Artisti Humanoid Team for RoboCup 2007

Emanuele Menegatti*[1], Alberto Pretto[12], and Enrico Pagello*[1]

[1] Intelligent Autonomous System Laboratory (IAS-Lab)
Department of Information Engineering
The University of Padua, Italy
*{emg, epv}@dei.unipd.it
[2] IT+Robotics S.r.l. Vicenza, Italy. A spin–off of the University of Padua

Abstract. In this paper, we describe the hardware and software design of the two humanoid robots of the team Artisti. The robots are called Galileo and Leonardo. They are fully autonomous humanoid robots, based on the Robovie-M platform of VStone. They has low computational power, low resolution camera and low memory space. The robot vision system uses a low resolution camera working in QVGA mode. We devoted many efforts to the problem of QVGA pattern demosaicing in order to obtain better image reconstruction. Our intention is to use omnidirectional camera as unique sensor. The decisional architecture is deliberative and the robots can communicate (using Bluetooth) in order to exhibit team coordination with role swapping. These approaches are based on the techniques we developed for the Artisti Veneti Team, our team in the Middle Size League.

1 Introduction

The RoboCup event represents an extraordinary challenge for the autonomous robots proposed in all Leagues. For Humanoid League some of the problems are:

1. robot vision feature extraction in variable dynamic environment;
2. localization of the robot in environment;
3. safe control and collision free path planning in high DOF systems;
4. fast motions and stability in a real-time platform in order to react to the quickly changing environment;
5. coordination of group of humanoid robot to achive a common task;
6. design hardware and software embedded systems.

Our robots are called Leonardo and Galileo. They are based on a modified Robovie-M platform of VStone. Robovie-M platform is an embedded low cost system and has a low computational power.

These robots are fully autonomous: CPU, power supply, sensor and obviously actuators are on-board. The two robots have a wireless communication interface to exchange information in order to cooperate to perform a common task.

2 Mechanical Structure

The Robovie-M platform of VStone has been modified, as shown in Fig. 1, in order to be compliant with the rules of Humanoid Kid Size League, see RoboCup (2006).

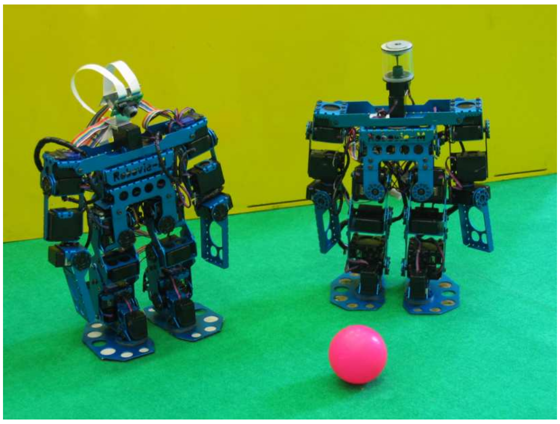


Fig. 1. Leonardo and Galileo in the two camera configurations.

Our modified version Robovie-M has size of $480 \times 235 \times 70$mm and a weight of 2.2kg. It is a fully autonomous humanoid robot that use as main sensor a camera. Our intent is to use a omnidirectional visor to play soccer in order to avoid the problem of limited visual field.

Our robots has 22 degrees of freedom distributed as follow: six for each lower limb, four for each upper limb and two for the bust; the actuators that move all DOF are servomotors. These actuators are directly controlled by CPU with PWM signals. The Table 1 shows the allowed movement of each joint and the Table 2 the specifications of the two types of actuators used in our platform. Servomotors has to be driven with a PWM signal with a range of the high level of the signal of $[0.5\text{ms}, 2.5\text{ms}]$. The period of PWM signal is fixed in about 17ms by a periodic interrupt.

3 Electrical Specifications

After a short experience with VS–7054 board by VStone, described in 4.1, we changed to K-Team's Korebot board. The new board's processor is Intel XS-CALE PXA-255 running at 400MHz, equipped with a FLASH memory of 32 MBytes and a RAM of 64 MBytes. It is able to support GNU-Linux Development Based Environment.

Part or joint	Rotation axes	Servomotor Type
Head and Body	Pitch, Yaw	*Hyper ERG-VB*
Shoulders	Roll, Pitch	*SPEC-APZ*
Arms	Pitch, Yaw	*SPEC-APZ*
Hip joint	Roll, Pitch	*Hyper ERG-VB*
Knees	Pitch	*Hyper ERG-VB*
Ankles	Roll, Pitch, Yaw	*Hyper ERG-VB*

Table 1. Type of servomotors associated with joints.

Motor	Torque	Speed	Size
Hyper ERG-VB	13 kg×cm (6V)	60/0.10s (6V)	39×20×37.4 mm
SPEC-APZ	4 kg×cm (4.8V)	60/0.20s (4.8V)	39×20×35.5 mm

Table 2. Technical specification of servomotors by Sanwa.

Korebot is intended to process image data, to control the behaviors of the robot and to output the signals for the 22 motors in a better way than the preceding architecture, whose abilities were strongly limited by the low memory capacity.

3.1 Sensors

The main and only external sensor, that gives information on the environment surrounding the robot, is the OV7620, a digital camera produced by OmniVision. This camera can be mounted as frontal or omnidirectional camera and can be used in two resolution modes: QVGA low resolution and VGA high resolution.

There is only one internal sensor: an ADXL202E a two axes accelerometer produced by ANALOG DEVICES. This is used to sense, if the robot is standing or if it is felt down.

4 Software Architecture

4.1 Preface to Software Architecture

Before adopting the current Korebot board, the two robots were powered, instead of the standard Robovie-M main board, by the VS–7054 board produced by VStone. The VS–7054 mounted a SH2–7054 MCU of Renesas running at 40Mhz with an internal FLASH memory of 384KByte and a RAM of 16KB. As external memory resources had got a RAM of 256K × 16bit and a EEPROM of 64KB. This board could control a digital camera.

The CPU offered enough computational power to process image data, to control the behaviors of the robot and to output the signals for the 22 motors.

4.2 Software Architecture

The Software Architecture described in this section is entirely based on VStone board presented in 4.1: as a consequence of this, algorithms presented were developed aimed to low computational power and low memory space.

One of our main goals is to advantage of the new Korebot board in order to build up better and more efficient vision and motion algorithms.

The firmware of VS–7054 for Leonardo and Galileo was developed by our team following the *Hierarchical Paradigm* as defined in Murphy (2000): the robot senses the environment, plans the next actions and then acts, as shown in Fig. 2. This very simple architecture is chosen because when the robot is moving the displacement of the camera is not controlled.

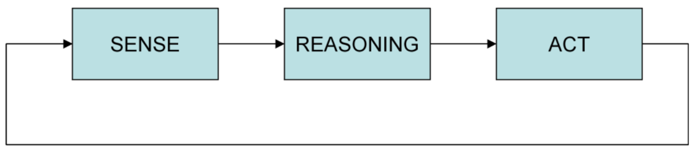


Fig. 2. Software architecture of RobotCore.

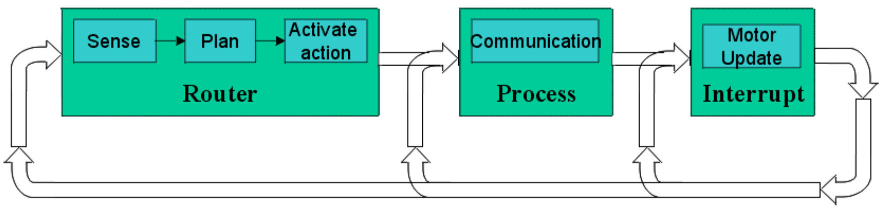


Fig. 3. Software architecture of RobotCore of Leonardo and Galileo.

The software architecture can be represented by three loops running at different speeds, as shown in Fig. 3. The inner loop controls the PWM motor output signals. These signals are obtained in asynchronous mode using interrupts. The medium loop include high speed functions as serial port and wireless port read–write operations and motor interpolation. The outer loop is the slowest one. In this loop is placed the behavior decision mechanism of robots. This module acquires and elaborates information on the surrounding environment, decides an action (accordingly to team task) and starts the action.

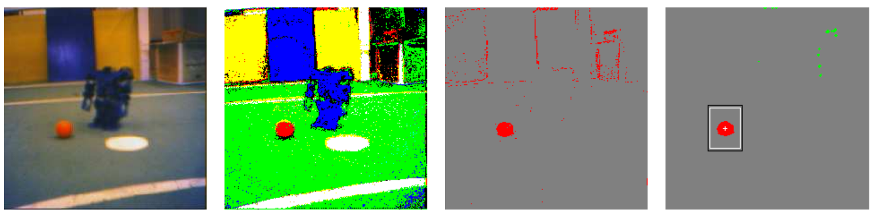
In Maggi *et al.* (2006) it is presented Galileo and Leonardo's simple behaviour selection architecture implemented with a finite state machine (FSM) in our robots. There are described highly optimized algorithms used for image processing and given some hints on how it is possibile to extend the flexibility of a low computational power humanoid with a customized operating system, such as the already cited VStone board. These solutions are anyway quite general and can be applied to any humanoid platform with low-computational power.

5 Research Approaches with Leonardo and Galileo

As already said a research approach in humanoid robots has to involve many disciplines as mechanics, electronics, control theory, robot vision, motion planning and artificial intelligence. The research in our team is focused especially on vision and motion planning.

5.1 Vision System and Image Processing

The vision system is designed to allow the use either of perspective cameras or omnidirectional cameras, see Menegatti, *et al.* (2002) and Maggi *et al.* (2006).



(a) Acquire. (b) Segmentation. (c) Blob detection. (d) Ball extraction.

Fig. 4. Example of image processing and features extraction of the ball.

Algorithm Flow. The image processing algorithm proceeds as follows, Fig. 4):

1. acquisition of the image, either in QVGA mode to acquire a complete frame covering the whole field of view of the robot's camera, or in VGA mode to acquire at higher resolution only the regions of interest (ROI) in which we want to focalize the attention;

2. demosaicing of Bayer pattern: for QVGA mode use *Periodic Reconstruction Interpolation*, for VGA mode using *Linear Interpolation with Laplacian Second–order Correction Terms*, as we described in Guseo (2006) and in Guseo *et al.* (2006);
3. color segmentation, with a look up table manually build offline and saved in EEPROM;
4. blob detection with labeling of connected components;
5. compute the centroids and the variance in two orthogonal direction of the pixel distributions of the ball, the goals, and the other robots.

Bayer Pattern Interpolation. Many digital cameras can function in two resolution modes: VGA (i.e., 640480) and QVGA (i.e., 320240). These cameras use a single sensor covered with a Color Filter Array (CFA). Several patterns exist for the CFA, the most common is the Bayer pattern. The CFA allows only one color component to be measured at each pixel, the remaining color components must be interpolated: this operation is called demosaicing. In Guseo (2006) and Guseo *et al.* (2006) we propose new solutions studying low computational cost algorithms for interpolating VGA Bayer pattern and QVGA Bayer pattern, which is important in low-cost and low-power embedded hardware.

5.2 3D Modeling

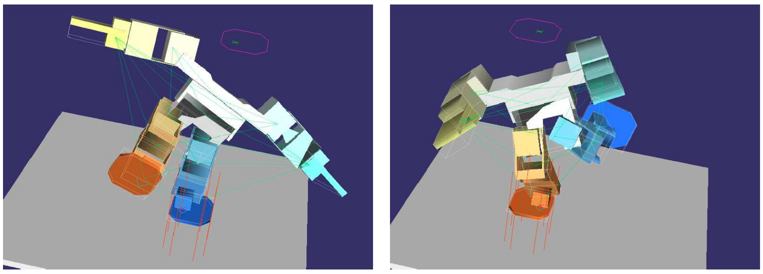
In Robocup one of the fundamental abilities which a robot has to demonstrate is to be able to move in a virtual environment, built up thanks to the sensorial signals received by its sensors.

For simulating our Robovie-M by Vstone robots and Robocup's playing field, we base on USARSim (Urban Search and Rescue Simulator): the dynamics and the appearance of the robot and the other objects within the simulator are faithfully reproduced in the virtual environment. Moreover, the virtual robot in USARSim can be controlled with the same program controlling the real robot, a program tool we developed in our laboratory.

5.3 Collision Free Robot Motion Planning

To design robot movements we are studying the use of Rapid Random–exploring Trees (RRT), LaValle (1998), to explore the joints configuration space, following the approach of Kuffner (2000) and Chen and LaValle (2001). An example is presented in Fig. 5 in which the robot has to swap the supporting leg, moving the COG from one foot to the other. In this movement, the hardest problem is to calculate the correct placement for the second foot. The problem was solved by a double RRT–Connect.

In RRT the exploration is rapid and complete due to the fact that the random search in unknown regions is proportional to size of this area. Another advantage is represented by precluded configuration that are represented by boolean



(a) Left. (b) Right.

Fig. 5. RRT motion planning single support swap.

functions. Trajectories generated by RRT are sub–optimal but these are very near to optimal trajectories.

In Carpin *et al.* (2006), it is surveyed state of the art research in motion planning for humanoids robots playing soccer: this suffers from many aspects of both theoretical and heuristic troubles, such as high dimensionality of configuration space, dynamic and static constraints, sometimes physical interaction with other robots, which render the choose for short and long term actions much more difficult than typical robotic problems.

5.4 Robot Simulator and Development ToolKit

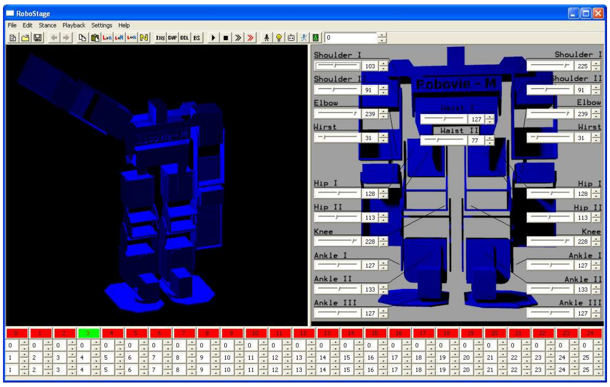
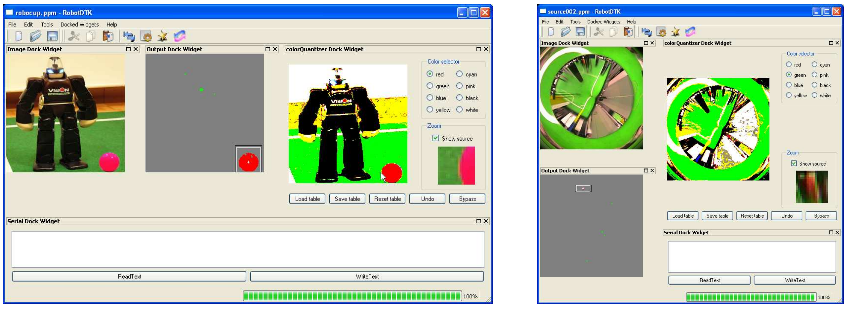


Fig. 6. Robot simulator.

A 3D visualizer of robot has been developed in order to allow the design of the realistic movements without the need of the real robot. We are working on a simulator with a dynamical model of the robot . A screenshot of the simulator is shown in Fig. 6.

A designed a simulator of the vision system and a development toolkit for vision processing called RobotDTK, Fig. 7. In this environment, all vision algorithm can be tested as if they where executed on the robot, with the advantages of using a faster CPU for the software design and trials. This development toolkit is useful in order to prepare offline the Look Up Table for color segmentation.



(a) Frontal. (b) Omnidirectional.

Fig. 7. Robot Vision Simulator and Development ToolKit.

6 Conclusion

In this paper, we introduced our humanoid robots named Leonardo and Galileo with 22 degrees of freedom and equipped with low computational power CPU. We present the software architecture we designed aiming not only at the RoboCup domain but also at different applications in general environments.

References

Carpin, S., Pagello, E.: The challenge of motion planning for humanoid robots playing soccer URL: http://www.dei.unipd.it/%7Eemg/whs2006/papers/HSR-114.pdf

Cheng P. and LaValle S. M.: Reducing metric sensitivity in randomized trajectory design. In Procedings IEEE/RSJ Int'l Conference on Intelligent Robots and Systems, 43–48 (2001).

Clemente, G., Pagello, E., Piaggio, M., Nardi, D., Adorni, G. Bonarini, A. and Chella, A.: Art99 azzurra robot team. In Veloso, M., Pagello, E. and Kitano, H., editors, *RoboCup–99: Robot Soccer World Cup III*, LNAI **1856**, Springer, 695–698 (2000).

Greggio, N., Silvestri, G., Antonello, S., Menegatti, E., Pagello, E.: A 3D Model of a Humanoid for USARSim Simulator First Workshop on Humanoid Soccer Robots December 2006, Genova, Italy, pp. 17–24, ISBN 88-900426-2-1 URL: http://www.dei.unipd.it/%7Eemg/whs2006/papers/HSR-105.pdf

Guseo, T.: *Architettura Software per Robot Umanoide Autonomo*, Tesi di laurea, University of Padova, Padova (2006).

Guseo, T., Menegatti, E.: Demosaicing Low Resolution QVGA Bayer Pattern 2nd International Conference on Computer Vision Theory and Applications, 8 - 11 March, 2007 Barcelona, Spain (to appear)

Kuffner J.J., LaValle S.M.: RRT-Connect: An Efficient Approach to Single-Query Path Planning, In Procedings IEEE Int'l Conf. on Robotics and Automation, 995–1001, (2000).

LaValle S.M.: Rapidly-Exploring Random Trees: A New Tool for Path Planning, TR 98–11, Computer Science Dept., Iowa State University, October (1998).

Maggi, A., Guseo, T., Pagello, E., Menegatti, E.: A light software architecture for a Humanoid Soccer Robot *First Workshop on Humanoid Soccer Robots* December 2006, Genova, Italy, pp. 25–31, ISBN 88-900426-2-1 URL: http://www.dei.unipd.it/%7Eemg/whs2006/papers/HSR-106.pdf

Menegatti, E., Pagello, E., Wright, M.: Using Omnidirectional Vision within the Spatial Semantic Hierarchy. *ICRA 2002* 908–914, (2002).

Murphy, R.R.: *Introduction to AI Robotics*, A Bradford Book, The MIT Press, Cambridge, Massachussetts, London, England (2000).

Pagello, E., Montesello, F., Garelli, F., Candon, F., Chioetto, P., Griggio, S.,: Getting Global Performance through Local Information in *PaSo-Team'98 RoboCup 1998* 384–389 (1998).

Pagello, E., Ferrari, C., D'Angelo, A., Montesello, F.: Intelligent multirobot systems performing cooperative tasks. In *Emergent Systems: Challenge for New System Paradigm Invited Session (IEEE/SMC Conference)* Tokyo, 754–760, October (1999).

Pagello, E., D'Angelo, A., Menegatti, E.: Cooperation Issues and Distributed Sensing for Multi-Robot Systems IEEE Proceedings of IEEE (in press)

RoboCup, compiled by Sven Behnke: *RoboCupSoccer Humanoid League Rules and Setup for the 2006 competition in Bremen, Germany*, URL: http://www.robocup.org/ (2006).