A Repartitioning Algorithm to Guarantee Complete, Non-overlapping Planar Coverage with Multiple Robots

Kurt Hungerford, Prithviraj Dasgupta and K.R. Guruprasad

Abstract We consider the problem of coverage path planning in an initially unknown or partially known planar environment using multiple robots. Previously, Voronoi partitioning has been proposed as a suitable technique for coverage path planning where the free space in the environment is partitioned into non-overlapping regions called Voronoi cells based on the initial positions of the robots, and one robot is allocated to perform coverage in each region. However, a crucial problem arises if such a partitioning scheme is used in an environment where the location of obstacles is not known a priori—while performing coverage, a robot might perceive an obstacle that occludes its access to portions of its Voronoi cell and this obstacle might prevent the robot from completely covering its allocated region. This would either result in portions of the environment remaining uncovered or requires additional path planning by robots to cover the disconnected regions. To address this problem, we propose a novel algorithm that allows robots to coordinate the coverage of inaccessible portions of their Voronoi cell with robots in neighboring Voronoi cells so that they can repartition the initial Voronoi cells and cover a set of contiguous, connected regions. We have proved analytically that our proposed algorithm guarantees complete, non-overlapping coverage. We have also quantified the performance of our algorithm on e-puck robots within the Webots simulator in different environments with different obstacle geometries and shown that it successfully performs complete, non-overlapping coverage.

Keywords Multi-robot systems · Coverage path planning · Voronoi partitioning

K. Hungerford · P. Dasgupta (✉)
Computer Science Department, University of Nebraska, Omaha, USA
e-mail: pdasgupta@unomaha.edu

K. Hungerford
e-mail: khungerford@unomaha.edu

K.R. Guruprasad
Department of Mechanical Engineering, National Institute of Technology, Mangaluru, Karnataka, India
e-mail: krgprao@gmail.com

N.-Y. Chong and Y.-J. Cho (eds.), *Distributed Autonomous Robotic Systems*,
Springer Tracts in Advanced Robotics 112, DOI 10.1007/978-4-431-55879-8_3

1 Introduction

Coverage path planning is a central aspect of multi-robot systems where the objective is to completely cover the surface area of an environment using multiple robots. Robotic coverage is used in several application domains of robots such as unmanned search and rescue, clearing an area of landmines, inspecting the health of engineering structures, as well as in civilian applications such as automated lawn mowing and vacuum cleaning. Using multiple robots for area coverage instead of a single robot offers several advantages such as reducing the time required to complete the environment's coverage and improving the robustness of the system to failure of a single or a few robots. However, using multiple robots also introduces the overhead of coordination between robots to avoid collisions and perform non-overlapping coverage. An attractive technique to implement non-overlapping coverage between robots is to partition the free space of the environment into disjoint regions or cells that can then be covered by robots [1–4]. In most of these partitioning-based coverage techniques, the cellular partitions are not changed once they have been determined. However, if the obstacles inside the environment are initially unknown to the robots, a robot might discovers that a cell is occluded by an obstacle while performing coverage. As shown in Fig. 1a, a robot then has to use path planning techniques to explore paths to reach the cell's occluded part. In a multi-robot coverage scenario, the path planning technique to reach the occluded portions of a cell involves significant computation and coordination between the robots [3], which might result in increased battery expenditure and completion time for the coverage. Therefore, it makes sense to investigate techniques that can reduce or avoid these path planning costs for robots by adaptively repartitioning cells and reallocating the repartitioned portions, so that other robots can cover the repartitioned cell with little overhead for navigation planning.

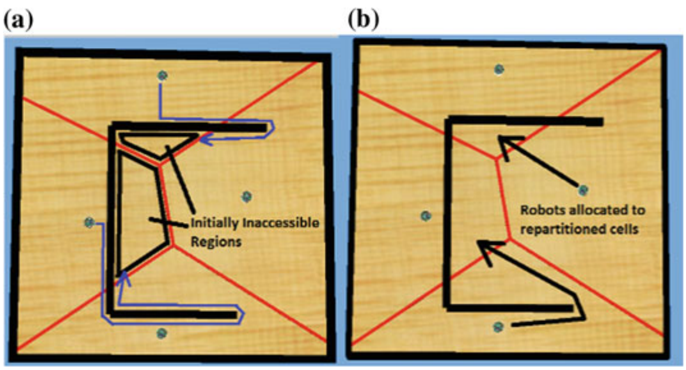

Fig. 1 **a** The Voronoi cells of two robots are partially inaccessible due to obstacles. The *blue solid arrows* show the path taken by a robot to reach the inaccessible portions of its cell using a bug-like path planning algorithm. **b** Robots coordinate with each other to repartition the initial Voronoi cells so that each robot has a contiguous region to cover

Our research in this paper is based on the key insight that when the initial partition of the environment is done equitably between robots, exactly one robot occupies a cell. Then, even if the cell that a robot is covering gets disconnected due to obstacles, because the free space is connected, the inaccessible portion of the cell must be adjacent to at least one of the neighboring cells and accessible to the robot in that cell. Consequently, the robot performing coverage in the adjacent neighboring cell could be requested to augment its coverage with the inaccessible portion of the disconnected cell, as shown in Fig. 1b. Based on this insight, we first partition the environment into complete non-overlapping cells using Voronoi partitioning [5] and then propose a novel algorithm called *Repart-Coverage*, where each robot performs boundary coverage of its initially allocated cell or region and then uses a low-overhead coordination protocol with other robots to systematically repartition only those portions of its cell that are inaccessible due to obstacles. We have shown analytically that our proposed technique guarantees complete, non-overlapping coverage. We have also verified the performance of the *Repart-Coverage* algorithm on simulated e-puck robots within the Webots simulator for different environments and different obstacle geometries and quantified its performance in terms of the areas of coverage regions and distances traveled by the different robots due to repartitioning of their initial cells.

2 Related Work

Coverage path planning has been a central topic in robot motion and an excellent survey is given in [6], including both single and multi-robot coverage. For multiple networked robots performing distributed coverage, the coordination strategies that have been proposed can be divided into two broad categories. In the first category, robots share maps of covered regions with each other while they perform coverage so that they can coordinate their movements to avoid each others' regions. In most of these techniques, the environment is divided into grid-based cells corresponding to the footprint of a robot. Robots then use different techniques to avoid repeated coverage such as sensing and avoiding already-covered neighboring cells [7], recording the regions covered by each robot as a coverage tree [8] and communicating the boundaries of covered regions between the robots, and, using a negotiation protocol along with a distance-based objective function to select regions to cover for different robots [9]. In [3], the authors proposed a technique called multi-robot Boustrophedon decomposition where the robots decompose the environment into cells in an online manner while performing coverage. Robots use two different roles - boundary coverage and area coverage. A pair of boundary coverage robots move in tandem along two parallel but opposite boundaries of the environment and infer about the presence of obstacles when the line of sight between them gets blocked. This information is used to define cell boundaries for subsequent coverage by the area coverage robots. The algorithm can guarantee complete, non-overlapping coverage, but the robots have to use complex calculations and tight coordination to guarantee that cell boundaries are correctly identified and multiple robots are not assigned to cover the same cell.

In contrast, in the second category of coverage coordination, the environment is partitioned into non-overlapping cells based on the initial positions of robots using strategies such as polygonal decomposition [2], Voronoi partitioning [4, 5, 10], etc. Recently, while extending this approach, Breitenmoser et al. have proposed an algorithm where robots initially partition the environment using Voronoi cells and start navigating towards target locations while continuously adapting the partitions and refining the target locations as they discover obstacles [11]. In [4, 12], the authors have proposed a multi-robot coverage technique where each robot communicates its position while it moves and dynamically adapts the partitions with neighboring robots to guarantee complete, non-overlapping partition of the environment. In contrast to our work, they do not explicitly address situations that prevent the complete coverage of a Voronoi cell assigned to a robot when a portion of the cell becomes inaccessible to the robot due to obstacles. Since the focus of our paper is on partitioning the environment for coverage, we use a boundary coverage algorithm called Egress [13] that enables a robot to determine and follow the boundary of its currently assigned region; we assume that suitable techniques for covering the internal area of a region such as ladder search [2] or spanning tree coverage(STC) [14] are utilized by the robot after it has determined the boundary of the region it has to cover. Also, in the rest of the paper we have used the term coverage to refer to boundary coverage.

3 Problem Formulation

Let $Q \subset \mathbb{R}^2$, a convex polygon, represent a region occupied by a set of obstacles O. Let $Q_{\text{free}} = Q \backslash O$, denote the free space within Q. We assume Q_{free} to be a topologically connected set. Our objective is to perform complete, non-overlapping coverage of the region $Q \backslash O$, using N autonomous mobile robots, each equipped with a coverage tool. Let $\mathcal{P}(t) = \{p_i(t) \in Q, i \in I_N\}$, where $p_i(t)$ denotes the position of the ith robot at time t.[1]

The *Voronoi partition*, generated by $\mathcal{P}$ is the collection $\{V_i(\mathcal{P})\}_{i \in I_N}$ with,

$$V_i(\mathcal{P}) = \left\{q \in Q| \parallel q - p_i \parallel \leq \parallel q - p_j \parallel, \forall j \in I_N\right\} \tag{1}$$

The Voronoi partition induces an undirected graph known as Delaunay graph, $\mathcal{G}_D$, where two nodes $i, j \in I_N$ are neighbors if the intersection of corresponding Voronoi cells V_i and V_j is a line segment. The set of neighbors of the node i is denoted as $N(i)$; for brevity we assume $N_i =| N(i) |$. Let B_{ij} denote the perpendicular bisector

[1] Robots can assume a well-distributed initial configuration in case their initial positions are close to each other using techniques in [11, 15].

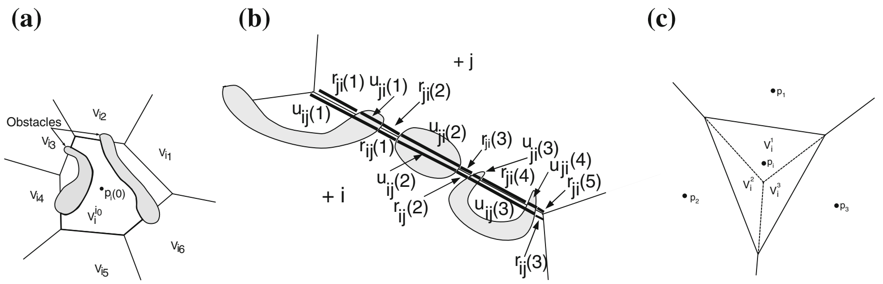

Fig. 2 **a** The region bounded by *dark lines* is $V_i^{i_0}$. **b** Illustration of A_{ij}^b and A_{ij}^f. **c** Illustration of V_i^j when V_i is repartitioned between robots $j \in N(i)$

of line joining $p_i(0)$ and $p_j(0)$ and let $A_{ij} \subseteq B_{ij}$ represent the common boundary between V_i and V_j. Let $\mathcal{C} = \{C_1, C_2, \ldots, C_M\}$ be a partition of Q_{free}. Let $S_i \in 2^{\mathcal{C}}$, $i \in I_N$, and each S_i, $i \in I_N$ is made up of contiguous cells from $\mathcal{C}$, that is, $\bigcup_{C_j \in S_i} C_j$ is a (topologically) connected set.

Distributed spatial partitioning problem: For each $i \in I_N$, the ith robot should construct S_i, a contiguous collection of topologically connected cells, such that the collection $\mathcal{S} = \{S_1, S_2, \ldots, S_N\}$ partitions Q_{free}.

3.1 Definitions and Notations

Let $V_i^{i_0} \subset V_i$ be the subset of V_i containing $p_i(0)$. If there are no obstacles within V_i, then $V_i^{i_0} = V_i$. The boundary of $V_i^{i_0}$ is made up of portions of A_{ij} and obstacle boundaries. A point $q \in A_{ij}$ is reachable to robot i from $p_i(0)$, if $q \in V_i^{i_0}$, and unreachable otherwise. Figure 2a illustrates $V_i^{i_0}$ with an example.

Let $A_{ij}^b = \{u_{ij}(k) | A_{ij} \supset u_{ij}(k) \notin V_i^{i_0}\}$, where $u_{ij}(k)$s are mutually disjoint convex sets, representing parts (line segments) of A_{ij} that are not reachable (blocked by obstacles) to the robot i. Similarly, let $A_{ij}^f = \{r_{ij}(k) | A_{ij} \supset r_{ij}(k) \in V_i^{i_0}\}$, where $r_{ij}(k)$s are mutually disjoint convex sets, representing parts (line segments) of A_{ij} that are reachable (not blocked by obstacles) to the robot i. See Fig. 2a for illustration. Note that $A_{ij}^f = A_{ij} \setminus A_{ij}^b$.

Let $N^{fb}(i) = \{j | A_{ij}^b = A_{ij}\} \subset N(i)$. When $j \in N^{fb}(i)$, entire A_{ij} is unreachable to the robot i; then the robot i can not enter V_j without entering V_k, for some $k \notin \{i, j\}$. Let $N^b(i) = \{j | A_{ij}^b \neq \emptyset\} \subseteq N(i)$. Note that $N^{fb}(i) \subset N^b(i) \subseteq N(i)$.

Note that $A_{ij} = A_{ij}^b \cup A_{ij}^f$, and $A_{ij}^b \cap A_{ij}^f = \emptyset$, thus A_{ij}^b and A_{ij}^f partition A_{ij}. If $A_{ij} = A_{ij}^b$ (that is, $A_{ij}^f = \emptyset$), then we say that A_{ij} is impermeable to the robot i. If $A_{ij} = A_{ij}^f$, then we say that A_{ij} is fully permeable to the robot i. If $A_{ij}^b \neq \emptyset$ and

$A_{ij}^f \neq \emptyset$, then A_{ij} is partially permeable to the robot i. Note that $A_{ij} = A_{ji}$, However $A_{ij}^b \neq A_{ji}^b$, and $A_{ij}^f \neq A_{ji}^f$, in general.

Let $V_j^i \subset V_j$, for $j \in N(i)$, be a portion of V_j that would have been part of V_i with node set $I_N \backslash \{j\}$. See Fig. 2c for illustration. $V_j^i = V_j \cap ({}^j\tilde{V}_i)$, where ${}^j\tilde{V}_i$ is the Voronoi cell of i with nodes $I_N \backslash \{j\}$, or just $N(j)$. Each portion of $V_i \backslash V_i^{i_0}$, is part of V_i^j, for some $j \in N(i)$. If A_{ij} is fully impermeable to the robot i, that is, $A_{ij} = A_{ij}^b$, then i will not be able to reach V_j^i.

4 Distributed Spatial Partitioning

In this section, we explain the proposed distributed spatial repartitioning scheme. The ith robot first explores $V_i^{i_0}$ and obtains the following information: (i) V_i, $N(i)$, $p_i(0)$, $p_i(t)$, $p_j(0) \in Q, \forall j \in I_N$ the position of itself and initial positions of all other robots; (ii) A_{ij}, A_{ij}^b, and A_{ij}^f, for each $j \in N(i)$; (iii) the sets $N^{fb}(i) \subset N^b(i) \subset N(i)$, and iv) $V_i^{i_0}$. Now,the robot broadcasts the following information: A_{ij}^b, A_{ij}^f, $\forall j \in N(i)$, and the sets $N^{fb}(i)$, $N^b(i)$, and $N(i)$. This communication is required only at the beginning of the distributed spatial partitioning.

Now the robot uses the available information and further exploration when required, to decide on the additional regions that need to be covered by it. The free regions in $V_j \backslash V_j^j$, $j \in I_N$ can not be covered by robot j and hence need to be covered by other robots. These regions are divided into *patches*. A patch is defined as a connected subset of Voronoi cells. Each patch is bounded by obstacles and/or line segments of B_{jk}, for some $j, k \in I_N$. The ith robot maintains a set S_i of patches it should cover. It is clear that $V_i^{i_0}$ is a patch in S_i. The ith robot adds to S_i patches in $(V_j \backslash V_j^j)_{\text{free}}$—the portion of obstacle free region with V_j, not accessible directly to the robot j, $j \in I_N \backslash \{i\}$. We say that two patches U and W are adjacent, if $U \cap W$ contains a line segment in B_{jk} (not necessarily A_{jk}), for some $j, k \in I_N$ (j and k are not necessarily neighbors)[2]. The significance of two patches U and W being adjacent is that a robot can move freely between these patches. The patches are created as robots explore the regions to be covered. We will discuss the process of constructing S_i in steps.

Scenario i. Patches in V_j^i, $j \in N(i)$ The robot i enters $V_j^i \subset (V_j \backslash V_j^{j_0})_{\text{free}}$, $j \in N(i)$, if and only if $\exists l \in \{1, 2, \ldots, | A_{ij}^f |\}$, s.t. $r_{ij}(l) \cap A_{ji}^b \neq \emptyset$. This condition is illustrated in Fig. 3a. This patch, say U_1, is adjacent to $V_i^{i_0}$ and is added to S_i.

Scenario ii. Patches in V_j^k, $k, j \in N(i), k \in N(j)$**:** If the robot i enters a patch $U_1 \subseteq V_j^i$, it explores U_1. If a portion U_2 of V_j^k, $k \in N(i) \cap N(j)$ is adjacent to U_1, then robot i will find out if k can reach this portion of V_j^k. Otherwise, this portion of

---

[2]As U and W belong to free space, $U \cap W$ is either $\emptyset$ or a permeable line segment.

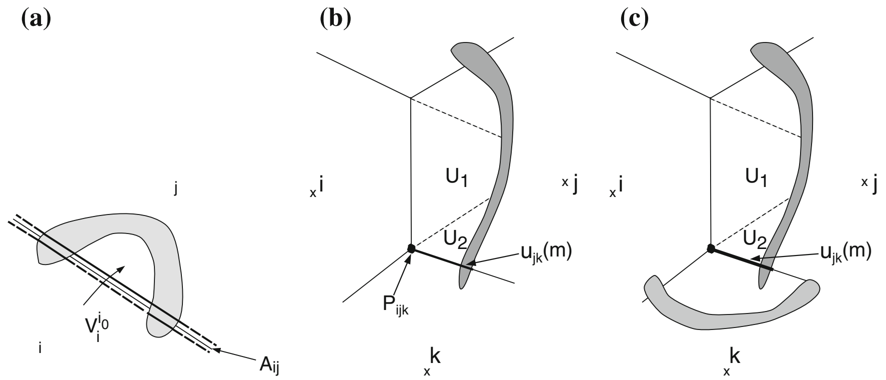

Fig. 3 **a** Robot i can help robot j to cover ${}^i(V_j^i)_{\text{free}}$. *Thick solid* and *dashed lines* represent the blocked and free components of A_{ij} respectively. **b–c** Conditions i checks to find out if it has to help a common neighbor k to cover a portion of the region $(V_j^k)^f$

V_j^k will be added S_i. We will discuss the situations in which robot i should or should not cover a patch in V_j^k. Let P_{ijk} be the vertex common to V_i, V_j, and V_k.

1. Consider a scenario, as illustrated in Fig. 3c, where $U_1 \cap V_j^k$ is a single line segment and $P_{ijk} \in U_1$. Let $u_{jk}(m) \in A_{jk}^b$ contains P_{ijk} (Such $u_{jk}(m)$ exists as P_{ijk} is assumed to be part of U_2 adjacent to U_1).

   1a. If $u_{jk}(m) \cap A_{kj}^f \neq \emptyset$, as illustrated in Fig. 3c, a, then k can reach U_2, and hence i will not cover it.

   1b. Otherwise, as illustrated in Fig. 3c, b, k can not reach U_2 and i should cover it. The robot i can check if $P_{ijk} \in U_1 \cap U_2$, and if $U_1 \cap U_2$ is a single connected piece, while physically exploring the boundary of U_1.

2. Consider a scenario, $U_1 \cap V_j^k$ is a not a single line segment or $P_{ijk} \notin U_1$, as illustrated in Fig. 4. In such a scenario, robot i will not be able to decide if U_2 needs to be added to S_i or not only based on available information. The patch U_2 is added to S_i, only if, while physically exploring the boundary of U_2, the robot i reaches a portion of A_{kj}^f.

Remark 1 Note that the robot i physically explores the boundary of a patch W which is adjacent to $U \in S_i$, only when the information about free and blocked regions of Voronoi cell boundaries (A_{ij}) is not sufficient to make a decision as to if W needs to be added to S_i. Such an exploration is local to the robot i and it does not affect the decisions of other robots. This can be observed from the illustrations in Fig. 4. The patch U_2 is added to S_i (Fig. 4a and b) when i concludes that $U_2 \notin S_k$, and is not added to S_i (Fig. 4c) when $U_2 \in S_k$. This ensures that the patch U_2 is covered exactly by one robot.

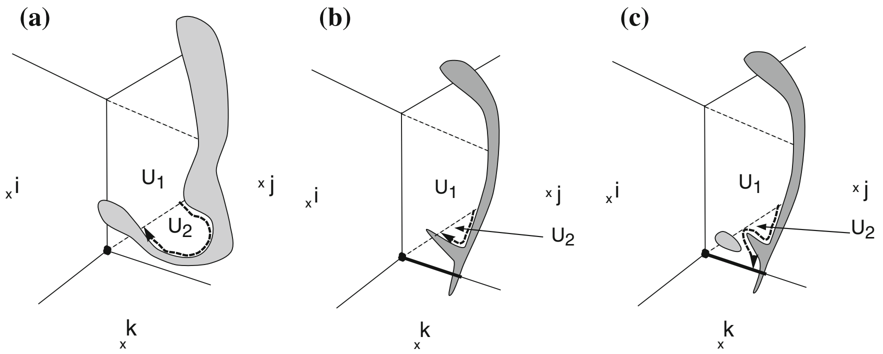

Fig. 4 The robot i explores $U_2 \subseteq V_j^k$ to check if k can reach it. The exploration path is shown with *dark dashed line* ending with an arrow. **a** U_2 is unreachable to the robot k, hence it is added to S_i. **b** Though U_2 is reachable to the robot k, it has to pass through U_1 to reach it. In other words, U_2 is not adjacent to V^k ($V^k \cap U_2 = \emptyset$). Thus $U_2 \in S_i$. **c** The robot i reaches a point on A^f_{kj} while exploring U_2 and hence $U_2 \notin S_i$ (as $U_2 \in S_k$)

Further, it can be noted the scenarios discussed above are exhaustive.

Scenario iii. Patches in V_j^l, $j \in N(i), l \in N(j), l \notin N(i)$ If $V_j^i \supset U_1 \in S_i$ and $U_2 \subset V_j^l$, s.t. U_2 is adjacent to U_1, then robot i has to make a decision on adding U_2 to S_i.

(1) If $V_j^l \backslash V_j^{j_0}$ is not accessible to robot l (based on A^f_{lj} and A^b_{jl}, discussed in scenario(i)), then U_2 is added to S_i. Such scenarios are illustrated in Fig. 5a and b.
(2) Otherwise, the robot i should explore[3] the boundary of entire portion of $V_j^l \backslash V_j^{j_0}$ connected to U_2. Only if no portion of A^f_{lj} is reached during this exploration, U_2 is added to S_i. Figure 5c shows a scenario where U_2 is not added to S_i after a point on A^f_{lj} is reached while exploration (indicating that $U_2 \in S_l$). Figure 5d shows a scenario where U_2 is added to S_i after the robot i fails to reach any point on A^f_{lj} while exploring (indicating that $U_2 \notin S_l$).

Scenario iv. A patch in $V_i \backslash V_i^{i_0}$ While the robot i is in $U_1 \subset V_i \backslash V_j^i$ for some $j \in N(i)$ and $U_1 \in S_i$, and a patch $U_2 \subset V_i \backslash V_i^{i_0}$ is adjacent to U_1, U_2 is added to S_i only if there is no $k \in N(i)$ such that U_2 is adjacent to $V_k^{k_0}$.A condition under which the robot i adds U_2 to S_i is illustrated in Fig. 5e. If U_2 is adjacent only to U_1, that is $P_{ijk} \notin U_1 \cap U_2, k \in N(i) \cap N(j)$, then U_2 is added to S_i.

[3]Remark 1 is also applicable here.

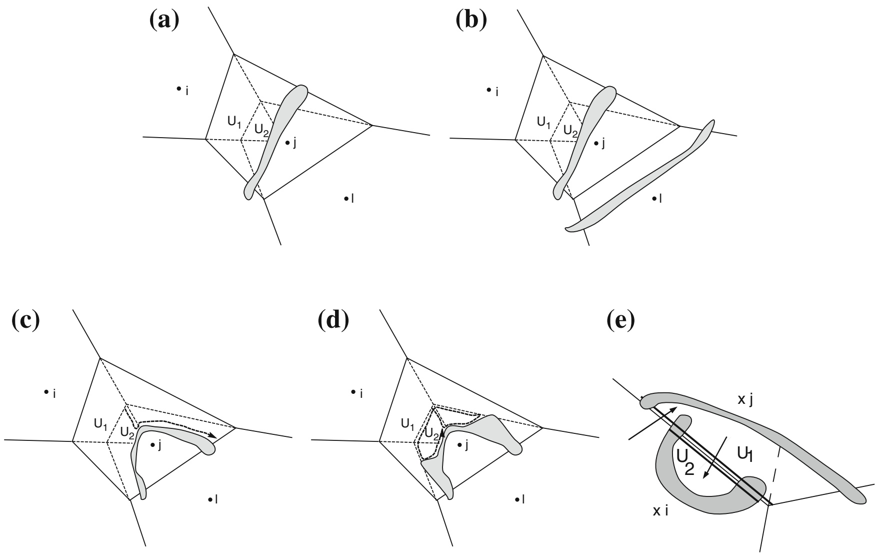

Fig. 5 In situations such as illustrated in **a** and **b** robot l will not enter any portion of V_j^l and hence robot i adds the region $U_2 \subset V_j^l$ to S_i. The robot explores a portion of V_j^l adjacent to U_1 and **c** reaches a point on A_{lj}^f, indicating that the robot l can reach and cover U_2, thus $U_2 \notin S_i$, **d** does not reach any point on A_{lj}^f, indicating that the robot l cannot reach U_2 and hence it is added to S_i. The exploration path is shown in *dark dashed line*. **e** Situations in which the robot i covers a part of $V_i \backslash V_i^{i_0}$

Scenario v. Beyond neighbors and neighbors of neighbors: Robot i continues looking for new patches and adds them to S_i based the following principle: Let $U \in S_i$, and W is adjacent to U. Now, W is added to S_i if either, W can not be reached by any other robot, or, if robot i is closer to W than any other robot.

The process continues until the robot finds no more adjacent patches to be added. At this point robot i performs area coverage of the current patch and returns to the previous patch. It finds if there are any new adjacent patches to be added; if not, it performs area coverage of this patch and goes back to the previous patch. Once robot i reaches the patch from which it was initially placed, $V_i^{i_0}$, it performs area coverage of that patch and stops.

4.1 Analytical Results

Lemma 1 *Let $V_{-i} \subset Q_{free} \in V_i \backslash V_i^{i_0}$ denote a region inaccessible to robot i. Then V_{-i} must be topologically connected to robot $j \in N(i)$'s Voronoi cell, V_j.*

Proof (*by contradiction.*) Assume that V_{-i} is not topologically connected to V_j. Suppose the only Voronoi boundary V_{-i} intersects is A_{ij}. There can be two cases of robot i's blocked boundary A^b_{ij} that resulted in V_{-i}:

Case 1. $A^b_{ij} = A^b_{ji}$. Since, $A_{ij} = A_{ji}$, this case implies $A^f_{ij} = A^f_{ji}$. Also, since the only Voronoi boundary V_{-i} intersect is A_{ij}, $V_{-i} \cap A_{ij} = A^b_{ij}$. Substituting this value of expression in the definition of free and blocked boundaries of A_{ji} and noting that $A^b_{ij} = A^b_{ji}$ and $A_{ij} = A_{ji}$, we get: $A^b_{ji} \cap A^f_{ji} = \{\emptyset\}$, or, $V_{-i} \cap A_{ij} \cap A^f_{ji} = \{\emptyset\}$, or, $V_{-i} \cap A_{ji} \cap A^f_{ji} = \{\emptyset\}$. But $A_{ji} \cap A^f_{ji} = A^f_{ji}$ (from the definition of A^f_{ji}). Therefore, we get, or, $V_{-i} \cap A^f_{ji} = \{\emptyset\}$. From the definition of a patch given in Sect. 4, a patch is bounded either by obstacles or by A_{ij}. Since V_{-i} is not accessible from V_i, it is bounded by obstacles from the side of V_i. And, $V_{-i} \cap A^f_{ji} = \{\emptyset\}$ implies it is not accessible (bounded by obstacles) from the side of V_j also. Since A_{ij} is the only Voronoi boundary intersecting V_{-i}, V_{-i} is bounded from all sides by obstacles. In other words, $V_{-i} \not\subset Q_{free}$, which contradicts our assumption.
Case 2. $A^b_{ij} \neq A^b_{ji}$. Suppose $A^b_{ji} \subset A^b_{ij}$.[4] Then the portion of A^b_{ij} that is not shared with A^b_{ji}, must be free (accessible) on the side of V_j; otherwise it would have been part of A^b_{ji}. The portion of boundary that is free only on side of V_j but not of side of V_i is $A^f_{ji} \backslash A^f_{ij}$. That is, $A^f_{ji} \backslash A^f_{ij} \subseteq A^b_{ij} \backslash A^b_{ji}$, or, $A^f_{ji} \backslash A^f_{ij} \subseteq A^b_{ij}$ (since $A^b_{ji} \subset A^b_{ij}$). This implies that the patch V_{-i} is topologically connected to V_j through A^f_{ji}, which contradicts our assumption that V_{-i} is not topologically connected to V_j. Hence proved.

Lemma 2 *A region $V_{-i} \subset Q_{free} \in V_i \backslash V_i^{i_0}$ that is inaccessible to robot i, must be topologically connected to V_j, $j \in I_N$.*

Proof The proof of Lemma 1 can be easily extended to a scenario where V_{-i} intersects more than one neighbor in $N(i)$ by considering the blocked boundary with each neighbor disjointly. For a more general case where V_{-i} is topologically connected only to $N^{(k)}(i)$, the kth hop Voronoi neighbor of i, $k > 1$ (scenarios iv. and v. in Sect. 4), the proof of Lemma 1 still holds between robots i and $j \in N^k(i)$. Varying k over 1 through the maximum hops between the farthest Voronoi cell from i, we get $N^k(i) = I_N$; hence proved.

Theorem 1 *The proposed distributed partitioning and coverage scheme ensures complete coverage of the free space.*

Proof By Lemmas 1 and 2, there must be a robot $j \in I_N$ whose Voronoi cell V_j is topologically connected to V_{-i}. This ensures that for every robot $i \in I_N$, the free space in its Voronoi cell V_i denoted by $Q_{free} \cap V_i$ gets covered by itself or by one or more robots in $j \in I_N$. The total region covered by all robots in I_N is then given

[4] A similar result can be proved for $A^b_{ij} \subset A^b_{ji}$ by interchanging indices i and j.

by $\cup_i (Q_{free} \cap V_i) = Q_{free} \cap (\cup_i V_i) = Q_{free} \cap Q$ (from definition of Voronoi cell) $= Q_{free}$ (since $Q_{free} \subseteq Q$). Hence proved.

Theorem 2 *The proposed distributed partitioning and coverage scheme achieves non-overlapping coverage.*

Proof The proof follows from the construction of patches using Voronoi cell boundaries. From definition, a patch between V_i and V_j is bounded either by obstacles or by the bisector line B_{ij} between robots i and j's initial positions $p_i(0)$ and $p_j(0)$. The Voronoi partitioning is done only once at the beginning, and by definition (Eq. 1) guarantees non-overlapping Voronoi cells. Since there is only one robot per Voronoi cell, the coverage of the initial Voronoi cell ($V_i^{i_0}$) is done only by robot i. When a region $V_{-i} \in V_i \backslash V_i^{i_0}$ is inaccessible from $V_i^{i_0}$, if V_{-i} is adjacent to only one other Voronoi cell V_j then only robot j covers V_{-i}. On the other hand, if V_{-i} is adjacent to more than one Voronoi cell $V_{j1}, V_{j2}, \ldots$ then each pair of robots j_a and j_b divide the region of V_{-i} into patches S_{j_a}, S_{j_b} by extending their bisector lines B_{j_a, j_b}. This construction ensures that $S_{j_a} \cap S_{j_b} = \{\emptyset\}$, or, patches S_{j_a}, S_{j_b} are non-overlapping; patch S_{j_k} is covered only by robot j_k. Therefore, for every robot i, $V_i^{i_0}$ and every inaccessible region V_{-i} is covered by exactly one robot. Hence proved.

Also, note that since the number of Voronoi cells is bounded by N (number of robots) and there is at least one Voronoi cell that is connected to any initially inaccessible region, therefore, the repartitioning technique takes at most N steps to find and connect the initially inaccessible region to another Voronoi cell. Consequently, the repartitioning mechanism is guaranteed to converge in a finite number of steps.

We have implemented the repartitioning algorithm using an auction protocol as shown in Algorithm 1. The robots use Voronoi partitioning to get their initial coverage regions corresponding to their Voronoi cells. Each robot then explores the boundary of its Voronoi cell. If, upon completing the exploration of its boundary, there are unexplored regions remaining in the Voronoi cell, these regions are allocated to neighboring robots using an auction protocol—robots in the neighboring Voronoi cells of the obstructed robot are sent a bid request message. Every neighbor robot calculates a bid for the region, and sends it to the auctioning robot. In the current implementation of the algorithm, these bids are calculated as the perimeter of the robot's current region. The robot that submits the lowest bid is selected as the winner of the auction and assigned the inaccessible portion of the Voronoi cell. The auctioning robot informs the winner, which then appends the region to the list of regions it needs to cover, and starts to perform boundary coverage of its newly assigned region. The auction algorithm possesses the essential properties (completion, non-overlapping coverage), but it reduces communication and coordination overhead by combining adjacent patches belonging to different robots, when the patches are accessible from each other.

Algorithm 1: Algorithm used by a robot to perform repartition coverage.

Repart-Coverage(V_i)
Input: V_i: Voronoi cell of robot i
Output: V_i': Repartitioned coverage region for robot
perform boundary coverage in V_i and determine $V_i^{i_0}$
$\mathbf{S}_{ij}^b$ ← set of blocked patches comprising $V_i \setminus V_i^i$
for *each* $S_{ij}^b \in \mathbf{S}_{ij}^b$ **do**
    j ← set of Voronoi neighbor robots of i that have Voronoi cell boundaries with S_{ij}^b
    send coordinates of polygon representing S_{ij}^b to all robot in **j**
    wait for bids
    bid ← set of bids received
    $j_{win} \leftarrow \arg\min_j \textbf{bid}$
    $V_i \leftarrow V_i \setminus S_{ij}^b$ //remove S_{ij}^b from V_i
    send message to robot j_{win} to add S_{ij}^b to $V_{j_{win}}$
handleBidMessages() //for robot j
if *received bid request for* S_{ij}^b *from robot i* **then**
    $bid_j = \begin{cases} \text{currently covered perimeter of } V_j, & \text{if} S_{ij}^b \text{ reachable} \\ \infty, & \text{otherwise} \end{cases}$
    send bid_j to robot i
if *received winner message for* S_{ij}^b *from robot i* **then**
    $V_j \leftarrow V_j \cup S_{ij}^b$ //add S_{ij}^b to V_j
    Repart-Coverage(V_j)

5 Experimental Results

We have implemented our proposed Repart-Coverage algorithm using simulated e-puck robots within the Webots simulator. E-puck robots use a ring of eight IR-based proximity sensors with a 4 cm range to avoid obstacles and follow obstacle boundaries. Robots use Bluetooth protocol for inter-robot communication, and have a GPS and compass for localizing w.r.t the environment. Figure 6a–d, show four different environments measuring 2×2 m^2 with different internal obstacles and with 5–7 robots, placed initially at arbitrary positions. These environments illustrate different scenarios where the Voronoi cell of one or more robots becomes partially inaccessible due to the obstacles in the environment, corresponding to the different scenarios discussed in Sect. 4. The red lines on the floor of the environment denote the Voronoi cells assigned to each robot. For reaching and following the boundary of its Voronoi cell, each robot uses a lightweight, bug-like algorithm called Egress [13] that enables a robot to start from any arbitrary internal point in its assigned region, find a path to the region's boundary using basic motions such as move-outward and wall-follow, and, completely explore the entire outermost boundary of the region. Each robot's initial location is at the center of its Voronoi cell; the path followed by the robot is marked with a dark red trail. Figure 6e–h show the scenarios for the different

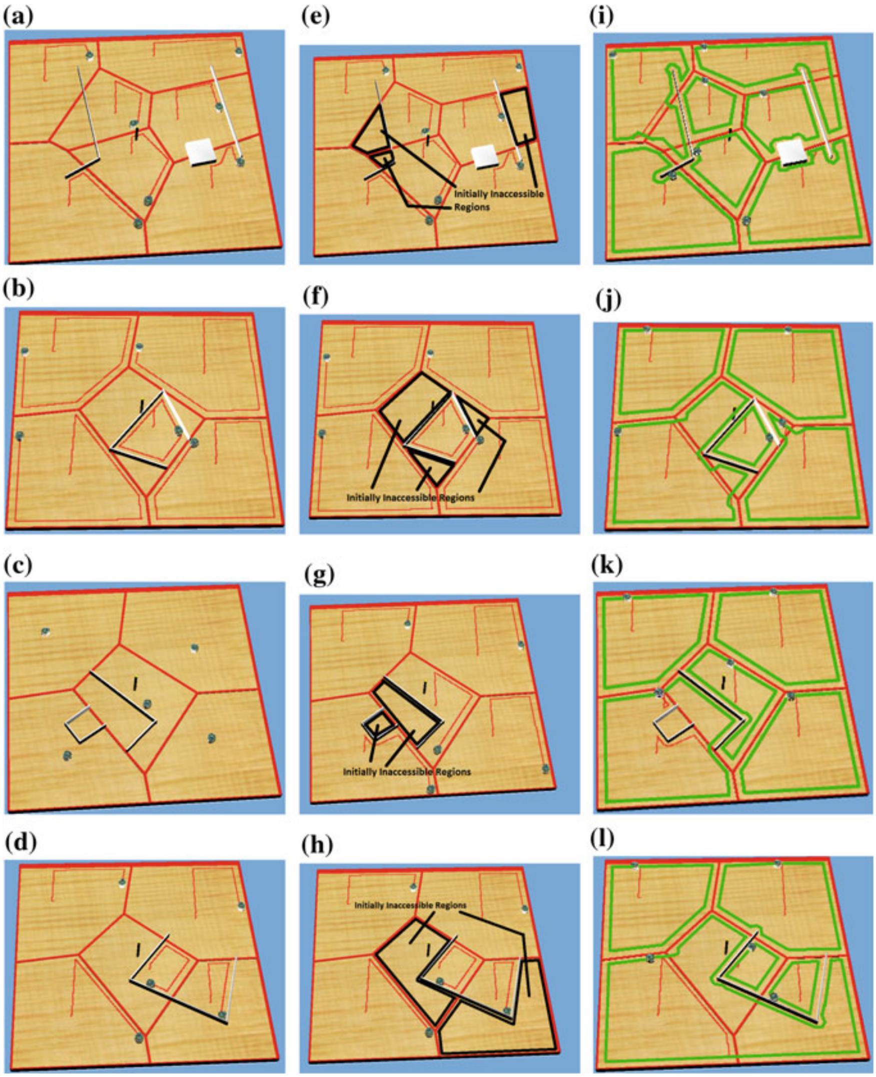

Fig. 6 Snapshots from Webots showing repartition coverage by 5–7 robots in different 2 × 2 m^2 environments with different obstacles. **a**–**d** initial Voronoi partition, **e**–**h** robots performing boundary coverage on original Voronoi cell, while showing inaccessible regions arising out of original Voronoi partition, **i**–**l** repartitioned cells and robots completing boundary coverage of entire environment; the final boundary of the cell that each robot covered is marked with a *green line*

environments at the end of boundary coverage along the Voronoi cell boundaries; the initially inacccesible regions of the respective Voronoi cells are marked with a black boundary. Finally, Fig. 6i–l show the result of our repartitioning algorithm. Robots from adjacent cells are allocated to cover each of the initially inaccessible regions

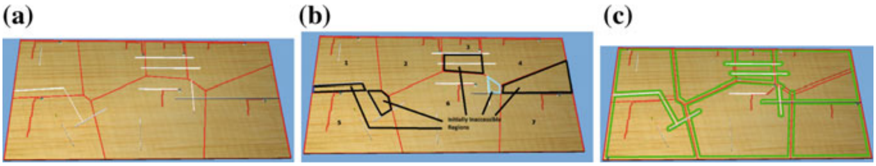

Fig. 7 Snapshots from Webots showing repartition coverage by 7 robots in a 3×6 m^2 environment with different obstacle features, **a** initial Voronoi partition, **b** robots performing boundary coverage on Voronoi cell, *black/light blue boundaries* show inaccesible regions. **c** repartitioned cells and robots completing coverage of entire environment

using the Repart-Coverage algorithm. The trail of the paths followed by the different robots shows that every region in the environment is covered by exactly one robot. This shows that our algorithm is successful to (re)-parititon the free space in the environment into complete, non-overlapping regions for coverage.

Figure 7a–c show another instance of the operation of the Repart-Coverage algorithm for a 3×6 m^2 environment with 7 robots. The scenario includes some unique obstacle features like narrow channels between obstacles and obstacles that span across multiple Voronoi cells, which require the inaccessible regions to be re-allocated to robots multiple times (similar to scenarios iv. and v. in Sect. 4). This shows that our algorithm successfully terminates and is able to find complete, non-overlapping regions even for complex obstacle geometries.

Finally, we have quantified the performance of our algorithm in terms of the area allocated to the different robots and the distances covered by them while performing boundary coverage. Table 1 shows the average area of the region allocated to each robot using our algorithm versus the area of the initial Voronoi cell for the different environments we have considered. Note that the initial Voronoi partition results in uncovered regions while the repartitioning guarantees complete coverage. The results for the different environments show that when obstacles result in larger inaccessible regions in the initial Voronoi cells, the coverage regions for each robot recalculated by the repartitioning algorithm have higher variance (std. dev, and max/min) than the initial Voronoi cells. This is because, with more complex obstacles, robots have to cover regions from other robots' initial Voronoi cells in addition to covering their own Voronoi cells.

6 Conclusions and Future Work

We proposed a novel technique for distributed spatial partitioning of an initially unknown region that guarantees a partitioning of the free space in the environment into a set of connected regions that can be covered by each robot. Currently, we are investigating techniques for each robot to dynamically build a map of the boundary of its currently allocated region instead of maintaining the end points of vertices of

Table 1 Area of cells to be covered before and after repartitioning for the different environments shown in Figs. 6a–d and 7a

Area of cell (m^2)	Env.9(a) before	Env.9(a) after	Env.9(b) before	Env.9(b) after	Env.9(c) before	Env.9 (c) after	Env.9(d) before	Env.9(d) after	Env.10(a) before	Env.10(a) after
Avg.	0.581	0.574	0.8	0.804	0.8	0.798	0.8	0.8	2.57	2.57
Stdev.	0.175	0.244	0.149	0.361	0.149	0.251	0.149	0.541	1.111	1.151
Min.	0.36	0.25	0.58	0.21	0.58	0.39	0.58	0.21	0.99	0.45
Max.	0.78	0.85	0.99	1.17	0.99	1.01	0.99	1.58	4.38	4.28
Uncov.	0.34	0	0.4	0	0.24	0	0.76	0	2.07	0

the boundary segments. The boundary map will enable a robot to efficiently plan its path to newly added regions instead of circumventing regions whose boundary it has already explored. Additionally, with boundary maps, the load (area covered) between different robots can be balanced by including factors such as the area of and distance to the newly allocated region, and, the area of the existing region in the robots' bids for new regions. Finally, we are implementing the proposed algorithm on physical robots.

Acknowledgments This work was partially supported by the U.S. Office of Naval Research as part of the COMRADES project.

References

1. Choset, H.: Coverage of known spaces: the boustrophedon cellular decomposition. Auton. Robots **9**, 247–253 (2000)
2. Hert, S., Lumelsky, V.: Polygon area decomposition for multiplerobot workspace division. Int. J. Comput. Geom. Appl. **8**, 437–466 (1998)
3. Rekleitis, I., New, A.P., Rankin, E.S., Choset, H.: Efficient boustrophedon multi-robot coverage: an algorithmic approach. Ann. Math Artif. Intell. **52**, 109–142 (2008)
4. Cortes, J., Martinez, S., Karata, T., Bullo, F.: Coverage control for mobile sensing networks. IEEE Trans. Rob. Auton. **20**(2), 243–255 (2004)
5. Bash, B.A., Desnoyers, P.J.: Exact distributed voronoi cell computation in sensor networks. In: Proceedings of the Sixth IEEE/ACM Conference On Information Processing in Sensor Networks, pp. 236–243 (2007)
6. Choset, H.: Coverage for robotics—a survey of recent results. Ann. Math. Artif. Intell. **31**, 113–126 (2001)
7. Altshuler, Y., Yanovski, V., Wagner, I.A., Bruckstein, A.M.: Multi-agent cooperative cleaning of expanding domains. I. J. Robot. Res. **30**(8), 1037–1071 (2011)
8. Agmon, N., Hazon, N., Kaminka, G.: The giving tree: constructing trees for efficient offline and online multi-robot coverage. Ann. Math Artif. Intell. **52**(2–4), 143–168 (2009)
9. Jäger, M., Nebel, B.: Dynamic decentralized area partitioning for cooperating cleaning robots. In: Proceedings of IEEE International Conference on Robotics and Automation, pp. 3577–3582 (2002)
10. Schwager, M., Rus, D., Slotine, J.-J.E.: Decentralized, adaptive coverage control for networked robots. I. J. Robot. Res. **28**(3), 357–375 (2009)
11. Breitenmoser, A., Schwager, M., Metzger, J.-C., Siegwart, R., Rus, D.: Voronoi coverage of non-convex environments with a group of networked robots. In: ICRA, pp. 4982–4989 (2010)
12. Durham, J.W., Carli, R., Frasca, P., Bullo, F.: Discrete partitioning and coverage control for gossiping robots. IEEE Trans. Robot. **28**(2), 364–378 (2012)
13. Guruprasad, K.R., Dasgupta, P.: Egress: an online path planning algorithm for boundary exploration. In: IEEE International Conference on Robotics and Automation, May 2012, pp. 3991–3996
14. Gabriely, Y., Rimon, E.: Spanning-tree based coverage of continuous areas by a mobile robot. Ann. Math Artif. Intell. **31**, 77–98 (2001)
15. Batalin, M., Sukhatme, G.S.: Spreading out: a local approach to multi-robot coverage. In: Proceedings of the International Symposium on Distributed Autonomous Robotic Systems, Fukuoka, Japan, Jun 2002, pp. 373–382. [Online]. http://robotics.usc.edu/publications/56/