An omnidirectional vision system for localization and object detection in middle size RoboCup

M. Jamzad, A. Hadj Khodabakhshi, and V.S. Mirrokni

Computer Engineering Department
Sharif University of Technology, Tehran, Iran
`jamzad@sharif.edu`
`kbakhshi@ce.sharif.edu`
Laboratory of Computer Science
Massachusetts Institute of Technology, Cambridge, MA, USA
`mirrokni@lcs.mit.edu`

Abstract. In this paper, we describe an omnidirectional viewing system used for object detection, localization and also collision detection. This viewing system uses two mirrors, a hyperbolic and a flat one. The flat mirror is used for detecting very close objects around robot body, and 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, position of other robots and the ball in a fixed coordinate system. A fast object detection method is introduced. It detects objects by searching around a set of jump points that their colors match that of the desired object being searched for. The localization system uses four fixed landmarks on the fields. The angle of these landmarks with respect to robot coordinate system is calculated for localization. Also in some situations, it uses two intersecting lines that are the border lines between the wall and green field, and the one between the goals and green field. This viewing system is combined with a front view that uses a plain CCD camera. This combination gave a total vision system solution with satisfactory results and near real time speed.

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

1 Introduction

The vision system described in this paper has been installed on Sharif CE middle size team robots since 2001. Sharif CE has participated in all RoboCup competitions since 1999. Our long experience convinced us that, although it is good to have different sensor devices on a robot, but looking into future and considering the fact that, RoboCup is going to evolve in humanoid, on that, most possibly there will be only two CCD cameras working as stereovision, therefore, it is worth to mainly concentrate on vision sensors.

Taking the above strategy in mind, we decided to use only vision sensors. We used to have only one front view CCD camera on our robots before RoboCup

2001. This vision system and its novel ideas won the Engineering challenging award in Seattle, 2001 [1]. However, to solve the problem of localization, collision avoidance, detecting objects on all areas of the soccer field, we designed and constructed an omnidirectional viewing system and added that to the front view.

As a result, this combination worked much better and we could overcome the limitations of the front view and also get an excellent solution to the vision problem.

The omnidirectional system uses two firewire (IEEE 1394) digital CCD cameras. They are connected to a laptop, that is the main processor of the robot via a switching hub. More explanation of our robots including its mechanics, hardware control and software are given in [3], [2].

In the following, we divide our work into three main sections, omnidirectional mirrors design, object detection and localization. At the end we will conclude with a few suggestion for further development in future.

2 Omnidirectional mirror design

The shape of an omnidirectional mirror is one of the most important factor in determining overall viewing performance [4]. There are two main factors in this regard:

1. Covering a wide area around the robot (if possible all soccer field).
2. Having an adequate resolution for objects located at a far distance from the robot. A good performance in this regard guarantees the ball (that is the smallest object in RoboCup) detection in most cases.

The most common omnidirectional mirrors have spherical, parabolic, conical and hyperbola shapes. An extensive survey on each type is given in [6]. In brief, we shall say that, the problem with spherical mirror is that, they have a large degree of resolution in areas close to their center, but they lack this property around the border of mirror. Therefore, a large portion of image central area will be covered by the robot body, and objects located at a far distance are seen very small and can not be distinguished. This is a serious problem especially ,for ball detection.

We face an opposite problem with conical mirror. It gives a high degree of resolution for far objects that are projected on its edges. But, the projected shape of object on mirror surface shows lots of distortion. In addition, we have a blind area around robot body where they are not projected on this mirror. Although it is possible to overcome this problem by determining the right angle of the conic, but in this case we may lose some of the visible area which is not acceptable.

We concluded that we shall use a hyperbolic mirror for the following reasons:

1. The projection of far objects on mirror border are large enough such that, even the ball can easily be detected at far distances.

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.

2.1 A model for hyperbolic mirror

Figure 1 shows a geometrical model for this mirror. We should obtain a map between coordinate of pixels in the image taken by the CCD camera of this viewing system and their corresponding points in the soccer field. This map is calculated with respect to a coordinate system fixed on the robot body.

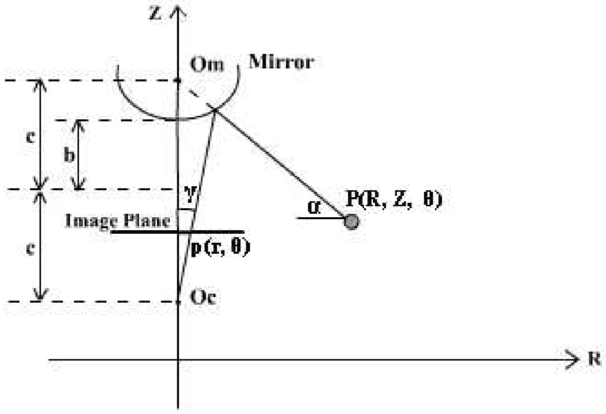

Fig. 1. Geometrical model for hyperbolic mirror

To find the map of a point $p(r, \theta)$ on the image plane and its corresponding point $P(R, Z, \theta)$ on the field, we use an important property in hyperbolic shape mirror [8], that is:

If a light beam initiated from a point P on the soccer field hits the mirror surface in such a way that its extension passes through the first focus of hyperbola which is located inside the mirror body, then it will reflect in a direction that will pass through the second focus of the hyperbola. Therefore, we install a CCD camera in the second focus of the hyperbola to capture the image reflected on the hyperbolic mirror. Figure 1 shows this property of hyperbolic mirror.

The equations of a hyperbolic mirror are as follows:

$$\frac{R^2}{a^2} - \frac{(Z-h)^2}{b^2} = -1 \quad and \quad c = \sqrt{a^2+b^2} \tag{1}$$

We can calculate the map using this property as follows:

$$Z = R \tan \alpha + c + h \tag{2}$$

Assuming all objects touch the field and their $Z = 0$, we will have:

$$\tan \alpha = -\frac{c+h}{R} \tag{3}$$

$$\tan \lambda = \frac{b^2+c^2}{b^2-c^2 \tan \alpha} + \frac{2bc}{c^2-b^2} * \frac{1}{\cos \alpha} \tag{4}$$

$$\tan \lambda = \frac{f}{r} \tag{5}$$

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

By equations (4), (5) and (6) we have:

$$\frac{f}{r} = a_1 \tan \alpha + \frac{a_2}{\cos \alpha} \tag{7}$$

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

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

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

$$(f^2 - r^2 a_2^2)R^2 - (2a_1 a_3 f r)R + r^2 a_3^2(a_1^2 - a_2^2) = 0 \tag{10}$$

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 robots' coordinate system.

It is important to notice that the angle of point P with respect to robots' coordinate system origin, is the same as the angle of point p on 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 also the center of mirror is set along the camera axis.

We have satisfied these three conditions in installation of the omnidirectional viewing system.

2.2 Camera calibration

As described in above, the map between points on image plan and their corresponding points on the field with respect to the coordinate system on the 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 the 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 system calibration. Our method for calibrating the system 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 the 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 which 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 the 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 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 of the output of mapping function and that of its real 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 sample points.

2.3 Calculating the parameters of hyperbolic mirror

If we install the hyperbolic mirror in such a height on the robot that the distance from the field to the focus of the mirror is the same as the height of the wall (50 Cm), and we cut the mirror hyperbola with a plane parallel to the field and at a height equal to 50.50 Cm (Figure 2 shows this configuration), then the picture projected on the resulted mirror will have the following properties:

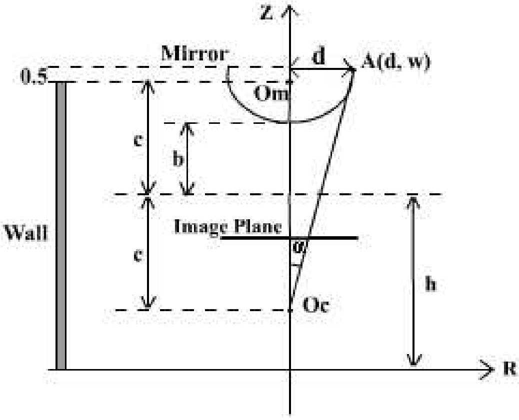

Fig. 2. Hyperbolic mirror installation specification

1. No matter where the robot is located on the field, the upper edges of the wall are always projected on a circle with constant center and radius. Obviously this circle always passes through both goals, so to find the goals we can search only on the circumference of this circle. Figure 3 illustrates this property.
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 2). This is very helpful for detecting the goals specially in the situations when a few robots and the goalie are in the goal area.

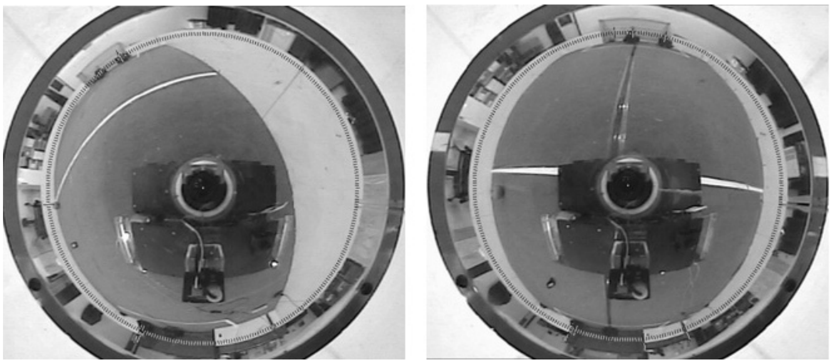

Fig. 3. The edge of walls and goals projected on circumference of surrounding circle.

In these situations, the posts of the goals (that are used as landmarks for localization) are detected by processing the upper part of the goals. This part will never be blocked by the robots because the maximum allowed height for the robots is smaller than the height of the goals. Using this setting we can now compute the parameters of the mirror.
According to figure 2 we can write the following equations:

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

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 2, O_m is the focus and d is the radius of the circle created by cutting the upper part of the hyperbola.

3 Flat mirror as a collision detection sensor

One of the rules in RoboCup indicates that a robot should not collide with other robots. Some teams use infra-red sensors to detect close objects. But, since we only use vision sensors, we solved this problem by adding a flat circle shape mirror on the omnidirectional mirror, as it is seen in figure 4. Depending on the height of the robot main body, the height at which the mirror is installed and the position of the camera, there is always a blind area around the body of robot that cannot be seen by the hyperbolic mirror. The flat mirror reduces the width of this blind area, as it is shown in figure 4. Therefore, to determine if some objects are located very close (about 8 Cm) to the robot body, we can check only the area of image plane that corresponds to flat mirror. In our robots, the radius of this flat mirror is 11Cm.
In practice, this mechanism worked well as a collision detection sensor.

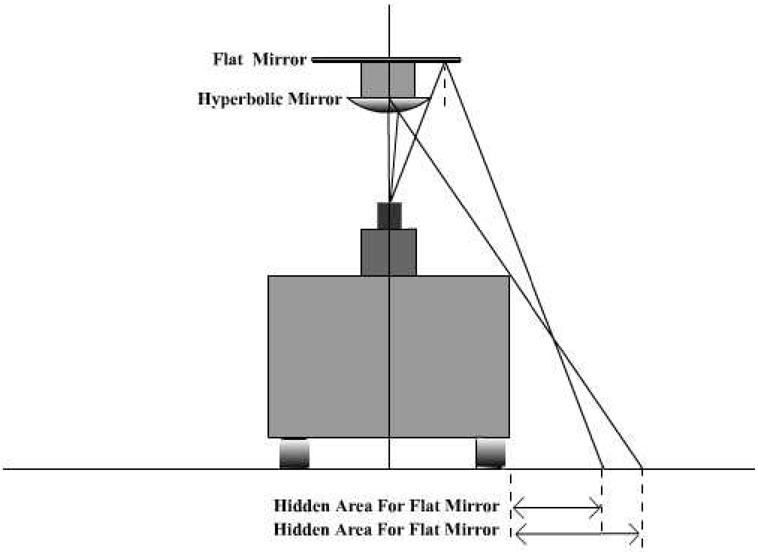

Fig. 4. Flat mirror as collision detection sensor

4 Object detection in omnidirectional viewing system

In a highly changing environment like RoboCup, we shall find very fast vision analysis routines that can respond in near real time rate. We have proposed a fast vision system based on checking a set of jump points in a perspective view of robot front CCD camera [1]. We have extended that idea in omnidirectional view as well, where we define a set of jump points on the image plane as it is seen in figure 5. 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 image plane center. 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 will be located on the smallest object, that is the ball, independent of its distance from robot.

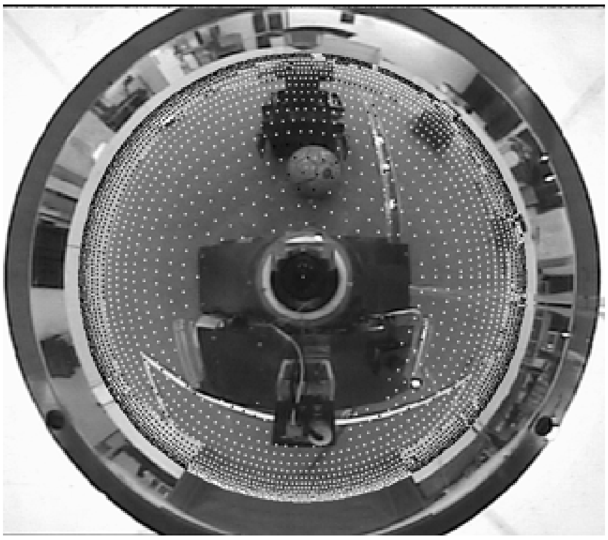

Fig. 5. Jump points distribution

To determine an object, we examine the color of image pixels at jump points. The color detection routine that is fully described in [1] returns a color code for a pixel. This color code stands for one of the standard colors in RoboCup (red, green, white, black, blue, yellow, light blue and purple) and an unknown color code for all other colors.
In the following, we explain how the ball is detected. If the color at a jump point is red, we search on the neighborhood of that jump point to find the ball. Although there are many algorithms such as region growing and boundary detection for this purpose [5], and they give very accurate solution to find the the object, but in a fast changing environment like RoboCup, we preferred to use *fast and almost correct algorithms* for object detection. Our object detection algorithm works as follow:

As it is seen in figure 6, we start from a jump point, and move on two perpendicular lines with equations $y = x \tan \alpha$ $(L1)$ and $y = \frac{-x}{\tan \alpha}$, $(L1)$. On

each line, we start moving from the jump point toward opposite directions, until hitting the border points A, B and C, D. A border point is a point on that there is a change of color, from red to a non red color. These four points determine the minimum and maximum angle (θ_1, θ_2) and the radius of sectors surrounding the ball, as it is shown in figure 6. The area inside these surrounding sectors is taken as an estimator for the ball. Some features of the ball, such as position, area, etc. are calculated from the above area.
However, if it happens that the first jump point is not located somewhere around the center of object, but is near its border, as it is seen in figure 6(left), the resulted area is not a good estimate for the ball. In this case we resume the search from a new search point (figure 6(left)). This new search point is selected on the center of the rectangle passing through points $A, B, CandD$. This procedure is repeated until the distance between the last two start search points is less than a given threshold. The procedure to detect all other objects is the same as that of the ball.

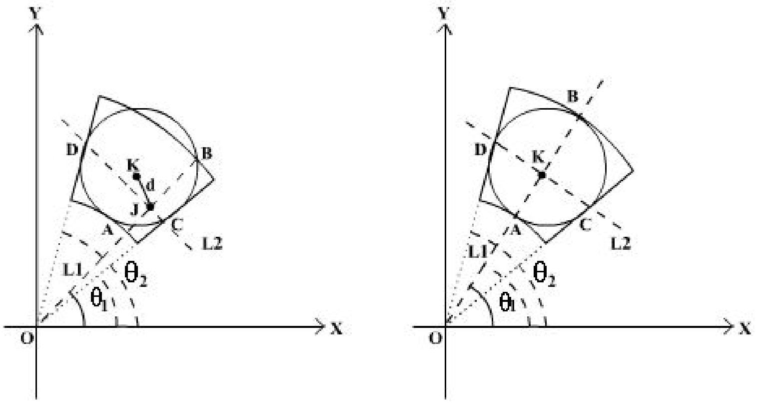

Fig. 6. A fast object detection method

5 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 are not able to perform team work in a multi agent system, unless they have a relatively accurate information about their own and each other locations on the field.
There are several accurate systems for localization which use Laser Range Finders (LRF) [7]. However, since we only use vision sensors on the robots, in the following we introduce a localization method based on omnidirectional

viewing system.
In our method we find the coordinate, angle or equation of several fixed landmarks on the field with respect to the robot coordinate system. These landmarks are, four intersection points between filed and the posts of goals , and also the intersection lines between walls and the field.

5.1 Localization using fixed points

There are two ways of localization using fixed points :

(a) Using the distance and angle of two fixed points in the robot coordinate system .

(b) Using the angle of three fixed points in the robot coordinate system.

Our method is based on the second category, but in the following section we describe both methods in detail.

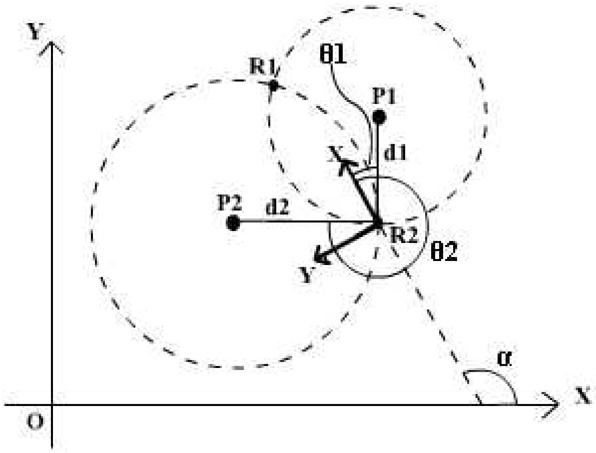

Fig. 7. Localization using angle and distance from two fixed points.

5.2 Localization using angle and distance of two fixed points

If a robot knows its distance and angle from two fixed points in the field, then it can compute its position. Note that by position we mean the coordinates and orientation of the robot in the field. As is shown in figure 7, the locus of points that have a known distance d_1 from a landmark P_1, is a circle with radius d_1 and center P_1. By calculating the distance of two such land marks from the robot, we can assume the robot to be on the intersection of the two circles, which are points R_1 and R_2. Also we can find the orientation of the robot in the field by knowing its location in the field and angle from one of these two landmarks. But as mentioned, there are two positions on the field that satisfy these conditions . The correct position can be determined by using the angle of robot from the other land mark.

Although this method seems to work well, but in practical situation, there are some limitations to it. That is, since it uses the information of only two points, it is very sensitive to noise. In addition, in this case we should use the mapping function between pixels on the image plane and their corresponding points in the field. The problem with this mapping function is that for situations in which the land mark is located far from the robot, a small error in detecting the land mark on the image plane, causes a large error in determining its corresponding point in the field. This fact is because of the compactness of the image around the border area of the mirror.

In practice, for example, two far points that have a distance of about $0.5m$ on the field, might have a distance of only a few pixels in the image plane.

5.3 Localization using angles from fixed points

To overcome the above problem we introduce a localization method that uses the angle of four fixed landmarks with respect to the robot coordinate system.

These landmarks are the intersection points 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 8, 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(Figure 8). 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 the robot and the relative angle to one of these four landmarks. We use the 4th landmark as a tester to verify the validity of the 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 perpendicular line on the field, is a straight line in the image plane that passes through its center. So each point on this straight line have the same angle in image plane. Also, as we mentioned in section 2.3 , independent of the robot position in the field, the upper edges of the wall will always be projected on a circle with constant radius and centered at image plane center. Since this circle passes through both goals ,to find the angle from the landmarks, we can start the search from this circle. In the following, this circle is referred to as surrounding circle. For finding these angles, our algorithm takes two main steps:

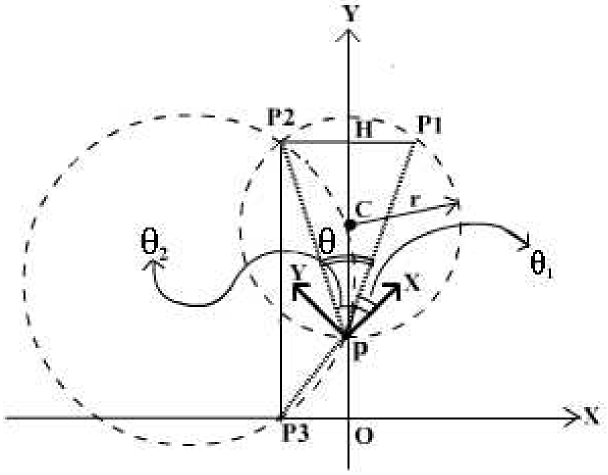

Fig. 8. Localization using angle from fixed points

(a) 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, which are located on the 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 in a goal area. In such cases, our algorithm checks the pixels on a neighborhood of P_i that is outside the surrounding circle. So even if the goal area is blocked by robots ,the algorithm will be able to find the landmarks correctly. As a result of this step, we can divide the points on the surrounding circle into three categories:
   i. Points on blue goal.
   ii. Points on yellow goal.
   iii. Points on the wall.

(b) In this step the algorithm finds the goal intervals. For example, for blue goal it finds the largest consecutive number of points on the surrounding circle that are located on a blue goal.

Figure 9 shows the application of this method in two different situations. As you can see in the left image the goal area is blocked by robots. As a result of these two steps, the blue and yellow goals intervals are calculated. Using these data, we can determine the angle of four landmarks with respect to robot coordinate system.

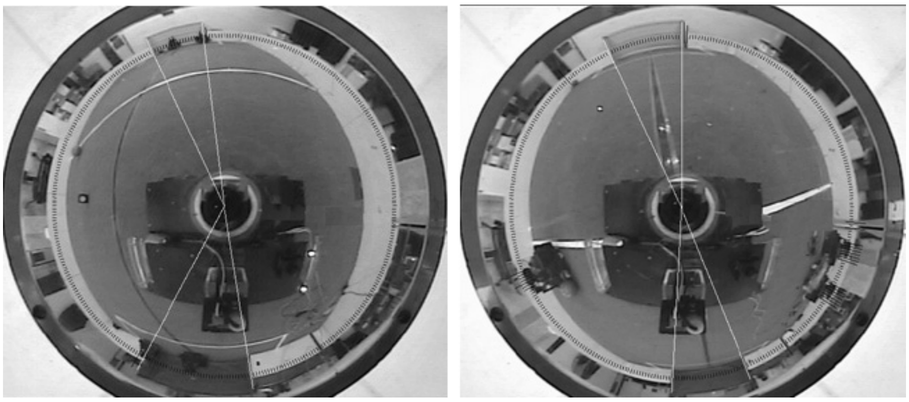

Fig. 9. Angles of goal posts with respect to robot.

5.4 Localization using fixed lines in the field

Lines are 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 situations, localization using the angle from the landmarks is impossible , we have developed an auxiliary method for localization using line detection.

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

(a) 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. A border point is the one which is on its corresponding color transition.
(b) Mapping the points: At this step, the algorithm maps all border points to their corresponding points in robot coordinate system. But, due to the inaccuracy of mapping of far points, as mentioned in section 5.2, border points found in far areas are rejected.
(c) Using Hough transform: All border points are passed to Hough transform algorithm for line detection [5]. Since these border points belong to several border lines, the equation of each of them is extracted from the result of Hough transform.

Figure 10 shows the application of this method in a real situation. In the above routine, if we determine the equations of two crossing fixed lines, then, the position of robot simply can be calculated.

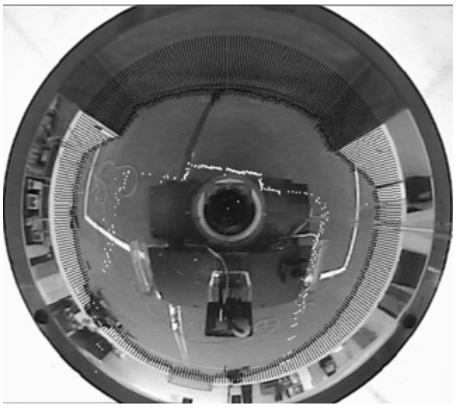

Fig. 10. Localization using fixed lines

6 Conclusion

Our experiments in RoboCup have shown that if we use only a front view on a robot, there will be lots of cases when the robot sight of view is blocked by other robots. In addition, it will be very difficult to perform a reliable localization method using only one CCD camera as a front view. One such method that is based on comparison of one image with a set of images in a database using wavelet transform is given in [9]. The usage of an omnidirectional viewing system will overcome both above mentioned problems. By combining these two viewing systems we can get a complete solution to the robots vision problem in RoboCup.

In this paper, we showed how to use omnidirectional vision for localization, object detection and collision detection. Although it is possible to use infrared and LRF to do this job, but, we believe it is a good idea to reduce the number of sensors and robots' hardware complexity by having only vision sensors. However, one of the problems with vision sensors is that, we must develop very fast algorithms, and even if we do so, the main processor of robot, due to its speed limitation, may not be able to give a real time performance. But, since each year the main processor speed is becoming almost double, it seems that, in future the problem of processor speed will be less serious.

The omnidirectional viewing algorithms were tested on a system with an AMD K6, 500MHz processor and 64MB of RAM. The speed of the localization algorithm itself, was about 48 frames per second, 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.
However, 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. A well organized team work, is one of the most important success factors for robots in RoboCup.

7 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 who made lots of contributions to the development of our team. In addition 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. We shall thank a lot all of our financial sponsors for their contributions.

References

1. M. Jamzad, et al, *A fast vision system for middle size RoboCup*, To appear in RoboCup-2001: Robot Soccer world Cup V, Springer (2002).
2. M. Jamzad, et al, *Sharif CE team description for RoboCup 2002*, To appear in RoboCup-2002: Robot Soccer world Cup VI, Springer (2003).
3. M. Jamzad, et al, *Middle sized Robots: ARVAND*, RoboCup-99: Robot Soccer world Cup III, Springer (2000), pp 74-84 .
4. 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.
5. R.C. Gonzalez, and R.E. Woods, *Digital Image Processing*, Addison-Wesley, 1993.
6. 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.
7. Jens-Steffen Gutmann, et al, *An experimental comparison of localization methods*, In *IROC'98*, Victoria, October 1998.
8. 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.
9. 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.