Robot Navigation in Unknown Generalized Polygonal Terrains Using Vision Sensors

Nageswara S. V. Rao

Abstract— **This paper considers the problem of navigating a point robot in an unknown two-dimensional terrain populated by disjoint generalized polygonal obstacles. A generalized polygon consists of a connected sequence of circular arcs and straight-line segments. The terrain model is not known a priori, but the robot is equipped with a vision sensor. A discrete vision sensor detects all visible (from a single position) portions of the obstacle boundaries in a single scan operation. The navigation problem deals with moving the robot through the terrain from a source position to a destination position, and the terrain model acquisition problem deals with autonomously building a model of the terrain. A complete solution to either problem is shown to require an infinite number of scan operations in cusp regions formed by a pair of convex and concave obstacle edges. Either problem is considered solved with a precision ϵ if the points that have not been scanned are those in a cusp region with a clearance less than ϵ from two obstacle edges. Three methods are proposed to solve both problems with a precision ϵ based on extensions of the generalized visibility graph, the generalized Voronoi diagram, and the trapezoidal decomposition. Then simplified versions of these structures are proposed to exactly solve the navigation and terrain model acquisition problems using a continuous vision sensor that detects all visible obstacle boundaries as the robot navigates along a path.**

I. Introduction

PATH planning and navigation are very important components in the operation of an autonomous mobile robot. In the past decade, various formulations of this fascinating problem have been solved by several researchers; see the recent book by Latombe [14] and the survey paper by Hwang and Ahuja [11] for a comprehensive treatment of this topic. The navigation problem deals with computing a collision-free path for a robot from a source position to a destination position in a terrain populated with obstacles. In a known terrain, a complete model of the terrain is available, and the path planning can be performed using several techniques such as retraction, decomposition, etc.; see Sharir [33] for an overview of these methods for known terrains. This paper deals with path planning in an unknown terrain, where the model of the terrain is not known but a sensor system is employed for navigational purposes. The navigation algorithms for unknown terrains are often referred to as the on-line navigation algorithms. In particular, we are interested in nonheuristic algorithms that can be shown to be correct within a stated framework of models for the robot, terrain and sensor system; a survey of these algorithms can be found in Rao *et al.* [30] and a comprehensive treatment of nonheuristic algorithms based on touch sensors can be found in Lumelsky [18]. For the main part this paper, we consider a point mobile robot equipped with a discrete vision sensor, which in a scan operation detects all visible parts of the obstacle boundary from the present location. We then briefly consider a continuous vision sensor that detects all visible parts of the obstacle boundaries as the robot moves along a path. In general, the operation of a continuous vision sensor cannot be simulated by a discrete vision sensor if only a finite number of discrete scan operations are allowed.

Nonheuristic algorithms for the robot navigation in unknown terrains have been studied by a number of researchers in the last decade. The navigation algorithms for a point robot equipped with a touch sensor were pioneered by Lumelsky and Stepanov [21], and have been extensively studied in subsequent work [22]. The case of a point robot equipped with a discrete vision sensor is studied by Rao [27] for terrains populated by polygonal obstacles; navigation of a polygonal robot capable of translational motion in the same type of terrains with the discrete vision sensors is studied by Foux *et al.* [8]. The algorithms for a point robot equipped with a continuous vision sensor are developed by Sutherland [35], and Lumelsky and Skewis [20]. Cox and Yap [7] present an algorithm to navigate a ladder (rod) with touch sensing capability in an unknown terrain of polygonal obstacles. Recently, Rimon and Canny [32] present algorithms to navigate in polyhedral terrains using abstract sensors called the critical point detectors and minimum passage detectors. The algorithms of Lumelsky and Stepanov [21], [20], consider the obstacles of arbitrary shapes. In these algorithms, the robot does not store the information about the parts of the terrain that it has sensed. On the other hand, the algorithms of Oommen *et al.* [24] and Rao [27] implement incidental learning, where the robot builds a model of the terrain by consolidating the sensory information. Their work, however, is applicable to point robots navigating only in polygonal terrains; furthermore the former is restricted only to convex polygonal obstacles. The navigation of circular robots in two-dimensional polygonal

Manuscript received August 27, 1993; revised August 5, 1994. This work was sponsored by the Engineering Research Program of the Office of Basic Energy Sciences, of the U.S. Department of Energy, under Contract DE-AC05-840R21400 with Martin Marietta Energy Systems, Inc. In addition, this work is also partially funded by National Science Foundation under Grant IRI-9108610 and by Old Dominion University Summer Faculty Award for 1991. A preliminary version of this work was presented at 1991 International Conference on Systems, Man and Cybernetics, Charlottesville, VA. The submitted manuscript has been authored by a contractor of the U.S. Government under Contract DE-AC05-840R21400.

The author was with the Department of Computer Science, Old Dominion University, Norfolk, VA 23529-0162 USA. He is currently with the Center for Engineering Systems Advanced Research, Oak Ridge National Laboratory, Oak Ridge, TN 37831-6364 USA.

IEEE Log Number 9410384.

obstacles has been studied using techniques based on the visibility graph [28] and the Voronoi diagram [31]. More recently, the nonheuristic algorithms that guarantee bounds on the distance traversed or ratio of the distance traversed to the shortest path length have received the attention of several researchers; see Baeza-Yates [2], Papadimitriou and Yannakakis [25], Blum *et al.* [5], Bar-Eli *et al.* [3], and Klein [13] for interesting formulations in this framework.

The terrain model acquisition problem deals with acquiring a complete model of the terrain by systematically visiting portions of the terrain. The motivation for this problem is that once the terrain model is completely built, the navigation algorithms of known terrains can be employed. Hence, the sensor can be switched off (at least in theory), thereby avoiding acquisition and processing of sensor data. Further, paths with the shortest distance between the start and goal positions can be computed using the terrain model (for example, using the algorithm of Laumond [15] for the present formulation); note that if the terrain model is not available, no algorithm can guarantee the shortest paths always. The terrain model acquisition problem for three-dimensional polyhedral terrains has been solved by using the visibility graph structure by Rao *et al.* [29] for the case of a discrete vision sensor. In the plane, the restricted visibility graph, which is obtained by removing all nonconvex vertices from the visibility graph, is shown to suffice [28]. The same problem can also be solved by using a method based on the Voronoi diagram [31]. The terrain model acquisition problem in the case of a robot with a continuous vision sensor has been solved by Lumelsky *et al.* [19].

In the case of a discrete vision sensor, there is a unifying theme behind the solutions of the navigation and terrain model acquisition problems based on a structure called the navigation course. The navigation course is a 1-skeleton (a collection of one-dimensional curves) embedded in the set of all free-positions of the robot. Rao [27] showed that if the navigation course satisfies the four properties of finiteness, connectivity, terrain-visibility and local-constructibility (precise definitions of these properties are given in Section II), then both the problems can be solved by employing a graph search algorithm on the navigation course. Informally, finiteness requires that the number of vertices and edges of the navigation course be finite; connectivity requires the topological connectedness of the navigation course; terrain visibility requires that every point in the free-space be visible from some vertices of the navigation course; and the local-constructibility requires that the navigation course be incrementally constructible by using the sensor information. For polygonal terrains, we can employ navigational structures based on the visibility graphs and the Voronoi diagrams [27]. In this paper, we present another method based on the trapezoidal decomposition of free-space; the idea of this method was first proposed by Kim [12], but we are unaware of the analysis of this method. It is not direct to extend the navigational algorithms for polygonal terrains to the generalized polygonal terrains where the boundary of each obstacle is a connected sequence of straight-line segments and circular arcs. As will be discussed later, the properties of terrain-visibility and local-constructibility are nontrivial to establish in this case.

Any navigation algorithm for generalized polygonal terrains (including the algorithms presented in this paper) can be used to solve the navigation problem for a circular robot in polygonal terrains. By using the technique of "obstacle growing," the path planning problem for a circular robot of radius r in polygonal terrains can be reduced to that for a point robot, where each obstacle is expanded by taking its Minkowski sum with a circle of radius r [17]. A critical point to note is that in the "grown" terrain, each circular segment of the boundary has the same radius. However, the transformation of path planning problem for a point robot in generalized polygonal terrains to that of a circular robot in polygonal terrains is, in general, not that direct (if indeed possible). The main difficulty seems to be that there is no easy way of taking into account the different radii of the boundaries of the obstacles. Extension of a path planning algorithm of a point robot to a circular one has been proven to be nontrivial even in known terrains. For example, the solution to the problem of planning a shortest path for a circular robot does not directly follow from that for a point robot as shown in Chew [6]; a generalization of the idea behind this extension is presented by Hershberger and Guibas [10]. The problem of navigating a point robot in a "known" generalized polygonal terrain has been solved by Laumond [15] by using the generalized visibility graph. The retraction method of O'Dunlaing and Yap [23] can be used to solve this problem by using the Voronoi diagram proposed by Yap [35]. In general, the structures of known terrains satisfy the properties of finiteness and connectivity and some augmentation to these structures is necessary to ensure the properties of terrain-visibility and local-constructibility. In addition to the retraction and visibility graph methods, we also propose a method based on a trapezoidal decomposition of the free-space.

We first show a general result that a solution to the navigation problem (or terrain model acquisition problem) could require an infinite number of scan operations in cusp regions formed by a pair of convex and concave obstacle edges. Thus if the obstacle boundaries consist of segments which are of higher algebraic complexity than a straight line, then it is not possible to completely solve the navigation problem using a vision sensor capable of only discrete scan operations. In a practical scenario consisting of curved objects, we either have to use additional sensors such as touch sensors, or continuous vision sensors, etc., or solve the problem approximately. We first follow the latter approach in this paper, and then show how a continuous vision sensor can be employed to exactly solve the problem.

The navigation or terrain model acquisition problem is considered to be solved with a precision ϵ if only parts of the free-space that are not scanned consist of points within a distance of ϵ from two obstacle edges in a cusp region; these points can be imagined to constitute fictitious obstacles. Intuitively, we ignore "narrow and curved corridors" by regarding them as obstacles. If the robot has nonzero dimensions, it would be natural to choose a value for ϵ such that the robot cannot be placed in the regions of fictitious obstacles without intersecting the obstacles.

We propose three navigational structures for generalized polygonal terrains that yield solutions to both the navigation and terrain model acquisition problems with a precision ϵ. The first one is obtained by suitably extending the generalized visibility graph (GVG) of Laumond [15] so that it satisfies the properties of terrain-visibility and local-constructibility. The second structure is obtained by extending the generalized Voronoi diagram (GVD) of Yap [35]. The third structure is obtained by using a dual graph based on the trapezoidal decomposition of the free-space.

We then consider the case of continuous vision sensors. The properties of connectivity and terrain-visibility are sufficient to ensure that a search algorithm invoked on the navigation course will solve the navigation and terrain model acquisition problems. We propose navigation courses with these properties by using simplified versions of the above three structures.

The organization of this paper is as follows. We discuss the definitions of the generalized visibility graph, Voronoi diagram, and dual graph based on the trapezoidal decomposition, and an algorithmic framework for the navigation in unknown terrains in Section II. In Section III, we present a result on the number of scan operations needed in solving the navigation and terrain model acquisition problems. In Section IV, we present the augmented versions of the generalized visibility graph, Voronoi diagram and dual graph based on trapezoidal decomposition that are suitable for navigation in unknown terrains using a discrete vision sensor. The algorithms for the navigational problem and terrain model acquisition problem are briefly discussed in Section V. In Section VI, simplified navigational courses based on the structures of Section IV are shown to provide solutions for the navigation and terrain model acquisition problems using a continuous vision sensor.

II. Preliminaries

We consider a finite two-dimensional terrain populated by a finite and nonintersecting set $O = \{O_1, O_2, \cdots, O_n\}$ of generalized polygons. Each obstacle $O_i \in O$ is a generalized polygon with a finite number of vertices, whose boundary is a sequence of circular arcs and straight-line segments. Let N denote the total number of obstacle vertices. A convex arc is the circular arc boundary of an obstacle such that the obstacle locally lies to the right of the local tangent as we move along the arc in the clockwise direction. If the obstacle locally extends to the left of the local tangent as we move along the arc in the clockwise direction, then the arc is called concave. An obstacle vertex is called a convex vertex if the angle included inside the obstacle by the edges that meet at this vertex is less than 180 degrees, and the obstacle vertex is called concave otherwise.

Let the free-space, denoted by Ω, be the subset of the plane given by $\cap_{i=1}^{n} O_i^C$, where O_i^C denotes the complement of O_i in the plane. Let $\overline{\Omega}$ denote the closure of Ω. Two points p and q in $\overline{\Omega}$ are visible to each other if the line segment joining p and q, denoted by $\overline{pq}$, is completely contained in $\overline{\Omega}$.

The robot, denoted by R, is point-sized and equipped with a vision sensor. A discrete vision sensor is characterized by a scan operation: A scan operation performed from a position (point) p returns the visibility polygon of p, which is the general polygonal region consisting of all points in the terrain visible to p (Fig. 1(a)). A continuous vision sensor when invoked as the robot moves along a path P returns the generalized polygonal region such that every point of this polygon is visible from some point on P (Fig. 1(b)), i.e., the polygon returned in this case is the union of the visibility polygons of all points on P. Thus, in general, the operation of a continuous vision sensor cannot be simulated by a discrete vision sensor if only a finite number of discrete scan operations are allowed. For the purpose of navigation, discrete vision sensors are sufficient for a terrain with polygonal obstacles, but not so with generalized polygonal obstacles. Continuous vision sensors are, however, adequate for a terrain with generalized polygonal obstacles as will be shown later.

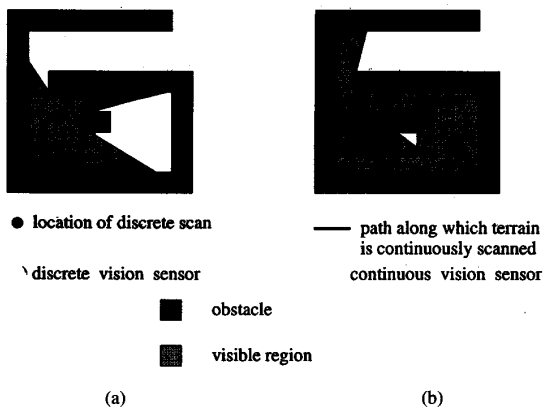


Fig. 1. Discrete and continuous vision sensors. In (a) the region visible in a single scan operation performed from location P is shown. In (b) the visible region corresponds to a continuous scan along the shown path.

We now describe three geometric structures that can be employed for the navigation in unknown terrains.

1) Generalized Visibility Graph (GVG): Lozano-Perez and Wesley [17] proposed one of the earliest methods for the path planning of a polygonal robot translating through a known polygonal terrain using the visibility graph. Laumond [15] proposed an algorithm for a polygonal robot through a known terrain composed of generalized polygons based on an extension of the visibility graph, called the generalized visibility graph. The generalized visibility graph is defined as follows [15]: i) An obstacle vertex p and an arc are visible to each other if there exists a point q on the arc such that p and q are mutually visible and the segment $\overline{pq}$ is tangent to the arc; point q of the arc constitutes a fictitious vertex; ii) Two arcs are visible if they have at least one common tangent segment $\overline{pq}$ such that p and q are visible; points p and q are fictitious vertices. The vertices of the GVG are then the vertices of the generalized polygons (actual vertices) and the fictitious vertices resulting from the existence of tangent segments to the arc. The edges of the GVG are of two types: straight-line segments and arcs. Two vertices are connected by a straight line edge if and only if they are visible from each other. Two vertices are connected by an arc if both

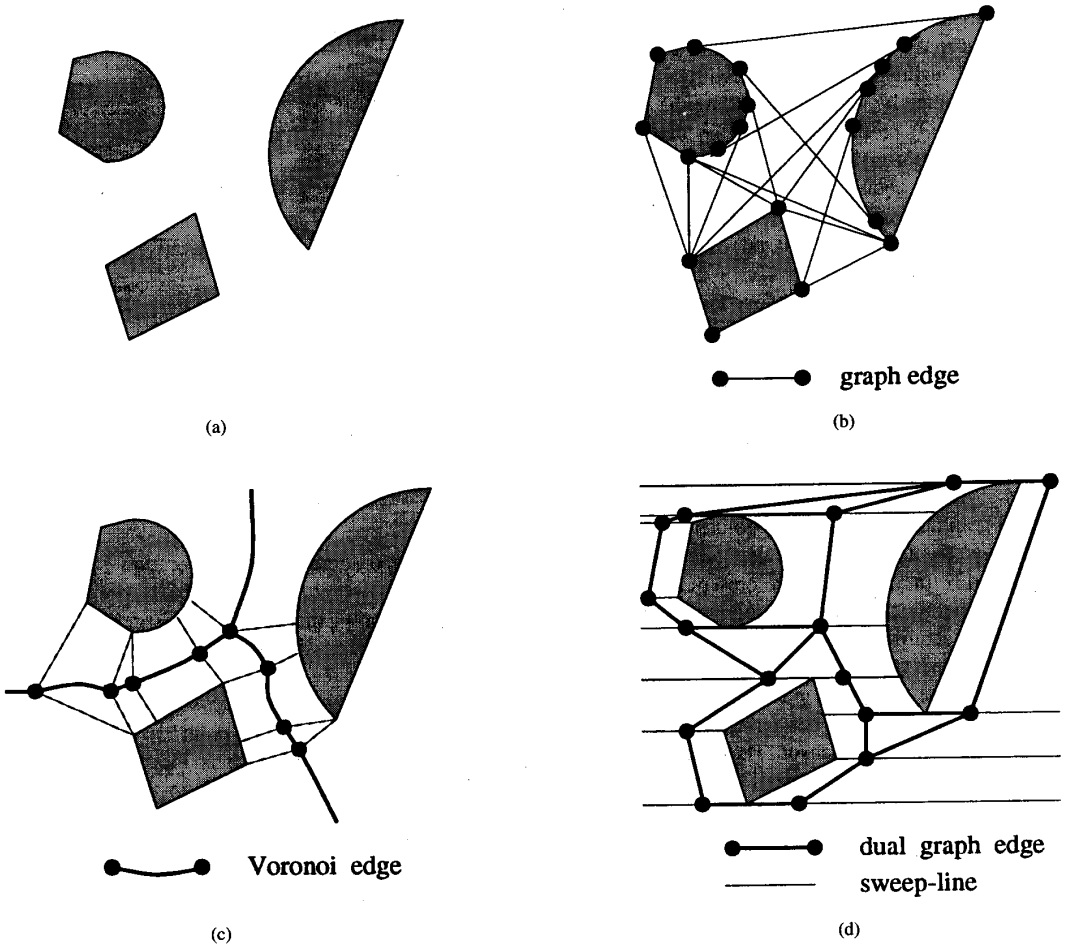


Fig. 2. Geometrical structures. For navigation in terrain in (a), any of the navigation courses in (b)-(d) can be utilized.

are located on an arc boundary of the same obstacle and we can traverse along the boundary from one vertex to the other without encountering any other vertices during the traversal. Fig. 2(b) shows an example of a GVG. A more general version of the GVG has been employed by Hershberger and Guibas [10] to plan paths for a convex translating body in polygonal terrains.

2) Voronoi Diagram: The Voronoi diagram corresponding to a set of line segments and circular arc segments has been studied by Yap [36]. The distance $d(p, s)$ between a point p in free-space and a boundary edge s is defined as $\inf\{d(p, q) \mid q \in s\}$. The clearance of a point p in free-space with respect to O is the minimum of $d(p, s)$ for some obstacle edge (segment or an arc) s of O. For $x \in \Omega$, we define $Near(x)$ as the set of points that belong to the boundaries of obstacles O_i, $i = 1, 2, \cdots, n$ and are closest (among all points on the obstacle boundaries) to x in terms of the metric d. The Voronoi diagram, $Vor(O)$, of the terrain populated by O is the set $\{x \in \Omega \mid Near(x)$ is a disconnected set$\}$, (i.e., for each $x \in Vor(O)$ the set $Near(x)$ contains more than one topologically connected components or equivalently $x \in Vor(O)$ is nearest two at least two distinct points on the obstacle boundary).[1] This definition implies that for each $x \in Vor(O)$ there are at least two distinct points on the obstacle boundary that are closest to x in the metric d. See Fig. 2(c) for an example. In this case, $Vor(O)$ is a union of $O(N)$ straight lines and algebraic arcs of degree at most four [36].

3) Dual Graphs Based on Trapezoidal Decomposition: First, we decompose the free-space into trapezoids by sweeping a line (for example, moving a horizontal line from top to bottom) such that whenever the line passes through a vertex, extend a sweep-line segment from this vertex into free-space until it touches an obstacle boundary or extends to infinity as shown in Fig. 2(d). Now free-space is partitioned into trapezoids. There are a number of ways of defining dual graphs based on the decomposition, and we consider a version that is suited for the present problem. For each sweep-line

[1] If terrain O consists of polygons only, then $Near(x)$ is a finite set of points, and the Voronoi diagram is then defined as set of points x in free-space such that $Near(x)$ contains at least two points. If O contains generalized polygons, $Near(x)$ can contain circular arcs. The current definition is more appropriate in such cases. See Yap [36] for more details on this aspect.

segment we have one of the two following cases: (a) if the segment is finite, the dual graph node corresponds to the mid point of the segment, or (b) if the segment is not finite, the dual graph node corresponds to a point on the segment at a distance δ from the vertex. Two nodes belonging to the boundary of the same trapizoid are connected by an edge of the dual graph. See Fig. 2(d) for an example of a dual graph based on the trapezoidal decomposition.

An algorithmic framework for solving the navigation and terrain model acquisition problems using discrete vision sensors has been proposed by Rao [27]. Here R uses a one-skeleton (recall that a one-skeleton is a collection of one-dimensional curves) $\xi(O)$, called the navigation course, embedded in $\overline{\Omega}$; $\xi(O)$ can be simply viewed as a combinatorial graph such that each ξ-vertex specifies a position for R and each ξ-edge (u, v) specifies a path from u to v. To navigate from s to d, R performs a "graph search" on $\xi(O)$. Initially $\xi(O)$ is not known to R, but it is incrementally constructed from the sensor operations. From s, R initially checks to see if d is reachable, and moves to it if it is. If not, R computes a start ξ-vertex v_o and moves to it, and from v_o, R keeps visiting new ξ-vertices until it reaches a ξ-vertex v_f from which d is found reachable. In the case d is not reachable from s, R visits all ξ-vertices and concludes that the destination is not reachable. In solving the terrain model acquisition problem, R systematically visits all ξ-vertices.

The requirement on the graph search algorithm is that it must be capable of visiting all vertices of a connected component of the $\xi(O)$ in which it is initiated. Examples for graph search algorithms that can be used here include the popular depth-first search [1] and A^* algorithm [26].

We now consider four properties for $\xi(O)$:

1) finiteness requires that the number of ξ-vertices and edges be finite;
2) terrain-visibility requires that every point in $\overline{\Omega}$ be visible from some ξ-vertex;
3) connectivity requires that every pair of ξ-vertices be connected by a graph path on $\xi(O)$;
4) local-constructibility requires that the adjacency list of an ξ-vertex be computable from the information obtained by performing a finite number of sensor operations.

Then the following result can be easily shown [27].

Result 1: Given a navigation course $\xi(O)$ for a terrain O, that satisfies the properties of finiteness, connectivity, terrain-visibility and local-constructibility, a graph search algorithm can be employed to solve the navigation and the terrain model acquisition problems using a discrete vision sensor.

For two-dimensional terrains with polygonal obstacles, modified versions of the visibility graph [28] and Voronoi diagram [31] can be used as $\xi(O)$. we show in Section IV that a dual graph based on the trapezoidal decomposition can also be used as $\xi(O)$; in general, dual graphs based on other types of decompositions such as triangulations can also be used for $\xi(O)$. In Section IV, we show how modified versions of these three basic structures can be used in terrains with generalized polygonal obstacles.

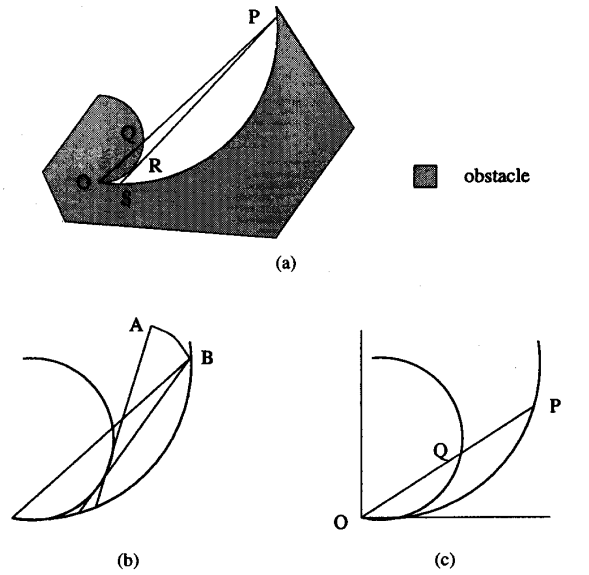


Fig. 3. Proof of Theorem 1. An example of cusp region is shown in (a). When viewed from P along $\overline{PS}$ the region to the right of $\overline{RS}$ cannot be seen from any scan operation performed to the right of $\overline{PR}$ (b) region not visible from A; (c) computation of Q.

III. Number of Scan Operations

We now show that an infinite number of scan operations are needed to ensure a complete solution to the navigation problem in generalized polygonal terrains[2] irrespective of the algorithm and/or solution framework employed. To solve the navigation problem or the terrain model acquisition problem, any algorithm must ensure that all points in the free-space are scanned in a worst-case. Note that at no intermediate point in the navigation, can an algorithm positively conclude that the destination is not reachable unless all the connected portions of the free-space containing the start position has been scanned.

Theorem 1: In a generalized polygonal terrain, scan operations performed from a finite set of sensing points Θ may not suffice to guarantee that every point in free-space is visible from some point in Θ.

Proof: We show that in a particular part of a terrain, a finite set of sensing points does not suffice to ensure that every point in free-space has been scanned. Consider a part of the terrain consisting of a corner formed by a convex and concave arcs as shown in Fig. 3(a). Let the vertex at which these two arcs meet be the origin O and recall that the free-space around this vertex is called the cusp region. We will show that for every point $P = (x_p, y_p)$ on the boundary of outer circle (concave arc), there exists $Q = (x_q, y_q)$ which is the intersection of the line joining the origin O to P. The existence of Q implies that there are going to be points inside the cusp region that are not visible in any scan operation performed from points to the right of the segment $\overline{PQ}$ when viewed from Q. This argument can be formalized as follows. The line segment $\overline{OP}$ can be rotated around P such that

[2]Similar result can be shown when the obstacles are fractals [5] using essentially the same approach as the proof of Theorem 1.

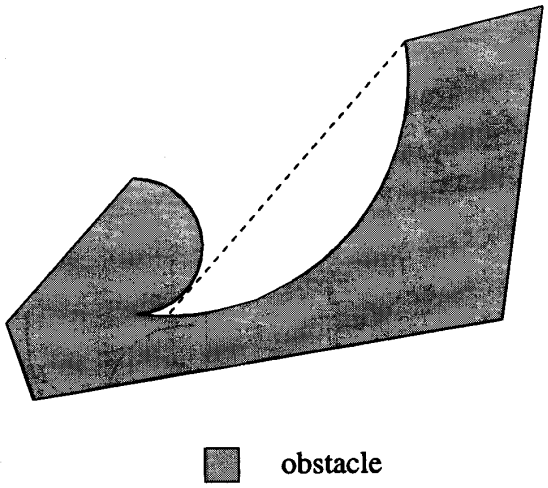


Fig. 4. A generalized visibility graph that does not satisfy the terrain-visibility property.

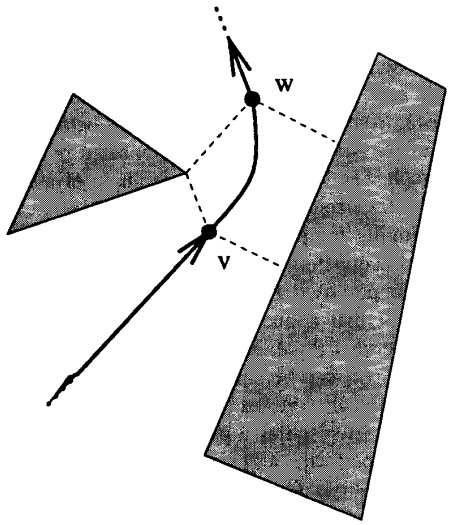


Fig. 5. A generalized Voronoi diagram that does not satisfy the local-constructibility.

$\overline{PR}$ is the tangent from P to the convex (inner) arc at R as shown in Fig. 3(a). This tangent $\overline{PR}$ meets the concave arc at S (Fig. 3(a)). Then the points to the right of the segment $\overline{RS}$, when viewed along the direction of $\overrightarrow{RS}$, are not visible from P.

Given the above result, the theorem can be shown by contradiction. Consider that a finite set of sensing points Θ exists in contradiction to the statement of the theorem. Then choose a point A closest to the origin O. Let the distance between O and A be r. Then consider a circle C with radius r centered at O and circle D obtained by extending the outer concave arc of the corner. Let B be the intersection of C with the outer circle D obtained by moving along C in the clockwise direction from A. Note that in the cusp region every point visible to A is visible to B and every point not visible to B is not visible to A (Fig. 3(b)).

Now we shall show that there exists Q for every point P such that $(x_q, y_q) = (tx_p, ty_p)$ for some t, $0 < t < 1$. For ease of presentation consider that the arcs are located as in Fig. 3(c). Let the inner arc correspond to circle of radius r_1 centered at $(0, r_1)$. Let the outer arc correspond to the circle D of radius $r_2 (> r_1)$ centered at $(0, r_2)$. Since P is on outer circle, we have

$$x_p^2 + y_p^2 = 2r_2 y_p. \tag{1}$$

The equation of the line through O and P is given by $x = \frac{x_p}{y_p} y$. By simultaneously solving this equation with that of the inner circle C we obtain $y_q = \frac{2r_1}{x_p^2 + y_p^2} y_p^2$. Using $y_q = ty_p$, we obtain

$$t = \frac{2r_1}{x_p^2 + y_p^2} y_p.$$

By using (1), we obtain $t = r_1/r_2$. Hence we have $0 < t < 1$. □

The above theorem shows that it is not always possible to solve the terrain model acquisition and navigation problems by using only a finite number of scan operations. But note that the difficulty is caused by the cusp regions (as described in the proof of Theorem 1) formed by a pair of convex and concave arcs. We will show that if we exclude the narrow parts of the cusp regions, the two problems can be solved using only a finite number of scan operations. In the cusp regions, we ignore the portions that correspond to narrow and curved corridors by suitably defining the precision parameter ϵ. We ensure that the only parts that are not scanned are those that are at most within a distance of ϵ from two obstacle boundaries, in a region formed by a pair of convex and concave obstacle edges. Note that it is not necessary that the convex and concave arcs of a cusp region meet at a vertex; at the narrow end of the corridor formed by the arcs, there could be other obstacle edges, and they could be left undetected if they are contained in the fictitious obstacles defined due to precision ϵ.

IV. Navigational Courses

We now present three types of graph structures that satisfy the four properties of the navigation course presented in Section II. These graph structures are based on visibility graph, Voronoi diagram and trapezoidal decompositions. The structures described in Section II do not satisfy all the required properties. It can be seen that a generalized visibility graph does not always satisfy the terrain-visibility property as shown in Fig. 4, because the part of the region formed by the two circular arcs is not visible from any of the vertices of the generalized visibility graph. In general, the Voronoi diagram may not satisfy the local-constructibility property as in Fig. 5 since the robot located at a Voronoi vertex v does not know where the other vertex w lies; this is because the edge on the "hind" side of the obstacle vertex determines the location of w, and in general this edge might not have been sensed

(a) (b)

Fig. 6. (a)–(b). Definition of boundary chains.

yet. We augment these structures so that each satisfies all four properties of Section II. We also discuss a navigation course based on the trapezoidal decomposition.

A. Augmented GVG

The Augmented Generalized Visibility Graph (AGVG) is a graph (V, E) defined as follows:

1) V is the union of the following sets of vertices:
   a) the set of actual vertices, which are obstacle vertices;
   b) the set of fictitious vertices resulting from the existence of tangent segments to the arcs (as described in Section II);
   c) the set of boundary vertices, which are obtained as follows. Consider the convex hull of O whose boundary consists of portions of circular arcs, and straight line obstacle edges. Now for each circular arc on the boundary of convex hull of O, we define two chains, clockwise and anticlockwise, which are obtained by navigating from one end of the circular arc to the other end in clockwise and anticlockwise, anticlockwise directions, respectively. Each chain is obtained by the repeated application of two steps (except for the last iteration): a) moving a distance of δ in the direction of tangent to a point X; b) computing a new tangent to the arc from X and moving along the tangent to touch the arc, (as shown in Fig. 6). For the last iteration, we may travel a distance less than δ. We can use either of the chains in the AGVG. The boundary vertices are the vertices that belong to one of the chains including the end vertices;
   d) the set of internal vertices which are obtained as follows. Consider the regions of $\overline{\Omega}$ that are not visible from any of the vertices belonging the sets defined in a) and b). These regions can be obtained by imagining a "sweep" operation from each of the vertices in a) and b). A sweep operation from a vertex consists of starting a line segment extended into free-space from the vertex until it meets an obstacle boundary; then the line segment is rotated clockwise direction until it becomes a tangent to the obstacle boundary at the vertex, and repeating the same in the anti-clockwise direction. The tip positions of this line segment as it is rotated together with the end tangents from the vertex form generalized polygon. The regions of the free-space that lie outside the union of generalized polygons obtained from the vertices of a) and b) are of interest now. The boundary of each such region must consist of obstacle boundaries which are portions of circular arcs. Further, it is not possible to form such region with two convex or two concave vertices as shown in Fig. 7(a) and (b), because the free-space contained in between the obstacle edges is seen by performing scan operations from the nodes indicated. The only possibility is as in Fig. 7(c). In this case, we define a suitable internal chain corresponding to the convex arc similar to the boundary chain, except that δ is a variable chosen such that the chain lies in the free-space. We curtail the internal chains when the "width" of the cusp region is smaller than 2ϵ as in Fig. 7(d).

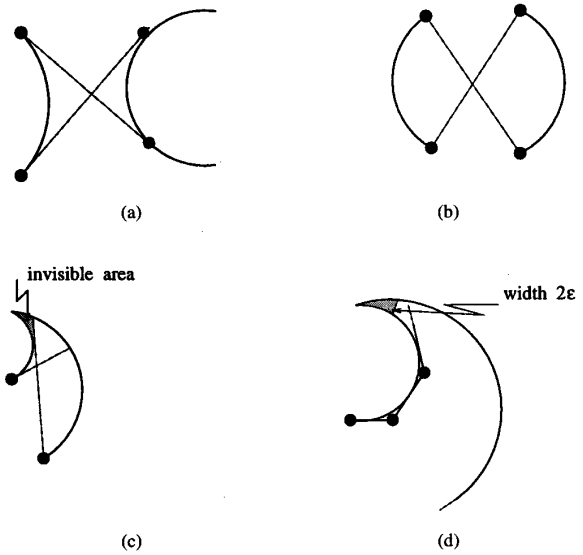


Fig. 7. Regions formed by "sweeping" operation. In the case of a pair of convex (or concave) arcs shown in (a) (or in (b)) the region in between the arcs can be seen from the end points of mutual tangents. In a cusp region (as in (c)), the interior points are not visible to the points at which tangents are drawn. A sequence of scan points are needed to explore the cusp region until a prescribed clearance is observed as shown in (d).

2) A line or a convex curve joining the vertices v_i and v_j is an edge $(v_i, v_j) \in E$ if and only if it is one of the following:
   a) a part of an obstacle edge (line segment or arc) and does not contain any other vertices (except at the end points),
   b) a tangent edge between two arcs or between an arc and an obstacle vertex, which is not intersected by any obstacle,
   c) edges corresponding to a polygonal chain (either clockwise or anti-clockwise) of arcs on the convex hull of O,
   d) edges corresponding to the internal chains discussed in 1–d).

We now establish that AGVG satisfies the required four properties.

1) Finiteness: Assume that the number of convex arcs is m. By the definition of GVG, the number of actual vertices is the number of obstacle vertices, N. We now compute the number of fictitious vertices, f_v, boundary vertices, b_v, and internal vertices i_v. To compute f_v we consider two cases concerning an arc.

Case 1: An Arc and an Actual Vertex: Since for each actual vertex there are at most two tangents to an arc, the number of fictitious vertices for an arc to an actual vertex is at most 2. So for this case, the number of fictitious vertices is less than or equal to $2mN$.

Case 2: Two Arcs: For every pair of arcs, there are at most 4 common tangents. Since each tangent edge contributes at most 2 fictitious vertices, there are at most 8 fictitious vertices for every pair of arcs. Thus these m arcs generate at most $8\binom{m}{2}$ fictitious vertices for this case.

Now to compute b_v, consider an arc that lies on the boundary of convex hull of O. Let $\tan\theta = \frac{\delta}{r_i}$, where r_i is the radius of the arc. Now for the arc in a worst-case, the number of boundary vertices corresponding to the arc is $\lceil \frac{2\pi}{\tan^{-1}\left(\frac{\delta}{r}\right)} \rceil$. For m arcs in a worst-case b_v is $m\lceil \frac{2\pi}{\tan^{-1}\left(\frac{\delta}{r}\right)} \rceil$. Similarly we can show that i_v is finite. Thus the total number of vertices of the AGVG is finite.

2) Connectivity: We claim that a shortest path between any two points in free space consists of i) straight-line segments, which are either tangent edges between obstacles or straight-line boundaries of obstacles; ii) convex arcs, which are parts of obstacle boundaries. Moreover, a shortest path runs through actual vertices, fictitious vertices and boundary vertices, in between two consecutive vertices, the subpath belongs to one of the categories of i) or ii).

We first prove the above claim. The following proof is based on Hershberger and Guibas [10]. Any shortest path is composed of subpaths that alternately follow obstacle boundaries and move along tangent edges between obstacles. The subpaths following obstacle boundaries are made up of obstacle edges. Thus the above claim is reduced to that of showing every nonboundary segment on the shortest path is a tangent edge of the AGVG. Suppose that $\overline{v_i v_j}$ lies on a shortest path from s to g and is not an edge of the AGVG. At least one of the end points of $\overline{v_i v_j}$, say v_j, lies on an obstacle boundary and v_j is not a tangent point of $\overline{v_i v_j}$. And since v_j is not a tangent point, we can find a point v_k which is beyond v_j and is visible from a point v_l on $\overline{v_i v_j}$. This means that the supposed shortest path could be shortened by replacing the subpath from v_l to v_k via v_j by $\overline{v_l v_k}$, which contradicts our assumption that $\overline{v_i v_j}$ lies on a shortest path. Now a shortest path between any two vertices runs through the edges of the AGVG, and hence it is connected.

3) Local-Constructibility: To show the property of local-constructibility, consider two cases.

1) A vertex of the AGVG lies inside convex hull of O (minimal convex region that contains all obstacles of O), denoted by $CH(O)$, but lies outside the cusp regions. The neighbors of an actual or a fictitious vertex inside $CH(O)$ can be obtained from the sensor operation. For the special case that the edges next to the vertex v are circular arcs, we use a suitable chain to navigate along those circular arcs.
2) A vertex v of the AGVG that corresponds to a convex arc of the boundary of $CH(O)$ or a cusp region. The adjacency list of v can be computed by suitably adding boundary vertices.

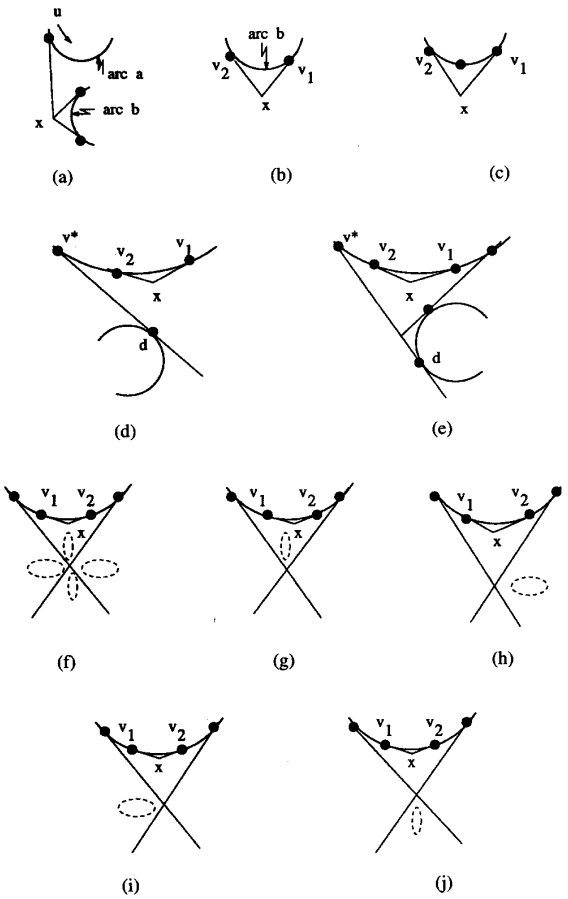


Fig. 8. (a)–(j) Terrain-visibility property of augmented GVG. The dotted circles represent the presence of one or more obstacle polygons.

Hence the AGVG satisfies the local-constructibility property.

4) Terrain-Visibility: Finally we show the property of terrain-visibility. A point x outside $CH(O)$ will be visible from either a boundary vertex or an obstacle vertex that lies on the boundary of $CH(O)$. Also every point in a cusp region can be seen within precision ϵ due to the corresponding chains. Now consider a point x located in the free-space inside $CH(O)$ and not in a cusp region. Pick a point y outside $CH(O)$. Now consider a shortest path P from x to y. If P does not touch the boundary of any obstacle, then rotate the point y around the center x with a radius of the distance between x to y. Such rotation must result in some path P that touches an obstacle boundary; otherwise there is no obstacle in the terrain. Now move from x towards the first point u of P that touches the boundary of the obstacle, say O_i. Here u may be an actual vertex or a point on an arc.

1) If u is an actual vertex, x is visible from actual vertex u.
2) Now consider that u is a point on an arc. Move u along the boundary of obstacles such that line segment $\overline{xu}$ lies in the free-space. If we visit an obstacle vertex in this process then we are done. Otherwise we have the case shown in Fig. 8(a): there exists an arc b such that both

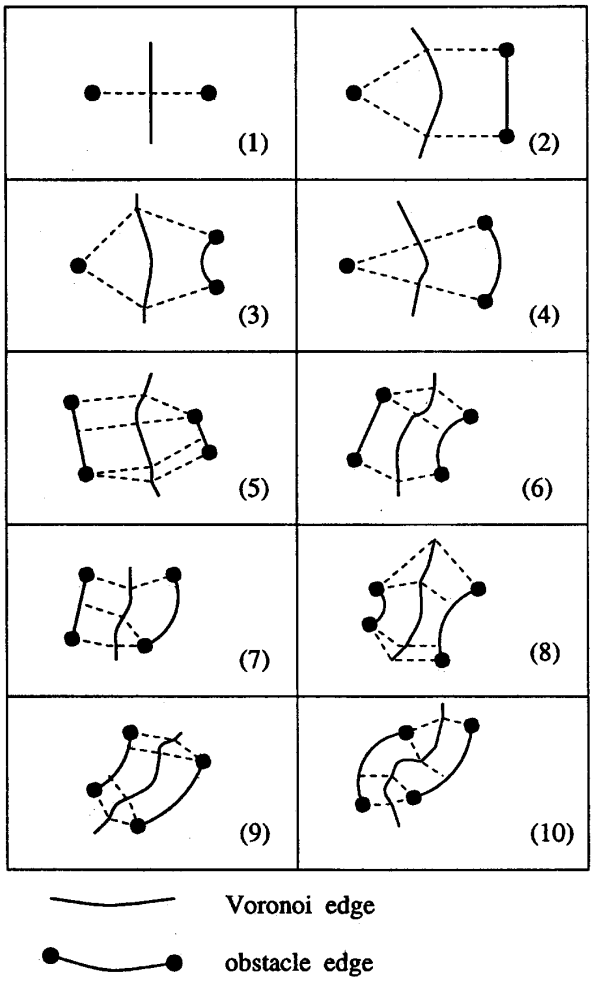


Fig. 9. Basic cases of generalized Voronoi diagram.

the tangents from x to this arc do not intersect any obstacle and no obstacles are encountered by the line segment from x to the boundary of the obstacle as we move the end point of the segment along the boundary of the obstacle from one tangent point to the other. To see that such arc b exists, start rotating the ray from x through u in the clockwise direction going through the obstacles. For concreteness assume that the obstacle containing u is to the right of the line from x through u. We will not be able to draw a second tangent to this arc a (containing u) only if the ray becomes tangent to another obstacle before reaching the second tangent to arc a. But as we continue the rotation, this event (of not being able to draw two tangents to a single arc) cannot indefinitely be continued because we have only finite number of obstacle. Let b be the arc that supports two tangents from x at points v_1 and v_2 (see Fig. 8(b)); note here that b could be same as a. Now we have two cases:

a) There is a fictitious vertex on the portion of b between v_1 and v_2, denoted by $[v_1, v_2]$ (Fig. 8(c)); in this case we are done because x can be seen from this fictitious vertex.

b) There are no fictitious vertices in between v_1 and v_2. It is easily seen that there must be fictitious vertices on at least one side of $[v_1, v_2]$; otherwise the portion $[v_1, v_2]$ will be on the convex hull boundary and hence x will be outside the convex hull (which is a contradiction). Now we have the following two cases:

- There fictitious vertices only to one side of $[v_1, v_2]$: without loss of generality assume that the side having fictitious vertices is obtained by going clockwise from v_1 to v_2. Let v^* be the first fictitious vertex while moving clockwise from v_2. Let arc d be the other arc of the tangent from v^*. The arc d can be to the left (Fig. 8(e)) or to the right (Fig. 8(d)) of the tangent as we move away from v^* towards d. Now for Fig. 8(d), x must be visible from the point at which the tangent touches d. The case of Fig. 8(e) will not occur because if we sweep the tangent from v_1 through x such that the starting point of the tangent moves in anticlockwise direction, we will generate a tangent such that there will be a fictitious vertex in the anticlockwise direction from v_1.
- There are fictitious vertices one both sides of $[v_1, v_2]$. Now sweep (move along the boundary of the arc containing v_1 and v_2) two tangents as follows. Start with tangent from v_1 through x and sweep it in anticlockwise direction until the first tangent is obtained. Similarly sweep in the opposite direction from v_2. Now there are four regions formed out of these two tangents as shown in Fig. 8(f). Out of these four cases, the case shown in Fig. 8(g) is not possible, because otherwise we would not be able to draw to tangents to the arc b during the rotation process. In the other three cases, x will be seen from one of the tangent points as shown in Fig. 8(h) through (j).

In summary we have shown the following theorem.

Theorem 2: The augmented generalized visibility graph satisfies the properties of finiteness, connectivity, local-constructibility and terrain-visibility with precision ϵ, i.e., the points that are not visible from any vertex are within a distance of ϵ from two boundary edges in a cusp region formed by a pair of convex and concave obstacle edges. □

B. Augmented Voronoi Diagram

We consider the general Voronoi diagram as a 1-skeleton embedded in the free-space $\overline{\Omega}$ (refer to the definition in Section II). The Voronoi diagram is a collection of 1-dimensional curves called the Voronoi edges (or V-edges) that end at the Voronoi vertices (or V-vertices); each Voronoi edge is formed by the interaction of two basic objects of the obstacle boundaries. The basic objects are points, line segments, convex circular arcs and concave circular arcs. Thus the Voronoi edges are formed by the interaction of pairs of these objects as summarized in Fig. 9. We have the following cases:

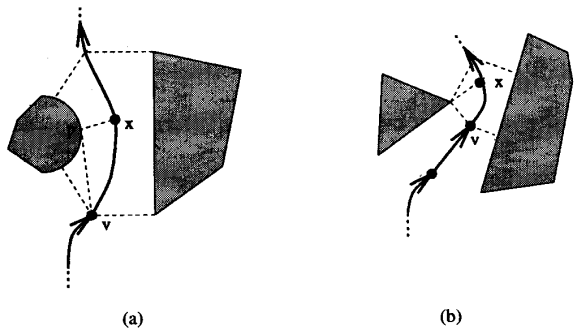


Fig. 10. Local-constructibility of the augmented Voronoi diagram. (a) tangent from a vertex v to the arc at p; (b) end vertices of the edges.

1) (point, point): Let the two points be p_1 and p_2. In this case the Voronoi edge is a straight-line which is the perpendicular bisector of the line segment $\overline{p_1p_2}$.

2) (point, line segment): Let the point be p and let the line segment be $\overline{q_1q_2}$. The Voronoi diagram consists of three segments: two line segments resulting from the (point, point) interaction of the pairs (p, q_1) and (p, q_2), and a parabolic arc with p as focus and $\overline{q_1q_2}$ as the directrix.

3) (point, convex arc): The Voronoi diagram consists of a line segment, followed by a fourth-degree curve, followed by a line segment.

4) (point, concave arc): The Voronoi diagram consists of a line segment, followed by a fourth-degree curve, followed by a line segment.

5) (line segment, line segment): The Voronoi diagram is a sequence consisting of a line segment, a parabolic arc, a line segment, a parabolic and a line segment.

6) (line segment, convex arc): The Voronoi diagram is a sequence consisting of a line segment, followed by at most three parabolic arcs, followed by a line segment.

The overall profile in each of the following four cases is the same as in (6), although the exact nature of the parabolic arc depends on the individual case:

7) (line segment, concave arc),
8) (convex arc, convex arc),
9) (convex arc, concave arc), and
10) (concave arc, concave arc).

Consider the convex hull $CH(O)$ of the union of all obstacles. Let $E(O)$ denote the region obtained by pushing the edges of $CH(O)$ outwards by a distance of s. We define $Vor_1(O) = (Vor(O) \cap E(O)) \cup \partial E(O)$, where $\partial E(O)$ is the boundary of $E(O)$, i.e., $Vor_1(O)$ consists of the Voronoi diagram contained inside $E(O)$ and the boundary of $\partial E(O)$. The set of vertices of $Vor_1(O)$ is the union of Voronoi vertices, vertices of the envelop $E(O)$ and intersection points of edges of $\partial E(O)$ with Voronoi edges.

Let us obtain the cellular decomposition of the closure of $\Omega \cap E(O)$ as follows. From each of V-vertex v, draw extension lines joining v to all its nearest obstacle edges and vertices. Furthermore, join each vertex of $E(O)$ to its corresponding obstacle vertex. These lines are also called extension lines. The extension lines, Voronoi edges, and obstacle edges partition the closure of $\Omega \cap E(O)$ into cells. Each cell has an edge of $Vor_1(O)$ and two extension lines and at most one obstacle edge. For each cell consisting of a convex arc, we define two chains, clockwise and anti-clockwise. The clockwise chain is obtained as follows: Consider a tangent from a vertex v to the arc at p (Fig. 10(a)). Let x be a point on $Vor_1(O)$ such that $p \in Near(x)$. If there is more than one candidate for x, the one closest to v is chosen. This process is repeated from x. x is called an internal vertex.

The $Vor_1(O)$ is augmented with

1) internal vertices of either a clockwise or anti-clockwise chain, and
2) boundary chains corresponding to circular arcs of $E(O)$ obtained as in the case of AGVG (Fig. 6).

The resultant structure is called the augmented generalized Voronoi diagram (AGVD).

1) Finiteness: First consider the part of the AGVD without the internal vertices. It can be easily seen that the number of vertices is $O(N)$. Consider an obstacle O_i and the subset of $Vor_1(O)$ such that each point on this subset has a nearest neighbor on the boundary of O_i. This subset consists of one cycle and a finite set of trees. We now define the dual $D(O)$ of $Vor(O)$ as follows: Draw perpendiculars to each obstacle edge at the convex end-points (obstacle vertices) and extend them outwards. Now Ω is partitioned into regions such that the points belonging to each region are closer to either an obstacle edge or an obstacle vertex than any other obstacle edge or vertex. We represent each region by a D-node. Two D-nodes are connected by a D-edge if and only if the corresponding regions meet at either a V-edge or a perpendicular. Using the dual, we can establish the following bounds using a derivation similar to that in [31].

1) $(n+5)/2 \leq \#V_1$-vertices $\leq 4N - n - 2$,
2) $3(n+1)/2 \leq \#V_1$-edges $\leq 6N - 3n - 3$.

Now considering that each convex arc introduces only a finite number of internal vertices (as shown in Section IV.A, we have shown the finiteness property of the AGVD.

2) Connectivity: A map $Im: \Omega \mapsto Vor(O)$ is defined by O'Dunlaing and Yap [11] as follows. Consider $x \in \Omega$. If x is on $Vor(O)$ then $Im(x) = x$; otherwise, $Near(x) = \{p\}$ for some point p on $\partial\Omega$, the boundary of Ω. Let L be the semi-finite straight line from p through x, and define $Im(x)$ to be the first point y (if it exists), where L intersects $Vor(\Omega)$. Intuitively, $Im(x)$ is obtained by "pushing" x away from the closest wall (or corner) until it lies on the Voronoi diagram. We state a Theorem from O'Dunlaing and Yap [11].

Fact 1: If Ω is bounded, then i) the map Im is a continuous retraction of Ω onto $Vor(O)$ (so $Vor(O)$ is a retract of Ω), and ii) if $Im(x) \neq x$, then the clearance is strictly increasing along the line-segment joining x to $Im(x)$. □

The obstacle-free region Ω is homeomorphic to the (real) plane with n closed discs removed from it, each disc corresponding to a single obstacle. Hence, Ω is (polygonally) path connected. Im is shown to be a continuous retraction of a bounded Ω onto $Vor(O)$ (Fact 1). Thus $Vor(O) \cap E(O)$ is a continuous image of a connected set Ω (bounded appropriately), and hence is connected. Now $E(O)$ is topologically

connected. $(Vor(O) \cap E(O)) \cap \partial E(O) \neq \phi$, and hence $Vor_1(O) = (Vor(O) \cap E(O)) \cup \partial E(O)$ is connected by the Clover-leaf Theorem. Then it is direct to see that within precision ϵ the AGVD preserves the connectivity of $Vor_1(O)$.

3) Local-Constructibility: If R is located at a V_1-vertex v, the sensor gives the clockwise listing of the obstacle vertices and intervals of obstacle edges that are visible from v. We compute the vertices and edge segments that are closest to v. There will be an edge of the AGVD emanating from v corresponding to each consecutive pair of objects found above. We can compute the equation for each of these edges, and we may not be able to compute the end vertices of these edges as in Fig. 10(b). This happens when one of the edges incident on a convex vertex is not detected by the sensor. In this case, we extend the known obstacle edge to intersect the Voronoi edge, at a sensing vertex. R first moves to the sensing vertex and then performs a scan operation and obtains the hidden obstacle edge. By scanning from at most two sensing vertices, the other end of the Voronoi edge can be computed. The sensing vertices correspond to the internal vertices of the AGVD. After including the sensing vertices, the AGVD satisfies the local-constructibility property, since every sensing vertex is an internal vertex.

4) Terrain-Visibility: Consider the cellular decomposition of the closure of $\Omega \cap E(O)$. Every point outside of $CH(O)$ is visible from some vertex of $E(O)$. Consider $x \in \Omega \cap CH(O)$. First, if $Im(x) \in CH(O)$ then move along the corresponding Voronoi edge e in some direction until either a Voronoi vertex is encountered or we move out of $CH(O)$. If the former occurs then x lies in the cell associated with e and hence is visible from its end vertices. If the latter occurs then reverse the direction of motion along e and traverse in the other direction until a V-vertex v is encountered. Second, if $Im(x)$ does not belong to $CH(O)$ then move along the corresponding V edge towards $CH(O)$ until $\partial E(O)$ is encountered at the intersection point y. As we traverse along $Vor(O)$ always choose the V-edge that is closest to x at V-vertices (if V-vertices are encountered). It is clear that the line joining x to y will be free of obstacles and hence x is visible from y.

In summary we have the following theorem.

Theorem 3: The augmented Voronoi diagram for a generalized polygonal terrain satisfies the properties of finiteness, connectivity, local-constructibility and terrain-visibility within precision ϵ, i.e., the points that are not visible from any vertex are within a distance of ϵ from two boundary edges in a cusp region formed by a pair of convex and concave obstacle edges. □

C. Trapezoidal Decomposition

We first consider the special case of polygonal terrains to illustrate the properties of finiteness, connectivity, local-constructibility, and terrain-visibility since the explicit proofs of these properties are not readily available, and the proofs for the general case easily follow from these proofs.

1) Polygonal Terrains: We first obtain a trapezoidal decomposition by sweeping a line L, of arbitrarily chosen orientation, in the direction normal to L, and forming a trapezoid when a vertex is encountered. Each trapezoid is bounded by (continuous segments of) two obstacle edges and at most two (but at least one) sweep-line segments corresponding to L; each such line segment contains at least one obstacle vertex. Let the dual graph based on the trapezoidal decomposition be denoted by $D_T(O) = (V_T, E_T)$.[3] Recall that for each sweep-line segment of a trapezoid we associate $v \in V_T$ which is a point on the segment such that a) if the segment is not finite, then v is at a distance δ from the vertex of the segment, b) if the segment is finite, then v is the mid point of the segment. Two vertices $v_1, v_2 \in V_T$ are connected by an edge if and only if they belong to the boundary of the same trapezoid. For this case of polygonal obstacles each trapezoid is convex, and the line segment joining two vertices of a trapezoid does not intersect the interior of any obstacle (note that the sweep-line segment is allowed to intersect the boundary of an obstacle).

a) Finiteness: An upperbound to the number of nodes of $D_T(O)$ is estimated in the following Lemma.

Lemma 1: Consider a terrain such that no two obstacles are co-linear with respect to the sweep-line. The number of vertices in the dual graph $D_T(O)$, based on the trapezoidal decomposition, is upperbounded by $N + 2n$ for a terrain of n polygonal obstacles consisting of a total of N vertices.

Proof: First consider the bound on the vertices of D_T. The obstacle vertices can be classified into two main categories, those which form an inflection point (a point that supports a local tangent) in the chosen sweep-line direction and those that do not; vertices of the first kind are called inflection vertices and those of the second kind are called noninflection vertices (see Fig. 11). Without loss of generality we assume that the sweep-line is horizontal. An inflection vertex v is pointing up (pointing down) depending on if the obstacle in the vicinity of v is below (above) the sweep-line through v. From each noninflection vertex there will be a segment of the sweep-line (through it) that contains precisely one node of D_T. In general each inflection vertex generates two vertices of D_T, and thus one can obtain a crude upper bound of $2N$ on the number of nodes of D_T. We now tighten this bound to $N + 2n$ nodes.[4]

Consider the terrain with convex polygonal obstacles such that no two vetices have the same X-coordinate. Here each obstacle contains precisely two inflection vertices, each of which generates two nodes of D_T, and every noninflection vertex generates a single D_T node. Thus the total number of nodes of D_T is exactly $N + 2n$ for this case. We now show that $N+2n$ is an upper bound for the case containing possibly nonconvex polygonal obstacles. As each obstacle is swept, we have two extreme inflection points corresponding to the first and the last times the obstacle is encountered; each of these inflection points account for two nodes of D_T.

We show that for each nonextreme inflection vertex v there corresponds a concave vertex w that does not generate a node of D_T; w accounts for one of the nodes on the segment through v. We show this result for pointing up nonextreme inflection

[3] We often omitt the operand O, since O is fixed for a given problem instance.

[4] If each polygon has three vertices, we have the largest value of $n = N/3$ which yields $N + 2n = 5/3N$ which is tighter than $2N$.

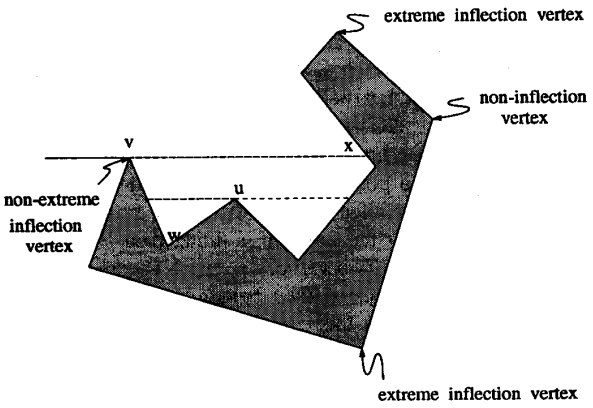


Fig. 11. Examples of extreme inflection vertex, non-extreme inflection vertex and non-inflection vertex.

points and that for pointing down nonextreme inflection points is similar. To make the discussion concrete, we assume that of the two graph nodes generated by v, the one to the left of v is accounted for by v and the one to the right is accounted by the yet to be described w. Visualize that we move the sweep-line from top to bottom. As we sweep, extend the sweep from nonextreme inflection point v to the right until it meets the obstacle that v belongs to at point x (pretend that the rest of the obstacles are transparent for this construction). Then consider the polygonal region P_1 enclosed by the line segment and the boundary of the obstacle from v to x. As the sweep-line moves down from v, we follow the intersection point of the boundary of P_1 with the sweep-line until we encounter a concave vertex w of the original obstacle. Note that no nodes of D_T are generated when the sweep-line meets w, since the sweep-line lies inside the obstacle around w. It is possible that before w is found another nonextreme inflection point u on the boundary of P_1 is met by the sweep-line; if that happens the graph node to the left of u is accounted by u. The proof is complete by noting that w always exists since P_1 is a polygonal region. □

b) Connectivity: Consider two points x and y in free-space and a shortest path P between them. Then P can be decomposed into line segments such that each segment is entirely contained in a trapezoid; let the resultant segments be denoted by $\overline{xp_1}, \overline{p_1p_2}, \cdots, \overline{p_{m-1}p_m}, \overline{p_my}$ listed in the sequence as we navigate from x to y. We now navigate along this sequence and generate a path on the dual graph as follows. In the trapezoid that contains x, note that p_1 lies on a sweep-line segment; now we slide the p_1-end of the segment $\overline{xp_1}$ along the sweep-line until we reach a graph node at point p_1^1. Then in the next trapezoid, we consider the segment $p_1^1p_2$, and we slide the p_2-end of this segment along the sweep-line until we meet the node p_2^1. This process is continued in the sequence until the last trapezoid is reached. Note that each slide operation is possible since each trapezoid in which the sliding operation takes place is convex. The resultant path $x, p_1^1, p_2^1, \cdots, p_m^1, y$ is a path on the graph D_T. By restricting x and y to the nodes of D_T, the connectivity property follows.

c) Local-constructibility: When a scan operation is performed from a node $v \in D_T$, the trapezoidal region that contains v will be contained in the visibility polygon returned by the scan operation. Given the sweep-line direction, the required trapezoid region can be obtained by computing the closest vertex to the sweep-line containing v in the required part of the visibility polygon. Thus local-constructibility property is satisfied.

d) Terrain-visibility: In the trapezoidal decomposition, the free-space is decomposed into trapezoidal regions and there is at least one node of V_T associated with each of the trapezoids. Since each trapezoid is convex every point in a trapezoid is visible from the corresponding node of D_T, and thus every point in the free-space is visible from some node of D_T.

In summary we have the following theorem.

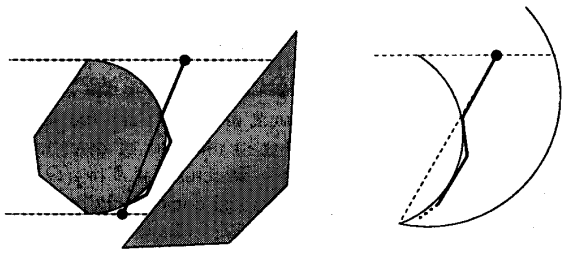
Theorem 4: $D_T(O)$ satisfies the properties of finiteness, connectivity, terrain-visibility, and local-constructibility. □

In practical cases, a scan operation could be very time-consuming. In such cases precise bounds on the nodes of the navigation course are very useful in judging the relative performance of the methods. For terrains with polygonal obstacles, the bounds on the number of vertices of a navigation course is given by $N - c$, where c is the number of concave obstacle vertices, and $5N - 2n - 3$ for the case of visibility graph and Voronoi diagram. The present navigation course D_T provides an intermediary in terms of the number of vertices of the navigation course.

Also, in practical implementations, it is difficult to navigate a robot along an obstacle boundary. The visibility graph methods require that the robot be capable of moving along the obstacle boundaries. The methods based on the Voronoi diagram deep the robot as far away from the obstacles as possible; these methods, however, have the disadvantage of generating long paths. Note that the present method again is as intermediary in that it does not require that the robot navigate along obstacle edges and also does not generate quite as long paths as the Voronoi diagram method (in cases where obstacles are well separated).

Finally, we wish to remark that other decompositions such as triangulation, convex polygonal decomposition, etc., can be used to generate suitable navigation courses. Also, even in the case of trapezoidal decomposition, there could be other ways of defining a dual graph. For example, each dual node could correspond to the centroid of a trapezoid, and a dual edge joins two nodes whose trapezoids share a sweep-line segment. Results similar to those of $D_T(O)$ can be shown by using these decompositions and dual graphs.

2) General Polygonal Terrains: Consider the trapezoidal decomposition of a terrain populated by generalized polygons such that we create sweep-line segments whenever a) the line meets an obstacle vertex or b) becomes a tangent to an obstacle boundary, by extending the line into free-space until an obstacle boundary is met or to infinity (is no obstacle is encountered). Here each trapezoid is formed by two obstacle edges and at most two sweep-line segments. Recall that if the segment is finite then the graph node is the midpoint of the segment, else it is point on the segment at a distance δ



(a) (b)

Fig. 12. Generation of edges for the augmented dual graph.(a) polygonal chain from one node to the other; (b) augmented generalized visibility graph.

from the corresponding obstacle vertex or the tangent point. There are two types of trapezoids depending on whether there are two sweep-line segments or one sweep-line segment. The edges of the navigation courses are generated as follows:

1) For each trapezoid with two sweep-line segments, two graph nodes are connected by a straight line edge if the line segment connecting them is entirely contained in a trapezoid. If this segment is intersected by a convex obstacle edge, then we place a polygonal chain from one node to the other as shown in Fig. 12(a). In general, if the location of the second graph is not known, then the initial direction of the chain towards the convex arc is not known. In such case any direction that meets the convex arc is chosen, and the rest of the chain is just as before.
2) For each trapezoid with one sweep-line segment and a pair of convex and concave arcs, we generate a polygonal chain from the node to the point of intersection of the two obstacle edges of the terrain as in the case of the augmented generalized visibility graph. See Fig. 12(b).

Thus the Augmented Dual Graph (ADG), $AD_T(O)$, is now formed by adding the nodes and edges of the required polygonal chains to the graph obtained above.

a) Finiteness: The total number of nodes of AD_T excluding the ones due to the above mentioned chains is no more than $4N$. To see this note that if no circular arcs supports a tangent in the sweep direction, then there can be no more than $2N$ graph nodes; each circular arc that supports a tangent in the sweep direction contributes to 4 graph nodes and there are no more than N such arcs. There are at most $2N$ polygonal chains and as described in the case of the generalized visibility graph (Fig. 6), and each chain consists of only a finite number of vertices. Thus in all there are only a finite number of vertices and edge of AD_T.

b) Connectivity: Now to show the connectivity property, we consider a shortest path between two points x and y of free-space. This path passes through a sequence of generalized trapezoids, and can be suitably distorted to generate a path on the dual graph AD_T as described in the case of polygonal obstacles.

c) Local-constructibility: Consider the generalized visibility polygon obtained by performing a scan operation from a node of AD_T located on a sweep-line such that a trapezoid on one side of the sweep-line is to be determined (the one on the other is already known). First we note that if the visibility polygon contains only straight-line and concave edges, then the required trapezoid can be computed. If the visibility polygons contains convex obstacle edges, we are not guaranteed to find the other sweep-line of the trapezoid, in this case we explore the convex edges by using the polygonal chains. The interior of the trapezoid will be entirely seen after the scan operations are performed as per the nodes of the polygonal chains (see Fig. 12). Thus local-constructibility is satisfied.

d) Terrain-visibility: Consider the addition of the polygonal chains as in the proof of local-constructibility. The terrain-visibility is satisfied because the interior of each trapezoid is visible within precision ϵ from a set of nodes contained in it and the union of all trapezoids is equal to the free-space.

In summary we have the following theorem.

Theorem 5: The augmented dual graph $AD_T(O)$ based on the trapezoidal decomposition of a terrain O of generalized polygonal obstacles satisfies the properties of finiteness, connectivity, terrain-visibility, and local-constructibility within a precision $\epsilon > 0$, i.e., the points that are not visible from any vertex are within a distance of ϵ from two boundary edges in a cusp region formed by a pair of convex and concave obstacle edges. □

We use the navigation courses developed in this section to solve the navigation and terrain model acquisition problem.

V. Navigational Algorithms

In the navigation problem, the task is to reach a destination position d, while avoiding obstacles on the way, or to conclude that d is not reachable. When we solve the problem with a precision ϵ, the destination d is required to be reached only if there is a path from s to g in the free-space outside the fictitious obstacles. Initially, R starts at s and performs a scan, and R moves to d, if it is reachable. If not, R computes a start vertex of the navigation course, and moves to it. Then R keeps visiting newly computed vertices until d is reachable. For example, R can employ the depth-first search to visit the vertices: R located at vertex s computes the adjacency list of s, marks s visited, and pushes s onto a stack. If any vertices adjacent to s has not been visited, R chooses one such vertex v^* and moves to it, and invokes the same algorithm recursively. Here some heuristics can be used in the selection of v^*; for example, if all obstacles are convex, it would be a good strategy to choose v^* to be of shortest straight-line distance to d among all neighbors of s. Also a more general heuristic search algorithm such as the A^* algorithm can be applied. Although any of these algorithms yields a solution to the navigation problem, the exact parameters such as the distance traversed, and number of scan opeations performed depend on the exact nature of the underlying graph search algorithm.

The solution to the terrain model acquisition problem involves visiting all vertices of the navigation course. The properties of finiteness, connectedness, terrain-visibility, and local-constructibility ensure that the navigation and the terrain model acquisition problems are solved with the specified precision.

In the case of polygonal terrains, the sufficient number of scan operations is determined by N and n. In the present case, the number of scan operations is also a function of the precision ϵ. As $\epsilon \rightarrow 0$, larger and larger number of scan operations will be required.

VI. Continuous Vision Sensors

In this section, we illustrate that a continuous vision sensor is more powerful than a discrete vision sensor: the navigation (or terrain model acquisition) problem can be solved with precision $\epsilon = 0$ in the terrains populated by generalized polygonal obstacles by using a continuous vision sensor. In terms of the general paradigm, we only require that the navigation course used by a continuous vision sensor is a 1-skeleton that satisfies the two following properties:

1) connectivity, which requires that there is a path on the navigation course between any two points of the navigation course, and
2) terrain-visibility, which requires that every point in the free-space is visible from some point on the navigation course.

We also require that the navigation course be a 1-skeleton in the plane such that the total length of the 1-skeleton is finite. given such a navigation course, if the robot has navigated along the entire navigation course, then the entire free-space will be visible. Thus a robot can navigate along the navigation course until the destination is reachable, or the entire navigation course has been traversed. Any such algorithm can be seen to solve the navigation and terrain model acquisition problems.

We now consider navigational courses for a continuous scan sensor based on simpler versions of the navigational courses of Section IV.

1. Generalized Visibility Graph: We use the generalized visibility graph as a navigation course for the continuous vision sensors. The generalized visibility graph satisfies the connectivity property as shown in Section IV. The terrain-visibility property can be shown along the lines of that for generalized visibility graph; consider a shortest path P from any point inside $CH(O)$ to a suitable point y outside $CH(O)$. As we move from x to y on P, consider the first line segment of P, given by $\overline{xp}$; here p is an obstacle vertex or $\overline{xp}$ is a tangent to a convex circular obstacle edge. In either case x will be visible from some point on the navigation course (more precisely a point on the corresponding convex obstacle boundary). Notice that no polygonal chains (of discrete scan case) are needed for ensuring the terrain-visibility for this case.

2. Generalized Voronoi Diagram: Consider the navigation course given by the structure $Vor_1(O) = (Vor(O) \cap E(O)) \cup \partial E(O)$ which is a 1-skeleton in Ω (refer to Section IV.B for the definition of the terms). Since the terrain is bounded, the total lengths of the edges of $Vor_1(O)$ is bounded. The connectivity property of $Vor_1(O)$ is shown in Section IV.B. The terrain-visibility property of $Vor(O)$ is very easily shown by using the property of the retraction $Im: \Omega \mapsto Vor(O)$; every point x in free-space is visible from $Im(x)$ since x is connected to $Im(x) \in Vor(O)$ along a straight line from $Near(x)$ through x. To show the terrain-visibility of $Vor_1(O)$ we observe that every point outside $CH(O)$ will be seen from a point on $\partial E(O)$. For any point x of free-space inside $CH(O)$, either a) $Im(x) \in E(O)$, or b) $Im(x) \notin E(O)$; in the first case x is visiblefrom a point $Vor(O) \cap E(O)$, and in the latter x is visible from the intersection point of $\partial E(O)$ and the segment joining x and $Im(x)$. Note that in the case of discrete scan sensor we have to augment the structure of $Vor_1(O)$ with additional vertices to ensure the local-constructibility.

3. Trapezoidal Decomposition: We obtain a navigation course by taking a dual graph based on the trapezoidal decomposition with the following modification.

1) If a trapezoid contains only one sweep-line segment and a pair of convex and concave obstacle edges, then we join the dual node of the trapezoid to the vertex at which the two obstacle edges meet using a smooth curve, say a Voronoi edge.
2) If the trapezoid contains two sweep-line segments and the line segment joining the two dual vertices intersects the obstacle then replace the line segment by a curve that follows the segment until it intersects the obstacle edge and then follows the obstacle edge until the other intersection point and the follows the line segment rest of the way (note that the intersection occurs only when the intersecting obstacle edge is convex). As in the case of discrete vision sensor, if the initial direction of the segment is not known, a direction that intersects the convex arc is chosen.

Connectivity of this structure is shown in Section IV.C. To show the terrain-visibility, consider a point x in free-space and the line segment through x along the sweep-line direction. If this segment coincides with a boundary of a trapezoid, then x is seen from a dual graph node. If not, x lies in a trapezoid of type i) or ii) above, or in a trapezoid containing one sweep-line segment and two obstacle edges that are line-segments or concave arcs. In the other two cases, the interior of the trapezoid is seen from the nodes of the trapezoid. In the other case the sweep-line through x must intersect the edge joining the dual node to the obstacle vertex in case i), or the dual edge that joins the dual vertices in the case ii). In either case x is visible from the corresponding intersection point.

VII. Conclusions

We considered two path planning problems in unknown two-dimensional terrains populated by disjoint obstacles whose boundaries are connected sequences of circular arcs and straight-line segments. The navigation problem deals with moving a point robot R through an unknown terrain from a source position s to a destination position d. The terrain model is not known a priori to R, but R is equipped with a vision sensor capable of detecting all visible edges of the obstacle polygons. The terrain model acquisition problem deals with autonomously building a model of the entire terrain. We first showed a general result that a solution for the navigation problem (or terrain model acquisition problem) requires an infinite number of scan operations in a cusp region consisting of a pair of convex and concave edges. Thus in practical scenarios consisting of curved objects, we either have to

use additional sensors such as touch sensors, etc., or solve the problem approximately. Following the latter approach, we consider that either problem is solved with a precision ϵ if a point p in a cusp region within a distance of ϵ from two obstacle boundaries may be considered to constitute fictitious obstacles. We proposed three navigational structures for generalized polygonal terrains that yield solutions to both the problems with a precision ϵ. The first one is obtained by suitably extending the generalized visibility graph so that it satisfies the properties of terrain-visibility and local-constructibility. The second structure is obtained by extending the generalized Voronoi diagram. The third structure is based on a dual graph defined on the trapezoidal decomposition of the free-space. An algorithm based on the augmented GVG has been implemented in C on Sun workstations using the graphic package CGI for display purposes [16]. Navigational algorithms for simulated unknown polygonal terrains have been implemented under an X-window/motif environment using visibility graph methods by Fu [9] and Sun [33] and using trapezoidal decomposition methods by Ye [37].

We barely scratched the surface of a potentially enormous area of sensor-based robot navigation algorithms. We have only shown the basic existence results for the required algorithms, and a more detailed analysis of the performance of these algorithms will be useful for practical implementations. An investigation of the impact of various graph search algorithms on the performance of the navigational algorithms is of future interest. Also algorithms that minimize, among a class of algorithms, bounds on the distance traversed or the number of scan operations will be of future interest.

ACKNOWLEDGMENT

The constructive and helpful comments from the three anonymous reviewers have greatly improved the clarity and presentation of this paper. The efforts of the ORNL reviewers E. M. Oblow and M. Beckerman are appreciated for reading several versions of this paper and providing critical comments. Implementations of various algorithms of this paper by Wei Sheng Sun, Pai Shan Lee, Ji Hua Ye, and Wenyu Fu as parts of their master's projects are also acknowledged.

REFERENCES

[1] A. V. Aho, J. E. Hopcroft, and J. D. Ullman, *The Design and Analysis of Computer Algorithms.* Reading, MA: Addison-Wesley, 1974.
[2] R. A. Baeza-Yates, J. C. Culberson, and G. J. Rawlins, "Searching in the Plane," *Information and Computation,* 1991.
[3] E. Bar-Eli, P. Berman, A. Fiat, and P. Yan, "On-line navigation in a room," in *Proc. 3rd Ann. ACM-SIAM Symp. Discrete Algorithms,* pp. 237–249, 1992.
[4] M. Barnsley, *Fractals Everywhere.* New York: Academic Press, 1988.
[5] A. Blum, P. Raghavan, and B. Schieber, "Navigation in unfamiliar terrain," in *19th 23rd Symp. Thory Computing,* pp. 494–504, 1991.
[6] L. P. Chew, "Planning the shortest path for a disc in $o(n^2 \log n)$ time," in *Proc. Symp. Computational Geometry,* pp. 214–220, 1985.
[7] J. Cox and C. K. Yap, "On-line motion planning: Moving a planar arm by probing an unknown environment," Technical report, Courant Institute of Mathematical Sciences, New York University, New York, July 1988.
[8] G. Foux, M. Heymann, and A. Bruckstein, "Two-dimensional robot navigation among unknown stationary polygonal obstacles," *IEEE J. Robotics Automat.,* vol. 9, no. 1, pp. 96–102, 1993.
[9] W. Fu, "Robot navigation in unknown terrains: Simulation under x-window/motif environment," Master's project report, Department of Computer Science, Old Dominion University, Norfolk, VA, 1993.
[10] J. Hershberger and L. J. Guibas, "An $o(n^2)$ shortest path algorithm for a non-rotating convex body," *J. Algorithms,* vol. 9, pp. 18–46, 1988.
[11] Y. K. Hwang and N. Ahuja, "Gross motion planning—A survey," *ACM Computing Surveys,* vol. 24, no. 3, pp. 219–291, 1992.
[12] M.-S. Kim, private communication.
[13] R. Klein, "Walking an unknown street with bounded detour," in *Proc. 32nd Ann. Symp. Foundations of Computer Science,* pp. 304–313, 1991.
[14] J. C. Latombe, *Robot Motion Planning.* Boston: Kluwer Academic, 1991.
[15] J. Laumond, "Obstacle growing in a nonpolygonal world," *Information Processing Lett.,* vol. 25, pp. 41–50, 1987.
[16] P. Lee, "Robot navigation using generalizd visibility graphs," Master's project report, Department of Computer Science, Old Dominion University, Norfolk, VA, 1990.
[17] T. Lozano-Perez and M. Wesley, "An algorithm for planning collision-free paths among polygonal obstacles," *Communications of the ACM,* vol. 22, no. 10, pp. 560–570, 1979.
[18] V. Lumelsky, "Algorithmic and complexity issues of robot motion in uncertain environment," *J. Complexity,* vol. 3, pp. 146–182, 1987.
[19] V. Lumelsky, S. Mukhopadhyay, and K. Sun, "Dynamic path planning in sensor-based terrain acquisition," *IEEE Trans. Robotics Automat.,* vol. 6, no. 4, pp. 462–472, 1990.
[20] V. Lumelsky and T. Skewis, "Incorporating range sensing in the robot navigation function," *IEEE Trans. Syst. Man Cyber.,* vol. 20, no. 5, pp. 1058–1069, 1990.
[21] V. J. Lumelsky and A. A. Stepanov, "Dynamic path planning for a mobile automaton with limited information on the environment," *IEEE Trans. Automat. Contr.,* pp. 1058–1063, 1986.
[22] V. J. Lumelsky and A. A. Stepanov, "Path planning strategies for a point mobile automaton moving amidsy unknown obstacles of arbitrary shapes," *Algorithmica,* vol. 2, pp. 403–430, 1987.
[23] C. O'Dunlaing and C. K. Yap, "A retraction method for planning the motin of a disc," *J. Algorithms,* vol. 6, pp. 104–111, 1985.
[24] B. J. Oomen, S. S. Iyengar, N. S. V. Rao, and R. L. Kashyap, "Robot navigation in unknown terrains using learned visibility graphs. Part I: The disjoint convex obstacle case," *IEEE J. Robotics Automat.,* vol. 3, pp. 672–681, 1987.
[25] C. H. Papadimitriou and M. Yanakakis, "Shortest paths without a map," *Theoretical Computer Science,* vol. 84, pp. 127–150, 1991.
[26] J. Pearl, *Heuristics: Intelligent Search Strategies For Computer Problem Solving.* Reading, MA: Addison-Wesley, 1984.
[27] N. S. V. Rao, "Algorithmic framework for learned robot navigation in unknown terrains," *IEEE Computer,* vol. 22, pp. 37–43, 1989.
[28] N. S. V. Rao and S. S. Iyengar, "Autonomous robot navigation in unknown terrains: Visibility graph based methods, *IEEE Trans. Syst., Man Cyber.,* vol. 20, no. 6, pp. 1443–1449, 1990.
[29] N. S. V. Rao, S. S. Iyengar, B. J. Oomen, and R. L. Kashyap, "On terrain model acquisition by a point robot amidst polyhedral obstacles," *IEEE J. Robotics Automat.,* vol. 4, pp. 450–455, 1988.
[30] N. S. V. Rao, S. Kareti, W. Shi, and S. S. Iyengar, "Robot navigation in unknown terrains: An introductory survey of non-heuristic algorithms," Technical report ORNL/TM-12410, Oak Ridge National Laboratory, Oak Ridge, TN, 1993.
[31] N. S. V. Rao, N. Stoltzfus, and S. S. Iyengar, "A 'retraction' method for learned navigation in unknown terrains," *IEEE Trans. Robotics Automat.,* vol. 7, no. 5, pp. 699–707, 1991.
[32] M. Sharir, "Algorithmic motion planning in robotics," *IEEE Computer,* pp. 9–20, March 1989.
[33] W.-S. Sun, "Implementation of navigation algorithms under X-window/Motif environment," Master's project report, Department of Computer Science, Old Dominion University, Norfolk, VA, 1993.
[34] I. Sutherland, "A method for solving arbitrary wall mazes by computer," *IEEE Trans. Computers,* vol. C-18, no. 12, pp. 1092–1097, 1969.
[35] C. K. Yap, "Algorithmic motion planning," in *Advances in Robotics, Vol. 1.* Hillsdale, NJ: Lawrence Erlbaum, 1987, pp. 95–143.
[36] C. K. Yap, "An $o(n \log n)$ algorithm for the Voronoi diagram of a set of simple curve segments," *Discrete and Computational Geometry,* vol. 2, pp. 365–393, 1987.
[37] J. Ye, "Robot navigation in unknown terrains using trapezoidal decomposition," Master's project report, Department of Computer Science, Old Dominion University, Norfolk, VA, 1993.

Nageswara S. V. Rao received the B.Tech degree in electronics and communications engineering from Regional Engineering College, Warangal, India, in 1982, the M.E. degree in computer science from the School of Automation, Indian Institute of Science, Bangalore, India, in 1984, and the Ph.D. degree in computer science from Louisiana State University, Baton Rouge, in 1988.

He is a Research Staff Member at the Center for Engineering Systems Advanced Research, Oak Ridge National Laboratory, Oak Ridge, TN, since 1993. He is an adjunct Associate Professor in the Department of Computer Science, Old Dominion University, Norfolk, VA, where he had been an Assistant Professor from 1988 until 1993. He is also an adjunct Assistant Professor in the Department of Computer Science, University of Tennessee, Knoxville, since 1994. His research interests include robot navigation, multiple sensor systems, N-learner systems, fault diagnosis and adaptive algorithms. He was a guest editor for Computers and Electrical Engineering for the special issue on Parallel and Distributed Computing for Intelligent Systems.

During his B.Tech program Dr. Rao was awarded three gold medals for securing the highest ranks in the department, college, and university. He has published more than 75 research articles in various journals, conference proceedings and books. His journal publications include articles in IEEE TRANSACTIONS ON COMPUTERS, IEEE TRANSACTIONS ON SYSTEMS, MAN AND CYBERNETICS, IEEE TRANSACTIONS ON ROBOTICS AND AUTOMATION, IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, and *Theoretical Computer Science.*