Real-time generic ball recognition in RoboCup domain

Daniel A. Martins, António J. R. Neves and Armando J. Pinho

Signal Processing Lab, DETI / IEETA
University of Aveiro, 3810–193 Aveiro, Portugal
dam@ua.pt, an@ua.pt, ap@ua.pt

Abstract. This paper proposes a solution to detect standard FIFA balls, independent of their color, in the context of the RoboCup Middle Size League. The proposed work is being developed for the robotic soccer team of the University of Aveiro, CAMBADA (Cooperative Autonomous Mobile roBots with Advanced Distributed Architecture). The vision system of CAMBADA robots are based on an hybrid vision system, formed by an omnidirectional vision sub-system and a perspective vision sub-system, that together can analyze the environment around the robots, both at close and long distances. The proposed approach is based on the use of an edge detection algorithm followed by the use of the circular Hough transform. Despite the proposed method is under development, the preliminary experimental results are very encouraging. Moreover, the processing time allows real-time ball detection.

Keywords: Robotics; robotic soccer; computer vision; object recognition; omnidirectional vision; color classification.

1 Introduction

The Middle Size League (MSL) competition of RoboCup is a standard real-world test for autonomous multi-robot control. Being yet a color-coded environment, despite the recent changes introduced, such as the goals without color, recognizing colored objects such as the orange ball, the black obstacles, the green field and the white lines is a basic ability for robots.

One problem domain in RoboCup is the field of Computer Vision, responsible for providing basic information that is needed for calculating the behavior of the robots. Catadioptric vision systems (often named omnidirectional vision systems) have captured much interest in the last years, because they allow a robot to see in all directions at the same time without having to move itself or its camera [1–5]. However, due to the huge dimension of the current field, several teams have also included in their robots a perspective camera to detect objects far from the robot [6].

On the RoboCup MSL, the color codes tend to disappear as the competition evolves. Being the color of the ball the next color scheduled to vary, in this paper we propose a

This work was supported in part by the Fundação para a Ciência e a Tecnologia (FCT) project, ACORD - PTDC/EIA/70695/2006.

solution to detect balls independently of their color. This solution is based in a morphological analysis of the image, being strictly directed to detect round objects in the field, in this case the ball.

This paper is organized as follows. In Section 2 we describe the design of our robots, in particular their vision system. Section 3 presents the proposed algorithm for real-time generic ball recognition. In Section 4 we present experimental results obtained by our system. Finally, in Section 5, we draw some conclusions.

2 Architecture of the robots

The vision system of CAMBADA robots is based on an hybrid vision system, formed by an omnidirectional vision sub-system and a perspective vision sub-system, that together can analyze the environment around the robots, both at close and long distances (see Fig. 1).

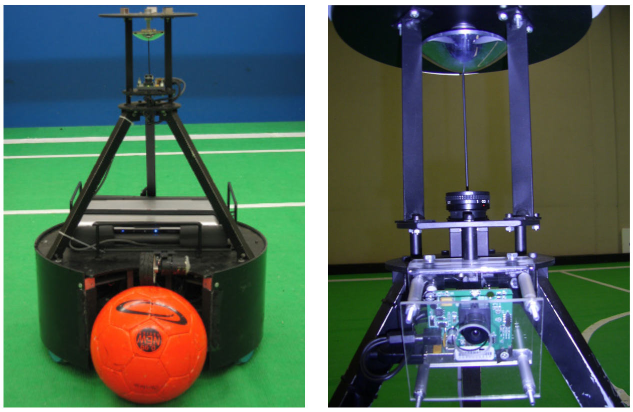


Fig. 1. One of the robots used by the CAMBADA middle-size robotic soccer team and its hybrid vision system.

The information regarding close objects, like white lines of the field, other robots and the ball, are acquired through the omnidirectional system, whereas the perspective system is used to locate other robots and the ball at long distances, which are difficult to detect using the omnidirectional vision system.

The software architecture is based on a distributed paradigm grouping main tasks in different modules. The software can be split in three main modules, namely the *Utility Sub-System*, the *Color Processing Sub-System* and the *Morphological Processing Sub-System*, as can be seen in Fig. 2. Each one of these sub-systems labels a domain area where their processes fit, as the case of *Acquire Image* and *Display Image* in the *Utility Sub-System*. As can be seen in *Color Processing Sub-System*, proper color classification and extraction processes were developed, along with an object detection process to extract information, through color analysis, from the acquired image [7, 8]. The *Morphological Processing Sub-System*, explained in Section 3, presents an early version of a color independent ball detection algorithm, that is still under heavy study and development.

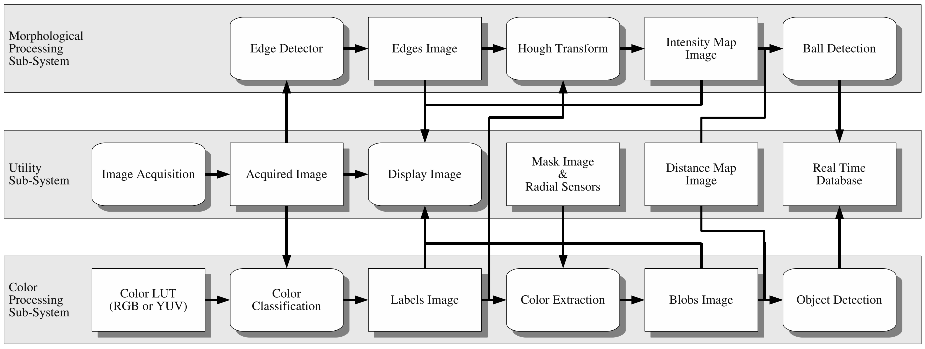


Fig. 2. The software architecture of the vision system developed for the CAMBADA robotic soccer team.

Despite the obvious differences between the omnidirectional and the perspective sub-systems, the software architecture used in both is the same, changing only the *Image Mask & Radial Sensors* and the *Distance Mapping Image* [7].

3 Proposed approach

Morphological object recognition through image analysis has became more robust and accurate in the past years, whereas still very time consuming even to modern personal computers [9–11]. Being the RoboCup a real-time environment, available processing time can become a big constrain when analyzing large amounts of data or executing complex algorithms. Many of the algorithms proposed during a previous research work showed their effectiveness but, unfortunately, their processing time is in some cases over one second [9].

The *Morphological Processing Sub-System* (see Fig. 2) was developed to overcome this obstacle using a two pass analysis to detect the ball. First, the image is searched for points of interest (potential locations) where balls can be found. Then a validation system is applied to the spots previously found to discard false ball locations.

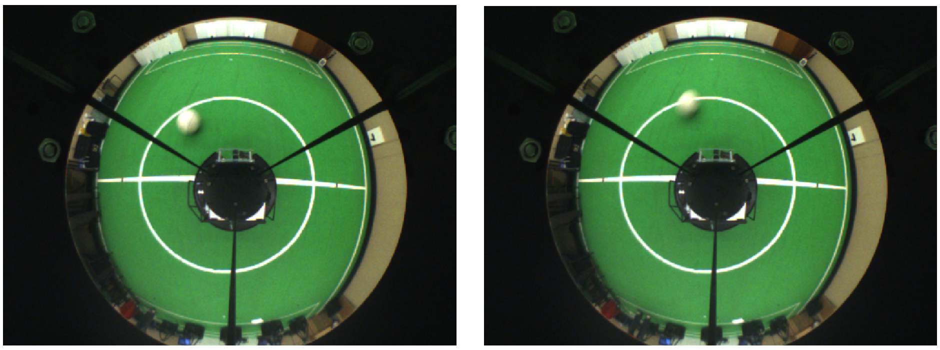
The search for potential spots is conducted taking advantage of morphological characteristics of the ball (round shape) using a feature extraction technique known as the *Hough* transform. First used to identify lines in images, the *Hough* [21–23, 16] transform has been generalized, through the years, to identify positions of arbitrary shapes, most commonly circles or ellipses, in images.

To feed the *Hough* transform process, it is necessary a binary image with the edge information of the objects. This image, *Edges Image*, is obtained using an image operator commonly called *edge detector*. In Section 3.1 it is presented an explanation of this process and its implementation.

3.1 Edge detection

Being this the first image processing step in the morphological detection, it must be as efficient and accurate as possible in order to not compromise the following processes. Besides being fast to calculate, the pretended resulting image must be absent of noise as much as possible, with well defined boundaries and be motion blur tolerant. Be tolerant to motion blur means that even when the objects present blur deformation in the image, the edge detector can retrieve its contours. In Fig. 3 b) it is shown an example of the motion blur deformation.

Before making the choice, some image edge detectors were compared. The comparison was made between the three main image operators used to find edges, Sobel [12–15], Laplace [15, 16] and Canny [17–20]. These are, by far, the most well known and efficient image operators used in edge detection. The tests occurred under two distinct situations: with the ball standing still and ball moving fast through the field. The test with the ball moving fast was realized to study the motion blur effect in the edge detectors on high speed objects captured with relatively low frame rates (30 fps). Figure 3 shows an image of each studied scenario.



(a) (b)

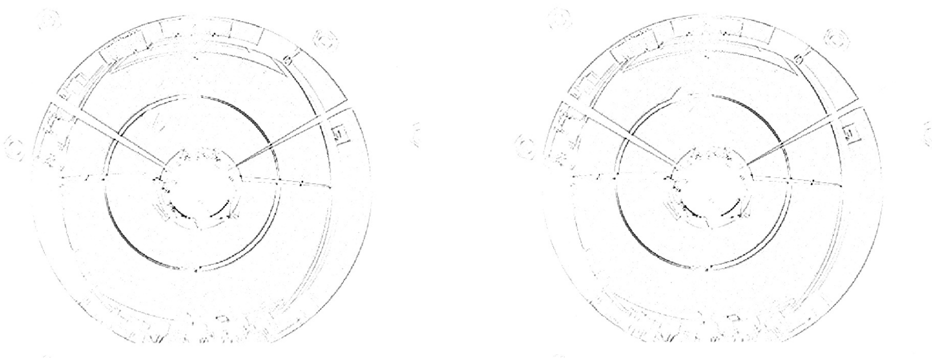
Fig. 3. Two examples of typical images captured in the RoboCup environment. In (a) a test image with the ball standing still and in (b) with the ball moving at high speed.

The three edge detection operators are based on convolving the image with a small, separable, and integer valued filter in the horizontal and vertical directions and are therefore relatively inexpensive in terms of computations. In both tests, all the operators used a convolution window of size 3.

The Sobel operator is widely used in edge detection algorithms [12–15]. It is a discrete differentiation operator, that computes an approximation of the gradient of the image intensity function. This approximation is relatively crude, in particular for high frequency variations in the image.

By calculating the gradient of the image intensity at each point, the operator gives the direction of the largest possible increase from light to dark and the rate of change in that direction. This shows how "abruptly" or "smoothly" the image changes at that point, and therefore how likely that part of the image represents an edge, as well as how that edge is likely to be oriented.

In Fig. 4 we can see the resulting images of the Sobel edge detector applied to the images in Fig. 3.



(a) (b)

Fig. 4. Two examples previously presented in Fig. 3, now with the Sobel operator applied. In (a) a test image with the ball standing still and in (b) with the ball moving at high speed.

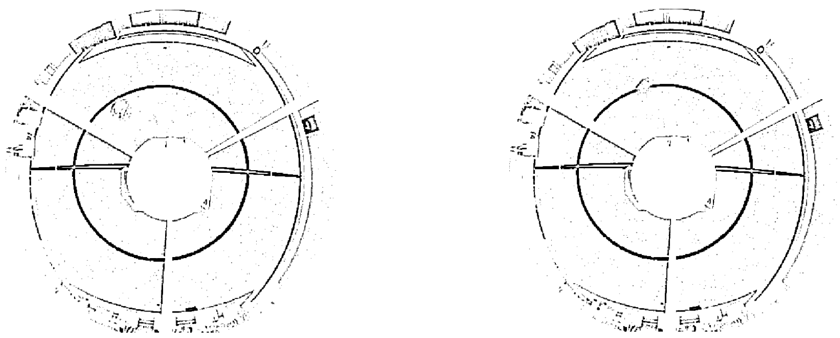
The Laplace operator, as the Sobel, is commonly used in image processing as an edge detection algorithm [15, 16]. Also called *Laplacian*, it is denoted by Δ or ∇^2 and is a differential operator.

In Fig. 5, the resulting images of the Laplace edge detector applied to the images in Fig. 3 are shown.

The Canny edge detection operator uses a multi-stage algorithm to detect a wide range of edges in images. The Canny operator was developed to be an optimal edge detection algorithm [17–20], presenting the following features as the optimal ones:

- Good detection - the algorithm should mark as many real edges in the image as possible;
- Good localization - edges marked should be as close as possible to the edge in the real image;
- Minimal response - a given edge in the image should only be marked once, and where possible, image noise should not create false edges.

In the Canny operator, these requirements are obtained using calculus of variations, a technique used to find a function which optimizes a functional. The optimal function

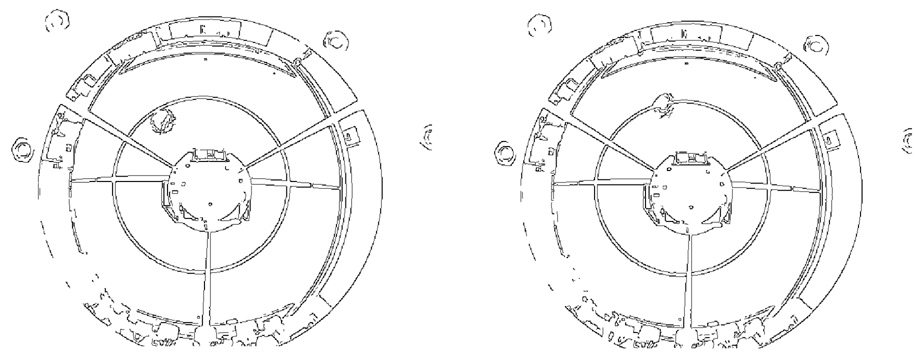


(a) (b)

Fig. 5. Two examples previously presented in Fig. 3, now with the Laplace operator applied. In (a) a test image with the ball standing still and in (b) with the ball moving at high speed.

in Canny detector is described by the sum of four exponential terms, but can be approximated by the first derivative of a Gaussian. Using a Canny filter it is possible to obtain an image of edges with edges with size of 1 pixel, due to its non-maximal suppression properties.

In Fig. 6 the resulting images of the Canny edge detector applied to the images in Fig. 3 are presented.



(a) (b)

Fig. 6. Two examples previously presented in Fig. 3, now with the Canny operator applied. In (a) a test image with the ball standing still and in (b) with the ball moving at high speed.

To choose the best edge detector, the results from the tests will be compared taking in account the image of edges and processing time needed by each edge detector. In one hand, the real-time capability must be assured with low processing times. In the other hand, the algorithm must be able to detect the edges of the ball independently of its motion blur effect due to the speed.

Note that in Figs. 4, 5 and 6, the edges are represented by the black pixels. In Fig. 4 it is possible to see the bad results provided by the Sobel edge detector. The edges are hard to see and the ball is almost invisible. In Fig. 5, the results improved a little. With this edge detector, the ball can be distinguished when standing still, but when the ball is moving a little faster, this edge detector delivers bad results. In Fig. 6, the ball can be seen perfectly when standing still. When moving, the ball can also be pointed out, but not as clearly as when it is standing still. Comparing Fig. 4 with Fig. 5 and Fig. 6, it is notorious the superior results provided by the Canny edge detector.

During these tests, the processing time of each algorithm was measured, being their mean time presented in Table 1. A first look at the mean time spent for each edge detector (see Table 1) may point out the Canny as the worst choice. In fact, the Canny processing time is the highest between the ones tested but, even so, it is fast enough to be used in real-time applications, and the resulting edge images show the effectiveness of this edge detector, way above the others.

Edge detector	Sobel	Laplace	Canny
Average processing time	5.17	4.11	10.59

Table 1. Average processing time obtained from the three tested edge detection algorithms. All times are in milliseconds.

Since the Canny filter was developed to be an "optimal edge detector", it was expected from the beginning its supremacy in the results, being finally confirmed.

3.2 Hough transform

The *Hough* transform is a technique widely used to find instances of objects, of a certain class of shapes (lines, circles or even ellipses) by a voting procedure [21–23, 16]. Note that finding the object is different from validating the object and with this method the image is only searched for points of interest. This voting procedure is carried out in a parameter space, from which object candidates are obtained as local maxima in a so-called *Intensity Map Image* (see Fig. 2) that is explicitly constructed by the algorithm for computing the *Hough* transform. Follows a detailed description of the algorithm implementation and optimization for real-time purposes, and the results.

In Fig. 7, it is shown an example of the algorithm used, where the dark continuous lines represent the edges found in the *Edges Image* and the dashed lines are the result from the *Hough* transform over discrete spots of the continuous lines. Due to the *Hough* circular transform especial features, a big round object in the *Edges Image* would produce in the *Intensity Map Image* a very intense peak in the center of the object, as shown on the left side of Fig. 7. For contrast, a non-round object would produce areas of low intensity in the *Intensity Map Image*, as represented in the right side of Fig. 7.

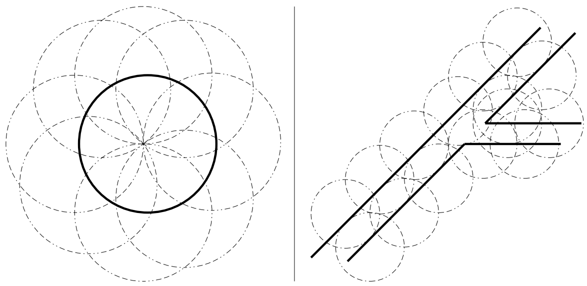


Fig. 7. *Hough* transform example. On the left, edges from the ball. On the right, edges from the field white lines.

This approach rises the following problem: as the ball moves away, its edge circle size scales down in the image. To solve this problem, information about the distance between the robot center and the ball is used to adjust the *Hough* transform.

As this algorithm requires drawing circles (in the *Intensity Map Image*) centered in the pixel actually being processed, it was developed a tool to accelerate this operation. Based in predefined circle sizes and in the image dimensions, an array of offsets is created in memory for each circle size. When added with the index of the pixel where the circle has to be centered, this array of offsets will directly map the circle in the pixels of the image, reducing drastically the processing time commonly needed for circle creation.

To improve the robustness of this algorithm, some extra details are taken into account. To obtain a clean image, the edges created by the robot reflection in the mirror are removed using information from the *Mask Image* [7, 8]. Furthermore, although the FIFA rules do not have restrictions for the ball color, it is considered that the ball is never green. This last assumption reduces the potential risk of false positives in the middle of the field due to other robots over white lines, and crossing white lines. To do so, information from the *Labels Image* [7, 8] is compared against the current pixel being processed. And finally, because distant balls are very irregular in the *Edges Image*, due to its reduced size, distant edges are discarded in the *Hough* transform to avoid erroneous false points of interest.

In some situations, in particular when the ball is not present in the field, the proposed method finds false ball positives. To reduce this problem and improve the ball information reliability, we proposed a validation system which would discard false positives based on information from the *Intensity Map Image* and *Labels Image*.

This validation is achieved by a two-steps algorithm. First, the *Intensity Map Image* is searched for local maximum points. These points are then filtered by a threshold limit after what are named as points of interest. Then, the green color of the ground is used to decide whether the point of interest is or not a valid ball. As the major number of false positives appear over the intersections of white lines, these areas are always surrounded by the green color of the field in the *Labels Image*. Acting accordingly to the distance between the robot and the point of interest, the validation system analyzes the image in the area surrounding the point, and if this area has more than a predefined percentage of green pixels, then the point of interest is discarded.

The algorithms described in this section were implemented both in the omnidirectional and perspective image sub-systems.

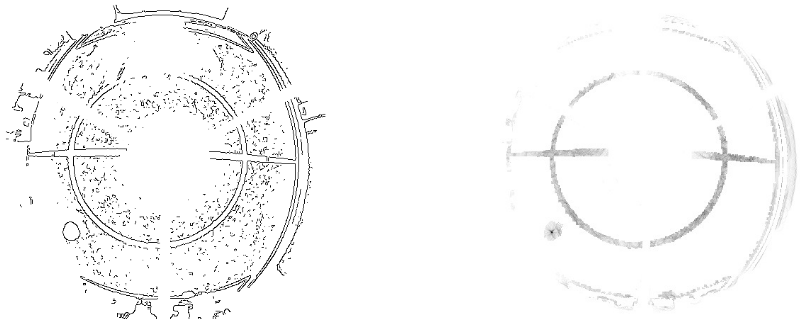
4 Experimental results

In Figs. 8 and 9 we can see an example of the *Morphological Processing Sub-System* pipeline, presented in Fig. 2. As can be observed, the ball in the *Edges Image* (Fig. 9) is not perfectly circular. Even so, as the *Hough* transform is very tolerant to gaps in feature boundary descriptions and is relatively unaffected by image noise [17, 24–26], it still performs well, as can be seen in Fig. 8. Notice that the center of the ball presents a very high peak when compared to the rest of the image in Fig. 9 b).

Fig. 8. Example of a captured image using the morphological ball detection system. The cross over the ball points out the detected position.

To assure good results in RoboCup competitions, the system was tested with the algorithms described above. Testing systems in working conditions generates much more realistic results. For that purpose, the robot was moved along a predefined path through the robotic football field, leaving the ball in a known location. The ball position given by the robot is then compared with the real position of the ball. The results in this test may be affected by the localization algorithm errors and the robot bumps while moving, being these external errors outside the scope of this study.

The robot path in the field may be seen in Fig. 10, along with the measured ball position. It is possible to notice that the average of the measured positions of the ball is almost centered in the real ball position, showing the effectiveness of this system. We obtained a very high detection ratio (near 100%) and a great accuracy, with the average measures very near the real ball position. However, with the proposed approach, the omnidirectional vision sub-system can detect the ball with this precision until distances up to 3 meters. With the perspective vision sub-system we can correctly detect the ball at higher distances (up to 6 meters).



(a) (b)

Fig. 9. In (a) the example previously presented in Fig. 8, now with the Canny edge detector applied. Notice as the robot reflection was avoided from the *Edges Image*. In (b) the example presented in a), now with the *Hough* transform applied. Notice that far edges are not processed and results over green pixels are not considered.

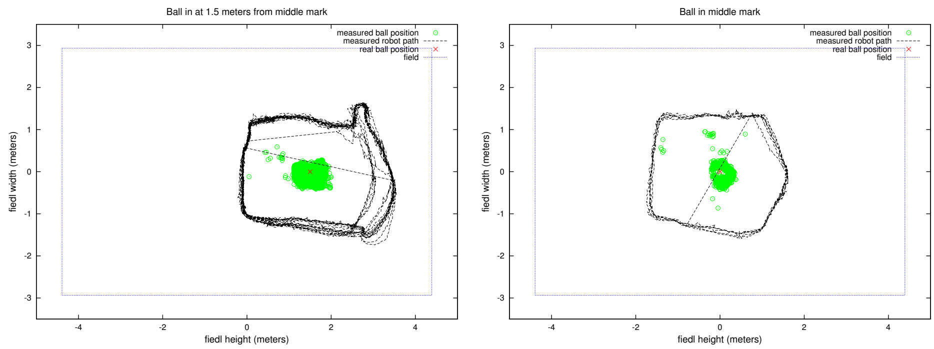


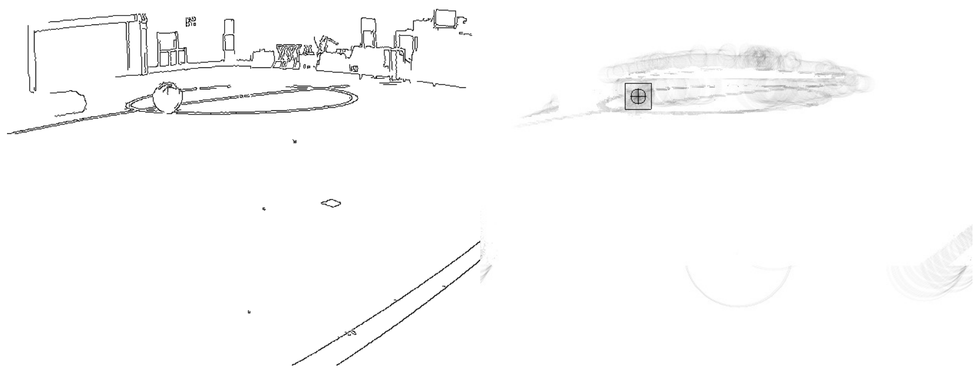
Fig. 10. Experimental results obtained by the omnidirectional sub-system using the morphological ball detection. In this experience the ball was positioned in two known positions of the field. The robot has performed a predefined trajectory while the position of the ball was registered. Both axis in the graphics are in meters.

In Fig. 11 it is presented an example of a ball detection using the perspective vision sub-system.

The average processing time of the proposed approach is approximately 25 milliseconds, both for omnidirectional and perspective vision sub-systems. Note that, the experimental results were obtained with acquired images of 640×480 pixels. Moreover, it was used a laptop with an Intel Core 2 duo at 2.0 GHz and 1 GB of memory.

5 Final remarks

This paper proposes a solution to detect standard FIFA balls, independent of their color, in the context of the RoboCup Middle Size League. The proposed approach is based



(a) (b)

Fig. 11. Example of an image processed by the perspective sub-system using the morphological ball detection. In (a) the Edges Images and in (b) the Intensity Image. The black square shows the area being analysed over the point of interest. The black circle with the cross points out the detected ball location.

on the use of an edge detection algorithm followed by the use of the circular Hough transform.

Our experimental results show that the Canny edge detector is the best choice among the other edge detection algorithms studied, considering the blur effect resulting from the movement of the ball.

The *Hough* transform revealed to be a good method to detect circular shaped objects, and showed to be very tolerant to gaps in feature boundary descriptions and is relatively unaffected by image noise.

Despite the proposed method is under development, the preliminary experimental results are very encouraging. This method will be used this year by the CAMBADA team in the mandatory challenge at RoboCup 2008, in Suzhou, China.

References

1. Zivkovic, Z., Booij, O.: How did we built our hyperbolic mirror omni-directional camera - practical issues and basic geometry. Technical report, Intelligent Systems Laboratory, University of Amsterdam (2006)
2. Wolf, J.: Omnidirectional vision system for mobile robot localization in the robocup environment. Master's thesis, Graz University of Technology (2003)
3. Menegatti, E., Nori, F., Pagello, E., Pellizzari, C., Spagnoli, D.: Designing an omnidirectional vision system for a goalkeeper robot. In: Proc. of RoboCup 2001. Volume 2377 of Lecture Notes in Computer Science., Springer (2001) 78–87
4. Menegatti, E., Pretto, A., Pagello, E.: Testing omnidirectional vision-based monte carlo localization under occlusion. In: Proc. of the IEEE Intelligent Robots and Systems, IROS 2004. (2004) 2487–2493
5. Lima, P., Bonarini, A., Machado, C., Marchese, F., Marques, C., Ribeiro, F., Sorrenti, D.: Omni-directional catadioptric vision for soccer robots. Robotics and Autonomous Systems **36** (2001) 87–102

6. Voigtlrande, A., Lange, S., Lauer, M., Riedmiller, M.: Real-time 3D ball recognition using perspective and catadioptric cameras. In: Proc. of the 3rd European Conference on Mobile Robots, Freiburg, Germany (2007)
7. Neves, A.J.R., Martins, D.A., Pinho, A.J.: 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
8. Neves, A.J.R., Corrente, G., Pinho, A.J.: An omnidirectional vision system for soccer robots. In: Proc. of the EPIA 2007. Volume 4874 of Lecture Notes in Artificial Inteligence., Springer (2007) 499–507
9. Mitri, S., Frintrop, S., Pervolz, K., Surmann, H., Nuchter, A.: Robust object detection at regions of interest with an application in ball recognition. In: Proc. of the 2005 IEEE International Conference on Robotics and Automation, ICRA 2005, Barcelona, Spain (2005) 125–130
10. Treptow, A., Zell, A.: Real-time object tracking for soccer-robots without color information. Robotics and Autonomous Systems **48** (2004) 41–48
11. Fourth Workshop on Intelligent Solutions in Embedded Systems: Embedded Real-Time Ball Detection Unit for the YABIRO Biped Robot, Fourth Workshop on Intelligent Solutions in Embedded Systems (2006)
12. Zou, J., Li, H., Liu, B., Zhang, R.: Color edge detection based on morphology. In: First International Conference on Communications and Electronics, ICCE 2006. (2006) 291–293
13. Zin, T.T., Takahashi, H., Hama, H.: Robust person detection using far infrared camera for image fusion. In: Second International Conference on Innovative Computing, Information and Control, ICICIC 2007. (2007) 310–310
14. Umbaugh, S.E.: Computer Vision and Image Processing. Prentice Hall (1999)
15. Zou, Y., Dunsmuir, W.: Edge detection using generalized root signals of 2-d median filtering. In: Proc. of the International Conference on Image Processing, 1997. Volume 1. (1997) 417–419
16. Blaffert, T., Dippel, S., Stahl, M., Wiemker, R.: The laplace integral for a watershed segmentation. In: Proc. of the International Conference on Image Processing, 2000. Volume 3. (2000) 444–447
17. Boyle, R., Thomas, R.: Computer Vision: A First Course. Blackwell Scientific Publications (1988)
18. Canny, J.F.: A computational approach to edge detection. IEEE Transactions on Pattern Analysis and Machine Intelligence **8** (1986)
19. Davies, E.: Machine Vision: Theory, Algorithms and Practicalities. Academic Press (1990)
20. Gonzalez, R., Woods, R.: Digital Image Processing. Addison-Wesley Publishing Company (1992)
21. Ser, P.K., Siu, W.C.: Invariant hough transform with matching technique for the recognition of non-analytic objects. In: IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP 1993. Volume 5. (1993) 9–12
22. Zhang, Y.J., Liu, Z.Q.: Curve detection using a new clustering approach in the hough space. In: IEEE International Conference on Systems, Man, and Cybernetics, 2000. Volume 4. (2000) 2746–2751
23. Grimson, W.E.L., Huttenlocher, D.P.: On the sensitivity of the hough transform for object recognition. IEEE Trans. on Pattern Analysis and Machine Intelligence **12** (1990) 1255–1274
24. Ballard, D., Brown, C.: Computer Vision. Prentice Hall (1982)
25. Jain, A.: Fundamentals of Digital Image Processing. Prentice Hall (1989)
26. Vernon, D.: Machine Vision. Prentice Hall (1991)