Multi-robot Cooperative Localization through Collaborative Visual Object Tracking

Zhibin Liu, Mingguo Zhao, Zongying Shi, and Wenli Xu

Department of Automation, Tsinghua University, Beijing, China
liu-zb04@mails.tsinghua.edu.cn

Abstract. In this paper we present an approach for a team of robots to cooperatively improve their self localization through collaboratively tracking a moving object. At first, we use a Bayes net model to describe the multi-robot self localization and object tracking problem. Then, by exploring the independencies between different parts of the joint state space of the complex system, we show how the posterior estimation of the joint state can be factorized and the moving object can serve as a *bridge* for information exchange between the robots for realizing cooperative localization. Based on this, a particle filtering method for the joint state estimation problem is proposed. And, finally, in order to improve computational efficiency and achieve real-time implementation, we present a method for decoupling and distributing the joint state estimation onto different robots. The approach has been implemented on our four-legged AIBO robots and tested through different scenarios in RoboCup domain showing that the performance of localization can indeed be improved significantly.

1 Introduction

Autonomous robots need to know their own positions within the environment, and the positions of other robots and moving objects in order to complete their tasks individually or in a cooperative way. However, it is not an easy job to accurately estimate the robot's own position as well as the state of the moving objects, because the information that robots receive through their sensors is inherently uncertain, and the control over their actuators is also inaccurate. Additionally, the estimating problem is made more difficult when there are unmodeled interactions or collisions between the robots or the robot haven't seen any distinct landmarks for a long time, which are especially typical in RoboCup domain.

During the soccer games, ball is the focus of robot's attention. Searching for ball, chasing and dribbling the ball and seeking for opportunities to kick a goal are usually the most important tasks of the robots. So, it is often the cases that there are few or no distinct landmarks in robot's sight. As a consequence, odometry errors accumulate as the time goes by without compensation, and the accuracy and reliability of localization result is seriously affected. However, just as mentioned above, the ball is usually in the sight of the robots. If there are some ways to improving the robots' self localization based on the ball information, much better performance can be expected.

U. Visser et al. (Eds.): RoboCup 2007, LNAI 5001, pp. 41–52, 2008.

Considering all of the factors mentioned above, we enable the robots to share information and improve their self localization cooperatively by making them track the moving objects collaboratively and then refine their self localization results based on the common knowledge of the objects. We implement this idea on a team of four-legged AIBO robots to collaboratively track and estimate the state of a ball and use the ball information to improve their self localization simultaneously.

This paper is organized as follows. After introducing the related works in the next section, we present our multi-robot cooperative localization and ball tracking method in Section 3. Experimental results are given in Section 4, followed by conclusions drawn in Section 5.

2 Related Work

In Recent years, multi-robot cooperative localization has received increasing attention in robotics community. Most of the works on this problem are based on the assumption that the robots have abilities to detect and identify each other and estimate their relative positions [1, 2, 3, 4, 5]. They usually requires sophisticated image processing methods or adding artificial marks onto the robot platform. However, these may not be granted in many cases, especially in RoboCup competitions. Because adding distinctly colored marks to the AIBO robots is not allowed by rules, so it is quite difficult to identify the robots or accurately estimate their relative positions, due to the irregular and complex shape of the robot.

To our knowledge, the first work using moving objects' information to improve the robots' self localization is [6], in which Schmitt et al presented a method for enabling a team of robot to estimate their joint positions in a known environment and track the positions of autonomously moving objects (e.g., the ball). By using the ball's position estimations received from the other robots to correct the robot's own pose, the state estimators of different robots can cooperate to increase the accuracy and reliability of the estimation process. But this method is based on Kalman filtering, which is inefficient to track multiple ball hypotheses in face of false positive ball detection and sensor noises. In [7], Kwok and Fox presented a Rao-Blackwellised particle filtering method for estimating the robot's self location as well as the ball state. It provides a powerful model for multiple model object tracking and also allows the robot to infer where it is by observing the ball. However, the cooperative localization or object tracking problem are not discussed in their work. In another most recent work [8], Göhring presented an approach to estimate the position of objects tracked by a team of mobile robots by using the spatial relation of the objects respect to stationary landmarks detected in the same camera images, and then use these objects for better self localization. Though the objects' position estimation resulted by this method is robust to the localization errors of the robots, it requires that each robot can detect the ball as well as some landmarks at the same time. Moreover, only the static object model is considered in their work.

3 Multi-robot Cooperative Localization through Collaborative Object Tracking Using Particle Filters

In this section, we will first describe the multi-robot localization and object (ball) tracking problem using Bayes net. Then, through formal analysis, we will show how this joint state estimation problem can be factorized and tackled using particle filters. Finally, we conclude that the moving ball can serves as a bridge to realize cooperative localization, and an efficient distributed implementation method is presented.

3.1 Problem Description Using Bayes Net

Without loss of generality, we consider a system consisting of a pair of robots and a ball. Let $< b_k, r_k^1, r_k^2 >$ denote the state of the system at time *k*. $b_k =< x_b, y_b, \dot{x}_b, \dot{y}_b, m_b >$ denotes the state of the ball in global coordinates, where $x_b, y_b, \dot{x}_b, \dot{y}_b$ represent ball location and velocity and $m_b \in \{0,1,2\}$ indicates the interaction model of the ball and robots. $m_b = 0$ means the ball is not grabbed by any teammate of the robots, while $m_b = 1$ or 2 indicates that the ball is grabbed by robot 1 or 2 respectively. $r_k^j =< x_r^j, y_r^j, \theta_r^j >$, $j = 1, 2$, is the robot location and orientation on the field. Moreover, denote the observations of the ball and landmarks made by robot *j* as z_k^j, which is provided in relative bearing and distance.

A graphical model description of the state estimation problem of the system is given in Fig.1, where the nodes represent different random variables and the arrows indicate dependencies between these variables. The model shows the following relationships:

1) Robot-*j*'s location at time *k*, r_k^j, only depends on the previous location r_{k-1}^j and the robot motion control u_{k-1}^j.
2) The observations z_k^j consist of $z_k^{j,L}$ and $z_k^{j,B}$, which describe landmark observations and ball observations respectively. $z_k^{j,L}$ only depend on the current robot location r_k^j (since the map of field is given); relative ball observations $z_k^{j,B}$ only depend on the current ball and robot positions.
3) The location, velocity and interaction model of the ball b_k typically depend on the previous ball state b_{k-1}, the actions of all robots, u_{k-1}^1, u_{k-1}^2, and the robots location r_k^1, r_k^2. However, just as the dashed arrows indicate, the existence of the relationship between robot location, motion control and ball state depends on which robot grab the ball, i.e. the component m_b in b_k. For example, if $m_b = 1$, i.e. ball is grabbed by robot 1, then the ball location is tightly attached to the robot location r_k^1 and the arrow from r_k^1 to b_k exists.

3.2 Factorizing the Joint State Space Posterior of Multi-robot Cooperative Localization and Object Tracking

Since the dependencies between different parts of the joint state space are defined based on Bayes net description, we can address the problem of filtering, which aims to compute the posterior over the joint state vector $< b_k, r_k^1, r_k^2 >$ conditioned on all sensor measurements obtained so far, i.e. to compute:

$$p(b_k, r_k^1, r_k^2 \mid z_{1:k}^1, u_{0:k-1}^1, z_{1:k}^2, u_{0:k-1}^2) \tag{1}$$

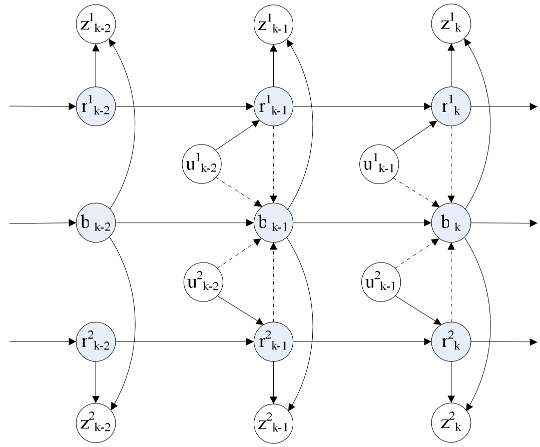

Fig. 1. Bayes net for multi-robot localization and ball tracking. The nodes in this graph represent the different parts of the dynamic system at consecutive time instances, and the edges represent dependencies between the individual parts of the state space. Filled circles indicate system state variable nodes, while the other circles stand for observations and motion control.

Based on the posterior estimation resulted from previous step, (1) can be written in a recursive form:

$$p(b_k, r_k^1, r_k^2 \mid z_{1:k}^1, u_{0:k-1}^1, z_{1:k}^2, u_{0:k-1}^2) = \iiint\limits_{b_{k-1}, r_{k-1}^1, r_{k-1}^2} p(b_k, r_k^1, r_k^2 \mid b_{k-1}, r_{k-1}^1, r_{k-1}^2, z_k^1, u_{k-1}^1, z_k^2, u_{k-1}^2) \cdot p(b_{k-1}, r_{k-1}^1, r_{k-1}^2 \mid z_{1:k-1}^1, u_{0:k-2}^1, z_{1:k-1}^2, u_{0:k-2}^2) db_{k-1} dr_{k-1}^1 dr_{k-1}^2 \quad (2)$$

The second term in (2) is the previous posterior, and the first term can be further factorized by employing the dependencies and independencies described in Bayes net model presented above. First, it can be factorized as:

$$p(b_k, r_k^1, r_k^2 \mid b_{k-1}, r_{k-1}^1, r_{k-1}^2, z_k^1, u_{k-1}^1, z_k^2, u_{k-1}^2) = p(b_k \mid r_k^1, r_k^2, b_{k-1}, r_{k-1}^1, r_{k-1}^2, z_k^1, u_{k-1}^1, z_k^2, u_{k-1}^2) p(r_k^1, r_k^2 \mid b_{k-1}, r_{k-1}^1, r_{k-1}^2, z_k^1, u_{k-1}^1, z_k^2, u_{k-1}^2) \quad (3)$$

Since when $r_k^1, r_k^2, b_{k-1}, u_{k-1}^1, u_{k-1}^2$ are given b_k can be determined, (3) can be written as:

$$p(b_k, r_k^1, r_k^2 \mid b_{k-1}, r_{k-1}^1, r_{k-1}^2, z_k^1, u_{k-1}^1, z_k^2, u_{k-1}^2) = p(b_k \mid r_k^1, r_k^2, b_{k-1}, z_k^1, u_{k-1}^1, z_k^2, u_{k-1}^2) p(r_k^1, r_k^2 \mid b_{k-1}, r_{k-1}^1, r_{k-1}^2, z_k^1, u_{k-1}^1, z_k^2, u_{k-1}^2) \quad (4)$$

Then, according to Bayes rule, we have:

$$p(b_k, r_k^1, r_k^2 \mid b_{k-1}, r_{k-1}^1, r_{k-1}^2, z_k^1, u_{k-1}^1, z_k^2, u_{k-1}^2) \propto p(b_k \mid r_k^1, r_k^2, b_{k-1}, z_k^1, u_{k-1}^1, z_k^2, u_{k-1}^2) p(z_k^1, z_k^2 \mid r_k^1, r_k^2, b_{k-1}, r_{k-1}^1, r_{k-1}^2, u_{k-1}^1, u_{k-1}^2) \cdot p(r_k^1, r_k^2 \mid b_{k-1}, r_{k-1}^1, r_{k-1}^2, u_{k-1}^1, u_{k-1}^2) \quad (5)$$

Since r_k^1 only depends on r_{k-1}^1, u_{k-1}^1, and r_k^2 only depends on r_{k-1}^2, u_{k-1}^2, the rightmost term in (5) can be factorized as:

$$p(r_k^1, r_k^2 \mid b_{k-1}, r_{k-1}^1, r_{k-1}^2, u_{k-1}^1, u_{k-1}^2) = p(r_k^1 \mid r_{k-1}^1, u_{k-1}^1)\, p(r_k^2 \mid r_{k-1}^2, u_{k-1}^2) \tag{6}$$

Exploiting the dependencies in the graph model, we know that z_k^1, z_k^2 are conditional independent from r_{k-1}^1, r_{k-1}^2, so the second term in (5) can be written as:

$$p(z_k^1, z_k^2 \mid r_k^1, r_k^2, b_{k-1}, r_{k-1}^1, r_{k-1}^2, u_{k-1}^1, u_{k-1}^2) = p(z_k^1, z_k^2 \mid r_k^1, r_k^2, b_{k-1}, u_{k-1}^1, u_{k-1}^2) \tag{7}$$

Substituting (6) and (7) into (5), we have:

$$\begin{aligned}
& p(b_k, r_k^1, r_k^2 \mid b_{k-1}, r_{k-1}^1, r_{k-1}^2, z_k^1, u_{k-1}^1, z_k^2, u_{k-1}^2) \\
& \propto p(b_k \mid r_k^1, r_k^2, b_{k-1}, z_k^1, u_{k-1}^1, z_k^2, u_{k-1}^2)\, p(z_k^1, z_k^2 \mid r_k^1, r_k^2, b_{k-1}, u_{k-1}^1, u_{k-1}^2) \cdot \\
& \qquad\qquad\qquad\qquad p(r_k^1 \mid r_{k-1}^1, u_{k-1}^1)\, p(r_k^2 \mid r_{k-1}^2, u_{k-1}^2) \\
& = p(b_k, z_k^1, z_k^2 \mid r_k^1, r_k^2, b_{k-1}, u_{k-1}^1, u_{k-1}^2)\, p(r_k^1 \mid r_{k-1}^1, u_{k-1}^1)\, p(r_k^2 \mid r_{k-1}^2, u_{k-1}^2) \\
& = p(z_k^1, z_k^2 \mid r_k^1, r_k^2, b_k)\, p(b_k \mid r_k^1, r_k^2, b_{k-1}, u_{k-1}^1, u_{k-1}^2)\, p(r_k^1 \mid r_{k-1}^1, u_{k-1}^1)\, p(r_k^2 \mid r_{k-1}^2, u_{k-1}^2) \\
& = p(z_k^1 \mid r_k^1, b_k)\, p(z_k^2 \mid r_k^2, b_k)\, p(b_k \mid r_k^1, r_k^2, b_{k-1}, u_{k-1}^1, u_{k-1}^2)\, p(r_k^1 \mid r_{k-1}^1, u_{k-1}^1)\, p(r_k^2 \mid r_{k-1}^2, u_{k-1}^2)
\end{aligned} \tag{8}$$

Substituting (8) into (2) we get:

$$\begin{aligned}
& p(b_k, r_k^1, r_k^2 \mid z_{1:k}^1, u_{0:k-1}^1, z_{1:k}^2, u_{0:k-1}^2) \\
& \propto p(z_k^1 \mid r_k^1, b_k)\, p(z_k^2 \mid r_k^2, b_k) \iiint_{b_{k-1}, r_{k-1}^1, r_{k-1}^2} p(b_k \mid r_k^1, r_k^2, b_{k-1}, u_{k-1}^1, u_{k-1}^2)\, p(r_k^1 \mid r_{k-1}^1, u_{k-1}^1) \cdot \\
& \qquad p(r_k^2 \mid r_{k-1}^2, u_{k-1}^2)\, p(b_{k-1}, r_{k-1}^1, r_{k-1}^2 \mid z_{1:k-1}^1, u_{0:k-2}^1, z_{1:k-1}^2, u_{0:k-2}^2)\, db_{k-1} dr_{k-1}^1 dr_{k-1}^2
\end{aligned} \tag{9}$$

It is clearly shown in equation (8) that, the variable b_k (ball) serves as a linkage between the states of the robots, r_k^1 and r_k^2, which allows the information flow to travel from one robot to another and vice versa to achieve cooperative localization.

3.3 Particle Filtering for Joint Estimation

To implement the idea presented in the previous subsection, we have to specify the representation of the posterior distribution. We utilize particle filtering, which represent posteriors by sets of weighted samples, or particles:

$$S_k = \{< s_k^{(i)}, w_k^{(i)} > \mid 1 \le i \le N\}$$

where each particle $s_k^{(i)} = < b_k^{(i)}, r_k^{1(i)}, r_k^{2(i)} >$ and N is the total number of samples. The task is to generate samples distributed according to (1) based on the samples drawn from the posterior at k-1, denoted by S_{k-1}. We generate the different components of $s_k^{(i)}$ stepwise according to (8). In the first step, a sample $s_{k-1}^{(i)} = < b_{k-1}^{(i)}, r_{k-1}^{1(i)}, r_{k-1}^{2(i)} >$ is drawn from S_{k-1}, and then we draw new robot pose $r_k^{1(i)}$ and $r_k^{2(i)}$ for robot 1 and robot 2 respectively, according to:

$$r_k^{1(i)} \sim p(r_k^{1(i)} \mid r_{k-1}^{1(i)}, u_{k-1}^1) \tag{10}$$

$$r_k^{2(i)} \sim p(r_k^{2(i)} \mid r_{k-1}^{2(i)}, u_{k-1}^2) \tag{11}$$

This gives us $s_k^{(i)} =< \_, r_k^{1(i)}, r_k^{2(i)} >$, where _ denotes uninitialized value. Then, the sample's ball state $b_k^{(i)}$ is estimated:

$$b_k^{(i)} \sim p(b_k^{(i)} \mid r_k^{1(i)}, r_k^{2(i)}, b_{k-1}^{(i)}, u_{k-1}^1, u_{k-1}^2) \tag{12}$$

Finally, the importance weight of the sample $w_k^{(i)}$ is calculated as:

$$w_k^{(i)} = \eta \cdot p(z_k^1 \mid r_k^{1(i)}, b_k^{(i)}) p(z_k^2 \mid r_k^{2(i)}, b_k^{(i)}) \tag{13}$$

where η is a normalizing factor which ensures all of the importance weights sum up to 1. Note that, since the observations z_k^j are composed of landmarks detection $z_k^{j,L}$ and ball detection $z_k^{j,B}$, equation (13) can be further factorized as:

$$\begin{aligned} w_k^{(i)} &= \eta \cdot p(z_k^{1,L}, z_k^{1,B} \mid r_k^{1(i)}, b_k^{(i)}) p(z_k^{2,L}, z_k^{2,B} \mid r_k^{2(i)}, b_k^{(i)}) \\ &= \eta \cdot p(z_k^{1,L} \mid r_k^{1(i)}) p(z_k^{1,B} \mid r_k^{1(i)}, b_k^{(i)}) p(z_k^{2,L} \mid r_k^{2(i)}) p(z_k^{2,B} \mid r_k^{2(i)}, b_k^{(i)}) \end{aligned} \tag{14}$$

where the facts that, when the robots' pose $r_k^{j(i)}$ and ball state $b_k^{(i)}$ are given the landmarks detection and ball detection are independent, and the landmark observation only depends on the robot location (as the map of the environment is already known), are used.

3.4 Distributed Implementation

There are different ways to implement our multi-robot cooperative localization and ball tracking method. The most intuitive one is to make every robot maintain and estimate the full joint state vector $< b_k, r_k^1, r_k^2 >$. But, unfortunately, it requires a large amount of particles to achieve satisfying estimation result, due to the high dimension of the joint state space. As the members of the robots increase, this problem becomes more serious. It will be computationally too demanding for the AIBO robots.

Here we present a distributed method in which each robot only have to estimate its own self location and ball state, then through communication the information are shared and cooperative localization and ball tracking is achieved.

Introducing two affiliated factors b_k^1 and b_k^2, which correspond to the ball estimation made by robot 1 and 2 respectively, the third term in equation (9) can be transformed as:

$$
\begin{aligned}
&p(b_k \mid r_k^1, r_k^2, b_{k-1}, u_{k-1}^1, u_{k-1}^2) \\
&= \iint_{b_k^1, b_k^2} p(b_k, b_k^1, b_k^2 \mid r_k^1, r_k^2, b_{k-1}, u_{k-1}^1, u_{k-1}^2) db_k^1 db_k^2 \\
&= \iint_{b_k^1, b_k^2} p(b_k \mid b_k^1, b_k^2) p(b_k^1, b_k^2 \mid r_k^1, r_k^2, b_{k-1}, u_{k-1}^1, u_{k-1}^2) db_k^1 db_k^2 \\
&= \iint_{b_k^1, b_k^2} p(b_k \mid b_k^1, b_k^2) p(b_k^1 \mid r_k^1, b_{k-1}, u_{k-1}^1) p(b_k^2 \mid r_k^2, b_{k-1}, u_{k-1}^2) db_k^1 db_k^2
\end{aligned} \tag{15}
$$

This is attractive, since it allows each robot to estimate the ball state individually and then through an information fusion process the *team* ball estimation b_k is obtained.

Now, we present our method for performing the joint estimation in a distributed form: first each robot only estimate the joint state vector $< b_k^j, r_k^j >$, i.e. the individual ball state and self location, based on its own observations; then they send their estimation results to their teammates as well as receive the information coming from their teammates; whereafter, the team ball state b_k is estimated and the partial joint state $< b_k, r_k^j >$ maintained by each robot is finally updated.

Additionally, we enable each robot to use two kinds of ball model: egocentric ball model and global ball model. The egocentric ball model represents the ball state in robot-centric coordinate. It is more robust against global localization errors, and its uncertainty is much smaller than global ball state so that fewer particles are needed to represent its probabilistic distribution. The global ball model represents the ball state in global allocentric reference coordinate, which is used for communicating information to other robots. By associating egocentric ball state with robot's self location, the global ball state can be calculated. And the global ball estimation resulting from all robots are fused to get the *team* ball estimation. It is none other but this team ball estimation that enables the robots to act harmoniously and position themselves strategically on the field, and further to improve their self localization cooperatively.

Suppose, for any robot *j*, we use $n_{b,L}$ particles ${}^l b_k^{j(i)}$, $i \in [1, n_{b,L}]$ to represent the probabilistic distribution of egocentric ball state, and n_r particles $r_k^{j(\tau)}$, $\tau \in [1, n_r]$ for self localization. The procedure of the cooperative localization and ball tracking algorithm running on each robot *j* is as follows:

1) **Predict self location:** generate robot pose $r_k^{j(\tau)} \sim p(r_k^{j(\tau)} \mid r_{k-1}^{j(\tau)}, u_{k-1}^j)$;
2) **Update self localization using landmark measurement:** if any landmark is detected, the weights $w_k^{j(\tau)}$ of the samples $r_k^{j(\tau)}$ are calculated as $w_k^{j(\tau)} = p(z_k^{j,L} \mid r_k^{j(\tau)}) \cdot w_{k-1}^{j(\tau)}$, if the sum of the weights is smaller than a given threshold, substitute the low-weight samples by new samples randomly drawn according to the observations (similar to the sensor resetting method presented in [9]); else, if no landmark is detected, go to next step;
3) **Predict egocentric ball state:** if the ball is grabbed by robot *j*, the relative position of the ball ${}^l x_b^{j(i)}$, ${}^l y_b^{j(i)}$ in all particles ${}^l b_k^{j(i)}$ are set to zero; else, if the

ball is not grabbed by robot j, the state of the ball is predicted as ${}^{l}b_k^{j(i)} \sim p({}^{l}b_k^{j(i)} \mid {}^{l}b_{k-1}^{j(i)}, u_{k-1}^{1})$, and the weights of these particles are set to be equal;

4) **Egocentric ball update:** *a)* if the ball is neither seen nor grabbed by the robot, go to the final step; *b)* if the ball is grabbed, go to next step; *c)* if the ball is seen, update the weights of egocentric ball particles as ${}_{b}^{l}w_k^{j(i)} = p(z_k^{j,B} \mid {}^{l}b_k^{j(i)}) \cdot {}_{b}^{l}w_{k-1}^{j(i)}$, and then normalize these weights;
5) **Generate robot pose hypotheses:** calculate robot pose hypotheses by clustering the particles $r_k^{j(\tau)}$ of the self location, and then pick out n_h three robot pose hypotheses with the highest probabilities;
6) **Generate ball particles in the global coordinate:** associate the $n_{b,L}$ egocentric ball particles ${}^{l}b_k^{j(i)}$ with each of the robot pose hypotheses resulting from the last step to generate $n_h \times n_{b,L}$ particles in global coordinate ${}^{g}b_k^{j(i)}$; calculate the weights ${}_{b}^{g}w_k^{j(i)}$ of ${}^{g}b_k^{j(i)}$ by multiplying the weights of the egocentric ball particles and the probability of the robot pose hypotheses;
7) **Subsample global ball particles to obtain representative particles:** in this step we follow the method presented in [10], i.e. first the soccer field is recursively split into cells to form a quad-tree with a maximum depth of $d_{\max}$; then for each cell a representative particle is calculated as the weighted average of the particles contained in that cell, and the weight of the representative particle is the sum-weight of the involved particles; finally, the n_{rep} representative particles with the highest weights are chosen with their weights normalized;
8) **Send/receive representative ball particles to/from teammates:** representative global ball particles resulting from the last step are sent to/received from the teammates through wireless communication;
9) **Calculate the entropy of robot pose estimation:** based on the particles and weights resulting from step 1) and 2), the entropy of the robot pose is calculated as a metric of the underlying uncertainty in the pose estimation;
10) **Estimate team ball location hypotheses based on the fused information:** *a)* if the entropy of robot pose is within a certain range, go to the final step; *b)* otherwise, if the entropy is higher than the given threshold, the robot will calculate the global ball position hypotheses by utilizing the received representative particles together with its own representative particles; these $n_{rep} \times n$ particles (n is the total number of robots) are classified into clusters following a clustering method similar to step 5), and the location hypothesis with the highest probability b_k^t is selected out;
11) **Update self localization using team ball estimation:** update the particle set representing the robot pose (resulting from step 1) and 2)) by calculating the weights as $w_k^{j(\tau)} \leftarrow p(z_k^{j,B} \mid r_k^{j(\tau)}, b_k^t) \cdot w_k^{j(\tau)}$; if the sum-weight of particles is smaller than a given threshold, substitute some of the lowest-weight samples by new samples drawn according to the team ball location b_k^t and ball observation $z_k^{j,B}$ (similar to the method used in Step *2)*);

12) **Particle weight normalization:** at this final step of iteration, the weights of the particles representing the robot pose $w_k^{j(\tau)}$ are normalized ensuring them sum up to 1.

4 Experiments and Results

To verify the effectiveness of the multi-robot cooperative localization and ball tracking method, we conduct experiments on Sony AIBO robots on the field of RoboCup Soccer 4-legged League. Our method is compared with the reference method presented in [11], which has been adopted by more than 6 different teams in RoboCup Soccer 4-legged League and its source code is publicly available.

We set up two scenarios in our experiments, both of which went on RoboCup 2006 Four-legged League soccer field. Throughout all of the experiments, the rule that '*the robot should not carry ball for longer than 3 seconds at one time*' is obeyed. The parameters in the algorithm presented in the previous section are set as: $n_{b,L}$ = 40, n_r = 100, n_h = 3, $d_{\max}$ = 6, n_{rep} = 12.

4.1 Scenario *A*: 1 Team of 4 Robots, RoboCup 2006 Field

In our first test scenario, a team of 4 robots are placed on the field, without opponents or other obstacles. This scenario represents a "best case" scenario to evaluate the performance of the two localization methods, because there is no collision between the robots, and the chances that the ball be occluded from the sight of the robots are smaller. During the experiment, robots NO.1~ NO.3 are expected to stay at the fixed points on the field (shown as the small solid black squares in Fig. 2(a)). They concentrate on tracking the ball, but also have to periodically distract their attention from it in order to see the landmarks and localize themselves. The positions of these three robots keep not changed, but their orientations can be adjusted by themselves so as to face directly to the moving ball and keep tracking of it. Robot NO.4 (its localization results are examined) can walk freely, chase the ball and carry ball to go toward 5 appointed locations (the small solid red squares labeled L#1~L#5 in Fig. 2(a)) sequentially. When the robot gets quite near to an appointed location, the experimenter would tap the back button on the robot manually so as to conduct it to change its destination and go to the next appointed location.

We compare the performance of the reference method with our cooperative localization method by running them in parallel on the robots and making them process exactly the same sensor data. The entropy [12, 13] of robot NO.4's pose estimations resulting from the two methods are automatically recorded in a log file one time per second by the robot. The ground truth of robot positions are obtained as follows: every ten seconds, the robot and localization algorithms pause; the real position of the robot is measured manually with the current localization results of the two methods recorded; then, by tapping the head button of the robot manually, it continue to move.

Fig.2 depicts the results for this scenario. At beginning, robot NO.4 was placed at the start point (small red solid circle in Fig. 2(a)), then every ten seconds its real position was recorded (magenta ☆ in Fig. 2(a)). The estimated positions of our

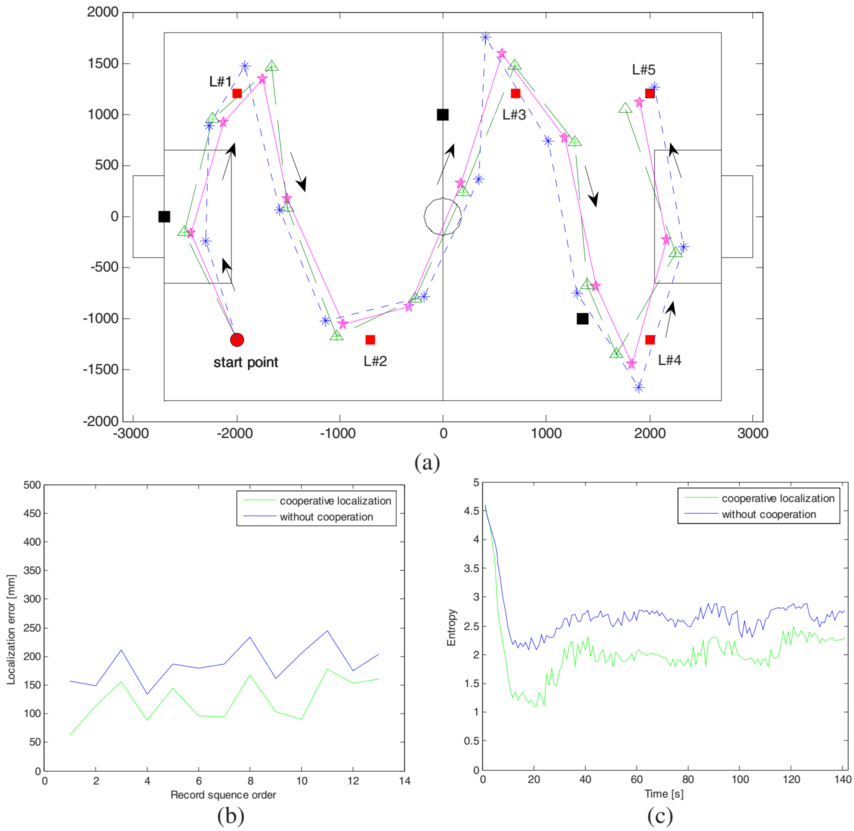

Fig. 2. Results for scenario A: (a) robot's real positions and the estimated positions of the two methods. (b) Localization errors. (c) Entropy of pose estimations at different time instances.

method and reference method are shown by green △ and blue * respectively. Note that, the colored lines linking the recorded positions in Fig. 2(a) do not stand for the trajectories, but only show the sequential order of the positions.

The localization errors, which are measured by the distance between the real positions and estimated positions, are shown in Fig. 2(b). In Fig. 2(c), the entropy of pose estimation resulting from the two methods is visualized. It is clear that both the localization errors and entropy of our cooperative localization method are significantly smaller than that of the reference method.

4.2 Scenario *B*: Real Game, 2 Teams of 6 Robots, RoboCup 2006 Field

This scenario aims to deal with the real game situation: two teams of robots play competitively on a standard RoboCup 2006 field. Through this scenario, we can examine that to what level our cooperative localization method can promote the performance of robot's self localization.

Since we only have 6 AIBO robots at hand, we can only assign 3 members for each team. Moreover, because it is a real soccer game, the ground truth of robots' positions are difficult to measure manually. But, the entropy of robot pose is a useful metric to measure the uncertainty of the robot's state. So, in this scenario, we evaluate the performance of the two methods by focusing on comparing the resulted entropy.

The experiment lasted for 5 minutes. Each team has 3 robots: goalie, defender and attacker. Since the attacker is the most active role in the team, it has more chances to collide with opponent robots when chasing the ball or seeking for opportunity to shoot. So, its self localization results can to some extent provide a "worst case" scenario for localization algorithms' performance. Therefore, we recorded the red attacker's pose estimation entropy during the game. The entropy was written into a log file by the robot at a rate of one record per second. Fig.3 depicts the results.

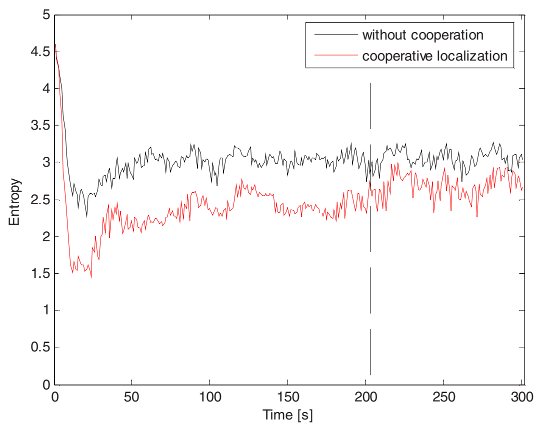

Fig. 3. Entropy of pose estimation at different time instances in Scenario B

It is clear that our cooperative localization method outperforms the reference method again. And, by examining the result carefully, we found that the difference lying in the performances of the two methods becomes less significant after the time instance labeled by the dashed line in Fig.3. This is due to the fact that there were more collisions between the attacker and the opponent robots and the ball was usually occluded by the robots. There were fewer chances for the red attacker's teammates to see the ball and provide accurate *team ball* estimation. So, the improvement made by utilizing ball information to promote self localization was affected, and became less significant. This is reasonable and consistent with our common knowledge.

5 Conclusion

In this paper we presented a probabilistic method for multi-robot cooperative localization and object tracking. By viewing the object and robots as a whole system, a Bayes net model is established to describe the joint state estimation problem. Then, through exploring the independences between different parts of the state space, we show how the posterior estimation of the joint state can be factorized and tackled using a particle filtering method. Finally, in order to improve computational efficiency and achieve real-time implementation, we distributed the joint state estimation task to different robots: first, each of the robot estimate their self location and ball state based on their own sensor data; then, by exchanging information between the robots the ball state estimation is refined; at last, each robot use the refined ball state estimation to correct their self localization.

By utilizing the proposed method, the state estimation modules of different robots can cooperate to increase the accuracy and reliability of their self localization and ball state estimation. It is capable of dealing with multiple hypotheses lying in the state of both the ball and robots. The experimental results show that the proposed method is effective and can evidently improve the robots' self localization in RoboCup domain.

References

1. Fox, D., Burgard, W., Kruppa, H., Thrun, S.: A Probabilistic Approach to Collaborative Multi-Robot Localization. Autonomous Robots 8, 325–344 (2000)
2. Kurazume, R., Hirose, S.: An Experimental Study of a Cooperative Positioning System. Autonomous Robots 8, 43–52 (2000)
3. Roumeliotis, S.I., Bekey, G.A.: Distributed Multirobot Localization. IEEE Transactions on Robotics and Automation 18(5), 781–795 (2002)
4. Madhavan, R., Fregene, K., Parker, L.E.: Distributed Cooperative Outdoor Multirobot Localization and Mapping. Autonomous Robots 17, 23–39 (2004)
5. Howard, A.: Multi-robot Simultaneous Localization and Mapping using Particle Filters. International Journal of Robotics Research 25, 1243–1256 (2006)
6. Schmitt, T., Hanek, R., Beetz, M., et al.: Cooperative Probabilistic State Estimation for Vision-based Autonomous Mobile Robots. IEEE Transactions on Robotics and Automation 18(5), 670–684 (2002)
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. Göhring, D., Burkhard, H.D.: Multi Robot Object Tracking and Self Localization Using Visual Percept Relations. In: Proc. of the 2006 IEEE International Conference on Intelligent Robots and Systems (IROS 2006), Beijing, China (2006)
9. Lenser, S., Veloso, M.: Sensor Resetting Localization for Poorly Modelled Mobile Robots. In: Proc. of the IEEE International Conference on Robotics and Automation (ICRA 2000), San Francisco (2000)
10. Nisticò, W., Hebbel, M., Kerkhof, T., Zarges, C.: Cooperative Visual Tracking in a Team of Autonomous Mobile Robots. In: 10th International Workshop on RoboCup 2006 (Robot World Cup Soccer Games and Conferences) (2006)
11. Röfer, T., et al.: GermanTeam RoboCup 2005. Technical report (2005), `http://www.germanteam.org/GT2005.pdf`
12. Fox, D., Burgard, W., Thrun, S.: Active Markov Localization for Mobile Robots. Robotics and Autonomous Systems 25(3-4), 195–207 (1998)
13. Hoffmann, J., Spranger, M., Gohring, D., Jungel, M., Burkhard, H.: Further studies on the use of negative information in mobile robot localization. In: Proc. of the IEEE International Conference on Robotics and Automation (ICRA 2006), Orlando,Florida (2006)