Robotics and augmented reality for elderly assistance

Francisco J. Lera, Alvaro Botas, Carlos Rodríguez,
Juan Felipe García, and Vicente Matellán
Grupo de Robótica
Departamento de Ingeniería Mecánica, Informática y Aeroespacial
Escuela de Ingenierías Industrial e Informática, Universidad de León
fjrodl@unileon.es, infabm01@estudiantes.unileon.es, carlosrhrh@hotmail.com, jfgars@unileon.es,
vicente.matellan@unileon.es

***Abstract*—This article presents a tele-assistance system based on augmented reality for elderly people that is integrated in a mobile platform. We propose the use of augmented reality for simplifying interaction with its users. The first prototype has been designed to help in medication control for elderly people. In this paper, both hardware and software architectures are described. The robotic platform is a slightly modified version of the Turtlebot platform. The software is based on ROS for the platform control, and in ArUco for the augmented reality interface. It also integrates other related systems in teleassistance such as VoIP, friendly user interface, etc.**

***Index Terms*—Low cost robotics, Augmented Reality, Human-Robot Interface.**

I. Introduction

Many robotics platforms have been proposed in the literature for tele-assistance. These devices have been envisioned as robotic tele-presence solutions but, until now, cheap and simple robots are still far away from average users.

During the last years many projects have been working in elderly care using new technologies as computers, mobile phones o robotics. AAL (Ambient Assisted Living) project [6] or Oasis project [5] are some of them. They try to improve the daily life of elderly people in various fields: working in self confidence, improving security on individuals with medical problems, prevent cases of society isolation or teaching healthy lifestyle, etc.

In this article, we present our proposal of a robot that could help in daily life. This is not a novel idea [15], since years many companies and universities have been working in health care robots.

Focus on the past decade we can find robots like Robovie3 from V-Stonem [16], Wakamaru from Mitsubishi, Hospi-Rimo from Panasonic or CareBot from GeckoSystems are only a few examples built by companies. Examples as Pearl [17] from Carnegie Mellon, Michigan and Pittsburgh Universities, Care-O-bot from Frauenhofer Institut, Nexi from MIT or Maggie from University Carlos III Madrid shows how universities also focus their efforts in care robots.

In this proposal,the robot has been thought for assisting caregivers in control of medication of elderly people. One of the major contribution of this paper is the use of augmented reality for daily medical control.

Augmented Reality [13], since now AR, is a live view of a real-world scenario whose elements are "augmented" by computer-generated information such as sound, video, graphics, etc. In our system we overlay virtual graphics to the image captured by the camera.

Last year at WAF [3] we shown our first works related to Augmented Reality (AR) and how the virtual information overlaid in a real image can help users to tele-operate robots.

In the prototype we want to show virtual information over the traditional pillbox of a person under treatment. The computer screen of the robot would show a virtual figure overlaying the real world, pointing the correct medication that the patient has to take in a given moment.

There are other alternatives as PAUTA [4] project, that proposes electronic pillboxes that improve the generic box adding electro-mechanical system. The problem, from out point of view, is the high cost of this device and the subsequent maintenance. In our solution any pillbox is liable to be modified easily and used as augmented pillbox. This is the main contribution, we implement a low cost way to help in daily medication using augmented reality.

Our prototype also integrates communication capabilities to assist the person under treatment. We have included VoIP communications both to increase the information provided by the AR system, and to simply allow tele-conferencing. In summary our prototype integrates two different technologies to:

1) Augmented reality
2) Telepresence

Both solutions are integrated in a single multi-platform software application based on OpenGL and ArUco libraries integrated in a Qt interface using as a hardware platform a pair of computers on board of a modified version of the TurtleBot built by Willow Garage[12].

The main reasons to choose the Turtlebot were its easy integration with ROS, the software for robot control that we are using and the open hardware approach of its design.

Although the cost of Turtleblot is not very high, thanks to openness of its architecture we think that the cost can be reduced if we make some changes to its design. This will be the second major contribution of the paper: the hardware modification of the original robot boards controllers, allowing the

Roomba iRobot vacuum cleaner to be a TurtleBot compatible robot.

In summary, in this article two main issues are presented:

1) Our hardware platform: construction of the modified version of the turtlebot.
2) Our software architecture: augmented reality interface and tele-assistance system.

The rest of the paper is organized as follows. In section II, we first present the original Turtlebot. Then, our hardware prototype is presented, focusing in the changes made in order to reduce the platform price.

In section III, software developed is presented, showing the different layers, from hardware to user, that we have created.

In section V, we present the prototype built. We show all necessary steps, as a user manual, to get the prototype working.

In section VI, the conclusions and further work are summarized, highlighting the main outlines of our work and how we want to deploy the next phases of our project.

II. Hardware Description

There are two different ways to get a TurtleBot robot. The first one is to buy it to a company in the USA. They sell an integrated version of TurtleBot, where every component has been mounted and tested. The second option is to buy each component separately and assemble them by ourselves. The problem in this second case is that some components are not distributed in Europe

In this section we are going to describe these modifications. First we will describe the original robot and then our modification.

A. *TurtleBot*

In this section we describe the original TurtleBot robot shown in figure 1. Its principal components are:

1) iRobot Create® + board (gyro)
2) ASUS1215N Laptop
3) Microsoft Kinect camera
4) TurtleBot Structure
5) TurtleBot communications.

The design of TurtleBot is open hardware, which means that anybody can change or redesign the system as needed to fit it to their requirements. All components are susceptible to be modified :

1) Base iRobot Create. It can be changed for other platforms.
2) Computing unit. It is originally a netbook that can be improved.
3) Structure. Willow Garage built 3 surfaces as a floors where components are placed. It can be changed.
4) Communications board. It connects the computer and the iRobot Create. It can be redesigned.
5) Extra board. It is mounted in a parallel port available in the iRobot Create base and supplies energy to the Kinect camera. A gyro can also be mounted as an extra sensor to TurtleBot.

Fig. 1: TurtleBot™ Willow Garage Design

We are going to build a slightly modified version, that we have named as the *"TurtleRoomBot"*, to reduce the cost of the platform. The name is given because we have changed the robotic platform, we have used the iRobot Roomba Vacuum cleaner instead of the iRobot Create, that is not available in Europe.

Changing the platform means that we also need to change the communications board. In next section we describe this board that not only let us change the robotic platform, but also replaces the extra board.

B. *TurtleRoomBot*

The first problem that we found was that the iRobot Create was not distributed in Europe, so the first component to change was this. The Roomba vacuum cleaner robot looked like the best alternative.

The main handicap of this platform is that it have not got a parallel port to mount the gyro board, so this sensor will not be available in our robot. The second problem was the power supply for the Kinect camera. We need an extra 12 V to get the camera running.

In the USA version the parallel port from the iRobot Create base provides 12V and the movement commands are sent to mini-DIN 7-pin port.

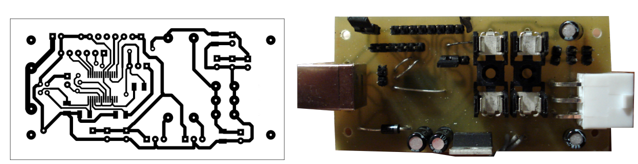

Fig. 2: First Board design and development

One alternative to this problem is to use the board designed by TurtleBot.eu [9]. This board gets 12 V from Roomba battery and lets the computer communicate with the base.

We think that this board can also be improved. We developed an initial prototype (figure 2) that has the same performance as the one by Turtlebot.eu with similar components but lower cost. The left connector is an USB port used to connect the laptop. The right connector in the picture (the bigger white dock) is used to connect the Kinect camera. The Kinect wire has to be modified to connect to this dock.

Once this board was tested we developed a more optimized production one shown in figure 3. In the left part this figure the original board by *turtlebot.eu* is shown. It is possible to see that our final version (right one) it is more compact and has less components than the original one.

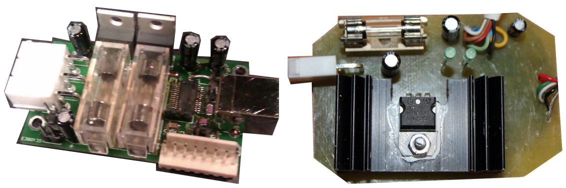

Fig. 3: Turtlebot.eu board and our final board design

This board has been built according to the Bus Powered Configuration documentation . The major change to the Turtlebot.eu board was the use of the RTS pin to activate the Pin DD (device detect) in Roomba connector. The BUS pin: BUS0 and BUS1 were configured as TXLED and RXLED. As in the original board, a simple voltage regulator circuit was built to supply energy to the Kinect camera.

The original board design is shown in the left picture in figure 3, our final board prototype can be seen in right picture in the same figure 3. Board encapsulation and connection are shown in figure 4. The mini-DIN 7-pin connector (down near the black rectangle in the figure) is what we use to communicate with the robotic platform and to get the power for the Kinect camera that will be our main sensor. More details of the communications and power supply board and can be seen in our *Cátedra* project website[1].

Fig. 4: First prototype running on Roomba

III. Software description

This section presents in more detail the software developed. In order to test it we have built a prototype that will be described in section V. We propose a system based on two main modules: ROS and MYRA.

[1]http://robotica.unileon.es/mediawiki/index.php/Catedraproject2012robotics

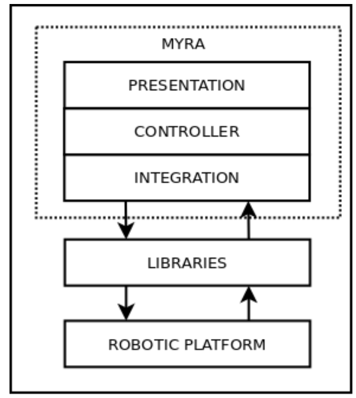

Fig. 5: MYRA layer architecture

ROS (Robot Operating System) is the set of libraries and tools used to build specific control software for robot developed by Willow Garage.

MYRA stands for “Elderly and Augmented Reality” (in Spanish *Mayores Y Realidad Aumentada*). It has been created at the *Cátedra Telefónica - Universidad de León* to help and improve the daily lives of elderly people through a toolbox for building AR interfaces.

A. ROS

A ROS application is defined as a meta-operating system for robots. It provides all common services to manage robot hardware and application software. ROS allows hardware abstraction and low level device drivers control. Libraries and other tools are provided to write, build and run code across multiple platforms.

ROS is defined as the peer-to-peer network of ROS processes (nodes) that are processing data together. But as we can see in the global architecture of figure 6, other concepts were introduced. For instance a *Package* contains everything to make a node run: sources, makefiles, etc. Many different packages, can be grouped to be distributed, in a bigger package called *stack*.

There are several benefits by using ROS: a big number of communities to exchange software and knowledge, repositories, distributions, new and improved releases, documentation, etc.

B. MYRA

MYRA has been designed following the model-view-controller (MVC) paradigm. For this reason we are able to include new independent modules, new functionalities, in the system. In this way, any of the modules could be improved without interfering with the others.

The MYRA architecture is a hierarchical architecture summarized in figure 5. Its three main levels are:

- Integration: level that provides connection with other libraries.
- Controller: modules to get information from the integration level and to provide information to the interface.
- Presentation: Interface generation.

The Integration level groups the software needed to connect to other libraries that MYRA needs, as for instance; OpenCV, ArUco, pjproject libraries or ROS ecosystem. These libraries provide image recognition, augmented reality and VoIP services respectively. ROS provides support for getting images from Kinect and sending commands to hardware platform to do the robot movement control. If a new library is needed a new interface in this level is created.

The *Controller* level consists of the mechanisms to get and use data from the *Integration* component. The software at the *Controller* level processes data received from the different subsystems and generates the information to the higher level, that is, to the *Presentation* component.

The Presentation level builds the interface that interacts to users.

In the figure 6 the layers to build a robotic assistant in dose control are presented. The robot and the laptop form the robotic platform layer. In the libraries layer the ROS libraries provides robot control (augmented for clarification), the ArUco library presents the AR, the OpenCV library offers computer vision algorithms, the Qt library facilitates the interface window creation and the PjProject library adds the communication capabilities.

In the management layer the *Controller* and the *Integration* MYRA components are presented. The management of ROS components is transparent to us and for this reason are presented as arrows.

The top layer provides the interaction interfaces with robot. Also the MYRA interaction interface (*Presentation* component) is available in this level. Each interface needs the corresponding management block.

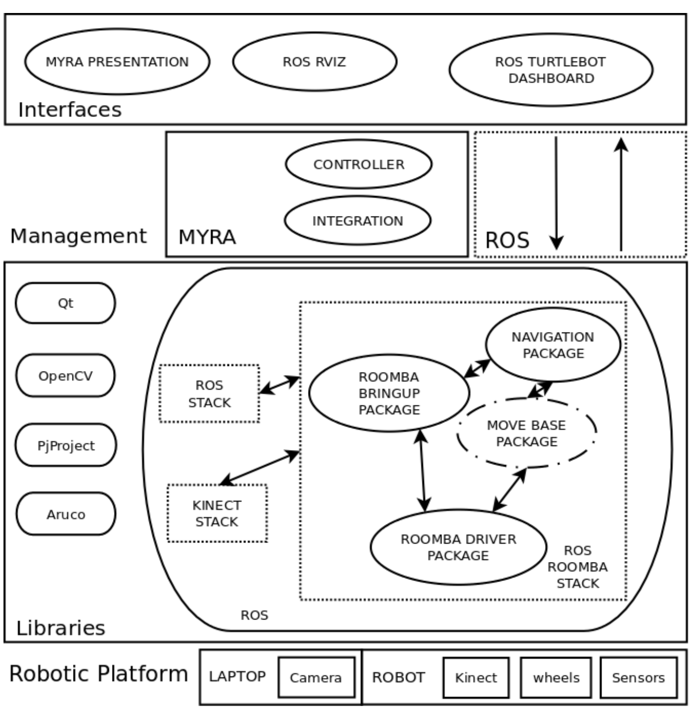

Fig. 6: Final robot high level architecture

IV. USE CASES

The typical situations where the system could be used are presented in figure 7.

We suppose that the user has the robot at home and needs to know which pill should be taken. If he can show the pillbox to the system, the next medication dose will be displayed on the screen using augmented reality. If he needs to call someone the telepresence system could be used.

If he cannot show the pillbox to the robot, the robot, by itself, or controlled from a remote care facility, will look for the pillbox and the next prescription will be shown.

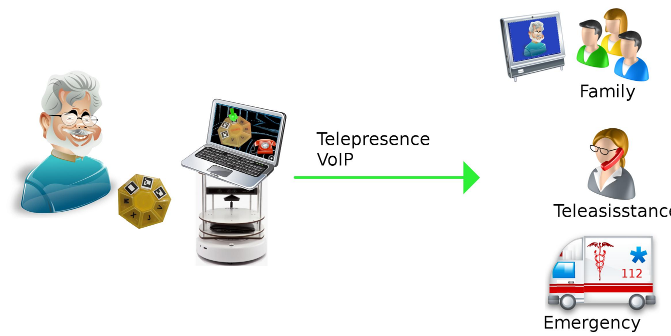

Fig. 7: Simple Architecture

Figure 8 tries to summarize the working possibilities of our prototype in three scenarios.

The first scenario is the robot running in autonomous mode in patient's home. This case corresponds to the middle block of figure 8. In this case MYRA runs onboard the robot. The communications with the robot are through two USB connectors. In figure 8, ROS libraries are also running in the robot in addition to MYRA. This allows autonomous behaviour of robot and reduce the number of necessary computers.

The second scenario presents MYRA running in an offboard computer wireless connected to robot. This computer shows the information from robot. The robot can be teleoperated using ROS software. This case corresponds to the right box of figure 8.

Third scenario is the robot-less one. In this situation using a webcam and a standalone computer we can install and run MYRA system. In this case ROS system is not needed, as shown in the left part of figure 8.

Finally, we want to stress that it is possible to make calls using the VoIP system to communicate with other MYRA users (with or without robot) and also to real phones using MYRA if the account is adapted.

The following section contains an explanation of each part necessary to run MYRA and how it is integrated in the first version of our prototype.

V. PROTOTYPE

In this section we describe the first prototype developed integrating all technologies described above.

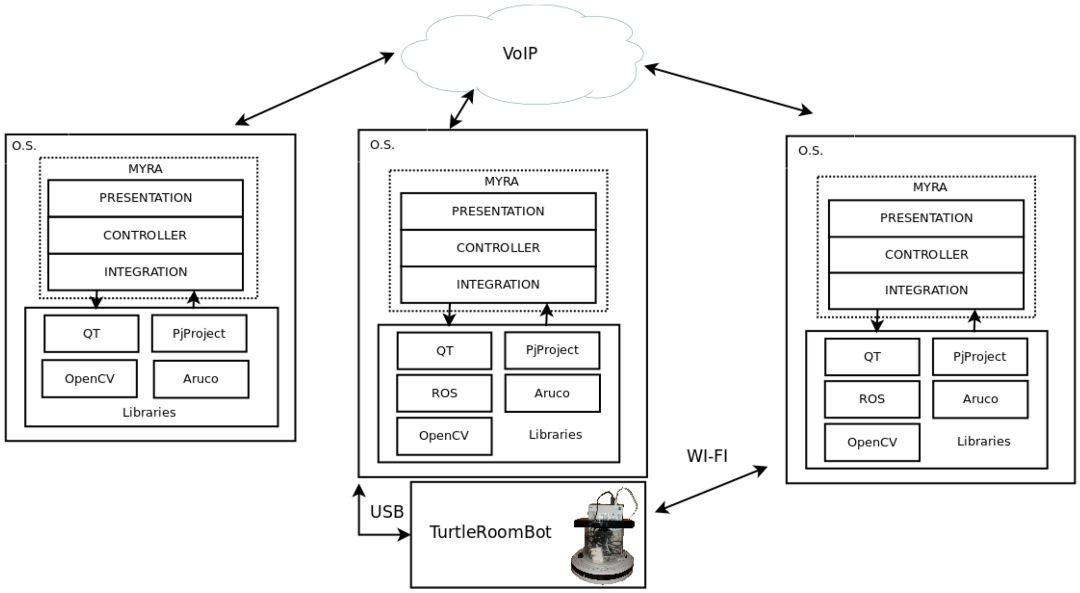

Fig. 8: MYRA configurations

A. Interface

The interface built for the prototype is made up by three different modules:

- Interaction Interface
- Telepresence
- Augmented Reality

The *Interaction Interface* is in charge of showing information to the user. This information is based on the images captured by the system, augmented with information added by the Controller component. The *Telepresence* module is in charge of managing the communications. In the preliminary prototype we are providing just voice calls, using the VoIP stack, which means that the remote user can be on a computer or on a phone. The *Augmented Reality* module manages the information included in the Presentation interface.

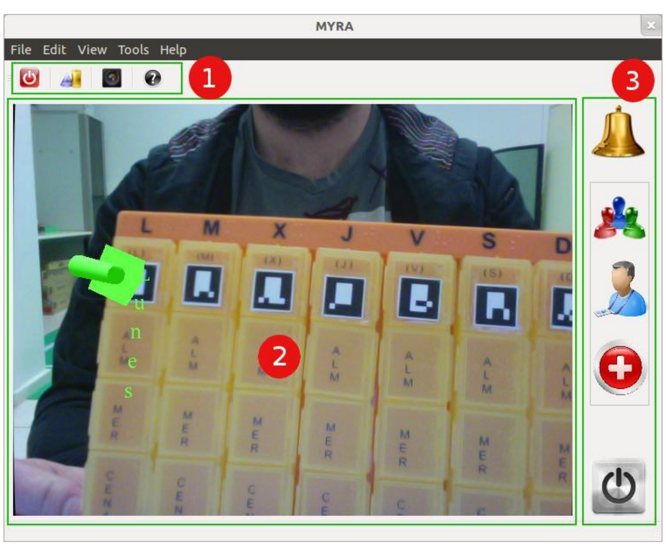

Fig. 9: Interface screenshot

In figure 9 we show a snapshot of the interface. We have tried to build an interface easy to use to facilitate the user experience. For this reason only a big frame for images and few buttons are used.

The frame, marked as 2 in figure 9, shows the video got from the camera at [10-20] frames per second. The speed depends on many parameters as the camera used (Kinect or computer camera), the image quality or the way of transmission (Wi-Fi transmitted or direct access).

Only four buttons, marked as 1 in the figure 9, are used to interface configuration: parameters as resolution or what kind of AR picture wants to be overlayed in the image frame. Another button is the interaction interface help to user.

In the right side, marked as 3, there are another four buttons: one for the application (the bell) and three for managing the tele-presence system.

The bell, when pushed, display a reminder with the lapsed time to the next dose. The other three buttons are used for the tele-presence system. The first one (from top) shows friends list, the next one allows the call to a day-center for elderly and the third one (the one with the cross) calls to the emergency line. The last button closes the application.

Our current MYRA implementation can make phone calls using VoIP. A SIP stack was used to implement the VoIP communications.

We integrated a modified version of QjSimple by Klaus Darilion distribuited under GPL license 2 [2] in our current implementation . This multiplatform application uses the PJSIP (Open Source SIP protocol stack).

We have kept on using ArUco [1] library because its ease integration in our developments.

In our interface, when a pillbox using our tags is shown to the camera, the AR system is activated and a shape is overlaid on the next dose medication.

Also a menu it is available to change the picture to show over the pillbox.

B. Pillbox and AR

Our system does not need an electronic or specifically designed pillbox. We can use any pillbox with a simple modification: adding a group of CODE's that has to be printed. It it's necessary to cover the corresponding moment of the day for the correct dose recognition.

In the example of figure 10, we show two different pillbox. In the picture of left side three days (Monday, Saturday, Sunday) are modified to be adapted to our system. Picture in right side has four different daily possible dose marked.

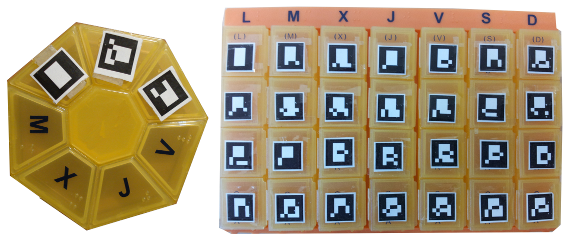

Fig. 10: Simple PillBox

As we explained before, we present a simple recognition system in order to help a final user to take the daily dose. Using our application MYRA and modifying whatever pillbox anyone can see the next dose.

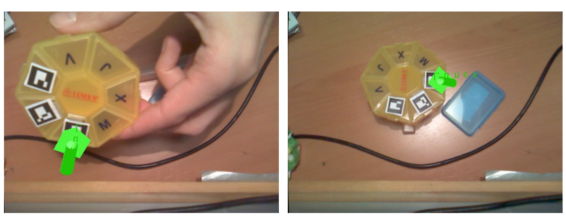

Fig. 11: Dose recognition with Augmented Reality

In the figure 11 we can see different snapshots of our pillbox and the AR marks, overlaid over the day. This model is the most simply case, with only seven possible cases, the week days. In the left image somebody grab the box and is showed to camera, in the right one, the box is on the table. In both cases the application shows the right medication dose.

C. Robot

The final prototype is shown in figure 12. In this prototype the board described in section II has been encapsulated in a box.

In this first prototype we are using an external computer to see the images obtained by the Kinect camera and teleoperate the robot, this corresponds with second use case described in section IV.

The netbook has been set into a plastic cylinder. This time the cylinder is transparent to see all wires and the board box to make sure everything works. In the next version a different solution is being considered such as to put the interface higher.

VI. Conclusion and further work

This paper describes the development of a cheap telepresence robot equipped with an augmented reality system for interaction.

Fig. 12: First complete Prototype

The two main contributions of our paper are:

- Hardware: The *TurtleRoomBot*, a low cost version of original *TurtleBot* manufactured by Willow Garage. This platform includes our redesign of the communication board, different platform and structure.
- Software: A human robot interaction architecture name MYRA used to build a system of medical dose control that includes augmented reality to improve the interaction with the robot and a telepresence system.

MYRA has been built using "live" open source libraries. This was a design decision that speed up the development, but also has its drawbacks: integration problems in some functionalities due to frequent libraries updates. The MYRA architecture developed for this project is available for downloading and testing.

As we presented along the paper, using our prototype is possible to follow the medical prescription and the medication dosages. Thanks to the help given for augmented reality,with just presenting the pillbox to our robot or to a camera if we use the MYRA computer solution. Also the telepresence system using VoIP, appears as a cheap way to communicate with family, friends or, with many improvements, as possible emergency system.

Hardware design is also available in the ROS community siteand our hardware improvements are available also in our website.

We hope to test the prototype in a real environment in the next months to get a real feedback, but we are studying certain hardware changes in order to elevate the camera and the display to be more comfortable with HRI.

Acknowledgment

This work has been partially funded by Cátedra Telefónica - Universidad de León (grant CTULE11-2).

References

[1] Research group, Aplicaciones de la Visión Artificial (A.V.A) Available: http://www.uco.es/investiga/grupos/ava/
[2] QjSimple, Online:*http://www.ipcom.at/en/telephony/qjsimple/*
[3] Juan F. García, Francisco J. Rodríguez, Vicente Matellán, Camino Fernández. *Augmented reality to improve teleoperation of mobile robots* XII Workshop en Agentes Físicos (WAF'2011). Albacete (España), September 2011.

[4] Project Pauta, [Online] Available:*http://www.proyectopauta.es/*.
[5] Project Oasis, [Online] Available:*http://www.oasis-project.eu/*.
[6] Project AAL, [Online] Available:*http://www.aal-europe.eu/*.
[7] Project OpenCV, [Online] Available:*http://opencv.willowgarage.com/wiki/*
[8] ROS, Robot Operating System, [Online] Available:*http://www.ros.org/wiki/*
[9] Turtlebot.eu project, KB Design [Online] Available:*http://turtlebot.eu*
[10] pjsip.org, Open source SIP stack and media stack for presence, im/instant messaging, and multimedia communication, [Online] Available:*http://www.pjsip.org/*
[11] QjSimple, [Online] Available:http://www.ipcom.at/en/telephony/qjsimple/
[12] TurtleBot, [Online] Available:http://www.willowgarage.com/turtlebot
[13] Ariel y la Fundación telefónica. *Realidad aumentada una nueva lente para ver el mundo*
[14] Azuma, R., Baillot, Y., Behringer, R., Feiner, S., Julier, S., MacIntyre, B., *Recent advances in augmented reality* ,IEEE Computer Graphics and Applications, 0272-1716, Volume 21, Issue 6, pp 3447, 2002
[15] J. Broekens, M. Heerink, H. Rosendal.*Assistive social robots in elderly care: a review*. Gerontechnology 2009; 8(2):94-103
[16] Ishiguro, H., Ono, T., Imai, M., Kanda, T., Nakatsu, R.: *Robovie: an Interactive Humanoid Robot.* International Journal of Industrial Robot, Vol. 28, No. 6 (2001) 498-503
[17] M. E. Pollack, S. Engberg, J. T. Matthews, S. Thrun, L. Brown, D. Colbry, C. Orosz, B. Peintner, S. Ramakrishnan, J. Dunbar-Jacob, C. McCarthy, M. Montemerlo, J. Pineau, and N. Roy, *Pearl: A Mobile Robotic Assistant for the Elderly*, AAAI Workshop on Automation as Eldercare, Aug., 2002