Vision-Based Self-Localization for Soccer Robots

Carlos F. Marques[*]

Instituto de Sistemas e Robótica
Instituto Superior Técnico
Av. Rovisco Pais, 1
1049-001 Lisboa
PORTUGAL
cmarques@isr.ist.utl.pt

Pedro U. Lima

Instituto de Sistemas e Robótica
Instituto Superior Técnico
Av. Rovisco Pais, 1
1049-001 Lisboa
PORTUGAL
pal@isr.ist.utl.pt

Abstract

In this paper, a method for robot self-localization based on a catadioptric omni-directional sensor is introduced. The method uses natural geometric landmarks of the environment. It is assumed that the robot moves on flat surfaces and straight lines can be identified in the sorrounding environment image acquired by the catadioptric system. This omni-directional vision system is based on a camera plus a convex mirror designed to obtain (by hardware) the ground plane bird's eye view. Results from the application to a real robot moving on RoboCup soccer field and concerning the method's accuracy are presented.

1 Introduction and Motivation

The navigation system is perhaps the most important sub-system of a mobile robot. In many applications, especially those concerning indoors well-structured 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 to stabilize its posture smoothly[8]. 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[9]. 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.

An increasing number of teams participating in RoboCup's middle-size league is approaching the self-localization problem [5]. 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. 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. Should the walls be removed, the method becomes not applicable.

In this paper, an omni-directional catadioptric (vision + mirror) system is used to determine the robot posture, with respect to (w.r.t.) a given coordinate system, from the observation of natural environment landmarks such as straight lines resulting from the intersection between the walls and the ground, as well as from *a priori* knowledge of the environment geometry. RoboCup's Agilo team proposes a vision-based approach to the self-localization problem too. A single frontal camera is used 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 [6] also use a single frontal camera and match the lines with a field model using the Hough Transform. Their approach considers lines detected locally, rather than a global field view, and uses odometry to remove ambiguities. Several teams use a vision-based omni-directional hardware system similar to the one described here, but only for ball, marking and oppos-

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

ing robots tracking. The robots of the Tuebingen team [5] use omni-directional vision for self-localization, but only the distance to the walls is used.

The paper is organized as follows: in Section 2, the proposed self-localization method is described in general terms, and its application to robotic soccer is detailed in Section 3. Results concerning accuracy of the postures obtained in this case study are presented in Section 4, together with a study of its dependence on robot field locations. Finally, some conclusions and a description of envisaged future work are drawn in Section 5.

2 The Self-Localization Method

2.1 Omni-Directional Catadioptric System

Most omni-directional catadioptric systems are based on one of two types of reflecting surfaces: spherical [3] or conic[1, 2]. Those systems require a perspective unwarping made by software, a time-consuming process that may prevent real-time robot guidance, in applications where fast motion is required. The solution used here, although developed independently, is similar to the one described in [4], whose catadioptric system preserves the geometry of a plane orthogonal to its symmetry axis, in particular providing a bird's eye view of the ground plane.

2.2 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 catadioptric system, the basic steps of the algorithm are:

1. Build a set $\mathcal{T}$ 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 $p^t \in \mathcal{T}$, compute the Hough Transform using the normal representation of a line[7]

$$\rho = x_i^t \cdot \cos(\phi) + y_i^t \cdot \sin(\phi), \qquad (1)$$

   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

$$\Delta\phi = |\phi_j - \phi_k| \qquad (2)$$
$$\Delta\rho = |\rho_j - \rho_k|. \qquad (3)$$

   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.

Figure 1: Catadioptric system assembled on the robot.

3 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

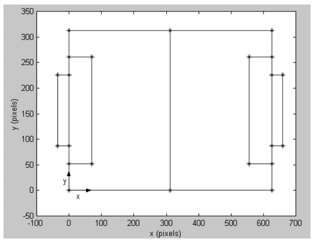

Figure 2: Soccer field model (coordinates in pixels).

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. To elliminate occlusion due to the use of mirror supports, the catadioptric system consists of an acrylic cylinder with a small color CCD camera inside and the mirror on the top. The camera is assembled in a 5 degree of freedom (3 of translation and 2 of rotation) support allowing the user to correctly position the camera with respect to the mirror. Acquired images are processed using the HSV color model[7].

Since all four robots in the team require the catadioptric system, the tradeoff between cost and final mirror machining quality had to be seriously considered. While it is true that image distortion due to poor mirror machining quality reflects in posture accuracy, the method here described was developed to be robust to considerable image distortion and to image noise due to irregular and poor illumination. As such, a non-expensive ($\simeq$ 150 Euros) mirror was used. Image distortion is severe on the external mirror zone, hence processing is only made within a circle with a 220 pixel radius, centered with the camera (see Fig. 3). Figure 1 shows the catadioptric system assembled on a Nomadic SuperScout II.

3.1 A Priori Knowledge

The bird's eye view of the soccer field, shown schematically in Fig. 2, 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 this field model.

3.2 Orientation Determination

Steps 1-6 of the algorithm described in Section 2 are followed to determine the initial robot orientation estimate (with a $\pm 90°$ or $0°/180°$ uncertainty, to be solved later). The set $\mathcal{T}$ 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. 3. The number of circles was determined based on a tradeoff between accuracy and CPU time.

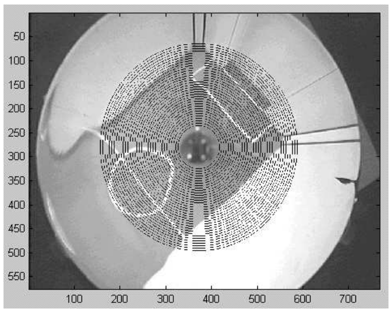

Figure 3: Image obtained by the catadioptric system with the 36 circles used to determine transition pixels.

The Hough transform is then applied to the pixels in $\mathcal{T}$ – 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 Figure 4. 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.

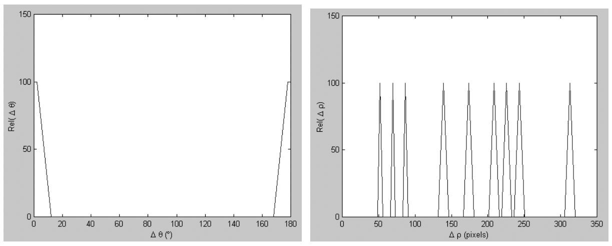

Figure 4: Relevance functions for $\Delta\phi$ and $\Delta\rho$.

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. 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.

3.3 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.

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:

$$\begin{bmatrix} x_f^r \\ y_f^r \\ 1 \end{bmatrix} = \begin{bmatrix} \cos\theta & \sin\theta & x_i^{\mathrm{ref}} + x_m^{\mathrm{ref}} \\ -\sin\theta & \cos\theta & y_i^{\mathrm{ref}} + y_m^{\mathrm{ref}} \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x_i^r \\ y_i^r \\ 1 \end{bmatrix} - \\ - \begin{bmatrix} 450 \\ 225 \\ 0 \end{bmatrix} \cdot f \tag{4}$$

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 de 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 (4):

$$\begin{bmatrix} x_i^g \\ y_i^g \\ 1 \end{bmatrix} = \begin{bmatrix} \cos\theta & \sin\theta & x_i^{\mathrm{ref}} + x_m^{\mathrm{ref}} \\ -\sin\theta & \cos\theta & y_i^{\mathrm{ref}} + y_m^{\mathrm{ref}} \\ 0 & 0 & 1 \end{bmatrix}^{-1} \cdot \\ \cdot \left(\begin{bmatrix} x_f^g \\ y_f^g \\ 1 \end{bmatrix} + \begin{bmatrix} 450 \\ 225 \\ 0 \end{bmatrix} \right) \cdot f \tag{5}$$

where the superscript g stands for goal.

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 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. Images were processed in less than 1 second each, in a Pentium 233MHz with 64Mb of RAM, the Super Scout II on board computer. Table 1 shows the results of the 90 experiments. The first column gives the average accuracy μ in x, y and θ, the second the variance of the accuracy σ^2 and the third column the accuracy for one standard deviation. In Fig. 5, the histogram of the accuracy, in x and y, is shown as well as an adjusted Gaussian function. The represented rectangle contains all the accuracies within one standard deviation from μ, i.e., 68,2% of the postures obtained have an accuracy of, e.g., 10 cm in X.

	μ	σ^2	$\mu - \sigma$
x	+3.2 (mm)	0.0099 (m^2)	10 (cm)
y	-18.0 (mm)	0.0084 (m^2)	9.18 (cm)
θ	0.22 °	3.14 $°^2$	1.77 °

Table 1: Posture accuracy statistics (mean and standard deviation).

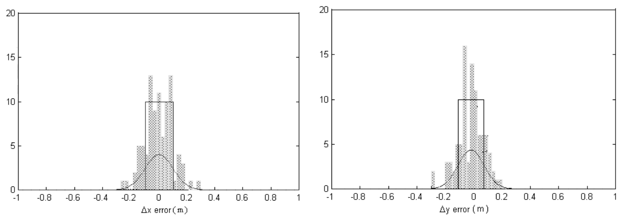

Figure 5: Distribution of the error.

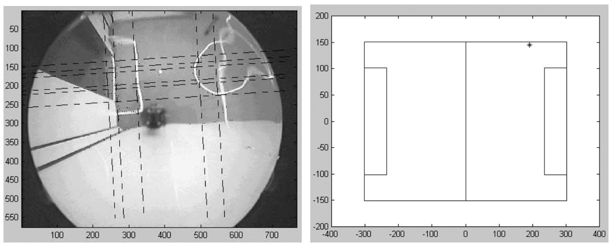

Figure 6: Test image results.

Figure 6 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

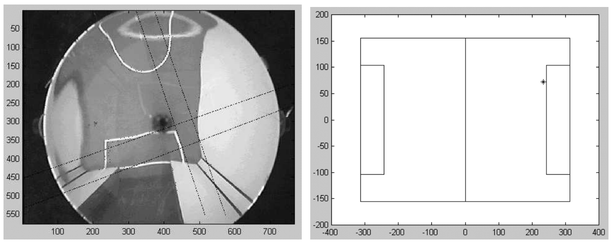

Figure 7: Bad test image results.

Δx=+1 cm, Δy= +1 cm and $\Delta\theta$= +1°. Note that, in this test, the robot is close to one of the field 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.

One example of a bad image, is shown in Fig. 7. In this case, the position was computed with an error of Δx=+10 cm, Δy= +1 cm and $\Delta\theta$= +21°. Even though the results shown in Fig. 7 are considerably worse, they are acceptable for the due purposes (except the orientation), considering the large image distortion. The method robustness meets the problem specifications.

5 Conclusions and Future Work

A vision-based algorithm for robot self-localization was tested on images taken from a low-cost catadiotric system mounted on a Super Scout II soccer robot. The algorithm was designed for well structured environments, where *a priori* knowledge about the word model is available, and straight lines can be used to describe environment features.

In the robotic soccer application, promising results were obtained concerning posture accuracy and method robustness to image noise and distortion. The team robots projection on the field is a circle of roughly 40cm, and typical position errors ranged from 0 to 10cm. One way to reduce the position error range is to use sensor fusion methodologies. The Tuebingen team fuses three different self-localization methods for the localization of their goalkeeper, using three sensors: an omni-directional camera, a Laser Range Finder and a compass [5]. The method can be used only in regions near the field goals. However, we rely only on an omni-directional vision system for self-localization which can be used everywhere in the field. Future work will include the method usage to periodically reset odometry and keep the whole team knowledgeable of the teammate postures. The main goal of our current work is to use the integrated odometry + vision-based system as feedback for guidance control of the ISocRob team soccer robots. The guidance controllers will use control laws for smooth posture stabilization and path following.

Acknowledgements

The authors would like to thank Luis Custodio, José Santos-Victor and Rodrigo Ventura for the fruitful discussions about the subject of this paper.

References

[1] Y. Yagi, M. Yachida. Real-time Generation of Environmental Map and Obstacle Avoidence using Omnidirectional Image Sensor with Conic Mirror. Trans. IEEE Robotics and Automation, Vol.10 No.1 , pp.11-22 2/ 28 ,1991

[2] L. Delahoche, C. Pégard,B. Marhic, P. Vasseur, A Navigation System Based on an Omni-directional Vision Sensor, proc. IEEE Int. Conf. on Intelligent Robots and Systems , pp. 718-724,1997

[3] B. Marhic, E. M. Mouaddib, C. Pegard. A localisation Method with an Omni-directional Vision Sensor Using Projective Invariant, in proc. of IEEE Int. Conf. on Intelligent Robots and Systems , pp. 1078-1083,1998

[4] R. A. Hicks, R. Bajcsy. Reflective Surfaces as Computational Sensors. CVPR9́9, Workshop on Perception for Mobile Agents,1999.

[5] ROBOCUP-99, Small and Middle Leagues, Team Descriptions, Editors: Silvia Coradeschi, Tucker Balch, Gerhard Kraetzschmar, Peter Stone, Sweden 27 July -6 August 1999

[6] L. Iocchi, D. Nardi. Self-Localization in the RoboCup Environment. Sixteenth IJCAI 99, The third International Workshop on RoboCup, pp 116-120,1999.

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

[8] C. Canudas de Wit, B. Siciliano, G. Bastin (Eds), Theory of Robot Control, CCE Series, Kluwer, 1996

[9] P. Lima, R. Ventura, P. Aparício, L. Custódio, A Functional Architecture for a Team of Fully Autonomous Cooperative Robots, accepted for publication in RoboCup-99: Robot Soccer World Cup III, Springer-Verlag, Berlin, 2000