Conference Paper

DisCoverage for non-convex environments with arbitrary obstacles

Author(s): Haumann, Dominik; Breitenmoser, Andreas; Willert, Volker; Listmann, Kim; Siegwart, Roland Y.

Publication Date: 2011

Permanent Link: https://doi.org/10.3929/ethz-a-010025258

Originally published in: http://doi.org/10.1109/ICRA.2011.5980415

Rights / License: In Copyright - Non-Commercial Use Permitted
DisCoverage for non-convex environments with arbitrary obstacles

Dominik Haumann, Andreas Breitenmoser, Volker Willert, Kim Listmann and Roland Siegwart

Abstract—DisCoverage is a distributed strategy for frontier-based multi-robot exploration. The robots coordinate by a partition of the environment, and choose their target points by optimizing a locally decomposable objective function. In [9] DisCoverage for convex regions was proposed. In this work, we extend DisCoverage to support arbitrary non-convex real-world environments with obstacles. Therefore, we introduce a transformation of non-convex environments to robot-centric star-shaped domains. This results in a general solution with broader applications for exploration and path planning. Simulations as well as experiments with real robots demonstrate the exploration progress.

I. INTRODUCTION

The exploration task is a well-known problem in cooperative robotics [17]. The goal is to map an unknown environment with multiple robots. The applications are diverse and include search & rescue missions, such as the search for survivors after a catastrophe in environments inaccessible to humans, planetary surface exploration or the exploration of underground mines.

We limit our attention to frontier-based exploration strategies [18]. In frontier-based exploration, a frontier is the boundary separating known (already explored) space from unknown (still to be explored) one. As long as a frontier exists, the robot moves to a location on the frontier and maps the new environment, which in turn pushes back the frontier. Repetition of this scheme eventually leads to a fully explored environment.

Although the frontier-based approach was initially introduced for a single robot, it can easily be extended to multiple robots: i) Each robot chooses a target point by optimizing an objective function, ii) through communication the robots coordinate and prevent each other from moving to the same location, iii) each robot plans a path to reach the target point and finally, iv) each robot moves to the target point to continue exploration.

In practice, the realization is non-trivial for two reasons. First, choosing the next target point is vital in the overall performance of the exploration strategy. Second, planning a collision-free path to reach the destination on the frontier is a key problem in non-convex environments.

A strategy for cooperative multi-robot exploration was recently proposed [9], namely the DisCoverage algorithm. This approach uses a Voronoi partition to assign different parts of the environment to each robot. Based on the information available in the respective partition, each robot then autonomously optimizes the objective function to derive control laws. These control laws determine where the robots move next to reach the frontier and hence, extend the map (Fig. 1). However, so far the DisCoverage approach only supports convex, obstacle-free environments.

In this paper, we introduce a new method that transforms any non-convex environment with arbitrary obstacles to a star-shaped domain. We apply this transformation to DisCoverage, to extend the algorithm for exploration of arbitrary non-convex environments. Further, we investigate the properties of the DisCoverage algorithm with a group of e-puck robots in experiments, and compare it to a state-of-the-art approach for exploration.

The paper is organized as follows: The next section identifies core features of existing exploration strategies and classifies related work. Section III provides preliminaries and a formal problem representation for the DisCoverage algorithm for non-convex environments. We propose the transformation of arbitrary environments to star-shaped domains in Section IV and present simulation results as well as real-world experiments in Section V. We conclude in Section VI.

II. RELATED WORK

All frontier-based exploration strategies can be formulated as an optimization problem. Then, selecting appropriate target points becomes equivalent to finding the optimum of an objective function, that represents the intended exploration behavior [7]. We differentiate between two classes to reach the target points: either the motion control laws need to be calculated in an additional path planning step, or are contained in the optimization procedure, e.g. by using potential-field methods [12]. We first investigate related work that uses an additional path planning step.

In [18] the target point is set to the point on the frontier that is closest to the robot, i.e., the travel distance is minimized. We refer to this as the MinDist approach. A further, more general approach realizes the idea of maximizing the expected information gain, i.e., the knowledge acquired each time a frontier is reached. In [4] the objective function consists of distance costs and the amount of frontiers in the circle with a specific radius around the target point.
Hence, the extent of the frontier is measured. In [15], [16] the expected information gain is defined as the amount of unknown space visible by the sensor at the target point.

All these approaches first determine the target point and then plan a collision-free path to reach the destination in non-convex environments. Explicit path planning allows fine-grained control over the robot movement [12]. However, for the remainder of this paper, we are interested to determine to what extent it is possible to use implicit path planning methods in order to reach the target points.

The idea of finding control laws that contain the path planning step in the optimization process is visualized in Fig. 1, and was initially proposed in [6] to solve the coverage problem. Here, the objective is to place multiple robots in the environment such that they maximize the covered area. This problem. Here, the objective is to place multiple robots in the environment such that they maximize the covered area. This problem.

The solution to the coverage problem in [6] only works for convex domains. An extension to non-convex regions with convex holes by using a diffeomorphism was proposed in [5]. However, the diffeomorphism distorts the original set such that the neighborhood of the robot possibly changes. In [3] the centroid of the Voronoi partition is projected to the boundary of obstacles in case it lies inside an obstacle. However, additional path planning is required to reach the centroid in the non-convex environment. Further concepts for dealing with non-convex environments are introduced in [8], [14]. In [14], the geodesic distance is split into linear path segments. Then, the direction along the first path segment together with the geodesic distance is used in the optimization to find control laws for solving the coverage problem.

The DisCoverage algorithm [9] for multi-robot exploration is inspired by the solution to the coverage problem [6] and works according to the feedback loop in Fig. 1. To achieve exploration instead of coverage, the objective function is changed such that the optimization only considers the frontier instead of the entire area in the Voronoi cell. However, the DisCoverage algorithm only works for convex, obstacle-free environments, as no information about obstacles is included in the optimization. To circumvent this drawback, we want to include the idea of using the geodesic distance as well as the first path segment from [14].

Before we explain how to extend the DisCoverage algorithm to non-convex environments the next section provides preliminaries and a formal problem representation.

III. PRELIMINARIES AND PROBLEM FORMULATION
A. Convex DisCoverage

For motivation, we first restate the DisCoverage algorithm proposed in [9]. An omni-directional sensor model and ideal measurements are assumed, i.e. uncertainty in the robot positions and the sensor readings are neglected. Additionally, ideal and bidirectional communication between Voronoi neighbors is given, so that robots of adjoining Voronoi cells communicate each other’s position and synchronize the map at any time. Further, only convex, obstacle-free environments $\mathcal{Q} \subseteq \mathbb{R}^2$ are considered with $\partial \mathcal{Q}$ as boundary. $\mathcal{Q}$ is a compact set.

The goal is to find control laws for the robot dynamics

$$\dot{p}_i = u_i,$$

where $p_i \in \mathbb{R}^2$ and $u_i$ represent the robot position and the control input of robot $i$.

The solution found in [9] works as follows: Let $\mathcal{P} = \{p_1, \ldots, p_N\}$ be the configuration of $N$ robots with robot positions $p_i$. Let $\delta_i \in [-\pi, \pi]$ be the orientation of robot $i$ and $\Delta = \{\delta_1, \ldots, \delta_N\}$ the set of all orientations. Then, $\mathcal{V}_i = \{q \in \mathcal{Q} \mid \|q - p_i\| \leq \|q - p_j\|, \forall j\}$ is the Voronoi cell of robot $i$ and $\mathcal{V} = \{\mathcal{V}_1, \ldots, \mathcal{V}_N\}$ the Voronoi partition of $\mathcal{Q}$. Robots with adjoining Voronoi cells are called neighbors. Define $\mathcal{S} \subseteq \mathcal{Q}$ as the explored space and let $\partial \mathcal{S}$ denote the frontiers. Let $\mathcal{S}_i = \mathcal{S} \cap \mathcal{V}_i$, i.e., $\mathcal{S} = \cup_i \mathcal{S}_i$ and define $\partial \mathcal{S}_i = \partial \mathcal{S} \cap \mathcal{V}_i \setminus \partial \mathcal{Q}$ as the frontiers in $\mathcal{V}_i$. The distributed optimization problem then reads

$$\mathcal{H}_{\text{discover}}(\mathcal{P}, \Delta) = \sum_{i=1}^{N} \mathcal{H}_{\text{discover},i}(p_i, \delta_i)$$

$$= \sum_{i=1}^{N} \int_{\partial \mathcal{S}_i} f(p_i, \delta_i, q) \phi(q) dq.$$  

(2)

The density function $\phi$ can be used to integrate prior knowledge of the environment. However, to focus on the
In short, we search for an orientation that optimizes the angular component of the robot orientation and the distance component as a distance dependent weighting.

Then control laws are given through continuous optimization of the orientation and read
\[ p_i = u_i = v \left( \frac{\cos \delta_i^*}{\sin \delta_i^*} \right) \]
with \( v \in \mathbb{R}^+ \) being a constant velocity for all robots. This again represents the feedback loop in Fig. 1, as the optimization of the orientations depends on the robot positions and the Voronoi partition, which change continuously in time.

### B. Non-Convex Regions

The convex DisCoverage algorithm is restricted to convex, obstacle-free environments. In order to support arbitrary non-convex environments with obstacles we need to develop a modified version of the convex DisCoverage algorithm. Hence, we formulate the problem as follows:

**Problem:** Given a configuration \( \mathcal{P} = \{p_1, \ldots, p_N\} \) of \( N \) robots in a non-convex environment \( \mathcal{Q} \subset \mathbb{R}^2 \) with arbitrary obstacles, find a partition and a modified version of (2) to derive control laws \( u_i \) such that \( S \rightarrow Q \) as \( t \rightarrow \infty \).

### IV. NON-CONVEX DISCOVERAGE

The objective function in (3) of the DisCoverage algorithm consists of the angular and the distance component. The angular component is responsible for finding optimal orientations, influenced by the distance dependent weighting. Both components assume visibility of the robot position and the frontiers \( q \) in the computation, which is true for convex environments.

**Definition 1:** Given a set \( \mathcal{M} \), let \( [u, v] \) be the closed segment denoted by \( [u, v] = \{ w \mid w = u + \lambda(v - u), \lambda \in [0, 1]\} \). Elements \( u, v \in \mathcal{M} \) are visible if \( [u, v] \subseteq \mathcal{M} \).

Convex regions unconditionally satisfy visibility. Contrary, in non-convex environments with obstacles the boundary \( \partial \mathcal{Q} \) as well as the pose and type of the obstacles have to be taken into account. There are two ways to address this problem.

Either we find a different objective function that works for non-convex environments by taking path planning into account. Or we first find a transformation for the environment and keep the objective function. Here, we transform the frontier such that we do not have to modify the optimization problem.

#### A. Finding a Transformation

The idea is to find a transformation with the property that all frontiers \( q \) are visible from the robot position \( p_i \). This property holds for star-shaped domains.

**Definition 2:** A domain \( S^* \subset \mathbb{R}^2 \) is called *star-shaped* if there exists \( u \in S^* \) such that \( [u, v] \subseteq S^* \) for all \( v \in S^* \).

Next, we introduce a transformation that maps any connected domains to a star-shaped domain as follows.

**Definition 3:** Let \( S_i \subseteq \mathcal{Q} \subset \mathbb{R}^2 \) and \( S_i^* \subset \mathbb{R}^2 \). Let \( d_q(p_i, q) \) be the geodesic distance from the robot position \( p_i \) to \( q \in S_i \). Let \( e(p_i, q) \) be the unit vector pointing into the direction of the first path segment of the geodesic path from \( p_i \) to \( q \). Then, the transformation is given by \( T_{p_i} : S_i \rightarrow S_i^* \).

**Theorem 1:** The map \( T_{p_i} \) transforms any connected set to a star-shaped domain with respect to \( p_i \).

**Proof:** Follows from definitions 2 and 3. The map \( T_{p_i} \) can be interpreted as a straightening of the geodesic path. Note, that \( T_{p_i} \) is not bijective, i.e., it may map arbitrary many elements from \( S_i \) to only one element in \( S_i^* \). A geometric interpretation is given in Fig. 3.

A significant property of \( T_{p_i} \) is that elements \( q \in S_i \) visible from \( p_i \) remain unchanged, i.e., if the closed segment \( [p_i, q] \subseteq S_i \Rightarrow T_{p_i}(q) = q \). This is due to the fact that the geodesic path reduces to the Euclidean norm in convex environments. This implies that convex environments reflect a sub-domain of non-convex environments and that they are always star-shaped for any point within the convex region.

Further, the star-shaped environment \( S_i^* \) (Fig. 3(b)), since there exists no upper bound for the length of the geodesic distance.

It turns out that the proposed transformation realizes the same idea that was presented by Pimenta et al. [14] in order to extend the solution to the coverage problem in [6] to non-convex environments. However, the formulation as a transformation \( T_{p_i} \) introduced above provides a more detailed explanation as well as a geometric interpretation of
how the environment is mapped to a star-shaped domain. Thus, the transformation allows to extend any frontier-based exploration strategy that works for convex environments to also work in arbitrary non-convex environments.

B. Applying the Transformation to DisCoverage

We first consider the partition of the environment for multi-robot exploration. The convex DisCoverage algorithm uses the Voronoi partition with the Euclidean norm to partition the environment. This is reasonable, as each robot reaches the frontiers in its cell faster than any other robot. In non-convex environments this property does not hold. Instead, we use the geodesic distance to create the geodesic Voronoi partition \( V^g = \{ V^g_1, \ldots, V^g_N \} \) with the geodesic Voronoi cell \( V^g_i = \{ q \in Q \mid d^g_i(p_i, q) \leq d^g_j(p_i, q), \forall j \} \) of robot \( i \). Analog to the convex case, robots with adjoining geodesic Voronoi cells are called neighbors. As before, define \( S^g \subseteq Q \) as the explored space and let \( \partial S^g \) denote the frontiers. Let \( S^g_i = S^g \cap V^g_i \), i.e., \( S^g = \cup_i S^g_i \) and define \( \partial S^g = \partial S^g \cap \bigcup_i V^g_i \) as the frontiers in \( V^g_i \).

Communicating neighbor positions, each robot calculates its geodesic Voronoi cell, e.g. by following [2]. The distributed optimization problem for each robot \( i \) then is given by

\[
H_{\text{discover},i}(p_i, \delta) = \int_{\partial S^g_i} f(p_i, \delta, q^* \phi(q^*) dq^* \tag{6}
\]

with \( q^* = T_p(q) \). Compared to (2), the set \( \partial S_i \) and the frontiers \( q \) are now replaced by \( \partial S^g_i \) and \( q^* \). The robot dynamics (5) and the continuous computation of optimal orientations (4) remain unchanged.

Through the transformation \( T_p \), the optimization of the objective function now includes the path planning by calculating the geodesic distance. Including path information is inevitable in order to find control laws that solve the exploration problem for non-convex environments, since otherwise obstacles and the boundary of the environment are neglected [11].

C. Relation to the MinDist Approach

In the MinDist approach for non-convex environments, the robot determines the nearest frontier by computing the geodesic path and follows this path until it reaches the frontier to extend the map. Thus, a single point on the frontier determines the final path of the robot.

The DisCoverage algorithm for non-convex environments works similarly, as it includes the geodesic path to the frontiers as well. However, the fundamental difference is that all frontiers have an impact on the final path. Since the optimization of the objective function yields orientations with as many frontiers as possible located directly in front of the robot orientation, the behavior is inherently more greedy compared to the MinDist approach.

As final remark, MinDist is contained in DisCoverage, as DisCoverage has exactly the same behavior if only the nearest frontier is considered in the objective function.

V. Results

A. Simulation of Single-Robot Exploration

We developed a C++ simulator\(^1\) for single-robot exploration to compare the DisCoverage and MinDist algorithm. The discrete DisCoverage control law is implemented as

\[
p^{(k+1)} = p^{(k)} + v \left( \frac{\cos \delta^{(k)}}{\sin \delta^{(k)}} \right), \tag{7}
\]

with the discrete time index \( k \) and the constant velocity \( v \in \mathbb{R} \). Analog to the convex case [9], the optimal orientation is given by

\[
\delta^{(k)} = \arg \max_{\delta} H^*_{\text{discover}}(p^{(k)}, \delta^{(k-1)}). \tag{8}
\]

The robot movement in (7) includes (6) through the optimization. The geodesic distance to the frontiers is computed with the Dijkstra algorithm [12] based on an occupancy grid map. Cells either are marked as frontiers (orange), explored (white) or unexplored (gray).

In the simulation we automatically adapt the standard deviation \( \sigma \) of the distance component by setting it to the length of the shortest path to the closest frontier. This avoids numerical issues when the nearest frontiers are far away.

In the MinDist approach, the control law is given by \( p^{(k+1)} = p^{(k)} + v e(p^{(k)}, q) \), where \( q \) is the frontier with minimal distance and \( e \) the direction of the geodesic path as introduced in Section IV.

In Fig. 4 the robot explores an area of \( 20 \text{m} \times 10 \text{m} \). The grid resolution is \( 0.1 \text{m} \times 0.1 \text{m} \). The robot has a vision radius of \( r = 2 \text{m} \) and a travel speed of \( v = 10 \text{cm/s} \). Fig. 4(a) and 4(b) visualize the robot navigation with the DisCoverage and the MinDist approach. In Fig. 4(c) the amount of explored area for each iteration is plotted for both approaches. Additionally, the theoretical optimum [7] is given as a lower bound for the time needed to explore the entire environment. Here, the robot explores the area \( A = 2vr \) each iteration.

Initially, both approaches perform equally in Fig. 4. However, toward the end of the exploration process, DisCoverage performs better. The reason for this is that MinDist plans the path based on a single frontier cell. Missed frontier cells in the corner of a room lead to larger travel distances during the exploration. The DisCoverage approach sometimes omits corners as well, but given the more greedy behavior focuses on promising directions. This behavior is responsible for the robot moving in the middle of the corridor, while with MinDist the robot often moves very close to walls. Both approaches lack behind of the theoretical optimum because of the corridor and the walls. Hence, the slope of the progress of the DisCoverage and the MinDist approach get close to the slope of the optimal case only in open environments.

The simulation run verifies, that the transformation \( T_p \) works for arbitrarily complex environments. Thus, the simulator helps to test exploration strategies in complex, non-convex environments with arbitrary obstacles. However, the

\(^1\)Available at http://gitorious.org/discoverage
performance of exploration strategies heavily depends on the sensing capabilities, initial robot positions and the environmental conditions. Hence, the single simulation run in Fig. 4 does not qualify as statistical survey. For detailed surveys, we refer to [7] and [1]. Nevertheless, the simulations help to find significant properties of exploration strategies such as the relation of DisCoverage to MinDist (Section IV-C).

B. Experimental Results

In addition to the simulation, we provide experimental results\(^2\) using two e-puck robots [13]. The robots explore an office like interior that is projected on a square area of 1m × 1m (represented by the black overlay in Fig. 5). Each robot carries a marker which is tracked by an overhead camera using the ARToolkit software [10]. The DisCoverage algorithm recompute the moving directions and speeds of the robots based on the robots’ current positions obtained from the tracker input. The control commands are sent back to the robots at each iteration over Bluetooth. The setup uses a central controller that directs the robots and runs Matlab in the loop.

On the left part of Fig. 5 an image sequence of the exploration is shown. The right half shows the Matlab representation of the shared grid map with a resolution of 0.025m × 0.025m. The map visualizes position and orientation information from the tracker input and the sensor range of 0.2m. Further, frontiers, explored and unexplored area are displayed in orange, white and gray respectively. In addition, it includes the planned discretized geodesic paths to the frontier cells assigned to each robot.

The implementation differs from the simulation as follows: The robots have finite physical size of diameter 7cm, which is considered in the algorithm by inflation of the obstacles. Note, that the inflation may influence the exploration behavior, as the robots approach different sets of frontier cells compared to the simulation and the distance to the frontier changes when navigating around the enlarged obstacles.

The robots move according to a bang-bang control, turning in place until their orientation matches a specified target orientation range, and then drive straight with velocity proportional to the distance to an intermediate goal point. Due to limitations in tracking speed of the system, noise is introduced in the system and the robots’ angular positions slightly under- or overshoot, which results in jagged robot paths. Given the noise, the DisCoverage algorithm appears to be robust in its implementation.

The sequences illustrate different stages in the exploration process. The robots start from an initial configuration shown in Fig. 5(a) and 5(b). In case all frontiers are assigned to a single robot, the other robot falls back to coverage, because no frontier exists in the cell. This happens between Fig. 5(c) and 5(e) due to the narrow passages of the chosen scenario. However, this is a temporary effect, as once the exploring robot reaches a junction, the frontier is reassigned among the robots. Thus, the second robot switches back to exploration as illustrated in Fig. 5(e) and 5(g).

After the entire environment is mapped, both robots fall back to cover the non-convex area. In contrast, the single-robot exploration in the previous section stops once the exploration is completed. Possibilities for fallback strategies include but are not limited to closer inspection of certain points in the map, assistance of another robot or monitoring of the area just explored. We choose the Voronoi coverage method for non-convex environments from [3] as fallback strategy. However, we use the Dijkstra algorithm for path planning. Distributed coverage guides the robots to the centroids of the Voronoi regions, which in our scenario leads to the final configuration in Fig. 5(i). The trajectories of the robots are shown in Fig. 5(j).

VI. Conclusion

This work extends the DisCoverage algorithm for distributed multi-robot exploration to support non-convex environments with arbitrary obstacles. This is achieved by reformulating an idea recently proposed in [14] as a robot centric

\(^2\)Videos of the experimental results accompany the paper.
transformation of the explored environment. We provide a geometric as well as a mathematical interpretation of the transformation and show that the transformation yields star-shaped regions with respect to the robot positions. Further, we note that the transformation is not limited to exploration or coverage, but rather can be seen as a general way of dealing with non-convex environments. Simulations as well as real-world experiments with e-puck robots demonstrate the exploration progress in non-convex environments.

In future work, we will investigate the exact conditions for convergence of the DisCoverage algorithm, since a proof is still missing. Next to this, we plan to add line-of-sight constraints as well as time asynchronous communication and noisy information.

REFERENCES