AUTONOMOUS NAVIGATION TROUP FOR COOPERATIVE MODELLING OF UNKNOWN ENVIRONMENTS

Josep Amat*, Francesc Esteva and Ramon López de Màntaras****

* Automatic Control Department. Universitat Politècnica de Catalunya
c/ Pau Gargallo, 5, 08028 Barcelona (Spain)
phone: + 34 3 401 69 74; fax: +34 3 401 70 45; e-mail: amat@esaii.upc.es

** IIIA - Artificial Intelligence Research Institute, CSIC.
Campus UAB, 08193 Bellaterra (Spain)
phone: + 34 3 580 95 70; fax: +34 3 580 96 61; e-mail: {esteva,mantaras@iiia.csic.es}

Abstract *Based on the information gathered by a set of small autonomous low cost vehicles, the generation of unknown environment maps is described, as well as the vehicles themselves. In order to improve the covering of the explored zone the vehicles show different behaviours. The host that controls the troup of vehicles generates the most plausible map of the environment from the information obtained by the different components of the troup, which at the end of their mission return back. To perform the map generation a two-step algorithm, fusion and completion, based on fuzzy techniques is presented.*

1. INTRODUCTION

With the aim to explore an environment that is unknown but easily passable, a system constituted by a set of low cost, small autonomous vehicles has been developed. These vehicles follow the already classical line of insect robots [1] [2] [3] [4]. The goal of these autonomous vehicles is to obtain partial information about the environment during their exploration runs and afterwards to supply this information to a Master Autonomous Robot (Host) that in turn will be able to compute the most plausible map. With this information, the host should be able to perform a given mission within such structured environment. Using this Master-Multislave strategy to generate a model of the environment, we expect to achieve a better efficiency than that which would be obtained based only on the Master perception capabilities.

The behaviour of these small autonomous vehicles has been programmed in an individual basis for obtaining a behaviour similar -to some degree- to that of ants. Here, the similarity lies in the vehicles capacity to cooperate in the process of environment map generation. The sensing capability of each small vehicle, to build its own partial map, comes from two kind of sensors: IR proximity sensors for environment data acquisition, and a relatively accurate odometric system for the estimation of the vehicle position during its run.

In order to increase the global efficiency, the vehicles developped have a partially random moving behaviour. They can also transfer each other the perceived environment when they meet. Sharing data in this way, allows the Master to get the information not only from the vehicles that successfully return after an exploratory run, but also from those that cannot return, provided that they have encountered vehicles that have safely returned.

This paper first describes the structure and the behaviour of the vehicles. Then, we describe the fuzzy logic-based algorithms that we have developped in order to compute the most plausible map based on the partial maps built by the succesfully returning vehicles from the data perceived and received from other vehicles during their runs. Finally the obtained results are presented.

2. VEHICLES STRUCTURE

The vehicles (ANTs) have been conceived with the aim to cooperatively explore an unknown environment acquiring data from the path run and communicating with the other vehicles when they pass one close to another, in a way that could remind the ants behaviour. In order for the system to be efficient and flexible, the vehicles, which must acquire and integrate the environment information and transmit it to a host, have been designed to be small and economical. Each vehicle must have a high autonomy and be endowed with a computer, with which they can get and memorize the environment map with the highest resolution. All these requirements have lead to a compromise solution consisting of several small vehicles with three wheels. Two of them are steering wheels having independent motors.

The vehicles environment perception system and the communications with the host or with the other vehicles are based on IR impulse modulated sensors. Since some of them may get lost and fail 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 vehicle reaches the host, it delivers both, the information acquired during its own run as well as the information obtained from other vehicles that it has encountered during its run. This communication process allows to get all the information of a non-returning vehicle provided that it has had the opportunity to transfer it to a returning one.

The vehicles are 21 cm. long and 15 cm. wide. Figure 1 shows a view of one vehicle. The 5cm. driving wheels allow the vehicles to save some small obstacles such as carpets or electrical wires. With the motors utilised, the vehicles can reach a speed of up to 0.6 m/sec., and since the battery has an hour autonomy at full functioning, each vehicle can theoretically do a run about 1000 m. long.

Figure 1. An ANT autonomous vehicle

Each vehicle is equipped with the following sensing elements:

- Impulse generators at each wheel for odometry.
- Five I.R. proximity sensors for obstacles detection.
- A proximity sensor for the detection of the terrain horizontal discontinuities.
- One omnidirectional IR Emitter/Receiver sensor to detect other vehicles and to transmit its data.
- One IR Emitter with a sector scope of 90^{o} to generate the priority signal (right hand preference).
- Safety microswitches for the detection of possible collisions.

Figure 2. IR Sensors distribution

The obtained odometric precision produces a 2x20 cm. average uncertainty ellypse after a 10 meters run without direction changes. This uncertainty is added to that produced when the vehicle turns and both have been experimentally evaluated. The host will compute the uncertainty value relative to the location of walls and obstacles based on the data received from the returning vehicles as well as the information of the trajectories.

3. VEHICLE CONTROL SYSTEM AND BEHAVIOUR

Each vehicle navigation system is provided with a random control strategy that determines its behaviour. The vehicles can turn a $\pm 45^{o}$ or $\pm 90^{o}$ angle randomly during their run. When they find an obstacle or a wall they follow it using a "bang bang" control strategy

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

P_1= vehicle with an "anxious" behaviour

P_2= vehicle with a "normal" behaviour

P_3= vehicle with a "routine" behaviour.

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

Consequently, the different vehicles of the exploration troup will not show an identical behaviour. They can behave in six different ways corresponding to the different combinations of behaviours and turning tendencies. With this strategy it is expected to obtain a better covering of an environment than that we would obtain with all the vehicles showing the same behaviour.

Once a vehicle has run a length $L(P_i)$, depending on the probability P_1, P_2 or P_3, where $L(P_1)<L(P_2)<L(P_3)$, it starts its way back towards the host.

The algorithm used to guide the vehicle back towards the host consists of returning over its own trajectory, which is stored in the vehicle's memory, to ensure an obstacle-free path but optimizing the run. The optimization consists in eliminating the loops that the vehicle has done during its random navigation.

The control unit in each vehicle 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 vehicles had to be based on a hardware flexible enough to allow the experimentation of different navigation and control strategies. These requirements have resulted in the design shown in figure 3 which contains three different functional modules.

Figure 3. Control System Architecture

The navigation module generates the trajectory to be followed and the steering module controls the motors in order to follow the generated trajectory. The perception module 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 standard PC 80C186 with 1MB RAM to store the perceived environment map. This map is discretized with a resolution of 4 cm which means that each vehicle can store maps of up to 40x40 m.

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

4. MAP GENERATION FROM ORTHOGONAL ENVIRONMENTS

The goal of map generation is to obtain the most plausible position of walls and obstacles based on the IR perception of several vehicles. The information about their position conveyed by the different vehicles is imprecise due to the odometry error. The vehicles detect portions of walls or obstacles with different degrees of precision depending on the length of the run and the number of turns that the vehicle has done. The main problem is to decide whether several detected portions, represented by imprecise segments (see 4.1), belong to the same wall or obstacle or not. This decision depends on the distance between the segments as well as on their imprecision. The distance is compared to a threshold to decide whether the segments represent different portions of the same wall or obstacle or not. 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 orthogonal environments in which the vehicles evolve. Finally, an alignment process also takes place so that two walls separated by a doorway are properly aligned.

The map generation algorithm consists of two steps. The first one is the fusion of the segments perceived by each vehicle since the same vehicle can observe more than one portion of the same wall, and the second step consists of an update of the latest fused map (resulting from the perception of other vehicles) with the new information conveyed by the last arrived vehicle. This incremental approach is necessary because not all troup members may return home to give the information to the host and because, on the other hand, we want to have at any time the most plausible map based on the information obtained so far. To summarize, any time a vehicle returns home the fusion algorithm is executed to update the map and then the completion and alignment processes take place based on the latest fused map.

Figure 4 schematically shows the map generation steps. The host gets three maps from each incoming vehicle : the path map, the perceived map and the communicated map (i.e. the union of all the maps communicated by vehicles that the incoming vehicle has encountered during its exploration). This information carried by the incoming vehicle is used to obtain a fused map. Next a second step uses the fused map of the incoming vehicle to update the current fused map. Next the completion process takes place starting with the segments, then completing corners and finally completing doorways. The last step is the alignment of those segments that being at both sides of a doorway are not co-linear. In section 4.1 we outline the segment fusion procedure (5) and in 4.2 we explain the completion and alignment processes.

Figure 4. Map generation steps

4.1 Segment fusion

An imprecise segment *s* is defined as a pair of coordinates representing its extremes and an error. Given the orthogonality of the structured environment, we have only two possible orientations for any segment : vertical or horizontal. That is *s* = ((x y_1), (x y_2), e) or *s* = ((x_1 y), (x_2 y), e,). Where e is the error due to odometry. Furthermore, for each segment we keep the list of coordinates of the singular points (i.e. corners and gaps in the real world) that have been either detected by the vehicles or generated by the completion process.

For any imprecise segment *s* we define a possibility distribution μ_s such that for any cartesian coordinates (x, y) gives the degree of possibility of the coordinates as being part of the portion of wall represented by the segment. That is $\mu_s : R \times R \rightarrow [0, 1]$. This function can be seen as a possibility distribution (6) in the sense of fuzzy logic (7) and is determined by the two extreme coordinates and the error of the segment. For example, figure 5 shows an example of the possibility distribution associated to an imprecise horizontal segment *s* = ((x_1 y), (x_2 y), e) and means that for any coordinates ((a b) (c b)), such that y - e < b < y + e and x_1+e > a > x_1 - e and x_2-e < c < x_2 + e, the possibility distribution gives the degree of possibility of the cordinates as being part of the portion of a wall represented by the segment.

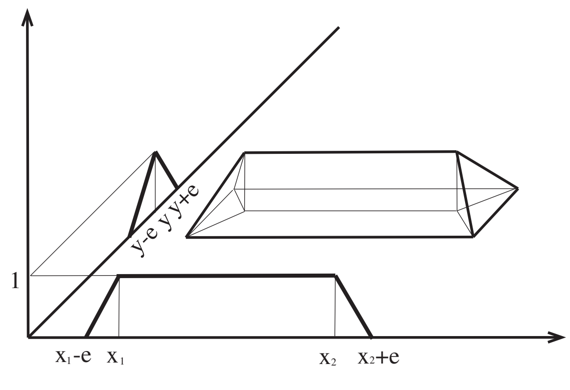

Figure 5. Horizontal Imprecise segment.

As we have said, the main problem is to decide whether several imprecise segments represent portions of the same wall or obstacle or not. This relative position is represented by two parameters Δx and Δy that represent the minimum distance of the segments coordinates with respect to both axis.

Given a map M, represented as a list of segments ordered by imprecision, the fusion process 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 (i.e. with the smallest error e), we compute its distance to the rest of horizontal (resp. vertical) segments in M with respect to both axis. Then, for those segments in M such that the 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 M in the appropriate order acording 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 the 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.

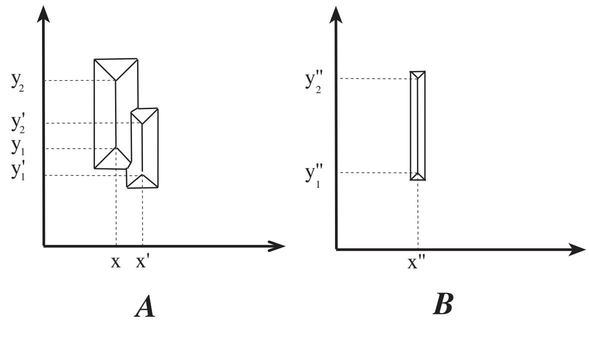

Figure 6. Vertical segments in A merged into the segment in B.

Figure 6 shows an example of vertical segment 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 center of gravity of the masses 1/e and 1/e' of *s* and s', being e and e' the error values of *s* and s'. The error e"of the merged segment is computed as a function of e and e'. One possible function is simply the sum of the masses i.e. 1/e" = 1/e + 1/e', however alternative functions (8), like for example 1/e" = max (1/e , 1/e'), are also possible. These functions can be given as parameters to the algorithm.

To clarify further the above example, figure 7 graphically shows the x projection of the merging result (B) for two vertical segments (A)

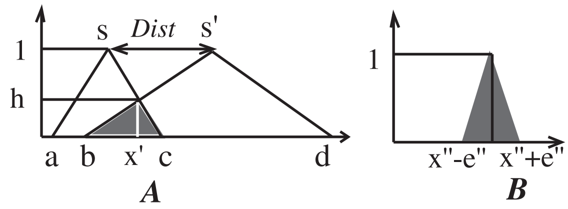

Figure 7. Merging of two vertical segment projections along the x axis.

4.2 Map completion

The fusion procedure is an initial step that only merges segments in a local basis using their geometrical properties. To improve the fused map, we perform a completion process based on: 1) a global view of the environment, including the map that each vehicle has from the other vehicles that it has encountered during its exploration; 2) knowledge about the characteristics of the environment; and 3) the paths followed by the vehicles. We adopt an heuristic knowledge-based approach by means of a set of 15 heuristic 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 and T shape)
- completing doorways

and are considered in this order.

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

If segments s_1 and s_2 are parallel **and**
there are no singular points in the
closest extremes of s_1 and s_2 **and**
$\Delta x\ (s_1, s_2) < k\alpha$ **and**
$\Delta y\ (s_1, s_2) < k\beta$ **and**
there is no path through the gap
between s_1 and s_2
Then Do (Vert_Merging (s_1, s_2))

Where α is the typical width of a doorway and β the typical width of a corridor and the function Vert_Merging is a function used by the fusion process (5).

Another example of rule for completing "L-shaped" corners is:

If segments s_1 and s_2 are
perpendicular **and**
there are no singular points in the
closest extremes of s_1 and s_2 **and**
$\Delta x\ (s_1, s_2) < k\alpha$ **and**
$\Delta y\ (s_1, s_2) < k\beta$ **and**
there are no paths through the gap
between s_1 and s_2 **and**
s_1 and s_2 do not meet at the
corner point (x,y)
Then make s_1 and s_2 to meet at (x,y) **and**
mark (x,y) as singular point

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

The very last step is to align segments that should have either the same x coordinate (vertical segments) or the same y coordinate (horizontal segments) because they are at both sides of a doorway. To do that we simply use the Vert_Merging (or Hor_Merging) functions of the Fusion process in order to compute the same final x (or y) position of the two alignated segments. The Merging functions are suitable because they weight the imprecision of the two segments in such a way that the final position is closer to the more precise of the two segments.

5. RESULTS

Several vehicle prototypes 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

Figure 8 shows different coverings of the environment obtained from a right-turning-tendency vehicle with three different behaviours : (a) anxious, (b) normal and (c) routine) and for the same path length. We can observe that the percentage of covered environment is very high with a relatively short running time. With the three maps of figure 8, the host has run the fusion algorithm described in section 4. The most plausible map obtained is shown in figure 9. Figure 10 shows the result of the alignment process based on the heuristic that most orthogonal environments have the walls aligned.

Figure 8 (a). Trajectory and map obtained with anxious behaviour

Figure 8(c). Trajectory and map obtained with routine behaviour

Figure 9. Fused map

Figure 8 (b). Trajectory and map obtained with normalbehaviour

Figure 10: Aligned map

6. CONCLUSIONS

These relatively small autonomous vehicles, in spite of being of very low cost, are provided with some of the sensing and control elements that usually have the standard autonomous vehicles developped in many research centers [9] [10] [11]. Since these autonomous vehicles are equiped with a quite high precise odometric system, it is possible to carry some not uniquely reactive navigation strategies. To be able to perform exploratory missions in unknown environments without having available an absolute positioning system, some constraints had to be assumed. In our case we have imposed the restriction of considering environments with orthogonal boundaries. When the autonomous vehicles detect one of these boundaries they follow it and, taken its direction as a reference, the vehicles can reset the angular accumulated error. This allowed us to obtain maps with a quite good precision. Furthermore, given the capability of the vehicles to exchange the information about the environment when they pass close enough to each other, the information that the returning vehicles can supply to the host is very important. Finally, the use of fuzzy logic based techniques by the fusion algorithm has allowed to obtain the most plausible map of the environments.

7. ACKNOWLEDGMENTS

We acknowledge the contributions of Carles Sierra in the discussions concerning the fuzzy logic approach to map generation as well as the contributions of Gil Arbós, Maite López, Albert Sànchez and Angel Toribio who have worked hard in the implementation and experimentation of the vehicles prototype.

8. REFERENCES

[1] R. Alami, R. Chatila, B Espiau, "Designing an Intelligent Control Architecture for Autonomous Robots", Int. conference on Avanced Robotics (ICAR'93), Tokyo, Japan 1993

[2] R. A. Brooks. "A Robust Layered Control System for a Mobile Robot" *IEEE Journal of Robotics and Automation*, RA-2, pp. 14-23, 1986

[3] R. A. Brooks. "Intelligence Without Reason" Proceedings of Int. Joint Conference on Artificial Intelligence (IJCAI'91), pp. 569-595, 1991

[4] A. Martinoli, F. Mondada "Collective and Cooperative Group Behaviours: Biologically Inspired Experiements in Robotics", Fourth International Symposium on Experiemental Robotics (ISER'95) . Stanford, California, June 30 - July 2, pp. 2-7, 1995

[5] J. Amat, R. López de Mántaras, C. Sierra "Cooperative Autonomous Low-cost Robots for exploring Enviroments", Fourth International Symposium on Experiemental Robotics (ISER'95) . Stanford, California, June 30 - July 2, pp. 28-33, , 1995

[6] R. López de Mántaras; Aproximate Reasoning Models. Ellis Horwood Series in Artificial Intelligence, GB. 1990

[7] L.A. Zadeh "Fuzzy sets as basis for a theory of possibility" *Fuzzy Sets and Systems* ,1, pp. 3-28, 1978

[8] D. Dubois, J-L. Koning; "Social choice axioms for fuzzy set aggregation", *Fuzzy Sets and Systems*, 43, , 257-274, 1991

[9] R. A. Brooks, L. A. Stein "Building brains for Bodies" Memo 1439, MIT, AI Lab., Cambridge, Massachussets 1993

[10] B. R. Donald, J. Jennings, D. Rus "Information invariants for cooperating Autonomous Mobile Robots", Robotics Research the Sixth International Symposium, pp. 29-48, Pittsburgh, USA 1993

[11] B. Jouvencel, J.E. Symphor; "The variable Modelling of Mobile Robot Environments", IEEE Int. Workshop on Intelligent Robots and Systems (IROS'91), Osaka, Japan 1991.