

Tracking Based Plume Detection

A Thesis

Submitted to the Faculty

in partial fulfillment of the requirements for the

degree of

Doctor of Philosophy

by

Glenn T. Nofsinger

DARTMOUTH COLLEGE

Hanover, New Hampshire

December 2006

Examining Committee:

George V. Cybenko (chairman)

Eric W. Hansen

Vincent H. Berk

Daniel Rockmore

(Signature of Author)

Charles K. Barlowe
Dean of Graduate Students

©2006 Trustees of Dartmouth College

Thayer School of Engineering
Dartmouth College
“Tracking Based Plume Detection”
Glenn Nofsinger
Doctor of Philosophy

Prof. George V. Cybenko (Chair)
Prof. Eric W. Hansen
Dr. Vincent H. Berk
Prof. Daniel Rockmore

ABSTRACT

The work in this thesis deals with a novel approach for detecting and tracking chemical plumes in distributed sensor networks. The ultimate objective is solving the inverse location problem: given unknown sources released at unknown times, the sensor network must estimate the location and number of sources. Traditional solutions to this problem suffer from the challenge of a large state space, and we therefore seek reduced dimensionality that would allow for real-time solutions. Existing methods are generally computationally intensive, and become intractable when applied to a large scale system that is capable of monitoring an entire city in real-time.

This problem is approached by tracking plumes using methods similar to multiple target tracking (MTT) as opposed to purely physical model inverse solutions. Although this tracking approach requires a higher density of sensor nodes, larger numbers of simple binary low resolution sensors have the advantage of higher spatial resolution. Binary sensor observations are partitioned into tracks based a wind history derived likelihood. With the use of an estimator-based joint probability, groups of sensors form and rank hypotheses that explain the set of observations in the network.

The main contribution of this work is track formation followed by the application of a custom estimator – this process is called the 2-step algorithm. We developed a customized estimator for plumes that allows the MTT-like algorithm to treat the plume tracking problem as the extreme instance of the multi-target tacking (MTT) problem. The central question: how can a MTT-like method be implemented for plumes in a network of simple sensors capable of only binary detection? The simulation experiments demonstrate that the tracking based approach outperforms uniform estimators in conditions of high wind direction variability.

Acknowledgements

I would like to thank the members of my PhD thesis committee for their invaluable input and time commitment: George Cybenko, Eric Hansen, Vince Berk, and Dan Rockmore of Dartmouth College. Without their dedication and selfless mentoring this project would not have progressed. I thank Prof. Cybenko for his patience, generosity, and liberal use of humor in the more challenging moments of this work.

In addition Mark Moran of CRREL/USACE has spent many hours helping with valuable experimental insight and important considerations of this work. My appreciation for his time in evaluating hardware and offering resources for future field testing.

Research described in this paper was partially supported by DHS/ODP Grant 2000-DT-CX-K001. All opinions expressed are solely those of the authors and not the sponsoring organizations.

Table of Contents

1	Introduction	1
1.1	Tracking non-localized distributed processes	1
1.2	Other work	9
1.3	Summary of major results	13
2	Background: plume dynamics and statistical theory	16
2.1	Classical dynamics of aerosol releases	16
2.1.1	Diffusion modeling	17
2.1.2	Wind modeling	23
2.2	Statistical theory	26
2.2.1	Bayesian state estimation	27
2.2.2	Estimators	28
2.2.3	Multi-target tracking theory	35
2.2.4	Process Query Systems (PQS)	37
3	The problem and solution	39
3.1	Problem definition	39
3.1.1	Graphical representation of problem	39
3.1.2	Formal problem statement	40
3.2	Observation correlation in target tracking	47

3.2.1	Role of gating in MTT	48
3.2.2	Source separation	51
3.3	State estimation	51
3.3.1	Plume state space estimation	51
3.3.2	Correlator	55
3.4	Optimizing location with 2-step algorithm	57
4	Plume simulation framework	62
4.1	Introduction	62
4.2	Software implementation	64
4.2.1	Diffusion and wind	64
4.2.2	Modeling of plume observations	65
4.2.3	Plume observation correlator	67
4.2.4	Tracking formation and state estimation	67
4.3	Single release time experiments	69
4.3.1	Studying the impact of wind uniformity	69
4.3.2	Sensor node density study	74
4.4	Continuous release time experiments	74
4.4.1	Releases throughout simulation	76
4.4.2	Continual track initiation	76
4.4.3	Data association - branching	78
4.4.4	Track pruning	80
4.5	Performance metrics	81
4.5.1	Independent variables	81
4.5.2	Error estimation	82
4.5.3	Source counting metric	85

5	Results and analysis of simulation	87
5.1	Single release time results	87
5.1.1	Example release types	87
5.1.2	Likelihood map performance comparison	89
5.1.3	Wind variation results	91
5.1.4	Sensor density results	97
5.1.5	Source number counting results	99
5.2	Continuous release time results	99
5.2.1	Example release types	99
5.2.2	Continuous release tracking	102
5.2.3	Tracking optimization	104
5.3	Scalability	106
6	Conclusion and future work	109
6.1	Needed theoretical work	110
6.2	Field testing	111
6.3	Expansion to other domains	111
6.3.1	Mobile source problem	111
6.3.2	Trusted Corridors	115
6.3.3	Information diffusion	120
A	Code descriptions	121
A.1	Main simulation	121
A.2	Scenario generator	121
A.3	Wind editor	123
B	Variables, constants, and definitions	124

List of Figures

1.1	City example of a non-localized distributed event	2
1.2	Quadrants of existing work	10
2.1	2D Brownian Motion	18
2.2	Random walk distance \sqrt{m} in m steps.	19
2.3	Diffusion equation solutions.	22
2.4	Ideal and noisy sensor response curves	23
2.5	Inverse solution with geometric method.	24
2.6	Peclet number and plume propagation.	24
2.7	Forward diffusion process.	25
2.8	Historical wind data.	26
2.9	Inverse probability for single observations.	31
2.10	Hurricane track showing forward likelihood.	31
2.11	Plume observation correlation functions.	32
2.12	Belief map generated from uniform estimator.	34
2.13	Superimposed correlators, no wind.	34
2.14	Correlation of multiple observations.	34
2.15	Sensor node correlation process.	36
2.16	MTT track formation and hypothesis management.	36
2.17	MHT process steps.	37

3.1	Graphical conventions for plume tracking.	40
3.2	Release matrix R^T	42
3.3	Boundary conditions for numerical solution of forward problem.	43
3.4	Time definitions	45
3.5	Plume problem formulation.	46
3.6	Data association and plume correlator.	48
3.7	MTT observation assignment.	49
3.8	Source separation problem.	50
3.9	Definition of a track.	52
3.10	Data association in 2-step algorithm.	53
3.11	Track evolution over time.	54
3.12	Likelihood map, $M(A)$	55
3.13	Updating the state estimate.	58
3.14	Sequential state estimate.	60
4.1	3D Plots of forward and inverse regions	68
4.2	Tracking results for 50 sensors, 4 sources.	70
4.3	Wind direction data.	73
4.4	Release functions for continuous releases.	77
4.5	Plume release at different times.	77
4.6	Data association for observations.	79
4.7	Belief map comparison with maximum likelihood performance.	84
5.1	Continuous release setup and response	88
5.2	Limited release setup and response	88
5.3	State estimation comparison results	92
5.4	Tracking Vs. No tracking performance.	93
5.5	Raw wind statistics	95

5.6	Wind for sigma=18 and 25 degrees	95
5.7	Wind for sigma=34 and 94 degrees	96
5.8	Wind σ performance summary.	96
5.9	Qualitative summary of wind performance.	97
5.10	Density performance $N = 50, 60, \dots, 300$	100
5.11	Sensor density performance summary.	101
5.12	Source number estimation as wind variation increases	101
5.13	Pulsed release and sensor response	102
5.14	Complex release and sensor response	103
5.15	Multiple track initiation.	104
5.16	Track formation, continuous releases.	105
5.17	Track scoring function.	106
5.18	Data association parameter	107
5.19	Runtime performance scalability	108
6.1	Tunable models.	116
6.2	Mobile target observations.	117
6.3	Mobile source no wind.	118
6.4	Mobile source with wind.	119
6.5	Mobile source problem.	120
A.1	Simulation code hierarchy.	122
A.2	Detailed code hierarchy.	122

List of Tables

1.1	Difficulty of inverse plume problem.	13
4.1	Example release array.	77
5.1	Separation ability table	91

Chapter 1

Introduction

1.1 Tracking non-localized distributed processes

We are all familiar with the problem of incomplete information on a distributed event. During a snowstorm, predicting the duration and magnitude of the storm is very difficult without knowing the current conditions in neighboring areas. Even very high resolution weather data, if only collected from a single location, provides very little information about future weather. Weather is a non-localized process, and observations from a single location are not sufficient to predict future states or reconstruct an understanding of the process. Obviously tracking or predicting future states of such a system requires the collection and merging of observations across time *and* space. The question becomes how to assimilate these spatially distributed observations and gain a better picture of an event happening over a large area.

The same problem of distributed information collection has an even more relevant application than weather. Imagine the following Gedanken-experiment: a sudden news report appears of the suspected detonation of a radioactive bomb near a major city. This city has been outfitted with fixed, binary sensors for radiation detection. Figure 1.1 shows the time sequence of sensor activations that might result. A stream of observations arrive at these large number of sensors but the questions remain: Where was the release? How many release sources? How are observations correlated?

Imagine a few hours later several nearby monitoring stations in the region have verified the radiation, but in the chaos and panic that follows conflicting reports abound in the media. The release site is unknown and the number of sources can not be determined. This situation is

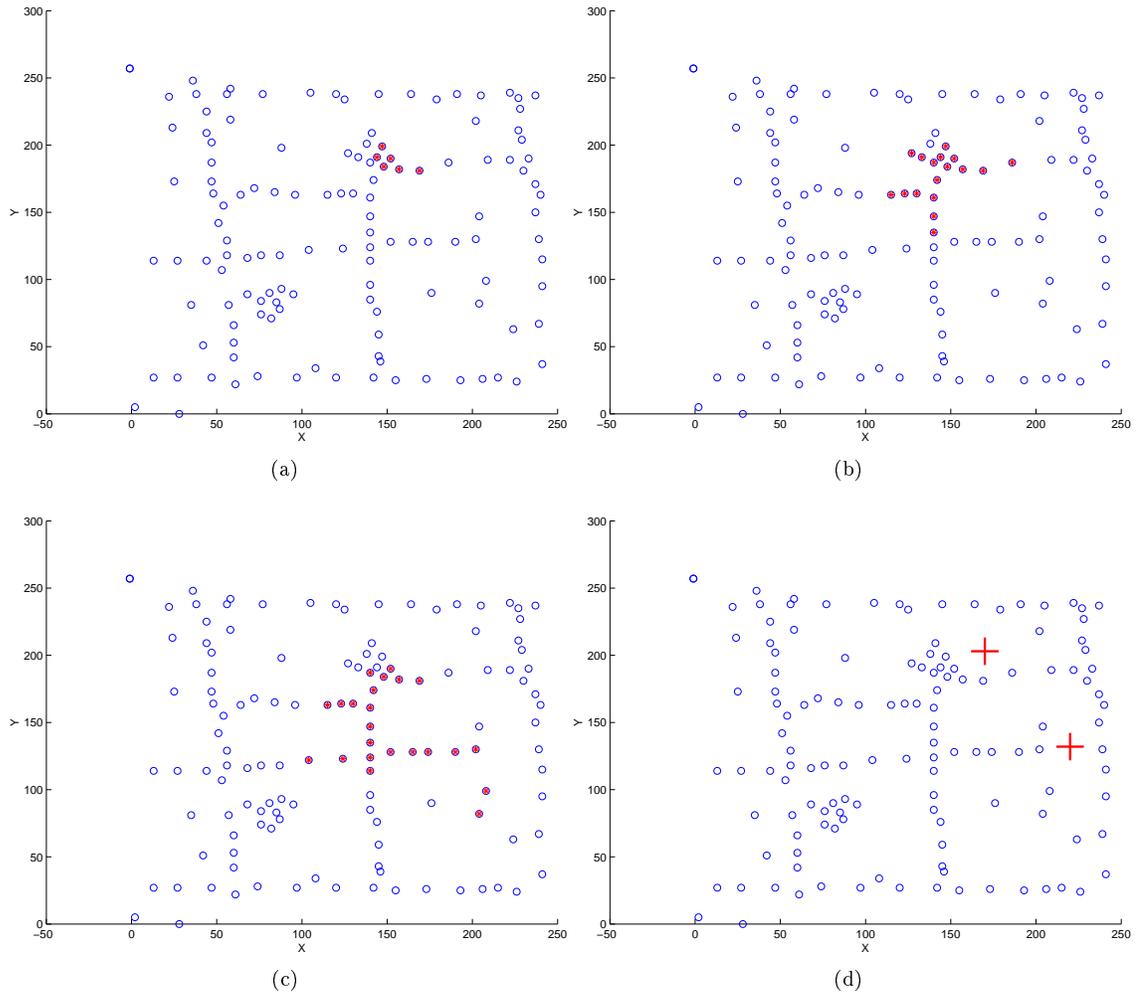


Figure 1.1: Binary harmful agent sensors in a city are activated over time in a non-obvious sequence. Circles indicate sensor position, and filled circles are sensors with a detection. At time $t = 20$ (a) a few localized detections are made of an agent. As time and wind evolve in $t = 100$ (b), $t = 175$ (c), it is not obvious where the source(s) are located. The truth is revealed in (d) as two sources marked with “+.” In this scenario of 250 time-steps the first source was released at time-step 0, and the second at time-step 100.

not due to a lack of information, rather the lack of ability to quickly process or correlate vast numbers of observations. In the United States the collection of air samples, laboratory processing, and dissemination of information is currently highly personnel intensive and involves multiple bureaucratic agencies including the EPA (Environmental Protection Agency), DHS (Department of Homeland Security), LRN (Laboratory Response Network), and the CDC (Center for Disease Control). Due to the major human component in analysis information may be lost, misplaced, or not be retrieved even if relevant. Historical examples of complex harmful agent events include:

- The serin nerve gas attacks in Japan
- The anthrax scare in U.S. Congress
- A false positive detection of tularemia on the Washington DC Mall by Bio Watch sensors

These scenarios required days or weeks to deconstruct and explain what events or sources led to sickness, deaths, or false detections [32]. If an automated network of sensors had been in place, it is possible that the time for information processing and retrieval could have been greatly reduced.

Japan's Subway Terrorism

Several attacks were carried out on judges in Japan by Aum Shinrikyo (a sect), but the judges were not harmed. Instead, unintended victims were harmed when the wind shifted and the airborne deployment of serin nerve gas changed direction. These efforts illustrate the difficulty of predicting the effects of an airborne agent attack, even by the attackers. Aum Shinrikyo improved its tactics for a Tokyo subway attack in 1995. This underground attack used the closed space of the subway as a containment zone, killing over 12 people and making many more ill [32].

BioWatch

Small amounts of *F. tularensis* were detected in the Mall area of Washington, DC the morning after an anti-war demonstration on Sept. 24, 2005. Biohazard sensors were triggered at six locations surrounding the Mall [15, 33]. To this date, no cases of tularemia infection have been reported as a result and this event can be considered a "false positive" threat detection. In the 2005 tularemia incident, an air monitoring program named "BioWatch" had pathogen sensors co-located with EPA stations, but filters from the stations must be collected once every 24 hours and shipped to a laboratory for processing. Equipment for BioWatch is located in select cities reportedly including Philadelphia, New York City, Washington, DC, San Diego, Boston, Chicago,

San Francisco, St. Louis, Houston, and Los Angeles [48]. The Department of Homeland Security does not confirm the exact number of cities engaged in the BioWatch program, nor the number of harmful agents being detected, but it is estimated that up to 120 cities may be included in the United States [6, 5]. While the exact cost of BioWatch is not published, the the yearly budget for one year is estimated at \$1 million per city [48]. President Bush said during his State of the Union address in 2003, immediately before the establishment of the BioWatch program:

Today, the gravest danger in the war on terror, the gravest danger facing America and the world, is outlaw regimes that seek and possess nuclear, chemical, and biological weapons. These regimes could use such weapons for blackmail, terror, and mass murder. They could also give or sell those weapons to terrorist allies, who would use them without the least hesitation. ¹

Unfortunately, the methodology applied by the expensive BioWatch program relies on a relatively low number of monitoring stations, as well as the human transport of air particle samples to laboratory facilities at 24-hour intervals. This type of monitoring worked well for traditional static large events such as factories or nuclear reactors, but does not offer an information sharing component needed to cover a large geographic area.

The main missing element of BioWatch is information sharing and correlation in an automated way. Even with large amounts of data available, identifying the underlying processes producing the data is presently very difficult. Catastrophic events such as dirty bombs, biological agents, hazardous chemicals, or radioactive devices have great uncertainty in time and space, with a highly uniform prior distribution on release locations [13]. We propose that the best method for achieving wide coverage areas of detection are small inexpensive devices that can be networked on a large scale.

Quickly assessing the current state of an affected region demands the automated processing of large numbers of observations yielding potentially competing hypotheses. We define a hypothesis as one particular explanation of the provided observations. This thesis presents a method capable of analyzing large numbers of chemical observations by assigning sensor observations to tracks and hypotheses which are constantly updated, pruned, and ranked on the basis of their likelihood. A set of observations can be explained by multiple hypotheses. As new observations arrive in time, the ranking of hypotheses can change [26]. The inspiration for attempting this approach is based

¹State of the Union Address, January 28, 2003, online at:
<http://www.whitehouse.gov/news/releases/2003/01/20030128-19.html>.

on established work in the field of “Multiple Hypothesis Tracking” or MHT [44].

The field of “sensor networks” emerged with the recent availability of microprocessors, sensors, and wireless communications hardware. Target tracking of a maneuvering target in a field of sensors has become a canonical sensor network problem [36]. To this date, most research has focused on low-level problems such as ad hoc routing of data within these networks, power conservation, geo-location, and communication protocols [22, 40, 51, 50, 17, 27]. We assume other groups will handle these important issues of power and connectivity, and therefore choose to focus on the information processing problem within a sensor network. That is, assuming large amounts of real time data are available to a central location, how can we prevent information overloading and reduce the amount of human interaction required to identify processes of interest in the data?

By leveraging sensor networking technologies, the promise of solving the distributed chemical plume problem with large numbers of simple sensor devices has become possible. Using concepts such as data fusion, information assimilation, and distributed object tracking in ad hoc networks, the potential of agent technologies in future applications has been illustrated. The availability of the hardware and software components for sensor networks enables us to consider solving the plume problem with novel information theoretic approaches. For example, instead of numerically solving complex models based on meteorology, data from a high density sensor network can be treated as other target tracking problems with a simple model. The use of higher sensor density allows the use of less complex models. Due to the fact that these sensor network technologies have matured, we believe that leveraging them for the solution of the plume location problem is an appropriate and timely effort.

Research groups have already successfully demonstrated the tracking of vehicles, humans, and computer attacks with the use of distributed sensor networks [17, 1, 51, 40, 31, 20]. But unlike finite targets, a non-rigid plume spans a region and requires the assimilation of information about a non-localized continuum of targets [51]. Unlike a target that exists in a single confined area at any given time, a plume target is constantly changing shape and size. This aspect of plume tracking (being non-localized) makes the problem unique and more challenging than the traditional target tracking problem. Determination of the entire region of the plume is beyond the scope of this thesis, and we focus on localization of the plume source as the target. In this thesis work the plume source is defined as the “target.” This differs greatly from traditional target tracking problems (i.e. following the path of an airplane) in that the target is stationary, but the medium transporting information about the target (diffusion and wind in the atmosphere) is highly non-stationary.

If a region on land, water, or air is instrumented with a network of physical sensors capable of detecting a specific chemical, it would be useful to infer from collected data as much information about the source(s) as possible. This thesis work aims to answer the following questions about observations in such a sensor network:

1. Were groups of sensor observations produced by the same source?
2. How many release locations were there?
3. What is the best estimate of where and when the releases occurred?

A traditional monitoring system (i.e. Bio Watch) does not have the ability to automatically combine sensor observations across a large geographic area; therefore it has limited capability to provide information about the source location. The system that we have developed takes into account multiple sensor observations in order to pinpoint the plume source(s). This new collaborative approach uses groups of sensor observations as opposed to singular detections.

In hazardous environments, typical measurements and calculations for plumes estimate the state of the plume in the future. Projecting current observations into the future is known as a *forward problem*. In the case of a hazardous chemical attack, it is imperative to know information about plume history. This requires solving the *inverse problem*: taking current observations and making assertions about the source [46, 16]. In this thesis, we run a simple forward model to generate observations and then collect observations in an attempt to solve the inverse problem. This inverse problem is solved by using a novel approach based on the Process Query System (PQS) which is well suited to overcome many of the inherent difficulties of such an inverse problem [12].

The primary difficulty of the inverse plume problem is the very large state space if the problem is to be solved in a purely numerical way. For example if a typical American city the size of Cleveland, Ohio ($\approx 200 \text{ km}^2$) were to be monitored for chemical attacks and the data were collected for centralized computation, the problem would be immense. Common atmospheric inverse models divide regions into 1m grids, resulting in a total of 2×10^8 nodes for Cleveland, as a purely *two-dimensional* problem. These numerical methods are fundamentally sensitive to initial conditions, and suffer from attempts to solve a highly non-linear chaotic system [18]. Because observations of the environment are generally spatially sparse, the models must be correspondingly complex.

More dense sensor arrays however, allow for less complex models. It seems likely that the only way these large scale atmospheric models will be solved at high resolution are via higher sensor

density, not more complex models. Anyone who follows state of the art snow forecasts as compared to actual results more than one day in advance is aware of the limitations of models running on modern day super computers. These models often must run for hours to make predictions for a few hours of data [19]. Given current constraints in computing power, a highly accurate numerical inverse solution to the real-time chemical plume tracking problem is not a viable solution for sensor networks.

Innate properties of chemical plumes make for a very difficult inverse problem. Plume concentrations are discontinuous: only a few meters from the source the gradient is too shallow to detect using time averaging. Turbulence results in filaments of high concentration at significant distances, but also high intermittency. It is common to measure a concentration of zero 80% of the time even in the proximity of the source [30, 31, 45]. Large readings may be present at great distances from the source. Plume propagation is largely determined by wind, which results in chaotic flow regimes and the possibility of extremely non-linear plume expansion.

A naive first order approach to the inverse plume problem might begin by a simple two-dimensional solution to the diffusion equation. However, the well behaved diffusion process is only a minor physical force for transport. Wind is approximately 10 times more relevant than diffusion for transport [42, 11, 10]. Traditional methods for inverse array signal processing based on precise signal intensity fail to transfer into the plume tracking domain due to the highly chaotic nature of plume fluid dynamics.

Another challenge is that chemical observations are often of extremely low resolution, and only indicate a positive or negative result. Common chemical and biological sensors have low false alarm rates, but a potentially high missed detection rate. Accurate chemical or biological sensors frequently are specific to a particular agent (i.e. hydrocarbons or carbon monoxide) which makes the deployment of a multipurpose fleet of sensors expensive. More generalized sensors such as smoke detectors generally have a simple threshold detection mechanism without selectivity to a specific agent. This makes the differentiation and attribution of plume sources difficult. The combination of unpredictable propagation patterns and low resolution sensors makes a solution of the inverse problem with concentration values and triangulation among sensor nodes a very difficult problem.

New hardware networking technologies allow one to consider solving the inverse plume problem with a large number of nodes as opposed to the use of highly complex atmospheric models with a smaller number of nodes. The use of a large number of nodes, however, introduces a new variety of

non-trivial challenges. How should these large numbers of sensor observations be collected, filtered, correlated, and assimilated? Obviously a city containing thousands or millions of embedded sensor nodes could not rely on human data interpretation for event identification.

Our research group at Dartmouth College has spent several years developing a concept called The Process Query System (PQS) which answers many of the needs presented by this large sensor network data correlation problem [12, 1]. PQS is designed for real time observation processing, which leads to a great reduction of large data-sets, allowing data analysts to focus on higher level processes within the data. In the plume application, we wish to select only the most relevant sensor observations, and group them into tracks which greatly speeds up the time required for data analysis.

PQS is based on the familiar signal processing concept of correlation and signal extraction. Given signals (or observations) embedded in an extremely noisy environment, can the presence of the signal of interest be detected? Another central notion in PQS is the concept of a process. All physical or virtual systems may potentially be modeled as finite state systems (even if those states may be hidden or unknown) and these states emit observables indicating the current state [43]. In the case of the inverse plume problem the state of the system might be posed as locations in a two-dimensional area that have or do not have a plume source.

Using the PQS paradigm one develops models matching possible states of the system, and correlates these models against current observations. In this way PQS is able to correlate or filter large numbers of observations and only return events or models of interest - shielding the end user from having to interpret large volumes of low level raw data. PQS has already been used successfully in the area of computer security, and one can think of plume sensor networks as an extension of computer networks into physical world in real time.

In addition, PQS utilizes the concept of hypotheses, developing the concept of multiple hypothesis tracking which is well established in the radar tracking community [44, 3, 2, 1]. A hypothesis is defined as a set of possible “tracks” which are not limited to the traditional notion of physical target tracks. In the case of the plume problem or computer security a track is a collection of observations belonging to a group. Groups of observations are assigned to tracks, and a particular permutation of track assignments is known as a “hypothesis.” Groups of competing hypotheses are maintained and ranked by statistical likelihood.

These concepts fit well with the goals of the large sensor array system seeking to track plumes, where the tracks will be assigned to trajectories or pathways in which the plume travels. Multiple

track permutations could generate the current set of observations, and therefore multiple hypothesis are maintained over time. By maintaining multiple hypotheses, tracks can be initiated, added to, or deleted as new observations arrive. Quickly assessing the current state of an affected region thus demands the automated processing of large numbers of observations yielding potentially competing hypotheses. The PQS approach addresses many of the special needs of the inverse plume challenge such as vast amounts of real time data, and the delayed arrival of observations.

1.2 Other work

This section briefly outlines other work on the problem of plume source detection. The approaches for plume source detection are summarized in Figure 1.2 which divides the existing work into four quadrants. The two axes used to classify approaches are model complexity and sensor mobility. Highly complex model approaches generally attempt numerical solutions to the plume physics. The fluid dynamics of plumes are described by differential equations, and inverse approximations to these equations are solved based on current observations. Simple models are generally rule based, independent of physical models, and give instructions to robots for moving closer to a source. Highly mobile robots typically attempt to receive guidance from these rule based algorithms to locate the plume source. Stationary sensor methods rely on a higher density of simple sensors that continuously monitor their position. The parameters of mobility and model complexity are the two primary features that differentiate the majority of the approaches.

Quadrant I consists of highly mobile agents that utilize complex computational models. An example of such a system is an advanced underwater plume sniffing robot prototype which analyzes high resolution concentration measurements based on Hidden Markov Models (HMMs) [20]. The cost of such robots currently prohibits the deployment of large fleets to cover a large region; however it is a fascinating area of work with future potential.

One of the most popular areas of research has been quadrant II: mobile robots with simple models [34, 29, 27]. These efforts focus on swarm intelligence with large numbers of simple agent robots. The advantages of large numbers of simple robots are the coverage of a large spatial region, simple models, and the ability of robots to track or move towards release sites. The problem of long lasting battery power for mobile robots is presently a limiting factor. Maintaining a fleet of fully charged robots is not presently viable for long term monitoring of a region.

We believe the most promising area to lie in quadrant III: low model complexity and non-

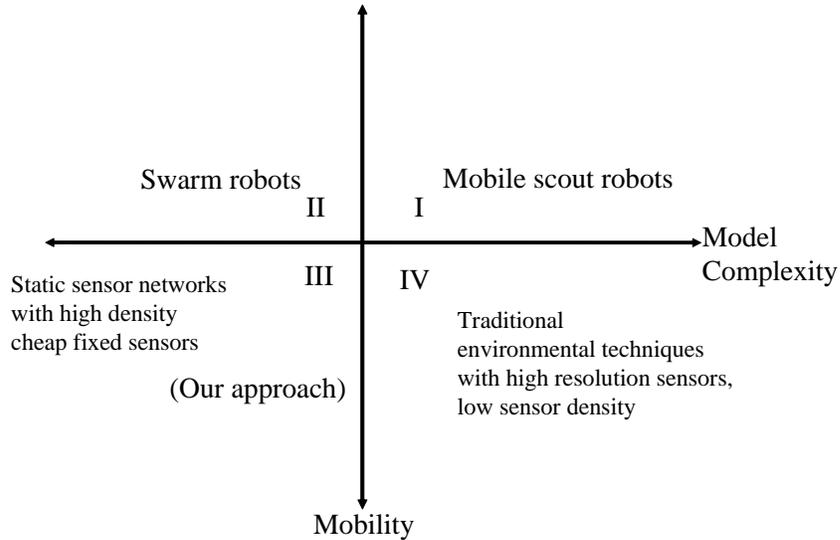


Figure 1.2: Quadrants of existing work

mobile sensors. Inexpensive static nodes allow for the lowest cost and highest sensor density, and much less power consumption. Simple devices which can be deployed with low power consumption could monitor a region for long periods of time. Similar to fire alarms that monitor a building until a incident occurs, there is a great advantage to having a system able to run in standby for many years. As an analogy, most buildings are equipped with stationary fire and smoke detectors – generally the best solution for high spatial coverage and long term monitoring. By choosing a static sensor platform of high density simple devices, it becomes possible to monitor large regions over long time periods. Static systems require much less maintenance, power, and can provide high density observations.

Context of problem to other work

The inverse source location goal of this thesis is addressed in the literature of meteorology, sensor networks, and most commonly robotics. Companies involved in environmental monitoring began as industrial pollutant monitors for industrial compliance, but have now have adapted models from meteorology to homeland security applications. Although similar to industrial pollution monitoring, the inverse plume source problem for terrorism attacks is different in that the number of potential source locations is much greater for an arbitrary attack than for the monitoring of a limited number of static industrial sites.

Most modern meteorology models are categorized as some type of data assimilation method. Current weather observations such as wind, temperature, and pressure are used to predict future

events based on the current state of observations. The Ensemble Kalman Filter Approach (EnKF) is one popular data assimilation method which can be used for the estimation of previous states leading to the current set of observations [18, 19]. The EnKF is a Monte Carlo approach, typically used in weather and climate simulations. It makes random trial guesses about the initial state and runs models forward, calculating the error of simulated observations to actual observations.

This random brute-force approach results in a large state space and is very computationally intensive. Acceptable results typically require the storage of 100 initial model states, and thus the CPU requirements are of the order of 100 model integrations up until the current time for each current observation set. It is well suited for situations in which parallel computers and workstations are available and each processor integrates a portion of the members of the ensemble. It would not be very appropriate for distributed sensor networks of low cost devices.

Particle filters are another type of statistical method, also known as Sequential Monte Carlo Methods (SMC) [9]. Particle filters estimate model parameters based on large numbers of simulations with different initial conditions. They seek to estimate a sequence of hidden parameters, x_k for $k = 0, 1, 2, 3, \dots$, based only on the available data y_k , for $k = 0, 1, 2, 3, \dots$. As with all Bayesian estimates of x_k , they follow the posterior distribution [24]. With particle filters, however, an estimate is obtained for the filtering distribution $p(x_k|y_0, y_1, \dots, y_k)$.

In the usual MCMC (Markov Chain Monte Carlo) approach the full posterior distribution is estimated: $p(x_0, x_1, \dots, x_k|y_0, y_1, \dots, y_k)$. SMC methods run large numbers of simulations and can be much faster than MCMC, and can be used to estimate Bayesian models very accurately [49]. SMC methods approximate a sequence of probability distributions using large sets of random samples, termed “particles.” With a sufficient number of particles (as the number of particles approaches infinity) guesses approach the Bayesian optimal estimate, and an active area of research is the optimal placement of these particles in time and space such that a reasonable number of particles can be used. For the same reasons of computational complexity, SMC is not appropriate for the application of low cost sensor networks at this time.

In mathematical terms, the prior distribution for the plume release points is much more uniformly distributed than for locating a factory or fixed source. Members of the sensor network research field and mobile robot community have adapted target tracking methods to the inverse plume problem [21, 20, 28]. These solutions are generally aimed at mobile robot platforms and emphasize control theory and decentralized robot control. The limitation of mobile robots for long term monitoring is the large amount of power needed for locomotion. These attempts fall into two

broad categories: 1) fleets of simple robot agents and 2) a small number of sophisticated robots. Both of these approaches have essentially posed the challenge as a control theory problem. Different models for robotic movement behavior, decision making, air sampling, information exchange, and collaborative decision making attempt to guide autonomous robot(s) to the plume origin. Many of the robotic behavior algorithms are adapted from animal tracking behavior such as bees, lobsters, or moths. The origin of these approaches is that groups of these animals in nature are exceedingly good at collaboratively locating scents.

Using land-based robots to sniff airborne concentrations, concentration gradients are detected and information is shared between nodes in the effort of collaborative source location [27]. In the venue of underwater robots, testing was performed with one sophisticated unmanned vessel in which concentration as a function of time was analyzed using Hidden Markov Models with the idea that concentration waveforms contain information about source distance [20, 43, 35]. In this experiment, when the robot traverses a plume, different concentration waveforms were observed far away from a source compared to near a source. This location method is termed plume mapping. Mobile robots will not be considered in this thesis work due to the constraints of power consumption and long term deployment. Robot deployment may be suitable for covering a large area with sensing assets after a release has occurred in a known region.

If a matrix is constructed summarizing the level of difficulty of inverse plume source problems, problems can be categorized into classes with increasing degrees of difficulty (A - D, with D being the most difficult). In general the solutions to class A, B, and C problems are well understood and very commonly implemented. The primary mode of operation for this work is indicated in class D where sensor density is high, however sensors will contain very quantized observations and intermittent reporting. The time of the release(s) is not known. These are conditions typical of inexpensive but highly distributed sensor nodes operating in ad hoc networks. The techniques mentioned at the beginning of this section are commonly used to solve the computationally intensive problem of accurate inverse advection-diffusion, but require the presence of high resolution observations. For this reason the observations are typically spatially sparse, with distances of miles or tens of miles between weather stations. These traditional methods are categorized as “easy” since they are well documented and effective.

If, however, we wish to gain higher monitoring resolution in chaotic flow regions such as cities, a higher sampling density is required. In this case large numbers of observations require the sacrifice of high concentration precision. Therefore, the inverse problem is now termed as “hard.”

PARAMETER (+/- factors)	A EASY	B	C	D HARD
Time of source release (+)	X	X	X	O
Prior Distribution (+)	X	X	O	O
Wind (-)	O	X	X	X
Unknown number of sources(-)	O	O	X	X
Intermittent Sensor Reports (-)	O	O	O	X
Quantized Sensor Resolution (-)	O	O	O	X

Table 1.1: The inverse plume source problem divided into four classes of difficulty. This thesis work focuses on class D with unknown release time(s), intermittent sensor reports, and variable or low resolution sensor observations. X=element is present, 0=element not present. (+) parameter assists solution. (-) parameter makes solution more difficult.

Working with highly noisy, highly quantized observations is the focus of this thesis, and falls in a class of work not yet established. It is the purpose of this research to solve the source location problem in scenarios that have limited information about potential source locations in the presence of wind, highly intermittent sensor reports, and low resolution observations. By taking advantage of models which are not purely physical in nature it is possible to make statistical correlations between observations without fully understanding the complex analytical solution to the problem.

1.3 Summary of major results

The main contributions of this work are (1) a real-time simulation system for testing the response of sensor networks to the release of a chemical plume; (2) a demonstration of the system tracking and locating the sources of plumes within a two-dimensional area; and (3) three new algorithms developed for the system:

- A plume sensor data association algorithm that determines the likelihood of association between observations made at neighboring binary sensors within a region. This data association value is based on the wind history, diffusion constant of the chemical substance, and the relative location of two sensors.
- A track building algorithm which creates *tracks* of sensor observations. Tracks are defined as collections of observations believed to originate from the same plume release source. Observations are added to tracks based on data association values with other observations already part of a track. Track building performance can be optimized by adjusting parameters of data association thresholds, as well rules for inclusion to a track. This is the first step of the *2-step algorithm*.

- A custom state estimator algorithm which operates on the observations selected to be included in a track. This algorithm performs an inverse source location estimation based on a set of observations. This is the second step of the *2-step algorithm*.

Initial efforts on this research were aimed at creating a hardware system capable of field testing the inverse plume source location tracking concepts. Field experiments require a focus on sensor networking, ad hoc networks, and data collection within the network. It was decided to focus on the data processing aspect of the tracking theory, and hardware efforts were abandoned in favor of simulation. Working in simulation allowed concentration on high level information extraction, and theoretical limits of the tracking approach to plume source detection.

We will show that adapting concepts from MTT to the plume tracking problem in sensor networks leads to a new method with several benefits. First, the method is effective in large sensor networks where nodes provide only intermittent observations. Chapter 2 will present plume dynamics theory and the standard approaches used in sensor networks for target tracking that are relevant to the plume tracking problem. Chapter 3 poses the exact mathematical problem and develops the theory and the specific solution developed in this thesis. Next, Chapter 4 presents the simulation system and describes the implementation of the concepts in software. Chapter 5 reviews the performance of the tracking system simulation and examines its advantages compared to non-tracking based systems.

For an unknown number of plume releases that diffuse as a Gaussian, we will estimate the number of releases, and a likelihood map for each release showing the estimated location in two dimensions. Under many conditions, the MTT-like method will be more accurate than analytical methods. To analyze the performance of the tracking approach two experimental studies will demonstrate the advantages of the MTT method.

1. Source estimation as a function of sensor density
2. Source estimation as a function of wind direction variability in the sensor field

The experimental results from these two studies presented in Chapter 5 reveals that the tracking based approach has a much greater ability to identify the number of plume sources, can operate in lower sensor density areas, and performs well in areas of high wind variation.

This thesis will show that it is possible to solve the inverse plume source problem with low resolution observations from simple sensors. Binary detection is a realistic approximation to many

current detectors for chemical, biological, or radiation threats. By using MTT-like data association methods groups of sensors will report a likelihood map indicating the most likely release point that generated the observations. The complete inverse problem of the parabolic diffusion equation is ill-posed and nonlinear, so a numerical solution is quite difficult. We solve the ill-posed problem by using probability, data association, and tracking methods.

Unlike traditional analytical “solutions” to inverse problems we will not produce a unique solution to each set of observations, but instead generate families of hypotheses which explain the observations. We have presented previous work which shows the viability of the tracking approach, but the results in the thesis will focus on the performance of hypotheses via likelihood maps [37, 38, 39]. These hypotheses will be in the form of likelihood maps which indicate the areas within the region which most likely produced the observations.

We will see that belief maps of simple sensor networks not using tracking will require a higher density of nodes for similar performance, and that sensor networks applying MTT before the generation of belief maps will predict the source location more accurately in conditions of highly variable wind. This advantage is especially apparent in situations with highly variable wind direction *and* low sensor density. In addition, the determination of the number of sources will have dramatically better results with MTT methods.

Chapter 2

Background: plume dynamics and statistical theory

This chapter introduces the forces acting on a plume from first principles. The two forces considered in the equation describing plume concentration are diffusion and advection (wind). The analytical solution of these equations leads to a method for inverting sensor observations to find the source location. This purely analytical approach to solving for source release locations breaks down once highly noisy or binary sensor observations are introduced. The binary sensor problem means that statistical approaches are required. The second half of the chapter introduces the relevant Bayesian statistics and tracking theory used in the tracking based solution to the plume problem. Finally the concept of PQS (Process Query Systems) are introduced; the tracking based approach to plume detection originates from PQS concepts.

2.1 Classical dynamics of aerosol releases

This section explains the classical models used to describe plume dynamics. Although the forward evolution of a plume is completely described by classical physics, performing the inverse problem based solely on these methods quickly becomes intractable [49]. Once binary sensors are introduced as the observation mechanism, the traditional analytical methods fail to extend. For this reason, a statistical Bayesian formulation to the problem is an obvious approach to consider. The next section will handle the statistical approach of using state estimation and target tracking techniques.

2.1.1 Diffusion modeling

Diffusion can be viewed as the random wandering of an ensemble of particles from regions of higher concentration to areas of lower concentration. This is the process by which matter is transported from one place to another as a result of random molecular motion. If it was possible to watch individual molecules, it would be observed that a single particle moves randomly, having no preferred direction of movement [10]. The behavior of clouds of particles can be derived from considering large numbers of these single particles. The mathematical model used to describe such random motion is called Brownian motion. Figure 2.1 illustrates the path of a single particle under such Brownian motion for two different numbers of steps ¹.

The most basic way to model diffusion is with the “random walk” of a single particle. In a $1D$ random walk each particle moves independently and moves during each time interval, Δt , according to the following rules:

- there is a 25% chance of the particle moving to the right
- there is a 25% chance of the particle moving to the left
- there is a 50% chance of the particle staying in the same position [11].

Due to the random and non-deterministic description of a random walk, we can only describe the future position of a particle in terms of probabilities. In the case of a $1D$ random walk, what is the expected likelihood that a particle will wander a distance $n\Delta x$ after m steps of time, Δt ? In terms of probabilities this can be defined:

$$p(n\Delta x, m\Delta t) \tag{2.1}$$

where the distance at each successive time $(m + 1)\Delta t$ is based on the distance at time, $m\Delta t$. It is well known that the expected distance from the starting position at time 0 for a random walk after m time-steps is proportional to \sqrt{m} . Figure 2.2 illustrates a random walk experiment performed in $1D$ for $M = 30$, the total number of time-steps. The actual distance of 5 is close to the expected value of $\sqrt{M} = \sqrt{30} = 5.48\dots$

The connection between random walks and analytical descriptions of diffusion occurs once we take the limit of the random-walk process for very small time-steps Δt and very small distances

¹http://en.wikipedia.org/wiki/Image:Brownian_hierarchical.png

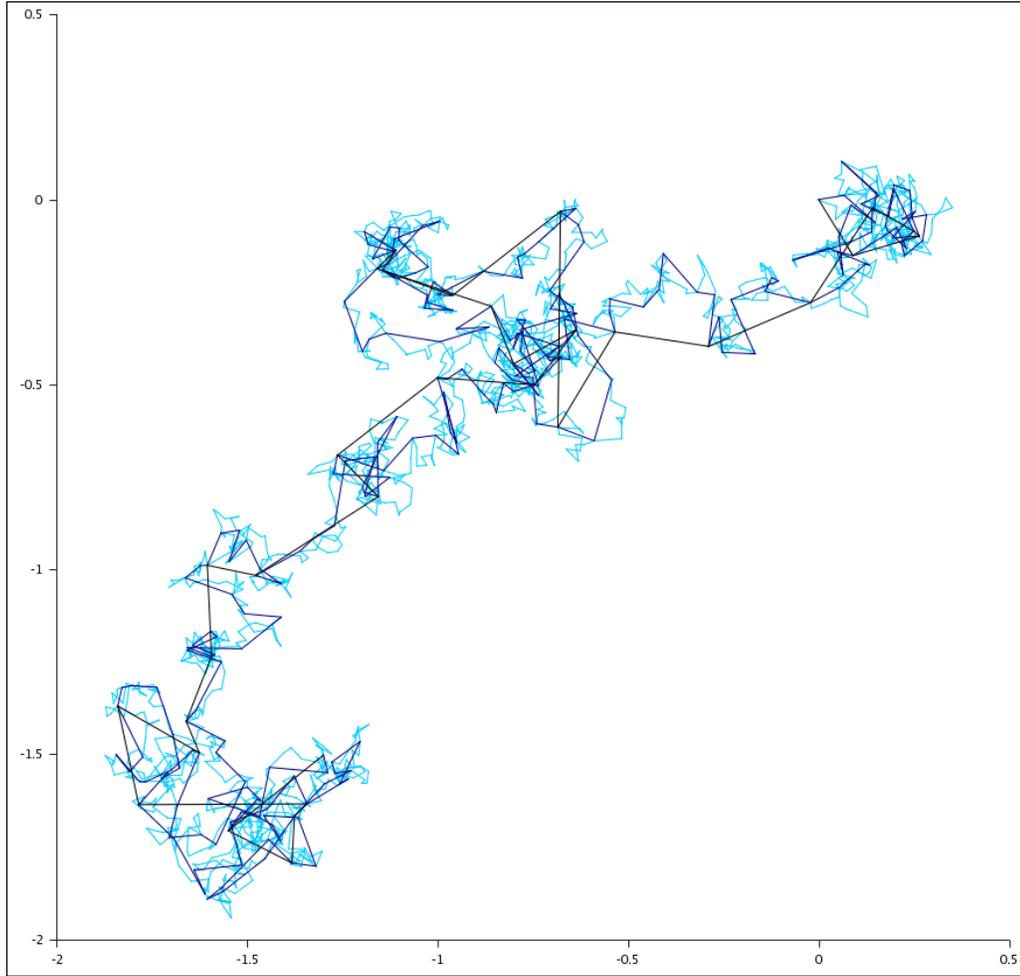


Figure 2.1: Position as a function of time for a particle undergoing the stochastic process known as Brownian motion, with 256 (dark blue) , and 2048 steps (light blue). The particle path begins in the upper right and ends in the lower right section of this plot. The diffusion equation produces an approximation of the time evolution of the probability density function associated with the position of the particle .

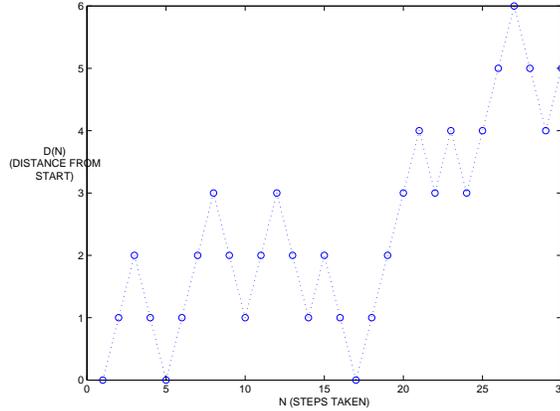


Figure 2.2: Random walk in 1D where \sqrt{m} is the expected distance after m steps.

Δx . Using a Taylor expansion of 2.1 for small incremental values of its arguments and making some mathematical simplifications produces an equation similar to the well known diffusion equation:

$$\frac{\partial p}{\partial t} = D \frac{\partial^2 p}{\partial x^2} + O(\Delta t, \frac{\Delta x^4}{\Delta t}), \quad (2.2)$$

where the diffusion constant D is defined:

$$D = \frac{\Delta x^2}{4\Delta t},$$

and the error term on the order $\frac{\Delta x^4}{\Delta t}$ goes to zero in the limit. When this occurs, the random walk equation produces:

$$\frac{\partial c}{\partial t} = D \frac{\partial^2 c}{\partial x^2}. \quad (2.3)$$

which is also known as the diffusion equation.

By developing this random walk model for particle diffusion, we can build a simulation to study plume tracking in sensor networks. A plume experiment consists of setting up the parameters for the plume (wind speed, total particle count, number of iterations to run, diffusion constants) and then placing sensors within the plume field. Wind is introduced by skewing the transition probabilities in a direction. Once the experiment concludes, the user has access to the concentration as a function of time at each of the sensor locations. Using this collected data, the plume sensor network may then implement a number of coordinated algorithms to estimate the plume source as well as plume boundaries. This method was attempted, but it soon became obvious that a random walk approach was too computationally intensive for desktop computer based experiments. Correctly

simulating a plume requires modeling thousands or millions of particles, and their independent locations.

Another approach to generating a plume diffusion model is developing an analytical expression for concentration from first principles. Diffusion as well as heat transfer are due to the random motion of molecules and obey the same mathematical theory. This was first recognized by Fick (1855) who adopted the heat equation developed by Fourier (1822). This mathematical theory states the rate of transfer of diffusing substance through unit area is proportional to the concentration gradient normal to that section,

$$F = -D \frac{\partial C}{\partial x}, \quad (2.4)$$

where F is the rate of transfer. This is also known as the First Law of Diffusion. The transfer of particles passing a single region (rate F) is determined by the concentration of particles C , diffusion constant D , time t , over some distance x [10]. The negative sign indicates that particles flow away from areas of higher concentration.

If we assume that mass is conserved, this leads to the conservation equation [41],

$$\frac{\partial C}{\partial t} = -\frac{\partial F}{\partial x}. \quad (2.5)$$

Combining this conservation of mass assumption with the First Law of Diffusion results in the Second Law of Diffusion in one dimension, which is also known as the “diffusion equation” as described above:

$$\frac{\partial C}{\partial t} = D \frac{\partial^2 C}{\partial x^2}. \quad (2.6)$$

Extending Equation 2.6 to two dimensions, assuming anisotropic diffusion (not uniform in both directions) along with linear terms for wind, we get an equation describing the rate of change of concentration of particles under the influence of both diffusion and advection (wind),

$$\frac{\partial C}{\partial t} = D_x \frac{\partial^2 C}{\partial x^2} + D_y \frac{\partial^2 C}{\partial y^2} + \alpha \frac{\partial C}{\partial x} + \beta \frac{\partial C}{\partial y}, \quad (2.7)$$

which leads to a standard analytical solution consisting of Gaussian distributions,

$$C(x, y, t) = \frac{A}{4\pi t\sqrt{D}} e^{(-\frac{(x-\alpha t)^2}{4D_x t} - \frac{(y-\beta t)^2}{4D_y t})}. \quad (2.8)$$

If we assume diffusion is uniform (isotropic) in the x and y directions, $D_x = D_y = D.$, meaning particles have a tendency to drift in all directions equally then:

$$\frac{\partial C}{\partial t} = D\left(\frac{\partial^2 C}{\partial x^2} + \frac{\partial^2 C}{\partial y^2}\right) + \alpha\frac{\partial C}{\partial x} + \beta\frac{\partial C}{\partial y}. \quad (2.9)$$

The wind constants α and β represent the corresponding wind speeds in the x and y directions. The α and β terms in a more general solution with time-varying wind would be replaced by $\alpha(t)$ and $\beta(t)$. The solution of interest for this differential equation for a point release at $x = y = 0$, at $t = 0$ is [11]:

$$C(x, y, t) = \frac{A}{4\pi t D} \exp\left(-\frac{(x - \alpha t)^2 + (y - \beta t)^2}{4Dt}\right), \quad (2.10)$$

with a single constant A due to the assumption of isotropic diffusion. In the case of anisotropic diffusion $D_x \neq D_y$, and A would be replaced by two constants A_1 and A_2 for the x and y dimensions. For fixed $t = t_0$, the profile of concentration $C(x, y, t_0)$ along any line $y = mx + b$ will have a Gaussian distribution. If the concentration profile begins as a point we will see spreading Gaussian distributions as time progresses. As a direct result of Fick's Law, the flux in any direction is proportional only to the gradient in that direction. If the diffusion constants are unequal, the cloud will disperse an-isotropically, growing more quickly along the axis with the greater diffusion constant. The characteristic propagation length along any axis will be proportional to the diffusion coefficient along that axis.

A common measure in environmental engineering for the boundaries of a plume is 4σ , that is the boundary that includes 4 standard deviations of the plume material or 95% of the material [11]. The rate of spreading is of interest later when we attempt to predict the location of a plume at a future time, $t \geq t_0$. The width traversed in the x and y directions for a two dimensional plume defined by the width 4σ undergoing pure isotropic diffusion (no wind) is given by:

$$L = 4\sigma = 4\sqrt{2Dt}, \quad (2.11)$$

meaning the width of a plume increases as the square root of time t .

Using the above equations for concentration as a function of time at a given position, Equation

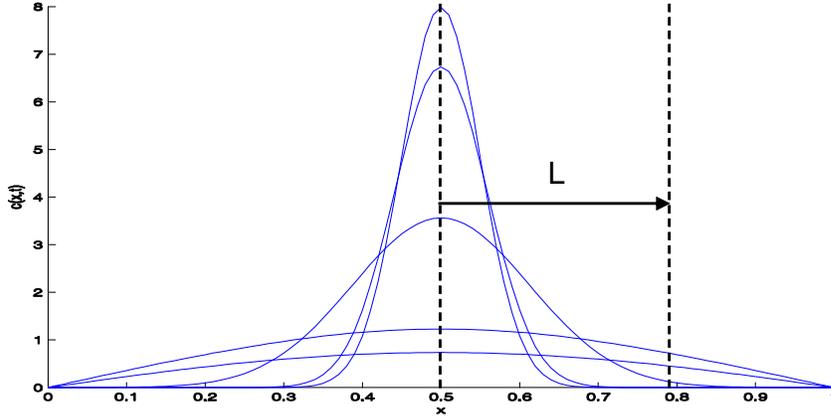


Figure 2.3: Diffusion curves as time increases showing widening Gaussian distributions. A patch of aerosol released acting under pure diffusion has these statistical distributions as time increases. A width of L or 4σ contains 95 % of plume material and is considered boundary by our definition.

2.10, and the distance of spreading for a plume boundary, Equation 2.11, it is possible to construct a series of equations that allow a group of sensors to invert sensor observations to find the source. Assuming a single source located at position (x_0, y_0) which releases at time $t = t_0$, the concentration response curve seen at any sensor S_n in this region would be described by Equation 2.10 as $c(x, y, t)$. One simple way to invert for (x_0, y_0) from a group of N sensors S_1, S_2, \dots, S_N is taking:

$$\frac{\partial c}{\partial t} = 0$$

for equation 2.10 and solving this equation for t . Then plug this t_{max} back into 2.10. It can be shown it is possible solve analytically for (x_0, y_0) .

One interesting property of these t_{max} times for a network of sensors is the unique set of values produced for a specific release point (assuming a single release). The differences between these t_{max} values are unique to a particular source position, and it can be show through solutions to the classic diffusion equation that two sensors are required to invert for a source location in 1D and 3 sensors are required in 2D. The problem with this approach is not knowing t_0 for the release. In addition this ideal analytical solution does not apply in most real world scenarios with highly noisy or binary sensors.

The introduction of binary sensors creates the problem that precise concentration measurements are not available, and inverting for source location based on algebraic systems of equations is not possible. A binary sensor begins sending a “hit” observation on the rising edge of a signal, and continues reporting a positive observation until the level drops below threshold. This ambiguity

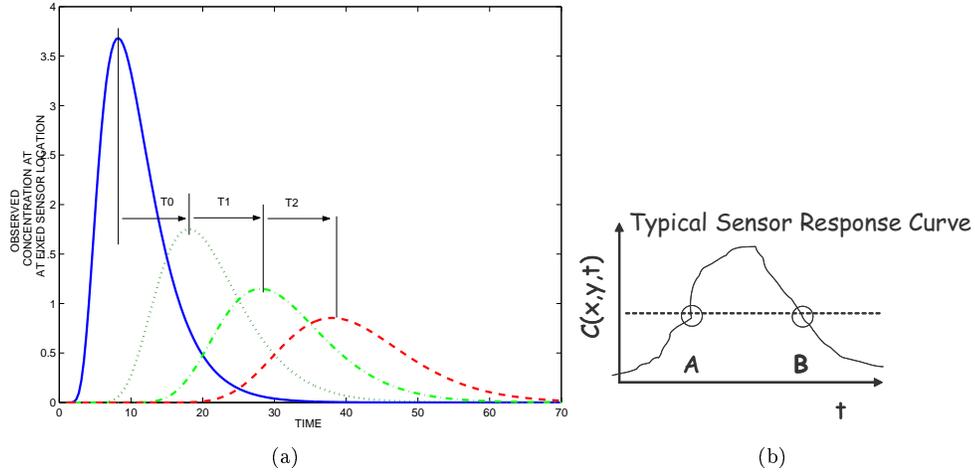


Figure 2.4: Based on the 2D analytical solution to the diffusion problem a fixed sensor has a characteristic response to a pure diffusion event, as a function of distance from the source. The curves in (a) show 3 sensors located at $d=10,20$, and 30 units from the source. For a fixed network of sensors, the relative times between observed maxima are a function of the source location. The response curve in (b) shows a typical noisy concentration response $C(x,y,t)$ of a chemical sensor. The level for binary threshold is set by the dotted line. Trigger-points A and B show boundaries for the region of potential sensor activation.

produces the situation of not knowing when the concentration reaches a maximum, or if the signal is rising or falling. This range of time that the sensor is active creates uncertainty in the release point location, as portrayed in Figure 2.5. In an ideal solution groups of sensors could invert for source location based on analytical geometry, but binary sensors produce a region of large error and uncertainty about the potential point of release. This limitation leads towards a statistical approach to the problem in the next chapter. Based on the binary nature of sensors, and the inability to form an analytical solution to the inverse source problem the next section introduces the notion of Bayesian statistics, and the tracking approaches used to solve the problem.

2.1.2 Wind modeling

With the introduction of wind the plume process becomes a combination of advection and diffusion. Wind is the dominant force effecting plume motion, and many degrees of complexity can be considered. Non-linear effects, turbulence, and barrier interaction would be included in a more general model, but we will limit this model the first order effects of a linear wind and simple advection. Under the assumptions of proper decorrelation lengths, wind can be assumed to be uniform. Under its classical definition, the decorrelation length of a random process is defined as the lag after which the correlation between two time instants is approximately zero [23]. For

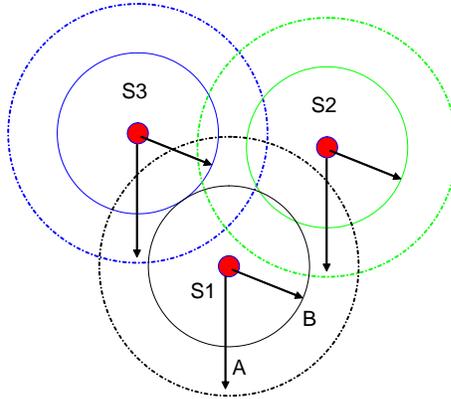


Figure 2.5: The introduction of binary sensors and noisy detection. in a network of three sensors S_1 , S_2 , and S_3 . Binary detections define two circles based on rising and falling edges of the detection.

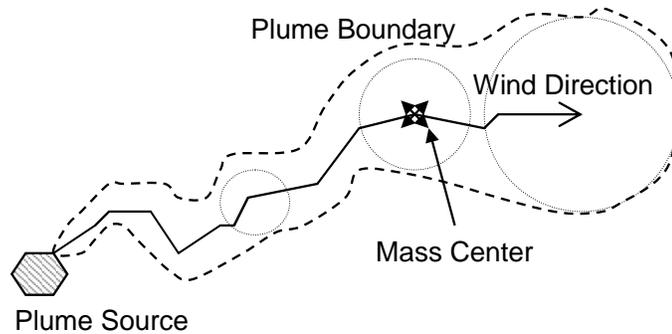


Figure 2.6: Forward plume model propagation is a function of diffusion constant D and wind W , with wind determining the dominant direction. The ratio $P_e = \frac{D}{W}$, known as the Peclet number, determines the width of the plume region. The Peclet number is a measure of the relative importance of diffusion to advection. We wish to invert observations of the plume to find its source.

a wind process the value of decorrelation length determines the range over which wind can be assumed uniform, and in our model this value is larger than the dimension of the simulation grid. This means wind in our model we can assume uniform wind effects on all cells in the $2D$ region A of the simulation.

The Peclet number measures relative strength of diffusion to wind. This ratio determines plume width in our model. For materials with a higher Peclet number, diffusion is a more dominant factor. A typical Peclet number is $\frac{1}{10}$, which is the value used for the simulations in this thesis. In this case the force of wind is one order of magnitude more important than diffusion. For that reason, the primary focus of our plume tracking algorithms will focus on wind dynamics and wind history.

The generation of wind data can be accomplished by making a Markov model where each wind direction is represented by one state. The change of wind direction is equivalent to a transition between states. This was the original model used in the thesis work. The problem with generating

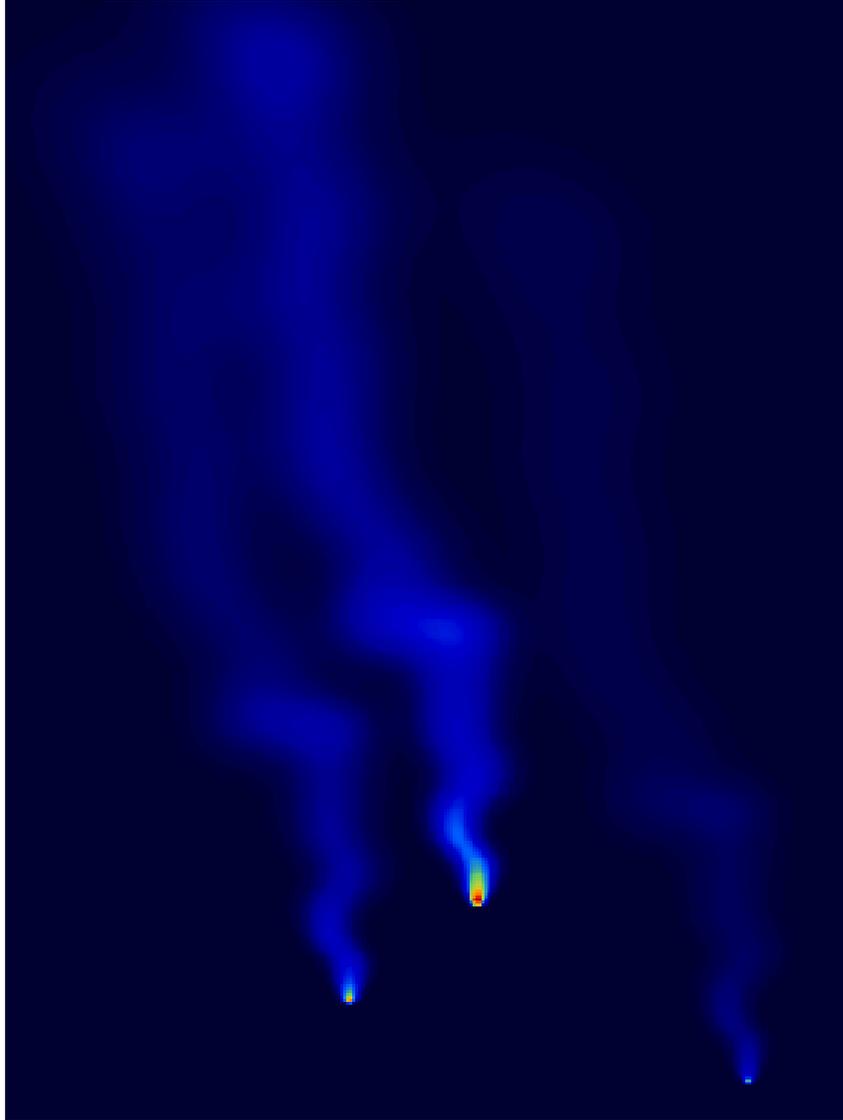


Figure 2.7: Forward diffusion process developed for this thesis based on diffusion and advection. Three plume sources of different initial concentrations started this simulation. Wind fields for the simulations are obtained from online database records obtained from NOAA, allowing the selection of hundreds of weather station locations around the United States.

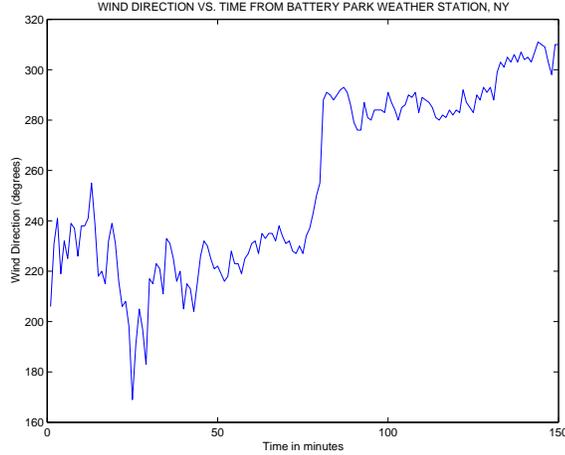


Figure 2.8: Historical wind data collected from battery Park, NY.

such data is the difficulty in proving its statistical properties resemble real wind. In order to create a more realistic model for plume behavior, real wind data can be integrated into a simulation if the length scales and units are scaled to appropriate values.

Methods such as bootstrapping can be used to generate test wind data from small samples of real wind data, however such large amounts of real time data are available, this method was not required. Real wind data from thousands of weather stations are available from the NOAA website. This data includes wind direction, speed, as well as temperature and other weather measurements such as humidity. By downloading this data and selecting blocks of time, the real wind data can be quantized from $(0 - 360^\circ)$ into 8 directions in the simulation. The wind history vector Θ is defined as the time series of wind directions observed at a sensor derived from the following data sets. Spatial uniformity and decorrelation length scales for wind are typically 50 – 200 miles, and the scale of the simulation work is set to this distance for the great simplification of assuming uniform wind. Decorrelation length scale of wind indicates the distances over which the spatial uniformity assumption is valid.

2.2 Statistical theory

As the previous section illustrated, ideal analytical expressions for plume concentration based on diffusion and wind quickly lose applicability when binary sensors replace ideal concentration sensors. Statistical modeling can estimate parameters that describe the phenomenon, but are not based on physical models. In this section Bayesian estimation, tools from target tracking, and

finally process query systems (PQS) will be introduced as alternative statistical approaches to solving the inverse location problem.

2.2.1 Bayesian state estimation

The goal of tracking problems is target localization. The unknown true state of the target at time t , x_t , may include position, velocity, and other parameters of the physical target. This target can be measured by some sensor, and the observations at time t produce observations, z_t . In the case of a plume “target” the target state x_t includes concentration for all the cells within the matrix A for a release event defined by the plume boundary of 4σ . The set of all target states is defined as a set X^T , where:

$$X^T = \{x_1, x_2, \dots, x_T\}$$

is the set of all states, and based on a set of sensor observations, Z^T , where:

$$Z^T = \{z_1, z_2, \dots, z_T\}.$$

The best possible estimate of X^T is desired, where an estimate of X^T is defined as \hat{X}^T . Each z_t contains position and whether or not the concentration at t is above threshold for sensors (in the case of binary sensors). There is a classic Bayesian formulation to this localization problem.

Bayes theorem relates the conditional and marginal probability distributions of random variables, and is a widely used result in probability. The probability for an event A conditioned on a second event B is generally different than the probability of event B conditioned on event A , although there may be a relationship. Bayes theorem relates these two conditional probabilities:

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}, \tag{2.12}$$

where each term is typically known by the following terms.

- $P(A)$ is the *prior probability* or *marginal* probability of A . It does not consider any knowledge about B and is therefore termed prior.
- $P(A|B)$ is the *conditional probability*, also known as the *posterior probability*, because it is conditioned on B
- $P(B|A)$ is the conditional probability of B given A

- $P(B)$ is the prior or marginal probability of B

In the application of interest we can adapt the Bayesian formulation by considering the observations at the sensors over a time interval T as the random variable Z^T , where the observation set $Z^T = \{z_1, z_2, \dots, z_T\}$. This observation set Z^T can be considered the observable on the true plume state space, called X^T . The sequence of states X^T evolves over time, where $X^T = \{x_1, x_2, \dots, x_T\}$. Based on Z^T estimates to the true state can be made, known as \hat{X}^T . Given an observation sequence Z^T what is the conditional probability with a certain X^T ? Using the standard Bayesian definitions, the problem components are:

- $P(x)$ - the prior probability of a specific plume state, independent of z
- $P(x|z)$ - the conditional probability or posterior probability of a plume state given a certain measurement
- $P(z|X)$ - the conditional probability of a plume observation given a certain plume state
- $P(z)$ - is the prior or marginal probability of an observation
- \hat{x}_t - our state estimate at time t
- x_t - the true state at time t

The goal of this approach is to obtain a good estimate of the plume state \hat{X}^T based on a measurement history Z^T which is as close as possible to the true state X^T . The tracking form of Bayes rule states the probability of a given state x given the observation z [14]. The relationship between a posteriori distribution, the a priori distribution, and the likelihood function is:

$$p(x|z) = \frac{p(z|x)p(x)}{\int p(z|x)p(x)dx} = \frac{p(z|x)p(x)}{p(z)}. \quad (2.13)$$

2.2.2 Estimators

Given the observation set z_1, z_2, \dots, z_T , what is:

$$p(x|z_1, z_2, \dots, z_T)?$$

The purpose of an estimator is making this calculation, and many different standard estimators are used. A few of the most common estimators are described here.

- *MMSE*: minimum-mean-squared error. This is a commonly used estimator in the standard estimation theory. Given a set of observations z_1, z_2, \dots, z_T the MMSE estimate produces the expected value or mean of the distribution $p(x|z_1, \dots, z_T)$, which can be written as,

$$\bar{x} = \int xp(x|z_1, \dots, z_T)dx, \quad (2.14)$$

where the MMSE estimator is the mean of the the so called posterior density.

- *MAP* : maximum a posteriori, maximizes the posterior distribution. It is derived the uniform cost function , and considers both information from the measurement and the prior information about the state:

$$\hat{X}_{MAP}^T = \operatorname{argmax}p(X^T|Z^T). \quad (2.15)$$

- *ML*: maximum likelihood, considers information in measurements only:

$$\hat{X}_{ML}^T = \operatorname{argmax}p(Z^T|X^T). \quad (2.16)$$

A Custom Estimator

We desire a custom estimator for the plume state estimate that minimizes the probability of error. Instead of including all observations at t , another alternative is selecting only the observations believed to originate from a specific plume source. As opposed to an estimator that considers all observations equally (such as the “uniform” or MMSE), this estimator first segments observations into tracks to determine which observations should be included in the estimate. Once this step is performed the assumption can be made that all estimators for nodes within the track correspond to the same source. An estimate will be performed separately for each group (partition) of sensors believed to belong to the same source. This estimate will be the second part of the *2-Step algorithm*, presented in Chapter 3.

Total Area of Detectability (TAD)

Figure 2.9 shows the distribution shapes for three cases of advection diffusion. In the first case with no wind, the agent disperses in a uniform circular pattern defined by pure diffusion. Under such conditions if a sensor in this vicinity has a detection, its estimator distribution will have the identical shape, due to Bayes rule. That is, the probability distribution of an observation

originating from points in the region of the sensor will have the same shape and size as the forward diffusion distribution. The distribution will be centered around the sensor with a detection. In the second case when constant wind is introduced, the same rule applies but now the forward distribution forms an “ice cream cone” shape, where the intensity represents the likelihood that a particle originating at the source will be located at various positions in the vicinity of the source. The difference when wind is introduced is that the estimator shapes form a mirror image of the forward distributions. This holds true for a wind series Θ in which the wind shifts.

The estimator region will be a mirror image of the forward distribution. This estimator region boundary is determined by the threshold of detection, and the entire area of this estimator is defined as the TAD, or the total area of detectability. Similar to the forward advection diffusion problem, the shape of the TAD is determined by diffusion constant D and wind history vector Θ . This means a source originating outside the TAD for a sensor could not have produced an observation. Conversely, for a single detection and corresponding TAD the source could have originated anywhere within the area defined by the TAD. Areas within the TAD having higher likelihood values are more likely source locations for the event producing the observation.

Figure 2.10 illustrates the forward prediction of a hurricane track. This distribution is analogous to the TAD, except the TAD is an inverse likelihood corresponding to events occurring in the past. In the hurricane example, if a sensor detected the presence of a hurricane at one location, the TAD for that sensor would estimate the likelihood of all locations from which the hurricane could have originated.

Single Observation Estimator

A single observation, z_t , can be used to produce an estimator based on Θ , the wind history vector at t . Under the assumption that a single source produced the observation, the sum of the area of the predictor is unity. The wind history vector is defined as the history of wind directions observed at a point over the time range of observation history. The wind history vector is defined as the set of values:

$$\Theta = \{\theta_1, \theta_2, \dots, \theta_T\},$$

where each element of Θ represents a wind direction at a previous time. If these wind vectors θ_t are placed tail to head in sequence, a sum of vectors represents the history of wind until present time. Figure 2.11 shows the resulting estimators formed from a single sensor under different wind history vectors (Θ). An observation will produce a likelihood function in two dimensions, which is

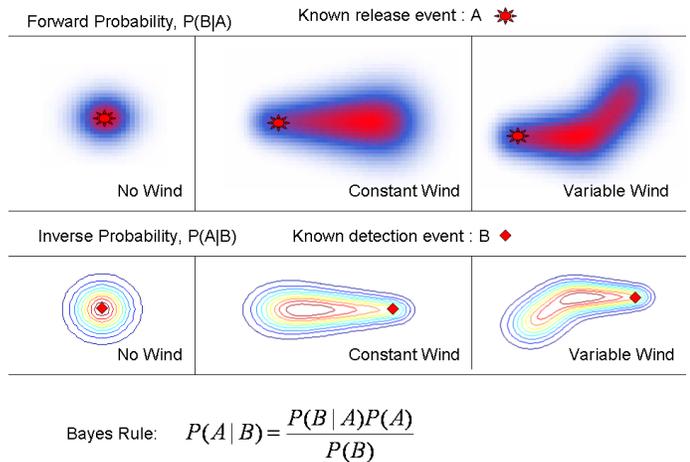


Figure 2.9: Estimator shapes derived from Bayes rule, showing probability of an observation originating from points in the region. These functions are a superposition of series of solutions to the forward advection-diffusion equation, but inverted in time.

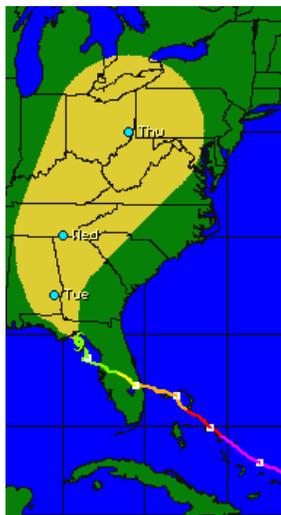


Figure 2.10: Hurricane track expressing probability of landfall at future times [7].

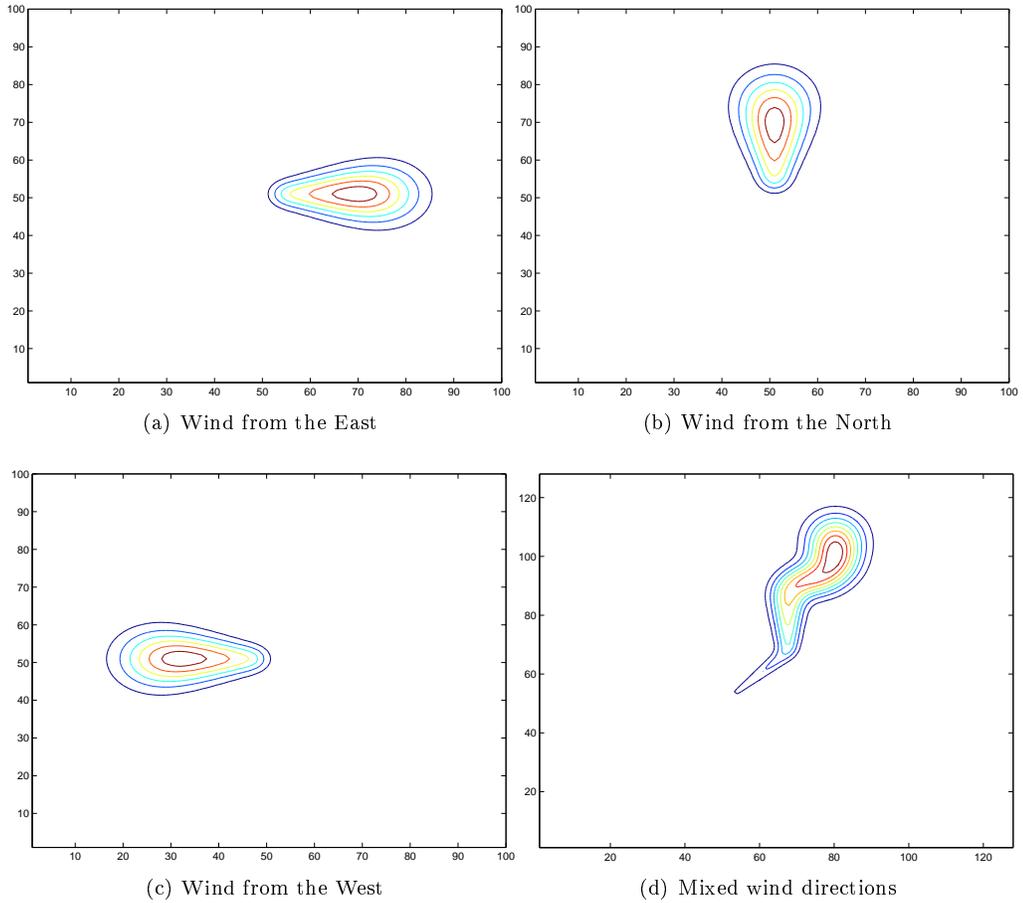


Figure 2.11: Inverse Plume geometry, a result of the diffusion and advection process run back-wards in time — which produces the TAD (Total Area of Detectability). The sensor is located at $(x=50, y=50)$. The contour value indicates the probability that a detection at this sensor originated from a point within the contour. Points outside the contour are below the threshold of detection. The shape depends on wind history Θ and diffusion constant D .

based on the inverse wind directions θ_t observed at that node from $t_0 \rightarrow t$. This region represents the likelihood that an individual z_t originated at a given upstream location. In the likelihood intensity plots, intensity represents this likelihood of attribution for z_i . Probabilities are higher along the axis of Θ , fading tangentially in a Gaussian distribution. Locations outside the TAD for a given Θ and D have an intensity value (likelihood) of 0 (white in the plots).

Multiple Observation Estimators

When multiple sensors in a region have simultaneous observations they must be combined to leverage all available information about the plume source. The most simple case of multiple estimators is illustrated in Figure 2.12 which shows the superposition of two individual estimators

to form a unified estimator. The estimator functions for any number of multiple sensors can create a combined likelihood map when they are superimposed, as seen in Figure 2.13. As more z_t observations are provided from sensor nodes, a greater degree of situational awareness develops. The state estimate of plume source location, \hat{x}_t , can be updated and refined as more observations arrive. The problem, however, is that without knowledge of the number of sources, a uniform estimator in which estimators from all sensors are equally weighed must be used to calculate joint probabilities. If this assumption of a single source is made incorrectly, the regions fail to be true probabilities and instead represent only likelihoods.

By estimating joint probabilities between multiple sensors and different TAD area, overlapping regions within boundaries provide additional information. Figure 2.14 shows the case of three sensors (A, B, C) with corresponding TAD regions with overlap. By calculating the joint probabilities of two or more TAD regions we can calculate joint probabilities of a plume origin with higher certainty. If multiple sensors have detections from overlapping TAD regions, we expect a lower probability of false detection for all current observations. For the sensors A and B , and the plume origin x :

$$P(A, B|x) = P(B|A, x)P(A|x) \quad (2.17)$$

The chain rule of conditional probability gives the following results for three sensors (with corresponding cones) A , B , and C :

$$P(A, B, C|x) = P(A|B, C, x)P(B|C, x)P(C|x) \quad (2.18)$$

And this extends into a general result for k cones:

$$P(A_1 \dots A_k) = P(A_1|A_2 \dots A_k)P(A_2|A_3 \dots A_k) \dots P(A_{k-1}|A_k)P(A_k) \quad (2.19)$$

Observation Correlation

In the case where more than one source may exist, we not only wish to estimate source location from a set of observations, but also to make estimates for which groups of observations belong to the same source. This process is known as track building. Although a sensor may be activated by

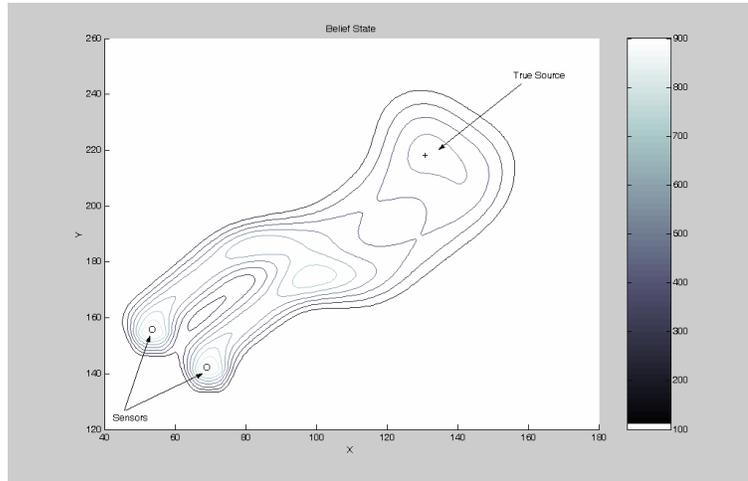


Figure 2.12: Belief map generated when estimators from two sensors are superimposed with equal weight .

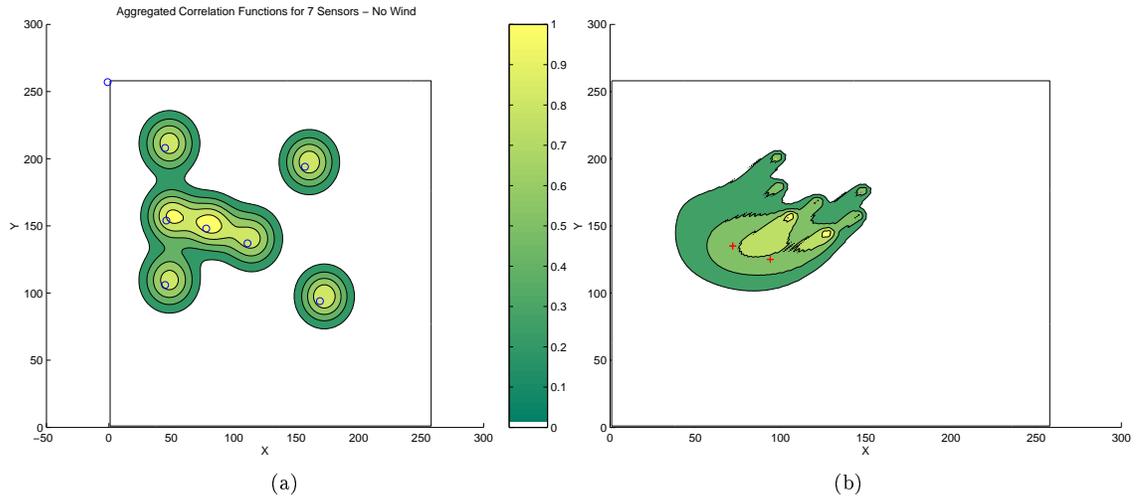


Figure 2.13: Superimposed sensor estimators for two networks of sensors. In (a) no wind, and in (b) with wind. The unidirectional likelihood distribution in (a) results from no wind. Because several sensors are positioned within the estimators of neighbor nodes, connectivity relationships can be created. The estimators also allow the calculation of connectivity values between neighboring sensor nodes.

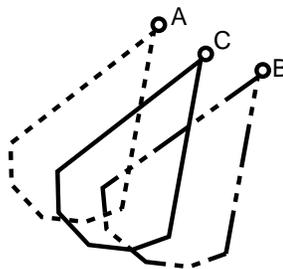


Figure 2.14: Combination of observations A B and C as their estimator regions overlap.

agent from multiple sources at the same time, it is possible to create a correlation function which estimates the probability that two independent observations are derived from the same target (source). The correlator function is based on the same principles as the estimator, but a value is calculated for two sensor node positions. Figure 2.15 illustrates the calculation of a correlation value between two sensor nodes. By estimating the previous path of a current detection at a sensor node, we determine if two activated sensors are located in the propagation path of an evolving plume. If the calculated correlation value is above threshold, an association is formed. The “strength” of this association is given by the correlation function which operates on two sensor locations and their corresponding wind history vector Θ .

When observations are available from the sensor network, the current wind vector history for the relevant node is weighted with the observation. Thus, a large number of observations at a particular node gives more weight to its particular correlation value with another node. Once a series of these correlation functions are calculated against neighbor sensor nodes, associations can be formed between nodes. These associations allow the creation of “tracks.” A track is defined as a collection of nodes all believed to produce observations from the same source.

Each track corresponds to a partition of the total observation set, and each track can be used to calculate an independent state estimate, also called a map. This map represents the likelihood of source locations that produced a collection of observations. When a large number of sensors are available in the region of interest, as N increases, the map converges to an approximation of the true sources. The cumulative probability of a track is the product of all the correlation values within the track.

For example for a track containing three sensor observations (A, B, C) with associated probabilities $P(B|A)=.5$, $P(C|B) = .1$, there are two sensor associations between the three points, and a cumulative track probability of $.5 \times .1 = .05$. This track probability assignment method can be applied in the multiple hypothesis tracking method as a way to rank competing hypotheses that explain the same set of sensor observations.

2.2.3 Multi-target tracking theory

The uniform estimator can be improved with advanced data association (DA) techniques such as multiple target tracking (MTT). MTT methods are well developed in the literature, and originally found use in applications for radar target tracking [44]. Collections of observations are partitioning into *tracks* believed to represent the same target. Figure 2.16 illustrates observations collected over

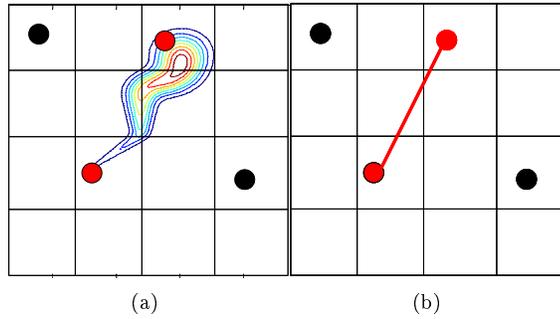


Figure 2.15: An association is formed between two nodes when the correlation function exceeds threshold. This calculation performed between two nodes determines whether a node is appended to a track as in (a). In this case a track is formed in (b) consisting of two nodes.

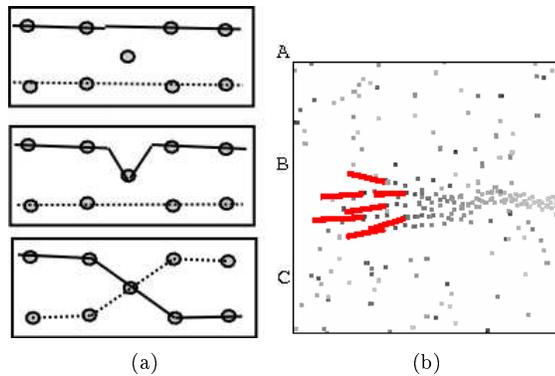


Figure 2.16: In (a) the same observation set produces three different hypotheses for observations from two targets (solid and dotted lines) collected at five time-steps. In (b) new observations (gray) may be assigned to existing tracks (red), or initiate new tracks.

five time-steps for two targets where three different track combinations are possible. By maintaining multiple tracks as explanations for the same observation set, observations received at a later time can be included to alter the ranking of these track likelihoods.

Multiple possible assignments of observations also creates the problem of exponential growth in *hypothesis sets*, where a hypothesis set is one particular collection of tracks. This large growth rate in hypotheses must be dealt with using pruning methods. MHT (multiple hypothesis tracking) handles this combinatorial growth of possible track assignments via accurate pruning and track maintenance. In Figure 2.17 the process of MHT contains a step known as track prediction. In this step, currently existing tracks are estimated for future time-steps in order to facilitate optimal observation assignment from new observations. Traditional target tracking for systems such as airplane radar use a technique known as Kalman Filtering, but this thesis develops a custom predictor tailored for plume tracking.

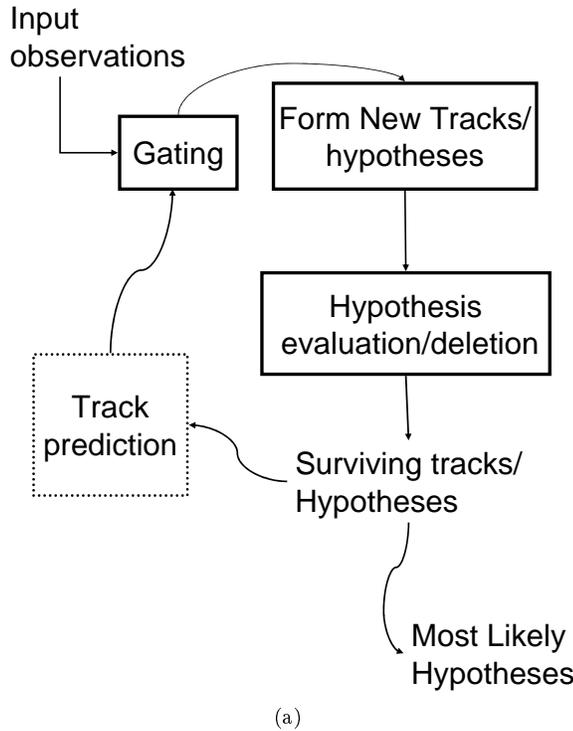


Figure 2.17: MHT process, and steps. Data collection, gating, track maintenance, and association. Track Prediction as a part of the MTT algorithm: assignment of new observations to existing tracks or the decision to create to a new track requires track prediction based on the last known observations.

2.2.4 Process Query Systems (PQS)

The process query system approach developed at Dartmouth College is based on multiple hypothesis tracking techniques introduced in the previous section. PQS is designed as a data filter that identifies the existence of models in a data stream, similar to the classic notion of filtering. In a filtering problem the goal is to accurately match an observation sequence with a model, recovering and identifying the presence of that model in the data. In this filter we desire high correlation even in the presence of noise, and a low likelihood of a match when the model is not present in the data. The challenge of a PQS system is the creation of a generic framework for disambiguation, detection, tracking, and state prediction of multiple discrete and/or continuous stochastic processes in noisy and lossy environments [1, 12].

Instead of domain experts spending large amounts of time identifying the presence of a process and data, expertise can be spent designing models for processes – which then are continuously

compared to arriving data. Examples of process models include Internet worms, social network activity, and physical target tracking such as humans or fish [1]. The plume tracking application is simply one more application of the more general problem of detection of a stochastic process. The approach is a good match for the plume source detection problem for the following reasons:

- lack of continuous observations,
- noisy observations,
- potential delayed or missed observations from simple sensors,
- the presence of multiple competing hypotheses,
- and the real-time nature of the plume problem.

Chapter 3

The problem and solution

“The true logic of this world is in the calculus of probabilities.”

-James Clerk Maxwell

3.1 Problem definition

3.1.1 Graphical representation of problem

The illustration of graphical conventions used to visualize plume location problems are located in Figure 3.1. The components used in the simulations include plume sources, agent sensors, descriptions of the plume agent in a region, tracks which are groups of sensors, and estimates about the agent – in the form of likelihood maps. The following graphical symbols and values are used:

- Region A . A is $m \times n$ area that may contain plume sources, sensors, and wind. This is the total area under surveillance, and contains $m \times n$ cells.
- (+) Plume sources. A plume source may be centered at any cell A_{ij} , and with an arbitrary number of cells. At each time step plume material is added to an area of width w , centered about A_{ij} .
- (o) Binary sensors. capable of detecting agent from plume source (+) located at one cell A_{ij} within A .
- (*) Filled circle indicates an activated binary sensor

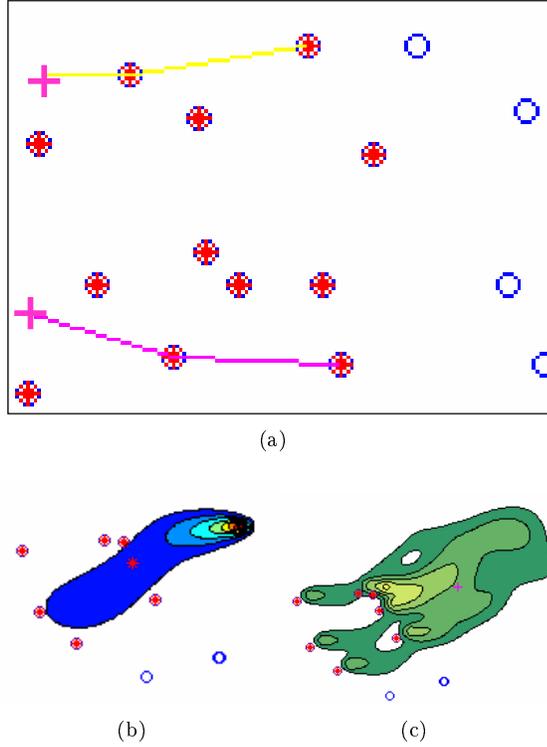


Figure 3.1: Graphical conventions for sources (+), sensors (O), sensors with detections (filled O), and tracks (lines). $S(A)$ state matrix in (a), and $M(A)$ likelihood map of region A in (b).

- $S(A)$, the state of A , is defined by the concentration of all cells within A , values are indicated by a blue concentration gradient.
- $M(A)$, a likelihood map of A , indicate likelihood of sources as seen by some set of active sensors. This map will change based on the method used by the sensor network, and which active nodes are used for the estimation. $M(A)$ intensity is indicated by a green concentration gradient.
- Tracks are a clustering of active sensor nodes, believed to detect the same event. Each track has a different color and consists of lines connected between sensor nodes members of a the track. It may be linear or have a tree structure.

3.1.2 Formal problem statement

Let A be a two dimensional area, $A \subset \mathbb{R}^2$, discretized into a regular grid of $m \times n$ cells. Define the state of A at time t , $S_t(A)$, as a matrix containing the concentration of an airborne agent in each of the $m \times n$ cells at a given time t . Each cell within A is indexed by (i, j) , where i and j

are the column and row indices of the cell. The concentration for a specific cell at time t , $C_t(i, j)$ is defined as the average mass of agent per unit area in the cell. Time is discretized into uniform intervals of length Δt , where k indicates the k^{th} interval of time. We assume $k \geq 0$.

$$t_k = t_{k-1} + \Delta t \tag{3.1}$$

$$= k\Delta t, \tag{3.2}$$

optimal

for $k \geq 0$.

Next we go about modeling the addition of plumes to A with a release matrix, R . This matrix R serves as a perfect record of all plume releases that occur in A during a time interval of interest. Assuming we do not know the time and location of the release events a priori, the goal becomes to reconstruct R^T as closely as possible. In this model we assume each cell within R is binary. This binary release matrix R_{t_k} represents the state of “release” or “no-release” for each cell (i, j) in A at time t_k .

$$R_{t_k}(i, j) = \begin{cases} 1 & \text{for all cells in "release state" at } t_k \\ 0 & \text{for all cells in "no-release state" at } t_k \end{cases} \tag{3.3}$$

The set R^T is composed of all available R_{t_k} matrices and fully describes all releases in the time interval $0 \rightarrow T$. Sources can oscillate over time, transitioning between the “release” and “no-release” states, creating pulses of agent over time. Each cell has an independent probability of being in the “release” state.

In reality the sequence of states R^T is hidden from observation, but we estimate R^T as closely as possible. Knowing R^T or at least a close approximation of R^T leads to the set of all plume release sources within A within time $0 \rightarrow T$. More advanced models of R could include a higher number of states for each cell.

When a plume is released within A , the agent is released into cells by adding concentrations to release points. A plume may have an arbitrary size, and be released into any number of cells in a region. A release incident somewhere in A at t_k is modeled by adding an integer amount of mass to cells of the previous state $S_{t_{k-1}}(A)$. The size of an incident is determined by the number of cells that are injected with agent, as well as the magnitude of agent added to each of the affected

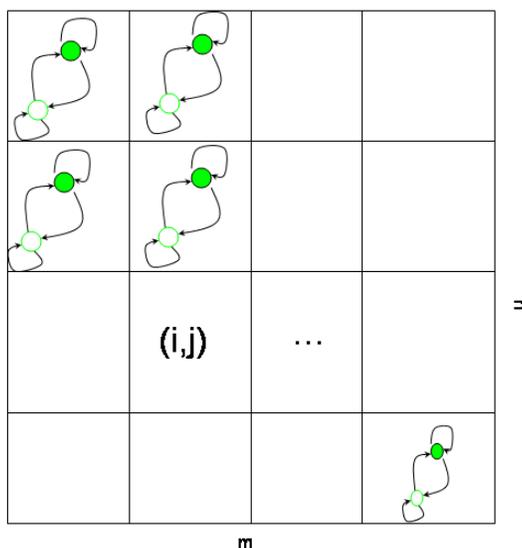


Figure 3.2: Finite state of each cell (i, j) exists in neutral state or plume emission state. Green dots represent the emission state, empty dots represent neutral state. The release matrix R^T is a set of all the possible release points over a time interval T .

cells. This model allows an incident to be of low concentration but covering a large number of cells, or of high concentration and very localized. Across time, events may be “continual release” where release point cells are injected at every time interval, or a “limited release” where injections occur at a single time t_k .

The eventual goal of this work is to determine $R_{t_k} \in R^T$ for times in which a release occurred. That is, we will estimate values (i, j) and t for which $R_t(i, j) = 1$. $R(i, j)$ is the release history over all available time for a specific cell, whereas R^T is the entire release history over the entire $m \times n$ space until current time T . $S_t(A)$ is a function of its initial state $S_0(A)$, sequence of release states R^T , as well as the physical processes of diffusion and advection operating on S_t , denoted $F(S^t)$. Next we provide a more detailed description of the process $F(S^t)$.

Evolution of $S(A)$ dynamics equations leading to $F(S_t)$

Next we develop a model for the evolution of $S(A)$, and dynamic equations leading to $F(S_t)$, from first principles. Given $S_t(A)$, $S_{t+1}(A)$ is determined by the physical processes of diffusion and advection (wind). Before tackling the inverse plume problem a mathematical understanding of the forward diffusion process is needed. This forward model provides simulated data of diffusion events until actual field data can be collected for solving the inverse problem. In addition, developing a forward model and running simulations provides fundamental insights into the characteristics of

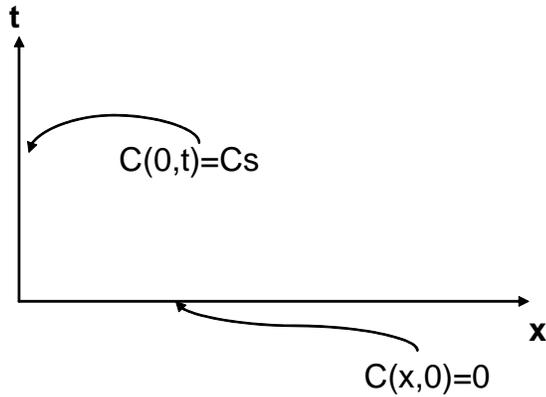


Figure 3.3: Boundary conditions used for numerical solution of advection-diffusion equation

the diffusion process.

At each time step, $S(A)$ evolves according to the diffusion of all particles within A . This is achieved with a forward numerical approximation of the diffusion equation with a central difference approximation. This means the values in a cell at the next time iteration are a linear combination of all the neighbor cells at the previous time-step. In order to solve the second order Fick's Law with numerical approximation we require one initial condition and two boundary conditions. Fick's Law can be solved under two basic conditions: infinite source diffusion and limited-source diffusion. This simulation forward problem can approximate either case. In the infinite source case material is released from the source continuously, whereas in the limited-source case material is only released during an short interval of time.

Using these boundary conditions, the agent will disappear when it approaches the edge of A . The infinite source diffusion boundary conditions are:

$$\begin{aligned} C(x,0) &= 0 \\ C(0,t) &= C_s \\ C(\infty,t) &= 0 \end{aligned}$$

In addition to the diffusion process, a uniform wind force Θ is applied to A at each time step which is uniform in direction and magnitude across A . The physical process state evolution function of $S(A), F(S)$, can be divided into these two components with relative magnitude D for diffusion, and Θ for wind. As Θ increases in magnitude, the relative importance of diffusion decreases. A

typical value of the ratio $\frac{\theta}{D} = 10$. This ratio is known as the Peclet number, and is unique for different materials and gases.

Θ may change over time in direction or magnitude, approximating a shifting wind. This means $S_t(A)$ at an arbitrary time evolves as a function of $S_{t-1}(A)$ as well as Θ and D . $S_t(A)$ is also determined by all the previous injections of chemical into A by the sources. $F(S)$ is the combination of the physical processes of wind (a linear additive constant) and the diffusion model developed above. This complete approximation of A and its time evolving state $S(A)$ will serve as our approximation of a region A , where $F(S)$ is the physical process model, and R is the release state.

$$S_{t+1} = F(S_t) + R_{t+1}. \quad (3.4)$$

The introduction of a realistic wind time series in simulated data is critical for estimating real world conditions. Historical wind data from 1991-present is available on the Internet from the National Weather Service. Stations are located on waterways, buoys, and land stations. Typical continuous wind data is sampled every 6 minutes and includes direction, speed, as well as temperature and humidity values. By taking advantage of this large preexisting data set in a large number of geographic locations, the tracking-based approach can be tested with urban, ocean, and open space wind conditions.

Description of sensors and observations

This area of interest is populated with sensors, which report concentration at their location and local wind. More specifically, A will contain some number of sensors N , where in general $N < m \times n$. Each sensor within A corresponds to one cell located at position (i, j) . Each of these N sensors within A reports an observation Z_t at time t whenever it detects the released agent at a concentration above some predetermined threshold C_{thresh} . For each sensor observation, $Z = (x, y, c, t, \theta, D)$, which consists of location (x, y) , concentration c_t at time t , absolute time t , wind direction and magnitude θ , and diffusion constant of the agent D . $Z(x_i, y_j)$ corresponds to all the observations available from a specific cell (x_i, y_j) over all time. Sensors will report a series of observations over time to a central processing location, thus producing a time sequence of observations Z^T , which is the set of all available observations up to the present time T . An individual sensor observation, $Z_t(i, j)$, is defined by:

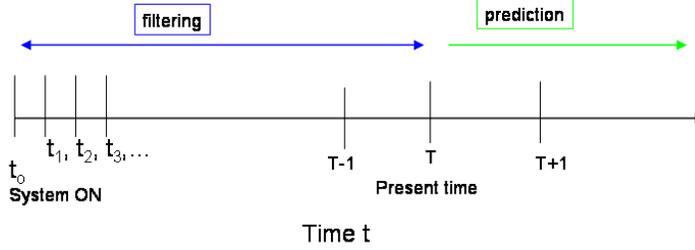


Figure 3.4: Time definitions in our model

$$Z_t(i, j) = \begin{cases} (x_i, y_j, c_t, \theta_t, \bar{D}) & c_t \geq C_{thresh} \\ \emptyset & c_t < C_{thresh} \end{cases} \quad (3.5)$$

Problem: Given Z^T , what is the probability of S^T ?

Given such a sequence, Z^T , we intend to estimate $P(S^T|Z^T)$, where S is an arbitrary state at time t , with

$$t_0 \leq t \leq T.$$

In the case where $t < T$ this is a filtering problem, operating only on currently available observations. This allows estimation of $S_t(A)$ at any arbitrary time prior to the present, and an approximation of conditions that led to the current Z^T . We desire to know S_t for times near times within R_t in which $R_t = 1$ (releases generated at R_t). The state S_t at the time of plume release t within R is an approximation to the source location for $R(i, j)$. By estimating $S_t(A)$, we estimate when and where releases occur. It is also possible to recover t_0 for a particular release – that is, the time of initial release. Another possibility is to predict future states of A , that is estimate $P(Z^T|S)$, where $t > T$. In this region of time, the problem is posed as prediction. These estimations of $S(A)$ have the ability to predict both forward and backward in time from the current observation set Z^T .

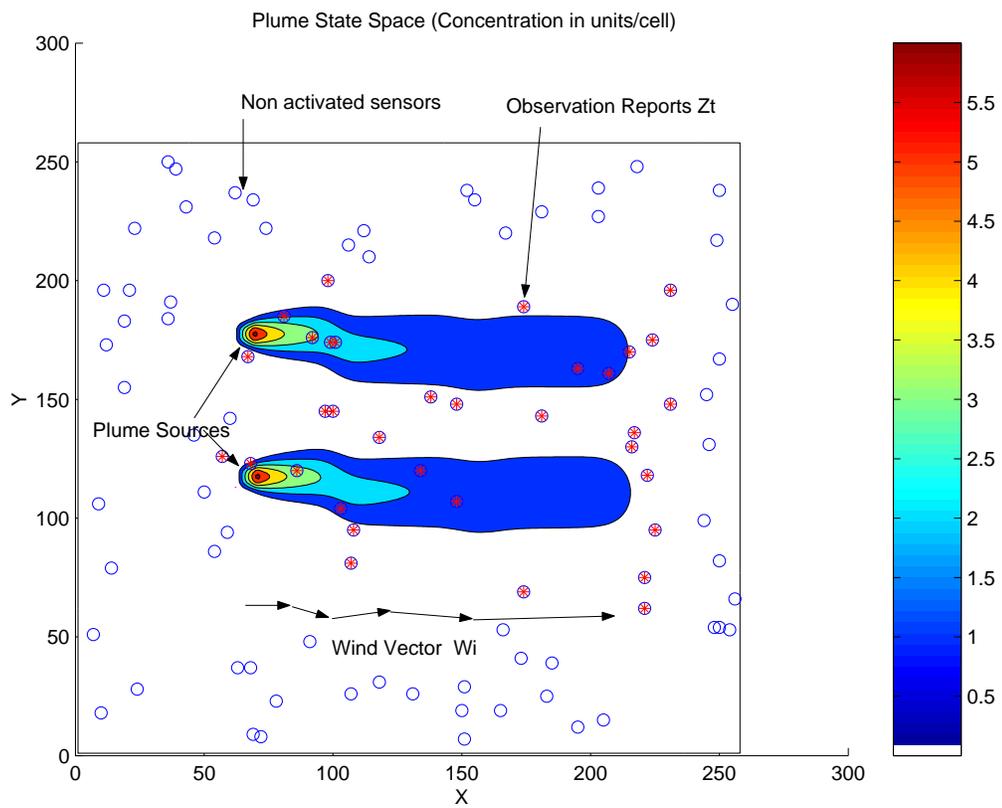


Figure 3.5: Formulating the solution of the Plume Problem with appropriate variables

3.2 Observation correlation in target tracking

A purely analytical solution to the inverse source location problem is not realistic, and we therefore seek to form likelihood associations between sensor observations. Instead of attempting a numerical solution to the inverse advection-diffusion equation with observed sensor concentrations, statistical likelihoods are estimated between sensor nodes. For example, given a positive observation at one sensor, what is the likelihood that the event triggered at this sensor also triggered its neighbor sensor? By estimating or assigning some likelihood value of connectivity between sensor pairs, a connectivity map can be built.

The standard Kalman predictor used in target tracking is not designed for large continuous mass; we require a custom predictor based on simple binary sensor information [8]. The percentage coverage for an area can be seen as the ratio between the TAD and the total area of interest:

$$P = \frac{\sum A_i}{m \times n}$$

for an area of size $m \times n$. The area $\sum A_i$ is the sum of the TAD areas based on detection threshold and wind history. This is the probability of detecting a random source located within A .

We present a method for monitoring plume sources with a MTT-like algorithm, maintaining tracks from collections of individual observations, and tracing observations back to their origin. Using the concept of tracks we can select an optimal subset of sensors, count number of sources, and deal with crossing paths of plumes. Unlike the canonical MTT problem which utilizes Kalman filtering, we can measure all the forces having an impact on the plume structure, whereas with traditional MTT the target may have an intelligent unpredictable component such as a pilot control [44, 4, 47]. Knowing the wind history vector Θ for each of the N nodes, the substance of interests diffusion constant D , and the relative location of all the nodes in the network, we may calculate a plume predictor value for each new observation. Each observation z_i at sensor n has a probability of correlation for being observed at a different sensor. The plume predictor estimates this probability, and uses the value in observation assignment to tracks, or track initiation. Essentially we ask, what is the probability that z_i and z_j originated from the same plume event?

The goal is to find source locations and rank their likelihoods, which will allow the estimation of the number of sources. How well do current observations Z^T in the sensing network correlate to the same original event (S_{t_0})? We propose the monitoring of a two dimensional area ($m \times n$) with a field of N stationary sensors, where N is large but much less than $m \times n$.

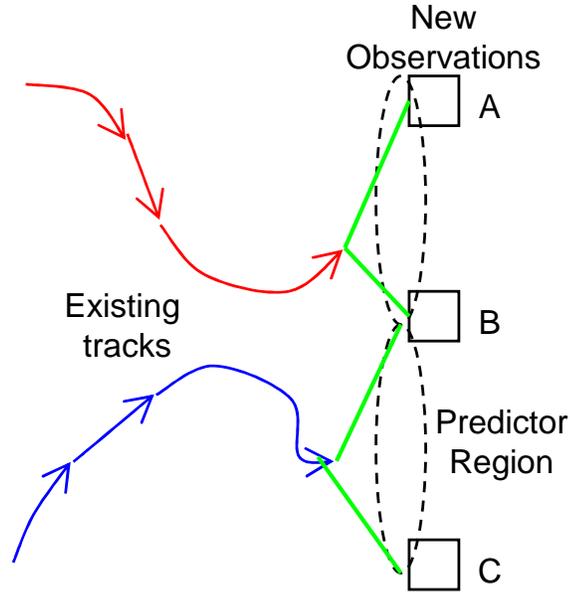


Figure 3.6: Data association and plume correlator for MTT

In our model, wind and diffusion across the the two-dimensional space are assumed uniform. Our model of a chemical plume will consider only the diffusion constant D and wind W , which can be observed at each sensor of known location. Given the observation of a known chemical at the sensor allows the sensor to look up D . Because the wind medium can be measured at each node, the sensor network has complete knowledge of the only two parameters affecting plume dynamics in our model.

The goals of our approach include:

- estimating the plume source location,
- determining plume tracks,
- having few false positives,
- having a scalable algorithm,
- and having near real time tracking ability.

3.2.1 Role of gating in MTT

The goal of this work is not improving upon MTT itself, but developing several plume predictors, which can then be inserted into existing implementations of MTT algorithms such as PQS

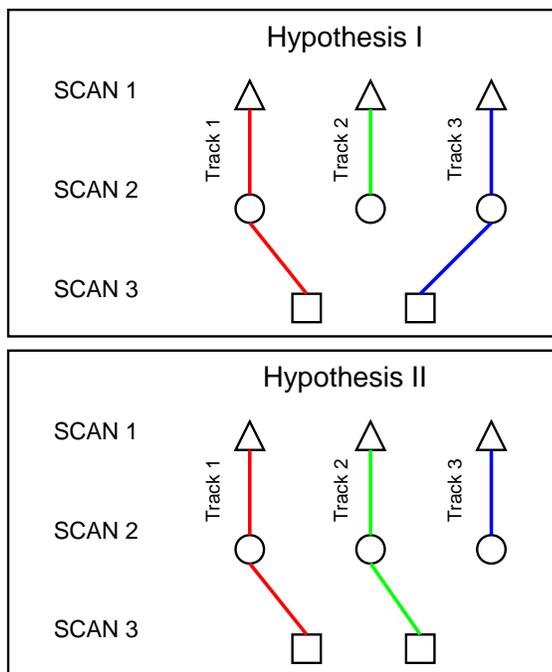


Figure 3.7: The role of the correlator in MTT is observation assignment. Given the same set of observation scans multiple track permutations and hypothesis sets are possible.

[44]. The function of this plume predictor is the statistical correlation of individual observations, and the assignment of new observations to tracks. This approach supports the eventual development of multiple high level models of a dispersing chemical plume. High level models will enable the development of end users to submit plume query process models. Once these models are developed, several will be simultaneously submitted to a PQS framework.

We define a plume as a region with a center of mass, with ideal observations expected to be based on the diffusion equation and distance from the plume center. In practice however, plumes split into separate discontinuous filaments - moving in a chaotic flow regime. The plume may generate low readings near the source, or intermittent high concentration readings to sensors at great distance. The use of Gaussian descriptions for atmospheric dispersion models can give rise to very misleading estimates of concentration fields. A plume is frequently not well dispersed, but rather consists of a long sinuous volume of material. As a result, a detector with a fast response time will report a series of relatively short bursts of high concentration adjacent to long intervals during which the concentration is close to zero. Detections are essentially binary in nature. This property of plume dispersion called intermittently results in concentration readings of zero for many samples, even for sensors near the source [30].

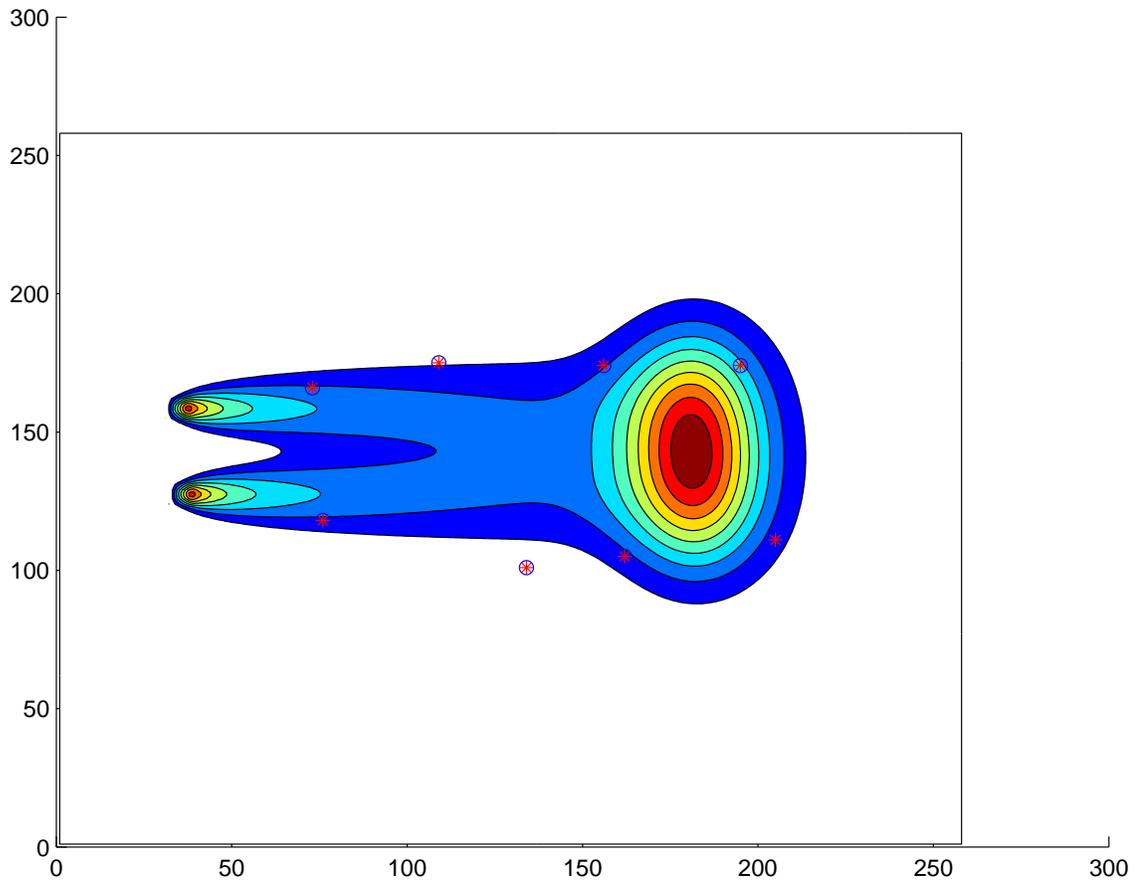


Figure 3.8: Source separation problem with two sources on the left, and constant wind to the right. The two tails combine to form a third peak concentration area which is blowing to the right. The objective of this sensor field is to correctly identify the presence of exactly two sources, however without sensor assignment to tracks this problem is very difficult.

MTT for plumes addresses the problem of assigning new observations to existing track hypotheses, or new hypothesis creation. (data association). Kalman or Bayesian filter predictors assume a strict temporal arrival order of observations and may have to throw out late-arrival data. The tracker may have to repeat calculations to integrate the late-arrival in prior calculations, thus degrading performance. MTT maintains a ranking of data associations, based on likelihood. MTT suffers from combinatorial explosion in data association, however additional knowledge about target kinematics or the environment can help prune hypotheses [51].

3.2.2 Source separation

Figure 3.8 demonstrates the classic problem of source separation in the domain of plume tracking. This example shows a possible scenario in which simple peak detection can lead to a totally erroneous estimate of source location, due to the superposition of two sources at a downwind location. One major advantage of the tracking-based technique is the ability to separate observations into partitions which minimize the effect of such superposition, essentially allowing for the detection of plume signal in a highly noisy environment. Track formation and the 2-step algorithm are not possible without MTT techniques.

3.3 State estimation

3.3.1 Plume state space estimation

State space likelihood maps are generated based on a backward propagation of likelihood from each sensor node having an observation at the same instant in time. As time evolves the likelihood map is updated, and without tracking we assume an equal weight for each observation. We assume that each sensor has knowledge of the entire wind history since t_0 , where t_0 is defined as the beginning of the simulation. Sensor backward propagation is performed from $t_i \rightarrow t_0$, where t_i denotes the current time step. Therefore sensors having the most recent observations will have likelihood maps enclosing upwind sensors, since they propagate their predictor functions for a longer duration than the initial sensor observations. This update process leads to likelihood map refinement in most cases as new observations arrive downstream. Likelihood maps can be calculated for any t_i , however the final map is used for comparison which allows direct comparison between likelihood maps based on the same number of time steps. In general the final time step likelihood

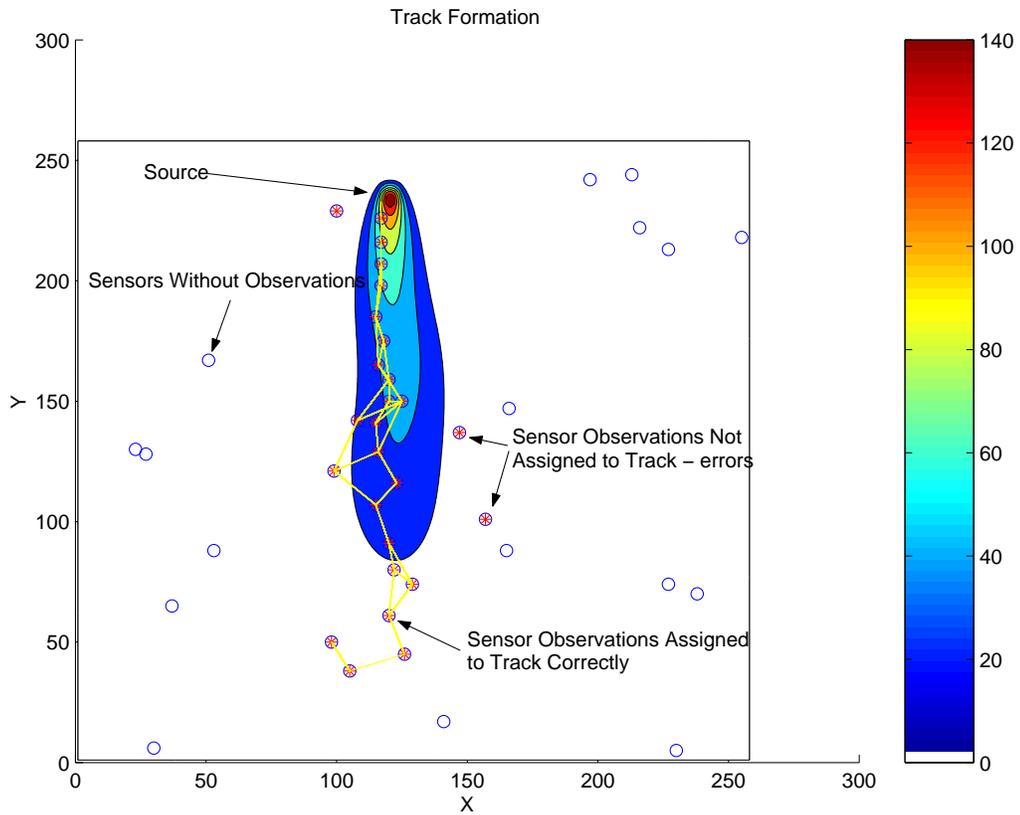


Figure 3.9: Track definition used in this thesis. Observations believed to originate from the same plume event are assigned to a single track. In this Figure solid circles indicate sensor observations, and a single plume source is released from the top center of the grid. Most observations in this case are correctly assigned to a single track. The track objective is the partitioning of all observations into subsets which contain the most information about a single source. Once an inverse likelihood function is applied to this subset of observations it will in general produce a more localized likelihood map.

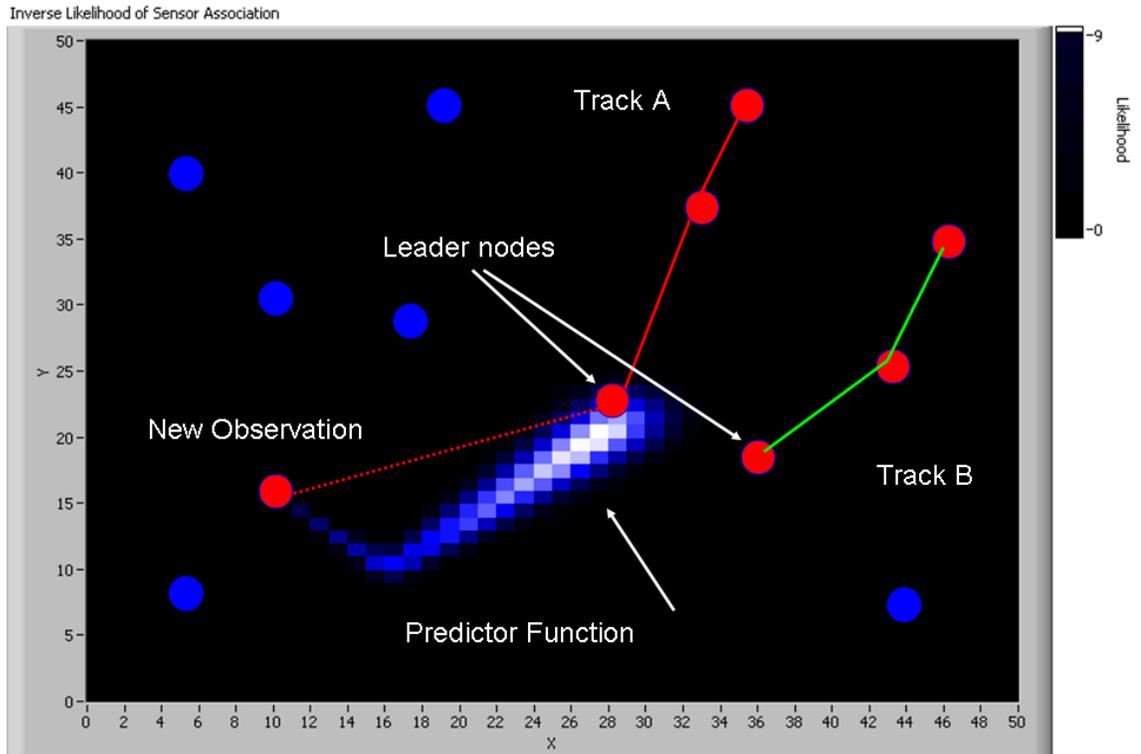


Figure 3.10: Data association process in the 2-Step algorithm. New observations are compared to all leader nodes with their unique predictor functions. An assignment to a track is made if the leader node location evaluated on a predictor function produces a score above threshold.

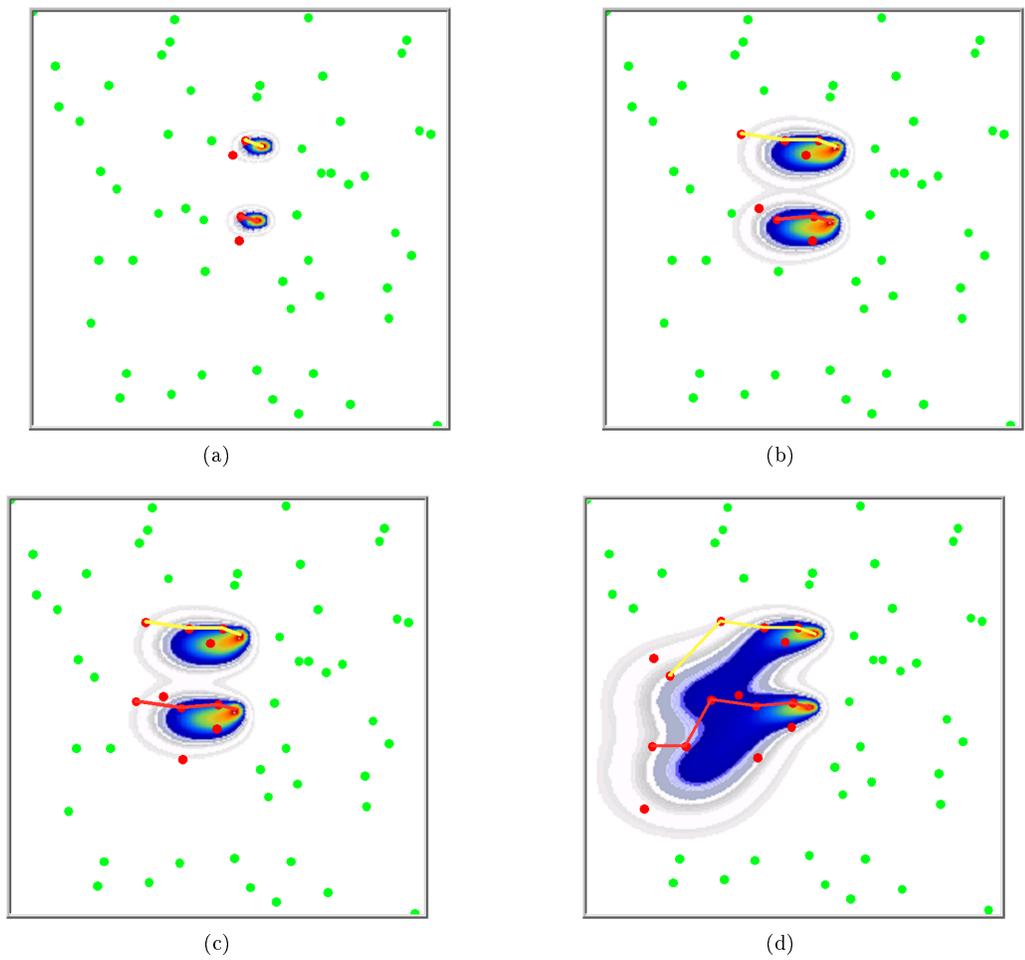


Figure 3.11: Growth of 2 tracks over time from 2 sources at a single release time. At the end of simulation run, sensor nodes within each track will form a single likelihood map, with the assumption that all members of the track are observing the same source. This partition of the entire collection of observations produces improved likelihood maps. Tracks evolve in the sequence (a), (b), (c), then (d).

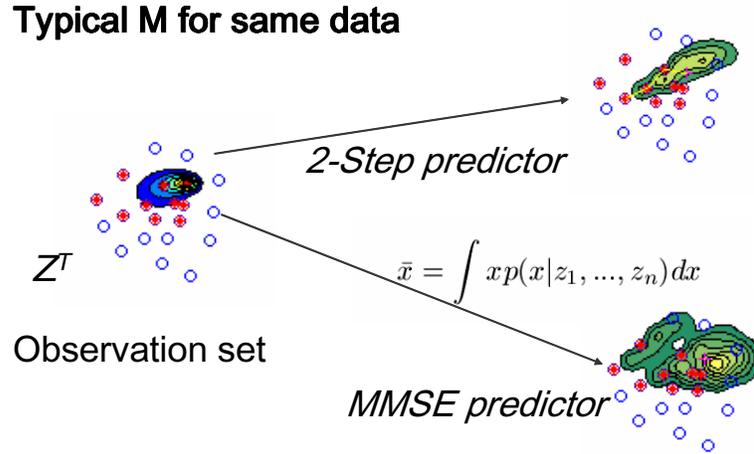


Figure 3.12: Likelihood map $M(A)$ for for 2-Step estimator compared to uniform estimator.

map for a trial is the most accurate.

Joint probability, likelihood and state estimation

Likelihood differs from probability or joint probability, and we use the concept of likelihood rather than a strict definition of probability in the state estimation procedure. This likelihood (not a probability since the assumption can not be made that all detections are coming from the same single source) estimates how likely a current observation originated at a location upstream. In the case of no wind this estimator is spatially uniform. A likelihood map $M(A)$ as seen in Figure 3.12 illustrates the predictor used in the 2-step algorithm as a custom type of estimator. In this custom estimator an observation set Z^T is partitioned in a partition ω , which attempts to assign only observations which originate from the same source. This is unlike a MMSE estimator which does not first perform the partition based on tracks. The attempted result is a likelihood estimation only for observations belonging to the same source. The decision whether or not to include an observation in a track is performed by a correlator function, explained in the next subsection.

3.3.2 Correlator

The purpose of correlator function is to assist in accurate data association (DA) between observations. The basis for this function is Bayes rule, derived in the previous chapter. Given an observation z_t what is the probability of a given x_t . This can be viewed as in inverse solution to the forward advection-diffusion problem, since we are estimating source locations from plume

observations. Assuming the Bayesian nature of such a source and observation system, these inverse likelihood areas are calculated with an inverted wind history vector Θ' for a particular location (X, Y) . For example if a sensor measures a wind sequence:

$$\Theta = \{20, 23, 21\},$$

where these measurements are in degrees - the inverse wind history, denoted Θ' , would be:

$$\Theta' = \{21, 23, 20\}.$$

The inverse probability distributions are calculated by solving the forward diffusion equation with Θ' instead of Θ . The identical values for wind speed, and diffusion constant D must also be used. This analytical method requires using a $2D$ evaluation function operating on the correct linear superposition of exponential functions of the form:

$$L = \exp\left(-\left(\frac{x-b}{a}\right)^2 - \left(\frac{y-c}{a}\right)^2\right), \quad (3.6)$$

where L is the likelihood value, b is x_{ctr} c is y_{ctr} and a is time. A sequence of wind at a sensor, when inverted and substituted into this equation with the correct diffusion constant will produce a Gaussian region proportional to the probability that an observation originated from that point. The more elements included in Θ , the greater the number of functions to added, one exponential for each time step before the present time.

By providing the relative X and Y locations of the 2 sensor nodes in question, and then inserting the inverted wind sequence parameters into Equation 3.6, a correlation value is produced for these two positions. If the second sensor location is defined as location $(x, y) = (0, 0)$, Equation 3.6 simplifies to:

$$L = \exp\left(-\left(\frac{x}{a}\right)^2 - \left(\frac{y}{a}\right)^2\right),$$

where x and y represent the relative location of the second sensor. The sensor with the current observation is defined as the XY origin.

3.4 Optimizing location with 2-step algorithm

In this section the 2-step algorithm is introduced and explained as a more efficient type of estimator. For a description of estimators see section 2.2.2. The “2-step algorithm” consists of:

- 1) Assigning optimal observations to tracks
- 2) State estimation of source location for each of these tracks independently

Collaborative localization

Collaborative localization of target sources and tracking have the goal of making a good estimate of the target state X^T based on a measurement history Z^T . One popular method applied in sensor networks is known as sequential estimation. Typically the measurements z are not all available at once, but must be updated continuously over time. This method is applicable to the plume tracking problem in a sensor network since a dispersing plume will activate a sequence of sensors over time.

A group of sensors within a network assigned to creating and maintaining the belief state of a plume source over time will be called a *track*. In practice this collection of sensors has overlapping coverage of the state estimate of the plume source over some span of time. As observations arrive over time the belief state is updated based on observations from each member of the track. When multiple sources are present two scenarios are possible:

1. Sources are far apart and their tracks are processed by multiple collaborative groups working in parallel.
2. If plume regions cross, their belief state estimates may overlap, and the collaborative groups of sensor nodes are no longer distinct. The sensor measurements in the region of overlap may now be associated with either of the two tracks.

Track formation

The first step of the 2-step algorithm is a rule based procedure for track formation. Correlation values are calculated between current sensor observations and sensors with current associations to tracks. Depending on the rule, single or multiple nodes within each track may form associations with the new observation. In the most simple track formation rule, the track is linear, and only track terminus nodes are used for correlation value calculations with new observations. In the more sophisticated track formation rule, mesh or tree structured tracks are allowed in which

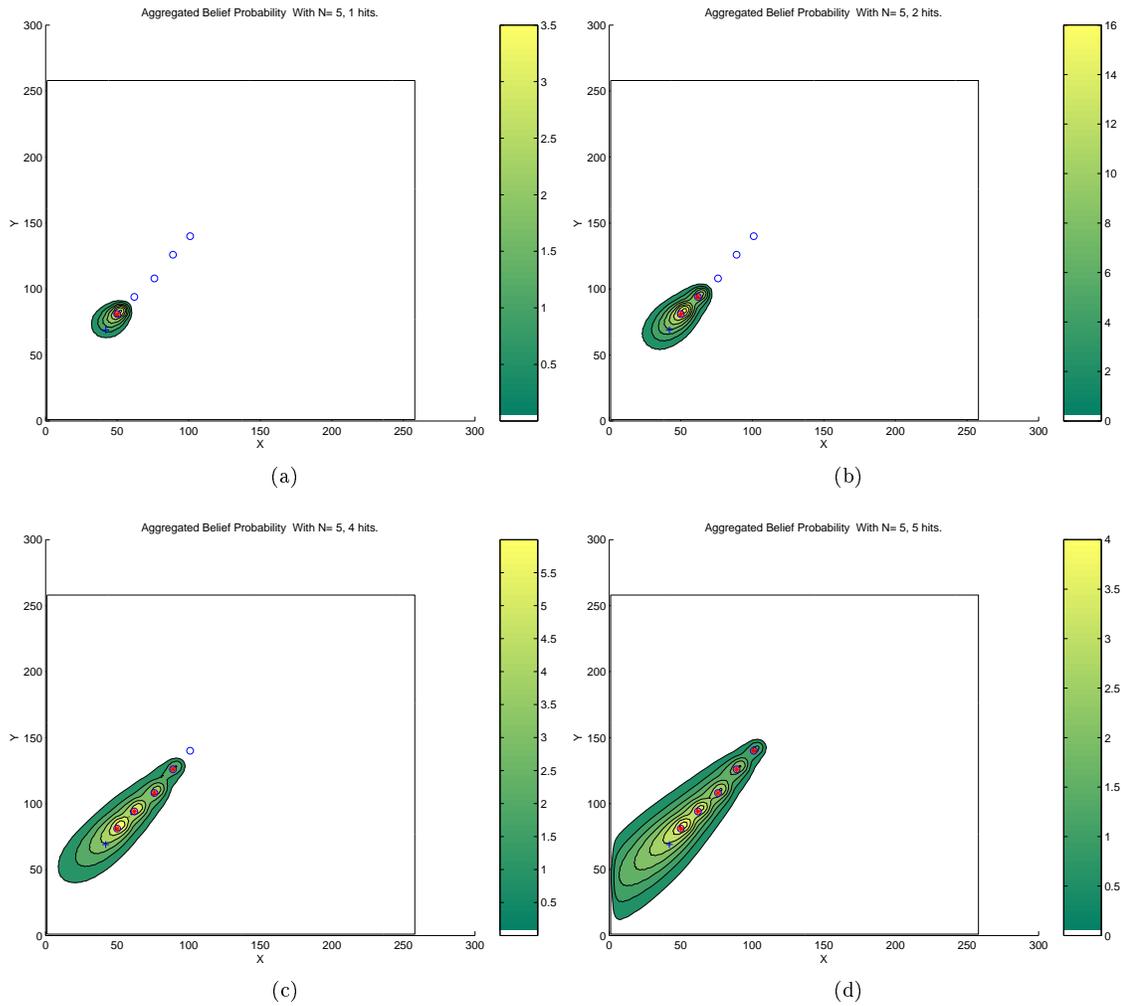


Figure 3.13: Sensor selection in a track. When new observations are added to the track they are used to update the state estimate. The estimation task here is to localize a stationary plume source labeled “+”. Circles represent sensors, and filled circles represent sensors whose measurements have already been incorporated in the state estimation. (a) State estimate after incorporating the leading sensor which has first detection. (b)-(d) State estimate after incorporating each additional measurement from a selected sensor.

new observations perform the correlation calculation with all node members of a track. For the preliminary experiments the following procedure is used for assignment of new observations to tracks, and the track building process. Later in chapter 5 a method will be developed that is useful when continuous track initiation is required.

Step 1: Track formation steps

- *Track Initialization* - Within the first 25 iterations of simulations all new observations create tracks. The terminal node on track is designated leader node.
- *Data Association* - All new sensors with observations calculate a likelihood function based on wind history. Correlation function evaluated at all leader nodes.
- *Track extension* - Observations that were associated in DA step become the new leader nodes.
- *Track termination* - The track is terminated once simulation ends or no new associations within cutoff parameter. Track outputs sent to Step 2
- *Likelihood map* - Each track sequence produces an individual likelihood map

Step 2: State Estimation An example of step 2 is illustrated in Figure 3.14. In this example only 4 sensor observations are used to form a belief map. The other observations in the sensor network are not included for the state estimation phase of the 2-step algorithm. With the progressive update of the state estimate, the estimate of the source location likelihood generally improves as new observations are associated to a track. This will be observed as higher and sharper regions near the true source location.

The motivation for separation of clustering and estimation steps is reducing the computational demands of calculating predictor functions for every sensor with an observation in the network. By selecting only nodes expected to return high information value we reduce the computational requirements, and also the accuracy of the likelihood maps generated. We expect performance improvement when this algorithm is applied compared to the uniform summation of the belief maps for *all* sensors with detections at a given time step.

Another benefit of first assigning observations to tracks is a convenient estimate of the number of plume sources present in the region. As sensor density approaches infinity we expect the number of tracks to equal the number of unique plume sources. The following chapter presents the results

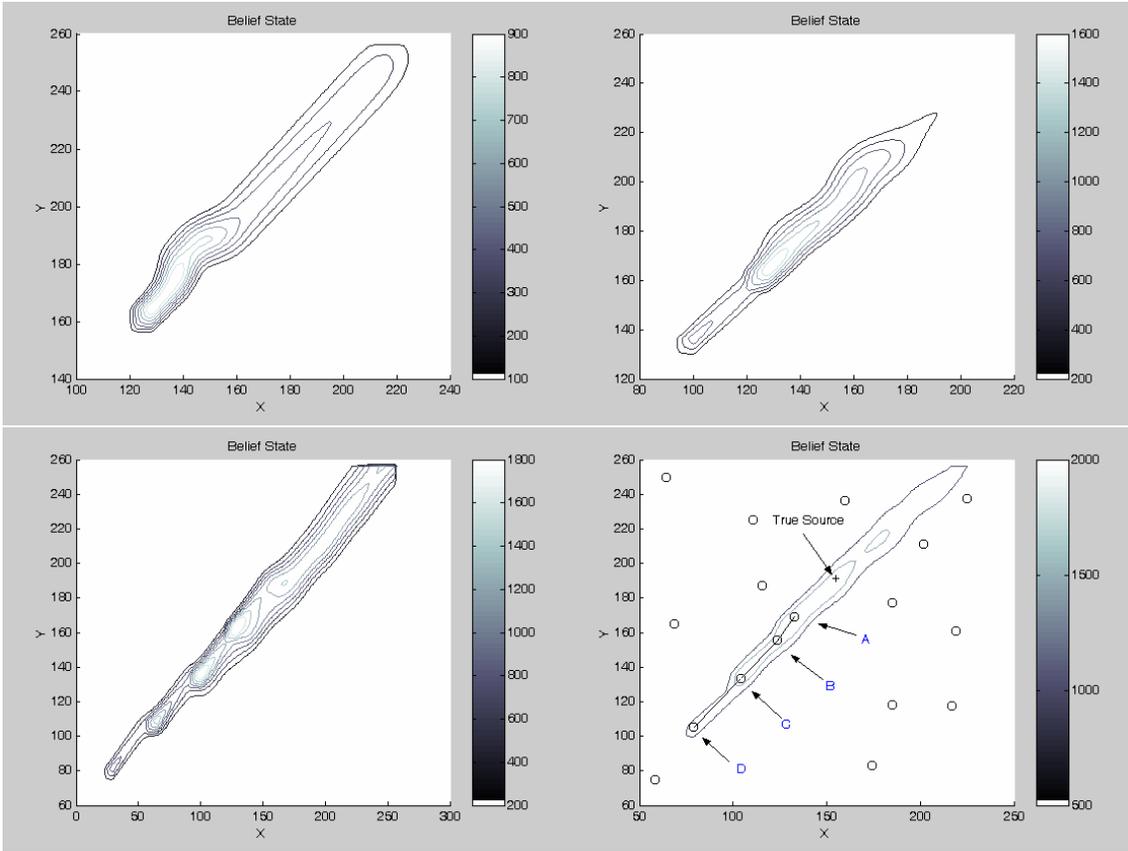


Figure 3.14: Gradual state estimate update with 2-Step algorithm - refined likelihoods as more observations added to the track in the order ABCD. Final update of estimated source position, as sensor data is aggregated along the path ABCD.

from a large number of experimental trials designed to demonstrate quantitatively the advantages of using a tracking based approach for plume source location.

Chapter 4

Plume simulation framework

4.1 Introduction

This chapter describes the simulation system and the experiments used to evaluate the tracking based approach to the plume source location problem developed theoretically in Chapter 3. The algorithms proposed in this thesis are *not* constrained to a particular laboratory setup. Initially a hardware based sensor network was planned with the sensing modality of semiconducting gas detectors for water vapor. However, it was soon evident that the effort required for a field experiment would detract from the quality of algorithm development.

For this reason the experimental platform was shifted to a strictly simulation-based computational experiment. The software experimental platform of the LabVIEW graphical programming environment is an industry standard for data-acquisition and signal processing. The plume tracking algorithms, forward diffusion scenarios, and analysis routines were developed in the LabVIEW graphical programming environment. The simulation runs with the National Instruments run time engine, and is also compiled as a standalone executable. This programming language allows for fast development time, quickly produces a GUI, and highly reusable code. Most signal processing algorithms, numerical analysis routines, and advanced statistical functions are included in the system. LabVIEW has been traditionally used for data acquisition and signal processing, but we feel it is an excellent simulation environment for rapid prototyping. The free run-time engine (similar to JAVA) allows easy application distribution. In addition any function from MATLAB can be called, and any LabVIEW vi can be compiled into a MATLAB mex file or dll. LabVIEW was originally selected for its renowned interfacing with sensor network hardware, and the obvious next step of

experimental work would involve adding physical data to the existing system. By developing the simulations in this programming language, future hardware field studies could be easily integrated with the existing algorithms. This chapter is divided into the following sections:

1. *Software Implementation*

A description of the simulation system design explains the hierarchy of the simulation system. The algorithms for representing the simulation system are described and divided into these modules:

- (a) *Diffusion and wind.* These algorithms perform the forward plume data generation. Wind data is based on files downloaded from historical records for real locations.
- (b) *Sensing.* The binary sensor network detection of plume material and generation of observations.
- (c) *Observation Correlator.* Performs data association between observations received in the sensor network.
- (d) *Tracking.* Given the output from the observation correlator, observations are partitioned into tracks believed to be produced by the same plume event.
- (e) *State Estimation.* Once tracks are formed, the inverse advection-diffusion process is applied from each track (collection of sensor locations) to generate a state estimate of plume sources.
- (f) *Scenario Generator.* This module controls batches of experiments, allowing a user to design a large series of experiments with custom input parameters such as numbers of sensors, sensor placements, tracking parameters, and wind series.

2. *Single release time*

The first collection of experiments had the goal of testing the performance of the system as a function of wind variation and sensor density. To perform this evaluation, a static network of sensors forms tracks, where each track corresponds to an estimate of a unique source. In this series of experiments track initiation occurs only in an initial time window of the simulation. This assumption greatly reduces track maintenance complexity. In addition, all plume sources are released at the beginning of the simulation. Tracks are assumed to be a line of single observation points, where only the sensor node with the data association value is chosen at each step.

3. *Continuous release times*

This enhanced series of experiments required considerable upgrade to the algorithms developed for the *Single Release* phase of experiments. The assumption of simultaneous plume release times was removed, demanding the introduction of continuous track initiation. With the need for continuous track initiation additional methods for track pruning, ranking, and data association were required. An arbitrary number of plume sources could be released at arbitrary times. The definition of track is expanded to include branching, where tracks can have a tree structure. These experiments were designed to show the successful identification of plume sources released throughout the simulation.

4. *Performance metrics*

A number of metrics were developed to quantify the performance of tracking for the single time release and continuous release experiments.

4.2 Software implementation

4.2.1 Diffusion and wind

Overview

The diffusion module performs the forward advection-diffusion operation in the simulation. In our forward simulation the algorithm uses the boundary conditions: concentration for all t is set to zero at the boundaries, the initial concentration values in the space are set by the user. Concentration values can be added throughout the simulation to represent sustained releases of different types. Three types of contaminant sources are possible in the simulation: *limited source* in which there is a one time release, *constant source* in which additional releases occur from the source at every iteration, and *arbitrary releases* in which an arbitrary source capable of being on or off at any time during the simulation.

The forward simulation implements an approximation to the center-difference solution of the advection diffusion equation, with a typical grid size of $m = 250$ and $n = 250$. The result is a two dimensional spreading of material in an approximation of the normal distribution centered at a fixed point, with a linear term added for wind. (We assume a uniform wind field in the region A).

Adding Plume Agent to A

A release is centered at a cell A_{ij} and represents a plume source. Depending on the magnitude

of the release p , a rectangle of cells within A receive concentration values of p at certain times when the source is releasing. The release centered at the cell A_{ij} has a width $2w$, where w is the distance between A_{ij} and the edge of the square release region. Its magnitude of p , is added to the the range of cells defined by:

$$A_{i-w,j-w} : A_{i+w,j+w}$$

and will be set to a value of p at the initial time-step. For later times the value p is added to each of the cells. All releases are assumed occur in a square of size $2w \times 2w$. At each future time step in the simulation the value p may be added to A_{ij} , depending on the release type. For a *continuous release* event a value of p is added at each time-step i where the total number of time-steps is determined by the length of the wind array Θ , where $\Theta = \theta_1, \theta_2, \dots, \theta_N$ is the wind-vector. For an *arbitrary release*, defined by the plume source matrix M , material can be added according to a custom function, where the release point is either on or off for every time i in Θ .

Diffusion and Wind

During each iteration i of the simulation, the diffusion operation is performed on the entire matrix A based on the diffusion constant D . The numerical approach using a standard centered-difference approximation was selected over a random walk particle method for reduced computation. An analytical solution was avoided due to the complexity introduced by potential wind shifts at every time step. Following the diffusion approximation operation smoothing function, all cells A_{ij} are shifted at time i according the current wind value by indexing Θ_i . The linear wind force is applied by rounding the wind direction in degrees (0 – 360) into 8 possible directions and shifting all concentrations within A in this direction. By quantizing wind into 8 directions a simple shift is applied to the matrix A .

4.2.2 Modeling of plume observations

For each time step the sensing module compares sensor locations (i, j) to the concentration values in A_{ij} . If above threshold a “hit” is recorded and the observation is passed to the tracking module. The sensor has a range R in which it will search for a cell above threshold. This value is normally set to a range of 1 cells, defining a sensor as a square region within A of size 3×3 cells. If any cell within the 9 cells is above threshold, the sensor is activated.

Algorithm 1 Forward diffusion algorithm

Inputs: Plume area matrix A of size $m \times n$, Diffusion constant D , Wind-vector Θ , Plume Source matrix M

Outputs: State of plume concentration matrix $S(A)$ after

1. Initialize simulation
 - (a) Set simulation duration = length(Θ)
 - (b) Initialize A to all zeros
 - (c) for $i=0$ to length(M)
 - i. set cells defined by M_i to concentration
 - end for
 2. Run simulation
 - (a) for $i=0$ to length(Θ)
 - (b) perform advection
 - (c) perform diffusion
 - (d) add new agent p to A_{ij} based on M_i
 - end for
-

Algorithm 2 Sensor Detection

Input: Current state matrix A , sensor positions S_i , detection range R , sensor threshold T

Output: Sensor nodes with detections S_i^*

1. for $i=0$ length(S_i)
 - (a) if $T \leq S_i$ set $S_i^* = 1$
 - i. else set $S_i^* = 0$
 - end if
 - (b) return S_i^* for all values = 1
 - end for
-

Algorithm 3 Observation correlation value algorithm

Input: x_1, y_1 sensor 1 position, x_2, y_2 sensor 2 position, wind history vector $\bar{\Theta}(x_1, x_2)$ for sensor 1

Output: $\text{Corr}(S_1, S_2, \bar{\Theta})$ - operates on two sensors positions, and wind history vector at time t

1. Invert $\bar{\Theta}$ such that $\Theta = \{\theta_1, \theta_2, \dots, \theta_N\} \rightarrow \Theta' = \{\theta_N, \theta_{N-1}, \dots, \theta_1\}$
 2. Using Θ' , perform advection-diffusion operation centered at (x_1, x_2)
 3. Given x_1, y_1 located at $A_{x_1 y_1}$ index value for $A(x_1, y_1)$ and return this value
-

4.2.3 Plume observation correlator

The correlator performs a likelihood calculation between an existing track, allowing for data association between observations in the sensor network. A new observation and returns the likelihood that the observation is related to the same event which caused the track. The purpose of the correlator is to assist in accurate data association (DA), by providing a reasonable estimate of the likelihood that two observations originated from the same plume source.

4.2.4 Tracking formation and state estimation

This module is the most computationally complex and handles the observation assignment and predictor function calculation for each observation. The following three components are the main functions:

1. Track initiation - New observations that do not associate with existing tracks based on correlator initiate new tracks. Several rules must kill (prune) new tracks to prevent overload.
2. Track pruning - Due to the large number of new tracks produced with the track initiation stage, tracks must be carefully pruned.
3. Inverse belief map generator - Given a track of observations, this code runs the forward diffusion algorithm with inverse wind directions. (insert Bayes rule inverse likelihood and *TAD* illustration). Uses existing diffusion .vi, but inverts the wind history. Observations not belonging to track not included in this estimate. This leads to belief sharpening, and focusing the belief map onto the region of a single target or release point.

Algorithm 4 Track formation steps

- Step 1 of two step: Track formation

1. Track Initialization - Within the first 25 iterations of simulations all new observations create tracks. The terminal node on track is designated leader node.
 2. Data Association - All new sensors with observations calculate a likelihood function based on wind history. Function evaluated at all leader nodes .
 3. Track extension - observations that were associated in step 2 become the new leader nodes.
 4. Track termination - The track is terminated once simulation ends or no new associations within cutoff parameter. Track outputs sent to Step 2 of 2-step algorithm
-

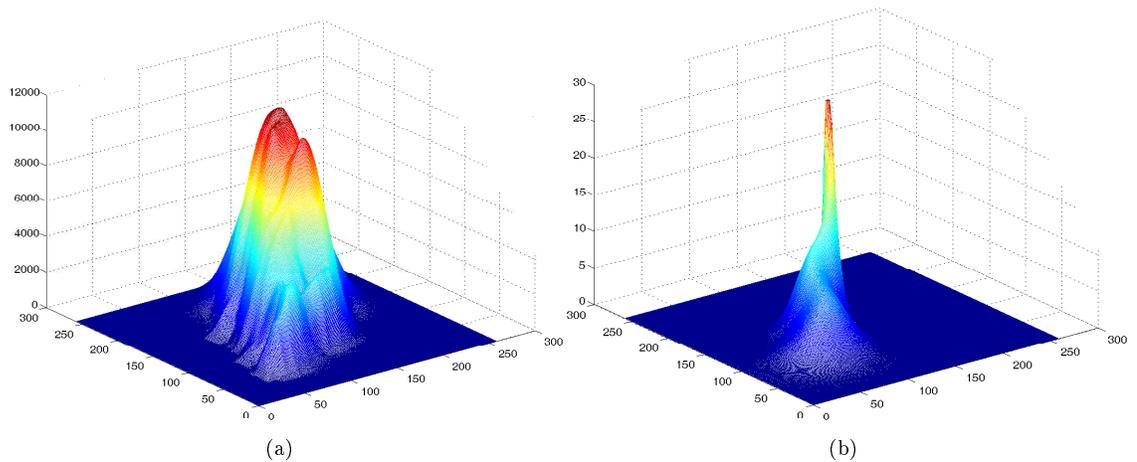


Figure 4.1: 3D plots of forward and inverse regions. Inverse likelihood (state estimation) from a uniform estimator (a) generated by a sensor field measuring the single release event plotted in (b). An ideal inverse likelihood region would be identical, however due to uncertainty the inverse likelihood region is considerably larger than the region containing the plume release.

4.3 Single release time experiments

The first phase of experiments has *Single release times*: we assume all plumes are released at the beginning of the simulation, which greatly simplifies track initiation, data association, pruning, and state estimation. We focus on the ability of a network of sensors to form tracks, determine the number of sources, and estimate their locations. Tracks are initiated only during the first portion of each scenario which greatly emphasizes the rate of track growth, and prevents the need for pruning. These experiments focus on testing the performance of the system under conditions of different sensor density, and a wide range of wind direction standard deviations.

Before running an experiment the simulation scenario generator module is invoked. Here a user can manually place sensors and sources or recalled saved scenarios for additional trials. At this time source size and concentration, wind type, and tracking type are selected. For performance estimation large numbers of simulations can be performed with the same sensor setup except different wind series.

Once a simulation begins plumes observations from the sensor nodes are collected at each sensor independently and reported to the tracking engine. The tracking engine is a simple implementation of MTT, and makes the decision of track initiation, or appending the observation to an existing set of observations (a track). The resulting output will be a graphical representation of likelihood map based on given observations. When tracking is enabled, tracks are graphically differentiated by colors. The result of a typical simulation scenario is displayed in Figure 4.2 with the forward plume generation of 4 sources on the left, and the state estimates by the sensor network on the right.

4.3.1 Studying the impact of wind uniformity

To examine the performance of the tracking based system compared to the non-tracking based source location method a careful analysis of source estimation quality as a function of wind type is required. The original experimental design for wind utilized an approximation of real with with a state based Markov model. Although the output of the model generated wind data that appeared to resemble real wind, this was difficult to verify. Because large volumes of historical wind records are freely available on the Internet, we use real wind data provided by NOAA. By selecting data from a number of weather station locations we can estimate how the system will perform under different environmental conditions. These large continuous wind data sets are available with

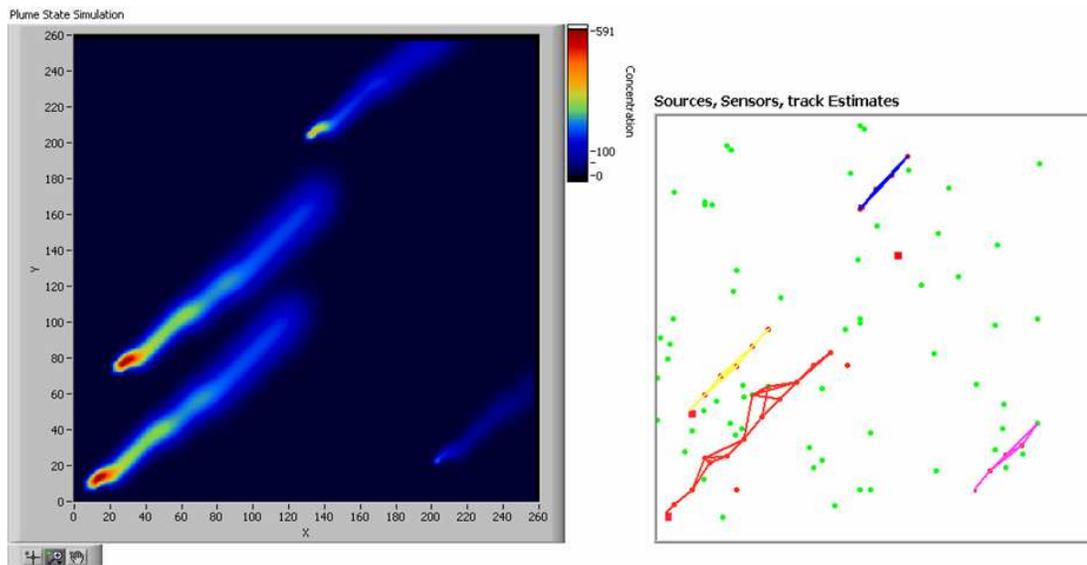


Figure 4.2: Tracking Simulation with 50 sensors, 4 sources, and the correct identification of 4 different tracks. Sensor observations are grouped into tracks based on plume predictor likelihood of observations.

sampling rates of 10 minutes, for the past several years.

Given that typical wind is highly complex, and the performance of the tracking system is expected to be a function of the uniformity of wind, we wish to better characterize the performance of the system under different wind conditions. In the most simple case of uniform wind of constant speed and direction, the solution of the inverse location problem is deterministic and easily solved by non-tracking based methods. With the introduction of a stochastic wind process however, solutions are more difficult. The goal of the wind study is to answer the question:

How does plume tracking accuracy converge on the true source as a function of wind direction uniformity?

Highly non-uniform wind direction (shifting wind) has a high rate of change in direction over a time interval, thus creating more mixing between plume sources and greater difficulty in source separation. A convenient metric for wind direction uniformity is standard deviation of direction. By randomly selecting a segment from a large data set and then calculating the standard deviation of that segment, we can collect wind time series with a wide range of direction uniformity. By producing different wind distributions and measuring the accuracy of the likelihood maps generated for each it is possible to map performance as a function of wind uniformity. The steps involved in generating wind data for simulation:

1. *Download wind data.* In order to study the system performance with real wind data under a wide range of wind standard deviation σ it is first required to select wind data sets containing samples within the desired range. The NOAA sample sizes contain on average 40,000 data points at 10minute intervals for hundreds of different weather station locations. By selecting stations near boundaries such as urban areas, greater ranges of σ are expected. Since each trial only requires on the order of 200 data points, there are potentially 100 unique data-sets per wind file. In reality however, many selections of wind data will have very close values of σ .
2. *Filter wind data.* The wind data requires preprocessing for missing data points, and a custom filter removes these discontinuities.
3. *User Selects a range and resolution of desired σ .* Input to the wind filtering software a desired range and the number of points desired for wind direction. The software will iterate until the desired range and number of points are found matching this requirement. This wind

direction study used data-sets with:

$$5 \leq \sigma \leq 90$$

where σ is in degrees, and sample size $N = 200$.

4. *Choose random position within wind data.* The wind data filtering software will choose a random position within the data set, select a block of 200 points, then calculate σ for that block. If the wind block is useful it will be retained in a file as input to the scenario generator. This iterates until enough wind data has been found (range in σ and number of data points for σ).
5. *Export wind data.* The wind time series with desired properties is saved for later use by the scenario generator.

Once a range of wind data has been found that satisfies the parameters for σ these wind time series data sets are saved to a file and can provide input to the scenario generator for use with any arbitrary configuration of sensor placement, sources, and tracking parameters. In the wind study the sensor placement, source locations, and tracking parameters are held constant as the collection of wind data-sets are run. This allows for a comparison of performance across wind uniformity while all other scenarios factors are held constant.

Wind data preprocessing

The filtering of wind data (*step 3* above) processes the wind data prior to the calculation σ . Considerable preprocessing is required for the wind data due to missed observations. If these these discontinuities in wind direction and wind speed are not removed calculations of standard deviation have little meaning. Due to the large size of most files (on the order or 40,000 samples) manual editing is not feasible.

Calculation of σ

The wind values are used for a standard calculation of for wind direction standard deviation, σ . The wind direction time series Θ takes on the values $\theta_1, \dots, \theta_N$ and the standard deviation is computed as follows:

$$\sigma = \sqrt{\frac{1}{N} \sum_{i=1}^N (\theta_i - \bar{\theta})^2}, \quad (4.1)$$

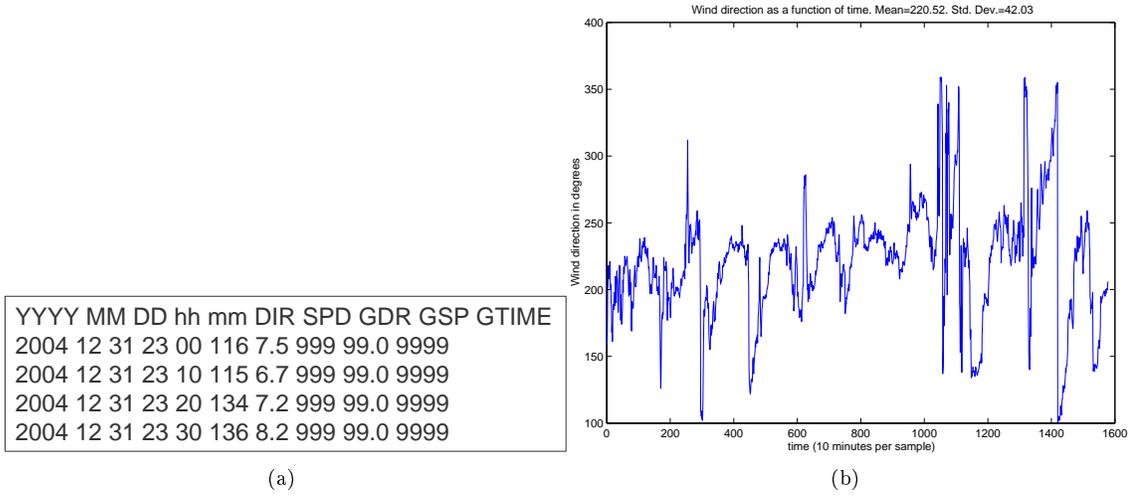


Figure 4.3: Sample raw wind data (a) from NOAA , and (b) wind direction as a function of time after filtering missing data. In this data set $\bar{\theta} = 220.52$ (mean), and $\sigma = 42.03$ (Std Dev). Data points represent 10 minute intervals. Typical trials use 200 data points per trial, which represents 2000 *minutes*.

where $\bar{\theta}$ indicates the mean of wind direction time series Θ . The mean of the wind time series Θ is calculated by the standard equation:

$$\bar{\theta} = \frac{1}{N} \sum_{i=1}^N \theta_i. \quad (4.2)$$

Batch experiments

A scenario was created, with the custom wind data set described in the previous paragraphs, and then run overnight without supervision. In order to directly compare the performance of tracking based source estimation with non-tracking based source estimation a large sequence of trials were performed. The estimation performance of each trial is affected by sensor density, wind variation (σ), sensor placement, number of sources, error rate in detections, source magnitude, and many other variables. The state estimate is then compared to to the actual source location(s). The metrics used for the quality of the produced state estimate will be described in Chapter 5, and will also include a detailed discussion of the results from this wind study. The parameters used for the wind study:

- σ range between 5 and 68 degrees
- wind data sets of 200 records

- 200 sensors

4.3.2 Sensor node density study

In the sensor density study the same wind time series is used, but the number of sensors N is incremented to isolate performance as a function of sensor density. This experiment required the running of 25 scenarios having an identical plume source location and magnitude, but varying sensor density from $N = 25 \rightarrow 300$ in steps of 10, with a grid size of 250. Beginning with $N = 25$, new sensors are added such that all preexisting sensors remain in place. With tracking enabled each trial required approximately 30 minutes to run on a desktop PC with 500 MB memory. This experiment illustrates the tracking based performance as a function of sensor density.

Selected belief maps which generated the likelihood values are illustrated in Figure 5.10. This sample Figure used the following scenario values:

- Increase number of sensors from $N=50, 100, 150, 200, 250, 300$ for a 250×250 grid.
- Random addition of new sensors to existing set.
- Source fixed
- Same wind series for each trial
- Compared performance of belief maps generated by sensor network using tracks Vs. No Tracks.

The discussion and results from the sensor density study will be presented in Chapter 5.

4.4 Continuous release time experiments

The second phase of experiments ¹: *continuous release times*, includes continual plume releases, in which plumes are allowed to release at arbitrary times throughout the simulation. Although the assumption of a single release time allows for more focus on the track formation algorithm, in reality the time of releases will not be known. In addition, a series of releases may occur in a region at different locations. We will model this possibility. Essentially this means adding an additional unknown parameter of time to the inverse plume problem. In the single release experiments we

¹The result of the thesis defense on August 22, 2006 was support for the concept and existing results, however the suggestion that four new tasks need to be considered

seek the location and number of sources, however in the continuous release case we also need an estimate for time.

The introduction of the extra variable of time required much greater code complexity in the areas of track initiation, and required the addition of a track scoring function. Track structure was modified from a simple linear track with leader nodes to a mesh or tree like structure. Track pruning was also introduced. Multiple hypothesis tracking was not included in these experiments, but could be included in future studies. Tracks are optimized at each time iteration, whereas a true MHT method could allow for a global maximum likelihood. The two primary goals of the continuous release time experiments:

1. *Continuous Initiation*

Track initiation expanded to entire simulation time. Focus will be on managing multiple tracks, not MHT. This will require track pruning, and a rule-based approach uses a time out value if no new observations added to track in a defined time interval. This scoring function decreases over time when tracks persist without new observations. Each track will estimate source location as well as time of release.

2. *Extreme case Experiments*

Will run experiments as in *Single Release Experiments*, but with sources released over time and continuous track initiation. Will consider a more advanced rule for observation data association, allowing tree structure for a track. Compare results for different DA rules. Explore the cases in which this method breaks down, regions in which large numbers of plume sources are added.

These two goals greatly increase the complexity of the problem and require the inclusion of the following new experimental (simulation) components:

1. *Plume Release Matrix*

With Releases throughout simulation, multiple sources released over time, a user interface

2. *Track initiation module*

Track initiation for entire simulation time.

3. *Advanced data association (DA)*

Originally just linear tracks, data association needs to form trees which include all new obser-

vations with association to members in the track. The committee expressed that discarding a large number of observations by forming strictly linear tracks loses much useful information.

4. *New Experiments*

Run experiments testing these new features. Test the functionality for large numbers of plume sources.

Originally it was assumed that adding these three features and performing the new experiments would require about a week of time. Continual releases were added without great difficulty, but continual track initiation and the more complex DA were much more difficult than anticipated. Time was spent upgrading the simulation to meet these requirements. Redesigning the code structure had the side benefit of major improvements in efficiency, with simulations running about 500% faster than the original. This section highlights some of the most relevant experiments from the second plume tracking simulation.

4.4.1 Releases throughout simulation

The purpose of this modification is to introduce multiple sources at different times throughout the simulation. In the original simulation, sources were only released at the initial time t_0 . Each new release must be defined by three parameters t_{on} , t_{off} , and amplitude α . The release function for each source is approximated by this square wave function.

Adding releases throughout the simulation was quite easy since the forward diffusion-advection component of code operates on a concentration matrix (*state matrix*) independent of the number of sources. No major modifications were required beyond the enhanced release array. The main simulation loop checks the current time against the release array and adds agent to the specified area at the current time. The scenario generator now contains a T_{on} and T_{off} for each source added into the simulation. Adding the extra variable of time was simple in the forward problem, but added a great deal of complexity in the inverse problem. (Essentially this adds the dimension of time into the inverse problem, taking the inverse problem from 2D to a 3D problem).

4.4.2 Continual track initiation

Adding track initiation to the simulation throughout the run greatly increased the complexity of the code. This feature was essential to handle releases throughout the simulation. In the previous simulation track initiation only occurred during the beginning of the simulation, and therefore

Time step	x	y	amplitude (α)
2	12	18	4
3	12	18	4
4	12	18	4
6	111	112	2
7	111	112	2
8	111	112	2
9	111	112	2
10	111	112	2
11	111	112	2

Table 4.1: Example release array, at each time step the main loop checks this release array for release events matching the current time. In this example there are 2 release events: one occurs at $t = 2 \rightarrow 4$, and the second occurs at $t = 6 \rightarrow 11$ at a different location and amplitude.

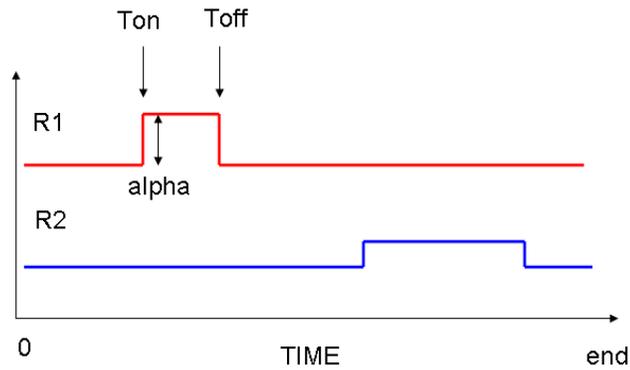


Figure 4.4: Release “truth” for three independent plume sources with amplitude α , initial time T_{on} , and final time T_{off} .

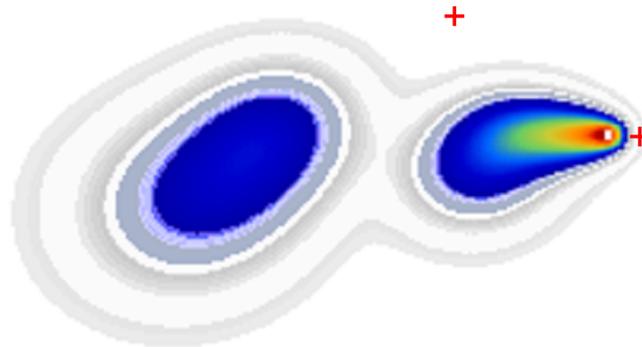


Figure 4.5: Two plumes released at different times at the red plus marks. The first source releases from $t = 0 \rightarrow 100$, and the second source releases from $t = 100 \rightarrow 200$ for a simulation lasting 200 time steps. The first releases (a “limited release”) appears as a moving cloud across the sensing space.

would not form new tracks later with releases introduced throughout the run-time. The challenge becomes deciding when to initiate a new track for unassigned observations, and when to ignore it.

The naive approach to track initiation is simply to create a new track each time an observation is received and fails to associate with an existing track. This simple solution presents two major problems. First, due to the continual sampling nature of these sensors, they report hits for typically 10-100 time-steps as a plume passes their area. Without careful filtering, this flood of observations may only refer to one correct association, and the remaining hits will produce thousands of new bogus tracks. In addition, it is possible for the same observation to associate with the same track many times resulting in a track littered with duplicate observations. The second problem is deciding when and how to terminate a track. If all tracks are kept alive with naive track initiation, the simulation quickly produces an unreasonable number of tracks to maintain. Both of these fundamental problems can be solved with aggressive track pruning.

To solve the duplicate observation problem, all new observations are compared to existing observations within the tracks, and immediately dropped if it has associated with a track at a previous time. This prevents duplicate DA between a sensor and a track. To prevent the creation of new bogus tracks, the same rule is applied - if a sensor is currently associated to an existing track, no new tracks can initiate from that sensor. After passing these two basic filters, an observation can then create a new track. Once a track has initiated it immediately receives a score of 1 ($\Gamma = 1$) which decays over time. A track pruning routine was developed to maintain these track scores over time and kill tracks once the score reach a threshold. (See next section for detail).

4.4.3 Data association - branching

One main complaint during the defense was the discarding of observations assigned to a track or source. In the previous simulation, the terminus of a track was always defined by a single *leader node*. Previously, data association for tracks was only performed between the current *leader node* of each track and the current set of new observations. This led to a strictly linear track shape. The current version adds the possibility to associate with all members of a track, and gets rid of the leader node designation. By allowing DA with all nodes in a track, observations that arrive between nodes later in time can still associate with a track.

A modular design was adopted that allowed experimentation with different DA algorithms. In the preliminary development a nearest nearest neighbor approach was used, before adding in the more complex wind-based likelihood data association.

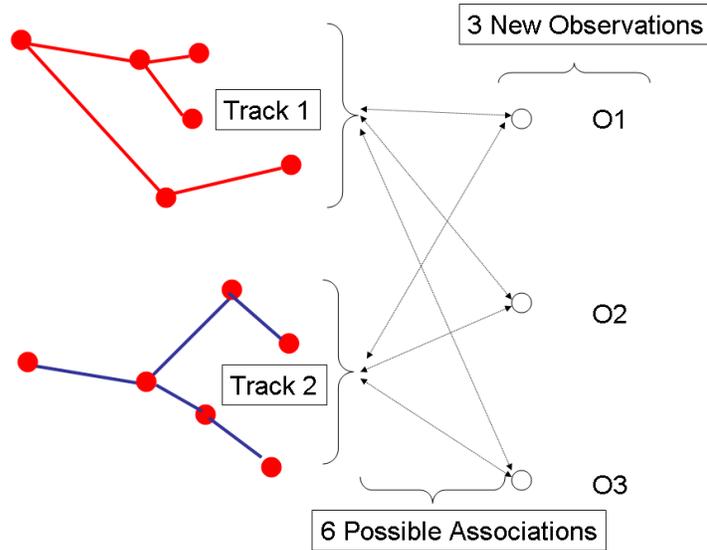


Figure 4.6: Data Association (DA) scheme. Track association possibilities for 3 new observations and 2 existing tracks. DA is based on likelihood function between each new observation and every track member for existing tracks. If DA succeeds, the score of the track is reset to 1, and continues to degrade over time. An observation may associated with multiple tracks. If no DA is performed on any track, a new track will be initiated with the observation.

Another choice in data association is assigning to single track or multiple tracks. The simplest approach is to assign to all tracks above threshold value for data association. Since we are not using multiple hypothesis management, want to assign the observation to all tracks meeting the DA threshold. This allows for a sensor to correctly participate in plumes from different sources that overlap or cross the same region. If observations were only allowed to be assigned to a single track, this would falsely terminate a secondary track crossing the region.

Data association is performed at each time step for all current observations and all existing tracks. If an association is formed with a track, the observation is concatenated to the track, its size increases by 1, and the score of the track is reset to $\Gamma = 1$. New observations can associate with all existing tracks if the likelihood function for any point within a track is above threshold. When new observations do not associate with any exiting tracks, a new track is initiated of length 1, and score $\Gamma = 1$.

Typically a sensor observation is active for a range of time iterations, therefore we must prevent observations produced at $t = t_i + 1$ from associating with observations from the same sensor at $t = t_i$. Sensors are not allowed to associate with themselves.

4.4.4 Track pruning

Producing a workable track termination scheme required considerable effort. Pruning must occur within the first few time-steps of the simulation to prevent the rapid buildup of large numbers of meaningless tracks.

A method to score tracks was developed for track termination (pruning). The track score $\Gamma(\tau, N)$ is function of the number of nodes N belonging to a track, and time since last data association, τ . This eliminates the large number of single tracks formed from a single observation which plague the use of continual initiation. At the time of track initiation ($\tau = 0, N = 1$), and each new track receives a score $\Gamma(\tau) = 1$. The value of a track then declines linearly over time at the rate r . The score at time τ for a track with N sensors is:

$$\Gamma(\tau) = (1 - r\tau).$$

This score will decrease linearly at the rate r for every time-step until a new DA with the track succeeds. Once $\Gamma(\tau)$ reaches a score below the score threshold Γ_{thresh} the track will be terminated.

A threshold score for killing a track, Γ_{thresh} was selected as $\Gamma = .10$, with a decay rate $r = 0.5$, meaning a track will persist for 18 time-steps without DA until Γ_{th} is reached and the track is killed. The performance of the tracking system is very sensitive to the values selected for r and Γ_{thresh} . Setting the decay rate determines the time a new track persists with no new observations. If r is too low this results in large numbers of small tracks, whereas a large r kills most tracks that can provide useful information about the source. An additional rule was added to preserve tracks that reached a size threshold, N_{th} . $N_{th} = 5$ was observed as a good value. Once a track reaches this length, it is maintained throughout the simulation with a score $\Gamma = 1$.

$$\Gamma(N \geq N_{th}) = 1$$

The careful selection of r and Γ_{th} will eliminate the large number of short spurious tracks, while maintaining most larger tracks of interest.

This introduces the problem of which tracks to consider for likelihood estimate at the end of a trial, should we include tracks that were terminated during the middle of the trial period? What if a track is terminated just a few iterations before the end of the simulation? The solution is to only consider tracks for state estimation that contain a minimum number of sensor observations.

4.5 Performance metrics

In this section several metrics will be developed for performance analysis of the MTT method for plumes. In the general multi-target tracking problem for plumes an unknown number of sources (targets) may appear and disappear at random times. They may persist for an unknown random time. Each plume source persists independently for a random length of time and then ceases to exist, based on the release matrix R^T . Under the most general setup, a varying number of indistinguishable sources emit material from a fixed location.

A perfect system would be able to identify the times of appearance, locations, and disambiguate source locations. In contrast to the canonical MTT problem the tracking algorithm must report not only an estimated path of travel and the target's estimated locations, but also the discontinuous two dimensional regions likely to contain the plume material. This increased dimensionality of the source location makes data association of observations more difficult.

These metrics are designed to demonstrate the accuracy of the tracking-based method compared to a non-tracking-based method (uniform estimator). In the uniform estimator method, assuming the number of sources are unknown, all sensor measurements are applied uniformly to the state estimation.

4.5.1 Independent variables

Three factors of interest to real sensor systems include needed density of nodes, performance in adverse wind conditions, and the ability to separate multiple sources. Although many parameters could be selected, these were chosen as the most relevant ones for preliminary study. Once the metrics are developed the performance of these two different methods will be compared against the following conditions:

- *Number of sensors (sensor density)*

As the number of sensors within the region of surveillance A is increased, how does location accuracy respond? We are interested in the minimum sensor density required to achieve a desired level of location precision.

$$density = \frac{N}{m \times n}$$

(Where N is the total number of independent sensor nodes in the network, and $m = n$ are

the dimensions of the region).

- *Wind uniformity*

If we measure wind uniformity as the standard deviation of wind direction for a segment of wind data, how does location accuracy respond as uniformity decreases? Wind standard deviation is defined as σ .

- *Number of sources*

As the number of sources released in the region of surveillance increases, how does the location accuracy for a single source respond? This is a measure of the source-separation ability of the system.

4.5.2 Error estimation

The output of a plume experiment is a state estimate of the likelihood of releases over a region. The state estimate is in the form of a belief map, $M(A)$. The value within each cell, $M(A_{ij})$ is the relative likelihood of a plume source originating in that cell. More precisely this state estimate \hat{x}_t at a time t is based on a partition of the observations ω within A . Each partition ω will generate, in general, a different $M(A)$. In general each different partition ω will generate for a We require a metric to compare this estimate to the true location(s) of plume release point(s). There are three metrics to evaluate the performance of a trial:

1. *Likelihood error*

What was the predicted likelihood of the true source location relative to the maximum likelihood location? Performance Metric Definition, For a Single Source:

$$P_k(M) = \frac{M_k(i, j)}{ML(M)}$$

Where :

- $M_k(i, j)$ is the likelihood value for a true source k ;
- $ML(M)$ is the maximum likelihood value within the map M ;
- and $P_k(M)$ is the performance for map M for source k .
- $\bar{P}(M)$ is the average performance for all k .

1. *Estimated position error*

How far from the true plume location is the maximum likelihood location? The definition of location error is the difference between the maximum likelihood location within $M(A)$, ($\text{argmax } M(A)$), and the true source location, $M(i, j)$, where the distance error E_{dist} is calculated:

$$E_{dist} = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2},$$

where the two points identified by the maximum likelihood and the true source are denoted (x_1, y_1) and (x_2, y_2) .

2. *Estimated number of sources error*

(This estimation method will vary for the tracking and the non-tracking based algorithms. non-tracking based method required the use of image processing techniques to count the number of peaks produced in the source likelihood map). What was the relative difference between the true number of sources, and the number of sources predicted by the number of active tracks at the end of the trial? That is, we estimate:

$$||\omega^*| - |\omega||,$$

where ω^* is the estimated partition. The absolute difference between the actual number of partitions and the estimated number of partitions (tracks) is the estimated source number error

The performance metric ratio between the predicted value of the true location, and the maximum value predicted within M provides a relative estimate of the certainty of the true location. In other words, if we select $ML(M)$ as the best value within M , how close is this value to the value predicted for the true location? Using this system a score of $P(M) = 1$ is the best, and $P(M) = 0$ is the lowest. A value of 0 indicates that the true source location had a likelihood value of zero after the state estimate.

A track in the plume problem is defined by an area in space-time occupied by particles originating from the same localized source. The essence of the MTT problem then is to find tracks from noisy observations, creating a correct partition ω of the observation set Z^T . Each partition ω requires making solutions to the problems:

1. Data association - finding a partition ω of observations Z^T such that each element in the

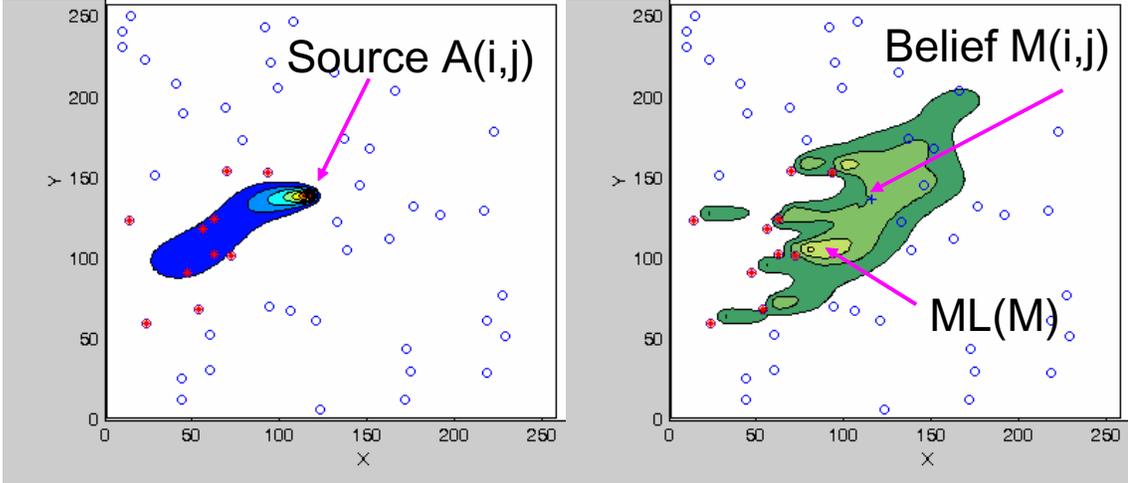


Figure 4.7: Results used for performance comparison - forward plume with its corresponding belief map and the values for ML and the true source.

collection of observations was generated by a single plume source.

2. State Estimation - estimating the state $S(A)$ concentration values of cells within A for prior times, $S(A) = [C(x, y)]^T$

For this reason it is appropriate to have performance metrics that address these two problems separately. In the data association case the measure of performance is the ratio of correct associations to incorrect or missed associations in the tracks. Correct associations will result in the ability to determine the number of unique sources. The state estimation performance is determined by the closeness of the approximation of the state space matrix, $S(A)$ which is the correct concentration of agent in the cells of A . Good performance in the state estimation results in accurate values of concentration.

In the first set of experiments the goal is to characterize the probability of detection for a single source located in the simulation field. In this case one and only one release is added to A as a limited release, which is an impulse of particles about a single point. The wind field is known at all sensors, and uniform across A . In the second set a sequence of different wind fields is applied to a constant sensor configuration and source location accuracy is compared to truth as the randomness of wind is increased. Source location accuracy is measured as a mean squared error (MSE) distance of the best guess for the source position to the actual position for each trial. In the third set of experiments sensor detection noise is added, along with a probability for false detection. This includes errors due to measurement quantization from the binary sensing modality,

and thresholding. Performance for a set number of sensors in a constant distribution is measured as noise is increased.

The primary metric will be the distance from the true plume release point to the estimated release point. In addition to location, a plume source released at cell number (i, j) within the region A (called $S_{i,j}$) contains a vector of attributes $S = (x, y, \sigma, t_0, t_1)$. The metric for estimating this vector of source attributes (location, initial width σ , time of initial release t_0 , and time of final release t_1) will include error estimates for this vector of attributes. Estimation of errors in: time, space (x, y) , release type (function of release), release amount, number of tracks compared to sources will be included. A user of this algorithm would also be interested in the confidence that the correct number of tracks is reported. For example, if the system reports the presence of 3 tracks, it is desirable to know the probability that the true number of tracks is either 2 or 4. More precisely, an important metric of performance is probability of false detection as well as probability of false-negative detection.

The last question from this list can be answered by producing different wind distributions and measuring the accuracy of the likelihood maps generated for each as a metric for quantifying wind uniformity.

As the P increases in a region A the density of potential tracks increases. More important than the number of potential tracks is the density of tracks, or tracks that are caused to overlap by highly variable winds. MTT methods are capable of handling large hypothesis sets, but begin to break down when data association is ambiguous. For this reason, performance is tested as a function of the number of plume sources in a set area, thus increasing the track density as P increases. The number of tracks itself may not matter, but if overlapping the problem becomes hard. In these simulations the region A is quite small (on the order of $L = 250$), therefore increasing P increases track density.

4.5.3 Source counting metric

A new metric is introduced, P_{count} , which is the performance in counting the number of targets. A score of 1 means all targets were correctly identified. This metric is defined

$$P_{count} = \frac{1}{E_{count}},$$

where the counting error E_{count} , is the number of sources is the relative difference between the ideal number of partitions (sources) and the actual number:

$$E_{count} = \frac{|\omega - \omega^*|}{\omega}.$$

Chapter 5

Results and analysis of simulation

5.1 Single release time results

The initial groups of experiments assumed a single release time at the beginning of the simulation, with the challenge of the sensor network to partition sources into the correct number of tracks, and then perform a state estimation on these sets of tracks. The number of sources is unknown *a priori*.

5.1.1 Example release types

In the single release time experiments there are essentially two characteristic release types: *continuous* and *limited*. The continuous release type releases agent at a constant rate throughout the duration of a simulation. Figure 5.1 is an example of the most simple case of a single release source, a single sensor downwind, and a constant wind direction and speed. This is the classic response curve, where the maximum peak is a function of the release rate, and the distance of the sensor from the source. The added complexity in the experiments however was the possibility for changing wind direction. The second characteristic release type used in the single release time experiments is seen in Figure 5.2. In the limited release case sources do not release for the entire simulation, and have a single T_{off} which defines the time window in which the release is active. These two examples show a single release, however in the experiments, these releases were also combined with multiple sources and time-varying wind. The main trait of the experiments, however, is that all sources begin release at the beginning of the simulation.

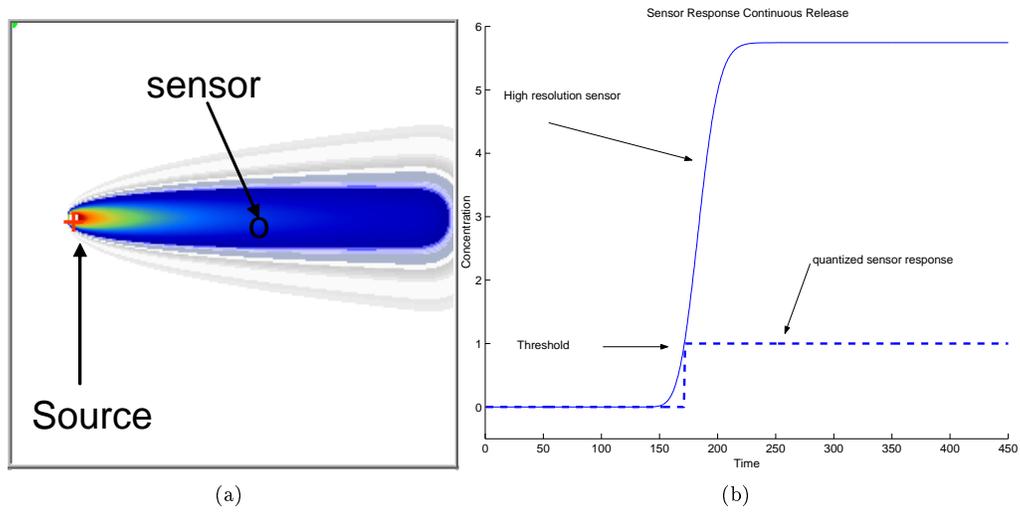


Figure 5.1: Source and sensor setup with response for continuous release type

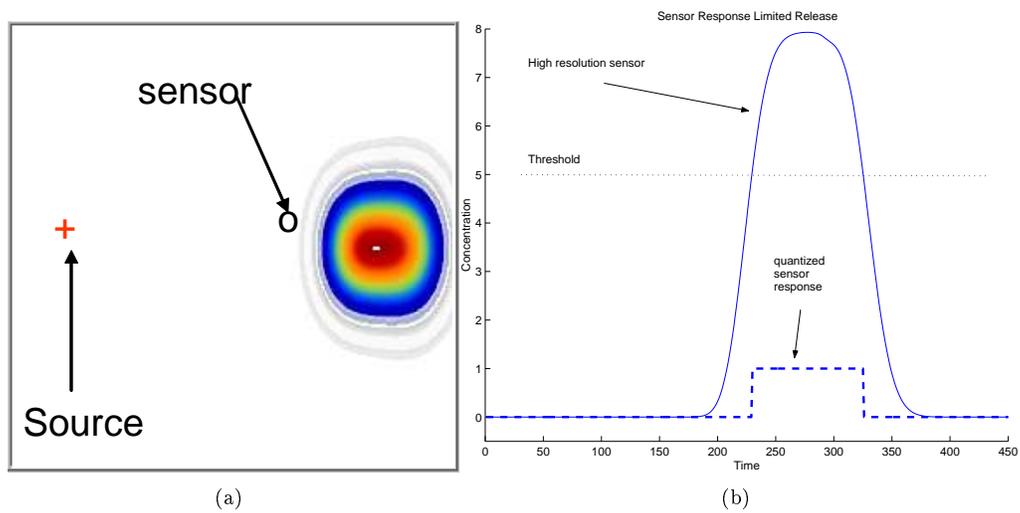


Figure 5.2: Source and sensor with response for limited release type

5.1.2 Likelihood map performance comparison

As explained in chapters 3 and 4, the output from a group of sensors is a likelihood map, where the total area of detectability (TAD) for a network of sensors defines the area of possible surveillance. The intensity of areas in the map represent a relative likelihood (not probability) of the presence of a source at that location. Since the sum of the area in each TAD=1, large TAD regions correspond to very low likelihood values in each cell. For example, a region consuming $50 \times 50 = 2,500$ cells, if uniformly distributed, would have a likelihood value in each cell of $\frac{1}{2,500} = .0004$ or $4E^{-4}$. The primary advantage of the tracking based method is a partitioning of the observations, thus a partitioning of the TAD into regions with increased likelihood per cell.

Likelihood maps for uniform estimator

The likelihood map generated by the uniform estimator method does not assume any information about the number of sources, and therefore the sum of the *TAD* is unity. When no information is available about the number of sources, all sensor observations are evenly considered. As illustrated in Figure 5.3, subplot (a), all sensor observations are included in the estimator and produce a likelihood map. Although the source *is* located within the map in this example, the value predicted for its location is quite small, due to the large size of the map region. In addition three peaks are noticeable within the map, and the true source lies off-center from any peak. The ratio between the likelihood value at the true source, and the maximum value is the performance metric used. Therefore, the maximum performance value is 1, which occurs when the maximum likelihood value occurs at the true source location.

Detecting the number of sources in a region using a uniform estimator is difficult, but can be performed with image processing techniques such as peak detection and edge detection. By performing operators such as the Laplacian or the gradient on the image map, regions of contrast can reveal the number of peaks, which can then be counted to approximate the number of peaks. This method, however, is *highly* unreliable due to the mixing effect of multiple sources. For example the combined likelihood maps of several sensors spaced uniformly and located downwind from two sources will produce a peak likelihood located exactly in the middle of two sources. The peak detection method is therefore highly dependent of sensor, source, and boundary placement. For this reason a peak counting method for determining the number of sources in a uniform estimator method was not implemented as a comparison to the tracking-based method.

Likelihood maps for 2-Step algorithm

In the 2-step algorithm case, when tracking is applied prior to state estimation, the assumption is made that each partition ω of sensors is currently sensing the same plume event. For that reason a smaller subset of the sensors within the network can make the unity likelihood assumption. That is, the total likelihood score for a likelihood region equals 1, and therefore a more narrow region leads to higher and more concentrated values of likelihood. This partitioning prior to state estimation results in sharper state estimate likelihood maps, where one such map is generated for each track. The comparison between these two types of likelihood maps for a collection of sensor observations is illustrated in Figure 5.3.

Source separation

A good illustration of the advantage of the tracking-based method is demonstrated by the ability of a network of sensors to perform source separation. In the continual release case two sources are placed in the middle of the sensor field with a separation distance of zero (no source separation possible) and then moved apart until two unique sources can be identified by the tracking method. The source separation distance is defined as the distance at which two tracks are initiated instead of only one. The separation distance of these two sources is described as a percentage of the region A , where A is a rectangular area of size $m \times m$. For example, if the distance in which two sources can be identified is 10 cells, and $m = 250$, then the ratio $\frac{10}{m} = \frac{10}{250} = .04 = 4\%$ is the separation ability. In the case of a region of size $m \times m$, two sources are placed in the center and moved to opposite edges until they are a distance m apart, which is considered 100% separation. The percentage values under several conditions are presented in Table 5.1.

The above source separation metric is somewhat problematic due to the finite dimensions of A , and the fact that the advection-diffusion model of agent movement does not conserve mass. Once plume agent wanders out of A the agent disappears, treating the edge regions of A as a sort of vacuum. It is acknowledged that this test is highly dependent on relative placements of sensors, sources and wind directions. However, in the region near the center of A the test is more realistic. Of course in a real world scenario, agent is conserved, and once drifting outside an area of surveillance can reenter at another point.

The true advantage of the tracking based source separation can be seen in conditions of high wind variation. In areas of zero wind, there is less information available for data association to tracks in our model, and the two methods perform more closely. However, when wind experiences

	Constant wind	Wind $\sigma = 30$	Wind $\sigma = 43$	Wind $\sigma = 65$
Tracking-based network	10	18	25	32
Non-tracking based network	35	68	80	NA

Table 5.1: Source separation (2 sources) ability for tracking and non-tracking. Distance reported as percentage of region A in which two sources reported.

rapid shifts (as occurs in many urban areas or areas at geological boundaries) the tracking based model excels and operates in regimes in which the non-tracking-based model breaks down. Values of $\sigma > 60$ are examples of such a regime. Chaotic flow regimes of air present in urban areas with tall buildings would be an excellent test for the tracking based model.

5.1.3 Wind variation results

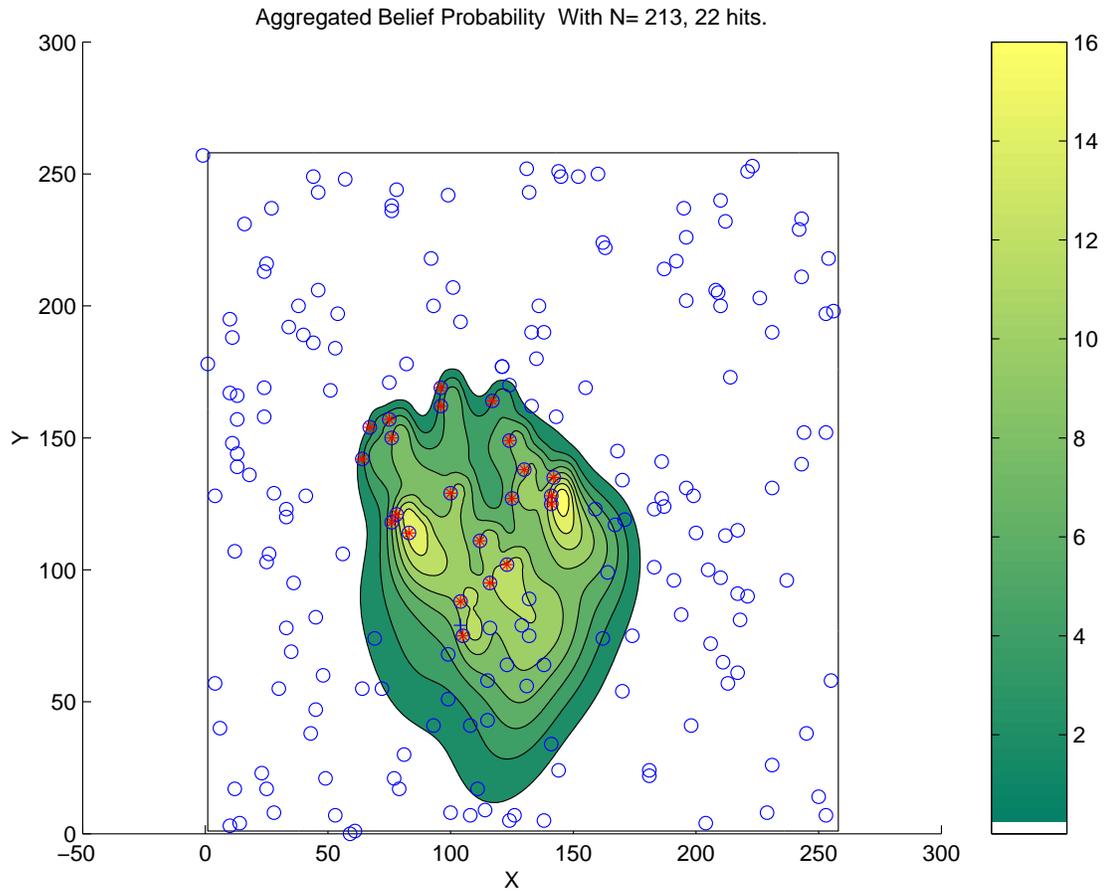
The purpose of the wind study was to analyze how tracking performance converges as a function of wind uniformity.

Wind dataset examples

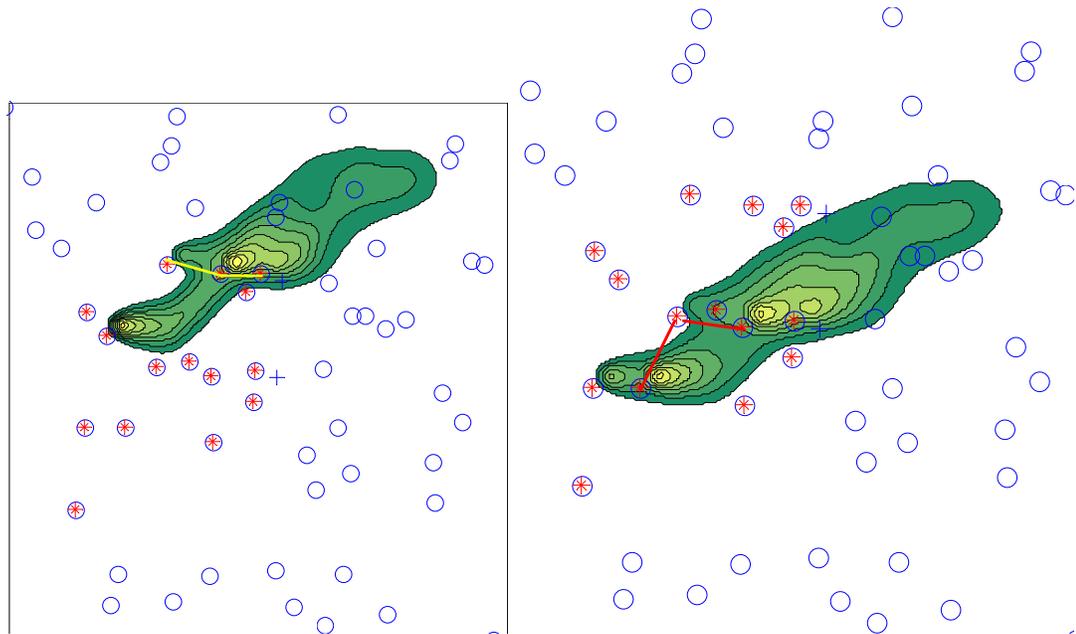
The sample wind data-sets are illustrated in Figures 5.6 and 5.7. On the left in each Figure is the quantized wind direction used in the simulation, which is derived from the original dataset to its right. These example data-sets represent σ values corresponding to $\sigma = 18, 25, 34, 94$ degrees. The large files from NOAA contain on the order of 40,000 records, and the goal of the wind processing software was to locate blocks of length 300 satisfying the needed range of σ which was $5 \rightarrow 95$ degrees. Once data-sets were selected to fill that range, each dataset was manually inspected to detect major discontinuities that could represent errors in the source data. (for example weather stations often report a value of 0 for periods in which the weather station is inactive). Other events such as hurricanes or power outages can inject such discontinuities into data. As described in chapter 3 and 4, the weather data was then quantized to take on the values:

$$\hat{\Theta} = \{0, 1, 2, 3, 4, 5, 6, 7\},$$

where $\hat{\Theta}$ represents an approximation to the original wind dataset. The mapping occurs as $0 \rightarrow 0$ degrees, and $7 \rightarrow 325$ degrees. Wind direction is then rounded to the nearest value between 0 and 7 in this mapping range. By looking at the entire meta-scale statistics for a single weather station, we gain insight on the typical range of wind σ for a given location. Each file contains weather data for an entire year at one station. The time interval of weather data wind direction sampling



(a)



(b)

(c)

Figure 5.3: State estimation results using a uniform estimator (a). In sub-Figures (b) and (c) the observation set is partitioned into tracks before state estimation.

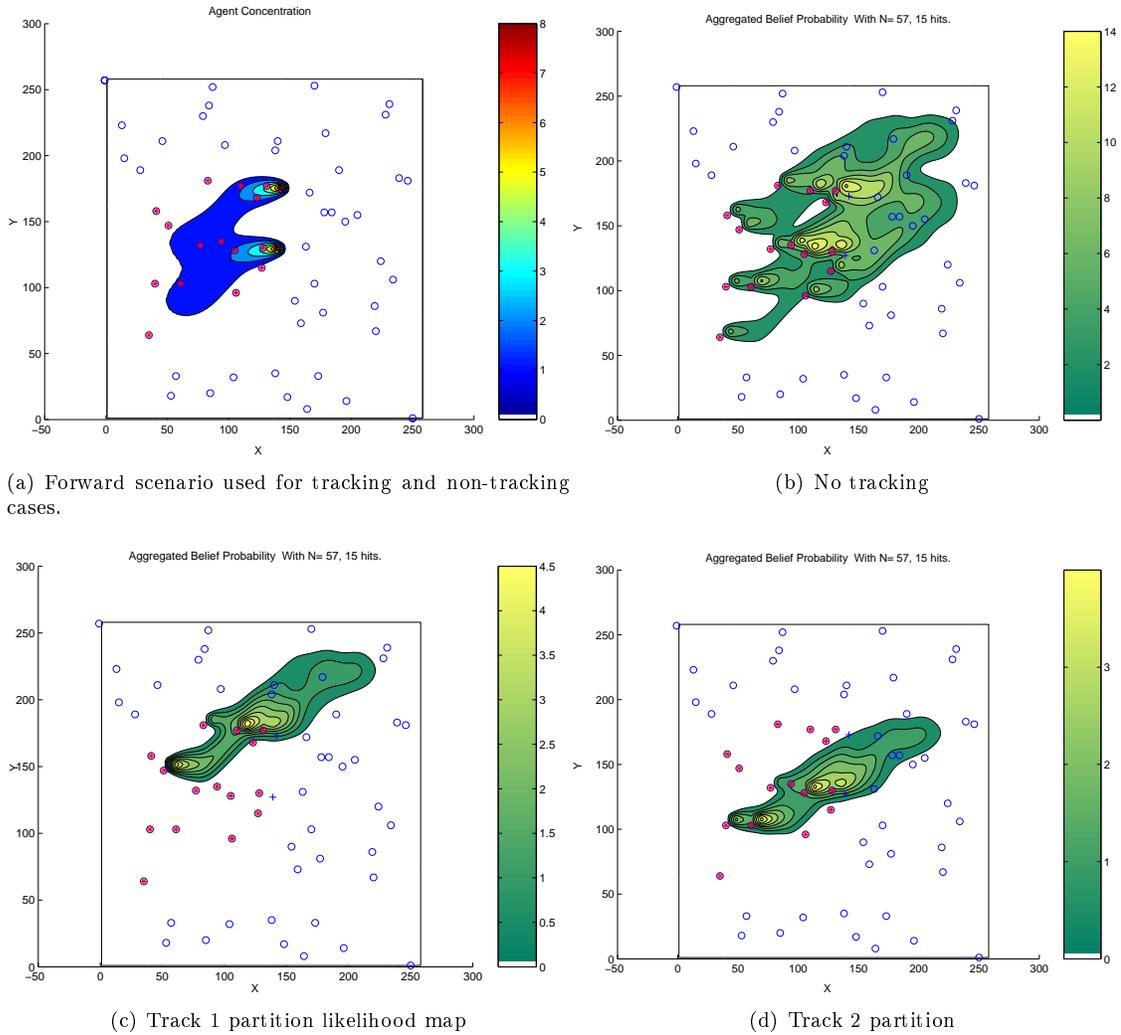


Figure 5.4: Direct comparison between likelihood maps generated using all sensor observations (no tracks) and the same observation set with tracks. The 2 tracks correctly partition the set of observations into 2 smaller subsets producing much higher likelihood values for the true source locations. Belief map for uniform predictor, aggregated belief probability for $N=57$ sensors, 15 hits. Probability of true plume sources are $6.9E^{-5}$ and $1.1E^{-4}$ for the two sources. For the other track (track 2) prob of plume sources = $1E^{-4}$ and $4.5E^{-5}$. Belief map for uniform predictor, aggregated belief probability for $N=57$ sensors, 15 hits. Probability of true plume sources are $6.9E^{-5}$ and $1.1E^{-4}$ for the two sources. Belief maps using tracks, 2 correctly identified tracks. Prob of plume tracks $3.4E^{-12}$ and $1E^{-4}$ for the correct track. $N=57$, 15 hits. For the other track.

is 10minutes, therefore each one-year record represents:

$$365 \frac{days}{year} \times 24 \frac{hrs}{day} \times 6 \frac{samples}{hour} = 52,560 samples,$$

of wind direction data. (with some fraction of those samples being discarded for missed data or extreme discontinuities). When this large block is analyzed we find:

$$\sigma = 68$$

which is the overall value for an entire year. This supports the decision to use values of σ between 5 and 90 degrees.

In addition to sharper and more precise likelihood maps generated by the tracking-based 2-step algorithm, we also see a major advantage in performance with sensor networks in conditions of high wind variation. This increased mixing effect greatly increases the complexity of the source separation problem, and the tracking approach allows for correct partition management even in high mixing scenarios. Figure 5.8 summarizes the wind results in the range:

$$\sigma : 10 \rightarrow 90,$$

where a higher σ corresponds to higher wind variation. We see a very sharp drop-off in performance for the non-tracking-based method for $\sigma > 40$. Since typical average values for σ are at least 60 - this means the non-tracking based methods are not expected to work well under typical environmental conditions. In comparison the tracking-based method operates above a performance value of 0.5 until a σ value of nearly 80 degrees. In conditions of low wind variation ($\sigma < 40$) we see the tracking performance still greatly exceeds the non-tracking based method (uniform estimator).

Wind speed

Until this point the effect of wind speed variation has not been considered . Although not included in the formal study, several zones of performance were observed pseudo-quantitatively during the hundreds of hours of observed simulations. A brief summary of the wind performance summary is presented in Figure 5.9. The Standard deviation of direction is divided into 4 groups, and wind speed is scaled into 4 groups, producing 16 total wind categories. As expected, the system performs optimally when wind direction change σ is low and wind speed is high. In this

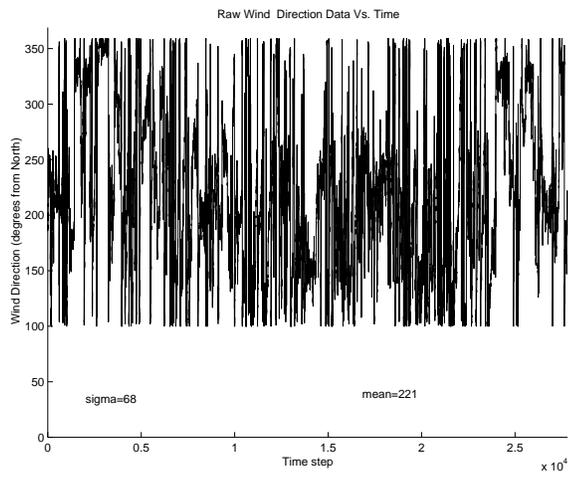


Figure 5.5: Raw wind data block showing meta-scale statistics for block of 30,000 points

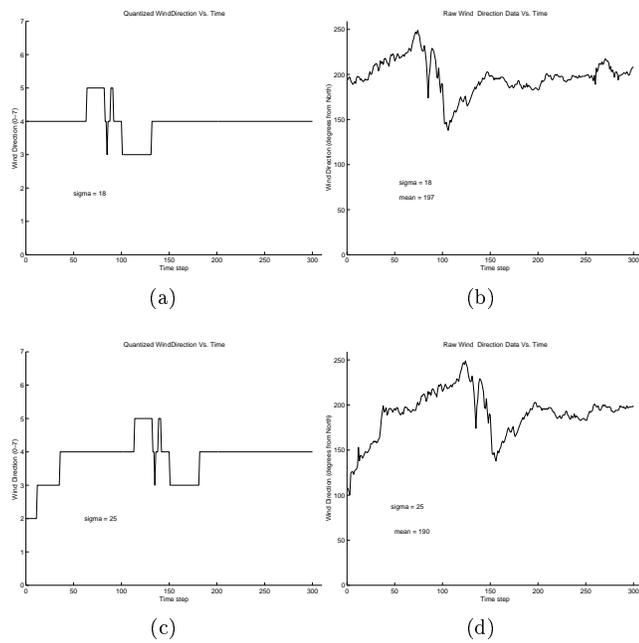


Figure 5.6: Wind for $\sigma = 18$ degrees and 25 degrees

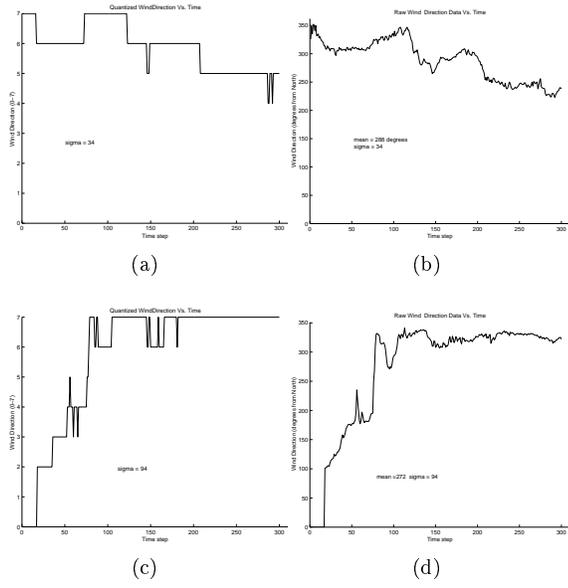


Figure 5.7: Wind for $\sigma = 34$ and 94 degrees

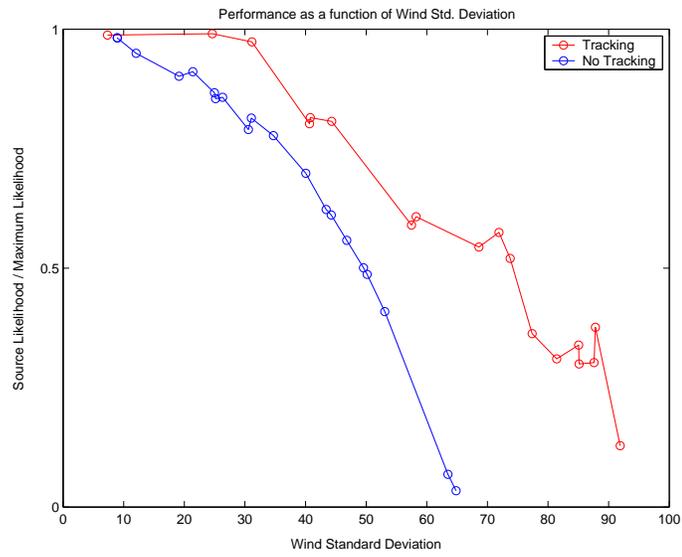


Figure 5.8: Likelihood scoring as a function of wind direction standard deviation. x axis is wind standard deviation in degrees. Y axis is normalized likelihood value of true plume source. Tracking based localization was able to maintain near perfect scores up to $\sigma = 30$ degrees and operated up to a value of $\sigma = 90$ degrees. The same network without tracking received likelihood scores=0 for $\sigma \geq 65$ degrees.

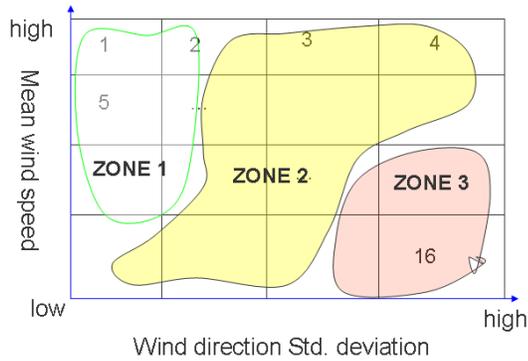


Figure 5.9: Summary of wind performance zones. Zone 1: Best performance zone, zone 2: intermediate, zone 3: Worst performance
 Zone: low wind Speed, with Frequent shifts

zone, the moving plume most closely resembles a moving rigid target.

5.1.4 Sensor density results

The density experiments were designed to illustrate the relative performance of tracking and non-tracking methods across a range of sensor densities. Figures 5.11 and 5.10 illustrate the results from the density experiment described in chapter 4. One result of interest is that an increase in the number of nodes beyond an optimal point in the non-tracking method results in higher uncertainty of source location. This can be explained by the involvement of an ever-increasing number of nodes which are not partitioned into target groups. (In the non-tracking based method no information is known about the number of sources).

Identical networks (node location, number of nodes, and sensing range) with tracking can achieve sharper maps with lower densities of sensors. The major advantage of using tracks is the ability to establish the number of unique sources, thus requiring a lower number of nodes per unit area. The theoretical information content of a sensor network grows as $\log(N)$, where N is the number of randomly placed nodes within the region. Therefore we see diminishing returns as N gets large. Another way to consider the value of adding more sensors is to consider the information content provided per cost. Assuming a constant cost per node as the number of nodes increases, we expect the information per cost to decrease as $\frac{1}{\log(N)}$ by adding additional sensors. Both networks approach this limit but at different rates. For this reason, we see information content per cost benefit for using the tracking based method.

Taking a closer look at Figure 5.10 we see the changing likelihood map as the number of

sensors in the region is increased from $N = 50 \rightarrow 300$. This example Figure shows a few snapshots representative of the many trials used to generate the density results. The first subplot, subplot (a) shows the example forward plume release. This single release experiences changing wind, and activation of a number of downwind nodes. As N increases, the preexisting nodes are left in place as new nodes are overlaid. This results in the nodes from previous trials always being included in the results of later trials with more nodes. This allows us to strictly examine the benefit from adding more random nodes to the region.

The maximum value of 300 nodes was selected and the experiment area grid size of 250 produces a total number of cells:

$$Cells = m \times n = 250^2 = 62,500,$$

and each sensor has a sensing footprint of 5×5 cells, meaning the total number of cells being sensed was:

$$Sensed = N \times 25 = 300 \times 25 = 7,500,$$

therefore on the order of 10% of all cells are monitored in the maximum density case. In the lowest density case, about 1% of all the cells are monitored. This range of 1% \rightarrow 10% cell coverage was selected to represent typical values of a city or region that implement an air monitoring program. In the case of a city with high value assets (such as national monuments) such a system would implement the top 10% of areas with likely targets with sensors.

The source location (indicated by a “+”) is at the head of the plume release in (a). Subplots (b), (c), and (d) show the resulting maps produced with increasing sensor density. The likelihood values in these plots have been normalized. Normalization is performed by dividing the entire matrix by the sum:

$$\sum_{i=0}^m \sum_{j=0}^n A(i, j),$$

where the sparse matrix will have a majority of the cells at zero concentration. Therefore, we are adding up the total area of the likelihood map “patch.” Once normalized the range of values per cell are $E^{-3} \rightarrow E^{-4}$, depending on the size of the patch. In this particular example, we notice minimal improvement with added sensors, as all performance metrics are on the order of E^{-4} . This example illustrates the non-tracking case, as well as numerous peaks that do not correspond with the actual source location. Although the source location is within the patch, its likelihood value is about 5 times less than the peak value, therefore in this example, the performance value ≈ 0.2 .

The combination of all trials is presented in Figure 5.11 which directly compares tracking and non-tracking based performance as a function of sensor density, with performance values ranging between 0.2 and 1.0. Once again, the performance metric used here is the ratio between predicted likelihood of source and the maximum likelihood, therefore a score of 1 means the source was located at a site of maximum predicted likelihood. This could be problematic as a metric if the likelihood distribution is totally uniform, or several regions of identical maximum likelihood are spread around the region. In practice however, the generated likelihood maps have reasonably sharp peaks distributed throughout the likelihood maps.

5.1.5 Source number counting results

An additional result for the single release experiments was the comparison between the tracking and non-tracking methods in determining the number of targets (sources). Figure 5.12 compares the performance in determining the correct number of targets. Performance metric for source counting was developed separately from the other performance metric, and was described in section 4.5.3. A score of 1 means the correct number of target was identified, and a score of zero means none were identified. As expected the tracking-based method excels at larger numbers of targets in the same sized region, due to mixing of agent across the sensor field. For non-tracking based sensor networks, the multi-source separation problem is very difficult. The region used for the results in Figure 5.12 used a region of size 250×250 , with a wind $\sigma = 10$ degrees. In this comparison the method used for source counting for the non-tracking based approach was peak detection from a standard MATLAB peak counting function found in the image processing toolkit. It is possible this method could be improved with enhanced image processing techniques using the non-tracking methods.

5.2 Continuous release time results

5.2.1 Example release types

The goal of the continuous release experiments is adding the variable of time to the problem of source location. By continuous release we mean sources may appear and disappear at any time during the simulation. The introduction of these more realistic releases have arbitrary initial times, arbitrary number of sources, and each source can also exhibit pulsing behavior. A charac-

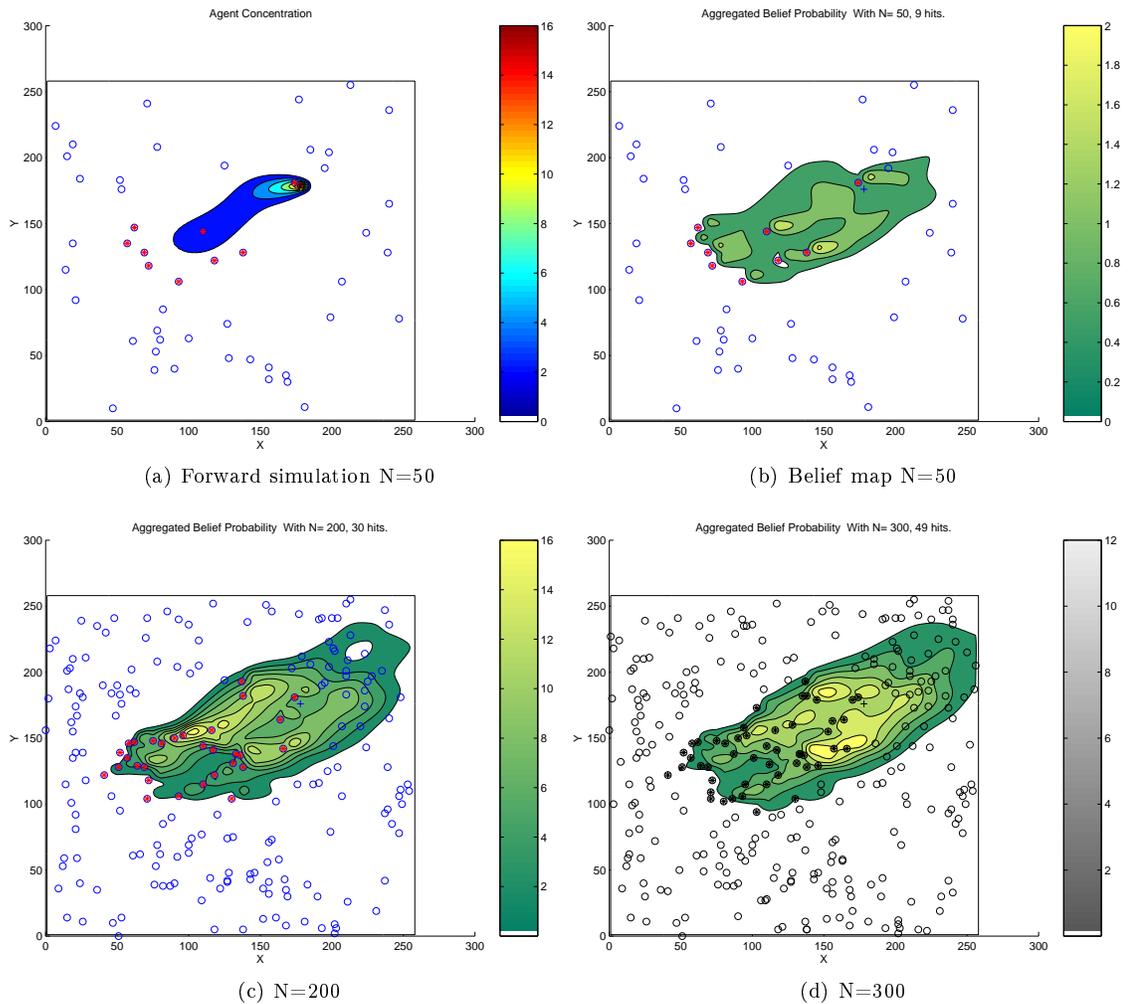


Figure 5.10: Density experiment scenario that used for $N=50, 60, \dots, 300$ nodes in 10 node increments. This Figure shows the forward plume for $N=50$ (a), and selected belief maps generated for $N=50$, $P(M) = 1E^{-4}$ (b), 100, $P(M) = 1E^{-4}$ (c), and 300, $P(M) = 1E^{-4}$ (d). Identical source location and wind series used in each trial. The likelihood scores for the true source location increase until optimal N , and additional sensors provide little added information. (optimal $N \approx 200$)

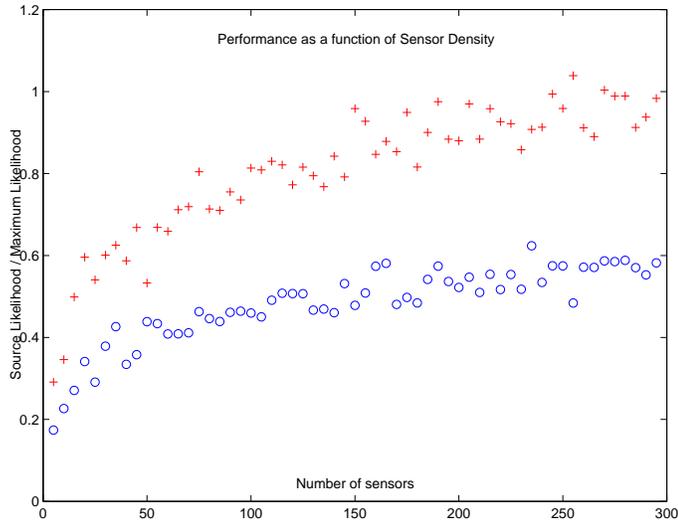


Figure 5.11: Sensor density performance results summary as N increases from 50 to 300 nodes. Tracking (red), and non-tracking (blue) performance.

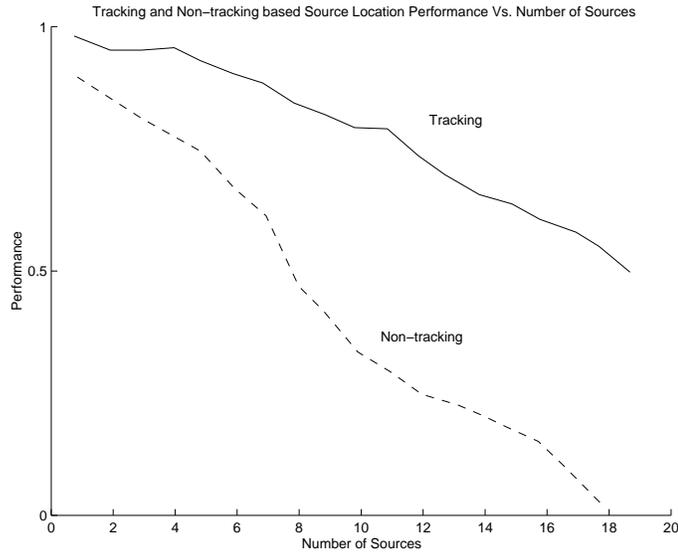


Figure 5.12: Source estimation performance as a function of number of sources for sensor fields experiencing same wind time series.

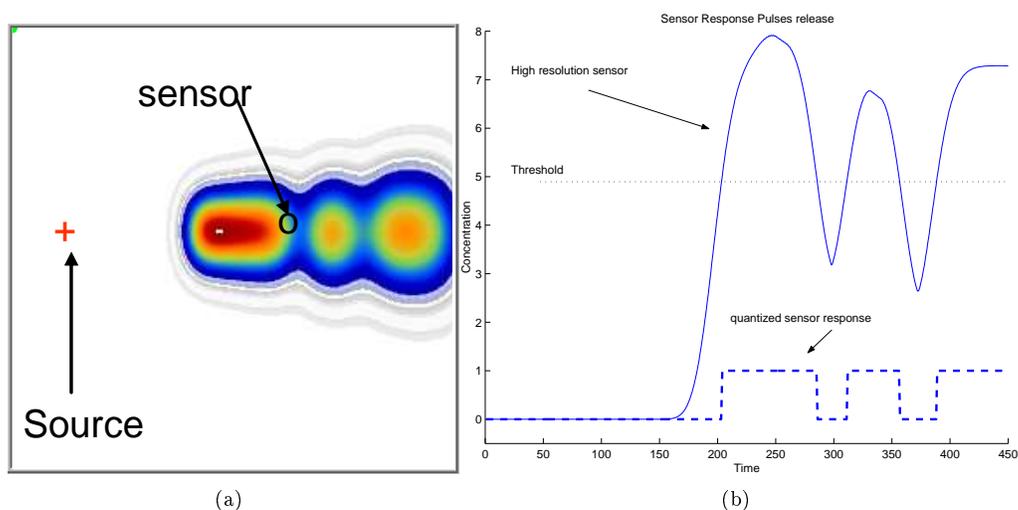


Figure 5.13: Pulses release type used in the continuous release experiments and sensor response for pulses of release.

Characteristic example pulsed release is illustrated in Figure 5.13 along with the typical binary response observations reported in a simulation. In this particular example, the source does not release for the entire duration of the simulation, and has lag periods between release puffs, and a constant wind direction. The second example release in Figure 5.14 illustrates a complex scenario with four sources, a time-varying wind, and pulsed releases. In this complex case, the binary observation sequence at a single sensor provides very little information about the initial states producing the observation. This complex release case is a good example of the advantage of the tracking-based approach and represents the type of scenario used in the experiments to differentiate tracking and non-tracking based state estimates in a sensor network.

5.2.2 Continuous release tracking

The goal of this section of experiments is verification of the ability of the 2-step algorithm to correctly track sources of the *pulsed* or *complex* variety introduced in section 5.2.1. When these complex release types are introduced, and the assumption removed that all sources are released at the initial time, a number of problems are introduced:

1. Track maintenance becomes non-trivial
2. Track pruning must be introduced
3. The dimension of time is introduced into location estimation

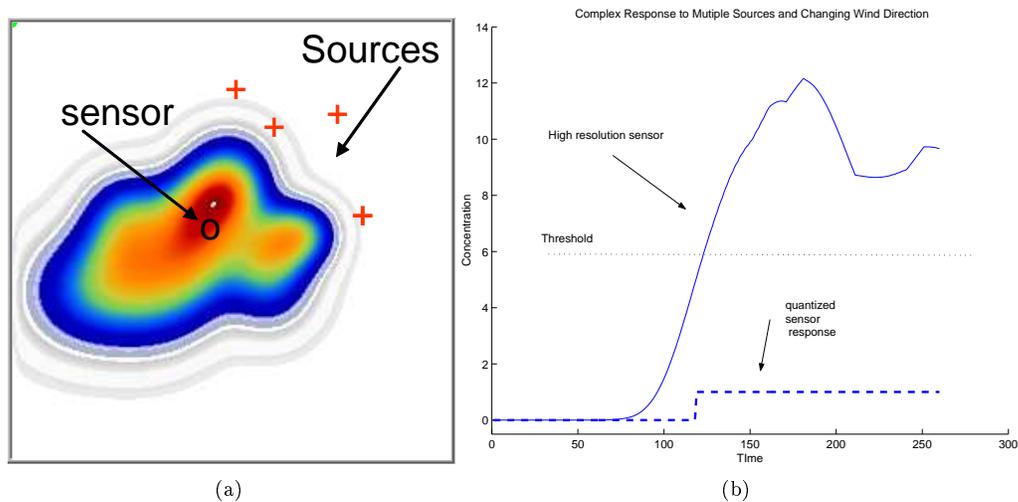


Figure 5.14: Complex release type with multiple sources and changing wind. Sources begin and end at different times.

4. More complex tree-like track structure

Once these three added complexities were considered and introduced into the tracking simulation system, a number of basic trials were used to test the added features. The fundamental features we seek are the ability to initiate and maintain tracks throughout the simulation, as well as the ability to correctly identify the number of sources.

The upgraded continuous release system maintains tracks that persist above a threshold score, and the score degrades over time. The system performance is highly sensitive to this value, which depends on wind conditions, sensor density, and other factors. If the threshold is set too low, data associations are missed and tracks terminate prematurely. When the threshold is set too high above optimum, the number of associations explodes leading to system lock-up. Thus, performance is optimized when the data association threshold is set to an optimum value. This value is studied for one case empirically, and not derived theoretically. Additional study of optimum values would be a ripe area for future study.

In addition the geometric limitation that tracks take on a strictly linear form was removed in this section. Tracks have a mesh or tree-like structure. Sensor observations are allowed to data associate with all nodes within a currently existing track, as opposed to only the terminating nodes or leaders, as in the previous chapter. An example simulation which demonstrates these more complex standards is seen in Figure 5.15. In this example 5 tracks are formed from the release of 2 sources. As the simulation persists, tracks can merge, die, or initiate at any time. For

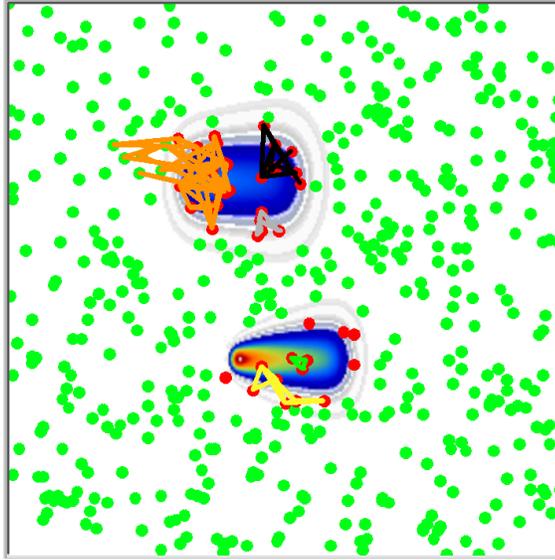


Figure 5.15: Multiple tracks initiated throughout the simulation for continuous releases.

this particular snapshot in time we see 5 tracks that previously were 8 tracks. The orange track is a combination a merger of 4 different tracks that had data association merging occur. Green nodes within the tracks correspond to sensors that were activated at a previous time, but are no longer sensing agent above threshold.

Figure 5.16 illustrates the superposition of agent concentration (in contours) with two tracks. These two tracks correctly correspond to the release of two different sources that occurred throughout the simulation. The first release was limited in nature, and originated at location $(200, 200)$ as indicated by the red $(+)$ mark. The limited release contour is seen as a puff drifting downwind from the release origin. Later in the simulation a second continuous release was initiated (at $t = 200$ out of a total $T = 400$) with an origin location of $(190, 100)$ just below the first track. The observations for this second event are correctly data associated to the second track, not to the initial event from $(200, 200)$. Several observations (red filled circles) to the left of the larger track were not associated with the track, and represent data association errors.

5.2.3 Tracking optimization

The introduction of continuous track initiation, as opposed to only allowing track initiation during the initial portions of a simulation adds the need for track scoring, and the track scoring function Γ , as introduced in Chapter 4. This function is sensitive to the threshold for track

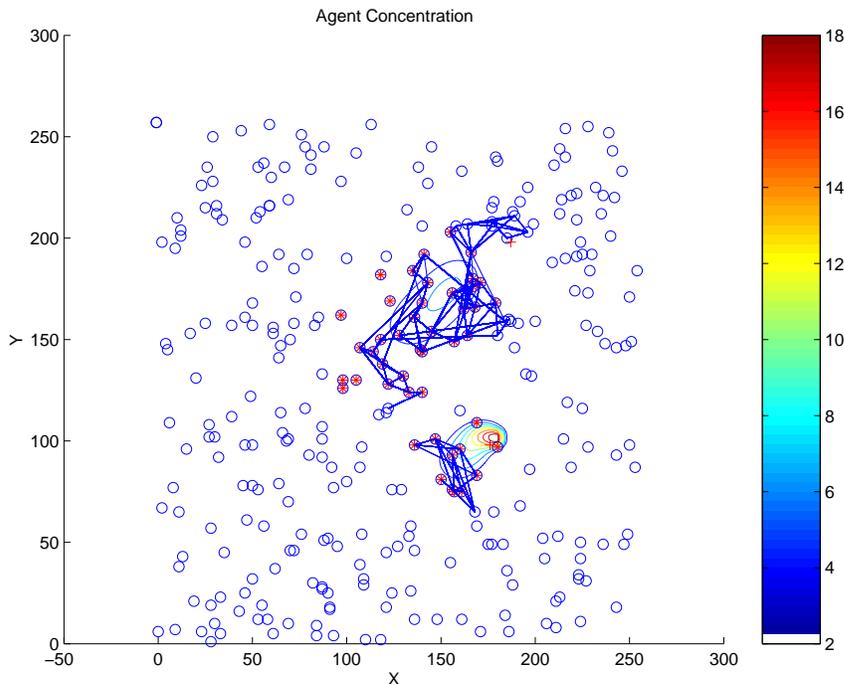


Figure 5.16: Time sequence results for two sources of complex nature, and resulting tracks.

termination, Γ_{thresh} , and a lowered threshold can easily result in a combinatorial growth in the number of tracks, and therefore careful empirical evaluation is required to identify proper tracking parameter constants.

Figure 5.17 demonstrates a track score function using the parameter $\Gamma_{thresh} = 0.2$. This was determined to be a proper value for the majority of experimental trials. In this example of the Γ function the score decreases over time at the rate r , with a value $r = 0.05$. This means at each time during the simulation in which no new data association occurs the score is decreased by this amount r . That is, in the no DA case,

$$\Gamma_{t+1} = \Gamma_t - r.$$

Once $\Gamma \leq \Gamma_{thresh}$ the track is terminated. In the Figure this can be seen as Γ is set to zero at time $t = 65$. This track scoring approach allowed the continuous initiation of tracks, and prevented excessive track number growth with a basic pruning algorithm.

The second major parameter critical to effective track pruning and proper data association is the data association threshold. This value determines the correlation score (Correlation score developed in chapter 4) required between two observations for an association to be formed. Similar

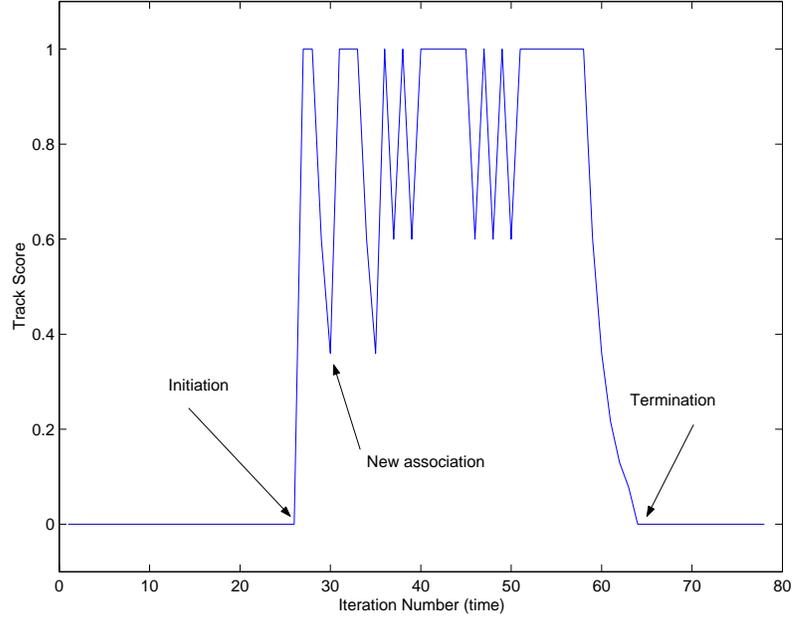


Figure 5.17: Track score as a function of time. $\Gamma = 1$ at track initiation, and reset to $\Gamma = 1$ for each new association.

to the Γ_{thresh} parameter it was determined empirically. Figure 5.18 presents the effects of changing this value over a range of $0 \rightarrow 27$. As this plot illustrates, large differences in the standard performance metric occur when the parameter is not set to an optimal value. Essentially a low value prevents correct associations from forming, and a high value leads to a high false positive data association rate. A large number of false positive DA formations manifests as all tracks merging into one large track, and the loss of the advantage of the 2-step algorithm observation partitioning.

5.3 Scalability

Due to the additional computations overhead of releases throughout the simulation, continual track initiation, and pruning - the entire simulation code had to be optimized and reworked. After code optimization, improvements in data structures - an improvement of 500% on average was observed compared to the original single release system. Running the same simulation initial conditions from before the thesis defense most trials required about one fifth of the time to run. This was accomplished by elimination of superfluous data passing, and more efficient storage of large arrays. In addition, the number of times the likelihood function is calculated was reduced. The one dimensional cluster array containing tracks was changed into a 3D array of numerics, along with many other major improvements in efficiency.

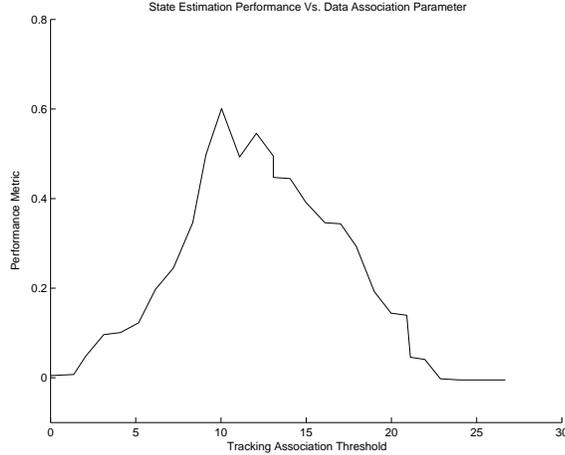


Figure 5.18: As the data association parameter increases there is generally an optimal setting for the heuristic based data association rule. Low values miss associations, whereas high values allow for false positive association.

The most computationally intensive portion of the simulation is the calculation of the likelihood function. Because the algorithm now considers all possible DA permutations between observations and tracks, the computation needs to be as efficient as possible. For example, if there are 5 tracks each containing 10 observations, and 4 new sensor observations, then the likelihood function must be calculated $5 \times 10 \times 4 = 200$ times! This is only for a single time-step. Each of these 200 likelihood scores would then be compared to the DA threshold value, and DA would be performed for each pair above threshold. The problem with this setup is that the average calculation time for a likelihood function is currently $200 - 400ms$, or close to $\frac{1}{2}$ a second. That means a minimum of 100 seconds per time-step for this example.

The approach was to minimize the number of likelihood function calculations. The goal was to stop performing likelihood calculations with a track as soon as DA within the track occurs. For example if DA occurs with the first analyzed point in a track, no more points in that track are analyzed. Since we are no longer maintaining leader nodes, a connection map among nodes in a track is no longer required. We only need to know only which nodes belong to a track. If we wanted to have a complete graph containing all edges and connections, this approximation would not work. A track is therefore more a collection of points, than a connected tree or graph.

Figure 5.19 shows the time required for a range of simulations using the improved simulation design as the number of plume sources increases from 1 to 20. Unfortunately run-times for simulations in the original phase were not collected to allow for comparison. However, it should be

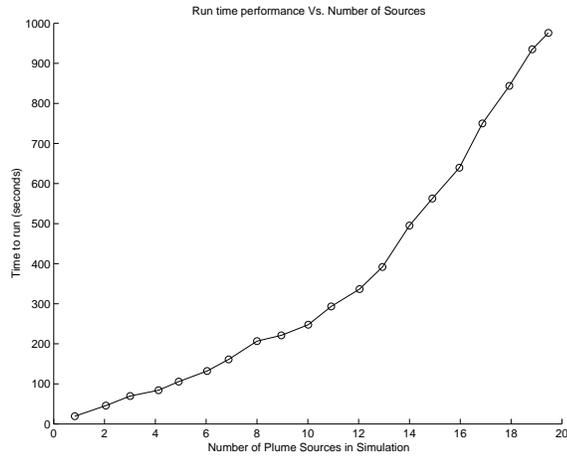


Figure 5.19: Runtime performance for algorithm as number of sources increases

noted, that the original design for similar numbers of plume sources, sensors, and tracking enabled required several hours each in some cases. Although performance was not addressed as a major theme of this thesis work, if the system were to be deployed in a sensor system of simple nodes, greater performances in computational efficiency could most likely be achieved.

Chapter 6

Conclusion and future work

This thesis introduced the new concept of tracking chemical plumes in sensor networks with simple binary detections, and then demonstrated its feasibility through a series of simulations. The two-step algorithm was developed from the fundamentals of Bayesian state estimation and used to derive likelihoods for plume sources based on a partition of observations. By performing the tracking step of the two-step algorithm first and correctly partitioning observations, the number of sources can be much more accurately estimated than using simple peak detection in likelihood maps. Experimental results indicated that tracking based plume source localization performs better under conditions of low sensor density as well as high wind variation. Although many additional studies can be performed characterizing the MTT based approach to plume tracking, these foundation studies verify the utility of the approach.

Due to the novelty of this approach to plume tracking a large amount of work remains unfinished. Future simulations might operate on more realistic forward simulations including barriers, sophisticated wind models, and three spatial dimensions. Future theoretical work needed includes optimization of sensor placement, more analytical approach to finding ideal sensor densities, the inclusion of negative observations, and the removal of the assumption that all observations are available at a central location. In addition, the sensor network issues of efficient routing, bandwidth limitations, and power need to be considered before field experiments are implemented. One such important application is the monitoring of mobile sources releasing particles, and the protection of corridors such as interstate highways. Once the more sophisticated theoretical and simulation problems have been addressed the verification of these models in physical field testing would be needed for ultimate validation. Many systems based on the fundamental process of diffu-

sion are good candidates for future study to illustrate the extensibility of tracking-based methods. The domains of disease propagation, information diffusion (tracking the content of news stories), and social processes involving diffusion are excellent candidates.

6.1 Needed theoretical work

The two-step method introduced in Chapter 3 makes the assumption that information is universally available throughout the sensor network. Real sensor networks typically employ ad hoc routing which requires a more distributed processing approach to target tracking. When communication losses are included along with limited bandwidth, the assumption of a central processing location is not realistic. A more distributed approach would adapt the two-step algorithm to existing methods developed for sensor networks in the distributed signal processing community.

The current system does not make use of negative observations due to the bandwidth demands in a sensor network for continuous reports. A future modified version, however, could query sensor nodes in the vicinity of current activity to retrieve relevant negative detections. For example, negative detections upwind from the sensor initiating a track could greatly increase the state estimate of the plume source when not enough positive detections are available. By addressing only nodes in the regions of current activity and requesting negative detection reports, a compromise could be reached between bandwidth conservation and performance.

Sensor placement was not examined in these studies, as all density studies added sensors in random locations. Much can be gained by optimal sensor placement, since the number of nodes available are a critical limitation in real world deployments. Once boundaries simulating buildings are added to the simulation, experiments could then be run to study the optimal distributions for sensors in a realistic city model. This would greatly enhance the basic sensor density studies performed in this thesis which assumed an empty two-dimensional area without barriers.

The simulation performance could be greatly improved by porting the code to a more efficient programming language such as C. Although LabVIEW is an excellent prototype development system, there is a compromise in efficiency. In addition many of the numerical approximation routines could be optimized with more careful design. Integrating the plume tracking simulation to the PQS framework developed at Dartmouth College offers the advantage of an optimized tracking code-base, and would likely generate much more efficient run-times. The most computationally intensive portion of the plume tracking code is track maintenance and pruning, which is the strength

of PQS.

6.2 Field testing

The first two years on this research topic invested time in the development of a physical sensor network platform that was never fully implemented. We attempted to leverage a custom designed sensor network system from Dartmouth College named MiniMe. Future field tests would likely benefit from a commodity sensor platform such as the Berkeley motes¹ due to the extreme time required to design and maintain a custom system. A recommended first stage experiment would be a limited indoor deployment with controlled wind (fans) and a controlled source with non-toxic gas such as CO_2 or H_2O vapor in the form of gas cylinders, a boiling water source, or a solid state humidifier. The OnewireTM protocol by Dallas Semiconductor can accommodate pre-manufactured weather stations for wind direction, and would easily interface with semiconductor gas sensors manufactured by Figaro, Inc. of Japan. Water vapor semiconductor gas sensors exhibit the fastest response time, and the non-toxic nature of working with water vapor makes this an ideal choice for a “plume” source. Experiments performed at Caltech with mobile robots used water vapor as the plume gas of choice [27].

The next phase of testing would be the expansion from limited indoor experiments to outdoor field trials. This introduces the complexity of chaotic wind, as well as unpredictable temperatures and humidity. If water vapor is to continue being the plume agent of choice, lower temperatures and relative humidities would allow for the greatest sensitivity. Summertime conditions with high humidity and temperatures would create a high background signal, therefore winter conditions are predicted to be ideal for testing.

6.3 Expansion to other domains

6.3.1 Mobile source problem

Some work has been performed on the mobile plume problem, however this problem remains mostly untouched. The extension of stationary plume source work to this application is a logical next step. The goal of the mobile source problem is to estimate the presence and location of an emitting source that could be hidden inside a shipping container, and the transportation

¹<http://www.xbow.com/>

infrastructure becomes the vector along which the source can travel. The problem is essentially the same as the stationary source location inverse problem, with reduced dimensionality due to more concentrated prior distributions on source probability. (Location of the source is constrained to a preexisting port, roadway, or interstate as opposed to an arbitrary unknown location). In the implemented simulation, roadways are approximated as a Manhattan geometry where driving decisions of the target at each intersection are made with a Markov model characteristic of the particular vehicle. The dynamics of plume are now superimposed with dynamics of vehicle. In the case of velocities four possible scenarios:

- No wind, no target motion (isotropic plume)
- Wind only, static source
- Moving source, no wind
- Moving source, wind

One desired property would be to separate sources based on observations correlating to different models of vehicle behavior. If two vehicles have sufficiently different behavior models, we would expect the PQS could perform source separation based on chemical observations in the region of the mobile plume. The increased complexity is due to the moving source location. We seek to apply the same Process Query System techniques applied in previous work to the problem of shipping and transportation monitoring infrastructure.

Vehicle movement and source emission in the forward simulation begin at location $x_i y_j$ and we consider the transport vehicle to have a sequence of states over time determined by position, velocity, and driving behavior. The emission characteristics of the source are identical at this stage to the source used in the stationary inverse source problem. The difference is that instead of a continual release injecting material at the same $x_i y_j$ location at each time iteration of the simulation, it has a characteristic vehicle movement, and may transition to a neighbor cell. For velocity V_x, V_y at t_i :

$$x_{i+1} = x_i + V_x$$

and

$$y_{j+1} = y_j + V_y.$$

Velocity as a function of time is determined by the constraints of a Manhattan geometry, therefore only velocities that conform to the current road orientation are allowed. Future simulations can easily import real road vectors from GIS data for matching real road information. Velocity state transitions can occur at “intersections” which are the points at which two allowable travel vectors cross. In the current implementation all roads are rectilinear, meaning at each intersection a vehicle has four possible options, which are calculated with the state transition matrix a_{ij} . A vehicle can stay in its current velocity state, or transition into any of the other allowable states at each intersection. A Markov model estimates can be adjusted to include more states arbitrarily. We consider a sequence of velocity states at increasing times t as $v(t)$. A specific sequence of these states is denoted

$$v^T = \{v(1), v(2), v(3), \dots, v(T)\},$$

which is a finite array of vehicle velocities which along with the initial position describe a unique vehicle path. In the example vehicle simulation, the vehicle can exist in any of the four states corresponding to $v(t) = \{1, 2, 3, 4\}$ (North, South, East and West), and has asymmetric transition probabilities. In this example the vehicle has a tendency to stay in its current direction (70%), and may turn into any of the other 3 directions with a lower probability (10%). In this transition matrix example,

$$a_{ij} = \begin{pmatrix} .7 & .1 & .1 & .1 \\ .1 & .7 & .1 & .1 \\ .1 & .1 & .7 & .1 \\ .1 & .1 & .1 & .7 \end{pmatrix}.$$

The model for state transition of $v(t)$ is given by:

$$P(v_j(t+1)|v_i(t)) = a_{ij}.$$

Sensors are deployed in the region of transport with one sensor per cell or sector. The example simulation divides a region A into 30 sectors, each is assigned one sensor in a random location within the cell, and the total size of A is $m = n = 256$. This assumes some degree of uniform sampling across A , but also allows for the error in sensor placement, as well as the possibility of multiple pathways within in single cell. Sensors are activated in a binary fashion, just as in the

previous work whenever a threshold level is detected. Wind direction and magnitude across A are similarly varied, and available at each sensor node for the calculation of a wind history vector and plume predictors. Once observations arrive at sensors in the network the wind history vectors can be estimated against neighbor sensor nodes as well as overlapping transportation vectors. Plume predictors overlapping a road are reported as observations with a high likelihood, and the observation location is defined as the roadway position within the plume predictor with highest value determined by the plume predictor function. In this way an extra constraint is added to the system when compared to the completely uniform distribution experienced in the stationary implementation.

Mobile Source Tracking Queries

We need a method for expressing our “queries” of mobile plume events. Using the PQS approach, we represent a plume query as a set points in a two dimensional space that have plume events originating at specific times. These queries are submitted to a server running a PQS-like engine and are correlated to all available observations until a match is found. Multiple queries running on PQS-Plume would correspond to multiple explanations of the same data. The model query handles plume origin, multiple source origins, amount of mass released, and the time of release. An example query would be, “Was there a medium sized release at 9am this morning along Interstate 95?” This high level query contains location, magnitude, and a time which may correlate with sets of observations collected at a future time. A query could also contain sets of locations, such as “Was there a release of material on multiple trains near New York City at 9AM?” If the observations within the sensor network correlate highly enough with this query, this query will be returned as a highly ranked hypothesis. The idea of PQS is that multiple competing queries may explain the same observation set. Rankings of returned hypotheses may change temporally as new observations are available.

Queries describe points in the region A , or sets of points in A . The query asks if a plume event has originated within A at time t_k at location (i, j) , or a set of (i, j) locations. These release points correspond to the unknown release matrix R^T which is the set of all release points across time. It is important to also define the concepts of track and hypothesis. A track is a collection of plume observations all related to the same plume release point, and can take on an arbitrary shape in A depending on wind conditions. For example if there was no wind and the agent dispersed uniformly in all directions, all the observations in that circle of dispersion would be assigned to

the same track. This notion of a track is somewhat different than the traditional idea in physical target tracking research. Unlike the tracking of a rigid object such as an airplane, the shape and size of the object being tracking changes over time. For this reason a track is more loosely defined as observations originating from the same event, and typically take the form of a meandering flow region or plume. A hypothesis is a collection of tracks explaining the entire set of plume observations, therefore multiple hypotheses typically exist for the same set of observations.

One problem with representing plume queries is the large state space for a practical application. To represent all possible location queries for a region A of size $n \times n$ would require 2^{n^2} different queries. That means a grid of size 2×2 with each cell in the binary state of “release” or “no-release” results in 16 different queries, and a grid size of 200×200 would require more than 10^{12041} queries to fully represent the space! This only deals with permutations of possible binary release cells, and does not consider concentrations within each cell. Obviously a state reduction scheme is required.

The concept of scalable or tunable models allows the iterative refinement of models from course to finer resolution. Model refinement allows the running of rough models that automatically refine over time when activated. Analogous to two dimensional optimization methods, or well-known root-finding methods such as Newton’s Method, first the local maximum is located at low resolution. This grid position of local maximum is then subdivided and re-optimized by another order. Taking once again the 256×256 grid example, it could be divided into 4 cells resulting in 16 possible states or queries. Once one of the 16 queries is ranked above a set threshold of probability, a new branch of queries could be automatically generated seeking higher resolution. Instead of submitting a query for a plume origin at a pinpointed location, a query can specify a large region of a city. Once this course query receives a non-trivial ranking, the query can be upgraded to finer resolution and be allowed to consume more computational resources. In addition targets within of a city may be of higher value than others (roads passing near monuments such as the Golden Gate Bridge, or other critical infrastructure) and these regions within a query should be represented at high resolutions by default. Variation of the grid resolution as required allows a tremendous state reduction and produces a less intractable model.

6.3.2 Trusted Corridors

The distributed process detection approach to trusted corridor threat identification requires data sampled across a range of spatial-temporal locations. Rather than using the currently available sensor data from a small number of weigh stations, this effort relies on simulated data that will

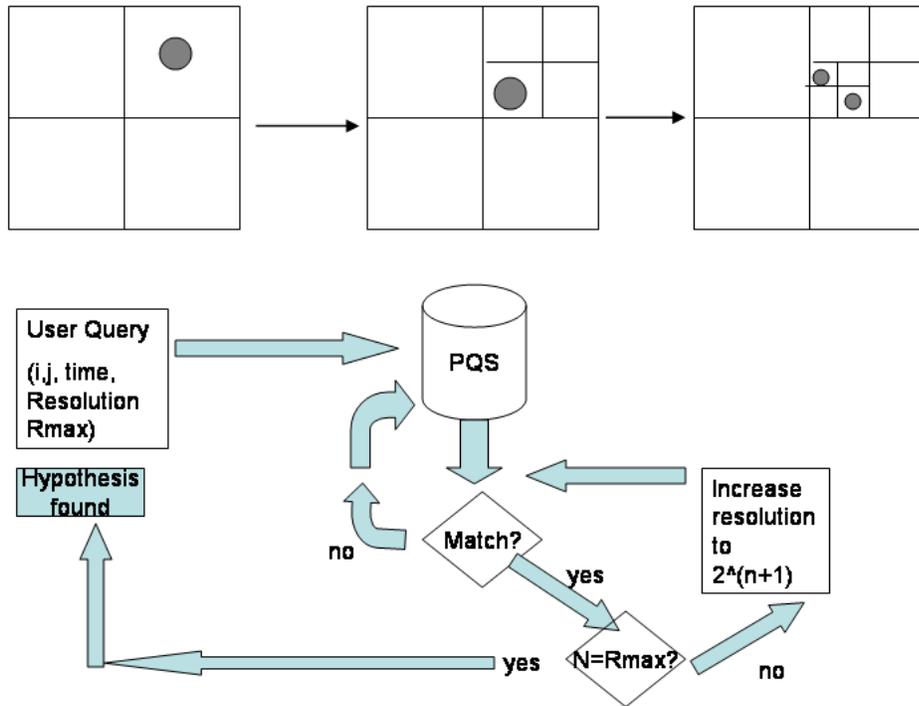


Figure 6.1: Tunable Models in 2D: First Model has $2^4 = 16$ possible states. The activated cell is refined to 4 new cells. This zooming model feature allows higher resolution models in areas of interest.

illustrate the increased value of a larger number of sensor locations along a specific corridor. This combination of large numbers of airborne substance sensors with relatively few high fidelity weigh stations serves to leverage data available across a larger temporal-spatial region. Detections missed at weigh stations can then be indicated across a larger network of inexpensive low resolution sensors. The power of process detection lies in correlation of large numbers of disparate sensor observations and finding collections of observations which support a single process.

Weigh stations provide identification of vehicles via manifest documents and serve as entry and exit points for the control of a specific corridor. Once a vehicle has entered a particular controlled corridor instrumented with sensors capable of chemical, biological, radiological, nuclear, and explosives detection (CBRNE) these distributed observations can be used in conjunction with weigh station identification information to track specific high risk vehicles. The type of information available at weigh stations alone may not be sufficient to identify suspicious traffic, especially given the risk of missed detections with short stopover times. The process detection component of this approach would flag a suspicious observation in the sensor network between weigh stations and using tracking algorithms predict when the suspicious vehicle is to arrive at the next control point.

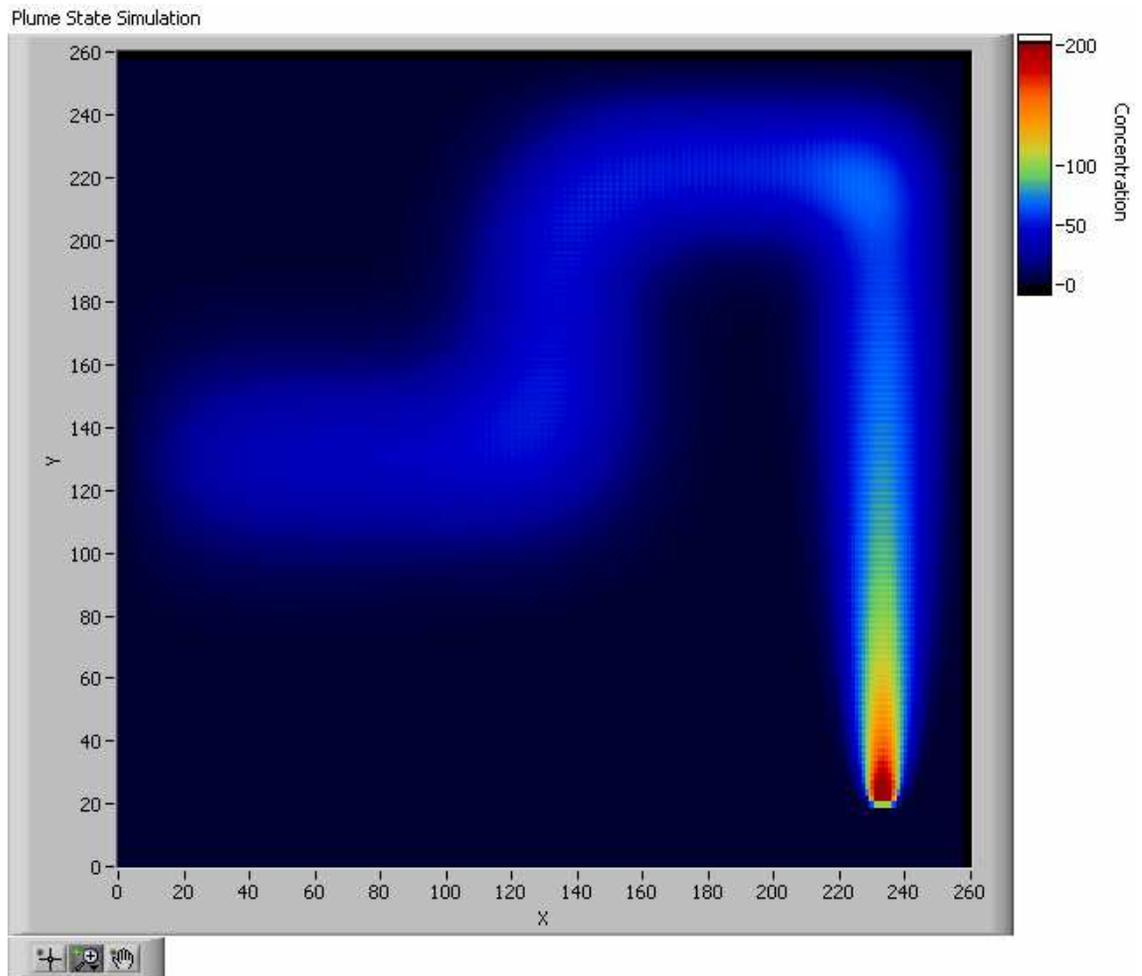


Figure 6.3: Ideal mobile source forward process, no wind

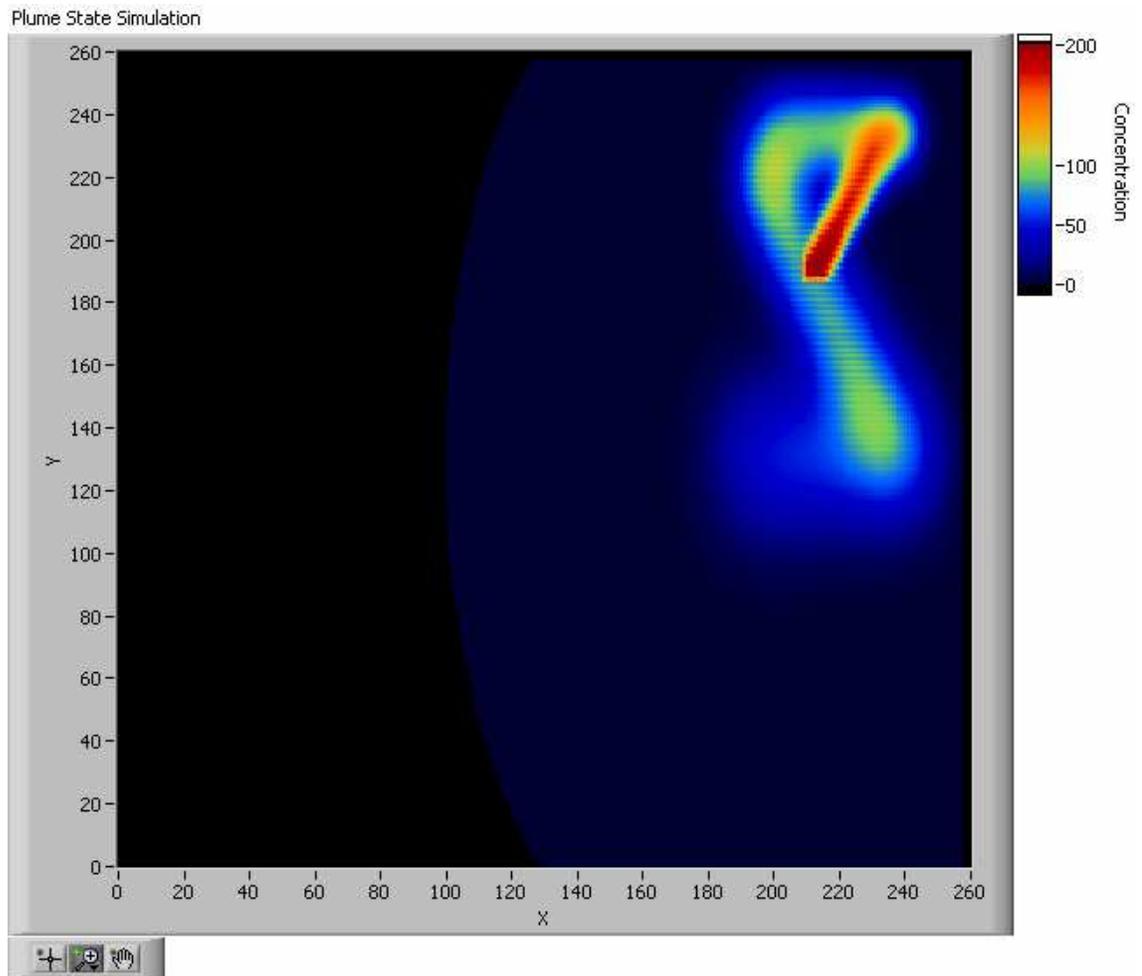


Figure 6.4: Mobile source in addition to wind process

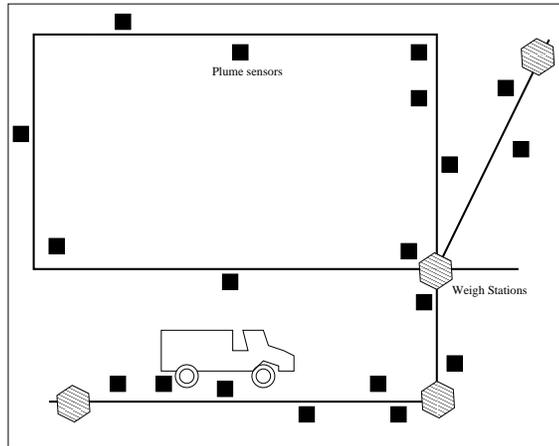


Figure 6.5: Coordination between more complex weigh stations collecting manifest data and highly distributed low resolution CBNRE sensing assets.

This hybrid approach will allow for advanced warning and the ability for intervention at stations further downstream on the controlled corridor.

6.3.3 Information diffusion

The dynamics of information propagation and the flow of topics through populations has been studied recently. By characterizing the process of information propagation from individual to individual, drawing on the infectious disease model, a diffusion based model of information flow throughout societies has been developed [25]. If we consider the blogosphere as a medium through which information “plumes” diffuse, many of the same concepts developed in this thesis could be applied to such a social system. Binary detections, observation partitions, tracks, state estimates, and “plume sources” all could be transferred into this problem. In the case of source detection or source counting, the goal of such a system would be correctly identifying the origins of ideas and information within the Internet’s interconnected web blogs known as the blogosphere. The same principles of source attribution, source separation, and source location likelihoods could be adapted into this blogosphere tracking problem. More generally, any system exhibiting diffusion as an underlying process could adapt the models developed in this thesis for the purpose of tracking.

Appendix A

Code descriptions

This appendix contains information about the simulation system hierarchy (written in LabVIEW and MATLAB programming languages) which would be useful for future use of the existing code libraries.

A.1 Main simulation

This simulation consists of a main panel user interface to view the simulation progress. The left pane show plume sources and concentrations within the space as they evolve in real time. This acts as the hidden state space, that is the sensors do not have access to this global image. The simulation window is a display of plume sources, sensors, tracks, and can also be overlaid with map files representing a geographical area for reference. For high quality graphical output images can be exported to MATLAB. This screen reports the final statistics after long batch runs.

A.2 Scenario generator

This interface allows the custom design of scenarios which can be saved to file and recalled in batch mode for large scale processing. Parameters include wind, sensor placement, grid size, wind type, tracking parameters, and source placement.

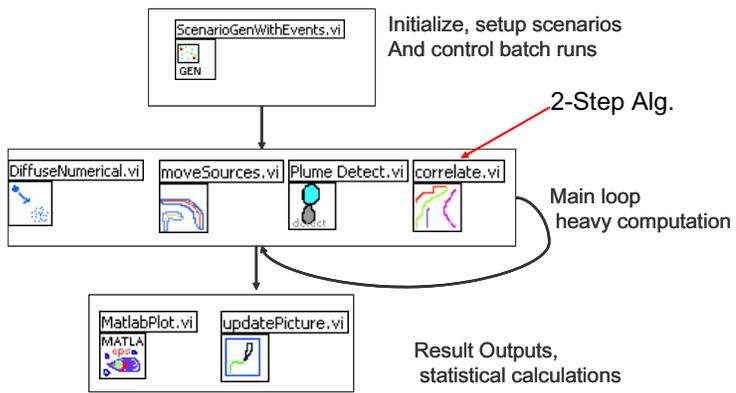


Figure A.1: LabVIEW simulation hierarchy

Simulation Hierarchy

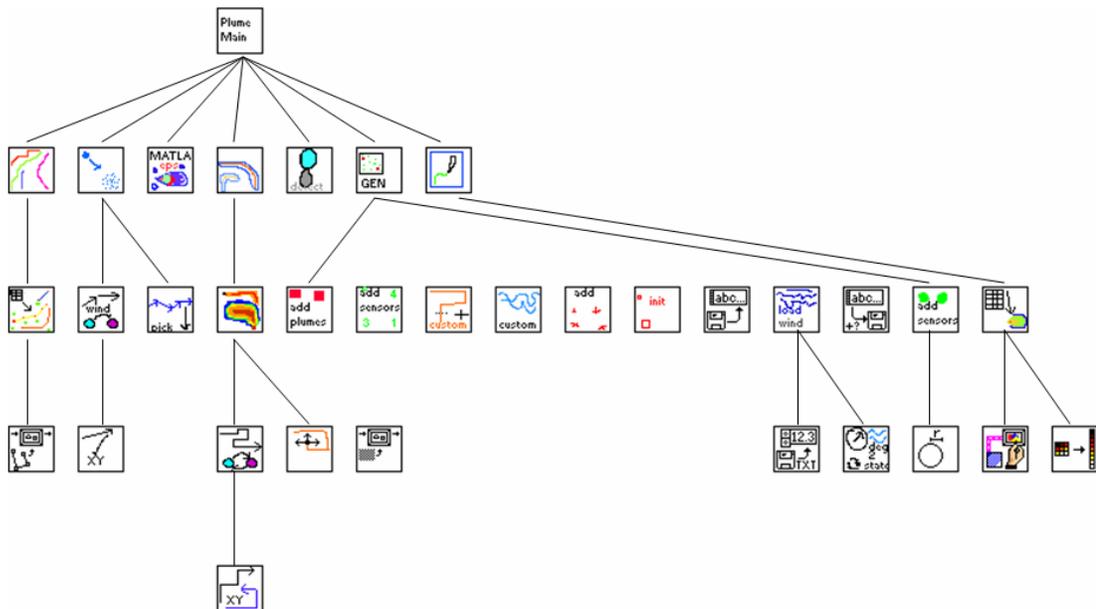


Figure A.2: Functional hierarchy of LabVIEW algorithms developed for this simulation. This chart only shows the top 5 levels out of 9 total. The second tier algorithms (7) are the core top level reusable algorithms that were developed.

A.3 Wind editor

The wind editor loads files downloaded from NOAA, performs filtering to remove missed observations, calculates wind statistics, and allows the user to select, edit, and view custom portions of large wind time series to be used in the simulations.

Appendix B

Variables, constants, and definitions

A	Region of size $m \times n$ containing cells
$S(A)$	State of region A
TAD	Total Area of Detectability
$F(S_t)$	Function operating on state S
T	Time interval
Z^T	Set of all observations, where $Z^T = \{z_1, z_2, z_3, \dots, z_T\}$
$z_t(i, j)$	Observation at time t for cell (i, j)
x_t	State at time t
X^T	Set of all true states, where $X^T = \{x_1, x_2, x_3, \dots, x_T\}$
\hat{x}_t	Estimate of state at time t
\hat{X}^T	Estimate of states
R^T	Release matrix
$M(A)$	Likelihood map of A
σ	Standard deviation
$C_t(i, j)$	Concentration matrix at time t

$R_{t_k}(i, j)$	Release matrix at time t_k . Each cell takes on a value of 1 or 0.
$S_t(A)$	State of A at time t
$Thresh_{min}$	Threshold for binary detection
P_e	Peclet number, ratio between advection and diffusion
D	Diffusion constant
W	Wind history vector where $W = \{w_{t_1}, w_{t_2}, \dots, w_{t_{i-2}}, w_{t_{i-1}}, w_{t_i}\}$
Z^i	Observation
S_{t_0}	Original event
$S_{m,n}$	Node placed on the space
t_0	Beginning of the simulation
CH4	
$A_{i,j}$	Cell in A
w	Width of patch for sensing
Θ	Wind history vector, where $\Theta = \{\theta_1, \theta_2, \dots, \theta_N\}$
M	Plume Source matrix
S_i	Sensor positions, where each sensor S_i has a position x_i, y_i
S_i^*	Sensor positions with detections
$\Theta(x_i y_i)$	Wind history vector at sensor S_i
$Corr(S_i^* S_j^* \theta_t)$	Correlation value between two sensors with detections at time t
N	Sample size of wind
T_{on}	Release times for sources, continuous
T_{off}	Release time for source
$\Gamma(\tau, N)$	Track scoring function

Γ_{thresh}	Track scoring function threshold
τ	Time since last data association in a track
N_{th}	Threshold length of a track for track maintenance
$M(i, j)$	Likelihood value for cell ij in A
$ML(M)$	Maximum likelihood of map M $\text{argmax}(M)$
$P(M)$	Performance of M where
	$\frac{M(i, j)}{ML(M)} = P(M)$
ω	Partition of an observation set
O^T	Observation set
T	Duration of surveillance
R	Region of surveillance, where $R = [0, m] \times [0, n]$
Ω	All possible partitions of an observation set
CAR	Correct assignment ratio
$ICAR$	Incorrect to correct assignment ratio
$S_{i,j}$	Source released at cell i, j
S	Source which has attributes $S = (x, y, \sigma, t_0, t_1)$
$S(A)$	concentration values of cells within A for prior times, $S(A) = [C(x, y)]^T$
λ_f	Probability of False Alarm rate
p_d	Detection probability
ω	Partition ω is one set of possible assignment of observations, where $\omega \in \Omega$. Ω is the set of all possible partitions given one set of observations Z^T ;
ω^*	Truth partition ω^* is the best possible partition of observations;
$SA(\omega)$	The set of all of associations in ω , $SA(\omega)$, Where $SA(\omega) = \{(\tau, t_i^\tau, t_{i+1}^\tau) : i = 1, \dots, \tau - 1, \tau \in \omega\}$, where t_i^τ is the time a track τ is observed i times;

$CA(\omega)$ The set of all correct associations in ω , $CA(\omega)$;

Bibliography

- [1] Vincent H. Berk, *Process Query Systems*, 2006.
- [2] Samuel Blackman and Robert Popoli, *Design and Analysis of Modern Tracking Systems*, Artech House, 1999.
- [3] Samuel S. Blackman, *Multiple-Target Tracking with Radar Applications*, Artech House, Inc., 1986.
- [4] Eli Brookner, *Tracking and Kalman Filtering Made Easy*, John Wiley and Sons, 1998.
- [5] David B. Caruso, *BioWatch' to Sound Alarm; Monitors Screen Air Quality for Bacteria Attack*, Houston Chronicle **July 29** (2003), p. A11.
- [6] ———, *Devices Sniff Out Bioterror, But Experts Question Effectiveness*, The Associated Press **July 11** (2003).
- [7] National Hurricane Center, <http://www.nhc.noaa.gov/> (2005).
- [8] T.M Chin and A.J Mariano, *Kalman filtering of large-scale geophysical flows by approximations based on Markov random field and wavelet*, International Conference on Acoustics, Speech, and Signal Processing **5** (1995), 2785 – 2788.
- [9] Mark Coates, *Distributed Particle Filters for Sensor Networks*, IPSN 2004 (2003).
- [10] John Crank, *The Mathematics of Diffusion*, Oxford University, London, 1975.
- [11] Bernoit R. Cushman-Roisin and E. Deleersnijder, *Introduction to Geophysical Fluid Dynamics - Physical and Numerical Aspects*, under contract with Academic Press, Dartmouth College Hanover, NH 03755, 2005.

- [12] George Cybenko and Vincent Berk, *Process Query Systems*, IEEE Computer **January** (2007), p. 62–70.
- [13] Robert Duncomb David Yong Douglas J. Holzhauser, George Ramseyer, *Chemical/Biological Plume Analysis Knowledge Source (CPAKS)*.
- [14] Richard O. Duda, Peter E. Hart, and David G. Stork, *Pattern Classification*, John Wiley & Sons, Inc., 2001.
- [15] Petula Dvorak, *Health Officials Vigilant for Illness After Sensors Detect Bacteria on Mall - Agent Found as Protests Drew Thousands of Visitors*, The Washington Post **Sunday October 2** (2005), Page C13.
- [16] Alaeddin Elayyan and Victor Isakov, *On an Inverse Diffusion Problem*, SIAM Journal on Applied Mathematics (1997).
- [17] Deborah Estrin, Ramesh Govindan, John Heidemann, and Satish Kumar, *Next century challenges: scalable coordination in sensor networks*, International Conference on Mobile Computing and Networking (1999), Pages: 263 – 270.
- [18] Geir Evensen, *Sequential Data Assimilation With a Non-linear Quasi-Geostrophic Model Using Monte Carlo Methods to Forecast Error Statistics*, Journal of Geophysical Research **99** (1994), no. C5.
- [19] Geir Evensen, *The Ensemble Kalman Filter: theoretical formulation and practical implementation*, Ocean Dynamics **Volume 53** (2003).
- [20] Jay A. Farrell, Shuo Peng, and Wei Li, *Plume Mapping via Hidden Markov Methods*, IEEE Transactions on Systems, Man, and Cybernetics **33** (2003), no. 6.
- [21] Chengwei Feng, *Application of Scientific Visualization and Quantification in Plume Study*, web, 1999.
- [22] Patrick D. French, John Lovell, and Nelson L. Seaman, *Distributed Sensor Network for local area atmospheric modeling*, Unattended ground sensor technologies and applications V, proceedings of SPIE **5090** (2003), 214–224.
- [23] Francisco M. Garcia and Isabel M. G. Lourtie, *Estimation Of Locally Stationary Covariance Matrices From Data*, IEEE ICASSP **VI** (2003), 737.

- [24] W.R. Gilks and S. Richardson, *Markov chain Monte carlo In Practice*, CRC Press, 1998.
- [25] Daniel Gruhl, R. Guha, David Liben-Nowell, and Andrew Tomkins, *Information diffusion through blogspace*, Proceedings of the 13th international conference on World Wide Web (2004), 491 – 501.
- [26] David L. Hall and Sonya H. McMullen, *Mathematical Techniques in Multisensor Data Fusion*, Artech House, 2004.
- [27] A. Hayes, A. Martinoli, and R. Goodman, *Swarm robotic odor localization*, 2001.
- [28] Adam T. Hayes, Alcherio Martinoli, and Rodney M. Goodman, *Distributed Odor Source Localization*, IEEE Sensors Journal **2** (2002), no. 3.
- [29] Hiroshi Ishida, Gouki Nakayama, Takamichi Nakamoto, and Toyosaka Morizumi, *Controlling a Gas/Odor Plume-Tracking Robot Based on Transient Responses of Gas Sensors*, IEEE Sensors Journal **5** (2005).
- [30] C.D. Jones, *On The Structure of Instantaneous Plumes In the Atmosphere*, Journal of Harardous Materials (1983).
- [31] S. Kazadi, R. Goodman, D. Tsikata, D. Green, and H. Lin, *An Autonomous Water Vapor Plume Tracking Robot Using Passive Resistive Polymer Sensors*, Autonomous Robots **9** (2000), no. 2, 175–188.
- [32] C. Robert Kline, *Optimizing Active and Passive Countermeasures*, IEEE Engineering In Medicine And Biology Magazine **January/ February** (2004), pages. 102–109.
- [33] Susan Levine and Sari Horwitz, *Test Results Cited in Delay of Mall Alert - CDC Explains Why Local Officials Weren't Told for Days About Bacterium Detection*, The Washington Post **Wednesday, October 5** (2005), Page B01.
- [34] Daniel Marthaler and Andrea L. Bertozzi, *Tracking environmental level sets with autonomous vehicles*.
- [35] Andrew W. Moore, *Hidden Markov Models*, Online pdf tutorial CMU, 2003.
- [36] Suman Nath, Amol Deshpande, phillip b. gibbons, and Srinivasan Seshan, *Mining a world of Smart Sensors*, Mobicom, 2002.

- [37] Glenn Nofsinger and George Cybenko, *Airborne Plume Tracking With Sensor Networks*, Proceedings of the SPIE , Defense and Security Symposium - Conference 6231 - Unattended Ground, Sea, and Air Sensor Technologies and Applications VII (April 2006).
- [38] _____, *Distributed Chemical Plume Process Detection*, IEEE Military Communications Conference, MILCOM (October 2005).
- [39] Glenn Nofsinger and Keston Smith, *Plume Source Detection Using a Process Query System*, Defense and Security Symposium 2004 Proceedings (Bellingham, Washington), SPIE, April 2004.
- [40] Songhwai Oh, Luca Schenato, and Shankar Sastry, *A Hierarchical Multiple-Target Tracking Algorithm for Sensor Networks*, Proc. of the International Conference on Robotics and Automation (2005).
- [41] Elaine S. Oran and Jay P. Boris, *Numerical Simulation of Reactive Flow*, Elsevier, 1987.
- [42] Ian L. Pepper, Charles P. Gerba, and Mark L. Brusseau, *Pollution Science*, Academic Press, 1996.
- [43] Rabiner, *A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition*, Proceedings of the IEEE **77** (1989), no. 2.
- [44] Donald B. Reid, *An Algorithm for Tracking Multiple Targets*, IEEE Transactions on Automatic Control **AC-24** (1979), no. 6, 843–854.
- [45] Andrew Russell, *Using Volatile Chemicals to Help Locate Targets in Complex Environments*.
- [46] Jerald L. Schnoor, *Environmental Modeling*, John Wiley & Sons, Inc., 1996.
- [47] Joris De Schutter, Jan De Geeter, Tine Lefebvre, and Herman Bruyninckx, *Kalman Filters: A Tutorial*.
- [48] Dana A. Shea and Sarah A. Lister, *The BioWatch Program: Detection of Bioterrorism*, Congressional Research Service Report **No. RL 32152** (November 19, 2003), <http://www.fas.org/sgp/crs/terror/RL32152.html>.
- [49] Albert Tarantola, *Inverse Problem Theory*, Society for Industrial and Applied Mathematics, 2005.

- [50] Feang Zhao and Leonidas Guibas, *Wireless Sensor Networks An Information Processing Approach*, Morgan Kaufmann, 2004.
- [51] Feng Zhao, Jie Liu, Leonidas Guibas, and James Reich, *Collaborative Signal and Information Processing: An Information Directed Approach*, Proceedings of the IEEE **91** (2003), no. 8, 1199–1209.