

Article

# Intelligent Mobile Wireless Network for Toxic Gas Cloud Monitoring and Tracking

Mateusz Krzysztoń <sup>1</sup>  and Ewa Niewiadomska-Szynkiewicz <sup>2,\*</sup> 

<sup>1</sup> Research and Academic Computer Network (NASK), Kolska 12, 01-045 Warsaw, Poland; mateusz.krzyszton@nask.pl

<sup>2</sup> Institute of Control and Computation Engineering, Warsaw University of Technology, Nowowiejska 15/19, 00-665 Warsaw, Poland

\* Correspondence: ens@ia.pw.edu.pl; Tel.: +48-602-411-914

**Abstract:** Intelligent wireless networks that comprise self-organizing autonomous vehicles equipped with punctual sensors and radio modules support many hostile and harsh environment monitoring systems. This work's contribution shows the benefits of applying such networks to estimate clouds' boundaries created by hazardous toxic substances heavier than air when accidentally released into the atmosphere. The paper addresses issues concerning sensing networks' design, focussing on a computing scheme for online motion trajectory calculation and data exchange. A three-stage approach that incorporates three algorithms for sensing devices' displacement calculation in a collaborative network according to the current task, namely exploration and gas cloud detection, boundary detection and estimation, and tracking the evolving cloud, is presented. A network connectivity-maintaining virtual force mobility model is used to calculate subsequent sensor positions, and multi-hop communication is used for data exchange. The main focus is on the efficient tracking of the cloud boundary. The proposed sensing scheme is sensitive to crucial mobility model parameters. The paper presents five procedures for calculating the optimal values of these parameters. In contrast to widely used techniques, the presented approach to gas cloud monitoring does not calculate sensors' displacements based on exact values of gas concentration and concentration gradients. The sensor readings are reduced to two values: the gas concentration below or greater than the safe value. The utility and efficiency of the presented method were justified through extensive simulations, giving encouraging results. The test cases were carried out on several scenarios with regular and irregular shapes of clouds generated using a widely used box model that describes the heavy gas dispersion in the atmospheric air. The simulation results demonstrate that using only a rough measurement indicating that the threshold concentration value was exceeded can detect and efficiently track a gas cloud boundary. This makes the sensing system less sensitive to the quality of the gas concentration measurement. Thus, it can be easily used to detect real phenomena. Significant results are recommendations on selecting procedures for computing mobility model parameters while tracking clouds with different shapes and determining optimal values of these parameters in convex and nonconvex cloud boundaries.

**Keywords:** phenomena clouds monitoring; boundary estimation; MANET; wireless sensor network; self-organization; artificial potential field; simulation



**Citation:** Krzysztoń, M.; Niewiadomska-Szynkiewicz, E. Intelligent Mobile Wireless Network for Toxic Gas Cloud Monitoring and Tracking. *Sensors* **2021**, *21*, 3625. <https://doi.org/10.3390/s21113625>

Academic Editor: Raffaele Bruno

Received: 25 April 2021

Accepted: 19 May 2021

Published: 23 May 2021

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Copyright:** © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

Modern industry produces substances whose volatilization causes a severe threat to the environment. Many of these hazardous substances form gas clouds heavier than air when accidentally released into the atmosphere. As a result, the gas cloud changes its shape, which is usually irregular, moves and covers a larger area with time. Toxic gases usually endanger humans [1]. The strategic step for conducting a situational assessment is to create a sensing system to determine a gas-covered area with a concentration greater than the safe value [2].

Research and commercial implementations in recent years have shown that self-configuring networks built from static and mobile wireless devices equipped with sensing modules can significantly enhance the capability to investigate and sense unknown environments [3]. Such networks can be successfully used to monitor phenomena, such as clouds created by heavy hazardous toxic substances [4]. Mobile devices equipped with a Central Processing Unit (CPU), radio transceiver, Global Positioning System (GPS), and gas detector can exchange data and dynamically change both the position and role in the network. However, to ensure accurate measurement and effective tracking of moving clouds, an efficient computing scheme for online trajectory planning and a reliable interconnection network of sensing devices is required. Moreover, low processing power, low-quality wireless communication, limited radio range and energy resources are significant limitations that make the implementation of such a network even more complex.

This paper presents a comprehensive approach to heavy gas cloud monitoring that can be used for disaster management and toxic spill counteraction using unmanned ground vehicles or drones. Specifically, it describes a computing scheme for creating sensing mobile ad hoc networks (MANETs) for estimating and tracking cloud boundaries and reporting the results to the central dispatcher of the system (base station) to build situational awareness. The proposed computing scheme incorporates gas dispersion modeling [5], wireless transmission modeling [6], motion modeling [7] and clustering techniques [8]. The network operation consists of three stages: (i) exploration of the sensing scene and gas cloud detection, (ii) exploration of the cloud and its boundary detection, (iii) tracking the evolving and moving cloud. The focus is on mobility modeling and adaptation of the boundary tracking algorithm to various shapes of gas clouds. Our approach's main originality is three algorithms for online network node displacement calculations in the mentioned phases of a gas cloud monitoring process. New positions of sensing devices are computed, solving the optimization problems with differentiable real-valued performance measures, which are viewed as energy. To improve the sensing network's long-term stable operation, multi-hop communication with the base station (BS) is implemented. Thus, a cooperative and coherent network comprising fully autonomous smart sensors that can operate without any external fixed communication infrastructure is proposed. It is assumed that sensing devices can constantly exchange data with their neighbors and finally transmit measurements to BS. There is no central control in the system, and all nodes make autonomous decisions based on sensor readings and data provided by their neighbors. Multi-hop communication with BS reduces the energy costs and communication delays compared to direct communication between BS and a given sensor node. It enables the system to monitor vast areas covered with toxic substances and tracking clouds located at a considerable distance from the base station. In general, it improves system scalability, resilience and reliability.

In most sensing systems described in the literature, an exact value of gas concentration and its gradient are used to calculate the expected positions of sensors. The concentration gradient indicates the boundary of a cloud. It is possible to assess the gas concentration in the whole sensing scene using adequate gas dispersion models and a series of measurements. Due to negative buoyancy, heavy gas clouds' behavior differs from typical pollution clouds [9]. The dynamics of changes are minor. Despite this, the monitoring and tracking of such phenomena is a challenging problem. In the case of gases heavier than air, even slight changes to the terrain seriously influence the concentration gradient. The extent and shape of the cloud depend on current atmospheric conditions, including wind strength. Unfortunately, most gas dispersion models do not consider changes to the terrain and obstacles [10]. It is assumed that the cloud moves over a flat area and ignores local perturbations of the gas concentration. It results in a simplified situation where the gas concentration decreases towards the cloud boundary. In reality, this situation does not usually occur. Moreover, heavy gas clouds are characterized only by a negative vertical concentration gradient, which does not help detect the cloud boundary. The vertical gradient causes turbulent gas mixing, further disturbing the horizontal gas concentration

regularity of the distribution. Therefore, methods that determine the direction of motion based on the exact value of the measured concentration can give outstanding results in simulators and fail in the real world. In our system, sensor readings are reduced to two values, namely the gas concentration below or greater than the safe value, to decrease its sensitivity to the accuracy of gas concentration measurements.

To sum up, this work's main contribution is to present the general overview of the system for detection of a heavy gas cloud in a working space  $W$ , detection of its boundary, and its estimation and tracking. The algorithms for motion planning and cooperation of the sensing devices used at each stage of the gas cloud exploration are presented and discussed. The system can be used for disaster management using unmanned vehicles or drones equipped with sensors. The data collected by the sensors and transmitted to the emergency management center will allow creating emergency awareness, supporting the evacuation of people from the endangered area, supporting the rescue teams conducting operational activities, and neutralizing a toxic spill.

The presented results extend the work described in [11]. The paper [11] addresses the problem of cloud boundary detection and estimation. In this paper, a brief overview of algorithms for cloud boundary detection and estimation is provided. The attention is paid to the most challenging task, namely, a boundary of slowly moving cloud tracking. The main contribution is a novel algorithm for convex and nonconvex gas cloud boundary tracking and five procedures for tuning the algorithm's parameters. An exhaustive simulation study that shows the effectiveness of these procedures depending on the considered scenario is presented. Finally, we provide recommendations for the choice of a procedure, respectively, for a given cloud shape.

The rest of the paper is structured as follows. Section 2 presents the survey of the application of MANETs to monitor and track phenomena clouds. Section 3 provides a formulation of the problem to be solved and the model of a network composed of smart wireless sensing devices. Section 4 describes the concept of a three-stage strategy for heavy gas cloud monitoring and tracking and computing schemes for online motion trajectories calculation. A comparative study of a few variants of procedures for tuning model parameters is presented in Section 5. Finally, Section 6 concludes the paper and highlights future research directions. The Appendix A provides a list of notations used that is standard across all of the sections.

## 2. Related Work

The problem of developing effective and valuable strategies for detecting, estimating and tracking boundaries using collaborating static and mobile sensing devices has been studied in recent years by many researchers. Various techniques have been investigated, implemented and tested through simulation and in testbeds constructed of real devices. Shu et al. in [12] discuss the research directions for existing and future gas leakage source detection and boundary estimation schemes with wireless sensor networks. The authors of [13] present a survey of selected approaches for monitoring phenomena clouds. They can be classified concerning various criteria, such as a task to be performed, equipment used and expected network configuration, sensing and control strategies, communication and computing algorithms, and the accuracy of utilized gas dispersion models. A significant problem addressed in the literature is the resilience of the monitoring network to failures and insufficient energy resources. Imran and Ko in [14] present an energy-efficient method for detecting and estimating continuous objects spread over a large area by a failure-prone network. Improving the accuracy of boundary estimation while reducing the energy consumption is considered in [15].

This paper focuses on mobile sensing networks for phenomena-cloud boundary estimation and tracking. In general, we can distinguish simple systems built of a single mobile sensor or more complex systems comprised of a team of sensing devices. The simple systems for gas cloud boundary tracking are described in [16,17]. Wang et al. in [16] propose the algorithm for boundary estimation and tracking using a gas concentration gradient. A

single moving platform equipped with the punctual sensor explores the sensing scene and constantly measures the gas concentration. The concentration gradient and consequently a new position of the platform are calculated based on these measurements. The algorithm for control the motion of a single unmanned underwater vehicle (UUV) equipped with sensors is presented and examined in [17]. Similarly to [16], both the initial detection of the cloud boundary and the tracking procedures are based on calculating the toxic substance concentration gradient. Moreover, the current speed measured by a dedicated sensor is taken into account in motion trajectory calculation. The proposed control algorithm was validated in the natural water reservoir with a dyeing substance imitating the toxic one. The paper [18] addresses the problem of boundary tracking for the mobile robot with uncertain dynamics and external disturbances. Sun et al. present and evaluate an adaptive control system using a radial basis function neural network to approximate a nonlinear function containing the uncertain model terms and toxic substance concentration gradient. Simulation results illustrate the stability of the system.

Another group of systems is networks built from cooperating or non-cooperating devices. The scheme for data gathering by multiple autonomous and non-cooperating underwater vehicles is described and discussed in [19]. In this approach, each device's target position is determined autonomously. The current positions of other team members do not influence this device's position calculation. A similar approach with multiple autonomous vehicles used to track boundaries is described in [20].

Many researchers address the problem of boundary tracking by a team of cooperating vehicles. Singh et al. in [21] describe a simple mechanism for unmanned vehicle cooperation. It is composed of two phases: first, mobile devices that form the network explore roughly the whole area of interest, then the initial shape of a boundary is discovered. In the second phase, each node follows the boundary to increase the accuracy of the initial estimation. Singh et al. demonstrate that their strategy can be easily adapted to a network consisting of multiple devices. In the case of a wide sensing area, it can be divided into subregions that are equally assigned to all nodes. Data collected by all nodes are gathered by a selected node that estimates the boundary's shape. Next, the estimated boundary is divided into parts that are assigned to nodes, respectively, for further investigation. This approaches' main drawback is that each device has to cover a long distance in the case of a vast cloud. Triandaf and Schwartz in [22] present a method for calculating the motion path for the formation of communicating sensors. The aim of all sensors is to follow a time-dependent concentration gradient. The algorithm allows the sensors to move in space in a non-stationary environment. It can be used for finding and tracking the boundary of any stable surface. The common feature of the solutions proposed in [23,24] is the concept of nominating a leader of the whole formation. This leader is selected from the group of devices located close to the cloud boundary. The objective of the sensing system described in [23] is to detect and track a wildfire border. It is assumed that network nodes measure the temperature of the air. The Rothermel model describing a spreading fire and the Kalman filter to detect a boundary in a sensing space are used to develop an algorithm for the motion control of all sensing devices. Unfortunately, because the Rothermel model is directly used to calculate all sensing devices' target positions, this computing scheme can not be quickly adapted to other use cases. In [24], a team of four mobile robots is used to track the boundary of a phenomena cloud. A concentration gradient is used to calculate the target positions of all robots. At every time step, all devices equipped with punctual sensors measure the concentration of substances, and the concentration gradient is calculated. All devices' target positions are calculated, solving the optimization problem with the performance function dependent on the concentration gradient. The application of algorithms for motion planning based on the concentration gradient and the concept of a virtual potential field [7], which are widely used in mobile robotics, is presented in [25]. Both techniques were verified and validated through simulation of the underwater oil spill tracking by unmanned underwater vehicles.

An alternative approach to tracking heavy gas clouds is to apply accurate models of the phenomena and use computer simulation to predict cloud dynamics and environmental risk. The heavy gas dispersion simulators SLAB [26] and Fluidyn-PANACHE [27] are commonly used. In [1], rapid assessment of exposure to chlorine released from a train derailment is described. However, to estimate the area covered by the heavy gas cloud, additional data and measurements have to be provided, i.e., landform, wind direction and strength, and obstacles presence. The quality of the cloud's boundary estimation is usually sensitive to the accuracy of these data.

Reviewing the literature shows that most of the proposed gas cloud tracking systems built from mobile sensing devices use accurate gas concentrations in a sensing space and centralized schemes for node position calculation. The disadvantages of such an approach are discussed in Section 1. Therefore, our research focused on developing a solution that does not utilize the concentration gradient and implements the distributed mobility model.

### 3. Problem Formulation and Model of a Mobile Wireless Sensing Network

The aim is to design a sensing network composed of intelligent mobile wireless devices to explore an unknown environment to detect the toxic gas cloud, detect its boundaries, and finally, monitor and track the boundary of the evolving and moving cloud. Consider a network comprised by a set  $\mathcal{V}$  of mobile sensing devices (network nodes)  $D_i, i = 1, \dots, N$

$$\mathcal{G} = (\mathcal{V}, \mathcal{E}), \mathcal{V} \neq \emptyset, \mathcal{E} \neq \emptyset, \quad (1)$$

where

$$\mathcal{V} = \{D_i, i = 1, \dots, N\}, \quad (2)$$

$$\mathcal{E} = \{(D_i, D_j) : D_i \in \mathcal{V}, D_j \in \mathcal{V}, d_{ij}^i \leq r_t, i, j = 1, \dots, N, i \neq j\}. \quad (3)$$

In the above formulas,  $\mathcal{E}$  denotes the set of active direct connections between each pair of devices  $D_i$ , and  $D_j, i \neq j, (D_i, D_j)$  is a bidirectional link. Each  $D_i$  is equipped with the radio transceiver with radio range  $r_t$  and a positioning system (GPS). Its position is described by a reference point  $\mathbf{x}_i = [x_{i1}, x_{i2}, x_{i3}]$ , which is the location of its antenna.  $d_{ij}^i = \|\mathbf{x}_i - \mathbf{x}_j\|$  is the Euclidean distance between  $D_i$  and  $D_j$ . Assume that each node  $D_i$  can freely change both its position and role in a network according to its knowledge about the environment and the network topology.

Each  $D_i$  uses a punctual sensor to measure the gas concentration  $g$  at a given point in a workspace  $W$  every regular time interval  $\Delta t$ . For the sake of simplicity, it is assumed that each sensor at time  $t$  returns one of two values

- $g(\mathbf{x}, t) = 1$ : the gas concentration exceeds the threshold value  $\bar{g}$ ,
- $g(\mathbf{x}, t) = 0$ : otherwise.

This approach does not use a concentration gradient. Therefore, local disturbances in gas concentration can be ignored.

Each  $D_i$  can move with the speed  $v_i \in [v_{i_{min}}, v_{i_{max}}]$  in a desirable direction. The common direction in motion planning is to apply an artificial potential function  $V$  that can be viewed as a landscape where the device moves from a high-value state to a low-value state. A value of this function can be viewed as energy and its gradient as a force.  $V$  can be constructed as a sum of repulsive and attractive potentials, i.e.,  $V(d) = V_-(d) + V_+$ , where  $d$  denotes a distance between a given device and other devices in a network or obstacles in the working space. The meaning of  $V_-(d)$  and  $V_+$  is straightforward; the obstacle repels the device, the target point attracts it. Hence, the sum of both these influences draws the device to the target position while deflecting it from obstacles.

In our research, we have adopted a network connectivity-maintaining, virtual force mobility model described in [28] to calculate motion trajectories of platforms carrying sensors. In this model, a simple artificial potential function drawing on the Lennard–Jones potential used in liquid crystals is applied to model the interactions between moving



devices and their displacement calculation. The function  $V_j^i$  that models the interactions between two devices  $D_i$  and  $D_j$  is as follows

$$V_j^i(d_j^i) = \left( \frac{\bar{d}_j^i}{d_j^i} - 1 \right)^2, \quad d_j^i = \|\mathbf{x}_i - \mathbf{x}_j\|, \quad (4)$$

where  $\bar{d}_j^i$  denotes the reference distance between  $D_i$  and  $D_j$ . The total potential between  $D_i$  and all other  $m$  objects in the working space, i.e., other sensing devices and obstacles in the working space and target position, can be expressed as follows

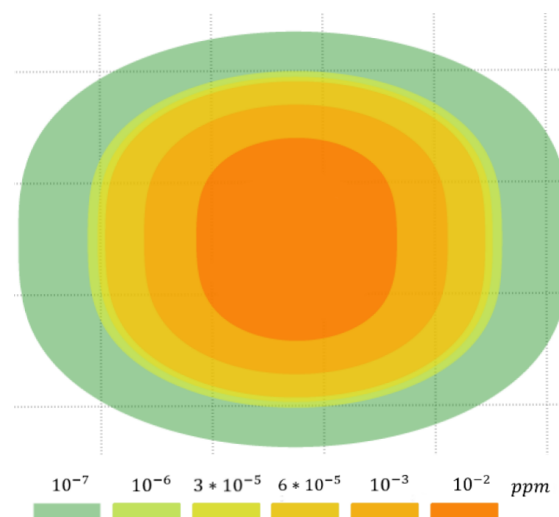
$$V_j^i(d_1^i, d_2^i, \dots, d_m^i) = \sum_{j=1}^m \epsilon_j^i V_j^i(d_j^i). \quad (5)$$

where  $\epsilon_j^i > 0$  denotes the weighting factor determining the importance of the impact of the object  $j$  on  $D_i$ . Moreover, since the external communication system can be broken or suffer from congestion in the disaster area, it is assumed that the distance between the selected, critical pairs of nodes should not be greater than the specified safe distance that guarantees permanent connectivity in a network.

The task is to calculate the optimal positions of all  $D_i, i = 1, \dots, N$  at given time steps for a given application scenario. From Equation (4), it is obvious that in the reference positions of all  $D_i$ , namely such that  $\bar{d}_j^i = d_j^i$ , an optimal network topology is obtained.

#### 4. An Overview of a Method for Heavy Gas Cloud Monitoring and Tracking

The application scenario considered in this paper is heavy gas cloud detecting and monitoring. Figure 1 shows a typical heavy gas cloud. Due to its dispersion characteristics, heavy gas forms a cloud with the largest cross-section at its base.



**Figure 1.** Typical characteristics of a heavy gas cloud.

In situational awareness tasks, determining the extent of contamination and delineating the area at risk is critical. Moreover, in most heavy gas propagation simulators, the cloud moves over a flat area and ignores local terrain and obstacles, which also cause changes in gas density. Therefore, although our virtual force motion model allows motion trajectories to be determined in three-dimensional space, the research presented in this paper focuses on the two-dimensional and obstacle-free workspace.

Consider the self-organizing network defined in Equations (1)–(5). Assume the sensing devices (network nodes)  $D_i$  ( $i = 1, \dots, N$ ) do not have any information about the environment, particularly the cloud location. However, they can exchange data with each other about their positions and the measured gas concentrations. Based on this information, they autonomously determine the direction and speed of movement in a workspace. The limitation of the number of sensor readings  $g(\mathbf{x}, t)$  to two values (0 and 1) allows fast detection of the cloud boundary. Changing the reading value indicates crossing the boundary.

Our heavy gas cloud monitoring method is composed of three main steps executed sequentially.

- Stage 1: workspace exploration to search for a gas cloud.
- Stage 2: cloud exploration to detect its boundary.
- Stage 3: permanent exploring and tracking of a cloud boundary.

All network nodes perform at regular intervals  $\Delta t$  of gas concentration measurements and exchange information about their locations and current sensors readings. Then, based on the obtained data, an optimal new target position of each device is calculated.

#### 4.1. Stage I: Gas Cloud Detection

A team of  $N$  sensing devices  $D_i$ ,  $i = 1, \dots, N$  is used to explore the sensing scene  $W$  to search for a gas cloud. The device  $D_1$  is responsible for maintaining the connectivity with the base station (BS)—the system's central station. Other devices enable constant communication with  $D_1$  using multi-hop transmission. Hence, all  $D_i$ ,  $i = 1, \dots, N$  create the coherent searching network that maintains continuous connectivity with BS.

The first stage begins with setting the target point  $\mathbf{c}$  in the workspace. It is randomly selected. The device closest to the point  $\mathbf{c}$  is elected to be a temporary leader  $D_{leader}$  of the team.  $D_{leader}$  is forced to move in the direction of  $\mathbf{c}$ , the other nodes are forced to follow  $D_{leader}$ . A new position of  $D_{leader}$  is calculated every  $\Delta t$  solving the following optimization problem

$$\min_{\mathbf{x}_i} \left[ V^i = \left( \frac{\bar{d}_c^i}{d_c^i} - 1 \right)^2 \right], \quad d_c^i = \|\mathbf{c} - \mathbf{x}_i\|, \quad (6)$$

with reference distance  $\bar{d}_c^i > 0$  closed to 0. Other devices follow the  $D_{leader}$ . They move in formation. The process is repeated until the cloud is detected by at least one sensing device  $D_k \in \mathcal{V}$ . Then  $D_k$  continues its movement, and the remaining devices are attracted to  $D_k$ . Their displacements are calculated as follows

$$\min_{\mathbf{x}_i} \left[ V^i = \left( \frac{\bar{d}_k^i}{d_k^i} - 1 \right)^2 \right], \quad i = 1, \dots, N, \quad i \neq k, \quad (7)$$

with positive reference distance  $\bar{d}_k^i \approx 0$ .

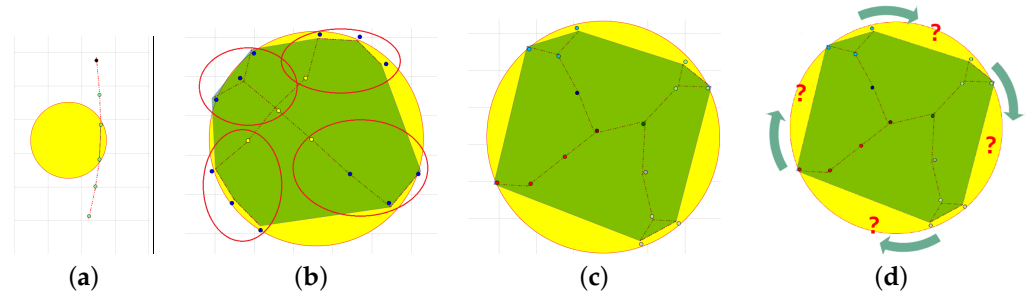
The cloud can move and change its shape over time. All  $D_i$  measure the gas concentration every  $\Delta t$  and in the case of  $g(\mathbf{x}_i, t) = 0$ , they move towards the cloud. The first stage ends when sensors of at least two devices detect gas concentration higher than  $\bar{g}$  (Figure 2a).

$$\exists_{(i,j), i \in \{1, \dots, N\}, j \in \{1, \dots, N\}, i \neq j} : g(\mathbf{x}_i, t) = 1 \wedge g(\mathbf{x}_j, t) = 1. \quad (8)$$

Then, the location  $\Psi$  of the gas cloud center is estimated based on data received from devices located inside the cloud.

$$\Psi = \frac{\sum_{D_i \in \mathcal{V}^+} \mathbf{x}_i}{|\mathcal{V}^+|}, \quad (9)$$

where  $\mathcal{V}^+$  denotes a set of devices located inside the cloud. The number of devices with positive sensor readings increases in time.  $\Psi$  is updated based on new data received from all these devices.



**Figure 2.** Gas cloud detection and monitoring: (a) Stage 1 completed (b) Stage 2: clusters ( $M = 4$ ) and temporal topology, (c) Stage 2 completed, (d) Stage 3: potential direction of boundary tracking.

#### 4.2. Stage 2: Gas Cloud Boundary Detection

The second stage aims to extract the cloud from the environment, namely, detect its boundary, not explore its inside. The boundary is defined as a curve dividing the workspace  $W$  into two regions, respectively, with a gas concentration greater and lower than a given concentration threshold  $\hat{g}$ . The optimal solution of Stage 2 is an even coverage of the cloud boundary with sensors. Hence, the target positions of all devices should be on the cloud boundary. To solve this task, we have to elbow sensing nodes, repulse them from the center of the cloud  $\Psi$ , and force them to move and take the target positions. The complexity of the problem grows in the case of vast clouds and an insufficient number of sensing devices. We developed and validated through simulations two computing schemes, centralized and distributed ones. In the centralized method, one node in a network is nominated for a network head  $D_H \in \mathcal{V}$ . Each device  $D_i, i = 1, \dots, N$  repetitively calculates its target position in  $W$ , solving the optimization problem

$$\min_{\mathbf{x}_i} \left[ V^i = \epsilon_{\Psi}^i V_{\Psi}^i + \sum_{D_j \in N_i, j \neq i} \epsilon_j^i V_j^i = \epsilon_{\Psi}^i \left( \frac{\bar{d}_{\Psi}^i}{d_{\Psi}^i} - 1 \right)^2 + \sum_{D_j \in N_i, j \neq i} \epsilon_j^i \left( \frac{\bar{d}_j^i}{d_j^i} - 1 \right)^2 \right], \quad (10)$$

under the constraint

$$\bar{d}_j^i < r_t. \quad (11)$$

In the above formulas,  $\epsilon_{\Psi}^i \geq 0$  and  $\epsilon_j^i \geq 0$  denote weighting factors determining the importance of the impact of, respectively, the center of the cloud  $\Psi$  and the neighboring node  $D_j$  on the new position of  $D_i$ .  $d_{\Psi}^i$  and  $d_j^i$  are real Euclidean distances between  $\mathbf{x}_i$  and, respectively,  $\Psi$  and  $\mathbf{x}_j$  after a network transformation.  $\bar{d}_{\Psi}^i$  and  $\bar{d}_j^i$  are the reference distances between  $\mathbf{x}_i$  and, respectively,  $\Psi$  and  $\mathbf{x}_j$ . These distances are repetitively calculated by the leader of a network  $D_H$ .  $N_i = \{(D_i, D_n) : \|\mathbf{x}_i - \mathbf{x}_n\| \leq r_t\}, n = 1, \dots, N$  is a set of neighboring nodes of the node  $D_i$  and  $r_t$  is the radio range.

In contrast to the centralized method, a clustering-based technique is proposed. The sensing network  $\mathcal{G}$  (1) is divided into  $M$  separated subnetworks (clusters of devices)  $\mathcal{V}_m, m = 1, \dots, M, \mathcal{V}_1 \cup \mathcal{V}_2 \cup \dots \cup \mathcal{V}_M = \mathcal{V}, \mathcal{V}_1 \cap \mathcal{V}_2 \cap \dots \cap \mathcal{V}_M = \emptyset$ . In each cluster  $\mathcal{V}_m, m = 1, \dots, M$  one node is nominated for a cluster head  $D_{H_m} \in \mathcal{V}_m$ . Moreover, we select one cluster head to be the head of the whole network,  $D_H \in \{D_{H_1}, \dots, D_{H_M}\}$ . All cluster heads are responsible for maintaining permanent connectivity with  $D_H$ . All members of a given



cluster have to maintain permanent connectivity with its cluster head. Hence, each network node repetitively calculates its target position in  $W$ , solving the optimization problem

$$\begin{aligned} \min_{\mathbf{x}_i} & \left[ V^i = \epsilon_{\Psi}^i V_{\Psi}^i + \sum_{D_j \in N_i, D_j \in \mathcal{V}_m} \epsilon_j^i V_j^i + \sum_{k \in IC_m} \epsilon_k^i V_k^i \right. \\ & \left. = \epsilon_{\Psi}^i \left( \frac{\bar{d}_{\Psi}^i}{d_{\Psi}^i} - 1 \right)^2 + \sum_{D_j \in N_i, D_j \in \mathcal{V}_m} \epsilon_j^i \left( \frac{\bar{d}_j^i}{d_j^i} - 1 \right)^2 + \sum_{k \in IC_m} \epsilon_k^i \left( \frac{\bar{d}_k^i}{d_k^i} - 1 \right)^2 \right], \end{aligned} \quad (12)$$

where

$$\bar{d}_k^i = \frac{\sum_{k \in IC_m} d_k^i}{2} + w_2, \quad w_2 > 0. \quad (13)$$

In the above formulas,  $\epsilon_k^i$  denotes a weighting factor determining the importance of the impact of the centroid of a cluster  $\mathcal{V}_k$ ,  $d_k^i$  an actual Euclidean distance between  $\mathbf{x}_i$  and a centroid of  $k$ -th cluster ( $\mathbf{c}_k$ ).  $IC_m$  denotes a set of indexes of two clusters closest to the  $m$ -th cluster that contains  $D_i$ .  $\bar{d}_k^i$  is an average distance between centroids of two clusters with indexes from  $IC_m$  increased by a slight distance margin  $w_2$ .

The calculation scheme and algorithm for selecting clusters for the set  $IC_m$  are described in detail in [11]. Moreover, the authors of [11] present the results of the comparative study of two computing schemes: centralized and distributed. After numerous simulations, some improvements to the cluster-based algorithm that speed up the target positions calculations and do not affect the final result were introduced. Since the task of cluster leaders  $D_{H_m}$ ,  $m = 1, \dots, M$  is limited to maintaining communication, the optimization problem (12) that has to be solved by each  $D_{H_m}$  can be simplified to the following one

$$\begin{aligned} \min_{\mathbf{x}_m} & \left[ V^m = \sum_{D_j \in N_m, D_j \in \mathcal{V}_m} \epsilon_j^m V_j^m + \sum_{k \in IC_m} \epsilon_k^m V_k^m \right. \\ & \left. = \sum_{D_j \in N_m, D_j \in \mathcal{V}_m} \epsilon_j^m \left( \frac{\bar{d}_j^m}{d_j^m} - 1 \right)^2 + \sum_{k \in IC_m} \epsilon_k^m \left( \frac{\bar{d}_k^m}{d_k^m} - 1 \right)^2 \right]. \end{aligned} \quad (14)$$

A sample result of the clustering-based algorithm is depicted in Figure 2b. The paper [29] defines criteria for sensing network topology assessment and describes the algorithm that can be used to temporarily detect optimal topology and complete the second stage of the computing scheme. In general, *stage 2* is completed when devices are almost evenly distributed on the boundary of the cloud, Figure 2c. The set of these devices' locations can be used to discover a given gas cloud boundary's shape.

#### 4.3. Stage 3: Gas Cloud Boundary Tracking

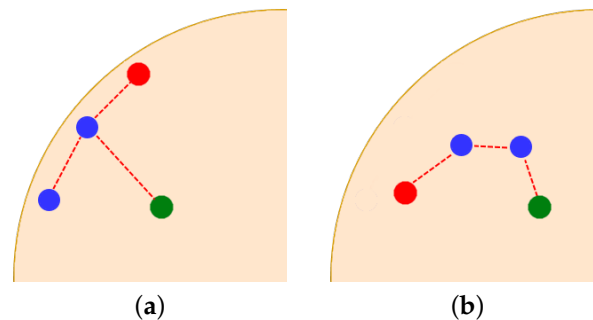
The final aim is to explore and track the gas cloud and report changes to its boundary over time. Let us assume that both nodes and clusters are almost evenly dispersed on the area covered by the phenomena; Figure 2c and Stage 3 starts. Each cluster head  $D_{H_m}$ ,  $m = 1, \dots, M$  selects one node from  $m$ -th cluster members. Let us denote it  $D_{P_m}$ , where  $D_{P_m} \in \mathcal{V}_m$ , and  $m$  is a cluster number. This node aims to move along the boundary over time to discover its current shape, Figure 2d. Other members of cluster  $m$  follow  $D_{P_m}$ ,  $m = 1, \dots, M$ . They serve as a communication link to the network head  $D_H$ , which collects data from all network nodes.

#### 4.3.1. Node $D_{P_m}$ Selection

Let us consider the cluster  $\mathcal{V}_m$ . One can distinguish two situations (Figure 3). Some members of the  $m$ -th cluster are located on a boundary (Figure 3a). The set of nodes located on a boundary at time  $t$  is defined as follows

$$\{D_i \in \mathcal{V}_m : \exists_{t^* \in [t - \Delta t_{out}, t]} g(\mathbf{x}_i, t^*) = 0\}, \quad (15)$$

where  $\Delta t_{out}$  denotes a fixed time period. The node most advanced in a clockwise direction is selected to play the role of  $D_{P_m}$ .



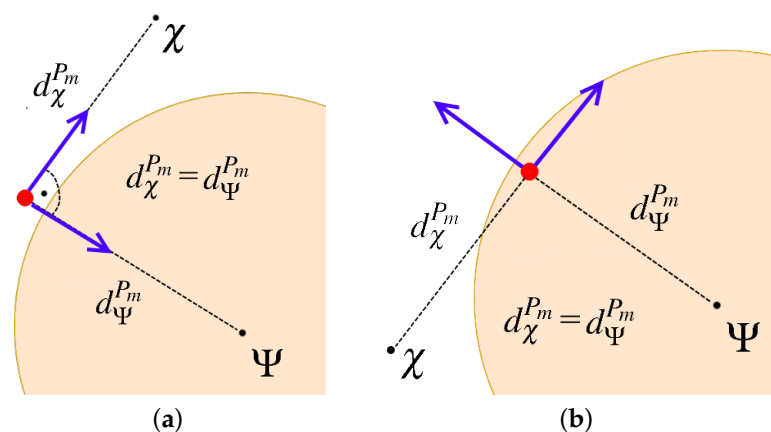
**Figure 3.**  $D_{P_m}$  selection procedure (green node— $D_{H_m}$ , red node— $D_{P_m}$ ): (a)  $D_{P_m}$  selected among devices located on the boundary, (b)  $D_{P_m}$  selected among devices located inside the cloud.

Otherwise, when all cluster  $m$ -th members are located inside the cloud, the node furthest from the cluster head  $D_{H_m}$  is nominated for  $D_{P_m}$  (Figure 3b). The distance between a given node  $D_i \in \mathcal{V}_m$  and  $D_{H_m}$  is calculated in hops, i.e., number of relay nodes.

#### 4.3.2. Motion Trajectory Calculation

Both  $D_{P_m}$ ,  $m = 1, \dots, M$  and other members of clusters repetitively calculate their target positions in  $W$ , solving the optimization problems with performance measures defined respectively to their roles in the sensing network.

Let us start from the leaders of tracking teams  $D_{P_m}$ ,  $m = 1, \dots, M$  displacements calculation. Two points that influence these nodes' motion trajectories can be defined, i.e., centroid of the cloud  $\Psi$  (9) and the point  $\chi \subset W$  that forces the node to follow the cloud boundary. Hence, we obtain two functions  $V_{\Psi}^{P_m}$  and  $V_{\chi}^{P_m}$  that model the interactions between each  $D_{P_m}$  and, respectively,  $\Psi$  and  $\chi$ . The method for determining point  $\chi$  is shown in Figure 4. Two cases are considered, the first one with the current position of  $D_{P_m}$  outside the cloud (Figure 4a) and the second one with  $D_{P_m}$  inside the cloud (Figure 4b).



**Figure 4.** The location of the point  $\chi$ : (a)  $D_{P_m}$  (red dot) outside the gas cloud, (b)  $D_{P_m}$  inside the gas cloud. The artificial potential forces  $V_{\Psi}^{P_m}$  and  $V_{\chi}^{P_m}$ —blue arrows.

Finally, a new position of  $D_{P_m}$  is calculated every  $\Delta t$ , solving the following optimization problem.

$$\begin{aligned} \min_{\mathbf{x}_{P_m}} & \left[ V^{P_m} = \epsilon_{\Psi} V_{\Psi}^{P_m} + \epsilon_{\chi} V_{\chi}^{P_m} \right. \\ & \left. = \epsilon_{\Psi} \left( \frac{\bar{d}_{\Psi}^{P_m}}{d_{\Psi}^{P_m}} - 1 \right)^2 + \epsilon_{\chi} \left( \frac{\bar{d}_{\chi}^{P_m}}{d_{\chi}^{P_m}} - 1 \right)^2 \right], \end{aligned} \quad (16)$$

where  $d_{\Psi}^{P_m}$  and  $d_{\chi}^{P_m}$  denote an actual Euclidean distances between  $D_{P_m}$  and, respectively, the estimated centroid of the cloud  $\Psi$  and point  $\chi$ .  $\bar{d}_{\Psi}^{P_m}$  and  $\bar{d}_{\chi}^{P_m}$  are reference distances between  $D_{P_m}$  and points  $\Psi$  and  $\chi$ . Note that the cloud is in motion. Each node  $D_{P_m}$ ,  $m = 1, \dots, M$  is forced to move inside or outside the cloud depending on the sensor reading. Hence, the motion trajectories oscillate around the cloud boundary. To force oscillations, the reference distances  $\bar{d}_{\Psi}^{P_m}$  in Equation (16) are defined as follows

$$\bar{d}_{\Psi}^{P_m} = \begin{cases} \max_{D_i \in \mathcal{V}^+} d_{\Psi}^i + w_1 & \text{if } g(\mathbf{x}_{P_m}, t) = 1, \\ \min_{D_i \in \mathcal{V}^+} d_{\Psi}^i & \text{if } g(\mathbf{x}_{P_m}, t) = 0, \end{cases} \quad (17)$$

where  $\mathcal{V}^+$  denotes the set of devices located inside the cloud. Other members of the clusters  $D_i \in \mathcal{V}_m$ ,  $m = 1, \dots, M$  (including  $D_{H_m}$ ) aim to maintain connectivity between nodes  $D_{P_m}$  and  $D_H$ . They are relay nodes in communication between  $D_{P_m}$  and  $D_H$ . It is evident that each node  $D_i \in \mathcal{V}_m$ ,  $D_i \neq D_{P_m}$  is forced to follow  $D_{P_m}$ . Hence, its new position is calculated every  $\Delta t$ , solving the optimization problem.

$$\min_{\mathbf{x}_i} \left[ V^i = V_{P_m}^i = \left( \frac{\bar{d}_{P_m}^i}{d_{P_m}^i} - 1 \right)^2 \right], \quad \bar{d}_{P_m}^i \leq r_t, \quad (18)$$

where  $d_{P_m}^i$  denotes an Euclidean distance between  $D_i \in \mathcal{V}_m$  and  $D_{P_m}$ ,  $\bar{d}_{P_m}^i$  is a reference distance between  $D_i$  and  $D_{P_m}$ .  $r_t$  is the radio transmission range.

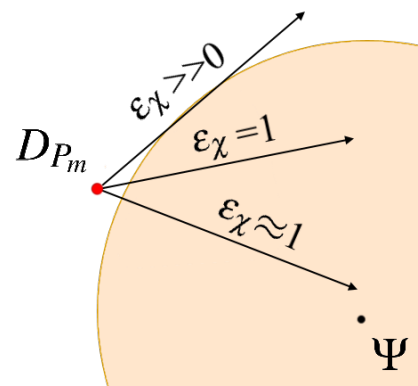
#### 4.3.3. Weighting Factor Calculation

Several parameters have to be identified in the optimization task (16). To simplify the model, we assumed without loss of generality  $\bar{d}^{P_m} = \bar{d}_{\chi}^{P_m} = \bar{d}_{\Psi}^{P_m}$  and  $\epsilon_{\Psi} = 1$ . Hence, the optimization problem (16) is reduced to the following one

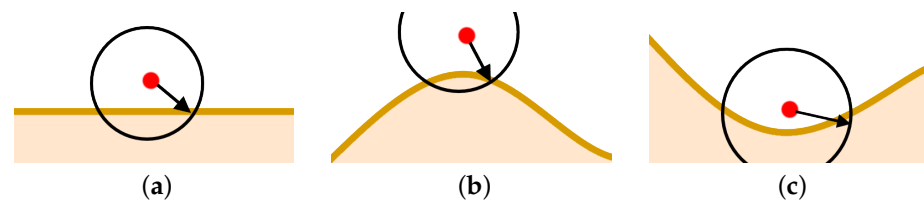
$$\min_{\mathbf{x}_{P_m}} \left[ V^{P_m} = V_{\Psi}^{P_m} + \epsilon_{\chi} V_{\chi}^{P_m} = \left( \frac{\bar{d}^{P_m}}{d_{\Psi}^{P_m}} - 1 \right)^2 + \epsilon_{\chi} \left( \frac{\bar{d}^{P_m}}{d_{\chi}^{P_m}} - 1 \right)^2 \right], \quad (19)$$

and only coefficient  $\epsilon_{\chi}$  must be tuned. The impact of  $\epsilon_{\chi}$  on the recommended motion direction of the node  $D_{P_m}$  is depicted in Figure 5.

The recommended motion direction should be fit to the shape of a given boundary. Figure 6 presents the examples of various shapes of the cloud boundary and adequate directions of  $D_{P_m}$  displacements.



**Figure 5.** The impact of the coefficient  $\epsilon_\chi$  on the motion direction of the node  $D_{P_m}$ .



**Figure 6.** Recommended directions of  $D_{P_m}$  displacements; various shapes of boundaries (a) flat; (b) convex; (c) concave.

Assume  $\tau_z, z = 1, 2, \dots$ , denote time stamps of  $D_{P_m}$  detector state changes ( $g(\mathbf{x}_{P_m}, \tau_z) \neq g(\mathbf{x}_{P_m}, \tau_z - \Delta t)$ ) and  $\tau_{z+1} - \tau_z = k_z \cdot \Delta t$ , where  $k_z$  is a number of sensor readings taken since the last change of the detector state. Five methods for the weighting parameter  $\epsilon_\chi$  in Equation (19) updating every  $\Delta t$ , i.e., at time steps  $t = \tau_z + \Delta t, \tau_z + 2\Delta t, \dots, \tau_z + (k_z - 1)\Delta t$ ,  $z = 1, 2, \dots$ , were proposed and tested. The input data contains a list of historical values of  $\epsilon_\chi$  and a list of historical bivalent sensor readings. The following items present the algorithms for adjusting the factor in the periods between subsequent sensor readings changes, i.e.,  $\tau_{z+1} - \tau_z, z = 1, 2, \dots$ .

**Variant I** A constant value of the weighting factor in the whole tracking horizon.  $\epsilon_\chi^{(k)} = \bar{\epsilon}_\chi$ ,  $k = 0, 1, \dots, k_z$  where  $\bar{\epsilon}_\chi$  is the predefined value, the same one regardless of the sensor state.

**Variant II** Decreasing the value of the weighting factor over time (the closer the device to the cloud boundary, the slower it moves). The new value of  $\epsilon_\chi$  is calculated based on the previous one, starting from the predefined value)

$$\epsilon_\chi^{(0)} = \bar{\epsilon}_\chi, \quad \epsilon_\chi^{(k)} = l \cdot \epsilon_\chi^{(k-1)}, \quad k = 1, 2, \dots, k_z, \quad (20)$$

where  $l \in (0, 1)$ .

**Variant III** The initial value of  $\epsilon_\chi$  for time stamps  $\tau_z, z \geq 3$  is calculated as a linear combination of a predefined constant value and value of  $\epsilon_\chi$  at the time stamp of the same change of sensor state in the past.

$$\epsilon_\chi^{(0)} = \begin{cases} \bar{\epsilon}_\chi, & z < 3, \\ \alpha \cdot \epsilon_\chi^{z-2} + (1 - \alpha) \cdot \bar{\epsilon}_\chi, & z \geq 3, \end{cases} \quad (21)$$

where  $\alpha \in (0, 1)$  is an experimentally adjusted model parameter and  $\bar{\epsilon}_\chi$  a predefined constant value.  $\epsilon_\chi^{z-2}$  is a value of the factor calculated at time stamp  $\tau_{z-1} - \Delta t$ . The scaling factor is decreased according to the Equation (20).

**Variant IV**  $\epsilon_\chi^{(0)} = \bar{\epsilon}_\chi$  at time stamps  $\tau_z, z < 3$ . At time stamps  $\tau_z, z \geq 3$  the initial value of  $\epsilon_\chi$  is calculated according to the following formula

$$\epsilon_\chi^{(0)} = \begin{cases} \bar{\epsilon}_\chi^{z-2}, & k_{z-2} > 1, \\ \bar{\epsilon}_\chi^{z-2} + \bar{b}, & k_{z-2} = 1, \end{cases} \quad (22)$$

where  $\tau_{z-1} - \tau_{z-2} = k_{z-2} \cdot \Delta t$ ,  $\bar{b} > 0$  is a constant value used to increase the speed of node  $D_{P_m}$ . The scaling factor is decreased according to the Equation (20).

**Variant V** Modification of variant IV by replacing the fixed  $\bar{b}$  with the variable  $b$ . This variable is updated every  $\tau_z, z = 1, 2, \dots$ . It is calculated as follows

$$b^z = \begin{cases} \bar{b}, & k_{z-2} > 1, \\ b^{z-2} + \Delta b, & k_{z-2} = 1, \end{cases} \quad (23)$$

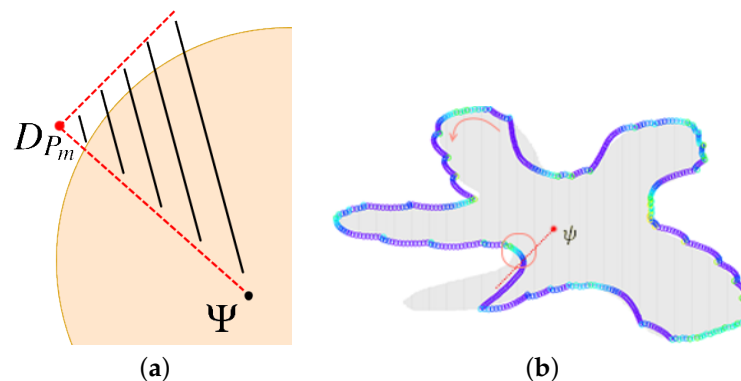
where  $\Delta b > 0$  is a fixed value and  $b^{z-2}$  a value of the variable  $b$  calculated at  $\tau_{z-1} - \Delta t$ . Finally, the initial value of the factor  $\epsilon_\chi$  at time stamps  $\tau_z, z \geq 3$  is calculated according to the following formula

$$\epsilon_\chi^{(0)} = \begin{cases} \bar{\epsilon}_\chi^{z-2}, & k_{z-2} > 1, \\ \bar{\epsilon}_\chi^{z-2} + b^z, & k_{z-2} = 1. \end{cases} \quad (24)$$

Next, the scaling factor is decreased according to the Equation (20).

#### 4.4. Nonconvex Boundary Tracking

The problem arises in the case of tracking the cloud boundary with concave sections. The possible target positions are limited to the quarter determined by the range of motion of  $D_{P_m}$ , Figure 7a. Therefore, it may take a long time to cover the border accurately, even if the value of factor  $\epsilon_\chi$  is close to zero, (Figure 7b).



**Figure 7.** Tracking concave sections of the cloud boundary: (a) possible locations of target points, (b) sample motion trajectory of  $D_{P_m}$  tracking a nonconvex cloud boundary.

Hence, to improve the quality and speed up the concave sections tracking, the following correction procedure is proposed. In the case of  $\epsilon_\chi < \epsilon_{min}$ , where  $\epsilon_{min}$  is a predefined threshold value, the point  $\chi$  is relocated. Its position is replaced by a new one calculated by rotating the original one by the angle  $r\frac{\pi}{2}$ ,  $r \in \{-3, -2, -1, 1, 2, 3\}$  in a clockwise direction relative to the position of  $D_{P_m}$ . Then, the optimization problem (19) is solved, and new positions of leaders of tracking teams  $D_{P_m}, m = 1, \dots, M$  are calculated. The parameter  $r$  is calculated according to Algorithm 1.

**Algorithm 1** Detection of concave parts of boundary and  $r$  calculation

---

```

1:  $r = 0$ 
2: if  $\epsilon_\chi < \epsilon_{min}$  and  $k_z > 1$  then
3:   if  $g(\mathbf{x}_{P_m}, t) = 1$  then
4:      $r = r - 1$ 
5:   else
6:      $r = r + 1$ 
7:   end if
8: else if  $g(\mathbf{x}_{P_m}, t) \neq g(\mathbf{x}_{P_m}, t - \Delta t)$  and  $k_{z-1} = 1$  then
9:   if  $r < 0$  then
10:     $r = r + 1$ 
11:   end if
12:   if  $r > 0$  then
13:     $r = r - 1$ 
14:   end if
15: end if

```

---

**5. Experimental Study**

The computing scheme for detecting and tracking heavy gas clouds was tested and evaluated through simulation. Many of the experiments were designed and conducted to tune the parameters of the developed algorithms for sensing devices' motion planning and evaluate the efficiency of developed sensing networks. The sensing networks comprising intelligent wireless mobile devices were implemented in the MobASim simulation platform [30]. MobASim is a software environment for prototyping and simulation of wireless ad hoc networks in two-dimensional space. For the experiments, a simple box model of an instantaneous heavy gas cloud [31] that hazard analysts widely adopt was implemented. It provides a convenient and efficient method for atmospheric dispersion modeling.

**5.1. Quality and Performance Metrics**

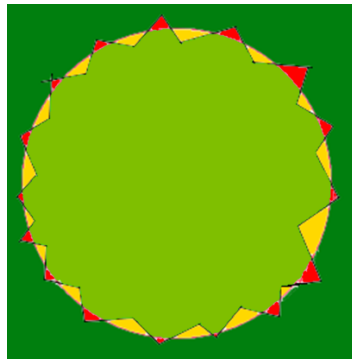
Three metrics to measure the quality and efficiency of developed algorithms were defined, i.e., (i) the accuracy of the boundary estimation, (ii) the time used to discover the boundary, (iii) frequency of boundary-crossing.

Compact sensors' coverage of the boundary of the cloud is the most crucial requirement. To evaluate this requirement, the widely used accuracy measure  $acc$  was proposed

$$acc = \frac{TP + TN}{TP + TN + FN + FP}. \quad (25)$$

Let us assume that  $t_{circle}$  denotes a lap time of the gas cloud by the sensing device  $D_{P_m}$ . Subsequent target positions of  $D_{P_m}$  are calculated every  $\Delta t$ . Hence  $t_{circle} = L \cdot \Delta t$ , where  $L$  is the number of calculated positions. Let us use linear interpolation. The input data set consists of  $L$  target positions of  $D_{P_m}$ . Finally, we can approximate the cloud boundary and use it to estimate the area covered by the cloud. Figure 8 shows the result for the circular gas cloud covering the part of the given working space. Then, we can calculate the values of  $TP$  (true positive),  $TN$  (true negative),  $FN$  (false negative) and  $FP$  (false positive) in Equation (25).  $TP$  and  $TN$  denote the correctly recognized area by our sensing devices, respectively, inside and outside the cloud, while  $FN$  and  $FP$  denote incorrectly recognized areas. It is evident that the greater the  $acc$  measure, the better the approximation of the boundary.





**Figure 8.** The cloud boundary approximation: TP (light green), TN (bottle green), FN (yellow) and FP (red).

In general, it is desirable for the sensing device to cross the boundary as frequently as possible. It means that the devices are still within the narrow margin of the boundary. Hence, the corresponding measure  $f_{crossing}$  can be defined

$$f_{crossing} = \frac{\sum_{t=1}^{t_{circle}-1} h}{t_{circle} - 1}, \quad h = \begin{cases} 0 & g(\mathbf{x}_{P_m}, t) = g(\mathbf{x}_{P_m}, t + \Delta t) \\ 1 & g(\mathbf{x}_{P_m}, t) \neq g(\mathbf{x}_{P_m}, t + \Delta t) \end{cases} \quad (26)$$

The linear interpolation better approximates the boundary of the cloud when there is a higher  $f_{crossing}$ . Finally, a cloud boundary exploration time  $t_{circle}$  should be as short as possible.

An indicator  $q$  that aggregates all described metrics has been introduced to simplify the sensing network performance evaluation

$$q = \frac{(f_{crossing})^{w_f} \cdot acc^{w_{acc}}}{(t_{circle})^{w_t}}, \quad (27)$$

where  $w_t > 0$ ,  $w_f > 0$  and  $w_{acc} > 0$  are parameters that indicate the importance of given measures.

## 5.2. Scenario Description

The testing scenario was a synthetic network comprised of 16 sensing devices designed to detect and track the chlorine cloud resulting from an immediate leak. All devices were equipped with radio transceivers with a maximum transmit power equal to  $-10$  dBm (transmission range  $r_t = 162.86$  m), sensors measuring gas concentration every  $\Delta t = 1$  s and moving platforms with a maximum speed 15 ms. In all experiments, the network was divided into four autonomous clusters. In general, the subnetworks tracked the cloud independently, and the only binding element was the center of cloud  $\Psi$ . The following values of parameters in indicator  $q$  (27) were fixed:  $w_t = 3$ ,  $w_f = 1$  and  $w_{acc} = 2$ .

Five simulation scenarios were tested: *Cloud 1*, a gas cloud with a circle-shaped border, *Cloud 2*, a gas cloud with an ellipse-shaped border, *Cloud 3*, a gas cloud with a mixed border, partially circle-shaped and partially ellipse-shaped, and two gas clouds, i.e., *Cloud 4* and *Cloud 5*, with nonconvex borders. The average values of performance metrics  $acc$ ,  $f_{crossing}$ ,  $t_{circle}$  and  $q$  calculated by leaders of all clusters are presented in the tables. All figures illustrate motion trajectories of device  $D_{P_m}$  from the same selected cluster. Consecutive points show successive  $D_{P_m}$  positions in the workspace ( $800 \times 800$  m). At this localization,  $D_{P_m}$  measured the gas concentration. Colors indicate the values of  $\epsilon_\chi$  used to calculate these points. Various shades of five colors: violet ( $\epsilon_\chi \in [0.1, 2.5)$ ), blue ( $\epsilon_\chi \in [2.5, 5.0)$ ), green ( $\epsilon_\chi \in [5.0, 7.5)$ ), yellow ( $\epsilon_\chi \in [7.5, 10)$ ), red ( $\epsilon_\chi \geq 10$ ) are used.

### 5.3. Model Parameters Tuning—Circle and Ellipse Shaped Clouds

The first set of experiments were designed to examine and compare the performance and efficiency of the developed sensing network for various values of parameters  $\bar{\epsilon}_\chi$ ,  $\bar{b}$ ,  $\Delta b$ ,  $l$  and  $\alpha$  that have to be arbitrarily selected by the user and the weighting factor  $\epsilon_\chi$  in Equation (19) calculated according to I-V variants described in Section 4.3.3. The aim was to test the sensitivity of the developed algorithm for boundary tracking of these parameters and to determine the values of all mentioned parameters and a variant of the  $\epsilon_\chi$  updating procedure that guarantees high-quality boundary estimation. The results of simulations conducted for two different shapes of gas clouds, namely scenarios *Cloud 1* and *Cloud 2*, are presented in Tables 1–4 and Figures 9 and 10. Tables 1 and 3 collect values of model parameters for which the highest value of the indicator  $q$  calculated for each variant of  $\epsilon_\chi$  updating was obtained in both scenarios. Tables 2 and 4 show the values of all metrics described in Section 5.1. The best ones are marked in green. Figures 9 and 10 depict the motion trajectories (positions in time) of the selected device  $D_{P_m}$  calculated for various variants of the  $\epsilon_\chi$  updating procedure and circle-shaped and ellipse-shaped clouds.

**Table 1.** Values of parameters used to calculate the optimal weighting factor  $\epsilon_\chi$ ; indicator  $q$ , simulation scenario *Cloud 1*.

	Variant I	Variant II	Variant III	Variant IV	Variant V
$\bar{\epsilon}_\chi$	2	5	10	2	2
$l$	-	0.8	0.8	0.8	0.8
$\alpha$	-	-	0.9	-	-
$\bar{b}$	-	-	-	0.5	1
$\Delta b$	-	-	-	-	1.3

**Table 2.** Values of performance metrics obtained for five variants of  $\epsilon_\chi$  updating; simulation scenario *Cloud 1*.

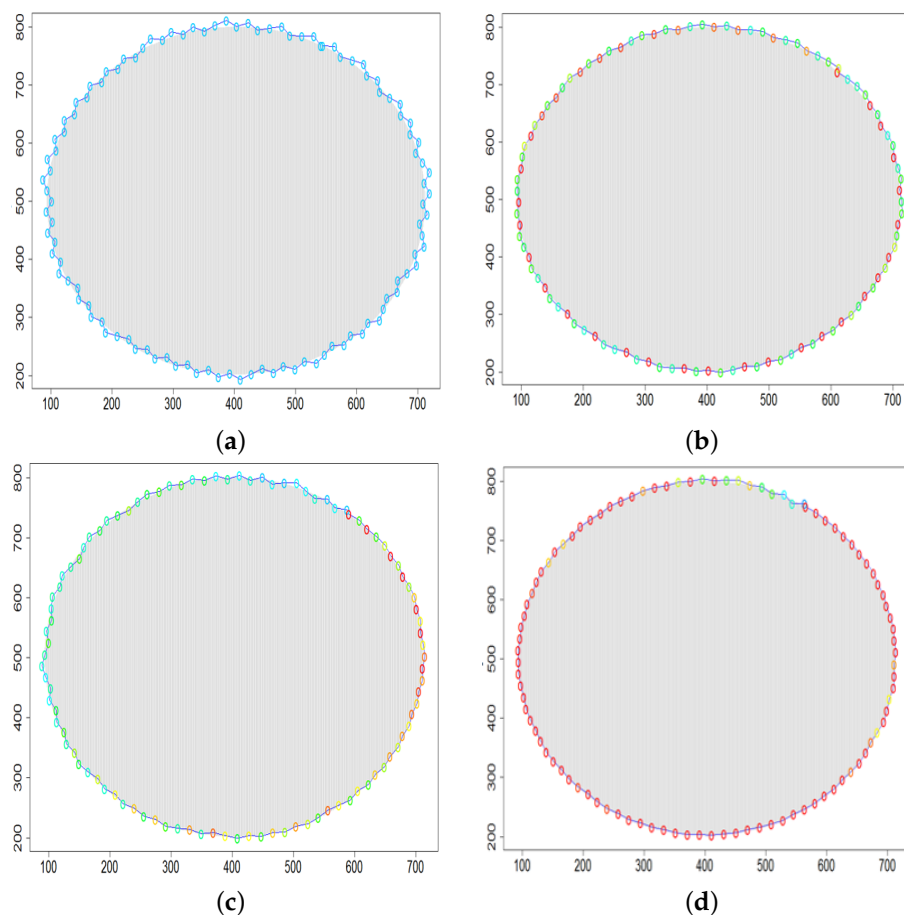
	Variant I	Variant II	Variant III	Variant IV	Variant V
$t_{circle}$	105.5	98.75	98.25	99.75	98.25
$f_{crossing}$	0.61	0.7	0.65	0.82	0.73
$acc$	0.977	0.988	0.988	0.99	0.991
$q$	4.9	7.0	6.7	8.1	7.6

**Table 3.** Values of parameters used to calculate the optimal weighting factor  $\epsilon_\chi$ ; indicator  $q$ , simulation scenario *Cloud 2*.

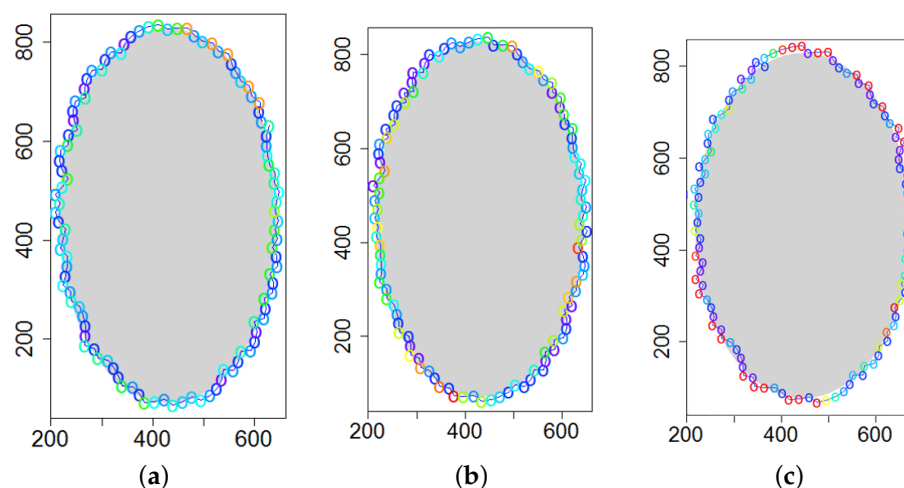
	Variant I	Variant II	Variant III	Variant IV	Variant V
$\bar{\epsilon}_\chi$	1	2	8	5	2
$l$	-	0.6	0.6	0.6	0.5
$\alpha$	-	-	0.9	-	-
$\bar{b}$	-	-	-	0.9	1
$\Delta b$	-	-	-	-	1.3

**Table 4.** Values of performance metrics obtained for five variants of  $\epsilon_\chi$  updating; simulation scenario *Cloud 2*.

	Variant I	Variant II	Variant III	Variant IV	Variant V
$t_{circle}$	107	111.5	106.5	108.75	108.5
$f_{crossing}$	0.45	0.56	0.57	0.65	0.64
$acc$	0.962	0.965	0.972	0.975	0.973
$q$	3.5	3.8	4.4	4.8	4.7



**Figure 9.** Successive  $D_{P_m}$  positions during cloud boundary tracking calculated for four variants of  $\epsilon_\chi$  updating procedures: (a) variant I, (b) variant III, (c) variant IV, (d) variant V; simulation scenario *Cloud 1*. The colors of the points correspond to different values of  $\epsilon_\chi$ .



**Figure 10.** Successive  $D_{P_m}$  positions during cloud boundary tracking calculated for three variants of  $\epsilon_\chi$  updating procedures: (a) variant III, (b) variant IV, (c) variant V; simulation scenario *Cloud 2*. The colors of the points correspond to different values of  $\epsilon_\chi$ .

It can be seen that, in general, the best values of all metrics were obtained for two variants of the weighting factor  $\epsilon_\chi$  updating, namely IV and V for both simulation scenarios. However, variant IV was the best with respect to the indicator  $q$  aggregating all metrics, although the time used to circle the cloud was slightly longer than in the case of variant

III. The optimal values of parameters  $\bar{\epsilon}_\chi$  are lower for ellipse-shaped cloud boundary. A higher value of  $\bar{\epsilon}_\chi$  increases the time for the device to return to the cloud region when the radius of the curvature of the cloud decreases rapidly. Moreover, it can be observed that each device needs more time to circle the ellipse-shaped cloud than the circle-shaped one. In general, tracking an ellipse-shaped cloud is a more difficult task.

#### 5.4. Model Parameters Adjusting—Convex Clouds

The second set of experiments was designed to check how the model parameters tuned for networks tracking gas clouds with circle and ellipse-shaped boundaries can be adjusted to networks tracking gas clouds with a mixed border, partially circle shaped and partially ellipse-shaped (scenario *Cloud 3*). Two series of experiments were conducted. The aim was to compare the quality of boundary tracking in two cases (i) the weighting factor  $\epsilon_\chi$  in Equation (19) calculated for values of parameters collected in Table 1, (ii)  $\epsilon_\chi$  calculated for values of parameters collected in Table 3. The results are presented in Tables 5 and 6.

**Table 5.** Values of performance metrics obtained for five variants of  $\epsilon_\chi$  updating and values of parameters from Tables 1 and 3; simulation scenario *Cloud 3*.

	Variant I		Variant II		Variant III		Variant IV		Variant V	
	Table 1	Table 3	Table 1	Table 3	Table 1	Table 3	Table 1	Table 3	Table 1	Table 3
$t_{circle}$	116.5	149	112	125	112	116.5	116.5	117	113.75	122.5
$f_{crossing}$	0.52	0.81	0.37	0.73	0.54	0.69	0.68	0.72	0.52	0.69
$acc$	0.976	0.979	0.963	0.982	0.983	0.986	0.985	0.987	0.971	0.984
$q$	3.2	2.4	2.5	3.6	3.7	4.2	4.2	4.4	3.4	3.6

**Table 6.** Successive  $D_{P_m}$  positions during cloud boundary tracking calculated for two variants of  $\epsilon_\chi$  updating procedures and two sets of parameters from Tables 1 and 3; simulation scenario *Cloud 3*.

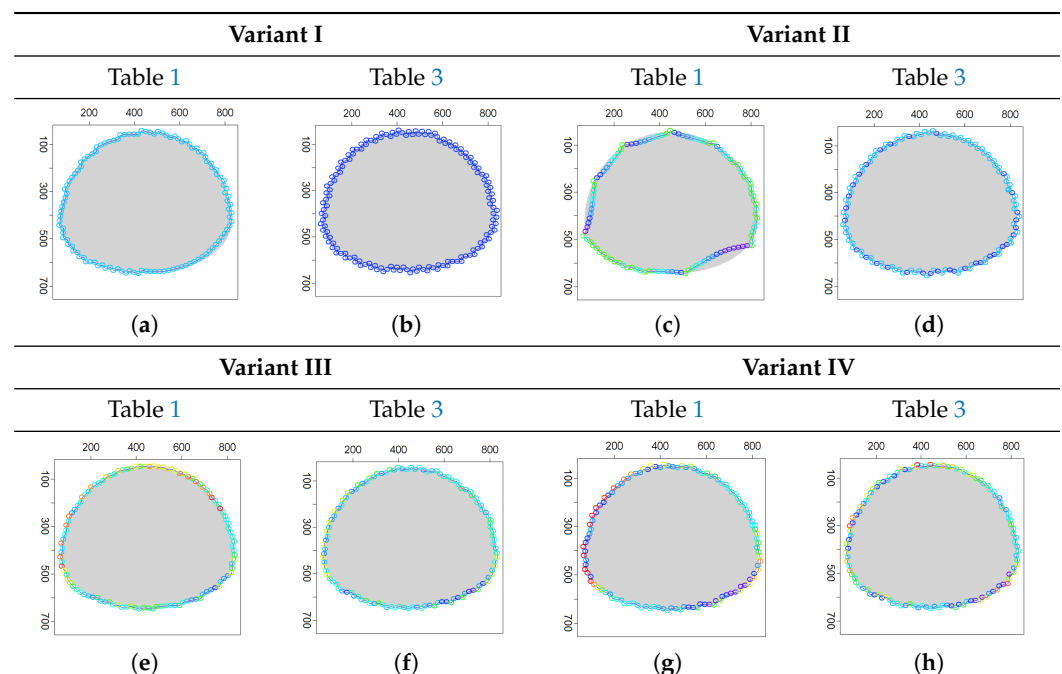
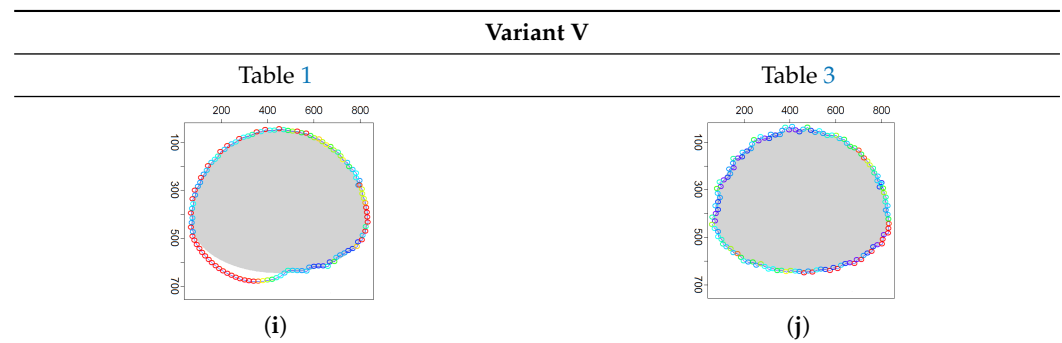


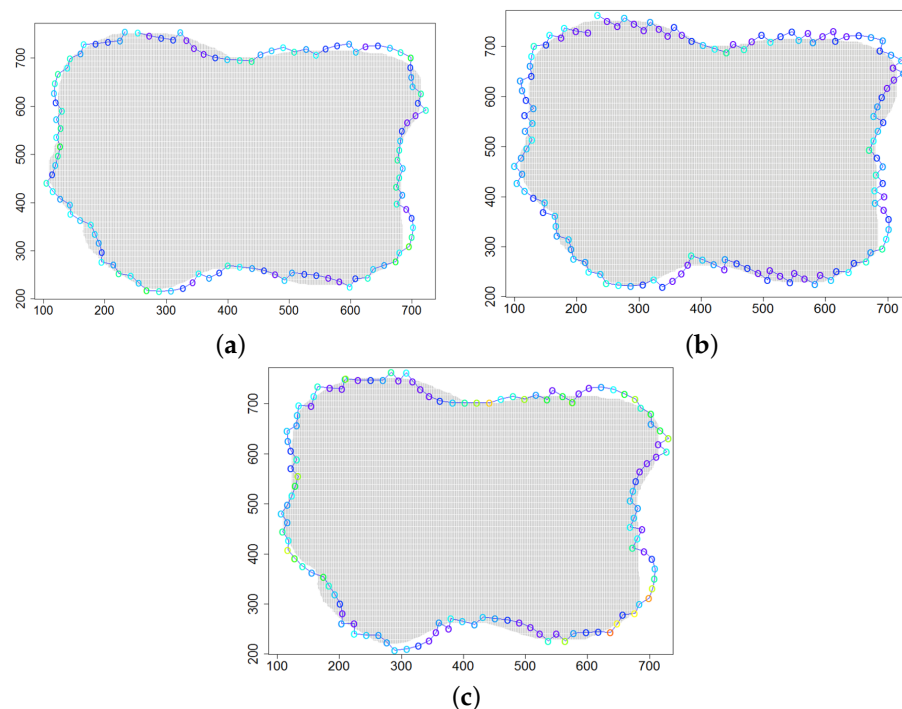
Table 6. Cont.



As can be seen, all results of boundary tracking are, in general, acceptable. However, better accuracy of the boundary estimation was obtained using values of parameters tuned for the ellipse-shaped cloud. It is clearly illustrated in Table 6. Similar to previous experiments described in Section 5.3, the highest value of the indicator  $q$  was obtained for variant IV of the  $\epsilon_\chi$  updating procedure. The time taken to circle the gas cloud was the shortest in the case of variant III. However, it should be noted that much worse results in comparison to those presented in Section 5.3 were achieved for variant V. It can be concluded that the presented approach is sensitive to model parameters. Hence, the number of parameters should not be too large as it is difficult to identify them for various shapes of clouds. Variant V can be successfully applied in tracking clouds with a big radius of curvature.

#### 5.5. Model Parameters Adjusting—Nonconvex Clouds

The experiments discussed in the two previous sections were conducted for convex clouds. Let us consider a nonconvex cloud with an irregular, nonconvex boundary, as presented in Figure 11, *Cloud 4*.



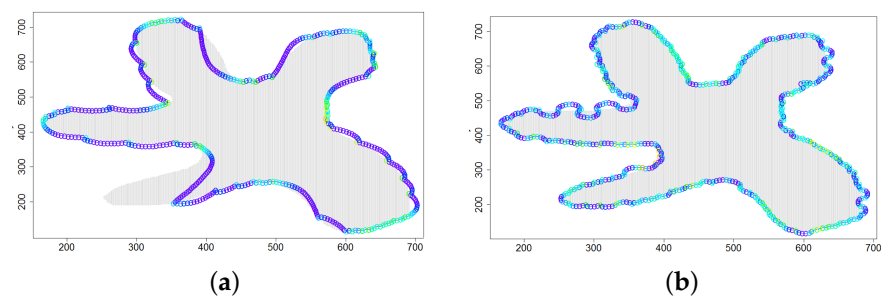
**Figure 11.** Successive  $D_{p_m}$  positions during cloud boundary tracking calculated for three variants of  $\epsilon_\chi$  updating procedures: (a) variant III, (b) variant IV, (c) variant V; simulation scenario *Cloud 4*. The colors of the points correspond to different values of  $\epsilon_\chi$ .

A series of simulations were conducted for three variants of the  $\epsilon_\chi$  updating procedure and parameters from Table 1. Table 7 shows the values of metrics obtained for three variants of  $\epsilon_\chi$  updating. Worse results were obtained than in scenarios with convex boundaries. It can be concluded that variant V of the  $\epsilon_\chi$  updating procedure is suitable only when the boundary has a shape with a constant radius of curvature. In the case of two other variants, the results are very similar, and the decision to choose one of them requires prioritizing the criteria. When fast detection of the boundary is crucial, variant III is recommended. In addition, simulations were performed in which  $\epsilon_\chi$  was calculated for parameters from Table 3. The results were a bit better, i.e.,  $acc = 0.962$  and  $q = 3.2$ .

**Table 7.** Values of performance metrics obtained for three variants of  $\epsilon_\chi$  updating; simulation scenario *Cloud 4*.

	Variant III	Variant IV	Variant V
$t_{circle}$	112	124	119.75
$f_{crossing}$	0.47	0.96	0.49
$acc$	0.956	0.991	0.949
$q$	3	2.9	2.6

The last series of experiments aimed to test the efficiency of the correction procedure described in Section 4.4. The results of simulations for the application scenario *Cloud 5*, i.e., a cloud with multiple concave parts of its boundary, are presented in Table 8 and Figure 12. The weighting factor  $\epsilon_\chi$  was calculated according to Equation (22) (variant IV) for values of parameters from Table 1. The threshold value  $\epsilon_{min}$  was fixed to 0.3. The motion trajectory of the device  $D_{P_m}$  presented in Figure 12b can be compared to the one calculated when the correction for concave parts of boundaries is disabled (Figure 12a). It can be seen that the correction procedure significantly improves the accuracy of cloud boundary estimation for very irregularly shaped clouds.



**Figure 12.** Successive  $D_{P_m}$  positions during cloud boundary tracking calculated for two variants: (a) correction procedure disabled, (b) correction procedure enabled; simulation scenario *Cloud 5*. The colors of the points correspond to different values of  $\epsilon_\chi$ .

**Table 8.** Values of performance metrics obtained for variant IV, parameters from Table 1, disabled and enabled correction procedure.

	Disabled	Enabled
$t_{circle}$	343.5	360.25
$f_{crossing}$	0.51	0.6
$acc$	0.987	0.99
$q$	$6.9 \times 10^{-10}$	$8 \times 10^{-10}$

## 6. Conclusions

The paper summarizes the research results concerned with designing and developing distributed sensing systems for heavy gas cloud monitoring. This system comprises autonomous unmanned vehicles, equipped with punctual sensors, radio transceivers and GPS modules that spontaneously create a network of devices that adapt to achieve



goals. The paper describes a three-stage strategy for boundary detection, estimation and tracking. In this approach, the optimal motion trajectories for all sensing devices' discovering the current shape of the boundary are calculated based on the data collected from sensing vehicles and the models for target position calculation that incorporates artificial potential functions. The effectiveness was tested through extensive simulations. The presented results of experiments show that the design of an efficient sensing system for phenomena clouds should account for trade-offs between detection accuracy and computational complexity, measurement quality and equipment cost, communication reliability and network load. The challenge for the designer of this type of network is to develop efficient algorithms for online calculating of motion trajectories of sensing nodes and determine the optimal values of model parameters. This paper presents such an algorithm and examines the quality of various computing schemes for its parameter tuning. Simulation experiments showed the sensitivity of the proposed system to the model parameters and the need to select a procedure for tuning the coefficients of the optimization task to a particular scenario. Nevertheless, even for nonconvex clouds with irregular shapes, satisfactory results were obtained for calculated values of model parameters. Moreover, the simulation results demonstrated that using only a rough measurement to indicate that the threshold concentration value was exceeded can detect and track a gas cloud boundary. Calculating new sensor positions only based on the information about exceeding the safe gas concentration value makes our sensing system less sensitive to the quality of the gas concentration measurement. Because of the inaccuracy of actual sensors, such an approach may be better suited to real-world applications. Moreover, it can be easily adapted to detect other phenomena clouds, namely oil spill and wildfires, etc. To sum up, the simulation results corroborate our analysis and confirm that mobile ad hoc networks can be successfully used to monitor dynamic phenomena, create situation awareness and support rescue teams.

The research in that domain will be continued. The future work will focus on the experiments conducted on more complex and realistic scenarios with more detailed toxic gas dispersion models and workspace with obstacles. The goal is to test the sensitivity of our system on the speed of boundary evolution and the impact of the obstacles on the quality of gas concentration measurement, radio communication and, consequently, the effectiveness of our monitoring system. Moreover, we plan to develop and investigate a version of the system to operate in a three-dimensional workspace. It will allow extending the possibilities of applications to monitor other phenomena such as oil spills, wildfire, moving groups of people.

**Author Contributions:** Conceptualization, M.K. and E.N.-S.; methodology, M.K. and E.N.-S.; software, M.K.; validation, M.K.; formal analysis, M.K. and E.N.-S.; investigation, M.K.; resources, M.K. and E.N.-S.; data curation, M.K.; writing—original draft preparation, M.K. and E.N.-S.; writing—review and editing, M.K. and E.N.-S.; visualization, M.K.; supervision, E.N.-S.; project administration, E.N.-S.; funding acquisition, M.K. All authors have read and agreed to the published version of the manuscript.

**Funding:** The work of the authors of this paper was partially supported by the European Commission, under grant agreement no. 833456 (GUARD).

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** Not applicable.

**Conflicts of Interest:** The authors declare no conflict of interest.

## Appendix A. Summary of Standard Notation

$\mathcal{G} = (\mathcal{V}, \mathcal{E})$	a sensing network; $\mathcal{V}$ —a set of sensing devices, $\mathcal{E}$ a set of active direct connections between each pair of devices
$D_i, \mathbf{x}_i, v_i$	$i$ -th sensing device, its position in the workspace $W$ and speed
$N_i$	a set of nodes in the radio range $r_t$ of the node $D_i$
$d_a^i, \bar{d}_a^i$	an actual Euclidean distance and a reference distance between $D_i$ and $a$
$\epsilon_b^i$	a weighting factor determining the importance of the impact of $b$ on the new position of $D_i$
$V_a^b$	a virtual force that models the interactions between two points in the workspace $W$
$g(\mathbf{x}, t)$	a gas concentration measured at point $\mathbf{x}$ and time $t$
$D_{leader}$	a formation leader in the gas cloud detection stage
$D_H$	a network head in the gas cloud boundary detection stage
$\mathcal{V}_m, D_{H_m}$	$m$ -th cluster and its cluster head
$D_{P_m}$	a leader of tracking team in $m$ -th cluster (cloud boundary tracking stage)
$\Psi, \epsilon_\Psi$	an estimated gas cloud center and its weighting factor
$\chi, \epsilon_\chi$	a point forcing a device to follow the cloud boundary and its weighting factor
$t_{circle}, f_{crossing}, acc, q$	quality metrics: a cloud boundary exploration time, a number of boundary crossing, an accuracy of boundary detection, an aggregated measure

## References

- Wenck, M.; Van Sickle, D.; Drociuk, D.; Belflower, A.; Youngblood, C.; Whisnant, M.; Taylor, R.; Rudnick, V.; Gibson, J. Rapid assessment of exposure to chlorine released from a train derailment and resulting health impact. *Public Health Rep.* **2007**, *122*, 784–792. [[CrossRef](#)] [[PubMed](#)]
- Shu, L.; Mukherjee, M.; Wu, X. Toxic gas boundary area detection in large-scale petrochemical plants with industrial wireless sensor networks. *IEEE Commun. Mag.* **2016**, *54*, 22–28. [[CrossRef](#)]
- Michini, M.; Hsieh, M.A.; Forgoston, E.; Schwartz, I.B. Robotic Tracking of Coherent Structures in Flows. *IEEE Trans. Robot.* **2014**, *30*, 593–603. [[CrossRef](#)]
- Ammad-uddin, M.; Ayaz, M.; Aggoune, E.H.; Sajjad, M. Wireless sensor network: A complete solution for poultry farming. In Proceedings of the 2014 IEEE 2nd International Symposium on Telecommunication Technologies (ISTT), Langkawi, Malaysia, 24–26 November 2014; pp. 321–325; [[CrossRef](#)]
- Britter, R.E.; McQuaid, J. *Workbook on the Dispersion of Dense Gases*; Health and Safety Executive: Bootle, UK, 1988.
- Rappaport, T. *Wireless Communications: Principles and Practice*, 2nd ed.; Prentice Hall PTR: Upper Saddle River, NJ, USA, 2001.
- Roy, R. *Handbook of Mobile Ad Hoc Networks for Mobility Models*; Springer: Berlin/Heidelberg, Germany, 2010.
- Abbasi, A.A.; Younis, M. A survey on clustering algorithms for wireless sensor networks. *Comput. Commun.* **2007**, *30*, 2826–2841. [[CrossRef](#)]
- Thai, M.T.; Tiwari, R.; Bose, R.; Helal, A. On Detection and Tracking of Variant Phenomena Clouds. *Acm Trans. Sens. Netw.* **2014**, *10*, 1–33. [[CrossRef](#)]
- Markiewicz, M. Mathematical modeling of heavy gas atmospheric dispersion over complex and obstructed terrain. *Arch. Environ. Prot.* **2010**, *36*, 81–94.
- Krzyszton, M.; Niewiadomska-Szynkiewicz, E. Heavy Gas Cloud Boundary Estimation and Tracking using Mobile Sensors. *J. Telecommun. Inf. Technol.* **2016**, *3*, 38.
- Shu, L.; Mukherjee, M.; Xu, X.; Wang, K.; Wu, X. A Survey on Gas Leakage Source Detection and Boundary Tracking with Wireless Sensor Networks. *IEEE Access* **2016**, *4*, 1700–1715. [[CrossRef](#)]
- Srinivasan, S.; Dattagupta, S.; Kulkarni, P.; Ramamritham, K. A survey of sensory data boundary estimation, covering and tracking techniques using collaborating sensors. *Pervasive Mob. Comput.* **2012**, *8*, 358–375. [[CrossRef](#)]

14. Imran, S.; Ko, Y.B. A continuous object boundary detection and tracking scheme for failure-prone sensor networks. *Sensors* **2017**, *17*, 361. [[CrossRef](#)] [[PubMed](#)]
15. Ping, H.; Zhou, Z.; Shi, Z.; Rahman, T. Accurate and energy-efficient boundary detection of continuous objects in duty-cycled wireless sensor networks. *Pers. Ubiquitous Comput.* **2018**, *22*, 597–613. [[CrossRef](#)]
16. Wang, J.W.; Guo, Y.; Zhang, L. Dynamic pollutant plume front tracking and monitoring by a single mobile robot. In Proceedings of the Chinese Automation Congress (CAC), Jinan, China, 20–22 October 2017; pp. 2987–2992.
17. Fahad, M.; Guo, Y.; Bingham, B.; Krasnosky, K.; Fitzpatrick, L.; Sanabria, F.A. Ocean Plume Tracking with Unmanned Surface Vessels: Algorithms and Experiments. *arXiv* **2018**, arXiv:1804.08669.
18. Sun, T.; Pei, H.; Pan, Y.; Zhang, C. Robust adaptive neural network control for environmental boundary tracking by mobile robots. *Int. J. Robust Nonlinear Control.* **2013**, *23*, 123–136. [[CrossRef](#)]
19. Zhang, S.; Li, C. Control and Decision Making for Ocean Feature Tracking with Multiple Underwater Vehicles. *J. Coast. Res.* **2018**, *83*, 552–564. [[CrossRef](#)]
20. Jin, Z.; Bertozzi, A. Environmental boundary tracking and estimation using multiple autonomous vehicles. In Proceedings of the 2007 46th IEEE Conference on Decision and Control, New Orleans, LA, USA, 12–14 December 2007, pp. 4918–4923; [[CrossRef](#)]
21. Singh, A.; Nowak, R.; Ramanathan, P. Active learning for adaptive mobile sensing networks. In Proceedings of the 5th international conference on Information processing in sensor networks, Nashville, TN, USA, 19–21 April 2006; pp. 60–68.
22. Triandaf, I.; Schwartz, I.B. A collective motion algorithm for tracking time-dependent boundaries. *Math. Comput. Simul.* **2005**, *70*, 187–202; [[CrossRef](#)]
23. Lin, Z.; Liu, H.H. Enhanced cooperative filter for wildfire monitoring. In Proceedings of the 2015 IEEE 54th Annual Conference on Decision and Control (CDC), Osaka, Japan, 15–18 December 2015; pp. 3075–3080.
24. Zhang, F.; Leonard, N.E. Generating contour plots using multiple sensor platforms. In Proceedings of the IEEE Swarm Intelligence Symposium, Pasadena, CA, USA, 8–10 June 2005; pp. 309–316.
25. Kemp, M.; Bertozzi, A.L.; Marthaler, D. Multi-UUV perimeter surveillance. In Proceedings of the 2004 IEEE/OES Autonomous Underwater Vehicles (IEEE Cat. No.04CH37578), Sebasco, ME, USA, 17–18 June 2004; pp. 102–107.
26. Ermak, D.L. *User's Manual for SLAB: An Atmospheric Dispersion Model for Denser-Than-Air-Releases*; Office of Scientific and Technical Information: Oak Ridge, TN, USA, 1990.
27. Fluidyn France. *User's Manual for Fluidyn-PANACHE: Simulation of Fluids for Process and Environment*, 4.0.7 ed.; Fluidyn France: Saint-Denis, France, 2010.
28. Niewiadomska-Szynkiewicz, E.; Sikora, A.; Marks, M. A Movement-Assisted Deployment of Collaborating Autonomous Sensors for Indoor and Outdoor Environment Monitoring. *Sensors* **2016**, *16*, 1497. [[CrossRef](#)]
29. Krzysztoń, M.; Niewiadomska-Szynkiewicz, E. Adaptation of MANET topology to monitor dynamic phenomena clouds. In Proceedings of the 2017 Federated Conference on Computer Science and Information Systems (FedCSIS), Prague, Czech Republic, 3–6 September 2017; pp. 865–872.
30. Niewiadomska-Szynkiewicz, E.; Sikora, A. A Software Tool for Federated Simulation of Wireless Sensor Networks and Mobile Ad Hoc Networks. *Appl. Parallel Sci. Comput.*, **2012**, *7133*, 303–313. [[CrossRef](#)]
31. McQuaid, J. The box model of heavy gas dispersion: A useful and practical tool. *J. Occup. Accid.* **1984**, *6*, 253–261. [[CrossRef](#)]