Distributed Object Modeling Using Object Relations in Dynamic Environments

Daniel Göhring

Institut für Informatik,
Humboldt-Universität zu Berlin,
Unter den Linden 6,
10099 Berlin, Germany

Abstract. In this paper we describe how a group of agents can commonly estimate the position of objects. Furthermore we will show how these modeled object positions can be used for an improved self localization. Modeling of moving objects is commonly done by a single agent and in a robo-centric coordinate frame because this information is sufficient for most low level robot control and it is independent of the quality of the current robot localization. Especially when many robots cooperate with each other in a partially observable environment they have to share and to communicate information. For multiple robots to cooperate and to share information, though, they need to agree on a global, allocentric frame of reference. But when transforming the egocentric object model into a global one, it inherits the localization error of the robot in addition to the error associated with the egocentric model.
We propose using the relation of objects detected in camera images to other objects in the same camera image as a basis for estimating the position of the object in a global coordinate system. The spacial relation of objects with respect to stationary objects (e.g., landmarks) offers several advantages: The information is independent of robot localization and odometry. It can be easily communicated. We present experimental evidence that shows how two robots are able to infer the position of an object within a global frame of reference, even though they are not localized themselves. We will also show, how to use this object information for self localization. A third aspect of this work will cope with the communication delay therefore we will show, how the Markov Model can be extended for distributed object tracking.

1 Introduction

In multi-agent systems for each robot it is important when performing a task to model its environment, its own position within the environment, and the position of other robots and moving objects. In RoboCup, the most important object to track is, naturally, the ball. The task of estimating the position of an object is made more difficult when it comes to the fact that the environment is only partially observable to the robot.

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.

A global model is used for communicating information to other robots [7], to commonly model a ball by many agents with Kalman filtering [2] or to model object-environment interactions [5]. 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. To achieve this, the sensing process needs to be examined more closely. 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*, *opponent player*, and *goal*. These percepts are then each passed on to different respective modules, e.g., the ball modeler receives (only) the ball percept. Percepts are commonly considered to be independent of each other to simplify computation, even if they are used for the same purpose, such as localization [6]. Using the distance of features detected within a single camera image to improve Monte Carlo Localization was proposed by [4]: when two landmarks are detected simultaneously, the distance between them yields information about the robot's whereabouts.

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.

Therefore 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. (The sensing error is thus solely comprised of errors associated with image processing.)

Global localization The object can be localized directly within the environment, independent of the quality of current robot localization. Moreover the object position can be used by the robot for self 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. This is in stark contrast to what is necessary to communicate the entire probability density function associated with an object.

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. 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. In a further step we will demonstrate how this knowledge about object position can be used to improve self localization. We will extend the Markov Model to count for communication delays.

Our experiments were conducted on the color coded field of the *Sony Four Legged League* as well as on the MidSize Field (12m*8m) 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, and the ball. We abstract from the algorithms which recognize the ball, the flags, and the goals in the image as they are part of the image processing routines. The following section presents the information gained by each perception.

2.1 Information gained by a single percept

If the robot sees a two colored flag, it actually perceives the left and the right border of this flag and thus the angle between those two borders. Because the original size of landmarks is known, the robot is able to calculate its own distance to the flag and its respective bearing (Fig. 2 a). In the given approach we don't need that sensor data 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. 2 b).

If a ball is perceived, the distance to the ball and its direction relative to the robot can be calculated. Lines or line crossings can also be used as reference marks, but the sensor model for lines is more complex than for a goal or a flag as there are many equally looking line segments on the field. For simplicity reasons we didn't use line information in the given approach.

Fig. 1. As testbed served the play field of the Sony 4-Legged League.

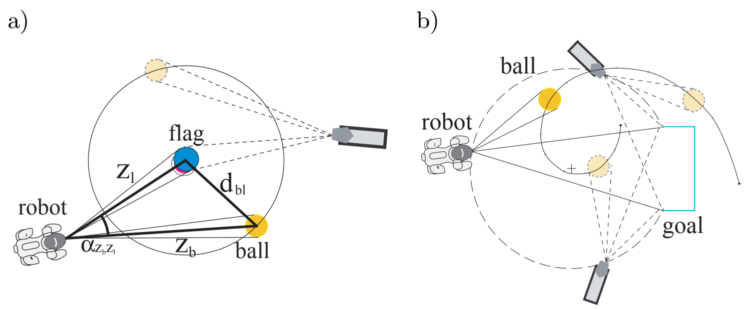

Fig. 2. Two percepts in one image a) a flag and a ball let the robot determine the ball's distance relative to the flag d_{bl}; all possible positions of the ball relative to the flag form a circle, b) the same calculation for a goal and a ball. The circular arc determines all possible positions for the robot, the spiral arc represents all possible ball positions.

2.2 Information gained by two percepts within the same image

If the localization object is visible together with another landmark, e.g., a flag or a goal, the robot does not only get information about distances to both objects but also information about the angle between them. With the *law of the cosine* the distance from the ball to a flag can be calculated (Fig. 2 c).

When a goal and a ball were seen, a similar determination of the position can be done for the ball, but the set of possible solutions leads to a spiral curve (Fig. 2 d). Now we have shown how object relations can help to constrain the set of possible ball positions. But we have also seen that one landmark and one ball alone are not sufficient to exactly determine the ball's position. One possibility to overcome this limitation would be to scan for other landmarks and take this information into account, but this could be time consuming. Another approach would be to let the robots communicate and interchange the necessary information for an accurate object localization. This has two advantages:

1. Apart from communication time which takes, in our case, about two or three tenth of a second, information transfer between robots is cheap in resources, as only few data needs to be transferred.
2. Many robots can gather more information than a single robot, because many robots can see more than one robot.

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 solution for possible ball positions. This is why a simple Kalman filter would not be sufficient [5]. 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 [3].

3 Sensor Combination in Multi Robot Systems

There are several methods to combine the knowledge of two or more robots. One method is to combine the belief of the robots after the modeling (belief fusion) [2]. Another method is to communicate the sensor data, i.e., the percept relations before the modeling (sensor fusion). We decided for the latter approach because it yields several advantages.

3.1 Sensor Fusion

Sensor Fusion means that both robots are communicating the percept relations before the modeling. Later all percept relations are incorporated into the modeling process equally. The advantage is that percept relations are cheap to communicate - in contrast to communicating particle distributions, and they are easy to merge. An example merge for a two robot scenario can be seen in figure 3. Furthermore, in static situations a normal Bayes filter can be used. In contrast, when communicating particle distributions, one has to think about how to combine both. In the experiment section 4 we will compare both methods.

3.2 Handling Communication Delays

When there are communication delays in dynamic situations the communicated information must be assigned to the time when it has been perceived. Fig. 4 demonstrates the effect of to late arriving percept relations.

a) b)

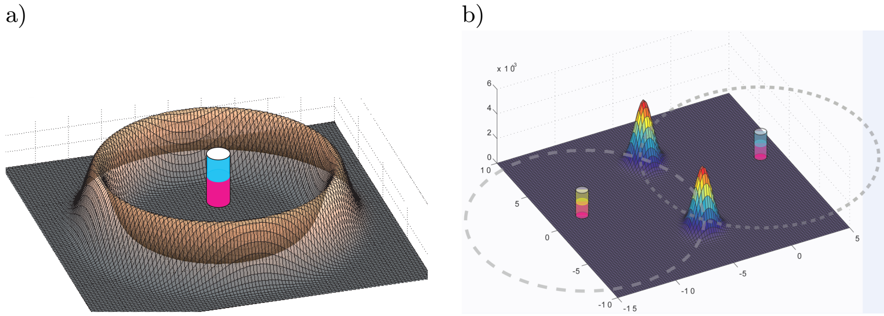

Fig. 3. Probability distributions for the ball position - a) for one robot seeing a ball and a landmark. b) For two robots, each seeing one landmark and the ball.

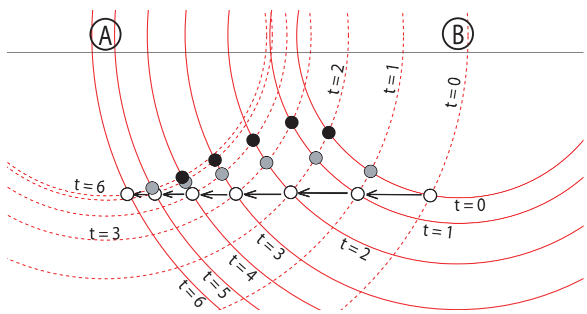

Fig. 4. Effects of communication delay. White ball moving from right to left. Two agents (not in the diagram), agent A sees the ball and flag A, agent B communicates its percept relation consisting of the ball and flag B. Gray balls represent modeled ball positions with communication delay of 1, black balls represent modeled positions resulting from a delay of 2.

We investigated how delayed information can correctly be incorporated into the modeling process. Therefore we assume that the time delay is constant and that it takes e time steps for the the percept to get from B to A. In case of $e = 0$ we get the Markov chain as in fig. 5. As a result, if one wants to model the object state s_t of time t, he has to wait until $t+e$ to get all sensor data from communicating agents.

As an example we want to model model state s_{t+e+1} with a given time delay of $e > 0$ time steps and two robots A and B. Our solution is, to use state s_{t+1}, where sensor information z^A_{t+1} and z^B_{t+1} is available, to remember the belief, e.g., particle distribution of s_{t+1} and then incrementally and a-priori model state s^*_{t+2}, s^*_{t+3}, ..., until s^*_{t+e+1} with just the sensor information z^A_{t+1} from A. In the next $t+e+2$ step we receive sensor data z^B_{t+2} from B, which we can use to revise the a-priori state estimation s^*_{t+2}. Therefore we remembered the belief function

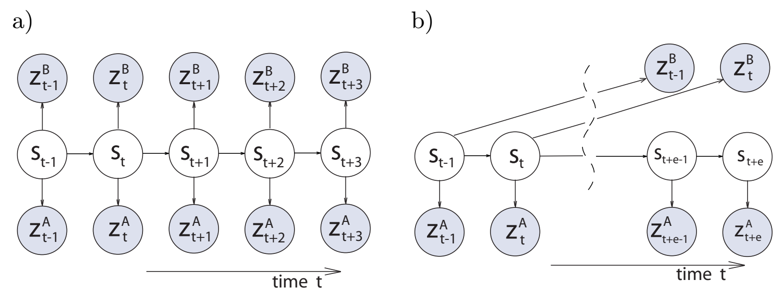

Fig. 5. Robot b is sending its percept to robot A. a) no communication delays; b) communication delay $e > 0$.

from s_{t+1}. Now we continue as already described. We call this method *history revision*. This approach allows incorporating communicated sensor data when time delays are given. It is feasible for small communication delays. But if the communication delay gets bigger the effort to revise the a-priori states from the past and to propagate them into the future gets linearly bigger.

3.3 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 $\overrightarrow{s}^i$

$$\overrightarrow{s}^i = \begin{pmatrix} q^i_{x_t} \\ q^i_{y_t} \end{pmatrix} \tag{1}$$

and its likelihood p^i.

The likelihood of a particle p^i can be seen as the product of all likelihoods of all gathered evidences [6], which means in our case that for all landmark-ball pairs a likelihood is being calculated. 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, as described in section 2.2. The resulting arc will be denoted as ξ^l. We showed in 2.2 that ξ^l has a circular form, when l is a flag and a spiral form, when l is a goal. The shortest distance δ^l from each particle $\overrightarrow{s}^i$ to ξ^l is our argument for a Gaussian likelihood function $\mathcal{N}(\delta, \mu, \sigma)$, where $\mu = 0$ and with a standard deviation σ, which is determined as described in the next section. 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:

$$p^i = \prod_{l \in L'} \mathcal{N}(\delta^l, 0, \sigma) \tag{2}$$

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

Multi Agent Modeling. To incorporate the information from other robots, percept relations are communicated to other robots. The receiving robot uses the communicated percepts for likelihood calculation of each particle the same way as if it was its own sensor data. This is advantageous compared to other approaches:

– Some approaches communicate their particle distribution, which can be useful when many objects are modeled in parallel. But when, as in our examples, two robots only know the arcs or the circular function on which the ball could be found, this would increase position entropy rather than decreasing it. Communicating whole particle sets can also be very expensive in resources.
– By communicating percept relations rather than particles, every robot can incorporate the communicated sensor data to calculate the likelihood of its particle set. Thereby we get a kind of sensor fusion rather than Belief-fusion as in case when particle distributions are communicated.

Because of this, we decided to let every robot communicate every percept relation (e.g., flag, ball) it has gathered to other robots.

Sensor Model. For the sensor model, we measured the standard deviation σ^l by letting a robot take multiple images of certain scenes: a ball, a flag, a goal 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 was walking the whole time on the spot to get more realistic, noisy images. The experiment results are shown in table 1.

Object		Standard Deviation σ	
	Distance in mm	σ_{Dst} in mm	σ_{Ang} in Rad
Ball	1500	170	0.015
Flag	2000	273	0.019
Goal	2000	25	0.021
Flag- Ball-Diff.	500	196	0.008
Goal- Ball-Diff.	500	175	0.0054

Table 1. Object Distance Standard Deviations

It can be seen that the standard deviation for the distance from the ball to the flag (or goal) is smaller than the sum of the distance errors given a ball and a flag (or goal). The same can be said for the angle standard deviation. This gives evidence that the sensor error for percepts in the same image is correlated, due to walking motions and head swings. Because in our experiments we coped with static situations only, we could abstract from network communication time and the delay after which percept relations were received.

3.4 Self Localization

For self localization we used the algorithm described in [6]. We used a three dimensional hypothesis space, two dimension for the field position of the robot and one dimension for its orientation. As sensor update input data served the angle to the goal posts and to the flag boundaries as in [6], plus in our approach, the distance and angle to the modeled ball.

4 Experimental Results

We will conduct experiments to investigate how the extended Markov Model can be used for an accurater object modeling not only of static but also of dynamic objects. We will also show, how object relations can be used for distributed object tracking.

The Aibo ERS-7 robot serves as a test platform for our work. We conducted several experiments with our robots to measure the accuracy and the speed of the given algorithm and to compare it to reference algorithms. In the first reference algorithm, 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 (Experiment A,B). In the next step 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 (the own particle distribution) 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 a different landmarks with partially overlapping fields of view, such that both robots could see the ball (Fig. 6).

One can see from the experiments that there is almost no convergence to a confined area for the case in which the two robots are communicating their particle distributions to each other. In case of percept communication, the particle distribution converges nicely to a confined area.

In the next experiment we put one robot in front of a flag and a ball and let it try to localize. The next reference algorithm we used was the self localization approach as described in [6]. As the robot could only see one landmark, the particle distribution did not converge to a certain area, two circle like clouds remained, one for the ball and one for the self localization particle distribution (fig. 7 a). As one can see, accurate self localization was not possible. It was also not possible in case for two robots not interchanging percept relations, because the ball particle distribution did not converge as in fig. 6 a). But when we took two robots and let them determine the ball position using percept relations, a

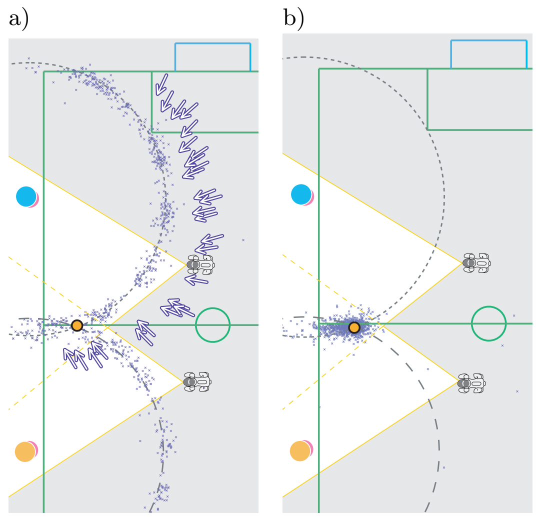

Fig. 6. Experiment A - two flags: a) no percept relations communicated, the robots are self localizing (arrows show SL-particles of the upper robot schematically), the ball positions (cloud of dots) are modeled egocentric and then transformed into global coordinates. The globally modeled ball particle distribution is then communicated to the other robot, which merges the communicated distribution with its own ball particle distribution. b) No self localization needed, percept relations used as described, two robots communicating object relations for calculating the particle distribution; the small circle at the center line marks the real ball position in the given experiment

robot can use its own distance and angle to the ball for improved self localization. Fig. 7 b) shows that self localization could be improved when using percept relation and the resulting ball position data. The lower entropy of the self localization particle distribution proves quantitatively, that using position data from objects modeled in allocentric coordinates can reduce uncertainty in self localization (fig. 8).

5 Conclusion

Object relations 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. 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. Fu rthermore, we were able to show how the process of object localization can be sped up by

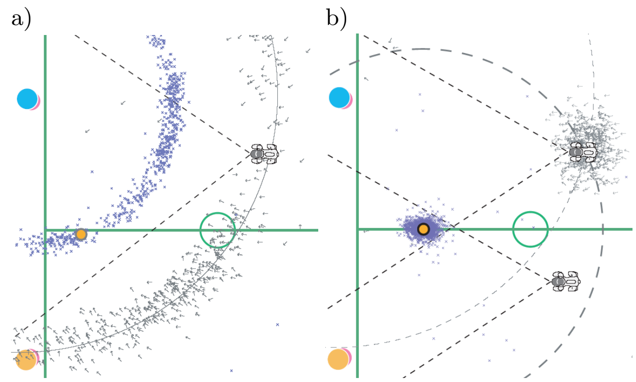

Fig. 7. Experiment C - Ball and robot localization: a) one robot is perceiving the ball and self localizing by the upper flag. A circular particle distribution remains for the robot positions (bigger circle) and the ball positions (smaller circle); b) two robots localizing the ball with percept relations, the upper robot is robot.

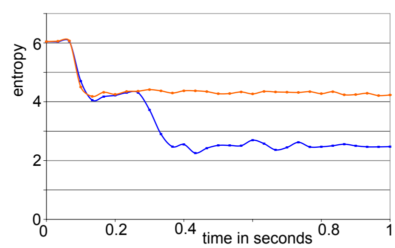

Fig. 8. The entropies for particle distributions of the self localization process (Experiment C). The orange line shows the self localization entropy when no object relations were used. Entropy decreases when perceiving the flag but remains at a high level; The self localization entropy becomes much lower when using visual object relations for ball modeling.

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. In a next step we showed how the gained knowledge about allocentric object positions can be used for an improved Markov self localization.

Future Work. Future work will investigate the use of other landmarks (e.g., field lines) for object localization. An active vision control is currently being developed to gain more images containing object relations, e.g., looking at the ball and landmarks at once if possible. Current work tries to extend the presented approach to moving objects.

Acknowledgments

Program code used was developed by the GermanTeam. Source code is available for download at http://www.germanteam.org. Special thanks to Christine Zarges, University of Dortmund for her support.

References

1. R. Arkin. *Behavior-Based Robotics*. MIT Press, Cambridge, MA, USA, 1998.
2. M. Dietl, J. Gutmann, and B. Nebel. Cooperative sensing in dynamic environments. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'01), Maui, Hawaii*, 2001.
3. J.-S. Gutmann and D. Fox. An experimental comparison of localization methods continued. In *Proceedings of the 2002 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2002.
4. K. Kaplan, B. Celik, T. Mericli, C. Mericli, and L. Akin. Practical extensions to vision-based monte carlo localization methods for robot soccer domain. In I. Noda, A. Jacoff, A. Bredenfeld, and Y. Takahashi, editors, *9th International Workshop on RoboCup 2005 (Robot World Cup Soccer Games and Conference)*, Lecture Notes in Artificial Intelligence. Springer, 2006. To appear.
5. C. Kwok and D. Fox. Map-based multiple model tracking of a moving object. In D. Nardi, M. Riedmiller, C. Sammut, and J. Santos-Victor, editors, *8th International Workshop on RoboCup 2004 (Robot World Cup Soccer Games and Conferences)*, volume 3276 of *Lecture Notes in Artificial Intelligence*, pages 18–33. Springer, 2005.
6. T. Röfer and M. Jüngel. Vision-based fast and reactive monte-carlo localization. In D. Polani, A. Bonarini, B. Browning, and K. Yoshida, editors, *Proceedings of the 2003 IEEE International Conference on Robotics and Automation (ICRA)*, pages 856–861. IEEE, 2003.
7. T. Schmitt, R. Hanek, M. Beetz, S. Buck, and B. Radig. Cooperative probabilistic state estimation for vision-based autonomous mobile robots. *IEEE Transactions on Robotics and Automation*, 18(5):670–684, October 2002.