Distributed Coverage and Exploration in Unknown Non-Convex Environments

Subhrajit Bhattacharya, Nathan Michael and Vijay Kumar

GRASP Laboratory,
University of Pennsylvania,
Philadelphia, PA 19104, USA
{subhrabh, nmichael, kumar}@grasp.upenn.edu

Summary. 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
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 [1]. 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 [2], 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 [3, 4], 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 [5], 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 [7], 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 [8] build upon this idea and develop decentralized control laws that position a mobile sensor network optimally with respect to a known probability distribution. In [9], 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 [11]. 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 sensors in the environment, and in particular the position of the i^{th} sensor is represented by $\mathbf{p}_i \in \Omega$ and the tessellation associated with it by W_i, $\forall i = 1, 2, \ldots, n$. 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 sensor positions $P = \{\mathbf{p}_1, \mathbf{p}_2, \ldots, \mathbf{p}_n\}$ and tessellations $W = \{W_1, W_2, \ldots, W_n\}$ such that $\mathbf{p}_i \in W_i, \forall i = 1, 2, \ldots, n$, 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, $f : \mathbb{R} \to \mathbb{R}$ is a smooth and strictly increasing function in the range of d, $\phi : \Omega \to \mathbb{R}$ 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 [12] and its continuous-time asynchronous implementations [7] 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. $d(\mathbf{p}, \mathbf{q}) = \|\mathbf{p} - \mathbf{q}\|_2$).

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,

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

Geometric methods for computing such geodesic Voronoi tessellations in non-convex polygonal environments are detailed in [10, 13]. 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.

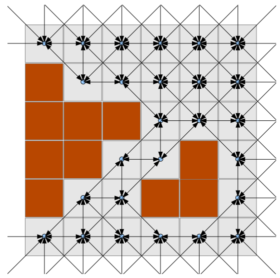


Fig. 1. An 8-connected grid graph created from a uniformly discretized environment.

4.2 Search-Based Algorithm for Finding Geodesic Voronoi Tessellations

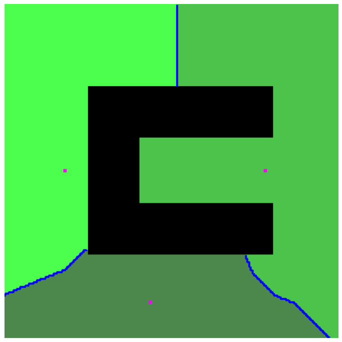
We propose a search-based algorithm for finding the geodesic Voronoi tessellations. We begin by uniformly discretizing Ω and creating a graph such that each node or vertex of the graph corresponds to a cell of the discretization with connections to admissible neighbors (see Fig. 1). This graph, $\mathcal{G}_\Omega$, 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. $\mathcal{N}(\mathbf{q}) := \{\mathbf{s} \in \mathcal{V}(\mathcal{G}_\Omega) \mid \overline{\mathbf{sq}} \in \mathcal{E}(\mathcal{G}_\Omega)\}$ 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. $c(\varepsilon) = \|\mathbf{v}_s(\varepsilon) - \mathbf{v}_t(\varepsilon)\|_2$).

For each of the sensors we perform a Dijkstra's search [14, 15] in $\mathcal{G}_\Omega$ starting from the vertex where the sensor itself is located, $\mathbf{p}_i$, and expand all the vertices in $\mathcal{G}_\Omega$. Thus, at the end of the expansions, for each vertex $\mathbf{q} \in \mathcal{V}(\mathcal{G}_\Omega)$ we obtain the values of $g_i(\mathbf{q})$, $\forall i = 1, 2, \ldots, n$, such that $g_i(\mathbf{q})$ gives the cost of the shortest path between $\mathbf{p}_i$ and $\mathbf{q}$ in $\mathcal{G}_\Omega$. The geodesic Voronoi tessellation is created by assigning the cells or vertices to the sensor which has the least g-value at that node. That is, $V_i = \{\mathbf{q} \in \mathcal{V}(\mathcal{G}_\Omega) \,|\, g_i(\mathbf{q}) \le g_j(\mathbf{q}), \forall j \neq i\}$ and $\mathbf{q} \in V_i \iff g_i(\mathbf{q}) \le g_j(\mathbf{q}), \forall j \neq i$. 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.

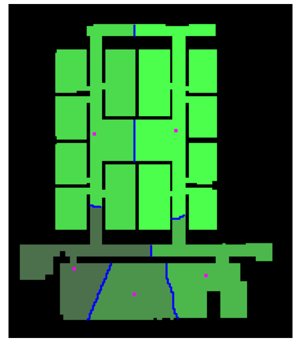
For the distributed architecture, each sensor maintains its own copy of $\mathcal{G}_\Omega$, updating it (for location of obstacles and values of entropies) using its own sensor readings as well as information acquired from its neighboring sensors about parts of their copies of $\mathcal{G}_\Omega$ close to $\mathbf{p}_i$. For determining which sensors need to be in the set of neighboring sensors, and how much information needs to be shared, we can use some heuristics. However, with our assumption of unlimited communication, for the simulations we choose to share entire $\mathcal{G}_\Omega$ and values of g_i among the sensors.

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 sensors that share the boundary are equal.



(a) 3 sensors in a simple environment (200×200 discretized).



(b) 5 sensors in an office environment (170×200 discretized).

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

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 sensor follow the gradient of $\mathcal{H}(P) := \mathcal{H}(P, V(P))$ given by the formula [10]:

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

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

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

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 sensor 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} sensor, 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 sensors 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 sensors).

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*,

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

in a non-convex environment is difficult [10]. However, a coverage control law over Voronoi tessellations such as that proposed in [7]:

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

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

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

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

$$\underline{\mathbf{C}}_{V_i} = \underset{\mathbf{q} \in V_i}{\operatorname{argmin}} \|\mathbf{q} - \mathbf{C}_{V_i}\|. \tag{7}$$

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,

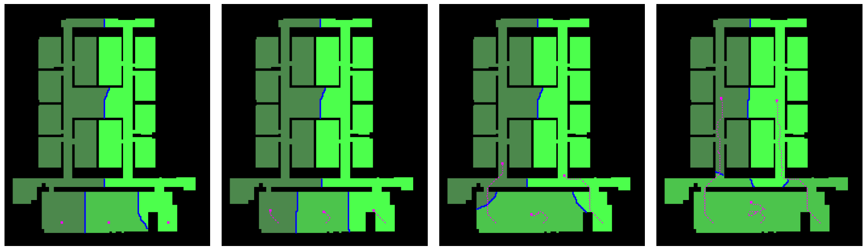
$$\overline{\mathbf{C}}_{V_i} = \begin{cases} \underset{\mathbf{q} \in V_i, \phi(\mathbf{q}) \geq \tau}{\operatorname{argmin}} \|\mathbf{q} - \mathbf{C}_{V_i}\| & \text{if it exists} \\ \underset{\mathbf{q} \in V_i}{\operatorname{argmin}} \|\mathbf{q} - \mathbf{C}_{V_i}\| & \text{otherwise.} \end{cases}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

for some threshold τ, and use this projection.

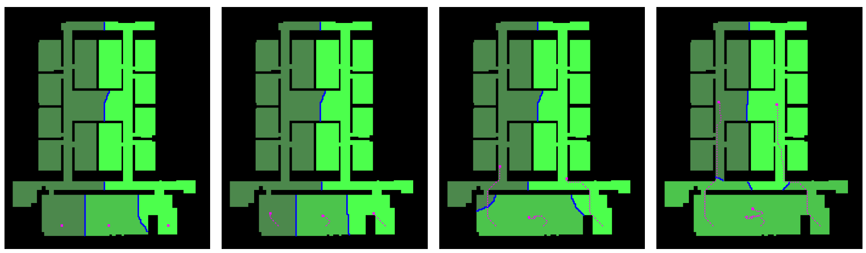
The control law for the i^{th} sensor given a discrete formulation reduces to taking one step towards $\overline{\mathbf{C}}_{V_i(P^t)}$ along the shortest path joining $\mathbf{p}_i^t$ and $\overline{\mathbf{C}}_{V_i(P^t)}$ (found via a single Dijkstra's search in V_i). This approach requires less computation 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 sensor positions $P^* = \{\mathbf{p}_1^*, \ldots, \mathbf{p}_n^*\}$ 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 sensors 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 $\overline{\mathbf{C}}_{V_i} = \mathbf{C}_{V_i}^{gen}$, 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, $\mathbf{C}_{V_i} = \mathbf{C}_{V_i}^{gen} \Rightarrow \overline{\mathbf{C}}_{V_i} = \mathbf{C}_{V_i}^{gen}$ (since $\mathbf{C}_{V_i}^{gen}$ always lies inside V_i). The condition under which $\mathbf{C}_{V_i} = \mathbf{C}_{V_i}^{gen}$ 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* [16] with respect to $\mathbf{C}_{V_i}$. A trivial



(a) $t = 0$ (b) $t = 10$ (c) $t = 50$ (d) $t = 150$ (converged)



(e) $t = 0$ (f) $t = 10$ (g) $t = 50$ (h) $t = 150$ (converged)

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.

case of this condition is when V_i is convex, when the algorithm becomes equivalent to the continuous-time Lloyd's Algorithm [7].

For comparison, Figs. 3(e)–3(h) show the evolution of the geodesic Voronoi tessellations and the trajectories followed by the mobile sensors 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.

5 Simultaneous Exploration and Coverage of Non-Convex Environments

In this section, we consider the problem of deploying n mobile sensors 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 sensor maintains and communicates a probability map for the discretized environment such that $p(\mathbf{q})$ is the probability that the vertex $\mathbf{q}$ is

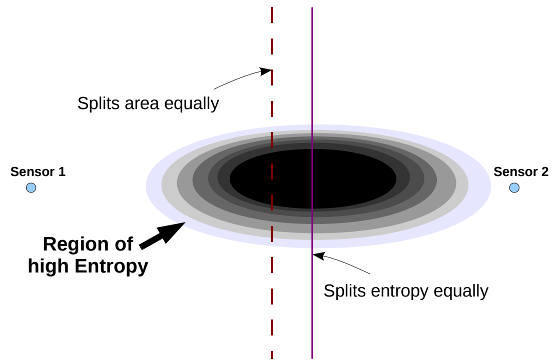


Fig. 4. Entropy-weighted Voronoi tessellation.

inaccessible (*i.e.* occupied or represents an obstacle), for all $\mathbf{q} \in \mathcal{V}(\mathcal{G}_\Omega)$. The Shannon entropy for each cell is

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

This 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 sensors towards regions of high entropy within the sensor'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 sensor 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 sensors. 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 sensors for cooperative exploration. This notion is illustrated in Fig. 4, where a high entropy region is placed asymmetrically between two sensors 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 sensor 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 accomodate 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:

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

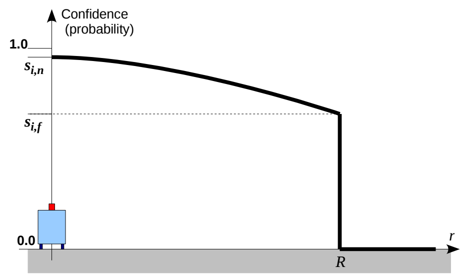


Fig. 5. The sensor model.

where $\mathbf{r}$ is a point on γ.

In terms of finding the Voronoi Tessellations by performing Dijkstra's searches in $\mathcal{G}_\Omega$ as described earlier, the only required change is to weigh the edges of the graph $\mathcal{G}_\Omega$ 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

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

where we add the second therm 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 sensors 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.

Sensor Model

We use a sensor model for each robot, $s_i(r)$, which gives the probability that the i^{th} 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 i^{th} sensor 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 [1].

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.

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:

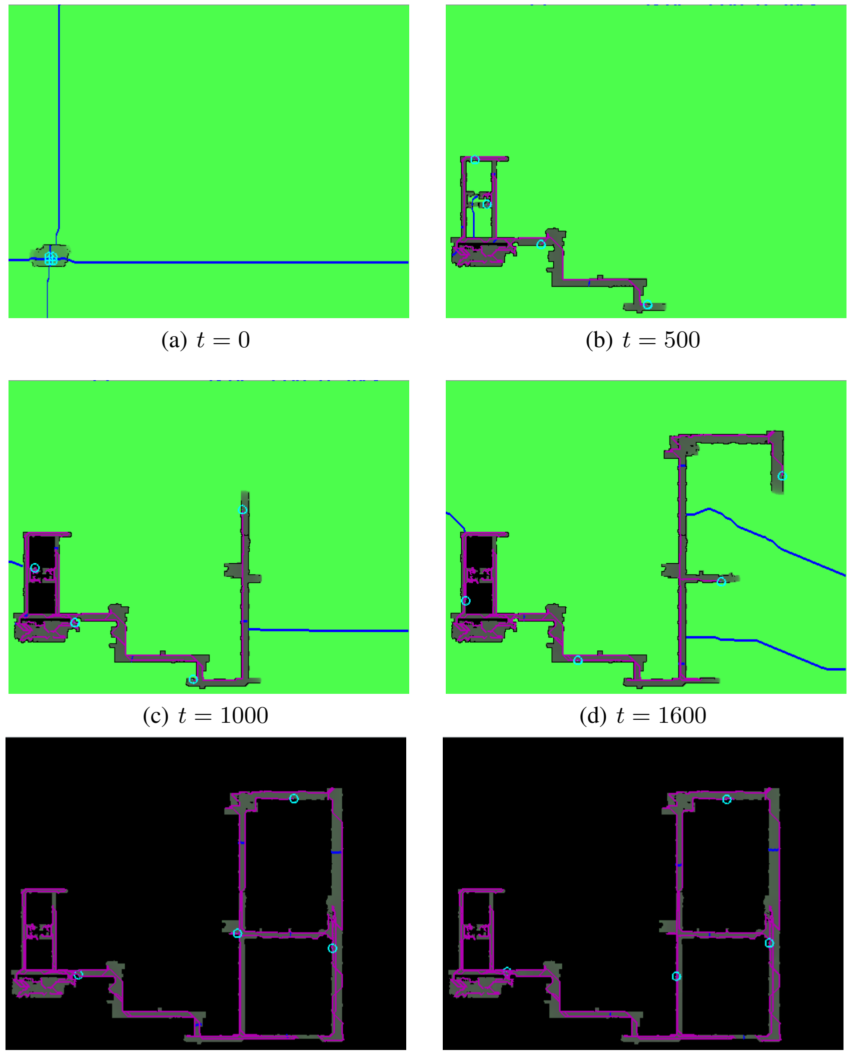
$$\phi(\mathbf{q},t) = e(\mathbf{q},t) = p^t(\mathbf{q})\ln(p^t(\mathbf{q})) + (1-p^t(\mathbf{q}))\ln(1-p^t(\mathbf{q})).$$

Conjecture 2 (Coverage 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 coverage and convergence.

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 sensor 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 coverage 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 $\tau' \approx \tau$, the density function becomes $\phi(\mathbf{q},t_{covered}) \approx \tau' \approx \tau$, which is constant and independent of time throughout the environment. Consequently by Conjecture 1 convergence is achieved at some $t_{converged} \geq t_{covered}$.

5.4 Results

Figure 6 shows the screenshots 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 sensor 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 sensors 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 sensors 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.



(a) $t = 0$ (b) $t = 500$

(c) $t = 1000$ (d) $t = 1600$

(e) $t = 2600$ (complete map built) (f) $t = 2750$ (convergence)

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

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.

We are currently moving forward in two directions with this work. First, we are actively preparing real experiments with multiple robots in an indoor environment with the same layout as Fig. 6. Toward this goal, we must address the fact that we assume the existence of ubiquitous communication and that our approach requires explicit coordination at each iteration of the algorithm between neighboring robots. Second, we are extending the sensor model to consider a sensor footprint similar to that of the scanning lasers that we plan to use in the experiments.

References

1. S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics (Intelligent Robotics and Autonomous Agents)*. The MIT Press, 2005.
2. C. Stachniss, G. Grisetti, and W. Burgard, "Information gain-based exploration using rao-blackwellized particle filters," in *Proc. of Robot.: Sci. and Syst.*, Cambridge, MA, June 2005, pp. 65–72.
3. W. Burgard, M. Moors, C. Stachniss, and F. Schneider, "Coordinated multi-robot exploration," *IEEE Trans. Robot.*, vol. 21, no. 3, pp. 376–378, June 2005.
4. C. Stachniss, "Exploration and mapping with mobile robots," Ph.D. dissertation, University of Freiburg, Freiburg, Germany, Apr. 2006.
5. A. Howard, L. E. Parker, and G. S. Sukhatme, "Experiments with a large heterogeneous mobile robot team: Exploration, mapping, deployment and detection," *Intl. J. Robot. Research*, vol. 25, no. 5-6, pp. 431–447, May 2006.
6. D. Fox, J. Ko, K. Konolige, B. Limketkai, D. Schultz, and B. Stewart, "Distributed multirobot exploration and mapping," *Proc. of the IEEE*, vol. 94, no. 7, pp. 1325–1339, July 2006.
7. J. Cortes, S. Martinez, T. Karatas, and F. Bullo, "Coverage control for mobile sensing networks," *IEEE Trans. Robot. Autom.*, vol. 20, no. 2, pp. 243–255, Apr. 2004.
8. M. Schwager, J. McLurkin, and D. Rus, "Distributed coverage control with sensory feedback for networked robots," in *Proc. of Robot.: Sci. and Syst.*, Philadelphia, PA, Aug. 2006.
9. M. Schwager, J. E. Slotine, and D. Rus, "Decentralized, adaptive control for coverage with networked robots," in *Proc. of the IEEE Intl. Conf. on Robot. and Autom.*, Rome, Italy, Apr. 2007, pp. 3289–3294.
10. L. C. A. Pimenta, V. Kumar, R. C. Mesquita, and G. A. S. Pereira, "Sensing and coverage for a network of heterogeneous robots," in *Proc. of the IEEE Conf. on Decision and Control*, Cancun, Mexico, Dec. 2008, pp. 3947–3952.
11. L. Wu, M. A. Garcia, D. Puig, and A. Sole, "Voronoi-based space partitioning for coordinated multi-robot exploration," *J. Physical Agents*, vol. 1, no. 1, pp. 37–44, Sept. 2007.
12. S. P. Lloyd, "Least squares quantization in PCM," *IEEE Trans. Inf. Theory*, vol. 28, pp. 129–137, 1982.
13. B. Aronov, "On the geodesic voronoi diagram of point sites in a simple polygon," in *Proc. of the Sym. on Computational Geometry*. New York, NY, USA: ACM, 1987, pp. 39–49.
14. E. W. Dijkstra, "A note on two problems in connexion with graphs," *Numerische Mathematik*, vol. 1, pp. 269–271, 1959.
15. T. H. Cormen, C. Stein, R. L. Rivest, and C. E. Leiserson, *Introduction to Algorithms*. McGraw-Hill Higher Education, 2001.
16. F. P. Preparata and M. I. Shamos, *Computational geometry: an introduction*. New York, NY, USA: Springer-Verlag New York, Inc., 1985.