Sensor and information fusion applied to a Robotic Soccer Team

João Silva, Nuno Lau, João Rodrigues, José Luís Azevedo and António J. R. Neves

IEETA / Department of Electronics, Telecommunications and Informatics
University of Aveiro, Portugal
{joao.m.silva,nunolau,jmr,jla,an}@ua.pt

Abstract. This paper is focused on the sensor and information fusion techniques used by a robotic soccer team. Due to the fact that the sensor information is affected by noise, and taking into account the multi-agent environment, these techniques can significantly improve the accuracy of the robot world model. One of the most important elements of the world model is the robot self-localisation. Here, the team localisation algorithm is presented focusing on the integration of visual and compass information. To improve the ball position and velocity reliability, two different techniques have been developed. A study of the visual sensor noise is presented and, according to this analysis, the resulting noise variation depending on the distance is used to define a Kalman filter for ball position. Moreover, linear regression is used for velocity estimation purposes, both for the ball and the robot. This implementation of linear regression has an adaptive buffer size so that, on hard deviations from the path (detected using the Kalman filter), the regression converges more quickly. A team cooperation method based on sharing of the ball position is presented. Besides the ball, obstacle detection and identification is also an important challenge for cooperation purposes. Detecting the obstacles is ceasing to be enough and identifying which obstacles are team mates and opponents is becoming a need. An approach for this identification is presented, considering the visual information, the known characteristics of the team robots and shared localisation among team members. The same idea of distance dependent noise, studied before, is used to improve this identification. Some of the described work, already implemented before RoboCup2008, improved the team performance, allowing it to achieve the 1st place in the Portuguese robotics open Robótica2008 and in the RoboCup2008 world championship.

1 Introduction

Robotic soccer is nowadays a popular research domain in the area of multi robot systems. RoboCup[1] is an international joint project to promote artificial intelligence, robotics and related fields that includes several leagues, each one with a different approach, some only at software level, others at hardware, with single or multiple agents, cooperative or competitive [1].

In the context of RoboCup, the Middle Size League (MSL) is one of the most challenging. In this league, each team is composed of up to 6 robots with maximum size

[1] http://www.robocup.org/

of 50x50cm base, 80cm height and a maximum weight of 40Kg, playing in a field of 18x12m. The rules of the game are similar to the official FIFA rules, with required changes to adapt for the playing robots [2]. Each robot is autonomous and has its own sensorial means. They can communicate among them, and with an external computer acting as a coach, through a wireless network. This coach computer cannot have any sensor, it only knows what is reported by the playing robots. The agents should be able to evaluate the state of the world and make decisions suitable to fulfil the cooperative team objective.

Fig. 1. Picture of the team robots.

CAMBADA, *Cooperative Autonomous Mobile roBots with Advanced Distributed Architecture*, is the Middle Size League Robotic Soccer team from Aveiro University. The project started in 2003, coordinated by the IEETA[2] ATRI[3] group and involves people working on several areas for building the mechanical structure of the robot, its hardware architecture and controllers and the software development in areas such as image analysis and processing, sensor and information fusion, reasoning and control.

To be able to accomplish the objective of playing soccer, it is important that the agent is able to build a good representation of its environment. In the CAMBADA team, this process is called integration. It is a step executed after image analysis and is responsible to take raw information from the vision and other robot sensors and make a sensor and information fusion of all the sources, estimating reliable information of the elements on the field (e.g.: self-localisation, ball position and velocity, obstacles).

For that task it may use the values stored in the previous representation, the current sensor measures (eventually after pre-processing) that has just arrived, the current actuator commands and also information that is available from other robots sensors or world state. This is essentially an information fusion problem. The most common methods to tackle information fusion are based on probabilistic approaches, including Bayes rule, Kalman filter and Monte Carlo methods [3].

All the information available from the sensors in the current cycle is kept in specific data structures (Fig. 2), for posterior fusion and integration, based on both the current information and the previous state of the world.

[2] Instituto de Engenharia Electrónica e Telemática de Aveiro - Aveiro's Institute of Electronic and Telematic Engineering

[3] Actividade Transversal em Robótica Inteligente - Transverse Activity on Intelligent Robotics

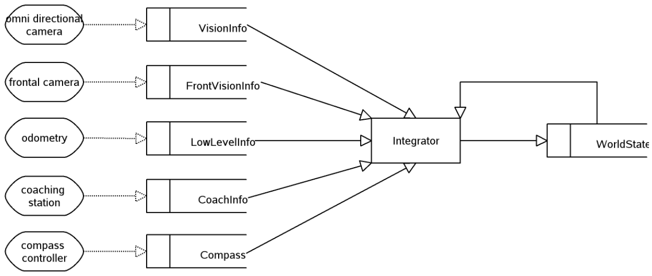


Fig. 2. Integrator functionality diagram.

This paper focuses on the description of some sensor and information techniques used in the CAMBADA team. Section 2 describes the fusion of sensorial data for self-localisation. The several aspects of ball integration are described in Section 3. Section 4 presents solutions for identification of visually detected obstacles. Finally, Section 5 presents the conclusion and team achievements.

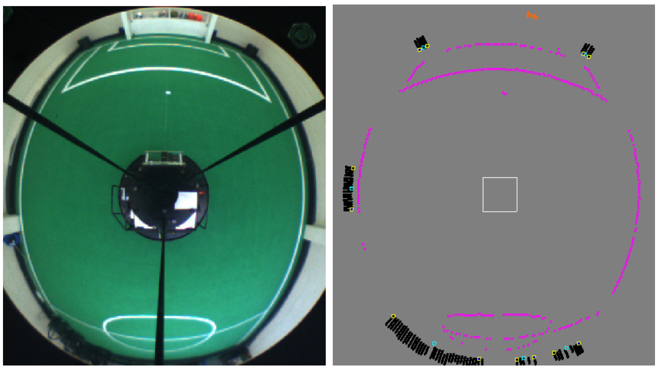
2 Localisation

Self-localisation of the agent is an important issue for a soccer team, as strategic moves and positioning must be defined by positions on the field. In the MSL, the environment is completely known, as every agent knows exactly the layout of the game field. Given the known mapping, the agent has then to locate itself on it.

The CAMBADA team localisation algorithm is based on the detected field lines, with fusion information from the odometry sensors and an electronic compass. It is based on the approach described in [4], with some adaptations. It can be seen as an error minimisation task, with a derived measure of reliability of the calculated position so that a stochastic sensor fusion process can be applied to increase the estimate accuracy [4].

From the centre of the image (the centre of the robot), radial sensors are created around the robot, each one represented by a line with a given angle. These are called *scanlines*. The image processing, in each cycle, returns a list of positions relative to the robot where the *scanlines* intercept the field line markings [5]. The idea is to analyse the detected line points, estimating a position, and through an error function describe the fitness of the estimate. This is done by reducing the error of the matching between the detected lines and the known field lines (Fig. 3). The error function must be defined considering the substantial amount of noise that affect the detected line points which would distort the representation estimate [4].

Although the odometry measurement quality is much affected with time, within the reduced cycle times achieved in the application, consecutive readings produce acceptable results and thus, having the visual estimation, it is fused with the odometry values to refine the estimate. This fusion is done based on a Kalman filter for the robot position estimated by odometry and the robot position estimated by visual information. This approach allows the agent to estimate its position even if no visual information is available. However, it is not reliable to use only odometry values to estimate the posi-



a) b)

Fig. 3. Captures of an image acquired by the robot camera and processed by the vision algorithms. Left **a)**: the image acquired by the camera; Right **b)**: the same image after processing with magenta dots over the detected field lines.

tion for more than a very few cycles, as slidings and frictions on the wheels produce large errors on the estimations in short time.

The visually estimated orientation can be ambiguous, i.e. each point on the soccer field has a symmetric position, relatively to the field centre, and the robot detects exactly the same field lines. To disambiguate, an electronic compass is used. The orientation estimated by the robot is compared to the orientation given by the compass and if the error between them is larger than a predefined threshold, actions are taken. If the error is really large, the robot assumes a mirror position. If it is larger than the acceptance threshold, a counter is incremented. This counter forces relocation if it reaches a given threshold. Fig. 4 shows situations where the threshold was reached and relocalisation was forced after some cycles.

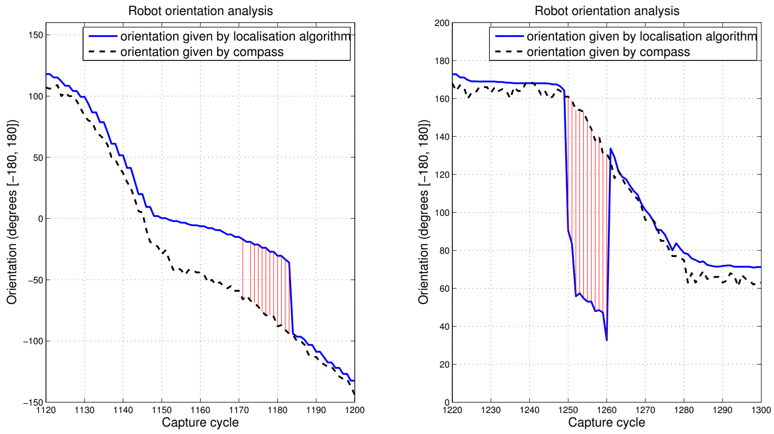
3 Ball integration

Within RoboCup several teams have used Kalman filters for the ball position estimation [6,7,8,9]. In [9] and [8] several information fusion methods are compared for the integration of the ball position using several observers. In [9] the authors conclude that the Kalman reset filter shows the best performance.

The information of the ball state (position and velocity) is, perhaps, the most important, as it is the main object of the game and is the base over which most decisions are taken. Thus, its integration has to be as reliable as possible. To accomplish this, a Kalman filter implementation was created to filter the estimated ball position given by the visual information, and a linear regression was applied over filtered positions to estimate its velocity.

3.1 Ball position

It is assumed that the ball velocity is constant between cycles. Although that is not true, due to the short time variations between cycles, around 40 milliseconds, and given



a) **b)**

Fig. 4. Illustration of two situations where relocalisation was forced. Left **a)**: the camera was covered while the robot moved. The estimated orientation error degrades progressively and after getting higher than the threshold, the cycle count starts and forces relocation; Right **b)**: the robot tilted. The estimated orientation error is immediately affected by more than threshold and the cycle count starts and forces relocation.

the noisy environment and measurement errors, it is a rather acceptable model for the ball movement. Thus, no friction is considered to affect the ball, and the model doesn't include any kind of control over the ball. Therefore, given the Kalman filter formulation (described in [10]), the assumed state transition model is given by

$$X_k = \begin{bmatrix} 1 & \Delta T \\ 0 & 1 \end{bmatrix} X_{k-1}$$

where X_k is the state vector containing the position and velocity of the ball. Technically, there are two vectors of this kind, one for each cartesian dimension (x,y). This velocity is only internally estimated by the filter, as the robot sensors can only take measurements on the ball position. After defining the state transition model based on the ball movement assumptions described above and the observation model, the description of the measurements and process noises are important issues to attend. The measurements noise can be statistically estimated by taking measurements of a static ball position at known distances (Fig. 5).

The standard deviation of those measurements is used to calculate the variance and thus define the measurements noise parameter. In practice, the measurements of the static ball were taken while the robot was rotating over itself, to simulate movement and the trepidation it causes, so that the measurements were as close to real game conditions as possible. Some of the results are illustrated in Fig. 5.

A relation between the distance of the ball to the robot and the measurements standard deviation is modeled by the 2nd degree polynomial best fitting the data set in a

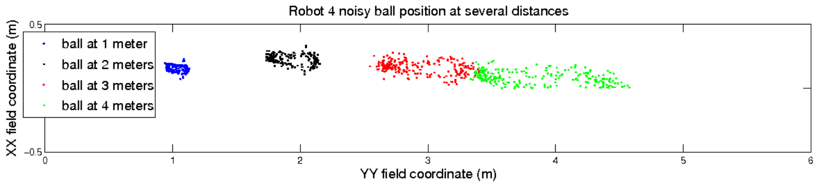


Fig. 5. Noisy position of a static ball taken from a rotating robot.

least-squares sense (Fig. 6). A 1st degree polynomial does not fit the data properly, and assumes negative values for positive distance, which is not acceptable. Given the few known points, a 3rd degree polynomial would perfectly fit all 4 of them. However, these known points are also estimated and thus cannot be taken as exact. For that reason, a curve that would exactly fit them is not desirable.

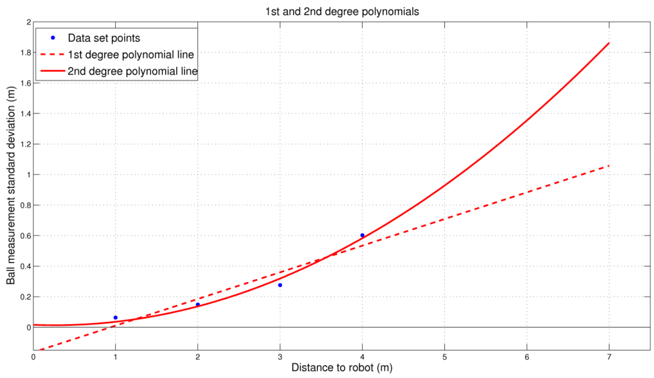


Fig. 6. Representation of the standard deviation value for variable distance to the robot. Data set points as blue dots. 1st degree polynomial as dashed line, 2nd degree polynomial as solid line.

As for the process noise, this is not trivial to estimate, since there is no way to take independent measurements of the process to estimate its standard deviation. The process noise is represented by a matrix containing the covariances correspondent to the state variable vector.

Empirically, one could verify that forcing a near null process noise causes the filter to practically ignore the read measures, leading the filter to emphasise the model prediction. This makes it too smooth and therefore inappropriate. On the other hand, if it is too high, the read measures are taken into too much account and the filter returns the measures themselves.

To face this situation, one had to find a compromise between stability and reaction. Given the nature of the two components of the filter state, position and speed, one may consider that their errors do not correlate.

Because we assume a uniform movement model that we know is not the true nature of the system, we know that the speed calculation of the model is not very accurate. A process noise covariance matrix was empirically estimated, based on several tests, so that a good smoothness/reactivity relationship was kept.

In practice, this approach proved to improve the estimation of the ball position. Fig. 7 represents a capture of a ball movement, where the black dots are the ball positions estimated by the robot visual sensors and thus are unfiltered. Red stars represent the position estimations after applying the Kalman filter. The ball was thrown against the robot and deviated accordingly and the robot position is represented by the black star in its centre and its respective radius. It is easily perceptible that the unfiltered positions are affected by much noise and the path of the ball after the collision is deviated from the real path. The filtered positions however, seem to give a much better approximation to the real path taken by the ball.

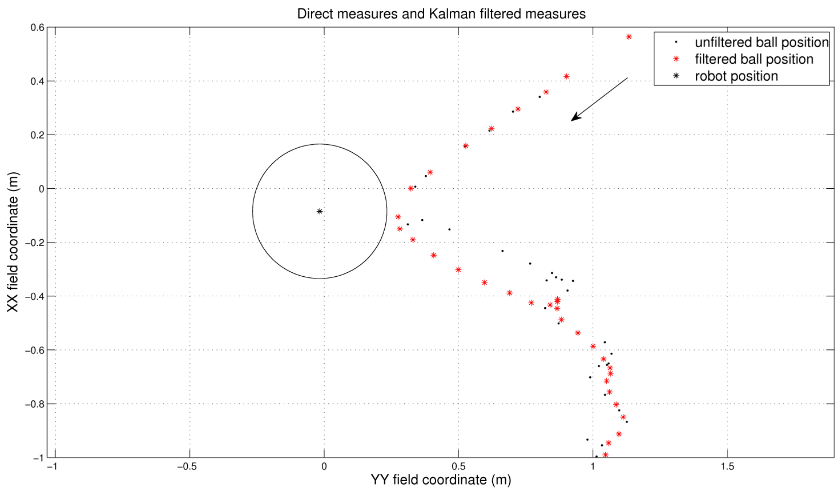


Fig. 7. Plot of a ball movement situation.

Using the filter *a-priori* estimation, a system to detect great differences between the expected and read positions was implemented, allowing to detect hard deviations on the ball path.

3.2 Ball velocity

The calculation of the ball velocity is a feature becoming more and more important over the time. It allows that better decisions can be implemented based on the ball speed value and direction. Assuming the same ball movement model described before, constant ball velocity between cycles and no friction considered, one could theoretically calculate the ball velocity by simple instantaneous velocity of the ball with the first order derivative of each component $\frac{\Delta D}{\Delta T}$, being ΔD the displacement on consecutive measures and ΔT the time interval between consecutive measures. However, given the noisy environment it is also predictable that this approach would be greatly affected by that noise and thus its results would not be satisfactory (as it is easily visible in Fig. 8.a).

To keep a calculation of the object velocity consistent with its displacement, an implementation of a linear regression algorithm was chosen. This approach based on linear regression [11] is similar to the velocity estimation described in [6]. By keeping a buffer of the last m measures of the object position and sampling instant (in this case buffers of 9 samples were used), one can calculate a regression line to fit the positions of the object. Since the object position is composed by two coordinates *(x,y)*, we actually have two linear regression calculations, one for each dimension, although it is made in a transparent way, so the description in this section is presented generally, as if only one dimension was considered.

When applied over the positions estimated by the Kalman filter, the linear regression velocity estimations are much more accurate than the instant velocities calculated by $\frac{\Delta D}{\Delta T}$, as visible in Fig. 8.b.

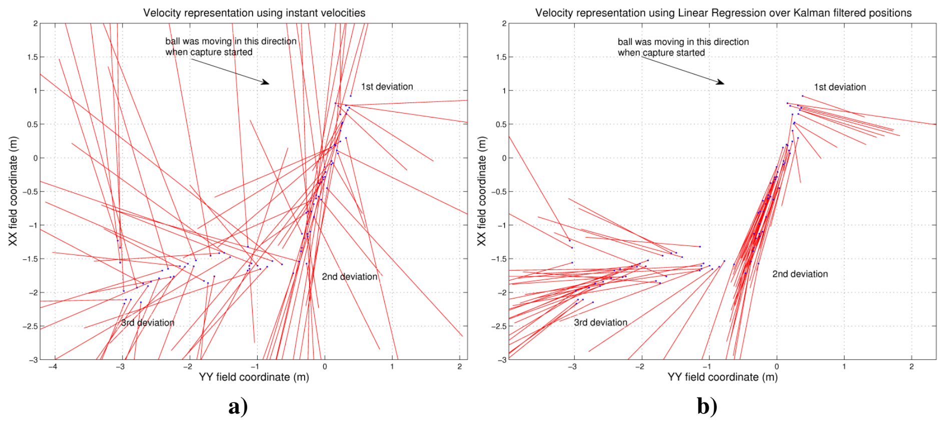


Fig. 8. Velocity representation using: Left, **a)**: consecutive measures displacement; Right, **b)**: linear regression over Kalman filtered positions.

In order to try to make the regression converge more quickly on deviations of the ball path, a reset feature was implemented, which allows deletion of the older values, keeping only the n most recent ones, allowing a control of the used buffer size. This reset results from the interaction with the Kalman filter described above, which triggers the velocity reset when it detects a hard deviation on the ball path.

Although in this case the Kalman filter internal functioning estimates a velocity, the obtained values were tested to confirm if the linear regression of the ball positions was still needed. Tests showed that the velocity estimated by the Kalman filter has a slower response than the linear regression estimation when deviations occur. Given this, the linear regression was used to estimate the velocity because quickness of convergence was preferred over the slightly smoother approximation of the Kalman filter in the steady state. That is because in the game environment, the ball is very dynamic, it constantly changes its direction and thus a convergence in less than half the cycles is much preferred.

3.3 Team ball position sharing

Due to the highly important role that the ball has in a soccer game, when a robot cannot detect it by its own visual sensors (omni or frontal camera), it may still know the position of the ball, through sharing of that knowledge by the other team mates.

The ball data structure include a field with the number of cycles it was not visible by the robot, meaning that the ball position given by the vision sensors can be the "last seen" position. When the ball is not visible for more than a given number of cycles, the robot assumes that it cannot detect the ball on its own. When that is the case, it uses the information of the ball communicated by the other running team mates to know where the ball is. This can be done through a function to get the statistics on a set of positions, mean and standard deviation, to get the mean value of the position of the ball seen by the team mates and assume it as its own.

Another approach is to simply use the ball position of the team mate that is closer to the ball, being the one that theoretically have more confidence in the detection. Whatever the case, the robot assumes that ball position as its own. When detecting the ball on its own, there is also the need to validate that information. Currently the seen ball is only considered if it is within a given margin inside the field of play as there would be no point in trying to play with a ball outside the field. Also, a maximum detection distance is considered, because of the large image distortion at long distances. Fig. 9 illustrates the general ball integration activity diagram.

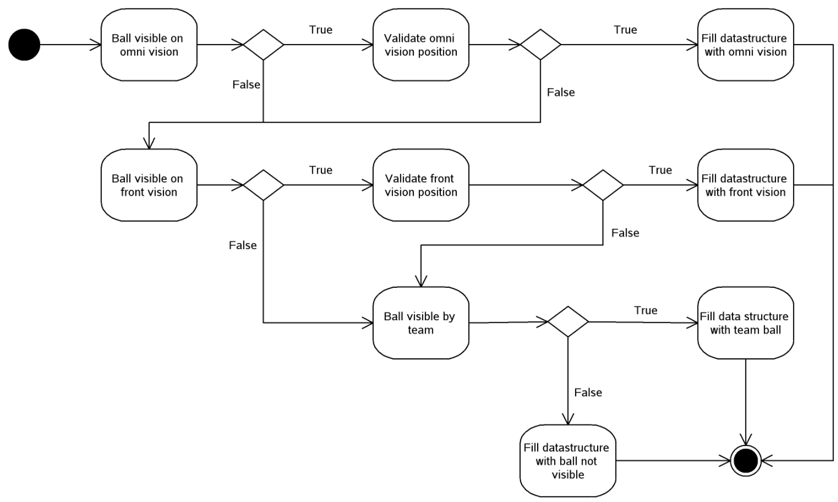


Fig. 9. Ball integration activity diagram.

4 Obstacle detection and sharing

An increasing necessity felt by the team, to improve its performance, is the need for a better obstacle detection and sharing of obstacle information among team mates. This need is important to ensure a global idea of the field occupancy, since the team formation usually keeps the robots spread across the field. With a good cover of field obstacles, passlines and dribbling corridors can be estimated more easily allowing improvements on team strategy and coordination. According to RoboCup rules, the robots are mainly black. Since in game robots play autonomously, every obstacles in the field are the robots themselves (occasionally the referee, which is recommended to have black/dark pants). The vision algorithm take advantage of this fact and detects the obstacles by evaluating blobs of black colour inside the field of play [12]. Through the mapping of image positions to real metric positions, obstacles are identified by their centre and left and right limits. The integration is then responsible for the identification of the obstacles.

In a first step, and since the maximum size of the robots is known, visual obstacles are separated by size. An obstacle can be a candidate to be a robot if it has acceptable dimensions, always considering an error margin, depending on the distance to it. With the known team mates positions (shared via wireless), a matching is tried by testing the obstacle estimated centre with the team mate position, considering the robot radius plus an error margin as matching area (Fig. 10.a)).

In a second step, the remaining large obstacles are also compared with the team mates not previously identified. These large obstacles are usually due to the robots being together, forming a unique black blob. In this case, the idea is somewhat opposed to singular obstacles, since in this case, the team mate position is to be tested with the obstacle area. A positive identification of a team mate within the detected obstacle area results in the division of the obstacle in 2 parts, a team mate obstacle and an opponent obstacle (Fig. 11).

The obstacles identified as team mates and opponents can afterwards be treated differently for team cooperation purposes.

5 Conclusion

The work already accomplished concerning sensor and information fusion, especially ball information treatment, helped to maintain a more reliable description of the state of the world.

The techniques chosen for information and sensor fusion proved to be effective in accomplishing their objectives. The Kalman filter allows to filter the noise on the ball position and provides an important prediction feature which allows fast detection of deviations of the ball path. The linear regression used to estimate the velocity is also effective, and combined with the deviation detection based on the Kalman filter prediction error, provides a faster way to recalculate the velocity in the new trajectory.

The increasing reliability of the ball position and velocity lead to a better ball trajectory evaluation. This allowed the development of a more effective goalie action, as well as other behaviours, such as ball interception behaviours and pass reception.

The obtained preliminary results regarding obstacle identification, provide tools for an improvement of the overall team coordination and strategic play.

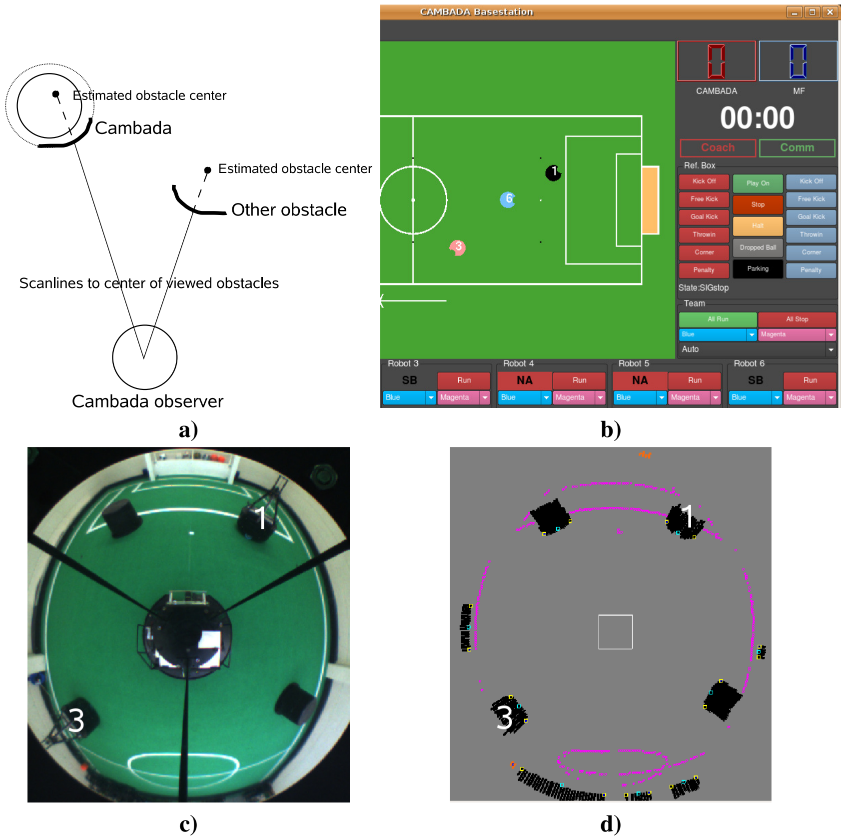


Fig. 10. Identification of single obstacles. Top Left **a)**: When a CAMBADA robot is on, the detected obstacles estimated centres are compared with the known position and tested if they are within the robot radius; the left obstacle is within the CAMBADA radius, the right one is not; Top Right **b)**: A screenshot of the CAMBADA base station, with 3 robots localised; Bottom Left **c)**: an image acquired from the middle robot, with robots 1 and 3 visible and other 2 single obstacles (opponents); Bottom Right **d)**: the same image processed where all the single obstacles are detected. 1 and 3 are the correctly detected CAMBADA robots, while the other 2 are marked as opponents.

The accomplished work improved the team performance, allowing it to distinctively achieve the 1st place in the Portuguese robotics open Robótica2008 and the 1st place in the RoboCup2008.

Acknowledgments

This work was partially supported by project ACORD Adaptive Coordination of Robotic Teams, FCT/PTDC/EIA/70695/2006.

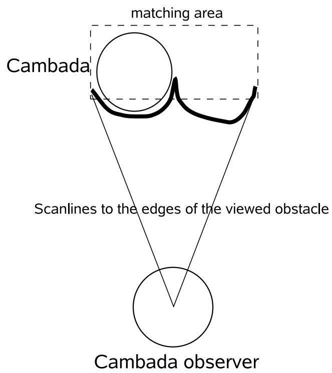


Fig. 11. Detection of multiple obstacles. The CAMBADA robot is matched as part of the detected obstacle, resulting in a division of the obstacle in 2 (team mate and opponent)

References

1. Kitano, H., Asada, M., Kuniyoshi, Y., Noda, I., Osawa, E.: RoboCup: The Robot World Cup Initiative. Proceedings of the first international conference on Autonomous agents (1997) 340–347
2. MSL Technical Committee 1997-2008: Middle Size Robot League Rules and Regulations for 2008 (2007)
3. Durrant-Whyte, H., Henderson, T.: Multisensor Data Fusion. In: Springer Handbook of Robotics. Springer (2008)
4. Lauer, M., Lange, S., Riedmiller, M.: Calculating the perfect match: an efficient and accurate approach for robot self-localization. In: RoboCup 2005: Robot Soccer World Cup IX. Lecture Notes in Computer Science, Springer (2006) 142–153
5. Neves, A., Martins, D., Pinho, A.: A hybrid vision system for soccer robots using radial search lines. In: Proc. of the 8th Conference on Autonomous Robot Systems and Competitions, Portuguese Robotics Open - ROBOTICA'2008, Aveiro, Portugal (2008) 51–55
6. Lauer, M., Lange, S., Riedmiller, M.: Modeling Moving Objects in a Dynamically Changing Robot Application. In: KI 2005: Advances in Artificial Intelligence. Volume 3698 of LNCS. Springer (2005) 291–303
7. XU, Y., JIANG, C., TAN, Y.: SEU-3D 2006 Soccer Simulation Team Description. In: CD Proc. of RoboCup Symposium 2006. (2006)
8. Marcelino, P., Nunes, P., Lima, P., Ribeiro, M.I.: Improving object localization through sensor fusion applied to soccer robots. In: Proc. Scientific Meeting of the Portuguese Robotics Open - Robótica2003. (2003)
9. Ferrein, A., Hermanns, L., Lakemeyer, G.: Comparing Sensor Fusion Techniques for Ball Position Estimation. In: RoboCup 2005: Robot Soccer World Cup IX. Volume 4020 of LNCS. Springer (2006) 154–165
10. Bishop, G., Welch, G.: An Introduction to the Kalman Filter. In: Proc of SIGGRAPH, Course 8. Number NC 27599-3175, Chapel Hill, NC, USA (2001)
11. Motulsky, H., Christopoulos, A.: Fitting models to biological data using linear and nonlinear regression. GraphPad Software Inc. (2003)
12. Neves, A., Corrente, G., Pinho, A.: An omnidirectional vision system for soccer robots. In: Progress in Artificial Intelligence, Proc. of the EPIA 2007. Volume 4874 of LNAI., Springer (2007) 499–507