

CAMBADA’2017: Team Description Paper

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

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’2017. During the last year, improvements have been made in a significant number of components of the robots. The most important changes include improvements on the goal-keeper hardware, perception of the white lines, ball and obstacles, global obstacle tracking and on development and debugging tools.

1 Introduction

CAMBADA¹ 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 (5th place in 2007, 1st in 2008, 3rd in 2009, 2010, 2011, 2013, 2014 and 2016), the European RoboLudens, German Open (2nd place in 2010), Dutch Open (3rd place in 2012) and the annual Portuguese Robotics Open (3rd place in 2006, 1st in 2007, 2008, 2009, 2010, 2011, 2012 and 2nd in 2013, 2014, 2015 and 2016). Moreover, the CAMBADA team achieved excellent results in the technical challenge of the RoboCup MSL: 2nd place in 2008 and 2014, and 1st place in 2009, 2012 and 2013. A 3rd place in 2013, 2nd place in 2012 and 2015, and 1st place in 2011, 2014 and 2016 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 changes. Section 3 explains the improvements on the perception level (on white lines, ball and obstacles detection). Section 4 addresses the implementation of a global obstacle tracker with multi-robot observations. Section 5 summarizes the refinements on development and debugging tools. Section 6 describes on-going work on the communications framework. And finally, Section 7 concludes the paper.

2 Current Platform

The CAMBADA team uses the same platform version since 2013, which has been thoroughly described in previous Team Description Papers. During 2016, the hardware improvements were aimed at the goal-keeper.

2.1 Goal-keeper

The CAMBADA goal-keeper hardware has been upgraded to cope with higher demands for a more efficient goal blocking on the league. Firstly, the frame is now extendable to defend higher shots, which copes with the current MSL rules - it extends a maximum of 10 cm for maximum of 1 second and waits for at least 4 seconds.

Moreover, the goalkeeper was equipped with the latest generation Microsoft Kinect, which tripled the resolution of the color image and also increased the resolution of the depth camera, while maintaining the 30 frames per second rate.

To cope with the requirements of the Kinect camera (USB3 and dedicated GPU), an NVidia Jetson TK2 was also included to deal with image processing of the Kinect.

3 Perception

There has been an effort in improving the robots perception capabilities. The MSL has in its roadmap the intention of increasing the field size. If the maximum height of the robots does not increase, this poses an additional challenge for localization, since it becomes even more difficult to detect field lines at long distances. Therefore, we worked on a new method for efficient line detection at long distances. Furthermore, there were improvements on the detection of both the ball and the obstacles, to account for partial mutual occlusions.

3.1 Line Detection

Previous line detection was performed using virtual scan-lines that detected sequential transitions of “green-white-green”, using *HSV* (Hue, Saturation, Value) color ranges segmentation. However, on further distances, the algorithm fails due to saturation of the *V* (Value) channel on the *HSV* transformation.

In order to overcome the issue of the V saturation, a new method is being used to detect white lines where the HSV method fails. It consists in a transformation from RGB to HSY' (instead of HSV):

$$Y' = 0.21R + 0.72G + 0.07B \quad (1)$$

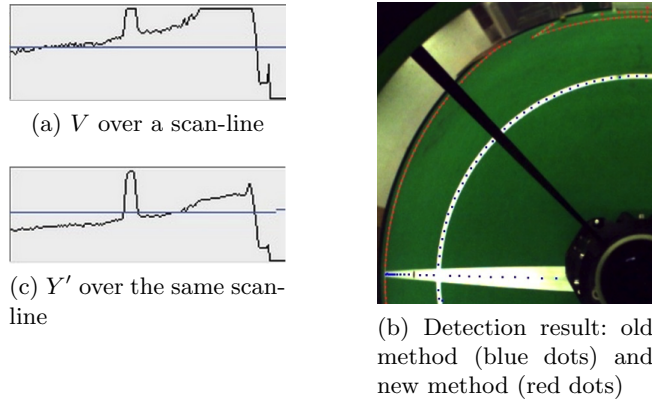


Fig. 1: New white line detection method

Fig. 1 contains an example of a scan-line detection. Using only color segmentation, the previous method failed to detect the lines at further distances, due to the saturation of the V channel (Fig. 1a). Fig. 1c shows the value of Y' for the same scanline. A peak-detection algorithm is then used in conjunction with color to detect the white-line feature at longer distances, thus improving the localization by providing more valid points to the localization algorithm.

3.2 Ball

Until 2016, the ball detection algorithm used by the CAMBADA robots was based on color segmentation (based on HSV color ranges) and feature extraction with heuristics for blob validation based on the blob size on the 2D image. A shortcoming of this approach is that the color segmentation has low resilience to changes in ambient lighting and, furthermore, heuristics can easily fail when the ball is partially occluded - the blob size is reduced, and the center of the ball has an offset with respect to the center of the blob.

Last year, we introduced two extra steps to further improve the perception of the ball: **edge detection** and **circle fitting**. After the pixels corresponding to the edges of the ball are detected (using dynamic scan-lines that are created in the center of the blob), a RANSAC algorithm is run to fit the “best” circle in those points.

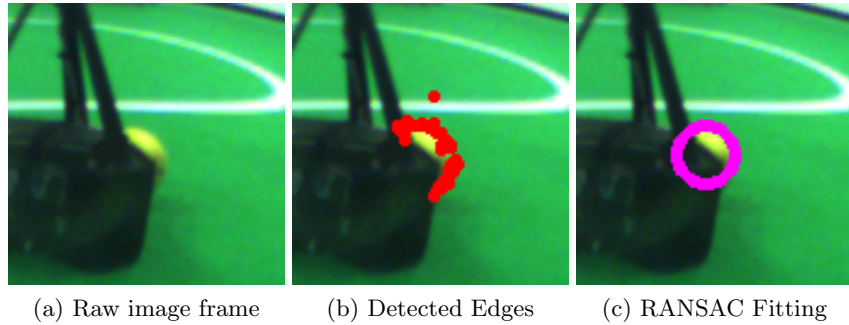


Fig. 2: Partial ball occlusion example.

Fig. 2 shows a situation of a ball that is partially occluded and can be detected with a promising accuracy using this method. The circle fitting using RANSAC performs well in the presence of ball edge outliers, such as in the example above.

3.3 Obstacles

Reliable and fast detection of obstacles is a critical issue to ensure safety and integrity of robots and humans. By correctly identifying obstacles around it, each robot is able to plan a trajectory and move around the field while avoiding any physical contact.

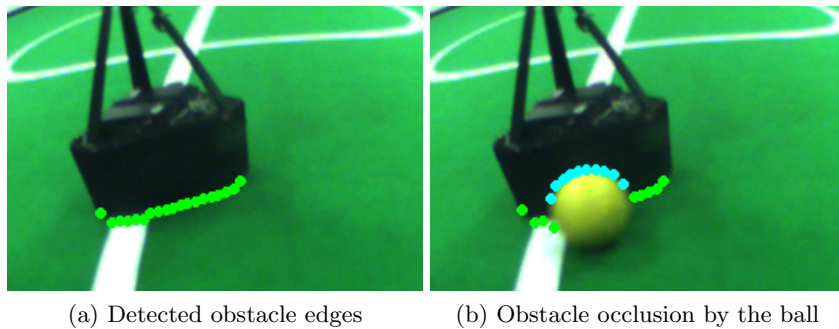


Fig. 3: Obstacle detection with partial ball occlusion.

Obstacle detection is based on radial scan-lines, starting in the center of the robot (near the center of the image) to the outer part of the mirror. Color segmentation is used to detect transitions such as “green-black” and “white-black” - Fig. 3a.

These detected points are assumed to be on the ground, and are then transformed from pixels to the local coordinate system of the robot (in meters). However, when the ball passes by an obstacle so that it becomes partially occluded, the detected obstacle points are above the ball (Fig. 3b), which breaks the previous assumption. The (undesired) result is that after clustering all points, the obstacle position is virtually projected further away from the observer robot, inducing an inexistent movement of the obstacle, when it could be stopped.

In order to deal with this special case, the points that are detected after a ball are marked by the detection algorithm and the relative distance of the obstacle is adjusted according to the median of the distances of the other points (not above the ball).

3.4 GPU-Based Vision System

The aforementioned vision system implementation performs most of its image processing operations using the CPU, except the goal-keeper, in which a GPU is used to process the distance data produced by the Kinect sensor. A new version of this system is currently being implemented aiming to use the GPU for all the image processing operations. This is expected to improve the vision processing performance, which in turn impacts the overall performance of the processes pipeline.

4 Global Obstacle Tracker

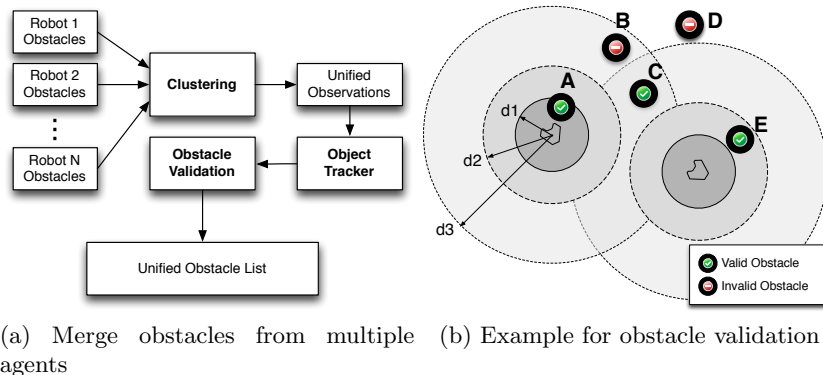


Fig. 4: Multi-object tracking with distributed sensing.

One of the main research goals on distributed autonomous agents in a Multi-Agent System is the development of mechanisms to form a better world model using information merging from different agents. Our newly developed object

tracker module [3] receives new observations each vision cycle and tries to match these observations with the current tracks. We assumed that each track can be modelled as a Gaussian process in the 2D space, so each uses a Kalman Filter.

Fig. 4a shows an overview of our strategy to merge information from multiple agents to form a unified representation of the obstacles spread around the field. We consider the shared obstacles of each robot as an observation. We implemented a Constrained K-Means clustering method to cluster observations belonging to the same object and then applied the multi-object tracker over the clusters. Finally, a validation is performed over the unified tracks (Fig. 4b).

5 Development and Debugging Tools

5.1 Basestation Log and Video In-Sync Replay

The CAMBADA Basestation is the application that is used to visualize different sorts of information regarding the CAMBADA team. This includes robots pose, velocities, battery state, temperature, role, behaviour, debug information, perception data (such as the ball and obstacles) per robot, etc.

The application holds an on-line replay mechanism for development purposes, that allows the user to stop the robots and carefully analyse the previous situation(s) frame-by-frame.

During the matches, this application is configured to log all data to be replayed after the match ends. Furthermore, our Basestation has been improved with a video player that can load a video file and playback the video perfectly synchronized with the log-data. The outcome is a tool that improves the ability to visually compare the worldstate info with real footage of the field, frame-by-frame, thus minimizing the time needed for off-line debugging.

5.2 Benchmarking and Statistical Analysis

In the standard log information collected by the Referee Box lies the key to create better debugging and benchmarking tools. These will help to develop and test the changing software components on the robots more systematically. Additionally, the creation of these tools will enable offline analysis of the opponent team strategy, allowing for a better adaptation to that strategy in following matches.

6 Communications Framework

The RtDB is a data structure responsible for sharing the world models of each robot. The data is updated and replicated to all players in real-time using schemas that were previously defined to grant that every player knows what is going to receive or what they need to send. There are several on-going improvements to make the RtDB more efficient, allowing it to send dynamic containers, thus avoiding sending unnecessary information. The information received in the RtDB will be more tolerant allowing the data to be parsed even when there are missing or unexpected fields.

7 Conclusions

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

Several improvements have been carried out on the goal-keeper hardware, in ball and obstacles perception, line detection, obstacle tracking and debugging tools. Furthermore, some on-going work includes enhancement on the vision processing performance using the GPU, on benchmarking tools and also on the communications framework.

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. *CMBADA 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. Ricardo Dias, Nuno Lau, Joao Silva, and Gi Hyun Lim. Multi-object tracking with distributed sensing. In *Proc. of the 2016 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI 2016)*, 2016.