On Autonomous Terrain Model Acquisition by a Mobile Robot

N.S.V. Rao and S.S. Iyengar
Louisiana State University
Baton Rouge, LA 70803

C.R. Weisbin
Oak Ridge National Laboratory
Oak Ridge, TN 37831

ABSTRACT

In this paper we consider the following problem: A point robot is placed in a terrain populated by unknown number of polyhedral obstacles of varied sizes and locations in two/three dimensions. The robot is equipped with a sensor capable of detecting all the obstacle vertices and edges that are visible from the present location of the robot. The robot is required to autonomously navigate and build the complete terrain model using the sensor information. We establish that the *necessary* number of scanning operations needed for complete terrain model acquisition by any algorithm that is based on 'scan from vertices' strategy is given by $\sum_{i=1}^{n} N(O_i)-n$ and $\sum_{i=1}^{n} N(O_i)-2n$ in two and three dimensional terrains respectively, where $O=\{O_1, O_2, \cdots, O_n\}$ is the set of the obstacles in the terrain, and $N(O_i)$ is the number of vertices of the obstacle O_i.

Keywords and Phrases:

Path Planning, Terrain Acquisition, Collision Avoidance.

1. INTRODUCTION

In recent years there has been an enormous amount of research activity generated in the area of *navigation and path planning* for mobile robots. Much of this work could be thought of as an offshoot of the pioneering works of Lozano-perez and Wesley [1], Reif [2], Schwartz and Sharir [3], and O'Dunlaing and Yap [4]. In this work the robot is located in a terrain whose model is precisely known. A path has to be planned to navigate a robot from a specified point to a specified destination point (if such path exists). A comprehensive survey of these and related techniques for robot path planning is available in Whitesides [5]. Another important problem is the *navigation in unexplored terrains*. Here the robot is equipped with a sensor with which the robot scans the terrain, and a navigation path is planned based on these sensor readings. In general several sensor operations are needed for planning a navigational course. Lumelsky and Stepanov [6] present nice solutions to a restricted version of this problem. Iyengar et al [7] and Rao et al [8] present a technique that utilizes the sensor readings to construct a world map through *incidental learning*. Oommen et al [9] presents a more formal treatment for the case of convex polygonal obstacles. In these approaches the terrain model acquisition is purely *incidental* i.e., the construction of the terrain model is only secondary and scanning is performed for the purpose of navigation.

Another important problem in the navigation in unexplored terrains is the *Terrain Acquisition Problem* in which the robot is required to autonomously navigate and build the complete terrain model through the sensor readings. In this paper we consider the following version of terrain acquisition problem: A point-sized robot M is placed in a two/three dimensional obstacle terrain O. The terrain O is populated by the set of obstacles $\{O_1, O_2, \cdots, O_n\}$, where O_i is a polyhedron. We assume that O is finite, i.e., O can be inscribed in a circle/sphere of finite radius in two/three dimensions. Furthermore each O_i is finite and had a finite number of vertices. Initially the sizes and locations of the obstacles are totally unknown to the robot. The robot M is equipped with an ideal sensor system capable of detecting all edges and vertices visible to the robot from its current position. The robot is required to autonomously navigate in the terrain and acquire the *complete* obstacle terrain model, i.e. obtain the locations of all edges and vertices of each obstacle of O. The main motivation for this problem stems from the fact that after terrain acquisition phase, the future navigation of the robot can be carried out without sensor operations using the techniques for navigation in known terrains. In many cases navigational path can be made optimal in terms of the distance to be traversed by the robot.

A solution to this problem is given by Rao et al [10] based on the incremental construction of the visibility graph of the terrain. The same technique is extended to a finite-sized robot in plane by Rao et al [11]. The algorithm of [10] is guaranteed to acquire the complete terrain model in finite time. The algorithm terminates when a scan operation is performed from each vertex of every obstacle and consequently the number of scanning operation required is $\sum_{i=1}^{n} N(O_i)$, where $N(O_i)$ is the number of vertices of the obstacle O_i. However, this is only a sufficient condition on the number of scan operations. In this paper we establish that for any terrain acquisition algorithm (based on scan from vertex strategy) there exists a terrain O such that the necessary number of scan operations is given by $\sum_{i=1}^{n} N(O_i)-n$ and $\sum_{i=1}^{n} N(O_i)-2n$ respectively for two and three dimensional terrains. In other words, no more than one (two) scan operations per obstacle can be skipped in two (three) dimensional terrains. We also show that a strategy that randomly skips one vertex (two vertices) per obstacle will not acquire the complete terrain model in two (three) dimensional terrains. We then list a number of issues for future research.

The organization of the paper is follows: In section 2, we briefly discuss the issues involved in the terrain acquisition problem and also the algorithm of [10]. In section 3, we present the bound on the necessary number of scan operations.

2. TERRAIN ACQUISITION METHODOLOGY

During the terrain acquisition the robot M is required to plan and execute a *navigational course*; robot stops at certain points, called the *sensing points*, on the path to carry out the scan operations. The terrain model is reconstructed by integrating the scanning information obtained from the individual scan operations. In general, the navigational path could only be planned in an incremental manner by utilizing the scan information because the terrain is unexplored. The main requirement on the terrain acquisition algorithm is that the *complete* terrain model should be acquired in a finite amount of time.

Here we deal with *vertex-based* terrain acquisition methods where the sensing points are always vertices of the obstacles, i.e., every scan operation is performed from an obstacle vertex. The robot M moves from vertex to vertex during the navigational course. The algorithm of [10] is based on this strategy. There are two key issues that are important for a terrain acquisition algorithm:

(a) Computing the next vertex to be visited,

(b) Detecting the completion of terrain acquisition (termination of the algorithm).

We now briefly discuss the terrain acquisition algorithm of [10]. Let $VER(O_i)$ denote the set of vertices of O_i. Let $V=\bigcup_{i=1}^{n} VER(O_i)$ be the set of all vertices of the obstacles. The *visibility graph* of the terrain O, denoted by $VG(O)$, is a graph (V,E), where an edge $(v_1,v_2)\in E$, $v_1,v_2\in V$, exists if and only if (v_1,v_2) is either an edge of an obstacle or v_1 is visible from v_2 and vice versa. In Fig.1. an obstacle terrain populated by three obstacles O_1,O_2 and O_3 is shown and its visibility graph is shown in Fig.2. A vertex is said to be *explored* if a scan operation is performed from v, and otherwise v is said to be *unexplored*. Once v is explored then the adjacency list of v in the visibility graph is known. The robot M is initially placed at a point in the obstacle terrain. Then M scans and moves to a vertex. From this point the terrain acquisition algorithm, called algorithm ACQUIRE, of [10] is invoked. Let M start at vertex $v_0\in V$. A scan is performed and the adjacency list of v_0 is stored. Then M moves to an adjacent unvisited vertex and recursively applies this method. When an unexplored vertex is visited it is pushed onto a stack called *path-stack*. Let M be located at a vertex v from which it performed a scan operation. Then M moves to a nearest unexplored vertex adjacent to v if one exists. The M can move to this chosen vertex in a straight line because it is seen from v. If all the vertices adjacent to v are visited then the path-stack is used to obtain the next sensing point. The top of the path-stack is recursively popped till a node v_1 with unvisited adjacent nodes is found. Shortest paths to all the unvisited adja-

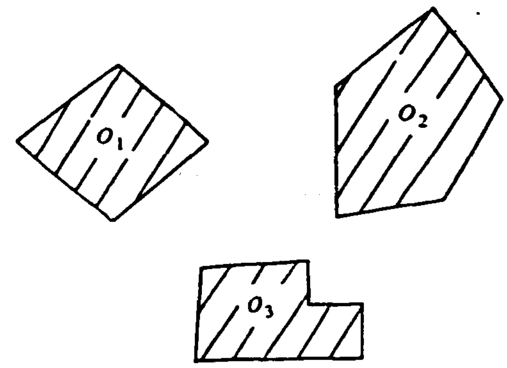


Fig. 1. Obstacle terrain

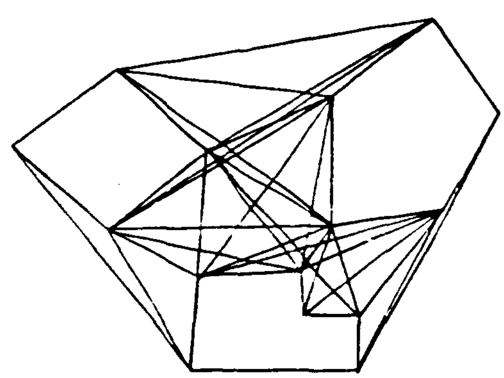


Fig. 2. The visibility graph for the terrain of Fig.1.

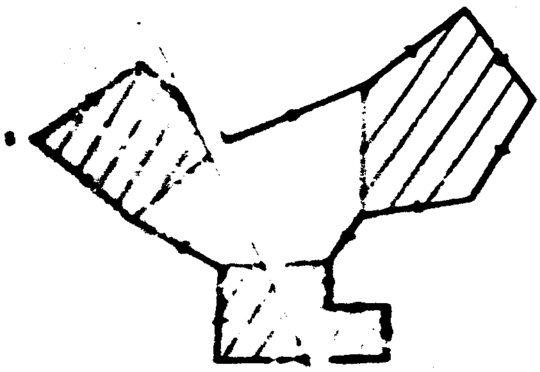


Fig.3. Navigational path [illegible] terrain acquisition [illegible] (shown in dark lines) [illegible] consists of [illegible] shortest path algorithm on the [illegible] graph. The path of minimum distance [illegible] and M moves to the appropriate vertex [illegible] the algorithm is [illegible] in Fig.3 [illegible] path taken by M is shown [illegible] Fig.3. In this example [illegible] the starting point. [illegible] stage of terrain [illegible] is shown in Fig.4.

The [illegible] showing the following

1) The [illegible] graph [illegible] exists [illegible]

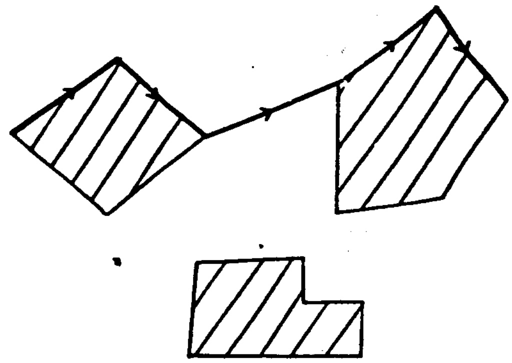
2) The manner [illegible] of the obstacles are explored [illegible] a depth first search [illegible]

The [illegible] all vertices are explored in a [illegible] a depth first search of a con[illegible] vertices in finite amount of time). [illegible] free space is seen by at least one scan [illegible] a vertex. Thus the terrain acquisition will be complete in finite amount of time. Clearly the number of scan operations carried out by M is $\sum_{i=1}^{n} N(O_i)$. This is only a sufficient condition on the number of scan operations.

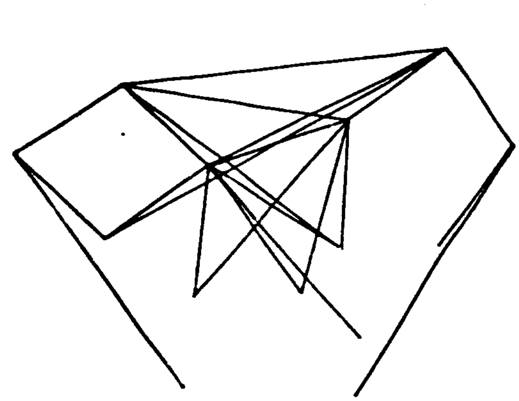
In the next section we show that for any vertex-based terrain acquisition algorithm there exists a terrain such that the necessary number of scanning operations is given by $\sum_{i=1}^{n} N(O_i)-n$.

3. NUMBER OF SCAN OPERATIONS

Consider a vertex-based terrain exploration algorithm (and algorithm of [10] is one such). The algorithm performs scans and detects newer vertices which will be explored in subsequent scans. During terrain exploration by a vertex based algorithm no more than *one* vertex per obstacle can be left unexplored in two dimensional terrain constructed as explained below. For three dimensional terrains no more than two vertices per obstacle can be left unexplored in our specially constructed terrain. The basic idea is illustrated in Fig.5. We consider a single convex polygonal obstacle in Fig.5(a). If M starts at a vertex it detects one new vertex with one exploration (except when the first vertex is explored) of a vertex as the robot moves along the circumference of the obstacle. In other words at no point of time the terrain acquisition could be declared complete if there are two unexplored vertices say v_1 and v_2. This is because the robot does not, in general, know what lies on the hinder (unexplored) side of the line joining v_1 and v_2. There could a single vertex or a number of edges on the other side of the line joining v_1 and v_2 as in Fig.5 (b) and (c). For three dimensional terrains, no more than *two* vertices per obstacle can be left unexplored. This is because if three vertices (say v_1, v_2 and v_3) are left unexplored then the information on the hinder side of the plane formed by the vertices v_1, v_2 and v_3 is not known in general. The hidden side of the obstacle can be either a simple plane or composed of a a number of planes as shown in Fig.6 (a) and (b).



(a) Navigational path (shown in dark)



(b) Partially built visibility graph

Fig.4. Intermediate stage of exploration

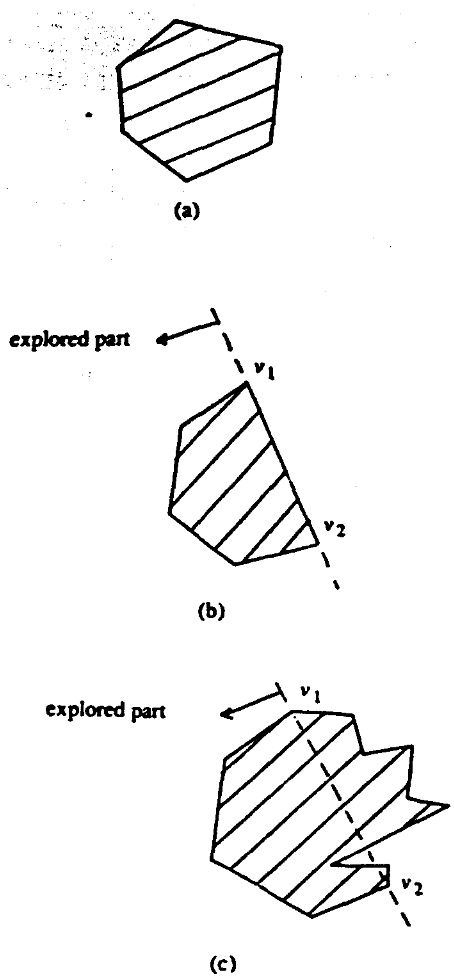


Fig.5. Two dimensional case

Theorem 1:

For a vertex-based terrain acquisition algorithm and given positive integer n there exists a terrain $\{O_1, O_2, \cdots, O_n\}$ of n polyhedral obstacles such that the necessary number of scan operations is

$\sum_{i=1}^{n} N(O_i) - n$ for two dimensional terrain

$\sum_{i=1}^{n} N(O_i) - 2n$ for three dimensional terrain

Proof: We use induction on the number of obstacles in the terrain. Consider $n=1$. In two dimensional terrains consider a convex polygon as in Fig 5(a). Note that from a vertex v_2, we can only see two vertices that are adjacent to v. Apart from the first scan, no more than one unexplored vertex can be seen in any scan operation. From the discussion above M has to carry out scanning till no more than one vertex is unexplored. Thus $N(O_1)-1$ is the necessary number of scan operations for two dimensional terrains. By similar arguments we can show that the necessary number of scan operation is

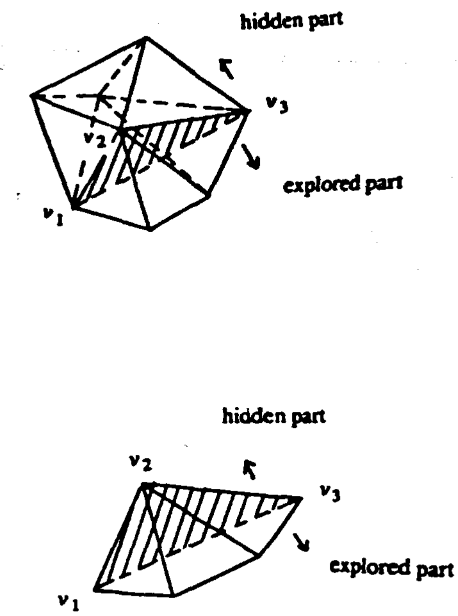
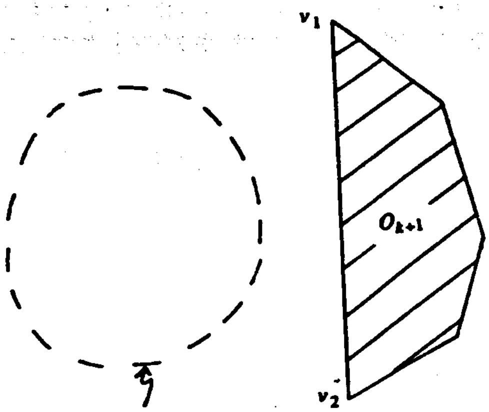


Fig.6. Three dimensional case

$N(O_1)-2$. Hence the claim is true for $n=1$.

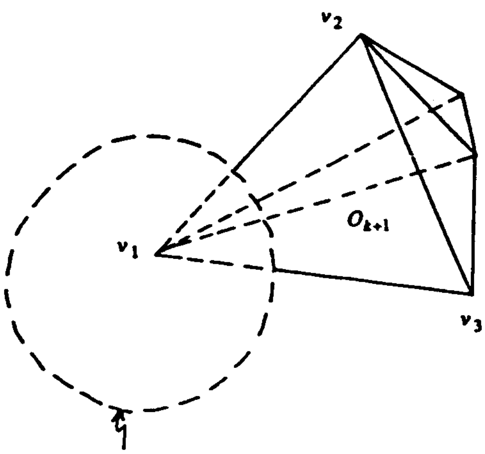
Assume that the claim is true for $n=k$. There exist a terrain of k obstacles with the necessary number of scan operations given in the theorem. Now construct a terrain of $k+1$ obstacles as follows: In two dimensions add a big polygon O_{k+1} outside the circle inscribing the terrain that satisfies the induction hypothesis as shown in Fig.7. The $k+1$th polygon has a long edge joining v_1 and v_2 that obscures the remaining edges of the polygon from the scan operations carried out in the terrain of k obstacle. Thus the scan operations needed during the exploration of the $k+1$th obstacle is $N(O_{k+1})-1$. Hence total number of necessary scan operations for two dimensional terrains is given by $\sum_{i=1}^{k+1} N(O_i)-(k+1)$. For three dimensional terrains the obstacle O_{k+1} is such that a plane formed by three vertices v_1, v_2 and v_3 obscures the rest of the obstacle from a scan in the terrain of k obstacles as in Fig.8. The O_{k+1} lies outside the sphere the encloses the terrain of k obstacles. Using the arguments similar to two dimensional case we can show that the necessary number of scan operations to acquire O_{k+1} is $N(O_{k+1})-2$. Thus the theorem follows by mathematical induction. □

In the above theorem we have seen that no more than one (two) vertices per obstacle can be left unexplored in two (three) dimensional terrain. The natural question is to ask if we can always skip one (two) vertices per obstacle for two (three) dimensional terrains. The answer is no as the vertices



Circle cont...ing k obstacles

Fig.7. Two dimensional case - Addition of O_{k+1}



sphere containing k obstacles

Fig.8. Three dimensional case - Addition of O_{k+1}

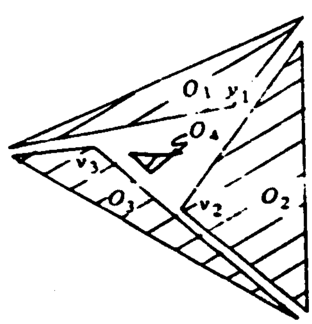


Fig. 9. Configuration - two dimensional case

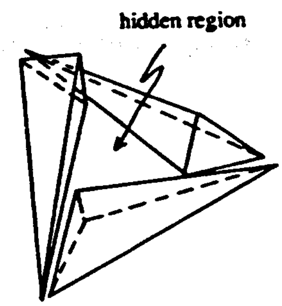


Fig. 10. Configuration - three dimensional case

are to be randomly skipped. This is illustrated in Fig. 9 and Fig. 10. In two dimensions the if the robot skips the vertices v_1, v_2 and v_3 then the obstacle O_4 will not be detected. Fig.10 shows a three dimensional example. The configurations such as shown in Fig.9 and 10 can be formed with any (finite) number of obstacles which could be other than triangles or tetrahedrons. Fig.11 shows one such example. It is open at this point to design a vertex-based terrain acquisition algorithm (or show algorithm does not exists) that skips one (two) vertices for each obstacle and guaranteed to acquire the complete obstacle terrain model.

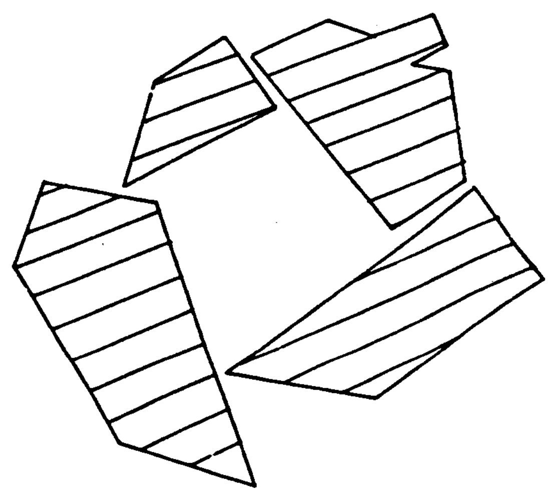


Fig. 11. A general configuration

4. CONCLUSIONS

In this paper we have shown that for any vertex-based terrain acquisition algorithm there exists a terrain such that the necessary number of scan operations is given by $\sum_{i=1}^{n} N(O_i)-1$ and $\sum_{i=1}^{n} N(O_i)-2$ respectively for two and three dimensional obstacle terrains. In other words, we do not expect to design a vertex-based terrain acquisition algorithm that has complexity lower than the above stated sums (in terms of the sensor operations). There exists a terrain acquisition algorithm with the number of sensor operations given by $\sum_{i=1}^{n} N(O_i)$ [10]. It would be interesting to see if there exists a terrain acquisition algorithm with only the necessary number of sensor operations given in this paper.

REFERENCES

[1] LOZANO-PEREZ, T., and M. A. WESLEY, An Algorithm for Planning Collision-free Paths Among Polyhedral Obstacles, Commun. ACM, Vol. 22, No. 10, October 1979, pp. 560-570.

[2] REIF, J., Complexity of the mover's problem and generalizations, *Proc. 20th Symp. on Foundation of Computer Science*, 1979, pp. 421-427.

[3] SCHWARTZ, J.T., and M. SHARIR, On the piano movers' problem I: The special case of a rigid polygonal body moving amidst polygonal barriers, Communications Pure Applied Mathematics, Vol. 36, 1983, pp. 345-398.

[4] O'DUNLAING, C., and C. YAP, A 'Retraction' method for planning the motion of a disc, Journal of Algorithms, Vol.6, 1985, pp.104-111.

[5] WHITESIDES, S.H., Computational Geometry and Motion Planning, *Computational Geometry*, Editor G.T. Toussaint, Elsevier Science Publishers B.V. (North-Holland), 1985, pp 377-427.

[6] LUMELSKY, V.J., and A.A. STEPHANOV, Effect of Uncertainity on Continuous Path Planning for an Autonomous Vehicle, *Proc. 23rd Conf. on Decision and Control*, Las Vegas, Nevada, December 1984.

[7] S.S. IYENGAR, C.C. JORGENSEN, S.V.N. RAO and C.R. WEISBIN, Robot Navigation Algorithms Using Learned Spatial Graphs, Robotica, Vol. 4, 1986, pp. 93-100.

[8] N.S.V. RAO, S.S. IYENGAR, C.C. JORGENSEN and C.R. WEISBIN, Robot Navigation in an Unexplored Terrain, Journal of Robotic Systems, Vol. , 1986, pp.

[9] B.J. OOMMEN, S.S. IYENGAR, N.S.V. RAO, R.L. KASHYAP, Robot Navigation in Unknown Terrains Using Learned Visibility Graphs. Part I: The disjoint Convex Obstacle Case, IEEE Transactions on Robotics and Automation, 1987, to appear.

[10] N.S.V. RAO, S.S. IYENGAR, B.J. OOMMEN and R.L. KASHYAP, Terrain Acquisition by Point Robot Amidst Polyhedral Obstacles, *Proc. The 3rd Conf. on Artificial Intelligence Applications*, Orlando, FL, Feb. 22-28, 1987, to appear.

[11] N.S.V. RAO, S.S. IYENGAR, C.C. JORGENSEN and C.R. WEISBIN, On Terrain Acquisition by a Finite-sized Mobile Robot in Plane, *Proc. of 1987 The IEEE Int. Conf. on Robotics and Automation*, Raleigh, North Carolina, March 30 - April 3, 1987, to appear.