Real-Time Radiological Source Term Estimation for Multiple Sources in Cluttered Environments

Kemp, Kumar, Bakker, Rogers

Real-Time Radiological Source Term Estimation for Multiple Sources in Cluttered Environments

Samuel Kemp[1], Satvik Kumar[2], Craig Bakker[3], and Jonathan Rogers[4]

Abstract— **A particle filter algorithm is presented to estimate the position, strength, and cardinality of an unknown number of radioactive point sources in an obstacle-rich environment using count measurements. The algorithm addresses gaps in the prior literature by incorporating two novel elements. The first is a precomputation step in which local terrain and obstacle data is processed to compute attenuation kernels throughout the search area. This enables rapid estimation performance in obstacle-rich environments as measurements are gathered. The second novel feature is a dynamic particle allocation technique in which the number of particles is adjusted in real time to meet convergence goals. This feature allows the algorithm to scale more efficiently to scenarios with a larger number of sources. A series of computational experiments using simulated data demonstrates the algorithm's performance in a cluttered environment with up to eight sources.**

I. Introduction

Governments and emergency responders are increasingly interested in using robotics to address nuclear accidents [1], radiological materials trafficking [2], and weapons proliferation [3]. There is now significant historical precedent for the deployment of robotic systems for nuclear disaster mitigation and surveying of sites with radiation contamination [4]–[7]. Missions for nuclear detection robots typically fall into two categories: mapping and source term estimation (STE). Mapping missions usually involve sites with widespread radiation contamination and require the robotic system to characterize the radiation environment over a defined area [8]–[10]. In contrast, radiological STE is the process of localizing and identifying parameters (location, strength, cardinality, etc.) of point sources of radiation [11]–[14]. Such STE missions may, for instance, stem from cases where nuclear materials are stolen or trafficked and must be located by law enforcement.

The STE problem for a single source has been solved by a variety of authors using least-squares regression [15], Kalman filters [16], particle filters [17]–[21], neural networks [22], and other heuristic algorithms [23]. STE for multiple sources can be much more difficult, however. In this case, the difficulty of the problem depends on whether measurements at all locations are only influenced by a single source (non-overlapping sources), or whether measurements at certain locations are influenced by multiple sources (overlapping sources). The degree to which sources overlap depends on their spacing, strengths, and the obstacle and terrain environment. The non-overlapping STE problem may be solved by estimating the source parameters independently, one source at a time (decomposing the problem into multiple single-source STE problems). This was the approach taken by Peterson *et al.* [24] and Chin *et al.* [25]. In [25], the authors introduce the concept of "fusion range," which limits the spatial range of influence that measurements can have, thereby restricting the method to use in non-overlapping cases only. Several authors have addressed overlapping STE problems, including Cook *et al.* [26], [27], Vavrek *et al* [28], Bandstra *et al* [29], Anderson *et al.* [30], and Ristic *et al.* [31]. In [26] and [27], the authors performed radiation contour mapping using simulated measurements but do not consider attenuation from obstacles or terrain. The work in [28] uses a sparse parametric imaging algorithm to reconstruct the 3D positions and activities of multiple gamma-ray point sources which works well for their custom sensing platform, and although obstacles were present for measurements, their effects were neglected due to the location of their measurements. Obstacles were later addressed in the single source case by the same group of researchers in [29]. The work in [30] proposes a particle filter (PF) capable of localizing multiple overlapping sources, but the sources in question must be of different isotopes. Thus the problem simplifies to single source STE for each isotope. Furthermore, the method was tested only for a two-source scenario in which the obstacle absorption parameters were known exactly *a priori*. On the other hand, Ristic *et al.* [31] propose a particle filter that handles multiple overlapping sources without requiring that they be different isotopes. While the algorithm was shown to exhibit good performance using experimental data, it was only validated with up to two sources and did not include a mechanism to account for attenuation by obstacles or terrain.

Prior work on STE has revealed several challenges. First, STE for larger numbers of sources (i.e., three to ten) remains a challenging problem due to the ambiguity caused by overlapping sources. Second, for particle filters, which have commonly been applied to STE, scalability problems arise due to the very large number of particles needed to represent hypotheses associated with numerous sources. Additionlly, STE in obstacle-rich environments has proven challenging for algorithms that are designed to run in real-time. This is because radiation transport modeling must be performed from each hypothesized source to calculate the predicted measurement at a particular location. For this reason, the majority of STE work has focused on a small number of

[1] School of Mechanical Engineering, Georgia Institute of Technology, Atlanta, GA 30332, USA, skemp32@gatech.edu.

[2] School of Aerospace Engineering, Georgia Institute of Technology, Atlanta, GA 30332, USA, skumar457@gatech.edu.

[3] National Security Directorate, Pacific Northwest National Laboratory, Richland, WA 99352, USA, craig.bakker@pnnl.gov.

[4] School of Aerospace Engineering, Georgia Institute of Technology, Atlanta, GA 30332, USA, jonathan.rogers@ae.gatech.edu.

sources (two or less) in mostly obstacle-free environments.

The current work seeks to address the gaps above through the design of a novel particle filter algorithm dubbed the Dynamic Discrete Particle Filter (DDPF). The goal of the algorithm is to find point-sources of gamma radiation in a defined search area that may include obstacles and terrain features. The number, location, and strengths of the sources is assumed to be unknown. The particle filter is designed to fuse multiple count measurements from Geiger-Müller (GM) counters, which may be fielded by personnel or mounted on vehicles. The DDPF improves upon existing particle filters for STE (such as that in [31]) by including two unique elements designed to improve runtime performance in cases with multiple sources and attenuation from terrain and obstacles. First, there is a discrete pre-processing step in which attenuation kernels are computed from discrete source locations to discrete measurement locations. Since attenuation calculations can be computationally intensive, this pre-processing step enables heavy computation to be done offline prior to the search, with results stored in a database for use in online processing of measurements. Such a decomposition allows the online portion of the algorithm to run on computationally-lightweight hardware (e.g., embedded computers) while also incorporating modeling results from complex 3-D radiation transport codes [32], [33]. The second innovation is the dynamic adjustment of the number of particles and their distribution over particle sets associated with different source cardinalities. This dynamic modification of the number and allocation of particles, modeled on the algorithm proposed in [34], allows the algorithm to handle scenarios with a large number of sources efficiently.

The paper proceeds as follows. Section II provides an overview of the problem statement. Section III details the particle filter design and includes detailed descriptions of the attenuation kernel computations and dynamic particle adjustment steps. Section IV provides a description of the simulation setup and metrics used for performance evaluation. Section V provides an extensive set of simulation results, demonstrating the benefits of the novel algorithm features by comparing performance to a benchmark algorithm based on that proposed in [31]. Overall, the proposed algorithm is shown to efficiently solve the STE problem for multiple sources in complex scenarios involving numerous obstacles with uncertain attenuation characteristics.

II. Problem Statement

Consider a search volume $\mathcal{A}$ containing known obstacles, terrain, and constant background radiation μ_b. The coordinate system is defined as (x, y, h) with respect to a local ground frame, where x and y represent Easterly and Northerly directions, respectively, and h is height above a reference altitude. There are an unknown number $r_{\text{true}} > 0$ of sources of gamma radiation present in $\mathcal{A}$. Point sources are indexed by $s \in \{1, ..., r_{\text{true}}\}$.

Each point source is parameterized by both its spatial coordinates $(x_s, y_s, h_s) \in \mathcal{A}$ and its equivalent strength φ_s. The equivalent strength characterizes the source's activity and photon energy level, along with conversions and scaling factors [35]. A set of K measurements, indexed by $k \in \{1, ..., K\}$, are taken in the space and parameterized by spatial coordinates $(x_k, y_k, h_k) \in \mathcal{A}$ and value $z_k \in \mathbb{N}$ which represents the total number of counts observed over time interval τ.

The problem statement is then to infer the source parameters (x_s, y_s, h_s) and $\varphi_s \ \forall \ s$ given the measurements. Furthermore, for the purposes of this work, it is desired that estimates for the source parameters be updated in real-time as new measurements are gathered.

It is known that the likelihood of measuring z counts during time interval τ from a source emitting an ideal count rate μ at a given measurement location is Poisson distributed [16], [31], [36]. Specifically, given the ideal number of counts $\lambda = \mu\tau$, an idealized measurement ζ is drawn from Eq. (1) where $\mathcal{P}$ denotes the Poisson distribution. However, due to the effects of saturation, the actual measured value z is given according to Eq. (2) which means z not necessarily Poisson distributed and is instead sampled from a saturated Poisson distribution $\tilde{\mathcal{P}}$ with parameters λ and μ_{sat} as shown by Eqs. (3) and (4),

$$\zeta \sim \mathcal{P}(\lambda) \tag{1}$$

$$z = \min(\zeta, \mu_{\text{sat}}) \tag{2}$$

$$z \sim \tilde{\mathcal{P}}(\lambda, \mu_{\text{sat}}) \tag{3}$$

$$\tilde{\mathcal{P}}(z; \lambda, \mu_{\text{sat}}) = \begin{cases} \mathcal{P}(z; \lambda) & z < \mu_{\text{sat}} \\ \sum\limits_{z \geq \mu_{\text{sat}}} \mathcal{P}(z; \lambda) & z = \mu_{\text{sat}} \\ 0 & z > \mu_{\text{sat}} \end{cases} \tag{4}$$

where μ_{sat} is the detector saturation value. This is the maximum number of counts the meter can report given its dead/recovery time, which is usually on the order of a few hundred microseconds [37]. As the radiation field propagates through the environment, the mean ideal count rate μ is subject to attenuation through the phenomena of absorption, scattering, and reflection.

III. Methodology

A. Overview

The structure of the DDPF proposed in this work is shown in Fig. 1. Two innovative aspects of the algorithm are highlighted in this figure. First, prior to gathering measurements, a radiation transport model is used to compute so-called "attenuation kernels". This pre-computation is meant to be done offline using a transport model of any desired complexity. As described in Section III-B, the attenuation kernels encode radiation transport modeling from a discrete set of candidate source locations to a discrete set of measurement locations. These attenuation kernels are used online as measurements are taken to perform recursive Bayesian estimation via sequential Monte Carlo methods (i.e., particle filtering). During the estimation process, the particle filter uses the Dynamic Particle Count Adjustment (DPCA) algorithm based on [34] to adjust the number of particles in order to balance speed

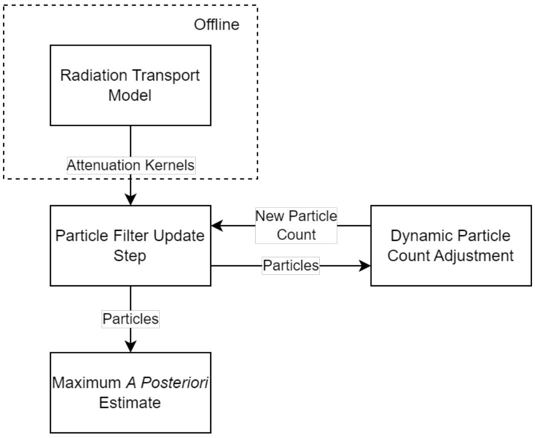
Offline
Radiation Transport Model
Attenuation Kernels
Particle Filter Update Step
New Particle Count
Dynamic Particle Count Adjustment
Particles
Particles
Maximum *A Posteriori* Estimate

Fig. 1. Overview of Dynamic Discrete Particle Filter Algorithm.

and accuracy. This element comprises the second major innovation of the DDPF design. The current best estimate of the source terms is updated online, as new measurements are gathered, via a maximum *a posteriori* estimation routine applied to the particle set. The various algorithm components are described in detail below with an emphasis on the novel elements.

Prior to describing each algorithm component, several brief definitions are provided. Since the particle filter is updated after each new measurement, iterations of the algorithm correspond to the current measurement index k. The particle filter uses a set of N particles $\mathbf{X}_k$, indexed by $n \in \{1, ..., N\}$. A single "particle" at iteration k, denoted $\mathbf{X}_k^{(n)}$, consists of a source parameter matrix $\boldsymbol{\theta}$ and weight w, and represents a single hypothesis for the solution of the STE problem posed in Section II. Unlike other particle filters designed for STE [16]–[21], [25], [30], a particle in this work can represent a combination of any number of sources, $r \in \{1, ..., r_{\text{max}}\}$.[1] The source parameter matrix for each particle is an $r \times 3$ matrix, $\boldsymbol{\theta} = (\theta_1, \cdots, \theta_r)$, where each row is a 1×3 source parameter vector $\theta_s = [x_s, y_s, \varphi_s]$ representing an individual source hypothesized by the particle. It is assumed that all sources lie on the terrain surface, and thus the height coordinates of the sources $h_s = h_s(x_s, y_s)$ are not included as independent parameters.

B. Precomputation of Attenuation Kernels

Let $\mathbf{X}_k^{(n)}$ be an arbitrary particle at iteration k, and let measurement z_k be taken at coordinate (x_k, y_k, h_k). Define $\lambda(\mathbf{X}_k^{(n)})$ as the ideal counts observed at location (x_k, y_k, h_k) due to the sources described by $\mathbf{X}_k$ as,

$$\lambda(\mathbf{X}_k^{(n)}) = \mu_{k,n}\tau \tag{5}$$

where $\mu_{k,n}$ is the ideal count rate due to the arbitrary particle $\mathbf{X}_k^{(n)}$. Then the likelihood of observing z_k given $\lambda(\mathbf{X}_k^{(n)})$, including the effects of detector saturation, is given by the saturated Poisson distribution, $\tilde{\mathcal{P}}$,

$$p\left(z_k|\mathbf{X}_k^{(n)}\right) = \tilde{\mathcal{P}}\left(z_k; \lambda\left(\mathbf{X}_k^{(n)}\right), \mu_{\text{sat}}\right) \tag{6}$$

The particle filter requires that the likelihood in Eq. (6) be computed at least $N + NS$ times every time a new measurement is obtained, where N is the number of particles and S is the number of progressive correction stages (described in Section III-C.5). Computing $\lambda(\mathbf{X}_k^{(n)})$ requires knowledge of the ideal count rate $\mu_{k,n}$ given the source parameters in $\mathbf{X}_k^{(n)}$. In settings involving obstacle and terrain occlusions, $\mu_{k,n}$ must be determined via a radiation transport model. Such models exhibit varying levels of fidelity (e.g., [32], [33]), but in general models that capture complex phenomena such as obstacle absorption or scattering can be so computationally intensive that they are infeasible to run online $N + NS$ times every time a measurement is obtained.

To address this computational bottleneck, the proposed DDPF replaces online radiation transport modeling with offline transport modeling over a set of hypothesis-measurement pairs. The linear relationship between source strength and ideal count rate that the kernels exploit allows the online likelihood calculation step to be reduced to a simple linear algebra operation. First, the search volume $\mathcal{A}$ is discretized into a set of $\mathcal{I}$ discrete source locations and $\mathcal{J}$ discrete measurement locations. Define a so-called *attenuation kernel* $\Theta_{i,j}$ between source location $i \in \mathcal{I}$ and measurement location $j \in \mathcal{J}$ which may model the effects of attenuation, reflection, transmission, etc. Then the ideal count rate $\hat{\mu}_{k,n,j}$ due to particle n at iteration k, not accounting for background interference, at location j is given by,

$$\hat{\mu}_{k,n,j} = \sum_{s=1}^{r_{k,n}} \varphi_k^{(n)}(s)\Theta_{i,j} \tag{7}$$

In (7), $r_{k,n}$ is the maximum number of sources hypothesized by particle $\mathbf{X}_k^{(n)}$, and $\varphi_k^{(n)}(s)$ is the strength of hypothesized source s of particle $\mathbf{X}_k^{(n)}$. The ideal count rate accounting for background radiation is then be obtained as,

$$\mu_{k,n,j} = \mu_b + \hat{\mu}_{k,n,j} \tag{8}$$

where μ_b is the average count rate due to background radiation.

Each time a new measurement is obtained, the particle filter computes the likelihood associated with all N particles. This likelihood calculation requires that the ideal count rates for all particles be recomputed and compared with the observed count rate. The simplicity with which the ideal count rates can be computed using the attenuation kernels makes this process fairly trivial computationally. Define vector $\boldsymbol{\Theta}_{k,n} = [\Theta_{i(k,n,1),j}, \cdots, \Theta_{i(k,n,r_{k,n}),j}]$ where $i(k, n, s)$ is the source location of source s of particle $\mathbf{X}_k^{(n)}$. Likewise, define vector $\boldsymbol{\varphi}_{k,n} = [\varphi_k^{(n)}(1), \cdots, \varphi_k^{(n)}(r_{k,n})]$. Then, using Eq. (7), the ideal count rates for all particles can be computed at the k^{th} measurement update (assumed to occur at location j) through the following N inner products:

[1] Note that in this work the null hypothesis of no sources present ($r = 0$) is excluded. This differs from [31].

$$\begin{bmatrix} \hat{\mu}_{k,1,j} \\ \vdots \\ \hat{\mu}_{k,N,j} \end{bmatrix} = \begin{bmatrix} \mathbf{\Theta}_{k,1} \cdot \boldsymbol{\varphi}_{k,1} \\ \vdots \\ \mathbf{\Theta}_{k,N} \cdot \boldsymbol{\varphi}_{k,N} \end{bmatrix} \tag{9}$$

Note that Eqs. (7) and (9) only make sense if source locations in each hypothesis $\mathbf{X}_k^{(n)}$ are restricted to lie at locations in $\mathcal{I}$. This assumption enables radiation transport modeling to be shifted offline, allowing for arbitrarily complex models to be used in real-time filtering. In practice, the discretization of $\mathcal{A}$ limits the spatial accuracy that can be achieved by the filter, and thus the resolution of the discretization must be chosen such that the solution for the source locations meets the desired accuracy requirements. An obvious trade-off exists between the spatial accuracy that can be achieved and the number of attenuation kernels that must be computed offline and stored. Note that, while the source locations are converted to discrete variables, the source strengths remain continuous variables.

For purposes of illustration, a simplified radiation transport model as described in [35] and [38] is used in this paper. This model is capable of modeling absorption through various media and mean background radiation, but it does not model reflection, scattering, or transmission.

Given a set of sources φ_s, $s \in \{1, ..., r\}$, the ideal count rate observed at a particular location (x, y, h) is given by,

$$\mu = \mu_b + \sum_{s=1}^{r} \frac{\varphi_s}{d_s^2} e^{-\beta_{\text{md}} d_s} \tag{10}$$

where $d_s = \sqrt{(x - x_s)^2 + (y - y_s)^2 + (h - h_s)^2}$. In (10), β_{md} is the absorption coefficient of the medium. Since the path from measurement to source may pass through various media, (10) is only valid along paths with a constant β_{md}. One method to handle this is through ray (or path) tracing. In this work, ray tracing is used to compute the attenuation kernels from a grid of possible source locations to a grid of measurement locations. This process is depicted in Figs. 2 and 3. In Fig. 2, the source grid is shown with light dots, and the measurement grid is shown with heavy dots. Starting from the source location, a ray is cast to the measurement location and the intersections where the ray passes through obstacles of different absorption coefficients are recorded (dashed lines in Fig. 3). Using a reference source strength of $\varphi_s = 1$ at the source location, Eq. (11) is used to compute a count rate at the first intersection using the β_{md} value of the first region (initial red line in Fig. 3 from 0-20 m).

$$\mu_s = \frac{\varphi_s}{d_s^2} e^{-\beta_{\text{md}} d_s} \tag{11}$$

Then, a new equivalent source strength is computed by solving Eq. (11) using the β_{md} of the next segment and the count rate at the first intersection. These two steps are repeated consecutively over all segments until the measurement location is reached, resulting in a count rate at the measurement location $(\mu_{i,j})$ given a unit source at the original source location. The attenuation kernel is then simply $\Theta_{i,j} = \mu_{i,j}$. While this ray tracing approach is used in this work to compute the attenuation kernels, any radiation transport model of desired complexity may be used as discussed above.

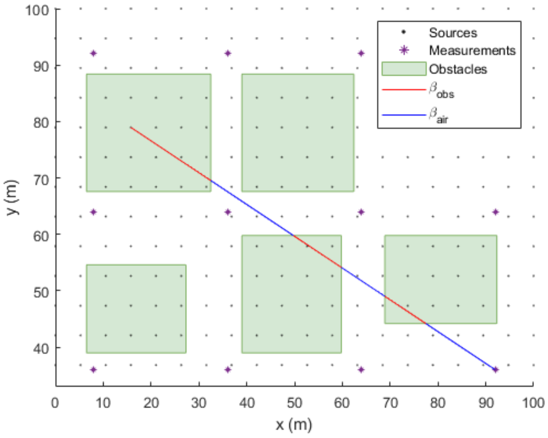

Fig. 2. Example of Discretized Ray Tracing Over Source and Measurement Grids.

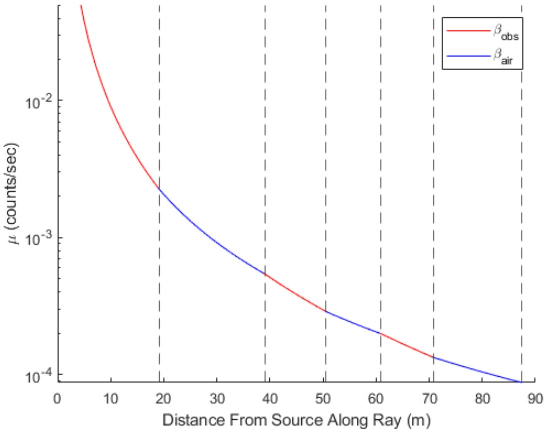

Fig. 3. Example of Equivalent Source Strength Values Computed Over Ray.

C. Particle Filter Update Step

1) Initialization: The filter is initialized by sampling N particles. Each particle has an associated source cardinality r. In [31], the r values are sampled uniformly from the integers in the set $\{1, \cdots, r_{\text{max}}\}$. While this correctly models the fact that the source cardinality is (presumably) unknown *a priori*, as the number of sources grows the spatial coverage of the particle set becomes very poor in high dimensions. For example, 100 particles may be adequate to represent possible source locations in a 2-dimensional space, but 100 particles in a 6-dimensional space results in extremely sparse coverage of the spatial domain. In this work, better results were obtained by initializing the particle set such that the number of particles allocated to higher dimensions grows exponentially with r. Specifically, r is sampled from a probability distribution wherein the probability of selecting

Algorithm 1 Particle Filter Update Step

```
 1: function $\mathbf{X}_k$ = UPDATE($\mathbf{X}_{k-1}$, $z$)
 2:     $X = \mathbf{X}_{k-1}$
 3:     for $r = 1, ..., r_{\max}$ do                                  ▷ Begin PC
 4:         $m = 0$
 5:         for $n = 1, ..., N$ do
 6:             if size$(X, 2) = r$ then
 7:                 $m = m + 1$
 8:                 $I^{(m)} = n$
 9:                 $Y^{(m)} = X^{(n)}$
10:             end if
11:         end for
12:         Select coefficients $\gamma_s$ s.t. $\sum_s \gamma_s = 1$
13:         for $s = 1, ..., S$ do
14:             $Mr = m$
15:             for $m = 1, ..., Mr$ do
16:                 $w^{(m)}$ = $C*$ Likelihood ($X^{(m)}$, $z$)$^{\gamma_s}$
17:             end for
18:             Resampling: $Y$ = Resample($Y$,$w$)
19:             Regularization: $Y$ = RegularizeParams($Y$)
20:         end for
21:         for $m = 1, ..., Mr$ do
22:             $X(I^{(m)}) = Y^{(m)}$
23:         end for
24:     end for
25:     for $n = 1, ..., N$ do                                         ▷ Begin FF
26:         $w^{(m)}$ = $C*$ Likelihood ($X^{(n)}$, $z$)
27:     end for
28:     Resampling: $X$ = Resample($X$,$w$)
29:     Regularize R: $X$ = RegularizeR($X$)
30:     $\mathbf{X}_k = X$
31: end function
32: function $w$ = LIKELIHOOD($X$, $z$)
33:     $w_{last} = 1$
34:     for k = 1:K do
35:         $w = w_{last} * p(z_k|X)$
36:         $w_{last} = w$
37:     end for
38: end function
```

$r = r'$ is given by

$$P\left(r = r'\right) = p^{r'} \tag{12}$$

where $r' \in \{1, \cdots, r_{\max}\}$ and p is the solution to

$$1 = \sum_{r=r_{\min}}^{r_{\max}} p^r \tag{13}$$

The maximum source cardinality $r_{\max}$ is chosen by the operator based on external information about the number of possible sources for which to search. Given N particles, each with a sampled r value, the spatial coordinates of each source for each particle may be sampled from uniform distributions dictated by the boundaries of $\mathcal{A}$. Likewise, the source strengths may be sampled from a uniform distribution within selected minimum and maximum values[2]. Note that additional *a priori* information may of course be used to change the initial sampling approach as desired in a specific scenario.

2) Likelihood Calculation: Algorithm 1 shows the particle filter update (i.e. Bayesian inference) step, which is executed every time a new measurement z_k is obtained. The algorithm takes as input all past measurements z and the source parameter hypotheses at the previous update step, denoted $\mathbf{X}_{k-1}$. The update step outputs an updated set of hypotheses (particles) given the current measurement. There are three critical operations in the update step: likelihood computation, resampling, and regularization.

The likelihood computation steps are shown in Algorithm 1, line 32. This is accomplished using Eq. (6) which shows that the probability of a measurement given a source parameter array $\mathbf{X}$ is given by sampling a saturated Poisson distribution with parameter λ found by using Eq. (5). This computation is not just performed on the current measurement, but on all past measurements for a given particle. Since each measurement is independent of the others, the probabilities of each measurement resulting from a given particle can be cumulatively multiplied together. This returns the probability that every measurement could have come from the given particle. The reasoning for using all the measurements (prior and current) is explained further in Section III-C.4. The particle weights are found by normalizing the probabilities of each particle by a constant, C, such that the weights sum to one.

3) Resampling Algorithm: The second critical operation is resampling as shown in Algorithm 1, lines 18 and 28. Systematic resampling [39], [40] is used in this work. This process involves computing the cumulative sum vector of the weights and using it to build a vector of indices representing the particles to be resampled. Further details of systematic resampling can be found in [39], [40] as well as other references, and thus are omitted here.

4) Regularization: The final major element of the update step is regularization (lines 19 and 29 of Algorithm 1). In line 19, the parameters themselves are regularized. Each parameter, x_s, y_s, and φ_s is perturbed by a Gaussian kernel, $\nu \sim \mathcal{N}(0, \sigma)$ (h_s is not directly perturbed as the terrain dictates the mapping from (x_s, y_s) to h_s). Here σ is selected as either $\sigma_{x,y}$ or σ_{φ}, which are the variances for the spatial coordinates and source strengths respectively.[3] Regularization helps counter particle degeneracy [41] by increasing particle coverage. A variance that is too small will result in insufficient coverage of the space for small numbers of particles, while a variance that is too large invalidates the Markov assumption (i.e., particles could retread old ground, losing information from past measurements). In this work, a larger variance is selected for greater coverage, and an informed "retraining" step is added to overcome information loss. Informed retraining is a strategy wherein the weights of the particle set are computed based on a selection of prior measurements instead of just the most recent one. In this work, since the set of measurements is limited and the search space is relatively small, all previous measurements are used for retraining. Note that this retraining step is further enabled by the ability to rapidly compute the ideal count rates using the attenuation kernels per Eq. (9) – this step would be far

[2] Results show a low sensitivity to initial φ, and thus the range is not critical for performance.

[3] In the discrete PF, the spatial parameters are fixed to a grid of points. During regularization, the new perturbed coordinates are rounded to the nearest discrete point.

more expensive if complex transport models had to be used online.

Regularization is also performed on the estimated number of sources r for each particle (line 29 of Algorithm 1). This creates a small probability of increasing or decreasing r for a particle based on tunable parameters p_{up} and p_{dn}, which represent the probability of adding a new source or removing a source from the particle. A new source is added to a particle by copying a randomly and uniformly selected source from the combined set of all hypothesized sources in the PF. When removing a source, a source is randomly and uniformly selected from those of a given particle and removed.

5) Progressive Correction: Progressive correction (PC) is a stochastic form of the particle flow with log-homotopy designed to reduce particle degeneracy [42]–[44]. More particles survive the correction step (the correction step is the combination of the regularization and resampling steps) during PC because the variance in the likelihood is reduced by the exponent γ_s. Essentially, PC splits the correction step into several smaller substeps. The progressive correction step begins on line 3 of Algorithm 1. Note the particles are only competing against particles of the same source cardinality which is in contrast to full field correction in line 25 where all particles compete against all other particles. Normally, PC is a substitute for the correction step. In this case, since PC is performed separately for each source cardinality hypothesis, the full field correction is needed to allow particles of different source cardinalities to compete against each other.

6) Maximum A Posteriori Estimate: The posterior PDF for all measurements up to step k, $p(\mathbf{X}_k|z_{1:k})$, is approximated by the empirical distribution of particles $\{w^{(n)}, \mathbf{X}_k^{(n)}\}_{n=1}^N$, where $w^{(n)}$ is the weight of the n^{th} particle. The maximum *a posteriori* (MAP) estimate for a source parameter is the mean of the marginal PDF of the parameter—since the weights are uniform after each update step, this simplifies to the mean of the given parameter across the particle set. The first step in estimating the source parameters from the particle set is to find the estimated source cardinality according to,

$$\hat{r} = \text{nint}\left(\frac{1}{N}\sum_{n=1}^{N}\left|\mathbf{X}_k^{(n)}\right|\right) \tag{14}$$

where $|\cdot|$ extracts the cardinality, or number of sources, of a particle and nint() is a function that rounds to the nearest integer. Then, particles for which $r \neq \hat{r}$ are removed from the particle set. k-means clustering is then performed on the remaining particles where $k = \hat{r}$. These clusters represent individual location and source strength hypotheses for each of the identified point sources. The MAP estimates of (x_s, y_s, φ_s) are then found as the centroid of each cluster. The heights of each source may then be estimated by interpolating a terrain map using the MAP estimates of x_s and y_s.

D. Dynamic Particle Count Adjustment Algorithm

The number of particles N is a tunable parameter with well-known tradeoffs between computational burden (high N) and inaccurate estimates due to sparse coverage of the domain (low N). Elvira *et al.* [34] developed an algorithm to dynamically adapt N in response to estimates of filter convergence as measurements are gathered. This method was shown to effectively balance the above tradeoffs. In the context of the current work, the ability to accurately identify large numbers of sources in the environment (e.g., more than five) hinges on use of a sufficiently large number of particles that produce adequate spatial coverage. By dynamically adjusting N, the filter maintains sufficient coverage when initial measurements are gathered but is not saddled with large numbers of extraneous particles later on, once the particles converge to the likely source parameters.

A Dynamic Particle Count Adjustment (DPCA) algorithm is developed in this work based on [34]. This method, shown in Algorithm 2, is executed after the PF's update step. All of the measurements taken up to the latest measurement step k are used, denoted as $\mathcal{Z}_{1:k} = \{z_1, \cdots, z_k\}$. Each measurement is paired with a set of fictitious measurements $\tilde{\mathcal{Z}} = \{\tilde{z}_1, \cdots, \tilde{z}_J\}$ drawn from J particles randomly and uniformly sampled from the PF (line 5) to compute a set of the negative log Poisson likelihoods $\mathcal{Q} = \{q_1, \cdots, q_k\}$ (line 8) that each of the true measurements could have come from the set of fictitious measurements given by the PF. The set is represented by its mean measurement value, $\bar{z}$. Using a *negative* log likelihood means a higher value represents a less likely source configuration. The maximum value of $\mathcal{Q}$, q_{max}, is compared to tunable q_h and q_l values, and the particle count is increased or decreased according to tunable functions, f_{h} or f_{l} (Lines 20 and 25). If more particles are added, they are added according to the initialization function used at the start of the PF algorithm (line 21). If the number of particles decreases, they are randomly and uniformly removed (line 26). Tuning J, q_h, q_l, f_{h}, and f_l is required to achieve a suitable balance between runtime and accuracy (e.g. a higher q_h indicates a higher tolerance for unlikely particles, and a more aggressive f_{h} would prioritize accuracy over runtime).

IV. Experimental Setup

A. Simulation Setup

Simulation results are presented in the next section to demonstrate performance of the algorithm and compare it to the particle filter design in [31]. All results in this paper use the search area shown in Fig. 4, which measures 100m × 200m. Building footprints were obtained from OpenStreetMaps (OSM) [45], and for ease of implementation were treated as solid structures. Furthermore, the terrain in this example location is relatively flat, so for the purposes of this work all terrain and source heights are assumed to be zero. Buildings are modeled as prisms such that the ground footprint extends upward uniformly to the building height. Building height data was taken from OSM, and buildings without height data were assumed to be 4.3m tall.

Algorithm 2 Dynamic Particle Count Adjustment Algorithm

```
function $\mathbf{X}_k$ = DYNAMIC($\mathbf{X}_k$, $\mathcal{Z}_{1:k}$)
    for $\kappa = 1 : k$ do
        for $j = 1 : J$ do
            $i \sim \mathcal{U}\{1, N\}$
            $\tilde{z}_j \sim p\left(z_\kappa|\mathbf{X}_k^{(i)}\right)$
        end for
        $\bar{z} = \text{mean}(\{\tilde{z}_1, \cdots, \tilde{z}_J\})$
        $q_\kappa = -\log \mathcal{P}(z_\kappa; \bar{z}, \mu_{\text{sat}})$
    end for
    $q_{\text{max}} = \max(\mathcal{Q})$
    if $q_{\text{max}} > q_h$ then
        $X, N = f_{up}(X, N)$
    end if
    if $q_{\text{max}} < q_l$ then
        $X, N = f_{dn}(X, N)$
    end if
    $\mathbf{X}_k = X$
end function
function $[X, N] = f_{up}(X, N_{old})$
    $N = f_{\text{h}}(N_{old})$
    $X_{new} = \text{initialize}(N - N_{old})$
    $X = X_{new} \cup X$
end function
function $[X, N] = f_{dn}(X, N_{old})$
    $N = f_{\text{l}}(N_{old})$
    $s \in \mathbb{N}, s \in [1, N - N_{old}]$
    $X = X^{(s)}$
end function
```

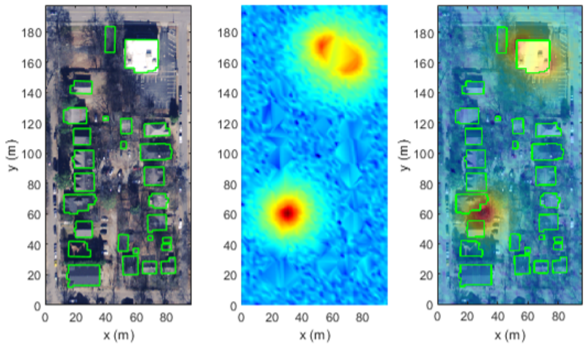

Fig. 4. Search Area for Simulation Trials. (left) Satellite view of search area with buildings outlined in green. (center) Ray-traced radiation intensity from three example sources (red = high intensity). (right) Combined plot of radiation and environment.

The attenuation kernels were calculated across this domain at a grid of 44 measurement locations and 4,900 possible source locations. The number of measurements was chosen to be as sparse as possible considering the range of possible source strengths and the size of the environment. This is why it is much less than the number of possible source locations, which was chosen to allow for a high resolution solution. The assumed measurement height was 3 m (simulating, for example, a detector mounted on top of a truck) and measurement locations are restricted to lie outside buildings. The grid of measurement locations is shown in Fig. 5 by yellow dots. When computing the attenuation kernels, linear absorption coefficients of $\beta_{\text{obs}} = 1 \times 10^{-2}$ m^{-1} and $\beta_{\text{air}} = 1 \times 10^{-6}$ m^{-1} were used for transmission through buildings and air, respectively.

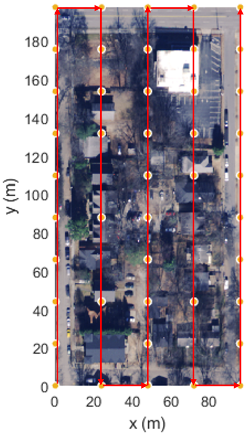

Fig. 5. Example Boustrophedon Search Pattern.

TABLE I

SIMULATION PARAMETERS.

Parameter	Value	Parameter	Value
N	varies	r_{max}	varies
Number of measurements	44	r_{true}	varies
Possible source location points	4,900	φ_{min} (counts/s)	5,000
β_{obs} m^{-1}	varies	φ_{max} (counts/s)	12,000
β_{air} m^{-1}	1.00E-06	S	20
μ_b (counts/s)	1	σ_{pos} (m)	3
μ_{sat} (counts/s)	5,000	σ_φ (counts/s)	20
$SNR_{min}(dB)$	25	p_{up}	0.003
$\tau_{max}(s)$	60	p_{dn}	0.003
$\tau_{min}(s)$	1	q_h	30
Measurement height (m)	3	q_l	10
Total search area width (m)	100	$f_{\text{h}}(N_{old})$	$50N_{old}$
Total search area length (m)	200	$f_{\text{l}}(N_{old})$	$N_{old}/1.2$

In the examples below, simulating measurements requires that an exposure time τ be selected. An exposure time is calculated as follows. First, λ_{min} is selected representing the minimum acceptable measured count. This value can be selected arbitrarily, or can be derived by solving Eq. (15) for λ_{min} [16]:

$$SNR_{min} = 10 \log_{10}(\lambda_{min}/\mu_b) \tag{15}$$

where SNR_{min} is a design variable representing the minimum acceptable signal-to-noise ratio. Then, given a count rate μ, τ is selected as $\tau = \lambda_{min}/\mu$. Finally, τ is constrained on the interval $[1, 60]$ s.

With the attenuation kernels pre-computed and stored, the online portion of the algorithm is executed as measurements are obtained. Measurements in all trials shown below are obtained through a boustrophedon search pattern [46] shown in Fig. 5. All simulation parameters are listed in Table I.

B. *Performance Metrics*

Performance of the algorithm is evaluated in two ways: run time and accuracy. Run time measures the computation time involved in the online portion of the algorithm only. Accuracy is measured differently depending on the source parameter. Accuracy of the source cardinality estimate is measured as the absolute difference between the true and estimated number of sources.

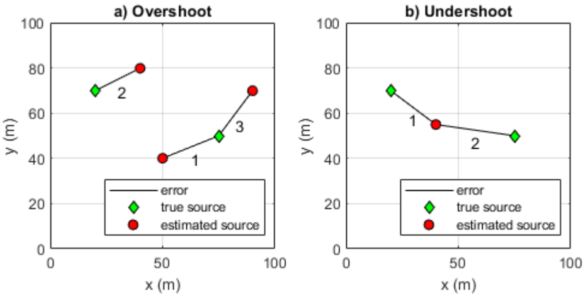

Fig. 6. Error Associations for Scenarios in which Particle Filter Overestimates (left) and Underestimates (right) Source Cardinality.

To assess the accuracy of the source location and strength estimates, comparisons are made between the estimated sources and the nearest true source by distance. In cases where the estimated number of sources $\hat{r}$ is higher or lower than the true number of sources r_{true}, the selection of the "nearest" source is not necessarily straightforward. Figure 6 shows how this is handled. If the number of sources is overestimated or if $\hat{r} = r_{\text{true}}$, the estimated sources are compared against the nearest actual sources (Fig. 6, left). This means a true source may factor into the error calculations more than once. If the number of sources is underestimated, each true source is compared to the nearest estimated source (Fig. 6, right). This means an estimated source might contribute to error calculations more than once.

With the "nearest sources" defined, accuracy of the source location estimate is assessed as the sum of the Euclidean distances between the true and estimated spatial parameters according to,

$$\epsilon_{\text{pos}} = \sum_{i \in \mathcal{R}} \Delta d_i \tag{16}$$

where $\mathcal{R} = \{1, ..., \max(\hat{r}, r_{\text{true}})\}$ and Δd_i is the Euclidean distance between the i^{th} estimated source and its nearest true source for the $\hat{r} \geq r_{\text{true}}$ case, or the Euclidean distance between the i^{th} true source and its nearest estimated source for the $\hat{r} < r_{\text{true}}$ case. Likewise, accuracy of the source strength estimate is measured as,

$$\epsilon_{\varphi} = \sum_{i \in \mathcal{R}} \Delta \varphi_i \tag{17}$$

where $\Delta\varphi$ is the absolute value of the difference in source strength between the nearest true and estimated sources.

V. Results

A. Example Simulation

An example simulation is provided to contextualize the Monte Carlo studies presented in the following sections. It is performed using three sources with random locations and source strengths. The locations are shown in Fig. 7 as green dots, and the source strengths (from the source at the maximum y value to the minimum y value) are 7,000, 12,000, and 12,000 counts/sec, respectively. Figure 7 also shows the hypothesized sources at various stages of the particle filter operation, from initialization (top left), successive update steps, and the final maximum *a posteriori* estimate (bottom right). Hypothesized sources are shown as colored dots where color indicates the particle's source cardinality estimate (blue = 1 source, red = 2 sources, yellow = 3 sources). The size of the particles correspond to their likelihood after a the measurement update. Measurement locations are shown as black asterisks. The building footprints are also shown in green for clarity in the final plot. The initial particle count is $N = 5,000$, and time histories of N and $q_{\max}$ as a function of time is shown in Fig. 8. The values for $q_h = 30$ and $q_l = 10$ are shown as dashed lines labeled "Increase" and "Decrease," respectively. Note that at $k = 33$ the DPCA algorithms grows the particle set significantly when $q_{\max}$ exceeds q_h due to an aggressive f_{h} function shown in Table I – this results in a large set of low-likelihood particles scattered across the domain as seen in the bottom center panel of Fig. 7.

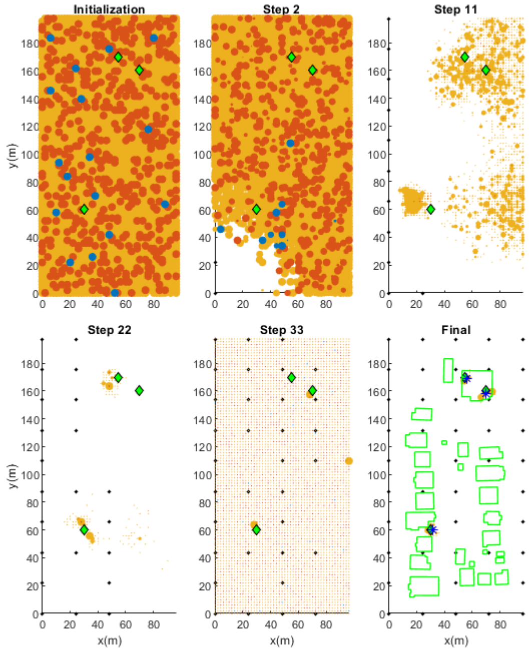

Fig. 7. Stages of the PF. (top-left) initialization. (bottom-right) post-processing. (all others) update.

The final maximum *a posteriori* results for this example case exhibit excellent accuracy, with $\hat{r} = 3$ sources, $\epsilon_{\text{pos}} = 2.3$ m, and $\epsilon_{\varphi} = 1,463$ counts/sec. The measurement updates, maximum *a posteriori* estimation, and DPCA algorithm steps (i.e., the steps performed online) required 32 sec to run on an AMD Ryzen 5 3600X 4.4 GHz processor. Note that this runtime number also excludes the detector dwell time, τ, required each time a measurement is obtained, as this is not pertinent to the algorithm runtime itself.

B. Attenuation Kernel Study

In the following sections, a series of Monte Carlo simulations is presented to highlight the tradeoffs of various components of the proposed algorithm. The first study investigates the effect of problem discretization using the

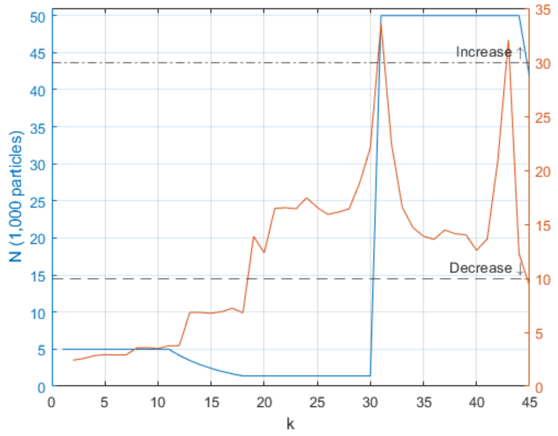

Fig. 8. Convergence Metrics and Particle Count for Sample Trial Using the DDPF.

attenuation kernels. Two algorithms are compared. The first is the proposed "discrete" particle filter, or DPF, which uses the discretized formulation presented in Section III-B. The second is the so-called "continuous" particle filter (CPF), which is based on the algorithm in [31]. This algorithm does not employ any spatial discretization, and instead requires ray tracing to be performed online for each hypothesis at each update step. In order to isolate the effects of discretization only, the DPF in this example does not include the DPCA algorithm; the particle count is fixed at $N = 500$ instead. This relatively low value of N is used to keep computation times reasonable for the CPF case.

A Monte Carlo simulation was performed to investigate general performance trends. In all cases, the particle filter was configured with $r_{\max} = 3$. Thirty random source configurations were generated. The true number of sources, r_{true} used in each scenario was randomly and uniformly selected between 1 and 3 (since the assumed maximum number of sources is not necessarily equal to the true number of sources). The source locations were then randomly sampled uniformly across the search area, and the strengths were randomized uniformly between 5,000 and 12,000 counts/sec. For each of the 30 random source configurations, 5 trials were run with the CPF and DPF using different random seeds to generate the initial particle set. This resulted in a total of 150 simulation trials for the CPF and DPF. Furthermore, to study the effects of the obstacles, these 150 trials were repeated with and without the obstacles.

The results of these simulations are shown in Table II. The line shows the portion of trials in which the source cardinality estimate $\hat{r}$ is correct. The means and standard deviations of ϵ_{pos} and ϵ_{φ} are shown as $\mu()$ and $\sigma()$, respectively. Overall, the accuracy of the CPF and DPF are not significantly different, as expected. However, the CPF exhibits average runtimes that are over two orders of magnitude greater than the DPF, even in the obstacle-free case. Furthermore, because the CPF must compute radiation transport online, its runtime increases with the number of obstacles (as seen by the difference in average CPF runtime between the obstacle and no-obstacle cases). Because this computation is shifted offline for the DPF, its online computation time is independent of the number of obstacles.

TABLE II
MONTE CARLO RESULTS COMPARING CPF AND DPF PERFORMANCE ($N = 500$).

Particle Filter	**CPF**		**DPF**	
Obstacles	No	Yes	No	Yes
$\hat{r}$ **Correct (%)**	84	83.33	87.33	86
$\mu(\epsilon_{\text{pos}})$ **(m)**	4.736	4.494	5.592	5.259
$\sigma(\epsilon_{\text{pos}})$ **(m)**	4.357	5.121	7.094	6.469
$\mu(\epsilon_{\varphi})$ **(counts/s)**	736.4	488.8	414.2	427.3
$\sigma(\epsilon_{\varphi})$ **(counts/s)**	2534	2392	2247	2653
Avg. Runtime (s)	1730	31,910	13.5	12.9
Std. Dev. Runtime (s)	51.08	5070	1.407	1.147

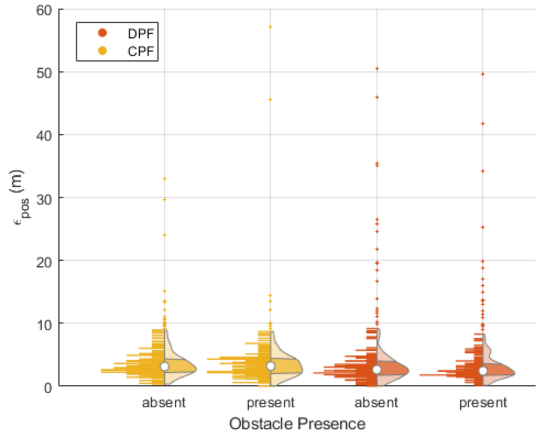

Fig. 9. Position Error for CPF and DPF in Cases With and Without Obstacles.

While runtime improvements are achieved by the DPF without significant loss of accuracy when viewed from the perspective of mean errors, the DPF is subject to more outlier solutions compared to the CPF. This is due to the low number of particles used in these cases. Figure 9 shows a series of violin plots corresponding to each of the configurations run. Each violin contains 150 trials. The left side of the violin is a histogram, the right side is the Inter-Quartile Range (IQR) with the white dot as the median. The values plotted in the violin exclude failure cases which are shown as colored points above the violin plots (failed cases are defined as those with performance metrics which are more than three scaled mean absolute deviations from the median). As shown in the figure, a handful of cases for both the CPF and DPF resulted in very high position errors, usually caused by undershooting or overshooting the source cardinality. The DPF resulted in a higher number of these "failure" cases, which is the cause of the slightly higher mean position errors in Table II. Note, however, that the median of the DPF position error is still lower than the CPF—this means that outside of these failure cases, the accuracy of the DPF was actually marginally better.

TABLE III

Monte Carlo Results with DPF Using Varying N.

Particle Filter	**DPF**					
Particle Count	**100**	**500**	**2 500**	**5,000**	**25,000**	**50,000**
$\hat{r}$ **Correct (%)**	78.67	86	92	92.67	94.67	96
$\mu(\epsilon_{\text{pos}})$ **(m)**	9.457	5.259	3.703	3.579	3.415	3.194
$\sigma(\epsilon_{\text{pos}})$ **(m)**	13.92	6.469	3.43	3.87	3.076	2.821
$\mu(\epsilon_{\varphi})$ **(counts/s)**	466.4	427.3	521.2	474.4	186.5	342
$\sigma(\epsilon_{\varphi})$ **(counts/s)**	3030	2653	2578	2139	1788	1826
Avg. Runtime (s)	7.524	12.95	34.66	76.57	1131	4137
Std. Dev. Runtime (s)	0.5937	1.147	1.918	5.964	168.9	666.6

C. Particle Count Study

The occasional poor solutions in the previous Monte Carlo study are caused by the use of a relatively low number of particles. A study was performed to examine how particle count affects errors in the solution. In these cases, the same 150 simulations (30 source configurations with 5 random seeds) were run with obstacles using the DPF only. These simulations were run with varying values of N, with results shown in Table III and Fig. 10. In Fig. 10, the solid line indicates the mean. As expected, higher N leads to improved accuracy. The rate at which $\hat{r}$ is correct increases monotonically with N. This in turn reduces the number of "failed" solutions (cases exhibiting unusually high position errors) as N grows, as seen in Fig. 10. However, increasing N shows diminishing returns in terms of improving accuracy, especially when measured against the runtime penalty when using a larger particle set as shown in Table III. These diminishing accuracy returns are due to measurement sparsity. With sparse measurements, there can be many parameter vectors with high likelihoods between which the PF cannot distinguish. Therefore, increasing the number of particles without increasing the density of measurements (particularly in certain areas) will not necessarily produce improved estimates. An illustrative example of this phenomenon can be seen in Fig. 4. In the top right of the three subplots, two sources lie close together inside of an obstacle. The radiation field contours are nearly circular outside of that obstacle. Without a measurement inside the obstacle between these sources, the addition of more particles in and of itself may not enable the PF to determine that there are two sources inside the obstacle.

D. Dynamic Particle Count Adjustment Algorithm Study

The prior section illustrates the well-known tradeoff between the desire to use a high number of particles for improved accuracy, and the desire to use a low number of particles to reduce runtime. For a practical system that is expected to operate in a range of scenarios with different spatial limits, obstacle densities, etc., it is nearly impossible to choose a suitable value of N that will work well in all scenarios. The DPCA algorithm described in Section III offers a solution to adapt the particle count in real-time, thereby balancing the accuracy-runtime tradeoff automatically for a specific scenario.

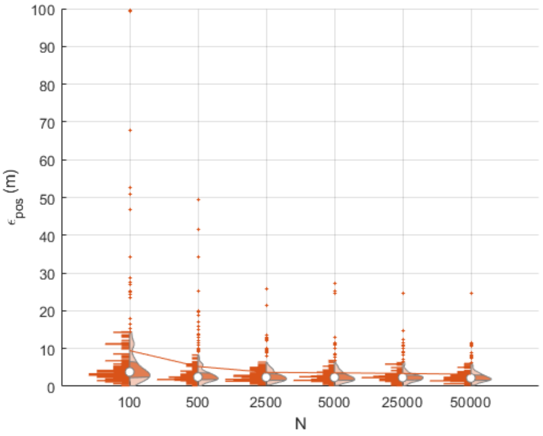

Fig. 10. Position Error for DPFs with Different Particle Counts.

To investigate the performance of the DPCA, the Discrete Particle Filter (DPF, which uses a fixed N) is augmented with the DPCA as depicted in Fig. 1. This algorithm is denoted as the Dynamic Discrete Particle Filter, or DDPF. Monte Carlo simulations were performed with the DPF and DDPF using the same methodology discussed in Section V-B. However, to study the effect of the number of sources, each Monte Carlo simulation was repeated for different values of r_{max} ranging from 2 to 8. Increasing r_{max} tends to challenge the algorithm as the particle filter must search a solution space that grows exponentially with the number of sources. Note that the initial particle count used by the DDPF, N_0, was set to 5,000 in each simulation.

The results of these Monte Carlo studies are shown in Figs. 11-13 and Table IV. In each figure, the solid lines indicates the mean values. The bolded numbers in Table IV show the superior value for a particular case, compared between the DPF and DDPF. In Table IV, it is evident that the DDPF only achieves a small improvement in mean position accuracy compared to the DPF, which is already quite accurate. Furthermore, the DDPF reduces the average runtime compared to the DPF for $r_{\text{max}} < 7$, but only by a small amount, and it even shows higher mean runtimes for $r_{\text{max}} = 7$ and $r_{\text{max}} = 8$. However, the main advantage of the DDPF is its ability to reduce the number of "failure" cases with very high position errors. Figure 11 shows that the DDPF is able to nearly eliminate these failure cases at values of $r_{\text{max}} > 5$ by increasing the size of the particle set adaptively as measurements are gathered. This allows the filter to operate at maximum efficiency, using large particle sets when necessary to address ambiguity, and small particle sets as uncertainty is reduced. The improvement in the number of failed cases is seen in the 95% upper bounds on position error shown in Table IV (denoted $\mathbf{P}_{95}()$), which are better for the DDPF in all cases of $r_{\text{max}} > 2$.

The reason for the DDPF's reduction of failure cases is encapsulated in Fig. 12. This figure shows the mean of the set of negative log likelihoods $\mathcal{Q}$ over all update steps. The negative log likelihood of the particle set is the metric used to increase or decrease N as described in Algorithm 2. A low average log likelihood (or high negative

TABLE IV

DPF AND DDPF COMPARED FOR VARYING r_{MAX}.

Particle Filter	DPF ($N = 5,000$)						
$r_{\max}$	**2**	**3**	**4**	**5**	**6**	**7**	**8**
$\hat{r}$ **Correct (%)**	**98.67**	**92.67**	**85.33**	76	60.67	54.67	47.33
$\mu(\epsilon_{\text{pos}})$ **(m)**	**2.509**	**3.579**	4.492	5.514	6.45	6.593	8.373
$\sigma(\epsilon_{\text{pos}})$ **(m)**	**3.445**	3.87	3.465	4.069	4.943	5.049	5.774
$P_{95}(\epsilon_{\text{pos}})$ **(m)**	**5.44**	11.08	11.41	13.9	15.88	13.22	20.3
$\mu(\epsilon_{\varphi})$ **(counts/s)**	387.1	474.4	510.8	527.6	**616.4**	**491.4**	**1064**
$\sigma(\epsilon_{\varphi})$ **(counts/s)**	**1688**	**2139**	2462	**2628**	2770	3142	3875
Mean Runtime (s)	75.46	76.57	77.87	79.01	79.95	**82.82**	**83.05**
Median Runtime (s)	77.25	78	79.1	79.9	81.15	83.99	83.81

Particle Filter	DDPF ($N_0 = 5,000$)						
$r_{\max}$	**2**	**3**	**4**	**5**	**6**	**7**	**8**
$\hat{r}$ **Correct (%)**	**98.67**	**92.67**	84	**76.67**	**63.33**	**62.67**	**49.33**
$\mu(\epsilon_{\text{pos}})$ **(m)**	2.636	3.76	**4.454**	**5.33**	**5.92**	**6.091**	**7.471**
$\sigma(\epsilon_{\text{pos}})$ **(m)**	2.846	**3.064**	**3.035**	**3.484**	**3.576**	**3.519**	**4.038**
$P_{95}(\epsilon_{\text{pos}})$ **(m)**	6.115	**10.56**	**10.74**	**12.66**	**12.41**	**12.19**	**14.68**
$\mu(\epsilon_{\varphi})$ **(counts/s)**	**304.7**	**348**	**298.5**	**460.7**	833.2	814.8	1170
$\sigma(\epsilon_{\varphi})$ **(counts/s)**	2090	2229	**2264**	2632	**2648**	**3119**	**3539**
Mean Runtime (s)	**31.68**	**43.41**	**69.26**	**68.47**	**77.63**	110.3	134.1
Median Runtime (s)	**27.07**	**26.95**	**27.39**	**27.98**	**29.11**	**36.3**	**42.11**

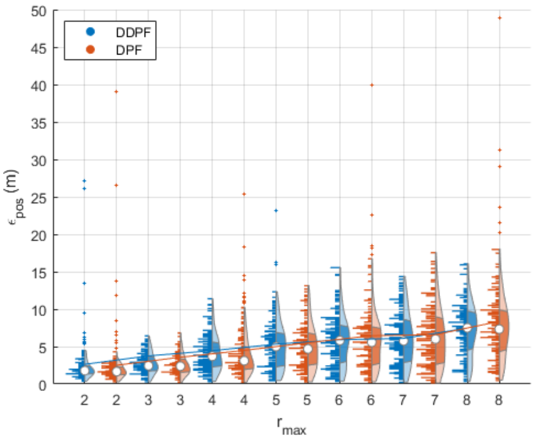

Fig. 11. Monte Carlo Position Errors for DPF and DDPF.

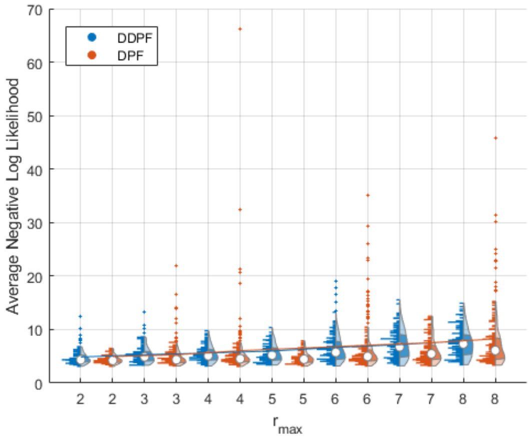

Fig. 12. Mean Negative Log Likelihoods Over All Update Steps for DPF and DDPF.

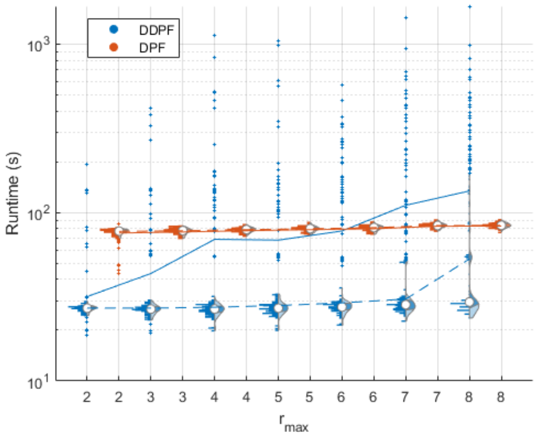

Fig. 13. Monte Carlo Runtimes for DPF and DDPF.

log likelihood) over the particle set means that the particle set poorly matches the measurements that have been obtained. Thus, a high mean negative log likelihood indicates that the particle filter may require more particles, or else will suffer from high estimation errors. Figure 12 shows that the DDPF is much less susceptible to very large mean negative log likelihoods, meaning that the quality of the particle set is in general better than the DPF in many cases. This improvement is achieved by increasing the number of particles when necessary, particularly at high $r_{\max}$.

The DDPF's ability to grow the particle set when necessary has obvious implications for runtime. Figure 13 shows runtime statistics for the DPF and DDPF. In this figure, the dashed line indicates the median values. The DPF runtimes are nearly flat with respect to $r_{\max}$, since runtime depends primarily on N, which is fixed. For the DDPF, most of the runtimes cluster around 20-30 sec, but several outlier cases are evident with much higher runtimes; the frequency of outliers grows with $r_{\max}$. In these cases, the DPCA algorithm increased N substantially at some point during the simulation. As a result, the mean runtime of the DDPF varies as a function of $r_{\max}$ and even exceeds that of the DPF for $r_{\max} > 6$. However, as verified in Table IV, the median runtime of the DDPF is substantially lower than the DPF for all $r_{\max}$. In general, these results show the favorable performance of the DDPF compared to the DPF. In most cases, the DDPF exhibits similar estimation accuracy to the DPF with lower runtimes. In challenging cases, the DDPF achieves much better estimation accuracy than the DPF, albeit at the expense of increased runtime. This tradeoff can be adjusted by tailoring the DPCA tuning parameters in Table I. However, in practice it is expected that longer runtimes would be an acceptable price to pay in order to avoid very large errors in parameter estimates.

E. Uncertain Absorption Parameters Study

In practical scenarios, it may be difficult or impossible to derive accurate absorption coefficients for all obstacles in an environment. It is therefore important to characterize performance of the particle filter in the presence of absorption parameter uncertainty. To test this, a Monte Carlo simulation was performed using the DDPF and the methodology described in Section V-B with $r_{\text{max}} = 3$. In each simulation, however, the absorption coefficients for each obstacle, β_{obs}, were randomized, while the particle filter's attenuation kernel was built with an assumed constant value of $\beta_{\text{obs}} = 10^{-2}$ m^{-1} for all obstacles. This creates a mismatch between the true obstacle absorption coefficient and the assumed value. Random absorption coefficients for each obstacle were generated by first sampling $\nu \sim \mathcal{N}(m, \sigma)$, and then computing $\beta_{\text{obs}} = 10^{-\nu}$. Several Monte Carlo simulations were performed at different values of m and σ, as shown on the x-axis of Fig. 14. The value of m dictates the bias in absorption coefficient estimates, while σ affects the randomness of the coefficients between buildings. The combinations of these values were selected to achieve a several orders-of-magnitude difference between the obstacle's true and assumed absorption parameter. The perturbed absorption parameter for a given obstacle stayed consistent across all 150 trials.

The results of this experiment are shown in Table V. The position and source strength estimates are fairly accurate for the $m = 2$ cases – very similar to the no-uncertainty case. This is presumably because the absorption coefficients do not exhibit any bias with respect to the assumed values. The exception is for the $m = 2$, $\sigma = 0.5$ case, where the large random perturbations in β_{obs} clearly reduce accuracy. The worst-performing cases were those for which $m = 1$. In these cases, the large bias led to numerous failed solutions. This is expected as the effects of β_{obs} decrease geometrically as its exponent ν increases linearly. Figure 14 shows the source position errors in each simulation for the different randomization parameters. The $m = 1$, $\sigma = 1$ case clearly exhibits numerous failed cases, but the other configurations generally show favorable accuracy. Note that in Table V, mean runtime serves as a proxy for algorithm performance since the DDPF increases N substantially in the failed cases to try to obtain accurate estimates.

The trend in source strength estimation error as a function of uncertainty in β_{obs} is shown in Fig. 15. This figure shows the mean source strength estimation error for each case, with the failed cases removed for each (m, σ) combination. In general, the source strength estimate undershoots when the true obstacle absorption is larger than assumed and overshoots when the attenuation is higher than assumed. This is expected—when the true obstacles attenuate more than the particle filter assumes, the particle filter posits that the sources must simply be weaker, and vice versa.

TABLE V

DDPF Performance with Obstacle Attenuation Parameter Uncertainty.

Particle Filter	**DDPF** ($N_0 = 5,000$)							
Obstacle m	none	2	1	1	2	2	3	3
Obstacle σ	N/A	0	1	.5	1	.5	1	.5
$\hat{r}$ **Correct (%)**	92.67	92.67	88.67	98	96	94	91.33	90.67
$\mu(\epsilon_{\text{pos}})$ **(m)**	3.51	3.76	16.03	5.736	4	3.576	3.801	3.968
$\sigma(\epsilon_{\text{pos}})$ **(m)**	2.726	3.064	26.81	8.758	2.52	2.637	2.779	2.969
$\mu(\epsilon_{\varphi})$ **(counts/s)**	730.8	348	1673	2364	1513	288	567.9	497.1
$\sigma(\epsilon_{\varphi})$ **(counts/s)**	1925	2229	9068	1584	1874	2079	2245	2738
Mean Runtime (s)	49.7	43.41	2297	327	234.8	50.12	48.26	53.32

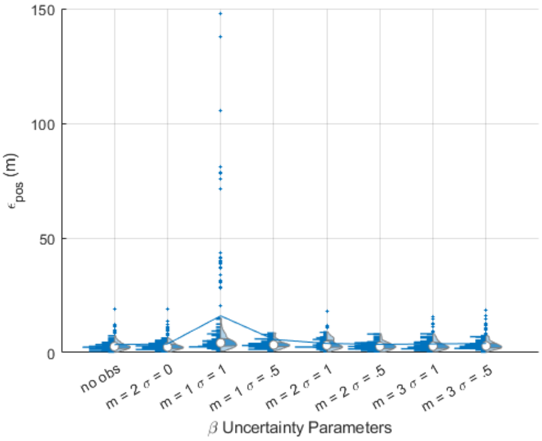

Fig. 14. Position Estimation Error for Varying Levels of Obstacle Attenuation Parameter Uncertainty.

VI. Conclusions

A novel particle filter algorithm has been proposed for radiological source term estimation. In light of the complexity of the likelihood calculations, the continuous parameter estimation problem is reduced in the spatial dimensions to a discrete estimation problem. This discretization allows measurements for each candidate source-measurement location pair to be simulated offline for a reference source strength using a radiation transport model of any desired complexity. The algorithm also includes a dynamic component that adjusts the number of particles in real-time to achieve a suitable balance between estimation accuracy and filter runtime.

Simulation results illustrate that the particle filter is capable of producing accurate source term estimates – even in obstacle-rich settings with relatively large numbers of sources (up to eight). This improves upon algorithms in prior work that have been tested only in scenarios with a handful of sources. Furthermore, Monte Carlo simulation results show that the proposed particle filter is reasonably robust to uncertainty in absorption coefficients, although further investigation is warranted using more detailed obstacle and environmental models. The proposed methodology may be applied to other estimation problems outside of radiological applications and may be advantageous whenever likelihood calculations involve complex models that are better suited for offline rather than online simulation.

VII. Acknowledgements

This material is based upon work supported by the Department of Energy / National Nuclear Security Administration under Award Number(s) DE-NA0003921. This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any

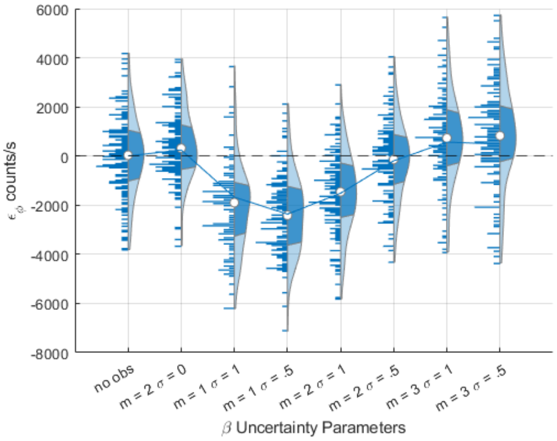

Fig. 15. Source Strength Estimation Error for Varying Levels of Obstacle Attenuation Parameter Uncertainty.

legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

References

[1] B. K. Sovacool, "A critical evaluation of nuclear power and renewable electricity in asia," *Journal of Contemporary Asia*, vol. 40, no. 3, pp. 369–400, 2010.

[2] G. I. Balatsky and W. R. Severe, "Illicit trafficking of radioactive and nuclear materials," in *Nuclear Safeguards, Security, and Nonproliferation*, Elsevier, 2019, pp. 357–387.

[3] T. S. Carless, K. Redus, and R. Dryden, "Estimating nuclear proliferation and security risks in emerging markets using bayesian belief networks," *Energy Policy*, vol. 159, p. 112 549, 2021.

[4] J. Abouaf, "Trial by fire: Teleoperated robot targets chernobyl," *IEEE Computer Graphics and Applications*, vol. 18, no. 4, pp. 10–14, 1998.

[5] Y. Sanada, T. Sugita, Y. Nishizawa, A. Kondo, and T. Torii, "The aerial radiation monitoring in japan after the fukushima daiichi nuclear power plant accident," *Prog. Nucl. Sci. Technol*, vol. 4, no. 7, 2014.

[6] S. Kawatsuma, M. Fukushima, and T. Okada, "Emergency response by robots to fukushima-daiichi accident: Summary and lessons learned," *Industrial Robot: An International Journal*, 2012.

[7] K. Ohno, S. Kawatsuma, T. Okada, E. Takeuchi, K. Higashi, and S. Tadokoro, "Robotic control vehicle for measuring radiation in fukushima daiichi nuclear power plant," in *2011 IEEE International Symposium on Safety, Security, and Rescue Robotics*, IEEE, 2011, pp. 38–43.

[8] D. T. Connor, K. Wood, P. G. Martin, *et al.*, "Radiological mapping of post-disaster nuclear environments using fixed-wing unmanned aerial systems: A study from chornobyl," *Frontiers in Robotics and AI*, p. 149, 2020.

[9] A. West, I. Tsitsimpelis, M. Licata, *et al.*, "Use of gaussian process regression for radiation mapping of a nuclear reactor with a mobile robot," *Scientific Reports*, vol. 11, no. 1, pp. 1–11, 2021.

[10] P. G. Martin, S. Kwong, N. T. Smith, *et al.*, "3d unmanned aerial vehicle radiation mapping for assessing contaminant distribution and mobility," *International journal of applied earth observation and geoinformation*, vol. 52, pp. 12–19, 2016.

[11] M. Hutchinson, H. Oh, and W.-H. Chen, "A review of source term estimation methods for atmospheric dispersion events using static or mobile sensors," *Information Fusion*, vol. 36, pp. 130–148, 2017.

[12] L. Delle Monache, J. K. Lundquist, B. Kosović, *et al.*, "Bayesian inference and markov chain monte carlo sampling to reconstruct a contaminant source on a continental scale," *Journal of Applied Meteorology and Climatology*, vol. 47, no. 10, pp. 2600–2613, 2008.

[13] F. K. Chow, B. Kosović, and S. Chan, "Source inversion for contaminant plume dispersion in urban environments using building-resolving simulations," *Journal of applied meteorology and climatology*, vol. 47, no. 6, pp. 1553–1572, 2008.

[14] G. Katata, M. Chino, T. Kobayashi, *et al.*, "Detailed source term estimation of the atmospheric release for the fukushima daiichi nuclear power station accident by coupling simulations of an atmospheric dispersion model with an improved deposition scheme and oceanic dispersion model," *Atmospheric chemistry and physics*, vol. 15, no. 2, pp. 1029–1070, 2015.

[15] J. W. Howse, L. O. Ticknor, and K. R. Muske, "Least squares estimation techniques for position tracking of radioactive sources," *Automatica*, vol. 37, no. 11, pp. 1727–1737, 2001.

[16] A. Gunatilaka, B. Ristic, and R. Gailis, "On localisation of a radiological point source," in *2007 Information, Decision and Control*, IEEE, 2007, pp. 236–241.

[17] N. S. Rao, S. Sen, N. J. Prins, *et al.*, "Network algorithms for detection of radiation sources," *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, vol. 784, pp. 326–331, 2015.

[18] N. Pinkam, A. Elibol, and N. Y. Chong, "Informative mobile robot exploration for radiation source localization with a particle filter," in *2020 Fourth IEEE International Conference on Robotic Computing (IRC)*, IEEE, 2020, pp. 107–112.

[19] M. Hutchinson, C. Liu, and W.-H. Chen, "Information-based search for an atmospheric release using a mobile robot: Algorithm and experiments," *IEEE Transactions on Control Systems Technology*, vol. 27, no. 6, pp. 2388–2402, 2018.

[20] M. Hutchinson, C. Liu, P. Thomas, and W.-H. Chen, "Unmanned aerial vehicle-based hazardous materials response: Information-theoretic hazardous source search and reconstruction," *IEEE Robotics & Automation Magazine*, vol. 27, no. 3, pp. 108–119, 2019.

[21] B. Ristic and A. Gunatilaka, "Information driven localisation of a radiological point source," *Information fusion*, vol. 9, no. 2, pp. 317–326, 2008.

[22] A. Fathi and S. F. Masoudi, "Lost gamma source detection algorithm based on convolutional neural network," *Nuclear Engineering and Technology*, vol. 53, no. 11, pp. 3764–3771, 2021.

[23] P. Gong, X.-B. Tang, X. Huang, *et al.*, "Locating lost radioactive sources using a uav radiation monitoring system," *Applied radiation and isotopes*, vol. 150, pp. 1–13, 2019.

[24] J. Peterson, W. Li, B. Cesar-Tondreau, *et al.*, "Experiments in unmanned aerial vehicle/unmanned ground vehicle radiation search," *Journal of Field Robotics*, vol. 36, no. 4, pp. 818–845, 2019.

[25] J.-C. Chin, D. K. Yau, and N. S. Rao, "Efficient and robust localization of multiple radiation sources in complex environments," in *2011 31st International Conference on Distributed Computing Systems*, IEEE, 2011, pp. 780–789.

[26] Z. Cook, "Radiative contour mapping using uas swarm," Ph.D. dissertation, University of Nevada, Las Vegas, 2017.

[27] Z. Cook, M. Kazemeini, A. Barzilov, and W. Yim, "Low-altitude contour mapping of radiation fields using uas swarm," *Intelligent Service Robotics*, vol. 12, no. 3, pp. 219–230, 2019.

[28] J. R. Vavrek, D. Hellfeld, M. S. Bandstra, *et al.*, "Reconstructing the position and intensity of multiple gamma-ray point sources with a sparse parametric algorithm," *IEEE Transactions on Nuclear Science*, vol. 67, no. 11, pp. 2421–2430, 2020.

[29] M. Bandstra, D. Hellfeld, J. Vavrek, *et al.*, "Improved gamma-ray point source quantification in three dimensions by modeling attenuation in the scene," *IEEE Transactions on Nuclear Science*, vol. 68, no. 11, pp. 2637–2646, 2021.

[30] R. B. Anderson, M. Pryor, A. Abeyta, and S. Landsberger, "Mobile robotic radiation surveying with recursive bayesian estimation and

attenuation modeling," *IEEE Transactions on Automation Science and Engineering*, 2020.

[31] B. Ristic, M. Morelande, and A. Gunatilaka, "Information driven search for point sources of gamma radiation," *Signal Processing*, vol. 90, no. 4, pp. 1225–1239, 2010.

[32] P. Wilson, R. Feder, U. Fischer, *et al.*, "State-of-the-art 3-d radiation transport methods for fusion energy systems," *Fusion Engineering and Design*, vol. 83, no. 7-9, pp. 824–833, 2008.

[33] P. P. Wilson, T. J. Tautges, J. A. Kraftcheck, B. M. Smith, and D. L. Henderson, "Acceleration techniques for the direct use of cad-based geometry in fusion neutronics analysis," *Fusion Engineering and Design*, vol. 85, no. 10-12, pp. 1759–1765, 2010.

[34] V. Elvira, J. Míguez, and P. M. Djurić, "Adapting the number of particles in sequential monte carlo methods through an online scheme for convergence assessment," *IEEE Transactions on Signal Processing*, vol. 65, no. 7, pp. 1781–1794, 2016.

[35] A. Martin and S. Harbison, "An introduction to radiation protection," Environmental Safety Group, Associated Nuclear Services, Epsom, Tech. Rep., 1986.

[36] J. R. Bourne, M. N. Goodell, X. He, J. A. Steiner, and K. K. Leang, "Decentralized multi-agent information-theoretic control for target estimation and localization: Finding gas leaks," *The International Journal of Robotics Research*, vol. 39, no. 13, pp. 1525–1548, 2020.

[37] H. Friedman, "Geiger counter tubes," *Proceedings of the IRE*, vol. 37, no. 7, pp. 791–808, 1949.

[38] T. Nicholas, *Measurement and detection of radiation*, 1995.

[39] G. Kitagawa, "Monte carlo filter and smoother for non-gaussian non-linear state space models," *Journal of computational and graphical statistics*, vol. 5, no. 1, pp. 1–25, 1996.

[40] J. D. Hol, T. B. Schon, and F. Gustafsson, "On resampling algorithms for particle filters," in *2006 IEEE nonlinear statistical signal processing workshop*, IEEE, 2006, pp. 79–82.

[41] F. Daum and J. Huang, "Particle degeneracy: Root cause and solution," in *Signal Processing, Sensor Fusion, and Target Recognition XX*, SPIE, vol. 8050, 2011, pp. 367–377.

[42] F. Daum and J. Huang, "Particle flow for nonlinear filters with log-homotopy," in *Signal and Data Processing of Small Targets 2008*, International Society for Optics and Photonics, vol. 6969, 2008, p. 696 918.

[43] N. Oudjane and C. Musso, "Progressive correction for regularized particle filters," in *Proceedings of the Third International Conference on Information Fusion*, IEEE, vol. 2, 2000, THB2–10.

[44] T. Hanselmann, Y. Zhang, M. Morelande, *et al.*, "Self-localization in wireless sensor networks using particle filtering with progressive correction," in *2010 5th International ICST Conference on Communications and Networking in China*, IEEE, 2010, pp. 1–6.

[45] OpenStreetMap contributors, *Planet dump retrieved from https://planet.osm.org*, `https://www.openstreetmap.org`, 2017.

[46] H. Choset and P. Pignon, "Coverage path planning: The boustrophedon cellular decomposition," in *Field and service robotics*, Springer, 1998, pp. 203–209.