Cooperative Object Localization Using Line-Based Percept Communication

Daniel Göhring

Institut für Informatik
LFG Künstliche Intelligenz
Humboldt-Universität zu Berlin, Germany
goehring@informatik.hu-berlin.de

Abstract. In this paper we present a novel approach to estimate the position of objects tracked by a team of robots. Moving objects are commonly modeled in an egocentric frame of reference, because this is sufficient for most robot tasks as following an object, and it is independent of the robots localization within its environment. But for multiple robots, to communicate and to cooperate the robots have to agree on an allocentric frame of reference. Instead of transforming egocentric models into allocentric ones by using self localization information, we will show how relations between different objects within the same camera image can be used as a basis for estimating an object's position. The spacial relation of objects with respect to stationary objects yields several advantages: a) Errors in feature detections are correlated. The error of relative positions of objects within a single camera frame is comparably small. b) The information is independent of robot localization and odometry. c) Object relations can help to detect inconsistent sensor data. We present experimental evidence that shows how two non-localized robots are capable to infer the position of an object by communication on a RoboCup Four-Legged soccer field.

1 Introduction

For a mobile robot to perform a task, it is important to model its environment, its own position within the environment and the position of surrounding objects, which can be other robots as well. This task is made more difficult when the environment is only partially observable. The task is characterized by extracting information from the sensor data and by finding a suitable internal representation (model).

In hybrid architectures [1], basic behaviors or skills, such as, e.g., following a ball, are often based directly on sensor data, e.g., the ball percept. Maintaining an object model becomes important if sensing resources are limited and a short term memory is required to provide an estimate of the object's location in the absence of sensor readings.

Modeling objects and localization is often decoupled to reduce the computational burden. In this loosely-coupled system, information is passed from localization to object tracking. The effect of this loose coupling is that the quality of

U. Visser et al. (Eds.): RoboCup 2007, LNAI 5001, pp. 53–64, 2008.

the localization of an object in a map is determined not only by the uncertainty associated with the object being tracked, but also by the uncertainty of the observer's localization. In other words, the localization error of the object is the combined error of allocentric robot localization and the object localization error in the robot coordinate frame.

For this reason, robots often use an egocentric model of objects relevant to the task at hand, thus making the robot more robust against global localization errors. A global model is used for communicating information to other robots [11] or to commonly model a ball by many agents with Kalman filtering [2]. In all cases, the global model inherits the localization error of the observer.

We address this problem by modeling objects in allocentric coordinates from the start. Furthermore in RoboCup one can see a removal of more and more uniquely identifiable landmarks during the last years. The number beacons in the Four-Legged League has decreased from six to two beacons within four years. Therefore in this paper we focus on using *object to field line relations*.

In feature based belief modeling, features are extracted from the raw sensor data. We call such features *percepts* and they correspond directly to objects in the environment detectable in the camera images. In a typical camera image of a RoboCup environment, the image processing could, for example, extract the following percepts: *ball*, *line point, so called edgel*, *opponent player*, and *goal*. A *edgel* describes in our case the detection of a point that lies on a field line. Here it contains the position of that point relative to the robot in 2D space and the normal vector angle of the field line in this point, relative to the robot. Usually percepts are considered to be independent of each other to simplify computation, even if they are used for the same purpose, such as localization. Using the distance of features detected within a single camera image to improve Monte-Carlo Localization was proposed by [6]. The idea of using object relations has already been used in various map buildings tasks [12]. Using the spacial ordering of landmarks in the image for self localization was introduced by [14].

When modeling objects in relative coordinates, using only the respective percept is often sufficient. However, information that could help localize the object within the environment is not utilized. That is, if the ball was detected in the image right next to a goal, this helpful information is not used to estimate its position in global coordinates.

We show how using the object relations derived from percepts that were extracted from the same image yields several advantages:

Sensing errors. As the object of interest and the reference object are detected in the same image, the sensing error caused by joint slackness, robot motion, etc. becomes irrelevant as only the relation of the objects within the camera image matters.

Global localization. The object can be localized directly within the environment, independent of the quality of current robot localization.

Communication. Using object relations offers an efficient way of communicating sensing information, which can then be used by other robots to update their belief by sensor fusion.

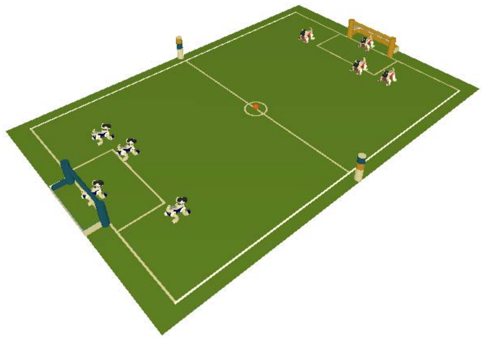

Fig. 1. As testbed served the play field of the Sony 4-Legged League. Flags, goals, lines and the ball can be found on the field at fixed positions as shown.

1.1 Outline

We will show how relations between objects in camera images can be used for estimating the object's position within a given map and in which way different types of information can be used for this task. Particularly we want to analyze how information from non-uniquely identifiable objects as field lines can be incorporated. We will present experimental results using a Monte-Carlo Particle Filter to track the ball. Furthermore, we will show how communication between agents can be used to combine incomplete knowledge from individual agents about object positions, allowing the robot to infer the object's position from this combined data.

Our experiments were conducted on the color coded field of the *Sony Four Legged League* using the Sony Aibo ERS-7, which has a camera resolution of $208 * 160$ pixels YUV and an opening angle of only 55^o.

2 Object Relation Information

In a RoboCup game, the robots permanently scan their environment for landmarks as there are flags, goals, the ball and field lines. We abstract from the algorithms which recognize the ball and the landmarks in the image as they are part of the image processing routines. In the next section we will give a brief overview over the information to be gained from each of the percepts, which is already described in more detail in [4].

2.1 Information Gained by Percepts

While describing percepts the robot receives, we want to distinguish uniquely identifiable objects from those which can not be uniquely identified. Fig. 2 gives an example of possible percepts the robot can perceive.

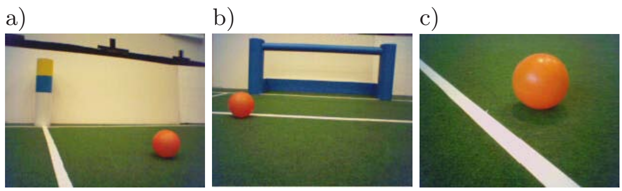

Fig. 2. Examples for what the robot can perceive: a) Flag and the ball, b) goal and the ball, c) a field-line and the ball

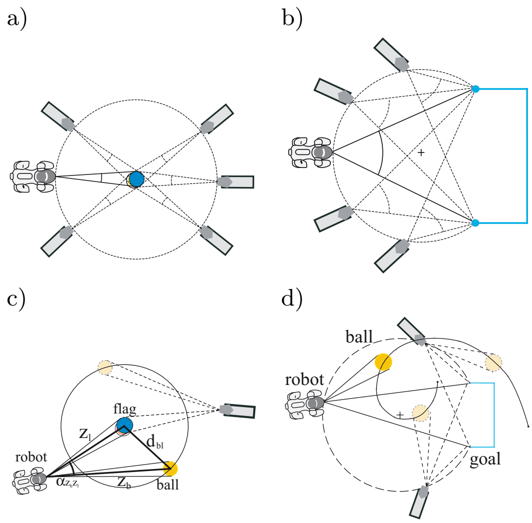

Fig. 3. Single percept: a) When a flag is seen, a circle containing all possible robot positions remains, b) The circle shows all possible positions for a seen goal. Light grey robot shapes represent possible robot positions; Two percepts in one image c) When seeing a flag and a ball in one image, the distance d_{bl} of the ball to the flag can be calculated; for all possible ball positions a circle remains, d) same situation for a seen goal and a ball, the spiral arc represents all possible ball positions.

Unique Objects. When seeing a two-colored flag, a robot actually perceives the left and right border of the flag, which enables it to calculate the distance and the angle to the flag (fig. 3 a). In the given approach this information is not being used for self localization but for calculating the distance from other objects as the ball to the flag. If a goal is detected, the robot can measure the angle between the left and the right goal-post. For a given goal-post angle the robot can calculate its distance and angle to a hypothetical circle center, whereas the circle includes the two outer points of the goal-posts and the point of the robot camera (fig. 3 b). If a ball is perceived, the distance to the ball and its direction relative to the robot can be calculated. So far all percepts we described are more or less unique, i.e., every percept can be assigned to a certain object in the robot's environment.

Table 1. Percept Statistics (Example)

Percentage of Percept Occurance in Images			
Ball	Flag	Goal	Line
35	52	22	59
Only Ball	Ball and Flag	Ball and Goal	Ball and Line
3	24	8	28

a) b)

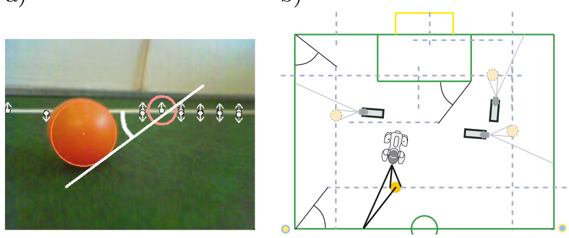

Fig. 4. a) A line-point (small circle) is seen together with a ball. The edgel data contains the position of the line on the field and the normal vector of the corresponding line (small arrows). Therefrom the ball distance to the line and the angle from the line-point to the ball can be calculated; b) Grey dotted lines represent all remaining possible ball positions on the field, when the ball-line percept is known; for better understanding the real robot position is drawn in detail, the other (schematic) robot drawings represent other possible robot and ball positions on the field.

Now we want to describe what kind of information can be gathered from field lines as an example for non-unique objects.

Non-unique Objects. On a soccer field, line information is a useful feature to reason about the robot position or about object positions. As can be seen in table 1, field lines are very often present in robot images - often together with other percepts as flags, goals or the ball. Now we will analyze which information line data can bring. We will investigate this question for the case, in which a ball and a line are seen simultaneously. When our robot perceives a line, it actually perceives one or more points of the line, together with the normal vector of the line, as fig. 4 a) shows. When a ball is seen in the same image as well, the robot can calculate a shape, containing all possible ball positions on the field. When, e.g., a ball is seen 10 cm away from an edgel in an angle of 45°, all points on the field are possible ball positions, which lie in 10 cm and an 45° angle from any edgel of the field, see fig. 4 b). Or very easy speaking: when the robot sees a ball directly on a field line, then every point on any field line could be a possible ball position.

When there is more than just one line percept in the image, many approaches exist to combine different edgels to one or more different lines [10]. Every edgel

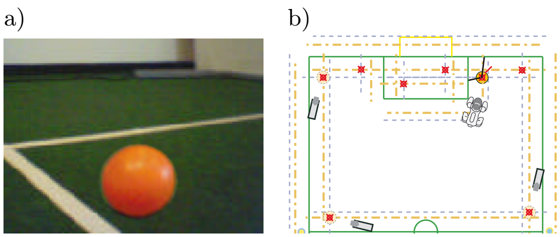

Fig. 5. a) The robot sees a ball next to two different lines; b) assuming, that the robot perceived for each of both lines an edgel percept, resulting in two solution sets (weak grey and strong orange dotted lines). One can calculate the remaining possible ball positions by cutting both solution spaces. The cut is then reduced by all solutions which would result in a wrong angle between the line percepts, related to the ball. Crossed circles (red) represent all remaining possible ball positions on the field.

can be treated as different evidence for modeling the object's position. This is especially interesting in situations where line crossings or other alignments of different lines occur at once. Being able to relate an object's position to different lines constraints the solution space for the remaining ball positions drastically, as fig. 5 b) shows. Every ball-edgel pair enables the robot to calculate possible ball positions on the field (the solution space) as in fig. 4 b). When seeing two or more of these ball-line pairs, the resulting ball positions can be calculated as the cut operation of all these solution spaces. The remaining solution space can be reduced even more, because the angle between the different edgels related to the ball is also measurable from the image (fig. 5 b)).

2.2 Dependencies between Percepts / Sensor Model

In this section we want to analyze the correlation between errors of different percepts within one image. For the sensor model, we measure the standard deviation σ^l by letting a robot take multiple images of certain scenes: a ball, a flag, a goal, a line and combinations of it. The standard deviation of distance differences and respectively angle differences of objects in the image relative to each other were measured as well. The robot is walking on the spot to keep the distance within the environment constant an to get noisy sensor data as during real robot motions. We found out that the angle errors of different percepts within the same image are strongly correlated which can be seen in fig. 6 in case of a ball and a flag.

3 Multi-agent Modeling

Now we want to describe a possible implementation of this approach. As the sensor data of our Aibo ERS-7 robot are not very accurate, we have to cope with a lot of sensor noise. Furthermore, the probabilistic distribution is not always unimodal, e.g., in cases where the observations lead to more than one

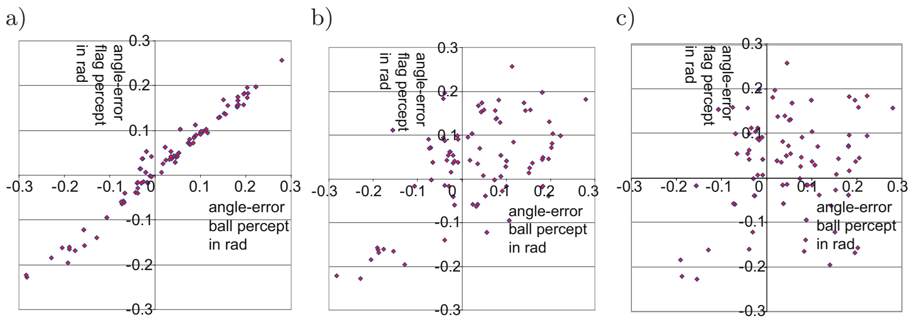

Fig. 6. The diagrams show the measured angle error to a ball and to a flag. The ball is located at a distance of 1.5m, the flag at 2.0m. a) flag and ball are seen in the same image, the angle errors between both are strongly correlated; b) the ball is seen 0.03 seconds earlier than the flag, lower correlation; c) the ball is seen 0.2 seconds earlier than the flag, almost no correlation between the angle errors.

solution for possible ball positions. This is why a simple Kalman filter would not be sufficient [7]. We chose an implementation using a Monte-Carlo Particle Filter because of its ability to model multimodal distributions and its robustness to sensor noise. Other approaches as Multi Hypothesis Tracking or Grid Based algorithms might work also [5]. As we cope with static situations this time only, we could abstract from network communication time and the delay after which percept relations were received.

3.1 Monte-Carlo Filter For Multi Agent Object Localization

Markov localization methods, in particular Monte-Carlo Localization (MCL), have proven their power in numerous robot navigation tasks, e.g., in office environments [3], in the museum tour guide Minerva [13], in the highly dynamic RoboCup environment [8], and outdoor applications in less structured environments [9]. MCL is widely used in RoboCup for object and self localization [7] because of its ability to model arbitrary distributions and its robustness towards noisy input data. The probability distribution is represented by a set of samples, called particle set. Each particle represents a pose hypothesis. The current belief of the object's position is modeled by the particle density, i.e., by knowing the particle distribution the robot can approximate its belief about the object state. Thereby the belief function $Bel(s_t)$ describes the probability for the object state s_t at a given time t. Using the Markov assumption and Bayes law, the belief function $Bel(s_t)$ depends only on the previous belief $Bel(s_{t-1})$, the last robot action u_{t-1} and the current observation z_t:

$$Bel^{-}(s_t) \longleftarrow \int \underbrace{p(s_t|s_{t-1}u_{t-1})}_{\text{process model}} Bel(s_{t-1}) ds_{t-1} \tag{1}$$

$$Bel(s_t) \longleftarrow \eta \underbrace{p(z_t|s_t)}_{\text{sensor model}} Bel^-(s_t) \tag{2}$$

whereas η is a normalizing factor. Equation (1) shows how the *a priori* belief Bel^- is calculated from the previous Belief $Bel^-(s_{t-1})$. It is the belief prior the sensor data, therefore called prediction. As our robots do not perform any actions with the ball and as the situation is static, our propagation step becomes very simple or can be left out. In (2) the a-priori belief is updated by sensor data z_t, therefore called update step. Our update information is information about object relations as described in section 2.1. The data from fig. 6 can serve as a sensor model, telling the filter how accurate the sensor data are. The particles are distributed equally at the beginning, then the filtering process begins.

3.2 Monte-Carlo Localization, Implementation

Our hypotheses space for object localization has two dimensions for the position q on the field. Each particle s^i can be described as a state vector $\overrightarrow{s}^i$

$$\overrightarrow{s}^i = \begin{pmatrix} q^i_{x_t} \\ q^i_{y_t} \end{pmatrix} \tag{3}$$

and its likelihood p^i.

The likelihood of a particle p^i can be calculated as the product of all likelihoods of all gathered evidence [12]. From every given sensor data, e.g., a landmark l and a ball (with its distances and angles relative to the robot) we calculate the resulting possible ball positions relative to the landmark l. The resulting arc will be denoted as ξ^l. We showed in 2.1 that ξ^l has a circular form, when l is a flag, a spiral form, when l is a goal or a set of lines, when l is an edgel. The shortest distance δ^l from each particle $\overrightarrow{s}^i$ to ξ^l is our argument for a Gaussian likelihood function $\mathcal{N}(\delta, \mu, \sigma)$. The parameters of the Gaussian where derived experimentally. The sensor model being assumed to be Gaussian showed to be a good approximation in experiments. The likelihood is being calculated for all seen landmarks l and then multiplied:

$$p^i = \prod_{l \in L'} \mathcal{N}(\delta^l, 0, \sigma) \tag{4}$$

In cases without new evidence all particles get the same likelihood. After likelihood calculation, particles are resampled.

Multi Agent Modeling. Percept relations from every robot are communicated to every other robot. The receiving robot uses the communicated percept relations the same way it uses its own for likelihood calculation of each particle

- By communicating percept relations rather than particles, every robot can incorporate the communicated sensor data to calculate the likelihood of its particle set.

4 Experimental Results

As a test platform served the Aibo ERS-7. In the first reference algorithm, to which we compare our approach, two robots try to localize and to model the ball in an egocentric model. As a result each robot maintains a particle distribution for possible ball positions, resulting from self localization belief and the locally modeled ball positions. In our situation neither robot is able to accurately determine the ball position. Then the two robots communicate their particle distribution to each other. After communication each robot creates a new particle cloud as a combination of its own belief and the communicated belief (communicated particle distribution). We want to check how this algorithm performs in contrast to our presented algorithm in situations, where self localization is not possible, e.g., when every robot can only see one landmark and the ball.

In our first experiment, we placed both robots in front of different landmarks, one in front of a goal and one in front of a line with partially overlapping fields of view, such that both robots could see the ball (fig. 7 and 8).

The robots cannot accurately model the ball position when just communicating particle distributions, whereas by communicating percept relations the modeled

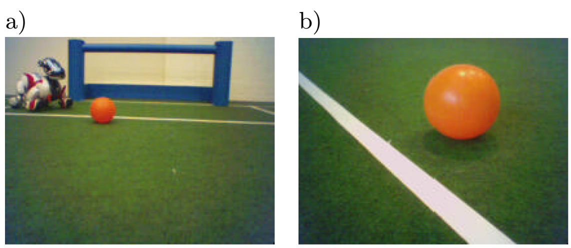

Fig. 7. Experiment A: views from two robots: a) robot A seeing a ball and a goal; b) robot B seeing a ball and a line

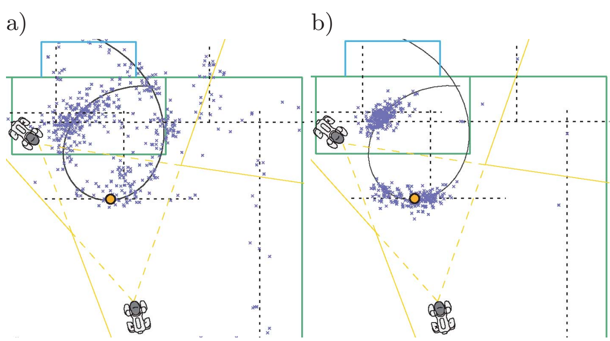

Fig. 8. Experiment A: the modeled ball position. a) both robots try to localize and have an egocentric ball model. After interchanging their particle distribution, the particle cloud does not convergence to a confined area; b) robots interchange the percept relations (ball-line and ball-goal), then updating and resampling the particle distribution. The distribution converges quickly to two small areas.

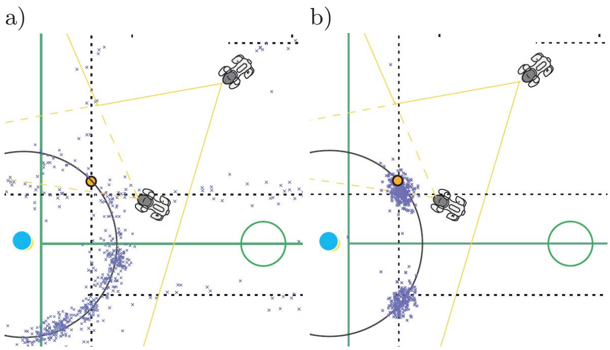

Fig. 9. Experiment B: The upper robot can see the ball and a line, the lower robot can see the flag only, because it is too far away to see the line. a) communicating particles does not lead to a convergence of the particles; b) communicating percept relations leads to convergence of the particle cloud to two small areas.

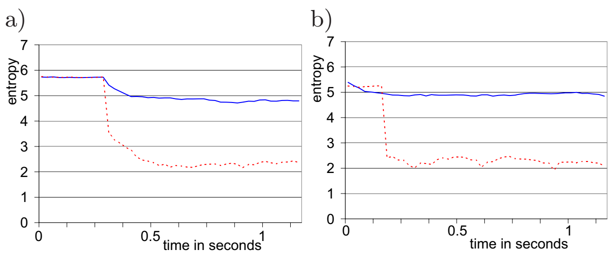

Fig. 10. Entropies over time for the experiments A and B. a) Experiment A: the dotted line represents the entropy for communicating percept relations, the continuous line represents the particle entropy for communicating percept distributions; b) Experiment B: dotted line represents ball particle entropy when communicating percept relations, continuous line for communicating particle distributions.

position converges to two small areas (fig. 8). Entropy measurement shows this quantitatively in fig. 10 a) - the entropy is much smaller, when percept relations are communicated. In Experiment B (fig. 9) one robot sees a flag, the other robot sees a line and both can see the ball. Again the robots try to localize and model the ball position egocentricly. Then they transform the egocentricly modeled ball particles into allocentric coordinates and communicate the particle distribution to each other. Simple particle communication does not lead to a convergence of the resulting particle distribution, whereas communicating percept relations leads to a convergence to a confined area (fig. 9 b)). Also entropy is much smaller again, when communicating percept relations 10 b).

5 Conclusion

Object relations, especially line information, in robot images can be used to localize objects in allocentric coordinates, e.g., if a ball is detected in an image next to a goal, the robot can infer something about where the ball is on the field. Percept relations can also help to detect image processing errors. Without having to be localized at all, it can accurately estimate the position of an object within a map of its environment using nothing but object relations. Furthermore, we were able to show how the process of object localization can be sped up by communicating object relations to other robots. Two non-localized robots are thus able to both localize an object using their sensory input in conjunction with communicated object relations.

Future Work. Future work will investigate, how the presented approach can be extended to moving objects, letting the robot infer not only about the position but also about the speed. Another interesting question would be, how redundant computation that is done by every agent can be distributed among the different robots while staying robust against system failures of different robots.

Acknowledgments

Program code used was developed by the GermanTeam. Source code is available for download at http://www.germanteam.org

References

1. Arkin, R.C.: Behavior-Based Robotics. MIT Press, Cambridge (1998)
2. Dietl, M., Gutmann, J., Nebel, B.: Cooperative sensing in dynamic environments. In: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2001), Maui, Hawaii (2001)
3. Fox, D., Burgard, W., Dellaert, F., Thrun, S.: Monte carlo localization: Efficient position estimation for mobile robots. In: Proceedings of the Sixteenth National Conference on Artificial Intelligence and Eleventh Conference on Innovative Applications of Artificial Intelligence (AAAI), pp. 343–349. The AAAI Press/The MIT Press (1999)
4. Göhring, D., Burkhard, H.-D.: Multi robot object tracking and self localization using visual percept relations. In: Proceedings of IEEE/RSJ International Conference of Intelligent Robots and Systems (IROS), pp. 31–36. IEEE (2006)
5. Gutmann, J.-S., Fox, D.: An experimental comparison of localization methods continued. In: Proceedings of the 2002 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE (2002)
6. Kaplan, K., Celik, B., Mericli, T., Mericli, C., Akin, L.: Practical extensions to vision-based monte carlo localization methods for robot soccer domain. In: Bredenfeld, A., Jacoff, A., Noda, I., Takahashi, Y. (eds.) RoboCup 2005. LNCS (LNAI), vol. 4020. Springer, Heidelberg (to appear, 2006)
7. Kwok, C., Fox, D.: Map-based multiple model tracking of a moving object. In: Nardi, D., Riedmiller, M., Sammut, C., Santos-Victor, J. (eds.) RoboCup 2004. LNCS (LNAI), vol. 3276, pp. 18–33. Springer, Heidelberg (2005)
8. Lenser, S., Bruce, J., Veloso, M.: CMPack: A complete software system for autonomous legged soccer robots. In: AGENTS 2001: Proceedings of the fifth international conference on Autonomous agents, pp. 204–211. ACM Press (2001)

9. Montemerlo, M., Thrun, S.: Simultaneous localization and mapping with unknown data association using FastSLAM. In: Proceedings of the 2003 IEEE International Conference on Robotics and Automation (ICRA), pp. 1985–1991. IEEE (2003)
10. Nguyen, V., Martinelli, A., Tomatis, N., Siegwart, R.: A comparison of line extraction algorithms using 2d laser rangefinder for indoor mobile robotics. In: Proceedings of the IEEE/RSJ Intenational Conference on Intelligent Robots and Systems, IROS, Edmonton, Canada. IEEE (2005)
11. Schmitt, T., Hanek, R., Beetz, M., Buck, S., Radig, B.: Cooperative probabilistic state estimation for vision-based autonomous mobile robots. IEEE Transactions on Robotics and Automation 18(5), 670–684 (2002)
12. Thrun, S., Burgard, W., Fox, D.: Probabilistic Robotics. MIT Press, Cambridge (2005)
13. Thrun, S., Fox, D., Burgard, W.: Monte carlo localization with mixture proposal distribution. In: Proceedings of the National Conference on Artificial Intelligence (AAAI), pp. 859–865 (2000)
14. Wagner, T., Huebner, K.: An egocentric qualitative spatial knowledge representation based on ordering information for physical robot navigation. In: ECAI 2004, Workshop on Issues in Designing Physical Agents for Dynamic Real-Time Environments (2004)