Tracking of Multiple Contaminant Clouds
François Septier, Avishy Carmi, Simon Godsill

To cite this version:

HAL Id: hal-00566637
https://hal-imt.archives-ouvertes.fr/hal-00566637
Submitted on 15 Apr 2013

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L’archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d’enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.
Abstract — In this paper, we address the problem of detection and tracking of multiple contaminant clouds. We develop a stochastic extension of the Gaussian puff model to characterize evolution of the average atmospheric pollutant concentration. To perform the sequential inference on this difficult problem, we propose a Markov Chain Monte Carlo (MCMC)-based Particle algorithm. Numerical simulations illustrate the ability of the algorithm to detect and track multiple contaminant clouds.

Keywords: Bayesian Inference, sequential MCMC, Tracking, contaminant cloud, environmental imaging.

1 Introduction

Nowadays, the threat of pollution due to the release, either accidentally or deliberately, of Chemical, Biological, Radiological or Nuclear (CBRN) agents is high. Indeed, many rogue nations and terror groups seek to employ asymmetric warfare and some groups will be attracted by the use of chemical weapons to achieve major impact. As a consequence, rapid detection and early response to a release of a CBRN agent could dramatically reduce the extent of human exposure. The capability to monitor and track contaminant clouds is therefore a problem of great importance.

Contour tracking for a single source emission is addressed in [1]. In this work, the tracking of the shape of contaminant cloud is performed by estimating points on the contour boundary using several local particle filters [2]. The sequence of these points generates the contaminant boundary of particular level of concentration.

In this paper, we propose an inference algorithm within a Bayesian framework to track directly the contaminant concentration, instead of some contaminant boundary as in [1], thus providing more information about the actual situation. Based on predictive models for the mean transport and turbulent diffusion of the contaminant through the atmosphere, we develop a stochastic extension of the well-known Gaussian puff model to characterize the evolution of these contaminant clouds. The proposed algorithm to identify and track these multiple contaminant clouds is based on a sequential Monte-Carlo Markov Chain (MCMC) mechanism which approximates the filtering distribution of interest.

The paper is organized as follows. The problem statement is discussed in Section 2. In Section 3, the dynamical model is described in a Bayesian framework. The inference algorithm is developed in Section 4. Simulation results are provided in Section 5. Finally, conclusions are given in Section 6.

2 Problem Statement

2.1 Atmospheric Dispersion Models

If any trace gas or pollutant is emitted into the atmosphere, it will undergo various processes that all together eventually lead to a concentration distribution somewhere downwind of the source. An idealized picture of the concentration distribution downwind of a source is that of a plume. The relatively simple and fast plume models yield a “statistical plume”, i.e. a concentration distribution that represents an ensemble average over a number of individual instantaneous plumes. The Gaussian plume model was originated from the work by Sutton [3] and consists of a single equation obtained by solving the Fickian diffusion equation for homogeneous turbulence and uniform wind profile. For a considerable part of the early twentieth century, therefore, dispersion modeling was essentially plume modeling.

A more sophisticated method of describing the evolution of pollutant plume is the puff model where a series of puffs is emitted from the source and their growth is modeled as they travel downwind. The simplest puff models assume a constant (e.g. Gaussian) distribution while more sophisticated models allow puff distortion, puff splitting or combine puffs with particles [4–6].

Finally, the most advanced dispersion model is the so-called Lagrangian particle dispersion model. Here thousands of individual particles (fluid elements) are
traced and their distribution yields an estimate for the concentration field [7]. Figure 1 illustrates these three types of atmospheric dispersion models.

![Figure 1: Different types of atmospheric dispersion models: plume models (top), puff models (middle) and particle models (bottom).](image)

The plume models suffer from several deficiencies. First of all, plume models always yield mean concentration distributions and thus cannot take into account variable (non-stationary) conditions. As an example, the evacuation plan should be based on the actually concentration field [7]. Figure 1 illustrates these three types of atmospheric dispersion models. Therefore, the evacuation plan should be based on the actually concentration field [7]. Figure 1 illustrates these three types of atmospheric dispersion models.

2.2 The Gaussian puff model
In the Gaussian puff model, the average concentration of \( N_{r,k} \) instantaneous releases is given at time \( t_k \) by:

\[
\tilde{c}_{x,y,k} = \sum_{i=1}^{N_{r,k}} \frac{w_{k,i}}{2\pi|\Sigma_{k,i}|^{1/2}} e^{-\frac{1}{2}(x - \mu_{k,i})^T \Sigma_{k,i}^{-1} (x - \mu_{k,i})} (1)
\]

with \( w_i \) the source term mass of the \( i^{th} \) release. In this model, the weights, the mean vector and covariance matrix of each Gaussian shape evolve over time as follows, if \( t_k \geq T_i \):

\[
w_{k,i} = W_i (2)
\]

\[
\mu_{k,i} = \left[ \begin{array}{c} X_i \\ Y_i \end{array} \right] + \nu(t_k - T_i) (3)
\]

\[
\Sigma_{k,i} = \Sigma_{0,i} + 2D(t_k - T_i) (4)
\]

where \( T_i, (X_i, Y_i), \Sigma_{0,i} \) are respectively the release time, the location and the initial covariance matrix of the \( i^{th} \) source term. \( \nu \) is the wind velocity vector and \( D \) is the diffusion matrix. We can remark that the average concentration obtained with this Gaussian puff model is a deterministic function of the source term parameters \( \{X_i, Y_i, w_i, T_i, \Sigma_{0,i}\}_{i=1}^{N_{r,k}} \). Due to the stochastic nature of the atmosphere, we propose a random extension of this model.

2.3 Stochastic Gaussian Puff Model
In order to take into account the random nature of the atmospheric dispersion and thus to provide more flexible and realistic model, we propose a stochastic extension of the Gaussian puff model. In this model, the average concentration of \( N_{r,k} \) instantaneous releases is given at time \( t_k \) by the same equation than in the Gaussian puff model, Eq. (1). However, the parameters of each Gaussian shape evolve independently over time as follows:

\[
p_u(w_{k,i}|w_{k-1,i}) = \phi(w_{k,i}|w_{k-1,i}, \sigma^2_w) (5)
\]

\[
p_u(\mu_{k,i}|\mu_{k-1,i}) = \mathcal{N}(\mu_{k,i}|\mu_{k-1,i} + \nu \tau, \sigma^2 I_2) (6)
\]

\[
p_u(\Sigma_{k,i}|\Sigma_{k-1,i}) = \mathcal{W}\left(\Sigma_{k-1,i} + 2D\tau, \nu, \sigma^2 \right) (7)
\]

where \( \phi(\cdot|\nu, \sigma^2) \) is the truncated normal distribution with mean \( \nu \) and variance \( \sigma^2 \) defined on \([0, +\infty)\) to ensure the positivity of the weight variable. The mean vector transition probability is a random walk with a linear drift depending on the wind velocity \( \nu \) and the covariance matrix is assumed to be a Wishart random matrix with a scaling matrix \( D\tau \) and a number of degrees of freedom \( n \). \( \tau = t_k - t_{k-1} \) corresponds to the sampling interval between \( t_k \) and \( t_{k-1} \). We should remark that a similar idea was proposed in [6]. In this work, authors have proposed to associate a Gaussian puff to each particle which evolves randomly over time by following a Lagrangian dispersion model.

2.4 Dispersion Model Uncertainty
Stochastic uncertainty arising from the turbulent nature of the atmosphere gives rise naturally to random concentration fluctuations in hazardous gas releases. The real concentration is thus generally assumed to be the sum of the average concentration \( \tilde{c}_{x,y,k} \) and a noise term:

\[
c_{x,y,k} = \tilde{c}_{x,y,k} + \omega_{x,y,k} (8)
\]

In order to ensure its positivity, the concentration at a given location is described by a clipped normal distri-
bution between \([0, +\infty]\) \cite{8}:

\[
p(c_{x,y,k}|\bar{c}_{x,y,k}) =
\begin{cases} 
0 & \text{if } c_{x,y,k} < 0 \\
\frac{1}{2} \left(1 - \text{erf} \left(\frac{c_{x,y,k}}{\sqrt{2\sigma^2(c_{x,y,k})}}\right)\right) & \text{if } c_{x,y,k} = 0 \\
\mathcal{N}(c_{x,y,k}|\bar{c}_{x,y,k}, V(c_{x,y,k})) & \text{if } c_{x,y,k} > 0
\end{cases}
\]

where \(\mathcal{N}(c_{x,y,k}|\bar{c}_{x,y,k}, V(c_{x,y,k}))\) is the Gaussian distribution with mean \(\bar{c}_{x,y,k}\) and variance \(V(c_{x,y,k})\) (with \(V(\cdot)\) a known function). The delta function at zero corresponds to the intermittency (periods of zero concentration) in observed concentrations.

### 2.5 Observation Model

For chemical, biological or radiological source term estimation, several sensors allow to measure the concentration in the atmosphere like the LCAD (Lightweight Chemical Agent Detector) and the MCAD (Mobile Chemical Agent Detector) but one of the most data rich sensors currently available is the Light Distance And Ranging (LIDAR) sensor. Clouds of dispersing biological material may fluoresce if illuminated by ultraviolet (UV) light. A LIDAR system can emit pulses of UV light and measure the spectral composition of returning visible fluoresced light as a function of time. The time measurement determines the range at which the fluoresced light was emitted. The fluorescence at a particular point in space is a function of the radial distance to the point \(r\). By considering a single wavelength range, the measured fluorescence is defined as follows:

\[
f_{r,\theta_k} = \frac{E}{r^2} e^{-\chi_{\text{out}} r} e^{-\chi_{\text{in}} r} F_{c_{r,\theta_k}} + c_{r,\theta_k} \tag{10}
\]

where \(\chi_{\text{in}}, \chi_{\text{out}}\) are the attenuation coefficients in air, \(F\) is the fluorescence coefficient, \(c\) is the actual concentration, \(E\) is the outgoing laser pulse energy and the measurement noise is \(c_{r,\theta_k} \sim \mathcal{N}(0, \sigma^2)\). At each time step \(t_k\), measurements of the concentration are obtained at a specified bearing \(\theta_k\) over different ranges \(r\). Let us denote by \(\mathbf{f}_k\) the set of measurements such as \(\mathbf{f}_k = \{f_{r,\theta_k}\}_{r=R_1}^{R_{max}}\). Due to the space dependence of the observation noise, the likelihood of the average concentration \(\bar{c}_k\) can be written as:

\[
p(\mathbf{f}_k|\bar{c}_k) = \prod_{r=R_1}^{R_{max}} p(f_{r,\theta_k}|\bar{c}_{r,\theta_k}) \tag{11}
\]

with

\[
p(f_{r,\theta_k}|\bar{c}_{r,\theta_k}) = \int p(f_{r,\theta_k}|c_{r,\theta_k}) p(c_{r,\theta_k}|\bar{c}_{r,\theta_k}) \, dc_{r,\theta_k} \tag{12}
\]

The closed form expression of this likelihood function is given in Eq. (37) (see Appendix A). In this work, we consider thresholded LIDAR observations, \(\tilde{f}_{r,\theta_k}\), that returns only 1 or 0, i.e. \(\tilde{f}_{r,\theta_k} = \{0, 1\}\). In this case, the likelihood function \(p(\bar{f}_k|\bar{c}_k)\) is given by:

\[
p(\bar{f}_k|\bar{c}_k) = \prod_{r=R_1}^{R_{max}} p(\tilde{f}_{r,\theta_k}|\bar{c}_{r,\theta_k}) \tag{13}
\]

with

\[
p(\tilde{f}_{r,\theta_k}|\bar{c}_{r,\theta_k}) = \begin{cases} 
0 & \text{if } f_{r,\theta_k} \geq \lambda\bar{c}_{r,\theta_k} \\
1 & \text{if } f_{r,\theta_k} < \lambda\bar{c}_{r,\theta_k}
\end{cases}
\]

where \(\lambda\) corresponds to the threshold and

\[
p(f_{r,\theta_k} < \lambda\bar{c}_{r,\theta_k}) = \int_{-\infty}^{\lambda} p(f_{r,\theta_k}|\bar{c}_{r,\theta_k}) \, df_{r,\theta_k} = 1 - p(f_{r,\theta_k} \geq \lambda\bar{c}_{r,\theta_k}) \tag{14}
\]

Unfortunately, this integrand can not be computed in closed form owing to the presence of the error function depending on \(f_{r,\theta_k}\) in Eq. (37). So, we propose to use the following approximation:

\[
p(f_{r,\theta_k} < \lambda\bar{c}_{r,\theta_k}) \approx A \mathcal{N}(f_{r,\theta_k}|0, \sigma^2) + (1-A)\mathcal{N}(f_{r,\theta_k}|\alpha F_{c_{r,\theta_k}}, \sigma^2 + \alpha^2 F^2 V(\bar{c}_{r,\theta_k})) \tag{16}
\]

with \(A = \frac{1}{2} \left[1 - \text{erf} \left(\frac{\bar{c}_{r,\theta_k}}{\sqrt{2\sigma^2}}\right)\right]\). By plugging this approximation into Eq. (15), we obtain:

\[
p(\tilde{f}_{r,\theta_k} = 0|\bar{c}_{r,\theta_k}) \approx \frac{\lambda}{2} \left[1 + \text{erf} \left(\frac{\lambda}{\sqrt{2\sigma^2}}\right)\right] + \frac{\lambda - \lambda}{2} \left[1 + \text{erf} \left(\frac{\lambda - \lambda}{\sqrt{2(\sigma^2 + \alpha^2 F^2 V(\bar{c}_{r,\theta_k}))}}\right)\right] \tag{17}
\]

By assuming \(L\) possible bearings for the LIDAR, a complete scan of the region is thus obtained by taking into account set of observations from \(L\) consecutive time steps (i.e. \(\tilde{f}_{r,\theta_k}\) for \(r=L+1, r+1, \ldots, r_L\)).

### 3 Bayesian Modeling

With both the dispersion model and observation model described in the previous section, we can now formulate the multiple contaminant cloud tracking problem in a Bayesian context. In this section, we will complete the model by introducing the dynamical model.

The average concentration \(\bar{c}_{x,y,k}\) in Eq. (1) is completely determined by the dynamic states \(\{\omega_{k,i}, \mu_{k,i}, \Sigma_{k,i}\}_{i=1}^{N_{r,k}}\). In a Bayesian framework, the aim is thus to compute the posterior distribution \(p(\{w_{k,i}, \mu_{k,i}, \Sigma_{k,i}\}_{i=1}^{N_{r,k}}|\tilde{f}_k\})\) where \(\tilde{f}_k\) is the observation set from time 1 to \(t_k\). Note that the number of Gaussian shapes used to represent the average concentration in Eq. (1), \(N_{r,k}\), can evolve over time. Of
course there is no reason either theoretically or computationally why a variable dimension quantity such as in this application should not be maintained throughout the model as is routinely done in problems of Bayesian model choice. However, in this work we choose equivalently to formulate this birth and death process of these Gaussian shapes explicitly in terms of $e_k$, a set of existence variables (each element $e_{k,i} \in \{0, 1\}$ model either active or inactive Gaussian shape). In this formulation, the state of interest is thus regarded as fixed dimensionally to formulate this birth and death process of these model choice. However, in this work we choose equivalently to formulate this birth and death process of these Gaussian shapes explicitly in terms of $e_k$, a set of existence variables (each element $e_{k,i} \in \{0, 1\}$ model either active or inactive Gaussian shape). In this formulation, the state of interest is thus regarded as fixed dimensionally to formulate this birth and death process of these Gaussian shapes explicitly in terms of $e_k$, a set of existence variables (each element $e_{k,i} \in \{0, 1\}$ model either active or inactive Gaussian shape). In this formulation, the state of interest is thus regarded as fixed dimensionally to formulate this birth and death process of these Gaussian shapes explicitly in terms of $e_k$, a set of existence variables (each element $e_{k,i} \in \{0, 1\}$ model either active or inactive Gaussian shape). In this formulation, the state of interest is thus regarded as fixed dimensionally to formulate this birth and death process of these Gaussian shapes explicitly in terms of $e_k$, a set of existence variables (each element $e_{k,i} \in \{0, 1\}$ model either active or inactive Gaussian shape). In this formulation, the state of interest is thus regarded as fixed dimensionally to formulate this

\[
\bar{c}_{x,y,k} = \sum_{i=1}^{N_{max}} \frac{e_{k,i} w_{k,i}}{2\pi |\Sigma_{k,i}|^{1/2}} e^{-\frac{1}{2} \left( \frac{x - \mu_{k,i}}{\Sigma_{k,i}} \right)^T \left( \frac{x - \mu_{k,i}}{\Sigma_{k,i}} \right)}
\]

(18)

Let us denote by $s_k = \{e_{k,i}, w_{k,i}, \mu_{k,i}, \Sigma_{k,i} \}_{i=1}^{N_{max}}$ the state of interest. Assuming a Markovian state transition, the standard Bayesian filtering prediction and update steps are given by:

\[
p(s_k | \hat{r}_{1:k}) \propto p(\hat{r}_k | s_k)p(s_k | \hat{r}_{1:k-1})
\]

(19)

with

\[
p(s_k | \hat{r}_{1:k-1}) = \int p(s_k | s_{k-1})p(s_{k-1} | \hat{r}_{1:k-1}) ds_{k-1}
\]

(20)

By assuming that the Gaussian shapes in Eq. (18) are independent of one another, we choose to expand the transition probability distribution of $s_k$ as follows:

\[
p(s_k | s_{k-1}) = \prod_{i=1}^{N_{max}} p(w_{k,i} | w_{k-1,i}, e_{k,i}, e_{k-1,i}) \\
\times p(e_{k,i} | e_{k-1,i})p(\mu_{k,i} | \mu_{k-1,i}, e_{k,i}, e_{k-1,i}) \\
\times p(\Sigma_{k,i} | \Sigma_{k-1,i}, e_{k,i}, e_{k-1,i})
\]

(21)

This transition probability can also be partitioned according to $e_{k,i}$ and $e_{k-1,i}$. Let $\Upsilon_b$ be the set of elements that have $e_{k,i} = 1$ and $e_{k-1,i} = 0$. This corresponds to the birth scenario where new Gaussian shapes become active. $\Upsilon_d$ is the set of elements that have $e_{k,i} = 0$. This is the inactive Gaussian shape set. Let $\Upsilon_a$ be the set of elements that have $e_{k,i} = 1$ and $e_{k-1,i} = 1$. This corresponds to active Gaussian shapes that will be updated between two consecutive time steps. Then we can write

\[
p(s_k | s_{k-1}) = \prod_{i \in \Upsilon_b} p_b(w_{k,i})p_b(\mu_{k,i})p_b(\Sigma_{k,i}) \\
\times \prod_{j \in \Upsilon_a} p_d(w_{k,j})p_d(\mu_{k,j})p_d(\Sigma_{k,j}) \\
\times \prod_{l \in \Upsilon_a} p_u(w_{k,l} | w_{k-1,l})p_u(\mu_{k,l} | \mu_{k-1,l})p_u(\Sigma_{k,l} | \Sigma_{k-1,l})
\]

(22)

We now describe the various dynamical models.

### 3.1 Existence Variables

Each target’s existence variable will be modeled as a discrete Markov chain [9] which is independent of all other states. Moreover, these existence variables are considered to be static during a complete scan and dynamic between two successive complete scans.

\[
p(e_{k,i} | e_{k-1,i}) = \begin{cases} g(e_{k,i} | e_{k-1,i}) & \text{if } k = (n-1)L + 1 \\ \delta(e_{k,i} - e_{k-1,i}) & \text{otherwise} \end{cases}
\]

with $n \in \mathbb{N}$ and

\[
g(e_{k,i} | e_{k-1,i}) = \begin{cases} P_B & \text{if } e_{k,i} = 1 \text{ and } e_{k-1,i} = 0 \\ 1 - P_B & \text{if } e_{k,i} = 0 \text{ and } e_{k-1,i} = 0 \\ P_D & \text{if } e_{k,i} = 0 \text{ and } e_{k-1,i} = 1 \\ 1 - P_D & \text{if } e_{k,i} = 1 \text{ and } e_{k-1,i} = 1 \end{cases}
\]

where $P_B$ and $P_D$ are respectively the birth and death probability.

### 3.2 Birth Case

A birth happens when $e_{k,i} = 1$ and $e_{k-1,i} = 0$. In this case, the probability distribution of the weight variables is defined as:

\[
p_b(w_{k,i}) = \mathcal{U}(0; w_{max})
\]

(23)

with $w_{max}$ is the maximum weight possible for a Gaussian shape. The contaminant could can appear anywhere uniformly in the surveillance area $\mathcal{R}$:

\[
p_b(\mu_{k,i}) = \mathcal{U}_\mathcal{R}
\]

(24)

And finally, the covariance matrix is assumed to be a Wishart random matrix with a scaling matrix $D_b$ and a number of degrees of freedom $n$:

\[
p_b(\Sigma_{k,i}) = \mathcal{W}\left(\frac{D_b}{n}, n\right)
\]

(25)

### 3.3 Inactive Case

An inactive case happens when $e_{k,i} = 0$. In this case, the probability distribution of the variables are defined as follows:

\[
p_d(w_{k,i}) = \delta(w_{k,i})
\]

(26)

\[
p_d(\mu_{k,i}) = \delta(\mu_{k,i})
\]

(27)

\[
p_d(\Sigma_{k,i}) = \delta(\Sigma_{k,i})
\]

(28)

### 3.4 Update Case

The update case happens when $e_{k,i} = 1$ and $e_{k-1,i} = 1$. In this case, parameters of each Gaussian shape will be updated according to the stochastic Gaussian puff model described by Eq. (5)-(7).


4 Bayesian Inference

The filtering distribution of interest is complex and highly nonlinear. Sequential Monte Carlo methods such as particle filters can be used to carry out the inference. However, given the high dimensionality of the state space, it will not be straightforward to design an efficient particle filter implementation.

Traditionally, MCMC is used to draw samples from probability distributions in a non-sequential setting. The advantages of MCMC are that it is generally more effective than particle filters in high-dimensional systems and it is easier to design for complex distributions if it can be used in a sequential fashion. Sequential approaches using MCMC can be found in [10–12]. In [10], a sequential MCMC algorithm was designed to do inference in dynamical models using a series of Metropolis-Hastings-within-Gibbs. A similar idea was applied in [12] for imputing missing data from nonlinear diffusion. In [11], a MCMC-Particles algorithm was proposed using a numerical integration of the predictive density but unfortunately its computational demand can become excessive as the number of particles increases owing to its direct Monte Carlo calculation of the filtering density at each time step. In [13], a MCMC-based particle algorithm was proposed for tracking coordinated targets. In this work, we will design a MCMC-based particle algorithm for tracking multiple contaminant clouds. The approach is distinct from the Resample-Move scheme [14] in particle filter where the MCMC algorithm is used to rejuvenate degenerate samples following importance sampling resampling: our method uses neither resampling or importance sampling.

Instead of the usual filtering distribution \( p(s_{t-1|L} \mid f_{1:L}) \), we will consider the joint distribution of \( \{s_{t-1|L}, s_{t-1|L+1:nL}\} \) corresponding respectively to the state of interest at time \( t-1 \) and state associated to a complete LIDAR scan over all the observation scene (i.e. from \( t-1 \) to \( nL \)):

\[
p(s_{t-1|L+1:nL} \mid f_{1:L}) \propto p(s_{t-1|L} \mid f_{1:L}) (29)
\]

To make the inference from this complex distribution sequentially, we will use a MCMC procedure. Since we do not have the closed form expression of the distribution \( p(s_{t-1|L} \mid f_{1:L}) \), it will be approximated by an empirical distribution based on the particle set:

\[
\tilde{p}(s_{t-1|L} \mid f_{1:L}) = \frac{1}{N_p} \sum_{p=1}^{N_p} \delta(s_{t-1|L} = s_p^L) (30)
\]

By using this approximation, since all the distributions involved in Eq. (29) are known, an appropriate MCMC scheme can be used to draw from \( p(s_{t-1|L+1:nL} \mid f_{1:nL}) \). The converged MCMC output are then extracted to give an empirical approximation of the posterior distribution of interest at time \( nL \), making possible the sequential inference. We call this scheme the MCMC-based particle algorithm.

At the \( n^{th} \) MCMC iteration, the following procedure is performed to obtain samples from \( p(s_{t-1|L+1:nL} \mid f_{1:nL}) \):

1. Make a joint draw for \( s_{t-1|L+1:nL} \) using a Metropolis-Hastings step (Algorithm 1),

2. Refine \( s_{t-1|L+1:nL} \) using a series of Metropolis-Hastings-within-Gibbs steps (Algorithm 2).

Algorithm 1 Joint Proposal

1: Propose \( s_{t-1|L}^* \sim \tilde{p}(s_{t-1|L} \mid f_{1:L}) \)
2: for \( i = 1 \ldots N_{max} \) do
3: for \( k = (n-1) \) to \( nL \) do
4: Propose \( e_{k,i}^* \sim p(e_{k,i} \mid s_{t-1|L}^*, f_{1:L}^*) \)
5: Propose \( w_{k,i}^* \sim p(w_{k,i} \mid e_{k,i}, s_{t-1|L}^*, f_{1:L}^*) \)
6: Propose \( \mu_{k,i}^* \sim p(\mu_{k,i} \mid f_{1:L}^* \mid e_{k,i}, s_{t-1|L}^*) \)
7: Propose \( \Sigma_{k,i}^* \sim p(\Sigma_{k,i} \mid f_{1:L}^* \mid \mu_{k,i}^*, e_{k,i}, s_{t-1|L}^*) \)
8: end for
9: end for
10: With \( s_{t-1|L+1:nL}^m \) calculate the acceptance ratio:
\[
\rho_1 = \min \left( 1, \frac{\prod_{k=(n-1)\rightarrow nL} p(e_{k,i}^*) p(w_{k,i}^*)}{\prod_{k=(n-1)\rightarrow nL} p(e_{k,i} p(w_{k,i})}\right)
\]
11: Sample a uniform random variable \( u \) from \( U(u \mid 0, 1) \) and set \( s_{t-1|L+1:nL}^m = s_{t-1|L+1:nL}^m \) if \( u \leq \rho_1 \), otherwise set \( s_{t-1|L+1:nL}^m = s_{t-1|L+1:nL}^m-1 \).

Algorithm 2 Refinement Step

1: for each Gaussian shape \( i \) do
2: Set \( \{e_{t-1|L}^{m_m}, w_{t-1|L}^{m_m}, \mu_{t-1|L}^{m_m}, \Sigma_{t-1|L}^{m_m}\} = \{e_{t-1|L}^m, w_{t-1|L}^m, \mu_{t-1|L}^m, \Sigma_{t-1|L}^m\} \)
3: for \( k = (n-1) \) to \( nL \) do
4: Propose \( e_{k,i}^* \sim p(e_{k,i} \mid s_{t-1|L}^m, f_{1:L}^*) \)
5: Propose \( w_{k,i}^* \sim p(w_{k,i} \mid e_{k,i}, s_{t-1|L}^m, f_{1:L}^*) \)
6: Propose \( \mu_{k,i}^* \sim p(\mu_{k,i} \mid f_{1:L}^* \mid e_{k,i}, s_{t-1|L}^m) \)
7: Propose \( \Sigma_{k,i}^* \sim p(\Sigma_{k,i} \mid f_{1:L}^* \mid \mu_{k,i}^*, e_{k,i}, s_{t-1|L}^m) \)
8: end for
9: With \( \{e_{t-1|L}^m, w_{t-1|L}^m, \mu_{t-1|L}^m, \Sigma_{t-1|L}^m\} \) calculate the acceptance ratio:
\[
\rho_2 = \min \left( 1, \frac{\prod_{k=(n-1)\rightarrow nL} p(e_{k,i}^*) p(w_{k,i}^*)}{\prod_{k=(n-1)\rightarrow nL} p(e_{k,i}^*) p(w_{k,i}^*)}\right)
\]
10: Sample a uniform random variable \( u \) from \( U(u \mid 0, 1) \). If \( u \leq \rho_2 \), set \( s_{t-1|L+1:nL}^m = s_{t-1|L+1:nL}^m \) otherwise set \( s_{t-1|L+1:nL}^m = s_{t-1|L+1:nL}^m-1 \).
5 Simulation Results

The scenario studied in this section consists of one cloud modeled by 3 Gaussian shapes then at time $t = 150$ s, an other cloud modeled by one Gaussian puff appears in the observation scene. The synthetic data are generated using the stochastic Gaussian puff model described by Eq. (5)-(7) with the model parameters described in Table 1.

The MCMC-Based Particle algorithm is used to track the contaminant clouds. All Gaussian shapes are initialized as inactive in order to allow the algorithm to identify all the Gaussian shapes necessary to model the actual contaminant clouds. The maximum number of Gaussian shapes in the algorithm is set to $N_{\text{max}}=5$. For each complete LIDAR scan, 4000 MCMC iterations are performed with burn-in of 1000 iterations. All 4000 MCMC output are kept as particle approximation to $p(S_{(n-1),n},L|\tilde{F}_{1:n:L})$.

Figure 2 shows both the true average concentration and the estimated average concentration as well as the observation set obtained from a complete LIDAR scan with thresholded fluorescence measurements. The estimated average concentration using the proposed algorithm is obtained by computing the following posterior expectation:

$$\hat{c}_{x,y,k} = \mathbb{E}_{\tilde{s}_{k},\tilde{r}_{k}}[\tilde{c}_{x,y,k}] \approx \frac{1}{N_p} \sum_{i=1}^{N_{\text{max}}} \sum_{p=1}^{N_p} \frac{e_{k,i}^p w_{k,i}^p}{2\pi |\Sigma_{k,i}^p|^\frac{3}{2}} \times 
\exp\left(-\frac{1}{2}\left(\begin{bmatrix} x \\ y \end{bmatrix} - p_{k,i}^p \right)^T \Sigma_{k,i}^{-1} \left(\begin{bmatrix} x \\ y \end{bmatrix} - p_{k,i}^p \right)\right)  \quad (31)$$

From this figure it can be recognized that the filtering algorithm is able to adequately identify and track the contaminant clouds in a difficult scenario with heavy clutter. Figure 3 shows the average number of Gaussian shapes detected using the proposed algorithm for the 10 Monte Carlo runs. This average number has been computed using the existence variables on the previous scenario but also in the case where no contaminant cloud is present in the observation scene. It can be seen that the algorithm is clearly able to differentiate these two different scenarios.

6 Conclusions

In this paper, we have addressed the tracking problem of multiple contaminant clouds. We described a stochastic extension of the well-known Gaussian puff model in order to characterize the evolution of the average atmospheric pollutant concentration. A MCMC-based particle algorithm is then proposed to approximate the filtering distribution of the high dimensional state of interest. The tracking performance of the filter is demonstrated in a complex scenario of heavy level of false alarm. Simulations show that the proposed algorithm is able to efficiently detect and track multiple contaminant clouds. Future development work in this area will include dealing with more complex dynamical concentration cloud model.

Acknowledgments

This research was sponsored by the Data and Information Fusion Defence Technology Centre, UK, under the Tracking Cluster. The authors thank these parties for funding this work. We also would like to thank Nathan Green of DSTL for helpful discussions on this tracking problem, and the Statistical and Applied Mathematical Sciences Institute - SMC program for providing a collaborative research environment that assisted with the development of our ideas.

A LIDAR Likelihood Function

The likelihood function at time $t_k$ is given by:

$$p(f_k|\tilde{c}_k) = \prod_{r=R_1}^{R_{\text{max}}} p(f_{r,\theta_k}|\tilde{c}_{r,\theta_k})  \quad (32)$$

with

$$p(f_{r,\theta_k}|\tilde{c}_{r,\theta_k}) = \int p(f_{r,\theta_k}|c_{r,\theta_k}) p(c_{r,\theta_k}|\tilde{c}_{r,\theta_k}) dc_{r,\theta_k}  \quad (33)$$

These two distributions are given respectively by:

$$p(f_{r,\theta_k}|c_{r,\theta_k}) = \mathcal{N}(f_{r,\theta_k}|\alpha_r F c_{\tilde{c},\sigma_r^2}^2)  \quad (34)$$

and from Eq. (9)

$$p(c_{r,\theta_k}|\tilde{c}_{r,\theta_k}) = A \delta(c_{r,\theta_k}) + \mathcal{N}(c_{r,\theta_k}|\tilde{c}_{r,\theta_k}, V(\tilde{c}_{r,\theta_k})) 1_{[0,\infty]}(c_{r,\theta_k})  \quad (35)$$

with $1_{\mathcal{R}}(.)$ the indicator function on subset $\mathcal{R}$ and $A = \frac{1}{2} \left[ 1 - \text{erf} \left( \frac{\tilde{c}_{r,\theta_k}}{\sqrt{2V(c_{r,\theta_k})}} \right) \right]$. By substituting Eq.
(34) and Eq. (35) into (33) and rearranging, we obtain:

\[
p(f_r, \theta_k | \bar{c}_r, \bar{g}_k) = \mathcal{N}(f_r, \theta_k | 0, \sigma^2) + \int_0^{+\infty} \mathcal{N}(f_r, \theta_k | \alpha_r f_r, \theta_k, \sigma^2) \times \mathcal{N}(\bar{c}_r, \bar{g}_k| \bar{c}_r, \bar{g}_k, V(\bar{c}_r, \bar{g}_k)) dc_r, \theta_k
\]

After several algebraic manipulations, the likelihood function of unthresholded LIDAR measurements is thus given by:

\[
p(f_k|\bar{c}_k) = \prod_{r=R_3}^{R_{max}} \left\{ \frac{1}{2} \left[ 1 - \text{erf} \left( \frac{\bar{c}_r, \theta_k}{\sqrt{2V(\bar{c}_r, \theta_k)}} \right) \right] \right. \\
\times \mathcal{N}(f_r, \theta_k | 0, \sigma^2) + \left[ 1 + \text{erf} \left( \frac{\alpha_r F_f \bar{g}_k, \bar{c}_r^2 + \bar{c}_r, \bar{g}_k^2 \sigma^2}{\sqrt{2\sigma^2 + 2\sigma^2 F_f V(\bar{c}_r, \bar{g}_k)}} \right) \right] \\
\times \mathcal{N}(f_r, \theta_k | \alpha_r F_f, \theta_k, \sigma^2 + \alpha_r F_f^2 V(\bar{c}_r, \bar{g}_k)) \right\}
\]

Table 1: Simulation Parameters

<table>
<thead>
<tr>
<th>Atmospheric Parameters</th>
<th>(v, D)</th>
<th>( \left[ \begin{array}{cc} 0.2 &amp; 0.3 \ 0.8 &amp; 0.4 \end{array} \right] )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Model Uncertainty</td>
<td>Noise Variance ( V(\bar{c}_r, \bar{g}_k) )</td>
<td>((0.8\bar{c}_r, \bar{g}_k)^2 + 1e-15))</td>
</tr>
<tr>
<td>LIDAR</td>
<td>Location (X, Y)</td>
<td>((1000, 250))</td>
</tr>
<tr>
<td></td>
<td>Start Time</td>
<td>(180^\circ \pm 15^\circ) by step of (1^\circ)</td>
</tr>
<tr>
<td></td>
<td>Time between 2 angles</td>
<td>18</td>
</tr>
<tr>
<td></td>
<td>Number of Samples</td>
<td>200</td>
</tr>
<tr>
<td></td>
<td>Resolution</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td>Laser Energy E</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>Fluorescence Coefficient ( F )</td>
<td>1e-4</td>
</tr>
<tr>
<td></td>
<td>Noise Variance ( \sigma^2 )</td>
<td>2e-23</td>
</tr>
<tr>
<td></td>
<td>Attenuation Coefficients ( (\kappa_w, \kappa_n) )</td>
<td>((1e-6, 3e-4))</td>
</tr>
<tr>
<td></td>
<td>Threshold ( \lambda )</td>
<td>6e-12</td>
</tr>
<tr>
<td>Gaussian puff</td>
<td>Location ((X, Y))</td>
<td>((-300, -250); (-300, 150))</td>
</tr>
<tr>
<td>Characteristics</td>
<td>Mass (W_0)</td>
<td>(((-50, -50); (50, 50); (-30, -20); (400, 150)))</td>
</tr>
<tr>
<td></td>
<td>Noise Variance ( (\sigma^2_{\alpha}, \sigma^2_{\beta}) )</td>
<td>{(160; 150; 170; 50}, (0.1, 1)}</td>
</tr>
</tbody>
</table>

References


Figure 2: Tracking performance. Showing : True average concentration in log scale (a),(d),(g). Estimated average concentration in log scale (b),(e),(h). Thresholded fluorescences obtained from a complete LIDAR scan (c),(f),(i).
