CAMBADA'2008: Team Description Paper

J. L. Azevedo, N. Lau, G. Corrente, A. Neves, M. B. Cunha, F. Santos, A. Pereira, L. Almeida, L. S. Lopes, P. Pedreiras, J. Vieira, D. Martins, N. Figueiredo, J. Silva, N. Filipe, I. Pinheiro

Transverse Activity on Intelligent Robotics
IEETA/DETI – Universidade de Aveiro
3810-193 Aveiro, Portugal

Abstract. This paper describes the CAMBADA middle-size robotic soccer team for the purpose of qualification to RoboCup'2008. Last year improvements have been made mostly in the vision system, in the high-level coordination and control and in the information integration and localization. Previous experience of some elements of the team in the RoboCup Simulation League has been highly relevant particularly in the design of the high-level coordination and control framework.

1 Introduction

CAMBADA[1] is the RoboCup middle-size league soccer team of the University of Aveiro, Portugal. This project started officially in October 2003 and, since then, the team has participated in four RoboCup competitions, namely, RoboCup'2004, RoboCup'2006, DutchOpen' 2006 and RoboCup'2007, and in the last four editions of the Portuguese Robotics Festival: Robotica2004, Robotica2005, Robotica2006 and Robotica2007. CAMBADA middle-size robotic soccer team won the 1st place in the Portuguese Robotics Festival'2007 and ranked 5th in the RoboCup World Championship'2007.

This paper describes the current development stage of the team and is organized as follows: Section 2 describes the general architecture of the robots focusing both on low-level control hardware aspects and on the general software architecture. Section 3 presents the current version of the vision system. Section 4 describes the high-level coordination and control framework and, finally, section 5 concludes the paper.

2 General Architecture of the Robots

The general architecture of the CAMBADA robots has been described in [1], [2], [11]. Basically, the robots architecture is centered on a main processing unit that is responsible for the higher-level behavior coordination, i.e. the coordination layer. This main processing unit (a PC) processes visual information gathered from the

[1] CAMBADA is an acronym of *Cooperative Autonomous Mobile roBots with Advanced Distributed Architecture*.

vision system, executes high-level control functions and handles the external communication with the other robots. This unit also receives sensing information and sends actuating commands to control the robot attitude by means of a distributed low-level sensing/actuating system. The PC runs the Linux operating system. The communication among team robots uses an adaptive TDMA transmission control protocol [3],[15] on top of IEEE 802.11b, that reduces the probability of transmission collisions between team mates thus reducing the communication latency. This transmission protocol is used to support a Shared Real Time Database (RTDB), which permits sharing selected state variables between the team mates.

The low-level sensing/actuation system (Fig. 1) is implemented through a set of microcontrollers interconnected by means of a network. For this purpose, Controller Area Network (CAN) [5], a real-time fieldbus typical in distributed embedded systems, has been chosen. This network is complemented with the FTT-CAN (Flexible Time-Triggered communication over CAN) [4],[6] higher-level transmission control protocol to enhance its real-time performance, composability and fault-tolerance. The low-level sensing/actuation system executes four main functions, namely, Motion control, Odometry, Kicking and System monitoring. The Motion control function provides holonomic motion using 3 DC motors. The Odometry function combines the encoder readings from the 3 motors and provides coherent robot displacement information that is then sent to the coordination layer. The Kick function includes the control of an electromagnetic kicker and of a ball handler to dribble the ball. Finally, the System monitor function monitors the robot batteries as well as the state of all nodes in the low-level layer.

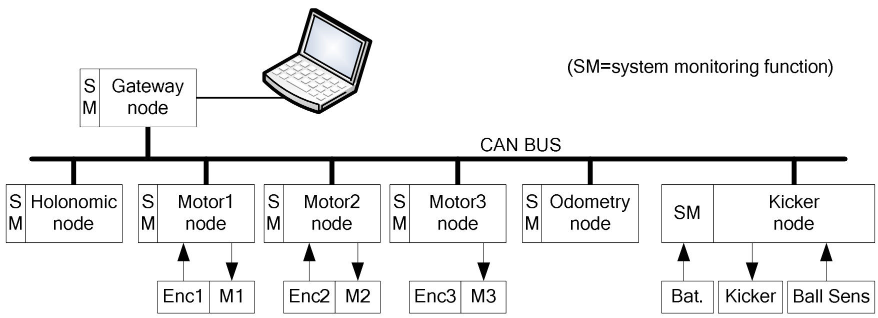

Fig. 1. The CAMBADA hardware architecture.

The low-level control layer connects to the coordination layer through a gateway, which filters interactions within both layers, passing through the information that is relevant across the layers, only.

The software system in each robot is distributed among the various computational units. High level functions are executed on the PC, while low level functions are executed on the microcontrollers. A cooperative sensing approach based on a Real-Time Database (RTDB) [1], [3], [7] has been adopted. The RTDB is a data structure where the robots share their world models. It is updated and replicated in all players in real-time.

The high-level processing loop starts by integrating perception information gathered locally by the robot, namely, information coming from the vision system and odometry information coming from the low-level layer. This information is afterwards stored in the local area of the RTDB. The next step is to integrate the robot local information with the information shared by team-mates, disseminated through the RTDB. The RTDB is then used by another set of processes that define the specific robot behavior for each instant, generating commands that are sent down to the low-level control layer.

3 Vision System

Some improvements have been made in the vision system, in particular the development of auto-calibration algorithms and the use of a hybrid vision system integrating an omni-directional and a perspective camera.

The omni-directional part of the vision system [13] is based on a catadioptric configuration implemented with a firewire camera and a hyperbolic mirror. We are using the camera in 640x480 RGB mode at 30 frames per second.

The perspective camera uses a low cost firewire web-camera (BCL 1.2 Unibrain camera with a ¼" CCD sensor and a 3.6mm focal distance lens) configured to deliver 640x480 YUV images at a rate of 30 frames per second.

The omnidirectional vision system is used to find the ball, the goals, detect the presence of obstacles and the white lines (used by the localization algorithm). The perspective vision is used to find the ball and obstacles in front of the robot at higher distances, which are difficult to detect using the omnidirectional vision system.

A set of algorithms have been developed to extract the color information of the acquired images and, in a second phase, extract the information of all objects of interest. To take advantage of the parallel processing capabilities of the hardware, the vision system main tasks, namely, image acquisition, color extraction, object detection and image visualization, are organized in separate processes which, when possible, are executed in parallel (Fig. 2). The implemented color extraction algorithm is based on lookup tables and the object detection in a radial model. The vision system is fast and accurate, having a processing time roughly independent of the environment around the robot.

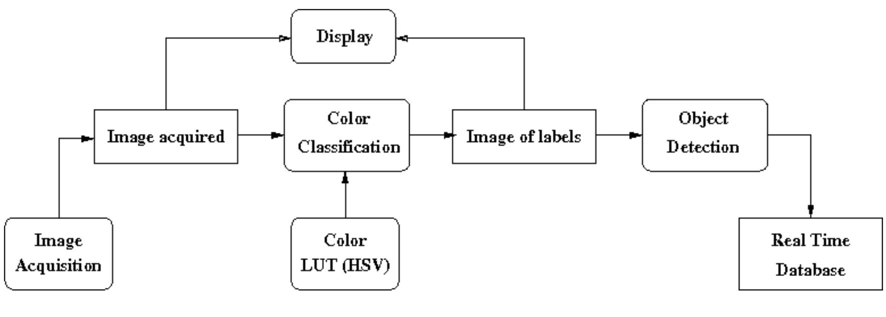

Fig. 2. Architecture of the vision system, applied both to the omnidirectional and perspective subsystem.

Image analysis in the RoboCup domain is simplified, since objects are color coded. This fact is exploited by defining color classes, using a look-up-table (LUT) for fast color classification. The table consists of 16777216 entries (24 bits: 8 bits for red, 8 bits for green and 8 bits for blue), each 8 bits wide, occupying 16 MB in total. The pixel classification is carried out using its color as an index into the table. The color calibration is done in HSV (Hue, Saturation and Value) color space. In the current setup the image is acquired in RGB or YUV format and is then converted to an image of labels using the appropriate LUT.

The image processing software uses radial search lines to analyze the color information. A radial search line is a line that starts in the center of the robot with some angle and ends in the limit of the image. The center of the robot in the omnidirectional subsystem is approximately in the center of the image. However, the center of the robot in the perspective subsystem is in the bottom of the image.

The regions of the image that have to be excluded from analysis (such as the robot itself, the sticks that hold the mirror and the areas outside the mirror) are ignored through the use of a previously generated image mask.

The objects of interest (a ball, obstacles and the white lines) are detected through algorithms that, using the color information collected by the radial search lines, calculate the object position and/or their limits in an angular representation (distance and angle). The position of the ball and the obstacle are stored in the RTDB.

The white lines are detected using an algorithm that, for each search line, finds the transition between green and white pixels. These detected white points are stored in the RTDB for later use by the robot self-localization process.

A set of algorithms have been also developed to perform the auto-calibration of the cameras. These algorithms use a white and a black area to calibrate the values of the white-balance, gain, exposure and brightness. Detailed information about these algorithms will be published soon. The experimental results obtained show the effectiveness of the algorithms, in particular its convergence independently of the original configuration of the cameras and the type of the environment light.

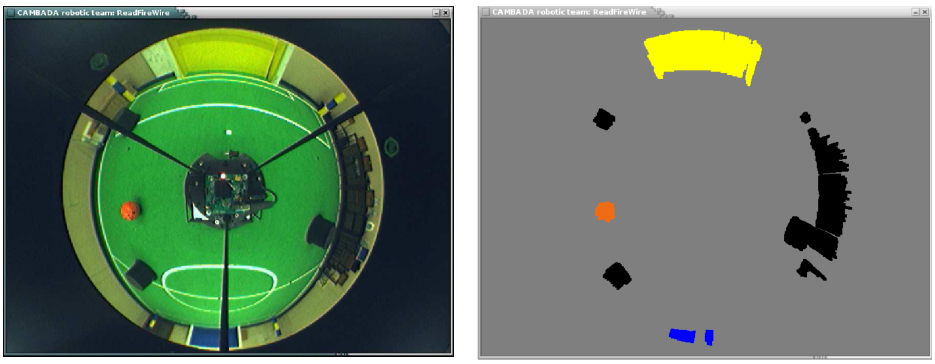

Fig. 3. An example of the blobs found in an acquired image. On the left, it is presented the original image. On the right, is it shown the color blobs found for that image. For each blob, we calculate useful information that is used later to calculate the position of each object.

3.1 Inverse Distance Map

The use of a catadioptric omni-directional vision system based on a regular video camera pointed at a hyperbolic mirror is a common solution for the main sensorial element found in a significant number of autonomous mobile robot applications. For most practical applications, this setup requires the translation of the planar field of view, at the camera sensor plane, into real world coordinates at the ground plane, using the robot as the center of this system. In order to simplify this non-linear transformation, most practical solutions adopted in real robots choose to create a mechanical geometric setup that ensures a symmetrical solution for the problem by means of single viewpoint (SVP) approach. This, on the other hand, calls for a precise alignment of the four major points comprising the vision setup: the mirror focus, the mirror apex, the lens focus and the center of the image sensor. Furthermore, it also demands the sensor plane to be both parallel to the ground field and normal to the mirror axis of revolution, and the mirror foci to be coincident with the effective viewpoint and the camera pinhole respectively. Although tempting, this approach requires a precision mechanical setup.

We developed a general solution to calculate the robot centered distances map on non-SVP catadioptric setups, exploring a back-propagation ray-tracing approach and the mathematical properties of the mirror surface [12]. This solution effectively compensates for the misalignments that may result either from a simple mechanical setup or from the use of low cost video cameras. Therefore, precise mechanical alignment and high quality cameras are no longer pre-requisites to obtain useful distance maps. The method can also extract most of the required parameters from the acquired image itself, allowing it to be used for self-calibration purposes.

In order to allow further trimming of these parameters, two simple image feedback tools have been developed.

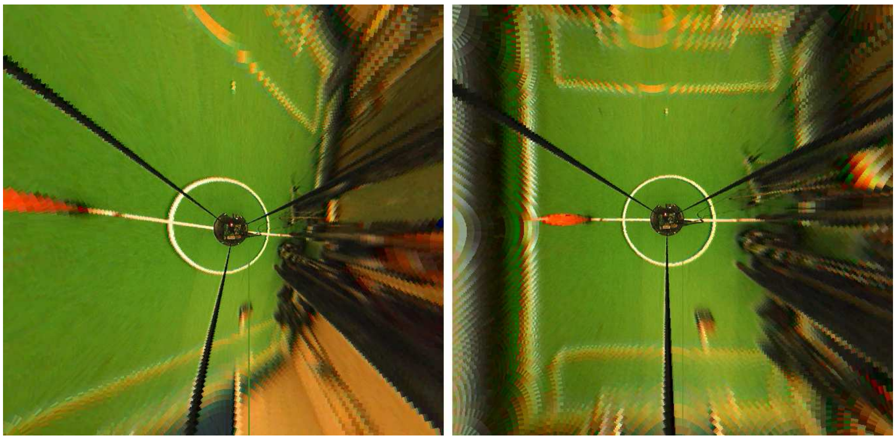

Fig. 4. Acquired image after reverse-mapping into the distance map. On the left, the map was obtained with all misalignment parameters set to zero. On the right, after automatic correction.

The first one creates a reverse mapping of the acquired image into the real world distance map. A fill-in algorithm is used to integrate image data in areas outside pixel

mapping on the ground plane. This produces a plane vision from above, allowing visual check of line parallelism and circular asymmetries (Fig. 4).
The second generates a visual grid with 0.5m distances between both lines and columns, which is superimposed on the original image. This provides an immediate visual clue for the need of possible further distance correction (Fig. 5). Since the mid-field circle used in this setup has exactly an outer diameter of 1m, incorrect distance map generation will be emphasized by grid and circle misalignment.

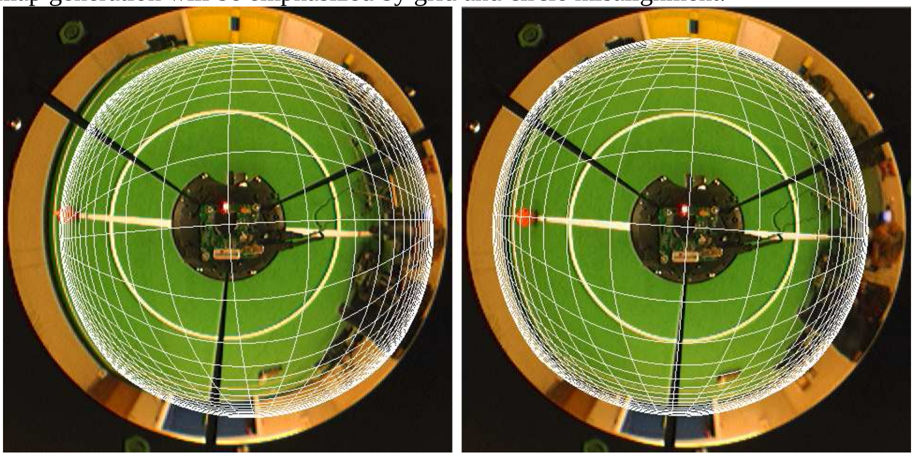

Fig. 5. A 0.5m grid, superimposed on the original image. On the left, with all correction parameters set to zero. On the right, the same grid after geometrical parameter extraction.

4 High-level coordination and control

The high-level decision is built around three main modules: sensor fusion, basic behaviors and high-level decision and cooperation. Monitoring of the whole team of robots is also one of the pursued lines of research [14]. The objective of the sensor fusion module is to gather the noisy information from the sensors and from other robots and update the RTDB database that will be used by the high-level decision and coordination modules. The basic behaviors module provides the set of primitives that the higher-level decision modules use to control the robot. It is essential to provide those modules with a good set of alternatives, each of which should be as efficient as possible. The high-level decision module is responsible for the analysis of the current situation and for the performing of decision-making processes carried out by each player in order to maximize, not only the performance of its actions, but also the global success of the team.

The sensor fusion module has recently been redesigned, in what concerns its interface with the other modules, in order to get a common view over all the sensor measures. Now all sensors write into adequate structures, but only the sensor fusion module is allowed to update the RTDB. A very important development has been the integration into the sensor fusion module of a self-localization lines-based engine, based-on the one described in [10], that allows a high level of confidence in the robots estimated self-position. In order to face the removal of the goal colors, which turned the field into a symmetric environment, an electronic compass has been included to setup the initial orientation of the robots.

The high-level decision module currently uses state-machine based modeled roles that switch the basic behavior of the robot in accordance with the current situation and the previous state. Coordination is achieved by the definition of formations of different roles [9] and by a higher-level module where role switching is performed. The concepts of roles, formations and set-plays have previously been used in the RoboCup by some Simulation and Middle-Size teams. The coordination is in the process of integrating the information coming from the new self-localization engine, which allows the use of coordination techniques like SBSP [8]. In some cases, such as kick-ins or corners, specific set-plays are activated where a coordinated and synchronized set of basic behaviors is performed by all team robots.

4.1 Communication-based Team Coordination

In this environment inter-robot communication and communication between base station and robots is a key issue, so that the team can maintain a coordinated behavior. Each robot uses part of the perception of the other robots, obtained through the RTDB, to improve its knowledge about the current positions and velocities of the other robots and of the ball. It is very important for our coordination model that each robot keeps an accurate estimation of the absolute position of the ball. The role assignment algorithm is based on the absolute position of both the robot and its teammates. The teammates' positions are not obtained through the vision system and rely completely on the communicated estimated self positions of others.

4.2 Multi Robot Ball Position Integration

The CAMBADA team is currently using a simple algorithm for Multi Robot Ball Position Integration. This is used to maintain an updated estimation of the ball position, whenever the vision subsystem is unable to detect the ball, and to validate robot's own ball perception when the vision subsystem detects a ball. When the agent doesn’t see the ball, it analyzes the ball information of playing teammates. The analysis consists in the calculation of the mean and standard deviation of all target ball positions, then discarding the values considered as outliers, and finally using the ball information of the teammate that has a shorter distance to ball.

4.3 Coordination Methodologies

Our coordination model is based on the definition of a strategy for a game, where each strategy may be composed of several tactics and each tactic defines a formation to be used at each situation. This model is merged with a role based coordination where different priorities are assigned to the different roles and positioning. All these items are maintained in a strategy configuration file to enable flexible changes to the current strategy. To maintain a correct formation all robots should have estimations of the ball absolute position obtained through the ball position integration method referred above.

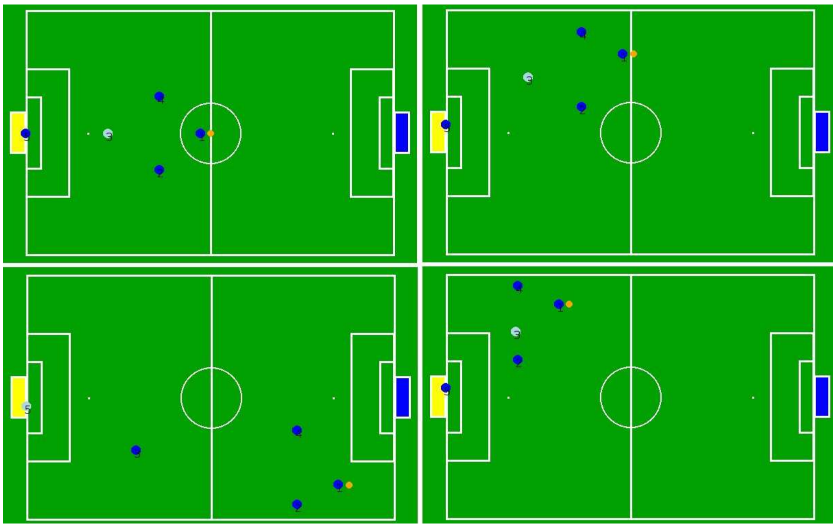

Fig. 6. Strategic positions for several different ball positions.

The role assignment algorithm is designed to support a varying number of active players in the team, resulting either from hardware or software malfunctioning or from referee orders. These are very common situations in the MSL.

5 Conclusion

This paper described the current development stage of the CAMBADA robots. Since the last submission of qualification material (in January/2007) several major improvements have been carried out, namely: the development of auto-calibration algorithms and the use of a hybrid vision system integrating an omni-directional and a perspective camera; the development of an analytical method to get the relationship between image pixels and real world distances and the re-design of the higher-level coordination and control software. These team improvements led to good results both in the Portuguese Robotics Open (1st place) and at RoboCup'2007 Atlanta (5th place). After RopoCup'2007 the development has been focused on perfecting the vision system and the high-level decision algorithms. CAMBADA development team currently includes 5 Msc. students.

References

1. L. Almeida, F. Santos, T. Facchinetti, P. Pedreiras, V. Silva, L.S.Lopes, "Coordinating distributed autonomous agents with a real-time database: The CAMBADA project". ISCIS'04, 19th International Symposium on Computer and Information Sciences. 27-29 October 2004, Kemer - Antalya, Turkey.
2. V. Silva, R. Marau, L. Almeida, J. Ferreira, M. Calha, P. Pedreiras, J. Fonseca, "Implementing a distributed sensing and actuation system: The CAMBADA robots case study", IEEE ETFA 2005, Catania, Italy. September 2005.
3. F. Santos, L. Almeida, P. Pedreiras, L.S.Lopes, T. Facchinnetti, "An Adaptive TDMA Protocol for Soft Real-Time Wireless Communication Among Mobile Computing Agents". WACERTS 2004, Workshop on Architectures for Cooperative Embedded Real-Time Systems (satellite of RTSS 2004). Lisboa, Portugal, 5-8 Dec. 2004.
4. Almeida L., P. Pedreiras, J. A. Fonseca, "The FTT-CAN Protocol: Why and How, IEEE Transactions on Industrial Electronics", 49(6), December 2002.
5. Controller Area Network - CAN2.0, Technical Specification, Robert Bosch, 1992.
6. Calha, M. J., J. A. Fonseca, “Approaches to the FTT-based scheduling of tasks and messages”, Proceedings of the 5th IEEE International Workshop on Factory Communication Systems (WFCS’04), Vienna, Austria, Sep/2004.
7. Pedreiras, P., F. Teixeira, N. Ferreira, L. Almeida, A. Pinho, F. Santos, "Enhancing the reactivity of the vision subsystem in autonomous mobile robots using real-time techniques", RoboCup Symposium: Papers and Team Description Papers, RoboCup-2005: Robot Soccer World Cup IX, Lecture Notes in Artificial Intelligence, Springer, 2006.
8. Reis, L.P. and N. Lau, "FC Portugal Team Description: RoboCup 2000 Simulation League Champion", RoboCup-2000, P. Stone, et al. (edtrs), p. 29-40, Springer Verlag, 2001.
9. Stone, P. and M. Veloso, "Task Decomposition, Dynamic Role Assignment and Low Bandwidth Communication for Real Time Strategic Teamwork", *Artificial Intelligence*, 110 (2), p. 241-273, 1999.
10. Martin Lauer, Sascha Lange and Martin Riedmiller, “Calculating the perfect match: An efficient and accurate approach for robot self-localisation”, In A. Bredenfeld, A. Jacoff, I. Noda and Y. Takahashi, editors, RoboCup 2005: Robot Soccer World Cup IX, LNCS. Springer, 2005
11. José Luís Azevedo, Manuel Bernardo Cunha, Luís Almeida (2007), Hierarchical Distributed Architectures for Autonomous Mobile Robots: a Case Study. ETFA2007- 12th IEEE Conference on Emerging Technologies and Factory Automation, pp. 973 - 980, Patras (Grece) September 25-28, 2007.
12. Bernardo Cunha, José Azevedo, Nuno Lau, Luís Almeida (2007), Obtaining the Inverse Distance Map from a Non-SVP Hyperbolic Catadioptric Robotic Vision System. RoboCup Symposium, Atlanta (USA) July 9-10, 2007 (poster).
13. António J. R. Neves, Gustavo Corrente, Armando Pinho (2007), An omnidirectional vision system for soccer robots. In Progress in Artificial Intelligence, edited by Springer Berlin / Heidelberg, Lecture Notes in Computer Science. Berlin, 2007.
14. João Figueiredo, Nuno Lau, Artur Pereira, Multi-Agent Debugging and monitoring framework, Proc. First IFAC Workshop on Multivehicle Systems (MVS'06), Brasil, October, 2006.
15. Frederico Santos, Gustavo Currente, Luis Almeida, Nuno Lau, Luis Seabra Lopes, Self-configuration of an Adaptive TDMA wireless communication protocol for teams of mobile robots, Proc. of the 13th Portuguese Conference on Artificial Intelligence, EPIA 2007, Guimarães, Portugal, December 2007.