Robust Multi-Robot Object Localization Using Fuzzy Logic

Juan Pedro Cánovas[1], Kevin LeBlanc[2], and Alessandro Saffiotti[2]

[1] Dept. of Information and Comm. Eng.
University of Murcia, 30100 Murcia, Spain
juanpe@dif.um.es

[2] AASS Mobile Robotics Lab, Dpt. Technology
Örebro University, 70182 Örebro, Sweden
{klc,asaffio}@aass.oru.se

Abstract. Cooperative localization of objects is an important challenge in multi-robot systems. We propose a new approach to this problem where we see each robot as an expert which shares unreliable information about object locations. The information provided by different robots is then combined using fuzzy logic techniques, in order to reach a *consensus* between the robots. This contrasts with most current probabilistic techniques, which average information from different robots in order to obtain a *tradeoff*, and can thus incur well-known problems when information is unreliable. In addition, our approach does not assume that the robots have accurate self-localization. Instead, uncertainty in the pose of the sensing robot is propagated to object position estimates. We present experimental results obtained on a team of Sony AIBO robots, where we share information about the location of the ball in the RoboCup domain.

1 Introduction

Cooperating robots can benefit in a number of ways from the exchange of information about perceived objects. For example, a robot which does not directly see an object can still get an estimate of its position. Also, an individual robot's estimate of an object position can be improved through information sharing. This occurs when some robots have more accurate and/or more reliable position estimates for that object, which can happen due a number of things. For example, a robot may be better localized, have a better view of the object, or have more effective sensors. In many situations, it is even possible for a group of robots, all having relatively poor estimates of an object's position, to obtain more accurate and reliable estimates through information sharing. However this requires that the information sharing be performed in an effective way.

The problem of cooperative object localization is the problem of fusing information from different sources in a way which produces agreement about object positions; the agreed upon positions should also be as close as possible to the real object positions. However, information fusion can result in degradation of

To appear in:
D. Nardi, M. Riedmiller and C. Sammut, Editors.
Proc. of the 2004 Int. RoboCup Symposium. Springer-Verlag, 2004.

information if it is not done carefully. For instance, if we combine a correct observation from a robot A with an incorrect observation from a robot B by simple averaging (or even weighted averaging), the result will be worse than what A would have established alone. This averaging problem occurs in many existing approaches (e.g. [11], [14], [17], [18]), as is mentioned in the next section.

Another limitation of many current approaches (e.g. [6], [18]) is that they assume that a robot has very little uncertainty about its own position. In reality, however, this assumption is often not valid. Ignoring self-localization uncertainty can severely restrict the applicability of these methods. For example, in the mid-sized and four-legged leagues of RoboCup [8], robots often have poor self-localization, due to the fact that the domain is highly dynamic, and also because of frequent undetected collisions.

In this paper, we propose a new approach to the cooperative localization problem, based on fuzzy logic. One distinctive point of our approach is that we see each robot as an expert, which provides unreliable information about object locations. We use fuzzy logic to fuse information provided by different sources in order to reach a *consensus* about object positions. This contrasts with many other methods, which yield a *compromise* between various data sources.

Moreover our method carefully takes into account different facets of uncertainty, including unreliability in perceptual information and uncertainty in self-localization. An important contribution of our method is that the uncertainty in a robot's own position is consistently propagated to its object position estimates. One of the obvious advantages of doing this is that high self localization accuracy is not required in order to get position estimates which reflect our knowledge. Our method provides estimates which are consistent with our observations, while taking into account the uncertainty present in both perception and in self-localization.

In the rest of this paper, we first describe some alternate approaches from the literature; then we describe our technique and discuss how we have implemented it on a team of Sony AIBO robots [5]; finally, we present experimental results based on the RoboCup domain (four-legged league).

2 Related Work

There are many existing approaches to cooperative object localization. One simple approach is to use a switching strategy. For example, Roth *et al* [13] use such a strategy for locating the ball in the four-legged league of RoboCup. In their implementation each robot maintains a local world model and a shared world model. A robot which has not seen the ball in some time selects the most probable location estimate from those available in the shared world model. Unfortunately it is fairly easy to select the wrong estimate, and even if the right estimate is selected, the accuracy is no greater than that achieved by a single robot.

Several approaches fuse information from multiple sources using variations on Kalman filtering techniques (e.g. [11], [14], [17]). Alternatively, Stroupe *et al* [18] represent observations as two-dimensional Gaussian distributions, and observations from different robots are merged by taking the average of the Gaussians. In case of disagreement, the merged position estimate will typically be weighted more heavily toward what the majority of robots believe. However, both these methods fuse observations using weighted averages of some sort, and fail to provide a robust solution in the presence of false positives and outliers. Observations which are incorrect, due to errors in perception or in self-localization, can make the fused estimate worse than that of some of the individual robots. In general, averaging produces a compromise between estimates. The method proposed in this paper, by contrast, seeks a consensus between them.

These methods have been improved upon through the use of gating strategies, which discard observations which are deemed invalid. Observations can be deemed invalid if, for example, they are very different from current position estimates. One could also discard observations which do not correspond to what the majority of robots believe. Marcelino *et al* [12] compare the method used by Stroupe *et al* [18] with a fusion algorithm described in [7], which uses a gating strategy to discard observations thought to be inconsistent with previous sensor readings. They show that the gating strategy improved performance and robustness. However simple gating strategies are often not enough. When a target object is unobserved for a certain time, or when a target object moves very rapidly, correct observations could be consistently disregarded since they may no longer correspond with the current belief about the state of the world. Typically, the confidence in the current belief should eventually decrease below some threshold, at which point valid observations would once again be accepted.

A more robust way to deal with outliers and false positives is to implement a *voting scheme*, which encourages belief in observations which are consistent with the majority opinion. Markov localization (e.g. [9]), which is widely used for both individual and cooperative object (and self) localization, implements such a voting scheme by maintaining a discrete, multi-modal probability distribution for position estimates. In the individual robot case, it encourages belief in positions which are consistent with previous and/or current sensor observations. The cooperative case extends this by also encouraging belief in positions which are consistent with information received from other robots. Markov localization is quite robust, though in general its accuracy is less than that of other common methods (e.g. Kalman filtering).

In [6] and [10], a hybrid method called Markov-Kalman localization (ML-EKF) is described. This method uses a grid-based version of Markov localization as a robust, low-resolution plausibility filter. It then uses an extended Kalman filter to compute precise object positions based on the observations which were deemed valid by the Markov process. This approach inherits the robustness of Markov localization and the precision of Kalman filtering. It provides an informed way of deciding which observations should be discarded, and yields increased robustness with respect to more arbitrary gating strategies. However it is still possible for false positives or outliers to affect the result, depending on how strict one is in tuning the plausibility filter. Moreover, this method assumes that a robot has very little uncertainty about its own position.

The method we propose in this paper can also be seen as a sort of voting scheme. However, there are two main differences with respect to other voting approaches. First, votes contain the full uncertainty in the agent's self-localization; second, this uncertainty can be multi-modal. One of the advantages of this careful treatment of uncetainty is that high self-localization accuracy is not required by our method.

A number of other methods for object localization using a single robot have been described in the literature, such as particle filters and multiple hypothesis tracking. However the cooperative aspects of these methods have not been thoroughly investigated. In [16], Schmitt *et al* describe an approach to cooperative perception using multiple hypothesis tracking; but careful tuning of the pruning parameters is required for this method to be effective.

3 Representing Location Information

3.1 Fuzzy Location Information

Location information may be affected by different types of uncertainty. Consider a robot that needs to grasp a given object. This task requires that the position of the object be known with a high degree of precision, as in the statement (a) "The object is at position $x = 81$". The statement (b) "The object is near the center of the table" is *vague*, since it does not give a crisp position. The statement (c) "The object is somewhere on the table" is *imprecise*, since it does not give a point position. The statement (d) "The object is either at position $x = 81$ or at position $x = 162$" is *ambiguous*, since it gives multiple options. And the statement (e) "The object was seen yesterday at position $x = 81$" is *unreliable*, as the object may no longer be there. An ideal uncertainty representation formalism should be able to represent all of these statements. Perhaps more importantly, it should account for the differences between these statements, by representing information at the level of detail at which it is available.

Fuzzy logic techniques are attractive in this respect [15]. We can represent information about the location of an object by a fuzzy subset μ of the set X of all possible positions [20, 21]. For any $x \in X$, we read the value of $\mu(x)$ as the degree of possibility that the object is located at x given the available information. Fig. 1 shows some examples, taken in one dimension for graphical clarity. Cases (a–e), correspond to the five items of information mentioned previously. Case (e) is especially interesting: in order to account for unreliability, we include in the distribution a uniform "bias" to indicate the possibility that the object could also be located somewhere else. Total ignorance, in particular, can be represented by the fuzzy set $\mu(x) = 1$ for all $x \in X$; that is, all locations are possible. Finally, case (f) shows a combination of the previous types of uncertainty.

Fuzzy positional information can be represented in a discretized format in a position grid: a tessellation of the space in which each cell is associated with a

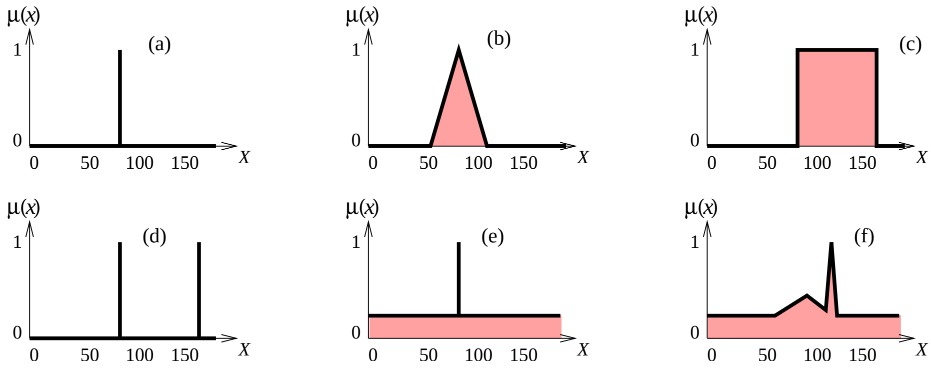

Fig. 1. Different types of fuzzy location information: (a) crisp, (b) vague, (c) imprecise, (d) ambiguous, (e) unreliable, (f) mixed.

number in the range $[0, 1]$, representing the degree of possibility that the object is in that cell. A common choice is to use a 2-dimensional grid of square cells with uniform size. A 3-D grid can be used if the orientation of the object is also relevant. In the approach proposed in this paper we use 2-D fuzzy position grids to represent our belief about the positions of objects, and a 3-D grid to represent our belief about the robot's own pose in the environment.

3.2 Fuzzy Information Fusion

An important component of a representation for uncertain information is how information coming from different sources can be *fused* together. Fusion can be used to combine the information provided by multiple robots, or by multiple sensors in the same robot.

If location information is represented by fuzzy sets, fusion can be performed by fuzzy intersection between these sets [1]. Let μ_1 and μ_2 be two fuzzy sets representing the information about the position of a given object, provided by sources 1 and 2, respectively. Then their combined information is given by the fuzzy set $\mu_{12} = \mu_1 \cap \mu_2$ defined by

$$\mu_{12}(x) = \mu_1(x) \otimes \mu_2(x), \tag{1}$$

where $\otimes$ is a t-norm.[1] Fig. 2 (a) illustrates fuzzy fusion. The result of the fusion of μ_1 and μ_2 is indicated by the shadowed area.

There are two facts about fuzzy fusion that should be noticed. First, only those locations which are regarded as possible by both sources are retained in the result of the fusion. Intuitively, the resulting fuzzy set μ_{12} represents the

[1] T-norms are the general operators used to perform intersection of fuzzy sets [19]. The most common examples of t-norms are minimum, product, and the Łukasiewitz operator $\max(0, a + b - 1)$. In this work, we use the product t-norm.

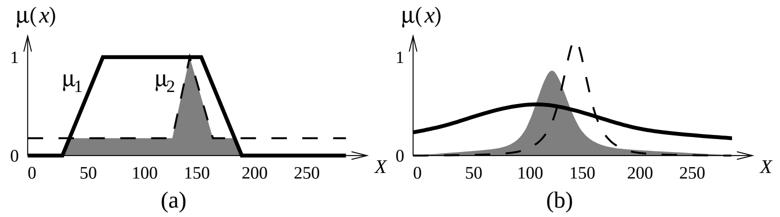

Fig. 2. Fuzzy fusion (a) computes the *consensus* between two sources of information. Probabilistic fusion (b) computes a *tradeoff* between them.

consensus between the two sources of information. This contrasts with standard probabilistic techniques, in which information fusion is typically performed by some sort of weighted average, representing a *tradeoff* between sources. Fig. 2 (b) shows how two pieces of information similar to the ones in Fig. 2 (a) might be fused in a probabilistic setting, by combining Gaussians. Notice that with fuzzy fusion the peak of the resulting distribution μ_{12} coincides with the peak of μ_2, since this is compatible with the peak of μ_1; it lies in between those peaks when using probabilistic fusion. As we mentioned earlier, averaging is often not the best solution when combining location information from multiple robots.

The second fact to note is that fuzzy fusion automatically discounts unreliable information. Consider the next example shown in Fig. 3. The information represented by μ_1 includes a high bias (0.8) to indicate that it is fairly unreliable, while the information represented by μ_2 only has a small bias (0.1). Correspondingly, the result of the fusion mostly reflects μ_2 and it is only marginally influenced by μ_1. In practice, this means that fuzzy fusion allows us to discard unreliable information, provided that this unreliability is correctly represented.

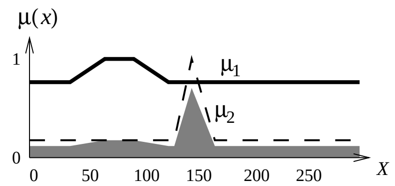

Fig. 3. Discounting unreliable information in fuzzy fusion. Information μ_1 is unreliable, as indicated by the high bias, and therefore only has a small impact on the result.

We sometimes need to extract a point estimate from the location information represented by a fuzzy set μ, e.g., to be used in other navigation modules. A common way to do this is by computing the center of gravity (CoG) of μ:

$$\hat{x} = \frac{\int_{x \in X} x\mu(x)\, dx}{\int_{x \in X} \mu(x)\, dx}. \tag{2}$$

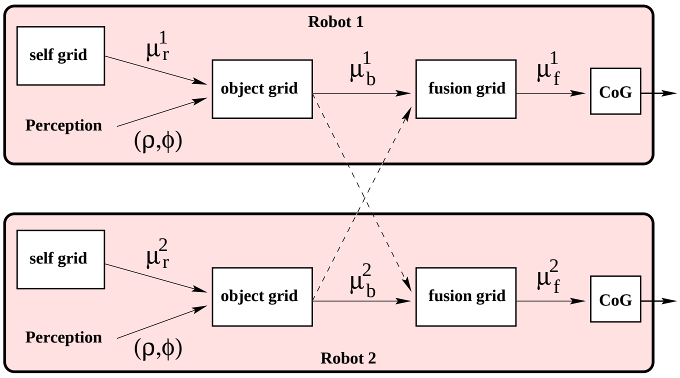

Fig. 4. Schema of our cooperative localization technique.

4 Sharing Location Information

Fuzzy location information can be shared in order to get a common view of the environment. The key point here is to see each robot as a source of unreliable information. This information is represented and fused using the techniques described in the previous section. The diagram in Fig. 4 summarizes our fusion schema in the case of two robots. The extension to n robots is straightforward.

We use three fuzzy position grids in each robot. The "self grid" contains the information μ_r about the robot's self location. The "object grid" contains the information μ_o about the global location of the target object, derived from perceptual observations. The "fusion grid" contains the result of the fusion of location information computed by different robots. This grid is kept separate from the object grid in order to avoid circular dependencies. A final defuzzification step obtains a point estimate, if needed, using formula (2). Temporal aspects aside, the robots should end up with identical object position estimates.

4.1 From the Self Grid to the Object Grid

In order to simplify the exchange of location information, we assume that all robots use a common global reference system F_g. Each robot r, however, acquires perceptual data from its own point of view, and it estimates the positions of objects in its local reference frame F_r. In order to represent this information in the global frame F_g, we need to apply a coordinate transformation function $T_r^g : F_r \to F_g$.

Assume that the robot has observed an object at polar coordinates (ρ, ϕ) with respect to its own reference frame F_r — see Fig. 5. If we knew the robot's pose (x_r, y_r, θ_r) in the global frame F_g, then the computation of the global coordinates (x_o, y_o) of the object would be straightforward. In our case, however, the robot's pose is not known with certainty, but is represented by a fuzzy set μ_r in F_g.

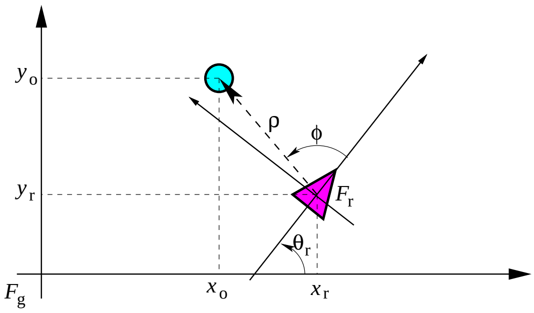

Fig. 5. Transforming local (observed) coordinates into global ones in the crisp case.

Accordingly, the object's global position is given by a fuzzy set $\mu_o(p)$ defined as follows: for any $p = (x, y)$, the degree of possibility $\mu_o(p)$ that the object is located at the global position p is given by

$$\mu_o(p) = \sup\{\mu_r(q, \theta) \mid d(\overline{pq}) = \rho \text{ and } \angle(\overline{pq}) = \phi + \theta\}, \tag{3}$$

where $q = (x', y')$ is any 2-D position, and $d(\overline{pq})$ and $\angle(\overline{pq})$ respectively denote the length and the orientation of the segment $\overline{pq}$. Intuitively, this says that the object can be at location p as long as there is some possible pose (q, θ) for the robot such that, if observed from that pose, the location p would appear at distance ρ and angle ϕ.

It should be emphasized that the above transformation preserves the full self-localization uncertainty contained in the fuzzy set μ_r. In particular, if the robot is highly uncertain about its own position, then many poses (q, θ) will have a high value in the fuzzy set μ_r. Correspondingly, there will also be many possible positions for the object, which will then have a high value in μ_o. In this respect, our approach is different from most existing approaches, in which the global position of the object is computed by assuming a point estimate for the location of the robot, and uncertainty is added after the transformation. These approaches do not correctly propagate the uncertainty in the robot's pose, since they do not take into account the non-linearities in the local-to-global coordinate transformation. Moreover, they cannot deal with a multi-modal distribution in the robot's self-localization. Our transformation (3) addresses both of these issues.

In a discretized position grid, the transformation (3) can be computed by the following algorithm.

```
foreach cell p
   μo(p) := 1 - reliability(observation)
   foreach cell q such that dist(p, q) = ρ                    (4)
      μo(p) := max{μo(p), μr(q, (∠(p, q) − φ))}
end
```

In general, the μ_o distribution computed through this algorithm contains a "bias"; that is, its minimum value β is strictly positive. This indicates some

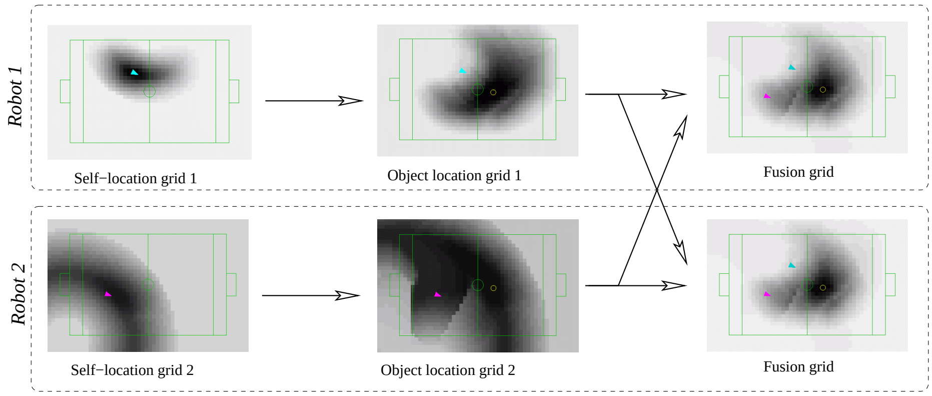

Fig. 6. Example of fusion of information between two robots.

unreliability in the position of the observed object, as discussed in section 3.1, and illustrated in Fig. 1 (c). This unreliability can originate from two sources: (i) unreliability in perceptual information, represented by 'reliability(observation)' in the algorithm; and (ii) unreliability in the self-localization of the robot, represented by the bias in μ_r which is propagated to μ_o. This bias is instrumental in discounting unreliable information in our fuzzy fusion scheme.

The behavior of our transformation is illustrated in Fig. 6. The picture shows the individual self and object grids for two robots during information sharing, as well as the result of the fusion. Darker cells indicate higher degrees of possibility. The triangles represent the robot's estimates of their own positions. The circle in the object and fusion grids indicates the real position of the object. In this example, the object is a ball in a soccer field in the RoboCup domain. Both robots can see the ball.

The self-localization grid of Robot 1 shows that this robot has a fairly good estimate of its own position. The corresponding object grid indicates a larger uncertainty in the position of the ball. The reason for this is that the uncertainty in the ball location is affected by the uncertainty both in the (x, y) position of the robot and in its orientation (not shown in the self-grid for graphical clarity). The effect of the non-linearity in the coordinate transformation (3) is clearly visible. Robot 2 has just observed a landmark after a long time during which no observation was made, hence the distribution μ_r is concentrated around a circle at a given distance from the landmark. The low quality of the current self-location estimate for Robot 2 is reflected in the high bias, which appears as a uniform gray background in the grid. The corresponding object grid provides a very rough estimate of the location of the ball.

4.2 From the Object Grids to the Fusion Grid

Fusion of information from different robots is performed by fuzzy intersection of their distributions according to equation (1). We use a non-idempotent t-norm operator $\otimes$, like the product operator, in order to reinforce object positions which are possible according to all robots. Recall that the bias acts as a reliability filter in fuzzy fusion: information with high bias has a small impact on the result of the fusion. Thus information coming from robots which have poor self-localization or poor perceptual information (and know it) is automatically discounted. In particular, if a robot has no current perceptual information about an object, all the cells in the corresponding object grid will have values close to 1; these values will not (significantly) affect the result of the fusion.

The last grids in Fig. 6 show the result of the fusion. Fuzzy intersection has produced a distribution that reflects the agreement between the two robots about the position of the object, and it is significantly better than the individual distributions. The result has been mostly influenced by the information provided by Robot 1 since this has a lower bias, reflecting higher reliability.

The above schema assumes that the full object location grids are exchanged between robots. This may be an expensive operation in terms of time and communication bandwidth. As reported below, we have used some devices in our implementation in order to reduce complexity. With these devices, we were able to run information sharing among four Sony AIBO robots in real time, using the limited on-board computational resources of this platform, and using the limited bandwidth allowed by the rules of RoboCup.

5 Experiments

We have tested our method in the RoboCup domain using a team of Sony AIBO legged robots. This is a challenging, highly dynamic domain, characterized by: (i) imprecise sensor information, since the main sensor available to each robot is a color camera with limited resolution and a limited field of view; and (ii) high localization uncertainty, since legged locomotion and poor perception make the self-localization problem difficult.

5.1 Implementation

We have implemented the schema described in the previous section using, in each robot, a 3-D fuzzy position grid to represent self-localization, and a 2-D position grid to represent the ball position. [2]

The computation of the global ball position from observations is done according to an optimized version of algorithm (4). We exploit the fact that for every cell p in the ball grid we only need to look at the cells q in the self grid at

[2] The self-localization grid is actually implemented using a $2\frac{1}{2}$-D grid to allow for efficient real-time computation, as detailed in [3].

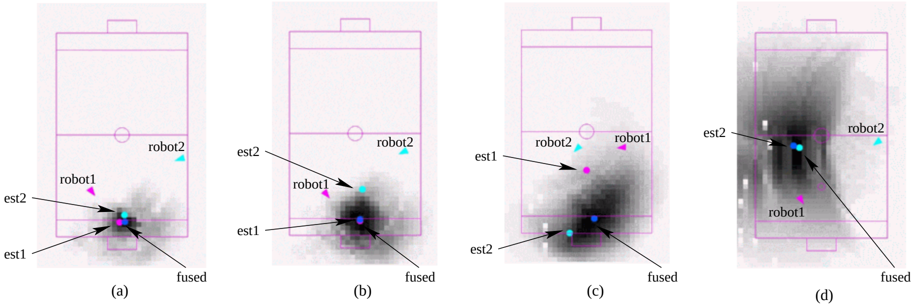

Fig. 7. Four examples of cooperative object localization in the RoboCup domain. The arrows show the individual position estimates produced by the two robots (est1, est2) and the fused estimate. The four cases are described in the text.

a distance ρ from p. This is equivalent to saying that we only need to evaluate the cells in the circle with center p and radius ρ. To do so, we use an adaptation of Bresenham's algorithm for plotting 2D circles in a digital grid [2].

The other expensive element of our schema is the transmission of the ball grids between robots. In order to preserve bandwidth, we convert each cell value to one byte and treat the grid as a gray-scale image. We then compress the grid using a simple run length encoding scheme. The ball grid is sent only if it contains information which is newer or of better quality than the one last sent.

5.2 Static Robots

A first set of experiments were performed to assess the quality of our fusion technique under static conditions. In these experiments, we used two robots standing at fixed positions and we put the ball at various known locations in order to evaluate the accuracy of object localization. The robots were alternatively looking at the landmarks in the field in order to self-localize, and at the ball in order to assess its position.

Two sample cases taken from these experiments are shown in Fig. 7 (a,b). The figure shows the fused object position grid resulting from the fusion of the two individual object grids (not shown). The three small circles represent the point estimates of the ball position according to each robot alone (marked 'est1' and 'est2') and as a result of the fusion (marked 'fused').

Case 1 (Fig. 7 a). Both robots can see the ball, and have accurate self-localization. Both the individual estimates and the fused estimate are very close to the real position of the ball (see quantitative data below).

Case 2 (Fig. 7 b). Both robots can see the ball, but while robot 1 is well localized, robot 2 is not. The result of the fusion in this case is almost identical to the information provided by robot 1, while the information provided by robot 2 is discounted since it is unreliable due to poor self-localization. This is similar to the case illustrated in Fig. 6.

5.3 Moving Robots

When robots move during a real game, self-localization may become very poor due to several factors: legged locomotion is poorly modeled; the tilting and rolling of the robot's body introduces inaccuracies in perception; and the robots are busy tracking the ball, and may note see the field landmarks very often. We performed a second set of experiments in which we used two constantly moving robots, and we placed the ball at various known positions. The following is a sample case taken from these experiments.

Case 3 (Fig. 7 c). Both robots have rather poor self-localization. Because of this, the estimates of the ball position are very inaccurate and quite different from each other. When intersecting the corresponding fuzzy position grids, however, we obtain a fairly accurate fused estimate of the ball position. Note that this position does not lie between the two individual estimates, and hence could not have been obtained by averaging (e.g. using a Kalman filter).

5.4 Blind Robot

In a last set of experiments we tested the ability of robots which do not see the ball to use information provided by other robots. The following case illustrates a typical situation taken from these experiments.

Case 4 (Fig. 7 d). The ball is behind robot 1, so only robot 2 can see it. In addition, the self-localization of both robots is rather poor. The fused information is similar to that provided by robot 2 alone. While this information suffers from poor self-localization, it is still made available to robot 1, who would otherwise have no knowledge of the ball position.

5.5 Results

In each experiment we have measured five quantities: the error in the self-location estimate for each robot, denoted by Δ^1_{self} and Δ^2_{self}; the error in the ball position estimate produced by each robot individually, denoted by Δ^1_{ball} and Δ^2_{ball}; and the error in the ball position estimate obtained by our fusion technique, denoted by $\Delta^{\text{fuzzy}}_{\text{ball}}$. Errors were measured by comparing the center of gravity of the fuzzy location sets with the ground truth. The results for the above four cases are summarized in the following table. All errors are given in mm.[3]

Case	Δ^1_{self}	Δ^2_{self}	Δ^1_{ball}	Δ^2_{ball}	$\Delta^{\text{fuzzy}}_{\text{ball}}$
1	110	230	78	103	85
2	186	439	117	609	106
3	n/a	n/a	940	502	180
4	312	566	n/a	842	862

[3] When the robots are moving (case 3) a source of ground truth of the robot positions was not available.

These results show that the estimates obtained by our fusion approach are, in practice, at least as good as the best individual estimate. When both robots gave good estimates (case 1) the fused estimate was as good as these. When one robot had a bad estimate due to localization errors (case 2), that robot's estimate was discounted in the fusion process, and the result was mostly influenced by the other robot's estimate. When both robots suffered from large localization errors (case 3), the fused estimate was at least as good as the individual estimates, and it was sometimes much better. When only one robot could see the object (case 4) the fused estimate was similar to the one from that robot.

The next table compares the error obtained by our fuzzy fusion technique with the error obtained by taking a simple average or a weighted average of the individual estimates (denoted by $\Delta_{\text{ball}}^{\text{avg}}$ and $\Delta_{\text{ball}}^{\text{wavg}}$, respectively). The weights in the latter case were given by the respective measure of reliability in self-localization. The errors refer to Case 3, above. As it can be seen, fuzzy fusion substantially outperformed averaging techniques in this situation.

Δ_{ball}^{1}	Δ_{ball}^{2}	$\Delta_{\text{ball}}^{\text{avg}}$	$\Delta_{\text{ball}}^{\text{wavg}}$	$\Delta_{\text{ball}}^{\text{fuzzy}}$
940	502	339	354	180

6 Conclusions

Our technique for multi-robot cooperative object localization has a number of advantages: it provides each robot with estimates which are, in practice, at least as good as the best ones available to any individual robot; it can effectively discount unreliable information; and there are no parameters to tune. The distinctive points of our approach are:

- The use of a sound technique to propagate the uncertainty in the self-localization of each robot to the uncertainty in object locations, and
- The use of an agreement seeking (fuzzy) operator instead of an averaging (e.g. probabilistic) operator to fuse the information provided by different robots.

We are currently using this technique in our RoboCup team [4]. Sharing ball information greatly improves the performance of robots in this domain. All robots can know the position of the ball if at least one member of the team sees it. Moreover, the fact that all the robots in the team have the same information about the ball position makes team coordination easier and more effective. The experiments reported here show that our technique produces useful results, even in the presence of high uncertainty. We are currently running a more systematic experimental evaluation of our approach, which includes empirical comparisons with other techniques.

Our future work is aimed at improving several aspects of our technique. First, our current approach is relatively demanding in terms of computation and bandwidth. Although we can run our algorithms in real time on the AIBO robots and using the limited bandwidth allowed by the RoboCup rules, it might be interesting to devise approximations of our technique which reduce the computational burden and/or the amount of information exchanged. Second, we would like to extend our technique to include sharing of information about multiple objects (e.g. opponent robots). Finally, we plan to investigate the extension of our approach to include cooperative self-localization, by merging the self-location grid of each robot with position grids from observing robots.

Acknowledgments

This work was supported by the EC Marie Curie Program, the Swedish CUGS (computer graduate school), the Consejería de Trabajo y Política Social (Región de Murcia) and the European Social Fund through the Seneca Foundation.

References

1. I. Bloch. Information combination operators for data fusion: A comparative review with classification. *IEEE Trans on Systems, Man, and Cybernetics*, A-26(1):52–67, 1996.
2. J.E. Bresenham. A linear algorithm for incremental display of circular arcs. *Communications of the ACM*, 20(2):100–106, 1977.
3. P. Buschka, A. Saffiotti, and Z. Wasik. Fuzzy landmark-based localization for a legged robot. In *Procs of the IEEE Int Conf on Intelligent Robots and Systems (IROS)*, pages 1205–1210, 2000.
4. Team Chaos. Team Chaos web site. http://www.aass.oru.se/Agora/RoboCup.
5. Sony Corporation. AIBO entertainment robot web site. http://www.aibo.com.
6. M. Dietl, J.-S. Gutmann, and B. Nebel. Cooperative sensing in dynamic environments. In *Procs of the IEEE Int Conf on Intelligent Robots and Systems (IROS)*, pages 1706–1713, 2001.
7. H. F. Durrant-Whyte. *Integration, Coordination, and Control of Multi-Sensor Robot Systems.* Kluver Academic Publishers, 1988.
8. The RoboCup Federation. RoboCup official web site. http://www.robocup.org.
9. D. Fox, W. Burgard, and S. Thrun. Markov localization for mobile robots in dynamic environments. *Artificial Intelligence Research Journal*, 11:391–427, 1999.
10. J.-S. Gutmann. Markov-Kalman localization for mobile robots. In *Procs of the Int Conf on Pattern Recognition (ICPR)*, pages 601–604, 2002.
11. R. Hanek, T. Schmitt, M. Klupsch, and S. Buck. From multiple images to a consistent view. In P. Stone, T. Balch, and Gerhard Kraetzschmar, editors, *Robocup 2000: Robot Soccer World Cup IV*, pages 169–178. Springer, 2001.
12. P. Marcelino, P. Nunes, P. Lima, and I. Ribeiro. Improving object localization through sensor fusion applied to soccer robots. In *Proc of the Scientific Meeting of the 3rd Robotics National Festival (Robotica 2003)*, Lisbon, Portugal, 2003.
13. M. Roth, D. Vail, and M. Veloso. A world model for multi-robot teams with communication. In *Procs of the IEEE Int Conf on Intelligent Robots and Systems (IROS)*, 2003.
14. S. I. Roumeliotis and G. A. Bekey. Synergetic localization for groups of mobile robots. In *Proc. IEEE Conf. on Decision and Control*, pages 3477–3482, 2000.

15. A. Saffiotti. The uses of fuzzy logic in autonomous robot navigation. *Soft Computing*, 1(4):180–197, 1997. Online at http://www.aass.oru.se/˜asaffio/.
16. T. Schmitt, R. Hanek, S. Buck, and M. Beetz. Cooperative probabilistic state estimation for vision-based autonomous soccer robots. In A. Birk, S. Coradeschi, and S. Tadokoro, editors, *Robocup 2000: Robot Soccer World Cup IV*, pages 193–203. Springer, 2002.
17. S. Spors and N. Strobel. A multi-sensor object localization system. In *Int Workshop on Vision, Modeling and Visualization (VMV)*, pages 19–26, 2001.
18. A. Stroupe, M. Martin, and T. Balch. Merging gaussian distributions for object localization in multi-robot systems. In *Procs of the Int Symposium on Experimental Robotics (ISER)*. Springer, 2000.
19. S. Weber. A general concept of fuzzy connectives, negations and implications based on t-norms and t-conorms. *Fuzzy sets and systems*, 11:115–134, 1983.
20. L. A. Zadeh. Fuzzy sets. *Information and Control*, 8:338–353, 1965.
21. L. A. Zadeh. Fuzzy sets as a basis for a theory of possibility. *Fuzzy Sets and Systems*, 1:3–28, 1978.