Omni-Directional Catadioptric Vision for Soccer Robots

P. Lima [a,*] A. Bonarini [c,2] C. Machado [b] F. M. Marchese [d,2]
C. Marques [a,1] F. Ribeiro [b] D. G. Sorrenti [d,2]

[a] *Instituto de Sistemas e Robótica, Instituto Superior Técnico - Av. Rovisco Pais, 1 - 1049-001 Lisboa, PORTUGAL*

[b] *Grupo de Automação e Robótica – Deapartamento de Electrónica Industrial – Universidade do Minho – Campos de Azurém – 4800 Guimarães, PORTUGAL*

[c] *Politecnico di Milano AI and Robotics Project, Dipartimento di Elettronica e Informazione, Politecnico di Milano, Piazza Leonardo da Vinci 32, I20133 Milano, Italy; http://www.elet.polimi.it/people/bonarini*

[d] *Dipartimento di Informatica, Sistemistica e Comunicazione, Università degli Studi di Milano - Bicocca, via Bicocca degli Arcimboldi 8, I20126, Milano, ITALY; http://www.ira.disco.unimib.it*

1 Introduction and State-of-the-Art

Omni-directional catadioptric vision systems have been around for years [1]. Using a suitable combination of lenses and mirrors, these systems, when assembled on a mobile robot, considerably enlarge the field of view of the imaging system.

There are many different ways of assemblying a camera on a robot:

[*] Please send all the correspondence to this author.
Email addresses: pal@isr.ist.utl.pt (P. Lima), bonarini@elet.polimi.it (A. Bonarini), carlos.machado@dei.uminho.pt (C. Machado), marchese@disco.unimib.it (F. M. Marchese), cmarques@isr.ist.utl.pt (C. Marques), Fernando.Ribeiro@dei.uminho.pt (F. Ribeiro), sorrenti@disco.unimib.it (D. G. Sorrenti).

[1] This work was supported by grant PRAXIS XXI BM /21091 /99 of the Portuguese Foundation for Science and Technology.

[2] This research is partially supported by the project “PRASSI”, co-funded by the Italian Ministry of University and Scientific and Technological Research through the ENEA.

- **Fixed camera pointing to the front of the robot**: in this case a typical image that can be seen from the robot is depicted in Fig. 1. The main disadvantages of this solution results from the limited amount of available information, and from the increasing occurence of occlusions of the scene background due to nearby objects.

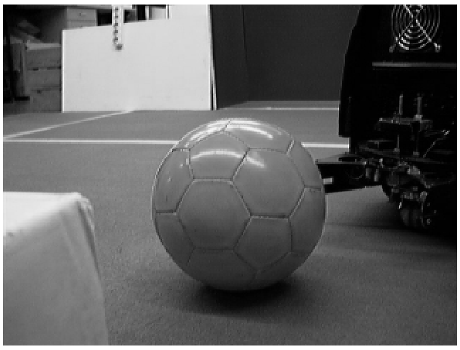


Fig. 1. Image seen by a robot with a camera pointing to its front.

- **Motorized camera**: this results from assemblying the camera on a structure linked to a motor. The field of view is increased by moving the camera up and down (tilt) and/or left and right (pan). The main problems here are due to the time required to move the camera.
- **More than one camera**: more than one camera can be used, on one hand, to achieve stereo vision and determine the distance to relevant objects; on the other hand, to watch different spots around the robot. This is however a costly solution. Furthermore, reliability is decreased due to the increasing number of devices and the power consumption can be considerable.

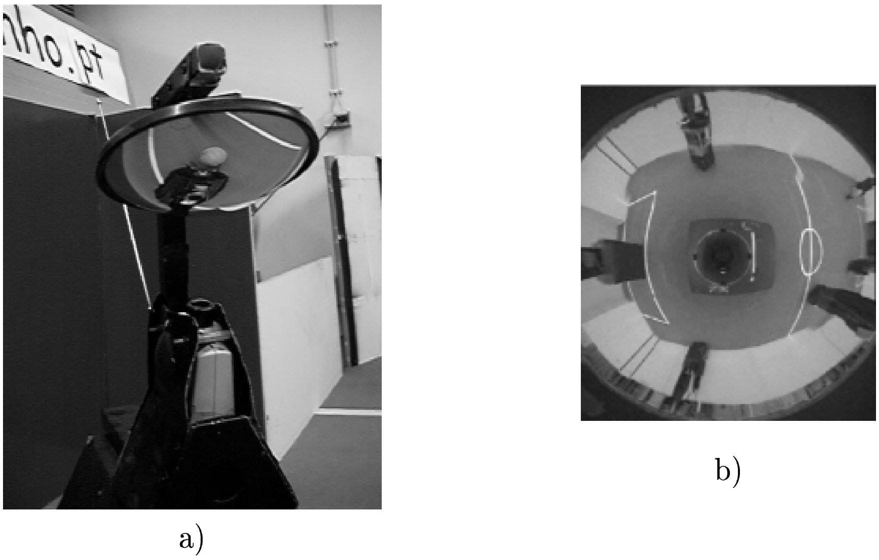


Fig. 2. a) Omni-directional catadioptric vision system consisting of a camera and a parabolic mirror; b) An image taken by a conical-spherical sensor.

- **Fixed camera pointed to one or more mirrors**: these belong to the class of solutions known as catadioptric vision systems. One important example are the omni-directional vision systems, based on a camera pointing upwards to a convex mirror (see Fig. 2-a)). Different mirrors profiles can be used, such as conic mirrors, parabolic mirrors or spherical mirrors, to name a few. If the mirror profile is precisely known, the image can be unwarped with a suitable transformation. Another solution is to design mirrors which unwarp the image directly, saving CPU time [2] [3]. The main disadvantage of omni-directional catadioptric vision systems is the distortion, on the image, of the shape of relevant objects in the observed scene. Nevertheless, if the information to be extracted from the image is only the relative orientation with respect to relevant objects, distortion is irrelevant, since the angles between radial lines are preserved [4].

Another potential problem is the support type used for the mirror. The support must be carefully chosen, since it may introduce further distortion and/or occlusion (e.g., in Fig. 2-a) the image is partially occluded by the supporting structure of the mirror and by the camera itself). An example of image captured by an omni-directional mirror can be seen in Fig. 2-b).

This paper focus on the design and use of omni-directional catadioptric vision systems for soccer robots. In the RoboCup-Soccer competitions, the field features are mainly distinguishable by their color (e.g., the field is green with white lines, the goals are blue and yellow, the ball is orange), hence vision is a sensor naturally shared by all participant teams.

In the middle-size league of RoboCup-Soccer, the teams are composed of fully autonomous robots, with no global view of the field and most, if not all, processing done on board. Among those, an increasing number of teams is using omni-directional catadioptric vision, so that many different important game features can be seen at once whenever an image frame is acquired. In the paper we describe the approach to omni-directional vision in the middle-size league of RoboCup-Soccer by three such teams:

- *ART Team*, partially represented here by the Politecnico di Milano and the Università degli Studi di Milano – Bicocca, Italy, and
- *Patriarcas*, from the University of Minho, Portugal,
- *ISocRob*, from the Instituto Superior Técnico, Portugal.

Three main topics are covered by the paper:

- The design of a multi-part omni-directional mirror.
- Virtual sensors to extract important environment features from the image.
- Omni-directional vision-based self-localization.

Each of the above groups concentrated on one of the topics (listed in the same order). This paper aims at demonstrating that an integration of the work done, based on the described catadioptric vision system with a multi-part mirror, is possible. Nevertheless, the results presented were obtained with mirrors separately designed by the different groups, each corresponding to particular parts of the multi-part mirror.

In the literature, different mirror geometries have been proposed [5] [6] and even in RoboCup-Soccer middle-size league some teams already used mirrors [7] [8] [9] with profiles other than the original conical one. In 1999, the first multi–part mirror designed to obtain specific properties of the image was presented at RoboCup [10] [11].

Many researchers have used several distinct approaches to self-localization in either indoors and outdoors environments, and either using natural or artificial environment landmarks [12]. One currently popular approach are the so-called Markov Localization methods [13,14].

An increasing number of teams participating in RoboCup-Soccer middle-size league is approaching the self-localization problem. The proposed solutions are mainly distinguished by the type of sensors used: Laser Range Finders (LRFs), vision-based omni-directional sensors and single frontal camera. The CS-Freiburg and Stuttgart-Cops teams can determine their position with an accuracy of 1 and 5 cm, respectively, using LRFs [15]. However, LRFs require walls surrounding the soccer field to acquire the field border lines and, in a sense, correlate them with the field rectangular shape to determine the team postures. Other teams propose a vision based approach to self- localization based on a single frontal camera, used to match a 3D geometric model of the field with the border line segments and goal lines in the acquired image [16] [17]. RoboCup's Agilo team [16] proposes a single frontal camera to match a 3-D geometric model of the field with the border lines and goals line segments in the acquired image. Only a partial field view is used in this method. Iocchi and Nardi [17] also use a single frontal camera and match the lines with a field model using the Hough Transform. Even though similar to the work on vision-based self-localization described in this paper, their approach considers lines detected locally (again due to a partial field view), rather than a global field view, and requires odometry to remove ambiguities. The robots of the Tuebingen team use omni-directional vision for self-localization, but only the distance to the walls is used [18]. Several teams use a vision-based omni-directional catadioptric system similar to the one described here, but only for ball and opposing robots tracking.

Omni-directional Vision-based approaches to self-localization have been used already outside RoboCup. One such approach is described in [19], where the authors use a conic mirror to implement a catadioptric vision system that extracts radial straight lines from the surrounding environment, and an Extended Kalman Filter to integrate the localization data so-obtained by triangulation with odometry.

The paper is organized as follows: in Section 2, the design of the multi-part omni-directional mirror is described. Applications to robotic soccer based on omni-directional catadioptric vision systems, which can use the different parts of the multi-part mirror are introduced in Sections 3 and 4: virtual sensors that locate relevant objects/landmarks in the scene and a self-localization algorithm, respectively. Finally, some conclusions and a description of envisaged future work are drawn in Section 5.

2 Designing a Multi-Part Omni-Directional Mirror

An omni-directional vision sensor should allow the robot to observe the relevant parts of the scene. The relevance of a feature derives from the application, in our case, of the rules and aims of RoboCup-Soccer middle-size league games. By analyzing them, it is possible to define a set of requirements for an effective perception system.

2.1 Inferring Requirements for the Perception System from RoboCup Rules

An omni-directional perception system should be able to detect points of interest (direction and distance) with the accuracy required by the application. The following requirements and applications have been identified in the RoboCup domain:

- When the point of interest is in contact or very near to the robot, a very good accuracy is required for both direction and distance, in order to properly control the robot motion. An example is the control of ball kicking.
- When the point of interest is within a few meters from the robot, a good accuracy is required for both direction and distance. Moreover, it is very useful that the measures, taken at different distances in this intermediate range, to be affected by the smallest amount of inaccuracy, independently from the distance. An example is self-localization which, basing on localization of known points, would be eased if inaccuracy is small.
- When the point is quite far, a good accuracy is required for the direction, less accuracy may be accepted for the distance. An example is moving to the ball: the directional accuracy is required in order to be able to head towards it.
- The last requirement deals with the markers, which allow to distinguish team-mates from opponents. To detect the markers at any distance implies the perception system to be able to observe in a given vertical angular sector.

In 1999, a mirror was designed [10] [11] only partially matching these requirements. The aims were both to have enough resolution to detect and localize the ball even when observed at the farthest distance, and to include in the image the maximum part of the ball when it is close to the robot body. These requirements could not be matched by any of the classical mirror shapes used till then, and we decided to implement a 2-part mirror. The first part was a conical mirror and the second one a spherical apex, sharing a common tangent at the intersection points. The spherical part projected scene points at the playground level up to 1.5 m from the sensor, thus allowing the angle of the conical part to be steep enough to observe points distant up to 6 m from the sensor (see Fig. 2-b)). The sensor, implemented with a large, low-cost mirror (18.5 cm of diameter) and a low cost camera, was good enough to make it possible the implementation of successful behaviors [11].

Since then, all requirements have been taken into account and the perception system was redesigned. We decided to develop a new design methodology to implement a new set of mirrors based on a comprehensive analysis of the above requirements and satisfying them through an accurate control of the distribution of the image resolution [3]. The control of the distribution can be implemented at two different levels, which will be explained in the next sections: at pixel level or at part level. The selection of the level depends on the specific requirements under consideration.

2.2 Isometric Mirror Part

Conventional conical mirrors introduce large distortions in the distance of objects at the playground level. This distortion grows quickly with the distance from the object to the observer. On one hand, it is quite obvious that the nominal value of the estimates can be easily corrected given the profile function of the mirror. On the other hand, the accuracy of these measurements is corrupted by the joint effect of such distortion and image sampling, without any possibility to compensate for it. The accuracy degradation implied by conventional mirrors conflicts with the requirement of a reasonably limited amount of inaccuracy for any distance measure in the intermediate range. Therefore, one of the objectives of this work was to develop an optical compensation of the above-described distortion, working directly on the mirror profile in such a way that the absolute localization error remains limited with respect to the object distance. In other words, the driving idea was to control the distribution of the image resolution on a pixel basis, in order to get the desired accuracy. The analytical setup for this optical compensation turned out to be very similar to previous work [2] (see also [20]) where the aim was to exploit *reflective surfaces as computational sensors*. This optical compensation results in a constant absolute error in the distance measurement. The transformation between two 2D Euclidean spaces (playground and sensor) performed by such camera/mirror system, keeps angles unchanged and changes lengths by a constant factor. This transformation, being linear, does not change the metric tensor, neglecting the constant. Therefore, we call *isometric* this kind of mirror because of its capabilities to keep the image metric, property that does not hold for conventional mirrors.

An even more relevant point driving our design concerns the detection of image features. The proposed design has the effect of keeping constant the image size of the scene features at the playground level, inside the covered range of distances. This makes less likely a detection failure when the feature is far from the observer.

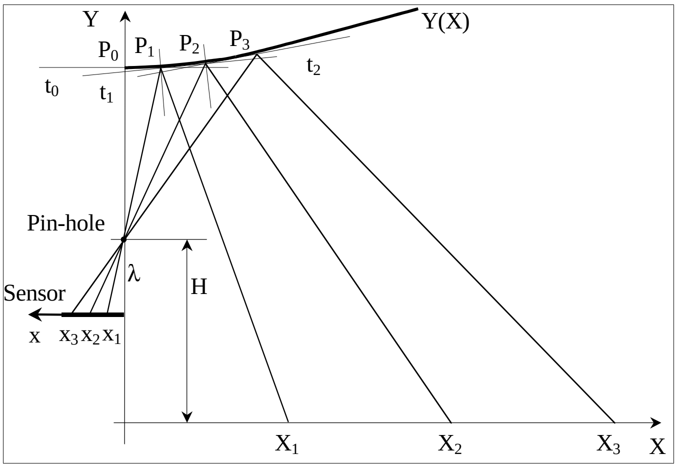


Fig. 3. Sketch for inferring the differential equation generating the isometric part of the mirror.

The design problem is modeled by the following differential equation 1, which can be inferred by applying the laws of Linear Optics (see Fig. 3).

$$\frac{\frac{X}{Y} + \frac{2Y'}{1-Y'^2}}{1 - \frac{X}{Y}\frac{2Y'}{1-Y'^2}} = \frac{\eta Y - X^2}{X(Y+H)'}; \quad \begin{cases} Y(0) = Y_0 \\ Y'(0) = 0 \end{cases} \tag{1}$$

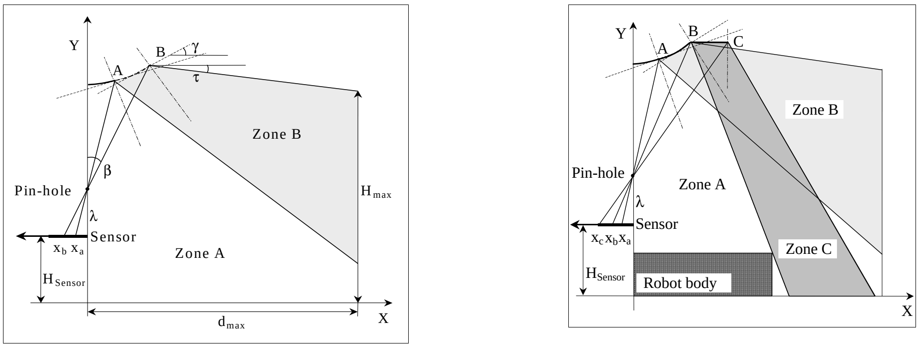
where: $Y' = dY/dX$, $\eta = k\,\lambda$, λ is the focal length, k is the proportionality constant from X to x, H is the pin-hole height from the playground. Differently from [2], [20] we developed a geometrical integration of equation 1. Our approach is based on a local first order approximation of the profile: at each point the mirror has been approximated by its tangent space. The resulting profile looks quite similar to the one obtained in [2]. It is convex into its first half, i.e. the part that goes from the axis of symmetry toward the outside of the mirror; then it has an inflection point and finally it gets slightly concave. Establishing point by point the relationship between the mirror profile and the scene is what we call the *pixel-level* control of the distribution of the image resolution.

2.3 Constant Curvature Mirror Part

It would be desirable if the above described design approach could cover the whole range of distances required for the RoboCup purposes, but the use of conventional low-cost color cameras does not allow a reliable detection of relevant features on the whole range of distances. Fortunately, for the farthest range of distances as well as for the marker detection and localization, it is not required an accuracy as high as for the other requirements. Notice that these requirements can be mapped jointly just covering a vertical angular sector with only one part of the image. Establishing the amount of image resolution devoted to each part is what we call the *part-level* control of the distribution of the image resolution. Notice that it is worthwhile to preserve the continuity between the two portions of the image in order to ease the association of the robot body to its marker, when they are across the two parts. Such image continuity can be granted by imposing the continuity of the tangent at the junction between the isometric and the new part of the mirror (point A in Fig. 4-a)). Another condition comes from fixing point $B = (X_B, Y_B)$ and setting the height H_{max} so that it can be observed at distance d_{max}. This constraint gives the tangent to the profile in point B.

$$tan(\theta) = \frac{(H_{max} - Y_B)}{(d_{max} - X_B)}, \quad tan(\beta) = \frac{x_b}{\lambda}, \quad tan(\gamma) = tan(\frac{(\beta + \theta - \pi)}{2}), (2)$$

where: λ is the focal length.



a) b)

Fig. 4. a) Sketch for the design of the constant curvature part of the mirror; b) Sketch for the design of the planar part of the mirror.

Because there is no other constraint, this portion of the mirror can be designed, e.g., by imposing a constant variation of the tangent between the two endpoints. Hence the name ”constant curvature” given to this part of the mirror. Notice that the relatively simple design problem makes not necessary to introduce a pixel-level control of the distribution of the image resolution. The mirror will cover completely the highest part of the scene (Zone B). On the other hand, when the robots are quite near, they will be observed by the first part of the mirror (Zone A).

2.4 Planar Mirror Part

The so far designed mirror does not satisfy the requirement concerning the nearer range of distances. Due to the robot occlusion (see Fig. 4-b)), it is not possible to observe the scene immediately close to the robot. The relatively small image of a feature, when very near and imaged in the isometric part, results in a less than required accuracy, while the highest should be obtained in the very close range. To satisfy the requirement, a third part of mirror has been introduced. This part should be the outmost to suffer the least occlusion from the robot body. The simplest solution to this design problem is a planar mirror lying on a plane perpendicular to the rotational axis. The height of this part has to be as low as possible, with respect to the camera, in order to give the largest images of the features. At the same time, this part should not be on the line sight of others. Hence the choice has been to have a planar mirror at the same height of the last point of the constant curvature part of the mirror (point B). The point C is set as follows:

$$\frac{X_C}{(Y_C - \lambda - H_{Sensor})} = \frac{x_c}{\lambda} \quad Y_C = Y_B, \tag{3}$$

where H_{Sensor} is the height of sensor plane and λ is the focal length. The ball image produced by this part is large enough to allow a reliable detection and an accurate localization. Notice that the relatively simple design problem makes not necessary to introduce a pixel-level control of the distribution of the image resolution.

2.5 The Resulting Mirror

The mirror profile resulting from the above described design is shown in Fig. 5-a). It enables the system to observe up to 6 m far away without image distortion at the playground level; thanks to the constant curvature part it can observe up to the maximum height, 600 mm, at the maximum distance in the playground (11.2 m). Its outer part allows the observation of objects from 0.39 m to 0.51 m. The last prototype of the mirror is depicted in Fig. 5-b); an image obtained by this mirror and a very low-cost camera is shown in Fig. 6. You may notice that such image have been collected after a rough mechanical setup. This activity should have aligned optical and mirror axis, put the mirror at the designed distance from the pin-hole, etc. It is extremely likely that some defect is still present on the image.

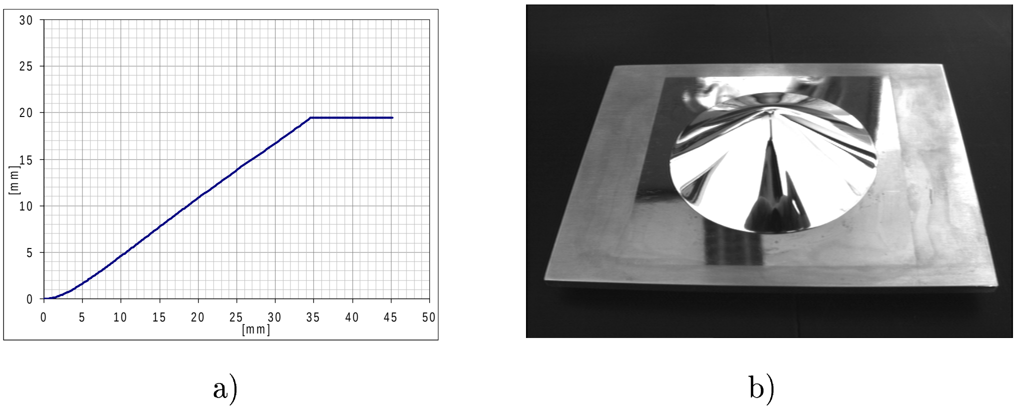
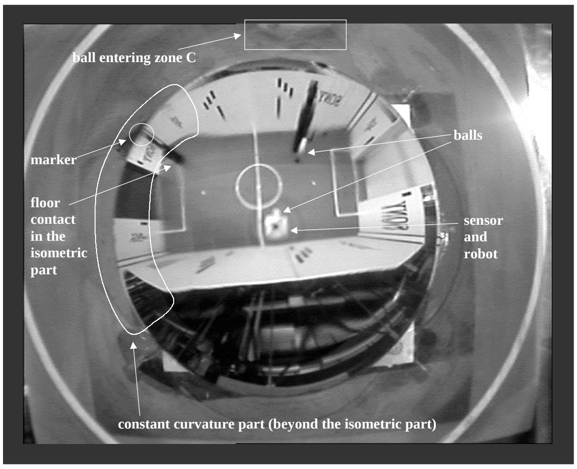


Fig. 5. a) Profile of the overall mirror; b)The last mirror prototype.



ball entering zone C
marker
floor contact in the isometric part
balls
sensor and robot
constant curvature part (beyond the isometric part)

Fig. 6. Image taken with the robot near to the center of the playground (Melbourne, 31.08.00, field B of the initial tournament). Notice the effect of the isometric optical compensation, which lasts up to 6 m; in the constant curvature part it is possible to detect a goal and the marker. However, their distances from the observer, thanks to the continuity with the first part, can be measured at the playground contact point with the limited error provided by the isometric design. Notice also the dimension of the farther ball, which is even larger that when nearby because of the isometric property holding at the floor level only. There was no marker on the robot besides the farther ball.

3 Virtual Image Sensors for Robotic Soccer

Many visual features are important in the RoboCup-Soccer domain. A set of virtual image sensors was designed to extract a crucial subset of those features for middle-size league robots, and handle the necessary actions, namely:

- other robots and walls, for obstacle avoidance,
- goals,
- far ball, to move towards it,
- near ball, to kick it,
- catadioptric system calibration.

Those virtual sensors can be used with any catadioptric vision system, particularly the one described in the precedent section. This section goes through their implementation details for a parabolic mirror that was built to test only the virtual sensors.

3.1 The Importance of Colors

After capturing an image, what can be done with it in order to instruct a robot to play football? An important step is the reliable extraction of visual features from the image, corresponding to relevant objects on the field.

First, the objects must be recognized. These are the two goals, the ball, the surrounding walls, the other robots and their markers. All those objects are recognized by their known colors. Eight different colors are used:

- the ball is red,
- the playing field is green,
- one goal is blue,
- the other goal is yellow,
- the surrounding walls are white (including some letters and symbols in black),
- the robots are predominantly black,
- one team color is magenta,
- the other team color is cyan.

These eight colors correspond precisely to the eight vertices of the RGB cube. The RGB cube is a color space representation where any color is represented by its different amounts of Red, Green and Blue, weighted from 0 to 1, in each axis (R, G and B) [21].

Color segmentation is obviously an important problem for RoboCup-soccer playing robots. The image processing system must not only correctly discriminate the eight significant colors, but also avoid the identification of objects external to the game as relevant ones. This is a critical issue since many people walks around the playing field wearing colourful T-shirt, and sometimes other red balls are left nearby the field.

3.2 Image Formats

Different cameras provide images in different formats. The most usual ones are RGB and YUV.

RGB is mainly used to display the image on computer monitors, and outputs a matrix of pixels, where each pixel color is represented by 3 bytes. Each byte contains a number between 0 and 255 representing the amount of Red, Green and Blue present in that pixel. Since a computer screen is a transmissive device, this means that the maximum value for the three colors (255, 255, 255) represents White, while the minimum value for the three colors (0, 0, 0) results in Black.

The other format, YUV, is more a video format rather than an image format. It was first developed to transmit television signal. Video has different color spaces due to its need to carry the signal in the fastest possible way. The luminance and color are separated. Television uses the color space model of one luminance and two chroma values because the chrominance values are easily compressed (using a lossy scheme). It is important to point out that the tone of an image carries most of the important image information (color is secondary). The lossy color compression does not compromise significantly the quality of the image since all the tone information is there. The U and V values represent the chrominance in the Red axis and in the blue axis. The sum of these two vectors indicates the real colour from a gamut.

Due to its video characteristics, YUV is the most suitable color space for color segmentation. Its main advantages can be described as:

- the signal is separated (to analyse shape, we don't need color, but just luminance);
- it is very much light independent;
- it is fast, since no hardware conversion is required;
- lookup tables are 2-dimensional, and thus they are easy to access and require less storage space than "true" 3-D color spaces;
- allows flexible conversions to RGB for display;
- many cameras output their image in the YUV format.

The main disadvantages are:

- it needs to be converted to RGB to be displayed on a computer screen;
- it is a format most suitable for video rather than for still images.

3.3 Sensor Readings from an Image System

Most sensorial information required in RoboCup-Soccer can be extracted by a vision system. Our approach was to define image windows where certain attributes are expected to be found. An example can be seen in Fig. 7, for a parabolic mirror. Notice that the image windows must be changed according to the particular mirror profile used and/or mirror assembly on the robot.

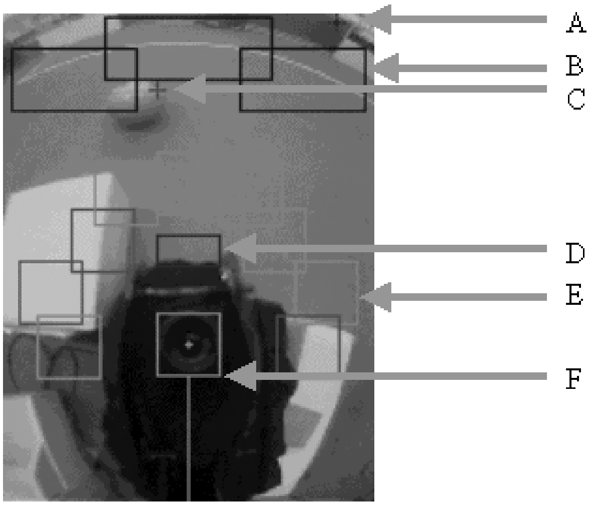


Fig. 7. Captured image, with superimposed defined windows for virtual sensors.

This system starts by applying a filter to every image pixel. All colors are segmented onto the eight possible and acceptable options (i.e., red, green, blue, yellow, magenta, cyan, black and white, or the eight vertices of the RGB cube).

The main virtual sensors used in this system are designated as Obstacle Avoidance, Goal Detection, Ball Detection, Eminent Kick and Catadioptric System Calibration. We shall now describe each of them in detail.

- Obstacle Avoidance The nine squares around the camera are used as virtual bumpers (window E in Fig. 7). The amount of black and white inside each square is calculated and, should it be over a certain predefined value, that square goes red, meaning that there is an obstacle in the window, forcing the motion controller to move the robot away from the obstacle. In the image shown, bumpers number 2, 3, and 9 are flagged as obstacles.
- Goal Detection By finding the maximum value of blue (window A in Fig. 7) in the image and applying a threshold, the blue goal can be found. A similar

technique is used for the yellow goal. In order to avoid noise from outside the field, which could be confused with the actual blue (yellow) goal, this maximum must be inside the top three rectangles on the image (window B in Fig. 7).

- Ball Detection The maximum value of red (window C in Fig. 7) represents the ball. The red color is the easiest to track and the one with least interference, since the ball has a very unique and bright color. Due to its motion, the ball can be seen anywhere on the image, and so can window C be located.
- Eminent Kick The robot should not activate the kicking device when the ball is not ready to be kicked, to save energy and avoid hurting its opponents. Therefore, the kicking device is activated only when the ball (red cross represented by letter C in Fig. 7) is inside the red rectangle (letter D on the same image). This also means that the robot will kick the ball only when the ball is touching the robot.
- Catadioptric System Callibration Should, for some reason, the mirror and/or the camera composing the catadioptric vision system be moved from its position, the robot will not find the relevant objects in the correct image windows. Therefore, for easy callibration of the catadioptric system, the camera lens must be placed inside the square given by letter F in Fig. 7.

Many other virtual sensors can be created. However, the number of sensors is critical for system performance and therefore their number must be limited, otherwise, the number of frames processed per second will substantially decrease. Code optimisation is therefore an extremely important topic here. The image processing functions are those using most CPU time. The Minho team has written them in assembly language, together with careful code optimisation.

4 Omni-Directional Vision-Based Self-Localization

The navigation system is one of the most important sub-system of a mobile robot. In many applications, especially those concerning well-structured indoors environments, one important feature of the navigation system concerns the ability of the robot to self-localize, i.e., to autonomously determine its position and orientation (posture). Once a robot knows its posture, it is capable of following a pre-planned virtual path or of smoothly stabilizing its posture. If the robot is part of a cooperative multi-robot team, it can also exchange the posture information with its teammates so that appropriate relational and organizational behaviors are established. In robotic soccer, these are crucial issues. If a robot knows its posture, it can move towards a desired posture (e.g., facing the goal with the ball in between). It can also know its teammate postures and prepare a pass, or evaluate the game state from the team locations [22].

In this section we describe a self-localization algorithm based on the isometric part of the multi-part mirror of the catadioptric vision system described in Section 2. The algorithm determines the posture of a middle-size league robot, with respect to a given coordinate system, from the observation of natural landmarks of the soccer field, such as the field lines and goals, as well as its correlation, in the Hough transform space, with a geometric field model. Even though the intersection between the field and the walls is also currently used, the wall replacement by the corresponding field lines would not change the algorithm. The algorithm is a particular implementation of a general method applicable to other well-structured environments, and was first introduced in [23].

4.1 Method Description

Even though the self-localization algorithm was designed motivated by its application to robotic soccer, it can be described in general terms and applied to other well-structured environments, with the assumption that the robot moves on flat surfaces and straight lines can be identified and used as descriptive features of those environments. An important requirement is that the algorithm should be robust to image noise. Given an image acquired from the isometric part of the catadioptric system, the basic steps of the algorithm are:

(1) Build a set T of *transition pixels*, corresponding to image pixel representatives of environment straight lines (e.g., intersection between corridor walls and ground, obtained by an edge detector).

(2) For all transition pixels $p^t \in T$, compute the Hough Transform using the normal representation of a line[21]

$$\rho = x_i^t \cdot \cos(\phi) + y_i^t \cdot \sin(\phi) \, , \tag{4}$$

where (x_i^t, y_i^t) are the image coordinates of p^t and ρ, ϕ the line parameters.

(3) Pick the q straight lines $(\rho_1, \phi_1), \ldots, (\rho_q, \phi_q)$ corresponding to the top q accumulator cells resulting from the Hough transform described in the previous step.

(4) For all pairs $\{(\rho_j, \phi_j), (\rho_k, \phi_k), \; j, k = 1, \ldots, q, \; j \neq k\}$ made out of the q straight lines in the previous step, compute

$$\Delta\phi = |\phi_j - \phi_k| \tag{5}$$

$$\Delta\rho = |\rho_j - \rho_k|. \tag{6}$$

Note that a small $\Delta\phi$ denotes almost parallel straight lines, while $\Delta\rho$ is the distance between 2 parallel lines.

(5) Classify, in the $[0, 100]$ range, the $\Delta\phi$s and $\Delta\rho$s determined in the previous step, for its relevance (function Rel(.)) using *a priori* knowledge of the

geometric characteristics of the environment (e.g., in a building corridor of width d, only $\Delta\phi \simeq 0$, $\Delta\phi \simeq 180$ and $\Delta\rho \simeq d$ should get high grades). For each pair of straight lines, assign a grade in the $[0, 200]$ range to the pair, by adding up $\mathrm{Rel}(\Delta\phi)$ and $\mathrm{Rel}(\Delta\rho)$.

(6) Pick up the most *relevant* pair of straight lines (i.e., the pair of largest $\mathrm{Rel}(\Delta\phi) + \mathrm{Rel}(\Delta\rho)$ in the previous step), and use it to extract some relevant feature regarding environment localization (e.g., the orientation θ of the robot w.r.t. the corridor walls, represented by the most relevant pair of parallel straight lines, in the example above).

(7) Use the relevant feature from the previous step to proceed. For instance, assuming θ in the corridor example is such a feature, it is used to select columns from the accumulator cells matrix referred in Step 3. The idea is to correlate a number of actual straight lines, found in the image, sharing the same descriptive parameter (e.g., the angle ϕ corresponding to θ) with the expected straight lines obtained from an environment model (e.g., the building layout). To attain this, up to n ρ values from the accumulator matrix column corresponding to ϕ are picked up, corresponding to up to n straight lines found in the image. To handle uncertainty in ϕ, an even better solution is to pick up not only one column but a few columns surrounding the accumulator matrix column corresponding to ϕ, using the top n ρ values from those columns. Concatenate all these Hough space points in an array and call it $\hat{\rho}_\phi$.

(8) Create an array ρ_ϕ similar to $\hat{\rho}_\phi$, but obtained from a geometric model of the environment. Actually, ρ_ϕ measures distances of environment straight lines to the origin of the world reference frame. Correlate ρ_ϕ and $\hat{\rho}_\phi$ by shifting one array over the other, and incrementing a counter for each matching $(\rho_\phi, \hat{\rho}_\phi)$ pair. The maximum of the correlation corresponds to the best match between up to n straight lines in the image and the n known environment straight lines. From this result and similar results obtained for other straight lines non-parallel to them (determined by the same procedure for different θs), the image coordinates of environment feature points, whose location in the world reference frame is known, are determined and used to determine the robot position w.r.t. that frame, by a suitable transformation from image to world coordinates.

4.2 Application to Robotic Soccer

The self-localization of a middle-size league soccer robot, using the method described in the previous section, takes advantage of the soccer field geometry and of the different colors used for the field (green), the surrounding walls and the field lines (white). The field is a 9×4.5 m flat rectangle that can be almost fully observed by the robot catadioptric system from most field locations.

The self-localization algorithm was implemented based on the isometric part of the catadioptric system mirror.

4.3 Geometric Field Model

The bird's eye view of the soccer field, shown schematically in Fig. 9-a), shows 6 horizontal and 7 vertical straight lines (considering interrupted lines as only one line). In this work, all horizontal lines and 5 of the vertical lines (excluding those corresponding to the back of the goals) were considered. Excluded lines were chosen because they are often occluded by the goalkeeper robots. All the distances between lines are known from RoboCup rules. Changes in the dimensions are parameterized in a table. The model reference frame is located at the bottom left of the model image.

4.4 Orientation Determination

Steps 1-6 of the algorithm described in Section 4.1 are followed to determine the initial robot orientation estimate (with a ±90° or 0°/180° uncertainty, to be solved later). The set T of transition pixels is obtained by determining the white-to-green and green-to-white image transitions over 36 circles centered with the robot, shown in Fig. 8. The number of circles was determined based on a tradeoff between accuracy and CPU time.

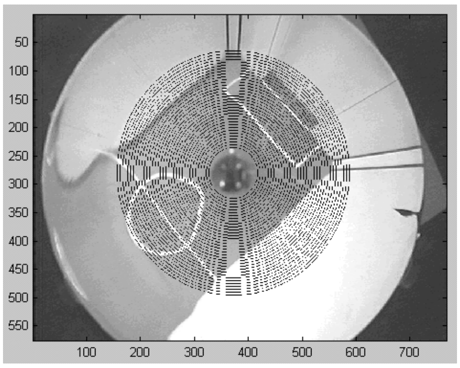
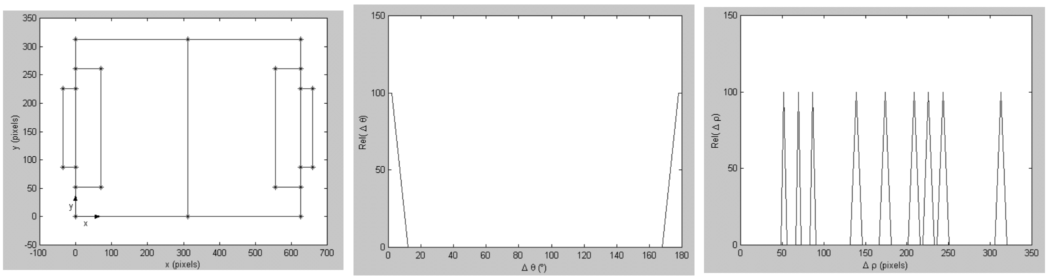


Fig. 8. Image obtained with a preliminary prototype of the isometric part of the catadioptric system mirror – notice the distortion on the outer part – showing the 36 circles used to determine transition pixels.

The Hough transform is then applied to the pixels in T – a variable number from image to image – depending on the number and length of observed lines. In Step 3, $q = 6$ is used, based on experimental analysis of the tradeoff between CPU time and accuracy. The *relevance* functions for $\Delta\phi$ and $\Delta\rho$, used in Steps 5-6, are plotted in Fig. 9-b) and c). The latter reflects *a priori* knowledge of the environment, by its use of the known distance between relevant field lines that can be observed by the catadioptric system in one image.



a) b) c)

Fig. 9. a) Soccer field model as seen in a bird's eye view image (coordinates in pixels). Also shown are the relevance functions for b) $\Delta\phi$ and c) $\Delta\rho$.

The accumulator cells of the Hough transform in Step 2 are obtained by incrementing ϕ from 0 to 180° in 0.5° steps, leading to an image line slope resolution of $\tan 0.5°$. ρ is incremented from 125 to 968 in steps of 1 pixel, corresponding to an actual field resolution of 6.95 mm [3]. The $\pm 90^o$ or 180^o ambiguity referred above results from the absence of information on which field lines lead to the most relevant pair. This information is obtained in Steps 7-8.

4.5 Position Determination

The final step in the self-localization process consists of determining the robot position coordinates in the soccer field. This is done together with the disambiguation of the relevant feature θ determined in Steps 1-6 of the self-localization method, by creating not only the ρ_ϕ and $\hat{\rho}_\phi$ arrays referred in Steps 7-8, but also their "orthogonal" arrays $\rho_{\phi+90}$ and $\hat{\rho}_{\phi+90}$. The correlation in Step 8 is made between all 4 possible pairs $(\rho_{\phi+90}, \hat{\rho}_{\phi+90})$, $(\rho_{\phi+90}, \hat{\rho}_\phi)$, $(\rho_\phi, \hat{\rho}_{\phi+90})$ and $(\rho_\phi, \hat{\rho}_\phi)$ with $n = 6$ (the maximum number of field lines that can be found in the image). The maximum of the 4 correlation maxima occurs for the array pair representing the best match between image and actual field lines. The array immediately identifies whether $\theta \pm 90°$ or $\theta = 0° \vee \theta = 180°$ is the robot orientation. A companion array pair exists for each best pair. The 2 pairs uniquely identify 2 (approximately) orthogonal field lines, by checking the array positions where the maximum occurred (vertical field lines are numbered $1, \ldots, 5$ from left to right and horizontal lines are numbered $1, \ldots, 6$ from top to bottom). The intersection of the two lines is a reference point, whose coordinates are known in the world reference frame, from the field model.

[3] The relation between ρ values and the actual field resolution is given by the scale factor k between field and image coordinates (see Section 2.2)

The explanation above is summarized in the following table (the best and companion pairs positions can be exchanged):

Best Pair	**Companion Pair**	θ
$(\rho_\phi, \hat{\rho}_\phi)$	$(\rho_{\phi+90}, \hat{\rho}_{\phi+90})$	$\theta = \phi \pm 90°$
$(\rho_\phi, \hat{\rho}_{\phi+90})$	$(\rho_{\phi+90}, \hat{\rho}_\phi)$	$\theta = \phi \vee \phi + 180°$

The robot position is computed from a rotation of θ (one of the possible values is used, with no special criterion), followed by a translation that expresses the center of the image (i.e., the robot position in image coordinates) in the model reference frame, and another translation plus a scale factor f to express it in world coordinates. The world reference frame is located in the middle of the soccer field, with the x axis pointing towards the blue goal and the y axis is such that a 3-D coordinate frame would have z pointing upwards. The orientation θ is measured from x to a pre-defined straight line passing through the robot center. The scale factor f depends on the geometry of the catadioptric system and can be callibrated experimentally. This transformation can be expressed by the following equation, using homogeneous coordinates:

$$\begin{bmatrix} x_f^r \\ y_f^r \\ 1 \end{bmatrix} = \left(\begin{bmatrix} \cos\theta & \sin\theta & x_i^{ref} + x_m^{ref} \\ -\sin\theta & \cos\theta & y_i^{ref} + y_m^{ref} \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x_i^r \\ y_i^r \\ 1 \end{bmatrix} - \begin{bmatrix} 450 \\ 225 \\ 0 \end{bmatrix} \right) \cdot f \tag{7}$$

where the subscripts i, m, f stand for the image, field model and actual field reference frames, and the superscripts ref and r stand for the reference point and the robot, respectively.

A further validation and disambiguation of the robot posture is required, since, when only two parallel lines are used to determine the position, and due to field symmetry, the robot side of the field is unknown, as well as its orientation. To solve this problem, two tests are made. First, the algorithm checks whether the robot position is not outside the field. The second test consists of using the current estimated posture to seek the nearest goal in the image.

This is achieved by selecting m points located inside one of the goals (blue or yellow) in the actual field and applying to each of those points of coordinates (x_f^g, y_f^g) the inverse transform of (7).

Should the majority of the corresponding pixels in the image have the same color of the field pixels, $\theta = 0°$ and the estimated position is validated. Should they have the color of the opposing goal, $\theta = 180°$ and the symmetrical coordinates of the current position estimate must be used for the robot position.

When the majority of image pixels is green, the top maximum of the correlation process is removed and the whole process re-started using the second maximum, and if needed, the third one and so on until the actual posture is determined.

4.6 Experimental Results

The described self-localization algorithm has been implemented in C and used to self-localize a robot. The method was applied to a set of 90 images obtained by a catadioptric system mounted on a Super Scout II robot. The images were taken at different field spots, with several images taken at each spot, and were processed in about 0.5 second each, in a Pentium 233MHz with 64Mb of RAM, the Super Scout II on board computer. The results from the 90 experiments give an average accuracy μ of 3.2 mm for the x coordinate, -18 mm for the y coordinate and 0.22° for θ, with standard deviations of 100 mm, 92 mm and 1.8°, respectively.

In Fig. 10, the histogram of the accuracy, for the x and y coordinates, is shown, as well as an adjusted Gaussian function. The rectangle on the plot contains all the accuracies within one standard deviation from μ, i.e., 68,2% of the postures obtained have an accuracy of less than or equal to 10 cm in x and 9 cm in y.

The accuracy was determined as the difference between the estimated values and the ones measured on the field, using pre-defined spots whose location is well known (e.g., the corner of the goal area). The precision (i.e., the difference between the measured value and the measurements average value for the same location) results are similar, and visual inspection made the average values seem trustable.

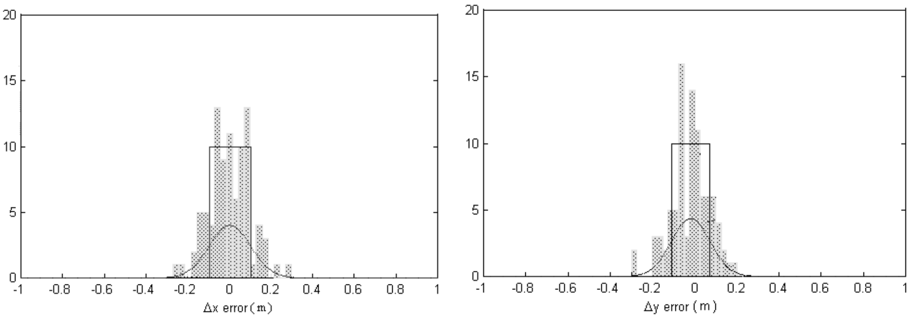


Fig. 10. Position error histogram.

Figure 11 shows an example of an image to be processed. The lines represented are the possible lines of the field. In this case, the $(\rho_\phi, \hat{\rho}_{\phi+90})$ pair achieved the top correlation value and position with an error of Δx=+1 cm, Δy= +1 cm and $\Delta\theta$= +1°. Note that, in this test, the robot is close to one of the field

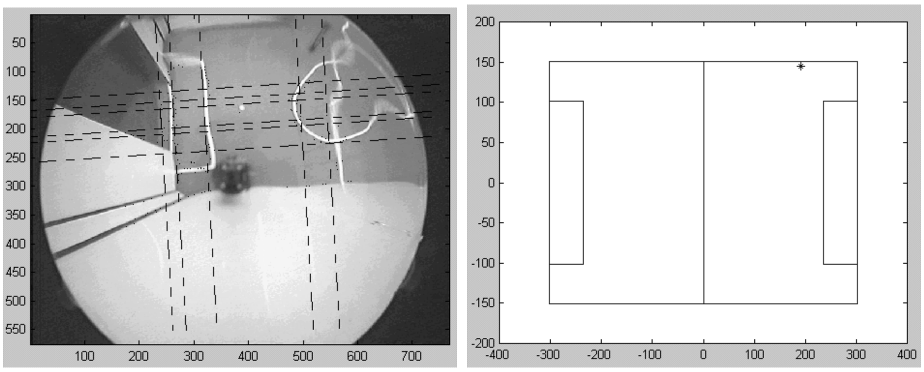


Fig. 11. Test image results.

walls, making harder the posture determination process, as, due to the limited image used, the other wall is not seen, and a relevant parallel line can not be found by the algorithm.

5 Conclusions

This paper has shown the potential of omni-directional catadioptric systems for comprehensive solutions for mobile robots moving within structured environments, ranging from the extraction of relevant image features to self-localization. Moreover, the paper introduces the design of a multi-part mirror which can be used, by controlling the distribution of image resolution onto the scene, to tackle all the requirements with the same device.

Further steps towards a more refined usage of the information provided by omni-directional vision systems, as described here, include:

- Endowing many teammates with such a system, so that they can share information on all teammate postures through communications, enabling the display of teamwork behaviors.
- Sharing also the information on other relevant objects (e.g., the ball, the opponent robots), so that a more accurate world model can be built and shared by all teammates.

References

[1] S. Nayar, Catadioptric vision, in: Proc. of the 7th Int. Symposium on Intelligent Robotic Systems, SIRS'99, 1999, pp. 1–11.

[2] R. A. Hicks, R. Bajcsy, Reflective surfaces as computational sensors, in: IEEE Workshop on Perception for Mobile Agents - Proc. of the 1999 IEEE Conference

on Computer Vision and Pattern Recognition (CVPR99), IEEE Computer Press, Piscataway, NJ, 1999.

[3] F. M. Marchese, D. G. Sorrenti, Omni-directional vision with a multi-part mirror, in: P. Stone, T. Balch, G. Kraetzschmar (Eds.), RoboCup 2000–Robot Soccer World Cup IV, Springer Verlag, Berlin, D, 2001.

[4] F. Ribeiro, C. Machado, S. Sampaio, B. Martins, Image processing applied to a robotic football team, in: G. Adorni, W. Hoeke (Eds.), Proc. of EuRoboCup 2000 Workshop – Amsterdam, 2000, pp. on a CD–ROM.

[5] Y. Yagi, S. Kawato, S. Tsuji, Real-time omni-directional image sensor (copis) for vision-guided navigation, IEEE Transactions on Robotics and Automation 10 (1) (1994) 11–22.

[6] E. Mouaddib, C. Pegard, Localization using omnidirectional vision, in: ICAR-95, IEEE Computer Press, Piscataway, NJ, 1995, pp. 133–138.

[7] J. B. Gregersen, A system for 360° vision in robot soccer, Ph.D. thesis, RMIT, Melbourne, AU (1998).

[8] S. Suzuki, T. Katoh, M. Asada, An application of vision-based learning for a real robot in robocup — learning of goal keeping behavior for a mobile robot with omni-directional vision and embedded servoing, in: M. Asada, H. Kitano (Eds.), RoboCup 98–Robot Soccer World Cup II, Springer-Verlag, Berlin, D, 1999, pp. 467–474.

[9] D. Nardi, G. Clemente, E. Pagello, Art azzurra robot team, in: M. Asada, H. Kitano (Eds.), RoboCup 98: Robot Soccer World Cup II, Springer Verlag, Berlin, D, 1999, pp. 467–474.

[10] A. Bonarini, P. Aliverti, M. Lucioni, An omni-directional sensor for fast tracking for mobile robots, IEEE Transactions on Intrumentation and Measurements 49 (3) (2000) 509–512.

[11] A. Bonarini, The body, the mind or the eye, first?, in: M. Veloso, E. Pagello, H. Kitano (Eds.), RoboCup 99–Robot Soccer World Cup III, Springer Verlag, Berlin, D, 2000, pp. 210–219.

[12] J. Borenstein, H. Everett, L.Feng, Where am i? – sensors and methods for mobile robot positionig, Tech. rep., University of Michigan (1996).

[13] S. Enderle, M. Ritter, D. Fox, S. Sablatnog, G. Kraetzschmar, G. Palm, Vision-based localization in robocup environments, in: P. Stone, T. Balch, G. Kraetzschmar (Eds.), RoboCup 2000–Robot Soccer World Cup IV, Springer Verlag, Berlin, D, 2001.

[14] D. Fox, W. Burgard, S. Thrun, Markov localization for mobile robots in dynamic environments, Journal of Artificial Intelligence Research (11) (1999) 391–427.

[15] P. Stone, T. Balch, G. Kraetzschmar (Eds.), RoboCup 2000–Robot Soccer World Cup IV, Lecture Notes in Computer Science, Springer Verlag, Berlin, D, 2001.

[16] R. Hanek, T. Schmitt, M. Klupsch, S. Buck, From multiple images to a consistent view, in: P. Stone, T. Balch, G. Kraetzschmar (Eds.), RoboCup 2000–Robot Soccer World Cup IV, Springer Verlag, Berlin, D, 2001.

[17] L. Iocchi, D. Nardi, Self-localization in the robocup environment., in: Proc. of 16th IJCAI 99, The third International Workshop on RoboCup, 1999, pp. 116–120.

[18] M. Plagge, A. Zell, Vision-based goalkeeper localization, in: G. Adorni, W. Hoeke (Eds.), Proc. of EuRoboCup 2000 Workshop – Amsterdam, 2000, pp. on a CD–ROM.

[19] L. Delahoche, C. Pégard, B. Marhic, P. Vasseur, A navigation system based on an omni-directional vision sensor., in: Proc. IEEE Int. Conf. on Intelligent Robots and Systems, 1997, pp. 718–724.

[20] R. A. Hicks, R. Bajcsy, Catadioptric sensors that approximate wide-angle perspective projections, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR00), IEEE Computer Press, Piscataway, NJ, 2000.

[21] R. Gonzalez, R. Woods, Digital Image Processing, Addison-Wesley, 1992.

[22] R. Ventura, P. Aparcio, P. Lima, C. Pinto-Ferreira, Isocrob - intelligent society of robots., in: RoboCup-99 Team Descriptions, Middle Size Robots League, 1999, pp. 150–159.

[23] C. Marques, P. Lima, A localization method for a soccer robot using a vision-based omni-directional sensor, in: P. Stone, T. Balch, G. Kraetzschmar (Eds.), RoboCup 2000–Robot Soccer World Cup IV, Springer Verlag, Berlin, D, 2001.