[20] H. P. Moravec, "Certainty grids for mobile robots," preprint of *Carnegie-Mellon University, The Robotics Institute*, Tech. Rep., 1986.
[21] H. P. Moravec, "Sensor fusion in certainty grids for mobile robots," *AI Magazine*, pp. 61–74, Summer 1988.
[22] Polaroid Corporation, "Ultrasonic ranging system," Ultrasonic Ranging Marketing, 1 Upland Road, Norwood, MA.
[23] U. Raschke and J. Borenstein, "A comparisson of grid-type map-building techniques by index of performance," in *Proc. 1990 IEEE Int. Conf. Robotics and Automation*, Cincinnati, OH, May 13–18, 1990, pp. 1828–1832.
[24] T. B. Sheridan, "Human supervisory control of robot systems," *Proc. 1986 IEEE Int. Conf. Robotics Automation*, vol. 2, 1986, pp. 808–812.
[25] C. F. Thorpe, "Path relaxation: Path planning for a mobile robot," *Carnegie-Mellon University, The Robotics Institute, Mobile Robots Laboratory, Autonomous Mobile Robots, Annual Report*, 1985, pp. 39–42.
[26] Cybermation, 1987, "K2A mobile platform," Commercial Offer, 5457 JAE Valley Road, Roanoke, VA 24014.

Autonomous Robot Navigation in Unknown Terrains: Incidental Learning and Environmental Exploration

NAGESWARA S. V. RAO AND S. S. IYENGAR

***Abstract*—The navigation of autonomous mobile machines, which are referred to as robots, through *unknown* terrains, i.e., terrains whose models are not *a priori* known is considered. We deal with point-sized robots in two- and three-dimensional terrains and circular robots in two-dimensional terrains. The two-dimensional (three-dimensional) terrains are finite-sized and populated by an unknown, but, finite, number of simple polygonal (polyhedral) obstacles. The robot is equipped with a sensor system that detects all vertices and edges that are visible from its present location. In this context, the work deals with two basic navigational problems. In the visit problem, the robot is required to visit a sequence of destination points, in a specified order, using the sensor system. In the terrain model acquisition problem, the robot is required to acquire the complete model of the terrain by exploring the terrain with the sensor. A framework that yields solutions to both the visit problem and the terrain model acquisition problem using a single approach is presented. The approach consists of incrementally constructing, in an algorithmic manner, an appropriate geometric graph structure (1-skeleton), called the navigational course. A point robot employs the restricted visibility graph and the visibility graph as the navigational course in two- and three-dimensional cases respectively. A circular robot employs the modified visibility graph. The algorithms to solve the visit problem and the terrain model acquisition problem based on the abovementioned structures are presented and analyzed.**

I. Introduction

A vital component of unmanned machines or rovers is the navigation system that enables these machines to autonomously navigate to the required destinations. The machines with autonomous navigation capability can be employed in various applications such as autonomous land navigation, unmanned extraterrestrial and underwater exploration, maintenance and repairs in nuclear power plants, operation in chemical and toxic industries, unmanned vigilance and security systems, etc. Such systems must be capable of navigating in known environments, i.e., the environments whose precise models are known, as well as in unknown environments, i.e., environments whose models are not known.

The problem of planning collision-free paths for moving a body through a known terrain has been extensively studied under the popular generic name of the piano movers problem. There has been a surge of research activities in this area due to the important contributions of Lozano-Perez and Wesley [6], O'Dunlaing and Yap [9], Reif [14], and Schwartz and Sharir [15]. Yap [17] and Sharir [16] present excellent surveys of various formulations of this problem and their solutions.

The problem of robot navigation through an unknown terrain, has been studied by several researchers although not to the extent of its counterpart in known terrains. Lumelsky and Stepanov [8] present two algorithms for a point robot to move from a source point to a destination point using touch sensing. In his survey paper, Lumelsky [7] presents a comprehensive discussion on several algorithmic and complexity issues dealing with a point robot in unknown terrains. For the terrains populated by convex polygonal obstacles, Oommen *et al.* [10] develop algorithms for a point robot to navigate to a destination point, and at the same time "learn" about the parts of terrain that are encountered on the way to the destination. Here the robot uses a combination of touch sensing and distance probing. In this treatment, several interesting obstacle configurations such as the mazes, traps etc., are not dealt with. The above problems can be grouped under a broad title, the visit problem, wherein a robot is required to visit a sequence of destination points through an unknown terrain. Another problem, called the terrain model acquisition problem is discussed by Rao *et al.* [13]. Here, a point robot is required to acquire the complete model of the terrain.

The visit problem and the terrain model acquisition problem have been solved independent of each other. Here we present a framework to solve these two problems using a *single approach* that implements a graph search on an incrementally-constructed graph called the *navigation course*. A general outline of this approach has been presented by Rao [11]. Here we present the visibility graph methods to implement this approach, by presenting the technical issues, such as proofs of the properties of various navigational courses, extension to circular robots, lower bounds on sensor operations, etc. We deal with point robots and circular robots. The method of Oommen *et al.* [10] uses the visibility graph (in plane) in their algorithms. We extend their work to terrains with nonconvex obstacles which include mazes and traps. In two-dimensional (2-D) terrains, we show that a subgraph of visibility graph, *the restricted visibility graph*, with only the convex obstacle vertices as nodes, suffices to solve these two problems. This results in a reduction in the number of sensor operations and the storage, if the terrain consists of non-convex corners. Also, we establish the lower bounds on the worst-cast number of scan operations performed by these algorithms.

Motivation for Navigational Problems

The visit problem and the terrain model acquisition problem have been motivated by a specific practical application involving the development of an autonomous rescue robot. However, our treatment is more general than this specific application. In this application, the robot is required to carry out rescue operations in nuclear power plants in the event of radiation leakages, and other events that prevent human operation. A solution to the visit problem enables the robot to carry out a set of operations

Manuscript received April 1, 1989; revised January 4, 1990.
N. S. V. Rao is with the Department of Computer Science, Old Dominion University, Norfolk, VA 23529-0162. Preliminary results on the part of this paper that deals with the visit problem have been presented at the 1988 IEEE Int. Conf. Robotics and Automation, Philadelphia, PA, April 25–29, 1988.
S. S. Iyengar is with the Department of Computer Science, Louisiana State University, Baton Rouge, LA 70803.
IEEE Log Number 9037604.

0018-9472/90/1100-1443$01.00 ©1990 IEEE

in different locations in unfamiliar environments. Since the motion planning in this case is essentially sensor-based, the robot may be required to perform a number of expensive sensor operations. Furthermore, the robot could temporarily navigate into local detours because of the partial nature of the information returned by the sensors. By incorporating the incidental learning feature, we reduce the expected number of sensor operations, and the expected number of detours, as the robot visits newer locations. Further, if the complete terrain model is available, the robot can avoid 1) local detours, 2) sensor operations. These two important points motivate the terrain model acquisition problem. In general, a dedicated rescue robot typically idles in between two rescue operations, and the rescue operations could be fairly infrequent. Thus there are definite advantages if the robot is employed in the terrain model acquisition process during this period. Our methodology provides a basic algorithmic framework that aids the design of a navigational system for the abovementioned rescue robot.

The organization of the paper is as follows: Preliminaries are presented in Section II. In Section III, we define the restricted visibility graph and the modified visibility graph, and prove some of their properties. rIn Section IV, we present solutions for the terrain model acquisition and the visit problems. We compare our method with the other methods in Section V.

II. Preliminaries

We consider a point robot R in two- and three-dimensional (3-D) terrains. Here, the location of R is also called the *position* of R. Additionally, in 2-D terrains, we consider a circular body R of radius δ, $(\delta \geqslant 0)$. The location of the center of R is called the *position* of R. The R houses a computational device with storage capability. The point robot R is capable of moving along a straight-line path in two- and three-dimensional terrains. Additionally, the circular R is capable of rotating around its center and also around a point on the circumference. R takes a finite amount of time to move through a finite amount of distance. Further, R is equipped with an algorithm B that plans a collision-free path (for R) through a known terrain. For example, in two dimensions, we can use the $O(N \log N)$ algorithm of O'Dunlaing and Yap [9] or Leven and Sharir [5] or Bhattacharya and Zorbas [1] to plan a path from a source location to a destination location for a circular robot, where N is the total number of obstacles corners. In three dimensions, R can use the algorithms of Reif [14]. For a circular robot, we can also use the algorithm of Chew [2] or Hershberger and Guibas [3], if shortest paths are required.

We consider a finite-sized *terrain* populated by a finite set $O=\{O_1,O_2,\cdots O_n\}$. Each O_i is called an obstacle; O_i is a simple polygon in the two-dimensional case and a polyhedron in the three dimensional case. In either case, O_i has a finite number of vertices. The terrain is completely unknown to R, i.e., the number of obstacles, and also the number and locations of vertices of each obstacle are unknown to R. The free-space is given by $\Omega = \bigcap_{i=1}^{n} O_i^C$, where O_i^C is the complement of O_i. The closure of the free-space is denoted by $\overline{\Omega}$. Let N denote the total number of vertices of all obstacles. A vertex v, of a polygon, is called *convex* if the angle included by the obstacle edges that are incident at v is less than Π. The vertex v, of an obstacle polygon, is called *nonconvex* if it is not convex. For two-dimensional terrains, let C be the number of nonconvex vertices.

We imagine a logical point of reference x on R for the sensor. A point $y \in \overline{\Omega}$ is said to be visible to R if the straight line joining x and y is entirely contained in $\overline{\Omega}$. R is equipped with a sensor that detects the maximal set of points on the obstacle boundaries that are visible from x. Such an operation is termed as the scan operation. We assume that a scan operation is error-free.

Two Navigational Problem

Initially, R is located at the position d_0 without intersecting any obstacle and at a finite distance from an obstacle. In the terrain model acquisition problem, R is required to acquire the model of the terrain to a degree such that it can navigate to any reachable destination location by planning a path using the known terrain algorithm B alone. In the case of a point robot this is tantamount to acquiring the entire model of the terrain. For a circular robot, an appropriate subset of the terrain boundary is to be identified depending on the radius of R and the initial location d_0. Note that after the terrain model is completely acquired, no sensor operations are needed for navigational purposes. Second, in the *visit problem*, R is required to visit the positions $d_1, d_2, \cdots, d_M$ in the specified order if there exists a path through these positions. If no such path exists, then R must report this fact in a finite amount of time.

Basic Algorithm

Here, R performs a "graph exploration type" of navigation using a combinatorial graph called the *navigation course*, $\xi(O)$, of the terrain O. A detailed treatment on this basic algorithm can be found in [11]. $\xi(O)$ is a 1-skeleton embedded in $\overline{\Omega}$. The nodes (edges) of $\xi(O)$ are called ξ-nodes (ξ-edges). In this paper, each ξ-node corresponds to an obstacle vertex, and specifies position for R such that it is entirely contained in $\overline{\Omega}$. For a point robot, a ξ-edge (v_1,v_2) specifies a line segment $v_1 + t(v_2 - v_1)$, $0 \leqslant t \leqslant 1$, that is entirely contained in $\overline{\Omega}$. Thus the edge provides a collision-free path to move from v_1 to v_2. For a circular robot, the edge (v_1,v_2) specifies a collision-free path, of finite length, from v_1 to v_2 for R. The $\xi(O)$ is initially unknown and it is incrementally constructed using the data obtained through the sensor operations. The navigational course $\xi(O)$ has to satisfy a set of properties, in order to yield correct solutions to the visit problem and the terrain model acquisition problem. The property of *local-constructibility*, means that the adjacency list of a ξ-vertex v can be computed from the information obtained by a scan operation performed from v. The *finiteness* property requires that $\xi(O)$ has a finite number of vertices. Also the graph *connectivity* property requires that any two ξ-vertices be connected by a path of ξ-edges. Now the fact that any graph exploration algorithm visits all the nodes of a finite connected graph in a finite amount of time, translates to the following observation:

Observation 1: If $\xi(O)$ satisfies the properties of finiteness, connectivity and local-constructibility, then, R, executing graph search algorithm, visits all vertices of $\xi(O)$ in a finite amount of time.

III. Navigational Courses

First, we present a $\xi(O)$ for a point robot, and in this case $\xi(O)$ is the visibility graph of O for three-dimensional terrains. For two-dimensional terrains, we consider the restricted visibility graph. For circular robots $(\delta > 0)$ we present a $\xi(O)$ based on the restricted visibility graph. In each of the cases we show that the proposed structure satisfies the properties of finiteness, connectivity, and local-constructibility. We also consider an additional property, namely *terrain-visibility*, which means that every point in the required subset of $\overline{\Omega}$, is visible from some ξ-vertex.

A. Point Robot

For a point robot we consider finite-sized 3-D terrains populated by polyhedral obstacles, i.e., O_i is a finite-sized polyhedron

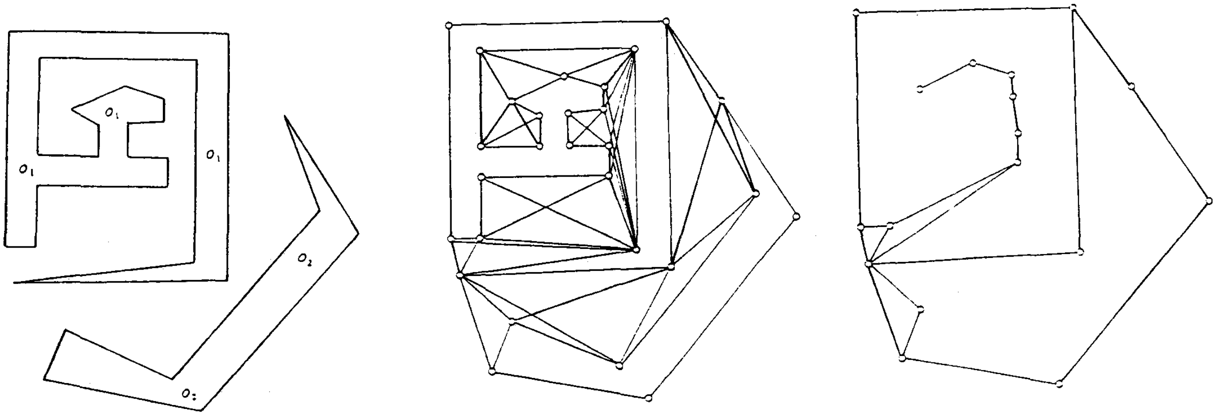


Fig. 1. Navigation courses based on visibility graphs. (a) Obstacle terrain $O = \{O_1, O_2\}$. (b) $VG(O)$. (c) $VG^*(O)$.

with a finite number of vertices. The visibility graph, $VG(O) = (V, E)$, of a terrain populated by the obstacle set O is defined as follows [6]: 1) V is the union of vertices of all obstacle polyhedra, 2) A line joining the vertices v_i and v_j forms an edge $(v_i, v_j) \in E$ if and only if it is either an obstacle edge or it is not intersected by any obstacle. See Fig. 1 for an example. The visibility graph is connected [13]. To show the terrain-visibility property, consider a point $x \in \bar{\Omega}$. Consider an infinitesimally small polyhedron P at x which can be imagined as a point at x. Consider the visibility graph of $O \cup \{P\}$. This graph is connected. Let v be a vertex of some $O_i \in O$ to which a vertex of P is connected. Then x is visible from v. We shall now summarize the properties of $VG(O)$:

Properties 1: The visibility graph $VG(O)$ satisfies the properties of finiteness, connectivity, local-constructibility and terrain-visibility.

We define the *restricted visibility graph* $VG^*(O) = (V, E)$ of a 2-D terrain O as follows: 1) V is the union of all convex vertices of obstacle polygons, 2) A line joining the vertices v_i and v_j forms an edge $(v_i, v_j) \in E$ if and only if it is either an obstacle edge of it is not intersected by any obstacle polygon. See Fig. 1 for an example. The $VG^*(O)$ is a subgraph of $VG(O)$, and it coincides with $VG(O)$ if every $O_i \in O$ is a convex polygon. The number of nodes of $VG^*(O)$ is $N - C$, where C is the number of non-convex obstacle vertices. In a general case where O contains non-convex vertices, the $VG^*(O)$ has a lesser number of nodes than $VG(O)$. We now have the following properties of $VG^*(O)$.

Lemma 1: The restricted visibility graph $VG^*(O)$ satisfies the properties of connectivity and terrain-visibility.

Proof: The key observation is that the shortest path between any two points in free-space is a polygonal path that runs through the obstacle vertices. Additionally we can show that such a path passes through convex obstacle vertices only. We can show this as follows: Let us say that the shortest path passes through a non-convex vertex v. Let v_1 and v_2 be the obstacles vertices adjacent to v on a shortest path i.e., the shortest path passes along the edges (v_1, v) and (v, v_2). Imagine a rubber band stretched (in the free-space) along the vertices v_1, v and v_2, and then released. The action of the rubber band can be visualized

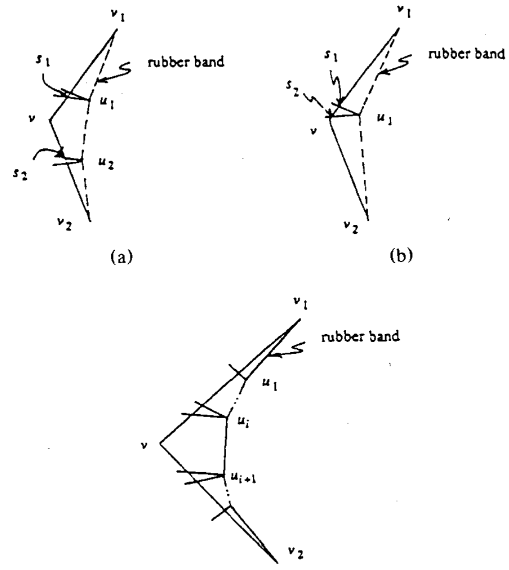


Fig. 2. Shortest path runs through the convex vertices only.

as follows: Imagine a long line segment (a ray) extending from v_1 through v. Rotate this ray around v_1 into the concavity until it encounters v_2 or a convex vertex, say u_1. Now rotate the ray around u_1 in a similar fashion. Note that each such rotation brings the line closer to v_2, and there can be only a finite number of rotations. Thus the rubber band will touch the convex vertices, say u_i, $i = 1, 2, \cdots k$, contained in the triangle formed by v_1, v and v_2 (see Fig. 2). It is clear from Fig. 2(a) and Fig. 2(b) that for cases $k = 1, 2$ the path followed by rubber band is shorter that the original path. For $k = 1$, draw perpendiculars at u_1 to segments $\overline{v_1 u_1}$ and $\overline{u_1 v_2}$. Here length of $\overline{v_1 u_1}$ ($\overline{u_1 v_2}$) is less than that of $\overline{v_1 s_1}$ ($\overline{s_2 v_2}$). Thus the path v_1, u_1, v_2 is shorter. If $k = 2$, the key idea is to note that the length of the original path contained in between the end perpendiculars of $\overline{u_1 u_2}$ is greater than or equal to the length of $\overline{u_1 u_2}$. Thus the path v_1, u_1, u_2, v_2 is shorter than v_1, v, v_2. For $k \geqslant 3$ we use the same argument. Draw perpendiculars at the end of each line segment joining u_i

and u_{i+1}. It is clear that the perpendiculars drawn at each u_i will include a positive angle. Now it is easy to see that for each segment $\overline{u_i u_{i+1}}$, the length of this segment is less than or equal to the length of the original path contained within the perpendiculars at u_i and u_{i+1}. Thus the path obtained by the rubber band is shorter than the original path. Thus the shortest path between any two points in the free-space is a polygonal path that runs exclusively through the convex obstacle vertices.

Now consider the shortest path between any convex obstacle vertices. By the above arguments these two vertices are connected by a polygonal path that runs exclusively through the convex obstacle vertices. This is precisely a path on the restricted visibility graph $VG^*(O)$. This proves the connectivity property of $VG^*(O)$. The terrain-visibility property of $VG^*(O)$ follows along the lines of that of the visibility graph. Hence the Lemma.

In summary we have the following properties.

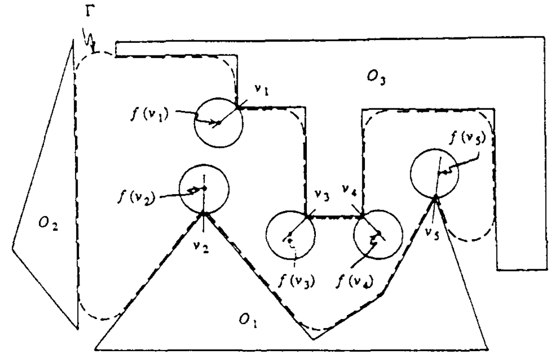
Properties 2: The restricted visibility graph $VG^*(O)$ satisfies the properties of finiteness, connectivity, terrain-visibility and local-constructibility.

B. Circular Robot

In this section, we define a family of graphs such that each of its members satisfies the required properties to be a navigational course. Consider the set FP of free-placements in which R is entirely contained in Ω. Note that the free-space Ω is an open polygonal region and the boundary of its closure is the boundary of $\bigcup_{i=1}^{n} O_i$, the union of obstacle polygons. The FP is composed of connected components, and let Ψ be the maximal connected component that contains the initial position x_0 of R. Any position of R connected to x_0 belongs to Ψ. Consider $\Gamma = \Psi \oplus R$, where $\oplus$ is the Minkowski sum, i.e., $\Gamma = \{x + y | x \in \Psi$ and $y \in R\}$. It is clear that Γ is an open connected set. The boundary of closure of Γ consists of intervals of edges of O_i's and circular arcs (possibly zero in number). The circular arcs are generated in the case when R is located in such a way that its closure intersects two distinct objects; an object is an obstacle vertex or an obstacle edge. Each such circular arc is formed by a unique pair of points at which $\overline{R}$ intersects boundary of obstacles; each such point is called the *end-vertex* and the corresponding pair is called the *end-pair*. Note that an end-vertex is either an obstacle vertex or a point on an obstacle edge.

Let $\theta(v)$ denote the angle subtended by an obstacle at its vertex v. Let the *equidistance line* of a convex vertex v, denoted by $EL(v)$, be a portion of the bisector of $\theta(v)$ that extends from v to the outwards of the obstacle. Now we have the property that any obstacle vertex contained in $\overline{\Gamma}$ is a convex vertex. These convex vertices can belong to one of the two categories. First category consists of all the convex vertices that form an end-pair. And second category consists of all *free* vertices which are convex vertices contained in $\overline{\Gamma}$ and do not form an end-pair. Note that by definition, we can place R so that it touches a free vertex v and we can rotate it around v. Let $\overline{v_1 v}$ and $\overline{v v_2}$ be the segments of obstacle edges contained in Γ. We can slide R along $v_1 v$ to v (at least through infinitesimally small distance) and rotate it around v and then slide it along the edge to $\overline{v v_2}$. Then during the rotation the center of R meets $EL(v)$ at one position. This shows that all points on $EL(v)$ within a distance of 2δ (from a free vertex v) are in $\overline{\Omega}$.

Let V be the union of the free vertices contained in $\overline{\Gamma}$. Consider a function f: $V \to \bigcup_{v \in V} EL(v)$ called the *sensing function*. This function assigns a unique point on $EL(v)$ for each $v \in V$, i.e. $f(v) \in EL(v)$. The *modified visibility graph* (MVG) of the obstacle terrain O with respect to a sensing function f, denoted by $VG_f(O)$, is a graph (V, E) such that there exists an edge $(v, w) \in E$ if and only if the line joining w and $f(v)$ lies entirely in $\overline{\Gamma}$, and does not cross the boundary of $\overline{\Gamma}$. For a given obstacle terrain O, there exists a *family* of modified visibility graphs, denoted by $\{VG_f(O)\}$ corresponding to all possible fs. See Fig. 3 for an example of $VG_f(O)$. We have the following lemma:



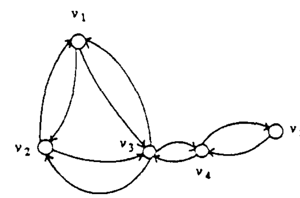


Fig. 3. Example of $VG_f(O)$. (a) Definition of f and the positions of R at the free vertices. (b) The graph $VG_f(O)$ for f and O of (a).

Lemma 2: The modified visibility graph $VG_f(O) \in \{VG_f(O)\}$ satisfies the connectivity, and terrain-visibility properties for all f such that $\|v - f(v)\| < 2\delta$, for all $v \in V$.

Proof: We first discuss the connectivity property. Consider 1) two free vertices $v_1, v_2 \in V$. Consider a shortest path from v_1 to v_2 that runs through $\overline{\Gamma}$ such that the path does not cross the boundary of $\overline{\Gamma}$. Such a path exists because Γ is a connected set. This path runs through only convex vertices of Γ. Using the arguments similar to those in the proof of Lemma 1 (using rubber band) we can show that the path runs through only the free vertices of Γ. Here the convex vertices that form an end-pair can be essentially treated as concave corners, and it the shortest path can be shown not to pass through them. Consider an edge (v_1, v_2) of such shortest path. Now consider a rubber band stretched from v_1 to v_2. Then move the v_1 end of the rubber band along $EL(v_1)$ to $f(v_1)$. In this state the rubber band might touch some other free vertices. Let the rubber band run through the free vertices $u_1, u_2, \cdots, u_r$. Here u_1 is visible from $f(v_1)$. Hence (v_1, u_1) is an edge of $VG_f(O)$. Apply the same technique from each of u_i's. It is clear that there is a path from v_1 to v_2. Thus the $VG_f(O)$ is connected.

Now consider the terrain-visibility property. Consider $x \in \Omega$. Now consider a shortest path from x to a free vertex u such that the path lies entirely in Γ as described above. Move on this path from x to the first free vertex u. Then imagine a rubber band stretched from x to u, and move its u end along $EL(u)$ to $f(u)$. If the line from x to $f(u)$ is not intercepted by any obstacle then we are done. Otherwise move from x along the stretched rubber band to the first free vertex, and apply the same procedure. The repeated application of the procedure results in free vertex u_1 such that x is visible from $f(u_1)$. Hence the Lemma.

It is clear that $VG_f(O)$ has at most $N-C$ vertices and $O(N^2)$ edges. Note that all free vertices that are visible from $f(v)$ can be obtained from the information from a single scan. Thus $VG_f(O)$ satisfies the local-constructibility property. We summarize all these properties as follows.

Properties 3: The graph $VG_f(O)$ for an f that satisfies the condition stated in Lemma 2, satisfies the properties of finiteness, connectivity, terrain-visibility and local-constructibility.

Chew [2] proposed the *path graph* that is an extension of the visibility graph. This path graph is used to plan an optimal path between two points through a two-dimensional terrain, and this graph has $O(N^2)$ vertices and $O(N^4)$ edges. This path graph can be used as a $\xi(O)$ for a circular robot. The modified visibility graph contains at most N vertices, which is important because the required number of scan operations in the solution to terrain model acquisition problem and the visit problem (in a worst-case) is equal to the number of vertices of $\xi(O)$.

IV. Navigation Algorithms

A. Circular Robot

A vertex of $VG_f(O)$ is a convex obstacle vertex v contained in $\overline{\Gamma}$ such that it does not form an end-pair. Consequently, we can place R such that it touches v since $v \in \overline{\Gamma}$. Since v does not form an end-pair, we can rotate R around v such that its center moves along a circular arc of radius δ. This arc extends between the perpendiculars to the obstacle edges incident on v. The Minkowski sum of R and this arc is free of obstacles. A vertex v of $VG_f(O)$ defines a position for R as follows. It is clear that R can be located such that its center lies on $EL(v)$ at a distance of exactly δ from v. Then $f(v)$ precisely defines the 'logical' position of sensor corresponding to vertex v. First, R locates its center on $EL(v)$ at a distance of δ from v. Then R rotates around its center until the reference point of the sensor lies on $EL(v)$. The R can rotate either clockwise or anti-clockwise to achieve this and in either case the logical position of the sensor corresponding to $f(v)$ that satisfies the condition in Lemma 2, i.e., $\|v - f(v)\| < 2\delta$. Thus a vertex v of $VG_f(O)$ specifies a position for R and for the sensor. Further, we use the depth-first graph search for R, which chooses a ξ-vertex that is closest to the present location of R. Subsequently, we establish the following aspects: (a) the information stored along the edges of $VG_f(O)$ suffices for the intermediate navigation that is required to move R from one vertex to the other, and (b) the appropriate vertices and adjacency lists of $VG_f(O)$ can be correctly computed from the scan information.

1) Navigation Along Edges: Consider the navigation of R from u to v, $u, v \in V$. Now $EL(u)$ is known, and $EL(v)$ may or may not be known. When v is detected, the portions (that are close to v) of the edges that are incident at v will be visible in a scan operation. If both the edges incident on v are visible during an earlier scan operation then $EL(v)$ can be computed. Note that at least an infinitesimally small portion of one of the edges incident at v will be seen in the scan operation in which v is detected. If the $EL(v)$ is known then, the navigation from u to v is carried out as follows: The subset of Ψ that corresponds to the free-space visible from $f(u)$ is computed (as subsequently described). Let v_1 (u_1) be a point on $EL(v)$ $(EL(u))$ at a distance δ from v (u). Consider $\overline{u_1 v_1}$ the line joining u_1 and v_1. This line intersects the boundary of the computed part of Ψ zero or more times. R moves along the $\overline{u_1 v_1}$ in the portions that lie in the free-space, and follows the computed boundary of free-space in the other portions of $\overline{u_1 v_1}$. There are only a finite number of detours during which R follows the boundary of free-space, and each detour specifies only a finite number of translational and rotational motions for R. If $EL(v)$ is not known, then R moves from u_1 to a point v_2 at a distance δ from v and lies on the perpendicular to the known edge of v. The motion of R from u_1 to v_2 can be handled similar to the above case. From v_2, R rotates around v until it can not rotate further. Then $EL(v)$ is computed and R rotates back it its position on $EL(v)$. The path corresponding to the navigation along an edge of $VG_f(O)$ is computed the first time R moves along this edge. This path is stored and used in subsequent traversals along this edge. In summary we have the following lemma.

Lemma 3: R can compute a path of finite number of translations and rotations to navigate along an edge of $VG_f(O)$.

2) Processing Scan Information: The scan information is to be processed so that the portion of $VG_f(O)$ corresponding to the "seen part" is constructed. We can use a variation of the algorithm of [5] to compute this part. More specifically, we compute the vertices of local non-convexity corresponding to the Minkowski sum of the disc corresponding to R and the visibility polygon returned by the sensor. Here the Minkowski sum is bounded by line segments and the circular arcs. The vertices corresponding to the arcs that do not correspond to points of non-convexity are the nodes of $VG_f(O)$. Conservatively, the complexity of this operation is $O(N \log N)$.

B. Terrain Model Acquisition

The algorithm *ACQUIRE*, for terrain model acquisition, is a direct implementation of a graph search algorithm. From the observation 1, R will visit all the ξ-vertices in a finite amount of time. And by the terrain visibility property $\xi(O), R$ would have seen the required portions of the free-space, after visiting all ξ-vertices. For completeness we state the following theorem which can be proved along the lines of [13].

Theorem 1: The algorithm *ACQUIRE* solves the terrain model acquisition problem in a finite amount of time such that for a point robot and a circular robot, the number of scan operations performed is $N-C$ in 3-D terrains. For a point robot the number of scan operations is N in 3-D terrains. The complexities of various tasks carried out by *ACQUIRE* are as follows: 1) the storage complexity is $O(N^2)$, 2) the cost of construction of $\xi(O)$ is $O(N^2 \log N)$ and $O(N^2)$ for circular robot and point robot respectively, 3) the total cost of path planning is $O(N^3)$.

C. Visit Problem

The algorithm *LNAV*, the navigates R from d_i to d_{i+1}, is obtained by simulating the graph search algorithm. Initially a scan operation is performed from d_i and if d_{i+1} is found reachable, then R moves to d_{i+1}. If d_{i+1} is not found reachable then R computes a ξ-vertex v_0 and moves to v_0. From v_0, the graph search algorithm *NAV* is invoked. Let R be located at v. After a scan is performed from v, R checks if d_{i+1} is reachable. If d_{i+1} is reachable, then R moves to d_{i+1} and terminates *NAV*. If not, R continues to execute *NAV* until the d_{i+1} is found reachable or until completion. The following theorem can be established by specializing the result in [11].

Theorem 2: Algorithm *LNAV* navigates R from d_i to d_{i+1} in a finite amount of time if the latter is reachable. If d_{i+1} is not reachable then R declares so in a finite amount of time. In executing the algorithm *LNAV* by a point R, the number of scan operations is at most $N-C$ and N respectively for two-

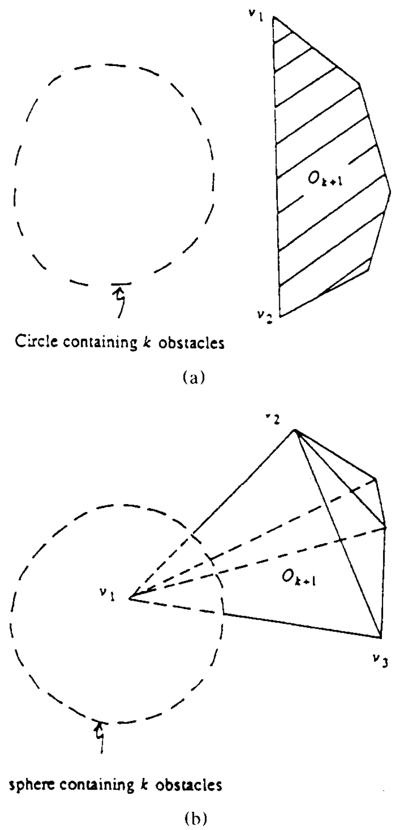


Fig. 4. Addition of O_{k+1} in the proof of Theorem 5. (a) Two-dimensional case. (b) Three-dimensional case.

and three-dimensional terrains. For a circular R, the number of scan operations is at most $N - C$ in two-dimensional terrains.

The computational complexity of *LNAV* is similar to that of *ACQUIRE*. We obtain the algorithm *GNAV* by extending *LNAV* as follows: We store the adjacency lists computed by R during earlier traversals. Further we store S, which is the set of all vertices that have been detected but not visited yet. Consider the navigation from d_i to d_{i+1}. Then *GNAV* computes a ξ-vertex that is reachable from d_i and moves to this vertex. Then R computes a ξ-vertex d^* that is closest to d_{i+1} according to some criterion such as distance. Then R moves along a path on $\xi(O)$ to d^*. From d^*, R uses *LNAV* to navigate to d_{i+1}. It is direct to see that *GNAV* correctly solves the visit problem. Moreover, R checks the set S after every scan operation. After S becomes empty, R switches-off its sensor and navigates using the algorithm B alone. At this stage R has acquired the terrain model that is sufficient to navigate to any reachable point [11]. Thus we have the following theorem.

Theorem 3: The terrain model will be completely built by R in at most $N + M - C$ and $N + M$ scans respectively for 2-D and 3-D terrains for a point R, then the execution of each traversal involves no scan operations. For a circular R the performance is same as that of a point robot in 2-D terrains.

Here the process by which R acquires the terrain is *incidental*, i.e., the present model of the terrain depends on the previous traversals. Let p_v, $v \in V$, $(\xi(O) = (V, E))$ be the probability that R visits v during a traversal. In this case, we can show that for $M \geqslant 1/\min_{v \in V}\{p_v\}$, the expected number of scan operations performed by *GNAV* is strictly less than the expected number of scan operations performed by *LNAV* [11]. Now consider that R has successfully navigated to d_i and it is now required to navigate to d_{i+1}. Let s_L and s_G be the random variables that denote the number of scan operations performed by R in cases of using *LNAV* and *GNAV* respectively in navigating from d_i to d_{i+1}. Let $E[x]$ denote the expected value of the random variable x. It is direct to see the following results: 1) $E[s_G]/E[s_L] < 1$, for $p_v < 1$, for $v \in V$, 2) $E[s_G] \to 0$, for $i \to \infty$.

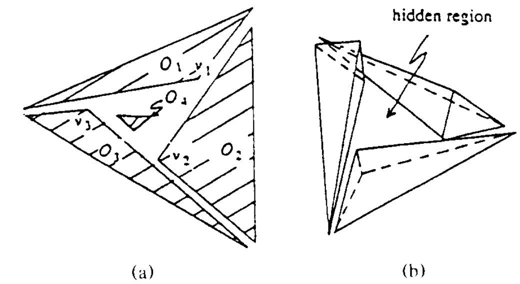


Fig. 5. Vertices can not be randomly skipped. (a) Two-dimensional case. (b) Three-dimensional case.

E. Lower Bound on Number of Scan Operations

We discuss the case of a point robot in two and three-dimensional terrains. We obtain a lower bound on the number of scan operations that are occasionally necessary. These algorithms are required to ensure that at the time of termination every point in the free space is "sensed." Consider a terrain of one obstacle. Now during execution of the algorithm, no more than *one* (*two*) vertices per obstacle can be left unexplored in a 2-D (3-D) terrains. If R starts at a vertex it detects one new vertex with one scan operation (except when the first vertex is explored) 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. For the three-dimensional terrain, 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.

Theorem 4: In the execution of *ACQUIRE* or *LNAV*, for a given positive integer n there exists a terrain $\{O_1, O_2, \cdots, O_n\}$ of polygonal and polyhedral obstacles such that the necessary number of scan operations is $N - n$ and $N - 2n$ for two and three dimensional terrains respectively.

Proof: We use induction on the number of obstacles in the terrain. For $n = 1$ the claim is true as explained above. Now, assume that the claim is true for $n = k$. Let the set of obstacles in this case be $\{O_1, O_2, \cdots, O_k\}$. 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 of k obstacles (that satisfies the induction hypothesis) as shown in Fig. 4(a). 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

TABLE I
COMPUTATIONAL COMPLEXITY

Quantity for Comparison	Restricted Visibility Graph	Modified Visibility Graph	Retraction Method
Storage	$O(N^2)$	$O(N^2)$	$O(N)$
Construction	$O(N^2)$	$O(N^2 \log N)$	$O(N^2 \log N)$
Path planning	$O(N^3)$	$O(N^3)$	$O(N^2\sqrt{\log N})$
Overall time complexity	$O(N^3)$	$O(N^3)$	$O(N^2 \log N)$

$N(O_{k+1})-1$, where $N(O_{k+1})$ is the number of vertices of O_{k+1}. For three-dimensional terrains the obstacle O_{k+1} is such that the plane formed by three vertices v_1, v_2 and v_3 obscures the rest of the obstacle from the scan operations performed in the terrain of k obstacles as in Fig. 4(b). The O_{k+1} lies outside the sphere the encloses the terrain of k obstacles. Thus the necessary number of scan operations to acquire O_{k+1} is $N(O_{k+1})-2$. Thus the theorem follows by 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 if the vertices are to be arbitrarily skipped. This is illustrated in Fig. 5(a). In two dimensions, if the robot skips the vertices v_1, v_2 and v_3 then the obstacle O_4 will not be detected. Fig. 5(b) shows a 3-D example.

V. Comparison of Performance

We use the worst-case execution of the algorithm *LNAV* or equivalently an invocation of algorithm *ACQUIRE* as a basis for comparison. We consider two-dimensional terrains. We compare the visibility graph methods with the retraction methods of [12] (based on the Voronoi diagram of O). R using the visibility graph methods, may be required to navigate along the boundaries of the obstacles. The paths based on the retraction method always keep R as far away from the obstacle boundaries as possible. In general, the paths generated by the retraction methods tend to be longer than those generated by the visibility graph methods. Using the visibility graph methods, a point robot always navigates along line segments. A circular robot using the visibility graph method will be required to rotate around a vertex. Whereas a point robot or a circular robot will be required to navigate along line segments and second order curves (the parabolic Voronoi edges) in the retraction method. In our methods, for point circular robots, the number of scan operations is at most $N-C$. In the retraction method, the upper bound on the number of scan operations is $4N-n-C-2$.

A summary of the computational complexities is presented in Table I. Consider point robots. It is clear that the adjacency list of the restricted visibility graph can be directly obtained from the scan information. Thus the construction cost for this case is $O(N^2)$ as opposed to the construction cost of $O(N^2 \log N)$ of the retraction based method. Similarly the retraction method has a better complexity for the path planning operations. In terms of the total computational complexity the retraction method has better complexity of $O(N^2 \log N)$ compared to $O(N^3)$ of the visibility graph method. Note that the overall time complexity of the visibility graph based method is dominated by the path planning part whereas that of the retraction method is dominated by the construction cost. Further more the storage complexity in case of retraction methods is $O(N)$ as opposed to $O(N^2)$ of the visibility graph method. For circular robots, the situation remains more or less the same, except that the construction cost of the modified visibility graph is $O(N^2 \log N)$. Thus, in both the cases, the retraction method has better overall time complexity compared to that of visibility graph method.

VI. Conclusion

We presented a framework that solves both the visit problem and the terrain model acquisition problem using a single approach of implementing a graph search on an incrementally constructed geometric structure called the navigational course. A point robot employs the restricted visibility graph and the visibility graph in two and three dimensions respectively. The restricted visibility graph extends the existing solution of [10] to non-convex obstacles for the visit problem. Further, it is better in terms of the bound on the number of scan operations if the terrain contains non-convex corners. A circular robot employs a modified visibility graph in two dimensions. We analyze the algorithms that solve both the visit problem and the terrain model acquisition problem. The proposed framework could be extended to consider more detailed models for the mobile robots in terms of geometric shape, and motion primitives. It would also be interesting to see if there exist general principles to design navigational courses in more detailed cases.

References

[1] B. K. Bhattacharya, and J. Zorbas, "Solving the two-dimensional findpath problem using a line-triangle representation of the robot," *J. Algorithms*, vol. 9, pp. 449–469, 1988.

[2] L. P. Chew, (1985), "Planning shortest path for a disc in $O(n^2 \log n)$ time, in *Proc. Symp. on Computational Geometry*, pp. 214–220, 1985.

[3] J. Hershberger and L. J. Guibas, (1988), "An $O(n^2)$ shortest path algorithm for a nonrotating convex body," *J. Algorithms*, vol. 9, pp. 18–46, 1988.

[4] S. S. Iyengar, C. C. Jorgensen, S. V. N. Rao, and C. R. Weisbin, "Robot navigation algorithms using learned spatial graphs," *Robotica*, vol. 4, pp. 93–100, Jan. 1986.

[5] D. Leven and M. Sharir, "Planning a purely translational motion for a convex object in two-dimensional space using generalized Voronoi diagrams," *Discrete and Computational Geometry*, vol. 2, pp. 9–31, 1987.

[6] T. Lozano-Perez, and M. Wesley, "An algorithm for planning collision-free paths among polyhedral obstacles," *Commun. ACM*, vol. 22, pp. 560–570, 1979.

[7] V. J. Lumelsky, "Algorithmic and complexity issues of robot motion in an uncertain environment," *J. Complexity*, vol. 3, pp. 146–182, 1987.

[8] V. J. Lumelsky, and A. A. Stepanov, "Path-planning strategies for a point mobile automaton moving amidst unknown obstacles of arbitrary shape," *Algorithmica*, vol. 2, pp. 403–430, 1987.

[9] C. O'Dunlaing, and C. K. Yap, "A retraction method for planning the motion of a disc," *J. Algorithms*, vol. 6, pp. 104–111, 1985.

[10] J. B. Oommen, S. S. Iyengar, S. V. N. Rao, and R. L. Kashyap, Robot navigation in unknown terrains using visibility graphs. Part I: The disjoint convex obstacle case, *IEEE J. Robotics and Automation*, vol. RA-12, pp. 672–681, 1987.

[11] N. S. V. Rao, "An algorithmic framework for learned robot navigation in unknown terrains, *IEEE Computer*, pp. 37–43, July 1989.

[12] N. S. V. Rao, N. Stoltzfus and S. S. Iyengar, "A 'retraction' method for terrain model acquisition," *Proc. 1988 IEEE Int. Conf. Robotics and Automation*, pp. 1224–1229, 1988.

[13] N. S. V. Rao, S. S. Iyengar, J. B. Oommen and R. L. Kashyap, Terrain acquisition by a point robot amidst polyhedral obstacles, *IEEE J. Robotics and Automation*, vol. 4, no. 4, pp. 450–451, 1988.

[14] J. Reif, "Complexity of mover's problems and generalizations," *Proc. 20th Ann. Symp. Found. Comput. Sci.*, pp. 421–427, 1979.

[15] J. T. Schwartz and M. Sharir, "On the piano-movers problem: I. The case of two dimensional rigid body moving amidst polygonal barriers," *Comm. Pure Appl. Math.*, vol. 36, pp. 345–398, 1983.

[16] M. Sharir, "Algorithmic motion planning in robotics," *IEEE Computer*, pp. 9–20, Mar. 1989.

[17] C. K. Yap, "Algorithmic motion planning," in *Advances in Robotics: Vol. 1: Algorithmic and Geometric Aspects of Robotics*, Ed. J. T. Schwartz and C. K. Yap. Hillsdale, NJ: Lawrence Erlbaum Associated Pub., 1987, pp. 95–144.