

# Algorithms for Distributed Chemical Sensor Fusion

Scott Lundberg, Randy Paffenroth, and Jason Yosinski

Numerica Corporation, 4850 Hahns Peak Drive, Suite 200, Loveland, CO 80538, USA

## ABSTRACT

The fusion of Chemical, Biological, Radiological, and Nuclear (CBRN) sensor readings from both point and stand-off sensors requires a common space in which to perform estimation. In this paper we suggest a common representational space that allows us to properly assimilate measurements from a variety of different sources while still maintaining the ability to correctly model the structure of CBRN clouds. We design this space with sparse measurement data in mind in such a way that we can estimate not only the location of the cloud but also our uncertainty in that estimate. We contend that a treatment of the uncertainty of an estimate is essential in order to derive actionable information from any sensor system; especially for systems designed to operate with minimal sensor data. A companion paper<sup>1</sup> further extends and evaluates the uncertainty management introduced here for assimilating sensor measurements into a common representational space.

**Keywords:** Chemical detection, modeling, uncertainty, ambiguity, data fusion, filtering

## 1. INTRODUCTION

The tracking and characterization of CBRN threats will likely require the fusion of a variety of different sensor measurements. When detecting CBRN hazards, long range sensor instruments such as radar, infrared (IR), electro-optical, and long wave hyper spectral; short range instruments such as Raman spectrometers; and a wide array of point sensors such as ion mobility spectrometers (IMS) and chemical-resistor arrays can potentially be used for sensing. In this paper we seek to begin to address the integration and fusion of these different sensor measurement types into a common tracking space. IR spectrometers,<sup>2-5</sup> IMS,<sup>6</sup> and Raman Spectrometers<sup>7</sup> have been proven to be some of the most successful chemical detection methods and hence will form the motivation for the sensor models considered in this paper.

Radar and IR systems are currently used to track airborne objects in real-time. However, how to properly track and characterize CBRN releases in real-time is still an open problem. The fusion of all the various sensor types into a common tracking space is an important step when defining such a system capable of real-time CBRN release tracking. Our hope is to contribute to the understanding of possible common tracking spaces by examining spaces that can be represented using low dimensional parameterizations.

It is a challenging task to take the sensor measurement types defined above and integrate them into a common space. While one can imagine very high-dimensional spaces, such as those produced by gridding schemes and finite element methods, it is difficult to properly fill in such a large parameter space without strong models to constrain the space. This is particularly true in our case because we seek to provide results for scenarios where we have a relatively low number of weak sensors. Rather than attempt to fill in such a large parameter space, here we instead attempt to parameterize the models. By doing this we hope to dramatically reduce the number of parameters needed to describe the scenario, and hence also improve our performance in settings with poor sensor data.

We believe that while good estimates of CBRN entities are important, any estimate of them must also come with an *ambiguity* measure describing the uncertainty of the estimate. There is an extensive body of literature on the proper handling of ambiguity in target tracking problems<sup>8-13</sup> (and references therein) as well as a similar, but substantially smaller, body of work that addresses ambiguity in chemical detection problems.<sup>14-18</sup> While an ambiguity analysis has been performed on various *components* of a system for WMD consequence management,

---

Author E-mails: scott.lundberg@numerica.us, randy.paffenroth@numerica.us, and jason.yosinski@numerica.us

the authors are not aware of any studies that address the interactions between sources of ambiguity. Accordingly, herein the authors describe how a representation of the errors or uncertainty of the *system*, conspicuously absent from many other techniques, may be provided.

Another way to look at the importance of ambiguity is to note that the information provided to the user in CBRN scenarios *must be correct*. The worst possible sin in such life and death circumstances would be to provide the user with information that is not factual. Note, there is an important and subtle distinction here. We consider providing the user with information whose veracity is unknown, or ambiguous, and *not informing them of this fact* to be as bad as providing incorrect information. The act of informing the user that a suspect cloud is composed of water vapor means very different things depending on whether one is 99.999% sure it is water vapor or one is 51% sure it is water vapor and 49% sure it is nerve gas, even though both situations lead to the same “most probable estimate”.

We approach the problem of correctly merging sensor measurements into a common tracking space that correctly handles ambiguity from the perspective of classical target tracking. By formulating the problem using the recursive Bayesian estimation framework we can apply many of same techniques prevalent in the target tracking domain to the characterization of CBRN releases. In Section 2 we develop a low-dimensional tracking space that attempts to represent the cloud at a level of detail appropriate for a limited number of measurements. We consider the associated measurement functions for different sensor types as well as a parameterization for the common tracking space. In Section 3 we present preliminary qualitative results from a sample implementation of the ideas presented in Section 2. These simulations are intended to illustrate our low-dimensional parameterization approach.

## 2. LOW DIMENSIONAL CLOUD MODELS

We begin our discussion of a common representational space for CBRN clouds with a discussion of how best to parameterize CBRN clouds. Herein we propose to analyze clouds that have been parameterized by way of some *low dimensional* representation. The search for a low dimensional parameterization of clouds is driven by the desire to make maximal use of our limited measurement data. The more parameters that need to be estimated the more sensor data will be needed in order to provide not only an estimate of the cloud but also an estimate of the uncertainty in our model.

### 2.1 Notation

In this discussion we do not propose to represent the cloud by a large collection of voxels. While such an approach is quite effective in many domains (such as finite-element analysis), we do not focus on such discretizations for the current analysis because it would necessitate the estimation of a large number of parameters (say the concentration of the chemical in each volume element). The concentration is also likely to be highly correlated between adjacent cells leading to substantial redundancy in the volumetric parameterization. The representational scheme we develop here is not unrelated to these discretization schemes, but attempts to describe the cloud using many fewer parameters.

We propose to represent the cloud by way of a function that can be evaluated at a point in space/time to give a concentration of the chemical of interest. In addition, to simplify the current exposition and visualization, we assume that the cloud is *two-dimensional*, though nothing in this text relies essentially on that fact. By assuming a two dimensional cloud we are essentially modeling the cloud density at head height across a certain region. As will be seen later, this is a reasonable assumption since modeling programs that deal with the effect of CBRN releases typically report concentrations at ground level. Given this we seek a function that, provided with a point in a two-dimensional space, returns the concentration of a chemical of interest at that point. Of course, one could also consider a vector of such functions, each of which returns the concentration of a different chemical compound, but in the current text we choose to focus on a single chemical compound.

As a purely expository example to illustrate the type of representations we seek, one might consider assuming that the cloud of interest has a maximum concentration  $c$  at a point  $x, y$ , and then decays linearly to 0 at some distance  $r$ . This is intentionally a quite rough representation whose only role is to be simple to describe and trivial to improve on later in this text. Plotting the concentration of such a cloud in  $x, y$  space would reveal a

distribution whose curves of constant concentration, or level sets, are circles and whose cross section through  $x, y$  is a cone. There are many such possible clouds and we *parameterize* them by way of a vector  $\alpha_c$  as in

$$\alpha_c = \begin{pmatrix} \mathbf{x} \\ r \\ c \end{pmatrix} \quad (1)$$

where  $\mathbf{x} = \begin{pmatrix} x \\ y \end{pmatrix}$  is the center of the cone and  $r$  and  $c$  are defined as described above. In more precise terms, we have a function  $h_c(\alpha_c, \mathbf{x}_i)$  which gives the concentration of the chemical of interest at a point  $\mathbf{x}_i = \begin{pmatrix} x_i \\ y_i \end{pmatrix}$  by way of the following function

$$h_c(\alpha_c, \mathbf{x}_i) = \begin{cases} c \left(1 - \frac{\|\mathbf{x} - \mathbf{x}_i\|_2}{r}\right) & \text{if } \|\mathbf{x} - \mathbf{x}_i\|_2 < r \\ 0 & \text{otherwise.} \end{cases} \quad (2)$$

Note that the function  $h_c(\alpha_c, \mathbf{x}_i)$  is not an actual cloud model that we will use, but rather an illustration of how such a model can be defined. In addition to modeling the chemical concentration in our two dimensional space, we also need to model the process by which this chemical concentration is measured. To this end the function  $h_c(\alpha_c, \mathbf{x}_i)$  can also be viewed as a measurement function for a point sensor. Taking this view we have introduced another concept, namely a *sensor parameter space*  $\mathbf{x}_i$ . In a scenario with a set of  $N$  such point sensors we will denote them by  $X = \{\mathbf{x}_1, \dots, \mathbf{x}_N\}$ . In Equation (2) we simply return the concentration at the given point, but one can imagine much more complicated measurement processes and we will discuss an example of one later.

While Equation (2) provides a model for a point sensor, the actual reported measurements come from point sensors observing a real cloud. Suppose we are provided a set of such measurements  $Z = \{z_1, \dots, z_N\}$  generated from the collection of sensors at positions  $X$ . Each  $z_i$  is then defined by

$$z_i = h_t(\mathbf{x}_i) + v_i \quad (3)$$

where  $h_t(\mathbf{x}_i)$  is the true cloud concentration at the point  $\mathbf{x}_i$  and  $v_i \in N(0, r_i)$  is the sensor noise. Given a set of these measurements, and assuming that  $h_c(\alpha_c, \mathbf{x}_i)$  can well model  $h_t(\mathbf{x}_i)$ , one can then define a non-linear least squares problem to estimate the unknown  $\alpha_c$  as

$$o_c(X, Z) = \min_{\alpha_c} \sum_{i=1}^N \|h_c(\alpha_c, \mathbf{x}_i) - z_i\|^2. \quad (4)$$

Even better, if the sensors also provide their measurement (co)variances  $R = \{r_0, \dots, r_N\}$ , then one can define the weighted non-linear least squares problem

$$o_c(X, Z, R) = \min_{\alpha_c} \sum_{i=1}^N (h_c(\alpha_c, \mathbf{x}_i) - z_i)^T r_i^{-1} (h_c(\alpha_c, \mathbf{x}_i) - z_i). \quad (5)$$

Even though our in current discussion  $h_c$ ,  $z_i$  and  $r_i$  as described above are all scalar valued, we use the more general vector valued notation in Equation (5) to emphasize that this model can describe multi-dimensional values as well. We also make two observations. First, we classically solve equations of the form of Equation (5) by way of methods such as the Levenberg-Marquardt algorithm. Second, when one solves such problems by way of the Levenberg-Marquardt algorithm one can also compute an approximation of the uncertainty of the parameter estimate  $\tilde{\alpha}_c = o_c(X, Z, R)$  in the form of a 4 x 4 covariance matrix  $P_{\tilde{\alpha}_c}$ .

At this point we have a well defined non-linear least squares parameter estimation problem for the parameter vector  $\alpha_c$ . Of course, the actual solution of such a problem is far from trivial. Such problems may have many

local minima, and one often requires a good initial guess to find the particular solution of interest. While we note the presence of such issues, we leave their resolution aside for the moment and focus our efforts on appropriate modifications to the above problem to make it more realistic.

## 2.2 First Calculations

While the simple and unrealistic cloud model in Equation (2) along with minimization in Equation (5) provides a classic example of non-linear least squares estimation, it also foreshadows several important ideas of our work. First, once one enters the realm of nonlinear estimation, one then has great freedom in their choice of parameterizations. For example, the parameterization  $\alpha_c = [\mathbf{x}, r, c]^T$  was chosen arbitrarily. One could just as easily have chosen  $\alpha_{c2} = [\rho, \theta, r, c]^T$  where  $\rho$  and  $\theta$  give the center of the disk in cylindrical coordinates and  $r, c$  are defined as before. Of course, not all parameterizations are created equal. For example, the latter parameterization is singular at  $\rho = 0$  in the sense that  $\theta$  is not defined at that point.

Let us now *change our toy model* into one which is slightly more realistic and perhaps worthy of some preliminary calculations. We start by considering the diffusion equation since the chemical clouds we are interested in tracking undergo a diffusion process that will significantly effect their final shape and distribution. While not intended to be an introduction to partial differential equations for diffusion processes, as motivation we start with the standard diffusion equation

$$\frac{\partial \phi(\mathbf{p}, t)}{\partial t} = \nabla[w(\phi, t)\nabla\phi(\mathbf{p}, t)] \quad (6)$$

which for the case of uniform diffusion where the diffusion coefficient  $w(\phi, t) = w$  for all densities  $\phi$  and times  $t$  the above reduces to

$$\frac{\partial \phi(\mathbf{p}, t)}{\partial t} = w\nabla^2\phi(\mathbf{p}, t) \quad (7)$$

The parameter  $t$  represents dispersion time and  $\mathbf{p} \in \mathbb{R}^n$  position. A fundamental solution (and Green's function) for Equation (7) can be obtained by applying the equation to a point emission source, with initial conditions  $\phi(\mathbf{p}, 0) = \delta(\mathbf{p})$ . Equation (8) describes the distribution of such a cloud originating from a point source in a windless environment.<sup>19,20</sup>

$$\phi(\mathbf{p}, t) = \frac{1}{(4\pi wt)^{n/2}} \exp\left(-\frac{\mathbf{p}^T \mathbf{p}}{4wt}\right) \quad (8)$$

Note that Equation (8) is a special case of a Gaussian! In a striking example of the beauty of mathematics, if we relax our assumptions to include constant wind and scaling effects we are left with a full multi-variate Gaussian describing the cloud density. Given this observation we can replace our toy  $h_c(\alpha_c, \mathbf{x}_i)$  with a slightly more realistic cloud model based on Gaussians.

## 2.3 Point Sensors

Following the same pattern that we used above we can define a new cloud parameterization  $\alpha_g$  to replace our toy  $\alpha_c$  parameterization. In order to account for scaling and rotation we relax the definition in Equation (8) to be a general Gaussian mass density function parameterized by

$$\alpha_g = \begin{pmatrix} \mathbf{x} \\ e \\ f \\ g \\ c \end{pmatrix}, \quad (9)$$

where  $\mathbf{x}$  defines the center of mass of the distribution,  $e$ ,  $f$ , and  $g$  give the size and shape of the distribution's elliptical level sets, and  $c$  is a scaling factor for the total mass of chemical in the cloud. Given this new parameterization one can define a point measurement function  $h_g(\boldsymbol{\alpha}_g, \mathbf{x}_i)$  motivated by the diffusion equation as

$$h_g(\boldsymbol{\alpha}_g, \mathbf{x}_i) = c \frac{1}{2\pi\sqrt{\|\Sigma\|}} \exp\left(-\frac{1}{2}(\mathbf{x} - \mathbf{x}_i)^T \Sigma^{-1}(\mathbf{x} - \mathbf{x}_i)\right). \quad (10)$$

where  $\Sigma = \begin{pmatrix} e & f \\ f & g \end{pmatrix}^*$ .

Similar to the setting discussed earlier, if we are provided with a set of measurements  $z_i$  as defined in Equation (3) then we can again define a weighted non-linear least squares problem by

$$o_g(X, Z, R) = \min_{\boldsymbol{\alpha}_g} \sum_{i=1}^N (h_g(\boldsymbol{\alpha}_g, \mathbf{x}_i) - z_i)^T r_i^{-1} (h_g(\boldsymbol{\alpha}_g, \mathbf{x}_i) - z_i), \quad (11)$$

a solution of which by the Levenberg-Marquardt algorithm leads to an estimate of  $\boldsymbol{\alpha}_g$  we will call  $\tilde{\boldsymbol{\alpha}}_g$  as well as its associated 6 x 6 covariance matrix  $P_{\tilde{\boldsymbol{\alpha}}_g}$ .

To avoid confusion, let us reiterate that the calculations we discuss here use Gaussian distributions in two ways:

First, we use them as a *Gaussian mass density function* to represent how gas is distributed in a cloud as defined by Equation (9). Note that in the parameterization  $\boldsymbol{\alpha}_g$  the parameter  $c$  differentiates the Gaussian mass density function parameterized by Equation (9) from a Gaussian probability density function (PDF). An integral of a PDF from  $-\infty$  to  $\infty$  is, by definition, 1, but there is no reason to assume that the mass density of our cloud satisfies this constraint. Accordingly, we scale our Gaussian mass density by  $c$  to allow its total mass to vary.

Second, we use a *Gaussian probability density function* to represent the statistics of the parameters of our chosen model using  $\tilde{\boldsymbol{\alpha}}_g$  as the mean, and  $P_{\tilde{\boldsymbol{\alpha}}_g}$  as the covariance.

So we assume that the concentration of chemicals in the cloud follows a Gaussian mass distribution, just as predicted by the diffusion equation. We also assume the parameters that we estimate—by way of taking measurements and then solve for using a weighted non-linear least squares formulation—have errors associated with them. For example, perhaps there are errors with the placement of a cloud's center of mass  $\mathbf{x}$ . We assume that these errors are well represented by a Gaussian probability distribution, so a covariance matrix (such as  $P_{\tilde{\boldsymbol{\alpha}}_g}$ ) completely characterizes them.

The gentle reader may decide whether it is a beautiful or confusing mathematical happenstance that we assume the parameters of our Gaussian mass density are described by a Gaussian probability density function. While we may relax these Gaussian assumptions in future work, we will endeavor to carefully distinguish between the two uses here.

## 2.4 Stand-Off Sensors

In the previous sections we have focused on sensors which sample the chemical mass density function at a single point. In this section we extend our model by examining an additional sensor type. Namely, we define a simple form of *stand-off* sensor which measures the integral of the chemical mass density along a line. Let  $Y = \{\mathbf{y}_1, \dots, \mathbf{y}_N, \dot{\mathbf{y}}_1, \dots, \dot{\mathbf{y}}_M\}$  define the positions  $\mathbf{y}_i$  and directions  $\dot{\mathbf{y}}_i$  of  $M$  stand-off sensors. Specifically, given a ray  $l(\mathbf{y}_j, \dot{\mathbf{y}}_j, t) \in \mathbb{R}^2$  defined by

$$l(\mathbf{y}_j, \dot{\mathbf{y}}_j, t) = \mathbf{y}_j + t\dot{\mathbf{y}}_j \quad (12)$$

---

\*We admit that  $\Sigma$  looks much like a covariance matrix, but thinking in these terms can lead one astray.  $e$ ,  $f$ , and  $g$  parameterize the shape of the cloud and have nothing to do with probability densities!

one can define a measurement function  $h_l$  as

$$h_l(\boldsymbol{\alpha}_g, \mathbf{y}_j, \dot{\mathbf{y}}_j) = \int_0^\infty h_g(\boldsymbol{\alpha}_g, l(\mathbf{y}_j, \dot{\mathbf{y}}_j, t)) dt. \quad (13)$$

To evaluate the above integral one can use any of a myriad of discretization schemes. As  $h_g$  has an exponential decay one can even evaluate the integral efficiently to within a given error bound (i.e. using just a few evaluations of  $h_g$ ) since it is *almost* compactly supported. Given a measurements for  $M$  stand-off sensors  $Q = \{q_1, \dots, q_M\}$ , with noise levels  $S = \{s_1, \dots, s_M\}$  we can define global sensor and measurement sets  $A = X \cup Y$ ,  $B = Z \cup Q$ , and  $C = R \cup S$ . Using  $A$ ,  $B$ , and  $C$  we can again solve a non-linear least squares problem

$$o_b(A, B, C) = \min_{\boldsymbol{\alpha}_g} \sum_{i=1}^N (h_g(\boldsymbol{\alpha}_g, \mathbf{x}_i) - z_i)^T r_i^{-1} (h_g(\boldsymbol{\alpha}_g, \mathbf{x}_i) - z_i) + \sum_{j=1}^M (h_l(\boldsymbol{\alpha}_g, \mathbf{y}_j, \dot{\mathbf{y}}_j) - q_j)^T s_j^{-1} (h_l(\boldsymbol{\alpha}_g, \mathbf{y}_j, \dot{\mathbf{y}}_j) - q_j), \quad (14)$$

where we have  $N$  point sensor measurements,  $M$  stand-off measurements, and the measurements  $q_j$  with noise  $u_j \in N(0, s_j)$  are defined as

$$q_j = \int_0^\infty h_t(l(\mathbf{y}_j, \dot{\mathbf{y}}_j, t)) dt + u_j \quad (15)$$

The measurement function  $h_l(\boldsymbol{\alpha}_g, \mathbf{y}_j, \dot{\mathbf{y}}_j)$  can be viewed as a weighted sum of a number of  $h_g$  sensors that lie along the ray defined by the sensor; in fact, that is also how it may be implemented. On the other hand, it can also be viewed as a simple model for a stand-off sensor. In particular, one can think of it as a sensor placed at  $\mathbf{y}_j$  which is pointed in the direction  $\dot{\mathbf{y}}_j$ . Figure 1 gives a picture of such a sensor.

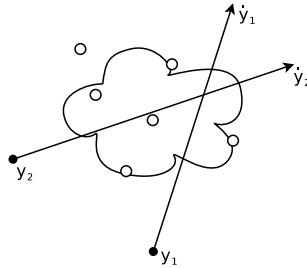


Figure 1. A canonical scenario which mixes stand-off and point sensors. The point sensors defined by the measurement function  $h_g$  are marked with an empty circle. The stand-off sensors with measurement function  $h_l$  are marked with a solid circle and a ray representing the space over which they integrate. Note that the stand-off sensors in this model can be defined in terms of a weighted sum of point sensors along the ray defined by the stand-off sensor. The parameter  $\mathbf{y}_1$  represents the location of the first stand-off sensor and  $\dot{\mathbf{y}}_1$  its direction. The same hold true for  $\mathbf{y}_2$  and  $\dot{\mathbf{y}}_2$  which define the second stand-off sensor.

Now, consider the scenario shown in Figure 2 with many more line sensors. In this scenario we happen to use several  $h_l$  sensors with the additional constraint that they all share a common origin  $\mathbf{y}_j$ . Note, that we can now solve the exact same non-linear least squares problem, as in Equation (14), to estimate  $\boldsymbol{\alpha}_g$ . The reason we choose such a configuration for standoff sensors is to provide a rough model of an *imaging sensor*. In particular, one can choose a line/plane, as in Figure 2, and create an “image” where each pixel of the image is set to have the value of the  $h_l$  sensor that passes through that pixel.

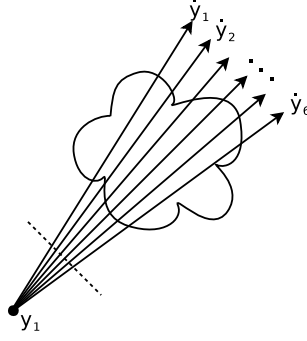


Figure 2. This figure shows a special set of line sensors. In particular, they all happen to share a single origin  $\mathbf{y}_1$ , but point in different directions  $\hat{\mathbf{y}}_1, \dots, \hat{\mathbf{y}}_6$ . This roughly models an *imaging sensor*. In other words, one can imagine selecting an image plane (shown here as a dotted line) and setting the value of each pixel in that plane to the value of the line sensor that passes through that pixel.

Note, we don't actually use the fact that there is an intuitive connection between imaging sensors and the sensor configuration shown in Figure 2 when we solve Equation (14). But it is helpful to keep this connection in mind when we consider modeling imaging sensors.

## 2.5 Towards the Representation of Realistically Complicated Clouds

The discussion above addresses the case where the underlying cloud follows a single Gaussian dispersion pattern. But suppose we are interested in modeling clouds that are potentially much more complicated than can be adequately described by one Gaussian. In order to represent these more complicated clouds we will need many more parameters. Rather than discarding the usefulness of the Gaussian representation, we consider how to combine multiple such representations to accurately represent complicated cloud structures.

Instead of  $\boldsymbol{\alpha}_g \in \mathbb{R}^6$ , consider extending this representation to a sum of  $K$  different Gaussians  $\boldsymbol{\alpha}_G \in \mathbb{R}^{6K}$  parameterized as

$$\boldsymbol{\alpha}_G = \begin{pmatrix} \boldsymbol{\alpha}_g^1 \\ \vdots \\ \boldsymbol{\alpha}_g^K \end{pmatrix} \quad (16)$$

with a corresponding definition of new measurement functions

$$h_G(\boldsymbol{\alpha}_G, \mathbf{x}_i) = \sum_{k=1}^K h_g(\boldsymbol{\alpha}_g^k, \mathbf{x}_i) \quad (17)$$

and

$$h_L(\boldsymbol{\alpha}_G, \mathbf{y}_j, \hat{\mathbf{y}}_j) = \sum_{k=1}^K h_l(\boldsymbol{\alpha}_g^k, \mathbf{y}_j, \hat{\mathbf{y}}_j). \quad (18)$$

Given this new parameterization we can now solve another least-squares minimization problem

$$o_s(A, B, C) = \min_{\boldsymbol{\alpha}_G} \sum_{i=1}^N (h_G(\boldsymbol{\alpha}_G, \mathbf{x}_i) - z_i)^T r_i^{-1} (h_G(\boldsymbol{\alpha}_G, \mathbf{x}_i) - z_i) + \sum_{j=1}^M (h_L(\boldsymbol{\alpha}_G, \mathbf{y}_j, \hat{\mathbf{y}}_j) - q_j)^T s_j^{-1} (h_L(\boldsymbol{\alpha}_G, \mathbf{y}_j, \hat{\mathbf{y}}_j) - q_j). \quad (19)$$

One would obviously be concerned that we now have  $K$  times as many parameters and therefore need more data to estimate them. One might also be concerned about the complexity of the  $\alpha_G$  parameterization. Note however that by choosing  $\alpha_G$  to be a sum of Gaussians we have constrained the optimization space significantly.

Given sufficient data, one can now solve Equation (19) just as one solves Equation (14). In principle there is nothing preventing us from using as many Gaussians as we need, up to the point where we have insufficient measurements to estimate the, perhaps many, parameters in question. The key motivating idea behind using sums of Gaussians is to make the best use possible of a limited number of parameters to describe the cloud while still allowing flexibility in our representation detail.

If a particular scenario only has a few point sensors taking readings then it makes sense to constrain the parameterization of our cloud to use only a few parameters. While it may be tempting to use thousands of parameters to define what looks to be a very realistic cloud, because we have so few measurements we cannot possibly be justified in drawing so precise a picture. Instead of trying to very precisely model a very uncertain cloud density, by using a Gaussian sum approach we attempt to allow for a proper choice of parameterization detail based on the conditions of any particular scenario.

How effective our choice of Gaussian sums proves to be when representing cloud structures in actual scenarios is still an area of continuing research. The following section however gives a description of promising preliminary results describing the effect of applying our parameterization method to 2D chemical cloud simulations.

### 3. SIMULATION RESULTS

In this section we simulate an actual chemical cloud to create ground level concentration reports using a software environment described below called PEGEM. Using this data we can then apply the estimation techniques discussed in the previous section in order to predict the concentration profile of the cloud as well as our uncertainty in that estimate. We note that in this section we provide qualitative results to demonstrate the applicability of Gaussian sums to CBRN estimation problems. For a more quantitative analysis we refer the reader to our companion paper which touches on how to evaluate CBRN estimation methods.<sup>1</sup>

#### 3.1 Using PEGEM to Simulate CBRN Events

A simulation environment called the Post Engagement Ground Effects Model (PEGEM) has been created to model the effects of intercepting warheads containing a variety of different payloads. We have obtained a copy of version 6.1 of the software in order to evaluate the effectiveness of our algorithms on realistically modeled data.

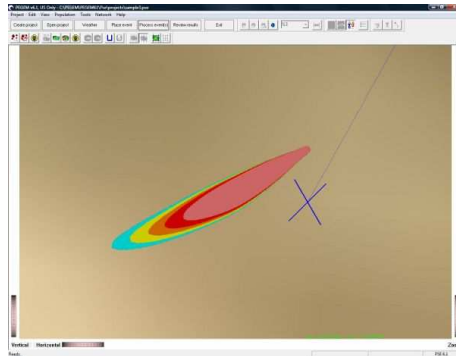


Figure 3. The Post Engagement Ground Effects Model (PEGEM) software allows us to define specific CBRN scenarios and extract the chemical concentrations that arise at any given time. This screen capture shows the inhalation hazard of a generic chemical threat at ground level. The 'X' marks where the missile was when it detonated at an altitude of  $1000m$ .

As is noted in its title, PEGEM is predominantly interested in the concentrations and effects of Chemical, Biological, Radiological, Nuclear, and high yield Explosive (CBRNE) threats at ground level. Figure 3 shows

a typical visualization from PEGEM that displays the inhalation hazard of a generic chemical warhead on a non-intercepted missile. As can be seen from the ‘X’ marked in the figure, the chemical hazard at ground level is not necessarily centered at the detonation point of the missile. This is primarily because of weather effects. The weather used in this scenario is standard historical data shipped with the PEGEM software.

There are many different settings and parameters that effect the simulations in PEGEM. For the figures in this discussion we have chosen a generic setup that models the effects of a chemical warhead carried in a ballistic missile.

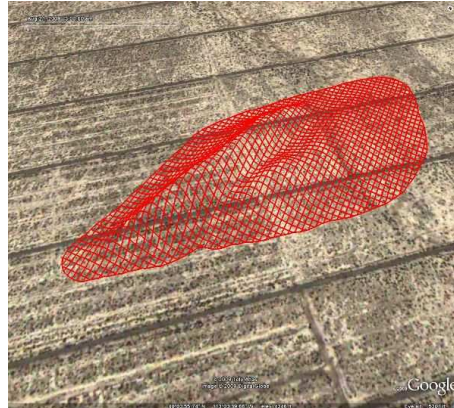


Figure 4. This is the chemical ground concentration calculated by PEGEM 5 minutes into a simulation with a chemical threat. The mesh surface shown represents not the actual vertical height of the cloud, but rather the amount of chemical present over the region where the concentration is above  $0.001 \text{ mg/m}^2$ . Note this is a different cutoff than in Figure 5 where we have drawn the concentration surface out to  $0.02 \text{ mg/m}^2$ .

The warhead itself carries a large amount of chemical agent, and bursts at an altitude of  $1000 \text{ m}$ . Given these settings PEGEM simulates the concentration levels over time as the chemical agent progresses. In the results that follow we attempt to model the 2D cloud at ground level as described by PEGEM 5 minutes after the start of the simulation. This time puts us in the middle of when the ground concentration cloud is growing and expanding. Figure 4 shows the concentration of the cloud calculated by PEGEM at 5 minutes. Note that the height of the surface shown in Figure 4 represents concentration level, not actual vertical height.

### 3.2 Noise Studies with PEGEM Truth Data

Instead of assuming that our model perfectly matches the data by using a Gaussian cloud for ‘truth’ during our simulations, we can use the calculated chemical concentrations from PEGEM as our ‘truth’ cloud. This allows us to observe the behavior of our algorithms on more realistic concentration data.

In the experiments shown here we use only point sensors. These sensors are modeled as measuring the truth (as defined by PEGEM) corrupted by white Gaussian noise as defined by Equation (3). In this setting the function  $h_t(\mathbf{x}_i) : \mathbb{R}^2 \mapsto \mathbb{R}$  defined earlier maps position coordinates to the chemical concentration values computed by PEGEM. Using this sensor model we can create scenarios with both low and high noise values and observe the performance of a Gaussian sum based estimation scheme.

Equation (19) can be solved (approximately) using the Levenberg-Marquardt (LM) method to produce both an estimate of the chemical cloud parameterized by  $\tilde{\alpha}_G = o_s(X, Z, R)$  and also an associated covariance matrix  $P_{\tilde{\alpha}_G}$ . We do this twice for two different noise levels, a low noise level  $r_i = 0.001 \text{ mg/m}^2$ , and a high noise level  $r_i = 0.1 \text{ mg/m}^2$ . The results of both sets of experiments are shown in Figure 5, with the top two pictures illustrating the results of the low noise experiment, and the bottom two pictures the results of the high noise experiment. Both scenarios use a parameterization of  $K = 2$  Gaussians.

In each run there are 50 different point sensors scattered randomly near the PEGEM cloud. Note that the random pattern of sensors is the same within both experiments. The red bars in the figures represent the sensor returns, the yellow surface represents our estimate, the red mesh represents truth, and the dark region on the ground represents the area outside of which we are 95% confident that the chemical concentration is less than  $0.02 \text{ mg}/\text{m}^2$ . This dark region is our way of visualizing the meaning of the covariance matrix  $P_{\tilde{\alpha}_G}$  and what it implies about our knowledge of the chemical concentration. The rationale behind drawing the covariance in this manner comes from the needs of decision makers to evaluate the safety of different areas based on the confidence of our estimate.

For both of the experiments shown in Figure 5, an initial position for the Levenberg-Marquardt solver was obtained by using a weighted average of the sensor returns, where the weight of each sensor’s position was scaled by its concentration height. Given a generic set of Gaussians at this location the Levenberg-Marquardt solver finds a least squares fit of the Gaussian sum to the sensor readings using Equation (19). The covariance around the fitted solution  $P_{\tilde{\alpha}_G}$  is then estimated as

$$P_{\tilde{\alpha}_G} = (\nabla^2 \mathcal{O}_s(X, Z, R))^{-1} \quad (20)$$

from the inverse Hessian over the  $\alpha_G$  parameters at the point  $\tilde{\alpha}_G$ . Note that this linearization is the most optimistic choice we can make since it is also equal to the Cramér-Rao<sup>21</sup> bound at the point  $\tilde{\alpha}_G$ . For the experiments shown in Figure 5 we assumed that all the point sensors were corrupted by an identical amount of independent Gaussian noise, so  $R$  is a set of identical values.

Since we are using PEGEM to generate moderately realistic data, it is important to note that we are still assuming a Gaussian concentration for our estimate. Without any adjustment, our current approach can lead to overly optimistic covariances due to inconsistencies between our model and the actual cloud shape. This can be addressed in the future using a variety of techniques, ranging from inflating the covariance matrix, to more accurately modeling the non-linearities in the problem.

### 3.3 Using Model Extrapolation to Interpolate Between Sensors

Looking at the noise levels in Figure 5 it can be seen that only a very rough guess can be visually inferred from the sensors. To see this look at the left plot for each experiment and try to guess the extents of the cloud without using your knowledge of the truth (shown as the red mesh in the right plots). Because of the large uncertainty in the measurements it is only because we assume that the distribution follows Gaussian diffusion from a small set of source points that the estimator is able to decide on any particular estimate.

By assuming that diffusion has a dominating effect on cloud shape we have made an assumption about how any concentration cloud is likely to be distributed. In fact, it is this assumption that allows us to make predictions about what the chemical concentration will be away from a sensor reading location. Figure 5 seems to indicate that, to a large extent, this assumption makes reasonable interpolations. There are many areas in the above figures that have not been sufficiently sampled by the point sensors. In these regions the only way we can predict anything is by using our model assumptions to fill in the gaps.

It should also be noted that there are many paths forward to improve this interpolation. For example if we were to know the mass of a cloud then a large number of possible solutions could be ruled out. This type of information might come from identifying the source of the cloud, or from a time in the past when the cloud was completely observed. Physical constants such as known total mass are powerful ways to constrain the space of possible solutions and improve the accuracy of our model.

## 4. CONCLUSION

By starting with the assumption that a low-dimensional parametrization of the cloud is preferable, we have derived a common representational space using Gaussian sums. Using sums of Gaussians any cloud density can be represented and the uncertainty of that estimate approximated by a linearization at the solution. These methods are a promising approach to deal with integrating multiple CBRN related sensor returns into a common operating picture. We believe that the key contribution of these approaches is the explicit handling of uncertainty, which allows not only an estimate to be provided to the user, but also a confidence in that estimate.

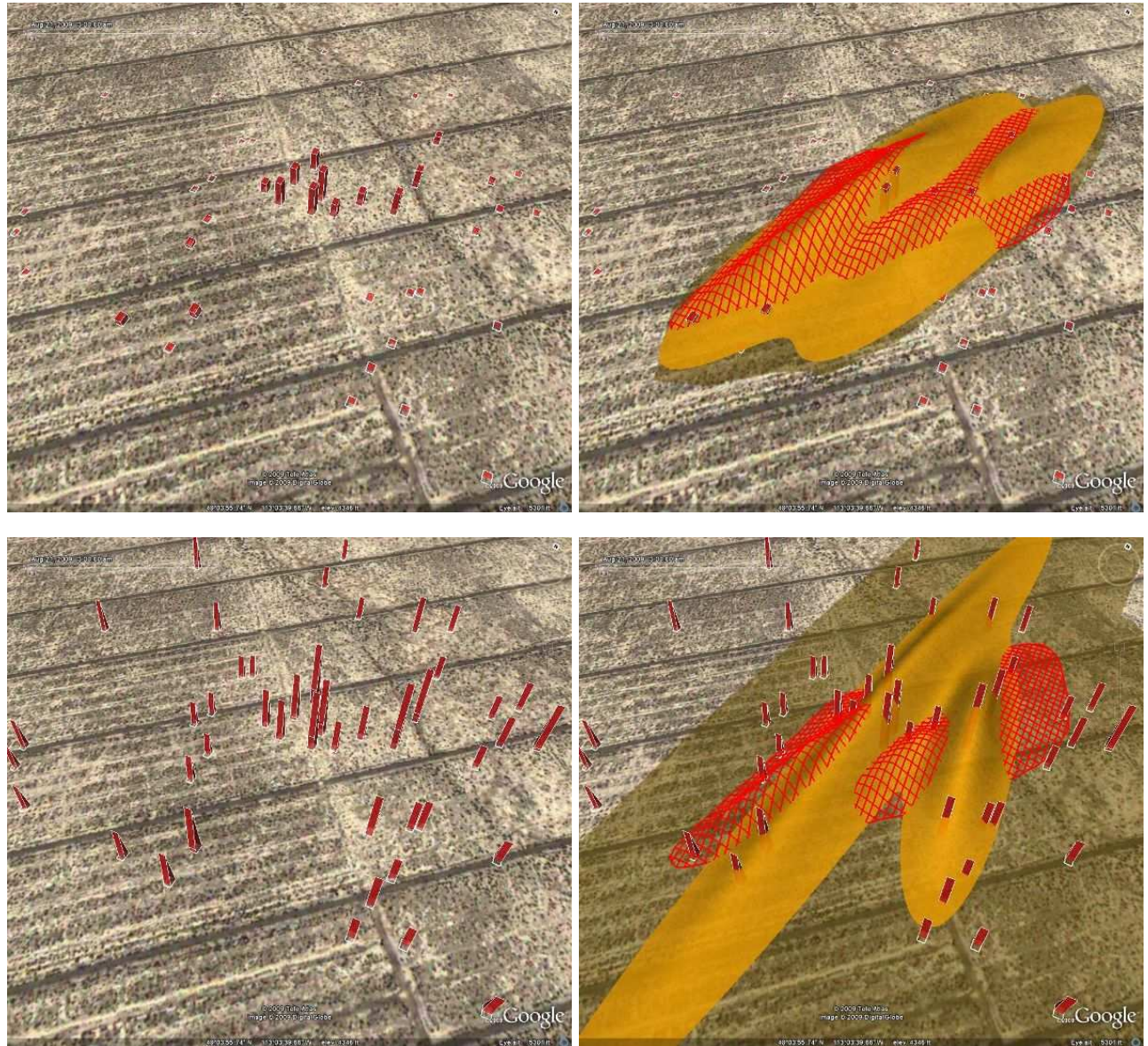


Figure 5. The two rows of figures show the results of fitting a sum of two Gaussians to point sensor measurements taking readings from PEGEM chemical cloud concentration values. The top two figures show results from when the measurements are corrupted with a small amount of noise according to Equation (3) with  $r_i = 0.001 \text{ mg/m}^2$ . The bottom two figures show the change in results when the noise is increased to  $r_i = 0.1 \text{ mg/m}^2$ . The left plots show just the sensor returns, where the length of the angled part of the bar represents the 95% confidence variation, and the middle of the angled portion of the bars represents the reported concentration level. In the right two figures the red mesh represents truth, the orange surface our estimate, and the dark region outside the estimate is the region outside of which we are 95% confident that the chemical concentration is less than  $0.02 \text{ mg/m}^2$ .

## ACKNOWLEDGMENTS

We want to extend a special thanks to Dr. Oliver Drummond, and Dr. Louis Scharf for their very helpful contributions to this work. We are also very grateful to the Defense Threat Reduction Agency (DTRA) for providing us with the opportunity to study this problem under the SBIR Phase I “CBRN Sensor and Sensor Netting Algorithms.” Additionally we would like to thank BAE Systems and the Army SMDC for making PEGEM 6.1 available to us for CBRN event simulations.

## REFERENCES

- [1] S. Lundberg, R. P. and Yosinski, J., “Analysis of CBRN sensor fusion methods,” (2010). To Appear.
- [2] Ewing, R. G., Atkinson, D. A., Eiceman, G. A., and Ewing, G. J., “A critical review of ion mobility spectrometry for the detection of explosives and explosive related compounds,” *Talanta* **54.3**, 515–529 (2001).
- [3] Long, D., [*Raman Spectroscopy*], McGraw-Hill, Texas (1977).
- [4] Bisson, S. E., “Long-wave ir chemical sensing based on difference frequency generation in orientation-patterned gas,” *Journal Applied Physics B: Lasers and Optics* **Volume 85, Numbers 2-3 / November, 2006**, 199–206 (2006).
- [5] Gittins, C. and Marinelli, W., “LWIR multispectral imaging chemical sensor,” in [*Air Monitoring and Detection of Chemical and Biological Agents*], Proceeding of SPIE (November 1998).
- [6] Collins, D. C. and Lee, M. L., “Developments in ion mobility spectrometry-mass spectrometry,” *Anal Bioanal Chem* **372**, 66–73 (2002).
- [7] Kay, S., Xu, C., and Emge, D., “Chemical detection and classification in raman spectra,” in [*Proceedings of SPIE*], Drummond, O. E., ed., **6969**, Signal and Data Processing of Small Targets (March 2008).
- [8] Drummond, O., “Multiple target tracking with multiple frame, probabilistic data association,” in [*Proceedings of SPIE*], Signal and Data Processing of Small Targets (1993).
- [9] Blackman, S. and Popoli, R., [*Design and Analysis of Modern Tracking Systems*], Artech House, Norwood, MA (1999).
- [10] Bar-Shalom, Y., Li, X.-R., and Kirubarajan, T., [*Estimation with Applications to Tracking and Navigation: Theory, Algorithms, and Software*], J. Wiley (2001).
- [11] Bar-Shalom, Y., ed., [*Multitarget-Multisensor Tracking: Applications and Advances*], Artech House, Dedham, MA (1992).
- [12] Bar-Shalom, Y. and Li, X.-R., [*Estimation and Tracking: Principles, Techniques, and Software*], YBS (1998).
- [13] Drummond, O., “Multiple-frame best-hypotheses target tracking with multiple sensors,” in [*Proceedings of SPIE*], Signal and Data Processing of Small Targets (2003).
- [14] Vanslyke, S. J. and Wentzell, P. D., “Real-time principal component analysis using parallel kalman filter networks for peak purity analysis,” *Analytical Chemistry* **63**, 2512–2519 (1991).
- [15] Wentzell, P. D. and Vanslyke, S. J., “Parallel kalman filter networks for kinetic methods of analysis,” *Analytical Chimica Acta* **257**, 172–181 (February 1992).
- [16] Wentzell, P. D. and Vanslyke, S. J., “Parallel kalman filters for peak purity analysis - extensions to nonideal detector response,” *Analytical Chimica Acta* **307**, 459–470 (May 1995).
- [17] Chen, J. C. and Rutan, S. C., “Identification and quantification of overlapped peaks in liquid chromatography with uv diode array detection using an adaptive kalman filter,” *Analytical Chimica Acta* **335**, 1–10 (December 1996).
- [18] Scott, S. M., James, D., and Ali, Z., “Data analysis for electronic nose systems,” *Microchimica Acta* (156), 183–207 (2007).
- [19] Ramasubramanian, S., “The normal distribution,” *Resonance* **Volume 2, Number 7 / July, 1997** (1997).
- [20] Crank, J., [*The mathematics of diffusion*], Oxford University Press, USA (1979).
- [21] Kay, S., [*Fundamentals of statistical signal processing: estimation theory*] (1993).