

# The Body, the Mind or the Eye, first?\*

Andrea Bonarini

AI and Robotics Project, Dipartimento di Elettronica e Informazione,  
Politecnico di Milano, Piazza Leonardo da Vinci, 32, Milano, Italy  
bonarini@elet.polimi.it

**Abstract.** We present an approach to shape robots on their sensorial ability. We argue that the interface with the external world may strongly condition the design of a robot, from the mechanical aspects to reasoning and learning. We show the implementation of this philosophy in the RoboCup middle-size player Rullit, shaped on its omnidirectional vision sensor.

## 1 Introduction

We argue that the way a robot perceives the environment should strongly affect the design of each component of the robot. As it happens for animals, and human beings, the modality of interaction with the external world is strongly related to the survival behaviors, neural structures, actuators and reasoning. For instance, all the most evolved predators in natural life have eyes pointing forward, since this makes easier to follow a prey; all the preys have lateral eyes, since this is effective to be aware of the presence of predators. In some special cases (e.g., the chameleon) predators have highly movable eyes that enable cost-effective chasing strategies. We share the opinion that in a systemic perspective all the components of an agent are inter-related. However, in designing an artificial autonomous agent we should start from somewhere; here, we propose to start by considering the component that provides input, since this is the most critical to achieve the desired behavior. In the F-2000 Robocup [3] [14] environment, for instance, a black and white camera or a sonar belt, could hardly be enough to play effectively, whereas they may provide enough information to achieve many other tasks.

We have followed this approach of "shaping an agent on its sensor" (*SAS*) in many projects, designing both real [5] [1], and simulated [4] [8] robots by starting from the definition of the sensors the agent can exploit to operate in

---

\* This research has been partially supported by the MURST Project CERTAMEN.

The camera has been kindly provided by Sony Italia. I have to thank G. Borghi, who introduced me to the world of omnidirectional vision, R. Cassinis, who first implemented an Italian Robocup player around an omnidirectional sensor (the ART goalie, Saracinescu), and all the members of the Politecnico di Milano AI&R Robocup Team for their effort in the implementation of Rullit: P. Aliverti, M. Bellini, G. Contini, N. Ghironi, P. Lucioni, C. Mambretti, A. Marangon, P. Meriggi.

its environment. We have implemented Rullit, our F-2000 Robocup player (see figure 1), following this philosophy, after the first experience in Robocup98 [2]. In Rullit, the importance of a sensor adequate to the soccer playing task in the Robocup environment becomes evident, and we take it as a running example to discuss the SAS approach. In this paper, we first motivate the need for accurate



Fig. 1. Rullit, our Robocup F-2000 player.

design of the sensorial apparatus able to extract the needed information from the environment; we discuss the general issues exemplified in the Robocup environment. Then, we will describe our sensor. Finally, we discuss how a sensor may condition the design of a robot, considering the kinematics and mechanical aspects, low level control, behaviors, and learning mechanisms. This also gives us the possibility to describe the main features of our robotic agent.

## 2 Designing the sensor

An agent needs information to perform a task. It obtains such information by elaborating sensor data. Reasoning may somehow fulfill lacks of information by completing the available data by inference, possibly increasing uncertainty and approximation. An accurate design of the sensorial apparatus may reduce such undesired factors. We first discuss general properties that should be considered, exemplifying them in the Robocup context.

## 2.1 What kind of information?

The first question we have to ask ourselves is: "What kind of information will my agent need?" From the answer to this question we may decide the kind of sensor we need to acquire data, and how to extract the information from raw data. Here below, we mention some relevant aspects.

**Information contents** What the agent has to know? For instance, in the Robocup context, we may decide that it is interesting to know the position of the agent and the relative positions of ball, goals, and other agents, maybe discriminating between teammates and opponents, or maybe identifying each single robot. Moreover it may be interesting to know this information for any object on the field: the more the agent can perceive, the less it has to rely on inference, presuppositions, expectations, and information explicitly coming from teammates.

**Information quality** Should the agent be certain about the facts it is inferring from data? Can it work also with uncertain facts? Should it explicitly represent uncertainty? What kind of precision is needed? In Robocup it may be interesting to have a good precision in the neighborhood of the agent, for instance to control the ball, and to interact with close players. What happens at a great distance may be considered qualitatively, since the environment is rapidly changing, and whatever happens far from the agent do not require precise intervention before than significant changes may occur.

**Information acquisition rate** How frequently should the information be updated? The acquisition rate should allow to build an effective model of the events that characterize the task. The fastest event on a Robocup F-2000 field is the movement of the ball, which may run at more than 1 m/sec. The information acquisition and management should be fast enough to allow enough time to act (or react), but, at the same time, it should not be too fast since this would produce a large amount of data that has to be interpreted with the computational resources available (on board).

**Information abstraction level** Which kind of abstraction from raw data do we expect to need, in order to obtain the required information? Can we reason on raw data or on abstractions, and which kind of abstractions? In Robocup, we are interested in the above mentioned information, which requires a good abstraction and classification activity on raw data.

**Acquisition robustness and adaptation** Is the environment known, stable and static? If it is not so, data acquisition should be robust and possibly able to adapt to changing conditions. Although the Robocup rules seem to define a highly structured environment, it is not so on the real field. Illumination is never

as expected, and the presence of objects in the visual field may make it changing a lot during a match.

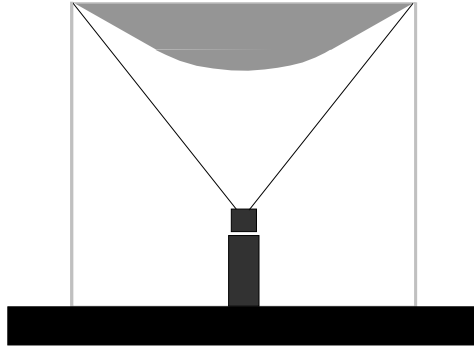
## 2.2 Why omnidirectional vision?

At Robocup98, most of the robots relied on vision sensors, since the organizers seem strongly oriented to give colors a primary role in the field setting. Most of the players (included those in our team, *ART - Azzurra Robocup Team* [12]) had a fixed, color camera, that could hardly match some of the specifications mentioned above. For instance, it gives information only about a small number of objects on the field, so that, in many situations, it is hard to self-localize the agent, to know where is the ball (which runs really fast), or to understand what is happening. Our team, as others, implemented strategies to patch this lack of information, based on information exchanging, which partially failed because of transmission problems. Other teams had mobile cameras (or high mobility of the body), but most of them still seemed too slow and imprecise to keep track of the fast events on the field. The '98 winner [10] had a sensor system matching perfectly all the design requirements mentioned above, and it was one of the few teams showing really interesting behaviors. It is also to be noticed that their sensor did not follow the organizers' implicit suggestion to rely on color vision, thus avoiding most of the problems related with this type of sensor on the Paris Robocup field.

We have decided to answer to the above introduced "first question" with an omnidirectional vision sensor, which is described in details elsewhere [7]. It consists of a camera pointed upwards beneath a coaxial, revolution mirror obtained by the intersection of a truncated cone and a sphere (see figure 2). A single image contains all the objects around the agent. The data acquisition system can give with sufficient precision distance and direction from all the objects around the agent. We have designed it to exploit the camera resolution and to improve radial resolution in the peripheral areas of the circular image, containing far objects. It is thus possible to reliably detect objects such as the ball, up to 6 meters from the agent. The precision about the distance from the objects is inversely proportional to the distance itself. Uncertainty about data classification is very low, due to the image analysis system we have implemented. This is also optimized to provide all the information once every 30 ms, giving an information acquisition rate very close to the limit of the PAL European standard for video frame acquisition, which is 25 frames/sec. Since an omnidirectional image contains at the same time a large portion of the field, the average brightness is quite stable, and adaptation to light intensity is limited to the first frames, to become acquainted with a new field. In case of a standard camera pointing towards the environment, the image may contain objects with different colors (e.g., black robots, or white walls), and this requires some compensation on the average brightness, which can be obtained either by mechanical adjustment of the camera iris (slow), or electronically, requiring the analysis of at least two images unavailable for object recognition. In the next section, we give some details about the specific choices we have done in the implementation of our sensor.

### 2.3 Our omnidirectional vision sensor for Robocup

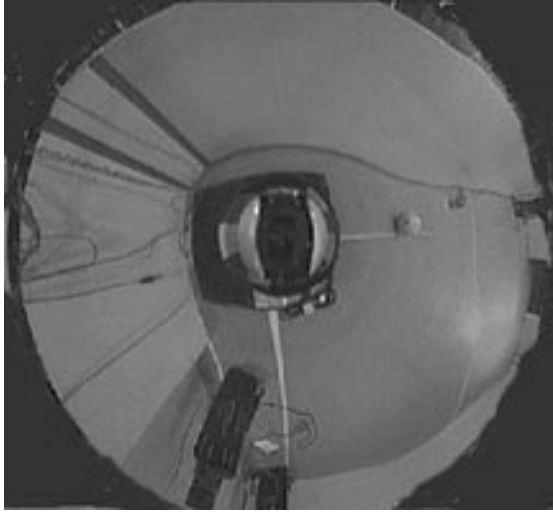
The sensor we are proposing is represented in figure 2. You can see that the



**Fig. 2.** The mirror, the camera pointing upwards, the plexiglass cylinder supporting the mirror and the visual angle of the camera.

central part of the mirror consists of a truncated sphere, making it possible to see objects very close to the robot (15 cm). Tangent to this is a truncated, reversed cone, giving enhanced radial resolution from 2 m to 6 m. This design does not require adjustment of the focal length of the camera, as proposed by [17], thus avoiding the time loss due to mechanical movements. The camera with which we took the image shown in figure 3 is a low cost card camera, having  $512 \times 582$  sensible elements, and a view angle of about  $60^\circ$ . We are now mounting a Sony XC-999P. In figure 3, in black, on the center of the image the body of the robot, on the right the yellow goal and the goalie, on the top a ball and, at a distance, another robot and the blue goal. To implement a fast image recognition system we took hints from biology, basing it on the idea of *receptor*. The generic term "receptor" is used for any biological unit able to perceive specific stimuli from the outside world and to transform them in nervous signals, then transmitted to the central nervous system. In computer vision, image pixels are often considered as receptors. To improve computational speed, we consider specialized receptors, each consisting of a 3 by 3 pixel matrix, and characterized by the averaged *HSV* value. Our receptors are distributed in a pattern designed to detect the smallest object on the field (the ball) in any position of the image. Thus, we analyze only a grid of receptors on the image, thus reducing the amount of information to be considered by more than two orders of magnitude.

The vision system we have designed firstly estimates on the image the likely position of the possibly interesting objects, by classifying the receptors, and aggregating them in clusters (called *target*) by color similarity and adjacency. A target is a part of the image where it could be present an interesting objects. Once identified the targets, it is possible to operate on the part of image defined



**Fig. 3.** A typical image taken by our sensor.

by each of them by adopting classical image processing (such as blob growing) on single pixels, object recognition and localization techniques. The application of these techniques significantly reduces the amount of information to handle and increases the processing speed. The image acquisition and processing requires less than 30 ms, on the on board PC, a 266 MHz AMD K6 CPU, with 32 Mb of RAM, a Matrox Meteor frame grabber, Linux RedHat 5.2 (Kernel version 2.0.36), and real-time kernel ETHNOS [13].

### 3 Sensors and behaviors

Now, let us discuss the impact of the type of available, sensorial information on the behaviors that can be implemented on the agent. We first consider the low-level control aspects, and then the higher level behaviors and strategies

#### 3.1 Low-level control

In our viewpoint about robot architecture, a low-level control system may be present on an agent to provide the higher levels with reliable actions. If the higher level behavior activation cycle is long, low-level control should ensure that the desired actions are actually done as expected. For instance, if the high level control states that the agent should turn 30 degrees on the left at a speed of 0.2 m/sec, it expects that this happens; if, as usual, the actuation is imperfect, the actual action may be different. We may either have a low level control system trying to realize what the higher-level control states, or have a higher-level control designed to cope with low level problems such as imperfect actuators,

and running fast enough. The information for such a kind of low-level control is different from that discussed above. Here, we need precise information about the movement of the robot wheels. We have to cover this need with other sensors, appropriate for this component of the control system, namely encoders on the wheels or on the engine axis.

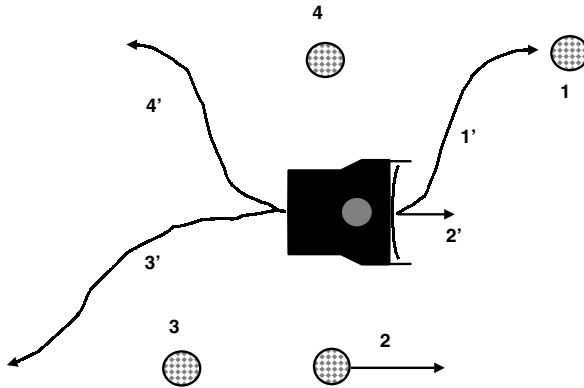
Rullit, our Robocup agent, has two independent traction wheels, and we decided to attach encoders to each wheel. The precision of each measure is less than 0.1 mm, enough to implement a good, speed and jog control. We have implemented it as a fuzzy controller, so that it is also quite robust with respect to noise [11]. Notice that this same sensor (encoder) is known to be inappropriate for position control [9], and that we did not implement such a kind of controller. This is another example of the relevance to select the proper sensor to achieve a task.

### 3.2 Behaviors

The Robocup environment changes so rapidly that we have decided to leave to the behavioral control the decision about where to go: a plan to reach a position should be probably continuously refined, since situations change rapidly, so a controller able to bring the agent at a given position would be restarted too often.

Information provided by the omnidirectional sensor is appropriate for high level control, and influences the design of behavioral modules. We may notice that the same behavior (for instance, *Go-To-Ball*, that brings the agent on the ball) may be implemented in different ways according to the available information. If we had reliable information only about objects that are in the range of a camera pointed forward, probably the behavior can reliably trigger only when the ball is in the range of the camera, and another behavior will make the agent searching for the ball. Moreover, probably, *Go-to-Ball* will bring the agent on the ball, only while keeping it in sight; this may bring the agent in undesired situations, such as bumping the ball against the wall. Another implementation of *Go-to-Ball* with the same sensor may infer the position of the ball from past information and from information coming from other players. This may help, but may also lead to clumsy behaviors, such as that happened in the challenge during the ART-Freiburg semi-final at Robocup99, where the player didn't check often enough the ball position and originated a situation hard to manage.

By contrast, having reliable information about the ball in any position with respect to the agent from a suitable sensor, such as omnidirectional vision, a different *Go-to-Ball* may decide how to approach the ball, while keeping it in sight. In figure 4 you may see some of the ball approaching behaviors we have implemented relying on the available omnidirectional information; these include going on the ball by moving backwards (tracks 3 and 4 in figure 4), or tracking the ball by the side (track 2 in figure 4). We have implemented the behavioral control by *fuzzy behaviors* [15] [8], that is control modules that trigger on conditions consisting of fuzzy predicates. We consider two sets of such fuzzy preconditions: the *cando preconditions* enable the behavior, and the *want preconditions* give the

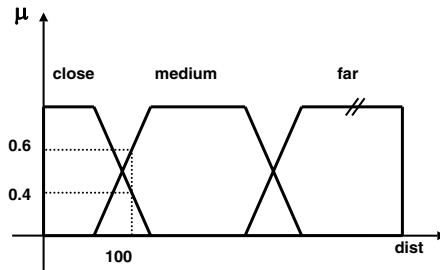


**Fig. 4.** Three possible trajectories (1', 3' and 4') to reach a position behind the ball (respectively in positions 1, 3, 4), and one (2') to track a moving ball(2).

amount of motivation for it. For instance, if we have the ball we can kick it, but we want to do it only if this makes sense, e.g., we are aligned with a free portion of the opponent's goal. Fuzzy predicates make it possible to classify the information coming from the sensor into higher level classes, which give it a meaning. Thus, it is possible to perform reasoning at a high level of abstraction, on a relatively small set of concepts [6], thus achieving high speed and robustness [11]. Moreover, a fuzzy interpretation gives the possibility to reason on overlapping classifications, which seems to be exactly what human beings, and some animals, do in most situations. For instance, in figure 5 we show the membership functions defining three fuzzy sets (*close*, *medium*, *far*) used to implement fuzzy predicates that classify the distance from objects. In the example, the measured distance (1 m) is classified as *close* with truth value 0.4 and *medium* with truth value 0.6. In real life, usually we adopt classifications which can be naturally represented by fuzzy predicates [11] whose definitions overlap, such as those presented in figure 5.

We associate to behaviors other two parameters: the static and the dynamic *relevance*. The first implements an a priori, partial ordering among behaviors, allowing to state, for instance, that avoiding crashes is always better than taking the ball. The dynamic relevance also implements a partial ordering, but it can be modified according to the situation faced by the agent, and it is used to implement strategies and learning mechanisms, as discussed in the next sections. At each high level control step we compute for each behavior instance whose cando preconditions are true above a given threshold, its *triggering level*, by composing: its two relevance values, the motivation coming from its want preconditions, and the possibility coming from its cando preconditions. As done by most biological beings, and in contrast with most of artificial fuzzy agents, the behavior with the highest triggering level is activated, and its actions done. We have a winner-





**Fig. 5.** An example of fuzzy classification: the distance value 100 [cm] is classified as *close* with truth value 0.4, and as *medium* with truth value 0.6.

take-all activation, instead of a composition of proposed actions typical of fuzzy systems, since this gives more coherence to behavior selection. If we decide that it is better to go on the ball instead than towards the goal expecting a passage, it does not make any sense to compose the two actions to obtain a hybrid whose success possibilities are questionable: it is better to take a decision and act coherently with it. If it was a wrong decision, it means that the activation conditions of the behavior modules have to be tuned, and this can be done also automatically, but only if it is clear which is responsible for the action taken [5]. This structure for behavioral modules is independent from the information we have decided to be needed for the Robocup task, but it can support it effectively. We adopt the same structure in other projects based on different information (and sensors). On the other side, the specific behaviors strongly depend on the available information both for their existence (a behavior needing unavailable information would have not been implemented), and for their specific implementation as discussed above for *Go\_to\_Ball*. The whole behavior system runs in only 7 ms on the on board PC.

### 3.3 Strategy

We consider that basic skills for an agent are implemented by behavioral modules. A higher level decision module may influence the behavior activation, acting on the priorities among behaviors, by modifying their dynamic relevance values. The strategic module we have implemented recognizes a situation by classifying high level data interpretations, again implemented as fuzzy predicates. According to this, it rearranges the dynamic relevance of the behaviors, to give the preference, in case of similarity of the other parameters, to a behavior or another. For instance, if a teammate has the ball, we may either protect it by hindering opponents, or follow its action expecting a passage. A deeper discussion of this topic is beyond the scope of this paper.

Also the instance of this module strongly relies on the available information. For instance, an omnidirectional vision sensor gives the possibility to detect a rich variety of situations without communicating with teammates. Probably, a fixed, front camera would not give by itself enough information to justify the existence of a strategic module, and also a mobile camera would be probably too slow to catch enough from the environment to detect the relevant facts that could make it possible to select strategies. For these reasons, most of the teams involved in the F-2000 Robocup Championship in 1998 either relied on communication to select strategies [10], or had simple, selfish strategies. In Robocup99, an increased number of teams had robots equipped by omnidirectional vision.

## 4 Sensors and learning

Learning and adaptation are interesting approaches to implement, or improve, control modules. We believe that, given that the present learning techniques, learning behaviors from scratch on the field is not cost-effective: a designer may develop by hand nice behaviors in less time and using less resources. The goal-keeper task [17] is simple enough to be either learnt or programmed, whereas the behaviors of the other players have to be developed on a large number of really complex situations, and have to be adapted to the behaviors of the specific opponents. Some researchers have proposed to learn more complex behaviors in the F-2000 league (such as ball passing [18]). The behaviors were learned in simulation, but hand-coded behaviors were preferred on the field. In these conditions, we consider that it is more effective to program simple behaviors and strategy modules, and then adapt them on line.

The role of sensors in learning in simulated environments is questionable. Having worked since long time on learning behaviors for simulated robots [4], we have come to the conclusion that in most cases the problems with simulation are far different from those in real world, apart from the cases where enough resources are devoted to produce sophisticated simulation environments. In particular, sensor models, and the quality of the information they provide, are usually oversimplified. Therefore, we would not like to discuss about the role of sensors in learning in simulated environments.

Adaptation is even more important than learning in applications such as a Robocup match among real robots, where the opponent's strategy is usually unknown. The quality of the incoming information is relevant for the quality of adaptation: the more and nicer information we have, the more the adaptation algorithms can exploit it to find regularities. On the other side, it would be hard to manage in real time a large amount of data.

Given the topics introduced above, we can easily imagine an adaptation mechanism it could be implemented to improve strategies to face specific opponents. For instance, an agent with the ball has to take different behaviors to contrast opponents that tend to rush on the ball, or opponents that tend to block actively any possible way to their goal. Moreover, this can only be decided on line, in real time, while playing against a specific team a specific match. We

have implemented an adaptation system based on reinforcement. At present, it works only on one-to-one strategies. The strategic module classifies the situation by evaluating fuzzy predicates that interpret the incoming information, compared with information previously acquired. Then, it selects one among some predefined strategies (each corresponding to a set of relevance values for the behaviors), and provides a value of dynamic relevance for the involved behaviors. This is repeated at each high level control step, always considering the same strategy, for coherence reasons, until the situation changes. At this time, a reinforcement is computed by evaluating the new situation, and it is used to update the value  $Q(s)$  of the selected strategy  $s$  according to the standard formula:

$$Q^t(s) = Q^{t-1}(s) + \alpha (r - Q^{t-1}(s))$$

We share the motivation for using this formula, and the general background with other researchers who studied learning in the Robocup framework in other leagues [16], but we consider different models to learn, more appropriate for adaptation in the F-2000 league.

## 5 Conclusions

While in nature we assisted to the co-evolution of sensor, motor and neurological apparatus, we claim that in robotic agent design we need to design first the appropriate sensors to get the quantity and quality of information we need to achieve a task; then, we may try to shape the other components of the robot architecture on this. We have discussed in details how we have applied this approach in the development of Rullit, our Robocup F-2000 player. We argue that most of the other components of a robot architecture could be designed in many different ways, but that the information provided by our sensor is appropriate and essential to effectively face the Robocup task. We have also shown how many problems may be solved at the sensor level, thus reducing the computational effort.

Rullit is built on our *Mo2Ro (Modular Mobile Robot)* base, which also provides the mechanical and electronic basic modules for other robots, built around other sensors, namely: *Pop-eye*, which has a camera mounted on top of a 5 DOF ultra-light arm on board, and *RoboCOPIS*, which adopts a standard black and white COPIS sensor [19].

## References

- [1] G. Agazzi and A. Bonarini. Problems and solutions in acquisition and interpretation of sensorial data on a mobile robot. In *IEEE Instrumentation and Measurement Technology Conference - IMTC99*, Piscataway, NJ, 1999. IEEE Computer Press.
- [2] M. Asada, editor. *RoboCup 98: Robot Soccer World Cup II*, Paris, F, 1998. Eurobot Project - CEC.
- [3] M. Asada, P. Stone, H. Kitano, A. Drogoul, D. Duhaut, M. Veloso, H. Asama, and S. Suzuki. The robocup physical challenge: goals and protocols for phase

- i. In H. Kitano, editor, *RoboCup 97: Robot Soccer World Cup I*, pages 42–61. Springer-Verlag, Berlin, D, 1997.
- [4] A. Bonarini. ELF: Learning incomplete fuzzy rule sets for an autonomous robot. In Hans-Jürgen Zimmermann, editor, *First European Congress on Fuzzy and Intelligent Technologies – EUFIT'93*, volume 1, pages 69–75, Aachen, D, 1993. Verlag der Augustinus Buchhandlung.
- [5] A. Bonarini. Evolutionary learning of fuzzy rules: competition and cooperation. In W. Pedrycz, editor, *Fuzzy Modelling: Paradigms and Practice*, pages 265–284. Kluwer Academic Press, Norwell, MA, 1996.
- [6] A. Bonarini. Reinforcement distribution to fuzzy classifiers: a methodology to extend crisp algorithms. In *IEEE International Conference on Evolutionary Computation – WCCI-ICEC'98*, volume 1, pages 51–56, Piscataway, NJ, 1998. IEEE Computer Press.
- [7] A. Bonarini, P. Aliverti, and M. Lucioni. An omnidirectional vision sensor for fast tracking for mobile robots. In *IEEE Instrumentation and Measurement Technology Conference – IMTC99*, Piscataway, NJ, 1999. IEEE Computer Press.
- [8] A. Bonarini and F. Basso. Learning to compose fuzzy behaviors for autonomous agents. *Int. J. of Approximate Reasoning*, 17(4):409–432, 1997.
- [9] J. Borenstein, H.R. Everett, and L. Feng. Where am i? sensors and methods for mobile robot positioning. Technical report, The University of Michigan, Ann Arbor, MI, 1996.
- [10] J-S Gutmann, W. Hatzack, I. Herrmann, B. Nebel, F. Rittinger, A. Topor, T. Weigel, and B. Welsch. The cs freiburg team. In M. Asada, editor, *RoboCup 98: Robot Soccer World Cup II*, pages 451 – 457, Paris, F, 1998. Eurobot Project - CEC.
- [11] G. J. Klir, B. Yuan, and U. St. Clair. *Fuzzy set theory: foundations and applications*. Prentice-Hall, Englewood Cliffs, MA, 1997.
- [12] D. Nardi, G. Clemente, and E. Pagello. Art azzurra robot team. In M. Asada, editor, *RoboCup 98: Robot Soccer World Cup II*, pages 467–474, Paris, F, 1998. Eurobot Project - CEC.
- [13] M. Piaggio and R. Zaccaria. Distributing a robotic system on a network: the ethnos approach. *Advanced Robotics Journal*, 12(8), 1998.
- [14] Robocup. The robocup initiative. <http://www.RoboCup.org/>, 1999.
- [15] A. Saffiotti, K. Konolige, and E. H. Ruspini. A multivalued-logic approach to integrating planning and control. *Artificial Intelligence*, 76(1-2):481–526, 1995.
- [16] P. Stone and M. Veloso. Tpot-rl: Team-partitioned, opaque-transition reinforcement learning. In M. Asada, editor, *RoboCup 98: Robot Soccer World Cup II*, pages 221 – 236, Paris, F, 1998. Eurobot Project - CEC.
- [17] S. Suzuki, T. Kato, H. Ishizuka, Y. Takahashi, E. Uchibe, and M. Asada. An application of vision-based learning for a real robot in robocup - a goal keeping behavior for a robot with omnidirectional vision and an embedded servoing. In M. Asada, editor, *RoboCup 98: Robot Soccer World Cup II*, pages 467 – 474, Paris, F, 1998. Eurobot Project - CEC.
- [18] E. Uchibe, M. Nakamura, and M. Asada. Cooperative and competitive behavior acquisition for mobile robots through co-evolution. In *Proceedings of the Genetic and Evolutionary Computation Conference – GECCO99*, pages 1406–1413, San Francisco, CA, 1999. Morgan Kaufmann.
- [19] Y. Yagi, S. Kawato, and S. Tsuji. Real-time omnidirectional image sensor (copis) for vision-guided navigation. *IEEE Transactions on Robotics and Automation*, 10(1):11–22, 1994.