Proc. of the IEEE Int. Conf on Methods and Models in in Automation and Robotics (MMAR) Miedzyzdroje, Poland, 2003. To appear.

COOPERATIVE OBJECT LOCALIZATION USING FUZZY LOGIC

J.P. Cánovas [1], K. LeBlanc [2], A. Saffiotti [2]

[1] Dept. of Information and Comm. Eng., University of Murcia, 30100 Murcia, Spain

[2] AASS Mobile Robotics Lab, Dpt. Technology, Örebro University, 70182 Örebro, Sweden

Abstract. Cooperative localization of objects is an important challenge in multi-robot systems. We propose a new approach to cooperative object localization by a group of communicating robots. In our approach we see each robot as an expert which provides unreliable information about the location of objects. The information provided by different robots is combined using fuzzy logic techniques, in order to reach agreement between the robots. This contrasts with current techniques, which average the information provided by different robots, and can incur well-known problems when information is unreliable. We have tested our technique on a team of Sony AIBO robots in the RoboCup domain. We present experimental results obtained by sharing information about the location of the ball.

Keywords: Cooperative robotics, Fuzzy logic, Multiple observers, Cooperative perception.

1. INTRODUCTION

Cooperating robots can benefit from exchanging information about objects perceived in the environment. One important aspect of this is the problem of establishing the positions of these objects. Different robots may see the same objects from different points of view, and with different levels of reliability. The problem of cooperative object localization is the problem of fusing the information provided by different robots in order to reach agreement about the positions of objects. Fusion of information can improve the perception of each individual robot, and can give a robot information about objects which it does not see directly. Fusion of information, however, can result in degradation of 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, the result will be worse than what A would have established alone.

Many current approaches to cooperative object localization fuse information received from multiple observers using some sort of weighted average, often implemented in a Kalman filter (e.g., [5, 6, 7]). However these methods do not typically provide a robust solution in the presence of outliers.

Gating strategies can improve robustness by discarding observations which are deemed invalid. However a simple gating strategy does not completely solve the problem. 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 do not correspond with the current belief about the state of the world.

One way to deal with outliers and false positives is to implement some form of *voting scheme*, which encourages belief in what the majority of robots perceive. For instance, Dietl et al. [4] use a grid-based version of Markov localization to filter out outliers. Observations deemed valid by the Markov process are integrated into a Kalman filter. This approach simplifies the decision concerning which observations should be discarded, while preserving the accuracy of Kalman filtering. However depending on how one tunes the Markov filter, false positives or outliers could still be allowed to affect the result, or valid observations might be discarded.

In this paper, we propose a new approach to cooperative localization, based on fuzzy logic. We see each robot as an expert, which provides unreliable information about the location of objects. Fuzzy logic allows us to combine the information provided by different robots in order to reach *agreement*. In the rest of this paper, we describe our technique, discuss how we have implemented it in the Sony AIBO robots in the RoboCup domain, and present experimental results.

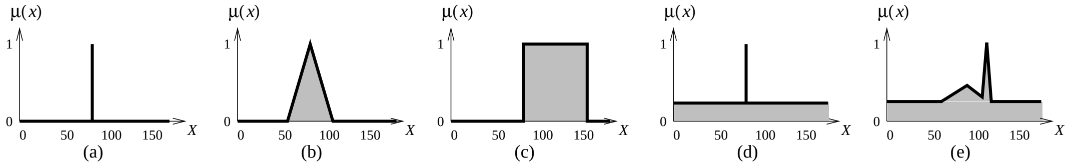

Figure 1: Different types of fuzzy locational information.

2. REPRESENTING LOCATIONAL INFORMATION

2.1. Fuzzy Locational Information

Locational 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 $(81, 132)$". The statement (b) "The object is about 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 unique position. And the statement (d) "The object was seen yesterday at position $(81, 132)$" is *unreliable*, as the object may no longer be there. An ideal uncertainty representation formalism should be able to represent all of the above statements. Perhaps more importantly, it should account for the differences between these statements, by representing the information at the level of detail which is available.

Fuzzy logic techniques are attractive in this respect. We can represent information about the location of an object by a fuzzy subset μ of the set X of all possible positions. 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), (b), (c) and (d) correspond to the four items of information mentioned above. Case (d) 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 (e) shows a combination of the previous types of uncertainty.

2.2. 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 sensors in the same robot, or by multiple robots.

If locational information is represented by fuzzy sets, fusion can be performed by fuzzy intersection between these sets. 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 $\mu = \mu_1 \cap \mu_2$ defined by

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

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

The second fact to note is that fuzzy fusion automatically discounts unreliable information. Consider the example shown in Fig. 2 (c). 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.

We sometimes need to extract a point estimate from the locational information μ, e.g., to be used in other navigation modules. A common approach to do this is by computing the center of gravity of μ:

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

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

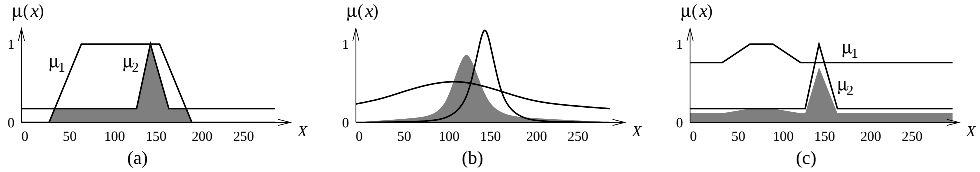

Figure 2: (a) Fuzzy fusion. (b) Probabilistic fusion. (c) Discounting unreliable information in fuzzy fusion.

2.3. Computational Aspects

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 number in $[0, 1]$ 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: for instance, we use a 3-D grid to represent the belief of the robot about its own pose in the environment.

A position grid must be associated to a reference system. In this work, we consider a global, robot-independent reference system F_g. This simplifies the exchange of locational information between different robots. Each robot r, however, acquires perceptual data from its own point of view, and it estimates the position 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 $T_r^g : F_r \to F_g$.

The pivot for the above transformation is the robot's pose in the global frame. In our case, this pose is given by a fuzzy set μ_r in the 3-D space of (x, y, θ) positions and orientations, so the computation of the transformation is more complex. Assume that the robot has observed an object at polar coordinates (ρ, ϕ) with respect to its own reference system F_r. Then, the degree of possibility $\mu(p)$ that the object is at global position $p = (x, y)$ is given by:

$$\mu(p) = \sup\{\mu_r(q, \theta) \mid d(\vec{pq}) = \rho \text{ and } \angle(\vec{pq}) = \phi{+}\theta\},$$

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

The above fuzzy set μ can be computed by the following algorithm.

```
foreach cell p
   μ(p) := 1 - reliability(observation)
   foreach cell q such that dist(p, q) = ρ            (3)
      μ(p) := max{μ(p), μ_r(q, (∠(p, q) − φ)}
end
```

The resulting distribution μ may contain a "bias", that is, its minimum value β is strictly positive. This indicates some unreliability and it can come from two sources: (i) unreliability in perceptual information, represented by 'reliability(observation)'; and (ii) unreliability in the self-localization of the robot, represented by the bias in μ_r which is propagated to μ.

3. SHARING LOCATIONAL INFORMATION

Fuzzy locational 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, which is represented and fused using the techniques described previously. The following diagram summarizes our fusion schema in the case of two robots — the extension to n robots is straightforward.

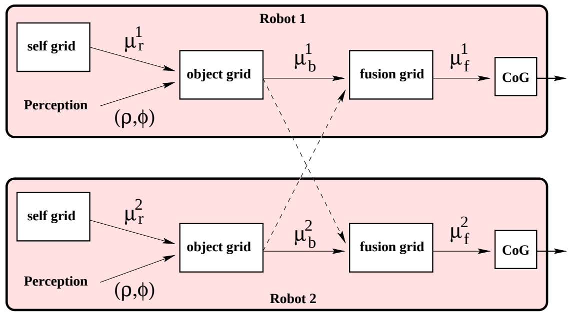

Figure 3: Schema for fusion of locational information.

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 about the global location of the target object, derived from perceptual observation by algorithm (3) above. (More grids may be needed if we are interested in several objects at the same time.) The "fusion grid" contains the result of the fusion of locational information computed by different robots. This grid is kept separate from the "object grid" in order to avoid circular dependencies in the information exchanged between robots. A final defuzzification step obtains a point estimate, if needed, using formula (2). Temporal aspects aside, the final estimate of the object's location computed by the two robots will be identical.

Fusion of information from different robots is performed by fuzzy intersection of the distributions provided by the two robots, 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 both 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 which know it) is automatically discounted. In particular, if a robot has no current perception of the object, all the cells in its object grid will have values close to 1, so these values will not (significantly) affect the result of the fusion.

The above schema assumes that the full position grids μ 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 experiments in order to reduce the complexity. With these devices information sharing among four robots can be run entirely on-board, in real time and with limited bandwidth.

4. THE ROBOCUP EXAMPLE

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

4.1. Implementation

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

The computation of the global ball position from observation is done according to an optimized version of algorithm (3). 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 a distance ρ from p. This is equivalent to saying that we only need to evaluate the cells in the circle with center q 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 grey-scale image. We then compress the grid using a simple run length encoding scheme. The ball grid is only sent by a robot if its information is new and/or of better quality than the one last sent.

4.2. Static Experiments

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. We presents the results of three different cases, corresponding to three different ball positions.

Case 1 Both robots can see the ball, and have accurate self-localization. This case is shown in Fig. 4. The left window shows the ball grid resulting from the fusion. Darker cells have higher degrees of possibility. The two triangles represent the robot's estimates of their own positions. The three small circles near the bottom represent the point estimates of the ball position according to each robot (lighter circles) and as a result of the fusion (darker circle). All estimates are very close to the real position of the ball. The middle and right windows show the self-localization grids for robots 1 and 2, respectively.

Case 2 The ball is behind robot 1, so only robot 2 can see it — see Fig. 5. The self-localization of robot 1 is fair but that of robot 2 is poor. The fused information is similar to that provided by robot 2 alone. While this information suffers from the poor localization of robot 2, it is made available to robot 1 which would otherwise have no knowledge of the ball position.

Case 3 Both robots can see the ball, but while robot 1 is well localized, robot 2 is not — see Fig. 6. The result of the fusion in this case is close to the information provided by robot 1, while the information provided by robot 2 is discounted since it is based on poor localization.

4.3. Dynamic Experiments

When robots move during a real game, self-localization may become very poor due to several factors: the robots are busy tracking the ball and do not often see the field landmarks, the motion of the legged robot is poorly modeled, and the tilting and rolling of the robot's body introduces inaccuracies in perception. We performed a second set of experiments in which we used two constantly moving robots, and we placed the ball at various known positions.

Fig. 7 shows a sample situation taken from these experiments. Both robots have rather poor self-localization, as can be seen from the blurring of the two individual self-grids in the middle and right windows. Correspondingly, the individual estimates for the ball positions are relatively inaccurate, and quite different from each other. When intersecting the fuzzy sets, however, we obtain a fairly accurate fused estimate of the ball position (left window).

[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].

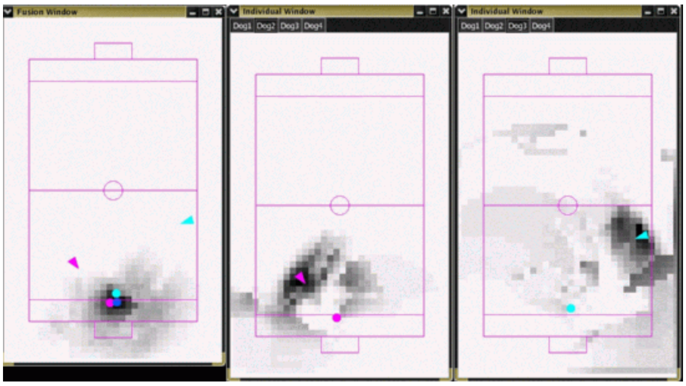

Figure 4: Experiment 1, Case 1. Both robots are well localized and can see the ball.

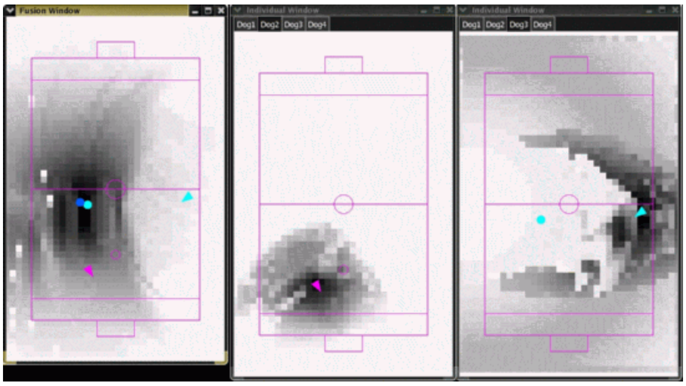

Figure 5: Experiment 1, Case 2. Only robot 2 can see the ball.

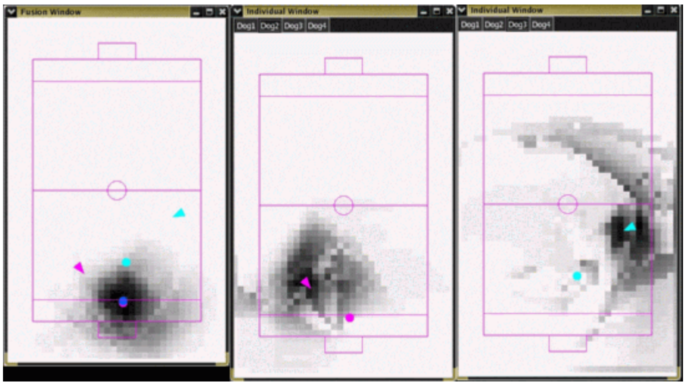

Figure 6: Experiment 1, Case 3. Second robot has a bad self-localization.

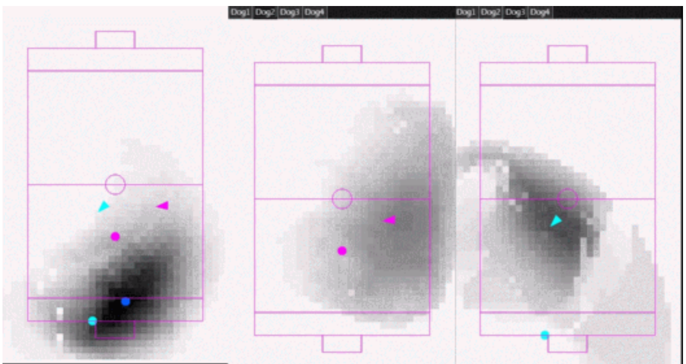

Figure 7: Experiment 2. Combining two blurred estimates gives an accurate one.

4.4. Results

For each static 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 $\Delta^{\mathrm{fuzzy}}_{\mathrm{ball}}$. Errors were measured by comparing the center of gravity of the fuzzy location sets with the ground truth. The results for the three cases presented above are summarized in the following table. All errors are given in mm.

case	Δ^1_{self}	Δ^1_{ball}	Δ^2_{self}	Δ^2_{ball}	$\Delta^{\mathrm{fuzzy}}_{\mathrm{ball}}$
1	*110*	78	*230*	103	85
2	*312*	n/a	*566*	842	862
3	*186*	117	*439*	609	106

These data show that the estimates obtained by our fusion approach are, in practice, at least as good as the best estimate obtained by each robot individually. When both robots give good estimates (case 1) the fused estimate is as good. When only one robot sees the object (case 2) the fused estimate is similar to the one from that robot. When a robot has a bad estimate due to a large localization error (case 3), that estimate is discounted in the fusion process and the result is mostly influenced by the other robot's estimate.

The results from the dynamic experiments were similar to the static ones. The quality of these estimates, however, was much worse due to poorer self-localization when the robots are moving. The following table quantifies the errors in the case shown in Fig. 7.

Δ^1_{ball}	Δ^2_{ball}	$\Delta^{\mathrm{avg}}_{\mathrm{ball}}$	$\Delta^{\mathrm{wavg}}_{\mathrm{ball}}$	$\Delta^{\mathrm{fuzzy}}_{\mathrm{ball}}$
940	502	339	354	180

This table also reports the errors obtained by making a simple average or a weighted average of the individual estimates, denoted by $\Delta^{\mathrm{avg}}_{\mathrm{ball}}$ and $\Delta^{\mathrm{wavg}}_{\mathrm{ball}}$ respectively. (The weights in the latter case were given by the measure of reliability in self-localization.) As it can be seen, fuzzy fusion substantially outperformed averaging techniques in this situation.

5. CONCLUSIONS

Our technique for multi-robot, cooperative object localization has a number of good points: it provides estimates which are, in practice, at least as good as the ones available to each single robot; it can effectively discount unreliable information from badly localized robots; and there are no parameters to tune. We have shown that this technique produces good results in actual experiments.

We are currently using this technique in RoboCup. 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 ball position makes team coordination easier and more effective.

There are two obvious ways in which our technique could be improved. First, the technique is relatively demanding in terms of computation and bandwidth. Although we can run our algorithms in real time on-board the Aibo robots, it might be interesting to devise approximations of our technique that reduce the computational burden and/or the amount of information exchanged between robots. Second, the discounting of unreliable information depends on the fact that the sending robot correctly estimates the reliability of the information that it is providing. It would be useful to be able to judge the reliability of information inside the fusion process itself. Our current work is addressing both problems.

Acknowledgements

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 Found 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. Bushka, A. Saffiotti, and Z. Wasik. Fuzzy landmark-based localisation for a legged robot. In *Proc. of IROS 2000*, pages 1205–1210, 2000.

4. M. Dietl, J.-S. Gutmann, and B. Nebel. *Cooperative sensing in dynamic environments*. *Proc. of IROS 2001*, pages 1706–1713, 2001.

5. R. Hanek, T. Schmitt, M. Klupsch, and S. Buck. From Multiple Images to a Consistent View. In *RoboCup-2000*, pages 288–296. Springer, 2000.

6. S. I. Roumeliotis and G. A. Bekey. Synergetic localization for groups of mobile robots. In *Proc. IEEE Conf. on Decision and Control*, Sydney, Australia, pp. 3477–3482, 2000.

7. A. Stroupe, M. Martin, and T. Balch. Merging gaussian distributions for object localization in multi-robot systems. *Proc. of ISER 2000*, 2000.