Mobile Sensor Network Design and Optimization for Air Quality Monitoring

by

Yun Xiang

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Electrical and Computer Engineering)
in The University of Michigan
2014

Doctoral Committee:

Associate Professor Robert Dick, Chair
Professor Stuart Batterman
Assistant Professor Prabal Dutta
Professor Mingyan Liu

ACKNOWLEDGEMENTS

This thesis was completed under the advice and guidance from my adviser, Prof. Robert P. Dick. He has given me the opportunity to start research and provided me great help to endure the toughest time of my Ph.D. career. It is unthinkable to finish my Ph.D. study without him. For that, I would sincerely thank him first.

I also want to say thanks to my collaborators. Professor Tam Chantem from University of Utah and her Ph.D. adviser Professor X. Sharon Hu from University of Notre Dame have provided unique insight and great suggestions for my first paper. Without them, my road for research would be much harder and longer. Prof. Qin Lv, Prof. Shang Li, and Prof Michael Hannigan, all from University of Colorado Boulder, are my collaborators for all the papers involved in this thesis. They have played a very important role in my research life. Therefore. I want to thank them here for their weekly inputs and discussions, and efforts in revising the papers. I feel really lucky to have the great opportunity to work with them.

I would like to express my gratitude to Professor Stuart Batterman, Professor Prabal Dutta, and Professor Mingyan Liu for serving in my Ph.D. committee. During my proposal defense, they have given many valuable suggestions. Some of them have been the motivation for the last piece of the work in this thesis. They have made this thesis better and more comprehensive. I would also like to thank my other collaborators. Ricardo Pierdrahita has worked with me on almost all of my works, except for the first one. He is an expert on environment engineering. He has given lots of valuable inputs and revised

and co-authored many papers with me. Most of my work can not be finished without his expertise and help. For that, I owe him my gratitude. Yifei Jiang is another co-author that I want to thank. He is an expert on system design and mobile applications. He has designed the mobile app, server, and database for the M-Pod. He has made my Ph.D. research life much easier. We have also co-authored many papers.

I would say thanks to my colleagues and friends. Xuejing He, Yue Liu, Lide Zhang, David Bild, Lan Bai, Xi Chen, and Phil Knag have all given me great suggestions as colleagues and spent great time together after work as friends. My life would be a lot more boring and dull without them. Furthermore, Lan has collaborated with me in the collaborative calibration work and made tremendous contributions towards its completion.

Finally, I would like to thank my family, especially my parents, Xinzhang Mao and Lipin Yang. Without their help and encouragement, my journey would not even be possible to start. Thus, I dedicate this thesis to them.

TABLE OF CONTENTS

LIST OF TABLES

Table

LIST OF FIGURES

Figure

ABSTRACT

Air quality and personal pollutant exposure measurement are important for the health and productivity of individuals. Accurate measurement of personal exposure is challenging because of the spatially and temporally heterogeneous distribution of pollutant concentrations. We propose to use low-cost and miniature mobile sensor networks to provide real-time measurement of the environment directly surrounding the user. However, there are many challenges, including sensor drift, cross sensitivity, and noises, to be addressed before mobile sensor network can be deployed in large scale and real-world applications.

My thesis aims to address those challenges by designing prototype sensor nodes of future generation mobile sensor networks, developing optimization techniques and systems, and evaluating the mobile sensor network in real-world deployments. My efforts can be divided into four categories: (1) we design the mobile sensor nodes and the mobile sensor network architecture that are capable of automatically collecting environment data and transferring them to a database; (2) we model the sensor drift based on measurement and develop techniques such as collaborative calibration and optimal human mobility-aware sensor placement to minimize the drift error of individual sensors; (3) we model the pollutant concentration in indoor environment considering inaccurate sensors and based on the model, we develop a hybrid sensor network synthesis technique to design accurate sensor networks under a cost constraint; and (4) we propose a Bayesian network based sensor noise reduction system that can correct abnormal sensor readings, re-calibrate the sensor

functions, and identify the gas composition is the environment simultaneously. All the techniques are evaluated and validated using the data collected from real-world deployment. Experimental and simulation results show that our technique can reduce drift error significantly. For example, compared with the closest technique, our collaborative calibration technique can reduce sensor network error by 23.2%; our hybrid sensor network synthesis technique can improve the result by 35.8%; and our noise reduction technique can outperform the existing technique by 34.1%.[1]

[1]This work was supported in part by NSF under award CCF-1217674.

CHAPTER I

Introduction

Air quality is important. Personal exposure to air pollutants is strongly related to the health and productivity of individuals. For example, long-term exposure to ozone (O_3), volatile organic compound (VOC), and particulate matter (PM) can cause chronic diseases, various cancers, and thus increased human mortality [27, 55]. Moreover, even some typically harmless and naturally existing gases, such as CO_2, can cause sick building syndrome and significantly reduce productivity if in high concentration. Thus, the demand for better air quality and tighter environmental regulation is increasing significantly worldwide. Sometimes, they can even cause social tension and unrest [2].

In response to a growing need for better air quality monitoring, mobile sensing applications are increasingly popular. The fast development of smartphones and sensor technology makes many such applications possible, e.g., mobile noise pollution sensing networks [46] and mobile personalized air quality sensor networks [35]. Compact, light, and energy-efficient sensors are now becoming available at prices that permit widespread use by non-scientists (and scientists). In the future, individuals will carry multiple unobtrusive sensors with them, within or networked with their smartphones, forming dense and interconnected sensor networks. Mobile sensing applications will soon become mainstream.

Mobile sensing systems have many advantages over conventional systems composed

of a few accurate, low-drift, stationary, and expensive sensing stations. For example, in the personal air quality sensing applications, many pollutants have nonuniform spatial distributions [66]. As a result, personal exposure is poorly estimated by using sparsely distributed stationary sensors. If each participant in a sensing system were to carry a sensor, we would be able to better understand human exposure and provide more relevant information to users.

However, before mobile air quality sensor networks can be used in real-world applications, there are still many challenges to overcome. Those challenges include, but not limited to, sensor drift, cross sensitivity, and sensor noise.

- *Sensor drift.* Drift is the gradual deviation of a sensor's readings from the ground truth value. It is affected by many factors that change the sensing surface and thus change the sensor function that translates the analog sensor inputs into pollutant concentrations. Mobile sensors are generally more susceptible to drift than stationary sensors due to trade-offs made for compactness and economy. Our deployment data has shown that even within a short period of time, such as several months, the drift can be significant enough to make the sensor useless. This problem is amplified because it is difficult to frequently calibrate mobile sensors, especially when they are carried by non-specialists. Thus, for sensor drift, the main challenge is, "how to model the drift and compensate for its error in real-world applications?"

- *Cross sensitivity.* Cross sensitivity refers to the sensor responding to gases in the air other than the targeting pollutant. The low-cost sensors typically have poor selectivity, i.e., their readings can be influenced by multiple pollutants, or even humidity. In real-world applications, the types of pollutant gases in the air are usually unknown and unpredictable, which cause additional uncertainties to the measurement

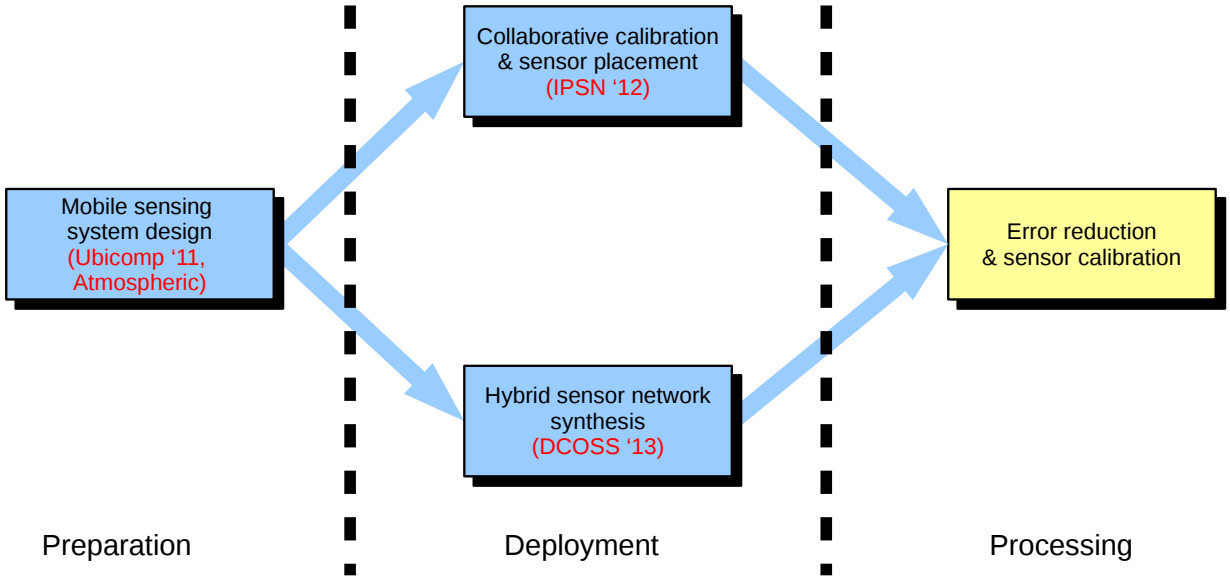


Figure 1.1: Flow chart of the thesis.

results and make the drift calibration more unreliable. For cross sensitivity, the main challenge is, "How to identify the gas composition in the air and quantify their concentration separately under the influence of drift?"

- *Sensor noise* The readings reported by the metal oxide sensors usually contain a significant amount of noises. They can be caused by random environment and electrical noises, cross sensitivity, and drift. The sensor error caused by random noises and cross sensitivity can be detected and compensated for using a Bayesian network based approach by exploiting the correlation between sensors. However, the abnormal readings caused by sensor drift can not be corrected by a basic Bayesian belief network directly. Thus, the main challenge is, "How to differentiate and remove the sensor noise caused by drift and re-calibrate the drifted sensor?"

In this work, We have demonstrated that **using indoor airflow based modeling, human mobility based sensor placement optimization, and Bayesian reasoning based machine-learning techniques can reduce error due to sensor drift and noise by more**

than 30% relative to the existing error compensation methods, making mobile air quality sensor networks more practical in real-world applications.

Specifically, in this work, we will design novel calibration and deployment schemes to minimize drift error, classify and correct noisy readings, design and build low-cost sensing devices and use them to validate the concept of mobile sensor network through real-world deployments. Figure 1.1 describes the steps to achieve these goals. I'll elaborate on each piece in the following subsections.

1.1 Mobile Sensor Network Design and Deployment

To form mobile sensor networks, the basic requirement is the availability of low-cost sensing devices capable of sensing multiple relevant environmental parameters. For example, we need several metal oxide gas sensors to monitor various types of pollutants in the air. We also need temperature and humidity sensors to calculate the pollutant concentration from the analog readings reported by the metal oxide sensors. Therefore, we have designed a personal mobile air quality sensing (MAQS) platform, which includes a small mobile pollution sensing pod (M-Pod) and a smartphone application. The M-Pod is a wireless embedded sensing, computation, and communication device based on the design of Arduino BT [1]. It supports detection of various air pollutants, including NO_2, CO, CO_2, O_3, and volatile organic compounds (VOCs). It can also measure temperature, humidity, and light intensity. The total cost of all the components of the sensing platform is less than $150.

Because of all the drift, cross sensitivity, reliability, and noise problems, the concept of mobile air quality sensor network needs to be evaluated and validated. We have designed a system, based on the M-Pod design, that can automatically collect data from the individual users, transfer them to the database via WiFi, and display them through a web

interface. Using our mobile sensor network system, we have performed various real-world deployments, which can provide user exposure data, help us understand the sensor drift and cross sensitivity, and build dataset for the evaluation of our techniques.

1.2 Collaborative Calibration and Sensor Placement

Another significant problem of the metal oxide sensors is drift. The low-cost sensors stationed on the M-Pod are susceptible to measurement drift and can accumulate substantial drift error in short time spans. The cause of drift has been demonstrated by many existing works [28, 57]. We have also performed a controlled experiment in a gas chamber to better understand and model drift error. To compensate for drift error, we propose a realistic drift model based on analysis of our drift experiment data. Based on the drift model, we have designed optimal collaborative calibration and stationary sensor placement techniques. By allowing the mobile sensors to calibrate with each other optimally and maximizing the rates at which mobile sensors can implicitly calibrate with stationary sensors, the overall accuracy of mobile sensor networks can be significantly improved.

1.3 Hybrid Sensor Network Modeling and Synthesis

The collaborative calibration technique can improve the accuracy of individual sensors under assumption of a densely deployed sensor network. However, in real-world applications, deployment is usually subject to cost constraint. Therefore, it is desirable to develop a sensor network synthesis technique to maximize the accuracy of the sensor network while controlling the total cost. We propose a hybrid sensor network architecture, which includes accurate stationary sensors (to support calibration) and inaccurate mobile sensors (to provide personalized measurement). The deployment field is divided into multiple zones. We have derived optimal models to estimate the pollutant concentration in

zones that are not covered or covered by inaccurate sensors. Based on the optimal model, we have developed a synthesis algorithm that can maximize the sensor network accuracy under a cost constraint.

1.4 Error Reduction and Sensor Re-calibration

For the low-cost sensors, one major problem that causes measurement error in real-world applications is cross sensitivity. Besides the targeting pollutant, the low-cost sensors usually respond to a wide range of pollutants. However, cross sensitivity also causes correlation between different types of sensors, which can be exploited to compensate for drift and re-calibrate the sensors.

To detect the abnormal readings and identify the gas composition in the air, we propose to use the Bayesian network to model and quantify the inter-dependencies of different types of sensors observing the same physical environment. Furthermore, to address the sensor drift problem which can not be handled by Bayesian network directly, we have designed a system incorporating virtual evidence and sensor function re-calibration. Based on the dataset derived from a real-world co-location deployment, it is shown that our technique can reduce error significantly.

1.5 Thesis Organization

This dissertation is organized as follows.

- Chapter II describes our custom-built M-Pod sensing platform, which is the basic sensing node of our mobile air quality monitoring system. This chapter explains the design of our system and some real-world deployment experiences.
- Chapter III describes the technique to automatically calibrate the sensors collaboratively, i.e., calibration among mobile sensors. It also presents the mixed-integer

linear programming (MILP) based stationary sensor placement technique to maximize the opportunities for calibration.

- Chapter IV talks about a hybrid sensor network synthesis technique based on indoor environment modeling. This technique aims to improve the accuracy of the sensor network given a budget constraint.
- Chapter V presents our Bayesian network based technique that can detect and recover the sensor noise caused by sensor drift, re-calibrate the sensor functions, and identify the gas composition in the environment simultaneously.
- Chapter VI concludes the thesis.

CHAPTER II

M-Pods and Air Quality Monitoring Systems Design

2.1 Introduction

Research has shown that people in the U.S. spend 90% of their time indoors [67]. Only 26% of buildings meet the air quality standards established by the American Society of Heating, Refrigerating, and Air Conditioning Engineers (ASHRAE) [31]. Poor air quality hurts human health, productivity, safety, and life quality [17, 40, 69]. We propose to use mobile environmental sensor networks to monitor personal air quality. Mobile personal air quality sensors have a tremendous advantage over stationary sensing systems: they measure pollution where their users (carriers) are.

Air quality data are presently primarily measured using accurate, professionally maintained, stationary, and expensive pollution sensing equipment. For example, the instrument used to measure carbon dioxide at Mauna Loa requires thousands of dollars to maintain and staff [63], while a portable infrared carbon dioxide sensor costs less than $100 [3].

Compared to stationary sensors, mobile sensor networks support more accurate personal pollution exposure measurement. Stationary sensors and instruments are usually sparse and many pollutants have nonuniform spatial and temporal distributions [66]. Although the on-going reduction of miniature sensors' prices might allow more dense stationary sensor networks in the future, the mobile sensors can still be more accurate in many

Figure 2.1: M-pod personal air quality sensor.

situations, e.g., while in transit or in locations visited by few people. Inaccurate personal exposure estimation can result in incorrect scientific conclusions, unnoticed health risks, and bad regulation decisions.

We describe a personal mobile environmental sensing network composed of a large number of compact, light, and energy-efficient pollution sensors [35]. We have developed the M-pod, a mobile air pollution sensing device for personal air quality monitoring. It uses miniature and inexpensive sensors. The low price of platforms such as the M-pod may permit widespread use by non-scientists as well as scientists.

2.2 Mobile Pollution Sensing Device

The M-pod (shown in Figure 2.1) is a mobile sensing platform supporting embedded sensing, computation, and wireless communication. Table 2.1 lists the components. It

Table 2.1: M-Pod Components

Hardware specs	MCU ATMEGA 168	Bluetooth WT11	Battery Off-the-shelf	Size (inch) 2×2.5
On-board sensors	Temperature TMP100	CO_2 S100	Humid. & Temp. SHT21	Light GL5528

supports detection of various air pollutants, including NO_x, CO, CO_2, ozone, and VOCs. It can also measure temperature, humidity, and light. The latest revision of the M-pod is compact (2×2.5 inches) and energy efficient, with a battery life of greater than 12 hours. The whole device, including a Li-ion battery with a capacity of 6,000 mA-h, is enclosed by a low-cost off-the-shelf case that can be carried using an armband or attached to a backpack. A 3.3 V DC fan is used to control airflow. A rectangular filter is installed around the sensors to increase sensing accuracy and prolong sensor life. Most of the power hungry on-board sensors are power gated and can be controlled by commands from smartphones. Data are temporally stored in a one megabyte non-volatile EEPROM. The total cost of the on-board components and sensors is less than $150 and can be reduced further if produced in quantity.

To receive, store, and present the data gathered by our M-pod device, we have developed on-board firmware, smartphone applications, data servers, and web interfaces. The firmware defines protocols of sensing, storing, and sending the environmental data. The smartphone application communicates with the M-pod via its Bluetooth interface. It can issue commands to and receive data from the M-pod. The data are transmitted to the online data server and stored in the databases. A web-based user interface allows users to access and analyze air quality data.

2.3 Deployment Experience

The M-pod has been used in several experiments at the University of Michigan and the University of Colorado Boulder. M-pods were introduced to students from Diné College

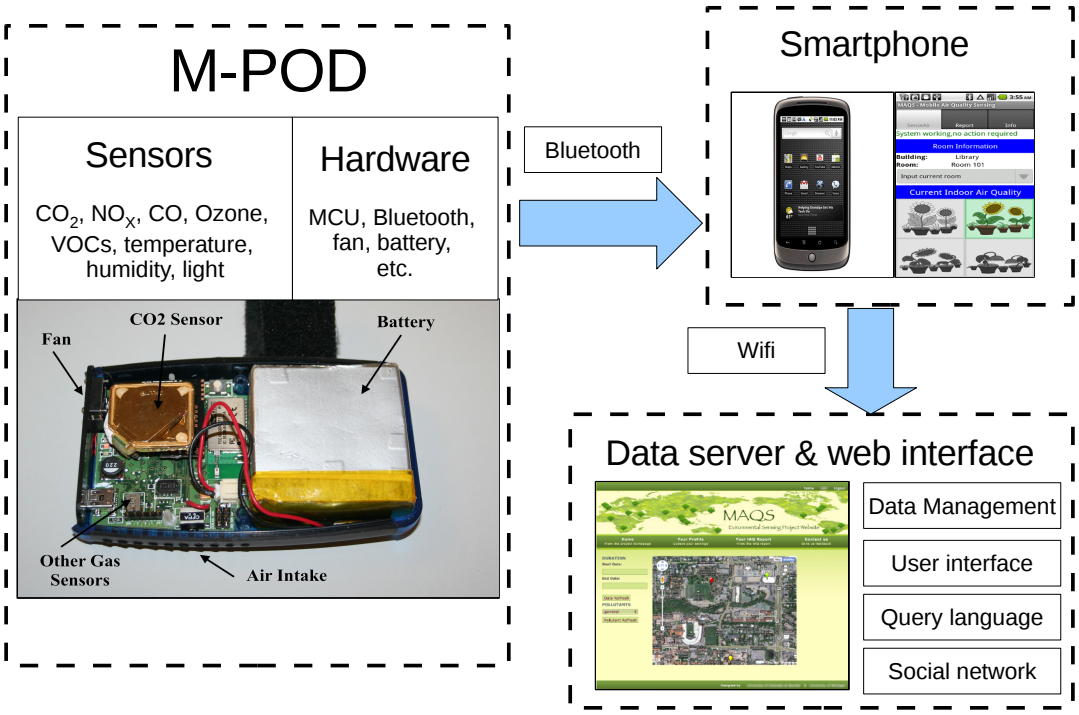


Figure 2.2: M-pod system overview.

at two workshops. At each workshop, approximately 10 participants paired up to carry 5 M-pods. The first workshop deployment lasted several days and the second workshop deployment lasted four weeks. Another co-location deployment, which lasts two months, allows us to investigate sensor drift. The details of this deployment can be found in Chapter V.

CHAPTER III

Collaborative Sensor Calibration and Sensor Placement

3.1 Introduction

During the deployment of our M-Pod system, as well as other metal oxide sensor based devices, a major problem we have encountered is sensor drift. Drift is a function of various factors such as sensing material, exposure to sulfur compounds or acids, aging, or condensate on the sensor surface [6, 28]. It is reported that short-term sensor drift can be modeled accurately with simple models but long-term drift is less predictable [21, 28, 57]. Erroneous measurements caused by sensor drift can result in incorrect scientific conclusions, false alarms, and bad decisions. Therefore, low cost sensors require frequent re-calibration.

Manually calibrating sensors to compensate for drift is time-consuming and burdensome; it can annoy users and limit their desire to use the sensors, which will result in an ineffective system. Automatic calibration (which requires no explicit user intervention) has the potential to solve these problems, thereby increasing mobile sensing opportunities.

We propose a system supporting automatic, opportunistic, and collaborative calibration among mobile sensors. Our solution takes into account the gradual increase in sensor drift error with time, and appropriately weights different calibration events based on the time-dependent estimated errors of the other sensors, i.e., we consider the temporal and spatial

properties of the graph formed by (transitive) calibration events. Although we do not require the presence of stationary sensors, we support their inclusion in the system, and also provide algorithms for determining their best locations. Our evaluation makes use of controlled sensor drift studies as well as measured human motion patterns.

The proposed collaborative calibration approach is appropriate for applications with the following characteristics.

1. Spatial variation of sensor readings are low within certain physical distance.

2. Sensor nodes are able to communicate with each other and detect when they are within calibration distance, e.g., either by tracking their own locations or by measuring signal attenuation between nodes.

3. Sensor drift can be compensated for using a drift predictor. The residual error of this predictor has a Gaussian distribution with variance that increases as a function of time, as explained in Section 3.4.2 and demonstrated in Section 3.6.1.

Our technique can potentially be used in many mobile sensing applications, such as radiation sensing applications in which sensors are carried by individuals and unmanned aerial vehicles, remote sensing applications in which detailed data are available from in-field sensors and sparse data are available from satellites, and personal environmental sensing. Although the concepts we develop apply to a broader range of mobile sensing systems susceptible to drift error, in the rest of paper, we focus our discussion on a personal air quality sensing application.

It should be noted that collaborative calibration minimizes the increase in the rate of uncompensable drift error, but does not eliminate error. Without the stationary accurate sensors, the mobile sensor network's overall accuracy degrades over time. The use of a few stationary accurate sensors to augment mobile collaborative calibration is beneficial;

it allows the drift error to be bounded.

Our work makes the following main contributions.

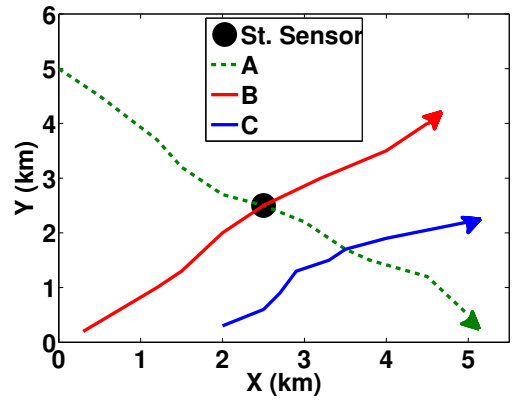
1. We formulate and solve the opportunistic collaborative mobile sensor calibration problem.

2. We formulate and solve the mobility aware stationary sensor placement problem to augment collaborative calibration.

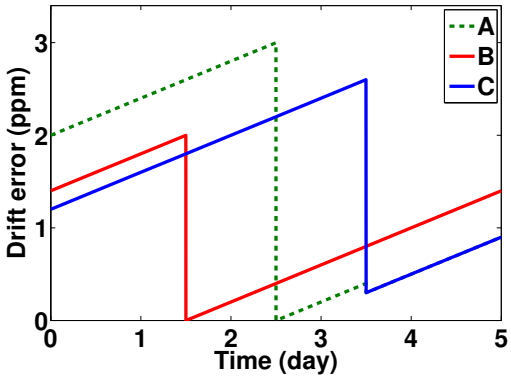
3. We propose a sensor drift model built using experimental data from 15 VOC sensors.

To better understand and characterize the effects of real-world human motion on calibration, we also carried out an indoor human motion pattern study on a university campus. Compared with our collaborative calibration scheme, the most advanced existing auto-calibration technique has an average error of 23.2%, while our efficient heuristic has an error of 2.2%. We also present two algorithms for placing stationary sensors to further improve mobile collaborative calibration. The use of well-placed stationary sensors within the collaborative calibration system techniques reduces sensing error significantly, e.g., by about 40% for a density of 1 stationary sensors per 25 mobile sensors. The approximation algorithm based placement technique results in only 6.2% more error than an mixed-integer linear programming (MILP) based technique.

The rest of this chapter is organized as follows. Section 3.2 gives a motivating example. Section 3.3 summarizes the related work on collaborative calibration and stationary sensor placement. Section 3.4 describes the sensor random drift model and our collaborative calibration method. Section 3.5 generalizes the human mobility model, and provides an MILP based solution for the human motion aware stationary sensor placement problem as well as an approximation algorithm. Section 3.6 describes our controlled-environment experiments for sensor drift and the data analysis results. It also evaluates the performance



(a)



(b)

Figure 3.1: (a) Human motion traces and calibration events and (b) drift errors for three sensors.

of our techniques using simulations based on real-world and synthesized human motion traces. Section 3.7 concludes the paper.

3.2 Motivating Example

Consider a mobile sensor network formed by sensing devices carried by individuals to monitor their air pollution exposures. Each device houses small, energy efficient, and inexpensive metal oxide gas sensors that measure various air pollutants. The sensor measurements gradually drift over time. Drift rates can vary greatly; to minimize error, the sensors must be re-calibrated frequently. In many cases, accurate stationary sensors are not readily accessible for users, and the occasional calibration opportunities they provide

are insufficient to cover all the participants in the sensing system. By using collaborative calibration together with optimized placement of stationary sensors, accuracy can be significantly improved.

Figure 3.1 illustrates an example of our mobile sensor network calibration technique. Figure 3.1(a) shows the trajectories of three mobile sensors (A, B, and C). Figure 3.1(b) shows their uncompensable drift errors over time. Each vertical drop in Figure 3.1(b) corresponds to one calibration event. Between calibration events, the drift error increases with time as a result of reduced drift prediction accuracy. Given the mobile sensor motion traces, our sensor placement approach decides where to put accurate stationary sensors to maximize the probabilities of mobile sensors being calibrated against the stationary sensor. In this example, the stationary sensor is located at a position both sensor A and B visit, thus providing ground truth calibration for two sensors. When sensor A and B get close to the accurate stationary sensor, their errors drop due to calibration (refer to Figure 3.1(b)). Our problem formulation and solution also consider a realistic human mobility model that considers individual motion traces able to represent day-to-day variation. With our collaborative calibration technique, even though sensor C never directly calibrates with any (accurate) stationary sensor, its drift error still reduces in the third day by calibrating with sensor A, which has a smaller error due to recent calibration with an accurate stationary sensor.

3.3 Related Work

This section summarizes prior work on auto-calibration and placement for distributed sensor networks.

Bychkovskiy et al. [12] proposed a two-phase post-deployment calibration technique for dense stationary sensor networks. In the first phase, linear relative calibration relations

are derived for pairs of co-located sensors. In the second phase, the consistency of the pair-wise calibration functions among groups of sensor nodes is maximized. Their technique requires a dense deployment of stationary sensors. In contrast, our work focuses on mobile sensor networks.

Miluzzo et al. [49] proposed an auto-calibration algorithm for mobile sensor networks, called CaliBree. In their approach, uncalibrated mobile nodes opportunistically calibrate themselves when interacting with stationary sensors. In their work, calibration events always involve stationary sensors. Our work supports calibration with stationary sensors, but in contrast also supports calibration among mobile sensors, allowing either higher accuracy or a reduction in the number (and therefore cost) of stationary sensors.

Tsujita et al. [65,66] studied calibration for air pollution monitoring networks. They [66] observed that at a certain time of day, the nitrous oxide pollutant concentration becomes low and uniform in certain areas. They use these opportunities to calibrate mobile sensors using the pollutant concentration reported from nearby environment monitoring stations. In their other work [65], when multiple sensors are close to each other, the average of their readings is used as ground truth to estimate sensor drift. In contrast, we account for the gradual increase in drift error as a function of time, allowing an optimal weighting for each of the many calibration events used to determine drift compensation parameters. Our experimental results show that the technique proposed by Tsujita et al. technique has 23.2% error relative to the optimal result; our proposed heuristic only has 2.2% error.

Berry et al. [7] used an MILP based method to solve the $\mathcal{NP}$-hard problem of placing sensors in water networks for optimal contamination detection. Chakrabarty et al. [13] tried to find an optimal sensor placement scheme to minimize the cost of sensors while meeting coverage constraints. Our problem formulation differs in that mobile sensors are carried by individuals. A realistic human mobility model is therefore necessary to solve

our placement problem. We build our human mobility model based on previous research and our indoor human motion study, and solve the stationary sensor placement problem using a high quality but potentially slow MILP method and an efficient approximation algorithm based technique.

3.4 Collaborative Calibration

This section describes our collaborative calibration technique. We present the problem definition, mathematical analysis, and our algorithm to solve this problem optimally.

3.4.1 Overview

Our collaborative calibration technique uses drift modeling and sensor fusion to reduce drift-related sensor measurement error. Sensor drift models, or drift predictors, are built based on past measured or estimated drift errors. They are used to estimate sensor drift at any point of time and (partially) compensate for drift errors in sensor measurements. In addition, the drift model allows the *residual error* of the drift predictor to be predicted as a function of time. Sensor fusion uses measurements from co-located sensors to improve the accuracy of the combined results. The fusion algorithm determines how to combine multiple sensor measurements based on their residual errors in order to maximize the combined accuracy. In implicit mobile calibration, sensor fusion happens whenever sensors happen to be close to each other; our calibration technique is opportunistic and collaborative.

Since nearby sensors are exposed to similar physical conditions, readings from co-located sensors can be combined to statistically improve accuracy. As mentioned before, each sensor has a residual error associated with its post-drift-compensation measurement. Each calibration event allows this error to be reevaluated and potentially reduced. If the two residual errors are independent, the measurement with the smaller residual error should be given more weight during combination. Calibration relationships introduce correlations

in sensors' residual errors that the calibration algorithm must account for. Section 3.4.3 describes our correlation-aware fusion algorithm in detail.

3.4.2 Collaborative Calibration Problem Definition

Our analytical framework can handle classes of mobile and stationary sensors with arbitrary drift rates. Without loss of generality, we will focus our discussions on systems composed of inexpensive, high drift rate mobile sensors, and expensive but accurate stationary sensors with low drift rates. We assume that these stationary sensors provide accurate readings, either because they are inherently resistant to error or because they are maintained by experts.

For the mobile sensors, we assume only that (1) there exists an unbiased drift predictor whose residual error has Gaussian distribution and that (2) we have knowledge of how its variance increases over elapsed time since the most recent calibration event. As explained in Section 3.6.1, we observed that high-quality predictors for our sensors have this property.

Our goal is to develop a distributed technique that automatically compensates for sensor drift error; there is no notion of a central controller that has access to data from all sensors. Avoiding dependence on a central controller can reduce sensing system energy consumption, cost, and security problems.

We now present the formal problem definition. Given N mobile sensors and M accurate stationary sensors, the location of a mobile sensor i at time t is $L_i(t), i \in N$. The location of accurate stationary sensor j is $L_j, j \in M$. Sensor i's raw reading (including drift error) at time t is $r_i(t)$. Its drift prediction function is $f_i(t, k_1, k_2, ..., k_n)$. The parameters of this function may be different for each sensor and may change over time. The error associated with the drift predictor $e(t)$ changes over time. The drift-compensated sensor

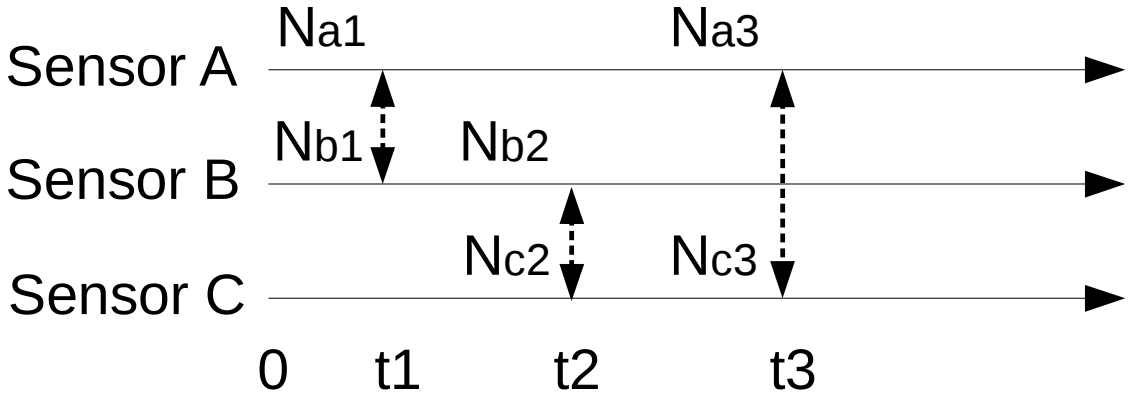


Figure 3.2: An example of sensor error correlation as a result of previous calibration events.

reading is $R_i(t) = r_i(t) - f_i(t)$. The accurate value of the monitored parameter at location l and time t is G_l^t. Let $C_i(t)$ be the post-calibration sensor reading. In other words, $C_i(t)$ is the sensor reading after drift compensation and sensor fusion. The goal is to determine k_1, k_2, ..., k_n for each sensor to minimize its total mean squared error, i.e., $\sum_t (G_l^t - C_i(t))^2$. Each sensor i at time t, only has access to $R_j(t)$ of sensor j when $|L_i(t) - L_j(t)| < D_c$ (D_c is the calibration range).

Our measurements in several rooms suggest that in well-ventilated rooms with no obvious pollution sources, the pollutant mixture is spatially homogeneous within 2 m distance. We will use this distance as calibration range D_c in simulations. Note that the spatial distributions of air pollutant concentrations vary based on nearby pollution sources and ventilation conditions, thus the calibration range depends on circumstances.

3.4.3 Error Estimation and Error Propagation

As we mentioned before, each sensor has a residual error that is adjusted after each calibration event. In this section, we describe how this residual drift error is calculated and minimized via calibration and prediction. We address the problem of predictor design for

one particular type of sensor in this paper. In general, the predictor should be provided by the sensor manufacturer or determined by pre-deployment lab calibration.

We start with a simple scenario where errors of two sensors are independent. Assume two co-located sensors A and B. Sensor A's current error estimate is n_a and sensor B's current error estimate is n_b, where n_a and n_b are random numbers with Gaussian distributions N_a and N_b and standard deviations E_a and E_b (in the rest of the paper, we use N to represent a Gaussian distribution, n to represent a random number following distribution N, and E to represent its standard deviation). Assume this is the first time sensors A and B calibrate with other sensors. N_a and N_b are independent and their standard deviations, E_a and E_b, are determined by how long the sensors remain uncalibrated. Let G be the ground truth value of the physical condition measured by the sensors. Readings from these two sensors can be represented as $R_a = G + n_a$ and $R_b = G + n_b$. The weighted sum of R_a and R_b is $R_{ab} = \alpha \cdot R_a + (1 - \alpha) \cdot R_b = G + N(0, \sqrt{\alpha^2 \cdot E_a^2 + (1 - \alpha)^2 \cdot E_b^2})$. It is easy to prove that when

$$\alpha = E_b^2/(E_a^2 + E_b^2), \tag{3.1}$$

the weighted sum has minimal standard deviation for both calibrated sensors, i.e., $G + N(0, E_a E_b/\sqrt{E_a^2 + E_b^2})$. A reading from the sensor with smaller error is given more weight. After calibration, both sensors should adjust their readings to R_{ab} and use R_{ab} to estimate their current ground truth readings as well as to predict future drifts.

Now we consider the scenario in which N_a and N_b are correlated. This may happen as a result of both sensors directly or transitively calibrating with the same mobile sensor prior to their calibration with each other. In this case, we need to know the correlation between N_a and N_b to compute the optimal combination of their readings. Let us consider the example shown in Figure 3.2. Assume three sensors A, B, and C all start operating at time 0. At time t_1, sensors A and B calibrate. Their calibration parameters

are independent of each other at that time and thus the analysis in the previous paragraph for independent errors can be applied. Assume weights of 0.2 and 0.8 are used, thus the error after calibration is $0.2n_{a1} + 0.8n_{b1}$. At time t_2, sensors B and C calibrate. Assume sensor B's drift prediction error increased by n_{b12} from time t_1 to t_2. The errors of B and C are still independent. Assume the optimal weight is 0.5 in this case. After calibration, B's and C's errors are $0.1n_{a1} + 0.4n_{b1} + 0.5n_{b12} + 0.5n_{c2}$. At time t_3, sensors A and C calibrate. A's error is now $n_{a3} = 0.2n_{a1} + 0.8n_{b1} + n_{a13}$ and C's error is $n_{c3} = 0.1n_{a1} + 0.4n_{b1} + 0.5n_{b12} + 0.5n_{c2} + n_{c23}$. Note that at that moment, these two sensors contain the same errors generated from the previous calibration, which are n_{a1} and n_{b1}. Now N_a and N_c are correlated and Equation 3.1 cannot be directly applied. However, it is still possible to use the weight assignment technique to find an optimal solution. To do that, we can remember all the independent distributions and weight assignments from previous calibration events.

Now we present the general approach that accounts for correlation introduced by transient calibration events among sensors. Each sensor's error distribution is represented as a weighted sum of multiple independent error distributions. Each independent distribution is from the other sensor's or its own increased prediction error over the uncalibrated time interval. Label the two calibrating sensors as sensor 1 and 2. Let S_1 and S_2 be the sets of independent error distributions for sensors 1 and 2. Let C be the intersection of S_1 and S_2, i.e., $C = S_1 \cap S_2$. Let C_1 and C_2 be S_1 and S_2's non-overlapping regions, i.e., $C_1 = S_1 - C$, $C_2 = S_2 - C$. Let W_{1i} and W_{2i} be the weights associated with the error distributions for sensors 1 and 2, δ_i be the standard deviation of each distribution, and G be the ground truth value of measured object. Sensor 1's reading after drift compensation

is

$$R_1 = G + \sum_{i \in C} W_{1i} N(0, \delta_i) + \sum_{j \in C_1} W_{1j} N(0, \delta_j). \tag{3.2}$$

Sensor 2's reading is

$$R_2 = G + \sum_{i \in C} W_{2i} N(0, \delta_i) + \sum_{k \in C_2} W_{2k} N(0, \delta_k). \tag{3.3}$$

In order to generate more accurate results by combining the readings of sensor 1 and 2, we use a linear weighted sum function to combine their drift-compensated measurements. Assuming the weights are α and $1-\alpha$ for sensor 1 and 2 respectively, the combined result is

$$\begin{aligned} R_{12} &= \alpha R_1 + (1-\alpha) R_2 \\ &= G + \sum_{i \in C} [\alpha W_{1i} + (1-\alpha) W_{2i}] N(0, \delta_i) \\ &\quad + \sum_{j \in C_1} \alpha W_{1j} N(0, \delta_j) + \sum_{k \in C_2} (1-\alpha) W_{2k} N(0, \delta_k). \end{aligned} \tag{3.4}$$

The variance of the error for the combined reading is

$$\begin{aligned} Var &= \sum_{i \in C} [\alpha W_{1i} + (1-\alpha) W_{2i}]^2 \delta_i^2 + \sum_{j \in C_1} W_{1j}^2 \alpha^2 \delta_j^2 \\ &\quad + \sum_{k \in C_2} W_{2k}^2 (1-\alpha)^2 \delta_k^2. \end{aligned} \tag{3.5}$$

The derivative of the variance is

$$\begin{aligned} \frac{dVar}{d\alpha} &= 2\alpha \sum_{i \in C} (W_{1i} - W_{2i})^2 \delta_i^2 + 2 \sum_{i \in C} W_{2i} (W_{1i} - W_{2i}) \delta_i^2 \\ &\quad + 2\alpha \sum_{j \in C_1} W_{1j}^2 \delta_j^2 + 2\alpha \sum_{k \in C_2} W_{2k}^2 \delta_k^2 - 2 \sum_{k \in C_2} W_{2k}^2 \delta_k^2. \end{aligned} \tag{3.6}$$

To minimize the variance, we have $\frac{dVar}{d\alpha} = 0$, therefore

$$\alpha = \frac{\sum_{i \in C} W_{2i} (W_{2i} - W_{1i}) \delta_i^2 + \sum_{k \in C_2} W_{2k}^2 \delta_k^2}{\sum_{i \in C} (W_{1i} - W_{2i})^2 \delta_i^2 + \sum_{j \in C_1} W_{1j}^2 \delta_j^2 + \sum_{k \in C_2} W_{2k}^2 \delta_k^2}. \tag{3.7}$$

Equation 3.7 gives the general expression for weight assignment. In the case of two independent sensors (C is empty), we have

$$\alpha = \frac{\sum_{k \in C_2} W_{2k} \delta_k^2}{\sum_{j \in C_1} W_{1j}^2 \delta_j^2 + \sum_{k \in C_2} W_{2k}^2 \delta_k^2} = \frac{E_2^2}{E_1^2 + E_2^2}, \tag{3.8}$$

which is consistent with Equation 3.1.

Note that the above analysis applies only to the scenario where collaborative calibration involves two sensors. It is possible to extend the evaluation to an arbitrary number of co-located sensors, although this would increase the complexity of the weight assignment expression.

3.4.4 Collaborative Calibration Algorithm

We have presented the key concept allowing the optimal calibration algorithm to combine readings from co-located sensors. Now we present the complete algorithm for collaborative calibration, which includes drift compensation, weight assignment, and drift reevaluation. Note that calibration opportunity detection is not part of our algorithm. There are multiple existing approaches to discover calibration opportunities, including radio communication (e.g., Bluetooth), ultrasound, and passive audio environment based proximity detection schemes [23, 35, 54].

The key data structure used is a table that stores all the independent error distributions and their corresponding weight assignments for each sensor. Each entry is a tuple of name, weight, and standard deviation. The names are used to distinguish independent error distributions. The calibration algorithm for a mobile sensor labeled i that calibrates with sensor j is shown in Algorithm 1.

Mobile sensors participating in the collaborative calibration system carry out three actions every time a calibration event happens: (1) estimate its current drift with its drift predictor and use the result to compensate its raw reading, (2) estimate the ground truth

value and update its error table, and (3) use the estimated ground truth value to recompute its drift, residual error, and drift predictor. The type of co-located sensor determines the details of step (2). If the co-located sensor is an accurate stationary sensor, its reading can be directly used as ground truth to estimate the mobile sensor's drift. The mobile sensor ignores its own reading and directly overwrites its own reading with the reading from the stationary sensor and its current error immediately drops to zero. As a consequence, it can forget all previous calibration errors as they become irrelevant (clear the table). Otherwise, if the co-located sensor is also a mobile sensor with a non-zero error, its drift-compensated reading is combined with the mobile sensor's drift-compensated reading according to Equation 3.7 to generate an estimate of ground truth and the error distribution table will be updated accordingly.

3.5 Stationary Sensor Placement

In this section, we consider placement of stationary sensors to further assist the collaborative calibration of mobile sensors. Our discussion will focus on human-carried sensors.

3.5.1 Overview

Adding stationary sensors to a system composed of collaboratively calibrating mobile sensors can further improve accuracy. The number of stationary sensors is constrained by cost; they must be carefully positioned to enable frequent calibration opportunities with mobile sensors. Fortunately, humans move with patterns that can be used to our benefit; some locations are more frequently visited than others [44].

Recent research has shown that most people's daily motion patterns are predictable [25, 58, 60]. We present a stochastic human mobility model capable of capturing the most relevant motion patterns for the stationary sensor placement problem. The field for stationary sensor deployment is modeled as a grid in which implicit calibration may occur among

Algorithm 1 Collaborative calibration algorithm for mobile sensor i

Require: r_i // i's raw reading
Require: R_j // j's calibrated reading
Require: T_i // i's error table
Require: T_j // j's error table
Require: t // current time
  if j is accurate stationary sensor **then**
    $R_i \leftarrow R_j$
    $Di'(t) \leftarrow r_i - R_i$
    Update drift model
    T_i.clear()
  else
    Predict current drift D_i
    $R_i \leftarrow r_i - D_i$
    $T_i.insert(i.t, g(t - last\_cali\_t), 1))$
    $C \leftarrow T_i \bigcap T_j$
    $C_1 \leftarrow T_i - C$
    $C_2 \leftarrow T_j - C$
    Compute α using Equation 3.7
    $R_{ij} \leftarrow \alpha R_i + (1 - \alpha)R_j$
    Update current drift $D_i'(t) \leftarrow r_i - R_{ij}$
    Update drift model
    for $k \in C$ **do**
      T_i[k].weight $\leftarrow$ T_i[k].weight $\times\alpha$ + T_j[k].weight $\times$ (1-α)
    end for
    for $k \in C_1$ **do**
      T_i[k].weight $\leftarrow$ T_i[k].weight $\times\alpha$
    end for
    for $k \in C_2$ **do**
      T_i[k] $\leftarrow$ (T_j[k].name, T_j[k].var, T_j[k].weight $\times(1 - \alpha)$)
    end for
  end if
  last_cali_t $\leftarrow$ t

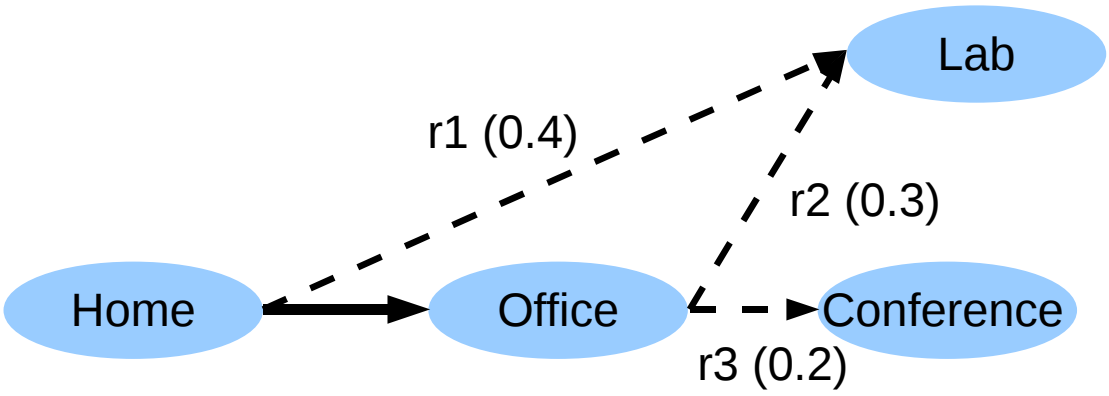


Figure 3.3: Example human motion trace with 3 patterns.

sensors in the same grid element. It is possible to eliminate discretization problems by making grid elements arbitrarily small and permitting calibration between nodes in multiple grid elements within the calibration distance. We define a motion pattern as a set of locations (grid elements) that a person is likely to visit on a particular day. An individual's mobility model is a probability-weighted collection of possible motion patterns. Extreme sensor drift typically occurs on a timescale of days, not hours, enabling a simplified model that neglects the order of visited locations within a single day. In our evaluation, these models are extracted from measured motion traces as well as those generated by software provided by human motion pattern researchers [44].

Daily motion patterns are weighted with probabilities. For example, as shown in Figure 3.3, there are three distinct patterns: r_1, r_2, and r_3. A value ranging from 0 to 1 is associated with each pattern to indicate its probability. It is possible for multiple stationary sensors to be encountered by a person in a day. However, encountering one is sufficient for calibration.

3.5.2 Sensor Placement Problem Definition and MILP-Based Solution

We now define the problem of stationary sensor placement to assist calibration of mobile sensors.

Problem Definition: The field for stationary sensor deployment can be represented by a grid G. A set of people S move within the grid. Each person $s \in S$ carries a mobile sensor. A person's motion pattern for a particular day, r_s, is a set of locations. R is the set of all motion patterns, and the motion patterns associated with a particular person s are represented with R_s. Each motion pattern r is associated with a value p_{sr}, which is the probability of person s having pattern r. The sum of the calibration probabilities of all patterns of person s is P_s. A total number of k sensors are deployed in the field. The optimization objective is to find a set of grid elements in which stationary sensors should be placed to maximize the average daily probability of mobile sensor calibration, i.e., $\frac{\sum_{s \in S} P_s}{k}$.

This problem is $\mathcal{NP}$-hard. Let each pattern be represented by an element associated with a probability weight and each possible stationary sensor placement location be represented by a subset. An element belongs to a subset if and only if the corresponding pattern contains the placement location. Given a resource constraint, k, the original problem can be stated as selecting at most k subsets such that the covered elements have maximum total weight. This is the weighted maximum coverage problem [38]. We will now describe an MILP formulation for the problem.

Maximize

$$\frac{\sum P_s}{k}, \forall s \in S,$$

subject to

$$\sum_{(i,j)\in G} x_{ij} \leq k, \tag{3.9}$$

$$\forall r \in R, \sum_{(i,j)\in r} x_{ij} - M d_r \leq 0, \tag{3.10}$$

$$\forall r \in R, \sum_{(i,j)\in r} x_{ij} - m d_r \geq 0, \tag{3.11}$$

$$P_s - \sum_{r\in R_s} d_r * p_{sr} = 0, \tag{3.12}$$

$$1 \geq x_{ij}, \text{ and } d_r \geq 0. \tag{3.13}$$

x_{ij}, d_r are integers. M and m are constants and are set to $k+1$ and 0.5. The probabilities p_{sr} are known. The properties of binary indicators x_{ij} and d_r are described below.

$$x_{ij} = \begin{cases} 1 & \text{if a sensor is placed at grid element } (i,\, j) \\ 0 & \text{otherwise,} \end{cases} \tag{3.14}$$

and

$$d_r = \begin{cases} 1 & \text{if pattern } r \text{ is covered by at least one sensor} \\ 0 & \text{otherwise.} \end{cases} \tag{3.15}$$

M is greater than the largest possible value of $\sum_{(i,j)\in r} x_{ij}$ (which is satisfied by setting M to be $k+1$) and m is less than the smallest possible non-zero value of $\sum_{(i,j)\in r} x_{ij}$ (which is satisfied by setting m to be 0.5).

3.5.3 Approximation Algorithm Based Placement Technique

Normally MILP-based solutions are not tractable for large instances of hard problems. Fortunately, the number of patterns per person is limited: it is possible to directly use the MILP formulation for substantial problem instances. The solver performance is further

Algorithm 2 Approximation based placement technique

Require: G // deployment field grid
Require: R // set of all patterns
Require: P // probabilities
Require: k // stationary sensor count constraint
  $C \leftarrow \{\}$ // output set
  while size(C) $\leq k$ **do**
    Select $g \in G$ s.t. $\sum_{r \in g} P_r$ is maximized
    Remove the covered patterns from R
    $C \leftarrow C \cup g$
  end while

improved because human motion traces tend to be spatially clustered [44]. We will show in Section 3.6.3 that our algorithm can be applied to deployment cases with up to 840 km^2 area or 200 patterns. It is conceivable that some problem instances will exceed the size tractable for MILP solvers. Therefore, we also present an approximation algorithm based polynomial time heuristic.

The maximum coverage problem can be solved with the polynomial time $(1 - \frac{1}{e})$-approximation algorithm shown in Algorithm 2. This is minimum achievable bound [38]. However, the $(1 - \frac{1}{e})$-approximation bound only applies for the average calibration probability between stationary and mobile sensors. There are many other factors influencing the network sensing accuracies, such as collaborative calibration events, calibration time, and calibration order. Section 3.6.3 evaluates the approximation algorithm based technique in detail.

3.6 Experimental Results

In this section, we first describe our controlled drift experiments (Section 3.6.1), which support the hypothesis in Section 3.4.2. Section 3.6.2 presents simulation results for our optimal and efficient collaborative calibration techniques and compares them with two existing works that are most related. Section 3.6.3 reports on the performance of our MILP based stationary sensor placement algorithm and compares it with the efficient approxi-

Figure 3.4: Calibration chamber used for sensor drift experiments.

mation algorithm we propose.

3.6.1 Calibration Procedure and Drift Experiments

Section 3.4.2 describes our sensor drift model. We assume that drift can be (partially) compensated for by an unbiased predictor, and the residual error can be modeled using a Gaussian distribution with a variance that predictably increases with time. To test this hypothesis, we have conducted a drift experiment in our controlled chamber.

Before the drift experiment, we manually calibrated all sensors. Calibrations were performed using de-humidified zero grade air (i.e., air with less than 1 ppm total hydrocarbons) and controlled-concentration iso-butylene (a VOC unlikely to damage graduate students when used at low concentration). The purpose of this calibration is to compensate for ini-

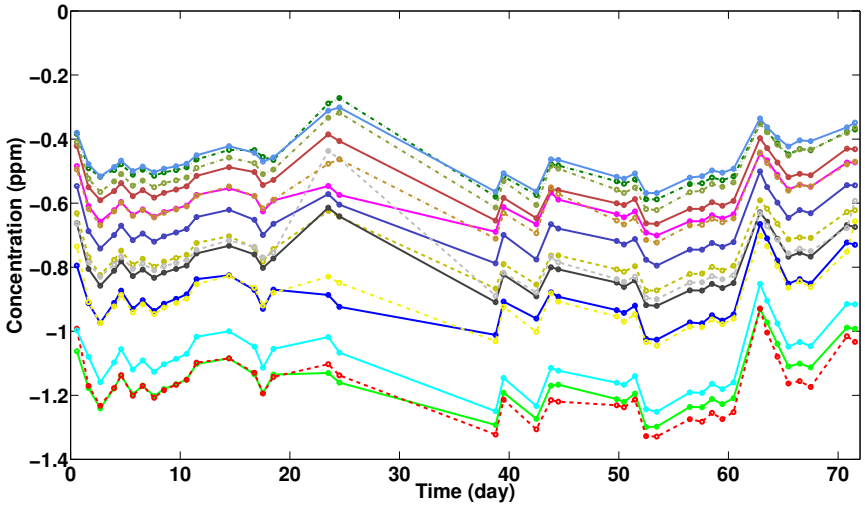


Figure 3.5: Measured drift error as a function of time for Figaro TGS2602 VOC sensors.

tial measurement offsets, possibly due to variation in the manufacturing process. During calibration and drift experiments, sensors are mounted on a custom printed circuit board enclosed in the 250 cm^3 polycarbonate chamber as shown in Figure 3.4. A fan is mounted inside the chamber to improve mixing and make convection heat loss from the sensors uniform. The temperature and humidity inside the chamber are stabilized at $43.8 \pm 1.3\,^{\circ}\mathrm{C}$, and $7.8 \pm 1.7\%$ respectively. A LabVIEW interface controls the gas mixture using mass flow controllers. During calibration runs, the sensors are held at concentrations of 0, 0.25, and 1.0 ppm (parts per million by volume) of iso-butylene in a total volume flow of 4 liters per minute, for 20 minutes each. The sensors are powered continuously throughout the experiment period, and were warmed up for two weeks prior to starting the experiments to allow the sensors to reach an initial equilibrium, as recommended by the manufacturer.

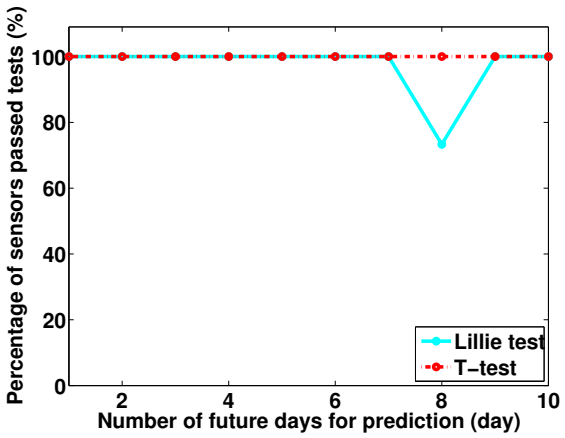
During the drift experiment, 15 pre-calibrated Figaro TGS 2602 VOC sensors are placed in the controlled gas chamber and exposed to 4 liters per minute air. These exposure tests last 120 minutes and are performed daily. Since the sensors are powered continuously, they should drift constantly during the experiment. The drift data are calculated by averaging the last 30 minutes of readings from each test to avoid any warm-up

effects from changes in the air flow rate.

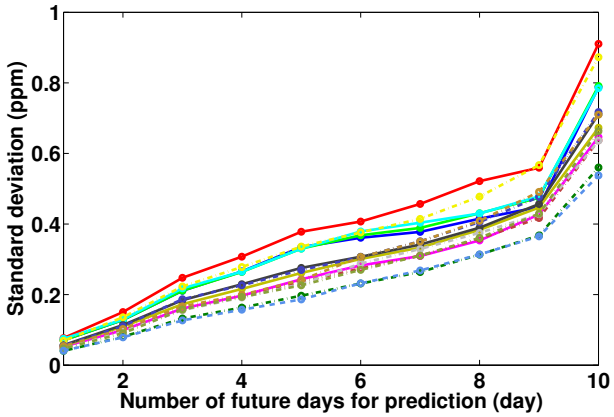
We use the analog to digital converter on Labjack U3 data acquisition modules to measure the voltage output of the TGS sensors, at a sampling frequency of 0.5 Hz. We use log-based transfer function to convert the voltages to VOC concentrations, based on calibrations performed before the experiment. The concentration readings after conversion are shown in Figure 3.5. Since the ground truth reading should be 0 ppm, the readings after the conversion already represent drift. Seven of the 48 measurements were discarded due to inconsistent air flow rate or relative humidity levels due to transient problems with the testing chamber air supply.

We now evaluate a simple drift predictor based on linear extrapolation of two consecutive drift errors to predict future errors. The difference between the predicted drift value and the measured drift is the portion of the drift error that is not captured by the drift model. We have also evaluated higher-order non-linear predictors but they did not have higher prediction accuracies than the linear predictor. The linear predictor compensated for 94.1% and 87.7% of the drift on average when predicting one day and two days ahead. We therefore consider it to be a good predictor for this kind of sensor. Note that for different sensor types, the forms of the predictor function may be different. In some cases, a higher order non-linear fitting function might be necessary.

We applied the Lillie normality test to the residual error of the linear predictor. The residual error has a Gaussian distribution, with an exception for predictions eight days in advance. For most cases, the linear predictor meets Gaussian residual requirement posed in Section 3.4.2. For specific sensors and time offsets passing the normality test, we perform t-tests to assess whether the distributions have means of 0 ppm. The significance levels used in the Lillie test and t-test are both 0.05 and the test results are shown in Figure 3.6(a). Figure 3.6(b) shows the standard deviation of the remaining drift error after applying the



(a)



(b)

Figure 3.6: (a) The normality test results and (b) the standard deviations of prediction errors using the 2-day linear predictor to compensate for 1 to 10 days of future drift.

linear predictor for up to 10 days in the future. The results clearly show an increasing trend for all the sensors, consistent with our hypothesis in Section 3.4.2 that the variance increases over time. The standard deviations of the short-term drift errors can be well predicted using simple linear functions.

With one possible anomaly at an eight-day offset, the drift experiment results confirm our hypothesis that the residual error after drift prediction has a Gaussian distribution with mean 0 and predictable variance that increases over time.

Table 3.1: Aggregated Sensor Error with Synthesized Human Motion Traces

Trace	Num. of cali. events			Total aggregated mean squared error			
	Total	Uncorrelated	Stationary	CaliBree	Averaging	Heuristic	Optimal
1	44,290	5,072	21,818	964.6	393.6	321.9	312.1
2	43,378	3,368	20,144	1,716.6	559.0	454.9	434.8
3	9,701	1,722	4,429	3,059.0	1,461.1	1,244.3	1,229.8
4	5,659	1,048	2,589	6,805.8	2,359.6	1,984.0	1,966.3
5	14,308	2,496	4,398	8,610.6	3,234.7	2,681.8	2,643.6
Average overhead (%)				224.8	23.2	2.2	0

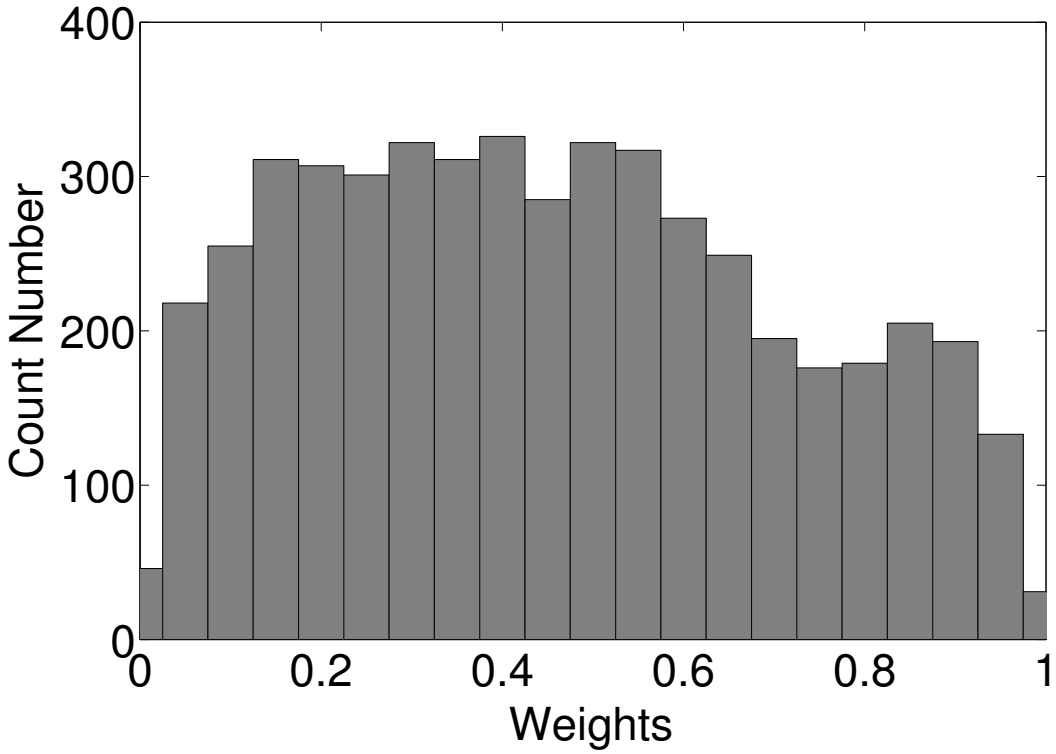


Figure 3.7: Histogram of assigned weights for an example trace using the optimal collaborative calibration scheme.

3.6.2 Evaluation of Collaborative Calibration

To evaluate our collaborative calibration algorithm, we compare it with two other approaches proposed in relevant and recent work. In the first approach, Calibree [49], all mobile sensors calibrate with stationary accurate sensors. In contrast, our calibration technique allows sensors to calibrate with each other as well as stationary sensors. In

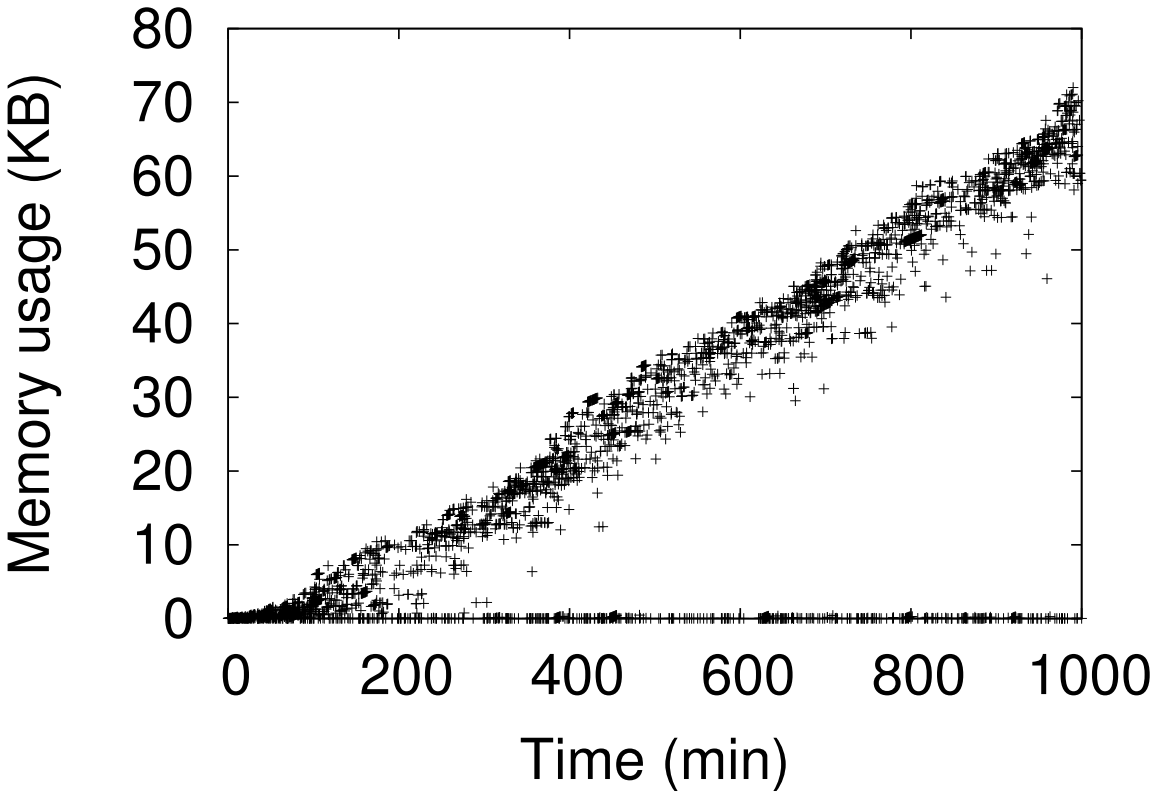


Figure 3.8: Memory use of the optimal collaborative calibration scheme.

the second approach [65], readings from co-located sensors are averaged to estimate the ground truth value. In contrast, our technique enables more accurate drift compensation by considering the differing drift prediction errors of calibration events, i.e., sensors. We also propose and evaluate a *calibration heuristic* that reduces computation complexity and memory use at the cost of a very slight reduction in calibration accuracy. This heuristic ignores correlations between prediction errors. Instead tracking independent error distributions from previous calibration events and temporal error growth, this algorithm only stores an aggregated error for each sensor. During calibration, it uses Equation 3.1 to assign weights to readings from co-located sensors. We evaluate the four approaches with the same set of motion traces and sensor placements, and compare the resulting accumulated mean squared error. For this experiment, we use 10 stationary accurate sensors placed at the most frequently visited locations and use a random walk model for sensor drift.

Section 3.1 shows the results for the four approaches with five synthesized motion traces generated using the SLAW human mobility model [44]. The second to the fourth columns present statistics for calibration events for the optimal algorithm. The second column shows the total number of calibration events. A pair-wise calibration between two sensors is considered to be two calibration events. The third column shows the number of calibration events in which the errors from two sensors are independent. The fourth column shows the number of calibrations with stationary accurate sensors. The last four columns show the aggregated mean squared errors of all sensors during the entire experiment.

On average, CaliBree [49] has 224.8% more error than optimal. This is because it only considers calibration events between stationary and mobile sensors, and thus misses opportunities for calibration between mobile sensors. 43.6% of calibration events occur between mobile and stationary sensors; the rest occur between pairs of mobile sensors.

Tsujita's technique (averaging) has 23.2% more error than optimal result. Figure 3.7 shows the distribution of the weights generated with the optimal algorithm for Trace 5. The weights are widely distributed from 0 to 1. Only 25.4% are in the range from 0.4 to 0.6. The structure of this histogram has implications for the effectiveness of Tsujita's approach: the closer weights are to 0.5, the more effective Tsujita's approach.

Our heuristic produces results with accuracy that deviates from optimal by only 2.2%. Even though the percentage of correlated events is fairly large (41.8%), ignoring the correlation does not significantly degrade accuracy. However, this algorithm greatly reduces required memory compared with the optimal algorithm. With the optimal algorithm, the memory use increases linearly with time for most sensors. Figure 3.8 shows the memory use over time for all sensor nodes in our experiment with trace 1. Each point corresponds to a sensor node involved in a calibration event. We therefore conclude that the heuristic is more efficient and likely to be appropriate for most practical applications.

Table 3.2: Statistics for Human Mobility Case Study

Participant	Duration (days)	On campus prob. (%)	# of patterns	# of locations
1	30	90.0	12	11
2	30	86.7	5	5
3	22	77.3	4	4
4	23	100.0	5	4
5	21	76.2	7	6
Average	25.3	85.2	6.6	6

The optimal algorithm allows us to evaluate the quality of various calibration approaches. In summary, utilizing the interactions among mobile sensors improves the accuracy by 224.8% compared to only permitting mobile sensors to calibrate with stationary sensors. The accuracy is improved by 23.2% by considering the heterogeneity of drift estimation parameters among different sensors. Considering correlations among sensors due to calibration imposes large computation complexity and memory use with a relatively small gain (2.2%). In summary, a technique using collaborative calibration among mobile sensors that considers heterogeneity in drift estimation parameters but ignores calibration event induced inter-sensor correlations represents a good trade off between accuracy and run-time overhead/complexity.

3.6.3 Evaluation of Stationary Sensor Placement

This section introduces our human motion pattern case study and evaluates our stationary sensor placement algorithms with both measured and synthesized human mobility traces.

Measured Human Mobility Case Study

Much human mobility modeling research is based on outdoor GPS data [25, 44, 60]. However, GPS is inaccurate indoors, where humans spend 90% of their time [22]. According to a survey-based model, office worker indoor activities can be modeled using a

Table 3.3: Statistics for Measured and Synthesized Human Motion Traces and Solver Performance

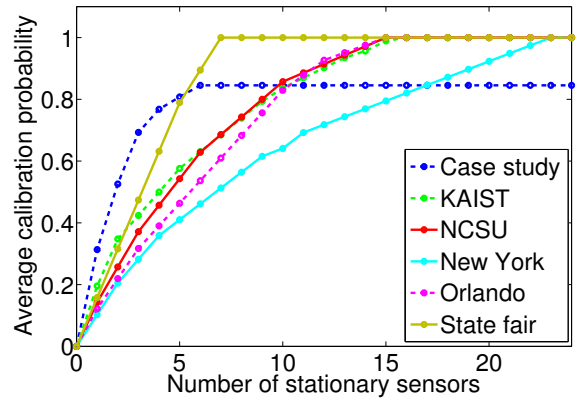
Trace	Area (km^2)	Total pat.	Sensor no.	Cand. loc.	Runtime (s)
Case study	N/A	33	5	17	0.01
KAIST	840.1	92	92	41,270	1.2
NCSU	142.3	35	35	10,691	0.13
New York	618.8	39	39	12,180	0.05
Orlando	122.0	41	41	26,662	0.07
State fair	1.2	19	19	4,422	0.03
1	0.01	200	50	1,225	0.13
2	0.01	200	50	1,001	0.24
3	1.0	200	50	26,448	2.44
4	1.0	200	50	39,695	5816.10
5	4.0	400	100	101,891	¿ 6 h

few patterns [39]. In our evaluation, we use mobility traces generated using algorithms proposed by other researchers as well as data gathered in our real-world human mobility study, which was conducted on the campus of University of Colorado Boulder.

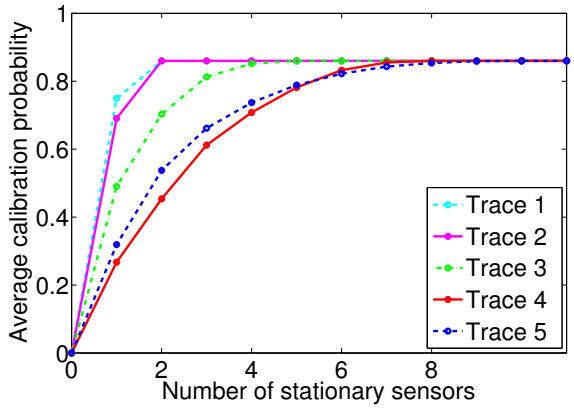
In our study, five graduate students, undergraduate students, and professors used their mobile phones to record their daily motion patterns. Participants manually entered locations and times into their smart phones as they moved and these data were sent to a server via the Internet. Locations in which users spent fewer than five minutes were omitted from the motion patterns. The study was conducted between August 3rd, 2011 and September 12th, 2011. Statistics from the study are shown in Table 3.2. Motion patterns contain 1.94 locations on average, which implies that the indoor activities of the participants were spatially concentrated, which is consistent with the findings of other human motion studies [39, 60].

Experiment on Measured and Synthesized Human Motion Traces

To solve the MILP problem, we use the CPLEX v.12.2 solver [32] on an Intel 4-core Xeon E31230 CPU running at 3.2 GHz with 8 GB of memory. The evaluation is performed



(a)



(b)

Figure 3.9: The MILP stationary sensor placement results for (a) measured human motion traces and (b) synthesized human motion traces.

on both real-world and mobility model generated [44] human motion traces.

The statistics of the real-world and synthesized human motion traces [44], as well as our case study trace, and their MILP solver performances are shown in Table 3.3. The case study trace does not contain detailed location information, but lasts for multiple days. The rest of the real-world traces contain detailed location information, but are finished within a day each, i.e., each person has one motion pattern. The duration for each trace is 4 days, i.e., each person has 4 patterns. According to our real-world case study, the average probabilities of the top 4 patterns are 0.48, 0.2, 0.1, and 0.08. The same probability values are used in the synthesized traces. The fourth column of the table shows the total number

Table 3.4: Aggregated Sensor Errors for Different Placement Algorithms

Trace	Sensor number			Aggregated error			
	MILP	Approx. Algo.	Improvement	All Mobile	MILP	Approx. Algo.	Improvement
KAIST	16	19	18.8%	9,880	7,875	8,465	7.5%
NCSU	15	15	0.0%	6,075	3,095	3,333	7.7%
New York	23	26	13.0%	4,720	2,076	2,504	20.6%
Orlando	15	16	6.7%	7,208	3,683	3,954	7.4%
State fair	7	7	0.0%	5,303	2,649	2,786	5.2%
1	2	2	0.0%	910	523	551	5.4%
2	2	3	50.0%	1,083	701	738	5.3%
3	5	5	0.0%	2,326	1,783	1,831	2.7%
4	8	9	12.5%	3,370	2,522	2,511	-0.4%
5*	10	11	10.0%	3,924	3,195	3,205	0.3%

*The MILP solution is derived by setting the relative tolerance of the MILP solver to be 0.3%.

of mobile sensors in each trace. The fifth column shows the total number of candidate locations where stationary sensors may be placed. Grid elements visited by one or more person are considered as placement location candidates. The total number of the candidate locations is equal to the number of variables x_{ij} in Equation 3.9.

The MILP placement algorithm quickly solves all the problem instances, except for synthesized trace 5. For this trace, the solver terminated after six hours without producing a solution. This trace contains 400 patterns and 101,891 candidate placement locations. We conclude that the MILP solution is suitable for many useful-scale problem instances, but there may be some real-world cases for which a more efficient solution is required, e.g., that in Section 3.5.3.

The results of the MILP placement algorithm are shown in Figure 3.9. For most of the solutions, the number of sensors is far less than the number of patterns. This is consistent with the hypothesis that people's motion traces tend to be clustered, repetitive, and frequently overlap each other. The synthesized human motion traces typically required

fewer sensors despite having more motion patterns because a relatively small geographical area was considered in these traces. In summary, although personal mobile sensors are needed to monitor the conditions experienced by many individuals, the accuracy of these sensors can be improved substantially by using a few accurate stationary sensors to assist a collaborative calibration technique.

The results of evaluating the algorithms on both real-world and synthesized human motion traces are shown in Table 3.4. We assume that repeated calibration with a stationary sensor during the same day does not further reduce error. The aggregated network error (the sum of mean square errors of all the sensors in the network for readings taken every 30 seconds) is measured when both placement algorithms are permitted to use the number of stationary sensor listed in the second column of Table 3.4. For the synthesized traces, we assume that all the patterns occur with the same probability. The fifth column of Table 3.4 shows the aggregated network error using our optimal collaborative calibration technique, assuming there are no stationary sensors. The results show that the approximation algorithm based technique increases aggregated network error by 6.2% compared to the MILP placement algorithm. Note that for Trace 4, the approximation algorithm based technique outperforms the MILP solution. In that case, the approximation algorithm had already reached 99% average calibration probability, making its solution essentially equivalent to the MILP solution. Note that in our placement problem formulation, the error caused by calibration order is neglected. However, since the uncompensable drift error within a day is small (less than 0.1 ppm as shown in Figure III.6(b)), this simplification has very little impact on solution quality.

3.7 Conclusions

We have presented a collaborative calibration and sensor placement framework for mobile sensor networks. We have developed a random sensor drift model based on controlled experiments and developed a collaborative calibration technique to compensate for drift error. We have also described placement techniques for stationary sensors used to augment collaborative calibration among mobile sensors. Experimental results indicate that, compared with our collaborative calibration algorithm, the most advanced existing work has an average sensor error of 23.2%. Our stationary sensor placement algorithms further reduce the effects of drift error.

CHAPTER IV

Hybrid Sensor Network Modeling and Synthesis

4.1 Introduction

In Chapter III, we have described a collaborative calibration technique to address the sensor drift problem. In that work, arbitrary number of stationary and mobile sensors can be included in the network. However, in the real-world applications, the number of sensors are usually constrained by cost. Therefore, in this work, we investigate the possibility of using both mobile and stationary sensors for indoor air quality monitoring and maximizing the accuracy of the network under cost constraint. It should be noted that our techniques can be easily extended to outdoor applications.

Indoor air quality is important. People spend more that 90% of their time indoor. Moreover, pollutant concentrations are usually much higher indoors than outdoors. Many indoor pollutants are closely related to various diseases, cancers, and human mortality [27, 55]. Other less dangerous indoor pollutants, such as carbon dioxide (CO_2), can have significant impact on office worker and students productivity, performance, and health [59, 64].

Indoor pollutant distribution can be very dynamic and heterogeneous. Indoor pollutant concentrations may vary significantly even within the same building, e.g., indoor VOC concentrations can differ by more than 7 times for different rooms in a same building [47].

Thus, a sensor network composed of a few stationary sensors is inadequate to estimate the indoor personal pollutant exposure.

The mobile sensors are susceptible to drift and require frequent calibrations. Comparing with the opportunistic collaborative calibration technique, calibrating with accurate stationary sensors is more predictable and accurate. However, the high cost of those sensors limits their number, which in term can reduce the calibration opportunities. Given a fixed budget, one must trade off the (cost constraint) inaccuracies of stationary sensor networks with the (drift) inaccuracies of mobile sensor networks.

We propose a hybrid sensor network architecture composed of both accurate stationary sensors and inaccurate mobile sensors. Stationary sensors can provide accurate readings and more importantly, calibration opportunities for the mobile sensors. Mobile sensors carried by individuals can measure more relevant personal exposure data. Note that although our technique focuses on the hybrid sensor network architecture, it is also capable of designing mobile-only or stationary-only sensor networks.

The purpose of this work is to provide a comprehensive solution for hybrid air quality sensor network architecture analysis and construction. Network performance analysis is challenging because it is difficult to predict actual concentrations given only readings from other locations and drift-influenced readings. The challenge for network synthesis is to maximize accuracy via sensor selection and allocation given a fixed budget. Our work addresses both analysis and synthesis problems.

This work makes the following contributions:

1. we formulate the problem of indoor pollutant concentration estimation and propose an optimal solution taking into account of sensor inaccuracies;

2. we describe algorithms for automatically designing hybrid sensor networks;

3. we demonstrate how to use real-world CO_2 measurement data to estimate the airflow inside a building, and use these estimates to evaluate our analysis and synthesis techniques.

To the best of our knowledge, this is the first work addressing the problem of optimal concentration prediction with inaccurate sensors and automated design for hybrid (mobile /stationary) air quality sensor networks.

The rest of this chapter is organized as follows. Section 4.2 discusses previous related work. Section 4.3 provides a motivating example and gives an overview of our analysis and synthesis system. Section 4.4 describes models to predict the indoor pollutant concentration optimally and estimate the prediction error. Section 4.5 presents algorithms to select and allocate different types of sensors to minimize average sensor network error. Section 4.6 describes our deployment and evaluation results.

4.2 Related work

This section summarizes the prior works on sensor network architecture, indoor environment modeling, and sensor noise reduction.

Sensor network architecture. Postolache et al. [53] described an ad hoc sensor network for indoor and outdoor air quality monitoring. Jiang et al. [36] described MAQS, a mobile environmental sensing network utilizing portable, indoor location tracking sensors. Common Sense [68], designed by Willett et al., tried to establish an environmental sensing network based on the response from communities. The placement problem of stationary sensors has also been well studied [7, 13]. Krause et al. [42] proposed a sensor placement algorithm based on sensing quality and communication cost prediction. In their approach, the sensor nodes are all stationary, while we consider both stationary and mobile nodes. Recently, Xiang et al. [61] proposed a mixed integer linear programming based place-

ment algorithm for stationary sensors in a hybrid sensor network. Our technique differs from previous work in that we consider both stationary and mobile sensors in our network design and exploit the cost and accuracy trade-off between them. Moreover, in contrast with prior work, we assume no prior knowledge of the types and quantities of sensors or the carriers of the mobile sensors. Instead of relying upon an established sensor network architecture, we describe how to construct hybrid sensor networks from scratch.

Indoor environment modeling. The single compartment mass balance-based model, developed by Hayes [29,30], is widely used in modeling indoor pollutant distributions [24, 26, 48] and was validated using real-world measured data [14]. Liu et al. [45] gave a detailed description of the model and used it together with a probability-based adjoint inverse method to back-track indoor pollution sources. In this work, we build an extended model based on the mass balance-based model. In most prior work, it is assumed that the readings reported by the sensors are always accurate, and the mass balance model is mainly used to interpolate the pollutant concentrations at the locations without sensors. However, this assumption is not true in real-world applications using low-cost sensors. We extend the current model by considering and optimally compensating for the drift error.

Sensor noise reduction. One major problem for the low-cost sensors is their unreliable readings caused by long-term drift. To reduce the sensor noise, Tsujita et al. [65, 66] proposed using accurate stationary sensors to calibrate mobile sensors. Bychkovskiy et al. [12] proposed a two-phase post-deployment calibration technique. Miluzzo et al. [49] proposed an auto-calibration algorithm for mobile sensor networks. Elnahrawy et al. [20] described a sensor noise cleaning framework based on Bayes' theorem. In this work, we evaluate the impact of sensor noise to the synthesis and construction process of sensor networks. In contrast with prior work, our model incorporates indirect observations, i.e., concentration levels of adjacent locations, and thus improves accuracy of the network.

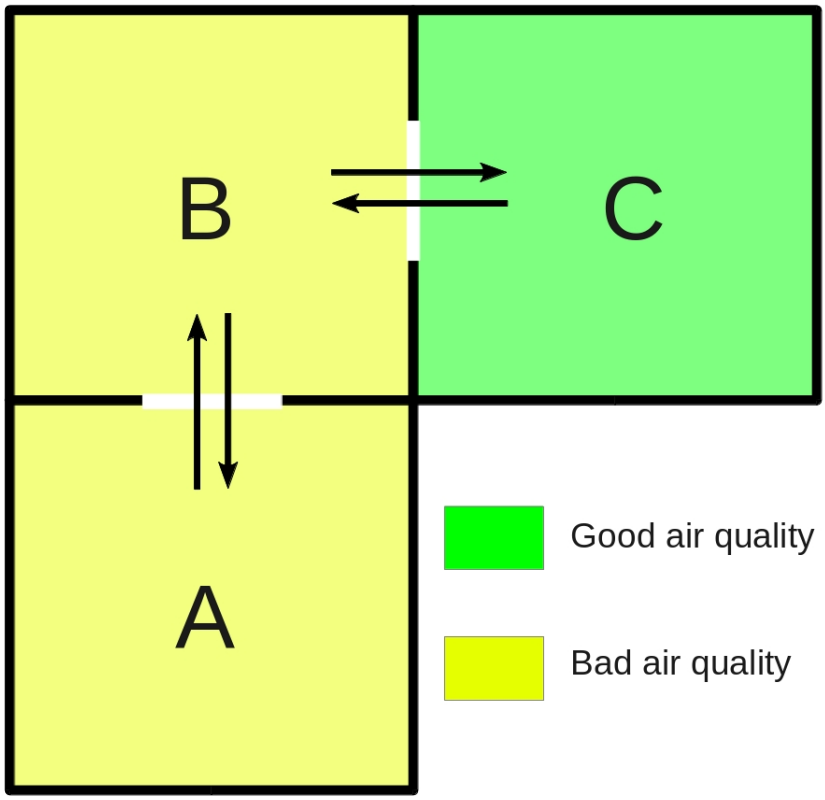


Figure 4.1: Motivating example.

4.3 Motivation and System Overview

In this section, we present a motivating example and give an overview of our hybrid sensor network analysis and synthesis system.

4.3.1 Motivating Example

This example describes the previously unsolved indoor pollutant concentration estimation and sensor network construction questions that motivate our work. The rest of this paper will provide answers to the questions appearing in this section.

Assume that a research team wants to deploy a small sensor network in the building as shown in Figure 4.1. The building contains 3 rooms: A, B, and C. All of the rooms are connected and hence have airflow between them. Assume that the budget is limited and

the team can only afford one accurate sensor, which is placed in room A. The first question is, **"How should the pollutant concentrations in rooms B and C be predicted based on the reading in room A?"**

Then a somewhat inaccurate sensor is placed in room B. Suppose that one day the sensor reports a reading of 0.8 parts per million (PPM) pollutant concentration, while the estimation based on A's measurement suggests that the concentration in room B should be 0.5 PPM. The second question is, **"How can these two estimates be reconciled to minimize the expected value of error?"**

Given a method of estimating pollutant concentrations, the problem of determining the numbers and types of mobile and stationary sensors remains. Subject to budget constraints, there are multiple options. One might deploy one stationary sensor and four mobile sensors, or two stationary sensors and two mobile sensors. The third question is, **"How should the numbers, types, and positions/carriers of sensors be determined to minimize the expected value of personal pollutant exposure error?"**

In this work, we aim to answer the three questions considered above. The first two questions led us to develop an optimal pollutant concentration prediction model based on analysis of indoor airflow and knowledge of pollutant source generation rate and sensor drift distributions. The third question led us to develop a hybrid sensor network synthesis algorithm that considers human mobility patterns and sensor costs.

4.3.2 Hybrid Sensor Network Synthesis System Overview

Figure 5.1 shows the overview of our hybrid sensor network synthesis system. The system has two major components: the concentration prediction model and synthesis algorithm. The concentration prediction model takes as inputs pollutant source generation rate distributions, sensor drift distributions, and sensor architecture information. By us-

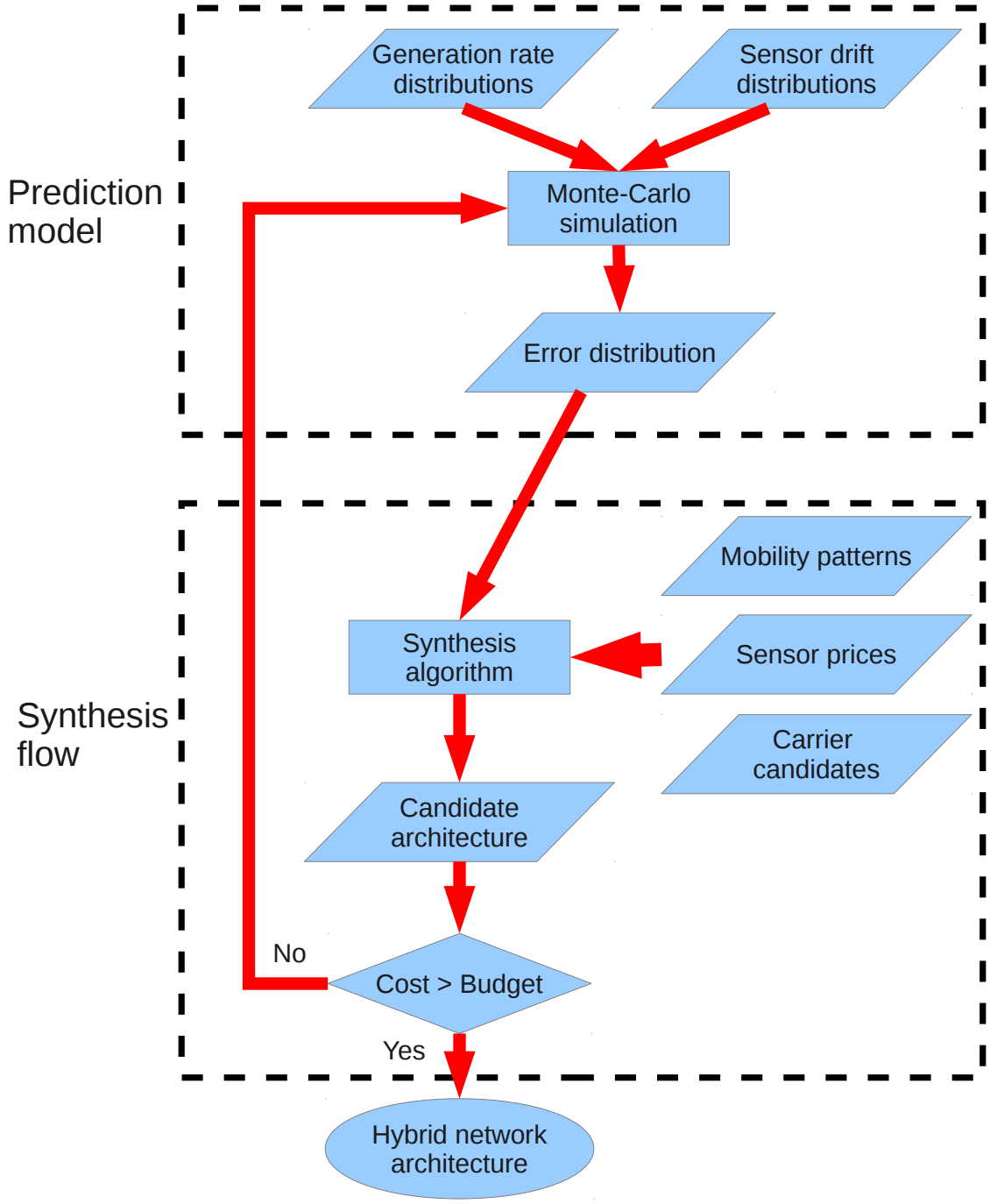


Figure 4.2: Hybrid sensor network synthesis system overview.

ing Monte-Carlo simulation, it can provide the concentration predictions and calculate the estimation errors.

Given the estimation error information, the sensor network synthesis flow searches

for a hybrid sensor network architecture that minimizes pollutant concentration estimation error. The synthesis algorithm requires the estimation error distributions, the motion patterns of individuals in the building, the prices of the available sensors, and the set of sensor carrier candidates as inputs. It searches the design space and records the solution with minimal error. The result is a hybrid network sensor architecture, including the types and quantities of sensors as well as their locations (for stationary sensors) and carriers (for mobile sensors).

4.4 Pollutant Concentration Prediction Models

In this section, we describe the design of an optimal pollutant concentration prediction model. Section 4.4.1 gives a problem definition. Section 4.4.2 introduces concentration and error estimation models. Section 4.4.3 describes the optimal model.

4.4.1 Problem and Term Definitions

The deployment field, which is typically a building, is divided into multiple zones with inhabitants moving inside. Within the same zone, the pollutant distribution is well-mixed and uniform. This can be achieved by subdividing zones when necessary. Depending on the pollutant type and ventilation conditions, a zone can be part of a room, an entire room, or multiple closely connected rooms.

A sensor network is deployed in a building so that a subset of the zones are covered, i.e., contain sensors. There are two potential causes of inaccurate concentration predictions. First, it is necessary to (imperfectly) estimate the pollutant concentrations of zones that are not covered. Second, sensor readings for covered zones may be inaccurate due to drift. We describe a model that takes into consideration both error sources and minimizes the expected value of prediction error.

We now define error. The error of the estimated concentration for zone i, denoted

as e_i, is the difference between the predicted concentration and the ground truth. Since the estimation error is a random number, it can not be used to directly evaluate models. Therefore, we use expected error, which is the standard deviation of the distribution that e_i follows, as the evaluation criteria. The expected error is denoted as E_i, and its relationship with estimation error e_i is

$$E_i = \text{std}(e_i). \tag{4.1}$$

Thus, an optimal pollutant concentration prediction is the concentration estimation with the minimal expected error.

The multi-zone pollutant concentration modeling problem can be defined as follows. Assume knowledge of the following deployment field information: inter- and intra-zone airflow, ventilation conditions, corresponding human motion patterns, pollutant source generation rates, and sensor drift information. A sensor network architecture, i.e., the types and quantities of the sensors, the locations of the stationary sensors, and the carriers of the mobile sensors, is deployed. Find a model to estimate the pollutant concentrations of all zones in the field so that the average expected error is minimized.

4.4.2 Pollutant Concentration Modeling and Analysis

In this section, we discuss concentration prediction models given various deployment schemes.

Concentration Estimation without Sensors

Assume that we want to evaluate the pollutant concentrations of all the zones in a building where no sensor is deployed. In general, the dynamic concentration change rate

can be modeled using the following multi-zone pollutant transport equation [45].

$$\begin{aligned}\frac{dC_i}{dt} &= \left[\sum_{j=1,\neq i}^{n}\left(\frac{F_{j,i}(1-\eta_{j,i})}{Q_i}\cdot C_j\right) - \frac{\sum_{j=1,\neq i}^{n} F_{i,j}}{Q_i}\cdot C_j\right] \\ &+ \left[\frac{s_i}{Q_i} + \frac{F_{i,0}\cdot C_0}{Q_i}\right] \\ &= \sum_{k=1}^{n} a_{ik}\cdot C_k + B_i. \end{aligned} \tag{4.2}$$

The coefficients in Equation 4.2 are

$$a_{ik} = \begin{cases} \frac{-\sum_{j=1,\neq i}^{n} F_{i,j}}{Q_i} (k=i) \\ \frac{F_{k,i}(1-\eta_{k,i})}{Q_i} (k\neq i) \end{cases}, \tag{4.3}$$

$$B_i = \frac{s_i}{Q_i} + \frac{F_{i,0}\cdot C_0}{Q_i}, \tag{4.4}$$

where C_i is the concentration for zone i, C_0 is the outdoor concentration, n is the total number of zones, $F_{i,j}$ is the airflow rate from zone i to j, $F_{i,0}$ is the net airflow rate between zone i and outdoor environment, $\eta_{i,j}$ is the efficiency of the pollutant filters in the heating, ventilation, and air conditioning (HVAC) system, Q_i is the air volume in zone i, and s_i is the local pollutant source generation rate. Note that the airflow rate $F_{i,j}$ is directional and $F_{i,j}$ is not necessarily equal to $F_{j,i}$. In our problem formulation, we neglect the kinetic reaction among various pollutants and local removal rate. Those parameters can be easily incorporated into the model if the information about other pollutants in the air is known.

Now consider a building with n zones. The estimated concentrations for all the zones in the building can be represented by a vector $C = [C_1, C_2 ... C_n]^T$. Thus, the pollutant transport function can be re-written as

$$\frac{dC}{dt} = A\cdot C + B, \tag{4.5}$$

and

$$A = \begin{bmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{n1} & \cdots & a_{nn} \end{bmatrix}, B = [B_1, B_2, ..., B_n]^T. \tag{4.6}$$

In the rest of the paper, matrix A is referred to as the airflow matrix. This model is widely used and found to be accurate in real-world experiments [14].

For most of the pollutants, the health and/or performance impact is evaluated on a time scale varying from days to years. Moreover, if some pollutant is released and causes a sudden change in local source generation rates, the indoor environment can return to a well-mixed state quickly. For example, it takes about 80 minutes for a 238 m^3 smoke lounge to become well-mixed [41]. Therefore, in personal exposure measurement applications, the dynamic variation in Equation 4.2 can be neglected [11], leaving $\frac{dC_i}{dt} = 0$.

The equilibrium state equation for zone i can be described using the following equation.

$$\sum_{j=1}^{n} a_{ij}C_j + b_i s_i + u_i C_0 = 0, \tag{4.7}$$

where b_i equals $\frac{1}{Q_i}$ and u_i equals $\frac{F_{i,0}}{Q_i}$. In matrix form, we have

$$A \cdot C + B = 0. \tag{4.8}$$

If all the zones are in the well-mixed state, the pollutant concentration in any zone is a linear combination of the concentrations of other zones (including outdoor environment) and its own local source generation rate.

The airflow matrix A can be estimated using multiple methods. For example, Liu et al. [45] suggest that we can derive the airflow matrix by applying the following procedure: (1) build the multi-zone model for a building; (2) determine the building leakage; and (3)

incorporate the effects of the HVAC systems. After we obtain the required information and parameters, we can derive the airflow matrix by solving the corresponding computational fluid dynamics equations using tools such as CONTAM [50]. Another approach is to use the existing sensors, with the help of regression analysis, to estimate the airflow matrix. We will show in Section 4.6.1 how to use a CO_2 sensor network to derive the average airflow matrix. The first approach does not require any existing sensor infrastructure. However, it is less accurate since it relies on the empirical estimation for parameters such as building leakages.

The inter-zone airflow may vary in time as the human behavior and ventilation conditions change, e.g., doors and windows opening and closing or changes in the state of the heating system. However, it is not necessary to derive multiple airflow matrices for all the scenarios. Since the concentration relationship between zones is linear, we can use a single averaged matrix as long as the equilibrium state assumption in Equation 4.7 holds.

To solve Equation 4.7, it is also necessary to know the pollutant source generation rate s_i. However, its value can not be accurately predicted and varies according to the characteristics and locations of the zones. The uncertainties in source generation rates cause uncertainties in pollutant concentrations, making more complete coverage by sensors valuable.

To estimate the pollutant concentrations of uncovered zones, we need to estimate the source generation rates. We assume that the source generation rates follow certain distributions with known mean values and standard deviations. The knowledge of the distributions can be obtained by analyzing the historical data or existing literature for buildings with similar characteristics [10, 19, 47]. The error of the estimation can be captured and compensated for by sensors located in or near the zone.

Assume that the source generation rate distribution for zone i is

$$S_i = L(m_i, v_i), \tag{4.9}$$

where L is the type of source generation rate distribution, m_i is its expected mean value, and v_i is its standard deviation. For each zone, its actual generation rate is a random number s_i that follows distribution S_i.

The optimal generation rate prediction, for any uncovered zone i, is the mean value m_i of its distribution. Thus, when there is no sensor deployed in the building, by solving Equation 4.7, the concentration of zone i can be estimated as

$$C_i = -\sum_{j=1}^{n} a'_{ij}(b_j m_j + u_j C_0), \tag{4.10}$$

where a'_{ij} is the element of the inverse matrix A^{-1} of the airflow matrix as shown in the following equation.

$$A^{-1} = \begin{bmatrix} a'_{11} & \cdots & a'_{1n} \\ \vdots & \ddots & \vdots \\ a'_{n1} & \cdots & a'_{nn} \end{bmatrix}. \tag{4.11}$$

Given that there is no sensor deployed, Equation 4.10 predicts pollutant concentration with minimal expected error. The ground truth concentration, denoted as g_i, can be calculated as

$$g_i = -\sum_{j=1}^{n} a'_{ij}(b_j s_j + u_j C_0), \tag{4.12}$$

where s_j is the ground truth source generation rate of zone j.

By its definition, the estimation error of zone i is the difference between the predicted concentration and ground truth and can be expressed as

$$e_i = C_i - g_i. \tag{4.13}$$

Note that e_i is a random number and its standard deviation is the expected error, E_i.

By replacing C_i and g_i in Equation 4.13 with Equation 4.10 and Equation 4.12, the estimation error becomes

$$e_i = -\sum_{j=1}^{n} a'_{ij} \cdot b_j(m_j - s_j). \tag{4.14}$$

Note that the outdoor concentration C_0 can be measured by accurate stationary monitoring stations. Thus, it is accurate and does not cause any errors in Equation 4.14.

Since the term $m_j - s_j$ in Equation 4.14 is a random number that follows distribution $L(0, v_i)$, we define the local generation rate vector H as

$$H = [b_1 h_1, b_2 h_2, ..., b_n h_n]^T, \tag{4.15}$$

where h_i equals $m_i - s_i$. Assume that the estimation errors of all the zones are $e = [e_1, e_2, ..., e_n]^T$. In the matrix form, the estimation errors can be calculated as

$$e = A^{-1} \cdot (-H). \tag{4.16}$$

Equation 4.10 gives the optimal pollutant concentration prediction with no sensors deployed. Equation 4.16 calculates the estimation errors for the prediction for all zones. As indicated in Equation 4.14, the estimation error is a random number which is the linear combination of the generation rates of all the zones.

Instead of predicting the pollutant concentration using empirical concentration distribution of each zone directly, we estimate the distributions of source generation rates and use them to calculate the concentrations. The reason is that unlike the source generation rates, the concentrations are highly correlated. For example, assume we have two zones i and j, with estimation errors e_i and e_j respectively. The airflow between i and j is high. If there is an accurate sensor located in zone i, the prediction error in zone j, based on the observation on zone i, should decrease significantly. However, if we model their empirical pollutant concentration distributions independently, the estimation error in zone j

remains the same, which greatly overestimates the error. By modeling the distributions of the independent source generation rates, one can avoid error overestimation resulting from ignoring correlations.

Concentration Estimation with Accurate Sensors

In the previous discussion, we have derived the optimal concentration prediction model for a non-monitored building in Equation 4.10. Now we consider a scenario in which stationary and accurate sensors are deployed. Specifically, we will evaluate how the deployment of accurate sensors affects the concentration estimation accuracies of uncovered zones.

Assume that in zone i, an accurate and stationary sensor is deployed. Thus, the estimation error for zone i is 0. The predicted concentrations are

$$\begin{cases} C_i = r_i & i \in R \\ \sum_{j=1, j\notin R}^{n} a_{ij} C_j + \sum_{j\in R} r_j + b_i s_i + u_i C_0 = 0 & i \notin R, \end{cases} \tag{4.17}$$

where r_i is the reading of the sensor in zone i and R is a subset of the set of all zones Z and contains the zones that are covered by accurate sensors. Thus, the airflow matrix A is

$$A = \begin{bmatrix} a_{11} & \cdots & a_{1i} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots & & \vdots \\ 0 & \cdots & a_{ii} & \cdots & 0 \\ \vdots & & \vdots & \ddots & \vdots \\ a_{n1} & \cdots & a_{ni} & \cdots & a_{nn} \end{bmatrix}. \tag{4.18}$$

In general, if a sensor is placed in zone i, all the elements $a_{ij,j\neq i}$ should be 0.

The stationary sensors are assumed to be accurate. Thus, r_i equals g_i. The prediction

error of zone i, instead of Equation 4.14, is calculated as

$$e_i = \begin{cases} 0 & i \in R \\ -\sum_{j=1, j \notin R}^{n} a'_{ij} \cdot b_j h_j & i \notin R, \end{cases} \tag{4.19}$$

where a'_{ij} is the elements of the inverse matrix A^{-1} of the modified airflow matrix. As shown in Equation 4.19, the source generation rate uncertainties of the covered zones no longer introduce errors in the concentration prediction of the uncovered zones. The source generation rate vector H is therefore

$$H = [b_1 h_1, ..., b_i h_i = 0, ..., b_n h_n]^T. \tag{4.20}$$

With the modified coefficients shown in Equation 4.18 and Equation 4.20, Equation 4.16 is still valid. As more stationary sensors are deployed, the overall uncertainty, i.e., the number of zones whose generation rates influence the prediction accuracies for other zones, decreases. Thus, there are two benefits of deploying accurate sensors: (1) the estimation errors of covered zones become 0 and (2) it can help reduce the expected errors of other uncovered zones.

Concentration Estimation with Inaccurate Sensors

The mobile, low-cost, and miniature sensors carried by individuals are essential to address the uneven spatial pollutant distribution problem. One problem for such sensors is that they typically suffer significant drift error [61]. In other words, if we have placed a mobile sensor in zone i, the sensor reading r_i is not equal to the ground truth g_i.

As demonstrated in Section 4.6.2, the long term drift of Figaro TGS2602 sensors, after compensation, can be modeled using a Gaussian distribution with mean 0. Thus, the relationship between the ground truth and inaccurate mobile sensor reading becomes

$$d_i = r_i - g_i, \tag{4.21}$$

where d_i is the sensor reading error and is a random number following Gaussian distribution $N(0, q_i)$, in which q_i is the standard deviation. Note that this error, caused by sensor drift, is independent of source generation rates, and hence independent of the concentration prediction errors.

Assuming that there is an inaccurate mobile sensor located in zone i, the airflow matrix and concentration prediction equation remain the same as in Equation 4.18 and Equation 4.17, while the estimation error is

$$e_i = \begin{cases} d_i & i \in R \\ -\left(\sum_{j=1, j\notin R}^{n} a'_{ij} \cdot b_j h_j + \sum_{k\in R} a'_{ik} d_k\right) & i \notin R, \end{cases} \tag{4.22}$$

where R is the set of zones that are covered by mobile sensors. Also the source generation rate vector H is

$$H = [b_1 h_1, ..., b_i h_i = d_i, ..., b_n h_n]^T. \tag{4.23}$$

4.4.3 Optimal Concentration Prediction Model

So far, we have derived concentration prediction and error estimation models for all the following scenarios: (1) no sensors; (2) stationary sensors only; and (3) mobile sensors only. However, the current solution for the inaccurate mobile sensors is sub-optimal.

For many types of low-cost mobile sensors, drift error eventually dominates empirical data based prediction error. For example, the 4-month uncompensated drift error of Figaro TGS2602 VOC sensor is about 0.8 PPM on average [61], while the standard deviation of VOC distribution in many environments is only around 0.3 PPM [19, 47].

Thus, if a sensor network contains inaccurate sensors and gives them the same trust as the accurate stationary sensors, there is no guarantee that deploying more such sensors can reduce the overall expected error. At some point, the sensor drift error may exceed the prediction errors caused by empirical estimations for the remaining uncovered zones. As

a result, the addition of new inaccurate sensors can instead increase the overall expected error of the sensor network. We will provide such an example in Section 4.6.3.

One naïve approach to address this problem is to discard a mobile sensor's reading when its expected drift exceeds a certain threshold. However, this approach is both inefficient and inaccurate. It is inefficient because it often unnecessarily shortens the useful lifespans of the sensors. It is inaccurate because it neglects the additional information provided by the mobile sensors, which can be useful even if the sensor readings are more inaccurate than the empirical data based predictions.

The prediction model should optimally balance the weightings of the inaccurate sensor readings and the similarly inaccurate source generation rate estimates to improve the overall prediction accuracy. We use a weight assignment technique to address this problem. The weights represent trustworthiness values and should be determined based on the distributions of the sensor drift and source generation rates.

Specifically, the weight-adjusted estimation of the concentration for zone i can be described as

$$\begin{aligned} C_i &= w_i \cdot C_{estimate} + (1 - w_i) \cdot C_{sensor} \\ &= -\frac{w_i}{a_{ii}} \left(\sum_{j=1, j \neq i}^{n} a_{ij} C_j + b_i m_i + u_i C_0 \right) + (1 - w_i) r_i, \end{aligned} \tag{4.24}$$

where $C_{estimate}$ is the estimated concentration for zone i assuming no sensor is located in that zone, C_{sensor} is the sensor reading for zone i, and w_i is the assigned weight that ranges from 0 to 1. If w_i equals 0, the sensor reading is considered accurate and hence determines the concentration of the zone. If w_i equals 1, it means there is no sensor located in the zone.

The ground truth concentration for zone i can be re-written as

$$g_i = -\frac{w_i}{a_{ii}} \left(\sum_{j=1, j \neq i}^{n} a_{ij} \cdot g_j + b_i s_i + u_i C_0 \right) + (1 - w_i) g_i. \tag{4.25}$$

Thus, the estimation error, defined as $C_i - g_i$, is

$$e_i = -\frac{w_i}{a_{ii}} \left(\sum_{j=1, j\neq i}^{n} a_{ij} \cdot e_j + b_i h_i \right) + (1 - w_i) d_i, \tag{4.26}$$

Therefore, in matrix representation, the airflow matrix A is

$$A = \begin{bmatrix} a_{11} & a_{12}w_1 & \cdots & a_{1n}w_1 \\ a_{21}w_2 & a_{22} & \cdots & a_{2n}w_2 \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1}w_n & a_{22}w_n & \cdots & a_{nn} \end{bmatrix}, \tag{4.27}$$

in which except for a_{ii}, each element in the ith row is multiplied by w_i.

By solving Equation 4.26, we have

$$e_i = -\sum_{j=1}^{n} a'_{ij} \cdot (w_j \cdot b_j h_j - (1 - w_j) \cdot a_{jj} d_j), \tag{4.28}$$

where the weight coefficient w_i is calculated as

$$w_i = \underset{w}{\text{argmin}}\ E_i(w), 0 \leq w \leq 1. \tag{4.29}$$

Since the optimal weight assignment minimizes the expected error, by definition it gives the optimal concentration prediction. Thus, the local generation rate vector H becomes

$$H = \begin{bmatrix} w_1 \cdot b_1 h_1 - (1 - w_1) \cdot a_{11} d_1 \\ w_2 \cdot b_2 h_2 - (1 - w_2) \cdot a_{22} d_2 \\ \vdots \\ w_n \cdot b_n h_n - (1 - w_n) \cdot a_{nn} d_n \end{bmatrix}. \tag{4.30}$$

Equation 4.16 can be used to calculate the estimation errors of all the zones.

The assigned weight, w_i, should be determined optimally based on the estimation accuracies of source generation rates and sensor drifts. However, finding the optimal weights is a non-trivial task. We can get a closed-form expression for expected error E_i if and only

if all h_i and d_i have Gaussian distributions. In this work we use Monte-Carlo simulation technique, which can accurately calculate the expected error regardless of the distributions of h_i and d_i. The details will be discussed in Section 4.6.2.

In general, Equation 4.24 gives the optimal concentration predictions. Equation 4.28 allows us to calculate the estimation error of the optimal prediction. They are both unified equations which can be applied for all the scenarios. Note that although we have presented equations for zones containing a single sensor, it is easy to extend the current solutions to cases where multiple sensors are co-located in a same zone.

4.5 Hybrid Sensor Network Synthesis

In this section, we describe algorithms to solve the hybrid sensor network synthesis problem based on our optimal prediction model. Section 4.5.1 generalizes the problem and provides definitions. Section 4.5.2 discusses the reasoning and underlying observations for the synthesis algorithm. Section 4.5.3 describes the details of algorithm.

4.5.1 Problem Definition

In a hybrid sensor network, there might be multiple types of sensors with varying accuracies, long-term drift rates, lifespans, and prices. Our work mainly focuses on the trade-off between accuracy and price. In other words, given the same budget, we want to minimize the personal exposure estimation error of the sensor network.

Note that the exposure error, denoted as E_i', is different from the estimation error e_i and expected error E_i as defined in Section 4.4.1. In real world applications, we are interested in personal exposure rather than indoor concentrations. Thus, the value of a sensor should be determined both by its measurement accuracy and the number of people it serves. For example, if a sensor is placed in an isolated zone with no people in it, even if its reading is accurate, it does not improve the quality of personal exposure measurement.

We define the exposure error for zone i as

$$E_i' = \sum_{m=0}^{k} E_i(t_0 + m\Delta t) \cdot P_i(t_0 + m\Delta t) \cdot \Delta t, \tag{4.31}$$

where E_i' is the exposure error, $E_i(t_0 + m\Delta t)$ is the expected error of zone i during time interval from $t_0 + m\Delta t$ to $t_0 + (m+1)\Delta t$, $P_i(t_0 + m\Delta t)$ is the number of people in zone i during the same time interval, Δt is a time interval during which the number of people and expected error of each zone are considered to be constant, and k is the total number of such time intervals in a day. Note that the expected error is a function of time because of the motion of sensor carriers.

The problem of hybrid sensor network synthesis can be described as follows: given a certain budget, find a sensor network architecture for which the total cost of sensors is within the budget while the average personal exposure measurement error for all the zones is minimized. One could modify this definition if the accuracy were more important for some people than others, e.g., those with respiratory health problems.

4.5.2 Synthesis Overview

To construct a hybrid sensor network, we need to determine the types and quantities of sensors first. This problem is similar to the knapsack problem, in which we have a budget and a list of items. Each item has a weight and value, and we need to find the set of items that maximizes value while meeting a weight budget. If each type of sensor has a fixed value, i.e., amount of exposure error reduction, the problem is equivalent to the knapsack problem and hence NP-hard.

In our problem formulation, the exposure error improvement of each type of sensor is not fixed. It is dependent on the inter-zone airflow, sensor location, sensor drift distribution, source generation rate distribution, and the sensor architecture. For example, different placement locations for a sensor can lead to significantly different exposure er-

ror improvement results. Therefore, to determine the correct value of each sensor, we must perform sensor placement and allocation algorithms during the process of sensor selection. However, the sensor placement problem, even for the stationary sensors, is also NP-hard [61].

To address this problem, we rely on the observation that the price of the accurate stationary sensors is much higher than that of the inaccurate mobile sensors. For example, an accurate photo-ionization detector (PID) based VOC sensor may cost about $600, while a metal oxide VOC sensor costs only about $7.50. Even after considering the cost of all the peripheral components, the stationary sensors are still several times more expensive than the mobile sensors. Moreover, the stationary sensors need to be manually calibrated frequently, which increases the maintenance cost.

Therefore, we decompose the synthesis problem into two sub-problems. The first sub-problem is the selection and placement of the stationary sensors, which we solve by exhaustively searching all the possible selection and placement schemes. There are mainly two reasons for this design: (1) the high cost of the stationary sensors constraints the quantities that can be deployed in the sensor network and (2) stationary sensors can provide calibration opportunities for the mobile sensors, thus help to improve the accuracy of the entire network.

The second sub-problem is the selection and allocation of the mobile sensors, which we solve using a greedy algorithm. Because of the relatively large quantity of the mobile sensors, it is no longer suitable to use exhaustive search. We use a heuristic in which we choose one sensor per iteration based on its unit value. Unit value is defined as the exposure error reduction per unit cost. This is repeated until the budget is met.

4.5.3 Algorithm

Algorithm 3 Hybrid sensor network synthesis algorithm

Require: Z // set of rooms
Require: S_M // set of mobile sensors
Require: S_{ST} // set of stationary sensors
Require: J // set of mobile sensor carriers candidates
Require: U // set of source generation rate distributions
Require: D // set of sensor drift error distributions
Require: T // set of sensor prices
Require: M // set of mobility patterns of all the individuals
Require: b // budget

$e_{min} = \infty$ // minimal personal exposure error
$Y_{min} \leftarrow \{\}$ // sensor network architecture of e_{min}
$Y_{ST} \leftarrow placement\_search(S_M, b)$ // Y_{ST} is the set of all the possible stationary sensor placement schemes under current budget
$\forall Y \in Y_{ST}, W(Y) \leftarrow weight\_calculation(Y, D, U)$// W is the weight table
for $Y \in Y_{ST}$ **do**
  $e_{pre} \leftarrow error\_calculation(Y, U, D, M, W(Y))$
  $Y_{pre} \leftarrow Y$
  $c \leftarrow total\_cost(Y, T)$
  while $c < b$ **do**
    $\Delta e_{int} \leftarrow 0$
    for $s \in S_M$ **do**
      for $j \in J$ **do**
        $X \leftarrow Y_{pre} \cup (s, j)$
        $W(X) \leftarrow weight\_calculation(X, D, U)$
        $e_{cur} \leftarrow error\_calculation(X, U, D, M, W(X))$
        $\Delta e_{cur} = \frac{e_{pre} - e_{cur}}{T(s)}$
        if $\Delta e_{cur} \geq \Delta e_{int}$ **then**
          $\Delta e_{int} \leftarrow \Delta e_{cur}$
          $e_{int} \leftarrow e_{cur}$
          $Y_{int} \leftarrow X$
        end if
      end for
    end for
    $e_{pre} \leftarrow e_{int}$
    $Y_{pre} \leftarrow Y_{int}$
    $c \leftarrow total\_cost(Y_{pre}, T)$
  end while
  if $e_{pre} < e_{min}$ **then**
    $e_{min} \leftarrow e_{pre}$
    $Y_{min} \leftarrow Y_{pre}$
  end if
end for

The detailed algorithm is shown in Algorithm 3. The algorithm first searches all the possible assignments for stationary sensors within the budget limit. For each stationary sensor assignment, a greedy algorithm is used to assign mobile sensors. As long as the budget is not exceeded, the greedy algorithm tries to find the mobile sensor and the corresponding carrier so that the exposure error reduction per unit price, $\Delta E'$, is maximized. $\Delta E'$ is defined as

$$\Delta E' = \frac{E'_{pre} - E'_{cur}}{T(s)}, \tag{4.32}$$

where E'_{pre} is the previous average exposure error before assigning the new sensor, E'_{cur} is the current average exposure error after the assignment, and $T(s)$ is the price of the sensor to be assigned. When the algorithm ends, it returns a sensor network architecture, i.e., a sensor selection and the location/carrier of each sensor, with the minimal exposure error that the algorithm can find within the budget limit.

Each time a new architecture is considered, the weights are calculated according to Equation 4.29 and recorded in the weight table. The weight table can help reduce computational overhead since mobile sensor carriers may visit the same zone at different times. In that case, if other conditions did not change, the previous weight assignments can be reused.

4.6 Experimental Results

This section describes the evaluation of our model and synthesis algorithms. Section 4.6.1 gives the CO_2 experimental measurement for an office building. Section 4.6.2 describes the experimental setup. Section 4.6.3 shows the evaluation results of our pollutant concentration prediction model. Section 4.6.4 presents the simulation results of our hybrid sensor network synthesis algorithm.

4.6.1 A CO_2 Sensor Network Deployment and Analysis

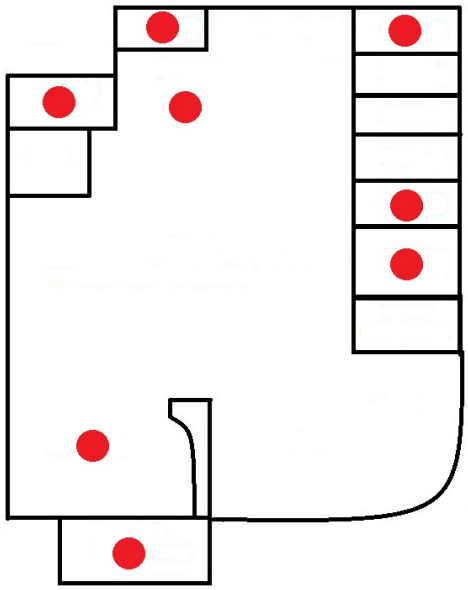
In this section, we describe our real-world CO_2 sensor network deployment and the data gathered with it.

Sensor Network Deployment

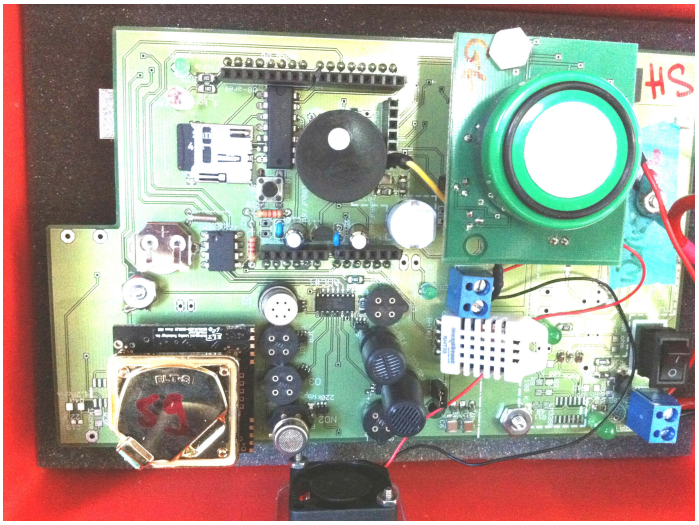
To estimate the airflow in a building, we performed a field experiment in which eight air quality sensing platforms were distributed throughout an office building. The sensor nodes, as shown in Figure 4.3(b), are custom-built with a processor-communication architecture based on the Arduino platform [5]. The sensor nodes are equipped with multiple sensors, including the non-dispersive infrared S100 CO_2 sensor from ELT. This sensor has high accuracy, low drift, and low sensitivity to temperature and humidity. The unit cost is approximately $60. The CO_2 concentration is sampled at 0.2 Hz and is stored with a time stamp on a micro-SD card. A fan is used to pull air through the sensors at a constant rate of around 1 liter per minute.

Sensor calibrations were performed in a gas chamber before deployment. Gas mixtures in the chamber were precisely set using mass flow controllers operated through a Labview control system. We performed the calibration at 3 different CO_2 levels: 0 PPM, 730 PPM, and 2,268 PPM. The exposure at each concentration level lasted 60 minutes. The CO_2 sensor readings had good linear relationships with the target pollutants.

Figure 4.3(a) shows the floorplan of the deployment building and sensor locations. The building is divided into eight zones, and contains room types such as single-occupancy office, large office with multiple occupants, and conference room. A sensor node is placed in each zone and collects data continuously from 8 June 2012 through 21 June 2012. The platforms were generally positioned near the room occupants, while trying to ensure they were far enough away to not be a nuisance, or be disturbed.



(a)



(b)

Figure 4.3: Deployment environment and equipment: (a) building for deployment and (b) custom-built CO_2 measurement equipment.

Data Analysis

The measurement data from the deployment are used to derive the indoor airflow matrix A. The daily average concentration for zone i can be estimated as

$$(-a_{ii}) \begin{bmatrix} C_i(1) \\ \vdots \\ C_i(l) \end{bmatrix} = \sum_{j=1, j\neq i}^{n} a_{ij} \begin{bmatrix} C_j(1) \\ \vdots \\ C_j(l) \end{bmatrix} + \begin{bmatrix} G_i(1) \\ \vdots \\ G_i(l) \end{bmatrix}, \tag{4.33}$$

where $C_i(t)$ is the average concentration for day t, l is the total duration of the experiment, and $G_i(t)$ is a constant determined by the daily outdoor concentration and indoor generation rates. The average concentration is calculated by averaging all sensor readings from 9:00 to 18:00. The airflow patterns between day and night are different. In this work, we focus on the daytime pattern when the majority of human activities take place. The nighttime pattern could easily be included as a separate time interval if desired. Linear regression analysis is applied to Equation 4.33 to estimate the airflow matrix A. The airflow matrix is later used in simulations to evaluate our concentration prediction and synthesis techniques in Section 4.6.3 and Section 4.6.4.

4.6.2 Simulation Setup

In this section, we describe the general experiment setup and Monte-Carlo simulation technique used to calculate the expected error.

General Setup

The prediction model and synthesis simulator is written in Matlab and runs on a 4-core Intel Xeon E31230 machine with 8 GB memory. The airflow in the simulated building for sensor deployment is assumed to be the same as in 4.6.1.

The expected error (standard deviation of the estimation error) without sensors is assumed to be around 0.3 PPM based on the indoor VOC concentration measurement of an

industrial area building [47]. The data have passed the Lillie normality test. Therefore, we assume that the distributions of indoor source generation rates are Gaussian. We estimate the sensor drift error based on existing work [61]. The drift error of Figaro TGS2602 VOC sensors, after compensation, is about 0.24 PPM. The drift error data have also passed the Lillie normality test and hence its distribution is assumed to be Gaussian.

In our synthesis, the mobile sensor is modeled on Figaro TGS2602 VOC sensors, which cost about $7.50 each. The stationary sensor is modeled on Baseline-MOCON VOC sensors. Its accuracy is determined by the resolution of the analog-to-digital converter interface, and is assumed to be 0.03 PPM. The cost of the accurate Baseline-MOCON sensor is about $600. Both of these sensors require peripheral circuitry to gather and transmit data and perform proximity detection. The cost of such supporting circuit is about $150 [36]. Thus, in this work, we assume that the total costs of the mobile and stationary sensor nodes are $150 and $750, respectively.

Mobile sensors are automatically calibrated when in the same zone with a stationary sensor. Typically, the mobile sensor requires calibration at 3 or 4 pollutant levels to compensate for the non-linearity of the concentration translation function. In this case, since the carriers' daily mobility patterns are highly concentrated and repetitive [25, 60], and the pollutant concentration can change from day to day [14], multi-level calibration for the mobile sensors is feasible. Thus, we consider the calibrated mobile sensors as accurate as the stationary sensors. Note that this assumption is not a necessity. Our technique can be used even if the calibration is imperfect or unfeasible.

To evaluate the sensor network performance and select appropriate mobile sensor carriers, human motion traces are needed. In this work, we generate motion traces using the human mobility model described by Kim et al. [39]. Their mobility model is based on a statistical survey of the existing literature and U.S. Bureau of Labor Statistics data. In the

model, the number of people in each zone is proportional to the area of the zone. Each individual's motion trace is determined based on the distribution and characteristics of arrival time, duration of work, and meetings. The details of the distributions and parameters can be found in the existing literature [39].

Monte-Carlo Simulation

To calculate the expected error based on Equation 4.28, it is necessary to calculate the standard deviation of e_i, which is a linear combination of several random variables. It is possible that those random numbers follow distributions other than Gaussian and thus we cannot find a closed form expression of the expected error. Therefore, we use Monte-Carlo simulation based on the following equation.

$$E_i = \text{std}\left(-\sum_{j=1}^{n} a'_{ij} w_i \cdot b_i \begin{bmatrix} h_{i,1} \\ \vdots \\ h_{i,k} \end{bmatrix} + a'_{ij}(1-w_i) \cdot a_{ii} \begin{bmatrix} d_{i,1} \\ \vdots \\ d_{i,k} \end{bmatrix} \right), \tag{4.34}$$

where k is the number of Monte-Carlo simulation trials, $h_{i,j}$ is a random number following distribution $L(0, v_i)$, and $d_{i,j}$ is a random number following distribution $N(0, q_i)$.

Monte-Carlo simulation is general enough to handle arbitrary source generation rates and sensor drift distributions. Its main disadvantage is the computational overhead. We set the number of trials to 10^5. By increasing the trial number tenfold to 10^6, the simulation results differ by 0.16% on average, thus we consider the current trial number sufficient.

4.6.3 Concentration Prediction Model Evaluation

In this section we evaluate our pollutant concentration prediction models. Since the stationary sensors are accurate and hence always have fixed weights of 0, we do not include

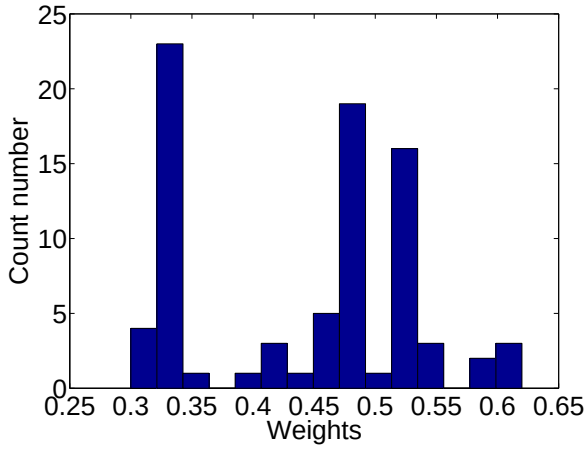


Figure 4.4: The sensor drift compensation weight distribution.

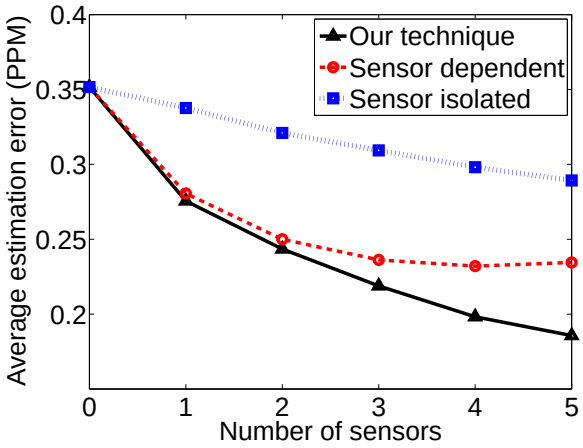


Figure 4.5: The average error for different error estimation schemes.

stationary sensors in this evaluation. We have randomly selected 5 carriers from the motion traces and varied the number of mobile sensors. Based on the resulting sensor network architectures, we apply different methods to predict the pollutant concentrations of all the zones. During sensor network construction, the weights and average expected errors are recorded. Figure 4.4 shows the distributions of all the weights. The X axis gives the weight values and the Y axis gives the frequency of appearance.

Figure 4.5 shows the expected errors of various concentration prediction schemes. The "sensor isolated" scheme assumes that a sensor's readings are not used to aid in estimating concentrations in other zones. The "sensor dependent" scheme uses sensor readings to aid

Table 4.1: Comparison Between the Heuristic and Optimal Solution

Budget ($)	Heuristic (PPM×minute)	Optimal (PPM×minute)	Differences (%)
750	108.93	108.93	0
900	88.72	88.72	0
1050	81.26	66.97	17.58
1200	73.32	62.24	15.11
1350	68.62	60.13	12.38
1500	59.00	59.00	0
Average			7.51

in estimates for distant zones; the prediction error is calculated based on Equation 4.22. In contrast to our technique, neither of the two schemes use weights to trade off position error and drift error. As a result, our technique improves the prediction accuracy by 40.4% on average compared with the "sensor isolated" method, and by 11.2% on average compared with the "sensor dependent" method. The results show that both indoor airflow modeling and weight adjustment are important.

When the deployed number of sensors increases from 4 to 5, the average prediction error of the "sensor dependent" method increases as shown in Figure 4.5. When we use the greedy algorithm to add sensors to the network, the new sensor is often located in an uncovered zone with highest estimation error. Thus, at some point the prediction errors of the remaining uncovered zones are smaller than the sensor drift error. As a result, without the weight adjustments used in our optimal prediction technique, increasing the number of inaccurate sensors in the network may cause the decrease of the overall average sensor network accuracy. When there are 5 sensors deployed, the "sensor dependent" method incurs 26.3% more error compared with our optimal technique.

4.6.4 Hybrid Sensor Network Evaluation

We compare hybrid sensor network architecture accuracy against that of two other architectures. The first contains only mobile, inaccurate, low-cost sensors. The second

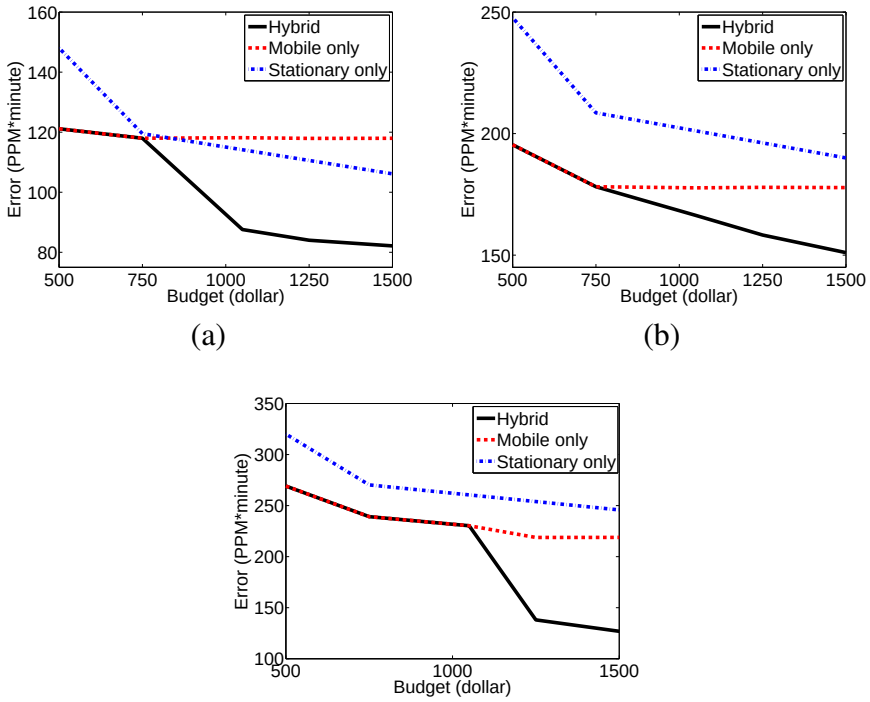


Figure 4.6: The synthesis results for (a) small, (b) medium, and (c) large human motion traces.

contains only stationary, accurate, expensive sensors. All of the three approaches use the algorithm described in Algorithm 3 to construct the network.

Figure 4.6 presents the simulation results. The simulation is performed on small, medium, and large human motion traces. There are 20 individuals and 4 sensor carrier candidates in the small trace, 30 individuals and 6 sensor carrier candidates in the medium trace, and 40 individuals and 10 sensor carrier candidates in the large trace. When the budget is less than $750, we are not able to afford any stationary sensors, thus the solution is the same for both the mobile-only and hybrid schemes. As the budget increases, the hybrid solution starts to outperform the other two solutions. Note that the stationary-only solution is optimal (but for a constrained problem definition), while the mobile only and hybrid solutions are heuristic due to the problem decomposition described in Section 4.5.2.

When the budget is very limited, the mobile-only solution outperforms the stationary-

only solution since no stationary sensor can be afforded. When we have a large enough budget, the stationary-only solution gives the most accurate measurement by placing an accurate sensor in every zone. The hybrid sensor network architecture, however, provides the best solution when the budget is between these extremes. In our simulation, when the budget is no less then $750 (thus can afford at least one stationary sensor), the hybrid architecture improves the sensor network accuracy by 23.9% on average compared with the mobile-only architecture, and by 35.8% on average compared with the stationary-only architecture.

Even though our proposed algorithm can significantly improve the personal exposure measurement accuracy, it is not optimal. We compared the algorithm with the optimal solution for a small trace with 20 individuals and 5 carrier candidates (computational cost prevented us from finding optimal solutions for larger problem instances). The optimal solution was found using exhaustive search for both the stationary sensors and mobile sensors. The results are shown in Table 4.1. In 3 of the 6 test cases, our heuristic returns the optimal solution. In the worst case, it has 17.58% more error. On average, our heuristic achieves an accuracy that is about 7.5% less than optimal.

It should be noted that this work addresses the long-term personal exposure monitoring problem. It requires an estimation of the average indoor pollutant generation rates and the average air flow rates over a long period of time. However, for short-term pollutant estimation, e.g., an emergent outbreak, since the air flow patterns and generation rates are dynamic and unpredictable, it cannot be guaranteed that our technique can always improve the prediction results. To improve the performance for instant event detection, a denser sensor network deployment and some additional information of the field, such as the ventilation conditions, are required.

4.7 Conclusion

We have described a synthesis and evaluation framework for hybrid sensor networks. This framework is composed of an optimal indoor concentration prediction and its error estimation model and hybrid sensor network synthesis algorithm. A field experiment was used to measure the inter-zone airflow. Our model improves accuracy by 40.4% on average by considering the trade-offs between location-dependent and drift-dependent measurement error. Simulations indicate that our hybrid sensor network architecture on average is 23.9% more accurate than the mobile-only architecture and 35.8% more accurate than the stationary-only architecture.

CHAPTER V

Mobile Sensing Networks Noise Reduction and Sensor Calibration

5.1 Introduction

In Chapter III and Chapter IV, we have described methods that can improve the sensor network accuracy by employing novel calibration, modeling, and synthesis techniques during the deployment. However, by analyzing the data collected from the deployments, we find that sensor data typically contain significant noises even with the accurate and undrifted sensor networks. Those noisy readings can trigger false alarms, lead to incorrect scientific conclusions, and generate sub-optimal solutions, all of which can greatly limit the application and usefulness of mobile sensor networks. Thus, this problem must be addressed.

There are several causes of noisy sensor readings. The metal oxide sensors are typically very sensitive to environment parameters, e.g., temperature and humidity, which cannot be perfectly measured near the sensor surface. The imprecise estimation of those parameters contributes to the noises. Moreover, there can be many unexpected problems in the real-world deployment, such as breakdown of electrical components, surge of power supplies, and signal noise in the circuits, all of which can introduce noises [20]. Another source of noises, observed and reported both by existing literature [57] and our own de-

ployment, is sensor drift. Sensor drift changes the sensor calibration function, shifting the measurement results from the ground truth without proper compensation. For example, in our own deployment, we find that the sensor drift can increase the average sensor error by orders of magnitude. Drifted sensors must be re-calibrated before they can be trusted and used again.

Sensor drift is typically one directional and stable within a short period of time, making it possible to compensate for it and recover the corrupted data. Once near an accurate, stationary, and regularly maintained air quality monitoring station [63], the drifted sensor can be calibrated using ground truth readings. However, such calibration opportunities are scarce. In many applications, people typically do not have frequent access to the ground truth readings. Moreover, the drift rates of sensors differ. They are determined by the sensor type and the actual environment the sensors are exposed to. For example, during our deployment, the CO sensor drifted by more than 30 times compare with ground truth on average, while the ozone sensor drifted by 5 times. Therefore, it is inadequate to use a predetermined offset to predict and compensate for drift. To address this problem, researchers rely on the observation that the co-located sensors nodes, which are equipped with the same type of sensors, observe the same physical environment and thus their readings are correlated and can be used to calibrate each other. However, in real-world application without a dense deployment, such calibration opportunities are still rare and heavily limited by the mobility patterns of individuals [70].

Another significant problem for mobile air quality sensor networks is cross sensitivity. The metal oxide sensors, utilizing either the oxidation or reduction reactions with the pollutant gases occurring in the sensor surface, can respond to and quantify the air pollutants with reasonable sensitivity and accuracy. However, for those sensors, many pollutants share the same reaction property. For example, both CO and NO_2 can cause oxidation

reactions with the surface materiel. Thus, the sensors usually respond to a wide range of pollutants other than the targeting gas. This property is called cross sensitivity [71]. Cross sensitivity is typically considered as a drawback for the metal oxide sensors since the gas composition in the environment is usually unknown and hard to differentiate. Because of cross sensitivity, the readings of different types of sensors are usually correlated. This property can be used to identify the compositions of pollutants in the environment [18].

We leverage the correlations of different metal oxide sensors to help identify and recover the abnormal readings, as well as addressing the cross sensitivity problem. In many recent mobile sensing network designs, researchers have built sensing devices equipped with multiple types of sensors to detect various pollutants co-existed in the environment [36, 68]. For such applications, it is possible to exploit the correlation of readings and reduce sensor errors using Bayesian belief networks [33]. The basic Bayesian network approach works well for the noises caused by random factors, but fails when sensors drift, which is common in real-world applications.

In this work, we aim to design a system that can efficiently reduce sensor noises, re-calibrated sensor functions, and identify the gas compositions in the air simultaneously. To achieve those goals, we have developed a Bayesian belief network based system which is capable of incorporating uncertain evidence and re-calibrating drifted sensors. The Bayesian network provides estimated ground truth readings for sensor re-calibrations, while the re-calibrated sensors can help the Bayesian network improve its estimates. To evaluate our technique, we have deployed 9 co-located mobile sensing devices equipped with different types of metal oxide sensors close to an air quality monitoring station in Denver, Colorado. The monitoring station can provide the ground truth reference, which allows us to determine and quantify the noise and drift.

In sum, this work makes the following contributions:

1. we have designed and implemented a Bayesian belief network based system to reduce sensor noises;

2. we incorporate and address the sensor function calibration problem within the Bayesian network framework; and

3. we have deployed a real-world mobile sensor network to investigate the sensor drift. The data from the deployment are later used to evaluate our technique.

By analyzing the collected data, we have observed significant drift within a short period of time, e.g., a couple of months for most of the sensors. For the drifted data, compared with the closest and state-of-art technique, our method can reduce error by 34.1% on average. Our system can recover 36.4% of the abnormal readings, which is 4 times better than the most relevant existing technique. Since our technique mainly targets the drift, it should have similar performance with the Bayesian network approach for the undrifted data. Experimental results show that our technique can achieve 87.3% abnormality detection rate, which is almost equal to the Bayesian belief network.

The rest of this chapter is organized as follows. Section 5.2 discusses existing related work. Section 5.3 provides an overview of the system. Section 5.4 describes the Bayesian belief network approach and how to use it to reduce sensor noises. Section 5.5 discusses the limitations of existing Bayesian network approaches and presents our solution. Section 5.6 describes our real-world deployment and the evaluation results of different techniques.

5.2 Related Work

The related work can be placed in three categories: co-located sensor calibration, sensor outliers detection and correction, and Bayesian network based approaches.

Co-located sensor calibration. Xiang et al. [61] developed a model to estimate sensor drift and designed a compensation technique to minimize the sensor drift assuming no access to ground truth readings. Their approach assigns weights to co-located sensors to combine sensor readings optimally given known sensor drift distributions, which can be derived from their model. Bychkovskiy et al. [12] have proposed a two-phase post-deployment sensor drift compensation technique in which co-located sensors are calibrated in pairs using linear functions. Miluzzo et al. [49] have proposed CaliBree, an auto-calibration algorithm for mobile sensor networks, in which mobile sensor nodes opportunistically interact with accurate stationary sensors and hence enable calibration to reduce sensor drift. Those techniques require that the co-located sensors are of the same type and thus should have the same response from the physical environment. However, such calibration opportunities are usually unrealistic and rare in real-world applications. In contrast to the previous work, our technique can work on mobile sensing devices containing various types of metal oxide sensors.

Sensor outliers detection and correction. Great efforts and resources have been invested in addressing the sensor outlier detection and cleaning problem [16, 72]. For example, Bettencourt et al. [8] have presented an abnormalities detection technique to identify errors during event detection in ecological wireless sensor networks. Their technique uses the spatio-temporal correlations of sensor data to detect outliers. Rajasegarar et al. [56] have proposed a support vector machine (SVM) based technique to detect sensor outliers. Their approach uses a one-class quarter-sphere SVM to classify and identify the local outliers. Unlike our technique, their method cannot estimate the actual ground truth readings and recover outliers. Papadimitriou et al. [51] have developed a technique that uses multi-granularity deviation factor to dynamically detect the outlier readings based on the correlations of local nodes. Their technique cannot address the sensor drift problem

though, when one or more sensors' readings are shifted persistently. Kumar et al. [43] proposed a technique that performs a two-stage drift correction. First, they use a Kriging-based approach to provide estimated ground truth readings. Then a Kalman-filter based technique is used to compensate for sensor drift. However, Kriging requires certain spatial density in sensor nodes deployment. Moreover, a Kalman-filter based approach relies on the assumption of a state-space underlying model and knowledge of the model parameters, which is unrealistic in real-world applications when the environment of the deployment field is often unknown and very dynamic.

Bayesian network based approaches. Elnahrawy et al. [20] have used a naive Bayesian network to identify local outliers and detect faulty sensors. This technique uses a trained Bayesian classifier for probabilistic inference. Each node locally computes the probabilities of each of its incoming readings and determine the readings as outliers if their probabilities are not the highest among all the possible outcomes. Their approach can only work for the homogeneous sensors. Janakiram et al. [33] have proposed a technique to detect sensor outliers based on Bayesian belief network. They leverage the conditional correlation of the readings from different types of sensors. However, their approach does not take into consideration sensor drift and sensor function re-calibration, which are considered and addressed by our method.

5.3 System Overview

Figure 5.1 shows the overview of our system. The input of the system is the raw analog sensor readings in the form of voltage or resistance. Note that actual ground truth readings are not required and only used for evaluation. The input sensor readings are first processed using a Bayesian belief network, which is trained with normal data from the in-field deployment. The Bayesian network can generate the estimated ground truth readings

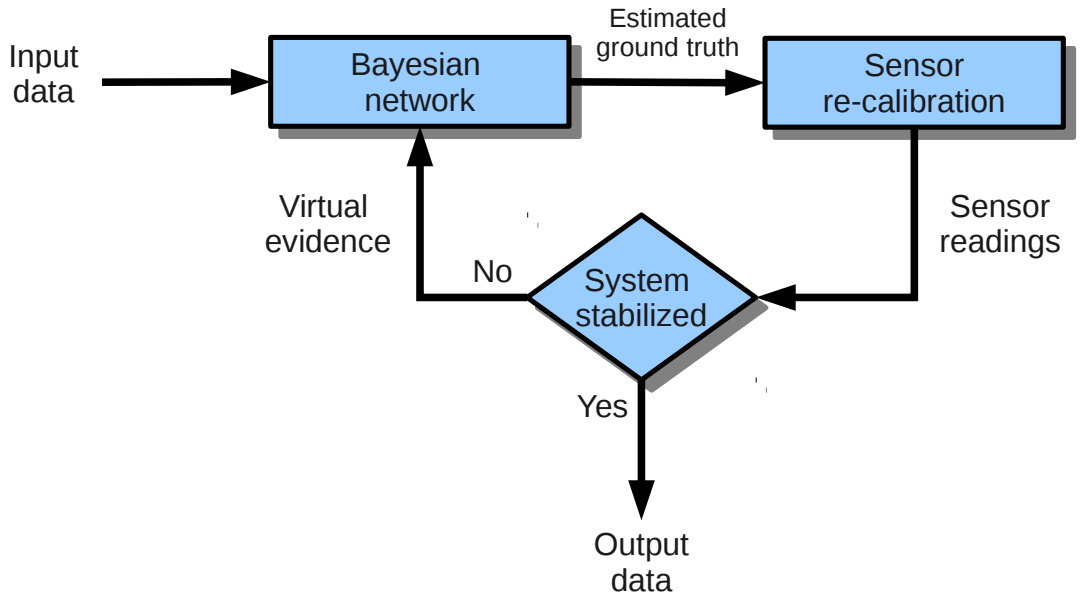


Figure 5.1: System overview.

based on the readings from all the correlated sensors. The estimated ground truth readings are then used to re-calibrate the sensors, i.e., generate the new sensor functions which can translate the input analog readings into pollutant concentration in the unit of parts per million (PPM). The new sensor functions are used to generate the sensor concentration readings, which can derive the error distribution together with the estimated ground truth. The error distribution can be used to update the virtual evidence of the Bayesian network. The virtual evidence is used by the Bayesian network to calculate the estimated ground truth, thus forming a loop. If the system is stabilized, the loop exits and the recovered sensor readings are produced.

5.4 Basic Bayesian Belief Network

In this section, we first introduce the basic Bayesian belief network. Then we discuss how to use it in real-world applications.

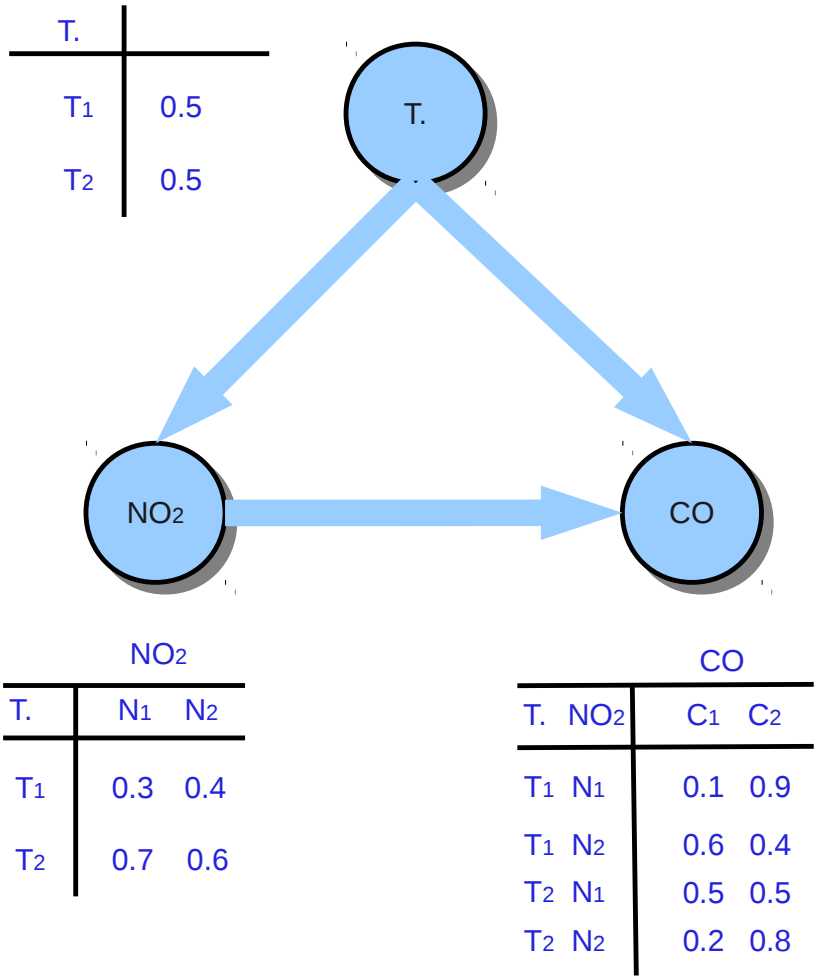


Figure 5.2: An example of Bayesian belief network.

5.4.1 Bayesian Network Introduction

Bayesian networks are widely used to detect and recover abnormal data points for sensor networks. The Bayesian network is built based on Bayes' theorem, which can be described using the following equation [20]:

$$P(t|o) = \frac{P(o|t)P(t)}{P(o)}, \tag{5.1}$$

where t is the ground truth reading and o is the observed sensor reading. Bayesian networks are capable of exploiting the inter-dependent or causal relationships of correlated sensors

readings. The types of the sensors involved can be different, which makes it appropriate for our application. A Bayesian network is a directed graph consisting of nodes and arcs [37]. The nodes represent variables, and in our application they represent readings from different types of co-located pollutant sensors. The arcs represent causal or conditionally dependent relationships between nodes. In our application, different types of sensors observing the same physical environment are considered to be conditionally dependent with each other. For example, the CO sensor and temperature sensor are correlated since the readings of the metal oxide sensors are heavily influenced by temperature.

Figure 5.2 shows an example Bayesian belief network for a simple sensor network. In this application, there are three different types of sensors, which can measure temperature (T.), carbon monoxide (CO), and nitrogen dioxide (NO_2), respectively. Each sensors' readings can be discretized into n values, with each discrete value denoted as T_n, C_n, and N_n, respectively. Without loss of generality, we assume two distinct discrete values for each sensor type. All the sensors are correlated. The readings of metal oxide sensors are strongly affected by the temperature [4]. Moreover, the readings of the NO_2 sensor and CO sensor are also correlated with each other because of cross sensitivity.

As shown in the figure, the Bayesian network describing this sensor network contains three nodes, with each representing one type of sensor. There are two arcs connecting the temperature sensor with the metal oxide sensors and one arc connecting the two metal oxide sensors. To calculate the probability inference of each variable given the input of other variables as evidence, each node is associated with a table, which is called conditional probability table (CPT). CPT describes the conditional dependence between any node with its parents. For the root node with no parents, CPT describes the distribution of the variable itself. CPT can be derived by training the network using historic data. The size of

the probability table is determined using the following equation.

$$N_i = (\prod_{j \in P_i} d_j) \times d_i, \tag{5.2}$$

where N_i is the total number of entries in the table for node i, d_i is the number of discrete values, and P_i is the set of direct parent nodes. The size of the CPT grows exponentially with the number and size of the direct parent nodes. Thus, to limit the requirement for the memory space, it is important to carefully design the network so that the number of parent nodes and their numbers of discrete values are appropriate. Based on the CPT and using the readings of other sensor nodes as evidence, we can calculate the probability inference for each discrete value using Equation 5.1. Note that the evidence can contain an arbitrary number of observed sensor nodes. For example, even if we only know the readings of the temperature sensor, we can still estimate the ground truth readings of the NO_2 and CO sensors. Increasing the number of inputs can improve the confidence of the output.

5.4.2 Bayesian Network for Real-world Applications

In this section, we discuss how to apply the Bayesian network technique to our real-world application, which is air quality monitoring using mobile sensing devices equipped with multiple types of sensors. Without loss of generality, we assume that there are four types of equipped sensors: temperature, NO_2, CO, and ozone (O_3). Their readings are all correlated. The Bayesian network graph for this application is shown in figure 5.3. In the graph, there are four nodes, denoted as T, CO(S), NO_2(S), and O_3(S), represent the temperature sensor and metal oxide sensors. Besides the sensor nodes, there is another type of nodes, which are instances of CO(T), NO_2(T), and O_3(T). Those three nodes represent the actual concentration (ground truth) of the corresponding pollutants in the environment.

In the figure, there are arcs connecting the temperature sensor to all the three types of metal oxide sensors since the readings of the temperature sensor influences them all. The

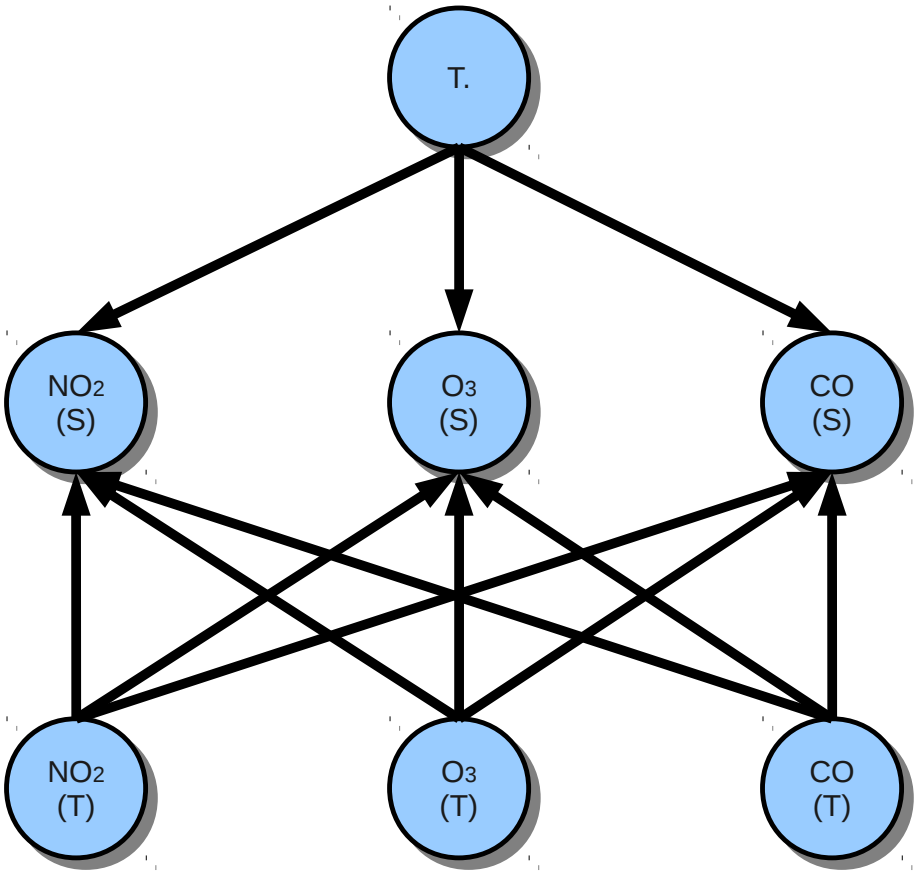


Figure 5.3: The basic Bayesian network structure for our application.

metal oxide sensors are assumed to be independent from each other, and the same is true for the ground truth concentration nodes. However, for each type of pollutant, its ground truth readings can have significant impact on the readings of all of the three metal oxide sensors. Thus, there are three arcs connecting the ground truth concentrations of each type of pollutant to all the three sensors. When the ground truth is not available, which is the case for most of the time, the probability inference of the three ground truth nodes can be calculated using the input of the four actual sensors. The value with the highest probability is considered as the estimated ground truth. In other words, the readings of the temperature

and metal oxide sensors are treated as input evidence, and the estimated values of ground truth concentrations are the output of the system.

5.5 Bayesian Network with Sensor Re-calibration

In this section, we first talk about the problems of the basic Bayesian network for real-world applications in which sensors may drift. Then we introduce virtual evidences to address the drift problem and the sensor re-calibration technique to improve the performance of the Bayesian network. Finally, we present the combined recursive system and describe the details and algorithm to implement it.

5.5.1 Problems for Basic Bayesian Network

As discussed in Section 5.4, Bayesian network can clean the corrupted data and detect abnormal readings by leveraging the inter-dependency of correlated sensors. For the sensor noises caused by random environment and electrical noises, it is quite efficient and sufficient. However, in our applications, sensors frequently drift. It has been shown, both by existing literature [57, 61] and by our own measurement data presented in 5.6.1, that sensor drift is a very common and severe problem in real-world applications for those metal oxide sensors. Significant drift can be accumulated within just a couple of months, making the sensors effectively useless afterwards if not re-calibrated. Thus, the problem of sensor drift and the error caused by drift must be addressed.

The basic Bayesian belief network approach described in Section 5.4 cannot address the drift problem. Drift can be considered a systematic deviation of the sensor readings from the ground truth caused by the changing of the sensor function. When multiple sensors drift, the basic Bayesian network approach can no longer identify the abnormal readings, let alone correct them and recover the ground truth. For example, consider a Bayesian network containing three nodes, which represent CO, NO_2, and O_3, respectively.

Table 5.1: An Example Error Distribution with Reported Reading of 1.5 PPM

	Ground truth prob. (%)		
	$0 \sim 1$PPM	$1 \sim 2$PPM	$2 \sim 3$PPM
Accurate	0	100	0
Drifted	30	70	0
Breakdown	33	33	33

Assume that the CO and NO_2 sensors are drifted and constantly report extreme values that can rarely be observed in the normal environment. In that case, even if the ozone sensor is not drifted, the results of the Bayesian network can still be erroneous because the two drifted sensors out-weight the one undrifted sensor. Thus, the basic Bayesian network cannot produce reasonable results due to the influence from multiple drifted sensors. Note that the scenario that we have more than one drifted sensors in the system simultaneously is not uncommon, as shown by our deployment results in Section 5.6.1. Thus, the system described in Figure 5.3 is inadequate to address the real-world problems. To apply the Bayesian network in such circumstances, we need to (1) incorporate a ranking mechanism that can quantify the sensor uncertainties into the Bayesian network and (2) design a drift compensation scheme to re-calibrate the sensor function and recover the corrupted data simultaneously within the Bayesian network framework.

5.5.2 Error Distribution and Uncertain Evidences

As the sensor drifts, its sensing sensitivity deteriorates and the uncertainty of its readings increases. A Bayesian network treats all its input equally, which is problematic considering sensor drifts. For example, if a CO sensor is recently calibrated while a O_3 sensor has not been calibrated for a long time, we should clearly give the CO sensor more weights in determining the final output of the Bayesian network. In other words, within a Bayesian network framework, we must have an evaluation mechanism which can rank and quantify the trustworthiness of each particular sensor.

To address this problem, we use error distributions to represent the sensitivity and trustworthiness of the sensors. An example of error distributions is shown in Table 5.1. In the example, we assume that the sensor has reported an environment concentration of 1.5 PPM. The actual ground truth ranges from 0 to 3 PPM and is divided into three discrete categories. We assume that in the environment the probability for the ground truth to be in any of these three categories is equal. As shown in the table, if the sensor is accurate, then the probability that the actual ground truth is within the range of 1 to 2 PPM given a reported reading of 1.5 PPM is 100%. If the sensor is drifted, the sensor becomes less accurate and the possible value of the ground truth spreads wider. If the sensor is breakdown, it loses most of its sensitivity and the ground truth is no longer correlated to the sensor readings.

In that way, we have translate the determined sensor readings into distributions, which inherently represent the trustworthiness of the sensors. Such input to the Bayesian network is called virtual evidence. Note that virtual evidence cannot be applied to the Bayesian network directly. The Bayesian network must be modified to incorporate such uncertain evidences.

5.5.3 Bayesian Network with Virtual Evidence

In this section, we discuss how to address the problem of noise reduction with drifted sensors using virtual evidences. For the basic Bayesian network, the inputs can only be determined value. Thus, virtual evidences cannot be directly applied to the Bayesian network. To incorporate the virtual evidences, some constraints, which is called Jeffrey's rule [34], must be honored. The concept of Jeffrey's rule is described as follows.

Suppose the universe of all the events is denoted as U. We have a set of mutually exclusive events $\gamma_1, ..., \gamma_n$, which is a subset of U, and P is the probability distribution of

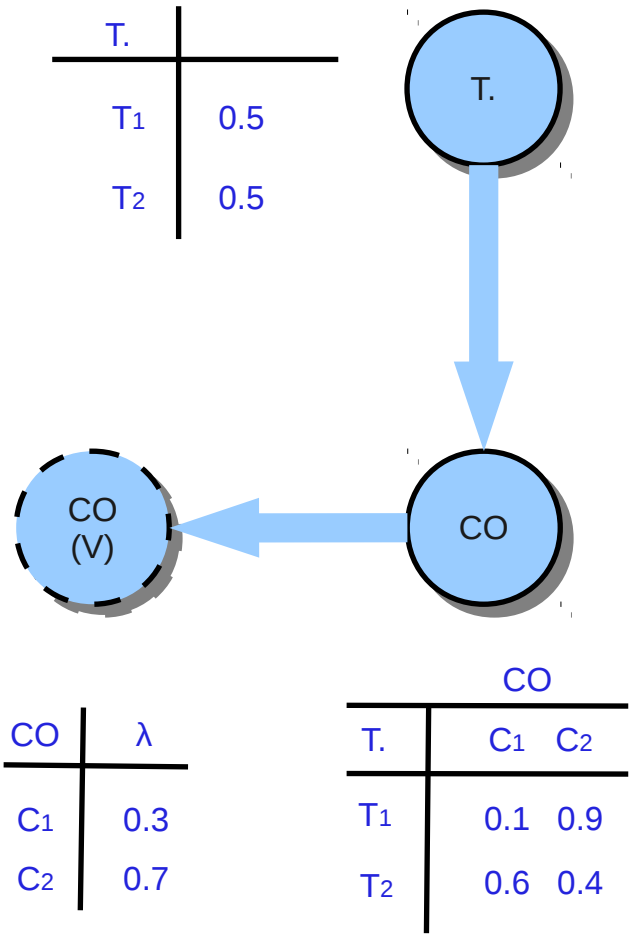


Figure 5.4: An example of virtual node.

those events. After applying the virtual evidence, the beliefs for events $\gamma_1, ..., \gamma_n$ change and the updated distribution is denoted as P'. P' should satisfy the following equation.

$$P(\alpha|\gamma_i) = P'(\alpha|\gamma_i), \forall i = 1, ..., n. \tag{5.3}$$

where α is any event in the universe. In other words, after the virtual evidence is accepted, the posterior probability of α can be changed, but the conditional probability for $\alpha \in U$ regarding to the events $\gamma_1, ..., \gamma_n$ must remain the same.

To treat the virtual evidence as determined value while honoring the Jeffrey's rule, the Bayesian network should be modified by adding a virtue node to the drifted sensor nodes [15]. Figure 5.4 shows an example Bayesian network with virtual nodes. In the figure, there are two sensor nodes, which are temperature and CO. The temperature sensor is assumed to be accurate and with little drift, while the CO sensor can drift. The CO sensor node is associated with a virtual node, denoted as CO(V). The virtual node also has its own conditional probability table. The CPT of the virtual node should be calculated using the error distribution of the actual sensor node so that the beliefs of the whole Bayesian network comply with Jeffrey's rule. The detailed methods and equations to calculate its probability table can be found in existing literature [15, 52]. Note that the virtual nodes is only dependent on the corresponding sensor node and independent of all the other nodes in the network.

Figure 5.5 shows the Bayesian network structure of our application after incorporating the virtual evidences. Since the temperature sensor and the hypothetical ground truth concentration sensors are assumed to be accurate, they are not associated with any virtual nodes. Each metal oxide sensor, which is prone to drift, is associated with a virtual node. The contents in the CPT of the virtual nodes can be calculated using the error distributions of the actual nodes, which can be derived with the information of the (estimated) ground truth readings and the sensor readings. Note that unlike the simple example shown in figure 5.4, in our real-world application, there are multiple sensor nodes associated with virtual nodes, reflecting the fact that more than one sensor can drift at the same time. We address the problem of multiple virtual nodes using a recursive method as suggested by Peng et al. [52].

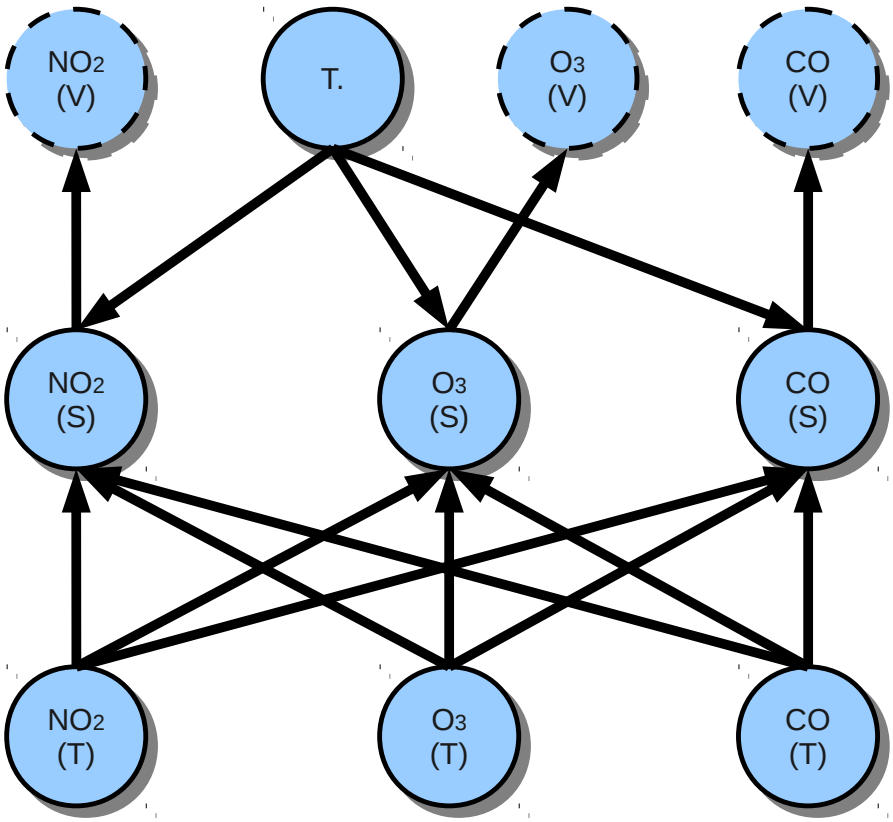


Figure 5.5: The Bayesian network with virtual nodes.

5.5.4 Sensor Function Re-calibration

For the metal oxide sensors, the signals gathered from the sensors are actually analog readings that indicate the voltage levels across the load resistance. Thus, we need a transfer function to translate the analog input signal into pollutant concentration. Such a function is called a sensor calibration function, or sensor function. The abnormal readings caused by environmental noises do not reflect a change of the sensor calibration function. However, when sensors are drifted, the sensor calibration functions change, which can result in a systematic increase in the number of abnormal readings. To address the drift problem, the

sensor functions can usually be compensated and corrected by comparing to the ground truth readings reported by accurate stationary sensors [49] or the sensor readings of the same types of mobile sensors nearby [12, 65, 66]. In our applications, the ground truth reading is assumed unavailable mostly and we usually do not have a deployment density high enough for frequent re-calibrations. Therefore, in our system, the sensor function is re-calibrated on-the-fly with the help of the observations from other type of sensors stationed in the same device and the estimated ground truth reported by the Bayesian network.

In this work, we apply a piece-wise linear function as the sensor function, which is shown in the following equation.

$$C = p_1 + p_2 * V + p_3 * T, \tag{5.4}$$

where C is the pollutant concentration, p_i are the fitting parameters, V is the voltage, and T is the temperature. The temperature information is reported by the on-board sensors. The parameters in the equation is derived by applying linear regression technique to the training data, which is composed of the analog input signal and the ground truth concentration. Since accurate sensors providing ground truth readings are usually not available, we use the estimated ground truth concentration returned by the Bayesian network instead. We apply the same linear regression technique to the estimated ground truth and generate the new sensor function. Note that as the sensitivity of the sensors deteriorates, the performance of this re-calibration scheme reduces. When a sensor breaks down and loses most of its sensitivity, the sensor can no longer be re-calibrated.

5.5.5 System Design

In this section, we describe the recursive method to improve our results and the flow and algorithm to implement our system.

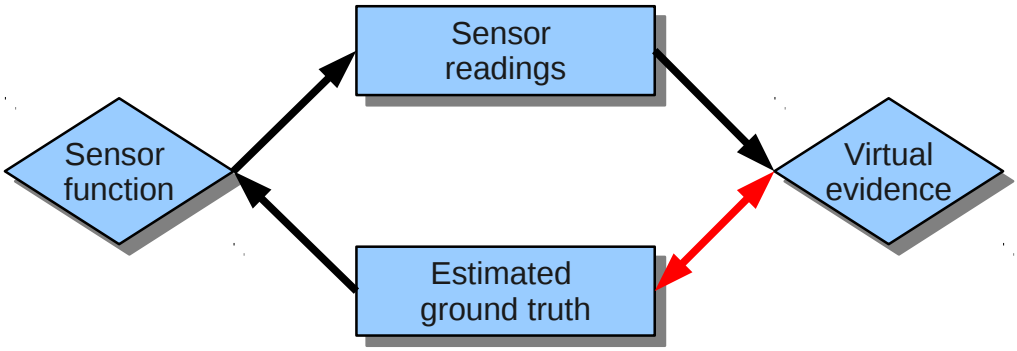


Figure 5.6: The relationship between components of the system.

Recursive Sensor Re-calibration

Since we have described the Bayesian network and the sensor re-calibration components of our system as shown in Figure 5.1, in this section we explain how to combine them together and form the recursive loop in the system. Given the analog sensor input, it is intuitive to use the Bayesian network to derive the estimated ground truth readings, and then use the estimated ground truth readings to re-calibrate the sensor function. However, this is insufficient for our application. Figure 5.6 shows the relationships of the components in the system. We start from the sensor function first. The sensor function is determined by the (estimated) ground truth readings, and is essential to derive the sensor readings. Subsequently, the sensor readings can change the error distribution, which is derived using the sensor readings and estimated ground truth readings. The virtual evidence is an interpretation of the error distribution, and thus is determined by the sensor readings and estimated ground truth readings. However, as indicated in the figure, when the virtual evidence is fed back into the Bayesian network, it in turn can impact the values of the estimated ground truth. Thus, these components of the system form a loop and a single run usually cannot generate a stabilized solution.

Therefore, in this work, we propose a recursive approach to address this problem. The

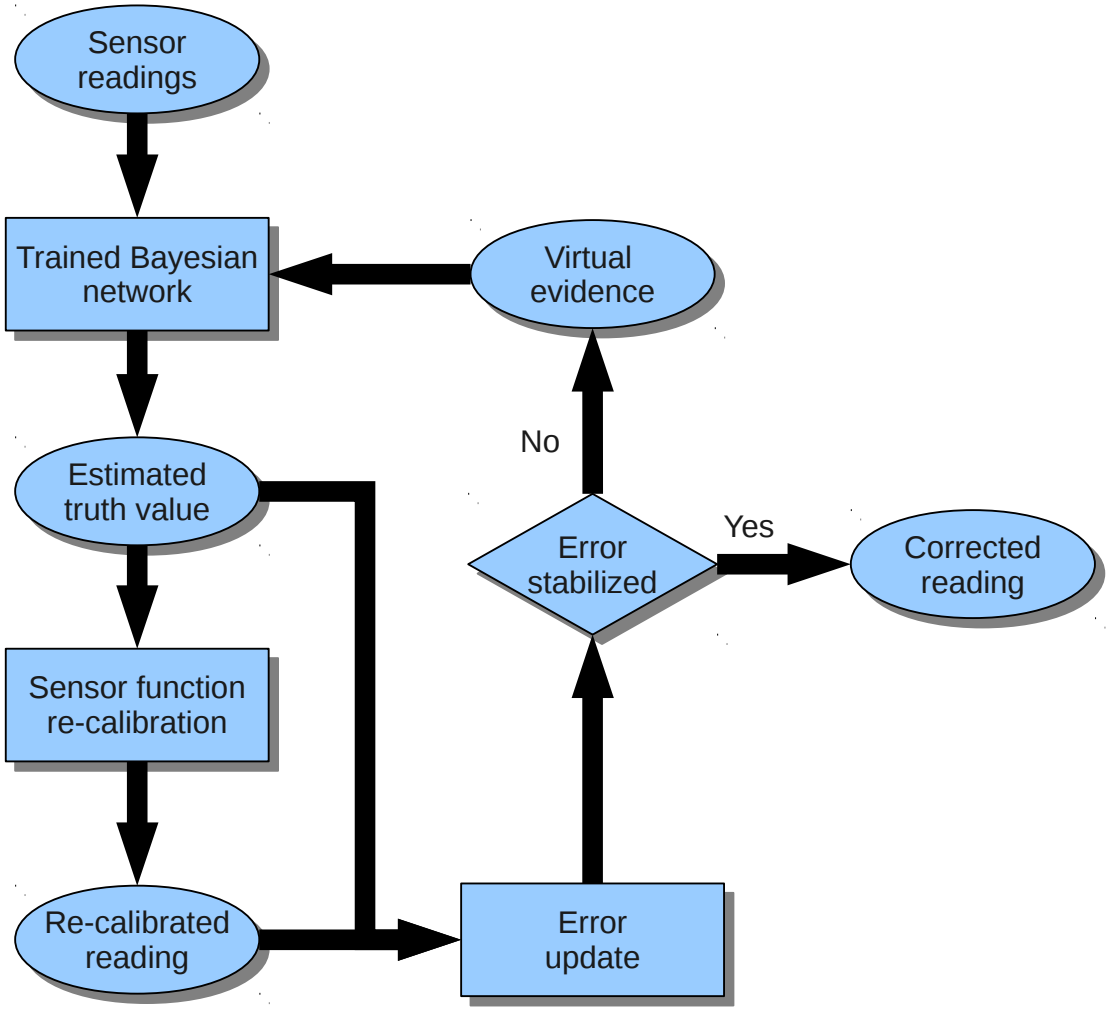


Figure 5.7: System flow.

estimated ground truth and the sensor functions are updated recursively until convergence. In that case, we assume that the final result is the best estimation possible for both the sensor function and the ground truth.

System Flow and Algorithm

In this section, we describe the flow of the system and the algorithm to implement it.

Figure 5.7 shows the flow of our system. The input sensor readings are first processed using a Bayesian belief network, which is trained using normal data from the in-field deployment. The Bayesian network can generate the estimated ground truth values based on

the conditional probability tables and readings from all the correlated sensors. The estimated ground truth readings are then used to re-calibrate the sensors, i.e., generate the new sensor functions which can translate the input sensor analog readings into actual pollutant concentrations. The new sensor functions are used to generate the sensor readings, which are compared with the estimated ground truth and derive the estimated error. The newly updated estimated error is compared with the previous estimations. If the change between them is within a certain threshold, we consider the system to be stabilized and the current results as the our best guess and hence, final output. If the system is not stabilized yet, the virtual evidence, which describes the error distributions of the input data, is updated using the new estimated concentration and subsequently used by the Bayesian network to generate the estimated ground truth readings for the next round of optimization. The loop continues after a certain number of runs or until the system converges.

The detailed algorithm for the implementation is described in Algorithm 4. The input of the system is the analog sensor readings. Before the loop starts, we first calculate the size of the input set and the sensor concentration readings using the current sensor functions. Then for each element in the input set, we use the Bayesian network, along with the virtual evidence, to calculate the corresponding estimated ground truth concentration. Subsequently, the estimated ground truth set, together with the input sensor readings, is processed using a linear regression function to generate the new sensor functions. Finally, the output set is derived using the new sensor function and the virtual evidence is updated. The process repeats until the output converges. As a result, the algorithm can generate our best estimation for both the ground truth concentrations and the sensor functions.

Algorithm 4 Algorithm for the Implementation of the System

Require: S // The input analog readings
Require: B // The trained Bayesian network
Require: O // The output set
Require: V // The initial distributions of the virtual evidences
Require: F // The initial sensor calibration function
  $N \leftarrow size(S)$
  $O \leftarrow F(S)$
  $E \leftarrow \emptyset$, E is the estimated ground truth set
  while O does not converge **do**
    for $i = 1 : N$ **do**
      $E(i) \leftarrow B(V(i), S(i))$
    end for
    $F \leftarrow Linear\_regression(E, S)$
    $O \leftarrow F(S)$
    Update V using O and E
  end while

5.6 Experimental Results

In this section, we first describe a real-world co-location deployment of 9 mobile sensor nodes and the analysis results for the deployment data. We then evaluate our system using the real-world data.

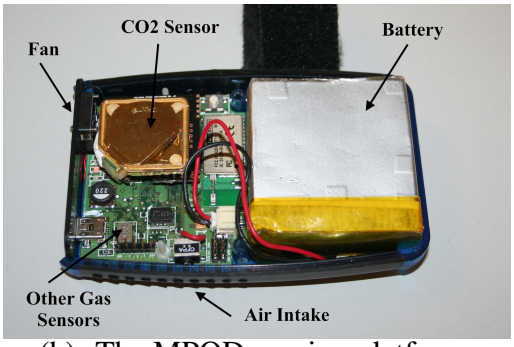
5.6.1 Mobile Sensor Network Deployment and Analysis

In this section, we discuss the details real-world deployment of a mobile sensor network and the implications of the environmental study results.

The Mobile Sensing Device

To investigate the effect of sensor drift in real-world applications and collect data to evaluate our data cleaning technique, we deployed a sensor network in Denver, Colorado. During the experiment, we deployed 9 M-Pods [36], which are shown in Figure V.8(b). The M-Pod is a custom-built mobile sensing device supporting embedded sensing, computation, and wireless communication. It supports detection of various air pollutants, including NO_2, CO, CO_2, O_3, and VOCs. It can also measure temperature, humidity, and light.

(a) The Denver air quality monitoring station.



(b) The MPOD sensing platform.

Figure 5.8: The deployment site and the M-Pod.

The latest revision of the M-Pod is compact (2×2.5 inches) and energy efficient, with a battery life of greater than 16 hours. The whole device, including a Li-ion battery with a capacity of 6,000 mA-h, is enclosed by a low-cost off-the-shelf case that can be carried using an armband or attached to a backpack. A 3.3 V DC fan is used to control airflow. A rectangular filter is installed around sensor to increase sensing accuracy and prolong sensor life. Most of the power hungry on-board sensors are power gated and can be controlled by commands from smartphones. Data are temporally stored in a one megabyte non-volatile EEPROM. The total cost of the on-board components and sensors is less than $150 and can be reduced further if produced in quantity.

To receive, store, and present the data gathered by our M-Pod device, we have developed on-board firmware, smartphone applications, data servers, and web interfaces. The

firmware defines protocols of sensing, storing, and sending the environmental data. The smartphone application communicates with the M-Pod via its Bluetooth interface. It can issue commands to and receive data from the M-pod. The data are transmitted to the on-line data server and stored in the databases. A web-based user interface allows users to access and analyze air quality data.

The Real World Deployment

The 9 M-Pods were used continuously from March to May 2013. The sensors were not changed throughout this period. For the majority of the time, the M-Pods were worn by users as part of an exposure assessment study. During three multi-day calibration periods in March, April, and May, the M-Pods were placed at a reference air quality monitoring site. The M-Pods were powered continuously on the roof of the monitoring building, in a ventilated enclosure near the air inlets for the reference monitors. The reference site, as shown in Figure V.8(a), monitors CO, NO_2, and O_3. It is located in downtown Denver, Colorado, and operated by the Colorado Department of Public Health and Environment (CDPHE). The highly accurate and regularly maintained air pollutant monitoring equipment in the station is used to provide the ground truth readings.

By co-locating the M-Pods with the reference monitors, we are able to derive both the sensor analog readings and ground truth, which can be used to determine the sensor calibration functions. The forms of the sensor calibration functions vary depending on sensor type. In this work, we use a piece-wise linear function. It is quite accurate according to lab and field measurements, and requires much less resources to compute compared with other more complicated forms of sensor functions. The calibrations are performed using the field data. Thus, it does not require specialized equipment, and can cover a wider range of environmental parameter space than lab calibrations. Before the fitting of the sensor

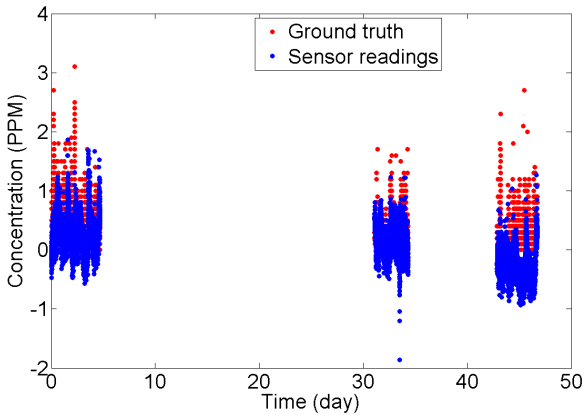
Table 5.2: The Statistics of the Original and Drifted Sensor Readings

Errors	Undrifted (PPM)			Drifted (PPM)		
	CO	NO	O_3	CO	NO	O_3
Average	0.31	16.13	0.04	10.72	112.45	0.20
Maximum	8.92	76.11	0.32	21.94	171.4	1.85
Std.	0.52	11.19	0.07	0.93	12.50	0.28
Corr. perct.	93%					

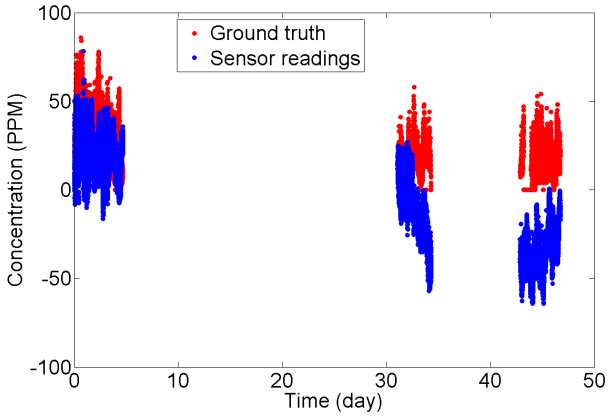
function, data filtering was performed to remove noise from the sensor readings. Minute medians were first calculated from the 6-second raw data. Then, we applied a filter based on difference in consecutive differences in the medians. There were two thresholds for the filter, an absolute threshold that was deemed unrealistic based on lab experiments, and 2 times the standard deviation of the differences. By performing calibrations periodically with the same sets of sensors, we were able to assess the change in baseline readings and sensitivity over time. The calibration functions derived by fitting to the data of the first calibration period, which is considered as the undrifted baseline, are applied to the entire data set.

Data Analysis

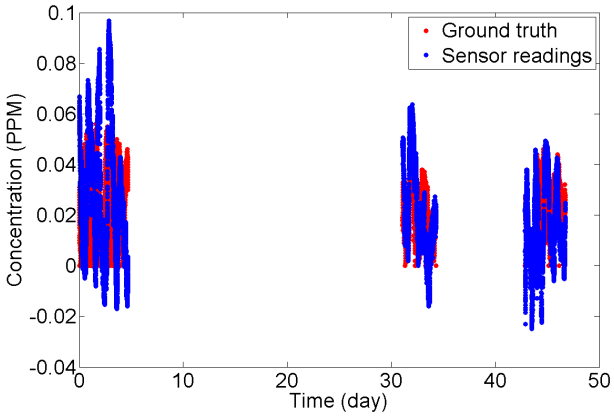
In this section, we present the analysis results of the collected data from the co-location deployment. We examine and compare the readings of the CO, NO_2, and O_3 sensors. An example of the measured data and the corresponding ground truth readings is presented in Figure 5.9. The X axis in the figure shows the time line of the deployment in the unit of days, while the Y axis shows the concentration of the pollutant in parts per million. Two sets of data are presented. The red dots represent the ground truth data measured by the accurate and regularly maintained equipment in the monitoring station, while the blue dots represent the data measured by the less accurate and drift-prone metal oxide sensors housed by the M-Pods. The total duration of the deployment is about two months. In the



(a) CO



(b) NO_2



(c) O_3

Figure 5.9: The measured data from the real-world deployment.

figure, there are three separate time periods, with each lasting for about one week. During that time period, the M-Pods are located in the station and calibrating. For the rest of the time, the M-Pods are carried by individual users and the ground truth readings of their exposed environments are unknown. Thus, the readings from those time periods are not included.

The resultant data show that the drift rates for different types of sensors vary. For the example in the figure, the NO_2 sensor experiences large drift. After two months, its error is increased more than 3 times. The CO sensor also suffers significant drift, though less compared to the NO_2 sensor with about 50% increase of error. But for the O_3 sensor, no significant drift is observed. The example shows that significant drift can occur within just a couple of months, rendering the corresponding sensor almost useless if not carefully re-calibrated. It demonstrated that drift is a real and severe challenge for those cheap sensors to be useful in real-world applications. Moreover, since the exposed environment and the properties of the sensors vary, different sensors usually exhibit different drift rates, making it impossible to re-calibrate the sensors using a predetermined model.

Among the 9 M-Pods deployed, we choose 6 of them during our analysis and evaluations. For the rest three, one of them did not return enough data due to transmission problem, and two of them have sensors completely dead within the two months deployment period. Table 5.2 shows the statistics of the sensing errors from the remaining 6 M-Pods. The error in the table are defined as the absolute variation between the sensor reading and the ground truth. We compare the drifted and undrifted data. The undrifted data are taken from the first time period as shown in Figure 5.9. The drifted data are taken from the third time period. The first three columns shows the average, maximum, and standard deviation of the error distributions. Significant drift can be observed for all the types of sensors. It should be noted that for some pollutants, such as NO_2 and CO, their

mean values change more significantly than the standard deviation, which implies a close to linear shift. The last column of the table shows the correlation percentage. Correlation percentage is defined as the percentage of the sensor pairs that shows strong correlation among all the possible pairs of all the sensors. The result shows a correlation percentage of over 93%, indicating that Bayesian network might be an appropriate solution.

The environment the sensors exposed to during the co-location experiment varies over time for different pollutants. For example, compared with the undrifted period, the average ground truth concentrations for the drifted period have shifted by 42.4%, 59.0%, and 4.7% for CO, NO_2, and O_3, respectively. The distribution of CO and NO_2 are highly dynamic and their concentrations differ significantly during the two time periods, which are separated by a time interval of about 2 months. The O_3 distribution, on the other hand, has much less deviations. We show in Section 5.6.2 that our technique works well for both scenarios.

In conclusion, our deployment data show that sensor drift and consequently the noise problem are very realistic and important for the metal oxide sensors. If not properly addressed, most of those sensors can be useless within just a couple of months. The drift rates are dependent on the environment and sensor properties and hence, vary for different sensors. Thus, it is not feasible to use predetermined correction methods: sensor calibration problem must be addressed using the field data. Moreover, different types of sensors show strong correlations, permitting noise reduction and sensor calibration.

5.6.2 Data Recovery and Sensor Calibration Results

In this section, we discuss the experimental environment setup and contrast our technique with the alternatives.

Experiment Setup

The sensor error cleaning and sensor re-calibration functions are written using Matlab, with the help of an external Bayesian network toolbox called bnt [9]. The program runs on a 4-core Intel Xeon E31230 machine with 8 GB memory. We use the data returned from 6 sensors out of a total of 9 sensors deployed, excluding the failed sensors and sensors with insufficient data. The failed sensors are not used since their readings are no longer correlated with each other and re-calibration cannot help improve the results. In other words, our technique does not have effect on them and they should be simply replaced. The failed sensor can be detected using both our technique and the Bayesian network method.

The CPT of the Bayesian network is derived from training. The training set is generated using the co-location data from undrifted (the first) time period. This approach is more appropriate since it require much less effort to cover a reasonable number of states than lab environment, and can provide us a more realistic prior distributions for temperature. The training dataset is filtered so that it contains only normal data. After the Bayesian network is trained, the contents in the CPT remain unchanged until the sensor is close to a reference station and have access to the ground truth readings again. For the parameter states that are not encountered during the training phase, we replace their contents with the encountered state of the closest distance, calculated using the Euclidean distance between those two states.

To evaluate our noise reduction and sensor re-calibration technique, we compare the following three approaches.

1. **Uncompensated**. This approach interprets the reported analog data using the predetermined sensor function from lab measurement and without any compensation

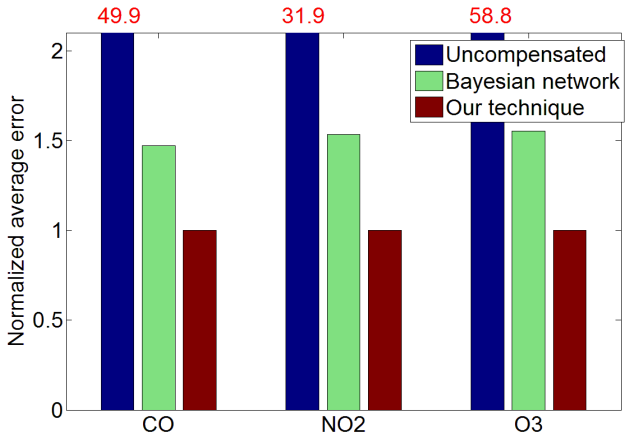


Figure 5.10: The data recovery results of various techniques for the drifted data.

scheme.

2. **Bayesian network**. This approach implements a Bayesian belief network based technique proposed by Janakiram et al. [33]. It is the most relevant and closely related work to the best of our knowledge.

3. **Our technique**. It improved upon the Bayesian network approach by incorporating the virtual evidence and sensor re-calibration.

We evaluate all the four approaches using the same set of testing data derived from our real-world deployment. We compare those techniques in terms of error reduction, recovery rate, and detection rate of the abnormal data. A data point is considered abnormal when its deviation from the ground truth value exceeding a certain threshold. In this work, the threshold is set as one standard deviation of the ground truth concentrations. Thus, recovery rate is defined as the percentage of abnormal readings becoming normal after being processed. The detection rate is defined as the percentage of correctly labeled data (normal or abnormal) for a dataset composed of undrifted data with noises.

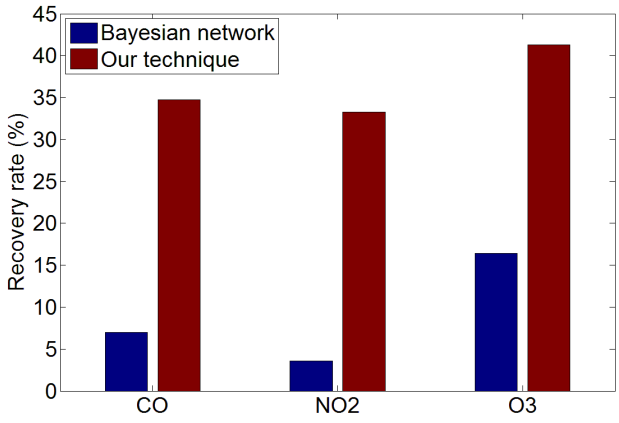


Figure 5.11: The percentage of successfully cleaned data.

Drifted Sensor Recovery Evaluation

Many existing abnormality detection approaches, such as distance based techniques [51, 62] or classification based techniques [56], cannot estimate the ground truth data and provide re-calibration opportunities for the drifted sensors. Thus, we do not include them in the comparison. Figure 5.10 shows the performance of various relevant data cleaning and recovery techniques. Since our technique focuses on the sensor drift and re-calibration problem, the experiment is performed on the third time period of the data set, which represents the drifted sensors. The Y axis of the bar graph shows the average errors, which are normalized to our recursive technique. Compared with the uncompensated approach, in which the sensor noises are not compensated and sensor calibration functions are not re-calibrated, our technique can incur only about 2.13% error on average. Moreover, compared with the Bayesian network approach, which is the closest existing technique, our technique is capable of reducing errors by 32.0%, 34.7%, and 35.5% for CO, NO_2, and O_3, respectively. Overall, our technique can reduce error by 34.1% on average.

After the estimated ground truth values are derived, we consider it as the ground truth concentration. However, since the ground truth concentration estimation is imperfect, the classification of sensor readings according to this estimate ground truth concentrations can

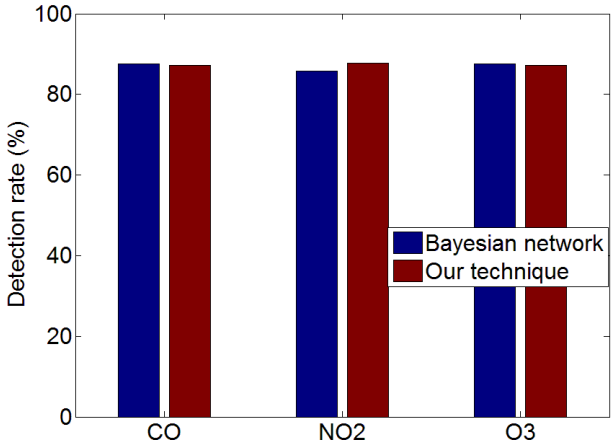


Figure 5.12: The abnormality detection results of various techniques for the undrifted data.

be wrong. Hereby we define data recovery rate as the percentage of corrected label data points after the data recovery scheme. Figure 5.11 shows the comparison results of various techniques in terms of data recovery rate. The rate is obtained by comparing the estimated readings against the ground truth. For our technique, the data recovery rates are 34.7%, 33.3%, 41.3% for CO, NO_2, and O_3, respectively. Compared with the Bayesian network approach, our technique is about 4 times better.

The execution times of our technique and the Bayesian network approach are quite similar. To process a day's data, which include 1440 data points, the average running time is 46 seconds for our technique and 39 seconds for the Bayesian network approach. This includes the time to train a Bayesian network using a training set consisting of more than 4,000 samples. Moreover, in this work the time resolution of the dataset is one minute, which is quite fine-grained compared with the requirement of many real-world applications. Thus, in general, we do not consider running time a problem.

5.6.3 Abnormality Detection and Cross Sensitivity

In addition to the data recovery and sensor function re-calibration for the drifted data, our technique is also capable of detecting abnormal readings caused by random noise

during undrifted period. The testing dataset in this case consists of undrifted data points, which are from the first time period. We create the testing dataset by manually setting the ratio of normal and abnormal data points. In this work, we set the ratio at 50%, which can be adjusted for the requirement of the application. We first pick all the abnormal readings from the dataset, then randomly choose the same number of random samples. Thus, in the testing set, the ratio of abnormal readings is set to be 50%. The detection rate is the combined correct classification ratio by excluding the false positives and false negatives. We compare the abnormality detection efficiency of our technique and the Bayesian network approach. The results are shown in Figure 5.12. The performance of our technique and the Bayesian network is quite similar, both having a detection rate of about 87%. This is as expected since during normal operation, the sensors are not drifted and thus, sensor function re-calibration should not have any significant impact on the results.

In addition to the sensor abnormality detection and drift compensation, another advantage of our technique, as well as the Bayesian network approach, is that it can automatically identify the pollutant composition in the air, thus addressing the cross sensitivity problem. In the real-world deployment, the deployment environment is often complex and heterogeneous. Therefore, without the knowledge of the pollutant composition in the air, it is very hard to get an accurate estimation of the pollutant concentration using the metal oxide sensors. Our technique can identify and quantify the pollutants in the air as long as they are previously included in the training set. However, the total number of pollutants in our system should be limited due to the constraint of storage space requirement.

5.7 Conclusion

In this work, we have presented a Bayesian belief network based system to reduce sensor noises and re-calibrate the sensor functions in the presence of sensor drift. Our method

improves upon the state-of-art Bayesian belief network techniques by incorporating the virtual evidence and adjusting the sensor calibration functions recursively. We have also performed a real-world deployment of mobile sensor network to investigate sensor drifts and validate our technique. Compared with the existing Bayesian network technique, our method can improve the result significantly. As a result, our technique can reduce error by 34.1% and increase the recovered data rate by 4 times on average.

CHAPTER VI

Conclusion

6.1 Conclusion

My thesis is dedicated to the design and validation of mobile air quality sensor networks, and developing techniques to solve the major challenges introduced by using the low-cost, compact sensors: drift, cross sensitivity, and noises. My contribution in this work can be summarized as follows.

1. We have designed a mobile sensing platform that can house multiple low cost metal oxide sensors. The platform is used to automatically collect personal exposure data, which are used in various researches.

2. To address the drift problem, we have developed a collaborative calibration technique for mobile sensors and sensor placement technique for the stationary sensors, which tries to maximize the calibration opportunity of the mobile sensors. We have also investigated the distribution of the sensor drift using the data collected from a custom-built chamber.

3. We observe that in the real-world application, deploying more mobile sensors is not always beneficial given that the sensor drift is significant. Thus, we propose a hybrid sensor network construction technique, which is based on the optimal indoor

pollutant concentration prediction model we developed. The hybrid sensor network consists of both stationary sensors and mobile sensors, and is able to achieve a better performance than both of them. Note that this work aims at long-term air quality monitoring and is not guarenteed to have a better prediction for instant events detection.

4. We observe that because of cross sensitivity, the metal oxide sensors housed on the M-Pod are all correlated with each other. By exploiting this correlation, we design a Bayesian network based system that can reduce sensor noise caused by sensor drift, re-calibrate the sensors, and identify the gas composition in the air.

APPENDICES

BIBLIOGRAPHY

BIBLIOGRAPHY

[1] Arduino BT. http://www.arduino.cc/en/Main/ArduinoBoardBluetooth.

[2] Chinese protesters accuse solar panel plant of pollution. http://www.nytimes.com/2011/09/19/world/asia/chinese-protesters-accuse-solar-panel-plant-of-pollution.html.

[3] Senseair K22 product specification. http://www.senseair.se/Datablad/ed_co2_engine_k22_oc.pdf.

[4] Metal oxide sensors. *Sensors and Actuators B: Chemical*, 33(1–3):198 – 202, 1996.

[5] Arduino open-source electronics prototyping platform. http://www.arduino.cc/.

[6] K. Arshak, E. Moore, G. M. Lyons, J. Harris, and S. Clifford. A review of gas sensors employed in electronic nose applications. *Sensor Review*, 24(2):181–198, 2004.

[7] J. Berry, L. Fleischer, W. Hart, C. Phillips, and J. Watson. Sensor placement in municipal water networks. *J. of water resources planning and management*, 131(3):237–243, 2005.

[8] LusM.A. Bettencourt, AricA. Hagberg, and LeviB. Larkey. Separating the wheat from the chaff: Practical anomaly detection schemes in ecological applications of distributed sensor networks. In *Distributed Computing in Sensor Systems*, volume 4549, pages 223–239. 2007.

[9] Bayes net toolbox for matlab. https://code.google.com/p/bnt/.

[10] S. K. Brown, M. R. Sim, M. J. Abramson, and C. N. Gray. Concentrations of volatile organic compounds in indoor air – a review. *Indoor Air*, 4(2):123–134, 1994.

[11] J. M. Burke, M. J. Zufall, and H. ozkaynak. A population exposure model for particulate matter: Case study results for PM2.5 in Philadelphia, PA. *Journal of Exposure Analysis and Environmental Epidemiology*, 11(6):470–489, 2001.

[12] Vladimir Bychkovskiy, Seapahn Megerian, Deborah Estrin, and Miodrag Potkonjak. A collaborative approach to in-place sensor calibration. In *Proc. Int. Symp. Information Processing in Sensor Networks*, pages 301–316, April 2003.

[13] K. Chakrabarty, S.S. Iyengar, H. Qi, and E. Cho. Grid coverage for surveillance and target location in distributed sensor networks. *IEEE Trans. Computers*, 51(22):1448–1453, December 2002.

[14] A. Chaloulakou and I. Mavroidis. Comparison of indoor and outdoor concentrations of CO at a public school. Evaluation of an indoor air quality model. *Atmospheric Environment*, 36(11):1769 – 1781, 2002.

[15] Hei Chan and Adnan Darwiche. On the revision of probabilistic beliefs using uncertain evidence. *Artificial Intelligence*, 163(1):67–90, 2005.

[16] Varun Chandola, Arindam Banerjee, and Vipin Kumar. Anomaly detection: A survey. *ACM Comput. Surv.*, 41(3):15:1–15:58, July 2009.

[17] J. M. Daisey, W. J. Angell, and M. G. Apte. Indoor air quality, ventilation and health symptoms in schools: an analysis of existing information. *Indoor Air*, 13, 2003.

[18] V. Di Lecce and M. Calabrese. Discriminating gaseous emission patterns in low-cost sensor setups. In *Proc. Int. Conf. Computational Intelligence for Measurement Systems and Applications*, pages 1–6, 2011.

[19] Lars E. Ekberg. Volatile organic compounds in office buildings. *Atmospheric Environment*, 28(22):3571 – 3575, 1994.

[20] Eiman Elnahrawy and Badri Nath. Cleaning and querying noisy sensors. In *Proc. Int. Conf. Wireless Sensor Networks and Applications*, pages 78–87, 2003.

[21] A. Emami-Naeini, M.M. Akhter, and S.M. Rock. Effect of model uncertainty on failure detection: the threshold selector. *IEEE Trans. Automatic Control*, 33(12):1106–1115, December 1988.

[22] EPA. Buildings and their impact on the environment: a statistical summary, 2009.

[23] S. P. Tarzia, Peter A. Dinda, Robert P. Dick, and Gokhan Memik. Indoor localization without Infrastructure using the acoustic background spectrum. In *Proc. Int. Conf. on Mobile Systems, Applications, and Services*, pages 155–168, June 2011.

[24] Panos G. Georgopoulos, Sastry S. Isukapalli, and Kannan Krishnan. *Modeling exposures to chemicals from multiple sources and routes*, pages 315–351. John Wiley & Sons, Ltd, 2010.

[25] M.C. Gonzalez, C.A. Hidalgo, and A.-L. Barabasi. Understanding individual human mobility patterns. *Nature*, 453(7196):779–782, 2008.

[26] Radha Goyal and Mukesh Khare. Indoor air quality modeling for PM10, PM2.5, and PM1.0 in naturally ventilated classrooms of an urban indian school building. *Environmental Monitoring and Assessment*, 176:501–516, 2011.

[27] H. Guo, S. C. Lee, L. Y. Chan, and W. M. Li. Risk assessment of exposure to volatile organic compounds in different indoor environments. *Environmental Research*, 94(1):57 – 66, 2004.

[28] J.-E. Haugen, O. Tomic, and K. Kvaal. A calibration method for handling the temporal drift of solid state gas-sensors. *Analytica Chimica Acta*, 407(1–2):23 – 39, 2000.

[29] S. R. Hayes. Estimating the effect of being indoors on total personal exposure to outdoor air pollution. *J. Air Pollution Control Association*, 39(11):1453–1461, 1989.

[30] S. R. Hayes. Use of an indoor air quality model (IAQM) to estimate indoor ozone levels. *J. Air & Waste Management Association*, 41(2):161–170, 1991.

[31] C. Huizenga, S. Abbaszadeh, L. Zagreus, and E. Arens. Air quality and thermal comfort in office buildings: Results of a large indoor environmental quality survey. In *Healthy Buildings 2006*, 2006.

[32] IBM ILOG CPLEX Division. IBM ILOG CPLEX 12.0 user manual, 2008.

[33] D. Janakiram, V. Adi Mallikarjuna Reddy, and A.V.U. Phani Kumar. Outlier detection in wireless sensor networks using bayesian belief networks. In *Proc. Int. Conf. Communication System Software and Middleware*, pages 1–6, 2006.

[34] Richard C Jeffrey. *The logic of decision*. University of Chicago Press, 1990.

[35] Y. Jiang, K. Li, L. Tian, R. Piedrahita, Y. Xiang, O. Mansata, Q. Lv, R. P. Dick, M. Hannigan, and L. Shang. MAQS: a personalized mobile sensing system for indoor air quality monitoring. In *Proc. Int. Conf. Ubiquitous Computing*, pages 271–280, September 2011.

[36] Y. Jiang, K. Li, L. Tian, R. Piedrahita, Y. Xiang, O. Mansata, Q. Lv, R. P. Dick, M. Hannigan, and L. Shang. MAQS: A personalized mobile sensing system for indoor air quality monitoring. In *Proc. Int. Conf. Ubiquitous Computing*, pages 271–280, September 2011.

[37] Steven M Kay. *Fundamentals of Statistical signal processing, Volume 2: Detection theory*. Prentice Hall PTR, 1998.

[38] S. Khuller, A. Moss, and J. Naor. The budgeted maximum coverage problem. *Information Processing Letters*, 70(1):39–45, 1999.

[39] J. Kim, V. Sridhara, and S. Bohacek. Realistic mobility simulation of urban mesh networks. *Ad Hoc Networks*, 7(2):411–430, 2009.

[40] L. Kirkeskov, T. Witterseh, L. W. Funch, E. Kristiansen, NewAuthor5, M. K. Hansen, and B. B. Knudsen. Health evaluation of volatile organic compound (voc) emission from exotic wood products. *Indoor Air*, 2008.

[41] N. E. Klepeis. Validity of the uniform mixing assumption: determining human exposure to environmental tobacco smoke. *Environ Health Perspect*, 107(Suppl. 2):357–363, 1999.

[42] Andreas Krause, Carlos Guestrin, Anupam Gupta, and Jon Kleinberg. Near-optimal sensor placements: maximizing information while minimizing communication cost. In *Proc. Int. Conf. Information Processing in Sensor Networks*, pages 2–10, 2006.

[43] D. Kumar, S. Rajasegarar, and M. Palaniswami. Automatic sensor drift detection and correction using spatial kriging and kalman filtering. In *Proc. Int. Conf. Distributed Computing in Sensor Systems*, pages 183–190, 2013.

[44] K. Lee, S. Hong, S. Kim, I. Rhee, and S. Chong. SLAW: A new mobility model for human walks. In *Proc. Int. Conf. Computer Communications*, pages 855–863, April 2009.

[45] Xiang Liu and Zhiqiang John Zhai. Prompt tracking of indoor airborne contaminant source location with probability-based inverse multi-zone modeling. *Building and Environment*, 44(6):1135 – 1143, 2009.

[46] N. Maisonneuve, M. Stevens, M. Niessen, P. Hanappe, and L. Steels. Citizen noise pollution monitoring. In *Proc. Int. Conf. Digital Government Research*, pages 96–103, 2009.

[47] M. R. Mannino and S. Orecchio. Polycyclic aromatic hydrocarbons (PAHs) in indoor dust matter of Palermo (Italy) area: Extraction, GC−MS analysis, distribution and sources. *Atmospheric Environment*, 42(8):1801 – 1817, 2008.

[48] S. Miller-Leiden, C. Lohascio, W. W. Nazaroff, and J.M. Macher. Effectiveness of in-room air filtration and dilution ventilation for tuberculosis infection control. *J. Air & Waste Management Association*, 46(9):869–882, 1996.

[49] Emiliano Miluzzo, NicholasD. Lane, AndrewT. Campbell, and Reza Olfati-Saber. Calibree: A self-calibration system for mobile sensor networks. In *Proc. Int. Conf. Distributed Computing in Sensor Systems*, volume 5067, pages 314–331, 2008.

[50] NIST. CONTAM: A multizone airflow and contaminant transport analysis software. http://www.bfrl.nist.gov/IAQanalysis/CONTAM/index.htm.

[51] S. Papadimitriou, H. Kitagawa, P.B. Gibbons, and C. Faloutsos. Loci: fast outlier detection using the local correlation integral. In *Proc. Int. Conf. Data Engineering*, pages 315–326, 2003.

[52] Yun Peng, Shenyong Zhang, and Rong Pan. Bayesian network reasoning with uncertain evidences. *J. Uncertainty, Fuzziness and Knowledge-Based Systems*, 18(05):539–564, 2010.

[53] O.A. Postolache, J.M.D. Pereira, and P.M.B.S. Girao. Smart sensors network for air quality monitoring applications. *IEEE Trans. Instrumentation and Measurement*, 58(9):3253–3262, Sept. 2009.

[54] N. Priyantha, A. Chakraborty, and H. Balakrishnan. The cricket location-support system. In *Proc. MOBICOM*, pages 32–43, 2000.

[55] A. Rabl and J.V. Spadaro. Public health impact of air pollution and implications for the energy system. *Annual Review of Energy and the Environment*, 25:601 – 628, 2000.

[56] S. Rajasegarar, C. Leckie, M. Palaniswami, and J.C. Bezdek. Quarter sphere based distributed anomaly detection in wireless sensor networks. In *Proc. Int. Conf. Communications*, pages 3864–3869, 2007.

[57] A.C. Romain and J. Nicolas. Long term stability of metal oxide-based gas sensors for e-nose environmental applications: An overview. *Sensors and Actuators B: Chemical*, 146(2):502 – 506, 2010.

[58] Uwe Schlink, Kathrin Strebel, Mark Loos, Rene Tuchscherer, Matthias Richter, Thomas Lange, Jakob Wernicke, and Ad Ragas. Evaluation of human mobility models, for exposure to air pollutants. *Science of The Total Environment*, 408(18):3918–3930, August 2010.

[59] D. G. Shendell, R. Prill, W. J. Fisk, M. G. Apte, D. Blake, and D. Faulkner. Associations between classroom CO_2 concentrations and student attendance in washington and idaho. *Indoor Air*, 14(5):333–341, 2004.

[60] C. Song, Z. Qu, N. Blumm, and A.-L. Barabasi. Limits of predictability in human mobility. *Science*, 327(5968):1018–1021, February 2010.

[61] Y. Xiang, L. S. Bai, R. Piedrahita, R. P. Dick, Q. Lv, M. P. Hannigan, and L. Shang. Collaborative calibration and sensor placement for mobile sensor networks. In *Proc. Int. Conf. Information Processing in Sensor Networks*, pages 73–84, April 2012.

[62] S. Subramaniam, T. Palpanas, D. Papadopoulos, V. Kalogeraki, and D. Gunopulos. Online outlier detection in sensor data using non-parametric models. In *Proc. Int. Conf. Very large data bases*, pages 187–198, 2006.

[63] Pieter Tans and Kirk Thoning. How we measured background co2 levels on Mauna Loa. http://www.esrl.noaa.gov/gmd/ccgg/about/co2_measurements.html.

[64] D. Tsai, J. Lin, and C. Chan. Office workers' sick building syndrome and indoor carbon dioxide concentrations. *Journal of Occupational and Environmental Hygiene*, 9(5):345–351, 2012.

[65] W. Tsujita, H. Ishida, and T. Moriizumi. Dynamic gas sensor network for air pollution monitoring and its auto-calibration. In *Proc. Int. Conf. Sensors*, pages 56–59, 2004.

[66] Wataru Tsujita, Akihito Yoshino, Hiroshi Ishida, and Toyosaka Moriizumi. Gas sensor network for air-pollution monitoring. *Sensors and Actuators B: Chemical*, 110(2):304 – 311, 2005.

[67] U.S. Environmental Protection Agency Green Building Workgroup. Buildings and their impact on the environment: A statistical summary, 2009.

[68] W. Willett, P. Aoki, N. Kumar, S. Subramanian, and A. Woodruff. Common sense community: Scaffolding mobile sensing and analysis for novice users. In *Pervasive Computing*, volume 6030, pages 301–318.

[69] D. P. Wyon. The effects of indoor air quality on performance and productivity. *Indoor Air*, 2004.

[70] Yun Xiang, R. Piedrahita, R.P. Dick, M. Hannigan, Qin Lv, and Li Shang. A hybrid sensor system for indoor air quality monitoring. In *Proc. Int. Conf. Distributed Computing in Sensor Systems*, pages 96–104, 2013.

[71] S. Zampolli, I. Elmi, F. Ahmed, M. Passini, G.C. Cardinali, S. Nicoletti, and L. Dori. An electronic nose based on solid state sensor arrays for low-cost indoor air quality monitoring applications. *Sensors and Actuators B: Chemical*, 101(1–2):39 – 46, 2004.

[72] Yang Zhang, N. Meratnia, and P. Havinga. Outlier detection techniques for wireless sensor networks: A survey. *IEEE Communications Surveys Tutorials*, 12(2):159–170, 2010.