Designing an omnidirectional vision system for a goalkeeper robot

Emanuele Menegatti*, Francesco Nori, Enrico Pagello†*, Carlo Pellizzari, and Davide Spagnoli

Intelligent Autonomous Systems Laboratory
Department of Informatics and Electronics
University of Padua, Italy
†*also with* Institute LADSEB of CNR
Padua, Italy
*{emg,epv}@dei.unipd.it

Abstract. The aim of this paper is to provide a guideline for the design of an omnidirectional vision system for the Robocup domain. We report the design steps undertaken, with a detailed description of the design of an omnidirectional mirror with a custom profile. In the second part, we present the software written to exploit the properties of the designed mirror. The application for which the vision system is designed is the Middle-Size Robocup Championship. The task performed by the vision system is to localise the robot and to locate the ball and other robots in the field of play. Several practical tricks and suggestion are provided.

1 Introduction

In the Robocup competitions several teams use omnidirectional vision sensors and their number is increasing year after year. Asada used a goal-keeper fitted with an omnidirectional vision system with a learning capability [8]. To reduce the learning time, the omnidirectional sensor is fitted with an attention control provided by an active zoom mechanism that permits to select a restrict area of the omnidirectional image. Lima used an omnidirectional sensor for the self-localization of the robot in the field of play [6]. The omnidirectional mirror is designed to give a bird's eye view of the field of play. This permits to exploit the natural landmarks of the soccer field (goals and fields lines) for a reliable self-localization. Despite this second example, most of the teams use simple commercial mirrors, often used for other task than playing Robocup competitions. Yagi, one of the pioneers of omnidirectional vision with his COPIS system [10], proposed to look at the mirror of a catadioptric systems as a design variable [9]. The insight is that the robot task determines the design of the mirror. This idea was stressed by Marchese and Sorrenti in [5]. Their paper has been fundamental for our work as well as the work of Bonarini [2]. In [5] they presented the design of a mirror composed by three parts:

- the *isometric part*, that permits to determine the position of the objects in the Robocup field;

- the *markers mirror*, that is devoted to the identification of the robots' markers;
- the *proximity mirror*, that is used to perform high resolution measures in a area close to the body of the robot;

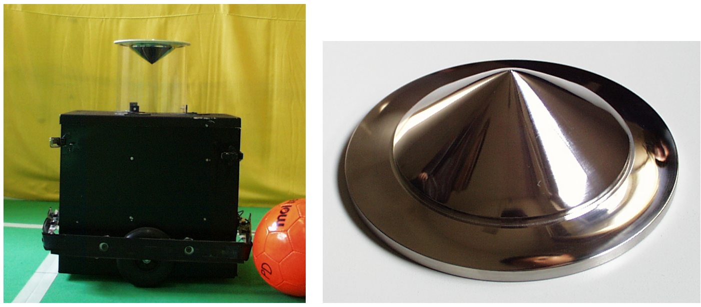

Fig. 1. (Left) Our goalkeeper robot. (Right) Our multi-part mirror

In this paper we follow the approach of [5], but we changed totally the characteristics of the mirror. In Figure 2, we sketch the image formation mechanism in a omnidirectional vision sensor. Consider the point P laying on the field of play. Using the pin-hole camera model and the optical laws, for every point at a distance d_{OP} from the sensor, it is possible to calculate the coordinates (x, y) of the corresponding image point P' on the CCD. Vice versa, knowing the coordinate (x, y) of a point in the image plane, we can calculate the distance d_{OP} of the corresponding point in the world (for a detailed theory of catadioptric image formation, refer to the homonym paper of Nayar [1]). Because of the finite size of the sensitive element of the CCD, we do not have access to the actual coordinates (x, y) of the image point, but to the discrete corresponding pair (x_d, y_d) (i.e. the location of the corresponding pixel of the CCD). So, if we calculate the distance d_{OP} from (x_d, y_d), it will be discrete. The discrete distance deviates from the actual distance with an error that depends on the geometry of the mirror.

2 How to design a mirror

In this section we expose an algorithm for the design of a custom mirror profile. The algorithm is a modification of the one presented in [5]. The starting point is to identify the function that maps point of the world into points of the CCD. This is a function $f : R^3 \rightarrow R^2$ that transform world point (X',Y',Z') into image points (x,y). Actually, what we want it is a simpler function. We want a function that maps points laying on the plane Y=0 around the sensor from a distance D_{MIN} up to a distance D_{MAX}. Therefore, exploiting the rotational symmetry of the system, the problem can be reduced to consider a function $f^* : [D_{MIN}, D_{MAX}] \rightarrow [0, d_{MAX}]$ where d_{MAX} is the maximum distance on the CCD. The exact solution of this problem can be found with a quite complex differential equation. In [4] a solution of this equation is reported for the simple case $d = f^*(D) = KD$. In [5] is reported an approximated solution. We used this same solution, consisting in a local linear approximation of the mirror profile with its tangent. Let us see in the detail the algorithm exploited.

1. Discretise the interval $[D_{MIN}, D_{MAX}]$ in a set $[D_{MIN} = D0, D1, D2..., D_N = D_{MAX}]$ that will be mapped by f^* in the set $[0, d_1, d_2, ..., d_N = d_{MAX}]$;
2. the tip of the mirror is in $P_0 = (0, Y_0)$ and the tangent to the mirror is $tan(arctan(D_{MIN}/y_0)/2)$. With this choice the point at distance $D = D_{MIN}$ is mapped into d=0. Let us call r_2 the line passing by P_0 whose derivative is $tan(arctan(D_{MIN}/y0)/2)$;
3. r_1 is the line passing by the focal point (0, f) and the point $(-d_1, h)$ on the CCD, where h is the height of the CCD on the ground. The intersection between r_1 and r_2 determines the point P_1. The line r_3 will be created as the line passing by the point P_1 and the point $(D_1, 0)$. Now the line r_3 and r_1 are the path the light has to follows if we want to map the world point $(D_1, 0)$ into CCD point $(-d_1, h)$. Therefore the tangent to the profile of the mirror in the point P_1 must be perpendicular to the bisector formed by r_3 and r_1. And so on, until all the point in the set $[D_{MIN} = D_0, D_1, D_2..., D_N = D_{MAX}]$ are mapped in the set $[0, d_1, d_2..., d_N = d_{MAX}]$;
4. The mirror profile is obtained joining the point P_i, i=0, 1, 2, ..., N;

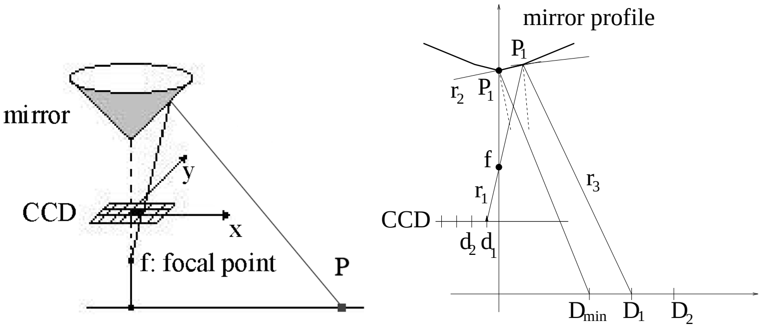

Fig. 2. (Left) Image formation sketch (Right) The geometrical contruction of the custom mirror

Now, we know how to design a mirror that realises a custom function mapping point of the world into points of the CCD sensor. In the following we will present the choices we made for the mapping function.

3 Comparison between different type of mirrors

To better understand the advantages and the disadvantages of the different shape of mirrors, we summarised them in Table 1.

Table 1. Comparison between different mirrors type

Mirror	**Advantages**	**Disadvantages**
Conic	– With an opportune choice of the geometric parameters it is possible to eliminate from the image the body of the robot. – Small *relative* error on the calculation of distances from the sensor.	– Maximum measurable distance is low. – Bad resolution close to the sensor.
Conic with spherical vertex	– Good resolution close to the sensor. – Small *relative* error on the calculation of the distances from the sensor.	– The inner part of the image is not utilizable because occupied by the body of the robot. – Maximum measurable distance is low.
Isometric	– Apparent size of the objects does not depend on the distance from the sensor. – Constant *absolute* error on the calculation of the distances from the sensor. – It is possible to perform the conversion from distances on the image to distances on the playground via a simple multiplication.	– The inner part of the image is not utilisable because occupied by the body of the robot. – Big *relative* error on the calculation of the distances for objects close to the sensor.

4 The mirror we designed

As we stated before, we designed a multi-part mirror composed by a inner part and two circular rings. As in [5] the two circular rings are devoted to the observation of the robot markers and of the area surrounding the robot. Let us consider first the inner part of the mirror, what we called the *measurement mirror*. This part is the one that observes the wider area of the field of play. For the *measurement mirror* we had two possibility: a mirror with a continuous curvature or a mirror with a discontinuity in the vertex, like conical mirrors. The latter has been our choice. In fact, in a multi-part mirror is possible to eliminate the disadvantages presented by a conic mirror, exploiting other sections of the mirror. Consider Table 1 and the disadvantages of conic mirrors there presented. The first disadvantage is eliminated by the middle circular ring of the mirror, i.e the *marker mirror*, that permits a long range field of view. The second is overcome by the outer part of the mirror the *proximity mirror*, that permits to have a very good resolution near the body of the robot. Therefore, considering the whole structure of the mirror we have no drawback to the advantages introduced by the conic mirror. Nevertheless, are we ready to renounce to the important features of the isometric mirror proposed in [5]? In an isometric mirror, the image of the field of play is not distorted. Usually, this implies that the apparent dimensions of an object does not change with the distance from the sensor. Moreover, it is possible to calculate the distance of the object from the sensor via a simple multiplication. This is not possible with a distorting mirror, like a conic one. We can overcome the problem using a look-up table (LUT). The LUT should contain an association between the coordinates of points in the image and the coordinates of the corresponding points in the real world. Thus, the velocity in calculating the position information for a point in the world is preserved. The second advantage of an *isometric mirror*, i.e. constant absolute error, is not such an advantage in our minds. In fact, a constant absolute error implies a bigger relative error at small distances. In the Robocup environment we need a good precision in the determination of the position of objects near the robots, while we can allow a certain amount of absolute error for distant objects.

4.1 The measurement mirror

We chose to design a mirror that maps world points on the CCD with a constant *relative error* α, i.e. an error not depending on the distance from the sensor. So, the position of an object close to the sensor will be determined with good precision while an object far away from the robot will have a sensible absolute error. Let us see how we can design such a mirror. We identify a set of discrete positions in the world and the corresponding discrete positions on the CCD sensor. We fix the minimum distance from the sensor for a visible object, called D_{MIN}. All the points at a distance D_{MIN} are mapped into the central pixel of the CCD (distance d_0). In the pixels contiguous to the central one (distance d_1) we will map the points at a distance $D_1 = D_{MIN} + \alpha D_{MIN}$, in the next pixels (distance d_2) we will map pixels at distance $D_2 = D_1 + \alpha D_1$, and so on. In such a way we map the set of point in the world $[D_0 = D_{MIN}, D_1, D_2 ..., D_N = D_{MAX}]$ into the set $[0, d_1, d_2 ..., d_N = d_{MAX}]$ in the image. Therefore if a point has a distance D from the sensor such as $D_{i-1} < D < D_i$ it will be mapped at a distance d from the centre of the CCD such as $d_{i-1} \leq d \leq d_i$. The worst case is that the point P at distance D_{i-1} will be mapped into the pixel at distance d_i. If we reconstruct the position of P from the image we have an error $e = D^* - D_{i-1} = D_i - D_{i-1} = \alpha D_{i-1} = \alpha D$ and then a relative error α. The resulting profile is the first part of the mirror profile sketched in Figure 3 and it maps the world points into the area I_1 of the CCD.

4.2 The marker mirror

In the Robocup competitions every robot has to wear a coloured marker that identifies the team the robot belongs to. The marker must be positioned at a maximum height of 60cm. Therefore, we need a part of the mirror pointing to objects over the field of play. This implies to see objects out of the game arena. The vision of object out of the field of play causes troubles to the vision systems of the robots that are designed for the highly structured Robocup environment. In order to reduce the bad influence caused by seeing many unknown objects, we dedicated just a small portion of the image to the vision of the markers. The light reflected by the *marker mirror* will be mapped in a low resolution annular ring: area I_2 in Figure 3(left). In this area we do not care about precision of the measurements, we want only to be able to detect the markers and to associate them to the robots localised with the *measurement mirror*.

4.3 The proximity mirror

As we stated before, the *measurement mirror* produces low resolution images close to the sensor. To clearly see object close to the body of the robot (one of the most important part of the pitch) we designed the *proximity mirror*. The proximity mirror is the outer part of the multi-part mirror, so it has the most convenient position to observe objects close to the body of the robot. This part is designed with a low curvature and a quite large portion of the image is dedicated to the light it gathers. Thus, object close to the robots have quite big apparent dimensions. To enhance this effect we made this latter part concave, so points closer to the robot are mapped in outer points of the image, Figure 3(left). Because the body of our robot does not have a circular symmetry (it is a rectangle $37cm \times 27cm \times 30cm$), some portion of the robot's body (the corners) will appear in the image, Figure 4. We can not avoid this if we want to see objects as close as possible to the body of the robot. Figure 3 presents a sketch showing how the different areas of the world are mapped on the image. Note that there is a region that is seen by both the measurement mirror and the proximity mirror. This is not a problem, on the contrary, can be used to perform more precise measures.

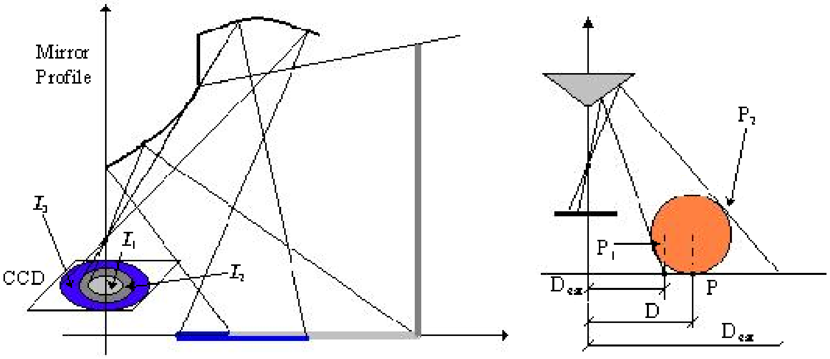

Fig. 3. (Left) A sketch of the multi-part mirror profile. (Right) The closest and farthest pixels of the ball

5 The omnidirectional software

The designed omnidirectional mirror has been mounted on a goalkeeper robot, called Lisa. Lisa has been the reserve goalkeeper of the Azzurra Robot Team (ART), the Robocup Italian National Team. Lisa has played successfully some of the games in Robocup Euro 2000 in Amsterdam, where ART won the second place cup. As a goal keeper, Lisa is required to present high reactivity and high precision in the calculation of trajectories. To achieve a high reactivity, the vision software should be able to process at least 10 frames per second. Because the hardware of this robot has limited capabilities (the CPU is a K6 200MHz with

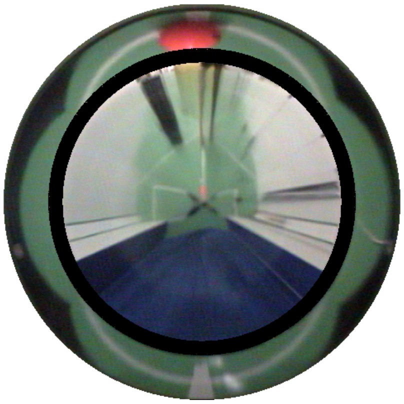

Fig. 4. An image taken with our multipart mirror. Note the ball both in the measurement and in the proximity mirror and the corners of the robot body on the outer annular ring

64 MB of RAM memory), we decided not to process the whole image, but to extract only the information needed by a goalkeeper: its position with respect to the goal and the position of the ball. In particular, we did not process pixels recognized as belonging to other robots and pixels in the area of the image containing the markers information. This was possible because we did not need to build a behavior of collision avoidance for the goalkeeper. The rules of the Robocup competitions consider every contact between a player and a goalkeeper as a foul move against the goalie.

5.1 Processing the image

The first operation performed on the image acquired by the omnidirectional sensor is to identify the pixels belonging to the ball and to the goal. Because the colours in the Robocup field of play are coded, we simply need to locate red pixels to find the ball and, to find the goal, yellow pixel (or blue pixels depending on the goal the robot is defending). The search for these pixels is not performed on the whole image but just on *samples* of it. There is a subset of pixels of the image, called *receptors* [3] disposed in a spiral starting from the centre of the image. During the first scan of the image, just these receptors are analysed. As soon as the colour of one of these receptors corresponds to the colour of an object of interest (the ball or the defended goal) the scan is interrupted. A `CreateBlob Algorithm` starts. This algorithm builds the blob corresponding to all the connected pixels of the same colour. Once the blob is completed the scan of the receptors continues. This procedure is chosen in order to avoid scanning every single pixel of the image, that would be too time consuming.

5.2 Calculating the Ball Position

Once the blob corresponding to the ball is built, the ball location is calculated from the position of its closest and farthest pixel, Figure 3(Right). The azimuthal coordinate of the ball corresponds to the azimuth of the closest and farthest pixel. The radial coordinate is more complicated. As we stated before, the distance of points seen in the image is calculated as if the point were on the field of play. As depicted in Figure 3(Right), neither the closest nor the farthest pixel corresponds to the actual point of support of the ball. There is an error $e = D - D_{est}$. Knowing the dimensions of the ball it is possible to correct this error. In the calculation of the actual support point of the ball, we used both the closest and farthest pixel. Using two estimates of the ball distance it is possible to reduce the position error. Moreover, it is possible to calculate the position of the ball even in situation in which only one of the two points is visible (for instance, if the ball is too close to the robot's body, the closest point is not visible to the sensor). To speed up this calculation, we created two look-up tables. The first contains the association between every image pixel and the corrected ball distance calculated from the closest pixel of the ball. The second contains the same data but calculated using the farthest pixel of the ball. The precision attainable with this method depends on the precision of the vision system, so as we stated before, it is better than 3%.

5.3 Localising the robot using the goalposts

A goal keeper should always know his position with respect to the goal he is defending. Our robot uses the odometric information to calculate its position. The odometers suffer of cumulative errors, so time by time they need to be reset. To precisely determine the robot location we used the information on the apparent position of the goalposts. The goalposts are easily detected with an omnidirectional vision sensor. They are mapped, as all vertical lines, into radial lines in the image. We choose to extract from the image the azimuths of two goalposts and the radial distance of one of the two. Resetting the position of the robot during the game is dangerous, even steady objects appear to move. If the goalkeeper re-locates its-self during a shoot, the ball appears to experience a sharp deviation from its direction, invalidating the reliability of the trajectory predictor module. Therefore, we decided to perform a re-location process just every 10 seconds, even if our robot could re-locate twice a second.

6 The Goal-Keeper Behavior

The designed omnidirectional mirror permitted to create an innovative moving for the robot. Most of the Robocup Teams have their goalkeepers moving along the goal line. Our robot moves on the semi-circumference, as shown in Figure 4a). This permits to design peculiar behavior for a more effective moving. Up to now, The deliberative level of our robot handles four cases triggering different behaviours.

Case 1: ball not seen. If the robot cannot see the ball or if the ball is farther than the midfield line, a behaviour called `CentreGoal` is activated. This positions the robot at 60 cm from the goal line, parallel to it. This is a *waiting behaviour*, i.e. the robot positions in the best location to reach the ball whenever the ball will be seen again.

Case 2: inactive ball. The ball is steady out of the penalty area or it aims out of the goal. In this case the ball is not *dangerous*. The behaviour called `ProtectGoal` is activated. The robot moves to cover the goal mouth. It moves to the point where the line joining the ball and the centre of the goal intersects

the semi-circumference of 60 cm radius centred in the goal centre, Figure 5a). As it is easy to understand from Figure 5a) and Figure 5b), this disposition, compared with the one adopted by most Robocup team, permits to protect a bigger position of the goal mouth and to intercept faster a shot.
Case 3: shot in goal. If the ball is moving, the *Arbiter* module calculates the line joining the current and the previous position of the ball. If this line intersects the goal line, the ball is heading the goal. The `CatchBall` behaviour is activated. The robot moves in a straight line toward the point of intersection of the shot line and the line on which is the robot. This is a simple linear motion that combines high reactivity with good stability, avoiding problems that occur when the robot maneuvers or makes fast turns, Figure 6a)
Case 4: dangerous ball. This is one of the most delicate situations. The ball was not catch with one of the previous behaviour and now it is steady within the penalty area. In this situation, a wrong move of the robot, trying to sweep the ball out of the area, could result in pushing the ball in its own goal. Up to now we do not have a behaviour dedicated to this event. So, we find a trick that resulted to work properly on the field. We take advantage of a Robocup rule stating that if the ball is steady for more than 10 seconds, it will be repositioned by the referee. So, we try to shield the ball from the other robots with the behaviour used in *Case 2*, i.e. `ProtectGoal` behaviour, Figure 6 b). This solution could be considered a kind of *emerging behaviour*, because it was not coded for this case but it resulted to be effective. On the other hand, there are some situation in which this solution does not work. For instance, if the ball is close to the centre of the semi-circle, the robots will move close to a goalpost, making it easier for the opponents to score, Figure 5c).

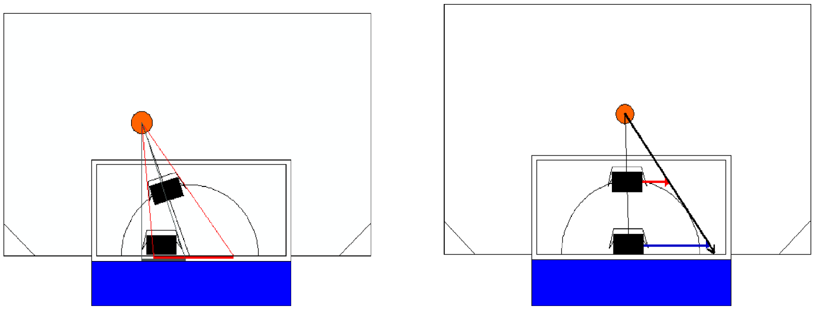

Fig. 5. Comparison between the proposed position of the goalkeeper (upper robot) and the standard position (lower robot).

7 Conclusion

In the first part of this paper we showed how to design a custom mirror for an omnidirectional sensor. The application we chose is the Robocup competition. The mirror produced with the profile outlined in this paper, showed to be reliable and effective in the Robocup domain. In the meanwhile we are working

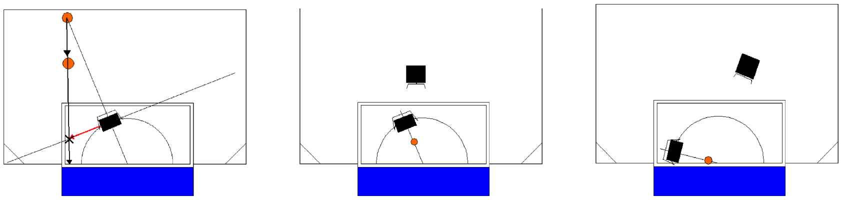

Fig. 6. (Left) Intercepting a shot. (Middle) Shielding the ball. (Right) Bad shielding of the ball.

with mirrors designed for other tasks, as navigation and mapping in unknown environments [7]. In the second part, we presented the behaviour that have been implemented to exploit the custom design of the mirror. The synergy of the custom mirror and the dedicated behaviour proved to be effective in field of play.

References

1. Simon Baker and Shree K. Nayar. A theory of catadioptric image formation. In *Proceeding of the 6th Intern. Conf. on Computer Vision*, January 1998.
2. Andrea Bonarini. The body, the mind or the eye, first? In Manuela Veloso, Enrico Pagello, and Hiroaki Kitano, editors, *RoboCup99: Robot Soccer World Cup III*, volume 1856 pp. 210-221 of *LNCS*. Springer, 2000.
3. Andrea Bonarini, Paolo Aliverti, and Michele Lucioni. An omnidirectional vision sensor for fast tracking for mobile robot. *Proceedings of the IEEE IMTC99*, 1999.
4. Andrew Hicks and Ruzena Bajcsy. Reflective surfaces as computational sensors. In *PProc. of the Second Workshop on Perception for Mobile Agents, Fort Collins*, pages pp. 82–86, 1999.
5. Fabio Marchese and Domenico G. Sorrenti. Omni-directional vision with a multipart mirror. *The fourth international workshop on robocup*, pages pp. 289–298, 2000.
6. Carlos F. Marques and Pedro U. Lima. Vision-based self-localization for soccer robots. In *Proceedings of the 2000 IEEE/RSJ International Conference on Intelligent robots and systems*, 2000.
7. Emanuele Menegatti, Enrico Pagello, and Mark Wright. A new omnidirectional vision sensor for the spatial semantic hierarchy. In *International Conference on Advanced Intelligent Mechatronics (to appear)*, July 2001.
8. Shoji Suzuki and Minoru Asada. An application of vision-based learing in robocup for a real robot with an omnidirectional vision system and the team description of osaka university "trackies". In Minoru Asada and Hiroaki Kitano, editors, *RoboCup98: Robot Soccer World Cup II*, volume 1604 pp. 316-325 of *LNCS*. Springer, 1999.
9. Yasushi Yagi. Omni directional sensing and its applications. *IEICE TRANS. INF. & SYST.*, VOL. E82-D(NO. 3):pp. 568–579, MARCH 1999.
10. Yasushi Yagi, Yoshimitsu Nishizawa, and Masahiko Yachida. Map-based navigation for a mobile robot with omnidirectionalimage sensor copis. *IEEE Transaction on Robotics and automation*, VOL. 11(NO. 5):pp. 634–648, October 1995.