Map-Based Strategies for Robot Navigation in Unknown Environments

Anthony Stentz
Robotics Institute
Carnegie Mellon University
Pittsburgh, Pennsylvania 15213
U. S. A.

Abstract

Robot path planning algorithms for finding a goal in a unknown environment focus on completeness rather than optimality. In this paper, we investigate several strategies for using map information, however incomplete or approximate, to reduce the cost of the robot's traverse. The strategies are based on optimistic, pessimistic, and average value assumptions about the unknown portions of the robot's environment. The strategies were compared using randomly-generated fractal terrain environments. We determined that average value approximations work best across small regions. In their absence, an optimistic strategy explores the environment, and a pessimistic strategy refines existing paths.

1 Introduction

Path planning for mobile robots has been extensively studied in the robotics literature. Many algorithms exist for determining an optimal path from a starting point to a goal point given complete information about the robot's environment [Latombe, 91]. If the robot's environment is not fully known during the planning phase, several algorithms exist that use local sensing to find the goal. The strategies employed by these algorithms include obstacle perimeter following, systematic search of the environment, and locally-directed wandering [Korf, 87] [Lumelsky and Stepanov, 86] [Pirzadeh and Snyder, 90] [Zelinsky, 92].

These approaches focus on completeness rather than optimality, that is, they are guaranteed to find a path to the goal if one exists but pay little attention to minimizing the cost of the robot's traverse. The key to minimizing the cost is to capitalize on what is known about the robot's environment, if anything at all. Often times the environment can be partially-known, approximated, or estimated. Furthermore, this knowledge of the environment (referred to as the *map*) can improve as the robot senses the environment, and the updated map can be used to improve both the current traverse and subsequent traverses.

This paper explores several navigation strategies for finding the goal in an unknown environment that use map information to minimize the cost of the traverse. We begin with an overview of D*, a planning algorithm that makes optimal use of map information to move a robot from start to goal. Second, we define what we mean by map information and discuss the environment modelled in our experiments. Third, we apply the D* algorithm to a set of path planning problems by varying the prior map information and quantifying the effects on the cost of the robot's traverse. Finally, we draw conclusions and describe future work.

2 The D* Algorithm

Consider the following approach for using map information during the robot's traverse. Let S be the robot's start state, G the goal state, X the current state, and M the robot's current map.

1. Store all known, approximated, estimated, and believed information about the robot's environment in M. Let X equal S.
2. Plan the optimal path from X to G using all information in M. Terminate with failure if no path can be found.
3. Follow the path from Step 2 until either G is reached (terminating with success) or the robot's sensor discovers a discrepancy between M and the environment.
4. Update M to include the sensor information and go to Step 2.

In short, this approach plans the optimal path to the goal using all known information and replans from the current state whenever new or conflicting information is discovered. We assert that this approach is not only logical but is similar to what humans do when navigating through an unknown or uncertain environment. This replanning approach produces an *optimal traverse* defined as follows:

> A traverse is the sequence of states visited by the robot enroute from S to G. A traverse is optimal if, for every state X in the sequence, the successor state to X is part of an optimal path from X to G given all aggregate map information known to the robot at state X.

The problem with this approach is strictly computational: replanning is an expensive operation. If the robot's prior map information is sparse or inaccurate, then too much time will be spent replanning in Step 2 for the approach to be viable.

The D* algorithm (Dynamic A*) [Stentz, 94] was developed to solve this computational problem. D* produces an optimal traverse by using incremental graph theory techniques to dramatically reduce the time required to replan. For environments with a large number of states, D* is capable of replanning hundreds of times faster than straightforward, brute-force replanning algorithms. Thus, D*

In Proceedings AAAI 96 Spring Symposium on Planning with Incomplete Information for Robot Problems.

enables replanning to be computationally viable even when the map information does not match the environment very well.

Like A*, D* uses a graph of states to represent the robot's environment, where each state is a robot configuration and the arcs adjoining the states represent the cost of travelling from state to state. Initially, D* computes the cost of reaching the goal to every state in the graph given all known cost information. As the robot follows a sequence of states to the goal, it may discover a discrepancy between the map and the environment. This discrepancy is manifested as an incorrect arc cost. All paths routed through this arc are invalidated and must be "repaired" [Boult, 87] [Ramalingam and Reps, 92] [Trovato, 90].

The computational advantage of D* over other techniques is twofold. First, D* only repairs those paths likely to be useful to the robot. Second, subsequent repairs can begin before the first set is completed. The net effect is less computation and a faster response, thus enabling D* to guide a robot through an unknown and uncertain robot in real-time.

Figure 1: Robot Environment with Obstacles

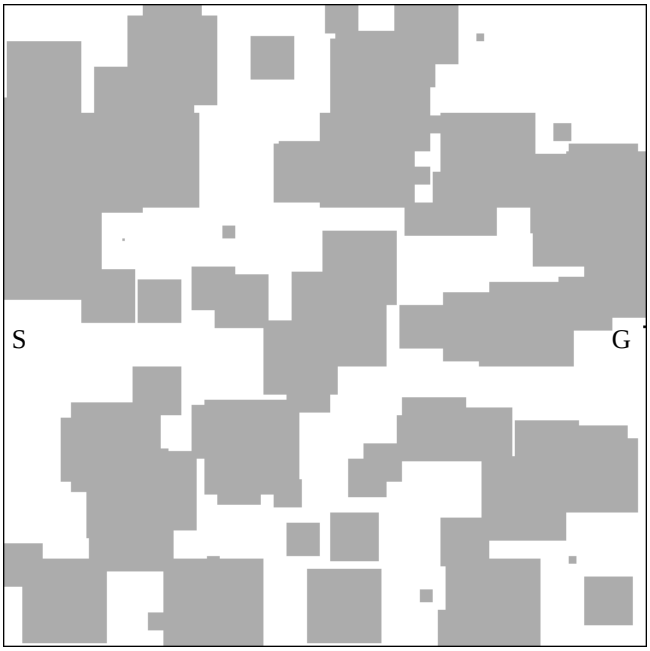


Figure 1 through Figure 4 show D* in action. Figure 1 shows a robot environment, consisting of a start state S, goal state G, and a field of obstacles shown in grey. Initially, the robot assumes that there are no obstacles (i.e., map consists solely of free space). The initial path is a straight line connecting S to G. As the robot follows the path, its radial sensor discovers the obstacles and it must replan repeatedly. Figure 2 shows this process. Only the obstacles detected by the sensor are shown. The solid line indicates the robot's traverse. The dashed line shows the path the robot is attempting to follow, given all information known in aggregate at X. This path changes frequently as more obstacles are discovered (see Figure 3). Finally, Figure 4 shows the entire traverse after the robot has reached the goal.

The basic D* algorithm is described extensively in an earlier paper [Stentz, 94]. The work was extended to further improve the computational performance given good focussing heuristics [Stentz, 95]. D* was tested both in simulation and on an actual robot [Stentz and Hebert, 95].

Figure 2: Replanning in Response to Detected Obstacles

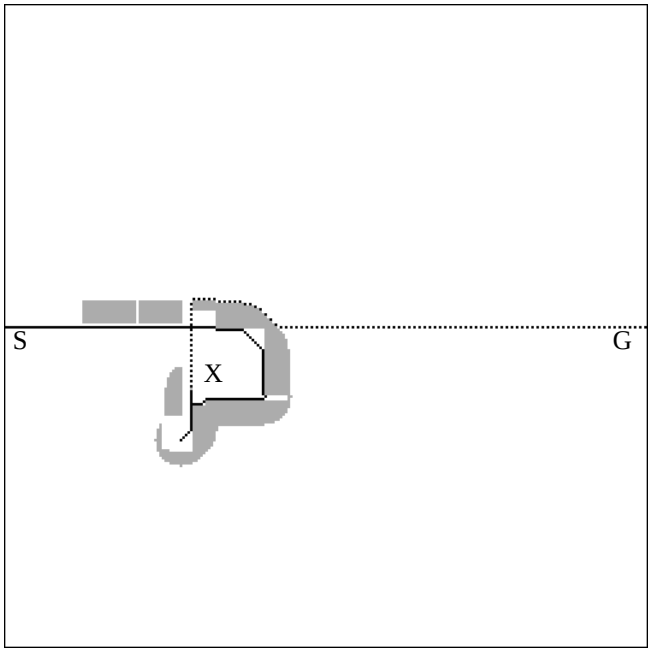


Figure 3: Robot's Target Path Changes Repeatedly

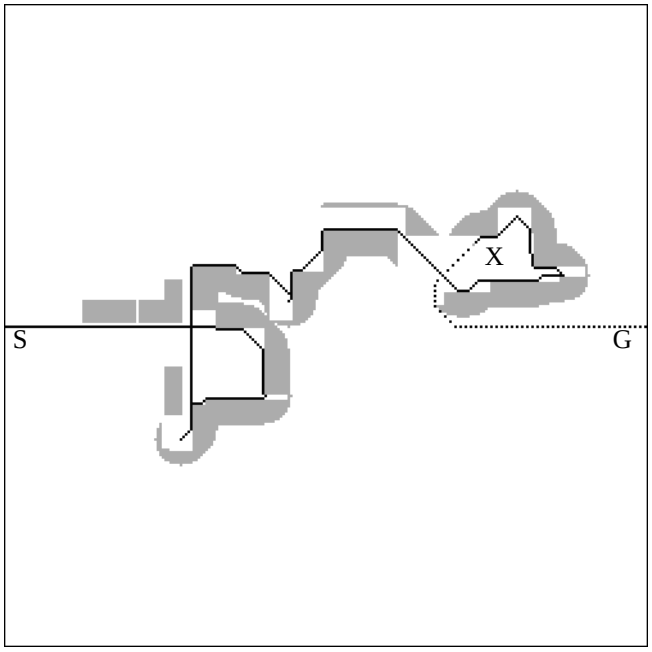
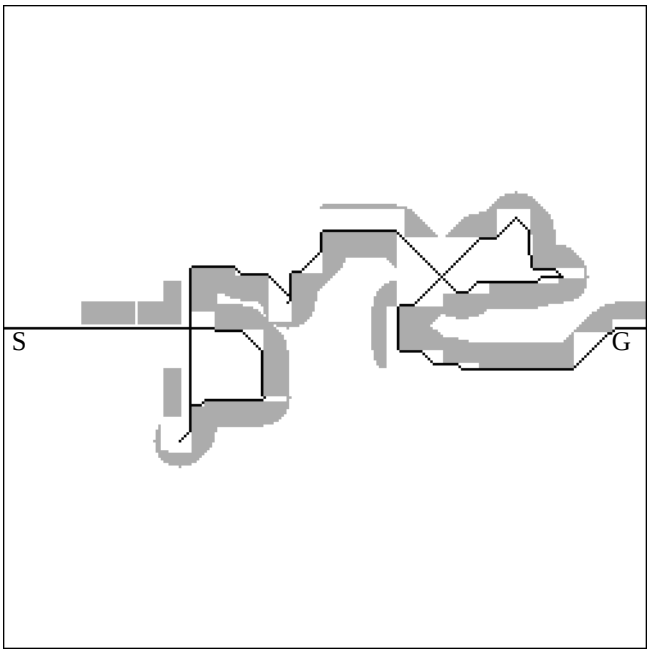


Figure 4: Robot's Complete Traverse



3 Map Information

The map holds the robot's knowledge of its environment. That knowledge may be exact cost values, estimates, or assumptions. In order to follow the approach outlined in Section 2, we need cost values for all arcs in the graph, that is, the map must be complete in the sense that unknown portions must be estimated or approximated if exact values are unavailable. Humans do the same. We operate with default assumptions such as worst case, best case, or average case scenarios when perfect information is unavailable.

3.1 Environment Model

In order to evaluate different strategies for navigating in an uncertain environment, we choose a specific type of robot environment for our experiments. Without a loss of generality, we assume that the robot's environment is a two-dimensional, eight-connected grid. Each grid cell contains a positive cost value representing the per unit cost of moving across the cell. If the cell contains an impassable obstacle, the cost value is *infinite*. The robot's map is a data structure of the same form with exact, estimated, or approximate costs for each corresponding environment cell.

For our experiments, we could use environments with cost values generated from a uniform distribution, but these do not typically match environments likely to be encountered by a real robot. Instead, we select a realistic model template and generate random examples from the template. We choose to simulate outdoor terrain, and we use a fractal model of gaussian hills and valleys to do so.

Each environment is 500 x 500 cells and is created by randomly selecting a location within the environment and centering a gaussian distribution on the location with a randomly selected amplitude. A positive amplitude is a hill, and a negative is a valley. The environment is split into four quadrants, and a smaller gaussian is added in each quadrant. This process recurs until the highest resolution is reached. The result is fractal terrain with "lumpy" hills and valleys. The elevation gradient is computed at each cell, and a low cost is assigned to the cell if the gradient is nearly level and a high cost if the gradient is steep. If the steepness exceeds a threshold, the cell is assigned an infinite cost (i.e., an obstacle). These environment costs model a robot that expends energy moving up and down outdoor terrain.

Figure 5: Randomly-Generated Fractal Terrain

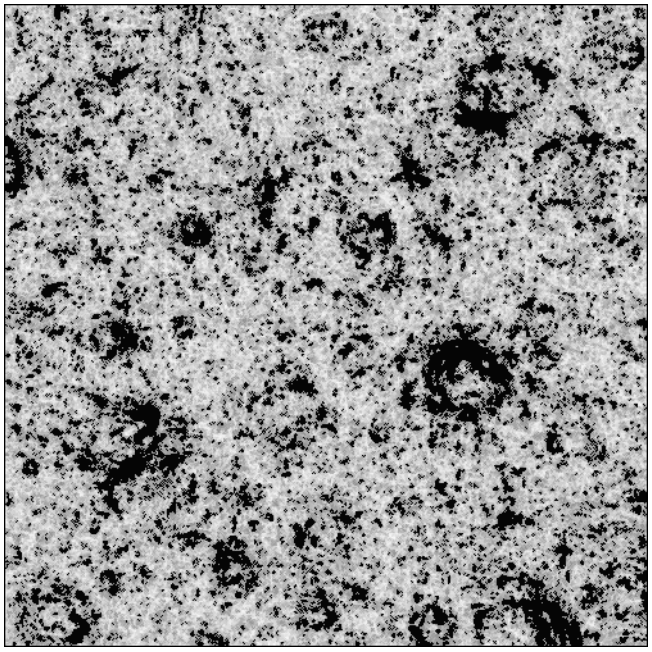


Figure 5 illustrates one such randomly-generated environment. The grey scales denote a range of costs from low (white) to high (dark grey). The obstacles are shown in black.

3.2 Map Strategies

Unless the robot's environment is completely known, there exists one or more cells for which the cost value must be estimated. There are innumerable strategies for estimating these values, but we limit our discussion to three that we feel are intuitive and characteristic of what humans do. The three are described below:

- **Optimistic strategy:** each cell for which the cost value is unknown is assumed to be of lowest cost. For the environment model above, this corresponds to level, easy-to-traverse terrain.
- **Pessimistic strategy:** each cell for which the cost value is unknown is assumed to be of highest cost. For the environment model above, this corresponds to the steepest terrain that is still traversable.

- **Average value strategy:** each cell for which the cost value is unknown is assumed to be equal to the average of known cells in the vicinity.

The optimistic and pessimistic strategies correspond to best case and worst case approaches, respectively. The average value strategy assumes that the first moment of the cost distribution is a good estimate of an individual member's value.

Note that none of the above strategies assumes that an unknown cell contains an infinite cost value (obstacle). Storing an obstacle in the map where one does not exist could cause the D* planning algorithm to wrongly assume that no path exists to the goal. This case would occur if an obstacle is incorrectly assumed to block the only passageway to the goal. To circumvent this problem, the pessimistic strategy uses the highest-cost value that is still admissible. For the average value strategy, infinite-value obstacles can not be averaged into the statistic; instead, an amount equal to the cost of moving around the obstacle is used.

The evaluations for the three strategies are discussed in the next section.

4 Strategy Evaluations

In this section, we evaluate the optimistic, pessimistic, and average value strategies by applying them to some randomly-generated environments and measuring the cost of the robot's traverse.

4.1 Description of Experiments

For our experiments, we randomly generated 100 environments according to the model in Section 3.1 and simulated a robot traverse through each environment. The environments were 500 x 500 cells and the most difficult (but navigable) terrain was five times more difficult to traverse than the easiest. The robot started in the lower-left corner of the environment and was given a goal in the upper-right corner. The robot was equipped with a 20-cell radial sensor that accurately measured cell costs within its field of view. The D* algorithm was used to move the robot from start to goal. Each time new information was detected by the robot's sensor, the map was updated and the path replanned.

The optimistic strategy used a value of 10 for the cost of unknown cells, and the pessimistic strategy used 50. The average value strategy was tested several times at different resolutions. At the lowest resolution (e.g., 1), the average cost was computed across the entire environment. At resolution N, the environment was partitioned into N x N squares, an average was computed within each square, and the average was stored into the map for each cell in the square. Resolutions of 1, 10, and 100 were tested, representing progressively better and more localized information.

For each strategy, we report the average traversal cost for the 100 trials and the average rank (where 1 is assigned to the best strategy and 5 to the worst for each trial). Additionally, we report the average traversal costs for a sequence of three traverses across each environment (from the same start to the same goal). The second and third traverses used the updated maps produced by their predecessors during their traverses.

4.2 Results

The results are given in Table 1. As the rank data indicates, the best strategy is to estimate unknown cells with the average value of cells in the vicinity. The size of the "vicinity" is very important, however. As the resolution decreases, the size of each vicinity increases, and the strategy degrades in performance. At the limit (i.e., resolution of 1), the strategy is actually slightly worse than both the optimistic and pessimistic approaches. The differences between the optimistic, pessimistic, and average value (resolution 1) strategies are rather small for the first traverse. The traverses for each strategy as applied to the environment shown in Figure 5 are shown in Figure 6 through Figure 10. These traverses typify the data set as a whole.

	Rank	Trav 1	Trav 2	Trav 3
Optimistic	3.17	53,142	58,946	55,885
Pessimistic	3.75	53,745	46,994	45,845
Avg Res 1	3.81	53,744	46,993	45,611
Avg Res 10	2.65	47,931	44,756	44,387
Avg Res 100	1.62	44,200	43,486	43,468

Table 1: Comparison Results

Figure 6: Optimistic Traverse

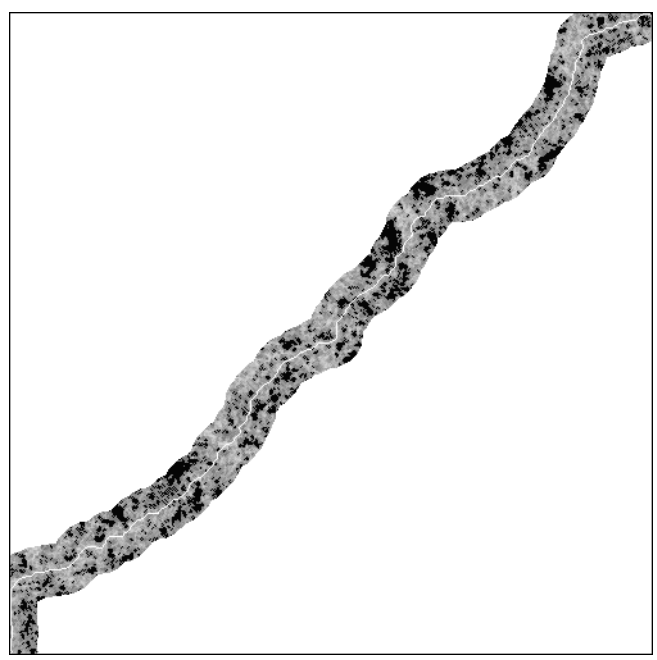


Figure 7: Pessimistic Traverse

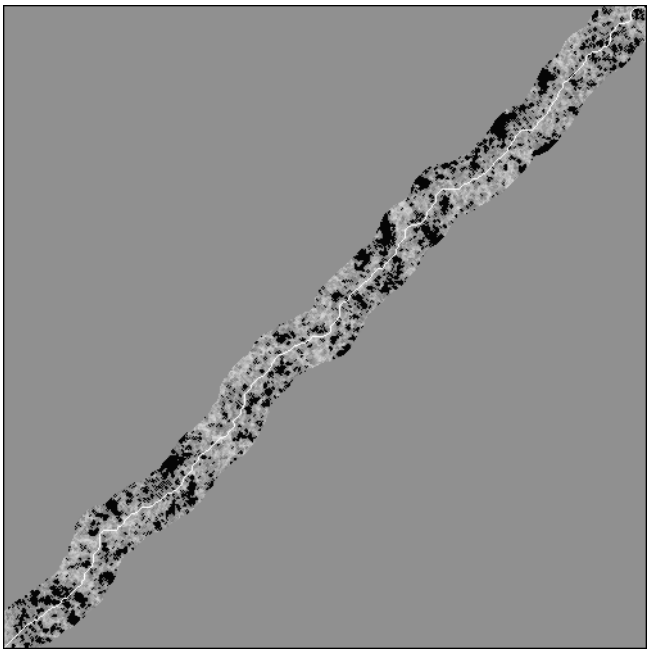


Figure 8: Average Value (Resolution 1) Traverse

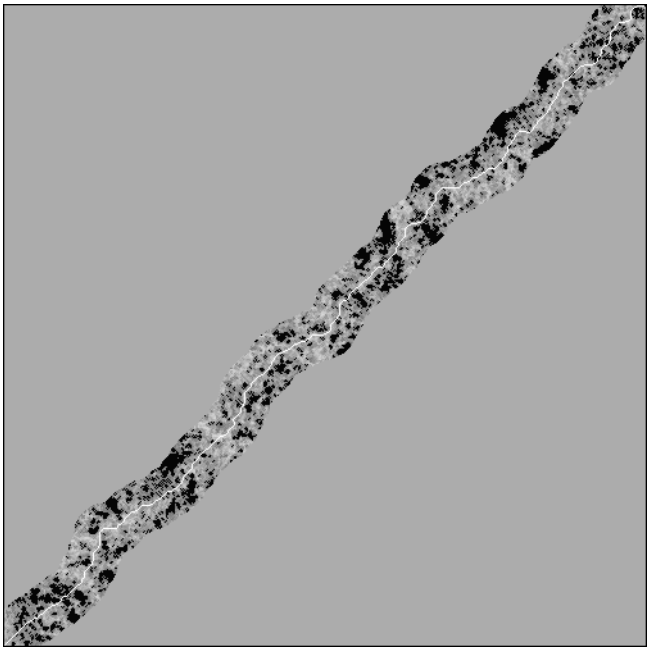


Figure 9: Average Value (Resolution 10) Traverse

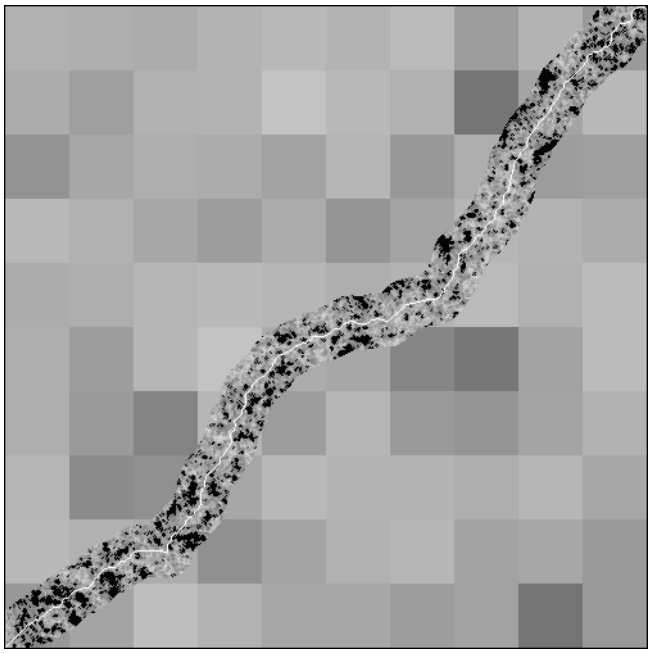
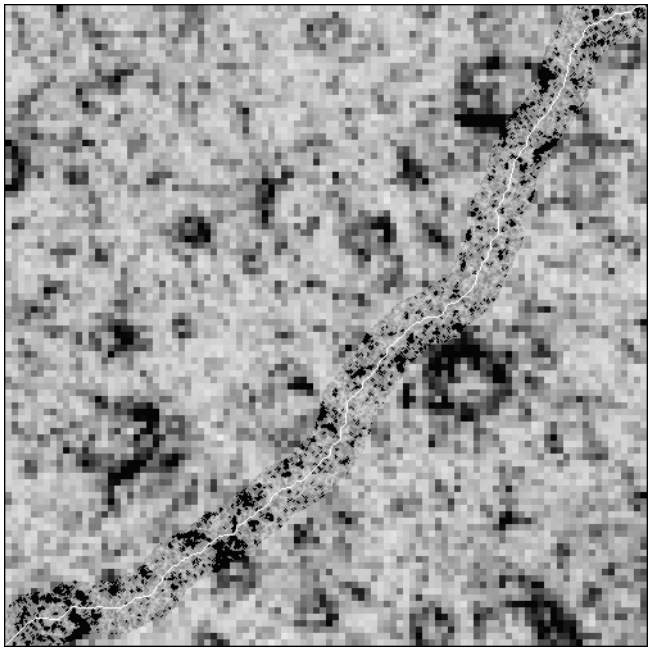


Figure 10: Average Value (Resolution 100) Traverse



The difference between the optimistic and pessimistic strategies becomes evident in subsequent traverses. The optimistic assumption is that the unknown world is better than what has been seen so far. The pessimistic assumption is that it is worse. Thus, a robot operating with the optimistic strategy will tend to explore new routes, and a robot operating with the pessimistic strategy will tend to stay within explored routes until forced to do otherwise.

Figure 11 and Figure 12 illustrate this effect. Figure 11 shows three traverses (numbered) through the environment in Figure 5 using the optimistic strategy. The largely straight-line path to the goal taken during the first traverse is close to the lowest-cost possible, yet in subsequent traverses the optimistic strategy searches on either side for a better route. For the typical environment, this strategy does not pan out and subsequent traverses are higher in cost.

Figure 11: Three Optimistic Traverses

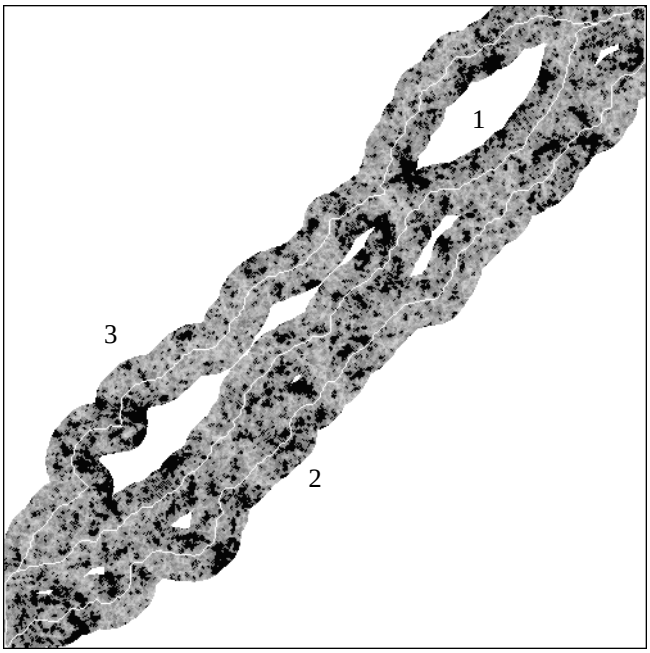


Figure 12: Three Pessimistic Traverses

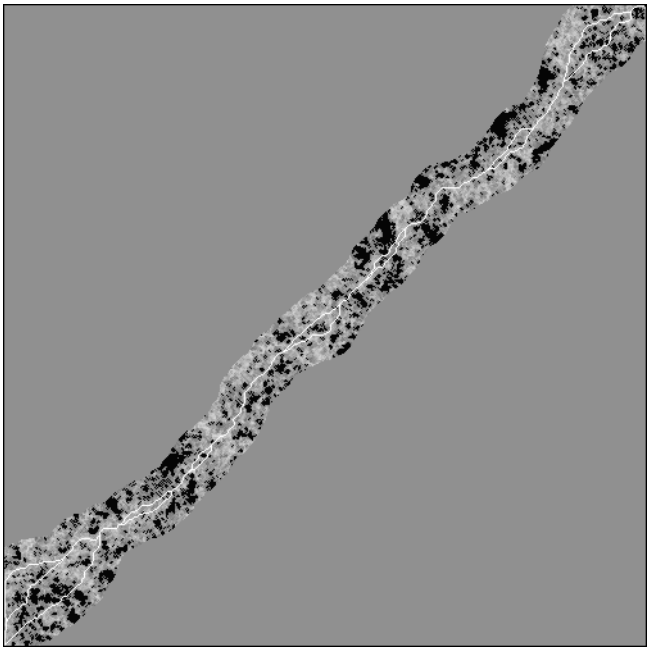


Figure 13: Atypical Terrain Environment

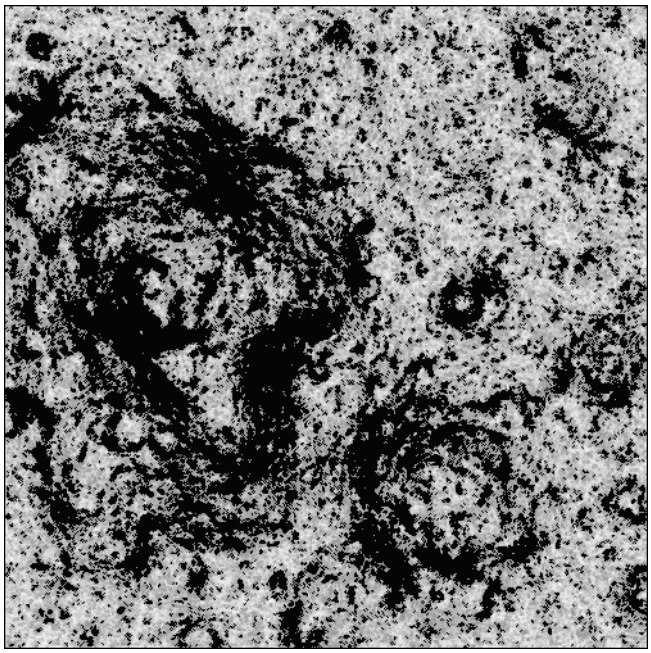


Figure 14: Optimistic Traverses Explore to Find Best Route

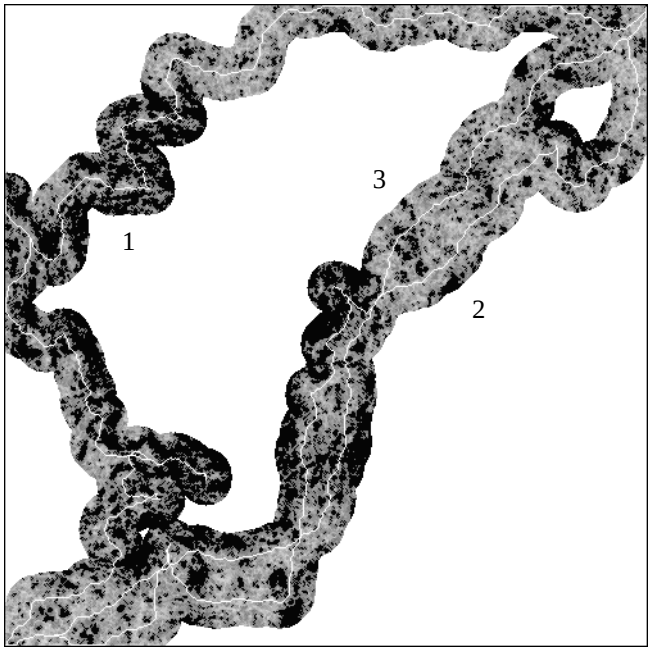


Figure 12 shows three traverses through the same environment using the pessimistic strategy. Rather than venturing into new terrain, the pessimistic strategy optimizes the path within the swath sensed during the first traverse. Thus, subsequent traverses are lower in cost. The same can be said for the average value strategies at all resolutions, with the exception that they are slightly more willing to venture from the previous path to “cut a corner” than the pessimistic strategy.

Figure 13 shows an atypical environment from the data set. In this case, the environment consists of a few very large obstacles rather than a scattering of smaller ones. Three optimistic traverses are shown in Figure 14. On the first traverse, the robot takes the high-cost route around the large obstacle to left. In subsequent traverses, it optimistically searches to the right and discovers a better route that significantly reduces the traversal cost.

Figure 15: Pessimistic Traverses Optimize Locally

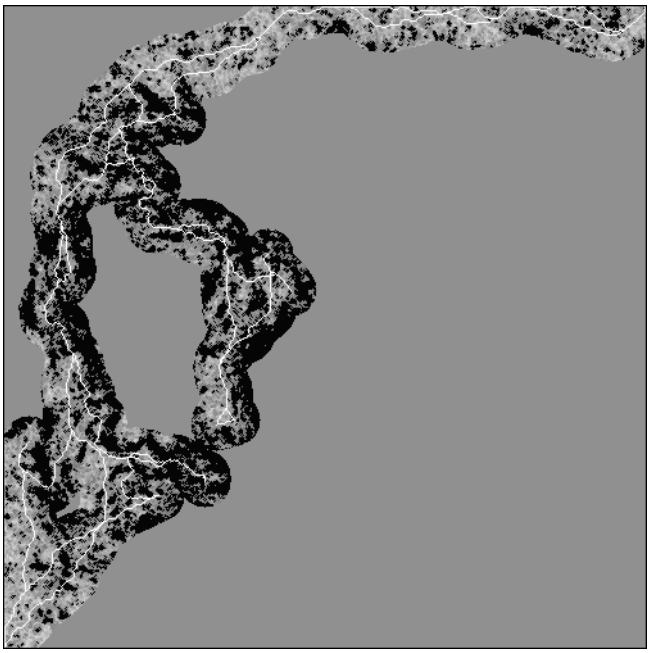
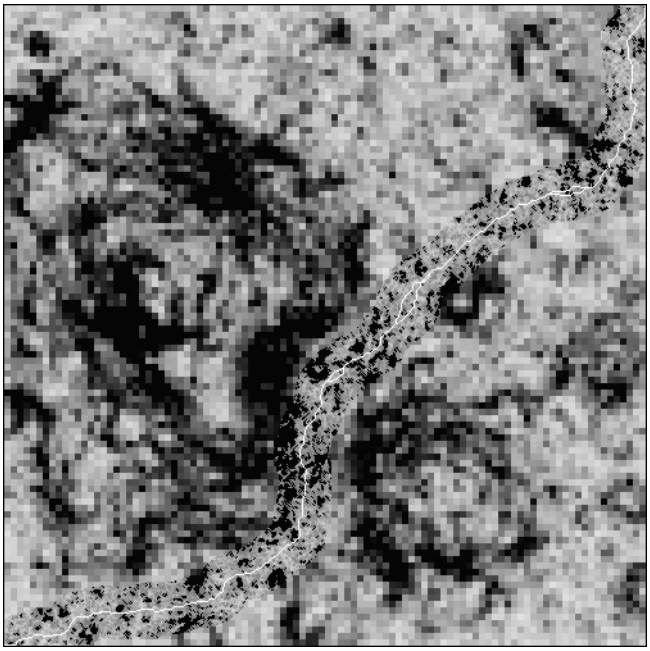


Figure 16: High-Resolution Traverses Converge Quickly



The pessimistic robot also takes a high-cost route on its first attempt (Figure 15). Subsequent traverses only optimize locally, and the robot never discovers the lower-cost route to the right. The medium- and high-resolution average value strategies (Figure 16) have enough prior map information available to focus quickly on the best route. Subsequent traverses lead to slightly lower-cost routes.

Thus, for complicated environments with convoluted paths to the goal for which too little prior information exists, the optimistic strategy is better than the pessimistic one since some exploration is required to locate the global minimum.

5 Conclusions

In conclusion, we have determined that for a class of outdoor terrain environments, estimating unknown portions of the environment with known portions in the vicinity yields the lowest-cost traverse. The smaller the vicinity the better. In the absence of such information, the optimistic or pessimistic strategies can be used. Repeated pessimistic traverses tend to converge to a local minimum in the solution space; repeated optimistic traverses tend to jump around the space and are more likely to discover the global minimum at the expense of local improvements.

In the future, we will explore a probabilistic representation for the map and will employ a decision-theoretic approach for minimizing the cost of the traverse. This approach fits well within the D* replanning paradigm.

Acknowledgments

This research was sponsored by ARPA, under contracts “Perception for Outdoor Navigation” (contract number DACA76-89-C-0014, monitored by the US Army TEC) and “Unmanned Ground Vehicle System” (contract number DAAE07-90-C-R059, monitored by TACOM).

References

[Boult, 87] T. Boult. Updating distance maps when objects move. In *Proceedings of the SPIE Conference on Mobile Robots*, 1987.

[Korf, 87] R. E. Korf. Real-time heuristic search: first results. In *Proceedings of the Sixth National Conference on Artificial Intelligence*, July 1987.

[Latombe, 91] J.-C. Latombe. *Robot Motion Planning*, Kluwer Academic Publishers, 1991.

[Lumelsky and Stepanov, 86] V. J. Lumelsky and A. A. Stepanov. Dynamic path planning for a mobile automaton with limited information on the environment. *IEEE Transactions on Automatic Control*, AC-31(11), November 1986.

[Pirzadeh and Snyder, 90] A. Pirzadeh and W. Snyder. A unified solution to coverage and search in explored and unexplored terrains using indirect control. In *Proceedings of the IEEE International Conference on Robotics and Automation*, May 1990.

[Ramalingam and Reps, 92] G. Ramalingam and T. Reps. An incremental algorithm for a generalization of the shortest-path problem. University of Wisconsin Technical Report #1087, May 1992.

[Stentz, 94] A. Stentz. Optimal and efficient path planning for partially-known environments. In *Proceedings of the IEEE International Conference on Robotics and Automation*, May 1994.

[Stentz, 95] A. Stentz. The focussed D* algorithm for real-time replanning. In *Proceedings of the International Joint Conference on Artificial Intelligence*, August 1995.

[Stentz and Hebert, 95] A. Stentz and M. Hebert. A complete navigation system for goal acquisition in unknown environments. *Autonomous Robots*, 2(2), 1995.

[Trovato, 90] K. I. Trovato. Differential A*: an adaptive search method illustrated with robot path planning for moving obstacles and goals, and an uncertain environment. *Journal of Pattern Recognition and Artificial Intelligence*, 4(2), 1990.

[Zelinsky, 92] A. Zelinsky. A mobile robot exploration algorithm. *IEEE Transactions on Robotics and Automation*, 8(6), December 1992.