

Remote Programming of Network Robots Within the UJI Industrial Robotics Telelaboratory: FPGA Vision and SNRP Network Protocol

Raul Marin, Germán León, Raul Wirz, Jorge Sales, José M. Claver, Pedro J. Sanz, *Member, IEEE*, and Josep Fernández, *Member, IEEE*

Abstract—This paper presents the UJI Industrial Robotics Telelaboratory, which lets Ph.D. and Master's degree students perform robotics and computer vision tele-experiments. By using this system, students are able to program experiments remotely via the Web, in order to combine the use of a field-programmable gate array (FPGA) to provide real-time vision processing, a conveyor belt, and a Motoman industrial manipulator. This paper introduces the novel SNRP protocol (i.e., Simple Network Robot Protocol), which permits the integration of network robots and sensors within an e-learning platform in a simple and reliable manner. As long as the students are able to interact remotely with a real robotic scenario, this system helps students very much to learn robotics control techniques like visual servoing control, vision for industrial applications, and robotics manipulation. The various components of the system are connected via a 100BaseT Ethernet network and follow the SNRP protocol, which grants simple access to generic networked devices using enhanced HTTP-based connections. Moreover, the whole telelaboratory is connected to the Internet through a router that permits the user to control the networked devices according to security constraints. The SNRP architecture is compared with a Common Object Request Broker Architecture-based approach, which was used in a previous telelaboratory. This paper describes two principle contributions: the design of a novel SNRP network architecture for the intercommunication of robots and sensors within an e-learning telelaboratory and the integration of a programmable FPGA vision system, which allows students to learn not only robotic techniques but also the design of high-performance circuits for industrial vision applications.

Index Terms—Distributed systems, e-learning, high-performance field-programmable gate array (FPGA) vision, industrial robotics telelaboratory, Internet, multirobot programming.

Manuscript received March 30, 2007; revised January 8, 2009. First published February 3, 2009; current version published November 6, 2009. This work was supported in part by the Spanish Ministry (MEC) and the European Commission FEDER funds under Grants DPI2005-08203-C02-01, DPI2008-06548-C03-01, TSI2004-05165-C02-01, TIN2006-15516-C04-02, and "Consolider Ingenio-2010" CSD2006-00046, in part by the Fundació Caixa Castelló under Grants P1-1B2003-15 and P1-1A2003-10, and in part by the EU-VI Framework Program under Grant IST-045269—"GUARDIANS" of the EC Cognitive Systems initiative.

R. Marin, G. León, R. Wirz, J. Sales, and P. J. Sanz are with the Computer Engineering and Science Department, Universitat Jaume I, 12071 Castellón, Spain (e-mail: rmarin@icc.uji.es; león@icc.uji.es; wirz@icc.uji.es; salesj@icc.uji.es; sanzp@icc.uji.es).

J. M. Claver is with the Computer Science Department, University of Valencia, 46071 Valencia, Spain (e-mail: jclaver@uv.es).

J. Fernández is with the Computer Science Department, Universidad Politècnica de Cataluña, 08034 Barcelona, Spain (e-mail: josep.fernandez@upc.edu).

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TIE.2009.2013250

I. INTRODUCTION

USING ROBOTIC telelaboratories is a very interesting way to give undergraduate and graduate students the opportunity of practicing with a real robotic scenario, which can be programmed *in situ* or even from home via the Internet (i.e., open telelaboratory) [1]–[4]. Providing access to expensive hardware like an industrial robot via the Web opens the doors of the laboratory and allows students to perform their experiments 24 h a day.

During the past seven years, the authors of this paper have been offering students access to an older telelaboratory, so that they can practice robotics, vision, and control experiments. The UJI Online Robot system, nowadays, provides two educational robots that can be remotely programmed through the Internet in an easy manner [5]. This system is particularly useful for undergraduate courses because it does not present any safety risk to the students, even if they are in the laboratory when the robots are moving.

Moreover, there are some control experiments (e.g., high-performance industrial applications) that need more sophisticated robotic environments and offer the scientists and students a higher precision in the robot positioning and speed. These were some of the main concerns that influenced the design of the UJI Industrial Telelaboratory, which now allows people to remotely program complex and high-performance industrial applications via the Web (see Fig. 1).

One essential part of an Industrial Telelaboratory is the interconnection of sensors, cameras, and robots via a networked system [6]–[8]. In fact, in this paper, we consider that every device (i.e., industrial robot, conveyor belt, field-programmable gate array (FPGA), etc.) is connected to the same Ethernet network, and they act as single Networked Robots that communicate with each other through the Simple Network Robot Protocol (SNRP) Web-based protocol (see next section for details). This architecture offers many advantages like scalability and maintainability, and it introduces interesting issues like device synchronization, bandwidth, and time delays [9].

As will be explained in detail in the following sections, enabling remote programming of robotic systems permits the development of external programs that take control over the whole set of robotic functions. Thus, for example, it is possible to design an experiment in Java for performing a closed-loop manipulation (i.e., remote visual servoing) or even to use this interface for designing a voice-operated robot.



Fig. 1. UJI Industrial Telelab robotics environment (i.e., programmable FPGA-based vision system, conveyor belt, on-hand camera, and monitoring camera from the top).

A very good example of remote robot programming, which justifies the design of these networking architectures for teaching purposes, is the remote visual servoing control problem [10], [11]. It uses sequences of camera inputs in order to bring the robots to the desired position, in an iterative way. In this paper, it is explained how students and researchers were enabled to use the provided telelaboratories in order to practice robotic visually guided control through a remote real environment, instead of using simulation tools.

Finally, this paper describes the implementation of a networked vision system implemented using an FPGA. A camera mounted on the robot gripper gives the input images to the FPGA, which performs the image processing, object segmentation, and grasping determination procedures at almost 50 ft/s. This device provides excellent improvements in the remote programming of visual servoing control loops, as explained in detail in the results. The telelaboratory includes a Web interface that enables students to upload their own FPGA vision algorithms into the system.

Why is the FPGA useful for students? What can they learn from it? Having a programmable FPGA integrated into a Web-based telelaboratory also permits students to practice other disciplines like computer architectures for high-performance vision. This kind of infrastructure can only be accessible to students in this way.

Real-time robotics tasks, such as visual servoing, object tracking, and motion control, require high computational power



Fig. 2. Web-based interface to the programmable FPGA-vision system (the video on the left represents the output from the FPGA vision system; the video on the right shows the output from a monitoring camera).

and data throughput, which often exceed the processing capabilities of the processor on computer platforms. In this case, application-specific hardware (e.g., ASICs,¹ DSPs,² or FPGAs) are considered as alternatives in dealing with this problem [12]–[15]. However, the FPGA solutions can be reprogrammed by students, and these solutions allow the rapid prototyping of circuits, which is very important for the telelaboratory design [16]–[20]. In fact, the reconfigurability of an FPGA via the Web greatly improves the learning capabilities of a telelaboratory, giving the opportunity to researchers and students to experiment directly with an expensive FPGA system from any computer connected to the Internet. In Fig. 2, we can see both the output from the FPGA vision system, which can be programmed via the Web by the student (i.e., left side), and the video obtained from a network camera for monitoring purposes.

In summary, as long as the FPGA systems are being considered in the industry as an interesting way to program high-performance vision algorithms, the availability of an e-learning platform that includes an expensive programmable FPGA-based vision system is very interesting for the learning of students.

II. UJI INDUSTRIAL TELELAB

The UJI Industrial Telelab is composed of the following interconnected devices designed to allow students and researchers to implement their industrial application experiments.

- 1) Motoman manipulator: This robot arm enables students to manipulate the objects coming along the conveyor belt and classify them on the auxiliary table.
- 2) FPGA and on-hand camera: The Motoman manipulator is equipped with an on-hand high-resolution camera (1024 × 768) that permits the implementation of visual servoing controls as well as object tracking algorithms.
- 3) Conveyor belt: It allows objects to be moved in any direction as well as at different speeds. Once an object reaches the end of the belt, two sensors indicate the situation and allow the operator to act accordingly.

¹Application-specific integrated circuit.

²Digital signal processor.

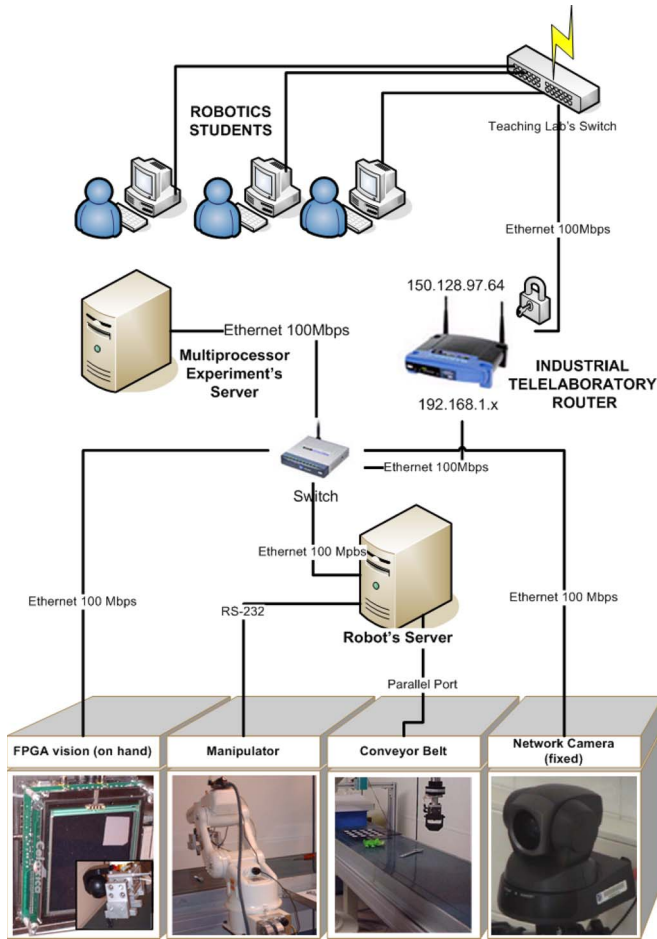


Fig. 3. UJI Industrial Telelab networking configuration.

- 4) Fixed network camera: On top of the conveyor belt, a network camera is located that provides real-time video streaming of the objects present on the conveyor belt. This camera is calibrated with the conveyor belt, which permits the implementation of robot control algorithms using on-top visual servoing techniques.
- 5) Robot's server: A pentium III PC is connected to the Motoman manipulator and the conveyor belt, enabling its remote control and programming through the corresponding SNRP servers (see next section for details).
- 6) Multiprocessor experiment's server: This computer holds the software SNRP framework for user identification, SNRP naming service, the telelaboratory educational webpage (i.e., moodle based [21]), and the experiment's holder server, which allows a Java experiment to be uploaded to the server and launched when appropriate.

Fig. 3 shows the network connectivity of the UJI Industrial Telelaboratory. In order to enhance the security measures, the whole telelaboratory is accessible through a unique router/firewall, which permits defining, in a very detailed way, the devices and services (i.e., ports) which are remotely accessible to students.

This network configuration has demonstrated to be simple and reliable enough to let students program their tele-experiments during several undergraduate robotics courses in the university. Some essential parts are the router, which creates

a private network within the telelaboratory in order to guarantee security, and the experiments server, which has demonstrated an overall increase of the system performance since it was added to the system.

III. SNRP PROTOCOL

As explained in [22], networked robotics is an emerging research area for creating intelligent robotic architectures that integrate embedded systems, sensors, and actuators networks.

The challenge is defining a software and network architecture within the network robotics area, which provides the following features: 1) simple, 2) open, 3) flexible, 4) dynamic, 5) robust, 6) scalable, 7) efficient, 8) secure, and 9) platform independent.

Simplicity is maybe the most important challenge of the network robotics architecture, due to the fact that it must be possible for a very broad range of devices to be part of it. In fact, as this paper describes hereinafter, it is possible to implement a prototype of SNRP Network Camera using an FPGA.

In the scientific literature, several works can be found that propose different ways and architectures to organize task-oriented applications of multiple network robots [23], [24]. Some of these architectures are focused on Internet software frameworks (e.g., Web Services) and have been extended from previous works in single-robot telerobotics [25].

Other works focus on the Internet network protocols themselves and study Internet transport protocols that enable real-time control and teleoperation of network robots over IP. In fact, as explained in [26] and [27], solutions can be found to cope with the problems associated to the Internet in order to control networked robots: 1) time-varying transmission delay and 2) not-guaranteed bandwidth.

In Fig. 4, we can see the software architecture of the SNRP framework, which provides the following modules.

- 1) SNRP_Robot: Every robot/device in the SNRP framework would provide an SNRP_Robot network interface, which allows any SNRP experiment to use a particular device connected to the network.
- 2) SNRPRobotsGroup: An SNRP robot can be the union of several SNRP robots (e.g., a mobile manipulator is the union of a mobile robot, a robot arm, and a set of sensors). Thus, SNRPRobotsGroup allows new services to be defined for the various network robots that work together as if they were a single robot.
- 3) SNRPNamingService: An SNRP network robot can register itself with a naming service in order to select a name (e.g., UJI/telelabs/industrial/motoman) and inform other peers of which IP and port he is attending to.
- 4) SNRPServiceHolder: The services provided by an SNRP robot can be programmed in a static manner within the SNRP module itself, or on the other hand, they can be added dynamically in runtime. For that, an SNRP service that follows a given interface will have to be uploaded into the SNRPServiceHolder.
- 5) SNRPExperiment: An SNRP experiment is a student program that can be uploaded into a service holder.

This completes the presentation of the software architecture for the SNRP framework. The SNRP protocol itself, which

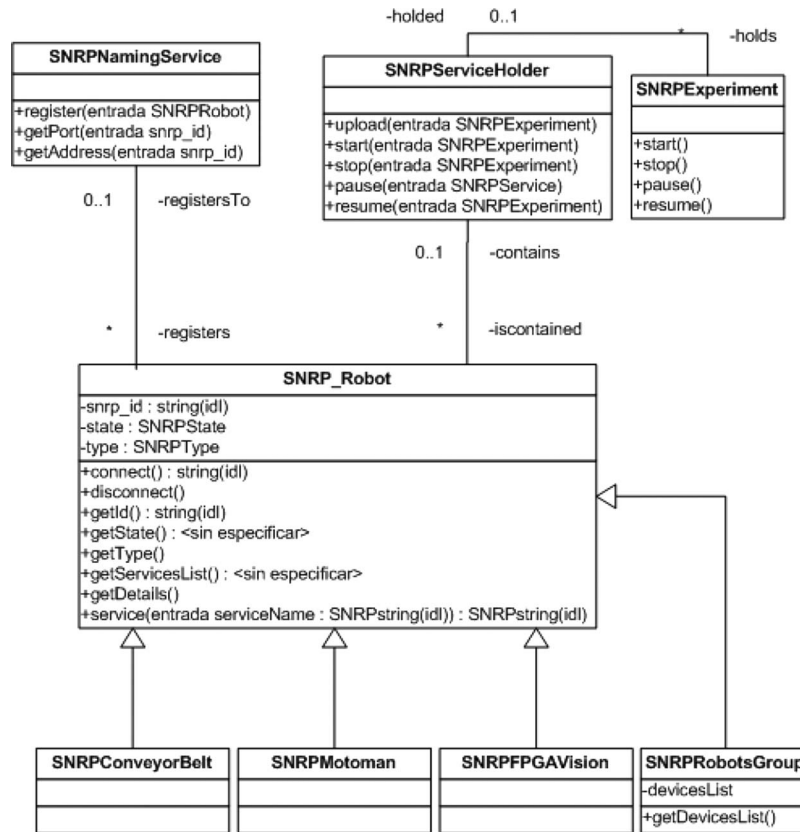


Fig. 4. Software architecture of the SNRP framework: UML diagram.

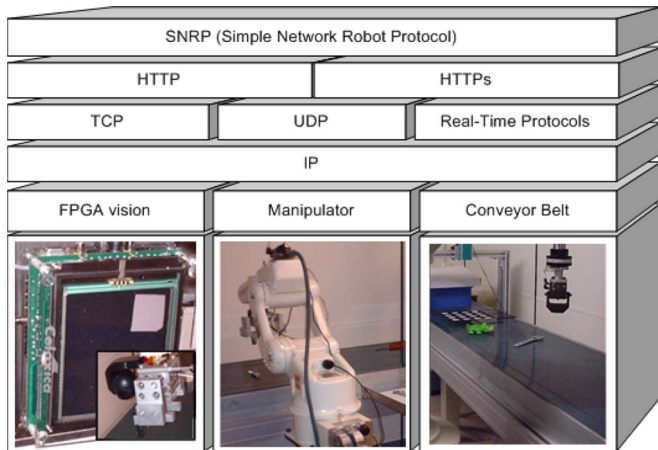


Fig. 5. SNRP network architecture.

allows SNRP experiments, holders, naming services, and robots to communicate with each other, will now be introduced.

Fig. 5 shows a detailed description of the SNRP network architecture. The first objective is to enable the devices to be accessed through the Internet, so that they should be able to manage the IP protocol. In addition, the SNRP framework enables the device to accept TCP and UDP connections. As explained before, UDP and TCP do not provide satisfactory solutions to the problems of performing remote control through the Internet, so the SNRP framework provides the possibility to transport the Internet datagrams through other protocols like “trinomial” [28], RTP (real-time transport protocol), the RAP (Rate-based adaptation protocol) [29], etc.

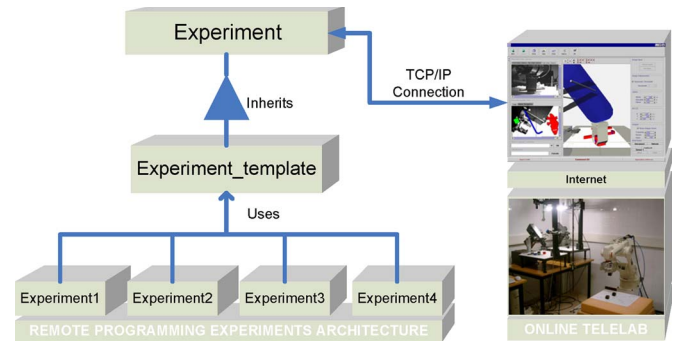


Fig. 6. Organization of the Experiments Library.

In order to make the SNRP simple to use and implement, it uses the HTTP protocol as a basis, which gives it even more interoperability and flexibility. However, for this kind of situation, the HTTP does not provide the following features: 1) event notification and 2) support for structured information. These two characteristics are very important for the design of the SNRP framework in the industrial robotics area. To accomplish this, we have incorporated into the SNRP protocol the REST model [30], which permits the implementation of state-oriented applications and a simple scenario to design event notification and structured information features.

To let students and researchers program the Telelaboratory in a simple and reliable manner, they are provided with a Java Library called “Experiments” that already manages the implementation details of the SNRP interface. This library already includes templates that are examples of simple experiments that manage the remote robots and the cameras (see Fig. 6).

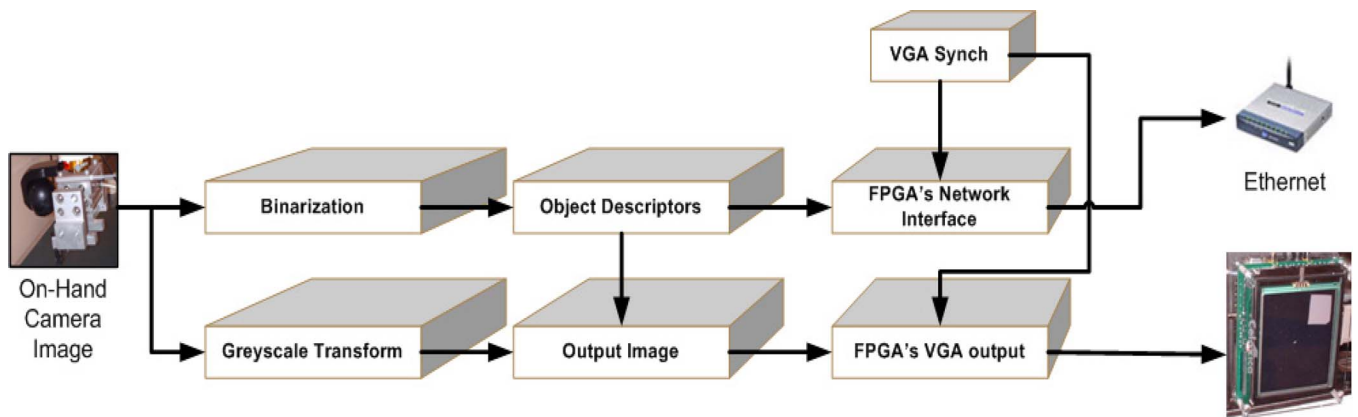


Fig. 7. FPGA image processing flowchart.

The robotics telelaboratory, developed previously (i.e., UJI Educational Robotics Telelab), uses Common Object Request Broker Architecture (CORBA) and Remote Method Invocation agents as network architecture [1]. Results will show the performance of visual servoing experiments performed on both the educational (CORBA-based) and the industrial (SNRP-based) telelaboratories.

IV. FPGA-BASED VISION

This section focuses on the FPGA design that has been carried out in order to obtain the high-performance SNRP vision system. One of the advantages of designing FPGA-based embedded systems in robotics is that they provide high performance without taking up much space. Moreover, within the e-learning platform context, an FPGA-based vision system has the advantage of allowing the user to design high-performance algorithms and computer architectures via the Web.

First, we have selected a prototyping FPGA platform based on the Virtex II chip, which provides a 100BaseT Ethernet connection, video input/output, high-resolution color camera, and a TFT display. In particular, the used platform is the Celoxica RC203E that has a Virtex_II 2V3000-4 FPGA, Ethernet interface, TFT, and other input/output interfaces.

Current efforts attempt to compile high-level languages, such as Matlab, directly into FPGA implementations [31]. Other tools use derived languages based on C such as Handel-C [32], based on C++ class libraries such as System C [33], or Java classes such as JHDL [34]. Recently, a more affordable approach for system designers is using “block-based design”, where a graphical tool (GUI) allows the interconnection of parameterizable IP cores (an IP core is a reusable unit of cell, logic or chip layout design) from a library for creating processing systems. PixelStreams [35] is an example of this sort of library for digital image and video processing.

The RC203E platform provides a development environment (i.e., DK) based on the Handel-C hardware description language, which has a similar syntax to the ANSI C language, and includes a library to design video systems called PixelStreams. Aside from this, the same vision experiments have been performed with the RC340 Virtex IV FPGA, which gives a much better performance in terms of processed frames per second.

The PixelStreams library is very appropriate for these kinds of applications, due to the fact that the circuit it supplies is already segmented, it has specific video input/output modules, and it comes with IPs for creating video filters and transforms.

The image processing task performed by means of the FPGA is divided into two principal data flows, as shown in Fig. 7. In the first data flow, the image taken by the “on-hand” camera is binarized. Then, object descriptors are obtained by using a specific module designed “ad-hoc” for this purpose [36]. This module is not present in the PixelStreams library and has been implemented from scratch using a high-level design tool. Finally, object moments are sent to the network through the FPGA network interface using the SNRP protocol.

In the second data flow, the image is transformed into a grayscale image and combined with visual information from the “Object Descriptors” module. Hence, an augmented reality image is shown on the TFT display of the FPGA board, showing in every frame the object centroid and the grasping line to be followed by a two-fingered parallel gripper robot. The two data flows are synchronized by means of the “VGA Synch” module in order to avoid conversion and visualization problems.

The Object Descriptors calculated by the FPGA system for frames captured by the camera include the following information (see Fig. 8).

- 1) *Object centroid*: For the captured object, the FPGA calculates the object centroid in camera coordinates.
- 2) *Inertia axis*: The direction of the minimum inertia axis (I_{\min}) and the maximum inertia axis (I_{\max}).
- 3) *Perimeter*: The estimation of the object perimeter.
- 4) *Grasping point*: In this version, the grasping points that enable a two-fingered parallel jaw gripper to pick the object up are calculated as the intersection of the centroid, the maximum inertia axis, and the object contour. Further work will focus on the application of a more sophisticated grasping determination algorithm that provides a list of possible grasps for a given object [37].

The next step consists of providing the object descriptors computed by the FPGA to the network, through the SNRP protocol. In fact, the FPGA provides a service for informing the client about the object properties in the robotic environment (i.e., grasping line, centroid, and geometrical descriptors). Since the client/server synchronization is on demand, the

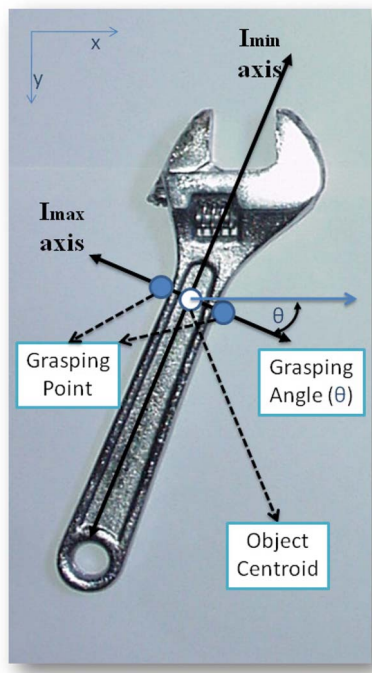


Fig. 8. Visual features computed by the FPGA vision system in order to obtain the centroid of the object, the minimum inertia axis (i.e., I_{min}), the maximum inertia axis (i.e., I_{max}), and the grasping angle of the object.

TABLE I
PERFORMANCE OF THE FPGA-BASED VISION SYSTEM

	RC203E	RC340
Image Resolution	704x576	1024x768
Segmented scenes per second	14,79	47,55

FPGA waits for a UDP datagram containing an SNRP command (i.e., GET/service/object/getGrasp). Once the content of the input is checked, the FPGA generates a new UDP datagram contained on an Ethernet frame using the sender MAC and IP address. The process is performed in a very fast manner, due to the fact that the FPGA does not need to implement the whole TCP/IP stack. This greatly enhances the performance of the system. Table I shows the performance results for the two different FPGAs used.

V. SYSTEM IN ACTION: REMOTE PROGRAMMING OF VISUAL SERVOING LOOPS

Visual servoing is a rapidly maturing approach to the control of robot manipulators that is based on visual perception of robot and work piece location. More specifically, visual servoing involves the use of one or more cameras and a computer vision system to control the position of the robot's end effector relative to the work piece as required by the task. It is a multidisciplinary research area spanning computer vision, robotics, kinematics, dynamics, control, and real-time systems.

Remote visual servoing techniques are normally used for teleoperation using a real-time communication bus. When using the Internet as communication media, the challenge is bigger, due to the fact that unpredictable time delays and bandwidth are introduced. Moreover, if the system is designed in a distributed

way and allows the concurrent control of multiple network robots, the challenge is even bigger.

The idea is to introduce the Internet communication channel inside the visual servoing loop and then to enable the remote programming of a real Internet telelaboratory to test those algorithms in a simple way. The proposed experiment will show the benefits of using the SNRP network architecture as well as the great advantages of implementing an SNRP FPGA vision system.

The remote experiments that have been tested are the following.

- 1) **Experiment 1 (Eye-In-Hand PC / CORBA):** For this experiment, the student gets an image as input extracted from a camera mounted on the robot arm. This image is transferred from the remote camera to the user via Internet. The gripper fingers of the robot will appear in the scene as two independent objects, so that the student has an additional difficulty when he comes to calculating the next robot movement. The feature extraction and the pose determination are performed by the user experiment on the client side (see Fig. 9). Moreover, as this experiment is performed with the older educational telelaboratory, the interconnectivity of the devices is based on the CORBA platform.
- 2) **Experiment 2 (Eye-In-Hand Server Based FPGA / SNRP):** This experiment is performed in a more recent industrial telelaboratory, where students are able to program an industrial robot, a conveyor belt, and an FPGA vision system. The network protocol is based on the SNRP architecture instead of the CORBA one. For this experiment, the students get the geometrical representation of the object (i.e., centroid, grasping line, etc.) from the FPGA, and he/she calculates the control law accordingly in order to grasp the object and place it on the required position. The image is extracted from a camera mounted on the robot arm, and the feature extraction and the pose determination are performed this time by an FPGA (see Fig. 10).

VI. REMOTE VISUAL SERVOING EXPERIMENTAL RESULTS

This section presents the results obtained from performing the experiments described earlier in order to compare both the previous CORBA-based architecture used in the educational telelaboratory, and the SNRP architecture and the FPGA vision systems within the industrial robotics telelaboratory.

A. Experiment 1

In this first experiment, the image acquisition, image processing, and pose determination are performed by the "Experiment" library at the student's side. As shown in Figs. 11 and 12, the students must bring the centroid of the object to the centroid of the gripper by iterative movements of the robot.

The whole computer vision process takes about 2.5 s of the whole manipulation operation, which means that we could improve the overall system performance by optimizing the binarization, segmentation, and features extraction procedures.

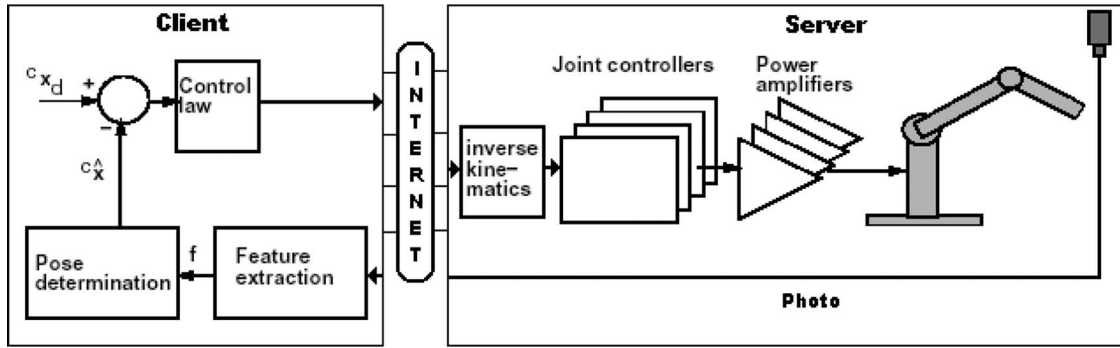


Fig. 9. Computer vision performed at the client side within the Experiment 1.

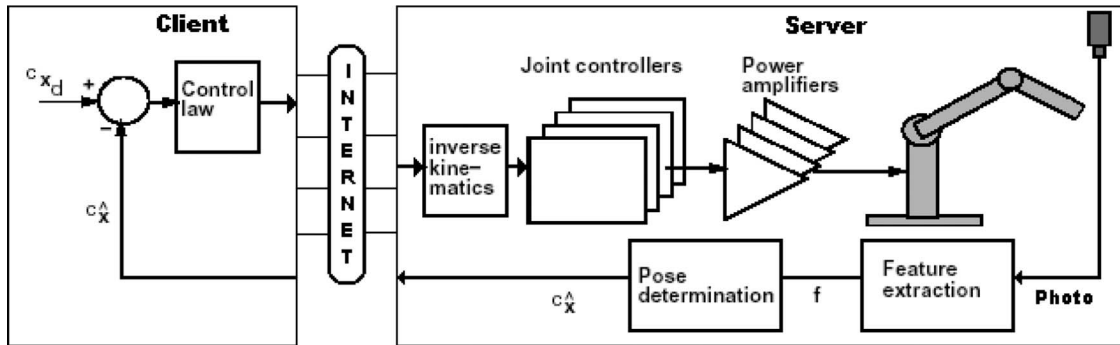


Fig. 10. Computer vision performed at the server side by the FPGA vision system in the Experiment 2. Geometrical features are provided to the student via the network.

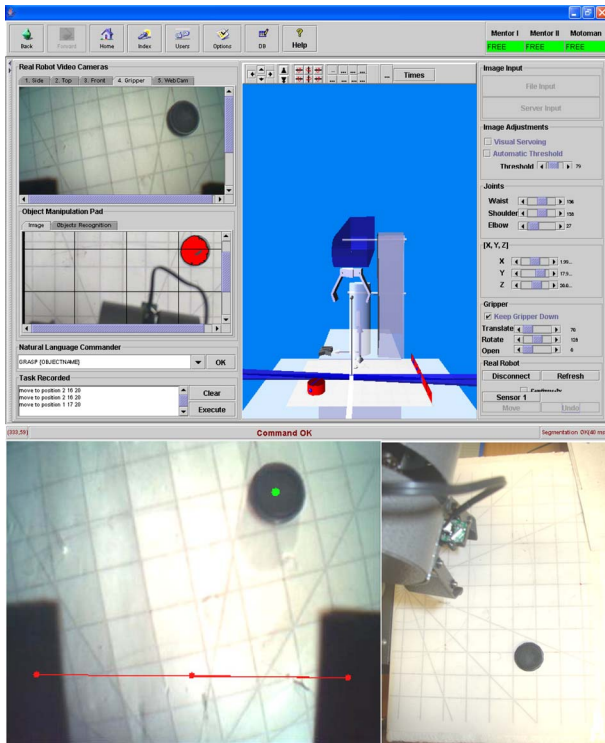


Fig. 11. Initial position of the robot as shown by the educational telerepository user interface and of the robot at the robotic scenario.

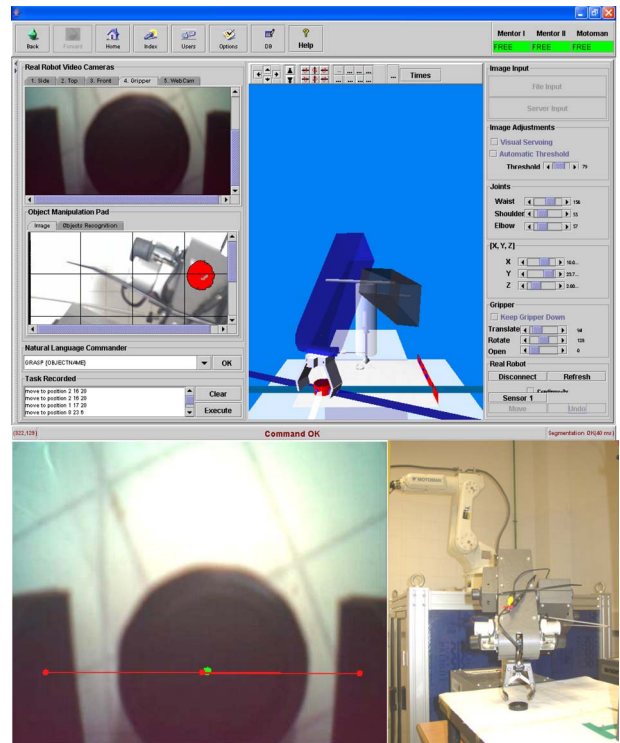


Fig. 12. Final position of the robot as shown by the educational telerepository user interface and of the robot at the robotic scenario.

This means that by including a high-performance vision system, the visual servoing performance could be greatly enhanced. The robot would then operate in a smoother way, which would show the students the way that robots usually work in industrial applications.

Figs. 13 and 14 show the times employed for both the client and the server side, respectively. Most of the time is used in sending the images through the Internet connection (i.e., GETIMAGE item in Fig. 13) and by waiting for the robot to

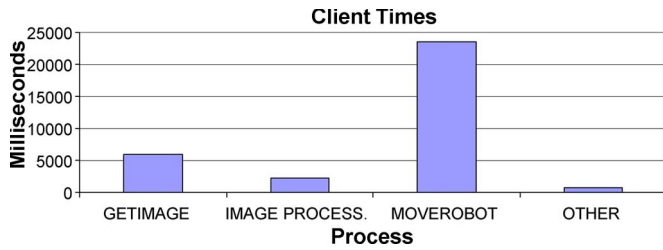


Fig. 13. Time employed by the client for Experiment 1 (GETIMAGE: Time invested to obtain the images from the telelaboratory server, including the communication latency; IMAGE PROCESSING: Time invested to segment the images and calculating the geometrical features of the objects; MOVEROBOT: Time since the movement command to the robot is sent, until the confirmation is obtained, including the networking latency).

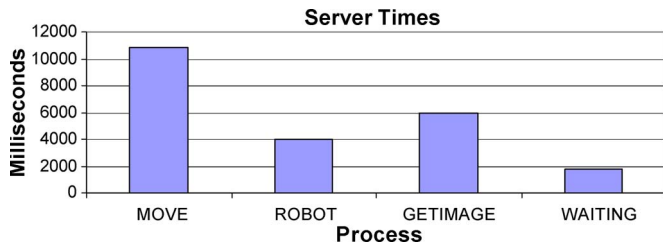


Fig. 14. Time employed by the server for Experiment 1 (MOVE: From the server side, the time invested by the telelaboratory server to communicate with the robot's CORBA server; ROBOT: From the Robot CORBA Server, time to move the articulations of the robot; GETIMAGE: Time invested by the server side to obtain the camera images from the video server; WAITING: Time that the server is waiting for the client instructions).

accomplish the required movement (i.e., MOVEROBOT item in Fig. 13). Moreover, the results show that it would be possible to improve the overall system performance by speeding up the image processing and pose determination functions (i.e., IMAGE PROCESSING item in Fig. 13) by applying parallelization techniques or by implementing these functions in hardware (e.g., FPGAs).

B. Experiment 2

In this second experiment, the image acquisition, image processing, and pose determination are performed by the FPGA. This information is provided to the students' experiment through the network using the SNRP architecture, instead of the CORBA one.

In this situation, as the FPGA capabilities are much bigger, we are using a higher image resolution (i.e., 1024×768) instead of the small images used in the educational telelaboratory (i.e., 352×288). Moreover, as we want the robot to move in a smoother way, the increments used in the control law (i.e., the visual servoing amplification factor) are decreased. In fact, for this situation, the whole visual servoing task is executed in an average of 25 loops per experiment instead of 10 loops that were observed in the first three experiments. The results obtained are shown in Fig. 15, where the average loop time is 221 ms. Comparing the results with the previous experiment, we can observe a great improvement in the average loop time.

In the FPGA side (see Fig. 16), it can be observed how the FPGA has wasted resources. In fact, the FPGA spends more time waiting than processing the image. Moreover, if we compare the time that the FPGA uses to perform the

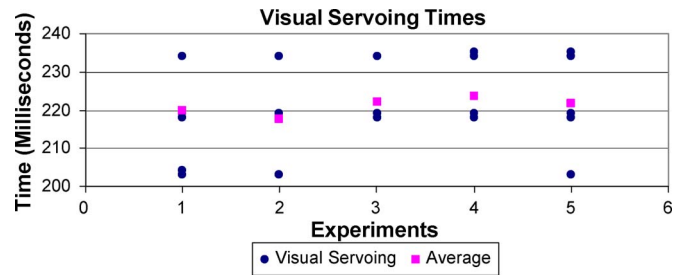


Fig. 15. Visual servoing times obtained for different experimental situations.

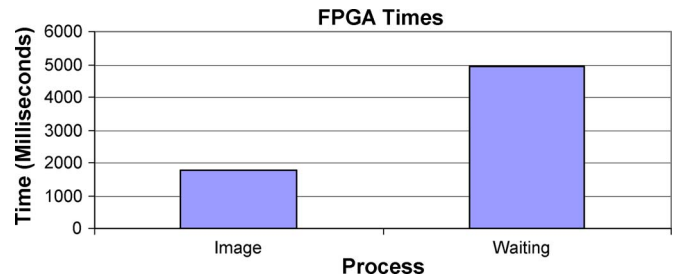


Fig. 16. Global FPGA times (IMAGE: Time invested by the FPGA to perform the computer vision; WAITING: Idle time at the FPGA).

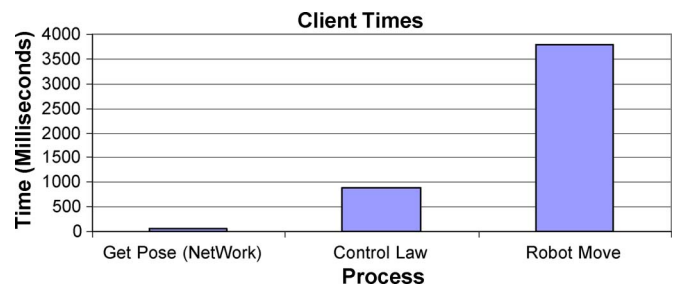


Fig. 17. Global client times (GET POSE: Time invested to obtain the object geometrical properties from the FPGA, including the network latency; CONTROL LAW: Time invested by the students experiment to calculate the next robot position; ROBOT MOVE: Time since we send a packet to the robot for the next movement until the acknowledgment is obtained).

image processing and pose determination (i.e., image column in Fig. 16) with the same time taken in the other experiments, it can be observed that using an FPGA makes it much faster even when using higher resolution images.

On the other hand, the implementation of the SNRP protocol makes a very big improvement. In fact, the average time invested to obtain the object features from the network is about 166 ms in Experiment 1 (i.e., using the CORBA-based approach). For the Experiment 2 (i.e., using the SNRP architecture), the object features are obtained from the SNRP FPGA vision system in an average of 10 ms, varying the responses from 3 to 15 ms (see Fig. 17 for a detailed distribution of global times for the client side).

VII. USING THE E-LEARNING PLATFORM IN EDUCATION

Experience has shown that the fact that students are able to perform their experiments with a real system is extremely motivating for them. They prefer to wait in order to be able to use a real robot rather than using a virtual system for their work.

The educational laboratory has now been used for several years in order to give classes to undergraduate students who are studying robotics and computer science. In addition, currently,

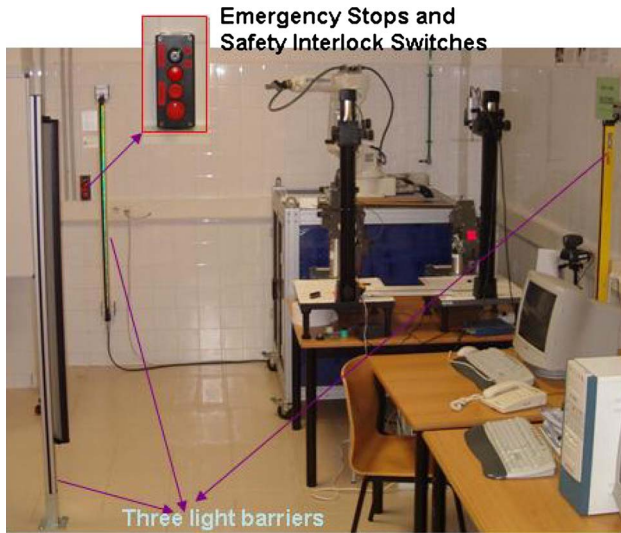


Fig. 18. Light barriers to guarantee students' safety within the educational telelaboratory.

the new industrial telelaboratory prototype is being used by undergraduate and postgraduate students in a more limited and controlled manner.

In summary, the features that both telelaboratories offer to the students when they perform their experiments are the following.

- 1) Perform robot control experiments via the Web.
- 2) Perform vision experiments with real cameras and real robotic scenarios.
- 3) Practice with the programming of real-time vision experiments with an expensive FPGA vision sensor, which can be reconfigured via the Web by the students.
- 4) Interact remotely with two different telelaboratories, the educational with simple robots and the industrial one.
- 5) The telelaboratories offer the students a simple and common way to communicate with every sensor and actuator of the robotic scenario, through the SNRP HTTP-based protocol.

On the other hand, an important aspect to be considered is "safety." While safety is always crucial for any robotic application, in the work presented here, which involves having students moving around the robots, this is an even more crucial consideration [38].

The best solution we found to keep the students completely safe is actually keeping them away from the robots while an experiment with them is being performed. As it is shown in Fig. 18, a system based on light-beam barriers was implemented, namely, "SICK's C 4000 Safety Light Curtain." This system is based on a safety light curtain, which guarantees the international standard IEC 61508 for plant safety. These units have been certified to Safety Integrity Level 3, the highest possible rating attainable for a safety light curtain.

VIII. CONCLUSION

This paper has presented the newly created UJI Industrial Telelaboratory, which permits students and researchers to perform remote programming experiments with a real robotic plat-

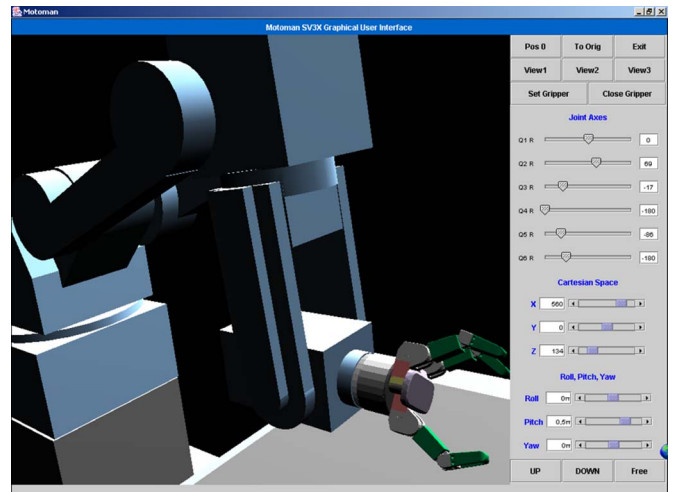


Fig. 19. Prototype of the SNRP v2 device browser showing a Motoman robot and a Barret Hand.

form. Moreover, the telelaboratory uses a distributed network architecture called SNRP that simplifies a lot the interaction between the different components of the system (i.e., robots, cameras, experiments, etc.). Results have shown that the SNRP protocol significantly enhances the performance of the whole system (over ten times faster).

Moreover, a FPGA has been used to implement a real-time vision system that provides SNRP services to the network. The FPGA takes as input the images from a camera mounted on the gripper of the robot. By having such an improvement in the computer vision module, we get the opportunity to program fast and reliable visual servoing controls over the industrial telelaboratory.

This paper has presented two real remote programming experiments that demonstrate that this technique is very appropriate for education. In fact, the remote visual servoing experiment has been selected to demonstrate that remote experiments could be used even in those situations where time response is crucial for performance (see Experiment 2 configuration). As shown in the results, by using the system in the same campus, the time delay could be considered reasonable when the control is based on position (as the example shown).

IX. FUTURE WORK

The next step will focus on the enhancement of the low-level implementation of the SNRP protocol, which allows a given experiment to get access to every device belonging to a telelaboratory. In fact, we are already preparing a new version of this protocol (SNRP v2) that would include an SNRP device browser in order to dynamically present a 3-D virtual environment for real SNRP devices (see Fig. 19). Thus, this interface opens the door to the design of semantic web services, knowledge storing, learning, etc.

At the same time, the experiments on visual servoing, which have been set up at the moment, are based on position. In the future, we will study the possibility of providing students with an API to control the speed of the robot too. This will be useful to analyze the remote visual servoing stability when having variable time delays.

REFERENCES

- [1] R. Marin, P. J. Sanz, P. Nebot, and R. Wirz, "A multimodal interface to control a robot arm via web: A case study on remote programming," *IEEE Trans. Ind. Electron.*, vol. 52, no. 6, pp. 1506–1520, Dec. 2005.
- [2] R. Wirz, R. Marin, and P. J. Sanz, "Remote programming over multiple heterogeneous robots: A case study on distributed multirobot architecture," *Ind. Rob.*, vol. 33, no. 6, pp. 431–442, 2006.
- [3] J. Fernández, R. Marín, and R. Wirz, "Online competitions: An open space to improve the learning process," *IEEE Trans. Ind. Electron.*, vol. 54, no. 6, pp. 3086–3093, Dec. 2007.
- [4] M. Wu, J.-H. She, G.-X. Zeng, and Y. Ohyama, "Internet-based teaching and experiment system for control engineering course," *IEEE Trans. Ind. Electron.*, vol. 55, no. 6, pp. 2386–2396, Jun. 2008.
- [5] R. Marín, P. J. Sanz, and A. P. del Pobil, "The UJI online robot: An education and training experience," *Auton. Robots*, vol. 15, no. 3, pp. 283–297, Nov. 2003.
- [6] R. Zurawski, "Industrial information technology is coming of age," *IEEE Trans. Ind. Informat.*, vol. 3, no. 1, pp. 1–2, Feb. 2007.
- [7] A. P. Kalogerias, J. V. Gialelis, C. E. Alexakos, M. J. Georgoudakis, and S. A. Koubias, "Vertical integration of enterprise industrial systems utilizing web services," *IEEE Trans. Ind. Electron.*, vol. 2, no. 2, pp. 120–128, May 2006.
- [8] J. García, F. R. Palomo, A. Luque, C. Aracil, J. M. Quero, D. Carrión, F. Gámiz, P. Revilla, J. Pérez, M. Moreno, P. Robles, and L. G. Franquelo, "Reconfigurable distributed network control system for industrial plant automation," *IEEE Trans. Ind. Electron.*, vol. 51, no. 6, pp. 1168–1180, Dec. 2004.
- [9] K. C. Lee, S. Lee, and M. H. Lee, "Worst case communication delay of real-time industrial switched ethernet with multiple levels," *IEEE Trans. Ind. Electron.*, vol. 53, no. 5, pp. 1669–1676, Oct. 2006.
- [10] L. Deng, F. Janabi-Sharifi, and W. J. Wilson, "Hybrid motion control and planning strategies for visual servoing," *IEEE Trans. Ind. Electron.*, vol. 52, no. 4, pp. 1024–1040, Aug. 2005.
- [11] S. Hutchinson, G. Hager, and P. Corke, "A tutorial on visual servo control, special issue on visual servoing," *IEEE Trans. Robot. Autom.*, vol. 12, no. 5, pp. 651–670, 1996.
- [12] A. J. Lacey, N. A. Thacker, S. Crossley, and R. B. Yates, "A multi-stage approach to the dense estimation of disparity from stereo SEM images," *Image Vis. Comput.*, vol. 16, no. 5, pp. 373–383, 1998.
- [13] S. Fathnam and G. Slavenburg, "Processing the new world of interactive media: The trimedia VLIW CPU architecture," *IEEE Signal Process. Mag.*, vol. 15, no. 2, pp. 108–117, Mar. 1998.
- [14] R. Dubey, P. Agarwal, and M. K. Vasanth, "Programmable logic devices for motion control—A review," *IEEE Trans. Ind. Electron.*, vol. 54, no. 1, pp. 559–566, Feb. 2007.
- [15] S. Jung and S. su Kim, "Hardware implementation of a real-time neural network controller with a DSP and an FPGA for nonlinear systems," *IEEE Trans. Ind. Electron.*, vol. 54, no. 1, pp. 265–271, Feb. 2007.
- [16] R. C. Cofer and B. Harding, *Rapid System Prototyping with FPGAs: Accelerating the Design Process*. Oxford, U.K.: Newnes, 2006.
- [17] E. N. Malamas, E. G. M. Patrakis, M. Zervakis, L. Petit, and J. Legat, "A survey on industrial vision systems, applications and tools," *Image Vis. Comput.*, vol. 21, no. 2, pp. 171–188, Feb. 2003.
- [18] E. Monmason and M. N. Cirstea, "FPGA design methodology for industrial control systems—A review," *IEEE Trans. Ind. Electron.*, vol. 54, no. 4, pp. 1824–1842, Aug. 2007.
- [19] J. J. Rodríguez-Andina, M. J. Moure, and M. D. Valdes, "Features, design tools, and application domains of FPGAs," *IEEE Trans. Ind. Electron.*, vol. 54, no. 4, pp. 1810–1822, Aug. 2007.
- [20] M. Casado, G. Watson, and N. McKeown, "Reconfigurable networking hardware: A classroom tool," in *Proc. IEEE Symp. HOTI*, 2005, pp. 151–157.
- [21] *Moodle Course Management System*. [Online]. Available: <http://moodle.org/>
- [22] G. T. McKee, D. I. Baker, and P. S. Schenker, "Network robotics: Dynamic reconfigurable architectures," in *Proc. SPIE Intell. Robots Comput. Vis. XXII: Algorithms, Techn. Active Vision*, 2004, pp. 360–369.
- [23] D. Baker, G. McKee, and P. Schenker, "Network robotics: A framework for dynamic distributed architectures," *IEEE IROS Int. Conf. Intell. Robots Syst.*, vol. 2, pp. 1768–1773, 2004.
- [24] R. Marín and P. Sanz, "Grasping determination experiments within the UJI robotics telelab," *J. Robot. Syst.—Internet Online Robots Telemanipulation Special Issue (Part 2)*, vol. 22, no. 4, pp. 203–216, Apr. 2005.
- [25] B. K. Kim, M. Miyazaki, K. Ohba, S. Hirai, and K. Tanie, "Web services based robot control platform for ubiquitous functions," in *Proc. IEEE ICRA*, Barcelona, Spain, Apr. 2005, pp. 691–696.
- [26] D. Lee and M. W. Spong, "Bilateral teleoperation of multiple cooperative robots over delayed communication networks: Theory," in *Proc. IEEE ICRA*, Barcelona, Spain, Apr. 2005, pp. 360–365.
- [27] P. X. Liu, M. Q. H. Meng, and S. X. Yang, "Data communications for Internet robots," *Auton. Robots*, vol. 15, no. 3, pp. 213–223, Nov. 2003.
- [28] P. X. Liu, M. Q. H. Meng, P. R. Liu, and S. X. Yang, "An end-to-end transmission architecture for the remote control of robots over IP networks," *IEEE/ASME Trans. Mechatronics*, vol. 10, no. 5, pp. 560–570, Oct. 2005.
- [29] R. Rejaie, M. Handley, and D. Estrin, "RAP: An end-to-end rate-based congestion control mechanism for realtime streams in the Internet," in *Proc. IEEE Infocom*, Mar. 1999, pp. 1337–1345.
- [30] R. T. Fielding and R. N. Taylor, "Principled design of the modern web architecture," in *Proc. ICSE*, 2000, pp. 407–415.
- [31] T. Hill, "Using MATLAB to create IP for system generator for DSP," Xilinx Inc., San Jose, CA, 2006. White Paper.
- [32] *Handel-C Language Reference Manual*, Celoxica Ltd., Oxfordshire, U.K., 2001. [Online]. Available: <http://www.celoxica.com/techlib/files/CEL-W0410251J4-60.pdf>
- [33] D. C. Black, J. Donovan, B. Bunton, and A. Keist, *System C from the Ground Up*. New York: Springer-Verlag, 2007.
- [34] P. Bellows and B. Hutchings, "JHDL—An HDL for reconfigurable systems," in *Proc. IEEE Symp. FPGA's for Custom Comput. Mach.*, 1998, pp. 175–184.
- [35] *The Pixel Stream Manual*, Celoxica Ltd., Oxfordshire, U.K., 2005. [Online]. Available: http://www.celoxica.com/cup/registered_users/manuals/default.pdf
- [36] G. León, J. M. Claver, and G. Fabregat, "Optimizing area on the generation of specific circuits on FPGAs for SIMD applications," in *Proc. Int. Workshop Appl. Reconfigurable Comput.*, 2005, pp. 160–167.
- [37] P. J. Sanz, A. Requena, J. M. Iñesta, and A. P. del Pobil, "Grasping the not-so-obvious: Vision-based object handling for industrial applications," *IEEE Robot. Autom. Mag.*, vol. 12, no. 3, pp. 44–52, Sep. 2005.
- [38] G. Giralt and F. Ingrand, "Dependability in human-centered robotics," *IEEE Robot. Autom. Mag.*, vol. 11, no. 2, p. 4, Jun. 2004.



Raul Marin received the B.Sc. degree in computer science engineering and the Ph.D. degree in engineering from the Universitat Jaume I, Castellón, Spain, in 1996 and 2002, respectively.

In 1996, he was with the Multimedia Department of BYG Systems Ltd., Nottingham, U.K., as a Software Developer. In 1997, he was with Lucent Technologies (Bell Labs Innovations), where he was a Researcher, Software Developer, and Software Architect with the Switching and Access Division. In 1999, he began to teach and conduct research

as a Professor at Universitat Jaume I, where he is currently a Researcher in the Department of Computer Science and lectures on computer networking and distributed systems. He is the author or coauthor of multiple research publications on these subjects. His research interests lie mainly in the field of multirobot distributed systems, high-performance FPGA-based vision, Internet telerobotics, network protocols, human-computer interfaces, and tele-education.



Germán León received the B.S. degree in computer science from the Universidad Politécnica de Valencia, Valencia, Spain. He is currently working toward the Ph.D. degree in computer engineering at the Universitat Jaume I (UJI), Castellón, Spain.

Since 1999, he has been an Associate Professor in the Department of Computer Science and Engineering, UJI. His research interests include design of FPGA-based systems, and reconfigurable and parallel computing focusing on high-level compilation techniques.



Raul Wirz received the M.Sc. and Ph.D. degrees from the Universitat Jaume I, Castellón, Spain, in 2005 and 2009, respectively. His Ph.D. thesis was related to the design of network protocols for educational robotics and teleoperation.

He is the author of several publications in journals and conference proceedings in these fields. His current research interests include network robotic systems, Internet tele-operation, network protocols for remote control, telemanipulation, high-performance computer vision, and e-learning platforms.



Jorge Sales received the B.Sc. degree in computer science engineering and the M.Sc. degree in intelligent systems from the Universitat Jaume I, Castellón, Spain, in 2006 and 2007, respectively, where he is currently working toward the Ph.D. degree in the Computer Science and Engineering Department, funded by the Spanish Ministry of Science and Technology.

His research interests rely mainly on the field of robotics, including telerobotics, wireless networks, object recognition, mobile robots, autonomous navigation, etc. The subject of his Ph.D. focuses on mobile robotics, navigation techniques, localization, and networked robotics.



José M. Claver received the M.Sc. degree in physics from the University of Valencia, Valencia, Spain, in 1984, and the Ph.D. degree in computer science from the Technical University of Valencia, Valencia, in 1998.

From 1985 to 1990, he was a Staff Member in the Electronics and Computer Architecture Department, University of Castilla-La Mancha, Albacete, Spain, where he was the Head of the department from 1989 to 1991. From 1991 to 2008, he was a Member of the Computer Science Department, Universitat Jaume I, Castellón, Spain, where he was the Head of the department from 2005 to 2007, Vice-Dean (1998–2001), and Dean of the Technical School of Engineering and Experimental Science. Since 2007, he has been an Associate Professor in the Computer Science Department, University of Valencia. He has taught undergraduate courses on computer architecture, electronics, and embedded systems, and graduate courses on high-speed networks, advanced computer architecture, and parallel computing. He is the author or coauthor of more than 30 research publications on these subjects. His research interests include computer architecture, parallel computing, high-speed QoS networks, network protocols, embedded systems, and reconfigurable computing.



Pedro J. Sanz (M'97) received the B.Sc. degree in physics from the University of Valencia, Valencia, Spain, in 1985, the M.Sc. degree in engineering (CAD/CAM) from the Technical University of Valencia, Valencia, in 1989, and the Ph.D. degree in computer engineering from the Universitat Jaume I, Castellón, Spain, in 1996.

He is currently an Associate Professor in the Department of Computer Science and Engineering, Universitat Jaume I, Castellón, Spain. He has been active since 1990 in R&D within several European and national research projects on advanced robotics. He is the author or coauthor of a broad range of research publications and other international scientific societies such as IAPR or ECCAI. His current research interests include multisensory-based dexterous manipulation, assistive robotics, telerobotics, and human-robot interaction.

Dr. Sanz has been an Associate Editor of the IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS—Part C since 2005 and the *IEEE Robotics and Automation Magazine* since 2008. He has also been a Program Committee member of several international outstanding conferences.



Josep Fernández (M'04) received the M.Sc. degree in computer science from the School of Computer Science of Barcelona, Barcelona, Spain, in 1989, and the Ph.D. degree in computer science from the Technical University of Catalonia, Barcelona, Spain, in 1998.

He is currently an Associate Professor in the Automatic Control and Computer Engineering Department, Technical University of Catalonia, where he lectures on digital systems design and robotics. He is currently working on projects involving telerobotics, online laboratories, and robotics systems for disabled people. He is the author of many contributions in international conference proceedings and book chapters in the field of computer vision and robotics. His research areas are in the field of computer vision and robotics, mainly oriented to sensor-based navigation in robotics applications, telerobotics, and telelaboratories.

Dr. Fernández is member of the EURON Education and Training group. He has been a member of the organizing committees of several international conferences.