



Contents lists available at ScienceDirect

# Mechatronics

journal homepage: [www.elsevier.com/locate/mechatronics](http://www.elsevier.com/locate/mechatronics)

## An efficient omnidirectional vision system for soccer robots: From calibration to object detection

António J.R. Neves \*, Armando J. Pinho, Daniel A. Martins, Bernardo Cunha

ATRI, IEETA/DETI, University of Aveiro, 3810-193 Aveiro, Portugal

### ARTICLE INFO

Article history:  
Available online xxxxx

Keywords:  
Robotic vision  
Omnidirectional vision systems  
Color-based object detection  
Shape-based object detection  
Vision system calibration

### ABSTRACT

Robotic soccer is nowadays a popular research domain in the area of multi-robot systems. In the context of RoboCup, the Middle Size League is one of the most challenging. This paper presents an efficient omnidirectional vision system for real-time object detection, developed for the robotic soccer team of the University of Aveiro, CAMBADA. The vision system is used to find the ball and white lines, which are used for self-localization, as well as to find the presence of obstacles. Algorithms for detecting these objects and also for calibrating most of the parameters of the vision system are presented in this paper. We also propose an efficient approach for detecting arbitrary FIFA balls, which is an important topic of research in the Middle Size League. The experimental results that we present show the effectiveness of our algorithms, both in terms of accuracy and processing time, as well as the results that the team has been achieving: 1st place in RoboCup 2008, 3rd place in 2009 and 1st place in the mandatory technical challenge in RoboCup 2009, where the robots have to play with an arbitrary standard FIFA ball.

© 2010 Elsevier Ltd. All rights reserved.

### 1. Introduction

The Middle Size League (MSL) of RoboCup is a forum where several research areas have been challenged for proposing solutions to well-defined practical problems. The robotic vision is one of those areas and, for most of the MSL teams, it has become the only way of sensing the surrounding world.

From the point of view of a robot, the playing field during a game provides a fast-changing scenery, where the teammates, the opponents and the ball move quickly and often in an unpredictable way. The robots have to capture these scenes through their cameras and have to discover where the objects of interest are located. There is no time for running complex algorithms. Everything has to be computed and decided in a small fraction of a second, for allowing real-time operation; otherwise, it becomes useless.

Real-time is not the only challenge that needs to be addressed. Year after year, the initially well controlled and robot friendly environment where the competition takes place has become increasingly more hostile. Conditions that previously have been taken for granted, such as controlled lighting or easy to recognize color coded objects, have been relaxed or even completely suppressed. Therefore, the vision system of the robots needs to be prepared for adapting to strong lighting changes during a game, as well as, for example, for ball-type changes across games.

In this paper, we provide a comprehensive description of the vision system of the MSL CAMBADA team (Fig. 1). Cooperative Autonomous Mobile robots with Advanced Distributed Architecture (CAMBADA) is the RoboCup MSL soccer team of the Institute of Electronics and Telematics Engineering of Aveiro (IEETA) research institute, University of Aveiro, Portugal. The team, which started officially in October 2003, won the 2008 MSL RoboCup World Championship and ranked 3rd in the 2009 edition.

We start by presenting and explaining the hardware architecture of the vision system used by the robots of the CAMBADA team, which relies on an omnidirectional vision system (Section 2). Then, we proceed with the description of the approach that we have adopted regarding the calibration of a number of crucial parameters and in the construction of auxiliary data structures (Section 3). Concerning the calibration of the intrinsic parameters of the digital camera, we propose an automated calibration algorithm that is used to configure the most important features of the camera, namely, the saturation, exposure, white-balance, gain and brightness. The proposed algorithm uses the histogram of intensities of the acquired images and a black and a white area, known in advance, to estimate the referred parameters. We also describe a general solution to calculate the robot centered distances map, exploring a back-propagation ray-tracing approach and the geometric properties of the mirror surface.

The soccer robots need to locate several objects of interest, such as the ball, the opponent robots and the teammates. Moreover, they also need to collect information for self-localization, namely, the position of the field white lines. For these tasks, we have developed fast and efficient algorithms that rely on color information.

\* Corresponding author.

E-mail addresses: [an@ua.pt](mailto:an@ua.pt) (A.J.R. Neves), [ap@ua.pt](mailto:ap@ua.pt) (A.J. Pinho), [dama@ua.pt](mailto:dama@ua.pt) (D.A. Martins), [mbc@det.ua.pt](mailto:mbc@det.ua.pt) (B. Cunha).



Fig. 1. The CAMBADA team playing at RoboCup 2009, Graz, Austria.

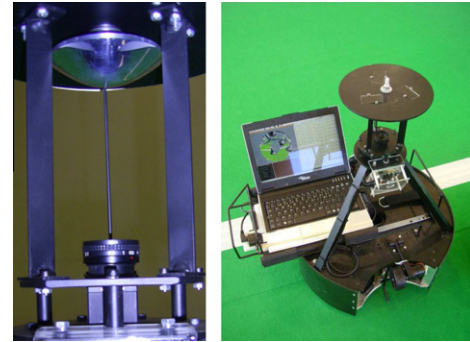


Fig. 2. On the left, a detailed view of the CAMBADA vision system. On the right, one of the robots.

The color extraction algorithms are based on lookup tables and use a radial model for color object detection. Due to the severe restrictions imposed by the real-time constraint, some of the image processing tasks are implemented using a multi-threading approach and use special data structures to reduce the processing time. Section 4 provides a detailed description of these algorithms.

As previously mentioned, the color codes assigned to the objects of interest tend to disappear as the competition evolves. For example, the usual orange ball used in the MSL will soon be replaced by an arbitrary FIFA ball, increasing the difficulty in locating one of the most important objects in the game. Anticipating this scenario, we developed a fast method for detecting soccer balls independently of their colors. In Section 5, we describe a solution based on the morphological analysis of the image. The algorithm relies on edge detection and on the circular Hough transform, attaining a processing time almost constant and complying with the real-time constraint. Its appropriateness has been clearly demonstrated by the results obtained in the mandatory technical challenge of the RoboCup MSL: 2nd place in 2008 and 1st place in 2009.

## 2. Architecture of the vision system

The CAMBADA robots [1] use a catadioptric vision system, often named omnidirectional vision system, based on a digital video camera pointing at a hyperbolic mirror, as presented in Fig. 2. We are using a digital camera Point Grey Flea 2, <sup>1</sup> FL2-08S2C with a 1/3" CCD Sony ICX204 that can deliver images up to 1024 × 768 pixels in several image formats, namely RGB, YUV 4:1:1, YUV 4:2:2 or YUV 4:4:4. The hyperbolic mirror was developed by IAIS Fraunhofer Gesellschaft <sup>2</sup> (FhG-AiS). Although the mirror was designed for the vision system of the FhG Volksbot <sup>3</sup> we are achieving also an excellent result with it in our vision system.

The use of omnidirectional vision systems have captured much interest in the last years, because it allows a robot to attain a 360° field of view around its central vertical rotation axis, without having to move itself or its camera. In fact, it has been a common solution for the main sensorial element in a significant number of autonomous mobile robot applications, as is the case of the MSL, where most of the teams have adopted this approach [2–9].

A catadioptric vision system ensures an integrated perception of all major target objects in the surrounding area of the robot, allowing a higher degree of maneuverability. However, this also implies higher degradation in the resolution with growing distances away from the robot, when compared to non-isotropic setups.

## 3. Calibration of the vision system

An important task in the MSL is the calibration of the vision system. This includes the calibration of intrinsic parameters of the digital camera, the computation of the inverse distance map, the detection of the mirror and robot center and the definition of the regions of the image that have to be processed. Calibration has to be performed when environmental conditions change, such as playing in a different soccer field or when the lighting conditions vary over time. Therefore, there are adjustments that have to be made almost continuously, for example if the playing field is unevenly illuminated, or less frequently, when the playing field changes. Moreover, a number of adjustments have also to be performed when some of the vision hardware of the robot is replaced, such as the camera or the mirror. All these calibrations and adjustments should be robust, i.e., they should be as much as possible insensitive to small environmental variations, they should be fast to perform and they should be simple to execute, so that no special calibration expert is required to operate them.

### 3.1. Self-calibration of the digital camera parameters

In a near future, it is expected that the MSL robots will have to play under natural lighting conditions and in outdoor fields. This introduces new challenges. In outdoor fields, the illumination may change slowly during the day, due to the movement of the sun, but also may change quickly in short periods of time due to a partial and temporally varying covering of the sun by clouds. In this case, the robots have to adjust, in real-time, both the color segmentation values as well as some of the camera parameters, in order to adapt to new lighting conditions [10].

The common approach regarding the calibration of the robot cameras in the MSL has been based on manual adjustments, that are performed prior to the games, or through some automatic process that runs offline using a pre-acquired video sequence. However, most (or even all) of the parameters remain fixed during the game.

We propose an algorithm that does not require human interaction to configure the most important parameters of the camera, namely the exposure, the white-balance, the gain and the brightness. Moreover, this algorithm runs continuously, even during the game, allowing coping with environmental changes that often occur when playing.

We use the histogram of intensities of the acquired images and a black and a white area, which location is known in advance, to estimate the referred parameters of the camera. Note that this approach differs from the well known problem of photometric camera calibration (a survey can be found in [11]), since we are not

<sup>1</sup> <http://www.ptgrey.com/products/flea2/>, Last accessed: 18/02/2010.

<sup>2</sup> <http://www.iais.fraunhofer.de/>, Last accessed: 18/02/2010.

<sup>3</sup> <http://www.volksbot.de/>, Last accessed: 18/02/2010.

interested in obtaining the camera response values, but only to configure its parameters according to some measures obtained from the acquired images. The self-calibration process for a single robot requires a few seconds, including the time necessary to start the application. This is significantly faster than the usual manual calibration by an expert user, for which several minutes are needed.

### 3.1.1. Proposed algorithm

The proposed calibration algorithm processes the image acquired by the camera and analyzes a white area in the image (a white area in a fixed place on the robot body, near the camera in the center of the image), in order to calibrate the white-balance. A black area (we use a part of the image that represents the robot itself, actually a rectangle in the upper left side of the image) is used to calibrate the brightness of the image. Finally, the histogram of the image intensities is used to calibrate the exposure and gain.

The histogram of the intensities of an image is a representation of the number of times that each intensity value appears in the image. For an image represented using 8 bits per pixel, the possible values are between 0 and 255. Image histograms can indicate some aspects of the lighting conditions, particularly the exposure of the image and whether it is underexposed or overexposed.

The assumptions used by the proposed algorithm are the following:

- (i) The white area should appear white in the acquired image. In the YUV color space, this means that the average value of U and V should be close to 127, that is to say, the chrominance components of the white section should be as close to zero as possible. If the white-balance is not correctly configured, these values are different from 127 and the image does not have the correct colors. The white-balance parameter is composed by two values, WB\_BLUE and WB\_RED, directly related to the values of U and V, respectively.
- (ii) The black area should be black. In the RGB color space, this means that the average values of R, G and B should be close to zero. If the brightness parameter is too high, it is observed that the black region becomes blue, resulting in a degradation of the image.
- (iii) The histogram of intensities should be centered around 127 and should span all intensity values. Dividing the histogram into regions, the left regions represent dark colors, while the right regions represent light colors. An underexposed image will be leaning to the left, while an overexposed image will be leaning to the right in the histogram (for an example, see Fig. 5a). The values of the gain and exposure parameters are adjusted according to the characteristic of the histogram.

Statistical measures can be extracted from the images to quantify the image quality [12,13]. A number of typical measures used in the literature can be computed from the image gray level histogram, namely, the mean

$$\mu = \sum_{i=0}^{N-1} iP_i, \quad \mu \in [0, 255], \quad (1)$$

the entropy

$$E = - \sum_{i=0}^{N-1} P_i \log(P_i), \quad E \in [0, 8], \quad (2)$$

the absolute central moment

$$ACM = \sum_{i=0}^{N-1} |i - \mu|P_i, \quad ACM \in [0 - 127] \quad (3)$$

and the mean sample value

$$MSV = \frac{\sum_{j=0}^4 (j+1)x_j}{\sum_{j=0}^4 x_j}, \quad MSV \in [0 - 5], \quad (4)$$

where  $N$  is the number of possible gray values in the histogram (typically, 256),  $P_i$  is the relative frequency of each gray value and  $x_j$  is the sum of the gray values in region  $j$  of the histogram (in the proposed approach we divided the histogram into five regions). When the histogram values of an image are uniformly distributed in the possible values, then  $\mu \approx 127$ ,  $E \approx 8$ ,  $ACM \approx 60$  and  $MSV \approx 2.5$ . In the experimental results we use these measures to analyze the performance of the proposed calibration algorithm. Moreover, we use the information of  $MSV$  to calibrate the exposure and the gain of the camera. The algorithm is depicted next.

```

do
do
  acquire image
  calculate the histogram of intensities
  calculate the MSV value
  if MSV<2.0 OR MSV>3.0
    apply the PI controller to adjust exposure
  else
    apply the PI controller to adjust gain
  set the camera with new exposure and gain values
while exposure or gain parameters change
do
  acquire image
  calculate average U and V values of the white
  area
  apply the PI controller to adjust WB_BLUE
  apply the PI controller to adjust WB_RED
  set the camera with new white-balance
  parameters
while white-balance parameters change
do
  acquire image
  calculate average R, G and B values of the black
  area
  apply the PI controller to adjust brightness
  set the camera with new brightness value
while brightness parameter change
while any parameter changed

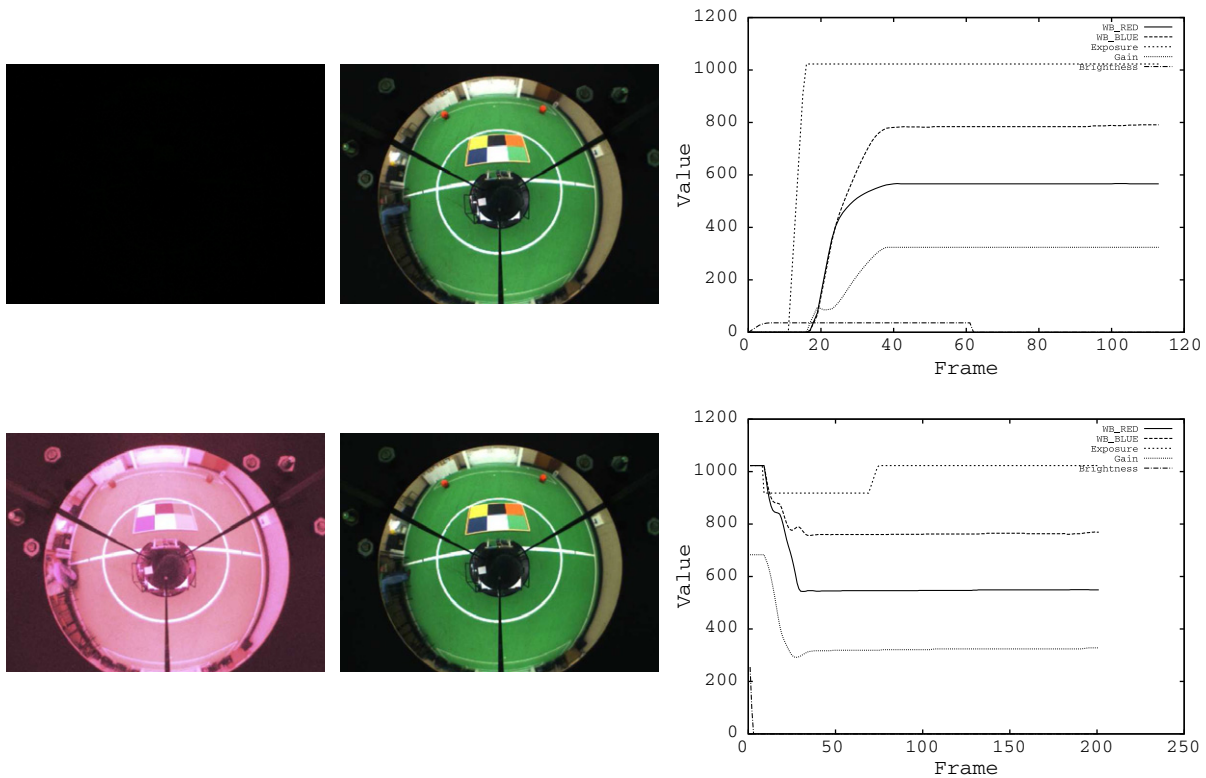
```

The calibration algorithm configures one parameter at a time, proceeding to the next one when the current one has converged. For each of these parameters, a PI controller was implemented. PI controllers are used instead of proportional controllers as they result in better control, having no stationary error. The coefficients of the controller were obtained experimentally: first, the proportional gain was increased until the camera parameter started to oscillate. Then, it was reduced to about 70% of that value and the integral gain was increased until an acceptable time to reach the desired reference was obtained [14]. The algorithm stops when all the parameters have converged. More details regarding this algorithm can be found in [15].

### 3.1.2. Experimental results

To measure the performance of this calibration algorithm, tests have been conducted using the camera with different initial configurations. In Fig. 3, results are presented both when the algorithm starts with the parameters of the camera set to zero, as well as



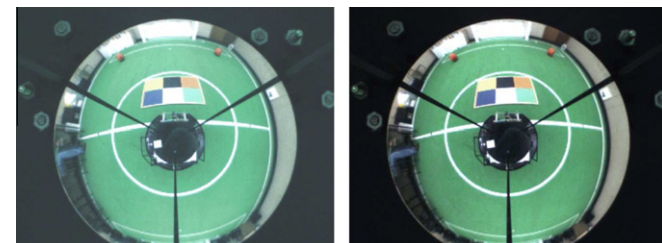


**Fig. 3.** Some experiments using the automated calibration procedure. At the top, results obtained starting with all the parameters of the camera set to zero. At the bottom, results obtained with all the parameters set to the maximum value. On the left, the initial image acquired. In the middle, the image obtained after applying the automated calibration procedure. On the right, the graphics showing the evolution of the parameters along the time.

when set to the maximum value. As can be seen, the configuration obtained after running the proposed algorithm is approximately the same, independently of the initial configuration of the camera. Moreover, the algorithm is fast to converge (it takes between 60 and 70 frames).

In Fig. 4, it is presented an image acquired with the camera in auto-mode. As can be seen, the image obtained using the camera with the parameters in auto-mode is overexposed and the white balance is not configured correctly. This is due to the fact that the camera analyzes the entire image and, as can be observed in Fig. 3, there are large black regions corresponding to the robot itself. Our approach uses a mask to select the region of interest, in order to calibrate the camera using exclusively the valid pixels. Moreover, and due to the changes in the environment when the robot is moving, leaving the camera in auto-mode leads to undesirable changes in the parameters of the camera, causing color classification problems.

Table 1 presents the values of the statistical measures described in (1)–(4), regarding the experimental results presented in Fig. 3.



**Fig. 4.** On the left, an example of an image acquired with the camera parameters in auto-mode. On the right, an image acquired after applying the automated calibration algorithm.

**Table 1**

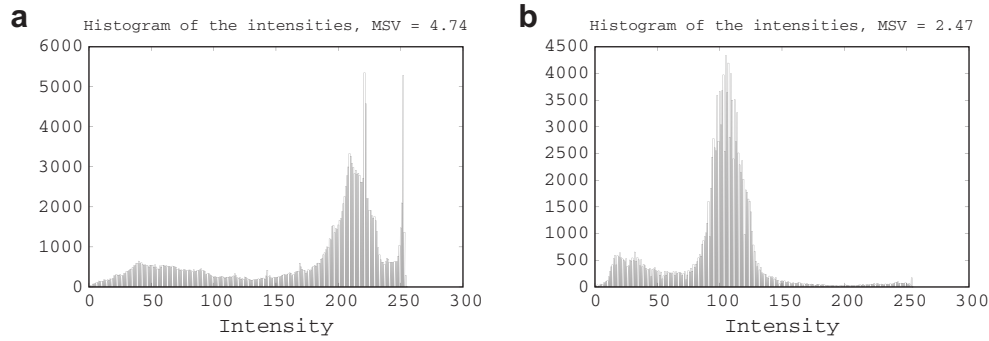
Statistical measures obtained for the images presented in Figs. 3 and 4. The initial values refer to the images obtained with the camera before applying the proposed automated calibration procedure. The final values refer to the images acquired with the cameras configured with the proposed algorithm.

Experiment	–	ACM	$\mu$	$E$	MSV
Parameters set to zero	Initial	111.00	16.00	0.00	1.00
	Final	39.18	101.95	6.88	2.56
Parameters set to maximum	Initial	92.29	219.03	2.35	4.74
	Final	42.19	98.59	6.85	2.47
Camera in auto-mode	Initial	68.22	173.73	6.87	3.88
	Final	40.00	101.14	6.85	2.54

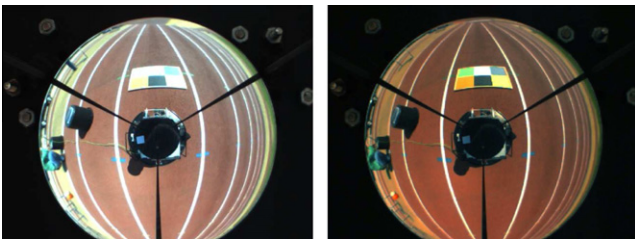
These results confirm that the camera is correctly configured after applying the automated calibration procedure, since the results obtained are close to the optimal. Moreover, the algorithm converges always to the same set of parameters, independently of the initial configuration.

According to the experimental results presented in Table 1, we conclude that the MSV measure is the best one for classifying the quality of an image. This is due to the fact that it is closer to the optimal values when the camera is correctly calibrated. Moreover, this measure can distinguish between two images that have close characteristics, as is the case when the camera is used in auto-mode.

The good results of the automated calibration procedure can also be confirmed in the histograms presented in Fig. 5. The histogram of the image obtained after applying the proposed automated calibration procedure (Fig. 5b) is centered near the intensity 127, which is a desirable property, as shown in Fig. 3 in the middle images. The histogram of the image acquired using the camera with all the parameters set to the maximum value (Fig. 5a) shows



**Fig. 5.** The histogram of the intensities of the two images presented in Fig. 4. In (a) it is shown the histogram of the image obtained with the camera parameters set to the maximum value. (b) Shows the histogram of the image obtained after applying the automated calibration procedure.



**Fig. 6.** On the left, an image acquired outdoors using the camera in auto-mode. As it is possible to observe, the colors are washed out. This happens because the camera's auto-exposure algorithm tries to compensate the black region around the mirror. On the right, the same image with the camera calibrated using our algorithm. As can be seen, the colors and the contours of the objects are much more defined.

that the image is overexposed, leading that the majority of the pixels have bright colors.

This algorithm has also been tested outdoors, under natural light. Fig. 6 shows that it works well even when the robot is under very different lighting conditions, showing its robustness.

### 3.2. Distance map calibration

For most practical applications, the setup of the vision system 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 a 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 [16]. Although tempting, this approach requires a precision mechanical setup.

In this section, we briefly present a general solution to calculate the robot centered distances map on non-SVP catadioptric setups, exploring a back-propagation ray-tracing approach and



**Fig. 7.** A screenshot of the tool developed to calibrate some important parameters of the vision system, namely the inverse distance map, the mirror and robot center and the regions of the image to be processed.

the geometric properties of the mirror surface. A detailed description of the algorithms can be found in [17] and a screenshot of the application is presented in Figs. 7 and 8. This solution effectively compensates for the misalignment that may result either from a simple mechanical setup or from the use of low cost video cameras. 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. 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. 9). The second generates a visual grid with 0.5 m 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. 10).

With this tool, it is also possible to determine some other important parameters, namely the mirror center and the area of the image that will be processed by the object detection algorithms (Fig. 11).

#### 4. Color-based object detection

The algorithms that we propose for object detection can be split into three main modules, namely the *Utility Sub-System*, the *Color Processing Sub-System* and the *Morphological Processing Sub-System*, as shown in Fig. 12. In the *Color Processing Sub-System*, proper color classification and extraction processes were developed, along with an object detection process to extract information from the acquired image, through color analysis. The *Morphological Processing Sub-System* presented in Section 5, is used to detect arbitrary FIFA balls independently of their colors.

In order to satisfy the real-time constrains in the proposed image processing system, we implemented efficient data structures to process the image data [18,19]. Moreover, we use a two-thread approach to perform the most time consuming operations

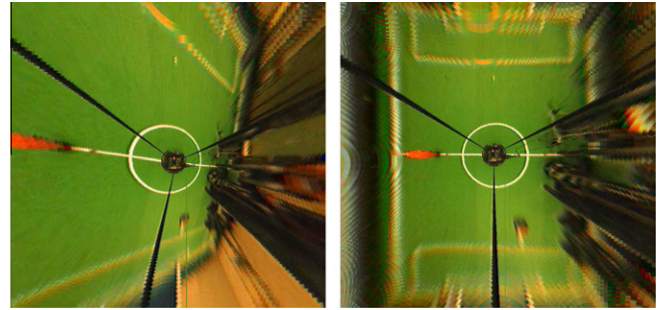


Fig. 9. Acquired image after reverse-mapping into the distance map. On the left, the map was obtained with all misalignment parameters set to zero. On the right, after automatic correction.

in parallel, namely the color classification and the color extraction, taking advantage of the dual core processor used by the laptop computers of our robots.

##### 4.1. Color extraction

Image analysis in the MSL is simplified, since objects are color coded. Black robots play with an orange ball on a green field that has white lines. Thus, the color of a pixel is a strong hint for object segmentation. We exploit this fact by defining color classes, using a look-up table (LUT) for fast color classification. The table consists of 16,777,216 entries ( $2^{24}$ , 8 bits for red, 8 bits for green and 8 bits for blue), each 8 bits wide, occupying a total of 16 MByte. Note that for other color spaces the table size would be the same, changing only the meaning of each component. Each bit expresses whether the color is within the corresponding class or not. This means that a certain color can be assigned to several classes at the same time. To classify a pixel, we first read the pixel's color and then use the color as an index into the table. The 8-bit value read from the table is called the "color mask" of that pixel.

The color calibration is performed in the HSV (Hue, Saturation and Value) color space, since it provides a single, independent, col-

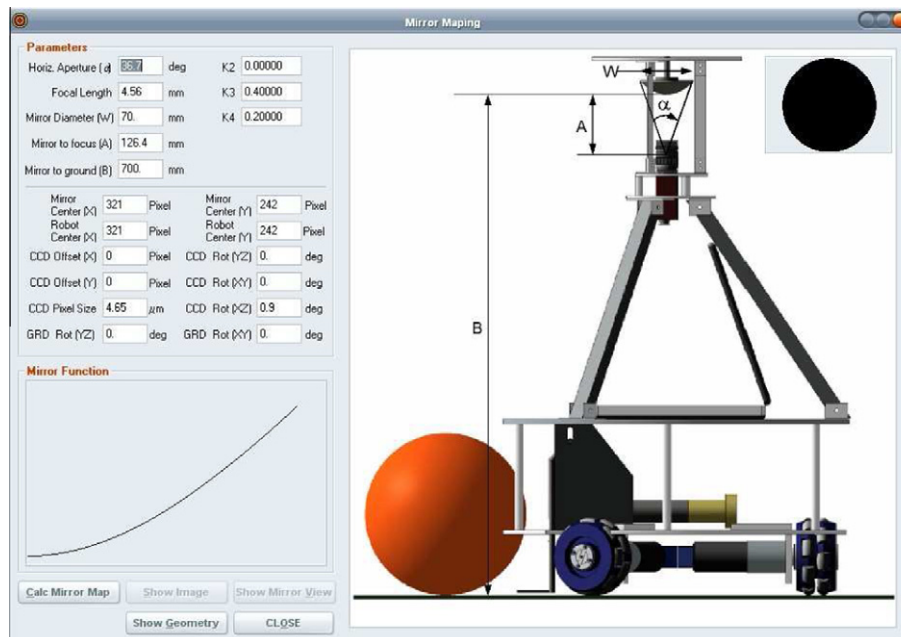
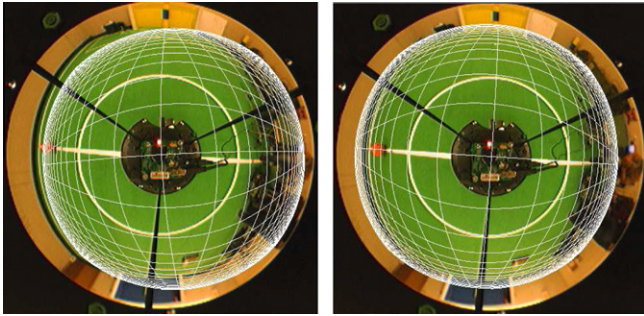
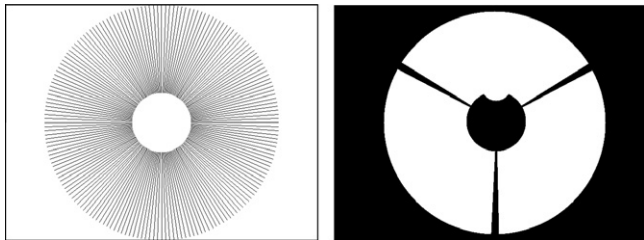


Fig. 8. A screenshot of the interface to calibrate some important parameters need to obtain the inverse distance map (these parameters are described in [17]).





**Fig. 10.** A 0.5 m 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.



**Fig. 11.** On the left, the position of the radial search lines used in the omnidirectional vision system, after detecting the center of the robot in the image using the tool described in this section. On the right, an example of a robot mask used to select the pixels to be processed, obtained with the same tool. White points represent the area that will be processed.

or spectrum variable. In the current setup, the image is acquired in RGB or YUV format and then is converted to an image of labels using the appropriate LUT. Fig. 13 presents a screenshot of the application used to calibrate the color ranges for each color class, using the HSV color space and a histogram based analysis.

Certain regions of the image are excluded from analysis. One of them is the part in the image that reflects the robot itself. Other regions are the sticks that hold the mirror and the areas outside the mirror. These regions are found using the algorithm described in Section 3.2. An example is presented on the right of Fig. 11, where the white pixels indicate the area that will be processed. With this approach, we can reduce the time spent in the conversion and searching phases and we also eliminate the problem of finding erroneous objects in those areas.

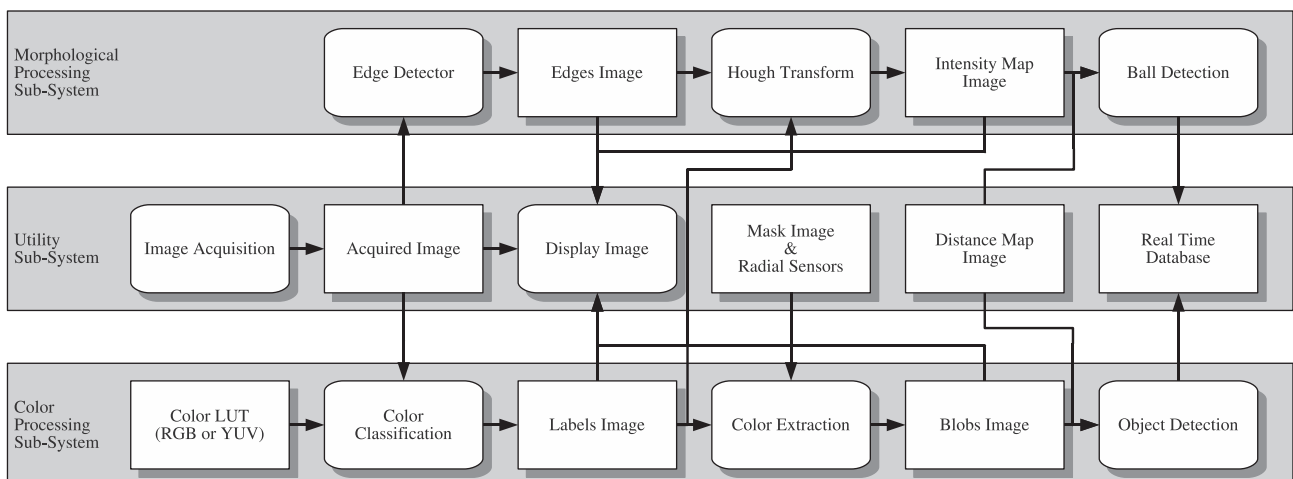
To extract color information from the image we use radial search lines, instead of processing the whole image. A radial search line is a line that starts at the center of the robot, with some angle, and ends at the limits of the image. In an omnidirectional system, the center of the robot is approximately the center of the image (see left of Fig. 11). The search lines are constructed based on the Bresenham line algorithm [20]. They are constructed once, when the application starts, and saved in a structure in order to improve the access to these pixels in the color extraction module. For each search line, we iterate through its pixels to search for transitions between two colors and areas with specific colors.

The use of radial search lines accelerates the process of object detection, due to the fact that we only process part of the valid pixels. This approach has a processing time almost constant, independently of the information that is captured by the camera. Moreover, the polar coordinates, inherent to the radial search lines, facilitate the definition of the bounding boxes of the objects in omnidirectional vision systems.

We developed an algorithm for detecting areas of a specific color which eliminates the possible noise that could appear in the image. For each radial scanline, it is performed a median filtering operation. Each time a pixel is found with a color of interest, the algorithm analyzes the pixels that follow (a predefined number). If it does not find more pixels of that color, it discards the pixel found and continues. When a predefined number of pixels with that color is found, it considers that the search line has that color.

Regarding the ball detection, we created an algorithm to recover lost orange pixels due to the ball shadow cast over itself. As soon as we find a valid orange pixel in the radial sensor, the shadow recovery algorithm tries to search for darker orange pixels previously discarded in the color segmentation analysis. The search is conducted in each radial sensor, starting at the first orange pixel found when searching towards the center of the robot, limited to a maximum number of pixels. For each pixel analyzed, a comparison is performed using a wider region of the color space, in order to being able to accept darker orange pixels. Once a different color is found or the maximum number of pixels is reached, the search along the current sensor is completed and the next sensor is processed. In Fig. 16, we can see the pixels recovered by this algorithm (the orange blobs contain pixels that were not originally classified as orange).

To accelerate the process of calculating the position of the objects, we put the color information that was found in each of the search lines into a list of colors. We are interested in the first pixel (in the corresponding search line) where the color was found and with the number of pixels with that color that have been found



**Fig. 12.** The software architecture of the omnidirectional vision system.

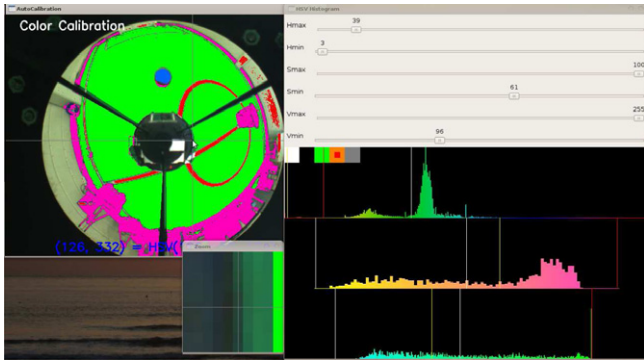


Fig. 13. A screenshot of the application used to calibrate the color ranges for each color class using the HSV color space.

in the search line. Then, using the previous information, we separate the information of each color into blobs (Fig. 16 shows an example). After this, it is calculated the blob descriptor that will be used for the object detection module, which contains the following information:

- Distance to the robot.
- Closest pixel to the robot.
- Position of the mass center.
- Angular width.
- Number of pixels.
- Number of green and white pixels in the neighborhood of the blob.

#### 4.2. Object detection

The objects of interest that are present in a MSL game are: a ball, obstacles and the green field with white lines. Currently, our system detects efficiently all these objects with a set of simple algorithms that, using the color information collected by the radial search lines, calculate the object position and/or its limits in a polar representation (distance and angle).

The algorithm that searches for the transitions between green pixels and white pixels is described next. If a non-green pixel is found in a radial scanline, we search for the next green pixel, counting the number of non-green pixels and the number of white pixels that meanwhile appeared. If these values are greater than a predefined threshold, the center of this region is considered a transition point corresponding to a position of a soccer field line. The algorithm is illustrated in Fig. 14 with an example. A similar approach has been described in [21].

The ball is detected using the following algorithm:

- (i) Separate the orange information into blobs.
- (ii) For each blob, calculate the information described previously.
- (iii) Perform a first validation of the orange blobs using the information about the green and white pixels in the neighborhood of the blob, to guarantee that only balls inside the field are detected.



Fig. 14. An example of a transition. “G” means green pixel, “W” means white pixel and “X” means pixel with a color different from green or white, for example resulting due to some noise or a not perfect color calibration. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

- (iv) Validate the remaining orange blobs according to the number of pixels. As illustrated in Fig. 15, it is known the relation between the pixel size at the ground plane and the distance to the center of the robot. Using this knowledge, we estimate the number of pixels that a ball should have according to the distance.
- (v) Following the same approach, the angular width is also used to validate the blobs.
- (vi) The ball candidate is the valid blob closest to the robot. The position of the ball is the mass center of the blob.

To calculate the position of the obstacles around the robot, we use the following algorithm:

- (i) Separate the black information into blobs.
- (ii) Calculate the information for each blob.
- (iii) Perform a simple validation of the black blobs using the information about the green and white pixels in the neighborhood of the blob, to guarantee that only obstacles inside the field are detected.
- (iv) The position of the obstacle is given by the distance of the blob relatively to the robot. The limits of the obstacle are obtained using the angular width of the blob.

More details regarding the detection and identification of obstacles can be found in [22]. Fig. 16 presents an example of an acquired image, the corresponding segmented image and the detected color blobs. As can be seen, the objects are correctly detected.

The position of the white lines, the position of the ball and the information about the obstacles are then sent to the Real-time Database [1,23] and used, afterward, by the high level process responsible for the behaviors of the robots [24,25,22,26].

#### 4.3. Experimental results

To experimentally measure the efficiency of the proposed algorithms, the robot was moved along a predefined path through the robotic soccer field, leaving the ball in a known location. The ball

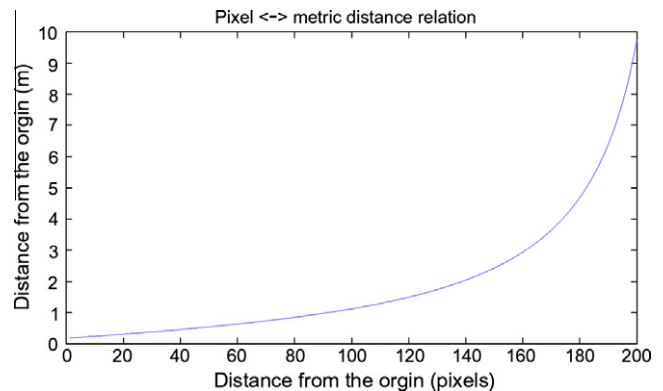
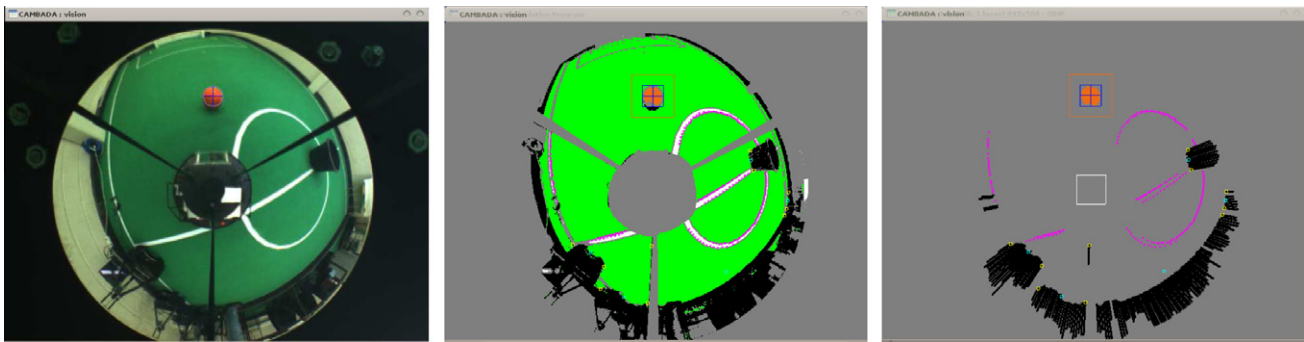


Fig. 15. Relation between pixels and metric distances. The center of the robot is considered the origin and the metric distances are considered on the ground plane.





**Fig. 16.** On the left, an example of an original image acquired by the omnidirectional vision system. In the center, the corresponding image of labels. On the right, the color blobs detected in the images. Marks over the ball point to the mass center. The several marks near the white lines (magenta) are the position of the white lines. The cyan marks are the position of the obstacles. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

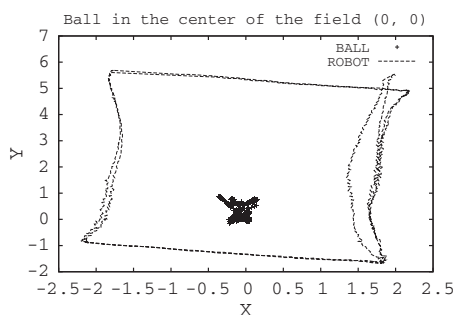
position given by the robot is then compared with the real position of the ball. Note that the results in this test may be affected by errors in the localization algorithm and by some bumps while the robot is moving. The separate study of these sources of error has been left outside this experimental evaluation. However, they should be performed, for better understanding the several factors that influence the correct localization of the ball.

The robot path across the field may be seen in Fig. 17, along with the measured ball position. According to that data, 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 the proposed algorithms. Our measures show a very high detection ratio (near 95%), and a good accuracy, with the average measures very close to the real ball position. In our experiments, we verified that the robots are able to detect the ball up to 6 m with regular light conditions and a good color calibration, easy to obtain after applying the proposed automated calibration algorithm described in Section 3.

The proposed algorithm has an almost constant processing time, independently of the environment around the robot, typically around 6 ms. It needs approximately 35 MBytes of memory. The experimental results were obtained using a camera resolution of  $640 \times 480$  pixels and a laptop with an Intel Core 2 duo at 2.0 GHz and 1 GB of memory.

## 5. Arbitrary ball detection

The color codes tend to disappear as the competition evolves, increasing the difficulty posed to the vision algorithms. The color of the ball, currently orange, is the next color scheduled to become arbitrary. In this section, we propose a solution for overcoming this



**Fig. 17.** Experimental results obtained by the omnidirectional system using the color ball detection. In this experiment, the ball was positioned in the center of the field, position (0,0). The robot performed a predefined trajectory while the position of the ball and the robot was recorded. Both axes in the graphics are in meters.

new challenge, i.e., a method for detecting balls independently of their colors. This solution is based on a morphological analysis of the image, being strictly directed to detect round objects in the field with specific characteristics, in this case the ball.

Morphological object recognition through image analysis has become more robust and accurate in the past years, whereas still very time consuming even to modern personal computers. Because RoboCup is a real-time environment, available processing time can become a serious constraint when analyzing large amounts of data or executing complex algorithms.

This section presents an arbitrary FIFA ball recognition algorithm, based on the use of image segmentation and the circular Hough transform. The processing time is almost constant and allows real-time processing. As far as we know, this approach has never been proposed. The experimental results obtained, as well as the classifications obtained by the CAMBADA team, seem to be very promising.

Regarding the vision system described in Fig. 12, it is possible to specify whether to use the *Morphological sub-system* to detect the ball or the current color-based approach. Currently, in the MSL, the shape-based detection is only necessary in the mandatory challenge of the competition, although it will be incorporated in the rules in the next years.

### 5.1. Related work

Many of the algorithms proposed during previous research work showed their effectiveness but, unfortunately, their processing time is in some cases over one second per video frame [27]. In [28], the circular Hough transform was presented in the context of colored ball detection as a validation step. However, no details about the implementation and experimental results have been presented.

Hanek et al. [29] proposed a Contracting Curve Density algorithm to recognize the ball without color labeling. This algorithm fits parametric curve models to the image data by using local criteria based on local image statistics to separate adjacent regions. This method can extract the contour of the ball even in cluttered environments under different illumination, but the vague position of the ball should be known in advance. The global detection cannot be realized by this method.

Treptow et al. [30] proposed a method to detect and track a ball without color information in real-time, by integrating the AdaBoost Feature Learning algorithm into a condensation tracking framework.

Mitri et al. [31] presented a scheme for color invariant ball detection, in which the edged filtered images serve as the input of an Adaboost learning procedure that constructs a cascade of

classification and regression trees. This method can detect different soccer balls in different environments, but the false positive rate is high when there are other round objects in the environment.

Coath et al. [32] proposed an edge-based arc fitting algorithm to detect the ball for soccer robots. However, the algorithm is used in a perspective camera vision system in which the field of view is far smaller and the image is also far less complex than that of the omnidirectional vision system used by most of the robotic soccer teams.

More recently, Lu et al. [33] considered that the ball on the field can be approximated by an ellipse. They scan the color variation to search for the possible major and minor axes of the ellipse, using radial and rotary scanning, respectively. A ball is considered if the middle points of a possible major axis and a possible minor axis are very close to each other in the image. However, this method has a processing time that can achieve 150 ms if the tracking algorithm fails, which might cause problems in real-time applications.

## 5.2. Proposed approach

The proposed approach is presented in the top layer of Fig. 12. The search for potential ball candidates is conducted taking advantage of morphological characteristics of the ball (round shape), using a feature extraction technique known as the Hough transform. This is a technique for identifying the locations and orientations of certain types of features in a digital image [34]. The Hough transform algorithm uses an accumulator and can be described as a transformation of a point in the  $x, y$ -plane to the parameter space. The parameter space is defined according to the shape of the object of interest, in this case, the ball presents a rounded shape. First used to identify lines in images, the Hough transform has been generalized through the years to identify positions of arbitrary shapes by a voting procedure [35–37].

Fig. 18 shows an example of a circular Hough transform, for a constant radius, from the  $x, y$ -space to the parameter space.

In Fig. 19, we show an example of circle detection through the circular Hough transform. We can see the original image of a dark circle (known radius  $r$ ) on a bright background (see Fig. 19a). For each dark pixel, a potential circle-center locus is defined by a circle with radius  $r$  and center at that pixel (see Fig. 19b). The frequency with which image pixels occur in the circle-center loci is determined (see Fig. 19c). Finally, the highest-frequency pixel represents the center of the circle with radius  $r$ .

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 edge detector operator. In the following, we present an explanation of this process and its implementation.

To be possible to use this image processing system in real-time, and increase time efficiency, a set of data structures to process the image data has been implemented [18,19].

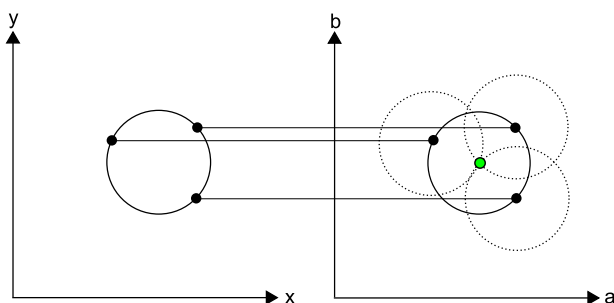


Fig. 18. The circular Hough transform.  $a$  and  $b$  represent the parameter space that in this application are the radius of the ball and the distance to robot, respectively.

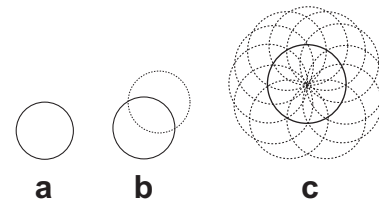


Fig. 19. Example of a circle detection through the use of the circular Hough transform.

The proposed algorithm is based on three main operations:

- (i) *Edge detection*: this is the first image processing step in the morphological detection. It must be as efficient and accurate as possible in order not to compromise the efficiency of the whole system. Besides being fast to calculate, the intended resulting image must be absent of noise as much as possible, with well defined contours, and be tolerant to the motion blur introduced by the movement of the ball and the robots.

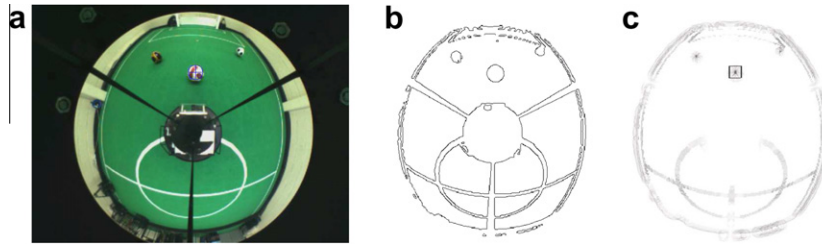
Some popular edge detectors were tested, namely Sobel [38,39], Laplace [40,41] and Canny [42]. The tests were conducted under two distinct situations: with the ball standing still and with the ball moving fast through the field. The test with the ball moving fast was performed in order to study the motion blur effect in the edge detectors, on high speed objects captured with a frame rate of 30 frames per second.

For choosing the best edge detector for this purpose, the results from the tests were compared taking into account the image of edges and processing time needed by each edge detector. On one hand, the real-time capability must be assured. On the other hand, the algorithm must be able to detect the edges of the ball independently of its motion blur effect. According to our experiments, the Canny edge detector was the most demanding in terms of processing time. Even so, it was fast enough for real-time operation and, because it provided the most effective contours, it was chosen. The parameters of the edge detector were obtained experimentally.

- (ii) *Circular Hough transform*: this is the next step in the proposed approach to find points of interest containing eventual circular objects. After finding these points, a validation procedure is used for choosing points containing a ball, according to our characterization. The voting procedure of the Hough transform is carried out in a parameter space. Object candidates are obtained as local maxima of a denoted *Intensity Image* (Fig. 20c), that is constructed by the *Hough Transform* block (Fig. 12).

Due to the special features of the *Hough* circular transform, a circular object in the *Edges Image* would produce an intense peak in *Intensity Image* corresponding to the center of the object (as can be seen in Fig. 20c). On the contrary, a non-circular object would produce areas of low intensity in the *Intensity Image*. However, as the ball moves away, its edge circle size decreases. To solve this problem, information about the distance between the robot center and the ball is used to adjust the Hough transform. We use the inverse mapping of our vision system [17] to estimate the radius of the ball as a function of distance.

- (iii) *Validation*: in some situations, particularly when the ball is not present in the field, false positives might be produced. To solve this problem and improve the ball information reliability, we propose a validation algorithm that discards false



**Fig. 20.** Example of a captured image using the proposed approach. The cross over the ball points out the detected position. In (b) the image (a), with the Canny edge detector applied. In (c), the image (b) after applying the circular Hough transform.

positives based on information from the *Intensity Image* and the *Acquired Image*. This validation algorithm is based on two tests against which each ball candidate is put through.

In the first test performed by the validation algorithm, the points with local maximum values in the *Intensity Image* are considered if they are above a distance-dependent threshold. This threshold depends on the distance of the ball candidate to the robot center, decreasing as this distance increases. This first test removes some false ball candidates, leaving a reduced group of points of interest.

Then, a test is made in the *Acquired Image* over each point of interest selected by the previous test. This test is used to eliminate false balls that usually appear in the intersection of the lines of the field and other robots (regions with several contours). To remove these false balls, we analyze a square region of the image centered in the point of interest. We discard this point of interest if the sum of all green pixels is over a certain percentage of the square area. Note that the area of this square depends on the distance of the point of interest to the robot center, decreasing as this distance increases. Choosing a square where the ball fits tightly makes this test very effective, considering that the ball fills over 90% of the square. In both tests, we use threshold values that were obtained experimentally.

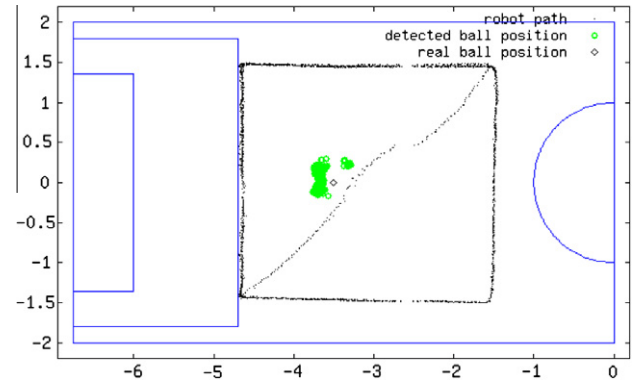
Besides the color validation, it is also performed a validation of the morphology of the candidate, more precisely a circularity validation. Here, from the candidate point to the center of the ball, it is performed a search of pixels at a distance  $r$  from the center. For each edge found between the expected radius, the number of edges at that distance are determined. By the size of the square which covers the possible ball and the number of edge pixels, it is calculated the edges percentage. If the edges percentage is greater than 70, then the circularity of the candidate is verified.

The position of the detect ball is then sent to the Real-time Database, together with the information of the white lines and the information about the obstacles to be used, afterward, by the high level process responsible for the behaviors of the robots.

### 5.3. Experimental results

Fig. 20 presents an example of the *Morphological Processing Sub-System*. As can be observed, the balls in the *Edges Image* (Fig. 20b) have almost circular contours. Fig. 20c) shows the resulting image after applying the circular Hough transform. Notice that the center of the balls present a very high peak when compared to the rest of the image. The ball considered was the closest to the robot, due to the fact that it has the high peak in the image.

To ensure good results in the RoboCup competition, the system was tested with the algorithms described above. For that purpose, the robot was moved along a predefined path through the robotic soccer 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 errors in the



**Fig. 21.** Experimental results obtained by the omnidirectional system using the morphological ball detection. In this experience, the ball was positioned in the penalty mark of the field. The robot performed a predefined trajectory while the position of the ball was recorded. Both axes in the graphics are in meters.

localization algorithm and by the robot bumps while moving. These external errors are out of the scope of this study.

The robot path in the field may be seen in Fig. 21, 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 accuracy of the proposed algorithms. We obtained a very high detection ratio (near 90%) and a false positive rate around 0%, which is a very significant result. With the proposed approach, the omnidirectional vision system can detect the ball within this precision until distances up to 4 meters.

The average processing time of the proposed approach was approximately 16 ms. It needs approximately 40 MBytes of memory. The experimental results have been obtained using a camera resolution of  $640 \times 480$  pixels and a laptop with an Intel Core 2 duo at 2.0 GHz.

## 6. Conclusions

This paper presents the omnidirectional vision system developed for the CAMBADA MSL robotic soccer team, from the calibration to the object detection. We presented several algorithms for the calibration of the most important parameters of the vision system and we proposed efficient color-based algorithms for object detection. Moreover, we proposed a solution for the detection of arbitrary FIFA balls, one of the current challenges in the MSL.

The CAMBADA team won the last three editions of the Portuguese Robotics Festival, ranked 5th in RoboCup 2007, won the RoboCup 2008 and ranked 3rd in RoboCup 2009, demonstrating the effectiveness of our vision algorithms in a competition environment. As far as we know, no previous work has been published describing all the steps of the design of an omnidirectional vision system. Moreover, some of the algorithms presented in this paper



are state-of-the-art, as demonstrated by the first place obtained in the mandatory technical challenge in RoboCup 2009, where the robots have to play with an arbitrary standard FIFA ball.

We are currently working in the automatic calibration of the inverse distance mapping and in efficient algorithms for autonomous color calibration, based on region growing. Regarding the object detection algorithms, as we have reduced the processing time to a few milliseconds, we are working on the acquisition of higher resolution images, capturing only a region of interest. The idea of work with higher image resolutions is to improve the object detection at higher distances. Moreover, we continue the development of algorithms for shape-based object detection, also to incorporate as a validation of the color-based algorithms.

### Acknowledgment

This work was supported in part by the FCT (Fundação para a Ciência e a Tecnologia).

### References

- [1] Neves A, Azevedo J, Cunha NLB, Silva J, Santos F, Corrente G, et al. CAMBADA soccer team: from robot architecture to multiagent coordination. In: Vladan Papić editor. *Robot soccer*. Vienna, Austria: I-Tech Education and Publishing; 2010 [chapter 2].
- [2] Zivkovic Z, Booi O. How did we built our hyperbolic mirror omni-directional camera-practical issues and basic geometry. Tech. rep. Intelligent Systems Laboratory, University of Amsterdam; 2006.
- [3] Wolf J. Omnidirectional vision system for mobile robot localization in the robocup environment. Master's thesis. Graz University of Technology; 2003.
- [4] Menegatti E, Nori F, Pagello E, Pellizzari C, Spagnoli D. Designing an omnidirectional vision system for a goalkeeper robot. In: Proc of RoboCup 2001. Lecture notes in computer science, vol. 2377. Springer; 2001. p. 78–87.
- [5] 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. p. 2487–93.
- [6] Lima P, Bonarini A, Machado C, Marchese F, Marques C, Ribeiro F, et al. Omnidirectional catadioptric vision for soccer robots. *Robot Auton Syst* 2001;36(2–3):87–102.
- [7] Liu F, Lu H, Zheng Z. A robust approach of field features extraction for robot soccer. In: Proc of the 4th IEEE Latin America robotic symposium, Monterey, Mexico; 2007.
- [8] Lu H, Zheng Z, Liu F, Wang X. A robust object recognition method for soccer robots. In: Proc of the 7th world congress on intelligent control and automation, Chongqing, China; 2008.
- [9] Voigttrande 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.
- [10] Mayer G, Utz H, Kraetzschmar G. Playing robot soccer under natural light: a case study. In: Proc of the RoboCup 2003. Lecture notes on artificial intelligence, vol. 3020. Springer; 2003.
- [11] Krawczyk G, Gosele M, Seidel H. Photometric calibration of high dynamic range cameras. Research Report MPI-I-2005-4-005. Max-Planck-Institut für Informatik, Stuhlsatztenhausweg 85, 66123 Saarbrücken, Germany; April 2005.
- [12] Shirvaikar MV. An optimal measure for camera focus and exposure. In: Proc of the IEEE southeastern symposium on system theory, Atlanta (USA); 2004.
- [13] Nourani-Vatani N, Roberts J. Automatic camera exposure control. In: Proc of the 2007 Australasian conference on robotics and automation, Brisbane, Australia; 2007.
- [14] Åström K, Hägglund T. PID controllers: theory, design, and tuning. 2nd ed. Instruments Society of America; 1995.
- [15] Neves AJR, Cunha AJPB, Pinheiro I. Autonomous configuration of parameters in robotic digital cameras. In: Proc of the 4th Iberian conference on pattern recognition and image analysis, IBERIA-2009. Lecture notes in computer science, vol. 5524. Póvoa do Varzim, Portugal: Springer; 2009. p. 80–7.
- [16] Baker S, Nayar SK. A theory of single-viewpoint catadioptric image formation. *Int J Comput Vis* 1999(2):175–96.
- [17] Cunha B, Azevedo JL, Lau N, Almeida L. Obtaining the inverse distance map from a non-SVP hyperbolic catadioptric robotic vision system. In: Proc of the RoboCup 2007. Lecture notes in computer science, vol. 5001. Atlanta (USA): Springer; 2007. p. 417–24.
- [18] Neves AJR, Martins DA, Pinho AJ. 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. p. 51–5.
- [19] Neves AJR, Corrente G, Pinho AJ. An omnidirectional vision system for soccer robots. In: Proc of the 2nd international workshop on intelligent robotics, IROBOT 2007. Lecture notes in artificial intelligence, vol. 4874. Springer; 2007. p. 499–507.
- [20] Bresenham JE. Algorithm for computer control of a digital plotter. *IBM Syst J* 1965;4(1):25–30.
- [21] Merke A, Welker S, Riedmiller M. Line base robot localisation under natural light conditions. In: Proc of the ECAI workshop on agents in dynamic and real-time environments, Valencia, Spain; 2002.
- [22] Silva J, Lau N, Rodrigues J, Azevedo JL, Neves AJR. Sensor and information fusion applied to a robotic soccer team. In: RoboCup 2009: robot soccer world cup XIII. Lecture notes in artificial intelligence. Springer; 2009.
- [23] Almeida L, Santos F, Facchinetti T, Pedreira P, Silva V, Lopes LS. Coordinating distributed autonomous agents with a real-time database: the CAMBADA project. In: Proc of the 19th international symposium on computer and information sciences, ISIS 2004. Lecture notes in computer science, vol. 3280. Springer; 2004. p. 878–86.
- [24] Lau N, Lopes LS, Corrente G, Filipe N. Roles, positionings and set plays to coordinate a msl robot team. In: Proc of the 4th international workshop on intelligent robotics, IROBOT'09. Lecture notes in computer science, vol. 5816. Aveiro, Portugal: Springer; 2009. p. 323–37.
- [25] Lau N, Lopes LS, Corrente G, Filipe N. Multi-robot team coordination through roles, positioning and coordinated procedures. In: Proc of the IEEE/RSJ international conference on intelligent robots and systems, MO, USA: St. Louis; 2009. p. 5841–48.
- [26] Silva J, Lau N, Neves AJR, Rodrigues J, Azevedo JL. Obstacle detection, identification and sharing on a robotic soccer team. In: Proc of the 4th international workshop on intelligent robotics, IROBOT'09. Lecture notes in computer science, LNCS 5816. Aveiro, Portugal: Springer; 2009. p. 350–60.
- [27] 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. p. 125–30.
- [28] Jonker P, Caarls J, Bokhove W. Fast and accurate robot vision for vision based motion. In: RoboCup 2000: robot soccer world cup IV. Lecture notes in computer science. Springer; 2000. p. 149–58.
- [29] Hanek R, Schmitt T, Buck S. Fast image-based object localization in natural scenes. In: Proc of the 2002 IEEE/RSJ international conference on intelligent robotics and systems, Lausanne, Switzerland; 2002. p. 116–22.
- [30] Treptow A, Zell A. Real-time object tracking for soccer-robots without color information. *Robot Auton Syst* 2004;48(1):41–8.
- [31] Mitri S, Pervolz K, Surmann H, Nuchter A. Fast color independent ball detection for mobile robots. In: Proc of the 2004 IEEE international conference on mechatronics and robotics, Aachen, Germany; 2004. p. 900–5.
- [32] Coath G, Musumeci P. Adaptive arc fitting for ball detection in RoboCup. In: Proc of the APRS workshop on digital image computing, WDIC 2003, Brisbane, Australia; 2003. p. 63–8.
- [33] Lu H, Zhang H, Zheng Z. Arbitrary ball recognition based on omni-directional vision for soccer robots. In: Proc of RoboCup 2008; 2008.
- [34] Nixon M, Aguado A. Feature extraction and image processing. 1st ed. Linacre House, Jordan Hill, Oxford OX2 8DP 225 Wildwood Avenue, Woburn, MA 01801-2041: Reed Educational and Professional Publishing Ltd.; 2002.
- [35] Ser PK, Siu WC. 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, vol. 5; 1993. p. 9–12.
- [36] Zhang YJ, Liu ZQ. Curve detection using a new clustering approach in the hough space. In: IEEE international conference on systems, man, and cybernetics, 2000, vol. 4; 2000. p. 2746–51.
- [37] Grimson WEL, Huttenlocher DP. On the sensitivity of the hough transform for object recognition. *IEEE Trans Pattern Anal Mach Intell* 1990;12:1255–74.
- [38] 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. p. 291–3.
- [39] Zin TT, 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. p. 310.
- [40] 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, vol. 1; 1997. p. 417–9.
- [41] 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, vol. 3; 2000. p. 444–7.
- [42] Canny JF. A computational approach to edge detection. *IEEE Trans Pattern Anal Mach Intell* 8 (6).