

Simultaneous Cooperative Exploration and Networking Based on Voronoi Diagrams

Jonghoek Kim, Fumin Zhang, and Magnus Egerstedt*

* *School of Electrical and Computer Engineering, Georgia Institute of Technology, USA (e-mail: jkim37@gatech.edu, fumin, magnus@ece.gatech.edu)*

Abstract: We develop a strategy that enables multiple intelligent vehicles to cooperatively explore complex territories. Every vehicle deploys communication devices and expands an information network while constructing a topological map based on Voronoi diagrams. As the information network weaved by each vehicle grows, intersections eventually happen so that the topological maps are shared. This allows for distributed vehicles to share information with other vehicles that have also deployed communication devices. Our exploration algorithms are provably complete under mild technical assumptions. A performance analysis of the algorithms shows that in a bounded workspace, the time spent to complete the exploration decreases as the number of vehicles increases. We further provide an analytical formula for this relationship. Time efficiency of the algorithms is demonstrated in MATLAB simulation.

Keywords: Cooperative exploration, Voronoi diagram, Sensor network, Information network

1. INTRODUCTION

Exploration of complex and dangerous territories posts great challenges for robotics research. For mobile robotics, the recent developments in simultaneous localization and map making (SLAM) (Thurn and Burgard (2005), Durrant-Whyte and Bailey (2006), Bailey and Durrant-Whyte (2006)) have provided valuable techniques to answer some of the challenges. Intuition suggests that using cooperative multiple vehicles will increase time efficiency. Coordination of multiple vehicles typically relies on communication between vehicles, but direct communication is easily blocked or at least attenuated by obstacles. Hence one major challenge for a successful multi-vehicle strategy is the lack of line of communication (LOC).

Recently, small and low power devices with short communication range, such as the Berkeley MOTES, become commercially available. A large number of such devices are deployed to form a sensor network in which each pair of communication devices has LOC between each other (Culler et al. (2004)). Each device serves as an information node and together they form an information network that relays messages between devices.

These developments have inspired us to investigate a strategy called Simultaneous Cooperative Exploration and NeTworking (SCENT). In this paper, we present SCENT algorithms that construct Voronoi diagrams as topological maps of the workspace. The workspace is considered completely explored if Voronoi diagrams, as defined in Kim et al. (2009), are fully constructed. The vehicles are initially deployed at arbitrary locations and are not necessarily aware of the existence of other vehicles. Each vehicle starts with boundary expansion (BE) algorithms developed in our recent work (Kim et al. (2009)) to explore its surroundings and to construct an information network.

Every vehicle drops communication devices and expands an information network while constructing a topological map, called the communication graph, based on Voronoi diagrams. As the information network weaved by each vehicle grows, intersections eventually happen so that the topological maps are shared by two or more vehicles. These shared maps allow for distributed vehicles to share information with other vehicles that have also dropped communication devices. This is a key feature that distinguishes SCENT algorithms from other approaches that only allow robot-to-robot communication.

Using BE algorithms (Kim et al. (2009)), each vehicle expands explored area not overlapping the area explored by other vehicles. However, one vehicle may find itself “boxed in” by areas already explored by other vehicles, i.e., the expansion of explored area is blocked by areas already explored by other vehicles. This case, using shared maps to detect unexplored region of the workspace, this blocked vehicle is redirected to unexplored region. In addition, we use shared maps to reduce the chance of occurrence of blocking events for multiple vehicles. We prove that such a cooperative strategy, based on shared maps, leads to time efficient construction of Voronoi diagrams in an unknown workspace.

Voronoi diagrams have been widely used for topological maps in robotics, c.f. (Choset et al. (1996); Lavelle (2006)), as well as for studying coverage problems in sensor networks (Cortés et al. (2004); Martinez et al. (2007)). This paper provides provably complete algorithms for constructing Voronoi diagrams.¹ A performance analysis of the algorithms shows that in a bounded workspace, the time spent to complete the exploration decreases as the

¹ Due to space limitations, our algorithms in this paper are not presented in the standard format, but rather described in English to increase readability.

number of vehicles increases. Also, time efficiency of the algorithms is demonstrated in MATLAB simulation.

Although many results exist in literature regarding construction of Voronoi diagrams, to our knowledge, SCENT algorithms are unique in proof, with MATLAB demonstration, of time efficiency for deploying multiple vehicles.

2. ALGORITHMS FOR A SINGLE VEHICLE

In this section, we briefly review BE algorithms (Kim et al. (2009)) that construct Voronoi diagrams of the workspace using a single vehicle.

2.1 Definitions and Assumptions

Consider a connected and compact workspace $W \subset R^2$ whose boundary, ∂W , is a regular curve. Let O_1, O_2, \dots, O_{M-1} be $M - 1$ disjoint, and compact obstacles such that $O_i \subset W$. O_M is a “virtual” obstacle that bounds the workspace, i.e., $\partial W \subset \partial O_M$. We denote the set of obstacles S_O by $S_O = \{O_1, O_2, \dots, O_M\}$.

We define the *Voronoi cell* for an obstacle O_i as the set of points that are closer to O_i than to any other obstacle in S_O for $i = 1, 2, \dots, M$. $\partial V(O_i)$ is the boundary of the Voronoi cell for O_i , i.e., $V(O_i)$. The *Voronoi diagram* of the workspace is defined as the union of all cell boundaries (Klein (1990)). A Voronoi edge between two Voronoi cells $V(O_i)$ and $V(O_j)$ is $E_{ij} = \partial V(O_i) \cap \partial V(O_j)$.

We define an *intersection* P as a point at which the following conditions are satisfied:

- (1) there exists a circle centered at the point P intersecting obstacle boundaries at more than two points. These points are called the *closest points* at the intersection.²
- (2) interior of the circle does not intersect any obstacles. The circle is called an *intersection circle*, and illustrated in Fig.1.

The lines connecting the intersection and the closest points on the obstacle boundaries partition the intersection circle into *sectors*. We can see that each sector is the “pie shaped area” within the intersection circle as seen in Fig. 1.

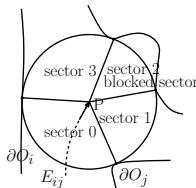


Fig. 1. The position of a vehicle is at the intersection. The *sector* i is the sector adjacent to the sector $i - 1$ in the counterclockwise direction.

The vehicle under control moves along E_{ij} until it visits an intersection P , as illustrated on Fig. 1. It will detect two closest points on ∂O_i and ∂O_j , since $P \in E_{ij}$. The sector that has these two closest points as its end points is defined as *sector 0* for the intersection P . Suppose that

² Suppose that the vehicle is at an intersection, then the closest points correspond to the points that have local minimal distances to the vehicle.

there are n sectors in the intersection circle as seen on Fig.1. Looking into the page, we then index the sectors in the counterclockwise direction from sector 0. The index k satisfies $0 \leq k \leq n - 1$. When two end points of a particular sector are on the same obstacle, the sector is called a *blocked sector* that is illustrated as “sector 2”. An *open sector* denotes a sector that is neither a blocked sector nor a sector 0, illustrated as “sector 1” and “sector 3” in Fig. 1.

If the intersection detected by a vehicle has an open sector that has not been visited by the vehicle, then the intersection is marked as *unexplored*. Otherwise, the intersection is marked as *explored*. The following assumptions are made about the workspace and the vehicle’s sensing and localization capability.

- (A1) $\partial V(O_i)$ is a simple closed curve for all $O_i \in S_O$. In other words, $\partial V(O_i)$ is continuous and no self-intersection occurs.
- (A2) there are finitely many intersections in W . All blocked sectors for these intersections are detectable by the vehicle.
- (A3) $\bigcup_{O_i \in S_O} \bar{V}(O_i) = W$.
- (A4) the initial position of a vehicle is such that an obstacle other than O_M is detected to the right of the vehicle³. The vehicle can distinguish O_M from other obstacles.

2.2 Expanding the Enclosing Boundary

We call a closed loop that contains intersections connected by Voronoi edges an *enclosing boundary* if there is no unexplored intersection strictly inside such loop and the loop has no self-intersection.

BE algorithms first construct an enclosing boundary that contains only one obstacle, then the enclosing boundary is expanded by adding one new obstacle at a time. At any moment in BE algorithms, the enclosing boundary is unique.

More specifically, the initial enclosing boundary is denoted as B_0 . Then, we update B_0 to obtain B_k for $k = 1, 2, \dots$ until B_k encloses all the obstacles except for O_M . There are $k + 1$ obstacles inside B_k where $0 \leq k \leq M - 2$.

We expand B_k while maintaining it as a simple closed curve tracked by the vehicle in the clockwise direction. This expansion is performed by first moving through an open sector of an intersection on B_k to construct a candidate segment formed by Voronoi edges, and then to replace certain segment of B_k with the candidate segment.

A set of rules are designed in Kim et al. (2009) to expand the enclosing boundary. We have proved in Kim et al. (2009) that the algorithms finish in finite time and a complete Voronoi diagram is obtained as a result.

3. SCENT ALGORITHMS

In this section, we present SCENT algorithms by extending the algorithms for a single vehicle to multiple vehicles.

³ Assumption (A4) is strictly speaking not a restriction, since the vehicle can initialize the heading orientation so that an obstacle other than O_M is detected to the right of the vehicle. In the case where multiple vehicles are involved, assumption (A4) is applied to every vehicle.

We denote a vehicle as v^i where $1 \leq i \leq N_v$ and N_v is the number of vehicles. Every vehicle v^i deploys communication devices on intersections. If necessary, communication devices are deployed on long Voronoi edges in order to relay data from one intersection to another intersection that is out of maximum radio range. These communication devices then form an information network. The methods for communication devices deployment by a robot platform are research topics that are not the focus of this paper. We also assume the information network is in place once the communication devices are deployed.

Let *enclosure* denote the area inside the enclosing boundary. Since multiple vehicles are involved, one major modification over BE algorithms is to expand the enclosing boundary for v^i in such a way that the enclosure built by v^i does not overlap with the enclosures built by other vehicles. Let B^i denote the enclosing boundary built by v^i . When v^i visits an intersection on $B^i \cap B^j$, v^i moves along the edges $B^i \cap B^j$ without expanding B^i . This in turn avoids overlaps.

Our results in this section require some basic knowledge of graph theory (Ji et al. (2008)). An undirected graph G is defined by a set $N(G)$ of nodes and a set $E(G) \subset N(G) \times N(G)$ of edges. Two nodes x and y are *neighbors* if $(x, y) \in E(G)$. A graph G is *connected* if there is a path connecting every pair of distinct nodes. The *subgraph* G_s of G is the pair $(N(G_s), E(G_s))$ where $N(G_s) \subset N(G)$ and $E(G_s) = \{(x, y) \in E(G) : x \in N(G_s), y \in N(G_s)\}$. We can write $G_s \subset G$.

3.1 Communication Graph

We define a *communication graph* as the graph where every node represents a deployed communication device and every edge represents a communication link. The nodes and edges of the communication graph are time-varying, since new node and edges are added to the graph when a vehicle deploys a communication device. For each vehicle, we distinguish three subgraphs : $G^i(t)$, $\hat{B}^i(t)$, and $C^i(t)$.

$G^i(t)$ where $i = 1, 2, \dots, N_v$ is the communication subgraph where every node represents a communication device deployed by the vehicle v^i . Since B^i is used to represent the enclosing boundary built by v^i , we use the notation $\hat{B}^i(t)$ for the communication subgraph where all nodes are on the enclosing boundary B^i . $N(\hat{B}^i(t))$ is the set of nodes along B^i , and $E(\hat{B}^i(t))$ is the set of edges of $\hat{B}^i(t)$. A special case here is that we allow $i = 0$ so that $\hat{B}^0(t)$ is the subgraph where all nodes are on $\partial V(O_M)$.

Suppose that a communication device is already deployed by v^i at an intersection P and that v^j visits P . Then, through the communication device deployed at P , v^i can relay data structure of $G^i(t)$ and $\hat{B}^i(t)$ to v^j and vice versa. Note that v^j does not have to drop a communication device at P , since communication link is already established through the communication device deployed by v^i .

Once communication link is established between a device deployed by v^i and a device deployed by v^j , both vehicles become aware of the structure of the combined communication graph. In this way, each vehicle v^i builds a combined communication graph $C^i(t) = (N(C^i(t)), E(C^i(t)))$ that

is the maximally connected graph such that $G^i(t) \subset C^i(t)$ where $i = 1, 2, \dots, N_v$. The relationship among the communication subgraphs is that $\hat{B}^i(t) \subset G^i(t) \subset C^i(t)$. Every vehicle v^i stores one $C^i(t)$ and all $\hat{B}^j(t)$ where index j is determined such that $G^j(t) \subset C^i(t)$.

The vehicle v^i uses $C^i(t)$ to find an unexplored intersection for building a new enclosing boundary. This is presented in the next subsection.

3.2 Resolve blocking or overlapping events

Every vehicle v^i expands B^i in a way that the enclosure of B^i does not overlap with the enclosure of B^j where $j \neq i$. However, there may be the case where the expanding enclosing boundary B^k is blocked by the enclosing boundaries constructed by other vehicles as illustrated on Fig.2. *Blocking* of B^k denotes the situation when $E(\hat{B}^k(t)) \subset \bigcup_{n \neq k} E(\hat{B}^n(t))$.

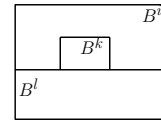


Fig. 2. The expanding enclosing boundary B^k is blocked by the enclosing boundaries constructed by other vehicles.

Another situation that is associated with blocking is overlapping. This can happen if the initial enclosing boundaries built by two different vehicles are identical. This is possible at the beginning of the exploration, if the initial position of v^i is that the obstacle to the right of v^i is also to the right of v^j . If $N(\hat{B}^i(t)) \subset N(\hat{B}^j(t)) (j \neq i)$ when B_0^i is built, then we define this case as *overlapping* of B^i .

In the case where blocking or overlapping occurs, we redistribute the blocked or overlapped vehicle to an unexplored intersection where a new enclosing boundary can be built. As long as there is an unvisited Voronoi edge in W , unexplored intersection exists on $C^i(t)$ for every vehicle v^i . This is stated as Lemma 1.

Lemma 1. If there exists an unvisited Voronoi edge in W , then there exists an unexplored intersection on $C^i(t)$ for all i .

Proof. We prove by contradiction. Suppose that all the intersections on $C^i(t)$ are explored. This implies that all the Voronoi edges connected to $C^i(t)$ are already visited by vehicles, and communication devices are deployed along the edges. Thus, the edge set of $C^i(t)$ does not contain any edges that lead to unexplored regions. This can only be true if $C^i(t)$ has all the Voronoi edges in W , which implies that all the Voronoi edges in W have been visited by vehicles. This is a contradiction. \square

The redirecting strategy works as follows. When blocking or overlapping occurs, then v^i searches for an unexplored intersection on $C^i(t)$. Note that this unexplored intersection will not lie on a blocked enclosing boundary. This is stated as the following Lemma 2.

Lemma 2. If an unexplored intersection is found on $C^i(t)$, this unexplored intersection is not on a blocked enclosing boundary.

Proof. We prove by contradiction. Suppose that an unexplored intersection is found on a blocked enclosing boundary B^j . This implies that there exists an unvisited edge that intersects the unexplored intersection. This unvisited edge can not be inside B^j , because there is no unexplored intersection inside B^j (Theorem 2 in Kim et al. (2009)). Since B^j is blocked, we have $B^j \subset \bigcup_{n \neq j} B^n \cup \partial V(O_M)$. This implies that there exists B^k such that the unvisited edge is inside B^k where $B^j \cap B^k \neq \emptyset$. However, this unvisited edge can not be inside B^k either, because there is no unexplored intersection inside B^k (Theorem 2 in Kim et al. (2009)). Therefore, unexplored intersection can not exist on a blocked enclosing boundary. \square

By applying the breadth-first search algorithm on $C^i(t)$, v^i can find the shortest (hop distance) path from the current position of v^i to all the unexplored intersections on $C^i(t)$. Among these unexplored intersections, v^i selects the one with the smallest hop distance and marks it as Q_{v^i} . The position of Q_{v^i} is relayed (broadcasted) across $C^i(t)$ to all other vehicles sharing $C^i(t)$. In the case where v^j visits Q_{v^i} , v^j ignores Q_{v^i} without changing B^j . In this way, Q_{v^i} is “reserved” for v^i until it is reached by v^i . Once v^i reaches Q_{v^i} , it builds a new enclosing boundary.

There may be the case where blocking or overlapping event occurs for another vehicle v^j ($j \neq i$) while v^i is moving toward Q_{v^i} . This case, among unexplored intersections that are not marked as Q_{v^i} , v^j selects the one with the smallest hop distance and marks it as Q_{v^j} . Q_{v^j} is relayed (broadcasted) across $C^j(t)$ to all other vehicles sharing $C^j(t)$, and v^j moves along the shortest path to reach Q_{v^j} .

This strategy relies on the availability of at least one unexplored intersection for each blocked or overlapped vehicle. Hence, we make the following assumption :

- (A5) When blocking or overlapping event occurs for v^i , there exists at least one unexplored intersection on $C^i(t)$, which has not been marked as Q_{v^j} by some other vehicle v^j ($j \neq i$).

This assumption seems strong. It is possible that blocked or overlapped vehicle v^i can not choose Q_{v^i} on $C^i(t)$, since every unexplored intersection on $C^i(t)$ has been already marked as Q_{v^j} by some other vehicle v^j ($j \neq i$). Thus, we develop a rule, which is introduced in the next subsection, so that we can reduce the chance of occurrence of blocking events for multiple vehicles.

3.3 Avoid blocking using the communication graph

Every vehicle v^i obeys the following *blocking avoiding rule* to avoid blocking event of some other vehicle v^k ($k \neq i$) such that $\hat{B}^k(t) \subset C^i(t)$.

- If expansion of B_n^i , which denotes the enclosing boundary for v^i updated after n steps, leads to blocking of some other vehicle v^k (i.e., $E(\hat{B}^k(t)) \subset \bigcup_{m \neq k} E(\hat{B}^m(t))$ where $k \neq i$ and $\hat{B}^m(t), \hat{B}^k(t) \subset C^i(t)$), then the expansion will not be performed.

Notice that if expansion of B_n^i leads to blocking of v^i itself (i.e., $E(\hat{B}^i(t)) \subset \bigcup_{m \neq i} E(\hat{B}^m(t))$ where $\hat{B}^m(t) \subset C^i(t)$),

then the expansion will be performed and the blocking of B^i will occur.

Fig. 3 illustrates the case where we avoid blocking event of B^k . B^i is not expanded even though v^i has moved along the arrows into the shaded area. Since the shaded area is not occupied by B^i , enclosure inside B^k can expand to the shaded area not overlapping the enclosures for other vehicles. This prevents the occurrence of blocking event depicted in Fig. 2.

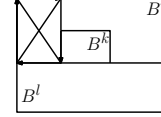


Fig. 3. Blocking of B^k is avoided by preventing the expansion of B^i into the shaded region.

The procedure in avoiding blocking only works for the connected graph $C^i(t)$. There exist situations where $C^i(t)$ and $C^l(t)$ are not connected. In Fig.4, we illustrate this case where $C^l(t)$ is not connected to $C^i(t)$. Here, existence of $C^l(t)$ is unknown to the vehicles that are only aware of $C^i(t)$. In this situation, the expansion of $\hat{B}^l(t)$ will eventually be blocked.

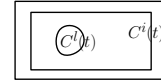


Fig. 4. $C^l(t)$ is not connected to $C^i(t)$. Hence, existence of $C^l(t)$ is unknown to the vehicles that are only aware of $C^i(t)$.

4. PERFORMANCE ANALYSIS

In this section, we provide an analytical formula for the total time spent on cooperative exploration of a regularized workspace. Each obstacle, other than O_M , is now simplified as one site (called a generator in Du et al. (1999)), then the workspace is partitioned by a centroidal Voronoi tessellation. In the case where there are sufficiently many Voronoi cells, each can be shown to be of hexagonal shape Du et al. (1999); Newman (1982).

We analyze the performance of our algorithms in this workspace where each cell has a hexagonal shape with identical size. Hexagonal Voronoi cells can be built using identical circular obstacles as illustrated on Fig. 5.

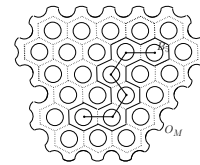


Fig. 5. All Voronoi cells, except for $V(O_M)$, have hexagonal shapes with identical size. Line segment connecting two centers of Voronoi cells inside B_k represents the adjacency of two Voronoi cells inside B_k .

4.1 Time upper bound of algorithms using one vehicle

We first present the time upper bound for a single vehicle to explore the entire bounded workspace using BE algorithms.

Theorem 3. Consider a single unit speed vehicle and workspace W with assumptions (A1)-(A4) satisfied. Suppose that there are M obstacles such that all obstacles, except for O_M , have hexagonal Voronoi cells with identical size. Using BE algorithms, the exploration time is bounded above as $T_c < T(\frac{5}{2}(M-1)^2 - \frac{5}{2}(M-1) + 1)$ where T denotes the time for a vehicle to traverse along the edges of one hexagonal Voronoi cell.

Proof. Recall that $k + 1$ obstacles are inside B_k . Since B_k is a simple closed curve, B_k divides hexagonal Voronoi cells into two groups : $k + 1$ Voronoi cells inside B_k , and Voronoi cells outside B_k .

Consider the case where $k = 0$. Since there is only one Voronoi cell inside B_0 , the time to construct B_0 is

$$T_{B_0} = T, \quad (1)$$

where T denotes the time for a vehicle to traverse along the edges of one hexagonal Voronoi cell. Next, consider the case where $k > 0$, i.e., there are more than one Voronoi cell inside B_k . In this case, any Voronoi cell inside B_k is adjacent to at least one other Voronoi cell inside B_k as illustrated on Fig. 5. Since there are at most 6 adjacent Voronoi cells for every hexagonal Voronoi cell, any Voronoi cell inside B_k has at most 5 adjacent Voronoi cells outside B_k . Thus, $k+1$ Voronoi cells inside B_k have at most $5(k+1)$ adjacent Voronoi cells outside B_k .

Hence, an upper bound for the number of Voronoi cells, which are outside B_k , intersecting the perimeter of B_k is $5k + 5$. Using Theorem 2 in Kim et al. (2009), at least one of the Voronoi cells, which are outside B_k , intersecting the perimeter of B_k is an addable obstacle. Recall that an addable obstacle is an obstacle that is inside B_{k+1} . Therefore, the vehicle's maximal traversal distance between the generation of B_k and that of B_{k+1} is $(5k+5)T$, since the vehicle has a unit speed. Thus, we have

$$T_{B_{k+1}} < T_{B_k} + (5k + 5)T, \quad (2)$$

where T_{B_k} denotes the time for a vehicle to construct B_k . Using (2), we obtain

$$\begin{aligned} T_{B_k} &< 5(1 + 2 + \dots + (k-1))T + 5kT + T_{B_0} \\ &= T(\frac{5}{2}k^2 + \frac{5}{2}k + 1), \end{aligned} \quad (3)$$

since $T_{B_0} = T$ using (1). There are $k + 1$ and $M - 1$ obstacles inside B_k and $\partial V(O_M)$ respectively. Therefore, our algorithms terminate when

$$k + 1 = M - 1. \quad (4)$$

Hence, replacing $k + 1$ in (3) by $M - 1$, we obtain the time upper bound for the construction of Voronoi diagrams as

$$T_c < T(\frac{5}{2}(M-1)^2 - \frac{5}{2}(M-1) + 1). \quad (5)$$

Therefore, expected exploration time is $O((M-1)^2)$. \square

4.2 Time upper bound of algorithms using multiple vehicles

We present the time upper bound for multiple vehicles to explore the entire bounded workspace using SCENT algorithms.

Theorem 4. Consider unit speed vehicles and workspace W with assumptions (A1)-(A5) satisfied. Suppose that there are M obstacles such that all obstacles, except for O_M , have hexagonal Voronoi cells with identical size. Also, suppose that there exist N_v vehicles and that every vehicle explores W using SCENT algorithms. Let T denote the time for a vehicle to traverse along the edges of one hexagonal Voronoi cell. The time required for a vehicle to traverse along the perimeter of W is $T_o(M)$ where M is used to indicate that T_o is a function of M . Then the time to construct a complete Voronoi diagram is bounded above by $T + \lceil \frac{M-1}{N_v} \rceil (T_o(M) + T) + \frac{5}{2}T \lceil \frac{M-1}{N_v} \rceil + \frac{5}{2}T(\lceil \frac{M-1}{N_v} \rceil)^2$.

Proof.

Suppose that blocking occurs $l - 1$ times before SCENT algorithms for v^i terminate. Since the number of obstacles is finite, l is also finite. Let k_j denote the updated step of enclosing boundary before each blocking occurs. In other words, after the enclosing boundary is updated after k_j steps, blocking occurs and a new enclosing boundary is built at a new position. Then the enclosing boundary is updated after k_{j+1} steps before blocking occurs again. This way, boundary update occurs in the order of $k_1 \rightarrow \dots \rightarrow k_l$ and blocking occurs between $k_j \rightarrow k_{j+1}$ where $1 \leq j \leq l - 1$. Similar to (4), SCENT algorithms for v^i finish when

$$\sum_{j=1}^l (k_j + 1) = \lceil \frac{M-1}{N_v} \rceil, \quad (6)$$

since SCENT algorithms for v^i end when sum of the area inside the enclosing boundaries built by v^i is greater than $\frac{A_w}{N_v}$. Here, A_w denotes the area inside $\partial V(O_M)$. In (6), the ceiling function is used since k_j is integer. Rearranging LHS of (6), we obtain

$$\sum_{j=1}^l k_j = \lceil \frac{M-1}{N_v} \rceil - l \geq 0, \quad (7)$$

where inequality holds, since $k_j \geq 0$ for all j .

Once blocking of B^i happens, v^i chooses Q_{v^i} among unexplored intersections on $C_i(t)$ and moves along the shortest path to reach Q_{v^i} . Note that Q_{v^i} is reserved for v^i to build a new enclosing boundary. Since v^i moves along the shortest path on $C_i(t)$ to reach Q_{v^i} , the time to reach Q_{v^i} is upper bounded by $T_o(M)$ that is the time required for a vehicle to traverse along the perimeter of W . Since blocking occurs $l - 1$ times, we get the upper bound of exploration time as

$$\begin{aligned} T_c &\leq T_{B_0} + T_o(M) + \sum_{j=1}^l T(\frac{5}{2}k_j^2 + \frac{5}{2}k_j + 1) \\ &\quad + (l-1)T_o(M), \end{aligned} \quad (8)$$

where (3) is used. On RHS of (8), $T_{B_0} + T_o(M)$ is added considering the overlapping of B^i . Here, T_{B_0} is the time required to build B_0^i , since overlapping of B^i is detectable

only after B_0^i is built. Once overlapping of B^i is detected, v^i chooses Q_{v^i} among unexplored intersections on $C_i(t)$ and moves along the shortest path to reach Q_{v^i} . The time to reach Q_{v^i} is upper bounded by $T_o(M)$.

Next, we express (8) as a function of N_v and $M-1$. Using (7) and (8), we obtain

$$T_c < T_{B_0} + lT_o(M) + lT + \frac{5}{2}T(\lceil \frac{M-1}{N_v} \rceil - l) + \frac{5}{2}T(\lceil \frac{M-1}{N_v} \rceil - l)^2, \quad (9)$$

since $\sum_{j=1}^l (k_j^2) \leq (\sum_{j=1}^l k_j)^2$. Furthermore, using (7) and (1), we get

$$T_c < T + \lceil \frac{M-1}{N_v} \rceil (T_o(M) + T) + \frac{5}{2}T \lceil \frac{M-1}{N_v} \rceil + \frac{5}{2}T(\lceil \frac{M-1}{N_v} \rceil)^2. \quad (10)$$

□

In the case where $\frac{M-1}{N_v} \gg 1$, (10) is approximated as

$$T_c < \frac{M-1}{N_v} (T_o(M)) + \frac{5}{2}T(\frac{M-1}{N_v})^2. \quad (11)$$

Suppose that $\frac{M-1}{N_v} (T_o(M))$ is a dominant term on RHS of (11). This case, as N_v increases by n times, the upper bound of exploration time decreases by n times. However, in the case where $\frac{5}{2}T(\frac{M-1}{N_v})^2$ is a dominant term on RHS of (11), the upper bound of exploration time decreases by n^2 times as N_v increases by n times.

Next, consider the case where N_v increases to ∞ ($\lceil \frac{M-1}{N_v} \rceil \rightarrow 1$). Then, from (10), we obtain

$$T_c < T_o(M) + 7T, \quad (12)$$

which implies that, as the number of vehicle increases, the effectiveness of adding more vehicles decreases. However, we acknowledge that, as N_v increases, assumption (A5) gets more difficult to be satisfied.

5. MATLAB SIMULATION RESULTS

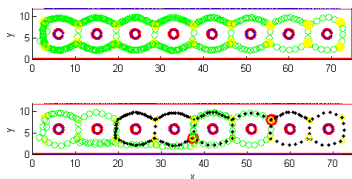


Fig. 6. Construction of a Voronoi diagram in a line shaped workspace.

In Fig. 6, the upper figure shows a single vehicle constructing a Voronoi diagram in a line shaped workspace. The obstacle boundary is shown in red, and the segments of obstacle boundary detected by the range sensors are shown in blue. The initial position of the vehicle is (1.5, 4), and the trajectory of the vehicle is marked with green circle. On the vehicle's trajectory, intersections are marked with large yellow dots. The exploration time using a single vehicle is 45.98 time unit.

The below figure in Fig. 6 shows two vehicles constructing a Voronoi diagram using SCENT algorithms. The trajectory of two vehicles are marked with green circle and black point respectively. The initial positions of two vehicles are (1.5, 4) and (19.5, 6) respectively. Recall that once v^i is blocked, it builds a new enclosing boundary at a reserved intersection Q_{v^i} . Reserved intersections are marked with large red dots ((38, 3) and (55, 8)) in Fig. 6. The exploration time using two vehicles is 16.26 time unit which is almost one third of the exploration time using one vehicle.

6. CONCLUSION

We develop SCENT algorithms for multiple vehicles. We prove that SCENT algorithms, based on communication graph, leads to time efficient construction of Voronoi diagrams in an unknown workspace. Also, time efficiency of SCENT algorithms is demonstrated in MATLAB simulation.

REFERENCES

- Bailey, T. and Durrant-Whyte, H. (2006). Simultaneous localization and map making: Part II. *IEEE Robotics and Automation Magazine*, 108–117.
- Choset, H., Konukseven, I., and Burdick, J. (1996). Mobile robot navigation: issues in implementing the generalized voronoi graph in the plane. In *Proc. of IEEE/SICE/RSJ International Conference on Multi-sensor Fusion and Integration for Intelligent Systems*, 241 – 248. Washington DC, USA.
- Cortés, J., Martínez, S., Karatas, T., and Bullo, F. (2004). Coverage control for mobile sensing networks. *IEEE Transactions on Robotics and Automation*, 20(2), 243–255.
- Culler, D., Estrin, D., and Srivastava, M. (2004). Overview of sensor networks. *IEEE Computer Magazine*, 37(8), 41–49.
- Du, Q., Faber, V., and Gunzburger, M. (1999). Centroidal voronoi tessellations : Applications and algorithms. *Society for Industrial and Applied Mathematics*, 41, 637–676.
- Durrant-Whyte, H. and Bailey, T. (2006). Simultaneous localization and map making: Part I. *IEEE Robotics and Automation Magazine*, 13, 99–108.
- Ji, M., Ferrari-Trecate, G., Egerstedt, M., and Buffa, A. (2008). Containment control in mobile networks. *IEEE Transactions on Automatic Control*, 53, 1972–1975.
- Kim, J., Zhang, F., and Egerstedt, M. (2009). An exploration strategy based on construction of voronoi diagrams. In *IEEE Conference on decision and control (accepted)*. Shanghai, China.
- Klein, R. (1990). *Concrete and Abstract Voronoi diagrams*. Springer.
- Lavalle, S.M. (2006). *Planning Algorithms*. Cambridge University Press.
- Martinez, S., Cortes, J., and Bullo, F. (2007). Motion coordination with distributed information. *IEEE Control Systems Magazine*, 27(4), 75 – 88.
- Newman, D. (1982). The hexagon theorem. *IEEE Transactions on Information Theory*, 28, 137–139.
- Thurn, S. and Burgard, W. (2005). *Probabilistic Robotics*. MIT, Cambridge, MA.