Multi-Robot Coverage and Exploration in Non-Euclidean Metric Spaces

Subhrajit Bhattacharya, Robert Ghrist and Vijay Kumar

Abstract Multi-robot coverage and exploration is a fundamental problem in robotics. A widely-used, efficient and distributable algorithm for achieving coverage of a convex environment with Euclidean metric is that proposed by Cortes, *et al.*, which is based on the discrete-time Lloyd's algorithm. It is significantly difficult to achieve the same in non-convex environments and with non-Euclidean metrics. In this paper we generalize the control law based on minimization of the *coverage functional* to spaces that are inherently non-Euclidean and are punctured by obstacles. We also propose a practical discrete implementation based on standard graph search-based algorithms. We demonstrate the applicability of the proposed algorithm by solving efficient coverage problems on a sphere and exploration problems in highly non-convex indoor environments. †

1 Introduction

The geometry underlying configuration spaces of multiple robots is a critical feature implicit in several important challenges in planning and coordination. Metric considerations are fundamental to problems of coverage [12, 6, 7, 3], exploration [16, 15, 14], and more. A well-know approach to solving coverage problems with n robots involve partitioning the appropriate configuration space into *n-tessellations* (a partition of the configuration space into simply-connected domains) [12, 6]. In particular, this method requires a Voronoi tessellation that implicates the configuration space geometry. While such a tessellation is easy to achieve in a convex environment with Euclidean metric, it becomes increasingly difficult in environments with obstacles and non-Euclidean metrics. The ubiquitous presence of obstacles removes all hope of convexity. Non-Euclidean metrics can arise in the geometry of a configuration space as inherited from the structure of the underlying domain (*e.g.*, from irregular terrain), or via direct manipulation of the configura-

Subhrajit Bhattacharya, Robert Ghrist and Vijay Kumar
GRASP Laboratory, University of Pennsylvania,
e-mail: [subhrabh,ghrist,kumar]@seas.upenn.edu

† We gratefully acknowledge the support of ONR Grants N00014-07-1-0829 and N00014-09-1-1031, and AFOSR Grant FA9550-10-1-0567.

tion space geometry for problem goals (*e.g.*, in multi-robot cooperative exploration problems [2]).

The problem of attaining balanced coverage of an environment is fundamental to many practical multi-robot problems. One common coverage control approach — efficient and distributable — is through the definition of feedback control laws defined with respect to the centroids of Voronoi cells resulting from the Voronoi tessellation of the domain. Lloyd's algorithm [12] is a discrete-time algorithm that minimizes a *coverage functional*. A continuous-time version of the algorithm is described in [5], where 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. To address the limitation of requiring a convex environment, the authors of [13] propose the use of *geodesic Voronoi tessellations* determined by the geodesic distance rather than the Euclidean distance. However such a method both involves computationally difficult geometric computations and is still limited to Euclidean environments with polygonal obstacles. Recent work [2] has used a graph search-based approach to develop tools for solving the coverage problem in non-convex environments with a non-Euclidean metric. However, in order to explicitly compute an analogue of a *generalized centroid* in non-convex tessellations, an approximate method involving *centroid projection* was used. Such a method is, admittedly, ad hoc, gives weak guarantees, and is difficult to implement when the configuration space is not sufficiently topologically simple (equivalent to a punctured simply-connected domain). There exists search-based discrete-time algorithms that explicitly searches every vertex in a tessellation to find the best position for the robot in every time-step (as in [10]). Although such a controller can solve the problem of multi-robot decentralized coverage on arbitrary metric graphs, the high computational complexity of this approach makes it impractical for fine discretization or large graphs.

In this paper we generalize the method for computing the control law that is described in [13] and adapt it to non-Euclidean metric spaces with obstacles that are not necessarily polygonal. The principal theoretical tools are Proposition 1 and Corollary 1, which relate geodesics, distance derivatives, and the metric tensor. This inspires the use of the control law in concert with a graph search-based methods to achieve an efficient discrete implementation. Our results are supported by the following demonstrations:

1. Coverage problems on non-Euclidean Riemannian manifolds with boundaries;
2. Multi-robot cooperative exploration; and
3. Human-robot interaction in exploration.

2 Background: Coverage Functional, Voronoi Tessellation and Continuous-time Lloyd's Algorithm

In this section we discuss the basic concepts behind deriving the control laws in the continuous-time version of the Lloyd's algorithm [12, 13]. Let Ω be a path-connected metric space that represents the environment, equipped with a distance

function, d. In [12, 13] Ω is assumed to be a subset of $\mathbb{R}^D$ and is equipped with the Euclidean metric tensor (a Riemannian metric) at every point. However, in the present scenario we will relax d to a more general class of distance functions (for example, the *path metric* induced by general Riemannian metric – *i.e.* the length of the shortest rectifiable path [11]. For Corollary 1, we will in fact be able to consider more general Finsler metrics.).

There are n mobile robots in the environment, and in particular the position of the k^{th} robot is represented by $\mathbf{p}_k \in \Omega$ and the tessellation [12, 13] associated with it by W_k, $\forall k = 1, 2, \ldots, n$ (Thus, $\{W_k\}_{k=1,\cdots,n}$ is a cover of Ω). By definition, the tessellations are such that $Int(W_k) \cap Int(W_l) = \emptyset, \forall k \neq l$, and $\cup_{k=1}^{n} W_k = \Omega$. For a given set of robot positions $P = \{\mathbf{p}_1, \mathbf{p}_2, \ldots, \mathbf{p}_n\}$ and tessellations $W = \{W_1, W_2, \ldots, W_n\}$ such that $\mathbf{p}_k \in Int(W_k), \forall k = 1, 2, \ldots, n$, the *coverage functional* is defined as:

$$\mathscr{H}(P,W) = \sum_{k=1}^{n} \mathscr{H}(\mathbf{p}_k, W_k) = \sum_{k=1}^{n} \int_{W_k} f_k(d(\mathbf{q}, \mathbf{p}_k))\phi(\mathbf{q})\, \mathrm{d}\mathbf{q} \tag{1}$$

where $f_k : \mathbb{R}^+ \to \mathbb{R}$ are smooth and strictly increasing functions, $\phi : \Omega \to \mathbb{R}$ is a weight or density function, and $\mathrm{d}\mathbf{q}$ represents an infinitesimal volume element.

The name "*coverage functional*" is indicative of the fact that $\mathscr{H}$ measures how *bad* the coverage is. In fact, for a given set of initial robot positions, P, the eventual aim of the algorithm is to devise a control law that minimizes the function $\tilde{\mathscr{H}}(P) := \min_{W} \mathscr{H}(P,W)$ (*i.e.* the best value of $\mathscr{H}(P,W)$ for a given P). It is easy to show [12, 13] that $\tilde{\mathscr{H}}(P) = \mathscr{H}(P,V)$, where $V = \{V_1, V_2, \cdots, V_n\}$ is the Voronoi tessellation given by

$$V_k = \{\mathbf{q} \in \Omega \mid f_k(d(\mathbf{q}, \mathbf{p}_k)) \leq f_l(d(\mathbf{q}, \mathbf{p}_l)), \forall l \neq k\} \tag{2}$$

Thus the control law for minimizing $\tilde{\mathscr{H}}(P) = \sum_{k=1}^{n} \int_{V_k} f_k(d(\mathbf{q}, \mathbf{p}_k))\phi(\mathbf{q})\, \mathrm{d}\mathbf{q}$ can be reduced to the problem of following its gradient. Although V_k are functions of P, it can be shown using methods of differentiation under integration [13] that

$$\frac{\partial \tilde{\mathscr{H}}(P)}{\partial \mathbf{p}_k} = \int_{V_k} \frac{\partial}{\partial \mathbf{p}_k} f_k(d(\mathbf{q}, \mathbf{p}_k))\ \phi(\mathbf{q})\, \mathrm{d}\mathbf{q} \tag{3}$$

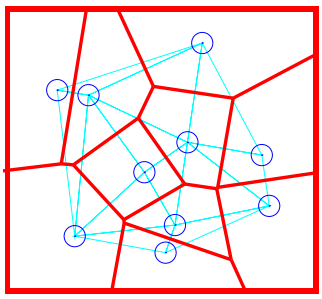
In practice, it is adequate to choose $f_k(x) = x^2$ for most practical implementations. However, a variation of the problem for taking into account finite sensor footprint of the robots, constructs a *power Voronoi tessellation* [13], in which one chooses $f_k(x) = x^2 - R_k^2$, where R_k can represent, for example, the radius of the sensor footprint of the k^{th} robot. In this paper we will be working with the following form for f_k

$$f_k(x) = x^2 + c_k \tag{4}$$

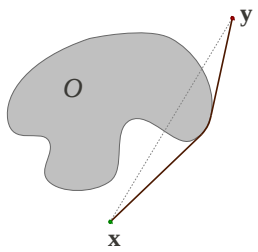
2.1 Euclidean and Non-Euclidean Metric Spaces

Until now we haven't made any major assumption on the distance function d. However, if the space Ω is convex, and the metric η is Euclidean, then d is the Euclidean distance given by $d(\mathbf{x}, \mathbf{y}) = \|\mathbf{x} - \mathbf{y}\|_2$. Under this assumption, and using the form of f_k in (4), the formula of (3) can be simplified to obtain

$$\frac{\partial \tilde{\mathscr{H}}(P)}{\partial \mathbf{p}_k} = 2A_k(\mathbf{p}_k - \mathbf{p}_k^*) \tag{5}$$



(a) Voronoi tessellation of a convex Ω (rectangular region) with $n = 10$ robots (blue circles). The red line segments show the boundary of the tessellations. Note how a boundary segment is the perpendicular bisector of the cyan line joining the robots sharing the boundary segment.



(b) Presence of holes/punctures (due to obstacles) in the Euclidean space changes the distance function in the punctured space. For example, in this figure, $d(\mathbf{x},\mathbf{y})$ is the length (induced by the Euclidean metric) of the thicker curve that is the shortest connecting $\mathbf{x}$ and $\mathbf{y}$ and lying entirely in $(\mathbb{R}^2 - O)$.

Fig. 1 Voronoi tessellation in a convex environment, and the difficulty in computing it in non-convex environments.

where, $A_k = \int_{V_k} \phi(\mathbf{q}) d\mathbf{q}$ is the weighted *volume* of V_k, and $\mathbf{p}_k^* = \frac{\int_{V_k} \mathbf{q}\phi(\mathbf{q}) d\mathbf{q}}{A_k}$ is the weighted centroid of V_k. Moreover, the Euclidean distance function makes computation of the Voronoi tessellation very easy: V, due to Equation (2), can be constructed from the perpendicular bisectors of the line segments $\overline{\mathbf{p}_k\mathbf{p}_l}$, $\forall k \neq l$, thus making each V_k a convex polygon, which are also simply connected (Figure 1(a)). This also enable closed-form computation of the volume, A_k, and the centroid, $\mathbf{p}_k^*$ when the weight function ϕ is uniform. Equation (5) yields the simple control law in continuous-time Lloyd's algorithm: $\mathbf{u}_k = -\kappa A_k(\mathbf{p}_k - \mathbf{p}_k^*)$, with some positive *gain*, κ. Lloyd's algorithm [12] and its continuous-time asynchronous implementations [5] are distributed algorithms for minimizing $\mathscr{H}(P,W)$ with guarantees on completeness and asymptotic convergence to a local optimum, when Ω is convex Euclidean.

However, the above simplification does not work when the distance function is not Euclidean. In robot configuration spaces punctured by obstacles, non-Euclidean distance functions can appear in the computations in two primary ways: **i.** Due to the presence of holes/obstacles, even when the path-metric is locally Euclidean in the interior of the domain, the global distance function is not Euclidean (Figure 1(b)), and **ii.** Locally non-Euclidean metric (*e.g.* in the case of a sphere, or a topologically Euclidean space with non-zero curvature). This makes the computation of the Voronoi tessellation in Equation (2) as well as the gradient of $\mathscr{H}$ in Equation (3) significantly difficult.

In Equation (3), with $f_k(x) = x^2 + c_k$, we observe that

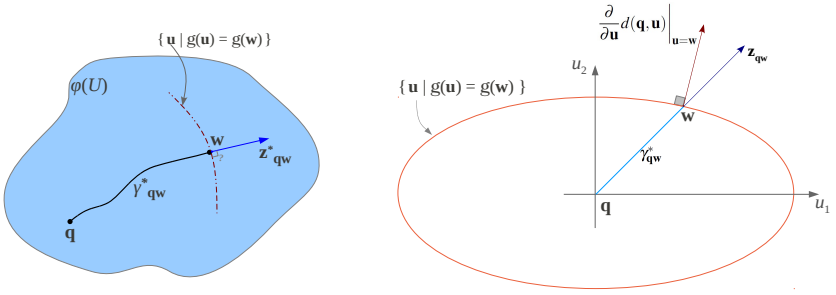
$$\frac{\partial}{\partial \mathbf{p}_k} f_k(d(\mathbf{q},\mathbf{p}_k)) = 2\, d(\mathbf{q},\mathbf{p}_k)\, \frac{\partial}{\partial \mathbf{p}_k} d(\mathbf{q},\mathbf{p}_k)$$

Thus, our first step will be to find an efficient way for computing $\frac{\partial}{\partial \mathbf{p}_k} d(\mathbf{q}, \mathbf{p}_k)$. In Section 3 we will show that in arbitrary metric spaces (satisfying certain conditions) this gradient of the distance function can be expressed in terms of tangents to geodesics. In Section 5 we will demonstrate how the later is computationally more favorable in a graph search-based algorithm for developing a generalized continuous-time Lloyd's algorithm.

3 Gradient of Distance Function in Non-Euclidean Metric Spaces

For computing $\frac{\partial}{\partial \mathbf{p}_k} d(\mathbf{q}, \mathbf{p}_k)$ we have the following proposition and corollary, which together gives a generalization of Proposition 4 of [13]. In the discussions that follow, we will assume summation over repeated indices, i and j, following the Einstein summation convention.

In Proposition 1 we essentially establish a relationship between gradient of the distance function and the tangent to a geodesic, provided the distance function satisfies some very restricted conditions (geodesics being unique and induced by Riemannian metric). In most robot configuration spaces, due to presence of non-convexity and obstacles, this proposition will not be sufficient. Thus we devise Corollary 1 to encompass a wider class of distance functions.



(a) Illustration for Proposition 1. It establishes a relation between the tangent to the geodesic $\gamma^*_{\mathbf{qw}}$ at $\mathbf{w}$, and the normal to the surface $\{\mathbf{u} | g(\mathbf{u}) = g(\mathbf{w})\}$ at $\mathbf{w}$.

(b) Illustration with a simple non-Euclidean, anisotropic metric. Note that the normal to the ellipse is not parallel to the tangent to the geodesic, $\mathbf{z}_{\mathbf{wq}}$. It is however parallel to the cotangent, $\mathbf{z}^*_{\mathbf{wq}}$, with coefficients $z^*_{j,\mathbf{qw}} = \eta_{ij}(\mathbf{w})\, z^j_{\mathbf{qw}}$.

Fig. 2 Relationship between tangent to a geodesic and the derivative of the distance function.

Proposition 1. *Let $C = (U, \phi)$ be a coordinate chart on a open subset U of a D-dimensional manifold, Ω with coordinate variables $u^1, u^2, \cdots, u^D$. Suppose U is Riemannian everywhere, equipped with a metric η, and the geodesic connecting any two points in U lies entirely in U.*

Let $d : \mathbb{R}^D \times \mathbb{R}^D \to \mathbb{R}$ be the distance function in $U \subseteq \Omega$ in terms of the coordinate chart C (i.e. $d(\mathbf{q}, \mathbf{w})$, for $\mathbf{q}, \mathbf{w} \in Img(\phi) \subseteq \mathbb{R}^D$, is the length of the shortest path connecting $\phi^{-1}(\mathbf{q})$ and $\phi^{-1}(\mathbf{w})$ in $U \subseteq \Omega$.). Suppose inside $Img(\phi)$, the distance d is induced by the Riemannian metric η, is smooth everywhere, and suppose there exists a unique geodesic of length $d(\mathbf{q}, \mathbf{w})$ connecting any two points $\mathbf{q}, \mathbf{w} \in Img(\phi)$.

Then the following is true for every $\mathbf{q},\mathbf{w} \in Img(\phi) \subseteq \mathbb{R}^D$ *and every coordinate chart, C, defined on U (Figure 2(a)),*

$$\left[\frac{\partial}{\partial \mathbf{u}} d(\mathbf{q},\mathbf{u})\Big|_{\mathbf{u}=\mathbf{w}}\right]_i \equiv \frac{\partial}{\partial u^i} d(\mathbf{q},\mathbf{u})\Big|_{\mathbf{u}=\mathbf{w}} = \frac{\eta_{ij}(\mathbf{w})\, z^j_{\mathbf{qw}}}{\sqrt{\eta_{mn}(\mathbf{w})\, z^m_{\mathbf{qw}} z^n_{\mathbf{qw}}}}$$

where, $\mathbf{z_{qw}} = [z^1_{\mathbf{qw}}, z^2_{\mathbf{qw}}, \cdots, z^D_{\mathbf{qw}}]^T$ *is a normalized coefficient vector of the tangent vector at* $\mathbf{w}$ *to the shortest geodesic connecting* $\mathbf{q}$ *to* $\mathbf{w}$*, and by* $\left[\frac{\partial f}{\partial \mathbf{u}}\right]_i$ *we mean the* i^{th} *component of* $\left[\frac{\partial f}{\partial u^1}, \frac{\partial f}{\partial u^2}, \cdots, \frac{\partial f}{\partial u^D}\right]$.

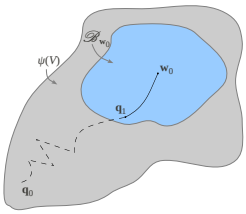
Proof. This result follows from well-known theorems in Riemannian geometry. A detailed proof is provided in [1]. □

If we define $g(\mathbf{u}) := d(\mathbf{q},\mathbf{u})$, $\forall \mathbf{u} \in Img(\phi)$ (*i.e.*, $g(\mathbf{w})$ is the length of the shortest geodesic connecting $\mathbf{q}$ to $\mathbf{w}$), the statement of the proposition essentially implies that the normals to the constant g surfaces in $\mathbb{R}^D$ are parallel to the cotangents to the geodesics. This is illustrated in Figure 2(a). The statement of the proposition essentially expresses the gradient of the distance function d (with respect to one of its arguments) in terms of the tangent to the geodesic connecting two points.

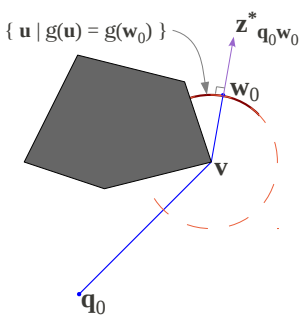
Examples:

1. We note that when the metric is Euclidean in the given chart (*i.e.* $\eta_{ij} = \delta_{ij}$ everywhere as was the case in [13]), the result of the proposition simply reduces to $\frac{\partial}{\partial u^i} d(\mathbf{q},\mathbf{u})\Big|_{\mathbf{u}=\mathbf{w}} = z^i_{\mathbf{qw}}$. This is no surprise since we know that the vector $\frac{\partial}{\partial \mathbf{u}} d(\mathbf{q},\mathbf{u})\Big|_{\mathbf{u}=\mathbf{w}}$ is essentially a unit normal to the sphere with center $\mathbf{q}$ (which is the surface of constant $d(\mathbf{q},\mathbf{u})$) at the point $\mathbf{u} = \mathbf{w}$, which is well-known to be parallel to the straight line connecting $\mathbf{q}$ to $\mathbf{w}$ (a radial line of the sphere).
2. If the metric is locally isotropic in the given chart (*i.e.* if the matrix representation of the metric is a multiple of the identity matrix at every point), and can be written as $\eta_{ij}(\mathbf{q}) = \zeta(\mathbf{q})\delta_{ij}$ for some $\zeta : \mathbb{R}^D \to \mathbb{R}$, then the result of the proposition reduces to $\frac{\partial}{\partial \mathbf{u}} d(\mathbf{q},\mathbf{u})\Big|_{\mathbf{u}=\mathbf{w}} = \sqrt{\zeta(\mathbf{w})}\ \mathbf{z}^T_{\mathbf{qw}}$ (where, $\mathbf{z}^T_{\mathbf{qw}} = [z^1_{\mathbf{qw}}, z^2_{\mathbf{qw}}, \cdots, z^D_{\mathbf{qw}}]$ is the transpose of the coefficient vector, $\mathbf{z_{qw}}$, of the tangent to the geodesic).
3. Finally, we consider a simple, yet nontrivial, example of a non-Euclidean, anisotropic metric. Consider the metric $\eta_{\bullet\bullet} = \begin{bmatrix} 1 & 0 \\ 0 & 4 \end{bmatrix}$. Since the Christoffel symbols vanish in this coordinate chart, one can infer from the geodesic equation that the geodesics are essentially represented by straight lines when plotted with u^i as orthogonal axes (Figure 2(b)). However, the curves of constant distance from $\mathbf{q}$ become ellipses centered at $\mathbf{q}$ and with aspect ratio of 2. Now consider the point $\mathbf{w} = \mathbf{q} + [1,1]^T$. A direct computation of the normal at this point to the ellipse, $(u^1 - q^1)^2/4 + (u^2 - q^2)^2 = c$, passing through this point, reveals the coefficient co-vector of $\frac{\partial}{\partial \mathbf{u}} d(\mathbf{q},\mathbf{u})\Big|_{\mathbf{u}=\mathbf{w}}$ to be parallel to $[\frac{1}{2}, 2]$. However, the coefficient vector of the tangent to the geodesic is $\mathbf{z_{qw}} = [\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}}]^T$. This gives the following:

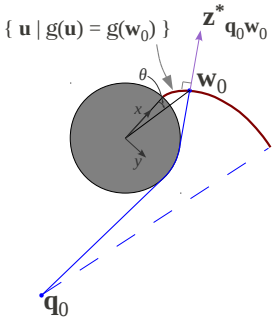
$\sqrt{\eta_{mn}(\mathbf{w})\, z^m_{\mathbf{qw}} z^n_{\mathbf{qw}}} = \sqrt{\frac{5}{2}}, \quad z_{1,\mathbf{qw}} = \sum_j \eta_{1j}\, z^j_{\mathbf{qw}} = \frac{1}{\sqrt{2}}, \quad z_{2,\mathbf{qw}} = \sum_j \eta_{2j}\, z^j_{\mathbf{qw}} = 2\sqrt{2}.$
Thus, the coefficient co-vector of $\mathbf{z}^*_{\mathbf{qw}}$ is parallel to $[\frac{1}{\sqrt{2}}, 2\sqrt{2}]$. This indeed is parallel to $[\frac{1}{2}, 2]$. The exact computation of the scalar multiple will require a more careful computation of $\frac{\partial}{\partial \mathbf{u}} d(\mathbf{q}, \mathbf{u})$.



(a) Illustration for Corollary 1. The pathologies outside $\mathscr{B}_{\mathbf{w}_0}$ do not effect the result of Proposition 1 holding for $\mathbf{q}_0$ and $\mathbf{w}_0$.



(b) The simplest example is that of a space that is equipped with Euclidean metric everywhere, but is punctured by a polygonal obstacles. This was the case considered in [13].



(c) A more interesting case is that of involutes generated in locally Euclidean space using the boundaries of arbitrary obstacles as the generating curves.

Fig. 3 Corollary 1 and examples demonstrating it.

Corollary 1. *Let $C = (V, \psi)$ be a coordinate chart on a open subset V of a D-dimensional manifold, Ω. Let $d : \mathbb{R}^D \times \mathbb{R}^D \to \mathbb{R}$ be the distance function on V in terms of the coordinate chart C (i.e. $d(\mathbf{q}, \mathbf{w})$, for $\mathbf{q}, \mathbf{w} \in Img(\psi) \subseteq \mathbb{R}^D$, is the distance between $\psi^{-1}(\mathbf{q})$ and $\psi^{-1}(\mathbf{w})$ in $V \subseteq \Omega$).*

We are given $\mathbf{q}_0, \mathbf{w}_0 \in Img(\psi) \subseteq \mathbb{R}^D$. Suppose there exists a open neighborhood $\mathscr{B}_{\mathbf{w}_0} \subseteq \mathbb{R}^D$ of $\mathbf{w}_0$ (Figure 3(a)) such that,

a. $g(\cdot) := d(\mathbf{q}_0, \cdot)$ is smooth everywhere in $\mathscr{B}_{\mathbf{w}_0}$,
b. The distance function restricted to $\mathscr{B}_{\mathbf{w}_0} \times \mathscr{B}_{\mathbf{w}_0}$ is induced by a Riemannian metric, η, such that for any two $\mathbf{u}, \mathbf{v} \in \mathscr{B}_{\mathbf{w}_0}$, there is a unique shortest geodesic $\gamma_{\mathbf{uv}}$ of length $d(\mathbf{u}, \mathbf{v})$.
c. A shortest path (of length $d(\mathbf{q}_0, \mathbf{w}_0)$) is defined between $\mathbf{q}_0$ and $\mathbf{w}_0$, such that the part of the shortest path connecting $\mathbf{q}_0$ and $\mathbf{w}_0$ that lies inside $\mathscr{B}_{\mathbf{w}_0}$ is unique,

Then the following holds,

$$\left[\frac{\partial}{\partial \mathbf{u}} d(\mathbf{q}_0,\mathbf{u})\Big|_{\mathbf{u}=\mathbf{w}_0}\right]_i \equiv \frac{\partial}{\partial u^i} d(\mathbf{q}_0,\mathbf{u})\Big|_{\mathbf{u}=\mathbf{w}_0} = \frac{\eta_{ij}(\mathbf{w}_0)\, z^j_{\mathbf{q}_0\mathbf{w}_0}}{\sqrt{\eta_{mn}(\mathbf{w}_0)\, z^m_{\mathbf{q}_0\mathbf{w}_0} z^n_{\mathbf{q}_0\mathbf{w}_0}}}$$

Note that the derivative $\frac{\partial}{\partial \mathbf{u}} d(\mathbf{q}_0,\mathbf{u})\Big|_{\mathbf{u}=\mathbf{w}_0}$ *is defined due to assumption 'a.', and the tangent* $\mathbf{z}_{\mathbf{q}_0\mathbf{w}_0}$ *exists due to assumptions 'b.' and 'c.'.*

Proof. This result follows from well-known theorems in Riemannian geometry and from Proposition 1. A detailed proof is provided in [1]. □

The statement of this Corollary encompasses a significantly wider class of metric spaces than Proposition 1. Here we only need to assume a Riemannian metric in the neighborhood of $\mathbf{w}_0$ (Figure 3(a)). This will enable us to use the result for locally Riemannian manifolds with pathologies outside local neighborhoods (*e.g.* boundaries/holes/punctures/obstacles – the kind of spaces we are most interested in), as well as opens up possibilities for more general metric spaces that may not be Riemannian outside $\mathscr{B}_{\mathbf{w}_0}$ (e.g. Manhattan metric in $\mathscr{M} \subset \Omega$, Riemannian metric elsewhere).

Examples:

1. The simplest example is that of a space that is locally Euclidean (*i.e.* equipped with a Euclidean metric everywhere), but is punctured by polygonal obstacles (Figure 3(b)). Due to the 'pointedness' of the obstacles, the constant-g manifolds are essentially circular arcs centered at $\mathbf{q}_0$ or a vertex $\mathbf{v}$ of a polygon. Thus, as illustrated by Figure 3(b), the normals to the arcs are parallel to the tangent to the segment joining $\mathbf{v}$ to $\mathbf{w}_0$.
2. A little less trivial example occurs when the obstacles are not polygonal. Then the statement of the corollary essentially reduces to the assertion that the normal at any point on an involute [8] is parallel to the 'taut string', the end of which traces the involute – and this is true irrespective of the curve used to generate the involute. While the statement has an obvious intuitive explanation by considering the possible directions of motion of the end of the taut string, we provide an explicit computation for an involute created using a circle (Figure 3(c)). Consider a taut string unwrapping off a circle of radius r (starting from $\theta = 0$ when it is completely wrapped). Thus, when the string has unwrapped by an angle θ, the string points at a direction $[\sin(\theta), -\cos(\theta)]^T$. Now, it is easy to verify that the involute is described by the parametric curve $x = r(\cos(\theta) + \theta\sin(\theta)), y = r(\sin(\theta) - \theta\cos(\theta))$. Thus we have $\frac{\mathrm{d}x}{\mathrm{d}\theta} = \theta\cos(\theta)$, $\frac{\mathrm{d}y}{\mathrm{d}\theta} = \theta\sin(\theta)$. Thus the normal to the involute pointing in the direction $[\frac{\mathrm{d}y}{\mathrm{d}\theta}, -\frac{\mathrm{d}x}{\mathrm{d}\theta}]$ is indeed parallel to the direction in which the string points.

4 Generalized Continuous-Time Lloyd's Algorithm

Corollary 1, along with the assumption that $f_k(x)$ is of the form $x^2 + c$, enables us to re-write Equation (3) for the most general metric setup as follows

$$\left[\frac{\partial \tilde{\mathscr{H}}(P)}{\partial \mathbf{p}_k}\right]_i = 2\int_{V_k} d(\mathbf{q},\mathbf{p}_k)\ \frac{\eta_{ij}(\mathbf{p}_k)\ z^j_{\mathbf{qp}_k}}{\sqrt{\eta_{mn}(\mathbf{p}_k)\ z^m_{\mathbf{qp}_k} z^n_{\mathbf{qp}_k}}}\ \phi(\mathbf{q})\ \ \mathrm{d}\mathbf{q} \tag{6}$$

This, as we will see later, gives an algorithm for approximately computing the gradient of $\tilde{\mathscr{H}}$. Note that $\mathrm{d}\mathbf{q}$ represents an infinitesimal volume element. Thus, in a discretized setup (with uniform discretization of the coordinate space, which we will discuss in the next section), when we replace the integral by a summation over the vertices of a graph, we need to use the appropriate volume representing $\mathrm{d}\mathbf{q}$ for each discretized cell. In particular, a uniform discretization of the coordinate space implies we use $\sqrt{\det(\eta_{\bullet\bullet}(\mathbf{q}))}$ (up to a constant scalar multiple) for volume of each discretized cell.

Once we have the gradient of $\tilde{\mathscr{H}}$, the control law for minimizing $\tilde{\mathscr{H}}$ would simply be for k^{th} robot to move in a direction opposite to the gradient

$$\mathbf{u}_k = -\kappa \frac{\partial \tilde{\mathscr{H}}(P)}{\partial \mathbf{p}_k} \tag{7}$$

This give a generalized Lloyd's algorithm with guarantee of asymptotic stability (as easily seen by considering $\tilde{\mathscr{H}}$ a Liapunov function candidate, and noting that its domain is a manifold). In the final converged solution, each robot will be at the *generalized centroid* of its related Voronoi tessellation (since that's when the gradient of $\tilde{\mathscr{H}}$ vanishes).

With the assumption of isotropy of the metric in the given chart (which will hold in most of the exploration applications we will be describing), this further reduces to

$$\frac{\partial \tilde{\mathscr{H}}(P)}{\partial \mathbf{p}_k} \ = \ 2\ \sqrt{\zeta(\mathbf{p}_k)} \int_{V_k} d(\mathbf{q},\mathbf{p}_k)\ \ \mathbf{z}^T_{\mathbf{qp}_k}\ \phi(\mathbf{q})\ \ \mathrm{d}\mathbf{q} \tag{8}$$

Typically, since we will be computing the control commands in a discrete setup, we are mostly interested in the direction of $\mathbf{u_k}$ rather than its magnitude. Moreover, the metric in the given chart will be isotropic in most practical robot planning problems. Thus we clump the leading term $2\ \zeta(\mathbf{p}_k)$ of (8) inside κ to obtain the following control law

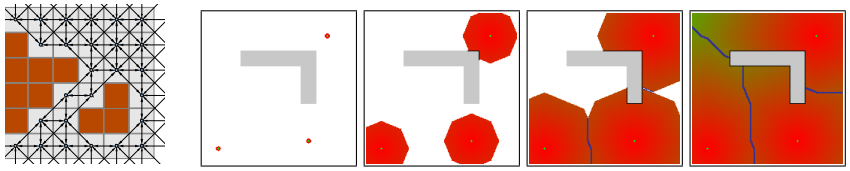
$$\mathbf{u}_k = -\kappa\ \int_{V_k} d(\mathbf{q},\mathbf{p}_k)\ \ \mathbf{z}_{\mathbf{qp}_k}\ \phi(\mathbf{q})\ \ \mathrm{d}\mathbf{q} \tag{9}$$

5 Graph Search-based Implementation

In order to develop a version of the generalized continuous-time Lloyd's algorithm for a general distance function, we first need to be able to compute the general Voronoi tessellation of Equation (2) for arbitrary distance function, d. We adopt a discrete graph-search based approach for achieving that, not very unlike the approach taken in previous work [2]. We consider a uniform square tiling (Figure 4(a)) of the space of coordinate variables (in a particular coordinate chart) and creating a graph G out of it (with vertex set $\mathscr{V}(G)$, edge set $\mathscr{E}(G) \subseteq \mathscr{V}(G) \times \mathscr{V}(G)$ and cost

function $\mathscr{C}_G : \mathscr{E}(G) \to \mathbb{R}^+$). The costs/weights of the edges of the graph are the metric lengths of the edges. It is to be noted that in doing so we end up restricting the metric of the original space to the discrete graph. Because of this, as well as due to the discrete computation of the integrations (as discussed later), this discrete graph-search based approach is inherently an approximate method, where we trade off the accuracy and elegance of a continuous space for efficiency and computability with arbitrary metric.

The key idea is to make a basic modification to Dijkstra's algorithm [9, 4]. This enables us to create a geodesic Voronoi tessellation. For creating Voronoi tessellations we initiate the *open set* with multiple start nodes from which we start propagation of the wavefronts. Thus the wavefronts emanate from multiple sources. The places where the wavefronts collide will hence represent the boundaries of the Voronoi tessellations. In addition, we can conveniently alter the distance function, the level-set of which represents the boundaries of the Voronoi tessellations. This enables us to even create *geodesic power Voronoi tessellation*. Figure 4(b) illustrates the progress of the algorithm in creation of the tessellations.



(a) Illustration of graph construction.

(b) Illustration of progress of the algorithm for tessellation and control computation in a 200×200 uniform square discretized environment. Frames show snapshots at iterations $100, 10100, 25100$ and 37300.

Fig. 4 (a): An 8-connected grid graph created from a uniformly discretized coordinate space. The brown cells represent obstacles. **(b)**: Progress of the algorithm for tessellation and control computation in an environment with a L-shaped obstacle. The graph is constructed by 200×200 uniform square discretization of the environment (see [2]). Tessellations are created starting from three points (the location of the agents) to the complete diagram after expansion of about 37300 vertices. The filled area indicates the set of *expanded* vertices. The boundaries of the tessellations are visible in blue.

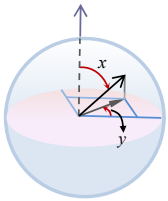
In order to compute the control command for the robots (*i.e.* the action of the robot in the next time step), we use the formula in Equation (9). In a general metric setup, the vector $\mathbf{z}_{\mathbf{q}\mathbf{p}_k}$ is the unit vector along the tangent at $\mathbf{p}_k$ to the geodesic joining $\mathbf{q}$ to $\mathbf{p}_k$. In a discretized setup, the position of the k^{th} robot corresponds to a vertex $p_k \in \mathscr{V}(G)$. Then, in the graph, the unit vectors $\mathbf{z}_{\mathbf{q}\mathbf{p}_k}$ is approximated as the unit vectors along edges of the form $[p'_k, p_k] \in \mathscr{E}(G)$ for some $p'_k \in \mathscr{N}_G(p_k)$ (the set of neighbors of p_k) such that the shortest path in the graph connecting p_k and q passes through p'_k. For a given q, we keep track of the index of the robot whose tessellation it belongs to, as well as the neighbor of the corresponding robot's vertex through which the shortest path leading to q passes. Thus, we can also compute the integration of (9) on the fly as we compute the tessellations. The complete pseudocode for the algorithm can be found in [1].

The complexity of the algorithm is the same as the standard Dijkrstra's algorithm, which for a constant degree graph is $O(V_G \log(V_G))$ (where $V_G = |\mathscr{V}(G)|$ is the

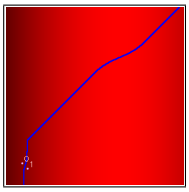
number of vertices in the graph). This is in sharp contrast to the complexity of search for optimal location as in [10].

5.1 Application to Coverage on Non-Euclidean Metric Spaces

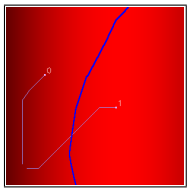
In this section we will illustrate examples of coverage using the generalized continuous-time Lloyd's algorithm on a 2-sphere. We use a coordinate chart with coordinate variables $x \in (0,\pi)$, the latitudinal angle, and $y \in [0,2\pi)$, the longitudinal angle (Figure 5(a)). The matrix representation of the metric on the sphere using this coordinate chart is $\eta_{\bullet\bullet} = \begin{bmatrix} 1 & 0 \\ 0 & \sin^2(x) \end{bmatrix}$. As usual, we use a uniform square discretization of the coordinate space to create an 8-connected grid graph [2]. However, in order to model the complete sphere (in the example of Figure 6), we need to establish appropriate edges between vertices at the extreme values of y, *i.e.* the ones near $y = 0$ and $y = 2\pi$. Similarly, we use an additional vertex for each pole to which the vertices corresponding to the extreme values of x connect.



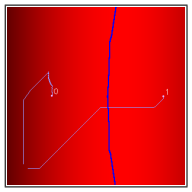
(a) The 2-sphere and a coordinate chart on it.



(b) $t = 1$.



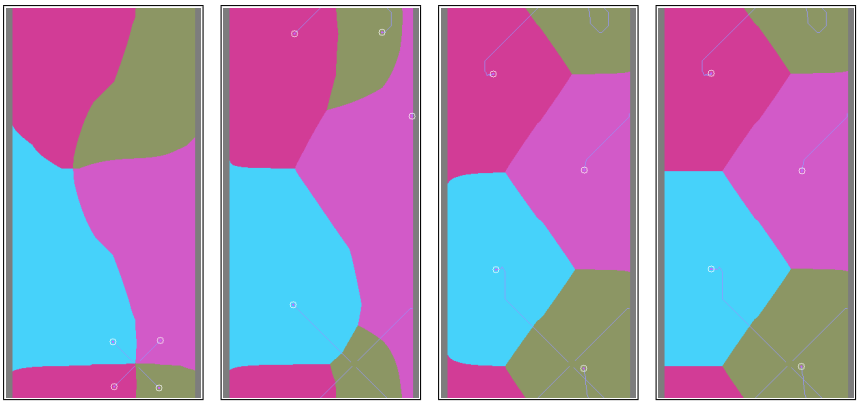
(c) $t = 150$.



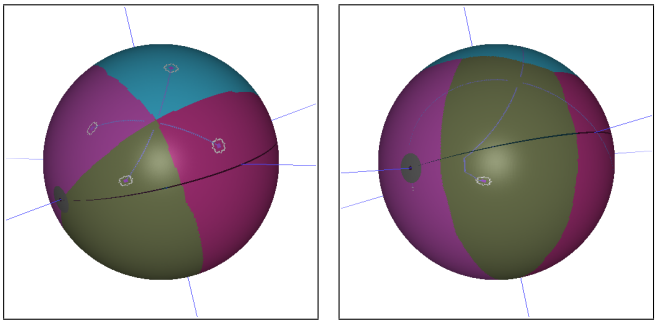
(d) $t = 250$. Converged solution.

Fig. 5 Coverage using discrete implementation of generalized continuous-time Lloyd's algorithm on a part of the 2-sphere. The chosen coordinate variables, x and y, are the latitudinal and longitudinal angles respectively, and the domain shown in figures (b)-(d) represent the region on the sphere where $x \in [\pi/16, 3\pi/4], y \in [\pi/16, 3\pi/4]$. x is plotted along horizontal axis and y along vertical axis on linear scales. The intensity of red indicates the determinant of the round metric, thick blue curves are the tessellation boundaries, and the thin pale blue curves are the robot trajectories.

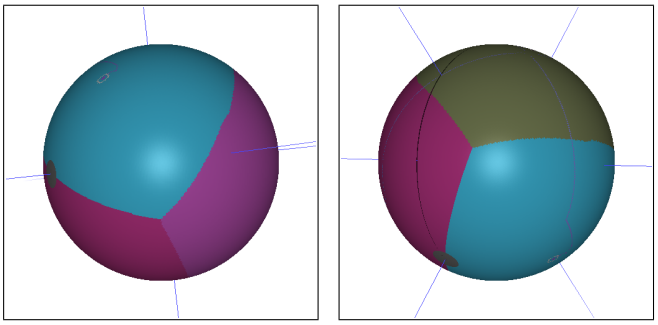
Figure 5(b)-(d) shows two robot attaining coverage on a geodesically convex subset of the 2-sphere by following the control command computed using the algorithm of Section 5 at every time-step. The region of the sphere that we restrict to is that of latitudinal angle $x \in [\pi/16, 3\pi/4]$, and longitudinal angle $y \in [\pi/16, 3\pi/4]$. The robots start off from the bottom left of the environment near the point $[0.42, 0.45]$, and follow the control law of Equation (7) until convergence is achieved. Note that the tessellations have a curved boundary in Figures 5(c) and 5(d) because it has to be a segment of the great circle on the sphere (note that the jaggedness is due to the fact that the curve actually resides on the discrete graph rather than the original metric space of the sphere). In the converged solution of Figure 5(d), note how the robots get placed such that their tessellation split up the area on the sphere equally rather than splitting up the area of the non-isometric embedding that depends on the chosen coordinate chart. The weight function is chosen to be constant, $\phi(\mathbf{q}) = 1$. For this example, the program ran at a rate of about 4 Hz on a machine with 2.1 GHz processor and 3 Gb memory.



(a) $t = 32$. (b) $t = 88$. (c) $t = 171$. (d) $t = 211$.



(e) $t = 32$. (f) $t = 88$.



(g) $t = 171$. (h) $t = 211$.

Fig. 6 Coverage on a complete sphere. In Figures (a)-(d), $x \in (0, \pi)$ (latitudinal angle) is plotted along horizontal axis and $y \in [0, 2\pi)$ (longitudinal angle) along vertical axis on linear scales. Figures (e)-(h) show the same plot mapped on the 2-sphere. The colors are used to indicate the tessellation of each robot. Note that in (e)-(h) different viewing angles are used to view the interesting parts of the sphere at a particular time-step. (d) and (h) are the converged solutions.

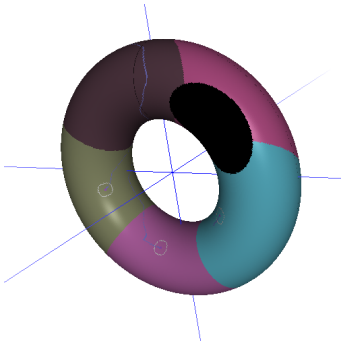


Fig. 7 Coverage on a 2-torus with obstacle on it (marked in black) attained by 5 robots.

A more complete example is shown in Figure 6 in which 4 robots attain coverage on a complete 2-sphere. The robots start off close to each other on the sphere, and follow the control law of Equation (7), until they converge attaining good and uniform coverage of the sphere. In order to avoid numerical problems near the poles, we 'chop off' small disks near the poles (marked by the gray regions), and establish 'invisible' edges across those disks connecting the vertices on their diametrically opposite points. Figures 6(a)-(d) show the tessellations of the robots in the coordinate chart with x plotted along the horizontal axis and y along the vertical axis (300×600 uniformly discretized). Figures 6(e)-(h) show the same tessellations mapped on the sphere. The weight function, once again is chosen to be $\phi(\mathbf{q}) = 1$. For this example, the program ran (control computation as well as plotting of the graphics) at a rate of about 1 Hz on a machine with 2.1 GHz processor and 3 Gb memory.

Figure 7 shows the final converged solution of a similar example of coverage on a 2-torus with obstacles by 5 robots. The metric tensor on the torus is given by $\eta_{\bullet\bullet} = \begin{bmatrix} r^2 & 0 \\ 0 & (R+r\cos x)^2 \end{bmatrix}$, where R is the radius of the axial circle, r the radius of the *tube* of the torus, x is the *latitudinal* angle, and y is the *longitudinal* angle. Note how the Voronoi tessellations in this case are not simply connected.

5.2 Application to Cooperative Exploration and Coverage Problem

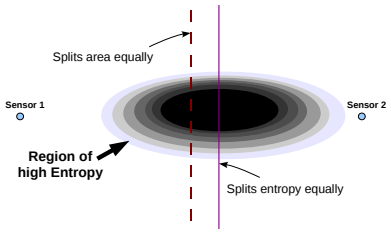


Fig. 8 Entropy-weighted metric for voronoi tessellation in exploration problem.

Next we apply the tools developed to the problem of cooperative exploration and simultaneous coverage. In previous work [2] we had used an approximate and ad hoc "projection of centroid" method in order to compute an analog of *generalized centroid* and device a control law that was essentially to follow the approximate

generalized centroid. However, now that we are equipped with the control law of Equation (7), we can achieve the same objectives in a more systematic way.

As detailed in [2], we choose Shannon entropy for constructing the density function as well as to weigh the metric (Figure 8). Thus, if $p(\mathbf{q})$ is the probability that the vertex $\mathbf{q}$ is inaccessible (*i.e.* occupied or part of an obstacle), for all $\mathbf{q} \in \mathscr{V}(G)$, then the Shannon entropy is given by $e(\mathbf{q}) = -\left(p(\mathbf{q})\ln(p(\mathbf{q})) + (1-p(\mathbf{q}))\ln(1-p(\mathbf{q}))\right)$. We use this for modeling the weight function, ϕ, and an isotropic metric, ζI (where I is the identity matrix). Noting that in an exploration problem, the occupancy probability, p, and hence the entropy e, will be functions of time as well, we use the following formulae for ϕ and ζ,

$$\phi(\mathbf{q},t) = \begin{cases} \varepsilon_\phi, \text{ if } e(\mathbf{q},t) < \tau \\ e(\mathbf{q},t), \text{ otherwise.} \end{cases}, \qquad \zeta(\mathbf{q},t) = \begin{cases} \varepsilon_\zeta, \text{ if } e(\mathbf{q},t) < \tau \\ e(\mathbf{q},t), \text{ otherwise.} \end{cases} \tag{10}$$

for some small ε_ϕ and ε_ζ representing zero (for numerical stability).

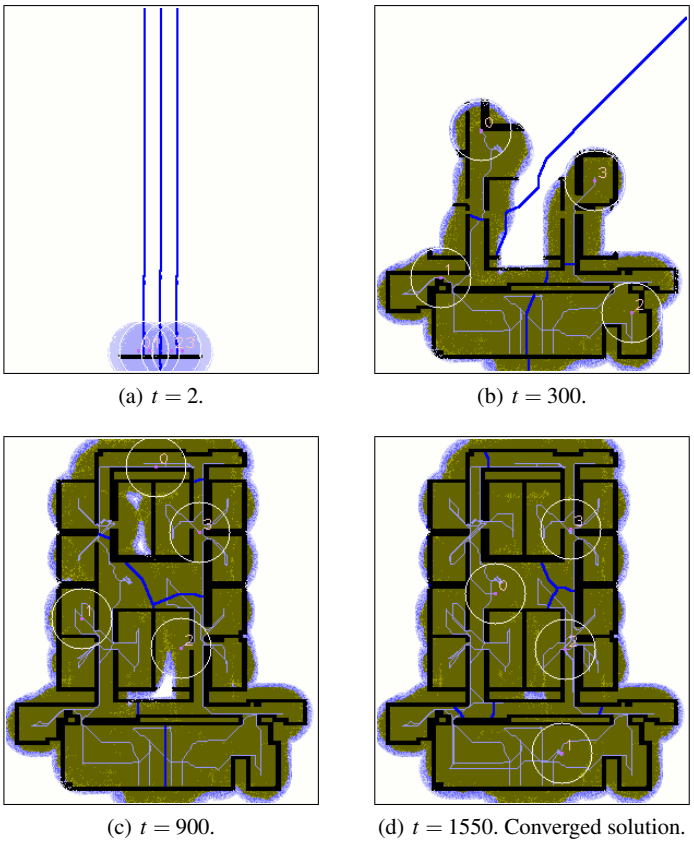
Each mobile robot maintains, updates and communicates a probability map for the discretized environment and updates its entropy map. We use a sensor model similar to that described in [2], as well as 'freeze' a vertex to prevent any change to its probability value when its entropy drops below some τ' $(< \tau)$.

In addition, to avoid situations where a robot gets stuck at a local minima inside its tessellation even when there are vertices with entropy greater than τ in the tessellations (this can happen when there are multiple high entropy regions in the tessellations that exert equal and opposite pull on the robot so that the net velocity becomes zero), we perform a check on the value of the integral of the weight function, ϕ, within the tessellation of the k^{th} robot when its control velocity vanishes. If the integral is above the value of $\int_{V_k} \varepsilon_\phi \, \mathrm{d}\mathbf{q}$, we switch to a greedy exploration mode where the k^{th} robot essential head directly towards the closest point that has entropy greater than the value of τ. This ensures exploration of the entire environment (*i.e.* the entropy value for every accessible vertex drops below τ). And once that is achieved, both ϕ and ζ become independent of time. Thus convergence is guaranteed.

Figure 9 shows screenshots of a team of 4 robots exploring a part of the 4^{th} floor of the Levine building at the University of Pennsylvania. The intensity of white represents the value of entropy. Thus in Figure 9(a) the robots start of with absolutely no knowledge of the environment, explore the environment, and finally converge to a configuration attaining good coverage (Figure 9(d)).

Figure 10 shows a similar scenario. However, in this case one of the robots (Robot 0, marked by red circle) gets hijacked and manually controlled by a human user soon after they start cooperative exploration of the environment. That robot is forced to stay inside the larger room at the bottom of the environment. Moreover, in this case we use a team of heterogeneous robots (robots with different sensor footprint radii), thus requiring to compute *power voronoi tessellations*. This simple example illustrates the flexibility of our framework with respect to human-robot interaction.

For either of the above examples, with the environment 284×333 discretized, the program ran at a rate of about $3-4$ Hz on a machine with 2.1 GHz processor and 3 Gb memory.



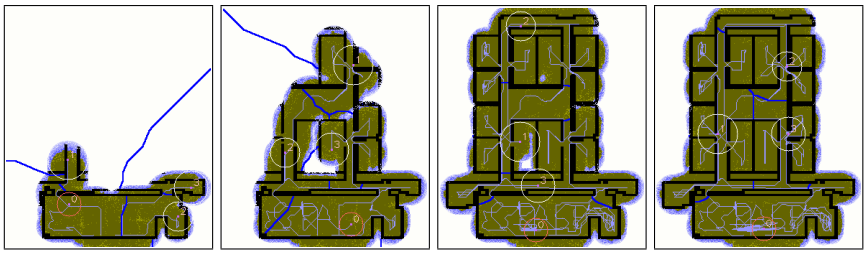
(a) $t = 2$. (b) $t = 300$.

(c) $t = 900$. (d) $t = 1550$. Converged solution.

Fig. 9 Exploration and coverage of an office environment by a team of 4 robots. Blue curves indicate boundaries of tessellations, intensity of white indicates the value of entropy.

6 Conclusion

In this paper we have extended the coverage control algorithm proposed by *Cortes, et al.* [5], to non Euclidean configuration spaces that are, in general, non convex and equipped with metrics that are not Euclidean. The key idea is the transformation of the problem of computing gradients of distance functions to one of computing tangents to geodesics. We have shown that this simplification allows us to implement our coverage control algorithm in any space after reducing it to a discrete graph. We have illustrated the algorithm by considering multiple robots achieving uniform coverage on a 2-sphere and an indoor environment with walls and obstacles. We have also shown the application of the basic ideas to the problem of multi-robot cooperative exploration of unknown or partially known environments.



(a) $t = 150$. (b) $t = 500$. (c) $t = 1450$. (d) $t = 2100$. Converged solution.

Fig. 10 Exploration and coverage of an office environment (284×333 discretized) by a team of 4 robots, with one of the robots (marked by red circle) being controlled by a human user.

References

1. S. Bhattacharya, R. Ghrist, and V. Kumar. Relationship between gradient of distance functions and tangents to geodesics. Technical report, University of Pennsylvania. See http://subhrajit.net/wiki/index.php?SFile=DistanceGradient.
2. S. Bhattacharya, N. Michael, and V. Kumar. Distributed coverage and exploration in unknown nonconvex environments. In *Proceedings of the 10th International Symposium on Distributed Autonomous Robotics Systems*, pages 1–14. Springer, 2010.
3. F. Bullo, J. Cortés, and S. Martínez. *Distributed Control of Robotic Networks: A Mathematical Approach to Motion Coordination Algorithms*. Applied Mathematics Series. Princeton University Press, 2009.
4. Thomas H. Cormen, Clifford Stein, Ronald L. Rivest, and Charles E. Leiserson. *Introduction to Algorithms*. McGraw-Hill Higher Education, 2001.
5. J. Cortes, S. Martinez, T. Karatas, and F. Bullo. Coverage control for mobile sensing networks. *IEEE Trans. Robot. Autom.*, 20(2):243–255, April 2004.
6. Jorge Cortez, S. Martinez, and Francesco Bullo. Spatially-distributed coverage optimization and control with limited-range interactions. *ESIAM: Control, Optimisation and Calculus of Variations*, 11:691–719, 2005.
7. Jorge Cortez, Sonia Martinez, Timur Karatas, and Francesco Bullo. Coverage control for mobile sensing networks. *IEEE Trans. Robot. and Automat.*, 20(2):243–255, 2004.
8. H. Cundy and A. Rollett. *Mathematical Models*. Tarquin Pub., 3rd edition, 1989.
9. Edsger W. Dijkstra. A note on two problems in connexion with graphs. *Numerische Mathematik*, 1:269–271, 1959.
10. J.W. Durham, R. Carli, P. Frasca, and F. Bullo. Discrete partitioning and coverage control for gossiping robots. *IEEE Transactions on Robotics*, 28(2):364–378, 2012.
11. M. Gromov, J. Lafontaine, and P. Pansu. *Metric structures for Riemannian and non-Riemannian spaces*. Progress in mathematics. Birkhäuser, 1999.
12. Stuart P. Lloyd. Least squares quantization in PCM. *IEEE Trans. Inf. Theory*, 28:129–137, 1982.
13. 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*, pages 3947–3952, Cancun, Mexico, December 2008.
14. C. Stachniss. *Exploration and Mapping with Mobile Robots*. PhD thesis, University of Freiburg, Freiburg, Germany, April 2006.
15. C. Stachniss, G. Grisetti, and W. Burgard. Information gain-based exploration using rao-blackwellized particle filters. In *Proc. of Robot.: Sci. and Syst.*, pages 65–72, Cambridge, MA, June 2005.
16. Sebastian Thrun, Wolfram Burgard, and Dieter Fox. *Probabilistic Robotics (Intelligent Robotics and Autonomous Agents)*. The MIT Press, 2005.