Robots Find a Better Way: A Learning Method for Mobile Robot Navigation in Partially Unknown Environments

Kristo Heero[1] **Jan Willemson**[1] **Alvo Aabloo**[2] **Maarja Kruusmaa**[2]
kristo@math.ut.ee jan@math.ut.ee alvo@ut.ee maarjakr@ut.ee

[1]Dept. of Computer Science, Tartu University, Liivi 2, Tartu, Estonia
[2]Institute of Technology, Tartu University, Vanemuise 21, Tartu, Estonia

Abstract

This paper represents a method for mobile robot navigation in environments where obstacles are partially unknown. The method uses a path selection mechanism that creates innovative paths through the unknown environment and learns to use routes that are more reliable. This approach is implemented on Khepera robot and verified against shortest path following by wave transform algorithms. Based on the experimental data, we claim that robot's trajectory planned by wave transform algorithms is very difficult to predict and control unless the environment is completely modeled and the localization errors are small. We show that even small unmodeled obstacles can cause large deviation from the preplanned path. Our complementary approach of path selection decreases the risk of path following as well as increases the predictability of robot's behaviour.

1. Introduction

Mobile robots, operating in human inhabited environments are expected to navigate safely as well as optimize their energy consumption and travel time. Since real-world environments are complex, often unstructured and dynamic, it is impossible to build a complete model of robot's surrounding and keep it up to date. At the same time, we expect the robot to operate as efficiently as possible with a rather limited amount of information.

Until now, the research in mobile robot path planning has focused on finding optimal routes from start to goal. The optimality is usually measured in terms of traveled distances [4]. Other measures are also used, e.g. confidence value [3]. For planetary rovers the efficiency of path is often expressed in terms of slope or roughness of the surface [1][2].

To navigate in partially unknown environments, robots use local replanning. Local re-planners use only local information to negotiate unexpected obstacles and since they do not use global knowledge, the behaviour of the robot is not globally optimized. Salich and Moreno have referred to this problem as the dilemma of authority vs. freedom [5]. The dilemma rises from that classic planners produce rigid orders while the behaviour of local reactive planners is unpredictable. Some researchers try to overcome this problem by incorporating global information to local decision making [7].

Path planners used in robotics have been proven to give a globally optimal solution in globally known static environments. Their efficiency in complex, dynamic and partially unknown environments during long periods of time is not investigated. Our experimental data suggests, that the dilemma local vs. global decision making is not so important as it is anticipated e.g. in [6]. It rather appears that if the global planner does not have all the global information about the environment, it anyway fails to create globally optimal plans.

We conclude, based on our experimental data that the environment has a much more significant effect on the behaviour of the robot than the algorithm used. Even if the robot always replanns globally and always uses all the global knowledge available, it has a very little effect on the total outcome unless the environment is completely modeled. Our tests also show that robot's trajectory is very difficult to predict and control. Small unmodeled obstacles can considerably deviate the robot from its globally planned path.

A good characteristic of a learning system is the predictability of its behaviour. The better the system can predict the outcome of its future actions the better it has learned its environment. A mobile robot can predict its behaviour when it knows its position with a great certainty after a certain period of time. The ability to predict the future outcome makes it possible to optimize the robot's behaviour (e.g. travel time, energy consumption etc.).

The problem we try to solve is thus how to optimize the behaviour of the robot in partially an unknown environment during a long period of time. There are two complementary approaches to increasing the predictability of robot's behaviour. On the one hand, it is possible to gather more information about the environment to plan paths that can be more certainly followed. But since our experiments show that even small imprecision in data or noise can considerably affect the robots trajectory, we have chosen an opposite approach. Instead of trying to model the environment we look for trajectories in a partially unmodeled environment that can be followed with a great precision.

We propose a method of covering a rectangular grid-based map with sub-optimal paths. In [8] we have described the method in detail and proven that the number of possible trajectories grows linearly with a small constant when the size of the map is increased. Therefore the method can be used even in large-scale environments. The robot will then try to follow these paths and memorize them until it finds some that is sufficiently stable and easy to follow.

In our previous work, reported in [9] we have tested or approach in a totally unknown changing environment. The results show that the robot is able to adapt to the changes when the unknown obstacles are frequently replaced and learns to use trajectories that take it safer to the goal.

In this paper we report a series of test to investigate the robots behaviour in a partially known environments. The environment is static to show the cause-effect relationship between the model of the environment and the robots behaviour. It allows us to draw a conclusion that the behaviour is influenced by the environmental model and the path planning algorithm but not by the robot's ability (or inability) to adapt to the changes.

Our paths selection algorithm is verified against shortest path following by a wave transform algorithm of [10] with global replanning.

When planning the experiments, our hypothesis was that the efficiency of our method decreases when the environment is better known and when the unknown obstacles are smaller. We estimated that the shortest path following with global replanning would soon outperform our method. The tests did not confirm that hypothesis. On the contrary, the experimental data shows that wave transform algorithms are very sensitive to small imprecisions in an environmental model. Even small unknown obstacles (or possibly sensor noise) can cause large deviation from the originally planned path.

Our method of path selection has two limits. First, it assumes that the robot will repeatedly traverse between two entry points. This assumption makes it possible to try several alternative trajectories. Fortunately there are plenty of mobile robot implementations (e.g. transportation, surveillance, convoying) that presume repeated traversal between specified target points.

Second, the robot needs a fairly precise positioning system to follow the trajectories it has planned. In our tests we use an overhead camera to determine the robot's pose. We therefore suggest that the method works equally well with a satellite or pseudolite-based navigation. Since we test our approach in an environment where some static obstacles are modeled, it is principally possible to use these objects as landmarks. Yet we do not have any experience on how the robot would behave when the localization errors are large, like it often happens with landmark based navigation.

In the next section we state the problem and list the assumptions we have made. We then describe briefly our path selection mechanism. After that, we describe the experiments and draw conclusions based on the experimental data.

2. Problem statement

It is further assumed that:

1. The environment is dynamic and large. It is not possible or feasible to model it precisely or keep the model constantly updated.
2. The environment contains obstacles with unknown size and location. Traversing this environment implies risk of colliding with these obstacles, being delayed when maneuvering around them or ending up in a deadlock.
3. Sensorial capabilities of the robot are insufficient to distinguish between static, dynamic and semi-dynamic obstacles (e.g. between pillars and people, steady and replaced furniture).
4. Mapping, path planning and localization are not the main objectives of the robot. They are presumptions that make it possible to successfully complete a mission. Therefore they cannot take all of the time and computational recourses.
5. The robot is expected to fulfill its mission as fast and safely as possible.
6. The localization errors are small and do not accumulate and it is therefore possible to follow a preplanned path rather precisely.

The assumptions 1 and 3 seem to contradict with the experimental design where the environment is actually kept static. However, a static environment is not the necessary precondition of the approach. The environment is kept static only to find out the causal relation between an environmental model and the behaviour of the robot.

The problem we aim at solving is the following: find reliable paths between previously determined target points so that following them minimizes risk of collisions and speeds up the mission.

Our approach to the problem is based on the following observation. In a dynamic environment with an unknown obstacle distribution, the best path to the goal is not necessarily the shortest. Depending on the nature of the environment, there may exist routes that are longer but easier to follow. By introducing a path generation algorithm, the robot can test several alternatives to reach the goal. By remembering its path following experiences, it can learn to follow paths that save time and reduce risk. As the environment changes, the robot will reevaluate its past experience and adapts to use new easily traversable paths.

3. Path selection

Theoretically, the number of different paths on a grid-based map is overwhelming. There are too many alternatives to travel between two points and the robot could never try them all. In addition, most of those paths are unfeasibly long, crooked and difficult to follow. So the aim of the path selection algorithm is to:

- generate paths that are easy to follow if free from obstacles;
- generate paths that are as much different from each other as possible to let the robot find out as many innovative solutions as possible;
- provide a mechanism that in practice is able to discover virtually all possible alternatives;
- cover the whole space of innovative solutions with as few alternatives as possible in order to maintain the robot's ability to generalize and keep the memory constrained.

We propose a method that works by dividing the grid into paths segments and then generating paths that cover all these segments. Full description of the method and its formal analysis is presented in [8].

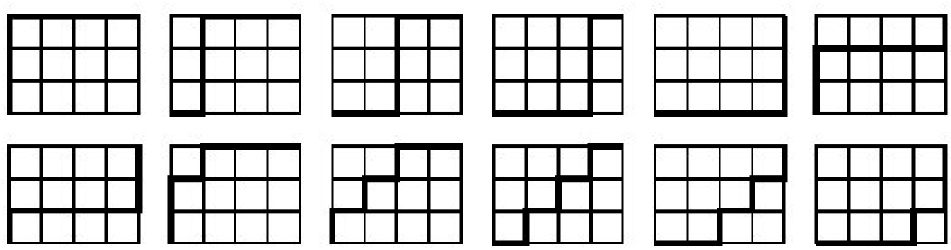

Figure 1: Cover of a 3 × 4 grid.

Figure 1 illustrates one possible cover of a 3 × 4 grid. The paths selected by the robot are limited to those not having back turns and covering all the grid segments of length 2. In practice, paths relaxation is used to smoothen the paths and the zig-zags will be straightened.

It is proven in [8] that for a grid of the size m × n, the cardinality of the minimal cover is 2m+2n-2 paths. It means that the number of different paths is very small and grows linearly with a small constant, which makes it well scalable for very large domains.

4. Experimental Design

The experiments are conducted using a mini-robot Khepera. It is a differential drive miniature circular robot (with radius 26 mm) equipped with IR sensors for collision avoidance and it can be connected to a PC over a serial link.

The localization system is presented in Figure 2. A video camera is mounted to the ceiling to recognize the position of the robot. The PC processes the camera image to find robot's position and a computer algorithm controls the robot over a serial link.

In this way the localization errors are rather small (usually comparable to the size of the robot).

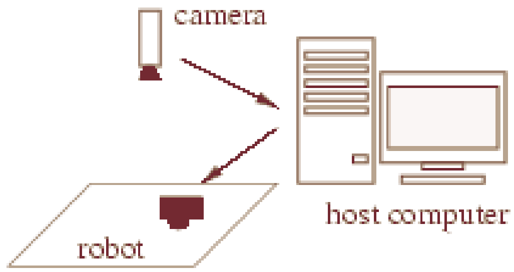

Figure 2. Localization system.

The size of our test environment is 1860 mm × 1390 mm. It is represented in Figure 3 to the left. The picture in the middle represents the same environment as shown from the overview camera. The picture to the right in Figure 3 is the graphical interface of the computer program that controls the robot and monitors its behavior.

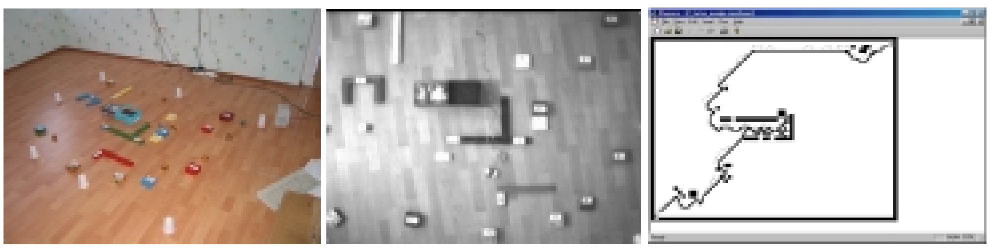

Figure 3. The test environment (to the left), the same environment seen through the overview camera (in the middle) and as modeled by the control program (to the right).

To test our path selection mechanism, the robot traverses repeatedly between the lower left corner and upper right corner of the environment in Figure 2. The physical environment for all test runs is the same. To determine how much the environmental model affects the results, we run the tests with 3 different maps represented in Figure 4.

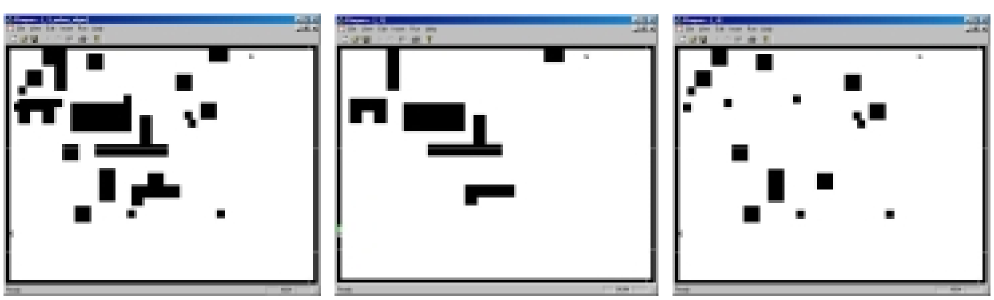

Figure 4. Environmental models used in experiments: a fully known environment (to the left), environment with large obstacles modeled (in the middle) and with small obstacles modeled (to the right).

The map to the left of Figure 4 is the precise model of the environment, containing the precise location of all obstacles. The map in the middle models only large obstacles

while the location of small obstacles in unknown. The map to the right models only small obstacles while the large obstacles are unknown.
We compare our path selection method to shortest path following by wave transform algorithm with global replanning. Table 1 shows the number of trials with every environmental model with both path planning algorithms, shortest path planning vs. path selection. The number of trials depends on how fast the process stabilizes.

Environmental model	Nr. of trials	
	Path selection	Shortest path
1.All obstacles known		10
2.Large obstacles known	20	50
3.Small obstacles known	20	50

Table 1. Number of trials

The efficiency of the path planning algorithm is characterized by four parameters: number of replannings, travel time, travel distance and the deviation from the originally pre-planned path.
One trial means planning a path from the lower left corner of the test environment to the upper right (or back again), following this path, replanning when an unknown obstacle is detected and recording the data when the robot reaches the goal.
The shortest path planning algorithm is the following:

1. Plan off-line a path from current start to current goal. This path is the shortest path to the goal calculated by a distance transform method [10].
2. Follow the path.
3. If an obstacle is detected plan a new path from its current position to the goal by a distance transform algorithm.
4. Repeat steps 2 and 3 until goal is reached.
5. Record travel time, travel distance, number of obstacles detected and deviation from the path planned at step 1.

The path selection algorithm is the following:

1. At the first trial select a sub-optimal path planned by the method described in Section 3.
2. Follow the path.
3. If an obstacle is detected plan a new path from its current position to the goal by a distance transform algorithm.
4. Repeat steps 2 and 3 until goal is reached.
5. Smoothen the actually followed path to remove cycles, zig-zags and gaps caused by localization errors.
6. Store the smoothened path together with the travel time, distance, number of replannings and deviation.
7. At next trial check if there is a stored path with acceptably low number of replannings. If yes, follow this path. If no, choose a new path by using a method described in Section 3.
8. Repeat steps 2 to 7.

5. Experimental Results

All data about the experiments, including recorded parameters at every trial, snapshots of every followed path and code of the control program are available at

http://math.ut.ee/~kristo/khepera/. We here represent only some general statistics to compare the path planning strategies described above.
Table 2 represents data on shortest path planning. Table 3 represents data on path planning with path selection. The efficiency of the path selection mechanism in case of a small number of trials depends largely on how fast the robot finds a sub-optimal path that is easy to follow. While running the test in the 3rd environment (with small obstacles known) the robot found an easy-to-follow sub-optimal path at the first trial. For the sake of an unbiased interpretation we also represent data of another experiment that shows the worst case we have encountered. The robot had to try 4 sub-optimal paths before it found one that was good enough. The last row of Table 3 therefore gives two figures for every parameter, the best result vs. the worst result.

Environmental model	Nr. of replannings	Travel time	Travel distance	Deviation from the preplanned path
1.All obstacles known	0.3	104	2555.0	43.8
2.Large obstacles known	12.7	123.3	2697.3	114.7
3.Small obstacles known	14.8	134.3	2768.0	107.0

Tabel 2. Results of shortest path planning

Environmental model	Nr. of replannings	Travel time	Travel distance	Deviation from the preplanned path
1.All obstacles known				
2.Large obstacles known	0	104.1	2584.6	29.2
3.Small obstacles known	0/5.7	129.8/123.5	2534.2/2805.5	29.2/145.0

Table 3. Results of planning with path selection

6. Discussion and conclusions

The first trials test the shortest path following strategy in a completely known environment (the first row in Table 2). This case is the ideal case where globally best paths are planned with all available information. A closer look to the statistical data (available at the website) shows that the behaviour of the robot is predictable and stable. It means that we are able to control the robot with the great precision. Localization errors, imprecision of mechanical linkages and sensor noise have no significant effect to the test results. Now, keeping all other things equal and changing only the environmental model or the path planning algorithm we can claim that the changes in experimental results are caused by one of the latter reasons.
Next we have verified the behaviour of the robot using two path planning strategies. Speaking in terms of decision-making theory, in case of shortest path planning, the robot can be described as a rational utility maximizing agent. It always tries to find the shortest path to the goal considering all information available. In the case of path selection, the robot can be described as an explorative agent. It randomly tries sub-optimal solutions to escape the local minimum and find a globally best solution.

The results show that by and all, the explorative agent is more successful. The advantage is apparent despite that the number of trials is quite low. Since the environment is static, larger number of trials would simply increase the advantages of the path selection mechanism since the robot would use the already found good solutions.

Another conclusion that can be drawn from the experimental data is that as soon as the environment becomes partially unknown, the trajectory of the robot is very difficult to predict and control. Table 2 shows that small obstacles can cause large deviation than large ones. The path selection algorithm represented here is one possibility to find reliable trajectories that increase the predictability of robot's behaviour.

Finally, we conclude that optimal (shortest) path planning is not a relevant problem in partially unknown environments. As soon as the robot does not have all global knowledge available, sub-optimal solutions give at least as good results as the optimal one. In order to increase the reliability of mobile robot applications, much more importance should be paid on modeling the environment and its changes.

References

[1] A. Howard, H.Seraji, "Vision-Based Terrain Characterization and Traversability Assessment", *Journal of Robotic Systems*, Vol. 18, No.10, pp. 577-587, Wiley periodicals, 2001.

[2] D.B.Gennery, "Traversability Analysis and Path Planning for Planetary Rovers", *Autonomous Robots*, Vol. 6. pp. 131-146, Kluwer, Academic Publishers, 1999.

[3] U.Nehmzow, C.Owen, "Robot Navigation in the Real World: Experiments with Manchester's Forty Two in Unmodified Large Environments", *Robotics and Autonomous Systems*, 33, pp.223-242, Elsevier Science, 2000.

[4] A.Yahja, S.Singh, A.Stentz, "An Efficient on-line Path Planner for Outdoor Mobile Robots". *Robotics and Autonomous Systems*, 32, pp. 129-143, Elsevier Science, 2000.

[5] M.A. Salichs and L. Moreno. "Navigation of Mobile Robots: Open Questions", *Robotica*, Vol. 18, pp. 227-234, Cambridge University Press, 2000. 6

[6] Robin R. Murphy, Ken Hughes, Alisa Marzilli and Eva Noll, "Integrating explicit path planning with reactive control of mobile robots using Trulla", *Robotics and Autonomous Systems*, Vol. 27, Issue 4, pp. 225-245, Elsevier Science, 1997.

[7] H.Seraji, "New Traversability Indices and Traversability Grid for Integrated Sensor/Map-Based Navigation", *Journal of Robotic Systems*, Vol. 20, No.3, pp. 121-134, Wiley periodicals, 2003.

[8] M.Kruusmaa, J.Willemson. "Algorithmic Generation of path Fragment Covers for Mobile Robot Path Planning", *Technical Report*, 2003.

[9] M. Kruusmaa, J.Willemson, K.Heero. "Path Selection for Mobile Robots in Dynamic Environments", *Proc. of the 1st European Conference on Mobile Robots,* Poland 2003, pp. 113 - 118.

[10] A. Zelinsky. "Using Path Transforms to Guide the Search for Findpath in 2D. *The Int. Journal of Robotics Research*, Vol. 3 No. 4, pp. 315-325, August, 1994.