum Tour-Guide Robot, Proc. of the 15th National Conference on Artificial Intelligence, 1998

[6] R, Simmons: Where in the world is Xavier, the robot?, Robotics and Machine Perception, Special Issue: Networked Robotics, Vol. 5, No. 1, March 1996, pp. 5-9.

[7] T. B. Sheridan: Space Teleoperation Through Time Delay: Review and Prognosis, IEEE Trans. Robotics and Automation, Vol. 9, No. 5, pp. 592-606, 1993.

[8] G. Caprari et al.: The Autonomous Micro Robot ALICE: A platform for Scientific and Commercial Applications, Proc. of MHS'98, Nagoya, Japan, November 25-28, 1998

[9] K. Arras, R. Siegwart: Feature Extraction and Scene Interpretation for Map-Based Navigation and Map Building, Proc. SPIE's Conf., Intelligent Systems & Advanced Manufacturing, Pittsburgh, USA, October 13-17, 1997