Robotica (2002) volume 20, pp. 281–290. © 2002 Cambridge University Press
DOI: 10.1017/S0263574701004003 Printed in the United Kingdom

A heuristic approach for local path planning in 3D environments

A. Elnagar

Department of Computer Science, University of Sharjah, P. O. Box 27272, Sharjah, U.A.E.

(Received in Final Form: November 12, 2001)

SUMMARY

In this paper we describe a heuristic potential based technique for solving the problem of path planning based on local information for a free flying robot (FFR) in a 3D static environment. A path which maximizes the sum of three functions: (repulsion from obstacles, attraction towards the goal, and level attraction) is chosen. While the repulsive and goal-attraction potentials depend on how close is the FFR from obstacles and goal, respectively, the level-attraction potential maintains a certain altitude at which the FFR should navigate. Two additional heuristics are proposed to overcome the local minima problem and to improve the efficiency of the search process. Simulation results of the proposed technique are presented.

KEYWORDS: Trajectory planning; Obstacle avoidance; 3D static environments.

1. INTRODUCTION

Until recently, the problem of robot motion planning is studied mostly in 2D static environments. Algorithms proposed are categorized as either being global or local. While a global method is one that assumes complete knowledge about its environment, a local method, on the other hand, assumes partial knowledge. With the availability of the extensive literature on this subject, researchers start to look at the problem in a more general and realistic form such as navigation in 3D and time-varying environments.

This paper aims at studying the motion planning problem in a 3D static environment as a step before extending the work to the time varying domain. We assume that a free flying robot (FFR) is equipped with sensors that provide local information about its environment. We propose a planning algorithm that will generate a free path, if one exists, in a 3D environment cluttered with obstacles.

Exact path planning algorithms require a path to be found if one exists. This can often be very difficult to accomplish even when a complete and precise description of the surroundings is available. Various researchers have proved that most exact path planning problems are intractable, unless the environment or the planning task is extremely simple. Heuristic strategies, on the other hand, do not guarantee finding a free path even if one exists. However, these methods have significantly lower time complexity.

The algorithm introduced in this paper is a heuristic strategy based on local information. There are two main factors that control our strategy: how far is the FFR from obstacles and how close it is to the goal. A third factor that controls navigation in 2D (i.e. at a certain altitude in 3D) is also taken into account to be used in certain applications. These factors are henceforth referred to as obstacle repulsion, goal attraction, and level attraction, respectively. Obstacle repulsion is a function of the distance from visible obstacles and goal attraction is proportional to the distance from the goal (i.e. attracts the FFR towards the goal). This is to ensure that a path will get closer to the goal in the planning process. Level attraction, on the other hand, is a function used to keep the robot at a certain altitude in a 3D environment or as close as possible within a pre-set range. For example, a robot on which a camera is mounted is sent inside a tank for exploring its interior surfaces by grabbing a sequence of images for inspection at a later stage.

In order to solve the problem of finding collision-free paths based on local information efficiently, it is often necessary to use heuristics. Thus, we introduce two special-purpose heuristics: the "follow-boundary" and the "avoid-region". Both heuristics are intended to prune the search space and to guide the FFR towards its destination while avoiding collisions and any possible local minima.

The next section briefly describes work in the area of path planning. Section 3 introduces the notation used and defines the problem. The potential field functions are described in Section 4. Heuristics for obstacle avoidance are introduced in Section 5. Section 6 describes the algorithm for planning a local path and, finally, simulation examples are given in Section 7.

2. RELATED WORK

Several researchers have described algorithms for robot motion planning in static environments. For a recent survey on this subject and a comprehensive study of the different methodologies on robot motion planning, see references [1] and [2], respectively.

There are few studies reported on the problem of motion planning in dynamic environments. Most of these works are just extensions of existing methods in static domains to time-varying environments.[3–7]

Although there exists an extensive literature on the motion planning problem in 2D static environments, very few attempts are reported in the 3D case. Canny[8] introduced the silhouette curves method that can be used theoretically in an N-dimension static environment. The cell decomposition technique is extended to 3D by representing the environment in an octree data structure.[2,9] Both methods require complete knowledge of the environment. Kutolakos et al.[10] proposed a general motion planning algorithm in a 3D unknown static environment after incorporating vision.

https://doi.org/10.1017/S0263574701004003 Published online by Cambridge University Press

This algorithm was based on the work presented in reference [11].

Extensions of the potential field approach, originally proposed by Khatib,[12] into 3D static domains are carried out independently by Miyazaki and Arimoto,[13] and Pavlov and Vornin.[14] Rimon and Koditschek[15] proposed an algorithm that enables a point-robot to navigate among spherical obstacles using potential functions. Later, they generalized their algorithm to work with star-shaped obstacles.[16] Another potential field behavior-based planning algorithm among spherical obstacles is described in reference [17]. Hwang and Ahuja[18] proposed an algorithm that uses potential functions to construct a connectivity graph of the free space which is searched at a later stage for a free path between start and goal positions. Unlike most potential field techniques, this one requires global knowledge of its environment.

Jun and Shin[19] proposed a probabilistic-based method which require global knowledge of the environment. A bi-directional search (search from both start and goal) based on the probabilities and path length is performed until a path is found.

Some fundamental questions related to the computational complexity of various formulations of the robot motion problem are addressed in references [20–24].

We propose a potential field based technique that uses heuristics to navigate in a 3D environment, not restricted to planar ground-based vehicles, cluttered with polyhedral obstacles. Heuristics are used to overcome the local minima problem and prune out the search space. In addition, we demonstrate how this algorithm can be applied in a real life situation.

3. NOTATION AND BASIC DEFINITIONS

Before introducing the notation and formal definitions, we state the problems that we would like to solve in this paper, as follows:

Problem 1: FFR has to be continuously moving from p_s (the starting point) to p_g (the goal) in a 3D space cluttered with static obstacles. At any point on its path, the FFR knows only its current coordinates and those of the goal. Nothing is known about the obstacles outside a certain visible region. Our objective is to make the FFR find a free path if one exists, or report failure otherwise.

Problem 2 (application of Problem 1): A FFR is placed inside a 3D contained environment such as a tank (or a pipe) to examine or collect intensity images of its interior surfaces. At any point on its path, the FFR knows only its current coordinates and the dimensions of the environment. The robot has to maintain a specific fixed distance from the boundary it is inspecting and to navigate at a pre-set altitude ($\hat{h}$) in 3D. The environment may contain obstacles. Our objective is to make the FFR navigate under these constraints in order to successfully collect images of the interior surfaces of its environment, if possible.

To introduce the problems formally, consider W to be the workspace of all possible positions in which the FFR may be placed. W is a subset of the Euclidean space $\Re^3$. F_W is a fixed frame of reference embedded in W. Similarly F_{FFR} is the cartesian frame of FFR. At any time, we shall consider the FFR (spherical/cubic object) as a point in W that can be represented by its coordinates with respect to F_W. The initial and goal positions in W are denoted by p_s and p_g, respectively. $\hat{p}$ defines the FFR position in W.

Definition 3.1. *A configuration q of* FFR *is a specification of the cartesian coordinates* (x, y, z) *in* W.

Definition 3.2. *Let $O_0, O_1, \ldots, O_{n-1}$ be fixed static polyhedral objects distributed in W. Each O_i represents a subset of W and is described by a set of features. A feature is either a face, an edge, or a vertex of a an obstacle.*

The distance from a point to any obstacle is the distance from that point to the closest feature of that obstacle. Given a fixed point and an obstacle, the closest feature (of this obstacle) to this point is the feature with the least distance from this point. The distance from a point to a feature is defined as the Euclidean distance between that point and the closest point on the feature.

Definition 3.3. *Visibility field (VF) specifies the visibility range of the FFR which is defined as a sphere with grid uniformly distributed on its outer surface. The number of grid points is determined by the number of circles that construct the sphere along x, y, and z axes. Reachability field (RF) is a subset of the visibility field, which determines the grid points that are reachable by the FFR from its current position.*

Two grid densities, normal and high, can be used. By normal density, the grid points are distributed on the three circles which represent the boundary of the sphere around the center (each circle is parallel to a plane (Figure 1)). Eight points are set on each circle to correspond to the basic directions on a compass (i.e. N, NW, W, SW, S, SE, E, NE). On the other hand, high density grid is distributed on the same three circles and some more circles, which are created by connecting the diagonal points from the original three

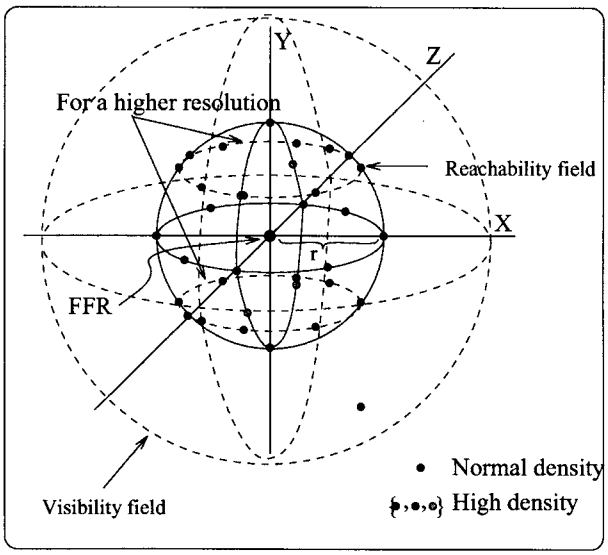


Fig. 1. Visibility and reachability fields.

https://doi.org/10.1017/S0263574701004003 Published online by Cambridge University Press

circles (each new circle is parallel to one of the coordinate planes). The same eight directions are set on each circle as potential grid points. Duplicate points are not considered. More grid points can be easily added by considering more circles along the reachability field (Figure 1).

Definition 3.4. *The local free workspace along the generated path is a subset of W given by*:

$$W_{free}^{local} = \{RF\}_{i \in I} \quad I = \{0, 1, \ldots, k-1\}$$

where k is the number of RFs used to plan a path from p_s *to* p_g.

Definition 3.5. *A free path followed by the FFR from* p_s *to* p_g *is a continuous mapping*

$$T\colon [0, t_f] \rightarrow W_{free}^{local},$$

where $T(0) = p_s$ *and* $T(t_f) = p_g$.

The free space is determined by the set of grid points which the FFR can move to without colliding with obstacles along its path. A free path is a path that resides entirely in the free space. Notice that all grid points in the reachability field are referred to as adjacent points.

Definition 3.6. *A local minima* (*maxima*) *is a point* ($q \in W_{free}^{local}$) *that has a potential value lower* (*higher*) *than all potential values of its adjacent points potentials. A local minima is also called a dead-end point.*

4. POTENTIAL FUNCTIONS

A potential field is one that takes into account certain factors to determine the best next configuration for the FFR to be in. The robot tries to reach the goal by moving in directions that minimize the value of the potential field function. Most potential field functions are constructed by a repulsive and attractive potentials. The repulsive potential is chosen so that it increases towards infinity as the FFR approaches an obstacle and decreases otherwise. The attractive potential, on the other hand, is constructed with a global minimum at the goal and rising as the FFR moves further from the goal. While the repulsive potential prevents collisions, the attractive potential causes the robot to move towards the goal. A third potential function that enables the FFR to maintain a certain altitude (specified by the user) in 3D is introduced to be used in inspection tasks.

4.1 General representation

There is a variety of potential field functions that may be used. We choose a simple analytical function that combines a repulsive potential function (R), similar to the one reported in [1], a goal attraction function (A), and a level-attraction function (C). We define the total potential cost F at a point q to be

$$F(q) = w_r R(q) + w_a A(q) + w_c C(q) \tag{1}$$

The constants w_r, w_a, and w_c are the relative weights for controlling the dominant potential(s) along the path. These constants control the path length, clearance from obstacles, and preference to navigate at a certain altitude (2D plane) as far as possible, respectively. The attraction potential is the force drawing the FFR towards the goal. It is defined in terms of the Euclidean distance between a 3D point (q) and the goal coordinates (p_g) as:

$$A(q) = a\, D(q, p_g) \tag{2}$$

where a is a constant that controls the magnitude of the attraction potential generated with respect to the other potential functions. A simple difference relation used to maintain navigation at a certain altitude in 3D is:

$$C(q) = c\,|H(q) - H(p_g)| \tag{3}$$

where $H(q)$ is the height of that point with respect to the frame of reference (e.g. y-component of q in Figure 2), and c is a positive constant that has similar function as a in Equation 2. The repulsive potential at a point q is given by the maximum potential produced by each obstacle within the visibility field. Formally,

$$R(q) = \max_i r_i(q) \qquad 0 \le i \le n-1 \tag{4}$$

where n is the number of obstacles detected in the visibility field. The repulsive potential induced by each obstacle is

$$r_i(q) = \begin{cases} 0 & \text{if } D(O_i, q) > \mu \\ \dfrac{1}{D(O_i, q) + \delta} - \dfrac{1}{\mu + \delta} & \text{otherwise} \end{cases} \tag{5}$$

where δ is a very small positive number, and μ is a threshold value that refer to the distance beyond which obstacles have no effect on the repulsive potential.

$$\mu = \min(\xi, D(q, p_g))$$

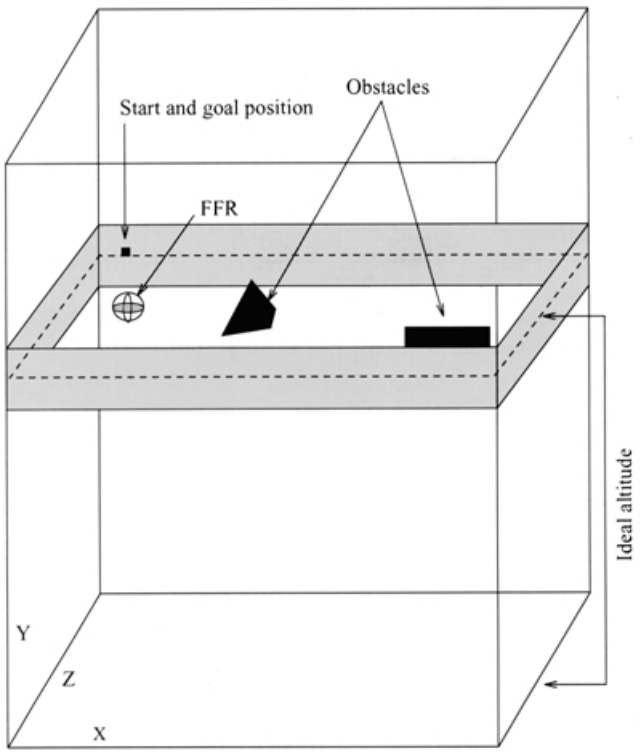


Fig. 2. Navigation in a 3D static environment.

https://doi.org/10.1017/S0263574701004003 Published online by Cambridge University Press

ξ is a value preset by the user. $D(O_i, q)$ is the distance between any obstacle (O_i) and the point q. It is computed by

$$D(O_i, q) = \min_j D(f_j, q), \qquad 1 \le j \le m \tag{6}$$

where m is the total number of features (i.e. surfaces, edges, and vertices) that define the obstacle (O_i), and

$$D(f_j, q) = \begin{cases} |p - q| & \text{if } f_j = \{p\}, \text{ where } p \text{ is a vertex} \\ \dfrac{|p_1\vec{p}_2 \times p_1\vec{q}|}{|p_1\vec{q}|} & \text{if } f_j \text{ is an edge } (p_1\vec{p}_2) \\ |(p - q)\dfrac{1}{|n|}n| & \text{if } f_j \text{ is a plane defined by normal } n \text{ and a point } p \end{cases} \tag{7}$$

The distance check above is performed only on the closest feature of an obstacle. To determine the closest feature, we use a similar algorithm as in reference [25]. The algorithm keeps track of a guess to what the closest feature is. Originally the guess is chosen arbitrarily to be the first face. Several geometric tests are applied on the closest feature and the position of FFR. Each test either confirms the guess as the closest feature or returns a closer feature. In the latter case, the guess becomes the closer feature and the geometric tests are re-applied again until the closest feature is found. All geometric tests work by specifying a plane for each feature of the obstacle that is physically connected to the current guessed feature (e.g. a face is connected to the edges bordering it). FFR must be in the region bounded by these planes for the guess to be the closest feature. If FFR is not in the region, the feature which falls between FFR and this region is closer and, therefore, the closer feature becomes the next guess and the tests are re-applied. If the guess is a vertex then the planes are the ones that contain this vertex and are perpendicular to one of the edges. If the guess is a line, two of these planes are perpendicular to the line and contain at least one endpoint. The other two planes contain the line and are perpendicular to a face bounded by the line (geometrically there must be exactly two faces bounded by the line). If the guess is a plane, two tests are applied. The first one determines if the point is on the same side of the plane containing the face as the obstacle. If so, another face is selected which does not have this problem (if such face is not found, then the FFR is inside the obstacle). For the second test, a plane is constructed for each edge bordering the guessed face and the normal geometric test is applied.

4.2. Requirements for heuristics

Heuristics are used to clear obstacles in situations where the FFR gets stuck in a point in front of an obstacle (i.e. local minima). There are two heuristics: "follow-boundary" and "avoid-region". The follow-boundary heuristic is invoked whenever the robot gets stuck in a dead-end position. It attempts to move the FFR in a certain direction (to be chosen randomly) to clear that obstacle rather than trying to reach the goal. In other words, instead of navigating in 3D, the FFR navigates in a 2D environment. This eliminates the need for the height potential function $C(q)$ and therefore we set $w_c = 0$. As a result of the fact that the goal point is temporarily ignored, $\mu = \xi$. Note that the objective of this heuristic is not reaching the goal but rather clearing the obstacle. Therefore, we choose the attraction potential to be:

$$A(q) = a\ FD(q) \tag{8}$$

where FD is the forward distance (i.e. radius (r) of the reachability field), Figure 1. This means moving in a direction to increase $A(q)$ brings the FFR further in this direction to clear the obstacle. The total potential cost at a point (q) is determined by Equation 1 but with the weighting factors w_a, w_r, and w_c are set to –1, 1, and 0, respectively. As for the avoid-region heuristic, no further changes are necessary.

4.3. Requirements for the inspection application

This section deals specifically with Problem 2. Recall that our intention is to make the FFR collect data about the inner surfaces of a tank, for example, for inspection purposes. We consider a convex polygonal environment that is described by a number of surfaces, S. Unless otherwise specified, all definitions introduced so far remain the same with the exception that μ is replaced by ξ. Each weight factor is set to $\frac{1}{3}$. If S_i denotes the side i of the tank, and $\hat{h}$ is the ideal altitude at which the FFR should navigate (specified by the user), we define S_{i+1} as:

$$S_{i+1} = S_{(i \odot S)+1},\ 0 \le i \le S$$

where $\odot$ is the modulus function. The total potential cost at a point q when following wall S_i is:

$$F(q, S_i) = w_r R(q) + w_a A(q, S_i) + w_c C(q) \tag{9}$$

The distance of the point q to S_i is defined as the perpendicular distance ($D(q, S_i)$). The attraction potential which draws the FFR towards the next wall of the tank is given by:

$$A(q, S_i) = a\ D(q, S_{i+1}) \tag{10}$$

The potential function that is responsible for keeping the robot at altitude $\hat{h}$ is:

$$C(q) = c\,|h(q) - \hat{h}| \tag{11}$$

5. HEURISTICS FOR OBSTACLE AVOIDANCE

The problem of local minima is considered to be the major drawback of potential field-based techniques. A local minima is a point where the FFR can get stuck. If the algorithm is merely based on artificial potential functions then there is no guarantee it will escape any local minima encountered.* This is simply because the repulsion force produced by all obstacles balances the attraction force produced from the goal. When the level potential is accounted for, all three potentials could also result in a local minima when added together.

The basic idea to escape a local minima is to back up to the point reached previously and to move in the best direction (i.e. the one that leads to the point with the

* See reference [2] for more details on this subject.

https://doi.org/10.1017/S0263574701004003 Published online by Cambridge University Press

maximum value of potential field) skipping all previously visited directions. When the only directions left from a point are the ones with lower potential cost, that point is then considered as another local minima and the FFR has to back up one more step. We propose two heuristics to make the FFR capable of escaping a local minima and, further, to reduce the search space.

5.1. Follow-boundary heuristic

The main algorithm for generating a path in a 3D environment based on the theory introduced so far, works well for eliminating any dead-end produced whenever the FFR faces either an edge or a vertex of an obstacle. However, to escape a local minima produced by a face of a large obstacle that is too large to clear, we introduce the follow-boundary heuristic. This heuristic is invoked whenever the closest feature of an obstacle is a face and released when this feature changes. In order to escape a local minima, a random direction, one out of four, is chosen at this position (Figure 3). The algorithm for randomly choosing a direction is explained as follows: construct two unit vectors: v_1 and v_2. v_1 is perpendicular to the normal (n) of the obstacle's face that is causing the local minima, whereas v_2 is set to be the normalized vector that is perpendicular to both v_1 and n. v_1 and v_2 form the basis of a two dimensional grid that is parallel to the closest face (Figure 3(A)). A direction between 0 and 3 is chosen at random and accordingly an ab-grid is defined as in Figure 3(B). The FFR moves in the direction of a until the feature changes or another local minima is encountered. In summary, once the FFR recognizes a dead-end position, it backs up a certain distance from the obstacle, causing the problem, and then follows its boundary in a randomly generated direction.

A new potential field function is constructed without taking into account the goal and the obstacle being followed. The attractive potential pulls the FFR in the chosen direction. The repulsive potential prevents collision with other obstacles. Notice that at this point the obstacle giving the problems is at a distance greater than the threshold (i.e. no repulsive potential). If one direction fails to clear the obstacle, another one of the remaining directions is chosen and tried. If all directions fail to escape the local minima, the FFR backs up one step along the normal direction and then re-applies the same algorithm again. However, another local minimum may appear, at which this heuristic is recursively applied, while following a certain direction. At this point, it is quite possible that the FFR will search areas that are already visited from other configurations while trying to escape the original local minima. Therefore, we introduce the second heuristic to prevent redundant searches. In the following section we introduce an example to explain the above mentioned procedure.

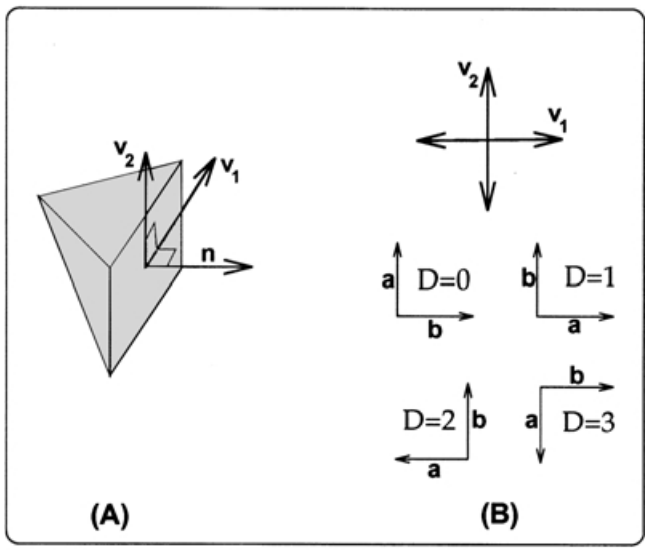


Fig. 3. Constructing the ab-grid.

5.2. Avoid-region heuristic

This heuristic is invoked from within the first heuristic to prune the search space. The basic idea is that in order to continue in the goal direction and avoid an obstacle at the same time, the FFR must continually diverge further and further from the direction that would be chosen if the FFR moved in the goal's direction. Since at this point the FFR is restricted to two dimensions, it can only diverge "left" or "right" based on the fact that the other two directions (i.e. forward and backward) are already dealt with by the first heuristic. The avoid region heuristic is used to force the FFR to diverge further and further in both directions and, therefore, to construct an avoidance region, which the FFR is not going to enter (Figure 4). This continues until the obstacle, causing the local minima, is cleared or the point at which the first heuristic was used becomes a dead-end. In the latter case, the current direction fails and another direction is tried while the second heuristic controls the search process.

To illustrate how both heuristics work, consider the scenario introduced in Figure 4. The FFR reaches a dead-end at D_1 and then applies the follow-boundary heuristic to escape this local minima. As a result, it backs a certain distance from the obstacle and follows its boundary in a randomly chosen direction (in the scenario $D=1$, Figure 3(B)). The FFR visits $R_1, \cdots, R_4$, and D_2, which is another dead-end. At this point the avoid region heuristic is invoked and an avoidance region is created, Figure 4(A). In this case it is only a line segment, $\overline{R_4 D_2}$, where R_4 is the reference point and D_2 is the base. The FFR backs up to R_4 and chooses the next best position based on potential costs of reachable points. Let's assume that the FRR chooses to move to another dead-end point, D_3. Accordingly, the avoidance region is updated to include $\overline{D_2 D_3}$ as a base of the avoidance region, described by the triangle R_4, D_2, and D_3 on Figure 4(B). The FFR backs up again to R_4 looking for the next possible position. If it chooses D_4, another dead-end location,the avoidance region will cover more region (triangle R_4, D_3, and D_4, Figure 4(B)).

Afterwards, the FFR returns again to R_4 and tries to find other possible configurations. If there are none, R_4 is

https://doi.org/10.1017/S0263574701004003 Published online by Cambridge University Press

considered a dead-end point and, therefore, the FFR will back one further step to R_3. This means a new reference point and henceforth the avoidance region becomes larger. The robot will repeat the same procedure above looking for a promising configuration that will lead to an escape.

Depending on the reachability field around R_3, R_4, and R_5, the FFR may or may not land in the avoidance region (Figure 4(C)). If the robot chooses a configuration that is inside this region, the second heuristic will eliminate it directly before consideration. In our scenario, we assume that the FFR moves from R_6 to R_5 and then R_6. The avoidance region will not change. Assuming moving straight ahead, from R_3 will decrease the potential, the FFR chooses a forward diagonal point, D_5.

Because D_5 is a dead-end, the avoidance region is updated as in Figure 4(C). If the second heuristic is not used, the FFR may go in the forward/downward direction from R_6, which will lead to a point inside the avoidance region and, therefore, the robot may reach previously visited dead end points. If there are no escapes at R_6 and R_5 then both are considered dead-end points and the FFR will return to R_3. The same scenario above could occur at the lower side as in Figure 4(D) generating a bigger avoidance region (triangle R_3, D_5, and D_6). The second heuristic constraints the choices of the first heuristic to points that are outside the avoidance region. When the FFR moves a long way following a boundary, the number of points eliminated by the second heuristic is significant. The scenario continues until the FFR

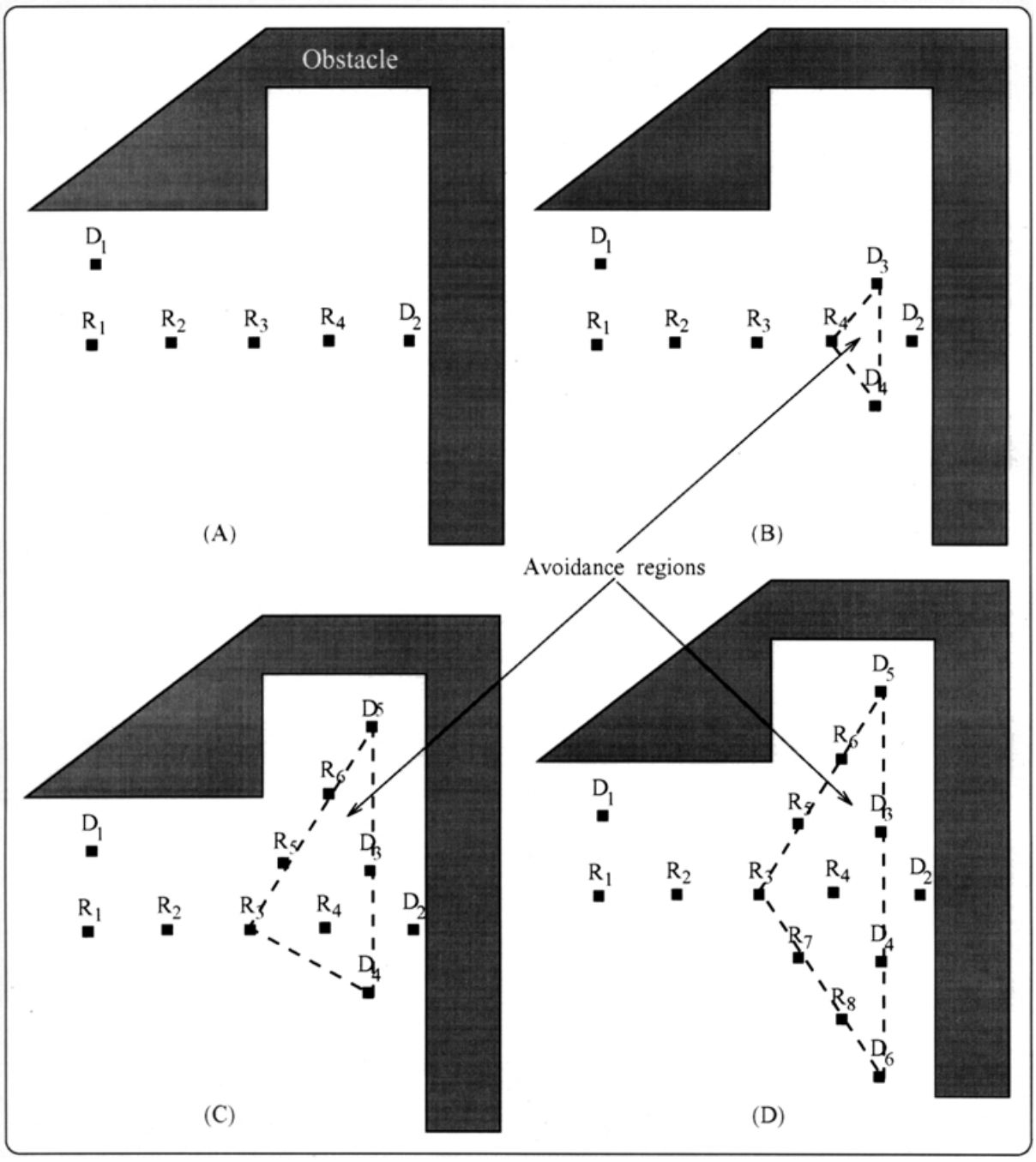


Fig. 4. An example of using both heuristics to escape local minima problems.

https://doi.org/10.1017/S0263574701004003 Published online by Cambridge University Press

backs up to R_1 and chooses to move left from there. In Section 7, we show complete simulation examples.

6. ALGORITHM

Recall that we model the local nature of information about the environment by considering a visibility field as in Figure 1. The general algorithm used can be described by the following steps:

I. *Discretize the reachability field:* since searching a continuous space for an optimum local solution is hard, we discretize the local visible region (i.e. sphere) into a set of equally spaced grid points, and search for the optimum next position in the discretized reachable space. The size of the visibility field is set by the user and the region that is reachable by the FFR is determined by the algorithm.
II. *Find optimum configuration:* we determine the reachability field (a sphere inside the visibility field) and then compute the potential functions at each grid point in the reachability field. The grid point that generates the maximum potential is chosen as the most promising configuration to lead towards the goal. Heuristics are used whenever a dead-end point is encountered (described below).
III. *Continue process:* repeat steps I and II until the goal is reached or failure is reported.

In the sequel, we describe how both heuristics work:

Heuristics:
If the current grid point is not a local minima
- Choose p with the minimum potential
- Eliminate p from the grid
- Push the parent point and grid to the stack

else
If the closest feature of the closest obstacle is a face
- Move a certain distance from obstacle
- Use the follow-boundary heuristic to clear this obstacle
- Clear stack and use the potential functions again

else
If stack is empty
Return failure
Otherwise
Pop old point and grid from the stack

7. SIMULATION RESULTS

In all following experiments, we assume a contained 3D environment such as a tank. The environment is represented as a cube of size 2 distance-units. The center of the environment is placed at (0, 0, 0). The weights for the goal attraction, correction factor for height and obstacle repulsion are left at their default values (specified earlier). In Figures 5 through 11, starting point (P_s) is the closest location of the FFR to the coordinate system and the goal point is the furthest point that the FFR is able to reach (p_g).

7.1 The effect of varying motion parameters

We start with a maximum visibility field of 0.25 distance-units and normalgrid density. Unless otherwise specified, all figures use the starting position: $(-0.9, -0.9, -0.9)$ and the goal is located at (0.9, 0.9, 0.9). Within each figure, the environment is the same for all experiments. Figure 5 shows the effect of choosing different directions to clear an obstacle. In all three runs of this experiment, the FFR reaches a dead-end created by the box and the pyramid near the goal. At this dead-end point the closest obstacle is the top-right pyramid. In Figure 5(A), the FFR escapes in the positive Z-axis direction. The FFR escapes the pyramid and progresses upward towards the goal. In Figure 5(B), it escapes in the negative Z-axis direction. When the pyramid is cleared, the FFR reached another dead end point. The FFR chooses to clear the box in the positive Z-axis direction. Afterwards, the robot successfully found its way to the goal. In Figure 5(C), the FFR clears the pyramid in thedirection towards its base (negative Y-axis). After clearing, the FFR progresses towards the goal until it reaches another dead end in front of the pyramid. This time the robot clears it in the negative Z-axis direction and then

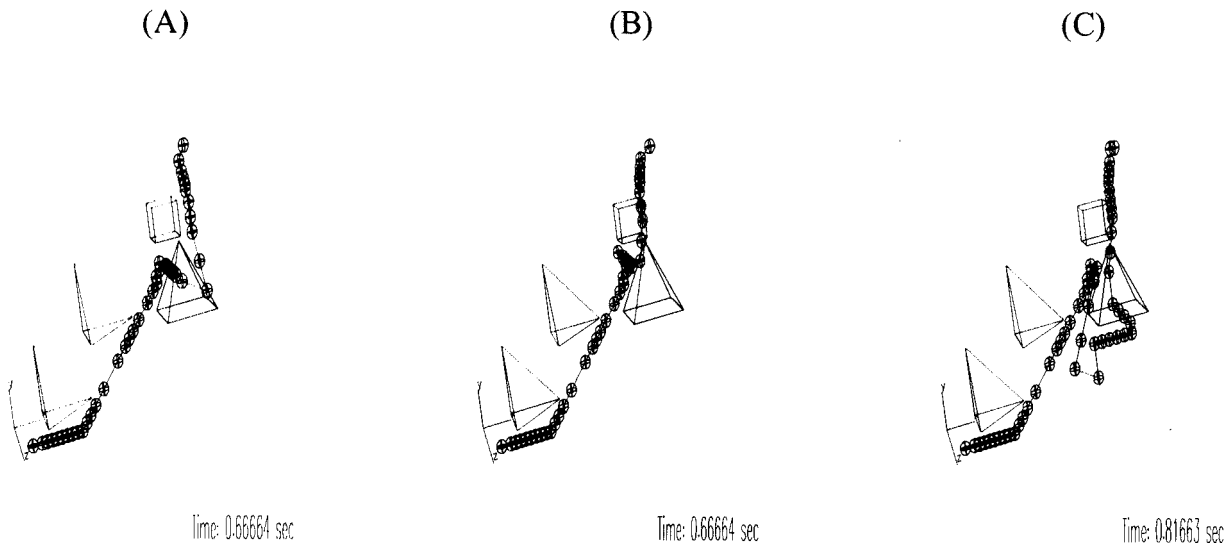


Fig. 5. Choosing different directions to clear an obstacle.

https://doi.org/10.1017/S0263574701004003 Published online by Cambridge University Press

progresses successfully to the goal. The time depicted beneath each simulation in Figure 5 refers to execution time.

In Figure 6, we investigate the effect of using normal grid density against high grid density in Figures 6(A) and 6(B), respectively. The path followed by the FFR in Figure 6(B) is smoother. However, the path in Figure 6(A) is generated over twice as fast. We conduct a similar simulation but using a cubic and a spherical type of robots as in Figure 7. The

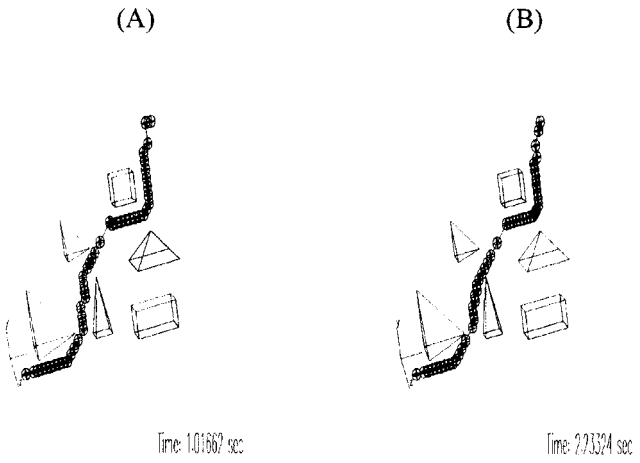


Fig. 6. The effect of using normal and high grid density distribution.

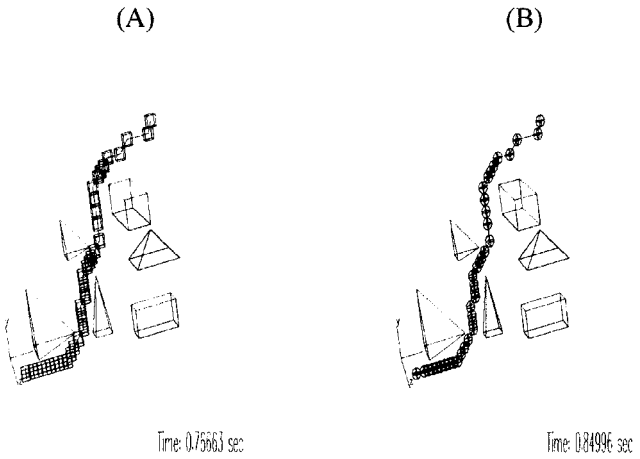


Fig. 7. Cubic vs. spherical FFR.

paths produced in this figure are slightly different than the ones shown in Figure 6 because one of the obstacles (top box near the goal) has been enlarged to produce a relatively cluttered environment. We repeat the same experiment but in a congested environment (Figure 8). Figure 8(C) shows another view of the path of Figure 8(A) to confirm that the FFR did not collide with any obstacles. The paths taken using the two grid densities are entirely different. This is because the FFR found a new path between the tetrahedrons under high density distribution (Figure 8(B)). Notice that under normal grid density the path, once the robot is away from the obstacles, is not smooth. This is because the robot continually switches the direction it is following between diagonal and straight up.

7.2 Use of heuristics

In Figure 9, we investigate the effect of increasing the maximum visibility field and the of use of the avoid-region heuristic. The robot and the obstacles are all assumed to be in a tank (bounded environment). The visibility range is set to 0.25 distance-units in Figure 9(A) and to 0.5 distance-units in Figure 9(B). Figure 9(C and D) are alternative views of Figure 9(A and B), respectively, to show that the robot did not collide with the obstacles and/or walls of the tank. In both experiments, the FFR reached a dead-end in front of the middle obstacle. The FFR tried to clear this obstacle inthe negative Z-axis direction but the back wall of the tank prevented the FFR from clearing the obstacle. As a result, the avoid-region heuristic is invoked to enable the robot to diverge further and further (into the negative X-axis direction) from the obstacle until the obstacle was cleared. Since the visibility range is larger in Figure 9(B), the robot moved further in each step when clearing the middle obstacle, which caused the robot to escape the obstacle faster.

In Figure 10, the system generates two paths (using the same parameters) for a cubic and spherical robots. The start position is $(-0.9, 0, 0)$ and the goal one is $(0.9, 0, 0)$. The robot is forced to enter the trap-obstacle, modeled as a closed box with one opening towards the starting point. Because the visibility field falls within the trap, the FFR keeps navigating towards the goal along the straight line connecting both starting and goal points until it reaches a dead-end near the back of the trap. This experiments shows

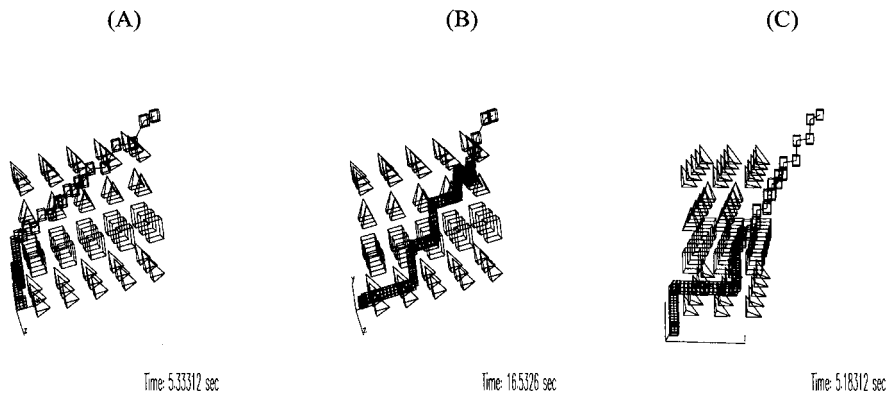


Fig. 8. The effect of normal and high grid denstity distribution in a cluttered environment.

https://doi.org/10.1017/S0263574701004003 Published online by Cambridge University Press

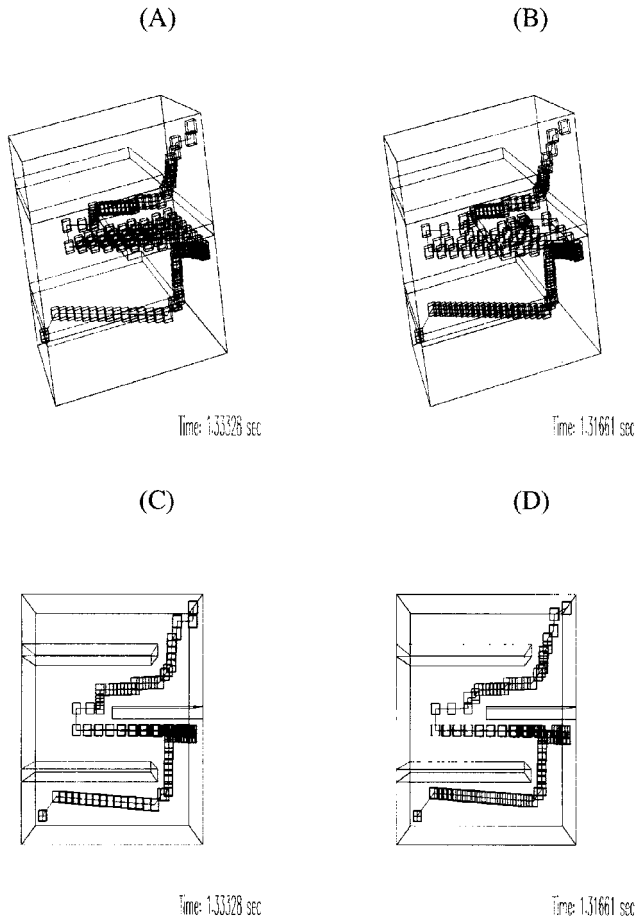


Fig. 9. The effect of increasing the reachability field.

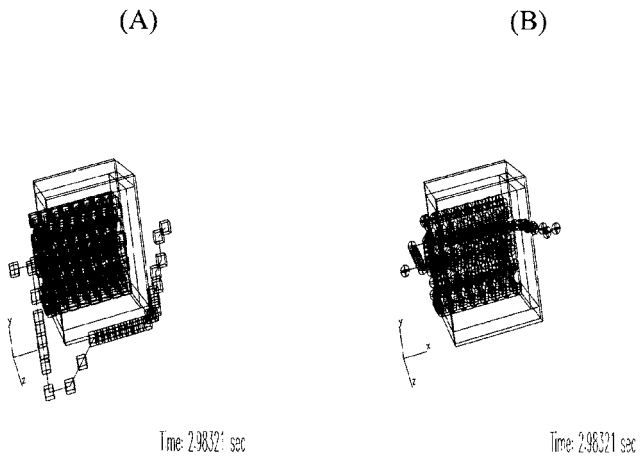


Fig. 10. The FFR escapes many local minima while trying to exit the trap.

how the FFR tried all different directions in order to clear the trap-obstacle. The use of heuristics helped the robot find its way effectively out of the trap. The FFR cleared the obstacle in two different directions in the experiment but since the robot backed out to the same grid point (heuristics effect), both trajectories took the same time.

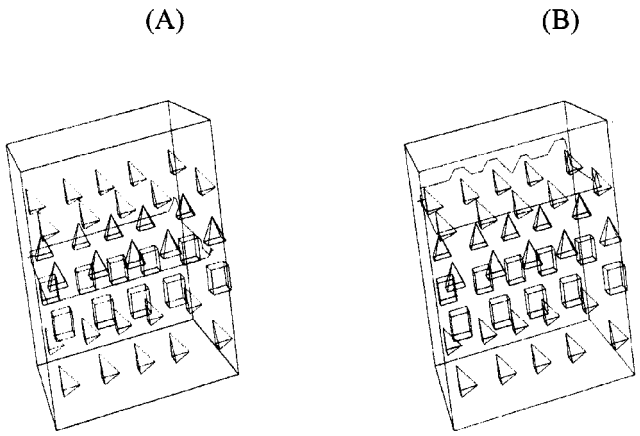


Fig. 11. Two different trajectories are generated at different altitude for inspection purposes.

7.3 An example of Problem 2

A simulation of an example of Problem 2 is shown in Figure 11. The weights were kept at their default values. The visibility range is 0.25 distance-units and the tank has the same dimensions as before. The FFR followed the walls of thetank but keeping a clearance distance of 0.1 distance-units from the inner surface. The robot started at the point $(-0.9, \hat{h} \ -0.9)$ where $\hat{h}$ is the pre-set altitude. The path of the FFR is counter-clockwise if considering a top view of the tank. Figure 11(A) is produced while setting the altitude at the middle of the tank (i.e. $\hat{h} = 0.0$). The robot is forced to go down when it reached the pyramid which is located near the second corner along its path. The height is restored shortly after clearing this obstacle. However, the FFR is forced to move upwards to avoid the next pyramid. Figure 11(B) is generated where $\hat{h}$ is set to 0.5 distance-units. Here the FFR kept on trying to move back to altitude $\hat{h}$ after clearing each obstacle encountered at this altitude. In conclusion, a robot can be sent inside a certain environment (i.e. tank or pipe) to collect data about its inner surfaces. The collected images need to be put together (image registration) in order to have a clear view of the interior surfaces of the environment for inspection.[26]

8. CONCLUSION

We presented a novel potential-based technique with heuristics to path planning using only local information and the knowledge of the start and the goal positions in a 3D environment. Simulation results showed how heuristics helped in enhancing the search process and the robot's ability to avoid local minimas. A robust algorithm that used the sum of obstacle repulsion, goal-attraction, and level-attraction potentials is described. To avoid obstacles, we proposed two special-purpose heuristics: the "follow-boundary" and the "avoid-region". While the first heuristic is used to overcome the problem of local minima, the second heuristic is designed to prune the search space. An application of this algorithm to inspect the inner surfaces of a tank is also described. In future, we intend to address the planning problem in the presence of moving obstacles.

https://doi.org/10.1017/S0263574701004003 Published online by Cambridge University Press

References

1. Y. Hwang and N. Ahuja, "Gross motion planning – a survey," *ACM Computing Surveys*, **24**(3), 219–291 (1992).
2. Jean-Claude Latombe, *Robot Motion Planning* (Kluwer Academic Publishers, London, 1991).
3. K. Fujimura and H. Samet, "Motion planning in a dynamic environment," *Proceedings of the 1989 IEEE International Conference on Robotics and Automation*, (1989), pp. 324–330.
4. K. Kant and S. Zucker, "Towards efficient trajectory planning: the path-velocity decomposition," *Int. J. Robotics Research*, **5**, 72–89 (1986).
5. T. Fraichard and C. Laugier, "Path-velocity decomposition revisited and applied to dynamic trajectory planning," *Proceedings of the 1993 IEEE International Conference on Robotics and Automation*, (1993), pp. 40–45.
6. A. Basu and A. Elnagar, "Safety optimizing strategies for local path planning in dynamic environments," *Int. J. Robotics and Automation*, **10**(4), 130–142 (1995).
7. T. Tsubouchi, K. Hiraoka, T. Naniwa and S. Arimoto, A mobile robot navigation scheme for an environment with multiple moving obstacles," *Proceedings of the 1992 IEEE/RSJ International Conference on Intelligent Robots*, (1992), pp. 1791–1798.
8. J. Canny, *The Complexity of Robot Motion Planning* (MIT Press, Cambridge, MA, 1988).
9. R. Brooks and T. Lozano-Pérez, "A subdivision algorithm in configuration space for findpath with rotation," *Proceedings of the 8th International Conference on Artificial Intelligence*, (1983), pp. 799–806.
10. K. Kutulakos, V. Lumelsky and C. Dyer, "Vision-guided exploration: A step toward general motion planning in three dimensions," *Proceedings of the 1993 IEEE Conference on Robotics and Automation*, (1993), **Vol. 1**, pp. 289–296.
11. V. Lumelsky and A. Stepanov, "Path planning strategies for a point mobile automation moving amidst unknown obstacles of arbitrary shape," *Algorithmica,* **2**, 403–430 (1987).
12. O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots," *Int. J. Robotics Research,* **5**(1), 90–98 (1986).
13. F. Miyazaki and S. Arimoto, "Sensory feedback based on the artificial potential for robotics," *Proceedings of the 9th Triannual World Congress of International Federation of Automation Control*, (1984), pp. 2381–2386.
14. V. Pavlov and A. Vornion, "The method of potential functions for coding cnstraints of the external space in an intelligent mobile robot," *Soviet Automatic Control*, (1984), **Vol. 6**, pp. 799–806.
15. E. Rimon and D. Koditscheck, "Exact Robot navigation using cost functions: The case of distinct spherical boundaries in e^n," *Proceedings of the 1988 IEEE International Conference on Robotics and Automation*, (1988), pp. 1791–1796.
16. E. Rimon and D. Koditscheck, "The construction of analytic diffeomorphism for exact robot navigation on star worlds," *Proceedings of the 1989 IEEE International Conference on Robotics and Automation*, (1989), pp. 21–26.
17. R. Arkin, "Behaviour-bases robot navigation for extended domains, *Adaptive Behavior,* **1**(2), 201–225 (1992).
18. Y. Hwang and N. Ahuja, "A potential field approach to path planning," *IEEE Transactions on Robotics and Automation,* **8**(1), 23–32 (1992).
19. S. Jun and K. Shin, "A probabilistic approach to collision free path planning," *Proceedings of the 1988 IEEE International Conference on Robotics and Automation*, (1988), pp. 220–227.
20. R. Brooks, "Solving the find path problem by good representation of free space," *IEEE Transactions on Systems, Man, Cybernetics,* **13**(3), 190–197 (March/April, 1983).
21. R. Cole and M. Sharir, "Visibility problems for polyhedral terrains," *Technical Report No. 266* (*New York University*, December 1986).
22. J. Canny and J. Reif, "New lower bound techniques for robot motion planning," *Proceedings of the 1987 IEEE Conference on Foundations of Computer Science,* (1987), 1: 49–60.
23. J. Reif and M. Sharir, "Motion planning in the prescence of moving obstacles," *Proceedings of the 25th IEEE Symposium on Foundations of Computer Science,* (1983), pp. 144–154.
24. J. Schwartz and M. Sharir, "On the piano movers' problem: I. The case of a two-dimensional rigid polygonal body motion amidst polygonal barriers," *Communications on Pure Applied Mathematics,* **36**, 345–398 (1983).
25. M. Lin and M. Canny, "A Fast algorithm for incremental distance calculation," *Proceedings of the 1991 IEEE International Conference on Robotics and Automation*, (1991), pp. 2: 1008–1014.
26. A. Elganar and A. Basu, "From 2D surface patches to 3D reconstructed models: Theory and Applications," *Pattern Recognition,* **13**(10), 1419–1430 (1998).

https://doi.org/10.1017/S0263574701004003 Published online by Cambridge University Press