CAMBADA'2016: Team Description Paper

R. Dias, F. Amaral, J. L. Azevedo, B. Cunha, P. Dias, N. Lau, A. J. R. Neves, E. Pedrosa, A. Pereira, J. Silva and A. Trifan

Intelligent Robotics and Intelligent Systems Lab
IEETA/DETI – University of Aveiro, Portugal

Abstract. This paper describes the CAMBADA Middle Size robotic soccer team for the purpose of qualification to RoboCup'2016. During the last year, improvements have been made in a significant number of components of the robots. The most important changes include improvements on ball and obstacles perception, obstacles avoidance, process synchronisation, motion model, software agent based on utilities that includes the use of set-plays, adaptive strategic positioning and dynamic passes.

1 Introduction

CAMBADA[1] is the RoboCup Middle Size League (MSL) soccer team of the University of Aveiro, Portugal. The project involves people working on several areas contributing for the development of all the components of the robot, from hardware to software.

The development of the team started in 2003 and a steady progress was observed since then. CAMBADA has participated in several national and international competitions, including RoboCup world championships (5^{th} place in 2007, 1^{st} in 2008, 3^{rd} in 2009, 2010, 2011, 2013 and 2014), the European RoboLudens, German Open (2^{nd} place in 2010), Dutch Open (3^{rd} place in 2012) and the annual Portuguese Robotics Open (3^{rd} place in 2006, 1^{st} in 2007, 2008, 2009, 2010, 2011, 2012 and 2^{nd} in 2013, 2014 and 2015). Moreover, the CAMBADA team achieved excellent results in the technical challenge of the RoboCup MSL: 2^{nd} place in 2008 and 2014, and 1^{st} place in 2009, 2012 and 2013. A 3^{rd} place in 2013, 2^{nd} place in 2012 and 2015, and 1^{st} place in 2011 and 2014 in the RoboCup Scientific Challenge were also achieved.

The general architecture of the CAMBADA robots has been described in [1, 2]. Basically, the robots follow a biomorphic paradigm, each being centered on a main processing unit (a laptop), which is responsible for the high-level behaviour coordination, i.e. the coordination layer. This main processing unit handles external communication with the other robots and has high bandwidth sensors, typically vision, directly attached to it. Finally, this unit receives low bandwidth sensing information and sends actuating commands to control the robot attitude by means of a distributed low-level sensing/actuating system.

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

This paper describes the current development stage of the team and is organized as follows: Section 2 briefly describes the hardware platform. Section 3 explains how the processes running on the laptop are synchronised. Section 4 addresses the world modelling and Section 5 describes the high-level coordination and control framework. Finally, Section 6 concludes the paper.

2 Current Platform

The current platform version reuses the model and functionalities that have proven to be efficient in the previous platform and introduces new changes in some aspects that require new approaches, namely the ability to move faster than 3 m/s top speed and the ability to actively control the ball in a more efficient way.

Some of the main issues that were addressed in the development of the platform include new omni-directional wheels and locomotion system. The custom made wheels are based on an aluminum 3 piece sandwich structure (see details in the mechanical drawings) in which 2 sets of 12 off-phase free rollers are supported. The traction system uses Maxon 150W DC motors and is based on synchronous belts and sprockets so that power is transmitted to the wheels through a synchronous belt system.

The ball handling mechanism is based on a double active handler similar to some of the solutions already presented by other teams, but uses omni wheels. Direction and speed of the ball interface rollers is closed loop controlled in order to ensure full compliance with current ball handling game rules.

The vision support system used to support the catadioptric mirror/camera consists of four titanium bars to interconnect the catadioptric set.

3 Process Synchronisation

In CAMBADA, processes synchronisation is achieved by using the library PMan (Process Manager). Figure 1 shows the previous state of the processes execution pipeline.

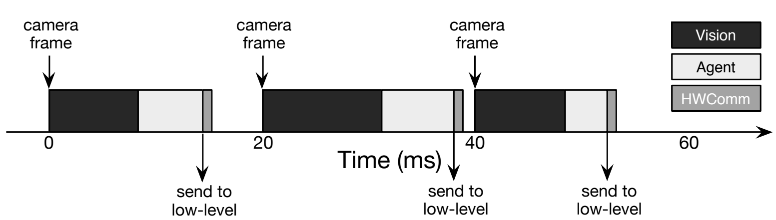


Fig. 1. The previous order of processes

In this pipelined execution chain, each process starts when the previous signals its finish. This way, the *HWComm* process execution timing is conditioned by the execution time of all the previous processes (*Vision* and *Agent*), that do not have a fixed execution time, increasing the temporal jitter on the low-level communication process. This has severe implications in the performance and tuning of high-level control loops as well as on some of the information reported by the hardware, since some values are reported as *delta*s (differences to last cycle), such as odometry measurements. Although this has little implications for the orders sent to the low level (velocity set-points, for example), the case of the information gathered from sensors is more critical. The jitter can eventually be taken into account when integrating the sensorial information, but it means that the resolution (and thus, the precision) of the low-level information changes between cycles. In one particular cycle, the information can be integrating over a longer period of time than in its consecutive cycle.

A proposed solution to this problem is presented in Figure 2. The *HWComm* cycle runs each time a new frame is received from the camera, an event that occurs each 20 ± 2 ms. The values sent to the low-level are the ones calculated by the agent in the previous cycle. In parallel, the vision process can take control of a different processor core and start analysing and processing the new image frame.

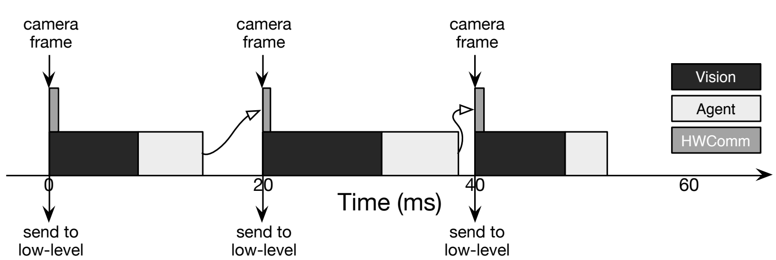


Fig. 2. The proposed order of processes

However, one constraint is introduced when compared with the previous solution: the total time of *vision* and *agent* together should not exceed the cycle period in order for the *HWComm* to send new values for the low-level hardware every cycle. In the pipeline approach, it was possible to have temporal overlapping on the processes, i.e. a new frame could arrive while the *agent* was still running. In this new approach, however, because the sending to the low-level is triggered by each new incoming camera frame, the *agent* must finish its cycle before this occurs, otherwise, the information sent to the low-level is the same as the previous cycle.

4 World modeling

Several improvements to the world model have been or are being made. This includes improvements on the ball and obstacles perception, and obstacles avoidance.

In terms of obstacles perception [3], methodologies are being developed for obstacle tracking for persistent representation in the world-state. This model represents the global information of the obstacles on the field, rather than an individual perspective of each robot. This representation will be used by the utility map, as described later.

The reactive component of the obstacle avoidance algorithm continues to be improved in order to reduce the probability of crash, or even touch, with the opponents or between robots from the same team. The system relies on a set of fully configurable virtual sonars, based on a set of parametric values, and is supported by the information provided by the vision system. This allows the use of different sonar configurations according to each particular game situation, their dynamic change according to the robot velocity and the evaluation of robot dynamics to anticipate feasible movements.

4.1 Kinect 3D Aerial Ball Detection

The current vision system of the CAMBADA robots is based on an omnidirectional setup described in [4]. The vision system has suffered several improvements in the last years, namely an algorithm for the self-calibration of the colormetric parameters of a digital camera [5] has been presented and a computer vision library for color object detection has been implemented [6]. In this section, we introduce an algorithm for the 3D detection of aerial balls using a Kinect sensor.

The pipeline of the vision system of our goalkeeper is presented in Fig. 3.

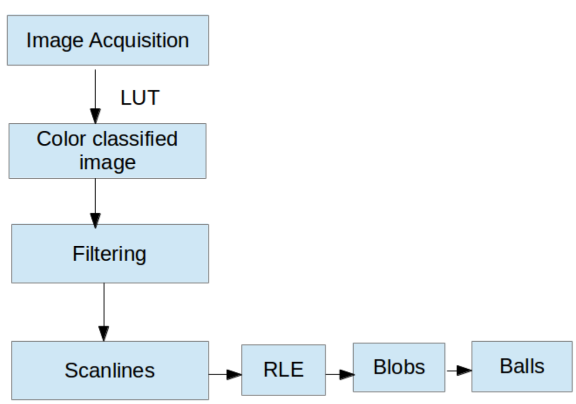
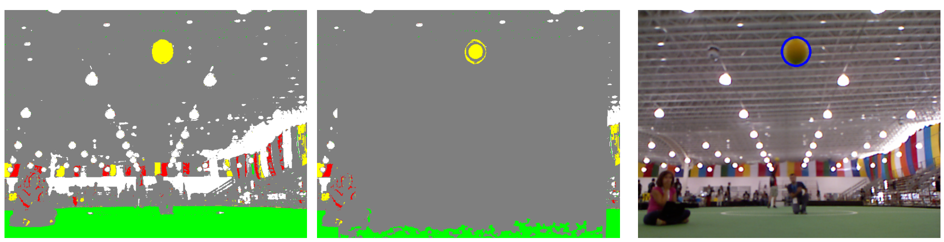


Fig. 3. Vision sytem pipeline of CAMBADA goalkeeper.

The first step of the algorithm is the detection of blobs of the ball color, using the UAVision [6] library. After performing a color segmentation on the input image using a look-up table (LUT), we apply a filter based on depth information in order to remove the color classification of objects that are outside the soccer field. Scan-lines are used for searching pixels of the color of interest (the color of the soccer ball). When scanning the image in search of the color of interest, the relevant found information is saved using a run-length encoding approach. The run-length information is used for forming blobs or clusters of the color of interest. These blobs have to pass a validation process in order to establish if a given blob is a ball. The validation procedure is based on calculating different features for each of the found blobs, such as the bounding box area, the circularity, and width-height relation.

The depth information from the Kinect sensor is used for discarding the color of the objects that are found farther than a certain distance (in this case, 7m were considered). This complements the previous step by filtering possible objects of the ball color found outside the field. As stated before, this step is applied after the color classification. A calibration between the RGB and depth images provided by the sensor have to be performed [7]. A result of the algorithm is presented in Fig. 4.

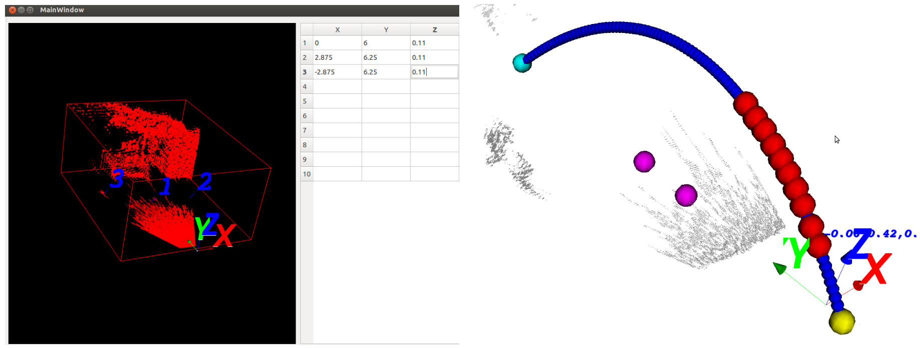


a) b) c

Fig. 4. On the left, the original classified image. In the center, the obtained color classified image after filtering. On the right, the original image with the ball correctly detected.

For the calibration of the Kinect sensor, an application based on [8] (see Fig. 5(a)) acquires on demand an image from Kinect and then allows the user to pick some points on the 3D cloud of points. The chosen points correspond to points in the world whose relative position to the robot are known by the user. The software then evaluates the rigid body transform between the 2 coordinates systems corresponding to the position of the Kinect and its orientation relatively to the origin of the robot coordinates system.

For the estimation of the ball trajectory, we use the algorithm described in [8]. Having the trajectory calculated, the goalkeeper can estimate the best position to intercept the ball using the projection of the trajectory on the ground, determined by the two magenta spheres drawn on Fig. 5(b).



a) b)

Fig. 5. On the left, the application for Kinect position calibration with 3 reference points on the Kinect cloud and the corresponding coordinates in robot coordinates system [8]. On the right, the calculated trajectory of the ball. The positions of the balls used for the computation are presented as large red spheres and the parabolic trajectory estimated is represented by the small blue spheres. The magenta spheres represent the projection on the ground of the detected balls.

4.2 Multi-Robot Aerial 3D Ball Detection

When detecting the ball in 3D space, an alternative to 3D sensors like Kinect is to use cooperative sensing, by triangulating the position of several omni-vision cameras from different robots. This research project has been done in the scope of an internship of one student of team Tech United in IRIS-Lab.

There were three main challenges addressed: first, the lines of sight from the robots might not intersect; also, the algorithm has to deal with network delay and possible packet loss in inter-robot communications; and finally, the amount of transmitted information is limited (agent should not expect new information from team-mates every cycle).

A data buffer is used to store information from the teammates and a Kalman filter that combines the (non-chronological) measurements with an hybrid ball model. An example is shown in figure 6, using two robots sharing information at 20Hz, each running the agent at 50Hz asynchronously.

5 High-level coordination

The CAMBADA agent software architecture allows high reconfigurability and fast prototyping of behaviours. In the context of MSL, with a highly dynamic environment, there is a growing need of predicting the near future, since making decisions based only on the very last available information is not very effective. This occurs either due to the delays in inter robot communication or because of the fast moving opponents and ball. The current software architecture model allows the agent not only to make decisions based on past and present information, but also on predicted states.

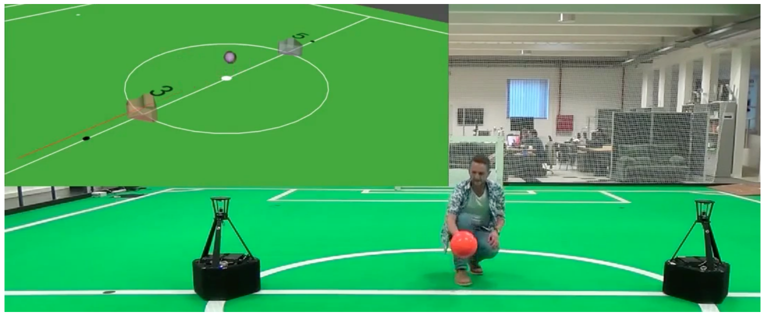


Fig. 6. Demo of the 3D ball positioning system using real robots

This software architecture streamlines the algorithm development of the various roles, by providing the agent an array of different choices in advance, each with some prior conditions and a given priority.

Furthermore, the agent is evolving to an hybrid model, which makes decisions based not only on simple conditions but also on a set of utilities, each one testing the expected success with a different option.

Additionally, the CAMBADA team uses a coach that allows the choice, in real time, of the best formation for the robots, based on a set of rules that evaluates several game statistics and the game state.

5.1 Adaptive Strategic positioning

The current version of the CAMBADA agent architecture supports spatial utility maps [9] that are used to decide the best position to occupy on the field, depending on the game state. This leads to more dynamic positions in relation to the opponents, and not only to the ball. These utility maps take into consideration the opponents and the ball positions as well as other constraints, such as field of view and game rules restrictions. These maps can be used to extract the most advantageous position, closest to the strategic position defined by SBSP or DT (as presented in the last years), for the robot in a certain moment.

6 Conclusions

This paper describes the current development stage of the CAMBADA robots, both in the hardware platform and at the software level.

Several improvements have or are being carried out, namely in ball and obstacles perception, obstacles avoidance, process synchronisation, motion model, kicking behaviour, a new model for the software agent based on utilities that includes the use of set-plays, adaptive strategic positioning and dynamic passes.

References

1. A. Neves, J. Azevedo, N. Lau B. Cunha, J. Silva, F. Santos, G. Corrente, D. A. Martins, N. Figueiredo, A. Pereira, L. Almeida, L. S. Lopes, and P. Pedreiras. *CAMBADA soccer team: from robot architecture to multiagent coordination*, chapter 2. I-Tech Education and Publishing, Vienna, Austria, In Vladan Papic (Ed.), Robot Soccer, 2010.
2. José Luís Azevedo, Bernardo Cunha, and Luís Almeida. Hierarchical Distributed Architectures for Autonomous Mobile Robots: a Case Study. In *ETFA2007- 12th IEEE Conference on Emerging Technologies and Factory Automation*, volume 1-3, pages 973–980, Patras, Greece, September 2007.
3. Joao Silva, Nuno Lau, António JR Neves, Joao Rodrigues, and José Luís Azevedo. Obstacle detection, identification and sharing on a robotic soccer team. In *Progress in Artificial Intelligence*, pages 350–360. Springer, 2009.
4. António J. R. Neves, Armando J. Pinho, Daniel A. Martins, and Bernardo Cunha. An efficient omnidirectional vision system for soccer robots: from calibration to object detection. *Mechatronics*, 21(2):399–410, mar 2011.
5. António J. R. Neves, Alina Trifan, and Bernardo Cunha. Self-calibration of colormetric parameters in vision systems for autonomous soccer robots. In *RoboCup 2013 Symposium*, volume LNCS 8371, Eindhoven, Netherland, September 2013.
6. Alina Trifan, António J. R. Neves, and Bernardo Cunha. UAVision: A modular time-constrained vision library for soccer robots. In *RoboCup 2014 Symposium*, volume LNAI 8992, Joao Pessoa, Brazil, July 25 2014.
7. António J. R. Neves, Alina Trifan, Paulo Dias, and José Luís Azevedo. Detection of Aerial Balls in Robotic Soccer Using a Mixture of Color and Depth Information. In *Proceedings of the 2015 IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC 2015)* , April 2015.
8. Paulo Dias, João Silva, Rafael Castro, and António J. R. Neves. Detection of aerial balls using a Kinect sensor. In *RoboCup 2014 Symposium*, volume LNAI 8992, Joao Pessoa, Brazil, July 25 2014.
9. António J. R. Neves, Filipe Amaral, Ricardo Dias, João Silva, and Nuno Lau. A New Approach for Dynamic Strategic Positioning in RoboCup Middle-Size League. In *Progress in Artificial Intelligence*, volume LNAI 9273 of *Lecture Notes in Computer Science*, pages 433–444. Springer International Publishing, 2015.