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, Thilo Weigel

Albert-Ludwigs-Universität Freiburg, Institut für Informatik

Am Flughafen 17, D-79110 Freiburg, Germany

⟨*last name*⟩@informatik.uni-freiburg.de

June 4, 1999

Abstract

Robotic soccer is an ideal task to demonstrate new techniques and to explore new problems. Moreover, problems and solutions can be easily communicated because soccer is a well-known game. Our intention in building a robotic soccer team and participating in RoboCup'98 was, first of all, to demonstrate the usefulness of the self-localization methods we have developed. Secondly, we wanted to show that playing soccer based on an explicit world model is much more effective than other methods. Thirdly, 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 on the first two points. Moreover, robotic soccer gave us the opportunity to study problems in distributed, cooperative sensing.

1 Introduction

Robotic soccer is an interesting research domain because problems in robotics, artificial intelligence, multi-agent systems, and real-time reasoning have to be solved in order 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 to explore new problems.

We started to design a robotic soccer team with the intention to participate in RoboCup'98 for three reasons. First of all, 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 modelling* and *accurate and robust self-localization* possible.

Secondly, 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. While it is possible to play robotic soccer by reacting on mostly uniterpreted sensor inputs as in a pure behavior-based [Werger *et al.*, 1998] or reinforcemnt 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 inputs. 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 *et al.*, 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.

Thirdly, we intended to address the problem of multirobot sensor integration in order to build a *global world model* and to 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 interesting topic for future research.

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

The rest of the paper is structured as follows. In the next section, we give a brief sketch of the robot hardware. Section 3 describes the general architecture of our soccer players and the soccer team. Section 4 focuses on our self-localization approach and Section 5 describes our player and ball recognition methods that are needed to create the local world model. The integration of these world models into a global model and the problems we encountered are described in Section 6. In Section 7 we sketch the behavior-based control of the soccer agents and show how a basic form of multi-agent cooperation is achieved. Finally, in Section 8 we describe our experience of participating in RoboCup'98 and conclude.

2 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 effectory skills. For this reason, we had to add a number of hardware components (see Fig. 1).

Figure 1: Three of our five robots: Two field players and the goal keeper

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 the above components, we added *PLS200* laser range-finders manufactured by *SICK AG* to all of our robots. These range finders can give depth information for a 180° field of view with an angular resolution of 0.5° and an accuracy of 5 cm up to a distance of 30 m.

Handling the ball with the body of the Pioneer 1 robot is not a very 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, in order to steer the ball we used flexible flippers that have a length of approximately 35 % 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 aethetically. 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 very annoying for everybody – in particular for spectators. Furthermore, without the ball steering mechanism the ball is very easily lost when running with the ball.

3 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 (see Fig. 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.

In order to initialize the soccer agents, to start and to stop the robots, and in order to monitor the state of all agents, we use a radio ethernet connection between the on-board computers and an off-field computer (see Fig. 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

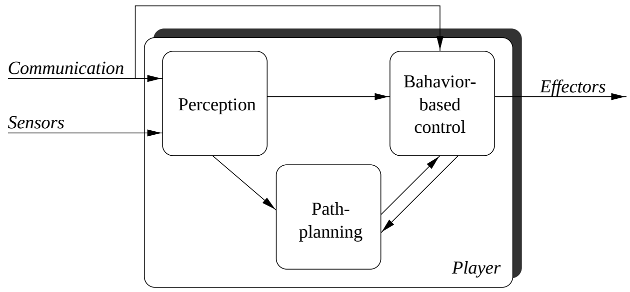

Figure 2: Player Architecture

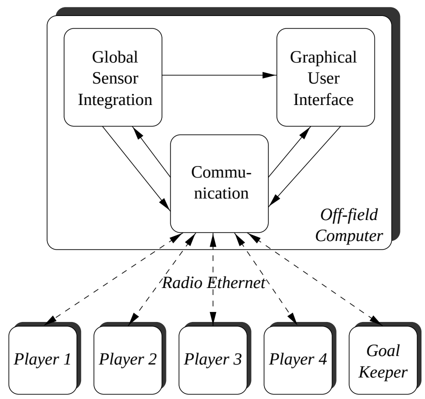

Figure 3: Team Architecture

league used a similar 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 employ this information to extend their own local view of the world. This means that the world model our players have is very 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].

4 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 employ such a method as one of the key components in our soccer agents.

There exist a number of different self-localization methods based on laser scans [Cox, 1990; Gutmann and Schlegel, 1996; Lu and Milios, 1994; Weiß and von Puttkamer, 1995]. However, these methods are only *local*, i.e., they can only be used to correct an already existing position estimation. This means that once a robot loses its position, it will be completely lost. Furthermore, all the methods are computationally very demanding, needing 100 msecs up to a few seconds on a modern computer. Global methods are even more costly from a computational point of view. For these reasons we designed a new self-localization method which 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 against an *a priori* model of the soccer field. In order to ensure that extracted lines really correspond to field-border lines, only scan lines significantly longer than the extend 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 *et al.* [1996]. Successful matchings lead to position hyptheses, of which there are only two if three field borders are visible (see Fig. 4).

After the brief sketch of the matching algorithm, one might suspect that the worst-case runtime 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 linearly in the number of model lines in "natural" settings such as office environments.

Our self-localization algorithm is implemented in a straight forward way (see Fig. 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 by using a Kalman filter. The Kalman filter returns the optimal estimate (the one with the smallest variance) for a given

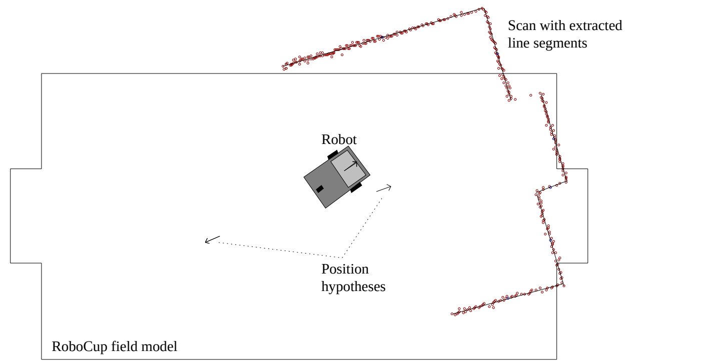

Figure 4: Scan matches lead to position hypotheses

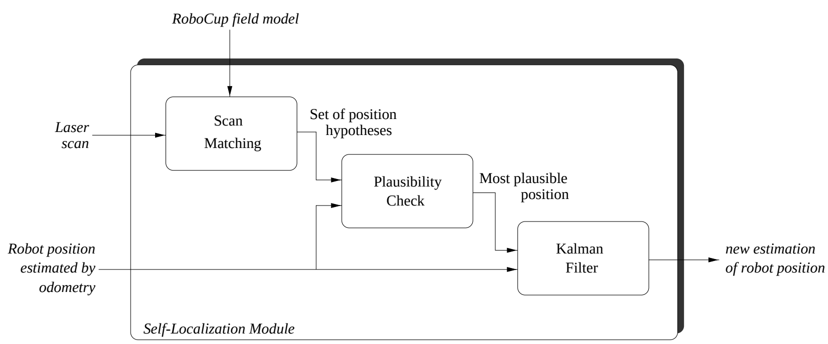

Figure 5: Scan matches lead to position hypotheses

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 per second using only a few milliseconds for computing position hypotheses and the position update. Although a laser scan may 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."

5 Building the Local World Model

After the self-localization module matched a range scan, the sonsor data is interpreted in order to recognize other players and the ball (see Fig. 6). Scan

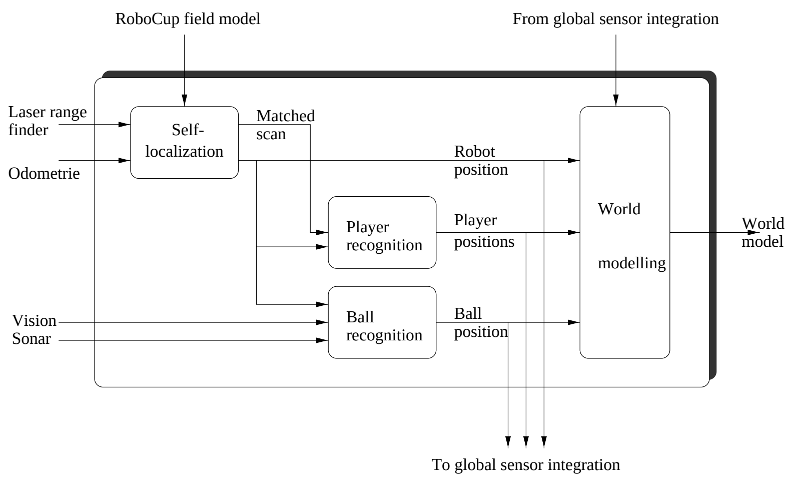

Figure 6: Line segments are extracted from a range scan, matched against the field lines and three players are extracted from the scan.

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 (see Fig. 7). Inherent to this approach is a systematic error depending on the shape of the robots.

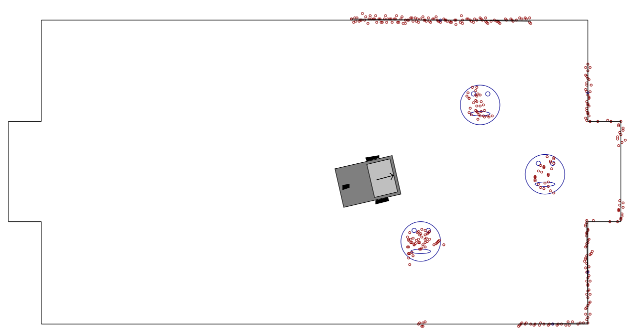

Figure 7: Line segments are extracted from a range scan, matched against the field lines and three players are extracted from the scan.

For ball reconition, 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, its width, height 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. In order 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 of all, if an object disappears temporarily from the robot's view, it is not immediately removed from the world model. Using its last known position and estimated heading and velocity, its most likely position is estimated for a few seconds. Secondly, information from the global world model is used to extend the local world model of a player.

6 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 ($< 1m/sec$), 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, very helpful in playing a cooperative game.

Other information that is very useful is the global ball position. Our vision hardware recognizes the ball only up to a distance of 3–4 m. 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 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 side.

It should be noted, however, that due to 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 100–400 msecs old. This means that it cannot be used to control the robot behavior directly. However, apart from the two uses spelled out above, 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. Firstly, the global world model could be used to reorient disoriented team members. Although we never experienced such a disorientation, such a fall-back mechanism is certainly worthwhile. Secondly, it provides a way to detect unreliable sensor systems of some of the soccer agents. Thirdly, the global world model could be used for making strategic decisions, such as changing roles dynamically [Veloso *et al.*, 1998].

7 Behavior-based Control and Multi-Agent Cooperation

The soccer agent's decisions are mainly based on the situation represented in the explicit world model. However, in order 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 inputs as is the case in the *Ullanta* team [Werger *et al.*, 1998]. In our case, high-level features that are derived from sensor inputs and from the communication with other agents determine what behavior is activated. Furthermore, behaviors may 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 competetion in Paris we refined some of them. The rules are evaluated every 100 msecs 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 of a field player, there are different rule sets.

The goalie is very 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 goal keeper turns to face the ball. In order 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 Fig. 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 below concern situations when the ball cannot be played directly, while the two last skills address ball handling:

Approach-position: Approach a target position carefully.

Go-to-position: Plan and constantly re-plan 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 robots heading such that the ball is in the center of focus. Track the ball without approaching it.

Search-ball: Turn the robot in order 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 which 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 which would just push the ball and opponents over the field we modified our behavior to not yield in such situations.

If all of the soccer player 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 to define *areas of competence* for these roles (see Fig. 8). If these

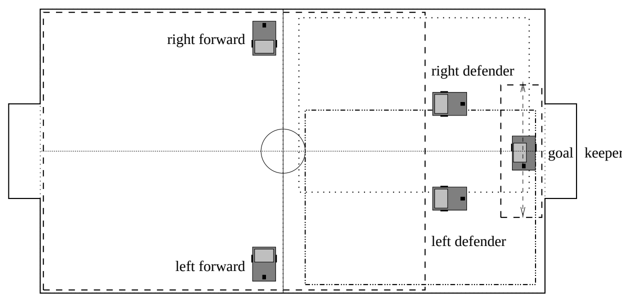

Figure 8: Roles and areas of competence

areas would be non-overlapping, interference between team members should 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 fall-back strategy when the radio communication is not working.

There are numerous problems with such a rigid assignment of competence areas, however. Firstly, players may interfere at the border lines between competence areas. Secondly, if a player is blocked by the other team, broken, or removed from the field, no player will handle balls in the corresponding area. Thirdly, if a defender has the chance of dribbling 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 robots way (see Fig. 9). This helps to avoid situations in which two team mates block each other. In other words, we also rely on *cooperation by communication* as the *Uttori* team [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 Fig. 8. Now, the forwards handle three quarters of the field and attacks are coordinated by exchanging the intentions.

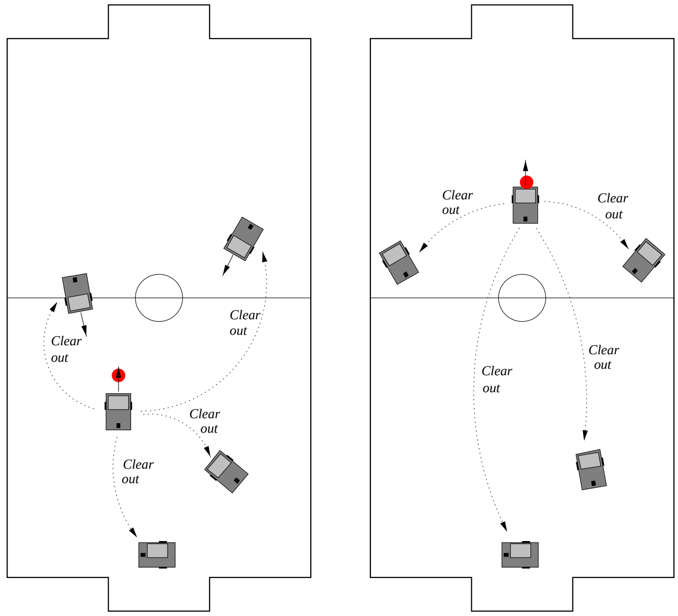

Figure 9: Cooperation by communication

We do not have any special coordination for defensive moves. In fact, defensive behavior *emerges* from the behavior-based control described above. When the ball enters our half of the field, our defenders go to the ball and by that block the attack. Surprisingly, this simple defensive strategy worked quite successfully.

8 Conclusion and Discussion

Participating in the RoboCup'98 tournament was very beneficial for us in two ways. Firstly, we got the opportunity to exchange ideas with other teams and learned how they approached the problems. Secondly, we learned much from playing. As pointed out at various places in the paper, we redesigned tactics and strategy during the tournament incorporating the experience we made 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% of the goals during the tournament. This performance was not accidental as demonstrated at the national German competition *VISION-RoboCup'98* on 30th of September and 1st of October 1998 in Stuttgart. Again, we won the tournament and did not lose any game.

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 multi-agent cooperation – which was instrumental in winning. Finally, our kicker and the ball steering mechanism certainly also played a role in playing successful robotic soccer.

Acknowledgements

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

References

[Asada and Kitano, 1999] M. Asada and H. Kitano, editors. *RoboCup-98: Robot Soccer World Cup II.* Lecture Notes in Artificial Intelligence. Springer-Verlag, Berlin, Heidelberg, New York, 1999. To appear.

[Burkhard *et al.*, 1998] Hans-Dieter Burkhard, Markus Hannebauer, and Jan Wendler. AT Humbold – development, practice and theory. In Kitano [1998], pages 357–372.

[Castellanos *et al.*, 1996] José A. Castellanos, Juan D. Tardós, and José Neira. Constraint-based mobile robot localization. In *International Workshop on Advanced Robotics and Intelligent Machines*. University of Salford, April 1996.

[Cox, 1990] Ingemar J. Cox. Blanche: Position estimation for an autonomous robot vehilce. In Cox and Wilfong [1990], pages 221–228.

[Cox and Wilfong, 1990] I.J. Cox and G.T. Wilfong, editors. *Autonomous Robot Vehicles.* Springer-Verlag, Berlin, Heidelberg, New York, 1990.

[Gutmann and Nebel, 1997] Jens-Steffen Gutmann and Bernhard Nebel. Navigation mobiler Roboter mit Laserscans. In P. Levi, Th. Bräunl, and N. Oswald, editors, *Autonome Mobile System 1997*, Informatik aktuell, pages 36–47, Stuttgart, Germany, 1997. Springer-Verlag.

[Gutmann and Schlegel, 1996] Jens-Steffen Gutmann and Christian Schlegel. Amos: Comparison of scan matching approaches for self-localization in indoor environments. In *Proceedings of the 1st Euromicro Workshop on Advanced Mobile Robots*, pages 61–67. IEEE, 1996.

[Gutmann *et al.*, 1998] Jens-Steffen Gutmann, Wolfram Burgard, Dieter Fox, and Kurt Konolige. An experimental comparison of localization methods. In *Proceedings of the International Conference on Intelligent Robots and Systems (IROS'98).* IEEE/RSJ, 1998.

[Kitano *et al.*, 1997] Hiroaki Kitano, Minoru Asada, Yasuo Kuniyoshi, Itsuki Noda, Eiichi Osawa, and Hitoshi Matsubara. RoboCup: A challenge problem for AI. *The AI Magazine*, 18(1):73–85, 1997.

[Kitano, 1998] H. Kitano, editor. *RoboCup-97: Robot Soccer World Cup I*, volume 1395 of *Lecture Notes in Artificial Intelligence.* Springer-Verlag, Berlin, Heidelberg, New York, 1998.

[Konolige *et al.*, 1997] Kurt Konolige, Karen Myers, Enrique H. Ruspini, and Alessandro Saffiotti. The Saphira Architecture: A Design for Autonomy. *Journal of Experimental and Theoretical Artificial Intelligence*, 9:215–235, 1997.

[Latombe, 1991] Jean-Claude Latombe. *Robot Motion Planning.* Kluwer, Dordrecht, Holland, 1991.

[Lu and Milios, 1994] Feng Lu and Evangelos E. Milios. Robot pose estimation in unknown environments by matching 2D range scans. In *IEEE Computer Vision and Pattern recognition Conference (CVPR)*, pages 935–938, 1994.

[Maybeck, 1990] Peter S. Maybeck. The Kalman filter: An introduction to concepts. In Cox and Wilfong [1990], pages 194–204.

[Suzuki *et al.*, 1998] S. Suzuki, Y. Takahashi, E. Uchibe, M. Nakamuro, C. Mishima, H. Ishizuka, T. Kato, and M. Asada. Vision-based robot learning towards RoboCup: Osaka University "Trackies". In Kitano [1998], pages 305–319.

[Veloso *et al.*, 1998] Manuela Veloso, Peter Stone, Kwun Han, and Sorin Achim. The CMUnited-97 small robot team. In Kitano [1998], pages 242–256.

[Weigel, 1998] Thilo Weigel. Roboter-Fußball: Selbstlokalisation, Pfadplanung und Basisfähigkeiten. Diplomarbeit, Fakultät für Angewandte Wissenschaften, Freiburg, Germany, 1998. In preparation.

[Weiß and von Puttkamer, 1995] Gerhard Weiß and Ewald von Puttkamer. A map based on laserscans without geometric interpretation. In U. Rembold, R. Dillmann, L.O. Hertzberger, and T. Kanade, editors, *Intelligent Autonomous Systems (IAS-4)*, pages 403–407. IOS Press, 1995.

[Werger *et al.*, 1998] Brian Werger, Pablo Funes, Miguel Schneider Fontan, Randy Sargent, Carl Witty, and Tim Witty. The spirit of Bolivia: Complex behavior through minimal control. In Kitano [1998], pages 348–356.

[Yokota *et al.*, 1999] K. Yokota, K. Ozaki, N. Watanabe, A. Matsumoto, D. Koyama, T. Ishikawa, K. Kawabata, H. Kaetsu, and H. Asama. Cooperative team play based on communication. In Asada and Kitano [1999]. To appear.