Sensor Modeling Using Visual Object Relation in Multi Robot Object Tracking

Daniel Göhring and Jan Hoffmann

Institut für Informatik
LFG Künstliche Intelligenz
Humboldt-Universität zu Berlin
Unter den Linden 6
10099 Berlin, Germany
`http://www.aiboteamhumboldt.com`

Abstract. In this paper we present a novel approach to estimating the position of objects tracked by a team of mobile robots. Modeling of moving objects is commonly done 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. For multiple robots to cooperate and share information, though, they need to agree on a global, allocentric frame of reference. 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: a) Errors in feature detection are correlated and not assumed independent. Furthermore, 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) As a consequence of the above, it provides a highly efficient method for communicating information about a tracked object and communication can be asynchronous.

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.

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 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 by the fact that the environment is only partially observable to the robot.

G. Lakemeyer et al. (Eds.): RoboCup 2006, LNAI 4434, pp. 279–286, 2007.

In hybrid architectures [1], basic behaviors or skills, such as 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.

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 [8], 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 a typical camera image of a RoboCup environment, the image processing could, for example, extract the following percepts: *ball*, *opponent player*, and *goal*. 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 [7].

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.

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.

Our experiments were conducted on the color coded field of the *Sony Four Legged League* using the Sony Aibo ERS-7.

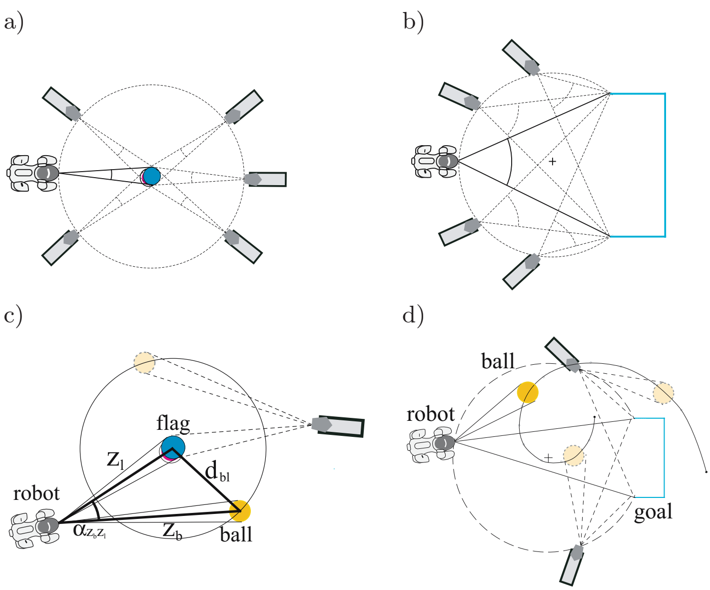

Fig. 1. Single percept: a, b) When a flag or a goal is seen, the robot can calculate its distance to it, but not its exact position, a circle remains for all possible robot positions. Two percepts in one image c, d) a flag/goal and a ball let the robot determine the ball's position relative to the flag/goal; all possible positions of the ball relative to the flag/goal form a circle/spiral arc.

2 Object Relation Information

In a RoboCup game, the robots permanently scan their environment for landmarks as there are flags, goals, and the ball. 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. 1 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 (Fig. 1 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.

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. 1 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. 1 d). But 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 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 [4].

3 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 [9] and in the highly dynamic RoboCup environment [6]. 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. It uses Bayes law and Markov assumption to estimate an object's position. 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. The ball position is modeled relative to the field, which makes it independent from robot motions. 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. Therefore a sensor model is needed, telling the filter how accurate the sensor data are. The localization is being ititialized with $Bel(s_0)$ at $t = t_0$. The particles from the particle set are distributed arbitrarily across the field. Every ball position is equally uncertain. If sensor data is gained, the particle set will be updated and after a few steps converge to a certain area.

3.1 Monte-Carlo Localization, Implementation

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

$$\overrightarrow{s}^i = \begin{pmatrix} q_x^i \\ q_y^i \end{pmatrix} \tag{1}$$

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 [7], 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. In fact, the sensor model is more complex than a Gaussian, but assuming it to be Gaussian showed to be a good approximation. 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 the particle distributions. 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

Table 1. Object distance and angle standard deviations

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

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.

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.

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.

4 Experimental Results

The Aibo ERS-7 robot serves as a test platform for our work. In our experiment, 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 the next step the two robots communicate their particle distribution to each other (or a part of it). 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

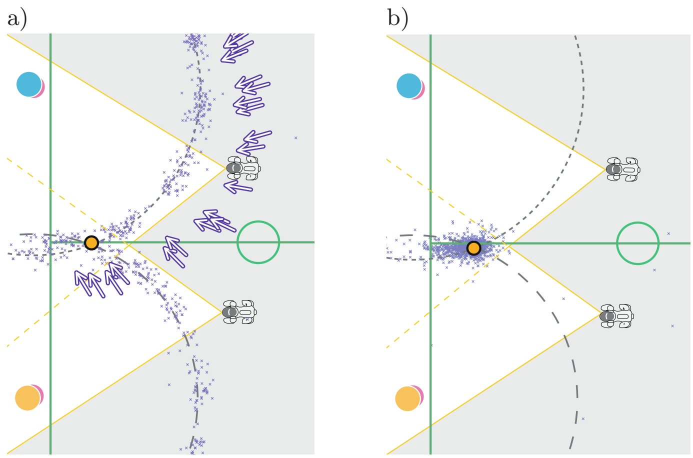

Fig. 2. Experiment with two flags: a) no percept relations communicated, the robots are self localizing (arrows show SL-particles of the upper robot schematically), the ball positions (cloud of dots) are modeled egocentric, transformed into global coordinates, then communicated to the other robot and merged with its 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.

possible, e.g., when every robot can only see one landmark and the ball. 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. 2).

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.

5 Conclusion

Object relations in robot images can be used to localize objects in global coordinates. 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 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. Furthermore, we will investigate how data about commonly modeled objects in field coordinates, e.g., the ball can be used for self localization.

Acknowledgments

Program code used was developed by the GermanTeam, a joint effort of the Humboldt University of Berlin, University of Bremen, University of Dortmund, and the Technical University of Darmstadt. Source code is available for download at http://www.germanteam.org.

References

1. Arkin, R.: Behavior-Based Robotics. MIT Press, Cambridge, MA, USA (1998)
2. Dietl, M., Gutmann, J., Nebel, B.: Cooperative sensing in dynamic environments. In: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'01), Maui, Hawaii (2001)
3. Fox, D., Burgard, W., Dellart, 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. AAAI Press/MIT Press, Cambridge (1999)
4. 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 Computer Society Press, Los Alamitos (2002)
5. 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)
6. Lenser, S., Bruce, J., Veloso, M.: CMPack: A complete software system for autonomous legged soccer robots. In: AGENTS '01: Proceedings of the fifth international conference on Autonomous agents, pp. 204–211. ACM Press, New York (2001)
7. Röfer, T., Jüngel, M.: Vision-based fast and reactive monte-carlo localization. In: Polani, D., Bonarini, A., Browning, B., Yoshida, K. (eds.) Proceedings of the 2003 IEEE International Conference on Robotics and Automation (ICRA), pp. 856–861. IEEE Computer Society Press, Los Alamitos (2003)
8. 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)
9. 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)