Proceedings of the First International Conference on Autonomous Agents

Marina del Rey, CA USA
February 5-8, 1997

Edited by W. Lewis Johnson

Sponsored by ACM SIGART

Co-sponsored by ACM SIGGRAPH and SIGCHI, AAAI, British Telecommunications, Intel, Microsoft, MERL — A Mitsubishi Research Laboratory, and USC / Information Sciences Institute

In cooperation with ECCAI and IEEE Computer Society

Generation of Unknown Environment Maps by Cooperative Low-Cost Robots

R. López de Màntaras[2], J. Amat[1], F. Esteva[2], M. López[2] and C. Sierra[2]

(1) Automatic Control Department. Universitat Politècnica de Catalunya
c/ Pau Gargallo, 5, 08028 Barcelona (Spain)
amat@esaii.upc.es
(2) IIIA - Artificial Intelligence Research Institute, CSIC.
Campus UAB, 08193 Bellaterra (Spain)
{esteva,mantaras,maite,sierra}@iiia.csic.es

Abstract

In this paper we present some results obtained with a troupe of low-cost robots designed to cooperatively explore unknown structured orthogonal environments. In order to improve the covering of the explored zone the robots show different behaviours (routine, normal and anxious) and cooperate by transferring each other the perceived environment when they meet; therefore, not all the information of the non-returning robots is lost provided that they had encountered robots that safely returned. The returning robots deliver to a host their perceived and communicated (by other robots) partial maps and the host incrementally generates the most plausible map of the environment. To perform the map generation, a fusion, completion and alignment process of the partial maps, based on fuzzy techniques, has been developed.

Introduction

With the aim to explore an environment that is unknown but easily passable, a troupe of low cost, small autonomous robots has been developed. These robots follow the already classical line of insect robots (Alami et al., 93) (Brooks, 86, 91). The goal of these autonomous robots is to obtain partial information about an orthogonal environment during their exploration runs and afterwards to supply this information to a host computer that will compute the most plausible map. Using this multi-robot strategy to generate a model of the environment, we expect to achieve a better efficiency than that which would be obtained based only on a single expensive robot.

The behaviour of these small autonomous robots has been programmed in an individual basis and is similar -to some degree- to that of ants in two aspects. First, in order to increase the coverage of the environment, the robots have a partially random moving behaviour; and second, the robots cooperate by transferring each other the perceived environment when they meet. Sharing information in this way, allows the host to get the information not only from the robots that successfully return after an exploratory run, but also some information from those that could not return, provided that they had encountered robots that safely returned.

The next section of this paper describes the structure and the behaviour of the robots. Then, we describe the statistical error analysis performed in order to know how the error intervals increase with the distance run and the number of turns. This analysis will be used to model the imprecise location of obstacles and walls by means of fuzzy techniques. The next section describes the fuzzy logic-based algorithms that we have developed in order to compute the most plausible map based on the partial maps perceived by the successfully returning robots. Finally we describe the results obtained to date, we briefly point to related work and we mention some future work.

Autonomous Agents 97, Marina Del Rey, California USA

Structure of Each Mobile Robot

Each robot has been designed with the aim of being small and cheap. They must have a high autonomy and be endowed with a low cost processor to memorise the perceived environment map. All these requirements have lead to a compromise solution consisting on small robots with three wheels. Two of them are steering wheels having independent motors.

The robots environment perception system and the communication with the host or with other robots is based on IR impulse modulated sensors. Since some of the robots may not be able to return to deliver their map to the host, the following communication process is established: when two of them meet along their exploration run, they back-up from one to the other all the information they have acquired so far from the environment. In this way, when a robot reaches the host, it delivers both, the information acquired during its own run as well as the information obtained from other robots that it has encountered. This communication process allows to get all the information of those non-returning mini robot that had been transferred to returning ones.

Mechanical Characteristics

Each robot is 21 cm. length and 15 cm. wide (see Fig. 1). The 5 cm. driving wheels allow to pass some small obstacles such as carpets or electrical wires. The robots can reach a maximum speed up of 0.6 m/sec., and since the battery has about one hour of autonomy, each robot can do, in principle, an exploration of about 2000 m. maximum.

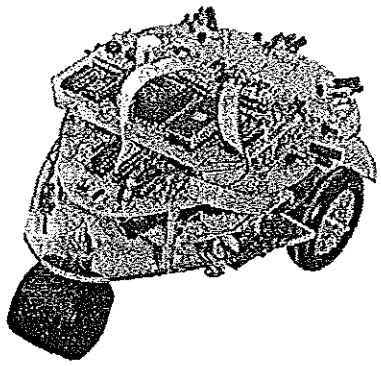

Figure 1. Autonomous Mini-Robot

Sensing Capability

Each robot is equipped with the following sensors:

- Impulse generators at each wheel for odometry.
- Five I.R. proximity sensors for frontal obstacles detection and for wall following.
- A proximity sensor for the detection of the terrain horizontal discontinuities.
- Safety micro switches for the detection of possible collision.
- One omnidirectional IR Emitter/Receiver sensor to detect other robots and to transmit data
- One IR Emitter with a scope of 90 degrees to generate a priority signal (right hand preference)

Navigation Strategy

The navigation system incorporated to each robot has a partially random behaviour: The robot does a $\pm 45^{o}$ or $\pm 90^{o}$ turn either, randomly or when it detects an obstacle.

The random turns are done with significantly different probabilities: $P_1 > P_2 > P_3$, corresponding to three differentiated behaviours:

P_1= robot with an "Anxious" behaviour

P_2= robot with "normal" behaviour

P_3= robot with "routine" behaviour.

When the robot finds a frontal obstacle, the turn can be done to the right or to the left based also on a probability value P_4. The robots having a probability $P4<0.5$ will show a tendency to turn to the right more often than to the left, whilst the robots having a probability $P4>0.5$ will behave inversely.

Consequently, the different robots of the exploration troupe will not show an identical behaviour. They can behave in six different ways corresponding to the different combinations of behaviours and turning tendencies.

The algorithm used to guide the robot back towards the starting point after running for a given amount of time, consists on returning following an improvement of the travelled path contained in memory (in order to assure a possible path free of obstacles). This improvement consists in eliminating loops.

Control System

The control unit in each robot has been designed having in mind that the hardware had to be as simple as possible but, on the other hand, it had to allow achieving a behaviour sufficiently smart in order to navigate efficiently. Furthermore the robot had to be based on a hardware flexible enough to allow for experimentation of navigation and control strategies. These requirements have resulted in a design shown in figure 4 which contains three different functional modules : *the navigation module* that generates the trajectory to be followed; *the steering module* that controls the motors in order to follow the generated trajectory; and *the perception module* that acquires information of the environment by means of IR sensors. However it is possible to replace this module by other modules adapted to different types of sensors.

The computer used to implement the navigation control unit is a 80C186 with a 1MB RAM to store the perceived environment map. This map is discretized with a resolution of 4 cm which means that each robot can store maps of up to 40x40 m.

The steering control module operates with a much higher resolution since each encoder corresponds to a displacement of only 2 mm. and it is implemented on a 80C552

Error Analysis

With the goal of studying the position error of each robot due to the imprecise odometry and to the imprecise steering, we have performed an analysis based on experimental data obtained from the real robots running straight (10 feet and 20 feet) and also turning 45 degrees left and 45 degrees right followed by a 10 feet straight run. We have performed 20 trials of each run and turning situation for each robot. With the data obtained, we have used the Kolmogorov normality test to verify that the experimental sample indeed follows a normal distribution both in the direction of the trajectory and in the direction perpendicular to the trajectory and we have tested that both distributions are independent. Based on this distributions we have determined the size of an error rectangle, comprising the 95% of the sample, associated to the final position of the robot after a straight run of 10 feet. This rectangle is 2.5 inches (in the direction of the trajectory) x 11 inches (in the direction perpendicular to the trajectory) in the average. We have also experimentally concluded that the size of the error rectangle is proportional to the covered distance. Concerning the additional error due to turning, we have obtained that when the robots turn 45 degrees there is, in the average, an error of about 2 degrees always towards the same direction. For example a robot with 2 degrees of error towards the left turns 43 degrees to the right instead of 45 degrees and turns about 47 degrees to the left instead of 45 degrees.

Error Propagation

In free space, a trajectory is composed of a set of alternating segments and turns. Given the error rectangle at the initial point of a trajectory, we want to determine the error rectangle at the end of each segment taking into account the turning error and the error accumulated along the segment. The next figure shows the error propagation after a right turn, a straight line, another right turn and finally another straight line.

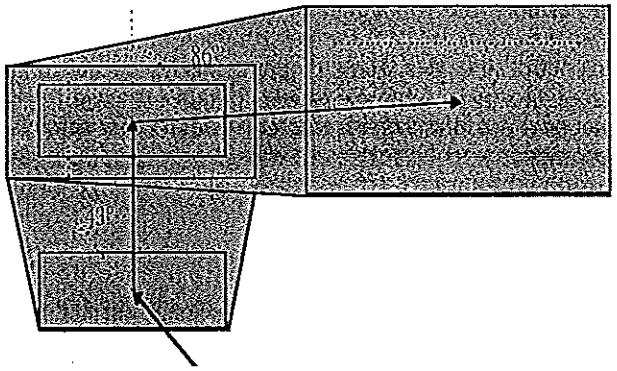

Figure 2. Error propagation

When following a wall, since the robot remains practically always at the same distance from the wall, the error along the direction orthogonal to the wall is taken to be constant and equal to the error that the robot has after turning to place itself parallel to the wall once the wall has been detected. This error analysis and error propagation study is performed on each robot and is used by the host to model the imprecise location of the detected walls and obstacles by means of fuzzy sets as described in the next section.

Environment Map Generation

The goal of map generation is to obtain the most plausible position of walls and obstacles based on the assumption that the environment is orthogonal, that is, the corners between walls are only angles of 90 degrees as well as the angles between any two connected sides of the obstacles. The information about the position of walls and other obstacles conveyed by the robots is imprecise due to the error in the robot position. Furthermore, different robots (or the same robot visiting twice) can detect portions of the same wall or obstacle with different degrees of imprecision. The main problem is to decide whether several detected portions (represented by imprecise segments, see next section) are of the same wall or obstacle or not. This decision depends on the relative position of the segments as well as on their imprecision. The relative position of the segments can be represented by their distance. The distance is compared to a threshold to decide whether the segments represent different portions of the same wall or obstacle. If two segments represent the same wall or obstacle a segment fusion procedure is applied to produce a single segment. This process of segment fusion is followed by a completion process in which hypothesis are made with respect to non observed regions. The completion process is achieved by means of hypothetical reasoning based on declarative heuristic knowledge about the structured environments in which ants evolve.

The map generation algorithm consists of two steps. The first one is the fusion of the map perceived by each robot, taking into account that the same robot can observe more than one portion of the same wall. The second step consists in a global fusion and completion of the maps perceived by several robots. The fusion function is the same in both steps. However, since not all troupe members may return to give the information to the host and since, on the other hand, we want to have at any time the most plausible map based on the information obtained so far, it is necessary to develop an incremental approach to the map generation algorithm. For any returning robot, the algorithm is executed to update the map. The overall algorithm is as follows:

```
Function
 Map_generation(NewAntsMap,CurrentMap)=
  Begin
     NewCurrentMap=Fusion(CurrentMap ∪
     Fusion(NewAntsMap,α,β),α,β);
     Return Completion(NewCurrentMap)
  End
```

where α and β are decision threshold parameters that will be explained later. In the following sections we define the imprecise segments in terms of fuzzy sets, we give details of the segment fusion procedure and we outline the completion process.

Imprecise Segments

Let us start by defining the basic object of our approach which is the imprecise segment. An imprecise segment S is a function that gives for any Cartesian coordinates (x, y) the degree of possibility of the coordinates as being part of a wall. That is $S: R \times R \rightarrow [0, 1]$. This function can be seen as a possibility distribution (López de Mántaras, 1990) in the sense of fuzzy logic (Zadeh, 1978) and is determined by a segment, that for simplicity, we assume has a precise length, plus an error in its position with respect to both axis. This error is the one obtained experimentally as explained earlier and is the support of the triangular fuzzy set modelling the imprecision of the segment. In fact the support is "2e" along the direction orthogonal to the trajectory and $e_1 + e_2$ + length of the segment in the direction of the trajectory (see figure 3). Since, while following a wall, the distance of the robot to the wall remains practically constant, the error, e, in the direction orthogonal to the trajectory remains constant, however, as we have explained in the error analysis section, the error along the direction of the trajectory increases with the distance travelled. That is, assuming in figure 3 a movement from left to right, we would have in fact that the error on the right would be higher than that on the left of the figure. Furthermore, for each segment we keep the list of coordinates of the singular points that have been detected (i.e. corners and gaps in the real world).

Given the orthogonality of the structured environment, we have only two possible orientations for the segment : vertical or horizontal. For example, an imprecise horizontal segment S = ((x_1 , y), (x_2 , y), e, e_1 , e_2) means that there is a portion of a wall with coordinates ((a b) (c b)) such that y - e < b < y + e and $x_1+e_1 > a > x_1 - e_1$ and $x_2-e_2 < c < x_2 + e_2$. Where $e_2 > e_1$ Similarly for the orthogonal case. For notation simplicity, in the sequel we will use the following notation for an imprecise segment S = ((x , y), (x , y), e) i.e. we use a single "e" to denote the errors.

Figure 3 shows an example of the fuzzy modelling of an imprecise horizontal segment.

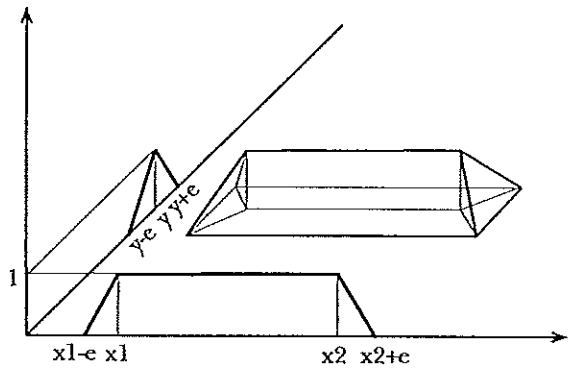

Figure 3. Horizontal Imprecise segment.

Segment Fusion

As we have said, the main problem is to decide whether several imprecise segments are of the same wall or obstacle or not. This decision depends on the relative position of the imprecise segments like for example the case shown in figure 4. This relative position is represented by two parameters Δx and Δy that represent the minimum distance with respect to the two axis. For example in the case of the two parallel vertical segments of figure 4 we have:

$$s = ((x, y_1),(x, y_2), e), s' = ((x', y'_1),(x', y'_2), e')$$

$$\Delta x(s, s') = |x - x'|$$

$$\Delta y(s, s') = \begin{cases} 0 & if\ (y_1 - y'_2) * (y_2 - y'_1) < 0 \\ \min(|y_1 - y'_2|, |y_2 - y'_1|) & otherwise \end{cases}$$

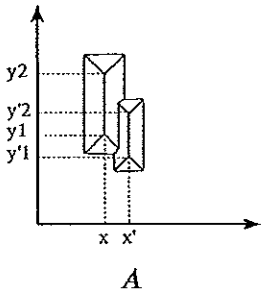

Figure 4. Example of two overlapping imprecise segments.

In the Δy expression, the first part stands for the case in which both segments overlap. In that case both terms in the product have a different sign.

Given a map M, represented as a list of segments ordered by imprecision, the "Fusion" function computes a new map M' whose difference with M is that some segments have been fused. For each horizontal (resp. vertical) segment s in M, starting with the most precise, we compute its distance to the other horizontal (resp. vertical) segments in M with respect to both axis. Then, with those segments in M such that both distances are below a threshold, the segment s' in M with the lowest average distance with respect to both axis is selected for fusion. Next we merge s and s' and we replace them by the resulting merged segment s" that is included in the list M in the appropriate place according to its error. Furthermore, the singular points are updated in order to keep the singular points of the merged segment properly connected with other segments meeting at these singular points. When no more fusions are possible, the process stops. The complexity of this algorithm is $O(n^2)$ where n is the number of segments.

Figure 5 shows an example of vertical segments fusion. The coordinates y"1 and y"2 of the fused segment s" are obvious. The x coordinate of s" is x" and is the centre of gravity of the masses 1/e and 1/e' of s and s' being e and e' the constant errors of s and s' along the direction orthogonal to the trajectory (see discussion above). The error e" of the fused segment is computed as a function of e and e'. We have used simply the sum of the masses, i.e. 1/e" = 1/e + 1/e', however alternative functions like 1/e" = max (1/e , 1/e') are also possible. These functions are given as parameters in our fusion algorithm. To clarify further this example, figure 6 shows the x projection of this fusion process for two vertical segments.

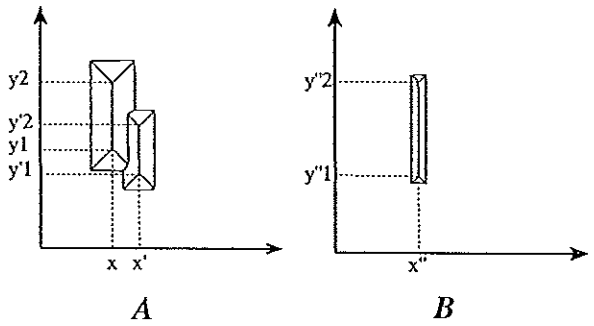

Figure 5. Vertical segments in A are merged into the segment in B.

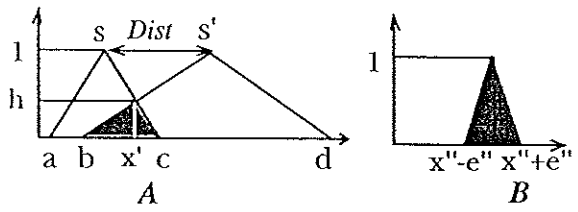

Figure 6. Merging of two vertical segment projections along the x axis.

Global Environment Map Completion

The fusion procedure is an initial step that only merges segments in a local basis using only geometrical relations between the segments. To improve the map, we perform a completion process based on: 1) a global view of the environment, including the map that each robot has got from the other robots that it has encountered during the exploration; 2) knowledge about the characteristics of the environment; and 3) the paths followed by the robots. We adopt an heuristic knowledge-based approach based on a set of rules covering the different situations that have been identified. These rules are divided into the following four rule sets :

- extending single segments
- combining two segments
- completing corners (L shaped and T shaped)
- completing doorways

and are considered in this order.

One example of heuristic rule for combining two vertical segments is:

```
If
  segments s1 and s2 are parallel and
  there are no singular points in the
  closest extremes of s1 and s2 and
  Δx (s1 , s2) < α and
  Δy (s1 , s2) < kβ and
  there is no path through the gap
  between s1 and s2

Then      Do (Vert_Merging (s1, s2))
```

Where α is the typical width of a doorway and β the typical width of a corridor and Vert_Merging is the function that takes care of the merging.

Next we give an example of a rule for completing "L-shaped" corners:

```
If
  s1 and s2 are perpendicular and
  there are no singular points in the
  closest extremes of s1 and s2 and
  Δx (s1 , s2) < kα and
  Δy (s1 , s2) < kβ and
  there are no paths through the gap
  between s1 and s2 and
  s1 and s2 do not meet at the corner
  point (x,y)

Then
  make s1 and s2 to meet at (x,y) and
  label (x,y) as singular point
```

Similar rules are used to extend single segments, to combine other situations involving two segments and to complete other situations of corners as well as doorways.

The very last step is to align segments that should but do not have the same x coordinate (vertical segments) or the same y coordinate (horizontal segments) because, for instance, they are at both sides of a doorway. To do this alignment we simply apply the Vert_Merging (resp. Hor_Merging) functions of the fusion process in order to compute the x (or y) coordinate of the aligned segments. These Merging functions are suitable because they weight the imprecision of the two segments in such a way that the aligned position is closer to the more precise of the two segments.

Results

Three prototypes of autonomous robots have been physically built and tested. Comparing the environment map with obstacles obtained through IR sensing in real operating conditions with the real map of the environment, it can be seen that the results are satisfactory.

Figures 7 shows one of the environments used. Figure 8 shows the status of the map after the information perceived by the first two returning robots has been fused, as well as the information perceived by a third robot before fusion; and figure 9 shows the final map obtained after applying the fusion process taking into account the information brought by the third returning robot. In figures 8 and 9, singular points are indicates by little circles. The three robots had each one a different behaviour, the third being the one with an anxious behaviour. We can see that the coverage of the environment is quite good. The obtained map is also a quite good approximation of the real one

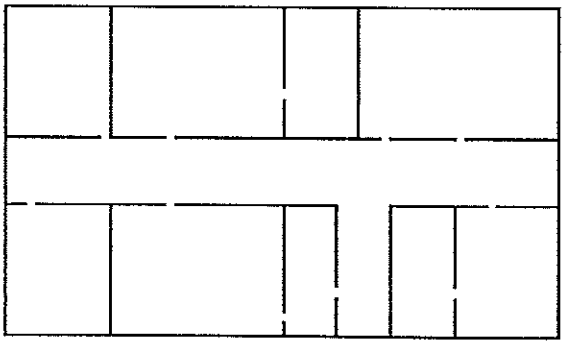

Figure 7. Example of environment

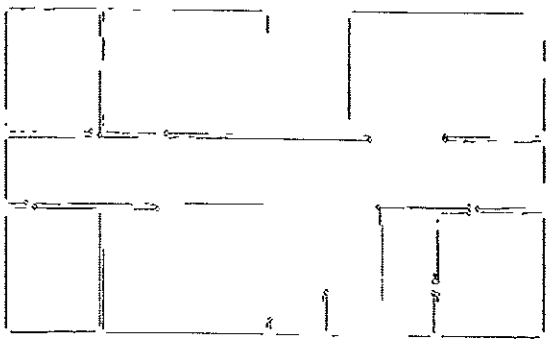

Figure 8. Fused Map from two returning robots plus information from a third one to be fused

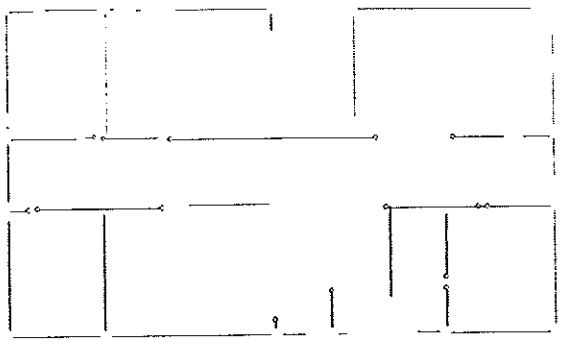

Figure 9. Final Map obtained by three robots

Relation to Other Work

Ultrasonic sensors have been used in map generation, however, due to the wide radiation angle, the uncertainties on angular location are too large. Another problem is that of false reflection (the beam may hit the receiver after bouncing with several obstacles); furthermore the ultrasonic beam can easily be lost due also to reflection when hitting obstacles with a large angle of incidence. Stochastic techniques (Elfes and Moravec 1985, Kuc and Siegel 1987, Hager 1990, Hwan et al 1992) have been used in modelling the ultrasonic sensors uncertainty when enough and well distributed data is available. Fuzzy sets are more versatile and computationally simpler than stochastic methods and have also been used by (Poloni et al. 1995) in map building based also on ultrasonic sensors but the obtained results are still too imprecise and for this reason in our work we have used IR sensors that are more precise.

Future Work

The results obtained with these prototypes are encouraging enough to undertake the construction of a whole troupe consisting of 15 robots. This troupe will allow us to perform exhaustive experimentation with different number of robots and with different behaviour combinations. Concerning map generation, future work will be focused on refining both the fusion and completion procedures based on the outcome of the experimental results that are being obtained. Another interesting future work will be to set up the conditions under which it may be advisable to send further robots to explore those portions of the environment that remain incomplete.

Acknowledgements. We acknowledge the help of Lluís Godo in performing the error analysis. Maite López research is supported by a doctoral scholarship of the CSIC.

References

Brooks, R.A. 1991 Intelligence Without Reason, In Proceedings of International Joint Conference on Artificial Intelligence, 569-595. Sydney, Australia: International Joint Conferences on Artificial Intelligence, Inc.

Elfes, A.; and Moravec, H.P. 1985. High resolution maps from wide angle sonar. In IEEE International Conference on Robotics and Automation, 116-121. St. Louis, MO.

Hager, G.D. 1990. *Task-Directed Sensor Fusion and Planning: A Computational Approach*. Norwell, MA: Kluwer Academic Publishers, International Series in Engineering and Computer Science Vol. SECS 99.

Hwan, J.; Dong, L.; and Cho, W. 1992. Physically based sensor modelling for a sonar map in a specular environment. In IEEE International Conference on Robotics and Automation, 1714-1719. Nice, France.

Kuc, R.; and Siegel, M.W. 1987. Physically based simulation model for acoustic sensor robot navigation, *IEEE Trans. Pattern Analysis Machine Learning* 9(6):766-778.

López de Mántaras, R. 1990. *Approximate Reasoning Models*. UK.: Ellis Horwood Series in Artificial Intelligence.

Poloni, M.; Ulivi, G.; and Vendittelli, M. 1995. Fuzzy Logic and autonomous vehicles: Experiments in ultrasonic vision. *Fuzzy Sets and Systems* 69:15-27.

Zadeh, A. 1978. Fuzzy sets as a basis for a theory of possibility. *Fuzzy Sets & Systems* 1:3-28.