

The CS Freiburg Team

Playing Robotic Soccer Based on an Explicit World Model

Jens-Steffen Gutmann, Wolfgang Hatzack, Immanuel Herrmann, Bernhard Nebel, Frank Rittinger, Augustinus Topor, and Thilo Weigel

■ Robotic soccer is an ideal task to demonstrate new techniques and explore new problems. Moreover, problems and solutions can easily be communicated because soccer is a well-known game. Our intention in building a robotic soccer team and participating in RoboCup-98 was, first, to demonstrate the usefulness of the self-localization methods we have developed. Second, we wanted to show that playing soccer based on an explicit world model is much more effective than other methods. Third, we intended to explore the problem of building and maintaining a global team world model. As has been demonstrated by the performance of our team, we were successful with the first two points. Moreover, robotic soccer gave us the opportunity to study problems in distributed, cooperative sensing.

Robotic soccer is an interesting research domain because problems in robotics, AI, multiagent systems, and real-time reasoning have to be solved to create a successful team of robotic soccer players (Kitano et al. 1997). Furthermore, it is an ideal task to demonstrate the feasibility of new ideas and techniques and explore new problems.

We started to design a robotic soccer team with the intention of participating in RoboCup-98 for three reasons: First, we intended to demonstrate the advantage of our perception methods based on laser range finders (Gutmann et al. 1998; Gutmann and Nebel 1997; Gutmann and Schlegel 1996), which make explicit world modeling and accurate and robust self-localization possible.

Second, we believe that soccer is a game, where it is advantageous to base deliberation and action selection on an explicit world model, and we intended to demonstrate that such an approach is superior to other approaches. Although it is possible to play robotic soccer by

reacting on mostly uninterpreted sensor input as in pure behavior-based (Werger et al. 1998) or reinforcement learning approaches (Suzuki et al. 1998), soccer seems to be a game that has a structure that requires more than just reacting on uninterpreted sensor input. Our claim is justified by the fact that the two winning teams in the simulation and the small-size league in RoboCup-97 used this approach (Burkhard, Hannebauer, and Wendler 1998; Veloso et al. 1998). Further evidence for our claim is the performance of our team at RoboCup-98, which won the competition in the middle-size league.

Third, we intended to address the problem of multirobot sensor integration to build a global world model and exploit it for cooperative sensing and acting. In the end, we identified more problems in this area than we solved. However, we believe that it is an interesting topic for future research.

Although perception and sensor interpretation were definitely the focus of our research, it was also necessary to develop basic soccer skills and forms of multiagent cooperation to show the advantage of our approach. Although this part certainly needs improvement, it was still effective enough to be competitive. Furthermore, based on an accurate world model, our robots were much more reliable than other teams.

The rest of the article is structured as follows: In the next section, we give a brief sketch of the robot hardware. We then describe the general architecture of our soccer players and the soccer team. The next section focuses on our self-localization approach, and then we describe our player- and ball-recognition methods that are needed to create the local world model. The integration of these world

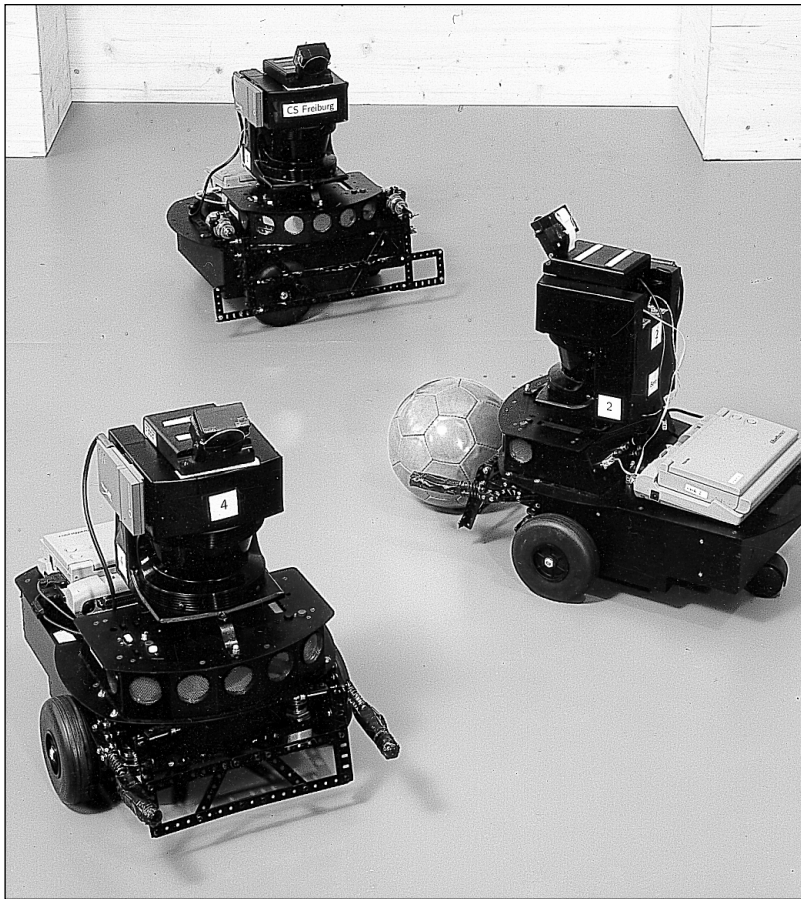


Figure 1. Three of Our Five Robots: Two Field Players and the Goal Keeper.

models into a global model and the problems we encountered are described in the next section. We then sketch the behavior-based control of the soccer agents and show how a basic form of multiagent cooperation is achieved. Finally, in the last section, we describe our experience participating in RoboCup-98 and present our conclusions.

Robot Hardware

Because our group is not specialized in developing robot platforms, we used an off-the-shelf robot—the PIONEER 1 robot developed by Kurt Konolige and manufactured by ActivMedia. In its basic version, however, the PIONEER 1 robot is hardly able to play soccer because of its limited sensory and effector skills. For this reason, we had to add a number of hardware components (figure 1).

On each robot, we mounted a video camera connected to the Cognachrome vision system manufactured by Newton Labs, which is used to identify and track the ball. For local information processing, each robot is equipped with

a Toshiba notebook LIBRETTO 70CT running LINUX. The robot is controlled using SAPHIRA (Konolige et al. 1997), which comes with the PIONEER robots. Finally, to enable communication between the robots and an off-field computer, we use the WAVELAN radio ethernet.

In addition to these components, we added PLS200 laser range finders manufactured by SICK AG to all our robots. These range finders can give depth information for a field of view with an angular resolution of 0.5 degrees, and an accuracy of 5 centimeters to a distance of 30 meters.

Handling the ball with the body of the PIONEER 1 robot is not an effective way of moving the ball around the field or pushing it into the opponent's goal. For this reason, we developed a kicking device using parts from the MÄRKLIN METALLBAUKASTEN. Furthermore, to steer the ball, we used flexible flippers that have a length of approximately 35 percent of the diameter of the ball. Although these flippers led to some discussions before the tournament, it was finally decided that the use of such flippers does not violate the RoboCup rules. In fact, we believe that taking the idea of embodiment seriously, such a ball-steering mechanism is necessary to play soccer effectively and authentically. In fact, without the flippers, it is almost impossible to retrieve the ball from the wall, which means that the referee must relocate the ball, which is annoying for everyone—in particular, for spectators. Furthermore, without the ball-steering mechanism, the ball is easily lost when running with the ball.

General Architecture

Our robots are basically autonomous robotic soccer players. They have all sensors, effectors, and computers on board. Each soccer agent has a *perception module* that builds a local world model (figure 2). Based on the observed state of the world and intentions of other players communicated by the radio link, the *behavior-based control module* decides what behavior is activated. If the behavior involves moving to a particular target point on the field, the *path-planning module* is invoked, which computes a collision-free path to the target point.

To initialize the soccer agents, start and stop the robots, and monitor the state of all agents, we use a radio ethernet connection between the on-board computers and an off-field computer (figure 3).

If the radio connection is unusable, we still can operate the team by starting each agent manually. A large number of the other teams in the middle-size league used a similar

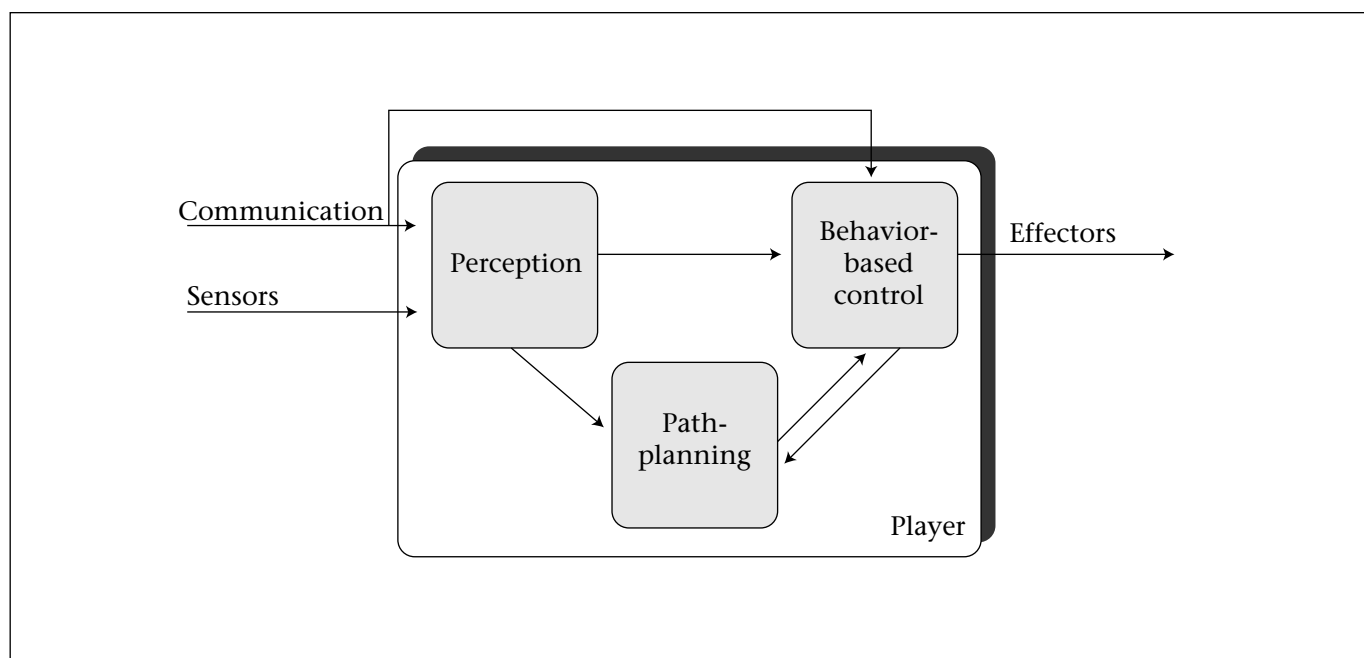


Figure 2. Player Architecture.

approach (Asada and Kitano 1999).

Unlike other teams, we use the off-field computer and the radio connection for realizing global sensor integration, leading to a *global world model*. This world model is sent back to all players, and they can use this information to extend their own local view of the world. Thus, the world model our players have is similar to the world model constructed by an overhead camera, as used in the small-size league by teams such as *CMUNITED* (Veloso et al. 1998).

Self-Localization

We started the development of our soccer team with the hypothesis that it is an obvious advantage if the robotic soccer agents know their position and orientation on the field. Based on our experience with different self-localization methods using laser range finders (Gutmann et al. 1998), we decided to use such a method as one of the key components in our soccer agents.

A number of different self-localization methods exist based on laser scans (Gutmann and Schlegel 1996; Weiß and von Puttkamer 1995; Lu and Milios 1994; Cox 1990). However, these methods are only *local*; that is, they can only be used to correct an already-existing position estimation. Thus, once a robot loses its position, it will be completely lost. Furthermore, all the methods are computationally demanding, needing 100 milliseconds to a few seconds on a modern computer. Global methods are even

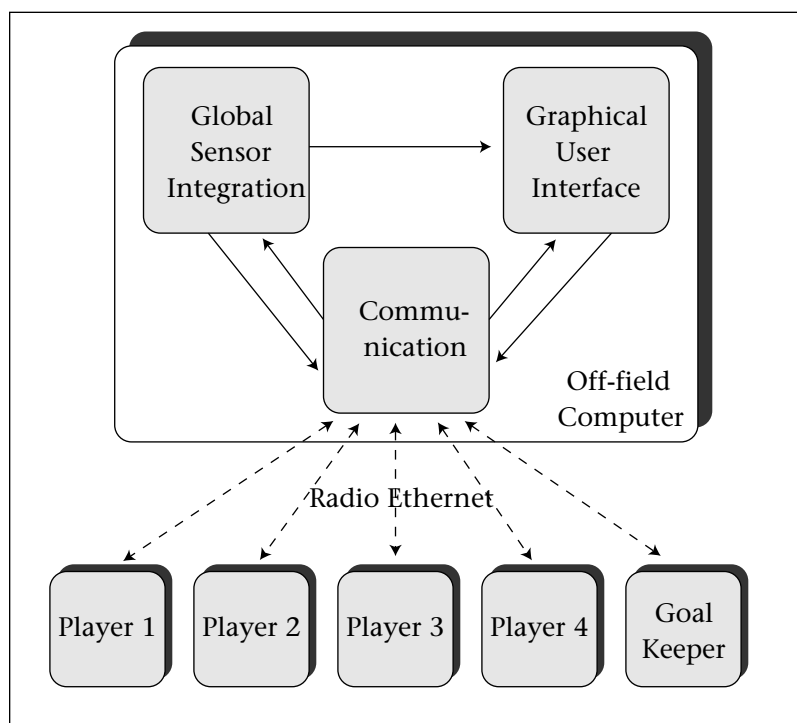


Figure 3. Team Architecture.

more costly from a computational point of view. For these reasons, we designed a new self-localization method that trades off generality for speed and the possibility of global self-localization. Our method first extracts line segments from laser range scans and matches them

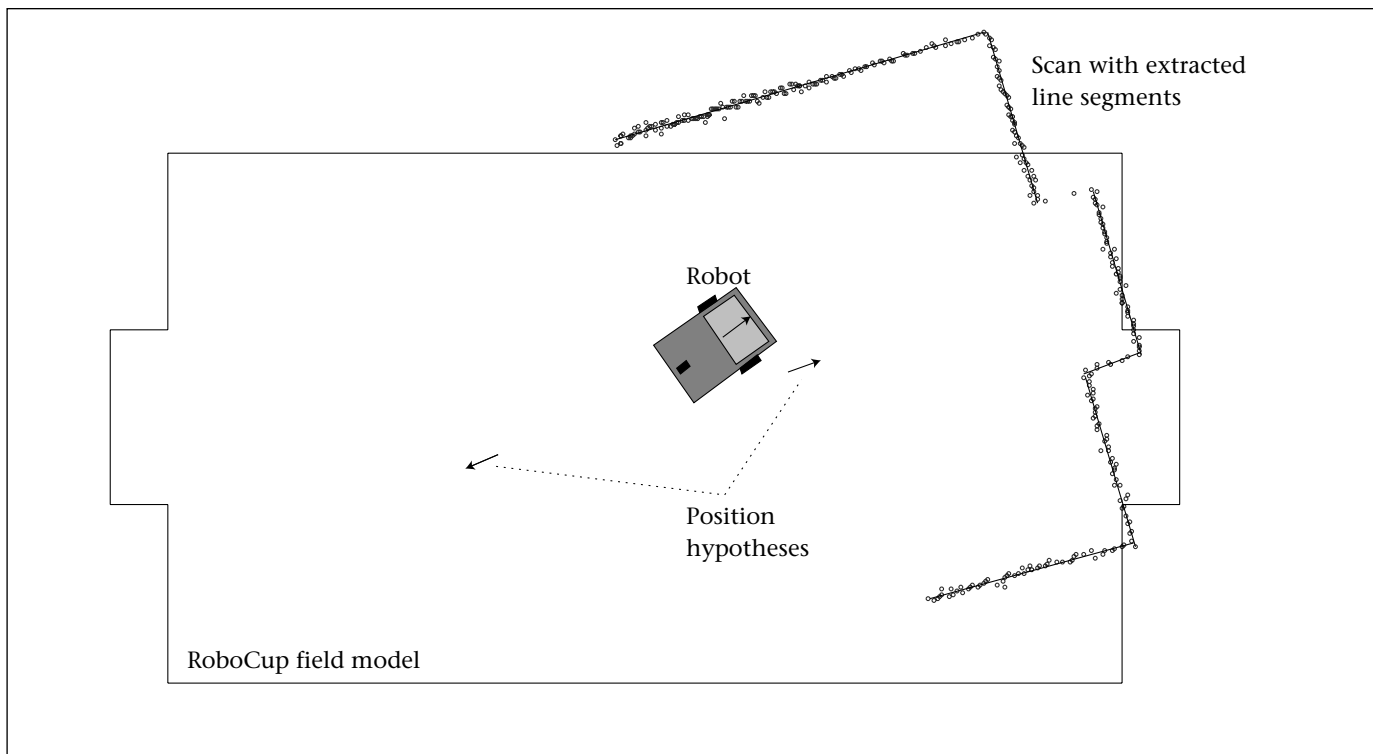


Figure 4. Scan Matches Lead to Position Hypotheses.

against an a priori model of the soccer field. To ensure that extracted lines really correspond to field-border lines, only scan lines significantly longer than the size of soccer robots are considered. Then, the correspondence problem between scan lines and lines of the a priori model is solved by backtracking over all possible pairings between scan lines and model lines—similar to the method described by Castellanos, Tardós, and Neira (1996). Successful matchings lead to position hypotheses, of which there are only two if three field borders are visible (figure 4).

After the brief sketch of the matching algorithm, one might suspect that the worst-case run time of the algorithm is exponential in the number of model lines. However, a closer inspection reveals it runs in cubic time because of geometric constraints (Weigel 1998). Moreover, we expect this algorithm to be almost linear in the number of model lines in “natural,” settings such as office environments. Our self-localization algorithm is implemented in a straightforward way (figure 5).

From a set of position hypotheses generated by the scan-matching algorithm, the most plausible one is selected and fused with the odometry position estimate using a Kalman filter. The Kalman filter returns the *optimal* estimate (the one with the smallest variance) for a given set of observations (Maybeck 1990). The

robot position is then updated, taking into account that the robot has moved since the scan was taken.

Our hardware configuration allows five laser scans a second, using only a few milliseconds for computing position hypotheses and the position update. Although a laser scan can include readings from objects blocking the sight to the field borders, we did not experience any failures in the position-estimation process. In particular, we never observed the situation that one of our robots got its orientation wrong and “changed sides.”

Building the Local World Model

After the self-localization module matched a range scan, the sensor data are interpreted to recognize other players and the ball (figure 6).

Scan points that correspond to field lines are removed, and the remaining points are clustered. For each cluster, the center of gravity is computed and interpreted as the approximate position of a robot (figure 7). Inherent to this approach is a systematic error depending on the shape of the robots.

For ball recognition, we use a commercially available vision system. If the camera sees an object of a certain color (a so-called *blob*), the vision system outputs the pixel coordinates of the center of the blob and its width, height,

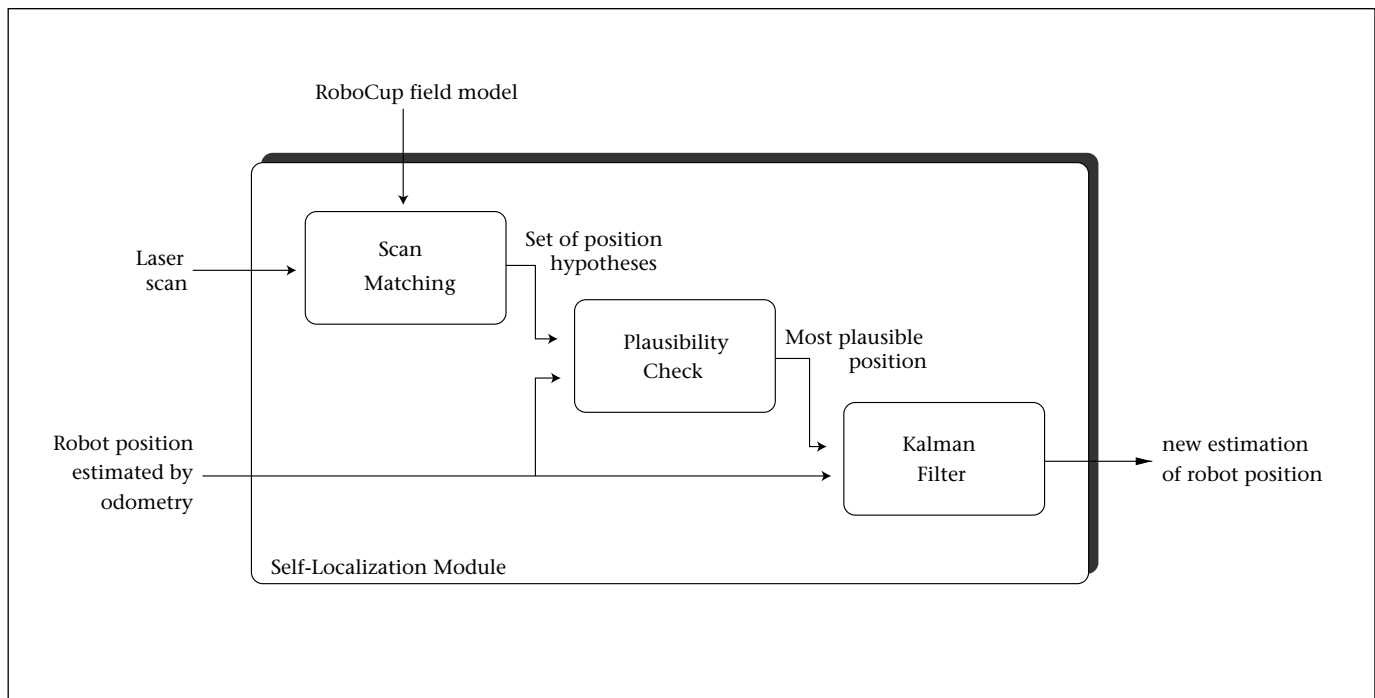


Figure 5. Self-Localization Module.

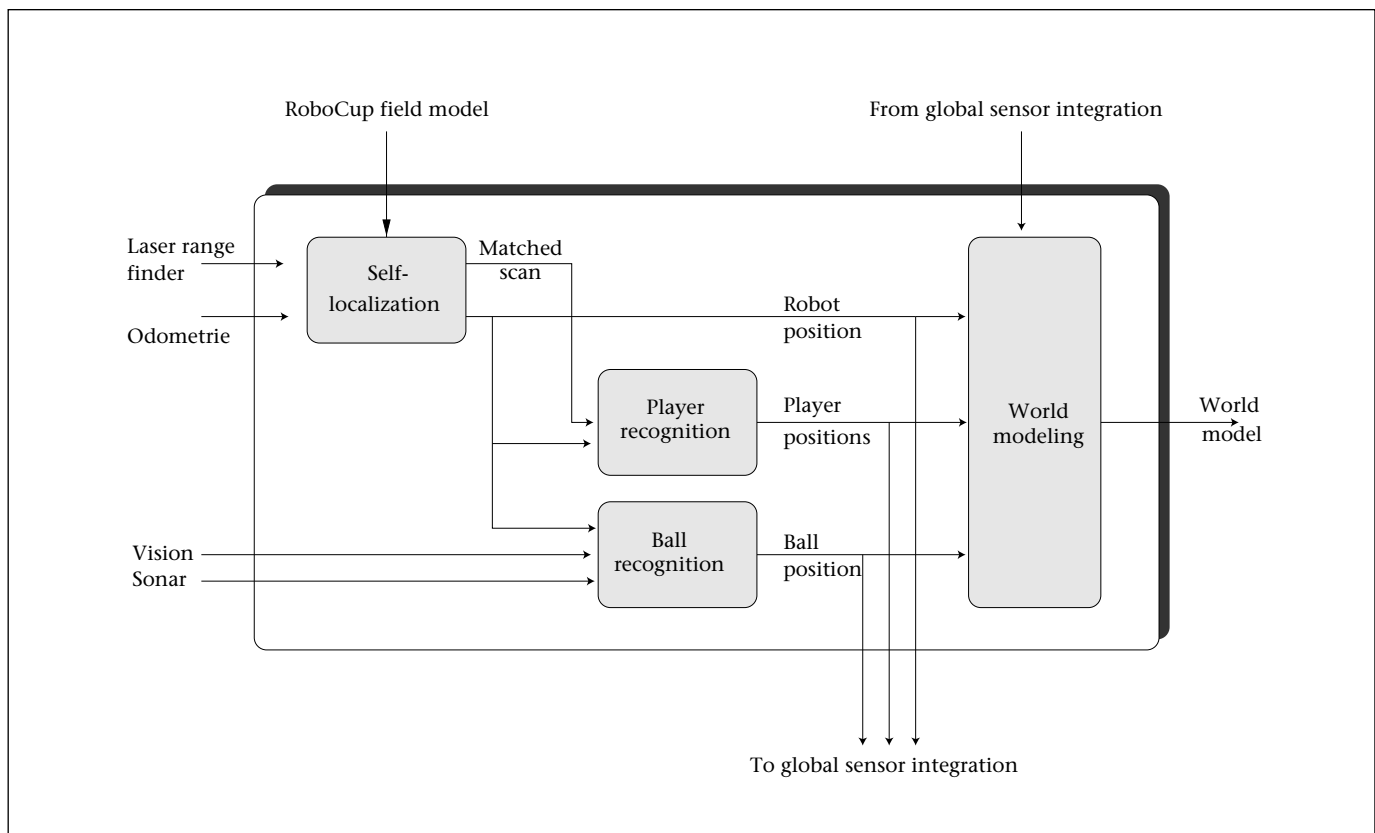


Figure 6. Perception Module.

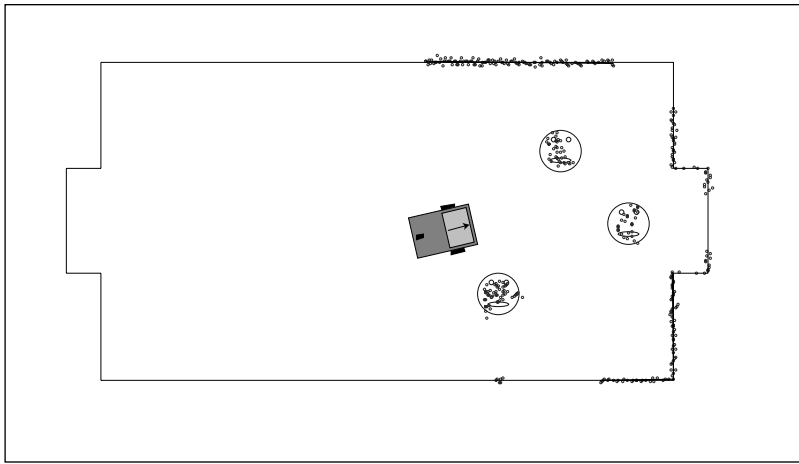


Figure 7. Line Segments Are Extracted from a Range Scan and Matched against the Field Lines, and Three Players Are Extracted from the Scan.

and area size. From these pixel coordinates, we compute the relative position of the ball with respect to the robot position by mapping pixel coordinates to distance and angle. This mapping is learned by training the correspondence between pixel coordinates and angles and distances for a set of well-chosen real-world positions and using interpolation for other pixels. To improve the quality of the position estimation, we use the sonar sensors as a secondary source of information for determining the ball position.

From the estimated position of the player, the estimated position of other objects, and the estimated position of the ball—if it is visible—the soccer agent constructs its own *local world model*. By keeping a history list of positions for all objects, their headings and velocities can be determined. To reduce noise, headings and velocities are low-pass filtered. Position, heading, and velocity estimates are sent to the *multirobot sensor integration module*.

In addition to objects that are directly observable, the local world model also contains information about objects that are not visible. First, if an object disappears temporarily from the robot's view, it is not immediately removed from the world model. Based on its last-known position and estimated heading and velocity, its most likely position is estimated for a few seconds. Second, information from the global world model is used to extend the local world model of a player.

Global World Model

The *global world model* is constructed from time-stamped position, heading, and velocity estimates that each soccer agent sends to the

global sensor-integration module. Because soccer players and balls tend to move slowly (< 1 meter a second), a simple greedy algorithm can be used to track objects. Furthermore, friends and foes can be identified by comparing sensed object positions with the positions of team members determined using the self-localization algorithm. Knowing who and where the team members are is, of course, helpful in playing a cooperative game.

Other information that is useful is the global ball position. Our vision hardware recognizes the ball only to a distance of 3 to 4 meters. Knowing the global ball position even if it is not directly visible enables the soccer robot to turn its camera into the direction of where the ball is expected, avoiding a search for the ball by turning around. This information is important in particular for the goal keeper, which might miss a ball from the left while it searches for the ball on the right.

It should be noted, however, that because of the inherent delay between sensing an object and receiving back a message from the global sensor integration, the information from the global world model is always 100 to 400 milliseconds old; thus, it cannot be used to control the robot behavior directly. However, apart from the two uses spelled out earlier, there are nevertheless a number of important problems that could be solved using this global world model—and we will work on these points in the future. First, the global world model could be used to reorient disoriented team members. Although we never experienced such a disorientation, such a fallback mechanism is certainly worthwhile. Second, it provides a way to detect unreliable sensor systems of some of the soccer agents. Third, the global world model could be used for making strategic decisions, such as changing roles dynamically (Veloso et al. 1998).

Behavior-Based Control and Multiagent Cooperation

The soccer agent's decisions are mainly based on the situation represented in the explicit world model. However, to create cooperative team behavior, actual decisions are also based on the role assigned to the particular agent and on intentions communicated by other players.

Although the control of the execution can be described as behavior based, our approach differs significantly from approaches where behaviors are activated by uninterpreted sensor input, as is the case with the ULLANTA team (Werger et al. 1998). In our case, high-level features that are derived from sensor input and

the communication with other agents determine what behavior is activated. Furthermore, behaviors can invoke significant deliberation, such as planning the path to a particular target point.

The behavior-based control module consists of a rule-based system that maps situations to actions. In the current version, only a few rules (less than 10) are needed, and all of them have been designed by hand and improved over time after gathering new experiences from playing games. Even during the competition in Paris, we refined some of them. The rules are evaluated every 100 milliseconds so that the module can react immediately to changes in the world. Depending on whether the agent fills the *role* of the goal keeper or a field player, there are different rule sets.

The goalie is simple minded and just tries to keep the ball from rolling into our goal. It always watches the ball—getting its information from the global world model if the camera cannot recognize the ball—and moves to the point where the robot expects to intercept the ball based on its heading. If the ball is on the left or right of the goal, the goalkeeper turns to face the ball. To allow for fast left and right movements, we use a special hardware setup where the “head” of the goalie is mounted to the right, as shown in figure 1. If the ball hits the goalie, the kicking device kicks it back into the field.

The field players have a much more elaborate set of skills. The first four skills concern situations where the ball cannot be played directly, and the two last skills address ball handling:

Approach-position: Approach a target position carefully.

Go-to-position: Plan and constantly replan a collision-free path from the robot’s current position to a target position and follow this path until the target position is reached. Path planning is done using the extended visibility graph method (Latombe 1991), which is fast enough to be executed in each execution cycle.

Observe-ball: Set the robot’s heading such that the ball is in the center of focus. Track the ball without approaching it.

Search-ball: Turn the robot to find the ball. If the ball is not found after one revolution, go to home position and search again from there.

Move-ball: Determine a straight line to the goal that has the largest distance to any object on the field. Follow this line at increasing velocity and redetermine the line whenever appropriate.

Shoot-ball: To accelerate the ball, either turn the robot rapidly with the ball between the flippers, or use the kicker mechanism. The

decision on which mechanism to use and in which direction to turn is made according to the current game situation.

The mapping from situations to actions is implemented in a decision tree-like manner. It should be noted that details of tactical decisions and behaviors were subject to permanent modifications even when the competition in Paris had already started. As a reaction to teams that would just push the ball and opponents over the field, we modified our behavior to not yield in such situations.

If all the soccer players would act according to the same set of rules, a *swarm behavior* would result, where the soccer players would block each other. One way to solve this problem is to assign different roles to the players and define areas of competence for these roles (figure 8).

If these areas were nonoverlapping, interference between team members would not happen, even without any communication between players. Each player would go to the ball and pass it on to the next area of competence or into the goal. In fact, this was our initial design, and it is still the fallback strategy when radio communication is not working.

There are numerous problems with such a rigid assignment of competence areas, however. First, players can interfere at the border lines between competence areas. Second, if a player is blocked by the other team, broken, or removed from the field, no player will handle balls in the corresponding area. Third, if a defender has the chance to dribble the ball to the opponent’s goal, the corresponding forward will most probably block this run. For these reasons, we modified our initial design significantly. Even during the tournament in Paris, we changed the areas of competence and added other means to coordinate attacks as a reaction to our experiences from the games.

If a player is in a good position to play the ball, it sends a clear-out message. As a reaction to receiving such a message, other players try to keep out of the playing robot’s way (figure 9), helping to avoid situations in which two teammates block each other. In other words, we also rely on cooperation by communication, as the *UTTORI* team did (Yokota et al. 1999). However, our communication scheme is much less elaborate than *UTTORI*’s. Based on communicating intentions, areas of competence can be made overlapping, as shown in figure 8. Now, the forwards handle three-quarters of the field, and attacks are coordinated by exchanging the intentions.

We do not have any special coordination for defensive moves. In fact, defensive behavior *emerges* from the behavior-based control

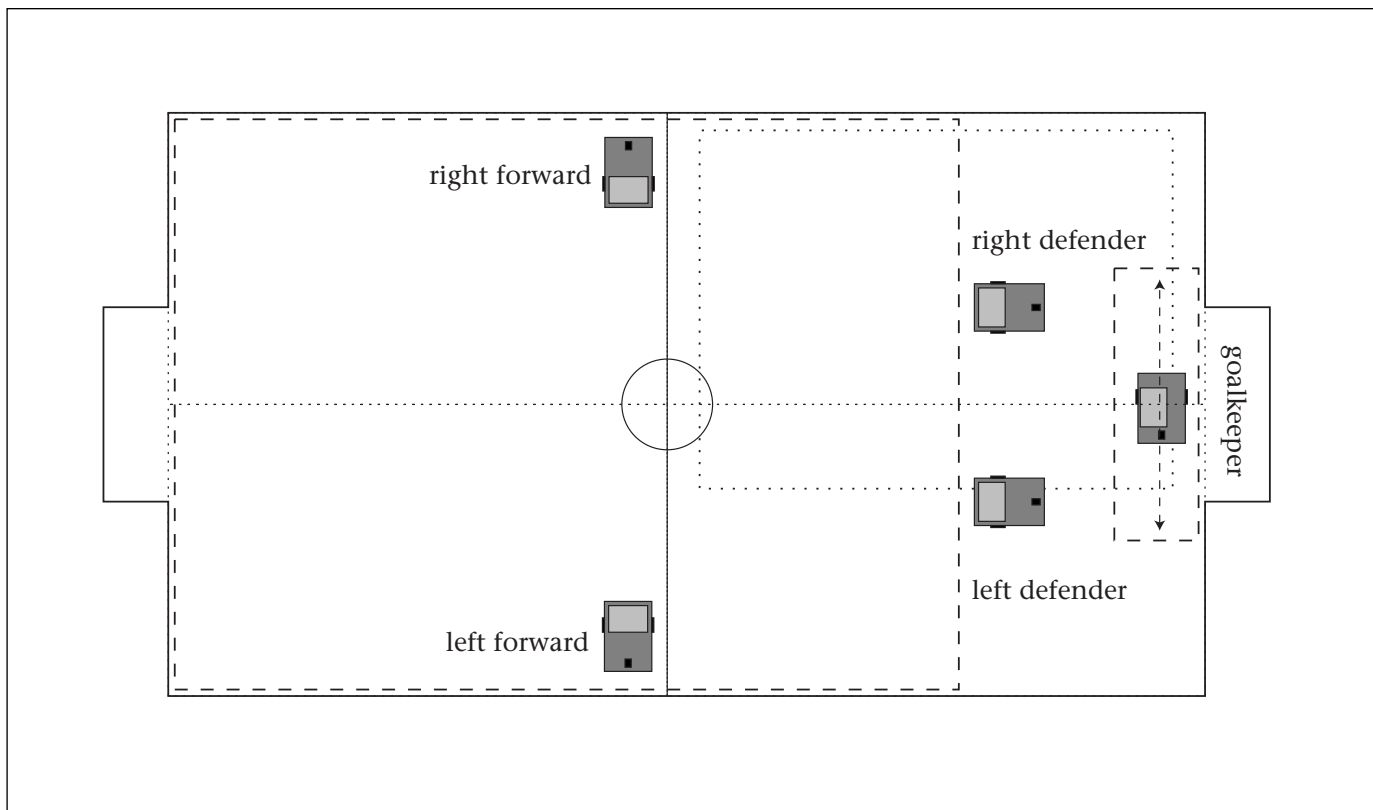


Figure 8. Roles and Areas of Competence.

described earlier. When the ball enters our half of the field, our defenders go to the ball and block the attack. Surprisingly, this simple defensive strategy worked quite successfully.

Conclusion and Discussion

Participating in the RoboCup-98 tournament was beneficial for us in two ways. First, we got the opportunity to exchange ideas with other teams and learned how they approached the problems. Second, we learned much from playing. As pointed out at various places in the article, we redesigned tactics and strategy during the tournament, incorporating the experience we attained during the games.

The performance of our team at RoboCup-98 was quite satisfying. Apart from winning the tournament, we also had the best goal difference (12:1), never lost a game, and scored almost 25 percent of the goals during the tournament. This performance was not accidental, as demonstrated at the national German competition VISION-RoboCup-98 on 30 September to 1 October 1998 in Stuttgart. Again, we won the tournament and did not lose any game. (**Endnote:** In 1999, our team again won the German VISION-RoboCup

and came in third at RoboCup in Stockholm.)

The key components for this success are most probably the self-localization and object-recognition techniques based on laser range finders, which enabled us to create accurate and reliable local and global world models. Based on these world models, we were able to implement reactive path planning, fine-tuned behaviors, and basic multiagent cooperation, which was instrumental in winning. Finally, our kicker and the ball-steering mechanism certainly also had a role in playing successful robotic soccer.

Acknowledgments

This work has been partially supported by Deutsche Forschungsgemeinschaft as part of the graduate school on Human and Machine Intelligence; Medien- und Filmgesellschaft Baden-Württemberg mbH; and SICK AG, which provided the laser range finders. Furthermore, we would like to thank ActivMedia and Newton Labs for their timely support, resolving some of the problems that occurred just a few weeks before RoboCup-98.

References

Asada, M., and Kitano, H., eds. 1999. *RoboCup-98: Robot Soccer World Cup II*. Lecture Notes in Artificial

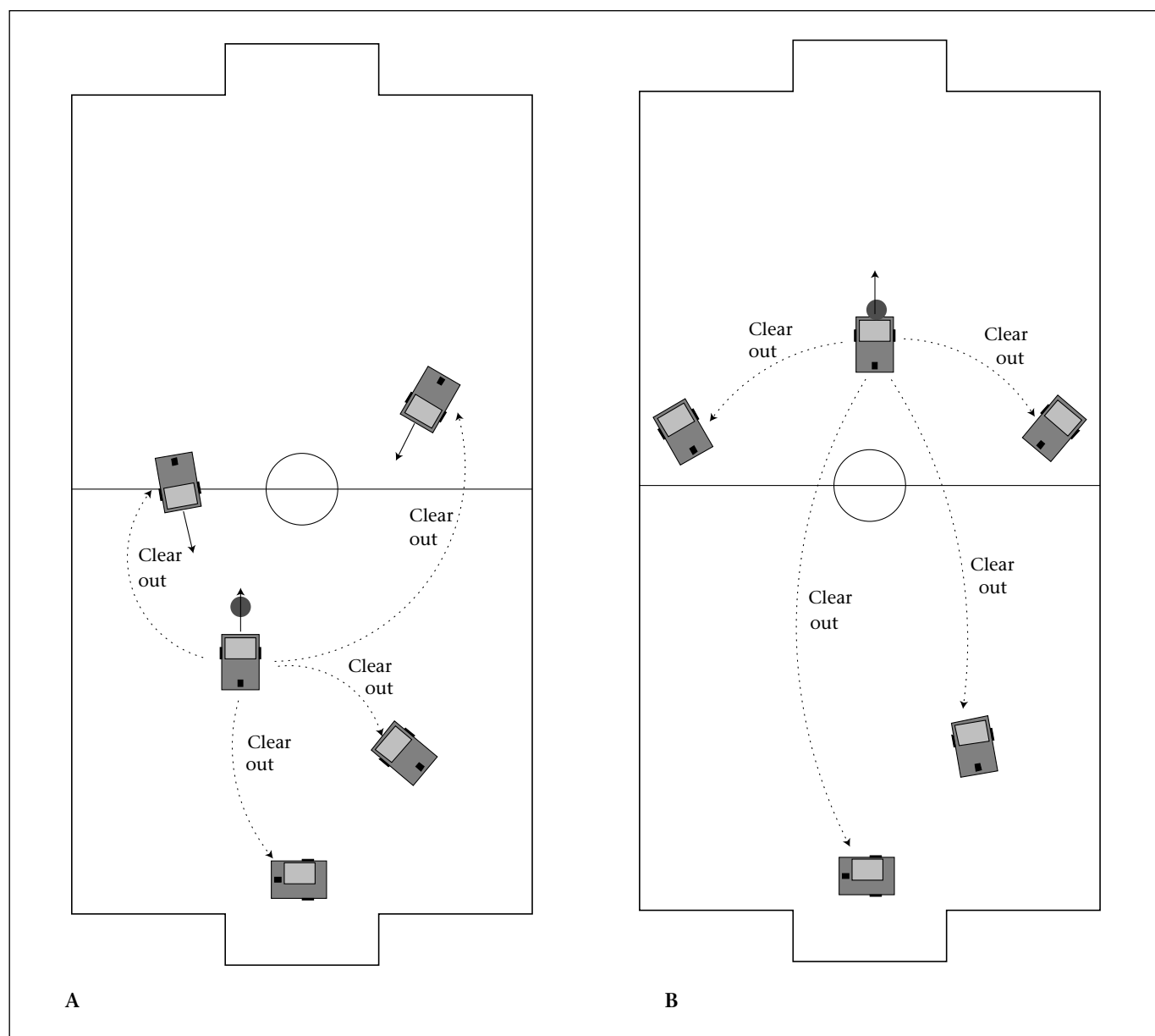


Figure 9. *Cooperation by Communication.*
 A. Player gets ball and notifies teammates. B. Player can run.

Intelligence. New York: Springer-Verlag.

Burkhard, H. D.; Hannebauer, M.; and Wendler, J. 1998. AT HUMBOLDT—Development, Practice, and Theory. In *RoboCup-97: Robot Soccer World Cup I*, ed. H. Kitano, 357–372. Lecture Notes in Artificial Intelligence, Volume 1395. New York: Springer-Verlag.

Castellanos, J. A.; Tardós, J. D.; and Neira, J. 1996. Constraint-Based Mobile Robot Localization. Paper presented at the International Workshop on Advanced Robotics and Intelligent Machines, 2–3 April, Manchester, United Kingdom.

Cox, I. J. 1990. BLANCHE: Position Estimation for an Autonomous Robot Vehicle. In *Autonomous Robot Vehicles*, eds. I. J. Cox and G. T. Wilfong, 221–228.

New York: Springer-Verlag.

Gutmann, J.-S., and Nebel, B. 1997. Navigation Mobiler Roboter mit Laserscans. In *Autonome Mobile System 1997*, eds. P. Levi, Th. Bräunl, and N. Oswald, 36–47. Informatik aktuell (Navigation of Mobile Robots Based on Laser Scans). Stuttgart: Springer-Verlag.

Gutmann, J.-S., and Schlegel, C. 1996. AMOS: Comparison of Scan-Matching Approaches for Self-Localization in Indoor Environments. In *Proceedings of the First Euromicro Workshop on Advanced Mobile Robots*, 61–67. Washington, D.C.: IEEE Computer Society.

Gutmann, J.-S.; Burgard, W.; Fox, D.; and Konolige,

K. 1998. An Experimental Comparison of Localization Methods. In *Proceedings of the International Conference on Intelligent Robots and Systems (IROS'98)*, 736–743. Washington, D.C.: IEEE Computer Society.

Kitano, H.; Asada, M.; Kuniyoshi, Y.; Noda, I.; Osawa, E.; and Matsubara, H. 1997. RoboCup: A Challenge Problem for AI. *AI Magazine* 18(1): 73–85.

Konolige, K.; Myers, K.; Ruspini, E. H.; and Saffiotti, A. 1997. The SAPHIRA Architecture: A Design for Autonomy. *Journal of Experimental and Theoretical Artificial Intelligence* 9(1): 215–235.

Latombe, J.-C. 1991. *Robot Motion Planning*. Dordrecht, The Netherlands: Kluwer.

Lu, F., and Milios, E. E. 1994. Robot Pose Estimation in Unknown Environments by Matching 2D Range Scans. In *IEEE Computer Vision and Pattern Recognition Conference (CVPR)*, 935–938. Washington, D.C.: IEEE Computer Society.

Maybeck, P. S. 1990. The Kalman Filter: An Introduction to Concepts. In *Autonomous Robot Vehicles*, eds. I. J. Cox and G. T. Wilfong, 194–204. New York: Springer-Verlag.

Suzuki, S.; Takahashi, Y.; Uchibe, E.; Nakamura, M.; Mishima, C.; Ishizuka, H.; Kato, T.; and Asada, M. 1998. Vision-Based Robot Learning toward RoboCup: Osaka University "TRACKIES." In *RoboCup-97: Robot Soccer World Cup I*, ed. H. Kitano, 305–319. Lecture Notes in Artificial Intelligence, Volume 1395. New York: Springer-Verlag.

Veloso, M.; Stone, P.; Han, K.; and Achim, S. 1998. The CMUNITED-97 Small Robot Team. In *RoboCup-97: Robot Soccer World Cup I*, ed. H. Kitano, 242–256. Lecture Notes in Artificial Intelligence, Volume 1395. New York: Springer-Verlag.

Weigel, T. 1998. *Roboter-Fußball: Selbstlokalisierung, Pfadplanung und Basisfähigkeiten (Robotic Soccer: Self-Localization, Path Planning, and Basic Skills)*. Diplomarbeit, Fakultät für Angewandte Wissenschaften, University of Freiburg, Germany.

Weiß, G., and von Puttkamer, E. 1995. A Map Based on Laserscans without Geometric Interpretation. In *Intelligent Autonomous Systems (IAS-4)*, eds. U. Rembold, R. Dillmann, L. O. Hertzberger, and T. Kanade, 403–407. Dordrecht, The Netherlands: IOS.

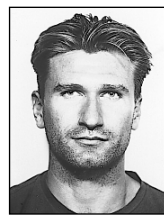
Wenger, B.; Funes, P.; Schneider Fontan, M.; Sargent, R.; Witty, C.; and Witty, T. 1998. THE SPIRIT OF BOLIVIA: Complex Behavior through Minimal Control. In *RoboCup-97: Robot Soccer World Cup I*, ed. H. Kitano, 384–356. Lecture Notes in Artificial Intelligence, Volume 1395. New York: Springer-Verlag.

Yokota, K.; Ozaki, K.; Watanabe, N.; Matsumoto, A.; Koyama, D.; Ishikawa, T.;

Kawabata, K.; Kaetsu, H.; and Asama, H. 1999. Cooperative Team Play Based on Communication. In *RoboCup-98: Robot Soccer World Cup II*, eds. M. Asada and H. Kitano, 333–347. Lecture Notes in Artificial Intelligence. New York: Springer-Verlag.



Jens-Steffen Gutmann is a Ph.D. student in the graduate school of Human and Machine Intelligence at the University of Freiburg. He received his diploma in computer science from the University of Ulm in 1996. His research interests are robust mobile robot navigation and robotic soccer playing. His e-mail address is gutmann@ieee.org.



Wolfgang Hatzack received his diploma in computer science (Diplom-Informatiker) from the University of Bonn in 1996. Currently, he is a Ph.D. student in the Artificial Intelligence Research Group at the University of Freiburg, focusing on air-traffic control and robotics.



Immanuel Herrmann is currently working on his Master's thesis in mathematics at the University of Freiburg. He is also a graduate student in computer science. He participated successfully in different contests such as the Association of Computing Machinery International Collegiate Programming Contest.



Bernhard Nebel is a professor in the Department of Computer Science at Albert-Ludwigs-University Freiburg, Germany, and director of the AI Research Lab. He received a Ph.D. (Dr. rer. nat) from the University of Saarland in 1989 for his work on knowledge representation and belief revision. He is a member of the International Joint Conferences on Artificial Intelligence, Inc., board of trustees and a member of the graduate school of Human and Machine Intelligence at Albert-Ludwigs-University. Among other professional services, he served as the program cochair for the Third International Conference on Principles of

Knowledge Representation and Reasoning (KR'92) and will serve as the program chair for IJCAI'01. In addition, he is a member of the editorial boards of *Artificial Intelligence* and *AI Communication* and an associate editor of the *Journal of Artificial Intelligence Research*. His research interests include knowledge representation and reasoning, with an emphasis on semantics, algorithms, and computational complexity, in particular in the areas of description logics, temporal and spatial reasoning, constraint-based reasoning, planning, belief revision, and robotics. His e-mail address is nebel@informatik.uni-freiburg.de.



Frank Rittinger is currently a graduate student at the University of Freiburg, Germany. Before joining the Artificial Intelligence Group in Freiburg, he studied for one year at the University of Sussex at Brighton, Great Britain. His e-mail address is frittinger@acm.org.



Augustinus Topor is a graduate student at the University of Freiburg, Germany. He is currently working on his thesis on path planning in dynamic environments.



Thile Weigel received his diploma in computer science in 1999 for his thesis on self-localization, path planning, and reactive control in the RoboCup context. His e-mail address is weigel@informatik.uni-freiburg.de.