CERIAS Tech Report 2010-22

Efficient and Robust Solutions for Sensor Network Detection and Localization

by Jren-Chit Chin

Center for Education and Research
Information Assurance and Security
Purdue University, West Lafayette, IN 47907-2086

Graduate School ETD Form 9
(Revised 12/07)

PURDUE UNIVERSITY
GRADUATE SCHOOL
Thesis/Dissertation Acceptance

This is to certify that the thesis/dissertation prepared

By Jren-Chit Chin

Entitled Efficient and Robust Solutions for Sensor Network Detection and Localization

For the degree of Doctor of Philosophy

Is approved by the final examining committee:

David K. Y. Yau Chair	
Bharat Bhargava	
Sonia Fahmy	
Nageswara S. V. Rao	

To the best of my knowledge and as understood by the student in the *Research Integrity and Copyright Disclaimer (Graduate School Form 20)*, this thesis/dissertation adheres to the provisions of Purdue University's "Policy on Integrity in Research" and the use of copyrighted material.

Approved by Major Professor(s): David K. Y. Yau

Approved by: William J. Gorman — July 07, 2009
Head of the Graduate Program — Date

Graduate School Form 20
(Revised 1/10)

PURDUE UNIVERSITY
GRADUATE SCHOOL

Research Integrity and Copyright Disclaimer

Title of Thesis/Dissertation:
Efficient and Robust Solutions for Sensor Network Detection and Localization

For the degree of Doctor of Philosophy

I certify that in the preparation of this thesis, I have observed the provisions of *Purdue University Teaching, Research, and Outreach Policy on Research Misconduct (VIII.3.1),* October 1, 2008.*

Further, I certify that this work is free of plagiarism and all materials appearing in this thesis/dissertation have been properly quoted and attributed.

I certify that all copyrighted material incorporated into this thesis/dissertation is in compliance with the United States' copyright law and that I have received written permission from the copyright owners for my use of their work, which is beyond the scope of the law. I agree to indemnify and save harmless Purdue University from any and all claims that may be asserted or that may arise from any copyright violation.

Jren-Chit Chin
Printed Name and Signature of Candidate

June 03, 2010
Date (month/day/year)

*Located at http://www.purdue.edu/policies/pages/teach_res_outreach/viii_3_1.html

EFFICIENT AND ROBUST SOLUTIONS FOR SENSOR NETWORK DETECTION AND LOCALIZATION

A Dissertation
Submitted to the Faculty
of
Purdue University
by
Jren-Chit Chin

In Partial Fulfillment of the
Requirements for the Degree
of
Doctor of Philosophy

August 2010
Purdue University
West Lafayette, Indiana

To my mom Peik-Wan Teng, and the memory of my dad Nget-Leng Chin.

ACKNOWLEDGMENTS

Throughout my doctoral studies, I have met numerous scholars. These individuals have directly or indirectly influenced my life, both professionally and personally. I would like to take this opportunity to express my gratitude to all of you who have given me strength and support.

I would like to express my greatest gratitude to my academic advisor, Dr. David Yau, who has given me the opportunity to work in this area and provided valuable advice over the last five years. Without him, this dissertation would not be possible. Throughout these years, Dr. Yau has imprinted a think-outside-the-box mentality in me, which has led to creative ways to solve difficult research problems. I would like to thank Dr. Nageswara Rao for his invaluable insight and guidance. The discussions we had have been rewarding. I appreciate Dr. Rao's patience and valuable time in entertaining all my questions.

I would like to thank Frank DeNap, and the Office of Naval Research for providing funding for this exciting research. That capital has supported the majority of the research described in this dissertation. Those assets include the purchase of various pieces of equipment needed to set up the test-bed necessary for real-life experiments. I would like to thank Dr. Mallikarjun Shankar and the team of the SensorNet project for their assistance during my visit to Oak Ridge National Laboratory.

Next, I would like to thank Dr. Sonia Fahmy, Dr. Bharat Bhargava, Dr. Dongyan Xu, and Dr. Ramana Kompella for their time and effort serving on my advisory, preliminary examination, and qualifying examination committees. Their feedback and advice have solidified my research in many ways. I am very thankful to Dr. Bharat Bhargava for the encouragement and confidence he gave me to achieve what I deemed impossible. In addition, I would like to express my gratitude to Dr. Wing-Kai Hon,

Dr. Santiago Pujol and Jeffrey Rautenberg with whom I collaborated in several projects. I enjoyed working with all of you.

I am very thankful to the staff of the Computer Science Department for their assistance in straightening the crooked path to graduation. Dr. William J. Gorman, the late Amy Ingram, and Sandra Freeman have helped me with various university procedures. I will never forget Amy Ingram, who had everything that a graduate student needed to know in her head, and provided accurate information for all questions with a sense of humor and a cheerful smile. Linda Byfield made sure all my reimbursements were in order. Ron Castongia made equipment purchasing easier than ever, and Melanie Church was efficient in troubleshooting software problems, enabling my experiments to run without a glitch.

I would also like to express my gratitude to Dr. Morris Chang at Iowa State University, where I obtained my bachelor's degree. I am grateful that Dr. Chang gave me the opportunity to experience life in graduate school, to be both an undergraduate teaching assistant as well as a research assistant.

I would like to thank my family for their unconditional support throughout these years. Without them, I would not be who I am today. A special thanks to my beloved Lixia Liu, who has brought sunshine into my life and has given me the strength to accomplish my doctoral studies. In addition, I am very grateful to Lisa Liu and Richard Barton, who spent their precious time meticulously proofreading this dissertation. After all the time they have spent, I hope they learned what I was working on. Last, but not least, I would like to thank my fellow lab mates Yu Dong, who graduated shortly after I came to Purdue, Chris Ma, and Wei-Chiu Chuang.

TABLE OF CONTENTS

Page

LIST OF TABLES

LIST OF FIGURES

Figure Page

ABSTRACT

Chin, Jren-Chit. Ph.D., Purdue University, August, 2010. Efficient and Robust Solutions for Sensor Network Detection and Localization. Major Professor: David K. Y. Yau.

The ability to quickly detect and locate stealthy radiological dispersal devices (RDDs) allows authorities to disarm and remove the RDDs before they can be detonated. Traditionally, the detection of RDDs was accomplished by using expensive and cumbersome radiation sensors strategically located in the surveillance area. However, with recent advancements in wireless technologies and sensing hardware, deploying a large scale sensor network with small sensors is now becoming a reality. In this dissertation, we study methods to detect and locate radiation sources quickly and accurately using a network of sensors.

Localization of a single radiation source can be achieved by using three sensors in a noise- and error-free environment. When both noise and errors are considered, we present a closed-form solution that outperforms existing algorithms. When more than three sensors are available, we present an efficient algorithm to exploit the additional sensor data, in order to further improve the robustness and accuracy of the localization.

To localize multiple sources in a sensor network, we propose a hybrid formulation of a particle filter with a mean-shift technique, in order to achieve several important features which address major challenges faced by existing multiple source localization algorithms. First, our algorithm is able to maintain a constant number of estimation (source) parameters even as the number of radiation sources K increases. Second, our algorithm "learns" the number of sources from the estimated source parameters instead of relying on expensive statistical estimations. Third, the presence of obstacles may improve the localization accuracy of our algorithm. Unfortunately, the presence of obstacles significantly degrades the accuracy of existing algorithms.

When no radiation source is present, the localization algorithms produce false positives as the algorithms assume that a radiation source is present. We propose the Localization Enhanced Detection (LED) method, that decides whether a source with the estimated parameters is present or absent, using a close-to-minimal number of measurements, while maintaining the false positive and false negative rates below a specified level. We evaluate the LED method using simulation and testbed experiments, and compare the effectiveness of the LED method with existing detection methods.

We build a cross-platform, cross-language, and versatile software framework that provides an abstraction for interfacing with sensors and supports building applications on radiation source localization. The software framework implements various localization algorithms that are ready to be deployed in an actual system. The components in the software framework are loosely coupled and are general enough to support application domains beyond radiation source localization. We demonstrate the versatility of the software framework in building the Rapid Structural Assessment Network.

1 INTRODUCTION

Urbanization is a powerful trend. In the United States, more than 80% of the population will live in a city by 2025. The higher concentration of people increases their susceptibility to stealthy attacks. In launching a stealthy attack, one of the possibilities is to use a home-made explosive device containing radioactive materials. Such a device is referred to as a radiological dispersal device (RDD), and is often called a “dirty bomb.” Although such a device is nothing like a nuclear explosive device, a RDD is more likely to be used in an attack because it requires limited technical knowledge to build and deploy, compared with a nuclear explosive device. In addition, one does not need weapons grade uranium or plutonium, which is difficult to obtain, to build an RDD. All radioactive materials, such as those used in medicine and industry, can be used to build a RDD.

When a RDD is detonated, the primary explosion might be small and dismissed as insignificant. However, harmful radioactive substances unleashed in such an attack could cause widespread health and environmental damage, since such substances would be extremely hard to clean up and would cause adverse effects from long term exposure. Moreover, attacks with radioactive material would cause psychological fear and economic disruption. Therefore, the ability to quickly detect and localize illicit radiation sources is crucial. Such swift action enables authorities to find and remove RDDs during transport or storage. Moreover, attackers may launch a coordinated assault involving multiple RDDs brought into an area and installed at different places. In such a scenario, we would need a system that can determine the number of devices, as well as the strength and location of each device. The majority of current localization systems are designed for a single source, and are not effective against multiple sources in a coordinated attack [1].

Radioactive materials, such as those found in RDDs, emit three types of ionizing radiation—alpha, beta, and gamma radiations. Alpha radiation is basically a Helium nucleus consisting of two protons and two neutrons. Due to its larger size and heavier weight, compared with beta and gamma radiations, alpha radiation can only travel a limited distance, around 10 cm, through the air, and can be shielded with a thin sheet of paper. Hence, alpha radiation is considered the least hazardous of the three types. Beta radiation is basically the ejection of high speed electrons from a nucleus. Due to its smaller size, beta radiation can travel through the air about 4 meters and requires a denser material, such as aluminum, for shielding. Gamma radiation, in contrast, is high energy electromagnetic radiation that can travel several kilometers through the air. Gamma radiation is similar to X-ray radiation commonly used in medical imaging, but with much higher energy and penetration power. Due to its high energy and penetrating properties, gamma rays are considered the most hazardous of the three types of radiation. When penetrating through live tissue, gamma radiation can affect DNA and cause cancer. Thus, exposure to gamma radiation is considered a serious health hazard [2].

These three types of ionizing radiation are invisible to the human eye. They can only be detected with sensors such as a Geiger-Müller counter or a sodium iodide scintillation counter. Currently, radiation sensors are deployed at United States ports and borders [3] as well as in urban centers such as Washington, D.C. [4]. These radiation sensors can detect the presence of radiation sources and alert authorities when traces of radiation sources are found. Nonetheless, these radiation sensors cannot indicate the location of a radiation source. Due to the high cost and large footprint of these sensor devices, they are deployed only at the most sensitive sites in the country. As sensor technologies mature, it is now possible to produce sensors at a lower unit cost with smaller form factors. Such sensors could then be embedded in mobile devices [5–7] such as cellular phones and law-enforcement vehicles [8,9]. At the same time, ubiquitous wireless connectivity, such as mobile broadband connection and vehicular network [10], allows sensors to disseminate real-time data to a control center.

At the control center, sensor measurements are fused to produce valuable information about important events occurring in the environment, in addition to basic intensity measurements. These technological advancements enable the authorities to deploy a large scale sensor network that monitors the physical environment for sources of radiation. Should such radiation sources be detected, data from the networked sensors could be used to derive the location of the sources.

1.1 Thesis

The thesis of this dissertation can be stated as follows:

As wireless networks become a commodity and inexpensive radiation sensors become available for general use, building a large scale radiation sensor network is becoming a reality. With large numbers of sensor measurements arriving at the control center in real-time, efficient algorithms that detect the presence, and estimate the location and strength, of radiation sources are critical for incidence response and can be implemented on commodity computing hardware. Moreover, exploiting the understood properties of sensor data in the application domain can lead to creating localization algorithms that are accurate, robust, efficient, and sustain real-time performance.

This thesis is supported by the following areas of research. First, the sensor data, which measures the strength of radiation sources, is shown to follow the Poisson random process and the inverse square decay. By exploiting the inverse square fading of intensity, and incorporating the statistical properties of the Poisson process into the algorithm design, the performance of the algorithm can be improved.

Second, localization algorithms require measurements from multiple sensors. Large network transmission latency in a wireless sensor network often leads to longer lengths of time waiting for sensor measurements. This delay prevents the localization algorithms from updating the location and strength estimates more frequently. In addition, the waiting time can be infinite if any of the sensors malfunction. However, by not requiring that all measurements be received before running the localization algorithm, the waiting time is eliminated. This improves the robustness of the algorithm by mitigating common problems in wireless sensor networks and allows the algorithm to achieve real-time performance.

Third, sensor measurements include noise from background radiation that may lead to excessive false positives in the localization algorithm. However, typical approaches where insignificant measurements are discarded may lead to false negatives. By formulating a detection algorithm, utilizing the network of sensors and a localization algorithm, the detection time of a radiation source can be reduced while maintaining the specified false positive and false negative rates.

1.2 Contributions

The main contributions of this dissertation are as follows:

- We elucidate the sources of errors in an actual sensor network, due to (1) background radiation and its inaccurate characterization; (2) variable efficiencies of different sensors and their incomplete determination; (3) the probabilistic nature of the underlying physical phenomenon causing statistical errors when the measurement time is limited, as required by the near real-time requirement; and (4) practical limitations of sensor designs and implementations.

- We present the Ratio of Square-Distance (RoSD) algorithm to localize a radiation source using three sensors. We show that the RoSD algorithm is more accurate and significantly faster than the existing log-space DTOA algorithm (LgDTOA) [11,12].

- We present the Iterative Pruning (ITP) algorithm to localize a radiation source with $N > 3$ sensors and show that the accuracy of the localization improves with the increase in number of sensors. We compare the performance of ITP with existing single source localization algorithms. We show that the ITP algorithm improves upon the Mean-of-Estimator (MoE) algorithm [13] by explicitly addressing the existence of *phantom estimates* and by preferring higher-SNR estimates in the fusion process. In addition, we show that the localization accuracy of the ITP algorithm is comparable to the Maximum Likelihood Estimation (MLE) algorithm [14]. However, the ITP algorithm generally runs faster because it does not require an expensive multidimensional optimization step over a large parameter space.

- We present a novel localization approach for multiple radiation sources in complex environments that (i) is efficient in handling multiple radiation sources; (ii) does not require the number of sources to be known in advance, or require expensive statistical estimations of the number of sources; (iii) does not require detailed specifications of obstacles in the environments. This localization algorithm incorporates the use of a hybrid, consisting of a particle filter and a mean-shift technique, with the concept of *fusion range* to selectively update the particles and thus limit the interference between close-by radiation sources. This allows our algorithm to handle multiple sources without explicitly modeling all radiation sources and obstacles. Subsequently, the algorithm has efficiency that scales to a large number of radiation sources. We provide quantitative results to show that the proposed algorithm can accurately localize radiation sources and achieve low false positive and false negative rates.

- We present the Localization Enhanced Detection (LED) method, which subsumes the detection and localization steps. Instead of a traditional approach, where a radiation source must first be detected before estimates of the location and strength of the detected source can be calculated, we show that the detection and localization steps can reinforce each other in the LED method [15–17]. The LED method uses a minimal number of measurements to detect the presence or absence of a radiation source, while ensuring the false positive and false negative rates to user specified values.

- For an extremely low level 0.911 μCi radiation source, we present actual testbed results in the performance evaluation. In addition, we present simulation results using a validated simulation model such that we can systematically explore the parameter space (e.g., vary the source strength), which is impossible in the actual testbed. We present a larger scale simulation based on the validated simulation model for stronger radiation sources, comparable to the intensity of a RDD. We show that a RDD can be localized in near real-time with good accuracy in practice.
- We have developed a versatile software framework to support simulation and testbed experiments on radiation source detection and localization. The software framework has been designed in such a way that the components are ready to deploy on actual system implementation. We have demonstrated the versatility of the software framework by presenting a further application for structural health monitoring—the Rapid Structural Assessment Network—using the software framework.

1.3 Organization

This dissertation is organized as follows. In Chapter 2, we describe the gamma radiation propagation and sensing model, and discuss the sources of noise and errors in sensing. In Chapter 3, we describe the Ratio of Square-Distance (RoSD) and Iterative Pruning (ITP) localization methods for localizing a single radiation source in a network of three and N sensors, respectively. In Chapter 4, we propose a hybrid formulation of a particle filter with a mean-shift technique for localizing multiple radiation sources. Chapter 5 presents the evaluation results of the proposed localization algorithms in both simulation and testbed experiments. In Chapter 6, we propose the Localization Enhanced Detection (LED) method to determine if a localized radiation source is actually present. In Chapter 7, we present a versatile software framework that is used to support all the simulation and testbed experiments. Chapter 8 describes related work in the area of detection and localization. Chapter 9 summarizes.

2 SYSTEM MODELS

We consider the localization of K point radiation sources of unknown strengths using a network of radiation sensors in a two-dimensional surveillance area. In this section, we describe the propagation model of gamma ray point radiation sources in Section 2.1 and the sensing model in Section 2.2. Section 2.3 discusses the sources of noise and error in sensor measurements, and Section 2.4 describes a calibration procedure that profiles the sensor parameters for use in the sensing model.

2.1 Gamma Radiation Propagation Model

Let $\mathcal{A} = \{\mathbf{A}_1, \ldots, \mathbf{A}_K\}$ denote the set of radiation sources. Each radiation source is parameterized by a three-value vector $\mathbf{A}_j = \left\langle A_j^{\mathrm{x}}, A_j^{\mathrm{y}}, A_j^{\mathrm{str}} \right\rangle$, for $1 \leq j \leq K$. The position of the source is given by its xy coordinates in centimeters (cm) and the strength of the source is a positive real number in micro-Curies (μCi). For convenience, we define $A_j^{\mathrm{pos}} = \left(A_j^{\mathrm{x}}, A_j^{\mathrm{y}}\right)$. In a free-space surveillance area, the strength of $\mathbf{A}$ measured at location $\mathbf{x}$ is

$$\mathcal{I}_{FS}\left(\mathbf{x}, \mathbf{A}\right) = A^{\mathrm{str}} \left(1 + |\mathbf{x} - A^{\mathrm{pos}}|^2\right)^{-1} \ . \tag{2.1}$$

The surveillance area may contain a number of obstacles that shield the radiation rays. The obstacles can be made of any type of material, be of any shape, and may be located anywhere in the surveillance area, unknown to the system. The material of the obstacle determines its effectiveness in absorbing radiation. Materials of higher atomic numbers and higher densities are more effective in absorbing gamma radiation. In addition, the amount of gamma radiation absorbed in the shielding material is proportional to the thickness of the material. For instance, lead that is 1 cm thick can block the same amount of radiation[1] as concrete that is 6 cm thick [18]. For

---

[1] Gamma radiation with energy 1 MeV.

simplicity, and without loss of generality, we assume that each obstacle is composed of homogeneous material. Given that the radiation source is on one side of the obstacle, the intensity of the source on the other side of the obstacle is modeled by

$$\mathcal{I}_S\left(l, \mathbf{A}\right) = A^{\mathrm{str}} e^{-\mu l}, \tag{2.2}$$

where l denotes the thickness of the obstacle and μ denotes the attenuation coefficient for the material of the obstacle. The values of μ for different materials are published in [18].

In a realistic setup, gamma rays traverse through both free space and obstacles. The total attenuation of the traversal is a combination of Equation (2.1) and (2.2). We model the total attenuation using the following equation: Given a radiation source $\mathbf{A}$ and a set of obstacles $\mathcal{B}$, the intensity of this radiation source at some location $\mathbf{x}$ is

$$\mathcal{I}\left(\mathbf{x}, \mathbf{A}\right) = \frac{A^{\mathrm{str}}}{1 + \left|\mathbf{x} - A^{\mathrm{pos}}\right|^2} \exp\left(\sum_{b \in \left(\mathcal{B} \cap \overline{\mathbf{xA}}\right)} -\mu_b l_b\right), \tag{2.3}$$

where $\mathcal{B} \cap \overline{\mathbf{xA}}$ denotes the set of obstacles that intersect with the straight line between $\mathbf{x}$ and A^{pos}, l_i denotes the total thickness of the obstacle b along $\overline{\mathbf{xA}}$, and μ_b denotes the attenuation coefficient of obstacle b.

2.2 Gamma Radiation Sensing Model

When gamma rays are emitted from a source, the radiation ionizes the particles it travels through. Sensors detect this ionization as an indication of the presence of alpha, beta, or gamma radiation. As alpha and beta radiation travels a limited distance, gamma radiation is considered the primary source of ionization. The intensity of the radiation is measured by counting the number of ionizations occurring at the sensor for a fixed interval. This intensity is usually reported in counts per minute (CPM). Let S_i, for $1 \leq i \leq N$, denote the known location of radiation sensor i. The sensor at S_i will record the total intensity from all the radiation sources. In addition, each sensor i will also record background radiation B_i (in CPM) that is universally present due to decay of naturally occurring radio-isotopes such as Carbon-14 (C-14) and Potassium-40 (K-40), cosmic rays from space, as well as from other man-made sources such as X-ray machines and nuclear fuels. Combining all of the above, the expected intensity (in CPM) at S_i is modeled by

$$I_i = 2.22 \times 10^6 \times E_i \sum_{j=1}^{K} \mathcal{I}\left(S_i, \mathbf{A}_j\right) + B_i \ . \tag{2.4}$$

Given the expected intensity I_i, the measurement for sensor i, $m\left(S_i\right)$, is modeled as a Poisson process with average rate $\lambda = I_i$.

In Equation (2.4) above, the constant 2.22×10^6 relates the total intensity of all radiation sources, given in micro-Curie units, to the number of nuclear disintegrations per minute (DPM). Each disintegration produces a gamma radiation that will be detected by the sensor. The fraction of gamma radiation detected by the sensor is denoted by the sensor efficiency E_i, of unit CPM/DPM. Some radiation sensors are more sensitive than others due to different manufacturing technologies and different sensor sizes. The value of E_i corrects the bias caused by the different efficiencies of sensors. The calibration procedure in Section 2.4 describes the process of obtaining E_i and B_i for a sensor.

2.3 Sources of Noise and Errors

Accurate measurements of the intensity of a radiation source, even in cases where there is only a single radiation source, are difficult due to the nature of nuclear disintegrations, the presence of background radiation, and the limitations of existing sensing hardware. First, gamma radiation emitted by a point radiation source is probabilistic and follows the Poisson process [12,19,20]. Because of the randomness of the phenomenon being observed, a single instantaneous measurement of the intensity of the source is not reliable in determining the actual strength of the source. In principle, a perfect measurement would require an infinite measurement time interval. In practice, only a limited number of samples can be used, particularly in the case of real-time or near real-time localization. The magnitude of error is given by the standard deviation of the difference between the measured intensity and the true intensity, and is proportional to $\frac{1}{\sqrt{m}}$ where m denotes the number of samples.

Second, the sensors also record the background radiation, in addition to the gamma radiation emitted by the source. Background radiation is universal, but is difficult to characterize due to many variables in natural and man-made sources. We are exposed to background radiation in everyday life, which is not a cause for concern and hence is classified as part of the background radiation B_i. However, because each source of background radiation is probabilistic, and their exact combinations are not known and vary with time and space, a perfect characterization of the background radiation is impossible.

Third, limitations of sensing technologies and cost considerations for the sensing hardware may affect the accuracy of measurements. Radiation sensors expose a sensing area with a scintillation material that detects ionizing radiation such as gamma radiation. The scintillation material is ionized when gamma radiation travels through the sensing area. This ionization signals a detection of gamma radiation. A sensor either keeps track of the number of detections internally or reports this detection to an external computer. The intensity of the gamma radiation is measured by either the rate that ionization occurs or the time interval between two ionizations. Because of this detection methodology, different types of scintillation materials affect the efficiency of the sensor [21]. In addition, a sensor exposing a larger sensing area increases the probability that gamma radiation travels through the sensing area and thus increases the efficiency of the sensor as well. Nonetheless, it is not practical to manufacture huge radiation sensors, and many applications do not need that level of efficiency. Therefore, a radiation sensor is only capable of detecting a fraction of the radioactive particles emitted by a point source. This introduces the third type of error in the measurements, and this error is corrected to some degree by the sensor efficiency E_i in our sensing model. However, dust and other contaminants gathering on the sensing area of the detectors over time may degrade sensor efficiency.

2.4 Sensor Calibration

The sensing model described in Section 2.2 requires that the background radiation intensity B_i and the sensor efficiency E_i be known in advance in order to measure the intensity due to the radiation sources. This section describes a calibration procedure to obtain the values of B_i and E_i with a controlled radiation source. The calibration procedure is performed at the deployment site because the background radiation differs at different locations. In addition, sensors should be recalibrated periodically and after significant changes to the surroundings, to ensure that measurements are accurate.

The calibration procedure is performed in two steps. First, we collect measurements over a period of time at the deployment site, at a time when no known radiation source is present in the area in order to estimate the background radiation. The background radiation is estimated by

$$B_i = \frac{\text{Total number of counts without a source over a period of } t}{t} \tag{2.5}$$

where t is the length of measurements in minutes.

Second, we place a radiation source with a known source strength on the sensor and collect measurements over a period of time. Then, collected measurements are used to compute the sensor efficiency by

$$I_i = \frac{\text{Total number of counts with a source over a period of } t}{t} \tag{2.6}$$

and

$$E_i = \frac{I_i - B_i}{A^{\mathrm{str}} \times 2.22 \times 10^6} \tag{2.7}$$

The length of the measurements t affects the accuracy of the calibration. Essentially, the value of t must be large enough such that the total number of counts is non-zero. A longer period of t results in a more accurate estimate of B_i and I_i, which subsequently affects the accuracy of E_i.

3 SINGLE RADIATION SOURCE LOCALIZATION

In this chapter, we consider the localization of a single radiation source ($K = 1$) in a surveillance area without obstacles. We first propose the Ratio of Square-Distance (RoSD) method to localize a single radiation source using a minimal three sensor network. Then, we describe the Iterative Pruning (ITP) method to localize a single radiation source using $N > 3$ sensors by combining data from RoSD with a data fusion method.

3.1 Ratio of Square-Distance Method for 3 Sensor Network

The RoSD algorithm [22] requires measurements from groups of 3 sensors to localize a radiation source. Given the group of sensors at S_i for $i = 1 \ldots 3$, the expected intensity due to a point radiation source at S_i is

$$I'_i = \frac{I_i - B_i}{E_i \times 2.22 \times 10^6} = \frac{A^{\text{str}}}{1 + |S_i - A^{\text{pos}}|^2} \tag{3.1}$$

according to the sensing model in Section 2.2. For real-time performance, we approximate expected intensity I_i with sensor measurement $m\left(S_i\right)$. The ratio of intensity measurements by two of the sensors, S_i and S_j, due to the radiation source, is hence

$$\frac{I'_j}{I'_i} = \frac{1 + |S_i - A^{\text{pos}}|^2}{1 + |S_j - A^{\text{pos}}|^2} \; . \tag{3.2}$$

To make the problem more tractable, such that a closed-form solution exists, we approximate Equation (3.2) with

$$\frac{I'_j}{I'_i} \approx \frac{|S_i - A^{\text{pos}}|^2}{|S_j - A^{\text{pos}}|^2} \; . \tag{3.3}$$

Using the relationship between the ratio of measured intensity and their distances in Equation (3.3), we compute the circular locus

$$L_{ij} = \left\{ \mathbf{x} \;\middle|\; \frac{I'_j}{I'_i} = \frac{|S_i - \mathbf{x}|^2}{|S_j - \mathbf{x}|^2} \right\} \tag{3.4}$$

where L_{ij} is a set of all possible locations of the radiation source that satisfies the ratio of intensity measurement at S_j to that at S_i (which is also the ratio of square-distance of S_i to S_j from the source).

The Ratio of Square-Distance (RoSD) method computes the source position estimate $\hat{A}^{\text{pos}} = [x \;\; y]^\top$ by solving the equation $L_{12} = L_{13}$. The solution of the equation is

$$\hat{A}^{\text{pos}} = [L_{12}^x \;\; L_{12}^y]^\top + L_{12}^r[\cos(\beta \pm \theta) \;\; \sin(\beta \pm \theta)]^\top, \tag{3.5}$$

where

$$\beta = \arctan \frac{L_{13}^y - L_{12}^y}{L_{13}^x - L_{12}^x} \quad \theta = \arccos \frac{{L_{12}^r}^2 - {L_{13}^r}^2 + C}{2L_{12}^r\sqrt{C}}$$

$$C = (L_{13}^x - L_{12}^x)^2 + (L_{13}^y - L_{12}^y)^2$$

$$L_{ij}^x = \frac{I_i' S_i^x - I_j' S_j^x}{I_i' - I_j'} \qquad L_{ij}^y = \frac{I_i' S_i^y - I_j' S_j^y}{I_i' - I_j'} \qquad L_{ij}^r = \tfrac{|S_i - S_j|}{I_i' - I_j'}\sqrt{I_i' I_j'}\ . \tag{3.6}$$

The geometric interpretation of the solution is shown in Figure 3.1. When $\theta \neq 0$, there are two solutions to Equation (3.5). Either solution could be the actual position of the source. We do not know which one of the two is the actual position of the source because a strong source located far away from the sensors may induce measurements that are similar to a weak source located close to the sensors. One of the two estimates is a false estimate, and we call that solution a *phantom estimate*. Identifying the phantom estimate produced by 3 sensors is not straightforward. The fusion step in Section 3.2 accomplishes this task by using measurements from $N > 3$ sensors. An example of solving $L_{12} = L_{13}$ is shown in Figure 3.2. In Figure 3.2, three sensors are located at $S_1 = (-100, 0)$, $S_2 = (100, 0)$, and $S_3 = (0, 100)$. The ratios of sensor measurements are $\frac{I_2'}{I_1'} = 0.40$ and $\frac{I_3'}{I_1'} = 3.00$. These ratios correspond to the loci L_{12} and L_{13}. The intersections of the loci are $\hat{A}^{\text{pos}}$.

In a practical implementation, when sensor measurements are noisy, the estimates produced by the algorithm are no longer precise. In fact, the algorithm may not produce any solution at all. This occurs when L_{12} and L_{13} form two disjoint circles, or one of the circles is contained within the other circle. In these cases, for robust

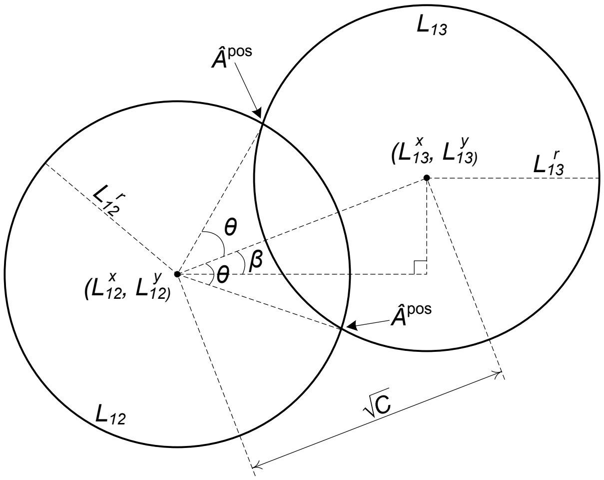


Figure 3.1. Geometric interpretation of the RoSD solution.

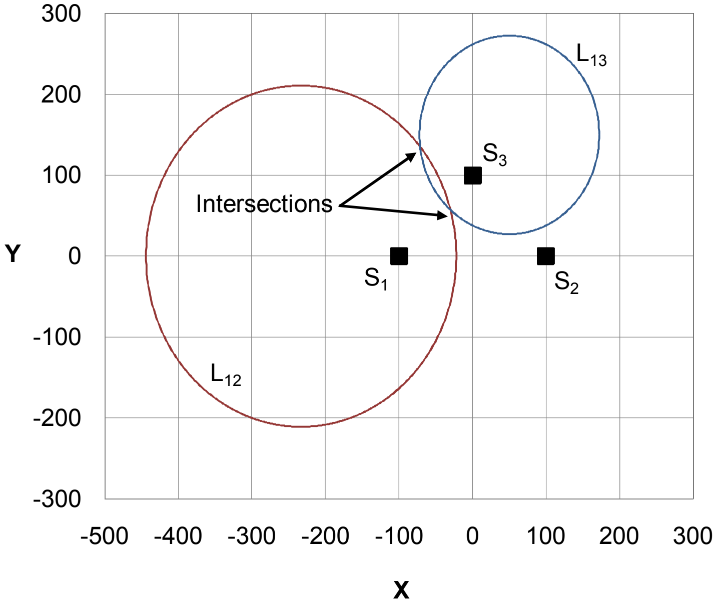
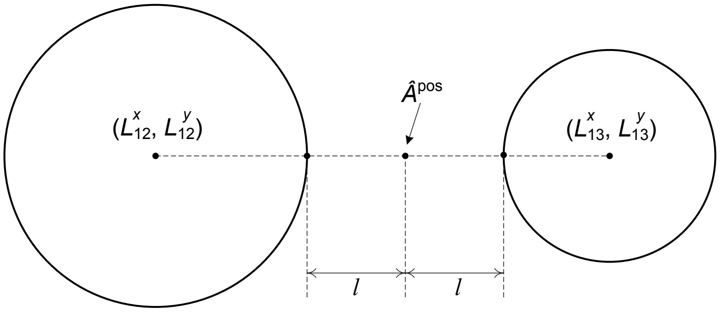


Figure 3.2. Solving the single source localization problem with RoSD.

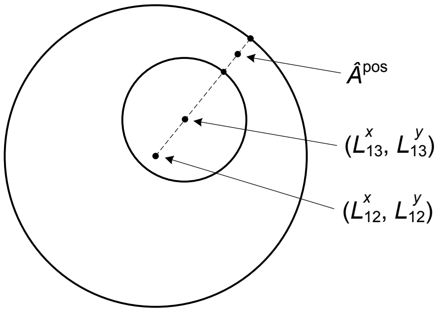
performance, RoSD finds the center of the closest points between the two loci when the loci do not intersect, and uses this center as the estimate. Figure 3.3 illustrates these cases.

Specifically, the solutions under non-intersecting loci are given by

$$\hat{A}^{\text{pos}} = (1-t)\,[L^x_{12} \;\; L^y_{12}]^\top + t\,[L^x_{13} \;\; L^y_{13}]^\top, \tag{3.7}$$



(a) The solution of RoSD algorithm when two loci are disjoint.



(b) The solution of RoSD algorithm when one locus is contained by the other locus.

Figure 3.3. RoSD enhancement for non-intersecting loci.

where

$$t = \begin{cases} \frac{1}{2\sqrt{C}} \left(\sqrt{C} + L^r_{12} - L^r_{13} \right) & \text{for} \quad L^r_{12} + L^r_{13} \leq \sqrt{C} \\ \frac{s}{2\sqrt{C}} \left(\sqrt{C} + L^r_{12} + L^r_{13} \right) & \text{for} \quad L^r_{12} + L^r_{13} > \sqrt{C} \end{cases} \tag{3.8}$$

$$s = \begin{cases} -1 & \text{for} \quad L^r_{12} + \sqrt{C} \leq L^r_{13} \\ 1 & \text{for} \quad L^r_{13} + \sqrt{C} \leq L^r_{12} \end{cases} \tag{3.9}$$

This improves the robustness of the algorithm when operating under realistic noise and measurement errors.

In addition to estimating the location of the radiation source, we estimate the strength of the radiation source by the following. With the source location estimate $\hat{A}^{\text{pos}}$, we have three source strength estimates

$$\hat{A}^{\text{str}}_i = I'_i \left(1 + \left| S_i - \hat{A}^{\text{pos}} \right|^2 \right) \tag{3.10}$$

for $1 \leq i \leq 3$. We combine these three estimates using coefficients that are inversely proportional to their variance estimates. As both the mean and variance are given by the estimator I'_i for the Poisson process, more weight is given to estimates with a lower variance. Thus, we have the following source strength estimator

$$\hat{A}^{\mathrm{str}} = \left(\sum_{i=1}^{3} (I'_i)^{-1} \right)^{-1} \sum_{i=1}^{3} \left(1 + \left| S_i - \hat{A}^{\mathrm{pos}} \right|^2 \right) . \tag{3.11}$$

3.2 Iterative Pruning Method for N Sensor Network

The Ratio of Square-Distance (RoSD) method with three sensors produces up to two position estimates of the source depending on the placement of the sensors and the location of the source. When there are two estimated positions, it is known that the ambiguity can be resolved by using four, instead of three, sensors [23] in the case of ideal deployment conditions without noise and errors. In a realistic deployment, however, noise and errors must be considered, in which case a larger number of sensors is needed to produce reliable localization results. Specifically, if N sensors are used, $\binom{N}{3}$ distinct subsets of three sensors are available, each of which will produce up to two source position estimates by RoSD. We refer to the set of position estimates produced by RoSD as the *candidate estimates* denoted by C. Using C, we design a fusion algorithm—one that is robust to noise and errors—to compute a *fused estimate* as the final result of the estimated source position.

A simplistic design for the fusion algorithm would be selecting the estimate with the smallest maximum distance to all the sensors in the surveillance area. This design gives preference to the candidate estimate that has the highest signal-to-noise (SNR) ratio. The rationale is that, by the inverse-square law, radiation sensors closer to the source will record stronger signals from the source (i.e., higher intensity measurements induced by the source) compared with sensors that are farther away from the source. The stronger signals are less vulnerable to noise and hence are more reliable. When phantom estimates are part of the candidate estimates, however, this reasoning may not always hold true. This is because a phantom estimate can appear to have a strong signal, which is, in fact, produced as a side effect of RoSD localization, and does not accurately approximate the actual source location. A solution to this problem, the design of a clustering algorithm that eliminates the phantom estimates with high probabilities, is the subject of Section 3.2.2.

3.2.1 Existing Fusion Methods

We now review two state-of-the-art fusion algorithms in the literature for radiation localization. Ajith Gunatilaka *et al.* [14] proposes the Maximum Likelihood Estimation (MLE) method to estimate the parameters of a radiation source, i.e., the 2D coordinates of the source location A^{pos} and the strength of the source A^{str}, by finding a solution that best fits the measurement data to the sensing model. Using a radiation propagation model similar to ours, the estimated radiation intensity is calculated. The main idea of MLE is to find values of $\hat{A}^{\text{pos}}$ and $\hat{A}^{\text{str}}$ that minimize the errors between the estimated radiation intensity and the actual measurements recorded by the sensors. This approach does not require a separate fusion step when scaling to $N > 3$ sensors, unlike our algorithm in Section 3.2.3, which treats the localization and fusion steps as separate problems. In MLE, finding the best-fit solution maps to a multi-dimensional optimization problem for which there are known existing solutions. Specifically, the `fminsearch` routine in MATLAB$^{©}$ is used, which implements the derivative-free downhill simplex method to solve the problem numerically. The disadvantage of this approach is that the result may not be the global optimum when phantom estimates are considered.

Rao *et al.* [13] proposes the Mean-of-Estimator (MoE) algorithm to fuse the candidate estimates. The MoE algorithm computes the fused estimate as the mean of all the candidate estimates. The advantage of the MoE algorithm is that it has linear time complexity and generally runs significantly faster than MLE. The main drawback is that the MoE algorithm is not explicitly designed to eliminate the phantom estimates in the fusion process. Phantom estimates can be detrimental to localization accuracy, particularly when they appear to be produced by strong (and hence presumably reliable) sensor measurements. As a result, the MoE algorithm can produce large localization errors when a significant fraction of the candidate estimates are phantom estimates.

3.2.2 Data Fusion with Phantom Estimates Eliminated by Clustering

Let us examine the characteristics of candidate estimates produced by RoSD. Assume an ideal deployment situation in which there is no noise due to background radiation, and the sensor measurements are accurate. As discussed in Section 3.1, RoSD may produce one or two candidate estimates with measurements from a group of three sensors, S_1, S_2, and S_3. In the case where there are two estimates, the phantom estimate is the one that is farther from the actual position of the source than the other estimate. Note that phantom estimates are produced not because the localization algorithm is inaccurate, but because the localization algorithm does not have sufficient information to compute an unequivocal solution. Suppose now that an additional sensor S_4 is available, and consider the group of three sensors S_1, S_2, and S_4. One or two candidate estimates are again produced. In the ideal deployment situation, one of the two estimates in each group (i.e., the group S_1, S_2, S_3 and the group S_1, S_2, S_4) will coincide at the actual position of the source. The other estimates, namely the phantom estimates, from the two groups are highly unlikely to coincide because two of the sensors from the groups are located differently. Thus, by counting the number of estimates at different positions, the position with the highest frequency would be the true position of the source.

In general, with N sensors, assuming the sensors are not colocated, there are $\binom{N}{3} = \frac{N}{6}\left(N-1\right)\left(N-2\right)$ groups of sensors available to perform RoSD localization, and each group produces a set of candidate estimates C_i for $i = 1, 2, \ldots, \frac{N}{6}\left(N-1\right)\left(N-2\right)$. At least one of the estimates in C_i is the true position of the source; thus there are at most $\binom{N}{3}$ estimates located at the true position of the source, and at most $\binom{N}{3}$ phantom estimates. The number of candidate estimates at the true position of the source may be less than $\binom{N}{3}$ because some groups of sensors may not produce any candidate estimates. Then, the source position can be estimated by $\hat{A}^{\mathrm{pos}} = \bigcap C_i$ for $N > 3$. With precise intensity measurements and lack of noise from background radiation, $N = 4$ is sufficient to accurately estimate the position of the source. Generalizing to many groups of three sensors, we see that in an ideal deployment situation, one of the estimates from every group will coincide with the actual source location, thus solving the localization problem redundantly.

When there are noise and errors, the candidate estimates are scattered. Hence, the non-phantom estimates will not coincide. The extent of the scattering depends on the magnitude of the noise/error. As a result, a simple intersection of the candidate estimates will likely produce a null set. Nevertheless, in spite of the scattering, the candidate estimates that are not phantom estimates will likely be close to each other and cluster around the true position of the source. On the other hand, the phantom estimates will not cluster together. Instead of finding the intersection of the candidate estimates, we can therefore "zoom in" to the actual position of the source by identifying the densest cluster of the estimates. Ideally, the densest cluster found does not contain any phantom estimates, such that an unbiased estimator of the candidate estimates in the cluster will give a fused estimate that closely approximates the actual position of the source. In practice, the phantom estimates may not be eliminated completely, but, if most of them are eliminated, their total contribution to the fused estimate will be insignificant.

In summary, we divide the localization problem using N sensors, under realistic noise and error conditions, into two solution steps:

1. Subproblem P1: *Clustering.* Find the smallest region in the surveillance area that contains most, if not all, candidate estimates that are not phantom estimates.

2. Subproblem P2: *Fusion.* Compute the fused estimate as an unbiased estimator of all the candidate estimates in the cluster found above.

3.2.3 Iterative Pruning (ITP) Clustering Algorithm

There exist clustering algorithms that are relevant to the Subproblem P1. Algorithms such as k-mean [24], CURE [25], and DBSCAN [26] are not directly applicable because they are concerned with classifying all samples into a number of closest clusters, whereas our problem aims to find the single largest cluster that is likely to contain most of the non-phantom estimates, and a few or none of the phantom estimates. Another algorithm, the Quality Threshold (QT) clustering algorithm [27], was invented by Heyer to cluster gene expression patterns. In their domain, the algorithm groups genes of high similarity into the same cluster. The measure of similarity is user defined, although the correlation of the gene expression is often used in their problem. The quality of the cluster computed is ensured by specifying a threshold such that all genes with a similarity measure within the threshold fall into the same cluster. Consider that there are n genes. The algorithm first builds n candidate clusters. It then outputs the largest cluster among the n candidates as a result. The genes produced are removed from the pool of n genes and the algorithm repeats the same step for the remaining genes. The candidate cluster is built by selecting the i-th gene from the pool, and then iteratively includes the most similar gene, the second most similar gene, and so on, until the similarity threshold is exceeded.

In our problem domain, the genes can be interpreted as our candidate estimates, and the similarity measure corresponds to the distance between the candidate estimates. The QT algorithm then basically associate each candidate estimate (the gene) with a cluster such that the diameter of the cluster does not exceed d (the similarity threshold of QT clustering algorithm). In contrast to the QT algorithm, we are interested only in the largest cluster in our problem, and therefore do not need to iteratively find the next largest cluster as needed in the original algorithm.

While useful for our purpose, the main disadvantage of QT clustering is that the time complexity is very high, $O\left(|\mathbf{C}|^4\right)$, even for finding only the maximum size cluster. For our problem, the algorithm does not scale to a large network (i.e., the number of sensors N is large and there are $O(N^3)$ samples for clustering) required for, say, the monitoring of large city areas. We now propose an algorithm, which we call the Iterative Pruning (ITP) algorithm, for our problem. The proposed algorithm can achieve similar performance as QT clustering, but has a greatly reduced time complexity. Specifically, ITP has a worst case time complexity of $O\left(|\mathbf{C}| \log_2 A\right)$, where A denotes the area of the surveillance region.

The ITP algorithm employs two steps. First, it solves the Subproblem P1 heuristically by pruning the low density space of candidate estimates in the surveillance area, such that a cluster with a high density of the estimates remains. Second, it solves the Subproblem P2 by computing the weighted centroid of the candidate estimates in the cluster as the fused estimate. The pseudo-code of the ITP algorithm is given in Algorithm 1. The algorithm takes three input parameters. The $\mathbf{C}$ parameter is the union of all the candidate estimates $\mathbf{C} = \bigcup C_i$ produced by the RoSD algorithm. The n and d parameters limit the maximum number of estimates in the remaining region, and the maximum size (in terms of area) of the smallest region, respectively.

The algorithm begins with the smallest rectangle that bounds all candidate estimates. The bounding rectangle is then divided into 5 regions $r_a = r_0 \cup r_1 \cup r_4 \cup r_5$, $r_b = r_2 \cup r_3 \cup r_6 \cup r_7$, $r_c = r_0 \cup r_2 \cup r_4 \cup r_6$, $r_d = r_1 \cup r_3 \cup r_5 \cup r_7$, $r_e = r_4 \cup r_5 \cup r_6 \cup r_7$, and 8 subregions $r_0, r_1, \ldots, r_7$. The subregion labels are shown in Figure 3.4. The number of candidate estimates in each of the sub-regions is tallied, and the region containing the most estimates is selected for the next iteration.

The algorithm continues until the number of candidate estimates remaining is less than n and the area of the bounding rectangle is smaller than d. In each iteration, the size of the bounding rectangle is reduced by at least half. This ensures that the algorithm will terminate in $O\left(\log_2 A\right)$ iterations. ITP divides the space into five overlapping regions, unlike existing partitioning algorithms such as the generalized

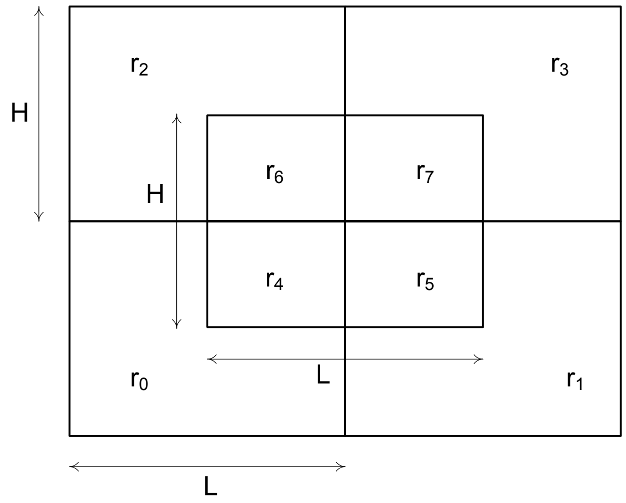
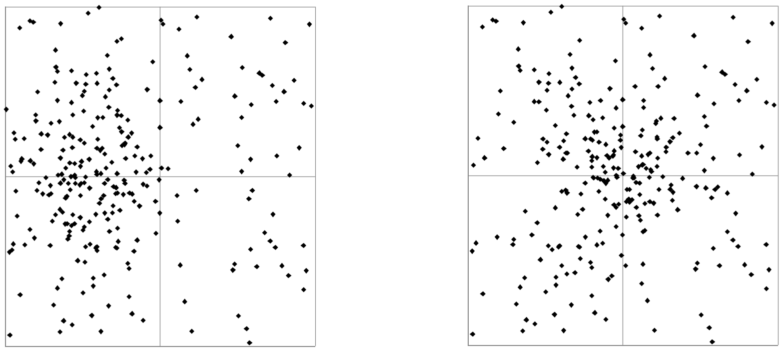


Figure 3.4. Region division in the ITP algorithm.

bisection method in [28], which iteratively divides the search space in halves. The overlapping minimizes the chance that the algorithm incorrectly prunes a region due to a cluster concentrating near the division boundaries. For instance, Figure 3.5 shows two scenarios in which non-overlapping partitioning fail. In particular, Figure 3.5(a) shows a scenario in which the candidate estimates are concentrated at the center of r_c. If a non-overlapping region division is used, the algorithm may choose either $r_0 \cup r_4$ or $r_2 \cup r_6$. Since the numbers of candidate estimates in both regions are close, however, it is better to choose $r_0 \cup r_2 \cup r_4 \cup r_6$ for the next iteration.

In the second part of the algorithm (Lines 22 to 26), the weighted center of the cluster is computed as the solution to the Subproblem P2. The weighting gives estimates produced by sensors closer to the radiation source higher weights because their measurements are less influenced by background noise. Although this is similar in concept to the use of SNR in wireless communication, there are subtle differences. In wireless communication, the SNR provides a measure of how intrusive the background noise is compared with the goodness of the signal received. In particular, the SNR quantifies the likelihood that the data received will be correctly decoded in spite of the noise. These quantities can be directly measured at the receiver side. In our



(a) Cluster located at the division of two regions.

(b) Cluster located at the center, division of four regions.

Figure 3.5. Two examples illustrating that non-overlapping region division may lose accuracy when the cluster is located near the division of regions.

problem, the weighting of a candidate estimate is a function of the SNRs of the three measurements that produced the candidate estimate. Essentially, the weighting of a candidate estimate measures the goodness of the estimate given the goodness of the measurements by the individual sensors. We use the average SNRs of all the three measurements as the weighting of the candidate estimate. Formally, we define the weighting of a candidate estimate as

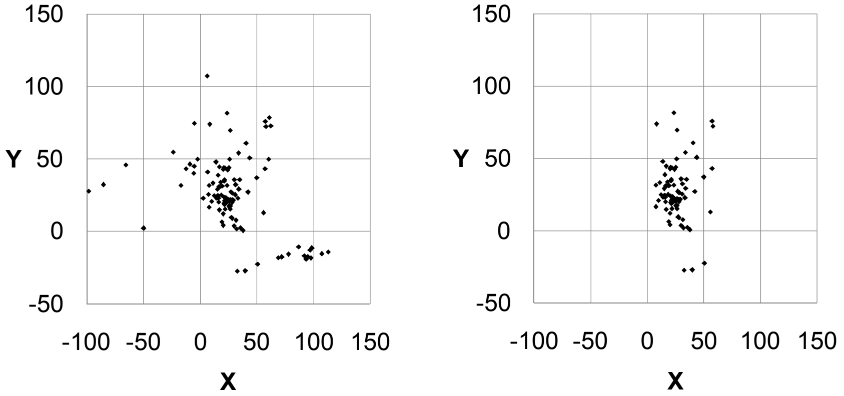
$$\mathrm{SNR}\left(c\right) = \frac{\sum_{x=\{i,j,k\}} m\left(S_x\right)}{\sum_{x=\{i,j,k\}} B_x} \tag{3.12}$$

where S_i, S_j, S_k are the three sensors that produce the estimate c.

Figure 3.6 shows a sample output tracing selected iterations of the ITP algorithm. The candidate estimates fed into the algorithm (Figure 3.6(a)) are iteratively pruned such that the region having the largest number of estimates remains after each iteration. The algorithm stops when the number of estimates remaining is less than n and the area of the region is smaller than d. Then, the weighted center of the remaining estimates is computed as the fused estimate. As the number of sensors increases, the candidate estimates produced increases as $O\left(N^3\right)$. As a result, the algorithm can identify a small region with a high density of candidate estimates as the most likely location to contain the radiation source. The experiment results in Section 5.2 confirm that the accuracy of ITP increases with the number of sensors used.

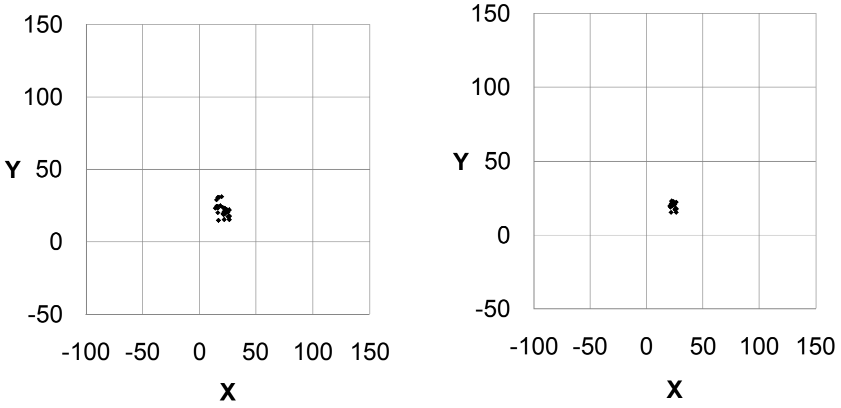
Algorithm 1: ITP ($\mathbf{C}$, n, d)

```
$r_S$ = smallest rectangular region that bounds all estimates $c \in \mathbf{C}$.
Let $b$ and $t$ denotes the bottom-left and top-right coordinate of $r_S$.
Divide $r_S$ into 8 regions $r_i$ for $i = 0, 1, \ldots, 7$ as in Figure 3.4.
Let $k_w$ denotes the number of candidate estimates in region $w$.
Initializes $k_w = 0$, for $w = r_0, r_1, \ldots, r_7$.
foreach $c \in \mathbf{C}$ do
    for $i = 0$ to $7$ do
        if $c$ is inside $r_i$ then
            $k_{r_i} = k_{r_i} + 1$
Let $r_a = r_0 \cup r_1 \cup r_4 \cup r_5$; Let $r_b = r_2 \cup r_3 \cup r_6 \cup r_7$; Let $r_c = r_0 \cup r_2 \cup r_4 \cup r_6$
Let $r_d = r_1 \cup r_3 \cup r_5 \cup r_7$; Let $r_e = r_4 \cup r_5 \cup r_6 \cup r_7$
$r_S = \underset{w \in \{r_a, r_b, r_c, r_d, r_e\}}{\arg\max} k_w$
if $r_S = r_a$ then $t.y = 0.5\,(t.y + b.y)$
else if $r_S = r_b$ then $b.y = 0.5\,(t.y + b.y)$
else if $r_S = r_c$ then $t.x = 0.5\,(t.x + b.x)$
else if $r_S = r_d$ then $b.x = 0.5\,(t.x + b.x)$
else
    $b.x = 0.25\,(t.x + b.x)$; $b.y = 0.25\,(t.y + b.y)$; $t.x = 0.75\,(t.x + b.x)$; $t.y = 0.75\,(t.y + b.y)$
if $|r_S| > n$ or $(t.x - b.x)\,(t.y - b.y) > d$ then
    return ITP ($\forall\, c \in r_S$, $n$, $d$)
else
    $e = (0, 0)$, $s = 0$
    foreach $c \in r_S$ do
        $e.x = e.x + c.x \times \mathrm{SNR}\,(c)$; $e.y = e.y + c.y \times \mathrm{SNR}\,(c)$
        $s = s + \mathrm{SNR}\,(c)$
    return $\left(\frac{e.x}{s}, \frac{e.y}{s}\right)$
```



(a) All candidate estimates

(b) Iteration 2



(c) Iteration 6

(d) Final Result

Figure 3.6. The candidate estimates are pruned in each iteration until the remaining estimates are clustered within *d* diameter.

4 MULTIPLE RADIATION SOURCE LOCALIZATION

In the previous chapter, we have considered the localization of a single radiation source in a surveillance area without obstacles. The concepts and algorithms presented there pave the way to a more realistic scenario in this chapter—the localization of multiple radiation sources ($K \geq 1$) in a surveillance area with or without obstacles. We first discuss the challenges in localizing multiple radiation sources in Section 4.1, then we propose an algorithm that solves those challenges. The algorithm we propose is a hybrid formulation of a particle filter with a mean-shift technique. We first present an overview of particle filter in Section 4.2, then describe the formulation of the algorithm in Section 4.3.

4.1 Challenges in Multiple Source Localization

Localization of multiple sources is an important challenge in radiation source localization. A multiple radiation source scenario may occur when multiple radiation sources are brought into an urban area in a coordinated RDDs attack. In this scenario, a single source localization algorithm is ineffective for the following reasons. First, a single source localization algorithm implicitly assumes that there is only one radiation source present in the surveillance area. However, this assumption is not valid when there are multiple sources present in the surveillance area. Second, a single source localization algorithm is unable to distinguish between multiple sources due to the superposition of the radiation intensity. In such a case, the single source localization algorithm may identify a single, combined source located at the centroid of all individual radiation sources.

When dealing with multiple sources, existing localization approaches attempt to build a model of all the sensors and sources. The model is then used to find the most probable source parameters, e.g., the location and strength of each source [29–31]. When the number of radiation sources K increases, the number of estimation parameters in the model increases proportionally and thus the algorithm complexity grows exponentially with regard to K.

In addition, because the number of sources K is not known *a priori*, the value of K needs to be estimated as well. Typically, a statistical test is used to select the most probable number of sources prior to estimating the parameters. Alternatively, some algorithms estimate the parameters of the sources for a range of assumed values of K, i.e., $K = 1, 2, \ldots$. Both approaches are computationally intensive. Moreover, they are inaccurate in large sensor networks due to: (i) the ability of K to assume any value, but that value is unknown; (ii) superposition of signal strengths from different sources; and (iii) large variances in sensor measurements caused by background radiation or other interfering sources. These factors have been mainly addressed individually by the existing algorithms, but the models of these algorithms become significantly more complex when all factors are jointly considered. For instance, a high measurement recorded by a sensor can be induced by a single strong source, or it may be induced by the combined strengths of multiple weak sources. Consequently, we may obtain equivocal results that appear equally valid for different values of K. Even if the estimate of K is accurate, multiple ambiguous solutions may exist due to the model dynamics, e.g., non-linear signal fading and a large number of interacting system parameters.

Furthermore, when taking obstacles into consideration such as building structures in urban areas, building an accurate model becomes infeasible unless the shape, size, material and location of all obstacles are accurately modeled. These data are difficult to obtain and incorporating such data into the model is a non-trivial process. Estimating the obstacle parameters is not an option because obstacles can come in any shape, size, or material and can be located anywhere in the area.

In this chapter, instead of building a model encompassing all sensor, source, and obstacle parameters, we propose a hybrid formulation of a particle filter with a mean-shift technique to address the challenges faced by the existing algorithms described above. In our formulation, we first generate hypotheses about the source parameters (known as *particles*). When a sensor measurement is received, the likelihoods of these particles being one of the radiation sources are evaluated. Unlike standard particle filter formulation, we *selectively* evaluate these particles based on the location of the sensor providing the measurement. Selective evaluation is a major feature in our approach that enables localization of multiple sources in a complex environment. After evaluation, the particles are refined to better estimate the location and strength of a source. To compute the fused estimate, the particles are used to construct a likelihood function. Then, a mean-shift technique is used to determine the most likely parameters of all sources. Because the mean-shift technique determines the parameters of all sources without knowing the number of sources K, we eliminate the expensive statistical test to estimate K. The above process is repeated when a new sensor measurement is received.

4.2 Overview of Recursive Bayesian Estimation and Particle Filter

Recursive Bayesian estimation is widely used to estimate a set of variables (called the *state variables*), given a set of observations. The objective is to find the best estimate of the state variables considering all available observations, and continuously refine the estimate as more observations become available.

In the formulation, at iteration t, the state to be estimated is represented by the state vector x_t, and the information acquired is represented by y_t. All available information at t is thus $D_t = \{y_i \mid i = 1 \ldots t\}$. At each iteration t, the state x_t is first predicted based on D_{t-1} using the prior probability distribution function (PDF),

$$\mathrm{P}\left(x_t \,|D_{t-1}\right) = \int \mathrm{P}\left(x_t \mid x_{t-1}\right) \mathrm{P}\left(x_{t-1} \,|D_{t-1}\right)\, dx_{t-1}. \tag{4.1}$$

The above equation captures the change in state over time during the estimation, e.g., when estimating the position of a moving vehicle. Then, as new information becomes available at t, the posterior PDF is computed via Bayes rule as

$$\mathrm{P}\left(x_t \,|D_t\right) = \frac{\mathrm{P}\left(y_t \mid x_t\right) \mathrm{P}\left(x_t \,|D_{t-1}\right)}{\int \mathrm{P}\left(y_t \mid x\right) \mathrm{P}\left(x \,|D_{t-1}\right) \, dx}. \tag{4.2}$$

The estimated state $\hat{x}_t$ is then computed by finding the expectation of Equation (4.2),

$$\hat{x}_t = \mathrm{E}\left[\mathrm{P}\left(x_t \,|D_t\right)\right] \ . \tag{4.3}$$

The above formulation is general and could be applied to any state estimation problem. When the system and measurement models are linear, and the noises are Gaussian of known variances, an analytical solution of the above can be found, which is the Kalman filter. In many cases, however, analytical solutions cannot be found and thus the procedure has practical limitations.

The particle filter approaches the Bayesian estimation by using random samples (so called *particles*) to approximate the PDF. Larger number of particles improves the accuracy of the approximation. A main feature of particle filter is that it can be applied to non-linear, non-Gaussian problems even if there is no general analytic expression for the required PDF.

Particle filter works in two phases—*predict* and *update*. In the *predict* phase, the prior PDF is estimated by evolving the particles from $t-1$ to predict the state at iteration t. Then, the *update* phase evaluates the probability that these particles represent the state given the prior observations. In other words, each particle x_i is associated with a weight w_i. In the end, we obtain a discrete distribution defined over all the particles $\{(x_1, w_1), \ldots, (x_N, w_N)\}$. Then, the particles are resampled into $\{x'_1, \ldots, x'_N\}$ such that $\mathrm{P}\left(x'_j = x_i\right) = w_i$ for any j. The state estimate is computed as the expectation of all particles, and the process is repeated when the next measurement arrives.

In a straightforward application of particle filter in radiation source localization, one would consider the source parameters, namely the xy-coordinates and the source strength, to be the state variables. The radiation intensity measurements then provide the data to drive the *predict* and *update* phases of the particle filter. For each particle, the expected radiation intensity on each sensor is computed according to Equation (2.4). Given the expected radiation intensity, the probability of obtaining the particular set of sensor measurements is calculated and is assigned as the weight of the particle. Over time, the expectation of all the particles gives an accurate estimate of the source parameters. This simple approach performs relatively well in scenarios with only a single radiation source. However, with multiple sources, the number of state variables increases proportionally. As each radiation source is modeled by three parameters, K radiation sources require $3K$ state variables. A linear increase in the number of parameters causes the parameter space to increase exponentially [32]. For instance, if there are K sources and each source can be located at m different places, the number of possible solutions is $\binom{m}{K} = O\left(m^K\right)$. Consequently, the number of particles needs to be increased exponentially in order to maintain a representative PDF for the estimation. As a result, this approach does not scale to a large number of sources. Furthermore, this approach also requires the number of sources to be known ahead of time, which limits its applicability in many realistic situations.

Another problem with the above approach, or with any existing radiation localization algorithm, is that the detailed parameters of obstacles are assumed to be known beforehand, in order for the model in Chapter 2 to be applied. These include the shape, location, and attenuation coefficient for each obstacle. Incorporating such data into the localization system is a burden to the user. Moreover, the attenuation coefficients of the obstacles need to be measured on-site unless accurate blueprints of the deployment buildings, showing construction materials and layouts, are readily available. Without that complete data, the expected intensities from the sources cannot be computed because radiation sources can induce a wide range of readings on radiation sensors depending on the quantity and variety of obstacles located between the sources and the sensors.

4.3 Algorithm Design

In this section, we describe our algorithm for localizing multiple sources within a complex environment; in the presence of unknown or incompletely known obstacles, using source strength measurements from a network of sensors. Similar to recursive Bayesian estimation, the proposed algorithm recursively refines the location and strength estimates of all the sources based on newly acquired measurements as well as all prior information. However, as opposed to a typical recursive Bayesian estimation formulation, our algorithm (i) considers one sensor measurement at a time, (ii) does not explicitly model each radiation source, and (iii) uses mean-shift technique to compute source parameters. Figure 4.1 outlines the flow of the algorithm. The algorithm starts by initializing a collection of particles, each of which hypothesizes the location and strength of a source. In each iteration, as a new sensor measurement becomes available, the algorithm predicts the location and strength of each radiation source based on the previous estimates. Then, the weights of all the particles are updated according to the newly received sensor measurement and the prior weights. After weighting the particles, the source parameter estimates are computed. Finally, a resampling procedure normalizes the weights of all the affected particles. This procedure then repeats when another new sensor measurement arrives.

The highlights of our formulation are as follows:

- We only consider one sensor measurement in each iteration as opposed to a traditional formulation that considers all sensor measurements in one iteration. This allows the algorithm to proceed as soon as possible, rather than waiting until all the sensor measurements have arrived. This feature is particularly relevant in wireless sensor networks because wireless transmissions are inherently unreliable due to environmental factors, low transmission power, and malfunctions of unreliable sensors. Moreover, network latency is usually high due to multi-hop wireless forwarding and signal interference among a large number of communicating sensor nodes. By allowing the algorithm to proceed as soon

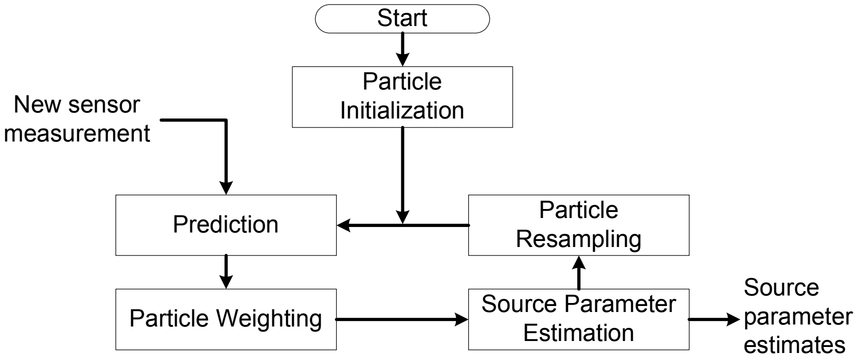


Figure 4.1. Overview of the hybrid particle filter with mean-shift algorithm.

as *any* sensor measurement is received, even if the arrival is out of order, the robustness of the algorithm is improved.

- Instead of explicitly modeling each radiation source, which is typically done [30, 33] in recursive Bayesian estimation, we compute the parameter estimates without attributing them to specific sources. When multiple estimates are resolved to the same location parameter, we classify them as the same source, and multiple sources are distinguished when their estimates resolve to different locations.

- We introduce the notion of *fusion range* for use in the particle filter, in which particles are selectively updated based on the locations of the sensors. As each particle can only estimate one of the many possible radiation sources, selectively updating the particle in each iteration essentially prevents the algorithm from using a particle to estimate multiple sources.

- We apply a mean-shift technique on the particles to compute the source parameter estimates. Using mean-shift is crucial in order to recover the parameters of multiple sources because our formulation does not explicitly model the individual sources. This is an important departure from the traditional use of particle filter to compute the estimates. Details of integrating mean-shift into our particle filter formulation are described in Section 4.3.4.

Combining the above, our algorithm solves the multiple radiation source localization problem without (i) knowing the number of sources, (ii) increasing the dimension of the parameter space as the number of sources increases, and (iii) knowing all the obstacles in detail. Detailed algorithm steps are described in the following sub-sections.

4.3.1 Particle Initialization

We define the initial iteration to be $t = 0$. At this time, the particle filter is initialized as follows. Given a surveillance area, let

$$\mathcal{P} = \left\{ \mathbf{p}_i^{(t_i)} \mid i = 1, ...N_{\mathcal{P}} \right\} \tag{4.4}$$

be a set of particles. Each particle $\mathbf{p}_i^{(t_i)} \in \mathcal{P}$ is a three-value vector in the parameter space $\mathbb{A}$ denoting the hypothesized source position and strength. The superscript t_i is an integer that denotes the iteration at which the particle is updated. We drop the superscript in cases where no ambiguity exists in the iteration number. We initialize $\mathcal{P}$ with uniform random particles, i.e., $\forall\ \mathbf{p} \in \mathcal{P}, w(\mathbf{p}) = \frac{1}{N_{\mathcal{P}}}$ where $N_{\mathcal{P}} = |\mathcal{P}|$ denotes the number of particles in $\mathcal{P}$. Uniform random distribution is used because we do not assume any *a priori* knowledge about the location or the strength of the source. If prior knowledge is available, the particles can be initialized instead in accordance with the pre-existing distribution. Doing so shortens the number of iterations required to obtain accurate estimates of the radiation sources.

The value of $N_{\mathcal{P}}$ affects the coverage of $\mathcal{P}$ in $\mathbb{A}$. A larger coverage gives a higher probability that some particles will be close to the actual radiation source and have similar strengths. Consequently, we can achieve more accurate and reliable location and strength estimates. Theoretically, one would want $N_{\mathcal{P}} \to \infty$ to cover the entire space $\mathbb{A}$, which is computationally infeasible. In practice, a modest number of particles (e.g., a few thousand) would provide an acceptable coverage for realistic applications.

We associate each particle $\mathbf{p} \in \mathcal{P}$ with a weight $w(\mathbf{p})$ such that

$$w(\mathbf{p}) \approx \sum_j \mathrm{P}(\mathbf{A}_j = \mathbf{p}) \tag{4.5}$$

and $\sum_{\mathbf{p} \in \mathcal{P}} w(\mathbf{p}) = 1$. As such, this weight measures the probability that an actual radiation source has the same parameters as $\mathbf{p}$. This weight is updated as a new sensor measurement becomes available to refine the source estimates. However, as the actual source position is unknown, $\sum_j \mathrm{P}(\mathbf{A}_j = \mathbf{p})$ cannot be computed. In Section 4.3.3, we will discuss in detail how we can approximate this probability using sensor measurements. During initialization, we assign uniform weights to all the particles assuming no prior knowledge about the radiation sources is available.

4.3.2 Prediction

After initializing the particles, the localization process can begin. Our algorithm refines the source estimates whenever there is a new sensor measurement. At a particular iteration, say t, a sensor located at S_i delivers a measurement $m(S_i)$. Then, we use this measurement to update our source estimates as follows. We construct a set $\mathcal{P}'$ such that all particles in $\mathcal{P}'$ are less than d_i distance away from S_i. This distance d_i is defined as the *fusion range*, and is specific to S_i. Formally, $\mathcal{P}'$ is defined as

$$\mathcal{P}' = \left\{ \mathbf{p}_i \,\middle|\, |S_i - p_i^{\mathrm{pos}}|^2 \leq d_i^2 \right\}. \tag{4.6}$$

The value of d_i is selected such that a particle is located within the fusion range of a handful of sensors. The fusion range controls the particles affected in the current iteration. It prevents the current update from affecting particles that are far away from the sensor. Because of the inverse square fading radiation strengths in open space (i.e., even if there are no further attenuating effects due to obstacles), sources that are far away will not make significant contributions to the sensor measurements. By limiting the affected particles, we would not use the particles to distinguish between different sources explicitly. Thus, the number of estimation parameters in our formulation, as well as the complexity of our algorithm, remain constant regardless of the number of sources.

Let us illustrate what would happen if the fusion range is not used. In this case, all particles will be affected in each iteration by a typical particle filter formulation. Figure 4.2 illustrates the behavior of the algorithm without the fusion range. When multiple sources exist, the particles will gravitate toward a source when the sensors near the source send updates. In Figure 4.2, the particles are concentrated near source B starting at iteration $t = 108$. The sensor located at (0,0) sends an update at $t = 109$ and is followed by the sensor located at (0, 20) . . . (0, 100). At $t = 113$ as shown in Figure 4.2(b), the particles are moving away from source B. After the update from the sensor at (20, 100), particles are concentrated near source A as shown in Figure 4.2(c). When sensors near source B update, the particles, again, move away from source A as shown in Figure 4.2(d). This example illustrates that the fusion range limits the interference between sources in our algorithm. Without the fusion range, the estimated source position will oscillate between source A and B.

Given a set of particles in $\mathcal{P}'$ updated prior to t, we aim to predict the whereabout of the source at iteration t using a user-defined update model,

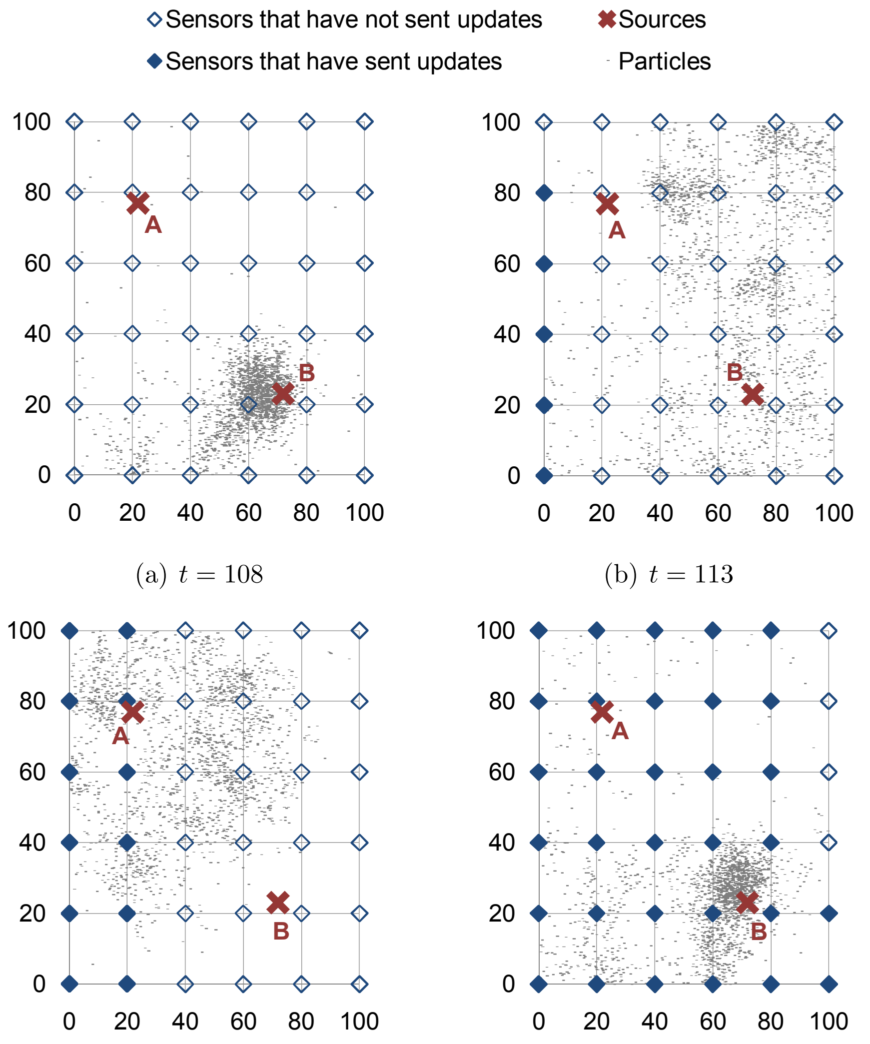
$$\mathcal{F}_{\text{movement}} : \mathbb{A} \rightarrow \mathbb{A}. \tag{4.7}$$

This update model predicts the value of $\mathbf{p}^{(t)}$ given $\mathbf{p}^{(t')}$ for any $t' < t$. This model captures the knowledge about how a radiation source changes over time. For instance, the model may capture the movements of a radiation source in transport. For a static radiation source, the parameters do not change over time. Thus, the movement model is simply

$$\mathcal{F}_{\text{movement}}\left(\mathbf{p}\right) = \mathbf{p}. \tag{4.8}$$

We update all the particles in $\mathcal{P}'$ by applying the movement model to each particle and produce the set

$$\mathcal{P}'' = \left\{ \mathbf{p}_i^{(t)} \,\middle|\, \mathbf{p}_i^{(t)} = \mathcal{F}_{\text{movement}}\left(\mathbf{p}_i^{(t')}\right), \forall\, \mathbf{p}_i^{(t')} \in \mathcal{P}' \right\}. \tag{4.9}$$



(a) $t = 108$ (b) $t = 113$

(c) $t = 120$ (d) $t = 140$

Figure 4.2. Progression of a particle filter without fusion range.

The set $\mathcal{P}''$ now consists of potential position and strength estimates of radiation sources given their prior estimates.

4.3.3 Particle Weighting

After predicting the whereabouts of radiation sources at iteration t, we compute the weight of each particle, which is the likelihood that an actual radiation source is located at where the particles are and also has the predicted source strength defined above. Although we do not know the actual locations and strengths of the radiation sources, we can estimate this probability using the sensor measurements. Assuming that the radiation source is at $\mathbf{p}$, we can compute the expected sensor measurement at S_i using Equation (2.4). The probability of obtaining the measurement $m(S_i)$ given the source is at $\mathbf{p}$ follows the Poisson distribution. Subsequently, we update the weight of each particle in $\mathcal{P}''$ such that

$$w\left(\mathbf{p}_i^{(t)}\right) = \mathrm{P}\left(m\left(S_i\right) \,\middle|\, \mathbf{p}_i^{(t)}\right) w\left(\mathbf{p}_i^{(t')}\right). \tag{4.10}$$

After updating the weights, we merge $\mathcal{P}''$ with $\mathcal{P}$ to form a global view of all particles. Let $\mathcal{P}'''$ denote the merged set, $\mathcal{P}'''$ is defined as

$$\mathcal{P}''' = (\mathcal{P} - \mathcal{P}') \bigcup \mathcal{P}''. \tag{4.11}$$

The weights of all particles are normalized again after the merging process such that $\sum_{\mathbf{p}\in\mathcal{P}'''} w(\mathbf{p}) = 1$.

This merging process plays an important role in enabling the particle filter to localize multiple sources without increasing the number of estimation variables. In a traditional particle filter formulation, where all the particles are updated in each iteration, each sensor measurement affects all the particles. This includes particles far away from the sensor that may not be relevant to current sensor measurement, or particles estimating mainly other radiation sources. Furthermore, the sensor measurements close to those particles (which will be received in the future) will provide a better view at those locations. By updating particles that are close to the sensor providing the measurement only, we allow multiple sources to co-exist in the area, without requiring separate variables for different sources. Equally important, this allows our algorithm to localize multiple sources without knowing the number of sources.

4.3.4 Source Parameter Estimation

Given $\mathcal{P}'''$ and associated weights, we now compute the position and strength estimates of the radiation sources. This step is very different from the traditional particle filter approach, in which the estimate is computed as the weighted sum of all the particles. In our case, if the weighted sum of all the particles was computed, we would not be able to differentiate multiple sources. Instead, we would obtain the centroid of the sources.

To obtain the parameter estimates for individual sources, we construct the probability distribution function

$$\begin{aligned} L_{\mathcal{P}'''}(\mathbf{x}) &= \mathrm{P}\left(\mathbf{x} \mid \mathcal{P}'''\right) \\ &= \left(\textstyle\sum_{\mathcal{P}'''} w\left(\mathbf{p}_i\right)\right)^{-1} \textstyle\sum_{\mathcal{P}'''} w\left(\mathbf{p}_i\right) \phi_{\mathbf{H}}\left(\mathbf{x}-\mathbf{p}_i\right). \end{aligned} \tag{4.12}$$

In the above, $\phi_{\mathbf{H}}$ is the Gaussian kernel function

$$\phi_{\mathbf{H}}(\mathbf{x}) = (2\pi)^{-3/2} \left|\mathbf{H}\right|^{-1/2} \exp\left(-\tfrac{1}{2}\mathbf{x}^{\top}\mathbf{H}^{-1}\mathbf{x}\right) \tag{4.13}$$

and $\mathbf{H}$ is a symmetric positive definite matrix denoting the kernel bandwidth [34]. In theory, the function $L_{\mathcal{P}'''}(\mathbf{x})$ is a mixture distribution, where the number of mixture components represent the number of radiation sources. Estimating the source parameters, therefore, is equivalent to finding the distribution parameters for each mixture component. This is accomplished by finding the values of $\mathbf{x}^*$ that maximize $L_{\mathcal{P}'''}$, i.e., $\nabla L_{\mathcal{P}'''}(\mathbf{x}^*) = \mathbf{0}$.

Note that multiple solutions exist (i.e., local optima) when there are multiple sources. We are interested in finding all the solutions, each of which corresponds to the estimate of one radiation source. This can be accomplished efficiently by using the mean-shift algorithm [34]. The algorithm does not impose *a priori* assumptions on the number of optimum points or the underlying shape of L. Using mean-shift, we find $\mathbf{x}^*$ by repeatedly applying $\mathbf{x}_{i+1} = M(\mathbf{x}_i)$ for $i = 1, 2, \ldots$ until convergence, where $M(\mathbf{x})$ is the mean-shift term derived from $\nabla L_{\mathcal{P}'''}(\mathbf{x})$ and given by

$$M(\mathbf{x}) = \frac{\sum_{\mathcal{P}'''} \mathbf{p}_i\, w\left(\mathbf{p}_i\right) \phi_{\mathbf{H}}\left(\mathbf{x}-\mathbf{p}_i\right)}{\sum_{\mathcal{P}'''} w\left(\mathbf{p}_i\right) \phi_{\mathbf{H}}\left(\mathbf{x}-\mathbf{p}_i\right)}. \tag{4.14}$$

Given an arbitrarily selected initial value of $\mathbf{x}_1$, the mean-shift algorithm will converge to the closest $\mathbf{x}^*$. To find all the radiation sources, we repeat the algorithm with different values of $\mathbf{x}_1$ and merge all the results that converge to the same $\mathbf{x}^*$.

The advantages of using the mean-shift algorithm in estimating the source parameters are: (i) the algorithm does not require knowledge of the number of sources when combined with our particle filter formulation, and (ii) the algorithm is robust under noisy sensor data.

4.3.5 Particle Resampling

The last step of our algorithm aims to prepare the particles for the next iteration. This step involves removing the particles of low weights and replacing them with particles of higher weights. It is often known as *resampling* in the literature. Resampling is important because it prevents the particle filter from hitting degeneracy problems. As an example, consider a scenario with a single static radiation source. We initialize $N_{\mathcal{P}}$ particles randomly in the space $\mathbb{A}$. Without resampling, in each iteration, the particles closer to the radiation source will have increasingly larger weights whereas the particles farther away from the radiation source will have increasingly lower weights. Over time, all the particles except for the particles closest to the radiation source will have a weight of zero. In this case, the particle filter essentially degenerates to a single particle. The parameter estimation will not be refined any further due to the lack of diversity in the particles. The resampling step essentially solves this problem by removing particles of lower weights and multiplying particles of higher weights. In our algorithm, resampling is only performed on particles that are affected in the current iteration, which are in $\mathcal{P}''$. Particles that are not affected in the current iteration do not require resampling because they remained unmodified after previous resampling.

The resampling step constructs the set $\mathcal{P}^{\text{resampled}}$ by sampling with replacement from $\mathcal{P}''$ with probabilities $\frac{w(\mathbf{p}_i)}{\sum_{\mathbf{p}_j \in \mathcal{P}''} w(\mathbf{p}_j)}$ for all $\mathbf{p}_i \in \mathcal{P}''$. The resampled particles are assigned uniform weights. After resampling, we merge $\mathcal{P}^{\text{resampled}}$ with $\mathcal{P}$ by

$$\mathcal{P}^{\text{new}} = (\mathcal{P} - \mathcal{P}') \bigcup \mathcal{P}^{\text{resampled}} \tag{4.15}$$

to form the set of particles $\mathcal{P}^{\text{new}}$ for the next iteration.

Although the resampling procedure mitigates the degeneracy problem, it does not completely solve the problem. Particles with higher weights may be duplicated multiple times and these particles will eventually collapse to a single point. Several approaches have been introduced in the literature to mitigate this further problem [35]. In our algorithm, we adopt the roughening method [36], which introduces noise in the duplicated particles. During resampling, when a particle is duplicated, we introduce zero-mean Gaussian noise into the duplicated particles. The standard deviation of the noise is a tunable parameter given by σ_N.

The above resampling procedure generates particles with parameters similar to the actual radiation sources, at the same time eliminating particles that do not correspond to any actual radiation sources. As time proceeds, areas with no radiation sources at $t = 0$ will have few or no particles. If a new radiation source moves into the surveillance area and is in one of these isolated locations, the radiation source will be undetected because Equation 4.6 produces a null set. As a provision for new radiation sources entering the area, we randomly replace a small percentage of particles, e.g., 5%, with randomly generated particles. This ensures that the surveillance area is sufficiently covered and new radiation sources will be detected and localized quickly.

5 EVALUATION OF LOCALIZATION ALGORITHMS

We conduct simulation and testbed experiments to evaluate the practicality and effectiveness of the proposed algorithms. Most of the results presented are based on simulation because actual experimentation with source intensities similar to that of a concealed dirty bomb is not safe. First, in Section 5.1, we evaluate the performance of the RoSD algorithm in localizing a single radiation source using 3 sensors. Then, in Section 5.2, we use a similar setup with an increased number of sensors to evaluate the performance of the ITP algorithm. In Section 5.3, we perform experiments in a small-scale testbed to evaluate the practicality of the algorithm. In addition, we construct an identical setting in a simulation environment to validate the accuracy of the models presented in Chapter 2. In Section 5.4, we evaluate the multiple source localization with varying number of sources, source strengths, background radiation, and obstacle placements, to show the robustness of the proposed algorithm.

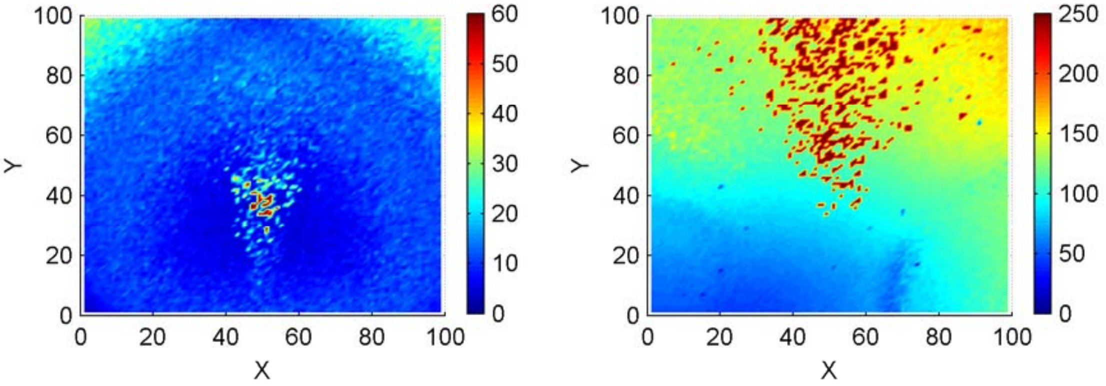
5.1 Single Source Localization with the RoSD Algorithm

We evaluate the performance of the RoSD algorithm presented in Section 3.1 by comparing the localization accuracy and false negatives (miss detections) of the RoSD and the log-space DTOA (LgDTOA) algorithm. We implement the LgDTOA algorithm according to the DTOA algorithm described in [11] and adapt the algorithm to log-space for radiation localization as described in [12]. We evaluate the algorithms in a surveillance area of 100×100 with a network of three sensors. The sensors S_1 and S_2 are located at (0, 0) and (100, 0) respectively. Sensor S_3 is located at either (50, 100), (75, 100), or (100, 50). A 4 μCi radiation source is randomly placed in the surveillance area. In this experiment, we assume there is no background radiation, $B = 0$ CPM, so the sensor readings are due to the radiation source only. Hence, the sensor measurements are simulated using the Poisson random process with λ equal to the expected intensity as described in Section 2.2. The sensor efficiency is set to $E = 0.02$ CPM/DPM.

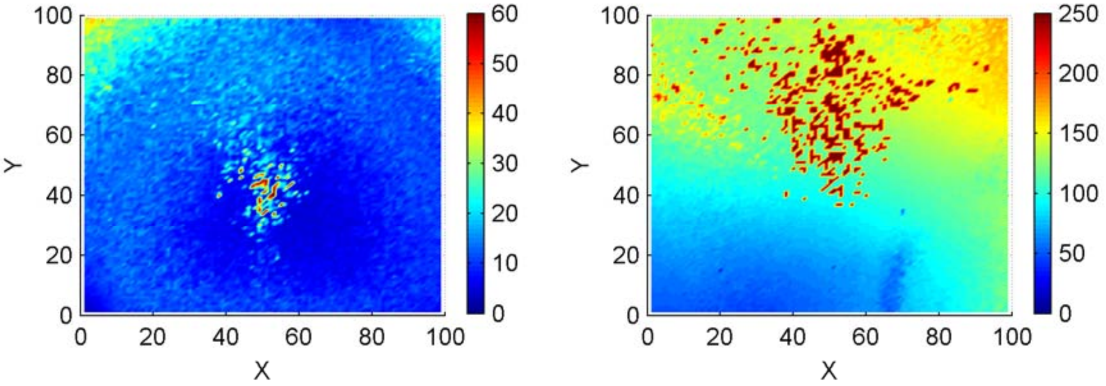
Using the simulated sensor measurements, we localize the radiation source using both the RoSD and LgDTOA algorithms. The LgDTOA algorithm is an iterative algorithm. It terminates when the error in the distance ratio is smaller than ϵ. We set the value of ϵ to 1.0×10^{-10}, in which case the algorithm usually terminates in less than 100 iterations. Nevertheless, we set the maximum number of iterations to 4000, to ensure that any error in the position estimate is not due to premature termination of the algorithm.

Both the RoSD and LgDTOA algorithms may produce up to two estimates for each set of measurements from 3 sensors. When there are two estimates, we assume that the algorithms are able to distinguish the phantom estimate by selecting the estimate that is closer to the actual source position, In other words, the estimate that is farther from the actual source position is the phantom estimate. We measure the performance of the algorithm by the localization error, which is the Euclidean distance between the estimate and the actual source position. In some cases, the algorithms would not produce any estimate due to singularity in the algorithm (for RoSD) or divergence (for LgDTOA). For instance, the RoSD algorithm will not produce any estimate when $I'_i = I'_j$ in Equation (3.6). In these cases, the result is not included in computing the localization error, but is counted as a false negative.

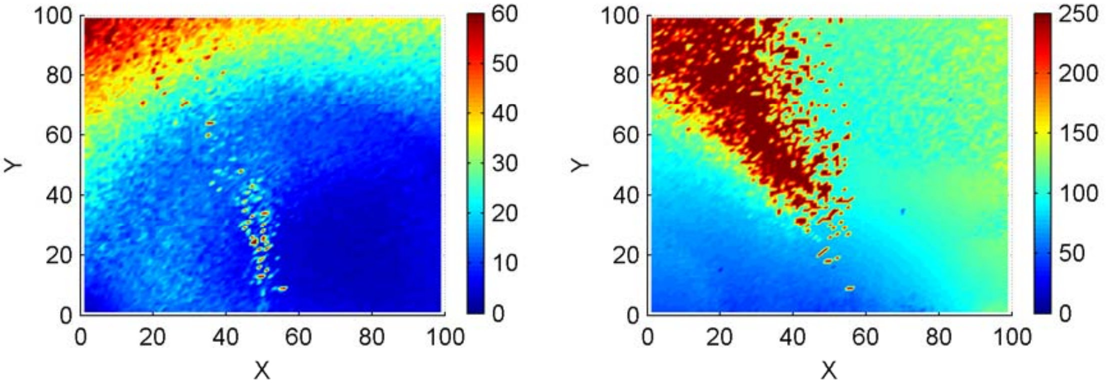
We perform the experiment with 25,000 different random source positions covering the entire surveillance area. For each source position, the experiment is performed with three different locations of S_3. To ensure the significance of the result, we repeat the experiment with 10 sets of sensor measurements for each combination of source and sensor position. Figure 5.1 shows the localization error of the RoSD and LgDTOA algorithms with a 4 μCi radiation source. The color in the figure denotes the average localization error at that particular location.



(a) RoSD, $S_3 = (50, 100)$ (b) LgDTOA, $S_3 = (50, 100)$



(c) RoSD, $S_3 = (75, 100)$ (d) LgDTOA, $S_3 = (75, 100)$



(e) RoSD, $S_3 = (100, 50)$ (f) LgDTOA, $S_3 = (100, 50)$

Figure 5.1. Localization error of the RoSD and LgDTOA algorithms.

The result indicates that the RoSD algorithm achieves low localization error when the radiation source is located close to at least two sensors, relative to the distance of the three sensors. The localization error increases as the overall distance between the source and sensors increases. This is expected because the radiation intensity recorded by the sensors is weaker when the radiation source is at a distance. In general, the algorithm achieves low localization error when the radiation source is located within the triangle formed by the three sensors. The triangle serves as the limit for the distance between the source and sensors. Using this property, a good sensor placement, which minimizes the localization error, should maximize the area of this triangle. This can be accomplished by, for example, placing the sensors in the corners of the surveillance area.

In addition, the radiation intensities measured by S_1 and S_2 are expected to be the same when the radiation source is located equidistance to S_1 and S_2. In practice, the measurements may not be identical due to the random nature of nuclear disintegrations, but the difference between I'_1 and I'_2 will be small. As a result, the RoSD algorithm is very sensitive to error because $L^x_{ij}, L^y_{ij}, L^r_{ij} \to \infty$ when $(I'_1 - I'_2) \to 0$. Because of this, the localization error is large when $x \approx 50$ as shown in Figure 5.1.

For a quantitative comparison, we show the average, median, and standard deviation of the localization over the entire surveillance area in Table 5.1. The result shows that the RoSD algorithm outperforms LgDTOA in both localization accuracy and false negative rate. In addition, the standard deviation of the localization error for the RoSD algorithm is significantly lower than for the LgDTOA algorithm. This implies that the RoSD algorithm consistently produces accurate estimates of the radiation source. The result also shows that the RoSD algorithm has negligible false negatives compared with the LgDTOA algorithm. False negative measures the number of times the algorithm does not produce an estimate of the radiation source out of the total number of simulation runs. Low values for false negatives indicate that the algorithm is robust to noise and errors.

Table 5.1
Localization error statistics of the RoSD and LgDTOA algorithms.

Source Strength	Algorithm	Sensor Location S_3	Average	Median	StDev	False Negative Rate
1 μCi	RoSD	(50, 100)	18.28	15.25	12.97	1.94%
		(75, 100)	19.05	15.78	13.79	1.98%
		(100, 50)	23.90	17.75	19.64	2.16%
	LgDTOA	(50, 100)	110.76	109.80	35.51	30.32%
		(75, 100)	110.68	109.29	34.45	28.93%
		(100, 50)	4497.89	98.02	$> 10^6$	34.40%
4 μCi	RoSD	(50, 100)	11.09	8.72	8.50	0.00%
		(75, 100)	11.47	9.02	8.85	0.00%
		(100, 50)	16.53	11.00	15.29	0.01%
	LgDTOA	(50, 100)	111.81	112.09	35.66	20.19%
		(75, 100)	112.19	112.18	35.32	19.41%
		(100, 50)	110.66	98.49	90.97	32.40%

The result in Table 5.1 also shows that the LgDTOA algorithm does not benefit from higher radiation source strength. In other words, localizing a strong radiation source is as difficult as localizing a weak radiation source with the LgDTOA algorithm. The localization error of the RoSD algorithm, however, is reduced when the source strength increases. In addition, a lower standard deviation in localizing a stronger source indicates that the RoSD algorithm produces a more reliable estimate when localizing a stronger radiation source, which is a desirable property for a radiation source localization algorithm.

Table 5.2
Execution time statistics of the RoSD and LgDTOA algorithms.

	Execution Time (ms)		
Algorithm	Average	Median	StDev
RoSD	0.0043	0.0018	0.0497
LgDTOA	27.0778	28.0159	10.2065

In addition to localization error, we measure the execution time for both the DTOA and the LgDTOA algorithms. Specifically, we measure the wall-clock times of the algorithms on an Intel® Xeon® E7450 CPU at 2.40 GHz. Each CPU has 12 MB shared L3 cache and the machine has 24 GB RAM. The machine runs Linux 2.6.32.7, and have no other programs running while the measurements are being taken. We repeat the measurement 1000 times and report the statistic in Table 5.2. The result shows that the RoSD algorithm runs 6297 times faster than the LgDTOA on average. The speed-up is primarily because the complexity of the RoSD algorithm is $O(1)$. The source estimate is computed directly from Equation (3.5). The LgDTOA algorithm, however, uses a bi-section procedure to iteratively search the parameter space for the solution that matches the sensor measurements down to a specified fractional tolerance ϵ. Depending on the initial guess of the solution and the value of ϵ, the algorithm may take more time to execute.

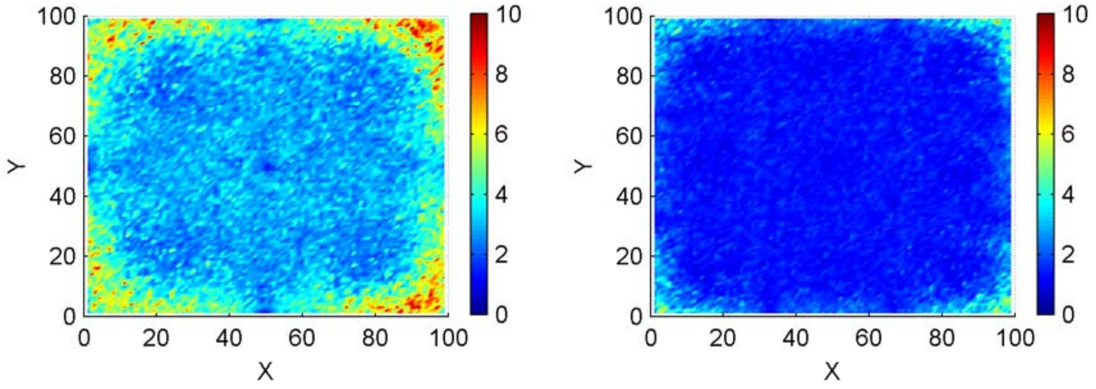
5.2 Single Source Localization with the ITP Algorithm

We now evaluate the performance of the ITP algorithm presented in Section 3.2. First, we evaluate the localization error of the algorithm with regard to the placement of the radiation source. In our experiments, we maintain the size of the surveillance area at 100×100 and place the sensors in a uniform grid in the surveillance area.

The uniform grid placement is used because the uniform grid maximizes the triangles formed by pairs of three sensors, which helps the RoSD algorithm to achieve low localization error as discussed in Section 5.1. We evaluate the algorithm with $N = 3^2, 4^2, \ldots, 6^2$ sensors and a 4 μCi radiation source randomly placed within the surveillance area. The background radiation at all sensor sites is $B = 0$ CPM. The sensor efficiency and tolerance of the ITP algorithm are set to $E = 0.02$ CPM/DPM and $d = 1$ in all subsequent experiments unless indicated otherwise. The value $d = 1$ is selected because the ITP performs best in this specified size of the surveillance area. The evaluation of this parameter is discussed later in this section. We perform the experiment with 25,000 different source positions covering the entire surveillance area. For each source position, we repeat the experiment with 10 sets of sensor measurements to ensure the significance of the result.

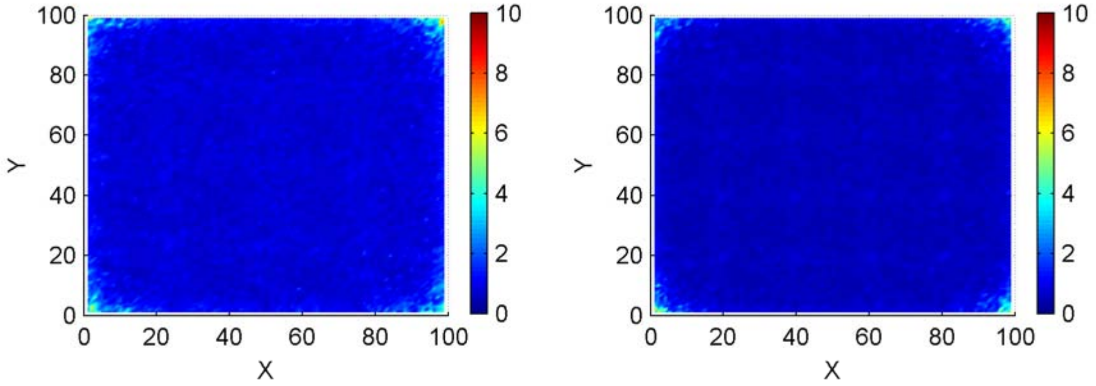
Figure 5.2 shows the average localization error of the ITP algorithm at each location in the surveillance area. The color in the figure denotes the magnitude of the localization error. The result shows that the ITP algorithm achieves a low localization error at the majority of the places in the surveillance area, except at places near the boundary of the surveillance area. As discussed in Section 5.1, the RoSD algorithm is most efficient when the radiation source is located within the triangle formed by the three sensors. When the radiation source is located near the boundary, the radiation source cannot be located within the triangle for the majority of the sensor combinations. As the localization errors for the majority of the candidate estimates increase, the localization error of the fused estimate increases as well. This effect becomes significant when the radiation source is located at the corner of the surveillance area.

Figure 5.3 shows the median localization error with respect to the number of sensors and different source strengths. The result shows that increasing the number of sensors and fusing the data with the ITP algorithm help improve the localization accuracy significantly. The median localization error of a 4 μCi radiation source with 9 sensors using the ITP algorithm is 2.44 units compared to 8.72 units with the RoSD algorithm, an increase of 4×. This improvement comes primarily from the fact that



(a) 9 Sensors

(b) 16 Sensors

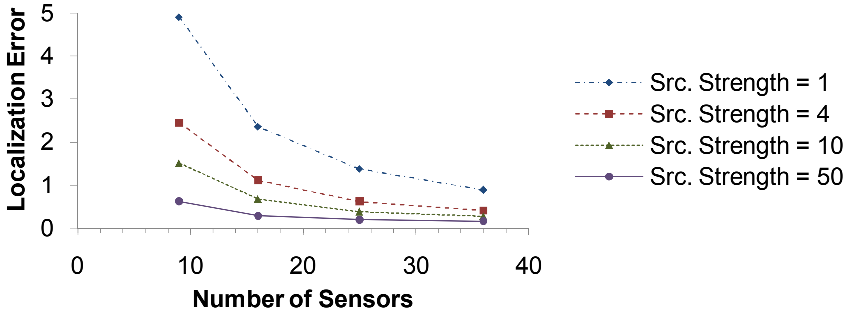


(c) 25 Sensors

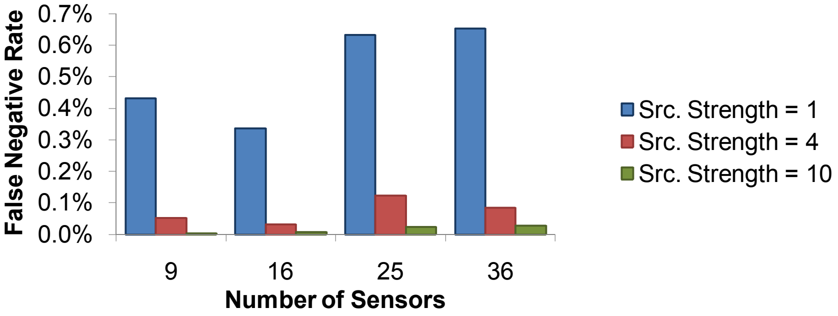
(d) 36 Sensors

Figure 5.2. Localization error of the ITP algorithm with varying number of sensors.

phantom estimates are eliminated by the ITP clustering algorithm and measurement errors are reduced by having more sensors in the surveillance area. In addition, the false negative rate of the ITP algorithm is negligible, less than 0.7%, as shown in Figure 5.3(b). Unlike the RoSD algorithm, in which false negatives occur when the algorithm fails to produce an estimate given the measurements from three sensors, false negatives in the ITP algorithm only occur when no significant cluster is formed given all the candidate estimates. This happens when the similarity threshold d is too small, or the noise/errors are unusually high. As expected, the false negative rate decreases as the source strength increases as shown in the figure because the RoSD



(a) Localization Error

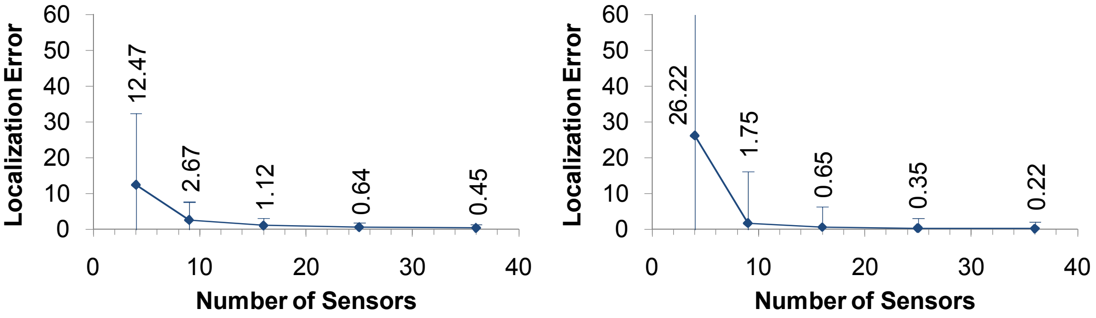


(b) False Negative Rate

Figure 5.3. Performance of the ITP algorithm with different number of sensors.

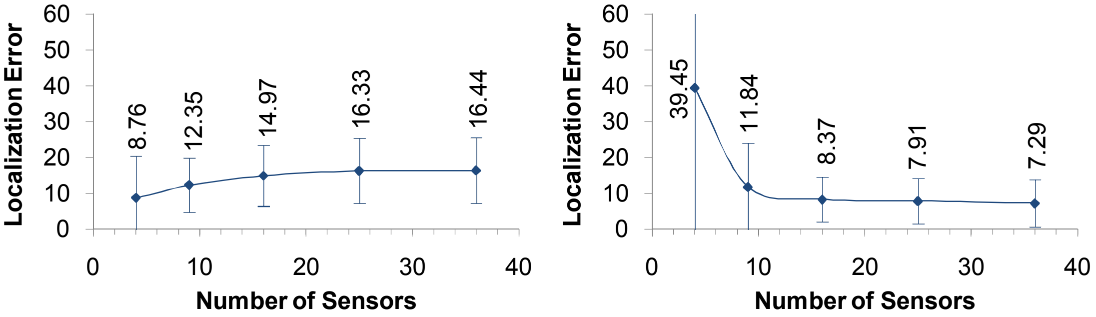
algorithm being used in the ITP algorithm performs better with stronger radiation sources.

To put the performance of the ITP algorithm into perspective, we compare the localization error of the ITP algorithm against other localization algorithms, including the Maximum Likelihood Estimation (MLE) algorithm [14] and the Mean-of-Estimator (MoE) algorithm [13] in a realistic setting, i.e., with background radiation. One problem with this comparison is that the MoE algorithm assumes the localization algorithm produces only one candidate estimate. In our case, the localization algorithm may produce up to two estimates for each subset of 3 sensors. Because selecting between the two candidate estimates for use in MoE has a large impact on the localization error, we evaluate two possible approaches:



(a) ITP

(b) MLE



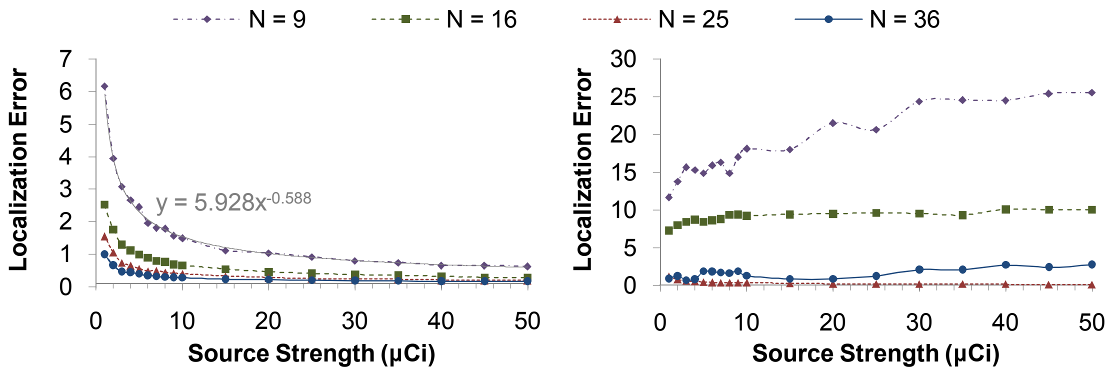
(c) MoE

(d) MoE/R

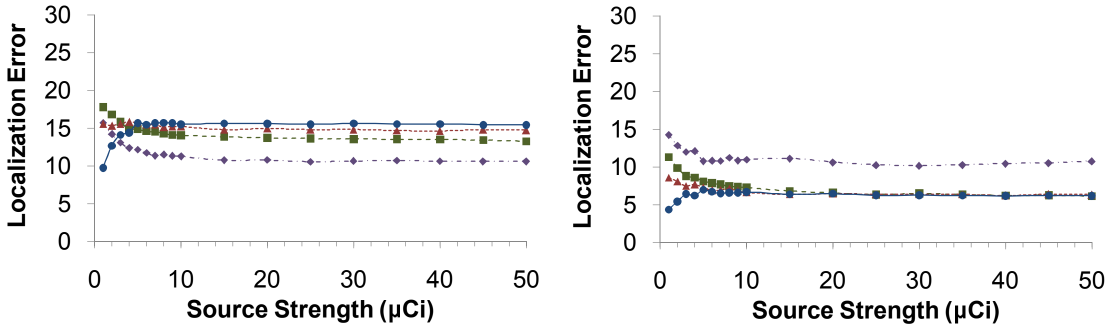
Figure 5.4. Localization error of various algorithms with different number of sensors.

- *Smaller maximum distance to sensors:* Labeled as *MoE* in the reported results (Figure 5.4), this approach selects the estimate that has the smaller maximum distance between the estimate and all the sensors.
- *Random:* Labeled as *MoE/R* in the reported results, this approach randomly selects one of the two candidate estimates for use with MoE.

In the following experiments, a 4 μCi radiation source is randomly placed in the surveillance area and the background radiation is $B = 10$ CPM. The experiment is repeated 1000 times. Figure 5.4 shows the median localization error of all algorithms. The error bar in Figure 5.4 denotes the standard deviation of the localization error. The result shows that ITP and MLE achieve low localization error when $N \geq 9$. However, the standard deviation of the MLE algorithm is 3 times higher, on average,



(a) ITP (b) MLE



(c) MoE (d) MoE/R

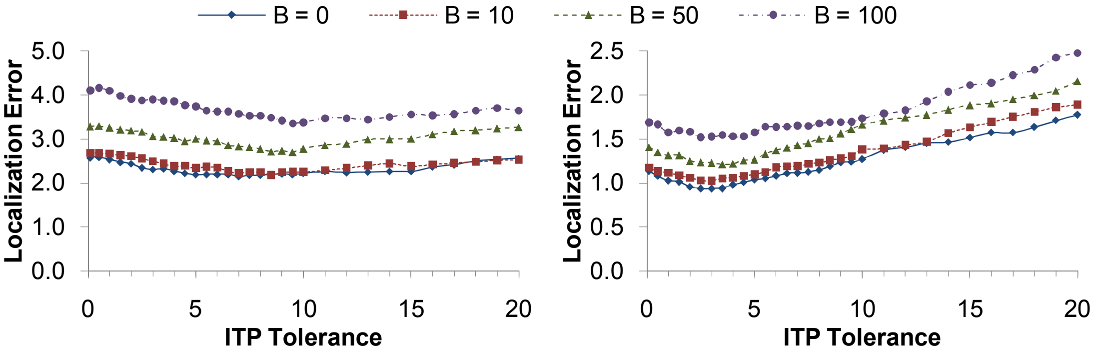
Figure 5.5. Localization error with different source strengths.

than the ITP algorithm, especially when there are fewer sensors in the surveillance area. This is because in some cases, the initial values chosen by the MLE algorithm are not close to the actual source position. As a result, the algorithm exceeds the maximum number of iterations allowed in refining the source estimate. In other words, the MLE algorithm may not consistently produce accurate estimates of the radiation source although the median of the localization error is comparable to the ITP algorithm. The MoE and MoE/R algorithms produce much larger localization errors compared with the ITP and MLE algorithms because the MoE and MoE/R algorithms cannot accurately disambiguate the phantom estimates.

The strength of the radiation source also influences the accuracy of the ITP algorithm. Figure 5.5 shows the median localization error of all algorithms. In general, a strong radiation source is easier to detect and localize than a weak one because the radiation from strong sources travels a longer distance. When a weak source is present in the surveillance area, fewer sensors will be able to detect the source. This leads to a smaller number of candidate estimates produced by the RoSD algorithm. As discussed in Section 3.2.3, the performance of the ITP algorithm is degraded when there are few candidate estimates. When more groups of sensors are able to produce accurate candidate estimates for the radiation source, the increased number of candidate estimates helps the ITP algorithm to calculate a more precise estimate of the source position.

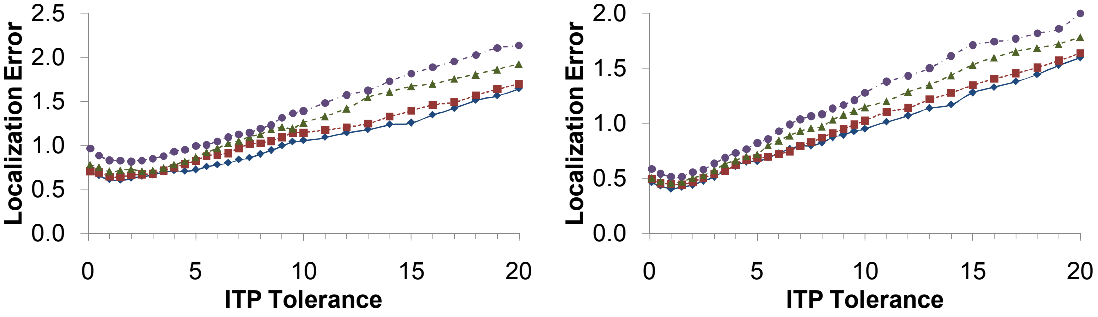
The median localization error of the MLE algorithm is comparable to the ITP algorithm in a network of $N \geq 25$ sensors. However, both the localization error and execution time of MLE have large variances, which indicates that the algorithm may not consistently achieve a low localization error or execution time. In a network of less than 25 sensors, the localization error of the MLE algorithm increases significantly. This increase is caused by greater distance between neighboring sensors when there are fewer sensors. As the distance between sensors increases, the intensity from the radiation source to sensors decreases, thus the noise component in sensor measurements increases relatively. In addition, a smaller number of sensors in the network leads to a coarse likelihood function being used in the MLE algorithm. As a result, the MLE algorithm has difficulty producing a consistent result and may often exceed the maximum number of iterations allowed.

The localization error of the ITP algorithm decreases as the source strength increases. This is a desirable property because strong radiation sources are dangerous, and thus there is a need to localize such a source quickly and accurately. On the other hand, given a stronger source, other algorithms do not produce a more accurate estimate as shown in Figure 5.5. The localization error does not decrease because the measured radiation intensity follows the Poisson process, where the variance of the intensity measurements is equal to the mean. This indicates that a stronger source will exhibit larger variance in the measurements. For the MLE algorithm, which seeks



(a) 9 Sensors

(b) 16 Sensors



(c) 25 Sensors

(d) 36 Sensors

Figure 5.6. Localization error with varying ITP tolerance.

the best estimate given all the sensor measurements, the increase in the variance of sensor measurements prevents the algorithm from producing a more accurate estimate. For the MoE and MoE/R algorithms, which average all candidate estimates produced, a larger variance undoubtedly leads to a larger error in the estimate produced. The localization error of the ITP algorithm, on the other hand, decreases in the order of $O\left(\frac{1}{\sqrt{A^{\text{str}}}}\right)$ as indicated by the best-fit line and equation in Figure 5.5(a). This is primarily because the candidate estimates that have a large error are pruned in addition to the phantom estimates. As a result, only the most accurate candidate estimates are used in computing the fused estimate.

The parameter d in the ITP algorithm controls the maximum size of the region that should not contain phantom estimates. A small value of d is sufficient with error-free measurements, e.g. $B = 0$ CPM. When considering noise and errors such

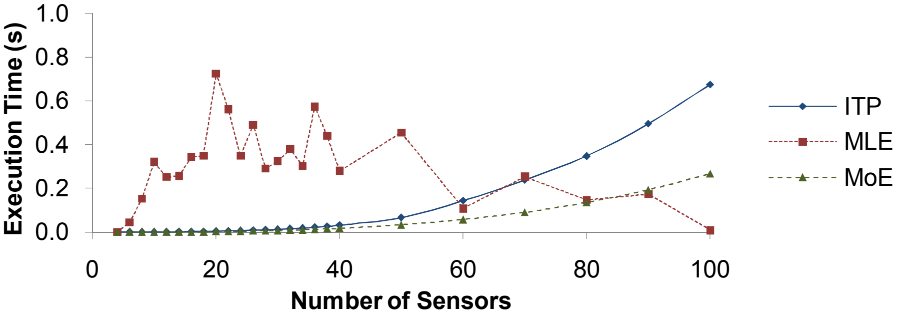


Figure 5.7. Execution time of the ITP, MLE and MoE algorithms.

as background radiation, a larger value of d is required to tolerate the noise/errors such that the ITP algorithm includes most, if not all, non-phantom estimates. Figure 5.6 shows the median localization error of the ITP algorithm with different values of d ranging from 0.1 to 20. As shown in the figure, setting the value of d too large degrades the localization accuracy because the algorithm is more likely to include phantom estimates within the region. In fact, setting $d = \infty$ causes the algorithm to behave similarly to the MoE algorithm. However, setting a small value of d also slightly degrades the localization accuracy as some of the non-phantom estimates are being excluded. Implicitly, the value of d gives a boundary to the region where the radiation source is most likely to be contained. The result indicates that setting the value of d to $\approx 6\%$ of the inter-sensor distance yields minimum localization error.

In terms of execution time, we repeat the experiment with 4 to 100 sensors and a 100 μCi radiation source. Both the radiation source and the sensors are randomly placed in the surveillance area. We measure the average execution time of the ITP, MLE, and MoE algorithms on an Intel® Xeon® E7450 CPU at 2.40 GHz. Each CPU has 12 MB shared L3 cache and the machine has 24 GB RAM. The machine runs Linux 2.6.32.7, and have no other programs running while the measurements are being taken. We repeat the measurement 1000 times and report the average execution time in Figure 5.7.

The result indicates that the ITP scales in $O\left(N^3\right)$ to the number of sensors N as expected. In fact, the execution time of the ITP algorithm fits to a 3rd degree polynomial with R^2 value of 0.9995 (where $R^2 = 1$ indicates a perfect fit) and with a root mean square error (RMSE) of 3.84 ms (where RMSE = 0 indicates a perfect fit). Nonetheless, the algorithm takes only 0.68 seconds to run, on average, even when the experiment is scaled to 100 sensors. Moreover, the standard deviation of the execution time for the ITP algorithm is between 0.07 ms and 74.4 ms, which indicates that the execution time is very predictable regardless of the location of the radiation source. The MoE algorithm is significantly simpler in fusing the candidate estimates and therefore has a shorter execution time compared with the ITP algorithm. However, the time required to enumerate the candidate estimates from all groups of three sensors is similar to the ITP algorithm. In Figure 5.7, the difference in execution time between the MoE and ITP algorithms represents the time in which the ITP algorithm spent on pruning the parameter space and computing the fused estimate.

The execution time of the MLE algorithm, on the other hand, shows that the algorithm executes faster with more sensors. This is due to the fact that the size of the surveillance area is kept constant. Therefore, more sensors in the surveillance area are closer to the radiation source thus measuring a stronger radiation intensity from the source. The measurements that clearly indicate the presence of the radiation source help the MLE algorithm to converge quickly, and thus the MLE algorithm has a faster execution time. Nonetheless, the standard deviation of the execution time of the MLE algorithm varies widely between 0.36 ms and 6369.48 ms, with the average of the standard deviation being 3643.88 ms. This indicates that the algorithm may take several seconds to process a single measurement. The variability of the execution time may pose a problem when using the algorithm in a near real-time system. The longer execution time and the larger variance are due to the cases where the initial values are not close to the actual source position or with some particular combination of sensor measurements. In these cases, the algorithm can take many iterations to converge, and frequently exceed the maximum number of iterations allowed in our experiments.

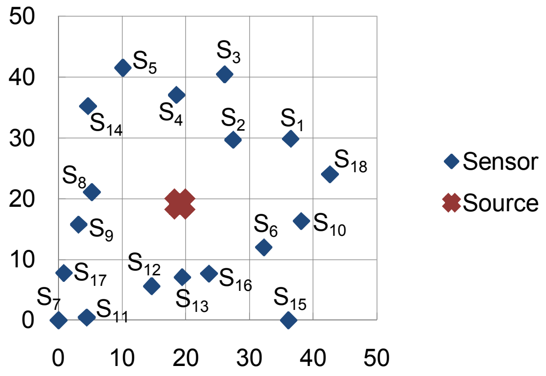


Figure 5.8. Placements of the radiation source and sensors.

5.3 Testbed Evaluation of Single Source Localization Algorithms

In this section, we evaluate the performance of the ITP algorithm in a testbed with an extremely low-intensity (0.911 μCi) point radiation source. We also set up an identical simulation environment in order to ensure our simulation model is realistic, as well as to validate the experiment result. First, we describe the setup of the simulation experiments in Section 5.3.1. Second, in Section 5.3.2, we validate the system models described in Chapter 2 to ensure that the simulation results are realistic. After that, in Section 5.3.3, we characterize and compare the performance of the RoSD algorithm with the LgDTOA algorithm. Lastly, in Section 5.3.4, we evaluate the performance of the ITP algorithm, and compare its performance with that of the MLE, MoE, and MoE/R algorithms in the case of more than 3 sensors.

5.3.1 Experiment Setup

In this experiment, there are 18 sensors in a 50 cm × 50 cm surveillance region. Their positions are given in Table 5.3. Not all the sensors are activated in all the experiments. A CS-137 point radiation source, which emits gamma rays, of intensity $A^{\text{str}} = 0.911$ μCi is placed at $(x_s, y_s) = (19.09, 19.09)$ in the testbed. All the dis-

Table 5.3
Sensor coordinates in testbed evaluation.

Sensor	X (cm)	Y (cm)	Sensor	X (cm)	Y (cm)
S_1	36.52	29.89	S_{10}	38.19	16.33
S_2	27.45	29.70	S_{11}	4.43	0.47
S_3	26.13	40.46	S_{12}	14.65	5.60
S_4	18.53	37.08	S_{13}	19.46	7.10
S_5	10.11	41.57	S_{14}	4.64	35.28
S_6	32.31	12.00	S_{15}	36.15	0.00
S_7	0.00	0.00	S_{16}	23.65	7.67
S_8	5.24	21.10	S_{17}	0.81	7.77
S_9	3.13	15.77	S_{18}	42.68	24.01

tance measurements are in centimeters (cm). The placement scenario is illustrated in Figure 5.8.

In the testbed experiments, the distance between the radiation source and each sensor is calculated according to Table 5.3. The radiation source is then placed at the calculated distance from the sensors. Traces for S_1 to S_{18} are collected by polling the corresponding sensors every four seconds. A total of 3738 measurements are collected for each sensor. The traces are not synchronized because they may not have been taken at the same time. However, this does not affect the experimental results because the underlying nuclear disintegrations follow a memoryless Poisson process. The traces collected are then used to drive the experiments.

The radiation source being used in the testbed experiments needs to be extremely weak for safety reasons. The intensity is so low that the sensors observe only the background radiation when they are placed more than four feet away from the source. This has restricted us to using a small size region in the experiments.

Table 5.4
Measured efficiency of RFTrax sensors.

Sensor Serial #	Average Source Intensity (CPM)	Average Background Intensity (CPM)	Efficiency (CPM/DPM)
000877	4202.6938	10.5217	0.2078%
000809	6275.1239	13.6521	0.3096%
000841	4882.2561	18.4220	0.2405%

We replicate the testbed setup in the simulation environment both to validate the testbed results and to illustrate the accuracy and practicality of the system models presented in Chapter 2. In the simulation/testbed experiments, we evaluate the localization algorithms with three variations of sensors:

1. RFTrax sensor (testbed) — The RFTrax RAD-CZT radiation sensor [37] is manufactured by RFTrax, Inc.. The sensor uses the Cadmium Zinc Telluride (CZT) detector technology and has a sensitivity of 0.1 mREM/hour. This is the sensor deployed in our testbed. The RFTrax sensor has an embedded micro-controller for data processing and communication. Unlike the majority of radiation sensors on the market, which keep track of the radiation count by minutes or report the individual radiation count, the RFTrax sensor keeps track of the radiation count for the past one second only. Hence, the CPM count reported by the sensor is the number of disintegrations detected in the past one second multiplied by 60. The variance of the measurements is therefore 60 times higher than normal. This problem is particularly acute for the extremely weak 0.911 μCi CS-137 point source being used in the experiments. To mitigate this problem, we apply a moving average with a window of 60 measurements to the sensor data. This is similar to taking the average of 60 seconds worth of data.

2. Ideal sensor (simulation) — We simulate an ideal sensor, defined as one that is capable of detecting all gamma rays radiated. The sensitivity of the sensor is $E = 1.0$ CPM/DPM. There is no background radiation.

3. Model of real sensor (simulation) — We create a simulation model of the RFTrax RAD-CZT radiation sensor using the specifications from the manufacturer. We discussed the architecture of the sensor with a knowledgeable Engineering Manager at RFTrax, Inc. and obtained the needed information, described above, to model the firmware processing in the RFTrax sensor accurately. The sensor efficiency being used in the simulation reflects the actual values obtained via the calibration procedure described in Section 2.4. These values are given in Table 5.4.

5.3.2 Validation of the System Models

We validate the system models presented in Chapter 2 by collecting traces of radiation counts obtained with the 0.911 μCi CS-137 radiation source placed at specified distances from the sensors. The measured radiation intensities and the corresponding distances between the sensor and the radiation source are recorded. The experiment is repeated 6 times with the source located at a different distance from the sensors for each repetition. A total of 18 traces are collected and each trace contains at least 3738 measurements. The average intensities computed from the traces are compared with the values computed from the sensing model. The result in Table 5.5 shows that the model predicted values and the actual measurements differ by less than 5 CPM for 89% of the time.

5.3.3 Evaluation of the RoSD Algorithm

In this section, we evaluate the performance of the RoSD algorithm in the testbed. We activate sensors S_7, S_{15}, and S_4 and measure the localization error of the RoSD

Table 5.5
Comparison of the model and actual measurements from RFTrax sensors.

Sensor Serial #	Distance (cm)	Average Source Intensity (CPM)	Model (CPM)	Difference (CPM)
000877	20.5	19.7025	20.4735	0.7710
	18.0	21.0433	23.4207	2.3774
	27.0	15.3083	16.2644	0.9561
	19.3	19.5326	21.7461	2.2135
	12.0	35.0893	39.4333	4.3440
	12.3	34.5478	38.0493	3.5015
000809	13.5	37.0099	47.8211	10.8113
	24.2	21.1236	24.3256	3.2020
	14.0	32.3821	45.4362	13.0541
	23.7	19.9144	24.7799	4.8655
	21.7	22.1469	26.9211	4.7741
	21.5	23.1495	27.1685	4.0190
000841	22.5	25.3450	28.0107	2.6657
	15.0	38.3307	39.9434	1.6128
	16.3	32.8900	36.6598	3.7699
	14.2	40.3292	42.4244	2.0952
	25.6	24.3019	25.8324	1.5304
	24.1	25.8622	26.7819	0.9197

algorithm only, i.e., without running the ITP fusion algorithm. Because the fusion algorithm is not used, phantom estimates produced by the RoSD algorithm are eliminated using the smallest maximum distance approach, i.e., we select the estimates that have the higher signal-to-noise ratio (SNR). To ensure that the simplified algo-

Table 5.6
Localization error statistics of the RoSD and LgDTOA algorithms in testbed.

Sensor Type	Algorithm	Average	Median	StDev	False Negative
Testbed	RoSD	43.31	42.83	6.45	25.11%
	LgDTOA	44.68	43.12	10.15	43.68%
Simulated	RoSD	43.29	42.27	5.32	11.16%
	LgDTOA	44.05	42.05	9.40	23.32%
Ideal	RoSD	1.56	1.32	1.08	0.00%
	LgDTOA	24.57	24.60	0.65	0.27%

rithm works, the three sensors selected are located such that the radiation source is contained within the triangle formed by the sensors. The experiment is repeated 1000 times with different measurements.

Table 5.6 shows the localization error statistics and the false negative rates, i.e., the percentage of time where the localization algorithm failed to produce an estimate, of the RoSD and LgDTOA algorithms. The table shows results for the testbed experiment (labeled *Testbed*), for the simulation using the simulated model of the RFTrax sensors (labeled *Simulated*), and for the ideal sensors (labeled *Ideal*). The results of the experiment, both in the testbed and simulated RFTrax sensors, show that the average and median localization error of both algorithms are similar. However, the standard deviation of the LgDTOA algorithm is 57% larger. This suggests that the RoSD algorithm has a more consistent result compared with the LgDTOA algorithm. In addition, the false negative rate of the RoSD algorithm is 74% lower than the LgDTOA algorithm. The lower false negative rate of the RoSD algorithm suggests that the RoSD algorithm is more robust to noise and errors. The cumulative probability function (CDF) plot of the localization error in Figure 5.9 again confirms that both the RoSD and LgDTOA algorithms have a similar localization error 90% of the

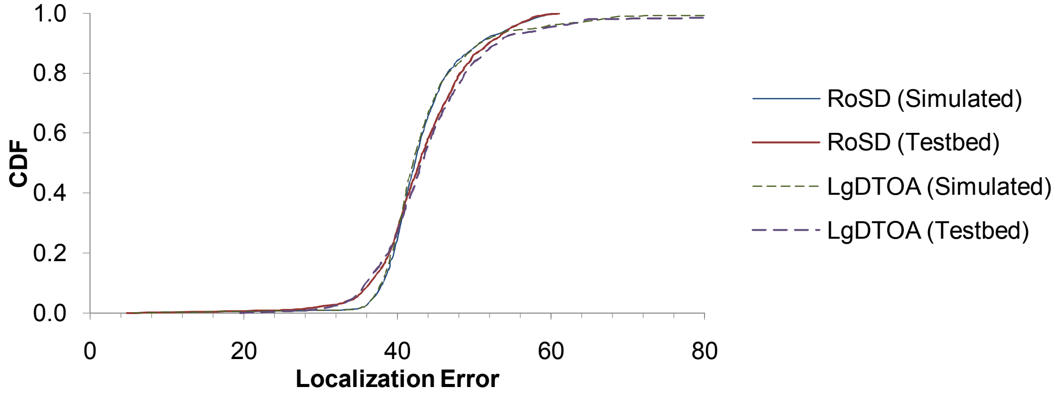


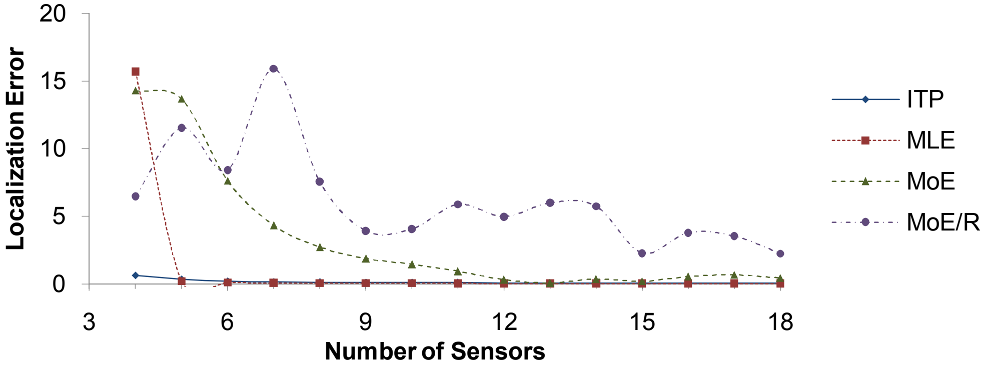
Figure 5.9. Localization error of the RoSD and LgDTOA algorithms in testbed.

time. The LgDTOA algorithm produces estimates that are outside the surveillance area 10% of the time.

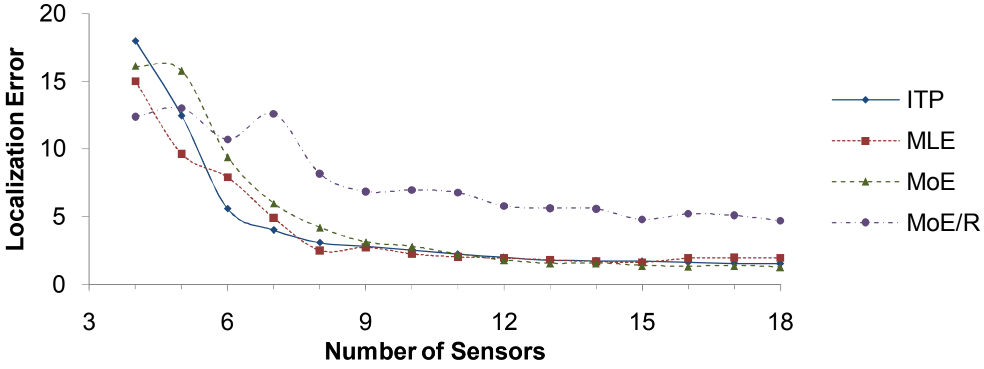
In addition, Table 5.6 also shows experiment results for ideal sensors. The result shows that the RoSD algorithm benefits a lot by having higher efficiency sensors. As indicated in the table, the median localization error is reduced 32 times from 42.83 to 1.32 units. The false negative rate of the RoSD algorithm is reduced to zero by using ideal sensors. On the other hand, the LgDTOA algorithm benefits less from higher efficiency sensors.

5.3.4 Evaluation of the ITP Algorithm

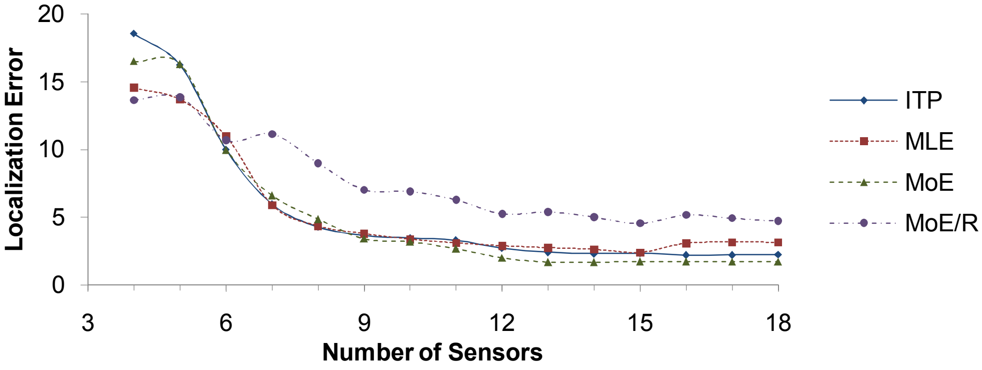
In this section, we evaluate the performance of the ITP algorithm in the testbed. We repeat the experiments in the previous section, but with more than three sensors and the ITP algorithm. In this experiment, we compare the localization accuracy of the ITP algorithm with other algorithms such as MLE, MoE and MoE/R. We systematically activate sensors S_1 to S_N for $N = 4, \ldots, 18$, and the sensor measurements are fed to all localization algorithms. The distances between the actual position of the source and the estimates produced by all algorithms are presented. In the experiment, the tolerance of the ITP algorithm is set to $d = 1$. Detailed evaluation of the ITP tolerance parameter in the testbed are presented later in this section.



(a) Ideal



(b) Simulated



(c) Testbed

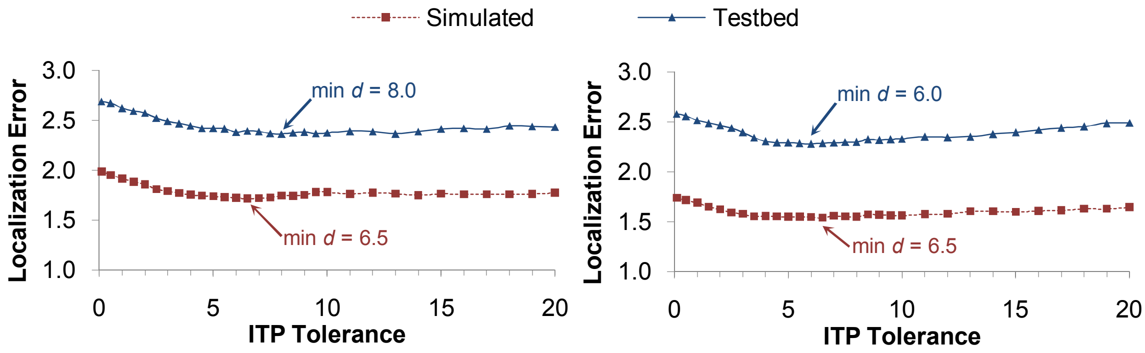
Figure 5.10. Localization error of various algorithms with varying number of sensors.

Figure 5.10 shows the localization error of the ITP, MLE, MoE, and MoE/R algorithms in the testbed experiment. The results show that the ITP and MLE algorithms have similar localization accuracy. In addition, the results also show that the MoE algorithm has a larger localization error when using ideal sensors, but the localization error is similar to ITP and MLE when using real sensors (simulated/testbed).

There are two major factors to consider when determining the errors of the fused estimates in the ITP algorithm. The first factor is the error produced by the localization algorithm. The second factor is the distance between the two candidate estimates. The former is determined by the accuracy of the localization algorithm and the magnitude of noise that is included in the computation. The latter is determined by the sensor measurements and the sensor placements. For example, although the MoE algorithm has no built-in design to disambiguate each pair of candidate estimates, the omission will not have a large impact on performance, i.e., selecting one over the other does not matter much, if the distance between the two candidate estimates is small ($\theta \approx 0^\circ$ in Equation (3.6)). The ITP algorithm, in contrast, has a built-in clustering step to disambiguate the candidate estimates. This is very useful in general, but appears less so if the phantom estimates are close to the actual position of the source.

With ideal sensors, the RoSD algorithm produces virtually zero localization errors due to high SNR measurements. Hence, the distance between the candidate estimates is huge compared with the (almost zero) localization error. This leads to a very low localization error in the ITP algorithm because the algorithm can accurately disambiguate the phantom estimates. However, the MoE and MoE/R algorithms produce large localization errors because the averaging includes phantom estimates that are farther away from the actual source position. This is particularly acute when there are fewer sensors in the surveillance area.

With real sensors (testbed/simulated), the localization error is much larger. Errors due to the probabilistic measurements and background radiation cause the estimate to deviate by as much as the distance between two sensors in the RoSD algorithm. At the same time, the distance between the candidate estimates is similar to the distance between the source and each estimate. In this situation, computing the average of all



(a) 15 Sensors (b) 18 Sensors

Figure 5.11. Localization error with varying ITP tolerance in testbed.

the candidate estimates, as in the MoE algorithm, may perform as good as the more expensive ITP algorithm.

In order to choose the best ITP tolerance value for the testbed, we evaluate the localization error with respect to various values of d. Figure 5.11 shows that the value of d that produces the minimum localization error is around $d = 6$. Compared with the simulation result in the previous section, where $d = 1$, the testbed experiment requires a larger value of d because of the lower sensor efficiency and a higher noise component in the measurements. Therefore, the non-phantom estimates are less concentrated. By setting d to a larger value, more non-phantom estimates are included in computing the fused estimate. Note that the value of d is not critical in the localization accuracy of the algorithm. As shown in Figure 5.11, the localization error increases very slowly when d increases or decreases.

5.3.5 Emulation of Larger Deployments and Stronger Sources

Our testbed is implemented on a 50 × 50 cm workbench using a point radiation source of 0.911 μCi. Larger monitoring areas and stronger sources can be emulated using the testbed as follows. Let r_{emulated} and r_{testbed} be the emulated distance and actual distance on the testbed, respectively, such that $r_{\text{emulated}} = s_f \cdot r_{\text{testbed}}$. Sensor measurements of a radiation source of strength A_{testbed} on the testbed correspond to the measurements of a source of strength $A_{\text{emulated}} = s_f^2 A_{\text{testbed}}$ in the emulated configuration.

For example, r_{emulated} is in meters for a large courtyard and r_{testbed} is in centimeters on the testbed. Then we have $s_f = 1 \text{ m}/1 \text{ cm} = 100 \text{ cm}/1 \text{ cm} = 100$. Thus, we can emulate sources with $A_{\text{emulated}} = 10000 \times A_{\text{testbed}}$ in the courtyard. The emulated source can have a strength of 9110 μCi, which is much higher than the safe level. Thus, much stronger sources can be emulated in our testbed using only a weak source because the distances between the source and the sensors are also scaled. Hence, we are able to retain the complexity of the localization problem in the testbed without using actual strong sources in the experiments. In particular, this emulation method can be used to map public open areas where radiation sources cannot be easily deployed. However, sensors can be deployed in such areas to obtain background radiation measurements, which can then be used as background measurements in the testbed. While this approach is limited when background measurements on the testbed are not the same as the emulated area, it is more accurate than a simulation-only approach.

5.4 Multiple Source Localization

We conduct experiments to evaluate the performance of our multiple source localization algorithm. We measure the performance of the algorithm by (i) localization error, (ii) number of false positives, and (iii) number of false negatives. In measuring the localization accuracy, the Euclidean distance between the actual source position and the closest estimate is used. However, each estimate must uniquely estimates a single source. In other words, an estimate cannot be counted twice. If no estimate is within 40 units from the actual source position, that source is considered not detected (i.e., a false negative). The estimates that cannot be traced to any actual source are considered false positives.

We simulate a 100×100 surveillance area with $N = 6 \times 6$ sensors placed in a uniform grid covering the entire surveillance area. All sensors receive a background radiation of 0, 5, 10, or 50 CPM. These values are chosen to match typical environmental background radiation, which is about 5–20 CPM as measured in the testbed in Section 5.3. Several radiation sources of strength 4–1000 μCi are randomly placed in the surveillance area. These source strengths correspond to typical dirty bombs that could be used in an actual terrorist attack [22]. To put these source strengths further in perspective, the radiation source in a radiotherapy machine is more than 1000 Ci strong [38]. Each simulation is repeated 10 times and the median results are reported. In all the simulations, the standard deviation of resampling noise σ_N is set to 3.0, and the fusion range is $d_i = 28$ for $1 \leq i \leq N$. We use the same fusion range for all the sensors because the sensors are arranged in a uniform grid.

In our simulations, sensor measurements arrive sequentially and in order. Here, we introduce the notion of time step T. In each time step, each sensor in the surveillance area submits one measurement update. Therefore, each time step is equivalent to N iterations. All the reported simulations have 30 time steps.

5.4.1 Multiple Source Results

We report an experiment with two sources of different strengths located at (47, 71) and (81, 42), and background radiation is 5 CPM. The performance of the proposed algorithm is shown in Figure 5.12. In the figure, the large localization error in the first few time steps is due to the fact that we initialized the particles uniformly and randomly at the beginning due to the lack of additional knowledge. As a result, the algorithm does not have enough information to accurately localize the sources. As shown in Figure 5.13, randomly placed particles start to cluster at the radiation sources as early as $T = 1$. As more sensor measurements become available, the localization error quickly reduces to a small value.

Because the particles are randomly located at $T = 0$, the algorithm may produce many false positives because many places are thought likely to contain the source given the very limited number of sensor measurements. As time proceeds, the number of false positives is reduced because the sensor measurements do not give high weights to those particles. In addition, there is an increase in the number of false positives for stronger sources. The primary reason is that the radiation from stronger sources can reach longer distances. This increases the ambiguities in the sensor measurements because the measurements could be due either to a stronger source located farther away or to a weaker source located closer to the sensor. As a result, the algorithm is unable to reduce the weights of particles far away from the actual source, thus producing false positives. The problem can be mitigated by reducing the fusion range. However, without knowing the source strengths in advance, reducing the fusion range can increase the false negatives because weaker sources are missed. The false negatives, on the other hand, are close to zero, except when the source strengths are very weak (e.g., 4 μCi) as shown in Figure 5.12(b). This is because the radiation from weak sources appears to be similar to background radiation. This makes it difficult for the algorithm to distinguish between radiation from a source and radiation from the background.

We evaluate the algorithm in a scenario with three sources. The three sources are located at (87, 89), (37, 14), (55, 51), and the background radiation is 5 CPM. In the three-source simulation scenario, the results are generally similar to the two-source case, as shown in Figure 5.14. However, the algorithm requires more time steps to produce accurate results. Especially with the 4 μCi source, the algorithm takes 9 time steps before producing accurate location estimates. In terms of false positives and false negatives, similar trends are observed for the two-source scenario.

We also evaluate our algorithm with four different levels of background radiation: 0, 5, 10, and 50 CPM in scenarios with two sources. The radiation sources are 10 μCi in strength and are located at (47, 71) and (81, 42). The results in Figure 5.15 show that our algorithm can tolerate above-average background radiation as compared to typical environments. Higher background radiation only affects the first few time steps of the algorithm. The results also show that the higher background radiation has no impact on the false positives and false negatives.

5.4.2 Results with Obstacles

To evaluate how obstacles affect the performance of our algorithm, we repeat the two-source simulations above, but added a U-shape obstacle in the middle of the surveillance area as shown in Figure 5.16. The thickness of the obstacle is two units, and the attenuation coefficient is $\mu = 0.0693$. This value of μ corresponds to halving the radiation intensity for every 10 units of thickness. It is selected such that the obstacle does not completely block the radiation, but allows some radiation to penetrate through it.

Figure 5.17 shows the localization error, false positives, and false negatives for various source strengths with an obstacle in the surveillance area as shown in Figure 5.16. The radiation sources are located at (47 ,71) and (81, 42), and the background radiation is 5 CPM. For comparison, we plot, in Figure 5.18, the localization error of the scenario without the obstacle normalized to that of the same scenario with the obstacle. Thus, in the figure, values greater than 1.0 imply that the scenario with the obstacle achieves better accuracy. The figure shows that the obstacle improves the accuracy of our algorithm by 24.5% for source 1, but degrade it by 2.4% for source 2. The shielding by the obstacle, although partial, reduces the interference between the two sources, causing the sensors to provide more accurate measurements for the individual sources. However, the false positives and false negatives, with or without the obstacle, are not significantly different, as shown in Figure 5.17. We conclude that obstacles in the environment may have positive effects, but do not have significant negative effects, on the localization. We will revisit the effects of obstacles in a large-scale network in Section 5.4.3.

Existing works that localize multiple radiation sources typically require modeling all the sources and their interactions. In many cases, ambiguities cannot be resolved because a strong radiation source can give a signature similar to multiple weak sources. Obstacles in the surveillance area pose additional challenges for existing algorithms. On the other hand, our algorithm can exploit the shielding effect of obstacles to improve the localization accuracy.

5.4.3 Large Network Results

In this section, we illustrate the scalability of our algorithm by simulating a scenario with 196 sensors placed in a grid and nine radiation sources of non-uniform strengths (between 10–100 μCi) randomly placed in the surveillance area. In addition, three obstacles, each with a different thickness, each without a uniform thickness, are placed in the surveillance area. The layout of this scenario is given in Figure 5.19.

In this simulation, we increase the number of particles to 15000, proportional to the area increase. We evaluate the performance of our algorithm with and without the obstacles. Figure 5.20 shows the localization error, false positives, and false negatives of source 1–4 in the nine-source scenarios, with and without the obstacles. The placements of the obstacles are shown in Figure 5.19. The data for source 5–9 are similar, thus are not shown. The results show that our algorithm achieves similar localization accuracy in both the large and small networks, regardless of the number of sources. Similar to the results in Section 5.4.1, the first few time steps produce a large number of false positives and false negatives. In this case, the total number of false positives and false negatives has increased by more than ten times due to the increased number of sources. However, they quickly reduce to around 0.5 on average after several time steps.

To illustrate the effects of the obstacles, we plot and compare the normalized localization errors for the experiments with and without the obstacles in Figure 5.21. The results show that some sources benefit from the obstacles more than the others.

This is largely due to the placements of the obstacles relative to the sources and sensors. Figure 5.22 shows the average improvements in the localization error in time steps 5–29 for the scenario with nine radiation sources. The first five time steps are omitted in the computation because they are not representative. The figure shows that the five sources, located at (21, 170), (85, 176), (126, 208), (220, 68), (193, 193), benefit from the obstacles. All these sources have at least one obstacle located near them. Three sources, located at (63, 58), (115, 29), (202, 118), have similar localization error with or without the obstacles. These sources do not have obstacles near them except for the source at (202, 118). Lastly, only one source, located at (156, 91), is affected negatively by the obstacle. The localization accuracy of this source drops by as much as 25%.

5.4.4 Running Time

In this section, we demonstrate that our algorithm can take advantage of a multi-core machine. This enables the algorithm to run quickly even for very large networks. In this experiment, we measure the wall-clock time of the algorithm for each iteration. We run the experiment on two machines. The first machine has a single socket with an Intel® Core$^{\text{TM}}$2 Quad CPU at 2.40 GHz. The CPU has 4 MB shared L2 cache and 2 GB RAM. The second machine has four sockets with 6-cores Intel® Xeon® E7450 CPU at 2.40 GHz. Each CPU has 12 MB shared L3 cache and the machine has 24 GB RAM. Both machines run Linux 2.6.32.7, and have no other programs running while the measurements are being taken. The average execution time of the algorithm is listed in Table 5.7.

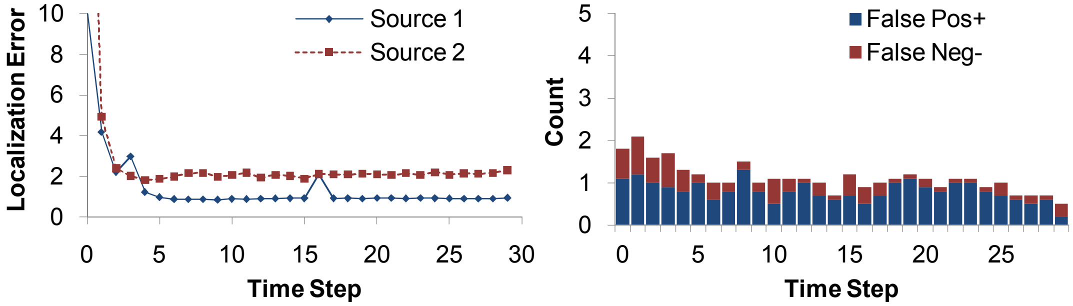
The result shows that our algorithm is highly scalable, with an average speed up of 5× in moving from four cores to 24 cores. The majority of the concurrency is achieved in computing the source parameter estimates using the mean-shift technique. Manipulating the particles in the predict and update steps does not consume

Table 5.7
Average execution time of the multiple source localization algorithm.

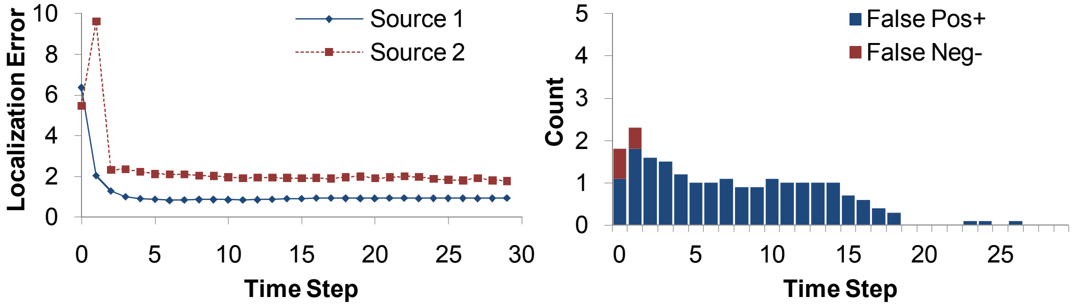
# Particles	Average Execution Time (seconds/iteration)			
	4 cores machine		24 cores machine	
	$N = 36$	$N = 196$	$N = 36$	$N = 196$
2000	0.23	0.22	0.04	0.05
5000	0.54	0.47	0.12	0.13
15000	2.86	1.99	0.50	0.42

a significant amount of time, because the selective update property of the algorithm effectively discards a large number of particles at the very first step of the algorithm.

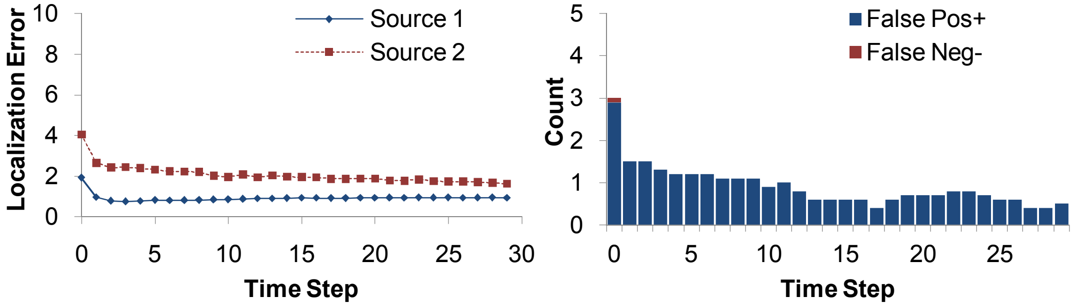
The algorithm takes a fraction of a second to execute on average, even with 196 sensors and 15,000 particles. However, we notice that the first few time steps take longer than average. This is due to the particles being uniformly and randomly distributed at the beginning of the algorithm. As time proceeds, the particles will concentrate at several spots. Therefore, the majority of the iterations only affects a small number of particles and thus shorter execution time.



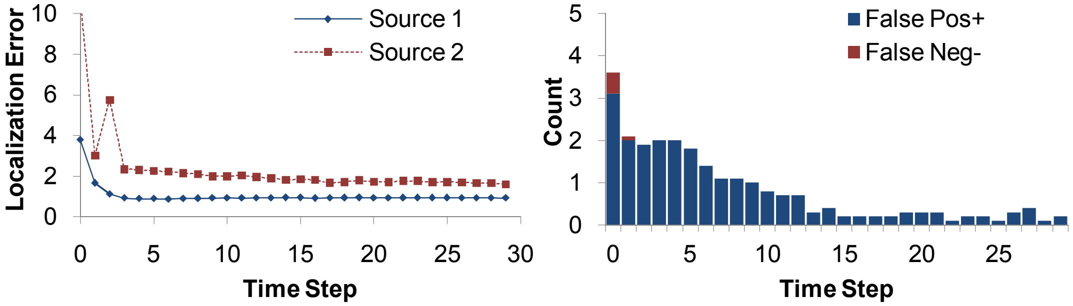
(a) 4 μCi, localization error (b) 4 μCi, false positives/negatives



(c) 10 μCi, localization error (d) 10 μCi, false positives/negatives

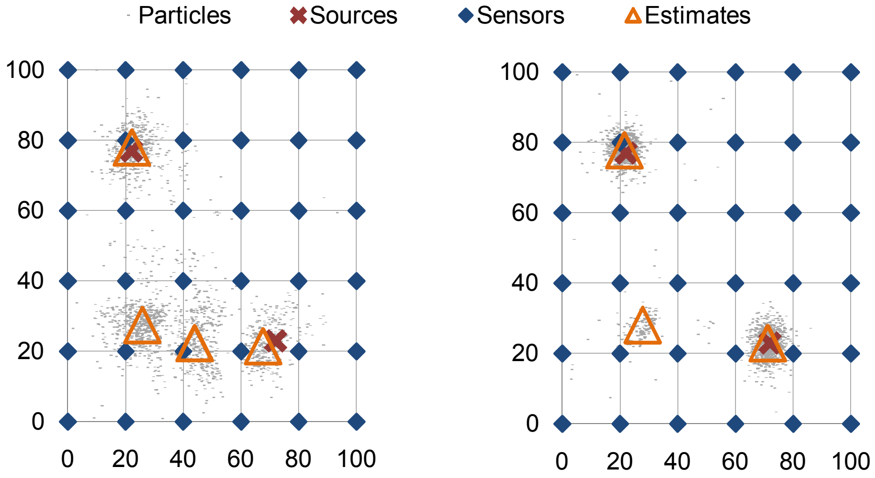


(e) 50 μCi, localization error (f) 50 μCi, false positives/negatives



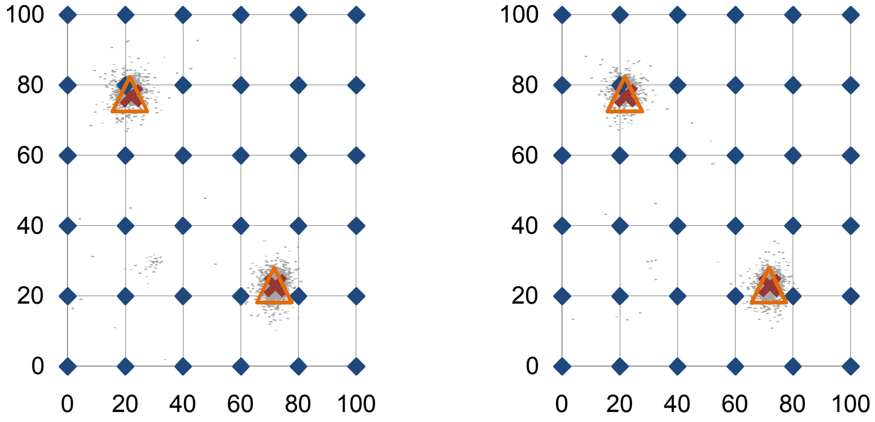
(g) 100 μCi, localization error (h) 100 μCi, false positives/negatives

Figure 5.12. Localization error (left) and false positives/negatives (right) for the two sources of different strengths, without the obstacle.



(a) Time step 1

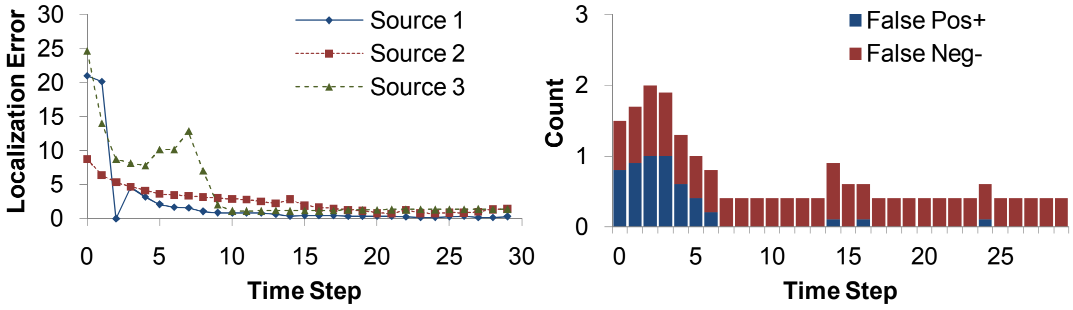
(b) Time step 3



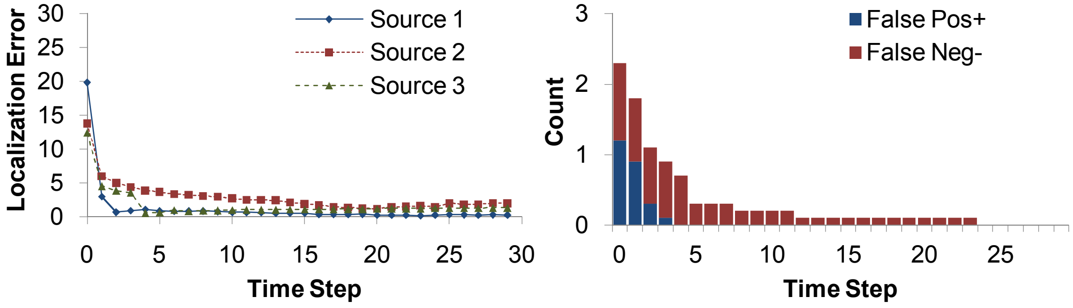
(c) Time step 5

(d) Time step 7

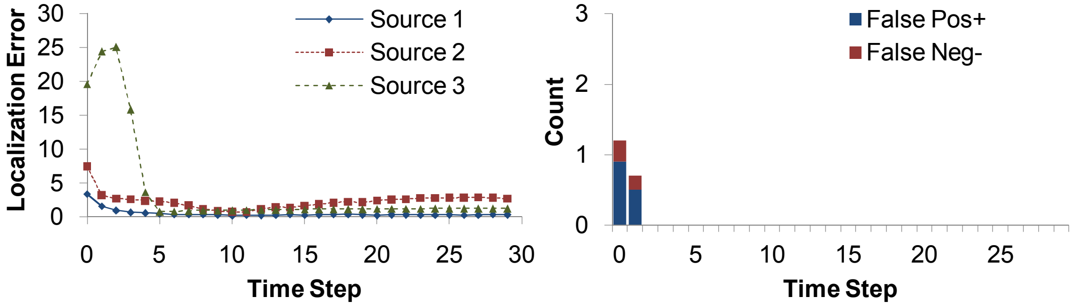
Figure 5.13. Progression of the particle filter over time.



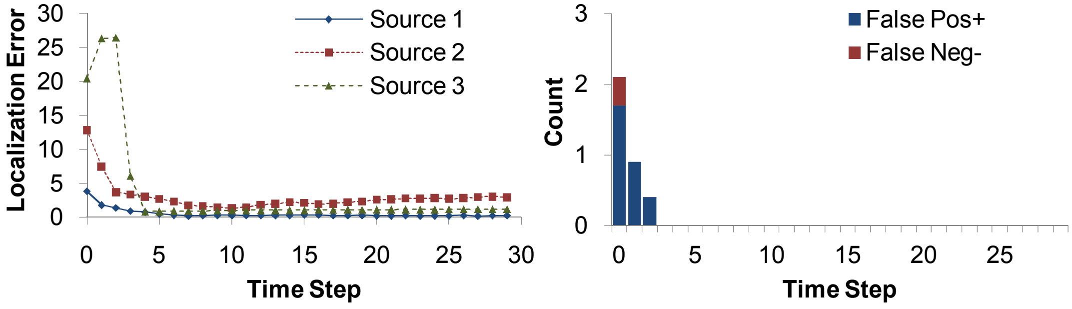
(a) 4 μCi, localization error (b) 4 μCi, false positives/negatives



(c) 10 μCi, localization error (d) 10 μCi, false positives/negatives

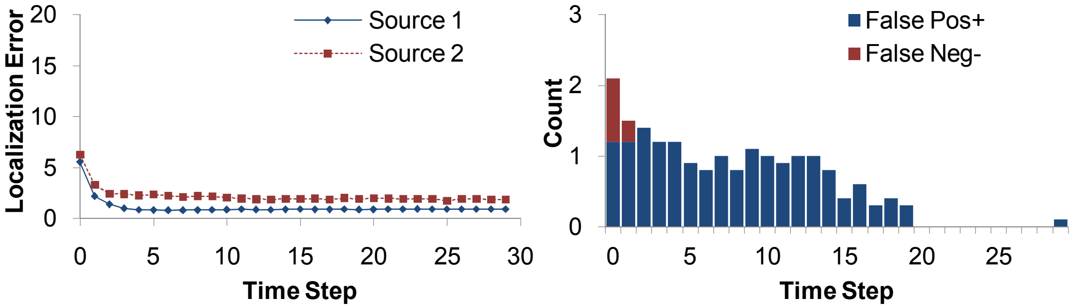


(e) 50 μCi, localization error (f) 50 μCi, false positives/negatives

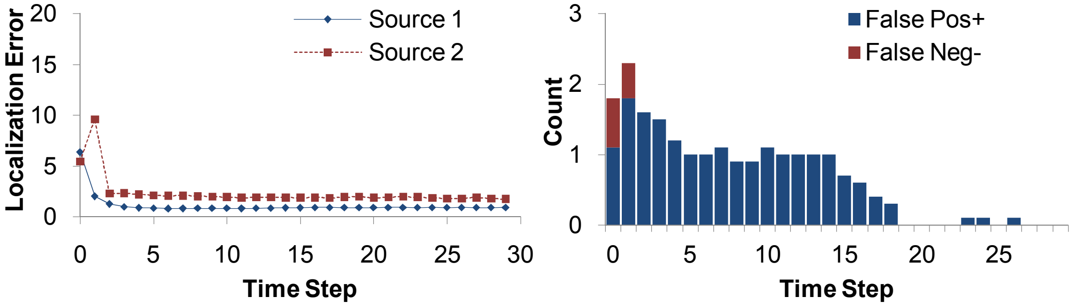


(g) 100 μCi, localization error (h) 100 μCi, false positives/negatives

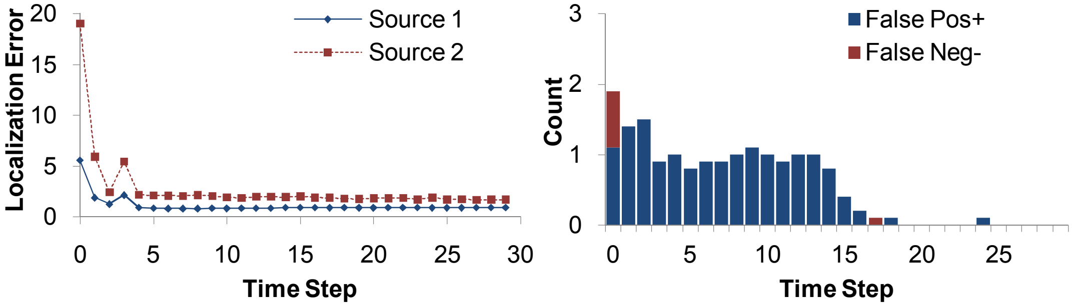
Figure 5.14. Localization error (left) and false positives/negatives (right) for the three sources of different strengths, without the obstacle.



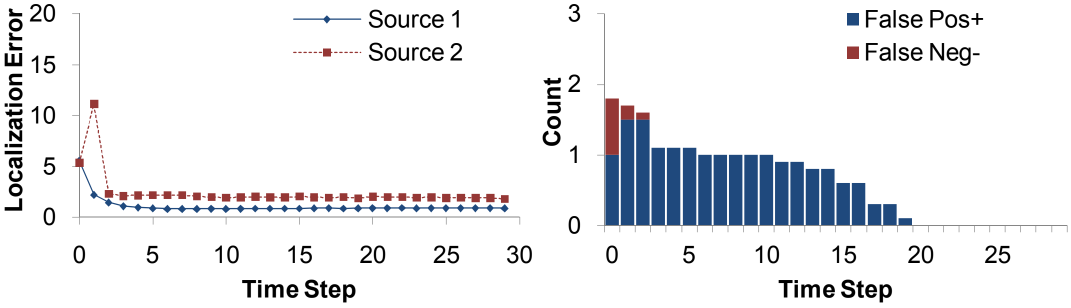
(a) 0 CPM, localization error (b) 0 CPM, false positives/negatives



(c) 5 CPM, localization error (d) 5 CPM, false positives/negatives



(e) 10 CPM, localization error (f) 10 CPM, false positives/negatives



(g) 50 CPM, localization error (h) 50 CPM, false positives/negatives

Figure 5.15. Localization error (left) and false positives/negatives (right) for the two sources under different background radiations, without the obstacle.

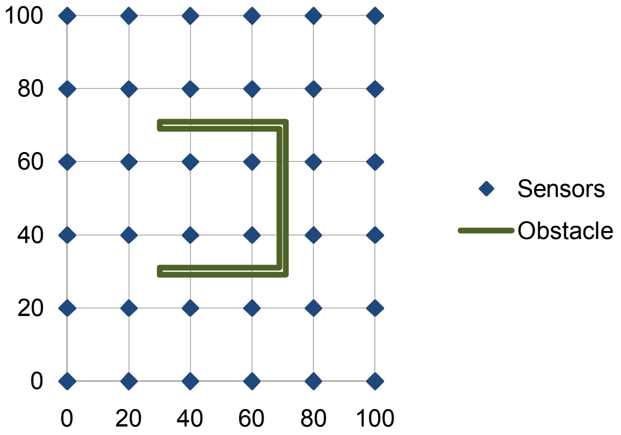
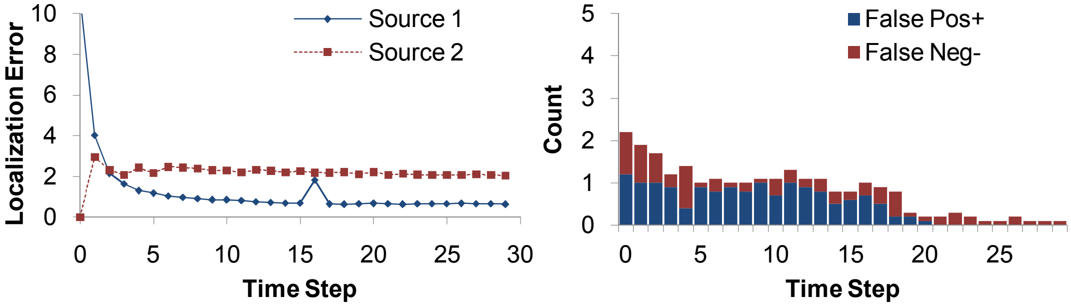
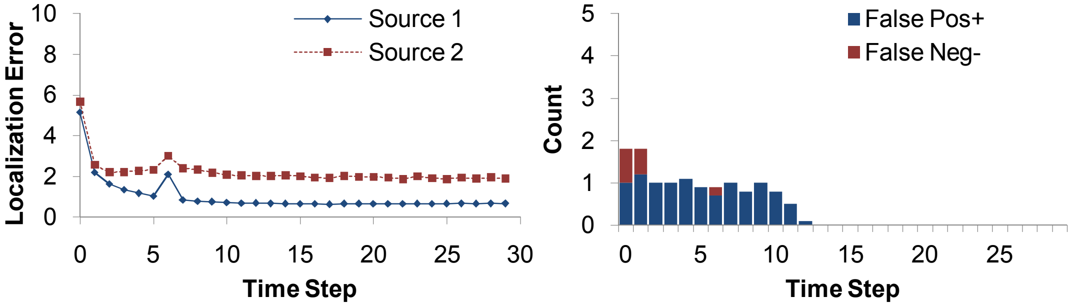


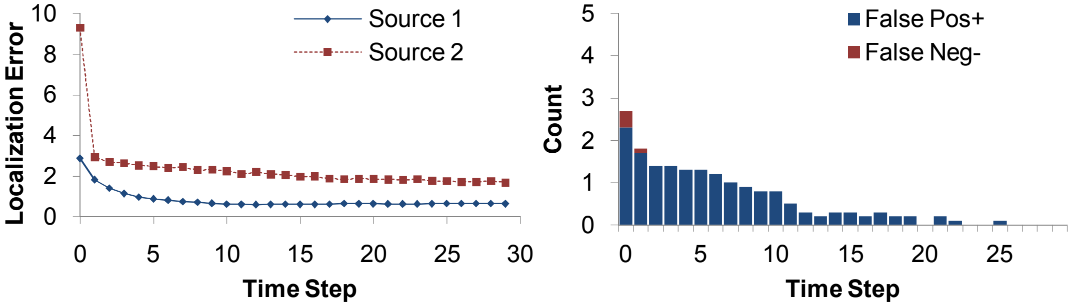
Figure 5.16. Location of the obstacle in the 36-sensor scenario.



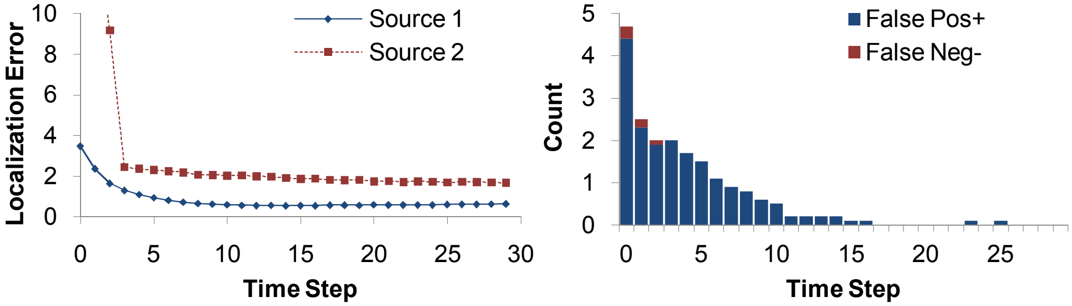
(a) 4 μCi, localization error (b) 4 μCi, false positives/negatives



(c) 10 μCi, localization error (d) 10 μCi, false positives/negatives



(e) 50 μCi, localization error (f) 50 μCi, false positives/negatives



(g) 100 μCi, localization error (h) 100 μCi, false positives/negatives

Figure 5.17. Localization error (left) and false positives/negatives (right) for the two sources of different strengths, with obstacle.

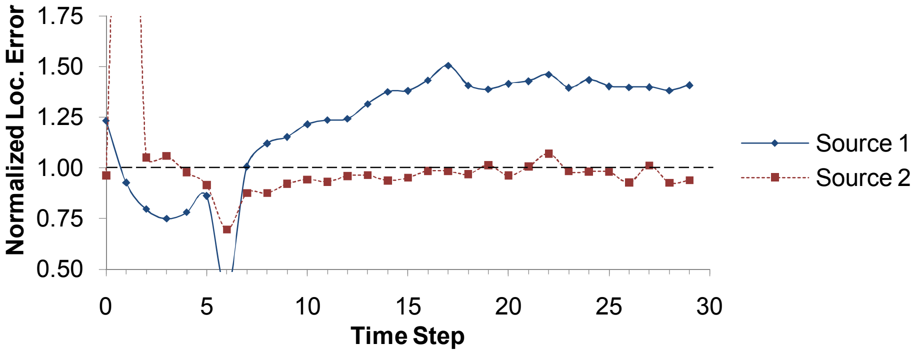


Figure 5.18. Normalized localization error of the 36-sensor scenario.

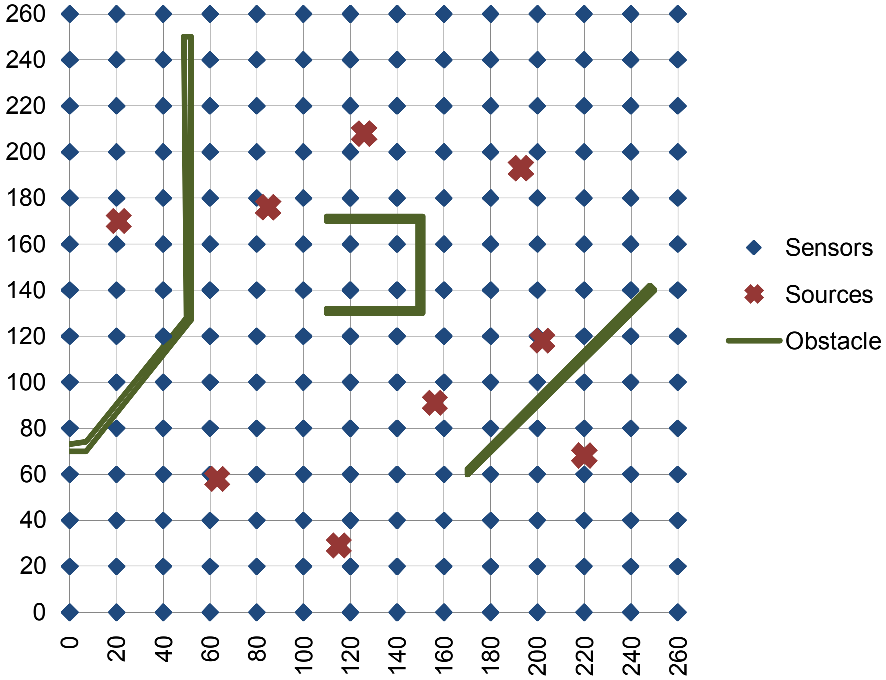
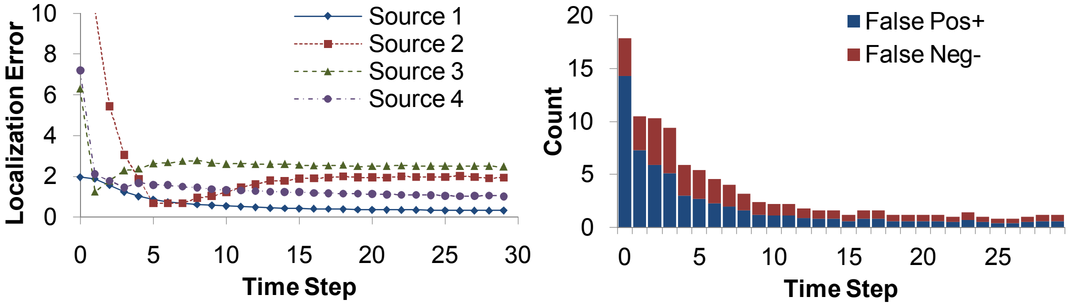
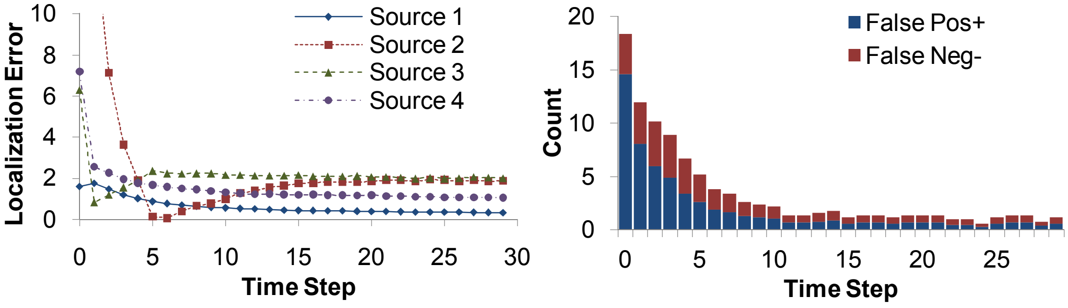


Figure 5.19. Location of the sensors, radiation sources, and obstacles in the 196-sensor scenario.



(a) Without the obstacle



(b) With the obstacle

Figure 5.20. Localization error and false positives/negatives in the 9-source scenario, with and without the obstacles.

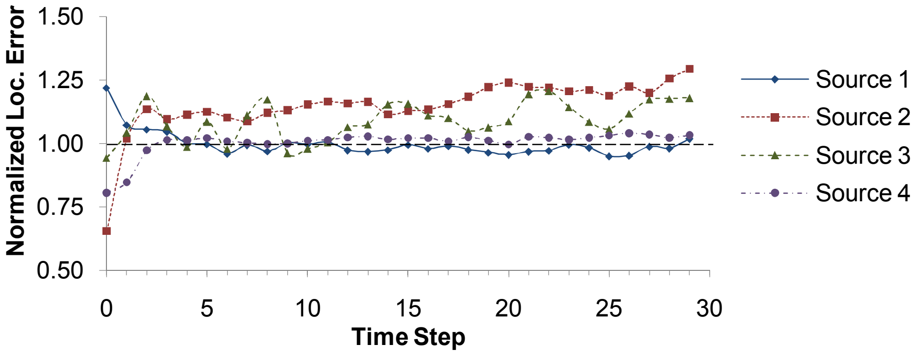


Figure 5.21. Normalized localization error of the 196-sensor scenario.

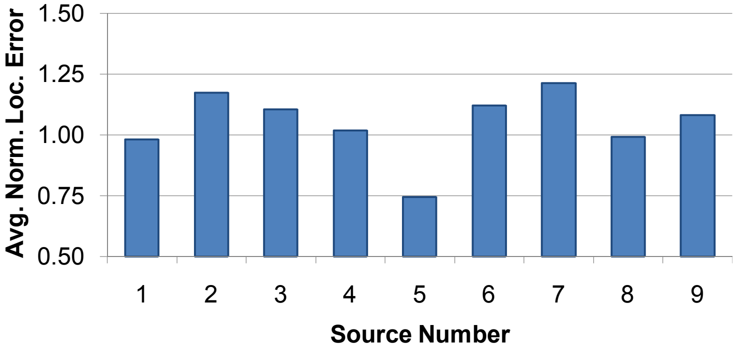


Figure 5.22. Localization error of the scenario without the obstacles normalized to that of the same scenario with the obstacles.

6 CONFIRMING THE PRESENCE OF A RADIATION SOURCE

In Chapter 3, we considered the localization of a radiation source. The algorithm presented there assumed that a radiation source was present, and therefore the location and strength of the radiation source could be estimated. In this chapter, we formulate the Localization Enhanced Detection (LED) algorithm that detects, and thus confirms, the presence of a radiation source.

The detection problem is particularly important when the radiation source is weak because the radiation level measured by the radiation sensors may be so low that it appears to be normal variations of background radiation. While a long-term average of the sensor measurements would result in elevated levels which could be detected eventually, the focus of this chapter is on detecting the radiation source quickly to ensure a fast response.

The detection of a weak radiation source is difficult due to two major reasons:

1. Varying background radiation: The background radiation may vary significantly from one deployment region to another as discussed in Section 2.2. Such variation may lead to false positives that could cause unnecessary panic if they are not interpreted carefully.

2. Probabilistic radiation measurements: Radiation sources generate probabilistic measurements as discussed in Section 2.3. Typically, gamma radiation from a point source follows the Poisson distribution. As the variance of the Poisson distribution is equal to the mean, we expect large fluctuations in the sensor measurements.

Due to the combined effect of the above factors, discriminating between measurements due to a radiation source versus background radiation is difficult. As a result, traditional detection methods that use *a priori* thresholds often return results corresponding to “ghost” sources—estimates that do not correspond to any actual radiation source.

In this chapter, we demonstrate that detection and localization of radiation sources can reinforce each other, in a two-step decision procedure. We present the Localization Enhanced Detection (LED) method—which subsumes the detection and localization problems—for a point radiation source using a network of radiation sensors. In the first step, we utilize a localization method described in Chapter 3 to estimate the location and strength of an actual or ghost radiation source. Consequently, using the estimated source parameters, we utilize a sequential probability ratio test (SPRT) to declare

1. the presence of a radiation source with the estimated parameters,
2. the absence of the radiation source, or
3. the insufficiency of the collected measurements to conclude.

The LED method utilizes the estimated background radiation of the given surveillance area and the estimated source parameters, to formulate a SPRT based on the Poisson point source model. Ghost sources, if estimated in the first step, will be rejected by the SPRT to achieve a specified false positive rate, as they do not lead to statistically consistent measurements. The estimated source parameters enable us to formulate a more specific SPRT compared to detecting a general increase in the radiation level, which in turn yields a decision with the least expected number of measurements. Our approach is in contrast with conventional approaches in which detection precedes localization as in several tracking applications [39, 40].

The balance of this chapter is organized as follows. In Section 6.1, we introduce the sequential probability ratio test for detecting a radiation source. Combining a single radiation source localization algorithm in Chapter 3 and the SPRT test, we present the Localization Enhanced Detection (LED) method in Section 6.2. Then, we evaluate the proposed LED method with a three sensor network and a N sensor network in Section 6.3 and Section 6.4, respectively.

6.1 Radiation Source Detection using Sequential Probability Ratio Test

Let $m_t(S_i)$ denote the measurement by sensor i at time t and $M_t(S_i) = \{m_j(S_i) \mid j = 1 \dots t\}$ denotes all the measurements collected by sensor i within a given time window $1 \dots t$. By the definition of the Poisson process with parameter λ, we have

$$\mathrm{P}(k) = \frac{\lambda^k e^{-\lambda}}{k!} \,. \tag{6.1}$$

Let $H_{c,i}$, where $c \in \{0, 1\}$, denote the hypothesis that the sensor measurements correspond to the intensity $\lambda_{c,i}$ at S_i, where

$$\lambda_{1,i} = \hat{A}^{\text{str}} \left(1 + \left|S_i - \hat{A}^{\text{pos}}\right|^2\right)^{-1} + B_i \tag{6.2}$$

and

$$\lambda_{0,i} = B_i \tag{6.3}$$

for the radiation source is present and not present, respectively. Utilizing the statistical independence property of the measurements, we define the likelihood function

$$\begin{aligned} L_{t,i} &= \frac{\mathrm{P}(M_t(S_i) \mid H_{1,i})}{\mathrm{P}(M_t(S_i) \mid H_{0,i})} \\ &= e^{(\lambda_{0,i} - \lambda_{1,i})t} \prod_{j=1}^{t} \left[\frac{\lambda_{1,i}}{\lambda_{0,i}}\right]^{m_j(S_i)} \end{aligned} \tag{6.4}$$

which measures the likelihood that the sensor measurements correspond to a radiation source of strength $\hat{A}^{\text{str}}$ and located at $\hat{A}^{\text{pos}}$, or just the background radiation. For clarity, we drop the subscript i and t where there is no ambiguity. Let the detection error probabilities be $\mathrm{P}(\text{accept } H_1 \mid H_0) = P_{10}$ (i.e., false positive rate) and $\mathrm{P}(\text{accept } H_0 \mid H_1) = P_{01}$ (i.e., false negative rate). It is shown that [41]

$$\mathrm{E}(L \mid H_0 \text{ is true and } H_1 \text{ accepted}) = \frac{1 - P_{01}}{P_{10}} \tag{6.5}$$

and

$$\mathrm{E}\left(L^{-1} \mid H_1 \text{ is true and } H_0 \text{ accepted}\right) = \frac{P_{01}}{1 - P_{10}} \,. \tag{6.6}$$

Equation (6.5) above states that the expected value of the likelihood ratio is $\frac{1-P_{01}}{P_{10}}$ when we have a false positive. In other words, if we declare a source to be present when $L > \frac{1-P_{01}}{P_{10}}$, the false positive rate will be less than P_{10}. Hence, we would obtain nearly the smallest possible number of samples if we declare a detection as soon as $L > \frac{1-P_{01}}{P_{10}}$. By a similar argument as Equation (6.6), we would obtain nearly the smallest possible number of samples if we declare non-detection as soon as $L < \frac{P_{01}}{1-P_{10}}$. Basically, the SPRT minimizes the expected number of steps needed to declare either H_1 or H_0. In particular, the SPRT minimizes $\mathrm{E}\left[\, t \mid H_0 \right]$ and $\mathrm{E}\left[\, t \mid H_1 \right]$ given the specified false positive and false negative rates P_{10} and P_{01}.

In summary, the SPRT procedure can be described as follows:

1. If $L < \frac{P_{01}}{1-P_{10}}$, declare no radiation source is present, namely hypothesis H_0.
2. If $L > \frac{1-P_{01}}{P_{10}}$, declare the presence of a radiation source with strength $\hat{A}^{\mathrm{str}}$ and located at $\hat{A}^{\mathrm{pos}}$, namely hypothesis H_1.
3. Else, declare that the measurements are insufficient to make a decision and continue collecting additional measurements.

The above SPRT procedure can be compactly expressed as

$$\frac{P_{01}}{1-P_{10}} < L_{t,i} < \frac{1-P_{01}}{P_{10}} \tag{6.7}$$

which also can be expressed in terms of the sum of measurements:

$$\frac{\ln \frac{P_{01}}{1-P_{10}} + t\left(\lambda_{1,i} - \lambda_{0,i}\right)}{\ln \lambda_{1,i} - \ln \lambda_{0,i}} < \sum_{j=1}^{t} m_j\left(S_i\right) < \frac{\ln \frac{1-P_{01}}{P_{10}} + t\left(\lambda_{1,i} - \lambda_{0,i}\right)}{\ln \lambda_{1,i} - \ln \lambda_{0,i}} \tag{6.8}$$

The above SPRT is derived under the assumption that the source and background radiation satisfy the Poisson distribution. While point radiation sources follow the Poisson distribution, background radiation may not as it could be a combination of multiple complex sources. In such a case, the false positive rate of the SPRT formulation can be different, and can be approximated by using an empirical distribution in computing $\mathrm{P}\left(M_t\left(S_i\right) \mid H_{0,i}\right)$ in Equation (6.4).

6.1.1 Expected Detection Time

In this section, we show that using an accurate estimate of the source strength minimizes the expected detection time. We proceed by dividing all sides of Equation (6.8) by t, yielding

$$\bar{m}\left(S_i\right) = \frac{1}{t}\sum_{j=1}^{t} m_j\left(S_i\right) < \frac{\ln \frac{1-P_{01}}{P_{10}} + t\left(\lambda_{1,i} - \lambda_{0,i}\right)}{t\left(\ln \lambda_{1,i} - \ln \lambda_{0,i}\right)} \tag{6.9}$$

for the upper threshold, and

$$\bar{m}\left(S_i\right) = \frac{1}{t}\sum_{j=1}^{t} m_j\left(S_i\right) > \frac{\ln \frac{P_{01}}{1-P_{10}} + t\left(\lambda_{1,i} - \lambda_{0,i}\right)}{t\left(\ln \lambda_{1,i} - \ln \lambda_{0,i}\right)} \tag{6.10}$$

for the lower threshold. As the following derivation is the same for both Equation (6.9) and (6.10), we express both equations as

$$\bar{m}\left(S_i\right) = \frac{1}{t}\sum_{j=1}^{t} m_j\left(S_i\right) \lessgtr \frac{\ln P + t\left(\lambda_{1,i} - \lambda_{0,i}\right)}{t\left(\ln \lambda_{1,i} - \ln \lambda_{0,i}\right)} \tag{6.11}$$

for $P \in \left\{\frac{P_{01}}{1-P_{10}}, \frac{1-P_{01}}{P_{10}}\right\}$. Solving the above equation for the value of t, we have

$$t = \frac{\ln P}{\left(\ln \lambda_{1,i} - \ln \lambda_{0,i}\right)\bar{m}\left(S_i\right) - \lambda_{1,i} + \lambda_{0,i}} \ . \tag{6.12}$$

The number of measurements required for SPRT to conclude H_1 (or H_0 respectively) is given by Equation (6.12) if t evaluates to a positive value. A non-positive value of t denotes that SPRT will never conclude H_1 (or H_0 respectively). Differentiating Equation (6.12) with respect to $\hat{A}^{\mathrm{str}}$ yields

$$\frac{\partial t}{\partial \hat{A}^{\mathrm{str}}} = \frac{\ln P\left(1 - \bar{m}\left(S_i\right)\lambda_{1,i}^{-1}\right)}{\left(1 + \left|S_i - \hat{A}^{\mathrm{pos}}\right|^2\right)\left(\bar{m}\left(S_i\right)\ln\frac{\lambda_{1,i}}{\lambda_{0,i}} - \lambda_{1,i} + \lambda_{0,i}\right)^2} \ . \tag{6.13}$$

By solving $\frac{\partial t}{\partial \hat{A}^{\mathrm{str}}} = 0$, we conclude that the SPRT achieves minimum detection time when

$$\lambda_1 = \bar{m}\left(S_i\right) \ . \tag{6.14}$$

The above result concludes that the time required to make a decision is minimal if the localization algorithm can provide an accurate strength and position estimate

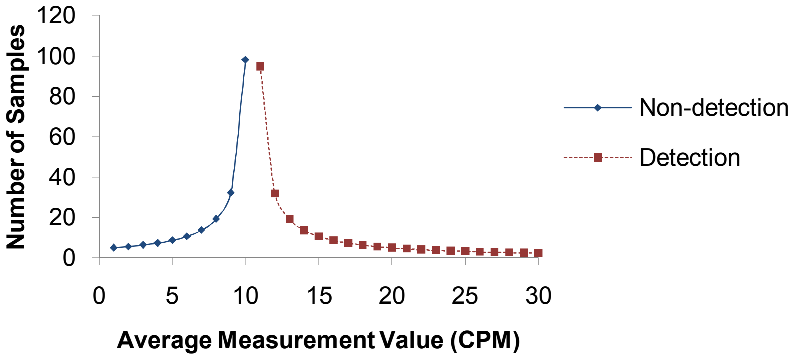


Figure 6.1. Expected number of samples needed for SPRT to make a decision.

of the radiation source as input to the SPRT detection. Because the background radiation, source strength, and location are estimated, errors in the estimation may cause an undesirably long detection time. Furthermore, the variance of $\bar{m}\left(S_i\right)$ is large because it is averaged over a small number of samples.

Figure 6.1 quantifies the average number of measurements required in the presence of imperfect estimations and measurements. The figure shows the expected number of measurements required for SPRT to conclude either H_1 or H_0 for a strength measurement that is only 10% higher than the background ($\lambda_0 = 10$ CPM, $\lambda_1 = 11$ CPM), and false postive and false negative rates both equal to $P_{01} = P_{10} = 0.01$. As expected, the SPRT does not conclude a detection when the average sensor measurement $\bar{m}\left(S_i\right)$ is lower than the estimated background radiation λ_0. The expected number of samples required to conclude non-detection increases as the average sensor measurement increases. When $\bar{m}\left(S_i\right) > \lambda_1$, the SPRT could conclude a detection. However, the SPRT could not immediately make the conclusion because the high measurement may be due to normal fluctuations in the sensor measurements. By considering more samples in the decision, the specified false positive and false negative rates are satisfied.

We use simulations to substantiate the above conclusion. We simulate a radiation sensor collecting measurements at a regular interval. First, the radiation sensor collects 1,500 measurement samples, in the presence of background radiation

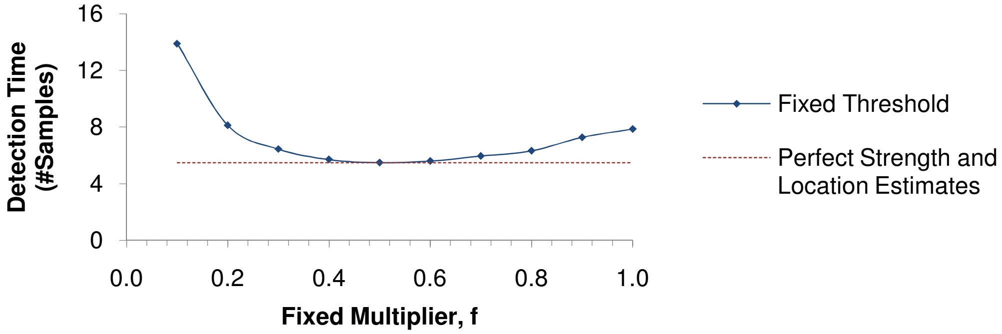


Figure 6.2. Average SPRT detection time with different thresholds.

only. The background radiation is modeled as a Poisson random variable with a mean of $\lambda_B = 10$ CPM. Subsequently, a radiation source is introduced and the measurements are modeled as a Poisson random variable with a mean of $\lambda_A = \lambda_B + A^{\mathrm{str}}\left(1 + |S_i - A^{\mathrm{pos}}|^2\right)^{-1}$. We perform SPRT on the measurements collected. In the first attempt, we use a fixed constant multiple f of the background estimate as $\lambda_1 = (1 + f)\, B_i$ and explore values of f between 0.1 and 1.0. In the second attempt, we assume a perfect estimate of the source strength and location, i.e., $\lambda_1 = \lambda_A$. Figure 6.2 shows the average detection time, with $\lambda_A = 15$ CPM and $P_{01} = P_{10} = 0.01$, using a perfect estimate of the strength and location versus using a fixed threshold for detection. The figure shows that the perfect strength and location estimates always yield the minimum detection time. This result indicates that if the strength and the location of the radiation source are not known in advance, the LED method will outperform a fixed threshold detection method.

Despite a shorter detection time, the LED method has a slightly higher false positive rate than the fixed threshold detection method, but the increment does not exceed the specified false positive rate P_{10}. Table 6.1 shows the average false positive rates of the fixed threshold method (the FT column), and the LED method (the LED column) with $f = 0.3$ and $P_{01} = 0.1$. The false positive rate of the LED method decreases as the source strength increases. The fixed threshold detection method, however, exhibits a relatively stable false positive rate. Table 6.2 shows the average

Table 6.1
False positive rates of the fixed threshold detection method (the FT column) and the LED method (the LED column).

	Achieved average false positive rates							
	$\lambda_A = \lambda_B + 1$		$\lambda_A = \lambda_B + 3$		$\lambda_A = \lambda_B + 5$		$\lambda_A = \lambda_B + 10$	
P_{10}	FT	LED	FT	LED	FT	LED	FT	LED
0.01	0.0055	0.0078	0.0055	0.0055	0.0055	0.0041	0.0055	0.0025
0.05	0.0291	0.0396	0.0293	0.0293	0.0291	0.0217	0.0289	0.0096
0.10	0.0593	0.0817	0.0594	0.0594	0.0593	0.0427	0.0589	0.0179
0.15	0.0868	0.1229	0.0869	0.0869	0.0868	0.0635	0.0865	0.0204
0.20	0.1161	0.1645	0.1162	0.1162	0.1161	0.0764	0.1156	0.0320
0.25	0.1455	0.2086	0.1458	0.1458	0.1457	0.1042	0.1452	0.0347

Table 6.2
False negative rates of the fixed threshold detection method (the FT column) and the LED method (the LED column).

	Achieved average false negative rates							
	$\lambda_A = \lambda_B + 1$		$\lambda_A = \lambda_B + 3$		$\lambda_A = \lambda_B + 5$		$\lambda_A = \lambda_B + 10$	
P_{10}	FT	LED	FT	LED	FT	LED	FT	LED
0.01	0.6328	0.0124	0.0069	0.0069	0.0001	0.0050	0.0000	0.0026
0.05	0.6726	0.0473	0.0335	0.0335	0.0008	0.0251	0.0000	0.0151
0.10	0.6961	0.0905	0.0647	0.0647	0.0031	0.0490	0.0000	0.0273
0.15	0.7145	0.1334	0.0991	0.0991	0.0081	0.0722	0.0000	0.0302
0.20	0.7283	0.1765	0.1339	0.1339	0.0149	0.0935	0.0001	0.0464
0.25	0.7342	0.2170	0.1580	0.1580	0.0210	0.1293	0.0003	0.0501

false negative rates with $f = 0.3$ and $P_{10} = 0.1$. The result suggests that obtaining accurate strength and location estimates of the radiation source is desirable as it reduces the false negative rate. As shown in the table, when $\lambda_A < (1 + f)\,B_i$, the false negative rate increases beyond the predefined threshold P_{01}. This is because the fixed threshold method expects the source strength to be higher than the threshold $(1 + f)\,B_i$. As the actual source strength is lower than the predefined threshold in this case, the method incorrectly concludes that no source is present, which leads to a high false negative rate.

6.2 The Localization Enhanced Detection Method

In this section, we develop a radiation source detection method by combining a localization method presented in Chapter 3 with the SPRT detection method. The LED method is designed to be executed at a centralized server. Initially, the system is put into training mode where the background radiation measurements are collected by each sensor and are then averaged to estimate the local background radiation level B_i for all sensors $1 \leq i \leq N$. Then, the network is put into monitoring mode, and the detection of the radiation source is achieved using the following procedure:

1. Using the measurements from the sensors, we estimate the source location and strength using a single source localization algorithm presented in Chapter 3.

2. Given the estimated location and strength of the radiation source from the previous step, the threshold $\lambda_{1,i}$ and $\lambda_{0,i}$ are determined for all sensors $1 \leq i \leq N$.

3. We utilize SPRT $L_{t,i}$ to conclude H_1 versus H_0 at each sensor i. We declare a source is present or absent if and only if the respective threshold conditions are satisfied at a majority of the sensors. Otherwise, more measurements are collected.

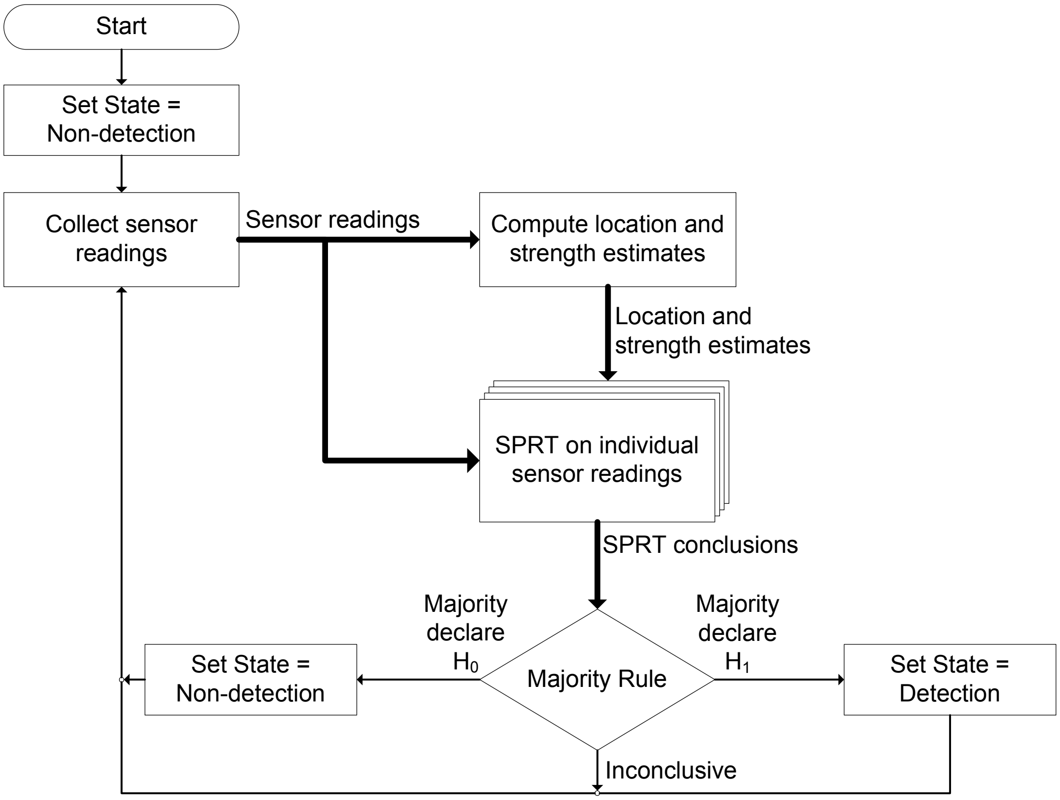
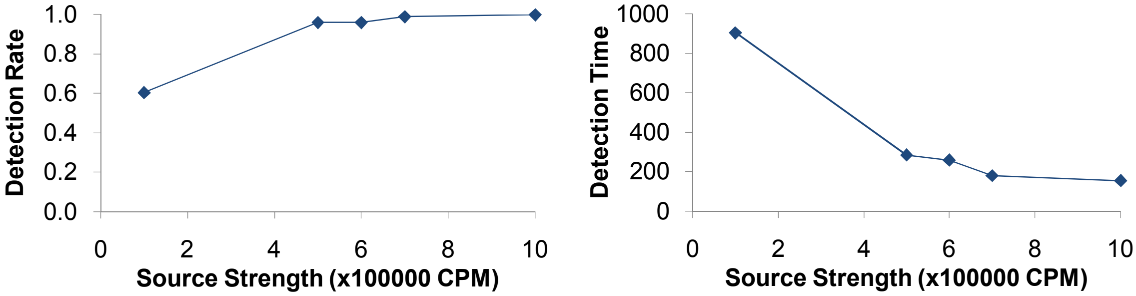


Figure 6.3. Functional diagram illustrating the LED method. Thick and thin arrows denote vector and scalar data, respectively.

Initially, the default hypothesis is H_0, and this hypothesis changes only if H_1 is declared by a majority of the sensors. The above procedure has the lowest false positive rate of the majority of sensors that declare H_1 in order to assert the presence of a source. Figure 6.3 shows a functional diagram illustrating the LED method.

6.3 Evaluation of the LED Method with Three Sensors

We evaluate the above LED method using a network of three sensors in simulation. We measure the effectiveness of the method by evaluating the detection rate, the false positive rate, and the detection time. In the simulations, we randomly generate 1000 radiation sources of strengths $A^{\text{str}} = 10^5, 5 \times 10^5, 6 \times 10^5, 7 \times 10^5, 10 \times 10^5$ CPM. These



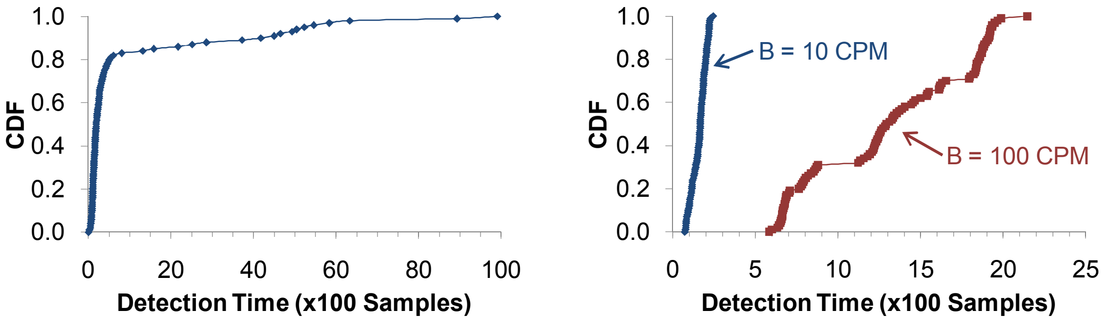
(a) Average detection rates.
(b) Average detection times.

Figure 6.4. Performance of the LED method with 3 sensors.

levels of source strength correspond to the actual radiation sources between 2 and 23 μCi when measured by a radiation sensor with efficiency $E = 0.02$ CPM/DPM. The average increase in radiation level over background level, at these levels of source strength, is below 10% for most of the cases. However, over a short period of time, variations in the background measurements could exceed 100%. In simulations, we use the LgDTOA algorithm to estimate the source location and strength.

Figure 6.4(a) shows the detection rates with various source strengths for $P_{01} = P_{10} = 0.1$. Although the average increase in the radiation level at the sensor locations is within 5–10% for $A^{\text{str}} = 5 \times 10^5$ CPM, the detection rate of the LED method is higher than 95%. In addition, the result shows that the detection rate for source strengths above $A^{\text{str}} = 10 \times 10^5$ CPM is 100%. The detection time shows a decreasing trend with increasing source strength as indicated in Figure 6.4(b). The trend is expected as it is easier to detect stronger radiation sources. The average detection time is less than 300 samples (or measurements) for $A^{\text{str}} \geq 5 \times 10^5$ CPM, even though the average increase in the radiation level at the sensor locations is within 5–10%. However, the detection times show large variations as illustrated in Figure 6.5(a), for the case of $A^{\text{str}} = 10 \times 10^5$ CPM. As shown in Figure 6.5(a), the detection time is less than 500 samples on 800 out of 1000 trials.

When no radiation source is present, the high threshold for H_1 will not be met. As a result, the source estimate produced by the localization step will be discarded. In



(a) With a $A^{\text{str}} = 10 \times 10^5$ CPM source and $B = 10$ CPM background

(b) No radiation source and $B = 10,\ 100$ CPM background

Figure 6.5. Cumulative probability distribution of detection times with 3 sensors.

our simulations with 1000 measurements and $B = 10,\ 100$ CPM, the proposed method does not generate any false positives. The average times to conclude non-detection are 159 and 1309 samples for $B = 10$ CPM and $B = 100$ CPM, respectively. The cumulative probability distribution of the detection times with no radiation source is shown in Figure 6.5(b).

When the expected background radiation level varies among sensors, the variance of the sensor measurements is different from one sensor to another. In other words, sensors with a higher background radiation level show larger fluctuations in the measurements. As a result, the location and strength estimates of the radiation source become less accurate and thus lead to a longer detection time. We evaluate the effects of background radiation level by comparing the detection time and detection rate for dissimilar background levels where $\langle B_1, B_2, B_3 \rangle = \langle 10, 20, 30 \rangle$ and $\langle 10, 30, 50 \rangle$, and with uniform background level $B_1 = B_2 = B_3 = 10$ CPM. Figure 6.6 shows the simulation result for different background radiation levels at different sensor locations. The solid, dotted, and dash lines denote a uniform background level ($B_1 = B_2 = B_3 = 10$ CPM), $\langle B_1, B_2, B_3 \rangle = \langle 10, 20, 30 \rangle$, and $\langle B_1, B_2, B_3 \rangle = \langle 10, 30, 50 \rangle$, respectively. The sensors are located at $S_1 = (0, 0)$, $S_2 = (1000, 0)$, $S_3 = (600, 1000)$, and $P_{01} = P_{10} = 0.1$. The result in Figure 6.6 agrees with our prediction. The average (and median) detection time with a uniform background radiation level is 230.96 (median = 37) samples,

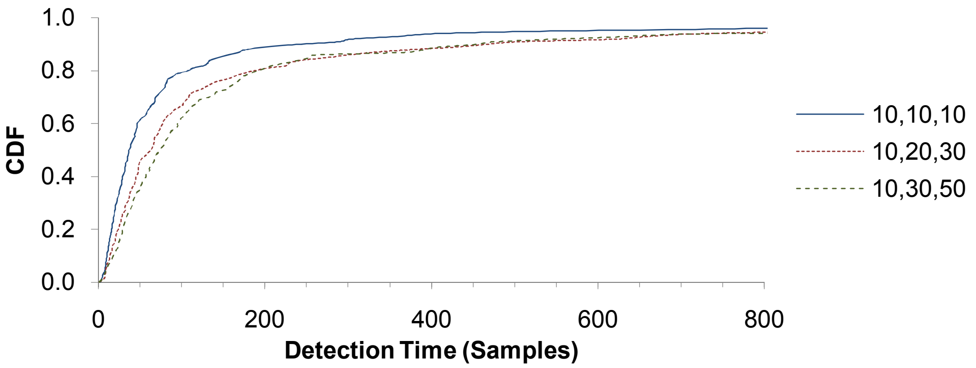


Figure 6.6. Cumulative probability distribution of detection times for different backgrounds at different sensor locations using the 3 sensors LED method.

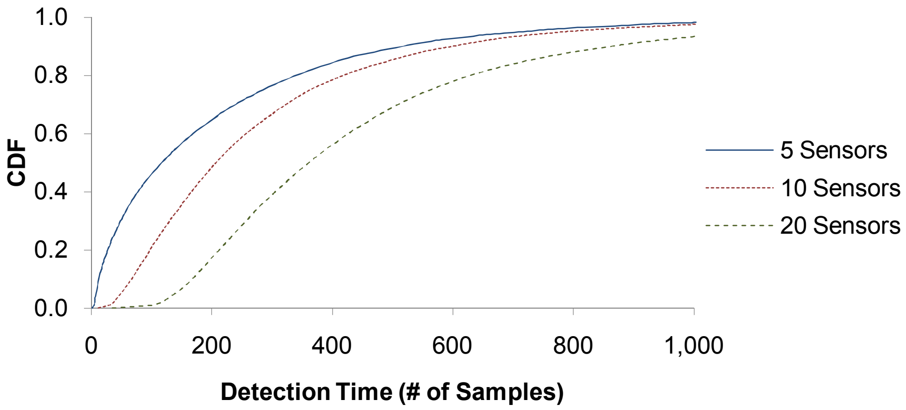
whereas the average (and median) detection time for non-uniform background levels are 341.56 (median = 63) samples and 619.83 (median = 74) samples for $\langle B_1, B_2, B_3 \rangle$ = $\langle 10, 20, 30 \rangle$ and $\langle 10, 30, 50 \rangle$, respectively. In all cases, the achieved detection rate is 94%, which is higher than the specified 90% detection rate.

It is instructive to compare the LED method with existing detection approaches:

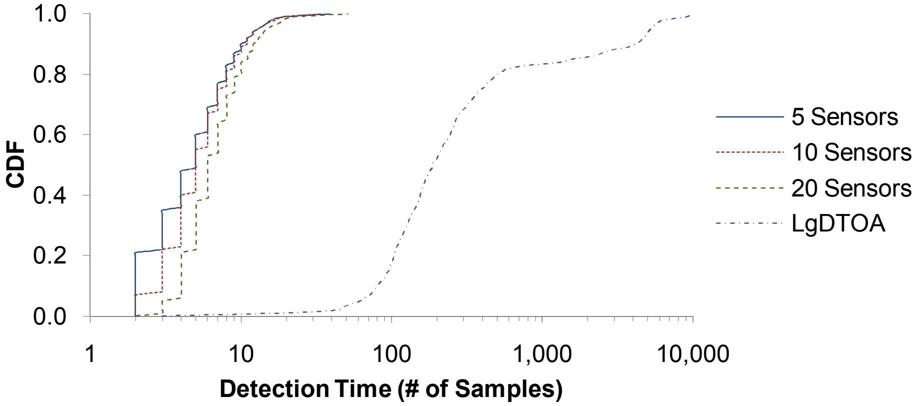
1. Compared with the existing detection methods, our method has a more focused goal of detecting a point radiation source rather than a general increase in radiation measurements. The SPRT guarantees that it is uniformly the most powerful test at a given false positive rate, in terms of maximizing the detection rate and minimizing the detection time.

2. Compared with the existing estimation methods, the ghost source phenomenon is strictly controlled by the specified false positive rate in the LED method. Furthermore, the *in situ* estimation of background radiation levels makes it sensitive to variations in the background radiation across the deployment region.

3. Compared with the existing methods that utilize a detection method followed by estimation, the LED method achieves a lower false positive rate since SPRT does not have to account for all the possible source locations and strengths.



(a) ITP, $A^{\mathrm{str}} = 10^5$ CPM



(b) ITP *vs.* LgDTOA, $A^{\mathrm{str}} = 10 \times 10^5$ CPM

Figure 6.7. Cumulative probability distribution of detection times with varying number of sensors.

6.4 Evaluation of the LED Method with N Sensors

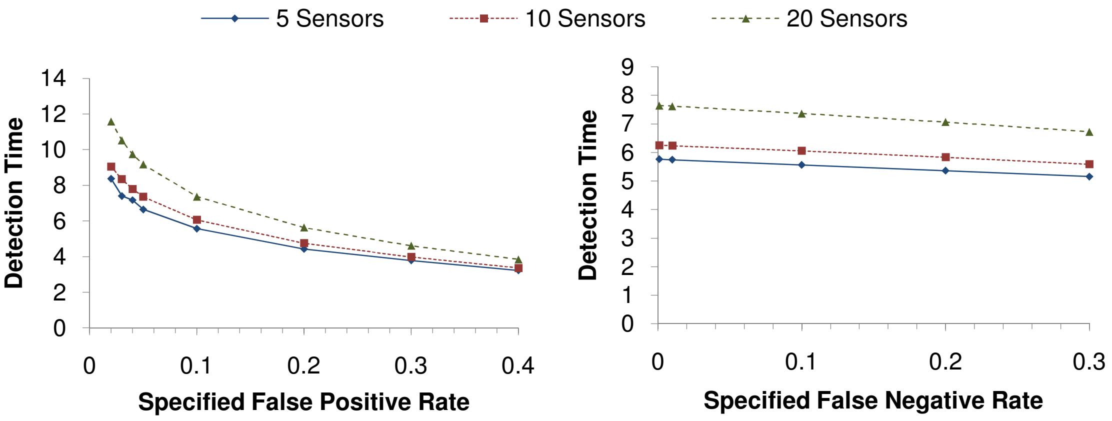
In this section, we evaluate the LED method in conjunction with the N-sensor ITP algorithm discussed in Section 3.2. Similar to the 3-sensor case, we measure the effectiveness of the algorithm by evaluating the detection rate, false positive rate, and detection time in simulation. We use the same simulation setup as in Section 6.3, except that the ITP algorithm is now used as the localization algorithm. Although the number of sensors has increased, the size of the surveillance area remains unchanged in the simulations.

Table 6.3
Statistics of detection time with varying number of sensors.

	$A^{\mathrm{str}} = 10^5$ CPM			$A^{\mathrm{str}} = 10 \times 10^5$ CPM		
Number of Sensors	5	10	20	5	10	20
Average	209.65	284.09	454.35	5.56	6.06	7.37
Median	117	207	360.5	5	5	6
StDev	263.94	256.20	328.49	3.76	3.58	3.96

Figure 6.7 and Table 6.3 show the cumulative probability distribution and statistics, respectively, of the detection time for $A^{\mathrm{str}} = 10^5$ CPM and $A^{\mathrm{str}} = 10 \times 10^5$ CPM. The background radiation level of the experiment is $B = 10$ CPM, and $P_{10} = P_{01} = 0.1$. The result shows that a system with $N > 3$ sensors achieves a faster detection time compared to the case with 3 sensors. With 20 sensors, the median detection time is merely 6 samples compared to 188 samples with 3 sensors (the LgDTOA plot in Figure 6.7(b) shows the detection time in log scale). The improvement is due to the increased localization accuracy of the ITP algorithm, which increases as the number of sensors increases. As discussed in Section 6.1.1, the detection time is at the minimum when source strength and location estimates are exact.

The detection time of a weaker source is longer because such a source is more difficult to distinguish from background radiation. This is shown in Figure 6.7, in which the median detection time is 117 samples for $A^{\mathrm{str}} = 10^5$ CPM radiation source, compared with 5 samples for $A^{\mathrm{str}} = 10 \times 10^5$ CPM radiation source. In addition, the detection time grows as the number of sensors increases as indicated in Figure 6.7. Comparing Figure 6.7(a) and Figure 6.7(b), the median detection time increases from 117 samples for 5 sensors, to 360.5 samples for 20 sensors. The increase in the detection time is due to the fact that more sensors are required to agree with the same conclusion before a decision is made. For instance, only three sensors need to conclude



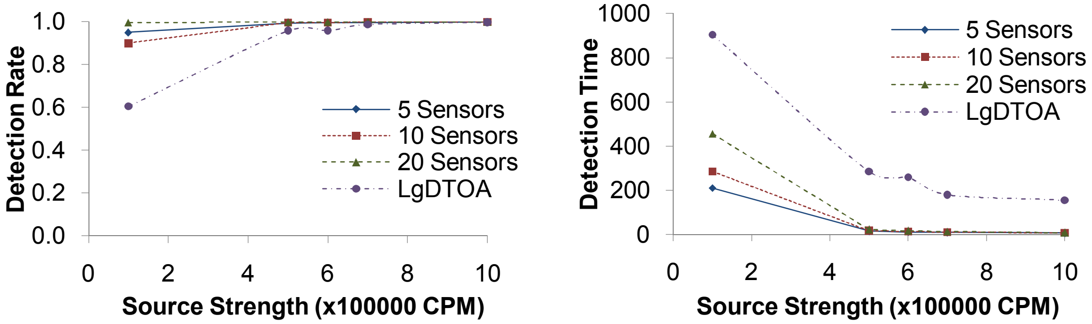
(a) Varying false positive rate with $P_{01} = 0.1$.

(b) Varying false negative rate with $P_{10} = 0.1$.

Figure 6.8. Detection time with varying false positive and false negative rates. $B = 10$ CPM, $A^{\text{str}} = 10 \times 10^5$ CPM.

H_1 in a network of 5 sensors before declaring a detection. However, 14 sensors are needed to conclude H_1 in a network of 20 sensors before declaring a detection. With a weak radiation source, more time is needed for many sensors to reach the same conclusion. Furthermore, some of the sensors located far away from the source will record measurements that have little difference from background radiation.

For real deployments, it is often useful to have short detection times, so that any identified threat can be handled promptly. Shorter detection times can be achieved by relaxing the specified false positive rate of the system, but there is a trade-off between fast detection and reliability. Figure 6.8(a) shows that increasing the specified false positive rate from 2% to 10% reduces the detection time by as much as 50%. Beyond 10%, the reduction in detection time becomes smaller as the specified false positive rate increases further. On the other hand, varying the specified false negative rate does not reduce the detection time significantly, as shown in Figure 6.8(b). The false positive rate, but not the false negative rate, determines the detection time because each additional measurement generally increases the likelihood that a radiation source is present (as the radiation source is actually present). The state of the system remains unchanged until the certainty about the radiation source has reached the specified



(a) Average detection rates.

(b) Average detection times.

Figure 6.9. Average detection rates and times of the LED method.

level, which is determined by the false positive rate and not the false negative rate. If the sensor measurements are close to the background level, e.g. there is no radiation source, then the false negative rate plays a major role in the decision of the LED method.

The detection rate metric shows the sensitivity of the LED method when dealing with weak radiation sources. Figures 6.9(a) and 6.9(b) show the detection rate and the corresponding detection time for various source strengths: 1×10^5, 5×10^5, 6×10^5, 7×10^5, 10×10^5, and 11×10^5 CPM. The background radiation in the experiment is $B = 10$ CPM, and $P_{10} = P_{01} = 0.1$. The results show that the LED method can effectively detect very weak radiation sources. The detection rate is close to 100% and the average detection time is less than 10 samples most of the time. The only exception is when the source is extremely weak, at 1×10^5 CPM, and there are only 5 or 10 sensors. In such a case, the detection rate drops to 95% (for 5 sensors) or 90% (for 10 sensors). The detection rate for 10 sensors is lower than that for 5 sensors, because in the 10 sensor case, some of the sensors are too far away for reliable measurements of the weak radiation source. As a result, it is more difficult for the algorithm to get a majority vote of the sensors to conclude a detection, thus lowering the detection rate. Even so, the false negative rate does not drop below the specified false negative rate of 10%.

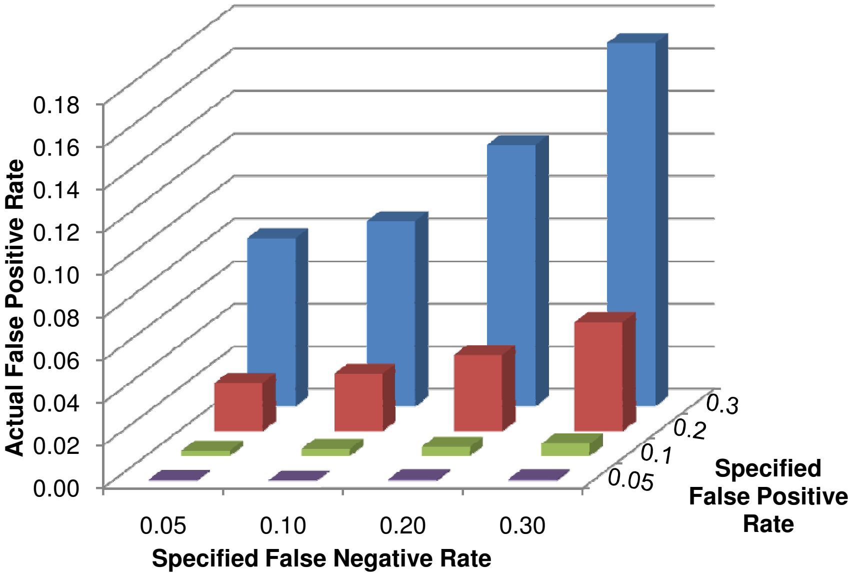


Figure 6.10. False positive rates of the LED method with 5 sensors using the ITP localization algorithm. $B = 10$ CPM.

The result illustrates the ability of the LED method to guarantee the detection rate. The cost for providing the guarantee is in the longer decision time when the measurements are less clear, i.e., when the source is weak. This phenomenon is illustrated in Figure 6.9(b), where the detection time for a 1×10^5 CPM radiation source is much longer than for stronger sources. Comparing Figure 6.9(a) and 6.9(b), even a 20-sensor network gives a detection rate close to 100% when given a weak radiation source, but the detection time is longer.

Compared with the LED method using 3 sensors (the LgDTOA plot in Figure 6.9), detection using N sensors improves the detection rate and detection time significantly, especially for weak radiation sources. For example, with a 1×10^5 CPM source, the detection time decreases from 904 samples for 3 sensors to 210 samples for 5 sensors. For a stronger radiation source such as $A^{\mathrm{str}} = 10 \times 10^5$ CPM, the detection time decreases from 154 samples to merely 6 samples.

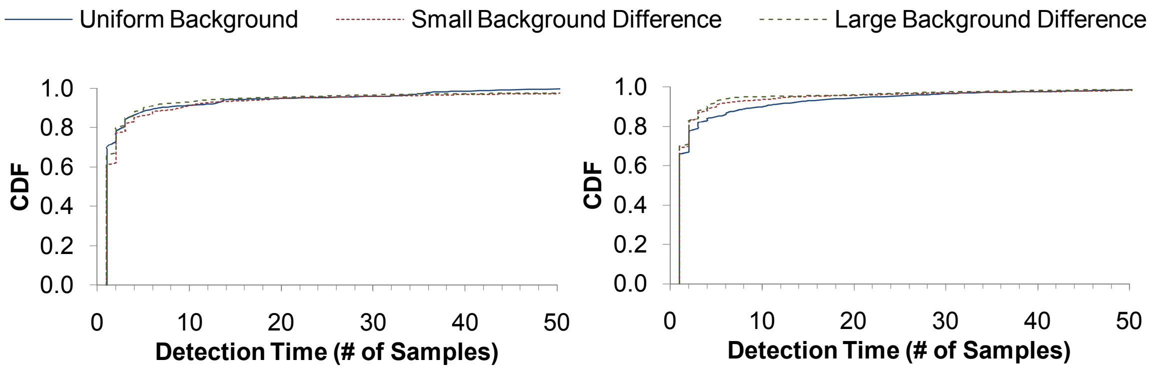
A false positive occurs when the LED method concludes a detection when there is in fact no radiation source present. This situation happens because radiation

Table 6.4
Background radiation level at each sensor.

	B_i (CPM)		
i	Uniform	Small var.	Large var.
1	10	14	40
2	10	19	8
3	10	18	37
4	10	18	19
5	10	26	23
6	10	22	63
7	10	36	47
8	10	13	51
9	10	28	3
10	10	19	24
11	10	39	15
12	10	15	49
13	10	16	40
14	10	22	40
15	10	10	52
16	10	33	10
17	10	24	56
18	10	28	29
19	10	38	21
20	10	24	30

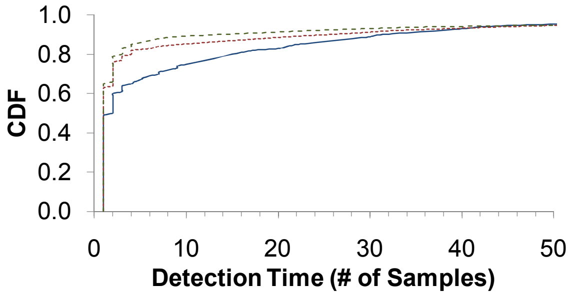
measurements are highly variable even when only background radiation is present. To evaluate the false positive rate of the LED method, we simulate scenarios in which there is no radiation source present and P_{10} and P_{01} are varied between 0.05 and 0.3. The result in Figure 6.10 shows that the actual false positive rates achieved are well below the specified threshold. With 5 sensors in the surveillance area, the false positive rate is only 0.33% for $P_{01} = P_{10} = 10\%$. With 10 and 20 sensors, there are no false positives at all. The result shows that the LED method is effective in rejecting ghost sources and in detecting the actual radiation sources. Moreover, the LED method is able to exploit information from additional sensors to arrive at a more accurate decision.

Unlike the LED method with 3 sensors, varying background radiation levels have minimal impact on the LED method with N sensors. We simulate three scenarios



(a) 5 Sensors

(b) 10 Sensors



(c) 20 Sensors

Figure 6.11. Cumulative probability distribution of detection times for different backgrounds at different sensor locations using the N sensor LED method.

with 5, 10, and 20 sensors, and three variations of the background radiation level. The first variation has all background levels fixed at 10 CPM. The second variation has the background levels randomly generated from a normal distribution with $\mu = 20$ CPM and $\sigma = 10$ CPM, and the third variation has $\mu = 30$ CPM and $\sigma = 20$ CPM. The exact values being used are listed in Table 6.4. The false positive rate, false negative rate, and source strength being used in the simulations are $P_{01} = P_{10} = 0.1$ and $A^{\text{str}} = 5 \times 10^5$ CPM, respectively. The simulation results in Figure 6.11 show that the detection time does not change significantly even when the background radiation varies significantly. The reason is that the larger background radiation actually reduces the detection time slightly due to a larger variance in the measurements. Although this may increase the false positives at individual sensors, these false positives do not impact the overall performance of the system with an increased number of sensors participating in the detection. The detection rate of the system remains close to 100% even when background radiation varies significantly. This finding agrees with the result presented in Figure 6.9.

7 SOFTWARE FRAMEWORK

We have built an extensive software framework to support experiments on radiation source detection and localization. The software framework is designed so that it can be readily deployed on actual systems with minimal changes. In addition, the software framework is general enough to support application domains that do more than just radiation source localization. For instance, in Section 7.3, we will demonstrate the use of our software framework in a structural assessment network.

The software framework follows the object-oriented programming paradigm and is built on top of Microsoft .NET framework. The .NET framework is selected because it is platform independent and cross language compatible. As a result, our software can be used on both the Microsoft Windows operating systems and many variants of the Linux operating system with Mono [42] installed. With cross language compatibility, users are not restricted to a particular programming language. Users may use a large variety of languages such as C++, C#, Visual Basic .NET, Fortran, and Pascal to build application software or run experiments using our software framework.

The software framework has two components. The first is the sensor driver interface which provides an abstraction layer for all sensor hardware. This allows applications to communicate with a large variety of sensors without worrying about the low-level, device specific details. In addition, this allows future extensions of the software framework to support new classes of sensors. Second, a library of tools and localization algorithms are included in the software framework, which enables users to build a localization system quickly as well as to evaluate the performance of localization algorithms.

This chapter is organized as follows. In Section 7.1, we describe the architecture of the sensor driver interface, and present the implementation of the RFTrax radiation sensor driver. In Section 7.2, we present the algorithms and tools included in the software framework. Finally, in Section 7.3, we present a further application for structural health monitoring—the Rapid Structural Assessment Network—to demonstrate the extensibility of the software framework.

7.1 Sensor Driver Interface

The sensor driver interface encapsulates the functionalities of generic sensor devices. The interface consists of two parts: The first part is an interface for the sensors, as defined by the `ISensor` class. This interface represents a physical sensor device such as a RFTrax radiation sensor. Applications may access the features of the sensors, such as hardware configurations and battery power level, through this interface. The second part is an interface of a sensing channel, as defined by the `ISensorChannel` class. The sensing channel is defined as a separate interface from the sensor device because a sensor device may contain multiple sensors. For instance, a single MDA300 multi-sensor module from Crossbow Technology [43] includes temperature and humidity sensors, as well as eleven analog-to-digital (ADC) input and six digital input channels.

7.1.1 RFTrax Sensor I/O

We have implemented a driver for the RFTrax radiation sensor. The RFTrax radiation sensor is a small form-factor radiation sensor with an embedded micro-controller. The micro-controller handles data collection and processing as well as data I/O. The sensor uses a RS-485 I/O interface for communication. The RS-485 I/O interface allows up to 32 sensors to be daisy-chained to the same network (see Figure 7.1). Because multiple sensors may share the same network, it is crucial that data communications are serialized correctly. For instance, when the host machine requests sensor measurements, the host machine needs to wait for a reply from the sensor before sending another packet to that same sensor, or to any other sensors on

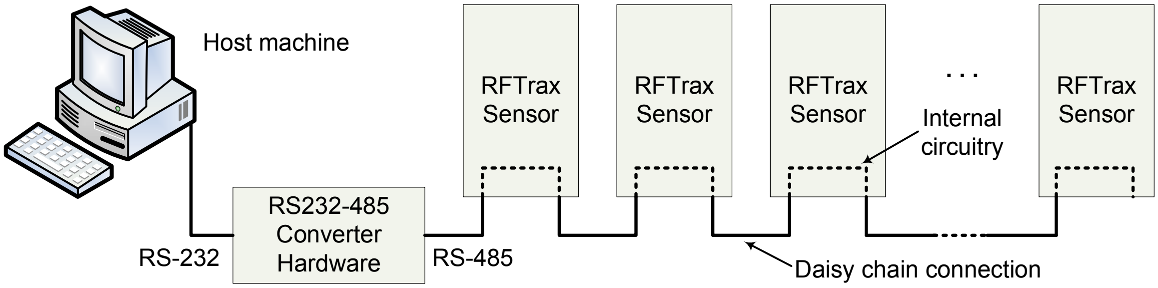


Figure 7.1. Daisy chain connection of RFTrax radiation sensors.

the same network. To simplify the process of communicating with the sensors, we have developed a protocol stack to abstract the details of the communication protocol.

The protocol stack for RFTrax sensors is based on a layering architecture similar to the OSI model. The architecture of the driver protocol stack is shown in Figure 7.2. At the highest level, the application communicates with the sensors through API function calls to the sensor endpoint, which implement the ISensor interface. The sensor endpoint encapsulates the sensor functionalities into a user friendly object. Application requests from the upper-layer are translated into data packets according to the data communication protocol specified in [44] and are handed to the *Channel Access Coordinator* at the lower layer.

The main functions of the Channel Access Coordinator (CAC) are to (i) multiplex data packets from the upper layer, (ii) demultiplex data frames from the lower layer, and (iii) regulate the access of multiple RFTrax endpoints to the communication channel. The CAC also provides data packet buffering when the upper layer protocol is not ready to process the packets received. When a data packet is received from the upper layer, the CAC encapsulates the data packet into a frame. The frame header contains an address that uniquely identifies the sensor hardware on the RS-485 network and a checksum for error checking. The frame is sent to the lower layer when the communication channel is ready.

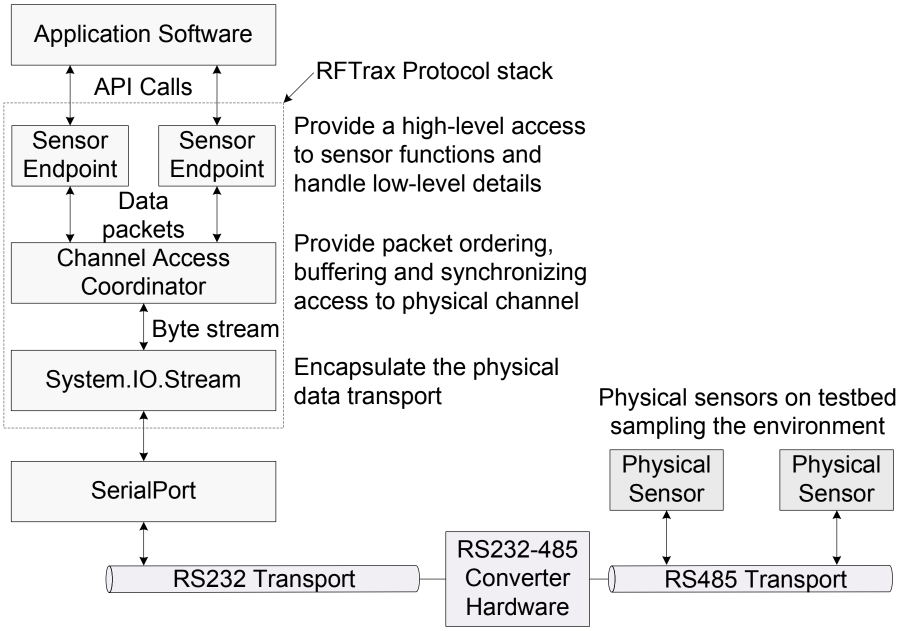


Figure 7.2. The RFTrax sensor communication protocol stack.

Below the Channel Access Coordinator is an abstraction of data stream System.IO.Stream, provided by the .NET framework. Examples of streams include RS-232 serial ports, TCP/IP sockets, inter-process communication pipes, and files. This abstraction allows RFTrax sensors to be connected to different types of networks without modifying the upper layer of the protocol stack. Depending on the actual instance of the network type, the data is then handed to appropriate operating system components for transmission. For instance, in Figure 7.2, we illustrate a connection to the sensor via a RS-232 serial port. We use a RS232-485 converter to convert the RS-232 signal to a RS-485 signal because RS-485 ports are not commonly available on desktop computers. However, if RS-485 ports are available on the host machine, the RFTrax radiation sensors can be connected directly to the host machine without the RS232-485 converter.

Communication originating from the sensors to the host machine is similar to the above except that the direction is reversed. Data frames sent from the RFTrax radiation sensors are received by the Channel Access Coordinator. When the CAC

Table 7.1
Single source localization algorithms provided in the software framework.

Class Name	Description
RoSD	Ratio of Square-Distance localization method.
DTOA	Difference of time-of-arrival localization method [11].
LogDTOA	DTOA localization method in log-space [12].
MLELocalization	Maximum Likelihood Estimation localization method [14].
RoSDITPLocalization	Iterative Pruning localization method.
RoSDMOELocalization	Mean-of-Estimator localization method with candidate estimates generated by RoSD [13].
RoSDQTLocalization	An experimental localization method that uses QT clustering algorithm.
PFMSLocalization	Hybrid particle filter with mean-shift multiple source localization method.

receives a data frame, a data checksum is performed to ensure the integrity of the data packet. Then, the target endpoint at the upper layer is identified according to the address specified in the frame header. The packet is then handed to the upper layer or buffered, if the sensor endpoint exists. Otherwise, the packet is discarded.

7.2 Algorithms and Tools Library

The software framework includes a library of tools and localization algorithms. This library enables users to build a localization system quickly as well as to evaluate the performance of the localization algorithms. Standard interfaces provided in the library enable future extensions of the library with new components and ensure interoperability between the new and existing components. Similarly, by using

Table 7.2
Analysis tools provided in the software framework.

Class Name	Description
CubicSpline	Cubic spline interpolation.
AverageVariance	Real-time computation of average and variance.
DownhillSimplex	Downhill simplex optimization algorithm.
GoldenSectionSearch	One dimensional golden-section search optimization method.
ExponentialAverage	Real-time computation of exponential average.
MovingAverage	Real-time computation of moving average.
PolynomialInterpolation	Polynomial interpolation.
Integral	Numerical integration using Romberg's method.
BisectionRootFinding	Bisection root-finding method.

standard interfaces, simulated radiation sensors can be used side by side with real radiation sensors and emulated sensors. This allows realistic large scale experiments to be performed with the software framework.

To support large scale evaluations of a radiation sensor network without using real radiation sensors, we have implemented the RFTrax emulator (see Section 7.2.1), which accurately models the behavior of a RFTrax sensor. For configuring and diagnosing the RFTrax sensor, we have implemented the RFTrax sensor console (see Section 7.2.2), which allows users to access to all functionalities of a RFTrax sensor from a graphical user interface (GUI). The software framework also includes a simulation environment modeling tool that allows users to specify a simulation environment in a XML document (see Section 7.2.3). This modeling tool is fully integrated with the localization algorithms to enable rapid evaluation of new algorithms.

Table 7.3
Probability distributions provided in the software framework.

Class Name	Description
BinomialDist	Binomial distribution.
GammaDist	Gamma distribution.
NormalDist	Normal distribution.
PoissonDist	Poisson distribution.
TDist	Student's T distribution.
BivariateNormalDist	Bivariate normal distribution.
BivariateNonparametricDist	Bivariate non-parametric distribution.
ExponentialDist	Exponential distribution.
GeometricDist	Geometric distribution.
NonparametricDist	Univariate non-parametric distribution.

In addition, we have included all the localization algorithms evaluated in Section 5 in the library. The list of algorithms included and their class names are given in Table 7.1. In addition, the library provides a number of widely used analysis tools, statistical distributions, as well as random number generators such as the Poisson random number generator for simulating sensor measurements and the Gaussian random number generator for simulating Gaussian noise. The analysis tools are optimized for real-time analysis of data while the experiment is running. All components are optimized for modern multi-core processors. Lists of components available are given in Table 7.2, 7.3, and 7.4.

7.2.1 RFTrax Emulator

To support simulation experiments, we have implemented an emulation of the RFTrax sensor, with help from an engineer at RFTrax, Inc. The radiation measurements

Table 7.4
Random number generators provided in the software framework.

Class Name	Description
BernoulliDeviate	Bernoulli deviate.
BinomialDeviate	Binomial deviate.
ExponentialDeviate	Exponential deviate.
NormalDeviate	Normal deviate.
PoissonDeviate	Poisson deviate.
UniformDeviate	Uniform deviate.
EcuyerRandom	L'Ecuyer random number generator with Bays-Durham shuffle [45].
ParallelRandom	Thread-safe random number generator.

from the emulated sensor are software programmable. The sensor emulator exposes an interface for customizing the sampling function. For instance, users can set the sampling function to a constant value, or use a Poisson random number generator to generate realistic, but synthetic, measurements. This allows maximum flexibility in simulation experiments.

Figure 7.3 shows the configuration of the protocol stack with the sensor emulator. The sensor emulator reuses many of the components from the protocol stack. Essentially, only the sensor endpoint from the protocol stack is replaced with an emulator component. The functionalities of the sensor emulator component basically complements the sensor endpoint. When using emulators, we use a memory mapped buffer to communicate data between the protocol stack and the emulator stack, as opposed to sending data to a physical I/O port. Other types of transport, such as a network loopback adapter, can be used also. However, a memory mapped buffer pro-

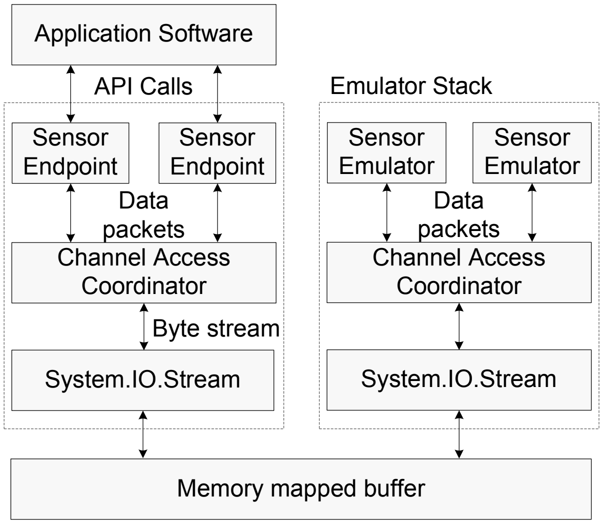
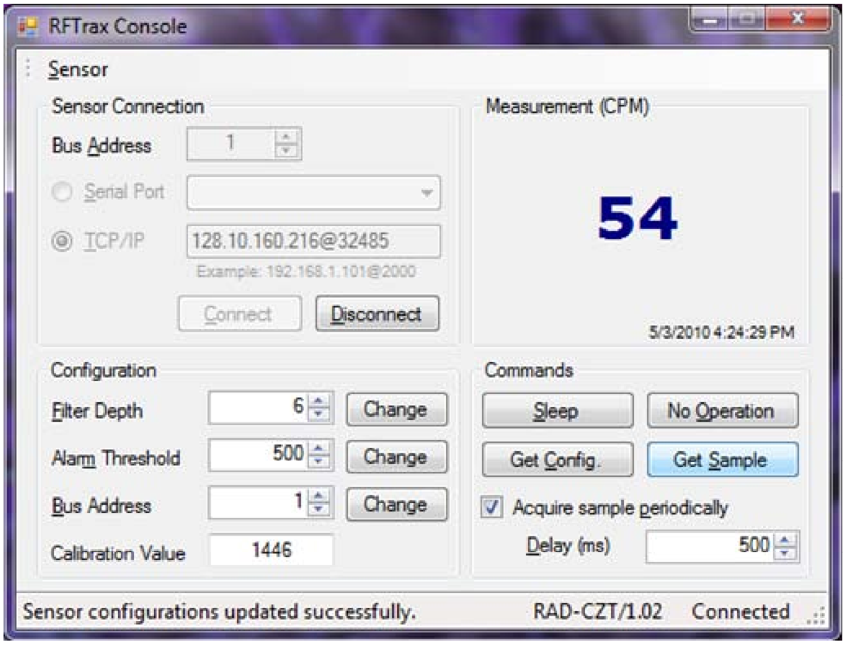


Figure 7.3. Attaching the RFTrax emulator to the protocol stack.

vides better performance because all the I/O occur in the user-space of the operating system.

7.2.2 RFTrax Sensor Console

We have developed an application for configuring and demonstrating the functionalities of RFTrax sensors. The application, named RFTrax Console, is built utilizing the protocol stack. It provides a graphical user interface (GUI) for users to interact with the RFTrax sensors as shown in Figure 7.4. The application supports connections via RS-232 serial ports or TCP/IP networks. Once a connection has been established, the application acquires all the configuration information from the sensor. This includes the filter depth, alarm threshold, sensor bus address, and the internal calibration value. Users can change these sensor configurations as well as issue commands, such as putting the sensor into a power saving mode, directly from the GUI. The application can be used to discover RFTrax sensors connected to the network as well, by sending `No Operation` commands to all possible addresses and listening for acknowledgments from the sensors. This application has been used to



Bus Address 1

○ Serial Port

● TCP/IP 128.10.160.216@32485

54

5/3/2010 4:24:29 PM

Filter Depth 6

Alarm Threshold 500

Bus Address 1

Calibration Value 1446

☑ Acquire sample periodically

Delay (ms) 500

Sensor configurations updated successfully. RAD-CZT/1.02 Connected

Figure 7.4. Graphical user interface of the RFTrax sensor console.

setup the testbed for the experiments presented in Section 5.3 and to help diagnose various problems we encountered during our experiments.

7.2.3 Simulation Environment

The software framework provides a user friendly way to specify simulation environments. A simulation environment specifies all the properties in a virtual surveillance area. This includes the sensor attributes (such as location, background radiation, and efficiency), radiation sources (such as location, strength, and movement model), and obstacles (such as shape, size, placement, and attenuation coefficient). Instead of scripting the simulation environment using an ad-hoc approach, we specify simulation environments in XML files and provide the infrastructure to load/save the simulation environment to the application.

The XML format is both machine friendly and user friendly. The data in the XML file is stored as plain text. Hence, users may edit the file with any text editor or

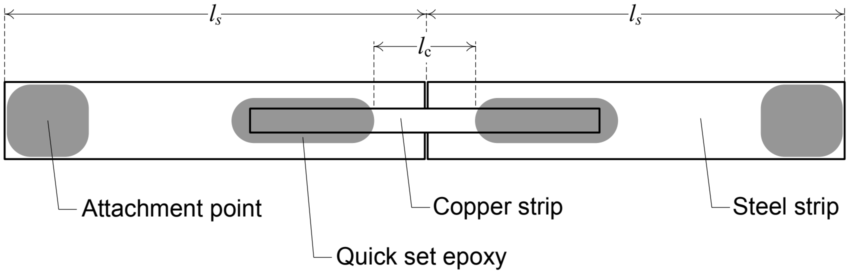


Figure 7.5. The design of a crack detecting gage.

sophisticated authoring tool. For instance, users may create a simulation environment in Google Earth and run the simulation on the realistic environment model. In addition, the XML file is structured so that it can be efficiently parsed by an XML parser. Many programming frameworks such as .NET framework and Java$^{\text{TM}}$, provide extensive support for parsing XML files. Moreover, the format is forward compatible, where future extensions to the data format are backward compatible.

7.3 Further Application to Structural Health Monitoring

The software framework presented has been used extensively for evaluating the localization algorithms presented in this dissertation. To demonstrate its versatility, we have extended this software framework to another domain, to develop the Rapid Structural Assessment Network (RSAN) [46]. The RSAN is a structural health monitoring system that uses a wireless sensor network and custom-built sensors, called gages, to unequivocally detect cracks in critical structural elements.

In monitoring the structural integrity of a building, we seek to detect significant cracks in critical structural elements, where crack widths exceed a predetermined threshold. When significant cracks are detected, the safety of the structure can be concluded directly.

7.3.1 Sensor Design

To detect these significant cracks, we embed low-cost sensors at strategic locations in the building structure. Each low-cost sensor, referred to as a *gage*, consists of a short and narrow copper strip and two steel strips. The design of these gages is shown in Figure 7.5. The copper strip being used to build these gages is 0.002 inch thick and 0.125 inch wide, and the steel strip is 0.005 inch thick and 0.375 inch wide. Each end of the copper strip is adhered to the steel strip. The total length of the gage l_s determines the area that can be monitored by a single gage. The clear distance between the areas where the copper strip is adhered to the steel strips is referred to as *gage length*, l_c.

The gage is attached to a structural element at each of the two attachment points. When cracks are formed on the structural element between the two attachment points of the gage, the stress on the copper strip increases. The copper strip breaks when the total deformation exceeds a predetermined value. The predetermined value is determined via experiments and is correlated to the length of the copper strip.

Essentially, the parameters l_s and l_c determine the *detection range* and *sensitivity* of the gage. Notice that if two cracks of widths l_1 and l_2, respectively, are formed between the attachment points, the gage will break when $l_1 + l_2$ exceeds the threshold.

7.3.2 Data Acquisition and Communication

To detect breakage of the copper strip, the gage is connected to a data acquisition device. Data collected by the device is transmitted to a remote control center via a wireless network. We discuss three alternative approaches to acquiring data: Passive RFID, the MICA2 motes with MDA300 data acquisition board from Crossbow Technology, and the higher precision ADAM-6017 data acquisition module.

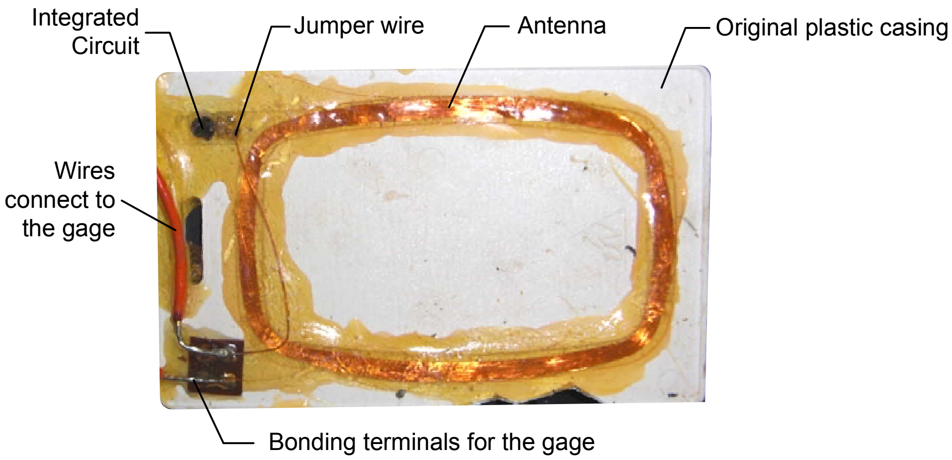


Figure 7.6. Embedding the gage in a passive RFID tag.

Passive RFID

Gages can be embedded into a passive RFID tag, so that the status of the gages can be queried by a nearby RFID reader. A RFID tag consists of an antenna loop and a small microchip containing a small amount of information such as a unique tag identifier. When a RFID reader excites the RFID tag with a wireless signal, the RFID tag reflects the information stored in the microchip back to the RFID reader. Because the RFID tag only reflects energy from the RFID reader, the RFID tag does not require power to operate. RFID readers, however, are usually powered. In our experiments, the RFID readers draw power from the USB port.

We embed the gage as part of the antenna loop as shown in Figure 7.6. When the gage is not broken, the gage completes the antenna loop. Therefore, the functioning RFID tag will respond when queried by an RFID reader. However, when the gage is broken, the RFID reader will not receive a response from the RFID tag, thereby allowing the underlying cracks to be detected.

In our implementation, we use RFID readers/tags manufactured by Phidgets Inc. [47]. The Phidgets RFID reader returns the unique tag identifier of the RFID tag whenever a tag comes into or goes out of range. The unique identifier of the RFID tag can be readily mapped to the deployment location of the tag, as the deployment locations are known when the monitoring system is set up.

Under normal operating conditions, the RFID reader continually detects the presence of the tag until the gage is broken. The RFID reader reports that the RFID tag has gone out of range when the gage is broken, signaling significant cracks have formed. Two limitations of the Phidgets RFID readers used in our deployment are that they have a small reading range (less than 30 cm), and they cannot selectively communicate with multiple RFID tags that are within range. Moreover, multiple RFID readers within reading range of each other can interfere. Therefore, we use one RFID reader per RFID tag to simplify the experiments. However, we note that this is a limitation of our current experimental hardware, and is not an inherent limitation of passive RFID technologies.

The Phidgets RFID reader has a USB interface for communication. We use a Keyspan US-4A USB server to connect multiple RFID readers to the network. The Keyspan US-4A USB server allows up to four USB devices to be used across an Ethernet network. By connecting the Keyspan USB server to a Linksys 802.11g router, which in turn connects to the Internet, we can monitor the RFID tags from a remote control center. This system configuration is shown on the left of Figure 7.7.

MICA2 Motes with MDA300 Data Acquisition Board

Crossbow MICA2 motes can be used to query the status of gages by connecting a MDA300 data acquisition board [43] to a MICA2 processor/radio module. The MDA300 data acquisition board has eleven channels of single-ended 0.0–2.5 V voltage ADC inputs and six digital I/O channels. It also provides a 2.5 V sensor excitation output that enables the gages to be powered by the data acquisition board instead of an external power source. In our experiments, the motes are powered by a pair of AA batteries. This provides power to the mote, data acquisition board, as well as the gages.

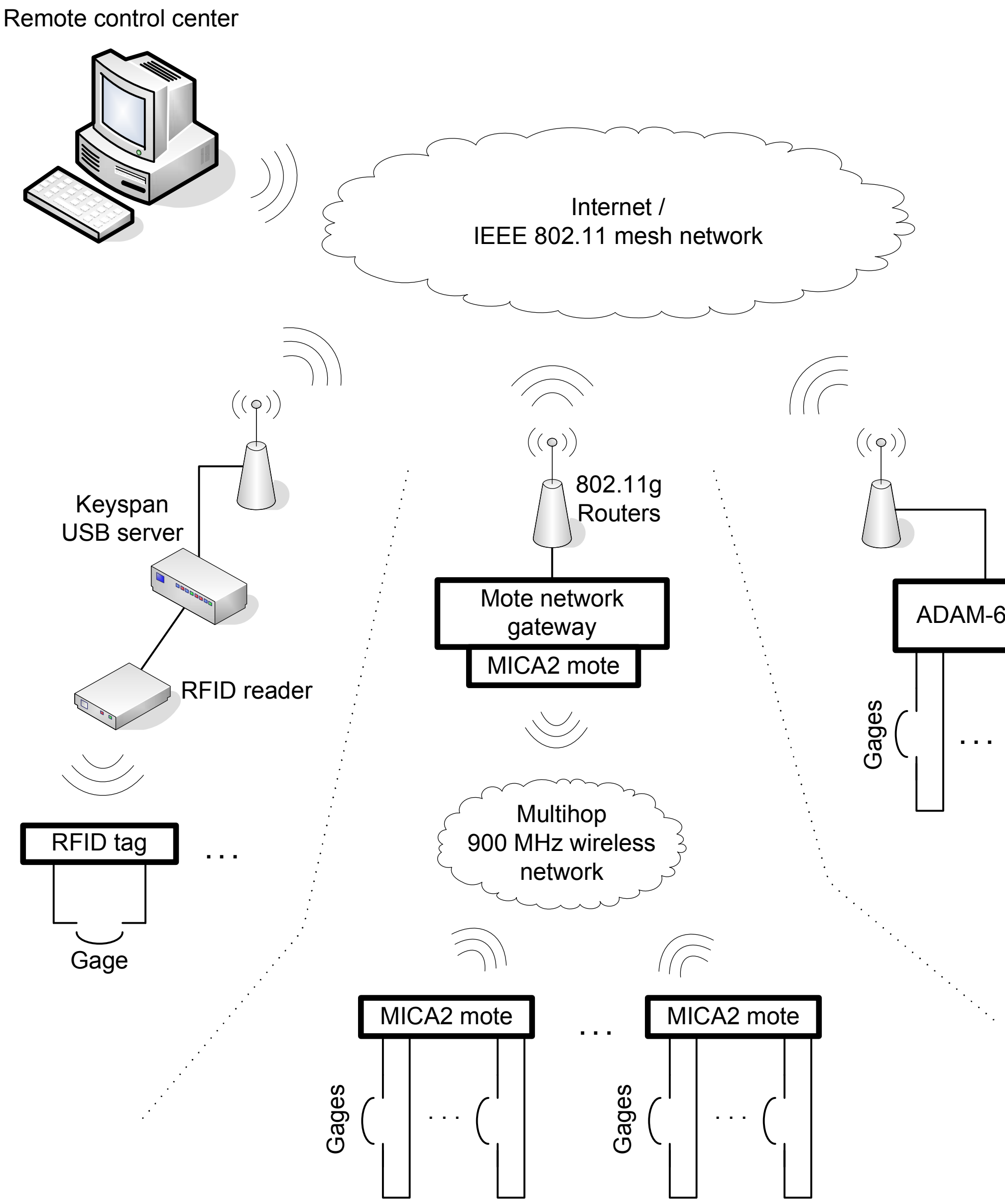


Figure 7.7. Three configurations of the remote monitoring network.

In our deployment, we connect a number of gages to both the analog and digital channels of a MDA300 data acquisition board. The electrical connection is shown in Figure 7.8. Terminal A and B in the circuit are connected to the input channel and ground, respectively, on the MDA300 data acquisition board. The direct current (DC) power source V_s in the circuit is connected to the 2.5 V sensor excitation output

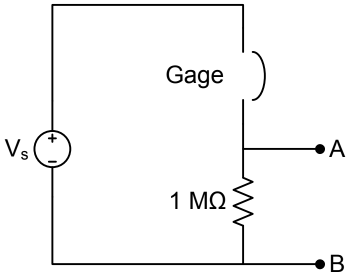


Figure 7.8. The wiring of a gage to the measurement hardware.

(labeled E2.5 on the MDA300 data acquisition board). The electrical circuit connects a power source, a gage, and a resistor in serial. In this configuration, the voltage between terminal A and B is equal to V_s when the gage is not broken. When the gage is broken, the circuit is broken and therefore the voltage between terminal A and B is 0 V. In an intermediate state in which the strip is stretched but not yet broken, the voltage measurement corresponds to the resistance in the gage circuit.

The MICA2 mote collects data from the gages connected to the MDA300 data acquisition board and sends that data over a 900 MHz wireless network. In our deployment, we use a number of other motes as wireless relay nodes to relay the sensor data to the Ethernet gateway (Crossbow MIB600) and then to the Internet. The Ethernet gateway forwards data from the mote system to the Internet-accessible remote control center. This setup is shown in the middle of Figure 7.7.

ADAM-6017 High Precision Data Acquisition Module

We have also experimented with an ADAM-6017 data acquisition module. The ADAM-6017 module connects to a set of gages similar to the mote system and serves a similar purpose. Compared with the MDA300 module, the ADAM-6017 module provides higher precision sensing and has a built-in Ethernet port. We connect a Linksys 802.11g router to the ADAM-6017 module to support wireless data communication, as shown in Figure 7.7.

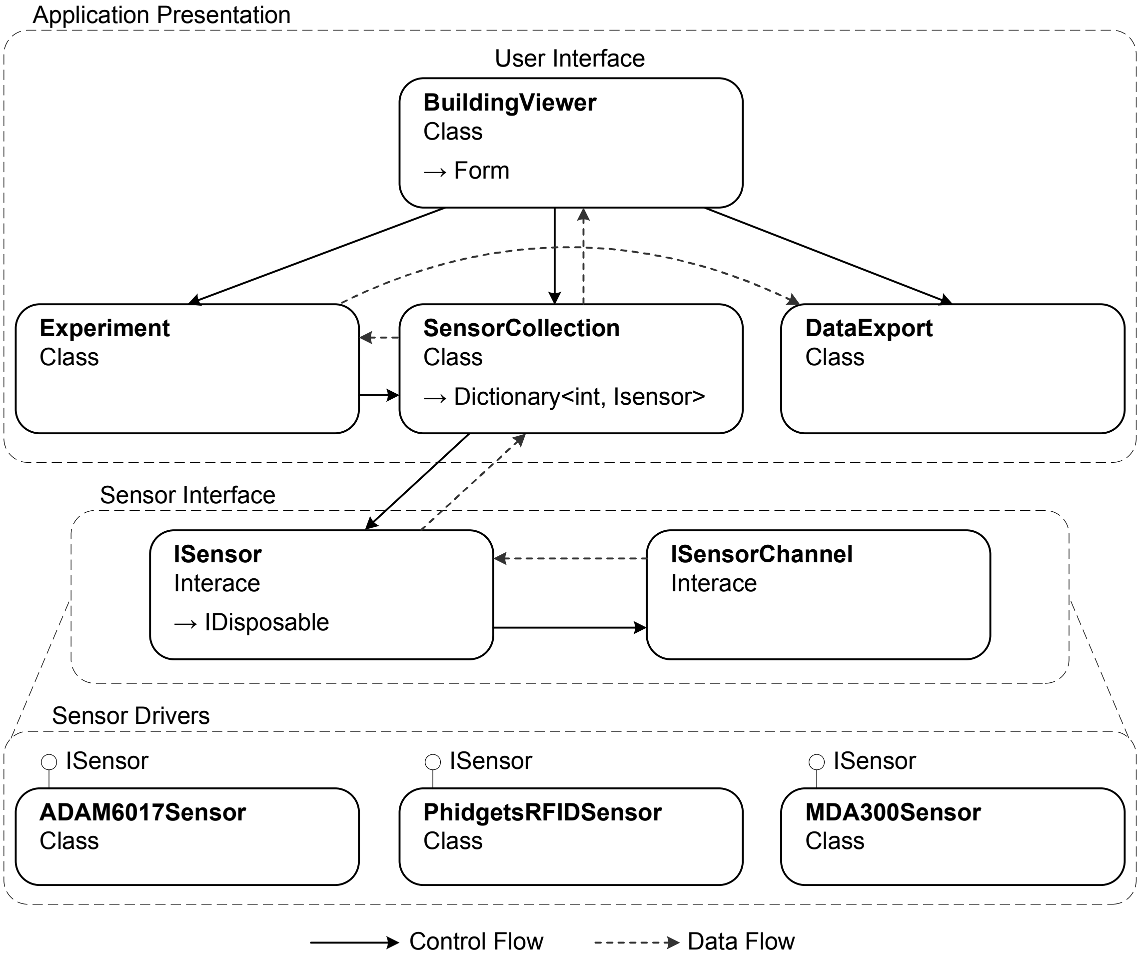


Figure 7.9. The architecture of RSAN software.

7.3.3 Software Architecture

We have developed a GUI application software to remotely monitor the gages installed on a building structure. The software is developed using the sensor driver interface and tools in the software framework presented in Section 7.1 and 7.2.

The core of the software contains a collection of software components, for essential sensor monitoring tasks including device and communication control, data collection, selection, logging, and presentation. A sensor network deployment can be specified by configuring the components as needed.

The architecture of the software is shown in Figure 7.9. Directed lines show the data and control flows between the software components. The software can be divided into three layers. The highest layer, shown at the top of the figure, is the *application presentation* layer; the middle layer is the *sensor interface*; the lowest layer, shown at the bottom, is the device dependent *sensor driver* layer.

Application Presentation Layer

The BuildingViewer component in the application layer directs the overall sensor network deployment. It interacts with the end user by accepting user commands and displaying the current status of the monitored building in a graphical user interface. BuildingViewer interacts with three components of class Experiment, SensorCollection, and DataExport. The Experiment component manages the schedule of collecting and logging data from the underlying sensors throughout the experiment. It controls tasks such as receiving data and events from the sensors, timestamping the received data and events, and setting experiment parameters such as the time of the experiment and the size of the receiving data buffers.

The Experiment component uses the SensorCollection component for reading the sensor data. In addition, SensorCollection manages the read and write of the sensor configuration file described in Section 7.3.4. The DataExport component is responsible for two important tasks: (1) Filtering data received by the Experiment component to remove data that are not needed in the user report, and (2) formatting the sensor data into a human readable output format. By extending the DataExport class, the data can be exported in various forms, including spreadsheets, charts, text, or a SQL database.

Sensor Drivers

The application communicates with the sensor devices via the sensor driver interface layer. The interfaces in this layer are provided by the software framework described in Section 7.1. All sensor drivers implement the ISensor interface and this layer enables interoperability among different types of sensor hardware.

Device dependent sensor drivers are implemented in the sensor driver layer. The sensor driver layer realizes the low level communication protocols needed for the sensor hardware to perform the tasks defined by the generic interfaces. We provide synchronization mechanisms where necessary to allow safe multi-threading operations. Thread safety is important because the application layer may run multiple threads that attempt to gain access to the sensor resources concurrently.

The characteristics of our sensor hardware that impact the sensing application are summarized as follows:

- **Phidgets RFID reader.** After initialization, the RFID reader monitors the RFID signal continuously and sends a Tag event, along with the tag data, to the driver when an RFID tag comes within range. Similarly, a TagLost event is sent to the driver when a previously in range RFID tag is no longer present. The driver captures these events from the RFID reader, and delivers the events to the higher layer software which has subscribed to the events. In our application, the Experiment component subscribes to the Tag and TagLost events. Note that because the RFID reader updates the status of a tag only when a tag is detected or lost, our software will not receive continual data updates from the sensor. Instead, the status will be updated only when the gage is broken due to cracks. Hence, the data rate of the RFID based sensor is extremely low in our deployment.

- **MICA2 motes with a MDA300 data acquisition board.** The driver for the MDA300 module handles communication between the control center and the mote network gateway (see Section 7.3.2) via a TCP connection. In normal operations, the driver receives heartbeat and data messages from the MICA2 motes accessible through the mote network gateway. The RSAN software infers the status of each mote using the heartbeat and data messages received. In particular, if no heartbeat message is received within the past ten seconds, the driver concludes that the MICA2 mote (radio/processor module) is either offline or malfunctioning. If no data from any data channel is received within the past ten seconds, the driver concludes that the data acquisition module is not connected or is malfunctioning. Otherwise, the driver parses the raw data obtained from the mote to remove any escape characters and demultiplexes the data to the correct sensing channels. The demultiplexed data are then sent up to subscribing upper layer software.

- **ADAM-6017.** The ADAM-6017 driver communicates with a remote ADAM-6017 data acquisition module over a MODBUS-TCP protocol stack. MODBUS is the *de-facto* open standard for communication with a wide range of industrial electronic devices, including the ADAM-6017 module. Our driver implements a subset of this protocol to configure and read data from the remote ADAM-6017 module. For instance, the driver periodically sends a MODBUS query message, at a user specified interval, to query all eight data channels of the ADAM-6017 module. The ADAM-6017 responds with a group of 16-bit unsigned integers, each of which is either a voltage or current reading from a sensing channel (see Section 7.3.2).

7.3.4 Software and Experiment Configuration

As the RSAN monitoring software is intended to be used for a wide range of applications and experiment scenarios, it is important that we provide a method for incorporating off-the-shelf sensor and communication devices from diverse manufacturers into a working system. In addition, a uniform interface should be supported to configure the different devices. We extend the simulation environment specification file described in 7.2.3 to incorporate physical sensor information. Therefore, the RSAN software basically contains a virtual view of the actual deployment of the sensor network. This demonstrates the extensibility of the software framework to other do-

```
<?xml version="1.0"?>
<sensorconfig>
  <sensor id="200" type="phidgetRFID">
    <description>Phidgets RFID Reader (17959)
    </description>
    <hardwareid>4627</hardwareid>
  </sensor>
  <sensor id="100" type="MDA300"><!-- omitted -->
  </sensor>
  <sensor id="300" type="ADAM6017">
    <description>ADAM6017 Data Acquisition Module
    </description>
    <port>TCP:192.168.1.160:502</port>
    <pollinterval>1000</pollinterval>
    <channel id="0">
      <description>ADAM6017 Channel #0</description>
      <range min="-5" max="5"/>
    </channel>
    <channel id="1">
      <description>ADAM6017 Channel #1</description>
      <range min="-5" max="5"/>
    </channel>
  </sensor>
</sensorconfig>
```

Figure 7.10. An example of a sensor configuration file.

mains and applications. An example of the configuration file is shown in Figure 7.10. In this example, we configure a Phidgets RFID sensor, a Crossbow MICA2 mote with a MDA300 data acquisition board, and an ADAM-6017 module. The root element of the XML document is the `sensorconfig` element, which contains a `sensor` element for each specific sensor device. The `sensor` element specifies the configurations of the sensor hardware. The `type` attribute identifies the type of the sensor, so that the appropriate drivers can be loaded during the initialization. There may be multiple instances of the same type of sensor, in which case the `id` attribute of the `sensor` element gives a unique ID for each of the instances. The `sensor` element, as well as its child elements, may contain information that is specific to the sensor hardware and, as such, is passed to the appropriate driver for interpretation.

As an example of driver dependent configurations, the `port` element of the ADAM-6017 module in the configuration file identifies the TCP network port where the sensor hardware is connected. Each `channel` element configures a data channel of the ADAM-6017 module, where a gage is connected. In this configuration, a data channel is named by the `channel` element, and is customized independently from one another. Two ADAM-6017 channels are shown in this example. The channels are uniquely identified by the `id` attribute of the `channel` element. As discussed in Section 7.3.3, the ADAM-6017 module returns a 16-bit unsigned integer when a channel is queried. The interpretation of this value depends on the configuration in the ADAM-6017 firmware. For instance, when the channel is configured to measure ±5 V, the integer value 0 denotes −5 V and the value 0xFFFF denotes +5 V. On the other hand, when the channel is configured to measure 0–500 mA current, an integer value 0 denotes 0 mA and the value 0xFFFF denotes 500 mA. The `range` element configures this mapping with the `min` and `max` attributes in the configuration file. All these configurations are device dependent and are defined by the sensor driver.

The encapsulation of the hardware details in specific XML elements, as shown in the above example, is highly desirable. This encapsulation allows the system to support other types of devices that are not shown in this example.

7.3.5 Conclusion

In this Rapid Structural Assessment Network, we have demonstrated the extensibility of the software framework that we have developed. The sensor driver interface we have designed allows simulated, emulated, as well as actual sensors to be used in the application by implementing small modules that contain device dependent code for the sensors. Environment model, simulation scenario, and application configurations can be specified in an user-friendly XML file. The software framework thus allows users to evaluate algorithms and systems efficiently, both in simulations as well as in actual deployments.

8 RELATED WORKS

The detection, localization, and tracking of signal sources (or *targets*) of various kinds, including radiation sources, chemical plumes, wideband radio signals, and acoustic signals have been well studied. In this chapter, we review related works in both single source detection and localization (Section 8.1), and multiple source detection and localization (Section 8.2).

8.1 Single Source Detection and Localization

Detection problems have been studied extensively over the past several decades in the areas of signal detection [48–51], classification [52, 53], estimation [54, 55], identification [56], and tracking [39], under various formulations. In general, the area of detecting various radiation sources using individual sensors has been well established in terms of both detection devices and detection methods [19, 57], most of which are dedicated to single or co-located sensor systems. Traditionally, the presence of a source is first confirmed based on measurements using a detection or classification rule. Then, the measurements are used to estimate the parameters of the source using an underlying linear or non-linear model with random noise components. For example, one of the most studied formulations, the Kalman filter [36], assumes a linear process model with additive independent Gaussian noise.

The detection and estimation of radiation sources of various kinds have been well studied, particularly using a single sensor [19, 58]. The detection of radiation sources amidst background radiation has been studied using the sequential probability ratio test (SPRT) for various scenarios such as long-term portal monitoring [57, 59, 60]. However, the existing works using the SPRT do not address the source localization problem.

Recently, using a network of sensors for detecting radiation sources has been proposed. In particular, a linear arrangement of detectors has been considered in [61–63], and an analysis of sensor network solutions has been carried out for a source moving in a linear path [64]. The detection of a point radiation source using a sensor network is addressed in [65], wherein sensor measurements are combined using a copula function that captures the sensor correlations.

Localization of a signal source has been extensively studied as well [66–71]. Localization has a wide range of important applications including location-aware monitoring [72], network routing [73,74], context-aware services [75] and security [76]. A well known class of localization is based on the time-of-arrival (TOA) of a signal. With TOA, the sensors record the time at which a signal from the source is received. The distance between the source and the sensor is then inferred by the time difference between when the signal is sent and when the signal is received. Subsequently, the location of the source is computed using the measurements from multiple sensors. This technique has been used in several indoor localization applications [77–79]. However, the TOA method requires strict time synchronization between the source and the sensors in order to achieve high localization accuracy. In some applications such as localizing hostile signal sources, time synchronization is difficult to achieve.

Another class of localization algorithms is based on the time difference of arrival (TDOA) [80,81]. TDOA (sometimes referred to as DTOA in the literature) measures the differences in time instants at which a signal emitted by the source reaches other sensors. Hence, although the exact distance between the source and a sensor is unknown, the differences in distance between the source and the sensors can be inferred from the time differences. TDOA has been used for the localization of plumes (e.g., radioactive, biological, chemical) assuming idealized product-form plumes and an exponential decay function, i.e., a source of strength A will register a signal strength of Ae^{-r} at a distance r away [82]. When measurement and computational errors are considered, a geometric approach has been proposed [11,83] to solve the TDOA problem with increased robustness. Specifically, their approach reduces the numerical instabilities when exact solutions of a system of quadratic equations are perturbed by noise.

When angle of arrival (AOA) measurements are available, a signal source can be located using a triangulation method [84–86]. In this approach, the sensors measure the direction of the incoming signal. For RF signals, this can be achieved using an array of directional antennas with each antenna pointing at a different direction. Using the directional information collected from multiple sensors, the location of the signal source is calculated. The AOA based technique can also combine with other techniques such as TDOA to improve the localization accuracy as proposed by Yang *et al.* [87].

Instead of inferring the distance between the source and sensors in the time domain, the received signal strength (RSS) can be used to infer the distance as well [88]. Signals such as acoustic wave and electromagnetic radiation attenuate when propagating through space. Assuming the signal strength at the source is known, measuring the attenuated signal reveals the distance that the signal has traveled. By measuring the RSS at multiple locations, the source location can be estimated. The advantage of a RSS based technique is that the measurement hardware is often less complex, and therefore less expensive, than the measurement hardware for time-based measurements. However, a RSS based method requires accurate modeling of the signal attenuation. Accurate modeling of the signal attenuation can be difficult especially in an indoor environment because obstacles, which may obstruct the signal or generate multi-path interference, are difficult to accurately incorporate into the model [89].

To overcome the difficulties in modeling, Azizyan *et al.* [90] proposes a fingerprinting solution. The proposed solution is divided into two steps—an offline (training) step, and an online (matching) step. First, in the offline step, the signal strengths at all locations in the surveillance area are measured. These measurements and their corresponding locations are stored into a database. Then, for monitoring in the online step, a measured signal is matched with the signals previously stored in the database. In this case, modeling the signal attenuation is not needed and is replaced by matching the signal strengths between the measured signal and the sample signals stored in the database.

Recently, several probabilistic localization methods have been proposed. In particular, Gunatilaka *et al.* [91] proposes a method for localizing a chemical plume using data taken periodically from a network of chemical sensors. The proposed method assumes that the chemical is released on the ground continuously. The localization algorithm utilizes a sequential Bayesian framework, and Monte-carlo integration to continuously compute the most likely position of the chemical plume. A similar approach is proposed by Robins *et al.* [92], which uses the Markov Chain Monte Carlo (MCMC) method to efficiently localize the movement of plumes in account for complex atmospheric dispersion.

In localizing a radiation source, the source strength is often used as the measurement to infer the distance from the radiation source. Localization of a single radiation source is typically solved by using the least square method [93] or the Maximum Likelihood Estimation (MLE) method [14]. Given a sensing model and sensor measurements, the least square method searches for the parameters of the radiation source that have the minimum error, whereas the MLE method searches for the parameters that have the highest probability in fitting to the model.

Rao *et al.* [12] proposes the log-space TDOA algorithm (LgDTOA) that adapts the TDOA algorithm [11] to log-space and uses the log-differences in source strength measurements from three sensors to infer the source location. To tolerate the large noise component in radiation strength measurements, a network of N sensors may be used. Rao *et al.* [13] proposes the Mean-of-Estimator (MoE) method which first uses the LgDTOA algorithm to localize the radiation source by all subsets of three sensors and then linearly combines the location estimates to produce the fused estimate. A similar fusion approach, where the localization results from all sensors are fused using the Iterative Pruning (ITP) algorithm, is presented in [22].

Vilim *et al.* [94] proposes a system for monitoring the transport of radiation sources with consideration of arbitrary obstacles in the surveillance area. The proposed system uses the MLE method, and measurements from a network of sensors, to localize radiation sources. To incorporate arbitrary obstacles in the surveillance area, an offline phase for fingerprinting the obstacles is required. In addition to estimating the location of radiation sources, the proposed system also computes the confidence interval in which the sources are located.

8.2 Multiple Source Detection and Localization

Localization of multiple signal sources is typically solved by using the Maximum Likelihood Estimation (MLE) method [31]. The MLE formulation is similar to the formulation for single source localization. However, the number of source parameters required to estimate increases in proportion to the number of signal sources. Therefore, in addition to estimating the source parameters, the number of signal sources needs to be estimated as well. Estimating the number of signal sources is often referred to as the model order selection problem because the number of signal sources affects the number of parameters in the model [32]. For instance, Ding *et al.* [95] uses multi-modal sensor data to localize and track signal sources by modeling the sources with a Gaussian mixture model and then using Bayesian Information Criterion (BIC), a model selection algorithm, to estimate the number of sources. Then, an expectation maximization (EM) algorithm and the mean-shift algorithm are used to localize and track the sources.

Morelande *et al.* [96] localizes sensor nodes in a wireless sensor network using a two-step approach. First, the distances among the sensor nodes are measured using a radio interferometric positioning system (RIPS), which uses the phase difference between signals that are simultaneously transmitted at a pair of nodes and received at another pair of nodes, to compute the sum of range differences between the four nodes. Then, the distance measurements are used as the input to a particle filter for estimating the location of the sensor nodes. In their formulation, the number of sensor nodes to be localized is known and therefore does not need to be estimated.

Xia *et al.* [97] considers localizing multiple targets, which have unknown transmitting signals, in multiple-input multiple-output (MIMO) radar systems. In MIMO radar systems, the transmitters and receivers have multiple antennas. The proposed solution assumes that the number of targets is known *a priori* and is less than the number of receiving antennas. In addition, the solution assumes that the transmitting antennas are sparsely located whereas the receiving antennas are closely placed. As a result, the same signal is received by multiple receivers' antenna and thus the direction to the transmitter can be estimated. Their paper proposes and compares several formulations of the MLE method to estimate the direction to the targets. Then, the mixing of the signals are estimated using blind signal processing (BSS).

For localizing radiation sources, similar solutions have been proposed. For instance, Morelande *et al.* [30] estimates the number of radiation sources using a model selection algorithm, and then computes the source parameters using the MLE method. As reported in [30], the accuracy of the model selection algorithm degrades when the number of sources increases. In addition, using the EM or MLE method to estimate the source parameters is computationally intensive when the model has many parameters. In fact, each additional radiation source increases the number of parameters in the model by three and the efficiency of the algorithms does not scale beyond four sources.

In [98], the authors propose to solve the multiple source localization problem by using a convex optimization method that assumes the radiation sources are located in a grid over a region of interest. The proposed method discretizes the search space and attempts to detect and localize a source in each discretized location in the search space. Depending on the granularity of the discretization, the algorithm can take up to 209 seconds to run on a Dell desktop computer with dual-core Intel® Pentium® CPU at 2.40 GHz and 4 GB RAM, as reported in the paper. This prohibits the algorithm from scaling to a large network of sensors.

The use of a particle filter to localize multiple radiation sources has been proposed. For instance, Ristic *et al.* [33] formulates the localization problem, both estimating the number of sources and the source parameters, as a sequential Bayesian estimation problem and solves the problem using a particle filter. In the solution, mobile sensors are tasked to measure the radiation intensity at various locations. These measurements are then used in the particle filter to refine the parameter estimates. To determine the next location for sensing, their formulation computes the anticipated information gain at future positions and selects the best location for the sensing. Similar solutions have been proposed [29,99]. The proposed solutions use joint multitarget probability density as a likelihood measure of the source parameters in the particle filter formulation.

Localizing targets in environments with obstacles has also been considered. In [100], the algorithm solves the localization problem by assuming that the placements of obstacles are known to the algorithm. The algorithm discretizes the search space and finds the probability p of a target being located inside a cell. Because the location is known, p can be calculated easily with consideration for the obstacles. Nonetheless, the complexity of the algorithm is exponential with regard to the number of sensors. In [101], the proposed method uses mobile robots to search for a point radiation source. In the paper, the authors focus on planning the robot movements such that each move maximizes the information gain in the search. The detection and localization of the source during the search is performed using a particle filter. Other efforts have mainly studied how obstacles affect sensor communication and motion planning rather than how they affect the source localization algorithms. For instance, the problem of tracking a target using mobile robots, while detecting and avoiding obstacles that may block the target signals, has been studied in [102,103].

9 SUMMARY

The ability to detect and locate radiation sources quickly is very important for protecting people against stealthy attacks from RDDs. In this dissertation, we have studied efficient methods to detect and locate radiation sources, especially shielded radiation sources that appear to be similar to background radiation, using a network of radiation sensors.

Radiation intensity measurements are inherently noisy. Using a minimum of three sensors, we have presented the Ratio of Square-Distance method (RoSD) that computes the source estimate in $O(1)$ time. Our evaluation has shown that the RoSD algorithm achieves significantly lower localization error and also runs faster compared with the existing log-space DTOA algorithm (LgDTOA).

When more than three sensors are available, we have presented the Iterative Pruning (ITP) algorithm to localize a radiation source. We have shown that the localization accuracy of the ITP algorithm improves with the number of sensors. Our evaluation results have shown that the ITP algorithm outperforms the Mean-of-Estimator (MoE) algorithm, and has an accuracy comparable to the Maximum Likelihood Estimation (MLE) algorithm. In addition, the ITP algorithm generally has a faster execution time compared with the MLE algorithm because the ITP algorithm does not require an expensive multidimensional optimization over a large parameter space.

To localize multiple radiation sources in a sensor network, we have presented a hybrid formulation of a particle filter with a mean-shift technique that works well even with the presence of obstacles. The proposed method is efficient in handling multiple radiation sources and is scalable to a large sensor network. Unlike existing algorithms, the proposed algorithm requires neither the number of sources to be known in advance nor an expensive statistical method to estimate the number of sources, and does not require detailed specifications of obstacles in the environments.

Thus, our algorithm can enable existing sensor networks to localize multiple sources without explicitly modeling all the radiation sources and obstacles, a capability not currently available to such networks. Our simulation results show that the proposed algorithm can accurately localize radiation sources with low false positive and low false negative rates.

To confirm the presence of the estimated source, we have presented the Localization Enhanced Detection (LED) method. This method incorporates the source estimate into the detection process, instead of localizing the source after a radiation source has been detected. We have shown that this approach outperforms existing detection methods with lower false positive and false negative rates, and with a faster detection time.

We have built an experimental testbed to evaluate the proposed methods in this dissertation with a 0.911 μCi CS-137 radiation source. In addition, we have developed a versatile software framework to operate the testbed hardware. The software framework is designed in such a way that the components are ready to be deployed in an actual system implementation. We have demonstrated the versatility of the software framework by using it to build the Rapid Structural Assessment Network for structural health monitoring.

LIST OF REFERENCES

LIST OF REFERENCES

[1] Rama Lakshmi. 61 killed, 350 hurt in bombings across India's Assam state. *The Washington Post*, October 2008.

[2] Introduction to ionizing radiation. `http://www.osha.gov/SLTC/radiationionizing/introtoionizing/ionizinghandout.html`. [Accessed: June 13, 2010].

[3] Radiation detection at US ports and borders. `http://www.allbusiness.com/services/health-services/3963418-1.html`, October 2006. [Accessed: May 9, 2010].

[4] Thermo Fisher Scientific successfully field tests mobile radiation detection system in Washington D.C. `http://www.domesticpreparedness.com/Industry/Industry_Updates/Thermo_Fisher_Scientific_Successfully_Field-Tests_Mobile_Radiation_Detection_System_in_Washington_D.C./`, September 2009. [Accessed: May 4, 2010].

[5] Marcus Foth. *Handbook of Research on Urban Informatics: The Practice and Promise of the Real-Time City*, chapter XXVIII. Information Science Reference, IGI Global, Hershey, PA, 2009.

[6] A. Barzilov, P. Womble, I. Novikov, J. Paschal, J. Board, and K. Moss. Network of wireless gamma ray sensors for radiological detection and identification. In *Proceedings of SPIE*, volume 6540, 2007.

[7] Tarek Abdelzaher, Yaw Anokwa, Peter Boda, Jeff Burke, Deborah Estrin, Leonidas Guibas, Aman Kansal, Samuel Madden, and Jim Reich. Mobiscopes for human spaces. *IEEE Pervasive Computing*, 6(2):20–29, 2007.

[8] States use LLNL-developed radiation detection advance for highway monitors, patrolling roads. `https://publicaffairs.llnl.gov/news/news_releases/2008/NR-08-10-01.html`, October 2008. [Accessed: May 5, 2010].

[9] Adaptable radiation area monitor (ARAM) radtruck. `https://www.rkb.us/contentdetail.cfm?content_id=148115`, May 2009. [Accessed: May 9, 2010].

[10] Zong Da Chen, H. T. Kung, and Dario Vlah. Ad hoc relay wireless networks over moving vehicles on highways. In *Proceedings of the 2nd ACM International Symposium on Mobile Ad Hoc Networking & Computing (MobiHOC)*, pages 247–250, 2001.

[11] Xiaochun Xu, Nageswara S. V. Rao, and Sartaj Sahni. A computational geometry method for localization using differences of distances. *ACM Transactions on Sensor Networks (TOSN)*, 6(2):1–25, 2010.

[12] Nageswara S. V. Rao, Mallikarjun Shankar, Jren-Chit Chin, David K. Y. Yau, Srinivasagopalan Srivathsan, S. Sitharama Iyengar, Yong Yang, and Jennifer C. Hou. Identification of low-level point radiation sources using a sensor network. In *Proceedings of the 7th International Conference on Information Processing in Sensor Networks (IPSN)*, pages 493–504, 2008.

[13] Nageswara S. V. Rao, Mallikarjun Shankar, Jren-Chit Chin, David K. Y. Yau, Yong Yang, Jennifer C. Hou, Xiaochun Xu, and Sartaj Sahni. Localization under random measurements with application to radiation sources. In *Proceedings of the 11th International Conference on Information Fusion (FUSION)*, June 2008.

[14] A. Gunatilaka, B. Ristic, and R. Gailis. On localisation of a radiological point source. In *Proceedings of Information, Decision and Control (IDC)*, pages 236–241, February 2007.

[15] Nageswara S. V. Rao, Charles W. Glover, Mallikarjun Shankar, Jren-Chit Chin, David K. Y. Yau, Chris Y. T. Ma, Yong Yang, and Sartaj Sahni. Improved SPRT detection using localization with application to radiation sources. In *Proceedings of the IEEE 12th International Conference on Information Fusion (FUSION)*, July 2009.

[16] Nageswara S. V. Rao, Jren-Chit Chin, David K. Y. Yau, and Chris Y. T. Ma. Localization leads to improved distributed detection under non-smooth distributions. In *Proceedings of the IEEE 13th International Conference on Information Fusion (FUSION)*, July 2010.

[17] Nageswara S. V. Rao, Jren-Chit Chin, David K. Y. Yau, and Chris Y. T. Ma. Cyber-physical trade-offs in distributed detection networks. In *Proceedings of the IEEE Conference on Multisensor Fusion and Integration*, September 2010.

[18] J. H. Hubbell. Photon cross sections, attenuation coefficients, and energy absorption coefficients from 10 KeV to 100 GeV. *NSRDS-NBS 29*, 1969.

[19] G. F. Knoll. *Radiation Detection and Measurement*. John Wiley, 2000.

[20] D. Mihalas and B. W. Mihalas. *Foundations of Radiation Hydrodynamics*. Courier Dover Publications, 2000.

[21] Sanjoy Mukhopadhyay. Field deployable gamma radiation detectors for DHS use. In *Proceedings of SPIE*, volume 6706, 2007.

[22] Jren-Chit Chin, David K. Y. Yau, Nageswara S. V. Rao, Yong Yang, Chris Y. T. Ma, and Mallikarjun Shankar. Accurate localization of low-level radioactive source under noise and measurement errors. In *Proceedings of Conference on Embedded Networked Sensor Systems (SenSys)*, pages 183–196, November 2008.

[23] J. Cox and M. B. Partensky. Spatial localization problem and the circle of apollonius. *ArXiv Physics e-prints*, January 2007.

[24] John A. Hartigan. *Clustering Algorithms*. Wiley, 1975.

[25] Sudipto Guha, Rajeev Rastogi, and Kyuseok Shim. CURE: An efficient clustering algorithm for large databases. In *Proceedings of the 1998 ACM SIGMOD International Conference on Management of Data*, pages 73–84, 1998.

[26] Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining (KDD-96)*, pages 226–231, 1996.

[27] Laurie J. Heyer, Semyon Kruglyak, and Shibu Yooseph. Exploring expression data: Identification and analysis of coexpressed genes. *Genome Research*, 9:1106–1115, 1999.

[28] Gyula Simon, Miklós Maróti, Ákos Lédeczi, György Balogh, Branislav Kusy, András Nádas, Gábor Pap, János Sallai, and Ken Frampton. Sensor network-based countersniper system. In *Proceedings of Conference on Embedded Networked Sensor Systems (SenSys)*, 2004.

[29] Mark R. Morelande, Christopher M. Kreucher, and Keith Kastella. A Bayesian approach to multiple target detection and tracking. *IEEE Transactions on Signal Processing*, 55(5):1589–1604, May 2007.

[30] Mark Morelande, Branko Ristic, and Ajith Gunatilaka. Detection and parameter estimation of multiple radioactive sources. In *Proceedings of the IEEE 10th International Conference on Information Fusion (FUSION)*, July 2007.

[31] Xiaohong Sheng and Yu-Hen Hu. Maximum likelihood multiple-source localization using acoustic energy measurements with wireless sensor networks. *IEEE Transactions on Signal Processing*, 53(1), January 2005.

[32] Chris Kreucher, Keith Kastella, and Alfred O. Hero. Multitarget tracking using the joint multitarget probability density. *Proceedings of IEEE Transactions on Aerospace and Electronic Systems*, 41(4):1396–1414, October 2005.

[33] Branko Ristic, Mark Morelande, and Ajith Gunatilaka. Information driven search for point sources of gamma radiation. *ACM Signal Processing*, 90(4):1225–1239, 2010.

[34] Dorin Comaniciu and Peter Meer. Mean shift: A robust approach toward feature space analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(5), 2002.

[35] Arnaud Doucet, Nando de Freitas, Neil Gordon, and A. Smith. *Sequential Monte Carlo Methods in Practice.* Springer, 2001.

[36] N. J. Gordon, D. J. Salmond, and A. F. M. Smith. Novel approach to nonlinear/non-Gaussian Bayesian state estimation. In *IEE Proceedings F In Radar and Signal Processing*, volume 140, pages 107–113, August 2002.

[37] RFTrax radiation sensor. `http://www.rftrax.com/radczt.html`. [Accessed: September 10, 2007].

[38] Richard A. Muller. The dirty bomb distraction. `http://muller.lbl.gov/TRessays/29-Dirty_Bombs.htm`, June 2004. [Accessed: March 12, 2010].

[39] Samuel Blackman and Robert Popoli. *Design and Analysis of Modern Tracking Systems.* Artech House Publishers, Boston, MA, August 1999.

[40] Yaakov Bar-Shalom. *Multitarget-multisensor tracking: Principles and techniques*. Yaakov Bar-Shalom Publishing, 1995.

[41] N. L. Johnson. Sequential analysis: A survey. *Journal of the Royal Statistical Society. Series A (General)*, 124(3):372–411, 1961.

[42] Mono - cross platform, open source .NET development framework. `http://www.mono-project.com/`. [Accessed: June 11, 2010].

[43] Crossbow Technology. MDA300 multi-function data acquisition board. `http://www.xbow.com/Products/Product_pdf_files/Wireless_pdf/MDA300CA_Datasheet.pdf`. [Accessed: May 8, 2010].

[44] RFTrax, Inc. *RFTrax Security Sensor Communications Protocol Specification*, April 2005.

[45] William H. Press, Saul A. Teukolsky, William T. Vetterling, and Brian P. Flannery. *Numerical Recipes in C (2nd edition): The Art of Scientific Computing.* Cambridge University Press, New York, NY, 1992.

[46] Jren-Chit Chin, Jeffrey M. Rautenberg, Chris Y. T. Ma, Santiago Pujol, and David K. Y. Yau. A low-cost, low-data-rate rapid structural assessment network: Design, implementation, and experimentation. In *Proceedings of International Conference on Mobile Ad-hoc and Sensor Systems (MASS)*, September 2008.

[47] Phidgets, Inc. PhidgetRFID kit. `http://www.phidgets.com/products.php?category=14&product_id=2002`. [Accessed: May 7, 2010].

[48] H. Vincent Poor. *An Introduction to Signal Detection and Estimation.* Springer, New York, NY, 1994.

[49] Neil A. Macmillan and C. Douglas Creelman. *Detection Theory: A User's Guide.* Lawrence Erlbaum Associates, Mahwah, NJ, 2005.

[50] Thomas D. Wickens. *Elementary Signal Detection Theory.* Oxford University Press, New York, NY, 2002.

[51] Ruixin Niu and Pramod K. Varshney. Distributed detection and fusion in a large wireless sensor network of random size. *EURASIP Journal on Wireless Communications and Networking*, 2005(4), 2005.

[52] Luc Devroye, Laszlo Györfi, and Gabor Lugosi. *A Probabilistic Theory of Pattern Recognition.* Springer, New York, NY, 1997.

[53] Richard O. Duda, Peter E. Hart, and David G. Stork. *Pattern Classification.* Wiley, 2001.

[54] Yaakov Bar-Shalom, Xiao-Rong Li, and Thiagalingam Kirubarajan. *Estimation with Applications to Tracking and Navigation.* John Wiley and Sons, 2001.

[55] Harold Joseph Kushner and George Yin. *Stochastic Approximation and Recursive Algorithms and Applications.* Springer, 2003.

[56] Lennart Ljung. *System Identification: Theory for the User.* Prentice Hall PTR, 1999.

[57] Karl E. Nelson, John D. Valentine, and Brock R. Beauchamp. Radiation detection method and system using the sequential probability ratio test, 2007. U.S. Patent 7,244,930 B2.

[58] Daniel E. Archer, Brock R. Beauchamp, Joseph G. Mauger, Karl E. Nelson, Michael B. Mercer, David C. Pletcher, Vincent J. Riot, James L. Schek, and David A. Knapp. Adaptable radiation monitoring system and method, 2006. U.S. Patent 7,064,336 B2.

[59] Paul E. Fehlau. Comparing a recursive digital filter with the moving-average and sequential probability-ratio detection methods for SNM portal monitors. *IEEE Transactions on Nuclear Science*, 40(2):143–146, April 1993.

[60] Kenneth D. Jarman, L. Eric Smith, and Deborah K. Carlson. Sequential probability ratio test for long-term radiation monitoring. *IEEE Transactions on Nuclear Science*, 51(4):1662–1666, August 2004.

[61] Robert J. Nemzek, Jared S. Dreicer, David C. Torney, and Tony T. Warnock. Distributed sensor networks for detection of mobile radioactive sources. *IEEE Transactions on Nuclear Science*, 51(4):1693–1700, August 2004.

[62] Sean M. Brennan, Angela M. Mielke, and David C. Torney. Radiation detection with distributed sensor networks. *IEEE Computer*, 37(8):57–59, August 2004.

[63] Angela M. Mielke, Mark C. Smith, Sean M. Brennan, David C. Torney, Diana Jackson, Josh F. Karlin, and Arthur B. Maccabe. Radiation detection with distributed sensor networks. In *SPIE Defense and Security Proceedings*, volume 5778, pages 403–411, 2005.

[64] Daniel L. Stephens Jr. and Anthony J. Peurrung. Detection of moving radioactive sources using sensor networks. *IEEE Transactions on Nuclear Science*, 51(5):2273–2278, October 2004.

[65] A. Sundaresan, P. K. Varshney, and Nageswara S. V. Rao. Distributed detection of a nuclear radioactive source using fusion of correlated decisions. In *Proceedings of the 10th International Conference on Information Fusion (FUSION)*, July 2007.

[66] Wei Wang, Vikram Srinivasan, Bang Wang, and Kee-Chaing Chua. Coverage for target localization in wireless sensor networks. In *Proceedings of the 5th International Conference on Information Processing in Sensor Networks (IPSN)*, pages 118–125, April 2006.

[67] Tian He, Sudha Krishnamurthy, John A. Stankovic, Tarek Abdelzaher, Liqian Luo, Radu Stoleru, Ting Yan, Lin Gu, Jonathan Hui, and Bruce Krogh. Energy-efficient surveillance system using wireless sensor networks. In *Proceedings of the 2nd International Conference on Mobile Systems, Applications, and Services (MobiSys)*, pages 270–283, 2004.

[68] Joe C. Chen, Ralph E. Hudson, and Kung Yao. Maximum-likelihood source localization and unknown sensor location estimation for wideband signals in the near-field. *IEEE Transactions on Signal Processing*, 50(8):1843–1854, August 2002.

[69] Dan Li, Kerry D. Wong, Yu Hen Hu, and Akbar M. Sayeed. Detection, classification, and tracking of targets. *IEEE Signal Processing Magazine*, 19(2):17–29, March 2002.

[70] Dale N. Anderson, David C. Stromswold, Sharon C. Wunschel, Anthony J. Peurrung, and Randy R. Hansen. Detection and location of gamma-ray sources with a modulating coded mask. *Technometrics*, 48(2):252–261, 2006.

[71] Annie Liu, Matt Wu, Mani Chandy, Daniel Obenshain, Mason Smith, and Ryan McLean. Design tradeoffs for radiation detection sensor networks. Preprint, available at `http://www.cs.caltech.edu/~aliu/documents/IPSN_final.pdf`.

[72] Alan M. Mainwaring, David E. Culler, Joseph Polastre, Robert Szewczyk, and John Anderson. Wireless sensor networks for habitat monitoring. In *Proceedings of the First ACM International Workshop on Wireless Sensor Networks and Applications (WSNA)*, pages 88–97, September 2002.

[73] Jinyang Li, John Jannotti, Douglas S. J. De Couto, David R. Karger, and Robert Morris. A scalable location service for geographic ad hoc routing. In *Proceedings of the 6th Annual International Conference on Mobile Computing and Networking (MOBICOM)*, pages 120–130. ACM, 2000.

[74] Konstantinos N. Amouris, Symeon Papavassiliou, and Miao Li. A position-based multi-zone routing protocol for wide area mobile ad-hoc networks. In *Proceedings of the 49th IEEE Vehicular Technology Conference*, volume 2, pages 1365–1369, July 1999.

[75] Nissanka B. Priyantha, Anit Chakraborty, and Hari Balakrishnan. The cricket location-support system. In *Proceedings of the 6th Annual International Conference on Mobile Computing and Networking (MOBICOM)*, pages 32–43, 2000.

[76] Chuan Han, Siyu Zhan, and Yaling Yang. Proactive attacker localization in wireless LAN. *ACM SIGCOMM Computer Communication Review*, 39(2):27–33, 2009.

[77] Chiara Falsi, Davide Dardari, Lorenzo Mucchi, and Moe Z. Win. Time of arrival estimation for UWB localizers in realistic environments. *EURASIP Journal on Applied Signal Processing*, 2006, 2006.

[78] Paul Cheong, Alberto Rabbachin, Jean-Philippe Montillet, Kegen Yu, and Ian Oppermann. Synchronization, TOA and position estimation for low-complexity LDR UWB devices. In *Proceedings of IEEE International Conference on Ultra-Wideband (ICU)*, pages 480–484, September 2005.

[79] Andreas Savvides, Chih-Chieh Han, and Mani B. Strivastava. Dynamic fine-grained localization in ad-hoc networks of sensors. In *Proceedings of the 7th Annual International Conference on Mobile Computing and Networking (MOBICOM)*, pages 166–179, 2001.

[80] Xiuzhen Cheng, A. Thaeler, Guoliang Xue, and Dechang Chen. TPS: A time-based positioning scheme for outdoor wireless sensor networks. In *Proceedings of the 23rd IEEE International Conference on Computer Communications (INFOCOM)*, volume 4, pages 2685–2696, March 2004.

[81] Andrew Thaeler, Min Ding, and Xiuzhen Cheng. iTPS: An improved location discovery scheme for sensor networks with long-range beacons. *Journal of Parallel and Distributed Computing*, 65(2):98–106, 2005.

[82] Nageswara S. V. Rao. Identification of simple product-form plumes using networks of sensors with random errors. In *Proceedings of the 9th International Conference on Information Fusion (FUSION)*, July 2006.

[83] Nageswara S. V. Rao, Xiaochun Xu, and Sartaj Sahni. A computational geometry method for DTOA triangulation. In *Proceedings of the 10th International Conference on Information Fusion (FUSION)*, 2007.

[84] Mustapha Boushaba, Abdelhakim Hafid, and Abderrahim Benslimane. High accuracy localization method using AoA in sensor networks. *Computer Networks: The International Journal of Computer and Telecommunications Networking*, 53(18):3076–3088, 2009.

[85] Dragos Niculescu and Badri Nath. Ad hoc positioning system (APS) using AOA. In *Proceedings of the 22nd Annual Joint Conference of the IEEE Computer and Communications (INFOCOM)*, volume 3, pages 1734–1743, March 2003.

[86] Tian Hui, Wang Shuang, and Xie Huaiyao. Localization using cooperative AOA approach. In *Proceedings of International Conference on Wireless Communications, Networking and Mobile Computing (WiCom)*, pages 2416–2419, September 2007.

[87] Chunhua Yang, Yi Huang, and Xu Zhu. Hybrid TDOA/AOA method for indoor positioning systems. In *The Institution of Engineering and Technology Seminar on Location Technologies*, pages 1–5, December 2007.

[88] Carsten Fritsche and Anja Klein. On the performance of hybrid GPS/GSM mobile terminal tracking. In *Proceedings of IEEE International Conference on Communications Workshops*, pages 1–5, June 2009.

[89] Gang Zhou, Tian He, Sudha Krishnamurthy, and John A. Stankovic. Impact of radio irregularity on wireless sensor networks. In *Proceedings of the 2nd International Conference on Mobile Systems, Applications, and Services (MobiSys)*, pages 125–138, 2004.

[90] Martin Azizyan, Ionut Constandache, and Romit Roy Choudhury. SurroundSense: Mobile phone localization via ambience fingerprinting. In *Proceedings of the 15th Annual International Conference on Mobile Computing and Networking (MOBICOM)*, pages 261–272, 2009.

[91] Ajith Gunatilaka, Branko Ristic, Alex Skvortsov, and Mark Morelande. Parameter estimation of a continuous chemical plume source. In *Proceedings of the 11th IEEE International Conference on Information Fusion (FUSION)*, June 2008.

[92] Peter Robins and Paul Thomas. Non-linear Bayesian CBRN source term estimation. In *Proceedings of the 8th IEEE International Conference on Information Fusion (FUSION)*, volume 2, July 2005.

[93] James W. Howse, Lawrence O. Ticknor, and Kenneth R. Muske. Least squares estimation techniques for position tracking of radioactive sources. *Automatica*, 37:1727–1737, 2001.

[94] R.B. Vilim, R.T. Klann, S.C. de la Barrera, P.L. Vilim, and I.A. Ross. Tracking of weak radioactive sources in crowded venues. In *IEEE Nuclear Science Symposium Conference Record (NSS/MIC)*, pages 995–1001, October 2009.

[95] Min Ding and Xiuzhen Cheng. Fault tolerant target tracking in sensor networks. In *Proceedings of the 10th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc)*, pages 125–134, 2009.

[96] Mark R. Morelande, Bill Moran, and Marcus Brazil. Bayesian node localisation in wireless sensor networks. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2545–2548, March 2008.

[97] Wei Xia and Zishu He. Multiple-target localization and estimation of MIMO radars with unknown transmitted signals. In *IEEE International Symposium on Circuits and Systems (ISCAS)*, pages 3009–3012, May 2008.

[98] Yang Cheng and Tarunraj Singh. Source term estimation using convex optimization. In *Proceedings of the 11th International Conference on Information Fusion (FUSION)*, June 2008.

[99] Chris Kreucher, Mark Morelande, Keith Kastella, and Alfred O. Hero. Particle filtering for multitarget detection and tracking. In *Proceedings of IEEE Aerospace Conference*, pages 2101–2116, March 2005.

[100] Yi Zou and Krishnendu Chakrabarty. Sensor deployment and target localization in distributed sensor networks. *ACM Transactions on Embedded Computing Systems*, 3(1), February 2004.

[101] B. Ristic, M. Morelande, and A. Gunatilaka. A controlled search for radioactive point sources. In *Proceedings of the 11th International Conference on Information Fusion (FUSION)*, June 2008.

[102] Christos Christodoulopoulos, Christos Kyriakopoulos, and Athanasios G. Kanatas. A realistic approach to source localization using a wireless robotic network. In *Proceedings of the 1st International Conference on Robot Communication and Coordination (ROBOCOMM)*, pages 1–4, Piscataway, NJ, 2007. IEEE Press.

[103] P. Fabiani, H. H. Gonzlez-Baos, J. C. Latombe, and D. Lin. Tracking an unpredictable target among occluding obstacles under localization uncertainties. *Robotics and Autonomous Systems*, 38(1):31–48, 2002.

VITA

VITA

Jren-Chit Chin received his B.Sc. in Computer Engineering from Iowa State University, Ames, Iowa, USA in August 2005. He received his Ph.D. degree from Purdue University, West Lafayette, Indiana, USA in August 2010, under the direction of Professor David K. Y. Yau. In September 2010, he joined Microsoft Corporation in Redmond, Washington, USA. His research interests are in wireless sensor networks, including target detection and localization, sensor coverage, mobility problems, and embedded systems.