Actas do Encontro Científico
3º Festival Nacional de Robótica - ROBOTICA2003
Lisboa, 9 de Maio de 2003.

IMPROVING OBJECT LOCALIZATION THROUGH SENSOR FUSION APPLIED TO SOCCER ROBOTS

Pedro Marcelino, Pedro Nunes, Pedro Lima, M. Isabel Ribeiro

Instituto de Sistemas e Robótica
Instituto Superior Técnico
Av. Rovisco Pais 1, 1049-001 Lisboa.
{pmpin,pmgn}@rnl.ist.utl.pt, {pal,mir}@isr.ist.utl.pt

Abstract: This paper introduces a method for representing, communicating and fusing distributed, noisy and uncertain observations of objects by multiple robots. The approach relies on re-parameterization of two-dimensional Gaussian distributions that are used to represent the positions of all players and the ball. The approach enables two or more observers to achieve greater effective sensor coverage of the environment and improved accuracy in object position estimation. We demonstrate empirically that, using this approach, more observers achieve more accurate object position estimates. Two different procedures for merging Gaussian distributions were implemented and tested in the *RoboCup Soccer Server*, a simulated environment for robotic soccer. We also present the first results obtained with middle-size league robots.

Keywords: Sensor Fusion, Object Localization, Simulation, RoboCup

1. INTRODUCTION

In recent years we have seen an increasing interest in the development of multi-sensory robot systems. The reason for this interest stems from the conclusion that there are fundamental limitations on the reconstruction of environment descriptions using only a single source of sensor information. If robot systems are ever to achieve a degree of intelligence and autonomy, they must be capable of using many different sources of sensory information in an active and dynamic manner. Typically, individual robots can only observe part of their environment at any moment in time. In dynamic environments, information previously collected about currently unobservable parts of the environment grows stale and becomes inaccurate. Sharing information among robots increases the effective instantaneous visibility of the environment, allowing for more accurate modeling and more appropriate response. If processed effectively, information collected from multiple points of view can provide reduced uncertainty, improved accuracy and increased tolerance to single point failures in estimating the location of observed objects. By combining information from many different sources, it would be possible to reduce the uncertainty and ambiguity inherent in making decisions based only in a single information source. Our goal is to implement and compare two different sensor fusion methods that are presented in Sections 4 and 5, using a simulation scenario based on the *RoboCup Soccer Server*, which is explained in Section 3. In Section 6 we discuss

⋆ Work supported by the FCT "Programa Operacional Sociedade de Informação (POSI)" in the frame of QCAIII.

our experimental results obtained with simulated a real robots. Finally, in Section 7 conclusions are drawn.

2. BACKGROUND AND RELATED WORK

Most robot soccer team approaches use vision and/or sonar to localize the robots and vision to locate objects in the environment. Some teams share information for planning and dynamic role assignment. Others fill-in blank areas in the world model with shared data. Other distributed sensing approaches include merging independent grid cell occupancy probabilities measured by multiple robots (possibly distributed in time), and curve fitting of models and observations by multiple robots. The tasks addressed in (Durrant-Whyte 1988) and (Stroupe, Martin and Balch 2000a) is distinct from the others described above. These approaches focus on fusing multiple simultaneous observations of the same object from distributed vantage points (as opposed to observations from the same vantage point over multiple instants in time). Our goal is to provide more accurate instantaneous estimations of the location of dynamic objects that are simultaneously visible by multiple robots without relying on historical data.

3. ROBOCUP SOCCER SERVER

The *RoboCup Soccer Server* (Itsuki 2001) is a soccer simulation system which enables teams of autonomous agents to play a match of soccer against each other. The system was originally developed in 1993 by Itsuki Noda (Itsuki 2001). A simulation soccer match is carried out in *client-server* style. The *soccer server* provides a domain (2-D virtual soccer field), simulates all the movements of objects in this domain and controls a soccer game according to several rules.

A RoboCup agent has three different types of sensors: a *visual* sensor, an *aural* sensor and a *body* sensor. In the remainder of this section we discuss the characteristics of the *visual* and *aural* sensors, which allow us to implement sensor fusion in this domain.

3.1 Visual Sensor

The visual sensor detects visual information about the field such as distance and direction to objects in the players current field of view (see Figure 1). This information is automatically sent to the player every 150 ms. All visual information given is *relative* from the player's perspective. As a result, a player cannot directly see his own global position or the global position of other players and

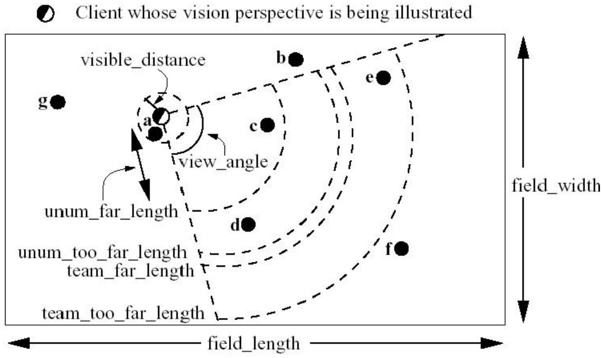


Fig. 1. Agent's field of view, reprinted from (Itsuki 2001)

of the ball. The agents thus need a way to derive global information from a visual message. For this purpose, several landmarks (flags, lines and goals), with known global positions, are placed on and around the field. By combining the known global positions of these landmarks and their relative positions (included in the visual message) an agent can determine his own global position and the global positions of the ball and other players.

One of the real-world complexities contained in the *soccer server* is that the precision of visual information decreases as the distance to an object increases. Noise is introduced into the visual sensor data by quantizing the values sent by the server. Distances to objects are quantized as:

$$Q\_Dist = Q(exp(Q(ln(Dist), StepVal)), 0.1) \qquad (1)$$

where $Dist$ and Q_Dist are the *exact* and *quantized* distance values respectively and $StepVal$ is a parameter denoting the quantize step. For players and the ball $StepVal$ is equal to 0.1 and for landmarks the value 0.01 is used. Furthermore,

$$Q(v, q) = rint(v/q) \cdot q \qquad (2)$$

where 'rint' denotes a function which rounds a value to the nearest integer. The amount of noise thus increases as the distance to the object increases. For example, when an object (ball or player) is roughly reported at distance 100.0m the maximum noise is about 10.0m, whereas when the reported distance is 10.0m the noise can be about 1.0m.

Because of the visual noise introduced by the *soccer server*, each agent has a different view of the environment.

In order to perform sensor fusion we approximate this noise model with a two-dimensional Gaussian distribution $N(\mu, C_L)$. μ is an vector representing the calculated position of the object and C_L is an

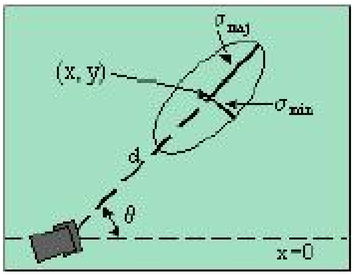


Fig. 2. Distribution parameter definitions: mean (x,y), angle of major axis (θ), major and minor standard deviations (σ_{maj}, σ_{min}) and distance to mean, reprinted from (Stroupe et al. 2000a).

diagonal matrix that represents the variance along both axis (see Figure 2 and (3)).

$$C_L = \begin{bmatrix} \sigma_{maj} & 0 \\ 0 & \sigma_{min} \end{bmatrix} \tag{3}$$

σ_{maj} is the variance along the axis that points from the robot towards the observed object, this value is calculated based on the quantization made by the *soccer server*. σ_{min} represents the variance along the perpendicular axis and is based on the maximum error in angle that an observation can have, namely $\frac{\pi}{180}$. We used the following approximations during our tests:

$$\sigma_{maj} \approx 2 \cdot \frac{Dist}{10} \tag{4}$$

$$\sigma_{min} \approx 2 \cdot \tan(\frac{\pi}{180}) \cdot \frac{Dist}{10} \tag{5}$$

3.2 Aural Sensor

The aural sensor detects messages which are received when another player issues a say command. The *soccer server* communication paradigm models a crowded, low-bandwidth environment in which the agents from both teams use a single, unreliable communication channel (Itsuki 2001). Spoken messages are *immediately* broadcast to all nearby players from both teams without perceptual delay. The player can hear at most one message every second simulation cycle. When multiple messages arrive during this time, the first one is chosen according to their order of arrival and the rest are discarded. Besides this, players also have to deal with the fact that their communication range is limited. A spoken message is transmitted only to players within 50 meters from the speaker. The maximum length of the message string is limited to 512 bytes. This enables the players to communicate their view of the environment to other teammates. In our case, we communicate the calculated positions and the associated variance matrix of all observed objects.

4. FUSING GAUSSIAN DISTRIBUTIONS

In order to use sensor fusion, we must exchange sensor information, between team members. This information exchange provides a basis through which individual sensors can cooperate with each other, resolve conflicts or disagreements, or complement each other's view of the environment. Our goal is to compare the efficiency of two known sensor fusion methods: the Stroupe *et al* (Stroupe et al. 2000a) and the Durrant-Whyte (Durrant-Whyte 1988) method. The first approach simply merges the Gaussian distributions of the observations made by the robot with the Gaussian distributions of the observations made by other robots. The second approach takes into account the last known position of the object and tests if the readings obtained rom several sensors are close enough, in order to make the fusion. When this test fails, no fusion is made and the sensor reading which has less variance is chosen. The conditions in which this test fails and succeeds are presented in Section 5. The remainder of this section provides the necessary mathematical background to understand how the merging of Gaussian Distributions is made in both methods.

4.1 Stroupe Method

We represent a single observation of an object as a two-dimensional Gaussian distribution (Figure 2), because the environment is two-dimensional and we don't want to determine the orientation of the robot. The center, or mean, of the distribution is the calculated position of the object and the standard deviations along the major and minor axes of the distribution correspond to estimates of the uncertainty (or noise) in the observation along each axis. The distribution corresponds to the conditional probability that the object is in that location, given the observation.

The diagonal covariance matrix, C_L, of an observation relative to coordinates aligned with the major and minor distribution axes is initially determined from the major and minor axis standard deviations (σ_{maj} and σ_{min}) in the local coordinate frame (designated L).

Since observations may be oriented arbitrarily with respect to the global coordinate frame (angle θ relative to global x-axis), they must be transformed to this frame. Rotation of X in (6) by θ leads to the following relationship.

$$C^{-1} = R(-\theta)^T C_L^{-1} R(-\theta) \Rightarrow \tag{6}$$
$$C = R(-\theta)^T C_L R(-\theta)$$

where R is an two-dimensional rotation matrix.

Once the observation is transformed to the global coordinate frame, we combine two individual covariance matrices of the same object into a covariance matrix C' representing the combined distribution. The derivations are provided with more detail in the technical report (Stroupe, Martin and Balch 2000b).

$$C' = C_1 - C_1[C_1 + C_2]^{-1}C_1 \tag{7}$$

where C_1 represents the individual covariance matrix of an object obtained with the robot that is merging the distributions and C_2 is the individual covariance matrix of the same object obtained with a second robot.

The mean of the resulting merged distribution, $\hat{X}'$, is computed from the individual distribution means and covariance matrices.

$$\hat{X}' = \hat{X}_1 + C_1[C_1 + C_2]^{-1}(\hat{X}_2 - \hat{X}_1) \tag{8}$$

where $\hat{X}_1$ and $\hat{X}_2$ are the calculated positions of the same object made by the two robots.

The principal axis angle of the merged distribution is obtained from the merged covariance matrix C' entries:

$$C' = \begin{bmatrix} A & B \\ B & D \end{bmatrix} \tag{9}$$

$$\theta' = \frac{1}{2}\arctan\left(\frac{2B}{A-D}\right) \tag{10}$$

Lastly, the resulting major and minor axis standard deviations are extracted by rotating the covariance matrix to align with those axes and reversing (6).

$$C'_L = R(\theta')^T C' R(\theta') \tag{11}$$

4.2 Durrant-Whyte Method

Durrant-Whyte considers a sequence of observations $\overrightarrow{z} = \{z_1, ..., z_n\}$, of a state of the environment $p \in P$, which are assumed to derive from a sensor modeled by a contaminated Gaussian Density, so that the i^{th} observation is given by:

$$f_i(z_i|p) = [(1-\epsilon)N(p, \Lambda_i^1) + \epsilon N(p, \Lambda_i^2)] \tag{12}$$

with $0.05 < \epsilon < 0.1$ and $\Lambda_i^2 \gg \Lambda_i^1$.

It is well known that if the prior distribution $\pi(p)$ and the conditional observation distribution $f(z|p)$ are modeled as independent Gaussian random vectors $p \sim N(\hat{p}, \Lambda_0)$ and $z_1 \sim N(\hat{p}, \Lambda_1)$ respectively, then the posterior distribution $\pi(p|z)$ after taking a single observation z_1 can be derived using Bayes Law and is also jointly Gaussian with mean vector

$$\hat{p'} = [\Lambda_0^{-1} + \Lambda_1^{-1}]^{-1}[\Lambda_1^{-1}z_1 + \Lambda_0^{-1}\hat{p}] \tag{13}$$

and covariance matrix

$$\Lambda' = [\Lambda_0^{-1} + \Lambda_1^{-1}]^{-1} \tag{14}$$

This method can be extended for n independent observations, as explained in (Durrant-Whyte 1988, page 111)

5. THE MULTI-BAYESIAN TEAM

In the Multi-Bayesian system, each team member individual utility function is given by the posterior likelihood, for each observation z_i:

$$u_i(p^w = \delta_i(z_i), p) = \pi(p|z_i) \approx N(\hat{p}, \delta_i), \tag{15}$$
$$i = 1, 2$$

A sensor or team member will be considered rational if for each observation z_i of some prior feature $\delta_i(z_i) \in P_i$ it makes the estimate which maximizes its individual utility $u_i(p_i, \delta_i(z_i)) \in \mathbb{R}$. In this sense, utility is just a metric for constructing a complete lattice of decisions, allowing any two decisions to be compared in a common framework. The team utility function is given by the joint posterior likelihood:

$$\begin{aligned} U(p^w(z_1, z_2), p) &= F(p|z_1, z_2) \\ &= f_1(p|z_1)f_2(p|z_2) \end{aligned} \tag{16}$$

The advantage of considering the team problem in this framework is that both individual and team utilities are normalized so that comparisons can be performed easily, supplying a simple and transparent interpretation to the group rationality problem. The team itself will be considered group-rational if together the team members choose to estimate $p^w \in P$ (feature of the environment), which maximizes the joint posterior density.

$$\begin{aligned} p^w &= \arg\max F(p|z_1, z_2) \\ &= \arg\max f(p|z_1)f(p|z_2) \end{aligned} \tag{17}$$

There are 2 possible results for (17)

(1) $F(p|z_1, z_2)$ has a unique mode equal to the estimate p^w;
(2) $F(p|z_1, z_2)$ is bimodal and no unique group-rational consensus estimate exists.

If $F(p|z_1, z_2)$ has a unique mode, as displayed in Figure 3, it will satisfy:

$$\max F(p|z_1, z_2) \geq \max f_i(p|z_i), \tag{18}$$

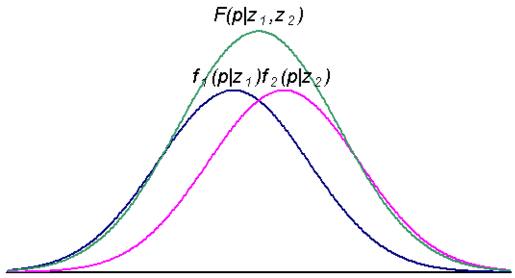


Fig. 3. Two Bayesian observers with joint posterior likelihood indicating agreement.

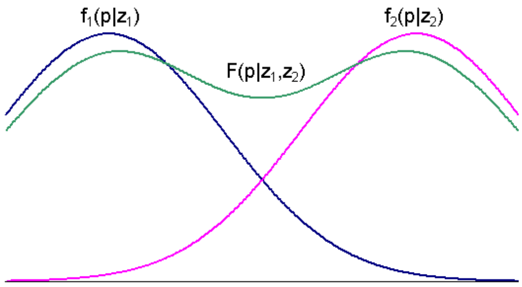


Fig. 4. Two Bayesian observers with joint posterior likelihood indicating disagreement.

$$i = 1, 2$$

Conversely, if $F(p|z_1, z_2)$ is bimodal, as displayed in Figure 4, then:

$$\max f_i(p|z_i) > \max F(p|z_1, z_2), \qquad (19)$$
$$i = 1, 2$$

A rational team member will maximize utility by choosing to either agree or disagree with the team consensus. If a team member satisfies (19), then it will not cooperate with the team estimate. Thus the decision made by a team member based of its observations z_i is:

$$p^w = \delta(z_i) = \qquad (20)$$
$$\arg\max\{f_i(p|z_i), F(p|z_1, z_2)\}, i = 1, 2$$

Whether or not the individual team members will arrive at a consensus, team estimate will depend on some measure of how much they disagree $|z_1 - z_2|$. If z_1 and z_2 are close enough then the posterior density $F(p|z_1, z_2)$ will be unimodal and satisfy (18), with the consensus estimate given by (17) (see Figure 3). As $|z_1 - z_2|$ increases, $F(p|z_1, z_2)$ becomes flatter and eventually bimodal (see Figure 4). At this point, the joint density will satisfy (19), no consensus team decision will be reached.

To find the point at which this space is no longer convex and disagreement occurs, all we need to ensure is that the second derivative of the function $F(p|z_1, z_2)$ is positive. Differentiating leads to:

$$\frac{\partial^2 F}{\partial p^2} = \frac{1}{f_1}\frac{d^2 f_1}{dp^2} + \frac{1}{f_1}\frac{d^2 f_2}{dp^2} + \frac{2}{f_1 f_2}\frac{df_1}{dp}\frac{df_2}{dp} \qquad (21)$$
$$= (\sigma_1^{-2} + \sigma_2^{-2}) - [\sigma_1^{-2}(p - z_1) + \sigma_2^{-2}(p - z_2)]^2$$

For this to be positive and hence $F(p|z_1, z_2)$ to be convex, we are required to find a consensus over the feature of the environment p which satisfies.

$$[\sigma_1^{-2}(p - z_1) + \sigma_2^{-2}(p - z_2)]^2(\sigma_1^{-2} + \sigma_2^{-2})^{-1} \qquad (22)$$
$$\leq 1$$

Notice that (22), is a normalized weighted sum, a scalar equivalent to the Kalman gain matrix. Then the consensus $\bar{p}$ which maximizes F is given by

$$\bar{p} = \frac{(\sigma_1^{-2} z_1 + \sigma_2^{-2} z_2)}{(\sigma_1^{-2} + \sigma_2^{-2})} \qquad (23)$$

Replacing (23) into (22), we obtain

$$\frac{(z_1 - z_2)(z_1 - z_2)}{(\sigma_1^{-2} + \sigma_2^{-2})} = D_{12}(z_1, z_2) \qquad (24)$$

where $D_{12} \leq 1$. The disagreement measure $D_{12}(z_1, z_2)$ is called the Mahalanobis distance.

6. FUSIONS COMPARISONS

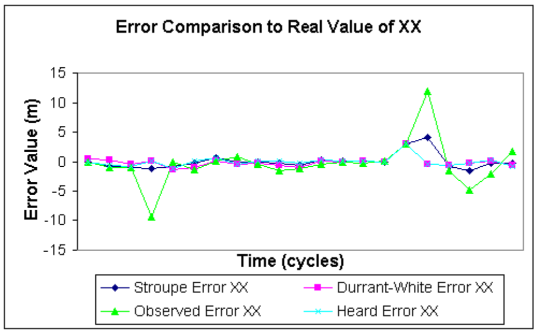


Fig. 5. Comparison of the errors between the fusions, observed data and heard data with the real value of the X-coordinate

6.1 Experimental Setup

An experiment was conducted on a Simulated Soccer Team based on (Stone, Veloso and Riley 1999) and (Reis and Lau 2001), which was modified to run both the Stroupe and Durrant-Whyte fusion algorithms. The data fusion was made based on the data observed and heard by all team members during a game. Since both players and ball were moving, their positions were dynamically evolving during the game time. In order to deal with this, we had to model both the heard

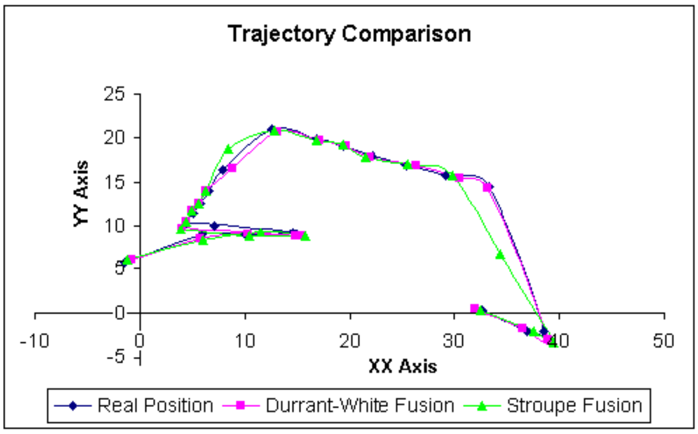
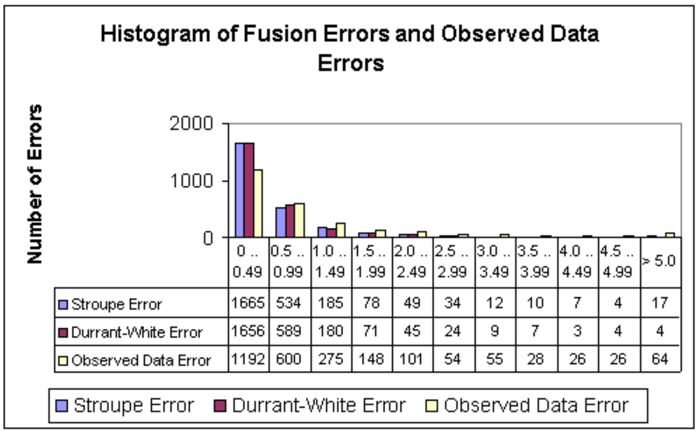


Fig. 7. Comparison of the real trajectory with the trajectories obtained with both fusion methods



	0 .. 0.49	0.5 .. 0.99	1.0 .. 1.49	1.5 .. 1.99	2.0 .. 2.49	2.5 .. 2.99	3.0 .. 3.49	3.5 .. 3.99	4.0 .. 4.49	4.5 .. 4.99	> 5.0
Stroupe Error	1665	534	185	78	49	34	12	10	7	4	17
Durrant-White Error	1656	589	180	71	45	24	9	7	3	4	4
Observed Data Error	1192	600	275	148	101	54	55	28	26	26	64

Fig. 8. Histogram of fusion Errors

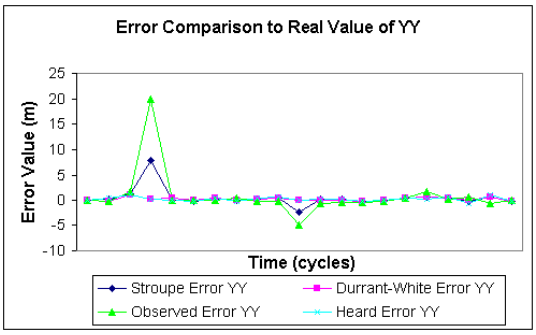


Fig. 6. Comparison of the errors between the fusions, observed data and heard data with the real value of the Y-coordinate

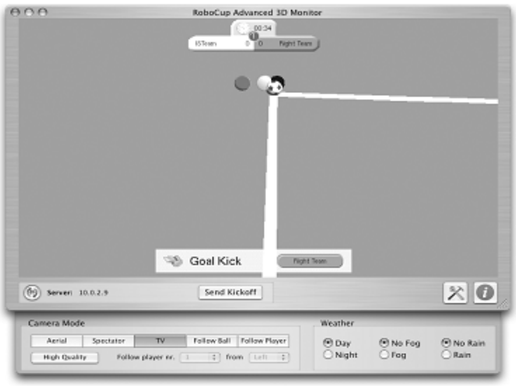


Fig. 9. 3d graphics client showing real ball position, heard data, observed data and Stroupe's fusion result

and observed data to reflect the uncertainty of the objects position based on its distance to the player who was collecting the data. The goal is to give a better estimate of all players positions and the ball position. The results were obtained and visualized on a 3D graphics client(see 9), which allowed the visualization of both fusions and of the raw data from the observed and heard data.

In the Durrant-Whyte fusion algorithm, the decision process to determine the ball position is made by first determining if the heard and observed ball position can be merged through the Mahalanobis distance. If this is possible, the ball position will be the result of the fusion, otherwise we choose from the heard and observed data the reading of smallest variance. The following results were taken from the goalkeeper of our team, but this is made for all players. Each player acts as a sensor, taking observations and reporting them to the other team members.

6.2 Experimental Results in Simulation League

First we show the positive impact that communication between agents can cause in an environment like the RoboCup simulation league. We present a brief comparison of the number of known player positions with and without communication. When communication is turned off, the player is able to see in average three to four players on each simulation cycle. Remember that the time interval between visual messages is 150ms and simulation cycles last 100ms. With communication the agents are capable of hearing in average four player positions each cycle. On average one of these positions is related to a previously unknown player. As can be seen, communication can double the updating rate of the information leading to more accurate and up to date information.

Some experimental results are shown graphically on Figures 5 and 6, which compare the errors obtained with both fusion methods, observed data and heard data. As can be seen, the Durrant-Whyte algorithm has better results than the Stroupe algorithm, when the sensor data changes abruptly. In this case the Durrant-Whyte algorithm takes advantage of taking into account the previous position of the object. This feature will not allow the fusion result to diverge too much from the real value, acting as a filter, eliminating erroneous and spurious sensor data. When the two sensors diverge a lot the Durrant-Whyte algorithm chooses the best sensor i.e. the one with smallest variance.

In Figure 7 we represent the trajectory made by the ball. As we can see the Durrant-Whyte manages to follow the real value with better accuracy, converging for the real value faster, and with little oscillations.

In Figure 8, we display a histogram of the number of errors of both axis; we can notice that the Durrant-Whyte method minimizes the error to the real position, preventing the error to grow to values too far from the real values. In the extreme case we only have four occurrences of more than ten meters errors in the Durrant-Whyte method, while the Stroupe method has seventeen occurrences. Analyzing these four occurrences, we concluded that they occurred in a situation in which both the heard and observed data were obtained when both players were too far away from the object. According to the sensor model of both the sensors, at such far away distance, the variance is very large, causing the Gaussians to be very dilated, and because of this the Durrant-Whyte managed to fusion the information. However this problem could be resolved with an higher number of observations, eliminating these spurious occurrences.

6.3 Experimental Results of Middle Size League Robots

The Durrant-White method was also implemented on a robot of RoboCup's middle size league. The goal was to obtain a better estimate of the ball position based on observations acquired from 2 different cameras. The ball was positioned on several previously known positions along a straight line and some observations were made with both cameras.

Figure 10 shows observations made with both cameras, the real position of the ball and the estimated position, calculated using the Durrant-Whyte method. The result obtained with this approach improves the estimation of the ball position, converging to the real ball location. As can be seen, the error of the computed position is the smallest, it also has fewer oscillations, so it establishes a better estimate than both observations independently.

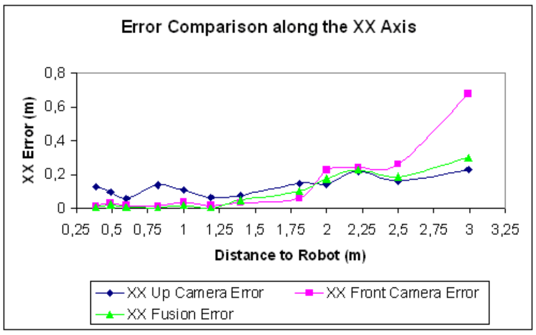


Fig. 11. Comparison of the absolute errors between the fusion and observed data with both cameras with the real value of the X-coordinate

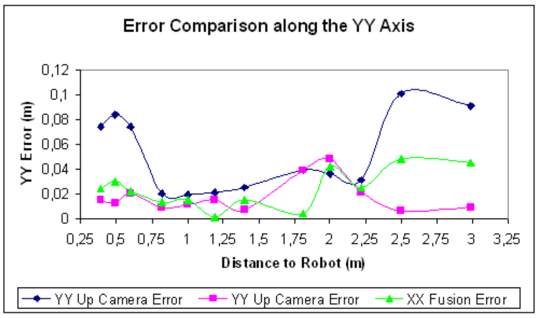


Fig. 12. Comparison of the absolute errors between the fusion and observed data with both cameras with the real value of the Y-coordinate

In figures (11 and 12) we can observe the absolute error according to the xx Axis and yy Axis. When the ball is between 0.4 and 1.5 meters away from the robot the fusion error is very small around 2 centimeters and in some cases the error is almost zero. Although in this experiment we never observed disagreement between both

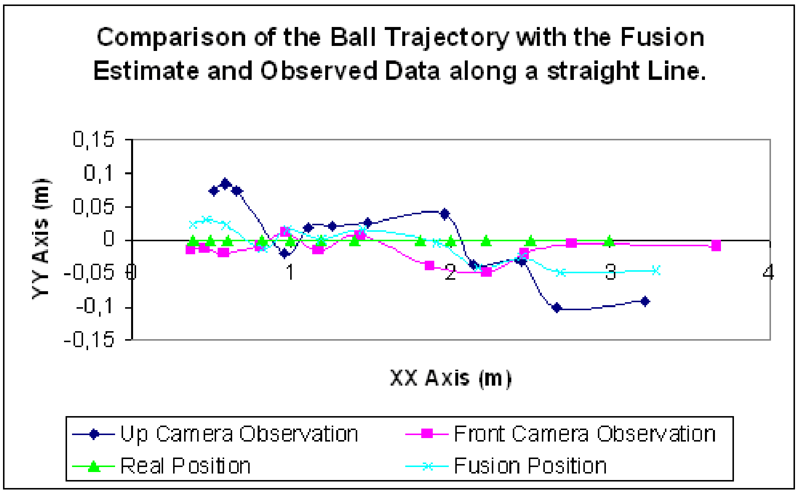


Fig. 10. Comparison of the Ball Trajectory with the Fusion Estimate and Observed Data along a straight Line

cameras , we can easily observe that the fusion estimate position converges to the observation of the front camera when the ball is between 0.4 and 1.5 meters away from the robot, when the ball is farther away it converges to the up camera observation. This happens because the sensor models of both cameras are different, since the front camera give a better estimation when the ball is closer and the up camera gives a better estimation when the ball is far.

7. CONCLUSIONS

We presented a comparison of two known methods to improve position estimates by fusing data from two or more robot agents. Both approaches are based on the Bayes Rule and implement a real-time sensor data fusion on a multi-robot system.

As shown in the paper the Durrant-Whyte approach is more effective, because it considers the previous ball position in the estimation of the new fusion of sensor data. Also, if sensor outputs vary to a such degree that the Mahalanobis distance increases and becomes greater than one, then we have two disagreement sensors information. In this case the Durrant-Whyte method chooses the sensor output with the smallest reading error. This will eliminate any erroneous and spurious data, that might appear, giving a much more accurate view of the world and of its state. These are the reasons why we chose to use this method in the real robots.

Stroupe's method fuses directly two Gaussian distributions, but it has the problem that if one of the sensor observations has errors associated with it, the result will diverge from the real value, due to a sensor anomaly, error reading data or bad sensor calibration.

In conclusion, the Durrant-Whyte fusion algorithm is better to determine the ball position than the Stroupe algorithm, and it should be implemented whenever possible do determine the ball position or the positions of other objects. The only limitation of the Simulated League is the fact that since each player acts as an sensor, they cannot communicate in real time all at the same time, in order share their information of the state of the environment. So rules must be created in order to give preference to those sensors whose information is more relevant for the environment features detection.

REFERENCES

Durrant-Whyte, H. F. (1988). *Integration, Coordination and Control of Multi-Sensor Robot Sistems*, Kluwer Academic Publishers.

Itsuki, N. (2001). Robocup soccer server: Users manual for soccer server version 7.07 and later.

Reis, L. and Lau, N. (2001). Fcportugal team description: Robocup 2000 simulation league champion, *RoboCup-2000: Robot Soccer World Cup IV* pp. 29–40.

Stone, P., Veloso, M. and Riley, P. (1999). The CMUnited-98 source code.

Stroupe, A., Martin, M. C. and Balch, T. (2000a). Merging gaussian distributions for object localization in multi-robot systems, *Proc. of the ISER '00 Seventh International Symposium on Experimental Robotics*, Springer-Verlag.

Stroupe, A., Martin, M. C. and Balch, T. (2000b). Merging probabilistic observations for mobile distributed sensing, *Technical Report CMU-RI-TR-00-30*, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA.