Experimental studies of swarm robotic chemical plume tracing using computational fluid dynamics simulations

Dimitri V. Zarzhitsky
*Computer Science Department,
University of Wyoming, Laramie, Wyoming, USA*

Diana F. Spears
Swarmotics, LLC, Laramie, Wyoming, USA, and

David R. Thayer
*Physics and Astronomy Department, University of Wyoming,
Laramie, Wyoming, USA*

Received 12 July 2010
Revised 31 July 2010
Accepted 10 August 2010

Abstract

Purpose – The purpose of this paper is to describe a multi-robot solution to the problem of chemical source localization, in which a team of inexpensive, simple vehicles with short-range, low-power sensing, communication, and processing capabilities trace a chemical plume to its source emitter

Design/methodology/approach – The source localization problem is analyzed using computational fluid dynamics simulations of airborne chemical plumes. The analysis is divided into two parts consisting of two large experiments each: the first part focuses on the issues of collaborative control, and the second part demonstrates how task performance is affected by the number of collaborating robots. Each experiment tests a key aspect of the problem, e.g. effects of obstacles, and defines performance metrics that help capture important characteristics of each solution.

Findings – The new empirical simulations confirmed previous theoretical predictions: a physics-based approach is more effective than the biologically inspired methods in meeting the objectives of the plume-tracing mission. This gain in performance is consistent across a variety of plume and environmental conditions. This work shows that high success rate can be achieved by robots using strictly local information and a fully decentralized, fault-tolerant, and reactive control algorithm.

Originality/value – This is the first paper to compare a physics-based approach against the leading alternatives for chemical plume tracing under a wide variety of fluid conditions and performance metrics. This is also the first presentation of the algorithms showing the specific mechanisms employed to achieve superior performance, including the underlying fluid and other physics principles and their numerical implementation, and the mechanisms that allow the practitioner to duplicate the outstanding performance of this approach under conditions of many robots navigating through obstacle-dense environments.

Keywords Chemical hazards, Localization, Robotics, Simulation, Fluid dynamics

Paper type Research paper

International Journal of Intelligent
Computing and Cybernetics
Vol. 3 No. 4, 2010
pp. 631-671
© Emerald Group Publishing Limited
1756-378X
DOI 10.1108/17563781011094205

This research effort was partially supported by the National Science Foundation, Grant No. NSF44288.

1. Introduction

This paper discusses an extension of our earlier theoretical work on physics-based chemical plume tracing (CPT), also called chemical source localization, with teams of cooperating, autonomous robots, published recently in a special issue of the *International Journal of Intelligent Computing and Cybernetics* on swarm robotics (Spears *et al.*, 2009). Here, we focus on the practical implementation of the collaborative control and search algorithms, supported by comprehensive and rigorous experimental evaluation using computational fluid dynamics (CFD) simulations. We refer to the group of mobile robots as a swarm, and assume very restricted capabilities in terms of each vehicle's on-board sensing, communication, and computation equipment. The following are the three major contributions of this paper:

(1) For the first time, an extensive experimental evaluation is presented comparing our physics-based CPT strategy, called fluxotaxis, against its leading competitors under an exceptionally wide range of parametric conditions and performance metrics. Our approach uses vehicle formations as a fluid flow sensor network by relying on the robots on-board processor to perform a physicomimetic, i.e. physics-based, real-time analysis of a chemical plume to determine the optimal direction to search for the source of a chemical leak. We derived the fluxotaxis strategy from the fundamental physical laws that govern fluid flow, including the fluid physics of a chemical emitter (Zarzhitsky *et al.*, 2004, 2005b) Basically, fluxotaxis uses mobile robots to measure and trace changes in the chemical mass flux (Section 3) in order to find the chemical emitter. The experimental results presented in this paper demonstrate the superior performance of our fluxotaxis approach.

The significance of this outcome is that we can say with confidence that CPT successes are most likely to come from an algorithm that takes the underlying fluid dynamics of the chemical plume into account. This conclusion is based on our thorough literature search (summarized in Section 2), and our experimental studies (presented in Sections 5 and 6). Furthermore, the physicomimetic foundation of fluxotaxis makes it amenable to standard mathematical analysis techniques, which were used by Spears *et al.* (2009) to explain why the fluxotaxis approach is more effective at finding and identifying sources of chemical plumes than biologically motivated CPT methods. Sections 4.6.4 and 4.7.4 describe two software implementations of fluxotaxis: the first is meant to test a laboratory-scale prototype of seven CPT robots, and the second is designed for much larger teams of plume-tracing vehicles. Although the CPT simulation framework we use is implemented in a 2D surface environment, it should be noted that all of our analytic and computational analyses are robust and can be extended to three dimensions, which will result in similar conclusions to the ones presented here.

(2) The extension of CPT capabilities to large groups (swarms) of autonomous, mobile, cooperating robots is an important novel contribution of this research. Our project was the first to move CPT into the realm of swarm robotics and to demonstrate that the effectiveness of CPT is directly proportional to the size of the swarm. This paper is the first to describe the algorithms that make "swarm CPT" successful.

Even though the individual robots are limited in what they can do on their own, operating as a team they can solve a challenging search and localization task.

This is significant because it reveals the important insight that CPT is inherently a swarm application, and we provide all of the algorithmic necessities for achieving "swarm CPT." Most of the current CPT research is concentrated almost exclusively on a single or just a few vehicles performing chemical source localization (Li *et al.*, 2006). In contrast, our work is focused on a fully cooperative, decentralized, many-vehicle (i.e. swarm) CPT system. Section 6 is dedicated to an extensive study concerning the effects of increasing swarm size on chemical emitter localization performance. An important distinction of our implementation is the physicomimetic approach to vehicle formation control (Spears *et al.*, 2004). Analysis of our experimental outcomes indicates that formation-based cooperation between the CPT robots is crucial for improving search performance, and Section 4.1 provides an overview of the robots' on-board controller.

(3) Another advantage of our approach is its seamless integration of obstacle avoidance into CPT. Our approach (i.e. fluxotaxis, with physicomimetics as its foundation) is elegant and scalable. This paper is the first to present our approach in detail (including the algorithms) along with explanations of how and why fluxotaxis solves both the CPT problem and the obstacle avoidance problem simultaneously.

The significance of this elegant approach is its efficiency; it is efficient because it requires no additional obstacle-handling algorithms. Effectively, our approach uses the robots' environment to compute an obstacle-free route. In the same way that fluids flow around solid obstructions, the swarm moves around objects in its path, with the virtual vehicle-to-vehicle formation forces mimicking the role of real molecular bonds. The details of our obstacle avoidance approach are described in multiple publications (Zarzhitsky and Spears, 2005; Zarzhitsky *et al.*, 2005a).

Owing to space limitations, we limit this paper to simulation-based aspects of our CPT work, and our experimental results with CPT in the presence of obstacles are presented in Sections 5.2, 6.1, and 6.2. The long-term goal of this research effort is a fully functioning swarm of autonomous robots capable of executing cooperative chemical source localization in both indoor and outdoor environments. (Spears *et al.*, 2006) provide an overview of our progress toward this objective.

2. Related CPT research

To solve the CPT problem, one needs to apply a CPT strategy, also referred to as a "method", "approach", or "algorithm" in the literature. We summarize the most popular CPT strategies in this section; however, for a paper devoted entirely to surveying the state-of-the-art CPT approaches, see the recent CPT taxonomy by Kowadlo and Russell (2008).

Some of the earliest solutions to the CPT problem adopted a time-averaged approach for identifying the chemical. More recent research, however, indicates that mean statistics converge too slowly to be of practical use (Farrell *et al.*, 2002; Liao and Cowen, 2002).

2.1 Biomimetic approaches to CPT

In 2002, when the *Environmental Fluid Mechanics* journal published a special issue on CPT (Cowen, 2002), the papers in that issue were groundbreaking because they laid an initial foundation for the field. Like most prior and current CPT strategies, those presented in the special issue are based on olfactory systems of living organisms, e.g. insects such as moths, land animals, and aquatic creatures (especially crustaceans). In other words, the CPT strategies outlined in that special issue are biomimetic, i.e. they are designed to mimic biological systems. Because biomimetic approaches have traditionally dominated the CPT literature, we present them first.

2.1.1 Chemotaxis. Chemotaxis is the best understood and most widely applied CPT strategy. It consists of tracing the chemical, typically by following a local gradient of the chemical concentration within a plume (Krishnanand and Ghose, 2006; Marques and de Almeida, 2006). Some of the earliest research on chemotaxis was performed by Sandini *et al.* (1993). Among the most extensive applications of chemotaxis are those of Lilienthal (2005) and colleagues. In some of their work, they show chemotaxis success in an uncontrolled indoor environment (Lilienthal and Duckett, 2003). They have also explored chemotaxis in ventilated corridors with weak chemical sources (Lilienthal *et al.*, 2001).

Although chemotaxis is very simple to perform, it frequently leads to locations of high concentration in the plume that are not the real source, e.g. a corner of a room (Song and Chen, 2006). Cui *et al.* (2004) investigated an approach to solving this problem by using a swarm majority vote, along with a communication and routing protocol for distributing information to all members of a robotic collective. However, that is a strong requirement, and Cui *et al.* (2004) also make an even stronger assumption that each robot in the collective has a map of the environment. In addition to the local maxima problem of chemotaxis, we show in Spears *et al.* (2009) that a chemotaxis search strategy can fail near the emitter's location, due to the fact that, for a typical time-varying Gaussian distribution profile, the chemical density gradient goes to zero near the distribution's peak.

2.1.2 Anemotaxis. The second, most commonly used CPT strategy is anemotaxis, which is sometimes alternatively called odor-gated rheotaxis. An anemotaxis-driven robot measures the direction of the fluid's velocity (typically using an anemometer for sensing the wind direction and sometimes its magnitude as well), and navigates "upstream" within the plume (Hayes *et al.*, 2001; Marques *et al.*, 2005; Kowadlo and Russell, 2006). Here, it should be noted that the observed speed of the airflow is sometimes considered when calculating the robot's desired upwind velocity. Hayes *et al.* (2001) and Grasso and Atema (2002) have done pioneering work on anemotaxis. The simulation results of Iacono and Reynolds (2008) show that the effectiveness of anemotaxis improves with increased wind speed. More complex wind-driven strategies may be found in Kazadi *et al.* (2000). Ishida *et al.* (2006) gain performance improvements by coupling anemotaxis with vision capabilities.

Anemotaxis can be performed successfully with a single robot, or with a group of independent robots. Although anemotaxis is very effective for some problems, its limitation is that it can lead to a wind source that is not the chemical emitter (Zarzhitsky, 2008).

2.2 Hybrid approaches to CPT

More recently, there has been an increasing trend toward the development of hybrid (also called "multi-strategy") biomimetic CPT algorithms that combine chemotaxis, anemotaxis, and other CPT strategies (Ishida *et al.*, 2001; Russell *et al.*, 2000).

For example, Farrell has combined anemotaxis with other strategies, especially for recovering a lost plume or for identifying a chemical emitter. His research focuses primarily on a single autonomous underwater vehicle (AUV) in an aquatic environment (Li *et al.*, 2006). Jakuba *et al.* (2005) also use an actual AUV for CPT, in this case for hydrothermal vent localization. They effectively apply a combination of chemotaxis (to track non-conservative tracers) and outlier detection, along with an increasingly refined sequence of nested searches.

Grasso (2001) recommends the use of multiple strategies, where the choice of plume-tracing strategy depends on the fluid conditions. Likewise, Li *et al.* (2006) successfully apply a subsumption-based architecture for an AUV in turbulent waters. A subsumption-based (also called behavior based) algorithm applies the appropriate behavior (or strategy) in any given situation.

2.3 Other approaches

A less common, but often effective, CPT strategy is infotaxis. An example is following the gradient of the chemical intermittency, which is defined as the percentage of time during which the chemical is above threshold. Liao and Cowen (2002) demonstrate the advantages of intermittency infotaxis over alternative plume-tracing approaches. Vergassola *et al.* (2007) have developed a CPT strategy that locally maximizes the expected rate of information gain, in an information theoretic sense. Another infotaxis method is the maximum likelihood approach of Balkovsky and Shraiman (2002).

Kikas *et al.* (2001) found correlation analyses (another type of infotaxis) to be quite effective for CPT. Likewise, Weissburg *et al.* (2002) discovered that the “spatial correlation between spanwise-separated sensors reveals the relative direction of the plume centerline very rapidly” and this can lead to a heuristic strategy of following the plume centerline in turbulent fluids. Martinez *et al.* (2006) also use a similar strategy.

Balkovsky and Shraiman (2002) as well as Kowadlo and Russell (2008), use a zigzagging CPT strategy to increase the likelihood of encountering an odor patch in high turbulence. Kazadi *et al.* (2000) and Hayes *et al.* (2001) invented the spiral-surge CPT algorithm, especially designed for turbulent outdoor scenarios. With their algorithm, when one robot on the team gets an odor “hit” (above-threshold concentration), it “surges” (travels) upwind for a set distance. After the surge, if it still encounters odor then it continues surging. If not, the robot moves in spirals to “cast about” for another plume hit. It continues making consecutive spirals until another hit, at which point it surges again. To perform the surge collaboratively with a team of robots, all of the other robots surge toward any robot that communicates that it got a hit. Lochmatter and Martinoli (2009) continue to find this algorithm to be effective, to the present.

Early robotic experiments based on solutions to fluid dynamic problems are reported by Decuyper and Keymeulen (1991). In their approach, a simplified model of fluid flow forms the basis of a simulated robot’s navigation strategy. This method was inspired by the fact that continuous fluid flow can be used for iterative optimization of the local-to-global route-finding task, since the pressure fields responsible for stable optimal flow paths are void of local minima. A later development of this strategy by Keymeulen and Decuyper (1994b) also relied on the concepts of a fluid source and sink, which are used to represent the robot’s position. However, this early physics-based algorithm was used to navigate inside an obstacle-filled room, and because it does not involve emitter localization, it only addresses a part of the overall CPT problem.

2.4 Fluxotaxis for CPT

In order to physically describe a fluid medium, we need to specify its density, velocity, specific weight and gravity, viscosity, temperature, pressure, and so on. Owing to their relevance and importance to the goals of CPT, we focus on the scalar density ρ, which has the units of mass per unit volume, and the vector velocity $\vec{V} = u\hat{i} + v\hat{j}$, where $\hat{i}$ and $\hat{j}$ are the orthonormal basis vectors along the *x*- and *y*-axes in 2D Cartesian space. Throughout our discussion, we will treat the density, velocity, and other characteristics of the fluid, i.e. the flow-field variables, as functions of both space and time, that is to say that the chemical flow is unsteady. Since a moving robot cannot distinguish the flow of the chemical from that of the carrier fluid, our model uses a single velocity term $\vec{V}$ to denote the direction and speed of the plume. In addition, we use the term "density" and its symbol ρ to denote the concentration of the chemical being traced by the robots, rather than the density of the surrounding medium.

There are several different mathematical formulations of the fundamental physical laws that govern fluid flow. The differential form of the conservation of mass equation is:

$$\frac{\partial \rho}{\partial t} = -\vec{\nabla} \cdot (\rho \vec{V}). \tag{1}$$

By integrating over a differential volume element, one can see that this equation expresses the physical fact that the time rate of decrease of mass inside the differential element (left-hand side) must equal the net mass flux flow (right-hand side) out of the element.

To better understand the physics behind fluxotaxis, consider the divergence of velocity, $\vec{\nabla} \cdot \vec{V}$, which is the time rate of change of the volume of a moving fluid element, per unit volume. The divergence is the rate at which a fluid expands, or diverges, from an infinitesimally small region. (A local vector field with positive divergence expands, and is called a source; a vector field with negative divergence contracts, and is called a sink.) Of course, in addition to the velocity, we also need to consider the plume's chemical density. The key notion we seek is that of chemical mass flux, or mass flux for short. The mass flux is the product of the (chemical) density and the velocity, i.e. $\rho\vec{V}$. Informally, this is "the stuff spewing out of the chemical emitter," and this idea can be formalized as the divergence of mass flux, expressed in 2D as:

$$\vec{\nabla} \cdot (\rho\vec{V}) = \vec{V} \cdot \vec{\nabla}\rho + \rho\vec{\nabla} \cdot \vec{V} = u\frac{\partial \rho}{\partial x} + \rho\frac{\partial u}{\partial x} + v\frac{\partial \rho}{\partial y} + \rho\frac{\partial v}{\partial y}. \tag{2}$$

The divergence of mass flux is the time rate of change of mass per unit volume lost at any spatial position. If this divergence is positive, it indicates a source of mass flux; if negative, it indicates a sink of mass flux. Sustained (over a period of time) positive divergence of chemical mass flux (DMF) implies chemical spewing outward from a chemical source emitter, as opposed to a transient source. A logical conclusion is that following the gradient of equation (2) will take the robots closer to the source. This is the basic premise of the fluxotaxis CPT strategy, i.e. to follow the gradient of the divergence of mass flux, abbreviated *GDMF*, where the gradient is the direction of steepest increase. The mathematical formula for the *GDMF* in 2D is:

$$\vec{\nabla}\left[\vec{\nabla} \cdot (\rho\vec{V})\right] = \vec{\nabla}\left(u\frac{\partial \rho}{\partial x} + \rho\frac{\partial u}{\partial x} + v\frac{\partial \rho}{\partial y} + \rho\frac{\partial v}{\partial y}\right).$$

$\overrightarrow{GDMF}$ combines information about both velocity and chemical density, in a manner motivated by the theory of fluid dynamics. This ability to apply established mathematical analysis techniques to the underlying physical process enabled Spears *et al.* (2009) to prove the following useful theorems about fluxotaxis and chemotaxis CPT algorithms:

- *Source theorem.* Both fluxotaxis and chemotaxis will move robots toward a chemical source, i.e. a region with positive chemical mass flux divergence.
- *Sink theorem.* Fluxotaxis will lead robots away from a chemical sink, i.e. a local density maximum that is not a true emitter. On the other hand, chemotaxis will fool the robots by misleading them right into the sink.
- *Constant flow speed theorem.* Assuming a temporally invariant speed of the carrier fluid (subject to certain realistic and reasonable preconditions), fluxotaxis will lead a group of robots toward a chemical source emitter.
- *Continuous emitter in diffusion theorem.* Assuming a continuous emitter in a diffusion-dominated environment, i.e. where transport of the chemical due to fluid flow (also known as advection) is negligible and the chemical density distribution has a Gaussian profile, both fluxotaxis and chemotaxis will guide robots toward the emitter. Furthermore, both of these CPT strategies improve their effectiveness as the robots get closer to the source emitter.
- *Puff emitter in diffusion theorem.* In a diffusion-dominated environment, with a single-puff emitter, both fluxotaxis and chemotaxis will correctly direct robots toward a source emitter. Nevertheless, as the robots get nearer this emitter, chemotaxis will lose its predictive ability, but the predictive performance of fluxotaxis will improve. This is because chemotaxis uses only the first derivatives of the density as guidance, whereas fluxotaxis also uses the second-order spatial derivatives.

Since anemotaxis is a relatively trivial strategy, i.e. anemotaxis-driven robots simply travel upwind, we did not include this strategy in our in-depth theoretical analyses. All of the above theorems have been confirmed with methodical experiments under controlled conditions, as described in Spears *et al.* (2009). The source and sink theorems are particularly important because they are widely applicable – in any fluid and source-emitter conditions, and both are confirmed by the experimental results presented in this paper.

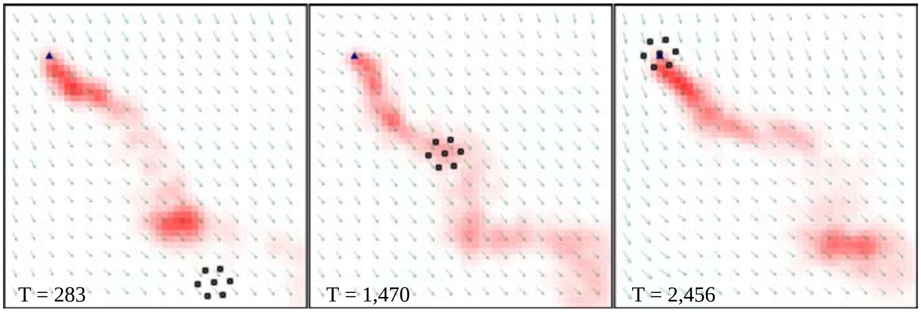
3. Modeling chemical plumes with CFD

In order to develop and evaluate a new multi-robot CPT strategy, we needed a way to create many different types of chemical plumes. Using real chemical pollutants in a controlled environment was impractical, due to the high costs and the effort involved. Drawing on the fundamental paradigm shift that occurred in the field of aircraft design, where the traditional use of experimental wind-tunnel testing has been largely supplanted by computation methods, we likewise have adopted a strategic decision to concentrate on computational models of the plume. This section explains the most important aspects of modeling the properties and long-term behaviors of chemical plumes using CFD, with a particular emphasis on the plume-tracing task.

In the context of chemical source localization, it is useful to categorize numerical fluid simulations as forward and inverse solutions. The "forward" solution first provides a time-accurate description of the chemical flow, and the "inverse" solution adds a plume-tracing algorithm for finding the source emitter (i.e. the latter is CPT). We rely on numerical modeling for the bulk of the CPT strategy design and refinement, which reduces the number of experiments needed to validate the real robots on chemical plumes in the laboratory. This section focuses on the forward solution; Section 4 will address the inverse solution.

To accurately predict fluid behavior, a system of governing equations is typically solved using computational techniques in order to obtain values of the flow-field variables at discrete positions in space, i.e. at "grid points." We construct a mapping between these grid points and the robots, motivated by our view of a team of cooperating robots as a distributed, adaptive, computational mesh that jointly senses the flow-field variables, shares them with their immediate neighbors, and then decides in which direction to move.

We researched and evaluated several different applications suggested in the literature, and as the result of this investigation, we selected the software developed by Farrell *et al.* (2002) at the University of California, Riverside, which we believe to be the most practical and well-developed among all of the plume-oriented simulators accessible to us. It is optimized for computational efficiency, while also serving as a realistic and faithful model of the environment, i.e. both its transient and steady-state statistics match closely the observations of actual airborne chemical plumes. A notable feature of the solver is its multi-scale aspect, including molecular diffusion of the chemical and the advective transport due to wind movement. Rather than a more conventional, continuous, time-averaged model, Farrell's simulator models the plume as a collection of filament-based emissions of chemical "puffs." Air currents smaller than the mean distance between puff centers are modeled as a white noise process. In other words they "mix" the puff components, mimicking the effect of small scale flow turbulence on the chemical plume. Air currents on the order of the puff size induce growth and distortion of the puffs' template shape, and are therefore modeled using differential equations. Fluid advection (obtained via a numerical solution) transports each puff as a whole, thus causing the ensemble of puffs to appear as a meandering sinuous plume, as shown in Figure 1.



Note: The lattice starts out in the lower right corner and uses the fluxotaxis algorithm to successfully trace the plume to its source (the triangle)

Figure 1.
CPT simulation with a seven-vehicle physicomimetic lattice (black rectangles), no obstacles, and a meandering plume originating in the top left corner (higher chemical concentration is shown with darker colors, and the arrows denote the airflow)

Because of the open source license for Farrell's work, we were able to re-implement the original simulator to better fit our research goals and to adapt it to the most recent and powerful workstation hardware. Our re-implementation improves the computational efficiency of the code via memory optimization methods and multi-threading technologies. Our new fluid model is also more general than that in the original code because it allows for solid obstacles to be introduced into the flow. It is very important to emphasize that the model is accurate for a wide range of fluids, and is capable of resolving dynamics of both gaseous and liquid plumes. The next few sections provide a more detailed description of the plume-simulation model.

3.1 Advection due to wind

Fluid velocity is generated using a simple procedure. First, random wind vectors are placed in the four corners of the world. Then, using linear interpolation among the four corners, new velocity vectors are computed along each of the boundary edges, such that at the end of this step, wind velocity is completely specified on the world boundaries (i.e. a Dirichlet condition). Once the boundary conditions are determined, we use a second-order accurate forward Euler in time, central difference in space numerical approximation algorithm to calculate wind velocity across the interior points. In the current version, all grid spacings are uniform, and wind velocity is obtained directly from the discretization equation.

3.2 Mixing due to random velocity

During each step of the simulation, a random amount σ is added to the advective velocity vector to model the diffusion of the plume about its centerline (i.e. the mixing of the filaments within the plume). The implementation of time correlation, bandwidth, and gain in the wind vectors follows the exact specification and code examples provided by Farrell *et al.* (2002).

3.3 Growth due to diffusion

It is assumed that each puff of the chemical ejected by the emitter has the same "template shape," and for convenience and efficiency, it is treated as a sphere (and approximated by a disk in 2D simulations) when the total plume density is computed. Thus, the only parameter that affects the diffusion model is the size of each filament k, with the radius $r_k(t)$ at time t specified by:

$$r_k(t) = \left(r_o^{2/3} + \gamma t\right)^{3/2} \tag{3}$$

where r_o is the initial radius of a puff when it is first ejected by the emitter, t is the age of the puff, and γ is the volumetric diffusion rate.

3.4 Density computation

The plume simulator only needs to keep track of the movement of the chemical puff filaments, driven by the wind and dispersed by inter-molecular forces. However, since we need a density map of the entire chemical plume for the tracing experiments, we determine the individual density contributions of each of the M puffs to the overall volumetric chemical concentration $\rho(x, y, t)$ at point (x, y) and time t via:

$$\rho(x,y,t) = \sum_{k=1}^{M} \rho_k(x,y,t),$$

where:

$$\rho_k(x,y,t) = \frac{Q}{\sqrt{8\pi^3} r_k(t)^3}\, e^{(-D(2/k)(t)/r_k(t)^2)} \tag{4}$$

and Q is the amount of chemical contained in each puff, $D_k(t)$ is the distance between the kth puff center at time t and the point (x,y), for which the density contribution of puff k with the radius $r_k(t)$, as defined in equation (3), to the total plume density $\rho(x,y)$ is calculated. Farrell *et al.* (2002) employed this update rule to reduce a spherical puff into a flattened, disk-like filament, assuming that the chemical has a Gaussian (i.e. exponential) distribution within the puff. Once the plume concentration is thus calculated for the entire search area at each time step, we can commence the tracing and source localization mission with the CPT robots – a task that is explained in detail in the following section.

4. Cooperative swarm-based CPT

The software framework we just described gives us the forward solution, i.e. it supplies us with large amounts of high-resolution plume data. In this section, we focus on the inverse solution, and explain how we develop and test new CPT strategies. As we mentioned in the beginning, even recent CPT strategies appearing in the literature tend toward either non-cooperative or centralized emitter localization algorithms (Marques and de Almeida, 2006). However, our goal is the development of a robust, fully distributed, multi-robot (i.e. swarm) solution to the CPT problem. To meet this primary objective, two other problems must be solved: the first is the cooperative control of the swarm, and the second is the extension/adaptation of the CPT strategies to a distributed, decentralized grid of mobile sensor nodes. The next few sections explain how we addressed these challenges.

4.1 Swarm control framework of physicomimetics

The physicomimetic design method provides the key control technologies that make it possible for us to achieve CPT objectives using a mathematically consistent approach. The name itself unambiguously reflects our goal of cleverly mimicking physical systems in order to obtain the desired results. Sometimes, we also refer to physicomimetics as artificial physics because we are not restricted to a slavish imitation of the real world, but rather we focus on the fundamental operational principles of real physical systems, while also employing a designer's license to formulate the rules in a way that suits the task at hand. In our formulation, virtual physics forces drive a multi-robot system to a desired configuration, where the virtual potential energy of the system is minimized. Unlike behavior-based approaches (Balch and Arkin, 1998), physicomimetics uses an interacting particle-force dynamics simulation. An important difference from other physics-inspired methods appearing in the literature is the fact that all of the physicomimetic virtual forces obey real physical laws, such as the fundamental $\vec{F} = m\vec{a}$ (force equals mass times acceleration) principle. Our experiments show that physicomimetics is a powerful swarm design tool, optimized for self-assembly and self-repair of robotic lattices, where dynamic grid-like vehicle formations are constructed via short-range (i.e. local) virtual physics forces.

A high-level description of the basic physicomimetic control algorithm is straightforward. During each decision cycle (or time step), every agent observes its neighbors and the environment. The agent then calculates the virtual forces imposed upon it by these entities. After taking a vector sum of all forces on the agent, it computes derivatives to convert the net force into a velocity vector for the agent's next move. Agents perform this cycle concurrently in the software simulated instantiation of physicomimetics. The physicomimetic swarm controller adheres closely to the physical model that governs the macroscopic behavior of real systems. That is why the equations of motion for a physicomimetic-controlled vehicle look identical to those commonly employed in classical particle physics.

Spears *et al.* (2005c) examined two different physicomimetic control laws in the context of a distributed surveillance application using a team of autonomous aerial vehicles. Drawing on their results, we designed a control law based on the Lennard-Jones potential, which we generalize as:

$$\vec{F}_{\text{formation}} = \varepsilon \frac{\mathscr{D}(R)^{\alpha}}{r^{\beta}} - \kappa \frac{\mathscr{D}(R)^{\gamma}}{r^{\delta}}. \tag{5}$$

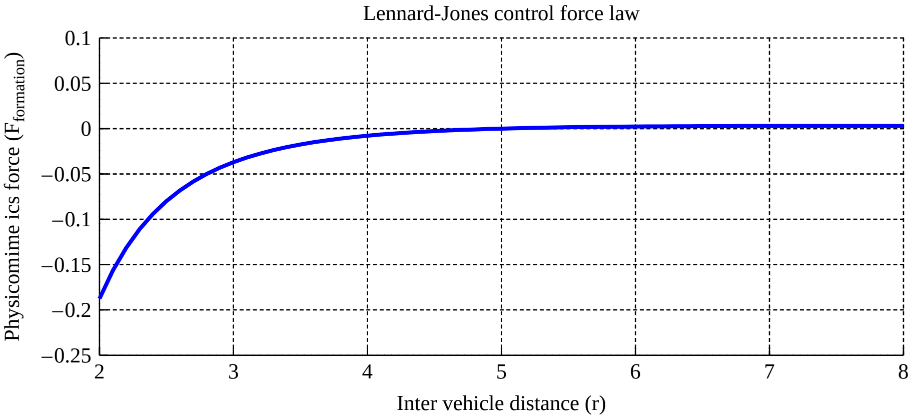
Assuming that all parameters are non-negative, the first term, $\varepsilon\mathscr{D}(R)^{\alpha}/r^{\beta}$ describes the attractive component of the formation force, and the second term, $\kappa\mathscr{D}(R)^{\gamma}/r^{\delta}$ specifies the repulsive component. Exponential coefficients α, β, γ, and δ determine the distance proportionality of the force power law, and in typical usage, $\alpha < \beta \leq \gamma < \delta$. Optional scalar coefficients ε, and κ are used to linearly scale the attractive and repulsive components of the formation force, which in turn affects the cohesion of the swarm by altering the strength of vehicle-to-vehicle bonds. Symbol r denotes the actual distance between two robots in a lattice, and the function:

$$\mathscr{D}(R) = \left[\frac{\varepsilon R^{\delta-\beta}}{\kappa}\right]^{(1/\gamma-\alpha)}$$

is used to compute the Lennard-Jones separation parameter in equation (5) for a given desired distance between the vehicles R, since in the Lennard-Jones control law, the exact distance at which the formation force achieves equilibrium depends on all of the scalar and exponential parameters. Figure 2 shows a plot of equation (5), with $\varepsilon = \kappa = 1$ and $\alpha = 1$, $\beta = \gamma = 2$, $\delta = 3$, so that for this limited, simplified case $\mathscr{D}(R) = R$. The Lennard-Jones force is mostly repulsive, with a weak attractive component (in Figure 2, $\vec{F}_{\text{formation}}$ is greater than zero for $R > 5$, albeit by a small amount). Because the attractive component of the force is small, this control law is especially well-suited for constructing swarms with liquid-like virtual formation bonds, which is desirable for robots operating near obstacles or narrow passageways.

4.2 Simulation model assumptions

In order to faithfully emulate our physical vehicle platforms, the CPT robots in simulation are modeled as homogeneous disk-shaped entities of a fixed radius, so that each simulated vehicle occupies a small amount of space, which is an important prerequisite for accurate modeling of obstacle and collision avoidance. The simulator makes a simplifying assumption that the obstacle and chemical sensors are mounted in the center of the robot's circular frame; thus, it is possible to determine the appropriate sensor observations



Note: The x-axis is marked with units of distance and the y axis with units of force

Figure 2.
Lennard-Jones physicomimetic force law with $R = 5$, $\varepsilon = \kappa = 1$ and $\alpha = 1$, $\beta = \gamma = 2$, $\delta = 3$

that should be accessible to the robot based only on the location of the vehicle, without having to account for its orientation. To facilitate efficient memory management, we discretized the simulated environment into an array-like collection of cells. Each of these cells contains the CFD solution for the chemical plume and the ambient flow, i.e. fluid velocity $\vec{V}$ and chemical density ρ, within a specific region of the environment. A typical test configuration also contains rectangular obstacles; regions occupied by obstacles are assumed to be free of chemical flow and are impenetrable by the robots. For measurements of discretized quantities, such as the plume chemical density, the simulator maps the robots' real-valued coordinates into the corresponding 2D data array.

As mentioned briefly in the introduction, the long-term goal of this work is a fully distributed, swarm-based CPT system capable of finding chemical sources in the context of practical, real-world scenarios. We anticipate that the very first implementation will consist of ground-based robots and airborne chemical plumes, partly due to the more accessible logistics associated with the ground/air scenario. In fact, we have already developed the prototype of a small-scale plume-tracing vehicle (Zarzhitsky, 2008), and the main reason that our experimental work here is focused on airborne chemical plumes is the need to establish a baseline for the future swarm experiments in the laboratory. (Once again, we emphasize that the theoretical foundation of this work is equally applicable to both 2D and 3D air, ground, and aquatic surface/sub-surface platforms.) As previously stated, a successful, swarm-based solution of the CPT problem requires advances in both the control and emitter localization algorithms, which is what motivated us to organize our experimental work into two major categories – the first one is focused on robot team control, and the second is more concerned with the plume-tracing and emitter localization performance of the different CPT strategies, as explained below.

4.3 Motivation for two different CPT studies

Our literature search (Section 2) failed to find a CPT implementation designed specifically for cooperative swarms, and even some of the most recent developments in this area (Balkovsky and Shraiman, 2002; Vergassola *et al.*, 2007), focus on single-robot solutions. To address this gap, we carried out many comprehensive CPT simulations of different plume and swarm configurations to study the key aspects of cooperative CPT, such as distributed sensing, local information sharing, scalability, and robustness. Thus, we adopted a two-part approach for our investigation. First, we studied the issues associated with physicomimetic control of CPT robots using a fixed number of vehicles. Once the control and communication technologies were designed, tested, and evaluated, we relaxed the fixed team size constraint, and explored scalability and the distributed aspects of swarm-based CPT as part of our second study. We split each study into two separate experiments, each one focusing on a particular aspect of the CPT problem, as explained in more detail below. Our presentation here follows the original, chronological order of this work, with each new experiment building on the conclusions of the previous investigation. Prior to presenting the experiments, we first give a concise summary of the common implementation details that remain the same in each of the two studies.

4.4 Hardware implementation aspects common to both studies

All of the experimental results presented in Sections 5 and 6 are based on a faithful software simulation of our laboratory plume-tracing robots and their chemical sensor payload. In prior work, we demonstrated successful cooperative CPT missions in an unstructured laboratory environment using as few as three robots based on the Freescale HCS12 micro-controller operating at 25 MHz with 16 KB of random access memory (Zarzhitsky, 2008). Each robot is equipped with a novel "neighbor" sensor, which was first developed by Heil (2004) and then refined by Kunkel (2006). This trilateration-based sensor, where reference separations are determined using a time-of-flight difference between electro-magnetic and acoustic pulses, allows each vehicle to determine the distance and bearing to all of its neighbors within a certain maximum sensor radius R_s. Since our vehicle platform is approximately 0.3 m in diameter, we limited the vehicles' localization sensor and communication range R_s to 1 m. Although this value is shorter than what is supported by our laboratory hardware, it is sufficient to demonstrate behavior of any CPT algorithm that only uses plume measurements reported by neighboring vehicles. In addition, we superimposed a data-carrier signal on top of the radio localization beacon, and used a token-based scheme for network packet collision arbitration (Spears *et al.*, 2006).

Swarm movement and vehicle formations are coordinated via strictly local interactions between agents, using the physicomimetics framework. We use a parametrized version of the Lennard-Jones formation force law (5), F_{LJ}, which we instantiated as:

$$F_{\mathrm{LJ}} = \frac{R}{r^2} - \frac{R^{1.7}}{r^{2.7}}.$$

This equation gives the virtual formation force between two vehicles separated by a distance of r, for a lattice with the ideal vehicle-to-vehicle separation set to R. We selected the given numeric values for the exponent coefficients based on the specifications of our laboratory plume-tracing hardware. In particular, we assumed the maximum robot speed of just 0.1 m/s, which is significantly slower than the maximum velocity supported by our laboratory vehicle platform. However, this speed limitation helped reduce the number of potential vehicle and obstacle collisions, particularly with the large swarm sizes evaluated in the second study.

Another implementation aspect worth mentioning is the sharing of vector-valued measurements, such as air velocity or chemical mass flux, between the vehicles.

To accomplish this task, the robots require a mechanism for establishing a common reference coordinate frame. We employed a simple and yet effective approach based on a small digital compass, which allowed us to perform appropriate coordinate transformations directly in the sensor hardware module, thus simplifying the implementation of the CPT algorithm (Spears *et al.*, 2006). The model for the chemical sensor itself is based on the TGS 2620 metal-oxide detector manufactured by Figaro Inc. (2008). Lastly, although we were unable to find a commercial, off-the-shelf anemometer that would meet our needs for a small-scale, low-power device capable of interfacing with the rest of our robot hardware, we based the wind sensor emulation on custom airflow measuring devices described in the CPT literature (Russell *et al.*, 2000; Ishida *et al.*, 2001; Harvey *et al.*, 2003).

4.5 Software implementation aspects common to both studies

All of the empirical work discussed in this paper employs the same 2D plume simulator, i.e. the forward solution. By changing the boundary conditions of the differential governing equations, we can vary the plume configuration for the two experiments that make up each study in a physically predictable manner. To ensure that these simulations have practical CPT relevance, each study includes a balanced mix of laminar, transitional, and turbulent flows, and all of the source localization results we report here are based on a methodical combinatorial evaluation of each CPT strategy against the same, pair-wise matched flow conditions (Zarzhitsky *et al.*, 2004). To further improve statistical properties of our experiments, we used a high-quality random number generator to create all of the CPT test environments. We performed several experiments in each environment and plume configuration with different lattice parameters, but the CPT objectives remained the same in each experiment – the lattice must first search the environment for the chemical plume (using the process called casting), and then determine the location of the chemical emitter (here, we assume a single, stationary origin for the chemical plume). In other words, each individual CPT run consists of a search for the chemical plume, followed by the trace to its source emitter. Because of some important implementation differences, we will explain the relevant details of each of these steps separately for each study in the next few sections.

Note that each CPT algorithm can be broken down into low-level lattice movement control routines and emitter localization functions. The low-level routines are responsible for moving each agent in a formation, executing collision avoidance, and reporting flow-field variable sensor readings when requested by the on-board CPT algorithms. Algorithm A1 (Appendix 1) shows the control decisions in every simulation step. Vehicle velocity is modified in response to the different constraints; thus the call to ap_maintain_formation will alter each robot's desired velocity according to the formation forces acting on the vehicle. The agent_do_cpt_strategy modifies the agents' velocities according to the current plume-tracing algorithm (implementation details are presented in the next few sections). Once the new velocity vector of each vehicle is computed, the final call to move-agents-with-constraints ensures that no agent goes out of the search area boundaries, that the agents' velocities are consistent with mechanical limits, and that there are no collisions between vehicles and obstacles. The robots use a hierarchical architecture to avoid obstacles; in other words, the output of the CPT algorithm is ignored when the collision avoidance behavior is activated by the sensor module in the vicinity of obstacles or other vehicles that are closer than the collision threshold separation. Finally, recall from the "Introductions" that one of the main contributions of this work is that our CPT approach does not require any additional obstacle-handling algorithms. In particular, obstacles are not specifically identified, nor are they treated differently from anything else in the environment; any environmental entity that is too close exerts a repulsive force on the robot.

Once again, notice the important role that formations of robots play in all of our CPT algorithms. The grid-like arrangement of robots in hexagonal formations is what allows the vehicles to function as a distributed computer. Earlier (Section 2.4) we explained how differential equations are used by physicists to model fluid flow. All of the governing equations are continuous, but in order to solve them numerically, we use a discrete approximation based on the finite-difference method (Faires and Burden, 2003). Our algorithms use the robots as sensor nodes to measure both spatial and temporal flow characteristics at specific points in space. The robot formations are dynamic, capable of adapting to a local geometry near obstacles, building corners, etc. Since each vehicle is equipped with a sensor that can estimate the range and bearing to neighboring robots, each member of the team knows the local topology of the resulting sensor network, and can "solve" the differential equations using the sensor data broadcast by the nearby teammates. Thus, the gradients and derivatives in the mathematical description of the chemical plume can be computed at each time step, and intelligent navigation decisions made based on this insight. The next few sections focus on the software implementation of these procedures.

4.6 Software implementation aspects specific to the seven-robot lattice

To better understand the requirements of physicomimetic-based CPT, we started out with a conceptually simple configuration of seven robots arranged in a hexagonal formation (with one vehicle in the center of the formation). The hexagonal lattice geometry was selected because it requires the least amount of sensor information to construct and maintain (Spears *et al.*, 2004), it represents a fundamental structural element within larger physicomimetic swarms (Spears *et al.*, 2005a), and it provides a computationally convenient surface for measuring the volumetric flow of chemical mass flux (Tannehill *et al.*, 1997). The following few sections give a more detailed description of the CPT strategies implemented as part of the first single, seven-robot hexagonal lattice study. All algorithm listings for these implementations appear in Appendix 2.

4.6.1 Casting algorithm. The sole purpose of the casting algorithm is to help the robots locate the chemical plume. As is the case with most search algorithms, it is important to minimize search time by means of an aggressive exploration of the environment. To help maximize spatial coverage, our implementation (Algorithm B1) combines the translational side-to-side, up-down motion of the center vehicle with the periodic expansion and contraction of the outer ring of the lattice in a pulsating motion. The center robot uses local waypoints to navigate (i.e. the location of each new waypoint is computed using the position data obtained by the vehicles on the lattice perimeter), while the expansion and contraction of the lattice is implemented by changing the desired inter-vehicle separation R on each robot. Each of the CPT strategies (defined below) uses this casting method to find the plume at the start of the mission, as well as during the tracing in cases where the robots unintentionally exit or otherwise lose track of the plume.

4.6.2 Chemotaxis algorithm. The chemotaxis CPT algorithm is based on the intuitive fact that the concentration of the trace element increases in the vicinity of the chemical source; then, one possible way to find the source is to follow the gradient of the chemical density. The chemotaxis algorithm relies extensively on the neighboring vehicles to share their chemical sensor information, since the gradient computation requires the chemical density reading at several spatially separated points. The implementation of the algorithm in pseudo-code is given in Algorithm B2.

4.6.3 Anemotaxis algorithm. The intuition behind the anemotaxis CPT strategy is to move the lattice upstream while keeping the vehicles inside the plume. This implementation of the anemotaxis algorithm (see Algorithm B3 for the pseudo-code) is based on the examples in the literature; however, one notable improvement is the explicit averaging of the ambient wind direction, calculated by combining the information about wind conditions as reported by each robot in the lattice.

4.6.4 Fluxotaxis algorithm. We pioneered an entirely new approach for finding chemical emitters by using the DMF as a guide. Originally, we postulated this methodology as a fluid physics-based method for identifying chemical sources. Once we completed the theoretical analysis of this technique in Spears *et al.* (2009), we realized that the chemical flux conveniently combines the information about the chemical concentration and fluid velocity. In particular, mathematically the DMF can be separated into two terms:

$$\vec{\nabla}\cdot(\rho\vec{V}) = \rho(\vec{\nabla}\cdot\vec{V}) + \vec{V}\cdot(\vec{\nabla}\rho). \tag{6}$$

Therefore, the $\overrightarrow{GDMF}$ is the gradient $\vec{\nabla}$ of the sum of two terms:

(1) $\rho(\vec{\nabla}\cdot\vec{V})$, which is the density times the divergence of the velocity field; and

(2) $\vec{V}\cdot(\vec{\nabla}\rho)$, which is the flow velocity field in the direction of the density gradient.

When the chemical plume is divergent, the first term takes precedence in guiding the robots, similarly to anemotaxis. Otherwise, $\vec{\nabla}\cdot\vec{V}$ is zero and only the second term, which is analogous to chemotaxis, matters. Therefore, *GDMF* automatically recovers chemotaxis and anemotaxis traits based on the environmental conditions. We extended and adapted the basic *GDMF* flux method for traversing and searching an evolving chemical plume. The fluxotaxis algorithm presented here (Algorithm B4) is one of the two implemented versions.

This initial version of fluxotaxis emphasizes the interactions between the CPT algorithm and the physicomimetic control framework that manages the robot lattice, reflected prominently in the tight integration of the CPT actions (e.g. acquiring plume sensor readings at different lattice radii, R_i) with the lattice formation control actions. Although we realize that this coupling does not support our overall goal of a scalable CPT algorithm for arbitrary-sized swarms, we created this version to study the impact of the lattice formation on the CPT algorithm's performance. The chem_region strategy in Algorithm. B4 contains references to chemical centroids, C_p, which serve as waypoints for the lattice. The position of these centroids, $\vec{r}_{C_p}$, is computed as a weighted sum based on the chemical density measurement reported by each robot. Mathematically, this computation is:

$$\vec{r}_{C_p} = \begin{bmatrix} x_{C_p} \\ y_{C_p} \end{bmatrix} = \frac{1}{N\sum_{i=1}^{N}\rho_i}\sum_{i=1}^{N}\rho_i \begin{bmatrix} x_i \\ y_i \end{bmatrix},$$

where $N = 7$ is the number of CPT robots in the lattice, ρ_i is the output of the chemical detector on robot *i*, whose location is (x_i, y_i).

Note that the implementation of the flux_ring strategy in Algorithm B4 is simply a discretization of the *GDMF* technique we first presented in Section 2.4. In addition, when performing this flux computation, the robots distinguish between incoming and outgoing fluxes, as determined by whether the chemical flow is into or out of the lattice. This explicit separation of the two types of fluxes is necessary because in this study, the single, virtual, hexagonal "surface" constructed by the seven-robot lattice is the only available control volume suitable for calculating the chemical mass flux. In order to compute the GDMF, the lattice expands its radius, making it possible to measure the surface flux across three virtual surfaces of increasing size. Because of the limited number of CPT agents, the gradient estimate based on the expanding-radius surface is consistent with the spatial characteristics of our *GDMF* analysis, but it is not accurate with respect to time, since the plume continues to evolve and move while the lattice is undergoing the radial expansion. We address this limitation in our second experiment, where we extend the fluxotaxis algorithm to work with arbitrary-sized swarms.

In the flux_ring strategy, the robots measure the chemical concentration and the airflow across each of the outer edges of the lattice, first computing the maximum incoming flux, and if that metric is below a preset threshold, they attempt to find the location of the maximum outflux. Note that the incoming flux is indicative of the condition in which the plume "impinges" on the leading edges of the lattice, thus the algorithm selects the location of maximum chemical influx in order to keep the robots inside the plume. Here, we use the term "lattice edge" as a label for the virtual connection between the outer edges of the hexagon formed by the vehicles, shown with bold black lines in Figure 3. Fluxotaxis agents compute the chemical mass flux, $\rho\vec{V}$, across each of the outer edges of the lattice using an average of the ρ and $\vec{V}$ values measured by each robot, and for the purpose of the flux computation, the distance between the adjacent vehicles is taken as the "surface area" of unit depth, matching the length of the virtual edge. If we denote the flux through the virtual edge between vehicle *A* positioned at (x_A, y_A) and vehicle *B* at (x_B, y_B) by $\vec{L}_{\mathrm{AB}}$, then the "origin" of this vector (x_L, y_L) is taken as the midpoint of the *AB* edge:

$$\begin{bmatrix} x_L \\ y_L \end{bmatrix} = \frac{1}{2}\begin{bmatrix} x_A + x_B \\ y_A + y_B \end{bmatrix}.$$

Any time the fluid flows out of the lattice, the averaged flux value is treated as the outflux by the fluxotaxis algorithm, and any time the fluid flows into the lattice, the flux calculation is treated as the influx. In the actual software implementation, this flux computation is performed using a standard technique of 2D finite-volume discretization for general control volumes (Tannehill *et al.*, 1997).

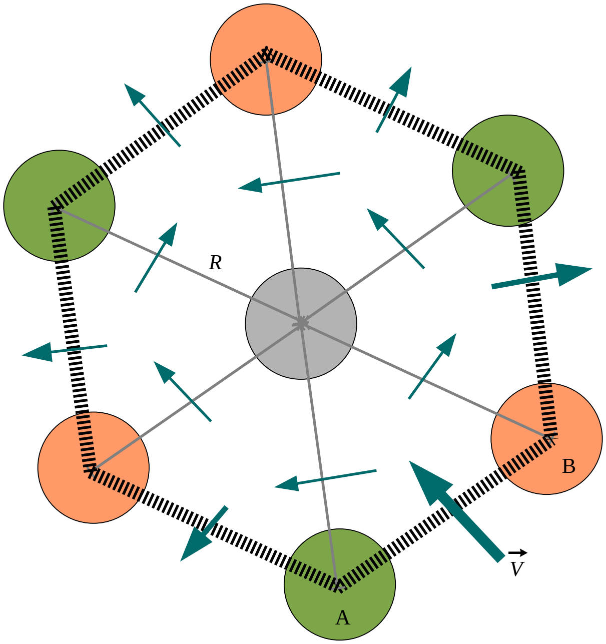


Figure 3.
A typical hexagonal lattice formation assumed in the first CPT study

Notes: The lattice radius, R, is a dynamic parameter that controls lattice expansions and contractions; the fluid passes through the virtual surfaces formed between the robots with velocity $\vec{V}$; the chemical mass flux across a lattice edge, $\rho\vec{V}$, can be estimated via an interpolation of the chemical density and flow velocity measurements obtained by the adjacent robots

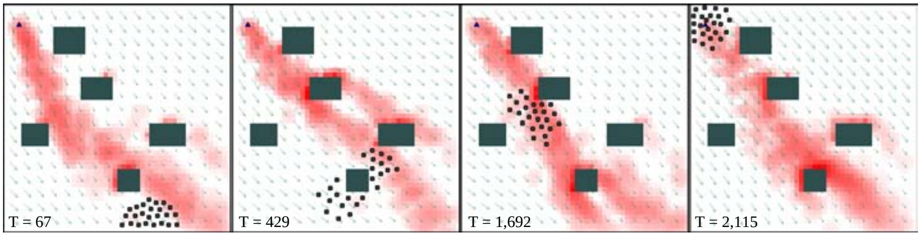
4.7 Software implementation aspects specific to the large swarm

The second CPT study removes the seven robot restriction. In fact, it makes no assumptions about how many vehicles are deployed, nor does it attempt to explicitly control the outermost shape of the virtual surface formed by the vehicles. Instead, it models a real-world configuration, where each robot makes its navigational decisions based solely on the local information from the on-board sensors. This means that the number of neighbors that each vehicle has, the neighbors' positions and their sensor readings, as well as the behavior of a given CPT strategy all represent fully dynamic swarm properties that update continuously throughout the simulation. Once again, we focus our attention on three different CPT algorithms. Similarly to the previous study, we adapted the standard implementations of chemotaxis and anemotaxis from the literature, and extended the fluxotaxis method to an arbitrary-sized and shaped swarm. The robots' on-board sensor package consists of an anemometer, which can compute local wind velocity $\vec{V}$, and a chemical concentration sensor, which measures and reports the value of ρ when it exceeds a predetermined threshold.

During the mission, vehicles form many dynamically stable, embedded hexagonal formations as they move about the plume-tracing area. These virtual formation bonds often rearrange or break altogether as the result of obstacle avoidance, and the movement of other neighboring vehicles, as shown in Figure 4. The most important difference between this implementation of the swarm controller as compared to its counterpart in Section 4.6 is the lack of waypoint navigation. Instead, the vehicles' velocities depend on real-time changes in the local swarm formation topology, as well as the presence of obstacles or the chemical in the vicinity of the robots. All pseudo-code listings for these implementations appear in Appendix 3.

4.7.1 Casting algorithm. Regardless of the CPT strategy, an identical casting algorithm is used in each experiment to help the swarm find the plume. For this study, we implemented a modified casting procedure, detailed in Algorithm C1, that moves the robot along 45° diagonal paths. Near obstacles and search area boundaries, each vehicle executes a "bounce-like" reflection maneuver. To ensure the swarm has the ability to coordinate its movement during casting, we simulate an explicit communication message, for which we use a recursive local broadcast to synchronize two key casting parameters, labeled as cast_latitude and cast_longitude in Algorithm C1. Each CPT agent starts out with the same casting state, and stops casting once the chemical plume is located. Each robot's casting state is distinct, so that a part of the swarm that has split up and moved out of the communication range can cast, while the remaining group can continue tracing the plume. Vehicles that are within the communication range will periodically synchronize their casting goals in order to reduce formation stresses.

4.7.2 Chemotaxis algorithm. The chemotaxis algorithm has a simple swarm-oriented implementation (Algorithm C2): all vehicles broadcast their own chemical sensor measurements, wait for their neighbors to do the same, and then compute a local chemical gradient, which acts as a goal force that propels the robots toward regions with high concentrations of the trace element. This algorithm implementation is a good model for a scalable sensor network that takes advantage of the automatic aggregation of the chemical signal that emerges directly from the physicomimetic control laws. Because the goal forces are balanced by the formation forces, the trajectory of each individual vehicle is an implicit function of all chemical density measurements collected by the swarm. This emergent behavior has two notable benefits: first, the sensor fusion method is implicit in the topology of the network – we never had to design one specifically for this problem, and second, the movement of each vehicle is minimized because the impact of the high-frequency transients (noise) in the chemical density signal is automatically filtered out due to the averaging of the individual robots' sensor readings. The practical utility inherent in these swarm properties presents yet another compelling reason for using the physicomimetic approach.



Notes: Note how the swarm "breaks up" into two groups in order to negotiate the obstacle (second image from the left), and then "rejoins" due to formation and plume-tracing forces (second image from the right); after finding the emitter, the fluxotaxis-controlled swarm surrounds the source and continues to enclose it throughout the rest of the CPT mission (the rightmost image)

Figure 4.
CPT trace near obstacles (dark rectangles)

4.7.3 Anemotaxis algorithm. The anemotaxis method (Algorithm C3) is another CPT algorithm that benefits from the emergent collaboration in the swarm. Please note that our "baseline" anemotaxis implementation does not share explicitly any information between the neighboring agents, nor does it attempt to keep a history of the vehicle's locations or plume observations. But it is still able to achieve similar functionality due to its operation within the lattice formation. Once again, the formation acts as a distributed sensor fusion network, performing implicit averaging and filtering of the fluid flow observations throughout the entire swarm.

4.7.4 Fluxotaxis algorithm. In the introduction to this second CPT study, we mentioned the fact that a large-scale swarm implementation requires a fluxotaxis design that is somewhat different from the version of the fluxotaxis algorithm we employed in the first study with a seven robot hexagonal lattice. Before, we had explicit control over the lattice radius, and it was very straightforward to compute the boundaries of the virtual flux surface. However, for a large swarm, repeating the same steps leads to an unnecessary coupling between the agents which we wish to avoid. Again, physicomimetics inspired our new design, where all swarm control operations are based on local inter-vehicle forces. Since the physicomimetic virtual forces are modeled after real conservative forces, such as gravity, there exists a scalar potential function, $U(\vec{r})$, so that $\vec{F}_{\text{formation}} = -\nabla U$. The important idea to keep in mind is that physicomimetics does not compute the potential function U – only the forces matter. However, the function U is useful for the long-term, predictive analysis of the physicomimetic systems. In particular, note that the agents in the swarm implicitly create the virtual potentials to which they respond by forming the lattice, similarly to how planetary systems maintain their orbital stability as the result of their own mass. This is an elegant concept, from both practical and theoretic perspectives, and we employed a similar methodology for improving our fluxotaxis algorithm.

Starting with the basic theory for the *GDMF* metric (Section 2.4), we designed the second version of the fluxotaxis algorithm to control a team of physicomimetic driven robots. Fluxotaxis requires that the vehicles be able to obtain point measurements of the chemical concentration, ρ, and ambient wind velocity, $\vec{V}$. The vehicles then communicate this information to their local neighboring robots so that all of the CPT agents can calculate the chemical mass flux, $\rho\vec{V}$. Revisiting the important concepts of influx and outflux first defined in Section 4.6.4, we must construct a set of virtual surfaces in order to compute the local surface fluxes that can be used to measure the volumetric DMF in the vicinity of the CPT robot. The basic operation behind this procedure is for each agent to compute the product $\rho_n \vec{v}_n$, where ρ_n is the chemical density measurement collected and reported to the agent by neighbor n, and $\vec{v}_n$ is the component of wind velocity $\vec{V}_n$ (observed by the neighbor) projected onto the neighbor line that joins the centers of the two agents. Figure 5 shows this situation from the perspective of agent A_0 in the center of the diagram.

The virtual surface, displayed as a solid line through the circular body of a CPT vehicle in Figure 5, across which the neighbor mass flux, $\rho_n \vec{v}_n$, flows is defined as a planar region that is perpendicular to the (dotted) neighbor line between the robot pairs. The orientation of this virtual patch with respect to the A_0 agent determines the amount of chemical mass flux that is associated with the position of each A_n neighbor vehicle. Depending on whether the flux is toward or away from A_0, the agent will record the corresponding influx or outflux. For the geometry and the airflow pattern shown in

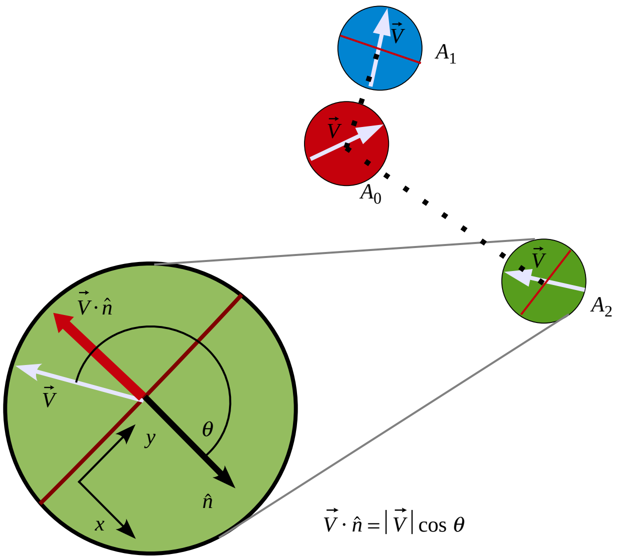


Figure 5.
Structure and position of the virtual surfaces (shown with a solid red line across the circular body of agents A_1 and A_2) in the fluxotaxis implementation for an arbitrary-sized swarm

Figure 5, from the viewpoint of robot A_0, the position of A_1 represents an outflux of $\rho_1 \vec{v}_1$, and the position of A_2 contains an influx of $\rho_2 \vec{v}_2$. The robot's on-board procedure for determining the magnitude and the classification of the chemical mass flux at a neighbor's location is listed in Algorithm C4.

Because the local plume density measurement, ρ_n, is a scalar value, neighboring vehicles can simply exchange their individual chemical sensor readings. The software implementation assumes that the robots can also share vector values, such as anemometer readings of wind speed and direction. This too is a trivial data exchange if the swarm shares a common frame of reference; however, standard coordinate transformations can be applied in a direct manner for cases where only relative orientation between the neighbors is known (Spears *et al.*, 2006). To obtain $\vec{v}_n$, each robot must compute the dot product, $\vec{V}_n \cdot \hat{n}$, of the airflow velocity, $\vec{V}_n$, and the unit normal vector for the virtual surface patch, $\hat{n}$. This is a straightforward computation, because each virtual surface patch is defined in terms of the neighbor line, that is itself simply a translation vector between the centers of the two neighbors, so that the unit normal vector, $\hat{n}$, can be computed from the information about the relative positions of the vehicles, which is readily available from the on-board physicomimetic control software. Finally, the familiar geometric identity, $\vec{A} \cdot \vec{B} = |A||B|\cos(\theta)$ is applied to calculate the $\vec{V}_n \cdot \hat{n}$ dot product.

The dot product method we just described allows fluxotaxis agents to distinguish between the outgoing and incoming chemical mass fluxes, and similarly to our earlier fluxotaxis implementation from Section 4.6.4, this version of the fluxotaxis CPT algorithm first attempts to move the robots in the direction of the neighbor who reported the maximum incoming flux, if one exists. Otherwise, if none of the neighbor fluxes are "influxes," the algorithm will search for a position that reported a maximum outward flux. We found that if we let the robot lattice act as a control surface in order to compute the divergence of mass flux, then the amount of influx will gradually decrease as we approach the chemical source, up to the point where the robots surround the emitter, after which the total integrated mass flux over the boundaries of the swarm will appear as a positive outflux (Thayer, 2008). An interesting observation about this formulation of the algorithm is that by selecting a location with the maximum incoming or outgoing flux, each robot is actually computing a gradient of mass flux. Since all of the robots interact with each other through the physicomimetic control mechanism, the trajectory of the whole formation is the implicit sum of all of these individual goal force vectors, as explained in Section 4.1. This implicit sum is in fact a discrete approximation of the chemical mass flux divergence in the vicinity of the robots. Thus, the preservation of individual formation bonds inside a physicomimetic swarm leads to an implicit computation of the "source" or a "sink" of the mass flux gradient. Effectively, when executing this modified _version of fluxotaxis, the swarm computes a first-order approximation of the *GDMF* metric defined in Section 2.4, and the quality of this approximation improves as the swarm grows in size (Zarzhitsky *et al.*, 2005a). In the current implementation, the algorithm deals with cases of zero mass flux or absence of nearby teammates by delegating to the chemotaxis and anemotaxis strategies, respectively.

5. CPT study of seven robots in a hexagonal lattice

This first study uses a hexagonal lattice of seven robots. Here, we demonstrate how to use physicomimetics to navigate the robot formation toward the chemical source. Since the number of agents is fixed, the swarm control parameters are selected a priori to the deployment. The study consists of two experiments: the first one is based on obstacle-free environments and looks at the effect of the lattice radius R on the CPT performance; the second experiment uses the optimal control parameters found in the first experiment and introduces obstacles into the search area. In both experiments, chemotaxis, anemotaxis, and fluxotaxis algorithms are evaluated and compared with respect to several different performance metrics, as explained below.

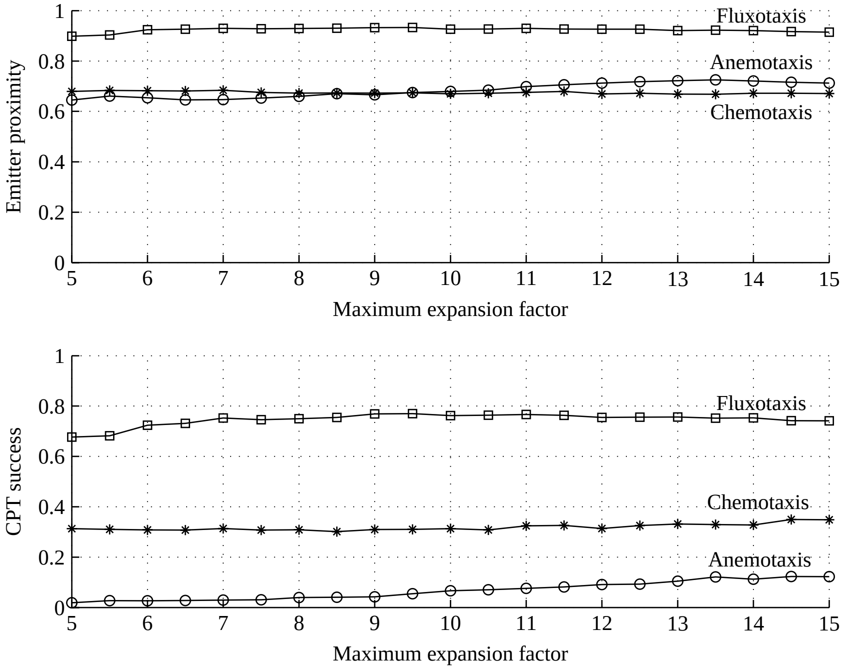
5.1 Experiment in an unobstructed environment

5.1.1 Purpose. The goal of this first CPT study is to understand the comparative performance of three different CPT algorithms as the maximum lattice expansion factor is methodically varied. Here, the maximum lattice expansion factor is defined as the largest separation distance between any two vehicles. The physicomimetic control framework has several parameters that determine key behaviors of the lattice, and we are interested in the effect that the lattice radius, R, has on the comparative CPT task performance of each plume-tracing strategy.

5.1.2 Setup and methodology. The test configuration for this experiment covered a representative mix of laminar, transitional, and turbulent flow regimes, each containing a dynamic chemical-gas plume evolving over a 930 m^2 area. All CPT algorithms were pairwise compared over consistent plume conditions, meaning that all forward solution parameters, as well as the lattice initial state were matched for each CPT strategy run, i.e. we were extremely careful to ensure fair and consistent evaluation of each CPT algorithm. Note that the plume evolution and lattice movement are concurrent, i.e. the plume is changing during the tracing. Each CPT run lasted 3,000 time steps, simulating approximately an hour of plume time. Lattice movement is determined using a set of waypoints spaced one meter apart. Since only the fluxotaxis algorithm includes a control output for the lattice radius, in order to keep the experimental conditions as similar as possible, anemotaxis and chemotaxis driven lattices were allowed to expand or contract their radii at random to increase exploration. The initial size of the lattice radius was fixed at 0.5 m, and the maximum expansion factor was set at 15, which means that the maximum lattice diameter was 7.5 m. We selected 35 plume configurations and ten expansion factors, and then evaluated each combination of the plume and expansion factor in a separate CPT scenario. For each plume and expansion factor setting, we selected 200 random locations within the environment where the lattice started the tracing.

5.1.3 Performance metrics. The evaluation metric consists of two related components: the proximity of the center vehicle to the true location of the chemical emitter, and a Boolean measure of emitter containment (i.e. whether the chemical source is inside the lattice) that we called a CPT success. Both metrics are normalized with respect to the "optimal" value, with 1.0 being the best, and are calculated at the conclusion of the trial with a global observer function. Note that the second metric indirectly measures the impact of the maximum lattice radius expansion factor: a larger radius implies a higher likelihood of a CPT success.

5.1.4 Results. The simulation results are plotted in Figure 6, showing the average performance of each CPT algorithm over all plumes with respect to the maximum lattice expansion factor. The experiment showed that on the proximity metric, a higher expansion factor allows anemotaxis to beat chemotaxis. This is due to a characteristic



Notes: Data are averaged over 35 plumes with 200 random starting locations; the y-axis is normalized with respect to dimensionless units

Figure 6. Performance of the three CPT algorithms on a lattice of seven robots as a function of the varying expansion factors

oscillation in the anemotaxis lattice at a larger radius: the lattice moves upwind to get nearer to the emitter, but moves past it, and then exits out of the chemical plume; at this point, it switches to casting, which causes the lattice to reverse its direction and move back into the plume, where the anemotaxis upwind-following behavior activates again, thus resulting in a cycle near the emitter. As mentioned earlier, due to the increased likelihood of the robots surrounding the emitter when the lattice is allowed to expand, the CPT success rate for all of the algorithms improves slightly as the expansion factor is increased. Performance averages obtained in this experiment are summarized in Table I. A Wilcoxon rank sum test indicated a statistically significant difference at the $p < 0.05$ confidence level in the fluxotaxis performance results as compared to those of chemo- and anemotaxis.

5.1.5 Conclusions. The plots in Figure 6 and the data in Table I support our claims regarding the advantage of employing the fluxotaxis strategy over the alternative algorithms – fluxotaxis shows a consistent improvement in being able to locate the emitter. The data show that at the end of the mission, it located and surrounded the chemical source in 75 percent of the trials, as compared to its closest competitor chemotaxis, which only contained the source in 32 percent of the trials. Likewise, it achieved a proximity rating of 92 percent, compared to the 68 percent for the anemotaxis technique.

5.2 Experiment in an obstructed environment

5.2.1 Purpose. This study looks at how different CPT algorithms perform when obstacles are introduced into the environment. Our aim is to understand how the three different plume-tracing strategies operate under varying environmental conditions.

5.2.2 Setup and methodology. In the previous experiment, in which we studied CPT performance as a function of the maximum lattice expansion radius, we demonstrated that the physicomimetics framework supports a large range of lattice radii. In this experiment, we fixed inter-vehicle spacing to 3 m, and focused on two different aspects of the mission: the number of obstacles and the starting position of the lattice. For each CPT trial, we generated 500 random vehicle starting locations, which we selected from a uniform distribution of positions mapped into an emitter-centered polar coordinate system. The initial lattice distance from the emitter varied from 0 to 25 m; the upper bound of 25 m helped minimize the duration of casting, and allowed us to focus on the behavior of the CPT algorithms instead.

We evaluated the effect of obstacles on CPT algorithms by solving fluid dynamic equations in varied environments with the following configurations: no obstacles, one 3 × 3 m obstacle (1 percent obstacle coverage), four 1.5 × 1.5 m obstacles (1 percent coverage), two 3 × 3 m obstacles (2 percent coverage), and with eight 1.5 × 1.5 m

Table I.
CPT algorithm performance on a lattice of seven robots, showing mean ± standard deviation

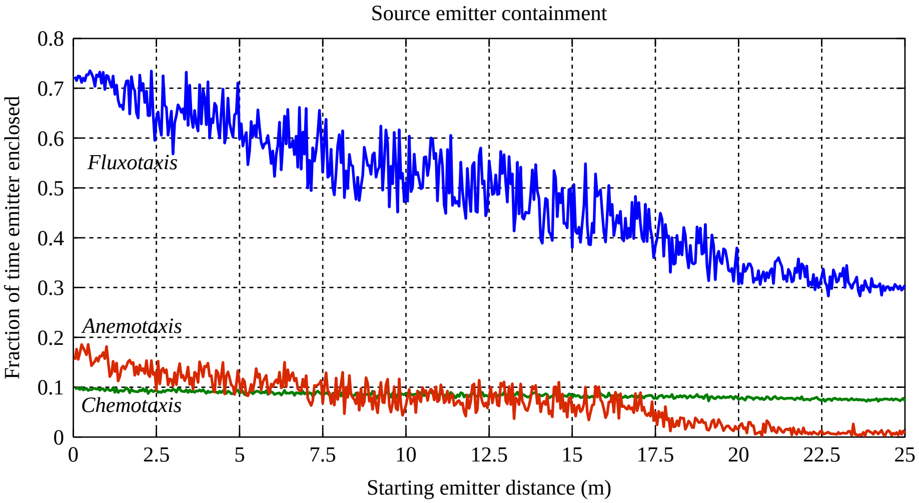
CPT algorithm	Emitter proximity	Localization success
Anemotaxis	0.6843 ± 0.0291	0.0667 ± 0.0362
Chemotaxis	0.6745 ± 0.0049	0.3184 ± 0.0132
Fluxotaxis	0.9235 ± 0.0089	0.7460 ± 0.0250

Notes: Data are averaged over 35 plumes and all parameter variations; higher values indicate better performance obstacles (2 percent coverage), with a total of 150 different plume and obstacle configurations. CFD parameters were the same for each obstacle course, but the chemical plumes in each new environment varied due to the size and placement of the obstacles. As before, all of the experimental conditions were carefully controlled to ensure consistent and unbiased evaluation of each CPT method. Since chemical contamination (e.g. a toxic spill) typically precedes the start of CPT efforts, we "turned on" the chemical emitter for 3,000 time steps (about an hour of real plume time) before activating the CPT robots. Each plume then evolved for an additional 7,000 steps as the CPT robots traced the chemical cloud to its source.

5.2.3 Performance metrics. For this experiment, we studied emitter localization characteristics of each CPT method as a function of the initial distance between the lattice and the chemical source. Therefore, this second portion of the seven-robot study required a new performance metric to indicate how well a given CPT strategy localizes the source. We called this a stability metric to differentiate it from the end-of-the-run CPT success measure used in the previous experiment. This criterion is a helpful evaluation tool because it measures consistency of the source localization solution obtained with each CPT algorithm. To measure this benchmark in simulation, we again used a global observer function, which computed the fraction of simulation time the plume emitter was contained inside the lattice.

5.2.4 Results. Results of this test are shown in Figure 7, showing that fluxotaxis achieves an average emitter containment rate of 49.1 percent, which is significantly higher than that of anemotaxis (the mean of 8.4 percent) and chemotaxis (the mean of 7.2 percent).

Since anemotaxis always moves upwind in the plume, it often drives past the emitter, and then spends over 90 percent of its mission in a cycle of first moving upstream in the plume, and then switching to casting after a subsequent miss of the emitter.



Note: The x-axis is the distance between the initial starting location of the lattice and the emitter, and the y-axis is the fraction of total CPT time (7,000 simulation steps) when the vehicles successfully surrounded the emitter

Figure 7. CPT algorithm performance results averaged over 150 plume and obstacle configurations

This explains why the anemotaxis performance is low even when it begins its tracing next to the emitter, and why the performance curve deteriorates with the increasing starting distance to the emitter. Chemotaxis does somewhat better when it starts out near the emitter, but its performance falls off rapidly, caused by the increase in the number of obstacle-induced local density maxima. However, even when chemotaxis begins its tracing within ten feet of the emitter, the average containment in this case is still only 14.3 percent, which means the lattice fails to find the source more than 85 percent of the time. The poor chemotaxis performance is caused by periodic turbulence and the resulting variance (e.g. wind gusts) in the flow, both of which give rise to a "shedding" effect, that manifests itself when large "chunks" of the plume in the vicinity of the emitter are "torn off" and carried away from the source. A Wilcoxon rank sum test applied to each experiment showed that the CPT performance improvement obtained with fluxotaxis is statistically significant in 94 percent of the experiments at $p = 0.01$ confidence level.

5.2.5 Conclusions. Fluxotaxis consistently and significantly outperformed chemotaxis and anemotaxis algorithms on the emitter containment metric. This experimental outcome is consistent with the theoretic predictions of the source and sink theorems we introduced in Section 2.4. Turbulence is the reason why fluxotaxis does not achieve a 100 percent emitter containment when it starts with the emitter already enclosed, because periodic wind gusts transport large portions of the ejected chemical away from the emitter, and the moving mass of chemical appears as a temporary pseudo-source. Future work on improving the fluxotaxis algorithm will address this detrimental impact of turbulence. In addition, we found that the CPT algorithms usually manage to navigate around obstacles well before active collision avoidance even becomes necessary. The obstacle avoidance problem is often simpler within a chemical plume, since the lattice follows the plume as the carrier fluid flows around the obstacles. This confirms the observations of Keymeulen and Decuyper (1994a), who found that using the direction of fluid flow is an efficient and successful strategy for navigating out of maze-like areas. This is an important and relevant observation, since CPT systems are deployed in hazardous areas, typically littered with debris inside unmapped passageways.

6. CPT study of a large decentralized swarm

Section 5 described our first CPT simulation study, which used a fixed-size lattice of seven vehicles. The results were definitive – our fluxotaxis approach is an improvement over both the chemotaxis and anemotaxis CPT methods in terms of being able to find the emitter (what we previously called a CPT success), and in being able to consistently contain the chemical source within the bounds of the seven vehicle hexagonal formation (a property we called an emitter containment). Of course, one notable limitation of the previous implementation is its dependence on a particular lattice configuration – all of the performance metrics, as well as the CPT algorithms themselves, assumed that the "swarm" consists of exactly seven robots, and the vehicles had to maintain the hexagonal formation at all times in order for the experiments to be valid. Therefore, we carried out a follow up CPT study to answer an important question: does fluxotaxis scale to an arbitrary-sized swarm? Experimental results we present in this section show that fluxotaxis retains its CPT advantage across a wide range of swarm sizes and environment conditions.

In this study, we rejected any a priori knowledge regarding the number of agents participating in the plume-tracing task, along with any restrictions on the swarm topology. This is a practical requirement, because if we consider the logistical concerns of operating a large number of robotic vehicles out in the field, we can no longer justify our earlier assumption of the swarm layout.

6.1 Experiment with increasing number of obstacles

6.1.1 Purpose. For the first experiment in this study, we measured performance of modified, swarm-oriented, fully decentralized CPT algorithms on different plume environments with many different obstacles. As before, our motivation here is to demonstrate how the three different approaches to chemical source localization compare against each other across a range of plume and search area conditions.

6.1.2 Setup and methodology. We simulated emitter localization by swarms controlled by random casting, chemotaxis, anemotaxis, and the fluxotaxis algorithms on a suite of 81 simulated plume scenarios with physically distinct flow configurations, each containing an airborne chemical plume evolving over a large 8,360 m^2 area. As in the previous study, we picked a range of CFD boundary conditions that produced an even mix of laminar, transitional, and turbulent flows in randomly created environments with: no obstacles, with nine, eighteen, twenty-seven, thirty-six 1.5 × 1.5 m obstacles, and with two, four, seven, and nine 3 × 3 m obstacles.

The trace chemical was ejected for 3,000 simulation steps (about an hour of real plume time) before a swarm of CPT robots was first deployed, and the plume tracing mission lasted for an additional 7,000 steps (corresponding to a realistic two-hour CPT time frame). The initial swarm starting location varied from a position precisely over the emitter to 60 m away from the emitter in 1-m increments (compare this with the 25 m maximum we examined as part of the first study in Section 5.2). We varied the number of vehicles in the swarm from seven to 70 robots, with a total of ten different swarm sizes per plume, obstacle, and initial starting location combination. Thus, a total of 40,500 CPT evaluation runs were performed as part of this experiment. For every new evaluation run, we made sure that the plume and the search area configuration were the same for all CPT strategies, thus all observed differences in the swarm's performance are the result of the different navigation paths computed by the individual strategies.

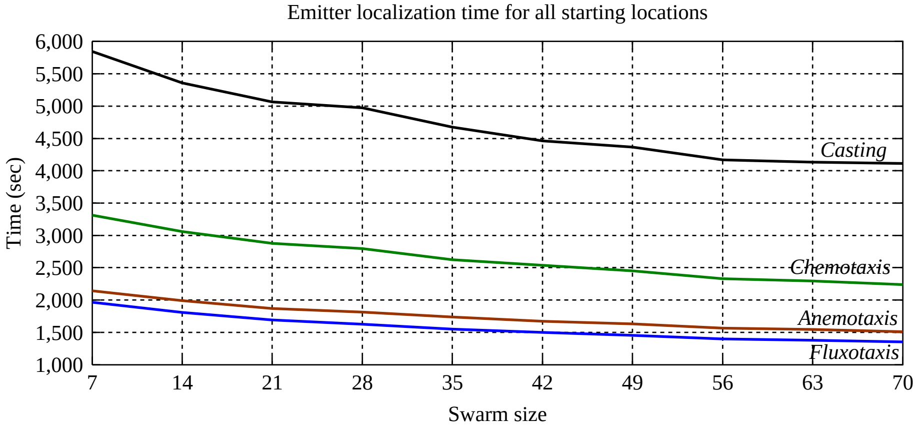
6.1.3 Performance metrics. In the first CPT study, we took advantage of the fact that the hexagonal lattice of seven robots would surround the chemical source in a very predictable manner, and all of our CPT performance metrics were based on this assumption. However, for this study, we have very little a priori knowledge of how the swarm will approach the emitter. Because each CPT vehicle functions as a completely independent entity, the old performance metric of emitter proximity based on the "center" robot in the lattice no longer makes sense, since there is no "center" in a decentralized swarm. Likewise, the containment metric needs to be adapted for very large swarms, otherwise we cannot measure the CPT performance as a function of swarm size.

Therefore, we developed two new CPT performance metrics that evaluate the swarm aspect of our physicomimetic solution. The first metric, called the arrival time, is designed to evaluate the speed with which the swarm locates the source. The arrival time value is equal to the earliest time step of when a vehicle from the swarm first drives within the sensor range, R_s, of the chemical emitter, and lower values indicate better CPT performance. Note that we do not assume that the vehicle somehow knew that it passed close to the emitter, just that it physically came within a short distance of the chemical source. Because in order to "succeed" on the arrival time metric the vehicle only needs to be near the emitter, we also evaluated the casting strategy as one of the CPT "algorithms." Because casting is a random method, this comparison provides us with the baseline of the performance level that can be achieved via an uninformed search of the environment. We expect all of the CPT strategies to do better than just a random search of the simulated world.

The second metric, which we call the emitter localization frequency or localization count is a measure of how many vehicles drove within the sensor range, R_s, of the emitter. This is a cumulative metric – it simply sums up the number of vehicles located within the circle of radius R_s for each time step in the simulation. We again use the score of our random casting strategy as the baseline for this characteristic, with the expectation that all three of our CPT algorithms will perform better than the random search.

6.1.4 Results. Results for the arrival time performance metric (i.e. the elapsed time before the first vehicle detects the emitter) are plotted in Figure 8, and are broken down by the type of obstacle course in Table II. The number of times that the chemical emitter was successfully localized by swarm vehicles is shown in Figure 9, with the performance break down based on obstacle coverage presented in Table III.

Our first observation from the two figures is that only fluxotaxis performs well on both metrics: it combines the speed performance demonstrated by anemotaxis with the localization frequency advantage shown by chemotaxis. Examining the performance data in Figure 9, we conclude that CPT is inherently a swarm application. We base this

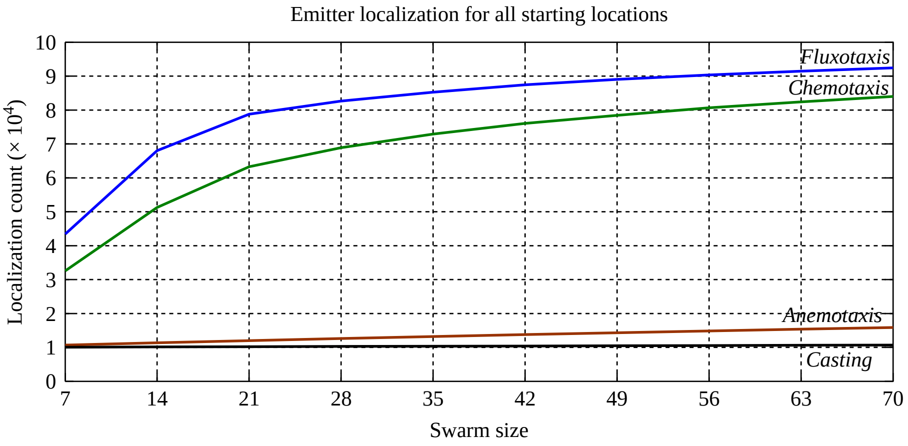


Note: Lower values indicate better performance

Figure 8.
Arrival time results for each CPT algorithm averaged over 81 plumes and 50 starting locations

Table II.
Arrival time metric of each CPT algorithm versus the world obstacle coverage, averaged over ten swarm sizes, 50 starting locations, and 81 plumes

	Obstacle coverage				
CPT algorithm	0.0%	0.25%	0.50%	0.75%	1.0%
Fluxotaxis	1,508	1,523	1,615	1,572	1,610
Anemotaxis	1,885	1,760	1,725	1,700	1,731
Chemotaxis	2,451	2,603	2,887	2,437	2,778
Casting	5,292	4,848	4,739	4,498	4,490



Note: Larger values indicate higher performance

Figure 9. Frequency of emitter localization by the swarm, averaged over 81 plumes and 50 starting locations

Table III. Cumulative count of emitter detections by the swarm for each obstacle coverage category, averaged over ten swarm sizes, 50 starting locations, and 81 plumes

	Obstacle coverage				
CPT algorithm	0.0%	0.25%	0.50%	0.75%	1.0%
Fluxotaxis	73,253	72,189	70,085	70,657	69,369
Chemotaxis	62,925	59,398	55,802	62,649	56,368
Anemotaxis	3,604	3,667	3,479	3,429	2,939
Casting	368	308	330	593	271

claim on the improvement in the consistency of the emitter localization, as captured by the localization frequency metric, for the fluxotaxis and chemotaxis algorithms observed as the result of the increasing swarm size. However, the poor emitter localization performance of the random casting algorithm suggests that the robot collective must be managed in order to realize the performance benefit of the swarm approach – simply adding more vehicles to the group is not sufficient to improve its performance.

The large difference in performance between fluxotaxis and anemotaxis, seen in Figure 9, can be attributed in part to the fact that the baseline implementation of the anemotaxis algorithm does not have a bias to direct the vehicles toward the emitter – instead, the upwind follower reaches the emitter in approximately the same time as does fluxotaxis (which can be inferred from Figure 8), but the swarm continues to move upstream, and thus moves past the emitter. After losing the emitter, the anemotaxis swarm switches to the casting behavior in order to find the plume, which is why its localization performance is similar to that of the casting strategy. In the literature, several heuristic methods have been suggested to address this problem. However, we view the fluxotaxis approach of seeking out the chemical mass flux as a superior method, because our physics-based approach does not require additional heuristics to achieve CPT goals.

Data in Tables II and III provide the evidence for the robustness and scalability of the physicomimetics framework, which are manifested in the framework's ability to adapt in a dynamic environment. The force law parameters are the same in each experiment, but the control software manages the swarm with good proficiency regardless of the number of agents or the size and number of obstacles.

6.1.5 Conclusions. The overall CPT performance of each algorithm in this experiment must be interpreted in the context of the type of plume information that is extracted and then acted upon by the given CPT strategy. In other words, the variations in the rates of success for each algorithm are due to the natural characteristics of each CPT approach. For instance, for the random casting strategy, the arrival time result improves with the increasing obstacle coverage because greater obstacle coverage implies less open area that the random searcher has to explore. On the other hand, performance of the chemotaxis strategy on both metrics is generally worse in the environments with more obstacles. This is a direct consequence of the fact that obstacles create local maxima in the density profile of the plume, because the chemical has a tendency to slow down and build up in the flow that impinges on the obstacles' edges. From the mathematical perspective, each obstacle induces a local chemical sink, and these sinks mislead the naïve chemical gradient follower. This behavior is the direct consequence of the chemotaxis algorithm's reliance on first-order derivative of the chemical concentration, which does not contain enough information to distinguish a true chemical source from an obstacle induced sink. The sink theorem we introduced in Section 2.4 provides a precise mathematical formulation for describing and detecting such regions with high chemical concentration that do not contain the emitter. The time spent by chemotaxis investigating these temporary "pseudo-sources" is reflected in the algorithm's increased localization time, and it also shortens the time that the chemotaxis agents spent in the vicinity of the true chemical source. However, when operating near the emitter, chemotaxis is expected to perform well on the localization frequency metric, since the peaks in the chemical concentration landscape are in fact close to the real source (see the source theorem in Section 2.4). The fact that fluxotaxis consistently outperforms the other two CPT methods on both performance metrics is a straightforward validation of our argument in Section 4.6.4, and equation (6) shows how the fluxotaxis algorithm can automatically select the best plume characteristic (i.e. the density gradient or the flow direction) to follow in accordance with the changing environmental conditions.

6.2 Experiment with increasing swarm size

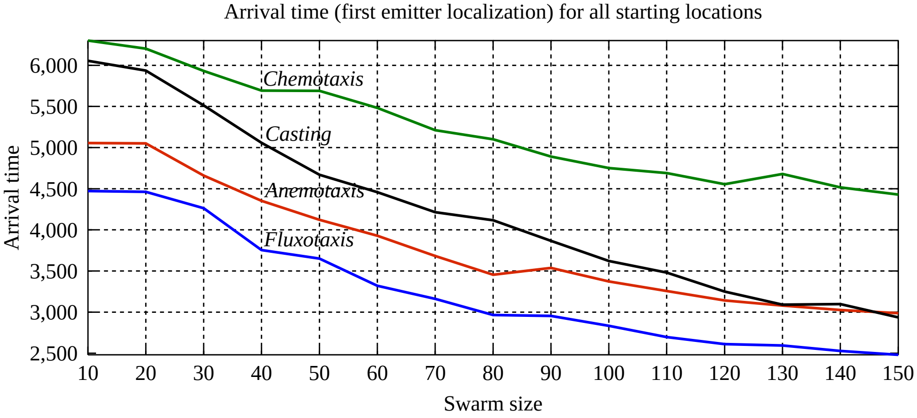
6.2.1 Purpose. Results of the previous experiment showed that the CPT performance is affected by the size of the swarm to a much greater degree than it is influenced by the obstacle configuration. Therefore, in this final experiment, we increased the number of CPT robots in the swarm to determine what performance gain can be realized with very large swarms.

6.2.2 Setup and methodology. We simulated ten different flow conditions, with the chemical emitter located inside a 8,360 m^2 region. As before, a choice of appropriate boundary conditions resulted in a diverse mixture of airflow patterns. Each plume-tracing area contained ten 1.5×1.5 m randomly placed obstacles. Similarly to our previous experiments, the chemical emitter activated 3,000 simulations steps (an hour of real plume time) before the CPT swarm deployed. We advanced each plume for 7,000 steps (corresponding to a two-hour time frame), and recorded the emitter arrival time and localization statistics for each CPT algorithm.

All of the data in this experiment come from matching evaluation runs for each CPT algorithm and the casting strategy on a large set of CPT scenarios, consisting of 15 swarm sizes (ranging from ten to 150 vehicles) and 30 different starting locations per each swarm size. The initial position of the swarm is selected at random, with the starting distances ranging from 0 to 60 m away from the emitter. As was the case in all of the other experiments, we made sure that the chemical plume, the search environment, and the evaluation criteria were identical for matching comparisons of the CPT strategies' tracings.

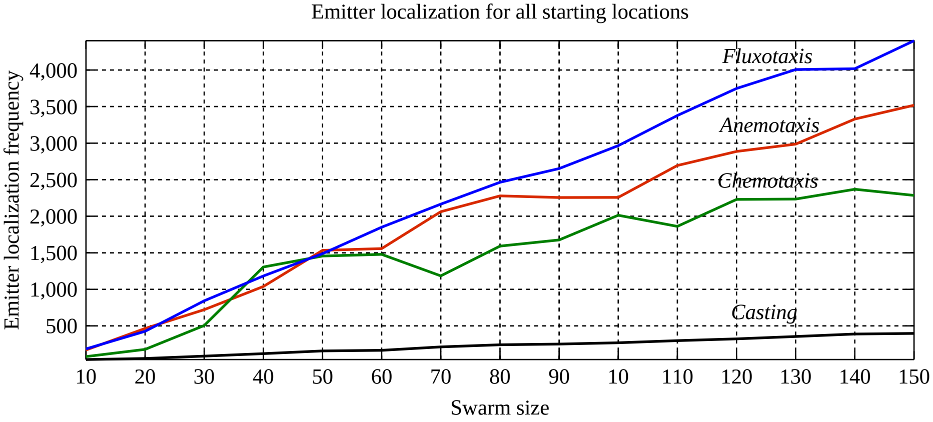
6.2.3 Performance metrics. Swarm performance in this experiment was evaluated using the same arrival time and localization frequency metrics first defined in Section 6.1.3.

6.2.4 Results. The performance of each CPT strategy as a function of the number of robots is shown in Figures 10 and 11. Table IV lists the cumulative performance average for each CPT method. Results of this experiment confirm that an increase in the size of



Note: Arrival event occurs when any swarm vehicle makes the first sensor contact with the emitter (as determined by a global observer)

Figure 10. Arrival time metric for each CPT algorithm over a range of swarm sizes (smaller values indicate faster localization)



Note: The emitter is considered to have been localized when a swarm vehicle makes sensor contact with the emitter, as determined by a global evaluation algorithm

Figure 11. Cumulative count of chemical emitter localizations by each CPT algorithm

the swarm improves both the speed and accuracy of the source localization. Each algorithm we tested displayed improved performance on both evaluation metrics, and we want to point out that the fluxotaxis performance curves clearly show the algorithm's ability to satisfy the CPT objectives in a stable and predictable manner. At the same time, note that chemotaxis is the most vulnerable of the CPT techniques, so that even the uninformed (i.e. random) world exploration yields better arrival time results than the greedy gradient-following chemotaxis. The chemotaxis-driven swarm is frequently misled by the local concentration maxima around obstacles, thus lowering its performance on the localization time metric. The anemotaxis approach initially outperforms simple casting in terms of the arrival time, but this advantage decreases as the swarm size increases. This finding can be explained by the fact that instantaneous wind velocity is generally a poor indicator of the true direction of the plume's source, and as the fluid periodically changes the flow direction, the anemotaxis strategy is misled by the isolated pockets of a fragmented plume, resulting in a time-inefficient zigzag plume traversal pattern often observed in insects (Grasso, 2001). This is an important observation because it shows that just adding more sensors and robots into the swarm is not enough, and once again we conclude that the sensors must be managed in an informed and theory-guided fashion. Given the decentralized swarm architecture, each team member must provide the means to facilitate this emergent sensor processing ability. Our fluxotaxis algorithm provides a practical example of how such functionality can be engineered into a complex system.

6.2.5 Conclusions. Our physicomimetic implementation provides a mathematical assurance for the swarm's long-term performance. The local nature of vehicle interactions results in efficient scalability, and the reactive control is what allows the swarm to succeed regardless of the environment configuration and initial conditions. Because both the fluxotaxis CPT algorithm and the swarm control framework have a rigorous mathematical foundation, fluxotaxis consistently outperforms the biomimetic methods. This experiment further supports our claims regarding the robustness and scalability of the physicomimetics control framework. The virtual forces controlling the swarm formation are identical for all of the experiments, even as the number of robots increases by an order of magnitude. This fact emphasizes the important role that a decentralized approach and reliance on strictly local information play in the design of stable and predictable autonomous systems (Spears *et al.*, 2005b).

7. Summary and future work

Our design of the fluxotaxis CPT algorithm illustrates a robust, theory-guided approach for accomplishing complex tasks with autonomous robots by intelligently exploiting the underlying physical principles of the problem. Our simulation experiments demonstrate that fluxotaxis is able to compute correct navigation waypoints using local sensor observations in a way that is superior to the most popular biologically inspired

Table IV.
Average performance of the three CPT algorithms over 30 starting locations, 15 swarm sizes, and ten plumes

CPT algorithm	Arrival time	Localization frequency
Casting	4,224.4	224.1
Chemotaxis	5,208.5	1,496.5
Anemotaxis	3,780.5	1,983.6
Fluxotaxis	3,249.9	2,384.8

plume-tracing methods. In addition, all three team-oriented CPT strategies we evaluated showed a gain in performance due to the cooperation between neighboring vehicles. This is a unique emergent property of the swarm supported by the automatic sensor aggregation feature of the physicomimetics control framework. By sharing the information about local flow conditions between neighboring vehicles in the group, each team member is able to construct a more accurate view of the surrounding plume, which in turn improves the accuracy of the emitter search algorithm. However, since fluxotaxis is founded on insights from fluid mechanics, the physics-based foundation allows this new algorithm to be consistently more effective in achieving the CPT goals than what is possible with the biomimetic approaches.

Drawing on the findings of our comprehensive study, we confidently conclude that CPT is inherently a swarm application, which means that significant gains in performance, such as reductions in the required search time and increased consistency of the localization estimate are realized in a predictable manner as the size of the swarm is increased. We also showed that these improvements in the CPT performance require a scalable algorithm; in other words, the CPT algorithm must manage the information flow in an efficient and intelligent manner. Our experiments revealed that a class of single-robot-oriented CPT algorithms like anemotaxis, which do not explicitly take advantage of the collaborative functionality of the swarm platform, can realize only small increases in the CPT performance when used on large-sized swarms. This discovery further reinforces the motivation for our dedicated effort of designing a new CPT algorithm specifically for swarms – by making sure that our fluxotaxis approach utilizes all of the cooperative mechanisms offered by the swarm implementation, we maximize the practical benefit of using the swarm platform for the chemical source localization problem.

The fact that our fluxotaxis algorithm realizes this increase in CPT performance in a fully emergent fashion speaks to the power and the flexibility of the physicomimetic design. First, we took advantage of the existing fluid dynamics understanding to construct the fluxotaxis algorithm, which allowed us to address significant gaps in the current state-of-the-art CPT research regarding emitter identification in obstacle-filled environments with unsteady, turbulent flows (Spears *et al.*, 2009). Next we employed the physicomimetic swarm control methodology to build a massively parallel, distributed computer out of simple, inexpensive robots with limited on-board capabilities. The resulting sensor and computation "mesh" is ideally suited for a variety of *in situ* analyses and monitoring activities that benefit from the robust self-organizing behavior of the grid-like robot formations. For the CPT problem, we identified the physical property of chemical mass flux flow as the crucial indicator of the location of the chemical source, and showed how a mathematically derived technique is implemented on simulated mobile robots. Finally, we demonstrated that this methodical, step-by-step construction of all the key components of our implementation resulted in a distributed system with predictable long-term behavior.

We showed that the physics-based fluxotaxis plume-tracing strategy combines the strengths of the popular chemo- and anemotaxis approaches, and outperforms these two biomimetic methods in terms of search time and localization accuracy. However, our experiments have also identified several problem areas where additional work is needed. One such area is discussed in Section 5.2, where we noted that air turbulence creates transient concentrations of the chemical that may appear as temporary sources to the plume tracing robots. We plan to devise mitigation strategies for transient effects present in turbulent flows, and to compare the performance of our fluxotaxis algorithm against some of the recently proposed infotaxis CPT methods as part of our future work. Another planned improvement is a more accurate modeling of the chemical flow that takes the movement of the vehicles into account. The current formulation of fluxotaxis assumes that both the chemical concentration and airflow sensors can provide an accurate, instantaneous measurement of the plume. However, in practice, issues such as sensor noise and flow occlusions will violate this assumption, and will require additional logic from the fluxotaxis algorithm to account for these artifacts. Other challenging plume-tracing problems we plan to address in the near future include multiple chemical sources and mobile emitter(s). Given the gains in fluxotaxis performance realized through an increase of the vehicle fleet size, we feel that our physicomimetic, swarm-centric approach will be effective in addressing these extended CPT scenarios.

Finally, we want to emphasize that the results discussed here are not limited to physical models and simulations, but in fact have already been tested in the laboratory setting with small-scale robotic vehicles. To date, our experiments focused on volatile organic compounds, and we have carried out a large number of wind tunnel experiments to study behavior of ethanol plumes in a variety of flow conditions. Concurrent with our physical modeling and chemical experimental work, we have also been developing a rugged, outdoor-ready vehicle platform that can support further investigation of the concepts and ideas presented in this paper.

References

Balch, T. and Arkin, R. (1998), "Behavior-based formation control for multi-robot teams", *IEEE Transactions on Robotics and Automation*, Vol. 14, pp. 1-15.

Balkovsky, E. and Shraiman, B. (2002), "Olfactory search at high Reynolds number", *PNAS*, Vol. 99 No. 20, pp. 12589-93.

Cowen, E. (2002), *Environmental Fluid Mechanics*, Vol. 2, Kluwer, Boston, MA (special issue on Chemical Plume Tracing).

Cui, X., Hardin, C.T., Ragade, R.K. and Elmaghraby, A.S. (2004), "A swarm approach for emission sources localization", *Proceedings of the 16th IEEE International Conference on Tools with Artificial Intelligence* (*ICTAI'04*)*, Boca Raton, FL*, pp. 424-30.

Decuyper, J. and Keymeulen, D. (1991), "A reactive robot navigation system based on a fluid dynamics metaphor", *Parallel Problem Solving from Nature* (*PPSN I*), Lecture Notes in Computer Science, Vol. 496, Springer, Berlin, pp. 356-62.

Faires, J.D. and Burden, R. (2003), *Numerical Methods*, Brooks/Cole – Thomson Learning, Pacific Grove, CA.

Farrell, J.A., Murlis, J., Li, W. and Carde, R.T. (2002), "Filament-based atmospheric dispersion model to achieve short time-scale structure of odor plumes", in Cowen, E. (Ed.), *Environmental Fluid Mechanics*, Vol. 2, Kluwer, Boston, MA, pp. 143-69 (special issue on Chemical Plume Tracing).

Figaro (2008), "TGS 2620 – for the detection of solvent vapors", Technical report, available at: www.figarosensor.com/products/2620pdf.pdf (accessed April 2, 2008).

Grasso, F. (2001), "Invertebrate-inspired sensory-motor systems and autonomous, olfactory-guided exploration", *Biology Bulletin*, Vol. 200, pp. 160-8.

Grasso, F. and Atema, J. (2002), "Integration of flow and chemical sensing for guidance of autonomous marine robots in turbulent flows", in Cowen, E. (Ed.), *Environmental Fluid Mechanics*, Vol. 2, Kluwer, Boston, MA, pp. 95-114 (Special Issue on Chemical Plume Tracing).

Harvey, D., Lu, T.-F. and Keller, M. (2003), "Wind sensor and robotic model wasp development", in Roberts, J. and Wyeth, G. (Eds), *Proceedings of the 2003 Australasian Conference on Robotics and Automation (ACRA'03), Brisbane*.

Hayes, A., Martinoli, A. and Goodman, R. (2001), "Swarm robotic odor localization", *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'01), Wailea-Makena, HI*.

Heil, R. (2004), "A trilaterative localization system for small mobile robots in swarms", Master's thesis, University of Wyoming, Laramie, WY.

Iacono, G. and Reynolds, A. (2008), "Modelling of concentrations along a moving observer in an inhomogeneous plume. Biological application: model of odour-mediated insect flights", *Environmental Fluid Mechanics*, Vol. 8 No. 2, pp. 147-68.

Ishida, H., Tanaka, H., Taniguchi, H. and Moriizumi, T. (2006), "Mobile robot navigation using vision and olfaction to search for a gas/odor source", *Autonomous Robots*, Vol. 20, Springer, Berlin, pp. 231-8 (Special Issue on Mobile Robot Olfaction).

Ishida, H., Nakamoto, T., Moriizumi, T., Kikas, T. and Janata, J. (2001), "Plume-tracking robots: a new application of chemical sensors", *Biology Bulletin*, Vol. 200, pp. 222-6.

Jakuba, M., Yoerger, D., Bradley, A., German, C., Langmuir, C. and Shank, T. (2005), "Multiscale, multimodal AUV surveys for hydrothermal vent localization", *Proceedings of the 14th International Symposium on Unmanned Untethered Submersible Technology, Durham, NH*.

Kazadi, S., Goodman, R., Tskikata, D., Green, D. and Lin, H. (2000), "An autonomous water vapor plume tracking robot using passive resistance polymer sensors", *Autonomous Robots*, Vol. 9 No. 2, pp. 175-88.

Keymeulen, D. and Decuyper, J. (1994a), "The fluid dynamics applied to mobile robot motion: the stream field method", *Proceedings of the 1994 International Conference on Robotics and Automation (ICRA'94), San Diego, CA*, Vol. 4, pp. 378-85.

Keymeulen, D. and Decuyper, J. (1994b), "The stream field method applied to mobile robot navigation: a topological perspective", *Proceedings of the Eleventh European Conference on Artificial Intelligence (ECAI'94), Amsterdam*, pp. 699-703.

Kikas, T., Ishida, H., Webster, D. and Janata, J. (2001), "Chemical plume tracking. 1. Chemical information encoding", *Analytical Chemistry*, Vol. 73, pp. 3662-8.

Kowadlo, G. and Russell, R.A. (2006), "Using naive physics for odor localization in a cluttered indoor environment", *Autonomous Robots*, Vol. 20 No. 3, pp. 215-30.

Kowadlo, G. and Russell, R.A. (2008), "Robot odor localization: a taxonomy and survey", *The International Journal of Robotics Research*, Vol. 27 No. 8, pp. 869-94.

Krishnanand, K. and Ghose, D. (2006), "Glowworm swarm based optimization algorithm for multimodal functions with collective robotics applications", *Multiagent and Grid Systems*, Vol. 2 No. 3, pp. 209-22 (Special Issue on Recent Progress in Distributed Intelligence).

Kunkel, T. (2006), "Hardware architecture of a swarm of robots", Master's thesis, University of Wyoming, Laramie, WY.

Li, W., Farrell, J.A., Pang, S. and Arrieta, R.M. (2006), "Moth-inspired chemical plume tracing on an autonomous underwater vehicle", *IEEE Transactions on Robotics*, Vol. 22 No. 2, pp. 292-307.

Liao, Q. and Cowen, E. (2002), "The information content of a scalar plume – a plume tracing perspective", in Cowen, E. (Ed.), *Environmental Fluid Mechanics*, Vol. 2, Kluwer, Boston, MA, pp. 9-34 (Special Issue on Chemical Plume Tracing).

Lilienthal, A. and Duckett, T. (2003), "Approaches to gas source tracing and declaration by pure chemo-tropotaxis", *Proceedings of the Autonome Mobile Systeme* (*AMS'03*)*, Karlsruhe*, pp. 161-71.

Lilienthal, A., Zell, A., Wandel, M. and Weimar, U. (2001), "Experiences using gas sensors on an autonomous mobile robot", *Proceedings of EUROBOT 2001, 4th European Workshop on Advanced Mobile Robots, Lund*, pp. 1-8.

Lilienthal, A.J. (2005), *Gas Distribution Mapping and Gas Source Localization with a Mobile Robot*, Logos, Berlin.

Lochmatter, T. and Martinoli, A. (2009), "Understanding the potential impact of multiple robots in odor source localization", *Distributed Autonomous Robotic Systems*, Vol. 8.

Marques, L. and de Almeida, A. (2006) in Marques, L. and de Almeida, A. (Eds), *Autonomous Robots*, Vol. 20, Springer, Berlin (special issue on Mobile Robot Olfaction).

Marques, L., Martins, A. and Almeida, A.d. (2005), "Environmental monitoring with mobile robots", *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems* (*IROS'05*)*, Edmonton*, pp. 3624-9.

Martinez, D., Rochel, O. and Hugues, E. (2006), "A biomimetic robot for tracking specific odors in turbulent plumes", *Autonomous Robots*, Vol. 20, Springer, Berlin, pp. 185-95 (Special Issue on Mobile Robot Olfaction).

Russell, R.A., Kleeman, L. and Kennedy, S. (2000), "Using volatile chemicals to help locate targets in complex environments", *Proceedings of the Australian Conference on Robotics and Automation, Melbourne*.

Sandini, G., Lucarini, G. and Varoli, M. (1993), "Gradient driven self-organizing systems", *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems* (*IROS'93*)*, Yokohama*.

Song, Z. and Chen, Y. (2006), "Challenges and some results for the MAS-net project", First Joint Emergency Preparation and Response Meeting, pp. 258-64.

Spears, D., Thayer, D. and Zarzhitsky, D. (2009), "Foundations of swarm robotic chemical plume tracing from a fluid dynamics perspective", *International Journal of Intelligent Computing and Cybernetics*, Vol. 2 No. 4, pp. 745-85 (Special Issue on Swarm Robotics).

Spears, W., Spears, D. and Zarzhitsky, D. (2005b), "Physicomimitics positioning methodology for distributed, autonomous swarms", *Proceedings of the Government Microcircuit Applications and Critical Technology Conference* (*Invited*)*, Las Vegas, NV*.

Spears, W., Spears, D., Hamann, J. and Heil, R. (2004), "Distributed, physics-based control of swarms of vehicles", *Autonomous Robots*, Vol. 17 Nos 2/3.

Spears, W., Zarzhitsky, D., Hettiarachchi, S. and Kerr, W. (2005c), "Strategies for multi-asset surveillance", *Proceedings of the IEEE Networking, Sensing and Control Conference, Tucson, AZ*.

Spears, W., Spears, D., Heil, R., Kerr, W. and Hettiarachichi, S. (2005a), "An overview of physicomimetics", *Lecture Notes in Computer Science – State of the Art Series*, Vol. 3342, Springer, Berlin.

Spears, W., Hamann, J., Maxim, P., Kunkel, T., Zarzhitsky, D., Spears, D. and Karlsson, C. (2006), "Where are you?", *Proceedings of the SAB Swarm Robotics Workshop, Rome*, pp. 129-43.

Tannehill, J.C., Anderson, D.A. and Pletcher, R.H. (1997), *Computational Fluid Mechanics and Heat Transfer*, Taylor & Francis, Washington, DC.

Thayer, D. (2008), "Theoretical foundations of chemical plume tracing with continuous and puff source emitters", University of Wyoming Technical Report, University of Wyoming, Laramie, WY.

Vergassola, M., Villermaux, E. and Shraiman, B.I. (2007), "Infotaxis as a strategy for searching without gradients", *Nature*, Vol. 445 No. 7126, pp. 406-9.

Weissburg, M., Dusenbery, D., Ishida, H., Janata, J., Keller, T., Roberts, P. and Webster, D. (2002), "A multidisciplinary study of spatial and temporal scales containing information in turbulent chemical plume tracking", in Cowen, E. (Ed.), *Environmental Fluid Mechanics*, Vol. 2, Kluwer, Boston, MA, pp. 65-94 (SpeciaI Issue on Chemical Plume Tracing).

Zarzhitsky, D. (2008), "A physics-based approach to chemical source localization using mobile robotic swarms", PhD thesis, University of Wyoming, Laramie, WY.

Zarzhitsky, D. and Spears, D. (2005), "Swarm approach to chemical source localization", *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics (SMC'05), Waikoloa, HI*, pp. 1435-40.

Zarzhitsky, D., Spears, D. and Spears, W. (2005a), "Distributed robots approach to chemical plume tracing", *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems* (*IROS'05*)*, Edmonton*, pp. 4034-9.

Zarzhitsky, D., Spears, D. and Spears, W. (2005b), "Swarms for chemical plume tracing", *Proceedings of the IEEE Swarm Intelligence Symposium* (*SIS'05*)*, Pasadena, CA*, pp. 249-56.

Zarzhitsky, D., Spears, D., Thayer, D. and Spears, W. (2004), "Agent-based chemical plume tracing using fluid dynamics", Lecture Notes in Computer Science, Vol. 3228, Springer, Berlin.

Appendix 1. CPT simulator control loop implementation

Algorithm A1. Top-level control operations of the CPT simulator

```
ALGORITHM: CPT_simulation(lattice)
while (TRUE)
   ap_maintain_formation(lattice)
   for agent in lattice
      agent_do_cpt_strategy(agent)
   end−for
   move_agents_with_constraints(lattice)
end−while
```

Appendix 2. CPT algorithm implementations for the seven-robot study

Algorithm B1. The casting algorithm implemented on a seven-robot lattice

```
ALGORITHM: cast
while (TRUE)
   if (lattice is expanding)
      if (expansion factor is less than maximum)
         increment expansion factor
      else
         change lattice mode to contraction
      end−if
   else
      if (expansion factor is greater than minimum)
         decrement expansion factor
      else
         change lattice mode to expansion
      end−if
   end−if
   radius = expansion_factor*$R_o$
```

```
    if (horizontal advance is blocked)
            reverse horizontal direction
    end–if
    if (vertical advance is blocked)
        reverse vertical direction
    end–if
    waypoint = direction_unit_vector*$R_o$
end–while
```

Algorithm B2. The chemotaxis algorithm implemented on a seven-robot lattice

```
ALGORITHM: chemotaxis
while (TRUE)
        ensure lattice radius and location are within limits
        if (lattice is within plume)
            execute move_to_max_density
        else
            excute cast
        end–if
end–while

STRATEGY: move_to_max_density
    obtain the sensor reading of $\rho$ across the lattice
    move to the location of the maximum $\rho$ reading
```

Algorithm B3. The anemotaxis algorithm implemented on a seven-robot lattice

```
ALGORITHM: anemotaxis
while (TRUE)
    ensure lattice radius and location are within limits
    if (lattice is within plume and wind sensors detect $\vec{V}$)
        execute move_upstream
    else
        execute cast
    end–if
end–while

STRATEGY: move_upstream
    average the direction of $\vec{V}$ across the lattice
    move one time step along the $-\vec{V}$ direction at maximum speed
```

Alogrithm B4. The fluxotaxis algorithm implemented on a seven-robot lattice

```
ALGORITHM: fluxotaxis
while (TRUE)
    ensure lattice radius and location are within limits
    if (lattice is within plume)
        if more than 50% of total $\rho$ is sensed by the center agent
            contract the lattice to minimal radius
        else
            execute chem._region
        end–if
    else
        execute cast
    end–if
end–while
```

```
STRATEGY: chem_region
    sense total lattice $\rho$ over 3 different lattice radii
    compute $\rho$ centroid $C_p$, where $p \in \mathrm{RADIUS}_{\{\mathrm{inner,middle,outer}\}}$
    if ($\rho$ increases with each radial increase)
        move to the centroid of the centroids $C_p$
    else
        if (outermost $\rho$ is greater than innermost)
            move to the location of the $C_{\mathrm{outer}}$ centroid
        else
            if ($\rho$ decreases with each increasing radius)
                execute flux_ring
            else
                execute cast
        end–if
    end–if

STRATEGY: flux_ring
    compute the maximum incoming flux, $\rho\vec{V}$, at 3 different lattice radii
    if (maximum influx exceeds a flux threshold)
        move to the location of the maximum incoming flux, $\rho\vec{V}$
    else
        compute the maximum outgoing flux, $\rho\vec{V}$
        if (maximum outflux exceeds flux threshold)
            move to the location of the maximum outgoing flux
        else
            execute cast
        end–if
    end–if
```

Appendix 3. CPT algorithm implementations for the large swarm study

Algorithm C1. Implementation of the swarm casting procedure

```
ALGORITHM: casting
    if (horizontal advance is blocked)
        reverse horizontal direction
        broadcast new horizontal direction
    end–if
    if (vertical advance is blocked)
        reverse vertical direction
        broadcast new vertical direction
    end–if
    velocity = direction_vector*time_step
```

Algorithm C2. Implementation of the swarm chemotaxis algorithm

```
ALGORITHM: chemotaxis
    if (neighbors are present)
        find the agent with the highest $\rho$ reading
        compute the local gradient $\nabla\rho = \rho_{\max} - \rho_{\mathrm{self}}$
        if ($|\nabla\rho| > 0$)
            velocity = $\nabla\rho$*time_step
        else
            execute casting
    else
          execute casting
    end−if
```

Algorithm C3. Implementation of the swarm anemotaxis algorithm

```
ALGORITHM: anemotaxis
    if ($\rho_{\text{self}} > 0$ and $|\vec{V}_{\text{self}}| > 0$)
       velocity = $-\vec{V}_{\text{self}}$*time_step
    else
       execute casting
    end−if
```

Algorithm C4. Implementation of the swarm fluxotaxis algorithm

```
ALGORITHM: fluxotaxis
    if (more than one neighbor)
       for neighbor in neighbors
          execute neighbor_flux(neighbor)
       end−for
       if (influx detected)
          compute bearing unit vector $F_{\text{influx}}$ toward the neighbor with maximum influx
          velocity = $F_{\text{influx}}$*time_step
       else if (outflux detected)
          compute bearing unit vector $F_{\text{outflux}}$ toward the neighbor with maximum outflux
          velocity = $F_{\text{outflux}}$*time_step
       else
          execute chemotaxis
       end−if
    else
          execute anemotaxis
    end−if

STRATEGY: neighbor_flux
    retrieve $\rho_{\text{neighbor}}$ and $\vec{V}_{\text{neighbor}}$
    let $u$ and $v$ be the $x$ and $y$ components of wind velocity $\vec{V}_{\text{neighbor}}$
    let $dx$ and $dy$ be the $x$ and $y$ components of the neighbor separation vector
    return $\rho_{\text{neighbor}} * |\vec{V}_{\text{neighbor}}| * \cos\left(\arctan\left(\frac{v}{u}\right) - \arctan\left(\frac{dy}{dx}\right)\right)$
```

About the authors

Dimitri V. Zarzhitsky received the PhD degree in Computer Science from the University of Wyoming (UW) in 2008. Presently, he is overseeing design, development, and deployment of unmanned systems for both humanitarian and military applications, and is active in several on-going cross-disciplinary efforts between government, commercial, and academic institutions. His work appeared as invited book chapters in the Springer series on Cooperative Control, as well as in the proceedings of several international conferences and workshops. He is currently an Adjunct Assistant Professor in the Computer Science Department at the UW, working on improving effectiveness of cooperative sensing and control algorithms for distributed autonomous systems. Dimitri V. Zarzhitsky is the corresponding author and can be contacted at: dimzar@uwyo.edu

Diana F. Spears received the PhD degree in Computer Science from the University of Maryland in 1990. She is currently the owner and Senior Research Scientist at Swarmotics, LLC, and also an Adjunct Associate Professor in the Mathematics Department at the UW. Previously, she worked at NASA Goddard Space Flight Center, the National Institute of Standards and Technology, and as an Associate Professor of Computer Science at UW. She is currently serving as an invited member on the AAAI Presidential Panel on the Future of Artificial Intelligence. She has over 80 publications, and is known internationally for her expertise on swarm robotics and multi-robot CPT. Her other research interests include behaviorally assured adaptive and machine learning systems, and mathematical/graphical modeling.

David R. Thayer received the PhD degree in Plasma Physics from MIT in 1983. He has been a Faculty Member of the Department of Physics and Astronomy at the UW since 2000. Previous positions have included: a Senior Scientist position at SAIC working directly under the CEO on projects including biotechnology, global change research, nanotechnology, submarine turbulent flows (DARPA funded), and fusion plasma transport (DOE funded), and a Staff Scientist position at Lawrence Berkeley Laboratory working on ICRF heating through turbulent edge plasma devices. Currently, David R. Thayer is working on CFD research with the Computer Science and Mathematics Departments at UW. In addition, David R. Thayer teaches many of the graduate and upper-division undergraduate physics courses at UW, including classical mechanics, electricity and magnetism, plasma physics, mathematical and computational physics and quantum mechanics. He is a member of the American Physical Society.