Real-Time Map Building and Navigation for Autonomous Robots in Unknown Environments

Giuseppe Oriolo, *Member, IEEE*, Giovanni Ulivi, *Member, IEEE*, and Marilena Vendittelli

***Abstract*—An algorithmic solution method is presented for the problem of autonomous robot motion in completely unknown environments. Our approach is based on the alternate execution of two fundamental processes: *map building* and *navigation*. In the former, range measures are collected through the robot exteroceptive sensors and processed in order to build a local representation of the surrounding area. This representation is then integrated in the global map so far reconstructed by filtering out insufficient or conflicting information. In the navigation phase, an $A^\star$-based planner generates a local path from the current robot position to the goal. Such a path is safe inside the explored area and provides a direction for further exploration. The robot follows the path up to the boundary of the explored area, terminating its motion if unexpected obstacles are encountered. The most peculiar aspects of our method are the use of fuzzy logic for the efficient building and modification of the environment map, and the iterative application of $A^\star$, a complete planning algorithm which takes full advantage of local information. Experimental results for a NOMAD 200 mobile robot show the real-time performance of the proposed method, both in static and moderately dynamic environments.**

I. Introduction

IN RECENT years, an increasing amount of robotics research has focused on the problem of planning and executing motion tasks autonomously, i.e., without human guidance [1]–[3]. Such a faculty is essential for robotic systems operating in hostile environments (space, sea, contaminated habitats) as well as in the emerging field of *service robotics*, which includes waste management, cleaning, luggage transfer, disabled people assistance, and others.

To reach a reasonable degree of autonomy, two basic requirements are *sensing* and *reasoning*. The first is provided by an on-board sensory system that gathers information about the robot itself and the surrounding scene. The second is accomplished by devising algorithms that exploit this information in order to generate appropriate commands for the robot. To make the use of robots feasible in real-life applications, it is necessary to reach a tradeoff between costs and benefits. Often, this prevents the use of expensive sensors (e.g., video cameras) in favor of cheaper sensing devices, and calls for efficient algorithms that can guarantee real-time performance in the presence of insufficient or conflicting data.

In this paper, we address the problem of autonomous navigation in a completely unknown environment for a robot equipped with ultrasonic range finders. If the robot must perform several motion tasks in the same static workspace, it may be convenient to perform a preliminary, exploratory phase in order to reconstruct accurately the environment [2, ch. 5]. The high cost of such operation will be compensated by the ease of planning on a globally known map. Moreover, a global knowledge of the environment allows to plan paths that optimize a given performance criterion.

In many cases, an exploratory phase is out of question, either because the characteristics of the environment are subject to change over time, or simply because the nature of the robot task would make it inefficient. The robot must then be able to accomplish the motion in the absence of initial knowledge. Various approaches to this problem have been proposed. Despite their differences, most techniques can be classified into two major streams—*reactive* and *deliberative* navigation.

In reactive methods, there is a stimulus–response relationship between sensors and actuators, with very limited or no world modeling at all [4]–[9]. In deliberative techniques, a world model is used to formulate plans to which the robot is more or less committed; one example of this kind is given in [10]. While reactive navigation proves to be flexible by virtue of its modular design approach, it may fail when confronted with difficult tasks. On the other hand, deliberative navigation suffers from high computational requirements and performance degradation in dynamic environments. Based on the idea that “dynamically acquired world models can be used to circumvent certain pitfalls that representationless methods are subject to” [11], a number of mixed solutions have been proposed, aimed at an efficient integration of world modeling and planning into reactive architectures [12], [13].

Our solution is somewhat in the line of mixed methods. In fact, it prescribes the incremental building of a dynamic world representation and the formulation of local plans in accordance. Its originality stands mainly in two aspects. The first is the use of a particular world model that is easily aggregated and modified during the motion through a memoryless algorithm that requires no post-processing. The second is the iterative application of $A^\star$, which is a global planning method but takes full advantage of the available local information.

In selecting an appropriate world model, one must face the fact that ultrasonic sensors, although low-cost and easy to use,

Manuscript received February 4, 1995; revised June 9, 1996 and March 2, 1997. This work was supported in part by the ENEA under the Antarctica Project and by ESPRIT BR Project 6546 (PROMotion).

G. Oriolo and M. Vendittelli are with the Dipartimento di Informatica e Sistemistica, Università degli Studi di Roma “La Sapienza,” 00184 Rome, Italy (e-mail: oriolo@labrob.ing.uniroma1.it; venditt@labrob.ing.uniroma1.it).

G. Ulivi is with the Dipartimento di Informatica e Automatica, Terza Università degli Studi di Roma, 00146 Rome, Italy (e-mail: ulivi@labrob.ing.uniroma1.it).

Publisher Item Identifier S 1083-4419(98)02616-8.

may behave poorly in certain conditions [14]. Therefore, rather than trying to reconstruct a deterministic model of the environment, we have chosen to adopt an intrinsically uncertain map, defined as a *fuzzy set*: a real number is associated to each point, quantifying the possibility that it belongs to an obstacle. The resulting representation is similar to an *occupancy grid*, commonly obtained using stochastic techniques [15]–[18]. We have found that fuzzy logic provides a more robust and efficient tool for managing the uncertainty introduced by the ultrasonic sensing process. In fact, the underlying theory is developed from less constraining axioms than probability theory, so that a wider choice of operators is available for modeling uncertainty and aggregating information coming from multiple sources [19].

As the environment map is incrementally built, the local planner is repeatedly invoked in order to generate a robot path from the current position to the desired goal. Such a path must be safe inside the area so far explored, and at the same time should provide directions for further exploration aimed at reaching the goal. This is realized by defining cost functions that characterize the risk of collision along a path, and by choosing a proper instance of the $A^\star$ class of graph search algorithms in order to obtain a minimum-cost path.

Our method represents an attempt to make use of strictly algorithmic techniques in the presence of unknown and/or dynamic environments, with as little as possible sophistication in the control structure of the robot. In fact, while it is possible to devise general control architectures that behave robustly in various situations [4], in this way one might be forced to give up interesting formal properties such as completeness (i.e., the capability of finding a solution whenever one exists), that can instead be guaranteed by algorithmic approaches. Further advantages of these are the possibility of analyzing complexity as well as the efficiency of the obtained paths.

The paper is organized as follows. In Section II, we present our solution approach and outline the overall structure of the navigation method. In Section III, we review the various phases of the fuzzy map building algorithm from ultrasonic measures. In Section IV, cost functions are introduced for characterizing safe paths on fuzzy maps and it is shown that the $A^\star$ algorithm can be applied to find minimum-cost local paths. Experimental results are presented in Section V to show the satisfactory performance of our approach both in static and moderately dynamic environments. A short review of the basic concepts of fuzzy set theory is given in the Appendix.

II. The Proposed Approach

Consider a mobile robot equipped with ultrasonic range finders that must travel from its initial position to a final desired position across a completely unknown two-dimensional (2-D) environment. Throughout the paper, we assume that a localization system provides the robot with its absolute position with respect to a fixed inertial frame.

Our approach is based on the use of two basic processes, *map building* and *navigation*, that are alternately performed during the task execution (see Fig. 1). A short qualitative description of their functions is given below.

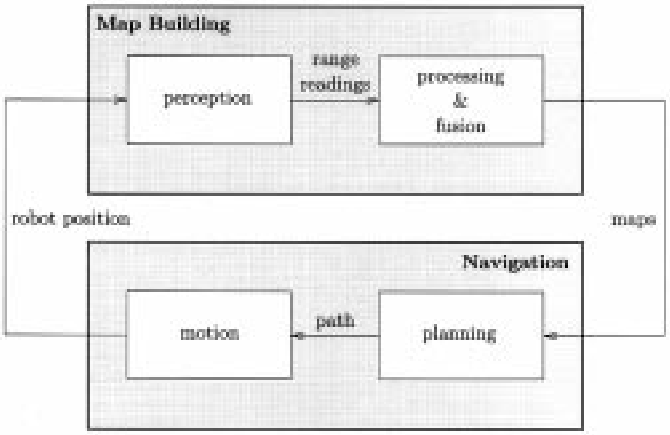


Fig. 1. The proposed solution approach.

Map Building:
The map building process is in charge of gathering through the sensors information about the environment at a given robot position and of processing it in order to update the available map in accordance. The basic steps are as follows.

Perception: The robot ultrasonic sensors are activated in a proper sequence and a packet of n range readings are collected.

Processing: Ultrasonic measures are processed in order to build a local representation of the surrounding scene in terms of empty and occupied space.

Fusion: The local representation is integrated in the global one by filtering out contradictory and insufficient information. In particular, two gray-level bitmaps $\mathcal{M}_m$ and $\mathcal{M}_p$ are computed. Unexplored areas are regarded as dangerous in $\mathcal{M}_m$ (*motion map*), while they are considered to be safe in $\mathcal{M}_p$ (*planning map*).

At the end of the last step, a new navigation process is started.

Navigation:
The navigation process generates robot motions on the basis of the information provided by the map building one. It prescribes the two following phases.

Planning: An $A^\star$-based planner computes a path from the current robot position to the goal on the planning map $\mathcal{M}_p$. This path will be safe inside the area explored so far, and will aim directly at the goal outside.

Motion: The planned path is followed as long as it is safe on the motion map $\mathcal{M}_m$, i.e., up to the boundary of the explored area. This phase is aborted if the proximity sensors detect unexpected obstacles that obstruct the motion.

If the goal has not been reached at the end of the motion phase, a new map building process is started.

The two following comments arise with reference to the approach so far outlined.

- Assume that the map building method provides an *exact* representation of the environment inside the explored area. It is easy to understand that, if the environment is static, the proposed method is *complete*, i.e., it provides a solution whenever one exists and returns failure otherwise. In fact, if a solution exists, the explored area is increased until it contains the goal. At this point, the completeness of $A^\star$ guarantees that a path inside this area (that is limited by construction) is found. Conversely, if the problem does not admit a solution, the robot will explore all the connected region that can be reached from the start position. When the map of this area is completed, the planner will return a failure. Although the underlying assumptions will not be satisfied in practice, this reasoning provides a nice theoretical support to the whole approach.

- In the above description, the robot is required to stop in order to collect ultrasonic measures during the map building process. For clarity of exposition, we shall keep this separation between the two processes. However, to reduce the time consumed by the perception phase—by far the most expensive—the latter can be performed during motion execution, provided that map building and navigation are implemented as concurrent processes.

In the following, we shall detail the above approach with reference to the NOMAD 200 mobile robot produced by Nomadic Technologies. As shown in Fig. 2, NOMAD has a cylindrical shape with an approximate radius of 0.23 m, and a kinematic model equivalent to a unicycle. The upper turret, which carries 16 Polaroid ultrasonic sensors, may be independently rotated. The robot control software runs under Unix on an IBM RISC 6000 that communicates with NOMAD through a radio link. All the algorithms for map building and navigation have been implemented in the C language.

III. The Map Building Process

The problem of building a map from ultrasonic measures is made difficult by the large amount of uncertainty introduced by the sensing process. This uncertainty consists in a *lack of evidence*: due to the inherent limitations of ultrasonic sensors, it is not always possible to decide whether a given point of the area of interest is occupied or not by an obstacle. Rather than deciding (i.e., classifying points of the space as either empty or occupied) in this unfavorable situation, a possible alternative approach is to convey all the available knowledge into an uncertain representation.

Fuzzy logic offers a natural framework in which uncertain information can be handled. Studies on the theory of fuzzy sets started in the early 1970's, with the seminal papers of Zadeh [20]. A review of the basic concepts needed for our purposes is presented in the Appendix.

Define the empty and the occupied space as two fuzzy sets[1] $\mathcal{E}$ and $\mathcal{O}$ over the universal set U (the environment), that is assumed to be a two-dimensional subset of $I\!R^2$ discretized in $\Sigma = \sigma_1 \times \sigma_2$ square cells of side δ. Their membership functions $\mu_\mathcal{E}(C)$ and $\mu_\mathcal{O}(C)$ quantify the degree of belief that the cell $C \in U$ is empty or occupied, respectively. This degree of belief should be computed on the basis of the available measures.

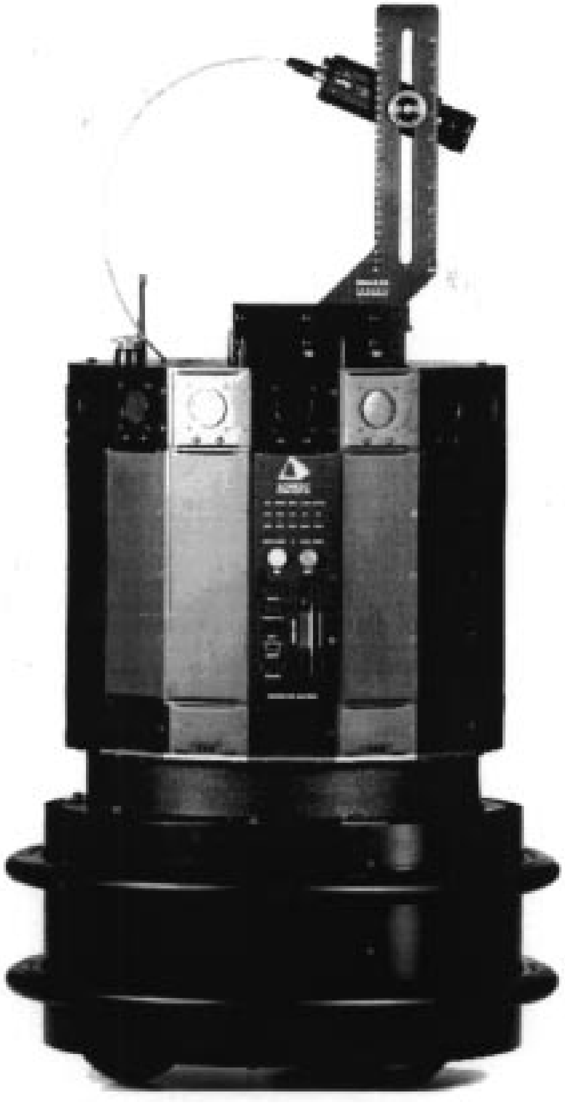


Fig. 2. The NOMAD 200 mobile robot.

In the fuzzy logic context, the two sets $\mathcal{E}$ and $\mathcal{O}$ are not complementary—the principle of *tertium non datur* does not hold. Therefore, for a given cell C, $\mu_\mathcal{E}(C)$ and $\mu_\mathcal{O}(C)$ convey independent information. This situation is particularly convenient in view of the characteristics of the ultrasonic sensing process (see Section III-A). In fact, an ultrasonic sensor detects the closest reflecting surface inside its radiation cone, thereby indicating the presence of an empty space up to a certain distance. On the other hand, no information is provided about the state of the area beyond such distance: the available evidence does not suggest emptiness or occupancy. Only by incorporating measures taken at different viewpoints it will be possible to discriminate between the two possibilities.

As mentioned in the previous section, the kth map building process consists of three phases, i.e., perception, processing and fusion (see Fig. 3). In the perception phase, a packet of ultrasonic measures $\{r_i^k\}$ $(i = 1, \ldots, n)$ are collected from the same robot position and fed to the processing phase, that is in charge of generating two local representations of the empty and the occupied space, i.e., two local fuzzy sets $\mathcal{E}^k$ and $\mathcal{O}^k$.

[1] Hereafter, we denote fuzzy sets by calligraphic capital letters.

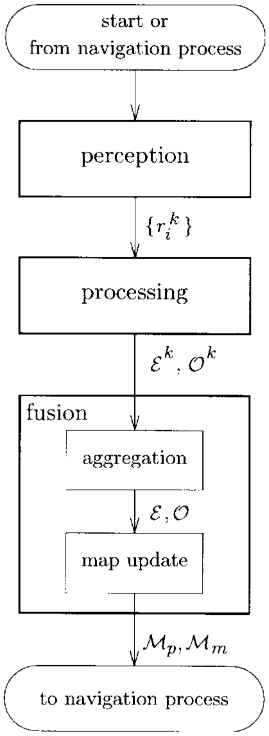


Fig. 3. The kth map building process.

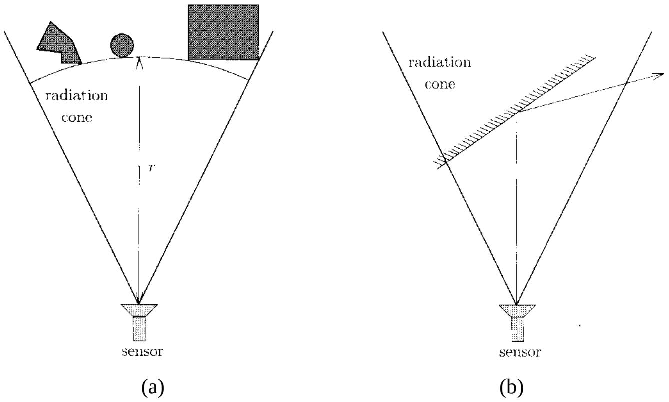


Fig. 4. Ultrasonic sensing: (a) objects in different positions can give the same distance reading r and (b) false reflections may occur for large angles of incidence.

During the fusion phase, the local information is aggregated to the global representation of the empty and the occupied space, which is contained in two global fuzzy sets $\mathcal{E}$ and $\mathcal{O}$. The final step is to elaborate the information contained in $\mathcal{E}$ and $\mathcal{O}$ so as to update two further fuzzy sets $\mathcal{M}_p$ and $\mathcal{M}_m$. Both are gray-level bitmaps conveying information about the risk of collision for each cell of the environment, but they are based on a different definition of risk. In particular, unexplored areas are dangerous in $\mathcal{M}_m$ (*motion map*), while they are safe in $\mathcal{M}_p$ (*planning map*).

Below, each phase of the map building process is described in detail.

A. Perception

Ultrasonic range finders measure the distance from obstacles in the environment by a simple conversion of the time of flight of the ultrasonic waves in air. As already mentioned, the mobile robot NOMAD 200 is equipped with a ring of 16 Polaroid ultrasonic range finders. These are constituted by a single transducer acting both as a transmitter and a receiver; a packet of ultrasonic waves is generated and the resulting echo is detected. The time delay between transmission and reception is assumed to be proportional to the distance of the sensed obstacle.

The multilobed beam pattern of the transmitter can be obtained from the radiation directivity function of a plane circular piston

$$D(\vartheta) = 2\frac{J_1(\omega p \sin\vartheta)}{\omega p \sin\vartheta}, \tag{1}$$

where $J_1(\cdot)$ is the first-order Bessel function, $\omega = 2\pi/\ell$ depends on the wavelength ℓ, p is the piston radius, and ϑ is the azimuthal angle measured with respect to the beam central axis. For the Polaroid sensor [21], it is $p = 0.019\,21$ m and $\ell = c/\nu$, where c is the sound speed in air and $\nu = 49.410$ kHz. For practical purposes, it is sufficient to take into account only the principal lobe of the pattern. As a consequence, the waves are considered to be diffused over a radiation cone of 25° width.

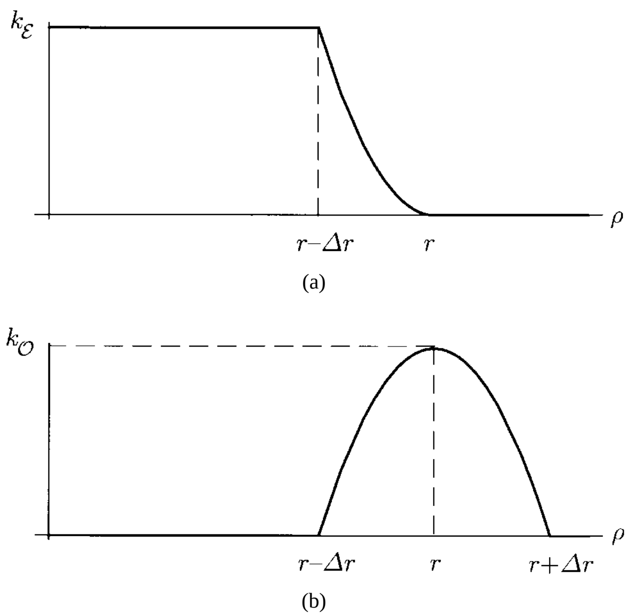
A single range reading r is affected by three basic sources of uncertainty as follows.

- The sensor has a limited radial resolution. The standard Polaroid range finder can detect distances from 0.12 to 6.5 m with 1% accuracy over the entire range.
- The angular position of the object that originated the echo inside the radiation cone is not determined. For example, all the three obstacles of Fig. 4 will give the same distance reading.
- If the incidence angle is larger than a critical value ϕ, the sensor reading is not significant because the beam may reach the receiver after multiple reflections, or even get lost (see Fig. 4). The angle ϕ depends on the surface characteristics, ranging from 7° to 8° for smooth glass to almost 90° for very rough materials.

During the perception phase, ultrasonic sensors are fired in such a sequence that interference phenomena are minimized, and measures are recorded together with the position of the corresponding sensor. At each robot position, the ultrasonic ring undergoes two consecutive rotations of 7.5°; as a consequence, $16 \times 3 = 48$ range readings are obtained. Each point of surrounding area falls inside a minimum of three radiation cones. This redundancy of measures will be exploited in the processing phase in order to achieve a more accurate estimate of the angular position of the detected obstacle.

B. Processing

The objective of this phase is to build two local fuzzy sets $\mathcal{E}^k$ and $\mathcal{O}^k$. $\mathcal{E}^k$ ($\mathcal{O}^k$) is obtained by merging the sets $\mathcal{E}_i^k$ ($\mathcal{O}_i^k$), for $i = 1, \ldots, n$, each representing the evidence that a cell is empty (occupied) provided by the single range reading r_i^k. The membership functions of $\mathcal{E}_i^k$ and $\mathcal{O}_i^k$ have a simple structure



(a)

(b)

Fig. 5. The two certainty functions (a) $f_{\mathcal{E}}$ and (b) $f_{\mathcal{O}}$ for a range reading r.

that reflects the previous qualitative description of the sensing uncertainty.

A single reading r provides the information that one or more obstacles are located somewhere along the 25° arc of circumference of radius r. Hence, while points located in the proximity of this arc are likely to be occupied, there is evidence that points well inside the circular sector of radius r are empty. To model this knowledge, we introduce the two functions

$$f_{\mathcal{E}}(\rho, r) = \begin{cases} k_{\mathcal{E}} & 0 \le \rho < r - \Delta r \\ k_{\mathcal{E}}\left(\frac{r-\rho}{\Delta r}\right)^2 & r - \Delta r \le \rho < r \\ 0 & \rho \ge r \end{cases} \tag{2}$$

$$f_{\mathcal{O}}(\rho, r) = \begin{cases} 0 & 0 \le \rho < r - \Delta r \\ k_{\mathcal{O}}\left[1 - \left(\frac{r-\rho}{\Delta r}\right)^2\right] & r - \Delta r \le \rho < r + \Delta r \\ 0 & \rho \ge r + \Delta r, \end{cases} \tag{3}$$

that describe, respectively, how the degree of belief of the assertions "empty" and "occupied" vary with ρ for a given range reading r. Here, ρ is the distance from the sensor, $k_{\mathcal{E}}$ and $k_{\mathcal{O}}$ are two positive constants corresponding to the maximum values attained by the functions, and $2 \cdot \Delta r$ is the width of the area considered "proximal" to the arc of radius r. The value of Δr should be selected in such a way to "augment" slightly the occupied area, providing a convenient safety margin for the navigation phase. The profile of $f_{\mathcal{E}}$ and $f_{\mathcal{O}}$ is displayed in Fig. 5.

Since the intensity of the waves decreases to zero at the borders of the radiation cone, the degree of belief of each assertion is assumed to be higher for points close to the beam axis. This is realized by defining a *modulation* function

$$m(\vartheta) = \begin{cases} D(\vartheta) & |\vartheta| \le 12.5° \\ 0 & |\vartheta| > 12.5° \end{cases}$$

where $D(\vartheta)$ is the radiation directivity function (1) (see Fig. 6).

Finally, we wish to limit the influence of the range reading r to an area close to the sensor location. In particular, by defining the *visibility* function

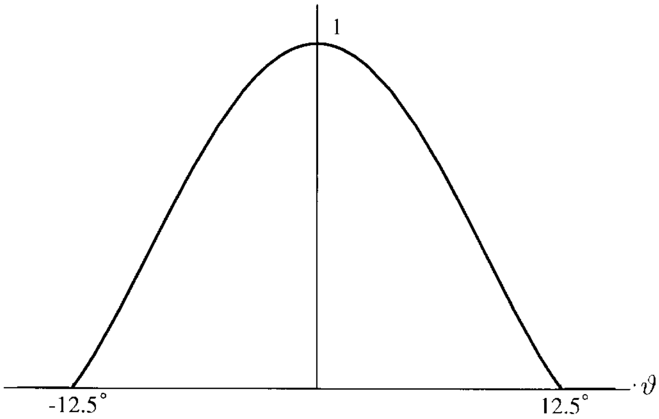


Fig. 6. The modulation function $m(\vartheta)$.

$$v(\rho) = \begin{cases} 1 & \rho \le \rho_v \\ 0 & \rho > \rho_v \end{cases}$$

the degree of belief of the assertions "empty" and "occupied" is nonzero only inside a circular sector of radius ρ_v centered at the sensor position. The motivation behind this choice is the following. Due to the sensor wide radiation angle, narrow passages (e.g., doors) would appear obstructed if seen from a large distance. By reducing ρ_v we guarantee that such passages are incorporated into the map only when they have been correctly detected. An interesting side effect of the use of $v(\rho)$ is to reduce the artifacts introduced in the map by false reflections; in fact, the latter typically produce longer range readings.

An appropriate value of ρ_v can be easily selected on the basis of qualitative characteristics of the environment. For example, in an office-like area it is reasonable to ask that the map building algorithm correctly detects passages of the size of a door, say 0.7 m. To this end, the perception point must be sufficiently close to the opening, so that the 25° radiation cone does not intersect the door features. A simple geometric construction shows that the maximum admissible distance is 1.5 m; therefore, it would be necessary to set $\rho_v < 1.5$ m.

For each range measure r_i^k, two fuzzy sets $\mathcal{E}_i^k$ and $\mathcal{O}_i^k$ are generated by defining their membership functions as

$$\mu_{\mathcal{E}_i^k}(\rho, \vartheta) = f_{\mathcal{E}}(\rho, r_i^k) m(\vartheta) v(\rho)$$
$$\mu_{\mathcal{O}_i^k}(\rho, \vartheta) = f_{\mathcal{O}}(\rho, r_i^k) m(\vartheta) v(\rho)$$

i.e., by *and*-ing the previously introduced certainty functions. These sets represent, respectively, how the degrees of belief of the assertions "empty" and "occupied" vary inside the radiation cone. Note that the above membership functions are expressed in local polar coordinates with respect to the sensor position, and assume nonzero values only inside the subset of the radiation cone within the visibility radius. Fig. 7 shows the typical shape of the two sets.

The final step of the processing phase consists in the computation of the fuzzy sets $\mathcal{E}^k$ and $\mathcal{O}^k$, that collect all the local information about emptiness and occupancy acquired at the kth perception point. This is simply realized by means of

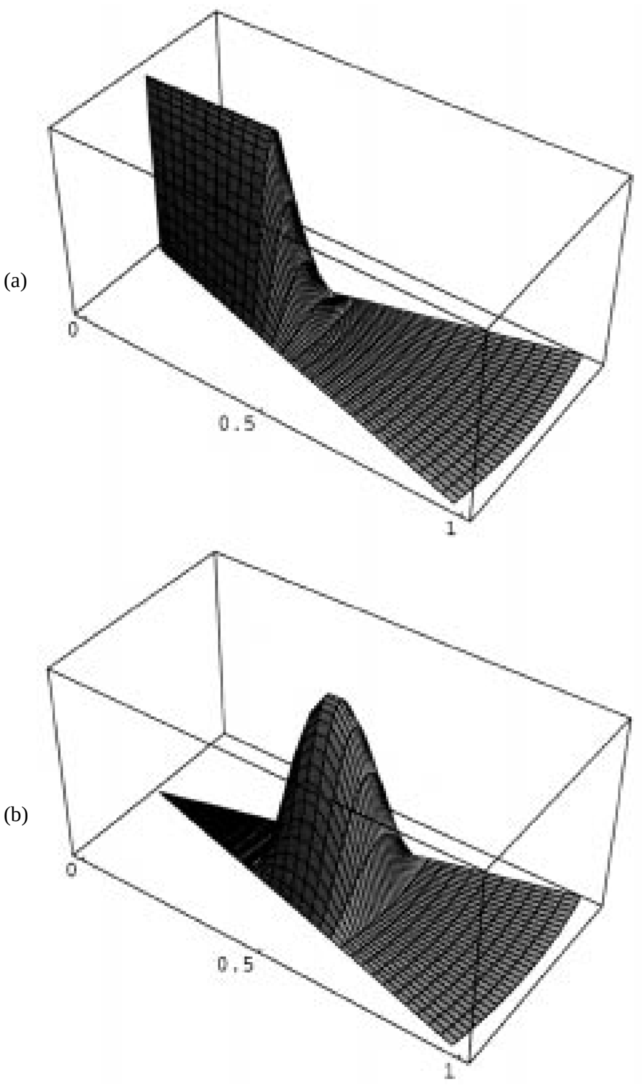


Fig. 7. Membership functions for (a) $\mathcal{E}_i^k$ and (b) $\mathcal{O}_i^k$ inside the sensor radiation cone. The sensor reading is assumed to be $r_i = 0.5$ m.

a fuzzy *union* operator

$$\mathcal{E}^k = \cup_i \mathcal{E}_i^k \tag{4}$$
$$\mathcal{O}^k = \cup_i \mathcal{O}_i^k. \tag{5}$$

In particular, for the above computation we have chosen the Dombi union operator, whose aggregation strength can be tuned by choosing a single parameter λ [see the Appendix, and specifically (18)].

Note the following points.

- Since we intend to build a grid-based representation of the environment, during the computation of $\mathcal{E}^k$ and $\mathcal{O}^k$ it is necessary to perform a conversion from polar coordinates, relative to each sensor location, to absolute Cartesian coordinates. As a result, two numerical values $\mu_{\mathcal{E}^k}(C)$ and $\mu_{\mathcal{O}^k}(C)$ are associated with each cell C, respectively quantifying its degree of membership to $\mathcal{E}^k$ and $\mathcal{O}^k$.
- The membership functions $\mu_{\mathcal{E}^k}(C)$ and $\mu_{\mathcal{O}^k}(C)$ may be nonzero only for cells contained in the circle of radius ρ_v centered at the kth perception point. Besides, due to the associative property of the union operator, $\mathcal{E}^k$ and $\mathcal{O}^k$ are independent on the order of computations in (4) and (5).
- The membership degree of a cell inside the radiation cone to the set $\mathcal{E}^k$ $(\mathcal{O}^k)$ is always increased with respect to the membership degree to each original set $\mathcal{E}_i^k$ $(\mathcal{O}_i^k)$. This fact, due to the use of the Dombi union operator, is consistent with the perception strategy described in Section III-A. In fact, assume that an obstacle cell is located at C. Since C falls inside q radiation cones ($q = 3$ or 4 in our implementation), there will be q concordant range readings—in the absence of false reflections—indicating the presence of an obstacle along an arc of circle of $25° + (q-1) \cdot 7.5°$ passing through C. The membership function $\mu_{\mathcal{O}^k}$ will be maximal where the q radiation cones overlap, i.e., in a small area around C. This expedient allows to circumvent the poor directionality of the ultrasonic sensor.
- The values of the parameters $k_{\mathcal{E}}$ and $k_{\mathcal{O}}$ in the certainty functions and of λ in the Dombi union can be directly chosen on the basis of the above reasoning. Consider a cell C that, during the kth perception, falls in the "empty" area of q cones. Each set $\mathcal{E}_i^k$ $(i = 1, \ldots, q)$ will contribute to $\mathcal{E}(C)$ a nonzero value, that is computed according to (4). Immediate saturation of $\mathcal{E}(C)$ should be avoided, so as to require the concordance of q measures to build up a high degree of certainty. To this end, it is necessary to set an upper bound on the membership function of $\mathcal{E}_i^k$ by choosing a "small" value for $k_{\mathcal{E}}$ in (2). Then, an appropriate λ can be selected for the Dombi union (see also the plots in Fig. 21). Similar considerations can be repeated for the choice of $k_{\mathcal{O}}$ in (3).

C. Fusion

The task of the fusion phase is twofold, i.e., i) to integrate the local information contained in $\mathcal{E}^k$ and $\mathcal{O}^k$ into the global fuzzy sets $\mathcal{E}$ and $\mathcal{O}$ of empty and occupied points (*aggregation*), and ii) to combine appropriately these sets in order to compute the two fuzzy maps $\mathcal{M}_m$ and $\mathcal{M}_p$ that are used in the navigation process (*map update*).

1) Aggregation: The fusion of data represented by fuzzy sets can be performed using many different *aggregation* operators (see Appendix). The selection of the most appropriate operator should be done on the basis of the specific nature and source of the data. An interesting survey covering this issue can be found in [22].

We have already found an example of data aggregation in the computation of the local fuzzy sets $\mathcal{E}^k$ and $\mathcal{O}^k$ by (4) and (5). The choice of the Dombi union operator (which is not idempotent) can be regarded from a more abstract point of view as a *consensus buildup* mechanism: for example, one range reading indicating that a cell C is empty induces a certain degree of belief that C belongs to $\mathcal{E}^k$, and this degree of belief is increased as more range readings confirm this opinion.

In order to incorporate $\mathcal{E}^k$ into $\mathcal{E}$ and $\mathcal{O}^k$ into $\mathcal{O}$, we can choose among union and averaging operators. The incremental consensus buildup obtained through the use of the Dombi union operator is still a suitable strategy in the presence of a *stationary* environment. In this case, the global fuzzy sets $\mathcal{E}$ and $\mathcal{O}$ are updated as

$$\mathcal{E} := \mathcal{E} \cup \mathcal{E}^k \tag{6}$$

$$\mathcal{O} := \mathcal{O} \cup \mathcal{O}^k. \tag{7}$$

When the environment is subject to changes over time, the aggregation operator must be able to *decrease* as well as to increase the degree of belief of the assertions "empty" and "occupied." Hence, one is naturally led to make use of an averaging operator

$$\mathcal{E} := \mathcal{E} \sqcup \mathcal{E}^k$$

$$\mathcal{O} := \mathcal{O} \sqcup \mathcal{O}^k.$$

A simple choice for the operator $\sqcup$ is the weighted arithmetic mean [see (19)]. In particular, the new values of the membership functions for a cell C are computed as

$$\mu_{\mathcal{E}\sqcup\mathcal{E}^k}(C) = \frac{(N_k(C)-1)\mu_{\mathcal{E}}(C) + \mu_{\mathcal{E}^k}(C)}{N_k(C)} \tag{8}$$

$$\mu_{\mathcal{O}\sqcup\mathcal{O}^k}(C) = \frac{(N_k(C)-1)\mu_{\mathcal{O}}(C) + \mu_{\mathcal{O}^k}(C)}{N_k(C)} \tag{9}$$

where $N_k(C)$ is the number of perceptions (including the kth) that have involved cell C so far.

The above formulas can be interpreted as follows: the more concordant opinions have been recorded on the status of a cell C (i.e., the greater is $N_k(C)$), the more reliable become the degrees of belief represented by the membership functions $\mu_{\mathcal{E}}$ and $\mu_{\mathcal{O}}$. At the same time, a large number of discordant opinions is needed to change appreciably the degrees of belief. This kind of behavior may be a drawback, because it affects the promptness of the map building method in recording changes of the surrounding scene. Besides, in dynamic environments the number of concordant opinions is not a guarantee in itself, unless they have been obtained over a uniform time distribution.

In order to reduce the inertia due to the number of measures, we have included a simple *saturation* mechanism in the computation of $N_k(C)$. Namely, in (8) and (9) we have used a modified value $N'_k(C)$ computed as

$$N'_k(C) = \min(N_{\max}, N_k(C)) \tag{10}$$

where $N_{\max}$ is an integer representing the maximum inertia we attribute to past measures. According with the terminology of [22], this implements a kind of *context-dependent* mean operator, i.e., an operator whose result does not depend only on the value of the arguments but also on external knowledge (in this case, the number of measures).

Some comments are as follows.

- More sophisticated versions of context-dependent mean operators could be implemented by choosing $N'_k(C)$ as a function, say, of the amount of contradiction in the available information. However, we have preferred to delay such filtering operations until the map update phase.
- Although mean operators perform well also in stationary environments, in this case their use is somewhat less efficient with respect to union operators. In fact, in order to apply (8) and (9) it is necessary to memorize the information relative to the number of measures for each cell, which is avoided with (6) and (7).
- Even in the presence of a dynamic environment, we shall keep the choice of union operators to perform the processing phase [see (4) and (5)]. This is reasonable as long as the characteristics of the scene do not change appreciably during a perception phase. In other words, we are assuming that the environment is *moderately* dynamic.
- Since $\mu_{\mathcal{E}^k}(C)$ and $\mu_{\mathcal{O}^k}(C)$ are nonzero only inside a circle of radius ρ_v centered at the perception point, it is necessary to update $\mathcal{E}$ and $\mathcal{O}$ only inside the same area. This is true regardless of the chosen aggregation operator.

2) Map Update: The final step of the map building process consists in updating the two fuzzy maps $\mathcal{M}_m$ and $\mathcal{M}_p$ to account for variations in $\mathcal{E}$ and $\mathcal{O}$.

Besides its computational efficiency, a fuzzy logic framework presents the advantage of allowing the detection of conflicting or insufficient information. In fact, since $\mathcal{E}$ and $\mathcal{O}$ are not complementary, their intersection is the fuzzy set of *ambiguous* cells, with the corresponding membership value representing the degree of contradiction

$$\mathcal{A} = \mathcal{E} \cap \mathcal{O}. \tag{11}$$

Similarly, the fuzzy set of *indeterminate* cells can be defined as

$$\mathcal{I} = \bar{\mathcal{E}} \cap \bar{\mathcal{O}}.$$

A conservative map $\mathcal{S}_m$ of the *safe-for-motion* cells is obtained by "subtracting" the *occupied*, the *ambiguous* and the *indeterminate* cells from the *very empty*[2] ones

$$\mathcal{S}_m = \mathcal{E}^2 \cap \bar{\mathcal{O}} \cap \bar{\mathcal{A}} \cap \bar{\mathcal{I}}. \tag{12}$$

The *motion map* $\mathcal{M}_m$ is built by complementing $\mathcal{S}_m$ and identifies cells that must be avoided during robot motion

$$\mathcal{M}_m = \bar{\mathcal{S}}_m.$$

While indeterminate cells are penalized in $\mathcal{M}_m$, for planning purposes we shall need a second fuzzy map in which they are regarded as admissible for planning. In particular, we define the *safe-for-planning* cells as

$$\mathcal{S}_p = \mathcal{E}^2 \cap \bar{\mathcal{O}} \cap \bar{\mathcal{A}} \cup \mathcal{I} \tag{13}$$

and the *planning map* $\mathcal{M}_p$ is

$$\mathcal{M}_p = \bar{\mathcal{S}}_p.$$

The complementation operator (16), the bounded product intersection operator (17) and the Dombi union operator (18) are used to perform the above computations. For compactness, the membership degree of a cell C to $\mathcal{M}_m$ and $\mathcal{M}_p$ will be simply denoted in the following by $\mu_m(C)$ and $\mu_p(C)$, respectively.

We emphasize that the sets of cells so far described (indeterminate, ambiguous, etc.) do *not* correspond to actual data

[2] By squaring the value of the membership function of $\mathcal{E}$, the difference between low and high values is emphasized: according to the fuzzy logic terminology, we are applying the linguistic modifier "very" to the "empty" concept.

structures that are memorized during the map building process. In fact, the maps $\mathcal{M}_m$ and $\mathcal{M}_p$ are directly computed from the fuzzy sets $\mathcal{E}$ and $\mathcal{O}$ by expliciting the various terms in (12) and (13), respectively, and exploiting the associative property of the various operators.

The worst-case time complexity of the corresponding algorithm is linear in Σ, that is the number of cells of the bitmap representation. Note also that, since $\mathcal{M}_m$ and $\mathcal{M}_p$ need to be updated only inside the visibility circle, the average complexity is much smaller.

3) Map Building—Experimental Results: Before proceeding with the exposition of the navigation process, we shall present experimental results of the map building process in a stationary environment. The experiment area, containing a corridor and a room, was represented as a bitmap of 180×120 square cells of side $\delta = 0.1$ m. Halfway along the corridor there is an intersection with a small obstacle. The open space is delimited by flat surfaces (walls and closed glass cabinets) with poor diffraction properties, an adverse condition for ultrasonic sensing. NOMAD performed 43 perceptions at different positions attained along a path under operator guidance.

For the map building algorithm, we have used the following set of parameter values: $k_{\mathcal{E}} = 0.1$, $k_{\mathcal{O}} = 0.25$, $\Delta r = 0.15$ m, $\rho_v = 1.2$ m, and $\lambda = 0.4$. Being the environment stationary, the Dombi union operator was used in the aggregation phase [see (6) and (7)]. Fig. 8 shows the resulting motion map $\mathcal{M}_m$. The average time needed at each perception point to update $\mathcal{M}_m$ from the range readings was approximately 0.2 s.

Note the satisfactory accordance of the map with the actual boundary of the open space. The small gray areas extending beyond the corridor walls are due to false reflections occurring for large angles of incidence. Nevertheless, the map building algorithm was able to reconstruct accurately the profile of the walls, by incorporating the range readings obtained for incidence angles smaller than the critical value.

An extensive comparison of the proposed map building method with Bayesian techniques based on probability theory was performed in [23]. The experimental results indicated that the method based on fuzzy logic is more robust with respect to the occurrence of false reflections in the measuring process. This is basically due to the fact that in our approach the information conveyed by $\mathcal{E}$ and $\mathcal{O}$ is not complementary, thus allowing to identify areas for which contradictory evidence has been gathered [see (11)] and to regard them as dangerous [see (12)]. On the other hand, stochastic techniques based on Bayesian updating are very sensitive to the occurrence of outliers in the measuring process.

The reader is referred to [24]–[26] for a thorough discussion of the relative benefits of fuzzy set theory and probability theory as uncertainty calculus methods. However, we mention here that existing stochastic techniques for ultrasonic map building exhibit other shortcomings. In fact, to keep the problem tractable, a *zero-order Markov field* is typically assumed, i.e., that no relationship whatsoever exists between the states of two cells C_i and C_j, even if they are adjacent. However, it has been observed [25] that this assumption may induce large errors in the presence of even a slight degree of dependence between the random variables—this is exactly the case for map building, since the occupied cells are not evenly distributed, but concentrated in clusters (obstacles). Moreover, the prior probabilities needed to initialize the field are typically estimated with the *maximum entropy* assumption, namely by regarding emptiness and occupancy as equiprobable. As a consequence, the convergence of the Bayesian updating procedure toward an acceptable characterization of the occupancy grid requires a large number of measures.

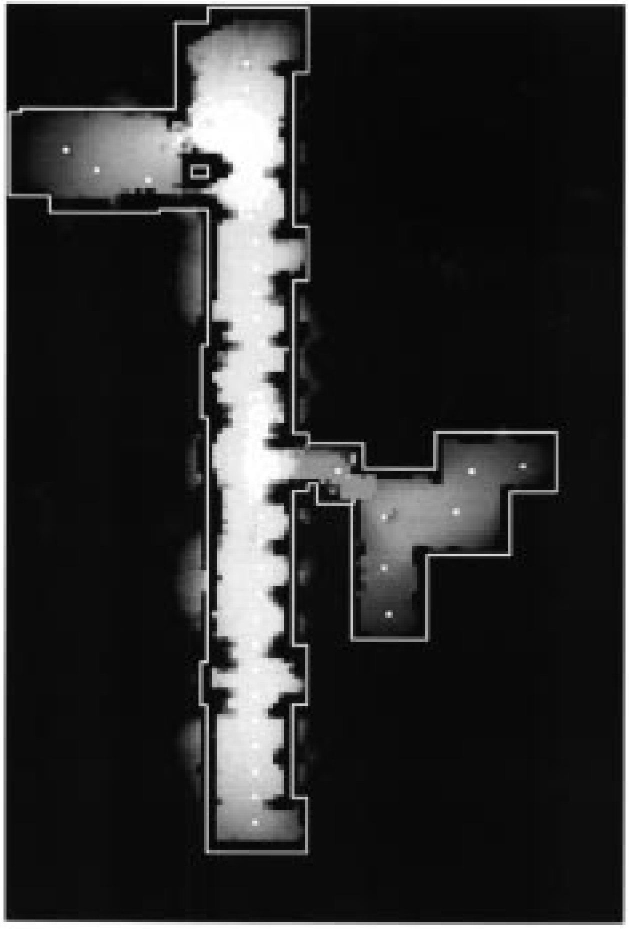


Fig. 8. The motion map $\mathcal{M}_m$ obtained for a stationary environment: darker areas correspond to higher values of μ_m. The actual profile of the corridor, the room and the obstacle are superimposed. White spots indicate the perception points.

IV. Navigation

When the kth map building process is completed, two updated fuzzy maps $\mathcal{M}_m$ and $\mathcal{M}_p$ are available for the next navigation process. The latter is accomplished by means of two sequential phases: *planning* and *motion* (see Fig. 9). During the first, an $A^{\star}$-based planner generates a subpath from the current position (the kth perception point) to the goal. By using $\mathcal{M}_p$, the planner makes use of all the available local information, but at the same time it is allowed to propose paths going through unexplored areas.

In the motion phase, the robot follows the planned subpath until the goal is reached, unless one of two *stop conditions* is true; namely, either the robot is leaving the explored area

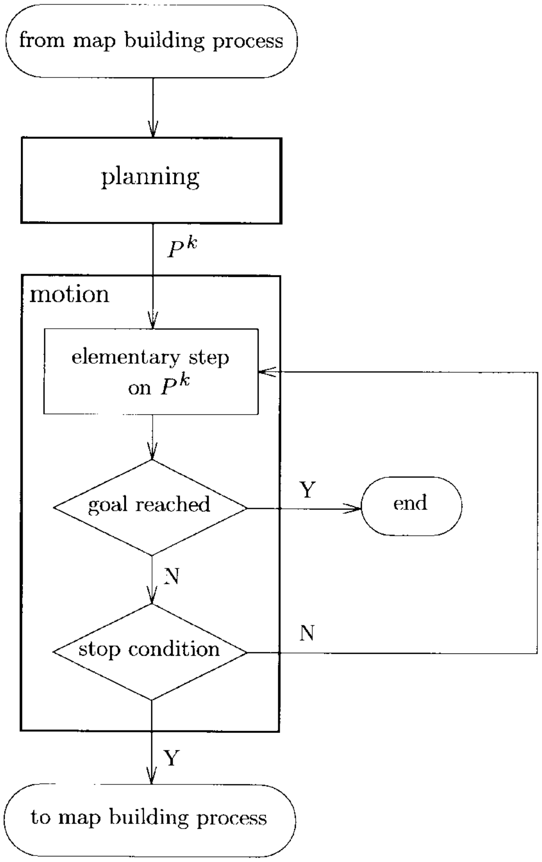


Fig. 9. The kth navigation process.

(which is clearly identifiable on $\mathcal{M}_m$) or an unexpected obstacle obstructs its motion. In this case, a new map building process takes place.

A. Planning

During the kth planning phase, a subpath P^k is produced from the current robot position (corresponding to the cell S^k) to the goal G by applying a graph search algorithm aimed at minimizing the risk along the path. The fuzzy map $\mathcal{M}_p$, in which the *indeterminate* cells are regarded as safe, is used in this step. The rationale for this choice is simple: in order to reach the goal, the robot will have to traverse regions that are indeterminate at the beginning of the motion (recall that the environment is a priori unknown). Thus, the planner must be allowed to propose subpaths going through such regions, providing directions for further explorations.

Below, we introduce various cost functions that characterize the risk associated to a path on $\mathcal{M}_p$, corresponding to more or less conservative attitudes. It is then shown how to select proper instances from the $A^\star$ class of graph search algorithms in order to compute minimum-cost paths.

1) Safe Paths on Fuzzy Maps: Consider the problem of finding a path from S^k to G (i.e., a sequence of adjacent cells $\{S^k, \ldots, G\}$) that is collision-free on $\mathcal{M}_p$. The uncertain nature of fuzzy maps does not provide a separation between the free and the occupied space. A natural planning strategy is to avoid areas of $\mathcal{M}_p$ where the risk of collision is high, that are identified by cells with large values of μ_p. This may be achieved by defining proper *cost functions* for a path P^k, and then searching for minimum-cost paths. For clarity of exposition, we first consider the case of a point robot; this assumption will be removed later.

The first and most intuitive cost function is defined as

$$g_1(P^k) = \sum_{C_i \in P^k} \mu_p(C_i),$$

that is, a measure of the integral risk along the path.

As a second possibility, we propose

$$g_2(P^k) = \sum_{C_i \in P^k} \mu_p^2(C_i).$$

Again, this corresponds to applying the modifier “very” to the “unsafe” concept. By doing so, the inclusion of cells with high values of μ_p in a minimum-cost path is less likely.

Finally, consider

$$g_3(P^k) = \max_{C_i \in P^k} \mu_p(C_i),$$

which represents the maximum risk encountered on the path.

These cost functions may be interpreted as *energy measures*[3] [19], [27] of the fuzzy set $\mathcal{M}_p$ over the path P^k, and characterize the distance of P^k from an ideally safe path, i.e., a path for which $g_i = 0$ $(i = 1, 2, 3)$. Their minimization is hence a reasonable objective.

Note that g_1, g_2, and g_3 measure the above distance according to different metrics. For example, assume that S^k and G are far, so that any admissible path connecting them consists of a large number of cells. A path minimizing g_1 will yield a tradeoff between length and risk, and may traverse a cell with a large μ_p, if in this way its length were significantly reduced. A more conservative strategy would be to minimize g_3, even if this might result in a longer path. The use of g_2 will typically generate paths with intermediate characteristics between g_1 and g_3.

On the other hand, consider a situation in which S^k is located in a region with uniform high values of μ_p. An optimal path with respect to g_3 might be unsatisfactory, since the maximum value of μ_p will probably be attained near S^k. Thereafter, any subpath leading to G is admissible as long as it does not increase the value of g_3. Minimization of g_1 or g_2 will produce better results, since both these functions depend on the *whole* path from S^k to G. The results of Section IV-A3 will further clarify this discussion.

2) Planning on Fuzzy Maps with the $A^\star$ Algorithm: In view of the above discussion, the kth planning phase must solve the problem “Find a minimum-cost (g_1, g_2, or g_3) path from S^k to G on $\mathcal{M}_p$.” As a planning method, we have adopted the $A^\star$ algorithm, which allows to incorporate heuristic information when available, resulting in an efficient search. We shall not recall here the details of the algorithm, that are well known [28].

To apply $A^\star$, we need as a basic tool a heuristic function $h(C)$ estimating the cost of the optimal path from the

[3] In the terminology of [27], g_1 and g_3 are, respectively, the *power* and the *height* of $\mathcal{M}_p$ over P^k.

generic cell C to the goal G. $A^\star$ will be complete under the *admissibility* condition

$$0 \le h(C) \le h^*(C), \quad \forall C \tag{14}$$

where $h^*(C)$ is the *actual* cost of the minimum-cost path from C to G. Moreover, the heuristic function $h(\cdot)$ is said to be *locally consistent* if, for any pair of adjacent cells (C_i, C_j), we have

$$0 \le h(C_i) \le h(C_j) + w(C_i, C_j)$$

being $w(C_i, C_j)$ the cost of the arc between C_i and C_j. Under this assumption, whenever C_i is expanded during the algorithm visit, the current path from S^k to C_i is already optimal. The choice $h(\cdot) \equiv 0$ is trivially admissible and locally consistent, resulting however in a *non-informed* algorithm.

The use of $A^\star$ to generate paths minimizing g_1 on the fuzzy map $\mathcal{M}_p$ is immediate. The cost of the arc joining two adjacent cells C_i and C_j is defined as

$$w_1(C_i, C_j) = \mu_p(C_j)$$

so that the cost of a path P^k coincides with $g_1(P^k)$, except for the additive constant $\mu_p(S^k)$. As for the heuristic function, we use

$$h_1(C_j) = d(C_j) \cdot \mu_p^{\min} \tag{15}$$

in which $d(C_j)$ is the minimum number of cells that compose a subpath from C_j to G, and $\mu_p^{\min}$ is the smallest value of μ_p over $\mathcal{M}_p$ (a quantity that can be memorized during the map building phase). The heuristic function (15) is clearly admissible and locally consistent.

Two remarks are necessary at this point.

- The value $d(C_j)$ depends on the adjacency definition on the map. If 1-adjacency is used, each cell has four adjacents and thus $d(C_j) = |x_G - x_j| + |y_G - y_j|$, being (x_j, y_j) and (x_G, y_G) the coordinates of C_j and G, respectively. When using 2-adjacency, each cell has eight adjacents and we get $d(C_j) = \max(|x_G - x_j|, |y_G - y_j|)$.
- To obtain an informed $A^\star$, it must be $\mu_p^{\min} > 0$. Hence, it is advisable to offset all values of μ_p by a small positive constant. In the following, we shall directly assume that $\mu_p^{\min} > 0$.

The resulting version of the $A^\star$ algorithm will be denoted by $A_1^\star$. The algorithm $A_2^\star$ for minimization of g_2 is obtained by obvious modifications of $A_1^\star$.

The computation of a minimum-g_3 path via $A^\star$ is somewhat nonstandard, the difficulty coming from the non-additive nature of this cost function. In order to define the arc cost $w_3(C_i, C_j)$ in such a way that the cost of a path is g_3, we let

$$w_3(C_i, C_j) = \begin{cases} \mu_p(C_j) - \hat{\mu}_p(C_i) & \text{if } \mu_p(C_j) > \hat{\mu}_p(C_i) \\ 0 & \text{else} \end{cases}$$

where $\hat{\mu}_p(C_i)$ denotes the maximum value of μ_p encountered on the optimal path from S^k to C_i. With this definition, the arc costs are not known *a priori*. Nevertheless, they are computable as the algorithm proceeds provided that a locally consistent function h_3 is used, since in this case we have simply $\hat{\mu}_p(C_i) = g_3(C_i)$.

To satisfy the admissibility condition, it is necessary to set $h_3(C_j) = 0$; in fact, if the maximum value of μ_p on the path is attained on the subpath from S^k to C_j, the remaining part of the path will not increase the cost function g_3, i.e., $h_3^*(C_j) = 0$. However, a non-informed $A^\star$ may be dramatically inefficient, due to the large number of *ties* that may occur during the algorithm expansion. To cope with this problem, we have used a tie-resolution strategy privileging cells whose distance d from G is smaller. With this modification, $A^\star$ works like a *depth-first* search method whenever this is possible without increasing the value of g_3. The algorithm just described will be indicated by $A_3^\star$.

Having described the main features of the planning algorithms, some remarks are in order regarding their iterative application as a component of the real-time navigation process.

- We can easily remove the point robot assumption as follows. Assume that the robot can be approximated by a circle of radius γ, whose center is located at (the center of) cell C. Since each bitmap cell has side δ $(\delta < \gamma)$, the robot body will be contained in a square of $\eta \times \eta$ cells centered at C, being $\eta = 2\,\text{round}(\gamma/\delta) + 1$, with round$(x)$ the nearest integer to x. Hence, we can build an *augmented* map $\mathcal{M}_p^a$ by defining $\mu_p^a(C)$ as the maximum value of μ_p attained in the square of $\eta \times \eta$ cells centered at C. Planning for a point in $\mathcal{M}_p^a$ is equivalent to planning for the actual robot in $\mathcal{M}_p$. Such procedure may be implemented by preliminarily processing the map $\mathcal{M}_p$, but this is not necessary. In fact, it is sufficient to modify $A^\star$ so as to compute $\mu_p^a(C)$ only when C is actually visited.
- Due to the incremental nature of the map building process, the planning map $\mathcal{M}_p$ searched by the planner will typically include indeterminate areas containing G. As mentioned above, cells inside these areas are characterized by a constant value $\mu_p^{\min}$, with $\mu_p^{\min} > 0$ by construction. Therefore, outside the explored area, the generated paths will aim directly at the goal in order to minimize the heuristic function that takes into account the distance d from the goal.
- It was already noted that, although $A_1^\star$ and $A_2^\star$ are informed and produce better paths (they minimize an integral risk) than $A_3^\star$, the maximum risk along the path may be higher. To alleviate this problem, we have chosen to perform an α- *cut* of $\mathcal{M}_p$ whenever $A_1^\star$ or $A_2^\star$ are used. That is, only cells belonging to the (crisp) subset

$$M_p^\alpha = \{C \in U : \mu_p \le \alpha\}$$

are considered admissible for planning. By choosing an appropriate value for α, we can obtain a reasonable tradeoff between the integral and the maximum risk.
- For a graph with m arcs and n nodes, the time complexity of $A^\star$ is $O(m \log n)$. Since our representation of the environment is a bitmap of Σ cells, the number of arcs is $m = 2\Sigma^2 - 3\Sigma + 1$ when 2-adjacency is used. The resulting worst-case time complexity for $A_1^\star$, $A_2^\star$ and $A_3^\star$ is $O(\Sigma^2 \log \Sigma)$.

TABLE I
RESULTS FOR SIMULATION 1

	g_1	g_2	g_3	cells in the path	expanded cells	computing time (sec)
$A_1^\star$	21.80	5.34	0.50	121	2066	3.13
$A_2^\star$	21.80	5.34	0.50	121	2394	6.21
$A_3^\star$	45.50	17.13	0.50	145	378	0.08

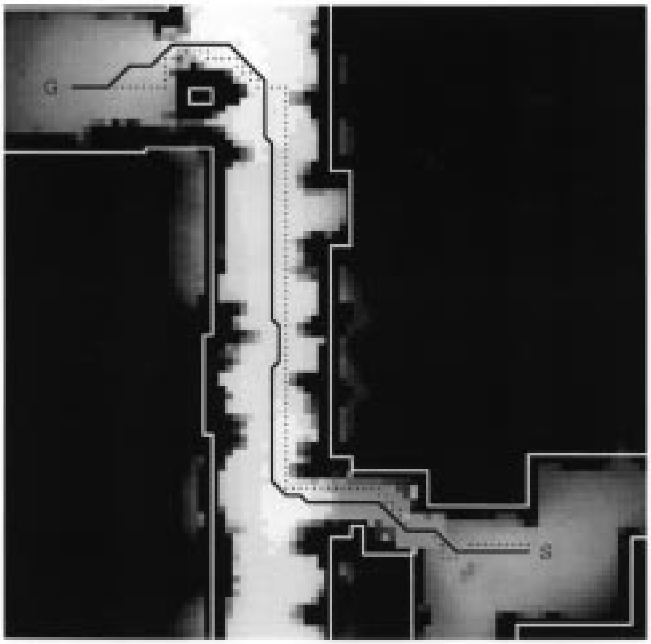


Fig. 10. Simulation 1: Paths generated by $A_1^\star$ and $A_2^\star$ (continuous) and $A_3^\star$ (dotted).

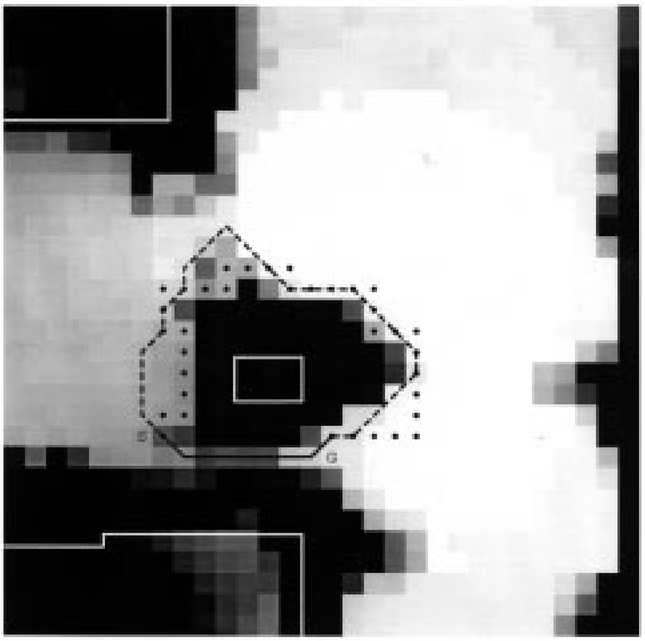


Fig. 11. Simulation 2: Paths generated by $A_1^\star$ (continuous), $A_2^\star$ (dashed) and $A_3^\star$ (dotted).

3) Planning—Simulation Results: In order to show the performance of the planning algorithms $A_1^\star$, $A_2^\star$, and $A_3^\star$, we present below some simulation results of their application in a *known* static environment. In particular, we have used the fuzzy map of Fig. 8. Experimental results obtained with real-time map building and navigation in unknown environments will be presented in Section V. In order to take into account the dimension of NOMAD, the aforesaid map augmentation procedure was performed with $\delta = 0.1$ m and $\gamma = 0.23$ m, giving $\eta = 5$.

In the first simulation, the start cell S is located inside the room, while the goal cell G is at the end of the left branch of the corridor (see Fig. 10). The straight line distance between S and G is approximately 12 m. Details on the solution paths are given in Table I. Note that, even if the paths produced by $A_1^\star$ and $A_2^\star$ coincide, the computing time differs substantially for the two algorithms. As for $A_3^\star$, the solution path is more dangerous in the integral sense (the value of g_1 is roughly doubled with respect to $A_1^\star$ and $A_2^\star$), but the maximum risk along the path is 0.5, as before. This value is encountered in correspondence of the narrow crossing between the room and the corridor (a forced passage since planning is performed on the augmented map).

The algorithm $A_3^\star$, although non-informed, is much faster than the other two algorithms. This is due to the tie-resolution strategy, that induces a depth-first behavior when possible. As a drawback, while 2-adjacency was chosen for $A_1^\star$ and $A_2^\star$, we had to use 1-adjacency for $A_3^\star$, to prevent the path from grazing the walls.

To better illustrate the different behavior of the algorithms, we have considered a second simulation, with the start cell S and the goal cell G located on the opposite sides of the small obstacle, as shown in Fig. 11 on a magnified view. Here, planning is performed for a point robot, and the risk value associated with the start cell is 0.5. Details on the performance of the algorithms are given in Table II. The path produced by $A_1^\star$ turns around the obstacle in the counterclockwise direction, while those generated by $A_2^\star$ and $A_3^\star$ take the clockwise direction. In particular, $A_1^\star$ returns a (shorter) path with the lowest possible value of integral risk g_1, but traversing more dangerous cells, as indicated by the value of $g_3 = 0.6$. Instead, the risk over the $A_3^\star$ path does never increase beyond the initial value 0.5. As expected, $A_2^\star$ displays in this case an intermediate behavior between the other two algorithms. In fact, while the maximum risk attained along the path is again 0.5, the integral risk is lower than for $A_3^\star$, since each cell in the path contributes to the cost function g_2.

In general, $A_3^\star$ is faster in returning a solution path, provided that no *backtracking* phase is necessary. On the other hand, $A_1^\star$ and $A_2^\star$ produce generally safer and smoother paths.

TABLE II
RESULTS FOR SIMULATION 2

	g_1	g_2	g_3	cells in the path	expanded cells	computing time (sec)
$A_1^\star$	3.1	1.43	0.60	9	108	0.09
$A_2^\star$	3.7	0.69	0.50	25	168	0.16
$A_3^\star$	5.8	1.22	0.50	35	65	0.01

Time efficiency was not stressed in our implementation. A considerable increase in speed could be obtained by using a suboptimal version of $A^\star$, as proposed in [29].

B. Motion

Once the chosen $A^\star$-based planner has produced a subpath P^k on $\mathcal{M}_p$ from the current robot position S^k to the desired robot goal G, the motion phase takes place. In particular, the robot follows the path P^k up to the goal unless one of the two following *stop conditions* is verified.

1) The robot is leaving the explored area. This condition is indicated by the fact that the robot is leaving a specified β-cut of the motion map $\mathcal{M}_m$, i.e., the (crisp) subset of cells defined as

$$M_m^\beta = \{C \in U : \mu_m \leq \beta\}.$$

In fact, indeterminate cells on the path are identified by high values of μ_m. To account for the actual robot dimensions, also the motion map can be augmented as seen in Section IV-A2.

2) The robot detects an "unexpected" obstacle that is closer than a minimum clearance. This can be realized by continuously monitoring the measures of proximity sensors, which may be either the ultrasonic range finder themselves or other available devices. For example, in our implementation on NOMAD we have exploited the availability of a ring of 16 infrared sensors. The choice of a small clearance is more hazardous, but may allow the robot to go across narrow passages in very cluttered environments.

If any of these is met, the robot stops and executes a new map building process.

V. EXPERIMENTAL RESULTS

We shall now present experimental results obtained by applying the proposed method to the mobile robot NOMAD. The robot is assigned two navigation tasks in a cluttered area of 7.6 × 6.3 m (the universal set U) contained in our laboratory. NOMAD is constrained to plan and move inside this area only. A profile of the most significant stationary obstacles is shown in Fig. 12. In both experiments, the environment is *a priori* unknown.

As before, U is discretized into a bitmap of square cells of size 0.1 m. The parameters of the map building algorithm are $k_\mathcal{E} = 0.4$, $k_\mathcal{O} = 0.4$, $\Delta r = 0.15$ m, $\rho_v = 1.5$ m, and $\lambda = 0.4$. As an aggregation operator, we have selected the weighted arithmetic mean (8)–(9) together with the saturation mechanism (10), in which $N_{\max} = 3$.

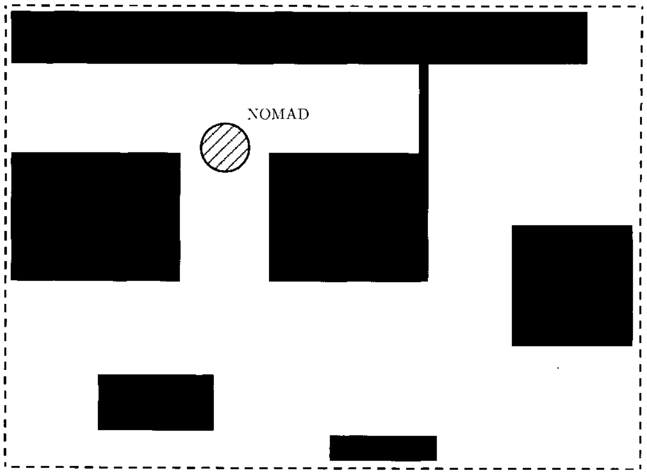


Fig. 12. Actual map of the most significant stationary obstacles inside the experiment area.

In the navigation process, $A_2^\star$ is used as a planning algorithm on the map $\mathcal{M}_p^a$ obtained through the augmentation procedure of Section IV-A2, with $\eta = 5$. Only cells having $\mu_p^a \leq 0.6$ are considered admissible for planning (i.e., $\alpha = 0.6$). During motion execution, the first stop condition is implemented on the (augmented) motion map by choosing $\beta = 0.9$. Finally, a clearance of 12 cm has been selected for the second stop condition, a value that represents the minimum detectable distance for the ultrasonic sensors but is well-contained in the infrared sensing range.

The start position at the beginning of the first experiment and the desired goal position are shown in Fig. 13. All the obstacles are stationary. Figs. 13–16 show the motion map $\mathcal{M}_m$ obtained after 4, 5, 13, and 22 perception steps, respectively, together with the path traced by the robot so far. Positions where a map building process has taken place are marked by a "+" sign.

At the beginning of the experiment, NOMAD tries to follow a direct path to G (Fig. 13) until the presence of an obstructing wall is detected and included in the fuzzy map (Fig. 14). Thereafter, it backtracks in order to find a convenient passage, driven by the heuristic information in the $A_2^\star$ planner. The robot correctly detects the presence of an opening between the two large obstacles and crosses it (Fig. 15). Afterwards, NOMAD proceeds toward the goal circumnavigating the obstacle and crossing another narrow passage. The final robot path, shown in Fig. 16, is reasonably efficient. Besides, the final map is fairly accurate.

A careful examination of the figures shows that the robot tends to stop more frequently when crossing narrow passages. This is due to the first stop condition being triggered by the

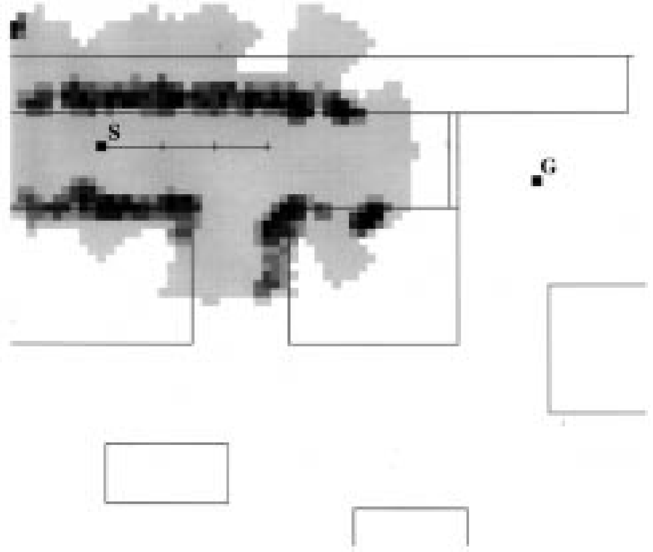


Fig. 13. Experiment 1: NOMAD tries a direct path to the goal.

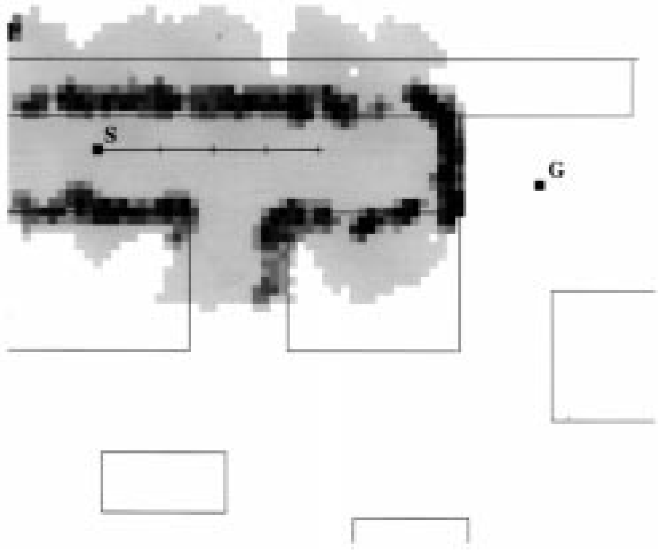


Fig. 14. Experiment 1: A wall obstructing the direct path to the goal is detected.

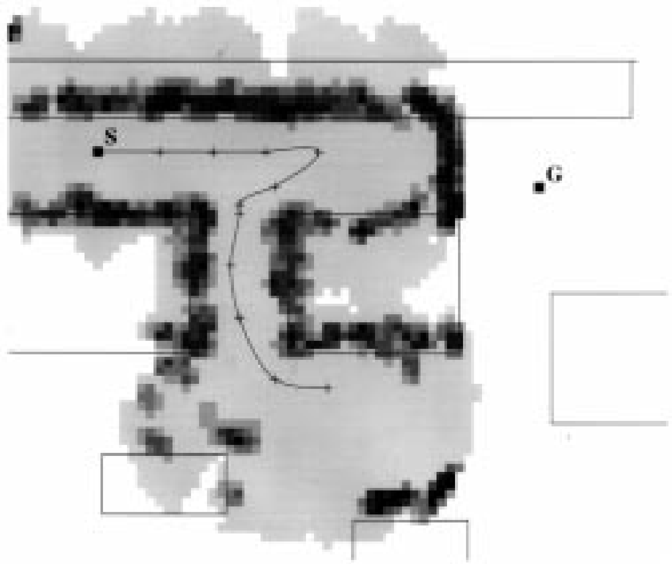


Fig. 15. Experiment 1: NOMAD tries an alternative path guided by the heuristic function of the $A_2^\star$ planner.

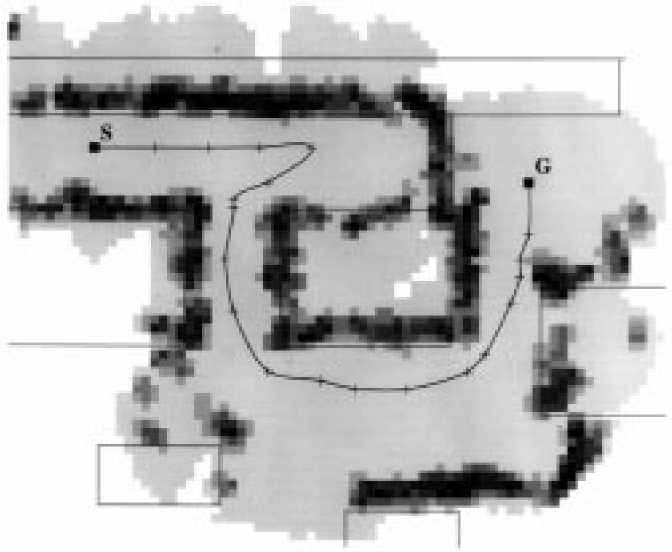


Fig. 16. Experiment 1: The final map and the path executed by NOMAD to reach the goal.

proximity sensors during the motion. Erroneous measures, frequently reported in such demanding operating conditions, are however corrected by performing a new perception phase.

The start and the goal position for the second experiment are displayed in Fig. 17. In order to show that the proposed method can cope with moderately dynamic environments, we have included a moving obstacle in the scene. The most direct path to the goal, which would cross the channel between the two large obstacles, is obstructed by the presence of a person (see Fig. 17). Therefore, NOMAD tries to find an alternative passage exploring the region on the left, until it reaches the boundary of the experiment area, as shown in Fig. 18 (note that the window is not reported into the map because it is located outside the experiment area). At this point, $A_2^\star$ returns failure because there is no path connecting the current robot position to the goal with maximum risk on $\mathcal{M}_p^a$ less than 0.6. The value of α is then tentatively increased to 0.7 in order to allow the robot to plan a path and, consequently, to move in the environment detecting possible changes. Other, less heuristic solutions can be easily envisaged to handle this kind of situation.

As a matter of fact, NOMAD is forced to come back toward the opening and recognizes that the way is now free because the person has moved (Fig. 19). Hence, the robot can cross the passage and reach the goal (Fig. 20). A comparison of the final map with the map in Fig. 18 shows that the obstacle corresponding to the moving person has been progressively erased as the robot performed new map building processes during its motion. This proves that the proposed aggregation procedure based on the weighted mean operator is effective in recording real-time changes in the environment. Indeed, a slight trace of the person is still present on the final

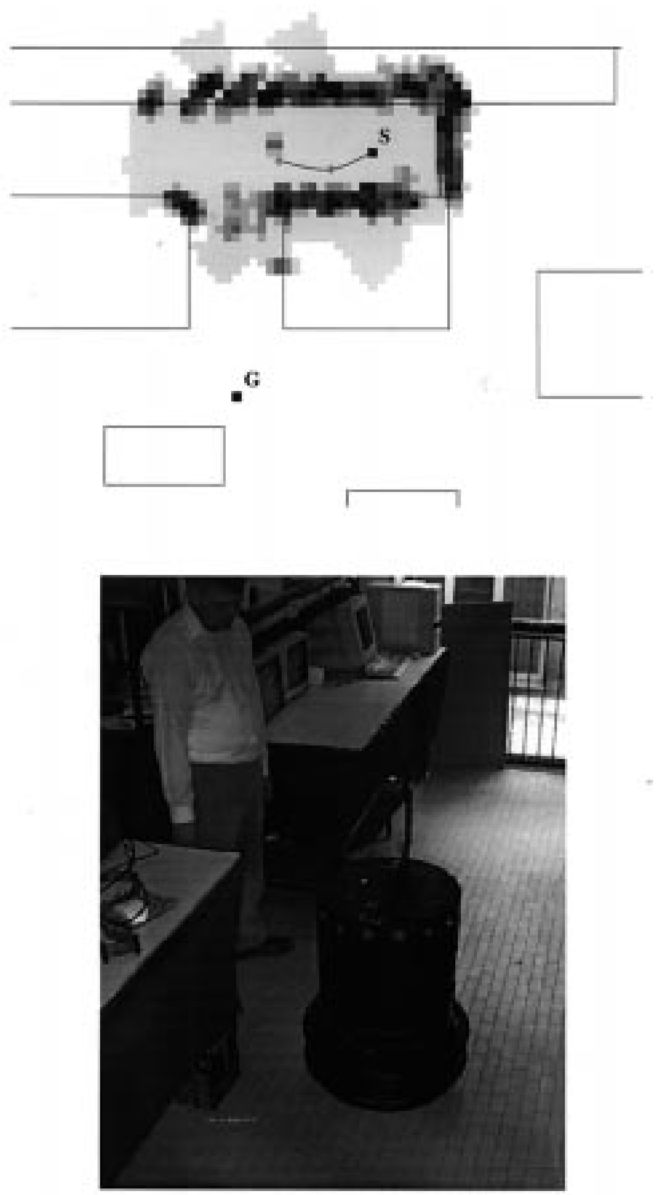


Fig. 17. Experiment 2: The channel is currently obstructed.

map, essentially because the choice $N_{\max} = 3$ results in an aggregation procedure that is relatively slow in "forgetting" previous map values. However, our experience shows that it is desirable to attribute a certain inertia to existing information to eliminate the possibility of oscillations in the map.

Both in the first and the second experiment, the total time needed to execute the given navigation task was around 2 min. However, it should be noticed that a remarkable portion of this time (about 70%) is used by the perception phase. In turn, the biggest part of the latter is spent for the radio communication between the host computer and the robot. The procedure could be sped up by bringing the whole control software on board the robot and by resorting to an architecture based on concurrent processes.

The above satisfactory behavior was confirmed by several other experiments [30], including nonstationary environments; failure to find a solution path was reported in very few situations, essentially due to the inherent limitations of ultrasonic sensors.

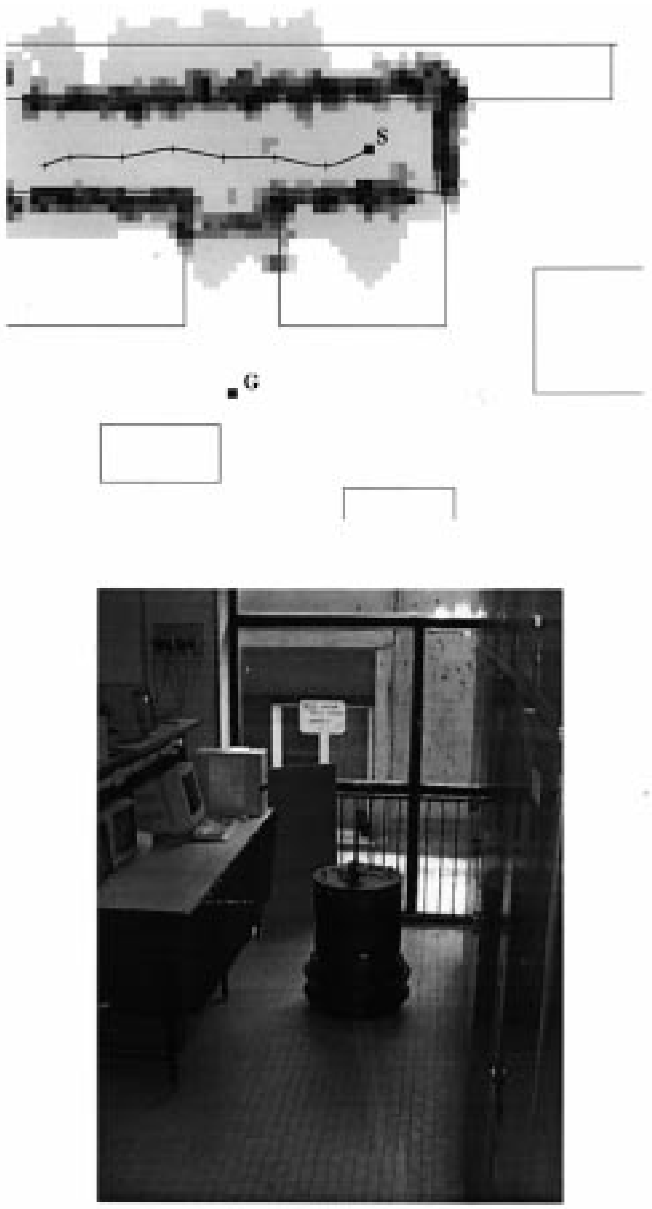


Fig. 18. Experiment 2: NOMAD tries to find an alternative path.

VI. Conclusion

A new method for real-time map building and navigation in unknown environments has been presented. Its basic features are as follows.

- Two fundamental processes are alternately executed: map building and navigation. In the former, the robot collects local information about the surrounding scene through its sensors, and updates accordingly the global representation so far reconstructed. In the latter, a suitable planning algorithm proposes a subpath P^k to the goal that avoids collisions in the explored region and indicates new areas to be visited. The subpath is followed by the robot up to

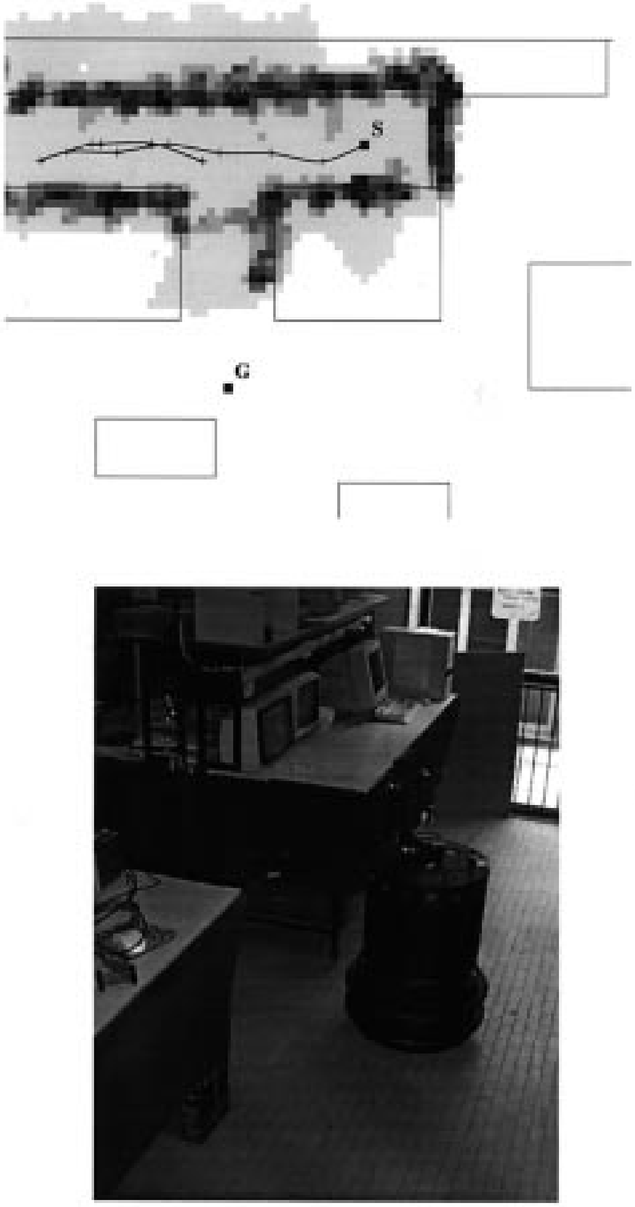


Fig. 19. Experiment 2: The channel is now free.

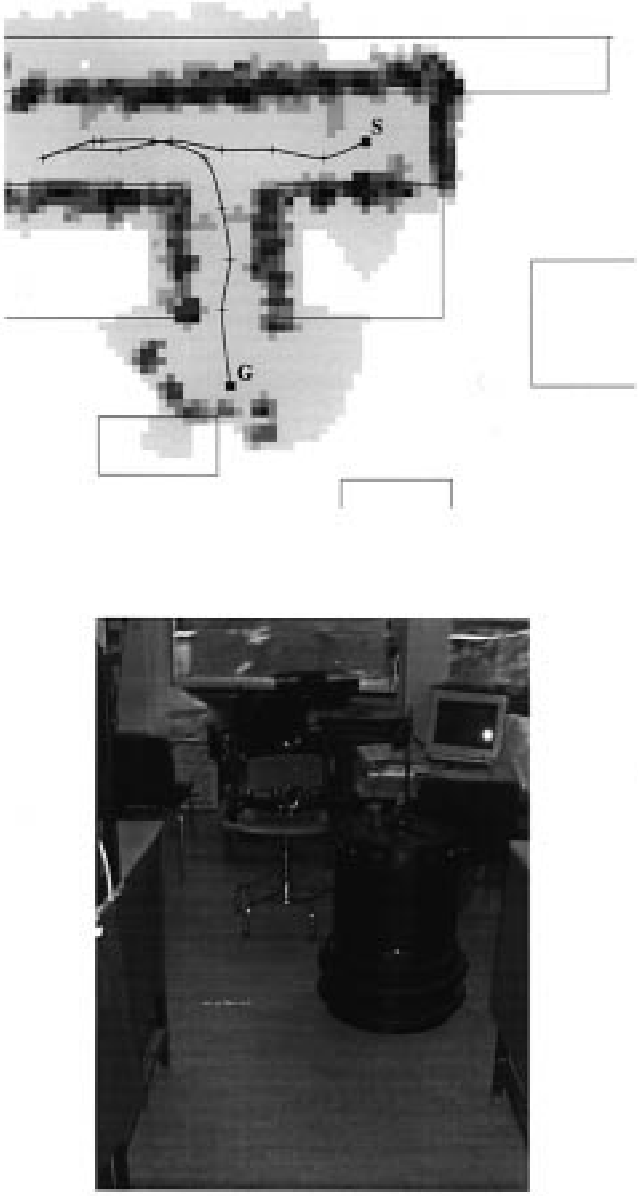


Fig. 20. Experiment 2: The final map and the path executed by NOMAD to reach the goal.

the boundary of the explored region, where a new map building process takes place.

- An intrinsically uncertain representation of the environment is used. In particular, fuzzy set operators are used to process ultrasonic sensor measures, updating the two gray-level bitmaps $\mathcal{M}_p$ and $\mathcal{M}_m$ that convey different risk information.
- Three cost functions g_1, g_2 and g_3 allow to quantify the risk of collision on $\mathcal{M}_p$ along the subpath P^k. All these may be interpreted as energy measures of $\mathcal{M}_p$ on P^k, therefore characterizing the distance of P^k from an ideally safe path.
- In the planning phase, $\mathcal{M}_p$ is searched for optimal paths. Proper instances of the $A^\star$ search algorithm have been identified for the minimization of g_1, g_2, and g_3. The three resulting algorithms correspond to different planning strategies.
- In the motion phase, $\mathcal{M}_m$ is used to identify unexplored areas, where a new perception step should be performed.

Experimental results have been reported to illustrate the satisfactory performance of the proposed technique, also in the case of moderately dynamic scenes. The obtained maps were quite accurate even in hostile environments, where false reflections often occur; in general, the paths followed by the robot are safe and effective. Space for adaptation is provided by the possibility of tuning various parameters as well by the choice of the path cost function.

A nice feature of the presented method is the homogeneity of its various components. Other modules (e.g., obstacle avoidance, localization) may be included in the overall control architecture of the robot in order to improve its performance. Among the current research directions, we mention i) the integration of information coming from other sensors (e.g., laser and infrared range finders) in the map building phase, that is straightforward within the fuzzy logic framework; ii) a localization procedure [31] with respect to the reconstructed map $\mathcal{M}_m$ to correct possible odometry errors; (iii) the use of a different planning methods. In [32] it was shown how *navigation functions* can be built on fuzzy maps in order to perform potential-based motion planning, extending the method of [33].

APPENDIX
A BRUSH-UP OF FUZZY SET THEORY

The very basic concepts of fuzzy set theory will be reviewed here. The interested reader may refer, for example, to [19], [34], [35].

Fuzzy sets may be easily introduced as extensions of standard crisp sets. For a *crisp set* A defined over the universal set U, the *membership function*

$$\mu_A : U \mapsto \{0,1\}, \quad \mu_A(x) = \begin{cases} 1, & \text{if } x \in A \\ 0, & \text{if } x \notin A \end{cases}$$

identifies those elements of U that belong to A.

For a *fuzzy set* $\mathcal{A}$ defined over U, the membership function

$$\mu_{\mathcal{A}} : U \mapsto [0,1]$$

may assume any real value within the interval $[0,1]$, expressing the degree of membership of any element of U to $\mathcal{A}$. Such a formulation may be adopted to represent two different kinds of uncertainty, namely i) *vagueness*, associated with the difficulty of using a crisp set to characterize a particular concept or property, or ii) *lack of evidence*, that does not allow to decide whether a given element belongs to a particular crisp set.

Basic fuzzy set operators (i.e., complementation, intersection, and union) can be defined as generalizations of the classical crisp set operators. In particular, it is necessary to satisfy proper sets of axioms [34] that, however, do not uniquely define the operators. As a consequence, several options for the same operation are available. This contributes to the richness and flexibility of fuzzy logic; on the other hand, the selection of the most suitable operators requires special care.

Any monotonic function

$$c : [0,1] \mapsto [0,1]$$

that satisfies the boundary conditions $c(0) = 1$ and $c(1) = 0$ may be used to define a *complementation* operator. In general, c is also required to be continuous. The most common complementation operator is obtained by letting

$$c(\mu_{\mathcal{A}}(x)) = 1 - \mu_{\mathcal{A}}(x). \tag{16}$$

Similarly, set *intersection* operators are defined through functions

$$i : [0,1] \times [0,1] \mapsto [0,1]$$

that are commutative, associative, and monotonic. Furthermore, i must satisfy the four boundary conditions $i(0,0) = i(1,0) = i(0,1) = 0$ and $i(1,1) = 1$. It is often requested that i is continuous, and sometimes that it is *idempotent*, i.e., $i(b,b) = b$. Standard intersection operators are

$$i_1(\mu_{\mathcal{A}}(x), \mu_{\mathcal{B}}(x)) = \min(\mu_{\mathcal{A}}(x), \mu_{\mathcal{B}}(x))$$

the *algebraic product*

$$i_2(\mu_{\mathcal{A}}(x), \mu_{\mathcal{B}}(x)) = \mu_{\mathcal{A}}(x) \cdot \mu_{\mathcal{B}}(x),$$

and the *bounded product*

$$i_3(\mu_{\mathcal{A}}(x), \mu_{\mathcal{B}}(x)) = \max(0, \mu_{\mathcal{A}}(x) + \mu_{\mathcal{B}}(x) - 1). \tag{17}$$

Set *union* operators are obtained from functions

$$u : [0,1] \times [0,1] \mapsto [0,1]$$

that are commutative, associative, monotonic, and satisfy the boundary conditions $u(0,0) = 0$ and $u(1,0) = u(0,1) = u(1,1) = 1$. Again, one may require u to be continuous and idempotent. Typical choices include

$$u_1(\mu_{\mathcal{A}}(x), \mu_{\mathcal{B}}(x)) = \max(\mu_{\mathcal{A}}(x), \mu_{\mathcal{B}}(x))$$

the *algebraic sum*

$$u_2(\mu_{\mathcal{A}}(x), \mu_{\mathcal{B}}(x)) = \mu_{\mathcal{A}}(x) + \mu_{\mathcal{B}}(x) - \mu_{\mathcal{A}}(x) \cdot \mu_{\mathcal{B}}(x)$$

and the *bounded sum*

$$u_3(\mu_{\mathcal{A}}(x), \mu_{\mathcal{B}}(x)) = \min(1, \mu_{\mathcal{A}}(x) + \mu_{\mathcal{B}}(x)).$$

Note that the triples (c, i_1, u_1), (c, i_2, u_2), and (c, i_3, u_3) satisfy De Morgan's law.

Another class of union operators has been introduced by Dombi [36]

$$\begin{aligned} &u_\lambda(\mu_{\mathcal{A}}(x), \mu_{\mathcal{B}}(x)) \\ &\quad = \frac{1}{1 + \left[\left(\frac{1}{\mu_{\mathcal{A}}(x)} - 1 \right)^{-\lambda} + \left(\frac{1}{\mu_{\mathcal{B}}(x)} - 1 \right)^{-\lambda} \right]^{-\frac{1}{\lambda}}} \end{aligned} \tag{18}$$

with $\lambda \in (0, \infty)$. One has

$$\lambda_1 < \lambda_2 \Longrightarrow u_{\lambda_1}(\mu_{\mathcal{A}}(x), \mu_{\mathcal{B}}(x)) > u_{\lambda_2}(\mu_{\mathcal{A}}(x), \mu_{\mathcal{B}}(x)).$$

This means that the Dombi operator produces "larger" union sets as λ is decreased. Equivalently, one may say that "weaker" unions are obtained for smaller values of λ. This behavior, illustrated in Fig. 21, implies also that the Dombi operator is not idempotent.

The classical fuzzy set operators of union and intersection so far described can be regarded as subsets of the more general class of *aggregation* operators. These are defined by functions

$$a : [0,1]^n \mapsto [0,1], \quad n \geq 2$$

that are monotonic, non decreasing and satisfy the boundary conditions $a(0, \ldots, 0) = 0$ and $a(1, \ldots, 1) = 1$. Fuzzy union and intersection belong to this class because of their associative property, but they do not exhaust the set of

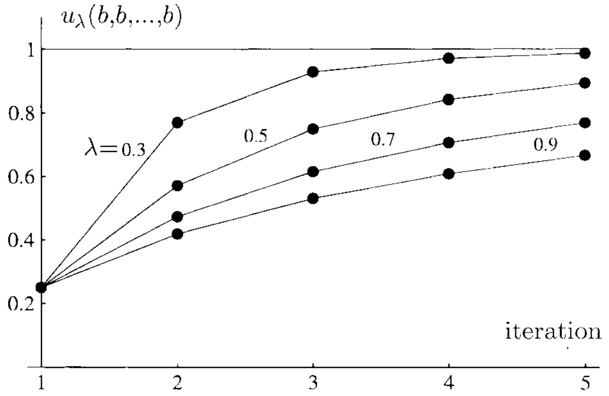


Fig. 21. Behavior of the Dombi union operator for different values of λ. The dots show the result of the iterated application of u_λ to the value $b = 0.25$.

all possible aggregation operators. In fact, there exists a third subclass of so-called *averaging* operators that produce intermediate results. In particular, aggregate fuzzy sets have membership function between the minimum and the maximum membership function of the original sets. A typical instance in this class is the *weighted arithmetic mean*

$$a(\mu_{\mathcal{A}_1}, \ldots, \mu_{\mathcal{A}_n}) = \sum_{i=1}^{n} \alpha_i \mu_{\mathcal{A}_i} \qquad (19)$$

where $\alpha_i \leq 1$, $i = 1, \ldots, n$, are positive real numbers.

References

[1] H. F. Durrant-Whyte, *Integration, Coordination and Control of Multi-Sensor Robot Systems*. Norwell, MA: Kluwer, 1988.

[2] *Autonomous Robot Vehicles*, I. J. Cox and G. T. Wilfong, Eds. New York: Springer-Verlag, 1990.

[3] J.-C. Latombe, *Robot Motion Planning*. Boston, MA: Kluwer, 1991.

[4] R. A. Brooks, "A robust layered control system for a mobile robot," *IEEE J. Robot. Automat.*, vol. 2, no. 1, pp. 14–23, 1986.

[5] O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots," *Int. J. Robot. Res.*, vol. 5, no. 1, pp. 90–98, 1986.

[6] R. J. Firby, "Adaptive execution in dynamic domains," Ph.D. dissertation, Dept. Comput. Sci., Yale Univ., New Haven, CT, 1989.

[7] D. W. Payton, "Internalized plans: A representation for action resources," *Robot. Auton. Syst.*, vol. 6, pp. 89–103, 1990.

[8] J. Borenstein and Y. Koren, "The vector field histogram—Fast obstacle avoidance for mobile robots," *IEEE Trans. Robot. Automat.*, vol. 7, pp. 278–288, June 1991.

[9] T. Skewis and V. Lumelsky, "Experiments with a mobile robot operating in a cluttered unknown environment," *J. Robot. Syst.*, vol. 11, pp. 281–300, 1994.

[10] A. Stentz, "The Navlab system for mobile robot navigation," Ph.D. dissertation, School of Comput. Sci., Carnegie Mellon Univ., Pittsburgh, PA, 1990.

[11] R. C. Arkin, "Integrating behavioral, perceptual, and world knowledge in reactive navigation," *Robot. Auton. Syst.*, vol. 6, pp. 105–122, 1990.

[12] E. Gat, "Integrating planning and reacting in a heterogeneous asynchronous architecture for mobile robot navigation," *SIGART Bull.*, vol. 2, pp. 70–74, 1991.

[13] A. Saffiotti, "Some notes on the integration of planning and reactivity in autonomous mobile robots," in *Proc. AAAI Spring Symp. Foundations Automatic Planning*, Stanford, CA, 1993, pp. 122–126.

[14] J. J. Leonard and H. F. Durrant-Whyte, *Directed Sonar Sensing for Mobile Robot Navigation*. Norwell, MA: Kluwer, 1992.

[15] A. Elfes and H. P. Moravec, "High resolution maps from wide angle sonar," in *Proc. 1985 IEEE Int. Conf. Robotics Automation*, St. Louis, MO, 1985, pp. 116–121.

[16] A. Elfes, "Occupancy grids: A stochastic spatial representation for active robot perception," in *Autonomous Mobile Robots: Perception, Mapping, and Navigation*, S. S. Iyengar and A. Elfes, Eds. Los Alamitos, CA: IEEE Comput. Soc. Press, 1991, pp. 60–71.

[17] D. W. Cho, "Certainty grid representation for robot navigation by a Bayesian method," *Robotica*, vol. 8, pp. 159–165, 1990.

[18] M. Weigl, B. Siemiatkowska, K. A. Sikorski, and A. Borkowski, "Grid-based mapping for autonomous mobile robot," *Robot. Auton. Syst.*, vol. 11, pp. 13–21, 1993.

[19] H. Bandemer and W. Näther, *Fuzzy Data Analysis*. Dordrecht, The Netherlands: Kluwer, 1992.

[20] L. A. Zadeh, "Outline of a new approach to the analysis of complex systems and decision process," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-3, no. 1, pp. 28–44, 1973.

[21] Polaroid Corporation, *Ultrasonic Ranging System*, 1987.

[22] I. Bloch, "Information combination operators for data fusion: A comparative review with classification," *IEEE Trans. Syst., Man, Cybern.*, vol. 26, pp. 52–67, Jan. 1996.

[23] F. Gambino, G. Oriolo, and G. Ulivi, "A comparison of three uncertainty calculus techniques for ultrasonic map building," in *Proc. 1996 SPIE Int. Symp. Aerospace/Defense Sensing Control*, Orlando, FL, 1996, pp. 249–260.

[24] J. R. Quinlan, "Consistency and plausible reasoning," in *Proc. 8th Int. Joint Conf. Artificial Intelligence*, Karlsruhe, Germany, 1983, pp. 137–144.

[25] S. J. Henkind and M. C. Harrison, "An analysis of four uncertainty calculi," *IEEE Trans. Syst., Man, Cybern.*, vol. 18, no. 5, pp. 700–714, 1988.

[26] P. P. Bonissone, "Reasoning, plausible," in *The Encyclopedia of Artificial Intelligence*, 2nd ed., S. Shapiro, Ed. New York: Wiley, 1991, pp. 1307–1322.

[27] A. De Luca and S. Termini, "Entropy and energy measures of a fuzzy set," in *Advances in Fuzzy Sets and Applications*, M. M. Gupta, R. K. Ragade, and R. R. Yager, Eds. Amsterdam, The Netherlands: North-Holland, 1979, pp. 321–338.

[28] P. E. Hart, N. J. Nilsson, and B. Raphael, "A formal basis for the heuristic determination of minimum cost paths," *IEEE Trans. Syst., Sci., Cybern.*, vol. SSC-4, no. 2, pp. 100–107, 1968.

[29] C. W. Warren, "Fast path planning using modified $A^\star$ method," in *Proc. 1993 IEEE Int. Conf. Robotics Automation*, Atlanta, GA, 1993, pp. 662–667.

[30] G. Oriolo, G. Ulivi, and M. Vendittelli, "On-line map building and navigation for autonomous mobile robots," in *Proc. 1995 IEEE Int. Conf. Robotics Automation*, Nagoya, Japan, 1995, pp. 2900–2906.

[31] G. Fortarezza, G. Oriolo, G. Ulivi, and M. Vendittelli, "A mobile robot localization method for incremental map building and navigation," in *Proc. 3rd Int. Symp. Intelligent Robotic Systems*, Pisa, Italy, 1995, pp. 57–65.

[32] G. Oriolo, M. Vendittelli, and G. Ulivi, "Path planning for mobile robots via skeletons on fuzzy maps," *Intell. Automat. Soft Comput.*, vol. 2, no. 4, pp. 355–374, 1996.

[33] J. Barraquand, B. Langlois, and J.-C. Latombe, "Numerical potential fields techniques for robot path planning," Rep. STAN-CS-89-1285, Dept. Comput. Sci., Stanford Univ., Stanford, CA, 1989.

[34] G. J. Klir and T. A. Folger, *Fuzzy Sets, Uncertainty, and Information*. Englewood Cliffs, NJ: Prentice-Hall, 1988.

[35] H.-J. Zimmermann, *Fuzzy Set Theory—And Its Applications*, Norwell, MA: Kluwer, 1991.

[36] J. Dombi, "A general class of fuzzy operators, the De Morgan class of fuzzy operators and fuzziness measures induced by fuzzy operators," *Fuzzy Sets Syst.*, vol. 8, pp. 149–163, 1982.

Giuseppe Oriolo (S'89–M'93) was born in Taranto, Italy, in 1962. He received the Laurea degree in electrical engineering in 1987 and the Ph.D. degree in system engineering in 1992, both from the University of Rome "La Sapienza," Rome, Italy.

From September 1990 to April 1991, he was a Visiting Scholar at the Center for Robotic Systems and Manufacturing, University of California, Santa Barbara. From 1992 to 1993, he taught linear system theory at the University of Siena, Siena, Italy. Since 1994, he has been a Research Associate with the Department of Computer and System Science, University of Rome "La Sapienza." His research interests focus on planning and control for robotic systems.

Giovanni Ulivi (M'84) was born in Rome, Italy, in 1950. He received the Laurea degree in electrical engineering in 1974 from the University of Rome "La Sapienza".

From 1974 to 1992, he was with the Department of Computer and System Science, University of Rome. Since 1992, he has been with the 3rd University of Rome, where he currently teaches automatic control in the Department of Computer Science and Industrial Automation. He is Tutor for several Ph.D. students. His research interests, begun with the control of electric motors, include nonlinear control of rigid and elastic robots, autonomous vehicle control, and scene recognition for motion planning. He is author of more than 80 scientific papers.

Mr. Ulivi is a member of the IFAC Technical Committee on Robotics (RobTeC) and has been member of program and organizing committees of several international conferences.

Marilena Vendittelli was born in Campobasso, Italy, in 1965. She received the Laurea degree in electrical engineering in 1992 from the University of Rome "La Sapienza," Rome, Italy, where she is currently pursuing the Ph.D. degree in system engineering in the Department of Computer and System Science.

From April 1995 to January 1996, she was a Visiting Scholar at LAAS-CNRS in Toulouse, France. Her main research interests are motion planning and control of mobile robots.