Cooperative Probabilistic State Estimation for Vision-Based Autonomous Soccer Robots

Thorsten Schmitt, Robert Hanek, Sebastian Buck, and Michael Beetz

TU München, Institut für Informatik, Arcisstrasse 21, 80290 München, Germany
{schmittt,hanek,buck,beetzm}@in.tum.de, http://www9.in.tum.de/agilo

Abstract. With the services that autonomous robots are to provide becoming more demanding, the states that the robots have to estimate become more complex. In this paper, we develop and analyze a probabilistic, vision-based state estimation method for individual, autonomous robots. This method enables a team of mobile robots to estimate their joint positions in a known environment and track the positions of autonomously moving objects. The state estimators of different robots cooperate to increase the accuracy and reliability of the estimation process. This cooperation between the robots enables them to track temporarily occluded objects and to faster recover their position after they have lost track of it. The method is empirically validated based on experiments with a team of physical robots.

1 Introduction

Autonomous robots must have information about themselves and their environments that is sufficient and accurate enough for the robots to complete their tasks competently. Contrary to these needs, the information that robots receive through their sensors is inherently uncertain: typically the robots' sensors can only access parts of their environments and their sensor measurements are inaccurate and noisy. In addition, control over their actuators is also inaccurate and unreliable. Finally, the dynamics of many environments cannot be accurately modeled and sometimes environments change nondeterministically.

Recent longterm experiments with autonomous robots [13] have shown that an impressively high level of reliability and autonomy can be reached by explicitly representing and maintaining the uncertainty inherent in the available information. One particularly promising method for accomplishing this is *probabilistic state estimation*. Probabilistic state estimation modules maintain the probability densities for the states of objects over time. The probability density of an object's state conditioned on the sensor measurements received so far contains all the information which is available about an object that is available to a robot. Based on these densities, robots are not only able to determine the most likely state of the objects, but can also derive even more meaningful statistics such as the variance of the current estimate.

Successful state estimation systems have been implemented for a variety of tasks including the estimation of the robot's position in a known environment,

A. Birk, S. Coradeschi, and S. Tadokoro (Eds.): RoboCup 2001, LNAI 2377, pp. 193–203, 2002.

the automatic learning of environment maps, the state estimation for objects with dynamic states (such as doors), for the tracking of people locations, and gesture recognition [12]. With the services that autonomous robots are to provide becoming more demanding, the states that the robots have to estimate become more complex. Robotic soccer provides a good case in point. In robot soccer (mid-size league) two teams of four autonomous robots play soccer against each other. A probabilistic state estimator for competent robotic soccer players should provide the action selection routines with estimates of the positions and may be even the dynamic states of each player and the ball.

This estimation problem confronts probabilistic state estimation methods with a unique combination of difficult challenges. The state is to be estimated by multiple mobile sensors with uncertain positions, the soccer field is only partly accessible for each sensor due to occlusion caused by other robots, the robots change their direction and speed very abruptly, and the models of the dynamic states of the robots of the other team are very crude and uncertain.

In this paper, we describe a state estimation module for individual, autonomous robots that enables a team of robots to estimate their joint positions in a known environment and track the positions of autonomously moving objects. The state estimation modules of different robots cooperate to increase the accuracy and reliability of the estimation process. In particular, the cooperation between the robots enables them to track temporarily occluded objects and to faster recover their position after they have lost track of it.

The state estimation module of a single robot is decomposed into subcomponents for self-localization and for tracking different kinds of objects. This decomposition reduces the overall complexity of the state estimation process and enables the robots to exploit the structures and assumptions underlying the different subtasks of the complete estimation task. Accuracy and reliability is further increased through the cooperation of these subcomponents. In this cooperation the estimated state of one subcomponent is used as evidence by the other subcomponents.

The main contributions of this paper are the following ones. First, we show that image-based probabilistic estimation of complex environment states is feasible in real time even in complex and fast changing environments. Second, we show that maintaining trees of possible tracks is particularly useful for estimating a global state based on multiple mobile sensors with position uncertainty. Third, we show how the state estimation modules of individual robots can cooperate in order to produce more accurate and reliable state estimation.

In the remainder of the paper we proceed as follows. Section 2 describes the software architecture of the state estimation module and sketches the interactions among its components. Section 3 provides a detailed description of the individual state estimators. We conclude with our experimental results and a discussion of related work.

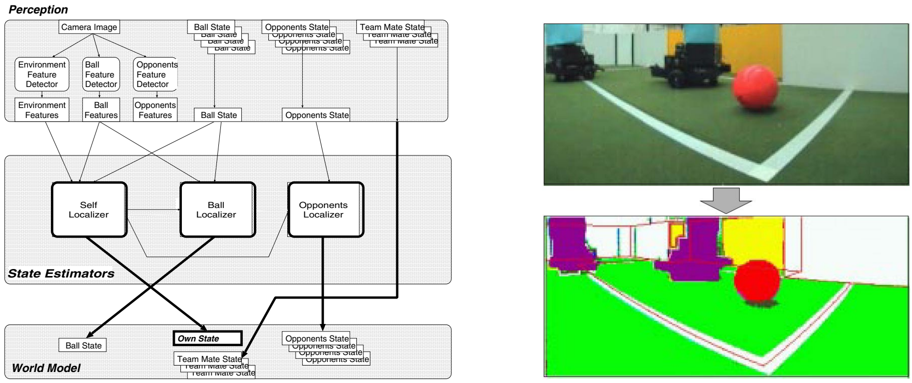

Fig. 1. (a) Architecture of the state estimator. (b) The figure shows an image captured by the robot and the feature maps that are computed for self, ball, and opponent localization.

2 Overview of the State Estimator

Fig. 1a shows the components of the state estimator and its embedding into the control system. The subsystem consists of the perception subsystem, the state estimator itself, and the world model. The perception subsystem itself consists of a camera system with several feature detectors and a communication link that enables the robot to receive information from other robots. The world model contains a position estimate for each dynamic task-relevant object. In this paper the notion of position refers to the x- and y-coordinates of the objects and includes for the robots of the own team the robot's orientation. The estimated positions are also associated with a measure of accuracy, a covariance matrix.

The perception subsystem provides the following kinds of information: (1) partial state estimates broadcasted by other robots, (2) feature maps extracted from captured images, and (3) odometric information. The estimates broadcasted by the robots of the own team comprise the estimate of the ball's location. In addition, each robot of the own team provides an estimate of its own position. Finally, each robot provides an estimate for the position of every opponent. From the captured camera images the feature detectors extract problem-specific feature maps that correspond to (1) static objects in the environment including the goal, the borders of the field, and the lines on the field, (2) a color blob corresponding to the ball, and (3) the visual features of the opponents.

The state estimation subsystem consists of three interacting estimators: the self localization system, the ball estimator, and the opponents estimator. State estimation is an iterative process where each iteration is triggered by the arrival of a new piece of evidence, a captured image or a state estimate broadcasted by another robot. The self localization estimates the probability density of the robot's own position based on extracted environment features, the estimated ball position, and the predicted position. The ball localizer estimates the probability density for the ball position given the robot's own estimated position and its perception of the ball, the predicted ball position, and the ball estimations broadcasted by the other robots. The positions of the opponents are estimated based on the estimated position of the observing robot, the robots' appearances in the captured images, and their positions as estimated by the team mates.

Every robot maintains its own global world model, which is constructed as follows. The own position, the position of the ball, and the positions of the opponent players are produced by the local state estimation processes. The estimated positions of the team mates are the broadcasted results of the self localization processes of the corresponding team mates

3 The Individual State Estimators

3.1 Self- and Ball-Localization

The self- and ball-localization module iteratively estimates, given the observations taken by the robot and a model of the environment and the ball, the probability density over the possible robot and ball positions. A detailed description and analysis of the applied alogrithms can be found in [6,7].

3.2 Opponents Localization

The objective of the opponents localization module is to track the positions of the other team's robots. The estimated position of one opponent is represented by one or more alternative object hypotheses. Thus the task of the state estimator is to (1) detect feature blobs in the captured image that correspond to an opponent, (2) estimate the position and uncertainties of the opponent in world coordinates, and (3) associate them with the correct object hypothesis. In our state estimator we use Reid's Multiple Hypotheses Tracking (MHT) algorithm [10] as the basic method for realizing the state estimation task. In this section we demonstrate how this framework can be applied to model dynamic environments in multi-robot systems. We extend the general framework in that we provide mechanisms to handle multiple mobile sensors with uncertain positions.

Multi Hypotheses Tracking We will describe the Multiple Hypotheses Tracking method by first detailing the underlying opponents model, then explaining the representation of tracked opponents position estimates, and finally presenting the computational steps of the algorithm.

The Opponents Model. The model considers opponent robots to be moving objects of unknown shape with associated information describing their temporal dynamics, such as their velocity. The number of the opponent robots may vary. The opponent robots have visual features that can be detected as feature blobs by the perception system.

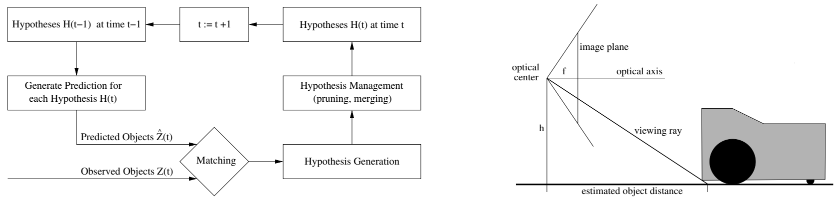

Fig. 2. (a) The multiple hypotheses framework for dynamic environment modeling. (b) An estimate of the robot's distance is given through the intersection of the viewing ray with the ground plane of the field.

The Representation of Opponent Tracks. When tracking the positions of a set of opponent robots there are two kinds of uncertainties that the state estimator has to deal with. The first one is the inaccuracy of the robot's sensors. We represent this kind of uncertainty using a Gaussian probability density. The second kind of uncertainty is introduced by the data association problem, i.e. assigning feature blobs to object hypotheses. This uncertainty is represented by a hypotheses tree where nodes represent the association of a feature blob with an object hypothesis. A node $H_j(t)$ is a son of the node $H_i(t-1)$ if $H_j(t)$ results from the assignment of an observed feature blob with a predicted state of the hypothesis $H_i(t-1)$. In order to constrain the growth of the hypotheses tree, it is pruned to eliminate improbable branches with every iteration of the MHT.

The MHT Algorithm. Fig. 2a outlines the computational structure of the MHT algorithm. An iteration begins with the set of hypotheses $H(t-1)$ from the previous iteration $t-1$. Each hypothesis represents a different assignment of measurements to objects, which was performed in the past. The algorithm maintains a Kalman filter for each hypothesis. For each hypothesis a position of the dynamic objects is predicted $\widehat{Z}_i(t)$ and compared with the next observed opponent performed by an arbitrary robot of the team. Assignments of measurements to objects are accomplished on the basis of a statistical distance measurement. Each subsequent child hypothesis represents one possible interpretation of the set of observed objects and, together with its parent hypothesis, represents one possible interpretation of all past observations. With every iteration of the MHT probabilities describing the validity of an hypothesis are calculated [1]. In order to constrain the growth of the tree the algorithm prunes improbable branches. Pruning is based on a combination of ratio pruning, i.e. a simple lower limit on the ratio of the probabilities of the current and best hypotheses, and the N-scan-back algorithm [10]. The algorithm assumes that any ambiguity at time t is resolved by time $t+N$. Consequently if at time t hypothesis $H_i(t-1)$ has n children, the sum of the probabilities of the leaf notes of each branch is calculated. The branch with the greatest probability is retained and the others are discarded.

The Unscented Transformation

The general problem is as follows. Given an n-dimensional vector random variable x with mean $\bar{x}$ and covariance C_x we would like to estimate the mean $\bar{y}$ and the covariance C_y of an m-dimensional vector random variable y. Both variables are related to each other by a non-linear transformation $y = g(x)$. The unscented transformation is defined as follows:

1. Compute the set Z of $2n$ points from the rows or columns of the matrices $\pm\sqrt{nC_x}$. This set is zero mean with covariance C_x. The matrix square root can efficiently be calculated by the Cholesky decomposition.
2. Compute a set of points X with the same covariance, but with mean $\bar{x}$, by translating each of the points as $x_i = z_i + \bar{x}$.
3. Transform each point $x_i \in X$ to the set Y with $y_i = g(x_i)$.
4. Compute $\bar{y}$ and C_y by computing the mean and covariance of the $2n$ points in the set Y.

Fig. 3. Outline of the unscented transformation.

Feature Extraction and Uncertainty Estimation This section outlines the feature extraction process which is performed in order to estimate the positions and the covariances of the opponent team's robots. Each opponent robots is modeled in world coordinates by a bi-variate Gaussian density with mean Ψ and a covariance matrix C_ψ.

At present it is assumed that the opponent robots are constructed in the same way and have approximately circular shape. All robots are colored black. Friend foe discrimination is enabled through predefined color markers (cyan and magenta, see Fig. 1b) on the robots. Each marker color may be assigned to any of the two competing teams. Consequently it is important that the following algorithms can be parameterized accordingly. Furthermore we assume that (see Fig. 2b) the tracked object almost touches the ground. The predefined robot colors allow a relatively simple feature extraction process.

Step 1: Extraction of Blobs Containing Opponent Robots From a captured image the black color-regions are extracted through color classification and morphological operators. In order to be recognized as an opponent robot a black blob has to obey several constraints, e.g. a minimum size and a red or green color-region adjacent to the bottom region row. Through this we are able to distinguish robots from black logos and adverts affixed on the wall surrounding the field. Furthermore blobs that contain or have a color-region of the own team color in the immediate neighborhood are discarded.
For all remaining regions three features are extracted: The bottom most pixel *row* which exceeds a predefined length, the column *col* representing the center of gravity and a mean blob *width* in pixels. For the latter two features only the three bottom most rows which exceed a certain length are used. In order to determine these rows, we allow also for occlusion through the ball. If the length of these rows exceeds an upper length, we assume that we have detected

$$\bar{\omega} = [\mathbf{\Phi}, row, col, width]$$

$$C_\omega = \begin{pmatrix} C_\phi & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

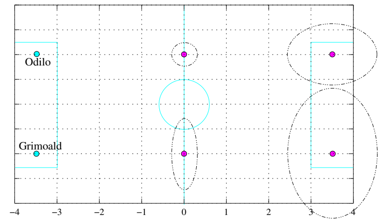

Fig. 4. (a) Intermediate $\bar{\omega}$ mean and covariance C_ω (b) Propagation of uncertainties.

two opponents which are directly next to each other. In this case two centers of gravity are computed and the width is halfed.

In order to detect cascaded robots, i.e. opponent robots that are partially occluded by other robots, our algorithm also examines the upper rows of the blobs. As soon as the length of a blob row differs significantly from the length of its lower predecessor and the respective world coordinates indicate a height of more than 10 cm above the ground we assume that we have detected cascaded robots. In this case we split the blob into two and apply the above procedure to both blobs. Empirically we have found that this feature extraction procedure is sufficient to determine accurate positions of opponent robots. Mistakenly extracted objects are generally resolved in a fast manner by the MHT algorithm.

Step 2: Estimation of Opponent Position and Uncertainty In the following we will estimate the position and covariance of an observed robot. For this the pose and the covariance of the observing robots as well as position of the detected feature blob in the image and the associated measurement uncertainties are taken into account.

We define a function *opp* that determines the world coordinates of an opponent robot based on the pose $\mathbf{\Phi}$ of the observing robot, the pixel coordinates row,col of the center of gravity and the width $width$ of the opponent robot's blob. Due to rotations and radial distortions of the lenses *opp* is non-linear. First the function *opp* converts the blob's pixel coordinates to relative polar coordinates. On this basis and the width of the observed blob the radius of the observed robot is estimated. Since the polar coordinates only describe the distance to the opponent robot but not the distance to its center, the radius is added to the distance. Finally the polar coordinates are transformed into world coordinates taking the observing robot's pose $\mathbf{\Phi}$ into account.

In order to estimate the position ψ and the covariance C_ψ of an opponent robot, we will use a technique similar to the unscented transformation [8] (see Fig. 3). First an intermediate mean $\bar{\omega}$ and covariance C_ω describing jointly the observing robot's pose and the observed robot is set up (see Fig. 4a). $\mathbf{\Phi}$, row, col and $width$ are assumed to be uncorrelated with a variance of one pixel. To this mean and covariance the unscented transformation using the non-linear mapping *opp* is applied. This yields the opponent robot's position ψ and covariance C_ψ. In Fig. 4b the uncertainties of objects depending on the uncertainty of the

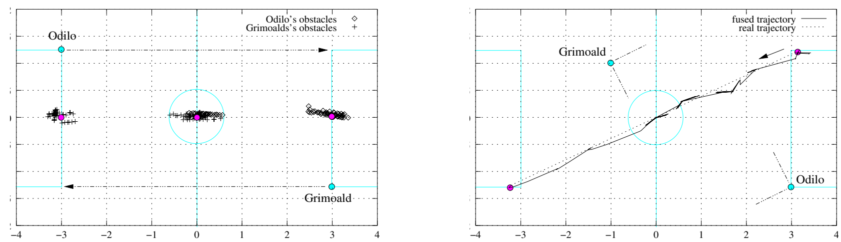

Fig. 5. (a) Two robots are traveling across the field, while they observe three stationary robots of the opponent team. The diamonds and crosses indicate the different measurements performed by the observing robots. (b) The resolved trajectory (continuous line) of an opponent robot, observed by two robots. The real trajectory is displayed as dotted line. The dashed lines indicate the robot's 90 degrees field of view.

observing robot and their relative distances are displayed using 1σ-contours. For illustrative purposes the uncertainty ellipses are scaled by an order of five. Each robot observes two obstacles in 3.5 and 7 meters distance. Robot Odilo is very certain about its pose and thus the covariance of the observed robot depends mainly on its distance. Robot Grimoald has a high uncertainty in its orientation ($\approx$ 7 degrees). Consequently the position estimate of the observed obstacle is less precise and is highly influenced by the orientation uncertainty.

Step 3: Association of Opponents to Object Hypotheses The association of an opponent robot's position with a predicted object position is currently performed on the basis of the Mahalanobis distance. In future we intent to use the Bhattacharyya distance, which is a more accurate distance measure for probability distributions.

4 Experimental Results

The presented algorithms are applied in our middle-size RoboCup team, the AGILO[1] RoboCuppers. At present, the RoboCup scenario defines a fixed world model with field-boundaries, lines and circles (see Fig. 4). Our approach was successfully applied in 1999 and 2000 during the RoboCup World Soccer Championship in Stockholm and Melbourne and the German Vision Cup. During a RoboCup match, every robot is able to process 15 to 18 frames per second with its on-board Pentium 200 MHz computer. When the robots planning algorithms' are turned off the vision system is easily able to cope with the maximum frame rate of our camera (25 fps). The localization algorithm runs with a mean processing time of 18 msec for a 16-Bit RGB ($384 * 288$) image. Only for 4% of the images the processing time exceeds 25 msec. A detailed analysis of the self- and ball-localization algorithm can be found in [6].

[1] The name is an homage to the oldest ruling dynasty in Bavaria, the Agilolfinger. The dynasties most famous representatives are Grimoald, Hugibert, Odilo and Tassilo

In the following we will present experiments that investigate the capability of tracking multiple opponent robots by our system. In the first experiment we have examined the capability of our algorithms to detect and estimate the opponent robots positions. The robots Odilo and Grimoald are simultaneously traveling in opposite directions from one side of the playing field to the other (see Fig. 5a). While they are in motion they are observing three stationary robots which are set up at different positions in the middle of the field. Diamonds and crosses indicate the observed opponents by Odilo and Grimoald, respectively. The variance in the observation is due to positions estimations over long distances (4 to 7 meters) and minor inaccuracies in the robots self-localization. Furthermore it is noteworthy that the vision system of both robots never mistook their teammate as opponent.

The second experiment examines the tracking and data fusion capability of our system. Odilo and Grimoald are set up at different positions on the field. An opponent robot crosses the field diagonally from the corner of the penalty area at the top right to the corner of the penalty area at the lower left (see Fig. 5b). The first part of the journey is only observed by Grimoald, the middle part of both robots and the final part only by Odilo. The 90 degrees field of view of both robots is indicated through lines.

The opponent robot was tracked using the MHT algorithm with a simple linear Kalman filter. The state transition matrix, described a constant velocity model and the measurement vector provided positional information only. The positions of the opponent robots and their uncertainties were computed according to the algorithm described in section 3.2. Positional variance for the pixel coordinates of the region's center of gravity and region's width was assumed to be one pixel. The process noise was assumed to be white noise acceleration with a variance of 0.1 meters. The Mahalanobis distance was chosen such that $P\{X \leq \chi_2^2\} = 0.95$. N-scan-back pruning was performed from a depth of $N = 3$. In general the update time for one MHT iteration including N-scan-back and hypo pruning was found to be less than 10 msec. This short update time is due to the limited number of observers and observed objects in our experiment. We expect this time to grow drastically with an increasing number of observing and observed robots. However within a RoboCup scenario a natural upper bound is imposed through the limited number of robots per team. A detailed analysis of the hypothesis trees revealed that only at very few occasions new tracks were initiated. All of them were pruned away within two iterations of the MHT. Overall the observed track (see Fig. 5b, continuous line) diverges relatively little from the real trajectory (dotted line).

5 Related Work

Related work comprises work done on object tracking and probabilistic localization in the robot soccer domain and probabilistic and vision-based tracking of moving targets. In the research area of autonomous robot soccer algorithms for probabilistic self-localization have been proposed. Gutmann et al. [5] have proposed a self localization method based on a Kalman filter approach by matching observed laser scan lines into the environment model. We differ from this approach mainly by using vision data instead of laser data. The advantage of using vision sensors is that the method can be applied more easily to other kinds of environments, for example outdoor environments. Enderle et al. [3] have developed a vision-based self-localization module using a sample-based Markov localization method. The advantage of Markov localization is that no assumption about the form of the probability distribution is made. However, in order to achieve high accuracy, usually a high number of samples is required. A good match between the observation and a sample pose leads to new randomly generated sample poses in the vicinity of the good sample. Hence, Markov localization leads to limited accuracy and/or relatively high computational cost. The self localization method that is proposed here has the advantage that it is faster and can be more easily integrated with the other state estimation modules. Our approach extends the Kalman filter approach [4] to self localization in that we are able to deal with nonlinearities because our approach performs a iterative optimization instead of a linear prediction.

To the best of our knowledge no probabilistic state estimation method has been proposed for tracking the opponent robots in robot soccer or similar application domains. Gutmann et al. [5] estimate the positions of the opponents and store them in the team world model but they do not probabilistically integrate the different pieces of information. This results in a less accurate and reliable estimate of the opponents positions. Probabilistic tracking of multiple moving has been proposed by Schulz et al. [11]. They apply sample-based Markov estimation to the tracking of moving people with a moving robot using laser range data. The required computational power for the particle filters is opposed by the heuristic based pruning strategies of the MHT algorithm.

Our approach to multiple hypothesis tracking is most closely related to the one proposed by Cox and Miller [2]. Indeed our algorithm is based on their implementation. We extend their work on multiple hypothesis tracking in that we apply the method to a much more challenging application domain where we have multiple moving observers with uncertain positions. In addition, we perform object tracking at an object rather than on a feature level.

6 Conclusions

In this paper, we have developed and analyzed a cooperative probabilistic, vision-based state estimation method for individual, autonomous robots. This method enables a team of mobile robots to estimate their joint positions in a known environment and track the positions of autonomously moving objects.

References

1. I. Cox and J. Leonard. Modeling a dynamic environment using a bayesian multiple hypothesis approach. Artificial Intelligence, 66:311–344, 1994.

2. I.J. Cox and S.L. Hingorani. An Efficient Implementation of Reid's Multiple Hypothesis Tracking Algorithm and Its Evaluation for the Purpose of Visual Tracking. *IEEE Trans. on PAMI*, 18(2):138–150, February 1996.
3. S. Enderle, M. Ritter, D. Fox, S. Sablatng, G. Kraetzschmar, and G. Palm. Soccer-robot locatization using sporadic visual features. In *IAS-6*, Venice, Italy, 2000.
4. O.D. Faugeras. Three-dimensional computer vision: A geometric viewpoint. *MIT Press*, page 302, 1993.
5. J.-S. Gutmann, W. Hatzack, I. Herrmann, B. Nebel, F. Rittinger, A. Topor, T. Weigel, and B. Welsch. The CS Freiburg Robotic Soccer Team: Reliable Self-Localization, Multirobot Sensor Integration, and Basic Soccer Skills. In *2nd Int. Workshop on RoboCup*, *LNCS*. Springer-Verlag, 1999.
6. R. Hanek and T. Schmitt. Vision-based localization and data fusion in a system of cooperating mobile robots. In *IROS*, 2000.
7. R. Hanek, T. Schmitt, M. Klupsch, and S. Buck. From multiple images to a consistent view. In *4th Int. Workshop on RoboCup*, *LNCS*. Springer-Verlag, 2000.
8. S. Julier and J. Uhlmann. A new extension of the kalman filter to nonlinear systems. The 11th Int. Symp. on Aerospace/Defence Sensing, Simulation and Controls., 1997.
9. C. Marques and P. Lima. Vision-Based Self-Localization for Soccer Robots. In *IROS*, 2000.
10. D. Reid. An algorithm for tracking multiple targets. IEEE Transactions on Automatic Control, 24(6):843–854, 1979.
11. D. Schulz, W. Burgard, D. Fox, and A.B. Cremers. Tracking Multiple Moving Targets with a Mobile Robot using Particle Filters and Statistical Data Association. In *ICRA*, 2001.
12. S. Thrun. Probabilistic algorithms in robotics. *AI Magazine*, 2000.
13. S. Thrun, M. Beetz, M. Bennewitz, W. Burgard, A.B. Cremers, F. Dellaert, D. Fox, D. Haehnel, C. Rosenberg, N. Roy, J. Schulte, and D. Schulz. Probabilistic algorithms and the interactive museum tour-guide robot minerva. *International Journal of Robotics Research*, 2000.