

# A Very High Level Interface to Teleoperate a Robot via Web including Augmented Reality

R. Marín, P. J. Sanz and J. S. Sánchez

**Abstract**– The system consists of a multirobot architecture that gives access to both, an educational and industrial robot, through the Internet, by using advanced multimedia and distributed programming tools like Java, Java3D and CORBA. The main effort is focused on the user interface that allows specifying a task into a 3D model and sending commands to the real robot once the operator approves them. This kind of interaction has the ability to consume very little bandwidth in order to perform a remote manipulation through the Internet. The user interface offers Augmented and Virtual Reality possibilities to the users, as well as Object Recognition in order to specify commands to the robot in a natural way (fe. Pick up the scissors). This high-level tasks specification would not be possible without the grasping determination module, which suggests manipulation characteristics for every scene's object.

**Index Terms**– Augmented Reality, Advanced User Interfaces, Object recognition, Telerobotics, Web-based technology.

## I. INTRODUCTION

Telerobotics combines a precise and fast device like a robot with the intelligence of a human being. As more facilities gives the system to the operator, by means for example of an advanced user interface, the task will be accomplished in a more simple, accurate and fast manner.

Thanks to advanced distributed programming tools like CORBA and Java it has been possible to implement such a system were the users are able to interact directly with both, an educational and an industrial robot through the Internet. Besides this, every time someone else is using the robot, there is still the possibility to interact with an Off-line virtual and augmented

This work was supported in part by the Spanish CICYT (Department of Science and Technology) under Grant No. TIC2000-1703-C03-03 and by the Spain-Germany Bilateral Agreement under Grant No. HA1999-0038.

R. Marín and P. J. Sanz are with the Department of Engineering and Computer Science, Jaume I University, Av. Vicent Sos Baynat s/n, E-12006 Castellón, Spain (e-mail: rmarin\_sanzp@uji.es).

J. S. Sánchez is with the Department of Computer Languages and Information Systems, Jaume I University, Av. Vicent Sos Baynat s/n, E-12006 Castellón, Spain (e-mail: sanchez@uji.es).

reality 3D model and then reproducing these movements on the real robot later on.

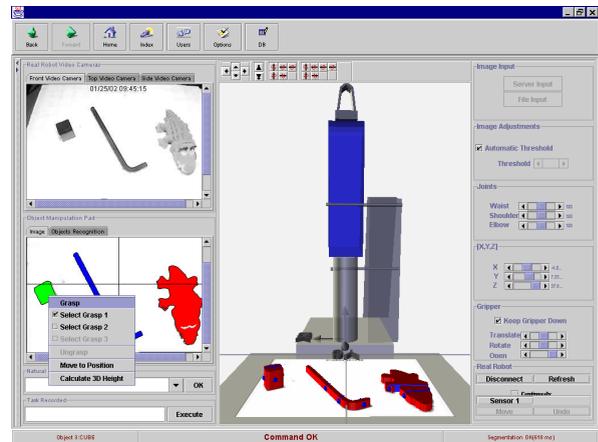


Fig. 1. Snapshot of the Virtual and Augmented Reality interface to manipulate objects on a board. Look how the system recognizes the scene objects allowing the interaction with the robot through a simplification of the natural language.

As we can see in figure 1, the 3D model allows moving the user viewpoint to any position, providing the feeling of a virtual reality environment. Moreover, as the virtual environment gives additional information to reality, by showing for example the projection of the Gripper Reference Point (GRP) over the board or by adding object recognition information to the top camera input, it means the user interface offers to the user an augmented reality environment. In some situations user has more information by seeing the virtual environment than watching directly the real robot.

The first telerobotic systems with a web-based interface were presented by researchers from the University of Southern California (USC) and the University of Western Australia (UWA). The Mercury Project [1], carried out at the USC, led to the development of a system in which the manipulator was a robotic arm equipped with a camera and a compressed air jet, and the interface consisted of a web page that could be accessed using any standard browser. The robot was used to explore and excavate a sandbox full of artifacts. The interface allowed the user to move the robot to a point in the workspace and blow a burst of compressed air into the sand directly below the camera. All robot controls were available via the mouse interaction. This

system was fully operating from August 1994 through March 1995 and was accessed by over 50.000 different computers.

The telerobotic system developed at the UWA [2] lets the user control an industrial robot to manipulate objects distributed on a table. The user interface allows an operator to specify the co-ordinates of the desired position of the arm, the opening of its gripper and other multiple parameters by filling forms and clicking images through a Java interface.

As described above, these two systems let the user send simple commands to the robot such as moving to a specific location in its workspace and performing an action with its attached tool. Very little attention has been paid to the use of more natural ways of interaction like natural languages, or even virtual reality.

We must comment that in [3] has been presented a more advanced user interface for the UWA telerobot where some information about the GRP is added to the images coming from the server in order to provide some augmented reality feeling.

In the system presented in this paper the human-robot interaction is based on both, a simplification of the natural language and mouse based operations. It means the program is able to respond commands like "Pick up the cube", thanks to an object recognition module developed through the standard CORBA [4]. The object recognition module and the capability to interact with the robot through an augmented and virtual reality environment are the major project innovations.

For those cases where someone else is using the robot, the system presents the possibility of interacting within the 3D virtual robotic environment where commands are processed in the same manner as it would be done with the real robot.

The paper is organized in four main parts. First of all a description of the Experimental Setup is presented. Secondly, the Low Level Architecture is described, giving details about the different modules implemented and their particular capabilities. Thirdly, there is a description of a single "Pick and Place" task accomplished through the Augmented and Virtual Reality interface. And finally the four point focuses on the way human-robot interaction has been improved by using two complex modules like Object Recognition and Grasping Determination.

## II. EXPERIMENTAL SETUP

In figure 2 the robot scenario in presented, where can be seen a stereo camera taking images from the top of the scene and a second camera from the side.

An interesting feature is the design of a specific circuit that allows the automatic connection of the environment lights when a remote user gets into the system.

This circuit is programmed through the server parallel port. At the moment the circuit allows a remote control over the lamps, and in a near future we will do the same with the cameras and even the robot, in order to avoid them to be always switched on.

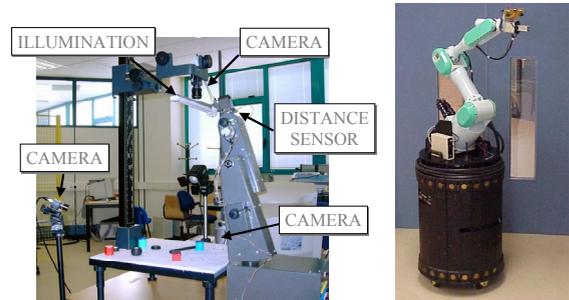


Fig 2. Experimental setup: Multirobot configuration: Vision guided Educational and industrial manipulators.

## III. LOW LEVEL ARCHITECTURE

The hardware architecture (fig. 3) shows the way client and servers machines are connected through the Internet. Note the server side is duplicated in order to allow the interaction with both, the educational and the industrial robot. Both manipulators are provided with vision capabilities and controlled through the same user interface.

For those situations where another user is having the control over the real robot, the system offers the possibility to interact with a 3D virtual environment. It means people have something interesting to do while waiting for the robot command.

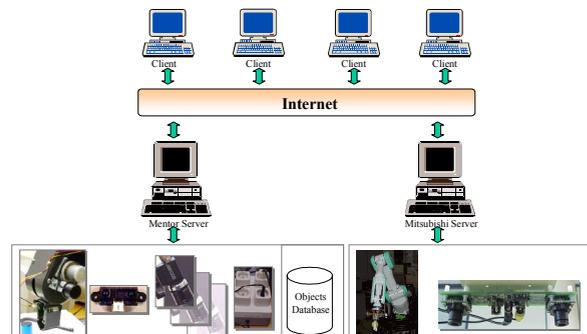


Fig 3: Telerobotic training system's hardware architecture

As we have introduced, the telerobotic system allows the manipulation of objects over a board by means of mouse interactions on the 3D virtual reality environment and also by using a simplification of the natural language. Thus, the system can respond to commands like "pick up the scissors". This kind of interaction is possible thanks to an optimized object recognition CORBA module that processes the camera images and

returns every objects name. As the program is able to learn new objects characteristics through the user interaction the system becomes more robust as time goes by. As introduced above, such a capability has no been reported into a Telerobotic system yet [1][2][4]. In figure 4 can be seen the telerobotic system's software architecture.

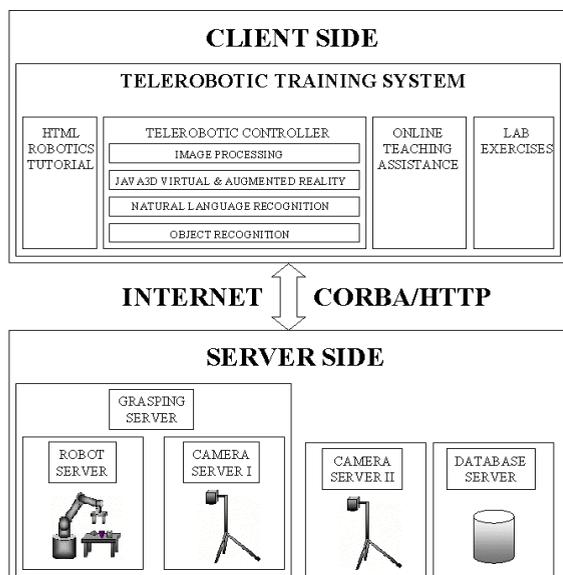


Fig 4: Telerobotic training system's software architecture

Note the software architecture is organized in several modules connected through the CORBA standard. It makes easier the integration of already existing software implemented in different programming languages and running over distinct platforms.

The system is structured in both client and server side. The client side consist of a single process implemented in Java and running through a web browser. Another possibility is installing the Java application into the client machine and running it directly from the command line. This second possibility increases a lot the client performance. The server side consists of several concurrent processes running on the server machine and interacting through the CORBA and HTTP standards.

Note the CORBA communication between the client and the server side can be tunneled through the HTTP port just in case there is a firewall avoiding the connection to different ports on the servers. Some telerobotic systems like UWA do not allow firewalled connections. In our case the situation can be managed by using the CORBA http tunneling offered by "Visibroker Gatekeeper", meaning the performance of the client/server connection goes down considerably.

In the server side there are several modules. The first one is the "Robot Server", that accepts CORBA request to move the real robot to a given world position (x,y,z)

managing directly the values for the joints, as well and controlling the opening of the gripper. The second CORBA module is called "Grasping Server", and is responsible for calculating the grasping points for every object present into the scene, assuring the grasping operation conforms to the stability requirements. The third one is called "Camera Server", and consists of a commercial product called "WebCam32" that offers a HTTP interface to the server cameras. An instance of this program must be launched for every camera connected to the system. And finally, as the system offers an object recognition capability in order to accept simplified natural language commands, it is necessary a database storing a mathematical description of every object already learned by the system. This database represents the robot knowledge, and is accessed by the multiple Java clients running over the Internet. It supposes the robot knowledge is common to the multiple users and, besides this, it is *robot independent*.

By the other hand the client side is composed of a single JAVA process that is organised in several modules. The first one is the "HTML Robotics Tutorial", that consists of a set of HTML pages that includes a well-classified description of the main topics of the Robotics subject. The second one is called "Telerobotic Controller", which offers access to the virtual and real robot using the advanced interaction explained before. The third one, "Online Teaching Assistance" allows the chat interaction between the different users connected to the Training System. And finally, the "Labs Exercises" consists of several screens that enable students to practice some advanced features like "Kinematics" or "Image Segmentation", allowing them to discover into the 3D virtual Robotic environment the consequences of their mathematical decisions.

As can be seen in figure 4 the Telerobotic Controller is divided in four submodules. The first one, "Image Processing", gives some services for capturing and segmenting images. The second, "Java 3D virtual & Augmented Reality" implements the 3D virtual environment that allows the virtual reality interaction with the robot as well as the augmented reality feature that shows graphical representation of the position of the GRP over the board as well as the superposition of objects information to the camera images. The "Natural Language Recognition" module consists of several JAVA classes that are able to interpret a simplified natural language command from the user. It translates this command into the appropriate sequence of actions inside the remote controller. And finally, the "Object Recognition" module is able to identify the different objects present into the scene in a fast and robust manner.

#### IV. VIRTUAL AND AUGMENTED REALITY

In this section the virtual and augmented reality capability is presented by means of an example of grasping an object over the board and dropping it into another place. It means the explanation is focused on the “Active Guided Programming” that can be accomplished by means of the Java module “Java 3D virtual & Augmented Reality” introduced above.

As can be seen in figure 5 the screen has 4 main parts. The first one the cameras, that give the user a continuous monitoring to the board scenario from two different viewpoints. Note a Manipulation Pad is included where user can click directly over the objects and send commands to the robot through a mouse interaction. The second the 3D virtual scenario, that monitors the real robot position and renders it over the screen. Note this capability avoids the need to include a third camera to monitor the whole robot movements. The third part are the robot movement controls, that allow the user to move the robot to a specific (x,y,z) location or even access directly the degrees associated to every join. The fourth part is the text-input field that allows the user to specify the robot tasks in a more natural manner (e.g. Pick up the cube).

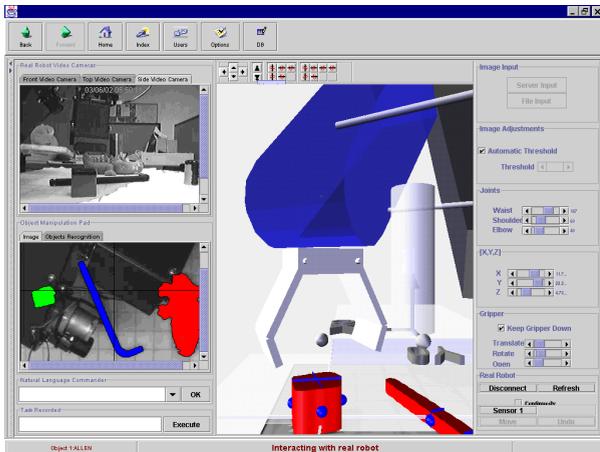


Fig 5: Locating the gripper over the object to be manipulated.

As can be seen in figure 5 in order to manipulate an object first we have to locate the gripper on top of it. To do so we can move the arm by using the controls at the right side of the screen or even easier by clicking with the mouse the appropriate object on the manipulation pad. Look at figure 5 how the real top camera input is mixed with the object recognition knowledge, providing an augmented reality information. By looking at the camera images can be seen the real position of the gripper over the object, prepared to execute the next action.

Note how useful are the projections of the gripper over the board in order to get a better understanding of

the position of the arm into the world (augmented reality). Besides this, note how interesting is the possibility to move the user view point in order to better specify a given task (virtual reality). See in figure 1 and figure 5 two different user view points of the same robot position. This feature allows the user to navigate through the robot world and interact with it by using any point of view.

As seen in figure 6 next step would be moving down the arm a little bit and then closing the gripper. The result is shown in the 3D environment as well as the real camera images.

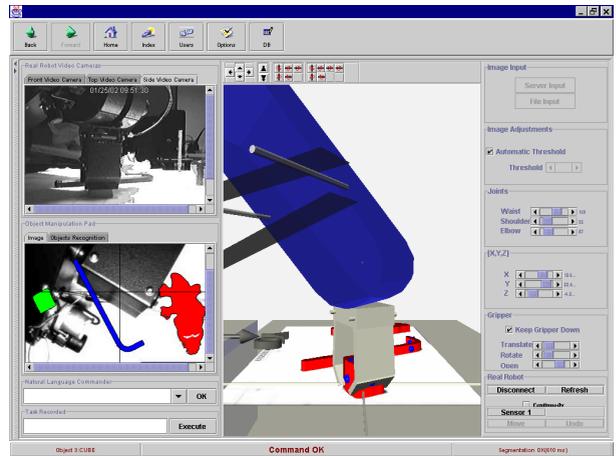


Fig 6: Grasping the object by moving down the arm and closing the gripper

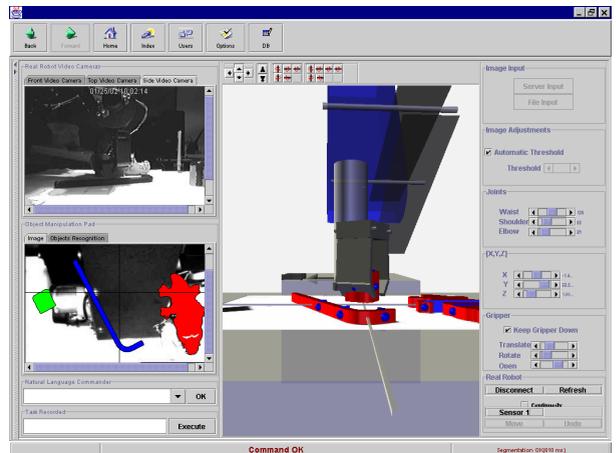


Fig 7: Bringing the object to the new location without having to move the robot to the original position in order to refresh the objects environment.

The next step can be appreciated in figure 7. It consists on moving the arm to another position in order to open the gripper and place the object over the new location. Note the importance of having a 3D model of the real objects on the scene. They allow monitoring the position of the grasped object inside the gripper

without having to move the manipulator to the original position in order to refresh the virtual environment objects representation.

## V. ADVANCED USER-ROBOT INTERACTION

To accomplish such a high level interaction between the user and the robot it has been necessary the integration of several complex modules into the system. We are going to focus on the overall description of the Grasping Determination and the Object Recognition modules.

### A. Grasping Determination

The visually guided grasping plays a crucial role in the telerobotic system. Otherwise, it would not be possible to command high level tasks like "Pick up the Screwdriver".

Every time the system evaluates the top camera input it identifies several grasping possibilities for every scene object, as well as a unique solution that will be adopted by default [5]. Thus, user can command a grasping by saying "Pick up the object 1" or even "Execute grasping 3".

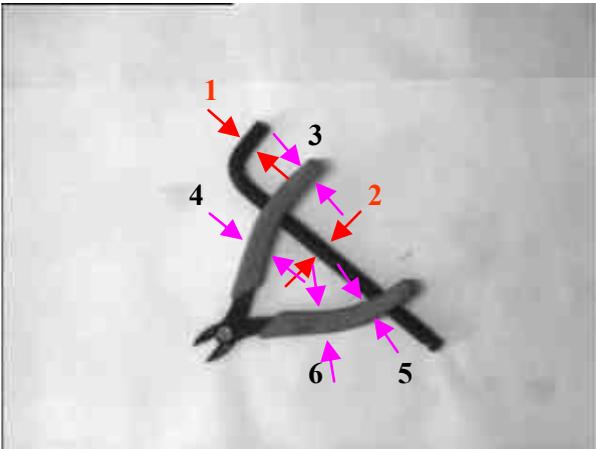


Fig 8: Different grasps obtained over a complex scene with overlapped objects.

For more complex situations where the scene's objects are overlapped (fig. 8), the system can still work out.

In these cases the automatic object identification will not be available until a previous isolation guided for the user is concluded. To make easier this procedure a number is associated to every appropriate grasp, allowing the user to execute one of them (fe. "Execute grasping 2"). This procedure permits the user to manipulate every scene's object until every one is isolated and able to be identified by the object recognition process.

Summarizing, the system incorporates the *active perception* paradigm, proposed by Bajcsy [6], which has demonstrated a very interesting application in this type of tasks.

### B. Object Recognition

For those situations where objects are isolated and able to be identified, user can request the execution of the object recognition process. This will allow the specification of high level tasks like "Pick up the screw".

As seen in figure 9 the "Image Processing" module receives the top camera input and segments it. Besides this, it obtains a mathematical representation of every scene's object that has the capability to be invariant to rotation, scale and translation [4]. We will refer to this representation as "Invariant Hu Descriptor".

Second step is applying object recognition algorithms in order to identify this mathematical representation with a set of classes "Training set" that have been already learned by the system. The output will be the class name (fe. "Screw") that matches the Invariant Hu Descriptor representation of the scene object. In [4] can be found an extended description of the different algorithms as well as a comparative that shows clearly which one of them are appropriate for the different system's applications.

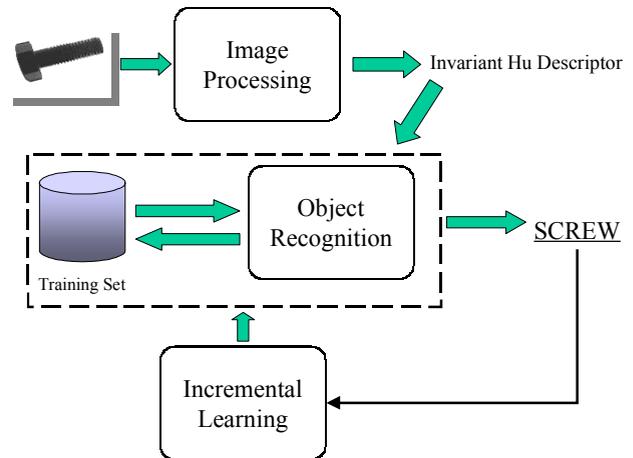


Fig 9: Object Recognition and Incremental Learning procedure

An *Incremental Learning* module has been included in the loop in order to allow the system to learn more object representations as well as refine the existing ones. This allows the user to specify a candidate class to be included into the Training Set or even to refine an existing class representation in the database. The way it works is simple. In order to protect the already existing Training Set from corrupted classes, first of all it stores the candidates samples into an auxiliary Training Set.

The system administrator is able to supervise the candidates samples in order to discard possible errors. Later on, he is able to launch the automatic incremental learning procedure that evaluates if a candidate class matches the specified class' mathematical constraints or not.

## VI. RESULTS AND CONCLUSIONS

Such an advanced user interface permits the specification of high level tasks into the client side without consuming Internet bandwidth constantly. At this point a performance table is presented showing the time consumed by the different operations of the system. Note the results have been obtained "On campus", due to the fact that this Telerobotic System is being used as part of an undergraduate Robotics course in our university [7].

TABLE I  
PERFORMANCE RESULTS

Operation	Time (sec)
Program Launching	5.06
Robot Initialization	3.32
Images acquisition	0.20
Image Segmentation and HU extraction	0.32
3D objects reconstruction	0.07
Objects Recognition	3.45
Robot movement execution	2.29

The performance results show the most time consuming operations are the "Program Launching" and the "Objects Recognition". Note in some situations the Object Identification will not be necessary. Operations like "Pick up object 2" or "Execute grasping 6" would still be available. Depending on the particular task user will decide launching the Object Recognition procedure or not.

## REFERENCES

- [1] Goldberg K. (2000): The Robot in the Garden: Telerobotics and Telepistemology in the Age of the Internet: The MIT Press Cambridge, Massachussets, London (England), 2000.
- [2] Taylor K., Dalton B, Internet Robots: A New Robotics Niche: *IEEE Robotics & Automation Magazine*, Vol. 7, No. 1: Global Remote Control Through Internet Teleoperation: U.S.A, 2000.
- [3] Behnke R., A User Interface for telecontrol of a robot over the Internet: *First IFAC Conference on Telematics Applications in Automation and Robotics TA 2001*, Weingarten, Germany, 2001
- [4] Marin R., J. S. Sanchez, Sanz, P.J., A comparative Study on Object Recognition Methods Applied to Web Telerobotic Systems. *IX Spanish Symposium on Pattern Recognition and Image Analysis*, Benicassim, Spain, 2001.
- [5] Morales A., Sanz. P.J., Heuristic Vision-Based Computation of Planar Antipodal Grasps on Unknown Objects. *IEEE Int. Conf. on Robotics & Automation*, 2001.
- [6] Bajcsy R., Active Perception and Exploratory Robotics. In "Robots and Biological systems: towards a new bionics?". Ed by Dario, Sandini & Aebischer, NATO ASI Series, Springer Verlag, pp. 3-20, 1993.
- [7] R. Marin and P. J. Sanz, "The UJI telerobotic training system", in *Proc. of the 1<sup>st</sup> Workshop on Robotics Education and Training*, Weingarten, Germany, 2001, pp. 25-30.