

Distributed Coverage and Exploration in Unknown Non-convex Environments

Subhrajit Bhattacharya, Nathan Michael, and Vijay Kumar

Abstract. We consider the problem of multi-robot exploration and coverage in unknown non-convex environments. The contributions of the work include (1) the presentation of a distributed algorithm that computes the *generalized Voronoi tessellation* of non-convex environments (using a discrete representation) in real-time for use in feedback control laws; and (2) the extension of this method to entropy-based metrics that allow for cooperative coverage control in unknown non-convex environments. Simulation results demonstrate the application of the control methodology for cooperative exploration and coverage in an office environment.

1 Introduction

We are interested in considering the following scenario: a team of robots enter an unknown and non-convex environment. The robots must control to explore the environment for map construction and converge to a formation in the map that disperses the robots to locations that permit them to continue to engage in activities such as persistent surveillance. This description lends itself to a broad class of robotics applications. In this work, we focus on an essential component toward this scenario: the development of decentralized individual robot control laws based on uncertain estimates of the environment that drive the team of robots to explore and cover the environment. The contributions of this work are:

1. the presentation of a distributed algorithm that computes the *generalized Voronoi tessellation* of non-convex environments (using a discrete representation) in real-time for use in feedback control laws; and

Subhrajit Bhattacharya · Nathan Michael · Vijay Kumar
GRASP Laboratory,
University of Pennsylvania,
Philadelphia, PA 19104, USA
e-mail: {subhrabh, nmichael, kumar}@grasp.upenn.edu

2. the extension of this method to entropy-based metrics that allow for cooperative coverage control in unknown non-convex environments.

Therefore, to limit the scope of the presentation, we assume the robots are able to (A) localize and build maps in a common frame; and (B) communicate.

The presentation begins by motivating the development of a geodesic Voronoi tessellation in non-convex environments. We detail a search-based algorithm for computing an equivalent tessellation in discrete environments (Sect. 4). We propose a method for computing the centroid of the Voronoi cells given a tessellation of a discrete environment, permitting the application of centroid-based robot control laws for cooperative coverage. In Sect. 5, uncertainty in the environment description is introduced through the development of an entropy-based metric; enabling the computation of the instantaneous geodesic Voronoi tessellation given an uncertain environment. We comment on the coverage and convergence guarantees resulting from this approach and present results that demonstrate through simulation its application in an office environment.

2 Related Literature

This work is most closely related to methods proposed in the cooperative exploration literature and coverage control literature. A common approach toward exploration is frontier-based exploration where control directions seek to minimize entropy or uncertainty in the robot pose or map [16]. Coordination for multi-robot exploration is generally accomplished through explicit coordination designed to reduce redundant exploration or implicit coordination that occurs when robots communicate and coordinate (e.g. share maps) when in close proximity but without considering other robots' history or future plans.

In [15], the authors propose an exploration strategy with feedback control laws that maximize information gain by considering uncertainty in both the robot pose and map. A key contribution of this work (and similar recent works) is the relaxation of the assumption of robot localization. It is for this reason that we believe the first assumption in the prior section is reasonable. A multi-robot exploration strategy is presented in [2, 14], where the robots coordinate to determine targets best served by each robot that maximize the information gain for the team of robots. The authors quantify the performance gain due to explicit coordination and increasing numbers of robots. In [7], experimental results are presented that demonstrate the use of a team of robots to address a scenario similar to the one described at the beginning of this work. The authors do not consider explicit coordination between robots and note that this results in redundant exploration. A similar multi-robot exploration study is presented in [6], where robots explore an indoor office environment while simultaneously localizing and mapping. Coordination is implicit as the robots exchange and merge maps when in close proximity.

A common coverage control approach is through the definition of feedback control laws defined with respect to the centroids of Voronoi cells resulting from the

Voronoi tessellation of an environment. In [4], the authors propose gradient descent-based individual robot control laws that guarantee optimal coverage of a convex environment given a density function which represents the desired coverage distribution. The authors of [12] build upon this idea and develop decentralized control laws that position a mobile sensor network optimally with respect to a known probability distribution. In [13], this approach is extended to consider near-optimal controllers that do not require prior knowledge of a desired coverage distribution. To address the limitation of requiring a convex environment, the authors of [10] propose the use of *geodesic Voronoi tessellations* determined by the geodesic distance rather than the Euclidean distance. An approach that considers both exploration and coverage using Voronoi tessellations is presented in [17]. However, this method differs greatly from ours and assumes that the boundary of the environment is known. The Voronoi tessellation is computed based on this assumed convex polygon and the robots control to the centroids of this tessellation and explore en route.

The primary point of differentiation between our work and existing methods is due to the fact that we are able to rapidly compute the Voronoi tessellation of non-convex environments in a distributed manner. The consequence of this result is that we can use the instantaneous tessellation to compute decentralized robot control laws. By considering exploration and coverage simultaneously, we enforce explicit coordination between the robots and yield optimal solutions.

3 Background

Let $\Omega \subset \mathbb{R}^N$ be a simply connected (in general non-convex) subset of \mathbb{R}^N that represents the environment. There are n mobile robots in the environment with on-board range sensors, and in particular the position of the i^{th} robot is represented by $\mathbf{p}_i \in \Omega$ and the tessellation associated with it by $W_i, \forall i = 1, 2, \dots, n$. By definition, the tessellations are such that $I(W_i) \cap I(W_j) = \emptyset, \forall i \neq j$, where $I(\cdot)$ denotes the interior of a set, and $\cup_{i=1}^n W_i = \Omega$. For a given set of robot positions $P = \{\mathbf{p}_1, \mathbf{p}_2, \dots, \mathbf{p}_n\}$ and tessellations $W = \{W_1, W_2, \dots, W_n\}$ such that $\mathbf{p}_i \in W_i, \forall i = 1, 2, \dots, n$, the *coverage functional* is defined as:

$$\mathcal{H}(P, W) = \sum_{i=1}^n \mathcal{H}(\mathbf{p}_i, W_i) = \sum_{i=1}^n \int_{W_i} f(d(\mathbf{q}, \mathbf{p}_i)) \phi(\mathbf{q}) d\mathbf{q},$$

where $d(\cdot, \cdot)$ is a distance function defined on Ω , $f : \mathbb{R} \rightarrow \mathbb{R}$ is a smooth and strictly increasing function in the range of d , $\phi : \Omega \rightarrow \mathbb{R}$ is a weight or density function, and $d\mathbf{q}$ represents an infinitesimal area or volume element. Throughout this paper we choose $f(x) = x^2$.

Lloyd's algorithm [8] and its continuous-time asynchronous implementations [4] are distributed algorithms for minimizing $\mathcal{H}(P, W)$ with guarantees on completeness and asymptotic convergence to a local optimum when Ω is convex and in an Euclidean distance setting (i.e. $d(\mathbf{p}, \mathbf{q}) = \|\mathbf{p} - \mathbf{q}\|_2$).

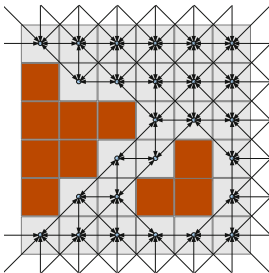


Fig. 1 An 8-connected grid graph created from a uniformly discretized environment

4 Coverage without Uncertainty

4.1 Geodesic Voronoi Tessellation

An extension of the continuous-time Lloyd's Algorithm algorithm to non-convex environments is presented in [10], where the distance function is defined as the *geodesic distance* in Ω . Consequently, the *Voronoi tessellations* under consideration are the *geodesic Voronoi tessellations*. That is, for a given P ,

$$V(P) = \underset{W}{\operatorname{argmin}} \mathcal{H}(P, W) \iff V_i(P) = \{\mathbf{q} \in \Omega \mid d(\mathbf{q}, \mathbf{p}_i) \leq d(\mathbf{q}, \mathbf{p}_j), \forall j \neq i\}. \quad (1)$$

Geometric methods for computing such geodesic Voronoi tessellations in non-convex polygonal environments are detailed in [10, 1]. These methods suffer from the inherent drawback of high complexity in modeling cluttered real environments with noise and small obstacles. Moreover, in this work we wish to modify the metric such that it is non-uniform in Ω , and hence $d(\mathbf{q}, \mathbf{p})$ is no longer the Euclidean length of the shortest path between \mathbf{q} and \mathbf{p} lying in Ω .

4.2 A Search-Based Algorithm for Finding Geodesic Voronoi Tessellations

We propose a search-based algorithm for finding the geodesic Voronoi tessellations. We begin by uniformly discretizing Ω and creating a graph such that each node or vertex of the graph corresponds to a cell of the discretization with connections to admissible neighbors (see Fig. 1). This graph, \mathcal{G}_Ω , consists of a vertex set, $\mathcal{V}(\mathcal{G}_\Omega)$, and edge set, $\mathcal{E}(\mathcal{G}_\Omega)$. For a vertex $\mathbf{q} \in \mathcal{V}(\mathcal{G}_\Omega)$, we use the same notation \mathbf{q} to denote the coordinate of the vertex in the configuration space of the agents. $\mathcal{N}(\mathbf{q}) := \{\mathbf{s} \in \mathcal{V}(\mathcal{G}_\Omega) \mid \overline{\mathbf{s}\mathbf{q}} \in \mathcal{E}(\mathcal{G}_\Omega)\}$ denotes the set of neighboring vertices of \mathbf{q} . The vertices joined by an edge $\varepsilon \in \mathcal{E}(\mathcal{G}_\Omega)$ are denoted by $\mathbf{v}_s(\varepsilon)$ and $\mathbf{v}_t(\varepsilon)$. We associate a cost, $c(\varepsilon)$, with every edge $\varepsilon = \overline{\mathbf{v}_s(\varepsilon)\mathbf{v}_t(\varepsilon)} \in \mathcal{E}(\mathcal{G}_\Omega)$. In particular, for elementary geodesic Voronoi tessellations, the cost of an edge is its Euclidean length (i.e. $c(\varepsilon) = \|\mathbf{v}_s(\varepsilon) - \mathbf{v}_t(\varepsilon)\|_2$).

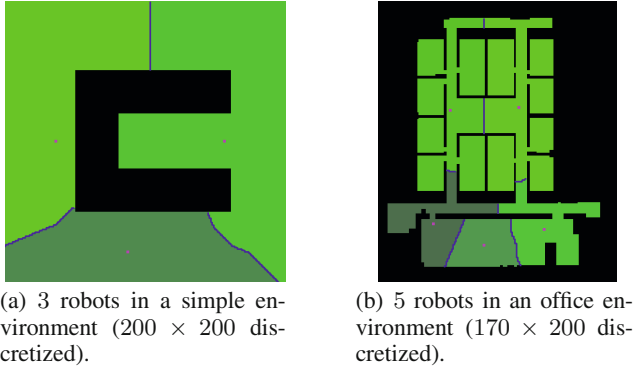


Fig. 2 The geodesic Voronoi tessellation of non-convex workspaces created using a uniformly discretized 8-connected grid-world. The robot locations are marked by enlarged magenta pixels.

For each of the robots we perform a Dijkstra's search [5, 3] in \mathcal{G}_Ω starting from the vertex where the robot itself is located, \mathbf{p}_i , and expand all the vertices in \mathcal{G}_Ω . Thus, at the end of the expansions, for each vertex $\mathbf{q} \in \mathcal{V}(\mathcal{G}_\Omega)$ we obtain the values of $g_i(\mathbf{q})$, $\forall i = 1, 2, \dots, n$, such that $g_i(\mathbf{q})$ gives the cost of the shortest path between \mathbf{p}_i and \mathbf{q} in \mathcal{G}_Ω . The geodesic Voronoi tessellation is created by assigning the cells or vertices to the robot which has the least g -value at that node. That is, $V_i = \{\mathbf{q} \in \mathcal{V}(\mathcal{G}_\Omega) \mid g_i(\mathbf{q}) \leq g_j(\mathbf{q}), \forall j \neq i\}$ and $\mathbf{q} \in V_i \iff g_i(\mathbf{q}) \leq g_j(\mathbf{q}), \forall j \neq i$. We use the same notation V_i to denote the sub-set of vertices in $\mathcal{V}(\mathcal{G}_\Omega)$ that belong to the Voronoi tessellation V_i .

For the distributed architecture, each robot maintains its own copy of \mathcal{G}_Ω , updating it (for probability of occupancy) using its own sensor readings as well as information acquired from its neighboring robots about parts of their copies of \mathcal{G}_Ω (described later).

Note that the least cost path between two points and hence the Voronoi tessellation, depends on the discretization of the environment and the definition of connectivity between neighboring vertices.

Figure 2 depicts geodesic Voronoi tessellations created using this algorithm. The boundary between two adjacent tessellations is such that the costs of the least cost paths from any cell on the boundary to either of the two robots that share the boundary are equal.

4.3 Continuous-Time Lloyd's Algorithm for Discrete Non-Convex Environments

The continuous-time Lloyd's algorithm for non-convex environments requires that each mobile robot follows the gradient of $\mathcal{H}(P) := \mathcal{H}(P, V(P))$ given by the formula [10]:

$$\frac{\partial \mathcal{H}}{\partial \mathbf{p}_i} = \int_{V_i(P)} \frac{\partial}{\partial \mathbf{p}_i} f(d(\mathbf{q}, \mathbf{p}_i)) \phi(\mathbf{q}) d\mathbf{q}. \quad (2)$$

In a discretized environment, finding the gradient in (2) approximately reduces to searching among the neighboring vertices of the robot's current location such that $\mathcal{H}(P)$ is minimized for the new robot position in the next time-step. That is, we seek to find

$$\begin{aligned} \mathbf{p}_i^{t+1} &= \operatorname{argmin}_{\mathbf{p} \in \mathcal{N}(\mathbf{p}_i^t)} \int_{V_i(P^t)} f(d(\mathbf{q}, \mathbf{p})) \phi(\mathbf{q}) d\mathbf{q} \\ &\approx \operatorname{argmin}_{\mathbf{p} \in \mathcal{N}(\mathbf{p}_i^t)} \sum_{\mathbf{q} \in V_i(P^t)} f(d(\mathbf{q}, \mathbf{p})) \phi(\mathbf{q}) \Delta \mathbf{q}, \end{aligned} \quad (3)$$

where the superscripts denote the time-step, the summation is over all of the nodes in $\mathcal{V}(\mathcal{G}_\Omega)$ that are inside $V_i(P^t)$, and $\Delta \mathbf{q}$ is the area of the discretization cell at \mathbf{q} . Thus, the control law for each mobile robot reduces to driving from the current positions \mathbf{p}_i^t to \mathbf{p}_i^{t+1} as prescribed by (3). In order to find \mathbf{p}_i^{t+1} from (3) for the i^{th} robot, at time instant t we perform Dijkstra's search and expand the states in $V_i(P^t)$ starting for each of the states in $\mathcal{N}(\mathbf{p}_i^t)$. This gives us the values for $d(\mathbf{q}, \mathbf{p}), \forall \mathbf{p} \in \mathcal{N}(\mathbf{p}_i^t), \mathbf{q} \in V_i(P^t)$; with \mathbf{p}_i^{t+1} computed directly from (3) by computing and comparing the summations for each $\mathbf{p} \in \mathcal{N}(\mathbf{p}_i^t)$.

Figures 3(a)-3(d) show the evolution of the geodesic Voronoi tessellations and the trajectories followed by the mobile robots upon following the above control algorithm. In this example, we set $\phi(\mathbf{q}) = 1$. The environment is a 170×200 uniformly discretized 8-connected grid-world. Starting from the shown configuration convergence is achieved in less than 150 iterations. Running on a single Pentium processor (2.1 GHz, 4 GB RAM), each iteration takes on average 0.1 s (including computation of the current tessellations and the desired positions for the next time-step for all robots).

4.3.1 Projection of Centroid Method

In the previous section we do not discuss how to find the centroid, \mathbf{C}_{V_i} , of the geodesic Voronoi tessellation, V_i . In general, the direct computation of the *generalized centroid*,

$$\mathbf{C}_{V_i}^{gen} = \operatorname{argmin}_{\mathbf{p}_i \in V_i} \int_{V_i} f(d(\mathbf{q}, \mathbf{p}_i)) \phi(\mathbf{q}) d\mathbf{q} \quad (4)$$

in a non-convex environment is difficult [10]. However, a coverage control law over Voronoi tessellations such as that proposed in [4]:

$$\mathbf{u}_i = k(\mathbf{C}_{V_i} - \mathbf{p}_i), \quad (5)$$

requires knowledge of a centroid for the tessellations. Moreover, for exploration, we desire to implement the standard Lloyd's Algorithm or a semi-continuous version of it, which invariably require the computation of a *centroid*. In order to find an analog

of a centroid for a non-convex tessellation, we project the geometric centroid inside the tessellation. We compute

$$\mathbf{C}_{V_i} = \frac{\int_{V_i} \mathbf{q} \phi(\mathbf{q}) d\mathbf{q}}{\int_{V_i} \phi(\mathbf{q}) d\mathbf{q}}, \quad (6)$$

and if \mathbf{C}_{V_i} lies outside V_i , find the point in V_i closest to it:

$$\underline{\mathbf{C}}_{V_i} = \operatorname{argmin}_{\mathbf{q} \in V_i} \|\mathbf{q} - \mathbf{C}_{V_i}\|. \quad (7)$$

This is an approximate method. In order to account for the non-uniformity of ϕ inside V_i , we may compensate by projecting the centroid in a *high ϕ region* in V_i . We modify (7) as,

$$\overline{\mathbf{C}}_{V_i} = \begin{cases} \operatorname{argmin}_{\mathbf{q} \in V_i, \phi(\mathbf{q}) \geq \tau} \|\mathbf{q} - \mathbf{C}_{V_i}\| & \text{if it exists} \\ \operatorname{argmin}_{\mathbf{q} \in V_i} \|\mathbf{q} - \mathbf{C}_{V_i}\| & \text{otherwise.} \end{cases} \quad (8)$$

for some threshold τ , and use this projection.

The control law for the i^{th} robot given a discrete formulation reduces to taking one step towards $\overline{\mathbf{C}}_{V_i(P^t)}$ along the shortest path joining \mathbf{p}_i^t and $\overline{\mathbf{C}}_{V_i(P^t)}$ (found via a single Dijkstra's search in V_i). This approach requires less computation than the gradient search method of (3). Further, simulation results suggest that the *projection of centroid* control method always converges and in cluttered environments differs little from the results obtained by the gradient search method. From these observations, we formulate the following conjecture.

Conjecture 1 (Convergence of Projection of Centroid Method). If $\phi(\mathbf{q}) = k < \tau$ is uniform (constant) for all \mathbf{q} , then there exists robot positions $P^* = \{\mathbf{p}_1^*, \dots, \mathbf{p}_n^*\}$ such that $\overline{\mathbf{C}}_{V_i(P^*)} = \mathbf{p}_i^*$, *i.e.* an equilibrium point, and the Projection of Centroid control method drives the robots to such a configuration.

Proof for a special case. We present a partial proof for a certain type of V_i . For Euclidean metric, if the $\overline{\mathbf{C}}_{V_i} = \mathbf{C}_{V_i}^{\text{gen}}$, this conjecture becomes a theorem and the control law described above is guaranteed to converge [10]. We thus investigate the cases where $\overline{\mathbf{C}}_{V_i}$ indeed is the *generalized centroid*. Clearly, $\mathbf{C}_{V_i} = \mathbf{C}_{V_i}^{\text{gen}} \Rightarrow \overline{\mathbf{C}}_{V_i} = \mathbf{C}_{V_i}^{\text{gen}}$ (since $\mathbf{C}_{V_i}^{\text{gen}}$ always lies inside V_i). The condition under which $\mathbf{C}_{V_i} = \mathbf{C}_{V_i}^{\text{gen}}$ is that $d(\mathbf{q}, \mathbf{C}_{V_i}) = \|\mathbf{q} - \mathbf{C}_{V_i}\| \forall \mathbf{q} \in V_i$. This condition is equivalent to saying that V_i be *star-shaped* [11] with respect to \mathbf{C}_{V_i} . A trivial case of this condition is when V_i is convex, when the algorithm becomes equivalent to the continuous-time Lloyd's Algorithm [4].

For comparison, Figs. 3(e)–3(h) show the evolution of the geodesic Voronoi tessellations and the trajectories followed by the mobile robots using this control method. Once again, we use $\phi(\mathbf{q}) = 1$, and convergence takes place in less than 150 iterations. However in this case the computation time per iteration is on an average 0.03 s.

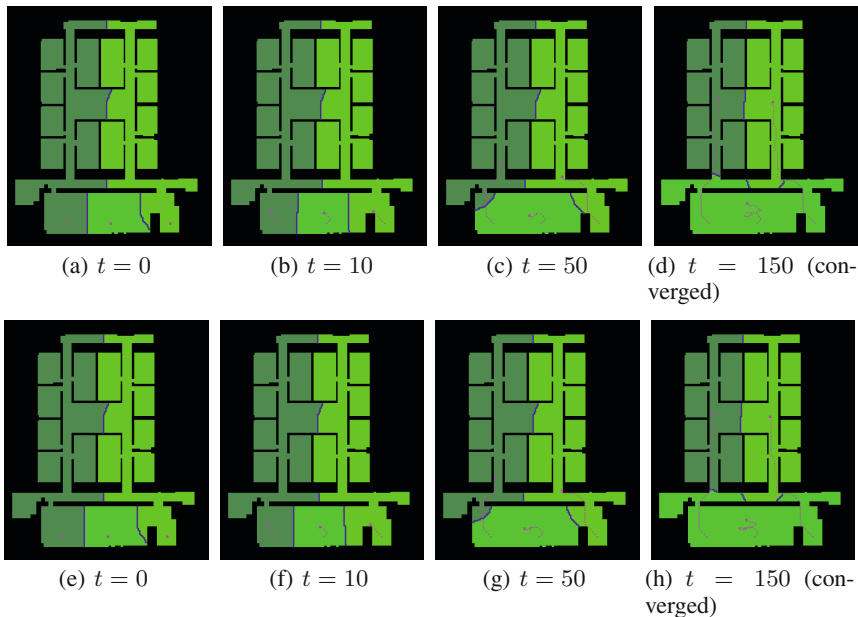


Fig. 3 Continuous-time Lloyd's algorithm in a discretized setting for optimal coverage. Figures 3(a)-3(d) use the gradient search method while Figures 3(e)-3(h) use the projection of centroid method.

5 Simultaneous Exploration and Coverage of Unknown or Partially-Known Environments

In this section, we consider the problem of deploying n mobile robots in an unknown or partially known environment, which upon collaborative exploration of the environment, will converge to an optimal or near-optimal coverage.

5.1 Entropy as Density Function

In order to address this problem each mobile robot maintains and communicates a probability map for the discretized environment such that $p(\mathbf{q})$ is the probability that the vertex \mathbf{q} is inaccessible (*i.e.* occupied or represents an obstacle), for all $\mathbf{q} \in \mathcal{V}(\mathcal{G}_\Omega)$. A threshold on the value of probability determines whether a particular node in $\mathcal{V}(\mathcal{G}_\Omega)$ is occupied/inaccessible for computation of the Geodesic Voronoi tessellations as well as control. Moreover the *Shannon entropy* for each cell can be computed as follows,

$$e(\mathbf{q}) = p(\mathbf{q}) \ln(p(\mathbf{q})) + (1 - p(\mathbf{q})) \ln(1 - p(\mathbf{q})). \quad (9)$$

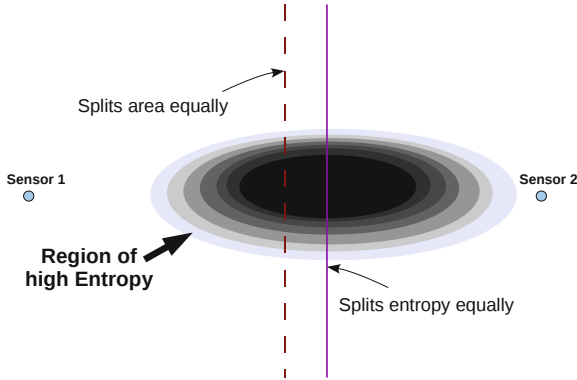


Fig. 4 Entropy-weighted Voronoi tessellation

This gives us an *Entropy map*, i.e. a value of entropy associated with each vertex \mathbf{q} - a map that represents uncertainty or the need to gather information within the environment. The Shannon entropy is such that it assumes high values for vertices for which the uncertainty is high (i.e. probability is close to 0.5), whereas it is low for known or visited vertices. Thus, we identify the weight or density function $\phi(\cdot)$ with the entropy $e(\cdot)$. This, by the construction of the control laws described before, will drive the mobile robots towards regions of high entropy within the robot's own tessellation, hence resulting in exploration of the environment.

For exploration of an unknown environment it is desired that we follow an analog of the traditional Lloyd's algorithm, where each robot visits completely a computed projected centroid of a tessellation at an earlier time-step (we call it a *target*), hence exploring the region, and subsequently recompute the next target, which is the projected centroid of the current tessellation.

5.2 Entropy-Based Metric

The geodesic Voronoi tessellation performed according to (1) ensures that the boundaries of the tessellations “bisect” the area lying between the robots. While the metric d for this can be the geodesic distance in the case of the coverage problem, for exploration and for environments with uncertainty the tessellation boundaries need to be such that they “bisect” the uncertainty (or entropy) among the adjacent robots for cooperative exploration. This notion is illustrated in Fig. 4, where a high entropy region is placed asymmetrically between two robots in a convex environment without obstacles. The dashed line shows the boundary of a Voronoi tessellation created using the standard distance metric. However, one mobile robot has a larger unexplored region than the other. An alternate division is depicted with a solid line that splits the unexplored region equally.

We now redefine $d(\cdot, \cdot)$ to accommodate uncertainty in the tessellation. Let $\Gamma(\mathbf{p}, \mathbf{q})$ represent the set of all paths in Ω connecting \mathbf{p} and \mathbf{q} . Then the original definition of the geodesic distance is written as $d(\mathbf{p}, \mathbf{q}) = \min_{\gamma \in \Gamma(\mathbf{p}, \mathbf{q})} \int_{\gamma} dl$, where dl represents an elemental length along γ . We modify the definition as follows:

$$d(\mathbf{p}, \mathbf{q}) = \min_{\gamma \in \Gamma(\mathbf{p}, \mathbf{q})} \int_{\gamma} e(\mathbf{r}) dl,$$

where \mathbf{r} is a point on γ .

In terms of finding the Voronoi Tessellations by performing Dijkstra's searches in \mathcal{G}_{Ω} as described earlier, the only required change is to weigh the edges of the graph \mathcal{G}_{Ω} by entropy in those regions instead of the Euclidean length of the edges. In particular, we now define the cost of an edge ε as

$$c(\varepsilon) = \frac{e(\mathbf{v}_s(\varepsilon)) + e(\mathbf{v}_t(\varepsilon))}{2} + \eta \|\mathbf{v}_s(\varepsilon) - \mathbf{v}_t(\varepsilon)\|_2$$

where we add the second term with a very small value of η in order to compensate for noise in near-zero values of entropy and to make sure that the cost of an edge doesn't vanish. Performing Dijkstra searches in this weighted graph and creating tessellations using the same procedure as before will split the unexplored regions between two neighboring robots equally.

5.3 Time Dependence of Entropy, Coverage, and Convergence

We now detail how the probability map is updated based on the sensor readings. For the discussion that follows, $p^t(\mathbf{q})$ represents the estimated probability of occupancy of \mathbf{q} at the t^{th} iteration based on all measurements.

5.3.1 Inter-robot Communication

As discussed earlier, for the distributed architecture, each robot maintains its own copy of probability and entropy maps. Each updates its own maps based on readings from its own on-board sensor as well as information acquired from its neighboring robots about parts of their copies of their probability maps. A sensor fusion model (described in next section) is used to aggregate the data. For communicating its own probability map to other robots, each robot broadcasts the *new* information acquired by its own sensor over a *time window* or *phase*. Essentially the broadcasting of probability maps by each robot is done in *phases*. During a phase, a robot broadcasts a constant message (part of its own probability map) with a fixed timestamp over and over (repeatedly). This is to make sure that other robots receive this message. The robot also broadcasts its unique identity along with the message. Also, instead of broadcasting the whole probability map in each phase, each robot broadcasts only whatever new it has sensed during the previous broadcast phase. Thus the broadcasted information actually comprises of a small window in the whole probability

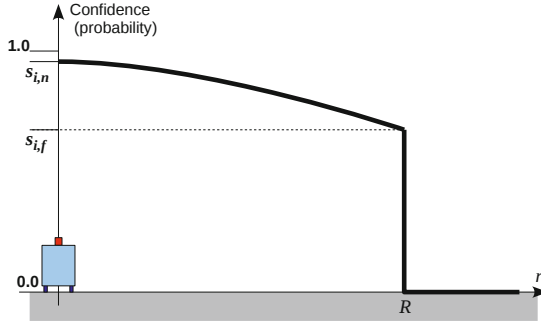


Fig. 5 The sensor model

map as well as in time, inside which the probability readings have changed. This makes each broadcast messages rather small. Essentially each robot maintains two buffers: The *current sensing buffer*, and the *broadcast buffer*. New readings from a robot's own laser sensor are added to the current sensing buffer, while things in the broadcast buffer are broadcasted. At the end of a broadcast *phase* the content of the broadcast buffer is pushed into the main probability map maintained by the robot, the content of the current sensing buffer is copied into the broadcast buffer, and the current sensing buffer is cleared for new sensor data. The information received from other robots about their map are directly added to the main probability map. This *differential* approach of communication significantly reduces the communication overhead required for sharing map data.

5.3.2 Sensor Model

We use a sensor model for each robot, $s_i(r)$, which gives the probability that the i^{th} robot's sensor measures the state of a grid cell located at a distance r from it correctly. In particular, in our simulations we use,

$$s_i(r) = \begin{cases} s_{i,n} + \frac{r^2}{R_i^2}(s_{i,f} - s_{i,n}) & \text{if } r \leq R_i \\ 0 & \text{otherwise,} \end{cases}$$

where R_i is the sensor range, and $0 \leq s_{i,f} \leq s_{i,n} \leq 1$ gives the far and near values of the confidence of the sensor.

Thus, if at time-step t the sensor of the i^{th} robot receives a measurement $z_i^t(\mathbf{q})$ (which is 1 for occupied, and 0 for unoccupied) for the cell \mathbf{q} , the probability that the cell is occupied based only on this measurement is given by $u_i^t(\mathbf{q}) = z_i^t(\mathbf{q}) s_i(\|\mathbf{q} - \mathbf{p}_i^t\|) + (1 - z_i^t(\mathbf{q}))(1 - s_i(\|\mathbf{q} - \mathbf{p}_i^t\|))$. We use a sensor fusion model to compute the net probability of occupancy for the cells based on the individual measured probabilities. In particular, one can compute $p^t(\mathbf{q}) = g^{-1} \left(\frac{\sum_{i,t'} g(u_i^{t'}(\mathbf{q}))}{\sum_{i,t'} 1} \right)$, where g is a strictly increasing function in $[0, 1]$, and the summations are taken over all the measurements by all sensors over all time instants [16]. For our experiments we

choose $g(x) = x^m$, $m > 0$. We note that by choosing $m \rightarrow \infty$, the value of $p^t(\mathbf{q})$ essentially becomes the *supremum* of all the measurements for \mathbf{q} . Alternatively, choosing $g(\cdot) = \log(\cdot)$ gives the geometric mean of the measurements, which has also been used in [16].

In order to compensate for the sensor noise the entropy map is smoothed by passing it through a min-filter. The smoothed entropy map is consequently used for computing the density function.

5.3.3 Time-Varying Density Function

A consequence of updating the probability map is that the entropies, and hence the weight function, ϕ , becomes a function of time:

$$\phi(\mathbf{q}, t) = e(\mathbf{q}, t) = p^t(\mathbf{q}) \ln(p^t(\mathbf{q})) + (1 - p^t(\mathbf{q})) \ln(1 - p^t(\mathbf{q})).$$

Conjecture 2 (Exploration and Convergence Guarantee). Assuming Conjecture 1 is true and the individual robot motion at each time step is determined by the Projection of Centroid Method (8), there exists a τ' , $0 < \tau' \leq \tau$ such that choosing $\phi(\mathbf{q}, t) = \max(\tau', e(\mathbf{q}, t))$ ensures complete exploration of the environment and convergence of the algorithm.

Proof. We assume there exists an ϵ radius around each mobile robot such that it is able to sense the occupancy and reduce the entropy of the cells within this radius below the value of τ' in a permanent manner. Due to the choice of our control method, each mobile robot drives closer to $\overline{\mathbf{C}}_{V_i}$ at every time-step. However, as long as there exists at least one cell $\mathbf{q} \in \mathcal{V}(\mathcal{G}_\Omega)$ such that $e(\mathbf{q}, t) \geq \tau$, a robot will drive to that cell where $\phi(\mathbf{q}, t) = \max(\tau', e(\mathbf{q}, t)) \geq \tau$ (due to equation (8) and since $\tau' \leq \tau$), and subsequently reduce the entropy of the region around \mathbf{q} below τ' . This process continues until the entropy of all the cells in the map goes below τ such that $\phi(\mathbf{q}, t_{covered}) = \max(\tau', e(\mathbf{q}, t_{covered})) \in [\tau', \tau)$ for all \mathbf{q} . This guarantees exploration of the environment with all cells having final entropy less than τ . Note that while τ' is sensor specific, τ is a design variable. Thus, if we choose $\tau = \tau'$, the density function becomes $\phi(\mathbf{q}, t_{covered}) = \tau' = \tau$, which is constant and independent of time throughout the environment. Consequently by Conjecture 1 convergence is achieved at some $t_{converged} \geq t_{covered}$.

5.3.4 The Overall Algorithm

So far we have described the various components of the algorithm. To put those in perspective, the steps below are what goes on at a higher level on each robot in sequence while exploring and covering an unknown or partially known environment in a distributed fashion.

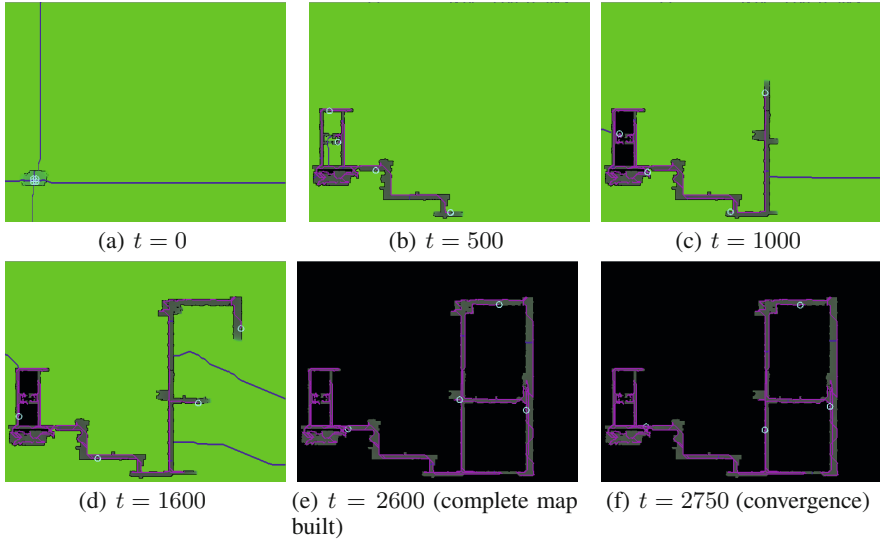


Fig. 6 Exploration and coverage of a large unknown environment. Green indicates uncertainty.

- i. Each robot maintains its own probability, entropy and obstacle maps.
- ii. Each robot use sensor data as well as communicate with its neighbors to update the maps. They also communicate their locations.
- iii. Each robot computes its own entropy-weighted Voronoi tessellation and the corresponding weighted projected centroid, and take a step towards that along the shortest path.

5.4 Results

Figure 6 shows the screenshots from a simulation of four robots exploring a large (1000×783 uniformly discretized) cluttered environment. The boundaries of the tessellations are shown by the bold blue lines. The robot positions are encircled by cyan circles. The dark lines show the robot trajectories. The intensity of the pixels in the environment represent the entropy, and the unreachable regions are colored in black. The mobile robots begin at the room in the lower left with no prior knowledge about the environment, hence the highest value of entropy, $\ln(0.5)$, is assigned to each cell. Besides collaboratively exploring the environment the robots distribute themselves in such a way that they maintain proper coverage of the explored environment both during exploration and after completely building the map. The mobile robots attain full exploration, coverage, and convergence within $t = 2750$ iterations. Each iteration, which involves computing the voronoi tessellations as well as the control commands for all the robots, takes about $1.7s$ running on a single processor as described in earlier results.

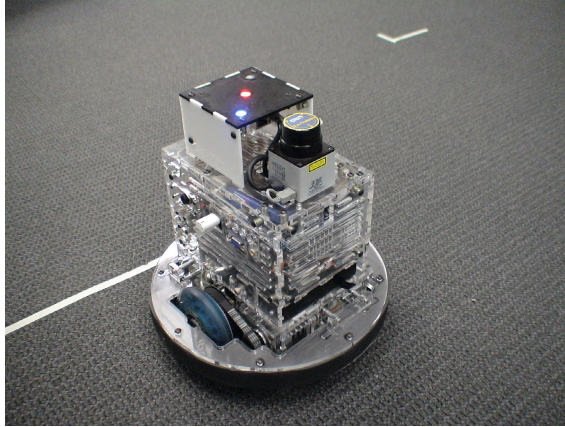


Fig. 7 The Scarab mobile robot platform

6 Conclusion and Future Work

We presented a search-based algorithm for computing a geodesic Voronoi tessellation in discrete environments. We propose a method for computing the centroid of the Voronoi cells given a tessellation of a discrete environment, permitting the application of centroid-based robot control laws for cooperative coverage. Uncertainty in the environment description is introduced through the development of an entropy-based metric; enabling the computation of the instantaneous geodesic Voronoi tessellation given an uncertain environment. We comment on the coverage and convergence guarantees resulting from this approach and present results that demonstrate through simulation its application in an office environment.

There are a few limitations in this paper that we are currently working on. First, as we move to experimentation with real robots, we must address real world issues surround localization and state estimation for the robots as well as inter-robot communication. To this end, we have already integrated our simulation model within the ROS (Robot Operating System) framework, and have started extending the implementation for running preliminary experiments on multiple Scarab robots [9] (see Fig. 7) which allow for on-board computation and localization using laser range sensors and monocular cameras. We are incorporating the anisotropy and finite field-of-view constraints that are characteristics of these sensors within our sensor model and the uncertainty associated with localization in our entropic measure. In addition, we are exploring models for inter-robot communication to relax the current assumption of a complete communication graph in the paper. Finally, we are also addressing algorithmic improvements to allow distributed computation and to enhance efficiency.

References

1. Aronov, B.: On the geodesic voronoi diagram of point sites in a simple polygon. In: Proc. of the Sym. on Computational Geometry, pp. 39–49. ACM, New York (1987), doi:<http://doi.acm.org/10.1145/41958.41963>
2. Burgard, W., Moors, M., Stachniss, C., Schneider, F.: Coordinated multi-robot exploration. *IEEE Trans. Robot.* 21(3), 376–378 (2005)
3. Cormen, T.H., Stein, C., Rivest, R.L., Leiserson, C.E.: Introduction to Algorithms. McGraw-Hill Higher Education (2001)
4. Cortes, J., Martinez, S., Karatas, T., Bullo, F.: Coverage control for mobile sensing networks. *IEEE Trans. Robot. Autom.* 20(2), 243–255 (2004)
5. Dijkstra, E.W.: A note on two problems in connexion with graphs. *Numerische Mathematik* 1, 269–271 (1959)
6. Fox, D., Ko, J., Konolige, K., Limketkai, B., Schultz, D., Stewart, B.: Distributed multi-robot exploration and mapping. *Proc. of the IEEE* 94(7), 1325–1339 (2006)
7. Howard, A., Parker, L.E., Sukhatme, G.S.: Experiments with a large heterogeneous mobile robot team: Exploration, mapping, deployment and detection. *Intl. J. Robot. Research* 25(5-6), 431–447 (2006)
8. Lloyd, S.P.: Least squares quantization in PCM. *IEEE Trans. Inf. Theory* 28, 129–137 (1982)
9. Michael, N., Fink, J., Kumar, V.: Experimental testbed for large multi-robot teams: Verification and validation. *IEEE Robot. Autom. Mag.* 15(1), 53–61 (2008)
10. Pimenta, L.C.A., Kumar, V., Mesquita, R.C., Pereira, G.A.S.: Sensing and coverage for a network of heterogeneous robots. In: Proc. of the IEEE Conf. on Decision and Control, Cancun, Mexico, pp. 3947–3952 (2008)
11. Preparata, F.P., Shamos, M.I.: Computational geometry: an introduction. Springer-Verlag New York, Inc., New York (1985)
12. Schwager, M., McLurkin, J., Rus, D.: Distributed coverage control with sensory feedback for networked robots. In: Proc. of Robot.: Sci. and Syst., Philadelphia, PA (2006)
13. Schwager, M., Slotine, J.E., Rus, D.: Decentralized, adaptive control for coverage with networked robots. In: Proc. of the IEEE Intl. Conf. on Robot. and Autom., Rome, Italy, pp. 3289–3294 (2007)
14. Stachniss, C.: Exploration and mapping with mobile robots. Ph.D. thesis, University of Freiburg, Freiburg, Germany (2006)
15. Stachniss, C., Grisetti, G., Burgard, W.: Information gain-based exploration using rao-blackwellized particle filters. In: Proc. of Robot.: Sci. and Syst., Cambridge, MA, pp. 65–72 (2005)
16. Thrun, S., Burgard, W., Fox, D.: Probabilistic Robotics (Intelligent Robotics and Autonomous Agents). The MIT Press (2005)
17. Wu, L., Garcia, M.A., Puig, D., Sole, A.: Voronoi-based space partitioning for coordinated multi-robot exploration. *J. Physical Agents* 1(1), 37–44 (2007)