2010 IEEE International Conference on Robotics and Automation
Anchorage Convention District
May 3-8, 2010, Anchorage, Alaska, USA

DisCoverage: A new Paradigm for Multi-Robot Exploration

A. Dominik Haumann, Kim D. Listmann and Volker Willert

Abstract— **The main aspect in multi-robot exploration is the efficient coordination of a group of robots. Inspired by previous results on the coverage problem, we propose a novel, frontier-based approach for multi-robot exploration. This approach merges the step of choosing appropriate target points with the step of planning a collision-free path. This is achieved by optimizing an objective function consisting of distance and orientation costs as well as an estimated information gain. The optimization yields motion control laws directly solving the exploration task. Using a Voronoi partition of the environment ensures, that each robot autonomously creates and optimizes the objective function to obtain a collision-free path in a distributed fashion. Simulations demonstrate the effectiveness of our approach.**

I. Introduction

Within the last decade the robotics community extended single-robot problems to the multi-robot domain. Multi-robot exploration [2], [6], [12] is one of such extensions of the single-robot case [10], [13]. Central to successful exploration is the effective coordination of a group of robots. Essentially, the problem is to assign appropriate *target points* to each robot such that all robots explore different regions of the environment. To this end, existing exploration strategies use a frontier-based approach [13], i.e., all robots simply move to target points on the borderline separating explored from unknown area. Intuitively, the main aspects in existing exploration approaches are how to choose those target points on the frontiers and how to get there. Choosing the target points is implemented by optimizing an objective function that consists of different key ingredients. For instance, in [2] the objective function consists of the expected information gain at a point and the distance costs, while the objective function in [10] additionally incorporates the localization quality. However, after determining the target points it is not clear how to choose appropriate motion control laws in order to reach those locations.

Hence, we can summarize the procedure of existing exploration strategies as follows: i) Each robot chooses a target point by optimizing an objective function, ii) the chosen destination (or similar information) is communicated to the group to prevent other robots 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 this work, we present a new approach to multi-robot exploration, inspired by the solution to the *coverage problem* in [3]. There, the steps of path planning and finding appropriate target points are merged. This is achieved by gradient-based motion control laws [9]. To this end, the authors use a Voronoi partition [4] of the environment. The task of each robot is to maximize coverage in its own Voronoi cell. Maximizing coverage is achieved by the optimization of an objective function, which exclusively uses information available in a single Voronoi cell. The gradient of the objective function is distributively computed and then used in the motion control laws.

The exploration strategies [2], [10] use an objective function to choose the target points, but then require an extra step for collision-free path planning and motion control. The authors in [8] derive control laws directly from optimizing an objective function. But they do not partition the environment. Hence, they need an additional component for collision avoidance as well. Contrary, using a Voronoi partition inherently incorporates collision avoidance, because Voronoi cells are guaranteed to be convex.

Our approach to multi-robot exploration combines the aforementioned work and extends the coverage problem to facilitate multi-robot exploration. Therefore, we refer to this as *DisCoverage*. The basic idea of DisCoverage is illustrated in Fig. 1 and works as follows: By using the Voronoi partition each robot i autonomously creates and optimizes its objective function $\mathcal{H}_i$ to derive an appropriate control input vector $\boldsymbol{u}_i$. The control input vector changes the state vector $\boldsymbol{x}_i$ according to the considered robot dynamics. This in turn changes the Voronoi partition and the process begins anew.

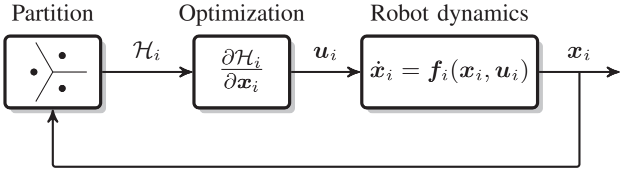

Partition
Optimization
Robot dynamics
$\mathcal{H}_i$
$\frac{\partial \mathcal{H}_i}{\partial \boldsymbol{x}_i}$
$\boldsymbol{u}_i$
$\dot{\boldsymbol{x}}_i = \boldsymbol{f}_i(\boldsymbol{x}_i, \boldsymbol{u}_i)$
$\boldsymbol{x}_i$

Fig. 1. Feedback loop of the exploration scheme

This procedure forms a closed loop and finding motion control laws resulting from the optimization step is of particular interest from a controls perspective. One of the most appealing properties of DisCoverage is that *the additional step of path planning and motion control can be skipped*. Another advantage is that our approach *inherently avoids collisions* as long as the robots remain in the assigned Voronoi cells.

The article is organized as follows: In the next section

All authors are with the Control Theory and Robotics Lab, TU Darmstadt, Landgraf-Georg-Str. 4, 64283 Darmstadt, GERMANY. {dhaumann, klistman, vwillert}@rtr.tu-darmstadt.de

This work was gratefully supported by the German Research Foundation (DFG) within the GRK 1362 "Cooperative, Adaptive and Responsive Monitoring of Mixed Mode Environments" (www.gkmm.tu-darmstadt.de)

978-1-4244-5040-4/10/$26.00 ©2010 IEEE

we formulate the problem properly and give necessary definitions and assumptions used throughout the paper. In Section III we derive the objective function and show how distributed optimization leads to appropriate motion control laws exploring new territory. In Section IV we provide computer simulations demonstrating the effectiveness of the proposed method and conclude the article in Section V.

II. Preliminaries and Problem Statement

To simplify the set-up, we assume *ideal measurements*, i.e., we neglect any uncertainty in the robot positions and the omni-directional sensor model. We assume *ideal, bidirectional communication* between Voronoi neighbors, so that robots of adjoining Voronoi cells communicate each other's position and synchronize the map at any time. Further, we consider a *convex*, *obstacle-free* environment $\mathcal{Q} \subset \mathbb{R}^2$.

We limit our attention to the simple dynamics

$$\dot{\boldsymbol{p}}_i = \boldsymbol{u}_i, \tag{1}$$

with the previously used state vector $\boldsymbol{p}_i = \boldsymbol{x}_i \in \mathbb{R}^2$, where $\boldsymbol{p}_i$ denotes the position and the vector $\boldsymbol{u}_i$ denotes the control input of robot i.

Let $\mathcal{P} = \{\boldsymbol{p}_1, \dots, \boldsymbol{p}_N\}$ be the configuration of N robots. Then, $\mathcal{V}_i = \{\boldsymbol{q} \in \mathcal{Q} \mid \|\boldsymbol{q} - \boldsymbol{p}_i\|_2 \leq \|\boldsymbol{q} - \boldsymbol{p}_j\|_2, \forall j\}$ is the Voronoi cell of robot i and $\mathcal{V} = \{\mathcal{V}_1, \dots, \mathcal{V}_N\}$ the Voronoi partition. Note that $\mathcal{Q} = \cup_i \mathcal{V}_i$, i.e., $\mathcal{V}$ is a partition of $\mathcal{Q}$. We refer to robots with adjoining Voronoi cells as *neighbors*. Similar to the solution of the coverage problem in [1], [3] we use an objective function of the form

$$\mathcal{H}_{\text{cover}}(\mathcal{P}) = \sum_{i=1}^{N} \mathcal{H}_{\text{cover},i}(\mathcal{P}) = \sum_{i=1}^{N} \int_{\mathcal{V}_i} f(\boldsymbol{q}, \boldsymbol{p}_i) \phi(\boldsymbol{q}) d\boldsymbol{q}, \tag{2}$$

where $\boldsymbol{q} \in \mathcal{V}_i$. We call $f(\cdot)$ *performance function* and $\phi(\boldsymbol{q})$ *density function*, which encodes any location dependent information gain at a point $\boldsymbol{q} \in \mathcal{Q}$. Our idea is to modify (2) such that optimization solves the exploration problem in a distributed manner. The fundamental difference to existing exploration strategies is that each robot is able to create and optimize (2) autonomously due to $\mathcal{H}_{\text{cover}}$ being a composition of N independent objective functions $\mathcal{H}_{\text{cover},i}$. Hence, all robots perform the optimization *in parallel* and not successively as e.g. in [2]. Distributed construction of a Voronoi cell is guaranteed as long as each robot is aware of the positions of its neighbors [3]. Hence, communication of robot positions and map synchronization between neighbors are the only requirements for our approach.

Focusing on exploration, we define $\mathcal{S} \subseteq \mathcal{Q}$ as the explored space with $\partial\mathcal{S}$ denoting the frontiers. Let $\mathcal{S}_i = \mathcal{S} \cap \mathcal{V}_i$, i.e. $\mathcal{S} = \cup_i \mathcal{S}_i$. Further, define $\partial\mathcal{S}_i = \partial\mathcal{S} \cap \mathcal{V}_i \setminus \partial\mathcal{Q}$ as the frontiers in $\mathcal{V}_i$. How to achieve exploration will be subject of the next sections, and we conclude with the definition of our problem as follows:

Given a configuration $\mathcal{P} = \{\boldsymbol{p}_1, \dots, \boldsymbol{p}_N\}$ of N robots in a convex, obstacle-free environment $\mathcal{Q} \subset \mathbb{R}^2$, find an objective function of the form (2) to derive control laws $\boldsymbol{u}_i$ such that $\mathcal{S} \to \mathcal{Q}$ as $t \to \infty$.

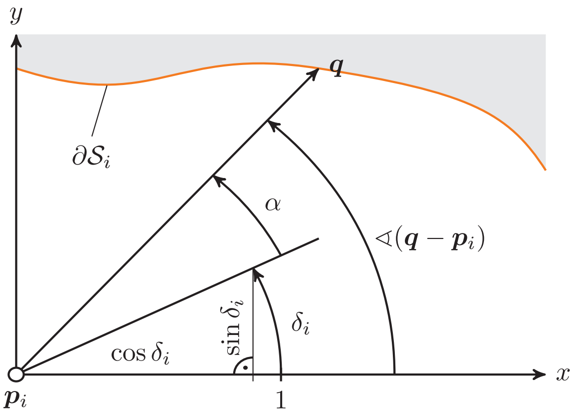


Fig. 2. Relation of the robot's orientation δ_i and angle α of the frontiers. The coordinate system in $\boldsymbol{p}_i$ is axis parallel to the global system and independent of the orientation δ_i.

III. Finding a Solution

A. Optimization Problem

Our goal is to find control laws $\boldsymbol{u}_i$ computable in a distributed manner such that each robot autonomously moves into a direction of unknown area in its Voronoi cell.

The significant property of the centroidal search in [3] is to cover all points as good as possible. Contrary, in the exploration process we want to focus on specific target points, where a robot will likely explore large parts of the environment, which is similar to the *utility of frontiers* in [2].

To find optimal target points, we first equip each robot with an orientation $\delta_i \in [-\pi, \pi]$, which defines the current moving direction of robot i. Inspired by the anisotropic sensor model in [7], where the performance depends on the distance *and* the orientation from the sensor to the target, we want to find orientations δ_i for each robot such that as many frontiers as possible are located directly in front of each robot. That is, we want to minimize the angle $\alpha \in [-\pi, \pi]$ between the orientations δ_i and all frontiers $\boldsymbol{q} \in \partial\mathcal{S}_i$ (Fig. 2) with

$$\alpha(\boldsymbol{p}_i, \delta_i, \boldsymbol{q}) = \underbrace{\sphericalangle(\boldsymbol{q} - \boldsymbol{p}_i) - \delta_i}_{\gamma} + \begin{cases} 2\pi & \text{if } \gamma < -\pi, \\ -2\pi & \text{if } \gamma > \pi, \\ 0 & \text{else.} \end{cases} \tag{3}$$

For instance, an angle of $\alpha = \pi$ implies that the considered point $\boldsymbol{q}$ lies behind $\boldsymbol{p}_i$, whereas an angle of $\alpha = 0$ implies that $\boldsymbol{q}$ lies in the direction of the robot's current orientation.

In contrast to [7], frontiers $\boldsymbol{q}$ do not necessarily lie within the maximum sensing range or maximum sensing direction. Hence, we need to find a different performance function. To this end, we introduce a continuous differentiable performance function f, which includes an angular component and a distance component, as follows:

$$f(\boldsymbol{p}_i, \delta_i, \boldsymbol{q}) = \underbrace{\exp\left(-\frac{\alpha(\boldsymbol{p}_i, \delta_i, \boldsymbol{q})^2}{2\theta^2}\right)}_{\text{angular component}} \underbrace{\exp\left(-\frac{\|\boldsymbol{q} - \boldsymbol{p}_i\|_2^2}{2\sigma^2}\right)}_{\text{distance component}}. \tag{4}$$

Note that the orientation δ_i represents an additional degree of freedom for the optimization and θ and σ describe the standard deviations of the Gaussians. We will refer to θ as *opening angle*. Small values of α lead to large values of the angular component, which reflects exactly the desired behavior described previously. The same holds for small values in the distance component, i.e., frontiers $\boldsymbol{q}$ closer to $\boldsymbol{p}_i$ imply a larger distance component.

Finally, with $\Delta = \{\delta_1, \delta_2, \dots, \delta_N\}$ we formulate a modified version of (2) given as

$$\begin{aligned}\mathcal{H}_{\text{discover}}(\mathcal{P}, \Delta) &= \sum_{i=1}^{N} \mathcal{H}_{\text{discover},i}(\boldsymbol{p}_i, \delta_i) \\ &= \sum_{i=1}^{N} \int_{\partial\mathcal{S}_i} f(\boldsymbol{p}_i, \delta_i, \boldsymbol{q})\phi(\boldsymbol{q})d\boldsymbol{q},\end{aligned} \tag{5}$$

with the aforementioned performance function f in (4). Note that (5) still is a composition of N objective functions, i.e., each robot can evaluate $\mathcal{H}_{\text{discover},i}$ autonomously.

Finding the optimal orientations δ_i^* is equivalent to the optimization problem

$$\delta_i^* = \arg\max_{\delta_i} \mathcal{H}_{\text{discover},i}(\boldsymbol{p}_i, \delta_i). \tag{6}$$

In δ_i^* the partial derivative of $\mathcal{H}_{\text{discover},i}$ with respect to δ_i vanishes, i.e., δ_i^* satisfies the necessary condition

$$\begin{aligned}&\frac{\partial \mathcal{H}_{\text{discover},i}(\boldsymbol{p}_i, \delta_i)}{\partial \delta_i} = \int_{\partial\mathcal{S}_i} \frac{\partial f}{\partial \alpha}\frac{\partial \alpha}{\partial \delta_i}\phi(\boldsymbol{q})d\boldsymbol{q} \\ &= \int_{\partial\mathcal{S}_i} -\frac{\alpha(\boldsymbol{p}_i, \delta_i, \boldsymbol{q})}{\theta^2} f(\boldsymbol{p}_i, \delta_i, \boldsymbol{q})\phi(\boldsymbol{q})d\boldsymbol{q} \overset{!}{=} \boldsymbol{0}.\end{aligned} \tag{7}$$

Using the optimal orientations, the design of a simple control law for each robot is possible.

B. Robot Dynamics

The solution to the optimization problem (6) yields optimal orientations δ_i^*, which can be used as control vectors in the first order dynamic system

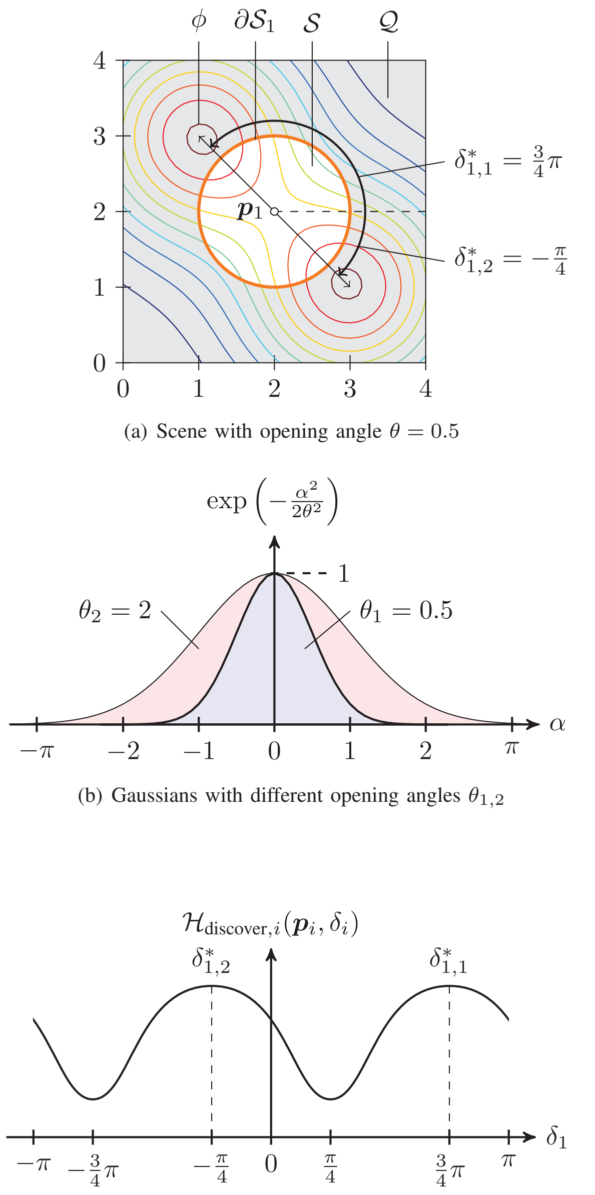
$$\dot{\boldsymbol{p}}_i = \boldsymbol{u}_i = v \begin{pmatrix} \cos\delta_i^* \\ \sin\delta_i^* \end{pmatrix}, \tag{8}$$

with $v \in \mathbb{R}^+$ being a constant velocity of all robots. As depicted in Fig. 1, equation (8) represents a closed loop as the optimization of the orientations depends on the robot positions and the Voronoi partition, which change continuously in time.

C. Impact of the Angular Component

Next, we focus on the proper choice of the opening angle θ in the angular component. For a better understanding we consider the scene depicted in Fig. 3.

In (a) the scene $\mathcal{Q}$ contains a single robot in $\boldsymbol{p}_1 = (2, 2)^\top$ with an omni-directional vision radius of one meter and a mixture of two Gaussians, with means at $(1, 3)^\top$ and $(3, 1)^\top$, as distribution function ϕ, illustrated by the contour lines. The robot already explored the environment $\mathcal{S}$ so that there is a circular border $\partial\mathcal{S}_1$ separating the explored parts from the unknown area.



(a) Scene with opening angle $\theta = 0.5$

(b) Gaussians with different opening angles $\theta_{1,2}$

(c) Objective function $\mathcal{H}_{\text{discover},i}$ with opening angle $\theta = 0.5$

Fig. 3. Influence of the opening angle θ on the angular component

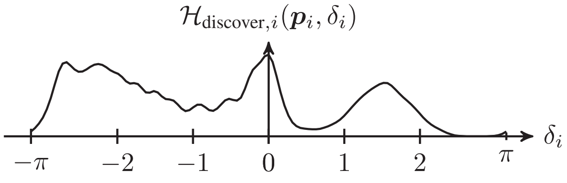
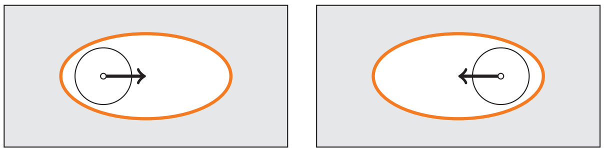


Fig. 4. Example of $\mathcal{H}_{\text{discover},i}$ with small θ in a grid map



(a) Optimal orientation: $\delta_i^* = 0$

(b) Optimal orientation: $\delta_i^* = \pi$

Fig. 5. Optimal orientation without distance component

The graph in (b) illustrates the Gaussian of the angular component with an opening angle $\theta_1 = 0.5$ and $\theta_2 = 2$ in the valid domain $[-\pi, \pi]$.

In (c) the graph shows the objective function $\mathcal{H}_{\text{discover},i}$ as a function of δ_1 with an opening angle of $\theta = 0.5$ for the robot's current position in (a). Optimizing (5) leads to two maxima $\delta^*_{1,1} = \frac{3}{4}\pi$ and $\delta^*_{1,2} = -\frac{\pi}{4}$, which reflects the desired behavior as each robot is supposed to first move to space with larger ϕ during the exploration process. Both orientations are equally well suited due to the symmetry in the scene. The distance component does not have any effect in this case as the Euclidean norm to all frontiers $\boldsymbol{q} \in \partial\mathcal{S}_1$ is the same.

Larger values of θ increase the width of the Gaussian of the angular component. This results in values near 1 for larger angles α, see Fig. 3 (b). Hence, optimizing (5) with too large opening angles θ leads to a compromise in finding the optimal orientation δ^*_i. E.g., for $\theta = 2$ the optimal orientations are $\delta^*_{1,1} = \frac{\pi}{4}\pi$ and $\delta^*_{1,2} = -\frac{3}{4}\pi$. In the worst case this could be an orientation pointing to explored space.

Choosing small opening angles θ leads to lots of local maxima in the objective function $\mathcal{H}_{\text{discover},i}$, especially when using a grid map representation of the environment, see Fig. 4. Additionally, very small values θ are numerically unstable and cannot be sufficiently represented by the floating point arithmetic.

D. Impact of the Distance Component

During an exploration process it may happen that frontiers are very far away from a robot. Without the distance component a robot might travel a long distance before exploring new environment, although frontiers are within immediate reach in a different direction. Such a situation is covered by Fig. 5. In (a) the optimization step produces an orientation $\delta^*_i = 0$, although an orientation of $\delta^*_i = \pi$ would lead into unknown space much faster.

Worse, as the optimization of (5) neglecting the distance component leads to orientations δ^*_i with as many frontiers having a small angle α as possible, the robot in Fig. 5 does not explore any new area. In (a) the robot moves to the right. Finally arriving in (b) the optimal orientation points to the left again. Thus, the robot oscillates in already explored area and never explores anything new, which is obviously a behavior we want to avoid.

IV. Simulation

A. Implementation Details

As listed in Algorithm 1, we have implemented a discrete time version of the motion control law (8), which basically works as follows: Each robot computes its Voronoi cell and optimizes its orientation δ_i according to (6) as

$$\delta_i^{(k)} = \arg\max_{\delta_i} \mathcal{H}_{\text{discover},i}(\boldsymbol{p}_i^{(k)}, \delta_i^{(k-1)}), \tag{9}$$

with the time step index k. Next, each robot updates its position according to the control law

$$\boldsymbol{p}_i^{(k+1)} = \boldsymbol{p}_i^{(k)} + v \begin{pmatrix} \cos\delta_i^{(k)} \\ \sin\delta_i^{(k)} \end{pmatrix}. \tag{10}$$

Algorithm 1 Discrete-Time Exploration Process

```
Initialization of $\boldsymbol{p}_i^{(0)}$, $\delta_i^{(0)}$ and $\mathcal{S}^{(0)}$
while $\mathcal{S}^{(k)} \neq \mathcal{Q}$ do
    Communicate robot positions $\boldsymbol{p}_i^{(k)}$ and map data $\mathcal{S}^{(k)}$ to neighbors
    for $i = 1, \dots, N$ in parallel do
          Partitioning:  $\mathcal{V}_i^{(k)}$ and $\partial\mathcal{S}_i^{(k)}$
        Optimization:  $\delta_i^{*(k-1)} \longmapsto \delta_i^{*(k)}$
      Motion control:  $\boldsymbol{p}_i^{(k)} \xmapsto{\boldsymbol{u}_i} \boldsymbol{p}_i^{(k+1)}$
          Exploration:  $\mathcal{S}_i^{(k)} \longmapsto \mathcal{S}_i^{(k+1)}$
    end for
end while
```

Finally, all robots explore new territory at $\boldsymbol{p}_i^{(k+1)}$ and extend the map in the vision radius.

To represent the environment we choose an occupancy grid map [5], [11], where each cell is of type unknown, frontier, free or obstacle. Cells marked as obstacle only appear at the border $\partial\mathcal{Q}$ of the environment, because $\mathcal{Q}$ is convex.

In the following simulation runs, all robots have a vision radius of $r = 2$m. We choose $\theta = 0.5$ as opening angle and set the standard deviation of the distance component to the vision radius, i.e. $\sigma = 2$. Each cell of the grid map represents $A = 0.4 \times 0.4\text{m}^2$ of the environment.

B. Single-Robot Exploration

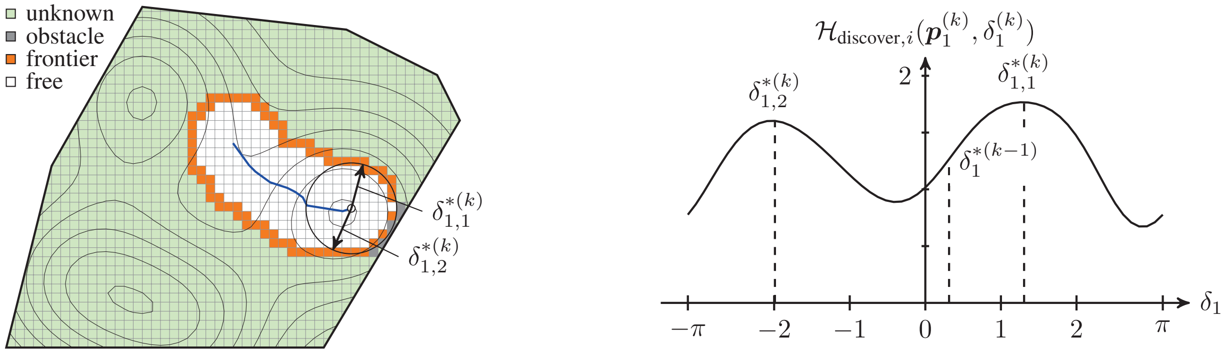
First, we consider an exploration scene with a single robot to investigate once again the behavior of the objective function $\mathcal{H}_{\text{discover}}$. We choose a convex region $\mathcal{Q}$, similar to the examples in [1], limited by the corners $(0,0)^\top$, $(2,8)^\top$, $(6,15)^\top$, $(15,14)^\top$, $(19,12)^\top$, $(20,10)^\top$, $(14,0)^\top$ as well as a mixture of unnormalized Gaussians with means at $\boldsymbol{c}_1 = (6,11)^\top$, $\boldsymbol{c}_2 = (3,3)^\top$, $\boldsymbol{c}_3 = (8,1)^\top$ and $\boldsymbol{c}_4 = (15,6)^\top$ as density function ϕ, i.e.

$$\phi(\boldsymbol{q}) = 3\sum_{j=1}^{4} \exp\left(-\frac{\|\boldsymbol{q} - \boldsymbol{c}_j\|_2^2}{2 \cdot 3^2}\right). \tag{11}$$

The velocity in (10) is set to $v = 0.4\frac{\text{m}}{\text{s}}$ in each iteration and, hence, matches the resolution of the grid.

Fig. 6(a) illustrates the exploration process after several iterations. The initial position of the robot was set to $\boldsymbol{p}_1^{(0)} = (10,9)^\top$ with an orientation of $\delta_1^{(0)} = 0$. Starting the simulation, the robot first moves to the heavier weighted territory until the border is reached. Fig. 6(b) shows the objective function $\mathcal{H}_{\text{discover},i}$ for the scene in Fig. 6(a). As we can see, there are two maxima in $\delta_{1,1}^{*(k)} \approx 1.31$ and $\delta_{1,2}^{*(k)} \approx -1.99$ corresponding to the upper right and lower left directions.

The optimization step is implemented by using a direct method. Finding the local maximum of $\mathcal{H}_{\text{discover},i}$ in iteration k is based on the previous orientation $\delta_i^{*(k-1)}$. When tracking the local maximum, robots follow their current path more steadily. Always choosing the global maximum works as well, but robots change their orientation more frequently to move to the most promising frontiers.



(a) Exploration after several iterations.

(b) Objective function in the current situation

Fig. 6. Single robot exploration process.

C. Multi-Robot Exploration

Next, we simulate the same scene with multiple robots, namely five, initially positioned in $\boldsymbol{p}_1^{(0)} = (8,7)^\top$, $\boldsymbol{p}_2^{(0)} = (9,6)^\top$, $\boldsymbol{p}_3^{(0)} = (10,9)^\top$, $\boldsymbol{p}_4^{(0)} = (10,10)^\top$, $\boldsymbol{p}_5^{(0)} = (11,9)^\top$ with orientations $\delta_i^{(0)} = 0$.

The entire exploration process is illustrated in Fig. 7 in several steps. In Fig. 7(a) we can see the initial configuration along with the Voronoi cells and the contour lines of the density function. As we can see in Fig. 7(b) the robots spread evenly into the unexplored area and explore different parts of the environment, where the density function returns a large value. During the exploration it happens that some cells remain unexplored and form small isles, explaining the isolated frontier cell, see Fig. 7(b) and Fig. 7(c). This is caused by the overlapping of the sensing radii over time. In the worst case, those parts have to be visited towards the end again.

In Fig. 7(c) the robot at the bottom left has almost entirely explored its Voronoi cell. Hence, $\mathcal{H}_{\text{discover},i}$ is constantly zero as no frontier cells are in its Voronoi cell in the subsequent iterations. Instead of continuing the exploration, the robot defaults to the centroidal search as explained in depth in [1], i.e., it covers the explored area as good as possible. After 70 iterations the exploration process is complete. This is illustrated in Fig. 7(d), where the Voronoi cells and the grid are omitted for increased visibility of the trajectories of the robots.

Fig. 7(e) plots the percentage of the environment explored over the time steps k and the time-optimal solution for comparison. In the time-optimal case, all robots can explore

$$2\sum_{i=1}^{N} \frac{v_i r_i}{A} = 2N\frac{vr}{A} = 50$$

cells per iteration. In the simulation, the environment explored increases in a linear fashion until iteration 20. Here, all robots explore approximately a mean of 36 cells per iteration, because for the chosen initial configuration the robot in $\boldsymbol{p}_1$ only explores very few cells (Fig. 7(a)). However, neglecting this robot, the other robots explore about 45 cells per iteration, which is a near time-optimal behavior. These circumstances illustrate, that the effectiveness depends on the initial configuration $\mathcal{P}$ of the robots. The amount of the explored environment after iteration 20 still increases linearly although with smaller slope. This indicates that robots are not in immediate reach of frontier cells anymore and first have to travel larger distances to explore new area. After 50 iterations 98% of the environment is explored. The remaining frontier cells are either isles or corners such as the top left corner in Fig. 7(c). These circumstances lead to large distances without exploring any new area. Hence, the robots require another 20 iterations to finish the exploration process. However, the aforementioned problems often appear in exploration strategies (e.g. [8]), as “the actual information that can be gathered by moving to a particular location is impossible to predict” [2].

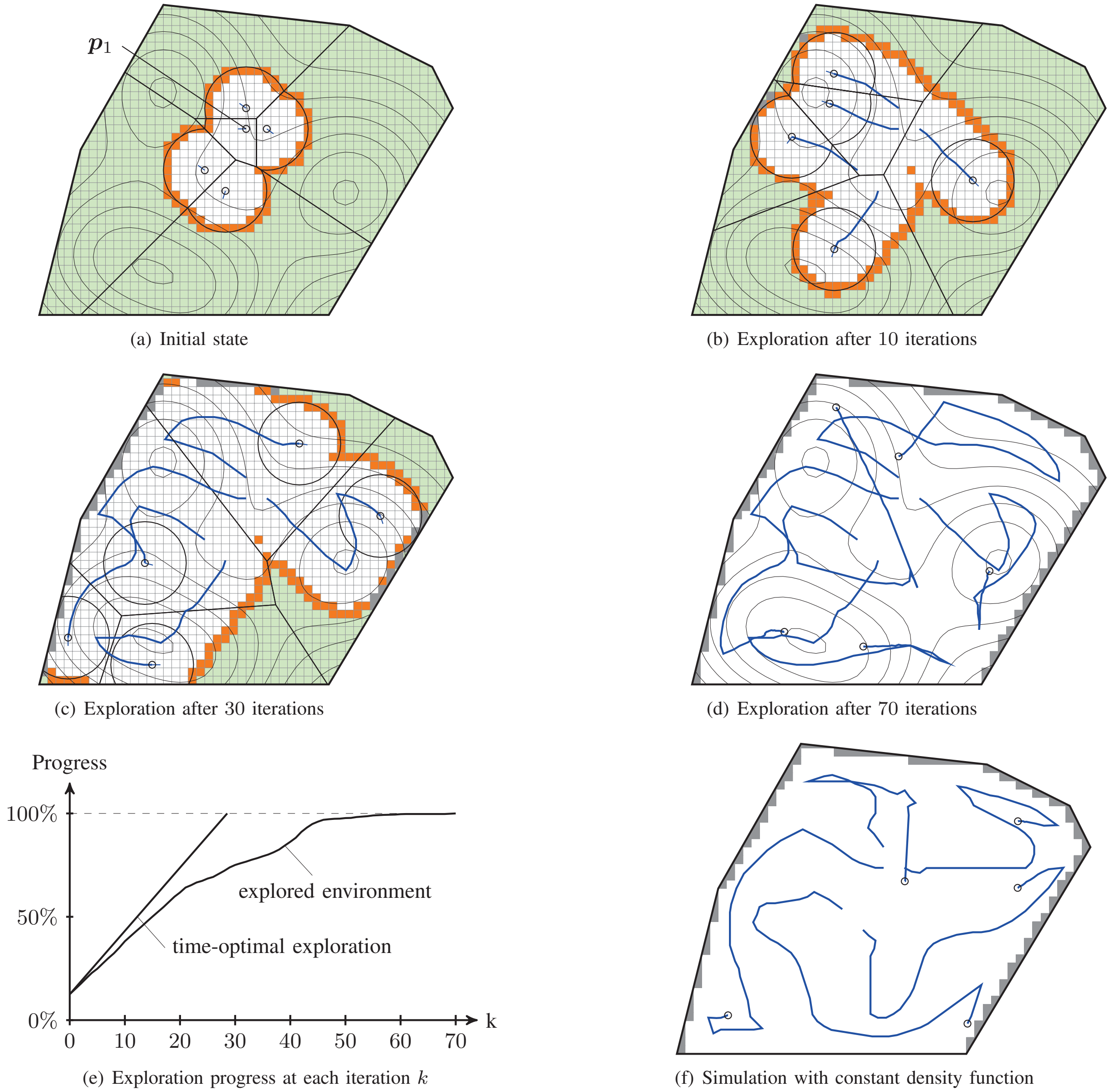
Fig. 7(f) shows the entire exploration process performed with constant density function $\phi(\boldsymbol{q}) = 1$, proving that ϕ is not the main factor driving the exploration process.

Intensive computer simulations show that varying the opening angle θ in the interval $[0.5; 1.0]$ results in a mean of 58.6 iterations for the entire exploration process in Fig. 7 with a standard deviation of 6.68 iterations. Varying the standard deviation σ of the distance component in the interval $[1.5; 3]$ results in a mean of 57 iterations with a standard deviation of 6.75. The simulations demonstrate the robustness of the exploration process with respect to the choice of the parameters θ and σ.

V. Conclusion

We presented a novel approach to multi-robot exploration. Based on a Voronoi partition, each robot optimizes a locally computable objective function to automatically obtain control vectors, which assure simultaneous exploration of different regions of unknown area. Therefore, we introduced the orientation as an additional degree of freedom in the optimization. The objective function is built such that our approach is optimal with respect to the orientations of the robots.

In future work, we will extend our approach to support more realistic scenes. Hence, next to considering more complex robot dynamics, we will focus on non-convex regions and take problems like localization into account.



(a) Initial state

(b) Exploration after 10 iterations

(c) Exploration after 30 iterations

(d) Exploration after 70 iterations

(e) Exploration progress at each iteration k

(f) Simulation with constant density function

Fig. 7. Simulation result of the step-by-step multi-robot mapping

Further, an open issue is to utilize alternative partitions of the environment to improve the performance of the exploration process.

References

[1] F. Bullo, J. Cortés, and S. Martínez, *Distributed Control of Robotic Networks*. Princeton University Press, 2009.

[2] W. Burgard, M. Moors, C. Stachniss, and F. Schneider, "Coordinated multi-robot exploration," *IEEE Transactions on Robotics*, vol. 21, no. 3, pp. 376–386, June 2005.

[3] J. Cortés, S. Martínez, T. Karatas, and F. Bullo, "Coverage control for mobile sensing networks," *IEEE Transactions on Robotics and Automation*, vol. 20, no. 2, pp. 243–255, April 2004.

[4] M. de Berg, O. Cheong, M. van Kreveld, and M. Overmars, *Computational Geometry: Algorithms and Applications*. Springer, 2008.

[5] A. Elfes, "Sonar-based real-world mapping and navigation," *IEEE Journal of Robotics and Automation*, vol. 3, no. 3, pp. 249–265, 1987.

[6] D. Fox, J. Ko, K. Konolige, B. Limketkai, D. Schulz, and B. Stewart, "Distributed multirobot exploration and mapping," *Proceedings of the IEEE*, vol. 94, no. 7, pp. 1325–1339, 2006.

[7] A. Gusrialdi, T. Hatanaka, and M. Fujita, "Coverage control for mobile networks with limited-range anisotropic sensors," in *Proceedings of the 47th IEEE Conference on Decision and Control*, 2008, pp. 4263–4268.

[8] I. Hussein and D. Stipanovic, "Effective coverage control for mobile sensor networks with guaranteed collision avoidance," *IEEE Transactions on Control Systems Technology*, vol. 15, no. 4, pp. 642–657, July 2007.

[9] S. M. LaValle, *Planning Algorithms*. Cambridge University Press, 2006.

[10] A. Makarenko, S. Williams, F. Bourgault, and H. Durrant-Whyte, "An experiment in integrated exploration," in *IEEE/RSJ International Conference on Intelligent Robots and System*, 2002, pp. 534–539.

[11] H. P. Moravec, "Sensor fusion in certainty grids for mobile robots," *AI Magazine*, vol. 9, no. 2, pp. 61–74, 1988.

[12] M. Rooker and A. Birk, "Multi-robot exploration under the constraints of wireless networking," *Control Engineering Practice*, vol. 15, no. 3, pp. 435–445, March 2007.

[13] B. Yamauchi, "A frontier-based approach for autonomous exploration," in *Proceedings of the 1997 IEEE International Symposium on Computational Intelligence in Robotics and Automation*, 1997, pp. 146–151.