Multi-strategic approach for robot path planning in an unknown environment

Giovanni Bianco° , Riccardo Cassinis°°

Abstract

The paper presents a proposal for an autonomous robot path planning system that uses several strategies to reach a target in an a-priori unknown environment.

The proposed method has learning capabilities that allow the robot to take advantage of previous experience, thus improving its performance in further travelling in the same environment.

1.- Introduction

Navigation in a-priori unknown environments has a wide spectrum of applications in advanced robotics. Traditionally, this problem has been addressed either by having the robot build a map of the environment (at least of what can be seen from its starting position) before planning the actual movement, or by applying some deterministic algorithms that are able to cope with unknown environments.

If a new movement has then to be planned in the same environment, the first method has the drawback that it only works well if no changes to the environment have occurred since when the map was built, while the second can take no advantage of previous experience, since it has no long-term memory.

The idea presented in this paper is that joining the two methods can help combining their advantages and eliminating many of the drawbacks. In other words, it seems useful to memorise some characteristics of the working area, but also to use algorithms that may withstand changes in the environment.

° Centro di Informatica e di Calcolo Automatico - Università di Verona

°° Dipartimento di Elettronica per l'Automazione - Università di Brescia

The final goal is to obtain a method that allows the robot to remember, for a given environment, which trajectory planning strategies were the best for navigating in any of its parts, so that, as time passes, its performance increases also if moderate changes to the environment take place.

2.- The state of the art

Current research on mobile robot path planning refers to two main models, based on two different hypotheses about planning information availability.

The first model, not considered in this paper, concerns path planning with complete information, as described in [1]. There are several proposals based on this model: the best known strategies are the Lozano-Péres ones [2] and the Brooks ones [3].

The second model concerns path planning with initial incomplete information. The lacking information is generally acquired using sensors, and is then included in some environmental representation.

As it is not possible to consider the whole environment all at once, in this model the path planning phase is time-distributed [4]. Significant strategies based on this model are Lumelsky's BUG1 and BUG2 [1].

The proposed path planning strategies for an a-priori unknown environment generally define planning algorithms, whose optimality and convergence are not always assured. Researchers have therefore introduced various methods for improving the strategies' convergence and optimality.

In [5] a strategy (BUG3) is described, that assures the robot's convergence to the target position by first applying BUG2 (efficient but subject to failures); if BUG2 fails, it applies BUG1, less efficient than BUG2 but always converging to the target position.

Other researchers [6] improve the path planner performance by using perceptive learning: re-planned trajectories consider previous information about the successful paths carried out in the past.

3.- The proposed multi-strategic solution

The proposed multi-strategic solution assumes that local environmental characteristics affect path planning strategie's optimality, efficiency and convergence. We will show some path planning strategies' examples to clarify this assertion.

The main reason that justifies the existence of BUG3 is the discovery of local path loops in BUG2 strategy, as stated before.

Failures (loops and, more generally, non-convergence) are not the only characteristic that makes a given strategy unusable in some environments. There are in fact many restrictions due to the ways the robot senses the surrounding environment. Let us consider, for example, a strategy that uses sonar sensors: these sensors are well suited for wide environments, but become useless in narrow, complex passages.

To formalise our hypotheses, we define a set S whose elements are all the possible path planning strategies; we also define a set C whose elements represent, in ways we do not formalise, all the possible environmental characteristics. Finally, we introduce a function $\square$, whose values range from 0 to 1, that represents the strategies' efficiency computed over a specific path p. We assume that $\square(p)=0$ whenever the strategy applied in p fails (it doesn't converge toward the target position or it loops); with $\square(p)=1$ the strategy reaches the target in the optimal way (following the minimum path at the greatest possible speed). In other cases ($0<\square(p)<1$) the strategy always reaches the target but not in an optimal way.

If $s_n(c)\ (s_n \in S, c \in C)$ is the effective planned path, our hypothesis is so formalised:

$$\forall s_n \in S \Rightarrow \exists \bar{c} \in C, s_m \in S: \square(s_m(\bar{c})) > \square(s_n(\bar{c})), m \neq n$$

that means that for each strategy there exists at least one environmental characteristic that makes another strategy more efficient than the former one. In other words, we assume that there is no absolutely optimal strategy.

The way used to divide the environment in local contexts is a set of square cells (A_i) whose dimensions are approximately the same as the robot's base size (Fig. 1).

The whole path, from the starting position to the final target, can be considered as an ordered sequence of not necessarily neighbouring cells

$$A_s \to A_1 \to \ldots \to A_n \to A_t$$

Therefore it consists of a sequence of elementary paths

$$A_i \to A_j$$

Each elementary path can be planned using a different strategy.

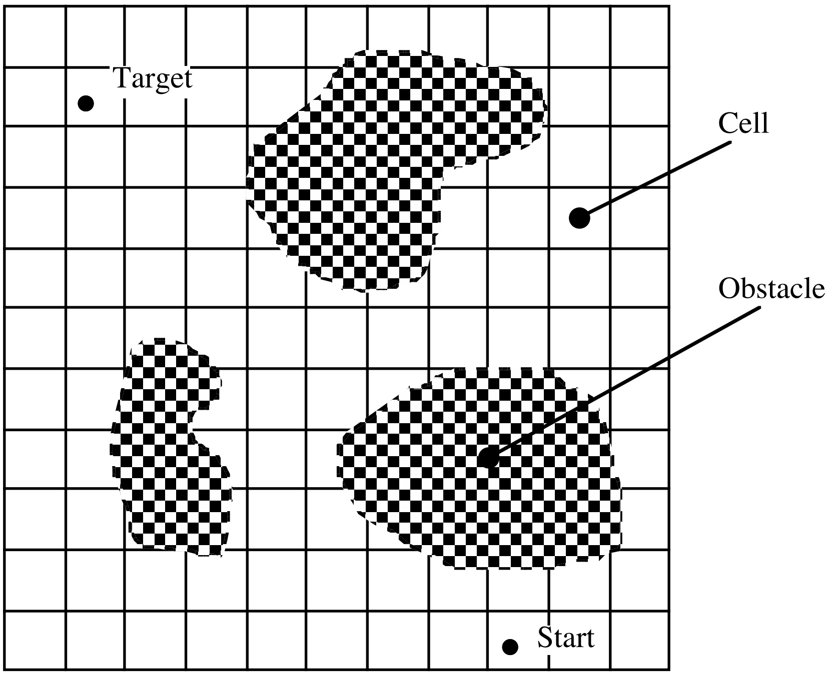

Fig. 1. - Partitioning the environment in cells.

The choice of a particular strategy for planning each elementary path is left to the robot, that chooses the most efficient one according to some information.

The parameters we associate to each strategy, that enable us to compare them with each other, are: applicability in a given local environment, presumable efficiency and actual efficiency in following a given path.

Applicability criteria for a given strategy are very difficult to define. On the other hand, it is quite easy to state that, for any strategy s_n there is at least one environmental characteristic c_{ij} whose existence guarantees that strategy s_n will certainly fail, i.e.[1]

$$\square(s_n(c_{ij})) = 0$$

where c_{ij} refers to the environmental characteristics that belong to the effective path, planned by strategy s_n, that starts from cell A_i and ends in cell A_j.

We will now make some considerations about the robot's presumable and actual efficiencies to understand the efficiency function $\square$ introduced above.

---

[1]For example, it is obvious that in a dark environment all vision-dependant navigation algorithms will fail.

The presumable efficiency is defined as the ratio between the minimum navigation time ($t_{\square}$) and the estimated navigation time ($t_{\square}$). So, for any path $A_i \square A_j$ a minimum time $t_{\square_{ij}}$ and an estimated time $t_{\square_{ij}}$ will be identified together with the ratio

$$\frac{t_{\square_{ij}}}{t_{\square_{ij}}}$$

The minimum navigation time is the time the robot will need to follow a straight path from the starting position (cell A_i) to the target (cell A_j) travelling at maximum speed, whereas the estimated time represents the time the robot will presumably spend to reach the target using the given strategy.

The actual efficiency is defined as the ratio

$$\frac{t_{\square_{ij}}}{t_{\square_{ij}}}$$

where $t_{\square_{ij}}$ is the actual navigation time, i.e. the time that was spent to follow the path. Clearly, the estimated time can be computed before path planning step while the actual time is only known after the robot has reached the target position.

The robot, using the information just described, can make a choice among the strategies it knows, and it will choose the strategy that seems the most efficient for going from cell A_i to cell A_j. So, for each known applicable strategy ($\square(s_n(c_{ij})) \neq 0$) it will choose the one that has the highest actual efficiency or, if the latter is missing, the one that has the highest presumable efficiency.

As it was said before, the robot makes its choice according to the information included in an environmental representation that, in our case, is not an analogue representation of the world.

Our environmental representation is a square matrix whose rows and columns are cells' identifiers, respectively starting cells and final position cells; each matrix element is linked with a table that contains a list of all the known strategies, together with specific information, as described in the sequel (see figure 2).

As stated, the robot needs three pieces of information from the surrounding environment, that are computed by three virtual sensors:

- **VSA (Virtual Sensor of Applicability)**: provides a binary response about the applicability of a given strategy in the local environment context.
- **VSPE (Virtual sensor of Presumable Efficiency)**: provides a numeric measure ranging from 0 to 1 given by the ratio

$$\frac{t_{\square}}{t_{\square}}$$

- **VSAE (Virtual Sensor of Actual Efficiency)**: Like VSPE, it provides a numeric output given by the ratio

$$\frac{t_{□}}{t_{□}}$$

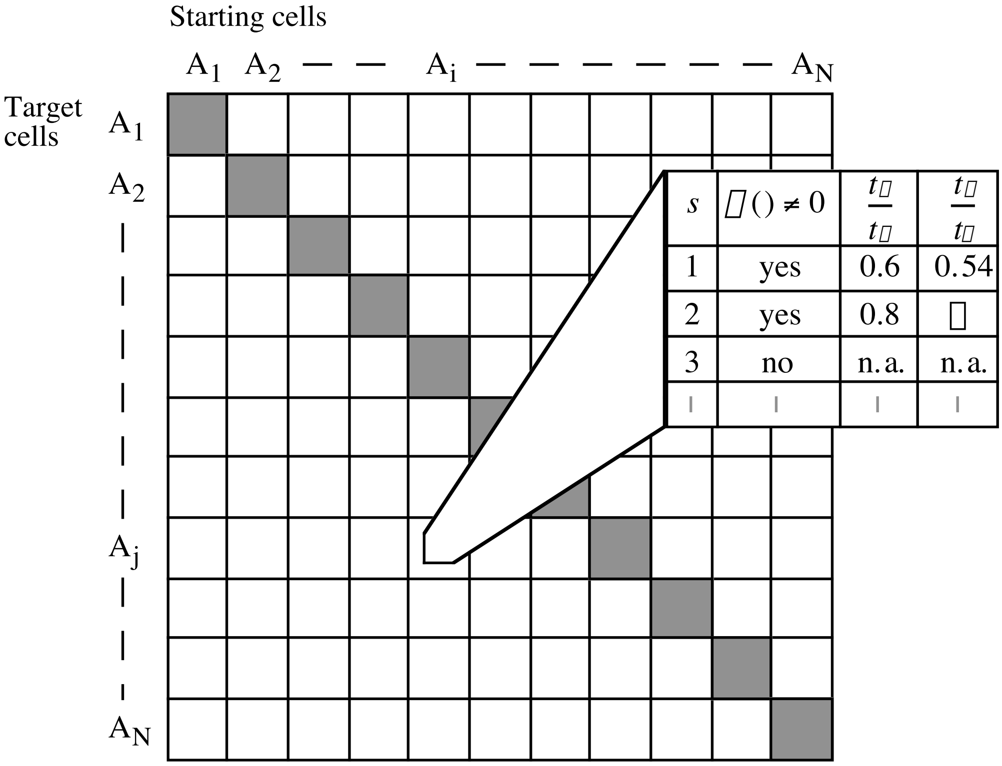

Fig. 2. - The matrix for representing paths

The above mentioned virtual sensors can be implemented using general knowledge about the world and/or by using active sensing (in [7] we can find a description where active sensing is incorporated in path planning methods). Active sensing and knowledge about the world are used to find out the environmental characteristics which may negatively affect strategy applicability or presumable efficiency.

In the prototype we compute $t_{□}$ mainly relating it to the most time-consuming robot's sensing activities. This can be done either formally (computing the time needed for each sensing activity in a given environmental situation) or statistically, using data from previous applications of the same strategy.

VSA is instead a simple set of rules, deriving from the experience of human operators.

Multi-strategic path planning can be achieved in three different ways, according to the philosophy the robot uses to collect sensory information about the local environment. These different approaches imply different path generation methods and therefore the planned paths are different in terms of complexity and efficiency.

The three possible approaches are:

- **Minimum Approach**: the robot plans the whole path using only one strategy and uses this strategy until it fails. Subsequent re-planning may use other strategies. Sensoriality is used to find out strategy failure situations.
- **Medium Approach**: the robot changes strategy as soon as it discovers certain local environment characteristics that make the strategy in use inapplicable. Sensoriality is used to discover these local environment conditions.
- **Full Approach**: the robot changes strategy whenever it finds, in any local environment, that another strategy has a better presumable efficiency than the one currently in use. Sensoriality is heavily used to compute, in each cell, the efficiency of each strategy known by the robot.

The chosen approach in the prototype is the minimum one.

It is obvious that, once data are stored in any matrix element, they can be used each time the robot traverses that particular element. It is not necessary to re-compute all data, and "experience" about strategies' failures and successes can be retained and efficiently re-used during the solution of further navigation problems in the same area, provided that environmental changes are limited, as will be said in the sequel.

In the following figures we consider three examples about multi-strategic approaches. In figure 3 a strategy s_1 is used until it fails (in this case it falls into an endless loop); then the robot uses another strategy (s_2).

In figure 4 a strategy s_1 is used until it fails (in this case it is not further applicable); then the robot uses another strategy (s_2).

Last, in figure 5 the strategy s_1 is used until another strategy becomes more efficient than the one in use; in this case the robot uses the most efficient strategy (first s_2 and then s_3).

4.- Issues on the multi-strategic proposed solution

The multi-strategic approach is not a path planning strategy: it is rather an improved path planning method. In fact, by using existing strategies we can sometimes improve the whole path efficiency or eliminate problems due to the non-convergence of a particular path planning strategy in certain environmental conditions.

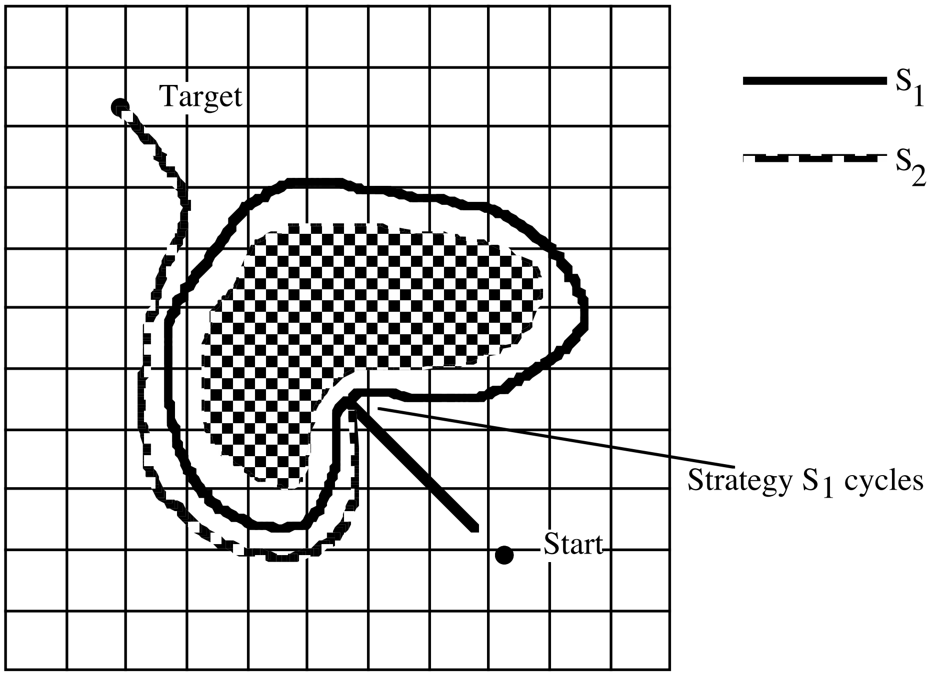

Fig. 3. - Changing a strategy when a loop is discovered.

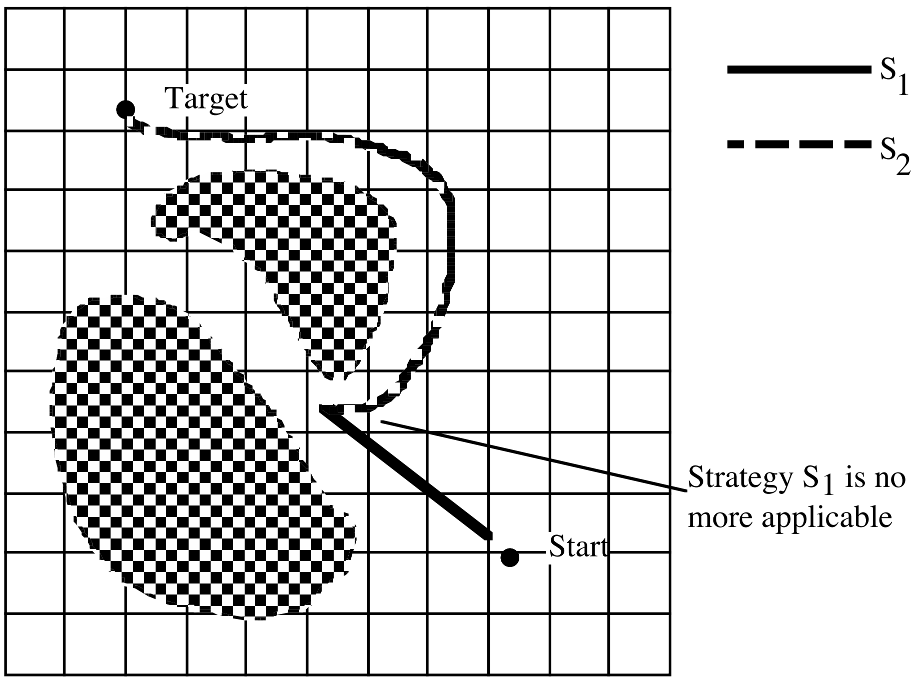

Fig. 4 - Changing strategy when it fails

Referring to the above considerations, we must compare the multi-strategic approach with other improvement methods and not with particular path planning strategies. Therefore, the main difference between the multi-strategic approach and the classic ones is that for each cell we define the *strategy*, and not the *path*, needed to reach the target. This is the same difference we can find between Perceptive Learning (the classic approach) and Behavioural Learning (our approach). Behavioural learning allows the robot to consider qualitative rather than quantitative information about the surrounding environment; therefore obstacle's movements and world's modifications, generally, affect the multi-strategic approach less than the classic ones.

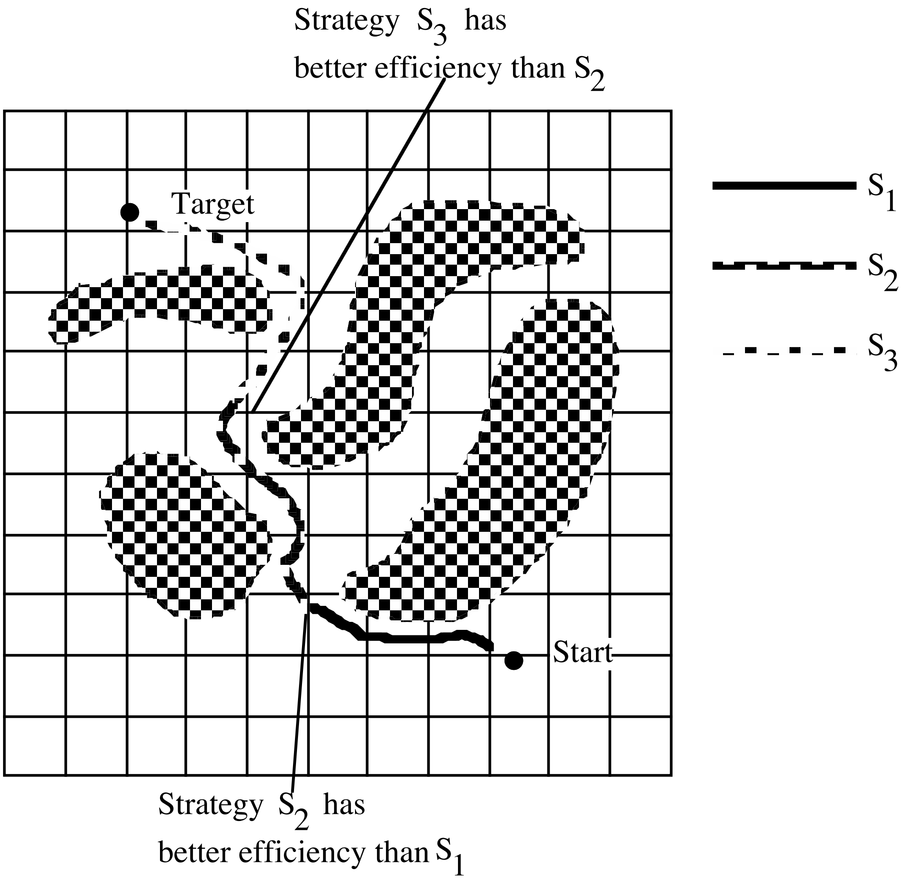

Fig. 5. - Using the best strategy.

The multi-strategic approach is specially advantageous in case of small environment's modifications; to explain this, let us consider the unknown world the robot is moving in. There are two possible kinds of unknown environments: non changing (static) environment and changing (dynamic) environment.

To formalise these environmental classes we can define N as the number of cells into which an environment E is partitioned, and $C_{E_t} \subseteq C$ the environmental characteristics considered at time t. If t_1 and t_2 (with $t_2 > t_1$) represent the times at which the robot has to plan a path and M represents the number of cells whose $C_{E_{t_2}}$ have changed with respect to the $C_{E_{t_1}}$, we can define:

- Static environment: $M = 0$.
- Fully dynamic environment: $M = N$.
- Partially dynamic environment: $0 < N < M$.

In this way we have established a link between the environmental changes and the used environmental representation[1]. As the multi-strategic approach stores needed information in the cell, we can argue the multi-strategic behaviour in different kinds of environments:

- Static environment: in this environment it is useless trying to improve the planning performance using the multi-strategic approach; other methods based on geometric mapping of the environment can give much better results.
- Fully dynamic environment: there is no advantage in storing information about the environment, since, each time the robot traverses it, it is completely different from the previous time. For this class of environments the multi-strategic approach is a good method to face the uncertainty of the world, but learning capabilities of the system are completely defeated.
- Partially dynamic environment: some parts of environment do not change, and previously acquired experience can be useful for further planning of paths.

5.- Effects of the multi-strategic approach

The use of the multi-strategic approach allows a dynamic path planning, that can be very useful in partially changing environments.

With any one of the three previously described approaches, for each cell the set of the presumed efficiencies e constitutes a distribution

$$PE = \left(e_1, e_2, \ldots, e_n\right)$$

that can be transformed into an empirical statistic distribution with the following operation:

$$D = \left(e_1 / V, e_2 / V, \ldots, e_n / V\right)$$

---

[1]For the sake of simplicity, we do not take into account changes that may occur *during* the navigation. It shoud however be kept in mind that several navigation algorithms are able to cope with moving obstacles.

where V represents the number of times the robot has travelled across the cell being considered.

If the strategy s_i is chosen according to an integer random number $i \in \{1, 2, \ldots, n\}$ generated using D, the robot will sometimes choose a strategy which is not the apparently optimal one. This may be of great help in the first plannings in a given environment, when the virtual sensors have not been properly trained yet, and the information they provide is quite uncertain.

6.- Conclusions

We have shown a method for planning robot navigation in a-priori unknown environments which is capable of:

- Selecting the apparently best suited algorithm among a library of navigation strategies;
- Learning about failures or successes of previously used strategies;
- Randomly choosing alternative strategies which, in many situations, can lead to better results than the apparently best ones.

The major drawback of the described methods is that it needs some means for self-localising, since the position of the robot with respect to the first starting position in the environment has to be known. Since the planning method is non geometric, however, coarse approximations in the localisation procedure can be allowed.

The proposed method is currently being implemented; first results should be available by fall 1993.

Bibliography

[1] V. Lumelsky, *Algorithmic Issues of Sensor-Based robot Motion Planning*, 26th IEEE Conf. on Decision and Control in Los Angeles, December 1987.

[2] T. Lozano-Péres, M. Wesley, *An Algorithm for Planning Collision Free Path Among Polyhedral Obstacles*, Communication of ACM, vol. 22, NO. 10, October 1979.

[3] V. Lumelsky, *On the Connection Between Maze-Searching and robot Planning Algorithms*, Proc. of the 28th IEEE Conf. on Decision and Control, vol. 2, December 1989.

[4] V. Lumelsky, A.A. Stephanov, *Effect of Uncertainty on Continuous Path Planning for an Autonomous Vehicle*, 23th IEEE Conf. on Decision and Control in Los Angeles, vol. 3, December 1984.

[5] V. Lumelsky, A.A. Stephanov, *Dynamic Path Planning for a Mobile Automaton with Limited Information on the Environment*, IEEE Trans. on Automatic Control, vol. AC-31, NO. 11, November 1986.

[6] H. Spandl, *The HALMOR System - a Machine Learning Approach for an Autonomous, Mobile Robot,* Karlsruhe University, Internal Report, 1990.

[7] V. Lumelsky, T. Skewis, *Incorporating Range Sensing in the robot Navigation Function*, IEEE Trans. on System, Man and Cybernetics, vol. 20, September-October 1990.