

Recent Results in Path Planning for Mobile Robots Operating in Vast Outdoor Environments

Alex Yahja, Sanjiv Singh and Anthony Stentz
{ay, ssingh, axs}@ri.cmu.edu

Robotics Institute
Carnegie Mellon University
Pittsburgh, PA 15213

Corresponding Author: Sanjiv Singh

Abstract

Mobile robots operating in vast outdoor unstructured environments often only have incomplete maps and must deal with new objects found during traversal. Path planning in such sparsely occupied regions must be incremental to accommodate new information and must use efficient representations. In this paper we report recent results in path planning using an efficient data structure (framed quadtrees) and an optimal algorithm (D^) to incorporate knowledge of the environment as it incrementally discovered. In particular, we show the difference in performance when the robot starts with no information about the world versus when it starts with partial information about the world. Our results indicate that, as would be expected, starting with partial information is better than starting with no information. However, in many cases, the effect of partial information is performance that is almost as good as starting out with complete information about the world, while the computational cost incurred is significantly lower. Our system has been tested in simulation as well on an autonomous jeep, equipped with local obstacle avoidance capabilities and results from both simulation and real experimentation are discussed.*

Keywords: *efficient and optimal path planning, outdoor, mobile robots.*

1 Introduction

Path planning for a mobile is typically stated as getting from one place to another. The robot must successfully navigate around obstacles, reach its goal and do so efficiently. Outdoor environments pose special challenges over the structured world that is often found indoors. Not only must a robot avoid colliding with an obstacle such as a rock, it must also avoid falling into a pit or ravine and avoid travel on terrain that would cause it to tip over. Vast areas have their own associated issues. Such areas typi-

cally have large open areas where a robot might travel freely and are sparsely populated with obstacles. However, the range of obstacles that can interfere with the robot's passage is large—the robot must still avoid a rock as well as go around a mountain. Vast areas are unlikely to be mapped at high resolution a priori and hence the robot must explore as it goes, incorporating newly discovered information into its database. Hence, the solution must be *incremental* by necessity. Another challenge is dealing with a large amount of information and a complex model (our autonomous vehicle is a three degree of freedom, non-linear, non-holonomic system). Taken as a single problem, so much information must be processed to determine the next action that it is not possible for the robot to perform at any reasonable rate. We deal with this issue by using a *layered* approach to navigation. That is, we decompose navigation into two levels—*local* and *global*. The job of local planning is to avoid obstacles, reacting to sensory data as quickly as possible while driving towards a subgoal [4][5]. A more deliberative process, operating at a coarser resolution of information is used to decide how best select the subgoals such that the goal can be reached. This approach has been used successfully in the past in several systems at Carnegie Mellon [1][13]. In this paper we concentrate the discussion on global planning.

Approaches to path planning for mobile robots can be broadly classified into two categories—those that use exact representations of the world (e.g. [6][15]), and those that use a discretized representation (e.g. [3][7]). The main advantage of discretization is that the computational complexity of path planning can be controlled by adjusting the cell size. In contrast, the computational complexity of exact methods is a function of the number of obstacles and/or the number of obstacle facets, which we cannot normally control. Even with discretized worlds path planning can be computationally expensive and on-line performance is typically achieved by use of specialized

computing hardware as in [3][7]. By comparison the proposed method requires general purpose computing only. This is made possible by precomputing an optimal path off-line given whatever a priori map is available, and then optimally modifying the path as new map information becomes available, on-line.

Methods that use uniform grid representations must allocate large amounts of memory for regions that may never be traversed, or contain any obstacles. Efficiency in map representation can be obtained by the use of quadtrees, but at a cost of optimality. Recently, a new data structure called a *framed quadtree* has been suggested as means to overcome some of the issues related to the use of quadtrees[2]. We have used this data structure to extend an existing path planner that has in the past used uniform (regular) grid cells to represent terrain. This path planner, D^* [12] has been shown to optimally incorporate knowledge of the environment as it is incrementally discovered. Coupling the two provides a method that is correct, resolution complete and resolution optimal. It also does this efficiently. The paths are always shorter, and in all but the most cluttered environments, it executes faster and uses less memory than when regular grids are used [16]. In general, the sparser or the more unknown the world is, the higher the advantages of using framed-quadtrees.

In this paper we explain the rationale of using framed quadtrees along with D^* . We describe the recent results of path planning in continuous-cost (simulated) fractal worlds. In particular, we show the difference in performance when the robot starts with no information about the world versus when it starts with partial information about the world. As would be expected, starting with partial information is better than starting with no information. However, in many cases, the effect of partial information is performance that is almost as good as starting out with complete information about the world, while the computational cost incurred is significantly lower. In addition, we present results of an implementation on an autonomous jeep based on binary-cost framed-quadtree D^* .

2 Map Representation

Apart from the fact that discretization of space allows for control over the complexity of path planning, it also provides a flexible representation for obstacles and cost maps, and eases implementation. One method of cell decomposition is to tessellate space into equal sized cells each of which is connected to its neighbors with four or eight arcs. This method has two drawbacks: resulting paths can be suboptimal and memory requirements high. Quadtrees address the latter problem, while framed quadtrees address both problems, especially in sparse environments.

2.1 Regular Grids

Regular grids represent space inefficiently. Natural terrains are usually sparsely populated and are often not completely known in advance. In the absence of map information unknown environments are encoded sparsely during initial exploration and many areas remain sparsely populated even during execution. Many equally sized cells are needed to encode these empty areas making search expensive since more cells are processed than actually needed. Moreover, regular grids allow only eight angles for direction, resulting in abrupt changes in path direction and an inability, in some cases, to generate a straight path through empty areas (Fig. 1a). It is possible to smooth such jagged paths, but there is no guarantee that the smoothed path will converge to the truly optimal path.

2.2 Quadtrees

One way to reduce memory requirements is to use a quadtree instead of a regular grid. A quadtree [9][10] is based on the recursive subdivision of a region into four equally sized quadrants. Quadtrees allow efficient partitioning of the environment since single cells can be used to encode large empty regions. However, paths generated by quadtrees are suboptimal because they are constrained to segments between the centers of the cells. Fig. 1b shows an example path generated using a quadtree.

2.3 Framed Quadtrees

To remedy the above problem, we have used a modified data structure in which cells of the highest resolution are added around the perimeter of each quadtree region. This augmented representation is called a *framed quadtree*. A path generated using this representation is shown in Fig. 1c. The small grey rectangles around the cells are the border cells of each quadrant. This representation permits many angles of direction, instead of just eight angles as in the case of regular grids. A path can be constructed between two border cells that are far away from each other. Most importantly, the paths generated more closely approximate optimal paths. The drawback of using framed quadtrees is that they can require more memory than regular grids in uniformly, highly cluttered environments because of the overhead involved in the book-keeping.

3 Incremental Planning

So far we have discussed path planning in cases when the world is known completely a priori. However, typical outdoor environments are often not only sparse but often if any information is available about the terrain, it is only at a coarse level. If complete and accurate maps were available, it would be sufficient to use A^* [8] once to search the map and produce a path. The robot could simply follow

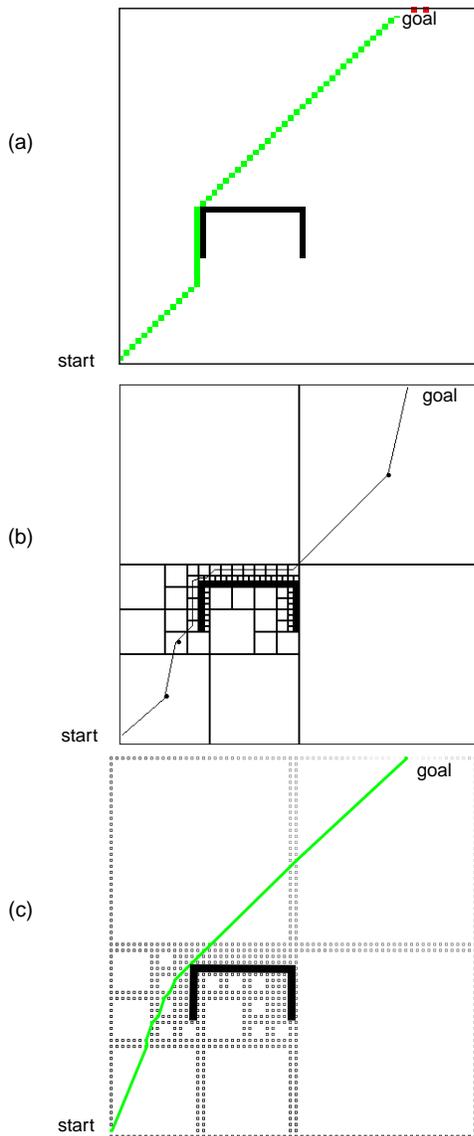


Fig. 1 An example of a path generated using (a) regular grid representation, (b) quadtree, (c) framed-quadtree. The black cul-de-sac is an obstacle.

this path during its traverse. Ideally, the robot should gather new information about the environment, and efficiently replan new paths based on this new information.

The idea is to produce a path based on all available information and replan from the current position to the goal every time new information is discovered. This is called “Best Information Planning”. This approach [14] has been shown to produce lower-cost traverses on average than other selected algorithms for unknown and partially-known environments. Furthermore, Best Information Planning is able to make use of prior information to reduce the

traversal cost.

Obviously, we can just use A* to replan a new path every time it is needed, but this approach is computationally expensive. Our approach is to use the incremental planner, D* [11][12], that allows replanning to occur in realtime. Incremental replanning makes it possible to greatly reduce computational cost, as it only updates the path locally, when possible, to obtain the globally optimal path. D* produces the same results as planning from scratch with A* for each new piece of information, but for large environments it is hundreds of times faster.

4 Simulation Results

We have conducted path planning experiments using simulated fractal terrains of varying complexity. The simulation environment is a binary 256 x 256 cell world with obstacles (each a 1 x 1 cell) distributed by a fractal terrain generator. The amount of clutter in the world is parameterized by a fractal gain.

We have used two representations for obstacles. The first is a simplified representation in which the terrain has a binary cost depending on a fractal generator. Either the terrain is passable in which case the cost to move from one cell to another is the euclidean distance, or, the terrain is impassible and the cost to move to a cell containing an obstacle is infinite. We have also experimented with a more realistic representation that encodes the cost of moving from one cell to another as a function of a fractal, resulting in a continuous-cost map. In this case, the fractal generator directly produces a cost map, that is, it directly produces the cost of traversing from one cell to another and can be thought of as a derivative of an elevation map.

4.1 Binary-Cost Worlds

An extensive set of simulations were conducted using regular grids and framed quadtrees and the results are available in [16]. Here we simply show the difference in the path produced when framed-quadtrees are used as opposed to regular grids. One of such comparison is depicted by Figure 2, in which the path length of framed-quadtrees (right) is shorter and smoother than the one resulted using regular-grids (left).

4.2 Continuous-Cost Worlds

Rarely is it possible to get perfect information about the world before the robot starts. Sometimes, no prior information is available and the robot must use its own sensors to discover the world. In some cases, however, the terrain that the robot must traverse is known at a low resolution such as would be produced by an aerial flyover. A

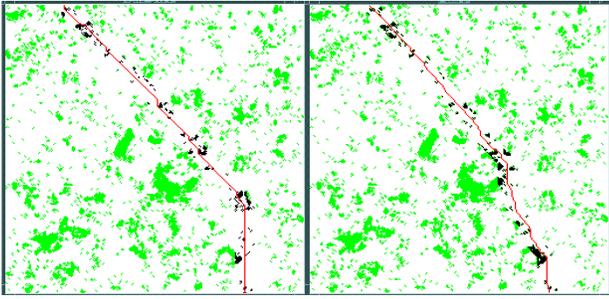


Fig. 2 Traverses generated in a binary fractal world using regular grids (left) and frame quadtrees (right). The lighter cells represent occupied areas that are unknown in advance. The dark cells represent the obstacles that are discovered by the vehicle's sensors. This world has a fractal gain of 12.

low resolution map (coarse map) contains partial information about the world. We have examined performance in all three cases.

For example, Fig. 3 shows traverses in a world of which the robot (a) has no knowledge before it starts (b) has a coarse map before it starts, and finally (c) has a complete map a priori. Fig. 3a shows the structure of framed-quadtrees generated by the end of the traverse. As can be seen, the path found when a coarse map is available resembles the best possible path that can be found between the start and the goal.

We have conducted a total of 3000 experiments and present the results below in terms of traversal. Terrain density is parameterized in ten steps of fractal gain. For each step in terrain density, we conducted 100 runs for each of the three cases (completely unknown, partially known and totally known). Below, we make comparisons on the basis of traversal cost, memory usage and execution time.

4.2.1 Traversal Cost

Note that traversal cost is not exactly the same as the traversal length. Hence a meandering path might be picked even if a direct path is available but the slope connecting the two is very large.

Figure 4 shows traverses in the fully known world have the lowest cost. Conversely, traverses in a completely unknown world have the highest cost because the robot often enters high cost areas and in some cases goes down dead ends before it backtracks. Traverses in the partially known world are in the between the two curves but it is interesting to note that even with 1/64th as much a priori information as in the fully known case, the traversal cost is not significantly higher.

4.2.2 Memory Usage

Figure 5 shows that comparatively encoding the fully known world uses the most memory, while the completely

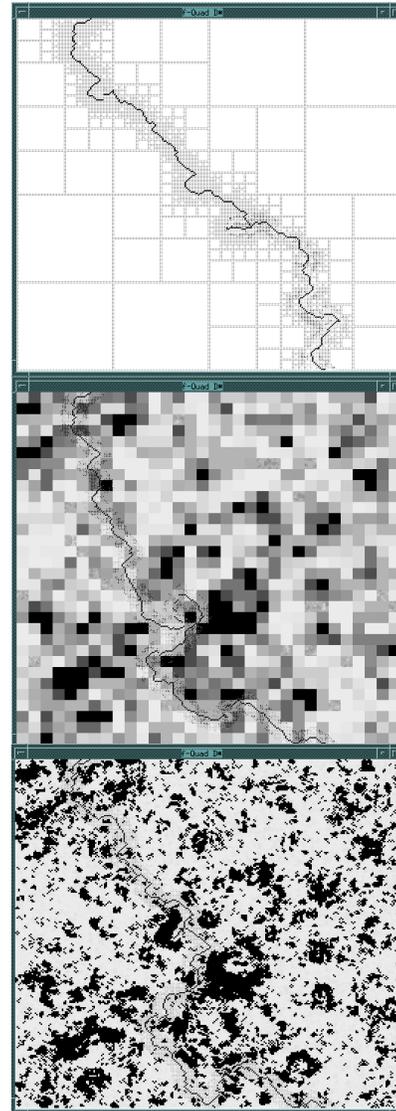


Fig. 3 A traverse in (a) completely unknown continuous-cost fractal world, (b) a coarse information continuous-cost fractal world. Each cell's cost corresponds to the average of an 8 x 8 area, and (c) a fully known continuous-cost fractal world

unknown world uses the least. The coarsely known world is in between. As expected, the more information you have, the more memory is required by a framed-quadtrees to represent that information.

4.2.3 Execution Time

Traverses in the fully known world execute faster for sparse worlds, but executes slower for denser worlds (due to the time needed to build up framed quadtrees and to propagate initial D* values) than the completely unknown world. However, traverses in the coarsely known world execute faster than both as fewer data structures are necessary than in the case of the fully known world, and

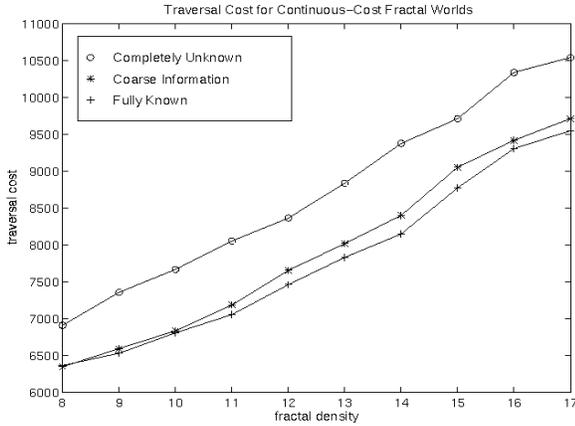


Fig. 4 Traversal cost comparison.

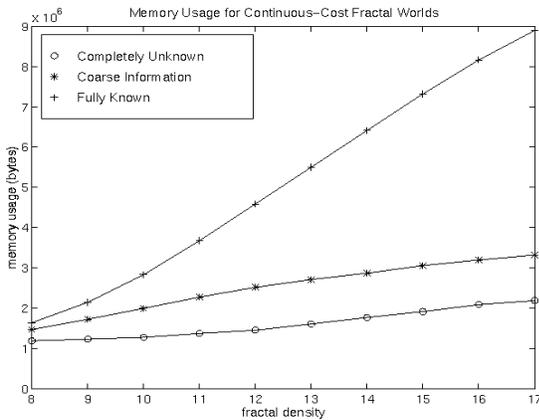


Fig. 5 Memory usage comparison.

its traverse is less likely to stumble into high cost areas than in the case of the completely unknown world. This is depicted in Figure 6.

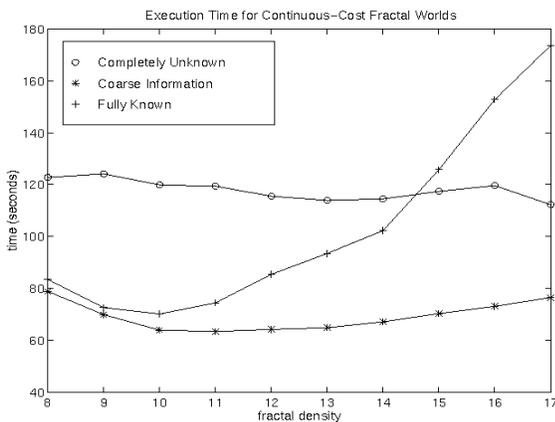


Fig. 6 Execution time comparison.

5 Test Results on Autonomous Vehicle

We have performed several tests on an automated military jeep (Fig. 7). Our vehicle uses a vertical-baseline stereo system to generate range images. The resulting images are

processed by the SMARTY local navigator [4], which handles local obstacle detection and avoidance. This obstacle map is fed to a global navigator running a path planning algorithm, such as framed-quadtree D*. Both the local and global navigators submit steering advice to an arbiter, which selects a steering command each time interval and passes it to the controller [13]. Fig. 8 shows the system modules and data flow.



Fig. 7 The autonomous vehicle (HMMWV) that used for our experiments. The vehicle is equipped with stereo vision, inertial guidance and GPS positioning.

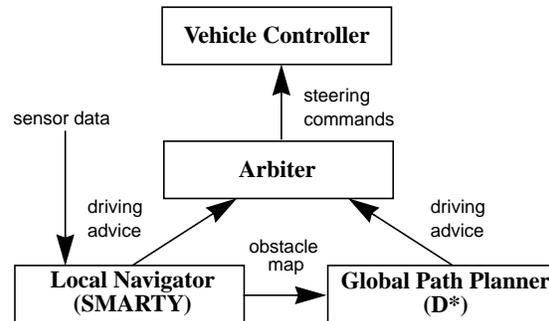


Fig. 8 Data flow in the implemented system.

Fig. 9 shows a successful traverse of the vehicle that covered 200 meters in 6 minutes. During this traverse, the vehicle detected and avoided 80 obstacles.

Fig. 10 shows a close-up of the data structure produced after the above run. As expected, a large part of the environment that is not explored is represented by a small number of cells.

6 Conclusions

We have extended our method for global path planning suited to autonomous vehicles operating in vast unstructured environments. Our previous method combined the D* algorithm, and a binary framed-quadtree data structure. While a binary world is acceptable for robots operat-

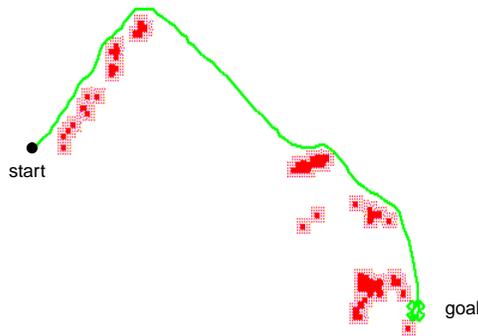


Fig. 9 Successful long traverse of the vehicle using framed-quadtree D* through a terrain with obstacles to the goal. The dark rectangles are obstacles detected and avoided during the traverse. The shaded areas surrounding the dark obstacles are potentially dangerous zones.

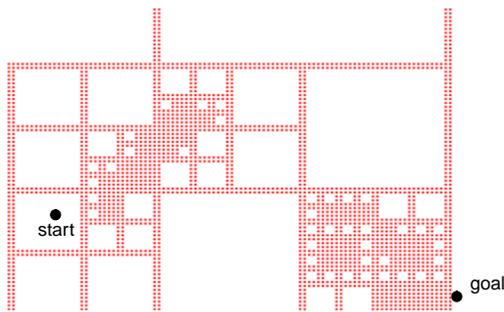


Fig. 10 A close up of the data structure produced from the execution of the path in Fig. 9.

ing indoors, it is desirable to use a representation that encodes the three dimensional nature of outdoor terrain. Our extended method can handle continuous-cost situations that are more representative of natural terrains.

The results from extensive simulation in continuous-cost worlds shows that coarse terrain information while not only practical can also result in reduced execution times while incurring only a small cost in optimality over the perfectly known world. In terms of memory usage, there is also a persuasive argument to use coarse information about the world. Savings in memory can range from 30% (sparse worlds) to 75% (cluttered worlds) over that required for a fully known world. As a comparison, note that the memory requirements remain constant, irrespective of the number of the objects in the world when a regular grid is used. Hence this method lends itself to the applications such as future planetary rovers which will have severe limitations in memory, but, nevertheless require relatively high performance.

Acknowledgments

This research was sponsored in part by DARPA, under contract DAAE07-96-C-X075 "Technology Enhancements for Unmanned Ground Vehicles" and by the Defense Research Establishment Suffield, Canada, under contract W7702-6-R577/001 "Performance Improvements

for Autonomous Cross-Country Navigation".

References

- [1] Brumitt, B.L., Stentz, A., "Dynamic Mission Planning for Multiple Mobile Robots," In Proceedings of the IEEE International Conference on Robotics and Automation, May 1996.
- [2] Chen, Szczerba, Uhran, "Planning Conditional Shortest Paths through an Unknown Environment: A Framed-Quadtree Approach," Proceedings of the IEEE International Conference on Robotics and Automation, May 1995.
- [3] Connolly and Grupen, "The Application of Harmonic Functions to Robotics," *Journal of Robotic Systems*, 10(7):931-946.
- [4] Hebert, Martial H., "SMARTY: Point-Based Range Processing for Autonomous Driving," *Intelligent Unmanned Ground Vehicle*, Martial H. Hebert, Charles Thorpe, and Anthony Stentz, editors, Kluwer Academic Publishers, 1997.
- [5] Kelly, A., "An Intelligent Predictive Control Approach to the High Speed Cross Country Autonomous Navigation Problem," Ph.D Thesis, 1995, Carnegie Mellon University, Pittsburgh, PA 15213.
- [6] Lozano-Perez, T., and Wesley, M. A. "An algorithm for planning collision-free paths among polyhedral objects," *Communications ACM* 22, 10, October 1979.
- [7] Lengyel, J. and Reichert, M. and Donald, B. R. and Greenberg, D. P., "Real Time Robot Motion Planning Using Rasterizing Computer Graphics Hardware," In Proc. SIGGRAPH. 1990.
- [8] Russell, S., Norvig, P., *Artificial Intelligence: A Modern Approach*, Prentice Hall, 1995.
- [9] Samet, H., "Neighbor Finding Techniques for Images Represented by Quadtrees," *Computer Graphics and Image Processing* 18, 37-57, 1982.
- [10] Samet, H., "An Overview of Quadtrees, Octrees, and Related Hierarchical Data Structures," NATO ASI Series, Vol. F40, 1988.
- [11] Stentz, A., "The Focussed D* Algorithm for Real-Time Replanning," Proceedings of the International Joint Conference on Artificial Intelligence, August 1995.
- [12] Stentz, A., "Optimal and Efficient Path Planning for Partially-Known Environments," Proceedings of the IEEE International Conference on Robotics and Automation, May 1994.
- [13] Stentz A., Hebert, M., "A Complete Navigation System for Goal Acquisition in Unknown Environments," *Autonomous Robots*, 2(2), 1995.
- [14] Stentz, A., "Best Information Planning for Unknown, Uncertain, and Changing Domains," AAAI-97 Workshop on On-line-Search.
- [15] Whitcomb, L. L. and Koditschek, D. E. "Automatic Assembly Planning and Control via Potential Functions," In Proc. IEEE/RSJ International Workshop on Intelligent Robots and Systems. 1991.
- [16] Yahja, A. Stentz, A, Singh, S. and Brumitt, B., "Framed-Quadtree Path Planning for Mobile Robots Operating in Sparse Environments," In Proceedings, IEEE Conference on Robotics and Automation, Leuven, Belgium, May 1998.