Object detection and Localization using an Omni directional vision system in RoboCup environment

Object detection and Localization using an Omni directional vision system in RoboCup environment

Mansour Jamzad, Alireza Hadj Khodabakshi and Vahab.S.Mirrokni*
Department of Computer Engineering
Sharif University of Technology, Tehran, Iran
emails: jamzad@sharif.edu,alireza@cs.sfu.ca, mirrokni@lcs.mit.edu

ABSTRACT

In this paper, we describe a design and construction method for an omni directional vision system and how to use it on autonomous robots for object detection, localization and also collision avoidance in middle size league of RoboCup. This vision system uses two mirrors, a flat and a hyperbolic one. The flat mirror is used for detecting very close objects around the robot body, moreover the hyperbolic one is used as a global viewing device to construct a world model for the soccer field. This world model contains information about the position and orientation of the robot itself and the position of other objects in a fixed coordinate system. In addition, a fast object detection method is introduced. It reduces the entire search space of an image into a limited number of *jump points*. The objects are detected by examining the color of pixels on these, jump points and a few pixels on their neighborhood. We introduce a fast and robust localization method, using the angle of several fixed landmarks on the field with respect to the robot. Our localization method is based on line detection that uses clustering and Hough transform method. The omni directional viewing system is combined with a front view that uses a plain CCD camera. This combination provided A total vision system solution that was tested in RoboCup 2001 competitions in Seattle and satisfactory results in near real time speed was obtained.

Key words

Omnidirectional vision, Localization, Object detection, Vision system, Middle-size RoboCup

1 Brief description on RoboCup

Before starting our discussion on robot localization, we would like to give a brief description on RoboCup for readers who are not familiar with with RoboCup environment in which the present work is implemented.

In RoboCup, a team of mobile agents plays soccer against another such team in a predefined environment. Among several leagues in RoboCup, middle-size league in the one in which agents are real mobile robots of approximate dimensions of 50x50x60cm. These robots are fully autonomous agents. Their movements are not manually controlled but are carried out by their own sensors and decision-making devices.

Figure 1. A picture of middle size RoboCup field

The field dimensions of middle-size league are about 5×10 meters. There is a white color wall of height 50cm all around the field. The field is covered with green carpet. Robots play soccer with a size 4 FIFA red ball. Robots are mostly black except an identifying color markers of light blue or purple, on top of each robot. The goals are 2 meters wide and $90Cm$ high. One goal is yellow and the other one is blue. Figure 1 shows a picture of a middle size league field taken in Seattle in RoboCup 2001.

The vision system of robots detects objects according to their colors. A robot soccer game is played between two teams of robots. Each team has three players and one goalie. A game is played in two, ten-minutes half times. These robots usually are equipped with one or more CCD cameras for their vision system, and may also have other sensing devices such as laser range finders and infrared. In addition, they are equipped with computer system, hardware control boards, and wireless communication devices that allow them to communicate among each other and with a server outside the field. For detail information about

*At present Mr.Hadj Khodabakshi is a PhD student at School of Computer Science, Simon Frazer University in Vancouver, BC, Canada and Mr.Mirrokni is a PhD student at Laboratory of Computer Science, MIT, Cambridge, MA, USA. This paper is part of a join research work when they were students and members of Middle size RoboCup team at Sharif University of Technology.

RoboCup one may refer to www.robocup.org and [1, 2].

2 Introduction

The vision system described in this paper has been installed on Sharif CE middle size robotic team since 2001. Sharif CE has participated in all RoboCup competitions since 1999 and has achieved remarkable results [3, 4].

Although we can take advantages of having different sensory devices on a robot, but in order to design simple and efficient mobile robots whose hardware handling is easier, it is worth concentrating only on vision sensors.

To solve the problem of localization, collision avoidance and object detection in a RoboCup field, we have designed and constructed an omni directional viewing system that was combined with a front view vision system. As a result, we could overcome the limitations of having only one front view camera on a robot.

Our vision system uses two fire-wire (IEEE 1394) digital CCD cameras. They are connected via a switching hub to a laptop, that is the main processor of robot. In order for a mobile robot to react quickly and accurately to all changes in its environment, it should be able to detect objects (obstacles) very rapidly and precisely.

In the following, we divide our work into three main sections, omni directional mirrors design, object detection and localization. We conclude with a few suggestions for future developments.

3 Omni directional mirror design

The shape of an omni directional mirror is one of the most important factors in determining overall viewing performance [8]. There are two main factors in this regard:

1. Covering a wide area around the robot (if possible all of the soccer field).

2. Providing an adequate resolution for objects in a relatively far distance. A good performance in this regard guarantees the ball (that is the smallest object in soccer field) detection in most cases, even if it is too far away from the robot.

The most common omni directional mirrors have spherical, parabolic, conical or hyperbola shapes. An extensive survey on each type is given in [10]. Considering the advantages and disadvantages of both spherical and conical mirrors for our application, we chose a hyperbolic mirror because of the following reasons:

1. The projection of far objects on mirror border are large enough such that even a ball located in a relatively far distance can be easily detected.

2. There is much less distortion in projection of object shapes on mirror surface compared to conical mirror. In addition, by selecting the proper parameters for the hyperbola, we can visualize all areas of the soccer field on this mirror.

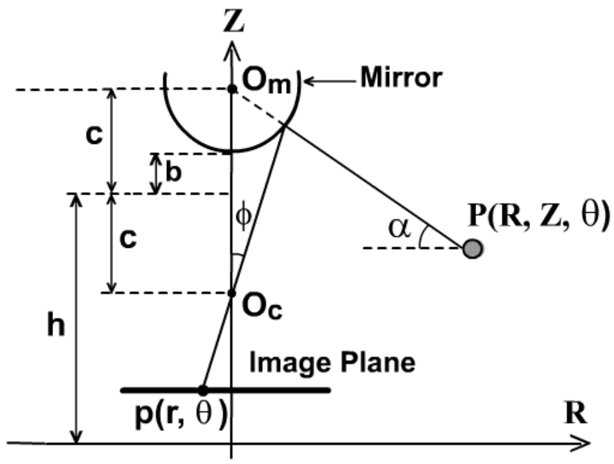

Figure 2. Geometrical model for hyperbolic mirror, a point P from the world coordinate is projected on point p on the image plane.

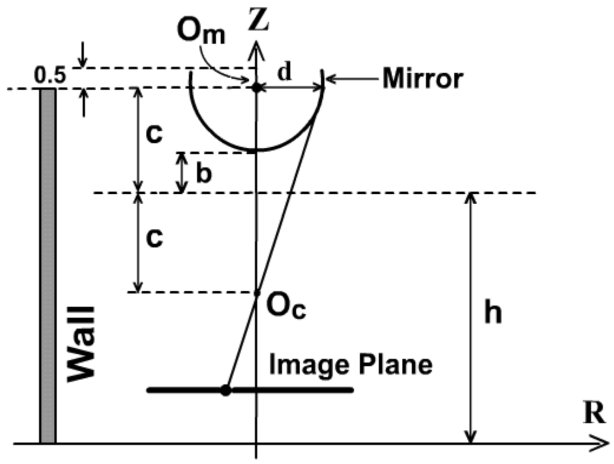

Figure 3. Hyperbolic mirror installation specification. R axis is laid on the field

3.1 A model for a hyperbolic mirror

Figure 2 shows a geometrical model for this mirror. Our purpose is to obtain a map between coordinate of pixels in the image plane and their corresponding points in the soccer field. This map is calculated with respect to a coordinate system fixed on the robot body.

To find the map of a point p with polar coordinate $p(r, \theta)$ on image plane and its corresponding point $P(R, Z, \theta)$ on the field, we use an important property in hyperbolic mirrors [14], that is:

As shown in figure 2, if a light beam initiated from a point P on the field hits the mirror surface in such a way that its extension passes through O_m (e.g., the first focus of hyperbola which is located inside the mirror body) then this light beam will reflect in a direction passing through the second focus of the hyperbola (point O_c). Therefore, we have to install a CCD camera in the second focus of hyperbola to capture the image reflected on hyperbolic mirror.

According to the illustrations of Figure 2 and 3, the equations of a hyperbolic mirror are given as follows:

$$\frac{R^2}{a^2} - \frac{(Z-h)^2}{b^2} = -1 \ \ where \ \ a = \sqrt{c^2 - b^2} \qquad (1)$$

We can calculate the map using the above property as follows:

$$Z = R\tan\alpha + c + h \qquad (2)$$

Since we know that all objects touch the field and their bottom has $Z = 0$, thus:

$$\tan\alpha = -\frac{c+h}{R} \qquad (3)$$

According to figure 2, we can write:

$$\tan\phi = \frac{b^2 + c^2}{b^2 - c^2\tan\alpha} + \frac{2bc}{c^2 - b^2} \times \frac{1}{\cos\alpha} \qquad (4)$$

To simplify the equations let

$$\tan\phi = \frac{f}{r} \qquad (5)$$

$$a_1 = \frac{b^2 + c^2}{b^2 - c^2} \quad a_2 = \frac{2bc}{c^2 - b^2} \quad a_3 = -(c+h) \qquad (6)$$

Therefore, by equations (4), (5) and (6) we have:

$$\frac{f}{r} = a_1\tan\alpha + \frac{a_2}{\cos\alpha} \qquad (7)$$

$$\frac{f^2}{r^2} - \frac{2a_1 f\tan\alpha}{r} + a_1\tan^2\alpha = a_2^2(1 + \tan^2\alpha) \qquad (8)$$

By equations (3) and (8) we have:

$$\frac{f^2}{r^2} - \frac{2a_1 f\frac{a_3}{R}}{r} + \frac{a_1^2 a_3^2}{R^2} = a_2^2(1 + \frac{a_3^2}{R^2}) \qquad (9)$$

$$(f^2 - r^2a_2^2)R^2 - (2a_1a_3fr)R + r^2a_3^2(a_1^2 - a_2^2) = 0 \quad (10)$$

According to the above equations, and considering $Z = 0$, if we have the coordinate of a point $p(r,\theta)$ in the image plane, by solving equation (10), we can find the coordinate of its corresponding point $P(R,\theta)$ in the field in robot's coordinate system.

It is important to notice that the angle of point P with respect to the robot's coordinate system origin is the same as the angle of point p on the image plane with respect to the image plane center.

This condition is satisfied if the hyperbolic mirror plane is set parallel to the field, the CCD camera axis is installed perpendicular to robot chassis (the field), and the center of hyperbolic mirror is set along the camera axis as shown in Figure 4.

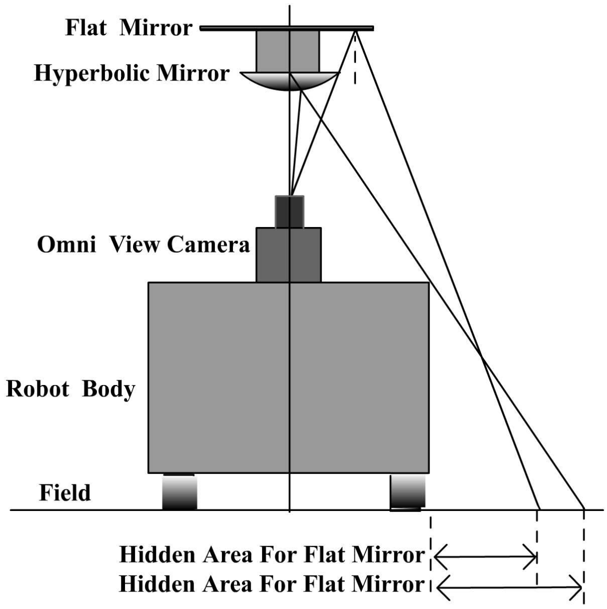

Figure 4. The camera axis is set along the center of hyperbolic mirror. The flat mirror installed on top of hyperbolic mirror is used as a collision detection sensor.

3.2 Camera calibration

As described in above, the map between points on image plane and their corresponding points on the field with respect to the coordinate system on robot is a function of camera and mirror parameters. These parameters are f as the camera focal distance, and a_1, a_2 and a_3, as given in equation (6), for mirror parameters.

Since these parameters can not be accurately obtained from camera and mirror specifications, therefore, in practical experiments these parameters are determined by using a few sample points from the map function (equation (10)). This process is called vision system calibration. Our method for such calibration is as follows:

We set four points $P_1(R_1,\theta_1)$, $P_2(R_2,\theta_2)$, $P_3(R_3,\theta_3)$ and $P_4(R_4,\theta_4)$ on the field, and then measure their corresponding reflected points $p_1(r_1,\theta_1)$, $p_2(r_2,\theta_2)$, $p_3(r_3,\theta_3)$ and $p_4(r_4,\theta_4)$ in the image plane. By putting these values in equation (10) we will have four unknowns and four equations, solving them we obtain the parameters f, a_1, a_2 and a_3.

To improve the accuracy of this map, we use the actual values obtained for above four parameters as an initial kernel in a local search to find a set of optimum values for them.

To perform this correction, we define a set of points f_i on the field such that these points are located on the circumference of a few concentric circles centered at the origin of the robot's coordinate system. For each point f_i, we locate its corresponding pixel p_i on the image plane, and pass the coordinate of p_i to the mapping function. The difference between the output of mapping function and that of its real

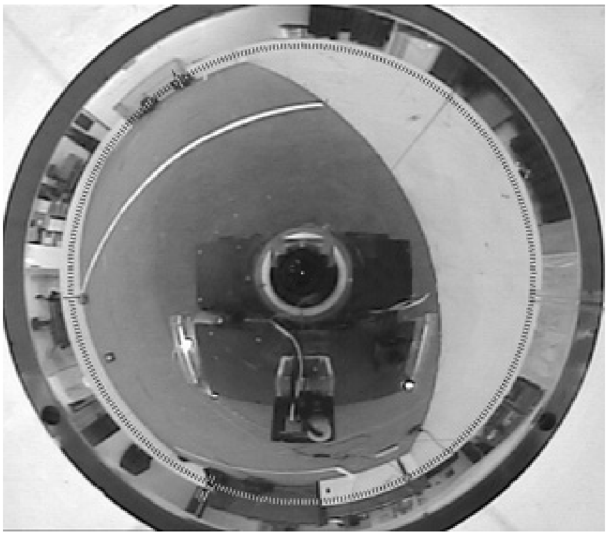

Figure 5. The edge of walls and goals are always projected on circumference of the surrounding circle.

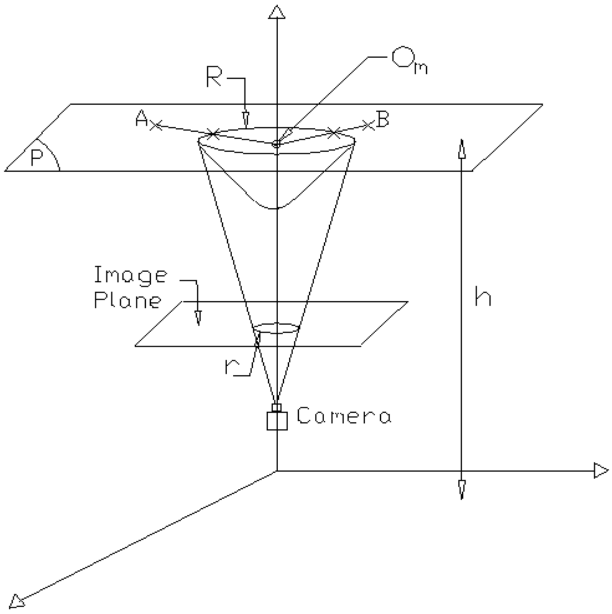

Figure 6. An explanatory three dimensional view for projection of objects in height h on a circle in image plane.

value is the error of mapping function for sample point pair (f_i, p_i). Then in the neighborhood of the kernel, we try to find new values for mapping function parameters, such that these parameters minimize the sum of squared errors for all sample points.

3.3 Calculating the parameters of hyperbolic mirror

If we install the hyperbolic mirror at such a height on the robot that the distance from field to the focus of mirror is the same as the height of wall (50 Cm), and we cut the mirror hyperbola with a plane parallel to field and at a height equal to 50.50 Cm (Figure 3 shows this configuration), then the picture projected on resulting mirror will have the following properties:

1. No matter where the robot is located on the field, the upper edges of walls are always projected on a circle with constant center and radius. The reason for this method of projection is explained in the following sections. Obviously, this circle always passes through both goals. Thus, to find the goals we can search only on the circumference of this circle. Figure 5 illustrates this property, where we can see the upper edges of walls are projected on the circumference of a large circle.

2. The upper part of two goals will always be projected on the image plane. This is because of the 0.5 Cm offset, as shown in figure 3. This is very helpful for detecting the goals especially in the situations when a few robots and the goalie are inside the goal area.

In these situations, the positions of the goals, that are used as landmarks for localization, are detected by processing only the upper part of the goals. This part will never be blocked by any robot because the maximum allowed height for the robots are smaller than the height of the goals (e.g., $90Cm$). Using this setting according to Figure 3, we can now compute the parameters of the mirror as follows:

$$\tan\alpha = \frac{d}{2c + 0.5} \tag{11}$$

Since point A satisfies the mirror equation (1), therefore:

$$\frac{d^2}{a^2} - \frac{(c + 0.5)^2}{b^2} = -1 \tag{12}$$

Thus from equations (1),(11) and (12) we conclude:

$$a = \sqrt{\frac{-d^2 + d\sqrt{d^2 + 4(c + 0.5)^2}}{2}} \tag{13}$$

As seen in Figure 3, O_m is the focus and d is radius of the circle created by cutting the upper part of hyperbola.

3.4 Projection of walls edge on image plane

There are white walls of height $50Cm$ all around the soccer field in middle size league of RoboCup. In the following we show why the projection of upper edge of the walls will be located on the circumference of a circle in image plane.

As it is seen in Figure 6, plane p is parallel to the field plane and passes through O_m, which is the focal point of hyperbolic mirror.

All points on plane p will be projected on the circumference of a circle with radius R on hyperbolic mirror. This circle is the intersection of plane p with the hyperbolic mirror. To explain the reason for this fact, let's assume two points A and B are located on plane p. To find the projection of these points on the mirror surface we shall extend the light beams initiated from them to point O_m. The intersection of these light beams with mirror surface are the projections of points A and B on the mirror surface. Since both O_m and points A and B are located on plane p therefore, such light beams intersect the mirror surface on the circumference of the circle that is the intersection of plane p and the hyperbolic mirror.

It is clear that in an omni directional viewing system, as described in section 3.1, the projection of a circle with radius R on hyperbolic mirror will be a circle with radius r in the image plane where R and r are determined according to the system setting.

Since the height of the walls is taken to be equal to the distance between point O_m and the field (e.g., h in Figure 6), therefore, the upper edge points of walls always will be located on plane p. Thus, the projection of upper edges of the walls are located on the circumference of a circle with radius r on image plane.

One can take advantage of this property of omni directional view to install the whole omni-view system including its mirror and camera on a vertical sliding mechanism. This mechanism that will be driven by a motor can slide up and down using for example a rack and pinion. By stopping the sliding mechanism in the desired height l from the field.

In this way if we want to search for objects located at height l we only need to examine the image data on the circumference of a circle with radius r on the image plane.

This device can have some industrial applications where we need to look for objects that we can guess they are located in a certain height from the ground (e.g., assume the robot moves on the ground). Moreover, the ability to precisely control the displacement of this sliding mechanism in omni directional vision system enables us to verify an initial guess and continue the search for finding the desired object.

4 Flat mirror as a collision sensor

One of the rules in RoboCup indicates that a robot should not collide with other robots. Some teams use infrared sensors to detect close objects. But, since we only use vision sensors, we solved this problem by adding a flat circle shape mirror on top of omni directional mirror, as it is seen in Figure 4. Depending on the height of robot main body, the height at which the mirror is installed and the position of camera, there is always a blind area around the body of robot that cannot be seen by the hyperbolic mirror.

To reduce this blind area, we installed a flat mirror of radius $11Cm$ at top of the hyperbolic mirror as shown in figure 4. Such flat mirror reduces the width of blind area. Therefore, to determine if some objects are located very close (e.g., about 8 Cm) to the robot body, we can check only the content of a ring shape area in image plane that corresponds to the flat mirror. The object detection

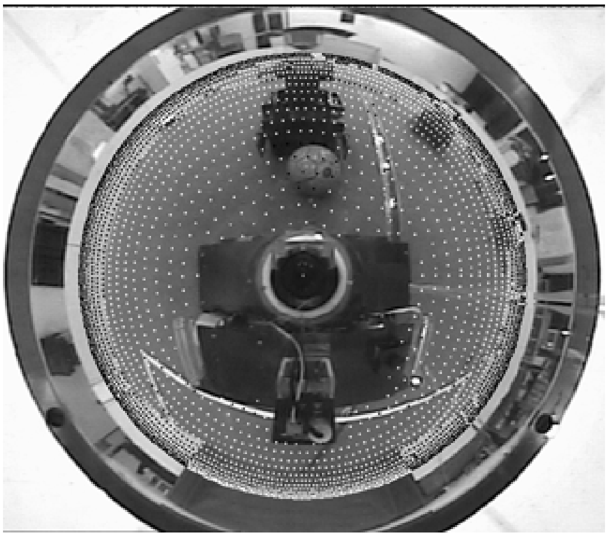

Figure 7. Jump points as shown by white dots are distributed on circumference of circles.

algorithm used for this ring shape area is similar to one used for that part of image corresponding to hyperbolic mirror.

In practice, this mechanism of object detection by flat mirror, worked well as a vision based collision detection sensor.

5 Object detection algorithm

In a highly dynamic environment like RoboCup, we shall use very fast and relatively accurate vision analysis routines that can respond in near real time speed (e.g., $\frac{1}{25}$ seconds). We have proposed a fast vision system based on checking a set of *jump points* in a perspective view of the robot front CCD camera [5] This paper has been given the "Best Engineering Challenge Award" in RoboCup 2001 Symposium in Seattle. We have extended that idea in omni directional view as well, where we define a set of jump points on the image plane as visualized by white dots in Figure 7. The distribution of these jump points is of high importance. These points are located on circumference of concentric circles. The center of these circles is the same as the center of image plane. The number of jump points on each circle reduces as the circle gets closer to the center. The distance between each two jump points and each two consecutive circles is determined in such a way that at least four jump points could be located on the smallest object, that is the ball, independent of distance of the object from robot.

To determine an object, we examine the color of image pixels at jump points. Our color detection routine that is fully described in [5] returns a color code for a pixel. This color code stands for one of the standard colors in RoboCup (e.g., red, green, white, black, blue, yellow, light blue and purple) and an unknown color code for all other colors.

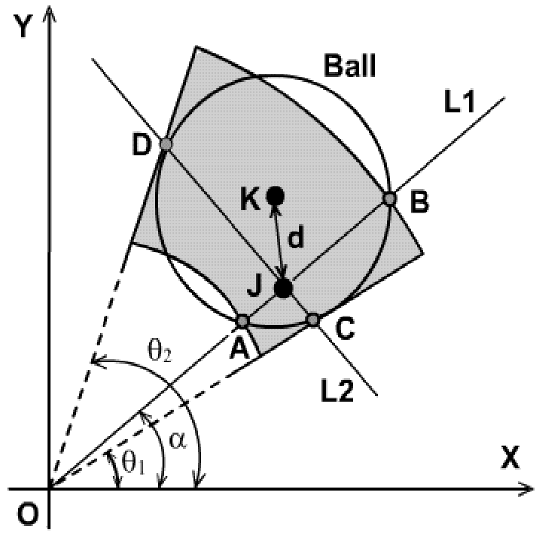

Figure 8. An initial rough estimate to surround the ball by a sector.

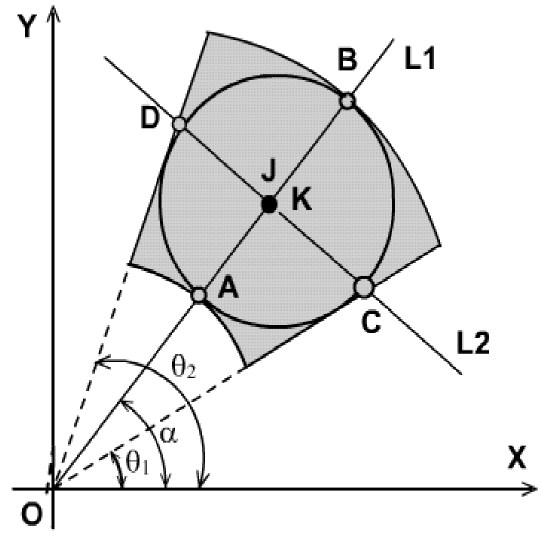

Figure 9. A final good estimate to surround the ball by a sector.

In the following, we explain how the ball is detected. If the color at a jump point is red, we search in a neighborhood of that jump point to find the ball. Although there are many algorithms such as region growing and boundary detection for this purpose [9], and they give a very accurate solution to find the object, but in a fast changing environment like RoboCup, we preferred to use *fast and almost correct algorithms* rather than slow and accurate methods for object detection. Our object detection algorithm works as follow:

As it is seen in figure 8, we start from a jump point J, and move on two perpendicular lines $L1$ and $L2$ that cross each other at point J. Line $L1$ passes through points O and J. From jump point J we start moving toward each end of lines $L1$ and $L2$, until hitting the border points A, B and C, D. A border point is a point on which there is a change of color, from red to a non red color (i.e.ball is red). These four points determine the minimum and maximum angle (θ_1, θ_2) and the radius of sectors intersecting (or in a best case surrounding) the ball, as it is shown in Figure 8 and 9. The area inside the sector passing through points $A, B, CandD$ is taken as an estimator for the ball. However, if K is the center of this sector, the distance d (i.e. the distance from jump point J to K) is a measure to check the accuracy of this estimate. If d is larger than a certain threshold, the estimate is not a good one. In this case we take point K as a new jump point and repeat the above procedure again, until d becomes less than the desired threshold. Figure 9 shows an accepted estimate for the ball.

At this stage, some features of the ball, such as its distance and angle with respect to the robot are calculated. As seen in Figure 9, OA and α are good estimates for ball distance and angle from the robot.

6 Localization

Localization is one of the most important and difficult tasks in mobile robots. In RoboCup, where a team of robots should have a cooperative behavior to achieve certain goals, the importance of localization becomes very clear. In short, robots will not able to perform team work in a multi agent system, unless they have a relatively accurate information about the position of robots of own team and also those of opponent team on the field.

For self localization in RoboCup environment, methods that use Laser Range Finders (LRF) and also the situation in other mobile environments are studied in [11, 12, 13].

However, since we only use vision sensors on the robots, in the following we introduce a localization method that is based on the omni directional viewing system that we have constructed.
In our method we find the coordinate, angle or equation of several fixed landmarks on the field with respect to robot coordinate system. These landmarks are the four intersection points between filed and the posts of goals, and also the intersection lines between walls and the field.

6.1 Localization using fixed points

There are two ways of localization using fixed points : Using the distance and angle of two fixed points in the robot coordinate system, or using the angle of three or more fixed points in the robot coordinate system.

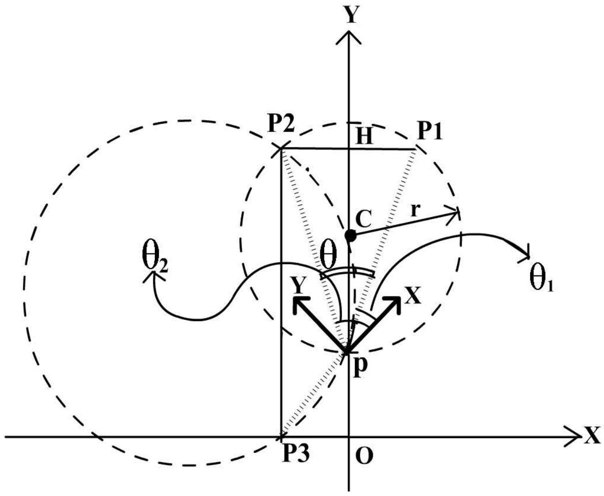

Figure 10. Localization using angle from fixed points

6.2 Localization using angles from fixed points

In the following we describe a localization method that uses the angle of four fixed landmarks with respect to robot coordinate system.

These landmarks are the intersection points of the posts of goals with the green field. Let's call these four land-marks P_1, P_2, P_3 and P_4. As it is seen in Figure 10, the locus of points P that have a fixed angle θ for $P_1\hat{P}P_2$ are located on a chord with center C and radius r. The center C is located on the perpendicular bisector of line segment P_1P_2 such that $CH = \frac{P_1P_2}{2*\tan\theta}$ and $r = \frac{P_1P_2}{2*\sin\theta}$, where $\theta = \theta_2 - \theta_1$. However, if we know the angle of robot with respect to three points such as P_1, P_2 and P_3, then the robot is located on the intersection of two chords. These two chords intersect each other on maximum of 2 points, such that one of these two points is either P_1, P_2 or P_3.

The orientation of a robot in the field can be computed using the coordinate of robot and its relative angle to one of these four landmarks. We use the fourth landmark as a tester to verify the validity of result. In the following, we describe the image processing method for finding the angles of these landmarks with respect to robot coordinate system.

There are two main interesting points that help us to find these angles efficiently and reliably in the image plane. The first point is that, the reflection of a line (e.g.,for example a vertical column of a goal) that is perpendicular to the field, is a straight line in the image plane that passes through the center of image plane. So each point on this straight line have the same angle in image plane. Also, as we mentioned in section 3.3 , independent of the robot position in the field, the upper edges of wall will always be projected on a circle with constant radius such that the center of this circle is located on the image center. Since this

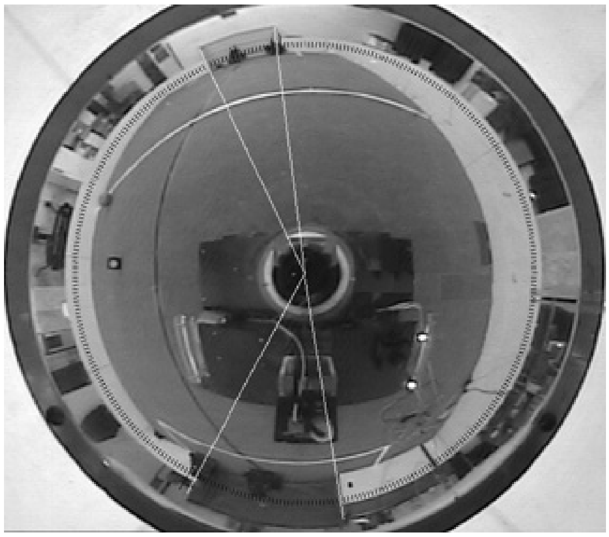

Figure 11. Angles of goal posts with respect to robot, when goal area is blocked by some robots.

circle passes through both goals, therefore, to find the angle from four landmarks (e.g., two goals vertical columns), we will search only on the circumference of this circle (it is called "surrounding circle" in the rest of this paper). In the following two steps, we describe how to find enough data to determine the angle of landmarks with respect to robot coordinate system. The search algorithm is as follows:

1. For each point P_i on the surrounding circle, determine if it is located on blue goal, yellow goal or the wall. To do this, the algorithm checks some points in the neighborhood of P_i, that are located along a radius of the surrounding circle that passes through P_i. In situations in which P_i is located on a moving object such as the ball or a robot, the algorithm should determine if this moving object is located near the wall, or it is in a goal area. In such cases, our algorithm checks the pixels on a neighborhood of P_i that is outside the surrounding circle. Therefore, even in cases that the goal area is blocked by robots, the algorithm will be able to find the landmarks correctly. At the end of this step, we can classify the points on the surrounding circle into three categories: points on blue goal, points on yellow goal and points on the wall.

2. In this step the algorithm finds the goals positions. For example, for blue goal it finds the largest consecutive number of points on the surrounding circle that are located on a blue goal.

Figures 11 shows how angles from image plane center to four landmarks are calculated in a case when the goal area is blocked by more than one robot. You can see two small black objects inside each goal. These objects are robots.

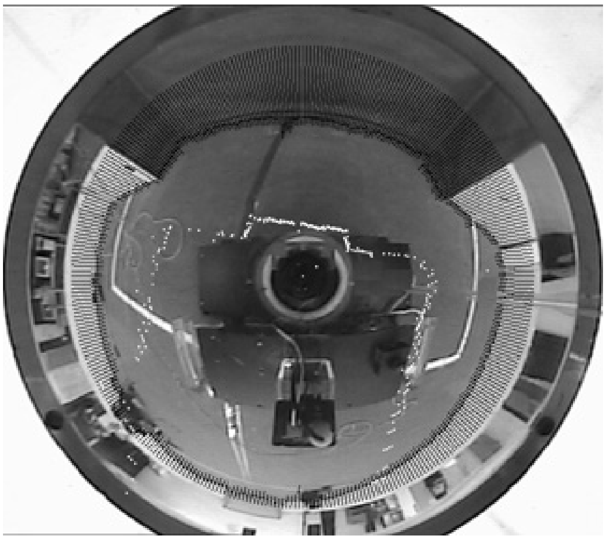

Figure 12. A visualization of mapping of field border points on robot coordinate system

6.3 Localization using fixed lines

Lines are some of the most convenient landmarks to be used for localization. Because of their easy equation they can be computed accurately by using a few points. Localization using lines is very straightforward and the algorithms that are based on line detection are very robust to noise. Since in some special cases, localization using the angle from the landmarks is impossible, we have developed an auxiliary method for localization using line detection.

We use two kinds of lines for localization. The border line between goals and the filed, and the borderline between walls and the field. This method has three main steps:

1. **Finding border points:** To find border points, we start the search from points on the circumference of surrounding circle and move on a radius line toward the center of circle. A border point is the one detected on its corresponding color transition.

2. **Mapping the points**: At this step, the algorithm maps all border points on their corresponding points in robot coordinate system. But, due to the inaccuracy existing in mapping of points that are too far from robot body, such far points are neglected after they are found.

3. **Using Hough transform [9]**: We assign some weights to the selected mapped points according to their importance and then pass them to a modified version of Hough transform algorithm for line detection [6]. Since these border points belong to several border lines that cross each other with an angle of 90 degrees, the equation of each such line is extracted from the result of Hough transform. As a result, knowing the equations of two crossing fixed lines, the position of robot can simply be calculated.

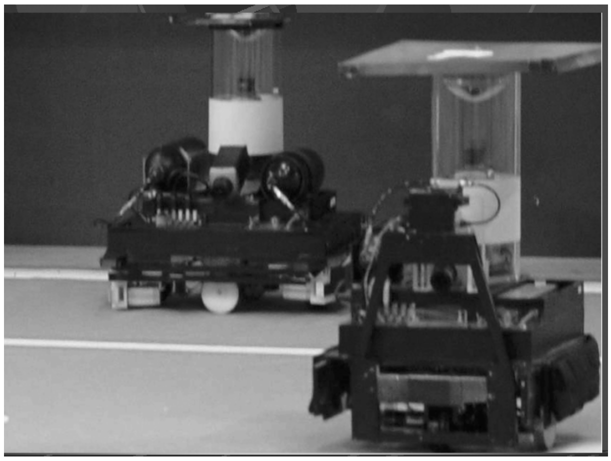

Figure 13. Two robots with their omni vision systems installed inside transparent vertical tubes. The front robot is a player and the one at back is a goal keeper.

Figure 12 shows a visualization of mapping the field border points (border of walls and the goals with the field) on robot coordinate system. The mapped points are shown as white dots that are scattered along straight lines near left, right and front side of robot body.

7 Conclusion

Our experiments in RoboCup have shown that if we use only a front view vision system on a robot, there will be several cases when the robot sight of view is blocked by other robots. In such situations, localization methods that are based on only one front view CCD camera will fail, simply because their sight of view is blocked. Moreover, even if a front view CCD camera can see the necessary scene, but depending on the position that the robot is actually located on the field, and the content of image acquired by the front view, the robot might fail to find its location.

We have implemented one such approach [15], that is based on finding the best match between one image acquired by front view CCD camera with a set of images in a database (e.g. these images are taken in a training stage from different positions and angles). The best match is obtained by comparison of a feature vector of the present image with those in the database. These feature vectors are constructed according the coefficient of wavelet transforms. However, since this method did not show a good performance in some special cases mentioned in above, therefore we decided to use omni directional vision system for localization and moreover to use many other advantages of it.

One of the main advantages of having an accurate localization method is that we will have the necessary information to make robots to perform a cooperative behavior in the soccer field.

In this paper, we have shown how to use omni directional vision for localization, object detection and collision detection. Although it is possible to use infra-red and LRF to do these jobs, but we believe it is a good idea to reduce the number of sensors and hardware complexity of robot by using only vision sensors. However, one of the problems with vision sensors is that, we must develop very fast algorithms that can be run in real time speed on the main processor used in the robot. Thinking of the complexity of the real time environment in a robot soccer field, the accuracy and real time speed of robot vision system remains a challenge.

Our omni directional viewing algorithms were tested on a processor with an AMD K6, 500MHz processor and 64MB of RAM. The speed of the localization algorithm itself was about 48 frames per second and that of object detection was about 41 frames per second. As a result, construction of the world model was performed in about 22 frames per second. This vision system was tested in Seattle in RoboCup 2001 completions and satisfactory results were obtained.

Figure 13 shows a picture of two robots designed and constructed by Sharif CE robotic team using the omni vision system described in this paper. As it can be seen in the figure, the omni vision system is installed inside a transparent tube that has a diameter of $15Cm$. The robot seen on the right is a player and the one on the left is a goalkeeper.

8 Acknowledgment

We would like to thank a lot to all Sharif CE team members from previous years specially A.Fourghnasiraie, R.Ghorbani, M.Kazemi, B.Sadjad, H.Chitsaz and M.T.Hajiaghayi who made many contributions to the development of our team. We would like to show our sincere gratitude to all high ranking managers of Sharif University of Technology that provided the necessary facilities to develop this research. In addition, we thank a lot all of our financial sponsors for their contributions.

References

[1] **Stone,P.,** et al, (Eds), RoboCup-2000: Robot Soccer world Cup IV, Springer, Berlin, (2001).

[2] **Birk,A.,** et al, (Eds), RoboCup-2001: Robot Soccer world Cup V, Springer, Berlin, (2002).

[3] The RoboCup Federation. RoboCup-99 Stockholm, Award Winners. *http://www.robocup.org/games/99stockholm/3132.html*.

[4] The RoboCup Federation. RoboCup-2000 Melbourne, Award Winners. *http://www.robocup.org/games/2000melbourne/3144.html*.

[5] **Jamzad, M.,** et al. A fast vision system for middle size robots in Robocop. In *RoboCup-2001: Robot Soccer World Cup V*, Birk,A., et al, (Eds), Lecture Notes in Computer Science 2377, pp. 71-80, Springer, Berlin, 2002.

[6] **M.T.Hajiaghayi**, M. Jamzad, *Simple, fast and robust self-localization in environments similar to the RoboCup environment*, CARs and FOF'2002, 17th International conference of CAD/CAM, robotics and factories, Porto, Portogal 3-5 July 2002.

[7] **M. Jamzad**, et al, *Middle sized Robots: ARVAND*, RoboCup-99: Robot Soccer world Cup III, Springer (2000), pp 74-84 .

[8] **A.Bonarini**, et al, *An omnidirectional vision sensor for fast tracking for mobile robots*, Instrumentation and Measurement, IEEE Trans.on, Vol:49 issue:3, June 2000, pp 509-512.

[9] **R.C. Gonzalez**, and R.E. Woods, *Digital Image Processing*, Addison-Wesley, 1993.

[10] **T.Svoboda**, *Central panoramic cameras design, geometry, egomotion*, Ph.D thesis, Center for Machine Perception, Department of Cybernetics, faculty of Electrical Engineering, Czech Technical University, Czech Republic, September 1999.

[11] **Gotmann, J.S., Weigel T.** and **Nebel, B.** Fast, accurate, and robust self-localization in the Robocop environment. In *RoboCup-99, Lecture Notes in Artificial Intelligence*, volume 1856, pages 304–317. Springer, Berlin, 2000.

[12] **Iocchi, L.** and **Nardi, D.** Self-localization in the Robocop environment. In *RoboCup-99, Lecture Notes in Artificial Intelligence*, volume 1856, pages 318–330. Springer, Berlin, 2000.

[13] **Olson, C.F.** Probabilistic self-localization for mobile robots. *IEEE Transactions on Robotics and Automation*, 16(1):55–66, February 2000.

[14] **Shoji Suzuki**, et al, *Behavior Learning for a Mobile Robot with Omnidirectional Vision Enhanced by an Active Zoom Mechanism*, Proc. of Intelligent Autonomous System 5(IAS-5), pp.242-249, 1998.

[15] **M.Jamzad**, M.Ebraahimi Moghadam, *Vision based localization for a mobile robot in RoboCup environment*, Proceedings of the International conference of image science, systems and technology, CISST,2001, Vol1, Las Vegas, Nevada, USA, June 25-28-2002, pp.114-119.