Distributed perception for a group of legged robots

Carlos E. Agüero, Antonio L. González, Francisco Martín, Vicente Matellán
Robotics Group. Departamento de Ingeniería Telemática y Tecnología Electrónica
ESCET - Universidad Rey Juan Carlos
{caguero,antol,fmartin,vmo}@urjc.es

Abstract **– Perception problem in robotics has been usually faced from the individual perspective, in this work we present preliminary results of a shared perception mechanism for a group of legged robots working in a dynamic scenario. The experiments has been conducted using a group of aiBo robots in the RoboCup environment. Each individual robot has its own perception of a common interesting feature, the ball, and they altogether build a shared perception.**

Keywords **– Multi-robot, distributed perception**

I. Introduction

RoboCup [1] is an international joint project to promote AI, robotics, and related field. It is an attempt to foster AI and intelligent robotics research by providing a standard problem where a wide range of technologies can be integrated and examined. RoboCup chose to use soccer game as a central topic of research, with the aim of innovate to be applied for socially significant problems and industries.

In order for a robot team performs a soccer game, several technologies must be incorporated including: Perception of the relevant features of the environment (the ball, the goals, the teammates, etc.), multi-agent collaboration, real-time reasoning, robotics, and sensor-fusion. RoboCup is a task for a team of multiple fast-moving robots under a dynamic environment.

The competition is organized in various leagues according to the type of robots used, their size, etc. The results showed in this paper have been used in the 4-legged category [2], where only aiBo robots are allowed.

RoboCup 4-legged league is organized around two main competitions: soccer matches and technical challenges. Soccer teams are composed by four robots and they must play football without the help of any external aid (human or computer). Only the communication among the members of the team is allowed.

This research has been partially sponsored by grants No. S-0505/DPI/0176 by Community of Madrid and No. DPI2004-07993-C03-01 by Spanish Ministry of Education corresponding to RoboCity and Acrace projects respectively.

Figure 1. aiBo robot with its main sensors detailed

II. From local to distributed perception

A repetitive task during a RoboCup game is to identify and locate some elements, as for example the ball, the other teammates, etc. In our team [1], each robot creates and maintains its own "map", where perceived elements have to be placed.

Usually, robots do not manage crips positions, but fuzzy estimations. So, we need to model positions as probability distributions. This approach will be also used to mix local and remote information, and the mixed information will be another probability distribution.

Robots can also receive information from other team's robots. In this case, we will need to mix locally perceived information, and remote information. Each robot perceives the elements using its own sensors. So, estimated positions are always relative to the actual position of the robot. We need to translate this local information to global coordinates to mix it. First we need to translate the local polar perception (r, ϕ) to a global Cartesian map, as shown in figure 2.

Local to global frame reference transformation is made in two steps. First, we transform from polar to Cartesian reference

[1] We are proud members of the TeamChaos, the team made up by three Spanish universities: University of Murcia, University Rey Juan Carlos and University of Alicante.

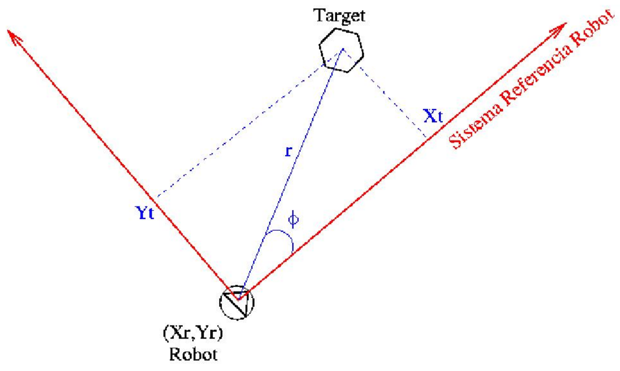

Figure 2. LOCAL PARAMETERS

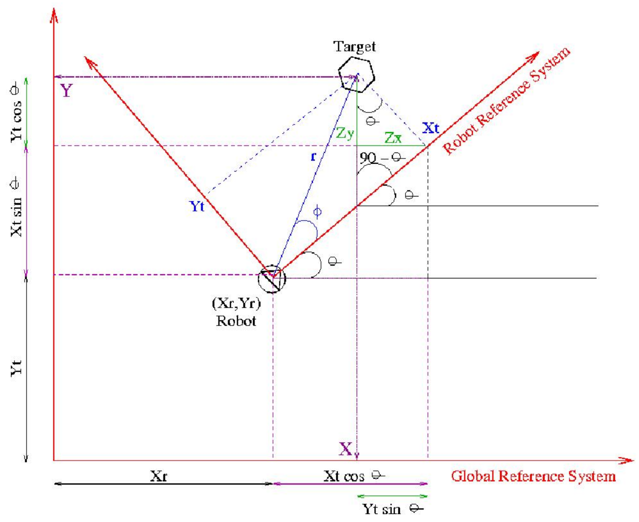

Figure 3. GLOBAL REFERENCE SYSTEM, AND PARAMETERS DEFINITION

system. So, we get X_T and Y_T. As shown in figure 2, we use the equations 1.

$$\begin{cases} X_T = f(r,\phi) = rcos(\phi) \\ Y_T = g(r,\phi) = rsin(\phi) \end{cases} \tag{1}$$

Second, we use the equations 2 to transform from local to global Cartesian reference system, as shown in figure 3.

$$\begin{cases} X = f(X_T,Y_T,X_r,Y_r,\theta_r) = X_r + X_T cos(\theta_r) - Y_T sin(\theta r) \\ Y = g(X_T,Y_T,X_r,Y_r,\theta_r) = Y_r + X_T sin(\theta_r) + Y_T cos(\theta r) \end{cases} \tag{2}$$

In order to incorporate the uncertainty, we will represent the position of an element as a normal distribution represented by the covariance matrix shown in equation 3 , where (α, β) represent the error in the estimation of the distance and angle. Our tests with the observation model revealed that the error in the distance estimation is linear $(\alpha * r)$ and the error in the angle can be considered constant (β).

$$C_m = \begin{pmatrix} \theta_r^2 & 0 \\ 0 & \theta_\phi^2 \end{pmatrix} = \begin{pmatrix} (\alpha r)^2 & 0 \\ 0 & \beta^2 \end{pmatrix} \tag{3}$$

If we express the target pose in this way, we need to apply the equation 4, in order to carry out the previous transformations. J corresponds to the Jacobian matrix of the covariance matrix to transform (C_m). To calculate the Jacobian matrix we need to calculate the partial derivatives of f and g with respect to all variables.

$$M_{new\_covar.} = JMJ^T \tag{4}$$

Therefore, if we start having a local object estimation based in local and polar coordinates, we will need to fulfill two transformations. The first one will turn the covariance matrix to Cartesian components (even in the robot system reference). The second transformation will convert it in the covariance matrix respect to the global reference system. The Jacobian matrix of the first transformation is showed in equation 5.

$$J_m = \begin{pmatrix} \frac{\partial f}{\partial r} & \frac{\partial f}{\partial \phi} \\ \frac{\partial g}{\partial r} & \frac{\partial g}{\partial \phi} \end{pmatrix} = \begin{pmatrix} cos(\phi) & -rsin(\phi) \\ sin(\phi) & rcos(\phi) \end{pmatrix} \tag{5}$$

The next Jacobian matrix is related to the last transformation where are involved the whole of five variables: Three of them concerning the pose of the robot (X_r, Y_r, θ_r) and the last pair pertaining to the position of the target (X_T, Y_T).

$$J = \begin{pmatrix} \frac{\partial f}{\partial X_T} & \frac{\partial f}{\partial Y_T} & \frac{\partial f}{\partial X_r} & \frac{\partial f}{\partial Y_r} & \frac{\partial f}{\partial \theta_r} \\ \frac{\partial g}{\partial X_T} & \frac{\partial g}{\partial Y_T} & \frac{\partial g}{\partial X_r} & \frac{\partial g}{\partial Y_r} & \frac{\partial g}{\partial \theta_r} \end{pmatrix} = \tag{6}$$

$$\begin{pmatrix} cos(\theta_r) & -sin(\theta_r) & 1 & 0 & -X_r sin(\theta_r) - Y_r cos(\theta_r) \\ sin(\theta_r) & cos(\theta_r) & 0 & 1 & X_r cos(\theta_r) - Y_r sin(\theta_r) \end{pmatrix} \tag{7}$$

Finally, to obtain the final covariance matrix, it is necessary to include the uncertainty in the robot pose (C_r) and the uncertainty of the target observation in local and Cartesian coordinates (C_T) and finally to apply the multiplication matrix with its Jacobian and the transpose (fig. 9).

$$C = J \begin{pmatrix} C_T & [0] \\ [0] & C_r \end{pmatrix} J^T = \tag{8}$$

$$J \begin{pmatrix} C_{T1,1} & C_{T1,2} & 0 & 0 & 0 \\ C_{T2,1} & C_{T2,2} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & C_{r1,1} & C_{r1,2} \\ 0 & 0 & 0 & C_{r2,1} & C_{r2,2} \end{pmatrix} J^T \tag{9}$$

III. EXPERIMENTATION ENVIRONMENT

Robot localization is a very challenging problem. Nevertheless, the purpose of our experiments is to validate a solution for the shared multi-robot perception problem. We have installed a groundtruth system [3] in our laboratory. In this way, we can forget the problems caused by the self-localization of the robots in the shared perception experiments isolating our task from others related.

In figure 4 we can see the calibration interface of the groundtruth program[3] used to adjust the camera parameters and color tables.

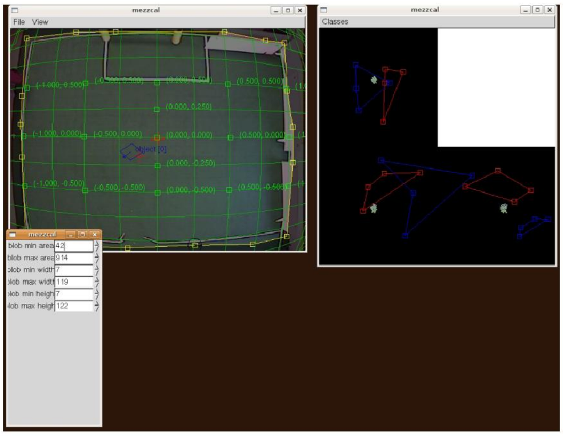

Figure 4. MEZZCAL INTERFACE

Figure 5 shows a capture of the application tracking three robots in real time. Each aiBo robot is equipped with an unique colored label for allowing object identification. We have developed a small program that reads the output generated by the groundtruth (a set of poses related to each object detected) and sends individual localization information to each robot.

The local localization system in each robot is overwritten by the information coming from the groundtruth. By this way, we obtain a very good quality localization in a transparent way.

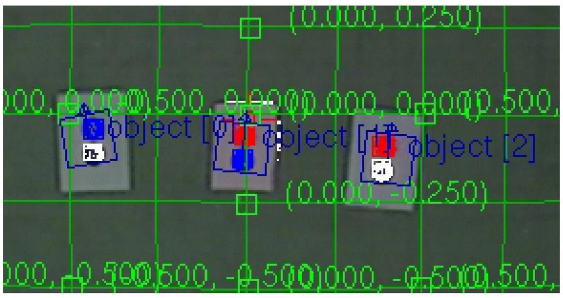

Figure 5. MEZZANINE TRACKING THREE OBJECTS

We have used the basic infrastructure of Team Chaos code [11] for the experiments. Team Chaos environment includes several modules for solving locomotion, visual perception, communication, amongst others.

The higher level behaviors module has been slightly modified for sending/receiving local information periodically. Each robot will use this information to create a global shared state that will be used in case the behaviors need it.

IV. BALL SHARING

A typical behavior at RoboCup environment is to look for the ball. This task is basically composed by a set of head movements travelling the field combined by some kind of heuristic for the robot's movement. In a typical match there are eight robots playing on a $24m^2$ field. It means that we have a large field with lots of moving objects hiding the ball.

We think that we can improve the time that a robot needs to localize the ball using shared information by its teammates. In our team, each robot should be spread on the field running different roles: *Goalkeeper*, *defender*, *kicker* and *supporter*. *Goalkeeper* remains in its area, *defender* stays in a defensive zone near its own net, *kicker* will be near the ball and *supporter* will be located on the attacking zone. It means that we have a team covering a large surface of the field. If they do not share any information, normally one or two of them will be tracking the ball but the others will be wasting their time looking for it.

A real team in any environment uses its members in a cooperative way. We think that we can exploit our communication features to share the ball position and maintain a global state of the ball pose among all the team members. Also, a robot could know the pose of the ball even if a set of robots will be hidden the ball.

Every robot in our team maintains a data structure called LPS (Local Perception System). LPS stores the current local estimation to every interesting object of the field (landmarks, nets and ball). The local estimation is a pair of numbers with the distance in millimeters, and the angle in degrees to the object measured from the robot system reference.

In order to use the shared approach in our team, we have included a new data structure for shared information. Now, every robot can consult its local information or the shared information. Note that the shared data does not replace the local information. In some cases, local estimations will be better and more precise. The shared information is a complement to local one.

Team cooperation is a key point in elaborated tasks. Last year, our team has been working in a system called Switch![4], that allocates robot roles in a dynamic way according to the play conditions, as we mentioned before. Every robot must send to the other teammates its information related to its position in the field and some local estimations to the objects. According to the information received and each own data, every robot choose one of the available roles and executes its associated behavior.

For this work, we have used the previous infrastructure developed for Switch! and include in the messages exchanged the pose and uncertainty of the ball. Now, when a robot receives information from a teammate, it can merge its own ball estimation with the estimation received from the other robot.

The merged estimation is the result of combining two covariance matrix (uncertainty of each ball estimation) and two position estimations.

$$C = C_1 - C_1[C_1 + C_2]^{-1}C_1 \tag{10}$$

$$\hat{X} = \hat{X}_1 + C_1[C_1 + C_2]^{-1}(\hat{X}_2 - \hat{X}_1) \tag{11}$$

Both estimations are merged into a new one by multiplying the distributions. We have used a two-dimensional statistical approach based on Bayes' Rule and Kalman filters explained in [5].

V. EXPERIMENTAL RESULTS

In order to test and measure the quality of the shared estimation to the ball we have made some tests. One side of the typical RoboCup field has been selected for making the experiments. First of all, we have marked on the green carpet of the field the position of 20 different points.

Next, we have put a pair of robots in two fixed positions, R1 $(0,0,-\frac{\pi}{2})$ and R2 (-1800,-1350,0) as you can see in figure 6.

Figure 6. SCENE OF THE TEST WITH BOTH ROBOTS TRACKING THE BALL

We began putting the ball on the point 1 during 10-15 seconds approximately. Our robots were running a log system activated from our suite of tools called ChaosManager. Also, the robots were receiving the ground truth information and they were updating their localization modules.

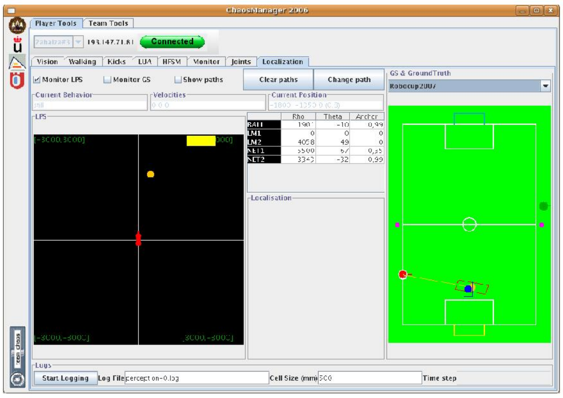

Figure 7. CHAOSMANAGER SHOWING LOCAL AND SHARED ESTIMATIONS

We can see a graphical representation of both local and shared estimations on the ChaosManager as shown figure 7. The orange ball represents the local estimation of robot 2, and the blue ball corresponds to the shared one. Note that the situation showed on the screen does not correspond with the scene showed in figure 6.

We triggered the log caption of the robot 1 from the ChaosManager and after 10-15 seconds we stopped the log. Then, we changed the position of the ball to the position 2 and repeated the process. At the end of the test we stored 20 files with the log of each test. In tables I and II we can see the data associated to each point. The error is represented as the Euclidean distance from the ideal position of the ball to the estimated position.

Point	Distance R1	Error R1	Std. Desv. Error
1	2102.97	412.40	477.84
2	1916.37	101.84	82.01
3	1850.0	91.43	51.03
4	1916.37	151.75	114.18
5	2102.97	386.69	473.60
6	1680.02	93.54	50.88
7	1439.61	51.92	19.66
8	1350.0	26.62	18.49
9	1439.61	43.85	21.02
10	1680.02	95.73	68.34
11	1312.44	62.90	10.71
12	986.15	110.40	15.60
13	850.0	29.54	6.74
14	986.15	28.25	10.63
15	1312.44	31.16	20.14
16	1077.03	109.70	8.35
17	640.31	51.37	3.69
18	400.0	24.44	1.74
19	640.31	10.03	2.49
20	1077.03	69.29	7.38

TABLE I.
LOCAL ESTIMATION AT DIFFERENT POSITIONS

Point	Distance R2	Error shared	Std. Desv. Error shared
1	2844.29	582.15	433.70
2	2353.72	252.81	237.76
3	1868.15	106.16	39.62
4	1392.83	63.02	38.08
5	943.39	149.12	192.95
6	2800.0	203.57	276.56
7	2300.0	82.30	52.70
8	1800.0	53.61	14.93
9	1300.0	36.87	18.20
10	800.0	36.78	28.40
11	2844.29	130.08	200.93
12	2353.72	144.49	69.12
13	1868.15	49.59	22.73
14	1392.83	46.89	6.33
15	943.39	18.59	11.44
16	2956.77	184.15	139.81
17	2488.47	108.20	93.11
18	2035.31	36.40	29.14
19	1610.12	29.63	14.50
20	1241.97	55.74	7.92

TABLE II.
SHARED ESTIMATION AT DIFFERENT POSITIONS

If we fuse the raw data of the tables into a single graph we can see how is the precision of the shared estimation compared with the local one. In figure 8 we can see both estimations. The location of the robot 2, which is located at larger distance towards the ball than robot 1, makes that shared estimation is better than local in six points of the twenty measured. If the teammate of the robot 1 were located closer to the ball than robot 1, the shared estimation would be better in all cases.

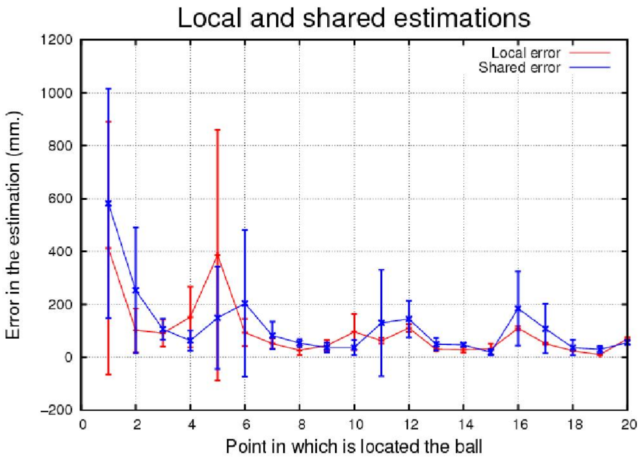

Figure 8. LOCAL AND SHARED ESTIMATION USING TWO ROBOTS

Despite the location of the robot 2, the shared estimation improve the accuracy of the estimation of robot 1 in all cases in which the distance between the ball and robot 2 is closer that the distance to the ball and robot 1. This information is available to every robot when it receives data from other teammate, so it is easy to know if the shared estimation could improve the local one.

VI. CONCLUSIONS AND FURTHER WORK

In this paper we have shown the preliminary results of a method for sharing information about relevant elements of the RoboCup environment. This technique could also be used in many other environments, where perceptive information is received from different sources.

This method is based in Bayes' rule and Kalman filter theory and can fuse two-dimensional Gaussian distributions into a single one, contributing more accuracy in some cases.

In a typical soccer RoboCup match, it is very common that a group of robots hide the ball to another. With this approach, every robot can maintain a parallel estimation that not only improve the precision, even in some cases will be the only way to locate the ball due to occlusions in the play. In that case the overall gain is infinite, due to with its local estimation the robot can not see the ball.

In our approach, we have isolated the uncertainty in localization, but this method allows to easily integrate the error in location of each robot.

Future lines will be addressed to combine more than two robots and to study different techniques to weight every data received, according to different parameters.

ACKNOWLEDGMENTS

Authors would like to thank the TeamChaos and Robotics Lab members for their support.

REFERENCES

[1] H. Kitano, M. Asada, Y. Kuniyoshi, I. Noda, and E. Osawa, "Robocup: The robot world cup initiative," in *ICJAI-95 - Workshop on Entertainment and AI/ALIFE*, 1995.

[2] The Robocup Federation, "Robocup Four-Legged League," http://www.tzi.de/4legged/bin/view/Website/WebHome.

[3] U. R. R. Lab, "Mezzanine," 2007, http://playerstage.sourceforge.net/mezzanine/.

[4] C. E. Agüero, V. Matellán, V. Gómez, and J. Cañas, "Switch! Dynamic roles exchange among cooperative robots," in *Proceedings of the 2nd International Workshop on Multi-Agent Robotic Systems - MARS 2006*. INSTICC Press, 2006, pp. 99–105.

[5] A. Stroupe, M. C. Martin, and T. Balch, "Distributed sensor fusion for object position estimation by multi-robot systems," in *IEEE International Conference on Robotics and Automation, May, 2001*. IEEE, May 2001.

[6] Sony, "Sony Global - AIBO Global Link," http://www.sony.net/Products/aibo/.

[7] J. Canovas, K. LeBlanc, and A. Saffiotti, "Robust multi-robot object localization using fuzzy logic," in *Proc. of the Int. RoboCup Symposium, Lisbon, PT*, 2004.

[8] Y. U. Cao, A. S. Fukunaga, and A. B. Kahng, "Cooperative mobile robotics: Antecedents and directions," *Autonomous Robots*, vol. 4, no. 1, pp. 7–23, March 1997.

[9] K. Lerman, C. V. Jones, A. Galstyan, and M. J. Mataric, "Analysis of dynamic task allocation in multi-robot systems," *International Journal of Robotics Research*, vol. 25, no. 4, pp. 225–242, 2006.

[10] Sony, "Sony AIBO SDE and Open-R SDK," http://openr.aibo.com, 2006.

[11] TeamChaos, "Team report 2005," 2006, online at www.teamchaos.es.

[12] C. E. Agüero, F. Martín, H. Martínez, and V. Matellán, "Communications and basic coordination of robots in TeamChaos," in *Actas VII Workshop de Agentes Físicos*, 2006, pp. 3–9.

[13] B. Gerkey and M. Mataric, "On role allocation in RoboCup," in *RoboCup 2003: Robot Soccer World Cup VII, 2004*. Springer-Verlag, 2004, pp. 43–53.