Cooperative Global Tracking Using Multiple Sensors

Roman Marchant, Pablo Guerrero, and Javier Ruiz-del-Solar

Department of Electrical Engineering,
Universidad de Chile,
Avenida Tupper 2007, Casilla 412-3, Santiago, Chile
{romarcha,pguerrer,jruizd}@ing.uchile.cl
http://www.die.uchile.cl/

Abstract. Multi-robot systems are increasingly present in nowadays applications. In order to allow an effective decision making, a reliable world representation is required. In a team of robots performing a given task, it is beneficial to share information about the world. In this work, a multi-object, multi-sensor and cooperative tracking method is proposed for the Robocup Standard Platform League (SPL), where two teams of humanoid robots play soccer against each other. Each robot is equipped with two low-cost, noisy, and narrowed-field-of-view cameras and two noisy sonar sensors. In addition, they are endowed with a wireless communication hardware. The on-board computer is a low capacity processing unit (x86 500[Mhz]). The proposed tracking system uses all these hardware elements and it is distributed, in the sense that it is executed in every robot. The proposed tracking system is validated in simulations and in real experiments. Main results show an important improvement on simulated and real results when tracking mobile-objects.

Keywords: Multi-robot tracking, Kalman Filter.

1 Introduction

Modeling a dynamic environment, i.e. determining the spacial location of the objects of interest, is one of the central challenges in mobile robotics. The existence of errors in the model of the environment may lead to mistaken actions. Therefore, the reliability of this model is of vital importance for making correct decisions.

Multi-robot systems are increasingly present in real applications. For instance, mining, agriculture and human-care require that multiple robots share information in order to succeed. Due to the small portion of the environment perceived by every robot, sharing sensory information may allow to expand largely the accessible portion of the environment.

The here-addressed problem is the multi-robot estimation of the state of multiple objects in a dynamic environment. The focus of this work is to define a cooperative tracking methodology and a matching algorithm. The complexity

X. Chen et al. (Eds.): RoboCup 2012, LNAI 7500, pp. 310–321, 2013.

of this problem depends strongly on the application being addressed, because the environment and the robot hardware determine how hard the tracking of multiple objects in a particular scene is. The most significant aspects of the environment that determine the complexity of the problem are the number of objects, their velocity variability and the size of the arena. While the robot hardware variables that affect are the range, precision and field-of-view of the robot sensory hardware, as well as the computational capabilities of the robot. In addition, the precision of the robot movements are important for odometry calculation.

In this work, a multi-object, multi-sensor and cooperative tracking method is proposed for a Robocup *Standard Platform League* (SPL) environment. The estimation of the state of mutiple objects (in this case the poses of the robots and the ball) is of central importance in the SPL since it allows the existence of complex behaviors such as making passes that consider the positions of the partners and opponents, avoiding robots that are not being currently perceived. The proposed tracking system is distributed, in the sense that it is executed in every robot.

The main contribution of this paper is that it proposes a cooperative multi-object, multi-sensor and high-rate tracking methodology for a real application, were the robots have a low capacity processing unit.

This paper is organized as follows: First, in section 2, some related work is reviewed. Then, the proposed methodology for tracking mobile objects is described on section 3. Section 4 presents experimental results on a simulated and a real robot. Finally, section 5 draws some conclusions and recommends future work.

2 Related Work

In the past decades, vast works have addressed cooperative state estimation in multi-robot systems. Those works have been implemented on a wide variety of applications, such as UAVs [1, 2], robot soccer [3–9], water vehicles [10] and general purpose robotics [11–13].

Regarding cooperative state estimation, it is possible to differentiate between two main areas of research. The first one is the cooperative tracking of one or more interesting objects in a scene [1–5, 7–9, 11–13], and the second area is focused on improving the localization of a mobile robot by adding other robots perceptions as inputs to the localization module [3, 6, 10].

Our work is very similar to [9], which explores deeply the areas of coordination and cooperation in multi-robot systems. Particularly on the mobile-object cooperative-tracking area, a classical Kalman filter approach is used. Problems such as measurement delay, distributed implementation, clock synchronization and how external information influence local estimations are addressed in [9]. The main limitation of that system is that it is designed and tested for an omni-directional perception which greatly simplifies the matching problem.

A similar approach is addressed by [4] and [8]. In these works, robots and obstacles are detected using a laser range finder. On the other hand, the ball is

detected using visual perceptions. These works present the idea of multi-object tracking using a *Kalman Filter* for every object on the scene, although results are only focused on the ball tracking and no results are presented for robot tracking. The main drawback is the use of a central computer that makes an estimation of every robot and then sends the fused information back to all team members.

The main difference of the here-proposed global tracking methodology is that it is designed and implemented in an humanoid robot soccer platform, were narrowed field-of-view cameras are used, which reduces the number of perceptions to any object. Furthermore, it merges the information from sonars, cameras and other robots in a distributed manner. Another important difference is that in this work an evaluation of the proposed methodologies, using a laser-based ground-truth system, is presented. Moreover, a quantitative comparison between different tracking approaches is shown.

3 Global Tracking Methodology

3.1 Framework

The methodology is designed to operate in a robot soccer environment, although it may be generalized to other environments. Teams of soccer robots have a group goal, therefore the chosen application exploits the need of cooperation between robots. The teams in the SPL use Nao humanoid robots [14], which are equipped with two noisy sonar sensors and two low cost cameras. Narrowed-field-of-view cameras perceive objects with low frequency, therefore the sonar sensors are used as a supporting feature when no camera perceptions are available for an object.

In robot soccer as in most common situations, a global static origin, $\boldsymbol{O}$, may be defined. Several objects are present and some of them may be used as landmarks to infer the auto-localization, such as goals, lines and corners among others.

Objects may be classified depending on their pose behavior through time relative to a fixed coordinate system [15]. *Fixed objects* are those whose kinematic state (KS) is constant through time (e.g. goals, lines, corners), whereas *mobile objects* have a variable KS through time. *Mobile objects* may be classified into *passive* or *active* depending on the source that determines their KS variations. *Passive objects* change their KS only due to actions executed by other objects (e.g. ball). On the other hand, *active-objects* have KS changes determined by their own actions. Finally, *active objects* may be classified into *partner* or *non-partner*. Partner objects share information (e.g. teammates) while non-partner robots do not (e.g. opponents). In the robot soccer application, objects move on a two dimensional plane, and the kinematic state (KS) of an object is defined as a vector that contains its pose, $\boldsymbol{k} = (x, y, \theta)^T$. (x, y) is the position of the object relative to $\boldsymbol{O}$ and θ is the orientation of the object relative to $\boldsymbol{O}$.

The control software in each robot must solve a complex problem and run in real time. Given the scarce perceptual information, the low availability of computational resources, and the requirement of a minimum frame rate (30fps),

the image processing algorithms cannot be as complex as the state of the art suggests. In order to achieve a high-level goal, several issues must be previously solved. Therefore, a software architecture consisting of four modules is implemented in each robot (Actuation, Decision Making, Perception and World Modelling). A detailed description of the here-relevant modules, Perception and World Modelling, is presented in the following paragraphs, while the connections between this two modules are illustrated in Fig. 1.

Perception

This module processes information coming from the sensors of the robot. The visual-perception sub-module classifies color pixels and then groups them into color blobs. Then, the objects are detected based on a set of rules applied to the blobs. For every visually detected object, a Gaussian PDF, $\boldsymbol{z}$, is generated for the pose of every object (containing the orientation only when necessary) relative to the observer robot. The mean of $\boldsymbol{z}$, $\overline{\boldsymbol{z}}$, is the calculated position using geometrical characteristics of the blobs and the 3D-pose of the camera. The covariance matrix Σ is calculated using previously obtained statistics. The perceived objects PDFs $\{\boldsymbol{Z}\}$ are partitioned into *fixed* $\{\boldsymbol{z}_f\}$, *passive* $\{\boldsymbol{z}_p\}$ and *active* $\{\boldsymbol{z}_a\}$ perceived objects.

The sonar-perception sub-module receives a set of up to nine measurements per sonar. Each value represents the Euclidean distance, d_i^r, between the observer robot and an i-th unknown object candidate. An unidimensional PDF, w, is generated for each measurement. The mean, $\overline{w}$, is the measured distance d_i^r, and the variance, δ, is previously obtained by an off-line statistic analysis. Additionally, the sonar ID w^{id} is included for each PDF ($w^{id} = 0$ for right sonar and $w^{id} = 1$ for left).

Finally, $\{\boldsymbol{z}_f\}, \{\boldsymbol{z}_p\}, \{\boldsymbol{z}_a\}, \{w\}$ are transmitted to the World-Modelling module, as shown in Fig. 1.

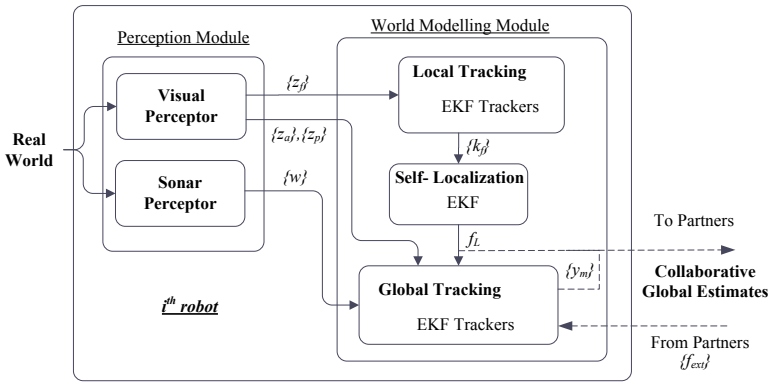


Fig. 1. Perception and World Modelling connection details

World Modelling

This module estimates the KS of the objects in the environment using information provided every time-step by the Perception module. Due to the noise and possible outliers generated by the Perception module, the World-Modelling module implements filtering and processing stages in order to achieve a precise representation of the KS of the surrounding objects.

The KS of an object is the mean of a Gaussian PDF, which is estimated for every object using a classical *Extended Kalman Filter* (EKF) as a tracker. The state variable $\boldsymbol{x}$ is the KS $\boldsymbol{k}$ of the object. The process model, f, its variance, R, the observational model, h, and its variance, Q, are defined for each tracker depending on the object type and observation source.

The World-Modelling module processes each received perception in a fashion that depends on its type. The $\{\boldsymbol{z}_f\}$ PDFs are delivered to the *local tracking* sub-module that implements a tracker for each object present in an a-priori-known map. The predictive stage for each tracker uses the odometry of the robot to move the estimated PDFs. The corrective stage uses the mean of $\boldsymbol{z}_f$ as an observation and Σ as the variance for a zero mean Gaussian PDF of the observational model.

The Self-Localization sub-module uses the KS estimations of the *fixed objects*, $\{\boldsymbol{k}_f\}$ (goals, lines and corners among others), to estimate recursively a Gaussian PDF ($\boldsymbol{f}_L$) of the current KS of the robot using an EKF-based self-localization algorithm. This localization estimation is relative to $\boldsymbol{O}$.

The global tracking sub-module uses $\{\boldsymbol{z}_a\}$, $\{\boldsymbol{z}_p\}$, $\{w\}$, $\boldsymbol{f}_L$ and the external estimated PDFs of the other robots mobile objects, $\{\boldsymbol{f}_{ext}\}$, as information sources.

Let us define the estimated PDFs of all mobile objects generated by the global tracking methodology as $\{\boldsymbol{y}_m\}$. In addition, an expression is defined for active, $\{\boldsymbol{y}_a\}$, and passive, $\{\boldsymbol{y}_p\}$, objects. These estimations are relative to $\boldsymbol{O}$.

Finally, the World-Modelling module transmits the estimated PDFs, relative to $\boldsymbol{O}$, of all mobile objects $\{\boldsymbol{y}_m\} = \{\boldsymbol{y}_a\} \cup \{\boldsymbol{y}_p\} \cup \{\boldsymbol{f}_L\}$ to the Decision-Making module and other robot's World-Modelling module.

The tracked objects KS is estimated using the methodology detailed in the following section.

3.2 Cooperative Global Tracking Methodology

The here-described methodology allows the tracking of multiple objects using multi-sensor ($\{\boldsymbol{z}_p\}$, $\{\boldsymbol{z}_a\}$, $\{w\}$) and cooperative $\{f_{ext}\}$ information.

A filtered PDF, y_m, referenced to O, is estimated for each mobile object using an EKF as a tracker. The maximum number of trackers, N, is previously determined according to a-priori knowledge of objects on the scene. In this particular framework, there are two types of mobile objects: N_p partner and N_o opponent robots, therefore $N = N_p + N_o$. Initially, all trackers are deactivated and linked

to a particular object with an unique ID. They are activated or deactivated on each time-step depending on a covariance value threshold. In every time-step, a predictive stage is executed for each active *global-tracker*, the predictive stage has no inputs since odometries are not directly received from other robots. However, $Q \neq 0$, therefore a fixed amount of uncertainty is added over y_m. Q is approximated as a function of the tipical average velocity of the *mobile objects*. If any perceptual PDF is matched to a tracker, a corrective stage is executed (The details of the matching procedure are presented on section 3.3). But $\{\boldsymbol{z}_p\}$, $\{\boldsymbol{z}_a\}$, $\{w\}$ are initially referenced to the observer robot, therefore, as the information needs to be coherent with partner estimated PDFs, a reference system transformation to $\boldsymbol{O}$ is needed (see block diagram in Fig 2). This transformation is applied to each perception using f_L. The resulting PDFs $\{\boldsymbol{z}'_p\}$, $\{\boldsymbol{z}'_a\}$, $\{w'\}$ are referenced to O.

Depending on the type of perceptual PDF, a different observational model function must be used on the corrective stage of the EKF. This is mainly because each type of sensor delivers data with different dimensionality. The output of the proposed methodology is the estimated KS, $\{\boldsymbol{k}_m\}$, of present mobile objects. For each object, $\boldsymbol{k}_m = \overline{\boldsymbol{y}_m}$, where $\overline{\boldsymbol{y}_m}$ is the mean of the estimated PDF $\boldsymbol{y}_m$. Although sonar measurements are generated with a higher frequency than visual perceptions, the absence in the angle information may lead to incorrect estimations. This will affect the trajectory and probably increase error because after a corrective stage of the EKF which only uses sonar information, the state mean will only be corrected in its radial component. This effect is presented in Fig. 3.

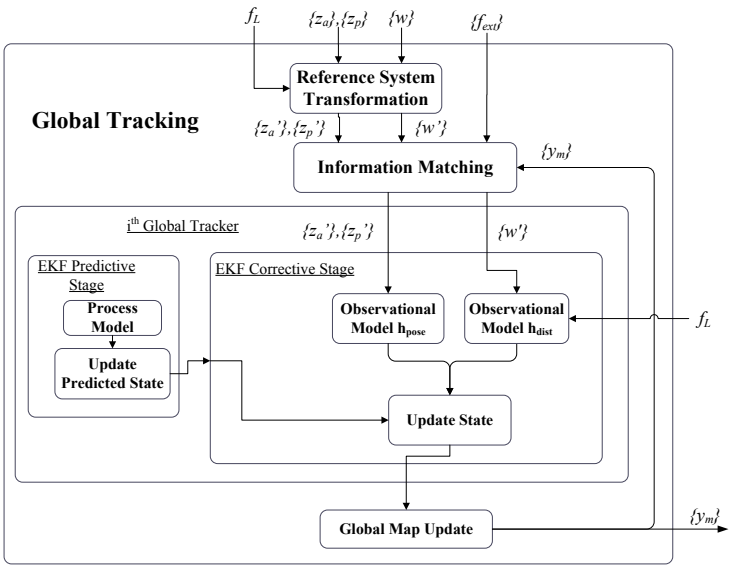


Fig. 2. Global Tracking Methodology Block-Diagram

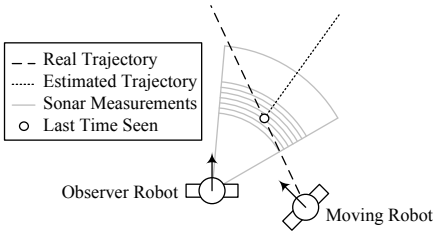


Fig. 3. Trajectory deformation effect produced by sonars

3.3 Matching Procedure

For every new perception, the tracker that is most likely to correspond to the received perception is selected in order to be updated. Perceptions are treated differently depending on their type. The output of this stage is the index of the *global-tracker* that best matches every input, or (-1) if the input is not associated to any tracker. All trackers are initially deactivated and they get activated when a visual perception or an external estimation cannot be associated to any active tracker.

Sonar Perception

When a *sonar-perception* PDF arrives, trackers whose state is outside the sonar range are not considered. For those active trackers whose state is inside the sonar range, an Euclidean distance measurement is used for deciding the index of the tracker associated to that particular input. If all trackers are deactivated or too far from the measurement (determined by the Euclidian distance threshold, γ), the sonar measurement is not associated to any tracker because there is no evidence that the measurement corresponds to an *active-mobile-object*. The matching algorithm for this type of perception is presented on Algorithm 1.

Algorithm 1. Matching Algorithm for Sonars

Let w be a sonar source PDF relative to O
Let γ be the Euclidian association threshold
Trackers outside sonar range are filtered
for active trackers in sonar range **do**
  Calculate tracker relative pose rP
  Calculate Euclidean distance D_{ot} observer-tracker
end for
if $\min\{D_{ot}\} > \gamma$ **then**
  return -1
else
  return Index of $\min D_{ot}$ tracker
end if

Visual Perception and External Estimations

This type of data are always associated to a tracker, except when all trackers are active and the Mahalanobis distance to each one of them is greater than

the Mahalanobis distance threshold ξ. This is usually the case when an false detection is occasionally perceived. The matching algorithm for this type of perception is presented on Algorithm 2.

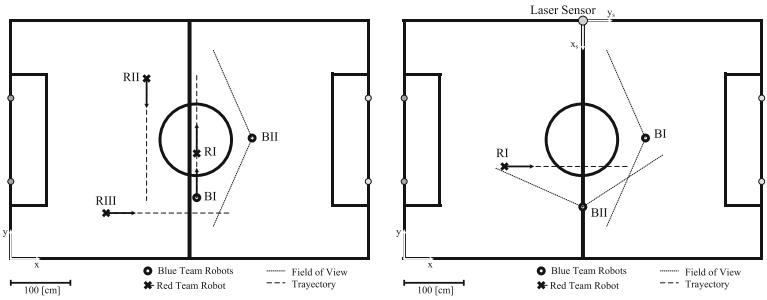
Algorithm 2. Matching Algorithm for Visual and External Estimates

```
Let z_m be a mobile-object visual source PDF relative to O
Let f_ext be an external source PDF relative to O
Let ξ be the Mahalanobis association threshold
Trackers are filtered depending on object class
for active trackers of interesting class do
   if tracker is active then
      Calculate Mahalanobis distance DM_ot observer-tracker
   end if
end for
if min{DM_ot} > ξ then
   if any tracker deactivated then
      return deactivated tracker index
   else
      return -1
   end if
else
   return Index of min DM_ot tracker
end if
```

4 Experiments

4.1 Experimental Setup

This section describes simulated and real experiments conducted on a SPL robot soccer environment. The first experiment uses a 2D simulator, which generates a time labelled database containing noisy robot perceptions, odometries and ground-truth data. The performance of the proposed system is evaluated on the same database but with different parameters. The experimental setup for the first experiment consists of five robots located on a simulated SPL soccer field. There are two teams of robots, the blue team and the red team. Three robots belong to the red team (RI, RII, and RIII) and two robots belong to the blue team (BI and BII). The initial position of the robots, their IDs and their ideal trajectories are illustrated in Fig. 4a. The idea of this experiment is that the blue robots BI and BII perceive the red team robots, and communicate their estimations between each other. The second experiment is executed using real robots on a SPL soccer field. The experimental setup is simpler than the simulated one due to operational complexity. In this experiment, the setup consists of two robots belonging to the blue team (BI and BII) and one robot belonging to the red team (RI). They are positioned on the field as illustrated in Fig. 4b. The ground-truth data is provided by a laser-based ground-truth system like the one proposed in [16], running on-line on an external computer. A time-labeled database is generated as in the simulation experiment.

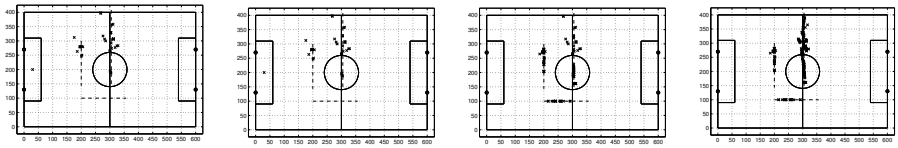


(a) Simulated Experiment (b) Real Experiment

Fig. 4. Experimental setup of robots in simulated and a real experiments

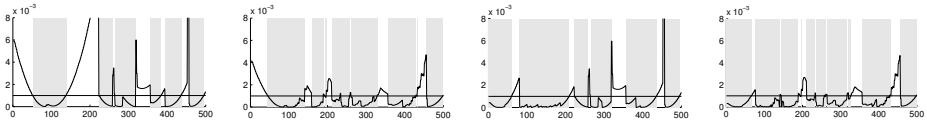
4.2 Results

The state-estimation error of the whole environment is calculated on both experiments every time-step for the moving robot BI, using four different configurations: (i) using only visual data, (ii) using visual and ultrasonic data, (iii) using cooperative estimations and visual data, and (iv) using cooperative estimations, visual and ultrasonic data. The estimated poses and expected trajectories of each tracker in all situations are presented in Fig. 5 for the simulated experiment and in Fig. 7 for the real experiment. The position error for tracker i at time k, denoted as e_k^i, is obtained by evaluating the difference between the estimated-PDF mean, $\boldsymbol{\mu}_k$, and the ground-truth position, $\boldsymbol{gt}_k$. The squared error ${e_k^i}^2$ is then calculated and normalized by the arena size $A_{size} = (600 * 400)[cm^2]$. The normalized squared error $\overline{{e_k^i}^2}$ for tracker III is plotted in Fig. 6 for the simulated experiment and in Fig. 8 for the real experiment. The normalized squared error is evaluated for the four different configurations (i to iv). The level of error at time k determines if the information is considered as useful or not. An arbitrary threshold ε, so-called decision-threshold, has been defined for this particular application with the value $(15[cm])^2, \overline{\varepsilon} = 0.001$ in the normalized space. The value of this threshold is the half of the max diameter of the tracked objects. A percentage of time when the information is considered as useful is calculated for each tracker and method. A table is generated with this values and is presented in Table 1 for the simulated experiment and in Table 2 for the real robot experiment. Furthermore, a gray background in Figures 6 and 8 represent visually when information is useful for robot decisions.Tables 1 and 2 show that the use of both ultrasonic and collaborative information is in all the analized cases the most reliable choice. Using these additional sources of information , the percentage of time in which the error is below $\overline{\varepsilon}$ grows between 20% and 40% in the simulated experiment and around 25% in the real experiment, from the case with only visual information. Additionally, the use of colaborative information is always positive, almost to the extent of the best case. The utility of the sonar information is not clear. While in table 1 it always appears to help, in table 2 it makes he results even worse than the case with only visual information.



(a) Only visual data (b) Visual and sonar data (c) Cooperative and visual data (d) Cooperative, visual and sonar data

Fig. 5. Results of the simulated experiment on a SPL reference system. Segmented lines indicate the ground-truth trajectory for each red-team robot. Crosses determine an estimated position for each time-step.

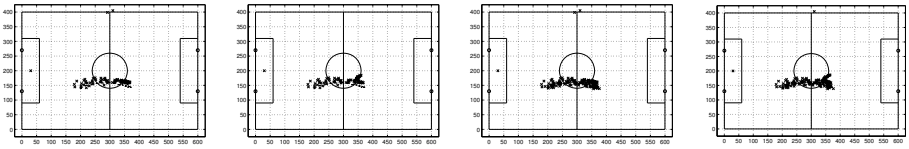


(a) Only visual data (b) Visual and sonar data (c) Cooperative and visual data (d) Cooperative, visual and sonar data

Fig. 6. Evaluation of the simulated experiment. Normalized error of the pose estimation of robot RI using different approaches (see subsection 4.2). Horizontal line represents decision-threshold $\overline{\varepsilon}$, and gray areas show when error is lower than $\overline{\varepsilon}$

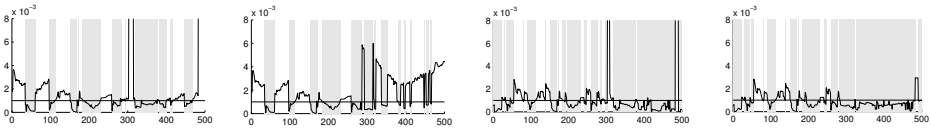
Table 1. Percentage of time when error is below $\overline{\varepsilon}$ for the simulated experiment. The different approaches (i), (ii), (iii) and (iv) are described in subsection 4.2

	(i)	(ii)	(iii)	(iv)
Opponent 1	18%	18%	62%	62%
Opponent 2	-	-	64%	64%
Opponent 3	48%	61%	67%	68%
$\overline{X}$	22%	26%	64%	65%



(a) Only visual data (b) Visual and sonar data (c) Cooperative and visual data (d) Cooperative, visual and sonar data

Fig. 7. Results of the real experiment on a SPL reference system. Segmented lines indicate the ground-truth trajectory for moving robot RI. Crosses determine an estimated position for each time-step.



(a) Only visual data (b) Visual and sonar data (c) Cooperative and visual data (d) Cooperative, visual and sonar data

Fig. 8. Evaluation of the real experiment. Normalized error of the pose estimation of robot RI using different approaches (see subsection 4.2). Horizontal line represents decision-threshold $\overline{\varepsilon}$, and gray areas show when error is lower than $\overline{\varepsilon}$.

Table 2. Percentage of time when error is below $\overline{\varepsilon}$ for the real robots experiment. The different approaches (i), (ii), (iii) and (iv) are described in subsection 4.2.

	(i)	(ii)	(iii)	(iv)
Opponent 1	45%	37%	63%	69%

5 Conclusion

A method for cooperatively tracking multiple objects using multi-sensory information was described and tested in a multi-robot application. The results show that the error of the kinematic state estimation of *mobile objects* decreases importantly when using information provided by cooperative robots, and decreases even more when using multi-sensorial information. In addition, the percentage of time when the estimation error is lower than an acceptable decision-threshold increases significantly (between 20% and 40% in the simulated experiment and around 25% in the real experiment) when using cooperative and sensor fusion techniques. As a future work, we expect to include velocity estimations to reduce sonar trajectory distortions and manage multiple tracker hypothesis for each object allowing multi-modal distributions.

Acknowledgment. This research was partially supported by FONDECYT (Chile) under Project Number 1090250.

References

1. Wheeler, M., Schrick, B., Whitacre, W., Campbell, M., Rysdyk, R., Wise, R.: Cooperative tracking of moving targets by a team of autonomous uavs. In: 2006 IEEE/AIAA 25th Digital Avionics Systems Conference, pp. 1–9 (October 2006)
2. Campbell, M., Whitacre, W.: Cooperative tracking using vision measurements on seascan uavs. IEEE Transactions on Control Systems Technology 15(4), 613–626 (2007)
3. Cánovas, J.P., LeBlanc, K., Saffiotti, A.: Robust multi-robot object localization using fuzzy logic. In: Nardi, D., Riedmiller, M., Sammut, C., Santos-Victor, J. (eds.) RoboCup 2004. LNCS (LNAI), vol. 3276, pp. 247–261. Springer, Heidelberg (2005)

4. Dietl, M., Gutmann, J.-S., Nebel, B.: CS freiburg: Global view by cooperative sensing. In: Birk, A., Coradeschi, S., Tadokoro, S. (eds.) RoboCup 2001. LNCS (LNAI), vol. 2377, pp. 133–143. Springer, Heidelberg (2002)
5. Karol, A., Williams, M.-A.: Distributed sensor fusion for object tracking. In: Bredenfeld, A., Jacoff, A., Noda, I., Takahashi, Y. (eds.) RoboCup 2005. LNCS (LNAI), vol. 4020, pp. 504–511. Springer, Heidelberg (2006)
6. Liu, Z., Zhao, M., Shi, Z., Xu, W.: Multi-robot cooperative localization through collaborative visual object tracking. In: Visser, U., Ribeiro, F., Ohashi, T., Dellaert, F. (eds.) RoboCup 2007. LNCS (LNAI), vol. 5001, pp. 41–52. Springer, Heidelberg (2008)
7. Nisticò, W., Hebbel, M., Kerkhof, T., Zarges, C.: Cooperative visual tracking in a team of autonomous mobile robots. In: Lakemeyer, G., Sklar, E., Sorrenti, D.G., Takahashi, T. (eds.) RoboCup 2006. LNCS (LNAI), vol. 4434, pp. 146–157. Springer, Heidelberg (2007)
8. Silva, J., Lau, N., Rodrigues, J., Azevedo, J., Neves, A.: Sensor and information fusion applied to a robotic soccer team. In: Baltes, J., Lagoudakis, M.G., Naruse, T., Ghidary, S.S. (eds.) RoboCup 2009. LNCS, vol. 5949, pp. 366–377. Springer, Heidelberg (2010)
9. Pagello, E., D'Angelo, A., Menegatti, E.: Cooperation issues and distributed sensing for multirobot systems. Proceedings of the IEEE 94(7), 1370–1383 (2006)
10. Moratuwage, M., Wijesoma, W., Kalyan, B., Patrikalakis, N., Moghadam, P.: Collaborative multi-vehicle localization and mapping in high clutter environments. In: 2010 11th International Conference on Control Automation Robotics Vision (ICARCV), pp. 1422–1427 (2010)
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. Dietl, M., Gutmann, J.-S., Nebel, B.: Cooperative sensing in dynamic environments. In: Proceedings of the 2001 IEEE/RSJ International Conference on Intelligent Robots and Systems, vol. 3, pp. 1706–1713 (2001)
13. Shuqin, L., Le, Z., Xiaohua, Y.: Design and implementation of multi-robot cooperative tracking. In: 2010 14th International Conference on Computer Supported Cooperative Work in Design (CSCWD), pp. 490–494 (April 2010)
14. Gouaillier, D., Hugel, V., Blazevic, P., Kilner, C., Monceaux, J., Lafourcade, P., Marnier, B., Serre, J., Maisonnier, B.: Mechatronic design of nao humanoid. In: IEEE International Conference on Robotics and Automation, ICRA 2009, pp. 769–774 (2009)
15. Coltin, B., Liemhetcharat, S., Meriç Andli, C., Tay, J., Veloso, M.: Multi-humanoid world modeling in standard platform robot soccer. In: 2010 10th IEEE-RAS International Conference on Humanoid Robots (Humanoids), pp. 424–429 (2010)
16. Marchant, R., Guerrero, P., del Solar, J.R.: A portable ground-truth system based on a laser sensor. In: Röfer, T., Mayer, N.M., Savage, J., Saranlı, U. (eds.) RoboCup 2011. LNCS, vol. 7416, pp. 234–245. Springer, Heidelberg (2012)