# Airborne Plume Tracking With Sensor Networks

Glenn T. Nofsinger and George V. Cybenko
Dartmouth College, Hanover, NH 03755 USA

## ABSTRACT

This paper presents a framework and demonstrates results from a process detection based approach to tracking an airborne plume in sensor networks. Data integration and pattern detection in large sensor networks measuring gas and radiation plumes suffer from low resolution observations, missed detections, and numerous false positive reports. Large numbers of nodes and the hypothesis management concept of a Process Query System (PQS) can compensate for lower data quality. A result of the process detection based approach to this problem is models that can be implemented in many different scenarios. Plume predictor models are illustrated which allow data association between sensor nodes in typical outdoor wind conditions.

We demonstrate a simulation of a mobile plume source in a sensor network designed for use in the same PQS. A kinematic model is developed for a vehicle carrying a plume source. Inverse models for this mobile plume source will work in conjunction with the existing software systems, thus allowing PQS to rapidly be adapted to a new problem domain with minimal modifications. This scenario of a mobile airborne plume source approximates a moving container emitting a detectable substance in a transportation network, where the container movement is restricted by existing vehicle corridors.

**Keywords:** plume tracking, process detection, sensor networks, transportation and shipping security

## 1. INTRODUCTION

The physical world will eventually be highly sampled by dense intelligent devices with impressive sensing abilities. We make the assumption that existing infrastructures of small inexpensive sensor networks will also include the ability to measure atmospheric contaminants in the near future. Beyond the collection of this dense data, new methods for fully utilizing such massive data sets in real time will be required. We propose a method based on pseudo-physical models, and explore the limits of such a system. With the advent of sensor networking technologies, the promise of solving this type of distributed chemical plume information sharing problem has become real. We have already successfully demonstrated the tracking of vehicles, humans, and computer attacks with the use of distributed sensor networks and PQS. 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 low level components in sensor networks enables us to consider solving the plume problem with novel information theoretic approaches. For example, instead of solving the inverse diffusion equation with powerful supercomputers to invert observations - data from a high density sensor network can be treated as other target tracking problems. Because many of these sensor network technologies have matured recently, we consider leveraging them for the solution of the plume location problem an appropriate and interesting effort.

Innate properties of chemical plumes make for a very difficult inverse problem. Plume concentrations are very 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. For example it is common to measure a concentration of zero 80% of the time even in the proximity of the source . 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 first order naive 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. Traditional methods for inverse array signal processing such as signal intensity gradient following for sound or seismic

Further author information: (Send correspondence to Glenn Nofsinger)
Glenn Nofsinger: gtn@dartmouth.edu, Telephone 603 646 3332
George Cybenko: gvc@dartmouth.edu

vibrations following fail to transfer into the plume tracking domain due to the highly unpredictable nature of plume fluid dynamics.

Another challenge is that chemical observations are often of extremely low resolution, and often only indicate a positive or negative result. Common chemical 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.

In sensor networks, 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, however, we may be interested in knowing information about the plume history. Doing this requires solving the *inverse problem:* taking current observations and making assertions about the source. We solve the inverse problem using a novel concept called the Process Query System (PQS) which is well suited to overcome many of the inherent difficulties of such an inverse problem. The result and output from PQS is not a typical solution set, but a ranking of most likely hypotheses.

## 2. INVERSE STATIONARY SOURCE PROBLEM

One of the primary challenges of the target tracking method is making some kind of statistical correlation between observations made within the sensor field. Because the state of this plume space is so complex and all parameters can never be accounted for, all relevant probabilities can be known and a statistical approach is possible. The general method for a solution to this ill-posed inverse problem is based on Bayesian decision theory. For a stationary source, we only consider the dynamics of wind history and diffusion constant which result in a probability density called a "plume predictor" originating at a sensor node. Wind is generated in the the forward simulation by downloaded data from a field site in the region of interest. Real wind data then undergoes the commonly used bootstrap method for data replication. This allows a pseudo-random generation of wind data from a single real time series. Data sets exist on-line from urban, rural, and coastal weather sites. (NOAA and NWS) The resulting data has the same harmonic components as the original data, but randomized phase for each new trial simulation. Every activated sensor node in a trial (observation values above threshold) then overlaps this plume predictor pattern with neighbor nodes, thus forming a correlation value between sensor node pairs. This plume predictor shape is derived from the shape of the forward advection-diffusion equation, however run back-wards in time. The result is a likelihood region for each sensor observation that estimates probable origin locations for the current measurement, which allows the formation and clustering of sensor data into "tracks." A track is the collection of sensor observations with a high probability of originating from the same stationary (non-mobile) emission source.

### 2.1. Problem Formalization

Let $A$ be a two dimensional area, $A \subset \mathbb{R}^2$ , discretized into a regular grid of $m \times n$ cells. $A$ serves as a representation of the state space of the region we are attempting to monitor with sensors. Define the state of $A$, $S_t(A)$, 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 its position $(i, j)$ , where $i$ and $j$ are the column and row indices of the cell. Airborne agent can enter any cell within $A$, drift between cells, or leave the region $A$ by drifting off the edge of $A$. The concentration for a specific cell at time $t$, $C_t(i, j)$, is defined as the average amount of agent at $(i, j)$ within the most recent time interval.

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. More advanced models of $R$ could include a higher number of states for each cell, but 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 $t_k$.

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

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 being in the "release" state. In reality the $R^T$ sequence of states 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 $0 \rightarrow T$.

When an incident occurs within $A$, agent is released into cells by adding concentrations to release points. An incident may have any number of plume release points. 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 cells. This model allows an incident to be low concentration but covering a large number of cells, or 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 goal 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 $F(S^t)$. This physical advection-diffusion process $F(S^t)$ can be solved by the Gaussian solution to the well known differential equation describing advection-diffusion. In two dimensions, along with a linear term for wind this differential equation is:

$$\frac{\partial C}{\partial t} = D_x \frac{\partial^2 C}{\partial x^2} + D_y \frac{\partial^2 C}{\partial y^2} + \alpha \frac{\partial c}{\partial x} + \beta \frac{\partial c}{\partial y} \tag{2}$$

In this differential equation $D_x$ and $D_y$ diffusion constants are independent, and we assume diffusion is uniform (anisotropic diffusion) in $x$ and $y$ directions. Therefore $D_x = D_y = D$. The wind constant terms $\alpha$ and $\beta$ represent the corresponding linear wind velocities in the $x$ and $y$ directions.The solution to the differential equation describing diffusion and advection (wind) in 2 dimensions has the solution:

$$C(x, y, t) = \frac{A_1 A_2}{4\pi t \sqrt{D_x D_y}} e^{(-\frac{(x-\alpha*t)^2}{4D_x t} - \frac{(y-\beta*t)^2}{4D_y t})}$$

where $D_x$ $D_y$ are the diffusion constants, and $\alpha$ and $\beta$ are the $x$ and $y$ wind velocities. The shapes generated by substituting values for $x, y, t$ produce plume shapes that show the probability of a plume event reaching a destination.

## 2.2. Bayesian Observation Correlation

By looking at a plume event (call it event $A$ ) from a probabilistic perspective, we can also think about the probability of a detection event (call this event $B$) at some point near $A$. This probability distribution is basically the forward diffusion solution. The probability of a detection $B$ given that there is a release $A$ can also be written in terms of conditional probabilities:

$$P(B|A)$$

That is, what is the probability of a detection at $B$, given a known release $A$. Similarly one can consider the inverse probability, that is given an observation at $B$ what is the probability of an event $A$? That is, what is $P(A|B)$? This can be expressed in terms of $P(B|A)$ using Bayes Rule:

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

This will lead to the realization that the inverse probabilities are based on running the diffusion equation back-wards in time. For example, given the two events A and B, the following figures show the forward likelihoods for particle movement, along with the inverse probabilities of detection - and it can be seen that the shapes are identical but inverted

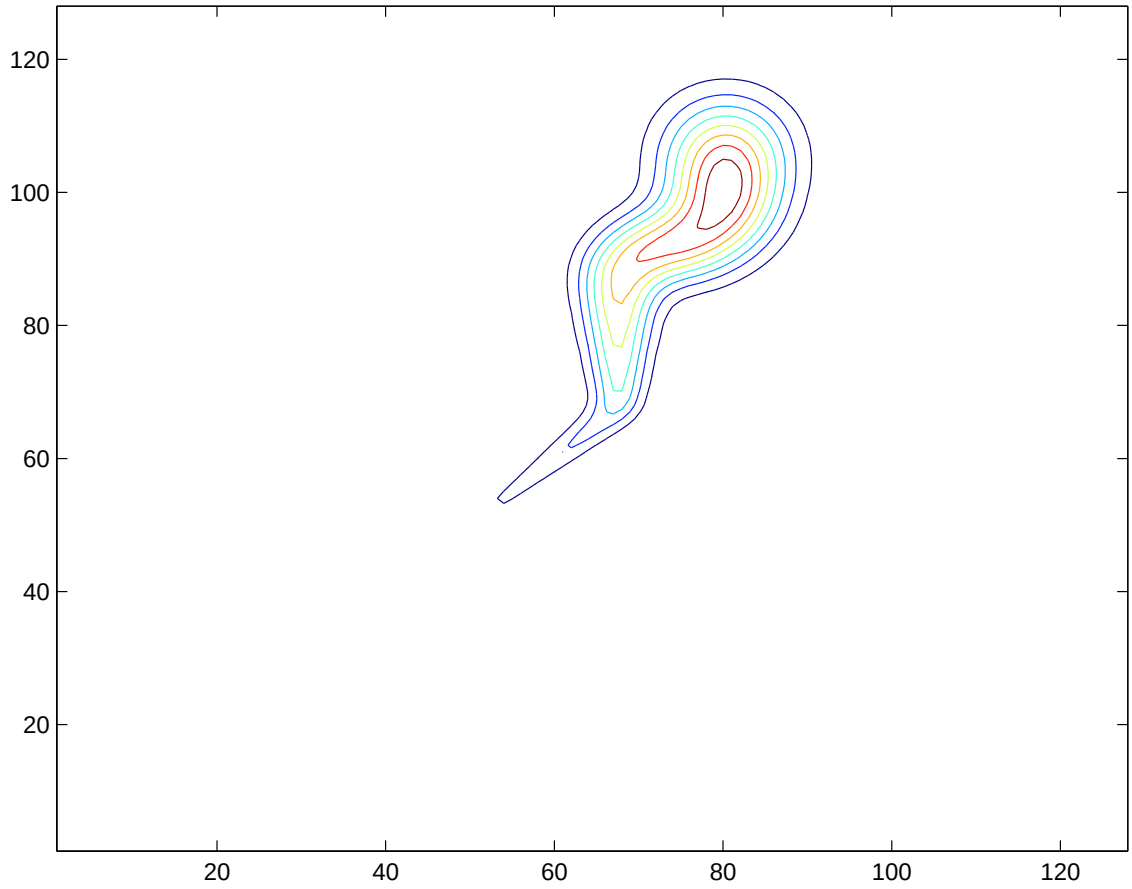


**Figure 1.** A complex plume predictor contour shape based on diffusion constant and wind history vector at a specific sensor origin. Oscillating winds are generated with bootstrapped data regenerated from true field data. For estimating the observation correlation likelihood between this sensor and any another sensor in the region a value is calculated based on this function.

in time. Knowing the wind history vector $W$ :($[W_{t_{i-1}}, ..., W_{t_{i-2}}, W_{t_{i-1}}, W_{t_i}]$) for each of the $N$ nodes, the substance of interests diffusion constant $D$, and the relative location of all the nodes in the network allows for the calculation of a plume predictor value for each new observation. More complex wind history vectors lead to more complex and potentially irregular probability regions as a predictor. The objective of this approach is to assign a probability that an observation at one sensor is related to an observation at a neighboring sensor, and thus allow data association and track formation.

One additional property of an ideal sensor is that a diffusion event will trigger a binary threshold sensor for a continuous period of time in which the observation is above threshold. In the ideal case this results in a rising exponential edge, followed by an exponential decay of signal. The sampling of the sensor at an arbitrary time in which it is activated does not disambiguate if the sensor is experiencing the rising edge of an event, or the falling edge of an event. This uncertainty leads to a region bounded by two concentric circles - one circle representing the leading edge of the plume event, and one representing the falling edge. The geolocation problem for a static source loses this degree of freedom if the sampling rate of the sensor is adequate. (faster than the duration of each activation period). In the fast sampling case, a time series of observations can be used for numerical differentiation in which a positive or negative slope information can be used to identify which segment of the plume event is currently active.

### 2.3. Software Implementation

As a prototype the plume tracking algorithms, forward diffusion scenarios, and plume observations were developed in the LabVIEW graphical programming environment. The simulation runs on the National Instruments runtime engine, and can also be compiled as a standalone executable. This rapid modeling language allowed for a fast development time, and

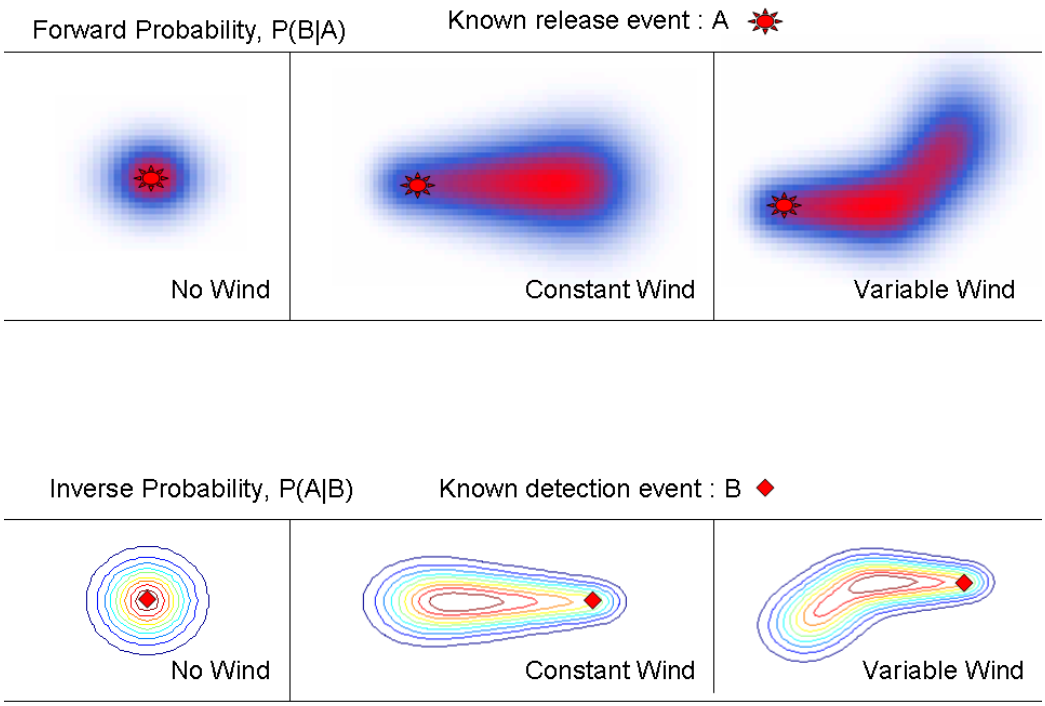


**Figure 2.** Forward and Inverse particle Probabilities as a result of the forward and inverse solutions to the advection-diffusion equation. A result of the Bayesian process. Collections of these probability relationships between nodes lead to tracks and estimates of plume origins.

produces a user friendly graphical interface for interacting with the simulation. This simulation consists of a 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 right pane is a display of plume sources, sensors, tracks, and can also be overlaid with map files representing a geographical area for reference.

Before setting up an experiment another module is invoked called the simulation scenario generator. Here a user can manually place sensors and sources. Define source size and concentration, as well as define parameters that effect wind such as transition probability and intensity. When more than a few nodes are desired or many trials are needed the scenario generator can place sensors and sources randomly within the confines of the 2D space.

Once a simulation begins running with defined wind, sensors, and plumes observations are collected at each sensor independently and reporting to the tracking engine. The tracking engine is a simple implementation of MHT, and makes the decision of track initiation, or appending the observation to an existing set of observations (a track). The result output will be a graphical representation of the most likely hypothesis set, which is a set of tracks explaining the reported sensor observations. Tracks are graphically differentiated by different colors, which are assigned at random.

## 3. INVERSE MOBILE SOURCE PROBLEM

The goal of 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 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 source is constrained to a pre-existing port, roadway, or interstate as opposed to an arbitrary unknown location). In the current simulation, roadways are

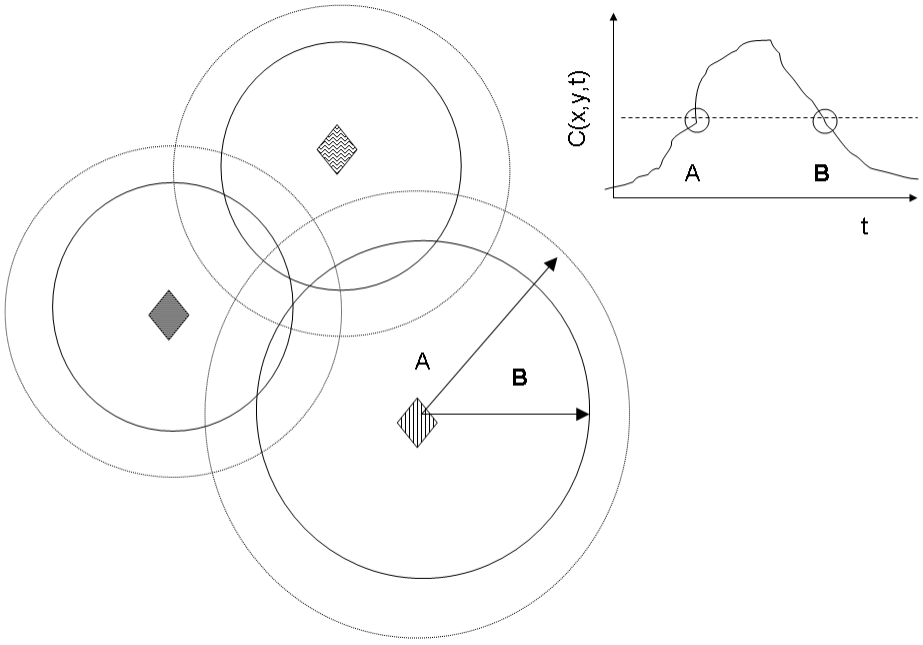


**Figure 3.** The inverse diffusion problem in a sensor network for a stationary source. The goal of this sensor network is to make the most accurate estimation of a single plume source origin location. In this network of three sensors a measurement above threshold can correspond to a region bounded by two circles - one for the leading edge and one for the falling edge above threshold. Overlapping regions between multiple sensors improve inverse geolocation ability.

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. 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.

### 3.1. Mobile Source

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

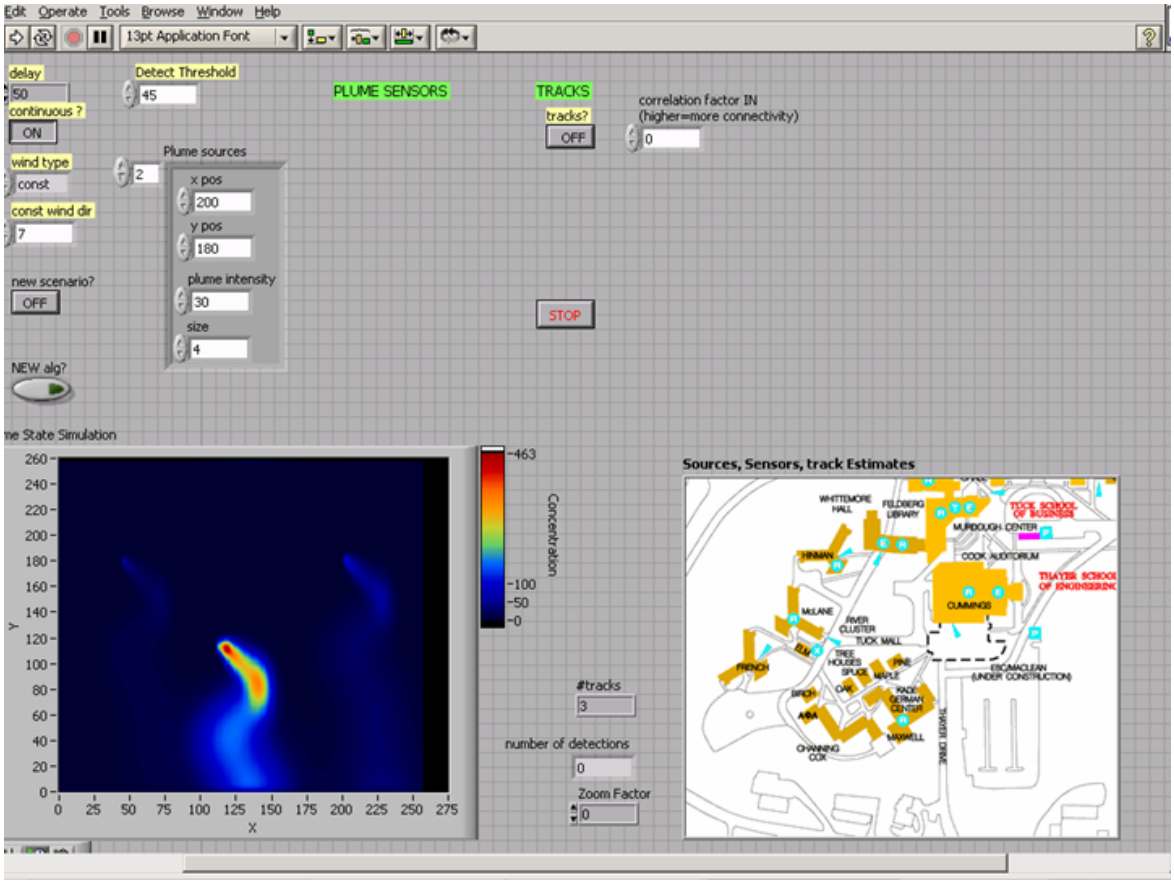


**Figure 4.** Plume graphical interface. The left intensity plot shows plume state space, and the right picture area displays sensors, tracks, as well as imported map backgrounds such as this example of the Dartmouth Campus.

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

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

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 tendencyto 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 roadwayposition within the plume predictor with highest value determined by the plume predictor funciton. In this way an extra constraint is added to the system when compared to the completely uniform distribution experienced in the stationary implementation.

### 3.2. 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 $2 \times 2$ with each cell in the binary state of "release" or "no-release" results in 16 different queries, and a grid size of $200 \times 200$ 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 sate 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 $256 \times 256$ 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 bee 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.

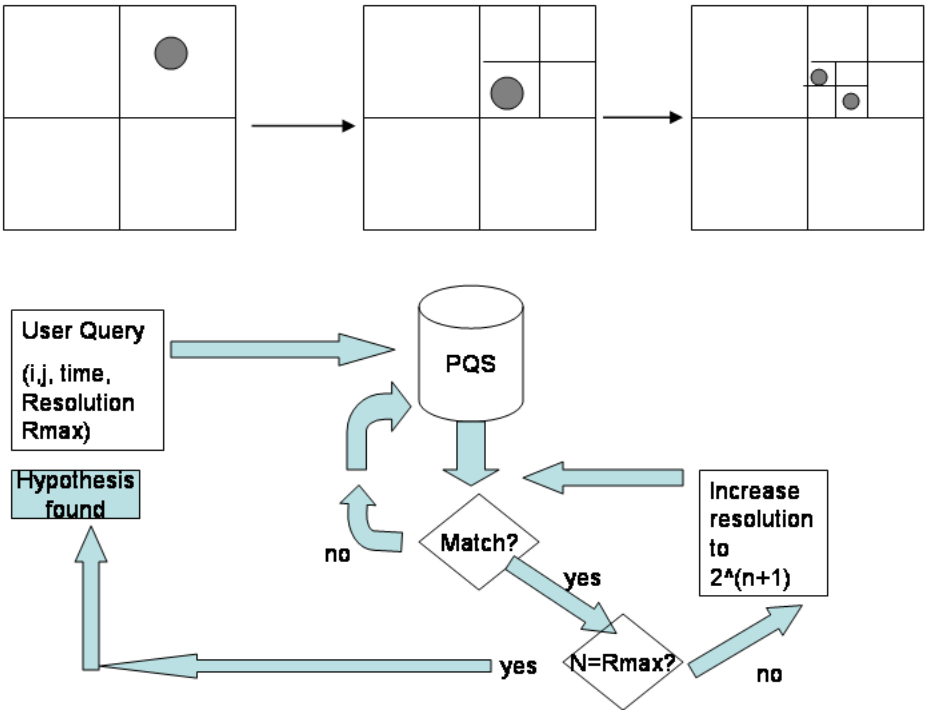


**Figure 5.** Tunable Models in 2D: First Model has $2^4 = 16$ possible states. The activated cell is refined to 4 new cells. This zooming model feature allows higher resolution models in areas of interest.

## ACKNOWLEDGMENTS

I would like to thank the members of my PhD thesis committee for their invaluable input and time commitment: George Cybenko, Eric Hansen, and Dan Rockmore of Dartmouth College. 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.

## REFERENCES

[1.] J. Crank, *The Mathematics of Diffusion,* Oxford University, 1974.

[2.] G. Cybenko, *An Overview of Process Query Systems,* Proceedings of the SPIE Vol. 5403, Sensors, and Command, Control, Communications, and Intelligence (C3I) Technologies for Homeland Security and Homeland Defense.

[3.] G. Nofsinger, *Plume Source Detection Using A Process Query System,* Defense and Security Symposium 2004 Proceedings.

[4.] K Tomas, *Thanks Dubai,* Forbes Magazine, March 27, 2006.

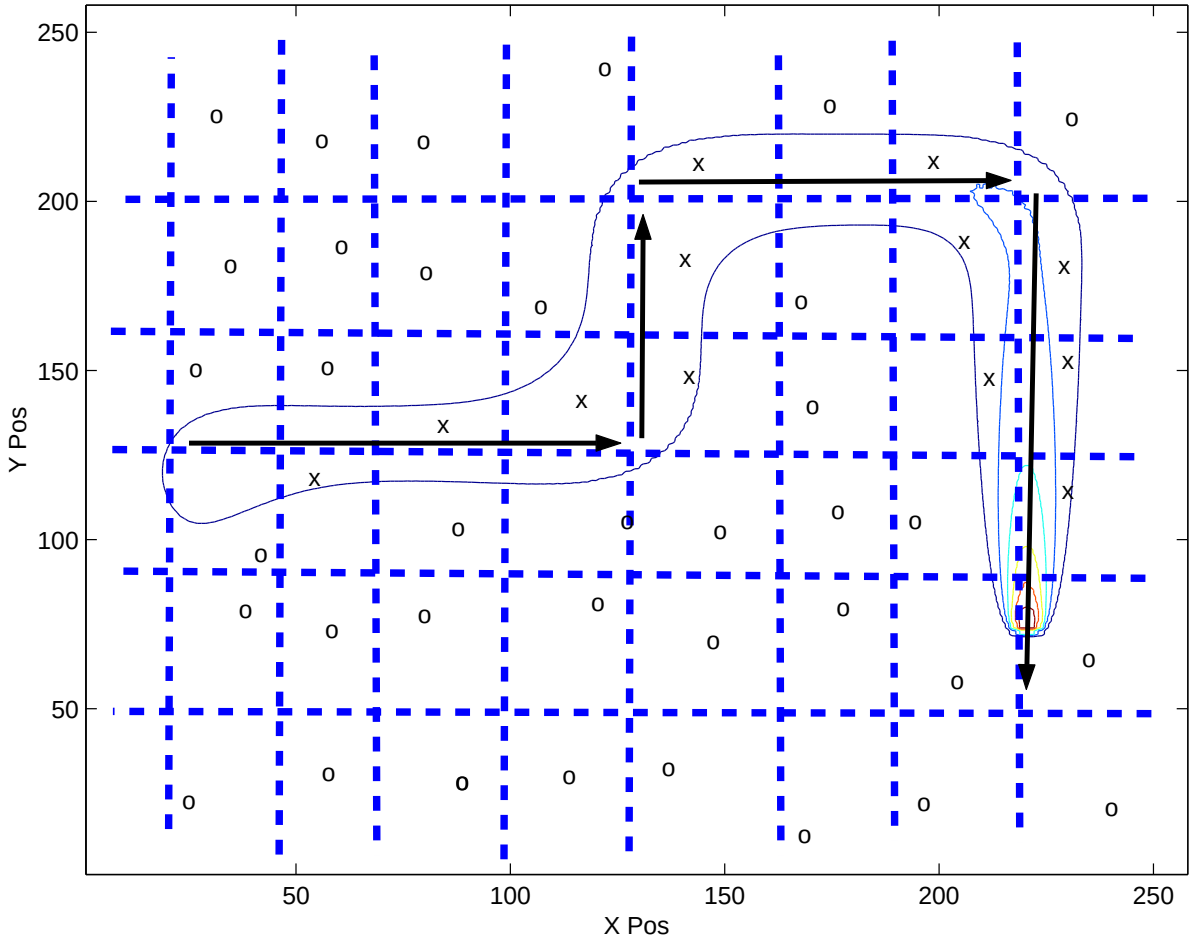


**Figure 6.** Mobile target moving with activated sensors (x) and inactive sensors (0), triggered by a threshold level. The contours show contamination concentration. The region $A$ is of size $m = n = 256$, and is divided into 40 cells, where each cell has one randomly placed sensor node. The dotted lines indicate allowable travel pathways, and the mobile source moves on this Manhattan geometry with a Markovian model at each intersection. The traversed path is indicated by the arrows.