International Journal of Assembly Automation, Vol. 21, No. 2

Internet-based Robotic Systems for Teleoperation

Huosheng Hu, Lixiang Yu, Pui Wo Tsui, Quan Zhou

Department of Computer Science, University of Essex,
Wivenhoe Park, Colchester CO4 3SQ, UK.
Email: {hhu, lyut, pwtsui, qzhou}@essex.ac.uk

Abstract

Today's Internet technology provides a convenient way for us to develop an integrated network environment for the diversified applications of different robotic systems. To be successful in real-world applications, Internet-based robots require a high degree of autonomy and local intelligence to deal with the restricted bandwidth and arbitrary transmission delay of the Internet. This paper describes the first step toward building such an Internet-based robotic system for teleoperation in the University of Essex. The system has a standard network protocol and an interactive human-machine interface. Using a Web browser, a remote operator can control the mobile robot to navigate in our laboratory with visual feedback and a simulated environment map via the Internet. The employment of an intuitive user interface enables Internet users to control the mobile robot and implement useful tasks remotely. Although at its first stage, the developed system has the potential to be extended to many real-world applications such as tele-manufacturing, tele-training and tele-service.

Keywords: Telerobotics, Internet, Mobile robots, Teleoperation

I. INTRODUCTION

The operation of current robotic systems in manufacturing industry and the service sector has remained separate and independent [7]. In other words, these robotic systems are isolated from one another by different environments and have no effective way to communicate. This has made the current robotic systems expensive and requiring a long developing cycle, which has in turn seriously hampered the day-to-day deployment of robot technology. Therefore it is crucial to develop an integrated network environment for robotic systems based on today's Internet technology.

With the rapid growth of the Internet, more and more intelligent devices or systems have been embedded into it for service, security and entertainment, including distributed computer systems, surveillance cameras, telescopes, manipulators and mobile robots. Although the notion of Internet robotics or web-based robotics is relatively new and still in its infancy, it has captured the huge interest of many researchers worldwide. Except for operating in hazardous environments that are traditional telerobotic areas, Internet robotics has opened up a completely new range of real-world applications, namely tele-manufacturing, tele-training, tele-surgery, museum guide, traffic control, space exploration, disaster rescue, house cleaning, and health care.

Although the Internet provides a cheap and readily available communication channel for tele-operation, there are still many problems that need to be solved before successful real-world applications can be achieved. These problems include its restricted bandwidth and arbitrarily large transmission delay, which influence the performance of the Internet-based telerobotic systems. Therefore, it is necessary to remove human operators from the feedback control loop, and equip the robots with a high degree of local intelligence in order for them to autonomously handle the uncertainty in the real world and the arbitrary network delay. Also an intuitive user interface is required for inexperienced people to control the robot remotely. The reliability of the system should be guaranteed so that Internet users can access the Internet robotic system 24 hours a day with minimum human maintenance.

The long-term goal of our research is to carry out fundamental research on multi-agent/multi-robot cooperation and learning towards real-world applications such as tele-manufacturing, tele-training, tele-repairing, remote surveillance, and distributed service robots for office, hospital and elderly care [8][9]. Cooperative Internet robots will be a useful test-bed for us to do this. Also this unique equipment can be shared with other researchers and Internet users who are interested in working in the same area. In this paper, the analysis of some related telerobot projects is given in section 2. Our project goal is briefly outlined in section 3. The main research focus, a framework for the network telerobotic system, is described in section 4. Some experimental results are presented in section 5. Finally, section 6 presents a brief conclusion.

II. RELATED WORK

Since the first networked device "Cambridge coffeepot" appeared on the Internet, the rapid growth of the WWW over the past several years had resulted in a growing number of telerobotics sites and web accessible devices on the Internet.

The 1st generation of Internet robotic systems came into existence in 1994. The Mercury project was first carried out at that time to build a 6 DOF tele-manipulator, allowing users to pickup and manipulate various objects within its reach [4]. The telerobotic garden extended this idea to tending a garden situated around an Adept 6 DOF arm to allow users to dig and water the plants [5]. Various other devices have become available over time, such as the Bradford robotic telescope [2], the NETROLAB at Reading [12], a Web-based tele-manipulator [14], the "FortyTwo" telerobot at Manchester [13], an interactive 3D art viewing system [6], the VISIT telerobot system via computer network [10], Internet-based remote teleoperation [1], and the "MAX" wireless teleoperation [3]. This generation of Internet robots is mainly based on robotic arms or simple mobile robots that are directly controlled by human operators. In other words, a human is in the control loop. These telerobots operate within a well-structured environment with little uncertainty and have no local intelligence.

In contrast, research on the 2nd generation of Internet robots has recently begun to focus on autonomous mobile robots that navigate in a dynamic and uncertain environment, including the Xavier -- an office exploring robot at CMU [15], the Khep-on-the-Web project for open access to a mobile robot on the Internet [18], and the Museum tour-guide robot [19]. The key features of this generation of Internet robotic projects are their autonomy and reactive behaviours which enable them to navigate and cope with uncertainty in the real world. Supervisory control is the main focus in building this generation of the networked robots. However, how to handle multi-robot co-operation and remote robot programming remain the major challenges.

Internet robotics involves controlling robots or devices from a web browser remotely and differs from traditional teleoperation in several aspects:

- ☐ The delay and the throughput of the Internet are highly unpredictable, unlike traditional teleoperation where the interfaces have fixed and guaranteed delays.
- ☐ Web-based teleoperation requires a high degree of tolerance to possible data-package loss due to packet discard when there is no existing remedy.
- ☐ Internet robots need innovative mechanisms for coping with shared control among multiple web users with different applications in mind.
- ☐ Internet robots are remotely operated by many people with little expertise and few skills. In contrast, traditional tele-robots were handled by trained operators.
- ☐ Since web users are a central part of the control loop in Internet robots, their behaviour becomes an important consideration in the system design.

All of these aspects contribute to the fact that there are not yet any commercial services provided with such technology. This in turn poses many new challenges since Internet robots need to deal with a complex and dynamic environment and the unpredictable delay in network communications.

III. THE PROJECT GOAL

The aim of our research is to build a new generation of networked telerobotic systems for real-world applications such as tele-manufacturing, tele-training and tele-service. The system combines the network technology with intelligent mobile robots. Rather than using either direct control or supervised control described earlier, the developed system will use cooperative learning control to achieve fully autonomous operation and coordination. Figure 1 shows the system configuration of the proposed cooperative Internet robots in which the agent-based approach is adopted. More specifically, console agents are resident in the client site, a supervisory agent runs in the server site, and a number of coordinating agents are embedded into individual mobile robots.

The first stage of our research was focused on the realization of some of the following features:

- A uniform interface for easy integration of different robots into the system's framework.
- An intuitive user interface and adequate feedback.
- A low-cost and easily extendable system for the addition of more complex functionality.
- Cooperative behaviours to implement complex tasks that can not be implemented by single robot.
- A high degree of local intelligence to deal with the problems caused by low bandwidth and transmission delay of the Internet.

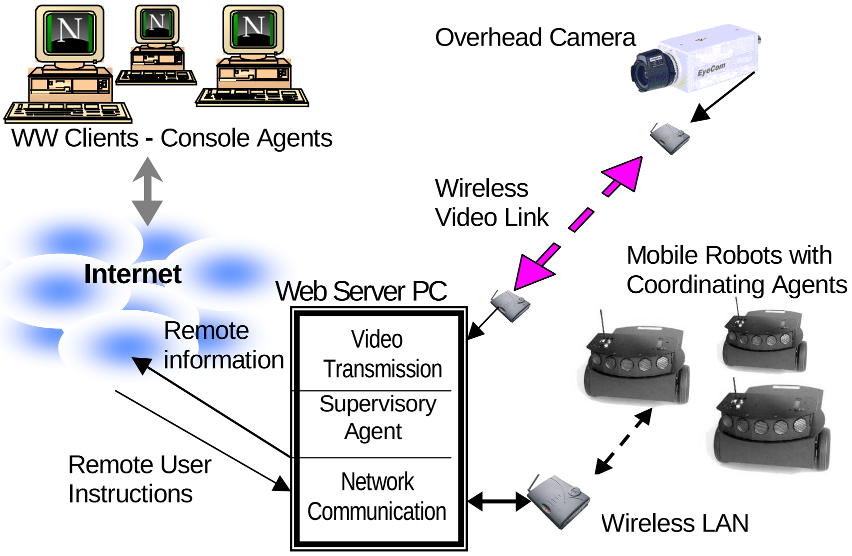


Figure 1 System Configuration of the proposed Cooperative Internet Robots

At the first stage, the main issue is to design and build a basic telerobotic system framework, i.e. a test-bed for testing theories and ideas on the tele-operated mobile robots. In this way, the Internet users, such as researchers and students, could control our mobile robots to explore our Laboratory remotely. In the rest of this paper, the focus will be placed on the construction of such testbed that will allow us to further develop the cooperative Internet robotic system described above.

IV. THE PROPOSED FRAMEWORK

With the rapid growth of the Internet there are many communications technologies available to execute requests in a networked environment. Currently the most widely used web browser is the HyperText Transfer Protocol (HTTP). It can be executed with the Communication Gateway Interface (CGI) for remote control, which is one of methods used in many web-based telerobot systems [15]. Through the

Hyper Text Makeup Language (HTML) form, a request can be passed from client to server to launch a process to perform some predetermined actions in the server. A dynamically generated HTML page will return the results to the client. But CGI has a number of limitations such as its slow response speed. Moreover, a complete HTML page must be generated with each request while the resulting page is still static. So it is not suitable for real-time remote control.

In contrast, Java provides the capability to implement network connections and thus avoid the limitations of CGI. A Java applet can operate within the browser and hence is accessible by most computers on the Internet. Rather than being static, a Java applet also enables an interface to dynamically change its content due to the fact that the Java applet is an executable within a web page. However, the Java executable must make connections to each server in the system. So the client must know the location of all servers and other clients it want to connect to, which can quickly become unmanageable as the number of servers and clients increases dramatically. There are also security restrictions associated with Java such as Java applets can only connect to the host they were served from.

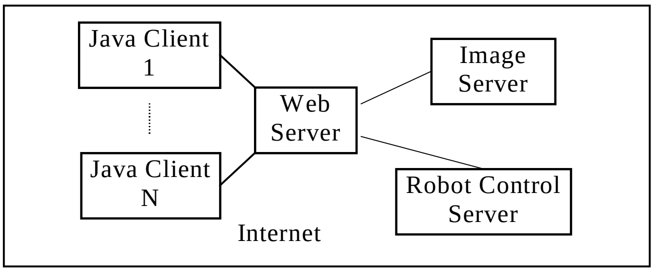


Figure 2 System Architecture

A more flexible and extendable approach is to use the central server architecture as shown in Figure 2. All the clients and servers are connected to a central web server, and only need to know the location of the web server and communicate with each other through the web server. With this architecture, we can either put all image service, robot control service and web service in one computer or put them in several computers and connect them with TCP sockets. With this architecture, it is also very easy to add more computers for robot control and image processing or for multi-robot control.

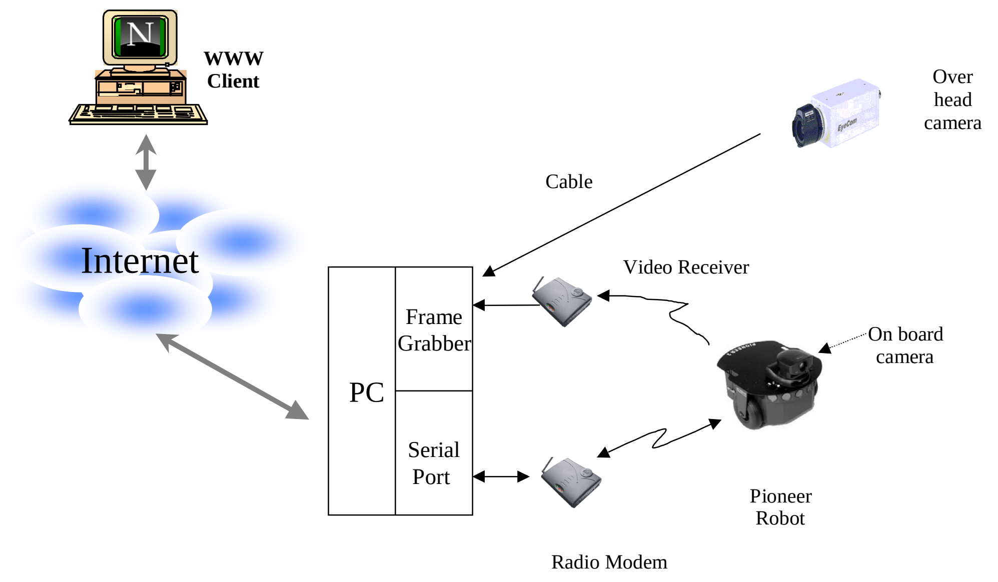


Figure 3 System Configuration of the Internet-based Robotic System at Essex

A. *Hardware*

The configuration of our current system hardware is shown in Figure 3. The host computer communicates with the mobile robot via a radio modem connected to a serial port. The video signal is captured by the frame grabber that is based on the bt848 chipset. The host computer is connected to the network with a standard Ethernet card.

The Pioneer mobile robot produced by ActivMedia is powered by two reversible DC motors coupled to two wheels with a diameter of five inches (12.5 cm) and equipped with eight ultrasonic sensors in which one is on each side and other sensors are forward facing. The data produced by these sensors is used to build a global map of the robot's environment, which is displayed at the client site. An on-board camera, connected to the server though the video transceiver, is placed on the front-top of the mobile robot in order to give the user a clear view of the environment in front of the robot. Another overhead camera is available to feedback a global view of the test site to the remote user.

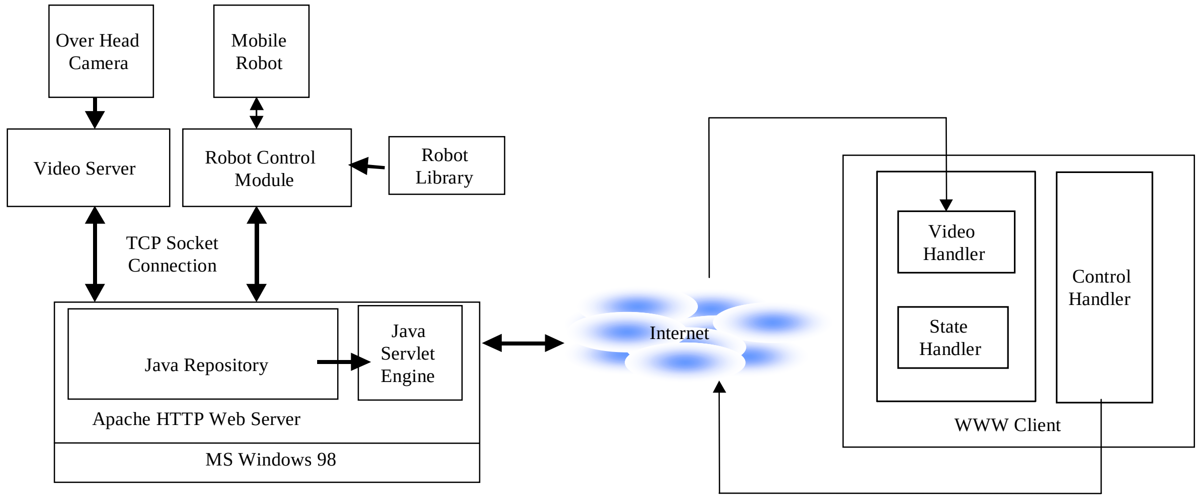


Figure 4 Software structure

B. *Software*

The web server program is Apache HTTP web server working on the Windows 98 platform. The whole system consists of several independent modules for custom service, and each of them includes a server-side program and client-side applets. These modules are *the robot control module, the visual feedback module and the virtual representation module*. The Java Servlet in the web server (Apache) handles the normal communication between the clients and the server, as shown in Figure 4.

Robot				Sonar 1				Sonar 2	
x	y	th	speed	x	y	Range	valid	...	...

Figure 5 Message structure for data feedback

C. *Robot Control Module*

The robot control module is responsible for controlling the mobile robot. Since the control program of the Pioneer robots was implemented in C++, it is necessary to build an interface to the Java program. Therefore, JNI (Java native Interface) is used to interface a DLL (Dynamic Link Library) file implemented with C++. No intelligence was integrated into the module at this stage, and only some basic motion commands such as forward, recede, change speed and direction are included, as shown in Figure 5.

The Java program will run continuously once the system starts. It receives commands sent from the client though a socket, and controls the movement of the mobile robot through the radio modem connected to the serial port. Only one user can control the robot at a time, the other users have to wait in a queue until the current operation is finished. The steps for handling the client's request are as follows:

```
Create a new byte array
Run forever
{
    Receive data from the socket
    Wait until one image frame is received
    Stored in the byte array
    Create a new image from the byte array
    Wait until the image is created
    Draw the image.
    Clear byte array buffer
}
```

At the same time, this Java program will feed back the robot information and the sonar readings to the clients every 100ms. In order to reduce the transmission time, all information were combined to form a string shown in Figure 4, and sent to all the clients connected to the server. This string will be interpreted at the client side to display an environment map and the necessary robot status.

1) Visual Feedback Module

The continuous and steady image stream feedback from the robot site is necessary when the Internet users control the mobile robot at the client site. Moreover, the image quality should be good enough to provide as much information as possible about the remote site for teleoperation. The steps are as follows:

```
Wait connection
While (request != bye)
{
   Receive request
   Check request
   Call related function in DLL to execute the
   command
}
Close connection
```

Most other projects use server push technology, where the video was made up from a stream of still images, and sent by a Java program to a Java applet via a socket, and interpreted by the applet in either GIF or JPEG format, as shown in Figure 6. In this system, the images are captured from the frame grabber based on the bt848 chipset and compressed to JPEG format by software implemented in C++. Then, these images are sent from the image server to the web server through a socket. The Java program streams these JPEG images to all the clients that are connected to this web server at a fixed interval. On the client side, the Java applet will recreate the image when it receives an entire frame and displays it.

2) Virtual representation module

The virtual representation module is the Applets that work on the web page to handle the information string sent by the robot control module. The virtual world model draws an icon representing the robot position and the direction, as well as the trajectory at the specific coordinates based on the internal odometer. An environment map can be built based on the sonar readings, and be updated every 100ms. With this virtual environment map, the user can find suspected obstacles nearby, and help to make correct decision when visual feedback suffers from serious time-delay or obstacles beyond the camera's scope.

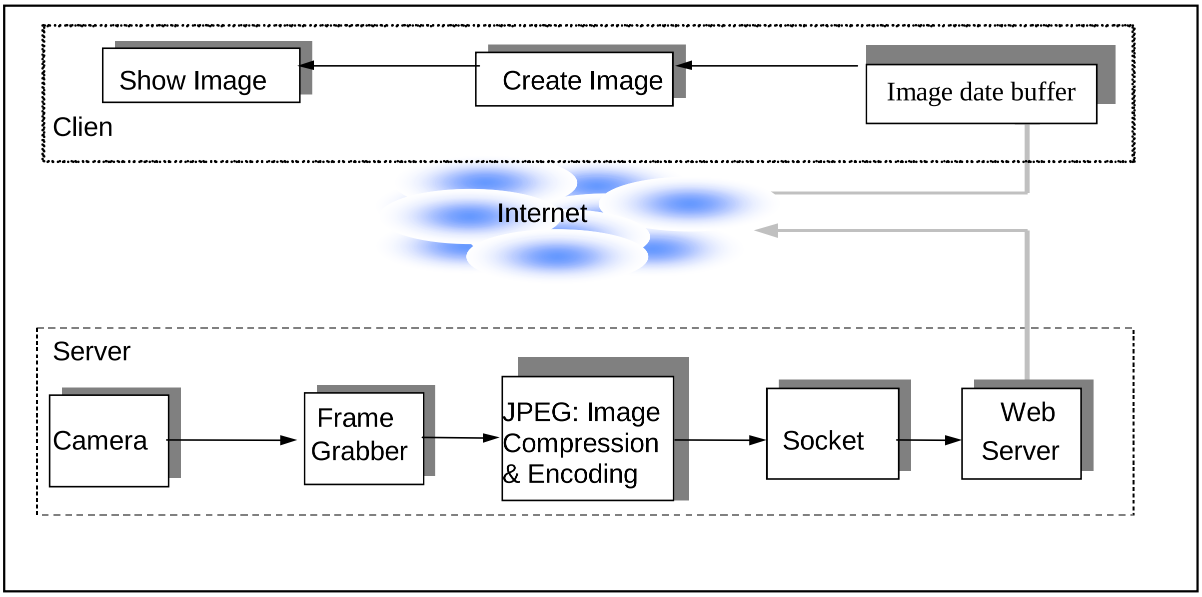
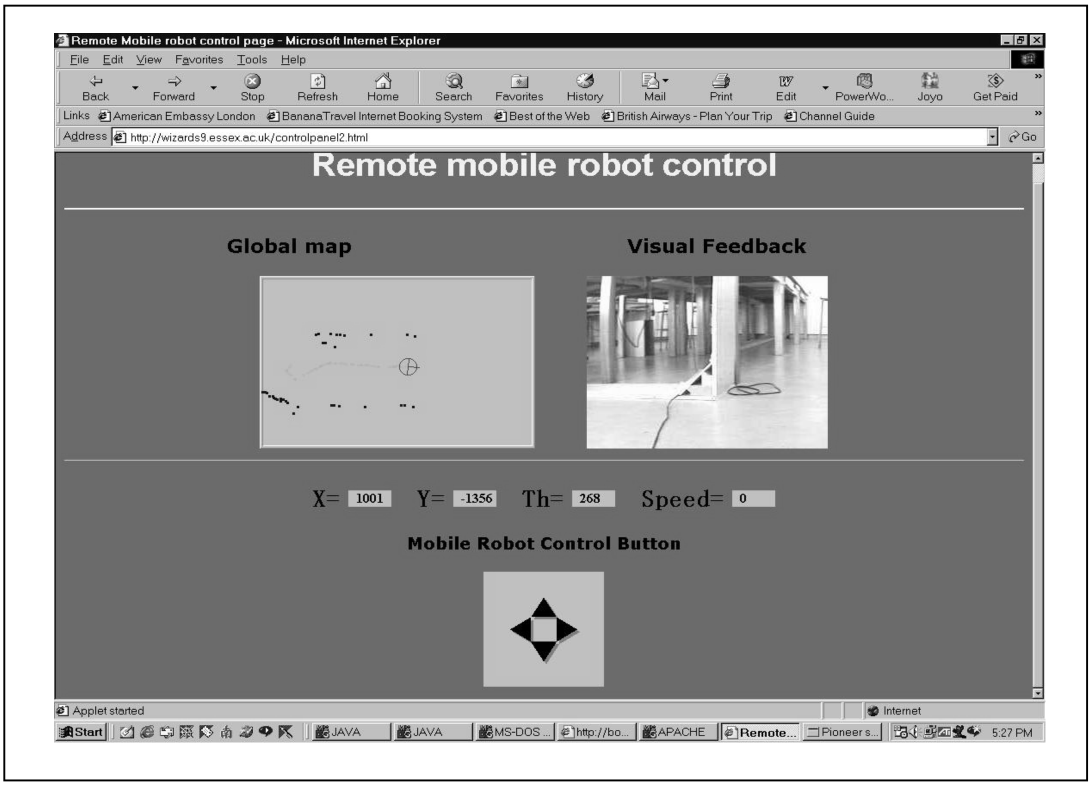


Figure 6 Video Transmission



http://wizards9.essex.ac.uk/controlpanel2.html

X= 1001 Y= -1356 Th= 268 Speed= 0

Figure 7 Web Interface

D. Web Interface

The user interface is designed with the intention of making it easy for researchers and students to interact with the mobile robot. A simple interface is designed to provide as much information as possible for teleoperation. This user interface consists of several Java Applets as shown in Figure 7. It can work on any web browser that supports Java1.2 or above. On-line instructions are supplied with this control console.

The control panel is made up of four direction buttons initially. The user can directly control the mobile robot by clicking the direction button on the control panel or by using the keyboard for fast and complex control such as change of speed or set fixed speed. The image display applet shows the visual feedback in a continuous jpeg image with 280x210 pixels at 24-bit colour depth. The virtual environment map applet displays some basic information about the mobile robot and the test site by analysing the data feedback from the mobile robot. The user can find the obstacles near the robot and the trajectory, the current position and the speed of the mobile robot. With this simple user interface, one user can control the movement of the mobile robot from the web browser with the visual feed back and a virtual representation map. The other users only have the visual feedback and a virtual map at the same time, and have to wait in queue until the first user logout at this stage.

V. EXPERIMENTS AND PRELIMINARY RESULTS

During the project development, different configurations were tested in different environments. The aim is to develop a more reliable system framework that can be used in the real world.

As shown in Figure 7, our Pioneer mobile robot was controlled to explore the laboratory we are working in, and an onboard camera was used. In another test, the pioneer mobile robot was controlled to push a ball to the goal while avoiding several static obstacles; an overhead camera was used in this test, shown as figure 8. One Pentium III 500+128M RAM PC resides at a static IP address and works as the web server and the robot control server. Another AMD-K6-300+64M RAM PC nearby runs under Windows98 and works as a video server for image processing.

Some preliminary results have been obtained from the tests. When the mobile robot moves at low speed and few obstacles block the path, the advantage of having direct control is that the user can see the result of his/her own action without any external contribution. However, the control of the robot is more difficult in a complex environment under the serious network transmission delay. Therefore, more autonomy should be put on the robot for better control and autonomous navigation at the next stage [18].

The overhead camera can give a good global view of the test site, which is very useful when the mobile robot has limited sensing capability. The users can see the movement of the mobile robot. But it also restricts the view of the robot in a small area. The on-board camera that connects with the server through the video link enables the robot to move in a quite large area and it is only restricted by the transmitter’s range. The user can see what the mobile robot sees. But the disadvantages are that the user can see the front site of the mobile robot and can not see the other sides of the robot and itself. This is often very disorienting, especially given the time-delay and the slow updating rate. A possible solution

Project name	Test environment	Technology	Image size	Speed(fps)
MAX[3]	Internet	server push	320x240	3-5
EPFL[16]	LAN	server push	200x150	10-15
HIVIEW[11]	JAENET	video conference	176x144	8
IntMedium[20]	28.8k modem	video conference	176x144	6-7
Essex	Internet	server push	200x150	7-8

Table 1 Comparison of image transmission speed

is to use both the on-board camera and the overhead camera, or a controllable on-board camera.

In the test, we achieved 9 to 12 frames per second in the local area network with time delay less than 200ms and 7-8 fps when the test was carried out from other universities in the U.K. Each frame has 200x150 pixels (medium image size). The comparison of the image transmission speed in different projects is shown in table 1. The speed ensures a quite reliable view in most experiments. Although the bigger image size and the high update rate can give more information, the network bandwidth and the performance of the computer restricted the video transmission speed.

Since all JPEG images were compressed in software, the video server can only generate about 12 JPEG frames per second although the frame grabber can capture 50 frames every second before compression. The variable network traffic also limits the packet size that can be sent in the Internet. Since transmission bandwidth is a product of the image resolution and the frame rate, there should be a mechanism to control the image resolution and the updating rate. For example, we can increase the image quality while reducing the updating rate if we need a clear view; or, we can increase the updating rate while reduce the image quality when avoiding a collision.

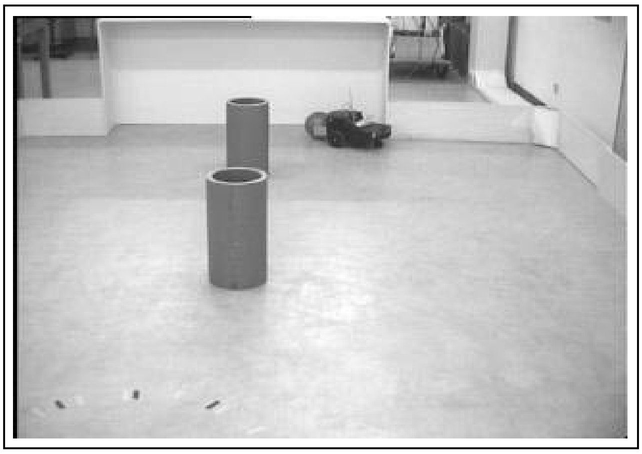


Figure 8 Pushing the ball into the goal while avoiding two obstacles

The virtual environment map is built based on the sonar readings and the internal odometer of the robot. An environment map provides much help when the mobile robot had to avoid a collision. But, due to the perceptual limitations, the sensor noise and the odometry errors accumulate over time, so it is impossible to build an accurate world model. This virtual global map can be used as a short-term environment map at this stage. At the next stage, a localization algorithm will be introduced into the system in order to build a correct environment map.

VI. CONCLUSION

We are interested in building a networked telerobotic system so that Internet users, especially researchers and students, can control the mobile robot to explore a dynamic environment remotely from their home and share this unique robotic system with us. The long-term goal of our research is towards real-world applications such as tele-manufacturing, tele-training, and tele-service.

As the first step towards a new generation of networked telerobotic systems, a modular framework for a networked telerobotics system and its web interface has been presented in this paper. The system enables Internet users to control our mobile robot from the web browser remotely. This framework has been tested in our Brooker laboratory, and the preliminary results look promising. The whole system is able to add more mobile robots and connect more video cameras. The visual feedback module written in Java allows for fast image updating, and presents a quite reliable view for the Web user.

We are currently investigating both supervisory control and cooperative learning control. The main focus is placed on how to provide a telerobot with a high degree of local intelligence to handle restricted bandwidth and transmission delay of the network and how to integrate multiple mobile robots into a telerobotics system to achieve redundancy and robustness. This will pave the way for the remote exploration of an unknown and complex environment through the Internet and other applications such as tele-training, tele-service, and tele-manufacturing.

References

[1] K. Brady and T.J. Tarn, Internet-based Remote Teleoperation, Proc. IEEE International Conference on Robotics and Automation, pages 65-70, Leuven, Belgium, 1998

[2] M. J. Cox and J.E.F. Baruch, Robotic Telescopes: An Interactive Exhibit on the World-Wide Web, Proc. 2nd International Conference of the World-Wide Web, Chicago, October 17-20th 1994

[3] A. Ferworn, R. Roque and I. Vecchia, MAX: Wireless teleoperation via the World Wide Web, Proc. IASTED Conf. on Robotics & Applications, pages 158-162, Santa Barbara, 28-30 Oct. 1999

[4] K. Goldberg, M. Mascha, S. Gentner, N. Rothenberg, C. Sutter, and J. Wiegley, Desktop tele-operation via the World Wide Web, Proc. IEEE Int. Conference on Robotics and Automation, pages 654-659, 1995

[5] K. Goldberg, J. Santarromana, G. Bekey, S. Gentner, R. Morris, J. Wiegley, and E. Berger, The Telegarden, Proceedings of ACM SIGGRAPH, pages 135-1140, 1995

[6] S. Goldberg, et al., DIGIMUSE: An interactive telerobotic system for remote viewing of 3D art objects, IROS'98: Workshop on Web Robots, pages 55-60, Victoria, Canada, 12-17 October 1998

[7] H. Hu and M. Brady, “Dynamic Global Path Planning with Uncertainty for Mobile Robots in Manufacturing,” IEEE Transactions on Robotics and Automation, Vol. 13, No. 5, pages 760-767, 1997

[8] H. Hu, D. Gu and M. Brady, A modular computing architecture for autonomous robots, International Journal of Microprocessors and Microsystems, Vol. 21, No. 6, pages 349-362, March 1998

[9] H. Hu and M. Brady, Towards advanced mobile robots for manufacturing, a Chapter in the Book ``Artificial Intelligence and mobile robots", Editors David Kortenkamp, Pete Bonasso, and Robin Murphy, AAAI Press/The MIT Press, ISBN 0-262-61137-6, March 1998

[10] K. Kosuge, J. Kikuchi and K. Takeo, VISIT: A Teleoperation system via computer Network, Proc. IROS'98: Workshop on Web Robots, pages 61-66, Victoria, Canada, 12-17 October 1998

[11] S. Levington, "HIGHVIEW: High-quality Resilint Video Over Existing Networks", Department of Computer Science, University of Essex, Colchester, England, http://cswww.essex.ac.uk/research/

[12] G.T. McKee, and Barson, R. NETROLAB: providing access to robotics technology using the Internet. Robotics and Machine Perception. Special issue: Networked Robotics. USA: SPIE, 5, 1, 1996

[13] U. Nehmzow, A. Buhlmeier, H. Durer and M. Nolte, Remote control of mobile robot via Internet, Dept. of Computer Science, University of Manchester, Technical Report Series, UMCS-96-2-3, 1996

[14] E. Paulo and J. Canny, Delivering real reality to the world wide web via telerobotics, Proceedings of IEEE International Conference on Robotics and Automation, pages 1250-1256, 1996

[15] R. Simmons, XAVIER: An autonomous mobile robots on the Web, Proceedings of IROS'98 Workshop on Web Robots, pages 43-48, Victoria, Canada, 12-17 October 1998

[16] R. Siegwart, C. Wannaz, P. Garcia and R. Blank, Guiding Mobile Robots through the Web, Proceedings of IROS'98 Workshop on Web Robots, pages 1-6, Victoria, Canada, 12-17 October 1998

[17] K. Taylor, M. Bennewitz, W. Burgard, A.B. Cremers, F. Dellaert, D. Fox, et al., MINER VA: A second-generation museum tourguide robot, Proc. Int. Conf. on Robotics and Automation, pages 1997-2085, 1999

[18] P. Saucy and F. Mondana, KhepOnTheWeb: Open access to a mobile robot on the Internet, IEEE Robotics and Automation Magazine, pages 41-47, March 2000

[19] D. Schulz, W. Burgard, D. Fox, S. Thrun, and A.B. Cremers, Web interface for mobile robots in public places, IEEE Robotics and Automation Magazine, pages 48-56, March 2000

[20] S. You, T. Wang, Q. Zhang, Internet-based Multimedia Interaction in Teleoperated Robotic System, Robotics Institute, Beijing University of Aeronautics and Astronautics, 2000