Integrated Computer-Aided Engineering 11 (2004) 239–258
IOS Press


# A fuzzy perceptual model for map building and navigation of mobile robots[1]

Eugenio Aguirre[a,*], José F. Argudo[b], Antonio González[a] and Raúl Pérez[a]
[a]*Department of Computer Science and Artificial Intelligence, E.T.S. de Ingenier ía Informática, University of Granada, 18071 Granada, Spain*
*Fax: +34 958 243317; E-mail: {eaguirre, A.Gonzalez, fgr}@decsai.ugr.es*
[b]*Software Engineering Department, Escuela Superior de Ingenier ía, University of Cádiz, C/Chile s/n, 11003 Cádiz, Spain*
*E-mail: josefidel.argudo@uca.es*

**Abstract**. In the area of the intelligent mobile robots the hybrid reactive-deliberative architectures for navigation are aimed at an efficient integration of reactive and deliberative skills. Reactive skills allow the robot a fast reaction to unexpected events whereas deliberative skills permit the generation of plans to carry out tasks. In these systems a high level of accuracy in the modeling of the environment is not usually necessary but some representation structure is needed to generate safe paths for the robot navigation. Thus, the perception of the environment and the construction of useful models are two main problems to deal with. In this work, we propose a fuzzy perceptual model and a map building process which would allow us to build a world topological map giving us the possibility for reasoning and planning about the robot motion in the world. The perceptual model deals with the uncertainty and vagueness underlying the sensor data, it carries out the data fusion from different sensors and it allows us to establish various levels of interpretation in the sensor data. The topological map is used to generate high-level abstraction paths and then the navigation is carried out using our own hybrid architecture and taking into account the perceptual model to represent the robot's beliefs about the world. Experiments in simulation and in real office-like environments are shown for validating the proposal.

## 1. Introduction

The hybrid deliberative-reactive architectures applied to mobile robot navigation are aimed at an efficient integration of deliberative [26] and reactive skills [11]. Typically these architectures use different levels of abstraction setting a hierarchical structure. Thus, in three layer architectures [5,20,35] the control is usually situated at the lowest level whereas the deliberation is situated at the highest level. The intermediate level is the responsible for executing the actions of the plan that has been previously computed by the planning level and it has also to take into account the current robot state and the perceived world. To achieve the deliberative skills, some world modeling is needed to build some kind of representation upon which a planner can work and search the best path that links the initial and final positions of the robot. An environment map is the representation model that is typically used for the navigation of mobile robots. This map can be directly provided to the robot so that it can use it during the navigation or the map can be constructed by the robot from the sensor data, increasing its autonomy in this way. In this case, before beginning its mission, the robot must explore the work space to acquire the necessary information with the objective of building and storing a representation of this space. Therefore, it will be necessary to have a perception model that allows us to process the sensor data in a suitable way and that can represent and manipulate the vagueness and uncertainty in these data in an effective way.

[1]This work has been supported by the MCYT under Project TIC2002-04146-C05-02.
[*]Corresponding author.

ISSN 1069-2509/04/$17.00 

In the field of mobile robots, two fundamental paradigms exist to represent the environment the robot has to cope with, geometric and topological approaches, although hybrid models have also been developed. Within the geometric approaches, we can find occupancy grid maps [17,31,32], that consist of a matrix of cells, each of which contains a measurement of the certainty with which the modeled space is occupied by an object. In other cases, geometric elements such as points, lines or polygons are used to represent the environment. These models include, among others, generalized cones [10], Voronoi diagrams [7], segment maps [13,18,40] and convex polygons maps [12]. On the other hand, the topological maps represent the robot environment as a graph in which the nodes correspond to distinguished situations or places and the arcs connect those nodes between which a direct path exists [16, 25,29]. Within the topological approaches, the semantics associated with the nodes and arcs is not always the same. For example, in [41] the topological map is obtained by partitioning a probabilistic occupancy grid into regions, that are considered as nodes, separated by passages, that are considered as arcs. However, in other models [25] the nodes represent places characterized by the sensor data and the arcs represent paths between places and they are associated with control strategies. Therefore, the main difference between the geometric and topological approaches lies in the stored information. In the geometric approaches, some geometric properties of the objects such as shape, size and position are represented in relation to a global coordinate system. On the contrary, the topological maps represent the world with a higher abstraction level, since they simply register the vicinity or connectivity relationships between the relevant places or objects. Thus, the construction of geometric maps requires to measure distances and angles, whereas this is not necessary to construct topological maps. Finally, hybrid models have arisen to integrate characteristics of both approaches. Some generate topological maps from grid maps [41], whereas in other cases a hierarchy of geometric maps linked by topological connections is defined [24,25]. In [4] a hybrid model is proposed to adapt the type of representation to the specific needs for navigation according to the characteristics of the environment.

In certain environments it may be necessary for a robot to know its exact location in terms of metric coordinates and then the best choice is to use a geometric map. In other environments, such as a highway network, the street grid of a city, or an office floor with rooms and corridors, a map specifying only the topology of the important places and the connections among them could be enough. In this work, the objective is that the robot be able to navigate through an indoor environment taking into account references as the walls, corridors, doors and other characteristic elements of this kind of environment, but knowing only the connections among these objects and not their exact locations. Thus, in this work we use a topological map since it can meet the needs of world representation allowing us, in addition, a more efficient planning than using a geometric map.

In this paper, we will focus only on the aspects relative to the perception of the environment and to the construction of its topological map. We propose a *fuzzy perceptual model* and a *map building process* that allow us to build a world topological map giving us, at the same time, various levels of interpretation of the sensor data and the possibility for reasoning and planning about the robot motion in the world. Both the perceptual model and the map building process are integrated into a hybrid deliberative-reactive architecture for behavior-based navigation.

There exist two approaches to build topological maps: one consists of creating the topological map directly [16,25,29,43], while in the other, the robot explores the environment and builds a geometric map, from which a topological model is extracted later by means of some process of analysis [41]. The method proposed in this work directly creates a topological map of an indoor environment, made up of several rooms that are linked by means of doors and corridors.

With regard to the perceptual model, firstly, to reduce the negative influence of the noise in the sensor data we define a new *fuzzy sensor model* that gives us a belief degree about the possible existence of a piece of wall that is being sensed perpendicularly in the direction to the sensor. Then fusing information gathered from different sensors, the contours around the robot can be determined and classified in *perceptual objects* like *walls*, *corners*, *corridors*, *doors*, *hallways* and other distinguished places using an incremental process. Fuzzy logic [44] is the tool used to manage the uncertainty and vagueness of the sensor data and to model the different perceptual objects that will be detailed along this work. The uses of fuzzy logic in robotic systems to connect perception to action have been numerous [9,21,27,30,37,38] and several autonomous robots have been equipped with a wide set of fuzzy behaviors as the cases of autonomous robots FLAKEY [36], in which the architecture Saphira has been developed, and MARGE [22]. Regarding environment modeling using fuzzy logic some approaches deal with the modeling of the uncertain geometric robot environment [23] using points and lines whereas others built sonar map using segments whose width and length are trapezoidal fuzzy sets [18]. There are other approaches that have proposed variants of Moravec and Elfe's occupancy grids [31] in which each cell is associated with a possibility distribution that expresses if a cell is free or occupied [32]. In our proposal and taking into account the perceptual objects defined, a topological map of the environment can be built through the exploration of the world. Then a planner will be able to find the best path that links the initial position of the robot to the final localization to reach.

This paper has been organized in the following way. First, we briefly introduce the hybrid deliberative-reactive architecture and the set of behaviors used for navigating. Next we show the kind of map that will be built through the exploration of the world. This map is based on the perceptual model which is explained in detail in Section 4. In this section firstly, the fuzzy sensor model for modeling the possible existence of perpendicular walls is properly detailed. Upon this fuzzy sensor model we develop the procedures to determine the perceptual objects or distinguished places setting the description of these places by means of fuzzy logic. Once the perceptual model is explained, the environment exploration algorithm and map building process are defined in Section 5. In the experimental section, some results of detection of perceptual objects and the map building process both in simulation and in a real office-like environment are shown. Then the perceptual model and the topological map proposed have been applied for navigating using our own deliberative-reactive architecture obtaining a safe behavior-based navigation, first in simulation and afterwards in a real environment. Finally, we comment on about related works, some limitations of the proposal and some conclusions and future work lines too.

## 2. Overview of the architecture

Our system is along the lines of behavior-based robots [6] and uses a three-level architecture to integrate deliberative techniques and reactive behaviors. This architecture is composed of three hierarchical layers: a planning, an executive-exploration and a control level. Figure 1 shows the relationship among these layers and the map of the environment.

The highest level must search for a safe and minimum-cost path from an initial position to a final desired position across an office-like environment, which is expressed by means of a map that contains topological information about the environment represented like a graph. Previously, the map is constructed performing an exploratory task in order to discover several kinds of distinctive places, which are regions in the world that have characteristics that distinguish them from their surroundings. In the next section, it is described how the world is represented by means of a graph and in Section 5 the map building process is explained in detail.

In the exploration phase, the intermediate level will deal with building the topological map of the robot work space. To do that, the robot must travel through the environment locating the perceptual objects that have to be stored into the map, by selecting the behaviors which should be activated at each moment, according to the context of the robot and the exploration strategy adopted.

The topological map provides information gathered directly from the interaction of the robot with the world, but it can additionally contain any information obtained independently of the robotic agent itself, such as maps obtained from floor-plans. This kind of information resolves, for example, the problem of detecting the presence of a staircase since this kind of object cannot be detected with the current sensor system. The topological map can be used by the planner to compute a minimum-cost path from the current position to the desired goal taking into account the estimated length of each arc and using a standard graph search algorithm, such as Dijkstra's shortest path algorithm or the A* algorithm.

The executive function of the second level must ensure the fulfillment of the plan by selecting which basic behaviors should be activated at a given time depending on the environment, the robot state and the current goal. Thus, this level defines the context of applicability of certain behaviors using a set of metarules that address the perceptual objects that form the current context. The detection of the perceptual objects is based on the perceptual model proposed in this work which is used to generate the level of belief in the existence of the different objects.

The robot, by activating its reactive behaviors, will be usually able to accomplish the plan in spite of the presence of unexpected obstacles along its path. Additionally, the executive level monitors the performance of the robot so that possible failures can be detected.

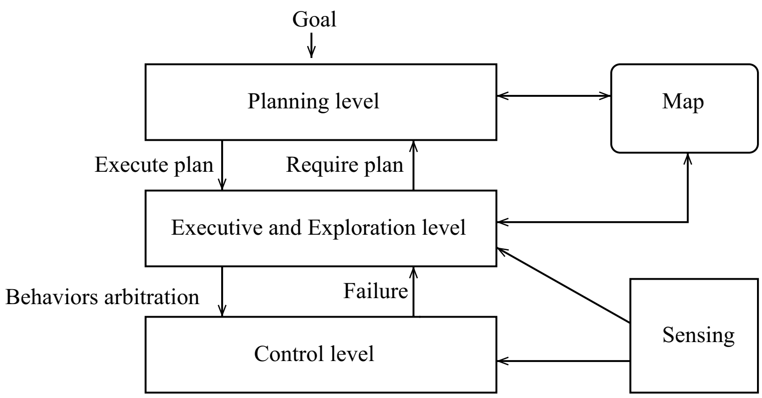



Fig. 1. Three-level architecture and map.

These failures may be given by changes in the environment which are not present in the current map. For example, a door that previously was opened, is closed now or some unexpected obstacle is blocking the path in the middle of a room or the robot becomes trapped among several unexpected obstacles. In these cases the execution of the original plan is interrupted by the execution level and, if it is necessary, one behavior specifically designed for these situations is activated to lead the robot to a safe area. The new location of the robot is computed by the execution level and the map is updated to show the new state of the environment. Then the planning level considers the new situation to decide if it is possible to fit the current plan, or on the contrary, it is needed to generate a new plan or to abort the mission after informing the user.

The robot localization during navigation is also a function of the intermediate level. This problem is resolved using an approach based on map matching [8, 14,34] so that the robot can know its position in the world but taking into account a reasonable level of uncertainty.

The lowest level deals with the control of the robot motion, coupling sensors to actuators. The control level is composed of several rule-based basic behaviors which can be combined to generate a more complex observable behavior [2]. Fuzzy logic is also used for designing the rules of the behaviors and to obtain the preferred action from each behavior and then to fuse these actions.

### 2.1. *The behaviors*

The design of the behaviors follows a methodology [3] by the authors, which is based on fuzzy control and fundamentals of regulatory control. This methodology sets a classification of the behaviors according to the use of several abstraction levels on the information. That is, we classify the behaviors according to the kind of input information that is used. Thus, the input data can be:

- Input data from the robot sensors with a simple pre-processing to avoid noisy data. Within this kind of behaviors we distinguish between:
  1. Behaviors addressed to reach and maintain an objective e.g. the following of a wall. We call these *Objective-oriented* behaviors and they are:
     * *Follow wall*. It follows the right or left wall to a certain distance and maintains the robot aligned with the wall.
     * *Follow Corridor*. It keeps the robot close to the middle of a corridor and in line with it.
     * *Face object*. This behavior moves the robot according to a certain orientation so that the robot aligns itself with the corresponding object, for example a wall, corridor or door.
  2. Behaviors that tightly couple perception to action such as *Avoid obstacle*. These are *Purely-reactive* behaviors.
- Input data from a sensor-derived world modeling. There is a temporary world representation but only the information necessary for the performance of a specific behavior is represented. For example *Cross door* is a behavior of this kind. We call these *Short-memory* behaviors.

More details about the behaviors, metarules and control level can be seen in [3] whereas details about the planning level and topological and geometrical maps integration can be viewed in [4]. Likewise, overall architecture is profusely detailed in [1]. In the rest of the work the attention is focused on the fuzzy perceptual model and on the topological map building process.

## 3. Environment modeling: The topological map

Considering that the navigation of the robot is lead by reactive behaviors, which permanently gather local information from the environment and act according to these inputs following certain rules, it is not necessary to use maps with complete information on the world. The idea is that the robot moves through the environment following routes in the style of the descriptions that the people use to indicate how to arrive at a place: *"go out through this door and follow the corridor towards the left until arriving at the next door"*. That is, the routes that the robot must follow are described in terms of the subobjectives or places that have to be reached by means of the interaction of different behaviors, instead of providing the robot a path as a series of coordinates in a plane.

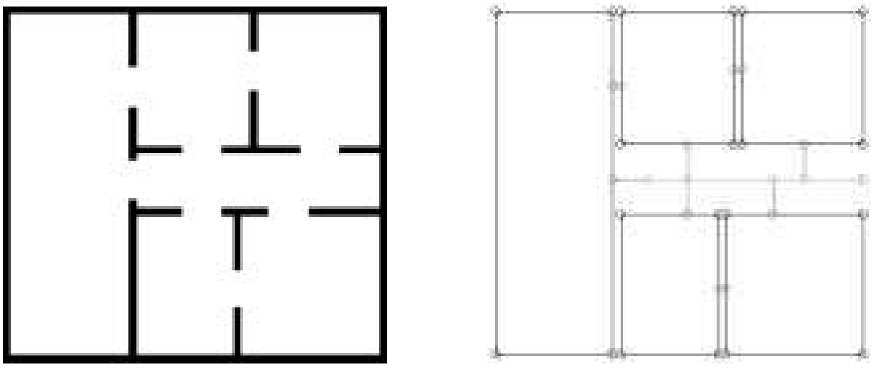
When representing the world in this way instead of using a geometric model, the navigation capabilities of the robot are better used. For example navigating following the contour of a wall, by means of the behavior *Follow wall* defined in the control level, generates a very robust final result because the robot moves with constant reference to an object that it is perceiving instead of doing it with respect to a representation of the object that hardly will be exact. In short, the objective-oriented behaviors and those of purely-reactive type are going to generate a final behavior more robust and less dependent on the accuracy of the representation models used, since they have a higher interaction with the environment.

Therefore, the model that we use to represent the work environment of the robot consists of a topological map represented as an undirected connected graph (Fig. 2). Formally, the topological map consists of a graph $G = (V, E)$, where $V = \{v_1, \ldots, v_N\}$ is the set of N nodes, and $E \subseteq \{e_{ij}, i = 1 \ldots N, j = 1 \ldots N\}$, where $e_{ij} = (v_i, v_j)$, is the set of M edges. The nodes of this graph correspond to distinguished places of the environment and the edges connect pairs of these places. The perception system classifies the distinguished places according to its morphological characteristics in: corners (c), doors (d), hallways (h), end of corridor (ec) and a default object type corresponding to a long irregular boundary (i). Then each edge of the graph represents a wall (w), a corridor (co), an edge that crosses a door (cd), or a link between an irregular type node and any other kind of node, and it expresses a transition between two distinguished places, that is, the behavior that the robot must activate to be transferred from one place to another. Both nodes and edges have a descriptor which contains information about the object type and a fuzzy estimation of the object's length.

Some explanations about the way of representing these distinguished elements in the graph follow:

– A door always gives rise to two nodes in the graph, each of them represents the door from a side of this one. The kind of these nodes can be door or hallway, depending on whether the door connects two rooms or a room with a corridor. In the first case, both nodes will be of the door type, one in each room, whereas if there is a door between a room and a corridor, then a node of door type will be had in the room and a hallway type node in the corridor. The reason for which this distinction is done is that the doors can be faced in the corridors and then the robot will detect such doors at the same time, when it moves throughout the corridor, thus they are represented as a single node of hallway type connected with different nodes of door type situated in the rooms to which they give way.
– The hallways are also going to represent the ending of a corridor in an open zone (which can also be considered a room from the point of view of the environment representation) without any door that mediates between both spaces.
– As a result of both previous points, a hallway is a place of transition between a corridor and one or several rooms, and it can represent one or several doors seen from the corridor or the ending of the corridor directly in a room too.
– The end-of-corridor node type is reserved to represent the ending of a corridor in a wall.

It is important to note that the number of adjacent edges of a node is determined by the kind of node. Concretely, a corner has two edges, the two walls that form it; a door in a room will always have two edges corresponding to the two walls that delimit it and an edge of connection with the other side; a hallway will have two, three or four adjacent edges, according to whether it connects with one or more rooms; and finally, the end of a corridor is a node with a single edge, the corridor that finishes. This characteristic of the representation model facilitates the exploration task largely, because when the robot arrives at a node it needs to know the



(a) Work environment (b) Topological map

Fig. 2. Work environment representation by means of a topological map.

number and type of the adjacent edges to learn which are unexplored and to decide which way the exploration should continue using the appropriate behavior.

In order to complete the topological map, it will be necessary to incorporate some information associated to the edges to be used as a help for the search of paths in the planning phase. That is, it will be necessary to assign some cost or weight to each edge of the graph like the length of the edge or the security level or difficulty whereupon the edge can be crossed, depending on the obstacle density that has been detected in the phase of exploration.

## 4. The fuzzy perceptual model

The proposed perceptual model allows us to build an approximate environment model by setting various perceptual objects or distinguished places that represent different levels of interpretation of the sensor data. Besides, this fuzzy perceptual model is used to determine the robot's beliefs about the perceptual objects present in the environment and it allows the behavior-based navigation using the deliberative-reactive architecture briefly introduced in Section 2. To define the model, first the influence of noise in the sensor data must be taken into account, so we have developed a process to reduce the negative influence of the noise. The idea is to define a sensor model for computing a level of belief about the possible existence of a straight contour around the robot and situated perpendicularly in the direction of the sensor which is sensing that contour. These contours belong to objects and walls of the environment but from the point of view of the robot the contours will be considered coming only from walls so that the concept of wall must be understood in a flexible way. Precedents of this way of understanding the walls can be found in [28] where the integration of qualitative maps and behavior-based robotic systems is demonstrated. Once we have a belief degree of the existence of a straight contour that is being perceived perpendicularly to some sensor, the different perceptual objects that form the topological map are defined.

### 4.1. The fuzzy sensor model

The sensing system of the robot is under the influence of multiple error sources that depend on the environment features and the kind of sensor. These errors concern the computation of distance and position of the sensed objects by the perceptual system. In our system the main sensors used to navigate are sonar and infrared sensors, but actually most measures are provided by the sonar sensors since infrared sensors have a very limited range. To consider the presence of uncertainty and vagueness in the sensing information, mainly provided by the sonar system, we have followed the process explained below. This process has been validated by experimenting with a mobile robot Nomad 200 [39], but notice that the underlying ideas can easily apply to any robot that uses ultrasound sensors.

The idea is to compute the belief in the correctness of the measure that has been sensed by a sensor. It will be higher if the echo of the emitted signal is returning from a surface perpendicular to the direction of that sensor. In this case it is well know that the computed measure is very accurate. Thus, a new fuzzy sensor model is developed to consider the influence of the uncertainty and vagueness in the measures. The idea of this sensor model is to increase our belief in a measure if the echo is returned from a surface perpendicular to the direction of the sensor. This sensor model makes sense because our

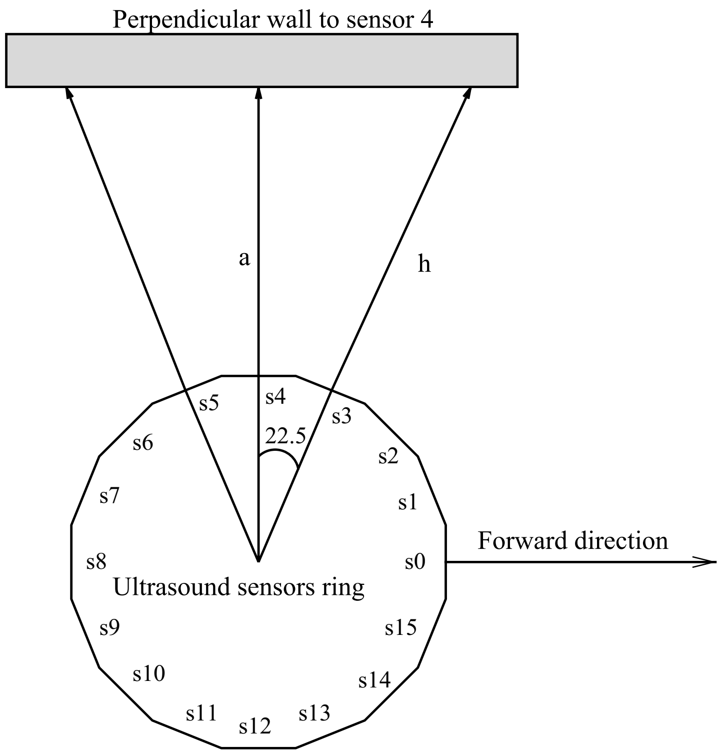



Fig. 3. Sensing a perpendicular wall to sensor 4.

architecture can use several behaviors that allow to the robot to follow the contours of objects, in fact *Follow wall* and *Follow corridor* behaviors move the robot following the right or left wall or following a corridor in the middle. In such situations, the sensor situated in the number 4 position of the sensor ring (see Fig. 3), if the robot is following the wall on the left, or in the number 12 position (see Fig. 3), if it is following the right wall, are going to be in very good situation for sensing the distance to the object with a high accuracy. However the echoes are not always returned from perpendicular surfaces so a measure of belief is needed to assess the accuracy of the values computed by any sensor. Thus, we propose the following system.

The first step is to model the situation of sensing a perpendicular echo and afterwards consider the influence of the error source. Figure 3 depicts the ideal situation when the sensor $s_4$ is sensing a perpendicular wall. In this ideal situation, we are supposing the absence of noise in the measures and that the echoes have a good angular resolution. Both features will be more realistic in the case of the echo from sensor $s_4$ but not for sensors $s_3$ and $s_5$. This fact will be dealt in a second phase.

In the ideal situation shown in Fig. 3, we define the relationship $d$, between the value of $h - a$ and the value of $a$, as $d = \frac{h-a}{a}$, or simplifying $d = \frac{h}{a} - 1$. Considering the situation shown in Fig. 3, a rectangle triangle exists among $a$, $h$ and the wall, so that $a = h \cdot \cos 22.5$ and replacing the value of $a$ in the first equation then:

$$d = \frac{h}{h \cdot \cos 22.5} - 1; \quad d = \frac{1}{\cos 22.5} - 1;$$
$$d = 0.0823.$$

Therefore the value of $h - a$ can be said to be $8.23\%$ the value of $a$ when a perpendicular wall to sensor $s_4$ is sensed. So far we have supposed the absence of noise and that the signal echoes return to the sensor describing straight trajectories. This assumption can be considered generally valid for sensor $s_4$, since this sensor is pointing in a perpendicular way to the sensed wall, but for sensors $s_3$ and $s_5$ a more complex model is needed. The reason is that the incidence angle of the ultrasound beam on the object affects to the accurate of the measure given by the sensor. Also, the ultrasound sensors can sense any object within a cone approximately 30 degrees wide and they can supply erroneous measures depending on the position the sensed object into the perception cone. Therefore, a 30 degree cone is used to model the width of the range of the sonar,

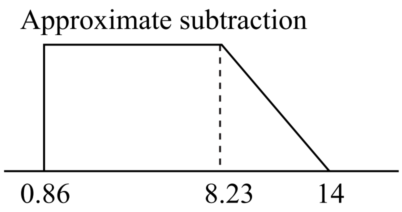



Fig. 4. Approximate subtraction between $h$ and $a$ normalized to the value of $a$ (in percentage).

so that the returned echo from the sensors $s_3$ and $s_5$ can come from some points of the wall different to the ones shown in Fig. 3 (corresponding to the arrowheads) and these points should probably be closer the point that has been sensed by the sensor $s_4$. In order to take this aspect into account, our sensor model considers that the echo can come from a point such that the angle between lines $h$ and $a$ will be a value belonging to the interval $[7.5, 22.5]$ (degrees), since the influence of the half sensing cone is considered (15 degrees). The value of $d$ if the angle is 22.5 degrees, has been already computed and we call it $d_{\max}$, in percentage, $d_{\max} = 8.23$. Through a similar process for an angle of 7.5 degrees, we obtain in percentage, $d_{\min} = 0.86$. To deal with the vagueness underlying to the assumed suppositions, including the fact that the sonar measures have an error of 1%, the fuzzy set shown in Fig. 4 is used as a soft representation for the value of $d$. The support of this fuzzy set is defined so that both $d_{\min}$ and $d_{\max}$ will have maximum membership degrees and the values greater than $d_{\max}$ will have gradually lower membership values. The gradient of the right part of this fuzzy set has been experimentally fixed.

This fuzzy set is considered as the *Approximate Subtraction* (*AS*) between the measures of two consecutive sensors $s_i, s_j$ normalized to the value of $s_i$, needed to establish whether in front of $s_i$, a perpendicular wall has been detected by the echo of $s_i$. Thus, the degree of belief in sensing a perpendicular wall to $s_i$ taking into account $s_j$, $B(s_i/s_j)$, is defined as the value of the membership function of $AS$, $\mu_{AS}$, in $(s_j - s_i)/s_i$ expressed in percentage, that is:

$$B(s_i/s_j) = \mu_{AS}((s_j - s_i) \cdot 100/s_i)$$

so that $B(s_i/s_j)$ will be a value that belongs to interval [0,1] since $\mu_{AS}$ is a membership function of a fuzzy set and it is defined in the interval [0,1]. Additionally, the previous sensor in the ring can also be considered in order to carry out a similar process. Thus, let $B(s_i/s_{i+1})$ be the degree of belief of perpendicular wall to sensor $s_i$ taking into account the value of $s_{i+1}$ and let $B(s_i/s_{i-1})$ be the belief of perpendicular wall to sensor $s_i$ but considering the measure from sensor $s_{i-1}$, then the final value $B(s_i)$ will be the result of fusing both values. To compute the final value, we consider that a perpendicular wall to $s_i$ exists if a perpendicular wall to $s_i$ has been detected depending on the measures of $s_{i+1}$ or $s_{i-1}$. Understanding $B(s_i/s_{i+1})$ as the possibility of the first event and $B(s_i/s_{i-1})$ the possibility of the second event, then the possibility of the union of both events is the maximum of both possibilities according to the possibility theory [45]. Thus the blended belief will be:

$$B(s_i) = \max\{B(s_i/s_{i+1}), B(s_i/s_{i-1})\}.$$

Anyway it is necessary to notice that this process is not avoiding all the possible noise sources, since the ultrasound sensors can be affected by several noise sources at the same time and it is usual the presence of non linearity in the data. That is, there may be some situations in which the result of the described operations renders the belief of a perpendicular wall to be 1, but the wall is not actually perpendicular. Therefore to draw more reliable conclusions from this sensor model some redundancy and data fusion are needed. Thus, in the process to determine the perceptual objects several values of this measure along the time must be considered and also the information of different sensors of the sensor ring must be properly blended. Both questions have been taken into account in next subsections in which, both the measures of the sensor model and the distances computed by the ultrasound sensors are understood as variables affected by vagueness and uncertainty.

Vagueness refers to the fact that the value of the variable under consideration is only known to belong to some subset of values that is not a singleton. Uncertainty refers to the lack of complete information that precludes a statement as to the certainty that the variable either belongs or does not belong to some subset. In this work, we use the possibility theory [45] for the modeling of information that is both vague and uncertain so that the perceptual information is understood as belief and it is dealt by the robot following the rules of the approximate reasoning [44]. Furthermore, with respect to the design of the behaviors, we use fuzzy control [15] since this kind of control is preferred for non linear systems, systems with no predictable disturbance or low accurate sensors and systems where there is a need to incorporate human experience [42].

### *4.2. Determining the perceptual objects*

In this subsection, we are going to show how the sensor model previously defined can be used to determine various perceptual objects or distinguished places that can be used for both building a topological environment map and navigating using the behaviors.

The philosophy that guides the design of these routines is the following. In the first place, a linguistic description of the object is given and then this description is represented using expressions with fuzzy sets. These fuzzy expressions use certain perceptual features that generate a belief level in the existence of the object. Then if a high level of belief is given during a time then the beginning of the object is considered. The perceptual routines are continuously computing the belief level thus if the level becomes lower than a certain threshold during a time, then the end of the object is fixed.

In next subsection this process is explained in detail for the case of determining the wall existence but when the other objects are explained, the attention will be mainly focused in the fuzzy expressions to determine the perceptual features.

### *4.3. Determining the existence of possible walls*

To determine the existence of a wall, the information provided by the previously explained sensor model is used. Through the sensor model a belief value of sensing a perpendicular wall can be associated to each sensor of the robot sensor ring. Summarizing, let $s_0$ to $s_{15}$ be the sixteen possible positions in the sensor ring then $B(s_i)$ is the belief value, between 0 and 1, of a perpendicular wall exists to the corresponding sensor at distance $D(s_i)$. This information is always available and it is continuously updated. Furthermore, taking into account the direction of the robot motion when it is navigating different kinds of walls can be determined. That is, the process will deal with the possible existence of walls on the right or on the left relative to the robot motion. Therefore our objective is to compute the value of possible left wall, which is expressed by $\text{belief}W(Left)$ or possible right wall, which is expressed by $\text{belief}W(Right)$. We are also interested in obtaining the distance to the wall. On the other hand, as the values of long distances can be more affected by noise, a distance threshold is considered so that a measure is rejected if it exceeds this threshold. Actually this threshold is a new fuzzy set that we call $WD$ (*Wall Distance*). This fuzzy set, shown in Fig. 5, allow us to

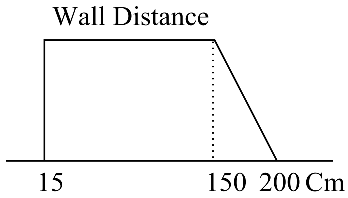



Fig. 5. Wall Distance.

smooth the threshold of maximum distance at which a wall can be sensed.

The linguistic description of a right or left wall could be as follows. A left (or right) wall exists if the sensors situated on the left (or right) are detecting an approximately perpendicular wall to the direction of those sensors and also the distance of detected wall is smaller than a certain threshold. This description is translated to the fuzzy expression beliefW(Left) which fuses the information provided by various sensors.

Here we suppose that the robot is following the left wall so that the value of $\text{belief}W(Left)$ is computed by:

$$\text{belief}W(\text{Left}) = \max_i\{\min\{B(s_i),$$
$$\mu_{WD}(D(s_i))\}\} \quad i = 2, 3, 4.$$

That is, the belief of the possible left wall is computed using the value of possible perpendicular wall detected in some of the positions sensed by the sensors number 2, 3 or 4, taking the maximum value of $\text{belief}W$ among the three sensors and provided that the distance to the possible wall was under the threshold. Let $s_{\text{iMax}}$ be the sensor in which the belief is maximum and being the distance to the wall within the established threshold on the distance, then the value of distance to the left wall is given by:

$$\text{distance}W(\text{Left}) = D(s_{i_{\text{Max}}}).$$

Another factor that must be considered to establish the beginning of a wall is the time factor, that is, to establish that a left wall is sensed, the value of $\text{beliefW(Left)}$ must be greater or equal than a certain threshold continuously during some time.

Let

$$\{\text{belief}W_1(\text{Left}), \text{belief } W_2(\text{Left}), \ldots,$$
$$\text{belief}W_n(\text{Left})\}$$

be $n$ consecutive measures of $\text{beliefW(Left)}$ and let $\delta \in [0, 1]$ be the considered threshold, then the beginning of the wall becomes true if:

$$\forall_i\{\text{belief}W_i(\text{Left}) \geqslant \delta\} \quad i = 1 \ldots n.$$

Thus we use redundant information to avoid the negative influence of noise sources. Finally, we should note that the values of parameters $n$ and $\delta$ have been determined experimentally after numerous tests have been carried out.

When the beginning of the wall is detected, the level of belief in that wall is continuously observed and updated so that if this value becomes lower than the threshold for a number of consecutive steps then that wall is considered finished. Other possibility that can provoke the end of the detected wall is the perception of a new perceptual object in the data, as for example the presence of a corner or a corridor.

Belief levels for walls on the right, front and back can be defined in a similar fashion to $\mathrm{belief}W(Left)$. These are their expressions:

$$\mathrm{belief}W(\mathrm{Right}) = \max_i\{\min\{B(s_i),$$
$$\mu_{WD}(D(s_i))\}\} \quad i = 12, 13, 14;$$
$$\mathrm{belief}W(\mathrm{Front}) = \max_i\{\min\{B(s_i),$$
$$\mu_{WD}(D(s_i))\}\} \quad i = 1, 0, 15;$$
$$\mathrm{belief}W(\mathrm{Back}) = \max_i\{\min\{B(s_i),$$
$$\mu_{WD}(D(s_i))\}\} \quad i = 7, 8, 9.$$

Once again, the distance to each wall is computed following the process explained above.

### 4.4. Determining the existence of possible corners

A corner is another possible distinguished place to be considered. More exactly we consider a corner as the intersection between two approximately perpendicular walls. Thus, a corner can be perceptually established by the determination of two nearby walls that are intersecting and forming it. We use the information $B(s_i)$ about the existence of "possible perpendicular wall" from all of the sixteen sensors of the ring. To detect the existence of a corner at the direction of the sensor $s_i$ the levels of belief in perpendicular wall of sensors $s_{(i+2)\mathrm{mod}16}$ and $s_{(i+14)\mathrm{mod}16}$ are used because these two walls are forming the corner in the direction of $s_i$. Additionally, the distance of these walls is bounded using a new fuzzy set in order to detect the corner in the moment in which the robot is reaching that corner. This new fuzzy set, that we call *Corner Distance* (CD), is shown in Fig. 6.

Thus, the belief of the existence of a corner in the direction of $s_i$ is computed as follows with $0 \leqslant i \leqslant 15$:

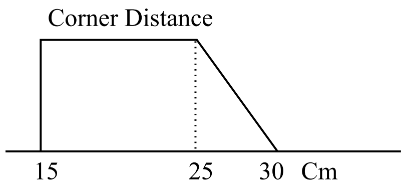



Fig. 6. Corner Distance.

$$\mathrm{belief}C(s_i) = \min_j\{\min\{B(s_j), \mu_{CD}(D(s_j))\}\}$$

where $j = (i+2) \bmod 16, (i+14) \bmod 16$ and $\bmod$ is the mathematical module operator. Therefore, a value of belief in the interval [0,1] is assigned to the possibility of existence of a corner in the direction of sensor $s_i$. The distance to this corner will be approximated using the minimum value between the distances to each wall.

Once a value of "possible corner" has been assigned to every position of the sensor ring, then these values are blended for computing the belief of "possible left or right corner" according to the direction of the robot motion. Using the maximum as an aggregation operator, the belief of left corner will be:

$$\mathrm{belief}C(\mathrm{Left}) = \max_i\{\mathrm{belief}C(s_i)\}$$
$$i = 1, 2, 3, 4.$$

Similarly for a right corner, it will be:

$$\mathrm{belief}C(\mathrm{Right}) = \max_i\{\mathrm{belief}C(s_i)\}$$
$$i = 12, 13, 14, 15.$$

The process of corner detection needs some redundancy in the measure of $\mathrm{belief}C(s_i)$ so that in order to establish the presence of the corner, a high level of $\mathrm{belief}C(s_i)$ is needed during some time using a process similar to the procedure described in subsection 4.3. The end of the corner is determined when the measure of $\mathrm{belief}C(s_i)$ is under a certain limit during some time. The values of the parameters used in this process, as well as the support and membership function of the fuzzy sets have been obtained experimentally. Convex corners, are not actually considered into the kind of the corners discussed here. They are considered as result of the end of a wall or a corridor. When a topological concept has to be associated to these corners they can be modeled as *hallways* and *irregular contours* which are described in subsection 4.7.

### 4.5. Determining the existence of possible corridors

We use a linguistic description of the concept of corridor to define a corridor. We call $\mathrm{belief}Co(Ahead)$ to

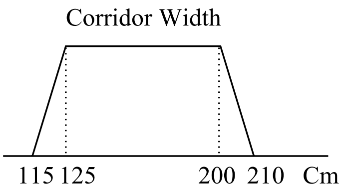



Fig. 7. Corridor Width.

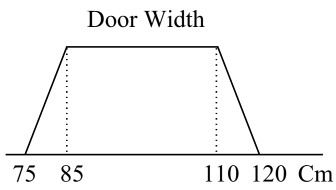



Fig. 8. Door Width (*DW*).

the level of belief on a corridor composed of "two parallel walls that are detected at both sides and separated to certain distance". Anyway the process deals with many more intermediate possibilities relative to the position of the robot and the walls since fuzzy sets are used to represent and process the sensor information. The information of the existence of the walls is taken from $\mathrm{belief}W(Left)$ and $\mathrm{belief}W(Right)$ whereas the width of the corridor $WideCo$ will be established by the following expression:

$$\mathrm{WideCo}\ (\mathrm{Ahead}) = \mathrm{distance}W(\mathrm{Left}) + \mathrm{distance}W(\mathrm{Right}) + 2 \cdot R$$

where $R$ is the radius of the robot turret.

This width must be within certain limits to be considered as a valid width of a corridor. These limits are represented by means of a new fuzzy set that we call *Corridor Width* (CW), which is shown in Fig. 7.

The level of belief of "possible ahead corridor" is defined blending the belief of wall on the left, wall on the right and the belief of these walls being separated within the considered limits. That is:

$$\mathrm{beliefCo(Ahead)} = \min\{\mathrm{belief}W(\mathrm{Left}), \mathrm{belief}W(\mathrm{Right}), \mu_{CW}(\mathrm{wideCo(Ahead)})\}.$$

The minimum is the operator used since the three conditions are necessary to determine the existence of the corridor. The detection of the end of the corridor is determined when the robot detects a frontal wall that interrupts the corridor or when it arrives at a hallway situated at the end of the corridor.

### 4.6. Determining the existence of possible doors

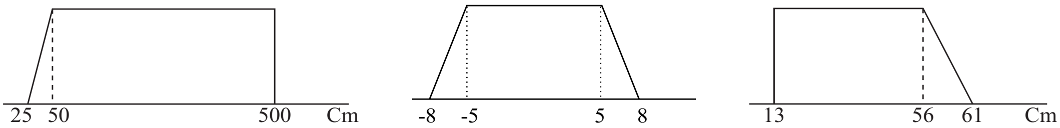
The detection of the doors is more difficult than the detection of the previous objects. There is an important reason for this. The robot try to detect doors when it is following the contour of walls. In this situation, it is not possible to determine the existence of a door from the measures taken only at a fixed moment. It is necessary to consider the measures of the sensors during a time interval because when the robot begins to perceive a possible door, the presence of the other frame that delimits the door cannot be detected with reliability. Obviously, we are speaking of opened doors, because the robot is going to try to detect the gap delimited by the frames of the door. Consequently, the linguistic description of a door is a gap within a wall and whose width is similar to the widths of doors commonly found in indoor environments. This width is defined as the fuzzy set shown in Fig. 8, called *Door Width*.

The detection of doors is made in two stages. Firstly, the position of the first frame that the robot finds when it is following a wall is calculated. That is, the existence of a possible door is considered when the robot begins not to perceive the followed wall. The end of a left wall is determined using the sensors $s_3$, $s_4$ and $s_5$ in the following way:

- Three fuzzy sets are defined (Fig. 9):
  * High Increment (*HI*) define a high increment between two consecutive readings of a sensor.
  * Zero Increment (*ZI*) define a very small variation between two consecutive readings of a sensor.
  * Wall Normal Distance (*WND*) represent the approximate distance to which the robot follows the walls.
- The membership degree of the variation between the two last readings of $s_3$, $\Delta s_3$, to the fuzzy set *HI* is $\mu_{HI}(\Delta s_3)$.
- The stability degree of the readings of the sensor $s_i$ when the robot is following a wall is defined as the measurement in which the variation between the two last readings of $s_i$ is very small and the distance to which $s_i$ is perceiving something is more or less that which the robot follows the walls.

$$stability(s_i) = \min\{\mu_{ZI}(\Delta s_i), \mu_{WND}(s_i)\}.$$

- Finally, it could be concluded that the left wall has ended and the robot has detected the beginning of a possible door, if $s_3$ has a sudden high increase, that is, it does not already perceive the wall, but $s_4$ continue perceiving the wall, that is, $s_4$ is sta-




(a) High Increment (*HI*) (b) Zero Increment (*ZI*) (c) Wall Normal Distance (*WND*)

Fig. 9. Fuzzy sets for detecting a door.

ble. However, in order to increase the confidence in this measurement, the degree of stability of the sensor $s_5$ is also calculated and the degree corresponding to the most stable sensor ($s_4$ or $s_5$) is taken. Thus, the degree of belief of end of a left wall or belief in the existence of the first frame on the left, $\text{belief}FF(Left)$, is calculated as the minimum of these two conditions.

$$\text{belief}FF(\text{Left}) = \min \left\{ \begin{matrix} \mu_{HI}(\Delta s_3), \\ \max\{\text{stability}(s_4), \\ \text{stability}(s_5)\} \end{matrix} \right\}$$

In a second phase, the size of the gap that constitutes the possible door is calculated, that is, the robot goes to try to detect the beginning of the second frame. To do that, the robot must continue advancing following the direction of the wall until finding the other frame. This one will be found when the robot detects again the wall to the same distance to which it left it ($\text{distanceW}_0(\text{Left})$) and whenever the traveled distance between a frame and another one ($\text{traveled\_distance}$) corresponds to Door Width (*DW*). Thus the degree of belief in the existence of the second frame on the left, $\text{beliefSF}(\text{Left})$, is:

$$\text{belief } SF(\text{left}) = \min \left\{ \begin{matrix} \text{beliefW(Left)}, \\ \mu_{ZERO}(\text{distance}W_0 \\ (\text{Left}) - \text{distance} \\ \text{W(Left))}, \\ \mu_{DW}(\text{traveled\_} \\ \text{distance}) \end{matrix} \right\}$$

where, $\text{belief}W(Left)$ is the belief in finding again the wall after the gap, $distanceW(Left)$ is the distance to this wall and $ZERO$ is a fuzzy set that represents "approximately 0".

Therefore, the final degree of belief that a door has been found, $\text{belief}D(Left)$, is defined as the minimum of the two conditions because both previous conditions are necessary:

$$\text{belief } D(\text{Left}) = \min\{\text{belief } FF(Left), \text{belief } SF(\text{Left})\}$$

The belief of a door on the right can be defined in a similar fashion considering sensors $s_{11}$, $s_{12}$ and $s_{13}$.

Notice that this process to detect the doors suffers from the usual weakness of the ultrasound sensors, therefore some redundancy on the measures is taken into account again. Anyway, some gaps in the walls can be classified as doors whereas they are not real doors. In a new version of this process, we are using visual information to increase the belief in the possible existence of the frames of a door so that this problem can be resolved.

### *4.7. Determining the existence of other distinguished places*

It is possible to find places with contours not sufficiently regular to be considered walls and that can not be classified in any of the previous categories. In this case, a special kind of object that we call *irregular contour* is used for modeling these places. For example, a convex corner at the end of a wall will be classified as this kind of object.

Regarding the places *hallways*, the robot detects them using a similar process for detecting a door when this one is located at a left or right side of the direction of robot motion. That is, if the robot is following a corridor and it arrives near to an opened door which links the corridor and a room, then it thinks that the corridor is interrupted by the presence of the opened door and classifies that place as a hallway, since it is a place that communicates the corridor with a room. Moreover if the corridor ends on a clear space then the robot will detect the open space in both sides and it will classify this place as a hallway too.

Figure 10 shows both situations describing the concept of hallways for the robot. Besides, this figure shows some convex corners and how these places can be included into the hallway places.

## 5. Environment exploration and map building

In this stage, the final mission of the robot is to explore the environment in order to determine all the nodes and edges of the graph and to build the topolog-

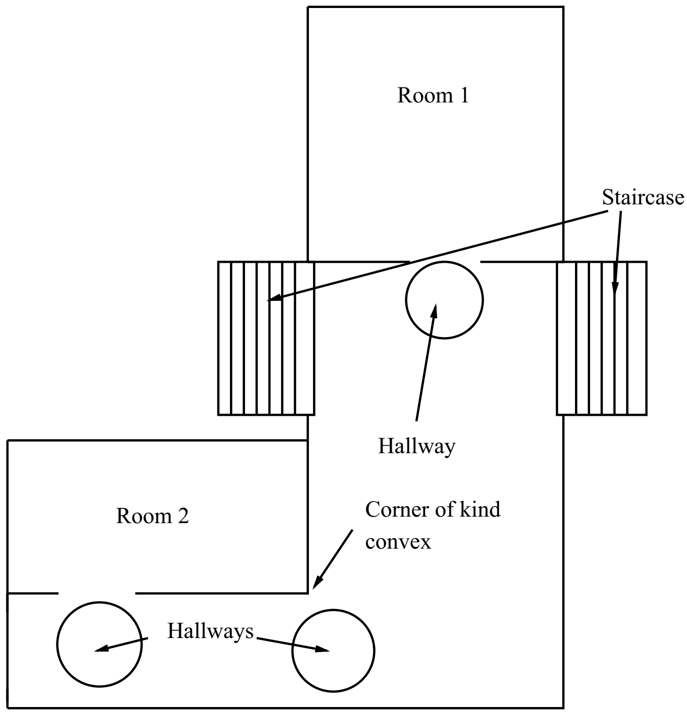



Fig. 10. Hallways.

ical map of the environment. For this, as it has been explained in Section 2, the executive-exploration level, that is located on the control level in the navigation system architecture, takes charge of selecting the behaviors that must be activated at every time. Which are these behaviors will depend on the context in which the robot is at every time and on the exploration strategy followed.

As the exploration progresses, a partial map of the environment is constructed consisting of all nodes and edges that have already been explored. Whenever the robot arrives at a node, it is necessary to solve two problems so that the robot can achieve its objective. On the one hand, it must select the edge by which to continue the exploration and to activate the behavior to traverse it. To select the most suitable edge to explore next, the robot needs to know the number and the type of the edges adjacent to the reached node. In our model of environment representation this is possible, because the kind of node directly determines the number and type of the adjacent edges. Therefore, for each node of the constructed part of the map, at any time it is known how many unexplored adjacent edges exist and their types, but the ends of these edges are unknown.

Although the map that we want to build is a topological map, sometimes the robot uses metric information in the map building process and therefore a method is necessary to correct the cumulative errors of the dead reckoning system. This problem is resolved by matching environment maps constructed by the robot at different times. The initial map built when the robot starts moving in the environment is used as a reference map since it provides the best estimate of the positions of the different perceptual objects detected in the environment. These positions are relative to a local coordinate system which is associated to that concrete sector of the environment. Subsequent maps obtained as the robot continues moving in the same sector are matched against the initial map and the transformation that brings them together is used to estimate the odometric system errors. Once these errors are corrected the uncertainty about the robot localization is bounded to certain level that depends on the size of the sector and the precision of the odometric system. However this uncertainty can be managed by the system since the position of the robot is represented by using fuzzy sets too. Actually a more accurate positioning system is not necessary because the global topological map has not to be metric and globally consistent to allow the behavioral navigation of the robot. This localization method is mainly based on the map matching method proposed in [19].

With regard to the exploration of the environment, the exploration algorithm has been designed based on the proposal of Panaite and Pelc [33] to explore an unknown undirected and connected graph. During the construction of the map, the edges that have not been explored yet are called *free edges*. A node is said *saturated* if it does not have free adjacent edges. The exploration of the complete graph is done visiting and saturating each node of the graph. Each node saturated during the robot's trip is marked, thus the graph will have been completely explored when all the nodes have been marked.

In order to saturate a node $v$, the robot follows the algorithm shown in Fig. 11. It traverses a path from $v$ successively selecting free edges until arriving at a saturated node or until returning to departure node $v$. In the first case, the robot turns back on the walked path until finding a nonsaturated node and it continues the exploration from there. But if the robot returns to the initial node $v$, then $v$ can be nonsaturated and the robot will continue the exploration as has been described, selecting successive free edges and going back from the saturated nodes until returning to $v$ again. This process will finish when the robot returns to node $v$ and it does not have free adjacent edges.

Once the initial node is saturated, if there are still nodes with free edges, a partial map of the environment will have been constructed. Then one of these nodes is selected and it is saturated by means of the previous procedure. The selection of the next node to saturate is made with the goal of reducing the total distance traveled by the robot, so the nearest nonsaturated node to the current node is chosen. When no unsaturated nodes remain, the exploration finishes and the map will have been completely constructed.

Likewise, the order in which the free edges of a node are selected is important to try to reduce the distance traveled by the robot during all the process. Thus, the robot does not leave a room until it has been completely explored. A corridor is not explored at once, but rather when the robot advances by a corridor, it goes into the rooms to explore them, crossing the doors that it finds throughout the corridor. According to this, the robot explores the environment by regions, room by room, so that when it visits the last room, probably nonsaturated nodes does not remain and therefore it will not have to return on its steps to complete the map.

## 6. Experimental results

As we said, the objects or distinguished places that have been explained can be used for both building a qualitative map of the environment and navigating using the set of suitable behaviors. Thus, in our architecture one important advantage of using a topological map is the assistance to the navigation since the considered distinguished places show features that can be used by the robot to navigate. For example, if a wall is detected then the robot can activate the *Follow wall* behavior, if a corridor is sensed then the suitable behavior will be *Follow corridor* and so on. This way of generating behaviors is quite robust because the robot can navigate according to the objects that it is sensing. Moreover, the qualitative descriptions of the environment are more suitable to adapt the behavior of the robot to the features of the world. For example, when a door is detected then a specific behavior called *Cross door* will be used. Therefore, as a consequence of the environment description using these distinguished places, a high level plan can be generated and then this plan give us a sequence of places to visit so that the robot can reach the final goal.

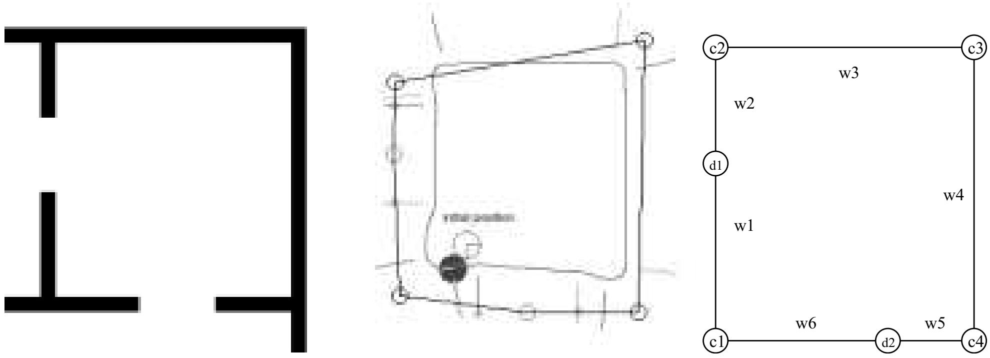
With regard to the concrete experiments, we first show an example of sensor data classification in perceptual objects while the robot is exploring an environment and building the topological map. The example is shown first in simulation and later in a real room. Next, we show a navigation task and its resolution using the topological map that has been built using the perceptual objects and the map building process previously explained. First, the navigation task is executed in a simulated environment and then another navigation task is shown but in the real world using a Nomad 200 mobile robot in an office-like environment. Both navigation tasks are accomplished using our hybrid deliberative-reactive architecture and the proposed perceptual model. It allows the robot to model the environment and navigate according to the context perceived.

### 6.1. *Perceptual objects detection and map building process*

In this section a couple of experiments are described in which the robot detects the perceptual objects and builds the topological map of a room. Figure 12 shows the exploration of a room in a simulated environment and the map built by the robot. In Fig. 12(b) we can see the robot's trajectory and the detected objects along its path. The robot starts in an initial position near the left lower corner, approaching the closest object, which it identifies as a wall. So, the behavior *Follow wall* is activated and the robot turns on the right and advances

```
Saturate(v):
    u = v
    Repeat while (u ≠ v or v nonsaturated)
        If u nonsaturated, then
            To traverse e, being e a free adjacent edge of u
            u = the end of e
        else
            Let [u_0, u_1, ... u_k] be the traversed path
                from v to u without the possible cycles
            To go back u_{k-1}
            u = u_{k-1}
```

Fig. 11. Saturation algorithm of a node $\nu$.




(a) Work environment (b) Exploration (c) Topological map

Fig. 12. Exploration and map building of a room with two doors.

parallel to the wall situated on the left. While the robot moves under the control of this behavior, it continuously analyzes the sensor data applying the procedures of detection of distinguished places, so as soon as it begins to detect one of them, it creates a new node in the graph along with the edge connecting the new node with the previous node and it continues the exploration. In Fig. 12(b) the start and the end of each distinguished place are marked with two perpendicular lines to the robot's trajectory. The approximated position of the detected object is marked with a circle. The start of an object and the circle indicating its approximated position have been drawn up by the robot in run time at the moment at which it begins perceiving the object. The end of each object has also been drawn up in run time when the robot finishes perceiving the object. The lines (edges) forming the contour of the figure have been drawn up linking the circles (nodes) and, because the nodes location is approximated, these lines do not represent the exact positions of the walls. This fact is more evident in the upper wall, since the localization error of the corner $c2$ is high and hence the edge $w3$ is far from the real position of the wall. These errors in the object locations are due to the signal rebounds on the corners and to the odometric errors. They do not affect the map building since we are actually interested in knowing what distinguished places exist and which are the connections between them.

The robot crosses the contour of the room under the control of the behavior *Follow wall*, except when it arrives at the gap of a door. Then, the robot loses the reference of the wall, which it identifies as the frame of a possible door. Then, a specific behavior called *Follow direction* is activated, which makes the robot to continue advancing, trying to follow the same direction that it had when it was following the wall, until it finds another wall that it identifies as the second frame of the door. The doors' frames are marked with perpendicular lines to the trajectory and the door is depicted with a circle on the middle point between both frames, linked by lines with previous and next nodes (see Fig. 12(b)). The map stored by the robot is shown in Fig. 12(c), which does not represent the exact position of the perceived objects but just the topological structure of the environment.

The second trial (Fig. 13) has been done in an environment of the real world, which is an office room with tables, chairs and other furniture. In this case, we can verify that the performance is equally robust and that the robot is able to determine the topological structure

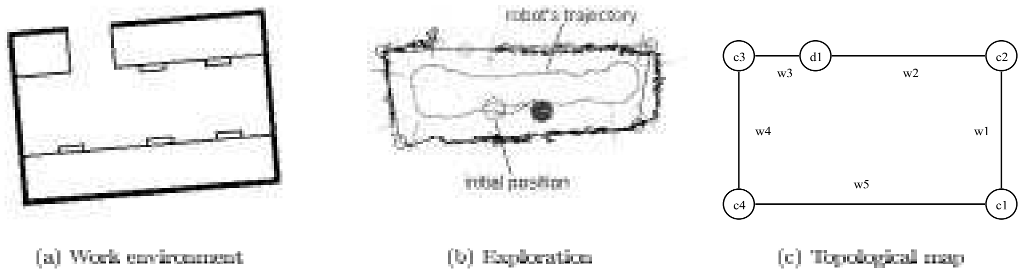



Fig. 13. Exploration and map building in the real world.

of the environment. The robot starts the exploration near the lower table shown in Fig. 13(a). The linguistic description of wall that is used causes that the table and the chairs near to the robot be interpreted like a wall. In this experiment, the perception of the robot is shown with points forming the objects contour that the robot interprets like a wall (Fig. 13(b)). Therefore, the robot advances under the control of the behavior *Follow wall* following the contour of the objects and detecting the corners until it reaches the gap between the two tables of the upper side of the figure. Due to the door concept that is used, this gap is interpreted like a door that gives way to another room or region of the environment.

### 6.2. *A navigation task that uses topological information*

This experimental example shows a navigation task in a simulated environment. Both the simulated environment and the result of navigation task are shown in Fig. 14. Before the navigation, a world topological map like the one displayed in Fig. 14 has been built using the perceptual model and map building process proposed in this paper.

The navigation task consists of reaching the corner labeled as $c15$ from an initial position at corner $c5$. The plan to resolve this navigation task is computed by the planning level using a minimum-cost path search algorithm such as Dijkstra's shortest path algorithm or the A* algorithm and taking into account an estimated length of each arc of the topological map. The plan that links the initial and final positions is composed of the following steps:

$$c5 \rightarrow w7 \rightarrow d3 \rightarrow w13 \rightarrow c8 \rightarrow w12 \rightarrow d5 \rightarrow$$
$$dh \rightarrow h2 \rightarrow co3 \rightarrow h3 \rightarrow hd \rightarrow d8 \rightarrow w21$$
$$\rightarrow c14 \rightarrow w22 \rightarrow c15.$$

In this kind of plan, it is very important the skill of the robot to detect the current context so that the robot can recognize the part of the plan in which it is. Thus, this plan is given to the executive level that uses the perceptual model of the system and information about its localization to set the state of the robot and determine the current perceptual context. According to this context several metarules are activated and therefore the corresponding behaviors are going to control the motion allowing robot to follow the walls, cross the doors, follow the corridor, etc. In fact, the robot achieves successively every sub-goal until it arrives to the desired final goal.

In detail, first the robot is situated near corner $c5$ and wall $w7$. While it is sensing such objects, it has to follow the wall on the right until door $d3$ begins to be detected. At this moment, an unexpected obstacle is perceived so that the *Avoid obstacle* behavior is activated and the robot follows the contour of the object until again the beginning of a new wall object is determined. This object is $w13$ which is ended by the beginning of perception of corner $c8$. After this, wall $w12$ is determined and ended by the perception of door $d5$. The robot crosses $d5$ using the *Cross door* behavior and then it detects hallway $h2$, where corridor $co3$ begins. In this moment *Follow corridor* behavior is activated and another obstacle is sensed and properly avoided. The navigation successively continues until it finally arrives at corner $c15$. Notice the robot adapts its observable behavior to the perceived context in each case which can be determined using the proposed perceptual model.

### 6.3. *A navigation task in the real world*

The results of navigation in simulated environments have been validated through trials in the real world. In this case the navigation task is accomplished in a real environment cluttered with chairs, tables and other objects.

Figure 15 depicts from left to right the real world environment, its representation in a topological map

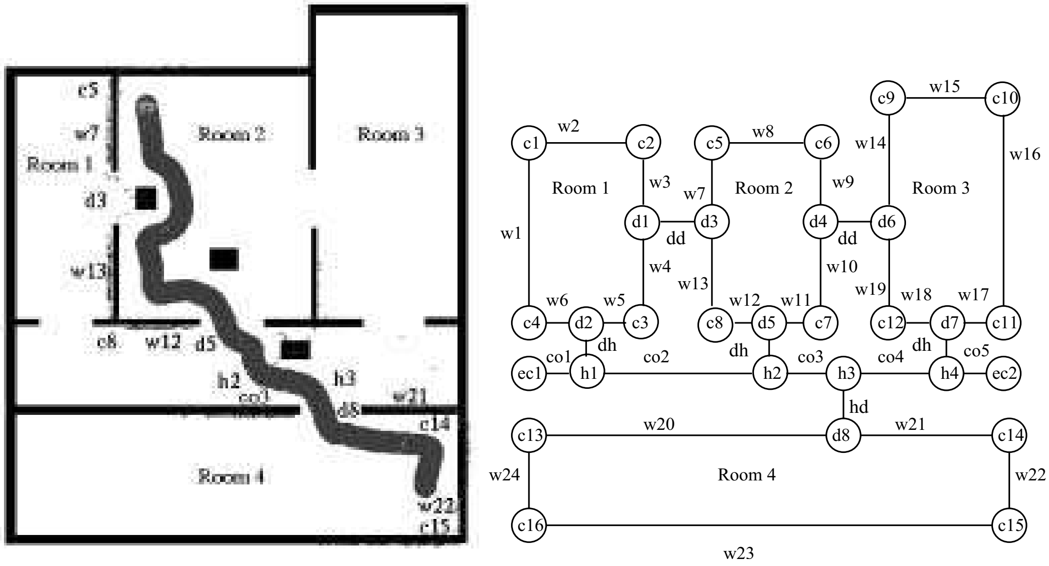



Fig. 14. From left to right: A navigation task and environment topological map.

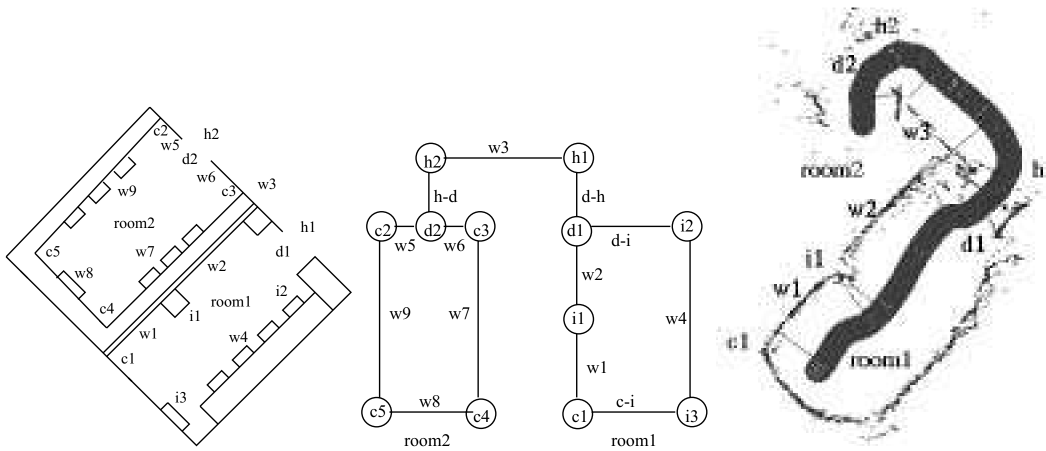



Fig. 15. A navigation task in the real world. From left to right: Environment, topological map and trace of the robot motion.

and the result of this navigation task. In this case, the task is to go into $room2$ from an initial position near corner $c_1$ of $room1$. The path computed by the planning level is formed by the next steps:

$$c1 \to w1 \to i1 \to w2 \to d1 \to dh \to h1 \to w3 \to h2 \to hd \to d2.$$

Again, the plan is supplied to executive level and this level generates the corresponding metarules to activate the behaviors according to the particular context and the defined goals. In this example, the determined perceptual objects have been marked out by flags in the trace of the navigation task shown in Fig. 15. The robot which, initially, is near to corner $c1$ detects the beginning of wall $w1$ and follows it until the belief of $w1$ decreases and the end of this wall is determined. The next object is defined as irregular contour $i1$ until a new object of type wall is detected. It is wall $w2$, the robot follows this wall until the belief on wall decreases and a new object is expected. The perceptual system determines the presence of an open door that is crossed using the *Cross door* behavior. When the robot goes out $room1$, hallway $h1$ is perceived and the robot turns on the left. After this, wall $w3$ begins to be detected and again the *Follow wall* behavior is activated until the end of the wall is determined by another open door. This open door is defining a new object hallway $h2$ that connects to door $d2$. The robot turns to face the door $d2$ then it crosses it and finally it goes into *room2*.

## 7. Related work

We found interesting advantages comparing our perceptual model and the kind of map that is built by exploration of the environment, with others approaches. Grid-based approaches [17,31,32] require a trade off between the level of detail of the grid map and the computational complexity and, the stored information can also be dramatically affected by changes in the environment. However, qualitative models are more compact, that is, the number of nodes is usually much smaller than the number of cells of grid-based maps and therefore the planning process will be faster [41]. Regarding other metric approaches such as [7,12,13], the qualitative models integrated into a deliberative-reactive architecture are more robust to changes in the environment because, generally, the behaviors are able to achieve their corresponding goals in spite of uncertainty and vagueness in the sensor data and in the representation model [28]. The proposed perceptual model is based on fuzzy logic and this allows the definition of different perceptual objects in a natural way and obtaining a great flexibility when the sensor data have to be interpreted as for instance *walls*, *corners* or *corridors* since each of these concepts includes different real situations with their owns particular features (surface, texture, smoothness).

On the other hand, the qualitative models present some weaknesses. Firstly, the perceptual objects or places are not always easy to distinguish and some times their locations can be confused or the model can classify the same distinguished place as two different perceptual objects. This problem is known as sensor aliasing and in our proposal is resolved by matching the approximate localization of the perceptual objects to be able to determine whether they are the same object. In second place, there are places that are not appropriate for a qualitative or perceptual description such as places near the stairwells and staircases. The third problem is related to the skill of the reactive behaviors that control the robot motion. In certain very cluttered environments the reactive behaviors could not achieve the current goal without additional assistance since fine motion control is needed to navigate among obstacles and to achieve the goal. These two last limitations can be overcome using some kind of geometric information of the environment to complement the topological representation. In another paper [4] we propose a hierarchical map to integrate topological and geometric modeling that can be constructed taking into account the topological map and the perceptual model proposed in this work.

In [4] the geometric information is modeled using a fuzzy segments map and a fuzzy occupancy grid-map to represent the zones that are not suitable to be modeled through the topological model as, for example, the places near the stairwells and staircases. Thanks to the hierarchical map, the robot can use the representation model more convenient in every case to achieve a safe and efficient navigation. On the other hand, the design in detail of the fuzzy behaviors used for the behavior-based navigation as well as the methods to coordinate and to fusion the different behaviors can be found in [2, 3].

## 8. Conclusions and future work

Into the area of the hybrid reactive-deliberative architectures for mobile robot navigation, some representation structure is needed to allow the robot to reason about its relationship with the environment and to plan a safe path that links its initial to desired final localizations in the environment. In this work, we have proposed a fuzzy perceptual model and a map building process that allow the construction of a world topological map giving us the possibility for reasoning and planning about the robot motion in the world. The nodes of the topological map are distinguished places like corners, doors and hallways among others, whereas the edges represent walls, corridors or other kinds of transitions between two distinguished places. An important aspect of our proposal is that the kind of edge is considered to select the corresponding behavior to control the robot so that the robot will use the more suitable behavior in every case. Moreover the way in that the topological map has been defined facilitates the exploration task because the robot knows the number and kind of the adjacent edges for exploring.

With regard to the fuzzy perceptual model, first a *fuzzy sensor model* is proposed to deal with the uncertainty and vagueness of perceptual information. Thus, the information from different sensors is blended to determine the level of belief about the possible existence of a straight contour around the robot and situated approximately perpendicular to the sensor. Upon this sensor model and following an incremental process, different perceptual objects are defined as *wall*, *corner*, *corridor* and *door*, so that various levels of interpretation in the data are managed. To determine the different perceptual objects, a linguistic description is given and then this description is represented using *fuzzy expressions* that define certain perceptual features for generating a belief level in the existence of the object. The determination of other perceptual objects as *hallways* are also commented on, but in a lesser detail level.

Both the map building process and the fuzzy perceptual model are integrated into our own hybrid deliberative-reactive architecture in order to resolve two basic requirements for intelligent mobile robot navigation: perception and reasoning. On the one hand, the uncertainty and vagueness of sensor data is properly managed by the perceptual model which is able to link sensory information to environment objects so that it is used to build the topological map of the environment. On the other hand, concerning reasoning, the topological map can be used in the deliberative layer to compute a plan that links the initial and the desired final positions. This plan is a high level abstraction plan since it only addresses the successive objects that the robot must reach while it is navigating and the motion control is under the responsibility of the control layer that can use different behaviors depending on the perceived context. The perceptual model is also used to state the robots beliefs in the moment of the execution of the plan, so that the context of applicability of the appropriate behaviors can be defined and suitably evaluated.

The perceptual model, map building and exploration processes form part of a whole architecture for mobile robot navigation. Numerous experiments, both simulated and in the real world, have been carried out to test the validity of the proposal. These results support the utility of the proposal to achieve a safe and intelligent navigation in office-like environments allowing world modeling, planning and connecting properly perception to action.

Regarding future works related to the perception it is important the integration of other kind of sensors into our perceptual model. For example, the usage of visual information could help to detect only true doors by locating the frames of the door and, during the navigation, it would also allow the robot to detect visual landmarks that cannot be perceived using the current perceptual model.

## References


[1] E. Aguirre, *A Behavior-Based Architecture for Mobile Robot Navigation*, PhD thesis, Department of Computer Science and Artificial Intelligence, Granada, Spain, 2000, in spanish.

[2] E. Aguirre, M. García-Alegre and A. González, A fuzzy safe follow wall behavior fusing simpler fuzzy behaviors, in: *Proc. of the Third IFAC Symposium on Intelligent Autonomous Vehicles*, Madrid, Spain, 1998, pp. 607–612.

[3] E. Aguirre and A. González, Fuzzy behaviors for mobile robot navigation: Design, coordination and fusion, *International Journal of Approximate Reasoning* **25** (2000), 255–289.

[4] E. Aguirre and A. González, Integrating fuzzy topological maps and fuzzy geometric maps for behavior-based robots, *International Journal of Intelligent Systems* **17**(3) (2002), 333–368.

[5] R.C. Arkin, Path planning for a vision-based autonomous robot, in: *Proc. of the SPIE Conference on Mobile Robots*, 1986, pp. 240–249.

[6] R.C. Arkin, *Behavior-Based Robotics*, The MIT Press, 1998.

[7] F. Aurenhammer, Voronoi diagrams. A survey of fundamental geometric data structure, *ACM Computing Surveys* **23**(3) (Sept. 1991), 345–405.

[8] R. Bauer and W.D. Rencken, Sonar feature based exploration, in: *Proc. of the IEEE/RSI Int. Conf. on Intelligent Robots and Systems*, 1995, pp. 148–153.

[9] A. Bonarini and F. Basso, Learning to compose fuzzy behaviors for autonomous agents, *International Journal on Approximate Reasoning* **17**(4) (1997), 409–432.

[10] R.A. Brooks, Solving the find-path problem by good representation of free space, *IEEE Transactions on Systems, Man and Cybernetics* **13**(3) (1983), 190–197.

[11] R.A. Brooks, A robust layered control system for a mobile robot, *IEEE Journal of Robotics and Automation* **RA-2** (1986), 14–23.

[12] R. Chatila and J. Laumond, Position referencing and consistent world modeling for mobile robots, in: *Proceedings of the IEEE Conference on Robotics and Automation*, 1985, pp. 138–170.

[13] J.L. Crowley, Navigation for an intelligent mobile robot, *IEEE Journal of Robotics and Automation* **RA-1**(1) (1985), 31–41.

[14] J.L. Crowley and P. Reignier, World modeling and position estimation for a mobile robot using ultrasonic ranging, in: *Proc. of the IEEE Int. Conf. on Robotics and Automation*, 1989, pp. 674–680.

[15] D. Driankov, H. Hellerdoorn and M. Reinfrank, *An Introduction to Fuzzy Control*, Kluwer Academic Publishers, 1993.

[16] G. Dudek, M. Jenkin, E. Milios and D. Wilkes, Robotic exploration as graph construction. *IEEE Transactions on Robotics and Automation* **7**(6) (1991), 859–865.

[17] A. Elfes, Using occupancy grids for mobile robot perception and navigation, *IEEE Computer Magazine, Special Issue on Autonomous Intelligent Machines* **22**(6) (June, 1989), 46–57.

[18] J. Gasós and A. Martín, A fuzzy approach to build sonar maps for mobile robots, *Computers in Industry* **32** (1996), 151–167.

[19] J. Gasós and A. Rosseti, Uncertainty representation for mobile robots: Perception, modeling and navigation in unknown environments, *Fuzzy Sets and Systems* **107** (1999), 1–24.

[20] E. Gat, *Reliable Goal-Directed Reactive Control of Autonomous Mobile Robots*, PhD thesis, Virginia Polytechnic Institute, 1991.

[21] S.G. Goodridge, M.G. Kay and R.C. Luo, Multi-layered fuzzy behavior fusion for reactive control of an autonomous mobile robot, in: *Proc. IEEE Int. Conf. on Fuzzy Systems*, Barcelona, Spain, 1997, pp. 579–584.

[22] S.G. Goodridge and R.C. Luo, Fuzzy behavior fusion for reactive control of an autonomous mobile robot: MARGE, in: *Proc. of the IEEE Int. Conf. on Robotics and Automation*, San Diego, CA, 1994, pp. 1622–1627.

[23] W.J. Kim, J.H. Ko and M.J. Chung, Uncertain robot environment modeling using fuzzy numbers, *Fuzzy Sets and Systems* **61** (1994), 53–62.
[24] B. Kuipers, The spatial semantic hierarchy, *Artificial Intelligence* **119**(1–2) (May, 2000), 191–233.
[25] B. Kuipers and Y.-T. Byun, A robot exploration and mapping strategy based on a semantic hierarchy of spatial representations, *Journal of Robotics and Autonomous Systems* **8** (1991), 47–63.
[26] J. Latombe, *Robot Motion Planning*, Kluver Academic Publishers, 1991.
[27] W. Li, Fuzzy-logic based reactive behaviour control of an autonomuous mobile system in unknown environments, *Engineering Applications of Artificial Intelligence* **7**(5) (1994), 521–531.
[28] M. Mataric, Integration of representation into goal-driven behavior-based robots, *IEEE Transactions on Robotics and Automation* **8**(3) (1992), 304–312.
[29] M.J. Mataric, *A distributed model for mobile robot environment-learning and navigation. Technical Report 1228*, MIT Artificial Intelligence Laboratory, May 1990.
[30] F. Michaud, Selecting behaviors using fuzzy logic, in: *Proc. IEEE Int. Conf. on Fuzzy Systems*, Barcelona, Spain, 1997, pp. 585–592.
[31] H.P. Moravec and A. Elfes, High resolution maps from wide angle sonar, in: *Proc. IEEE Int. Conf. on Robotics and Automation*, 1985, pp. 116–121.
[32] G. Oriolo, G. Ulivi and M. Vendittelli, Real-time map building and navigation for autonomous robots in unknown environments. *IEEE Transactions on Systems, Man and Cybernetics. Part B: Cybernetics* **28**(3) (1998), 316–333.
[33] P. Panaite and A. Pelc, Exploring unknown undirected graphs, *Journal of Algorithms* **33**(2) (1999), 281–295.
[34] W.D. Rencken, Concurrent localization and map building for mobile robots using ultrasonic sensors, in: *Proc. of the IEEE/RSI Int. Conf. on Intelligent Robotics and Systems*, 1993, pp. 2192–2197.
[35] A. Saffiotti, K. Konolige and E. Ruspini, A multivalued logic approach to integrating planning and control, *Artificial Intelligence* **76** (1995), 481–526.
[36] A. Saffiotti, E.H. Ruspini and K. Konolige, Blending reactivity and goal-directedness in a fuzzy controller, in: *Procs. of the IEEE Int. Conf. on Fuzzy Systems*, San Francisco, California, 1993, pp. 134–139.
[37] M. Sugeno and M. Nishida, Fuzzy control of model car, *Fuzzy Sets and Systems* **16** (1985), 103–113.
[38] T. Takeuchi, Y. Nagai and N. Enomoto, Fuzzy control of a mobile robot for obstacle avoidance, *Information Sciences* **43** (1988), 231–248.
[39] Nomadic Technologies, *User's Manual*, 1995.
[40] S. Thrun, To know or not to know: On the utility of models in mobile robotics, *AI Magazine* **18**(1) (1997), 47–54.
[41] S. Thrun, Learning metric-topological maps for indoor mobile robot navigation, *Artificial Intelligence* **99**(1) (1998), 21–71.
[42] R.R. Yager and D.P. Filev, *Essentials of Fuzzy Modeling and Control*, John Wiley & Sons, Inc., 1994.
[43] B. Yamauchi and R. Beer, Spatial learning for navigation in dynamic environments, *IEEE Transactions on Systems, Man, and Cybernetics –Part B: Cybernetics. Special Issue on Learning Autonomous Robots*, 1996.
[44] L.A. Zadeh, The concept of linguistic variable and its applications to approximate reasoning, *Part I Information Sciences vol. 8, pp. 199–249, Part II Information Sciences vol. 8, pp. 301–357, Part III Information Sciences vol. 9*, 1975, pp. 43–80.
[45] L.A. Zadeh, Fuzzy sets as a basis for a theory of possibility, *Fuzzy Sets and Systems* **1** (1978), 3–28.