University of Coimbra

Faculty of Science and Technology

Department of Electrical and Computer Engineering

Multi-Robot and Swarm Olfactory Search

PhD Thesis

Ali Marjovi

Coimbra, Sep. 2012

University of Coimbra
Faculty of Science and Technology
Department of Electrical and Computer Engineering

Multi-Robot and Swarm Olfactory Search

Thesis submited:

to the Electrical and Computer Engineering Department of the Faculty of Science and Technology of the University of Coimbra in partial fulfillment of the requirements for the Degree of Doctor of Philosophy.

Ali Marjovi

Coimbra, Sep. 2012

This thesis is realized under Supervision of

Professor Doctor Lino José Forte Marques

Auxiliary Professor of the

Faculty of Science and Technology, University of Coimbra

To the memory of my beloved parents Mohammad and Molouk
and
to my dear Sarvenaz

Resumo

A procura de alvos olfactivos com recurso a robôs móveis tem recebido bastante atenção por parte da comunidade científica nos últimos anos. Isto deve-se, em parte às possíveis aplicações em cenários tais como a detecção de fugas químicas, monitorização ambiental, monitorização de agentes poluentes, inspecção de aterros e operações de busca e salvamento. Algumas destas tarefas são realizadas em ambientes extremamente perigosos para os humanos, o que torna o recurso a robôs ainda mais apelativo. A assistência de robôs autónomos em tarefas como estas reduz o risco envolvido para os humanos nestas operações.

Esta tese tem como objectivo ir de encontro ao problema da busca de alvos olfactivos através do uso de um grupo de robôs móveis, equipados com sensores de gás. Este grupo pode formar uma rede de sensores móvel, permitindo a cobertura de áreas mais extensas num período de tempo mais curto, o que se traduz num melhor desempenho em diversos ambientes (e.g., na presença de ambientes ruidosos e amplos) quando comparado com uma solução que utiliza apenas um robô. Nesta tese são considerados dois tipos de grupos de robôs: "sistemas multi-robô" e "sistemas de enxames de robôs". Para cada um destes sistemas, define-se um objectivo de forma a ir ao encontro de vários desafios no âmbito da busca olfactiva com robôs móveis.

"Busca olfactiva e exploração com recurso a múltiplos robôs em ambientes desconhecidos e estruturados" e "busca olfactiva com recurso a enxames de robôs em ambientes desconhecidos e não-estruturados" são os dois objectivos principais desta tese. Em ambos os objectivos o problema de "encontrar a pluma de odor", o primeiro passo na localização de fontes de odor, é considerado como o problema principal.

Para cumprir o primeiro objectivo, esta tese apresenta um método multi-robô de procura olfactiva desenvolvido para ambientes estruturados. Este consiste num algoritmo de fronteira, descentralizado, optimizado por uma função de avaliação de custo/utilidade que considera pistas de odor (concentração de odor e informação de correntes de ar) em cada fronteira, o que leva a que o robô tente encontrar fontes de odor o mais depressa possível. Para além do desafio da "procura olfactiva", vários desafios tais como a "cooperação e partilha de tarefas", "localização e mapeamento multi-robot" e "programação e processamento" foram abordados. O método proposto foi avaliado em simulações e num ambiente estruturado real de pequena escala. Os resultados das experiências reais confirmam o efeito que as pistas de odor exercem no comportamento dos robôs e mostram que ao utilizar o algoritmo proposto, os robôs exploram primeiro as áreas com maior probabilidade de existência de fontes de odor. Os resultados das simulações confirmam a vantagem da utilização de mais robôs em ambientes mais complexos.

O segundo objectivo, "busca olfactiva com recurso a enxames de robôs em ambientes desconhecidos e não-estruturados", é abordado através de um algoritmo de optimização analítico. Definindo uma cobertura singular e múltipla (fixa e móvel) para os sensores de gás, encontrou-se a configuração óptima para N sensores móveis em diferentes ambientes, uma das principais contribuições desta tese. Descobriu-se que a topologia óptima consiste numa configuração em linha, com os sensores equidistantes. Independentemente do número de sensores, a distância óptima entre pares de sensores vizinhos depende da velocidade do vento. calculou-se uma função matemática capaz de estimar de forma óptima a distância entre pares vizinhos de sensores, com base na velocidade do vento através de estimação por regressão não-linear. Com base nos resultados das optimizações, desenvolveram-se e apresentaram-se um conjunto de forças de controlo de atracção/repulsão para enxames de robots influenciadas pelo vento. O comportamento emergente, converge para as formações óptimas. Os resultados das optimizações foram validados e avaliados por experiências em ambientes realistas.

Finalmente esta dissertação propõe um método para o "seguimento de plumas de odor e declaração de fontes de odor com exames de robôs”. Apresenta-se a implementação do método proposto em simulações e apresentam-se alguns resultados preliminares da sua validação bem como a discussão de trabalho futuro nesta área.

Abstract

Searching for olfactory targets with mobile robots has received much attention in the recent years due to its applications in chemical leak detection, environmental monitoring, pollution monitoring, inspection of landfills, and search and rescue operations. Some of these tasks are done in scenarios extremely dangerous for humans, making it more desirable to use robots instead. The use of autonomous robots to assist such tasks reduces the risks involved in these operations.

This thesis aims to address the problem of searching for olfactory targets with a group of mobile robots. A group of robots with on-board sensors can actually form a mobile sensor network, so they can cover larger area in a shorter time and they show better performance in several scenarios (e.g., in the face of noisy and large environments) in comparison to a single robot. Two types of robotic groups are considered in this thesis; "multi-robot systems" and "swarm robotic systems." For each of these systems a certain objective is defined to address several challenges in robotic olfactory search.

"Olfactory search and exploration with multiple robots in structured unknown environments" and "swarm robotics olfactory search in unstructured unknown environments" are the two defined objectives of this thesis. In both objectives, the problem of "odor plume finding" that is the first step to localize odor sources is considered as the main problem.

To fulfill the first objective, this thesis presents a multi-robot olfactory search and exploration method designed for structured environments. The method is a decentralized frontier based algorithm enhanced by a *cost/utility* evaluation function that considers the gas cues (odor concentration and airflow) at each frontier which makes the robots to try to find the odor sources as fast as possible. Several challenges namely “task sharing and cooperation”, “localization and multi-robot mapping”, “programming and processing” in addition to “olfactory search challenges” are addressed. The proposed method is evaluated in simulations and in small-scaled realistic structured environments. The results of real world experiments confirm the effect of gas cues on the behavior of the robots and show that using the proposed algorithm, robots firstly explore the areas with higher probability of existence of odor sources. Simulation results show that having more robots is more advantageous in a complex environment than in a simple environment.

We address the second objective, “swarm robotics olfactory search in unstructured unknown environments”, through an analytical optimization approach. Defining single and multiple (stationary and mobile) gas sensors coverage and finding the optimal configuration of N mobile sensors in different environmental conditions are among the main novelties of this thesis. We found that the topology of optimal solutions is in the form of line configuration, with equal distance between each pair of neighboring sensors. Regardless of number of sensors, the optimal distance between neighboring pairs depends on the wind speed. A mathematical function that can accurately estimate the optimal distances based on the wind speed was computed by nonlinear regression estimation. Moreover, based on the results of optimizations, we present and design a set of wind-biased virtual attractive/repulsive control forces for the swarm robots such that their emergent behavior converges to the optimal formations. The optimization results were validated and evaluated by experiments in realistic environments.

Finally, this dissertation proposes a method for “swarm robotics odor plume tracking and source declaration.” We present the implementation of the proposed method in simulations and provide some preliminary results of its validations and discuss the future works of this work.

Acknowledgment

I would like to thank Professor Doctor Lino Marques for his supervision, support and encouragement. I should also thank ISR-UC for the facilities provided and Foundation for Science and Technology of Portugal for the granted scholarship SFRH /BD /45740 /2008 during my study. I am very grateful to my friends and colleges specially Gonçalo Cabrita, Pedro Sousa, João Nunes and Doctor Mahmoud Tavakoli for their support on my project and for their friendship.

I owe my special thanks to my love, Sarvenaz, for all her love, patience, support and inspiration. Her support is much appreciated and has led to many interesting and good-spirited discussions relating to this research. My special gratitude is due to my sisters and brothers and their families for their loving support. Last, but not least, I thank my late father Mohammd and my late mother Molouk, who were my first teachers and their love and encouragement inspired my passion for learning.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

"A gas leak near the intersection of Cambie Street and West 7th Avenue in Vancouver forced the evacuation of at least 10 buildings for five hours Thursday. The leak could be smelled for several blocks in the area. 'We advise any citizens in the affected area that smell gas in their premise, to open their windows to ventilate and exit the building,' said Roder, Fire and Rescue Services Captain." CBC News,[1] Sep. 20 , 2012.

This quote from a recent news article describes a few harmful effects of a chemical release in a residential area. The consequences of a chemical leak may be more detrimental than this story (e.g., Bhopal disaster,[2] 1984), and may become a worldwide disaster. In these scenarios, the origin of the contamination should be located as quickly as possible to minimize the damage. Since the chemicals are often hazardous to human health, a promising solution to this problem is to deploy a group of intelligent robotic vehicles that can autonomously navigate in the environment and look for desired olfactory targets, instead of jeopardizing the lives of human operators.

This thesis describes the results of four years of research about olfactory searching and exploration with groups of robots. In particular, we study the case in which a cooperating team of independent mobile robots uses local observations of a chemical concentration to rapidly converge to the location of the release source. Although several goals are achieved and multiple

[1] http://www.cbc.ca/news/canada/british-columbia/story/2012/09/20/bc-cambie-gas-leak.html

[2] http://en.wikipedia.org/wiki/Bhopal_disaster

novel contributions are provided by this thesis, the work is still on-going and a lot of questions are still remained unanswered.

1.1 Motivation

Olfaction is a sense widely used by social animals to find mates, to detect hidden resources or threats, and to mark paths and territories. Some birds (e.g., red knots and dunlins), insects (e.g., ants, bees, termites, locusts) and marine animals (e.g., fish, antarctic krill, and small shrimp-like crustaceans) communicate and cooperate with each other while searching for olfactory targets [Wya03]. Robots equipped with olfactory sensors can potentially show similar behaviors with applications in areas such as search and rescue operations.

In the early 1990s, researchers started to study the integration of olfactory abilities into mobile robots in order to replace trained animals (e.g., search and rescue dogs) in search operations [ISNM94, DTRMS94]. Compared to animals, robots can work for longer periods without fatigue, they can be deployed quickly and maintained with lower cost, and most importantly, they can enter dangerous zones. It is expected that olfactory robotics plays more significant roles in areas such as searching for harmful gas leakages, smuggling drugs (e.g., heroin), survivors in collapsed buildings, mines in battle fields, and chemical targets in anti-terrorist operations.

Searching for olfactory targets with mobile robots has received much attention in the recent years. Robotics olfactory search finds applications in environmental monitoring [DM12], chemical leak detection [RTDMS95], pollution monitoring [FCL12], inspection of landfills [HBLNT12], and search and rescue operations [WZWH10]. Some of these tasks are done in extremely dangerous scenarios for humans, being desirable to use robots instead. The use of autonomous robots to assist such tasks reduces the risks involved in these operations.

The problem of autonomous search and exploration have been extensively addressed with single robot systems. However, in several cases (e.g., in noisy and large environments) most single robot approaches fail or show very low performance [HMG03]. Multiple robots with on-board sensors can actually create a mobile sensor network, so they cover instantaneously a larger area than a single robot. In the case of olfactory search where the target profile is inherently noisy and it does not have smooth gradient, using multiple robots can potentially increase the performance in comparison to a single robot. Multi-robot systems usually extend the limitations of single robot systems and in most cases have higher performance, although the complexity of the system usually increases.

Among the advantages of multiple robot systems (in comparison to single robot systems), it can be highlighted that these systems can potentially provide higher task accomplishment speed, higher accuracy and fault tolerance. For example, multiple robots can localize themselves more accurately [LBL10], fulfill search and exploration missions faster [SYTX06, BMSS05], and generate maps of unknown environments more accurately [HB05]. As argued in [JZ00], performance/cost ratio is the main advantage of multi-robot systems over single-robot systems. Using a group of simple robots with a subset of the capabilities to accomplish a particular task is usually less expensive to design and maintain than a single monolithic robot with all of the necessary capabilities.

Several studies have concluded that the system-level cooperative behavior of many biological systems such as social insects is robust, flexible and scalable [CDF$^+$01, BP95, HM02], and these characteristics are also envisaged for multi-robot olfactory systems. The *robustness* of a group of robots is related to its ability to continue operating, although at lower performance, despite failure of individuals, or disturbances in the environment. The *flexibility* is defined as the ability of robots to adapt themselves to environmental changes. The *scalability* requires that a robotic system be able to operate under a wide range of group sizes. The robots' coordination mechanisms should be relatively undisturbed by changes in the group sizes.

The mentioned attributes are desirable for olfactory search systems. This study presents cooperative multi-robot and swarming methodologies for olfactory search problems and tries to achieve the mentioned benefits that are offered by such systems.

1.2 Olfactory search problem

One of the main problems in olfactory robotics is localizing static odor sources in an environment [XL05, MNdA02b, KRRJ06, LLD06, LRZ03, IKNM96]. In real situations, this problem consists of three sub-problems:

1. Finding traces of the odor plume,
2. Tracking the odor plume towards its source,
3. Localizing and declaring the odor source.

"Declaration of the odor source" consists in accurately localizing the odor source in close vicinity. This problem is either complemented by other sensing modalities, like vision [KRRJ06] or is addressed by analyzing the proximity of plume detection points [FPLA03]. This subproblem is out of scope of this thesis but it is discussed in some parts of this dissertation.

"Tracking the respective odor plume", i.e., following the plume toward its source, has been the problem more deeply studied, though mostly in laboratory conditions [LRZ03, IKNM96, LRM+08], using heuristic and bio-inspired odor tracking algorithms [MNdA02b, FCM+09] and mainly through centralized robotic approaches. This subproblem is not the main focus of this dissertation, however it is discussed in some sections.

"Finding odor plumes", that is, searching the environment randomly or systematically in order to find odor clues, is the main goal of this study. This is a less studied sub-problem and the first phase in search for odor sources [HMG03]. Plume finding problem is usually addressed by general exploration methods [MNMdA09a], mapping [LCLG08, LD04], or coverage techniques namely zig-zag sweeping, casting [PBiBB+06], random wandering [ITTM06], biased random walks [MNdA02b], lévy-taxis [PBG09], and spiral movements [FCM+09], which are also used for other spatial exploration tasks and are not efficiently designed for odor plume finding.

Sensor coverage [LWBSL12, BS07], exploration [BMSS05, FKK+06], and environmental monitoring/mapping [LCLG08, DM12] are three correlated problems to "odor plume finding". Sensor coverage is the problem of spatially deployment of sensors in an environment in order to provide maximum coverage on a known working area. Exploration is the task of visiting all areas of an unknown environment with mobile robots to obtain knowledge about that area. Environmental mapping and monitoring is the problem of acquiring sensory data about all the positions of a working environment using mobile robots. The techniques of coverage, exploration, and monitoring are valuable to olfactory search and exploration, but they do not provide a complete solution.

1.3 Problem constraints and assumptions

To design and propose robotic algorithms, several parameters such as robots types, required equipments and sensors, considered environment and problem assumptions are important. The considered hardware and robotic platforms restrict the computational complexity of a strategy. For example, the use of a relatively sophisticated robot in contrast to an inexpensive robot significantly affects the functionality and performance of proposed solutions. We classify these parameters into following main categories.

1.3.1 Robotic systems

Regarding the robots and required hardware, this work mostly uses (and adapts) off-the-shelf materials. Based on relatively simple available robots, two types of systems are considered in this thesis:

- Cooperative multi-robot systems,
- Swarm robotic systems.

Cooperative multi-robot systems

“A multi-robot system is a group of robots, which are organized into a multi-agent architecture so as to cooperatively carry out a common task, [WdS08].” There are two different architectures for multi-robot systems; centralized and decentralized (also known as distributed) architectures. For both architectures, higher efficiency in performing the tasks is achieved by dividing the main task to a set of subtasks and distributing the subtasks between the individual robots. In the centralized approaches, a base station is responsible for distributing the subtasks between the robots, whereas in decentralized architectures no such base station exist. It is widely claimed (e.g., in [BHD94, Ark92, Ste94]) that decentralized architectures have several inherent advantages over centralized architectures, including fault tolerance, reliability, and scalability [Bro94]. Moreover, there are hybrid centralized/decentralized architectures ([CP95, Nor93]) wherein there is a central planner that applies high-level control over robots which have some degrees of autonomy in their task distribution.

As argued in [JZ00], performance/cost ratio is the main advantage of cooperative multi-robot systems over single-robot systems. Using heterogeneous robots with a subset of the capabilities to accomplish a particular task, one can use simpler robots that are less expensive to engineer than a single monolithic robot with all of the necessary capabilities.

Swarm robotic systems

Swarm robotics is a relatively new field that has gained significant attention in the scientific community since the pioneering work by Reynolds on simulating flocks of birds [Rey87]. Swarm robotics is inspired by the observation of social insects like ants, termites, bees and wasps, which are the fascinating examples of how a large number of simple individuals can interact to create collectively highly intelligent systems [Şah04]. Several studies have concluded that there is no centralized coordination mechanisms behind the operation of many biological systems such as social insects [Sha05].

A swarm robotic system consists of (usually) large number of simple robots that operate autonomously and independently under decentralized control laws. Although each agent follows relatively simple rules, the swarm can collectively accomplish complex tasks at the macroscopic level. Therefore in swarm robotics, complex global behavior of the system emerges from the interaction of simple agents.

Comparision of swarm and multi-robot systems

1. **The relation between performance and number of robots:** From the social behaviors of some social animals (e.g., fish schools and ant colonies), it can be seen that adding (or removing) group members to (from) the system does not affect the functionality of the system. This is the main difference between multi-robot and swarm robotic systems, i.e., removing several group members in a multi-robot system might strongly decrease its functionality while in a robotic swarm does not. Additionally, in many cases, a swarm robotic system achieves its emergent global behavior only if the swarm size is very large, however, in most cooperative multi-robot systems, the functionality can be fulfilled even with a few number of robots.

2. **Complexity of the robots:** Usually, in swarm robotic systems the individual robots are very simple and weak in terms of robotic abilities. The robots' sensors, actuators, processing resources, communication and localization capabilities are usually poor in swarm robotic systems in comparison to the cooperative multi-robot systems.

3. **The overall behavior of the system:** In swarm robotics, the global behavior of the system usually emerges from the interaction of simple robots and it is beyond the ability of each individual robot. On the other hand, each individual in multi-robot systems is usually able to perform the whole task.

This thesis proposes different strategies for multi-robot and swarm robotic systems. In both systems, we seek decentralized control approaches where the robots are independent, thus cooperation is achieved through individuals' communications.

1.3.2 Environment

Working environment is an important factor for the performance of search algorithms. The profiles of a variety of emissions modalities namely chemicals, light, sound, RF waves, and radioactivity, depend on the size, shape and conditions of the environment. Particularly, regarding chemical release, the Reynolds number of the environment affects how chemical substances are dispersed.

In environments with low Reynolds numbers (i.e., in which viscosity dominates over inertia), the dispersal is dominated by molecular diffusion resulting in Gaussian concentration distributions with peak at chemical source [Hin86]. This condition is encountered when the robots are relatively small in comparison to the size of the particles in the environment, such as at microscopic scales in [HWPG07] and nanorobotics applications [GHL05], and also inside fluids with high viscosity or moving at very low speeds [Vog96] .

At medium and high Reynolds numbers, odor molecules are carried by the flow forming an odor plume due to the fact that their dispersal is dominated by the flow. As the odor travels away from the source, the plume becomes more diluted due to diffusion. Two processes cause the diffusion of odors: molecular diffusion and turbulence [RW02]. Molecular diffusion is a slow process whose effect can be neglected in compare to turbulence. This is the condition of most robotic scenarios where the robots are relatively huge, compared to the chemical particles. Plumes may meander, become erratic, and contain fluctuating instantaneous concentration gradients with isolated peak concentrations that are much higher than the surrounding average concentrations [MEC92]. These conditions are often encountered in many robotic applications. Thus, olfactory search at medium and high Reynolds numbers conditions is suited to the problem of this thesis.

In the medium and high Reynolds number conditions, there are two environmental parameters that directly affect the chemical emission profiles:

1. **State of the air flow**

   In the absence of strong background flow, chemical profiles often resemble a Gaussian distribution [Hin86]. In this condition, usually a simple gradient climbing algorithm can efficiently solve the problem of olfactory search [RBHSW03].

   However, in the presence of background flow, chemical emissions are carried from their source by wind currents. A robot equipped with anemometers to measure wind direction can direct its search upwind when it encounters the chemical [MdA06a]. Turbulence in the background flow creates packets of variable chemical concentration in the air [Hin86]. This patchiness has a significant impact on the progress of a search algorithms. This thesis investigates olfactory search solutions in environments under stable airflow.

2. **Shape of the environment and the obstacles**

   In terms of physical shape we can categorize the environments into the following groups:

   (a) **Outdoor environments:** Field observations show that in outdoor natural environments, the airflow magnitude is usually not weak (e.g., ≥ 0.2 m/s) and its direction changes randomly and rapidly with large scales [Ary99]. In such conditions, the distribution of the odor also changes quickly. There are many previous studies that model the odor dispersal in these types of environments based on the results of field observations [Bri73, Gif60].

   (b) **Indoor/semi-outdoor, scattered obstacles environments:** In very large indoor environments (e.g., warehouses) and some outdoor areas, the state of the air-flow stays unchanged for long periods of time. In these environments, there might exist a few obstacles scattered in the area which cause turbulences in the flow. However, the odor plume shape does not change rapidly in comparison to outdoor environments.

   (c) **Indoor structured environments:** In most human built facilities (e.g., houses, hospitals, tunnels, mines, etc.), the structure of the environment dictates the air-flow direction, thus a chemical release is highly constrained by the environment's shape. In these environments most of the general odor dispersal models do not provide accurate estimations. In cases where the map of the working environment is *a-priori* known, computational fluid dynamics (CFD) simulations may be used to model the odor dispersal [LBSS+12].

This thesis focuses on indoor/semi-outdoor structured and unstructured environments (case b and c).

To efficiently search structured environments, the robots should use the map of the working area in their algorithms since the odor dispersal is highly dependent on the environment's shape. However, in most real applications the map of the environment is not *a-priori* known so the robots need to generate the map while searching for olfactory targets. This requires a high level of reasoning, communication and cooperation to efficiently search, explore and map these environment. On the other hand in obstacle free and scattered obstacles environments, the odor dispersal follows the airflow direction detectable by the robots. In these environments a lower level of reasoning can be used to search for olfactory targets since mapping the environment and using its structure is not essential for the algorithms. Due to these two facts, this thesis presents swarm robotic approaches (i.e., simple local reactive behaviors) for olfactory search in unstructured environments, while proposes multi-robot systems (i.e., relatively complex high level cooperative algorithms) for olfactory search and exploration in structured environments.

1.3.3 Algorithms' complexity

The order of complexity of algorithms and the control period are two other important parameters in designing search and exploration methodologies for group of robots. These parameters define the processing power requirement of the system. In swarm robotic systems, in contrast to many multi-robot systems, the robots do not have high processing resources and thus they are not able to run heavy processing tasks. This thesis avoids using very sophisticated robotic algorithms for swarming systems. On the other hand regarding multi-robot systems, we presents algorithms that can be run in many simple robotic platforms.

1.3.4 Communication

Robots need to communicate with each other due to at least two reasons; first, to coordinate actions such that the task are efficiently accomplished without conflicts, and second, to exchange their knowledge about the mission.

The amount of information transfered between the robots, the communication range and the communication rate and speed are among the most important parameters in designing methodologies for multi-robot and swarm robotic systems. In cooperative multi-robot systems, the robots are usually able to communicate globally and exchange relatively high amounts of data. In contrast, in swarm robotic systems the robots usually exchange small messages locally in a short range. Therefore, the designed algorithms for multi-robot systems in this thesis use global communications to exchange relatively high amounts of data, while the algorithms for swarm robotic systems only exchange short messages between the robots.

1.3.5 Number of robots

The number of robots used in a mission has significant impact on the functionality and performance of most search methodologies.

In cooperative multi-robot systems, the more robots employed in a task, the more important the coordination between their actions becomes. In these systems, scalable high-level cooperation mechanisms are required to take the advantage of maximum number of robots and to increase the efficiency of the system. In contrast, swarm robotic systems usually consist of large number of simple robots where the desired collective behaviors emerge from the interactions between the robots, and sometimes interactions between the robots and the environment. Usually, these systems are functional only if a large number of robots are employed whereas the performance does not change remarkably by small changes in the number of robots.

One of the main constraints that restricts the number of robots in multi-robot or in swarm systems is the total cost of the system. The robots in cooperative multi-robot systems are usually more expensive than the robots in swarm robotic systems (due to higher capabilities). Hence, the number of robots in multi-robot systems is usually limited to a few robots, whereas, the number of robots in a swarm may be countless.

The proposed methodologies in this thesis are flexible and scalable to the number of robots, although most of the experiments are done using a few robots in small scaled scenarios.

1.3.6 Olfactory sensors

A variety of different gas sensor types have been developed during the last decades. A review of these sensors is presented by Arshak et al. in [AML$^+$04]. Metal oxides semiconductor (MOX), conductive and fluorescent polymer composites, optical, gas sensitive field effect transistors, quartz crystal micro-balance (QCM), and surface acoustic wave (SAW) sensors are some examples. The metal oxide semiconductor gas sensors are the most popular in mobile robotics due to their high quality/cost ratio. A problem of most of these gas sensors is that their response time and their recovery time both are lengthy (of the order of several seconds to tens of seconds [LMWZ11]). In other words, these sensors do not respond immediately to the fast concentration fluctuations. This issue restricts the search and navigation algorithms to the strategies that only consider long-term exposure models for odor dispersal. This work considers only algorithms suitable for the robots equipped with off-the-shelf metal oxide semiconductor gas sensors.

1.4 Objectives

The research objective of this thesis is twofold; (i) to study multi-robot systems which cooperatively explore structured environments and search for olfactory clues, (ii) to research on swarming robots which cooperatively and efficiently search for odor plumes in unstructured environments.

1.4.1 Multi-robot olfactory search and exploration in structured environments

In unknown structured environments, robots need to gain information about the working environment in order to be able to efficiently search for olfactory targets. Thus, olfactory search problem can be complemented with exploration methods. The problem of *multi-robot exploration* is the problem of visiting all areas of an unknown environment with a set of mobile robots. This problem is important for a wide range of applications, like search and rescue operation, cleaning, patrolling, demining, grass cutting, etc. Most of the works in the area of exploration have been implemented with a single robot that needs to have complete knowledge about its localization. The ability to perform online complete exploration with multiple cooperative mobile robots is presented in this thesis.

Olfactory search is a different but related problem to *exploration*. In an *exploration* method basically a group of robots want to visit the entire environment, detect eventual odor clues and localize their source. This is very different in comparison to *search* where a group of robots navigate in the environment in order to find the odor clues as fast as possible while they finally *explore* the whole environment.

This objective seeks to answer the following questions in real-world indoor structured environments. How can robots in a multi-robot system cooperate with each other without having a leader? How can a group of robots share the tasks in a distributed way without existence of a central task allocator? What informations should the robots exchange and when? How can a robot integrate the gained information of another robot into its own data, in order to increase its knowledge about the environment, the mission and the other robots' status? How can the robots cooperatively process a computationally heavy task?

1.4.2 Olfactory swarm search strategies in unstructured environments

In unstructured environments under airflow the odor profile is directed by the wind and is predictable by estimation models. Simple swarm robotic behaviors can be designed to efficiently find the plumes in these environments. Despite many other works in the field of olfactory swarming which propose heuristic and intuitive approaches, this thesis looks for analytical solutions.

The second objective of this thesis is to address the problem of odor plume finding by robotic swarms in unknown unstructured environments. The goal is to propose an efficient methodology to use a large number of simple mobile sensing nodes to increase the probability of finding an odor plume.

This objective seeks answers for the following questions. How can swarm robots cooperatively navigate in an environment without having a leader? What is the best strategy for the swarm to look for olfactory targets in an unknown environment? What is the best spatial formation of swarm robots in search for odor plumes? How should the individual's behaviors be designed in order to achieve the desired emergent swarm behaviors?

1.5 Challenges

Several challenges need to be addressed to fulfill the two mentioned objectives of these thesis. Some of these challenges are mainly related to the first objective (multi-robot olfactory search and exploration in structured environments) and some of them are more related to the second objective (olfactory swarm search in unstructured environments). This section describes these challenges.

1.5.1 Task sharing and cooperation

Since in multi-robot/swarm-robotic systems there are several robots working together to solve a global problem, the cooperation technique is a critical issue on their performance. One of the main challenges in robotics is how the individuals can cooperate in a decentralized way in order to increase the whole system's performance. This challenge is addressed differently in multi-robot systems than in swarm robotic systems. Usually in multi-robot systems, the individual robots communicate with each other and by using a distributed common policy/protocol the subtasks are taken by the robots autonomously. In swarm robotics, usually the robots have minimal communications, thus, the global goals are achieved through the emergent result of basic individual's behaviors.

1.5.2 Navigation and formation

One of the main problems in cooperative robotics (specially in swarm robotics) is to design control laws such that the group as a whole will achieve the intended goals, such as collision free navigation or forming a spatial structure. Many researchers have been working on this challenge during the recent years. Lack of a central control station poses a tough challenge in designing these tasks.

1.5.3 Localization and multi-robot mapping

This challenge is mainly related to the first objective of this thesis in situations where the environment is not a-priori known and there is no global localization system. While multiple robots cooperatively search and explore an environment, information from individual robots must be integrated to produce a single globally consistent map. This is a difficult problem to solve when the robots do not have a global positioning system nor a common reference frame [HB05].

1.5.4 Programming and processing in multi-robot systems

Programming a group of robots is not like other types of programming. The model of distributed computing (using many computers to work on a single large task) is somehow similar to cooperative robotic programming. However, unlike distributed computing, each individual in robotic groups deals with unique stimuli (e.g., each robot is in a different location at any given time). The design of distributed algorithms is a main challenge in these robotics systems. Some approaches use a control unit that coordinates other robots. Other approaches use techniques borrowed from nature to give the system itself a type of collective intelligence. Most the current research in robotics focuses on finding the most efficient and simple ways to program individuals to achieve desired group behaviors. On the other hand, in most multi-robot systems, an individual robot is not capable to solve computationally hard problems due to lack of high processing power. In several situations, an approach is needed to empower these systems in their problem solving.

1.5.5 Olfactory challenges

In addition to the general robotics challenges mentioned so far, this thesis deals with several other challenges raised from olfactory aspect of work. Olfactory robotics techniques are highly dependent on the environmental conditions. Proposing an olfactory search and exploration method for a group of robots that needs less requirements/assumptions is a tough challenge in the literature.

1.5.6 Swarm olfaction challenges

Finding the best spatial formation for the swarm robots in searching for an odor plume is a challenge that none of the works either in the olfactory search area or in the swarm robotics field has ever addressed. Another challenge is to design and propose simple control methodologies for swarm robots to emerge to the best spatial formations.

1.6 Novelties and contributions

The main theme of this work, olfactory search and exploration with a group of robots, raised several related problems whose solutions provided some original contributions.

1.6.1 Olfactory-based multi-robot exploration in structured environments

This work presents a novel strategy that takes olfactory clues into the decision making of exploration of unknown structured environments with multiple robots. The proposed method is a decentralized frontier based algorithm enhanced by a $cost/utility$ evaluation function that considers the odor concentration and airflow at each frontier so that the robots try to find the odor sources as fast as possible. It is shown that modified exploration methods practically address the problem of searching for olfactory sources in unknown structured environments by a team of robots.

1.6.2 Olfactory coverage maximization

This work presents an analytical method to find the optimal spatial formation of gas sensors which maximizes their coverage area in unstructured environments. Defining single and multiple gas sensors coverage and finding the optimal configuration of N stationary sensors in different environmental conditions are among the main novelties of this thesis. Several works exist on optimal coverage for different types of sensors, but none of them has ever addressed and found the best formation of gas sensors for finding any possible odor plume in an unknown environment.

1.6.3 Finding optimal formation of mobile swarm robots in odor plume finding

This thesis analytically finds the optimal spatial formation of swarm robots for odor plume finding considering cross-wind movement for the swarm. This is a strong initial step towards odor source localization with a swarm robotics approach.

1.6.4 Wind-biased potential fields

This work proposes a set of wind-biased virtual attractive/repulsive control forces for swarm robots such that their emergent behavior converges to the optimal formations. None of the previously designed control systems in olfactory robotics community has ever biased the virtual forces by the wind effect.

1.6.5 Robotic clusters concept

The novel concept of *robotic clusters* to empower multi-robot systems in their problem solving is introduced in this work. A robotic cluster is a group of individual robots which are able to share their processing resources, therefore, the robots can solve difficult problems by using the processing units of other robots. None of the studies in robotics has ever suggested to share the processing resources of agents/robots to increase the processing efficiency of each individual in the system.

1.6.6 Parallel solution for topological map merging

This work presents a novel parallel solution for the conventional problem of multi-robot topological map merging. This solution is based on robotic clusters concept. This is the first parallel implementation of topological map merging problem.

1.7 Outline

Multi-robot olfactory search and exploration in structured environments that is the first of objective of this thesis is studied in chapter 2. This chapter provides a detailed state of the art of multi-robot olfactory search and exploration and presents and explains a proposed method for this problem. Multi-robot task sharing and cooperation, olfactory search, localization, topological mapping, parallel map merging and shared processing are the main challenges discussed in this chapter. The proposed method is validated by simulations and also in small-scaled realistic structured environments.

Swarm robotics olfactory search that is the second objective of the thesis is studied in chapter 3. This chapter first reviews the related works about the problem of swarm robotics odor plume finding and then presents analytic optimizations for this problem. Afterwards, it explains the design and implementation of swarming behaviors for individual robots to achieve optimal configurations. Pseudo-Guassian odor dispersal model, definition of gas sensor area coverage, optimal sensor deployment and optimal mobile sensor formation are the main topics of this chapter. The methodology is validated in simulations and in realistic environment.

Although the two problems of “swarm odor plume tracking” and “swarm odor source declaration” are not among the main objectives of this thesis, chapter 4 provides a state of the art and presents a perspective proposed method for addressing these problems. Simulations results of the proposed methodology are presented in this chapter. Finally chapter 5 present the overall conclusions and future work of this study.

Chapter 2

Multi-robot olfactory search and exploration

"A thousand people cannot convince one by words to the extent that one person can convince a thousand by action."
Zoroaster, 11th-10th century BC

The first main objective of this thesis is to develop a group of autonomous robots applied to search and explore an unknown structured environment. Search and rescue operations inside buildings, caves, tunnels and mines can be extremely dangerous tasks. An example of such extremely risky situations is human operations inside industrial warehouses during a fire. In these scenarios human senses can become severely impaired; smoke reduces the visibility, communication is rendered impossible by the noise caused by the fire, and additionally dangerous vapors and toxins may be released. The use of autonomous robots to assist such tasks reduces the risks involved in these operations. Robots can search for toxic chemicals and other desired targets while they explore the environment, providing real-time data about the discovered map and the status of the facility. In unknown environments, search operations are frequently complemented with the environment exploration.

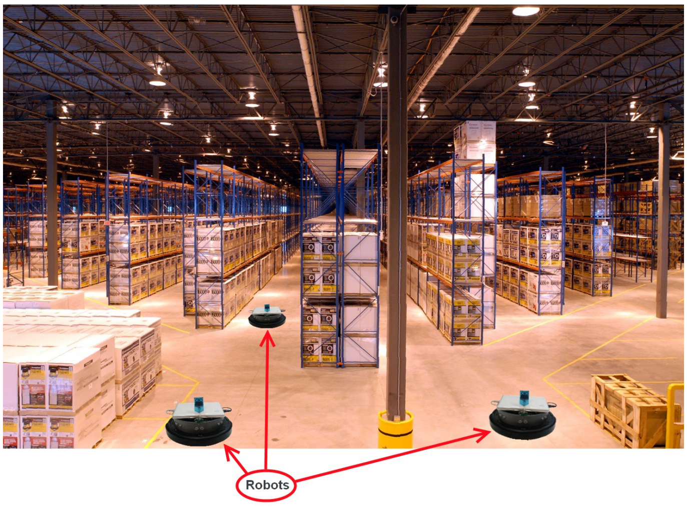


Figure 2.1: Robots searching in a warehouse.

2.1 Problem statement

Consider a group of N mobile robots, moving in R^2 that are labeled as $R_1, R_2, ..., R_N$. Each robot $R_i (i = 1, ..., N)$ is able to communicate with the other robots located at a short distance Δ. The robots are equipped with sensors for measuring the odor concentration and air flow direction. There are unknown number of odor sources in the area which emit odor gas into the environment. There is no central base-station for the system, so the robots should act separately and independently from each other. There is no global localization system and the robots' odometry is not very accurate. There is no prior knowledge about the environment except that it is a structured building similar to a warehouse containing corridors, corners, branches, crosses, etc, (Fig. 2.1). The problem is to localize all odor sources in the environment, explore the whole area, and generate a map of the environment.

2.2 Related works

Efficient search and exploration in unknown environments is a fundamental problem for mobile robotics. As this problem becomes increasingly solved by single robots, especially in obstacle free environments, the next problem is to extend these techniques to groups of robots on structured environments. Using multi-robot systems may potentially provide several advantages over single robot systems, namely faster exploration, better accuracy, and system fault tolerance. However, in addition to the problems occurring in single robot olfactory search and exploration, the extension to multiple robots poses following challenges:

- Multi-robot olfactory searching,
- Multi-robot task sharing and cooperation,
- Localization and multi-robot mapping,
- Multi-robot shared processing.

2.2.1 Multi-robot olfactory searching

Odor molecules released from their source are taken away by the wind and create a plume. This process is called *advection*. While patches of molecules move away from the source, they spatially expand and their mean concentration decreases. This is due to two different processes: *molecular diffusion* and *turbulent diffusion* [MEC92]. Molecular diffusion is described as the random motion of the molecules that causes them to move gradually apart. Turbulent diffusion is a process that causes the patch of molecules to physically torn apart by flow turbulence [KRRJ06]. Molecular diffusion is a slow and small-scale phenomenon. In contract, turbulent diffusion is strong in a wide range of temporal and spatial scales. These processes yield to the following three-dimensional advection-diffusion equation for the odor concentration [Sto11]:

$$\frac{\partial C}{\partial t} + \nabla C\vec{U} = \nabla(K\nabla C) + Q \tag{2.1}$$

where C represents the odor concentration, U is the wind speed, K denotes the turbulent diffusion coefficient and Q is the source release rate. In this equation, the term $\nabla C\vec{U}$ represents advection and $\nabla(K\nabla C)$ stands for the turbulent diffusion.

The above equation implies that the air-flow direction and speed are the key parameters in the distribution of odor in the environments. In the environments with no (or a few) obstacles, this differential equation can be solved and a probabilistic pseudo-Gaussian model can be find that describes the odor concentration in average time exposure (chapter 3 goes to the details of this model). However, in structured environments where the airflow is governed by the structure of the facility, no general solution can be achieved. If the map of a structured environment is known, computational fluid dynamics (CFD) modeling can be used to estimate the odor concentration in any point of interest [ZC07]. If the map is unknown, neither mathematical models nor numerical simulations can be used for concentration estimation or for source localization. In this case, this thesis proposes modified exploration methods to be used for olfactory search.

Researchers have developed methods that employ combinations and variations of plume acquisition [MAdA03, MNdA02b] and plume upwind following [MNS$^+$10, MAdA03, LANM10] using reactive control algorithms (a comparison of these methods is in [RBHSW03]). Most of these approaches are inspired by very simple creatures like moths [FCM$^+$09], glowworms [KG08], etc, therefore they are mostly low-level algorithms making the robot react to the environmental changes based on some defined rules. On the other hand the majority of the works have focused on either single robot experiments [LCLG08, LD04, FCM$^+$09] or multiple robots operating in open areas free of obstacles [HMG02, MNS$^+$10, MNdA06] with a background fluid flow. Despite most of these works, this chapter needs to fulfill the goal with higher level of cooperation of multiple robots in real structured environments. How can a group of robots efficiently localize odor sources in structured buildings? This is a challenge that should be more studied in the robotic community.

2.2.2 Task sharing and cooperation

Cooperation is a key parameter in the performance of a group of robots which are trying to solve a general search problem. The more robots that search an environment, the more important the coordination between their actions becomes. The cooperation, communication, and management of the robots in a multi-agent system can be done either in centralized way by using a base station as the server, or decentralized way by having a distributed behavioral based method (like in [MNMdA09a, BMSS05, TP07]). This study tackles the searching problem in unknown environments without using a central station. The lack of a central station makes it more difficult to distribute the tasks between the robots. Since the environment is unknown, the robots are unaware of the tasks before searching the area, i.e. there can be no kind of task allocation before the start of the mission. Task allocation must be done automatically during the mission by the participating robots.

The problem statement in this chapter explained that there are unknown number of odor sources in the environment and the robots should localize all of them. Since the robots have no a priori knowledge about the number of odor sources they must operate in the environment until they cover all the area, i.e. they must continue searching while there are still unexplored areas. Therefore this search problem should be complimented with unknown environment exploration problem.

Singh and Fujimura [SF93] presented a decentralized online approach for heterogeneous robots. In their method the robots work independently most of the time. When a robot finds a situation that is difficult to solve by itself, it sends the problem to another robot which may be able to solve the situation. The candidate robot is chosen by trading off the number of areas to be explored, the size of the robot and the straight-line distance between the robot and the target region. This technique generates a grid geometric map; therefore, the accuracy of the map depends on the grid size. Moreover, all the robots need to have a considerable amount of memory to store the entire map. Yamauchi [Yam98] proposed a distributed method for multi-robot exploration, yielding a robust solution even with the loss of one or more vehicles. A key aspect of that approach involves sharing map information among the robotic agents so that they execute their own exploration strategy independently of all other agents. The robots move to the closest frontier[1] according to the current map, however, there is no coordination component which chooses different frontiers for the individual robots. If the robots know their relative locations and share a map of the explored area, then effective coordination can be achieved through guiding the robots into different, non-overlapping areas of the environment. In other words, effective coordination can be achieved by extracting exploration frontiers from the partial maps and assigning robots to frontiers based on a global measure of performance [BMSS05, MNMdA09b]. Frontier based exploration is a simple approach for decentralized

[1] Frontiers are the borders of the partial map, between explored space and unexplored area.

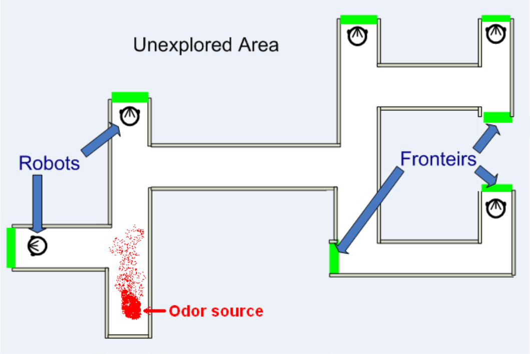


Figure 2.2: Multi-robot frontier-based search and exploration.

multiple robot task allocation (Fig. 2.2). These frontiers, thus, represent locations that are reachable from within the partial map and provide opportunities for exploring unknown terrain, thereby allowing the robots to greedily maximize information gain [ZSBDT02]. However, these methods should have a strategy to avoid sending two robots toward the same frontier.

Most of the related works in this area (namely [MNMdA09b] and [BMSS05]) try to explore the environment with minimal excess effort in the minimum possible time, however, the main goal of this project is olfactory search rather than exploration. The exploration must be done in such a way that the group of robots automatically intends to look for the odor sources in the environment.

2.2.3 Localization and multi-robot mapping

While multiple robots cooperatively search and explore an environment, information from individual robots must be integrated to produce a single globally consistent map. Most of the existing approaches to coordinate multi-robot mapping assume that all agents know their locations in a shared (partial) map of the environment [MNMdA09a, BMSS05, MNMdA09b]. Having a physical general positioning system is an undesired constraint in unknown areas where there is no previous knowledge about the environment.

Simultaneous localization and mapping (SLAM) has been a topic of much interest because it provides an autonomous vehicle with the ability to discern and represent its location in a feature rich environment. Some of the statistical techniques used in SLAM include extended

Kalman filters, particle filters (Monte Carlo methods) and scan matching of range data [ENT05]. However, in the context of metric map building, SLAM's performance depends on the accuracy of the environmental sensors and requires very high data processing and also communication between the robots.

While most research on multi-robot mapping has addressed the problem by creation of occupancy grid maps ([GNSPK00, Thr01, KSF$^+$03]), some research has been done on feature based or topological maps ([MM11a, MM10, DS00b, FNL02, WJD08]). Topological maps provide a brief characterization of the navigability of a structured environment, and, with measurements easily collected during exploration, the vertices of the map can be embedded in a metric space [HB05]. These maps use a graph to represent possibilities for navigation through an environment and need less memory than their metric counterpart. The proposed approach employs a topological mapping technique, so the robots only exchange few environmental features.

Using topological maps, the problem of "map merging" is reduced to a "graph merging" problem [HB05, KH04]. Jennings et al. [JKWT98] used individual robots to create topological partial maps, and then used a simple distance metric to merge the maps considering a global reference frame for all robots. Dudek et al. [DJMW98] created maps of a graph-like world under the assumption that all robots start from the same point in the graph. Konolige et al. [KFL$^+$03] analyzed and showed the efficiency of taking a feature-based approach to merging instead of attempting to match occupancy data.

Topological map merging is a difficult problem to solve when the robots do not have a common reference frame or physical global positioning system [HB05]. Most of the works in multi-robot mapping assume that all robots in the system have a common reference frame, an assumption not made in this thesis. A few notable exceptions tackle the problem without a common reference frame. Ko et al. [KSF$^+$03] presented a method in which robots exchange occupancy maps and localize themselves in each others maps using particle filters. Dedeoglu and Sukhatme [DS00b] proposed a different approach for merging landmark-based maps without a common reference frame. They used heuristics to estimate a transformation between two maps using a single-vertex match found and paired other vertexes that are approximately close to each other under this transformation. The work most related to ours is that of Huang and Beevers [HB05] who have presented a method that creates vertex and edge pairings using the structure of the maps and then estimates a transformation using this match. By comparing the map structure, mismatches can be discard earlier in the algorithm, and the transformations can be computed using multiple-vertex matches instead of single-vertex matches. In our previous experiments in [MM10], it was assumed that although there is no common reference frame, the robots start the mission from the same initial node. This assumption made it easier to solve the problem of map merging with multiple robots. Here we do not make any of these assumptions and present an approach for this problem considering no common reference frame and no common initial node. Moreover we develop a novel parallel algorithm for this problem to run in robotic clusters using parallel programming approaches.

Whereas most approaches to topological map merging and related problems have focused on using either map structure or map geometry [DS00a], the proposed algorithm in this chapter takes advantage of both. Similar to [HB05], the use of map structure allows quick identification of potential vertex matches in the maps (and rejection of mismatches), while the use of map geometry enables the algorithm to directly merge maps with multiple (disconnected) overlapping regions. However, if the geometric data of the local maps are not obtained through a unique coordinate system these methods are not functional. This chapter deals with two issues concerning map merging, the first, dealing with the uncertainty in the localization of each robot by correcting it using the information of the partial maps, and the second, presenting an approach for map merging for the case in which the robots' coordinate systems are different from one another.

2.2.4 Multi-robot shared processing (Robotic Clusters)

Multi-robot systems mostly consist of simple robots that usually have low computational capability due to cost constraints [YN10]. However, there are situations where high computational capabilities are required to solve complex problems. This thesis proposes a novel approach based on computer cluster concepts to increase the processing efficiency of each individual robot inside a distributed robotic system. The general idea is to design a corporation mechanism for high performance computing in multi-robot systems and solving complex problems in robotics applications.

Parallelism in multi-robot systems in particular is achieved by similar methodologies defined in multi-agent systems [MNSB11]. Robots usually run a distributed algorithm and based on a given criteria, the tasks are assigned among the robots. In these systems, usually the individual robots do not have a global view of the problem and they pick up the subtasks based on different strategies (e.g., market-based approaches [DZKS06], priority approaches [CBR02] or coalitions [FTL07, MNMdA09a]). Each one of them solves its local problem and the emergent result will be the global solution.

Basically, in multi-robot systems, robots share the physical subtasks in order to increase the efficiency of the team [CFK97, DJMW96]. None of the studies in this field has ever suggested to share the processing resources of agents/robots to increase the processing efficiency of each individual in the system. However, there are several robotic applications which require high processing resources for the individual robots in a multi-robot system, namely "mapping" [BEL$^+$08, FP11], "robotics vision" [TAT$^+$06], "path planning" [CP09] and "large-scale signal processing" [BHH09]. For example, in multi-robot exploration and mapping, usually robots share their local topological maps and each one of them merges them to generate a global map of the whole environment. This problem is computationally hard if the robots do not share a reference coordinate frame in complex environments. In fact, topological map merging in this case is an NP-hard "maximum common subgraph isomorphism" problem. One of the best solutions for this problem in robotics with several simplifications and assumptions is an $O(n^4 log(n))$ algorithm [HB05], i.e., if the number of vertexes raised to several hundred, most robots current are not able to merge two maps in real time. Considering the fact that robots usually have to merge more than two maps (based on the number of robots participating in the mission) and the low processing capability of the usual robots, this problem poses a tough challenge. There are many similar applications in this field that require high processing resources for individual robots.

Most of distributed robotic systems tend to use simple robots with advanced communication capabilities (e.g., [BTI11]). Nowadays, robotics tend to use small processing

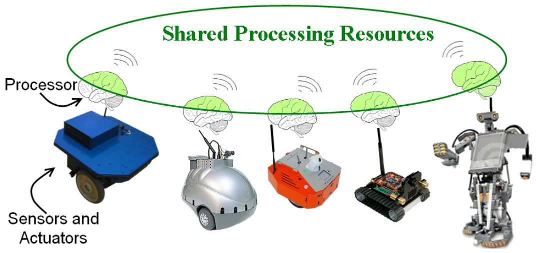


Figure 2.3: Concept of robotic clusters

boards which run operating systems (e.g., Linux) and come with embedded wireless LAN adapters (e.g., Gumstix[1]). Utilizing robots' communication capabilities, this work proposes a corporation mechanism based on computer cluster concepts to facilitate problem solving in multi-robot systems.

Robotic clusters definition

In a ***robotic cluster*** systems, each robot is a computation node of a computer cluster in order to empower multi-robot systems in solving computationally hard problems [MCM12]. Computer clusters are typically independent computers connected via a single local area network (LAN). Robots can also establish a computer cluster through wireless networks to form a ***robotic cluster***. We propose the following definition for a robotic cluster and then explain its characteristics, requirements and implementations in details.

Definition 1 ***A* robotic cluster *is a group of individual robots which are able to share their processing resources among the group in order to quickly solve computationally hard problems.***

In a robotic cluster, the processing resources can be shared by applying computer clustering approaches. In these systems, each node is still able to run its own tasks independently, moreover, the CPUs of these robots are shared in the cluster. Therefore, when a robot sends some processing jobs to the others, the robots simultaneously run their own tasks and also the

[1] http://www.gumstix.net

shared requested job. Fig. 2.3 demonstrates the concept of a robotic cluster which consists of multiple heterogeneous robots connected through a wireless network.

Applications of robotic clusters are tasks that demand

- extreme processing resources, and
- relatively cheap designs.

Processing high amount of data or mapping a large and complex environment with multiple simple robots are two examples of these applications. Processing and cost, both, are hard constraints that emphasize simplicity of the individual robots, and thus motivate a cluster approach to solve computationally hard problems. Excessive research is needed to find methodologies that allow designing and implementing robotic clusters to address several challenges in robotics research. The concept, requirements, characteristics and architecture of robotic clusters are explained in [MCM12], and the problem of "topological map merging" is addressed here by using "robotic clusters" concept.

Characteristics of robotic clusters

Robotic clusters benefit from a number of advantages offered by concepts of computer clusters, including:

I. Processing Power: Each robot in a cluster is potentially able to use the processors of other robots. Most of the previous experiments with real robots show that the robots usually perform simple tasks (e.g., data logging and navigation) that do not require high processing power, but in some situations they need to process a huge amount of data (e.g., when a robot needs to process sensory data or finds itself in a different situation or needs to make a hard decision). Therefore in a group of robots, most of the robots usually perform simple tasks and only few robots need high processing resources. Using the clustering concepts the robots which need processing power will be able to use the processing units of the other robots. This idea can be more developed such that there can exist some special robots that do not participate directly in robotic mission but help the other robots with sharing their processing power.

II. Reduced Cost: The price of off-the-shelf simple computers or small laptops has decreased in recent years. These simple computers can be used as the processing unit of each individual robot and as a cluster they provide high processing capacity. Generally, in comparison with single robots systems, the parallel processing power of a robotic cluster is, in many cases, more cost efficient than a single robot with the same processing capability.

III. Scalability: One of the greatest advantages of robotic clusters is the scalability that they offer. While complex single robots with high processing resources have a fixed processing capacity, robotic clusters are easily expandable by adding additional nodes to the system when conditions change.

IV. Availability: When a single supper robot fails, the entire mission fails. However, if a robot in a robotic cluster fails, its operations can be simply transferred to another robot within the cluster, ensuring that there is no interruption in the service.

V. Required equipments for robotic clusters: For implementing a robotic cluster, there is no need of extra hardware rather than a wireless network and some robots equipped with simple computers. Most of the recently developed robots are already equipped with small computers running an operating systems capable of clustering. Only some software packages (named middleware) and applications should be installed on the robots to form a robotic cluster.

For more details about the idea of robotic clusters the reader is referred to reference [MCM12] that is one of the recent author's publications. In that article several issues namely "programming in robotic clusters", "middleware of robotic clusters", "load distribution in robotic clusters", "Communication in robotic clusters", "Dynamic architecture of robotic clusters", and "Robotic clusters' spatial distribution" were discussed in details. This chapter addresses the problem of topological map merging using robotic clusters concept.

One of the contribution of this study is to propose a novel parallel solution for the problem of topological map merging based on the concept of robotic clusters [MCM12]. The robots are able to establish a computer cluster and share their processing resources (brains) in solving complex problems (in this case topological map merging problem) when needed. None of the previously designed multi-robot systems in robotics community has ever used the computer cluster concepts in order to increase the robots' processing capability. Additionally, this is the first parallel solution for the problem of topological map merging in multi-robot systems.

Another significant contribution of this thesis is taking olfactory clues into the decision making of exploration of unknown structured environments with multiple robots. The proposed method is a decentralized frontier based algorithm enhanced by a $cost/utility$ evaluation function that considers the odor concentration and airflow at each frontier so that the robots will try to find the odor sources as fast as possible. The researchers who have presented cooperative multi-robot approaches in the field of olfactory search, have not addressed the problem by a frontier-based search and exploration method (maybe one of the reasons is that most of them do not consider the problem in structured environments and with multiple robots). On the other hand, the researchers who have been working on multi-robot unknown environment exploration have not addressed the problem of olfactory search. We believe that for searching olfactory sources in unknown structured environments by a team of robots, modified exploration methods can practically address the problem.

2.3 The proposed method for multi-robot olfactory search and exploration

It is desired to find odor sources and explore the whole environment as fast as possible. Therefore, it is essential that the robots share their tasks and individually achieve the objectives through optimal paths towards the odor sources. In an unknown environment, the immediate goals are the frontiers. while the robots are exploring an area, there are several unexplored regions, which poses a problem of how to assign specific frontiers to the individual robots without the existence of a specific task allocator. In the proposed approach, the robots firstly decide to explore the frontiers which indicate higher odor concentration. The robots must avoid selecting the same frontier, this may result in collision concerns. Another problem is the lack of base station, so the robots should be able to explore autonomously and also avoid collisions between themselves. To address these problems, the proposed method is based on a behavioral decision-making exploration strategy that is shown in algorithm 1. The flowchart of this method is depicted in Fig. 2.4.

Algorithm 1: Odor source search and exploration algorithm

```
// Definitions:
//     feature: features are corridors, corners, etc. in a structured environment.
//     node: a node represents an environmental feature in the topological map.
//     frontier: an unexplored link in a node in the map.
begin
    while there is at least one frontier in the map do
        repeat
            Go forward and follow potential field algorithm()
            Path planning improvement() // explained in section 2.3.4
            Feature extraction() // explained in section 2.3.2
        until getting a different environmental feature;
        C = Measure odor concentration()
        W = Measure wind direction by anemometer()
        if current node exists in the map then
            Update map's data()
            Localization correction() // explained in section 2.3.6
        else
            Add current node to map()
        if (M >odor_threshold) and (W is not explored) then
            Set Objective to go(W)
        else
            F = unassigned frontier with highest $Utility - Cost()$
            Assign frontier F to this robot() // in the map
            D = the best path based on the A* algorithm(F)
            Set the new Objective to go(D)
        if Odor source detected() then
            Report this place as an odor source into the map()
        Send the new local map to the other robots()
        if received map data from other robots then
            Map Merging() // explained in section 2.3.5
end
```

2.3.1 Odor source search and exploration algorithm

Algorithm.1 describes the decision making technique that should be run on every robot. The robots start exploring the environment independently. Each robot goes forward to get into new features in the environment. It generates its own local topological map out of the detected features of the environment and also transmits this local map to the other robots which are working on the same environment. When a robot has to make a decision to select its future path (e.g., in the branches), it first measures odor concentration in that place. If the odor concentration is more than a certain threshold, it means that the robot is in an odor plume and it must go upwind direction in order to localize the source. However if a robot is traveling inside an already

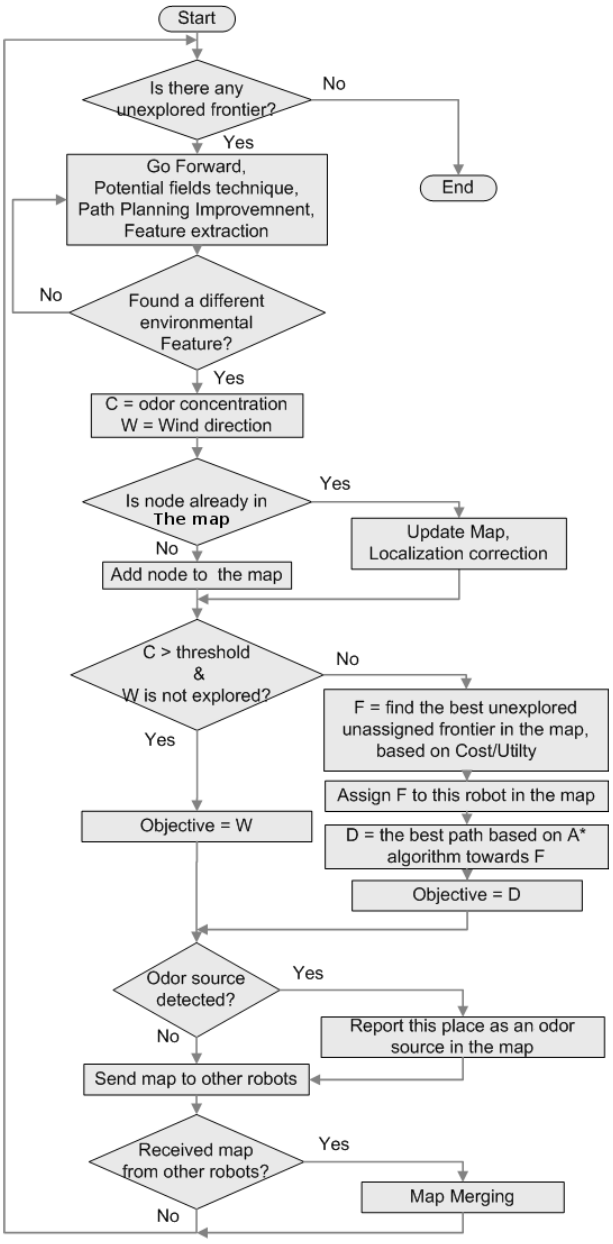

Start
Is there any unexplored frontier?
No
End
Yes
Go Forward, Potential fields technique, Path Planning Improvemnent, Feature extraction
No
Found a different environmental Feature?
Yes
C = odor concentration W = Wind direction
Is node already in The map
Yes
Update Map, Localization correction
No
Add node to the map
C > threshold & W is not explored?
No
F = find the best unexplored unassigned frontier in the map, based on Cost/Utilty
Assign F to this robot in the map
D = the best path based on A* algorithm towards F
Objective = D
Yes
Objective = W
Odor source detected?
Yes
Report this place as an odor source in the map
No
Send map to other robots
Received map from other robots?
Yes
Map Merging
No

Figure 2.4: The flowchart of the proposed method

explored area and wants to select a frontier to explore, the frontier should be selected based on the *cost* of reaching it and the *utility* it can provide to the search. The *cost* is calculated through the A* method [HNR68], where it simultaneously determines the optimal path to reach the frontier and its distance. Therefore, the *cost* is proportional to the distance that the robot has to pass to reach the frontier.

$$cost(i,R) = dist(A^*_{i=0,n}[(X_R,Y_R),(X_{f_i},Y_{f_i})]) \qquad (1)$$

where:

$(X_R,Y_R) \rightarrow$ position of the robot R

$(X_{f_i},Y_{f_i}) \rightarrow$ position of the frontier i

$n \rightarrow$ number of frontiers

The *utility* depends on the level of odor concentration in that frontier, which means that if there are several frontiers at similar distances, the robot will go to the one that has higher *utility*, i.e., higher odor concentration.

$$\forall i \in \{1..n\}$$

$$utility(i) = Odor\_Concentration(i) \qquad (2)$$

The $profit$ of a frontier i for a robot R is then calculated as

$$profit(i,R) = utility(i) - \beta \times cost(i,R) \qquad (3)$$

where β represents a coefficient representing the relative values of *cost* and *utility*.

Since each robot always tries to maximize the $profit$ function in its frontier selection, the frontiers with higher odor concentration will be explored faster by the robots which are more close to them. Fig. 2.5 shows a snapshot of three robots searching in a structured environment. The already explored area is highlighted and the frontiers are colored by green. The robots select frontiers based on the explained algorithm.

During exploration and navigation, the robots are simultaneously acquiring olfactory information (odor concentration and air flow direction) of the environment. While a robot is traveling up-wind direction, if the level of detected gas decreases suddenly it means that the robot has passed an odor source during its path. On the other hand, if a robot is traveling in the down-wind direction and it starts to detect a high concentration of odor, it means that there is an odor source in this location.

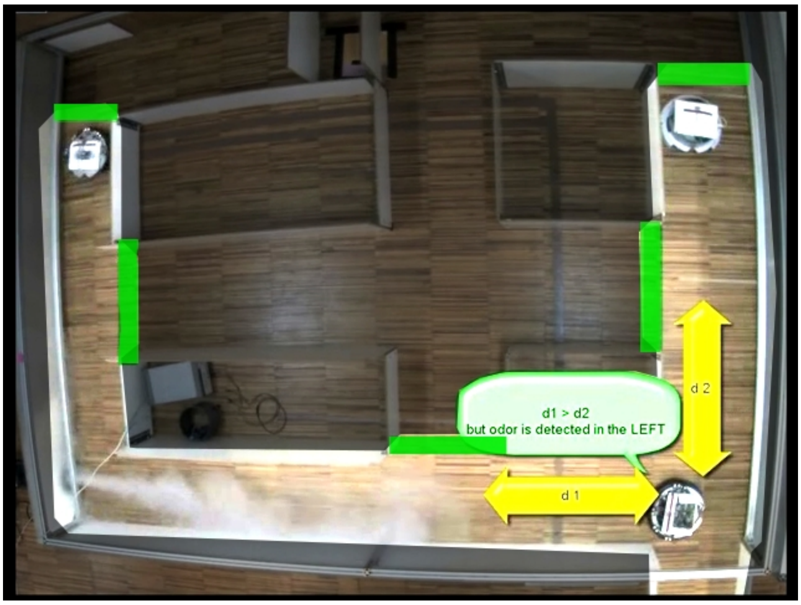


Figure 2.5: 3 robots searching in a small-scaled structured environment.

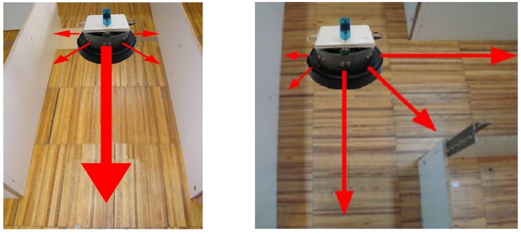


Figure 2.6: Feature detection, left: corridor, right: branch.

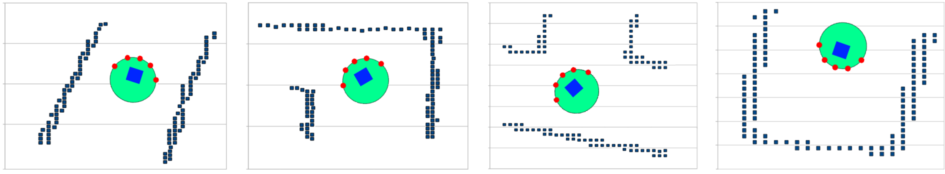


Figure 2.7: Local sensing frame in different features. Left to right: Corridor, Corner, T-junction, dead-end.

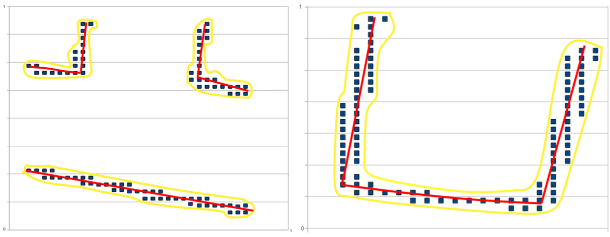


Figure 2.8: Feature detection process, blob detection and Segment-line extraction in a T-junction (left) and a dead-end (right). The blobs are presented in yellow and the extracted lines are presented in red.

The robots generate the topological map of the environment during their search and exploration mission. Within the topological map, besides having information regarding the kind of nodes, their position and the odor concentration in that feature, it also has data describing the location of the robots and their target frontier. Through this data, a robot can see which frontiers are unexplored, their position and if any robot has targeted them as its objective (see Fig. 2.10). Therefore, the robots will not attempt to explore the same frontiers. Each robot is aware of the frontier that the other robots have aimed to explore, so it will choose another frontier that is unexplored and unassigned to any other robot. As a result, the robots will autonomously pick up the tasks in a way that not any frontier will be assigned to more than one robot.

The robots repeat this procedure while there is at least one unexplored frontier in the environment. At the end all odor sources will be localized and the whole environment will be explored, no matter how many odor sources are in the environment or how many robots participate in the search.

2.3.2 Environmental features

With the type of environment considered in this chapter, there are five types of features that can be extracted; corridors, corners, crosses, T-junctions (branches) and dead-ends. The robots recognize environmental features based on a local metric map (also called local sensing frame) constructed with the range sensors while the robot moves across the environment. Fig. 2.6 shows a robot in different environmental features and Fig. 2.7 shows local sensing frames extracted by the robots. The known features are classified based on the differences that they show in their sensing frames. Similar to several other studies (e.g., [RM06]) this thesis uses pattern recognition algorithms that has been addressed in its literature. Blob detection [SS02], line-segment extraction [BHR86] and corner detection [GD04] are used to classify the features (See Table 2.1). A dead-end (shown in Fig 2.8) has only one blob that is a unique identification of this feature among all the other ones. T-junction has 3 blobs, cross has 4 blobs, corner has 2 and corridor also has 2 blobs. To correctly identify the features, similar to the method used in [DSA01] for corridor detection, the histogram of dominant angles of the extracted blobs

Table 2.1: Characteristics used in classifying the environmental features.

Feature name	No. of Blobs	No. of Segments	No. of Corners	Histogram Peaks
Corridor	2	2	0	0
Corner	2	4	2	0 and 90
T-junction	3	5	2	0 and 90
Cross	4	8	4	0 and 90
Dead-end	1	3	2	0 and 90

is calculated to provide more clues to correctly identify the features. For example one of the differences between a corridor and a corner is that a corner has a peak in the 90 degree value of its histogram since it has two 90 degree angles in its segment lines, but the corridors don't. This histogram is also useful to avoid the false positive results and is applied to detect the other features (e.g., dead-ends). Table 2.1 presents the characteristics of different features based on mentioned parameters. The function *IsSimilar()* in Algorithm 2 does the blob detection, corner detection, segment-line extraction and histogram calculation on the acquired data and returns *True* if the local sensing frame presents the same characteristics as the compared known feature.

Algorithm 2: Feature detection algorithm

```
// Inputs:
//    Known_Feature[]: dataset of sensory data of the known features
//    M: Number of needed iterations to process an area
//    N: Number of needed similar features to recognize that feature as detected
// Output: Detected feature
Start :
i = 1
repeat
    Found = False
    while !Found do
        S[] = acquire sensors' data()
        foreach Known_Feature[k] do
            if IsSimilar(S[],Known_Feature[k]) then
                F[i++] = Known_Feature[k]
                Found = True
                Break
until $i \geq N$;
MF = Most_repeated_Value_in_F[]
Freq = Frequency_of_MF_in_F[]
if $Freq \leq M$ then
    Goto Start            // did not detect any feature!
return (MF)
```

In algorithm 2 the robot constantly tries to detect the environmental features of its surrounding area. If at least in M out of N iterations the robot detects one feature, it considers this feature as the detected feature of the local area. The values of M and N depend on (i) the velocity of the robot, (ii) the accuracy of the sensors and (iii) the characteristics of environmental features. If the robot could not detect M similar features in N iterations, it must slow-down its speed and does the feature extraction of that place from the beginning. Based on this method the robots were able to accurately classify the environmental features in the testing environments. It was found that this feature extraction and classification process highly depends on the accuracy of sensory system. The accuracy of the sonars that are used in the maze-like environment tests (explained in Sec. 2.4.1) is less than 3 centimeters in the range of 1.5 meters. Having this accuracy, the robots could perfectly distinguish the environmental features from each other. If the environment has more complex features the sensors of the robots also need to be more complex. In all of experiments in this chapter, it is assumed that the robots are able to accurately classify the environmental features.

This study has tried to design and explain the algorithm independent from feature extraction methods. In fact any kind of environmental feature extraction technique that can lead to topological mapping is appropriate to be used in this project. We used five sonar sensors mounted on different sides of robots and algorithm 2 to distinguish basic environment features from each other, i.e., each robot can recognize if it is located in a corridor or in a corner or any other environmental feature. We have also tested this process using a Laser range finder instead of sonars with a few minor changes in extracting local sensing frames.

2.3.3 Robots' motion

The low level of autonomous navigation of a robot relies on the ability of the robot to simultaneously achieve its target goal and avoid the obstacles in the environment. In an unknown environment the robot should reactively avoid the obstacles while exploring. To avoid the obstacles, a reactive potential field control method [Kha86] was used. The targeted goal of the potential field is provided by the exploration algorithm (explained in Sec 2.3.1) and the distance to obstacles are measured by the range sensors. In this method, the goal generates

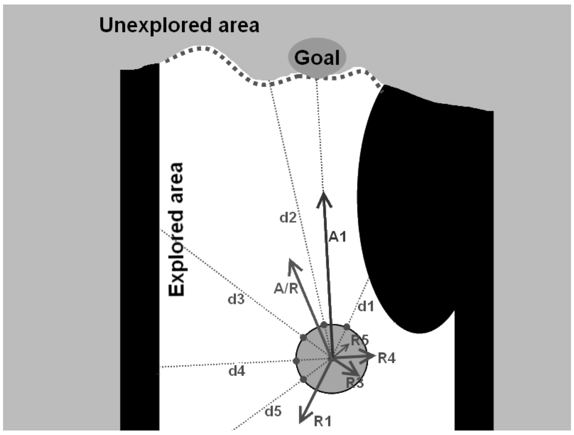


Figure 2.9: Artificial attractive/repulsive forces in an unknown environment. d1 to d5 correspond the distance measured by five sonar sensors. R1 to R5 are the artificial repulsive forces that are calculated based on the inverse distance of d1 to d5. "A1" represents the artificial attraction to the targeted goal. "A/R" shows the summation of the attractive and repulsive forces and the direction that the robot actually takes.

an attractive artificial force, while the obstacles generate artificial repulsive forces to the robot. The resulting motion is obtained by the summation of these attractive/repulsive forces. Fig. 2.9 illustrates the action of this method in a corridor. Considering the robot holonomic with N range sensors, its next velocity $\vec{v}(t+\Delta t)$ can be calculated based on its current velocity $\vec{v}(t)$ and the forces that are applied to it $\vec{F}(t)$;

$$\vec{v}(t+\Delta t) = \vec{v}(t) + \mu\vec{F}(t)\Delta t \tag{2.2}$$

with μ a constant coefficient. The equation of forces that is applied to a robot is given by:

$$\vec{F}(t) = c_1\Big(Goal\_Pose - x(t)\Big)^n + \sum_{j=1}^{N} \frac{c_2}{\left|d(j)\right|^m}(\vec{Vec_j}) \tag{2.3}$$

The position of the robot at time t is described by $x(t) \in R^2$. $Goal\_Pose - x(t)$ is the distance vector between the robot and its targeted goal. The first term in (2.3) is an attraction force towards the goal that its amplitude is relative to the distance between the robot and its goal. Since $d(j)$ is simply the distance between the robot and the surrounding environment that is reported by the sensor j, the second term is a force that is the inverse function of the distance of the robot to the surrounding obstacles. $\vec{Vec}_j$ is a predefined vector whose magnitude is set to one and its direction is from sensor j towards the center of the robot. c_1 and c_2 are two positive coefficients and n and m are even integer parameters, therefore the first term is always a force towards the goal and the second term is a forces from obstacles towards the robot, meaning that, the robot will try to get far from the obstacles and reach its targeted goal.

```
Algorithm 3: Path planning improvement
// This algorithm is called in Algorithm. 1, line 9.
Calculate a confined circular area around the robot()
if any other robot is inside the circle then
    if there is a possibility of being in a direct collision pattern then
        if it is a direct collision path then
            Recalculate the objective()
        else
            if both currently exploring frontiers OR both moving inside explored area then
                //Give priority to the one which has lowest ID.
                if the robot ID is higher that the other robot's ID then
                    go back to previous node and stop for a while() // so the other robot passes
                else
                    Continue navigating
            else
                //Give priority to the one that is exploring a frontier.
                if the robot is inside explored area then
                    go back to previous node and stop for a while() // so the other robot passes
                else
                    Continue navigating
else
    Continue search and exploration algorithm.
```

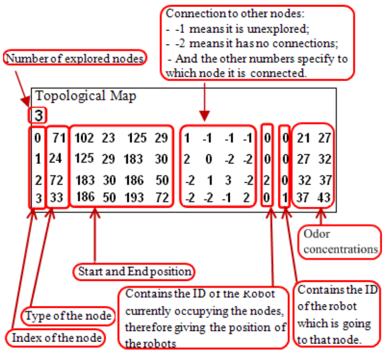
2.3.4 Path planning improvement for exploration enhancement

Robots' motion is controlled through potential fields. This technique of motion planning avoids static obstacles very well, but in situations that two or more moving robots are facing to each other in narrow corridor, this technique makes the movements very slow and the performance of the mission gets low. In order to improve the exploration process, this thesis presents some control rules to prevent this kind of situations. These rules are shown in Algorithm 3. In this algorithm, each robot considers a virtual bubble around itself. Once another robot enters to this bubble, this robot checks the probability of direct collision between them. The policy is to give priority to one of them, so one of them stops and the other passes through the area. If one of the robots is located in the already explored area, the one that is exploring a frontier will have higher priority. But if both are in the same conditions the one that has lower robot ID has the priority (Algorithm. 3).

2.3.5 Map merging

Consider several robots navigating in an environment and each one has its own coordinate system. Their X and Y axes do not match with each other and even they do not know where the reference point of the other's localization system is. Each robot is generating its own topological map of visited local area. Topological maps provide a brief characterization of the navigability of a structured environment, and, with measurements collected during exploration, the vertexes of the map can be embedded in a metric space [HB05]. These maps use a graph to represent possibilities for navigation through an environment and need less memory than their metric counterparts. The robots repeatedly send these local self generated topological maps to each other. The submaps generated by individual robots contain translation and orientation errors. The problem is how each one of them can integrate the data coming from the others to its local map and generate a more complete map. Fig. 2.11 shows an example of this problem.

Whenever a robot finds a new feature in the environment, it adds this feature as a new node to its local map and sends a message to all the other robots and reports the new map. If a robot is walking inside already explored environment and gets into another feature that is different from its previously detected feature, it modifies this feature in its local map and again sends a message to all the other robots and reports the modifies map. On the other hand, each robot has a running memory-resident program that always is listening to the network and receives all the messages that are sent by the other robots. When a robot receives a message that shows another robot has found or modified a feature in its map, it starts the hypothesis building process by creating the list of all vertices in the local topological map that are structurally compatible with the new map (conditions (4) and (5)). Vertices are tested for compatibility by examining their attributes: exactly known attributes (e.g., vertex type) must match perfectly; inexactly known



Connection to other nodes:
- -1 means it is unexplored;
- -2 means it has no connections;
- And the other numbers specify to which node it is connected.

Number of explored nodes

Topological Map

3

0	71	102	23	125	29	1	-1	-1	-1	0	0	21	27
1	24	125	29	183	30	2	0	-2	-2	0	0	27	32
2	72	183	30	186	50	-2	1	3	-2	2	0	32	37
3	33	186	50	193	72	-2	-2	-1	2	0	1	37	43

Odor concentrations

Start and End position

Contains the ID of the Robot currently occupying the nodes, therefore giving the position of the robots

Contains the ID of the robot which is going to that node.

Type of the node

Index of the node

Figure 2.10: An example of topological map data.

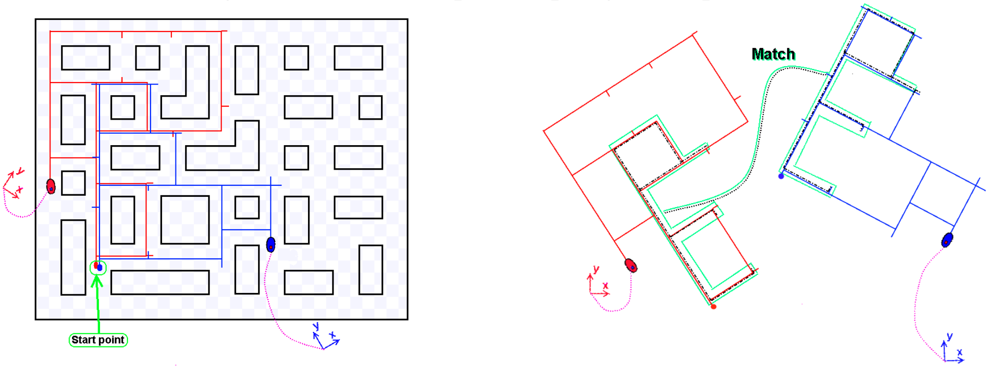


Figure 2.11: An environment being explored by two robots with different coordinate systems (left). Matching generated maps of two robots exploring the environment (right).

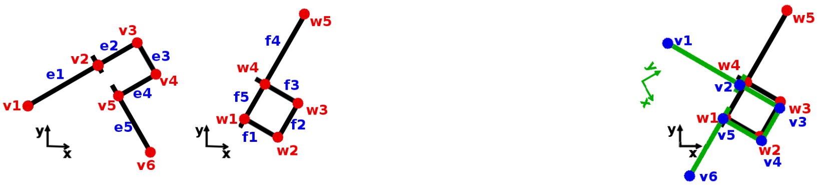


Figure 2.12: Two maps with different coordinate systems (left). The result of merging maps (right).

attributes (i.e., due to measurement error) must be compared with a similarity test. This task is computationally intensive since it requires an exhaustive search of data. In this case the robot needs a high processing resource to analyze the maps and merge them together. This thesis provides two solutions for this problem; (i) sequential algorithm, (ii) parallel algorithm using roboti clusters concept [MCM12].

Sequential solution

Algorithm 4, $Merge\_seq$, works on two given maps $M_1 = [V, E, k_1]$ (where V is the set of vertexes V_i , E is the set of edges e_i and k_1 (called size) is the number of edges of the map) and $M_2 = [W, F, k_2]$ and finds the largest common connected subgraph between these two and calculates the best transformation function that maps M_1 on M_2. This transformation function matches the coordinate reference frames of the two maps. Therefore, the final merged map is derived by overlapping only the vertexes' positions using the transformation function.

This thesis defines the stated terms in the algorithms as following:

- "$V\_equiv(v, w, F)$": checks if vertexes v and w are equivalent. Two vertexes are equivalent if their environmental features match with each other and their positions can be approximately matched by function F.

- "$E\_equiv(e_i, f_j)$": Two edges e_i and f_j are equivalent if their lengths are approximately equal, i.e.
  $Length(f_j) - \Delta_T < Length(e_i) < Length(f_j) + \Delta_T$
  and their connecting vertexes are equivalent. Δ_T is a distance threshold that should be defined based on the model of robots' localization error.

- "Transform_func(V_i,W_j,V_k,W_l)": calculates a transformation function that maps vertex V_i on W_j and vertex V_k on W_l. Having the positions of two vertexes in map M_1 and two vertexes in map M_2, with a geometric calculation a transformation function can be simply extracted that contains translation and rotation parameters.

- "$Adj(e_i)$": Returns the two vertexes of edge e_i.

- "Index(Adj(e_i))": Returns the indexes of two vertexes connected to edge e_i.
- "$Link(V_{IG1}, M, F, N)$": Finds the Nth connected vertex to V_{IG1} in map M, considering the rotation parameter in function F.
- "Max Node Deg" is the maximum degree of vertexes in the maps, i.e., the maximum number of edges connected to a vertex.

Algorithm 4: Merge_seq(M_1,M_2,MAP)

```
inputs:
        $M_1 = [V, E, k_1]$
        $M_2 = [W, F, k_2]$
Output: A map that is the result of merging $M_1$ and $M_2$
begin
    LIST = Null
    for $i = 1; i < k_1; i++$ do
        for $j = 1; j < k_2; j++$ do
            if $E\_equiv(e_i, f_j)$ then
                $(I_1M_1, I_2M_1)$ = Index(Adj($e_i$))
                $(I_1M_2, I_2M_2)$ = Index(Adj($f_j$))
                $F$ = Transform_func($V_{I_1M_1}, W_{I_1M_2}, V_{I_2M_1}, W_{I_2M_2}$)
                L_equ_v = Null
                $S$ = Com_subgraph($M_1, M_2, F, I_1M_1, I_1M_2, 0$)
                if $S > 0$ then
                    add $[F, S, L\_equ\_v]$ to $LIST$
                $F$ = Transform_func($V_{I_2M_1}, W_{I_1M_2}, V_{I_1M_1}, W_{I_2M_2}$)
                L_equ_v = Null
                $S$ = Com_subgraph($M_1, M_2, F, I_2M_1, I_1M_2, 0$)
                if $S > 0$ then
                    add $[F, S, L\_equ\_v]$ to $LIST$
    $I$ = the index of largest S saved in $LIST$
    if $I == Null$ then
        return(Null)
    $F_I$ = Transformation function saved in $LIST(I)$
    $L\_equ\_V_I$ = List of equivalent vertexes saved in $LIST(I)$
    $M1_{mapped} = F_I(M_1)$
    MAP = Save_overlap($M1_{mapped}, M_2, L\_equ\_V_I$)
    return(MAP)
end
```

Algorithm *Merge_seq* compares all the edges in M_1 and M_2 one by one. Once an edge e_i in M_1 is found equivalent of an edge f_j in M_2, a transformation function F will be calculated that maps e_i on f_j. Then the algorithm 5, *Com_subgraph*, finds the largest common connected subgraph between the two maps starting from the edge e_i and f_j to check the compatibility of

Algorithm 5: Com_subgraph($M_1, M_2, F, IG1, IG2, size$)

```
inputs:
        $M_1$ = [$V, E, k_1$], $M_2$ = [$W, F, k_2$]
        $F$ = The transformation function that maps $M_1$ on $M_2$
        $IG1$ = the index of the current under process vertex of $M_1$
        $IG2$ = the index of the current under process vertex of $M_2$
        size = the current size of common subgraph
Output: Size of the largest common subgraph
begin
    if !(V_equiv($V_{IG1}$, $W_{IG2}$, $F$) then
        return 0
    if [$IG1$,$IG2$] $\notin$ L_equ_v then
        L_equ_v = L_equ_v $\cup$ {[$IG1$,$IG2$]}
        size ++
        for $N$ = 1; $N$ < Max Node Deg ; $N$++ do
            $IG1$ = Link($V_{IG1}, M_1, F, N$)
            $IG2$ = Link($W_{IG2}, M_2, 1, N$)
            if $\exists$ $V_{IG1}$ & $\exists$ $W_{IG2}$ then
                size = Com_subgraph($M_1, M_2, F, IG1, IG2, size$)
            if size == 0 then
                return 0
    return(size)
end
```

the vertexes based on the transformation function F. Since each edge has two vertexes, for every common edge, the algorithm 5 is applied two times; once assuming that the first vertex of e_i is matched with the first vertex of f_j and once assuming the second vertex of e_i is matched with the first vertex of f_j.

Com_subgraph is a recursive algorithm that creates lists of corresponding vertexes between the two maps (each list is called a *hypothesis*) by locally expanding single vertex matches. It starts with the starting vertexes, compares them, and if they are compatible it finds the next connected vertex to these vertexes and runs the algorithm with the new found vertexes. If the algorithm finds any conflict (mismatch), it returns zero indicating that matching e_i and f_j using function F will lead to a conflict in the rest of map edges and this match should be ignored. Otherwise, if there were no mismatch during the process of *Com_subgraph* algorithm, it saves the list of compatible vertexes in *L_equ_v* and returns the number of the found common connected subgraph (one hypothesis). Algorithm *Merge_seq* saves the data (size, matched vertexes and the transformation function) of the generates hypothesis in *LIST* and then

Table 2.2: List of similar edges and processed vertexes in algorithms 4 and 5 for the maps shown in Fig. 2.12.(left). Mismatches are bolded in this table.

Edges	Processed vertexes	Edges	Processed vertexes
$e_1 - f_4$	$\mathbf{V_1W_4}$	$e_3 - f_2$	$V_3W_2, \mathbf{V_2W_1}$
$e_1 - f_4$	$V_1W_5, V_2W_4, \mathbf{V_3W_1}$	$e_3 - f_2$	$\underline{V_3W_3, V_4W_2, V_5W_1, V_2W_4}$
$e_2 - f_1$	$\mathbf{V_2W_1}$	$e_4 - f_1$	$\underline{V_4W_2, V_5W_1, V_3W_3, V_2W_4}$
$e_2 - f_1$	$\mathbf{V_3W_1}$	$e_4 - f_1$	$\mathbf{V_4W_1}$
$e_2 - f_2$	$\mathbf{V_2W_2}$	$e_4 - f_2$	$V_4W_2, \mathbf{V_5W_3}$
$e_2 - f_2$	$V_3W_2, \mathbf{V_4W_1}$	$e_4 - f_2$	$V_4W_3\mathbf{V_5W_2}$
$e_2 - f_3$	$\mathbf{V_2W_3}$	$e_4 - f_3$	$V_4W_3, \mathbf{V_5W_4}$
$e_2 - f_3$	$\underline{V_3W_3, V_4W_2, V_5W_1, V_2W_4}$	$e_4 - f_3$	$\mathbf{V_4W_4}$
$e_2 - f_5$	$\mathbf{V_2W_1}$	$e_4 - f_5$	$\mathbf{V_4W_1}$
$e_2 - f_5$	$\mathbf{V_3W_1}$	$e_4 - f_5$	$\mathbf{V_4W_4}$
$e_3 - f_1$	$\mathbf{V_3W_1}$	$e_5 - f_4$	$\mathbf{V_5W_4}$
$e_3 - f_1$	$V_3W_2, \mathbf{V_4W_1}$	$e_5 - f_4$	$\mathbf{V_5W_5}$

processes the next edges in M_1 and M_2. Finally, the *LIST* will contain all common subgraphs that can be generated between M_1 and M_2 with different transformation functions. Algorithm *Merge_seq* finds the biggest hypothesis in the *LIST* and generates $M1_{mapped}$ based on its transformation function. Eventually, $M1_{mapped}$ and M_2 have common coordinate reference frames and we have the list of matched vertexes between these two, merging these two maps can be easily done by overlapping them on each other and generate a new map.

Table 2.2 shows how the algorithm *Merge_seq* and *Com_subgraph* work on two example maps of Fig. 2.12.(left). Starting from e_1 and f_1, equivalent edges are $(e_1 - f_4)$, $(e_2 - f_1)$, $(e_2 - f_2)$, etc. These edges are listed in the first column of table 2.2. Algorithm *Com_subgraph* compares the vertexes of the two maps, starting from the ones that are connected to the equivalent edges, that generates hypotheses. The second column of table 2.2 has listed the vertexes that are compared for each pair of edges. The mismatches are colored red. For instance, processing equivalent edges $(e_1 - f_4)$, algorithm *Com_subgraph* considers the first hypothesis which includes V_1,W_4. Since V_1 is of degree one and W_4 is of degree four, they do not match, so the algorithm does not continue processing this hypothesis and returns zero (V_1W_4 is colored red in the table to indicate the mismatch). Processing current equivalent edges $(e_1 - f_4)$, the algorithm takes next hypothesis into account which starts with V_1,W_5.

These vertexes (V_1 and W_5) are compatible so the next vertexes V_2 and W_4 will be compared. Since these two are also compatible the next vertexes V_3 and W_1 are compared that are incompatible (so we have $V_1W_5, V_2W_4, \mathbf{V_3W_1}$ in the table). Finally algorithm 4 will have the list of all hypotheses saved in $LIST$. For the mentioned example above, $LIST$ contains three hypotheses because only those which do not lead to any mismatch are saved in $LIST$. These hypotheses are underlined in table 2.2. Based on the biggest found hypothesis (or one of the biggest ones), algorithm $Merge\_seq$ will map the M_1 with the associated transformation function to that hypothesis, this action will overlap M_1 on M_2 (Fig. 2.12.(right) shows this overlap). Finally the complete map is driven by overlapping matched submaps.

Parallel solution

We provide a parallel solution for the topological map merging problem using message passing interface (MPI) standard. This is the first parallel implementation of topological map merging problem. Algorithm 6 demonstrates the pseudo-code of the proposed algorithm for parallel topological map merging. The parallel part of this algorithm, lines 8 to 30, is run on multiple robots of the cluster and the rest of the code is run serially only on the robot which has initially started the program.

I. Load distribution: Each robot individually makes its decision to connect or to establish a cluster or exit from it [MCM12]. When a robot needs to distribute a task between the other robots in the cluster, it first should know how many robots have joined to the cluster and then distribute the load between them.

Algorithm 6 dynamically distributes the load between the available robots in the cluster in real time. This algorithm first finds the number of available operational nodes in the cluster (by MPI_comm_size in line 10) and then it distributes the load between them. Therefore, if one or more nodes in the cluster have failed or have lost the communication connection, algorithm 6 distributes the load only between the operational nodes. However, if after distributing the load, one of the robots fails (or loses the connection), the robot which has initially distributed the task will get a timeout (i.e., an MPI error message) instead of the results. The algorithm returns the error (lines 33 and 34 of algorithm 6) and this robot has to start over running the algorithm, so

Algorithm 6: Merge_parallel(M_1,M_2,MAP), MPI solution

```
inputs:
        $M_1 = [V, E, k_1]$
        $M_2 = [W, F, k_2]$
Output: A map that is the result of merging $M_1$ and $M_2$
begin
    MPI_Init(...)
    MPI_Barrier(MPI_COMM_WORLD)
    begin
        MPI_comm_rank (MPI_COMM_WORLD, &id)
        MPI_comm_size (MPI_COMM_WORLD, &p)
        $Low\_value = (id - 1) \times k_1 / p + 1$
        $High\_value = (id) \times k_1 / p$
        LIST = Null
        for $i = Low\_value; i < High\_value; i++$ do
            for $j = 1; j < k_2; j++$ do
                if $E\_equiv(e_i, f_j)$ then
                    $(I_1M_1, I_2M_1)$ = Index(Adj($e_i$))
                    $(I_1M_2, I_2M_2)$ = Index(Adj($f_j$))
                    $F$ = Transform_func($V_{I_1M_1}, W_{I_1M_2}, V_{I_2M_1}, W_{I_2M_2}$)
                    L_equ_v = Null
                    $S$ = Com_subgraph($M_1, M_2, F, I_1M_1, I_1M_2, 0$)
                    if $S > 0$ then
                        add $[F, S, L\_equ\_v]$ to $LIST$
                    $F$ = Transform_func($V_{I_2M_1}, W_{I_1M_2}, V_{I_1M_1}, W_{I_2M_2}$)
                    L_equ_v = Null
                    $S$ = Com_subgraph($M_1, M_2, F, I_2M_1, I_1M_2, 0$)
                    if $S > 0$ then
                        add $[F, S, L\_equ\_v]$ to $LIST$
    end
    MPI_Bcast (&LIST, ...)
    if $id == S_{id}$ then
        MPI_Reduce (&LIST,...)
        if $!MPI\_SUCCESS$ then
            return($MPI\_ERROR$)
        $I$ = the index of largest S saved in $LIST$
        if $I$ == $-1$ then
            return(NULL)
        $F_I$ = Transformation function saved in $LIST(I)$
        $L\_equ\_V_I$ = List of equivalent vertexes saved in $LIST(I)$
        $M1_{mapped} = F_I(M_1)$
        MAP = Save_overlap($M1_{mapped}, M_2, L\_equ\_V_I$)
        return(MAP)
end
```

that it will distribute the load between the current available robots again.

The main source of potential parallelism in this application is data decomposition i.e., dividing problem search space between the robots. To automatically divide the search space between the robots (without having a central base station), this algorithm uses the robots' IDs as unique indicators. In algorithm 6 all of the processing nodes (robots) have complete data of two maps of M_1 and M_2 and each robot finds the biggest common subgraphs between the two maps that starts in its share of data. Each robot should work on a block (that is a division of whole search space) specified by two variables named *Low_value* and *High_value*. Low_value_i and $High\_value_i$ are computed in robot i based on the following formulas:

$$Low\_value_i = (id_i - 1) \times \frac{k_1}{p} + 1 \tag{2.4}$$

$$High\_value_i = (id_i) \times \frac{k_1}{p} \tag{2.5}$$

where id_i denotes the ID of robot i in the cluster, p is the total number of nodes and k_1 is the size of map M_1. To explain these formulas, we provide an example; imagine the map size k_1 is 1000 and the *current* number of available robots p is 5. Based on (2.4) and (2.5), for the robot whose *id* is 1, Low_value_1 and $High\_value_1$ are respectively 1 and 200, and for the robot with id = 2, Low_value_2 is 201 and $High\_value_2$ is 400. Thus, Robot 1 works on interval [1,200], robot 2 on interval [201,400], robot 3 on [401,600] and so on. If the current number of robots was 10, the share of each robot would be intervals with length 100.

Using function MPI_comm_rank() in the algorithm the robots find their own ID and calling function MPI_comm_size() they will be aware of the total number of robots (p) participating in the cluster at the moment. It is worth to mention that in MPI programming, the variables are private, i.e., changes in a variable in a node (inside the parallel part of code) does not have any effect in the value of the same variable in another node.

By considering k_1 (and not k_2) in (2.4) and (2.5), map M_1 is divided between the robots in equal shares. Now each node runs the previously presented sequential algorithm but starting from its given share of data in M_1 and finds the biggest common subgraphs between M_1 and M_2 (see algorithm 6).

II. Functionality: Similar to algorithm 4, algorithm $Merge\_parallel$ compares all the edges in a block of M_1 and M_2 one by one. Once an edge e_i in M_1 is found equivalent with an edge f_j in M_2, a transformation function F will be calculated that maps e_i on f_j. Then the algorithm 5, $Com\_subgraph$, finds the largest common subgraph between the two maps starting from the edge e_i and f_j to check the compatibility of the vertexes based on the transformation function F. Each robot saves its found common subgraphs into its $LIST$. Finally, the $LIST$s will contain all common subgraphs that can be generated between M_1 and M_2 with different transformation functions. These $LIST$s are generated by different robots and need to be aggregated to one node and be processed.

III. Aggregating the results: MPI_Bcast is called in the algorithm, so that each robot will broadcast its results ($LIST$) to the cluster. In this algorithm, the robot with $id = S_{id}$ is responsible to aggregate the distributed results and calculate the final result. S_{id} denotes the id of the robot which initially requested the map merging process. This robot runs MPI_Reduce to gather the results of processed data from the other computers to its own $LIST$. Based on the biggest found common subgraph (hypothesis), algorithm $Merge\_parallel$ will map the M_1 with the associated transformation function to that hypothesis and generates $M1_{mapped}$. Finally the output map (MAP) is generated by overlapping $M1_{mapped}$ and M_2.

2.3.6 Localization correction

Robots' localization is a key issue in multi-robot mapping and exploration. This work doesn't consider a global positioning system. The only tool that the robots have for determining their position is their odometry. However, the odometry is unreliable because of uneven floors and wheel slippage. It is therefore necessary to increase localization accuracy by measuring position of the robot relative to known objects in the environment.

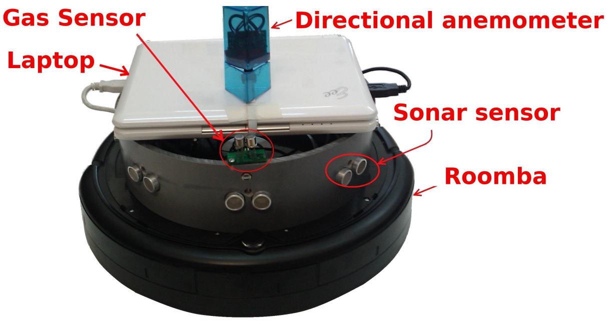
Normally odometry errors accumulate incrementally as the robot is traveling, therefore, the robot's localization is more accurate at the beginning of an experiment and loses its accuracy during the test. If the robot enters in an environmental feature that has already been explored, it can look at the map and find the start position of that feature, then correct its localization based on the data that has been stored in the map. It does not matter if this feature was added to the map by the current robot or by another robot in the team, since the feature has been added to the map in the past; it means that the location that was saved in the map is more reliable than the current localization of the robot. This method is only used when the robot is passing an area that has already been explored. In addition to correcting odometry errors using map's data, the robots are able to correct their internal odometry angle based on the features of the environment. Since in usual structured buildings, all the corridors are parallel or perpendicular to the main direction, similar to [BY10], when the robot detects a corridor and is moving along it, the robot is able to correct the localization angle. This method is used in the unknown area as well as in the already explored environment.

2.4 Experimental results

The proposed method has been tested and validated both in real world and in simulation. The functionality of the method in searching and exploration in a structured real environment has been verified. Afterwards, in order to evaluate the method in more complex environments, the method has been tested in simulation. We divide the experimental results in two parts; multi-robot olfactory search results and multi-robot mapping result.

2.4.1 Experiments in structured environments

The proposed method was tested in different reduced scale maze-like environments, like the one shown in Fig. 3.43. This testing arena, with $3 \times 4m^2$ area by 0.5 meters height, has controlled ventilation through a manifold that extracts air from the testing environment through a honeycomb mesh integrated into one of the walls. The opposite surface of the environment contains a similar mesh that allows the entrance of clean air that flows through the environment. Controlled gas sources are simulated with ethanol vapor, generated using bubblers and pumped to different places of the environment through a set of Polyvinyl chloride (PVC) tubes. Fig. 2.13.(right) shows a chemical release mechanism that is used as odor source in the experiments. Inside this environment, a maze like structure is built to simulate real structured buildings. This environment is consist of corridors, crosses, corners and dead-ends.



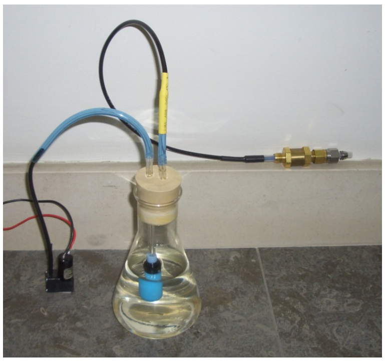


Figure 2.13: Left: iRobot Roomba robot equipped with a laptop and sonar, gas, and anemometer sensors. Right: the chemical release mechanism

The robots

The iRobot[1] Roomba was used in the experimental tests. This is an attractive platform because it is inexpensive, readily available and can be fully monitored and commanded through a serial port interface. In the current work, a set of Roombas were upgraded with a small laptop computer (ASUS Eee PC 901) running a Linux based operating system and the Player[2] environment. The computer interfaces through a micro-controller board with a set of five sonars, three 2-D anemometers and a gas sensing board (Fig. 2.13.(left)). Based on values measured by sonar sensors, the robots recognize the basic environmental features (e.g., corridors, dead-ends, etc.). The gas concentration was measured with a custom sensing board based on metal oxide gas sensors (Figaro[3] TGS2620). A triangular anemometer was designed based on three self heated NTC[4] sensors placed around a triangular prism and processing the raw measurements with a method similar to the one previously described in order to estimate the wind direction [MdA06b]. The Player/Stage driver for Roomba robots makes it possible to run the same code either in simulation or on the real Roomba robots.

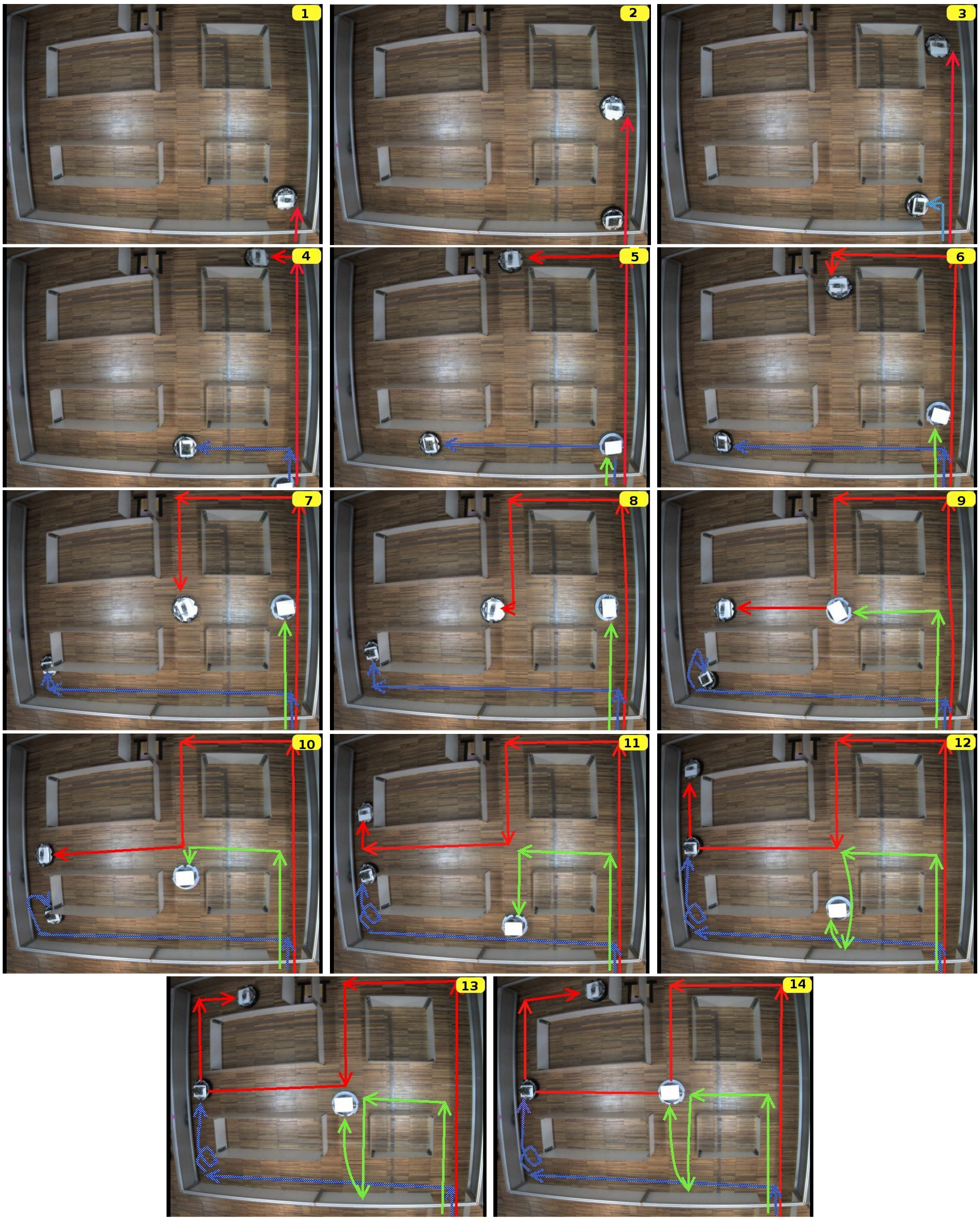


Figure 2.14: Three robots exploring a gas free environment.

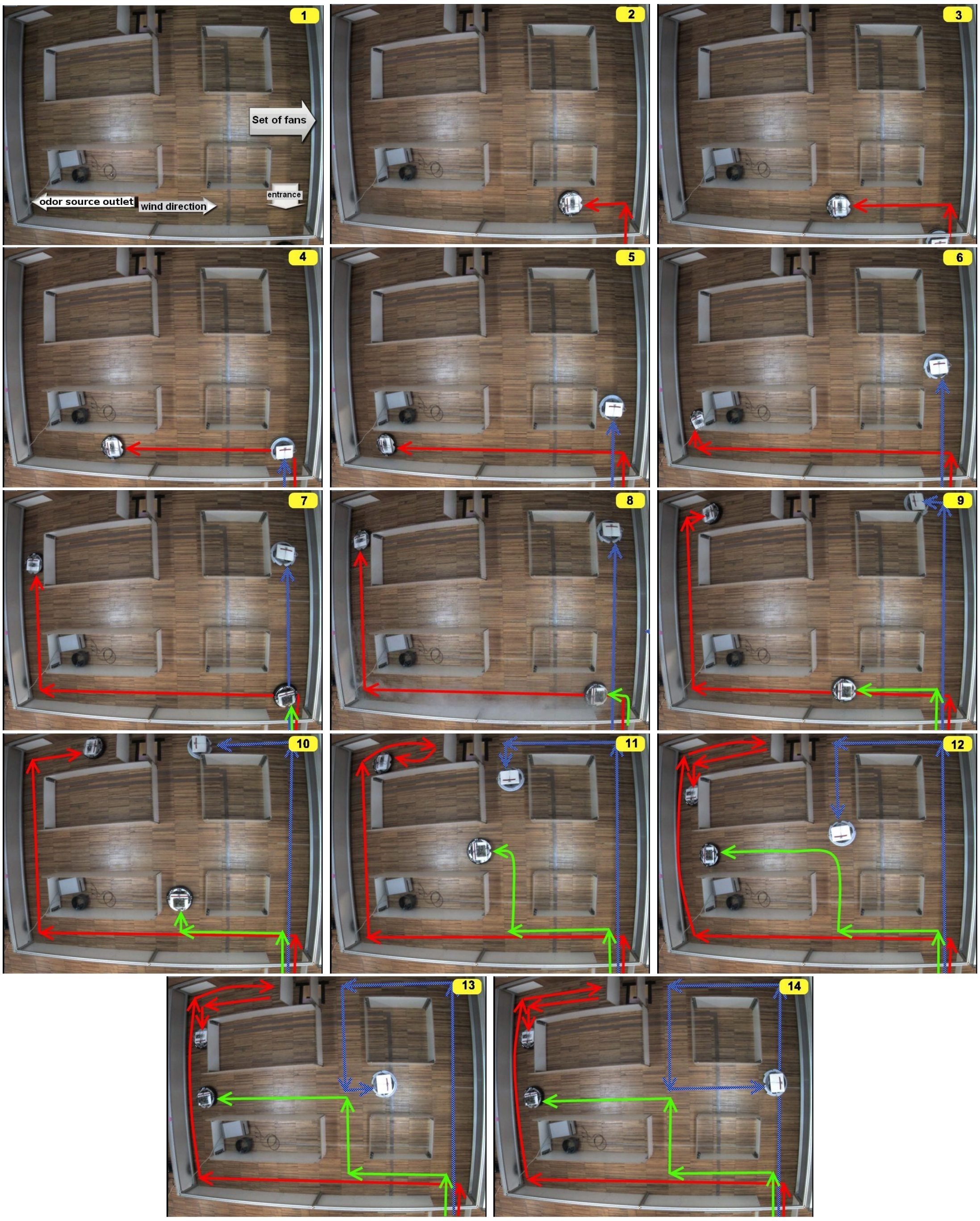


Figure 2.15: Three robots exploring the environment and finding the odor sources

Experimental validation

In order to evaluate the method, the algorithm was experimented in the previously described environment with one, two and three Roomba robots separately, first without having any odor source, and then with an odor source releasing gas in the left-bottom corner of the arena at eight centimeters height, as shown in Fig. 2.15.1. The robots have no a-priori knowledge about the number of odor sources existing in the environment. When there is no odor source in the environment, all frontiers have the same *utility* and the search algorithm acts like a pure frontier based exploration algorithm that is enhanced by avoiding collisions between the robots. The method is validated by comparing the results and analyzing the behavior of the robots.

Fig. 2.14 shows three robots exploring a small maze while searching for odor sources. In this experiment there were no odor sources. All robots started from the same point but not at the same time and were run a few seconds after each other (like if they were deployed at the entrance of a warehouse). The red footprint shows the first robot's path, the blue footprint is related to the second robot and the green shows the footprint of the third robot. These experiments show the functionality of algorithm in different maze structures and with different number of robots. Frame 3 of Fig. 2.14 shows an example of coordination between the robots; when the second robot reached the junction it figured out that the path in the front was already explored so it chose the left path. In frame 5, the third robot had to decide to either take the front or the left path, since both paths were already explored, it checked the nearest unexplored and unassigned frontier in the map that was provided by the other robots and finally chose the front path. "Path planning improvement" is presented in frames 9 and 10, when the second robot goes backward for a while to avoid possible collisions with the first robot. Fig. 2.16 shows a topological map generated by the robots in one experiment over the real topology of the testing environment. The red lines and vertexes demonstrate the generated topological map before implementing the proposed localization correction method, and the green shows a map after applying the localization correction technique. The effect of localization correction technique is significant,

[1] http://www.irobot.com

[2] http://playerstage.sourceforge.net

[3] http://www.figarosensor.com

[4] Negative Temperature Coefficient

e.g., all the lines of the green map are parallel to the corridors. The robots correct their odometry during the mission so the maps' errors do not accumulatively increase. The maps out of several tests were similar, however they did not match perfectly. In these experiments the algorithm 4 was used to merge the maps since the size of topological maps in these scenarios were small. Section 2.4.3 evaluates the presented parallel solution of map merging (algorithm 6) in details.

Fig. 2.15 shows the tests done with the same maze structure and the same robots, but with an ethanol odor source on the left side of the environment. The first robot in the first branch decided to go to the left-way because it sensed an odor clue and the wind was from that direction (red footprints). The other robots also automatically changed their path in accordance with the new conditions. The results show the effect of odor concentration on the behavior of the robots. In each test the exploration time, that is the time it takes to cover all the maze paths, is measured. Fig. 2.17 shows that the complete exploration time is higher when considering gas cues, however the difference is less than 21 percent. Fig. 2.18 shows the time to reach the target (the location of the odor source) in these two scenarios. The chart shows that the robots reach the target much faster with having gas cues rather than without having it, which shows the functionality of the algorithm. Each result is the average of five similar tests. Different tests with constant conditions had similar results with about eight percent variance. The maximum speed of the robots was kept constant in the tests. More experiments were done by placing the odor source in other locations of the arena and the results were similar to the graph in Fig. 2.18.

The robots were initially released at the downwind side of the testbed (similar to many other previous works namely [LANM10, MAdA03]) in the experiments. This is the same task that the handlers of SaR dogs do at the start of the search mission. In fact if we position the robots on the other side of the environment (upwind side) they will not be able to sense the odor and therefore they will explore the environment until they eventually get into the active odor plume area. The same point is valid if there is no air-flow in the environment. In this case the robots mostly explore the environment rather than tracking odor plumes, Although they always intend to explore the frontiers with higher odor concentration first.

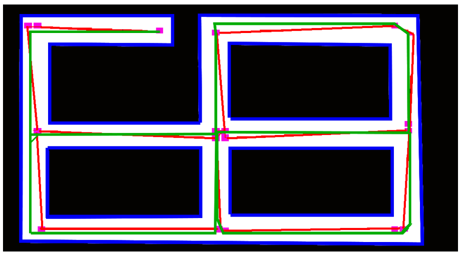


Figure 2.16: Generated maps for the environment in Fig. 3.43, red: without localization correction, green: with localization correction.

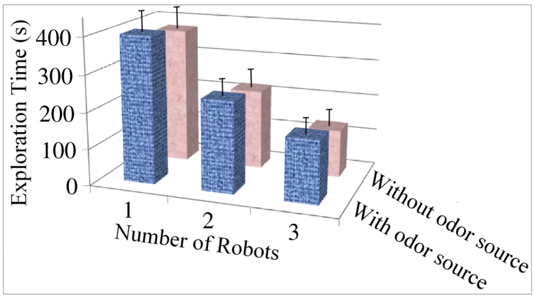


Figure 2.17: Exploration time, with/without odor source.

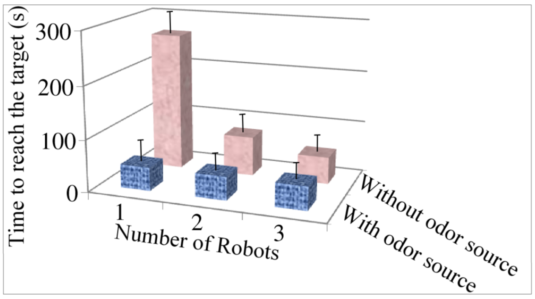


Figure 2.18: Reaching the target (the location of the odor source), With/without olfactory cues. Without odor source the method acts like a pure frontier based exploration algorithm.

2.4.2 Olfactory search performance assessment

For evaluating the exploration algorithm and measuring its performance, the Player/Stage framework [GVH03] was used, since in the real world, there are several constraints that do not allow easily testing the proposed method. The Player/Stage framework provides a very easy way of testing and monitoring the proposed algorithm in different configurations. Most of the parts of proposed method including decision-making, feature extraction, robots' motion, localization correction, path planning improvement, and map merging are imported to the simulation with a few minor modifications. Only some parts related to sensory interfaces are totally changed in the simulations. Regarding the world model, in compare with real-world experiments, bigger and more complex environments are tested in the simulations.

PlumeSim [CSM10] is used for simulating odor plumes inside Player/Stage. It is capable of feeding simulated chemical plumes into Player/Stage from a wide variety of sources, from analytical models to data generated by CFD software. In this work the model of testing environment that is presented in Fig. 2.19 was given to ANSYS Fluent CFD [ANS06] software to simulate odor sources and provide odor concentration data. 3-D ANSYS Fluent is used to simulate the odor plume, however, the robots move on the 2-D floor with gas sensors at 10 centimeters height, so, only the airflow and concentration measured by the robots at that height are relevant for their decisions. Fig. 2.20 shows four snapshots of 3-D plumes propagation during the time in one of the tested scenarios. The 2-D data is given to the PlumeSim simulator in Player/Stage and is shown in Fig. 2.21. This environment contains three simulated odor sources in three corners of the arena (the other corner is the entrance door). The airflow was ventilated from the inlet side (left) with constant speed of 0.5 m/s. Since the environment contains several obstacles, the flow velocity varies in different parts of the environment. For example in the left-top corridor, the air has lower velocity relatively to the bottom corridor since there is an obstacle in its front. As it is shown in the simulation snapshots of Fig. 2.20, the odor propagation directly depends on the airflow velocity.

Fig. 2.22 shows the average results of five tests with one, two and three robots. A conclusion from Fig. 2.22 is that having more robots is more advantageous in finding more odor sources. Similar to many multi-agent systems, reduction of operation time by using more

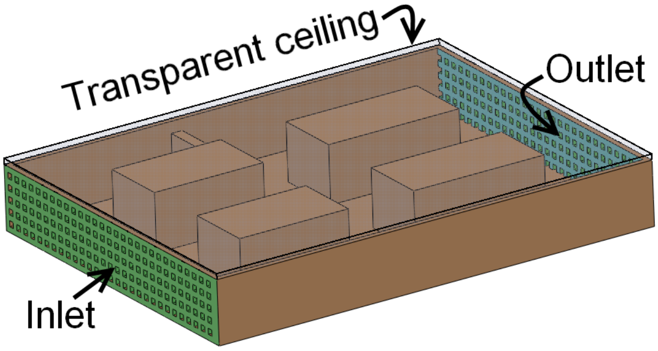


Figure 2.19: Model of a testing environment in simulations

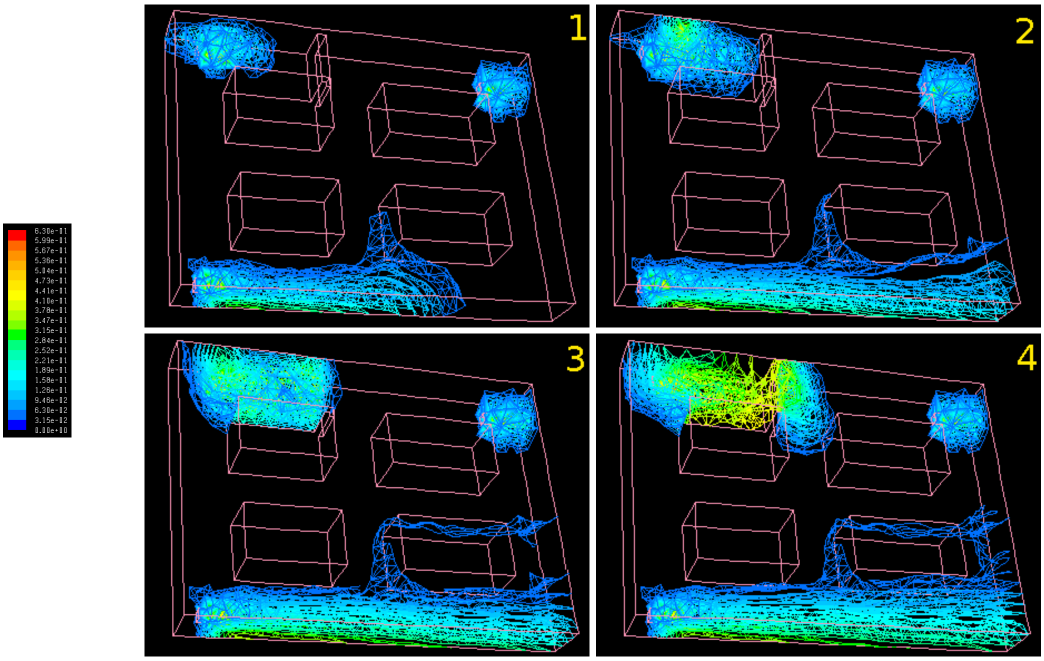


Figure 2.20: ANSYS Fluent simulation results; contours of mass fraction of ethanol propagated in the testing environment of Fig. 2.19 during the time.

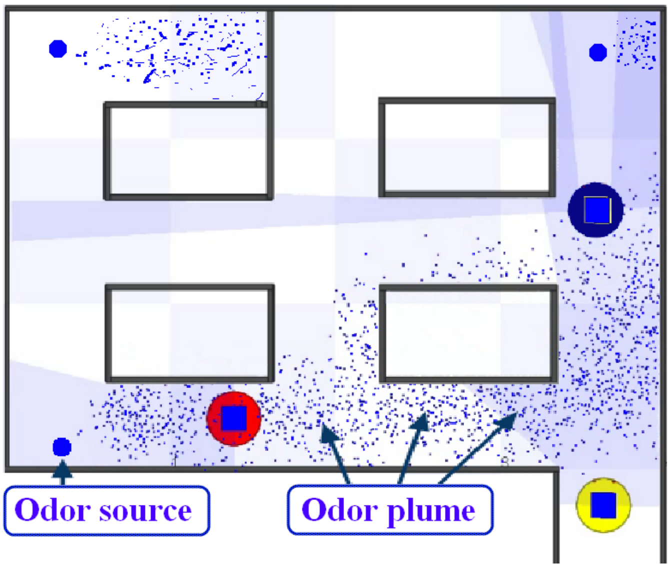


Figure 2.21: Robots searching for simulated odor sources in Player/Stage

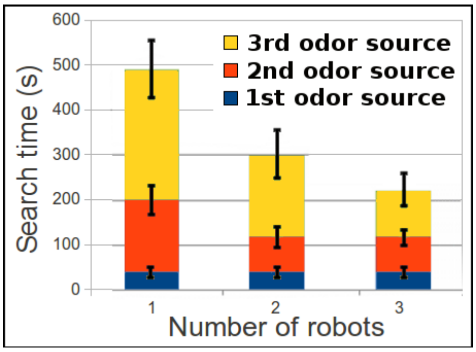


Figure 2.22: Results for different number of robots finding multiple sources

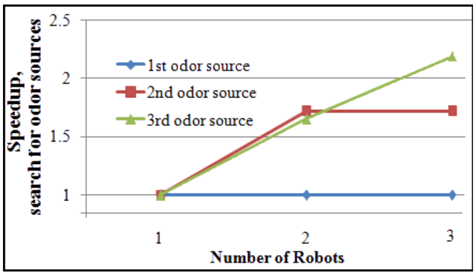


Figure 2.23: Speedup for different number of robots finding multiple sources

robots is calculated by speed up formula as bellow:

$$\text{Speedup} = \frac{T_{single}}{T_{multi}}$$

where:

- T_{single}: Operation time of the method using one robot.
- T_{multi}: Operation time of the method using several robots.

In Fig. 2.23, it can be seen that speedup is more for more odor sources, i.e. having more robots in the operation will result in better search time in looking for more odor sources.

The worst case in the search operations is when the robots search for odor sources but there is no odor source in the environment. In this case the robots have to cooperatively explore the whole environment without having any olfactory cue. Therefore, the method is tested in simulation in different environments without having any odor source to evaluate its "exploration" ability in complex scenarios.

One of the possible ways to measure the performance of the proposed method is to compare it with an optimal method. However, there is no optimal method for exploring an unknown world. If the robots had prior map of the environment, the exploration problem would be reduced to the standard well-known "multi-agent traveling salesman" (TSP) problem. It is very obvious that the exploration method will never be better than solution of traveling salesman problem, because the robots do not have a priori map of the environment before exploring but TSP does have the world model before the mission. However, the proposed search and exploration method are compared with the results of traveling salesman solution as an unreachable, more than perfect, optimal method. This can be a good criterion for evaluating the method.

The method was tested with different number of robots in three different mazes (see Fig. 2.24, Fig. 2.25 and Fig. 2.26). These mazes were also used to run an optimal TSP algorithm and the results are compared in Fig. 2.28. The graphs show the average of five tests for each data. The variance was less than five percent. The results show that the exploration time improves with higher number of robots. Another conclusion is that having more robots is more advantageous

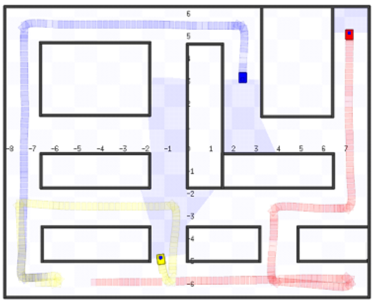


Figure 2.24: A maze with 36 nodes

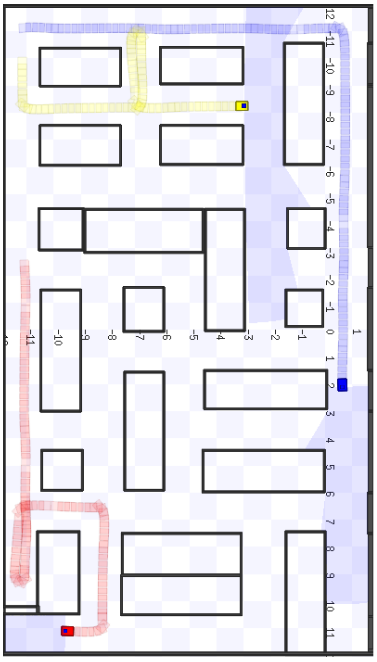


Figure 2.25: A maze with 81 nodes

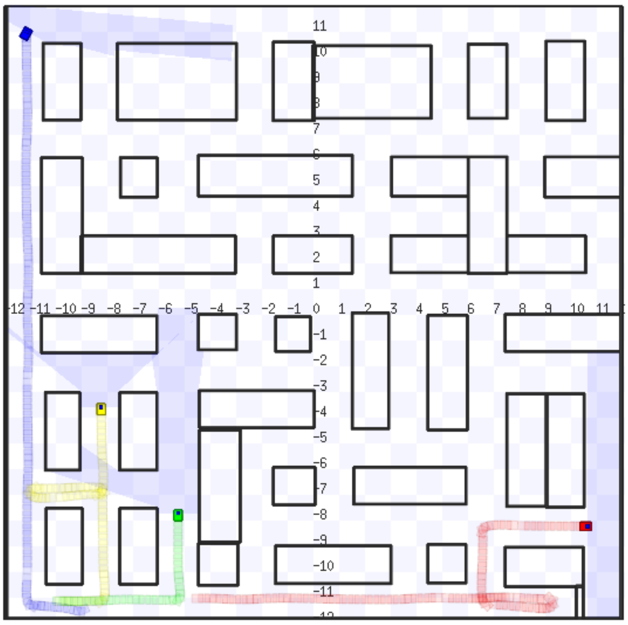


Figure 2.26: A maze with 136 nodes

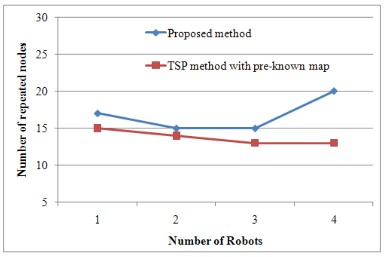


Figure 2.27: Number of repeated nodes. Comparing the results of the proposed method with TSP method in maze shown in Fig. 2.24

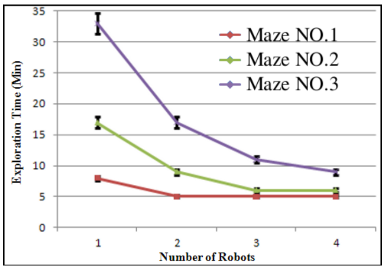


Figure 2.28: Test of various numbers of robots against complexity of the environment, 1: maze in Fig. 2.24, 2: maze in Fig. 2.25, 3: maze in Fig. 2.26

in a complex maze than in a simple maze. This also shows that the cooperation algorithm in this approach is efficiently functional.

A repeated node is a node that robots pass more than once. Another good parameter for measuring the performance of the method is counting repeated nodes since these represent losses of performance. Fig. 2.27 shows the number of nodes that have been repeated more than once in the TSP algorithm as well as in the proposed method for the maze shown in Fig. 2.24. Although Fig. 2.27 shows that the number of repeated nodes is more in the proposed method comparing with TSP, the results are acceptably comparable especially considering the fact that TSP has prior knowledge about the world but the robots in the proposed method don't.

A conclusion from Fig. 2.27 and Fig. 2.28 is that there is a trade-off between the number of robots and the size of the world since the number of repeated nodes tends to increase with the the number of robots (e.g., when the number of robots is four, for the maze shown in Fig. 2.24, the number of revisited nodes is more than when there are three robots). However, Fig. 2.28 shows that even in this case the exploration time is improved.

The algorithms and their parameters were modified and adjusted after several simulation experiments. For example, the value of k (common subgraph size) in conditions (4) and (5) is a critical issue. Based on simulation experiments, it was figured out that it is good enough to set k to eight. This means that if two robots find a common subgraph with eight vertices, they can merge their local maps together. However, this value is highly dependent on the structure of the environments. If an environment is rich of unique features this value should be less and if an environment is full of similar features this value must be higher.

The effectiveness of the algorithm was investigated as a function of the number of robots available. Ideally, it would be expected that doubling the number of agents would halve the time required for search and exploration. However, Even with a priori map information it is often not possible to achieve this amount of performance increase. The ideal number of agents for the most efficient exploration of an environment depends on the nature of the environment. The large environment of Fig. 2.26 showed greater improvements in the exploration performance when the number of agents was increased. Another significant point is that the method is fully functional even in the case of some robots' failures. If one or several robots fail during the operation the other robots will accomplish the mission. A not studied problem is when there are partially malfunctioning robots, in this case the robots might send wrong information about the environment to the others and the whole mission may fail.

Regarding olfactory search the presented method does not have any assumption regarding air flow. If there is detectable air flow in the environment the robots take this clue inside their decision makings, but if there is no air flow the search algorithm acts like a cooperative exploration method. In the later case, the robots can eventually find odor sources (if there is any) during the exploration of the environment.

2.4.3 Map merging experimental results

We have designed and implemented a robotic cluster using modified iRobot Roomba[1] robots. The iRobot Roomba robot is an attractive platform because it is inexpensive, readily available and can be fully monitored and commanded through a serial port interface. In this work, a set

[1] http://www.irobot.com

of eight Roomba robots were upgraded with small laptop computers (ASUS Eee PC 901 and ASUS Eee PC 1015PEM) running Ubuntu[1] operating system and the Player[2] environment to control the robots.

After configuring some necessary parameters (e.g., IP addresses) on the robots, an MPI software is needed to be installed. MPI is actually a library of routines that can be called usually from Fortran or C programs. There are a large number of implementations of MPI, two open source versions are LAM[3] and MPICH. LAM is an effective way for fast client-to-client communication and is portable to all Unix-based machines. MPICH is another freely available, portable implementation of MPI, a standard for message passing for distributed memory applications used in parallel computing, available for most flavors of Unix and other operating systems.

We installed MPICH libraries because it provides more programming facilities on Ubuntu and its installation is straight forward. After installing LAM or MPICH or a similar MPI library, the cluster is established, it is only needed to write and compile parallel programs on the robots. Compiling MPI programs can be done with several compilers, e.g., *mpic++*, with extended MPI libraries.

The algorithms have been run on the implemented robotic cluster using up to eight modified Roomba robots. For evaluation of the method, all algorithms are tested ten times with different sets of maps. Several topological maps were experimentally generated by real robots using the results of our past experiments in [MM11a, MNMdA09a, MM10, MNMdA10]. Fig. 2.24 and Fig. 2.26 show three simulated environments being mapped by multiple robots.

Fig. 2.29 and 2.30 demonstrate an example of the map merging experiments. In Fig. 2.29 two partial maps of one environment are shown that were generated by two robots without having common reference frames. About 10 percent of vertexes and edges of these partial maps are equivalent. Fig. 2.30 presents the result of merging these maps. The execution time for each merging experience is measured five times and their mean values are shown in Fig. 2.31.(left). These results show that if the maps are large enough (map size bigger than 54 in this case),

[1] http://www.ubuntu.com

[2] http://playerstage.sourceforge.net

[3] http://www.lam-mpi.org

execution time decreases as number of robots increase.

Parallel algorithms are usually evaluated by analyzing their speedup and their efficiency over the serial algorithms. Reduction of processing time by using more robots (speedup) can be calculated by:

$$Speedup = \frac{T_{seq}}{T_{Parallel}}$$

where:

- T_{seq}: execution time of code using one robot.
- $T_{parallel}$: execution time of parallel code using several robots

A parallel algorithm is called "efficient" if its speedup is nearly linear to number of processors [KRS90]. Fig. 2.31.(right) presents the speedup of the parallel algorithm for different size of maps. It can be seen that speedup has almost linear relation to the number of robots for all size of N in large maps, thus, proving the efficiency of the proposed parallel algorithm. This means that having more robots in the cluster increases the speed of solving this problem. It should be mentioned that, according to the *Amdahl law* [Amd67, Ale07], it is very difficult, even in an ideal parallel system, to obtain a speedup value equal with the number of processors because each program, in terms of running time, has a part that cannot be parallelized and has to be executed sequentially by one single processor.

The size, the topology and the degree of connectivity of the maps are also important effective parameters in the speed of the algorithms. Sometimes a smaller map takes more time to process than a bigger one because of the other parameters. For evaluating the methods in a more scientific way we generated several artificial maps similar to the maps of real structured environments by running a C program. The maps generated by this software are structurally similar to each other but their sizes are different. The maps are split to two submaps having about 10 percent of vertexes and edges in common. Then one of the submaps is rotated with a random angle and the results are fed to the algorithms as inputs. Fig. 2.32 shows two examples of these generated maps. Fig. 2.33 shows the merged map resulted from running the algorithms on the maps of Fig. 2.32. The execution time and speedup are measured for these

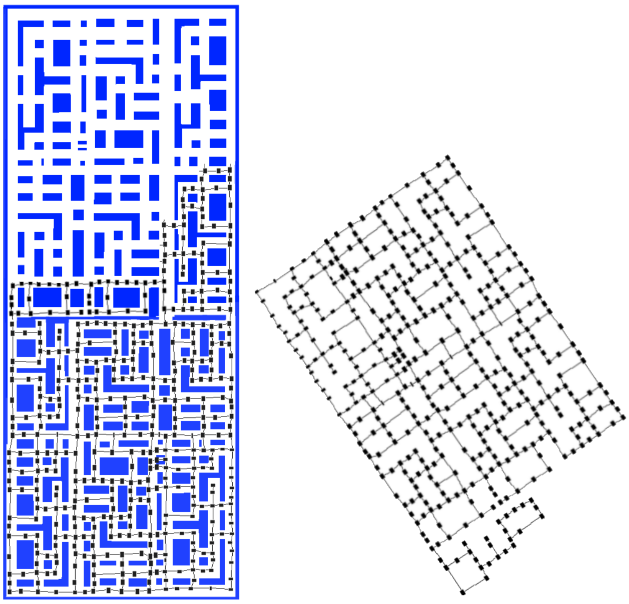


Figure 2.29: Partial maps generated by 2 robots with different coordinate systems

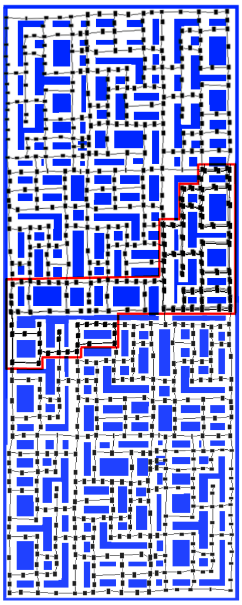


Figure 2.30: The result of map merging algorithm for the maps of Fig. 2.29

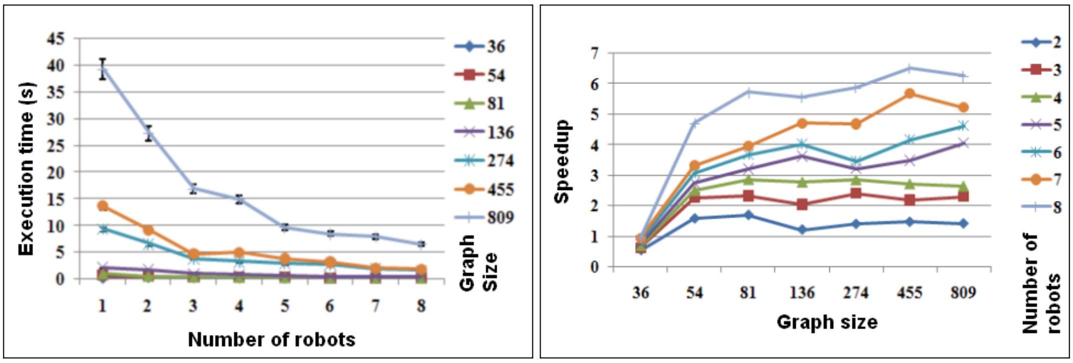


Figure 2.31: Left: Execution time results for different size of real world and simulation maps based on number of robots. Right: Speedup for different sizes of problem in the real world and simulations.

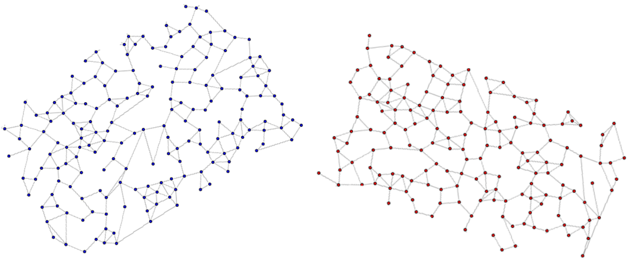


Figure 2.32: Two given artificial maps to be merged by the algorithms.

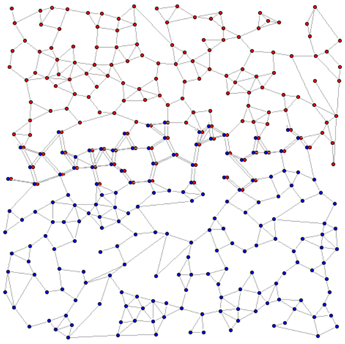


Figure 2.33: Merged map with about 1000 vertexes and 900 edges.

experiments and are reported in Fig. 2.34. These results show that for large enough maps, the speedup increases by the number of robots in the cluster.

Algorithm 7: Merge(M_1,M_2,MAP), with clustering solution

```
inputs:
        $M_1 = [V, E, k_1]$
        $M_2 = [W, F, k_2]$
Output: A map that is the result of merging $M_1$ and $M_2$
begin
    if ($k_1 < M$) and ($k_2 < M$) then
        Merge_seq($M_1$,$M_2$,MAP)
        return(MAP)
    else
        Merge_parallel($M_1$,$M_2$,MAP)
        return(MAP)
end
```

Another parameter to evaluate the performance of parallel algorithms is *efficiency* which measures the utilization rate of the processors in the execution of a parallel program. It is equal to the ratio of speedup and the number of processors used. We have measured this parameter for the proposed map merging parallel algorithm. Fig. 2.35 (left) demonstrates the efficiency of the proposed parallel algorithm on the real world and simulation topological maps, while, Fig. 2.35 (right) presents the efficiency of the algorithm on the artificially generated topological maps. Both graphs show similar results, i.e, the efficiency is mostly between 0.6 and 0.8 specially on large graphs it converges to 7.2. This is a significant result that proves the efficiency of the method in reality, since it was obtained by implementation of the algorithm on a real robotic cluster.

The presented results showed that the proposed method can speed up the execution time of a map merging algorithm. We believe that this approach can be used in many other multi-robot applications that demand for high processing resources.

2.5 Conclusions and discussion

An algorithm for multi-robot odor source search, exploration and topological mapping inside large buildings composed by parallel and perpendicular corridors (warehouse-like

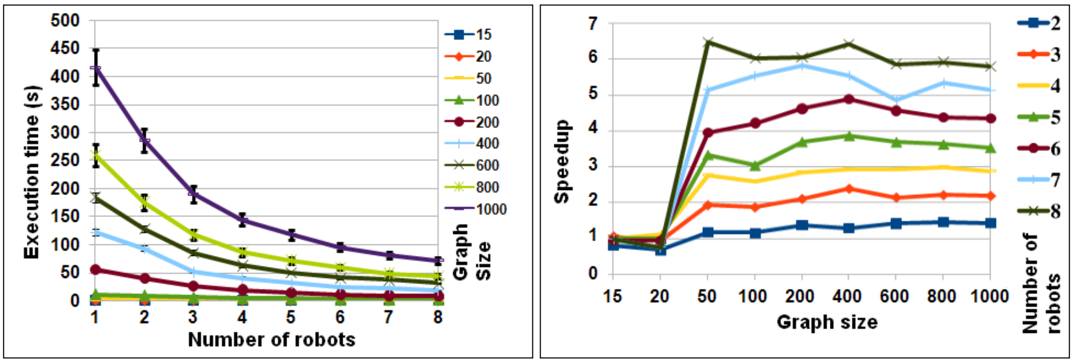


Figure 2.34: Left: Execution time results for different size of artificial maps based on number of processors. Right: Speedup for different sizes of artificial topological maps

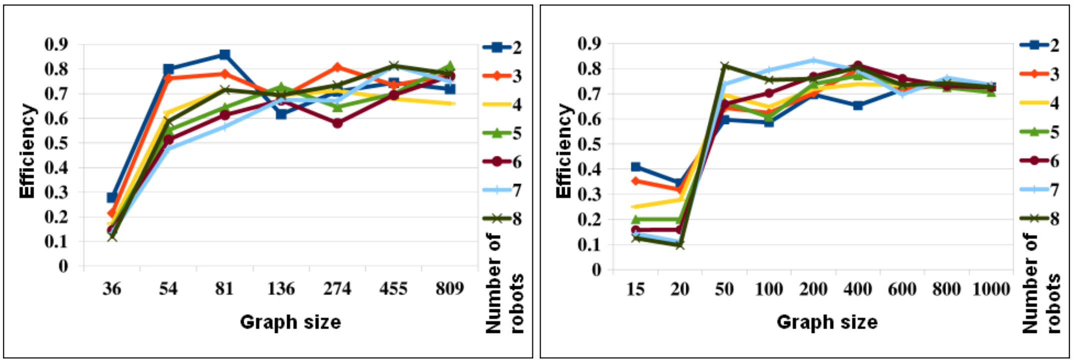


Figure 2.35: Left: The efficiency of parallel algorithm in different size of real and simulated topological maps. Right: The efficiency of parallel algorithm in different size of artificially generated topological maps.

environments) was proposed. The algorithm was experimentally validated in a small, realistic testing environment with up to three Roomba robots. Details about all the complementary aspects required to such implementation are provided. These include motion control, feature extraction and classification, localization and map merging. The exploration efficiency is improved by integrating odor sensing cues in the frontiers selection so they navigate towards the odor sources and localize them while cooperating with each other by sharing information in their local maps. If the robots do not sense any olfactory clue, they try to efficiently explore the environment until they get into active odor plume area. When a robot is inside an odor plume (sensing high odor concentration) it intends to track the plume by traveling in the upwind direction. In the case that a robot is in a situation that wants to make decision to select a frontier, it picks up the frontier with highest odor concentration. In terms of mapping, the robots generate the topological map of environment during exploration. Map sharing is the main tool for automatic distributed task sharing and cooperation in this method. The robots merge their topological maps based on common subgraph isomorphism techniques even if they do not have common coordinate systems. Based on the stored information inside the maps, each robot knows the location of unexplored frontiers and assigns one of them to itself as a target to explore, so the other robots will not select the same frontier. Two solutions were presented for the problem of topological map merging; sequential solution and parallel solution. The parallel solution is run on a robotic cluster platform. In the real world experiments presented in this chapter we implemented the sequential approach since the size of maps were relatively small. However, the parallel approach was also implemented on a real robotic cluster and its performance was measured using various types of topological maps. The results show a significant speed-up in merging large maps using the proposed method.

In addition to real world experiments, the algorithm has been simulated against a large variety of configurations in the Player/Stage framework. Odor plumes have been simulated using PlumeSim in Player/Stage. The effect of the number of robots on exploration in different type of environments has been analyzed and discussed. The results of real world experiments show the effect of gas cues on the behavior of robots and it shows that using the proposed algorithm, robots firstly explore the areas with higher probability of existence of odor sources.

Simulation results show that having more robots is more advantageous in a complex maze than in a simple maze.

Finally, there are some issues that should be discussed here. Although, most of the real world human-made structures can be modeled to the kind of environments that are tested in this chapter but the question is if this approach can be easily adapted to a situation in which the robot cannot be assumed to be able to reliably perceive the feature it is traveling through. In more complex environments, as it was previously stated, the feature extraction method that is used in this study can be replaced by another methodology providing better results in the target environment. Even if the range of the sensors is limited or there is noise in sensing, the robot can perform some movements in the environment to correctly identify the environmental features. The number of defined features can also be increased in such complex environments. However, if the robots are not able to correctly identify the environmental features, the topological maps will have some degree of unreliability and map merging cannot be done based on similarity checks of this chapter. In most of the human constructed facilities, the robots are able to classify the general features of the environment (maybe by using more complex sensors including 3D Laser range finders and cameras) and so this method can be applied.

Regarding possible conflicts between the robots in their task-allocations, three points should be mentioned. First, the topological maps are very small data files that are transferred in a few milliseconds and there is not a big delay in the communications between the robots. Second, it must be very rare that two robots want to choose frontiers exactly at the same time because most of the times the robots are traveling without changing their frontiers. Third, even if two robots are going to choose frontiers at the same time, they might automatically choose different frontiers based on their positions and the utility/cost function. Considering these three points, the probability of two robots to choose the same frontier is very low. Even if this happens (that we've never seen in the experiments and simulations) two robots will temporarily travel to explore one frontier, however, as soon as one of them gets to a new detected feature, it will process the map to pick up a frontier and for sure this time it will not choose the same frontier since it is already taken by the other robot. The other point is that in the algorithm 1 the robots make their decisions independently and there is no condition that may lead to a deadlock between the robots. In algorithm 2 that is for improving the path planning of the robots, there are priority checkings between the robots. The robot that has lower priority should go back or wait for some seconds until the other robot passes.

Chapter 3

Swarm olfactory search

"If you justify a mistake with one thousand reasons, you will end up with one thousand and one mistakes."
Avicenna (Abu Ali Sina), 980-1037

The second objective of this thesis is to address the problem of odor plume finding by robotic swarms in an unknown unstructured environment. As mentioned earlier, odor plume finding is the first step in localizing odor sources. The goal is to propose an efficient methodology to use a large number of simple sensing nodes, increasing the probability of finding an odor plume. This chapter considers unstructured environments under turbulent airflow in designing swarm search methodologies.

3.1 Problem statement

The problem of "odor plume finding using a swarm of robots" is stated as following:

Consider a swarm of N robots that are able to communicate with each other over a distance Δ_d and are equipped with olfactory sensors for sensing the odor concentration $\bar{C}$ and airflow speed $\vec{U}$. There is no central controller for the system, so the robots act independently. The problem is: "How to maximize the probability of finding odor plumes in an unknown environment with a set of N searching swarm robots without a central coordinator having minimal communications?"

To maximize the probability of finding odor plumes in an unknown environment, one should answer the following questions:

1. What is the best spatial formation for the swarm robots in searching for an odor plume?
2. What is the best movement strategy for the swarm in odor plume searching?

These questions seem dependent upon each other. Probably the best spatial formation for a particular movement strategy is not the best for another movement strategy. To find the best formation of swarm robots in search for odor plumes, this thesis considers two strategies regarding the swarm movements.

- Initially we assume no global movements for the swarm and analytically find the best formation of robots (section 3.3).
- Afterwards assuming cross-wind movements for the robots we find the best formation of swarm robots in search for odor plumes (section 3.6).

It is analytically proved and generally accepted that cross-wind movement is the best strategy to acquire the maximum information in searching for odor plumes in an unknown environments under stable airflow [SS84]. None of the works either in the olfactory search area or in the swarm robotics field has ever found the best spatial formation in the mentioned problem.

3.2 Related works

Global swarm behaviors emerge consequently from local individual robots' interactions, without needing any global engineering principles. In design of swarm robotics systems, complex techniques are generally avoided, while instead, principles such as locality of sensing and communication, simplicity of distributed behavioral algorithms and homogeneity are followed. The main benefits of pursuing a swarm robotics approach are scalability in the number of robots, robustness in noisy conditions, and fault tolerance against individual failure [NCD08].

Swarming has been rarely exploited in olfactory searching, but there have been many groups working on other aspects of swarm engineering. Swarm robotics pattern formation [NCD08, GP02], aggregation [SŞ07], navigation [CBCR99], self-assembly [GBMD06], and foraging [TDK02] are the main challenges in this field. Beside these challenges, there are extensive works on modeling [MEA04], learning [PM07], communications [CRPM06] and analytics [GP02] of swarm robotics. Moreover, there are several applications that have been addressed by swarm robotics approaches, namely, items clustering [HM99], area coverage [SKS06], stick pulling [IMBG01], collective transportation [KB00b], self-organized task allocation for foraging tasks [KB00a], and landmine detection [CBCR99]. However, this thesis is focused towards tackling the olfactory search problem for large groups of robots following swarm robotics principles.

Although research in this fields started with centralized and leader based approaches, most of the recent research is concentrated on decentralized approaches, which offer reduced computational complexity and robustness to failures. Such approaches include behavioral [BS05], artificial potential functions [GP04b], virtual agents [BL02], and probabilistic [SS05].

For olfactory search problem, a mobile sensor network is more advantageous, in comparison to a single robot that can measure only the odor concentration on its own place. The airflow that carries the odor patches can be very irregular and chaotic, thus the resulting distribution of odor concentration may be also very irregular with large intermittency in the region downwind an odor source. Additionally, the search space may be much larger than the active area of an odor source. In these conditions, using multiple sensing nodes spread throughout the environment improves the detection process, increasing the probability of finding an odor plume in a given time. A swarm of robots can establish a dynamic mobile sensor network and move in the area of interest to find the olfactory targets.

Most of the works concerning olfactory search have focused on odor plume tracking [MRH06, LFC01, LM09, LKV06] whereas plume finding has received less attention. Most of these studies (e.g., [GDC^{+}97, KRRJ06, VVS07]) assume that robots start their search within or very near the plume. Plume finding problem is usually addressed through general exploration methods [MNMdA09a], mapping [LCLG08, LD04], or coverage techniques namely zig-zag sweeping, casting [PBiBB+06], random wandering [ITTM06], biased random walks [MNdA02b], lévy-taxis [PBG09], and spiral movements [FCM+09], which are also used for other spatial search tasks and are not efficiently designed for odor plume finding.

Recently, a few studies were reported that tackle the problem of optimal gas sensor deployment mainly for safety systems in process facilities. Legg et al. [LBSS+12] presented a method that utilizes computational fluid dynamics (CFD) simulations to optimize gas sensor locations in order to maximize the likelihood of early detection of gas clouds in specific facilities. Miyata and Mori [MM11b] introduced another procedure for optimization of gas detector locations by using gas dispersion simulation tools in specific chemical plants. However in these studies, the gas source location (leak point) and the map of environment are both a-priory known, and best sensors' positions among a list of candidate locations is found by simulations. A-priory knowing the source location and the map of the environment and having a list of candidate positions for the sensors are three assumptions that we do not make in this thesis. Moreover, instead of CFD simulations, this chapter provides analytical results using gas dispersion models.

Despite most of the previous studies, this thesis presents an analytical method to find the optimal spatial formation of swarm robots in plume finding strategies (described in sections 3.3.1 and 3.6). Among of the main novelties of this thesis is definition of single and multiple (stationary and mobile) gas sensors coverage and finding the optimal configuration of N mobile sensors in different environmental conditions. Moreover, based on the results of optimizations, we present and design a set of wind-biased virtual attractive/repulsive control forces for the swarm robots such that their emergent behavior converges to the optimal formations (explained in sections 3.4 and 3.7.1). None of the previously designed control systems in olfactory robotics community has biased the virtual forces by the wind effect. The optimization results were validated and evaluated by experiments in small scale realistic environments (in sections 3.5 and 3.8).

3.3 Optimal coverage of stationary gas sensors

Finding the best spatial formation of swarm robots for detection of odor clues is the first step of searching for olfactory targets in a given space using a swarm of robots. This section formulates the problem of odor plume detection by a network of stationary gas sensors and analytically proposes an optimal configuration for plume detection, under a set of assumptions for the problem. This solution was analyzed and verified by simulations and finally experimentally validated in a reduced scale realistic environment using a set of Roomba-based mobile robots. The simulations and the experimental results verified the analytically found optimal formations.

This section finds the best configuration of robots to maximize their sensing coverage area and section 3.4 designs swarming behaviors of individual robots to reach to the found configurations.

3.3.1 Odor dispersion model

The effort to design and develop robotic olfactory search strategies faces the problem of understanding how the odor molecules disperse in the environments under naturally turbulent flow. As already mentioned in Chapter 2, odor patches released by an odor source are mainly transported by the airflow, forming an odor plume. As the plume travels away from the source, it becomes more diluted due to molecular diffusion and turbulence that mixes the odor molecules with the clean air [RW02]. Molecular diffusion is a slow process whose effect on the plume shape can be neglected. The dispersion of odor molecules is dominated by flow turbulences in ventilated indoor or in outdoor environments. The odor molecules move downwind due to mean flow velocity $\vec{U}$ while their net motion is a random walk because of the fluctuations. In large scale environments, fluctuations happen also in the initial direction of the plume that create undulating and meandering patterns. As the flow carries patches of odor, the amplitude of the concentration within a patch decreases away from the source, and the average time between two successive patches increases. The instantaneous odor concentration strongly fluctuates intermittently with peaks above three orders of magnitude around the average concentration value [CWK02]. Under these circumstances, a chemical sensor located far enough downwind of the odor source measures no odor concentration most of the time. The probability of encountering an odor patch at any given point is determined by the relative location of the sensor to the odor source, the statistics of the flow and the shape of the environment and the obstacles [Sut47, Gif60]. The velocity of the airflow is set by the environmental conditions and hence stays unchanged for long periods of time compared with the time scale of odor fluctuations. Fig. 3.1 presents the nature of an odor plume from different scales.

Probability density function of odor dispersion in a turbulent medium is represented by the pseudo-Gaussian model for odor distribution in average-term exposure [Sut47, Gif60, RW02]. The Gaussian plume models that are solutions for the differential equation (2.1), yield results that match experimental results reasonably well [Jon83]. If an odor source is located at position $(0,0,0)$, its release rate is Q and the average wind speed is $\bar{U}$ toward x-axis (Fig. 3.2), then the mean concentration of odor in position (x,y,z) is given by the following probability density function:

$$\bar{C}(x,y,z) = \frac{Q}{2\pi\bar{U}\sigma_y(x)\sigma_z(x)} \exp\left\{\frac{-y^2}{2\sigma_y^2(x)} + \frac{-z^2}{2\sigma_z^2(x)}\right\} \tag{3.1}$$

where x, y, and z (here and throughout this article) denote the downwind, crosswind, and vertical position coordinates relative to the odor source with x positive along the mean wind direction $\overrightarrow{U}$.

The standard deviations $\sigma_y(x)$ and $\sigma_z(x)$ model the horizontal and vertical dispersion of the plume. These standard deviations are not constant. It was experimentally found by Briggs [Bri73] that both parameters are functions of the downwind distance from the source (x) according to the environmental conditions, as expressed in Table 3.1. In this table the following environmental conditions are considered; A: neutral, B: slightly stable, C: stable, D: isothermal, E: moderate inversion, F: strong inversion.

Fig. 3.2 shows the mean odor concentration in a 2-D plane of $z = 0.1m$, generated from an odor source at (0,0,0) when release rate is 0.01 g/s and wind speed is $1m/s$ toward the x-axis direction, in (left) neutral/slightly stable (A-B) and (right) moderate/strong inversion (E-F) environmental conditions, based on equation (3.1) and Table 3.1.

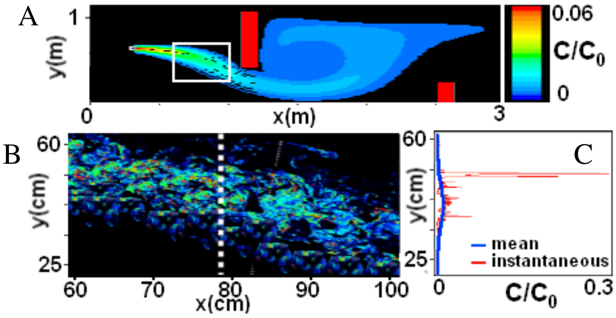


Figure 3.1: Odor plume structure in an environment with obstacles. A. mean structure, B. instantaneous structure, and C. instantaneous slice.

Table 3.1: Standard deviations for an urban environment in various environmental conditions [Bri73].

Env.	$\sigma_y(x)$	$\sigma_z(x)$
A-B	$0.32x(1+0.0004x)^{-0.5}$	$0.24x(1+0.001x)^{0.5}$
C	$0.22x(1+0.0004x)^{-0.5}$	0.20x
D	$0.16x(1+0.0004x)^{-0.5}$	$0.14x(1+0.0003x)^{-0.5}$
E-F	$0.11x(1+0.0004x)^{-0.5}$	$0.08x(1+0.0015x)^{-0.5}$

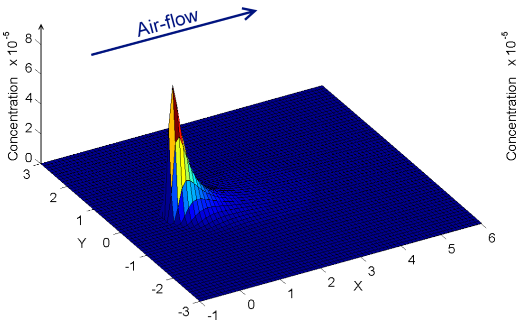


Figure 3.2: The mean concentration in the 2D plane of $z = 0.1m$. source location:(0,0,0), release rate = $0.01\ g/s$ and $\bar{U} = 1m/s$. Left: A-B conditions, right: E-F conditions

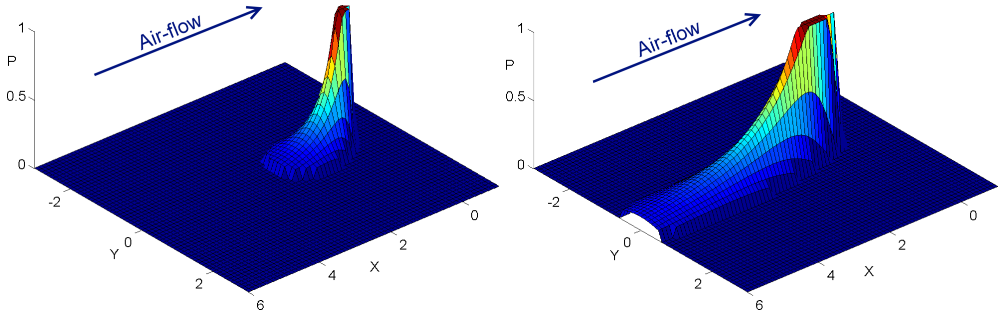


Figure 3.3: The probability of detecting odor patches by a sensor at (0,0) if the odor source is located in various points in the plane, when source release rate = $0.01\ g/s$, k = 10^5 and $\bar{U} = 0.5m/s$ in A-B environmental conditions (left), and E-F environmental conditions (right).

3.3.2 Gas sensor area coverage

Most gas sensors show pseudo-linear responses to gas concentrations [AML+04]. Considering the odor dispersion model in (3.1) at a fixed height (z = constant = source height) this 3D phenomena can be treated as a 2D problem. Similar to [BS02] and [MYWZ11], we conclude that if an odor source is at $O(x_0,y_0)$, the conditional probability of detecting odor patches by a stationary gas sensor located at position (x,y) is given by:

$$P(D_{xy}|O_{x_0y_0}) = \frac{kQ}{2\pi\bar{U}\sigma_y(x)\sigma_z(x)}\exp\{-\frac{(y-y_0)^2}{2\sigma_y^2(x)}\} \tag{3.2}$$

where k is the sensitivity parameter of a gas sensor to the odor concentration. In other words, if a sensor is located at position (x,y), its probability of detecting an odor patch released from a source located at (x_0,y_0) is given by $P(D_{xy}|O_{x_0y_0})$ in (3.2). Equation (3.2) defines that the higher the concentration of the odor is, the higher the probability of detecting by a sensor becomes. It should be mentioned that the environment in this model (and throughout this chapter) is presented by uniform grid maps, so any Cartesian (x,y) denotes a grid cell with center at (x,y). Since this equation presents a probability function, its result is truncated to $[0,1]$. Although this equation has been simplified by considering $z = constant = z_0$, standard deviations of vertical direction ($\sigma_z(x)$) exists and plays a significant role in this probability function. From equation (3.2), if there is a gas sensor at a given position (x,y), where $(y-y_0) >> \sigma_y(x)$, its probability of finding an odor patch is very small. Fig. 3.3 is an example that presents the distribution of this probability when a sensor is located at (0,0), $\bar{U}$ is 1 m/s, and the environment is in moderate/strong inversion conditions (E-F type). As it is shown in Fig. 3.2 and Fig. 3.3, the odor plume emitted from a source, shapes towards the airflow direction, whereas, the probability of detection of a gas sensor, shapes towards the opposite direction of the airflow.

Given N independent sensors s_i located at (x_i,y_i), $i = 1...N$, we compute the total probability $P(D_{(x_i,y_i)_{i=1}^N}|O_{x_jy_j})$ of detecting odor patches released from a source in $O(x_j,y_j)$, due to the combined efforts of all sensors. This probability is one minus the probability that all sensors fail to detect:

$$P(D_{(x_i,y_i)_{i=1}^N}|O_{x_jy_j}) = 1 - \prod_{i=1}^{N}\{1 - P(D_{x_iy_i}|O_{x_jy_j})\} \tag{3.3}$$

In other words, if an odor source is located at $O(x_j,y_j)$, the probability that at least one sensor (out of N applied sensors) detects odor patches is given by $P(D_{(x_i,y_i)_{i=1}^N}|O_{x_jy_j})$ in equation (3.3).

The probability functions (3.2) and (3.3) inherently define probabilistic coverage area for gas sensors. To obtain the area covered by the gas sensors, a sensitivity threshold S_{th} for the probability of odor patch detection should be considered. This is based on the fact that most gas sensors show a sensitivity threshold i.e. bellow a certain odor concentration, the sensors do not detect any odor patch. Thus, this thesis defines the binary coverage of a gas sensor to a point of interest as following:

Definition 1 (Single Gas Sensor Binary Coverage) *Given a standstill sensor s_i at position (x_i,y_i) and a point of interest $p_j = (x_j,y_j)$ the coverage of the sensor s_i to the point p_j is defined as:*

$$C_s[s_i(x_i,y_i),p_j] = \begin{cases} 1, & P(D_{x_iy_i}|O_{x_jy_j}) > S_{th} \\ 0, & Otherwise \end{cases} \tag{3.4}$$

where $P(D_{x_iy_i}|O_{x_jy_j})$ is given by (3.2). Fig. 3.4.A presents the coverage area of one standstill gas sensor when the wind is towards right direction. Despite most of the coverage areas of other types of sensors (e.g., acoustic, thermal, vision) which are either circular or directional sectors towards sensor's heading, the coverage area of gas sensors is ellipsoid shape biased towards the upwind direction.

With the knowledge of the coverage between sensor s_i and all points of interest, the overall coverage by sensor s_i can be defined by aggregation. If there are m points of interest, then the total coverage by a sensor s_i is defined as:

Definition 2 (Overall Coverage by a Sensor) *The overall coverage "$cover_s[s_i]$" by a sensor s_i over a region with m points of interest in R^2 is given by:*

$$cover_s[s_i] = \sum_{j=1}^{m} C_s[s_i(x_i,y_i),p_j] \tag{3.5}$$

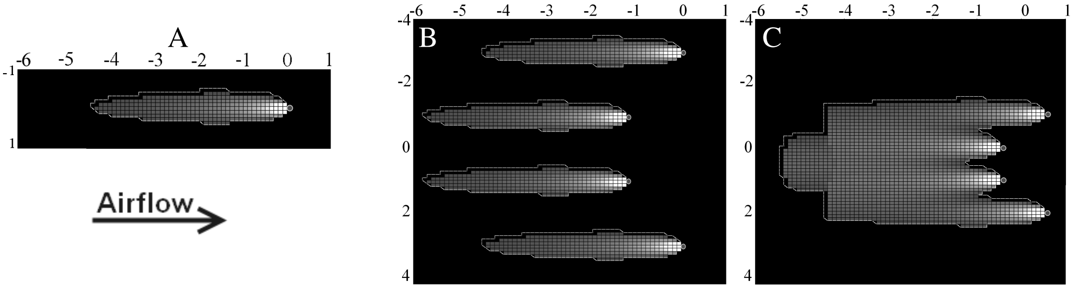


Figure 3.4: Arbitrary placement of sensors and their coverage area when S_{th} = 0.3, $k = 10^5$, Q=0.01g/s, and U=0.2m/s in strong inversion conditions. The red circles show the gas sensors and the white regions represent the probabilistic coverage area. Note: the coverage area in C is larger than in B.

In this chapter, m is the total number of grid cells of a region.

Given N sensors s_i located at (x_i,y_i), $i = 1...N$, we define their combined coverage to the point p_j as:

Definition 3 (N Gas Sensors Binary Coverage) *The combined binary coverage of N standstill sensors s_i, $i = 1...N$ on a point p_j is defined as:*

$$C_s[S_{(x_i,y_i)_{i=1}^N}, p_j] = \begin{cases} 1, & P(D_{(x_i,y_i)_{i=1}^N} | O_{x_jy_j}) > S_{th} \\ 0, & Otherwise \end{cases} \tag{3.6}$$

$P(D_{(x_i,y_i)_{i=1}^N} | O_{x_jy_j})$ is given by equation (3.3) and is the combined probability of detection of odor patches if the odor source is located at $O(x_j,y_j)$ by N sensors and S denotes the set of (x_i,y_i) positions of the sensors.

Finally, the overall coverage of N sensors over a region is defined by:

Definition 4 (Overall Coverage by N Sensors) *The overall coverage "$cover_s[S]$" by N sensors s_i, $i = 1...N$ over a region with m points of interest in R^2 is given by:*

$$cover_s[S] = \sum_{j=1}^{m} C_s[S_{(x_i,y_i)_{i=1}^N}, p_j] \tag{3.7}$$

This equation implies that the overall coverage is a function of:

- sensors' positions,

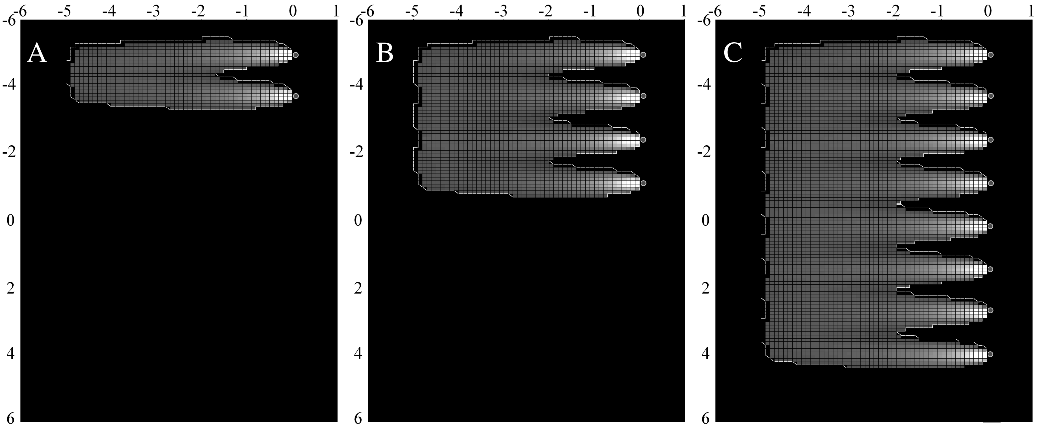


Figure 3.5: The optimized configuration of two, four and eight gas sensors in an area when the wind-speed is $0.2m/s$ (moderate/strong inversion (E-F) conditions). The coverage area in B is larger than in Fig. 3.4.B and also Fig. 3.4.C and any other configuration of 4 sensors.

- source release rate,
- average wind speed,
- sensors sensitivity,
- and distribution standard deviations related to environmental conditions.

Using these equations, Fig. 3.4.B and Fig. 3.4.C present the coverage area of four sensors in two different configurations. It should be pointed out that, the coverage area in Fig. 3.4.C is larger than the coverage area in Fig. 3.4.B, showing that the coverage area of N sensors depends on their spatial topology in fixed conditions.

3.3.3 Optimal sensor deployment

Optimal sensor deployment aims to position the sensors in a way that overall coverage is maximized. Thus, we are looking for a series of sensors positions $s_i = (x_i, y_i)$ such that:

$$\{s_1, s_2, ...s_N\} = \text{arg max } cover_s[S]$$

The optimal sensor positions are where the coverage area of the sensors is maximized. Therefore maximizing the area of sensor coverage, defined in (3.7), is used as the criterion of

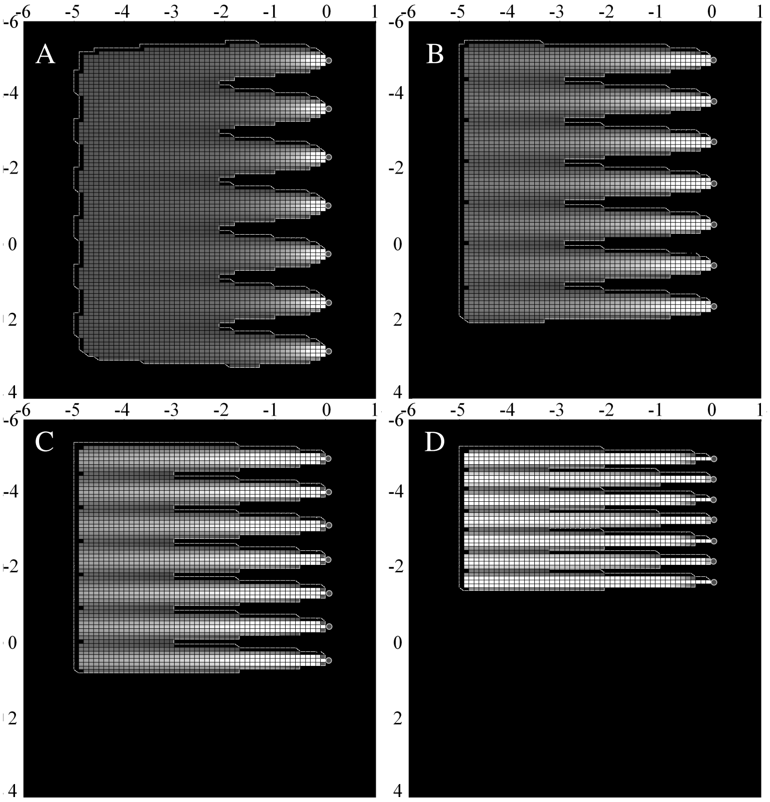


Figure 3.6: The optimized configuration of seven gas sensors in an area when the wind speed is A: 0.2, B: 0.5, C: 2, and D: 4 m/s in moderate/strong inversion (E-F) environmental conditions.

our optimization. We optimize this criterion with various number of sensors and different average wind speeds in four environmental conditions. Without loss of generality, we assume constant values for the following parameters during the optimizations: $S_{th} = 0.35$, $Q = 0.01g/s$, $k = 10^5$ and the environment size = $50 \times 50m^2$. These values are close to real world experimental measurements [CABJ+11].

The Powell's conjugate gradient descent method [PFTV02] was used (in Matlab) to optimize this problem, since it does not need the derivative of the function and its convergence is fast even in high dimensional spaces. N sensors on a 2-D plane require 2N dimensional search space. For each combination of sensors' position, the coverage area is computed. The solution is a set of positions for sensors that their coverage area is the largest.

3.3.4 Optimization results of stationary gas sensor deployment

The optimal coverage area was measured for different number of sensors from 3 to 16, and different wind speeds from 0.1 to 15 m/s in the four environmental conditions listed in Table 3.1. The topological shape of the sensors in the optimal solutions was analyzed in each case.

Fig. 3.5 shows examples of optimized positions of two, four and eight sensors and their maximum coverage area in an environment under moderate/strong inversion (E-F) conditions when the wind speed is equal to 0.2 m/s. Different values of U, and N in different environmental conditions result in similar (but not equal) solutions. Fig. 3.6 shows another example of optimized positions of seven sensors with different values for the wind speed. A remarkable point from all of the optimized solutions is that:

Conclusion 1 (Cross-wind Line Topology) *The topology of all of the optimal solutions is **line** configuration towards cross wind direction, with equal distance between each pair of neighboring sensors.*

Fig. 3.7 is an example that shows the optimal distance between the adjacent sensors in the optimized configuration in neutral/slightly stable (A-B) and moderate/strong inversion (E-F) environmental conditions when the number of sensors is 3 to 16. This chart only shows the optimal results when the environmental conditions is A-B and F-E types (see table 3.1), however in the other environmental conditions, the obtained results for the same number of sensors are similar to this figure. By analyzing these results of the optimizations, it can be seen than, in constant wind speed, when the number of sensors changes (from 3 to 16), the optimal distance between the sensors changes only for a few centimeters and is almost constant even in various environmental conditions (see the examples in Fig. 3.5 and Fig. 3.7). The maximum mean square distance of each chart to their average (for any value for the airflow) in Fig. 3.7 is about 10.5cm. Therefore, we can draw the second conclusion as follows:

Conclusion 2 (Wind Dependent Distance) *The distance between neighboring pairs in optimal configurations depends mainly on the wind speed, whereas, the number of sensors and the environmental conditions do not show significant impact on optimal configurations.*

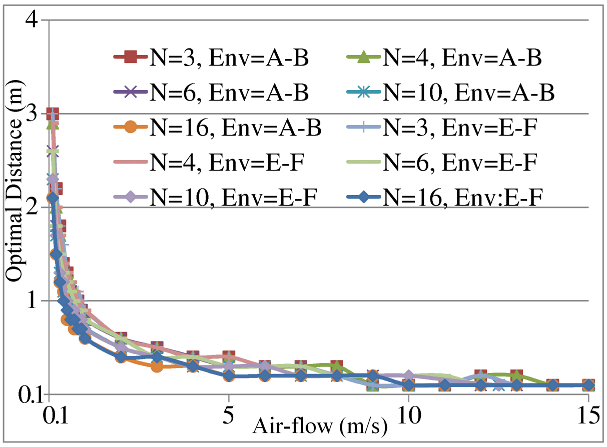


Figure 3.7: The optimal distance between the neighboring sensors in the optimized configurations. Number of sensors (N) varies from 3 to 16, environmental conditions (Env) is listed in Table 3.1, and the airflow is between 0.1 to 15 m/s.

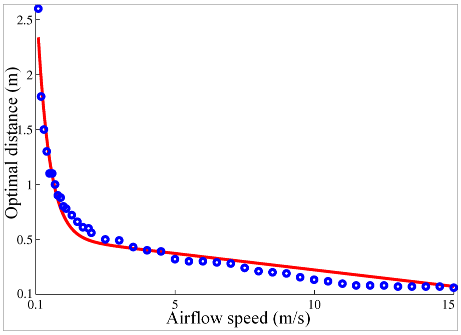


Figure 3.8: The average optimal distance between neighboring sensors in different airflow speeds in different environmental conditions.

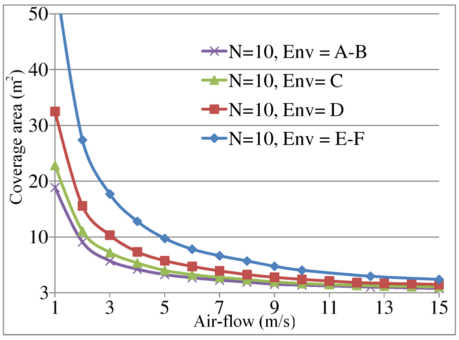


Figure 3.9: The maximum coverage area of 10 gas sensors in 4 different environmental conditions while the airflow varies from 0.1 to 15 m/s.

These conclusions are drawn after the results obtained from numerical optimizations and are among the most significant contributions of this thesis. The results in Fig. 3.7 show that **the higher wind speed, the smaller the optimal distance**. Therefore, as the wind speed increases, the distance in-between the sensing nodes should decrease in order to maintain optimal coverage, however as the wind speed decreases sensors should get apart and keep a larger distance in order to maximize their coverage area. Fig. 3.8 shows the average optimal distance between adjacent sensors in different airflow speeds in different environmental conditions.

Taking the results shown in Fig. 3.8 and using a non-linear regression analysis, an analytical equation was obtained that describes the optimal distance between the sensing nodes as a function of the wind speed:

$$f(U) = 2.28e^{-2.3U} - 0.03U + 0.52 \tag{3.8}$$

The red line in Fig. 3.8 is the fitted function (3.8) and the scattered blue circles are the results of the optimizations. The mean square error of this regression line is 7.3×10^{-3}. This function is later used by moving senor robot to estimate the optimal distance based on the wind speed.

Fig. 3.9 presents the overall coverage area achieved by 10 sensors in the optimal configurations for four different environmental conditions. Although the optimal distance for the neighboring sensors is not dependent to the environmental conditions or to the number of sensors, Fig. 3.9 shows that when the environment is under moderate/strong inversion conditions, the coverage area of a group of gas sensors is larger than when the environmental conditions is neutral/slightly stable. On the other hand, in constant environmental conditions, **when the wind speed is lower the coverage area increases**. Therefore, in a windy environment more sensors are required to cover a given area. Similar results were obtained considering other numbers of gas sensors.

It should be mentioned that the obtained results are valid for the values of S_{th}, k and Q defined in section 3.3.3, however for other values, the optimal configuration for sensors is the same (i.e. a cross-wind line) and only the values of the optimal distance between the sensors and regression function (3.8) change. This process of optimization can be repeated and optimal results can be achieved for other conditions.

3.4 The proposed method, wind-biased potential fields, “cross-wind line-up” behavior

The optimization results showed that, to maximize the probability of detecting odor plumes by a swarm, the robots should line-up cross-wind with equal distances from each other. There is no central node for swarm robots and the formation topology of the swarm is the emergent result of individual robots movements. Therefore, for the swarm to have a desired formation topology, each robot should move in the space with a correct and well-defined control manner. To control the motion of the robots to reach the optimal formations, this section presents a novel method based on the virtual attraction/repulsion forces [GP04a, GP04b]. Despite previous works on swarm formations, we take the wind direction and the wind speed into account and bias the attraction/repulsion forces by the wind to implement the desired cross-wind line-up formation. This method is a suitable control strategy for the swarming robots since it does not need a central control node and it is flexible to impediment other robotic behaviors (e.g obstacle avoidance). We define a behavior named “cross-wind line-up” for the individual robots in order to implement line formation for the swarm. This behavior defines two types of virtual forces that are applied to the robots: robot-to-robot and robot-to-environment forces.

3.4.1 Robot-to-robot forces

To line-up the robots toward the cross-wind direction, each robot measures the air-flow direction $\overrightarrow{U}$ and assumes this direction as its internal X-axis coordinate system and then it measures the relative distance to its neighboring robots. Then, the robots try to minimize their X-axis distances from their neighbors and maintain a constant distance with them in their Y-axis. Hence, we define a nonlinear bounded potential between each pair of neighboring robots i and j at time t:

$$< \overrightarrow{X}\ axis > \quad \equiv \quad < \overrightarrow{U} > \tag{3.9}$$

$$\overrightarrow{F}^{ij}_{cr}(t) = \overrightarrow{Fx}^{ij}_{cr}(t) + \overrightarrow{Fy}^{ij}_{cr}(t) \tag{3.10}$$

$$\overrightarrow{Fy}^{ij}_{cr}(t) = \begin{cases} -\mu_1\Big(\|\overrightarrow{Y}_{ij}\| - D_1\Big)\Big[\frac{1}{\|p_{ij}(t)\|^2}\Big]\Big[\frac{\overrightarrow{Y}_{ij}}{\|\overrightarrow{Y}_{ij}\|}\Big], & 0 < \|\overrightarrow{Y}_{ij}(t)\| < D_1 \\ -\mu_2\Big(\|\overrightarrow{Y}_{ij}\| - D_1\Big)\Big[\frac{1}{\|p_{ij}(t)\|^2}\Big]\Big[\frac{\overrightarrow{Y}_{ij}}{\|\overrightarrow{Y}_{ij}\|}\Big], & D_1 < \|\overrightarrow{Y}_{ij}(t)\| < D_2 \\ 0\,, & \|\overrightarrow{Y}_{ij}(t)\| > D_2 \end{cases} \tag{3.11}$$

$$\overrightarrow{Fx}^{ij}_{cr}(t) = \begin{cases} -\mu_3\overrightarrow{X}_{ij}\Big[\frac{1}{\|p_{ij}(t)\|^2}\Big], & 0 < \|\overrightarrow{X}_{ij}(t)\| < D_2 \\ 0\,, & \|\overrightarrow{X}_{ij}(t)\| > D_2 \end{cases} \tag{3.12}$$

$$D_1 = 2.28e^{-2.3U} - 0.03U + 0.52 \tag{3.13}$$

where

- $\overrightarrow{F}^{ij}_{cr}(t)$ is the force applied to robot i by robot j at time t. $\overrightarrow{Fx}^{ij}_{cr}(t)$ and $\overrightarrow{Fy}^{ij}_{cr}(t)$ are respectively the x and y components of $\overrightarrow{F}^{ij}_{cr}(t)$.

- $\|p_{ij}(t)\|$ is the distance between robots i and j. The term $[\frac{1}{\|p_{ij}(t)\|^2}]$ correlates the force between each pair of robots to their inverse square distance. Therefore, the robots in close vicinity apply large magnitude forces to each other while they do not apply significant forces to the robots which locate very far.

- $X_{ij} = x_i - x_j$ and $Y_{ij} = y_i - y_j$ where (x_i, y_i) is the relative position of robot i and (x_j, y_j) denotes the relative position of robot j. It is obvious that $\|\overrightarrow{Y}_{ij}\|$ denotes the magnitude and $\Big[\frac{\overrightarrow{Y}_{ij}}{\|\overrightarrow{Y}_{ij}\|}\Big]$ is the direction of the vector $\overrightarrow{Y}_{ij}$ (either +1 or -1).

- μ_1, μ_2 and μ_3 are constant coefficients for tuning acceleration of the robots. μ_1 is the Y-component repulsing coefficient and μ_2 is the Y-component attracting coefficient while μ_3 is the X-component attracting coefficient.

- D_1 is a design parameter that specifies the desired distance interval between the neighboring robots. We defined D_1 using the equation (3.8) to be equal to the optimization results.

- D_2 defines the margin of the area that a robot applies forces to the other robots. Logically for line formation D_2 should be bigger than D_1 and smaller than $2D_1$. Moreover, $0 < D_1 < D_2 < \Delta_d$.

The design of the above equations is inspired by the Hooke’s law, thus the forces are similar to the forces in the physical springs. Hence, the robots try to minimize their X-component distance to zero and to maintain a distance of D_1 (that is the optimized distance) in their Y-component distance (see Fig. 3.10). Since the X-axis in the robots is selected to be toward the air-flow direction $\overrightarrow{U}$, the robots will line up cross-wind with constant distance of D_1 towards the Y-Axis.

Using the above equation, the total “cross-wind line-up” force $\overrightarrow{F}^i_{cr}(t)$ for robot i is determined as:

$$\overrightarrow{F}^i_{cr}(t) = \sum_{j=1;j\neq i}^{N} \overrightarrow{F}^{ij}_{cr}(t) \tag{3.14}$$

It is worth to mention that, although the summation of the force is over all the other robots (N), only those within the detection range (Δ_d) of robot i which are closer than D_2 actually effect the value of $\overrightarrow{F}^i_{cr}(t)$.

3.4.2 Robot-to-environment forces

The low level of autonomous navigation of a robot relies on the ability of the robot to simultaneously achieve its target goal and avoid the obstacles in the environment. To avoid the obstacles, a reactive potential field control method [Kha86] is used. Fig. 3.11 is an example that shows the virtual potential forces applied to a robot in an environment. Considering M range sensors, we define the applied forces to robot i by its surrounding environment as:

$$\overrightarrow{F}^i_{obs}(t) = \sum_{j=1}^{M} \frac{c_1}{\left|d_i(j)\right|^n} \overrightarrow{(Vec_i j)} \tag{3.15}$$

Since $d_i(j)$ is the distance between robot i and an obstacle that is reported by the range sensor j, the force is an inverse function of the distance of the robot to the surrounding obstacles. $\overrightarrow{Vec_i j}$ is a predefined vector whose magnitude is set to one and its direction is from sensor j toward the center of robot i. c_1 is a positive coefficient and n is an even integer parameter.

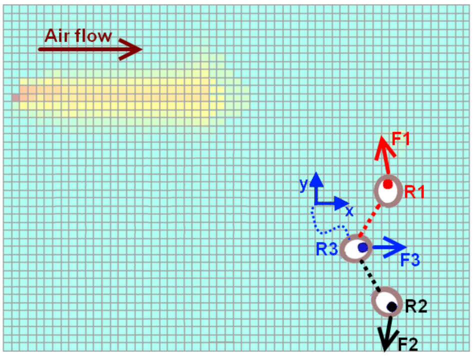


Figure 3.10: Cross-wind line-up behavior; forces applied to the robots based on equations (3.9-3.16). F1 demonstrate the total forces applied to R1 from R2 and R3. F2 and F3 present the total forces applied to R2 and R3 respectively.

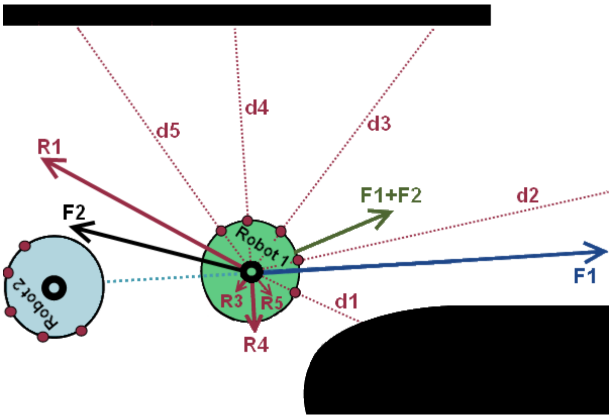


Figure 3.11: Obstacle avoidance for a robot with five range sensors. d1 to d5 correspond the distance measured by the sensors. R1 to R5 are the artificial repulsive forces. “F1” represents the artificial robot-to-robot force and “F2” illustrates the summation forces of obstacle avoidance. Vector ‘F1+F2” shows the total force applied to robot 1.

3.4.3 Swarm movements

A question that is not yet answered is that “what should the swarm do if none of the robots detect any odor patches after performing the cross-wind line-up formation?” Implementing the virtual force in equations (3.33) and (3.15) will converge to a steady-state line topology for the robots. If none of the robots detect any odor clue for a long time, it means that with high probability there is no odor source in the coverage area of the swarm. In this case the swarm robots should move spatially and explore the environment. Several different search and exploration strategies namely zig-zag casting, spiral movements, random or biased random walks, levy taxis, etc., can be taken. As an example of swarm movements, one tactic is that if the swarm robots hold the desired formation and still do not detect any odor plume, they move up-wind while keeping their line formation. Hence, they will sweep and cover the environment toward up-wind. We do not claim that this movement strategy is a perfect strategy (and it is not the goal of this section to show that), but this is only a sample strategy that we use to show how the formation configuration can be hold while the swarm moves. For the swarm to have a desired movement trajectory, each robot should move in the space in the correct direction. To implement this movement strategy, we define $\overrightarrow{F}^i_G(t)$ that is an applied virtual force to robot i at time t towards the swarm's goal. $\overrightarrow{F}^i_G(t)$, in this example, is equal to an up-wind control virtual force, $\overrightarrow{F}^i_G(t) = \overrightarrow{F}^i_{UpW}(t)$ where:

$$\overrightarrow{F}^i_G(t) = \overrightarrow{F}^i_{UpW}(t) = \begin{cases} 0\,, & \|\overrightarrow{F}^i_{cr}(t)\| > F_{th} \\ -\alpha\overrightarrow{U}_i(t)\,, & \|\overrightarrow{F}^i_{cr}(t)\| \leq F_{th} \end{cases} \tag{3.16}$$

F_{th} is a threshold value for the forces applied to a robot, α is a constant positive coefficient and $\overrightarrow{U}_i(t)$ is the airflow vector that the robot i measures at time t. The above formula checks if $\|\overrightarrow{F}^i_{cr}(t)\|$ is bigger than a defined threshold or not. If $\|\overrightarrow{F}^i_{cr}(t)\|$ is very small it means that the resultant virtual forces applied to robot i are near zero, i.e, the topology of the robot and its neighbors is in the form of a cross-wind line and it is in its steady state. In this case a force in the opposite direction of the airflow is applied to the robot ($-\alpha\overrightarrow{U}_i(t)$) and robot moves toward up-wind direction.

3.4.4 The total force

The total applied force to a robot in “cross-wind line-up” behavior is:

$$\overrightarrow{F}^i_s(t) = \overrightarrow{F}^i_{cr}(t) + \overrightarrow{F}^i_{obs}(t) + \overrightarrow{F}^i_G(t) \tag{3.17}$$

For a swarm of N individual robots in Euclidean plane, the desired direction of motion of robot i is given by:

$$\theta^i_d(t) = \arctan(\overrightarrow{F}^i_y(t), \overrightarrow{F}^i_x(t)) \tag{3.18}$$

where $\theta^i(t)$ denotes the steering angle of robot i at time t, $\overrightarrow{F}^i_y(t)$ and $\overrightarrow{F}^i_x(t)$ represent the x and y components of the force $\overrightarrow{F}^i_s$. Now, a proportional controller is used for the orientation dynamics of the robot:

$$w_i(t) = -\lambda(mod((\theta_i(t) - \theta^i_d(t)) + \pi, 2\pi) - \pi) \tag{3.19}$$

where λ is a positive proportional gain. Finally, the next velocity of the robot $\overrightarrow{v}_i(t)$ is calculated based on its last velocity $\overrightarrow{v}_i(t - \Delta t)$ and the forces applied to it $\overrightarrow{F}^i(t)$;

$$\overrightarrow{v}_i(t) = \overrightarrow{v}_i(t - \Delta t) + \eta \overrightarrow{F}^i_s(t)\Delta t \tag{3.20}$$

while η is a constant coefficient multiplied to the acceleration of the robot.

Algorithm 8: Cross-wind line up behavior

```
Behavior = Cross-wind Line-up
while Behavior == Cross-wind Line-up do
    $c_i$ = Measure odor concentration()
    $U_i$ = Measure air-flow velocity()
    $\{P_i\}$ = Measure distances to the neighbors()
    $\{d_i , Vec_i\}$ = Measure distances and direction to the obstacles()
    Navigate while holding formula(3.17)
    if ($c_i$ > odor_threshold) || (A robot indicates finding odor plume) then
        Broadcast(plume detected)
        Behavior = Plume_tracking
```

Algorithm 8 presents the pseudo-code that each robot runs to perform cross-wind line-up behavior. The robots maintain this behavior until one (or some) of them gets into an odor plume by sensing odor concentrations higher than a defined threshold. Plume tracking is not in the scope of this thesis, however a robot which gets into the odor plume can perform another behavior to inform the other robots to get into the plume and track it (discussed in chapter 4).

3.5 Validation of “cross-wind line-up” behavior

The presented optimizations and proposed wind-biased potential fields were validated in both simulations and realistic experiments.

3.5.1 Simulations

The system was tested in several different simulation environments containing obstacles with different number of robots. This section goes to the details of these simulations and presents the results.

Testing environment

Models of several testing environments were given to ANSYS Fluent CFD[1] software to simulate odor sources and provide odor concentration data. The olfactory data generated by ANSYS Fluent was exported to Matlab to be used in simulations. One of the environments designed for these simulations is depicted in Fig. 3.12. The dimension of designed arenas for simulations varied from 4×6 m^2 to 30×40 m^2. The airflow was ventilated from the inlet side (left) with different speeds from 0.5 to 20 m/s. In the environments with obstacles, the flow velocity varies in different parts of the arena. Fig. 3.13 shows several 3D snapshots of an odor plume propagation during the time in one of the tested scenarios. As shown in the simulation snapshots, the odor propagation is time variant and under turbulent flow. Although the odor plume was simulated in 3D, the robots move on the floor with their gas sensors always at the same height, so only the odor concentration measured in the 2D plane at the height of the sensors is relevant to the robots’ decisions. We extracted the odor concentrations and airflow velocities of 10

[1] ANSYS Fluent CFD, “FLUENT user’s manual” Software Release, vol. 6, 2006.

centimeters height from the 3-D odor plumes and fed it to the robots in the simulations. Fig. 3.14 and Fig. 3.15 present some snapshots of two extracted 2-D odor plumes in two scenarios.

Robots

Robots were simulated in Matlab as independent entities with no shared variables. The environmental data including odor concentrations, wind speed and obstacles locations are shared among the robots such that the robots can measure the odor concentration and air-flow speed of their places. Robots are able to measure their distances to the obstacles existing in the neighborhood or to the other neighboring robot. The neighborhood range is an adjustable parameter that can be modified in different tests. The wind-biased potential forces (explained in section 3.4) were implemented for the movement control of the robots.

Figure 3.16 shows the virtual forces that the swarm robots generate in the "cross-wind line-up" behavior in different configurations. Each arrow in a place shows the magnitude and the direction of virtual forces that would be applied to another robot if it was located in that place. By adding (or removing) robots to these scenarios the configuration of forces will change, however, these figures only show the virtual forces in the current setup of the figures before adding another robot. These forces are obtained by implementing the equations (3.9) to (3.13). As shown in these figures, the wind direction affects the virtual forces amplitudes. The red "X" marks in each figure show the locations that the virtual forces converge to.

Validation

Fig. 3.17 shows a series of snapshots during a simulation that show the functionality of the system. The first frame of this figure shows that 10 robots are released randomly in one part of the environment. The next frames demonstrate the cross-wind line-up behavior, where they get apart from each other toward the cross-wind direction. The last frame shows that a robot (the red one) gets into the plume and detects it. Since the wind speed was 5 m/s in this example, the individual robots computed their desired distance (D_1) using equation (3.13) as $0.37m$. Fig. 3.17 shows that the robots tend to reach to the analytic optimal configuration. In this test, the coefficient parameters of the method were set as following: $\Delta_d = 1m$, i.e. the range of

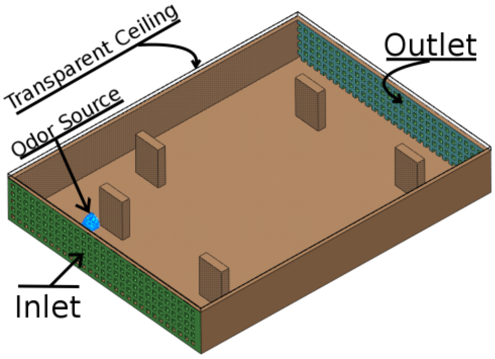


Figure 3.12: The model of a testing environment with $4 \times 6m^2$ dimensions.

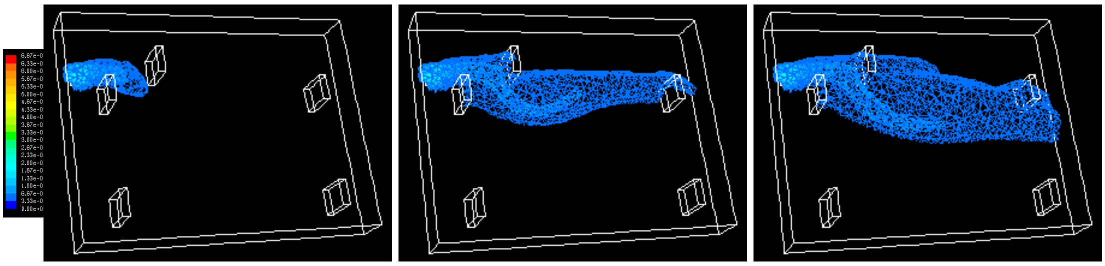


Figure 3.13: ANSYS Fluent three dimension simulations; contours of mass fraction of ethanol propagated in the testing environment of Fig. 3.12.

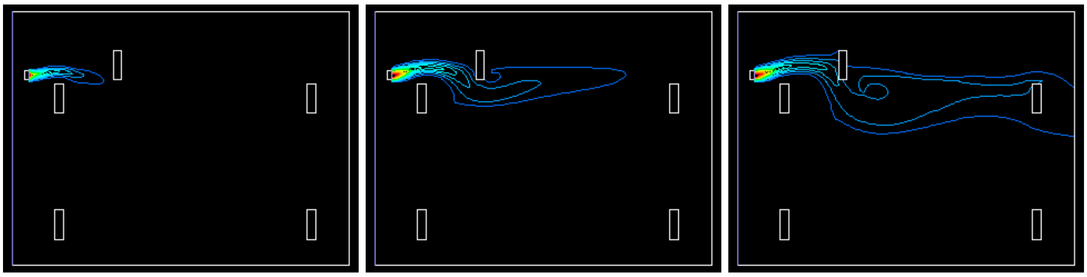


Figure 3.14: Extracted 2-D odor contours of mass fraction of ethanol propagated in the testing environment of Fig. 3.13 during the time.

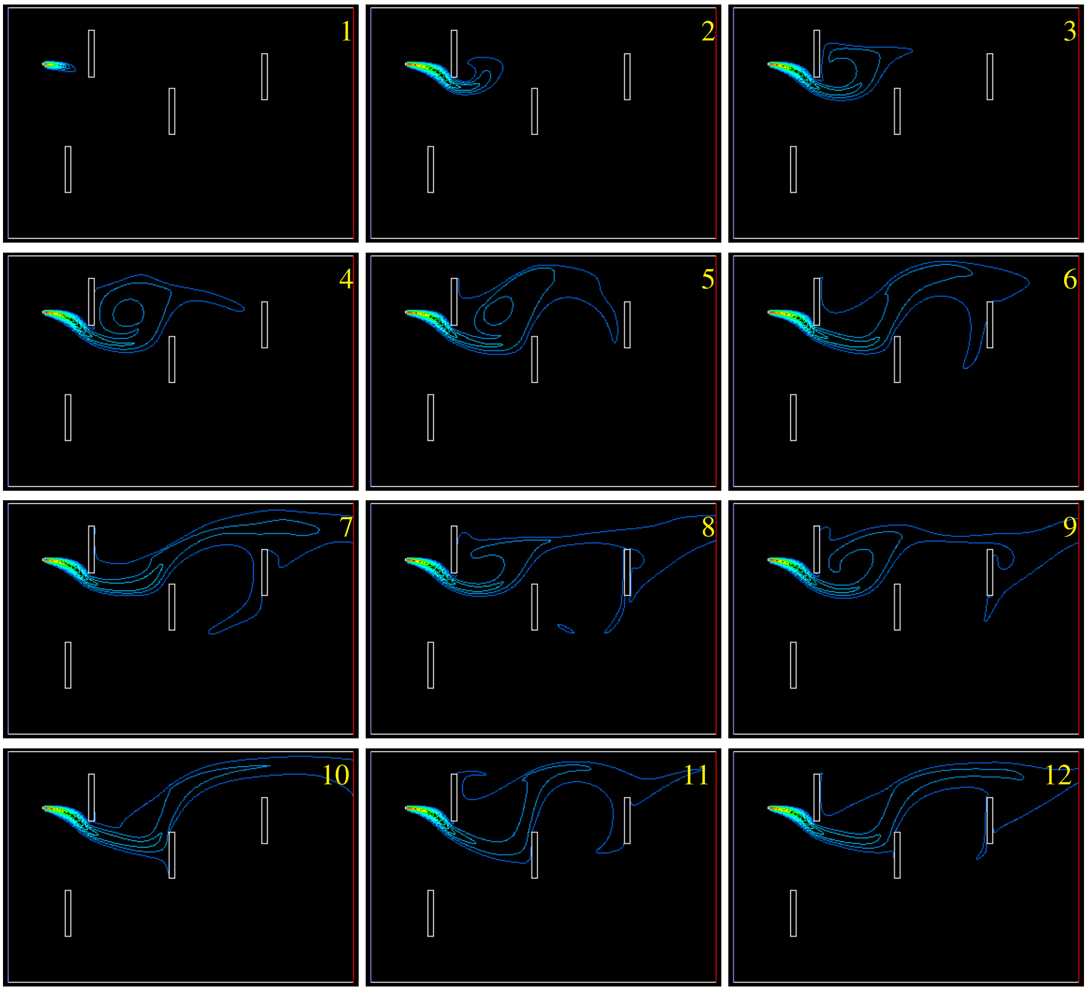


Figure 3.15: Extracted 2-D odor contours of mass fraction of ethanol propagated in a $30 \times 40m^2$ testing environment during the time. The figure shows a part ($10 \times 15m^2$) of this eivronment.

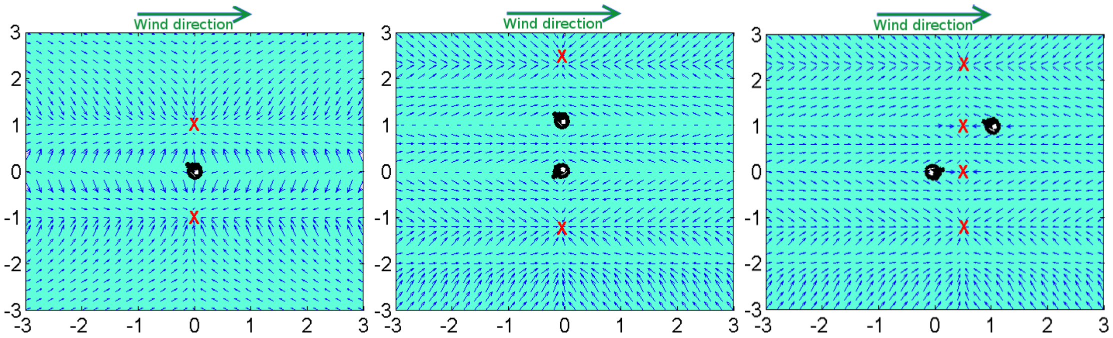


Figure 3.16: Virtual forces generated by robots when the wind direction is left to right. The X marks show the locations that the virtual forces converge to. If another robot is added to this system, it will move to one of the marked places. Note: the robots in the right figure are not on a cross-wind line, thus, their configuration is not stable.

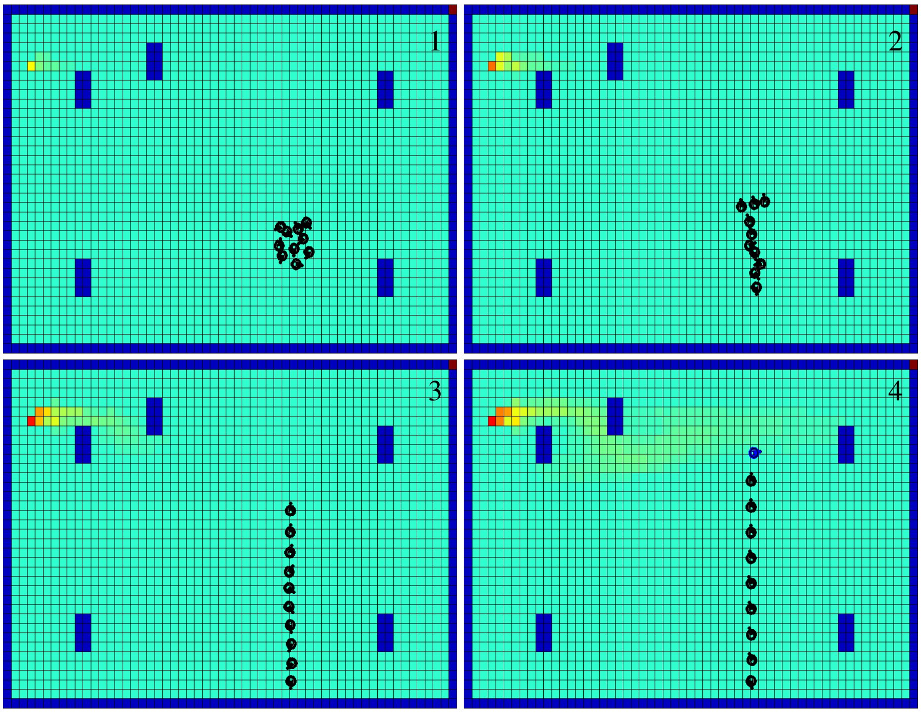


Figure 3.17: 10 swarm robots performing cross-wind line formation.

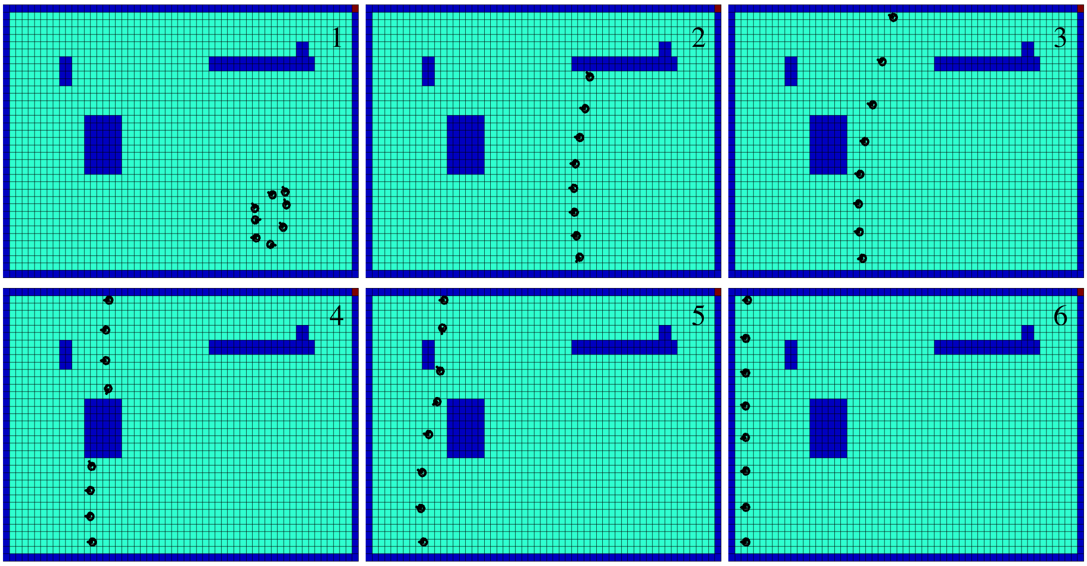


Figure 3.18: Eight swarm robots searching in an environment for possible odor sources. The swarm dynamically changes its topology to deal with environmental changes. There is no odor source in the environment. The airflow is 10 m/s from left to right, $F_{th} = 0.01$ and $\alpha = 0.1$.

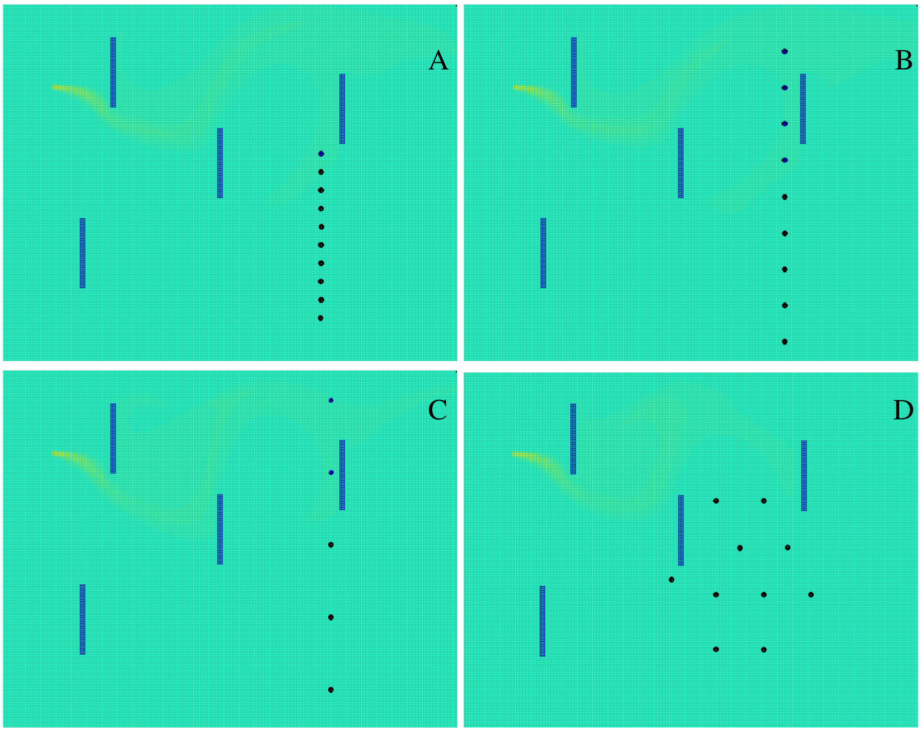


Figure 3.19: A part of a $30 \times 40m^2$ environment. A: line formation, $D_1 = 0.5m$. B: line formation, $D_1 = 1m$. C: line formation, $D_1 = 2m$. D: hyperball formation, distance = $1m$. The odor plume is shown in yellow.

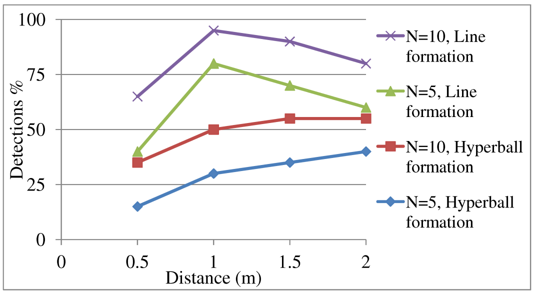


Figure 3.20: Odor plume detection success, during 20 tests in each configuration.

communication between the robots is considered to be 1 meters, $\mu_1 = 2$ and $\mu_2 = \mu_3 = 1, \eta = 0.2, \lambda = 0.1, c_1 = 1$, based on the dynamics of simulated robots to achieve a maximum speed of $0.1\ m/s$.

One simulation that shows the functionality of the method considering the movement for the swarm (described in section 3.4.3) is shown in Fig. 3.18. In this test, we intentionally did not put any odor source in the environment to better demonstrate this behavior. The robots expand toward cross-wind in a line and when they are stable they start to move up-wind. The swarm's topology changes with the environmental changes dynamically. The robots cover a large area towards up-wind, searching for any possible odor plume. In the other simulations and experiments in this section, $F^i_{UpW}(t)$ in equation 3.17 is considered to be zero to disable the up-wind movements of the swarm and only evaluate the formation strategy.

Evaluation

The method was tested in a large environment ($30 \times 40 m^2$) with 5 and 10 robots repeatedly. Fig. 3.19 shows a part of this environment that is $10 \times 15 m^2$ and includes an ethanol source. The release rate was set to $0.01\ g/s$ and the wind speed was $0.5\ m/s$. To evaluate the optimization results we measured the plume detection ability of swarm robots in two different formation strategies; 2-D hyperball formation (similar to [GP04b]) and cross-wind line formation strategy and we manually set the parameter of distance between the robots (D_1) to 0.5, 1 and 2 meters in different tests to find the best configuration. Each test was repeated for 20 times for every formation and value of D_1. If at least one robot could detect the odor plume in less than one minute after the swarm formation was established, we consider a success in plume detection. The number of successfully detecting the odor plume was counted.

The results in Fig. 3.20, show that the best performance between tested configurations is the one with cross-wind line up formation when D_1 is 1 m. On the other hand, using the results of sensor placement optimization in section 3.3.1 (Fig. 3.7.A and equation (3.8)), in the conditions of these simulations, the best formation strategy is line formation with $D_1 = 1.22m$. The best configuration between the simulated ones is very close to the found analytic formation. This validates the analytical optimization conclusions.

3.5.2 Experimental results

In addition to simulations, the method was validated with our currently available equipments.

Realistic environment

The method was tested in a reduced scale unstructured environment shown in Fig. 3.21 (left). This arena, with 3×4 m^2 area by 0.5 meters height, has controlled ventilation through a manifold that extracts air from the testing environment through a honeycomb mesh integrated into one of the walls. The opposite surface of the environment contains a similar mesh that allows the entrance of clean air that flows through the environment. A controlled ethanol gas source using bubblers is pumped to arbitrary places of the environment through a set of PVC tubes. The ethanol release rate was about 0.01 g/s during the tests. The ceiling of this testbed is covered by a sheet of transparent Plexiglas to be visualized from the outside.

The robots

A set of iRobot Roomba robots were upgraded with small laptop computers (ASUS EeePC901) running ROS[1] to control the robots. The wind-biased potential fields formulas were implemented in C and were run in ROS as the control strategy of the robots. The robots were equipped with Laser Range Finders (Hokuyo URG-04LX) for obstacles avoidance. Adaptive Monte Carlo localization (AMCL[2]) libraries were used in ROS to localize the robots in the environments. WifiComm[3] was used in ROS that allows multiple robots to communicate with each other peer to peer through an ad-hoc network. Each robot was equipped with an e2v MiCS-5521[4] and a Figaro TGS2620 gas sensor to measure the odor concentration. The robots repeatedly broadcast their localization data, and accordingly, they measure their x-axis and y-axis distances. In these tests, the airflow was intentionally ventilated and controlled towards the x-axis of the robots and wind speed was manually provided (broadcasted) to the robots. Fig. 3.21 (right) presents one of these developed robots.

[1] http://www.ros.org

[2] http://www.ros.org/wiki/amcl

[3] http://www.ros.org/wiki/wifi_comm

[4] http://www.e2v.com

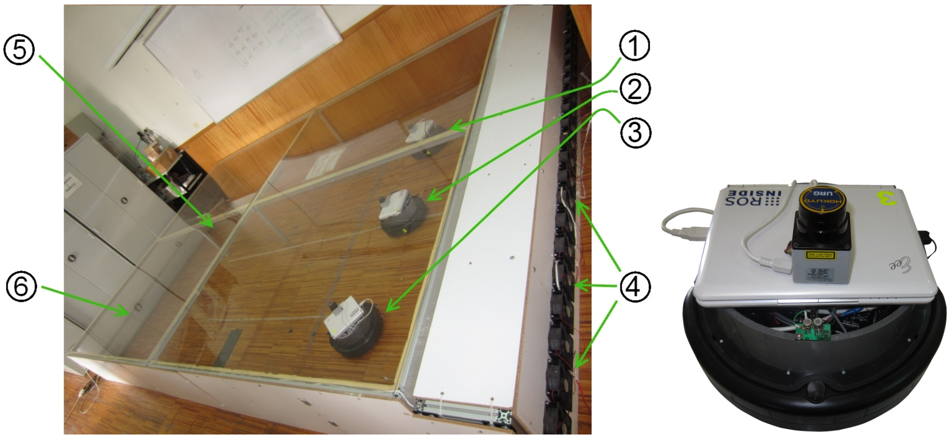


Figure 3.21: Left: the realistic testbed environment. 1,2,3: robots, 4: ventilation system, 5: transparent Plexiglas ceiling, 6: odor source.
Right: One of the developed robots containing gas sensors, Laser Range Finder, and iRobot Roomba controlled by a laptop.

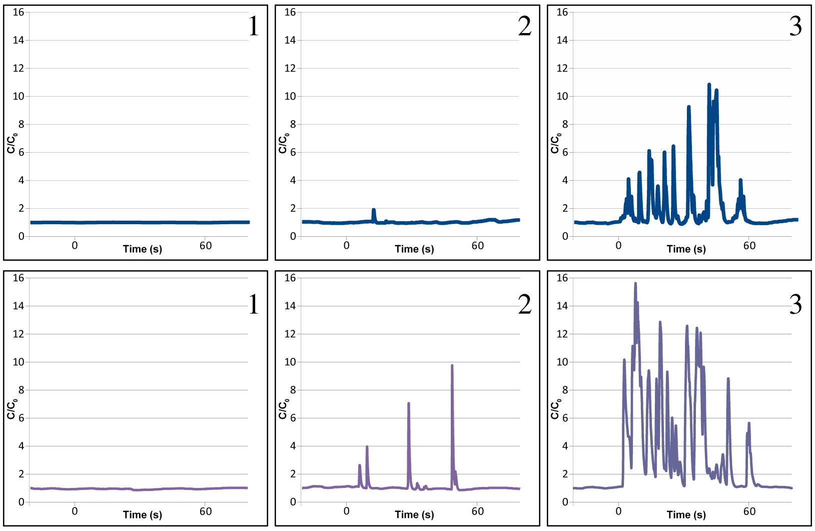


Figure 3.22: The output of e2v sensors of robots 1, 2 and 3 in Fig. 3.21 (left) from left to right in each row respectively. The wind speed was $1 \pm 0.1 m/s$ in the first row and $0.5 \pm 0.1 m/s$ in the second row.

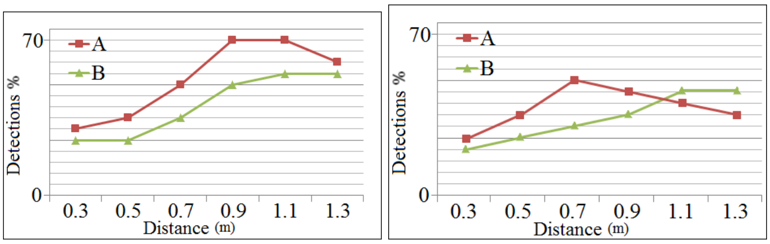


Figure 3.23: Successful detections rate against the distance between the neighboring robots when the wind speed is (left) $0.6 \pm 0.1 m/s$, and (right) $1 \pm 0.1 m/s$. A. line formation, B. hyperball formation.

Validation

Fig. 3.21 (left) shows three robots maintaining cross-wind line-up formation finding an odor plume. The robots spatially construct a line formation in the cross wind direction and maintain a specific distance from each other. If one of the robots moves toward a direction, regardless of the cause of this movement, the other robots dynamically move to maintain the line formation.

Fig. 3.22 is an example that shows the output of the e2v sensors in two tests. Each row shows three graphs corresponding to the three robots of the Fig. 3.21 (left). The first row was taken when the wind speed was $1 \pm 0.1 m/s$ and the second row was taken when the wind speed was $0.5 \pm 0.1 m/s$. In both cases the robot 3 (that was close to the center line of the plume) has detected the odor plume whereas robot 1 did not detect the plume. Robot 2, whose distance to robot 3 is 0.7 m in this example, has detected the plume when the wind speed was $0.5 m/s$ but not when the wind speed was $1 m/s$. In each test if at least one of the robots detects the odor plume we consider a success in plume detection.

Similar to the presented simulations, for evaluating the optimization results, the experiments were done with manual values for D_1 testing cross-wind line formation and also hyperball (triangle) formation. The wind speed was $0.6 \pm 0.1 m/s$ and we set D_1 to 0.3, 0.5, 0.7, 0.9, 1.1 and 1.3m in different tests. Three robots were released randomly about two meters down-wind the source and each test was repeated 15 times and the plume detections were counted. The period of each test was one minute.

The results demonstrated in Fig. 3.23 (left) show that line formation provides more detections and the maximum number of success is reported when the distance between the robots is 0.9 and 1.1 meters. Using equation (3.8), the optimal distance in this configuration is 1.07 meters that agrees with the results of the real experiments.

The experiments in the realistic testbed were repeated by changing the wind-speed to $1 \pm 0.1 m/s$. Fig. 3.23 (right) demonstrates the results and shows that when the robots have line formation and their distance is 0.7, the maximum number of detections will be achieved. Based on equation (3.8), the optimal distance in this configuration is 0.71 meters that again agrees with the results of these experiments.

3.6 Optimal formation of swarm robots in search for odor plumes

This section finds the best formation of mobile robotic gas sensors (i.e., swarm robots equipped with gas sensors) to maximize their sensing coverage area during their crosswind movements.

3.6.1 Mobile gas sensor coverage area

Considering definition 1 for single stationary gas sensor coverage, and assuming crosswind movement for a robotic gas sensor, the binary coverage of the mobile sensor s_i to the point p_j during its crosswind traveling is defined as:

Definition 5 (Single Mobile Gas Sensor Binary Coverage in Crosswind Movement) *Given a sensor s_i, moving crosswind from position (x_i, y_i) to position $(x_i, y_i + L)$, and a point of interest $p_j = (x_j, y_j)$, the coverage of the sensor s_i to the point p_j is defined as:*

$$C_m[s_i, L, p_j] = \begin{cases} 1, & \sum_{l=0}^{L} C_s[s_i(x_i, y_i + l), p_j] > 0 \\ 0, & Otherwise \end{cases} \tag{3.21}$$

where $C_s[s_i(x_i, y_i + l), p_j]$ is given by (3.4) and L is the number of unit steps (grids) that the mobile sensor travels crosswind in the environment. This definition implies that a mobile sensor covers a point if at least in one location during its trajectory it can detect odor patches released from a possible odor source in that point.

With the knowledge of the coverage between sensor s_i and all points of interest, the overall coverage can be defined by aggregation. If there are m points of interest, then the total coverage by a sensor s_i while traveling crosswind is defined as:

Definition 6 (Overall Coverage by a Mobile Gas Sensor in Crosswind Movement) *The overall coverage by a mobile sensor s_i over a region with m points of interest in R^2 while traveling for L unit steps is given by:*

$$cover[s_i, L] = \sum_{j=1}^{m} C_m[s_i, L, p_j] \tag{3.22}$$

where m is the total number of grid cells in an environment. Although the coverage is defined in binary form, its nature is still probabilistic. Fig. 3.25 presents the coverage area of a mobile gas sensor traveling for one and two meters.

Assuming crosswind movement for all N robotic gas sensors, the binary coverage of the sensors to the point p_j is defined:

Definition 7 (Multiple Mobile Gas Sensors Binary Coverage in Crosswind Movement) *Given a set of N sensors s_i, $i = 1...N$, moving crosswind from position (x_i, y_i) to position $(x_i, y_i + L)$, and a point of interest $p_j = (x_j, y_j)$, the coverage of the sensors to the point p_j is defined as:*

$$C_m[S_{(x_i,y_i)_{i=1}^N}, L, p_j] = \begin{cases} 1, & \sum_{l=0}^{L} C_s[S_{(x_i,y_i+l)_{i=1}^N}, p_j] > 0 \\ 0, & Otherwise \end{cases} \tag{3.23}$$

where $C_s[S_{(x_i,y_i+l)_{i=1}^N}, p_j]$ is given by (3.6).

Finally, the overall coverage of N mobile gas sensors traveling crosswind over a region is defined by:

Definition 8 (Overall Coverage by N Mobile Sensors in Crosswind Movement) *The overall coverage by N mobile sensors s_i, $i = 1...N$ traveling crosswind for L unit steps, over a region with m points of interest in R^2 is given by:*

$$cover[S_{(x_i,y_i)_{i=1}^N}, L] = \sum_{j=1}^{m} C_m[S_{(x_i,y_i)_{i=1}^N}, L, p_j] \tag{3.24}$$

This equation implies that the overall coverage is a function of traveling distance of the mobile sensor, sensors' initial positions, source release rate, average wind speed, sensors sensitivity, and distribution standard deviations related to environmental conditions.

Fig. 3.26 shows the coverage area of three mobile gas sensor traveling crosswind for one meter holding various spatial formations. It is obvious that in constant conditions the coverage area of N mobile gas sensors depends on their spatial topology.

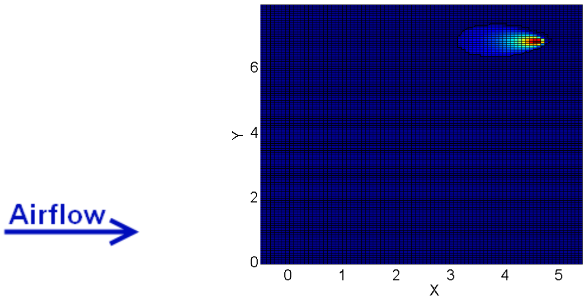


Figure 3.24: Coverage area of a standstill gas sensor when $S_{th} = 0.3$, $k = 10^5$, $Q = 0.01g/s$, and $U = 0.2m/s$ in an environment in Neutral/slightly stable (A-B) conditions. The direction of airflow in this figure is from left to right.

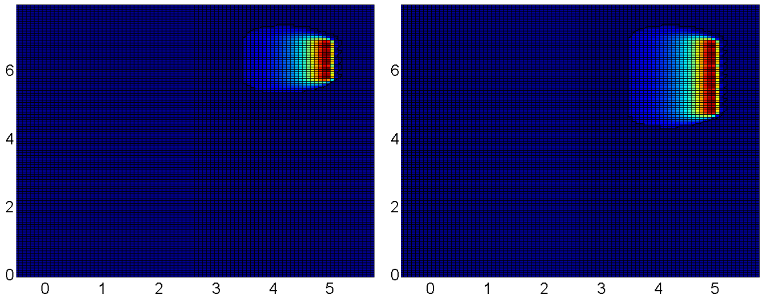


Figure 3.25: Coverage area of a mobile gas sensor traveling for $1m$ (left), and $2m$ (right). The environmental conditions are the same as in Fig. 3.24

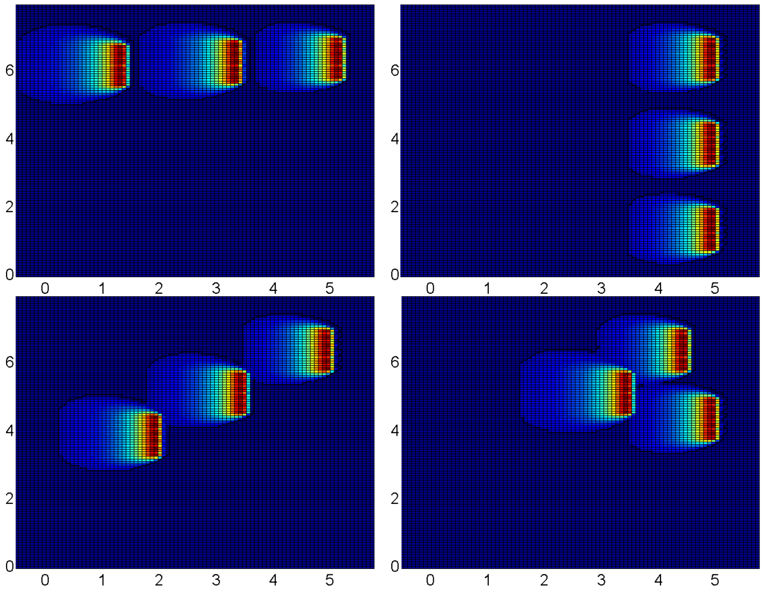


Figure 3.26: Coverage area of three mobile gas sensors traveling for $1m$ with different spatial formations.

Table 3.2: Parameters of optimizations

N	L	U	Env.
2, 3, 6, 10, 16	0.5, 1, 2, 4, 8	0.1-10	A-B, C, D, E-F

3.6.2 Optimal sensor formation

Similar to section 3.3.3, the aim of this section is to find the optimal mobile sensors formation in a way that overall coverage is maximized. Thus, we look for a series of sensors positions $s_i = (x_i, y_i)$ such that:

$$\{s_1, s_2, ..., s_N\} = \text{arg max } cover[S_{(x_i,y_i)_{i=1}^N}, L]$$

Optimal sensor formations are the configurations where the coverage area of the sensors is maximized. Therefore maximizing the area of sensor coverage, defined in (3.24), is used as the criterion of our optimization. We optimize this criterion with various number of sensors and different average wind speeds in four environmental conditions considering various traveling distances (L) for the mobile sensors. All considered assumptions in this section are similar to section 3.3.3 and again the Powell's conjugate gradient descent method was used (in Matlab) to optimize this problem.

3.6.3 Optimization results of mobile gas sensor formation

The optimal coverage area was measured for different number of sensors from 2 to 16 traveling cross wind for 0.5, 1, 2, 4 and 8 meters, and different wind speeds from 0.1 to 10 m/s in the four environmental conditions (see Table 3.2). The topological shape of the sensors in the optimal solutions was analyzed in each case. Fig. 3.27 shows examples of optimized formation of three mobile sensors while traveling one meter crosswind ($L = 1m$) with different values for the wind speed. Fig. 3.28 shows another example of optimized configuration of two to five sensors and their maximum coverage area in an urban environment under neutral/slightly stable (A-B) conditions when the wind speed is equal to 1 m/s and the swarm crosswind movement (L) was 1 m. Different values of U, N and L in different environmental conditions result in similar (but not equal) solutions. One interesting point from all of the optimized solutions is that:

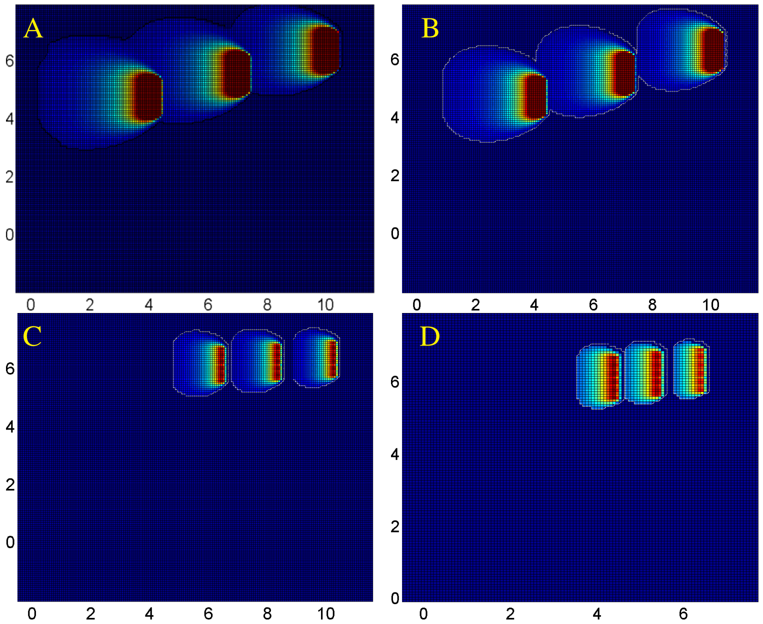


Figure 3.27: The optimized formation of three mobile gas sensors in an area when the wind speed is A: 0.2, B: 0.3, C: 1 and D: 3 m/s in A-B environmental conditions.

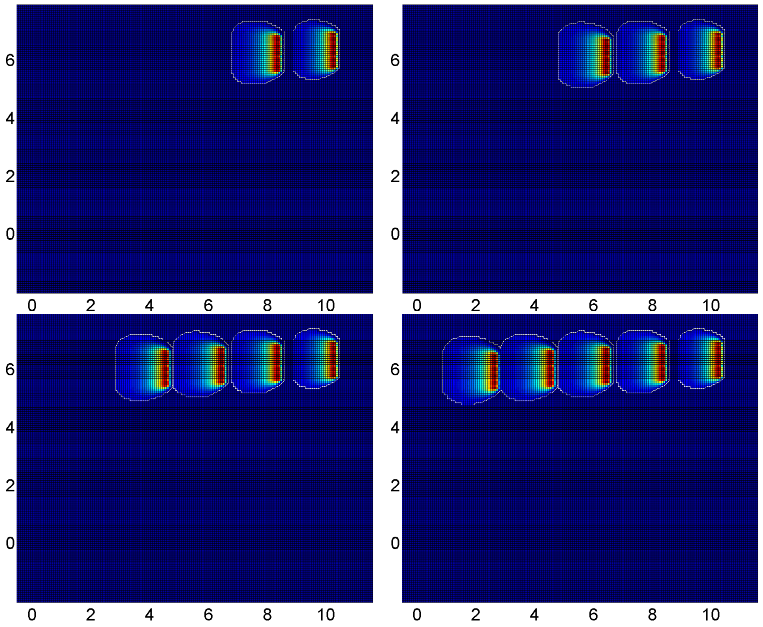


Figure 3.28: The optimized formation of 2 to 5 mobile gas sensors in an area when the wind speed is $1m/s$ in A-B environmental conditions.

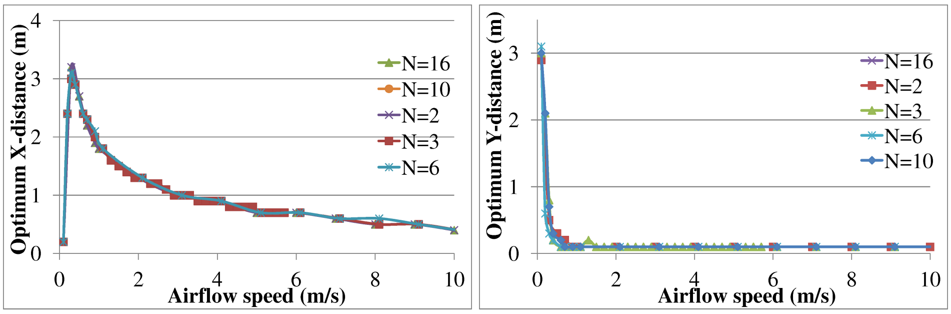


Figure 3.29: The X and Y components of optimal distance between the neighboring sensors in the optimized configuration in A-B environmental conditions when the number of sensors (N) is 2 to 16 and the swarm's crosswind movement distance (L) is 1*m*.

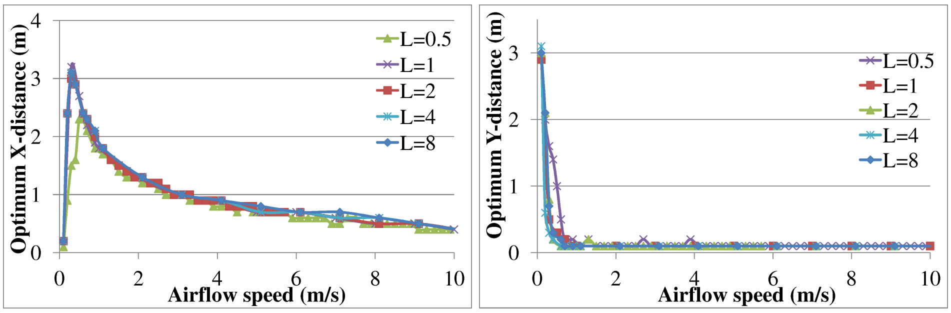


Figure 3.30: The X and Y components of optimal distance in A-B environmental conditions when N = 10 and L varies from 0.5 to 8*m*.

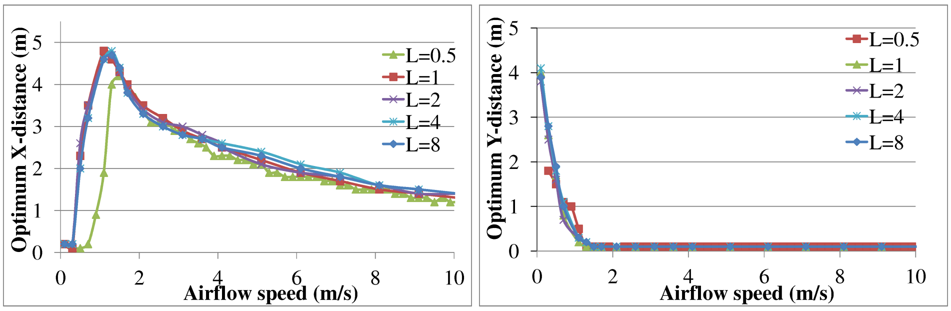


Figure 3.31: The X and Y components of optimal distance in E-F environmental conditions when N = 10 and L varies from 0.5 to 8*m*.

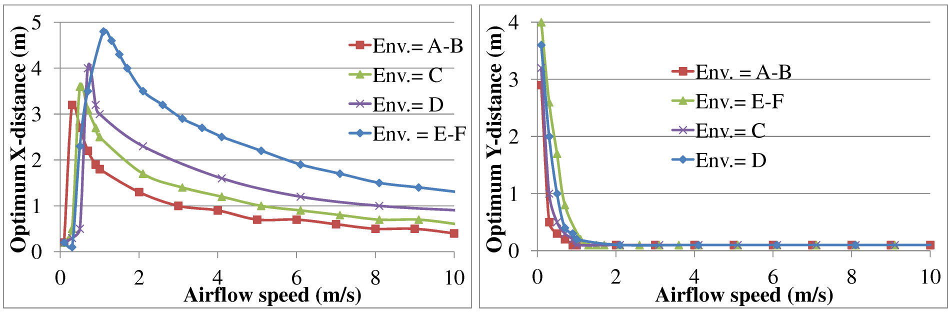


Figure 3.32: The optimal distance between the neighboring sensors in the optimized configurations in different environmental conditions. The chart is the average of the results for N = 2 to 10 and L = 1 to 8*m*.

Conclusion 3 (Diagonal Line Topology) *The formation of all of the optimal solutions for mobile sensors is* ***diagonal line*** *configuration, with equal distance between each pair of neighboring sensors.*

The distance between the robots in the optimal formations is called optimal distance. We study the X and Y components of the optimal distance in every optimization set (listed in Table 3.2).

Fig. 3.29 presents some results of the optimizations that shows the X and Y components of optimal distance between the neighboring sensors in optimized configurations in neutral/slightly stable (A-B) environmental conditions when the number of sensors is 2 to 16 and the swarm's crosswind movement distance L is $1m$. This chart shows that in this conditions, changing N from 2 to 16 does not affect the optimal distance. Although this chart only shows the optimal results for one set of parameters, in other conditions, changing N from 2 to 16 does not show a significant change in the optimization results as well. By analyzing these results of the optimizations, it can be seen that in constant wind speed, when the number of sensors changes (from 2 to 16), the optimal distance between the sensors changes only for a few centimeters and is almost constant even when the swarm movement distance is changed (see the examples in Fig. 3.28).

Fig. 3.30 presents another set of optimization results when $N = 10$, $Env. = A - B$ type, U varies from 0.1 to $10m/s$ and L has different values form 0.5 to $8m$. This chart shows that different values of L (swarm's crosswind movement distance) provides [almost] the same optimization results. Only when L is very small (L=0.5) and the wind speed is less than a certain value (about 2 m/s) the optimal distances show different results. For other values of optimization parameters similar results were obtained. Fig. 3.31 presents the optimization results in another environmental conditions that shows similar conclusions.

By considering the results in Fig. 3.29, Fig. 3.30 and Fig. 3.31 we can conclude that:

Conclusion 4 (Wind & Environment Dependent Distance) *The distance between neighboring mobile sensor pairs in optimal configurations depends mainly on the wind speed and the environmental conditions, whereas, the number of sensors and the swarm's crosswind movement distance do not show significant impact on optimal configurations.*

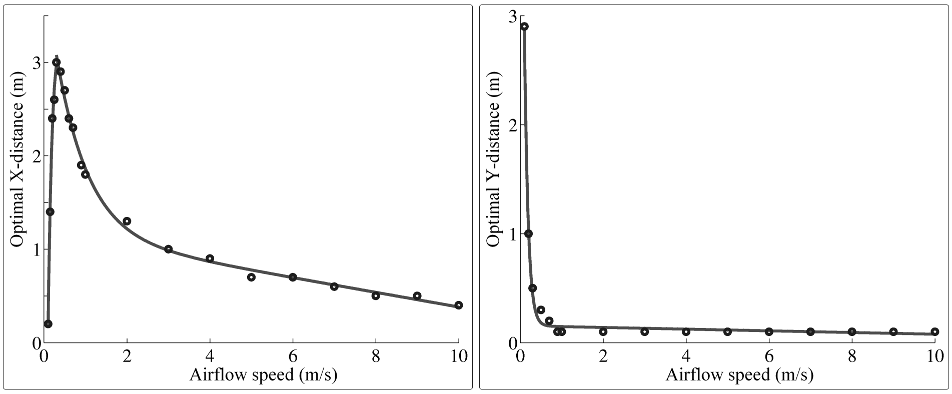


Figure 3.33: The average optimal distance between neighboring sensors in A-B environmental conditions and an non-linear regression estimation.

These conclusions are drawn after the results obtained from numerical optimizations and are among the most significant contributions of this thesis.

Fig. 3.32 shows the effect of environmental conditions on the optimization results. This chart is the average result of optimizations when $N = 2$ to 16 and $L = 1$ to $8m$. This chart shows that in an environment under neutral/slightly stable conditions the distance (specially X-component) between the neighboring sensors is larger than in an environment under strong inversion conditions. The results in Fig. 3.32 show that in fixed environmental conditions, when the wind speed is less than a certain value the higher wind speed, the larger the optimal X-distance. The certain values (peaks in the graphs) depend on the environmental conditions. It is 0.3 for A-B, 0.5 for C, 0.7 for D, and 1.1 for E-F conditions. However when the wind speed is more than the certain value the higher wind speed, the smalled the optimal X-distance. Therefore, as the wind speed increases, the X-distance in-between the sensing nodes should decrease in order to maintain optimal coverage. When the wind is smaller than mentioned certain values, the Y-distance is very high and it decreases when the wind speed increases. The Y-distance dramatically decreases and converges to a value near $0.1m$ when the wind speed is higher than the certain values.

Considering the results shown in Fig. 3.32 and using a non-linear regression analysis, the following analytical equation was obtained that describes the optimal distance between the sensing nodes as a function of the wind speed in neutral/slightly stable environmental conditions.

$$D_x = \begin{cases} -9.6e^{-8.8U} - 2.6U + 4.4, & U \leq 0.3 \\ 2.8e^{-1.3U} - 0.07U + 1.1, & U > 0.3 \end{cases} \tag{3.25}$$

$$D_y = 8.3e^{-11.1U} - 0.007U + 0.15 \tag{3.26}$$

The graph in Fig. 3.33 is the fitted function (3.25) and (3.26) while the scattered circles are the average results of the optimizations in A-B environmental conditions. The mean square error of this regression line is 4×10^{-3}. For the other environmental conditions, estimation functions can be calculated in the same way. This function is later used by moving senor robot to estimate the optimal distance based on the wind speed.

It should be mentioned that the obtained results are valid for specific values of S_{th}, k and Q defined in section 3.6.2, however for other values the optimal configuration for the sensors is the same (i.e., a line) and only the values of the optimal distance between the sensors and regression function (3.25) are changed. This process of optimization can be repeated and optimal results can be achieved in other conditions.

3.7 The proposed method, wind-biased potential fields, “diagonal line-up” behavior

Conclusions 3 and 4 state that the robots should line-up with equal distances from each other while moving cross-wind, to maximize the probability of the finding odor plumes. Despite in section 3.3, here the line is not always towards the cross-wind. It is a diagonal line composed by the robots which their x-distance components and y-distance components are given by equations (3.25) and (3.25) (in the mentioned environmental conditions). Similar to section 3.4 we define a behavior named “diagonal line-up” for the individual robots in order to implement diagonal line formation for the swarm. This behavior defines two types of virtual forces that are applied to the robots; robot-to-robot and robot-to-environment forces.

3.7.1 Robot-to-robot forces

Similar to section 3.4, each robot measures the air-flow direction $\overrightarrow{U}$ and assumes this direction as its internal X-axis coordinate system and then it measures the relative distance to its neighboring robots. Then, the robots try to maintain a specific distance at their X-axis from their neighbors and maintain another specific distance distance with them at their Y-axis. Therefore,

$$< \overrightarrow{X}\ axis > \ \equiv \ < \overrightarrow{U} > \tag{3.27}$$

$$\overrightarrow{F}^{ij}_{sl}(t) = \overrightarrow{Fx}^{ij}_{sl}(t) + \overrightarrow{Fy}^{ij}_{sl}(t) \tag{3.28}$$

$$\overrightarrow{Fy}^{ij}_{sl}(t) = \begin{cases} -\mu_4\Big(\|\overrightarrow{Y}_{ij}\| - D_3\Big)\Big[\frac{1}{\|p_{ij}(t)\|^2}\Big]\Big[\frac{\overrightarrow{Y}_{ij}}{\|\overrightarrow{Y}_{ij}\|}\Big]\,, & 0 < \|\overrightarrow{Y}_{ij}(t)\| < D_3 \\ -\mu_5\Big(\|\overrightarrow{Y}_{ij}\| - D_3\Big)\Big[\frac{1}{\|p_{ij}(t)\|^2}\Big]\Big[\frac{\overrightarrow{Y}_{ij}}{\|\overrightarrow{Y}_{ij}\|}\Big]\,, & D_3 < \|\overrightarrow{Y}_{ij}(t)\| < D_4 \\ 0\,, & \|\overrightarrow{Y}_{ij}(t)\| > D_4 \end{cases} \tag{3.29}$$

$$\overrightarrow{Fx}^{ij}_{sl}(t) = \begin{cases} -\mu_6(\overrightarrow{X}_{ij} - D_5)\Big[\frac{1}{\|p_{ij}(t)\|^2}\Big]\,, & 0 < \|\overrightarrow{X}_{ij}(t)\| < D_4 \quad \& \quad \overrightarrow{Y}_{ij} \geq 0 \\ -\mu_6(\overrightarrow{X}_{ij} + D_5)\Big[\frac{1}{\|p_{ij}(t)\|^2}\Big]\,, & 0 < \|\overrightarrow{X}_{ij}(t)\| < D_4 \quad \& \quad \overrightarrow{Y}_{ij} < 0 \\ 0\,, & \|X_{ij}(t)\| > D_4 \end{cases} \tag{3.30}$$

$$D_5 = \begin{cases} -9.6e^{-8.8U} - 2.6U + 4.4, & U \leq 0.3 \\ 2.8e^{-1.3U} - 0.07U + 1.1, & U > 0.3 \end{cases} \tag{3.31}$$

$$D_3 = 8.3e^{-11.1U} - 0.007U + 0.15 \tag{3.32}$$

where

- $\overrightarrow{F}^{ij}_{sl}(t)$ is the applied force to robot i by robot j at time t. $\overrightarrow{Fx}^{ij}_{sl}(t)$ and $\overrightarrow{Fy}^{ij}_{sl}(t)$ are respectively the x and y components of $\overrightarrow{F}^{ij}_{sl}(t)$.

- μ_4, μ_5 and μ_6 are constant coefficients for tuning acceleration of the robots. μ_4 is the Y-component repulsing coefficient and μ_5 is the Y-component attracting coefficient while μ_6 is the X-component coefficient.

- D_3 is a design parameter that specifies the desired Y-component distance interval between the neighboring robots. We defined D_3 using the equation (3.26) to be equal to the optimization results.

- D_5 is a design parameter that specifies the desired X-component distance interval between the neighboring robots. We defined D_5 using the equation (3.25).

- D_4 defines the margin of the area that a robot applies forces to the other robots. Logically for line formation D_4 should be bigger than $max(D_3, D_5)$ and smaller than $2 \times max(D_3, D_5)$. Moreover, $0 < D_3, D_5 < D_4 < \Delta_d$.

The above equations are designed inspired by the Hooke’s law, thus, the forces are similar to the forces in the physical springs. Hence, the robots try to maintain their X-component distance to D_5 and to maintain a distance of D_3 in their Y-component distance. The result will be a diagonal line that its slope is D_3/D_5).

Using the above equations, the total “diagonal line-up” force $\overrightarrow{F}^i_{sl}(t)$ for robot i is determined as:

$$\overrightarrow{F}^i_{sl}(t) = \sum_{j=1;j\neq i}^{N} \overrightarrow{F}^{ij}_{sl}(t) \tag{3.33}$$

3.7.2 Robot-to-environment forces

The robot-to-environment forces are exactly the same as the ones defined in section 3.4.2 so they will not be repeated here.

3.7.3 Swarm movements

It is desired for the swarm robots to hold the “diagonal line-up” formation and move cross-wind. Hence, they will sweep and cover the environment toward cross-wind. To do so, each robot should move in the space in the correct direction. Here we set the $\overrightarrow{F}^i_G(t)$ equal to $\overrightarrow{F}^i_{CrW}(t)$ that is defined as :

$$\overrightarrow{F}^i_G(t) = \overrightarrow{F}^i_{CrW}(t) = \overrightarrow{Fx}^i_{CrW}(t) + \overrightarrow{Fy}^i_{CrW}(t) \tag{3.34}$$

$$\overrightarrow{Fx}^i_{CrW}(t) = 0 \tag{3.35}$$

$$\overrightarrow{Fy}^{i}_{CrW}(t) == \begin{cases} 0\,, & ||\overrightarrow{F}^{i}_{sl}(t)|| > Fs_{th} \\ F_{\beta}\,, & ||\overrightarrow{F}^{i}_{sl}(t)|| \leq Fs_{th} \end{cases} \tag{3.36}$$

where $\overrightarrow{Fx}^{i}_{CrW}(t)$ and $\overrightarrow{Fy}^{i}_{CrW}(t)$ are respectively the X and Y competent of the defined cross-wind force $\overrightarrow{F}^{i}_{CrW}(t)$. Fs_{th} is a threshold value for the forces applied to a robot and F_{β} is a constant predefined force. The above formula checks if $||\overrightarrow{F}^{i}_{sl}(t)||$ is bigger than a defined threshold or not. If $||\overrightarrow{F}^{i}_{sl}(t)||$ is very small it means that the resultant virtual forces applied to robot i are near zero, i.e, the topology of the robot and its neighbors is in the form of a diagonal line and it is in its steady state. In this case a force (F_{β}) toward Y-axis direction (i.e, cross-wind) is applied to the robot and robot moves toward cross-wind direction.

3.7.4 The total force

The total applied force to a robot in "diagonal line-up" behavior is:

$$\overrightarrow{F}^{i}_{sm}(t) = \overrightarrow{F}^{i}_{sl}(t) + \overrightarrow{F}^{i}_{obs}(t) + \overrightarrow{F}^{i}_{G}(t) \tag{3.37}$$

Similar to section 3.4.4, the desired direction of motion of robot $\theta^{i}_{d}(t)$, a controller for the orientation dynamics of the robot $w_i(t)$ and the next velocity of the robot $\overrightarrow{v}_i(t)$ can be calculated by replacing $\overrightarrow{F}^{i}_{s}(t)$ by $\overrightarrow{F}^{i}_{sm}(t)$ in equations 3.18, 3.19 and 3.20. Algorithm 9 presents the pseudo-code that should be run in each robot to perform this behavior.

Algorithm 9: Diagonal line-up behavior

```
Behavior = Diagonal Line-up
while Behavior == Diagonal Line-up do
    $c_i$ = Measure odor concentration()
    $U_i$ = Measure air-flow velocity()
    $\{P_i\}$ = Measure distances to the neighbors()
    $\{d_i\ ,\ Vec_i\}$ = Measure distances and direction to the obstacles()
    Navigate while holding formula(3.37)
    if ($c_i$ > odor_threshold) || (A robot indicates finding odor plume) then
        Broadcast(plume detected)
        Behavior = Plume_tracking
```

3.8 Validation of “diagonal line-up” behavior

The presented method was validated in both simulations and realistic experiments.

3.8.1 Simulations

The method was tested in several different simulation environments containing obstacles with different number of robots. Most of the details are similar to the simulations details explained in section 3.5.1. Therefore this section only provides the differences and the results.

Testing environment

Similar to section 3.5.1, the environments were imported to ANSYS Fluent CFD and the odor concentrations and airflow velocities of 10 centimeters height from the 3-D odor plumes were extracted and fed to the robots in the Matlab simulations. Since the wind in these simulations was stable and laminar toward one direction, this environments are under neutral and slightly stable conditions.

Robots

Robots were simulated in Matlab exactly similar to section 3.5.1. The only difference is the wind-biased potential forces. The virtual forces in section 3.7 were implemented in the robots in order to demonstrate the “diagonal line-up” behavior. Figures 3.34 and 3.35 show the virtual forces that the swarm robots generate in the “diagonal line-up” behavior in different configurations. By adding (or removing) robots to these scenarios the configuration of forces will change, however, these figures only show the virtual forces in the current setup of the figures before adding another robot. These forces are obtained by implementing the equations (3.27) to (3.32).

Validation

Fig. 3.36 shows a series of snapshots during a simulation that show the functionality of the “diagonal line” behavior and swarm’s cross-wind movement and obstacle avoidance behavior.

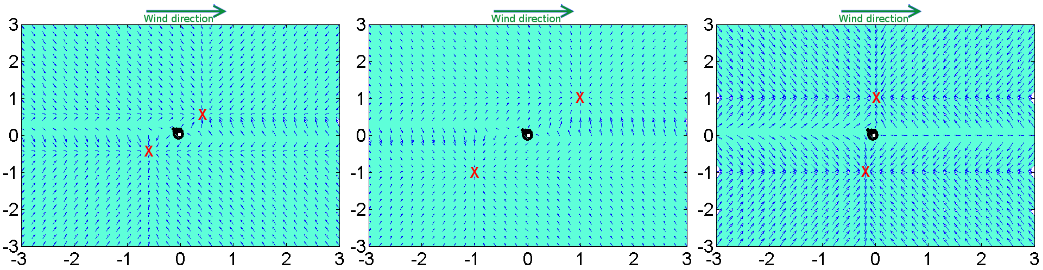


Figure 3.34: Virtual forces generated by a robot when the wind direction is left to right. Left: $D_3 = 0.5m$, $D_5 = 0.5m$, middle: $D_3 = 1m$, $D_5 = 1m$ and right: $D_3 = 1m$, $D_5 = 0.15m$. The X marks show the locations that the virtual forces converge to. If another robot is added to this system, it will move to one of the marked places.

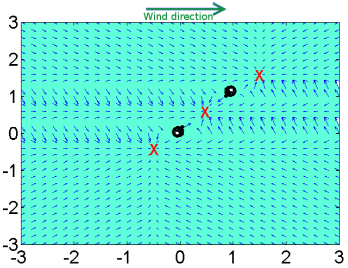


Figure 3.35: Virtual forces generated by two robots while $D_3 = 0.5m$, $D_5 = 0.5m$. If a robot is added to this system it will converge to one of the X points. This configuration is not stable and the robots are under attraction forces toward each other.

In this scenario there was no odor source so the robots maintain these behaviors and do not find anything.

Fig. 3.37 shows another scenario that ten robots perform the same behaviors in an environment including an odor source. The first frame of this figure shows 10 robots released randomly in one part of the environment. The next frames demonstrate the “diagonal line-up” behavior, where they get apart from each other toward a diagonal line. The robots maintain the diagonal line formation and move towards cross-wind and avoid the obstacles. The last frame shows that a robot (the red one) gets into the plume and detects it. In this scenario D_3 and D_5 were set manually to $D_3 = D_5 = 0.3m$ and D_4 was set to $0.45m$. The rest of the simulation parameters were set to the same values of the simulation in Fig. 3.17.

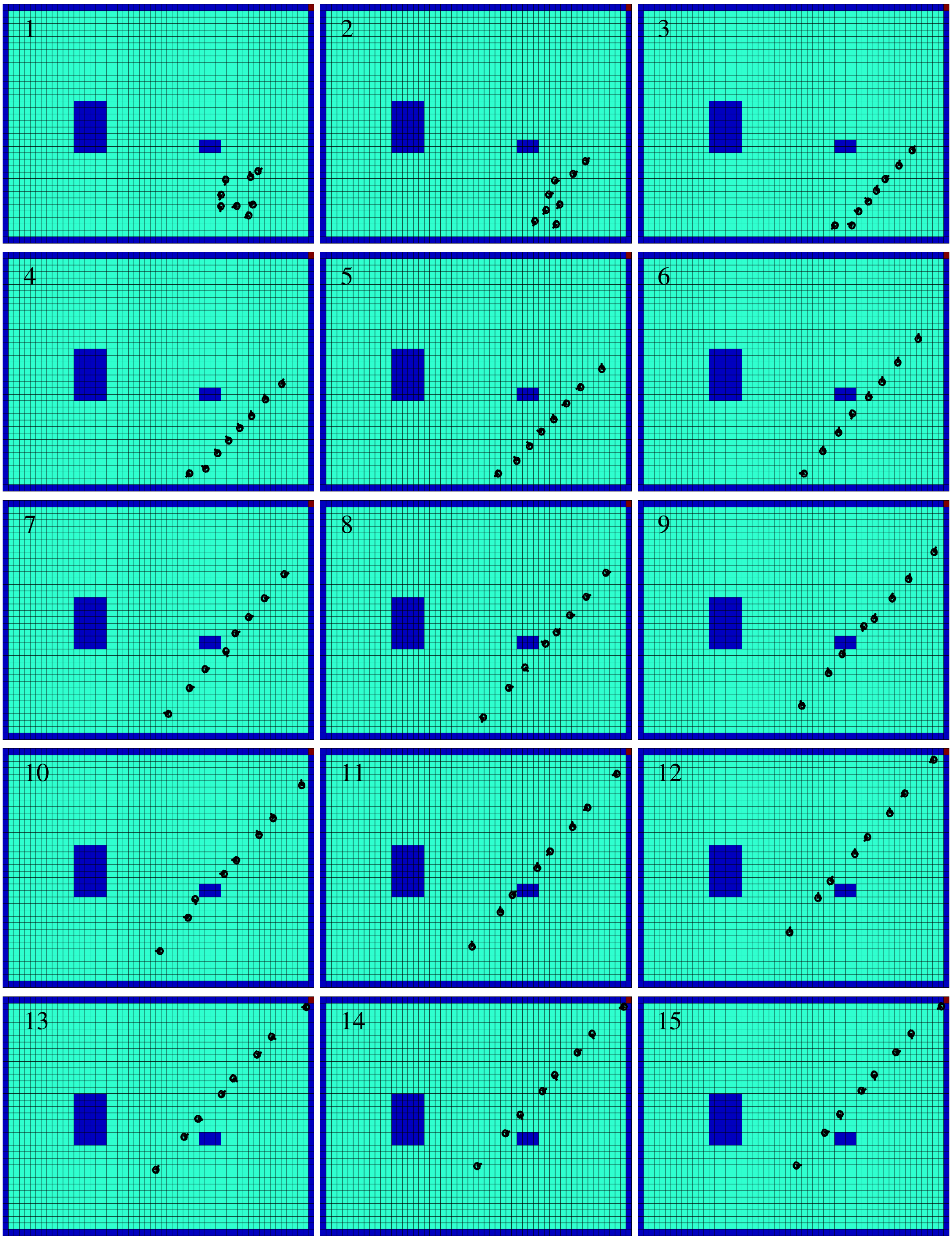


Figure 3.36: Eight swarm robots forming "diagonal line" formation and move cross-wind while avoiding obstacles. The swarm dynamically changes its topology to deal with environmental changes. There is no odor source in the environment and the wind is ventilated from left to right. In this simulation, the environment is $4 \times 6m^2$ and we set $D_3 = 0.4m$ and $D_5 = 0.4m$.

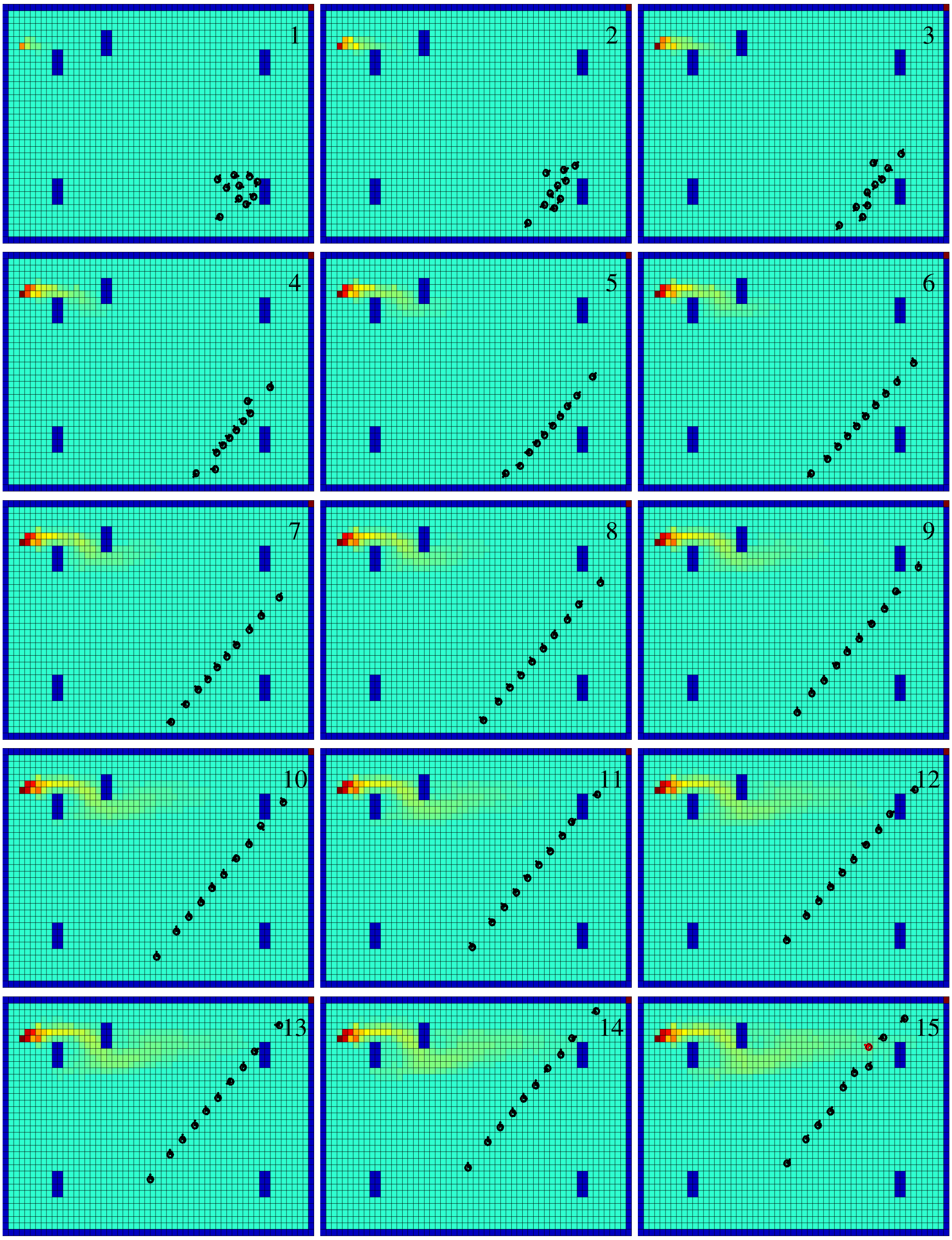


Figure 3.37: Ten swarm robots searching in an environment for possible odor sources. The swarm dynamically changes its topology to deal with environmental changes. There is no odor source in the environment. In this scenario: $D_3 = 0.3$, $D_5 = 0.3$, $D_4 = 0.45$,$Fs_{th} = 0.01$ and $F_\beta = 0.1$.

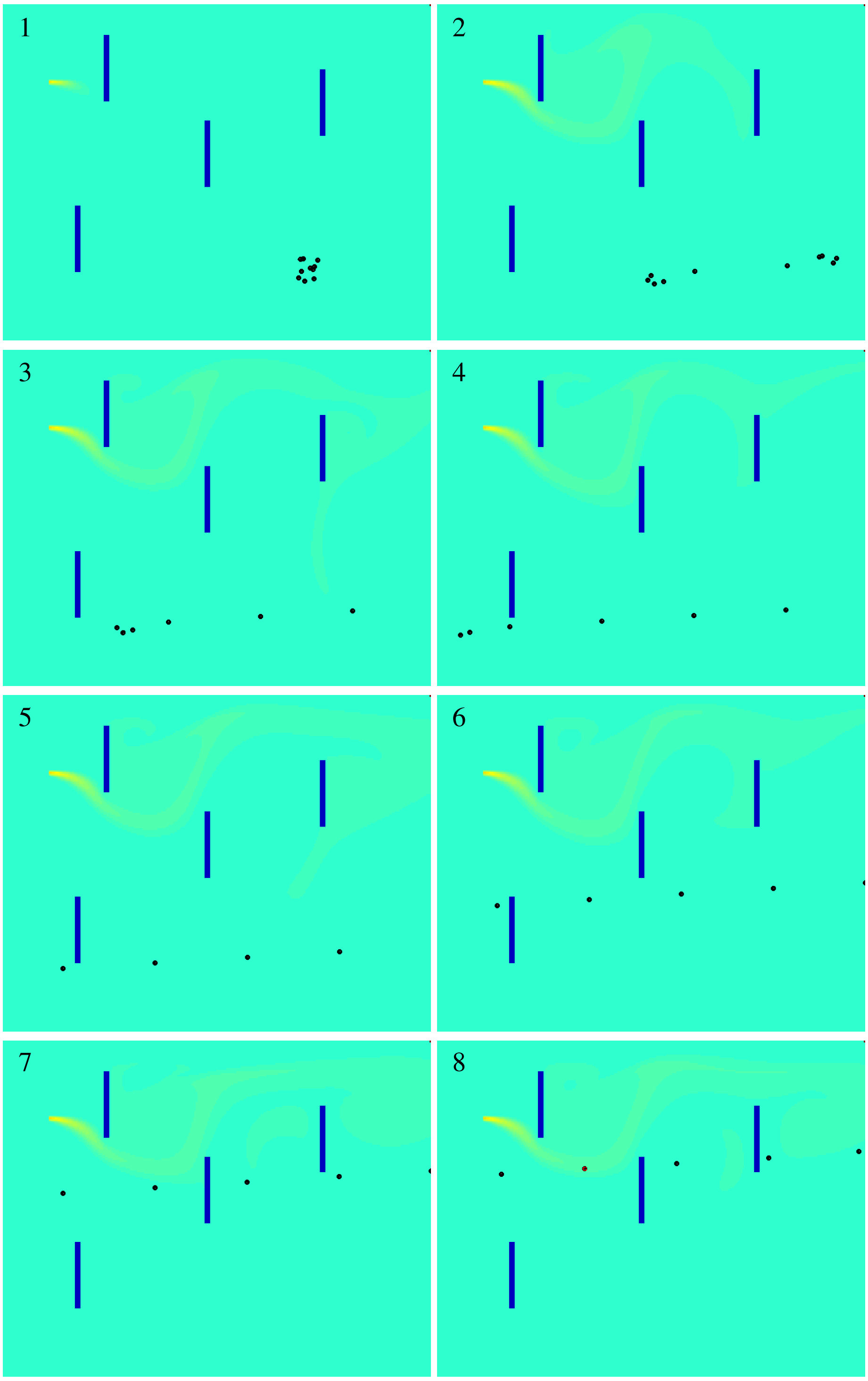


Figure 3.38: Ten swarm robots searching in a large environment for possible odor sources. The swarm dynamically changes its topology to deal with environmental changes. The airflow is 0.5 m/s from left to right. The robots perform the diagonal line-up behavior until one of them finds the plume.

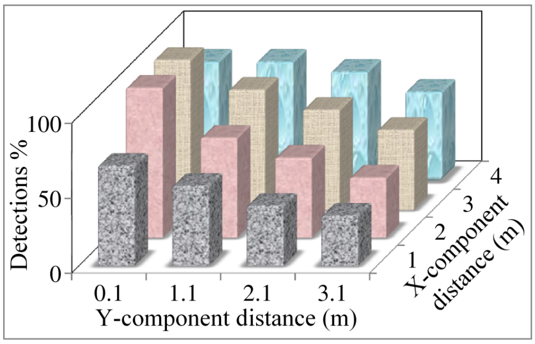


Figure 3.39: Odor plume detection success of diagonal line-up behavior , during 15 tests using 10 swarm robots in each configuration.

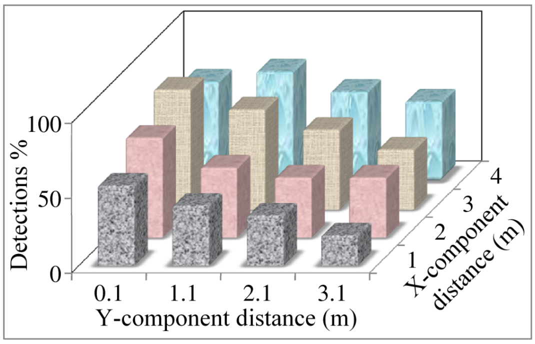


Figure 3.40: Odor plume detection success of diagonal line-up behavior, during 15 tests using 5 swarm robots in each configuration.

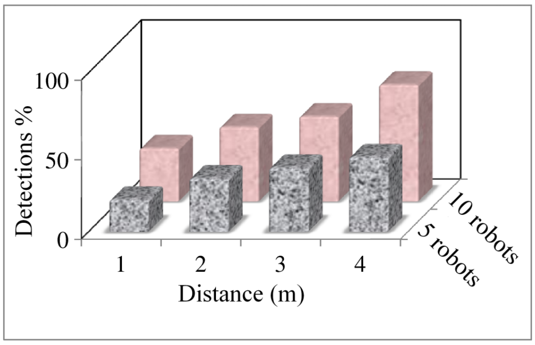


Figure 3.41: Odor plume detection success of hyper-ball behavior, during 15 tests using 5 and 10 swarm robots in each configuration.

Evaluation

The method was tested in a large environment ($30 \times 40m^2$) with 5 and 10 robots repeatedly. Fig. 3.38 shows a part of this environment that is $10 \times 15m^2$ and includes an ethanol source. The release rate was set to 0.01 g/s and the wind speed was 0.5 m/s. The robots perform the diagonal line-up behavior until one of them finds the plume. Similar to section 3.5.1, to evaluate the optimization results, we measured the plume detection ability of swarm robots in two different formation strategies: 2-D hyperball formation and diagonal line formation strategy. In diagonal line formation, we manually set the parameter of distance between the robots (D_5) to 1, 2, 3 and 4 and (D_3) to 0.1, 1.1, 2.1 and 3.1 meters in different tests to find the best configuration. In 2-D hyperball formation the desired distance between neighboring robots was set to 1, 2, 3 and $4m$ (similar to D_5). Each test was repeated for 15 times for every formation and value of D_3 and D_5. If at least one robot could detect the odor plume in less than one minute after the swarm formation was established, we consider a success in plume detection. The number of successfully detecting the odor plume was counted.

The results, in Fig. 3.39, show that the best performance between tested configurations is the one with diagonal line up formation when D_3 is $0.1m$ and D_5 is $2m$ or $3m$. On the other hand, using the results of optimization in section 3.3.1 (Fig. 3.7.A and equation (3.8)), in the conditions of these simulations, the best formation strategy is diognal line formation when D_3 is $0.17m$ and D_5 is $2.52m$. The best configuration between the simulated ones are very close to the found analytic formation. This validates the optimization achievements.

3.8.2 Experimental results

In addition to simulations, the method was experimented with our currently available robotic facilities.

Realistic environment

The method was tested in the reduced scale environment shown in Fig. 3.43. This arena was already explained in section 3.8.2.

The robots

A set of LSE miniQ[1] robots were developed at our laboratory based on the 2WD miniQ[2] platform. The motivation behind this was to build cheap and simple robotic units for swarm olfactory research.

The LSE miniQ (presented in Fig. 3.42) communicates with a host computer using XBee[3]. The host computer runs ROS programs (nodes) to control the robots. Each robot is controlled by an individual ROS node. A single computer can run several ROS nodes, i.e., a swarm of LSE miniQs. Each robot contains an e2v MiCS-5524 gas sensor to allow the detection of volatile organic compounds. Two LEDs (one blue and one red) placed on the top of the robot are used for visual tracking by a camera mounted at the ceiling of the laboratory. The SwisTrack[4] software is used to track and localize the robots. The robots repeatedly broadcast their localization data, and accordingly, they measure their x-axis and y-axis distances. In these tests, the airflow was intentionally ventilated and controlled towards the x-axis of the robots and wind speed was manually provided (broad-casted) to the robots.

Validation

Fig. 3.44 shows four robots maintaining diagonal line-up formation finding an odor plume. The robots spatially construct a diagonal line topology and move towards cross-wind direction and

[1] http://lse.isr.uc.pt/news/lseminiqrobot
[2] http://www.dfrobot.com
[3] http://www.digi.com/xbee/
[4] http://www.ros.org/wiki/swistrack

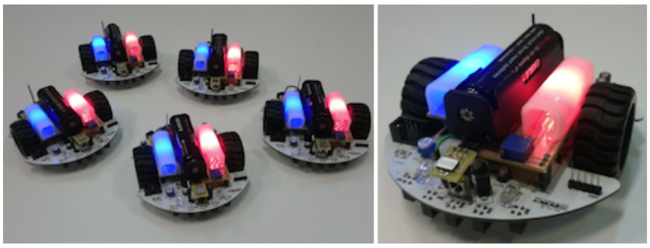


Figure 3.42: The developed LSE MiniQ robots containing gas sensors, XBee modules and LEDs.

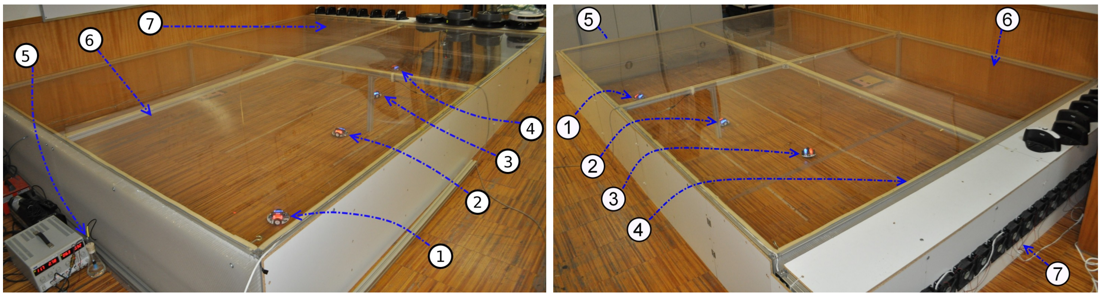


Figure 3.43: The realistic testbed environment viewing from two different angles. 1,2,3,4: robots, 5: odor source (Acetone release bubbler), 6: transparent Plexiglas ceiling, 7: ventilation system

maintain a specific distance. If one of the robots moves toward a direction, regardless of the cause of this movement, the other robots dynamically move to maintain the line formation. In this particular test we set D_3 and D_5, both, to 0.4m and the odor source was located at position $(0,0.8)m$. If one robot is located close to a boundary (or an obstacle) of the testing area, it tries to avoid it. The robots trajectory and the odor map generated by the robots is presented in Fig. 3.45 and Fig. 3.46.

Similar to the presented simulations, for evaluating the optimization results, the experiments were done with manual values for D_3 and D_5 testing diagonal line formation and also hyperball formation while moving towards cross-wind. The wind speed was about $0.7 \pm 0.1m/s$ and we tested the diagonal line formation strategy with different number of robots (2, 3 and 4) in various sets of parameters (D_5 from 0.5 to 3.0 m and D_3 0.1m and 0.5m). In different tests, one, two, three and four robots were released in one corner of the testbed and each test was repeated seven times and the plume detections were counted. We considered the following two positions for the

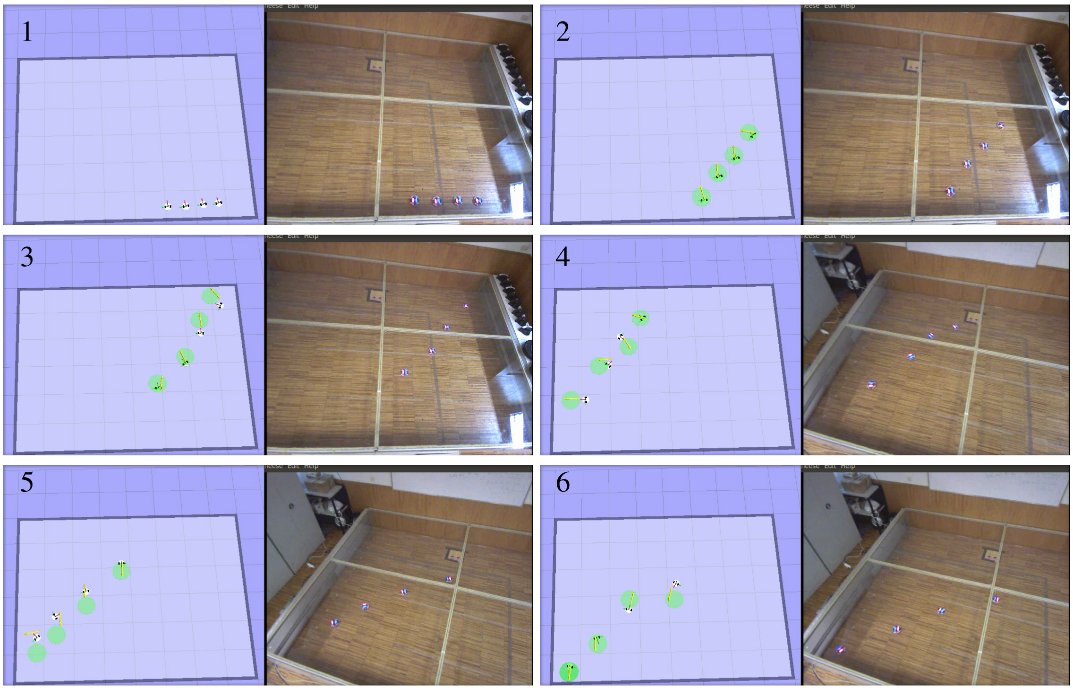


Figure 3.44: A series of snapshots during an experiment. The left frames show *rviz* environments where the internal status of the robots and their goals are presented. The right frames show the real LSE MiniQ robots in the testbed.

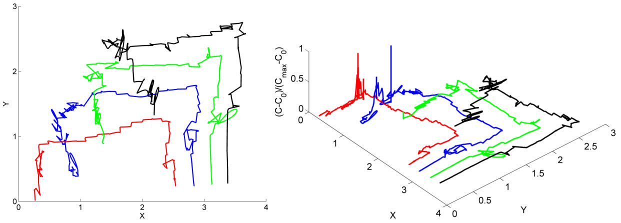


Figure 3.45: Left: the trajectory of the robots during the experiment shown in Fig. 3.44. Right: the Z-axis in this figure shows the chemical concentration sensed by the robots during their maneuvers.

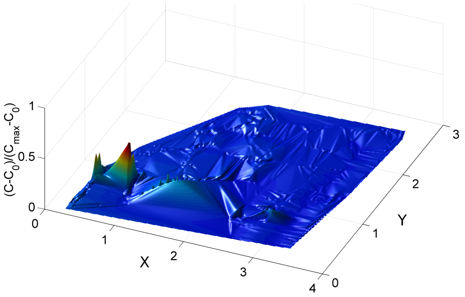


Figure 3.46: The chemical map of the environment generated by the robots after an experiment.

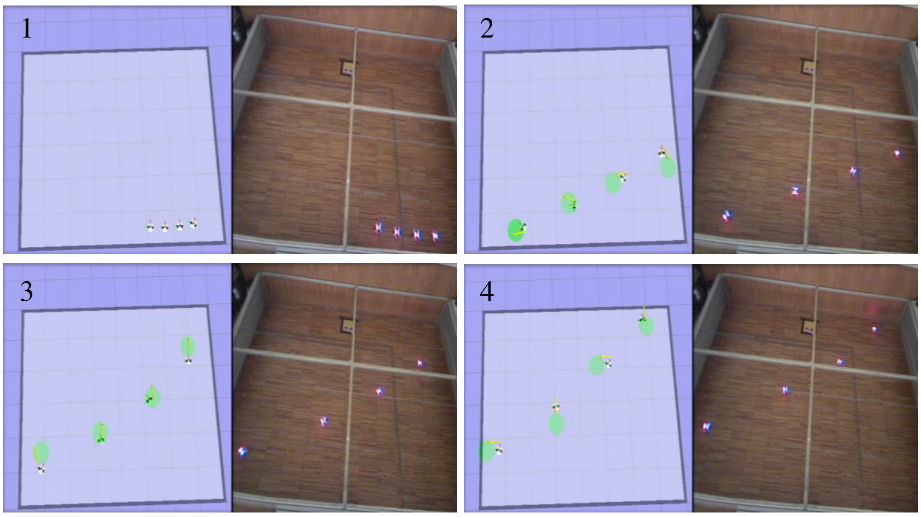


Figure 3.47: A series of snapshots during an experiment while $N = 4$, $D_3 = 0.5m$ and $D_5 = 1m$.

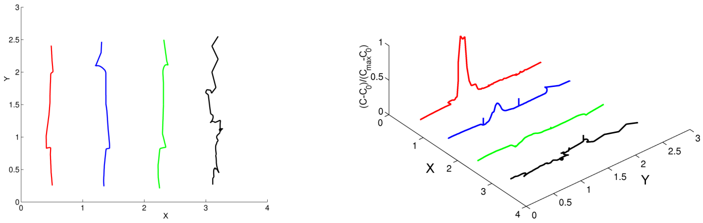


Figure 3.48: Left: the trajectory of the robots during an experiment while $N = 4$, $D_3 = 0.1m$ and $D_5 = 0.75m$. Right: the Z-axis in this figure shows the chemical concentration sensed by the robots during their maneuvers.

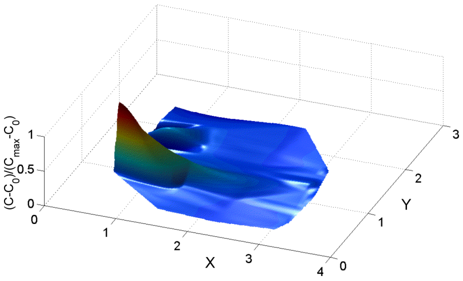


Figure 3.49: The chemical map of the environment generated by the robots after the experiment described in Fig. 3.48.

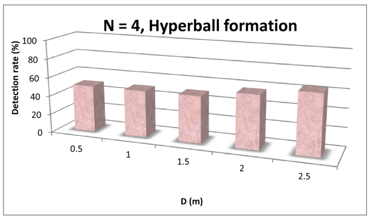


Figure 3.50: Detection rate in hyperball formation against different values for the distance between the neighboring robots (D).

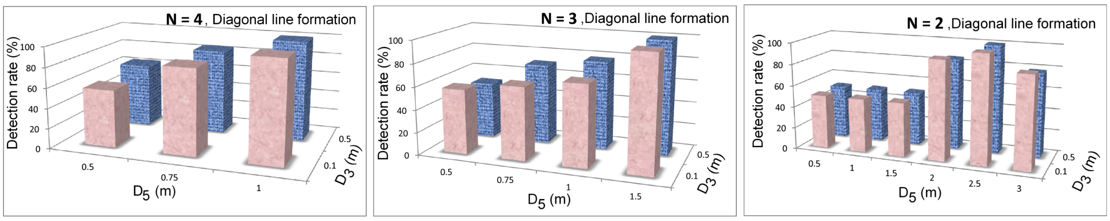


Figure 3.51: Detection rate of diagonal line formation against different values for D_3 and D_5, when (left) the number of robots is 4, (middle) the number of robots is 3, and (right) the number of robots is 2.

odor source release: (0,0.8) and (2,2). Each particular test was repeated two times (once having the source at (0,0.8) and another time at (2,2)). In each test if at least one of the robots detects the odor plume we consider a success in plume detection. Fig. 3.47 shows a series of snapshots in one of these experiments. Fig. 3.48 presents the trajectory of four robots in one experiment and Fig. 3.49 shows the chemical map of the environment generated by the data captured by the robots after an interpolation process.

Fig. 3.50 and 3.51 show the results of the realistic experiments, testing hyperball and diagonal line formations with different parameters . Comparing the results shown Fig. 3.50 to the three charts in Fig. 3.51, proves that hyperball formation provides less detections in comparison to diagonal line formation. The results in Fig. 3.51 (left) show that if there are four robots in this arena, when the X-distance between each pair of robots is one meter, they can detect the odor sources in 100 percent of the tests. It should be mentioned that we can not increase D_5 more than one meter (for four robots) since the length of the testbed is $4m$. Fig. 3.51 (middle) shows that if there are three robots in this environment they reach to one hundred percent detection when their X-distance is $1.5m$. The results in Fig. 3.51 (right) show that if there are two robots, they detect the odor patches when their X-distance is $2.5m$. Since two robots can provide (almost a) full coverage over this area, it is obvious that having more number of robots also will fully cover the environment and will not provide useful information regarding optimal distance between the robots. Therefore, considering the small size of the testbed, the best distance of two robots should be considered to be the best distance of the robots in this environment. The best X-component distance of two robots is 2.5m and the Y-component can be both 0.1 and 0.5m. During these tests, changing D_5 from 0.1 to 0.5m does not show a significant change in the results since the testbed is small.

Using equations (3.31) and (3.32), the optimal X-distance in this configuration is $2.1m$ and the optimal Y-distance is $0.15m$ that approximately agree with the results of the real experiments. The difference between the results of the real world experiments and the results of the optimizations is because the airflow and the detection of the sensors in reality are not as constant as what is assumed in the analytical optimizations. Nevertheless, the realistic experiments show very close results to the optimizations.

3.9 Conclusion and discussion

Optimal spatial formation of swarm robots to maximize the probability of detection of odor plumes was studied under a set of assumptions. We first considered no global movements for the swarm and found the best optimal configurations of the gas sensors in search for odor plumes. Afterwards, considering cross-wind movement for the swarm we found the optimal formation of swarm robots that maximizes their probability of finding an odor plume in an unknown environment.

This chapter showed that assuming no global movement for a swarm, their optimal topology is line configuration toward cross wind direction with equal distance between each pair of adjacent sensors. Regardless of number of sensors, the optimal distance between neighboring pairs depends on the wind speed.

Considering cross-wind movement for the swarm, the formation of the optimal solutions is diagonal line configuration, with equal distance between each pair of neighboring sensors. The distance between neighboring pairs in optimal configurations depends mainly on the wind speed and the environmental conditions, whereas the number of sensors and the swarm's crosswind movement distance do not show significant impact on optimal configurations.

Mathematical functions that can accurately estimate the optimal distances based on the wind speed was computed by nonlinear regression estimation. Moreover, swarm robotics wind-biased attractive/repulsive virtual forces were designed to emerge to the optimal configurations.

The method was tested and validated in simulations and in a reduced scale realistic environment under laminar and stable controlled airflow. The results verify the functionality of the swarming formation strategy and also validate the obtained optimization results. A few additional experiments were done in an indoor environment under unstable and variable airflow. The swarm successfully performed the diagonal line-up behavior in any condition, however the results of the experiments in the setup of our indoor environment are not comparable with the optimizations due to instability of the airflow.

Chapter 4

A perspective on odor plume tracking and source declaration with a swarm

"It is the same whether you take it that the Earth is in motion or the Sky,
In both the cases, it does not affect the Astronomical Science.
It is just for the Physicist to see if it is possible to refute it."
Abu Rayhan Biruni, 973-1048

"Tracking odor plumes" and "odor source localization" problems are not among the main objectives of this thesis. However these are the next problems that we need to address in future in order to have a complete olfactory swarm system for localizing the odor sources in an unknown environment. This chapter proposes a method to address these problems using swarm robotic approaches. Although the proposed method is implemented in simulations and preliminary results validate its functionality, still, excessive study is needed in order to analytically prove its functionality.

4.1 Problem statement

Considering three phases for "odor source localization", the problems of odor plume finding and odor source declaration using a swarm of robots are stated as following:

Consider a swarm of *N* robots that are able to communicate with each other over a distance

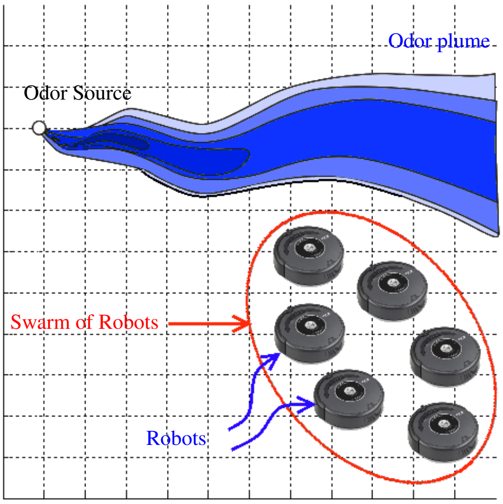


Figure 4.1: A swarm of robots navigating and searching for odor source.

Δ_d and are equipped with olfactory sensors for sensing the odor concentration $\bar{C}$ and airflow speed $\vec{U}$. There is no central controller for the system, so the robots act independently. The problems are:

1. How can a swarm of robots cooperatively track an odor plume?
2. How to maximize the probability of identifying a true odor source and minimizing the probability of false detections?

4.2 Related works on odor plume tracking

The use of robotic systems to track odor plumes has been broadly investigated in the literature. However, tracking plumes using multiple robots has received much less attention. In this problem, a large number of robots, equipped with sensors and communication tools, track the odor plumes in minimal time. This section presents a short survey on the previous works on this problem with more attention on the multi-robot approaches.

Three main families of algorithms that provide general solutions to odor plume tracking have been investigated. The simplest are derived from classical hill climbing or gradient ascent techniques [RMV91]. The second is a family of algorithms inspired by behaviors of biological systems, such as E. coli bacteria [DSR04] or other biological swarms [GP04b, LP02]. A final family employs probabilistic methods which we select Bayesian occupancy grids approach to mention [FJC$^+$07].

4.2.1 Hill climbing (gradient ascent) algorithms

The first attempts to use a mobile robot to track a plume were implemented with simple gradient ascent algorithms. Rozas et al. [RMV91] presented one the first attempts to use a mobile robot equipped with chemical sensors to follow a gas concentration gradient towards its source. In these approaches, due to intermittent structure of the plume, odor concentration is seen as a noisy function and it is assumed that the odor source is located at the maximum of this function. Most of these hill climbing algorithms show the possibility of being trapped in local maximums.

Some works have tried to implement hill climbing algorithm using multiple robots in order to track the plume mainly by considering the robots as nodes of a sensor network. Sandini et al. [SLV93] presented a gradient ascent cooperative multi-robots coverage approach and showed that it is faster than single robot approaches. Some attempts have been made to avoid stalling at a local maximum by incorporating a random walk [DSR04], swarm cooperation [CHRE04a], or probabilistic models which will be discussed later in this section.

Turbulent behavior of airflow, lack of smooth odor concentration gradient, patchiness of odor depression, meandering and time variant characteristics of odor plumes imply that concentration gradient climbing methods (chemotaxis [RBHSW03, GDC$^+$97]) alone cannot efficiently track odor plumes in an environment under turbulent flow.

4.2.2 Biologically inspired algorithms

Since the ability of tracking odor plumes is crucial to many living species, several robotic algorithms are inspired by various organisms in Nature. Most bio-inspired algorithms are based on the behaviors of moths [HMG02, FCM$^+$09, LRZ03] that perform three main actions [Rus99]:

- Up-wind surge: when exposed to odor patches, go towards upwind.
- Casting: after losing the plume, swing from side-to-side for a few seconds, in the hope to reacquire the plume
- Spiral: if still the plume is not found, perform circular motion patterns.

Based on the combinations of these three actions, researchers have developed various state-based algorithms. Hayes et al. [HMG02] derived a similar odor plume tracking strategy inspired by moths behaviors, called spiral surge, to implement a distributed and cooperative search, based on odor and wind measurements in an environment with background flow. Ferri et al. [FCM+09] successfully applied a modified spiral algorithm in the absence of airflow by relying on communication between agents rather than wind measurements to track the plume.

Another group of algorithms are inspired by the behaviors of bacteria. Russell et al. [Rus03] compared three chemo-tactic algorithms based on E. coli bacteria, the silkworm moth, and the dung beetle in environments of varying degrees of turbulence. This work found that success of each algorithm depends on the consistency of the plume: the E. coli algorithm is limited to chemical concentrations with smooth variations, the silkworm moth algorithm is preferred in tracking a sharply fluctuating plume, and the dung beetle algorithm is more successful when tracking a homogeneous plume.

We categorize the multi-robot bio-inspired approaches as following:

General swarm methodologies

Swarming algorithms have provided more robust solutions to plume tracking problem. Cui et al. [CHRE04b] have developed a swarm-based fuzzy logic algorithm to locate a single hazardous emission source in the presence of multiple local emission concentrations. The social foraging behavior of E. coli bacteria, also known as bacteria foraging optimization (BFO), have inspired a robust swarming algorithm in search for sources [LP02, TCK+12]. This approach is well suited for noisy source searches because the algorithm successfully targets the global optimum in spite of weaker local optima.

Kazadi [Kaz03] proposed a swarming approach for finding odor plumes in a windy environment by trying to extend the well-known spiral surge and up-wind plume tracking to a swarm of robots. In this method, each robot independently searches for odor plumes, if a robot found an odor clue it calls the other robots to come to the place and follow up-wind direction. Hayes et al. [HMG03] have also presented a work to extend the spiral surge to a swarm of robots. They ran single robot spiral surge algorithms on multiple robots individually and tried to optimize this algorithm's parameters by monitoring the group of robots using a centralized learning process.

Swarm biased random walk

Dhariwal et al. [DSR04] have described a method inspired by the chemotaxis of E. coli bacteria. In the BRW algorithm, all agents act independently without communication or synchronization and execute two actions: a run and a tumble. A run is a movement in a straight line, and a tumble is a random reorientation in a new direction. In the presence of an emission source, the concentration gradient affects the length of the robot's run. If a run is in the direction of a positive gradient, the length of the run is extended by a bias. By extending the length of runs in the direction of the positive gradient, the robot moves toward the gradient source. While this approach is prevalent in the field, Virk et al. [VK98], Marques et al. [MNdA02a] and Russell et al. [Rus03] are the first to simulate the biased random walk algorithm on a large robotic network to locate multiple generic gradient sources.

Biasing Expansion Swarm Approach (BESA)

Cui et al. [CHRE04a] have presented an algorithm to trace an unspecified number of chemical sources in a large, unknown area, called Biasing Expansion Swarm Approach. The agents obey three basic swarm control rules of separation, cohesion, and alignment with a bias towards areas of higher emission concentration. The key point of BESA algorithm is the formation of a global ad-hoc network in which agents are limited to local communication. Through this ad-hoc network, each agent can communicate with the other agents and sense their concentrations and use them in its local BESA algorithm.

Each agent uses a occupancy grid to plan its next target cell within the framework of the swarm rules. To maintain the rule of separation, an agent avoids entering a cell already occupied by another agent. To maintain swarm cohesion, the agent uses a gradual expansion algorithm. This gradual expansion algorithm maintains the cohesion property of the swarm as well as the ad-hoc network by requiring that an agent stays within local communication range of at least one member of the swarm.

In order to steer the swarm toward the emission sources, the BESA algorithm assigns a parameter to each agent's expansion cells. The agent will choose the expansion cell with the highest biasing parameter as its target cell [CHRE04a]. The effect of this biasing action is that agents target empty cells that are nearest to the cells where high concentrations are measured. As a result, the swarm moves in the direction of higher concentrations. However, since the odor dispersion is time-variant and patchy, these approaches usually do not show high performance in olfactory search.

Particle Swarm Optimization (PSO)

PSO is another heuristic approach that is mainly used for finding maximum or minimum values of a function [KES01]. PSO is inspired by the behavior of schools of fish or flocks of birds to find food. The coordination of movements of the individuals in the swarm is the central aspect that inspires PSO. A PSO algorithm maintains a population of particles (the swarm), where each particle represents a location in a multidimensional search space (also called problem space). The particles start at random locations and search for the minimum (or maximum) of a given objective function by moving through the search space. The movements of a particle depend only on its velocity and the locations where good solutions have already been found by the particle itself or other neighboring particles in the swarm. Note that, unlike many deterministic methods for continuous function optimization, PSO uses no gradient information [Tal09].

In a typical PSO algorithm each particle keeps track of the coordinates in the search space which are associated with the best solution it has found so far. The corresponding value of the objective function (fitness value) is also stored. Another value that is tracked by each particle is the "best" value obtained so far by any particle in its neighborhood. When a particle takes the whole population as its neighbors, the best value is the global best. At each iteration of the PSO algorithm the velocity of each particle is changed toward the personal and global best (or neighborhood best) locations. Moreover, several random components are also incorporated into the velocity update.

Marques et al. adapted the PSO algorithm to the multi-robot odor searching problem for finding odor sources in large spaces [MNdA06]. Li et al. proposed a probability particle swarm optimization (P-PSO) algorithm for multi-robot based odor source localization in ventilated indoor environments and evaluated it through simulations [LMB+08]. Akat and Gazi proposed a decentralized and asynchronous algorithm named DAPSO (decentralized and asynchronous PSO) that allows for dynamic neighborhood topology and time delays in information exchange [AG08]. Recently, Turduev et al, experimentally showed that the performance of DAPSO is higher than ant colony optimization (ACO) and bacterial foraging optimization (BFO) [TCK+12] in finding odor sources.

Ant Colony Optimization (ACO)

ACO is another heuristic method for solving optimization problems. It is inspired by the way real ants find shortest paths from their nest to food sources [DBS06]. An essential aspect thereby is the indirect communication of the ants via pheromone, i.e., a chemical substance which is released into the environment and that influences the behavior or development of other individuals of the same species. Ants mark their paths to the food sources by laying a pheromone trail along their way. The pheromone traces can be smelled by other ants and lead them to the food source. The pheromone guides following ants of the next iteration so that they search near the paths to good paths (solutions). The pheromone from older iterations does not influence the following iterations for too long, because it evaporates. Thus, an ACO algorithm is an iterative process where pheromone information is transferred from one iteration to the next one. The process continues until some stopping criterion is met: e.g., a certain number of iterations has been done or a solution of a given quality has been found [BP95].

Several studies have utilized ACO algorithm for odor source localization in simulation environments [MLLZ06, ZLC09]. Turduev et al, implemented ACO algorithm in a realistic environments using small mobile robots and compared its performance with DAPSO and BFO in chemical gas concentration mapping [TCK$^+$12].

Attractant/Repellent swarm model analysis

Gazi and Passino [GP04b] presented a swarm model for individuals moving in an attractant/repellent profile which includes stability and convergence analysis of the swarm. The model controls agents individually but it is centralized in that each agent needs to know the relative positions of all other agents. In that work, simulations were performed primarily to verify analysis. Attractant/Repellent approaches are usually used in the swarming search methods to design spatial formations and movements.

4.2.3 Probabilistic algorithms

Another family of algorithms address plume tracking through probabilistic methods. Farrell et al. [FPL03] have used hidden Markov methods (HMM) with measurements of chemical concentrations and fluid dynamics to predict the likelihood of odor detection that results in traveling towards odor source. Pang and Farrell [PF06] used similar measurements coupled with Bayesian inference to build a source likelihood map that identifies a narrow search area for the source. Olfati-Saber [OS07] has simulated distributed target tracking on a mobile sensor network by integrating a network of cooperative Kalman filters with a flocking-based mobility model that preserves network connectivity. Scerri et al. [SGO$^+$07] have proposed a distributed bayesian binary filter grid (BBFG) that allows teams of unmanned aerial vehicles (UAVs) to use received signal strength indicator (RSSI) sensors to locate multiple RF emitters over a large area. Each UAV maintains its own BBGF and shares some sensor readings with other UAVs. The BBGF output is converted into a map of information entropy, and all UAVs plan paths through regions of highest entropy to maximize the amount of information gained.

4.3 Related works on odor source declaration

The problem of accurately detect and declare an odor source is not deeply studied in the literature. In many studies, other modalities rather than olfactory, e.g., vision, are used to identify the source when an agent passes sufficiently near it.

Li et al. [LMS$^+$09] presented a three-step odor source declaration method which include robots convergence, odor concentration persistence judgment and odor mass throughput calculation. They showed experimental results in both simulation and indoor environments under airflow.

Lilienthal et al. [LUF$^+$04] evaluated two machine learning approaches, artificial neural network (ANN) and support vector machine (SVM), to propose a methodology for declaring whether an object is a gas source or not from a series of concentration measurements recorded while the robot performs a rotation maneuver in front of a possible source. They reported a maximal hit rate of approximately 87.5 % using a support vector machine.

A few probabilistic algorithms also offer solutions to this problem. The hidden Markov methods (HMM) proposed in [FPL03] estimate the location of the odor source probabilistically. The method presented by Pang and Farrell [PF06] also builds a source likelihood map that identifies a narrow search area for the location of the source.

4.4 A proposed method for swarm odor plume tracking

Despite most of other works in this area that propose centralized approaches and address the problem in a very simple conditions, this chapter presents a distributed method in which robots localize an odor source in an unknown environment where the odor distribution is time-variant and patchy, the flow is turbulent, and there are obstacles in the environment. The main contribution in this method is that the motion of swarm robots is not only based on the robot-to-robot virtual forces; the olfactory sensory data affects the motions of the robots and also the virtual forces that they apply to each other. In this method, when one robot finds the odor plume, all of the swarming robots aggregate together and track the odor plume based on their swarm formation and the stochastic concentration gradient and airflow direction. Finally, once some of the robots identified the odor source, using a consensus system, they cooperatively declare the source. The individual robots switch between these behaviors automatically without any central coordinator. The performance of the method is evaluated against various number of swarming robots, the emission rate of the odor source, the number of obstacles in the environment and the size of the testing environment. The emergent behavior of the swarm proves the functionality, robustness and scalability of the system in different conditions in simulations. Real world experiments with large number of robots are required to prove its functionality in real scenarios.

The problem here is: "How can a swarm of robots cooperatively track an odor plume efficiently?". Since the plume's shape and the environment are both unknown, there is no optimal solution for this problem. However, to address this problem in a turbulent odor plume, we propose a method where the swarm robots establish a cohesive spatial sensor network. Since the robots are spatially distributed in the environment, they can cooperatively measure the odor concentration and airflow velocity of different locations and track the plume more efficiently.

Based on the nature of odor distribution model [BS02] and formula (3.2), the following fact should be considered: when a robot observes an odor patch, it is obvious that the best strategy is to make a step in the direction from which the patch has arrived. Since the wind carries the odor patches, if (x_0, y_0) is the position of the observed odor patch, with a very high probability the source can only be located in the interior of a cone that can be estimated with: $y - y_0 = \pm(x - x_0), y < y_0$ (red lines in Fig. 4.2 and equation (3.2), re-noting that Y-axis direction is assumed to be the air-flow direction). In other words, if a patch is observed at position (x_0, y_0), one should visit the places close to (x_0, y_0) and toward the upwind direction. Each odor patch observation reduces the uncertainty about the odor source position.

We propose swarm formation and navigation methods to cooperatively track the plume toward its source. The pseudo-code of the main algorithm that is run on each individual robot is presented in Algorithm 10. In this method, when a robot finds the odor plume (i.e. detects some odor patches in a place), the other robots get close to that robot in order to increase the probability of detecting the odor patches of that area. This sensor network navigates toward upwind direction and visits the places around the detected odor patches.

On the other hand, considering equations (3.6) and (3.3), when robots get close to each other, the coverage area of the swarm gets expanded toward the opposite direction of the airflow. Fig. 4.3 compares the coverage area of three different configurations of seven sensors in an environment. The right figure that is a mesh configuration shows the longest sensor coverage toward opposite wind direction. Maintaining a mesh-like topology the swarm will detect the odor patches coming from very further possible locations.

The formation is obtained by implementing virtual attraction/repulsion forces between the robots considering minimum communications. The robots which sense more odor patches (higher odor concentration), attract other robots with stronger forces to cover the region. Each robot individually measures the odor concentration and the airflow speed and its distance to the neighboring robots and tries to travel toward up-wind direction and higher odor concentrations while maintaining a constant distance to the neighbors. As an emergent result, the swarm moves toward the consensus direction of all the swarm members and spatially sweeps the region of interest. To do so, each robot performs three tasks simultaneously:

4.4.1 Maintaining formation of the sensor network (swarm)

The first task requires the robots aggregate together to keep a specific distance margin between each other and try to virtually push and pull each other. If a robot goes toward a specific direction, the robots in behind, to maintain the sensor network's cohesion, go toward the same direction and the robots in front also go to that direction in order to keep their distance. Therefore, the swarm always moves toward the direction that more robots tend to go while keeping the formation of established sensor network. For instance if ten robots try to go to the left and five robots to the right, the whole swarm (maybe with lower speed) will move to the left because the resultant virtual forces that is applied to each robot from the other robots is in that direction.

To implement this behavior, the virtual forces applied to a robot i from another robot j is

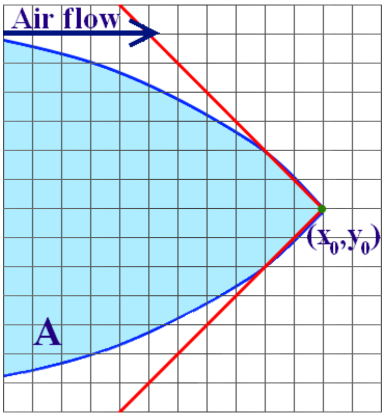


Figure 4.2: An odor patch is detected at (x_0, y_0). Based on equation (3.2), with a high probability this patch has come from a source in the blue highlighted area A.

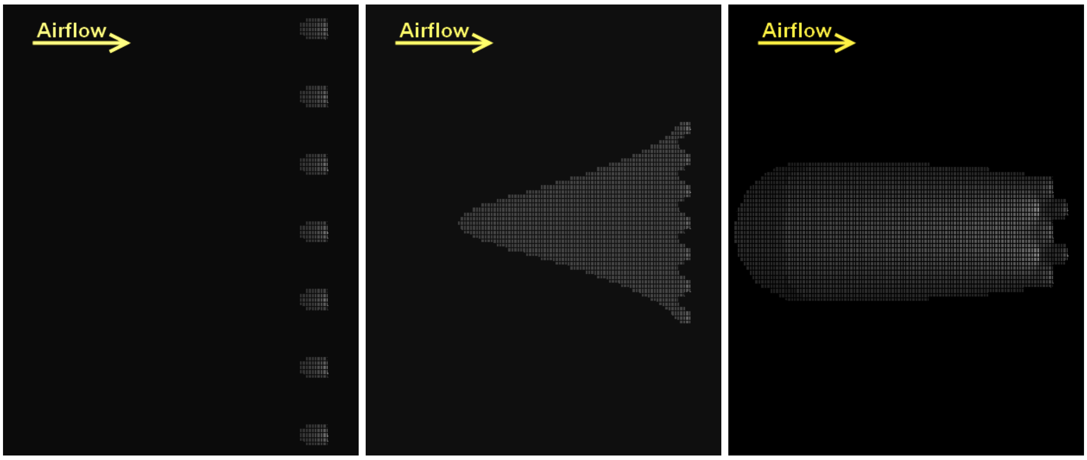


Figure 4.3: Sensor coverage area of seven sensors in three different configurations while $P_{threshold} = 0.2$. When the sensors get close to each other the covered area expands toward the opposite direction of the air flow.

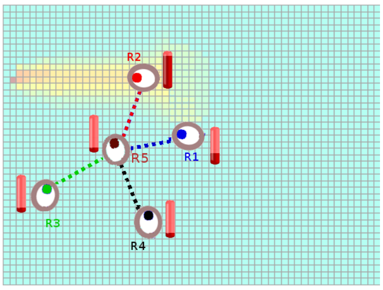


Figure 4.4: Plume tracking behavior; The vertical gauges present the odor concentration that the robots sense at the moment. R1 and R2 which sense high odor concentrations tend to track the plume by climbing the gradient and navigating toward up-wind (holding equation 4.4) while the other robots maintain the cohesion of the swarm formation (holding equation 4.1, 4.2, 4.3). The robots which sense higher odor concentration, (R1 and R2), apply bigger forces to the other robots.

Algorithm 10: Olfactory swarm algorithm for each individual robot

```
Behavior = Plume_finding
Mission_accomplished = False
while !Mission_accomplished do
    switch Behavior do
        case Plume_finding
            $c_i$ = Measure odor concentration()
            $U_i$ = Measure air-flow velocity()
            $\{P_i\}$ = Measure distances to the neighbors()
            $\{d_i\ ,\ Vec_i\}$ = Measure distances and direction to the obstacles()
            Navigate while holding formula(3.37 or 3.37)
            if ($c_i$ > odor_threshold) || (A robot indicates finding odor plume) then
                Broadcast(plume detected)
                Behavior = Plume_tracking
        case Plume_tracking
            $c_i$ = Measure odor concentration()
            $U_i$ = Measure air-flow velocity()
            Broadcast to neighbors($c_i, U_i$)
            Maintain formation of sensor network by formula(4.3)
            Navigating toward odor source by formula(4.4)
            Avoid obstacles by holding formula(3.15);
            if Odor_Source_Identified() then
                Behavior = Source_identification
            if No robot indicates finding odor plume then
                Behavior = Plume_finding
        case Source_identification
            if !Odor_Source_Identified() then
                Behavior = Plume_tracking
            else
                Broadcast to neighbors("source is identified")
    if (more than M robots are sending "source is identified") then
        Broadcast to neighbors("source is identified")
        Mission_accomplished = True
End of algorithm
// odor_threshold = threshold for odor plumes active area.
// Odor_Source_Identified() is explained in section 4.5.
```

defined as:

$$\vec{F}^{ij}_{co}(t) = \begin{cases} \frac{\mu_4 q_j}{||\vec{p}_{ji}(t)||^2}(\frac{\vec{p}_{ji}(t)}{||\vec{p}_{ji}(t)||}), & ||\vec{p}_{ji}(t)|| < R_1 \\ 0, & R_1 \leq ||\vec{p}_{ji}(t)|| \leq R_2 \\ \frac{-\mu_5 q_j}{||\vec{p}_{ji}(t)||^2}(\frac{\vec{p}_{ji}(t)}{||\vec{p}_{ji}(t)||}), & ||\vec{p}_{ji}(t)|| > R_2 \end{cases} \tag{4.1}$$

while μ_4 and μ_5 are two positive coefficients and R_1 and R_2 define a distance margin between each two neighboring robots ($0 < R_1 < R_2 < \Delta_d$). If two neighboring robots are farther than the defined threshold R_2 the equation acts like an attraction force, however, if the neighboring robots are closer than R_1 the formula acts like a repulsion force. This equation maintains the swarm in a cohesive form.

q_j is defined as:

$$q_j = \beta_i \frac{c_j - C_0}{C_{max} - C_0} \tag{4.2}$$

where parameter c_j represents the current odor concentration measured by robot j, C_{max} is the maximum odor concentrations that the swarm robots have reported so far, and C_0 denotes the minimum value that the olfactory sensors report in clean-air conditions. β_i is a constant that is predefined by the system designer as the effectiveness of the robot i on the other robots.

In most previous studies in swarm formation (e.g., [GP04b]), force formula is the inverse function of the distance between the agents and other parameters such as q_j are ignored. However, we used q_j as a parameter of effectiveness of the robots on each other and defined a formula for that based on the problems' modality (i.e. olfactory) to increase the efficiency of the method. Each robot has a parameter q so that the force that the robots apply to each other will be dependent on that. A robots with bigger q applies bigger forces to its neighboring robots. Therefore, a robot which senses high odor concentrations applies high amplitude virtual forces to its neighbors, in contrast, a robot that does not sense high odor, does not apply significant forces to its neighbors.

By the above equations, the total cohesion force $\vec{F}^{i}_{co}(t)$ for a robot i is determined as:

$$\vec{F}^{i}_{co}(t) = \sum_{j=1;j\neq i}^{N} \vec{F}^{ij}_{co}(t) \tag{4.3}$$

4.4.2 Navigating toward odor source

While the robots maintain the cohesion of the swarm, they must navigate toward the odor source. A navigational force is defined for each robot to perform this behavior:

$$\vec{F}^{i}_{odor}(t) = \begin{cases} -K_1\vec{\Delta}_{Ui}(t) + K_2\vec{\Delta}_{Ci}(t) & , c_i > C_T \\ 0 & , c_i < C_T \end{cases} \tag{4.4}$$

K_1 and K_2 are two positive constant coefficients, $\vec{\Delta}_{Ui}(t)$ is the airflow direction, $\vec{\Delta}_{Ci}(t)$ is the direction of the gradient of odor concentration measured by robot i, and C_T is a defined threshold for odor concentration. The first term in the above equation applies a force towards up-wind direction to the robot, whereas the second term applies a force towards the higher odor concentrations.

Not all the robots in a swarm are always located in an area of active odor plume. At a given moment, some of them may sense high odor concentrations and air flow while others do not sense any odor clues. Equation (4.4) implies that if a robot senses high odor concentration and also high air-flow speed, it goes toward up-wind direction and higher odor concentration, however, if the robot does not sense air-flow, it only follows the odor gradient. On the other hand if a robot does not sense high odor concentrations, it follows the swarm to maintain the sensor network's cohesion.

4.4.3 Obstacle avoidance

The obstacle avoidance forces applied to the robots in this behavior are the same as forces defined in section 3.4.2.

Total Force:

The total force applied to a robot i in this behavior is given by:

$$F^{i}_{plume}(t) = F^{i}_{co} + F^{i}_{odor} + F^{i}_{obs} \tag{4.5}$$

Now, $\theta_i(t)$, $w_i(t)$ and $\vec{v}_i(t)$ are calculated by formula (3.18), (3.19) and (3.20) respectively,

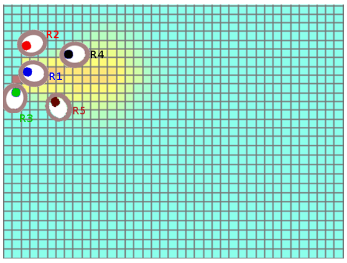


Figure 4.5: Source identification behavior; R1, R2 and R3 identify the odor source and broadcast a message to other robots nearby.

replacing $F^i_s(t)$ with $F^i_{plume}(t)$.

As an emergent result, the robots which detect high chemical concentrations tend toward up-wind direction and toward higher odor gradients (due to equation 4.4), while all the robots maintain their sensor network formation (due to equation (4.1). If some robots tend to go to one direction and others to another direction the whole swarm will actually move toward resultant summation of virtual forces of all the robots.

4.5 A discussion on swarm odor source declaration

The problem here is: “How to maximize the probability of identifying a true odor source and minimizing the probability of false detections?” This thesis does not go deep into the mechanisms of odor source identification, however, a simple approach is used for the robots to declare an odor source. An individual robot identifies an odor source if one of the following conditions holds:

1. While a robot is traveling up-wind direction, if the mean level of detected gas decreases suddenly it means that the robot has passed an odor source during its path.

2. If a robot is traveling in the down-wind direction and it starts to detect a very high concentration of odor, it means that there is an odor source in this location.

3. If there is no air-flow in the environment and the robot detects a very high concentration of odor, it means that there is an odor source nearby.

The probability of detecting an odor source by a robot can be modeled bases on the robot's sensitivity and the characteristics of the odor source and the environment. Due to the patchy nature of odor plumes, there might be some false positive results in detection of odor sources. Therefore, odor source identification should not be dependent on only one robot (also because of the fact that in any swarm system there might be some faulty robots). This thesis suggests a consensus system in order to detect the odor source by the swarm. Considering p as the probability of detection and q as the probability of false detection of an odor source by a robot, we should maximize the overall swarm's source detection probability and minimize the overall swarm's false detection probability. Since the robots identify the source independently, using the binomial probability distribution, the probability of false detection of odor source by at least M independent robots is given by:

$$P(Odor\ source\ detection\ by\ at\ least\ M\ out\ of\ N\ robots) = \sum_{i=M}^{N} (\frac{N!}{i!(N-i)!}) p^i (1-p)^{(N-i)} \quad (4.6)$$

$$P(False\ detection\ of\ source\ by\ K\ robots) = q^K \quad (4.7)$$

Where N is the total number of the robots in the region. The above equations shows that the higher value for M results in the lower probability of false detection. On the other hand, the higher M the lower probability of detection of the odor source by M robots. This shows a trade-of in selecting the value of M.

While the robots are navigating toward source, if a robot is in a situation that thinks the odor source is very close to its location it will broadcast a message to other neighbor robots to announce that. We call this message as "mission accomplished" or "odor source identified" message. Any other robot which is in the same situation will broadcast a similar message. On the other hand, if a robot has not yet identify an odor source but is receiving "odor source identified" message from more than a defined number of robots (M) it will stop working and it will also broadcast the same message. The emergent behavior is that if several robots start to send "odor source identified" message and their number is more than a defined threshold, after a short time all the robots will start sending the same message and will stop working (this is a kind of consensus system in the swarm). The required number of robots which should declare the odor source (M) depends on several parameters such as the certainty of each of them in detecting the odor source, the swarm size, and the width of the odor plume.

Suppose that there are 10 robots in the region of an odor source and the probability of source detection of each of them is 0.7 and their probability of false detection is 0.2. If one selects $M = 3$ in this method, the probability of at least 3 out of 10 robots identifying the source is 0.998 (equation (4.6)) and if exactly three robots identify the source the probability of the false detection of them is 0.008 (equation (4.7)).

4.6 Preliminary validation

We evaluate the method with different swarm sizes in environments with obstacles in which there are turbulences in the plume and the concentration of odor is patchy and does not change gradually. Similar to the previous chapter the model of the testing environments were given to ANSYS Fluent CFD software to simulate odor sources and provide 3-D odor concentration data and 2-D olfactory data was extracted. The olfactory data generated by ANSYS Fluent were exported to Matlab to be used in simulations. Robots were simulated in Matlab as independent entities with no shared variables similar to section 3.5.1.

Fig. 4.6 shows the virtual forces that the swarm robots generate in “plume tracking” behavior in various configurations. Each arrow in a place show the magnitude and the direction of virtual forces that would be applied to another robot if it was located in that place. By adding (or removing) robots to these scenarios the configuration of forces will change, however, these figures only show the virtual forces in the current setup of the figures before adding another robot that its q_i is 1. These forces are obtained by implementing the formula (4.1) and (4.2). As shown in these figures, each robot implies repulsive forces in its close surrounding area and attractive forces far from it. The robots which sense higher odor concentrations imply higher forces in their neighborhood (e.g., Fig. 4.6.B). As a result, if there was any other robot near these robots, the virtual forces would attract it toward these robots until a certain margin. This

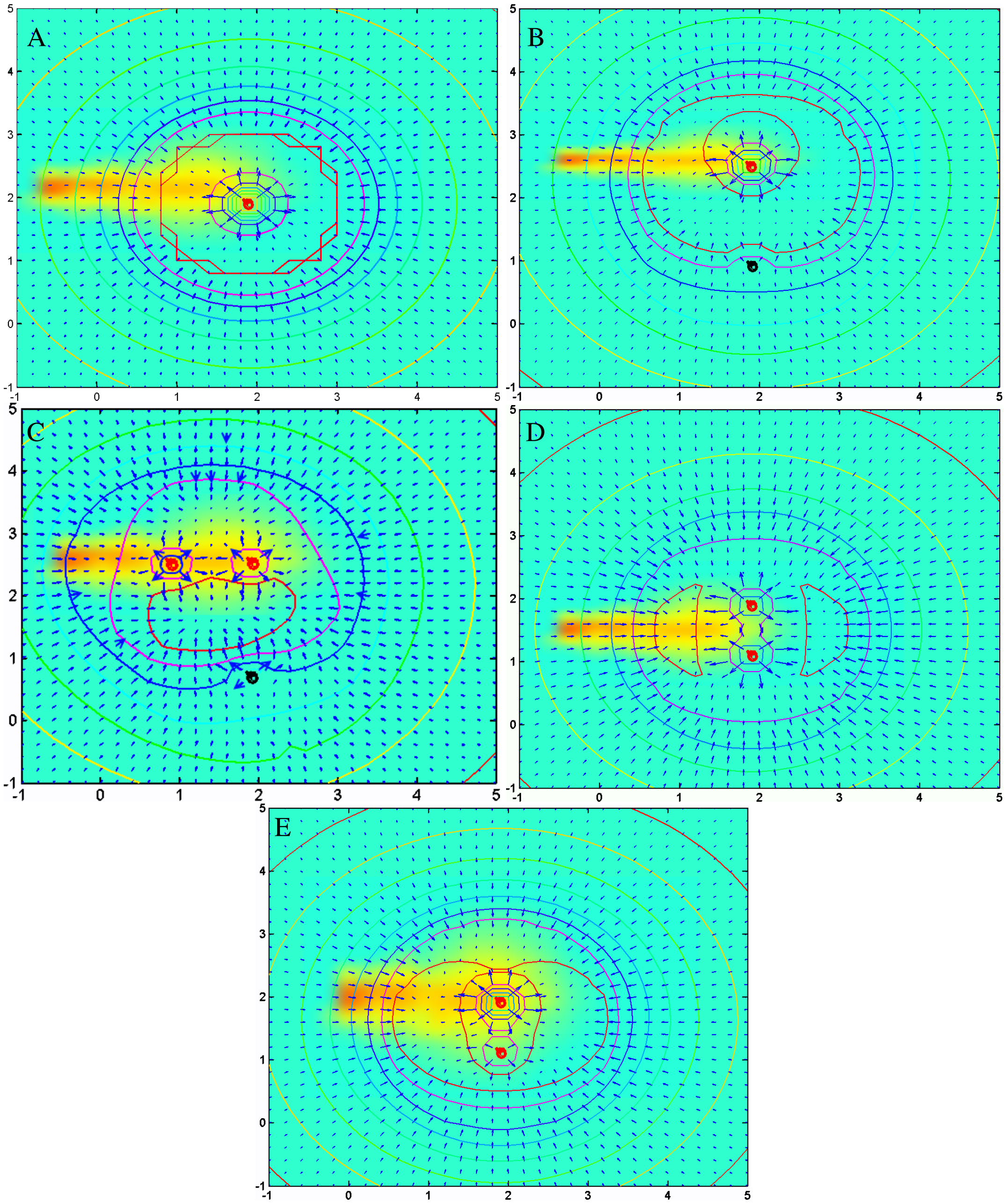


Figure 4.6: Virtual forces in "plume tracking" behavior. A: one robot sensing odor concentration greater than C_T (*odor_threshold*). B: only the upper robot senses odor concentrations greater than the threshold. C: only the two upper robots sense odor concentrations greater than the threshold. D: two robots sense equal odor concentrations greater than the threshold. E: two robots sense odor concentrations greater than the threshold while the upper robot senses more odor concentration than the lower robot.

margin can be adjusted by modifying the parameters R_1 and R_2. It must be mentioned here that the movements of the robots in this behavior is not only based on these virtual forces. Based on formula (4.4), each robot tends to go toward higher odor concentrations and up-wind direction while being affected by the swarm virtual forces. The direction and amplitude of navigation forces depend on the amount odor concentration and the air-flow direction in any specific location and is highly time-variant in these tests.

4.6.1 Validation

Fig. 4.7 demonstrates a series of snapshots during a simulation that shows the functionality of the method. The first frame of this figure shows that 10 robots are released randomly in one part of the environment. The next frames demonstrate the cross-wind line-up behavior, where the robots do not sense high odor concentrations and they get apart from each, form a diagonal line and move toward cross-wind direction. To better explain the functionality of the method, we have colored the robots based on their status in these simulations. If the mean odor concentration that a robot senses is less than a certain threshold, the color of that robot is black, but, if the robot senses more odor concentrations the robot turns to red. The blue robots are the ones which have detected and identified the odor source. In frame 5 of this figure one robot finds the odor plume, so it turns to red, therefore the other neighboring robots aggregate toward this robot in the next frames. From frame 5 to frame 12, the swarm robots track the plume toward higher concentrations while upholding sensor network's cohesion. The robots dynamically (and automatically) change their colors in this behavior and the emergent result is that the swarm is able to pass through the obstacles by changing its shape and travel toward the odor source. Frame 13 shows that a robot has detected the odor source (a very high concentration), and it turns its color to blue. In frame 14 some other robot detects the odor source and turns to blue. Since the swam size is not huge we defined $M = 3$ in the voting system explained in section 4.5. Therefore when three robots indicate that they have detected an odor source, another robot which detects these three robots will turn to blue to broadcast the message. The next frames show that more robots are indicating the odor source and this message is propagating in the swarm. In this experiment, the coefficient parameters of the method were set as following: $\Delta_d = 1m$, i.e.

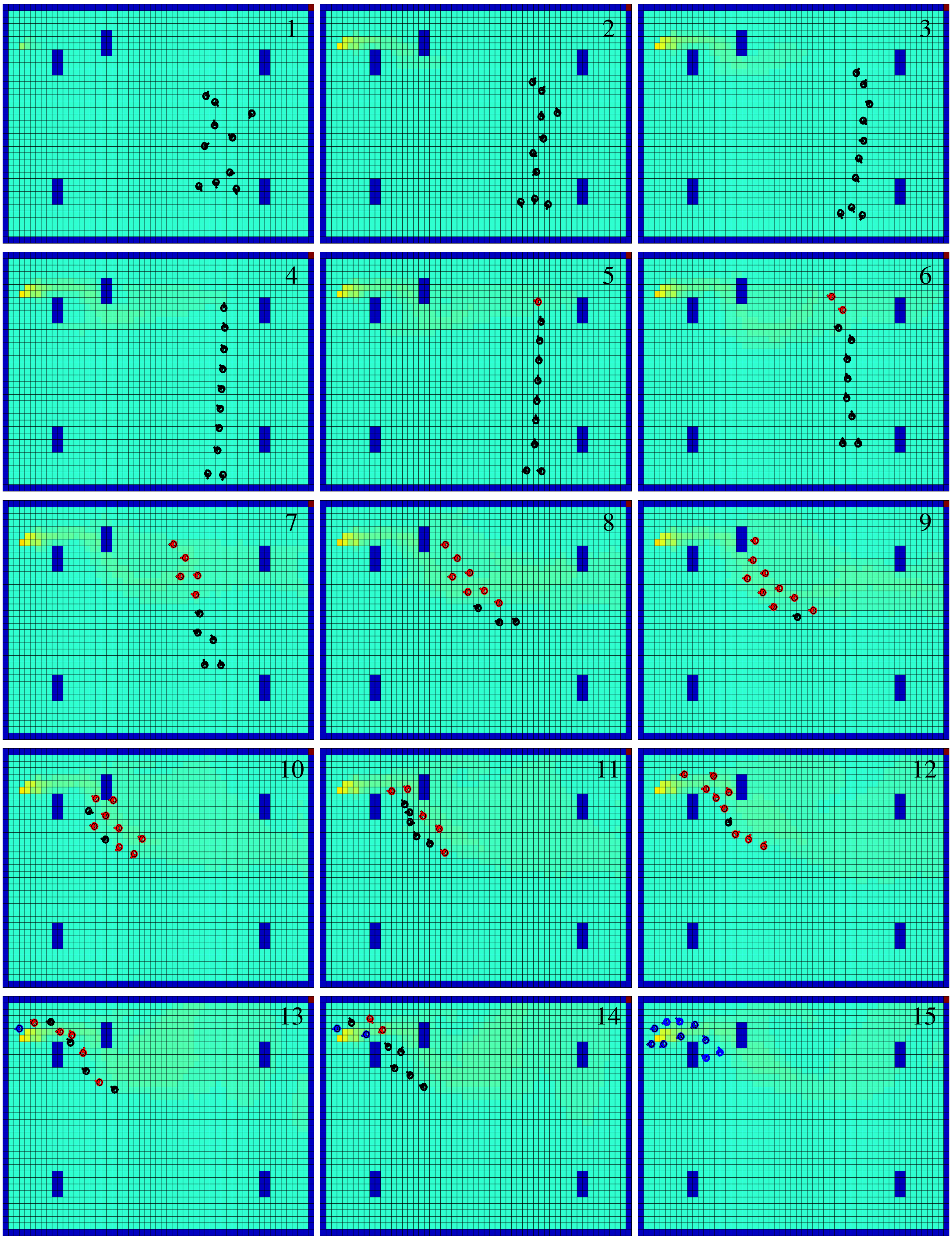


Figure 4.7: Swarm of 10 robots searching a $4 \times 6m^2$ environment for an odor source. The emission rate of odor source is 0.01 g/s. The red color means that the robot senses high odor concentration.

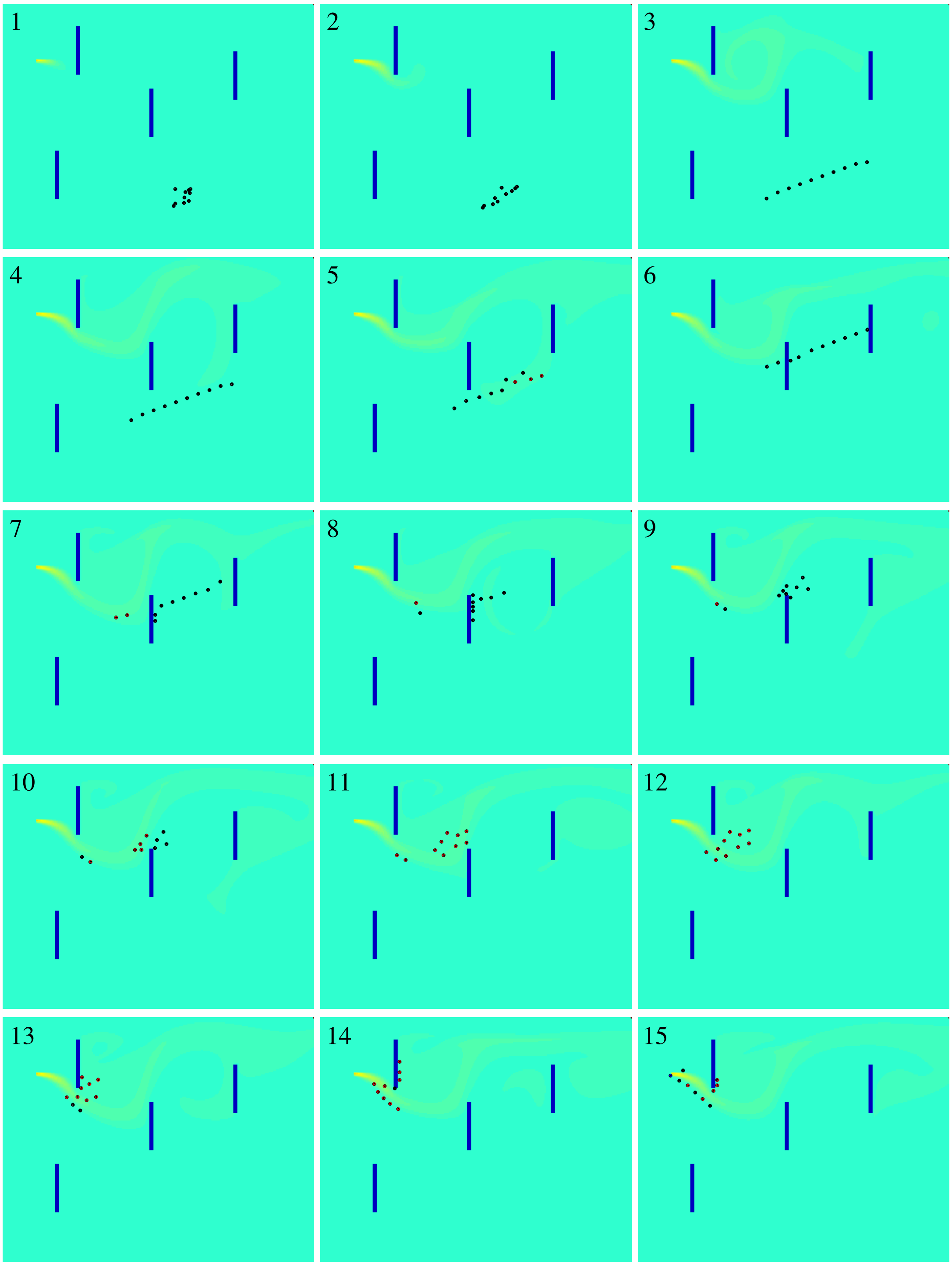


Figure 4.8: Swarm robots searching for an odor source in a $30 \times 40m^2$ environment. The figures only show a $10 \times 15m^2$ part of this environment.

the range of communication between the robots is considered to be one meters, $D_1 = 0.25m$ and $D_2 = 0.37m$ to keep the distance between the neighboring robots (during plume-finding behavior) between 25 to 37 cm, $\beta_i = 1$ for all the simulated robots since they are all equivalent, $R_1 = 0.1m$ and $R_2 = 0.5m$ for keeping the cohesion of the sensor network (during plume tracking behavior) and keep the neighboring robots in range of 10 to 50 centimeters apart, $K_1 = 0.5$ and $K_2 = 0.5$ to equally correlate the odor gradient, the wind and the forces applied to the robots, $C_T = 0.1$ as odor threshold and the other of the parameters were set equal to the parameters in section 3.5.1.

Fig. 4.10 shows another experiment that demonstrates the movements of 10 swarming robots in a large environment. In this experiment, the diagonal line-up behavior is used for finding the plume. In this experiment, the coefficient parameters of the method were set to the same values of previous simulations except following parameters: $\Delta_d = 2m$, $R_1 = 0.7m$, $R_2 = 1.5m$, $D_3 = 0.50m$ and $D_5 = 0.15m$. As shown in Fig. 4.10, when the plume meanders, the shape of the swarm automatically changes and the robots track the plume toward its source.

4.6.2 Evaluation

The method was evaluated against the following parameters:

- the number of swarming robots,
- the emission rate of the odor source, and
- the number of obstacles.

The first parameter is to test the scalability of the method in different conditions. The second parameter is to evaluate the sensitivity of method in different amount of odor in the environment. The last is to measure the performance of the method against airflow turbulences caused by obstacles.

Three environments with 0, 5 and 10 obstacles have been tested with 5, 15, 20 and 30 robots, while the emission rate of the odor source in the left side of the testbed was set to 0.01, 0.02 and 0.05 g/s. The wind speed was set to 0.5 m/s in these simulations. Fig. 4.9 shows the

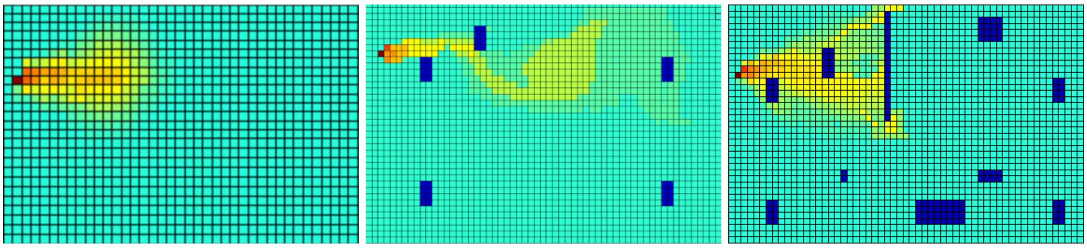


Figure 4.9: Snapshots of odor plumes in three simulated environments. left: obstacle-free environment, middle: environment with 5 obstacles, right: environment with 10 obstacles. The dimension of these environments is $4 \times 6m^2$.

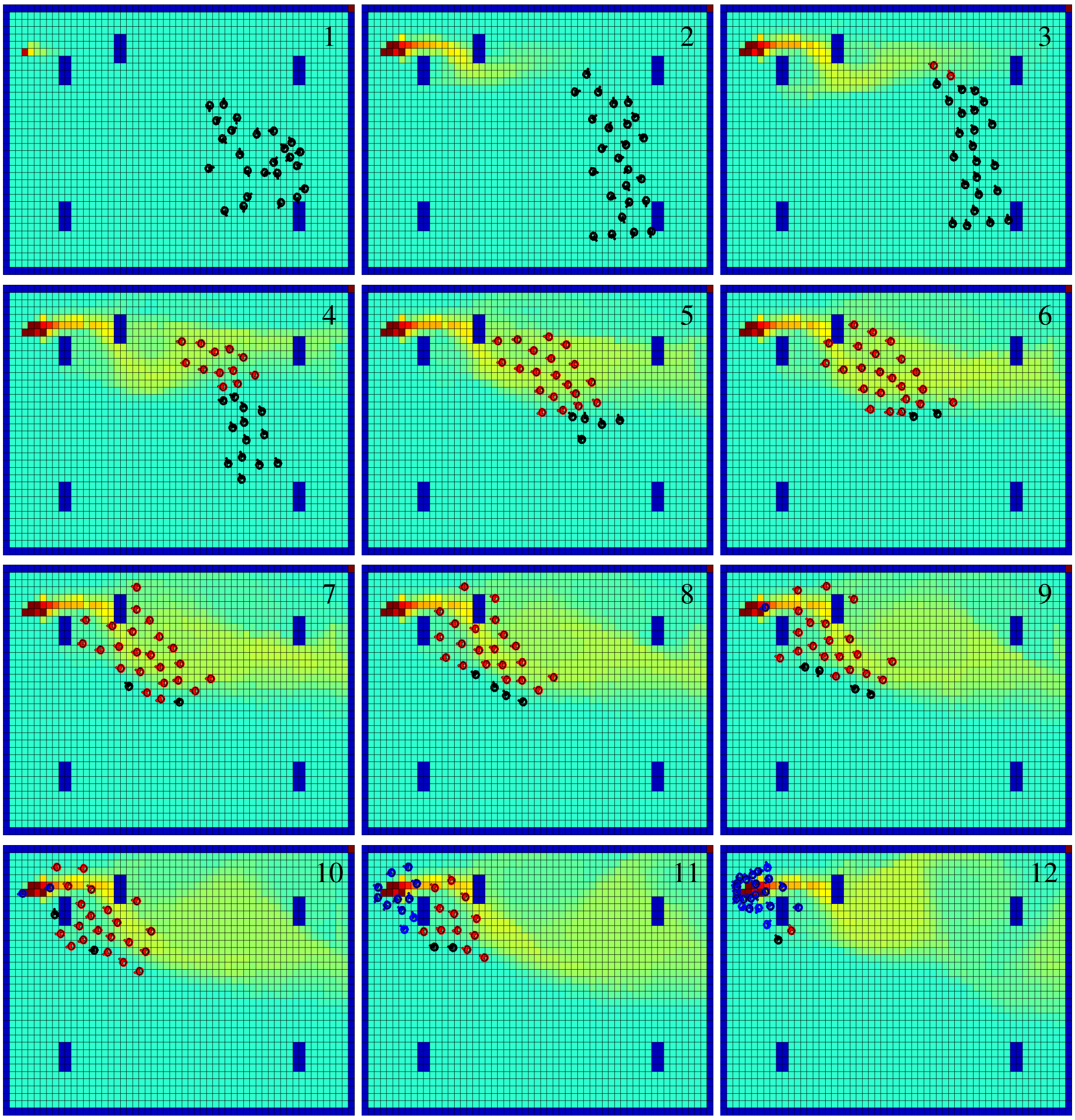


Figure 4.10: A swarm of 25 robots searching for an odor source in a $4 \times 6m^2$ environment. The emission rate of odor source is 0.05 g/s.

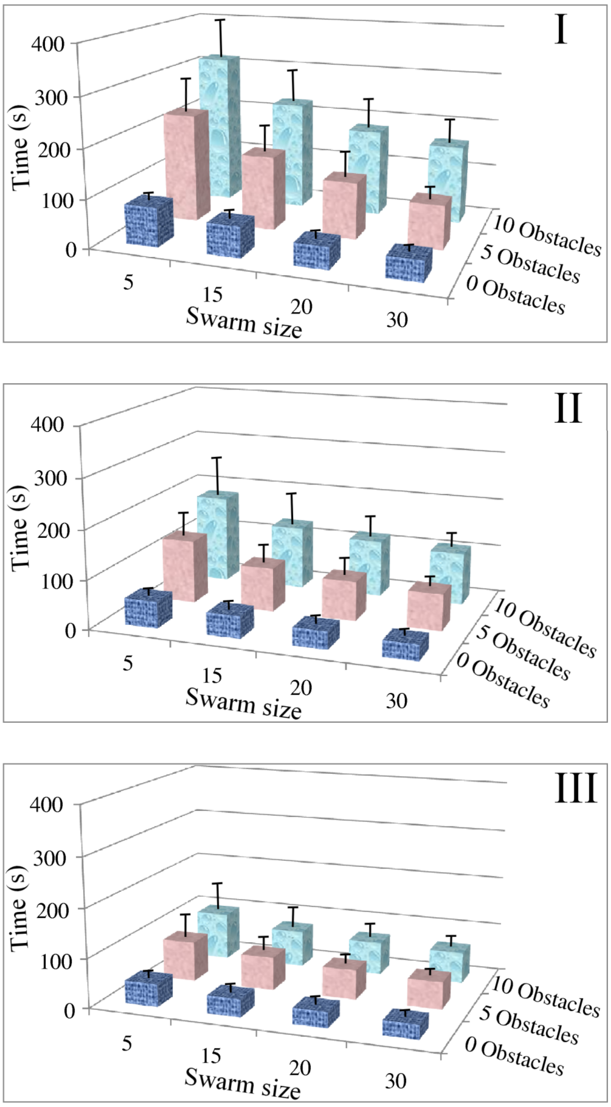


Figure 4.11: Search time in different environment with different number of robots while the emission rate of odor source was I: $0.01g/s$, II: $0.02g/s$, and III: $0.05g/s$

simulated odor plumes in these environments. Each experiment was repeated for ten times. In each experiment the other parameters were kept constant and the search time is measured. Since the goal of this section is to evaluate only the "plume tracking" method, we ran cross-wind line-up behavior for plume finding and measured the duration of plume tracking from the time that a robot finds the plume.

Fig. 4.11 shows the average search times in the mentioned conditions. The charts show that a bigger swarm finds the odor source faster than a small swarm specially when there are more obstacles and the emission rate of the odor source is low. The swarm was able to deal with the obstacles, turbulence and local maximums.

4.7 Conclusion and discussion

This chapter presented a proposed approach for a swarm of robots to track odor plumes and localize the source in near realistic conditions where the environments is under flow turbulent and the odor distribution is time-variant and patchy. The method was designed based on the swarm robotics principles of simplicity of individuals, distributed control and minimum communications. To track the plume, the swarm robots establish a dynamic sensor network by maintaining a cohesive formation between themselves to overcome the problems of local maximums, patchiness and turbulences in the odor plumes. The odor concentration and air-flow speed were considered in the swarm formation virtual forces' equations to more efficiently preform the tasks. The swarm identifies the odor source proposing a consensus approach that decreases the probability of false detection of the source. The plume tracking method has been simulated and the results prove functionality, robustness and scalability of the proposed approach in different conditions. The swarm shows acceptable performance for reduced swarm sizes and provides increasing performance with increasing swarm sizes specially when there are more obstacles in the environment and the emission rate of the odor source is low.

An issue that needs to be discussed is the emergent behavior of the swarm in dealing with very intermittent plumes where the robots lose the plume frequently during the plume tracking behavior. While a group of robots are performing plume-tracking behavior, they cooperatively try to track the plume by going up-wind direction and toward higher gradients while maintaining sensor network's cohesion. If suddenly all of the robots lose the plume, they all change their state to plume finding behavior (lines 23 and 24 of algorithm 10). In this case the robots start to perform plume finding behavior, until one of them finds the plume again and then all of them aggregate to that place. The main point here is that the robots are able to automatically and individually change their behavior without having a central station when the environmental condition changes.

Aditional study is still needed in order to analytically prove the functionality of the proposed method of this chapter. A formulation similar to what we presented for odor plume finding is required to analytically find the best strategy of swarm robots in these two problems. Moreover, the proposed method needs to be experimentally validated in real world scenarios using a swarm of robots.

Chapter 5

Concluding remarks and future works

"My heart was never deprived of knowledge
few secrets remain that I have not learned,
For seventy-two years I have pondered day and night,
now I know this: Nothing is really known."
Omar Khayyám, 1048-1131

5.1 Conclusions and remarks

This thesis provided a solution for the problem of "olfactory search and exploration in structured unknown environments" with a multi-robot system and another solution for "odor plume finding in unstructured unknown environments" using a swarm robotic system. The proposed methods were evaluated in simulations and in small-scaled realistic environments and the results were discussed. Moreover, the thesis proposed a swarm robotic approach for "odor plume tracking" and "odor source declaration" problems and provided preliminary results of its simulations.

5.1.1 Multi-robot olfactory search in structured environments

A cooperative distributed approach for searching odor sources in unknown structured environments with multiple mobile robots was proposed in chapter 2. The method is a decentralized frontier based exploration algorithm enhanced by a cost/utility evaluation function that considers the odor concentration and airflow at each frontier. Therefore, frontiers with higher probability of containing an odor source will be searched and explored first. While searching and exploring the environment, the robots independently generate on-line local topological maps and by sharing them with each other they construct a global map. The method also improves path planning of the robots for exploration process by presenting a priority policy. Since there is no global positioning system and each robot has its own coordinate reference system for its localization, this work used topological graph matching techniques for map merging. Two solutions were presented for the problem of topological map merging; sequential solution and parallel solution. The parallel solution is run on a *robotic cluster* platform. The results show a significant speed-up in merging large maps using the proposed method. The proposed method was tested in both simulation and real world environments with different number of robots and different scenarios. The search time, exploration time, complexity of the environment and number of double-visited map nodes were investigated in the tests. The results of real world experiments show the effect of gas cues on the behavior of robots and it shows that using the proposed algorithm, robots firstly explore the areas with higher probability of existence of odor sources. Simulation results show that having more robots is more advantageous in a complex maze than in a simple maze.

5.1.2 Swarm robotics olfactory search

Optimal spatial formation of swarm robots to maximize the probability of detection of odor plumes was studied in chapter 3. First no global movements for the swarm was considered and the best optimal configurations of the gas sensors in search for odor plumes were found. Afterwards, considering cross-wind movement for the swarm we found the optimal formation of swarm robots that maximizes their probability of finding an odor plume in an unknown environment. Assuming no global movement for a swarm, their optimal topology is line configuration toward cross wind direction with equal distance between each pair of adjacent sensors. Regardless of number of sensors, the optimal distance between neighboring pairs depends on the wind speed. Considering cross-wind movement for the swarm, the formation of the optimal solutions is diagonal line configuration, with equal distance between each pair of neighboring sensors. The distance between neighboring pairs in optimal configurations depends mainly on the wind speed and the environmental conditions, whereas the number of sensors and the swarm's crosswind movement distance do not show significant impact on optimal configurations. Mathematical functions that can accurately estimate the optimal distances based on the wind speed was computed by nonlinear regression estimation. Moreover, swarm robotics wind-biased attractive/repulsive virtual forces were designed to emerge to the optimal configurations. The method was tested and validated in simulations and in a reduced scale realistic environment. The results verify the functionality of the swarming formation strategy and also validate the obtained optimization results.

5.1.3 Swarm odor plume tracking and odor source declaration

An approach for a swarm of robots to track odor plumes and localize the source in near realistic conditions was proposed in chapter 4. To track the plume, the swarm robots establish a dynamic sensor network by maintaining a cohesive formation between themselves to overcome the problems of local maximums, patchiness and turbulences in the odor plumes. The odor concentration and air-flow speed were considered in the swarm formation virtual forces' equations to more efficiently preform the tasks. The swarm identifies the odor source proposing a consensus approach that decreases the probability of false detection of the source. The plume tracking method has been simulated and the results prove functionality, robustness and scalability of the proposed approach in different conditions. The swarm shows acceptable performance for reduced swarm sizes and provides increasing performance with increasing swarm sizes specially when there are more obstacles in the environment and the emission rate of the odor source is low. Additional study is still needed in order to analytically prove the functionality of the proposed method of this chapter.

5.2 Future works

Although the two objectives of this thesis are fulfilled in real world scenarios, yet the work is not finished and many areas can still be explored. This section identifies future works and categorizes possible developments which can be considered to continue this study.

5.2.1 Multi-robot systems

Regarding multi-robot olfactory search in structured environments the following future works can be mentioned:

- **Improvement of the feature extraction technique:** In order to apply the presented method in real world complex environments, the feature extraction/detection technique should be improved by better pattern recognition approaches. Since this issue was not the main objective of this thesis, a simple pattern recognition approach was used that can distinguish five different environmental features from each other. However, most real world environments contain more complex structures, so more advanced methodologies are needed to reliably distinguish the environmental features. Fusing the sensory data of multiple robots which observe an environmental feature from different points of view might improve feature extraction. We call this ***cooperative environmental feature extraction***. The robots should cooperate with each other to acquire useful information from the surrounding environment from different points of view and then fuse and process this data in order to accurately extract its features. Since sensor fusion approaches are usually heavy processing tasks, the idea of robotic clusters can be useful to execute cooperative environmental feature extraction in real time.

- **Cooperative communication:** The proposed method in this thesis does not necessarily require perfect communications, however communication failures decrease its efficiency. In real world structured environments, there might exist zones with no network signal. Using an ad-hoc architecture (e.g., OLSR[1], BATMAN[2], etc.) that the robots connect to each other and establish a network wirelessly and automatically without using a router, might be beneficial in these scenarios. In this case, the relative spatial topology of the robots is an important parameter in the performance and functionality of the system that should be studied.

- **Robotic platforms for real world environments:** The Roomba and LSE-MiniQ robots

[1] http://www.olsr.org/

[2] http://www.open-mesh.org

used in the experiments of this thesis are not able to navigate in rough trains. They cannot even overcome usual obstacles in human made facilities like taking stairs or using elevators. Robots which are able to navigate in these situations are needed to be employed in real world scenarios.

5.2.2 Odor plume tracking and odor source declaration

- **Mathematically analyzing odor plume tracking and odor source declaration:** Although the proposed method for swarm robotic odor plume tracking and source declaration was implemented in simulations and preliminary results validated its functionality, still additional studies are needed in order to analytically prove its functionality. A formulation similar to what was presented for odor plume finding is required to analytically find the best strategy of swarm robots in these two problems.

- **Experiments:** The proposed method for odor plume tracking and source declaration needs to be experimentally validated in real world scenarios using a swarm of robots.

5.2.3 Larger number of robots

Due to restrictions of available hardware resources, the methods were evaluated using a few number of robots. It would be valuable if the methods can be evaluated using a large number of robots in future.

5.2.4 Larger environments

The size of the environment relative to the size of robots is an important parameter in search scenarios. The methods in this thesis were experimented in small-scaled environments. If the methods can be evaluated in very larger environments (compare to size of robots), it will be scientifically valuable.

5.2.5 Outdoor environments

This work considered indoor and semi-outdoor environments. It would be interesting to check the validity of the obtained results in outdoor environments. Similar results are expected if the state of the wind is laminar and does not change rapidly. In situations that the wind direction changes rapidly, a deep study is required to adapt the proposed methods to those scenarios.

References

[AG08] S.B. Akat and V. Gazi. Decentralized asynchronous particle swarm optimization. In *IEEE Swarm Intelligence Symposium*, pages 1–8, 2008.

[Ale07] F. Alecu. Performance analysis of parallel algorithms. *Journal of Applied Quantitative Methods*, page 129, 2007.

[Amd67] G.M. Amdahl. Validity of the single processor approach to achieving large scale computing capabilities. In *Spring joint computer Conf.*, pages 483–485. ACM, April 1967.

[AML+04] K. Arshak, E. Moore, GM Lyons, J. Harris, and S. Clifford. A review of gas sensors employed in electronic nose applications. *Sensor Review*, 24(2):181–198, 2004.

[ANS06] F. ANSYS. FLUENT user's manual. *Software Release*, 6, 2006.

[Ark92] R.C. Arkin. Cooperation without communication: Multiagent schema-based robot navigation. *Journal of Robotic Systems*, 9(3):351–364, 1992.

[Ary99] S.P. Arya. *Air pollution meteorology and dispersion*. Oxford University Press New York, 1999.

[BEL+08] D. Borrmann, J. Elseberg, K. Lingemann, A. Nüchter, and J. Hertzberg. Globally consistent 3D mapping with scan matching. *Robotics and Autonomous Systems*, 56(2):130–142, 2008.

[BHD94] R. Beckers, OE Holland, and J.L. Deneubourg. From local actions to global tasks: Stigmergy and collective robotics. In *Artificial life IV*, volume 181, page 189. MIT Press, 1994.

[BHH09] F. Beutler, M.F. Huber, and U.D. Hanebeck. Probabilistic instantaneous model-based signal processing applied to localization and tracking. *Robotics and Autonomous Systems*, 57(3):249–258, 2009.

[BHR86] J.B. Burns, A.R. Hanson, and E.M. Riseman. Extracting straight lines. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 4:425–455, 1986.

[BL02] R. Bachmayer and N. E. Leonard. Vehicle networks for gradient descent in a sampled environment. In *IEEE Conf. on Decision and Control*, Nevada, USA, 2002.

[BMSS05] W. Burgard, M. Moors, C. Stachniss, and F. Schneider. Coordinated multi-robot exploration. *IEEE Trans. on Robotics*, 21(3):376–386, 2005.

[BP95] G. Bilchev and I. Parmee. The ant colony metaphor for searching continuous design spaces. *Evolutionary Computing*, pages 25–39, 1995.

[Bri73] GA Briggs. Diffusion estimation for small emissions ATDL, contribution file No. 97. *Air Resources Atmospheric Turbulence and Diffusion Laboratory, NOAA, Oak Ridge, Tennessee*, 1973.

[Bro94] R.A. Brooks. *Interaction and intelligent behavior*. PhD thesis, Massachusetts Institute of Technology, 1994.

[BS02] E. Balkovsky and B. Shraiman. Olfactory search at high Reynolds number. In *Proc. National Academy of Science, 99(20)*, USA, 2002.

[BS05] E. Bahceci and E. Sahin. Evolving aggregation behaviors for swarm robotic systems: A systematic case study. In *IEEE Swarm Intelligence Symposium*, California, USA, 2005.

[BS07] M.A. Batalin and G.S. Sukhatme. The design and analysis of an efficient local algorithm for coverage and exploration based on sensor network deployment. *IEEE Trans on Robotics*, 23(4):661–675, 2007.

[BTI11] D. Bhadauria, O. Tekdas, and V. Isler. Robotic data mules for collecting data over sparse sensor fields. *Journal of Field Robotics*, 28(3):388–404, 2011.

[BY10] S. Bando and S. Yuta. Use of the parallel and perpendicular characteristics of building shape for indoor map making and positioning. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, Taiwan, 2010.

[CABJ$^+$11] K.C. Cheng, V. Acevedo-Bolton, R.T. Jiang, N.E. Klepeis, W.R. Ott, O.B. Fringer, and L.M. Hildemann. Modeling exposure close to air pollution sources in naturally ventilated residences: Association of turbulent diffusion coefficient with air change rate. *Environmental science & technology*, 2011.

[CBCR99] R. Cassinis, G. Bianco, A. Cavagnini, and P. Ransenigo. Strategies for navigation of robot swarms to be used in landmines detection. In *Proc. of the Third European Workshop on Advanced Mobile Robots*, 1999.

[CBR02] C.M. Clark, T. Bretl, and S. Rock. Applying kinodynamic randomized motion planning with a dynamic priority system to multi-robot space systems. In *IEEE Aerospace Conf. Proceedings*, volume 7, pages 7–3621, 2002.

[CDF^{+}01] S. Camazine, J.L. Deneubourg, N. Franks, J. Sneyd, G. Theraulaz, and E. Bonabeau. *Self-Organisation in Biological Systems*. Princeton University Press, USA, 2001.

[CFK97] Y. Uny Cao, Alex S. Fukunaga, and Andrew B. Kahng. Cooperative mobile robotics: Antecedents and directions. *Autonomous Robots*, 4:7–27, 1997.

[CHRE04a] X. Cui, C.T. Hardin, R.K. Ragade, and A.S. Elmaghraby. A swarm approach for emission sources localization. *IEEE Computer Society*, 2004.

[CHRE04b] X. Cui, T. Hardin, RK Ragade, and AS Elmaghraby. A swarm-based fuzzy logic control mobile sensor network for hazardous contaminants localization. In *Proc. of the IEEE Int. Conf. on Mobile Ad-hoc and Sensor Systems*, 2004.

[CP95] O. Causse and LH Pampagnin. Management of a multi-robot system in a public environment. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, volume 2, pages 246–252, 1995.

[CP09] S. Carpin and E. Pagello. An experimental study of distributed robot coordination. *Robotics and Autonomous Systems*, 57(2):129–133, 2009.

[CRPM06] C.M. Cianci, X. Raemy, J. Pugh, and A. Martinoli. Communication in a swarm of miniature robots: The e-puck as an educational tool for swarm robotics. In *Proc. of the 2nd Int. Conf. on Swarm robotics*, pages 103–115. Springer-Verlag, 2006.

[CSM10] Gonçalo Cabrita, Pedro Sousa, and Lino Marques. Player/stage simulation of olfactory experiments. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, Taiwan, 2010.

[CWK02] J.P. Crimaldi, M.B. Wiley, and J.R. Koseff. The relationship between mean and instantaneous structure in turbulent passive scalar plumes. *Journal of Turbulence*, 3(14):1–24, 2002.

[DBS06] M. Dorigo, M. Birattari, and T. Stutzle. Ant colony optimization. *IEEE Computational Intelligence Magazine*, 1(4):28–39, 2006.

[DJMW96] G. Dudek, M. Jenkin, E. Milios, and D. Wilkes. A taxonomy for multi-agent robotics. *Autonomous Robots*, 3(4):375–397, 1996.

[DJMW98] G. Dudek, M. Jenkin, E. Milios, and D. Wilkes. Topological exploration with multiple robots. In *Int. Symp. on Robotics and Applications*, Anchorage, AK, USA, 1998.

[DM12] Matthew Dunbabin and Lino Marques. Robotics for environmental monitoring: Significant advancements and applications. *IEEE Robotics and Automation Magazine*, 19(1), March 2012.

[DS00a] G. Dedeoglu and G. Sukhatme. Landmark-based matching algorithm for cooperative mapping by autonomous robots. In *Proc. Int. Symp. on Distributed Autonomous Robotics Systems*, Knoxville, TN, 2000.

[DS00b] G. Dedeoglu and G.S. Sukhatme. Landmark-based matching algorithm for cooperative mapping by autonomous robots. In *Proc. Int. Symp. on Distributed Autonomous Robotics Systems*, 2000.

[DSA01] J.F. Diaz, A. Stoytchev, and R.C. Arkin. Exploring unknown structured environments. In *Int. Fairs Conf.*, 2001.

[DSR04] A. Dhariwal, G.S. Sukhatme, and A.A.G. Requicha. Bacterium-inspired robots for environmental monitoring. In *Proc. IEEE Int. Conf. on Robotics and Automation*, pages 1436–1443, 2004.

[DTRMS94] R. Deveza, D. Thiel, A. Russell, and A. Mackay-Sim. Odor sensing for robot guidance. *The Int. Journal of Robotics Research*, 13(3):232–239, 1994.

[DZKS06] M.B. Dias, R. Zlot, N. Kalra, and A. Stentz. Market-based multirobot coordination: A survey and analysis. *Proceedings of the IEEE*, 94(7):1257–1270, 2006.

[ENT05] C. Estrada, J. Neira, and J.D. Tardós. Hierarchical SLAM: Real-time accurate mapping of large environments. *IEEE Trans. on Robotics*, 21(4):588–596, 2005.

[FCL12] H. Fu, H. Chen, and P. Lin. APS: Distributed air pollution sensing system on Wireless Sensor and Robot Networks. *Computer Communications*, 35(9):1141 – 1150, 2012.

[FCM+09] G. Ferri, E. Caselli, V. Mattoli, A. Mondini, B. Mazzolai, and P. Dario. SPIRAL: A novel biologically-inspired algorithm for gas/odor source localization in an indoor environment with no strong airflow. *Robotics and Autonomous Systems*, 57(4):393–402, 2009.

[FJC+07] G. Ferri, M.V. Jakuba, E. Caselli, V. Mattoli, B. Mazzolai, D.R. Yoerger, and P. Dario. Localizing multiple gas/odor sources in an indoor environment using Bayesian occupancy grid mapping. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2007.

[FKK+06] D. Fox, J. Ko, K. Konolige, B. Limketkai, D. Schulz, and B. Stewart. Distributed multirobot exploration and mapping. *IEEE Special Issue on Multi-Robot Systems*, 94(7):1325–1339, 2006.

[FNL02] J.W. Fenwick, P.M. Newman, and J.J. Leonard. Cooperative concurrent mapping and localization. In *Proc. IEEE Int. Conf. on Robotics and Automation*, volume 2, pages 1810–1817, 2002.

[FP11] J. Faigl and L. Přeučil. Self-organizing map for the multi-goal path planning with polygonal goals. *Artificial Neural Networks and Machine Learning*, pages 85–92, 2011.

[FPL03] J.A. Farrell, S. Pang, and W. Li. Plume mapping via hidden Markov methods. *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, 33(6):850–863, 2003.

[FPLA03] J.A. Farrell, S. Pang, W. Li, and R. Arrieta. Chemical plume tracing experimental results with a REMUS AUV. In *OCEANS Proceedings*, volume 2, pages 962–968. IEEE, 2003.

[FTL07] E. Ferranti, N. Trigoni, and M. Levene. Brick&Mortar: an on-line multi-agent exploration algorithm. In *Proc. IEEE Int. Conf. on Robotics and Automation*, pages 761–767, 2007.

[GBMD06] R. Grofi, M. Bonani, F. Mondada, and M. Dorigo. Autonomous Self-assembly in a Swarm-bot. In *Proc. of the 3rd Int. Symp. on Autonomous Minirobots for Research and Edutainment*, page 314, 2006.

[GD04] D. S. Guru and R. Dinesh. Non-parametric adaptive region of support useful for corner detection: a novel approach. *Pattern Recognition*, 37(1):165 – 168, 2004.

[GDC$^+$97] F.W. Grasso, J.H. Dale, T.R. Consi, D.C. Mountain, and J. Atema. Effectiveness of continuous bilateral sampling for robot chemotaxis in a turbulent odor plume: implications for lobster chemo-orientation. *The Biological Bulletin*, 193(2):215–216, 1997.

[GHL05] A. Galstyan, T. Hogg, and K. Lerman. Modeling and mathematical analysis of swarms of microscopic robots. In *Proc. IEEE Swarm Intelligence Symposium*, pages 201–208, 2005.

[Gif60] F. Gifford. Peak to average concentration ratios according to a fluctuating plume dispersion model. *Int. Journal of Air Pollution*, 3:253, 1960.

[GNSPK00] R. Grabowski, L.E. Navarro-Serment, C.J.J. Paredis, and P.K. Khosla. Heterogeneous teams of modular robots for mapping and exploration. *Autonomous Robots*, 8(3):293–308, 2000.

[GP02] V. Gazi and K.M. Passino. Stability analysis of swarms. In *Proc. American Control Conference*, pages 1813–1818, 2002.

[GP04a] V. Gazi and K.M. Passino. A class of attractions/repulsion functions for stable swarm aggregations. *International Journal of Control*, 77(18):1567–1579, 2004.

[GP04b] V. Gazi and K.M. Passino. Stability analysis of social foraging swarms. *IEEE Trans. on Systems, Man, and Cybernetics, Part B*, 34(1):539–557, 2004.

[GVH03] B. Gerkey, R.T. Vaughan, and A. Howard. The Player/Stage project: Tools for multi-robot and distributed sensor systems. In *Proc. IEEE Int. Conf. on Advanced Robotics*, Coimbra, Portugal, 2003.

[HB05] W.H. Huang and K.R. Beevers. Topological map merging. *The Int. Journal of Robotics Research*, 24(8):601, 2005.

[HBLNT12] V. Hernandez Bennetts, A.J. Lilienthal, P.P. Neumann, and M. Trincavelli. Mobile robots for localizing gas emission sources on landfill sites: is bio-inspiration the way to go? *Frontiers in Neuroengineering*, 4:20, 2012.

[Hin86] J.O. Hinze. Turbulence, 1975. *McGraw-Hill, New York*, 173:303–356, 1986.

[HM99] O. Holland and C. Melhuish. Stigmergy, self-organization, and sorting in collective robotics. *Artificial Life*, 5(2):173–202, 1999.

[HM02] J. Handl and B. Meyer. Improved ant-based clustering and sorting in a document retrieval interface. *Parallel Problem Solving from Nature-PPSN VII*, pages 913–923, 2002.

[HMG02] A. T. Hayes, A. Martinoli, and R. M. Goodman. Distributed odor source localization. *IEEE Sensors Journal, Special Issue on Artificial Olfaction*, 2(3):260–271, 2002.

[HMG03] A.T. Hayes, A. Martinoli, and R.M. Goodman. Swarm robotic odor localization: Off-line optimization and validation with real robots. *Robotica*, 21(04):427–441, 2003.

[HNR68] P.E. Hart, N.J. Nilsson, and B. Raphael. A formal basis for the heuristic determination of minimum cost paths. *IEEE Trans. on Systems Science and Cybernetics*, 4(2):100–107, 1968.

[HWPG07] S. Heo, T. Wiguna, H.C. Park, and N.S. Goo. Effect of an artificial caudal fin on the performance of a biomimetic fish robot propelled by piezoelectric actuators. *Journal of Bionic Engineering*, 4(3):151–158, 2007.

[IKNM96] H. Ishida, Y. Kagawa, T. Nakamoto, and T. Moriizumi. Odor-source localization in the clean room by an autonomous mobile sensing system. *Sensors and Actuators B: Chemical*, 33(1-3):115–121, 1996.

[IMBG01] A.J. Ijspeert, A. Martinoli, A. Billard, and L.M. Gambardella. Collaboration through the exploitation of local interactions in autonomous collective robotics: The stick pulling experiment. *Autonomous Robots*, 11(2):149–171, 2001.

[ISNM94] H. Ishida, K. Suetsugu, T. Nakamoto, and T. Moriizumi. Study of autonomous mobile sensing system for localization of odor source using gas sensors and anemometric sensors. *Sensors and Actuators A: Physical*, 45(2):153–157, 1994.

[ITTM06] H. Ishida, H. Tanaka, H. Taniguchi, and T. Moriizumi. Mobile robot navigation using vision and olfaction to search for a gas/odor source. *Autonomous Robots*, 20(3):231–238, 2006.

[JKWT98] J. Jennings, C. Kirkwood-Watts, and C. Tanis. Distributed map-making and navigation in dynamic environments. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, volume 3, pages 1695–1701, 1998.

[Jon83] C.D. Jones. On the structure of instantaneous plumes in the atmosphere. *Journal of Hazardous Materials*, 7(2):87–112, 1983.

[JZ00] D. Jung and A. Zelinsky. Grounded symbolic communication between heterogeneous cooperating robots. *Autonomous Robots*, 8(3):269–292, 2000.

[Kaz03] S.T. Kazadi. Extension of plume tracking behavior to robot swarms. In *Proc. of the IIIS/SCI Conf. on Systems, Cybernetics, and Informatics*, Orlando, USA, 2003.

[KB00a] M.J.B. Krieger and J.B. Billeter. The call of duty: Self-organised task allocation in a population of up to twelve mobile robots. *Robotics and Autonomous Systems*, 30(1-2):65–84, 2000.

[KB00b] C.R. Kube and E. Bonabeau. Cooperative transport by ants and robots. *Robotics and Autonomous Systems*, 30(1-2):85–102, 2000.

[KES01] J. Kennedy, R. C. Eberhart, and Y. Shi. *Swarm Intelligence*. Morgan Kaufmann, San Francisco, CA, 2001.

[KFL+03] K. Konolige, D. Fox, B. Limketkai, J. Ko, and B. Stewart. Map merging for distributed robot navigation. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, volume 1, pages 212–217, 2003.

[KG08] K.N. Krishnanand and D. Ghose. Theoretical foundations for rendezvous of glowworm-inspired agent swarms at multiple locations. *Robotics and Autonomous Systems*, 56(7):549–569, 2008.

[KH04] E.B. Krissinel and K. Henrick. Common subgraph isomorphism detection by backtracking search. *Software: Practice and Experience*, 34(6):591–607, 2004.

[Kha86] O. Khatib. Real time obstacle avoidance for manipulators and mobile robots. *Int. Journal of Robotics Research*, 5(1):90–99, 1986.

[KRRJ06] G. Kowadlo, D. Rawlinson, R. A. Russell, and R. A. Jarvis. Bi-modal search using complementary sensing (olfaction/vision) for odour source localization. In *Proc. IEEE Int. Conf. on Robotics and Automation*, Florida, USA, 2006.

[KRS90] C.P. Kruskal, L. Rudolph, and M. Snir. A complexity theory of efficient parallel algorithms. *Theoretical Computer Science*, 71(1):95–132, 1990.

[KSF$^+$03] J. Ko, B. Stewart, D. Fox, K. Konolige, and B. Limketkai. A practical, decision-theoretic approach to multi-robot mapping and exploration. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, volume 4, pages 3232–3238, 2003.

[LANM10] T. Lochmatter, E. Aydin, I. Navarro, and A. Martinoli. A plume tracking algorithm based on crosswind formations. In *Proc. Int. Symp. on Distributed Autonomous Robotics Systems*, Lausanne, Switzerland, 2010.

[LBL10] K.Y.K. Leung, T.D. Barfoot, and H. Liu. Decentralized localization of sparsely-communicating robot networks: a centralized-equivalent approach. *IEEE Trans. on Robotics*, 26(1):62–77, 2010.

[LBSS$^+$12] SW Legg, AJ Benavides-Serrano, JD Siirola, JP Watson, SG Davis, A. Bratteteig, and CD Laird. A stochastic programming approach for gas detector placement using cfd-based dispersion simulations. *Computers & Chemical Engineering*, 2012.

[LCLG08] A. Loutfi, S. Coradeschi, A.J. Lilienthal, and J. Gonzalez. Gas distribution mapping of multiple odour sources using a mobile robot. *Robotica*, 27(02):311–319, 2008.

[LD04] A. Lilienthal and T. Duckett. Building gas concentration gridmaps with a mobile robot. *Robotics and Autonomous Systems*, 48(1):3–16, 2004.

[LFC01] W. Li, J.A. Farrell, and R.T. Cardé. Tracking of fluid-advected odor plumes: strategies inspired by insect orientation to pheromone. *Adaptive Behavior*, 9(3-4):143, 2001.

[LKV06] C. Lytridis, E.E. Kadar, and G.S. Virk. A systematic approach to the problem of odour source localisation. *Autonomous Robots*, 20(3):261–276, 2006.

[LLD06] A.J. Lilienthal, A. Loutfi, and T. Duckett. Airborne chemical sensing with mobile robots. *Sensors*, 6(11):1616–1678, 2006.

[LM09] T. Lochmatter and A. Martinoli. Tracking odor plumes in a laminar wind field with bio-inspired algorithms. In *Experimental Robotics*. Springer, 2009.

[LMB$^+$08] F. Li, Q.H. Meng, S. Bai, J.G. Li, and D. Popescu. Probability-PSO algorithm for multi-robot based odor source localization in ventilated indoor environments. *Intelligent Robotics and Applications*, pages 1206–1215, 2008.

[LMS$^+$09] F. Li, Q.H. Meng, J.W. Sun, S. Bai, M. Zeng, et al. Single odor source declaration by using multiple robots. In *American Institute of Physics (AIP) Conference Proceedings*, volume 1137, page 73, 2009.

[LMWZ11] J.G. Li, Q.H. Meng, Y. Wang, and M. Zeng. Odor source localization using a mobile robot in outdoor airflow environments with a particle filter algorithm. *Autonomous Robots*, 30(3):281–292, 2011.

[LP02] Y. Liu and km. Passino. Biomimicry of social foraging behavior for distributed optimization: Models, Principles and emergent behaviors. *Journal of Optimization Theory and Applications*, 115(3):603–628, 2002.

[LRM+08] T. Lochmatter, X. Raemy, L. Matthey, S. Indra, and A. Martinoli. A comparison of casting and spiraling algorithms for odor source localization in laminar flow. In *Proc. IEEE Int. Conf. on Robotics and Automation*, California, USA, 2008.

[LRZ03] A. Lilienthal, D. Reimann, and A. Zell. Gas source tracing with a mobile robot using an adapted moth strategy. *Autonome mobile Systeme*, page 150, 2003.

[LUF+04] A. Lilienthal, H. Ulmer, H. Frohlich, A. Stutzle, F. Werner, and A. Zell. Gas source declaration with a mobile robot. In *ICRA*, volume 2, pages 1430–1435, 2004.

[LWBSL12] S.W. Legg, C. Wang, A.J. Benavides-Serrano, and C.D. Laird. Optimal gas detector placement under uncertainty considering conditional-value-at-risk. *Journal of Loss Prevention in the Process Industries*, 2012.

[MAdA03] L. Marques, N. Almeida, and A. T. de Almeida. Olfactory sensory system for odour-plume tracking and localization. In *IEEE Int. Conf. on Sensors*, Canada, 2003.

[MCM12] A. Marjovi, S. Choobdar, and L. Marques. Robotic clusters: Multi-robot systems as computer clusters a topological map merging demonstration. *Robotics and Autonomous Systems*, 2012.

[MdA06a] L. Marques and A. T. de Almeida. Editorial to mobile robot olfaction. *Autonomous Robots Journal, Special Issue on Mobile Robot Olfaction, Kluwer*, 20(3):183–184, 2006.

[MdA06b] L. Marques and A. T. de Almeida. ThermalSkin: a distributed sensor for anemotaxis robot navigation. In *Proc. 5th IEEE Int. Conf. on Sensors*, pages 1515–1518, South Korea, 2006.

[MEA04] A. Martinoli, K. Easton, and W. Agassounon. Modeling swarm robotic systems: A case study in collaborative distributed manipulation. *The International Journal of Robotics Research*, 23(4-5):415, 2004.

[MEC92] J. Murlis, J.S. Elkinton, and R.T. Carde. Odor plumes and how insects use them. *Annual review of entomology*, 37(1):505–532, 1992.

[MLLZ06] Q.H. Meng, J.C. Li, F. Li, and M. Zeng. Mobile robots odor localization with an improved ant colony algorithm. In *Proc. IEEE Int. Conf. on Robotics and Biomimetics*, pages 959–964, 2006.

[MM10] A. Marjovi and L. Marques. Multi-robot topological exploration using olfactory cues. In *Proc. Int. Symp. on Distributed Autonomous Robotics Systems*, Lausanne, Switzerland, 2010.

[MM11a] A. Marjovi and L. Marques. Multi-robot olfactory search in structured environments. *Robotics and Autonomous Systems*, 52(11):867–881, 2011.

[MM11b] E. Miyata and S. Mori. Optimization of gas detector locations by application of atmospheric dispersion modeling tools. *Sumitomo Kagaku*, I, 2011.

[MNdA02a] L. Marques, U. Nunes, and A. de Almeida. Cooperative odour field exploration with genetic algorithms. In *Proc. 5th Portuguese Conf. on Automatic Control (CONTROLO)*, pages 138–143, 2002.

[MNdA02b] L. Marques, U. Nunes, and A. T. de Almeida. Olfaction-based mobile robot navigation. *Thin Solid Films*, 418(1):51–58, 2002.

[MNdA06] L. Marques, U. Nunes, and A. T. de Almeida. Particle swarm-based olfactory guided search. *Autonomous Robots Journal, Special Issue on Mobile Robot Olfaction, Kluwer*, 20(3):277–287, 2006.

[MNMdA09a] A. Marjovi, J. Nunes, L. Marques, and A. T. de Almeida. Multi-robot exploration and fire searching. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, St. Louis, MO, USA, 2009.

[MNMdA09b] A. Marjovi, J. Nunes., L. Marques, and A. T. de Almeida. Multi-robot fire searching in unknown environment. In *Field and Service Robotics*, Boston, MA, USA, 2009.

[MNMdA10] A. Marjovi, J. Nunes, L. Marques, and A. T. de Almeida. Multi-robot fire searching in unknown environment. In *Field and Service Robotics*, volume 62, pages 341–351. Springer Tracts in Advanced Robotics, 2010.

[MNS$^+$10] A. Marjovi, J Nunes, P. Sousa, R. Faria, and L. Marques. An olfactory-based robot swarm navigation method. In *Proc. IEEE Int. Conf. on Robotics and Automation*, Alaska, USA, 2010.

[MNSB11] Y. Martínez, A. Nowé, J. Suárez, and R. Bello. A reinforcement learning approach for the flexible job shop scheduling problem. *Learning and Intelligent Optimization*, pages 253–262, 2011.

[MRH06] D. Martinez, O. Rochel, and E. Hugues. A biomimetic robot for tracking specific odors in turbulent plumes. *Autonomous Robots*, 20(3):185–195, 2006.

[MYWZ11] Q.H. Meng, W.X. Yang, Y. Wang, and M. Zeng. Collective odor source estimation and search in time-variant airflow environments using mobile robots. *Sensors*, 11(11):10415–10443, 2011.

[NCD08] S. Nouyan, A. Campo, and M. Dorigo. Path formation in a robot swarm. *Swarm Intelligence*, 2(1):1–23, 2008.

[Nor93] F.R. Noreils. Toward a robot architecture integrating cooperation between mobile robots: Application to indoor environment. *The Int. Journal of Robotics Research*, 12(1):79–98, 1993.

[OS07] R. Olfati-Saber. Distributed tracking for mobile sensor networks with information-driven mobility. In *American Control Conference*, pages 4606–4612, 2007.

[PBG09] Z. Pasternak, F. Bartumeus, and F.W. Grasso. Lévy-taxis: a novel search strategy for finding odor plumes in turbulent flow-dominated environments. *Journal of Physics A: Mathematical and Theoretical*, 42:434010, 2009.

[PBiBB+06] P. Pyk, S. Bermúdez i Badia, U. Bernardet, P. Knüsel, M. Carlsson, J. Gu, E. Chanie, B.S. Hansson, T.C. Pearce, and P.F.M. Verschure. An artificial moth: Chemical source localization using a robot based neuronal model of moth optomotor anemotactic search. *Autonomous Robots*, 20(3):197–213, 2006.

[PF06] S. Pang and J.A. Farrell. Chemical plume source localization. *IEEE Trans. on Systems, Man, and Cybernetics, Part B: Cybernetics*, 36(5):1068–1080, 2006.

[PFTV02] W.H. Press, B.P. Flannery, S.A. Teukolsky, and W.T. Vetterling. Direction set (Powell's) methods in multidimensions. *Numerical recipes in C++: The art of scientific computing*, 1992:417–424, 2002.

[PM07] J. Pugh and A. Martinoli. Parallel learning in heterogeneous multi-robot swarms. In *IEEE Congress on Evolutionary Computation*, pages 3839–3846, 2007.

[RBHSW03] R.A. Russell, A. Bab-Hadiashar, R.L. Shepherd, and G.G. Wallace. A comparison of reactive robot chemotaxis algorithms. *Robotics and Autonomous Systems*, 45(2):83–97, 2003.

[Rey87] C. W. Reynolds. Flocks, herds, and schools: A distributed behavioral model. In *14th Annual Conf. on Computer Graphics and Interactive Techniques*, volume 21, New York, USA, 1987.

[RM06] A. Romeo and L. Montano. Environment understanding: Robust feature extraction from range sensor data. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pages 3337–3343, 2006.

[RMV91] R. Rozas, J. Morales, and D. Vega. Artificial smell detection for robotic navigation. In *Proc. IEEE Int. Conf. on Advanced Robotics*, pages 1730–1733, 1991.

[RTDMS95] RA Russell, D. Thiel, R. Deveza, and A. Mackay-Sim. A robotic system to locate hazardous chemical leaks. In *Proc. IEEE Int. Conf. on Robotics and Automation*, 1995.

[Rus99] R.A. Russell. *Odour detection by mobile robots*, volume 22. World Scientific Publishing Company Incorporated, 1999.

[Rus03] R.A. Russell. Chemical source location and the robomole project. In *Proc. of the Australasian Conf. on Robotics and Automation*, 2003.

[RW02] P.J.W. Roberts and D.R. Webster. *Turbulent diffusion*. ASCE Press, Reston, Virginia, 2002.

[Şah04] E. Şahin. Swarm robotics: From sources of inspiration to domains of application. *Swarm Robotics*, pages 10–20, 2004.

[SF93] K. Singh and K. Fujimura. Map making by cooperating mobile robots. In *Proc. IEEE Int. Conf. on Robotics and Automation*, 1993.

[SGO+07] P. Scerri, R. Glinton, S. Owens, D. Scerri, and K. Sycara. Geolocation of RF emitters by many UAVs. In *American Institute of Aeronautics and Astronautics Infotech@ Aerospace*, 2007.

[Sha05] C. Shao. *Biologically-inspired optimal control*. PhD thesis, University of Maryland, 2005.

[SKS06] D. Spears, W. Kerr, and W. Spears. Physics-based robot swarms for coverage problems. *International Journal on Intelligent Control and Systems*, 11(3):11–23, 2006.

[SLV93] G. Sandini, G. Lucarini, and M. Varoli. Gradient driven self-organizing systems. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, volume 1, 1993.

[SS84] M.W. Sabelis and P. Schippers. Variable wind directions and anemotactic strategies of searching for an odour plume. *Oecologia*, 63(2):225–228, 1984.

[SS02] L. Shapiro and G. Stockman. *Computer Vision*. Prentice Hall, 2002.

[SS05] O. Soysal and E. Sahin. Probabilistic aggregation strategies in swarm robotic systems. In *IEEE Swarm Intelligence Symposium*, California, USA, 2005.

[SŞ07] O. Soysal and E. Şahin. A macroscopic model for self-organized aggregation in swarm robotic systems. *Swarm Robotics*, pages 27–42, 2007.

[Ste94] L. Steels. A case study in the behavior-oriented design of autonomous agents. *From animals to animats*, 3:445–452, 1994.

[Sto11] J.M. Stockie. The mathematics of atmospheric dispersion modeling. *Siam Review*, 53(2):349–372, 2011.

[Sut47] O.G. Sutton. The problem of diffusion in the lower atmosphere. *Quarterly Journal of the Royal Meteorological Society*, 73(317-318):257–281, 1947.

[SYTX06] W. Sheng, Q. Yang, J. Tan, and N. Xi. Distributed multi-robot coordination in area exploration. *Robotics and Autonomous Systems*, 54(12):945–955, 2006.

[Tal09] E.G. Talbi. *Metaheuristics: from design to implementation*. Wiley, 2009.

[TAT$^+$06] H. Tamimi, H. Andreasson, A. Treptow, T. Duckett, and A. Zell. Localization of mobile robots with omnidirectional vision using Particle Filter and iterative SIFT. *Robotics and Autonomous Systems*, 54(9):758–765, 2006.

[TCK$^+$12] Mirbek Turduev, Gonalo Cabrita, Murat Krtay, Veysel Gazi, and Lino Marques. Experimental studies on chemical concentration map building by a multi-robot system using bio-inspired algorithms. *Autonomous Agents and Multi-Agent Systems*, pages 1–29, 2012.

[TDK02] P. Tangamchit, J.M. Dolan, and P.K. Khosla. The necessity of average rewards in cooperative multirobot learning. In *Proc. IEEE Int. Conf. on Robotics and Automation*, volume 2, 2002.

[Thr01] S. Thrun. A probabilistic on-line mapping algorithm for teams of mobile robots. *The Int. Journal of Robotics Research*, 20(5):335–363, 2001.

[TP07] F. Tang and L.E. Parker. A complete methodology for generating multi-robot task solutions using ASyMTRe-D and market-based task allocation. In *Proc. IEEE Int. Conf. on Robotics and Automation*, Italy, 2007.

[VK98] G.S. Virk and E.E. Kadar. Co-operative navigation for target searching in a diffusion field. In *Proc. of the IEEE Int. Conf. on Control Applications*, pages 423–427, 1998.

[Vog96] S. Vogel. *Life in moving fluids: the physical biology of flow*. Princeton University Press, 1996.

[VVS07] M. Vergassola, E. Villermaux, and B.I. Shraiman. 'infotaxis' as a strategy for searching without gradients. *Nature*, 445(7126):406–409, 2007.

[WdS08] Y. Wang and C.W. de Silva. A machine-learning approach to multi-robot coordination. *Engineering Applications of Artificial Intelligence*, 21(3):470–484, 2008.

[WJD08] H. Wang, M. Jenkin, and P. Dymond. Enhancing exploration in graph-like worlds. In *IEEE Canadian Conf. on Computer and Robot Vision*, pages 53–60, 2008.

[Wya03] T.D. Wyatt. *Pheromones and animal behaviour: communication by smell and taste*. Cambridge University Press, 2003.

[WZWH10] H. Wang, M. Zhang, J. Wang, and M. Huang. Engineering an emergency search and rescue application with wireless sensor network and mobile robot. In *Int. Conf. on Measuring Technology and Mechatronics Automation*, volume 2, pages 112–115, 2010.

[XL05] Chen Xin and Yangmin Li. Odor localization using swarm intelligence. In *IEEE Int. Conf. on Robotics and Biomimetics*, Hong Kong, 2005.

[Yam98] B. Yamauchi. Frontier-based exploration using multiple robots. In *Proc. of 2nd Int. Conf. on Autonomous Agents*, 1998.

[YN10] C.H. Yu and R. Nagpal. A Self-Adaptive Framework for Modular Robots in Dynamic Environment: Theory and Applications. *The Int. Journal of Robotics Research*, 2010.

[ZC07] TF Zhang and Q. Chen. Identification of contaminant sources in enclosed environments by inverse cfd modeling. *Indoor air*, 17(3):167–177, 2007.

[ZLC09] Y. Zou, D. Luo, and W. Chen. Swarm robotic odor source localization using ant colony algorithm. In *Proc. IEEE Int. Conf. on Control and Automation*, pages 792–796, 2009.

[ZSBDT02] R. Zlot, A. Stentz, M. Bernardine Dias, and S. Thayer. Multi-robot exploration controlled by a market economy. In *Proc. IEEE Int. Conf. on Robotics and Automation*, Washington DC, USA, 2002.

Appendix A

Publications

Journal papers and book chapters

- "Robotic clusters: Multi-robot systems as computer clusters, a topological map merging demonstration" Ali Marjovi, Sarvenaz Choobdar, Lino Marques. Robotics and Autonomous Systems, 60(9),1191-1204, 2012.
- "Multi-robot olfactory search in structured environments" Ali Marjovi, Lino Marques. Robotics and Autonomous Systems, 52(11):867-881, 2011.
- "Optimal Spatial Formation of Swarm Robotic Gas Sensors in Odor Plume Finding" Ali Marjovi, Lino Marques. submitted to, Autonomous Robots. Under review.
- "Multi-Robot Topological Exploration Using Olfactory Cues" Ali Marjovi, Lino Marques. in Distributed Autonomous Robotic Systems, Springer Tracts in Advanced Robotics, 2012.
- "Multi-Robot Fire Searching in Unknown Environment" Ali Marjovi, João Nunes, Lino Marques, Anibal T. de Almeida. in Field and Service Robotics, Springer Tracts in Advanced Robotics, V.62, pp. 341-351, 2010.

Conference papers

- "Optimal Coverage of Multiple Robotic Gas Sensors in Finding Odor Plumes" Workshop on Robotics for Environmental Monitoring in IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, Vilamoura, Portugal, 2012.
- "An Olfactory-Based Robot Swarm Navigation Method" Ali Marjovi, João Nunes, Pedro Sousa, Ricardo Faria, Lino Marques. IEEE Int. Conf. on Robotics and Automation, Anchorage, Alaska, USA, 2010.
- "Olfactory Search and Topological Exploration with Multiple Robots" Ali Marjovi, Lino Marques. 10th Int. Symp. on Distributed Autonomous Robotics Systems, Lausanne, Switzerland, 2010.

- “Multi-Robot Exploration and Fire Searching” Ali Marjovi, João Nunes, Lino Marques, Anibal de Almeida. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, St. Louis, MO, USA, 2009.
- “Guardians Robot Swarm Exploration and Firefighter Assistance” Ali Marjovi, Lino Marques, Jacques Penders. Workshop on NRS in IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, St. Louis, MO, USA, 2009.
- “A Scalable benchmark for motion control of mobile robots” Ali Marjovi, Lino Marques. 11th International Conference on Climbing and Walking Robots (CLAWAR), Coimbra, Portugal, 2008.
- “A climbing robot for inspection of 3D human made structures” Mahmoud Tavakoli, Ali Marjovi, Lino Marques. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, Nice, France, 2008.
- “A step toward autonomous pole climbing robots” Mahmoud Tavakoli, Ali Marjovi, Lino Marques, Anibal T. de Almeida. 11th International Conference on Climbing and Walking Robots (CLAWAR), Coimbra, Portugal, 2008.
- “BDel, an emotional Fuzzy Based Card player” Ali Marjovi, Saeed Bagheri. Int. Conf. on Modeling, Simulation and Applied Optimization, Sharjah, U.A.E, 2005.
- “Direct Imitation, a novel approach for controlling humanoid robots” Ali Marjovi, Saeed Bagheri. 10th Int. CSI Computer Conf., Tehran, Iran, 2005.
- “Design of a controller for a 4 DOF parallel pole climbing robot” Ali Marjovi, Saeed Bagheri, 10th Int. CSI Computer Conf., Tehran, Iran, 2005.