

Nuclear threat detection with mobile distributed sensor networks

Dorit S. Hochbaum · Barak Fishbain

Published online: 3 November 2009

© The Author(s) 2009. This article is published with open access at Springerlink.com

Abstract The ability to track illicit radioactive source in an urban environment is critical in national security applications. To this end, two modes of operation are common: positioning individual portal monitors, and deploying a network of distributed sensors. We address here the use of multiple detectors, mounted on moving vehicles, for the purpose of detecting nuclear threats. An example scenario is that of multiple taxi cabs each carrying a detector. The detectors' positions are known in real-time as these are continuously reported from GPS data. The level of detected risk is then reported from each detector at each position. The problem is to delineate the presence of a potentially dangerous source and its approximate location by identifying a small area that has an elevated concentration of reported risk. This problem of using spatially deployed mobile detector networks to identify and locate risks is modeled and formulated here. The problem is shown to be solvable in polynomial time and with a combinatorial network flow algorithm. The efficiency of the algorithm enable its use in real time, and in areas containing a large number of deployed detectors. A simulation study, that takes into account false-positive and false-negatives reports from individual sensors, demonstrates the effectiveness of the algorithm in using the sensor network's detection capabilities.

Keywords Nuclear threat detection · Network flow · Parametric cut · Counter-terrorism · Discrete event simulation

1 Introduction

When considering the risks to our vulnerable populations posed by potential terrorism, the nuclear threat is never far from center stage. Although weapons-grade nuclear materials

CBET-0736232 and NSF award No. DMI-0620677.

D.S. Hochbaum · B. Fishbain (✉)
University of California, Berkeley, CA 94720, USA
e-mail: barak@berkeley.edu

D.S. Hochbaum
e-mail: hochbaum@ieor.berkeley.edu

are heavily guarded, a rather plausible scenario involves the detonation of a very simple Radiological Dispersion Device (RDD) which is capable of broadcasting non-fissile but highly radioactive particles over a densely populated area. In most cases, such a device and its payload must be transported to a target destination. Detection and intervention is the final defense in such a situation.

Detecting nuclear threats is a challenging problem under any circumstances. Distributed Sensor Networks (DSN) are frequently considered in support of this goal (Brennan et al. 2004, 2005; Cunningham 1995; Nemzek et al. 2004; Stephens and Peurrung 2004). Nemzek et al. examine the signal-to-noise behavior that arises in the simple combination of data from networked radiation sensors and conclude that, for real-life scenarios, the individual portal monitors may be equivalent to, or better than, networked sensors. Brennan et al. (2004, 2005) exploit Bayesian methods for radiation detection with distributed sensor networks. Bayesian methods for RDD parameter estimation exhibit a computational complexity exponential in the number of parameters estimated (Nemzek et al. 2004), limiting their implementability and fostering concerns that their specialization for detection might also be too computationally intensive to ever be useful (Brennan et al. 2005). Additionally, the methods presented in the above publications are based on stationary fixed sensors and apply strict constraints on the RDD's velocity and on the environmental conditions. Naturally, this makes them impractical for use in real-life scenarios. Moreover, these methods do not directly lead to improved performance in terms of detection efficiency and reduction in false detection rate (Stephens and Peurrung 2004).

With recent technology it has become operational and cost-effective for DSN to be mounted on vehicles in public service. Sodium Iodide detectors are currently deployed on vehicles such as police cars, fire trucks, trains, buses or even taxi cabs. Thus, creating mobile sensor networks, in which, both the RDD and the sensors may be in motion. The position of each detector is known at any point in time from GPS information transmitted to a central control data processing facility. This makes the detection task even more challenging when the relative positions of the detector and the RDD, if exists, are unknown. The sensitivity of the detectors is diminishing with distance from the RDD, thus their geographic position impacts the reliability of their reporting. Further, a detector may fail to detect correctly an existing threat (false-negative), or report an alert on the existence of a RDD when there is none (false-positive). The likelihood of false reports is diminished and their effect is mitigated when relying on reports from several independent sensors.

We are interested in delivering a decision support system that will delineate a region of elevated risk of the existence of nuclear threat that requires taking further means. These means might involve deploying a task team and possibly isolating the region. Due to the expense and disruption of such means, the decision support system should be highly robust and reliable.

A polynomial time algorithm, for delineating the presence of a potentially dangerous RDD and its approximate location, using mobile sensor networks, has been presented by Hochbaum in the proceedings of the 11th INFORMS Computing Society (ICS) Conference. The suggested algorithm addresses the threat detection as the *concentrated alert problem*, combining two goals: One goal is to identify a small region; another goal is to have large number of alerts, or high concentration of alerts in the region. These two goals are potentially conflicting—focusing on a large number of alerts within an area is likely to result in the entire region; on the other hand focusing on concentration alone would result in a single block of the area containing the highest level of reported alert, thus disregarding information provided by other detectors in the adjacent area. The model is then to identify, at every period of time, a region which relatively to its size has high concentration of alerts. The two goals

are balanced by an appropriate selection of weights. The algorithm, presented in Hochbaum (2009) is referred to as CA (Concentrated Alert) algorithm.

In this paper we extend the discussion in Hochbaum (2009). The results here constitute a significant extension in implementing the algorithm and running simulation results. This is achieved through providing a discussion of false alarm probabilities as a function of the number of detectors per block; by presenting a simulation software and by suggesting methods of aggregating the results of the optimization algorithm over time. The latter allows for a more accurate and robust detection.

The simulation software, presented here, takes into account the nuclear shielding effects in a densely built urban environment. The simulation allows to evaluate the fundamental limitations of distributed multi-sensor network and its ability to cope with a subset of faulty sensors. The simulation provides evidence as to the quality of the performance of the algorithm in correctly identifying danger zones, or mobile RDDs if exist.

The rest of the paper is organized as follows: Sect. 2 provides the formalism for describing the CA and shows how the problem is solvable in polynomial time as a minimum s, t -cut problem on an associated graph. Section 3 provides the theoretical analysis of the probability for misdetection as a function of the density of randomly distributed set of sensors. Section 4 describes the various aspects of the simulation software, and Sects. 5 and 6 present the decision support system that aggregates the CA algorithm's output over time. Section 7 details how the algorithm handles mobile RDD. The conclusions are given in Sect. 8.

2 Solving the concentrated alert problem

2.1 Notation

We introduce here graph theoretic notation to be used in formulating the problem. Without loss of generality, the region where the detectors are deployed, is assumed to be a rectangular area subdivided in grid squares. These will be small enough to contain geographical area which can be effectively contained in a case of threat detection, a single street block for example. Let V be the collection of positions (blocks) in the area considered.

We construct a directed graph G with the set of nodes V corresponding to the set of grid blocks. As each grid point corresponds to a geographical area, we define adjacent grid points whose corresponding areas share a boundary. For each pair of adjacent blocks, i and j , we add a pair of directed arcs, (i, j) and (j, i) , each with capacity 1. These arcs are referred to as the "adjacency" arcs of G , and denoted by A_a . We denote in general the capacity of an arc (i, j) by u_{ij} . Let $S \subset V$ be the set of blocks of a selected sub-region. For each adjacent pair of grid points, if one is within the region S and the other outside, the added length to the boundary, the perimeter of S , is 1.

We measure the *size* of the area delineated by S , by the length of its boundary, counted as the number of grid block sides that separate S from $\bar{S} = V \setminus S$. The length of the boundary of a subset of grid points S is then $\sum_{i \in S, j \in \bar{S}, (i, j) \in A_a} u_{ij} = \sum_{i \in S, j \in \bar{S}, (i, j) \in A_a} 1$, since the set A_a contains arcs of capacity $u_{ij} = 1$. This length is equal to $C(S, \bar{S}) = |\{(i, j) | i \in S, j \in \bar{S}\}|$. Note that there is no requirement that the set S is contiguous. It can be formed of several components, each disconnected from the rest.

Let G_{st} be a graph (V_{st}, A_{st}) , where $V_{st} = V \cup \{s, t\}$ and $A_{st} = A \cup A_s \cup A_t$ in which A_s and A_t are the source-adjacent and sink-adjacent arcs respectively. A flow vector $f = \{f_{ij}\}_{(i, j) \in A_{st}}$ is said to be *feasible* if it satisfies:

1. *Flow balance constraints*: for each $j \in V$, $\sum_{(i,j) \in A_{st}} f_{ij} = \sum_{(j,k) \in A_{st}} f_{jk}$ (i.e., $\text{inflow}(j) = \text{outflow}(j)$)
2. *Capacity constraints*: the flow value is between the lower bound and upper bound capacity of the arc, i.e., $0 \leq f_{ij} \leq u_{ij}$

A *maximum flow* is a feasible flow, f^* , that maximizes the flow out of the source (or into the sink), called the *value of the flow*. The value of the maximum flow is $\sum_{(s,i) \in A_s} f_{si}^*$. An s, t cut in G_{st} (or *cut* for short) is a partition of V_{st} to $(S \cup \{s\}, T \cup \{t\})$. The capacity of the cut is $C(S \cup \{s\}, T \cup \{t\})$. The minimum s, t cut is the cut of minimum capacity, referred to here as *min-cut*. It is well known (Ford and Fulkerson 1956) that the maximum value of the flow is equal to the capacity of the min-cut.

2.2 Formulating the objective function

Since our objective involves multiple goals, we address these by setting a trade-off between the importance of the short boundary versus the high concentration of alerts. One way of trading these off is by minimizing a ratio function—of the length of the boundary divided by the concentration of alerts in the region. Another, used here, is to optimize a weighted combination of the goals.

To formalize the goal of “small area” we define an area to be of small size if it is enclosed by a “short” boundary. The boundary of an area is then the number of edges that separate in-region from out-region, or the rectilinear length of the boundary. In the graph $G = (V, A_a)$ the length of the boundary of a set $S \subset V$ is $C(S, \bar{S})$. Prior to proceeding, we note that this definition of length needs certain tweaking. Let the set of boundary blocks of the entire area considered be denoted by B . With the definition of the set of arcs A_a , the selection of any subset of B reduces the defined size of the region. For example, if the selected region is all of V then the length of the circumference $C(V, \emptyset)$ is equal to 0. To prevent that, we add a set of arcs A_B from the boundary nodes to an imaginary point in space. This will be quantified in a manner explained later. We let the corner grid blocks contribute 2 to the length of the boundary, if included in the set. The length of the boundary is thus $C(S, \bar{S}) + |B \cap S|$ where we count the corner blocks twice in B (instead of introducing additional notation).

Next we formalized the goal of identifying high concentration of alerts. One indication of the level of alert in an area is the number of alerts at higher than threshold level within this area. In general, each alert could be accompanied by a weight, specifying the confidence with which the alert is sounded. We first consider a “binary” alert scenario, where these weights are either 0 or 1 for all alerts. Then we consider general weights to accommodate varying levels of alerts. Let $D \subseteq V$ be the set of positions occupied by vehicles. Let $D^* \subseteq D$ be the set of positions reporting alerts. Part of our objective is then to identify a subregion of positions containing S so that $|D^* \cap S|$ is maximized.

Following the discussion above, the linearized objective of the concentrated alert (CA) problem is the linear combination of the two objectives, using α and β as a weight for the relative importance of the weights:

$$(CA) \quad \min_{S \subseteq V} \{C(S, \bar{S}) + |B \cap S| - \beta |D^* \cap S| + \alpha |\bar{D}^* \cap S|\} \tag{1}$$

Maximizing the number of alerts within the selected region is an objective that has some pitfalls. For instance, if the region considered, S , contains, in addition to the alerts, also a relatively high number of no-alerts $\bar{D}^* \cap S$, for $\bar{D}^* = D \setminus D^*$, then this should diminish the significance of the alerts in the region. Therefore, we add yet another minimization objective, $\min |\bar{D}^* \cap S|$. This objective is then combined with the length of the boundary objective, as

$\min\{C(S, \bar{S}) + |B \cap S| + \alpha|\bar{D}^* \cap S|\}$ Although, in terms of the model, we do not restrict the value of α , it is reasonable that α should be a small value compared to the contribution of alert positions, as discussed below. If we choose to disregard the number of no-alerts in the region, then this is captured by setting $\alpha = 0$.

A potential problem with this formulation is ignoring the positions that contain no detectors. It is possible that two non-adjacent regions are identified as high risk regions. In this case, the area between those two regions may also contain threat, but get no indication as such since there is no detector there. An example illustrating this is presented, for specific choices of α and β , in Sect. 3.1 and Fig. 2. To address this, we assign a value, denoted by γ , to each vacant grid point, where γ is much smaller than β . The affect of assigning values to vacant positions is also shown in Sect. 3.1 and Fig. 2. Depending on users’ considerations, γ can always be set to 0.

Incorporating the use of the weight γ for vacant positions, the objective of the CA problem becomes:

$$(CA) \quad \min_{S \subseteq V} \{C(S, \bar{S}) + |B \cap S| - \beta|D^* \cap S| + \alpha|\bar{D}^* \cap S| - \gamma|S \setminus D|\} \quad (2)$$

2.3 The weighted version of the concentrated alert problem

The information captured by a detector is a spectrum of gamma ray emissions recording the frequency at each energy level. As such this is not scalar-valued information which may be too ambiguous to translate to a simple binary statement of the form of alert or no-alert. The analysis of the detected energies spectrum therefore presents a challenge. The analysis process is currently under development using advanced data mining techniques (by e.g. the DONUTS research group at UC Berkeley (2009)). Therefore, one can refine the binary alert levels to continuous levels, or to a collection of discrete levels. We use here the latter and define a threshold level, beyond which there are several alert categories and below it, several no-alert categories. The alert profile, if above threshold, transmitted from a detector in position i is mapped to a weight value p_i that is monotone increasing with the increased confidence in the significance of the alert information. If the alert profile is under threshold that it is mapped to q_i which increases monotonically with the confidence in the no-alert reporting.

The modified *weighted concentrated alert* problem is then to find a sub-region S , optimizing the function,

$$(WCA) \quad \min_{S \subseteq V} C(S, \bar{S}) + |B \cap S| + \alpha \sum_{i \in \bar{D}^* \cap S} q_i - \beta \sum_{i \in D^* \cap S} p_i - \gamma|S \setminus D| \quad (3)$$

2.4 Constructing the graph

Let the grid points be represented by a collection of nodes V of a directed graph $G = (V, A)$, where each point is represented by a node. The set of nodes is appended by a *source* dummy node s and *sink* dummy node t .

The definition of the adjacency arcs, A_a , as given in Sect. 2.1, allows the adjacency to be of any form. To this end both *4-neighbors* adjacency or *8-neighbors* adjacency can be used. The perimeter’s length depends the definition of the neighborhood. In our illustrations and experiments we used 4-neighborhood. If 8-neighborhood is used the perimeter’s length is then roughly 3 times longer and α and β values should be changed accordingly to get the same output.

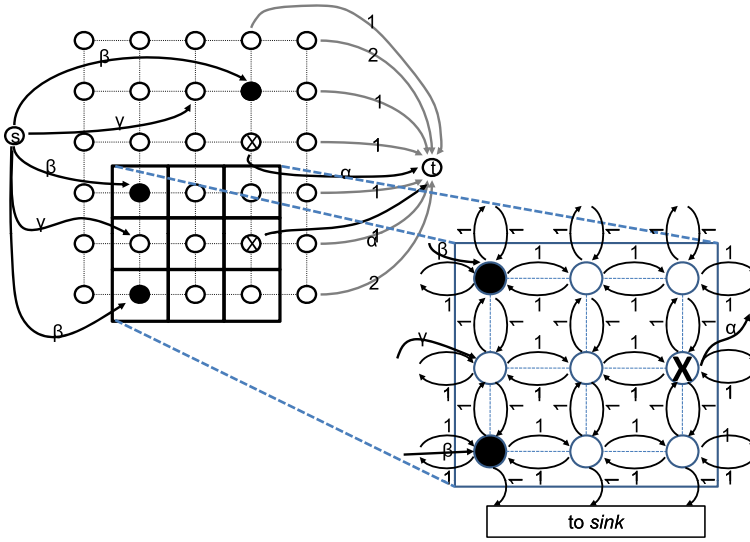


Fig. 1 The weighted graph G_{st}^W

We connect the set of arcs A_B , representing the boundary of the monitored region, to the sink t with capacities of 1. For corner blocks, we add an arc to the sink node with capacity 2. Each position i that contains an alert in D^* is set to be adjacent to s with a directed arc of capacity βp_i . Each position i that contains no-alert, in \bar{D}^* , is set to be adjacent to t with a directed arc of capacity αq_i . We denote these sets of arcs by A_β and A_α respectively.

We therefore constructed a directed s, t graph $G_{st} = (V_{st}, A)$, where $V_{st} = V \cup \{s, t\}$ and $A = A_a \cup A_B \cup A_\beta \cup A_\alpha$. The construction of the graph is illustrated in Fig. 1 where an alert position is indicated by a solid circle, and a no-alert position by a crossed circle.

We now have a graph $G = (V \cup \{s, t\}, A)$ with arc capacities u_{ij} for arc $(i, j) \in A$, on which we define the minimum s, t -cut problem and show that solving it provides the optimal solution to our WCA problem.

Theorem 1 (Hochbaum 2009) *The source set of the minimum cut in the graph G_{st}^W is the optimal region for problem WCA.*

Proof The full proof and its derivation from the unweighted concentrated alert problem is given in Hochbaum (2009). We include the proof for the weighted case here, for the sake of completeness. Let $(S \cup \{s\}, T \cup \{t\})$ be an s, t -cut in G_{st}^W , of capacity:

$$\begin{aligned}
 & C(S \cup \{s\}, T \cup \{t\}) \\
 &= |B \cap S| + \alpha \sum_{i \in \bar{D}^* \cap S} q_i + \beta \sum_{j \in D^* \cap T} p_j + \gamma |T \setminus D| + \sum_{(i,j) \in A, i \in S, j \in T} 1 \\
 &= |B \cap S| + \alpha \sum_{i \in \bar{D}^* \cap S} q_i + \beta \left(\sum_{i \in V} p_i - \sum_{j \in D^* \cap S} p_j \right) + \gamma (|V \setminus D| - |S \setminus D|) + C(S, T) \\
 &= \beta \sum_{i \in V} p_i + \gamma |V \setminus D| + |B \cap S| + C(S, T) + \alpha \sum_{i \in \bar{D}^* \cap S} q_i - \beta \sum_{j \in D^* \cap S} p_j - \gamma |S \setminus D|
 \end{aligned}$$

Since $\beta \sum_{i \in V} p_i + \gamma |V \setminus D|$ is a constant, the source set of the minimum cut is also minimizing $|B \cap S| + C(S, T) + \alpha \sum_{i \in \bar{D}^* \cap S} q_i - \beta \sum_{j \in D^* \cap S} p_j - \gamma |S \setminus D|$.

We conclude that solving the concentrated alert problem reduces to finding the minimum s, t cut in the graph G_{st}^W . The region we are seeking will then correspond to the source set S of the minimum cut $(S \cup \{s\}, T \cup \{t\})$. □

For finding the minimum s, t cut in G_{st}^W we use the parametric pseudoflow algorithm, (Hochbaum 2008). The source code of the solver we use is available at Chandran and Hochbaum (2009).

3 Error analysis

Any metric for judging the performance of a network of sensors will ultimately reduce to two fundamental parameters. The first of these parameters is the probability that a particular RDD will indeed be detected by the sensor network. This parameter is often called the *detection efficiency*, and is the complement of the *false negative* probability. The second of these parameters is the probability per unit time that a sensor network will incorrectly identify the presence of a RDD that is not actually present. In the interests of clarity this paper will refer to this parameter as the *false positive* probability and quantify it as a probability per measurement interval. Ideally, the detection probability will be high and the false detection probability will be very low. This ensures confident detection of real targets while minimizing the resources consumed responding to false detection events. It should be noted that both parameters are intimately related in that for a specified scenario (transit time, background count rate, RDD strength, etc.) they can both be derived from the selected detection threshold.

3.1 Algorithm’s parameters

To illustrate the considerations in choosing the values of the parameters α, β and γ , we consider first the simpler, binary, case where each sensor can report either an alert or no-alert. The computed region of elevated risk, S , is computed through minimizing Eq. 2, where $\beta = 3.99, \alpha = \beta/2$ and $\gamma = 0.021$. The reason why the value of β is just under 4 is to prevent the generation of regions consisting of singletons of alert positions.

In Fig. 2(a) the set V is a 7×10 grid. The set of three alert positions forms the optimal solution. The optimal region is indicated by darker shade. Notice that in this example there

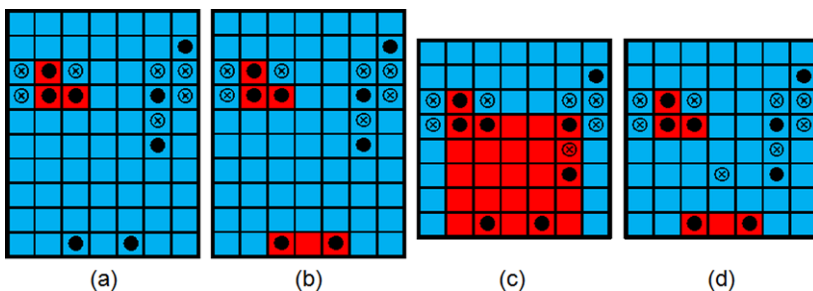


Fig. 2 The effect of introducing γ values to vacant grid positions

are, on row 10, two alert positions separated by an empty position. Although these might indicate an elevated alert status for that area, the vacant grid position rules out selecting this set. The presence of vacant positions therefore should not necessarily be interpreted as diminishing the alert level. As we see in Fig. 2(b), the addition of the γ coefficient indeed changes the optimal solution, and now we have two alert regions, one of which contains an empty position. However, if the two regions are “close”, and in the given configuration they are 3 rows apart, as shown in Fig. 2(c), then the two regions merge into one. In Fig. 2(d) one sees that adding one no-alert position within the region has the effect of separating the two regions. The determination of which values to set and when regions should be consolidated is to be determined by nuclear detection experts and the geographical parameters of the region, as well as the density of the detectors’ distribution in the region.

The following considerations should be taken into account when setting values for β : β reflects the alerts’ concentrations, the ratio between the region’s size (represented by its circumference) and the sum of weighted alerts reported within it. The values of β are associated with the way the weights p_i are set. Suppose we let $p_i = 1$ correspond to a *definite threat*, any value of $\beta > 4$ results in regions consisting of singletons of alert positions. Definite threat refers to a situation where the presence of a nuclear threat is undoubted. For example, when high gamma and neutron radiation levels are measured in the same position. In those cases it is up to system designer to determine whether to generate an alarm based upon a single, possibly faulty, report. In the next set of examples we set $\beta = 4.01$. In this way, every singleton report with a weight higher than 0.998 generates an alarm. There are two cases in which $p_i > 0.998$: (i) when a sensor has detected a *serious threat*, or (ii) one of the sensors is faulty. This choice of parameter ensures that any serious threat reported by a single detector with $p_i > 0.998$, results in alarm. With that, if the region that contains the sensor reporting a definite-threat, happens to contain other sensors reporting all-clear, the algorithm uses this information to diminish the impact of the faulty sensor report.

3.2 False positives probability

In the remainder of this section, the following notations are used:

1. K is the total number of sensors ($K = |D|$).
2. $k_S \leq K$ is the number of sensors occupy an arbitrary region $S \subseteq V$, hence $k_S = |D \cap S|$.
3. Each sensor can be assigned with three levels of alerts: *Definite Threat*, DT , corresponds to $p_i = 1$; *Possible Threat*, PT , corresponds to $p_i = 0.995$; and *All Clear*, AC , where $q_i = 1$.
4. k_S^{DT} , k_S^{PT} and k_S^{AC} are the number of sensors within S that are reporting *definite alert*, *possible alert* and *all-clear* respectively. Note that $k_S^{DT} + k_S^{PT} + k_S^{AC} = k_S$.
5. P_S is the a-priori probability that a sensor occupies S . In the following discussion the sensors’ positions are uniformly distributed, thus $P_S = |S|/|V|$.
6. $P(DT|AC)$, $P(PT|AC)$ and $P(AC|AC)$ are the probabilities of a single detector reporting *definite alert*, *possible alert* and *all-clear*, conditioned on an *all-clear* event. Those probabilities are derived from the detector’s design parameters such as energy resolution, geometrical characteristics and counting interval (Attix 1986).

Evaluating Eq. 3, S , computed by the algorithm, is not an empty set only if:

$$|B \cap S| + C(S, T) + \alpha \sum_{i \in D^* \cap S} q_i - \beta \sum_{j \in D^* \cap S} p_j - \gamma \cdot |S \setminus D| < 0 \tag{4}$$

Using the above notions and noting that $|B \cap S| + C(S, T) = \text{circumference}(S)$, we conclude that $S \neq \emptyset$, hence an alarm is generated, if:

$$\beta \cdot k_{DT} + 0.995 \cdot \beta \cdot k_{PT} + \gamma \cdot (|S| - k) - \alpha \cdot k_{AC} - \text{circumference}(S) > 0 \tag{5}$$

Let us now define $\delta(S, k_S^{DT}, k_S^{PT}, k_S^{AC})$, as:

$$\delta(S, k_S^{DT}, k_S^{PT}, k_S^{AC}) = \begin{cases} 1 & \text{if } (\beta \cdot k_{DT} + 0.995 \cdot \beta \cdot k_{PT} + \gamma \cdot (|S| - k) - \alpha \cdot k_{AC} - \text{circumference}(S) > 0) \\ 0 & \text{otherwise} \end{cases}$$

The probability of an alert in S , given that k_S sensors occupy S , is then given by:

$$P(\text{Alert}|k_S) = \sum_{k_S^{DT}=0}^{k_S} \sum_{k_S^{PT}=0}^{k_S - k_S^{DT}} \sum_{k_S^{AC}=0}^{k_S - k_S^{DT} - k_S^{PT}} \left\{ \frac{k_S!}{k_S^{DT}! k_S^{PT}! k_S^{AC}!} \cdot \delta(S, k_S^{DT}, k_S^{PT}, k_S^{AC}) \cdot P(DT|AC)^{k_S^{DT}} \cdot P(PT|AC)^{k_S^{PT}} \cdot P(AC|AC)^{k_S^{AC}} \right\} \tag{6}$$

Then, given, that $P(k_S|K)$ is the probability that k_S sensors occupy S , given by:

$$P(k_S|K) = \frac{K!}{k_S!(K - k_S)!} \cdot (P_S)^k \cdot (1 - P_S)^{(K - k_S)} \tag{7}$$

The probability of false positive is:

$$P_{\text{False-Positive}} = \sum_{k_S=0}^K P(k_S|K) \cdot P(\text{Alert}|k_S) \tag{8}$$

Figure 3 presents the probability of false detection in a single grid point as a function of sensors' concentration, thus the average number of sensors, per grid point. The probability of error is computed for an increasing number of sensors on a fixed size area for three sets of detectors. The first set of detectors is characterized by $P(DT|AC) = 2\%$ and $P(PT|AC) = 8\%$ (solid line); the second set is characterized by $P(DT|AC) = 4\%$ and $P(PT|AC) = 16\%$ (dashed line), while the third set's parameters are $P(DT|AC) = 8\%$ and $P(PT|AC) = 32\%$ (dash-dot line). For low concentrations, the chance of a sensor actually being in the block in question is small, and therefore no alarm is reported and no false detection occurs. With that, low concentration values, generates high probability of false negatives. This is addressed in Sect. 3.3. When the number of sensors increases, the probability of error increases, till there is enough sensors that other detectors in the same block can conceal the faulty reports. As one can see, for the three sets of sensors, for a certain concentration rate, the derivatives of the probability functions decreases to 0. Meaning that the number of sensor has reached a saturation. The minimum asymptotic error, though, is higher for the second and third sets of sensors, which presents worse detection capabilities. Thus, when the sensor network reaches to a saturation state, the quality of detection can be improved by using a better set of sensors.

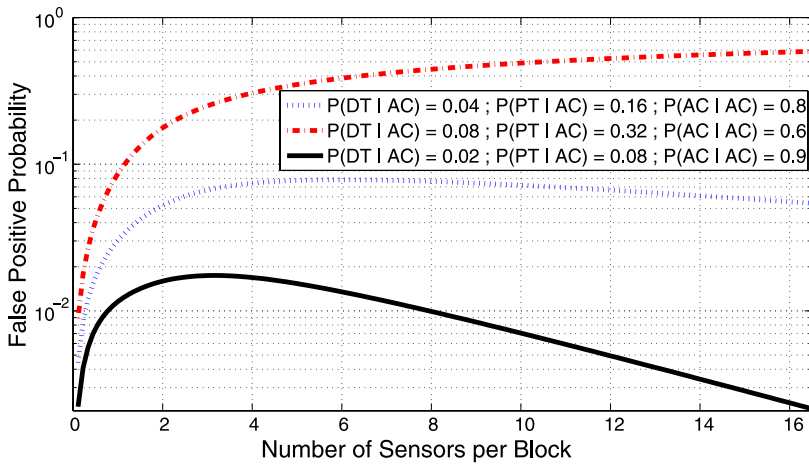


Fig. 3 Probability of false positive vs. the number of sensors per block

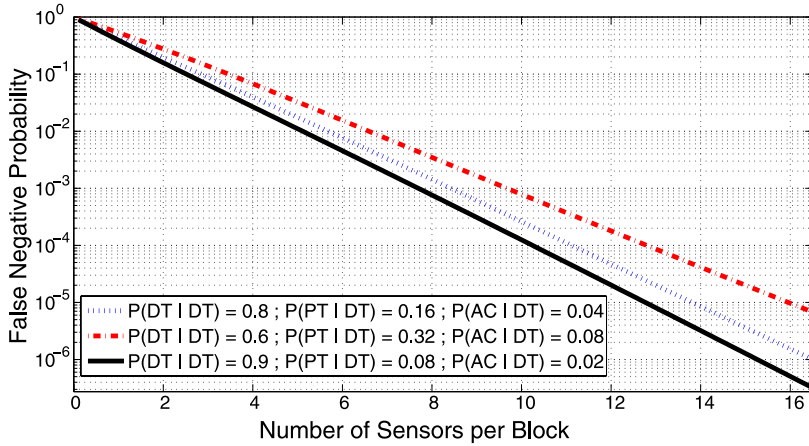


Fig. 4 Probability of false negative vs. the number of sensors per block

3.3 False negatives probability

Similarly to the derivation of Eq. 8, the probability of a false negative is given by:

$$P_{\text{False-Negative}} = \sum_{k=0}^K P(k|K) \cdot P(\text{No-Alert}|k) \tag{9}$$

where $P(k|K)$ is given by Eq. 7 and $P(\text{No-Alert}|k)$ is given by Eq. 6, for all $\{k_{DT}, k_{PT}, k_{AC}\}$ triplets such that:

$$\beta \cdot k_{DT} + 0.995 \cdot \beta \cdot k_{PT} + \gamma \cdot (|S| - k) - \alpha \cdot k_{AC} < \text{circumference}(S) \tag{10}$$

Figure 4 depicts the probability of a false negative in a single grid point region, for different sensor concentration rates, for three sets of detectors. The first set (solid line) is char-

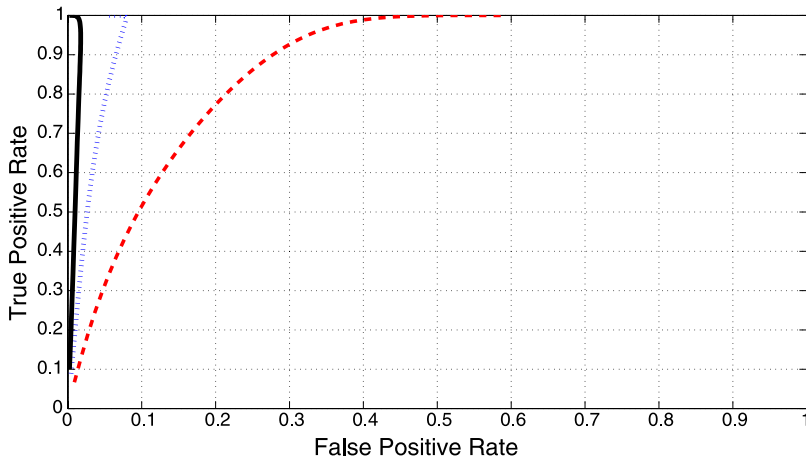


Fig. 5 Detectors' receiver operating characteristic (ROC)

acterized by $P(AC|DT) = 2\%$ and $P(PT|DT) = 8\%$, the second set (dotted line) characteristics are $P(AC|DT) = 4\%$ and $P(PT|DT) = 16\%$ and the last set's characteristics (dash-dot line) are $P(AC|DT) = 8\%$ and $P(PT|DT) = 32\%$. As expected, for the three sets of sensors, low sensor concentrations result in high probability of false negative. When the concentration increases, the error decreases rapidly.

The Receiver Operating Characteristic (ROC) curves are often used to present the detection system's performance. The ROC curve expresses the inspection system's detection rate as a function of the rate of false positives (Fawcett 2006). The corresponding ROC curves of the three detectors are presented in Fig. 5.

3.4 Practical considerations

The decision, whether to generate a system alarm, for a particular risk, is executed by the system once per predefined 'processing period'. In order to come up with a decision, the system integrates all measurements received during this period. To this end, it is possible that, throughout the processing period, the same sensor reports from two different locations, or two different sensors report from the exact same location.

The results in Sects. 3.2 and 3.3 show that a density of 10 sensors per grid point per processing period saturates the system, meaning that adding sensors beyond 10 will not improve the system's performance significantly. Deploying the detectors on taxis in densely populated urban environments, such as Manhattan NYC, will provide sufficient sensors' density, (Schaller 2006). For less populated areas, one can aggregate the sensors' readings over time till a sufficient number of measurements is reached. This solution is somehow less effective when the RDD is mobile, or when we would have a high correlation in the false positive/false negative readings. An example for such scenario is a taxi having a malfunctioning detector. This could undermine the usefulness of the aggregation over time and might require an operator's intervention.

The monitored area (V) might consist of areas with varying taxis' (detectors') density. One way to address this is to use readings over different time windows for each region. For highly populated area the algorithm regards only readings that have been acquired during the last processing period, whereas for less populated areas, measurements from a temporal

window of several processing rounds are used. For example, if an area requires on average 3 periods in order to reach sufficient density, then the readings processed by the algorithm in period t will be the accumulated readings over $t - 2$, $t - 1$ and t . In general, for n time periods' window the readings, for t , is the accumulated samples over $t - (n - 1)$, $t - (n - 2)$, \dots , t . The value of n is preset for each grid block, or may be adaptively adjusted on-line—for each grid block wait until a sufficient number of readings have been acquired.

4 Simulation software

Our simulation models the detection of the presence or the transport of radioactive RDD's in an urban environment by means of mobile sensor networks mounted on public service vehicles, e.g. taxis. The simulation software is using, as subroutine, the pseudoflow algorithm, described in Hochbaum (2008), and its implementation software available at Chandran and Hochbaum (2009). The simulation was written under the following assumptions:

1. The RDD can be moving or stationary.
2. The speed of each moving object, thus the sensors and the RDD, in the case it is being transported, is randomly, uniformly distributed, set to be between 11 and 45 [Miles/Hour]. Meaning that each object, a sensor or the RDD, may have different speeds than the other moving objects.
3. When a vehicle approaches an intersection it may turn left, turn right or keep its original direction with equal chance.
4. The RDD's energy corresponds to the specific RDD used and is 0.662 [MeV], for *Cesium-137*, and 1.1173 or 1.332 [MeV] for *Cobalt-60*.
5. Background radiation was taken to have a (relatively high) expected value of 30 [μ Rad/Hour], with normal distribution and standard deviation of 15 [μ Rad/Hour]. In practice the background radiation level can be measured beforehand and can have different values in different places on the map.
6. Data were collected for consecutive time intervals, each of 1 [s] duration.
7. The x -coordinate is the *North-South* axis, and the y -coordinate is the *East-West* axis. (For simplicity, the topographic elevation for all positions on the map, is assumed equal to zero.)
8. The monitored area, V , was taken to be a rectangle area. Its size can be set by the user, and can be 4 up to 225 street blocks.
9. Each street block's width and length are 295 [feet].
10. Each street block is separated from its adjacent one by a two-way road with width of 33 [feet].
11. Sensors' locations, as they are mounted on vehicles, are limited to roads only.

A map, generated by the simulation software is presented in Fig. 6. The map is segmented by a grid of 3×7 into 21 blocks. The center of each block is represented by a node in G_{st}^W . In this example each node corresponds to a street intersection. This however is not necessarily the case in general, and grid points can be present inside buildings. The monitored region, V , is occupied with 17 sensors, mounted on vehicles ($|D| = 17$). In the simulation instance, presented in Fig. 6, 4 sensors detect the presence of the RDD, $|D^*| = 4$. Other 13 sensors report *all clear*, $|D^*| = 13$. The way the radiation detected by the sensor is computed, is elaborated on in Sect. 4.1.

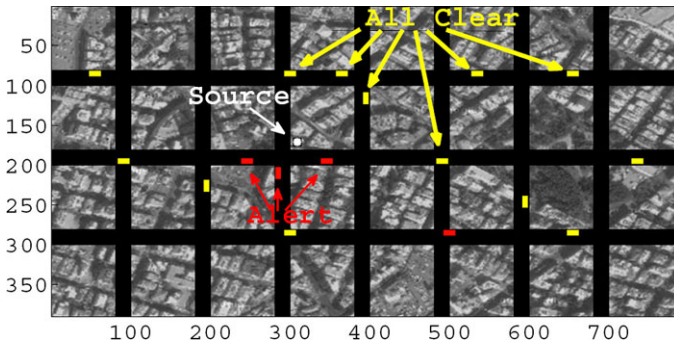


Fig. 6 Simulated city map

4.1 Radiation field

In our considerations, we take the radioactive source to be a small quantity of *Cesium-137* or *Cobalt-60*; these isotopes were chosen because their availability in industrial or medical sources and in powdered form make them a potential element of a RDD (“dirty bomb”).

The sensor’s radioactivity measurements are based on the count of emitted particles from the RDD. Therefore the Radiation Exposure Rate (RER), the number of particles counted by a sensor, is inversely proportional to the square of the distance between the RDD and the sensor:

$$RER(d) = \frac{\hat{\Gamma}}{d^2} \tag{11}$$

where $\hat{\Gamma}$ is a RDD specific parameter (Unger and Trubey 1982) and d is the distance between the radioactive source and the sensor.

In an ideal environment, the $RER_{(d)}$ defines the amount of radiation measured by a sensor with distance d from the RDD. However, in an urban environment, there are additional two factors that affect the radiation field. The first one is the radiation propagation medium being inhomogeneous. The radiation attenuation rate depends on the RDD’s characteristics and the propagation medium. For two types of media, air and concrete, the Radiation Attenuation Rate (RAR) is given by:

$$RAR(d_{air}, d_{concrete}) = \exp\{\mu_{air} \cdot |d_{air}|\} \cdot \exp\{\mu_{concrete} \cdot |d_{concrete}|\} \tag{12}$$

where μ_{air} and $\mu_{concrete}$ are RDD and propagation mediums specific constants (Berger et al. 1998) and $|d_{air}|$ and $|d_{concrete}|$ are the propagation distances in air and concrete respectively. Note that $d_{air} + d_{concrete} = d$.

The second factor, that affects the radiation is background radiation (BGR). Background radiation is the ionizing radiation constantly present in the environment, emitted from a variety of natural and artificial sources (UN Scientific Committee on Effects of Atomic Radiation 2000) and usually is known for major metropolises.

Incorporating these two factors, the total radiation at a given location is given by:

$$Rad(d_{air}, d_{concrete}) = RER(d_{air} + d_{concrete}) \cdot RAR(d_{air}, d_{concrete}) + BGR \tag{13}$$

Figure 7 depicts a simulated city map and its corresponding radiation maps for *Cobalt-60* source, located in two different positions. Figures 7(b) and (c) are the radiation maps, produced by the simulation software, which correspond to the RDD’s locations (A) and (B),

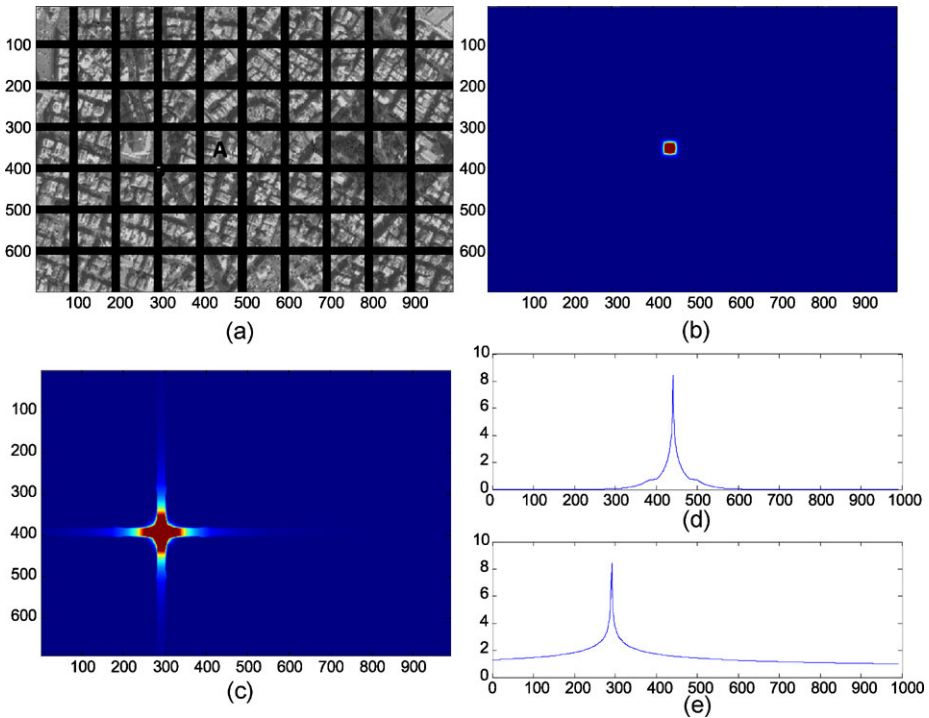


Fig. 7 Simulated city map and its corresponding radiation maps

marked on map (a). Figures 7(d) and (e) are cross sections (to the power of 0.1), going through the RDD positions, of the radiation maps (b) and (c), respectively. It is evident that a nuclear threat that is shielded inside a building is difficult to detect. On the other hand, when the RDD is located in the open, which is the case when transported or moved, the radiation measured range is significantly larger.

4.2 Simulation parameters

The simulation software, allows one to set the following simulation parameters:

1. *Number of Street Blocks*: Following the Manhattan model, we call pathways going from *east* to *west* as ‘*streets*’, while pathways going from *north* to *south* are referred to as ‘*avenues*’. Both, the number of streets and avenues, can be set by the user, thus determining the size of the area of interest.
2. *Number of Simulation Steps*: The number of consecutive time intervals, in which the simulation is performed.
3. *Number of Sensors*: The number of sensors, hence the number of vehicles taking part in the simulation.
4. *Radioactive Source Type*: The simulation allows to choose one out of two possible radiation sources, which a RDD can consist of: *Cesium-137* or *Cobalt-60*. Following the radioactive source’s type selection, the radiation energy and consequentially the $\widehat{\Gamma}_{(s)}$, μ_{air} and $\mu_{concrete}$ parameters are set automatically by the simulation software.
5. *Manual RDD Positioning*: By default the simulation software places the RDD in a random place on the map. In some cases, the user would like to position the RDD in a certain

position on the grid, in order to evaluate specific behavior of the algorithm. One example can be found in Sect. 4.1, where the RDD, in Fig. 7 is placed in the middle of a block, and on a streets intersection. Other example can be peripheral vs. center location on the map.

5 Aggregating the system's output over time

The general goal of the system is to reach a decision as soon as sufficient amount of data was collected so that the input is deemed reliable. The probabilities computed in Sects. 3.2 and 3.3 refer to the probability of reporting an error accumulating the measurements taken in a specific region. The discussion concludes that encompassing more sensors' readings does not contribute the increased reliability of the input data once the number of readings reaches saturation. Section 3.4 suggests aggregating samples over time until the number of samples collected saturates the system and only then invoke the algorithm.

In this section, we suggest to extend the time integration over successive algorithm's outputs. This is achieved through applying moving average on the algorithm's generated output threat map.

The output of the algorithm, described in this paper, for each processing period is a binary map, where for all $i \in D^*$ the output is 1, and 0 otherwise. This mechanism fails to point out for the user regions which have reported to contain threats more than the others. This can be achieved by simply assigning each region with a grade proportional to the number of times it was reported to contain a threat. For Δt the number of processing periods, each is region is assigned with the following grade:

$$G_{(i,t)} = \sum_{i=t}^{t+\Delta t} \frac{\text{Output}(i, \hat{t})}{\Delta t} \quad (14)$$

Figures 8(a), (b) and (c) depict simulation's outputs of three different times in one simulation run, where for each, respectively, the area found to contain risk, S , is incomplete. With that, $G_{(i,t)}$, presented in Fig. 8(d), not only mark all contaminated areas as such, it also indicates the severity of the threat for each of those regions and accurately detects the RDD's position. The darker the region, the higher the threat. In this simulation, we display the position of the RDD. Note however, that in real-life application, it is *not* available to the people in the central control room.

6 Interpolation

The algorithm's output, when an actual threat is present, is an area of elevated risk. Once an area has been determined to have elevated threat, we use interpolation to help pinpointing the RDD's location. The purpose of the interpolation procedure is to generate a continuous function from the discrete signal/position collection of samples. We seek a function that coincides with the value reported at each data point (sample's position). Methods for generating this type of function are reviewed in Yaroslavsky et al. (2009). Of these methods we select the iterative reconstruction procedure of the Papoulis (1975) type shown in the flow diagram of Fig. 9.

One might think that the interpolation method can be used to replace the algorithm. This is not true as the continuous interpolated function has the same values in the samples'

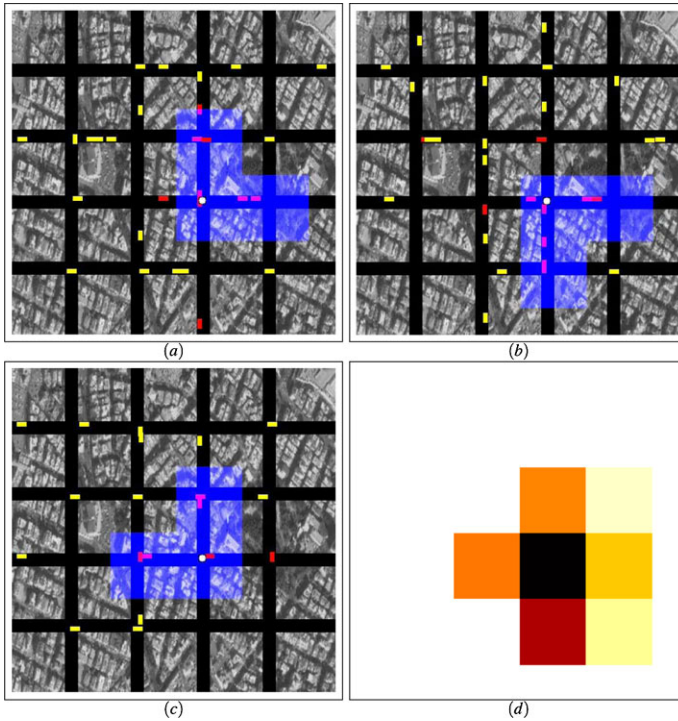


Fig. 8 Three processing periods outputs (figures (a), (b) and (c)) and their corresponding Moving average computation

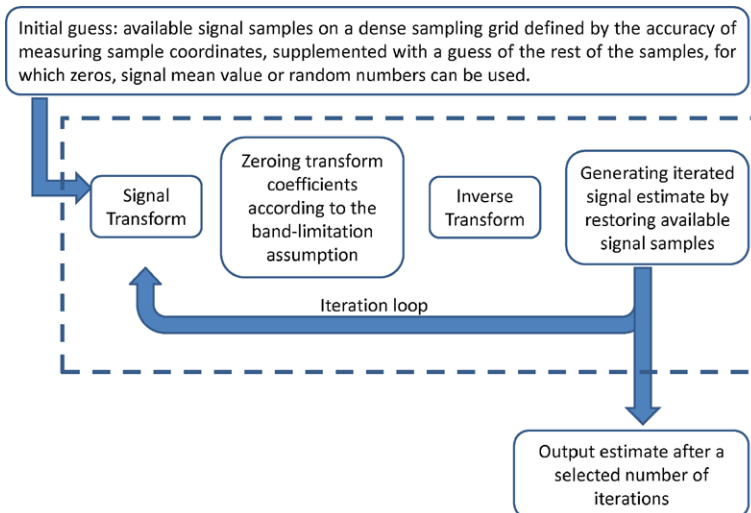


Fig. 9 Flow diagram of the iterative signal recovery procedure

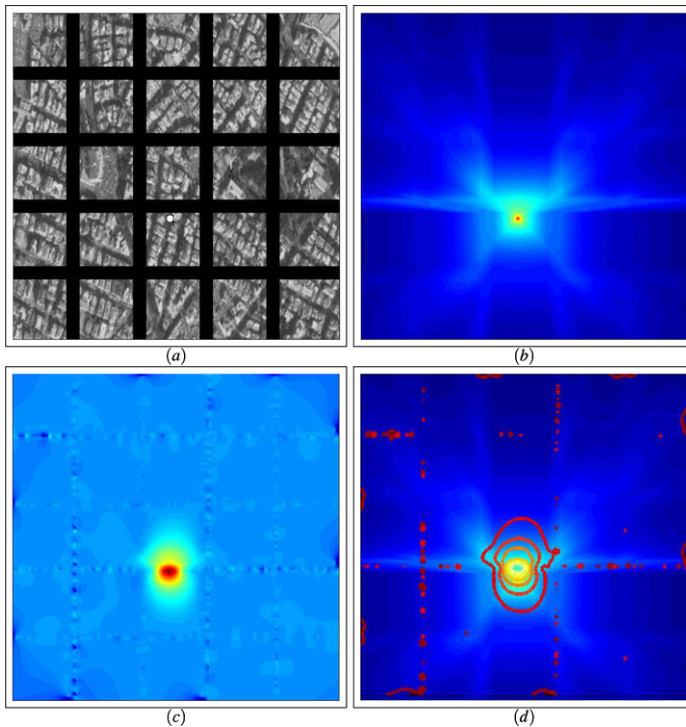


Fig. 10 Radiation field interpolation

positions as the corresponding data points. This implies that false positives will perpetually create multiple false peaks and alarms. This would lead to the same results as if we used the raw data collected from the sensors and trigger an alert whenever a single sensor reports an alert. Additionally the interpolation algorithm might be computationally complex. We cope with these shortcomings of the interpolation method by computing it only after an alarm has been generated by the algorithm, and using it to support followup actions.

Figure 10 shows the interpolation results. Figure 10(a) is the simulated map. Figure 10(b) is the radiation field, as computed by Eq. 13. Figure 10(c) presents, by dots, the sampling points taken over 750 consecutive processing periods with the interpolated radiation map. Finally, Fig. 10(d) combines Figs. 10(b) and (c) by presenting the interpolated map level lines on top of the computed radiation map. While it is not known in practice, the RDD's location is marked on each of the Figs. 10(a)–(d).

7 Mobile RDD

Sections 5 and 6 address the case of stationary RDD. In this section the contribution of the methods, described in those sections, when the RDD is mobile is presented. As derived from the discussion in the previous sections, the integration over time of the sensors' measurements when the RDD is stationary results in a better detection accuracy. When the RDD is mobile the aggregation of the system's output over time does not reveal the RDD's exact location. With that, the accumulated sensors data, over time, may be used for extracting the

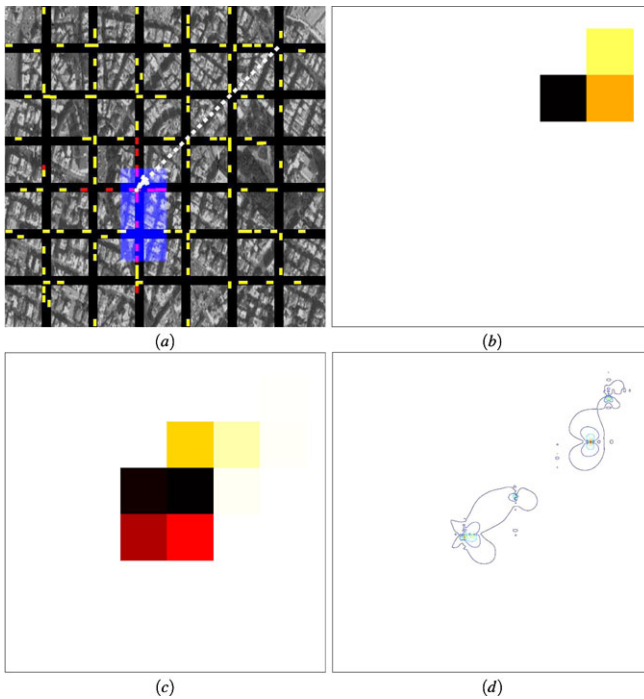


Fig. 11 Integration over time of the system's output for mobile RDD

RDD's trajectory. This is illustrated in Fig. 11. Figure 11(a) shows the RDD's trajectory from the upper-right section to the lower-left part of the map. Figure 11(b) depicts the interpolated radiation map, while Figs. 11(c) and (d) present the moving average computed at two different points in time. Figure 11(d) presents the moving average at a later time than (c). It is evident, that the RDD's positions and trajectory over time can be extracted from Figs. 11(b)–(d). In real-life applications, the deduced trajectory may be used to pinpoint the target of the attack.

8 Conclusions

We present here a formulation and an efficient algorithm solving the concentrated alert problem at each given point in time. The efficiency of the algorithm permits to use it online and aggregate the results over time. Statistical analysis of the approach presented, shows that such use of the algorithm in a decision support system reduces the likelihood of false-positive and false-negative alerts. The system's performance is shown to be linked to the density of the vehicles carrying detectors and with the detectors quality. Additionally, it has been shown that aggregation of the algorithm's output over time results in more robust and accurate output. The plan for follow-up research is to exploit the presented capabilities of the simulation software for fine-tuning of the scheme's various parameters (α , β , γ and Δt), for different setups. Eventually, the simulation software will be used for developing a real-life experiment, which will demonstrate the effectiveness of scheme presented here.

Acknowledgements This research of the first author has been supported in part by CBET-0736232 and NSF award No. DMI-0620677. The second author has been supported by CBET-0736232.

The authors wish to thank Prof. Eric B. Norman and Erik Swanberg, Department of Nuclear Engineering, University of California at Berkeley for the help in building the radiation field simulation. The first author thanks Fred Roberts for proposing (at the ARI Washington DC conference) the CA problem as a problem of interest.

Open Access This article is distributed under the terms of the Creative Commons Attribution Noncommercial License which permits any noncommercial use, distribution, and reproduction in any medium, provided the original author(s) and source are credited.

References

- Attix, F. H. (1986). *Introduction to radiological physics and radiation dosimetry*. New York: Wiley.
- Berger, M. J., Hubbell, J. H., Seltzer, S. M., Chang, J., Coursey, J. S., Sukumar, R., & Zucker, D. S. (1998). *Xcom: Photon cross sections database. Standard reference database 8 (xgam)*. National Institute of Standards and Technology.
- Brennan, S. M., Mielke, A. M., Torney, D. C., & MacCabe, A. B. (2004). Radiation detection with distributed sensor networks. *Computer*, 37(8), 57–59.
- Brennan, S. M., Mielke, A. M., & Torney, D. C. (2005). Radioactive source detection by sensor networks. *IEEE Transactions on Nuclear Science*, 52(3), 813–819.
- Chandran, B., & Hochbaum, D. S. (2009). Pseudoflow solver. <http://riot.icor.berkeley.edu/riot/Applications/Pseudoflow/maxflow.html>.
- Cunningham, C. T. (1995). *Detecting and track of a stochastic target using multiple measurements* (Technical Report UCRL-ID 122786). Lawrence Livermore National Laboratory.
- Fawcett, T. (2006). An introduction to roc analysis. *Pattern Recognition Letters*, 27(8), 861–874.
- Ford, L. R., & Fulkerson, D. R. (1956). Maximal flow through a network. *Canadian Journal of Mathematics*, 8(3), 339–404.
- Hochbaum, D. S. (2008). The pseudoflow algorithm: A new algorithm for the maximum-flow problem. *Operations Research*, 56(4), 992–1009.
- Hochbaum, D. S. (2009). The multi-sensor nuclear threat detection problem. In J. W. Chinneck, B. Kristjansson, & M. J. Saltzman (Eds.), *Operations research/computer science interfaces series: Vol. 47. Operations research and cyber-infrastructure* (pp. 389–399). New York: Springer.
- Nemzek, R. J., Dreicer, J. S., Torney, D. C., & Warnock, T. T. (2004). Distributed sensor networks for detection of mobile radioactive sources. *IEEE Transactions on Nuclear Science*, 51(4), 1693–1700.
- Papoulis, A. (1975). A new algorithm in spectral analysis and band-limited extrapolation. *IEEE Transactions on Circuits and Systems*, 22(9), 735–742.
- Schaller, B. (2006). *The New York city taxicab fact book* (Technical report). Schaller Consulting, 94 Windsor Place, Brooklyn, NYC, NY, US.
- Stephens, D. L., & Peurrung, A. J. (2004). Detection of moving radioactive sources using sensor networks. *IEEE Transactions on Nuclear Science*, 51(5), 2273–2278.
- UN Scientific Committee on Effects of Atomic Radiation (2000). *Report to the general assembly, with scientific annexes, volume I: Sources* (Technical report). United Nations General Assembly.
- Unger, L. M., & Trubey, D. K. (1982). *Specific gamma-ray dose constants for nuclides important to dosimetry and radiological assessment* (Technical Report ORNL/RSIC-45/R1). Oak Ridge National Laboratory.
- University of California at Berkeley (2009). Domestic nuclear threat security initiative. <http://donuts.berkeley.edu>.
- Yaroslavsky, L. P., Shabat, G., Salomon, B. G., Ideses, I. A., & Fishbain, B. (2009). Nonuniform sampling, image recovery from sparse data and the discrete sampling theorem. *Journal of Optical Society of America A*, 26(3), 566–575.