

# Distributed Multi-Robot Exploration

Vom Fachbereich  
Elektrotechnik und Informationstechnik  
der Technischen Universität Darmstadt  
zur Erlangung des akademischen Grades eines  
Doktor-Ingenieurs (Dr.-Ing.)  
genehmigte Dissertation

von

**Dipl.-Inform. Dominik Haumann**

geboren am 27. November 1982 in  
Heidelberg/Baden-Württemberg

Referent: Prof. Dr.-Ing. J. Adamy  
Korreferent: Prof. S. Roth, PhD  
Tag der Einreichung: 29. April 2014  
Tag der mündlichen Prüfung: 02. Juli 2014

D17  
Darmstadt 2015



# Vorwort

Die vorliegende Arbeit ist im Rahmen meiner wissenschaftlichen Tätigkeit am Institut für Automatisierungstechnik am Fachgebiet Regelungsmethoden und Robotik an der TU Darmstadt entstanden. Für die Unterstützung und den Freiraum, meine Ideen verwirklichen zu können, möchte ich mich bei meinem Doktorvater Herrn Professor Adamy ganz herzlich bedanken. Ebenso danke ich Herrn Professor Roth für die freundliche Übernahme des Korreferats.

Meine Promotion wurde durch ein Stipendium der Deutschen Forschungsgemeinschaft (DFG) im Graduiertenkolleg GRK 1362 gefördert. Die damit verbundenen Rahmenbedingungen erlaubten mir nicht nur den wissenschaftlichen Austausch mit den involvierten Stipendiaten des GRKs, sondern boten auch die Möglichkeit, an diversen Summer Schools teilzunehmen und regelmäßig Gastredner einzuladen. Hierfür bin ich sehr dankbar – da diese Rahmenbedingungen während einer Promotion insbesondere in finanzieller Hinsicht nicht selbstverständlich sind.

Mein ganz besonderer Dank gilt Herrn Willert für die unzähligen Diskussionen, Anregungen und die kreative Arbeitsatmosphäre oftmals bei leckerem Kaffee und Kuchen – eine bessere Zusammenarbeit und fachliche Betreuung kann man sich nicht wünschen!

Ein herzliches Dankeschön möchte ich meinen Kollegen und Freunden am Fachgebiet und im Graduiertenkolleg für die vielen fachlichen und nicht-fachlichen Diskussionen und das tolle Arbeitsklima aussprechen. Thomas Guthier, Jochen Grieser und Lukas Klodt haben meine Arbeit Korrektur gelesen. Weiterhin bedanken möchte ich bei Arne Wahrburg und meinen Bürokollegen Kim Listmann und Stefan Gering für den stets feinfühligsten Humor – bleibt Linux und KDE treu! Des weiteren möchte ich mich bei Frau Heid und Frau Gelman für die sehr gute Zusammenarbeit in all den Jahren bedanken – ohne euch läuft am Fachgebiet nichts.

Mein ganz besonderer Dank gilt meinen Eltern für die großartige und vielfältige Unterstützung zu jeder Zeit in jeglicher Hinsicht!

*Für meine Eltern*

*und in Erinnerung an meine Großeltern.*

# Contents

<b>List of Symbols</b>	<b>VIII</b>
<b>Abstract</b>	<b>XI</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Autonomous Robot Exploration . . . . .	2
1.2 Problem Statement and Contributions . . . . .	3
1.3 Outline of the Dissertation . . . . .	6
<b>2 Mathematical Foundations</b>	<b>8</b>
2.1 Computational Geometry . . . . .	8
2.1.1 Convex Sets . . . . .	8
2.1.2 Nonconvex Environments . . . . .	10
2.1.3 Reachability and Visibility . . . . .	11
2.1.4 The Voronoi Partition . . . . .	12
2.1.5 Generalized Voronoi Partition . . . . .	14
2.2 Stability Theory . . . . .	15
2.2.1 Dynamical Systems . . . . .	15
2.2.2 Lyapunov Functions . . . . .	16
2.2.3 Invariance Principle . . . . .	18
<b>3 Literature Review</b>	<b>20</b>
3.1 The Coverage Problem . . . . .	20
3.1.1 Introduction . . . . .	20
3.1.2 Distributed Optimization Problem . . . . .	22
3.1.3 A Note on Choosing Weights . . . . .	24
3.1.4 Gradient of the Objective Function . . . . .	25
3.1.5 Continuous-Time Motion Control Laws . . . . .	27
3.1.6 Performance Functions . . . . .	28
3.1.7 Stability Analysis . . . . .	32
3.1.8 From Static to Dynamic Coverage . . . . .	33
3.1.9 Coverage of Nonconvex Environments . . . . .	34

3.2	Multi-Robot Exploration . . . . .	38
3.2.1	Simultaneous Localization and Mapping . . . . .	39
3.2.2	Objective Functions for a Single Robot . . . . .	40
3.2.3	On Coordinating Multiple Robots . . . . .	41
3.2.4	Coordination by Partitioning the Environment . . . . .	42
3.3	Discussion . . . . .	43
3.3.1	Gradient-Based Motion Control . . . . .	43
3.3.2	A Note on Performance Measures . . . . .	44
<b>4</b>	<b>From Coverage to DisCoverage</b>	<b>45</b>
4.1	Preliminaries and Notation . . . . .	45
4.2	Centroidal Search-Based DisCoverage . . . . .	46
4.2.1	Modifying the Density Function . . . . .	47
4.2.2	Building the Partial Derivative . . . . .	50
4.2.3	DisCoverage in Convex Environments . . . . .	51
4.2.4	DisCoverage in Nonconvex Environments . . . . .	60
4.2.5	Fallback Strategy when Using Multiple Robots . . . . .	73
4.2.6	Possible Extensions to the Vehicle Dynamics . . . . .	76
4.3	Orientation-Based DisCoverage . . . . .	77
4.3.1	Introducing Robot Orientations . . . . .	78
4.3.2	Continuous-Time Control Law . . . . .	80
4.3.3	Impact of the Angular Component . . . . .	81
4.3.4	Impact of the Distance Component . . . . .	85
4.3.5	Handling Nonconvex Environments . . . . .	87
4.3.6	Behavior of Multiple Robots . . . . .	90
4.3.7	Varying the Vehicle Dynamics . . . . .	93
<b>5</b>	<b>Simulation Results and Lab Experiments</b>	<b>94</b>
5.1	The Time-Optimal Case . . . . .	94
5.2	Discrete-Time Motion Control Laws . . . . .	96
5.2.1	Multi-Robot Exploration Strategies . . . . .	96
5.2.2	Fallback Strategy . . . . .	98
5.3	Environmental and Robotic Setup . . . . .	99
5.4	Statistical Evaluation . . . . .	101
5.4.1	Results in the Convex Environment . . . . .	102
5.4.2	Results in the Nonconvex Allowable Environment . . . . .	106
5.5	Experimental Results with E-puck Robots . . . . .	109
<b>6</b>	<b>Discussion</b>	<b>112</b>
6.1	Simulation Results and Lab Experiments . . . . .	113

---

6.2	Performance Analysis . . . . .	114
6.3	Convergence Properties . . . . .	115
6.4	Optimality of Space Partitions . . . . .	116
6.5	Higher State Space Dimensions . . . . .	116
6.6	Map Representation . . . . .	117
6.7	Discrete-Time Systems . . . . .	117
<b>7</b>	<b>Conclusion</b>	<b>119</b>
7.1	Summary . . . . .	119
7.2	Future Research . . . . .	120
<b>A</b>	<b>Computational Geometry</b>	<b>123</b>
A.1	k-Means Clustering and Coverage . . . . .	123
A.2	Distributed Computation of the Voronoi Partition . . . . .	124
A.3	Software . . . . .	127
<b>B</b>	<b>Hardware Specifications</b>	<b>128</b>
B.1	The E-puck Robot . . . . .	128
	<b>Publications and Patents</b>	<b>129</b>
	<b>Bibliography</b>	<b>132</b>

# List of Symbols

## Abbreviations

e.g.,	exempli gratia – for example
i.e.,	id est – that is
DisCoverage	combined coverage and exploration
SLAM	Simultaneous Localization and Mapping
MIT	Massachusetts Institute of Technology

## Functions

$\phi(\cdot)$	density function
$d_g(\mathbf{p}, \mathbf{q})$	geodesic distance from a point $\mathbf{p}$ to a point $\mathbf{q}$
$d_g(\mathbf{p}, \mathcal{Q})$	geodesic distance from a point $\mathbf{p}$ to a set $\mathcal{Q}$
$\text{diam } \mathcal{Q}$	diameter of the set $\mathcal{Q}$
$\text{dist}(\mathbf{p}, \mathcal{Q})$	Euclidean distance from point $\mathbf{q}$ to the set $\mathcal{Q}$
$\text{dscn}(f)$	discontinuities of a function $f$
$\mathbf{e}(\mathbf{p}, \mathbf{q})$	unit vector along the first path segment on the geodesic path from $\mathbf{p}$ to $\mathbf{q}$
$f(x)$	performance function for distance $x$
$f_-(x), f_+(x)$	one-sided limits at $x$
$\mathbf{f}, \mathbf{f}_i$	vector-valued function
$\mathcal{H}$	objective function
$\mathcal{H}_{\text{area}}$	objective function of the area problem
$\mathcal{H}_{\text{cover}}$	objective function of the coverage problem
$\mathcal{H}_{\text{lim}}$	objective function of the limited centroidal search
$\mathcal{H}_{\text{discover}}$	centroidal search-based DisCoverage objective function
$\mathcal{H}_{\text{discover}}^*$	centroidal search-based DisCoverage objective function in nonconvex environments
$\mathcal{H}_{\text{orient}}$	orientation-based DisCoverage objective function
$\mathcal{H}_{\text{orient}}^*$	orientation-based DisCoverage objective function in nonconvex environments



$\mathcal{H}_{\text{MinDist}}$	objective function of the MinDist exploration approach
$\mathbf{T}_{\mathbf{p}_i}(\mathbf{q})$	transformation to star-shaped domain
$V(\mathbf{x})$	Lyapunov function candidate

## Notation

$\ \cdot\ , \ \cdot\ _2$	Euclidean norm
$\square$	quod erat demonstrandum – end of a proof
$\prec (\succ)$	negative (positive) definiteness of a matrix
$\mathbf{0}$	vector or matrix of zeros, depending on the context
$\mathbb{E}(\cdot)$	expected value
$\star$	indicates star-shaped sets (e.g., visibility sets)

## Sets

$\Delta$	set of all robot orientations
$\mathbb{N}$	natural numbers
$\mathbb{R}^d$	$d$ -dimensional real vector space
$\partial\mathcal{M}$	boundary of the set $\mathcal{M}$
$\overline{B}(\mathbf{p}, r)$	closed ball at point $\mathbf{p}$ with radius $r$
$\mathcal{Q}$	allowable $d$ -dimensional environment
$\mathcal{Q}_\delta$	$\delta$ -contraction of $\mathcal{Q}$
$\mathcal{P}$	robot configuration
$\mathcal{V}$	Voronoi partition
$\mathcal{V}_i$	(geodesic) Voronoi cell of robot $i$
$\mathcal{V}_{i,r}$	$r$ -limited Voronoi cell of robot $i$
$\mathcal{V}_{i,\bar{r}}^*$	$\bar{r}$ -limited visibility set in the Voronoi cell $\mathcal{V}_i$
$\mathcal{V}_{i,\bar{r}}^{\star\varepsilon}$	$\bar{r}$ -limited visibility set in the $\varepsilon$ -contraction of $\mathcal{V}_i$
$\mathcal{S}$	explored parts of the environment $\mathcal{Q}$
$\mathcal{S}_i$	explored parts in Voronoi cell $\mathcal{V}_i$
$\tilde{s}$	frontier to unexplored space
$\tilde{s}_i$	frontier to unexplored space in Voronoi cell $\mathcal{V}_i$
$\mathcal{W}$	arbitrary partition of $\mathcal{Q}$
$\mathcal{W}_i$	one cell of an arbitrary partition $\mathcal{W}$

## Scalars

$N$	total number of robots or agents
$A_\phi$	weighted surface area
$k_i$ and $k_p$	positive gain
$r$	sensing range of a robot
$\bar{r}$	integration range of a robot
$\delta_i$	robot orientation
$\theta$	opening angle of the angular component

## Vectors and Matrices

$H$	Hessian
$\mathbf{m}_\phi$	weighted center of mass (centroid)
$\mathbf{n}_{\text{out}}(\mathbf{q})$	outward normal vector in $\mathbf{q}$
$\mathbf{p}_i$	robot position in the environment $\mathcal{Q}$
$\mathbf{q}$	location in the environment $\mathcal{Q}$
$\mathbf{u}_i$	control input vector of robot $i$
$\mathbf{x}$	state vector of a dynamical system
$\mathbf{x}(t)$	evolution of the trajectory of state $\mathbf{x}$
$\mathbf{x}_e$	equilibrium point

# Abstract

This dissertation introduces the *DisCoverage paradigm*. DisCoverage describes a novel scheme for distributed multi-robot exploration. The objective of the multi-robot exploration problem is to explore and map an a priori unknown environment as quick as possible with a group of autonomous robots. DisCoverage solves this problem through effective coordination of the group, such that the robots simultaneously explore different parts of the environment.

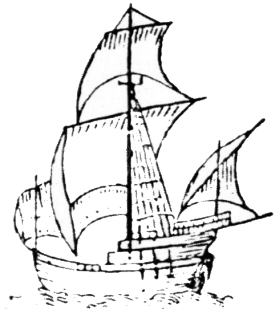
In contrast to existing approaches, DisCoverage provides a distributed solution to the multi-robot exploration problem: Robots communicate and exchange data only with robots in the respective neighborhood, such that no central coordinating unit is required. As a result, the local data exchange among the robots allows the group to globally act as one team, facilitating robust and efficient exploration of the entire environment.

# Kurzfassung

In der vorliegenden Dissertation wird das *DisCoverage-Paradigma* vorgestellt. Dieses Paradigma beschreibt ein neuartiges allgemeines Schema zur verteilten Multi-Roboter-Exploration. Ziel der Multi-Roboter-Exploration ist es, mittels einer Gruppe autonom agierender Roboter eine vorab unbekannte Umgebung zu erforschen und eine Karte in möglichst kurzer Zeit zu erstellen. DisCoverage erfüllt diese Forderung durch eine effektive Koordinierung der Roboter, sodass die Roboter unterschiedliche Bereiche der unbekannteten Umgebung erkunden.

Gegenüber existierenden Verfahren liegt der wesentliche Vorteil von DisCoverage darin, dass der Explorations-Prozess verteilt abläuft: Die Roboter kommunizieren lediglich mit ihren nächsten Nachbarn, sodass keine zentrale Instanz zur Koordinierung benötigt wird. Somit wird durch ausschließlich lokaler Interaktion mit der Umgebung und lokaler Kommunikation zwischen Nachbarn ein globales Verhalten der Gruppe erzielt, welches zu robuster vollständiger Exploration der unbekannteten Umgebung führt.

# 1 Introduction



To-day I got the ship afloat [...] in search of gold and spices, and to discover land.

*Journal of the First Voyage of Columbus* [46]

Christopher Columbus (1451–1506) is well-known for his expeditions to find and explore a new continent. Although Columbus was never aware that he discovered America, without a doubt his expeditions significantly contributed to the exploration of the Earth. Over the centuries, many explorers helped to develop detailed maps of about any location of our planet.

Today, the terrestrial mapping of the Earth is complete. Nevertheless, our environment constantly changes, either due to unpredicted incidents such as an earthquake, or due to planned changes such as road work, the construction of new buildings, new furnishing in the office, or even due to minor changes such as the movement of a chair in a room. Consequently, a priori generated maps are not guaranteed to be accurate. Therefore, the problem of exploring unknown environments is a frequently discussed topic among researchers in the robotics community [127]. In fact, exploring an unknown environment – also referred to as **exploration problem** – is one of the key problems in autonomous robotics, since without any information about the environment, a robot is not able to interact with it. The applications for autonomous robot exploration are manifold [38, 127]: Recent advances apply autonomous robot navigation in everyday life, such as robotic lawnmowers and vacuum cleaners, or driverless cars. More complex tasks include search and rescue missions after a catastrophe such as earthquakes or nuclear accidents as happened in Fukushima (Japan) in 2011 caused by a tsunami, or planetary exploration such as Mars missions that require a high degree of autonomy due to significant communication latencies. Other applications exist, such as the clearing of mine fields. In this case, the environment itself is not subject to the exploration task. Instead, the location of the land mines *underground* build the unknown.

## 1.1 Autonomous Robot Exploration

From a robotics perspective, a common approach to solving the exploration problem is denoted by the **frontier-based** method. The frontier is defined by the boundary separating explored from unknown space. The key idea behind frontier-based exploration was first introduced in 1997 for a single robot by Yamauchi [136], stating: *”To gain the most new information about the world, move to the boundary between open space and uncharted territory.”* Simply put, as long as a frontier exists, the robot moves to a **target point** 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.

Exploring an unknown environment with a robot may protect human lives especially in environments that are extremely dangerous or even inaccessible to humans. However, using a single robot may not be feasible, for instance if the domain to explore is very large. Therefore, in the last decade, researchers focused on exploring unknown environments with multiple robots, which became known as **multi-robot exploration**. Multi-robot exploration has the advantage that – as Guzzoni et al. [68] state – *“many robots make short work,”* meaning that the exploration task is accomplished more quickly with multiple robots compared to the single-robot case. In addition to faster exploration, multi-robot exploration has the benefits of increased robustness and higher accuracy [38, 42, 55]. Increased robustness describes the tolerance with respect to robotic failures: If a robot malfunctions, other robots jump in to take over the respective task and the exploration continues. Higher accuracy is obtained through redundancies in, e.g., the sensors. As a result, the information obtained about the environment is more reliable.

However, these advantages only hold if **effective coordination** of the group can be guaranteed. Burgard et al. [38] describe the problem of effective coordination as follows:



The key problem to be solved in the context of multiple robots is to choose appropriate target points for the individual robots so that they simultaneously explore different regions of the environment.

Typically, coordination techniques are classified in terms of central, hierarchical, and distributed coordination. In central coordination, one coordinator among the robots determines the target points for all robots.

Similarly, in hierarchical coordination robots within communication range form clusters that again coordinate in a centralized way. Central and hierarchical coordination allow to assign optimal target points to each robot due to the increased knowledge available to the coordinators, resulting in well-performing approaches to multi-robot exploration. However, since central and hierarchical coordination are vulnerable to failures of the coordinators, the property of increased robustness is lost. Contrary, in **distributed coordination** no single coordinator exists. Instead, each robot acts solely based on local information obtained from its sensors and through data exchange with robots within communication range. The local data exchange among the robots allows the group to globally act as one team with one common goal. Martínez et al. [93] define distributed coordination as *"obtaining global behavior from local interaction,"* retaining the advantages of multi-robot exploration.

## 1.2 Problem Statement and Contributions

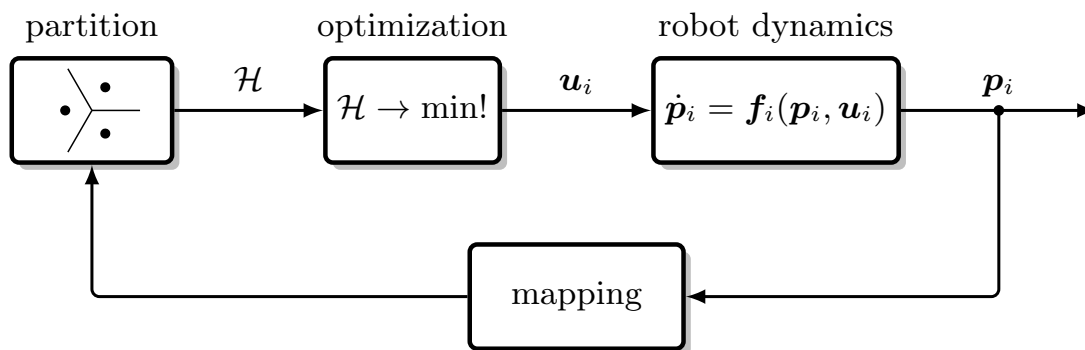
Existing approaches to multi-robot exploration often apply central or hierarchical coordination techniques, losing the advantages of distributed coordination. Hence, the goal of this dissertation is to find approaches to multi-robot exploration that apply distributed coordination techniques. Defining the set  $\mathcal{S}(t) \subseteq \mathcal{Q}$  as the explored parts of a  $d$ -dimensional unknown environment  $\mathcal{Q} \subset \mathbb{R}^d$  at time instant  $t$  with  $\mathcal{S}(t=0) = \emptyset$ , the problem to be solved is formalized as follows:

**Problem statement.** Given a set of  $N$  mobile robots with state  $\mathbf{p}_i$ ,  $i = 1, \dots, N$ , in an unknown environment  $\mathcal{Q}$ . Find distributed coordination techniques based on well-defined communication constraints such that  $\mathcal{S}(t) \rightarrow \mathcal{Q}$  as  $t \rightarrow \infty$ .

Therein, the two design criteria are:

- (i) **Distributed coordination:** Each robot acts solely based on locally available information, i.e., no dedicated coordinating robots exist.
- (ii) **Communication constraints:** The communication topology is such that each robot exchanges information only with a subset of robots within their communication range.

The solution to this problem statement presented in this dissertation essentially builds on a partition of the environment that assigns a region



**Figure 1.1:** Proposed continuous-time feedback loop (cf. Haumann et al. [8])

of dominance to each robot. Based on this partition, each robot locally computes control laws to move to the frontier, solely based on information available in its region of dominance. These control laws are obtained by solving an optimization problem that facilitates multi-robot exploration.

The proposed coordination techniques describe a closed feedback loop that consists of four blocks, namely the **partition**, the **optimization**, the **robot dynamics**, and the **mapping**, as depicted in Figure 1.1. In order to ensure distributed coordination, the Voronoi partition is used as partition of the environment, since it can be computed distributively. That is, each robot is able to compute its region of dominance – from now on called Voronoi cell – solely based on local data exchange with neighboring robots. Based on the Voronoi cells, each robot then optimizes an objective function  $\mathcal{H}$  in the optimization step. Minimizing  $\mathcal{H}$  results in a control input vector  $\mathbf{u}_i$  for the  $i$ -th robot. The robot state vector  $\mathbf{p}_i$  and the control input vector  $\mathbf{u}_i$  are then used by the robot dynamics  $\dot{\mathbf{p}}_i = \mathbf{f}_i$ , moving the robots towards unexplored space. Finally, arriving at the frontier, each robot maps the environment within sensing range, which in turn pushes back the frontier to unexplored space. The robot movements and the newly available information about the environment change the partition continuously over time, closing the feedback loop.

This feedback loop fully satisfies the design criterion *distributed coordination*, since i) each robot is able to compute its Voronoi cell on its own, and ii) based on the Voronoi cell, all following steps solely rely on information available in the respective Voronoi cells. Further, the design criterion *communication constraints* is also satisfied by the Voronoi partition, since the Voronoi partition is dual to the Delaunay triangulation [71], and therewith each robot only requires a well-defined set of robot neighbors in order to compute its Voronoi cell.



The presented multi-robot exploration approaches heavily rely on the solution to the **coverage problem** proposed by Bullo et al. [36]. In order to avoid confusion, it is important to stress that the coverage problem itself is *unrelated* to the exploration problem. The objective of the coverage problem is to place a set of robots in an *a priori known* domain such that the domain is optimally covered. Here, optimal coverage is defined by minimizing the expected distance from the robot positions to all locations in the domain.

Essentially, the key contribution of this dissertation is to solve the exploration problem by transferring the solution to the coverage problem proposed by Bullo et al. [36] to the exploration problem. For this reason, the proposed approaches are said to apply the **DisCoverage paradigm**, since DisCoverage combines coverage and distributed multi-robot exploration. In addition to the DisCoverage paradigm, the contributions are as follows:

- Since the solution to the coverage problem by Bullo et al. [36] is restricted to convex environments, another major contribution lies in extending the solution to the coverage problem to support nonconvex environments.
- Two new multi-robot exploration strategies – the *centroidal search-based* and the *orientation-based* DisCoverage approach – are introduced first for convex and then for nonconvex polygonal environments, both following the DisCoverage paradigm.
- A proof of convergence is provided for the centroidal search-based DisCoverage approach, meaning that this approach is always guaranteed to explore all parts of an unknown environment.
- Introducing the orientation-based DisCoverage approach, a new transformation to nonconvex environments is proposed that maps any nonconvex polygonal environment to star-shaped regions. This transformation can be understood as a tool to generic path planning in nonconvex environments. Consequently, the proposed transformation is not restricted to the exploration problem.
- Extensive statistically significant simulations and lab experiments are provided to validate the proposed multi-robot exploration strategies along with an in-depth discussion.
- The time-optimal case for exploring an unknown environment with multiple robots is formulated as a theoretical lower bound for the minimum time needed to explore the entire environment.

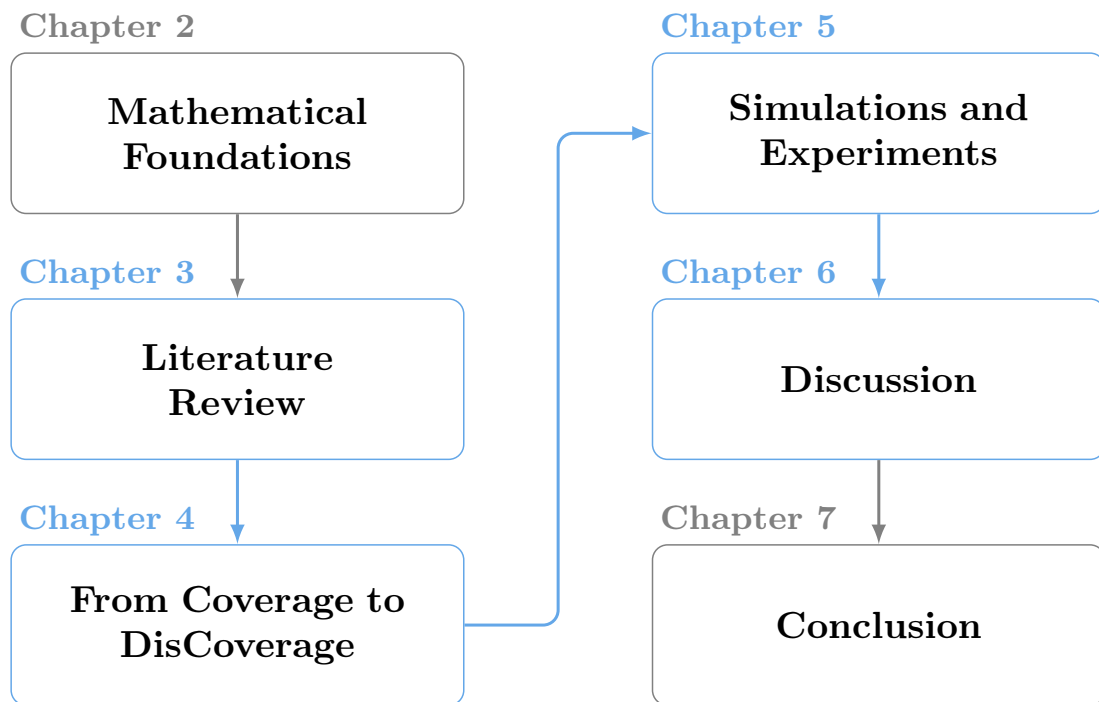
- A measure for the *degree of star-shapedness* with respect to a robot position is introduced, allowing a robot to evaluate the optimality of its position in the environment in the context of the coverage problem.
- A modification to the distance transform based on the geodesic distance is proposed, extending the distance transform traditionally known in image processing.

### 1.3 Outline of the Dissertation

The outline of this dissertation is depicted in Figure 1.2. The next chapter introduces foundations in computational geometry and stability theory as well as conventions and notation used throughout this dissertation. Thereafter, Chapter 3 discusses related literature: The coverage problem is formally introduced, and existing solutions to the coverage problem in convex environments as well as attempts to solve the coverage problem in nonconvex environments are discussed. Similarly, existing multi-robot exploration strategies are presented and classified in terms of coordination techniques and communication constraints. Finally, a short discussion about the relation of coverage and exploration is provided, stressing the fact that a multi-robot exploration approach following the idea of the solution to the coverage problem does not yet exist.

Chapter 4 introduces DisCoverage as a new paradigm to multi-robot exploration. Following this paradigm, the *centroidal search-based* and the *orientation-based* DisCoverage approaches are proposed. Initially restricted to convex environments, the DisCoverage approaches are then extended to support nonconvex environments. The theory to support nonconvex environments is not limited to multi-robot exploration, therefore connections to related research topics are highlighted. The behavior for multiple robots as well as more complex vehicle dynamics are discussed.

In Chapter 5, the proposed approaches are evaluated through extensive simulations. The DisCoverage approaches are compared to the time-optimal case as well as to the *MinDist* approach – a standard path planning approach to exploration. Further, results of lab experiments with e-puck robots, conducted in collaboration with the MIT and ETH Zurich, are presented. Both, the simulations as well as the experiments, validate the DisCoverage paradigm.



**Figure 1.2:** Outline. Chapters with contributions are highlighted in blue.

The results of the simulations and experiments are discussed in Chapter 6, pointing out advantages of the proposed approaches to multi-robot exploration. In addition, several aspects such as performance, convergence properties, and optimality of space partitions are investigated. A conclusion of this dissertation, consisting of a summary as well as possible future research directions, is given in Chapter 7.

## 2 Mathematical Foundations

In this chapter, necessary mathematical foundations of computational geometry and stability theory are provided in Section 2.1 and Section 2.2, respectively.

### 2.1 Computational Geometry

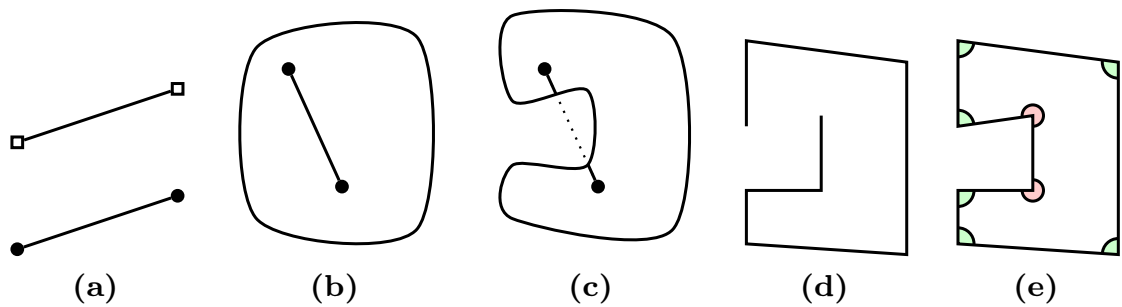
From a mathematical perspective, the explored and unexplored parts as well as the a priori unknown environment itself are sets. Likewise, the surroundings sensed by robots can be modeled in terms of visibility sets. Therefore, this section introduces properties of sets that are used extensively throughout the dissertation. If not noted otherwise, the upcoming notation and definitions closely follow the ones introduced by Bullo et al. [36].

#### 2.1.1 Convex Sets

**Definition 2.1** (Convex sets). Given two points  $\mathbf{p}, \mathbf{q}$  in the  $d$ -dimensional space  $\mathbb{R}^d$ . The sets  $(\mathbf{p}, \mathbf{q}) = \{\mathbf{x} \in \mathbb{R}^d \mid \mathbf{x} = \mathbf{p} + \lambda(\mathbf{q} - \mathbf{p}), \lambda \in (0, 1)\}$  and  $[\mathbf{p}, \mathbf{q}] = \{\mathbf{x} \in \mathbb{R}^d \mid \mathbf{x} = \mathbf{p} + \lambda(\mathbf{q} - \mathbf{p}), \lambda \in [0, 1]\}$  are called the **open** and the **closed line segment**, respectively. A set  $\mathcal{Q} \subseteq \mathbb{R}^d$  is called **convex** if for all  $\mathbf{p}, \mathbf{q} \in \mathcal{Q}$  the closed line segment  $[\mathbf{p}, \mathbf{q}]$  is contained in  $\mathcal{Q}$ , i.e.,  $[\mathbf{p}, \mathbf{q}] \subseteq \mathcal{Q}$ . If a set is not convex, it is called **nonconvex**.

Convex sets are defined for arbitrary dimensions  $d$ . The boundary of a convex (or nonconvex) set  $\mathcal{Q} \subseteq \mathbb{R}^d$  is denoted with  $\partial\mathcal{Q}$ . According to Definition 2.1 a convex set includes its boundary. Open and closed line segments, convex and nonconvex sets are depicted in Figure 2.1(a)–2.1(c). Next to convex sets, closed line segments also form a basis for polygonal lines.

**Definition 2.2** (Polygonal line). Given a finite set of  $k$  points  $\mathbf{q}_1, \dots, \mathbf{q}_k \in \mathbb{R}^d$ . The **polygonal line** is defined by the union of all closed line segments  $\cup_{i=1}^{k-1} [\mathbf{q}_i, \mathbf{q}_{i+1}]$ . The points  $\mathbf{q}_1, \dots, \mathbf{q}_k$  of a polygonal line are called **vertices**.



**Figure 2.1:** (a) open and closed line segment, (b) convex set, (c) nonconvex set, (d) polygonal line, (e) nonconvex polygon with 6 convex vertices (green) and 2 concave vertices (red).

If in addition the polygonal line contains the closed line segment  $[q_k, q_1]$ , then the vertices define a **closed polygonal line**.

Based on the definition of closed polygonal lines a polygon is defined in the 2-dimensional space as follows.

**Definition 2.3** (Polygon). Given a closed polygonal line in  $\mathbb{R}^2$ . Then, a **polygon** is defined by the union of a closed polygonal line and its interior. A polygon is said to be a **convex polygon** if its set is convex according to Definition 2.1.

By definition, a polygon always represents a compact set, i.e., the set is closed and bounded. A polygon can further be characterized by investigating its vertices.

**Definition 2.4** (Convex vertices). Given a polygon defined by the vertices  $q_1, \dots, q_k \in \mathbb{R}^2$ . A vertex  $q_i$  is called **convex** if its interior angle is less than or equal to  $\pi$  rad. If a vertex is not convex, it is called **concave**.

Apparently, a polygon is convex if and only if all its vertices are convex. Definition 2.2 allows self-intersections of the polygonal line. However, for the rest of the dissertation, non-intersecting polygonal lines and polygons are assumed. A polygonal line is depicted in Figure 2.1(d), and the respective polygon defined by its convex and concave vertices is shown in Figure 2.1(e).

*Remark 2.1* (Generalization). The notion of a polygon is by definition limited to the 2-dimensional space. However, the definitions of polygonal lines and polygons can analogously be extended to higher dimensions  $d$ . For instance, a polygon is referred to as a **polyhedron** in case of  $d = 3$ , or as a **polytope** for  $d > 3$ .

Prior to introducing nonconvex environments, it is necessary to define the notion of connectedness.

**Definition 2.5** (Connected sets). A set  $\mathcal{Q} \subseteq \mathbb{R}^d$  is said to be **connected**, if for all  $\mathbf{p}, \mathbf{q} \in \mathcal{Q}$  there exists a path from  $\mathbf{p}$  to  $\mathbf{q}$  such that the entire path is contained in  $\mathcal{Q}$ . If a set is not connected, it is called a **disconnected** set. Further, the **diameter** of a connected set  $\mathcal{Q}$  is defined as  $\text{diam } \mathcal{Q} = \max_{\mathbf{p}, \mathbf{q} \in \mathcal{Q}} d_g(\mathbf{p}, \mathbf{q})$ , where  $d_g(\mathbf{p}, \mathbf{q})$  defines the geodesic distance from  $\mathbf{p}$  to  $\mathbf{q}$  in  $\mathcal{Q}$ .

The geodesic distance  $d_g(\mathbf{p}, \mathbf{q})$  returns the shortest distance from  $\mathbf{p}$  to  $\mathbf{q}$  in the set  $\mathcal{Q}$ . In convex sets, the geodesic distance is equivalent to the Euclidean distance.

### 2.1.2 Nonconvex Environments

Subject to the exploration problem are usually nonconvex environments. As a consequence, a proper definition for these nonconvex environments is required. Throughout this dissertation nonconvex environments with obstacles are assumed as follows.

**Definition 2.6** (Allowable environment). An **allowable environment** is defined by a tuple  $(\mathcal{B}, \mathcal{O})$ , where  $\mathcal{B} \subseteq \mathbb{R}^d$  is a polytope denoting the environment and its boundary, and  $\mathcal{O}$  is a set of  $k$  polytopes  $o_1, \dots, o_k \subset \mathcal{B}$  representing obstacles. Consequently, an allowable environment is denoted by the set  $\mathcal{Q} := \mathcal{B} \setminus \cup_{i=1}^k o_i$ .

Definition 2.6 allows both the polytope  $\mathcal{Q}$  as well as all obstacles to be nonconvex. A property of an allowable environment  $\mathcal{Q}$  is that the boundary  $\partial\mathcal{Q}$  is continuously differentiable in all  $\mathbf{q} \in \partial\mathcal{Q}$  along the boundary *except* in the finite set of convex and concave vertices. Therefore, it is convenient to define a  $\delta$ -contraction, which was first proposed by Udupa et al. [130] and Lozano-Pérez and Wesley [87] for collision-free path planning of robotic manipulators.

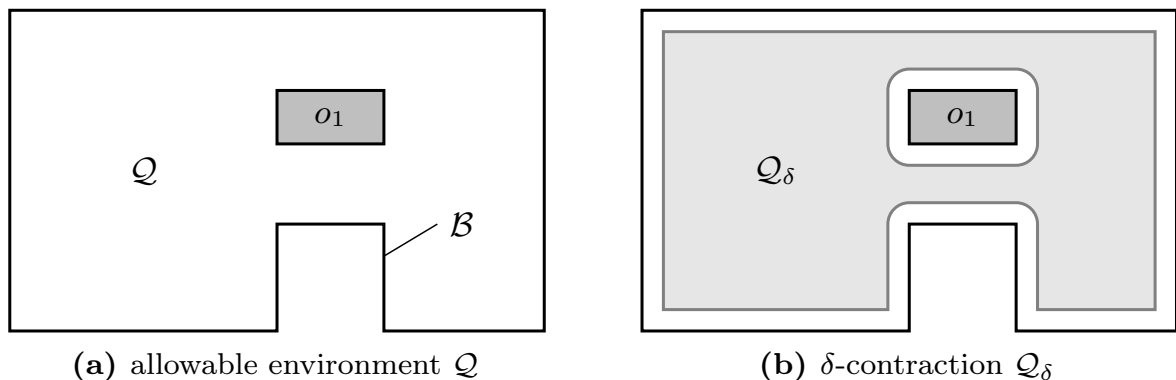
**Definition 2.7** ( $\delta$ -contraction). For an allowable environment  $\mathcal{Q} \subset \mathbb{R}^d$ , the  $\delta$ -**contraction** of  $\mathcal{Q}$  is defined by  $\mathcal{Q}_\delta = \{\mathbf{q} \in \mathcal{Q} \mid \text{dist}(\mathbf{q}, \partial\mathcal{Q}) \geq \delta\}$  with  $\text{dist}(\mathbf{q}, \partial\mathcal{Q}) = \min_{\mathbf{q}' \in \partial\mathcal{Q}} \|\mathbf{q} - \mathbf{q}'\|$ .

The dist-operator defines the minimum Euclidean distance from  $\mathbf{q}$  to the set  $\partial\mathcal{Q}$ .

**Corollary 2.1** (Differentiability of the  $\delta$ -contraction). For arbitrary small  $\delta > 0$  the boundary  $\partial\mathcal{Q}_\delta$  of the  $\delta$ -contraction  $\mathcal{Q}_\delta$  is continuously differentiable in all  $q \in \partial\mathcal{Q}_\delta$  along the boundary  $\partial\mathcal{Q}_\delta$  *except* in the convex vertices. All concave vertices “grow” by  $\delta$ , yielding a differentiable circular segment.

**Corollary 2.2** (Connectedness of the  $\delta$ -contraction). Depending on the magnitude of  $\delta$  applying the  $\delta$ -contraction results in an environment  $\mathcal{Q}_\delta$  that is not necessarily connected.

An example of an allowable environment is shown in Figure 2.2. The allowable environment is defined by a boundary  $\mathcal{B}$  and an obstacle  $o_1$  (cf. Figure 2.2(a)). Its  $\delta$ -contraction is depicted in Figure 2.2(b).



**Figure 2.2:** Example of an allowable environment  $\mathcal{Q}$  and its  $\delta$ -contraction  $\mathcal{Q}_\delta$

### 2.1.3 Reachability and Visibility

In the context of robotic systems, Corollary 2.2 poses a tough limitation: If the  $\delta$ -contraction  $\mathcal{Q}_\delta$  of an allowable environment  $\mathcal{Q}$  is not connected, locations  $p, q \in \mathcal{Q}_\delta$  exist without any path leading from  $p$  to  $q$  in  $\mathcal{Q}_\delta$ . This observation leads to the notion of reachability.

**Definition 2.8** (Reachability). Given a set  $\mathcal{Q} \subset \mathbb{R}^d$  and a point  $p \in \mathcal{Q}$ . A point  $q \in \mathcal{Q}$  is said to be **reachable** from  $p$  if there exists a path from  $p$  to  $q$  that is contained in  $\mathcal{Q}$ . The set of all points  $q \in \mathcal{Q}$  reachable from  $p$  is the **reachability set** with respect to  $p$ .

In addition to the reachability property, the concept of visibility with respect to a point  $p \in \mathcal{Q}$  is introduced.

**Definition 2.9** (Visibility). Given a set  $\mathcal{Q} \subset \mathbb{R}^d$  and a point  $p \in \mathcal{Q}$ . A point  $q \in \mathcal{Q}$  is **visible** from  $p$ , if the closed line segment  $[p, q]$  is contained

in  $\mathcal{Q}$ , i.e.,  $[\mathbf{p}, \mathbf{q}] \subseteq \mathcal{Q}$ . The set of all points  $\mathbf{q} \in \mathcal{Q}$  visible from  $\mathbf{p}$  is the **visibility set** with respect to  $\mathbf{p}$ .

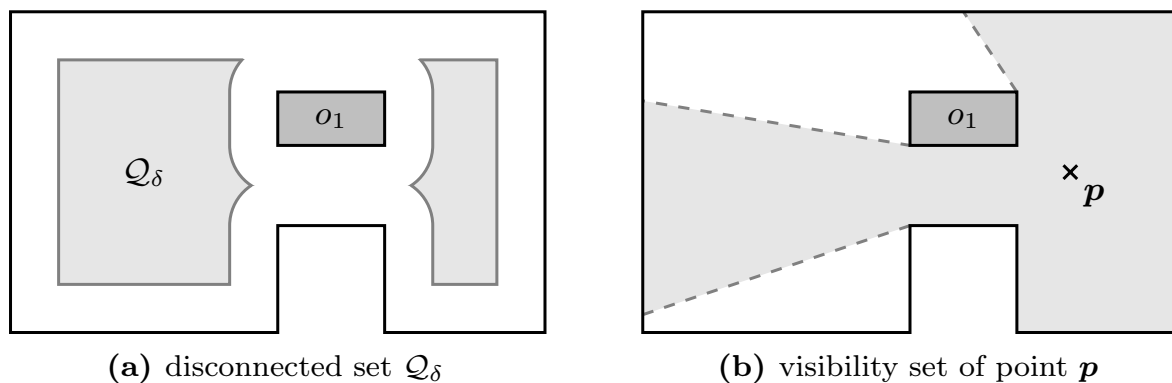
The visibility set can alternatively be described as a star-shaped domain as follows.

**Definition 2.10** (Star-shaped sets). A set  $\mathcal{Q} \subset \mathbb{R}^d$  is called **star-shaped** if there exists a  $\mathbf{p} \in \mathcal{Q}$  such that  $[\mathbf{p}, \mathbf{q}] \subseteq \mathcal{Q}$  for all  $\mathbf{q} \in \mathcal{Q}$ .

A visibility set is star-shaped by definition and represents exactly the parts of the environment that are visible from a robot location. Since the immediately perceived environment of robots is often limited by a radial sensing range  $r$ , it is convenient to define the  $r$ -limited visibility set.

**Definition 2.11** ( $r$ -limited visibility set). Given a set  $\mathcal{Q} \subset \mathbb{R}^d$  and a point  $\mathbf{p} \in \mathcal{Q}$ . The  **$r$ -limited visibility set** with respect to a point  $\mathbf{p} \in \mathcal{Q}$  is defined by intersection of the visibility set at  $\mathbf{p}$  with the closed ball  $\overline{B}(\mathbf{p}, r) = \{\mathbf{q} \in \mathcal{Q} \mid \|\mathbf{p} - \mathbf{q}\| \leq r\}$  at  $\mathbf{p}$  with radius  $r$ .

Figure 2.3 shows the  $\delta$ -contraction  $\mathcal{Q}_\delta$  of the nonconvex allowable environment in Figure 2.2 for a larger  $\delta$ . As a result  $\mathcal{Q}_\delta$  is not connected anymore, meaning that not all points in  $\mathcal{Q}_\delta$  are reachable from all points in  $\mathcal{Q}_\delta$  (cf. Figure 2.3(a)). The (star-shaped) visibility set with respect to a specific point  $\mathbf{p} \in \mathcal{Q}$  is shown in Figure 2.3(b).



**Figure 2.3:** Reachability and visibility set of an allowable environment  $\mathcal{Q}$

### 2.1.4 The Voronoi Partition

The Voronoi partition was first found by G. Lejeune Dirichlet in 1850 [53] and systematically analyzed by Georgy F. Voronoi in 1907 [132]. It is generally defined for both convex and nonconvex environments by using



the geodesic distance  $d_g(\mathbf{p}, \mathbf{q})$ , which reflects the length of the shortest path from  $\mathbf{p}$  to  $\mathbf{q}$  in  $\mathcal{Q}$ .

**Definition 2.12** (Voronoi partition). Given a connected set  $\mathcal{Q} \subset \mathbb{R}^d$ . Let  $\mathcal{P} = \{\mathbf{p}_1, \dots, \mathbf{p}_N\}$  be a set of  $N$  **generator points** in  $\mathcal{Q}$ . Then, the **Voronoi cell**  $\mathcal{V}_i$  of generator point  $\mathbf{p}_i$  is defined by

$$\mathcal{V}_i = \{\mathbf{q} \in \mathcal{Q} \mid d_g(\mathbf{p}_i, \mathbf{q}) \leq d_g(\mathbf{p}_j, \mathbf{q}) \text{ for all } j = 1, \dots, N\}.$$

The set  $\mathcal{V} = \{\mathcal{V}_1, \dots, \mathcal{V}_N\}$  of all Voronoi cells  $\mathcal{V}_i$  is known as the **Voronoi partition** of  $\mathcal{Q}$ .

Informally, a Voronoi cell  $\mathcal{V}_i$  contains all points in  $\mathcal{Q}$  whose geodesic distance to generator point  $\mathbf{p}_i$  is less than the geodesic distance to all other generator points  $\mathbf{p}_{j \neq i}$ . Since Definition 2.12 contains a *less than or equal to* operator, the separating boundaries are assigned to two Voronoi cells. Distinct Voronoi cells  $\mathcal{V}_i$  and  $\mathcal{V}_j$  that share a boundary, i.e.,  $\mathcal{V}_i \cap \mathcal{V}_j \neq \emptyset$ , are called **adjacent** or **Voronoi neighbors**. Apart from the boundaries, all Voronoi cells are disjoint and  $\mathcal{Q} = \cup_{i=1}^N \mathcal{V}_i$  holds, meaning that  $\mathcal{V}$  is a partition of  $\mathcal{Q}$ .

The Voronoi partition of a nonconvex environment is often referred to as **geodesic Voronoi partition**. Further, it is worth to state the following corollary about convex environments.

**Corollary 2.3** (Convex Voronoi partition). If  $\mathcal{Q}$  is a convex set, the geodesic distance  $d_g(\mathbf{p}, \mathbf{q})$  equals the Euclidean norm  $\|\mathbf{p} - \mathbf{q}\|$  and all Voronoi cells are convex.

Related to the Voronoi partition is the **Delaunay triangulation**, introduced in 1934 by Delaunay [52]. The Delaunay triangulation – also known as **Delaunay graph** – is dual to the Voronoi partition [71]. The nodes of this graph are defined by the generator points  $\mathbf{p}_i$ . Two nodes  $\mathbf{p}_i$  and  $\mathbf{p}_j$  are connected with an edge if the respective Voronoi cells  $\mathcal{V}_i$  and  $\mathcal{V}_j$  are adjacent, i.e., if  $\mathbf{p}_i$  and  $\mathbf{p}_j$  are Voronoi neighbors. Voronoi partitions are said to be computable *spatially distributed over the Delaunay graph*. This implies that only the Voronoi neighbors as defined by the Delaunay graph are required in order to compute a Voronoi cell. Hence, the computation of a Voronoi cell is distributed [50]. Since distributed computation is an essential property in distributed systems, Appendix A.2 provides further details on convex Voronoi partitions and an algorithm for a distributed computation of a Voronoi cell.

Analog to the  $r$ -limited visibility set, the  $r$ -limited Voronoi partition is defined next.

**Definition 2.13** (*r*-limited Voronoi Partition [36]). Let  $\mathcal{P} = \{\mathbf{p}_1, \dots, \mathbf{p}_N\}$  be a set of  $N$  generator points in  $\mathcal{Q}$ . Then, the ***r*-limited Voronoi cell**  $\mathcal{V}_i$  of generator point  $\mathbf{p}_i$  is defined by the intersection of Voronoi cell  $\mathcal{V}_i$  with the closed ball at  $\mathbf{p}_i$  with radius  $r$ , i.e.,

$$\mathcal{V}_{i,r} = \mathcal{V}_i \cap \overline{B}(\mathbf{p}_i, r) = \{\mathbf{q} \in \mathcal{V}_i \mid \|\mathbf{p}_i - \mathbf{q}\| \leq r\}.$$

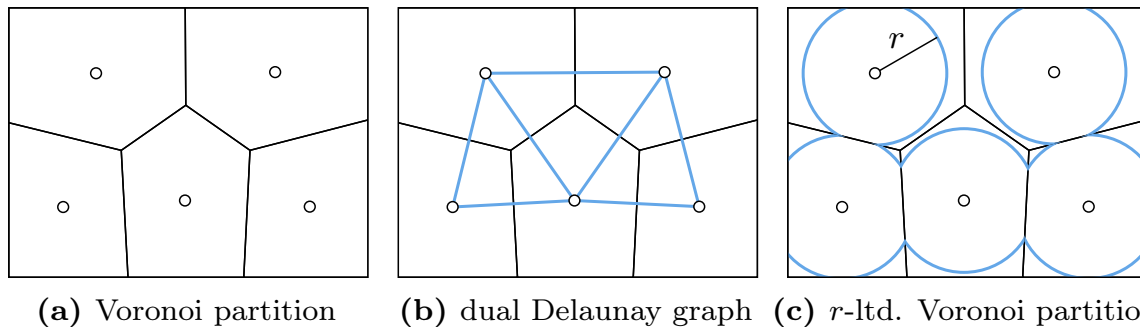
The set  $\{\mathcal{V}_{1,r}, \dots, \mathcal{V}_{N,r}\}$  of all *r*-limited Voronoi cells  $\mathcal{V}_{i,r}$  is known as the ***r*-limited Voronoi partition**. Further, if

$$\mathcal{V}_{i,r} = \overline{B}(\mathbf{p}_i, r)$$

holds, the *r*-limited Voronoi cell  $\mathcal{V}_{i,r}$  is said to be **radially unbounded**.

*Remark 2.2.* Strictly speaking, the *r*-limited Voronoi partition is not a partition of  $\mathcal{Q}$  since generally  $\mathcal{Q} \neq \cup_{i=1}^N \mathcal{V}_{i,r}$ .

Examples of the Voronoi partition, the corresponding Delaunay graph, and the *r*-limited Voronoi partition of a convex set are shown in Figure 2.4.



**Figure 2.4:** Duality of Delaunay graph and Voronoi partition, and *r*-limited Voronoi partition

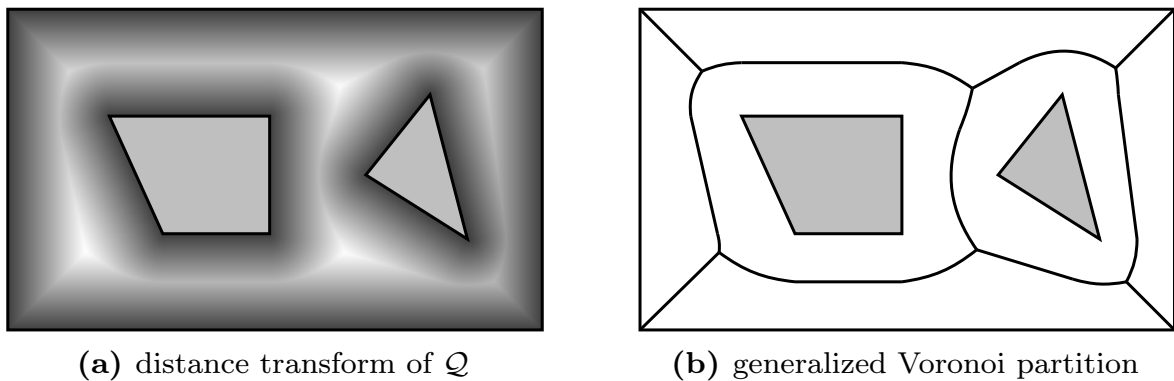
### 2.1.5 Generalized Voronoi Partition

Another common problem in mobile robotics is to plan a collision-free path through nonconvex environments. One such approach to collision-free path planning is given by the distance transform, also referred to as *maximum clearance method* [82].

**Definition 2.14** (Distance transform). Given a nonconvex set  $\mathcal{Q} \subset \mathbb{R}^d$ . The **distance transform** is a function  $f_d : \mathcal{Q} \rightarrow \mathbb{R}_{\geq 0}$  that assigns each point  $\mathbf{p} \in \mathcal{Q}$  the minimum Euclidean distance to the boundary  $\partial\mathcal{Q}$ , i.e.,  $\mathbf{p} \mapsto f_d(\mathbf{p}) = \min_{\mathbf{q} \in \partial\mathcal{Q}} \|\mathbf{p} - \mathbf{q}\|$ .

A distance transform of a nonconvex environment with two obstacles is shown in Figure 2.5(a). Darker shades of gray represent points closer to the boundary.

**Definition 2.15** (Generalized Voronoi partition [81]). Given a nonconvex set  $Q \subset \mathbb{R}^d$ . Let  $\mathcal{D}(\mathbf{p}) = \{\mathbf{q} \in \partial Q \mid \|\mathbf{p} - \mathbf{q}\| = f_d(\mathbf{p})\}$ . With the cardinality  $|\mathcal{D}(\mathbf{p})|$ , define the separating boundary of the **generalized Voronoi partition** as the set  $\{\mathbf{p} \in Q \mid |\mathcal{D}(\mathbf{p})| > 1\}$ .



**Figure 2.5:** (a) Distance transform of a nonconvex environment  $Q \subset \mathbb{R}^2$ , and (b) the corresponding generalized Voronoi partition.

The separating boundary of the generalized Voronoi partition is depicted in Figure 2.5(b). By definition, the boundary always keeps the maximum distance to obstacles. Hence, the path along the boundary minimizes the risk of collisions while navigating a mobile robot through the environment.

## 2.2 Stability Theory

This section introduces the concepts and essentials of dynamical systems and stability theory, closely following Khalil [75]. In addition, these concepts are summarized in detail in Adamy [19].

### 2.2.1 Dynamical Systems

A dynamical system is a process that evolves over time following an ordinary differential equation. Generally, such a system is defined by the system state  $\mathbf{x} \in \mathbb{R}^n$  and a vector-valued function  $\mathbf{f}(\mathbf{x}, \mathbf{u})$  that depends on  $\mathbf{x}$  as well as on the control input vector  $\mathbf{u} \in \mathbb{R}^m$ . Without loss of generality, the

differential equation has the form

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u}). \quad (2.1)$$

The system state  $\mathbf{x}$  contains all variables needed to fully characterize the system. Consequently, the trajectory  $\mathbf{x}(t \geq t_0)$  is uniquely defined for any initial state  $\mathbf{x}(t_0)$  and control input trajectory  $\mathbf{u}(t \geq t_0)$ .

From a feedback control perspective, the course of a trajectory  $\mathbf{x}(t)$  is of interest. To this end, the equilibrium point as well as its characteristics are next defined for autonomous systems  $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{0})$ .

**Definition 2.16** (Equilibrium point). In a dynamical system  $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u})$  the state  $\mathbf{x}_e \in \mathbb{R}^n$  is called **equilibrium point** if  $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}_e, \mathbf{0}) = \mathbf{0}$ .

**Definition 2.17** (Attraction). An equilibrium point  $\mathbf{x}_e$  is called **locally attractive**, if there exists a  $\delta > 0$  such that  $\|\mathbf{x}(t_0) - \mathbf{x}_e\| < \delta$  implies  $\mathbf{x}(t) \rightarrow \mathbf{x}_e$  as  $t \rightarrow \infty$ .

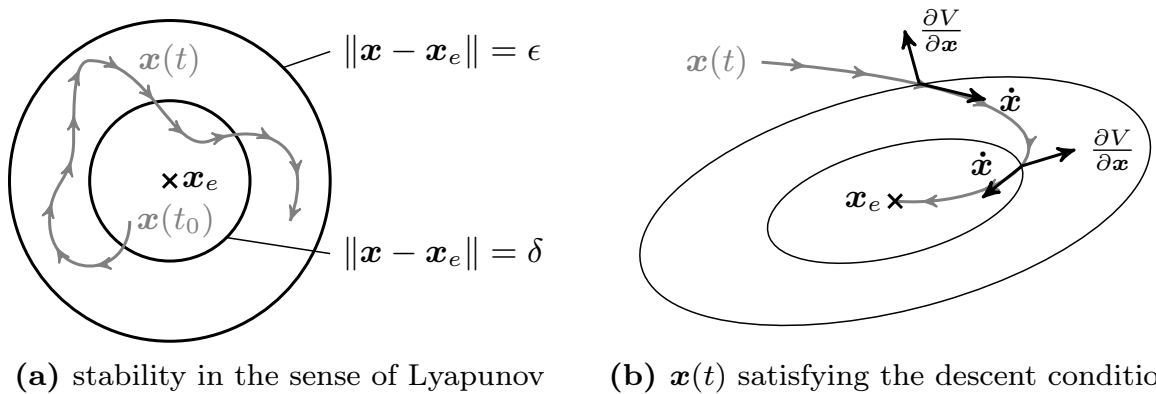
It is worth to note, that the notion of locally attractive equilibrium points allows trajectories satisfying  $\|\mathbf{x}(t_0) - \mathbf{x}_e\| < \delta$  to diverge arbitrarily far away from  $\mathbf{x}_e$  before they finally converge into the equilibrium point. The definition of stability is more restrictive in this regard.

**Definition 2.18** (Lyapunov stability). An equilibrium point  $\mathbf{x}_e \in \mathbb{R}^n$  is called **stable** in the sense of Lyapunov, if for each  $\epsilon > 0$  there exists a  $\delta > 0$  such that  $\|\mathbf{x}(t_0) - \mathbf{x}_e\| < \delta$  implies  $\|\mathbf{x}(t) - \mathbf{x}_e\| < \epsilon$  for all  $t \geq t_0$ . If an equilibrium point is not stable, it is **unstable**. Further, if an equilibrium point  $\mathbf{x}_e$  is stable and locally attractive, it is said to be **asymptotically stable**.

Definition 2.18 introduces a terminology on the stability of an equilibrium point (cf. Figure 2.6(a) for an example of stability in the sense of Lyapunov). However, it does not provide any methods to analyze stability based on the differential equation (2.1). This is addressed in the next section.

## 2.2.2 Lyapunov Functions

The rudimentary idea of Aleksandr M. Lyapunov is to formulate an energy function that describes the overall energy available in the system [89]. Due to friction, the energy level inevitably decreases over time until it reaches a minimum, meaning that an equilibrium point is reached. This idea is generalized in the following definition.



**Figure 2.6:** (a) shows the  $\epsilon$ - $\delta$  stability in the sense of Lyapunov applied to the equilibrium point  $\mathbf{x}_e$ . (b) shows the contour lines of a Lyapunov function  $V$  as well as the descent condition for a trajectory  $\mathbf{x}(t)$  converging towards  $\mathbf{x}_e$ .

**Definition 2.19** (Lyapunov function). Without loss of generality, let  $\mathbf{x}_e = \mathbf{0}$  be an equilibrium point for (2.1) with  $\mathbf{u} = \mathbf{0}$ . A continuously differentiable function  $V : \mathcal{D} \rightarrow \mathbb{R}_{\geq 0}$  on a domain  $\mathcal{D} \subset \mathbb{R}^n$  that satisfies

- (i)  $V(\mathbf{x}) > 0$  for all  $\mathbf{x} \in \mathcal{D} \setminus \{\mathbf{0}\}$  and  $V(\mathbf{0}) = 0$ , and
- (ii)  $\dot{V}(\mathbf{x}) \leq 0$  for all  $\mathbf{x} \in \mathcal{D}$

is called **Lyapunov function**.

The first condition guarantees that a Lyapunov function assigns strictly positive values to all states  $\mathbf{x}$  except in  $\mathbf{x} = \mathbf{0}$ , where it is zero. The second condition implies that the value of  $V$  never increases over time, meaning that the trajectory  $\mathbf{x}(t)$  evolves such that  $V(\mathbf{x})$  either decreases or remains on the same level curve. A more explicit expression is obtained by applying the chain rule  $\dot{V}(\mathbf{x}) = \frac{\partial V}{\partial \mathbf{x}} \frac{d\mathbf{x}}{dt}$ . Consequently, the second condition is equivalent to

$$\frac{\partial V}{\partial \mathbf{x}} \dot{\mathbf{x}} \leq 0. \quad (2.2)$$

Condition (2.2) is also known as *descent condition* and plays a vital role in convex as well as nonconvex optimization problems [33, 61]. If the descent condition strictly holds, i.e.,  $\dot{V}(\mathbf{x}) < 0$ , the angle between the gradient  $\frac{\partial V}{\partial \mathbf{x}}$  and the time derivative  $\dot{\mathbf{x}}$  is greater than  $90^\circ$ , implying that the trajectory  $\mathbf{x}(t)$  decreases the value of the Lyapunov function  $V$ . In case of  $\dot{V}(\mathbf{x}) = 0$  the trajectory stays on the contour line of  $V$ . A Lyapunov function for a trajectory  $\mathbf{x}(t)$  as well as the descent condition is shown in Figure 2.6(b).

In line with Definition 2.18, Lyapunov functions provide a tool to determine stability, as stated by the next theorem.

**Theorem 2.1** (Lyapunov stability theorem [75]). Let  $\mathbf{x}_e = \mathbf{0}$  be an equilibrium point for (2.1) with  $\mathbf{u} = \mathbf{0}$ . If a Lyapunov function exists, the equilibrium point  $\mathbf{x}_e = \mathbf{0}$  is stable in the sense of Lyapunov. If in addition  $\dot{V}(\mathbf{x}) < 0$  holds for all  $\mathbf{x} \in \mathcal{D} \setminus \{\mathbf{0}\}$ , the equilibrium point  $\mathbf{x}_e = \mathbf{0}$  is asymptotically stable.

According to Theorem 2.1 asymptotic stability requires a Lyapunov function whose derivative is negative for any state other than  $\mathbf{x}_e = \mathbf{0}$ . However, this assumption can be relaxed by stating the following theorem.

**Theorem 2.2** (Krasovskii's method [75]). Let  $\mathbf{x}_e = \mathbf{0}$  be an equilibrium point for (2.1) with  $\mathbf{u} = \mathbf{0}$ . If a Lyapunov function exists with  $\dot{V}(\mathbf{x}) \leq 0$ , the equilibrium point  $\mathbf{x}_e = \mathbf{0}$  is asymptotically stable, if all trajectories satisfying  $\dot{V}(\mathbf{x}) = 0$ ,  $\mathbf{x} \neq \mathbf{0}$ , leave the contour line of  $V$  after a finite amount of time.

The proofs for Theorem 2.1 and Theorem 2.2 can be found in [75].

### 2.2.3 Invariance Principle

The idea of equilibrium points and the characteristics as introduced in the previous section can be generalized. To this end, invariant sets are defined.

**Definition 2.20** (Invariant sets). A set  $\mathcal{M}$  is said to be **invariant (positively invariant)**, if all initial values  $\mathbf{x}(t_0) \in \mathcal{M}$  imply  $\mathbf{x}(t) \in \mathcal{M}$  for all  $t$  (for all  $t \geq t_0$ ).

According to Definition 2.20 trajectories  $\mathbf{x}(t)$  that once enter a positively invariant set  $\mathcal{M}$  always remain in  $\mathcal{M}$  in the future. The notion of positive invariance is compatible with Lyapunov functions  $V$ , for instance, the set  $\mathcal{M} = \{\mathbf{x} \in \mathbb{R}^n \mid V(\mathbf{x}) \leq c \in \mathbb{R}_{>0}\}$  with  $\dot{V}(\mathbf{x}) \leq 0$  is positively invariant. Since the course of the trajectory is often of interest, a trajectory is said to **approach** a set  $\mathcal{M}$  if for each  $\epsilon > 0$  exists a  $T > t_0$  such that  $\text{dist}(\mathbf{x}(t), \mathcal{M}) < \epsilon$  for all  $t > T$ . Based on this notion, Krasovskii's Theorem 2.2 can be generalized to invariant sets as follows.

**Theorem 2.3** (Krasovskii-LaSalle invariance principle [75]). Let  $\mathcal{D} \subset \mathbb{R}^n$  be a compact set that is positively invariant with respect to the system dynamics (2.1) with  $\mathbf{u} = \mathbf{0}$ . Let  $V : \mathbb{R}^n \rightarrow \mathbb{R}$  be a continuously differentiable function such that  $\dot{V}(\mathbf{x}) \leq 0$  for all  $\mathbf{x} \in \mathcal{D}$ . Let  $\mathcal{M}_0$  be the set of all points in  $\mathcal{D}$  where  $\dot{V}(\mathbf{x}) = 0$ . Let  $\mathcal{M}$  be the largest invariant set in  $\mathcal{M}_0$ . Then, every solution starting in  $\mathcal{D}$  approaches  $\mathcal{M}$  as  $t \rightarrow \infty$ .

---

The Krasovskii-LaSalle invariance principle generalizes the concept of equilibrium points to invariant sets. At the same time, it provides a tool to prove convergence to  $\mathcal{M}$  for all trajectories starting in  $\mathcal{D}$ . Contrary to the Lyapunov theory, Theorem 2.3 does not strictly require a Lyapunov function  $V$ . Instead, it is sufficient if  $\dot{V}(\boldsymbol{x})$  is negative semidefinite. Further details on invariance principles are presented in Blanchini [29].

## 3 Literature Review

This chapter reviews literature relevant throughout this dissertation. Section 3.1 introduces the solution to the coverage problem as well as extensions to dynamic coverage and nonconvex environments. Section 3.2 gives a survey on multi-robot exploration strategies, focusing on the coordination of multiple robots as well as on communication aspects. Subsequently, Section 3.3 discusses the properties of the presented approaches and adds a remark on performance measures.

### 3.1 The Coverage Problem

The coverage problem belongs to the class of geometric optimization problems. In robotics, its goal is to place a group of mobile robots in an environment in such a way, that – given an objective function – the environment is optimally covered. In the following, existing solutions to this problem are presented – first for convex environments, and later for nonconvex environments.

#### 3.1.1 Introduction

Informally, the coverage objective is to place robots in an a priori known environment such that the expected distance to all possible locations in the environment is minimized. Discussed in detail in [47, 48, 50, 93] by Francesco Bullo, Jorge Cortés, and Sonja Martínez and in depth especially in the textbook [36], a prominent solution to the coverage problem is given by the Lloyd algorithm. The Lloyd algorithm was originally formulated by Stuart P. Lloyd in 1957 and later published in 1982 in [86]. The Lloyd algorithm – meanwhile also known as  $k$ -means clustering [28, 120] (cf. Appendix A.1) – works in its original form according to the following scheme:

**Algorithm 3.1** (Lloyd Algorithm [86]). Given a  $d$ -dimensional compact set  $\mathcal{Q} \subset \mathbb{R}^d$  and a set of  $i = 1, \dots, N$  **generator points**  $p_i \in \mathcal{Q}$ , compute

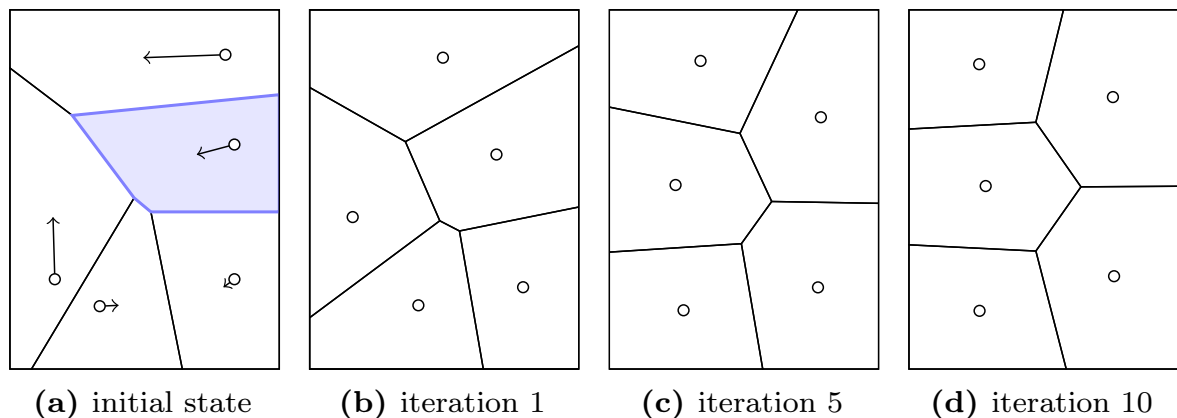


the optimal positions of the generator points  $\mathbf{p}_i$  by iteratively applying the following steps:

- (i) Compute the Voronoi partition based on the generator points.
- (ii) Compute the centroid for each Voronoi cell.
- (iii) Move each generator point  $\mathbf{p}_i$  into the centroid of its Voronoi cell.
- (iv) Terminate on convergence of  $\mathbf{p}_i$ , else continue with (i).

As result, one obtains a **centroidal Voronoi partition**.

Algorithm 3.1 uses the Voronoi partition to decompose the set  $\mathcal{Q}$  into  $N$  disjoint Voronoi cells based on  $N$  generator points. An example of the Lloyd algorithm is depicted in Figure 3.1.



**Figure 3.1:** Lloyd algorithm. The circles denote the generator points. (a) highlights a Voronoi cell, and the arrows point to the centroid of the Voronoi cells. (b) shows the Voronoi partition after one iteration. (c) depicts iteration 5. In (d) the algorithm converges and, thus, terminates.

Bullo et al. [36] apply this **centroidal search** to a set of mobile robots, i.e., each robot corresponds to one generator point in the Lloyd algorithm. In each iteration the robots get closer to the optimal destination by moving towards the centroids. Repeating this scheme, the robots eventually converge into the centroids of the respective Voronoi cells.

The Lloyd algorithm 3.1 depicts an algorithmic approach, where robots always move into the respective centroids in the first step, and *then* the Voronoi partition is updated in a second step. However, robots move continuously in time, and therefore the Voronoi partition also changes continuously. How to analyze this behavior in continuous time, and how to synthesize motion control laws for each robot to move to the respective centroids is subject of the next section.

### 3.1.2 Distributed Optimization Problem

In order to solve the coverage problem, a formal description in terms of a mathematical optimization problem is required. This optimization problem is basically defined by an objective function, quantifying the solution. Prior to formulating the optimization problem, several terms and definitions are introduced following Bullo et al. [36].

Considered is a  $d$ -dimensional compact set  $\mathcal{Q} \subset \mathbb{R}^d$ , which is also referred to as area or environment that is to be covered. If not stated otherwise, the area  $\mathcal{Q}$  is assumed to be convex. Further,  $N$  robots are situated in the area at positions  $\mathbf{p}_i \in \mathcal{Q}$ ,  $i = 1, \dots, N$ . The set  $\mathcal{P} = \{\mathbf{p}_1, \dots, \mathbf{p}_N\}$  is called **configuration**.

Next to the area  $\mathcal{Q}$ , define a **density function**  $\phi : \mathcal{Q} \rightarrow \mathbb{R}_{\geq 0}$ ,  $\mathbf{q} \mapsto \phi(\mathbf{q})$ , that assigns to each location  $\mathbf{q} \in \mathcal{Q}$  a nonnegative scalar real value. The values  $\phi(\mathbf{q})$  can be interpreted as a weighting of the area. A larger weighting implies an increased benefit of covering the location  $\mathbf{q} \in \mathcal{Q}$ . As an example, the density function could indicate the likeliness that an important event occurs at the given location.

Further, a monotonically increasing, continuously differentiable function  $f : \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}$  with a finite number of discontinuities is called **performance function**. Accordingly, larger input values result in monotonically increasing output values. For instance, when passing the distance  $x = \|\mathbf{p}_i - \mathbf{q}\|$  of a robot  $\mathbf{p}_i \in \mathcal{P}$  to an arbitrary location  $\mathbf{q} \in \mathcal{Q}$  as input value, the performance function could be defined as  $f(x) = x^2$ . In fact, this performance function is used by Cortés et al. [50] to solve the centroidal search.

Equipped with these basics, Cortés et al. [50] define an initial variant of the objective function for the coverage problem as  $\mathcal{H} : \mathcal{Q}^N \rightarrow \mathbb{R}$  with

$$\mathcal{H}(\mathcal{P}) = \int_{\mathcal{Q}} \min_{i \in \{1, \dots, N\}} f(\|\mathbf{q} - \mathbf{p}_i\|) \phi(\mathbf{q}) d\mathbf{q}. \quad (3.1)$$

Here, the minimum operator identifies the best performance by choosing the robot position  $\mathbf{p}_i$ ,  $i \in \{1, \dots, N\}$ , that is closest to location  $\mathbf{q} \in \mathcal{Q}$ . Each performance value is then weighted according to the density function  $\phi(\mathbf{q})$ . The final fitness value of the configuration is given by integrating over all locations  $\mathbf{q} \in \mathcal{Q}$ .

The value of  $\mathcal{H}(\mathcal{P})$  in (3.1) solely depends on the choice of the configuration  $\mathcal{P}$ . Therefore, in order to obtain an optimal solution to the coverage problem, a configuration  $\mathcal{P}$  is to be found that minimizes the objective function in (3.1), i.e.,

$$\min \mathcal{H}(\mathcal{P}). \quad (3.2)$$

The optimization problem in (3.2) has the disadvantage, that it cannot easily be computed distributively. However, distributed computation is a required feature, since each robot is supposed to act autonomously. Consequently, it is of interest to find an equivalent formulation of (3.1) such that each robot is capable of optimizing the relevant part of the objective function on its own.

To this end, Cortés et al. [50] partition the environment  $\mathcal{Q}$  into a set of non-empty disjoint cells  $\mathcal{W} = \{\mathcal{W}_1, \dots, \mathcal{W}_N\}$ . Analog to the Voronoi partition, the union of the partition suffices the condition  $\mathcal{Q} = \cup_{i=1}^N \mathcal{W}_i$ . The idea of this partition is that each robot  $i$  maximizes the coverage in its assigned cell  $\mathcal{W}_i$ . This way, the objective function in (3.1) is recast into

$$\mathcal{H}(\mathcal{P}, \mathcal{W}) = \sum_{i=1}^N \int_{\mathcal{W}_i} f(\|\mathbf{q} - \mathbf{p}_i\|) \phi(\mathbf{q}) d\mathbf{q}. \quad (3.3)$$

The objective function in (3.3) has an additional degree of freedom, namely the partition  $\mathcal{W}$ . Since the properties of the partition  $\mathcal{W}$  are not further specified, a location  $\mathbf{q} \in \mathcal{Q}$  might be assigned to a cell  $\mathcal{W}_i$  of robot  $i$ , such that the performance function results in a larger value compared to the objective function in (3.1). As a consequence, next to an optimal configuration  $\mathcal{P}$ , an optimal partition  $\mathcal{W}$  of  $\mathcal{Q}$  is to be found in order to solve the coverage problem.

**Theorem 3.1** (Optimal partition of the environment). Denote the configuration of  $N$  robots as  $\mathcal{P} = \{\mathbf{p}_1, \dots, \mathbf{p}_N\}$  in the area  $\mathcal{Q}$ . Further, let  $f$  be an arbitrary performance function,  $\mathcal{V} = \{\mathcal{V}_1, \dots, \mathcal{V}_N\}$  the Voronoi partition for  $\mathcal{P}$  in  $\mathcal{Q}$ , and  $\mathcal{W} \neq \mathcal{V}$  an arbitrary partition of  $\mathcal{Q}$ . Then,

$$\mathcal{H}(\mathcal{P}, \mathcal{V}) < \mathcal{H}(\mathcal{P}, \mathcal{W})$$

holds, meaning that the Voronoi partition is the optimal partition with respect to the objective function in (3.3).

*Proof.* A proof is given in Bullo et al. [36]. □

Although a formal proof is omitted here, Theorem 3.1 is intuitively clear, since the Voronoi cells  $\mathcal{V}_i$  by definition partition the environment such that the distances to the robot positions  $\mathbf{p}_i$  are minimized. Using Theorem 3.1, the objective function is reformulated to

$$\mathcal{H}(\mathcal{P}) = \mathcal{H}(\mathcal{P}, \mathcal{V}) = \sum_{i=1}^N \int_{\mathcal{V}_i} f(\|\mathbf{q} - \mathbf{p}_i\|) \phi(\mathbf{q}) d\mathbf{q}. \quad (3.4)$$

Since the Voronoi partition  $\mathcal{V}$  is uniquely defined by the robot configuration  $\mathcal{P}$ , it is convenient to omit the parameter  $\mathcal{V}$  and write  $\mathcal{H}(\mathcal{P})$  in (3.4). The objective function  $\mathcal{H}(\mathcal{P})$  in (3.4) is equivalent to the original objective function in (3.1). However, the fundamental difference to (3.1) is that each robot optimizes (3.4) autonomously due to  $\mathcal{H}$  being a composition of  $N$  summands. Hence, all robots perform the optimization *in parallel*, and the optimization process is *spatially distributed* over the Delaunay graph, which is dual to the Voronoi partition.

In summary, based on (3.4) the optimization problem in (3.2) can be rewritten more explicitly in terms of the **distributed optimization problem**

$$\min \mathcal{H}(\{\dots, \mathbf{p}_i, \dots\}) \quad (3.5)$$

for  $i = 1, \dots, N$ . In contrast to (3.2), (3.5) describes  $N$  optimization problems that are coupled through the Voronoi partition  $\mathcal{V}$ . Each robot  $i$  optimizes its part in (3.5) reducing the value of the objective function  $\mathcal{H}(\mathcal{P})$ , which in turn leads to an optimal configuration  $\mathcal{P}$ .

Equation (3.4) together with (3.5) form the fundamental idea of the solution to the coverage problem. The next sections present further details about the density function, the performance function, and the computation of the gradient.

### 3.1.3 A Note on Choosing Weights

The weighting of the environment is described by the density function  $\phi(\mathbf{q})$ . As noted in the previous section, the weighting indicates the significance of locations  $\mathbf{q} \in \mathcal{Q}$ . If normalized, the density function can be interpreted in terms of a probability measure.

Bullo et al. [36] model the density function as a mixture of Gaussians [28]. The Gaussians are placed at locations  $\mathbf{q} \in \mathcal{Q}$  with high likeliness that an important event occurs near  $\mathbf{q}$ . The mixture of Gaussians is constant over time but is allowed to change at distinct switching moments. Cortés et al. [49] further discuss the design of density functions. It is shown how to obtain geometric patterns such as a line or elliptic robotic formations. Finally, the concept is extended to time-varying density functions  $\phi(\mathbf{q}, t)$  that are applied to target-tracking problems [49, 84, 104]. Target-tracking based on time-varying density functions is further elaborated on in Section 3.1.8 in the context of dynamic coverage.

### 3.1.4 Gradient of the Objective Function

Finding an optimal configuration  $\mathcal{P}$  for the distributed optimization problem (3.5) implies the necessary condition

$$\frac{\partial \mathcal{H}(\mathcal{P})}{\partial \mathbf{p}_i} = \mathbf{0} \quad (3.6)$$

for each  $i = 1, \dots, N$ . Therefore, the gradient of  $\mathcal{H}$  with respect to the robot position  $\mathbf{p}_i$  is next derived in detail, since the understanding of how to build the gradient is required again in the discussion about nonconvex coverage. For simplicity, the performance function  $f$  is first modeled as a continuously differentiable function with no discontinuities.

#### Without discontinuities in the performance function

In the distributed optimization problem (3.5), the Voronoi cells  $\mathcal{V}_i$  depend on the robot positions  $\mathbf{p}_i$ , meaning that changing a robot position  $\mathbf{p}_i$  also changes the Voronoi cell  $\mathcal{V}_i$  and the Voronoi cells  $\mathcal{V}_j$  of the neighbors  $j \in \mathcal{N}_i = \{j \in \{1, \dots, N\} \mid \partial \mathcal{V}_i \cap \partial \mathcal{V}_j \neq \emptyset, j \neq i\}$ . This coupling needs to be considered when building the partial derivative of  $\mathcal{H}(\mathcal{P})$  with respect to a robot position  $\mathbf{p}_i$  in (3.6). Therefore, the Leibniz integral rule for differentiation under the integral sign as described by Flanders [60] needs to be applied.

Using an extended form of the Leibniz integral rule for differentiation under the integral sign [60], and defining the position<sup>1)</sup> of robot  $i$  as  $\mathbf{p}_i^\top = [x_i \ y_i]$ , the  $x_i$  component of the gradient (3.6) decomposes to

$$\frac{\partial \mathcal{H}(\mathcal{P})}{\partial x_i} = \sum_{j=1}^N \int_{\mathcal{V}_j} \frac{\partial}{\partial x_i} f(\|\mathbf{q} - \mathbf{p}_j\|) \phi(\mathbf{q}) d\mathbf{q} \quad (3.7a)$$

$$+ \sum_{j=1}^N \int_{\partial \mathcal{V}_j} f(\|\mathbf{q} - \mathbf{p}_j\|) \phi(\mathbf{q}) \mathbf{v}_j^\top \mathbf{n}_j ds, \quad (3.7b)$$

where  $\mathbf{v}_j = \frac{\partial \mathbf{q}}{\partial x_i}$ ,  $\mathbf{q} \in \partial \mathcal{V}_j$ , describes the derivative of boundary points with respect to  $x_i$ , interpreted as velocities of the moving boundaries. Further,  $\mathbf{n}_j$  is the outward facing unit normal on the respective boundary and  $ds$  is the element of arc length [60, 73].

Following Cortés et al. [47], the derivative (3.7) can be simplified as follows: Investigating term (3.7a), all summands  $j \neq i$  are zero with respect

<sup>1)</sup>For simplicity, the planar case  $\mathbf{p}_i \in \mathbb{R}^2$  is considered here.

to differential movements in  $x_i$ . The term (3.7b) describes the differential movement of the boundaries of the Voronoi cells. By observing that only the boundaries of Voronoi cells adjacent to Voronoi cell  $\mathcal{V}_i$  and  $\mathcal{V}_i$  itself change, (3.7b) can be formulated in terms of

$$\int_{\partial\mathcal{V}_i \setminus \partial\mathcal{Q}} f(\|\mathbf{q} - \mathbf{p}_i\|) \phi(\mathbf{q}) \mathbf{v}_i^\top \mathbf{n}_i ds + \sum_{j \in \mathcal{N}_i} \int_{\partial\mathcal{V}_i \cap \partial\mathcal{V}_j} f(\|\mathbf{q} - \mathbf{p}_j\|) \phi(\mathbf{q}) \mathbf{v}_j^\top \mathbf{n}_j ds. \quad (3.8)$$

Therein, the first integral describes the movement of points  $\mathbf{q}$  on the boundary of Voronoi cell  $\mathcal{V}_i$  that do not intersect with the boundary of the environment  $\mathcal{Q}$ . The second integral corresponds to the movement of the boundaries of all Voronoi neighbors  $j \in \mathcal{N}_i$  adjacent to  $\mathcal{V}_i$ . Due to the definition of the Voronoi cells, the integrand  $f(\|\mathbf{q} - \mathbf{p}_i\|) \phi(\mathbf{q}) \mathbf{v}_i^\top \mathbf{n}_i$  and the integrands  $f(\|\mathbf{q} - \mathbf{p}_j\|) \phi(\mathbf{q}) \mathbf{v}_j^\top \mathbf{n}_j$ ,  $j \in \mathcal{N}_i$ , of (3.8) are identical in both integrals, with the exception that the outward normals point in opposite directions, i.e.,  $\mathbf{n}_i = -\mathbf{n}_j$ . Therefore, the term (3.8) is equivalent to

$$\int_{\partial\mathcal{V}_i \setminus \partial\mathcal{Q}} (f(\|\mathbf{q} - \mathbf{p}_i\|) - f(\|\mathbf{q} - \mathbf{p}_j\|)) \phi(\mathbf{q}) \mathbf{v}_i^\top \mathbf{n}_i ds. \quad (3.9)$$

Since by definition  $f(\|\mathbf{q} - \mathbf{p}_i\|) = f(\|\mathbf{q} - \mathbf{p}_j\|)$  holds on the boundary of adjacent Voronoi cells  $\mathcal{V}_i$  and  $\mathcal{V}_j$ , the entire term vanishes, and the derivative (3.7) reduces to

$$\frac{\partial \mathcal{H}(\mathcal{P})}{\partial x_i} = \int_{\mathcal{V}_i} \frac{\partial}{\partial x_i} f(\|\mathbf{q} - \mathbf{p}_i\|) \phi(\mathbf{q}) d\mathbf{q}. \quad (3.10)$$

The partial derivative with respect to movements in the  $y_i$  component is performed analogously to the  $x_i$  component. Therefore, the partial derivative of the objective function  $\mathcal{H}$  with respect to the robot position  $\mathbf{p}_i$  is given by

$$\frac{\partial \mathcal{H}(\mathcal{P})}{\partial \mathbf{p}_i} = \int_{\mathcal{V}_i} \frac{\partial}{\partial \mathbf{p}_i} f(\|\mathbf{q} - \mathbf{p}_i\|) \phi(\mathbf{q}) d\mathbf{q}. \quad (3.11)$$

*Remark 3.1.* The value of the objective function  $\mathcal{H}(\mathcal{P})$  depends on the robot configuration  $\mathcal{P}$  and consequently the Voronoi partition. However, the gradient of  $\mathcal{H}$  with respect to a robot  $\mathbf{p}_i$  in (3.11) solely depends on information available in the respective Voronoi cell  $\mathcal{V}_i$ . Therefore, the optimization of  $\mathcal{H}$  can be performed *distributively* based on the Voronoi partition of  $\mathcal{Q}$ .

### Allowing discontinuities in the performance function

In preparation for the  $r$ -limited centroidal search, Cortés et al. [47] allow a finite set of discontinuities in the performance function  $f$ . These discontinuities need to be considered when building the gradient (3.11). Cortés et al. [47] denote with  $a \in \text{dscn}(f)$  the set of discontinuities of  $f$  and define the one-sided limits

$$f_-(a) = \lim_{x \rightarrow a^-} f(x), \quad f_+(a) = \lim_{x \rightarrow a^+} f(x).$$

The gradient (3.11) of  $\mathcal{H}$  with respect to the robot position  $\mathbf{p}_i$  then extends to

$$\begin{aligned} \frac{\partial \mathcal{H}(\mathcal{P})}{\partial \mathbf{p}_i} &= \int_{\mathcal{V}_i} \frac{\partial}{\partial \mathbf{p}_i} f(\|\mathbf{q} - \mathbf{p}_i\|) \phi(\mathbf{q}) d\mathbf{q} + \\ &\sum_{a \in \text{dscn}(f)} (f_-(a) - f_+(a)) \int_{\mathcal{V}_i \cap \partial \bar{B}(\mathbf{p}_i, a)} \mathbf{n}_{\text{out}, \bar{B}(\mathbf{p}_i, a)}(\mathbf{q}) \phi(\mathbf{q}) d\mathbf{q}. \end{aligned} \quad (3.12)$$

The vector  $\mathbf{n}_{\text{out}, \bar{B}(\mathbf{p}_i, a)}(\mathbf{q})$  describes a normal unit vector in  $\mathbf{q}$  on the boundary of the intersection of the closed ball  $\bar{B}(\mathbf{p}_i, a)$  with the Voronoi cell  $\mathcal{V}_i$ . All these normal vectors are weighted according to the density  $\phi(\mathbf{q})$ , and then integrated and multiplied by the magnitude of the jump discontinuity.

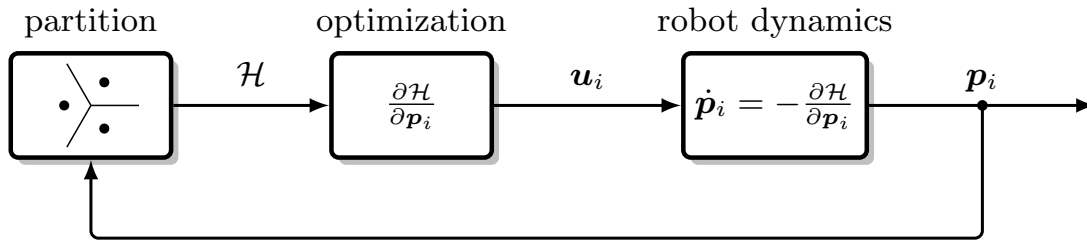
*Remark 3.2.* Condition (3.6), i.e.,  $\frac{\partial \mathcal{H}}{\partial \mathbf{p}_i} = \mathbf{0}$ , constitutes only a necessary condition, implying that a configuration  $\mathcal{P}$  may represent a saddle point or a maximum of the optimization problem. Cortés et al. [50] mention, that a sufficient condition for a minimum is a positive definite Hessian of  $\mathcal{H}$  at  $\mathbf{p}_i$  for all  $i = 1, \dots, N$ . This is further elaborated on in Du et al. [54].

#### 3.1.5 Continuous-Time Motion Control Laws

Next, it is of interest how to use the gradient of the objective function  $\mathcal{H}$  with respect to the robot positions to derive motion control laws moving the robots into the centroids. Cortés et al. [47] assume simple single-integrator dynamics for each robot according to

$$\dot{\mathbf{p}}_i = \mathbf{u}_i.$$

This dynamics describes the kinematics of a point mass of first order with motion control law  $\mathbf{u}_i$  of robot  $i$ . The control input  $\mathbf{u}_i$  is computed by



**Figure 3.2:** Feedback loop of the coverage problem (cf. Haumann et al. [4])

*continuously* optimizing the gradient of the objective function, i.e.,

$$\mathbf{u}_i = -\frac{\partial \mathcal{H}(\mathcal{P})}{\partial \mathbf{p}_i}, \quad |\mathbf{u}_i| \leq u_{\max}. \quad (3.13)$$

The condition  $|\mathbf{u}_i| \leq u_{\max}$  reflects input amplitude constraints, since a robot's actuators, and thus also its maximum speed, are limited. The motion control laws (3.13) work according to the feedback-loop in Figure 3.2: By continuously computing the Voronoi cell  $\mathcal{V}_i$ , each robot evaluates the objective function based on solely local information available in its Voronoi cell, resulting in motion control laws  $\mathbf{u}_i$  that move the robot such that the optimization problem is successively solved. As a consequence, one obtains optimal robot positions maximizing the coverage. The differences between the feedback loop in Figure 3.2 and the feedback loop in Figure 1.1 (page 4) are that the mapping block is omitted, the optimization is defined as the gradient with respect to the robot positions, and the robot dynamics are defined in terms of gradient-based motion control laws.

These motion control laws distributively coordinate the group of robots based on the Voronoi partition. Therefore, both design criteria – the *distributed coordination* and the *communication constraints* – stated in the problem formulation in Section 1.2 (page 3) are met.

### 3.1.6 Performance Functions

The only missing part in the distributed optimization problem is the choice of the performance function  $f$ . Following Bullo et al. [36], several performance functions and their behavior are discussed.

#### Centroidal Search

The performance function for the centroidal search is chosen as  $f(x) = x^2$  without discontinuities (cf. Figure 3.3(a)). Substituting  $\mathcal{H}$  in (3.4) with



this performance function results in

$$\mathcal{H}_{\text{cover}}(\mathcal{P}) = \sum_{i=1}^N \int_{\mathcal{V}_i} \|\mathbf{q} - \mathbf{p}_i\|^2 \phi(\mathbf{q}) d\mathbf{q} = \sum_{i=1}^N J_\phi(\mathcal{V}_i, \mathbf{p}_i). \quad (3.14)$$

$J_\phi(\mathcal{V}_i, \mathbf{p}_i)$  is the polar moment of inertia at  $\mathbf{p}_i$  of the weighted area  $\mathcal{V}_i$ . Applying the parallel axis theorem [30], (3.14) can be written as

$$\mathcal{H}_{\text{cover}}(\mathcal{P}) = \sum_{i=1}^N J_\phi(\mathcal{V}_i, \mathbf{m}_\phi(\mathcal{V}_i)) + \sum_{i=1}^N A_\phi(\mathcal{V}_i) \|\mathbf{p}_i - \mathbf{m}_\phi(\mathcal{V}_i)\|^2, \quad (3.15)$$

where the function  $A_\phi(\mathcal{V}_i) = \int_{\mathcal{V}_i} \phi(\mathbf{q}) d\mathbf{q}$  denotes the weighted area of Voronoi cell  $\mathcal{V}_i$ , and the function  $\mathbf{m}_\phi(\mathcal{V}_i) = A_\phi^{-1}(\mathcal{V}_i) \int_{\mathcal{V}_i} \mathbf{q} \phi(\mathbf{q}) d\mathbf{q}$  denotes the weighted center of mass – also called *centroid* – of  $\mathcal{V}_i$ . In the special case where  $\mathbf{p}_i = \mathbf{m}_\phi(\mathcal{V}_i)$  the second term in (3.15) vanishes. This is equivalent to the configuration  $\mathcal{P} = \{\mathbf{m}_\phi(\mathcal{V}_1), \dots, \mathbf{m}_\phi(\mathcal{V}_N)\}$  and represents a **centroidal Voronoi configuration**.

As the first term in (3.15) does not depend on the position  $\mathbf{p}_i$ , the gradient of  $\mathcal{H}_{\text{cover}}$  with respect to  $\mathbf{p}_i$  reads

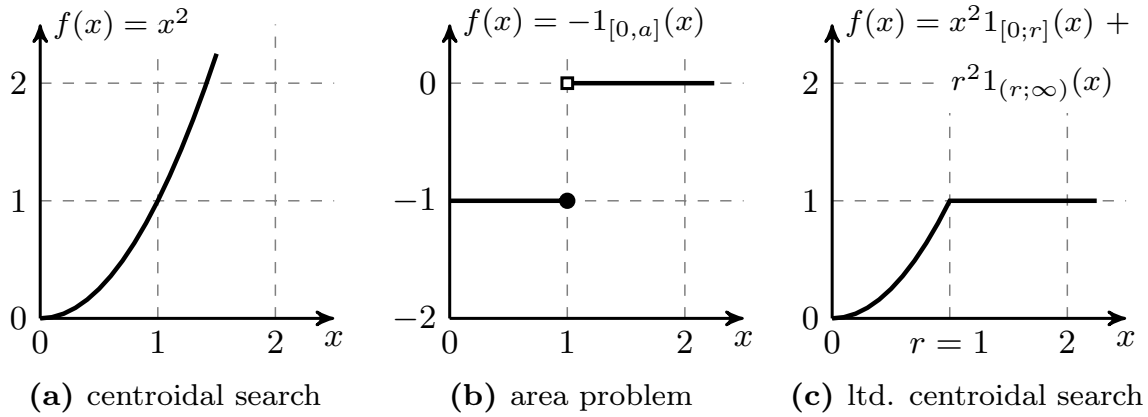
$$\frac{\partial \mathcal{H}_{\text{cover}}(\mathcal{P})}{\partial \mathbf{p}_i} = \underbrace{2A_\phi(\mathcal{V}_i)}_{k_i} (\mathbf{p}_i - \mathbf{m}_\phi(\mathcal{V}_i)). \quad (3.16)$$

The characteristic property of this gradient is that the vector  $\mathbf{p}_i - \mathbf{m}_\phi(\mathcal{V}_i)$  points straight from the centroid into the robot position, which reflects exactly the desired behavior of the centroidal search when inserted in (3.13). Further, the scalar  $k_i$  can be interpreted as a proportional factor in the control input (3.13), defining the magnitude of how fast the robot moves towards the centroid. In principle, an example of the centroidal search in discrete-time is given in the introduction to the coverage problem in Figure 3.1 (cf. Section 3.1.1, page 21) for a rectangular region with constant weights.

### Area Problem

Instead of finding the centroids, subject to the area problem is to find robot positions  $\mathbf{p}_i$  that maximize the covered area. To this end, Cortés et al. [47] introduce the performance function

$$f(x) = -1_{[0;a]}(x) = \begin{cases} -1, & \text{if } x \in [0; a], \\ 0, & \text{otherwise,} \end{cases} \quad (3.17)$$



**Figure 3.3:** Performance functions of the centroidal search, the area problem for  $a = 1$ , and the limited centroidal search for  $r = 1$ .

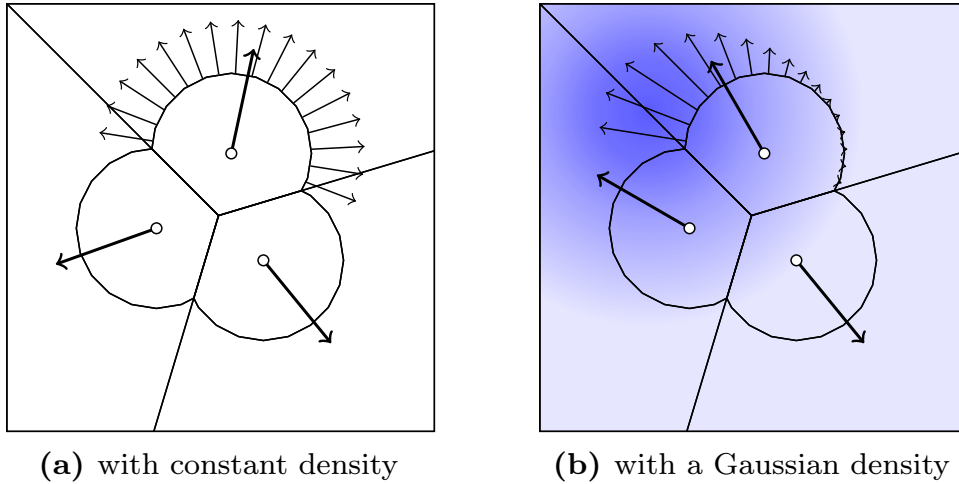
with  $a \in \mathbb{R}_{>0}$ . Depicted in Figure 3.3(b), the performance function (3.17) acts as an *indicator function* and has the same effect as using an  $r$ -limited Voronoi partition with radius  $a = r$ . As a consequence, the objective function turns into

$$\begin{aligned}
 \mathcal{H}_{\text{area}}(\mathcal{P}) &= - \sum_{i=1}^N \int_{\mathcal{V}_i} 1_{[0;r]}(\|\mathbf{q} - \mathbf{p}_i\|) \phi(\mathbf{q}) d\mathbf{q} \\
 &= - \sum_{i=1}^N \int_{\mathcal{V}_i \cap \bar{B}(\mathbf{p}_i, r)} \phi(\mathbf{q}) d\mathbf{q} \\
 &= - \sum_{i=1}^N A_\phi(\mathcal{V}_i \cap \bar{B}(\mathbf{p}_i, r)). \tag{3.18}
 \end{aligned}$$

Obviously,  $\mathcal{H}_{\text{area}}$  represents the weighted area covered by all robots, implying that minimizing (3.18) drives the robots away from each other to minimize the overlap. Since performance function (3.17) has a single discontinuity in  $a = r$  with magnitude  $-1$  and is constant otherwise, the gradient in (3.12) reduces to

$$\frac{\partial \mathcal{H}_{\text{area}}(\mathcal{P})}{\partial \mathbf{p}_i} = - \int_{\mathcal{V}_i \cap \partial \bar{B}(\mathbf{p}_i, r)} \mathbf{n}_{\text{out}, \bar{B}(\mathbf{p}_i, r)}(\mathbf{q}) \phi(\mathbf{q}) d\mathbf{q}. \tag{3.19}$$

Informally, this gradient is defined as the sum of all negative normal unit vectors on the boundary of the closed ball in the respective Voronoi cell, weighted with the density. These normal vectors as well as the resulting negative gradient are illustrated in Figure 3.4 both for a constant and a Gaussian density function.



**Figure 3.4:** Gradient of the area problem

### Limited Centroidal Search

A mix of both the centroidal search and the area problem is defined by the performance function

$$f(x) = x^2 1_{[0;a]}(x) + a^2 1_{(a;\infty)}(x), \quad (3.20)$$

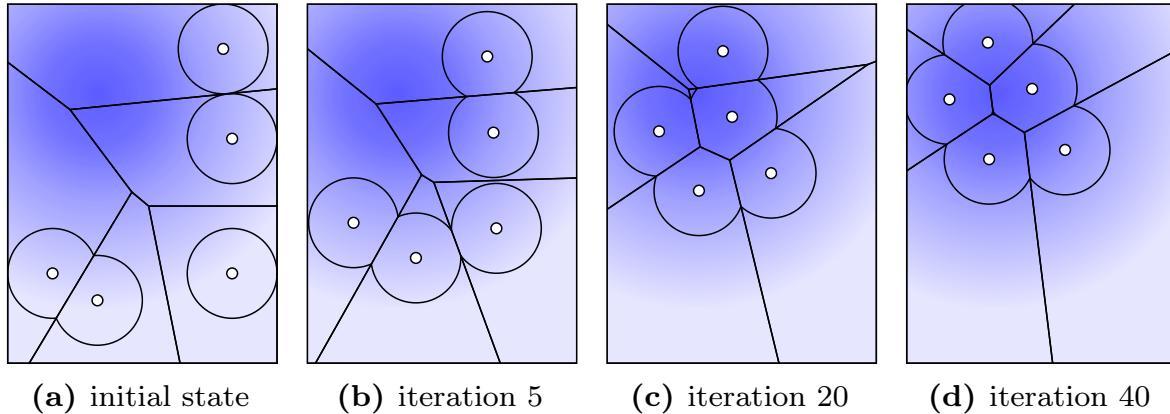
which is continuous in  $a$ , but not continuously differentiable (cf. Figure 3.3(c)). For  $a = r$  and the  $r$ -limited Voronoi cell  $\mathcal{V}_{i,r} = \mathcal{V}_i \cap \overline{B}(\mathbf{p}_i, r)$ , the objective function reads

$$\begin{aligned} \mathcal{H}_{\text{lim}}(\mathcal{P}) &= \sum_{i=1}^N \int_{\mathcal{V}_{i,r}} \|\mathbf{q} - \mathbf{p}_i\|^2 \phi(\mathbf{q}) d\mathbf{q} + \sum_{i=1}^N \int_{\mathcal{V}_i \setminus \mathcal{V}_{i,r}} r^2 \phi(\mathbf{q}) d\mathbf{q} \\ &= \sum_{i=1}^N J_\phi(\mathcal{V}_{i,r}, \mathbf{p}_i) + r^2 \sum_{i=1}^N A_\phi(\mathcal{V}_i \setminus \mathcal{V}_{i,r}). \end{aligned} \quad (3.21)$$

Contrary to the performance function (3.17),  $f$  in (3.20) is continuous. And since  $f$  is continuous, the one-sided limits in  $f(x \rightarrow a = r)$  are equal, and therefore no outward normals appear when building the gradient according to (3.12). Consequently, the second term in (3.21) vanishes, and the gradient of (3.21) reduces to the centroidal search in the  $r$ -limited Voronoi cell  $\mathcal{V}_{i,r}$  (cf. Definition 2.13, page 14). As result, the gradient of  $\mathcal{H}_{\text{lim}}$  with respect to the robot position  $\mathbf{p}_i$  yields

$$\frac{\partial \mathcal{H}_{\text{lim}}(\mathcal{P})}{\partial \mathbf{p}_i} = 2A_\phi(\mathcal{V}_{i,r})(\mathbf{p}_i - \mathbf{m}_\phi(\mathcal{V}_{i,r})). \quad (3.22)$$

Compared to the centroidal search in a Voronoi cell, the gradient now points straight from the centroid of the  $r$ -limited Voronoi cell into  $\mathbf{p}_i$ , weighted by the factor  $k_i = 2A_\phi(\mathcal{V}_{i,r})$ . An example of the limited centroidal search is depicted in Figure 3.5.



**Figure 3.5:** Weighted centroidal search with limited Voronoi cells. Darker colors imply higher weights. The initial states match the ones in Figure 3.1, page 21.

### 3.1.7 Stability Analysis

In the following, a proof of convergence for the limited centroidal search is given as suggested by Cortés et al. [50]. As stated in (3.13), the motion control law for each robot is defined as

$$\dot{\mathbf{p}}_i = \mathbf{u}_i = -\frac{\partial \mathcal{H}(\mathcal{P})}{\partial \mathbf{p}_i}, \quad |\mathbf{u}_i| \leq u_{\max}. \quad (3.23)$$

Based on these motion control laws, the goal is to apply the Krasovskii-LaSalle invariance principle (cf. Theorem 2.3, page 18). This invariance principle requires the time derivative of the objective function  $\mathcal{H}_{\text{lim}}(\mathcal{P})$  to be negative semi-definite such that it decreases over time. Building the time derivative of  $\mathcal{H}_{\text{lim}}(\mathcal{P})$  results in

$$\dot{\mathcal{H}}_{\text{lim}}(\mathcal{P}) = \frac{\partial \mathcal{H}_{\text{lim}}(\mathcal{P})}{\partial \mathcal{P}} \dot{\mathcal{P}} = \sum_{i=1}^N \frac{\partial \mathcal{H}_{\text{lim}}(\mathcal{P})}{\partial \mathbf{p}_i} \dot{\mathbf{p}}_i, \quad (3.24)$$

that is, the scalar product in (3.24) can be decomposed into a sum of  $N$  scalar products. Inserting the partial derivative (3.22) and (3.23) into (3.24),

one obtains the inequality

$$\dot{\mathcal{H}}_{\text{lim}}(\mathcal{P}) = -k_p \sum_{i=1}^N \|\mathbf{p}_i - \mathbf{m}_\phi(\mathcal{V}_{i,r})\|^2 \leq 0 \quad (3.25)$$

with a positive gain  $k_p$ . Inequality (3.25) always holds due to the sum of squares and  $k_p$  being positive. Using (3.25) the Krasovskii-LaSalle invariance principle can be applied. Thereafter,  $\mathcal{H}_{\text{lim}}(\mathcal{P})$  is minimized over time, and consequently the robots move to the largest invariant set, which equals the set of all centroidal Voronoi configurations.

### 3.1.8 From Static to Dynamic Coverage

Up to this point, a time-invariant density function  $\phi$  was assumed such that the robots converge to a static configuration. This setup is known as the *static* coverage problem. However, in large-scale environments or a priori unknown environments, an increased amount of robots may be required in order to satisfy sufficient coverage of the entire domain. In this case, the solution to the coverage problem is impractical [25].

This dilemma is addressed by the *dynamic coverage problem*. Batalin and Sukhatme [25] define dynamic coverage as the constant motion of robots such that the entire area is repeatedly covered over time. Interestingly, covering the entire environment over time matches the definition of the exploration problem. Consequently, solutions to the dynamic coverage problem are also of interest in the context of multi-robot exploration.

Hussein and Stipanovic [70] solve the problem of dynamic coverage by initializing each point in the domain with a coverage value. The goal is to reach a certain coverage level for the entire domain. This is achieved by a gradient-based motion control law that incorporates the information within the sensing range. If the coverage level is reached within the sensing range, the gradient is zero. In this case the controller is switched to move to the closest point where the coverage level is not met. Extensions add repulsive forces to the control law that avoid collisions among the robots. However, nonconvex environments that are typical for the exploration problem are not considered.

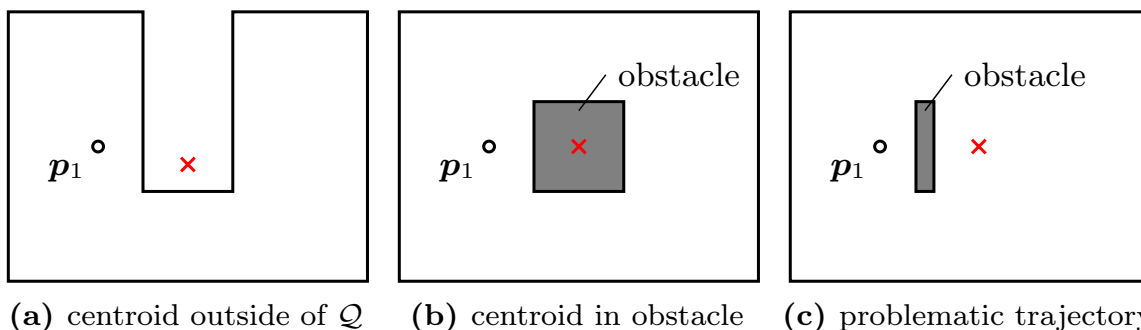
A related approach to dynamic coverage was proposed by Cortés et al. [49] back in 2002 to solve the target-tracking problem: The target is represented through a time-varying density function, such that the solution to the coverage problem as introduced in Section 3.1 can be applied. Consequently, instead of converging to a static configuration, the robots

continuously move, following the density. A drawback of this approach is that convergence to the target cannot be guaranteed, since the dynamics of the time-varying density function is unknown. Pimenta et al. [104] extend this idea in 2008 by providing a gradient-based controller that provably converges to the target under the assumption that the dynamics of the target is partially known. Independently of Pimenta et al. [104], the same solution was proposed again in 2013 by Lee and Egerstedt [84].

Dynamic coverage is also considered in [25, 113, 118] but the proposed solutions are not of interest in the context of multi-robot exploration as discussed in this dissertation.

### 3.1.9 Coverage of Nonconvex Environments

In essence, the robots always move to the centroid of the respective ( $r$ -limited) Voronoi cell in order to solve the coverage problem. However, in nonconvex allowable environments, simply moving to the centroid is not directly applicable. Figure 3.6 depicts a robot in nonconvex allowable environments  $\mathcal{Q}$  with constant density. The robot is supposed to maximize coverage in  $\mathcal{Q}$  by moving to the centroid of  $\mathcal{Q}$ . However, this approach is problematic for the following reasons: In Figure 3.6(a) the centroid lies outside of the allowable environment  $\mathcal{Q}$ , in Figure 3.6(b) the centroid of  $\mathcal{Q}$  lies inside an obstacle, and in Figure 3.6(c), although the centroid lies inside  $\mathcal{Q}$ , the resulting trajectory collides with the obstacle. Hence, simply moving to the centroid may result in collisions with the nonconvex environment and therefore a modified approach to nonconvex coverage is required.



**Figure 3.6:** Problematic locations of the centroid  $\times$  of nonconvex allowable environments for a single robot

A simple approach to nonconvex coverage is presented by Pimenta et al. [103]. In essence, the authors use the geodesic Voronoi partition and apply

the geodesic distance for each point  $\mathbf{q} \in \mathcal{V}_i$  to  $\mathbf{p}_i$  in the performance function. The main contribution of Pimenta et al. [103] is a transformation that projects each point  $\mathbf{q} \in \mathcal{Q}$  assigned to a robot at  $\mathbf{p}_i$  to  $\mathbf{q}' = \mathbf{p}_i + d_g(\mathbf{p}_i, \mathbf{q})\mathbf{n}_{\mathbf{p}_i, \mathbf{q}}$ , where  $\mathbf{n}_{\mathbf{p}_i, \mathbf{q}}$  denotes the unit normal of the first path segment along the geodesic path from  $\mathbf{p}_i$  to  $\mathbf{q}$ . The mapping of all  $\mathbf{q} \mapsto \mathbf{q}'$  avoids moving into obstacles. At discontinuities of the geodesic path the normal  $\mathbf{n}_{\mathbf{p}_i, \mathbf{q}}$  is undefined. In this case, the authors fall back to Clarke's generalized gradient [45]. As a result, the authors claim to obtain the optimal coverage of the partition in nonconvex environments. In addition to this approach to nonconvex coverage, Pimenta et al. [103] also present an approach to *convex* coverage based on the *power diagram* [23], which is an extension to the Voronoi partition. Using power diagrams, the authors solve the coverage problem for a group of robots with heterogeneous sensing range, i.e., each robot  $i$  may have a distinct sensing range  $r_i$ . Focusing on the area problem in nonconvex environments, the idea of using power diagrams was proposed only recently in Thanou et al. [124]. Further, nonconvex coverage solving the area problem in the geodesic Voronoi partition was also investigated in Thanou et al. [125].

The method by Breitenmoser et al. [34] uses the Voronoi partition with the Euclidean norm to assign to each robot the centroid of its Voronoi cell. If obstacles block the direct path to the centroids, the robots use the TangentBug algorithm [111] to avoid collisions. Further, if a centroid lies inside an obstacle, it is projected to the nearest point on the boundary of the obstacle, leading to the best possible approximation.

Another approach to solving the nonconvex coverage problem is presented by Caicedo-Núñez and Žefran [40]. The authors apply a diffeomorphism to nonconvex environments to obtain convex regions. However, as a side effect the diffeomorphism changes the mass distribution. Consequently the optimality property of the centroids in the nonconvex environment is lost, yielding a suboptimal solution to the nonconvex coverage problem. This issue is addressed by Caicedo-Núñez and Žefran [39], where the authors prove the existence of an appropriate diffeomorphism. Nonetheless, it is generally unclear how to compute the transformation, rendering this approach in many applications computationally unfeasible.

### Nonconvex Coverage in Visibility Sets

Also related to coverage in nonconvex environments is the problem presented by Zhong and Cassandras [138]. The authors consider a modified coverage framework, where overlapping sensor areas increase the quality of coverage,

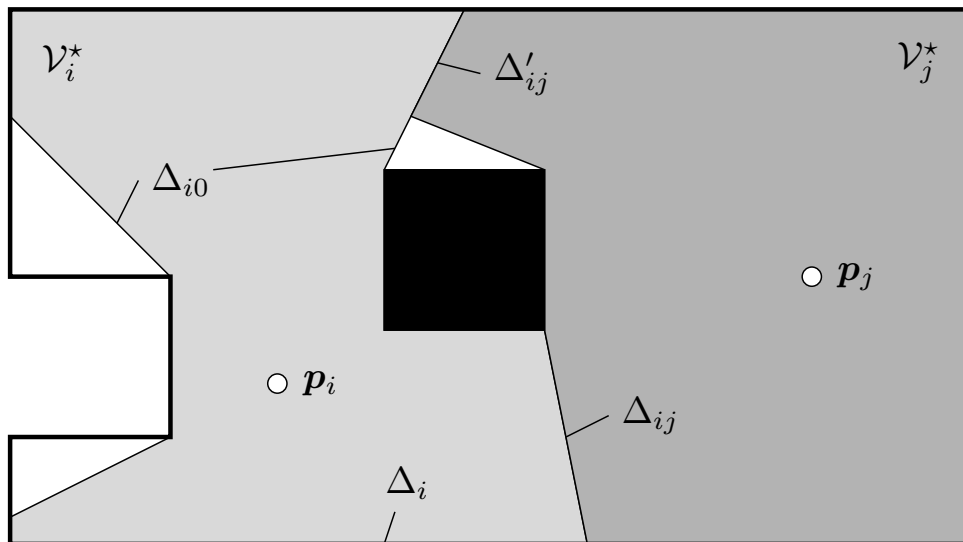
and the environment with polygonal obstacles is no longer partitioned into Voronoi cells. The approach is related to coverage in visibility sets in that obstacles attenuate or block the sensing abilities of the robots, leading to nonsmooth analysis of the gradient of the objective function. Zhong and Cassandras [138] show that the use of a generalized gradient [45] is necessary due to a new term that arises from the boundary to invisible regions.

The combination of visibility-based coverage in the Voronoi partition has only been considered recently by Lu et al. [88] and Marier et al. [92]. Lu et al. [88] propose to evaluate the coverage integral in the unlimited visibility set in the geodesic Voronoi cell. Following this approach, the resulting centroid may lie outside of the area  $\mathcal{Q}$ , e.g., inside an obstacle. To work around this issue, the authors project the centroid to the closest point in the Voronoi cell that is visible from the respective robot to avoid collisions. A mathematical analysis and a proof of convergence is not provided by Lu et al. [88], though.

Marier et al. [92] also use the notion of visibility sets in Voronoi cells similar to Lu et al. [88], with the difference that they allow disconnected cells. The authors closely investigate the gradient of the objective function similar to Zhong and Cassandras [138]. Since the gradient in visibility sets is of importance for the multi-robot exploration approach proposed in Section 4, it is investigated here following Marier et al. [92]. To this end, two robots located in  $\mathbf{p}_i$  and  $\mathbf{p}_j$  and their respective visibility sets,  $\mathcal{V}_i^* = \{\mathbf{q} \in \mathcal{V}_i \mid [\mathbf{p}_i, \mathbf{q}] \subset \mathcal{V}_i\}$  and  $\mathcal{V}_j^*$ , are depicted in Figure 3.7. The boundary of the visibility sets can be distinguished into several boundary types:

- (i)  $\Delta_i$  represents the boundary that is limited by obstacles or the area  $\mathcal{Q}$ . It is invariant with respect to robot movements, i.e., it remains unchanged when  $\mathbf{p}_i$  changes.
- (ii)  $\Delta_{ij} = \Delta_{ji}$  defines the boundary whose distance of points  $\mathbf{q} \in \Delta_{ij}$  to  $\mathbf{p}_i$  equals the distance to  $\mathbf{p}_j$ . In other words,  $\Delta_{ij}$  equals the definition of the separating bisectors defining the Voronoi cells.
- (iii)  $\Delta_{i0}$  is defined by all boundary points  $\mathbf{q} \in \partial\mathcal{V}_i^*$  that are adjacent to uncovered area. The uncovered area  $\mathcal{Q}_0$  is the set of points  $\mathbf{q} \in \mathcal{Q}$  that are not visible from any robot position  $\mathbf{p}_i$ .
- (iv)  $\Delta'_{ij} = \Delta'_{ji}$  defines the boundary parts where two visibility sets are adjacent. However, in contrast to (ii), the distance of points  $\mathbf{q} \in \Delta'_{ij}$  to  $\mathbf{p}_i$  does *not* equal the distance to  $\mathbf{p}_j$ .





**Figure 3.7:** Nonconvex coverage in visibility sets

In convex sets, the boundary types (iii) and (iv) do not exist, and as shown in the derivation of the gradients in Section 3.1.4 (page 25), the outward normals generated by the boundary type (ii) cancel out such that only outward normal vectors in discontinuities in the performance function  $f$  remain. In contrast to (3.9) (page 26), when building the gradient of the objective function  $\mathcal{H}$  in visibility sets, the boundary types (iii) and (iv) do not cancel out. As shown by Marier et al. [92], the gradient in this case with respect to the  $x_i$  component of the robot position  $\mathbf{p}_i^\top = [x_i \ y_i]$  for the performance function  $f(x) = x^2$  results in

$$\frac{\partial \mathcal{H}(\mathcal{P}, \mathcal{V})}{\partial x_i} = \sum_{j=1}^N \int_{\mathcal{V}_j^*} \frac{\partial}{\partial x_i} \|\mathbf{q} - \mathbf{p}_j\|^2 \phi(\mathbf{q}) d\mathbf{q} \quad (3.26a)$$

$$+ \sum_{j=1}^N \int_{\Delta'_{ij}} \|\mathbf{q} - \mathbf{p}_j\|^2 \phi(\mathbf{q}) \mathbf{v}_j^\top \mathbf{n}_j ds \quad (3.26b)$$

$$+ \sum_{j=1}^N \int_{\Delta_{j0}} D^2 \phi(\mathbf{q}) \mathbf{v}_0^\top \mathbf{n}_0 ds \quad (3.26c)$$

$$+ \int_{\mathcal{Q}_0} \frac{\partial}{\partial x_i} D^2 \phi(\mathbf{q}) d\mathbf{q} \quad (3.26d)$$

where again  $\mathbf{v}_j = \frac{\partial \mathbf{q}(\partial \mathcal{V}_j^*)}{\partial x_i}$  and  $\mathbf{v}_0 = \frac{\partial \mathbf{q}(\Delta_{j0})}{\partial x_i}$  are the derivatives of boundary points  $\mathbf{q} \in \partial \mathcal{V}_j^*$  and  $\mathbf{q} \in \Delta_{j0}$  with respect to  $x_i$ , interpreted as velocities of the moving boundaries. Further,  $\mathbf{n}_j$  and  $\mathbf{n}_0$  are the outward facing unit

normals on the respective boundaries,  $ds$  is the element of arc length [60, 73], and  $D \in \mathbb{R}_{\geq 0}$  is a scalar that can for instance be set to  $\text{diam } \mathcal{Q}$ . As before, all elements of the sum in (3.26a) for  $j \neq i$  are zero. However, the terms (3.26b)–(3.26d) are non-zero, leaving partial derivatives  $\frac{\partial \mathcal{H}}{\partial \mathbf{p}_i}$  that are significantly more complex to compute in practice. In addition, the boundary parts  $\Delta'_{ij}$  can be computed only by knowing the visibility set of neighboring robots, which implies increased communication payload. Marier et al. [92] use these gradients to find optimal configurations with respect to coverage in visibility sets and also provide a proof of convergence for environments without holes. The problem that agents might leave the environment due to the normal vectors (3.26b) and (3.26c) persists and is again handled by a projection procedure.

Another research area is the problem of maximizing the visibility set of a nonconvex area by deploying a minimum amount of sensors. This deployment task is also known as the *art gallery problem* [44], where the sensors are cameras that monitor an art gallery. Although the visibility problem is akin to the coverage problem [36, 65, 100], the objectives and the corresponding optimization problems and solutions of deployment tasks are not equivalent.

### Further Research

Additional research about optimization problems in nonconvex as well as convex environments exists, e.g., [20, 43, 69, 80, 93, 107, 121, 123, 137, 138]. Nevertheless, since the relevance is limited in the context of multi-robot exploration as addressed in this dissertation, these approaches are not further discussed.

## 3.2 Multi-Robot Exploration

Multi-robot systems potentially solve the exploration task more efficiently while at the same time being more fault-tolerant and more reliable [42, 55]. As motivated in the introduction of the dissertation, these advantages only hold if the robots effectively coordinate each other. In accordance with the solution to the coverage problem in Section 3.1, the coordination of a group of robots is typically achieved by casting the exploration task into an optimization problem based on an appropriate objective function. This objective function consists of multiple components that rely on the currently available information about the environment, such as map data

and robot positions. Therefore, the coordination of the group inherently relies on a reliable communication topology in addition to the optimization problem.

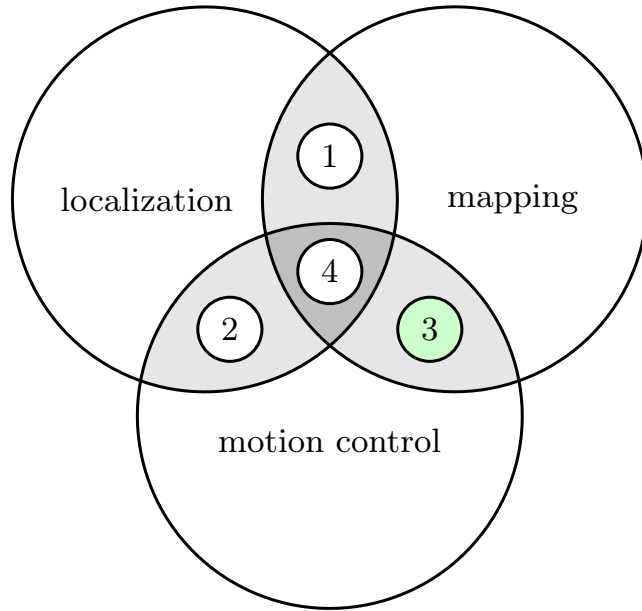
The next section introduces the fundamental problem of simultaneous localization and mapping. After that, objective functions and communication constraints of existing exploration approaches are discussed first for a single robot and then for a group of robots. Finally, a discussion follows including gradient-based exploration strategies and performance measures.

### 3.2.1 Simultaneous Localization and Mapping

Makarenko et al. [91] divide the exploration problem according to Figure 3.8 into three tasks: *mapping*, *localization* and *motion control*. Mapping is the process of extending the map with new information gained about the environment. However, mapping an unknown environment requires an accurate robot position, otherwise it is unclear *where* to extend the map. Consequently, mapping is tightly coupled to the localization task: Inaccurate localization implies a bad quality of the map and vice versa. This dependency is depicted by region ① in Figure 3.8. In addition, a bad localization quality also has an impact on the motion control (cf. region ②): If a robot has a faulty estimate of its position, motion control may lead to collisions with obstacles in the environment, which in turn implies a coupling of motion control and mapping (cf. region ③). The interaction of these three tasks is represented by region ④ in Figure 3.8 and was first introduced by Leonard and Durrant-Whyte [85] by stating “*which came first, the chicken or the egg? (The map or the motion?)*” before it became known as *Simultaneous Localization and Mapping* (SLAM) [85, 128].

In fully autonomous exploration, a robot needs to take care of all three tasks (cf. region ④ in Figure 3.8). In practice, this is a non-trivial process due to non-ideal sensors and actuators, and inaccuracies in the generated map of the environment. Solutions to SLAM are discussed in Bailey and Durrant-Whyte [24], Durrant-Whyte and Bailey [57], Thrun and Leonard [128], and also in Thrun et al. [127].

This dissertation focuses on exploration approaches, meaning that an ideal localization and mapping process is assumed. Therefore, the exploration problem considered reduces to the “two-part definition” [114] (i) *where to move next* and (ii) *how to get there*, which is represented by green region ③ in Figure 3.8.



**Figure 3.8:** Simultaneous localization and mapping (SLAM) as in [91]

### 3.2.2 Objective Functions for a Single Robot

The surveys of Burgard et al. [38] and Stachniss et al. [119] refer to a variety of exploration strategies that focus on a single robot. In all these approaches, the robot moves to the unknown region that is closest to the robot. In essence, this equals the frontier-based approach as introduced by Yamauchi [136]: For a single robot at position  $\mathbf{p}$ , the objective function of this minimum distance strategy and the corresponding optimization problem read

$$\mathcal{H}_{\text{MinDist}}(\mathbf{p}, \mathbf{q}) = d_g(\mathbf{p}, \mathbf{q}) \quad \Rightarrow \quad \arg \min_{\mathbf{q} \in \tilde{s}} \mathcal{H}_{\text{MinDist}}(\mathbf{p}, \mathbf{q}). \quad (3.27)$$

The objective function  $\mathcal{H}_{\text{MinDist}}$  solely consists of the geodesic distance costs  $d_g$  from the robot position  $\mathbf{p}$  to  $\mathbf{q}$  on the frontier  $\tilde{s}$ . Makarenko et al. [91] extend this objective function by two components: the localization quality and the expected information gain on the frontier. The localization quality accounts for the SLAM problem, so that the robot maintains an accurate position estimate during the exploration. The expected information gain represents the amount of unknown space that the robot is expected to explore when reaching a target point.

While these approaches work for a single robot, they are not directly applicable to the multi-robot domain, since the corresponding objective

functions do not contain any ingredients to coordinate multiple robots. Therefore, the next section reviews exploration strategies explicitly designed for multiple robots.

### 3.2.3 On Coordinating Multiple Robots

Moors [96] proposes a centralized approach to coordinated multi-robot exploration. Assuming unconstrained communication capabilities, a robot chooses a target point on the frontier by optimizing a *utility function* that consists of the expected information gain on the frontier and the distance costs. In order to prevent other robots in the group from moving to the same location, the expected information gain at the region around the chosen target point is reduced and communicated to the group. The reduction of the expected information gain minimizes the potential overlap when choosing target points. This approach was applied to multi-robot exploration of indoor environments and published by Burgard et al. [37] and Simmons et al. [116]. A hierarchical version of this approach is presented by Burgard et al. [38]. Here, robots within communication range form a cluster and apply the same algorithm to choose target points for each robot.

The assumption of a shared map is further relaxed in Fox et al. [63, 64] by enabling multiple robots to explore an indoor environment from different, unknown locations. Due to the unknown initial locations, robots within communication range actively seek to verify their relative positions. Whenever relative positions are verified, the robots form a cluster and continue exploration in a group following Burgard et al. [38]. Analog to Makarenko et al. [91] for a single robot, these approaches account for the SLAM problem with multiple robots.

Other approaches focus on maintaining a communication network during the exploration task. Simulations are provided by Rooker and Birk [108]. Assuming limited communication range, the authors add a component to the objective function that prevents robots from moving too far away from the group. However, deadlock situations may occur that need to be detected and resolved by applying the rendez-vous principle [110]. This drawback does not exist in the work of Sheng et al. [115], as a distributed bidding model is applied that coordinates the movement of the robots such that distance constraints are satisfied in each point in time. Experimental results are given by Rekleitis et al. [106]. Here, the communication constraints are satisfied by using a fixed amount of robots that collaborate by moving through the environment in a predefined pattern.

### 3.2.4 Coordination by Partitioning the Environment

Following the solution to the coverage problem, another concept to robot coordination is to spatially partition the environment such that each robot is assigned a region of dominance to be explored. Consequently, coordinately choosing target points reduces to finding appropriate partitions.

Solanas and Garcia [117] partition the unknown parts of the environment by forming spatially disjoint clusters with the  $k$ -means clustering algorithm [86, 120], where the amount of clusters matches the amount of robots. Each robot is then assigned one cluster. The objective function is formulated such that each robot moves to a target point on the frontier of its cluster that minimizes the distance costs to the robot. Wu et al. [134] essentially propose the same algorithm, except that the objective function additionally contains a term that prevents a robot to get too close to frontiers of neighboring clusters. This approach was again extended in Wu et al. [135]. However, drawbacks of these approaches are the central computation of the cluster centers in unknown space and the assumption of unconstrained communication. Therefore, without further modifications these approaches are not applicable in distributed systems.

Bhattacharya et al. [27] transfer the solution to the coverage problem to explore nonconvex environments based on the geodesic Voronoi partition. Each robot computes the centroid of its geodesic Voronoi cell. This centroid is then projected to the closest point in the respective cell where the density function is higher than a specific threshold value. This projection method guarantees that over time the robots move to all locations in the environment. The density function is defined in terms of the Shannon entropy. The entropy is a measure for the uncertainty in a point: In unknown space, the uncertainty is highest, whereas it is zero in explored regions. Moving to the projected centroid, the robots permanently reduce the entropy according to their limited range sensor model, which in turn influences the position of the centroid. The authors give a proof for convergence under the assumption that all robots can communicate with their geodesic Voronoi neighbors and exchange map data as well as robot positions from time to time.

Other approaches to multi-robot exploration as well as related research exist (e.g., [102, 107, 112, 131]). However, the relation to the DisCoverage approaches presented in this dissertation is minor and therefore these publications are not further discussed.

## 3.3 Discussion

Neither the related work presented for the coverage problem nor the one presented for the exploration problem are exhaustive. Instead, it emphasizes the huge variety of solutions to both problems with different focus and assumptions. In the context of multi-robot exploration, most of these strategies (e.g., [37, 38, 63, 64, 116, 134, 135]) follow the two-step approach: First, target points are chosen, *then* path planning algorithms are applied to reach the destinations. Contrary, many solutions to the coverage problem apply gradient-based motion control laws similar to Cortés et al. [50] as illustrated by the closed feedback loop in Figure 3.2 (page 28). Therein, only a single optimization problem is continuously optimized, resulting in motion control laws that solve the coverage problem. A distributed coordination technique to the multi-robot exploration problem following the solution to the coverage problem was only recently investigated by Bhattacharya et al. [27]. However, as discussed the authors need a projection method as well as path planning to prevent robots from moving into obstacles, again introducing additional steps next to solving the optimization problem.

### 3.3.1 Gradient-Based Motion Control

According to Koren and Borenstein [77], potential field methods have several inherent limitations for mobile robot navigation. Next to oscillations due to narrow passages or obstacles, a major drawback are trap situations due to local minima in the optimization problem. Therefore, gradient-based motion control needs a careful design in order to avoid these disadvantages. This holds especially in distributed systems, since no entity with global knowledge exists.

With respect to the coverage problem in Section 3.1, the solution provided by Bullo et al. [36] yields optimal configurations that maximize the overall coverage level. However, the final robot configuration significantly depends on the choice of the density function and the initial robot configuration  $\mathcal{P}$ , and since the robots apply gradient-based control laws, it is not guaranteed to find the globally optimal robot configuration.

Therefore, a multi-robot exploration approach based on the solution to the coverage problem satisfying the design criteria of distributed coordination under well-defined communication constraints (cf. Section 1.2 on page 3) needs to be carefully designed such that full exploration of the environment is always guaranteed.

### 3.3.2 A Note on Performance Measures

Surprisingly, a very limited amount of research directs its attention to performance measurements. Generally, performance is measured in terms of time needed to complete the exploration process, which usually corresponds to minimizing the overall travel distance. Such measurements are performed in Amigoni [21] for selected strategies in a simulation environment. As a result, the author claims that exploration strategies that balance cost and potential information gain perform better than the ones that solely consider the potential information gain *or* the distance costs. Other research that examines robotic systems exists: Lee and Recce [83] performs experiments for a variety of exploration strategies with only a single robot. Kramer and Scheutz [78] focus on robotic development environments and middleware solutions, Michael et al. [94] describe an experimental testbed for multi-robot teams, and Kudelski et al. [79] provide a framework with realistic communication models. However, extensive performance tests on multi-robot exploration strategies are not performed.

As a consequence, the exploration strategies proposed in the next chapter are compared to the time-optimal case. Although the time-optimal case provides a measure of optimality from a purely theoretical perspective, it is still suitable for reference, as it provides a lower bound for the time needed to explore the entire environment.



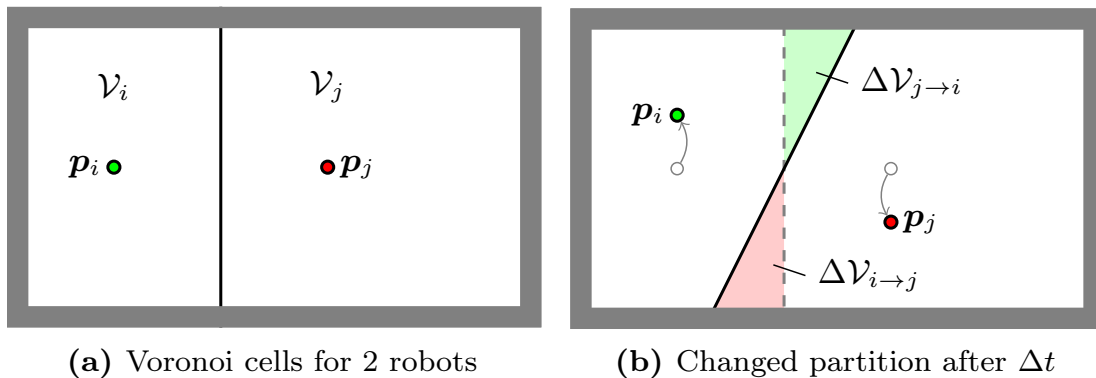
## 4 From Coverage to DisCoverage

Motivated by the solution to the coverage problem, this chapter introduces new approaches to distributed multi-robot exploration. The proposed approaches rely upon spatially distributed optimization problems and therefore closely follow the closed feedback loop in Figure 3.2 (page 28). As a result, gradient-based motion control laws are obtained that cause the robots to fully explore the environment, without any need of explicit path planning algorithms. Since the idea of gradient-based motion control laws as well as the underlying distributed optimization problem originate from the coverage problem, the presented approaches are referred to as **DisCoverage approaches**. The results of the centroidal search-based DisCoverage approach in Section 4.2 were published in Haumann [5], Haumann et al. [8, 9] and Klodt et al. [12]. Similarly, the orientation-based DisCoverage approach was discussed in Frank et al. [2], Haumann et al. [3, 4] and Haumann et al. [6].

### 4.1 Preliminaries and Notation

Throughout this chapter, planar allowable environments  $\mathcal{Q} \subset \mathbb{R}^2$  according to Definition 2.6 (page 10) are considered, i.e.,  $\mathcal{Q}$  is a connected set defined by a polygonal environment with polygonal obstacles. Further, it is assumed that  $\mathcal{Q}$  remains constant over time, implying a static environment without moving obstacles. The explored parts of the environment are defined as  $\mathcal{S}(t) \subseteq \mathcal{Q}$ , where the frontier is defined as  $\tilde{s}(t) = \partial\mathcal{S}(t) \setminus \partial\mathcal{Q}$ . Initially, the entire environment is unexplored and  $\mathcal{S}(t=0) = \emptyset$  holds. The robots are denoted by the configuration  $\mathcal{P}(t) = \{\mathbf{p}_1(t), \dots, \mathbf{p}_N(t)\}$  with robot positions  $\mathbf{p}_i(t) \in \mathcal{Q}$ . The frontier in the region of dominance  $\mathcal{V}_i(\mathcal{P}(t))$  for robot  $i$  is given as  $\tilde{s}_i(\mathcal{P}(t), t) = \tilde{s}(t) \cap \mathcal{V}_i(\mathcal{P}(t))$ . Typically,  $\mathcal{V}_i(\mathcal{P}(t))$  refers to the respective (geodesic) Voronoi cell in convex (nonconvex) allowable environments  $\mathcal{Q}$ . For ease of notation, the dependency on time  $t$  and on the configuration  $\mathcal{P}$  is omitted in most places, resulting in the variables  $\mathcal{S}$ ,  $\tilde{s}$ ,  $\tilde{s}_i$ ,  $\mathcal{P}$  and  $\mathcal{V}_i$ .

Closely following the assumptions of Bullo et al. [36], all robots are equipped with omni-directional sensors with sensing range  $r$  that perform ideal measurements in order to map the environment. Furthermore, the problem of simultaneous localization and mapping (SLAM) is avoided by neglecting any uncertainties in the robot positions as discussed in Section 3.2.1 (page 39). During the exploration process, bidirectional communication among Voronoi neighbors is assumed. Consequently, the robots maintain a communication network spatially distributed over the Delaunay graph, which is dual to the Voronoi partition as described in Section 2.1.4 (page 12). Robots of adjacent Voronoi cells communicate each other's position and exchange map data to account for the time-varying Voronoi cells according to Figure 4.1. For  $\Delta t \rightarrow 0$ , neighbors  $i, j$  differentially exchange map data of the areas  $\delta\mathcal{V}_{i \rightarrow j}$ , i.e.,  $\Delta\mathcal{V}_{i \rightarrow j}$  in Figure 4.1 approaches  $\delta\mathcal{V}_{i \rightarrow j}$ . Therewith, each robot  $i$  is aware of the explored parts  $\mathcal{S}_i$  and the frontier  $\tilde{s}_i$  in its region of dominance  $\mathcal{V}_i$  in each point in time.



**Figure 4.1:** Exchanged map data among Voronoi neighbors

## 4.2 Centroidal Search-Based DisCoverage

The limited centroidal search introduced in Section 3.1 (page 20) presents an elegant solution to solve the coverage problem with limited sensing and communication range. In literature, this approach is applied to solve the coverage problem (e.g., [36, 50]) or target tracking problems (e.g., [49, 104]), mostly focusing on convex environments. The goal of this section is to modify the limited centroidal search such that it can be applied to solve the exploration problem with a group of robots. Therefore, in convex environments, the objective function in (3.21) on page 31 is

slightly modified in terms of

$$\begin{aligned} \mathcal{H}_{\text{discover}}(\mathcal{P}, \mathcal{S}) = & \sum_{i=1}^N \int_{\mathcal{V}_{i, \bar{r}}} \|\mathbf{q} - \mathbf{p}_i\|^2 \phi(\mathbf{q}, \tilde{s}_i) d\mathbf{q} + \\ & \bar{r}^2 \sum_{i=1}^N \int_{\mathcal{V}_i \setminus \mathcal{V}_{i, \bar{r}}} \phi(\mathbf{q}, \tilde{s}_i) d\mathbf{q}. \end{aligned} \quad (4.1)$$

The modifications in (4.1) with respect to  $\mathcal{H}_{\text{lim}}$  in (3.21) are twofold: First, the density  $\phi(\mathbf{q}, \tilde{s}_i(\mathcal{P}(t), t))$  is now a function of the time-varying frontier  $\tilde{s}_i$ . It is worth to note that the density  $\phi(\mathbf{q}, \tilde{s}_i)$  was never proposed before: Instead, Bhattacharya et al. [27], Cortés et al. [49], Lee and Egerstedt [84], Pimenta et al. [104] propose a density  $\phi(\mathbf{q}, t)$  as a function of time. Second, a scalar **integration range**  $\bar{r} > 0$  is introduced, which is used to compute the  $\bar{r}$ -limited Voronoi cells

$$\mathcal{V}_{i, \bar{r}} = \{\mathbf{q} \in \mathcal{V}_i \mid \|\mathbf{p}_i - \mathbf{q}\| \leq \bar{r}\}. \quad (4.2)$$

The integration range  $\bar{r}$  does not correspond to any sensing capabilities. A proper interpretation is given in later sections. In what follows, the idea is to adapt the density function  $\phi(\mathbf{q}, \tilde{s}_i)$  over time so that the motion control laws

$$\dot{\mathbf{p}}_i = \mathbf{u}_i = -\frac{\partial \mathcal{H}_{\text{discover}}(\mathcal{P}, \mathcal{S})}{\partial \mathbf{p}_i}, \quad i = 1, \dots, N, \quad (4.3)$$

solve the exploration problem. To this end, Section 4.2.1 formally defines the density function  $\phi(\mathbf{q}, \tilde{s}_i)$  for convex and nonconvex allowable environments. The partial derivative of  $\mathcal{H}_{\text{discover}}$  in (4.3) is given in Section 4.2.2 for convex environments. Further, Section 4.2.3 discusses the properties of the integration range  $\bar{r}$  in detail, and a proof of convergence for the proposed exploration approach is given. Thereafter, the proposed approach is generalized to nonconvex allowable environments (cf. Section 4.2.4). Section 4.2.5 investigates the behavior when using multiple robots. Finally, Section 4.2.6 discusses possible extensions to the vehicle dynamics.

### 4.2.1 Modifying the Density Function

As proven in Section 3.1.7 (page 32), the objective function of the limited centroidal search along with the gradient-based motion control laws move the robots to locations with high density, maximizing coverage in convex environments. Therewith, it is fundamental that the density function

$\phi : \mathcal{Q} \rightarrow \mathbb{R}$  is designed such that its values are maximal in the critical locations and monotonically decrease with increasing distance to the critical points.

For the coverage problem, Cortés et al. [49] propose a Gaussian density function  $\phi : \mathcal{Q} \rightarrow \mathbb{R}_{\geq 0}$  that is interpreted in terms of a probability density function. As a result, the objective function

$$\mathcal{H}(\mathcal{P}) = \sum_{i=1}^N \underbrace{\int_{\mathcal{V}_i} f(\|\mathbf{q} - \mathbf{p}_i\|) \phi(\mathbf{q}) d\mathbf{q}}_{\text{expected value}} \quad (4.4)$$

essentially computes  $N$  expected values which equal the expected mean performance from a probability theory perspective. Therefore, Bullo et al. [36] refer to (4.4) as an *expected-value multicenter function* that is optimal for centroidal Voronoi configurations  $\mathcal{P}$ .

Martínez et al. [93] define  $\phi$  as a mixture of Gaussians, allowing several distinct locations with maximum density. Next to a mixture of Gaussians, Cortés et al. [49] propose density functions that form elliptic or line shapes (cf. Section 3.1.3, page 24). Combining these ideas with the time-varying density functions  $\phi(\mathbf{q}, t)$  in [27, 49, 84, 104], it seems natural to define the density as a function of the time-varying frontier in terms of the Gaussian mapping  $\phi : \mathcal{Q} \times \mathcal{Q} \rightarrow \mathbb{R}_{\geq 0}$ ,

$$\phi(\mathbf{q}, \tilde{s}_i(\mathcal{P}(t), t)) = \begin{cases} \exp\left(-\frac{1}{2\sigma^2} d_g^2(\mathbf{q}, \tilde{s}_i(\mathcal{P}(t), t))\right), & \text{if } \mathbf{q} \in \mathcal{S}_i, \\ 1, & \text{if } \mathbf{q} \notin \mathcal{S}_i, \end{cases} \quad (4.5)$$

where  $\sigma \in \mathbb{R}_{>0}$  denotes the standard deviation. According to (4.5), the density in unexplored space is equal to one. In the convex case, the metric  $d_g(\mathbf{q}, \tilde{s}_i(\mathcal{P}(t), t)) = \min_{\mathbf{q}' \in \tilde{s}_i(\mathcal{P}(t), t)} \|\mathbf{q} - \mathbf{q}'\|$  computes the minimum Euclidean distance from  $\mathbf{q}$  to the time-varying frontier  $\tilde{s}_i(\mathcal{P}(t), t)$  through explored space  $\mathcal{S}_i(\mathcal{P}(t), t)$ . In the nonconvex case,  $d_g(\mathbf{q}, \tilde{s}_i(\mathcal{P}(t), t)) = \min_{\mathbf{q}' \in \tilde{s}_i(\mathcal{P}(t), t)} d_g(\mathbf{q}, \mathbf{q}')$  computes the minimum geodesic distance from  $\mathbf{q}$  to the time-varying frontier. Since the density function is a Gaussian, its value equals one on the frontier and in unexplored space. In explored space, the value of the density function monotonically decreases with increasing distance to the frontier, depending on the value of the standard deviation  $\sigma$ .

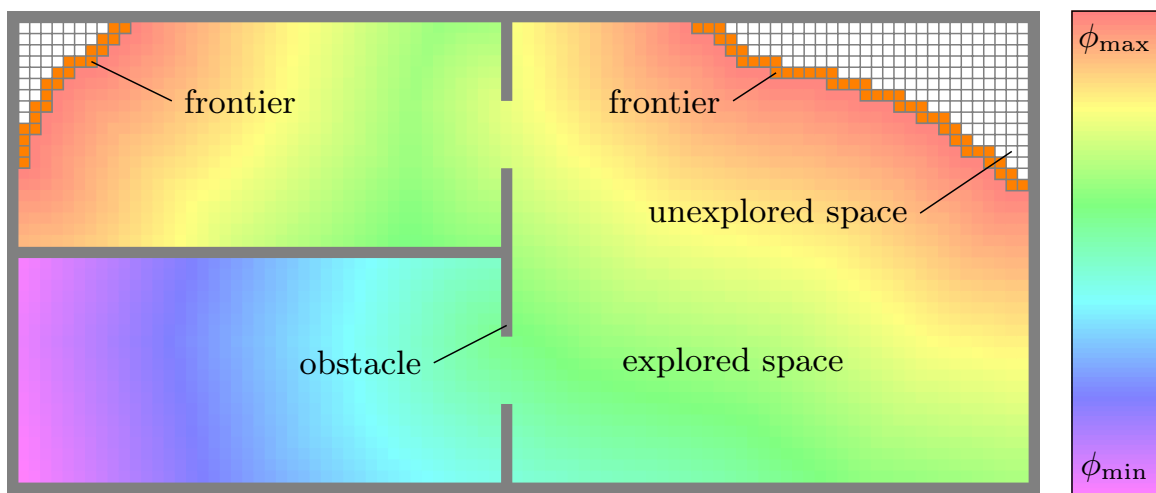
Further densities are applicable, such as the gamma distribution with rate parameters both equal to one (cf. Bishop [28]). In fact, the requirements on the weighting can be relaxed to monotonically decreasing functions on

$\phi : \mathcal{Q} \times \mathcal{Q} \rightarrow \mathbb{R}$ . Accordingly, negative weights are allowed and a density function in explored space could be defined simply through the geodesic distance metric

$$\phi(\mathbf{q}, \tilde{s}_i(\mathcal{P}(t), t)) = -d_g(\mathbf{q}, \tilde{s}_i(\mathcal{P}(t), t)). \quad (4.6)$$

However, it should be noted that (4.6) does neither allow for a physical interpretation in terms of densities nor a probability-theoretical interpretation due to the negative values.

In this dissertation the Gaussian density function (4.5) is used. An example of this density function with geodesic distance metric  $d_g$  is shown in Figure 4.2. The nonconvex allowable environment is represented in terms of an occupancy grid map (cf. Elfes [58], Moravec [97]). Cells of the grid map are either *unexplored*, an *obstacle* on the boundary of  $\mathcal{Q}$ , or *free*. A cell on the boundary  $\partial\mathcal{Q}$  of the environment switches its state from *unexplored* to *obstacle*, as soon as the cell intersects with a circle around the robot positions  $\mathbf{p}_i$  defined by the sensing range  $r$ . A cell in the interior of  $\mathcal{Q}$  switches its state from *unexplored* to *free*, if it is fully contained in one of the sensing circles. The frontier is defined by all unexplored cells adjoining free cells. The density decreases with larger distance and is depicted by the color gradient on a logarithmic scale.



**Figure 4.2:** Density function in a nonconvex allowable environment

*Remark 4.1* (Relation to the distance transform). The computation of the weights of all points  $\mathbf{q}$  in the explored region  $\mathcal{S}_i$  relies on the distance from each  $\mathbf{q}$  to the frontier  $\tilde{s}_i$  in a Voronoi cell  $\mathcal{V}_i$ . Interestingly, this equals the distance transform in Definition 2.14 (page 14), except for the following differences: First, the distance is computed with respect to the frontier

instead of the obstacles. Second, in nonconvex allowable environments, the geodesic distance norm is required instead of the Euclidean norm. Thus, the modified distance transform can be thought of as an extension to the distance transform known in image processing (e.g., Fabbri et al. [59]).

### 4.2.2 Building the Partial Derivative

Cortés et al. [49], Pimenta et al. [104], and Lee and Egerstedt [84] consider time-varying density functions  $\phi(\mathbf{q}, t)$  for target tracking. As a consequence, the control laws depend on the dynamics of the centroid of the  $r$ -limited Voronoi cells, which in turn depends on  $\phi(\mathbf{q}, t)$ . This dependency is required, since if the dynamics of  $\phi(\mathbf{q}, t)$  is ignored, convergence and thus successful target tracking cannot be achieved.

Contrary, the density function  $\phi(\mathbf{q}, \tilde{s}_i(\mathcal{P}(t), t))$  in (4.5) does not explicitly depend on the time. Instead, it depends on the state of the frontier  $\tilde{s}_i(\mathcal{P}(t), t)$  in the Voronoi cell  $\mathcal{V}_i(\mathcal{P}(t))$ . This in turn implies that the frontier  $\tilde{s}_i$  is constant over time if the robots do not move, i.e., if  $\dot{\mathcal{P}}(t) = \mathbf{0}$ . Considering moving robots, the density in a Voronoi cell changes only in two cases: First, the robot moves into unexplored regions of the environment and consequently pushes back the frontier. Second, the partition changes such that parts of the frontier are assigned from Voronoi cell  $\mathcal{V}_i$  to  $\mathcal{V}_j$  or vice versa. With this background in mind, it is not crucial to model the dynamics of the density function in the gradient (4.3). Therefore, the following property is assumed.

**Assumption 4.1.** Building the partial derivative of  $\mathcal{H}_{\text{discover}}$  in (4.1) with respect to the robot positions  $\mathbf{p}_i$ , the density  $\phi$  can be modeled as a quasi-stationary function.

Under Assumption 4.1, one obtains the well-known gradients [50]

$$\frac{\partial \mathcal{H}_{\text{discover}}(\mathcal{P}, \mathcal{S})}{\partial \mathbf{p}_i} = k_i(\mathbf{p}_i - \mathbf{m}_\phi(\mathcal{V}_{i, \bar{r}})), \quad (4.7)$$

in convex allowable environments for each robot  $i = 1, \dots, N$ . Therein,  $k_i \in \mathbb{R}_{>0}$  denotes a positive gain, and  $\mathbf{m}_\phi(\mathcal{V}_{i, \bar{r}})$  denotes the weighted centroid of the  $\bar{r}$ -limited Voronoi cells  $\mathcal{V}_{i, \bar{r}}$ . Obviously, the negative gradient points straight from the robot position into the centroid, which reflects the desired behavior described in (4.3).

In line with the solution to the coverage problem, the gradients in (4.7) depend solely on local information available in the respective  $\bar{r}$ -limited

Voronoi cell  $\mathcal{V}_{i,\bar{r}}$ . Hence, although the objective function  $\mathcal{H}_{\text{discover}}$  is defined on the entire domain  $\mathcal{Q}$ , its optimization is spatially distributed over the Delaunay graph. This in turn implies that the communication topology relies on the Voronoi partition, and the coordination of the robots based on the gradients is distributed, meeting the design criteria in the problem statement (cf. Section 1.2, page 3).

### 4.2.3 DisCoverage in Convex Environments

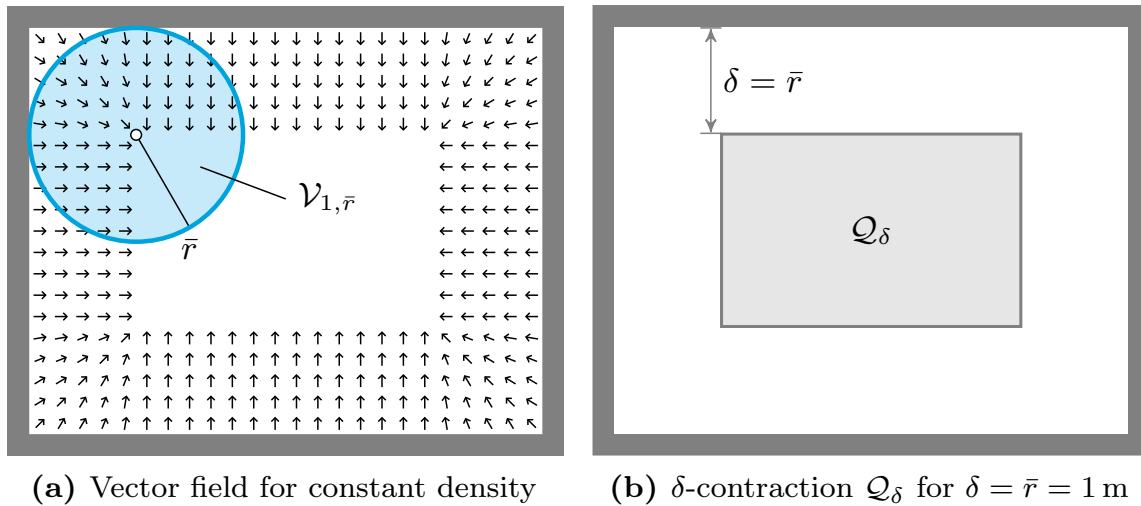
In this section, the DisCoverage multi-robot exploration approach is investigated for convex (obstacle-free) allowable environments  $\mathcal{Q}$  as published in Haumann et al. [8]. In this case, all Voronoi cells  $\mathcal{V}_i$  are convex. Upper and lower bounds for the integration range  $\bar{r}$  are given. Further, the sensing range  $r$  is defined as a function of the integration range  $\bar{r}$  and the shape of the convex polygonal environment. Finally, a proof of convergence is provided.

#### On Choosing the Integration Range

In addition to the modified density, the objective function  $\mathcal{H}_{\text{discover}}$  in (4.1) depends on the integration range  $\bar{r}$  which defines the  $\bar{r}$ -limited Voronoi cells  $\mathcal{V}_{i,\bar{r}}$  in the Voronoi cell  $\mathcal{V}_i$  (cf. Definition 2.13, page 14). The  $\bar{r}$ -limited Voronoi cells  $\mathcal{V}_{i,\bar{r}}$  form the base for computing the motion control laws (4.3) through the partial derivatives (4.7) for each robot  $i = 1, \dots, N$  (cf. (3.22), page 31). In the following, the impact of the integration range  $\bar{r}$  on the motion control laws (4.3) and the sensing range  $r$ , and therefore the entire exploration process, is examined.

A good understanding of the impact of the integration range is obtained by considering an explored, convex environment  $\mathcal{Q}$  with constant density  $\phi_{\text{const}} = 1$  as shown in Figure 4.3. In Figure 4.3(a) the vector field is shown for a single robot with an integration range of  $\bar{r} = 1$  m. The normalized gradients are computed by evaluating (4.7) inserted into (4.3) in each cell of the explored environment. Obviously, the boundary acts as repulsive force such that all trajectories lead into an invariant set in  $\mathcal{Q}$  that is defined by all points  $\mathbf{q} \in \mathcal{Q}$  whose distance to the boundary  $\partial\mathcal{Q}$  is greater than or equal to the integration range  $\bar{r}$ . This is formalized, first for a single robot, as follows.

**Theorem 4.1.** Given a single robot in  $\mathbf{p}_1 \in \mathcal{Q}$  with  $\bar{r}$ -limited Voronoi cell  $\mathcal{V}_{1,\bar{r}}$ , integration range  $\bar{r}$ , and a constant density  $\phi$ . All gradients (4.3) are



**Figure 4.3:** Fully explored, convex allowable environment  $\mathcal{Q}$  with constant density. The boundary acts as repulsive force. Legend:  $\circ$  robot position,  $\text{---}$  integration range  $\bar{r}$ , grid map cell resolution:  $0.2 \text{ m} \times 0.2 \text{ m}$ .

zero in the  $\delta$ -contraction  $\mathcal{Q}_{\delta=\bar{r}}$  of  $\mathcal{Q}$ , which defines an invariant set. All negative gradients of (4.7) in (4.3) for  $\mathbf{p}_1 \in \mathcal{Q} \setminus \mathcal{Q}_\delta$  point towards  $\mathcal{Q}_\delta$ .

*Proof.* If  $\mathbf{p}_1 \in \mathcal{Q}_\delta$ , the  $\bar{r}$ -limited Voronoi cell  $\mathcal{V}_{1,\bar{r}}$  is radially unbounded. In this case, the centroid  $\mathbf{m}_\phi(\mathcal{V}_{1,\bar{r}})$  lies in  $\mathbf{p}_1$ , and the gradients in (4.7) vanish. If  $\mathbf{p}_1 \in \mathcal{Q} \setminus \mathcal{Q}_\delta$ , the  $\bar{r}$ -limited Voronoi cell  $\mathcal{V}_{1,\bar{r}}$  intersects with the boundary. In this case, the centroid  $\mathbf{m}_\phi(\mathcal{V}_{1,\bar{r}})$  does not equal  $\mathbf{p}_1$ . Instead, it is pushed away from the boundary towards  $\mathcal{Q}_\delta$ . Hence, the negative gradients (4.7) used in (4.3) point straight from  $\mathbf{p}_1$  into the centroid, and the trajectories starting in  $\mathcal{Q} \setminus \mathcal{Q}_\delta$  approach  $\mathcal{Q}_\delta$ .  $\square$

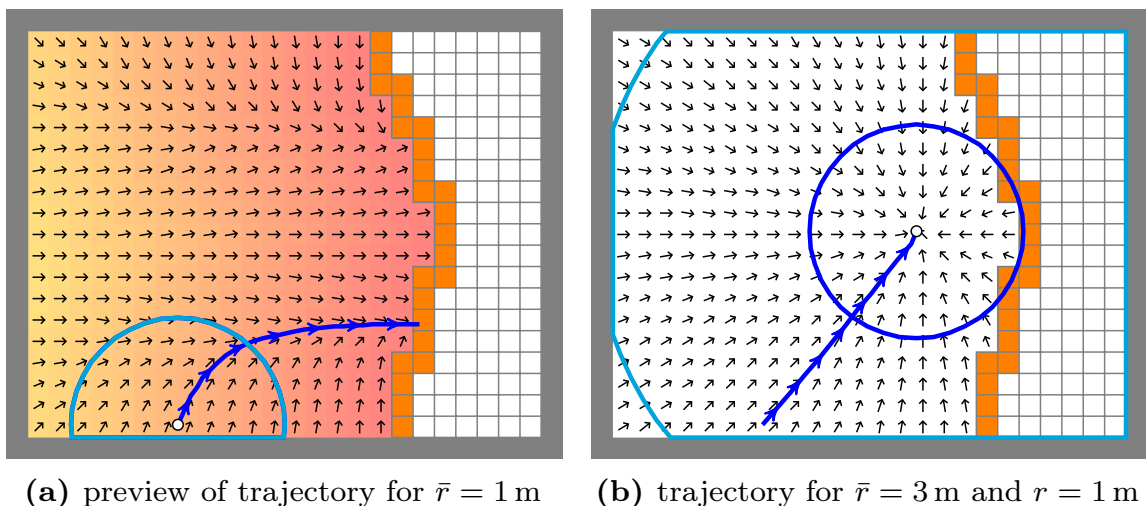
Theorem 4.1 is an inherent property of the limited centroidal search [36]. The  $\delta$ -contraction of  $\mathcal{Q}$  is also known as *growing of obstacles* in robotics for collision-free path planning (cf. Lozano-Pérez and Wesley [87], Udupa et al. [130]). The invariant set, where all gradients are zero, is equal to  $\mathcal{Q}_\delta$  for  $\delta = \bar{r}$  as shown in Figure 4.3(b). Consequently, the integration range can be thought of as a safety distance to the boundary. This further allows for a physical interpretation: When using real robots with dynamics  $\dot{\mathbf{p}}_i = \mathbf{u}_i$  as in (4.3), the integration range  $\bar{r}$  must satisfy the lower bound

$$\bar{r}_{\min} = \frac{1}{2} \text{diam}_{\text{robot}}. \quad (4.8)$$

Contrary, if (4.8) is violated, the risk of colliding with the environment rapidly increases with decreasing integration range.



Unfortunately, (4.8) does not strictly hold for non-uniform densities such as (4.5), since higher densities may shift the centroid closer to the boundary depending on the slope of the density, such that the distance from the centroid to the boundary is significantly less than the integration range  $\bar{r}$ . Thus, applying Theorem 4.1 for non-uniform densities, (4.8) is only a necessary but not a sufficient condition to avoid collisions. Nevertheless, the  $\delta$ -contraction can still be regarded as an approximate safety distance. This is depicted in Figure 4.4(a) for a single robot in a partially explored convex environment  $\mathcal{Q}$ . In line with Figure 4.2, the density is illustrated through the color gradient. Despite the non-uniform density, the trajectory of the robot still adheres roughly to the safety distance of  $\bar{r} = 1$  m, finally approaching unexplored space.



**Figure 4.4:** Vector field for a rectangular environment with a single robot. Legend:  $\circ$  robot position,  $-$  integration range  $\bar{r}$ ,  $-$  sensing range  $r$ ,  $\rightarrow$  trajectory. Grid map cell resolution:  $0.2 \text{ m} \times 0.2 \text{ m}$ .

Next to the lower bound  $\bar{r}_{\min}$  of the integration range, the question arises whether an upper bound  $\bar{r}_{\max}$  exists. Indeed, closely investigating Figure 4.4(b) reveals that too large values for  $\bar{r}$  result in trajectories that all lead into a single time-invariant equilibrium point. In fact, for  $\bar{r} \rightarrow \text{diam } \mathcal{Q}$ , the limited centroidal search equals the unlimited centroidal search [50] as discussed in Section 3.1.6 (page 28), which is known to maximize coverage for the entire convex domain and therefore not suited for exploration. This also gets clear by recognizing that the sensing range of  $r = 1$  m is not sufficient to explore new parts of the environment in Figure 4.4(b). From Figure 4.3 and Theorem 4.1, it can be concluded that the  $\delta$ -contraction of  $\mathcal{Q}$  can be interpreted in terms of a defensive approximation of the *reachability*

set for  $\delta = \bar{r}$  (cf. Definition 2.8, page 11). The reachability set  $\mathcal{Q}_\delta$  contains all points that can safely be reached while strictly maintaining the safety distance  $\delta$  to the boundary  $\partial\mathcal{Q}$ . Increasing values for the integration range  $\bar{r}$  push the robots further away from the boundary  $\partial\mathcal{Q}$ . Finally, a value  $\delta_{\max}$  exists such that the reachability set  $\mathcal{Q}_\delta$  degenerates into a single line or a point. Considering Figure 4.4(b), the reachability set is defined by a horizontal line when setting  $\delta_{\max}$  to half of the height of the convex environment. The  $\delta$ -contraction of  $\mathcal{Q}$  for  $\delta > \delta_{\max}$  yields an empty set  $\mathcal{Q}_\delta = \emptyset$ . Therefore, if a robot should be able to navigate in the environment  $\mathcal{Q}$  with control laws (4.3), the condition

$$\bar{r} \leq \bar{r}_{\max} = \delta_{\max} \quad (4.9)$$

must hold. Interestingly,  $\mathcal{Q}_\delta$  for  $\delta = \delta_{\max}$  equals the paths defined by the generalized Voronoi partition [81].

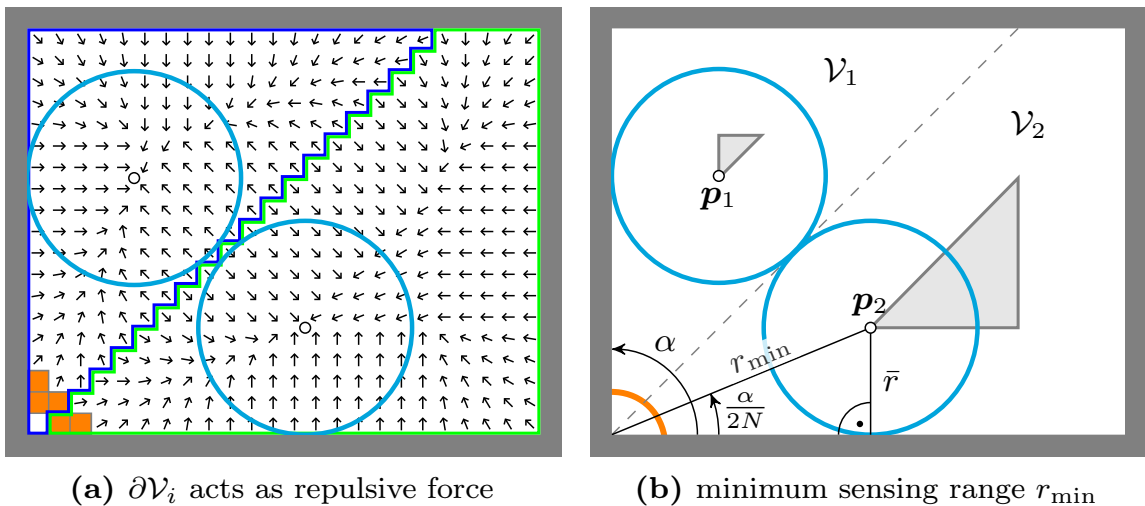
It is worth mentioning that choosing larger values for the integration range  $\bar{r}$  and, thus, violating (4.9) does not have fatal consequences: In this case, the unlimited centroidal search results in a centroidal Voronoi configuration and the robots remain in this configuration forever. Therefore, the upper bound does neither impose a necessary nor a sufficient condition to explore the entire convex environment. Instead, the sensing range must simply be large enough in order to sense the entire environment from the corresponding configuration. How to choose a sufficiently large sensing range  $r$  as a function of the integration range  $\bar{r}$  is discussed next.

### On Choosing the Sensing Range

In order to find a lower bound  $r_{\min}$  for the sensing range  $r$ , it is necessary to observe that the boundary  $\partial\mathcal{V}_i$  of a Voronoi cell  $\mathcal{V}_i$  acts as a repulsive force exactly like the boundary  $\partial\mathcal{Q}$  of the environment (see Figure 4.5(a)). As a consequence, the minimum sensing range depends on the Voronoi partition, the number of robots and the smallest interior angle  $\alpha$  of the polygonal environment.

**Theorem 4.2** (Lower bound for the sensing range  $r$ ). Given  $N$  robots with integration range  $\bar{r} > \bar{r}_{\min}$ . Denote with  $\alpha$  the smallest interior angle of the convex polygonal environment  $\mathcal{Q}$ . The minimum sensing range  $r_{\min}$  required to explore  $\mathcal{Q}$  is defined by

$$r_{\min} = \frac{\bar{r}}{\sin(\frac{\alpha}{2N})}. \quad (4.10)$$



**Figure 4.5:** Sensing range  $r_{\min}$  in dependence of  $N$  and the interior angle  $\alpha$

*Proof.* Since the boundary of the Voronoi cell acts as repulsive force, it is sufficient to consider the Voronoi cell  $\mathcal{V}_i$  of a single robot. Therein, the reachability set is defined by the  $\delta$ -contraction of the Voronoi cell  $\mathcal{V}_i$  (cf. Figure 4.5(b)). The interior angle  $\frac{\alpha}{2N}$  in the Voronoi cell along with the minimum sensing range  $r_{\min}$  and the integration range  $\bar{r}$  define a right triangle. Using the sine function, one obtains  $r_{\min}$  as hypotenuse.  $\square$

*Remark 4.2.* From (4.10)  $r_{\min} \geq \bar{r}$  follows since  $\sin(\frac{\alpha}{2N}) \in (0, 1]$  for the smallest interior angle  $\alpha \in (0, \pi]$ .

According to condition (4.10) in Theorem 4.2, the sensing range  $r$  must satisfy  $r \geq r_{\min}$  to allow for complete exploration of the environment. Based on the discussion for the sensing range  $r$  and the integration range  $\bar{r}$ , these results are summarized as follows.

**Corollary 4.1** (Choice of integration and sensing range). Let  $\mathcal{Q}$  be a convex allowable environment. Then, the entire environment can be explored over time if the integration range  $\bar{r}$  and the sensing range  $r$  satisfy (4.8) according to Theorem 4.1, and (4.10) of Theorem 4.2, respectively.

### The Separatrix: Unstable Invariant Sets

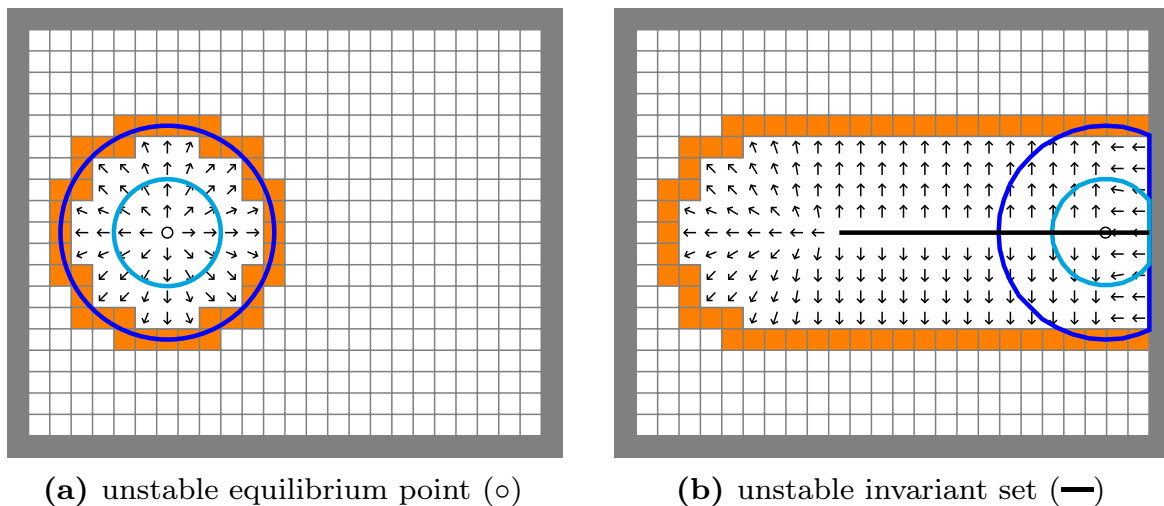
The density function  $\phi(\mathbf{q}, \tilde{s}_i)$  in (4.5) assigns decreasing real values to each  $\mathbf{q}$  with increasing distance to the frontier. However, considering that the frontier may have arbitrarily curved shapes during the exploration, the non-uniform weights in the explored parts  $\mathcal{S}$  of the environment  $\mathcal{Q}$

introduce multiple maxima and minima, where

$$\frac{\partial \mathcal{H}_{\text{discover}}(\mathcal{P}, \mathcal{S})}{\partial \mathbf{p}_i} = \mathbf{0} \quad (4.11)$$

holds. As frequently discussed in the solution to the coverage problem [36], the objective function  $\mathcal{H}$  exhibits its minima for centroidal Voronoi configurations  $\mathcal{P}$ , maximizing coverage. However, (4.11) imposes only a necessary condition for an extremum, implying that a configuration  $\mathcal{P}$  may represent a maximum or a saddle point. A maximum is the worst case for a centroidal Voronoi configuration  $\mathcal{P}$ , since the gradients are zero. Consequently, the robots remain at their position forever without moving to unexplored regions.

In fact, these worst-case situations appear frequently in the proposed approach to multi-robot exploration, as Figure 4.6 shows. In Figure 4.6(a),



**Figure 4.6:** Unstable equilibrium points and invariant sets. Legend:  $\circ$  robot position,  $\text{—}$  integration range  $\bar{r} = 0.5$  m,  $\text{—}$  sensing range  $r = 1$  m.

the vector field is shown for a single robot in an unknown environment right after mapping the surrounding area for the first time. Obviously, the gradient in the robot position is zero, and all other gradients in the explored region point away from the robot to the frontier. Hence, the robot is positioned in a maximum of the objective function  $\mathcal{H}_{\text{discover}}$ , which is equal to an unstable equilibrium point from a feedback control perspective. Theoretically, the robot stays in this position forever, and a random infinitesimal small perturbation needs to be added to the gradient to continue exploration. Then, in Figure 4.6(b), the robot continued the exploration to the east, finally arriving at the boundary  $\partial Q$  of the

environment. This time, the vector field reveals an invariant set in the shape of a line instead of a single equilibrium point. Analog to the circular case, all other gradients point towards the frontier, therefore the invariant set is unstable.

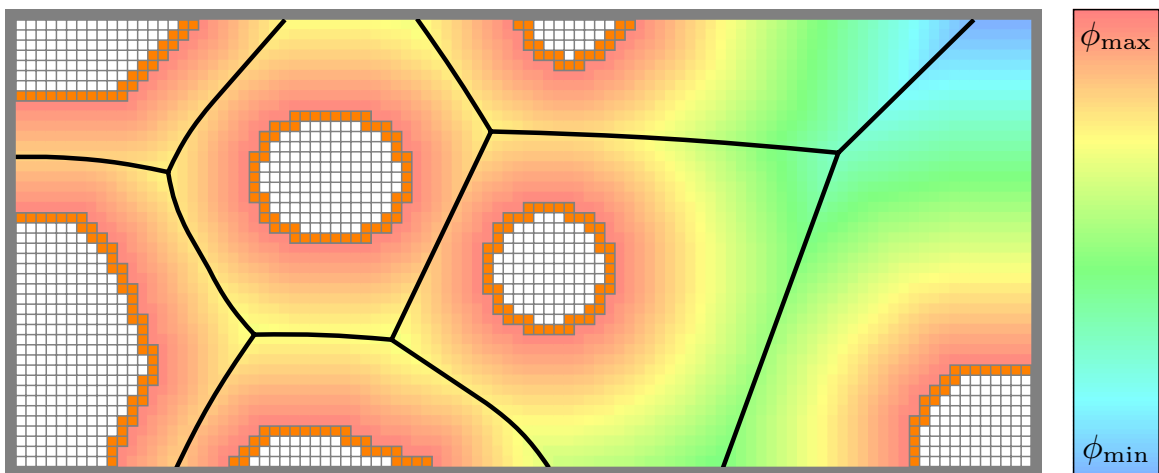
From a mathematical point of view, these unstable invariant sets can be detected by checking the sufficient condition

$$\mathbf{H}_{\mathcal{H}_{\text{discover}}}(\mathcal{P}, \mathcal{S}) = \left[ \frac{\partial^2 \mathcal{H}_{\text{discover}}(\mathcal{P}, \mathcal{S})}{\partial p_{ij} \partial p_{ik}} \right]_{j,k=1,\dots,d} \prec 0 \quad (4.12)$$

in the  $d$ -dimensional space  $\mathbb{R}^d$ , i.e., the Hessian is required to be negative definite ( $\prec 0$ ) in  $\mathbf{p}_i(t)$ . However, it is much easier to detect this case according to the following observation.

**Corollary 4.2** (Leaving unstable invariant sets). If condition (4.11) holds for any robot  $i$ , i.e., the gradient of  $\mathcal{H}_{\text{discover}}$  of robot  $i$  is  $\mathbf{0}$ , and if there still exists a frontier  $\tilde{s}_i$  in the Voronoi cell  $\mathcal{V}_i$  of the robot, then the robot is located in the unstable invariant set and an arbitrary perturbation must be added to the gradient for the robot to leave the invariant set and continue exploration.

Before further analyzing convergence properties of DisCoverage, it is worth taking a closer look at Figure 4.7. Therein, all points in the unstable



**Figure 4.7:** Separatrix of the vector field, keeping maximum distance to the frontier.

invariant set form the separatrix, highlighted by the black line in the explored regions. Interestingly, the separatrix appears to be defined in line with the generalized Voronoi partition (cf. Definition 2.15, page 15), except

for the fact that the distance is maximized with respect to the frontier instead of to the boundary  $\partial\mathcal{Q}$ . Thereafter, the separatrix equals the set defined by the maximum clearance method in robot navigation [82].

### Stability Analysis and Proof of Convergence

According to the problem formulation (cf. Section 1.2, page 3), the exploration process is complete if  $\mathcal{S}(t) \rightarrow \mathcal{Q}$  as  $t$  approaches  $\infty$ . Since the density function is constructed such that it is maximal on the frontier and monotonically decreases with increasing distance to the frontier, the following theorem proves that the limited centroidal search with integration range  $\bar{r}$ , denoted by the objective function (4.1) together with the gradient-based motion control laws (4.3), moves the robots into regions with maximum density, approaching a  $\bar{r}$ -limited centroidal Voronoi configuration.

**Theorem 4.3** (Frontier-based centroidal search). Let  $\mathcal{Q}$  denote a convex allowable environment and let  $\mathcal{S}$  denote the explored region in  $\mathcal{Q}$ . Let  $\mathcal{P} = \{\mathbf{p}_1, \dots, \mathbf{p}_N\}$  be the configuration of  $N$  robots in  $\mathcal{Q}$  with integration range  $\bar{r}$ , and denote with  $\mathcal{V} = \{\mathcal{V}_1, \dots, \mathcal{V}_N\}$  the Voronoi partition of  $\mathcal{Q}$  for  $\mathcal{P}$ . Then, applying (4.1) and (4.3), the robots approach a  $\bar{r}$ -limited centroidal Voronoi partition as  $t \rightarrow \infty$ .

*Proof.* Essentially, the proof follows the one of the limited centroidal search in Section 3.1.7 (page 32): Building the time-derivative of (4.1) and inserting (4.3) yields

$$\begin{aligned} \dot{\mathcal{H}}_{\text{discover}}(\mathcal{P}, \mathcal{S}) &= \frac{\partial \mathcal{H}_{\text{discover}}(\mathcal{P}, \mathcal{S})}{\partial \mathcal{P}} \dot{\mathcal{P}} \\ &= - \sum_{i=1}^N \left\| \frac{\partial \mathcal{H}_{\text{discover}}(\mathcal{P}, \mathcal{S})}{\partial \mathbf{p}_i} \right\|^2. \end{aligned} \quad (4.13)$$

Under Assumption 4.1, the partial derivatives of  $\mathcal{H}_{\text{discover}}$  with respect to the robot positions  $\mathbf{p}_i$  are given by (4.7). Inserting (4.7) into (4.13) is equivalent to

$$\dot{\mathcal{H}}_{\text{discover}}(\mathcal{P}, \mathcal{S}) = -k_p \sum_{i=1}^N \|\mathbf{p}_i - \mathbf{m}_\phi(\mathcal{V}_{i, \bar{r}})\|^2 \leq 0. \quad (4.14)$$

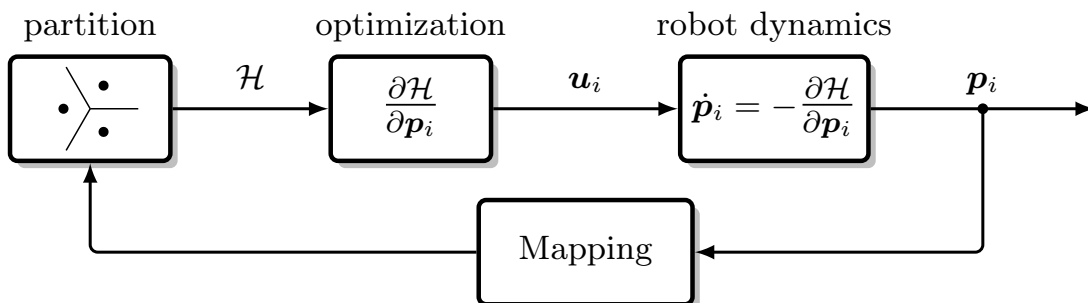
Obviously, the sum of squares as well as  $k_p$  in (4.14) are non-negative, implying that  $\mathcal{H}_{\text{discover}}$  decreases over time. Applying the Krasovskii-LaSalle invariance principle, the robots move to the largest invariant set, which equals the set of all  $\bar{r}$ -limited centroidal Voronoi configurations.  $\square$

With the help of Corollary 4.1 and 4.2, and Theorem 4.3, a proof of convergence for the DisCoverage-based exploration approach is given next.

**Theorem 4.4** (Proof of convergence). Let  $\mathcal{Q}$  denote a convex allowable environment and let  $\mathcal{S}(t)$  denote the explored region in  $\mathcal{Q}$ . Let  $\mathcal{P} = \{\mathbf{p}_1, \dots, \mathbf{p}_N\}$  be the configuration of  $N$  robots in  $\mathcal{Q}$  with integration range  $\bar{r}$ , and denote with  $\mathcal{V} = \{\mathcal{V}_1, \dots, \mathcal{V}_N\}$  the Voronoi partition of  $\mathcal{Q}$  for  $\mathcal{P}$ . Then, if  $\bar{r} \geq \bar{r}_{\min}$  and  $r \geq r_{\min} \geq \bar{r}$  holds according to Corollary 4.1, and applying Corollary 4.2, continuous optimization of (4.1) and the motion control laws (4.3) solve the multi-robot exploration problem and  $\mathcal{S}(t) \rightarrow \mathcal{Q}$  as  $t \rightarrow \infty$ .

*Proof.* Applying Corollaries 4.1 and 4.2, the lower and upper bound for the integration range  $\bar{r}$  results in trajectories that approach the frontier, converging to an  $\bar{r}$ -limited centroidal Voronoi configuration as stated by Theorem 4.3. Since for the sensing range  $r \geq r_{\min}$  holds, the frontier is within sensing range  $r$  and therefore pushed back, which in turn changes the density function  $\phi(\mathbf{q}, \tilde{s}_i)$ . Therewith, the  $\bar{r}$ -limited centroidal Voronoi configuration is never reached and the exploration continues until the entire environment is explored, in which case  $\mathcal{S}(t) = \mathcal{Q}$  holds.  $\square$

In line with the solution to the coverage problem by Cortés et al. [50], the optimization of the DisCoverage approach follows exactly the closed feedback loop in Figure 4.8 with  $\mathcal{H} = \mathcal{H}_{\text{discover}}$  and the addition of the *mapping* block. The partition is defined by the Voronoi partition  $\mathcal{V}$ , the optimization by the frontier-dependent density function (4.5) and the objective function (4.1), and the robot dynamics by the motion control laws (4.3) and (4.7). The mapping block denotes that each robot continuously maps unknown parts of the environment in its sensing range.



**Figure 4.8:** DisCoverage continuous-time feedback loop (cf. Haumann et al. [8])

#### 4.2.4 DisCoverage in Nonconvex Environments

In this section, the DisCoverage approach is extended to support nonconvex allowable environments  $\mathcal{Q} \subset \mathbb{R}^2$ , which are defined by a polygonal environment with polygonal obstacles. Consequently, the partition  $\mathcal{V} = \{\mathcal{V}_1, \dots, \mathcal{V}_N\}$  now is defined by *nonconvex geodesic* Voronoi cells  $\mathcal{V}_i$ . The nonconvexity of the geodesic Voronoi cells  $\mathcal{V}_i$  introduces corner cases especially with respect to the separatrix. Therefore, preliminary considerations are carried out next in order to shed light into the necessary steps to transfer the centroidal search-based DisCoverage approach to nonconvex environments. The results presented in this section were preliminarily published in Haumann et al. [9].

##### Preliminary Considerations: Applying the Centroidal Search to the $\bar{r}$ -limited Visibility Sets

Due to the nonconvexity of the geodesic Voronoi cells  $\mathcal{V}_i$ , the definition (4.2) of the  $\bar{r}$ -limited Voronoi cells  $\mathcal{V}_{i,\bar{r}}$  is not applicable anymore. Instead, the  $\bar{r}$ -limited Voronoi cells  $\mathcal{V}_{i,\bar{r}}$  are substituted by the  $\bar{r}$ -limited visibility sets

$$\mathcal{V}_{i,\bar{r}}^* = \{\mathbf{q} \in \mathcal{V}_i \mid \|\mathbf{p}_i - \mathbf{q}\| \leq \bar{r} \text{ and } [\mathbf{p}_i, \mathbf{q}] \subset \mathcal{Q}\}, \quad (4.15)$$

$i = 1, \dots, N$ . The superscript  $\star$  in (4.15) indicates that  $\mathcal{V}_{i,\bar{r}}^*$  define star-shaped regions with respect to the robot positions  $\mathbf{p}_i$ . As discussed in Section 3.1.9 (page 34), Marier et al. [92] have shown that the gradients in visibility sets introduce normal vectors that significantly increase the computational burden when building the gradients. This effect is undesired, especially since the contribution of the normal vectors on the boundary to the resulting value of the objective function may be marginal. Therefore, the following simplification is introduced:

**Assumption 4.2** (Quasi-stationary visibility sets). When building the partial derivatives of the objective function with respect to the robot positions  $\mathbf{p}_i$ , the star-shaped visibility sets  $\mathcal{V}_{i,\bar{r}}^*$  are assumed to be quasi-stationary and therefore can be modeled as constant.

Note, that at first glance, Assumption 4.2 may not appear feasible. However, it will be justified in Remark 4.8 later in this section (page 71). Turning the  $r$ -limited Voronoi cells  $\mathcal{V}_{i,r}$  into the star-shaped visibility sets  $\mathcal{V}_{i,\bar{r}}^*$  according to (4.15), the preliminary optimization problem under



Assumption 4.2 is given by

$$\mathcal{H}_{\text{discover}}(\mathcal{P}, \mathcal{S}) = \sum_{i=1}^N \int_{\mathcal{V}_{i,\bar{r}}^*} \|\mathbf{q} - \mathbf{p}_i\|^2 \phi(\mathbf{q}, \tilde{s}_i) d\mathbf{q} \longrightarrow \min! \quad (4.16)$$

Note, that the objective function  $\mathcal{H}_{\text{lim}}$  in (3.21) of the  $r$ -limited centroidal search (page 31) includes an additional integral over the set  $\mathcal{V}_i \setminus \mathcal{V}_{i,r}$ . This part vanishes in the partial derivative with respect to the robot positions  $\mathbf{p}_i$ . Contrary, due to Assumption 4.2 the  $\bar{r}$ -limited visibility sets  $\mathcal{V}_{i,\bar{r}}^*$  are constant. Therefore, the additional term is not required, since the partial derivatives of (4.16) with respect to  $\mathbf{p}_i$  simplify to

$$\begin{aligned} \frac{\partial \mathcal{H}_{\text{discover}}(\mathcal{P}, \mathcal{S})}{\partial \mathbf{p}_i} &= \frac{\partial}{\partial \mathbf{p}_i} \sum_{j=1}^N \int_{\mathcal{V}_{j,\bar{r}}^*} \|\mathbf{q} - \mathbf{p}_j\|^2 \phi(\mathbf{q}, \tilde{s}_j) d\mathbf{q} \\ &= \int_{\mathcal{V}_{i,\bar{r}}^*} \frac{\partial}{\partial \mathbf{p}_i} \|\mathbf{q} - \mathbf{p}_i\|^2 \phi(\mathbf{q}, \tilde{s}_i) d\mathbf{q} \\ &= k_i(\mathbf{p}_i - \mathbf{m}_\phi(\mathcal{V}_{i,\bar{r}}^*)). \end{aligned} \quad (4.17)$$

Therein,  $\mathbf{m}_\phi(\mathcal{V}_{i,\bar{r}}^*)$  defines the centroid of  $\mathcal{V}_{i,\bar{r}}^*$ . Further, due to Assumption 4.2, the extended Leibniz integral rule for differentiation under the integral sign is not required anymore. In fact, Assumption 4.2 allows for another simplification: Since the  $\bar{r}$ -limited visibility sets are quasi-stationary with respect to robot movements, the optimization problem (4.16) can be decomposed into  $N$  independent optimization problems

$$\begin{aligned} \mathcal{H}_{\text{discover}}(\mathcal{P}, \mathcal{S}) &= \sum_{i=1}^N \mathcal{H}_{\text{discover},i}(\mathbf{p}_i, \mathcal{S}_i) \\ &= \sum_{i=1}^N \int_{\mathcal{V}_{i,\bar{r}}^*} \|\mathbf{q} - \mathbf{p}_i\|^2 \phi(\mathbf{q}, \tilde{s}_i) d\mathbf{q} \longrightarrow \min! \end{aligned} \quad (4.18)$$

Building the partial derivative of (4.18) with respect to robot position  $\mathbf{p}_i$  yields

$$\frac{\partial \mathcal{H}_{\text{discover}}(\mathcal{P}, \mathcal{S})}{\partial \mathbf{p}_i} = \frac{\partial \mathcal{H}_{\text{discover},i}(\mathbf{p}_i, \mathcal{S}_i)}{\partial \mathbf{p}_i} = k_i(\mathbf{p}_i - \mathbf{m}_\phi(\mathcal{V}_{i,\bar{r}}^*)). \quad (4.19)$$

Using (4.19), the motion control laws read

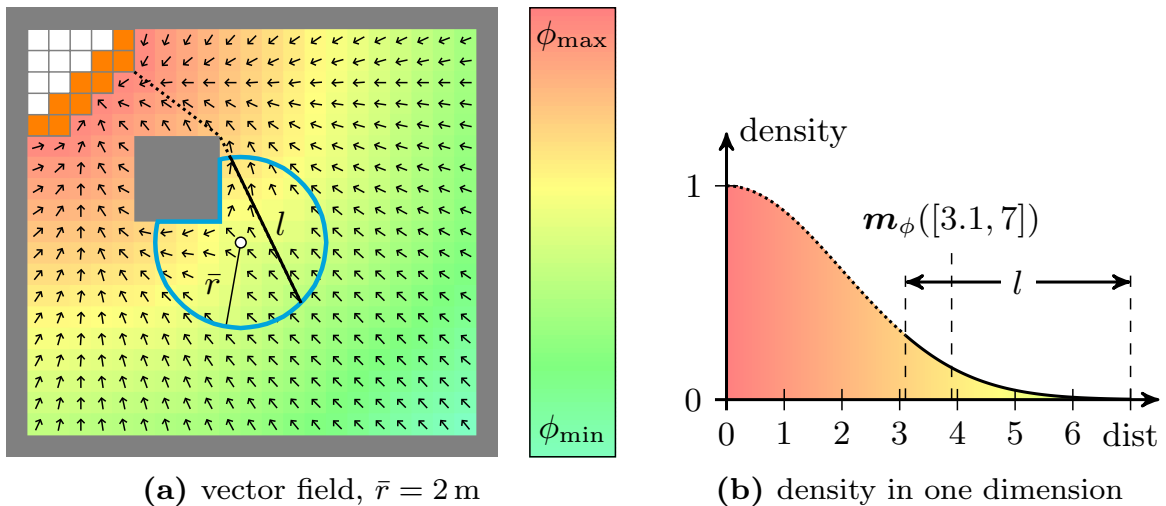
$$\dot{\mathbf{p}}_i = \mathbf{u}_i = - \frac{\partial \mathcal{H}_{\text{discover},i}(\mathbf{p}_i, \mathcal{S}_i)}{\partial \mathbf{p}_i}, \quad i = 1, \dots, N. \quad (4.20)$$

In contrast to (4.1), (4.3), and (4.7), equations (4.18)–(4.20) apply the  $\bar{r}$ -limited visibility sets  $\mathcal{V}_{i,\bar{r}}^*$  in (4.15) instead of the  $\bar{r}$ -limited Voronoi cells  $\mathcal{V}_{i,\bar{r}}$  in (4.2). Based on these equations, the separatrix is considered next.

In convex environments, the centroid of the  $\bar{r}$ -limited Voronoi cell  $\mathcal{V}_{i,\bar{r}}$  is always contained in  $\mathcal{V}_{i,\bar{r}}$ . As a consequence, the separatrix builds an invariant subset of the allowable environment  $\mathcal{Q}$ , meaning that no trajectory on the separatrix ever leaves  $\mathcal{Q}$ . In order to leave this invariant set, a perturbation needs to be added to the motion control laws (cf. Corollary 4.2, page 57).

Contrary, in nonconvex allowable environments the centroid of the star-shaped  $\bar{r}$ -limited visibility sets  $\mathcal{V}_{i,\bar{r}}^*$  are not necessarily contained in  $\mathcal{Q}$ . Therefore, the separatrix does not strictly define an invariant set in  $\mathcal{Q}$ . Instead, trajectories on the separatrix may point straight into *concave* vertices (cf. Definition 2.4, page 9) on the polygonal boundary  $\partial\mathcal{Q}$ . Such a situation is depicted in Figure 4.9 for a nonconvex environment with one obstacle, represented as a grid map with a cell resolution of  $0.5\text{ m} \times 0.5\text{ m}$ . The integration range is set to  $\bar{r} = 2\text{ m}$  and the standard deviation of the density function is  $\sigma = 2\text{ m}$ . In Figure 4.9(a) the vector field is computed through the negative gradient (4.19) in the  $\bar{r}$ -limited visibility set (4.15). All trajectories approach the frontier, except the trajectory leading into the lower right corner of the obstacle. The reason for this behavior is the choice of the non-uniform density function, as depicted in Figure 4.9(b). Therein, the density on the path, highlighted by the dotted and solid line in Figure 4.9(a), is displayed exemplarily along the horizontal axis in terms of the distance to the frontier. Further, the solid line  $l$  represents the 1-dimensional line that is contained within the  $\bar{r}$ -limited visibility set  $\mathcal{V}_{1,\bar{r}}^*$ . Computing the weighted centroid  $m_\phi = \mathbf{m}_\phi$  in the interval  $[3.1, 7]$  yields a value of  $m_\phi([3.1, 7]) \approx 3.91$ . Due to the non-uniform density, this value is close to the lower bound of the interval  $[3.1, 7]$ .

Transferring the 1-dimensional example to the 2-dimensional  $\bar{r}$ -limited visibility set  $\mathcal{V}_{1,\bar{r}}^*$ , the non-uniform density in Figure 4.9(a) pulls the center of mass  $\mathbf{m}_\phi(\mathcal{V}_{1,\bar{r}}^*)$  close to the regions with high density. Although this is the desired behavior in terms of the centroidal search, concave vertices on the boundary  $\partial\mathcal{Q}$  expose this critical problem and the centroid may lie outside  $\mathcal{Q}$ , which leads to collisions of robots with the environment. Therefore, equations (4.18)–(4.20) based on  $\mathcal{V}_{i,\bar{r}}^*$  in (4.15) are not directly applicable to apply the centroidal search to nonconvex environments. Instead, a solution needs to be found such that the trajectories along the separatrix never leave the allowable environment  $\mathcal{Q}$ , analog to the convex case. Or,



**Figure 4.9:** Vector field in nonconvex allowable environments

formulated from a feedback control perspective, all trajectories along the separatrix must approach an equilibrium point that is contained in  $\mathcal{Q}$ .

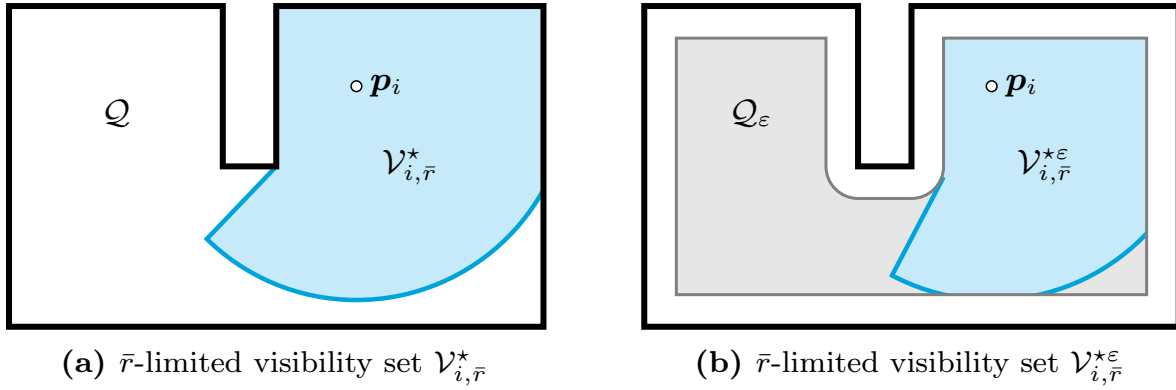
*Remark 4.3* (Projecting the centroid). As depicted in Figure 4.9, the centroid of a nonconvex set may lie outside of  $\mathcal{Q}$ , resulting in trajectories that collide with the boundary  $\partial\mathcal{Q}$  of the environment. This effect appears independent of Assumption 4.2, and is also observed by Zhong and Cassandras [138], Breitenmoser et al. [34], Lu et al. [88], Marier et al. [92], and Bhattacharya et al. [27]. The common solution proposed in these publications is to project the centroid to a valid location inside the environment  $\mathcal{Q}$  that is visible from the respective robot position  $\mathbf{p}_i$ . Thus, collisions are avoided when moving to the centroid.

### The Solution: Introducing the $\bar{r}$ -limited Visibility Sets $\mathcal{V}_{i,\bar{r}}^{\star\varepsilon}$ in $\mathcal{Q}_\varepsilon$

Instead of using a projection method as mentioned in Remark 4.3, the  $\delta$ -contraction can be applied to the nonconvex allowable environment  $\mathcal{Q}$  for an arbitrarily small  $\delta = \varepsilon > 0$  *prior to* computing the  $\bar{r}$ -limited visibility sets  $V_{i,\bar{r}}^\star$ . The  $\bar{r}$ -limited visibility sets are then denoted by

$$\mathcal{V}_{i,\bar{r}}^{\star\varepsilon} = \{\mathbf{q} \in \mathcal{V}_i \mid \|\mathbf{p}_i - \mathbf{q}\| \leq \bar{r} \text{ and } [\mathbf{p}_i, \mathbf{q}] \subset \mathcal{Q}_\varepsilon\}, \quad (4.21)$$

where  $\mathcal{Q}_\varepsilon$  is referred to as the  $\varepsilon$ -contraction of  $\mathcal{Q}$  (cf. Definition 2.7, page 10). The difference of  $\mathcal{V}_{i,\bar{r}}^\star$  and  $\mathcal{V}_{i,\bar{r}}^{\star\varepsilon}$  is depicted in Figure 4.10. As a consequence, all concave vertices on the boundary of  $\mathcal{Q}$  are turned into concave circular segments in  $\mathcal{V}_{i,\bar{r}}^{\star\varepsilon}$  in the  $\varepsilon$ -contraction of  $\mathcal{Q}$  (cf. Corollary 2.1, page 11). In



**Figure 4.10:** Visibility sets without and with  $\varepsilon$ -contraction of  $\mathcal{Q}$

summary, under Assumption 4.1 and Assumption 4.2<sup>1)</sup> the optimization problem reads

$$\begin{aligned} \mathcal{H}_{\text{discover}}^*(\mathcal{P}, \mathcal{S}) &= \sum_{i=1}^N \mathcal{H}_{\text{discover},i}^*(\mathbf{p}_i, \mathcal{S}_i) \\ &= \sum_{i=1}^N \int_{\mathcal{V}_{i,\bar{r}}^{*\varepsilon}} \|\mathbf{q} - \mathbf{p}_i\|^2 \phi(\mathbf{q}, \tilde{s}_i) d\mathbf{q} \rightarrow \min! \end{aligned} \quad (4.22)$$

with partial derivatives

$$\frac{\partial \mathcal{H}_{\text{discover},i}^*(\mathbf{p}_i, \mathcal{S}_i)}{\partial \mathbf{p}_i} = k_i (\mathbf{p}_i - \mathbf{m}_\phi(\mathcal{V}_{i,\bar{r}}^{*\varepsilon})) \quad (4.23)$$

and motion control laws

$$\dot{\mathbf{p}}_i = \mathbf{u}_i = - \frac{\partial \mathcal{H}_{\text{discover},i}^*(\mathbf{p}_i, \mathcal{S}_i)}{\partial \mathbf{p}_i}, \quad i = 1, \dots, N. \quad (4.24)$$

Using the  $\bar{r}$ -limited visibility sets  $\mathcal{V}_{i,\bar{r}}^{*\varepsilon}$ , one can state the following theorem.

**Theorem 4.5** (Progression of trajectories in nonconvex environments). For a nonconvex allowable environment  $\mathcal{Q}$  and an arbitrarily small  $\varepsilon > 0$ , continuous application of (4.22)–(4.24) within the  $\bar{r}$ -limited visibility sets  $\mathcal{V}_{i,\bar{r}}^{*\varepsilon}$  (4.21) results in trajectories that never leave the invariant set  $\mathcal{Q}_\varepsilon$ .

*Proof.* Applying the  $\varepsilon$ -contraction with an arbitrarily small  $\varepsilon > 0$  to  $\mathcal{Q}$  turns all concave vertices in  $\partial\mathcal{Q}$  into continuously differentiable circular

<sup>1)</sup>Due to Assumption 4.2, the objective function  $\mathcal{H}_{\text{discover}}$  is marked with a star '★'.

path segments in  $\partial\mathcal{Q}_\varepsilon$ . For concave locations  $\mathbf{q}$  on the boundary  $\partial\mathcal{Q}_\varepsilon$ , denote with  $\text{tang}(\mathbf{q})$  the tangent in the continuously differentiable location  $\mathbf{q}$ . Approaching concave locations  $\mathbf{q}$  on the boundary  $\partial\mathcal{Q}_\varepsilon$ , the centroid of a tangentially  $\bar{r}$ -limited visibility set  $\mathcal{V}_{i,\bar{r}}^{\star\varepsilon}$  never lies behind the half-plane defined by  $\text{tang}(\mathbf{q})$ . Hence, no trajectory exists that leaves  $\mathcal{Q}_\varepsilon$ . Consequently, continuously applying (4.22)–(4.24), all equilibrium points are contained in  $\mathcal{Q}_\varepsilon$ .  $\square$

From Theorem 4.5 two observations immediately follow.

**Corollary 4.3** (Separatrix in nonconvex environments). Given a nonconvex allowable environment  $\mathcal{Q}$  and its  $\varepsilon$ -contraction  $\mathcal{Q}_\varepsilon$  for an arbitrarily small  $\varepsilon > 0$ . Following Theorem 4.5, all trajectories starting on the separatrix approach an equilibrium point that is contained in the invariant set  $\mathcal{Q}_\varepsilon$ .

Further, analog to Corollary 4.2 (page 57), leaving unstable invariant sets on the separatrix is facilitated by

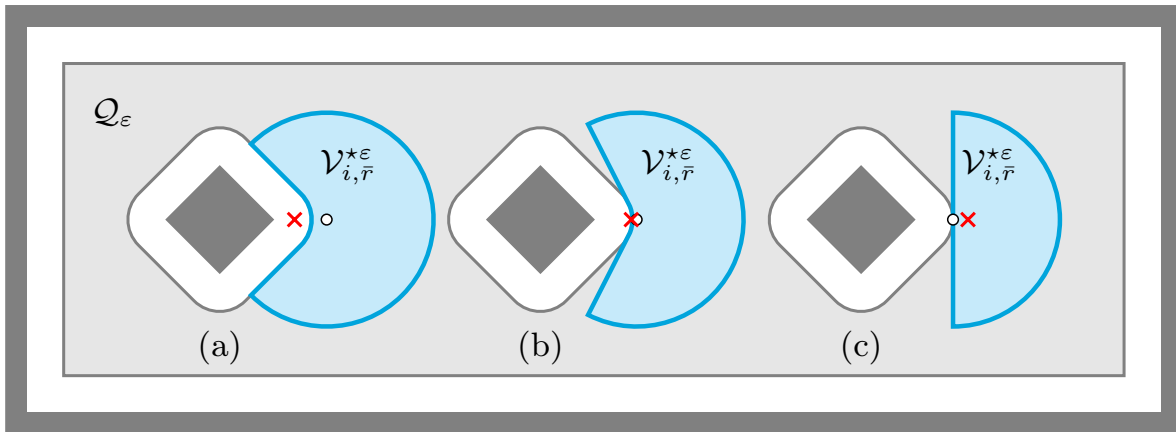
**Corollary 4.4** (Leaving the separatrix in nonconvex environments). Given a nonconvex allowable environment  $\mathcal{Q}$  and its  $\varepsilon$ -contraction  $\mathcal{Q}_\varepsilon$  for an arbitrarily small  $\varepsilon > 0$  and integration range  $\bar{r}$ . If the gradient of  $\mathcal{H}_{\text{discover},i}^\star$  in (4.23) of robot  $i$  vanishes, i.e., if

$$\frac{\partial \mathcal{H}_{\text{discover},i}^\star(\mathbf{p}_i, \mathcal{S}_i)}{\partial \mathbf{p}_i} = \mathbf{0}, \quad (4.25)$$

and if there still exists a frontier  $\tilde{s}_i$  in the Voronoi cell  $\mathcal{V}_i$  of the robot, i.e.,  $\tilde{s}_i \neq \emptyset$ , then the robot is located in an equilibrium point on the separatrix, and an arbitrary perturbation must be added to the gradient for the robot to leave the separatrix and continue exploration.

The implications of Theorem 4.5 and Corollary 4.3 are depicted in Figure 4.11 for a nonconvex allowable environment  $\mathcal{Q}$  and a robot that approaches the concave vertex of an obstacle. Getting closer to the boundary  $\partial\mathcal{Q}_\varepsilon$  increasingly restricts  $\mathcal{V}_{i,\bar{r}}^{\star\varepsilon}$  as depicted in Figure 4.11(a)–4.11(c). As a result, the centroid of the  $\bar{r}$ -limited visibility set  $\mathcal{V}_{i,\bar{r}}^{\star\varepsilon}$  moves away from the obstacle the more  $\mathcal{V}_{i,\bar{r}}^{\star\varepsilon}$  is “convexified” by the tangent. Hence, trajectories on the separatrix always approach equilibrium points in the  $\varepsilon$ -contraction  $\mathcal{Q}_\varepsilon$ . This observation further allows to derive the following conclusion.

**Corollary 4.5** (Collision avoidance with the environment). Since  $\mathcal{Q}_\varepsilon$  defines an invariant set for arbitrary small  $\varepsilon > 0$ , i.e., all trajectories



**Figure 4.11:** Centroid  $\times$  of the  $\bar{r}$ -limited visibility set  $\mathcal{V}_{i, \bar{r}}^{* \varepsilon}$  in the  $\varepsilon$ -contraction  $\mathcal{Q}_\varepsilon$

starting in  $\mathcal{Q}_\varepsilon$  always remain in  $\mathcal{Q}_\varepsilon$ , the parameter  $\varepsilon$  allows for a sufficient condition to avoid collisions with the environment: If the robots have physical diameter  $\text{diam}_{\text{robot}}$  collisions with the environment are avoided by setting

$$\varepsilon \geq \frac{1}{2} \text{diam}_{\text{robot}}. \quad (4.26)$$

In contrast to the necessary condition (4.8) derived in Section 4.2.3 (page 52) for convex environments, condition (4.26) describes a *sufficient* condition for collision avoidance with the nonconvex allowable environments. Since convex allowable environments are a special case of nonconvex allowable environments, Corollary 4.5 can be transferred to convex allowable environments as well by first applying the  $\varepsilon$ -contraction for sufficiently large  $\varepsilon > 0$ .

*Remark 4.4* (Collision avoidance among robots). Currently, the  $\bar{r}$ -limited visibility sets  $\mathcal{V}_{i, \bar{r}}^{* \varepsilon}$  are defined in the  $\varepsilon$ -contraction  $\mathcal{Q}_\varepsilon$ . Hence, the boundary separating two Voronoi cells is not  $\varepsilon$ -contracted beforehand. In order to guarantee collision avoidance among robots, the  $\varepsilon$ -contraction must be individually applied to the Voronoi cells  $\mathcal{V}_i$  before computing  $\mathcal{V}_{i, \bar{r}}^{* \varepsilon}$ , i.e.,

$$\mathcal{V}_{i, \bar{r}}^{* \varepsilon} = \{\mathbf{q} \in \mathcal{V}_i \mid \|\mathbf{p}_i - \mathbf{q}\| \leq \bar{r} \text{ and } [\mathbf{p}_i, \mathbf{q}] \subset \mathcal{V}_{i, \varepsilon}\}. \quad (4.27)$$

However, for the remainder of this dissertation the  $\bar{r}$ -limited visibility sets  $\mathcal{V}_{i, \bar{r}}^{* \varepsilon}$  are defined as in (4.21).

*Remark 4.5.* The  $\bar{r}$ -limited visibility sets  $\mathcal{V}_{i, \bar{r}}^{* \varepsilon}$  are defined based on the  $\varepsilon$ -contraction  $\mathcal{Q}_\varepsilon$  for an arbitrarily small  $\varepsilon > 0$ . From Theorem 4.5 follows, that this contraction is mandatory in nonconvex environments. However,

this contraction is not to be confused with the  $\delta$ -contraction  $\mathcal{Q}_{\delta=\bar{r}}$ . As discussed, the latter defines a defensive approximation of the reachability set.

Following Remark 4.5, all nonconvex allowable environments  $\mathcal{Q}$  are from now on subject to the (mandatory)  $\varepsilon$ -contraction to obtain  $\mathcal{Q}_\varepsilon$ , avoiding collisions in concave vertices of  $\partial\mathcal{Q}$  according to Theorem 4.5. Therewith, applying the  $\delta$ -contraction with the integration range  $\bar{r}$  on top of the  $\varepsilon$ -contraction results in the  $\varepsilon + \bar{r}$ -contraction  $\mathcal{Q}_{\varepsilon+\bar{r}} = (\mathcal{Q}_\varepsilon)_{\bar{r}}$ . Analog to  $\mathcal{Q}_{\delta=\bar{r}}$  in convex environments,  $\mathcal{Q}_{\varepsilon+\bar{r}}$  defines an approximation of the reachability set. However, this time  $\mathcal{Q}_{\varepsilon+\bar{r}}$  consists of two parts:  $\mathcal{Q}_\varepsilon$  reflects an invariant set the robots can *never* leave, meaning that  $\mathcal{Q}_\varepsilon$  is exact and *not* an approximation. The additional contraction of  $\bar{r}$  in  $\mathcal{Q}_{\varepsilon+\bar{r}}$  still reflects a defensive approximation due to the non-uniform density function  $\phi$ .

### The Integration Range in Nonconvex Environments

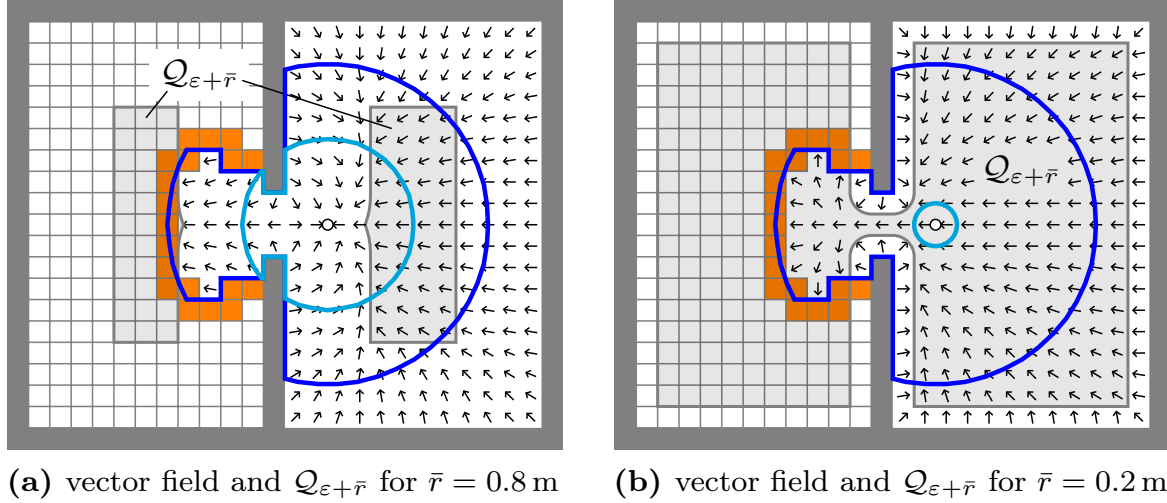
**Finding a lower bound  $\bar{r}_{\min}$ :** In the convex case a lower bound for the integration range  $\bar{r}$  was given in (4.8) (page 52) in terms of a necessary but not sufficient safety distance to the boundary. Contrary, Corollary 4.5 provides a sufficient condition for collision avoidance with the environment by computing the  $\bar{r}$ -limited visibility sets  $\mathcal{V}_{i,\bar{r}}^{\star\varepsilon}$  within the already  $\varepsilon$ -contracted environment  $\mathcal{Q}_\varepsilon$ . Therefore, sufficiently large values  $\varepsilon > 0$  automatically imply collision avoidance with the environment. From this observation follows that there is no lower bound for the integration range  $\bar{r}$ . This formally implies

$$\bar{r} > \bar{r}_{\min} = 0. \quad (4.28)$$

*Remark 4.6.* The integration range  $\bar{r}$  in condition (4.28) must be strictly positive, since for  $\bar{r} = 0$ , the  $\bar{r}$ -limited visibility sets  $\mathcal{V}_{i,\bar{r}}^{\star\varepsilon}$  degenerate into single points according to the definition in (4.21), i.e.,  $\mathcal{V}_{i,\bar{r}}^{\star\varepsilon} = \mathbf{p}_i$ . For  $\mathcal{V}_{i,\bar{r}}^{\star\varepsilon} = \mathbf{p}_i$ , the gradients (4.23) and therewith the motion control laws (4.24) always equal zero in this case, implying the robots stay at fixed positions  $\mathbf{p}_i$  forever.

**Finding an upper bound  $\bar{r}_{\max}$  for a single robot:** For a better understanding of the upper bound  $\bar{r}_{\max}$  of the integration range  $\bar{r}$ , the single-robot case is considered in a nonconvex allowable environment  $\mathcal{Q}$ , represented by a grid map with a cell resolution of  $0.2 \text{ m} \times 0.2 \text{ m}$  as depicted in Figure 4.12. Therein, the environment is divided into a left and a right part, separated by a narrow passage of width  $d_{\text{pass}} = 0.6 \text{ m}$ . Although

not explicitly visualized, it is important to remark that the width of this passage is virtually reduced by the  $\varepsilon$ -contraction  $\mathcal{Q}_\varepsilon$ . In this case,  $\varepsilon > 0$  is assumed to be infinitesimal small.



**Figure 4.12:** Impact of the integration range in nonconvex environments. Legend:  $\circ$  robot position,  $-$  integration range  $\bar{r}$ ,  $-$  sensing range  $r = 1.5$  m. Grid resolution:  $0.2 \text{ m} \times 0.2 \text{ m}$ .

Apparently, the robot must pass through this passage in order to explore the entire environment. From Figure 4.12(a) follows that too large values  $\bar{r}$  for the integration range lead to a disconnected reachability set  $\mathcal{Q}_{\varepsilon+\bar{r}}$ . This becomes also clear by observing that the gradients at the narrow passage define a vector field that prevents the robot from moving to unexplored space. Instead, the trajectories approach a stable equilibrium point in the right part of  $\mathcal{Q}$ . Decreasing the integration range  $\bar{r}$  as depicted in Figure 4.12(b), the reachability set  $\mathcal{Q}_{\varepsilon+\bar{r}}$  is connected. The connectedness of  $\mathcal{Q}_{\varepsilon+\bar{r}}$  leads to the following conclusion.

**Theorem 4.6.** Given a nonconvex allowable environment  $\mathcal{Q}$ , its  $\varepsilon$ -contraction  $\mathcal{Q}_\varepsilon$  for  $\varepsilon > 0$ , and a single robot in  $\mathcal{Q}_\varepsilon$ . Then, the robot with integration range  $\bar{r} > 0$  is able to explore the entire environment if the reachability set  $\mathcal{Q}_{\varepsilon+\bar{r}}$  is connected.

*Proof.* The  $\bar{r}$ -limited visibility set  $\mathcal{V}_{i,\bar{r}}^{\star\varepsilon}$  in  $\mathcal{Q}_{\varepsilon+\bar{r}}$  does not intersect with the boundary  $\partial\mathcal{Q}_\varepsilon$ . Since the reachability set  $\mathcal{Q}_{\varepsilon+\bar{r}}$  is connected, trajectories through the narrow passage in  $\mathcal{Q}_{\varepsilon+\bar{r}}$  exist where  $\mathcal{V}_{i,\bar{r}}^{\star\varepsilon}$  is radially unbounded, implying that no repulsive force impacts the gradients. As a result, if the density  $\phi$  increases along the narrow passage, the robot can pass through the narrow passage and continue exploration.  $\square$



From Theorem 4.6 an upper bound of the integration range immediately follows.

**Corollary 4.6.** Given a nonconvex allowable environment  $\mathcal{Q}$  and a single robot. Denote with  $d_{\text{pass}}$  the diameter of the narrowest passage of  $\mathcal{Q}$  the robot needs to pass in order to explore the entire environment. Then, the upper bound of the integration range  $\bar{r}$  is defined by

$$\bar{r}_{\text{max}} = \frac{d_{\text{pass}} - 2\varepsilon}{2} = \frac{d_{\text{pass}}}{2} - \varepsilon \quad (4.29)$$

and  $\mathcal{Q}_{\varepsilon + \bar{r}_{\text{max}}}$  defines the connected reachability set.

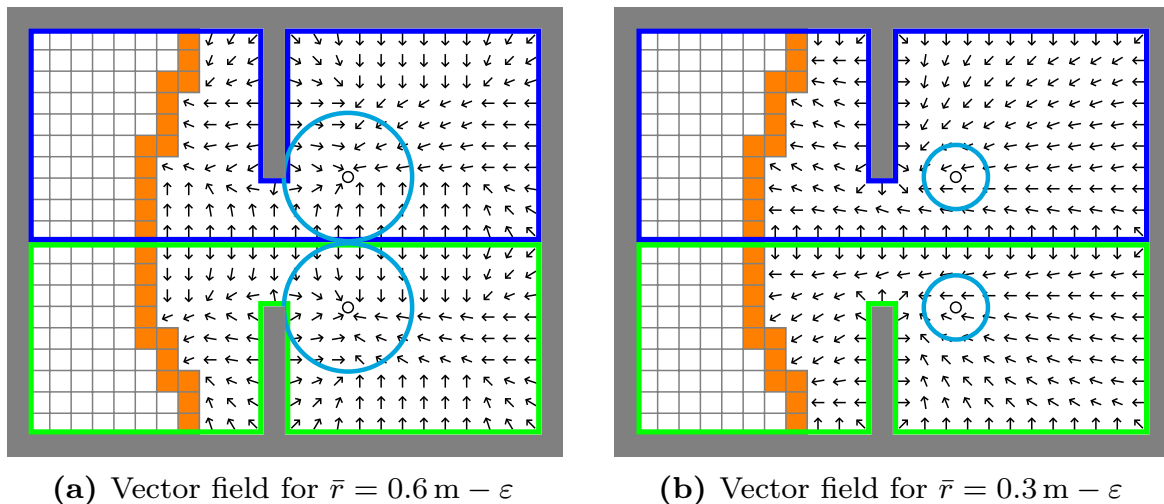
If (4.29) is violated,  $\mathcal{Q}_{\varepsilon + \bar{r}}$  defines a disconnected reachability set and consequently full exploration of the entire environment is not guaranteed due to the trap situation in the equilibrium points. According to Koren and Borenstein [77], such trap situations are a major drawback of gradient-based motion control laws. However, satisfying (4.29) in Corollary 4.6 prevents this drawback, and exploration of the environment  $\mathcal{Q}$  is guaranteed.

In Figure 4.12(b), the cell resolution of the grid map is  $0.2 \text{ m} \times 0.2 \text{ m}$ . Accordingly, the diameter of the narrow passage is  $d_{\text{pass}} = 0.6 \text{ m}$ . Following Corollary 4.6, the upper bound for the integration range  $\bar{r}$  is denoted by  $\bar{r}_{\text{max}} = 0.3 \text{ m} - \varepsilon$ .

**Generalizing  $\bar{r}_{\text{max}}$  to  $N$  robots:** The upper bound  $\bar{r}_{\text{max}}$  for the integration range in (4.29) unfortunately only holds for the single-robot case. This gets clear by investigating Figure 4.13. Therein, two robots are placed in a nonconvex environment  $\mathcal{Q}$  that again contains a narrow passage of width  $d_{\text{pass}} = 1.2 \text{ m}$ . Inserting  $d_{\text{pass}}$  into (4.29) results in a maximum integration range  $\bar{r}_{\text{max}} = 0.6 \text{ m} - \varepsilon$  with an  $\varepsilon > 0$ . However, choosing  $\bar{r} = 0.6 \text{ m} - \varepsilon$  leads to a deadlock situation as depicted in Figure 4.13(a). The stable equilibrium points can be explained by observing that the boundary of the Voronoi cells act as repulsive force, exactly like the boundary  $\partial\mathcal{Q}_{\varepsilon}$ . In fact, although unlikely, if using  $N$  robots, all of the  $N$  robots might want to move through the narrow passage simultaneously. This observation leads to the following generalization.

**Theorem 4.7** (Maximum integration range). Using  $N$  robots to explore a nonconvex allowable environment  $\mathcal{Q}$  with  $\varepsilon$ -contraction  $\mathcal{Q}_{\varepsilon}$  and integration range  $\bar{r}$ . Then, the robots explore the entire environment and  $\mathcal{S}(t) \rightarrow \mathcal{Q}$  if the integration range  $\bar{r}$  satisfies the upper bound

$$\bar{r} \leq \bar{r}_{\text{max}} = \frac{d_{\text{pass}} - 2\varepsilon}{2N}. \quad (4.30)$$



**Figure 4.13:** Deadlock situations in narrow passages. Legend:  $\circ$  robot position, — integration range  $\bar{r}$ , — Voronoi partition, grid resolution:  $0.2 \text{ m} \times 0.2 \text{ m}$ .

Therein,  $d_{\text{pass}}$  denotes the width of the narrowest passage the robots need to pass in order to explore  $\mathcal{Q}$ .

*Proof.* The term  $d_{\text{pass}} - 2\varepsilon$  denotes the width of the passage in the  $\varepsilon$ -contraction  $\mathcal{Q}_\varepsilon$ . Coincidentally, situations may appear in which each robot has assigned parts of the frontier on the other side of the passage. Therefore, it may happen that  $N$  robots want to pass the passage simultaneously. Therefore, in order to avoid the repulsive force of the boundary of the Voronoi cells  $\mathcal{V}_i$ ,  $d_{\text{pass}} - 2\varepsilon$  must be divided by the robot count  $N$ . Further, the factor 2 in the denominator turns the diameter  $d_{\text{pass}} - 2\varepsilon$  into a radius, which completes the proof.  $\square$

Applying Theorem 4.7 to the previous example with  $N = 2$  robots results in Figure 4.13(b). Since the diameter is denoted by  $d_{\text{pass}} = 1.2 \text{ m}$ , the upper bound yields  $\bar{r}_{\text{max}} = 0.3 \text{ m} - \frac{\varepsilon}{2}$ . The vector field for  $\bar{r} = \bar{r}_{\text{max}}$  with arbitrarily small  $\varepsilon > 0$  shows that both robots now are able to pass the narrow passage simultaneously. Therefore, deadlock situation in stable equilibrium points as in Figure 4.13(a) are successfully avoided.

### On Choosing the Sensing Range

Section 4.2.3 introduced a lower bound  $r_{\text{min}}$  for the sensing range  $r$  in (4.10) in terms of the right triangle defined by the hypotenuse, the smallest inner angle  $\alpha$  and the integration range  $\bar{r}$  (cf. page 54). Since the  $\delta$ -contraction

is now defined by  $\delta = \varepsilon + \bar{r}$ , (4.10) needs to be adapted according to the following theorem.

**Theorem 4.8** (Lower bound for  $r$  in nonconvex allowable environments). Given  $N$  robots with integration range  $\bar{r}_{\max} \geq \bar{r} > \bar{r}_{\min}$ . Denote with  $\alpha$  the smallest interior angle of the nonconvex allowable environment  $\mathcal{Q}$ . The minimum sensing range  $r_{\min}$  required to guarantee exploration of  $\mathcal{Q}$  is defined by

$$r_{\min} = \frac{\varepsilon + \bar{r}}{\sin(\frac{\alpha}{2N})}. \quad (4.31)$$

*Proof.* Instead of  $\mathcal{Q}_{\delta=\bar{r}}$ , the reachability set in nonconvex allowable environments is defined by  $\mathcal{Q}_{\varepsilon+\bar{r}}$ . Based on this reachability set, the proof is performed analogously to the proof of Theorem 4.2 (page 54).  $\square$

*Remark 4.7.* Analog to Remark 4.2 (page 55) for the convex case, from (4.31) the necessary condition  $r_{\min} \geq \bar{r}$  follows, since  $\sin(\frac{\alpha}{2N}) \in (0, 1]$  for the smallest interior angle  $\alpha \in (0, \pi]$  and  $\varepsilon + \bar{r} > \bar{r}$ .

### Stability Analysis and Proof of Convergence

The idea of the stability analysis as well as the proof of convergence are performed analogously to the convex case in Section 4.2.3 (page 51). However, it is necessary to emphasize that the analysis for nonconvex environments  $\mathcal{Q}$  only holds if Assumption 4.1 and Assumption 4.2 are valid. The validity of Assumption 4.1 is assured by the discussion in Section 4.2.2 (page 50). Assumption 4.2 is also justified by the following remark.

*Remark 4.8* (Note on Assumption 4.2). Since the integration range  $\bar{r}$  is chosen following Theorem 4.7 (page 69), trajectories through  $\mathcal{Q}_{\varepsilon+\bar{r}_{\max}}$  exist that are radially unbounded. Therewith, the  $\bar{r}$ -limited visibility sets  $\mathcal{V}_{i,\bar{r}}^{*\varepsilon}$  are convex. This in turn implies that the control laws are equal to (4.7) (page 50) of the convex case as discussed in Section 4.2.3 (page 51ff). In the radially unbounded case, Assumption 4.2 is inactive, and therefore not required most of the time.

Following Remark 4.8, Assumption 4.2 is a weak restriction. This observation is essential in the stability and convergence analysis that follows. First, convergence to the centroidal Voronoi partition is shown.

**Theorem 4.9** (Frontier-based centroidal search). Let  $\mathcal{Q}$  denote a nonconvex allowable environment and let  $\mathcal{S}$  denote the explored region in  $\mathcal{Q}$ . Denote with  $\mathcal{Q}_\varepsilon$  the  $\varepsilon$ -contraction for an arbitrarily small  $\varepsilon > 0$ . Let

$\mathcal{P} = \{\mathbf{p}_1, \dots, \mathbf{p}_N\}$  be the configuration of  $N$  robots in  $\mathcal{Q}_\varepsilon$  with integration range  $\bar{r}$ , and denote with  $\mathcal{V} = \{\mathcal{V}_1, \dots, \mathcal{V}_N\}$  the geodesic Voronoi partition of  $\mathcal{Q}$  for  $\mathcal{P}$ . Then, applying (4.22)–(4.24) results in a  $\bar{r}$ -limited centroidal Voronoi partition for  $t \rightarrow \infty$ .

*Proof.* Building the time-derivative of (4.22) and inserting (4.24) yields

$$\begin{aligned} \dot{\mathcal{H}}_{\text{discover}}^*(\mathcal{P}, \mathcal{S}) &= \frac{\partial \mathcal{H}_{\text{discover}}^*(\mathcal{P}, \mathcal{S})}{\partial \mathcal{P}} \dot{\mathcal{P}} \\ &= - \sum_{i=1}^N \left\| \frac{\partial \mathcal{H}_{\text{discover},i}^*(\mathbf{p}_i, \mathcal{S}_i)}{\partial \mathbf{p}_i} \right\|^2. \end{aligned} \quad (4.32)$$

Under Assumption 4.1 and Assumption 4.2, the partial derivatives of  $\mathcal{H}_{\text{discover},i}^*$  with respect to the robot positions  $\mathbf{p}_i$  are given by (4.23). Inserting (4.23) into (4.32) is equivalent to

$$\dot{\mathcal{H}}_{\text{discover}}^*(\mathcal{P}, \mathcal{V}) = -k_p \sum_{i=1}^N \left\| \mathbf{p}_i - \mathbf{m}_\phi(\mathcal{V}_{i,\bar{r}}^{*\varepsilon}) \right\|^2 \leq 0. \quad (4.33)$$

The sum of squares as well as  $k_p$  in (4.33) are non-negative. Applying the Krasovskii-LaSalle invariance principle, the robots move to the largest invariant set, which equals the set of all  $\bar{r}$ -limited centroidal Voronoi configurations. According to Theorem 4.5, the equilibrium points of the  $\bar{r}$ -limited centroidal Voronoi configurations are contained in  $\mathcal{Q}_\varepsilon \subset \mathcal{Q}$ .  $\square$

*Remark 4.9 (Related work).* In essence, Theorem 4.9 transfers the solution to the coverage problem in convex domains to nonconvex allowable environments based on the  $\bar{r}$ -limited visibility sets  $\mathcal{V}_{i,\bar{r}}^{*\varepsilon}$  in  $\mathcal{Q}_\varepsilon$ . Therewith, this coverage approach directly extends the work of Lu et al. [88] and Marier et al. [92] (and to a lesser extent also the work of Pimenta et al. [103] and Bhattacharya et al. [27]) with the difference, that the projection method of centroids (cf. Remark 4.3, page 63) is not required at the expense of calculating the  $\bar{r}$  limited visibility sets in the  $\varepsilon$ -contraction  $\mathcal{Q}_\varepsilon$ . The projection method is not required, since the motion controls are always such that the robots remain in the invariant sets  $\mathcal{Q}_\varepsilon$ . Solving the nonconvex coverage problem in the  $\varepsilon$ -contracted visibility sets  $\mathcal{V}_{i,\bar{r}}^{*\varepsilon}$  was published in Klodt et al. [12].

Based on Theorem 4.5 (page 64), Corollaries 4.3 and 4.4 (page 65), and Theorem 4.9, a proof of convergence of the proposed exploration approach for nonconvex allowable environments is given next.

**Theorem 4.10** (Proof of convergence). Let  $\mathcal{Q}$  denote a convex allowable environment and let  $\mathcal{S}$  denote the explored region in  $\mathcal{Q}$ . Denote with  $\mathcal{Q}_\varepsilon$  the  $\varepsilon$ -contraction for an arbitrarily small  $\varepsilon > 0$ . Let  $\mathcal{P} = \{\mathbf{p}_1, \dots, \mathbf{p}_N\}$  be the configuration of  $N$  robots in  $\mathcal{Q}_\varepsilon$  with integration range  $\bar{r}$ , and denote with  $\mathcal{V} = \{\mathcal{V}_1, \dots, \mathcal{V}_N\}$  the geodesic Voronoi partition of  $\mathcal{Q}$  for  $\mathcal{P}$ . Then, if  $0 = \bar{r}_{\min} < \bar{r} \leq \bar{r}_{\max}$  according to (4.30) in Theorem 4.7 (page 69), and  $r \geq r_{\min}$  according to (4.31) in Theorem 4.8 (page 71) hold, and applying Corollary 4.3, continuous optimization of (4.22) and the motion control laws (4.24) solve the multi-robot exploration problem and  $\mathcal{S}(t) \rightarrow \mathcal{Q}$  as  $t \rightarrow \infty$ .

*Proof.* Choosing the integration range  $\bar{r}$  according to Theorem 4.7 ensures that trajectories generated by the gradient-based control law (4.24) approach the frontier, converging to an  $\bar{r}$ -limited centroidal Voronoi configuration as stated by Theorem 4.9. Thereby, local maxima on the separatrix are avoided by applying Corollary 4.4 based on Theorem 4.5 and Corollary 4.3. Since for the sensing range  $r \geq r_{\min}$  holds according to Theorem 4.8, the frontier is within sensing range  $r$  and therefore pushed back, which in turn changes the density function  $\phi(\mathbf{q}, \tilde{s}_i)$ . Therewith, the  $\bar{r}$ -limited centroidal Voronoi configuration is never reached and the exploration continues until the entire environment is explored, in which case  $\mathcal{S}(t) = \mathcal{Q}$  holds.  $\square$

*Remark 4.10* (Relation to convex environments). If the geodesic paths from all  $\mathbf{q} \in \mathcal{V}_{i,\bar{r}}^*$  to the frontier  $\tilde{s}_i$  equal the Euclidean paths, then, for  $\varepsilon = 0$ , the exploration behaves exactly like the one in convex environments.

### 4.2.5 Fallback Strategy when Using Multiple Robots

In the previous subsections the characteristics of the centroidal search-based DisCoverage approach were investigated. In addition, a formal proof of convergence for convex as well as nonconvex allowable environments was given. Next, it is of interest how the centroidal search-based DisCoverage approach behaves when using multiple robots. To this end, it is helpful to first formulate the closed feedback loop in Figure 4.8 in terms of Algorithm 1. In line with the introduction in Section 4, instead of  $\mathbf{p}_i(t)$ ,  $\tilde{s}(t)$ ,  $\mathcal{V}_i(\mathcal{P}(t))$ ,  $\tilde{s}_i(\mathcal{P}(t), t)$  the respective variables are written without the dependency on time and the partition  $\mathcal{P}$  for better readability. Further, the communication of  $\Delta\mathcal{V}_{i \rightarrow j}$  equals the exchanged map data as described in Figure 4.1.

Next, Algorithm 1 is applied to a nonconvex allowable environment with  $N = 3$  robots in Figure 4.14. As can be seen in Figure 4.14(a), the robots evenly spread into the environment, exploring unknown space within the

---

**Algorithm 1** Continuous-Time DisCoverage Feedback Loop
 

---

 Initialization of  $\mathbf{p}_i(0)$  and  $\mathcal{S}_i(0)$  for  $i = 1, \dots, N$ 
**while**  $\mathcal{S}(t) \neq \mathcal{Q}$  **do**
**for**  $i = 1, \dots, N$  in parallel **do**

 Partition: communicate  $\mathbf{p}_i$  to Voronoi neighbors  $j$   
 compute  $\mathcal{V}_i$  and  $\mathcal{V}_{i,\bar{r}}^{\star\varepsilon}$   
 compute  $\mathcal{S}_i$  and  $\tilde{s}_i$ 

 Motion control: compute  $\phi(\mathbf{q}, \tilde{s}_i)$  for  $\mathbf{q} \in \mathcal{V}_{i,\bar{r}}^{\star\varepsilon}$ 

$$\dot{\mathbf{p}}_i = -\frac{\partial \mathcal{H}_{\text{discover},i}^{\star}(\mathbf{p}_i, \mathcal{S}_i)}{\partial \mathbf{p}_i}$$

 communicate  $\Delta \mathcal{V}_{i \rightarrow j}$  to respective neighbor  $j$ 

 Exploration: map  $\mathcal{V}_{i,r}^{\star}$ 
**end for**
**end while**

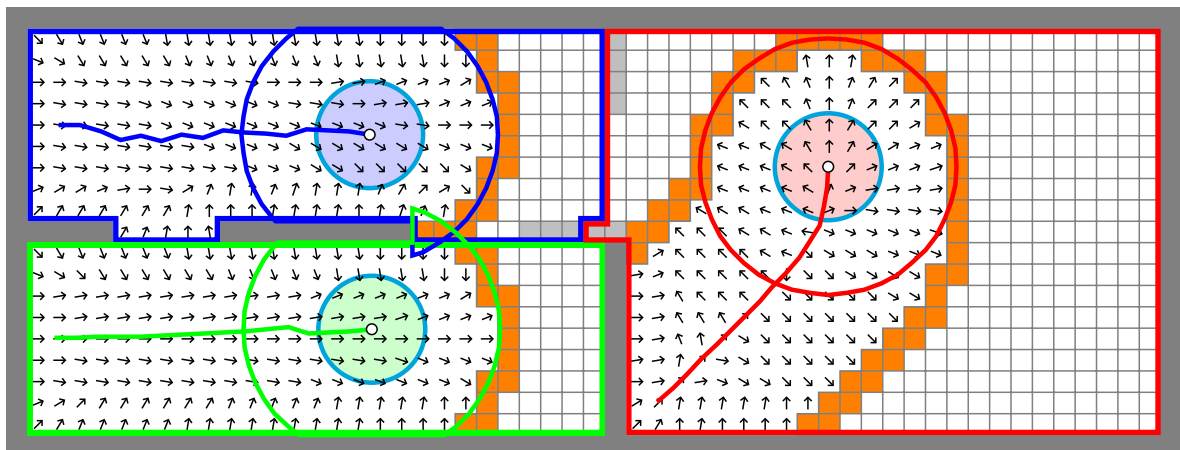

---

respective geodesic Voronoi cells. In addition to the geodesic Voronoi cells, the trajectories as well as the vector field for each robot are shown. It is worth to note that the boundary of the geodesic Voronoi cells acts as a repulsive force, and thus behaves like the boundary of obstacles. This is a desired property, since therewith collisions among the robots are avoided.

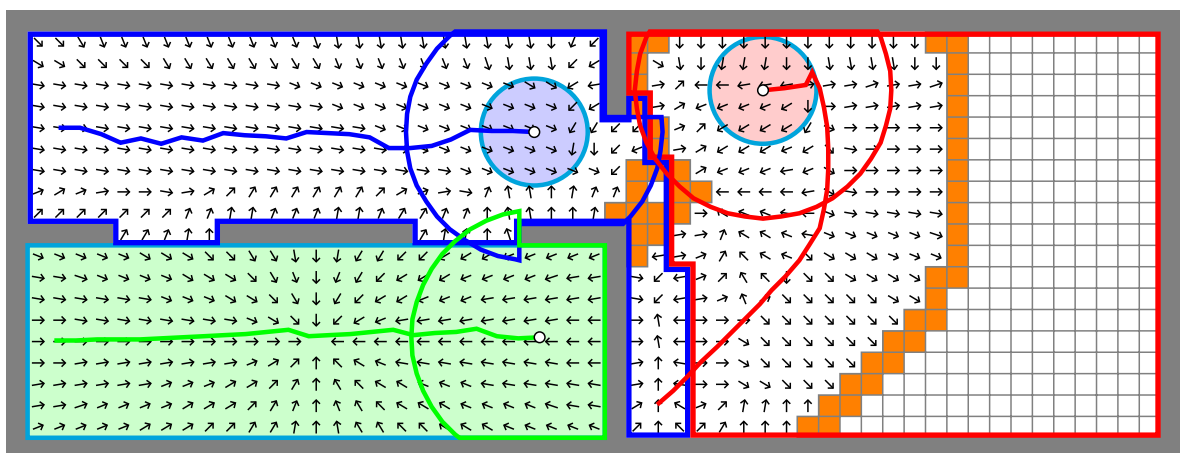
After 23 iterations (cf. Figure 4.14(b)), the robot centered in the lower part of the environment does not have any frontier in its Voronoi cell. Therefore, the gradient vanishes and

$$\dot{\mathbf{p}}_i = -\frac{\partial \mathcal{H}_{\text{discover},i}^{\star}(\mathbf{p}_i, \mathcal{S}_i)}{\partial \mathbf{p}_i} = \mathbf{0} \quad (4.34)$$

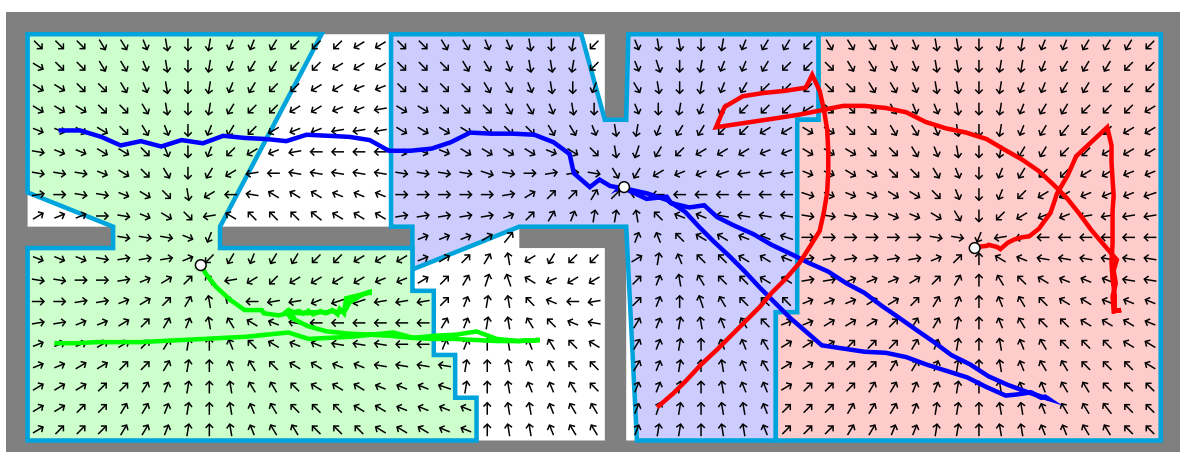
holds. This is an undesirable effect when using a partition of the environment to assign a region of dominance to each robot, since robots with fully explored Voronoi cells do not contribute to the exploration process anymore. To avoid robots from remaining in their position, robots without a frontier fall back to coverage in the *unlimited* visibility set. This unlimited centroidal search within the respective  $\varepsilon$ -contracted Voronoi cell equals the centroidal search with an unbounded integration range  $\bar{r}$  and a constant density function  $\phi$ . The notion of a **fallback** or **secondary strategy** was first introduced in Frank et al. [2]. It is worth to emphasize the difference to the unlimited centroidal search by Pimenta et al. [103]: There, the authors maximize coverage in the entire geodesic Voronoi cells by first applying a transformation of the environment to handle nonconvexity before optimizing the coverage functional. Contrary, the proposed fallback



(a) robot configuration after 15 iterations



(b) robot configuration after 23 iterations



(c) centroidal Voronoi configuration after 85 iterations

**Figure 4.14:** DisCoverage for  $N = 3$  robots in a nonconvex environment. Parameters: integration range  $\bar{r} = 0.499$  m,  $\varepsilon = 0.001$ , sensing range  $r = 1.2$  m.

strategy optimizes the coverage functions only for the unlimited visibility sets  $\mathcal{V}_i^{\star\epsilon}$  similar to Marier et al. [92].

The exploration process continues until the entire environment is explored after 65 iterations. Then, all robots apply the centroidal search within the unlimited visibility set according to the fallback strategy. Another 20 iterations later, the robots reach a centroidal Voronoi configuration (cf. Figure 4.14(c)). The Voronoi partition is omitted, since it equals the respective shaded unlimited visibility sets in large parts. Further, the sensing ranges are omitted for clarity. The white parts in Figure 4.14(c) depict the uncovered area.

*Remark 4.11* (The degree of star-shapedness). The unlimited visibility set  $\mathcal{V}_i^\star = \{\mathbf{q} \in \mathcal{V}_i \mid [\mathbf{p}_i, \mathbf{q}] \subseteq \mathcal{V}_i\}$  allows for an interpretation in terms of the **degree of star-shapedness** of  $\mathcal{V}_i$  with respect to the robot position  $\mathbf{p}_i$ . Denoting with  $|\cdot|$  the area of a set, the degree of star-shapedness can formally be defined as

$$J^\star = \frac{|\mathcal{V}_i^\star|}{|\mathcal{V}_i|} \in [0, 1]. \quad (4.35)$$

A value of  $J^\star = 1$  implies that the entire Voronoi cell is visible from  $\mathbf{p}_i$ . Contrary, values less than one imply that only parts of the Voronoi cell are visible. Further measures in this research area quantify the nonconvexity of sets, e.g., Boxer [32], Rote [109], Zunic and Rosin [139].

In the context of the fallback strategy, Remark 4.11 allows to evaluate the optimality of the coverage functional. A value of  $J^\star = 1$  is equivalent to covering the entire Voronoi cell. Decreasing values imply that only a subset of the Voronoi cell is visible from  $\mathbf{p}_i$ , and therefore only a subset of  $\mathcal{V}_i$  is covered. The lower the degree of star-shapedness of  $\mathcal{V}_i$  with respect to  $\mathbf{p}_i$ , the more non-starshaped is the Voronoi cell  $\mathcal{V}_i$  with respect to the robot position  $\mathbf{p}_i$ . Low values  $J^\star$  may imply that a robot should search for another location with increased  $J^\star$  that maximizes coverage. It is noteworthy that the idea of using the degree of star-shapedness as indicator for optimal coverage positions is new and therefore qualifies as contribution of this dissertation.

## 4.2.6 Possible Extensions to the Vehicle Dynamics

Up to this point, DisCoverage was introduced for the single integrator dynamics

$$\dot{\mathbf{p}}_i = \mathbf{u}_i \quad (4.36)$$



with motion control laws

$$\mathbf{u}_i = -\frac{\partial \mathcal{H}_{\text{discover}}(\mathcal{P}, \mathcal{S})}{\partial \mathbf{p}_i} \quad (4.37)$$

for each robot  $i = 1, \dots, N$  in convex environments (with  $\mathcal{H}_{\text{discover},i}^*(\mathbf{p}_i, \mathcal{S}_i)$  in nonconvex environments). However, Cortés et al. [50] show that the centroidal search can also be applied to systems with passive dynamics such as double integrator dynamics by using a proportional derivative (PD) motion control law, as well as to unicycle dynamics following the state space model

$$\begin{bmatrix} \dot{p}_i^x \\ \dot{p}_i^y \\ \dot{\delta}_i \end{bmatrix} = \begin{bmatrix} \cos \delta_i & 0 \\ \sin \delta_i & 0 \\ 0 & 1 \end{bmatrix} \mathbf{u}_i, \quad \text{with} \quad \mathbf{u}_i = \begin{bmatrix} v_i \\ \omega_i \end{bmatrix}, \quad (4.38)$$

where  $\delta_i$  denotes the orientation of the robot and  $\mathbf{u}_i$  denotes its control input with velocity  $v_i \in \mathbb{R}_{\geq 0}$  and angular velocity  $\omega_i$ . Cortés et al. [50] propose a slightly modified version of the controller in Astolfi [22] to solve the centroidal search with unicycle dynamics (4.38).

*Remark 4.12* (Arbitrary vehicle dynamics). Bullo et al. [36] advocate that the centroidal search can be applied to vehicles with arbitrary dynamics in a discrete-time implementation. The only requirement is that all vehicles must strictly decrease the value of the respective contribution to the objective function  $\mathcal{H}$  in the time intervals between communication rounds where the Voronoi cells are updated. Additionally, all vehicles need to strictly remain inside the respective Voronoi cells to avoid collisions with obstacles and other robots. This observation also holds for the proposed DisCoverage approach.

## 4.3 Orientation-Based DisCoverage

The proposed DisCoverage approach in Section 4.2 solely relies on the density to derive motion control laws. The density itself depends on the distance of all points in the  $\bar{r}$ -limited visibility set  $\mathcal{V}_{i,\bar{r}}^*$  to the frontier  $\tilde{s}_i$  for each robot  $i$ . Therefore, the objective function  $\mathcal{H}_{\text{discover}}$  is a frontier-based approach that essentially uses the distance to determine where to move next. However,  $\mathcal{H}_{\text{discover}}$  does not include the length of the frontier which correlates with the expected information gain. Therefore, the centroidal search-based DisCoverage approach does not maximize the

expected information gain. Instead, it solely minimizes the distance to the frontier for all points in the  $\bar{r}$ -limited visibility set  $\mathcal{V}_{i,\bar{r}}^*$ .

The objective of this section is to find an exploration strategy that optimizes both the distance costs as well as the expected information gain. The orientation-based DisCoverage approach for convex environments as presented in the following section was published in Haumann et al. [3, 4]. Section 4.3.5 introduces the orientation-based DisCoverage approach for nonconvex allowable environments, which was published along with lab experiments in Haumann et al. [6].

### 4.3.1 Introducing Robot Orientations

Each robot is equipped with an orientation  $\delta_i \in [-\pi, \pi]$  defining the current moving direction of robot  $i$ . In line with the distributed optimization problem (3.4) and (3.5), the objective function of the **orientation-based DisCoverage** approach (cf. Haumann et al. [4]) reads

$$\begin{aligned} \mathcal{H}_{\text{orient}}(\mathcal{P}, \Delta, \mathcal{S}) &= \sum_{i=1}^N \mathcal{H}_{\text{orient},i}(\mathbf{p}_i, \delta_i, \mathcal{S}_i) \\ &= \sum_{i=1}^N \int_{\tilde{s}_i} f(\mathbf{p}_i, \delta_i, \mathbf{q}) \phi(\mathbf{q}) d\mathbf{q}, \end{aligned} \quad (4.39)$$

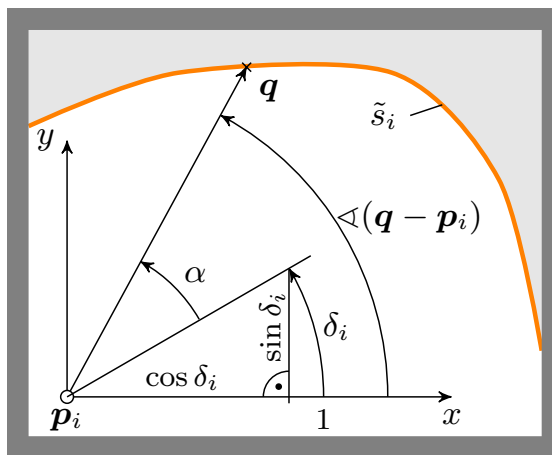
where  $\Delta = \{\delta_1, \dots, \delta_N\}$  denotes the set of all robot orientations, and  $f$  and  $\phi$  denote the performance function and the density function which is constant over time. The objective function  $\mathcal{H}_{\text{orient}}$  differs with respect to  $\mathcal{H}_{\text{discover}}$  in two ways: First, the performance function  $f$  does not take the distance from the robot position  $\mathbf{p}_i$  to a location  $\mathbf{q}$  on the frontier  $\tilde{s}_i$  as parameter. Instead,  $f$  is relaxed and has three degrees of freedom: The robot position  $\mathbf{p}_i$ , its orientation  $\delta_i$  and the respective location  $\mathbf{q}$  on the frontier  $\tilde{s}_i$ . Second,  $\mathcal{H}_{\text{orient},i}$  integrates over the frontier  $\tilde{s}_i$  instead of the  $\bar{r}$ -limited visibility set  $\mathcal{V}_{i,\bar{r}}^*$ . Consequently, optimizing (4.39) always takes the *entire* frontier into account.

Next, the goal of each robot  $i$  is to find an orientation  $\delta_i$  such that as many points  $\mathbf{q} \in \tilde{s}_i$  on the frontier as possible are located directly in front of the robot. This is equivalent to minimizing the absolute value of the angle  $\alpha \in [-\pi, \pi]$  between the orientation  $\delta_i$  and all frontier points  $\mathbf{q} \in \tilde{s}_i$

with

$$\alpha(\mathbf{p}_i, \delta_i, \mathbf{q}) = \underbrace{\sphericalangle(\mathbf{q} - \mathbf{p}_i) - \delta_i}_{\gamma} + \begin{cases} 2\pi & \text{if } \gamma < -\pi, \\ -2\pi & \text{if } \gamma > \pi, \\ 0 & \text{else.} \end{cases} \quad (4.40)$$

For instance, an angle of  $\alpha = \pi$  implies that the considered point  $\mathbf{q}$  lies behind, whereas an angle of  $\alpha = 0$  implies that  $\mathbf{q}$  lies in the direction of the robot's current orientation  $\delta_i$ . Figure 4.15 illustrates the relation of the orientation  $\delta_i$ ,  $\sphericalangle(\mathbf{q} - \mathbf{p}_i)$ , and  $\alpha$ .



**Figure 4.15:** Relation of the robot's orientation  $\delta_i$  and angle  $\alpha$  of the frontier (cf. Haumann et al. [4]). The coordinate system in  $\mathbf{p}_i$  is axis parallel to the global system and independent of the orientation  $\delta_i$ .

Based on the idea of minimizing the absolute value of the angle  $\alpha$  between  $\delta_i$  and the frontier, the continuously differentiable performance function  $f$ , consisting of an *angular component* and a *distance component*, is introduced as a product of two Gaussians as

$$f(\mathbf{p}_i, \delta_i, \mathbf{q}) = \underbrace{\exp\left(-\frac{\alpha(\mathbf{p}_i, \delta_i, \mathbf{q})^2}{2\theta^2}\right)}_{\text{angular component}} \underbrace{\exp\left(-\frac{\|\mathbf{q} - \mathbf{p}_i\|^2}{2\sigma^2}\right)}_{\text{distance component}}. \quad (4.41)$$

Therein, both  $\theta$  and  $\sigma$  describe the standard deviations of the Gaussians. In the following,  $\theta$  is referred to as *opening angle*. Small values of  $\alpha$  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., frontier points  $\mathbf{q}$  closer to  $\mathbf{p}_i$  imply a larger distance component. In summary, the performance function  $f$  yields large

values for frontier points directly in front of the robot orientation, weighted with the distance.

The next section introduces the distributed optimization problem as well as the applied vehicle dynamics first for convex environments. Thereafter, the properties of the proposed performance function are investigated and then the concept is transferred to nonconvex allowable environments.

### 4.3.2 Continuous-Time Control Law

Finding the optimal orientations  $\delta_i^*$  is equivalent to the optimization problem

$$\delta_i^* = \arg \max_{\delta_i} \mathcal{H}_{\text{orient},i}(\mathbf{p}_i, \delta_i, \mathcal{S}_i). \quad (4.42)$$

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

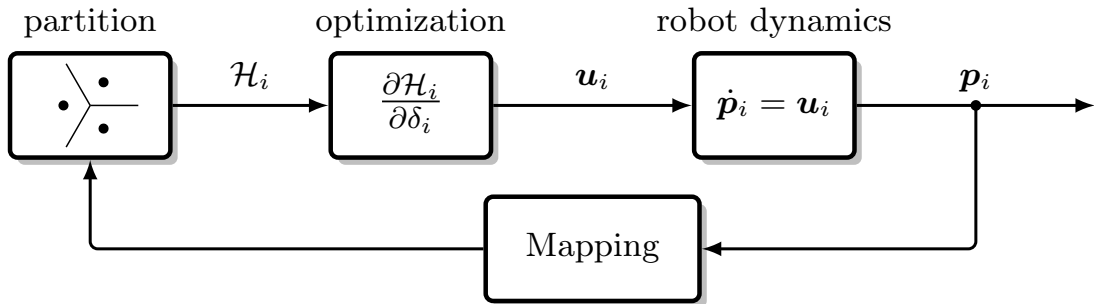
$$\begin{aligned} \frac{\partial \mathcal{H}_{\text{orient},i}(\mathbf{p}_i, \delta_i, \mathcal{S}_i)}{\partial \delta_i} &= \int_{\tilde{\mathcal{S}}_i} \frac{\partial f}{\partial \alpha} \frac{\partial \alpha}{\partial \delta_i} \phi(\mathbf{q}) d\mathbf{q} \\ &= \int_{\tilde{\mathcal{S}}_i} \frac{\alpha(\mathbf{p}_i, \delta_i, \mathbf{q})}{\theta^2} f(\mathbf{p}_i, \delta_i, \mathbf{q}) \phi(\mathbf{q}) d\mathbf{q} \stackrel{!}{=} 0. \end{aligned} \quad (4.43)$$

Using the optimal orientations  $\delta_i^*$  from (4.42), the design of a simple control law for each robot is possible by simplifying the unicycle dynamics in (4.38) to the first order dynamic system

$$\dot{\mathbf{p}}_i = \mathbf{u}_i = v \begin{pmatrix} \cos \delta_i^* \\ \sin \delta_i^* \end{pmatrix}. \quad (4.44)$$

The control input  $\mathbf{u}_i$  is denoted by the constant velocity  $v \in \mathbb{R}_{\geq 0}$  of all robots and the optimal orientation  $\delta_i^*$ . As depicted in Figure 4.16 for  $\mathcal{H}_i = \mathcal{H}_{\text{orient},i}$ , equations (4.44) and (4.42) represent a closed loop as the optimization of the orientations depends on the robot positions and the Voronoi partition, which change continuously over time as the robots move.

Similar to the orientation-based optimization in (4.43), Gusrialdi et al. [67] first optimized an objective function with respect to the orientation to align an anisotropic sensor model for optimal coverage. The proposed optimization significantly differs from the one in Gusrialdi et al. [67] in that  $\mathcal{H}_{\text{orient}}$  integrates over the frontier instead of the area within sensing range.



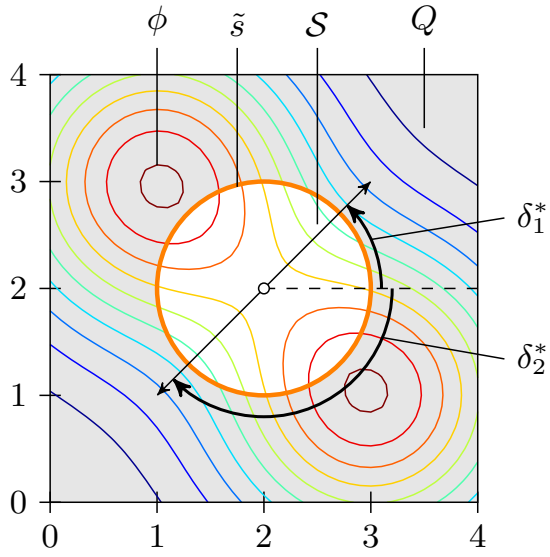
**Figure 4.16:** Orientation-based DisCoverage continuous-time feedback loop (cf. Haumann et al. [4])

### 4.3.3 Impact of the Angular Component

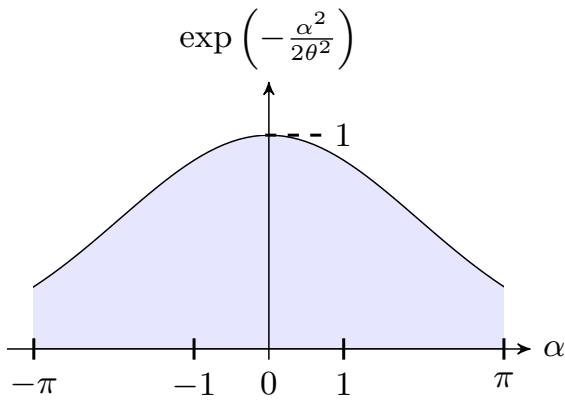
This section discusses the proper choice of the opening angle  $\theta$  in the angular component. For a better understanding the scene depicted in Figure 4.17 and Figure 4.18 is considered. Both figures show the same scene  $\mathcal{Q}$  with a single robot in  $\mathbf{p} = [2 \ 2]^\top$  m with a sensing range of  $r = 1$  m. The robot already explored the environment  $\mathcal{S} \subset \mathcal{Q}$  within its sensing range, resulting in a circular border  $\tilde{s}$  separating the explored parts from the unknown area. Illustrated by the contour lines, the scene contains a density function consisting of a mixture of two Gaussians with means at  $[1 \ 3]^\top$  m and  $[3 \ 1]^\top$  m. The density  $\phi$  can be thought of as a priori knowledge, indicating more important regions in  $\mathcal{Q}$  that should be explored first.

Next, both figures are used to analyze the impact of the opening angle  $\theta$  in the angular component. The distance component does not have any effect in this case as the Euclidean norm from  $\mathbf{p}$  to all frontiers  $\mathbf{q} \in \tilde{s}$  is the same. In Figure 4.17, the opening angle is set to  $\theta = 2$  (cf. Figure 4.17(b)), whereas the opening angle in Figure 4.18 is set to  $\theta = 0.5$  (cf. Figure 4.18(b)). Both Gaussians show a single maximum in  $\alpha = 0$ , since in this case the arguments of the exponential functions are zero. As a result, locations  $\mathbf{q} \in \mathcal{S}$  in the direction of the orientation  $\delta$  are maximally weighted.

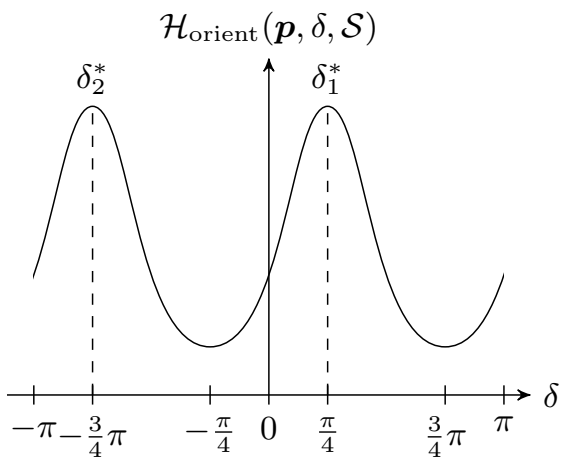
The objective function  $\mathcal{H}_{\text{orient}}$  for the robot positioned in  $\mathbf{p}$  in Figure 4.17(a) and Figure 4.18(a) is depicted in Figure 4.17(c) and Figure 4.18(c), respectively, over the  $\delta$  axis within the domain  $[-\pi, \pi]$ . Apparently, the optimal orientations  $\delta_1^*$  and  $\delta_2^*$  in Figure 4.17(c) fundamentally differ from the optimal orientations  $\delta_3^*$  and  $\delta_4^*$  in Figure 4.18(c). The exponential function in the angular component tends to become parallel to the  $\alpha$  axis for opening angles  $\theta \rightarrow \infty$  (cf. Figure 4.17(b)). This is equivalent to an angular component equal to 1 *independent* of the angle  $\alpha$ . In fact, already



(a) scene showing optimal  $\delta$  for  $\theta = 2$

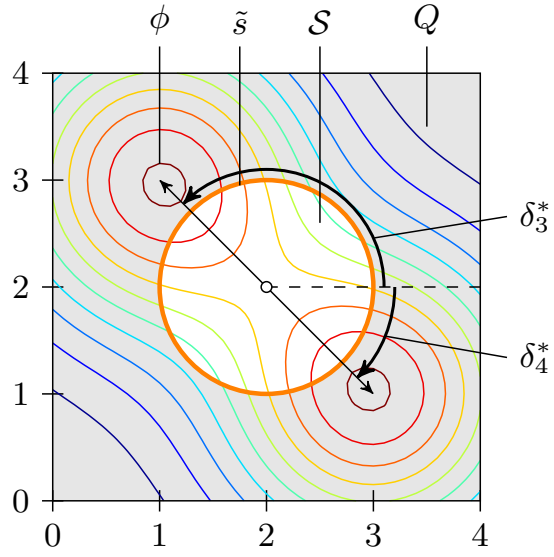


(b) Gaussian for  $\theta = 2$

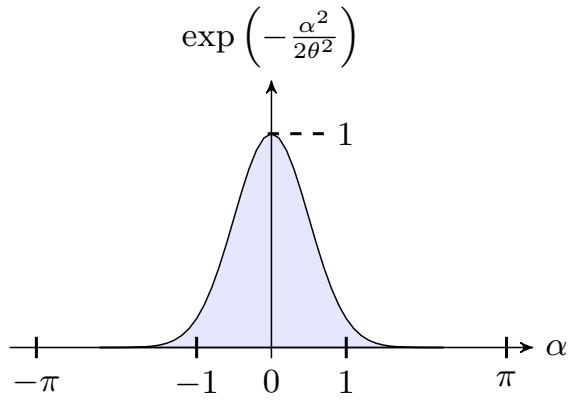


(c)  $\mathcal{H}_{\text{orient}}$  in dependence of  $\delta$  for  $\theta = 2$

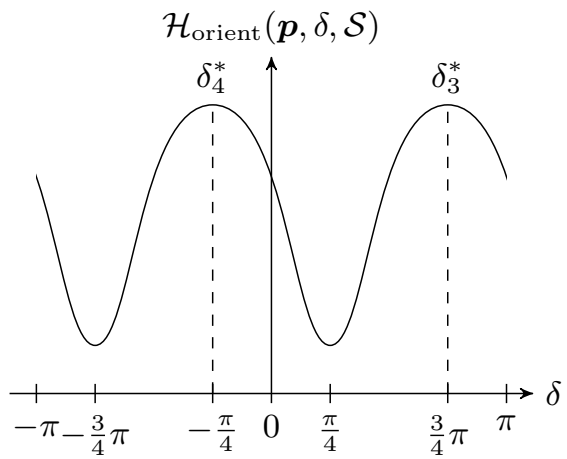
**Figure 4.17:** Orientation-based DisCoverage with large opening angle



(a) scene showing optimal  $\delta$  for  $\theta = 0.5$



(b) Gaussian for  $\theta = 0.5$



(c)  $\mathcal{H}_{\text{orient}}$  in dependence of  $\delta$  for  $\theta = \frac{1}{2}$

**Figure 4.18:** Orientation-based DisCoverage with small opening angle

an opening angle of  $\theta = 2$  results in optimal orientations that do not point to the desired directions as clearly visible in Figure 4.17. Consequently, optimizing (4.39) with too large opening angles  $\theta$  leads to a compromise in finding the optimal orientation  $\delta^*$ , which could result in an orientation pointing to already explored space in the worst case.

Decreasing opening angles such as  $\theta = 0.5$  lead to a slim Gaussian with a distinct peak in  $\alpha = 0$ . According to the objective function  $\mathcal{H}_{\text{orient}}$  in Figure 4.18(c) the optimal orientations point into directions where the weights are maximal. This reflects the desired behavior as each robot is supposed to first move to unknown space with higher density  $\phi$  during the exploration process.

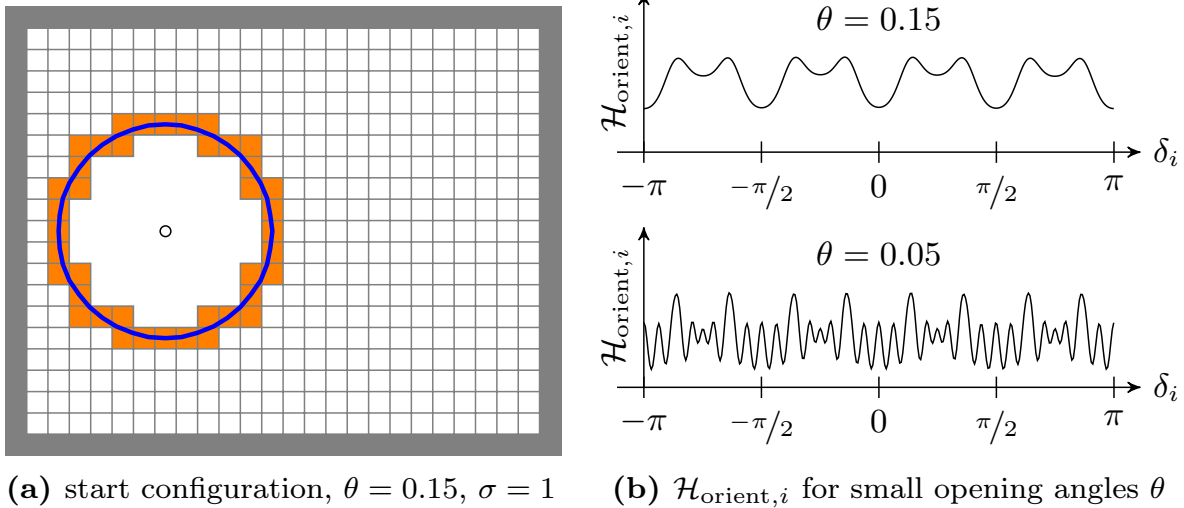
Figure 4.19 shows an example for the opening angles  $\theta = 0.15$  and  $\theta = 0.05$ . From this scene it can be observed that too small opening angles  $\theta$  lead to lots of local maxima in the objective function  $\mathcal{H}_{\text{orient},i}$ . As plotted in Figure 4.19(b), this is especially an issue when using a grid map representation of the environment. A grid map consists of discrete cells, meaning that the frontier is defined as a finite amount of points  $\mathbf{q} \in \tilde{s}$ . In Figure 4.19(a) these points  $\mathbf{q}$  are denoted by the center of the frontier cells. Modeling the frontier as a finite set, the integral over the frontier in the objective function reduces to a sum over discrete points, and therewith one obtains the objective function

$$\mathcal{H}_{\text{orient},i}(\mathbf{p}_i, \delta_i, \mathcal{S}_i) = \sum_{\mathbf{q} \in \tilde{s}_i} f(\mathbf{p}_i, \delta_i, \mathbf{q}) \phi(\mathbf{q}) \quad (4.45)$$

for each robot  $i = 1, \dots, N$ . Evaluating (4.45), each point  $\mathbf{q} \in \tilde{s}_i$  tends to facilitate a peak in the objective function if  $\theta$  is smaller than the grid resolution (cf. Figure 4.19(b)). This is an undesired effect, since the goal is to rather obtain only few local maxima that imply a high information gain in terms of lots of frontier cells. Additionally, very small values  $\theta$  are numerically unstable and cannot be sufficiently represented by the floating point arithmetic.

*Remark 4.13* (Relation to the MinDist strategy). With the Kronecker delta [35]

$$\delta_x^\otimes = \begin{cases} 1, & \text{if } x = 0, \\ 0, & \text{else} \end{cases} \quad (4.46)$$



**Figure 4.19:** Example of  $\mathcal{H}_{\text{orient},i}$  in a grid map with cell resolution  $0.2 \text{ m} \times 0.2 \text{ m}$  and varying opening angle  $\theta$ . In this case, the frontier  $\tilde{s}_i$  describes a finite set.

and the finite set  $\mathcal{M}_{\delta_i} = \{\mathbf{q} \in \tilde{s}_i \mid \alpha(\mathbf{p}_i, \delta_i, \mathbf{q}) = 0\}$ , the discrete valued objective function (4.45) in the limit  $\theta \rightarrow 0$  results in

$$\begin{aligned} \mathcal{H}_{\text{orient},i}(\mathbf{p}_i, \delta_i, \mathcal{S}_i) &= \sum_{\mathbf{q} \in \tilde{s}_i} \delta_{\alpha(\mathbf{p}_i, \delta_i, \mathbf{q})}^{\otimes} \exp\left(-\frac{\|\mathbf{q} - \mathbf{p}_i\|^2}{2\sigma^2}\right) \phi(\mathbf{q}) \\ &= \sum_{\mathbf{q} \in \mathcal{M}_{\delta_i}} \exp\left(-\frac{\|\mathbf{q} - \mathbf{p}_i\|^2}{2\sigma^2}\right) \phi(\mathbf{q}). \end{aligned} \quad (4.47)$$

Consequently, each robot moves into the direction  $\delta_i$  where the distance component of all frontier cells is maximal. Although not equivalent, this behavior is similar to the MinDist strategy.

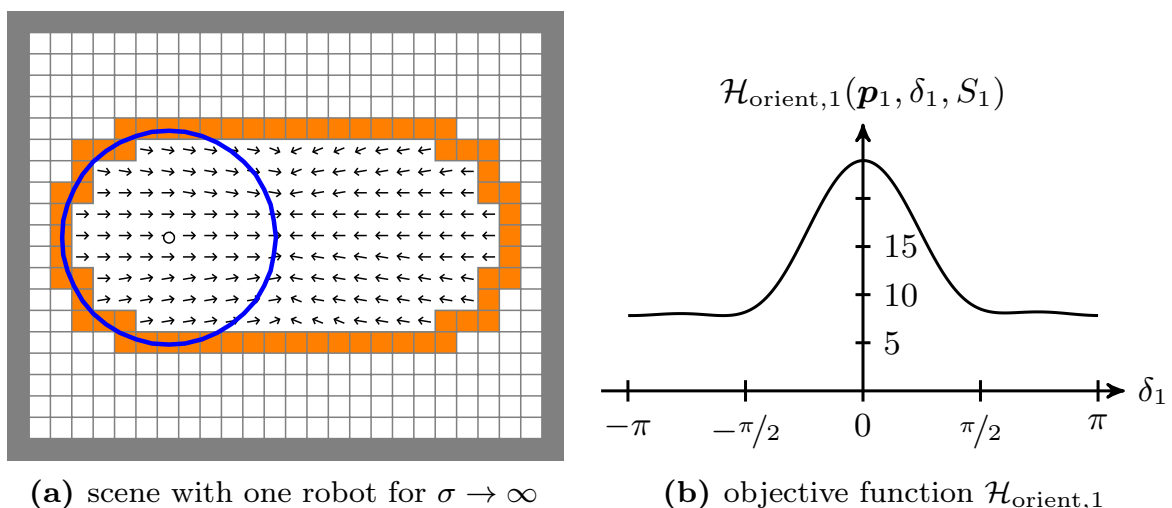
*Remark 4.14* (Optimal choice of the opening angle). As noted in Haumann et al. [4], a formal rule for choosing the opening angle  $\theta$  in the angular component is an open issue, and therefore still subject to research. Nevertheless, simulations in Haumann et al. [4] and experimental results in Haumann et al. [6] validate an opening angle of  $\theta = 0.5$ .

*Remark 4.15* (Choice of the density function). The density function in Figures 4.17 and 4.18 is illustrated by the contour lines. For the rest of this dissertation, the density function is modeled as a constant  $\phi(\mathbf{q}) = 1$ , since no a priori knowledge is assumed during the exploration process. Hence, in contrast to the centroidal search-based DisCoverage approach, the density function only plays a minor role in the orientation-based DisCoverage approach.



### 4.3.4 Impact of the Distance Component

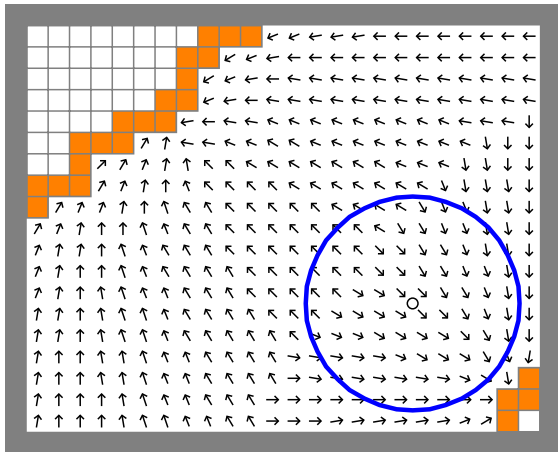
Next, the impact of the distance component is investigated. Therefore, the distance component is first omitted in the performance function (4.41). Omitting the distance component equals building the limit of the standard deviation  $\sigma \rightarrow \infty$ . Without a distance component, each robot optimizes the objective function  $\mathcal{H}_{\text{orient},i}$  such that as many frontier points as possible are located in the direction of the robot's orientation, *independent* of the distance to the frontier. The effect of this setting is depicted in Figure 4.20. Therein, a single robot is located in a rectangular scene in Figure 4.20(a). The vector field indicates that all trajectories end in a separatrix located



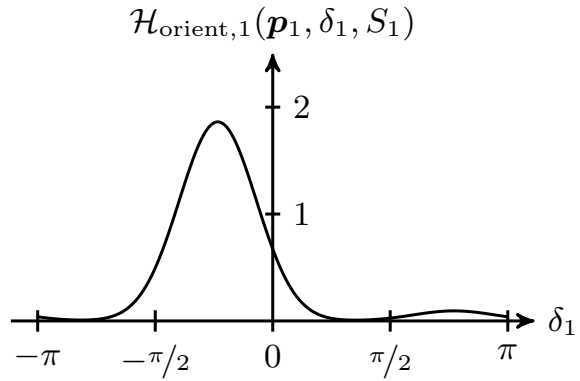
**Figure 4.20:** Optimal orientation without distance component

in the middle of the scene. The corresponding objective function for the robot is plotted in Figure 4.20(b), showing only one maximum at  $\delta_1 = 0$ . From this observation, it can be deduced that too large values for the standard deviation  $\sigma$  of the distance component yield invariant sets that do not facilitate exploration. Further, this observation underlines that the distance component is needed in order to favor locations near the frontier.

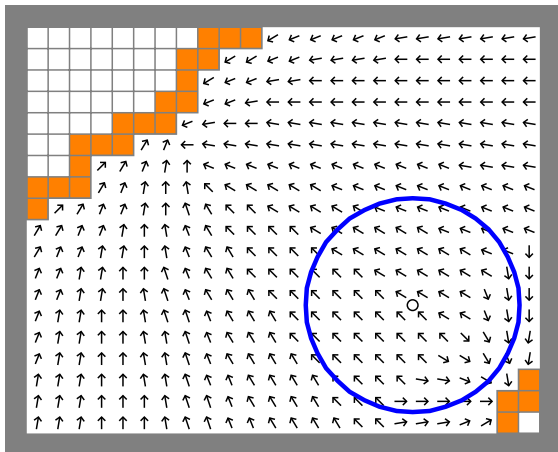
Therefore, the orientation-based DisCoverage approach is investigated next for several standard deviations as depicted in Figure 4.21. Therein, large parts  $\mathcal{S}_1$  of the environment  $\mathcal{Q}$  are already explored. The standard deviation varies from  $\sigma = 1$  in Figure 4.21(a)–4.21(b) over  $\sigma = 2$  in Figure 4.21(c)–4.21(d) to  $\sigma = 3$  in Figure 4.21(e)–4.21(f). Clearly, the standard deviation  $\sigma$  has an impact on the vector field and consequently influences the trajectories of the robots during the exploration process. Larger values for  $\sigma$  imply higher weights even for points on the frontier



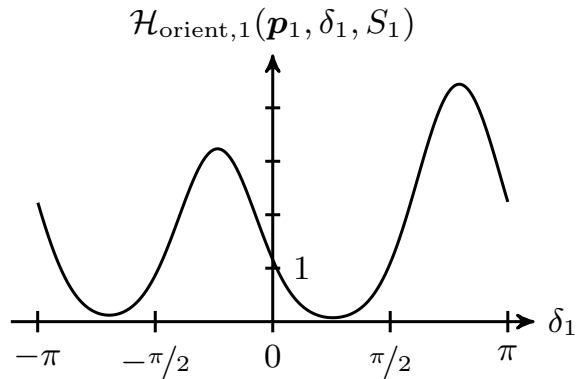
(a) exploration scene with  $\sigma = 1$



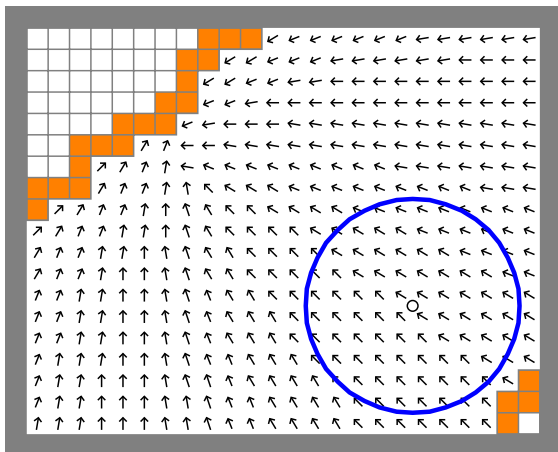
(b) objective function  $\mathcal{H}_{\text{orient},1}$  for  $\sigma = 1$



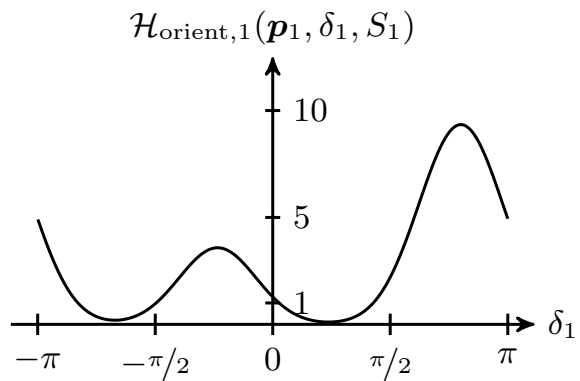
(c) exploration scene with  $\sigma = 2$



(d) objective function  $\mathcal{H}_{\text{orient},1}$  for  $\sigma = 2$



(e) exploration scene with  $\sigma = 3$



(f) objective function  $\mathcal{H}_{\text{orient},1}$  for  $\sigma = 3$

**Figure 4.21:** Impact of the distance component for a scene with one robot

located further away from the robot. Therefore, the standard deviation  $\sigma$  controls whether the robot should focus on frontier points that are close (cf. Figure 4.21(a)–4.21(b)) or on frontier points that are further away *if* the extent of the frontier is larger (cf. Figure 4.21(e)–4.21(f)). In essence, this equals a tradeoff between the distance to reach the frontier and the expected information gain on the frontier.

*Remark 4.16* (Optimal choice of the distance component). As noted in Haumann et al. [4], a formal rule for choosing the standard deviation  $\sigma$  in the distance component is an open issue, and therefore still subject to research. Nevertheless, simulations in Haumann et al. [4] and experimental results in Haumann et al. [6] validate a standard deviation of  $\sigma = 2$ .

*Remark 4.17* (Following local maxima). Considering Figure 4.21(e) the more efficient exploration approach is to first explore the bottom right corner before focusing on the area in the top left. This can be achieved by following the local maximum  $\delta_1 = -\pi/4$  in Figure 4.21(f). Following the local maximum usually has the advantage that the greedy behavior of  $\mathcal{H}_{\text{orient}}$  is slightly suppressed in favor of exploring remaining parts on the frontier before focusing on larger parts on the frontier. This situation is discussed again in Section 4.3.6.

### 4.3.5 Handling Nonconvex Environments

The objective function  $\mathcal{H}_{\text{orient}}$  in (4.39) (page 78) of the orientation-based DisCoverage approach 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 frontier points  $\mathbf{q} \in \tilde{\mathcal{S}}$  with respect to the robot position in the computation, which is a property that always holds in convex environments. Contrary, in nonconvex allowable environments with obstacles, the boundary  $\partial\mathcal{Q}$  as well as the pose and shape of the obstacles have to be taken into account in order to find suitable orientations  $\delta_i$ . Because of that, a transformation of the environment is introduced next as in Haumann et al. [6] such that the objective function

$$\mathcal{H}_{\text{orient}}(\mathcal{P}, \Delta, \mathcal{S}) = \sum_{i=1}^N \int_{\tilde{\mathcal{S}}_i} f(\mathbf{p}_i, \delta_i, \mathbf{q}) \phi(\mathbf{q}) d\mathbf{q}, \quad (4.48)$$

of the orientation-based DisCoverage can be used without modifications.

### Transformation to star-shaped domains

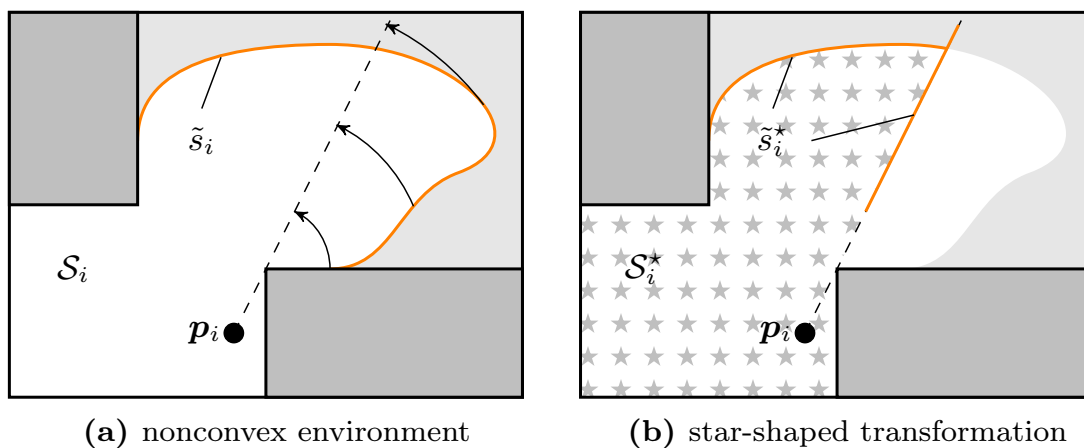
The idea is to find a transformation with the property that all points  $\mathbf{q}$  on the frontier are visible from the robot position  $\mathbf{p}_i$ . This property holds for star-shaped domains. Therefore, a transformation is introduced next, that maps any connected domain to a star-shaped domain.

**Definition 4.1.** Let  $\mathcal{S}_i \subseteq \mathcal{Q} \subset \mathbb{R}^2$  be a connected set and  $\mathcal{S}_i^* \subset \mathbb{R}^2$  a star-shaped set. Let  $d_g(\mathbf{p}_i, \mathbf{q})$  be the geodesic distance from the robot position  $\mathbf{p}_i$  to  $\mathbf{q} \in \mathcal{S}_i$ . Let  $\mathbf{e}(\mathbf{p}_i, \mathbf{q})$  be the unit vector pointing into the direction of the first path segment of the geodesic path from  $\mathbf{p}_i$  to  $\mathbf{q}$ . Then, the transformation is given by  $\mathbb{T}_{\mathbf{p}_i} : \mathcal{S}_i \rightarrow \mathcal{S}_i^*$ ,  $\mathbf{q} \mapsto \mathbf{q}^* = \mathbb{T}_{\mathbf{p}_i}(\mathbf{q}) = \mathbf{p}_i + d_g(\mathbf{p}_i, \mathbf{q})\mathbf{e}(\mathbf{p}_i, \mathbf{q})$ .

**Theorem 4.11.** The map  $\mathbb{T}_{\mathbf{p}_i}$  transforms any connected set to a star-shaped domain with respect to  $\mathbf{p}_i$ .

*Proof.* All points  $\mathbf{q} \in \mathcal{S}_i$  visible from  $\mathbf{p}_i$  remain unchanged and, thus, are invariant with respect to the map  $\mathbb{T}_{\mathbf{p}_i}$ . Points not visible from  $\mathbf{p}_i$  are mapped by  $\mathbb{T}_{\mathbf{p}_i}$ . This applies also to all points on the geodesic path from  $\mathbf{p}_i$  to a specific  $\mathbf{q}$ . Therefore, the resulting set  $\mathcal{S}_i^*$  is connected. As a consequence of  $\mathbb{T}_{\mathbf{p}_i}$ , all points are visible from  $\mathbf{p}_i$ . Hence, the set  $\mathcal{S}_i^*$  is star-shaped with respect to  $\mathbf{p}_i$ . This concludes the proof.  $\square$

The map  $\mathbb{T}_{\mathbf{p}_i}$  can be interpreted as a straightening of the geodesic path. Note, that  $\mathbb{T}_{\mathbf{p}_i}$  is not bijective, i.e., it may map arbitrary many elements from  $\mathcal{S}_i$  to only one element in  $\mathcal{S}_i^*$ . A geometric interpretation of the map  $\mathbb{T}_{\mathbf{p}_i}$  is given in Figure 4.22.



**Figure 4.22:** Transformation to star-shaped domains (cf. Haumann et al. [6])

A significant property of  $\mathbb{T}_{\mathbf{p}_i}$  is that elements  $\mathbf{q} \in \mathcal{S}_i$  visible from  $\mathbf{p}_i$  remain unchanged, i.e., if the closed segment  $[\mathbf{p}_i, \mathbf{q}] \subseteq \mathcal{S}_i \Rightarrow \mathbb{T}_{\mathbf{p}_i}(\mathbf{q}) = \mathbf{q}$ . This is due to the fact that the geodesic path reduces to the Euclidean path in convex environments. This implies that convex environments reflect a sub-domain of nonconvex environments and that they are always star-shaped for any point within the convex region.

Further, the star-shaped environment  $\mathcal{S}_i^*$  (Figure 4.22(b)) is not necessarily a subset of the original environment  $\mathcal{S}_i$  (Figure 4.22(a)), since there exists no upper bound for the length of the geodesic distance in arbitrary nonconvex environments. In fact, it may even be the case that the transformation  $\mathbb{T}_{\mathbf{p}_i}$  results in star-shaped regions  $\mathcal{S}_i^*$  that are not a subset of the environment  $\mathcal{Q}$ , i.e.,  $\mathcal{S}_i^* \subseteq \mathcal{Q}$  does not necessarily hold.

*Remark 4.18* (Related work). It turns out that the proposed transformation realizes the same idea that was presented by Pimenta et al. [103] in order to extend the solution to the coverage problem by Cortés et al. [50] to nonconvex environments. However, the formulation as a transformation  $\mathbb{T}_{\mathbf{p}_i}$  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 nonconvex environments.

### Orientation-based DisCoverage for nonconvex environments

In line with the solution to the coverage problem and the DisCoverage approach based on the centroidal search, the orientation-based DisCoverage approach first partitions the allowable environment  $\mathcal{Q}$  into  $N$  geodesic Voronoi cells (cf. Section 2.1.4, page 12). Communicating neighbor positions, each robot calculates its geodesic Voronoi cell. Using the superscript  $\star$  to indicate the objective function in star-shaped regions, the distributed optimization problem for each robot  $i$  is then given by

$$\begin{aligned} \mathcal{H}_{\text{orient}}^*(\mathcal{P}, \Delta, \mathcal{S}) &= \sum_{i=1}^N \mathcal{H}_{\text{orient},i}^*(\mathbf{p}_i, \delta_i, \mathcal{S}_i) \\ &= \sum_{i=1}^N \int_{\tilde{\mathcal{S}}_i} f(\mathbf{p}_i, \delta_i, \mathbf{q}^*) \phi(\mathbf{q}) d\mathbf{q} \end{aligned} \quad (4.49)$$

with  $\mathbf{q}^* = \mathbb{T}_{\mathbf{p}_i}(\mathbf{q})$  as in Definition 4.1. Compared to (4.48), points  $\mathbf{q}$  on the frontier are replaced by  $\mathbf{q}^*$  in the performance function in (4.49).

---

**Algorithm 2** Continuous-time Orientation-based DisCoverage Loop
 

---

 Initialization of  $\mathbf{p}_i(0)$  and  $\mathcal{S}_i(0)$  for  $i = 1, \dots, N$ 
**while**  $\mathcal{S}(t) \neq \mathcal{Q}$  **do**
**for**  $i = 1, \dots, N$  in parallel **do**

 Partition: communicate  $\mathbf{p}_i$  to Voronoi neighbors  $j$   
 compute  $\mathcal{V}_i$  and  $\tilde{s}_i$ 

 Motion control: compute  $\delta_i^* = \arg_{\delta_i} \max \mathcal{H}_{\text{orient},i}^*(\mathbf{p}_i, \delta_i, \mathcal{S}_i)$ 

$$\dot{\mathbf{p}}_i = v \begin{pmatrix} \cos \delta_i^* \\ \sin \delta_i^* \end{pmatrix}, \quad v \in \mathbb{R}_{\geq 0}$$

 communicate  $\delta \mathcal{V}_{i \rightarrow j}$  to respective neighbor  $j$ 

 Exploration: map  $\mathcal{V}_{i,r}^*$ 
**end for**
**end while**


---

The robot dynamics (4.44) and the continuous computation of optimal orientations (4.42) remain unchanged. Through the transformation  $\mathbb{T}_{\mathbf{p}_i}$  the optimization of the objective function now includes the path planning by calculating the geodesic distance.

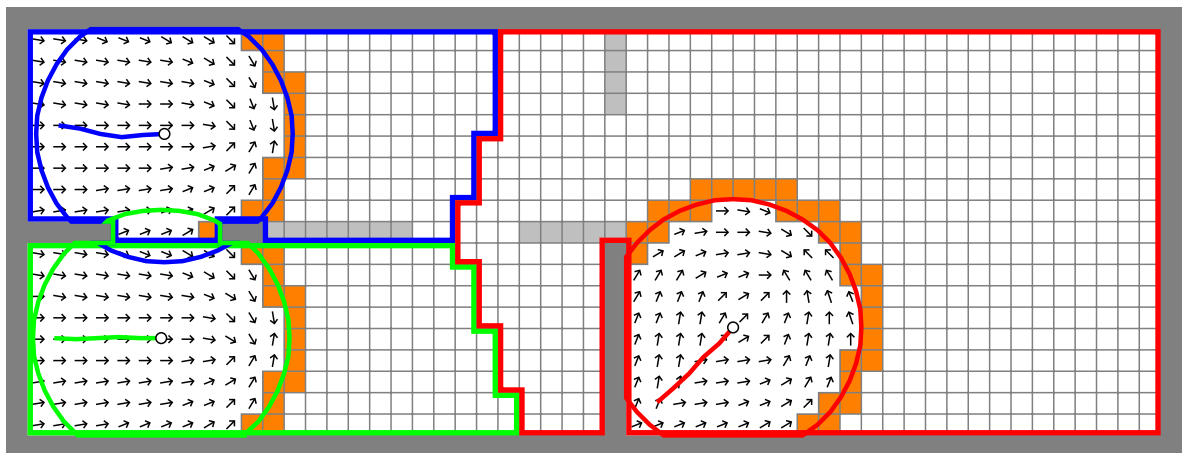
### 4.3.6 Behavior of Multiple Robots

Analog to Section 4.2.5 (page 73) it is of interest how the orientation-based DisCoverage approach behaves when using multiple robots in a nonconvex environment. To this end, each robot continuously calculates its geodesic Voronoi cell and optimizes its orientation-based DisCoverage objective

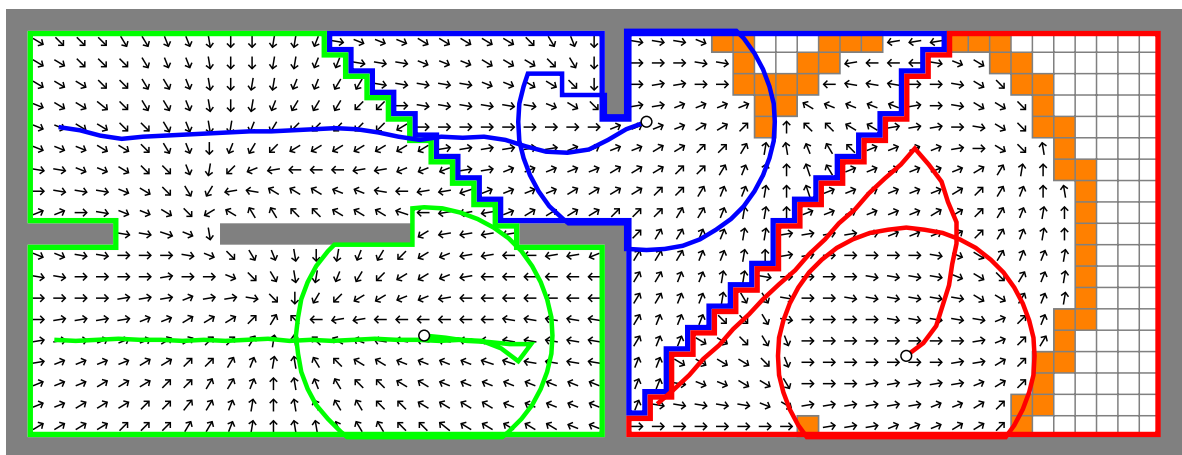
$$\mathcal{H}_{\text{orient},i}^*(\mathbf{p}_i, \delta_i, \mathcal{S}_i) = \int_{\tilde{s}_i} f(\mathbf{p}_i, \delta_i, \mathbf{q}^*) \phi(\mathbf{q}) d\mathbf{q}. \quad (4.50)$$

Therein,  $\mathbf{q}^*$  is computed by applying the map  $\mathbb{T}_{\mathbf{p}_i}$  to points  $\mathbf{q}$  on the frontier  $\tilde{s}_i$  in the nonconvex environment  $\mathcal{Q}$ . The resulting algorithm is given in Algorithm 2. In contrast to Algorithm 1, the motion control law is computed by optimizing the orientation to obtain an optimal  $\delta_i^*$ . Further, the dynamics follow the simplified unicycle dynamics depending on a velocity  $v \in \mathbb{R}_{\geq 0}$ , and the communication of  $\delta \mathcal{V}_{i \rightarrow j}$  equals the exchanged map data as described in Figure 4.1 (page 46).

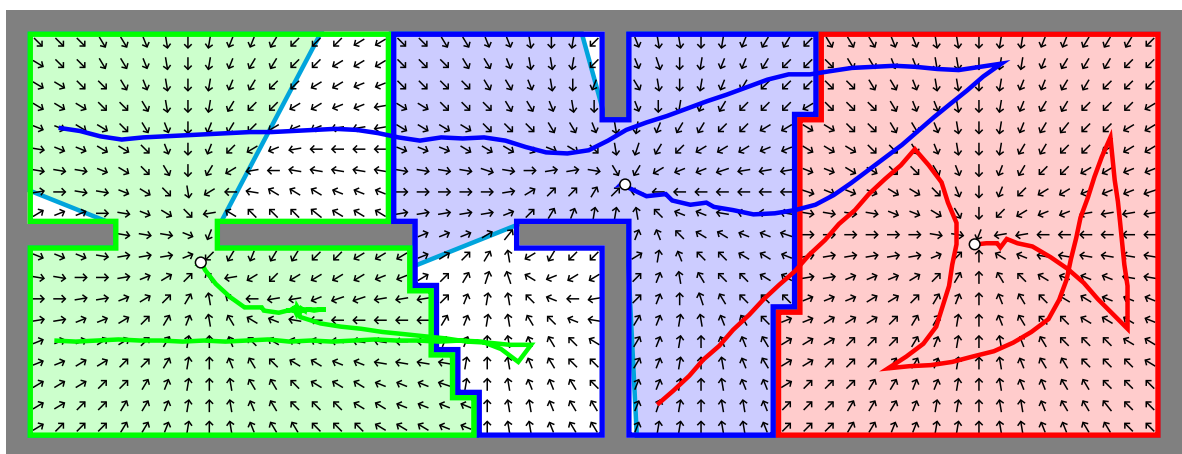
Algorithm 2 is applied to explore a nonconvex allowable environment  $\mathcal{Q}$  with  $N = 3$  robots as shown in Figure 4.23. Figure 4.23(a) shows the scene after 5 iterations. In addition to the trajectories, the sensing range, and the



(a) robot configuration after 5 iterations



(b) robot configuration after 28 iterations

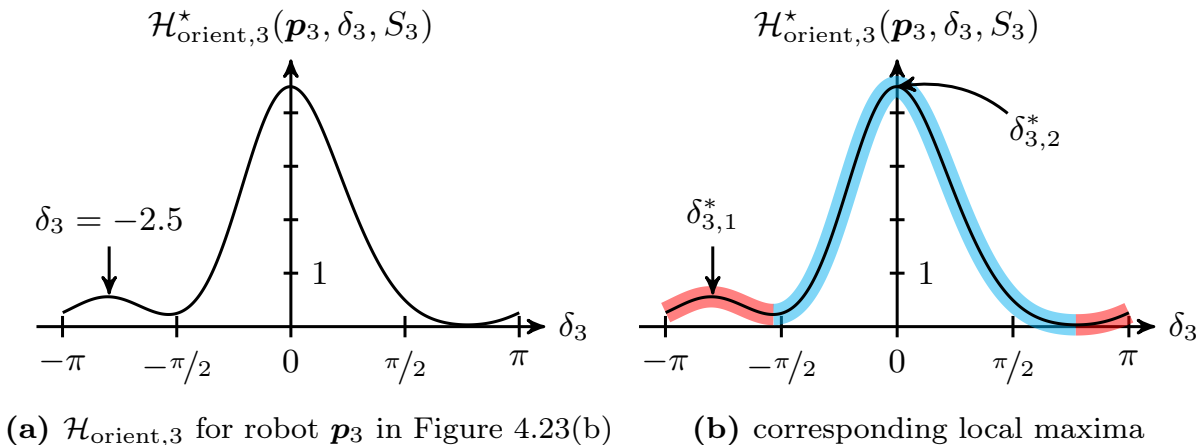


(c) centroidal Voronoi configuration after 75 iterations

**Figure 4.23:** Orientation-based DisCoverage for  $N = 3$  robots in a nonconvex environment. The opening angle is set to  $\theta = 0.5$ . The robots follow local maxima and the distance is automatically adjusted to the nearest frontier cell.

geodesic Voronoi cells, the figure shows the vector field. It can be observed that all gradients point toward the frontier. However, approaching the frontier, the gradients tend to evolve parallel to the frontier. This behavior is by design, since the optimization yields orientations such that as many points  $\mathbf{q}$  on the frontier as possible are located directly in the direction of the orientation.

Figure 4.23(b) shows the scene after 28 iterations. Similar to the simulation in Figure 4.14 (page 75) the robot in  $\mathbf{p}_1$  (green) does not have any frontiers in its Voronoi cell. Therefore, the robot applies the centroidal search within its unlimited visibility set  $\mathcal{V}_i^{*\varepsilon}$ . Further, the vector field in the Voronoi cell of the robot in  $\mathbf{p}_3$  (red) points to the frontier on the right. However, as mentioned in Remark 4.17 (page 87), in this case it would be of advantage if the robot first explored the remaining frontier cell at the bottom. The objective function of this situation is plotted in Figure 4.24. Therein,  $\mathcal{H}_{\text{orient},3}^*$  has two maxima, one in  $\delta_{3,1}^* = -2.5$  pointing to the bottom left, and another one in  $\delta_{3,2}^* = 0$  pointing to right. The orientation-based DisCoverage exploration strategy is therefore modified such that the optimal orientation  $\delta_i^*$  is chosen by following the gradient  $\frac{\partial \mathcal{H}_{\text{orient},i}^*}{\partial \delta_i}$  based on the current orientation  $\delta_i(t)$ , see Figure 4.24(b), which is referred to as following the local maximum in Remark 4.17. Therefore, the robot  $\mathbf{p}_3$  in Figure 4.23(b) continues following the local maximum in  $\delta_{3,1}^*$  and, hence, explores the remaining frontier cell.



**Figure 4.24:** Following the local maximum is of advantage compared to the greedy behavior when always following the global maximum. Depending on the current value of  $\delta_3(t)$ , the resulting maximum is either  $\delta_{3,1}^*$  or  $\delta_{3,2}^*$ .

The exploration process is complete after 55 iterations. Thereafter, all robots fall back to the centroidal search within the respective unlimited



visibility sets  $\mathcal{V}_i^{*\varepsilon}$ . The centroidal Voronoi configuration is reached after a total of 75 iterations in Figure 4.23(c).

### 4.3.7 Varying the Vehicle Dynamics

Generally, the remarks about extending the vehicle dynamics in Section 4.2.6 (page 76) equally apply to the orientation-based DisCoverage approach. In fact, the single integrator dynamics

$$\dot{\mathbf{p}}_i = \mathbf{u}_i = v \begin{pmatrix} \cos \delta_i^* \\ \sin \delta_i^* \end{pmatrix} \quad (4.51)$$

previously used in (4.44) with the optimal orientation  $\delta_i^*$  denoted by a local maximum in (4.42) is already closely related to the unicycle dynamics

$$\begin{bmatrix} \dot{p}_i^x \\ \dot{p}_i^y \\ \dot{\delta}_i \end{bmatrix} = \begin{bmatrix} \cos \delta_i & 0 \\ \sin \delta_i & 0 \\ 0 & 1 \end{bmatrix} \mathbf{u}_i, \quad \text{with} \quad \mathbf{u}_i = \begin{bmatrix} v_i \\ \omega_i \end{bmatrix}, \quad (4.52)$$

where  $\delta_i$  denotes the orientation of the robot and  $\mathbf{u}_i$  denotes its control input with velocity  $v_i \in \mathbb{R}_{\geq 0}$  and angular velocity  $\omega_i$ . The difference between (4.52) and (4.51) is that the optimal orientation changes continuously over time. Again, following Bullo et al. [36], arbitrary dynamics can be applied as long as all vehicles strictly decrease the objective function over time.

# 5 Simulation Results and Lab Experiments

As noted in Section 3.3.2 (page 44), relatively few results exist that compare different approaches to multi-robot exploration. In fact, the influencing factors on the exploration process are manifold and include the shape of the nonconvex environment, the robotic setup such as the amount of robots in the group, the typically limited communication and sensing capabilities, inaccuracies in locomotion with navigation constraints due to possibly non-holonomic vehicle dynamics, or limited power supply.

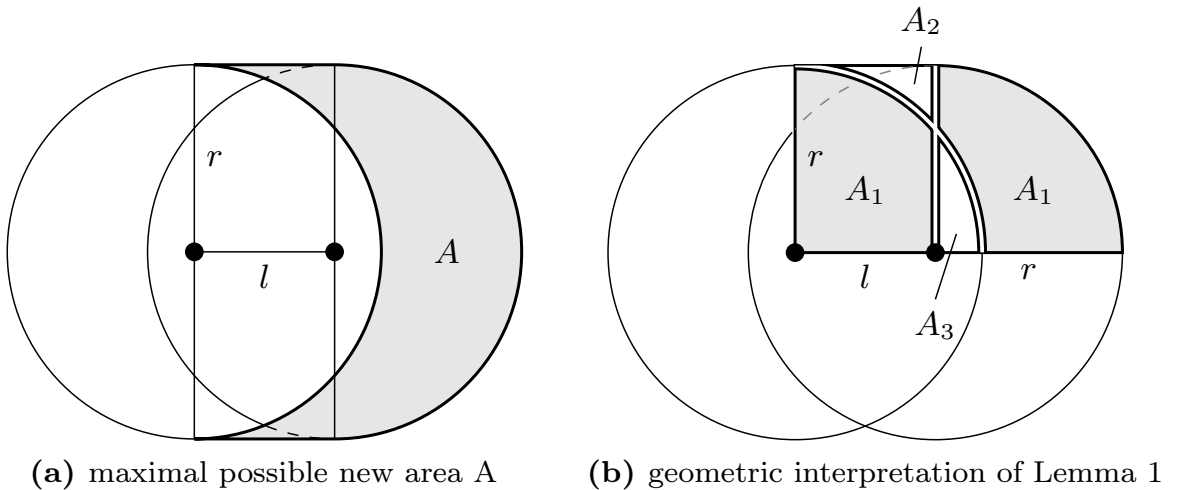
In this chapter, the exploration strategies proposed in the previous chapter as well as the MinDist strategy are compared in terms of the time needed to explore the environment. As published in Frank et al. [2], the time-optimal case acts as reference, as it provides a lower bound for the minimum time needed to explore the entire environment. Given the constraints in real-world environments, the time-optimal case must be considered as a measure of optimality from a purely theoretical perspective. Next to simulations, experiments with e-puck robots are conducted for the orientation-based DisCoverage approach.

## 5.1 The Time-Optimal Case

In order to evaluate the effectiveness of an exploration strategy, it is useful to use the time-optimal exploration strategy as reference. In the time-optimal case, it is assumed that a single robot maps the maximal region it is able to scan within a traveled distance  $l \in \mathbb{R}_{\geq 0}$ . As in the proposed DisCoverage approaches, it is assumed that a robot uses a  $360^\circ$  scanner to map the environment within a given vision radius  $r \in \mathbb{R}$  as illustrated in Figure 5.1(a). Then, the following lemma holds.

**Lemma 5.1.** Given a robot  $i$  at position  $\mathbf{p}_i \in \mathbb{R}^2$  with  $360^\circ$  sensing capabilities limited by the vision radius  $r \in \mathbb{R}_{\geq 0}$ , a maximal area of

$$A = 2rl \tag{5.1}$$



**Figure 5.1:** Time-optimal exploration with vision radius  $r$  and traveled distance  $l$  as published in Frank et al. [2]

can be explored by the robot for a traveled distance  $l$ .

*Proof.* The case  $l \geq r$  is trivial, since the newly explored area equals the rectangular region  $A = 2rl$ . Hence, let  $l < r$ . According to Figure 5.1(a) the half  $\frac{A}{2}$  of the maximal area  $A$  to be explored can be divided in two sub-areas  $A_1$  and  $A_2$

$$\frac{1}{2}A = A_1 + A_2. \quad (5.2)$$

The area  $A_1$  is present two times in Figure 5.1(b), because it is given as  $A_1 = \frac{1}{4}r^2\pi - A_3$ . Therefore,  $A_2$  is given by the rectangular area  $rl$  and  $A_1$  and therewith is denoted as

$$A_2 = rl - A_1. \quad (5.3)$$

Inserting (5.3) into (5.2) leads to the maximal area  $A = 2rl$ , which is equal to the case of  $l \geq r$ .  $\square$

Given an environment  $\mathcal{Q}$  and a discrete-time implementation of the exploration process, and denoting with  $v \in \mathbb{R}_{\geq 0}$  the traveled distance  $l$  per iteration, the lower bound  $\#it$  of iterations needed to explore the entire environment in the single-robot case is therefore given by

$$\#it = \frac{|\mathcal{Q}|}{2rv}, \quad (5.4)$$

where  $|\mathcal{Q}|$  denotes the area of  $\mathcal{Q}$ . In the multi-robot case, it is assumed that all robots explore according to (5.1). Hence, as mentioned in Haumann et al.

[4], the minimal amount of iterations  $\#it$  for  $N$  robots jointly exploring the environment is bounded by the lower bound

$$\#it = \frac{1}{N} \frac{|\mathcal{Q}|}{2rv}. \quad (5.5)$$

To summarize, the amount of iterations given in (5.5) provides a theoretical lower bound for the iterations needed to explore the entire environment  $\mathcal{Q}$  with  $N$  robots.

## 5.2 Discrete-Time Motion Control Laws

All simulations are performed in the *DisCoverage Multi-Robot Exploration Framework* [5]. This software provides a flexible framework for simulating multi-robot exploration strategies in discrete-time. The possibly nonconvex allowable environment  $\mathcal{Q}$  is represented in terms of a grid map [58, 97] consisting of quadratic cells. Each cell represents an area of  $0.2\text{ m} \times 0.2\text{ m}$  and is either *unexplored*, *occupied* by obstacles or the boundary of the environment, or *free*. A cell on the boundary  $\partial\mathcal{Q}$  of the environment switches its state from *unexplored* to *obstacle*, as soon as the cell intersects with a circle around the robot positions  $\mathbf{p}_i$  defined by the sensing range  $r$ . As mentioned in Section 4.2.1 (page 47), a cell in the interior of  $\mathcal{Q}$  switches its state from *unexplored* to *free*, if it is fully contained in one of the sensing circles. Unexplored cells adjoining explored cells further define the frontier. These cells are referred to as *frontier cells* in the following. Since the amount of cells in the grid map is finite, all sets including the set  $\mathcal{Q}$ , the Voronoi cell  $\mathcal{V}_i$  as well as the frontier sets  $\tilde{s}$  and  $\tilde{s}_i$  are *finite* sets, and the integrals in the respective optimization problems reduce to sums. The time is measured in iterations. In each iteration, a robot moves in the direction according to its control law according to its velocity  $v_i \in \mathbb{R}_{\geq 0}$ . Since the moving direction is value-continuous, the robot positions  $\mathbf{p}_i \in \mathbb{R}^2$ ,  $i = 1, \dots, N$ , are also value-continuous. The maximum velocity is bounded by  $0.2\text{ m}$  per iteration and therewith matches the cell resolution.

### 5.2.1 Multi-Robot Exploration Strategies

After splitting the optimization problem among the robots based on the Voronoi partition, the following exploration strategies are used for comparison. Each strategy is explained in short together with its spatially distributed objective function. Due to the discrete-time implementation,

the time is measured with the discrete index  $k$ . For instance, the robot position is referred to as  $\mathbf{p}_i^k$  instead of  $\mathbf{p}_i(t)$ . Similarly, the objective functions are also formulated based on the finite frontier sets.

**MinDist:** Following this strategy, each robot seeks the closest frontier cell in its Voronoi cell using the geodesic distance. The approach solely relies on the distance costs to determine where to move next. It is a standard approach in robotics motion planning and control [82] and the objective function together with the optimization is given by:

objective function	$\mathcal{H}_{\text{MinDist},i}(\mathbf{p}_i^k, \mathbf{q}) = d_g(\mathbf{p}_i^k, \mathbf{q})$
optimization step	$\mathbf{q}^* = \arg \min_{\mathbf{q} \in \tilde{\mathcal{S}}_i^k} \mathcal{H}_{\text{MinDist},i}(\mathbf{p}_i^k, \mathbf{q})$
motion control	$\mathbf{p}_i^{k+1} = \mathbf{p}_i^k + v_i \mathbf{e}(\mathbf{p}_i^k, \mathbf{q}^*)$

Therein,  $\mathbf{e}(\mathbf{p}_i, \mathbf{q})$  denotes the unit vector along the first path segment of the geodesic path from  $\mathbf{p}_i$  to  $\mathbf{q}$ .

**DisCoverage based on the centroidal search:** This strategy was introduced in Section 4.2 (page 46). Essentially, the objective function consists of distance costs of points within the integration range  $\bar{r}$  to the frontier. The moving direction is obtained by computing the centroid of the  $\bar{r}$ -limited visibility set  $\mathcal{V}_{i,\bar{r}}^{\star\epsilon}$  in the  $\epsilon$ -contraction  $\mathcal{Q}_\epsilon$ . The objective function, the optimization step and the motion controls in discrete time are given by:

objective function	$\mathcal{H}_{\text{discover},i}^*(\mathbf{p}_i^k, \mathcal{S}_i^k) = \sum_{\mathbf{q} \in \mathcal{V}_{i,\bar{r}}^{\star\epsilon}} \ \mathbf{p}_i^k - \mathbf{q}\ ^2 \phi(\mathbf{q}, \tilde{\mathcal{S}}_i^k)$
optimization step	$\mathbf{u}_i^k = \mathbf{m}_\phi(\mathcal{V}_{i,\bar{r}}^{\star\epsilon}) - \mathbf{p}_i^k$
motion control	$\mathbf{p}_i^{k+1} = \mathbf{p}_i^k + v_i \frac{\mathbf{u}_i^k}{\ \mathbf{u}_i^k\ }$

**Orientation-based DisCoverage:** In the orientation-based DisCoverage strategy explained in detail in Section 4.3 (page 77), each robot optimizes its orientation  $\delta_i$  to find a suitable exploration direction. The objective function, combining distance costs and the expected information gain, and the optimization problem are stated as:

objective function	$\mathcal{H}_{\text{orient},i}^*(\mathbf{p}_i^k, \delta_i^k, \mathcal{S}_i^k) = \sum_{\mathbf{q} \in \tilde{s}_i^k} \exp\left(-\frac{\alpha^2(\mathbf{p}_i^k, \delta_i^k, \mathbf{q}^*)}{2\theta^2} - \frac{d_g^2(\mathbf{p}_i^k, \mathbf{q}^*)}{2\sigma^2}\right)$
optimization step	$\delta_i^* = \arg \max_{\delta_i \in [-\pi, \pi]} \mathcal{H}_{\text{orient},i}^*(\mathbf{p}_i^k, \delta_i^k, \mathcal{S}_i^k)$
motion control	$\mathbf{p}_i^{k+1} = \mathbf{p}_i^k + v_i \begin{bmatrix} \cos \delta_i^* \\ \sin \delta_i^* \end{bmatrix}$

Therein, the angle  $\alpha(\mathbf{p}_i, \delta_i, \mathbf{q}^*)$  is denoted as in (4.40) and  $\mathbf{q}^* = \mathbb{T}_{\mathbf{p}_i}(\mathbf{q})$  as in Definition 4.1 (page 88). Further, the density  $\phi$  is equal to one as noted in Remark 4.15 (page 84).

### 5.2.2 Fallback Strategy

All approaches use the (geodesic) Voronoi partition to assign a region of dominance to each robot. Whenever the Voronoi cell  $\mathcal{V}_i$  of robot  $i$  is fully explored, it falls back to the centroidal search in the unlimited visibility set  $\mathcal{V}_i^{*\varepsilon}$  in order to maximize coverage in  $\mathcal{V}_i^{*\varepsilon}$  to position itself optimally with respect to the distance to all points in  $\mathcal{V}_i^{*\varepsilon}$ . Assuming a constant density function  $\phi(\mathbf{q}) = 1$ , one obtains

objective function	$\mathcal{H}_{\text{cover},i}^*(\mathbf{p}_i, \mathcal{V}_i^{*\varepsilon}) = \sum_{\mathbf{q} \in \mathcal{V}_i^{*\varepsilon}} \ \mathbf{q} - \mathbf{p}_i\ ^2$
optimization step	$\mathbf{u}_i^k = \mathbf{m}_\phi(\mathcal{V}_i^{*\varepsilon}) - \mathbf{p}_i^k, \quad \text{s.t.} \quad \ \mathbf{u}_i^k\  \leq 1$
motion control	$\mathbf{p}_i^{k+1} = \mathbf{p}_i^k + v_i \mathbf{u}_i^k$

## 5.3 Environmental and Robotic Setup

The simulation of the multi-robot exploration strategies is performed in two allowable environments of size  $15\text{ m} \times 10\text{ m}$  as depicted in Figure 5.2. The environment is represented in terms of an occupancy grid map [58, 97]. The resolution of each grid cell is set to  $0.2\text{ m} \times 0.2\text{ m}$ . The environment in Figure 5.2(a) is defined by a convex rectangular region. Contrary, Figure 5.2(b) shows a nonconvex indoor environment with four distinct rooms, which are separated from each other by 3 meter wide doorways. All simulations are performed with  $N = 3$  robots, all having a sensing range of  $r = 2\text{ m}$ . Throughout the simulation, all robots move with a constant velocity of  $v_i = 0.2\frac{\text{m}}{\text{s}}$ . As an exception, whenever a robot is unemployed, it applies the unlimited centroidal search as fallback strategy and velocities smaller than  $v_i = 0.2\frac{\text{m}}{\text{s}}$  are allowed in order to reach the centroid.

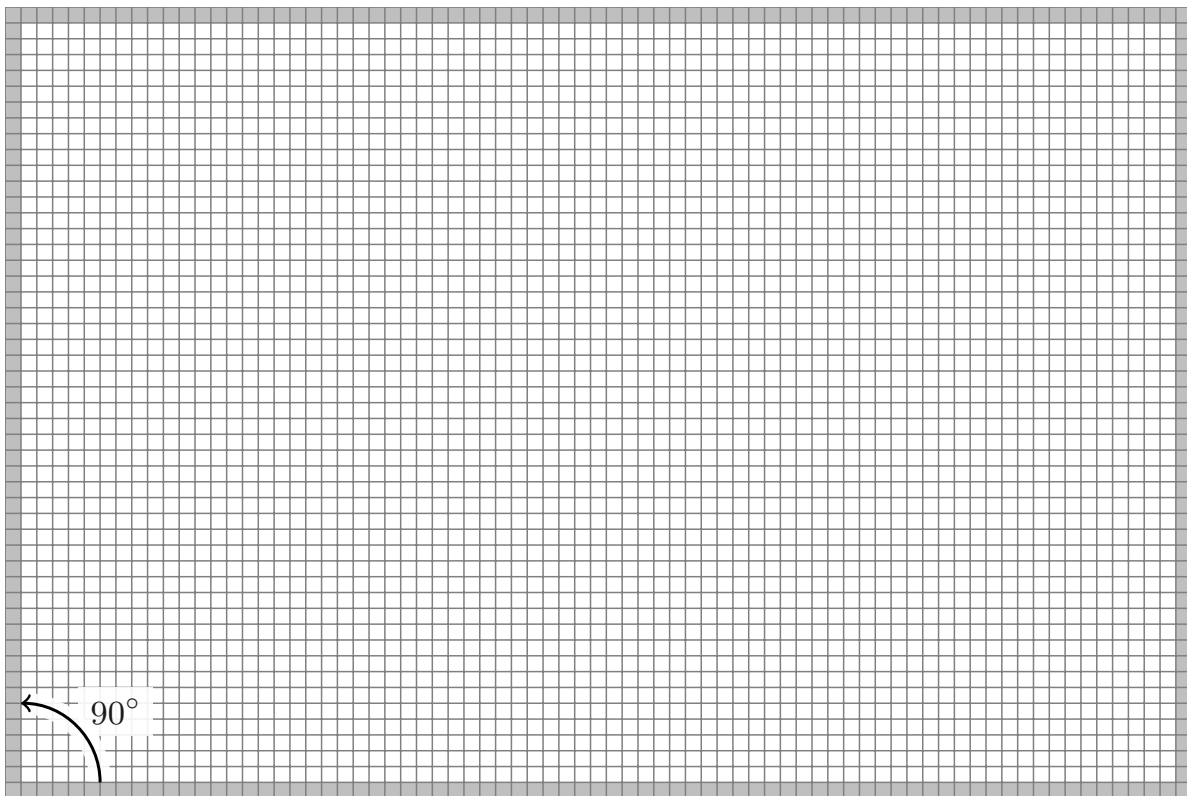
For the centroidal search-based DisCoverage approach (cf. Section 4.2, page 46), an integration range of  $\bar{r} = 0.5\text{ m}$  is used. Since the smallest interior angle is  $\alpha = 90^\circ$  in both the convex and the nonconvex environment (cf. Figure 5.2(a) and 5.2(b)), the sensing range  $r$  satisfies

$$r = 2\text{ m} \geq r_{\min} = \frac{\bar{r}}{\sin(\frac{\alpha}{2N})} \approx 1.93\text{ m} \quad (5.6)$$

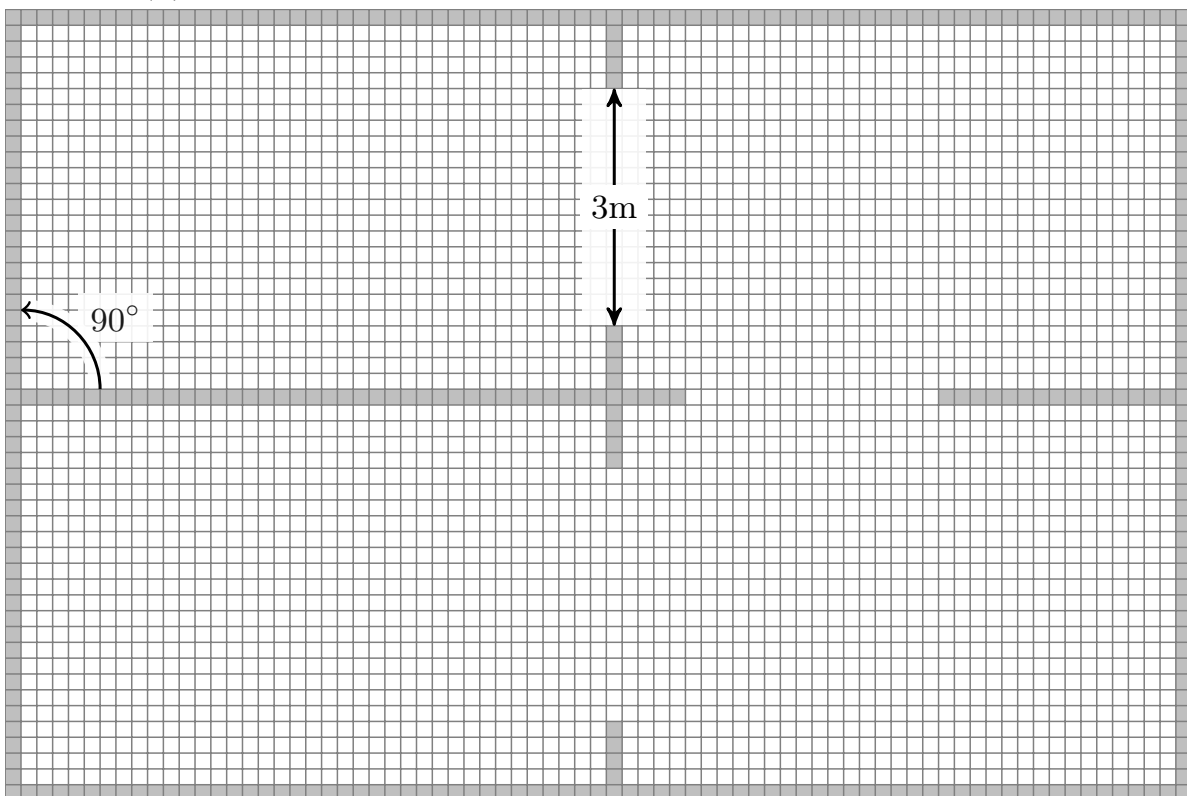
according to the minimum sensing range (4.10). In the nonconvex environment, the  $\varepsilon$ -contraction first needs to be applied as discussed in Section 4.2.4. Strictly speaking, without the  $\varepsilon$ -contraction collisions with the environment are possible on concave vertices that are not continuously differentiable along the boundary  $\partial\mathcal{Q}$ . However, in practice the  $\varepsilon$ -contraction can be neglected for the following reasons: First, moving on an unstable separatrix that points into a concave vertex is a special case that appears only very rarely in very specific setups. Second, as implementation detail, the simulator prevents the robots from moving into the boundary  $\partial\mathcal{Q}$  by adding an arbitrarily small perturbation to the robot position in this case, that moves the robot away from the boundary. Therefore, the minimum sensing range in the nonconvex environment is also defined by (5.6). Setting  $\varepsilon = 0$  in (4.30) further simplifies the computation of the maximum integration range to

$$\bar{r}_{\max} = \frac{d_{\text{pass}} - 2\varepsilon}{2N} = \frac{3\text{ m}}{2 \cdot 3} = 0.5\text{ m}. \quad (5.7)$$

Obviously, the chosen integration range  $\bar{r} = 0.5\text{ m} \leq \bar{r}_{\max}$  satisfies (5.7), and therewith all robots are able to simultaneously pass the doorways.



(a) Convex allowable environment, grid resolution:  $0.2\text{ m} \times 0.2\text{ m}$



(b) Nonconvex allowable indoor environment, grid resolution:  $0.2\text{ m} \times 0.2\text{ m}$

**Figure 5.2:** Applied simulation environments of size  $15\text{ m} \times 10\text{ m}$



For the orientation-based DisCoverage approach (cf. Section 4.3, page 77), the opening angle of all robots is set to  $\theta = 0.5$  (cf. Remark 4.14, page 84) and the distance component matches the sensing range with  $\sigma = 2\text{ m}$  (cf. Remark 4.16, page 87).

## 5.4 Statistical Evaluation

The MinDist-approach, the centroidal search-based DisCoverage approach, and the orientation-based DisCoverage approach are all simulated in both the convex and nonconvex allowable environments in Figure 5.2. In order to obtain statistically significant results, each of the 6 combinations consists of 1000 simulation runs. Each simulation run starts with distinct, randomly generated initial robot positions and an unexplored map. The time of a simulation run is measured in the number of iterations needed to fully explore the map. In each iteration, the exploration progress is measured in percent. A value of 100% implies a fully explored environment. Further, in each iteration the *unemployment* is measured in percent. As discussed previously, a robot is said to be unemployed, if no frontier cells are in its Voronoi cell. Consequently for  $N = 3$  robots, the unemployment in each iteration is either 0%,  $33.\bar{3}\%$  or  $66.\bar{6}\%$ . A value of 100% unemployment is excluded, since 100% resembles a fully explored environment. In this case, the unemployment is set to 0% as the robots are open for new tasks.

Statistical evaluations of 1000 simulation runs in the convex and the nonconvex allowable environment in Figure 5.2(a) are given in Figure 5.3 and in Figure 5.5, respectively, for all three exploration strategies. In each diagram, the number of iterations is displayed on the horizontal axis. The vertical axis shows the exploration progress in percent. A value of 100% implies that all grid cells are explored. Each iteration vertically shows a box plot: the  $\pm 25\%$  band around the median represents 500 of the 1000 simulation runs. In addition, the ‘best case’ and the ‘worst case’ is shown in each iteration of the 1000 runs. The best case equals the maximum of the explored space in percent in the respective iteration. Analog to the best case, the ‘worst case’ shows the minimum of the explored space in percent in the respective iteration. Consequently, all 1000 simulation runs always lie within the interval defined by the worst and the best case. In addition to the box plot, the mean and standard deviation of the iterations needed to explore 90%, 95%, 98% and 100% are depicted horizontally ( $\text{---}\circ\text{---}$ ) in terms of a Gaussian distribution for all 1000 simulation runs. Further, the theoretical optimum is displayed as straight line as reference to the

time-optimal case. As discussed in Section 5.1 (page 94), the time-optimal case defines a lower bound of the iterations required to explore the entire environment.

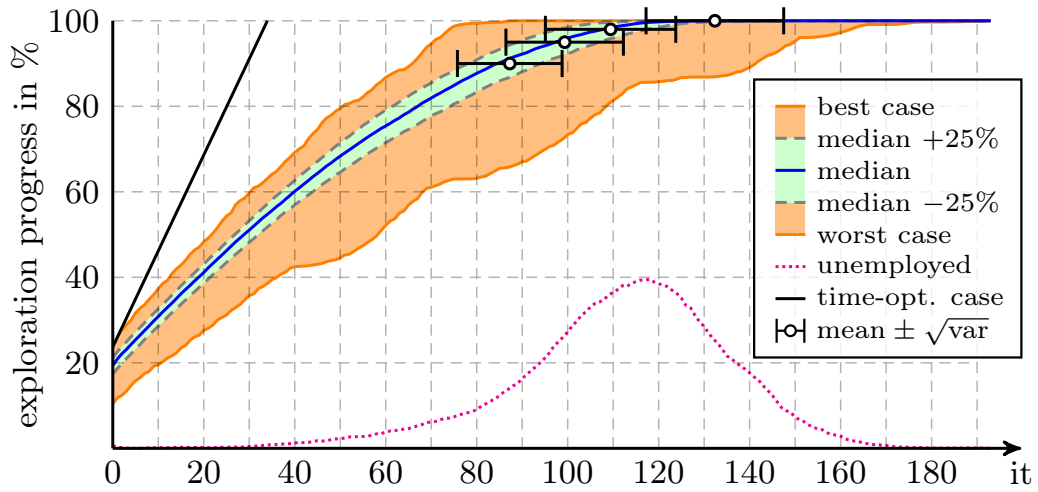
Finally, the unemployment is plotted over time. Since the value is averaged over 1000 simulation runs, one obtains a smooth curve that grows over time until a peak is reached. Thereafter, the unemployment decreases again, since more and more simulation runs already finished the exploration and therefore the robots are counted as employed as they are open for new tasks.

### 5.4.1 Results in the Convex Environment

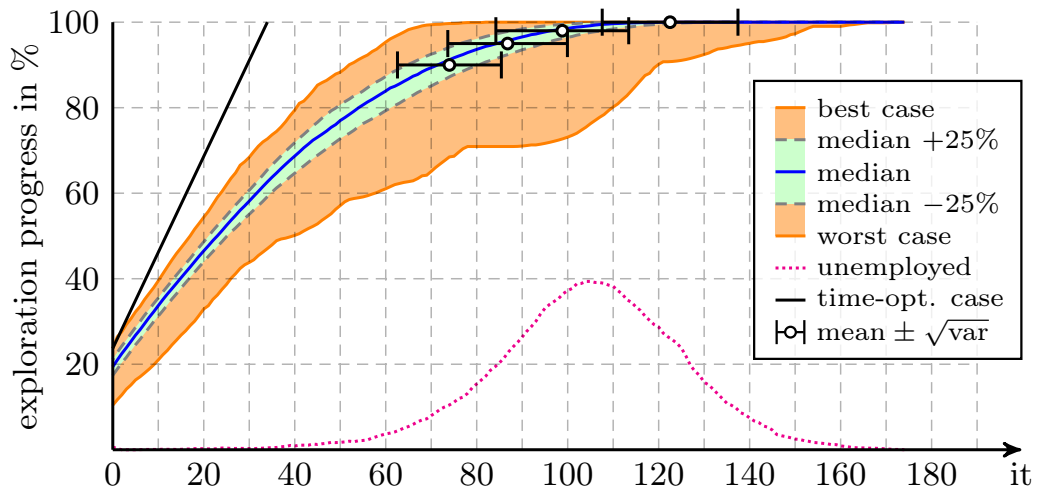
From Figure 5.3, it follows that all exploration strategies always accomplish the exploration task. The width of the 50% band around the median is comparable for all three strategies. However, in the first third of the iterations, the slope of the centroidal search-based DisCoverage and the orientation-based DisCoverage approach is higher than the slope of the MinDist approach. This, for instance, can be observed at iteration 60: While the median of the MinDist approach is at about 75% (cf. Figure 5.3(a)), the DisCoverage-based approaches both already explored about 84% of the environment (cf. Figure 5.3(b) and 5.3(c)).

There are several reasons for this. Following the MinDist approach, each robot moves to the closest point on the frontier. Due to the grid map representation, the frontier is defined in terms of a finite set of frontier cells. Especially at the beginning of the exploration, these frontier cells are typically located on the border of the circle defined by the sensing range around the respective robot position. However, due to the grid map discretization, the distance of distinct frontier cells to the robot position is smaller than the distance of other frontier cells. Therefore, the robots follow preferred orientations. As a result, the robots change the moving direction after each iteration which results in zig-zag trajectories (cf. left column in Figure 5.4). Therewith, the effective distance traveled is shorter compared to moving along a straight line, and consequently the amount of explored area is reduced.

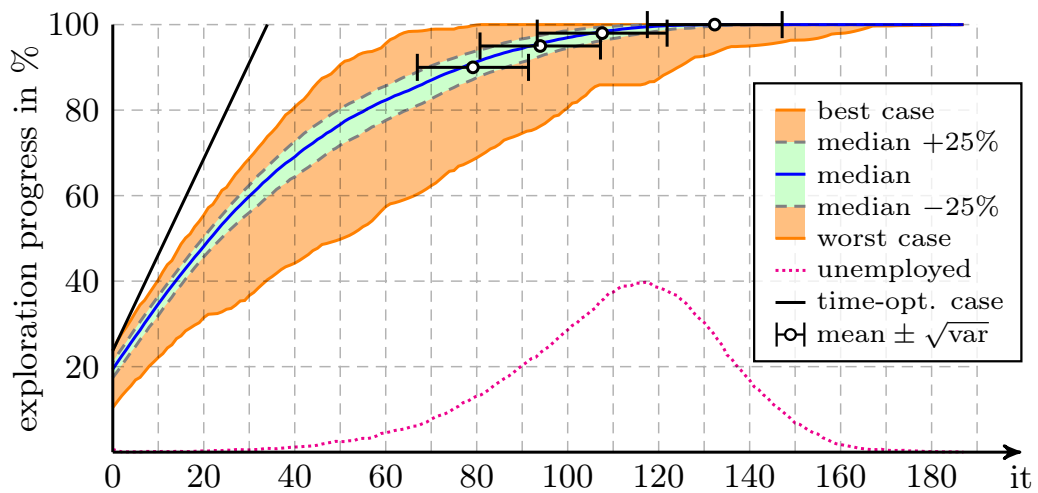
The centroidal search-based approach does not have this disadvantage, since integrating over all points in  $\mathcal{V}_{i,\bar{r}}^{\star\epsilon}$  results in smoothed trajectories (cf. middle column in Figure 5.4). Further, the centroidal search-based approach has the advantage of keeping the safety distance of  $\bar{r} = 0.5$  m to the boundary  $\partial\mathcal{Q}$ . Therewith, the perceived area is possibly larger compared to approaches that allow moving closer along the boundary.



(a) MinDist strategy

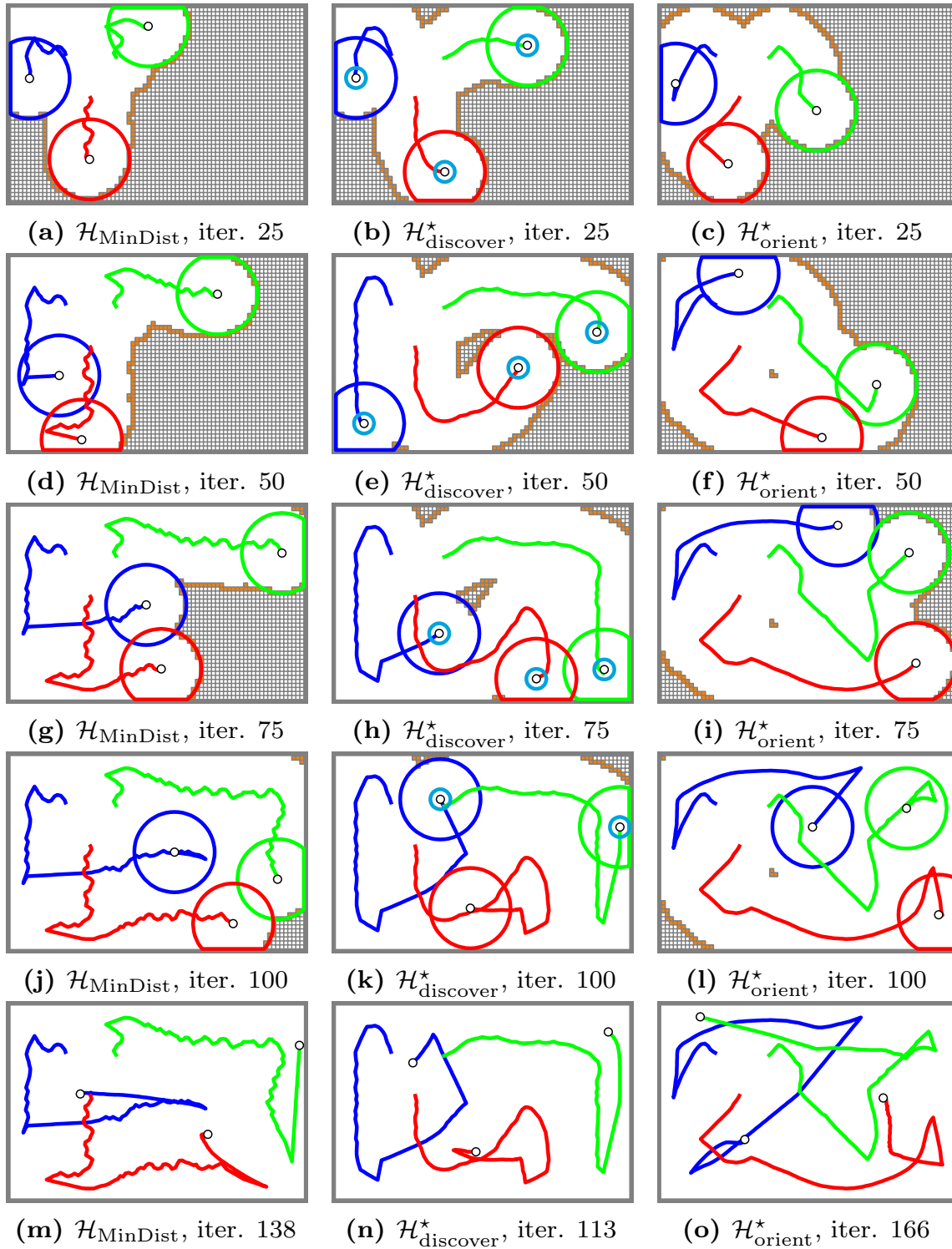


(b) centroidal search-based DisCoverage



(c) orientation-based DisCoverage

**Figure 5.3:** Statistics over 1000 simulation runs in the convex environment



**Figure 5.4:** Exemplary robot trajectories of the first simulation run in the convex environment. The initial robot positions are equal for all three strategies. Left column: MinDist strategy, middle column: centroidal search-based DisCoverage, right column: orientation-based DisCoverage.

**Table 5.1:** Iterations needed to explore 90%, 95%, 98%, and 100% of the convex environment in Figure 5.2(a). The values are given in terms of the mean and the standard deviation ( $\pm$ ) and also visible in Figure 5.3.

	$\mathcal{H}_{\text{MinDist}}$	$\mathcal{H}_{\text{discover}}^*$	$\mathcal{H}_{\text{orient}}^*$
90%	$87.27 \pm 11.83$	$74.00 \pm 11.75$	$79.17 \pm 12.55$
95%	$99.34 \pm 13.24$	$86.79 \pm 13.48$	$93.97 \pm 13.57$
98%	$109.43 \pm 14.66$	$98.81 \pm 14.93$	$107.58 \pm 14.61$
100%	$132.36 \pm 15.46$	$122.51 \pm 15.25$	$132.33 \pm 15.16$

Similarly, following the orientation-based DisCoverage approach, each robot chooses its moving direction by optimizing its orientation such that as many frontier cells as possible are directly located in front of the robot. As a result, the robot always moves into directions with lots of frontier cells, which maximizes the information gain. This effect can be observed in the first 40 iterations in Figure 5.3(c): Here, the slope of the orientation-based DisCoverage approach surpasses the slope of the other two approaches.

The further course of the statistics in Figure 5.3 are best examined by analyzing Table 5.1. Therein, the iterations needed to explore 90%, 95%, 98%, and 100% of the environment are listed in terms of the mean and the standard deviation. The centroidal search-based DisCoverage approach outperforms both the MinDist and the orientation-based DisCoverage approach. The standard deviations in each row are similar. Interestingly, the orientation-based DisCoverage approach outperforms the MinDist approach most of the time. However, towards the end of the exploration process, the mean of the iterations needed to explore 90%, 95%, 98%, and 100% of the environment again converges to the performance of the MinDist approach. This fact can be explained by the greedy behavior of the orientation-based DisCoverage approach. Maximizing the information gain, the robots focus on parts of the environment that are largely unexplored. During this procedure it happens, that small parts of the environment – parts such as a single frontier cell in a corner – remain unexplored (cf. right column in Figure 5.4). Due to these frontier isles, the robots need to travel large distances again before finishing the exploration task. Therefore, the performance of the orientation-based DisCoverage approach degrades towards the end.

Next, the unemployment in Figure 5.3 is discussed. At the very beginning of the exploration process, all robots have frontier cells in the respective Voronoi cells. Therefore, the unemployment initially is at 0%. Over time, the unemployment increases, meaning that some robots do not have any

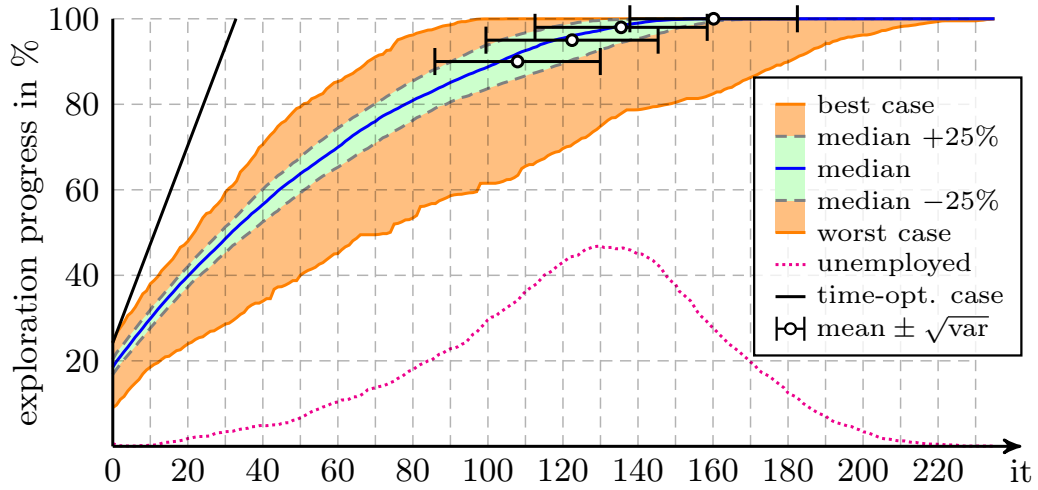
frontier cells in their Voronoi cell. As mentioned before, in this case the robots apply the centroidal search within the unlimited visibility set as fallback strategy. Statistically, the unemployment of all strategies increases further until it reaches its peak of 40% in the interval [98%, 100%]. From these statistics one can deduce that the exploration strategies behave mostly the same in terms of the unemployment.

### 5.4.2 Results in the Nonconvex Allowable Environment

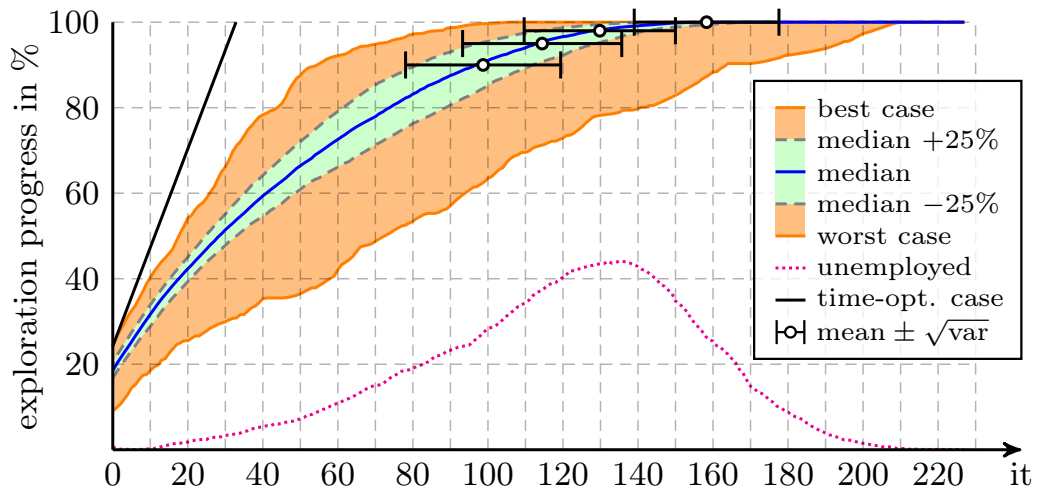
Next, the exploration process in the nonconvex allowable environment in Figure 5.2(b) is examined, analogously to the convex environment. The statistical results are shown in Figure 5.5. Again, all simulation runs accomplish the exploration task. Due to the nonconvexity, the mean traveled distance to explore the entire environment is larger compared to the convex case. This reflects in the increased number of iterations needed to fully explore the environment.

The statistical results in Figure 5.5 are similar to the convex case in Figure 5.3. However, the unemployment differs from the convex: In the convex case, the unemployment increases roughly after iteration 40, while the unemployment in the nonconvex environment increases much earlier already at iteration 20. The reason for this are the restrictions imposed by the nonconvexity, meaning that the geodesic Voronoi cells partition the environment such that robots do not have any assigned frontier cells. The unemployment increases until it reaches its peak at about 98% for the MinDist and the centroidal search-based DisCoverage approach. Interestingly, the peak for the orientation-based DisCoverage approach lies approximately at iteration 137, and therewith reaches its maximum well after the 98% mark. This effect can be explained by investigating Figure 5.6(1). Therein, the robot in the upper half explored almost all parts of the room, but the trajectories are such that several frontier cells close to the boundary remain unexplored. Therefore, the robots often have remaining frontier cells towards the end of the exploration process in the respective Voronoi cell and consequently are counted as employed.

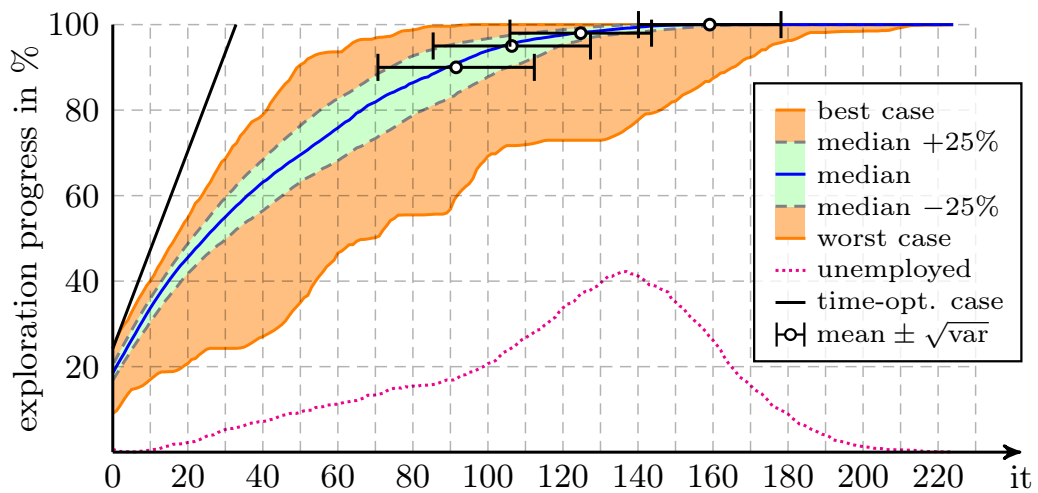
This hypothesis is also backed up by Table 5.2. Here, the iterations needed to explore 90%, 95%, 98%, and 100% of the nonconvex environment are listed in terms of the mean and the standard deviation. While the orientation-based DisCoverage approach in the right column performs best with respect to exploring 90%, 95%, and 98%, the performance again equals the other two approaches when comparing the 100% level. This again



(a) MinDist strategy

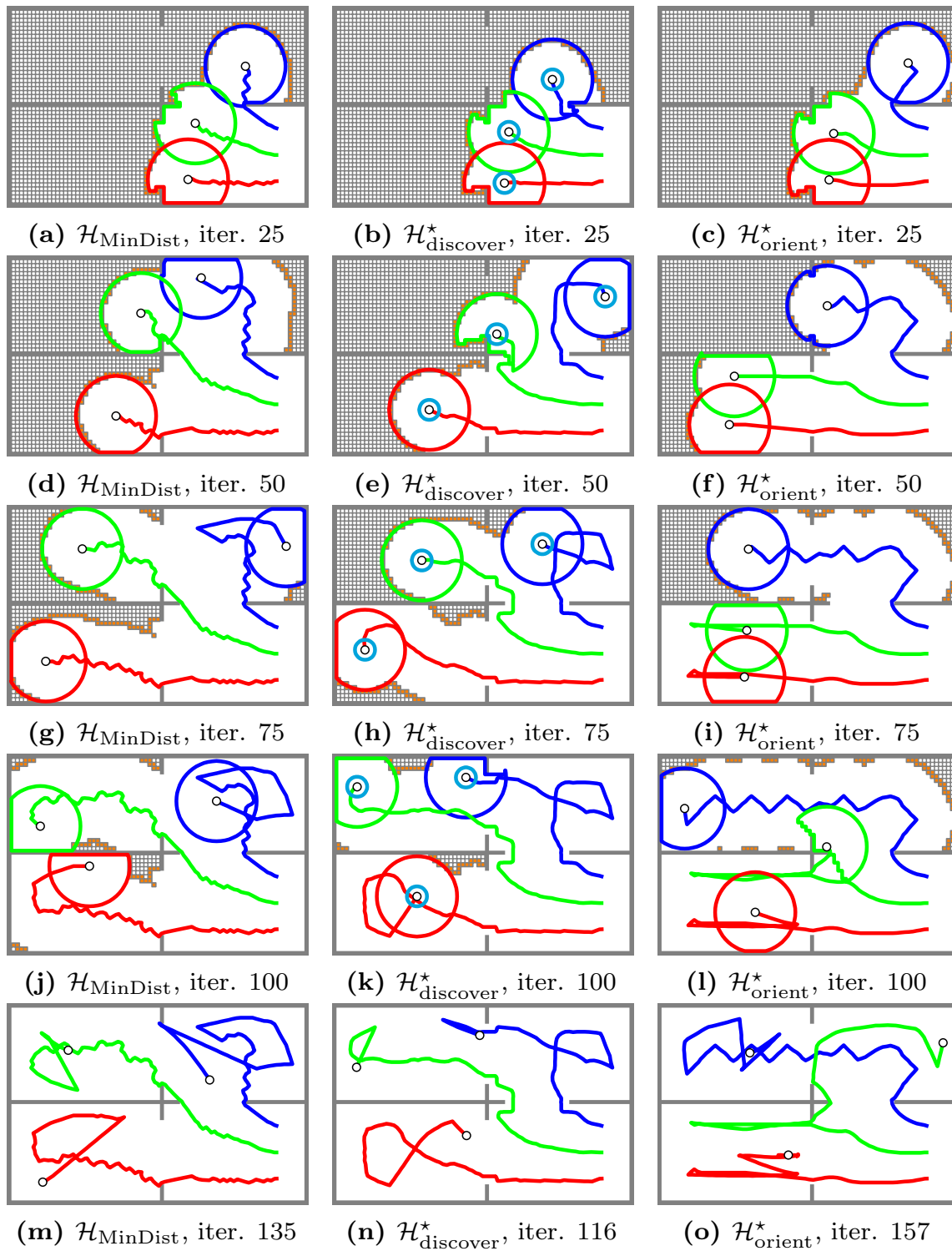


(b) centroidal search-based DisCoverage



(c) orientation-based DisCoverage

**Figure 5.5:** Statistical evaluation of 1000 simulation runs



**Figure 5.6:** Exemplary robot trajectories of the first simulation run in the nonconvex environment. The initial robot positions are equal for all three strategies. Left column: MinDist strategy, middle column: centroidal search-based DisCoverage, right column: orientation-based DisCoverage.



**Table 5.2:** Iterations needed to explore 90%, 95%, 98%, and 100% of the nonconvex environment in Figure 5.2(b). The values are given in terms of the mean and the standard deviation ( $\pm$ ) and also visible in Figure 5.5.

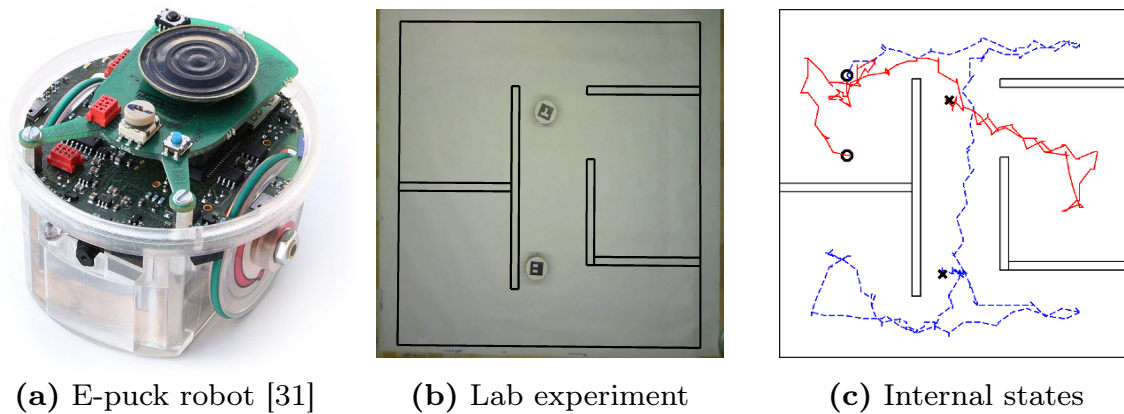
	$\mathcal{H}_{\text{MinDist}}$	$\mathcal{H}_{\text{discover}}^*$	$\mathcal{H}_{\text{orient}}^*$
90%	$107.91 \pm 22.45$	$98.72 \pm 21.07$	$91.53 \pm 21.25$
95%	$122.42 \pm 23.35$	$114.46 \pm 21.61$	$106.37 \pm 21.37$
98%	$135.50 \pm 23.35$	$129.82 \pm 20.56$	$124.74 \pm 19.27$
100%	$160.16 \pm 22.73$	$158.29 \pm 19.69$	$159.13 \pm 19.40$

underlines the greedy behavior maximizing the short term information gain of the orientation-based DisCoverage approach. Next to this observation, the performance of the centroidal search-based DisCoverage approach lies between the performance of the MinDist and the orientation-based DisCoverage approach. Besides this, the standard deviations in each row are approximately the same, meaning that from a statistics point of view all approaches spread equally around the respective expected values.

For completeness, the trajectories of all three exploration strategies in the nonconvex environment are exemplarily shown in Figure 5.6 for the first simulation run.

## 5.5 Experimental Results with E-puck Robots

In addition to the simulation, lab experiments were performed in collaboration of the Technische Universität Darmstadt with ETH Zurich at the Massachusetts Institute of Technology (MIT) in 2010, and published in Haumann et al. [6]. In the experiments, two e-puck robots [95] as depicted in Figure 5.7(a) explore a nonconvex allowable environment  $\mathcal{Q}$  of size  $1 \text{ m} \times 1 \text{ m}$  with the orientation-based DisCoverage approach. In line with the simulation results, the standard deviation of the angular component and the distance component are set to  $\theta = 0.5$  and  $\sigma = 0.2 \text{ m}$ , respectively. The boundary of the environment is represented by the black overlay in Figure 5.7(b). The e-puck robots have a diameter of 7 cm and are equipped with a Bluetooth interface (further specifications are listed in Appendix B.1, page 128). In the setup, each e-puck robot carries a unique marker which is tracked by an overhead camera using the ARToolkit software [72]. Accordingly, the localization of the robots is carried out by the camera and the problem of Simultaneous Localization and Mapping (SLAM) is avoided. The camera is connected to a central controller running in MATLAB<sup>®</sup> in



**Figure 5.7:** Experimental results with two e-puck robots exploring a nonconvex allowable environment

the loop. This central controller keeps track of the explored and unexplored parts of the environment in terms of a grid map with a grid cell resolution of  $0.025\text{ m} \times 0.025\text{ m}$ . Based on this information and the current robot positions obtained from the visual tracker, the controller optimizes the objective  $\mathcal{H}_{\text{orient},i}$  for each of the two e-puck robots. The computed moving directions are then sent back to the robots over Bluetooth in each iteration, and the e-puck robots move into the designated directions.

The implementation of the experiments differs from the simulation as follows: Since the e-puck robots have a finite physical size of diameter 7 cm, the obstacles are inflated by applying the  $\varepsilon$ -contraction in order to avoid collisions with the environment. The inflation may influence the exploration behavior, as the robots approach different sets of frontier cells compared to pure simulation and the distance to the frontier changes when navigating around the enlarged obstacles.

The e-puck 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 trajectories. However, apart from this noise, the orientation-based DisCoverage algorithm appears to be robust in this implementation. During the robot movements, the MATLAB<sup>®</sup> program tracks the explored parts around the e-puck robots by simulating omnidirectional sensing capabilities with a sensing range of  $r = 0.2\text{ m}$ .

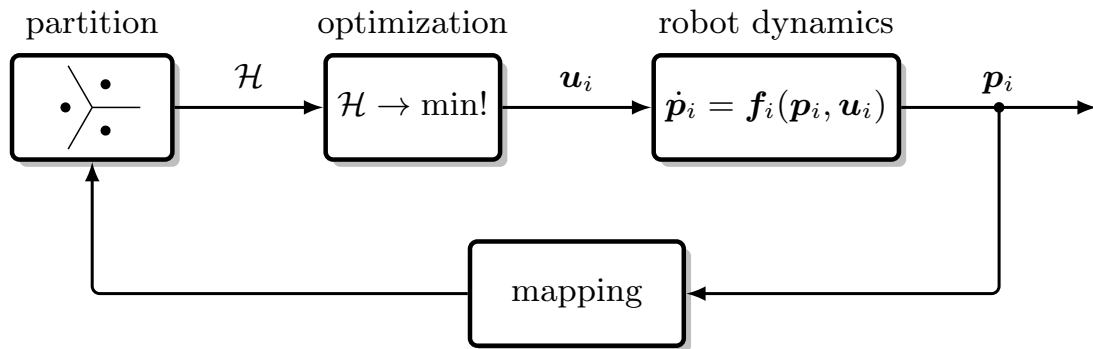
Figure 5.7(b) shows the final configuration. Therein, the robots moved through the environment such that all parts of the environment were visible following the robot trajectories. As fallback strategy, the robots apply the Voronoi coverage algorithm for nonconvex environments proposed in Breitenmoser et al. [34]. The trajectories of the e-puck robots are tracked by the controller and depicted in Figure 5.7(c). The initial positions are marked by the two circles, and the final positions by the crosses.

## 6 Discussion

In Chapter 4, two new approaches to multi-robot exploration were introduced. Both approaches are based on the DisCoverage paradigm, which itself is derived from the solution to the coverage problem discussed in detail by Bullo et al. [36]. The approaches were both first introduced for convex environments in terms of a distributed optimization problem. The distributivity is based on the fact that the computation of the Voronoi partition is spatially distributed over the Delaunay graph. The DisCoverage paradigm forms a closed feedback loop: Based on the partition of the environment, the robots autonomously create and optimize an objective function. The partial derivatives of this objective function with respect to the robot positions and orientations are used as control input for each robot. The robot movements change the partition and the loop begins anew. This process is continuously performed until the entire environment is explored.

Both approaches are extended to the nonconvex domain. As a foundation for the centroidal search-based DisCoverage approach, Assumption 4.2 (page 60) was introduced according to which the visibility sets are quasi-stationary when building the partial derivatives. Further, the  $\varepsilon$ -contraction was applied, which turns concave vertices on the boundary  $\partial\mathcal{Q}$  into continuously differentiable concave curved path segments. These properties were then utilized in  $\mathcal{H}_{\text{discover}}^*$  to derive nonconvex control laws. The orientation-based DisCoverage approach was extended to support nonconvex environments by first applying a transformation  $\mathbb{T}_{\mathbf{p}_i}(\mathbf{q})$  (cf. Definition 4.1, page 88) to each location  $\mathbf{q} \in \mathcal{Q}$  obtaining a star-shaped domain with respect to the robot positions. Therewith, orientations obtained as results of optimizing  $\mathcal{H}_{\text{orient}}^*$  always point into directions in  $\mathcal{Q}$  avoiding collisions with the boundary  $\partial\mathcal{Q}$ .

Figure 1.1 of Chapter 1 is depicted again in Figure 6.1. With respect to the centroidal search-based DisCoverage approach, the optimization block optimizes  $\mathcal{H}$  with respect to the robot positions  $\mathbf{p}_i$ , resulting in distributed motion control laws  $\mathbf{u}_i = -\frac{\partial\mathcal{H}}{\partial\mathbf{p}_i}$ . These in turn are used in the robot dynamics in terms of  $\dot{\mathbf{p}}_i = \mathbf{f}_i(\mathbf{p}_i, \mathbf{u}_i) = \mathbf{u}_i$ . Similarly, with respect to the



**Figure 6.1:** Closed multi-robot exploration feedback loop

orientation-based DisCoverage approach,  $\mathcal{H}$  is optimized with respect to the robot orientations  $\delta_i$ , resulting in distributed motion control laws

$$\dot{\mathbf{p}}_i = \mathbf{u}_i = v \begin{pmatrix} \cos \delta_i^* \\ \sin \delta_i^* \end{pmatrix}. \quad (6.1)$$

Therewith, both DisCoverage approaches follow the closed feedback loop as initially proposed in the problem statement. The design criterion of *distributed coordination* is fulfilled, since all robots autonomously solve the optimization problem and apply distributed motion control laws. Further, the *communication constraints* are defined by the Voronoi partition, which is known to be computable in a distributed manner. This in turn fulfills the problem statement as well as the design criteria mentioned in the introduction (cf. Chapter 1).

## 6.1 Simulation Results and Lab Experiments

As shown in Chapter 5 (page 94 ff.), all exploration strategies accomplish the exploration task, with varying performance. Generally, the performance of the MinDist approach is reduced due to the zig-zag trajectories which are immanent for implementations relying on a discrete set of frontier cells. The centroidal search-based DisCoverage approach also solely relies on computing distances to the frontier, and there is no information gain based component. However, since the distances from all points in the  $\bar{r}$ -limited visibility sets  $\mathcal{V}_{i,\bar{r}}^{\star\epsilon}$  are computed, the resulting trajectories are smooth. Further, due to the integration range  $\bar{r}$ , the boundary  $\partial\mathcal{Q}$  of the environment as well as the boundary of the Voronoi cells act as repulsive force. This repulsive force pushes the robots away from the boundaries up to a certain safety distance depending on the integration range. This effect can

be observed in Figure 5.6 (page 108) in the middle column, and is especially visible for the green robot trajectory in Figure 5.6(h). In fact, smooth trajectories and keeping a safety distance are considerable advantages over the MinDist approach. In comparison, the MinDist approach in the left column in Figure 5.6 often slices the boundary.

The orientation-based DisCoverage approach also slices the boundary, since no components in the objective function pushes the robots away from the boundary. Figure 5.6(1) shows this property for the green robot: The closest frontier cells pull the robots around the nonconvex boundary. In general, a safety distance can strictly be enforced for all exploration strategies by first inflating the environment  $\mathcal{Q}$  by applying the  $\varepsilon$ -contraction to  $\mathcal{Q}$ . This approach is also known as *growing of obstacles* [87, 130] and a well-established strategy for collision-free path planning of robotic manipulators [82].

The lab experiments in Section 5.5 (page 109) show that applying the DisCoverage paradigm in terms of the closed feedback loop successfully accomplishes the exploration task. In this case, two e-puck robots explored a nonconvex environment with the orientation-based DisCoverage approach. Since the e-puck robots have a physical diameter of 7 cm, inflating the boundary of the nonconvex environment  $\mathcal{Q}$  was a hard requirement to avoid collisions.

## 6.2 Performance Analysis

The simulation results as well as the experimental results demonstrate that the proposed multi-robot exploration strategies based on the DisCoverage paradigm successfully solve the exploration task. Although each approach was simulated one thousand times in order to obtain statistically significant results, it needs to be emphasized that neither the simulation nor the lab experiments account for all degrees of freedom that influence real-world exploration tasks. This in turn makes it hard to measure the performance of the approaches, meaning that it is close to impossible to nominate a clear winner among the strategies – neither in this dissertation nor in related publications.

As already noted in the discussion about related work (cf. Section 3.3.2, page 44), the time-optimal case in terms of the theoretical optimum was therefore used as lower bound of the time required to explore the entire environment. As a result, all three exploration strategies perform similarly with slight differences: the performance of the MinDist approach degrades

due to the zig-zag trajectories; the centroidal search-based DisCoverage approach has the advantage of inherently keeping a safety distance, and the orientation-based DisCoverage approach initially performs better due to its greedy behavior.

It should be emphasized, though, that the simulations and experiments were performed to validate the proposed multi-robot exploration approaches, as similarly noted in Frank et al. [2]. With this background in mind, the main goal is not to find the best performing multi-robot exploration strategy. Instead, the main contribution of the simulations and experiments is to validate the idea of transferring the solution to the coverage problem as proposed by Bullo et al. [36] to the multi-robot exploration problem, fulfilling the design criteria in Section 1.2 (page 3).

## 6.3 Convergence Properties

The solution to the coverage problem as proposed by Bullo et al. [36] is provably correct and always finds an optimal solution. The proof is based on the fact that the objective function acts as Lyapunov function, whose gradient is used in the control input. This in turn allows to apply the Krasovskii-LaSalle invariance principle to prove convergence to centroidal Voronoi configurations which maximize coverage.

The centroidal search-based DisCoverage approach closely follows the solution to the coverage problem for convex environments. Under Assumption 4.1 (page 50) a quasi-stationary density function is assumed and therefore the proof of convergence as well as the proof for solving the multi-robot exploration problem are performed analogously to the proof for the solution to the coverage problem. Under Assumption 4.2 (page 60), these proofs are extended to nonconvex domains defined in terms of allowable environments. Essentially, this extension relies on the  $\varepsilon$ -contraction defining the  $\bar{r}$ -limited visibility sets. Based on this extension, the theorems and proofs for convex environments are transferred to nonconvex allowable environments. Although not discussed in detail, it is noteworthy that this solution can readily be applied to solve the nonconvex coverage problem as well, as published in Klodt et al. [12]. Compared to existing solutions to nonconvex coverage, the proposed solution only solves the coverage problem in the (un-)limited visibility set, instead of the nonconvex (un-)limited Voronoi cell. However, this does not impose any restrictions in the context of the multi-robot exploration problem.

Although validated in the simulation and the experiments with the e-puck robots, a formal proof of convergence for the orientation-based DisCoverage approach is an open research question: Currently, the orientation-based DisCoverage approach works successfully with the specified standard deviations in the angular component and the distance component. How to choose these parameters is subject to further research, though.

## 6.4 Optimality of Space Partitions

Robots not having a frontier in their Voronoi cells fall back to the unlimited centroidal search in the respective visibility set in order to maximize coverage within the respective Voronoi cell. In the statistical evaluation, these robots are counted as unemployed, as they effectively do not contribute to the exploration task. In fact, having robots not contributing to the exploration task is the same as mapping the environment with less robots. From this observation, it follows that especially towards the end of the exploration process a more effective coordination of the group is required. However, this coordination cannot be achieved by strictly using the Voronoi partition. This raises an interesting research question: The Voronoi partition is the optimal partition for solving the coverage problem [36]. However, the Voronoi partition is not the optimal partition for the multi-robot exploration task with respect to minimizing the time needed to explore the entire environment. Nevertheless, the Voronoi partition is still a widely used partition of the environment. This is due to the fact that its computation is spatially distributed over the Delaunay graph, which is a property that is of advantage especially in convex environments.

This background in mind, finding an optimal partition suited for distributed multi-robot exploration is an open problem. An optimal partition may decrease the time needed to accomplish the exploration task, since it allows for a better assignment of robot target points.

## 6.5 Higher State Space Dimensions

The proposed exploration strategies were discussed in detail for the two-dimensional case. However, the centroidal search and therewith also the centroidal search-based DisCoverage approach both can be readily applied to higher dimensions without any modifications. This is not the case for the orientation-based DisCoverage approach: Here, the robots derive  $\mathcal{H}_{\text{orient},i}^*$



with respect to the orientations  $\delta_i$ . In higher dimensions, the optimal orientation in the angular component needs to be optimized with respect to each additional dimension. For instance, in the three dimensional space, the optimization must be performed with respect to the horizontal and the vertical direction. Using the spherical coordinate system probably provides a tool to model the optimization procedure in this case.

## 6.6 Map Representation

Throughout the statistical evaluation and the lab experiments an occupancy grid map was used to represent the environment. Occupancy grid maps [58, 97] have the advantage of being easy to implement, since each cell equals one element in a huge  $n$ -dimensional matrix. However, Shen et al. [114] note, that occupancy grid maps do not scale to the three dimensional space or even larger dimensions. This is due to the curse of dimensionality [28], according to which the state space grows exponentially with growing dimension. For instance, increasing memory usage is an issue for high resolution occupancy grid maps. Therefore, it is worth to look into alternative map representations such as the metric representation or topological maps based on graph theory [126]. For instance, Durham et al. [56] recently proposed an approach to solve the coverage problem based on a graph, where each node represents e.g. a room of an indoor environment, and the edges encode information used to navigate among the rooms. In related research, Keidar and Kaminka [74] investigate possibilities of how to efficiently detect the frontier when running experiments.

## 6.7 Discrete-Time Systems

The DisCoverage paradigm and the solution to the coverage problem are both introduced and analyzed in terms of continuous-time dynamical systems: The partition is continuously kept up-to-date, the optimization problems are continuously optimized, the robots move continuously according to the derived control laws, and, finally, all theorems and the respective proofs are performed for the continuous-time case.

In practice, however, the implementation of these strategies is typically performed in discrete time. This also holds for the DisCoverage simulator [5] and the central controller in the experiments with the e-puck robots [6]. Therein, the environment is represented as a (discrete) occupancy grid.

As a consequence, the frontier is defined by a finite set of frontier cells, and the integrals in the optimization problems  $\mathcal{H}_{\text{MinDist}}$ ,  $\mathcal{H}_{\text{discover}}^*$ , and  $\mathcal{H}_{\text{orient}}^*$  reduce to sums over these finite sets (cf. Section 5.2, page 96). Accordingly, the continuous-time motion control laws  $\dot{\mathbf{p}}_i = \mathbf{u}_i$  translate into the discrete-time equations  $\mathbf{p}_i^{k+1} = \mathbf{p}_i^k + \tau \mathbf{u}_i^k$  with sample time  $\tau$ .

Albeit very similar, Cortés et al. [50] and Cortés et al. [47] show that the convergence analysis for discrete-time systems relies on a discrete version of the Krasovskii-LaSalle invariance principle. The authors show that the solution to the coverage problem can be transferred to the discrete-time domain including the proof of convergence to centroidal Voronoi configurations. However, this dissertation extends this theory to nonconvex environments within the (un-)limited visibility sets in continuous time and a value continuous domain. Therefore, strictly speaking, the proof of convergence for the proposed DisCoverage approaches in discrete time is subject to further research.

In fact, explicitly modeling the system in discrete time has several advantages, since the Voronoi partition is only updated at distinct points in time. First, no communication among robot neighbors is required as long as the Voronoi cells remain constant. Second, the computational overhead of computing the possibly geodesic Voronoi cells is vastly reduced. Third, Bullo et al. [36] note that the centroidal search can be applied to vehicles with arbitrary dynamics, *if* the objective function strictly decreases in the time intervals between communication rounds where the Voronoi cells are updated. Following this observation, modeling the exploration process in discrete time has huge potential to extend the proposed DisCoverage methods to complex and more realistic and heterogeneous vehicle dynamics.

## 7 Conclusion

This final chapter reviews the achievements of this dissertation and also gives an outlook on possible future research.

### 7.1 Summary

Motivated by an extensive literature review in the area of the coverage problem, this dissertation introduced the DisCoverage paradigm, which transfers the solution to the coverage problem to the multi-robot exploration domain. Since the DisCoverage paradigm closely follows the feedback loop of the solution to the coverage problem, the design criteria of distributed coordination based on well-defined communication constraints as motivated in the introduction are fulfilled. Hence, the DisCoverage paradigm enables a group of robots to autonomously explore unknown environments. More specifically, two approaches to autonomous multi-robot exploration were derived and discussed in detail.

First, the centroidal search-based DisCoverage approach applies the limited centroidal search to derive control laws moving the robots into unexplored parts of the environment. To this end, the limited centroidal search was extended to support nonconvex environments by applying the  $\varepsilon$ -contraction to the environment. The  $\varepsilon$ -contraction defines an invariant set, and consequently no additional projection methods are required for collision avoidance between robots and obstacles. Applying the centroidal search within the visibility sets in the  $\varepsilon$ -contraction of the respective Voronoi cells solves both the coverage problem in nonconvex environments as well as the multi-robot exploration problem in nonconvex unknown environments.

Second, the orientation-based DisCoverage approach solves the exploration problem by finding robot orientations that lead the robots into unknown parts of the environment. Initially proposed for convex environments, this approach was extended to nonconvex environments by applying a transformation to the environment obtaining star-shaped regions with respect to the robot positions. Although only applied for the orientation-based DisCoverage approach, it is worth emphasizing that the proposed

transformation is suited for generic path planning problems in nonconvex environments. Compared to the centroidal search-based DisCoverage approach, the orientation-based DisCoverage approach additionally tries to maximize the expected information gain in addition to minimizing the distance costs. Therefore, the orientation-based DisCoverage approach resembles a greedy exploration strategy.

These approaches were compared to the time-optimal case as well as the MinDist approach in extensive, statistically significant simulations. The time-optimal case defines a new measure in terms of a lower bound for the time needed to explore an unknown environment. Although a formal proof of convergence of the orientation-based DisCoverage approach poses an open research question, lab experiments with e-puck robots validate this approach. A proof of convergence was given for the centroidal search-based DisCoverage approach, and contributions in the area of computational geometry are highlighted.

The DisCoverage paradigm provides a well-defined approach to fully distributed, frontier-based multi-robot exploration. Each robot acts autonomously solely based on locally available information such that the responsibility of the exploration problem is equally shared among the robots. In conclusion, the design criteria of distributed coordination under well-defined communication constraints as mentioned in the introduction are met (cf. Section 1.2, page 3).

## 7.2 Future Research

The DisCoverage paradigm proposed in this dissertation provides a practicable approach to distributed multi-robot exploration under well-defined communication constraints. In the following, several aspects are discussed that are especially suited for future research directions.

### Workload Balancing

As pointed out in the discussion (cf. Section 6.4, page 116), the Voronoi partition is not necessarily an optimal partition of the environment with respect to the multi-robot exploration problem. Consequently, in order to increase the effectiveness of the exploration process, it is worth looking into how to assign target points on the frontier to each robot such that the workload among the robots is balanced. As a result, robots would never get *unemployed*, potentially decreasing the time needed to accomplish the

exploration task. However, not relying on the Voronoi partition possibly requires to find new distributed coordination techniques. Therefore, how to balance the work among the robots in a distributed way allows for research on a much broader scale.

### **Discrete Approaches to Multi-Robot Exploration**

As indicated in the discussion in Section 6.7 (page 117), the solution to the nonconvex coverage problem and the DisCoverage multi-robot exploration approaches presented in this dissertation, and the initially proposed solution to the convex coverage problem by Bullo et al. [36] are all formulated in terms of optimization problems that are continuous in both time and space. For instance, the frontier in the exploration problem is described by a continuous line that separates explored from unexplored parts of the environment. Similarly, the density function is usually defined in terms of a continuously differentiable function in the environment. Therefore, solving the coverage and DisCoverage optimization problems typically involves numerical integration and differentiation methods, since finding analytical solutions becomes impractical. This in turn implies that software implementations solving the coverage or exploration problem as discussed in this dissertation *always* discretize the underlying optimization problem at some point. For exactly this reason, Bullo et al. [36] introduce all proofs and algorithm twice: once for the continuous case, and once for the discrete case.

With this background in mind, it appears straightforward to examine whether approaches to multi-robot exploration exist that by design are discrete in both time and space. For instance, Durham et al. [56] recently proposed a solution to the nonconvex coverage problem by representing the map as a graph instead of a polygonal environment, resulting in a value-discrete optimization problem that is solved at distinct points in time. Therefore, future research could investigate multi-robot exploration in terms of value-discrete approaches.

### **Asynchronous Distributed Coordination**

The solution to the coverage problem by Bullo et al. [36] and the DisCoverage approaches rely on a continuously up-to-date Voronoi partition of the environment. The DisCoverage Multi-Robot Exploration Framework [5] provides a discrete-time implementation of the proposed approaches, in which the robots synchronously update their Voronoi cell after each itera-

tion. Therewith, from a global perspective, the resulting Voronoi partition is always consistently updated. Instead of continuously updating the Voronoi partition, Cortés et al. [50] and Nowzari and Cortés [98] propose to use asynchronous algorithms [26, 129] to solve the coverage problem. These asynchronous algorithms allow the usage of outdated information to a certain degree, such that updating the Voronoi cells is self-triggered by the respective robots. Transferring this idea to the proposed DisCoverage approaches is subject to further research.

### **Stochastic Modeling of Map and Robot States**

Finally, the work presented in this dissertation assumes ideal localization, mapping, and communication capabilities. Although lab experiments (cf. Section 5.5, page 109) validate the DisCoverage paradigm, in practice it is inevitable to cope with noisy measurements, poor localization quality, and faulty communication. Therewith, inference methods for distributed state estimation as well as stochastic models to represent the map build an essential component in real-world multi-robot exploration. Several contributions with respect to distributed state estimation emerged as byproduct of this dissertation (cf. Euler et al. [1], Haumann et al. [7, 10], Willert et al. [16]). Still, formulating the objective functions based on non-deterministic state estimates and map information is subject to further research.

# A Computational Geometry

## A.1 $k$ -Means Clustering and Coverage

The solution to the coverage problem as proposed by Bullo et al. [36] applies the continuous-time Lloyd algorithm. This in turn implies that there is a tight relation between the solution to the coverage problem and the  $k$ -means clustering algorithm. This relation is explained in the following.

The discrete-time Lloyd algorithm is also known as the  $k$ -means clustering algorithm [28, 90]. It clusters a finite set  $\mathcal{Q}_d$  consisting of  $m$  data points  $\mathbf{q}_i$  into  $k$  disjoint clusters  $\mathcal{V}_i$  with  $i = 1, \dots, k$ , such that the union of all clusters  $\mathcal{V}_i$  defines a partition of  $\mathcal{Q}_d$ , i.e.,  $\mathcal{V}_i \cap \mathcal{V}_{j \neq i} = \emptyset$  and  $\cup_i \mathcal{V}_i = \mathcal{Q}_d$ . The objective of the  $k$ -means clustering algorithm is to find a partition  $\mathcal{V}_i$ ,  $i = 1, \dots, k$  that solves the optimization problem

$$\mathcal{H} = \sum_{i=1}^k \sum_{\mathbf{q}_j \in \mathcal{V}_i} \|\mathbf{p}_i - \mathbf{q}_j\|^2 \longrightarrow \min! \quad (\text{A.1})$$

Therein, the  $k$  cluster centers are denoted with  $\mathbf{p}_i$ , and  $\|\cdot\|$  defines the Euclidean norm. Interestingly, the inner sum  $\sum_{\mathbf{q}_j \in \mathcal{V}_i} \|\mathbf{p}_i - \mathbf{q}_j\|^2$  is minimal for the Voronoi partition, implying that each data point  $\mathbf{q}_j$ ,  $j = 1, \dots, m$  is assigned to the cluster center  $\mathbf{p}_i$  with minimum distance. Therewith, finding the clusters reduces to finding the optimal cluster centers  $\mathbf{p}_i$  for all  $i = 1, \dots, k$ . Finding the cluster centers  $\mathbf{p}_i$  minimizing (A.1) is achieved by applying Algorithm 3, which is an iterative algorithm with time index  $l$ . Algorithm 3 consists of two steps in each iteration: first the membership of the data points  $\mathbf{q}_i$  is determined based on the Voronoi cells. Second, the cluster centers are updated in order to minimize the objective function  $\mathcal{H}(l)$  at time step  $l$  by placing each cluster center  $\mathbf{p}_i$  into the center of mass of its data set  $\mathcal{V}_i$ . These two steps are repeated until the desired fitness of  $\mathcal{H}$  is achieved. As shown by MacQueen [90], Algorithm 3 in general converges to a local minimum.

In the current form, Algorithm 3 assumes that all data points  $\mathbf{q}_i \in \mathcal{Q}_d$ ,  $i = 1, \dots, m$ , are equally important. A simple generalization extends the

---

**Algorithm 3**  $k$ -means Clustering

---

Initialization of  $\mathbf{p}_i(0)$  and  $\mathcal{V}_i(0)$  for  $i = 1, \dots, k$   
**while**  $\mathcal{H}(l-1) - \mathcal{H}(l) > \varepsilon$  **do**  
   $l \mapsto l + 1$   
  **for**  $i = 1, \dots, k$  **do**  
    Update Membership: compute cluster cells  $\mathcal{V}_i(l)$   
    Update Cluster Centers:  $\mathbf{p}_i(l) = \frac{1}{|\mathcal{V}_i(l)|} \sum_{\mathbf{q}_j \in \mathcal{V}_i(l)} \mathbf{q}_j$   
  **end for**  
**end while**

---

$k$ -means clustering algorithm to other distributions. In fact, Bullo et al. [36] allow nonuniform density functions in infinite sets  $\mathcal{Q}$  – known as the environment throughout this dissertation. Allowing infinite sets  $\mathcal{Q}$  in turn implies that each Voronoi cell consists of infinitely many data points  $\mathbf{q} \in \mathcal{V}_i$ . This observation implies that (A.1) turns into

$$\mathcal{H}(l) = \sum_{i=1}^k \int_{\mathcal{V}_i(l)} \|\mathbf{p}_i(l) - \mathbf{q}\|^2 \phi(\mathbf{q}) d\mathbf{q} \longrightarrow \min! \quad (\text{A.2})$$

Extending (A.2) by switching from the discrete-time index  $l$  to the continuous-time variable  $t$  further results in

$$\mathcal{H}(t) = \sum_{i=1}^k \int_{\mathcal{V}_i(t)} \|\mathbf{p}_i(t) - \mathbf{q}\|^2 \phi(\mathbf{q}) d\mathbf{q} \longrightarrow \min! \quad (\text{A.3})$$

Continuously computing (A.3) also requires to continuously compute the cluster centers  $\mathbf{p}_i$ . Bullo et al. [36] implement this by continuously moving the cluster centers  $\mathbf{p}_i(t)$  in the direction of the centroids  $\mathbf{m}_\phi(\mathcal{V}_i)$ . These motion control laws are discussed in detail in Section 3.1. Consequently, the continuous-time Lloyd algorithm solving the coverage problem follows exactly the idea of the discrete-time  $k$ -means clustering algorithm.

## A.2 Distributed Computation of the Voronoi Partition

The Voronoi partition plays a vital role in solving geometric optimization problems, such as collision-free path planning of mobile or industrial



robots [76]. Therefore, this section provides further details on how to compute a Voronoi partition for convex sets.

Generally, computing the Voronoi partition requires the knowledge of all generator points. However, in the context of mobile robotics and distributed systems this requirement cannot necessarily be met due to restricted communication capabilities such as a limited communication range. Consequently, it is desirable to find a distributed algorithm that is capable of computing a Voronoi cell based on the neighboring generator points. Such an algorithm was proposed in [41, 50]:

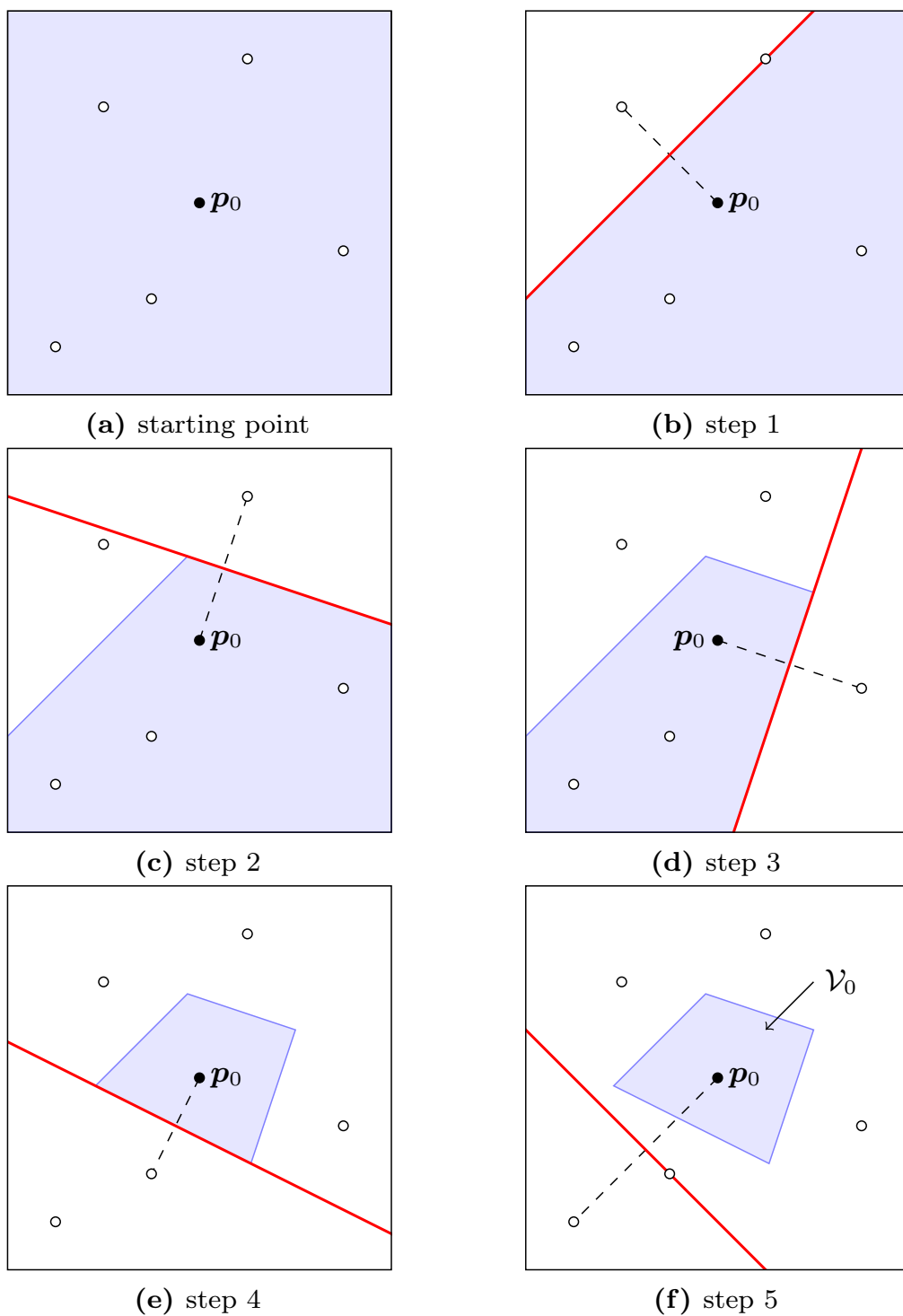
**Algorithm A.1** (Distributed computation of a Voronoi cell). Given a convex set  $\mathcal{Q} \subset \mathbb{R}^d$ . The Voronoi cell  $\mathcal{V}_i$  of a generator  $\mathbf{p}_i \in \mathcal{Q}$  can be computed distributively based on a finite set of  $k$  neighboring generator points  $\mathbf{p}_j$  according to the following algorithm:

- (i) Initialize Voronoi cell  $\mathcal{V}_i$  of  $\mathbf{p}_i$  by setting  $\mathcal{V}_i = \mathcal{Q}$ .
- (ii) For each neighboring generator  $\mathbf{p}_j, j = \{1, \dots, k\}$ 
  - a) compute the orthogonal bisector of the side of  $\mathbf{p}_i$  and  $\mathbf{p}_j$ ,
  - b) clip the polytope  $\mathcal{V}_i$  by this bisector.

As a result, one obtains the Voronoi cell  $\mathcal{V}_i$  of generator  $\mathbf{p}_i$ .

Applying Algorithm A.1 to all generator points  $\mathbf{p}_1, \dots, \mathbf{p}_n$ , one obtains all Voronoi cells. The time complexity of Algorithm A.1 is  $O(k^2)$ , where  $k \in \mathbb{N}$  is the number of neighboring generator points. Faster algorithms with respect to the computational complexity exist at an expense of losing the property of distributivity. Thus, these algorithms are not suited for distributed computation.

An example of Algorithm A.1 in the two dimensional space is depicted in Figure A.1 for a generator  $\mathbf{p}_0$  with five neighbors. The initialization of Voronoi cell  $\mathcal{V}_0$  is shown in Figure A.1(a). In Figure A.1(b)–A.1(f) the polytope is iteratively clipped by applying bisection. It is worth to note, that the final step in Figure A.1(f) does not change the Voronoi cell anymore. In a smart implementation this property can be used to ignore such cases [41], reducing the computational time. Further information on Voronoi partitions, their applications, and extensions is available in [23, 51, 62, 76, 101, 133].



**Figure A.1:** Distributed computation of a Voronoi cell

## A.3 Software

All simulations were performed with the DisCoverage Multi-Robot Exploration Framework [5], written in C++ and Qt [105]. Qt is a powerful cross-platform application and user interface (UI) framework. The DisCoverage Multi-Robot Exploration Framework allows to simulate exploration strategies in discrete-time in arbitrary environments, represented as a grid map. Batch runs are supported in order to obtain statistically significant results.

The lab experiments were performed with MATLAB<sup>®</sup> in combination with an overhead camera using the ARToolkit software [72]. This toolkit provides a tracking library which is used to identify both e-puck robots from above.

Further, the VisiLibity [99] library – a C++-library for floating-point visibility computations – as well as Boost.Geometry [66] – a generic geometry C++-library – were used to create all figures with respect to the coverage problem as well as visibility sets. It is noteworthy, that most graphics were post processed with PGF/TikZ [122], an excellent T<sub>E</sub>X macro package for generating vector graphics.

# B Hardware Specifications

## B.1 The E-puck Robot

Technical details of the e-puck robot [95]:

diameter	7 cm
height	5 cm
weight	200 g
max speed	13 cm/s
battery life time	2 hours moving
processor	dsPIC CPU (16 bit), 64 MHz
memory	8 kB RAM
flash	144 kB
actuators	2 stepper motors
perception	8 infrared proximity and light (TCRT1000)
camera	640 × 480 pixel in color
sensors	3D accelerometers

---

## Publications and Patents

- [1] Euler, J., Horn, A., Haumann, D., Adamy, J., and von Stryk, O. (2012). Cooperative N-boundary tracking in large scale environments. In *IEEE 9th International Conference on Mobile Ad hoc and Sensor Systems (MASS)*, Las Vegas, Nevada.
- [2] Frank, S., Listmann, K., Haumann, D., and Willert, V. (2010). Performance analysis for multi-robot exploration strategies. In *Simulation, Modeling, and Programming for Autonomous Robots (SIMPAN)*, pages 399–410, Darmstadt, Germany. Springer.
- [3] Haumann, A. D., Listmann, K. D., and Adamy, J. (2009). DisCoverage: Distributed optimization for multi-robot exploration. In *Workshop on Network Induced Constraints in Control (NESTCOC)*, Stuttgart, Germany.
- [4] Haumann, A. D., Listmann, K. D., and Willert, V. (2010). DisCoverage: A new paradigm for multi-robot exploration. In *IEEE International Conference on Robotics & Automation*, pages 929–934, Anchorage, Alaska.
- [5] Haumann, D. (2013). DisCoverage Multi-Robot Exploration Framework. Software electronically available at <http://gitorious.org/discoverage>.
- [6] Haumann, D., Breitenmoser, A., Willert, V., Listmann, K., and Siegwart, R. (2011). DisCoverage for non-convex environments with arbitrary obstacles. In *IEEE International Conference on Robotics and Automation (ICRA)*, Shanghai, China.
- [7] Haumann, D., Loeffler, M., and Willert, V. (2012a). Distributed Kalman filtering for noisy consensus networks with time delays. In *IEEE 9th International Conference on Mobile Ad hoc and Sensor Systems (MASS)*, Las Vegas, Nevada.
- [8] Haumann, D., Willert, V., and Listmann, K. D. (2013a). DisCoverage: From coverage to distributed multi-robot exploration. In *4th IFAC*

*Workshop on Distributed Estimation and Control in Networked Systems*, Koblenz, Germany.

- [9] Haumann, D., Willert, V., and Listmann, K. D. (2013b). DisCoverage: Von Coverage zur Multi-Roboter-Exploration. In *Workshop des VDI/VDE GMA FA 1.40 'Theoretische Verfahren der Regelungstechnik'*, Salzburg, Austria.
- [10] Haumann, D., Willert, V., and Wahrburg, A. (2012b). Kalman filtering in mobile consensus networks. In *IEEE International Symposium on Intelligent Control (ISIC)*, pages 944–950, Dubrovnik, Croatia.
- [11] Kahn, S., Haumann, D., and Willert, V. (2014). Hand-eye calibration with a depth camera: 2D or 3D? In *Proceedings of the International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISAPP)*, Lisbon, Portugal.
- [12] Klodt, L., Haumann, D., and Willert, V. (2014). Revisiting coverage control in nonconvex environments using visibility sets. In *IEEE International Conference on Robotics and Automation (ICRA)*, Hong Kong, China.
- [13] Sachidananda, V., Costantini, D., Reinl, C., Haumann, D., Petersen, K., Mogre, P. S., and Khelil, A. (2010). Simulation and evaluation of mixed-mode environments: Towards higher quality of simulations. In *Simulation, Modeling, and Programming for Autonomous Robots (SIMPAN)*, pages 133–143, Darmstadt, Germany. Springer.
- [14] Wahrburg, A., Haumann, D., and Willert, V. (2013). Minimum-variance fault isolation observers for discrete-time linear systems. In *Proceedings of the IEEE Conference on Decision and Control (CDC)*, volume 52, pages 5643–5649, Florence, Italy.
- [15] Wahrburg, A., Haumann, D., and Willert, V. (2014). Optimale Fehlerisolation mittels strukturbeschränkter Kalman-Filter. In *Workshop des VDI/VDE GMA FA 1.40 'Theoretische Verfahren der Regelungstechnik'*, Salzburg, Austria.
- [16] Willert, V., Gering, S., and Haumann, D. (2013). Bayes'sche Consensus-Regelung in dezentralen vernetzten Systemen. *at - Automatisierungstechnik*, 61(8):583–595.

- 
- [17] Willert, V., Haumann, D., and Gering, S. (2014). Decentralized Bayesian consensus over networks. In *Proceedings of the European Control Conference (ECC)*, Strasbourg, France.
- [18] Willert, V., Haumann, D., and Hartkopf, S. (2012). Erfindung betreffend die Hand-Auge-Kalibrierung von Kameras, insbesondere Tiefenbildkameras. Patent. EP12190676.2.

# Bibliography

- [19] Adamy, J. (2009). *Nichtlineare Regelungen*. Springer.
- [20] Ahmadzadeh, A., Motee, N., Jadbabaie, A., and Pappas, G. (2009). Multi-vehicle path planning in dynamically changing environments. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 2449–2454.
- [21] Amigoni, F. (2008). Experimental evaluation of some exploration strategies for mobile robots. In *Proceedings of the IEEE International Conference on Robotics & Automation (ICRA)*, pages 2818–2823.
- [22] Astolfi, A. (1999). Exponential stabilization of a wheeled mobile robot via discontinuous control. *Journal of dynamic systems, measurement, and control*, 121(1):121–126.
- [23] Aurenhammer, F. (1987). Power diagrams: Properties, algorithms and applications. *SIAM Journal on Computing*, 16(1):78–96.
- [24] Bailey, T. and Durrant-Whyte, H. (2006). Simultaneous localization and mapping (slam): part ii. *IEEE Robotics Automation Magazine*, 13(3):108–117.
- [25] Batalin, M. and Sukhatme, G. (2007). The design and analysis of an efficient local algorithm for coverage and exploration based on sensor network deployment. *IEEE Transactions on Robotics*, 23(4):661–675.
- [26] Bertsekas, D. P. and Tsitsiklis, J. N. (1997). *Parallel and Distributed Computation: Numerical Methods*. Athena Scientific.
- [27] Bhattacharya, S., Michael, N., and Kumar, V. (2013). Distributed coverage and exploration in unknown non-convex environments. In *Distributed Autonomous Robotic Systems*, volume 83 of *Springer Tracts in Advanced Robotics*, pages 61–75. Springer Berlin Heidelberg.
- [28] Bishop, C. M. (2006). *Pattern Recognition and Machine Learning*. Springer, 1 edition.



- [29] Blanchini, F. (1999). Set invariance in control. *Automatica*, 35(11):1747–1767.
- [30] Böge, A., Böge, G., Böge, W., Schlemmer, W., and Weißbach, W. (2011). *Technische Mechanik: Statik - Dynamik - Fluidmechanik - Festigkeitslehre*. Lehr- und Lernsystem Technische Mechanik. Vieweg+Teubner Verlag.
- [31] Bonani, M. and Mondada, F. (2009). The e-puck mobile robot. [http://en.wikipedia.org/wiki/E-puck\\_mobile\\_robot](http://en.wikipedia.org/wiki/E-puck_mobile_robot). Accessed: 2013-08-07.
- [32] Boxer, L. (1993). Computing deviations from convexity in polygons. *Pattern Recognition Letters*, 14(3):163–167.
- [33] Boyd, S. and Vandenberghe, L. (2004). *Convex Optimization*. Cambridge University Press, New York, NY, USA.
- [34] Breitenmoser, A., Schwager, M., Metzger, J.-C., Siegwart, R., and Rus, D. (2010). Voronoi coverage of non-convex environments with a group of networked robots. In *Proceedings of the IEEE International Conference on Robotics & Automation (ICRA)*, pages 4982–4989.
- [35] Bronstein, I. N., Semendjajew, K. A., and Musiol, G. (2005). *Taschenbuch der Mathematik*. Deutsch (Harri).
- [36] Bullo, F., Cortés, J., and Martínez, S. (2009). *Distributed Control of Robotic Networks*. Princeton University Press.
- [37] Burgard, W., Moors, M., Fox, D., Simmons, R., and Thrun, S. (2000). Collaborative multi-robot exploration. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, volume 1, pages 476–481.
- [38] Burgard, W., Moors, M., Stachniss, C., and Schneider, F. (2005). Coordinated multi-robot exploration. *IEEE Transactions on Robotics*, 21(3):376–386.
- [39] Caicedo-Núñez, C. H. and Žefran, M. (2008a). A coverage algorithm for a class of non-convex regions. In *Proceedings of the IEEE International Conference on Decision and Control (CDC)*, pages 4244–4249.

- [40] Caicedo-Núñez, C. H. and Žefran, M. (2008b). Performing coverage on nonconvex domains. In *17th IEEE International Conference on Control Applications*, pages 1019–1024.
- [41] Cao, M. and Hadjicostis, C. N. (2003). Distributed algorithms for voronoi diagrams and applications in ad-hoc networks. Technical report, University of Illinois at Urbana-Champaign, Urbana.
- [42] Cao, Y. U., Fukunaga, A. S., and Kahng, A. B. (1997). Cooperative mobile robotics: Antecedents and directions. *Autonomous Robots*, 4:226–234.
- [43] Choset, H. (2001). Coverage for robotics – a survey of recent results. *Annals of Mathematics and Artificial Intelligence*, 31:113–126.
- [44] Chvátal, V. (1975). A combinatorial theorem in plane geometry. *Journal of Combinatorial Theory, Series B*, 18(1):39–41.
- [45] Clarke, F. (1987). *Optimization and Nonsmooth Analysis*. Classics in Applied Mathematics. Society for Industrial and Applied Mathematics.
- [46] Columbus, C. and Markham, C. (2010). *Journal of Christopher Columbus (During His First Voyage, 1492-93): And Documents Relating the Voyages of John Cabot and Gaspar Corte Real*. Cambridge Library Collection - Hakluyt First Series. Cambridge University Press.
- [47] Cortés, J., Martínez, S., and Bullo, F. (2005). Spatially-distributed coverage optimization and control with limited-range interactions. *ESAIM, Control, Optimisation and Calculus of Variations*, 11(4):691–719.
- [48] Cortés, J., Martínez, S., Karatas, T., and Bullo, F. (2002a). Coverage control for mobile sensing networks. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 1327–1332.
- [49] Cortés, J., Martínez, S., Karatas, T., and Bullo, F. (2002b). Coverage control for mobile sensing networks: variations on a theme. In *Proceedings of the 10th Mediterranean Conference on Control and Automation*, pages 1–9, Lisbon, Portugal.
- [50] Cortés, J., Martínez, S., Karatas, T., and Bullo, F. (2004). Coverage control for mobile sensing networks. *IEEE Transactions on Robotics and Automation*, 20(2):243–255.

- [51] de Berg, M., Cheong, O., van Kreveld, M., and Overmars, M. (2008). *Computational Geometry: Algorithms and Applications*. Springer.
- [52] Delaunay, B. N. (1934). Sur la sphère vide. *Bulletin of Academy of Sciences of the USSR* 7, 6:793–800.
- [53] Dirichlet, G. L. (1850). Über die reduktion der positiven quadratischen formen mit drei unbestimmten ganzen zahlen. *Journal für die Reine und Angewandte Mathematik*, 40(2):209–227.
- [54] Du, Q., Faber, V., and Gunzburger, M. (1999). Centroidal voronoi tessellations: Applications and algorithms. *SIAM Review*, 41(4):637–676.
- [55] Dudek, G., Jenkin, M. R. M., Milios, E., and Wilkes, D. (1996). A taxonomy for multi-agent robotics. *Autonomous Robots*, 3:375–397.
- [56] Durham, J., Carli, R., Frasca, P., and Bullo, F. (2012). Discrete partitioning and coverage control for gossiping robots. *IEEE Transactions on Robotics*, 28(2):364–378.
- [57] Durrant-Whyte, H. and Bailey, T. (2006). Simultaneous localization and mapping: part i. *IEEE Robotics Automation Magazine*, 13(2):99–110.
- [58] Elfes, A. (1987). Sonar-based real-world mapping and navigation. *IEEE Journal of Robotics and Automation*, 3(3):249–265.
- [59] Fabbri, R., da F. Costa, L., Torelli, J. C., and Bruno, O. M. (2008). 2d euclidean distance transform algorithms: A comparative survey. *ACM Computing Surveys*, 40(1):1–44.
- [60] Flanders, H. (1973). Differentiation under the integral sign. *The American Mathematical Monthly*, 80(6):615–627.
- [61] Fletcher, R. (1987). *Practical Methods of Optimization*. John Wiley and Sons, 2nd edition.
- [62] Fortune, S. (1986). A sweepline algorithm for voronoi diagrams. In *Proceedings of the Annual ACM Symposium on Computational Geometry*, pages 313–322.
- [63] Fox, D., Burgard, W., Kruppa, H., and Thrun, S. (1999). Collaborative multi-robot localization. In *Proc. of the German Conference on Artificial Intelligence (KI) and the 21st Symposium on Pattern Recognition (DAGM)*. Springer.

- [64] Fox, D., Ko, J., Konolige, K., Limketkai, B., Schulz, D., and Stewart, B. (2006). Distributed multirobot exploration and mapping. *Proceedings of the IEEE*, 94(7):1325–1339.
- [65] Ganguli, A., Cortés, J., and Bullo, F. (2008). Distributed coverage of nonconvex environments. In Saligrama, V., editor, *Networked Sensing Information and Control*, pages 289–305. Springer.
- [66] Gehrels, B., Lalande, B., Loskot, M., and Wulkiewicz, A. (2013). Boost Geometry: a generic geometry library. Software electronically available at <http://www.boost.org/libs/geometry>.
- [67] Gusrialdi, A., Hatanaka, T., and Fujita, M. (2008). Coverage control for mobile networks with limited-range anisotropic sensors. In *Proceedings of the IEEE Conference on Decision and Control (CDC)*, pages 4263–4268.
- [68] Guzzoni, D., Cheyer, A., Julia, L., and Konolige, K. (1997). Many robots make short work. *AI Magazine*, 18(1):55–64.
- [69] Hexsel, B., Chakraborty, N., and Sycara, K. (2013). Distributed coverage control for mobile anisotropic sensor networks. Technical report, Robotics Institute, Pittsburgh, PA.
- [70] Hussein, I. and Stipanovic, D. (2007). Effective coverage control for mobile sensor networks with guaranteed collision avoidance. *IEEE Transactions on Control Systems Technology*, 15(4):642–657.
- [71] Jaromczyk, J. W. and Toussaint, G. T. (1992). Relative neighborhood graphs and their relatives. *Proceedings of IEEE*, 80(9):1502–1517.
- [72] Kato, H. and Billinghurst, M. (1999). Marker tracking and HMD calibration for a video-based augmented reality conferencing system. In *Proceedings of the International Workshop on Augmented Reality*.
- [73] Katz, V. J. (1979). The history of Stokes’ theorem. *Mathematics Magazine*, 52(3):146–156.
- [74] Keidar, M. and Kaminka, G. A. (2013). Efficient frontier detection for robot exploration. *International Journal of Robotics Research*. In press.
- [75] Khalil, H. K. (2001). *Nonlinear Systems*. Prentice Hall, 3 edition.
- [76] Klein, R. (2005). *Algorithmische Geometrie: Grundlagen, Methoden, Anwendungen*. Springer, 2 edition.

- [77] Koren, Y. and Borenstein, J. (1991). Potential field methods and their inherent limitations for mobile robot navigation. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 1398–1404.
- [78] Kramer, J. and Scheutz, M. (2007). Development environments for autonomous mobile robots: A survey. *Auton. Robots*, 22(2):101–132.
- [79] Kudelski, M., Gambardella, L. M., and Caro, G. A. D. (2013). Robonet-sim: An integrated framework for multi-robot and network simulation. *Robotics and Autonomous Systems*, 61(0):483–496.
- [80] Kwok, A. and Martinez, S. (2011). A distributed deterministic annealing algorithm for limited-range sensor coverage. *IEEE Transactions on Control Systems Technology*, 19(4):792–804.
- [81] Latombe, J.-C. (1991). *Robot Motion Planning*. Kluwer Academic Publishers, Norwell, MA, USA.
- [82] LaValle, S. M. (2006). *Planning Algorithms*. Cambridge University Press.
- [83] Lee, D. and Recce, M. (1997). Quantitative evaluation of the exploration strategies of a mobile robot. *Int. J. Rob. Res.*, 16(4):413–447.
- [84] Lee, S. G. and Egerstedt, M. (2013). Controlled coverage using time-varying density functions. In *Proceedings of the 4th IFAC Workshop on Distributed Estimation and Control in Networked Systems (NecSys)*, Koblenz, Germany.
- [85] Leonard, J. and Durrant-Whyte, H. (1991). Simultaneous map building and localization for an autonomous mobile robot. In *Proceedings of the IEEE/RSJ International Workshop on Intelligent Robots and Systems (IROS)*, pages 1442–1447.
- [86] Lloyd, S. P. (1982). Least squares quantization in pcm. *IEEE Transactions on Information Theory*, 28(2):129–137.
- [87] Lozano-Pérez, T. and Wesley, M. A. (1979). An algorithm for planning collision-free paths among polyhedral obstacles. *Commun. ACM*, 22(10):560–570.

- [88] Lu, L., Choi, Y.-K., and Wang, W. (2011). Visibility-based coverage of mobile sensors in non-convex domains. In *8th International Symposium on Voronoi Diagrams in Science and Engineering (ISVD)*, pages 105–111.
- [89] Lyapunov, A. M. (1992). The general problem of the stability of motion. *International Journal of Control*, 55(3):531–534.
- [90] MacQueen, J. B. (1967). Some methods for classification and analysis of multivariate observations. In *Proceedings of 5-th Berkeley Symposium on Mathematical Statistics and Probability*, volume 1, pages 281–297, Berkeley. University of California Press.
- [91] Makarenko, A., Williams, S., Bourgault, F., and Durrant-Whyte, H. (2002). An experiment in integrated exploration. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and System*, pages 534–539.
- [92] Marier, J.-S., Rabbath, C., and Lechevin, N. (2012). Visibility-limited coverage control using nonsmooth optimization. In *American Control Conference (ACC)*, pages 6029–6034.
- [93] Martínez, S., Cortes, J., and Bullo, F. (2007). Motion coordination with distributed information. *IEEE Control Systems Magazine*, 27(4):75–88.
- [94] Michael, N., Fink, J., and Kumar, V. (2008). Experimental testbed for large multirobot teams. *IEEE Robotics Automation Magazine*, 15(1):53–61.
- [95] Mondada, F., Bonani, M., Raemy, X., Pugh, J., Cianci, C., Klapotocz, A., Magnenat, S., Zufferey, J.-C., Floreano, D., and Martinoli, A. (2009). The e-puck, a Robot Designed for Education in Engineering. In *Proceedings of the 9th Conference on Autonomous Robot Systems and Competitions*, pages 59–65, Portugal.
- [96] Moors, M. (2000). Koordinierte multi-robot exploration. Master’s thesis, Department of Computer Science, University of Bonn.
- [97] Moravec, H. P. (1988). Sensor fusion in certainty grids for mobile robots. *AI Magazine*, 9(2):61–74.
- [98] Nowzari, C. and Cortés, J. (2011). Self-triggered coordination of robotic networks for optimal deployment. In *American Control Conference (ACC)*, pages 1039–1044.

- [99] Obermeyer, K. J. and Contributors (2008). The VisiLibity library. Software electronically available at <http://www.visilibity.org>.
- [100] Obermeyer, K. J., Ganguli, A., and Bullo, F. (2011). Multi-agent deployment for visibility coverage in polygonal environments with holes. *International Journal of Robust and Nonlinear Control*, 21(12):1467–1492.
- [101] Okabe, A., Boots, B., Sugihara, K., and Chiu, S. N. (2000). *Spatial tessellations: Concepts and applications of Voronoi diagrams*. Probability and Statistics. Wiley, NYC, 2nd edition.
- [102] Okabe, A. and Suzuki, A. (1997). Locational optimization problems solved through Voronoi diagrams. *European Journal of Operational Research*, 98(3):445–456.
- [103] Pimenta, L. C. A., Kumar, V., Mesquita, R. C., and Pereira, G. A. S. (2008a). Sensing and coverage for a network of heterogeneous robots. In *Proceedings of the 47th Conference on Decision and Control (CDC)*, pages 3947–3952.
- [104] Pimenta, L. C. A., Schwager, M., Lindsey, Q., Kumar, V., Mesquita, R. C., and Pereira, G. A. S. (2008b). Simultaneous coverage and tracking (scat) of moving targets with robot networks. In *Proceedings of the 8th International Workshop on the Algorithmic Foundations of Robotics (WAFR)*.
- [105] Qt Project (2013). Qt: A cross-platform application and UI framework. Software electronically available at <http://qt-project.org/>.
- [106] Rekleitis, I., Dudek, G., and Milios, E. (2000). Multi-robot collaboration for robust exploration. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, volume 4, pages 3164–3169.
- [107] Renzaglia, A., Doitsidis, L., Martinelli, A., and Kosmatopoulos, E. (2011). Adaptive-based distributed cooperative multi-robot coverage. In *American Control Conference (ACC)*, pages 468–473.
- [108] Rooker, M. and Birk, A. (2007). Multi-robot exploration under the constraints of wireless networking. *Control Engineering Practice*, 15(3):435–445.

- [109] Rote, G. (2013). The degree of convexity. In *European Workshop on Computational Geometry (EuroCG)*, volume 28, pages 69–72, Braunschweig, Germany.
- [110] Roy, N. and Dudek, G. (2000). Collaborative robot exploration and rendezvous: Algorithms, performance bounds and observations. *Autonomous Robots*, 11:117–136.
- [111] Schwager, M., Bullo, F., Skelly, D., and Rus, D. (2008). A ladybug exploration strategy for distributed adaptive coverage control. In *Proceedings of the International Conference on Robotics and Automation (ICRA)*, pages 2346–2353, Pasadena, CA.
- [112] Schwager, M., Rus, D., and Slotine, J.-J. (2011). Unifying geometric, probabilistic, and potential field approaches to multi-robot deployment. *International Journal of Robotics Research*, 30(3):371–383.
- [113] Sharma, P., Salapaka, S., and Beck, C. (2012). Entropy-based framework for dynamic coverage and clustering problems. *IEEE Transactions on Automatic Control*, 57(1):135–150.
- [114] Shen, S., Michael, N., and Kumar, V. (2012). Autonomous indoor 3d exploration with a micro-aerial vehicle. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 9–15.
- [115] Sheng, W., Yang, Q., Tan, J., and Xi, N. (2006). Distributed multi-robot coordination in area exploration. *Robotics and Autonomous Systems*, 54(12):945–955.
- [116] Simmons, R., Apfelbaum, D., Burgard, W., Fox, D. and Moors, M., Thrun, S., and Younes, H. (2000). Coordination for multi-robot exploration and mapping. In *Proceedings of the AAAI National Conference on Artificial Intelligence*.
- [117] Solanas, A. and Garcia, M. (2004). Coordinated multi-robot exploration through unsupervised clustering of unknown space. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, volume 1, pages 717–721.
- [118] Song, C., Feng, G., Fan, Y., and Wang, Y. (2011). Decentralized adaptive awareness coverage control for multi-agent networks. *Automatica*, 47(12):2749–2756.



- [119] Stachniss, C., Martínez Mozos, O., and Burgard, W. (2008). Efficient exploration of unknown indoor environments using a team of mobile robots. *Annals of Mathematics and Artificial Intelligence*, 52(2-4):205–227.
- [120] Steinhaus, H. (1957). Sur la division des corps matériels en parties. *Bull. Acad. Pol. Sci., Cl. III*, 4:801–804.
- [121] Stergiopoulos, J. and Tzes, A. (2009). Voronoi-based coverage optimization for mobile networks with limited sensing range – a directional search approach. In *American Control Conference*, pages 2642–2647.
- [122] Tantau, T. (2007). *The Tik Z and PGF Packages: Manual for version 2.10*.
- [123] Teraoka, S., Ushio, T., and Kanazawa, T. (2011). Voronoi coverage control with time-driven communication for mobile sensing networks with obstacles. In *50th IEEE Conference on Decision and Control and European Control Conference (CDC-ECC)*, pages 1980–1985.
- [124] Thanou, M., Stergiopoulos, Y., and Tzes, A. (2013a). Distributed coverage of mobile heterogeneous networks in non-convex environments. In *21st Mediterranean Conference on Control Automation (MED)*, pages 956–962.
- [125] Thanou, M., Stergiopoulos, Y., and Tzes, A. (2013b). Distributed coverage using geodesic metric for non-convex environments. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 933–938.
- [126] Thrun, S. (2002). Robotic mapping: A survey. In *Exploring Artificial Intelligence in the New Millennium*. Morgan Kaufmann.
- [127] Thrun, S., Burgard, W., and Fox, D. (2005). *Probabilistic Robotics (Intelligent Robotics and Autonomous Agents)*. The MIT Press.
- [128] Thrun, S. and Leonard, J. J. (2008). Simultaneous localization and mapping. In *Springer Handbook of Robotics*, pages 871–889. Springer.
- [129] Tsitsiklis, J., Bertsekas, D., and Athans, M. (1986). Distributed asynchronous deterministic and stochastic gradient optimization algorithms. *IEEE Transactions on Automatic Control*, 31(9):803–812.

- [130] Udupa, S., of Technology. Division of Engineering, C. I., and Science, A. (1977). *Collision Detection and Avoidance in Computer Controlled Manipulators*. CIT theses. California Institute of Technology.
- [131] Visser, A. and Slamet, B. (2008). Balancing the information gain against the movement cost for multi-robot frontier exploration. In H. Bruyninckx, L. Přeučil, M. K., editor, *Proceedings of the European Robotics Symposium*, pages 43–52. Springer.
- [132] Voronoi, G. (1907). Nouvelles applications des paramètres continus à la théorie des formes quadratiques. *Journal für die Reine und Angewandte Mathematik*, 133:97–178.
- [133] Wallgrün, J. O. (2010). *Hierarchical Voronoi Graphs - Spatial Representation and Reasoning for Mobile Robots*. Springer.
- [134] Wu, L., García, M. A., Puig, D., and Sole, A. (2007). Voronoi-based space partitioning for coordinated multi-robot exploration. *Journal of Physical Agents*, 1(1):37–44.
- [135] Wu, L., Puig, D., and García, M. A. (2010). Balanced multi-robot exploration through a global optimization strategy. *Journal of Physical Agents*, 4:35–44.
- [136] Yamauchi, B. (1997). A frontier-based approach for autonomous exploration. In *Proceedings of the IEEE International Symposium on Computational Intelligence in Robotics and Automation*, pages 146–151.
- [137] Zhong, M. and Cassandras, C. G. (2008a). Asynchronous distributed optimization with minimal communication. In *Proceedings of the 47th Conference on Decision and Control (CDC)*, pages 363–368.
- [138] Zhong, M. and Cassandras, C. G. (2008b). Distributed coverage control in sensor network environments with polygonal obstacles. In *17th IFAC World Congress*, pages 4162–4167, Seoul, Korea.
- [139] Zunic, J. and Rosin, P. (2004). A new convexity measure for polygons. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(7):923–934.

# Lebenslauf

---

## Persönliche Daten

---

Name	Adrian Dominik Haumann
Anschrift	Waldstr. 13 64646 Heppenheim
Geburtsdatum	27. 11. 1982
Geburtsort	Heidelberg/Baden-Württemberg

---

## Werdegang

---

seit 10/2014	CST AG, Darmstadt
08/2013 – 05/2014	Promotionsstipendium/Wissenschaftlicher Mitarbeiter der TU Darmstadt
08/2009 – 07/2012	DFG Promotionsstipendium im Graduiertenkolleg 1362, TU Darmstadt
08/2006 – 01/2007	Auslandsaufenthalt in Schweden
04/2003 – 07/2009	Studium Diplom-Informatik, TU Darmstadt
07/2002 – 04/2003	Zivildienst, Heppenheim
1993 – 2002	Starkenburger-Gymnasium, Heppenheim
1989 – 1993	Eichendorff-Schule (Grundschule), Heppenheim

---