CONF-960448--15

Cooperative Terrain Model Acquisition by a Team of Two or Three Point-Robots†

Nageswara S. V. Rao
Center for Engineering Systems Advanced Research
Oak Ridge National Laboratory
Oak Ridge, Tennessee 37831-6364
raons@ornl.gov

Nachimuthu Manickam
Department of Mathematics
DePauw University
Greencastle, IN 46135
manickam@depauw.edu

V. Protopopescu
Center for Engineering Systems Advanced Research
Oak Ridge National Laboratory
Oak Ridge, Tennessee 37831-6364
protopopesva@ornl.gov

Submitted to: *1996 IEEE International Conference on Robotics and Automation*, April 22-28, 1996, Minneapolis, Minnesota.

†Research sponsored by the Engineering Research Program of the Office of Basic Energy Sciences, of the U.S. Department of Energy, under Contract No. DE-AC05-96OR22464 with Lockheed Martin Energy Research Corp.

Cooperative Terrain Model Acquisition by a Team of Two or Three Point-Robots

N. S. V. Rao, N. Manickam, V. Protopopescu

ABSTRACT

We address the model acquisition problem for an unknown planar terrain by a team of two or three robots. The terrain is cluttered by a finite number of polygonal obstacles whose shapes and positions are *unknown*. The robots are point-sized and equipped with visual sensors which acquire all visible parts of the terrain by scan operations executed from their locations. The robots communicate with each other via wireless connection. The performance is measured by the number of the sensor (scan) operations which are assumed to be the most time-consuming of all the robot operations. We employ the restricted visibility graph methods in a hierarchical setup. For terrains with convex obstacles and for teams of $n(=2,3)$ robots, we prove that the sensing time is reduced by a factor of $1/n$. For terrains with concave corners, the performance of the algorithm depends on the number of concave regions and their depths. A hierarchical decomposition of the restricted visibility graph into n-connected and $(n-1)$-or-less-connected components is considered. The performance for the $n(=2,3)$ robot team is expressed in terms of the sizes of n-connected components, and the sizes and diameters of $(n-1)$-or-less connected components.

I. INTRODUCTION

The *terrain model acquisition problem* (TMAP) deals with robots autonomously acquiring a complete model of a terrain (or environment) by systematically visiting portions of it. The motivation for this problem is at least two-fold.

(a) **Efficiency in Future Navigation:** Once the terrain model is completely acquired, the navigation algorithms of known terrains can be employed for path planning with two potential advantages. First, the sensors may be switched off (at least in theory) in future navigation, thereby avoiding the time-consuming sensor operations involved in the acquisition and processing of sensor data. Second, navigation paths with the shortest distance between the start and goal positions may be computed using the terrain model. If the terrain model is not available, no algorithm can always guarantee the shortest paths. Consequently, a robot can only recognize a dead-end corner after it has moved into it and explored it; of course, such situations can be avoided if the terrain map is a priori available.

(b) **Assistance to Human Model Builders:** In applications involving mobile robots in indoor environments for repetitive operations, typically a human operator is in charge of model building (which is tedious and time-consuming). Robots capable of autonomous terrain model acquisition (even in only small parts of the terrain) can be employed to relieve the operator from part of the work.

Research sponsored by the Engineering Research Program of the Office of Basic Energy Sciences, of the U.S. Department of Energy, under Contract No. DE-AC05-84OR21400 with Martin Marietta Energy Systems, Inc.

The first and third authors are with the Center for Engineering Systems Advanced Research, Oak Ridge National Laboratory, Oak Ridge, Tennessee 37831-6364. The second author is with the Department of Mathematics, DePauw University, Greencastle, IN 46135.

The focus of this paper is on algorithms that are *guaranteed to converge* within the assumptions of the formulation. The terrain model acquisition problem for three dimensional polyhedral terrains has been solved by using the visibility graph structure by Rao *et al.* [14] for the case of a discrete vision sensor. In the plane, the restricted visibility graph, which is obtained by removing all non-convex vertices from the visibility graph, is shown to suffice [13]. The same problem can also be solved by using a method based on the Voronoi diagram [16]. The TMAP in the case of a robot equipped with a continuous vision sensor has been solved by Lumelsky *et al.* [11]. The same problem when the obstacle boundaries consist of circular arcs and straight lines can be solved by the methods of visibility graphs, Voronoi diagram and trapezoidal decomposition using discrete and continuous vision sensors [12]. A survey of non-heuristic algorithms for navigation in unknown terrains and terrain model acquisition can be found in Rao *et al.* [15].

To our knowledge, the problem of model *acquisition* of an unknown terrain by a *team* of robots has not been addressed in the formulation of non-heuristic algorithms. This problem, however, has been studied by a number of researchers using different formulations. For example, Ishioka *et al.* [7] describe a cooperative map generation by heterogeneous autonomous mobile robots (also see Dudek *et al.* [4]). A cooperative recognition system for the environment using multiple robots has been developed by Ishiwata *et al.* [8]. Our orientation is more along the lines of the unknown terrains algorithms pioneered by Lumelsky [10].

On the other hand, the *navigation* of multiple robots in known and unknown terrains has been studied by a number of researchers. Most of the existing papers on this problem are devoted to the case of known terrains, i. e. a terrain model is supplied to the robot (Latombe [9]).

In terms of the cooperative terrain model acquisition by two robots, the formulation that comes closest to ours is the maze-exploration algorithm by two pebble automata by Blum and Kozen [1]. The communication between the pebble automata is achieved by using the pebbles that are placed on free cells. Also, the pebble automata can recognize the pebbles but are not equipped with computational mechanisms to keep track of the coordinates of the cells. On the contrary, we assume that the robots can store and transmit information with arbitrary precision.

In terms of the terrain, the maze consisting of a two-dimensional arrangement of cells [1] is much simpler than the polygonal terrain considered here. This maze exploration problem is similar to the terrain model acquisition problem here in that the automata are required to visit all free cells of the maze. For unknown terrains, the recent study by Harinarayan and Lumelsky [6] indicates that the simultaneous navigation of two robots cannot be solved if no "cooperation" is present between them. Note that our overall objective is different from theirs in two ways:

(a) we are interested in terrain model acquisition, and
(b) we wish to explore the cooperation mechanisms so that the objective can be achieved more effectively by a team of robots instead of a single robot.

In "very bad " cases, e. g. the robots are initially located at one end of a "long narrow polygonal corridor", there may not be any advantage in employing a team of robots: the robots are forced to "stay" together. However, if the terrain has "branches" so that the robots can explore different parts of the terrain, a team is likely to acquire the terrain faster than one robot.

We prove that the terrain model acquisition method based on the Restricted Visibility Graph (RVG) method [13] can be advantageously implemented by a team of two or three robots. In particular, if all obstacles are convex, the sensing time can be essentially reduced to N_c/n for $n = 2, 3$. The performance of the algorithm for general terrains depends on the number of concave regions and their depths. To tackle this situation, a hierarchical decomposition of the restricted visibility graph into n-connected and $(n-1)$-or-less-connected components is proposed. The performance for the $n(= 2, 3)$ robot team is expressed in terms of the sizes of n-connected components, and the sizes and diameters of $(n-1)$-or-less connected components. This analysis highlights the critical properties such as 2- and 3-connectivity, depth of hierarchy, etc. that support or impede the parallel acquisition of the terrain model.

The paper is organized as follows. Preliminaries are described in Section II. The TMAP in convex polygonal terrains, and along tree and 2-connected structures are discussed in Sections III and IV respectively. The TMAP in polygonal terrains is considered in Section V. Some variations of the proposed methods are presented in Section VI. Section VII contains a discussion of the results.

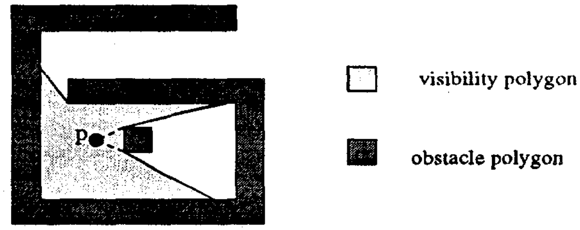


Fig. 1. Visibility polygon from location p.

II. Preliminaries

We consider a finite two-dimensional terrain cluttered by a finite and non-intersecting set of polygonal obstacles. An obstacle vertex is *convex* if the angle included inside the obstacle by the edges that meet at this vertex is less than 180 degrees; otherwise the obstacle vertex is *concave*.

Two points p and q in plane are *visible* to each other if the line segment joining p and q, denoted by $\overline{pq}$, lies entirely outside the interior of all obstacle polygons.

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 that consists of all points in the terrain visible to p (Fig. 1). We assume that the most time consuming-part of the robot operation is the scan operation. In vision-based robots, each scan may take several minutes including the time required to acquire and process the sensory data. The total *sensing time* is given by the number of scan operations performed by the robot(s) in sequence.

The robots communicate with each other in terms of finite sequences of real numbers such that a real number [1] can be communicated in a small time unit via the wireless connection.

Let $|G|$ denote the number of nodes of the graph G, and let the diameter of G, given by the number of nodes on the longest path of G, be denoted by $d(G)$. We shall also use some terminology from graph theory, e. g. connectivity, condensation, decomposition, etc., whose definition can be found in books on graph theory (e. g. Harary [5]).

The *restricted visibility graph* is defined as follows [13]. The vertices of the RVG are the convex obstacle vertices. Two vertices are connected by an edge if and only if they are visible from each other or they are the end vertices of an obstacle edge (Fig. 2). The RVG is connected and satisfies the terrain-visibility property which implies that the union of the visibility polygons of all the vertices of the RVG contains the entire free-space [13]. The latter property implies that any graph search algorithm implemented by a robot will completely acquire the terrain model in a sequence of N_c scan operations, where N_c is the number of convex vertices of the terrain.

[1]Since in general a real number carries an infinite, uncompressible amount of information, this hypothesis may seem unrealistic. However, for the specific aspects of the present problem, this is not crucial. This hypothesis is similar in spirit to the infinite precision arithmetic often assumed to be available in the study of path planning problem [9].

III. Convex Polygonal Terrains

In this section, we consider terrains composed of *convex* polygonal obstacles. The objective of the terrain model acquisition algorithm is to perform a scan operation from every node of the RVG which guarantees that the entire free-space is seen.

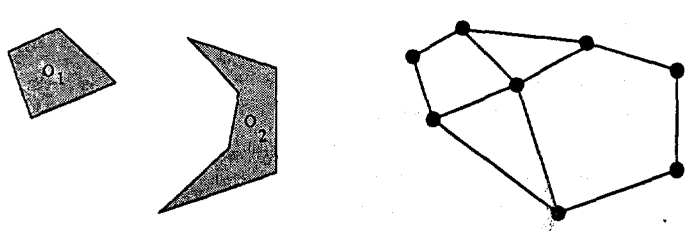
The overall algorithm for a team of n robots is based on the robots executing a graph search algorithm in a cooperative manner. At any step, each robot has the same version of an incomplete RVG. For the team of robots $R_1, R_2, \ldots, R_n$, let R_1 have the highest priority, R_2 have second highest priority, and so on. Each robot performs a scan operation and obtains the resultant visibility polygon. Each robot computes its own adjacency list and communicates it to the other robots. R_1 sends to R_2 its next destination d_1 which is one of the nodes adjacent to its present location. Then R_2 marks d_1 as visited and computes its next destination d_2. R_2 communicates d_1 and d_2 to R_3, and d_2 to R_1. This process is repeated until R_n computes its destination. Then the robots move to their chosen destinations and repeat the algorithm. For concreteness, we consider the depth-first search (Corman *et al.* [2]), where R_2 chooses an unvisited vertex adjacent to its present location, if such vertex exists. If not, R_2 backtracks along the path towards its starting vertex until it is located at vertex with an adjacent vertex that had not been visited so far or has been chosen by R_1.

Note that an adjacency list computed after a scan operation consists of a (possibly empty) set of visited vertices and not yet visited vertices. The above algorithm terminates when all known vertices have been visited; the connectivity property together with the terrain-visibility property ensure that the terrain model is completely acquired.

A. Two-Robot Team

Due to the connectivity of the RVG, R_1 is guaranteed to find a destination in each step. In order that the above algorithm be executed, we need to establish that R_2 can always find its destination. The required property is the 2-connectedness of the RVG which is established for convex polygonal terrains.

Lemma 1 *The RVG of a terrain cluttered by a finite number of convex polygonal obstacles satisfies the following properties: (a) there is a path between any two nodes u and v containing a node w, and also there is a path between u and v not containing w, and (b) there are two node-disjoint paths between any two nodes of the RVG, i. e. RVG is 2-connected.*



(a) terrain (b) restricted visibility graph

Fig. 2. Restricted visibility graph.

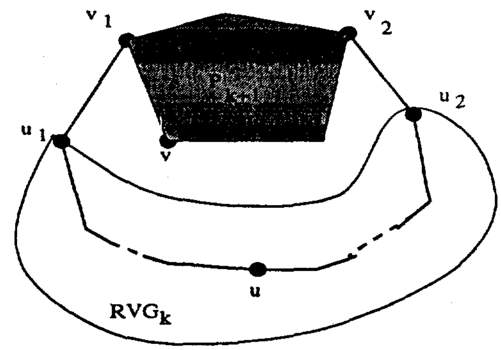


Fig. 3. Illustration of the inductive step for Lemma 1.

Proof: We prove this lemma by induction on the number of obstacles. Both Part (a) and (b) are true for a terrain consisting of a single convex polygon. Assume that the claim is true for a terrain of k obstacles; let RVG_k denote the RVG of the k obstacles. Now place the $(k+1)$th polygon P_{k+1}. The edges of the RVG_k that are intersected by the new polygon are rerouted along the boundary of P_{k+1}.

First consider Part (a). The claim is true individually for the sets of vertices of P_{k+1} and vertices of RVG_k. Now consider the properties between the vertices of P_{k+1} and RVG_k. There are at least two edges between the vertices of P_{k+1} and the vertices of RVG_{k+1} as illustrated in Fig. 3. Any node u can be included in a path between v_1 and v_2 of P_{k+1} by using the path v_1, u_1, u, u_2, v_2. Similarly any node v can be included in a path between pair w_1 and w_2 of RVG_k as follows. By Theorem 5.14 of Harary [5], there are two node disjoint paths joining w_1 to u_1 and w_2 to u_2 (since by hypothesis RGV_k is 2-connected); then the required path is given by $w_1, u_1, v_1, v, v_2, u_2, w_2$. Now consider a path between a vertex a of P_{k+1} and a vertex b on RVG_k. By connectivity of RVG_{k+1}, there is a path P_{ab} between a and b. If P_{ab} does not include a vertex v of P_{k+1}, the part of P_{ab} can be rerouted along the boundary of P_{k+1} to include v. If a vertex u of RVG_k is not included in P_{ab}, then the part of P_{ab} the lies on RVG_k can be adjusted to include u. An almost identical argument shows the second part of (a) that a chosen vertex can be excluded from the path between two vertices.

To prove Part (b), we observe that the 2-connectivity among the nodes of P_{k+1} is trivially satisfied. We now show that the required 2-connectivity among the nodes of RVG_k is preserved since no paths are broken by P_{k+1}, and any pair of vertex disjoint paths intersected P_{k+1} can be rerouted along the two opposite sides of P_{k+1} so as to preserve vertex disjointedness. First note that if two vertex disjoint paths are intersected by P_{k+1}, then it intersects two edges e_1 and e_2 of the paths. There are two cases. If e_1 and e_2 do not intersect, then the rerouting is simple as shown in Fig. 4(a). If $e_1 = (v_{11}, v_{22})$ and $e_2 = (v_{12}, v_{21})$ intersect, then the intersection can be removed by switching the paths as shown in Fig. 4(b). Here, we connect v_{11} to v_{21}

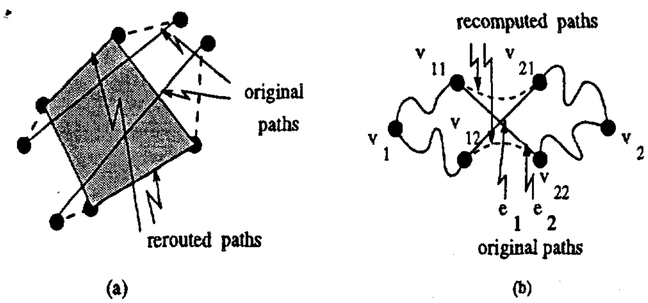


Fig. 4. Rerouting to ensure vertex disjoint pair of paths.

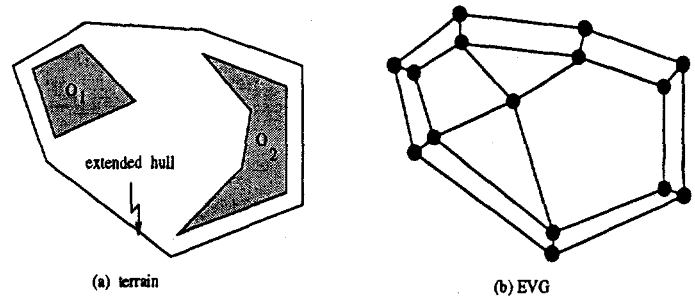


Fig. 6. Definition of the EVG.

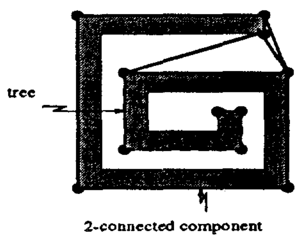


Fig. 5. Decomposition of the RVG into 2-connected parts and trees.

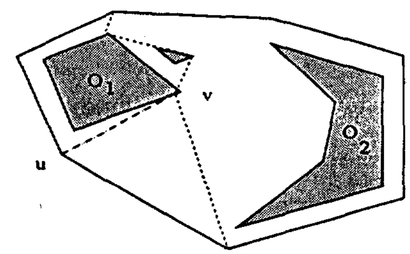


Fig. 7. Illustration for the proof of Lemma 2.

as follows (the other path is similarly constructed): rotate e_1 around v_{11} in the direction of v_{21} until v_{21} is reached or a new obstacle vertex a is reached; in the latter case the process is repeated.

Now consider the 2-connectivity between a vertex v of P_{k+1} and u of RVG_k. By hypothesis, there is a path P_u between u_1 and u_2 via u going through only the vertices of RVG_k. If the path P_u does not intersect P_{k+1}, then the paths v, v_1, u_1, u and v, v_2, u_2, u obtained by employing the pieces of P_u, are vertex disjoint. If the path P_u intersects P_{k+1}, then reroute P_u along the boundary of P_{k+1} so as to include v between u and u_1 (wlog). Then the required two paths are given by segment of P_u between u and v, and the path u, u_2, v_2, v_1, u_1, v. □

Note that 2-connectivity implies that the above RVG cannot be disconnected by removing a single vertex. Indeed, let the next destination chosen by R_1 at any step be denoted by v. By connectivity, if there is an unvisited node (other than v), then there is at least one unvisited node adjacent to the paths traced by R_1 or R_2. If not, all the unvisited nodes can only be reached via v, which makes v a cut point; this in turn contradicts the 2-connectedness of the RVG. Thus by the time R_1 performs $\lceil N_c/2 \rceil$ scan operations, R_2 would have performed scan operations from the remaining nodes of the RVG.

B. Three-Robot Team

The extended visibility graph (EVG) is the RVG augmented as follows. Consider the convex hull of the terrain which is the smallest convex region that contains all obstacles. The extended hull is obtained by expanding the convex hull by a fixed non-zero amount. Then vertices of the RVG on the convex hull are connected to the corresponding vertices on the extended hull as shown in Fig. 6. Note that the degree of all vertices in the EVG must be at least 3. To prove the main theorem we need the following result.

Lemma 2 *The EVG of a terrain cluttered by a finite number of convex polygonal obstacles is 3-connected, i. e. there exist three vertex disjoint paths between any pair of vertices.*

Proof: Any two vertices of u and v of the EVG fall into one of the situations described below:

(i) Both u and v belong to the RVG: Note that the RVG is the EVG minus the extended hull. Consider the convex hull of the polygons containing u and v. Remove all the obstacles that are outside this convex hull. Since the RVG of the resultant graph is 2-connected there must be two vertex disjoint paths between u and v without going through the extended hull. Then shrink the polygons containing u and v to point polygons and connect u and v to vertices on the extended hull such that the connecting edges are outside the above convex hull. Then there is a third vertex disjoint path between u and v along the periphery of the extended hull. Now expand the point polygons at u and v and restore the removed polygons. Then reroute the paths between u (and also v) and the extended hull along the boundaries of the restored and expanded polygons. The resultant path between u and v will be vertex disjoint from the two paths on the RVG.

(ii) Both u and v are on the extended hull: Let u' and v' be the vertices of the RVG corresponding to u and v respectively. There are two vertex disjoint paths between u and v along the boundary of the extended hull and the third path can be obtained by the shortest path between u' and v' which is guaranteed to be vertex disjoint from the boundary paths.

(iii) One of u and v belongs to extended hull and the other belongs to the RVG: Let u be on the extended hull.

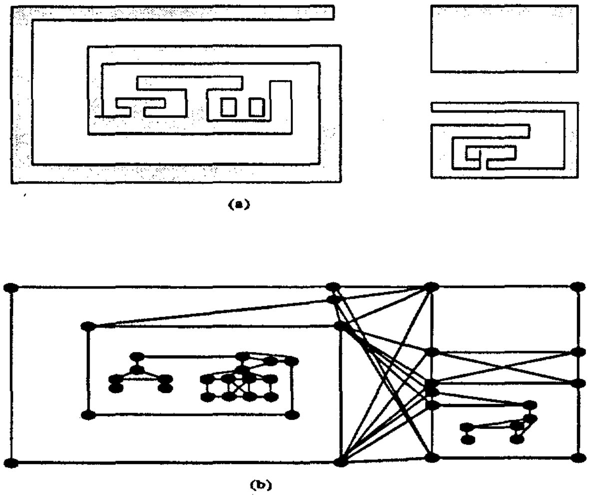


Fig. 8. Example of terrain (a) and the corresponding RVG (b).

Then the shortest path on EVG between u and v provides us with one path. We obtain two more vertex disjoint paths along the boundary of extended hull as follows. Extend the last edge of the path to the other side of v. Then rotate this extended ray around v once to the clockwise direction and once in the anti-clockwise direction. Stop the rotation when first obstacle or extended hull vertex is encountered; if it is an obstacle vertex then rotate the segment around the vertex in the same direction. This process is continued until a vertex on the convex hull is reached; then this vertex is connected to its corresponding vertex on the extended hull. It is easy to see that the paths obtained by clockwise and anti-clockwise rotations are vertex disjoint since the obstacles are convex. Then two paths along the boundary of the extended hull are easily constructed as shown in Fig. 7.

For $n = 3$, the 3-connectivity ensures that R_1, R_2 and R_3 are guaranteed to find their destinations in each step since the EVG cannot be disconnected by removing two vertices. In the actual execution of the algorithm no scan operations are performed from the vertices of the extended hull; these vertices are just computed. The following theorem is a straightforward generalization of the arguments of the last section.

Theorem 1 *The model of terrain cluttered by convex polygonal obstacles can be obtained by a team of* $n = 2, 3$ *robots in a sequence of* $\lceil N/n \rceil$ *scan operations, where* N *is the total number of obstacle vertices.*

IV. Tree and 2-Connected Components

The RVG for a polygonal terrain can be decomposed into trees and 2-connected components (see Fig. 5).

First assume that the team of robots explore a tree. Notice that in a worst case, $d(T)$ is the minimum time required to explore a tree T by a team of two robots. The strategy is for both robots to stay together until the first opportunity occurs to move along two edges of a tree. While the robots are in two different branches of the tree, sensor operations are done simultaneously. At the same time the robots will not be together for more than $d(T)$ time since the diameter is the longest possible distance (in terms of sensor operations) that the robots will stay together without branching off. To see this, assume that it is not true, then we have sequences of paths (without branching) whose total length is longer than $d(T)$; since the tree is connected and has no cycles, the union of these paths constitutes a path of length larger than $d(T)$, which is a contradiction. Thus $|T| - d(T)$ scan operations are performed while the robots are not together. Hence, the sensing time required to explore a tree is upper-bounded by $d(T) + |T|/2 - d(T)/2 = \frac{1}{2}[|T| + d(T)]$.

Now consider the case of three robots acquiring a 2-connected graph G. The strategy is to keep R_2 and R_3 together as a two-robot team, while R_1 explores in parallel. Using the argument above, the time robots R_2 and R_3 stay together does not exceed $d(G)$, and R_1 will not be forced to be together with the other robots. Thus the sensing time is upper-bounded by $d(G) + [|G| - 2d(G)]/3$. The results are summarized in the next lemma.

Lemma 3 *The sensing time of exploring a tree* T *of* $|T|$ *nodes by two robots is upper-bounded by* $d(T)/2 + |T|/2$. *The sensing time of three robots exploring a 2-connected graph* G *is upper-bounded by* $d(G)/3 + |G|/3$.

V. Polygonal Terrains

First consider the case $n = 2$ in detail. We assume that the initial location of the robots is outside the convex hull of the obstacle vertices. We identify the 2-connected component corresponding to the initial location. This 2-connected component for the RVG of Fig. 8(b) is shown in Fig. 9(a). Then we remove this component and all the trees that are emanating from it and identify the 2-connected components of the next level as shown in Fig. 9(b). The same process is repeated to identify the next levels of 2-connected components as shown in Fig. 9(c).

Trees of various levels are identified as follows. For any level, we specify the trees that emanate from the nodes of the 2-connected components of that level. Fig. 9(d) and (e) show the trees emanating from 2-connected components of level 1 and 2 respectively. There are two types of trees. The *first type* are the trees that connect the nodes of one level with nodes at another level, and the *second type* are the trees that strictly belong to one level. In Fig. 9(d), the left and right trees belong to the former type and the middle one belongs to the second type.

We obtain a *hierarchy tree* from the RVG by condensing each 2-connected component of the hierarchical decomposition to a node and removing the trees of second kind. The resultant tree is denoted by T_0. An example of hierarchy tree is shown in Fig. 10 where the hollow circles represent nodes obtained by the condensation of 2-connected components.

The overall strategy for solving the TMAP by two robots is to avoid keeping two robots in the same tree T_i to the maximum extent possible. Using this strategy, the robots

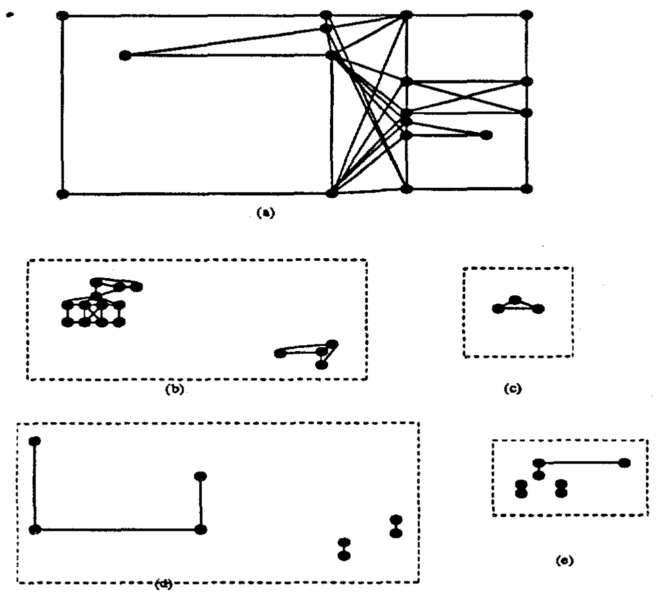


Fig. 9. Illustration of hierarchical decomposition of RVG of Fig. 6.

will explore different trees until there is at most one tree left to be (possibly partially) explored concurrently at the current level of hierarchy. This strategy can be implemented as follows. Notice that the end points of trees can be recognized by a local concavity, but a local concavity does not necessarily indicate the presence of a tree. The strategy is to delay sending the robots into the same local concavity until this becomes the only available option at that particular level.

Let us analyze the performance of the above method. Let the 2-connected components and the trees of this decomposition be denoted by $\{C_1, C_2, \ldots, C_{n_c}\}$ and $\{T_1, T_2, \ldots, T_{n_t}\}$ respectively. Since the terrain is unknown, the order in which the individual trees are explored is unknown. We carry out a worst case analysis for each of the l levels of the hierarchy. Consider the level k with 2-connected components $C_1^{2,k}, C_2^{2,k}, \ldots, C_{n_c^k}^{2,k}$, and the trees $T_{11}^k, T_{12}^k, \ldots, T_{1n_t^{1k}}^k$ and $T_{21}^k, T_{22}^k, \ldots, T_{2n_t^{2k}}^k$ of first and second type respectively. Also let

$$T_k = \{T_{11}^k, T_{12}^k, \ldots, T_{1n_t^{1k}}^k\} \cup \{T_{21}^k, T_{22}^k, \ldots, T_{2n_t^{2k}}^k\}.$$

The size of the tree that is left to be explored last is upper-bounded by

$$\max_{I,J} \left| \sum_{T \in I} |T| - \sum_{T \in J} |T| \right|$$

where the maximum is taken over all sets I and J such that $I \cup J = T_k, I \cap J = \phi, ||I| - |J|| = 1$. Now note that this quantity is upper-bounded by $\max_{T \in T_k} |T|$, which in turn implies, from Lemma 3, that the sensing time is upper-bounded by $\frac{1}{2} \max_{T \in T_k} [d(T) + |T|]$. For the level k, the number of scan operations that are performed simultaneously by the two robots is at least $\sum_{T \in T_k} |T| - \frac{1}{2} \max_{T \in T_k} [|T| + d(T)]$. Thus the total sensing time for level k required by a team of two robots is upper-bounded by

$$\frac{1}{2} \sum_{i=1}^{n_c^k} |C_i^k| + \frac{1}{2} \left[\sum_{T \in T_k} |T| - \frac{1}{2} \max_{T \in T_k} [|T| + d(T)] \right] + \frac{1}{2} \max_{T \in T_k} [|T| + d(T)]$$

$$\leq \frac{1}{2} \sum_{i=1}^{n_c^k} |C_i^k| + \frac{1}{2} \sum_{T \in T_k} |T| + \frac{1}{4} \max_{T \in T_k} [|T| + d(T)].$$

The summation of above quantity over all levels yields an upper-bound on the sensing time of the team. The summation of the first two quantities yields $\frac{1}{2} \sum_{i=1}^{n_c} |C_i|$ and $\frac{1}{2} \sum_{j=1}^{n_t} |T_i|$ respectively. The summation of the third term is handled as follows. At every level only one tree (if any) either of type one or two is last explored by the two robots (in a worst case). Thus the contribution to the upper-bound by the trees of type two is no more than $\frac{l}{4} \max_i [d(T_i) + |T_i|]$, and the contribution to the upper-bound by the trees of first type is upper-bounded by $\frac{1}{4}[d(T_0) + |T_0|]$.

We now turn to the case $n = 3$. The EVG can be decomposed into 3-connected components $C_1^3, C_2^3, \ldots, C_{n_3}^3$, 2-connected components $C_1^2, C_2^2, \ldots, C_{n_2}^2$ and trees $T_1, T_2, \ldots, T_{n_t}$. The C_i^3's can be assigned to l different levels in a hierarchical decomposition of EVG (see Section 5 for precise definitions). Some T_j's and C_j^2's connect C_i^3's of different levels while some are attached to a C_i^3 of a single level. Then the hierarchy T_0 is obtained by condensing each of C_i^3's and C_i^2's to a node and removing the 2-or-less connected components that do not connect C_i's of different levels. By using above approach and Lemma 3 to estimate the sensor time corresponding to 2-connected components, the sensing time achieved by a team of three robots is upper-bounded by

$$\frac{1}{3} \left(\sum_{i=1}^{n_3} |C_i^3| + \sum_{i=1}^{n_2} [|C_i^2| + d(C_i^2)] \right) + \frac{1}{2} \sum_{j=1}^{n_t} |T_j| + \frac{1}{4}[d(T_0) + |T_0|] + \frac{l}{4} \max_{i \in \{1, \ldots, n_t\}} [d(T_i) + |T_i|].$$

Note that the number of two connected components for the decomposition for two-robot team (n_c) is different from that for a three-robot team (n_2).

Theorem 2 *The sensing time for $n = 2, 3$ robots to acquire the model of a terrain of polygonal obstacles is upper-bounded by*

$$\frac{1}{2} \sum_{i=1}^{n_c} |C_i| + \frac{1}{2} \sum_{j=1}^{n_t} |T_i| + \frac{1}{4}[d(T_0) + |T_0|] + \frac{l}{4} \max_{i \in \{1, \ldots, n_t\}} [d(T_i) + |T_i|]$$

and

$$\frac{1}{3} \left(\sum_{i=1}^{n_3} |C_i^3| + \sum_{i=1}^{n_2} [|C_i^2| + d(C_i^2)] \right) + \frac{1}{2} \sum_{j=1}^{n_t} |T_j|$$

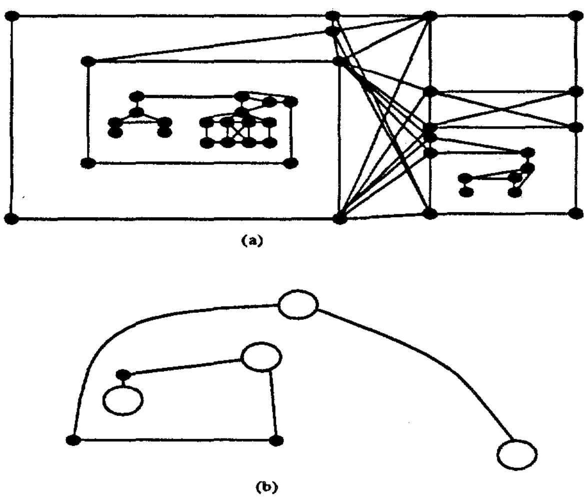


Fig. 10. Illustration of hierarchy tree.

$$+\frac{1}{4}[d(T_0)+|T_0|]+\frac{l}{4}\max_{i\in\{1,\ldots,n_t\}}[d(T_i)+|T_i|]$$

for $n=2$ and $n=3$ respectively.

For a comparison, note that the total sensing time for a single robot, based on the RVG method, is given [2]by

$$\sum_{i=1}^{n_c}|C_i|+\sum_{j=1}^{n_t}|T_i|$$

based on the decomposition for $n=2$ or by

$$\sum_{i=1}^{n_3}|C_i^3|+\sum_{i=1}^{n_2}|C_i^2|+\sum_{j=1}^{n_t}|T_i|$$

for the case $n=3$.

Notice that for terrains with convex polygonal obstacles, RVG and EVG consist of only one 2- and 3-connected component respectively. Thus this theorem subsumes Theorem 1. If the RVG is a single tree T, then $T_0=T$; for this case, since Theorem 2 yields a weaker upper-bound, it does not precisely subsume Lemma 3. The performance is decided (in a worst case) by the depth l of the hierarchy described above. For typical office indoor environments l is of the order 2. On the other hand, deeply nested mazes can generate large values for l.

VI. Variations

We now consider two geometric structures that are used for terrain model acquisition in unknown terrains.

(a) **Voronoi Diagram:** The Voronoi diagram corresponding to a set of line segments and circular arc segments has been studied by Yap [17]. The *distance* $d(p,s)$ between a point p in free-space and a boundary edge s is defined as $\inf\{d(p,q)|q\in s\}$. 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, $i=1,2,\ldots,n$ 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 $\{x\in\Omega|Near(x)$ 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). See Fig. 11 for an example.

(b) **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. 12. For each sweep-line 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 trapezoid are connected by an edge of the dual graph. See Fig. 12 for an example.

[2]It is possible to "see" the entire terrain boundary by performing less than the stated number of scan operations, but, we are unaware of any algorithm that is guaranteed to acquire the terrain model with less number of scan operations.

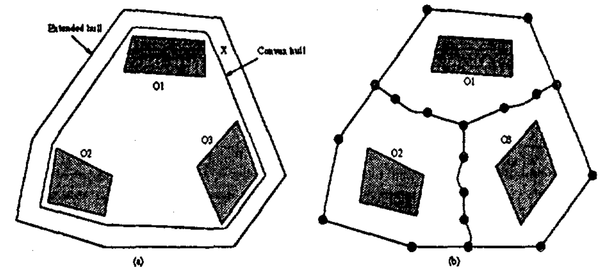


Fig. 11. Navigation course based on Voronoi diagram.

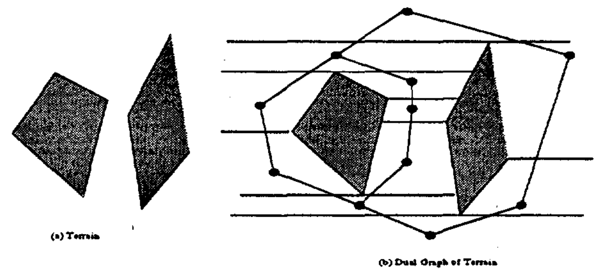


Fig. 12. Navigation course based on trapezoidal decomposition.

In terms of the sensing time for a single robot, the Voronoi diagram method could require a larger number of scan operations, whereas trapezoidal decomposition method yields about the same number as required by the RVG method. The RVG method requires that the robots be capable of navigating along the obstacle boundaries, whereas Voronoi diagram method keeps them as much away from the obstacles as possible. The trapezoidal decomposition method could require that the robot navigate close to obstacles, but less frequently than the RVG method.

Notice that both structures can be decomposed into connected components and trees and thus results along the lines of Theorem 2 can be derived for the case of two or three robots. In particular, the structure of the hierarchy tree for these two will be similar to that of the RVG.

The strategy of Section IIIA can be in principle replaced by several other methods. A possible algorithm for a team of two robots can be outlined as follows: Consider the convex hull of the terrain. The boundary of the terrain is called the *outer path* which consists of polygonal obstacle boundaries separated by straight line segments. Then obtain the *inner path* by (a) identifying the alternative paths for the non-obstacle parts of the outer path, and (b) replacing each obstacle chain of the outer path by the other path around the obstacle. An illustration is shown in Fig. 13. Then all obstacles that are part of the paths at this level are removed, and the procedure is recursively carried out. As a result we obtain layers of inner and outer paths. The algorithm for TMAP is to have the two robots move along different layers as long as possible.

VII. Discussion

This paper focuses on TMAP where it would be beneficial to employ a *team* of robots to perform a task rather than a single robot. Only the sensor time is considered here as a measure of performance, and the main discussion is based on the visibility graph methods. In this context, we have identified the parts of the terrain that can be advantageously explored in parallel and the parts in which having more than one robot may be ineffective or even wasteful (in a worst case). The estimates for the sensing time derived here are conservative. We believe that alternative characterizations and better performance estimates are possible. Also the method discussed is restricted to one particular way of solving the TMAP, namely, using a graph search on a navigation course [12]. In general these methods do not guarantee that the sensing time or the distance traversed by a single robot is close to the optimal achievable if the terrain model is known. The recently studied class of competitive algorithms for the TMAP by Deng *et al.* [3] guarantee that the distance traversed by a single robot is bounded within a factor of the minimum possible value achieved if the terrain model is available. Improving the performance of the algorithms of this type by employing a team of robots will be of future interest.

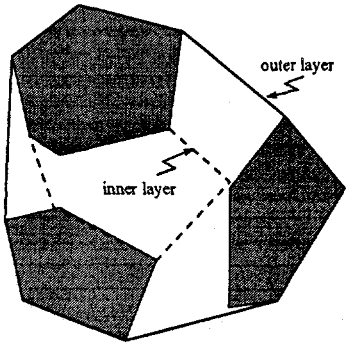


Fig. 13. Illustration of inner and outer layers.

The effectiveness of employing a team of robots for the TMAP might be judged by other measures of performance such as distance traversed, total time of sensor operations, travel time, etc. For example in the RVG method for a single robot, the distance traversed in solving the TMAP is a function of the search algorithm employed, whereas the sensor operations is given by N_c (fixed for a terrain). The analysis of the parameter such as the distance appears to be significantly difficult even for the case of the RVG and warrants further research.

References

[1] M. Blum and D. Kozen. On the power of compass (or why the mazes are easier to search than graphs). In *18th Annual Symposium on Foundations of Computer Science*, pages 132–142, October 1978.

[2] T. H. Cormen, C. E. Leiserson, and R. L. Rivest. *Introduction to Algorithms*. MIT Press, Cambridge, MA, 1990.

[3] X. Deng, T. Kameda, and C. H. Papadimitriou. How to learn an unknown environment. In *Proc. 32nd Ann. Symp. on Foundations of Computer Science*, pages 298–303, 1991.

[4] G. Dudek, M. Jenkins, E. Milios, and D. Wilkes. Robust positioning with a multi-agent robotic system. In *1993 IJCAI Workshop Series: Dynamically Interacting Robots*, pages 118–123, 1993. Working Notes.

[5] F. Harary. *Graph Theory*. Addison-Wesley Pub. Co., Reading, MA, 1969.

[6] K. R. Harinarayan and V. J. Lumelsky. Sensor-based motion planning for multiple mobile robots in an uncertain environment. In *Proceedings of IEEE/RSJ/GI Int. Conf. on Intelligent Robots and Systems*, pages 1485–1492, 1994.

[7] K. Ishioka, K. Hiraki, and Y. Anzai. Cooperative map generation by heterogeneous autonomous mobile robots. In *1993 IJCAI Workshop Series: Dynamically Interacting Robots*, pages 57–67, 1993. Working Notes.

[8] Y. Ishiwata, M. Inaba, and H. Inoue. Cooperative recognition of environments by multiple robots. In *Proc. of JSME Annual Conf. on Robotics and Mechatronics*, pages 79–84, 1992.

[9] J. C. Latombe. *Robot Motion Planning*. Kluwer Academic Pub., Boston, 1991.

[10] V. Lumelsky. Algorithmic and complexity issues of robot motion in uncertain environment. *Journal of Complexity*, 3:146–182, 1987.

[11] V. Lumelsky, S. Mukhopadhyay, and K. Sun. Dynamic path planning in sensor-based terrain acquisition. *IEEE Transactions on Robotics and Automation*, 6(4):462–472, 1990.

[12] N. S. V. Rao. Robot navigation in unknown generalized polygonal terrains using vision sensors. *IEEE Transactions on Systems, Man and Cybernetics*, 1994. to appear.

[13] N. S. V. Rao and S. S. Iyengar. Autonomous robot navigation in unknown terrains: Visibility graph based methods. *IEEE Transactions on Systems, Man and Cybernetics*, 20(6):1443–1449, 1990.

[14] 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 Journal of Robotics and Automation*, 3:450–455, 1988.

[15] 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.

[16] N. S. V. Rao, N. Stoltzfus, and S. S. Iyengar. A 'retraction' method for learned navigation in unknown terrains. *IEEE Transactions on Robotics and Automation*, 7(5):699–707, 1991.

[17] C. K. Yap. An $O(n \log n)$ algorithm for the Voronoi diagram of a set of simple curve segments. *Discrete and Computational Geometry*, 2:365–393, 1987.

DISCLAIMER