CS Freiburg: Global View by Cooperative Sensing*

Markus Dietl, Jens-Steffen Gutmann**, and Bernhard Nebel

Institut für Informatik
Universität Freiburg
79110 Freiburg, Germany
{dietl,gutmann,nebel}@informatik.uni-freiburg.de

Abstract. Global vision systems as found in the small size league are prohibited in the middle size league. This paper presents methods for creating a global view of the world by cooperative sensing of a team of robots. We develop a multi-object tracking algorithm based on Kalman filtering and a single-object tracking method involving a combination of Kalman filtering and Markov localization for outlier detection. We apply these methods for robots participating in the middle-size league and compare them to a simple averaging method. Results including situations from real competition games are presented.

1 Introduction

Mobile robots usually can only perceive a limited area of their environment at a time. In general, sensors such as laser range finders (LRFs), ultrasonic sensors or cameras have a limited field of view, so an agent cannot sense all objects around him from one sensor frame. Furthermore, a robot does not know about objects that are occluded by other objects, e.g. by walls in an office environment.

There are two ways to overcome these limitations. For one, an agent can keep a history of sensor frames (or interpretations thereof) to reason about possible objects and their locations in the environment. For example, a robot can map an environment by maintaining an occupancy grid [9] where the grid cells represent possible object locations. However, if the environment is dynamic, the grid cells only reflect snapshots of situations when the data was recorded.

Another possibility is to deploy multiple robots in the environment, each with its own sensors, and to communicate their sensor informations to a module for multi-agent sensor fusion. This approach is especially useful for dynamic environments where moving objects are present that cannot be reliably tracked by a single robot.

This work addresses the second class of methods. We develop methods for two different kinds of scenarios in dynamic environments. One, where an unknown number of multiple objects have to be tracked by a group of robots under the assumption that the sensed data is noisy but reliable, and second, the tracking of a single object by a group of robots where sensor readings are noisy and unreliable.[1] In both cases we assume that

* This work has been partially supported by *Deutsche Forschungsgemeinschaft* (DFG), by *Medien- und Filmgesellschaft Baden-Württemberg mbH* (MFG), and by *SICK AG*.

** New address: Sony Corporation, Digital Creatures Laboratory, Tokyo 141-0001, Japan.

[1] By reliable we mean that there are usually no incorrect measurements, that is, no false positives, whereas unreliable means that a robot might sense objects that are actually not present.

each robot in the team knows its own position with high accuracy and that all sensor information is acquired by on-board sensors only.

We apply these methods to the world of RoboCup for the middle size league where in our approach a team of robots maintains a consistent and accurate world model of team mates, opponents and the ball. Note, however, that the methods are general enough to apply to other domains, too, under the assumptions given above.

2 Multi Sensor Fusion in the RoboCup Environment

When developing an object tracking method, one usually has to deal with track initiation, track update including prediction and data association, and track deletion [1, 2]. This section describes the steps in our system necessary for tracking objects in the RoboCup environment.

Each robot first localizes itself by matching a scan from the LRF with an *a priori* line model of the soccer field. Experiments show that this localization technique is very accurate (usually < 2 cm and $< 1°$) and that it is faster and more robust than competing methods [5].

After estimating its own position, the robot extracts other players that are present in the field by discarding all scan points belonging to field walls and clustering the remaining ones. For each cluster the center of gravity is assumed to correspond to the center of another robot. Inherent in this approach is the systematic error due to the different robot shapes. However, it can be noted that measurements are usually reliable (no false positives).

Since our lasers are mounted at a level that prohibits the detection of the ball, we use a camera for obtaining information about where the ball is. Unfortunately our vision system has a limited quality and colors are difficult to train. Thus, ball observations are noisy and unreliable, e.g. we had problems with white field markings because when training the ball color, shiny reflections on the ball appeared to have a similar color.

All players send their own position, the position of observed other players and the position of the ball (if detected) to a global sensor integration unit which in our case runs on a central computer outside the soccer field. Here the measurements are fused in the following way.

For a player's own position, no further actions have to be carried out since the robots already determined their own position with high accuracy. However, for each player a track is initiated for fusing player observations coming from its team mates.

For each player detected by a robot, a new track is initiated or, if the player can be associated with an already existing track, it is fused with this track. We use a geometric method for data association and Kalman filtering for data fusion. Tracks containing measurements from an own player are marked as *team mate*, all others as *opponent*.

The ball position is determined by a probabilistic integration of all ball measurements coming from the players. Here we use a combination of Kalman filtering and Markov localization for achieving maximum accuracy and robustness.

If no measurements can be associated with a track for a certain amount of time (e.g. 5 secs in our implementation), the track is deleted.

The fused positions of players and ball are sent back to all robots on a regular basis where they are integrated into each robot's world model. This enables a player to extend its own view with objects it does not currently perceive and also to know about which player is friend or foe. As a result, our players have a much greater knowledge of the world than when using local perception only. Especially knowing where the ball is, appears to be useful in almost all situations and we use the sharing of this information for developing sophisticated skills [14].

Details about our probabilistic sensor fusion methods for tracking multiple and single objects are described in the next section.

2.1 Multi-Object Tracking from Reliable Data

Consider the tracking of an *a priori* unknown number of objects by a team of robots under the assumption that the measurements are reliable but might be corrupted by Gaussian noise. This scenario occurs in the RoboCup environment when players extracted from the robot's range finders are fused in the global sensor integration module.

Each of our robots detects other players in the field from data recorded by the LRF and computes heading and velocity information based on the last few observations belonging to this player using differentiation. Position, heading and velocity of each object are then communicated to the multi-sensor integration module. Thus, the observation model is a random variable $\mathbf{x}_s = (x_s, y_s, \theta_s, v_s, \omega_s)^T$ with mean $\hat{\mathbf{x}}_s$ and covariance Σ_s where (x_s, y_s) is the position, θ_s the heading and v_s and ω_s are the translational and rotational velocities of the object respectively.

As our robots know their own position with high accuracy and the LRF provides accurate data, we assume that for player observations, Σ_s is a constant diagonal matrix

$$\Sigma_s = diag(\sigma_{x_s}^2, \sigma_{y_s}^2, \sigma_{\theta_s}^2, \sigma_{v_s}^2, \sigma_{\omega_s}^2) \tag{1}$$

where $diag(\ldots)$ is a square matrix with its diagonal elements set to the given arguments and all other elements set to 0, and σ_{x_s}, σ_{y_s}, σ_{θ_s}, σ_{v_s} and σ_{θ_s} are constant standard deviations for position, heading and velocity which we manually adjusted through experiments.

Whenever a robot sends information about a player for which no already existing track can be found, i.e. if the distance to all existing tracks exceeds a certain threshold, a new track is initiated. Tracks are modeled as Gaussian variables $\mathbf{x}_r$ with mean $\hat{\mathbf{x}}_r$ and covariance Σ_r. Thus, when initiating a new track, it is set to

$$\hat{\mathbf{x}}_r = \hat{\mathbf{x}}_s, \;\; \Sigma_r = \Sigma_s \tag{2}$$

For predicting the state of a track over time, we use a simple motion model where we assume that the object moves and rotates with constant speed. Given a certain time interval t, the track is projected according to

$$\hat{\mathbf{x}}_r \leftarrow F_s(\hat{\mathbf{x}}_r, t) = \begin{pmatrix} \hat{x}_r + \cos(\hat{\theta}_r)\hat{v}_r t \\ \hat{y}_r + \sin(\hat{\theta}_r)\hat{v}_r t \\ \hat{\theta}_r + \hat{\omega}_r t \\ \hat{v}_r \\ \hat{\omega}_r \end{pmatrix} \tag{3}$$

$$\Sigma_r \leftarrow \nabla F_s \Sigma_r \nabla F_s^T + \Sigma_a(t) \tag{4}$$

where ∇F_s is the Jacobian of F_s and $\Sigma_a(t)$ is the covariance of some additive Gaussian noise with zero mean:

$$\Sigma_a(t) = diag(\sigma_{x_a}^2 t, \sigma_{y_a}^2 t, \sigma_{\theta_a}^2 t, \sigma_{v_a}^2 t, \sigma_{\omega_a}^2 t) \tag{5}$$

with σ_{x_a}, σ_{y_a}, σ_{θ_a}, σ_{v_a} and σ_{ω_a} being some constant standard deviations which we estimated through experiments.

Now, when a new measurement $\hat{\mathbf{x}}_s$ arrives from one of our robots which corresponds to a track $\mathbf{x}_r$, we fuse observation and track according to:

$$\hat{\mathbf{x}}_r \leftarrow (\Sigma_r^{-1} + \Sigma_s^{-1})^{-1}(\Sigma_r^{-1}\hat{\mathbf{x}}_r + \Sigma_s^{-1}\hat{\mathbf{x}}_s) \tag{6}$$

$$\Sigma_r \leftarrow (\Sigma_r^{-1} + \Sigma_s^{-1})^{-1} \tag{7}$$

Note, that since our sensor model does directly observe the system state, we can utilize the simplified Kalman filter equations found in Maybeck [8].

The success of a Kalman filter depends on a reliable data association method. In our implementation we use a geometric method that assigns measurements to tracks by minimizing the sum of squared error distances between observations and tracks [13]. Although this already yields reasonable results in practice, we want to note that the application of a probabilistic method such as joint probabilistic data association filters (JPDAF) [2, 11] might still improve our results.

2.2 Single-Object Tracking from Noisy Data

Often the task is not to track multiple objects but a single object where observations are both noisy and unreliable. Consider the case of keeping track of the ball in the RoboCup scenario. There can only be one ball in the field during a game. However, since for ball recognition one usually employs vision, accuracy and robustness are in general low compared to LRFs which – in our case – are mounted above a height where they can detect the ball.

Again we use a Kalman filter as described in section 2.1 for accurately tracking the ball. Each of our agents regularly sends ball observations to the global sensor integration module. However, the vision sensor is only able to determine the heading to the ball with good accuracy but fails to provide accurate range data, especially if the ball is far away from the robot. For a ball observation $\hat{\mathbf{x}}_b$ we cannot assume a constant covariance Σ_b due to this characteristics.

Given the range $\hat{r}_b = \sqrt{(\hat{x}_b - \hat{x}_{rob})^2 + (\hat{y}_b - \hat{y}_{rob})^2}$ and heading $\hat{\phi}_b = \tan^{-1}(\hat{y}_b - \hat{y}_{rob})/(\hat{x}_b - \hat{x}_{rob})$ of the ball with respect to the robot located at position $(\hat{x}_{rob}, \hat{y}_{rob})$, we model the uncertainty $\Sigma_{r\phi}$ of ball position as

$$\Sigma_{r\phi} = diag(\hat{r}_b \sigma_{r_b}^2, \sigma_{\phi_b}^2) \tag{8}$$

where σ_{r_b} and σ_{ϕ_b} are some constant standard deviations we adjusted by hand. From this, we can compute the observation error as

$$\Sigma_b = \nabla P \Sigma_p \nabla P^T \tag{9}$$

where

$$P(\hat{r}_b, \hat{\phi}_b, \hat{\theta}_b, \hat{v}_b, \hat{\omega}_b) = \begin{pmatrix} \hat{x}_{rob} + \hat{r}_b \cos(\hat{\theta}_b + \hat{\phi}_b) \\ \hat{y}_{rob} + \hat{r}_b \sin(\hat{\theta}_b + \hat{\phi}_b) \\ \hat{\theta}_b \\ \hat{v}_b \\ \hat{\omega}_b \end{pmatrix}$$

$$\Sigma_p = diag(\hat{r}_b \sigma_{r_b}^2, \sigma_{\phi_b}^2, \sigma_{\theta_b}^2, \sigma_{v_b}^2, \sigma_{\omega_b}^2)$$

and σ_{θ_b}, σ_{v_b} and σ_{ω_b} are further constant standard deviations we estimated by hand.

For track initiation we create a new track $\mathbf{x}_r$ and set it to the last ball observation:

$$\hat{\mathbf{x}}_r = \hat{\mathbf{x}}_b, \;\; \Sigma_r = \Sigma_b \tag{10}$$

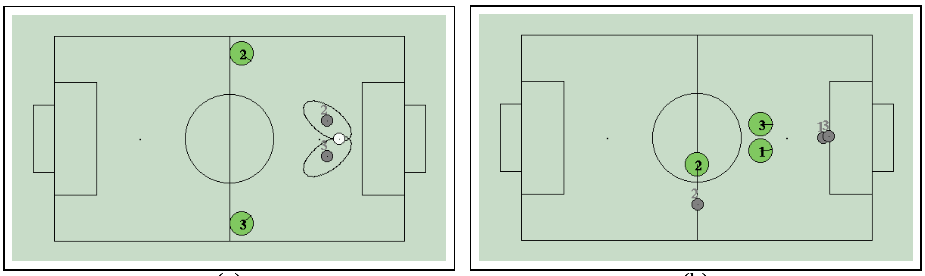
For predicting the ball state over time we use a similar function as for player movements but assume that the ball rolls in a straight line and slows down with deceleration a_b,

$$\hat{\mathbf{x}}_r \leftarrow F_b(\hat{\mathbf{x}}_r, t) = \begin{pmatrix} \hat{x}_r + \cos(\hat{\theta}_r)(\hat{v}_r - a_b t')t' \\ \hat{y}_r + \sin(\hat{\theta}_r)(\hat{v}_r - a_b t')t' \\ \hat{\theta}_r \\ \hat{v}_r - a_b t' \\ \hat{\omega}_r \end{pmatrix} \tag{11}$$

$$\Sigma_r \leftarrow \nabla F_b \Sigma_r \nabla F_b^T + \Sigma'_a(t) \tag{12}$$

where $t' = \min(t, \hat{v}/a_b)$ and $\Sigma'_a(t)$ is a similar constant covariance matrix as $\Sigma_a(t)$ to flatten the Gaussian distribution over time. Finally, fusing new observations to this track is analogous to equations (6) and (7).

It should be noted that a Kalman filter with this sensor model produces more accurate results when fusing observations from different viewpoints than e.g. a simple averaging method. This can be seen from Fig. 1(a) where the ball estimates of two players (indicated by grey circles) are integrated. Even though there is a large range error, the Kalman filter can effectively triangulate measurements from two separate viewpoints to localize the object much more precisely, because the angular error is small.



(a) (b)

Fig. 1. Fusing ball observations. Triangulation (a) and false positive observations by Player 2 (b).

The Kalman filter for ball tracking presented in this section assumes noisy but reliable data, that is, no outliers are integrated. However, in our system we sometimes observed completely wrong ball measurements by one of our robots due to reflections on walls or poorly trained colors. One possibility to filter out such outliers is to use a validation gate that discards measurements whose Mahalanobis distance is larger than a certain threshold d, where d is chosen from a χ^2 distribution.

Such a validation gate, however, has the problem that when one robot is constantly sending out wrong observations and the Kalman filter for some reasons is tracking these wrong observations and filters out all others, the global ball position becomes unusable. Furthermore, when the robot stops sending wrong observations it takes several cycles until other observations are taken into account again. For these reasons we decided to develop a more sophisticated filter method described in the next section.

2.3 Markov Localization as Observation Filter

In localization experiments carried out on a mobile robot, Gutmann, Fox, Burgard and Konolige found out that Markov localization is more robust, while Kalman filtering is more accurate when compared to each other [3]. They propose to use a combination of both methods to get a maximum robust and accurate system.

In this paper, we follow the same idea and show how to use a Markov process as an observation filter for our Kalman filter. We use a grid-based approach with a 2-dimensional (x, y) grid where each cell z reflects the probability $p(z)$ that the ball is in this cell. We initialize this grid with a uniform distribution before any observation is processed. The integration of new ball measurements is then done in two steps: prediction and update.

In the prediction step, ball motion is modeled by a conditional probability $p(z \mid z')$ which denotes the probability that the ball is at position z given that it was at position z'. Upon ball motion, the new ball position is calculated as:

$$p(z) \leftarrow \sum_{z'} p(z \mid z')p(z') \tag{13}$$

Grid-based Markov localization can be computational expensive if the size and especially the dimension of the grid is large. For efficiency, we only use a 2-dimensional grid that does not store any heading or velocity information of the ball, which means that we cannot accurately estimate the position when the ball moves. For the motion model $p(z \mid z')$ we assume that all directions are equally possible and velocities are normally distributed with zero mean and covariance σ_v^2. Therefore, $p(z \mid z')$ can be expressed as a Gaussian distribution around z':

$$p(z \mid z') \sim N(z', diag(\sigma_v^2 t, \sigma_v^2 t)) \tag{14}$$

where t is the time passed during ball motion.

In the update step, a new ball observation z_b is fused into the probability distribution according to Bayes' law:

$$p(z) \leftarrow \frac{p(z_b \mid z)p(z)}{\sum_{z'} p(z_b \mid z')p(z')} \tag{15}$$

The sensor model $p(z_b \mid z)$ determines the likelihood of observing z_b given the ball is at position z. We model it according to:

$$p(z_b \mid z) \sim N(\hat{z}_b, \Sigma'_b) \tag{16}$$

where $\hat{z}_b$ are the (x, y) components of ball observation $\hat{\mathbf{x}}_b$ as defined in section 2.2 and Σ'_b is the upper left 2×2 sub matrix of Σ_b as calculated in equation (9).

Maintaining the multi-modal probability grid makes it very easy to distinguish which ball measurement should be integrated by the Kalman filter and which not. After updating the grid with a new measurement we determine the most likely ball position, that is, the cell with the highest probability. Only measurements that are close to the most likely position are fused into the Kalman filter and all others are considered as outliers. Furthermore, if the current state of the Kalman filter does not correspond to the most likely ball position in the grid, we re-initialize the Kalman filter using this position. By using this dual probabilistic localization method we achieve high accuracy through Kalman filtering together with high robustness through Markov localization. Experimental results using this technique are presented in the next section.

3 Results

The methods presented in the previous section have been implemented on our mobile robot soccer team (see Fig. 2) and have been successfully used since our participation in the RoboCup 1999 world competition. For our first participation in 1998 we developed a similar but much simpler approach for fusing measurements from different robots [4]. In this approach, a greedy nearest-neighborhood method handled data association. For computing object positions, a weighted averaging of observations from the different players was employed. In this section we compare this simple averaging method to our new probabilistic approach from Section 2.

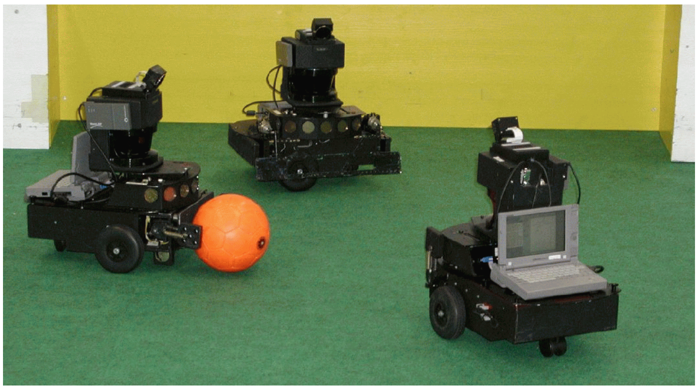


Fig. 2. CS Freiburg players. Each one is a *Pioneer I* robot equipped with *SICK* LRF, *Cognachrome* vision system, *Libretto* notebook, *WaveLan* wireless ethernet, and semi-professional kicking device developed by *SICK AG*. For an overview, see [4, 14].

In the case of multi-object tracking, we generally observed a more consistent world model for the new tracking method with a slightly higher run time due to the more sophisticated data association method. However, the differences in the world model were only marginal which we attribute to the fact that after solving the correspondence, the methods are very similar.

On the other hand, we got much better results for our new single-object tracking algorithm compared to the simple algorithm. Therefore, our main focus in this section is on results obtained by the single-object tracker.

3.1 An Ambiguous Situation

We now show how the algorithm for tracking single objects performs in an ambiguous situation. Fig. 1(b) displays a setup where two robots, Player 1 and 3, see the ball at the true location in front of the goal but one robot, Player 2, gets it all wrong and thinks the ball is somewhere on the center line. If we assume that all three players send their ball observations to the global sensor integration module on a regular basis, we get the probability distributions as shown in Fig. 3.

When integrating the first three measurements, all of them are fused by the Kalman filter since none of them has been detected to be an outlier yet. Note that after updating the grid with the 2nd measurement, the probability distribution has a sharp peak at the center line caused by the low uncertain measurement of Player 2 which thinks the ball

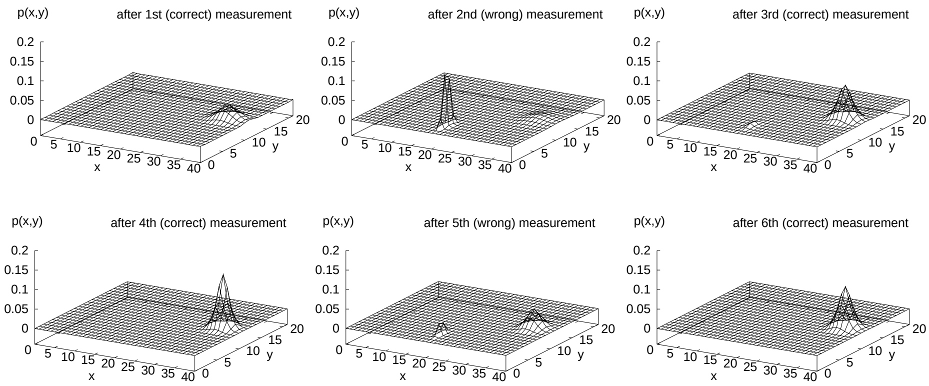
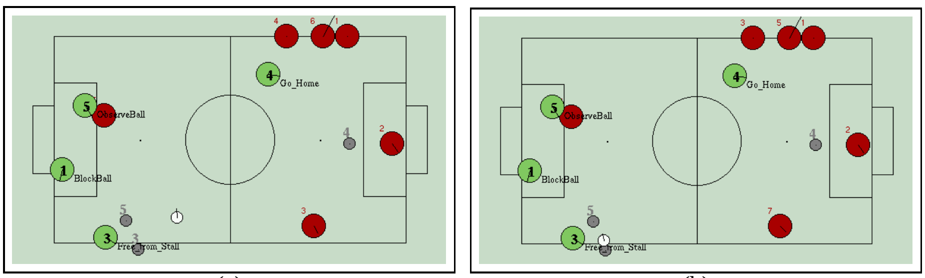


Fig. 3. Evolution of the position probability grid.

is close. After integrating more measurements, the probability distribution concentrates more and more on the true location of the ball and further measurements from Player 2 (graph in the center of bottom row in Fig. 3) cannot out-weigh the true location of the ball anymore. Thus, after the first integration of observations from all players, subsequent readings from Player 2 are filtered out and the Kalman filter tracks the ball based on observations from Player 1 and 3 only.

3.2 Real Game Scenarios

Another example is shown in Fig. 4(a). This situation has been recorded in the final game against the CoPS Stuttgart team during the German domestic *VisionCup* competition held in October 1999.



(a) (b)

Fig. 4. Ball tracking using simple averaging (a) and combination of Markov localization and Kalman filtering.

Depicted is a situation where the ball is at the wall close to Player 3, Player 3 and 5 observe the ball approximately at the right location but Player 4 thinks the ball is right in front of the opponent goal. The simple averaging method computes a global ball position somewhere between all these measurements which is shown as a white circle in Fig. 4(a). It is obvious that this global ball position is unusable and – if taken seriously by Player 3 – could lead to a disadvantageous situation where, after Player 3 moving towards the global ball position, the opponent player close to Player 5 could take over ball control.

Using our single-object tracking method, the measurements from Player 5 are reliably filtered out and the global ball position actually represents the true location of the ball (see Fig. 4(b)). Taking this information into account, Player 3 can safely approach the ball and clear the situation by pushing the ball towards the opponent's half.

In a further investigation, we examined data from the RoboCup 2000 competition held in Melbourne, Australia. During this competition, our robots recorded a total of about 120,000 ball observations in 10 games with a total playing time of more than 2 hours[2]. It turns out that in only about 0.7% of all ball observations (about 50 secs) our Markov localization approach detected an outlier. This low number can be explained by the fact that false positive measurements are rare and simultaneous observations by 2 or more robots do not occur all the time. If we compare our dual approach to one where only a Kalman filter integrates all measurements then in 72% of the filtered cases the ball position changes by more than 30 cm. Of course it is hard to tell which method is closer to the real world as there is no ground truth information from the games.

A last experiment was carried out to find out about the accuracy of our approach compared to the simple averaging one. In our laboratory, the ball was fixed at the center of the field (the origin of our global world model) with three robots around observing it. From 2000 measurements the simple averaging method reported a mean of (18 cm, 0 cm) with standard deviation (47 cm, 19 cm) whereas our method delivered a mean of (-2 cm, 4 cm) with standard deviation (8 cm, 9 cm). Thus, our new method is significantly more accurate than the averaging one.

4 Related Work

Our work is related to two research areas: object tracking and multi-robot systems.

Kluge *et al.* [7] track multiple moving objects around a wheelchair by employing a maximum-flow algorithm for data association. We use a similar geometric method but also probabilistically integrate the measurements using motion and sensor models.

Schulz *et al.* [11] use JPDAFs [2] for tracking multiple moving targets around their robot and employ particle filters for achieving robust state estimation. However, they report a running time of 2 scans per second which is infeasible on our system where each robot sends all of its observations every 100ms.

Multi-robot systems gained significant attraction in recent years. For all different aspects of mobile robot navigation, multi-robot solutions have been developed that make use of the exchange of information about the robots' beliefs and their intentions.

Probably, the most related work to ours is that of other teams in the RoboCup middle size league. The CMU Hammerheads RoboCup team also uses Kalman filters for object state estimation and reports promising results on the accuracy of this method [12]. However, they do not apply a motion model to the states, thus observations have to be taken at the same time. Furthermore, they do not consider data association and use validation gates for detecting outliers. We use a more sophisticated outlier detection method which is superior to simple validation gates.

A similar probabilistic approach to ours is that of the Agilo team [10]. They use an iterative optimization technique for estimating object positions [6] and employ JPDAFs for data association [10]. However, they do not deal with outliers as we do.

[2] See http::/www.informatik.uni-freiburg.de/~robocup for watching log files of CS Freiburg's competition games in a Java applet.

5 Conclusion

We developed new methods for tracking multiple and single objects from noisy and unreliable sensor data and compared them to a simple averaging method. Experiments show that the methods are more robust and accurate than the simple averaging one due to better probabilistic modeling of motion and sensing of objects, and a dual sensor integration approach involving Markov localization and Kalman filtering.

Our approach presented in this paper is very similar to a decentralized filter with feedback to local filters [1, pp. 371-377]. As decentralized filters can be sub-optimal, it would be interesting to compare our results to a central filter that reads all sensor data from our robots.[3] This is part of future work.

References

1. Robert Grover Brown and Patrick Y.C. Hwang. *Introduction to Random Signals and Applied Kalman Filtering*. Wiley & Sons, 1996.
2. I.J. Cox. A review of statistical data association techniques for motion correspondence. *Int. Journal of Computer Vision*, 10(1):53–66, 1993.
3. Jens-Steffen Gutmann, Wolfram Burgard, Dieter Fox, and Kurt Konolige. An experimental comparison of localization methods. In *Int. Conf. on Intelligent Robots and Systems*, 1998.
4. Jens-Steffen Gutmann, Wolfgang Hatzack, Immanuel Herrmann, Bernhard Nebel, Frank Rittinger, Augustinus Topor, and Thilo Weigel. The CS Freiburg team: Playing robotic soccer on an explicit world model. *AI Magazine*, 21(1):37–46, 2000.
5. Jens-Steffen Gutmann, Thilo Weigel, and Bernhard Nebel. Fast, accurate, and robust self-localization in polygonal environments. In *Proc. International Conference on Intelligent Robots and Systems (IROS'99)*, Kyongju, October 1999.
6. Robert Hanek and Thorsten Schmitt. Vision-based localization and data fusion in a system of cooperating mobile robots. In *Int. Conf. on Intelligent Robots and Systems*, 2000.
7. B. Kluge, C. Köhler, and E. Prassler. Fast and robust tracking of multiple moving objects with a laser range finder. In *Int. Conf. on Robotics and Automation*, Seoul, Korea, 2001.
8. Peter S. Maybeck. The Kalman filter: An introduction to concepts. In I.J. Cox and G.T. Wilfong, editors, *Autonomous Robot Vehicles*, pages 194–204. Springer-Verlag, 1990.
9. H.P. Moravec and A. Elfes. High resolution maps from wide-angle sonar. In *Proc. International Conference on Robotics and Automation (ICRA'85)*, March 1985.
10. Thorsten Schmitt, Sebastian Buck, and Michael Beetz. Agilo RoboCuppers 2001: Utility- and plan-based action selection based on probabilistically estimated game situations. In *RoboCup 2001: Robot Soccer World Cup V*. Springer-Verlag, to appear.
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 *Int. Conf. on Robotics and Automation (ICRA'01)*, pages 1665–1670, Seoul, Korea, 2001.
12. A.W. Stroupe, M.C. Matrin, and T. Balch. Distributed sensor fusion for object position estimation by multi-robot systems. In *Int. Conf. on Robotics and Automation (ICRA'01)*, pages 1092–1098, Seoul, Korea, 2001.
13. M. Veloso, M. Bowling, S. Achim, K. Han, and P. Stone. The CMUnited-98 champion small-robot team. In M. Asada and H. Kitano, editors, *RoboCup-98: Robot Soccer World Cup II*, Lecture Notes in Artificial Intelligence. Springer-Verlag, 1999.
14. Thilo Weigel, Willi Auerbach, Markus Dietl, Burhard Dümler, Jens-Steffen Gutmann, Kornel Marko, Klaus Müller, Bernhard Nebel, Boris Szerbakowski, and Maximiliam Thiel. CS Freiburg: Doing the right thing in a group. In *RoboCup 2000: Robot Soccer World Cup IV*. Springer-Verlag, 2001.

[3] We did not consider this approach due to the amount of data that would have to be transmitted through wireless communication during a game.