3-D Path Planning in a Dynamic Environment Using an Octree and an Artificial Potential Field

Yoshifumi KITAMURA†, Takaaki TANAKA‡ Fumio KISHINO†, and Masahiko YACHIDA‡

† ATR Communication Systems Research Laboratories
2-2 Hikaridai, Seika-cho, Soraku-gun, Kyoto, 619-02 Japan
<kitamura, kishino>@atr-sw.atr.co.jp

‡ Faculty of Engineering Science, Osaka University
1-3 Machikaneyama, Toyonaka, Osaka, 560 Japan
<t-tanaka, yachida>@sys.es.osaka-u.ac.jp

Abstract

We propose an efficient and simple method for finding a collision-free path and orientation for a rigid robot in a dynamic observable 3-D environment for Unmanned Aerial Vehicles (UAVs). The method uses an octree for representing every object (robot and static/dynamic obstacles) in the environment without any distinction of the movability of objects; therefore, all objects are dealt with in the same way. The path of a robot from its starting position to the given goal with arbitrary motion (i.e., translation and rotation) in a 3-D environment is efficiently searched for in successive adjoining regions (octree white cells) by using the potential field generated from each black cell of the octree. The algorithm is simple, so it can easily be accelerated by using parallelization techniques. Experimental results obtained under several conditions, such as a point shaped robot and arbitrarily shaped robots in static/dynamic environments, are reported.

1 Introduction

Unmanned Aerial Vehicles (UAVs) which are autonomous mobile robots such as helicopters in a 3-D environment, are attracting a great deal of attention. They are able to move in more complex environments than the autonomous land vehicle (ALV) in a 2-D environment. In this case, however, they have to control more motion parameters, they cannot remain stationary in air without producing thrust, they are readily subject to disturbances in their environment, and generally, they themselves as well as moving obstacles must move fast. Consequently, a more efficient algorithm capable of finding collision-free paths and orientations in dynamically-changing 3-D environments is necessary for UAVs.

Path planning is a fundamental problem and extensive research efforts have been directed toward this problem [1, 2]. Most reports have concluded that path planning works well in 2-D environments, but at the same time, that it requires a much larger computation time in 3-D environments or environments with more dimensions. The potential field approach [3, 4] is one of the most popular methods for this problem. The main feature of this approach is its scalar potential field which represents both a repulsive force from obstacles and an attractive force to the goal. Therefore, a robot's path from its starting position to the goal is found by threading the valleys of the potential field. The principle is quite simple and gives good results in many cases. However, since local minima of the potential are sometimes produced, the robot is trapped before it can reach the goal in such cases. In addition, the generation of a potential field may require a large computation time in a complicated environment that includes concave objects.

Hierarchical subdivision approaches such as quadtree- or octree-based methods are also popular. They divide the space into regions, and efficiently find a safe path for the robot by searching successive adjoining regions from the starting region to the goal. However, because they present difficulties in representing movable objects, robots or movable obstacles have to be restricted to simple shapes. Moreover, these movable objects are represented in a different manner from the environment, e.g., only points or circles are used for robots [5], primitive shapes [6], or simple polygons [7]. In any case, because the robot and environment are represented in a different manner, the algorithms are not simple, and they have difficulty with dynamic environments.

The algorithm proposed in this paper represents everything in the environment (a robot and static/dynamic obstacles) by using an octree without any distinction of the movability of objects. Therefore, all objects are dealt with in the same way. A collision-free path for a robot with arbitrary motion (i.e., translation and rotation) in a 3-D environment is efficiently searched for by using the potential field generated from each black cell of the octree. Parallel compu-

0-8186-7108-4/95 $4.00 © 1995 IEEE

Proceedings of the International Conference on Intelligent Robots and Systems (IROS '95)
0-8186-7108-4/95 $10.00 © 1995 IEEE

tation is used to generate the potential field, so the method can be used in a dynamically-changing environment. Experimental results obtained under several conditions, for example, arbitrarily shaped robots in static and dynamic environments, are reported.

2 Representation of the Environment

2.1 Octree for a Dynamic Environment

An octree can represent any object's shape by recursively subdividing a region into octants. A tree node is labeled black (white) if it is completely contained within the object (free space); otherwise, the node is labeled gray. Using this environmental representation scheme, a robot path planning problem can be solved efficiently, i.e., a collision-free path can be successfully established by finding the successive adjoining white octree cells from the starting cell to the goal.

Several methods use an octree (a quadtree in a 2-D environment) to represent static obstacles in the environment. [5] proposes a hierarchical path-searching method using a quadtree to represent a static environment. Unfortunately, the shape of the robot is restricted to a circular cross-section, and no dynamic environment including moving obstacles can be employed. [6] proposes an octree-based method, but the environment is limited to a static one and the robot is represented by a collection of primitive shapes such as spheres, cylinders, and so on. [7] also uses a quadtree-based method for a static environment and a simple-shaped robot represented by polygons. These examples show that all existing algorithms using a hierarchical spatial subdivision technique have limitations, i.e., they are difficult to apply to dynamic environments, because the robot or movable obstacles (if any) are restricted to simple shapes. [8] treats the path planning problem among obstacles with a completely known motion. However, the robot shape is degenerated to points. None of them use the same manner to represent the robot and obstacles.

On the other hand, we use an octree to represent every object in the environment; i.e., robots and static/movable obstacles are represented in the same hierarchical shape representation manner. For the motion of objects represented by the octree, a fast algorithm capable of updating the octree after arbitrary rotation and translation [9] is used.

2.2 Artificial Potential Field

Potential field approaches have been well studied as robot path planning methods. In these approaches, both the repulsive force from obstacles and the attractive force to the goal are represented by scalar functions, and they generate the so-called potential field. A robot path from a robot's starting position to the goal can be found by threading the valleys of the potential field [3, 4]. The principle is quite simple and gives good results in many cases. It should be noted, however, that potential field techniques sometimes produce local minima of potential functions, thereby trapping the robot before it can reach the goal. Several proposals to solve this problem have been made. For example, [10] proposes an "oval potential" to reduce the possibility of the appearance of local minima of potentials; however, local minima are easily produced when multiple obstacles exist in the environment. A Laplacian potential [11], on the contrary, does not produce local minima in principle, but solving a Laplacian equation to generate a potential requires an enormous amount of computation.

In our approach, we use the simple potential function employed in [12]. This function uses potential function p_{HA} at point (x, y, z) in a 3-D environment,

$$p_{HA} = \frac{1}{\delta + \sum_{i=1}^{s}(g_i + |g_i|)} \tag{1}$$

where g_i is a linear function that represents the boundary of the convex region (a set of inequalities $\bigcap_{i=1}^{s} g_i \leq 0$ represents the convex region), s is the number of boundary face segments, and δ is a small positive constant. p_{HA} has its maximum value of $1/\delta$ inside the region and decreases as the inverse of the distance outside the region. This potential function (1) demands that the region be convex. Otherwise, the concave region must be modeled as a collection of convex pieces. This means that the regions in the environment eventually become a collection of many different shaped concave regions each having a different number of boundary face segments. As a result, designing an efficient algorithm to compute the potential field is difficult; acceleration by parallelization in paticular is not easy.

We generate a potential for each cell of the octree using equation (1) (therefore, $s = 6$ always). Since each cell is a cube, the regions are guaranteed to be convex and of the same shape; therefore, the potential field can be calculated in quite a simple way and this computation can easily be accelerated by parallelization. The potential at any point in the environment is given by the maximum of the potentials due to individual cells. This prevents local maxima of the potential from appearing in free space.

3 Path Planning Algorithm

A path planning algorithm is described as using an octree to represent every object (robot and static/dynamic obstacles) in the environment, and a potential field is described as a heuristic cost function to search for a path.

3.1 Renewal of a Dynamic Environment Represented by an Octree

In the following, we assume that for each object in the environment the octree representation and its position and orientation are known. The methods for constructing the octree from sensor information (such as images) are described in [13]. If any motion parameters of objects (robot and obstacles) are observed, the octree representation for each object is updated. Note that an efficient and parallel algorithm capable of updating the octree of a three-dimensional object for arbitrary rotation and translation is described in [9]. The basic algorithm for this is to apply a motion

Proceedings of the International Conference on Intelligent Robots and Systems (IROS '95)
0-8186-7108-4/95 $10.00 © 1995 IEEE

transformation matrix to each black cube in the octree to be moved (called the source octree) separately (for a series of motions, the same source octree is always used and only the motion matrix is changed; this prevents digitization errors from continually increasing) and to test recursively, starting from the largest cube, for intersections between the transformed black cubes and the standard, upright (i.e., faces parallel to the standard euclidean coordinate axes) cubes of the new octree to be created (called the target octree). Standard cubes in the target octree are labeled white, gray, or black depending on whether they do not intersect with a transformed black cube, intersect partially with a transformed black cube, or are completely inside a transformed black cube. After a transformed source black node is tested for intersection with a target octree gray node, the gray node is tested to see if its eight children are all labeled black or all labeled white; if so, then the children are erased and the gray node is labeled identical to how the children were (this is called condensing the octree).

3.2 Generation of Potential Field

Both the repulsive force from obstacles and the attractive force to the goal are represented by a scalar potential field. The simple potential function employed in [12] is basically used for the repulsive force. In this function, if the unit generating the potential is in a convex region, the computation of the potential field is quite easy. In this case, we can compute the potential field of the environment by integrating the potentials generated by each octree black cell of the obstacles.

We use equation (2) which represents the repulsive force at point $A(x, y, z)$ from $B_{i,j}$(the black cell numbered i of obstacle O_j).

$$p_{i,j} = \frac{p_{max}}{1+g} \tag{2}$$

Here,

$$\begin{aligned} g(x,y,z) &= (x_0 - l/2 - x) + |x_0 - l/2 - x| \\ &+(x - x_0 - l/2 + 1) + |x - x_0 - l/2 + 1| \\ &+(y_0 - l/2 - y) + |y_0 - l/2 - y| \\ &+(y - y_0 - l/2 + 1) + |y - y_0 - l/2 + 1| \\ &+(z_0 - l/2 - z) + |z_0 - l/2 - z| \\ &+(z - z_0 - l/2 + 1) + |z - z_0 - l/2 + 1| \end{aligned} \tag{3}$$

where

p_{max} : the maximum potential
(x_0, y_0, z_0) : the center coordinate of $B_{i,j}$
l : the side length of $B_{i,j}$

This function is maximum inside $B_{i,j}$ and decreases monotonously as the distance from $B_{i,j}$ increases.

The potential at point A in the environment is given by the maximum of the potentials due to individual cells as follows.

$$P_o = Max\{p_{i,j}\} \quad (1 \le i \le m,\ 1 \le j \le n) \tag{4}$$

where m is the number of octree black cells of obstacle O_j, and n is the number of obstacles in the environment. This prevents local maxima of the potential from appearing in free space. (If "sum" is used instead of "max", local maxima will easily appear.)

Finally, the attractive force generated by the goal is represented by equation (5) which grows linearly as the distance from the goal point increases.

$$P_g = C\sqrt{|x - x_g|^2 + |y - y_g|^2 + |z - z_g|^2} \tag{5}$$

where $G(x_g, y_g, z_g)$ is the position of the goal, and C is a constant. The potential in the environment is represented by the sum of P_o and P_g in this paper, i.e.

$$P = P_o + P_g. \tag{6}$$

We compute the above potential field using parallel implementation. Before the parallel algorithm is run, a precomputation step is run to divide the black cubes of all obstacles in the environment evenly among the processors. In other words, if there are N processors, then processor i will receive the i, $i+N$, $i+2N$, ... black cubes. After the precomputation step, each processor runs the above algorithm of equation (2) on the assigned cubes and computes the potential. After all processors compute the individual potentials, one of the processors creates the potentials using equation (4).

3.3 Exploring the Path

The path of a robot with arbitrary motion (i.e., translation and rotation) from its initial position to the goal in a 3-D environment is efficiently searched for in the successive adjoining regions by using the heuristic cost functions. The average overall points in each cell are used for the heuristic cost of this octree cell. For static environments, the best-first search method is used, and the robot path is searched for from the robot's starting (initial) position to the goal. For dynamic environments, the hill-climbing method is used to search for the current path from the current cell to its adjoining cell at each processing cycle, and then the next path is searched for. This process is repeated. Figure 1 shows the control flow of our proposed path planning method in a dynamic environment.

3.4 Rotation and Translation of an Arbitrarily Shaped Robot

An arbitrarily shaped robot represented by an octree can be used. This robot can go through narrow corridors by rotating its body even though it can not carry out search with only translation. For this purpose, a heuristic cost function is used for the robot's rotation, i.e., the total potential on the robot calculated by integrating the potential over the region of black octree cells of the robot. Here, translation is given a higher priority than rotation, allowing the robot to attempt to find a collision-free path with only translation at first. If this search fails, however, rotation is examined at one of the leaf nodes of the search tree. At this point, the direction minimizing the above cost

Proceedings of the International Conference on Intelligent Robots and Systems (IROS '95)
0-8186-7108-4/95 $10.00 © 1995 IEEE

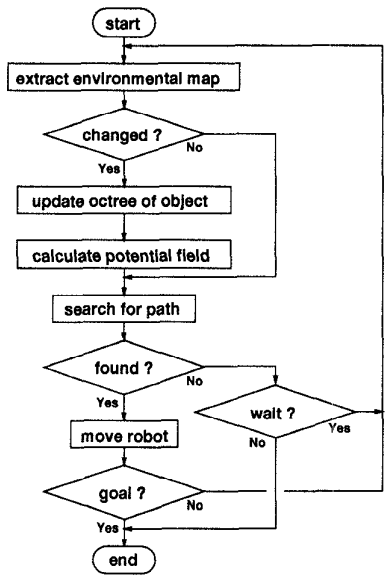


Figure 1: Control flow of the proposed path planning in a dynamic environment

for rotation is determined. To speed up the method, the cost for rotation α degrees apart is calculated, and then the direction having the minimum cost is chosen. For arbitrary rotation and translation of the robot represented by the octree, the same algorithm as described in 3.1 for updating the environment is used.

4 Experiments

The experiments involved an arbitrarily shaped robot in static and dynamic environments. For clear description, the results obtained from 2-D environments using a quadtree are explained first. Then, the results obtained from a 3-D environment are given.

4.1 Examples in a 2-D Environment

Before describing the results from the arbitrarily shaped robot, the basic performance of the proposed method is evaluated using a point robot.

4.1.1 A Point Robot in a Static Configuration

A robot represented as a simple point robot was tested in three different static configurations to evaluate the performance of the proposed method. For each given configuration, the potential field was calculated first. Figure 2 shows a perspective view of the generated potential field for one of the configurations (Configuration 2 in Figure 3).

Figure 3 shows the experimental results for given starting and goal points. The gray levels of each cell show the potential of the cell in these figures, i.e., a darker cell has a higher potential, and black cells excluding starting and goal cells represent obstacles. Each robot path was searched for by the best-first search method using the potential for the heuristic cost function. The cells with the • mark at their centers represent the searched cells. The line connecting these points shows the established path.

For each configuration, we compare our proposed method using quadtree representation (figures (a)(b)(c)) against a subdivision method using uniform grids (figures (d) (e) (f)). Here a level-6 quadtree is used, and the size of the uniform grids is equal to the size of a level-5 quadtree cell. Table 1 simply compares the number of nodes explored for each method. This table shows that a uniform grid-based method can find more complicated paths in more computation steps than a quadtree-based method for every configuration.

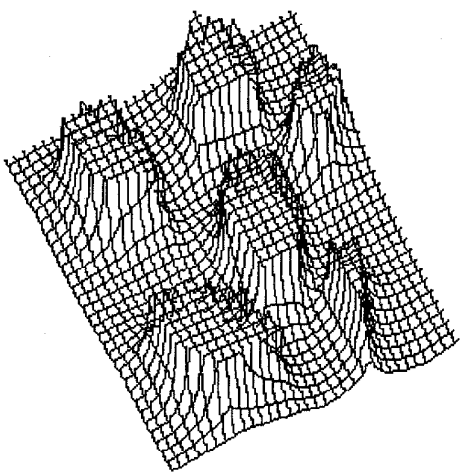


Figure 2: Generated potential field for Configuration 2 in Figure 3.

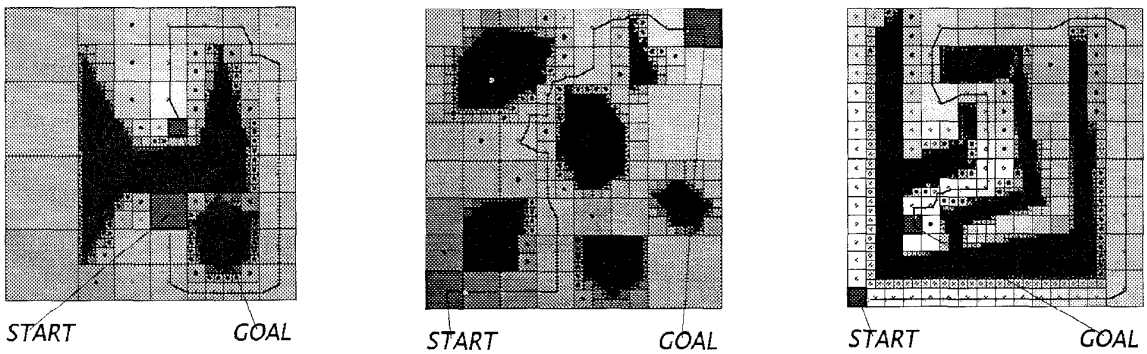
Table 1: Comparison of the number of nodes explored

Configuration	proposed method (quadtree)	compared method (uniform grid)
1	90	334
2	91	153
3	296	637

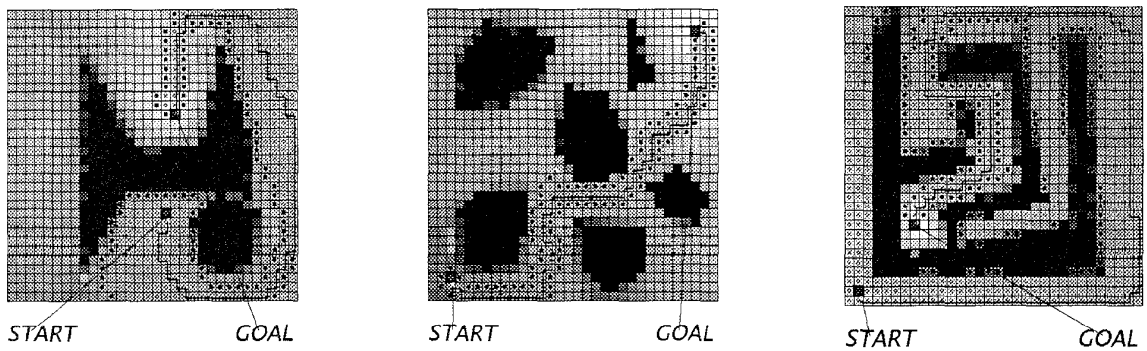
4.1.2 A Robot in a Static Environment

An arbitrarily shaped robot was tested in a static 2-D environment. The robot is shown in Figure 4. In this figure, the robot's shape represented by a quadtree produced by rotating the shape 30 degrees at a time is shown.

Proceedings of the International Conference on Intelligent Robots and Systems (IROS '95)
0-8186-7108-4/95 $10.00 © 1995 IEEE



(a) Configuration 1 (quadtree) (b) Configuration 2 (quadtree) (c) Configuration 3 (quadtree)



(d) Configuration 1 (grid) (e) Configuration 2 (grid) (f) Configuration 3 (grid)

Figure 3: Comparison of experimental results for a point robot in static configurations represented by quadtrees (upper row) and uniform grids (lower row)

For a given environment, the potential field was calculated first. Figure 5 shows a perspective view of the generated potential field for the environment. Figure 6 shows the experimental result for given starting and goal points. The gray levels of each cell show the potential of the cell in this figure, i.e., a darker cell has a higher potential, and black cells excluding starting and goal cells represent obstacles. The robot path was searched for by the best-first search method using the potential for the heuristic cost function. The cells with the • mark at their center represent the searched cells. The line connecting these points shows the established path. At the entrance of narrow corridors, the robot rotated its body enabling a collision-free path and the robot's directions to be found.

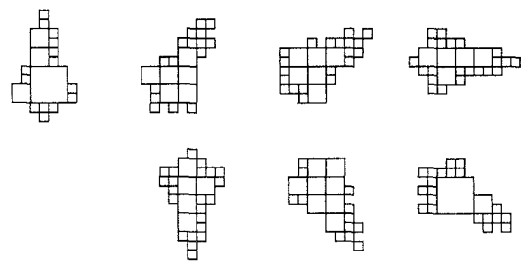


Figure 4: An experimental robot represented by a quadtree for figure 6. The upper left one is the original shape, and the others are rotated shapes of the robot.

4.1.3 A Robot in a Dynamic Environment

Another arbitrarily shaped robot (shown in Figure 7) was tested in a dynamic environment. Figures 8 and 9 show the experimental results for given starting and goal points. The gray levels of each cell show the potential of the cell in these figures, i.e., a darker cell has a higher potential, and black cells excluding starting and goal cells represent obstacles. The robot path was searched for by the hill-climbing method using the potential for the heuristic cost function. The cells with the • mark at their center represent the searched cells. The robot loitered on the left side of a large obstacle as shown in figure (a), because no path existed leading to the goal in this environment. But after the obstacle was moved, the robot could find a path to the goal as

Proceedings of the International Conference on Intelligent Robots and Systems (IROS '95)
0-8186-7108-4/95 $10.00 © 1995 IEEE

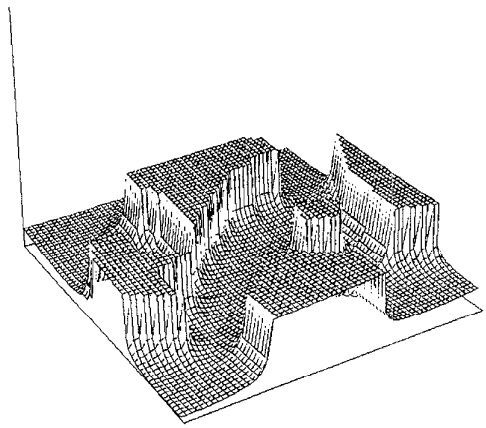


Figure 5: Generated potential field for the given environment in 4.1.2.

shown in figure (b). The line connecting these points shows the established path.

4.2 A Robot in a 3-D Environment

An arbitrarily shaped robot represented by an octree was tested in a 3-D environment. The robot (space shuttle) is shown in Figure 10 along with a rotated shape. For a given environment, the potential field was calculated first. Figure 11 shows the experimental result for given starting and goal points. The robot path was searched for by the best-first search method using the potential for the heuristic cost function. The line shows the established path. At the entrance of narrow corridors, the robot rotated its body enabling a collision-free path and the robot's directions to be found.

5 Summary and Conclusion

We proposed an efficient and simple method for finding a robot's collision-free path and orientation in a dynamic observable environment. The method uses an octree to represent every object (rigid robot and static/dynamic obstacles) in the environment without any distinction of the movability of objects; therefore, all objects are dealt with in the same way. The path of the robot with arbitrary motion (i.e., translation and rotation) from its starting position to the given goal in a 2-D and 3-D environment was efficiently searched for in the successive adjoining regions (quadtree or octree white cells) by using the potential field generated from each black cell of the octree.

Experimental results obtained under several conditions, such as a point shaped robot and arbitrarily shaped robots in static/dynamic environments, were reported. In the proposed algorithm, both the step of updating environments and robot position, and the step of generating potentials are quite simple; therefore, acceleration by parallelization was easily achieved. In addition, every object in the environment was dealt with in the same way, i.e., the octree was used to represent every object without any distinction of the movability of objects. Therefore, a multiple movers' problem can be easily solved by the method.

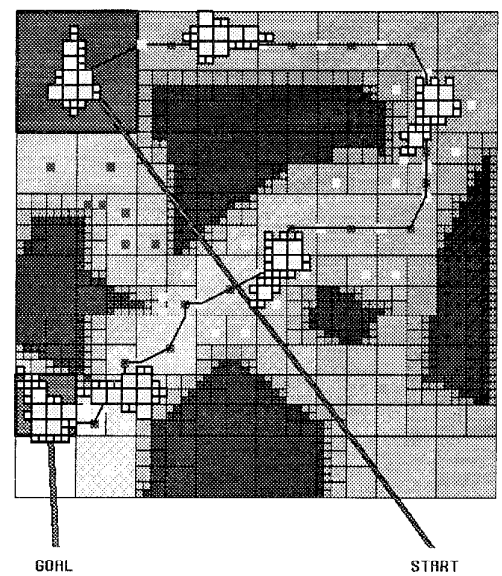


Figure 6: Experimental result of an arbitrarily shaped robot in the static environment of Figure 5.

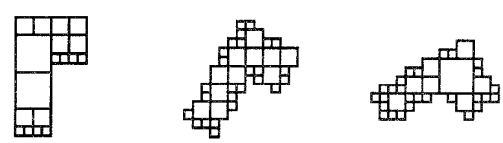


Figure 7: An experimental robot represented by a quadtree for Figures 8 and 9 with rotated shapes.

Acknowledgements

The authors would like to thank Prof. Narendra Ahuja of the University of Illinois at Urbana-Champaign, USA for his useful discussions.

References

[1] Latombe, J.C. *Robot motion planning.* Kluwer Academic Publishers, Boston, 1991.

[2] Hwang, Yong K. and Ahuja, Narendra. Gross motion planning – a survey. *ACM Computing Surveys*, Vol. 24, No. 3, pp. 219–291, 1992.

[3] Miyazaki, F. and Arimoto, S. Sensory feedback for robot manipulators. *Robotic Systems*, Vol. 2, No. 1, pp. 53–72, 1985.

Proceedings of the International Conference on Intelligent Robots and Systems (IROS '95)
0-8186-7108-4/95 $10.00 © 1995 IEEE

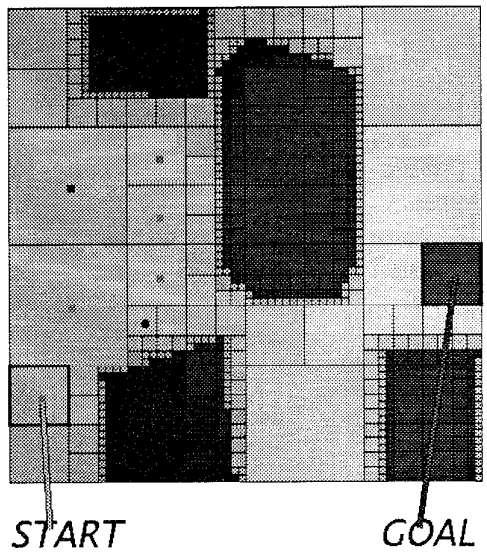


Figure 8: Experimental result of an arbitrarily shaped robot in a dynamic environment — before movement of an obstacle

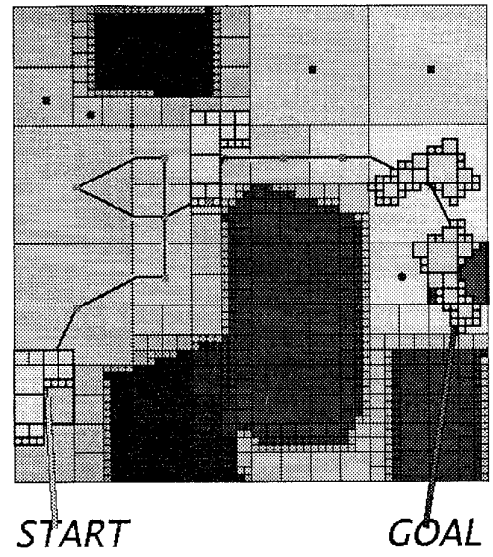


Figure 9: Experimental result of an arbitrarily shaped robot in a dynamic environment — after movement of an obstacle

[4] Khatib, Oussama. Real-time obstacle avoidance for manipulators and mobile robots. *International Journal of Robotics Research*, Vol. 5, No. 1, pp. 90–98, 1986.

[5] Kambhampati, Subbarao and Davis, Larry S. Multiresolution path planning for mobile robots. *Journal of Robotics and Automation*, Vol. RA-2, No. 3, pp. 135–145, 1986.

[6] Herman, Martin. Fast, three-dimensional, collision-free motion planning. In *International Conference on Robotics and Automation*, pp. 1056–1063. IEEE, 1986.

[7] Noborio, H., Naniwa, T., and Arimoto, S. A feasible motion-planning algorithm for a mobile robot on a quadtree representation. In *International Conference on Robotics and Automation*, pp. 327–332. IEEE, 1989.

[8] Fujimura, Kikuo and Samet, Hanan. A hierarchical strategy for path planning among moving obstacles. *Trans. on Robotics and Automation*, Vol. 5, No. 1, pp. 61–69, 1989.

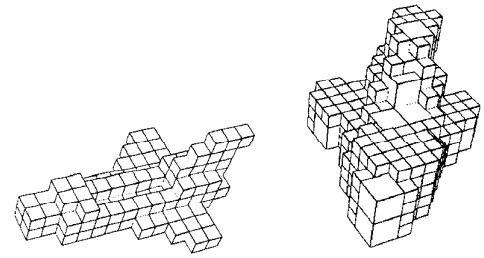
[9] Smith, A., Kitamura, Y., and Kishino, F. Efficient algorithms for octree motion. In *IAPR Workshop on Machine Vision Applications*, pp. 172–177, 1994.

[10] Okutomi, M. and Mori, M. Decision of robot movement by means of a potential field. *Advanced Robotics*, Vol. 1, No. 2, pp. 131–141, 1986.

[11] Connolly, C.I. and Burns, J.B. Path planning using Laplace's equation. In *International Conference on Robotics and Automation*, pp. 2102–2106. IEEE, 1990.

[12] Hwang, Yong K. and Ahuja, Narendra. A potential field approach to path planning. *IEEE Transactions on Robotics and Automation*, Vol. 8, No. 1, pp. 23–32, 1992.

[13] Chen, Homer H. and Huang, Thomas S. A survey of construction and manipulation of octrees. *Computer Vision, Graphics, and Image Processing*, Vol. 43, No. 3, pp. 409–431, 1988.



(a) given robot (b) a rotated robot

Figure 10: Experimental space shuttle in a 3-D environment for Figure 11

Proceedings of the International Conference on Intelligent Robots and Systems (IROS '95)
0-8186-7108-4/95 $10.00 © 1995 IEEE

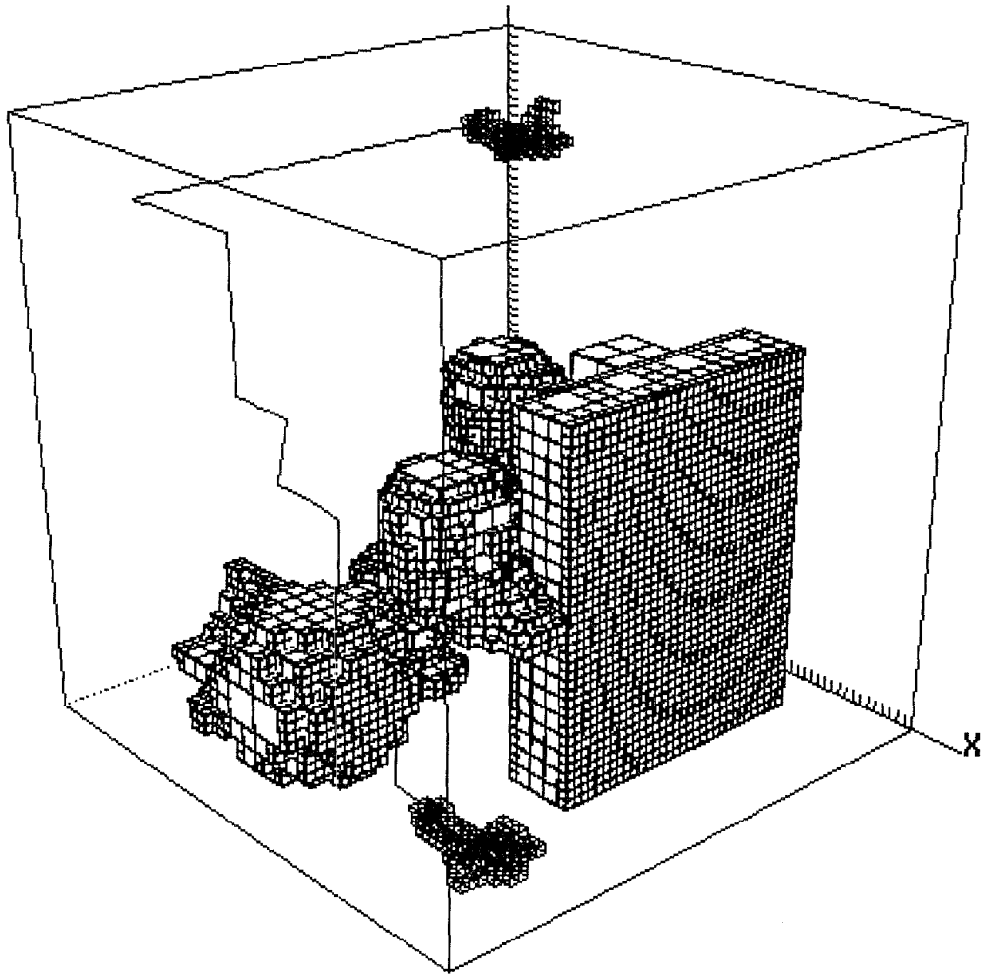


Figure 11: Experimental result of an arbitrarily shaped robot in a 3-D environment

Proceedings of the International Conference on Intelligent Robots and Systems (IROS '95)
0-8186-7108-4/95 $10.00 © 1995 IEEE