A Framework for Participatory Sensing Systems

Diego Mendez Chaves
University of South Florida, dmendezc@mail.usf.edu

Follow this and additional works at: http://scholarcommons.usf.edu/etd
Part of the American Studies Commons, and the Computer Sciences Commons

Scholar Commons Citation

This Dissertation is brought to you for free and open access by the Graduate School at Scholar Commons. It has been accepted for inclusion in Graduate Theses and Dissertations by an authorized administrator of Scholar Commons. For more information, please contact scholarcommons@usf.edu.
A Framework for Participatory Sensing Systems

by

Diego Méndez Chaves

A dissertation submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy
Department of Computer Science and Engineering
College of Engineering
University of South Florida

Major Professor: Miguel A. Labrador, Ph.D.
    Wilfrido Moreno, Ph.D.
    Rafael Perez, Ph.D.
    Kandethody Ramachandran, Ph.D.
    Sudeep Sarkar, Ph.D.

Date of Approval:
    May 7, 2012

Keywords: spatial interpolation, kriging, PCA, ICA, spatial outliers, user’s location, spatial data-mining

Copyright © 2012, Diego Méndez Chaves
DEDICATION

To my family, Víctor, Patty and Lilita, who always support and encourage me to improve every day. To Caitrin and Satrajit, who are always there for me.
ACKNOWLEDGEMENTS

I would like to thank Dr. Miguel Labrador for his invaluable guidance and support, and for his patience and belief in my work. I would also like to thank the committee members, Dr. Wilfrido Moreno, Dr. Rafael Perez, Dr. Kandethody Ramachandran, and Dr. Sudeep Sarkar, for taking the time to be a part of my committee and providing important input. I would like to thank the Environmental Protection Commission of Hillsborough County and emQbit for his technical support and contributions. I would like to thank everyone in my research group for their important feedback regarding my research. And to all of my friends, Idálides, Óscar, Ingo, Mehrgan, Luis and Alfredo, for sharing all these great experiences over the past years.
### TABLE OF CONTENTS

LIST OF TABLES iv  
LIST OF FIGURES v  
ABSTRACT viii  

CHAPTER 1 INTRODUCTION 1  
1.1 Motivation 1  
1.2 Problem Statement 5  
1.3 Contributions 8  
1.3.1 A Framework for the Design and Implementation of PS Systems 8  
1.3.2 Data Verification 8  
1.3.3 Data Visualization 9  
1.3.4 Density Maps 10  
1.3.5 Structure of the Dissertation 11  

CHAPTER 2 LITERATURE REVIEW 12  
2.1 Data Verification 12  
2.1.1 Neighborhood Generation 13  
2.1.2 Spatial Outlier Detection Algorithms 14  
2.2 Data Visualization 17  
2.3 Density Maps 20  
2.4 Related PS Applications 22  

CHAPTER 3 P-SENSE AND A FRAMEWORK FOR PS SYSTEMS 24  
3.1 P-Sense: A PS System for Air Pollution Monitoring 24  
3.1.1 Sensing Devices 25  
3.1.2 First-Level Integrator 26  
3.1.3 Data Transport Network 28  
3.1.4 Server 29  
3.1.5 Pollution Data Collected 31  
3.2 A Framework for PS Systems 32  
3.2.1 Sample Size Determination 33  
3.2.2 Data Collection 34  
3.2.3 Data Verification 36  
3.2.4 Data Visualization 37  
3.2.5 Density Maps 37  

i
LIST OF TABLES

Table 4.1  The accuracy and range of the pollution sensors. 42
Table 4.2  Parameters used in the experiments. 59
Table 5.1  Maximum $R^2$ for each variable with the corresponding grid size and delta time parameters. 92
Table 5.2  Configuration of the expert predictors for all the variables. 93
Table 6.1  Coefficient of determination $R^2$ and total number of users. 106
**LIST OF FIGURES**

| Figure 1.1 | Resolution of the traditional system (each cell is $2.5 \times 2.5$ Km) versus the PS system (each cell is $50 \times 50$ m). | 3 |
| Figure 1.2 | Participatory Sensing areas of research. | 4 |
| Figure 1.3 | CO interpolated using real data collected at the USF campus, with (left) and without (right) data verification. | 7 |
| Figure 3.1 | P-Sense’s system architecture. | 25 |
| Figure 3.2 | The ArduinoBT board and the environmental sensors utilized in P-Sense. | 26 |
| Figure 3.3 | Graphical user interface. | 27 |
| Figure 3.4 | Mobile-side software architecture. | 27 |
| Figure 3.5 | A Bluetooth environmental data frame. | 28 |
| Figure 3.6 | Server application software architecture. | 29 |
| Figure 3.7 | Visualization Web service. | 30 |
| Figure 3.8 | Effect of the aggregation process. | 32 |
| Figure 3.9 | A general framework for PS applications and its associated techniques. | 33 |
| Figure 4.1 | A snapshot of the variables of interest over the USF campus after the data verification process. | 41 |
| Figure 4.2 | The generation of the areas of influence. | 45 |
| Figure 4.3 | The hybrid algorithm for data verification and removal. | 46 |
| Figure 4.4 | The local and spatial outlier detection methods. | 48 |
| Figure 4.5 | The two steps of the Hybrid approach. | 51 |
| Figure 4.6 | The neighborhood for the Sparse (left) and Dense (right) locations. | 52 |
Figure 4.7 The ratio between dense and sparse locations versus the conspiracy percentage.

Figure 4.8 $R^2$ versus the conspiracy level (left) and the conspiracy percentage (right) for all four variables and the three algorithms.

Figure 4.9 The execution time of the three algorithms for data verification.

Figure 4.10 The $R^2$ for the cooperative attacks where only some of the users surrounding the center cooperate.

Figure 4.11 The $R^2$ for the cooperative attacks where all of the users surrounding the center cooperate.

Figure 4.12 The effect of the verification process in the final interpolation for temperature, RH, CO$_2$ and CO.

Figure 5.1 Cliques definition.

Figure 5.2 Experimental semivariogram and the fitted curve of an isotropic spherical model.

Figure 5.3 The map (top row) and surface (bottom row) representation of the synthetic data created for the experiments.

Figure 5.4 A comparison between the results for the spatial interpolation using kriging (top row) and MRF (bottom row).

Figure 5.5 $R^2$ for MRF under different conditions of noise and missing ratios.

Figure 5.6 $R^2$ for kriging under different conditions of noise and missing ratios.

Figure 5.7 An example of the result of spatial interpolation using kriging.

Figure 5.8 The proposed kriging plus PCA/ICA method, versus multivariate cokriging.

Figure 5.9 Contour map for the spatial interpolation using kriging with PCA (top row) and kriging with ICA (bottom row).

Figure 5.10 Modeling time using an additional dimension.

Figure 5.11 $R^2$ as a function of the grid size and the delta time.

Figure 5.12 Known and predicted values after using kriging interpolation along with PCA and time information.

Figure 6.1 Contour maps for all variables (temperature, RH, CO$_2$, CO, air quality, and combustible gasses) in the USF campus.
| Figure 6.2 | Gradient field and contour maps for CO$_2$ in the USF campus. | 98 |
| Figure 6.3 | The linear mapping between the Mean Magnitude of Change (MMC) per region, to the number of locations per region. | 100 |
| Figure 6.4 | $R^2$ for all variables in the univariate case using CO as the variable of control. | 102 |
| Figure 6.5 | Location of the users after 10 iterations for all variables, when using only CO as the control variable. | 104 |
| Figure 6.6 | Increasing the number of users ($max_{loc}$) per region. | 105 |
| Figure 6.7 | $R^2$ for all variables for the multivariate case, using RH and CO are the control variables. | 107 |
ABSTRACT

Participatory sensing (PS) systems are a new emerging sensing paradigm based on the participation of cellular users in a cooperative way. Due to the spatio-temporal granularity that a PS system can provide, it is now possible to detect and analyze events that occur at different scales, at a low cost. While PS systems present interesting characteristics, they also create new problems. Since the measuring devices are cheaper and they are in the hands of the users, PS systems face several design challenges related to the poor accuracy and high failure rate of the sensors, the possibility of malicious users tampering the data, the violation of the privacy of the users as well as methods to encourage the participation of the users, and the effective visualization of the data. This dissertation presents four main contributions in order to solve some of these challenges.

This dissertation presents a framework to guide the design and implementation of PS applications considering all these aspects. The framework consists of five modules: sample size determination, data collection, data verification, data visualization, and density maps generation modules. The remaining contributions are mapped one-on-one to three of the modules of this framework: data verification, data visualization and density maps.

Data verification, in the context of PS, consists of the process of detecting and removing spatial outliers to properly reconstruct the variables of interest. A new algorithm for spatial outliers detection and removal is proposed, implemented, and tested. This hybrid neighborhood-aware algorithm considers the uneven spatial density of the users, the number of malicious users, the level of conspiracy, and the lack of accuracy and malfunctioning sensors. The experimental results show that the proposed algorithm performs as good as the best estimator while reducing the execution time considerably.
The problem of data visualization in the context of PS application is also of special interest. The characteristics of a typical PS application imply the generation of multivariate time-space series with many gaps in time and space. Considering this, a new method is presented based on the kriging technique along with Principal Component Analysis and Independent Component Analysis. Additionally, a new technique to interpolate data in time and space is proposed, which is more appropriate for PS systems. The results indicate that the accuracy of the estimates improves with the amount of data, i.e., one variable, multiple variables, and space and time data. Also, the results clearly show the advantage of a PS system compared with a traditional measuring system in terms of the precision and spatial resolution of the information provided to the users.

One key challenge in PS systems is that of the determination of the locations and number of users where to obtain samples from so that the variables of interest can be accurately represented with a low number of participants. To address this challenge, the use of density maps is proposed, a technique that is based on the current estimations of the variable. The density maps are then utilized by the incentive mechanism in order to encourage the participation of those users indicated in the map. The experimental results show how the density maps greatly improve the quality of the estimations while maintaining a stable and low total number of users in the system.

P-Sense, a PS system to monitor pollution levels, has been implemented and tested, and is used as a validation example for all the contributions presented here. P-Sense integrates gas and environmental sensors with a cell phone, in order to monitor air quality levels.
Participatory Sensing (PS) is a new sensing paradigm that relies on the voluntary cooperation of cellular users equipped with embedded or integrated sensors. Given the large and increasing number of cellular users, this new data collection method provides the opportunity to address large-scale societal problems like never before [1, 2, 3]. Compared with traditional monitoring systems, PS systems promise to be more cost effective and precise, as the data collector does not have to invest capital in the user equipment nor in its deployment, maintenance, and operation while being able to collect a considerably larger number of samples and from places that cannot be afforded today, providing better spatial resolution and precision in the estimations. Examples of PS systems are applications that collect pollutant measurements to determine the air quality index (AQI) [4], measure radioactivity levels and pollen, real-time GPS locations to assess traffic congestion and travel times, and collect noise, humidity, and temperature measurements to show real-time maps of these variables.

1.1 Motivation

PS systems have several important advantages when compared to traditional monitoring systems. Traditional monitoring systems usually consist of very few static and most of the time expensive monitoring stations located in strategic places in a county, city or state. As a result, with traditional systems, spatial estimations for regions far from the monitoring stations could be too coarse, which might not be very helpful to those individuals who are
very sensitive to the variables of interest, or to those organizations trying to assess, monitor, and control these variables or activities.

On the other hand, PS systems consist of many mobile and cheap stations, limited by the number of cellular phone users located in the region of interest. Therefore, a PS application is capable of providing considerably larger amounts of data and with better spatial resolution, i.e., from many more places, than current measurement systems. It is argued that the spatial resolution that a PS system can provide is a strong motivation for considering this approach as a replacement of traditional measurement systems. By using a PS system, it is now possible measuring, detecting, and analyzing events that occur at different spatial scales, ranging from blocks and neighborhoods, up to counties and states. As an example, Figure 1.1(a) shows the interpolation of Carbon Monoxide (CO) measurements in Los Angeles County in California, which has a high density of static pollution stations, and Figure 1.1(b) shows the interpolation of the data collected using a PS system in the University of South Florida (USF). It is important to emphasize that one little cell in the LA plot is similar in size to the entire area of measurements shown in the USF plot, and those variations and different CO levels in USF are impossible to detect and visualize using interpolations from distant stations. Finally, another important motivation in favor of participatory systems is cost. In a PS system, sensors are meant to be cheap and easy to deploy, and since the units are in the hands of the users, the data collector does not incur in capital, deployment, operational, and maintenance costs.

In the same way as any other real system, before any PS system is deployed and used in a large-scale application, several important research challenges need to be addressed. These challenges, which are categorized in five areas, are described next:

- Privacy: mechanisms are needed to protect the personal information of the user and the places she visited as part of the PS application. If the privacy of the users is not protected, they may decline their participation in the application. For instance, the work in [5] develops a correlated noise model that can be utilized to perturb location-
(a) Traditional monitoring system, LA CO levels. (b) PS system, USF CO level.

Figure 1.1: Resolution of the traditional system (each cell is 2.5 × 2.5 Km) versus the PS system (each cell is 50 × 50 m).

tagged data in a way that protects both user data and the location while ensuring accurate reconstruction of community statistics.

• **Security**: the transmission, processing, and storage of the information must be secured to protect delicate data of the users and provide data integrity. These mechanisms need to do so while being simple and energy efficient, so they can be run in a resource-constrained device.

• **Data Validity**: a PS application must provide a method to verify the validity of the data. For example, in the air pollution context, users may fake the readings of the sensors (e.g., to avoid contamination fines). Mechanisms must be in place to detect short and long-term erroneous measurements. The work in the area of multivariate statistical analysis and diagnosis of air pollution monitoring systems [6, 7] usually does not consider the localization correlation of the measurements that a PS system with massive users would have. Also, the non-stationary characteristics of the air quality measurements imply sophisticated algorithms to infer and detect doubtful measurements.

• **Visualization**: PS applications will collect huge amounts of data, more precisely, 2-dimensional data series with gaps in time and space. Interpolation techniques to
effectively visualize all this information are also needed. This problem is strongly connected with data validity since an inference of a value could help filling the gaps of the series. Techniques, such as kriging interpolation are useful when the correlation between variables, locations, and time (previous measurements), should be considered.

- **Incentives**: an additional problem involves the provision of incentives for participation. Why should I participate in a PS application? Why should I bother to get samples and transmit them? What would I gain? Why should I use my time, my battery, etc., to collect data for others? Appropriate mechanisms need to be included to encourage the participation of the users who will collect the data of interest. The work in [8] is one of the first ones that addresses the incentives problem in PS.

In addition to the software and hardware design and implementation that are also required, all these five areas need to be addressed when designing a real PS application. Figure 1.2 is a pictorial representation of how these areas together create a complete PS system.

![Figure 1.2: Participatory Sensing areas of research. Taken from [9] © 2011 IEEE.](image)

In this dissertation, a PS system called P-Sense [9] is utilized to study PS related problems, with focus on environmental applications. P-Sense brings the possibility of having not only pollution data with considerably better spatial resolution [10], from places not possible
before, such as your community, school, and neighborhood, but also the possibility to assess your exposure to pollution according to the places visited during your daily activities [11]. Further, with the appropriate sensors integrated into a cellular phone, P-Sense promises to achieve these goals in a continuous and invisible manner, as the application can run in the background of cellular users’ phones.

P-Sense has been collecting environmental data (\textit{CO} (ppm), \textit{CO}_2 (ppm), combustible gases (ppm), air quality (4 discrete levels), temperature (F), and relative humidity (%)) across the University of South Florida (USF) campus for several months. This dataset has been used to validate all the proposed methods and techniques presented in this dissertation.

1.2 Problem Statement

Most of the challenges in PS systems are consequences of the fact that now the measuring devices are cheaper (of lesser quality than the static ones) and they are in the hands of the users: 1) the accuracy of the individual samples might not be as good as the quality of the samples from expensive static stations; 2) the measuring devices may fail with higher frequency when compared to the expensive ones; 3) the data are more prone to being intentionally tampered; 4) the cellular phone users need to have a good motivation to participate in the data collection process; and 5) the PS system needs to implement security and privacy mechanisms in order to guarantee the protection of the user’s identity and the integrity of the data.

In addition, there is no unifying approach to guide the design of an entire PS system considering all these challenges in a comprehensive manner. Part of the work of this dissertation is meant to fill this gap presenting a general framework to successfully guide the designer in the creation of a PS system. This framework splits the design process in 5 modules: sample size determination, data collection, data verification, data visualization, and density maps generation modules. The remaining part of the dissertation aims to solve three of these stages of the framework, more specifically, data verification, data visualization, and density maps generation.
The problem of sensor data verification is addressed first, which considers the detection and removal of invalid data, whether intentionally or unintentionally produced, in a computationally efficient manner. The computational problem is very challenging given the large amounts of data and the fact that current inference tools use all samples to produce estimates in other locations. Invalid data can be the result of two possible scenarios: 1) unintentional errors produced by malfunctioning sensors that generate abnormal values; or 2) intentional modifications inserted by the users, e.g., polluters (companies, countries, etc.) who must comply with certain maximum levels of emitted pollutants to the atmosphere in order to avoid penalty fees, and would like to report modified values of the collected pollution data.

The problem of wrong data detection and removal is also important because of its strong connection with the problem of sensor data visualization and analysis. Visualization is largely used to find spatial patterns on the variables [12]. In general, inconsistent or invalid data can be the reason of an incorrect (or biased) analysis of the global status of the variable due to the strong influence of a small set of data in the statistical model [13]. Figure 1.3 shows this problem when the same interpolation technique is applied to the data with and without the removal of the outliers. From the graph, it can be seen how the variable of interest, in this case the spatial interpolation of carbon monoxide data collected in the University of South Florida (USF) campus, is completely different when the outliers are detected and removed (left) than when data verification is not performed (right).

After looking at Figure 1.3 one could argue that particular outliers could be visually detected and manually removed from the data, however, it is better if an automated procedure can take care of these problems. Further, in some cases, such as those maliciously introduced by the users, the visual detection might not be so obvious. This also shows the importance of creating an application that automatically detects and removes outliers using not only the location information but also the behavior of the variables [11].

Although it could be argued that a PS system might not need of data interpolation techniques, the reality is that the density and mobility of the users and the user selection
Figure 1.3: CO interpolated using real data collected at the USF campus, with (left) and without (right) data verification.

techniques given by incentive mechanisms for participation, produce time series with many measurement gaps both in time and space. Thus, interpolation techniques are also needed. However, current interpolation techniques, might not be appropriate. Since the dataset collected by PS systems normally consists of measurements from multiple variables at different locations and times, these interpolation techniques should be modified to successfully exploit the advantages that such a dataset brings. When compared to the spatial interpolation of a single variable, including the correlation among variables, close locations and time, strongly improves the quality of the final estimations.

This dissertation also addresses the problem of how to define the ideal location of the users, which for the first time is connected to the problem of incentives. The question to be answered is the following: given the large number of potential participants, where should the system take samples from, or which cellular phone users should the system select to take samples from, so that the variable of interest is well characterized and at a low cost? Once this question is answered, incentive mechanisms should use that information to motivate those users, in a region of interest, to actively participate and collect the data. Although some incentive mechanisms in the literature have included some embedded algorithms to address this particular question, most of them focus on one particular aspect, like maintaining the level of user participation over time, lowering the total sampling cost,
or covering the area of interest [14], and none has determined the number and locations of
the participants based on the quality of the variable being sampled.

1.3 Contributions

The main contributions of this dissertation are presented next. Each one of these con-
tributions will be elaborated in depth in subsequent chapters.

1.3.1 A Framework for the Design and Implementation of PS Systems

The first contribution of this dissertation is the design of a framework to guide the design
and implementation of PS systems that considers the challenges explained before [15]. The
proposed framework splits the design of PS application in five modules: sample size deter-
mination, data collection, data verification, data visualization, and density maps generation
modules. Chapter 3 describes this framework in detail.

The following sections present the remaining contributions, that are mapped one-on-
one to three different modules of this framework: data verification, data visualization and
density maps.

1.3.2 Data Verification

Chapter 4 of the dissertation includes mechanisms to detect and remove invalid data,
whether introduced by malfunctioning sensors or malicious users. The detection and removal
of outliers as a result of malfunctioning sensors that generate abnormal values is fairly simple
if multiple measurements for the same location are available. A solution to this problem
is proposed based on the Mahalanobis distance on multivariate data [16, 17]. The results
show that this technique is very successful removing this type of outliers.

On the other hand, intentional modifications to the data by the users are harder to
detect. To address this problem, known spatial data mining techniques for spatial outlier
detection and removal are utilized, and a new hybrid method based on the classification of
the data in neighborhoods is proposed and implemented [18]. An observation is considered a
spatial outlier if its value is notably different compared to the values of the locations around it [19]. Pollution data are classified into Sparse (distant neighbors or heterogeneous measurements) or Dense (nearby neighbors and homogeneous measurements) neighborhoods, considering spatial (location of the samples) and non-spatial (environmental variables) information. A more complex adversary model is presented in [20] where now the users cooperate with each other in order to confuse the data verification system, but the hybrid algorithm still successfully detects and removes the spatial outliers in this scenario.

This new method considers different characteristics of a typical PS application, such as the density of the neighborhood of each location and the behavior of the variables to reconstruct the original variables regardless of the intrusion of malicious users modifying their measurements. The results indicate that this new hybrid approach detects malicious modifications of the data in a very efficient manner with conspiracy levels of up to 22% of the total number of users, and it is capable of reconstructing the variables of interest even if the users are cooperatively conspiring against the system. This hybrid approach also reduces the computational time applying computationally expensive techniques for invalid data detection only in those neighborhoods that make sense while using very inexpensive and simple techniques in the rest of the neighborhoods.

1.3.3 Data Visualization

Depending on the dataset under analysis, the optimal data interpolation technique can be different. Having this in mind and considering different density of users in the area of interest, two techniques were studied. Kriging [21, 22] and Markov Random Fields (MRF) [23] are widely used for spatial interpolation and image restoration, respectively. This study shows that because of the kriging computational requirements, MRF are preferred when having a high density of measurements in the space, but kriging clearly outperforms MRF in the case of low density conditions (a low number of measurements in the map), since it generates estimations with a higher coefficient of determination. Since for the P-Sense
case, and almost for any PS application, the density of measurements is really low, kriging is utilized for the spatial interpolation process.

Kriging is utilized to interpolate the data collected by P-Sense, using one (univariate) or multiple (multivariate) variables [24]. The visualization results for the univariate case show that the spatial resolution of the information is quite good, which is now very useful for those people sensitive to the variables of interest, or those agencies or organizations in need of more precise data. The multivariate case of kriging is very important because correlations among the variables are supposed to provide additional information that should improve the quality of the estimations. However, the use of cokriging (the multivariate extension of kriging) is very expensive [25], especially for systems with large amounts of data, like PS systems. To address this problem, the use of kriging along with Principal Component Analysis (PCA) and Independent Component Analysis (ICA) for multivariate spatial interpolation is proposed. Finally, a new method to make interpolations in space and time is presented, which is more appropriate for PS systems.

The results indicate that the accuracy of the estimates improves with the amount of data, i.e., one variable, multiple variables, and space and time data. Also, the results clearly show the advantage of a PS system compared with a traditional measuring system in terms of the precision and spatial resolution of the information provided to the users. With a PS system like P-Sense, it is now possible to detect critical environmental conditions that strongly impact different aspects of life in a precise, unobtrusive, and cost-effective manner.

1.3.4 Density Maps

This dissertation also proposes the use of a separate module in PS systems to calculate and generate density maps [26]. Density maps will provide incentive mechanisms input information for them to encourage the participation of users located in those specific areas where it is more important to take samples from, so the variable of interest is properly characterized (provides good information), and lowers the cost, avoiding selecting users who do not provide new or additional valuable data.
Once the data have been collected and spatially estimated, the PS system should be able to generate the desired density map based upon the measurements and the location of the users. This map is nothing more than the desired locations from where to collect measurements in the next iteration. The idea behind the density maps is based on the observation that if a variable is fairly constant in a region of the map, a low density of users would be required to reconstruct it. On the other hand, if the pollutant has high variability in a region, more locations (measurements) would be needed to characterize it. An univariate and a multivariate mechanism to generate these maps based on the magnitude of the gradient, has been proposed, implemented and tested. Consequently, the incentives mechanism should encourage the displacement of the users by raising or decreasing the amount of reward, in order to follow the requirements established by the density maps. The experiments show how the density maps greatly improve the quality of the estimations while maintaining a stable and low total number of users in the system.

1.3.5 Structure of the Dissertation

The rest of the dissertation is structured as follows. Chapter 2 contains the literature review on areas related to data verification, data visualization and density maps, and also PS systems that are relevant to this dissertation. Chapter 3 presents the details about P-Sense and the proposed framework for PS applications. Following this framework, the subsequent chapters are mapped one-to-one to some of its stages. Chapter 4 details the problem of data verification and the proposed solutions. Chapter 5 presents the techniques developed for data visualization and interpolation in PS applications. Chapter 6 introduces the concept and techniques related to density maps. Finally, Chapter 7 concludes the dissertation and presents possible areas of future research.
CHAPTER 2
LITERATURE REVIEW

A significant amount of work has been done defining the general aspects of Participatory Sensing, including general architectures, the role of the users, incentive mechanisms, privacy and security issues, data management, and general strengths and challenges [1, 27, 28, 9, 29]. However, the work in the specific areas of data verification, data visualization and density maps in the context of PS systems is poor, or rather, non-existent. The following sections will introduce the most significant works in these three different areas, establishing the pros and cons if used on a PS context, and how the work presented in this dissertation solves the challenges related to the areas of data verification, data visualization and density maps. Some interesting applications, related to pollution monitoring and PS systems are also presented at the end of the chapter.

2.1 Data Verification

The problem of data validation has not been addressed yet in the context of participatory sensing. There is work on data verification in similar areas, such as wireless sensor networks, but this work cannot be applied directly to PS systems. For instance, a survey made by Chandola et al. in [30] presents the challenges for anomaly detection in sensor networks, generated by sensor fault or intrusion detection. Some of these challenges are the different types of data that are being collected, the noisy environment, non-accurate sensors and transmission channels. A multivariate approach to solve the problem of data verification in sensor networks is presented in [31] where local outliers, called elliptical anomalies, are removed in a distributed way. While some of these requirements still apply, in PS
applications the processing of the data is not made in the mobile devices and then the com-
putational restriction of the devices and the online/distributed fashion are not considered
now.

In this section, the main focus will be on the most relevant work in the areas of neigh-
borhood generation and spatial outlier detection algorithms. This work was the initial
point of most of the contributions in this dissertation in the area of data verification for PS
application.

2.1.1 Neighborhood Generation

The idea of finding a neighborhood for each location has been considered before but in
different contexts. Depending on the system, the requirements might change and different
techniques need to be applied. In the research area of geology [32], the nearest neighbor
method of strain analysis is evolved by using the Delaunay triangulation to calculate the
natural neighbors. This technique has proven practical and computationally efficient when
compared to similar techniques, such as the Fry or the Crespi methods.

In the area of boundary elements simulation, the Petrov-Galerkin method [33] which is
based on the Voronoi maps (a dual graph of the Delaunay triangulation) and the Delaunay
triangulation, is compared to the finite element method (FEM), a well-established numerical
method used in different fields. It was shown that the numerical results agree with the
ones found by using FEM, but being more computational efficient because of the use of
the Delaunay triangulation. The Delaunay triangulation has also been used for spatial
interpolation, being the basis of the Height technique as presented in [34].

In the specific area of spatial outlier detection, initial approaches used k-nearest neigh-
bors to define the neighborhood for each location, but these approaches did not consider
the spatial or non-spatial attributes of the data to generate the neighborhoods. Taking this
into consideration, more sophisticated techniques has been designed. The work in [35] uses
self-organizing maps (SOM) to define the neighborhood of each location. Voronoi maps (the
dual graph of the Delaunay triangulation) are used in [36] to generate what they call micro
and macro neighborhoods, and the work in [37] utilizes the Delaunay triangulation in the
generation of the neighborhoods. In [38], neighborhoods are generated using a technique
based on random walks only on the spatial attributes.

As it can be seen, all these previous techniques rely on the spatial attributes of the data to
generate the neighborhood for each location. As part of this dissertation, a hybrid algorithm
classifies each location into a Dense (close neighbors with similar measurements) or a Sparse
(distant neighbors or with very different measurements) neighborhood. In order to do this,
the generation of the neighborhood should also consider the non-spatial attributes. Having
this in mind, the use of a clustering technique on the non-spatial attributes is also included
in order to successfully classify each location.

The proposed hybrid technique calculates an initial estimate of the neighborhood using
the Delaunay triangulation and also utilizes a Gaussian Mixture Model (GMM) as the clus-
tering algorithm on the non-spatial attributes. GMM have been identified as an important
promising machine learning technique for pattern recognition [39]. For instance, the work
in [40] presents a comparative study where GMM outperforms radial basis functions and
similar models for voice recognition. In [41], a GMM is utilized to detect clusters embedded
in different feature subspaces, proving its great performance on different standard datasets.
Even in the area of PS applications, Gaussian Mixture Models have been used along with
Support Vector Machines (SVM) in order to infer the user’s context to properly control
the way the data can be collected [42]. In this dissertation, Gaussian Mixture Models are
utilized instead of a simple clustering algorithm, such as K-means, since a GMM takes
into consideration the variability of the attributes through the integration of the covariance
matrix.

2.1.2 Spatial Outlier Detection Algorithms

Extracting interesting and important patterns from spatial data is more difficult than the
Corresponding patterns from traditional numerical and categorical data due to the spatial
autocorrelation, relationships and data types [12]. This complexity limits the usefulness
of traditional data mining techniques. The data inputs of spatial data mining have two
distinct types of attributes: spatial and non-spatial attributes. The relationships among
non-spatial attributes are explicit, for instance, ordering, membership, subclass, etc. On the
other hand, the relationships among spatial attributes are often implicit, such as, overlap,
intersect, etc. This particular property implies the use of special data mining techniques.

Spatial statistics (also known as geostatistics) literature provides two different types of
test for detecting spatial outliers: graphical and quantitative tests. Graphical tests, based
on visualization data, work on highlighting outliers in order for a trained analyst to detect
them. Some examples of graphical tests are variogram clouds and Moran scatterplots.
These methods are limited by the lack of an exact criteria to classify the spatial outliers. On
the other hand, quantitative outliers present a concise technique to detect outliers based,
mainly, on the comparison of each data to its neighborhood [12]. Due to its advantages, the
quantitative approach was utilized in this dissertation in order to detect spatial outliers,
and therefore, verify data.

After generating a neighborhood, all the neighbors have the same impact in the estima-
tion of the real value for that location, and this is not a problem when a neighborhood is
Dense, in the sense of having close neighbors with similar measurements. However, when
the neighbors are really far away from the estimation point, they should not affect in the
same way to the estimation of the real value for that location. This is the reason why, when
creating the neighborhood, not only the spatial but also the non-spatial attributes should
be considered. Two main aspects are important for the analysis of the previous work in
the area of spatial outliers detection and removal: does the technique consider univariate
or multivariate non-spatial attributes? and how is the neighborhood generated?

In general, a global approach, where an outlier is defined as an observation which appears
to be inconsistent with the remainder of the data, is not utilized. A value that appears to
be an outlier from a global point of view, might not be an outlier in its vicinity, and on the
contrary, it can be representing an important and interesting event to be considered [43, 44].
A considerable number of papers [16, 44, 12] introduce the general concepts on spatial outlier
detection and removal by creating a comparison between the spatial attributes and the estimations made by the neighborhoods of each location. This technique is also known as local differences. These initial approaches only consider one variable and the generation of the neighborhood for each location is based solely on distance, e.g., the k-closest neighbors. Another univariate approach is presented in [36], but in this case, the generation of the neighborhoods is through the use of Voronoi maps.

The technique presented in [45] uses a local differences approach, with a Mahalanobis distance criterion, but again, the generation of the neighborhoods relies only on distance (k-closest neighbors). The work in [35] also presents a multivariate approach to detect outliers through the use of the Mahalanobis distance, but the definition of the neighborhood only considers spatial attributes through self-organizing maps. After testing, it was found that these techniques based only on the Mahalanobis distance performed poorly and were really sensitive to the final tuning of the parameters. The work in [38] uses random walks and K-means to estimate the neighborhood, but the final technique to calculate the neighborhood function per location remains the same across all locations. Another inconvenient of the previous technique is the sensitivity to the selection of the number of neighbors, and the univariate nature of the approach.

The work in [13] presents an interesting use of the local kriging technique along with local differences, similar to the approach presented in this dissertation, but it is designed only for the univariate case and it does not estimate a proper neighborhood for each location. Other disadvantages are the univariate nature of the approach, and that the authors assume the possibility of grid sampling over the area of interest, something that might not be possible in a PS scenario where the users are moving freely.

The algorithms presented in [46] use ordinary kriging in an incremental global approach to determine the spatial outliers. However, it only applies for the univariate case and it depends on the estimation of a clean initial dataset (a dataset with no spatial outliers whatsoever), which might not occur in cases when a lot of users are conspiring (tampering
data) against the system. The work in [19] extended this approach to the multivariate case by using cokriging, but it again depends on the selection of a good initial dataset.

Finally, the work in [43] gives a general framework for spatial outlier detection estimating a neighborhood based on spatial and non-spatial attributes, but it does not consider the variability of the attributes when using K-means nor a different neighborhood function for each location. It is important to note that none of these algorithms considers an irregular density of the users as it is done in this dissertation, or in the best case they consider a semi-regular grid.

For the area of data verification, this dissertation presents a technique that not only considers multivariate non-spatial attributes, but also generates the neighborhood based on spatial and non-spatial attributes of the data. Moreover, this technique is computational efficient by classifying each location into Dense or Sparse, and then applying robust and computational demanding technique only when necessary.

2.2 Data Visualization

The specific problem of spatial interpolation in the context of participatory sensing applications has not been addressed yet, but in this section a review of the related work and similar techniques is presented. In general, the focus of this section will be on predictive monitoring of pollution data, and the kriging technique, with some extensions considering multivariate data and computational efficient execution.

It is necessary to consider the special characteristics of these applications since the mobility of the users and the use of noisy sensors produce a low density of measurements in time and space. There is work on the area of data inference to fill measurement gaps and predictive monitoring on time series, but most of these approaches only use the information from one location. In this dissertation, kriging a spatial interpolation technique is used, and some modifications are also proposed in order to include information from multiple variables and time.
It is interesting to have a general idea of the work on predictive monitoring on environmental data. The work in [47] includes an illustrative example to show how to use PCA for air quality variables analysis. In [48], the authors include a statistical analysis of air pollution data using meteorological information using factor analysis and VARIMAX rotation to detect underlying behaviors in the variables. The work in [49] includes the VARIMAX rotation to the work on PCA, and the eigenvalues are the criterion to define the number of principal components. A recent work in [6] presents a predictive monitoring system that is able to detect abnormal fluctuations of the data, but they only considered data from a single station.

Data visualization is essentially a problem of spatial inference based upon irregularly spaced points of measurements. This is specially interesting for pollution monitoring agencies in order to predict the effects of emissions within non-monitored areas [11, 50]. Many of the traditional monitoring stations are positioned to detect high concentrations of the pollutants and support decision making relative to these values. Nevertheless, when these values are viewed in an independent way, they give an incomplete picture of the potential pollution problems even in areas of high monitor density. The work made in [51] gives a nice presentation of the general aspects, the importance and benefits of data extraction and visualization for wireless sensor networks (WSN), which gives a good reference point to PS application, even though mobility of the users imposes stronger restrictions. For the spatial interpolation process, the simple Sheppard technique is used on a single variable. Finally, the work in [52] presents an implementation of an online web-based visualization system for sensor data that considers the different resolution of the sensor data and also plots the mobility of the sensors. All this previous work shows the importance of appropriate spatial interpolation techniques that consider the special characteristics of PS data.

The kriging technique is known as a BLUE (Best Linear Unbiased Estimator) interpolator, i.e., it is a linear estimator that matches the correct expected value of the population (unbiased), also minimizing the variance of the observations (best) [53]. Nevertheless, there are other spatial interpolation techniques that could be used for much simpler scenarios,
such as, electronic imaging systems [54], stochastic interpolation using radial basis functions [55], bayesian inference [56], splines [57, 58], just to cite some; but after considering them, all of these options failed for not considering either multiple variables, or time correlation, or low density of points. That is the reason why an initial comparison between a well known and widely used probabilistic technique for spatial interpolation and image reconstruction, Markov Random Fields (MRF), and kriging was made first.

Kriging and Markov Random Fields (MRF) are widely used techniques for spatial interpolation and image restoration, respectively. For instance, a comparison in the use of kriging and MRF for the prediction of particulate matter PM$_{10}$ concentrations around Pittsburgh was made in [59]. Some algorithmic techniques are presented in [60] to improve the belief propagation in Markov Random Fields for image restoration. A good introductory material about MRF and its application to image processing can be found in [61], and the work in [23] introduces the basic theory about MRF, stochastic relaxation and the annealing scheduling for image restoration. The work in [62] uses GMRF for spatial interpolation, and a procedure to estimate the parameters of the model is presented.

The work presented in [53] gives a clear presentation of the kriging technique, its computational problems, and the implementation difficulties. Also, the work in [63] is an interesting starting point to understand most of the details about geostatistical tools, such as, kriging and cokriging. The analysis for cokriging shows the practical difficulty in specifying all the parameters for the cokriging case. Even though it was not used, the work in [25] presents an implementation of cokriging. Finally, in [64] the authors give an insight on important considerations when selecting kriging for the analysis of experimental data. For a complete review of the kriging technique, its aspects and applications, the interested reader can follow the work in [65, 66].

As it was mentioned before, the recent work around the kriging technique has focused on its computational time and memory requirements. Although the time constraint for a PS application in air pollution is not too strong, the ideas and tools in this work are useful. The work in [67] presents an interesting mixture of kriging with Gaussian Markov
Random Fields (GMRF) to partially solve the computational problem of kriging, but the time correlation between multiple variables is not included. The work in [68] presents an efficient implementation of the kriging technique for real-time applications. In a similar way, an efficient implementation of kriging via matrix vector products is presented in [69]. None of these previous works considers multivariate information in the spatial interpolation process.

In order to consider the multivariate characteristic of PS data, the use of PCA and ICA along with kriging has been utilized in this dissertation. Nevertheless, there is work on the usage of kriging and PCA for these purposes. The work in [70] uses fixed measuring stations to monitor the temperature on water streams in a temporal climatological scale. Principal component analysis is employed to construct the physiographical space and build semivariograms for characterizing the correlation structure. In [71] the authors also use principal component analysis for summarizing the relationships among the soil variables at different spatial scales. In the area of soil properties analysis, the work in [72] uses principal components along with kriging to interpolate fields of 16 soil variables. In [73], kriging and PCA are used to assess the horizontal and vertical distribution and transport of an industrial contaminant in soil.

The work presented in this dissertation differs from these others in the use of not only principal component analysis (PCA), but also independent component analysis (ICA) along with the traditional kriging technique. Further, in this work not only one variable, but multiple variables and interpolation in space and time, are presented.

2.3 Density Maps

Most of the previous work on PS system do not analyze the importance of the location of the users in the finally quality of the estimation of the variable(s) of interest. In general, the work has been focused on how the users are encouraged to participate in the collection process and how the recruitment process selects the “best” users for the system. There are similar applications on pollution and environmental monitoring, such as [74, 75], that
keep track of the behavior of the participants and their measurements, in order to estimate (directly or indirectly) the variable of interest in other locations. While this approach can be useful for some applications, it can poorly estimate or completely miss events that occur in small localities or at different scales.

Some applications are starting to consider the location and mobility of the users in the data collection process. In [76] the authors present a set of metrics in order to evaluate the individual participants’ fit with any given project, which can be useful in order to improve the quality of the collected data. This approach can keep a profile for each user, creating at the same time a set of “trusted” users, which can be useful for the data verification process.

The work in [77] considers the mobility profile of the users and the spatio-temporal coverage to create coordination services, enabling the efficient selection of the users. It also considers the level of commitment of each participant to modify the selection process. A disadvantage of this system is the lack of feedback to the incentives mechanism. Considering this, the work in [78] improves the previous application feeding back information from the current performance of the system for the recruitment process in the next iteration, something that is also considered in this dissertation since is very important for the correct reconstruction or estimation of the variables of interest. One strong assumption is the complete coverage of the area by the users, something that might not always happen.

As it can be seen, most of the current PS applications simply collect the information from the participants without considering any quality metric based upon the performance of the estimation process. In this dissertation, not only the locations, but also the value of the measurements are considered in order to generate a set of desired locations of the users (density maps). These locations should be used by the incentives mechanisms to collect the data, either by selecting only a subset of the users or by encouraging their displacement.

Through the work on density maps, this dissertation has proven that the displacement of the users is imperative for a PS system. In order to achieve this, incentives mechanism using reverse auctions, like the one presented in [79], could be utilized to encourage this desired displacement, without hurting the privacy protection of the user. By offering a
higher reward for the regions were an increase of the number of users is needed, and a lower reward for the regions with low density requirements, the users would be forced to move to the desired locations.

This data collection can be performed in two ways: opportunistic or interactive [80]. By using an opportunistic data collection, the users are not aware of the exact moments at which the data will be collected and, in general, will not require its interaction. On the other hand, using the interactive collection, the users would have to actively interact with the cellphone in order to measure the variable of interest. The algorithms proposed in this dissertation will not be affected by the way the data are collected, since the incentives mechanisms and the phone applications deal with these issues.

2.4 Related PS Applications

This last section presents some Participatory Sensing systems that in one way or another are connected to the work developed in this dissertation. The focus is mainly on PS applications addressing the problem of pollution monitoring, in the same way P-Sense [9] does it. The work in [81] presents an interesting comparison between sensor networks and sensor grids for air pollution monitoring. This work is a good reference to show the advantages of a participatory sensing approach compared to these static networks.

Without using gas sensors, some work has focused on estimating the air quality in indirect ways. In [74], by tracking the behavior of the users, the authors can indirectly estimate the air quality levels in a region of interest. In a similar approach, the PEIR project [82] relies on demographics and location data to estimate the environmental exposure of the users. The PS system presented in [75] tracks the behavior of car drivers in order to estimate the most efficient paths for gas saving, allowing drivers to find the most fuel efficient routes between two arbitrary points in the map. Even though it does not include a direct or indirect measuring of the pollution levels, it shows the importance of “green” approaches to reduce the pollution levels in the cities.
The initial approaches to monitor air pollution utilized sensor arrays, GPRS modems and static networks. In [83] the authors present the challenges and importance of air pollution monitoring, and executed a field study, recruiting 7 taxi drivers and 3 students in Accra, Ghana to collect data with a GPS logger integrated with SO$_2$ and NO$_2$ sensors. The work in [84] presents an array of sensors with GPRS modems capable of collecting air pollution data (CO, NO$_2$ and SO$_2$) and send it back to an Internet enabled pollution server. In a similar way, the work in [85] presents a GPRS based system to collect atmospheric pollution data (CO, CO$_2$, SO$_2$, NO$_2$ and O$_3$), also generating email and SMS messages if toxic levels are detected. The main problem again is the fact that the location of the sensors is fixed. However, the approach made with P-Sense exploits the capabilities and mobility of cell phones, integrating them with multiple gas sensors, bringing an additional flexibility compared to static arrays of sensors.

The work in [86] presents a handheld device that collects pollution variables (CO, NO$_x$ and O$_3$), and by integrating a GPRS modem and a GPS module, it reports air quality levels. In the Haze Watch project [87], cell phones and pollution sensors (CO, O$_3$, SO$_2$, and NO$_2$) were integrated in order to measure the air quality, and then transmit the data to a database server. It also allows a web interface overlaying pollution maps on Google maps, in a similar ways like P-Sense does.

The work in [88] proposes a mobile system that monitors carbon monoxide and also includes a good analysis of the visualization problem. Also, in the area of data visualization, the work in [89] presents a framework for Web-based visualization of a large dataset of mobile sensing devices for air pollution.
CHAPTER 3

P-SENSE AND A FRAMEWORK FOR PS SYSTEMS

In this chapter, P-Sense, the PS system that was developed to collect environmental data in order to have a real dataset for all of the experiments, is presented. It also helps introducing most of the problems related to real PS applications. After introducing P-Sense and all its technical details, a new framework for PS applications is introduced.

3.1 P-Sense: A PS System for Air Pollution Monitoring

P-Sense is a participatory sensing system for air pollution monitoring and control. P-Sense provides large amounts of pollution data in time and space with different spatial resolutions. With P-Sense, government officials will have data to monitor and control the Air Quality Index (AQI) [4, 90] of a city, state, or country; doctors will be able to correlate respiratory problems of their patients to the AQI they are exposed to during their daily activities, in the places they work and live; county officials, community developers, and real state agents will have data to determine the best place where to build a new school, hospital, or community and advertise properties according to the AQI where they are located. P-Sense achieves this massive data collection in time and space in a simple and cost-effective manner. Figure 3.1 shows P-Sense’s system architecture. As it can be seen from the figure, P-Sense consists of four main components: the sensing devices, the first-level integrator, the data transport network, and the servers. P-Sense is based on the G-Sense system architecture presented in [91] and the doctoral dissertation of Alfredo Perez [92]. Each of the components of this architecture are described next.
3.1.1 Sensing Devices

The environmental data are collected by a set of individual external sensors integrated in a board and working like one single sensor device with multi-sensing features. These sensors, integrated to the cellphone via Bluetooth, become the mobile sensing devices or the mobile wireless sensor network. The sensing system can be carried along with the cellphone, or can work as a standalone device due to its independent power supply. Similarly, wireless sensor networks made of static environmental sensors can be integrated into the system. For this case the expensive static pollution stations would be part of this static wireless network. All the environmental data acquired from the sensors are transmitted to a first-level integrator device. P-Sense integrates the following external sensors (shown in Figure 3.2):

- The ArduinoBT development board [93]: a Bluetooth capable AVR based board that integrates digital and analog sensors.
- Gas sensors by Figaro Sensors [94]: carbon dioxide (CDM-4161), combustible gas (FCM-6812), carbon monoxide (TGS-5042), and air quality (AM 1-2602).
- Temperature and relative humidity sensor by Sensirion [95]: SHT-75 digital sensor.
3.1.2 First-Level Integrator

The cellular phone is the first level integrator of the mobile sensing network. In P-Sense a PRO200 Sanyo cellular phone with GPS and Bluetooth capabilities was utilized. It receives the Bluetooth frames coming from the ArduinoBT board and captures the corresponding GPS fixes. This component has the capability to perform initial data analysis and a graphical user interface (shown in Figure 3.3) to show the measurements to the user in real-time and provide feedback. In the case of the static sensor network, another first-level integrator device and the appropriate software interfaces are needed to collect and integrate the environmental data from these networks into P-Sense. This device is the usual sink node available in most wireless sensor networks.

In addition to the graphical user interface, the cellular phone runs an application to manage the reception of data from the sensors, the reception and transmission of data between the phone and servers, and other important functions. This application was implemented using the Java ME platform for resource-constrained devices. Figure 3.4 shows the software architecture of the cellular phone application, which is based entirely upon the work in [92], but only a subset of its functionality is used here. The most important modules of the architecture are described next.
The Sensor Communications and OS Layer module abstracts the communication between the sensors and the mobile device. In P-Sense, the Bluetooth interface communicates with the acquisition board to receive the air quality measurements. A connection is opened with the ArduinoBT module through which the environmental data frames (see Figure 3.5) are constantly received.

The Location Management component is divided into two modules, the location acquisition module and the critical point management module. The former module acquires the GPS fixes, monitors the location information at all times, and displays this information in the GUI. The critical point management module is meant to reduce the amount of traffic.
on the network, and the transfer costs and energy consumption in the client [92], but this functionality is not used in P-Sense.

For the Sensor Management component, the data acquisition module obtains the information from the sensor communication layer, transforms the data into meaningful units (Fahrenheit, percentage of humidity, and parts-per-million) and displays the data for immediate monitoring.

The Server Communication Management component handles the communication with the servers and consists of the communication management and the session management modules. The communication management module provides the way to transfer data from the mobile device to the servers. This communication uses UDP, but other standard interfaces, such as TCP, can be used to implement this data transmission. UDP was selected since it is more efficient in terms of energy consumption when no high reliability is needed. The session management module provides the methods to exchange session data with the servers. A basic session management module was implemented in the system allowing the servers to recognize the identity of the user.

### 3.1.3 Data Transport Network

The first level integrators transmit the environmental data using networks based on IP technology, such as current Internet links available in existing static and cellular networks.
3.1.4 Server

This part of the architecture serves as the data repository for the entire system as well as the platform for data processing, analysis, and visualization. Users of the environmental data will “connect” to these servers to get access and visualize the data.

In order to perform these tasks, the server is equipped with a database for data storage and an application server to develop and run methods for data analysis. The server application was developed using the Java SE platform (version 6 update 13 or better) and deployed using Sun Glassfish Enterprise Server V 2.1. The database utilized were Postgres 8.3.7-1 and Postgis 1.3.6-1 along with the JDBC version 3 drivers for Postgres. For the visualization and data access part, Google Web Toolkit 1.5.3, Google Web Toolkit Maps API, and Eclipse Europa (or newer) were utilized. With these tools, any data user with Internet access and a web browser is able to access and visualize the data in a customized Google map.

The modules of the software architecture of the server application are based completely upon the work in [92] and are shown in Figure 3.6.

![Server application software architecture](image-url)

Figure 3.6: Server application software architecture. Taken from [91] © 2011 IEEE.
The *Mobile Wireless Sensor Network Management* component takes care of the connectivity with the mobile sensing devices. The functionality of this component matches that described in the server communication management component of the mobile-side software architecture.

The *Data Management* component handles the data from the sensors. It consists of two modules: data collection and data visualization. The data collection module obtains the data from the packets and interacts with the database communication module in order to store the data.

The data visualization module allows clients to see the variables of interest in real time. This feature was implemented using the Google Web Toolkit and Web 2.0 tools. The visualization service allows the users to choose the variables to be visualized from a selectable list box. The collected information for one specific variable is displayed over the Google Map using colored markers (see Figure 3.7). The users can get information, such as range of variation, information about the user who collected the data, the time when it was collected and the exact value. If the zoom level is large enough, a new representation is added, showing each value as a colored circle shadow around the point.

![Figure 3.7: Visualization Web service. Taken from [9] © 2011 IEEE.](image)

The *Database Management* component allows the definition, implementation, access, and maintenance of the information in the database. The database communication module provides access to the database, including insertion and query operations. To perform these
tasks a Java Database Connectivity (JDBC) resource was created in the server application. The spatial and relational database module consists of the specific database to store the system data. In this system the data definition is based on PostgreSQL and its extension PostGIS, which adds support for geographic objects. SQL is the manipulation language and pgAdmin III the software engine.

3.1.5 Pollution Data Collected

The campus of the University of South Florida, in Tampa, Florida was the location for the collection of the data. The area of the campus is approximately $3.9 \text{km}^2$. During three months data were collected 3 times a day for one hour, 3 to 4 days a week. Two users carrying the sensor board and the corresponding cellular phones measured the air quality on predefined points in the campus. These points were defined in order to have an uniform density of measurements around the campus. Since in a real PS system the density of the measurements will not be uniform, the use of incentives is necessary in order to force the users to collect data in the required locations, and in some way, control the density of the measurements per area.

The first assumption for the collection of the data is that all measurements collected through the same hour are assumed to be simultaneous, and belonging to the same snapshot. This helps increasing the number of effective measurements per snapshot (the set of all measurement across the campus) of the system and therefore improve the accuracy of the spatial interpolation process.

P-Sense measures $CO$ (ppm), $CO_2$ (ppm), combustible gases (ppm), air quality (4 discrete levels), temperature (F), and relative humidity (%). Once the measurements are available in the central server, they are aggregated and averaged. In this stage, the data are aggregated based on a virtual grid with equal size cells built on top of the campus map. As it will be shown later, the size of the grid affects the accuracy of the predictions in the spatial interpolation process (refer to Chapter 5). Once the measurements are aggregated, an average is calculated by using all values that fall in the same cell of the grid, and leaving
only one valid value per cell. The aggregation process and the calculation of the average is determined by the data verification process (refer to Chapter 4).

After experimenting, it was seen that the use of an average value per cell, instead of using all possible values, does not affect the performance of the system, but it does improves the speed of the execution of the algorithms. For instance, using the kriging technique for the interpolation of a single point, the use of an average value per cell reduces the number of values used for the interpolation from around 600 to only 20 to 30 values. A complete description of these techniques can be found in Chapter 5 of this dissertation. Figure 3.8 shows the effect of the aggregation stage. All collected measurements are shown at the left; the same scenario but after the aggregation and averaging of the values is shown at the right. The darker circles in the left graph indicate more than one value at that location.

![Figure 3.8: Effect of the aggregation process.](image)

### 3.2 A Framework for PS Systems

The right hand side of Figure 3.9 shows the modules of the proposed framework, which consists of the sample size determination module, the data collection module, the data verification and visualization modules, and the density map generation module. The left hand side of the figure identifies some of the techniques used to address the challenges and goals of each module. Each of the modules are described in more detail next. Some techniques associated to each module are introduced here, and some others will be presented in subsequent chapters as they are part of the contributions of this dissertation.
3.2.1 Sample Size Determination

The Sample Size Determination Module needs to find the appropriate sample size so that a PS system can be comparable in accuracy to traditional measurement systems. By exploiting the large number of cellular users, a PS application should be able to improve the low accuracy of a single sensor by using multiple samples and spatial data aggregation. This aggregation process requires the definition of a grid in the map and a minimum number of samples per cell in the grid. By using the given accuracy of the sensors, the sample size determination module should be able to define the sample size per cell to achieve a determined accuracy level.

The sample size determination is based on the fact that the arithmetic average of a measurement repeated $n$ times approximates to the real value with a standard deviation given by the inverse of $\sqrt{n}$. This means that by having multiple samples the accuracy of the measurements can be increased. Obviously, the number of samples cannot be increased indefinitely, since it would cost more than what is affordable for the system.
The sensors can be characterized using the accuracy reported by the manufacturers or through the standard deviation from the real values. Therefore, given a desired accuracy and confidence level, the number of samples can be determined. The work in [96] uses the Lagrange multiplier method to solve this optimization problem, i.e., the cost of a single measurement versus the overall accuracy. Since it only depends on the sensor’s accuracy, this is a one time calculation and the sample size per cell will not change unless the whole accuracy of the system is required to change.

There are two ways to achieve this number of measurements in practice. The first one is to use the cellular phones to sample the sensors multiple times and then send a complete packet to the server. The second way uses the measurements of several users that are close to each other and aggregates the measurements in space, as explained previously. This increases the effective number of measurements per location, requiring fewer samples from individual users. The aggregation process is also essential to reduce data redundancy, improve data accuracy, and reduce the computational requirements of posterior processing.

3.2.2 Data Collection

The second module in the framework is the **Data Collection Module**, which has to deal with the problems of incentives, privacy, and security for user participation. Without appropriate incentives, users will not be willing to participate. Having a target density map, incentive mechanisms need to collect enough samples per region. Further, these mechanisms need to keep enough users participating at all times, otherwise, the system may die or it may not be able to collect enough samples. Incentive mechanisms need to consider coverage and location as well. It is very important to know where the users are and how much of the area of interest they are covering. The system will not be very effective if most of the samples are collected from users who are located very close to each other. This module is also in charge of protecting the users’ privacy and the integrity of the data. Therefore, this module must include privacy protecting mechanisms, so no one can determine who took the
samples and from where, as well as security mechanisms to guarantee that the data comes from the right user and was not modified by someone else.

In general, users need to gain something, be rewarded somehow, for their participation, otherwise, they will not be willing to carry extra sensors, spend their cellular phone batteries, and incur in extra costs delivering the data. There are several ways in which users can be motivated to participate in a PS system. One obvious method is to offer users some sort of monetary reward for their data. In this category many schemes based on micro-payments [27] and auctions can be found. For example, the reverse auction scheme presented in [79] utilizes virtual credits and return on investments in order to keep prices low and a reasonable number of participants over time. Other methods are based on the benefit that the users receive from collecting the data, as the participants are also the users of the collected data. For example, a person with allergies might be willing to collect pollen data for free if the system provides her with access to the pollen map built by all participants. Other incentive schemes are of a different type, such as entertainment and reputation. If the data collection process somehow is part of a game that users enjoy playing, they might also be willing to participate without any economic reward. Similarly, if the system ranks the participants according to the quality or quantity of information that they collect, the users might participate given the recognition they get among their peers.

The practical implementation of incentive mechanisms is not easy. These mechanisms need to be economically feasible and aware of budget limits to acquire the data; fair, in the sense that all users should be able to participate if they so desire; include mechanisms to attract new participants and keep a minimum level of user participation over time; and cover the area of interest, i.e., encourage users to collect data from the desired locations only, and therefore control the density of the measurements per area. These are only a few of the most important challenges in this area.

Another key issue in PS systems is privacy. Even with good incentives for participation, users might not be willing to collect data if their privacy is not guaranteed, i.e., there should be no possibility to infer the identification and location of the user by looking at
the collected data. Privacy preserving mechanisms are very challenging in PS systems due to the computational and energy limitations of cellular phones and the complexity of the algorithms needed. Privacy can be achieved using encryption, obfuscation, and anonymization mechanisms, and combinations of them [97, 98]. In addition to the energy consumption, the efficacy of these schemes is evaluated based on the degree of privacy protection given by the Information Loss (IL) metric, which evaluates how far the reported locations are from the real ones.

Finally, security mechanisms must be included to guarantee the integrity of the data and make sure that the data come from the authorized users. Digital signatures are commonly used here but more work is needed given the computational complexity and energy consumption of these mechanisms, among the most important issues.

3.2.3 Data Verification

The third module is the Data Verification Module. Since the sensors are cheap units in the hand of the users, PS systems are more prone for unintentional and/or intentional invalid measurements. Unintentional invalid measurements can be attributed to malfunctioning sensors, which report abnormal levels of the variables. Intentional invalid measurements are caused by non-trusted users inserting fake values for their own benefit. This last case can occur, for example, in the scenario where industries must comply with certain maximum levels of emitted pollutants to the atmosphere in order to avoid penalty fees. Therefore, before drawing conclusions from the data, i.e., before applying other procedures to analyze the data (e.g., data visualization), the PS system needs to verify the data and remove these abnormal measurements.

Considering that outliers could come from malfunctioning sensors, which generate out-of-range values, or from malicious users who modify the measurements for their own benefit, appropriate mechanisms need to be included to deal with both aspects. The complete details about data verification are given in Chapter 4.
3.2.4 Data Visualization

The next module is the **Data Visualization Module**, which is in charge of showing the data in an easy to understand and efficient manner. Here, the selection of appropriate interpolation techniques to efficiently visualize the data is very important, as they have different prediction accuracies and computational complexities. Further, considering that the raw data in a PS is a 2-dimensional time series, widely-used interpolation techniques, like kriging, need to be revised to interpolate in time and space.

For the expert analyst, as well for the end-users, a complete visual representation of the variables of interest is desired (see figure 4.1). Since only scattered data across the map are available, spatial interpolation techniques are necessary to estimate the values of the variables for the locations on the map where no measurements exist. For those situations, ordinary kriging, considered the best linear unbiased estimator [68], is the best choice for the spatial interpolation process. However, kriging implementations normally estimate each variable independently and without using temporal information. Given that PS users can be sensing several environmental variables at the same time, it is beneficial to consider multiple variables in the interpolation process because correlations among the variables provide additional information to improve the quality of the estimations. Nonetheless, the use of cokriging (a multivariate extension of kriging) is very expensive, especially for systems with large amounts of data. To address this problem, the use of kriging along with PCA and ICA for multivariate spatial interpolation is recommended. The complete details of these mechanisms will be explained in depth in Chapter 5.

3.2.5 Density Maps

The last module of the framework is the **Density Maps Generation Module**. Once the invalid measurements are removed and the data are spatially interpolated, the PS system should be able to generate a desired density map. This map is nothing more than the desired locations from where to collect measurements in the next iteration, which is based upon the current behavior of the variable. For example, the density of measurements for the
next round should be decreased in a region of the map where the variable/event/factor of interest is fairly constant; however, it should be increased in a region where the variable presents high variability. That is why the output of this module is fed back into the data collection module, i.e., the incentive mechanism should use this density map to increment and decrement the rewards to the users in those particular regions.

As it can be seen from Figure 3.9, the density maps close the loop of the framework providing feedback to the incentive mechanism in the data collection module. In the next iteration, the incentive mechanism should consider this information to encourage the displacement of the users to the regions where new measurements are needed to properly characterize the behavior of the variables.

The idea behind the generation of the density maps is that if a variable is fairly constant in a region of the map, a low number of locations would be required to reconstruct the variable. On the other hand, if the pollutant has high variability in a region, more locations (measurements) would be needed to better characterize it. Considering this, a gradient-based metric is used to define the number of users that are required to properly characterize the variable of interest, taking into consideration a limited budget. The complete details about this module will be presented in Chapter 6.
CHAPTER 4

DATA VERIFICATION

This chapter studies the problem of sensor data verification in Participatory Sensing (PS) systems using and P-Sense, an air quality/pollution monitoring application, as a validation example. Data verification, in the context of PS, consists of the process of detecting and removing spatial outliers to properly reconstruct the variables of interest. This dissertation proposes, implements, and tests a hybrid neighborhood-aware algorithm for outlier detection that considers the uneven spatial density of the users, the number of malicious users, the level of conspiracy, and the lack of accuracy and malfunctioning sensors. The algorithm utilizes the Delaunay triangulation and Gaussian Mixture Models to build neighborhoods based on the spatial and non-spatial attributes of each location. With this neighborhood definition, it can be shown that it is not necessary to apply accurate but computationally expensive estimators to the entire dataset to obtain good results, as equally accurate but computationally cheaper methods can also be applied to part of the data and obtain good results as well. The experimental results show that this new hybrid algorithm performs as good as the best estimator while reducing the execution time considerably.

The chapter is organized as follows. Section 4.1 describes how invalid data is generated. Section 4.2 introduces the proposed hybrid algorithm for data verification and removal, and finally, Section 4.3 presents the results of the experiments and their corresponding analysis.

4.1 Modeling the Generation of Invalid Data

This section describes the models used to generate erroneous data due to sensor malfunctioning and lack of accuracy and malicious attacks. Like it was presented before, P-Sense
was developed in order to collect environmental data to create a dataset that characterizes PS-related problems.

### 4.1.1 Location Sampling

The data collected during one hour on a randomly chosen day is used as the dataset for the experiments presented here. Nevertheless, the training of the parameters was done in different sets for different days of the week and hours of the day. Figure 4.1 shows the measurements of the six environmental variables after applying a spatial interpolation using a multivariate approach with the combination of the widely-known ordinary kriging technique and PCA (Principal Component Analysis) and ICA (Independent Component Analysis) [24] (refer to Chapter 5). The figure shows the fine granularity of the measurements provided by P-Sense, definitively not attainable by static expensive stations located miles away from the campus. This is one of the most important advantages of a PS system when compared with current measuring systems. In this case, it is now possible to identify that the peaks in the $CO_2$ plot corresponded to parking lot areas, definitively with higher levels of the gas compared to the rest of the campus.

In order to be able to properly characterize the problem of location sampling, the data across the USF campus have been spatially interpolated. The interpolated data is used now as the data that the PS system would like to estimate, (see figure 4.1) and the sampling process selects locations and values from these data. In almost all cases for spatial outliers detection algorithms, a constant spatial density and/or a grid distribution of the users/locations is assumed [46, 13, 16, 19]. Nevertheless, this might not be true for a real PS application. With an ideal incentives policy, it could be argued that the location of the measurements can be controlled generating a perfect grid. Nevertheless, on a real application, not all areas in the space can be covered since the users have the tendency to concentrate in specific areas, such as downtown. This problem is presented in more detail in Chapter 6.
In order to model this behavior a Gaussian sampling method has been created. Using four 2-dimensional Gaussian functions, the centers of concentration and the corresponding dispersion of the users are defined through the use of the corresponding means and covariance matrices. Following this spatial distribution, $N$ random locations are generated, and the corresponding values for all six variables are picked from the original interpolated dataset. These measurements are the ideal values, but in the following sections a process for adding noise to model real sensors is introduced. By using these four Gaussian functions a good coverage of the campus is achieved, but still having different location densities (refer to figures 4.5 and 4.6).

It is also worth noticing that on every instance of the experiments, different locations are selected by using the previous sampling method. This approach brings a more complex test scenario, providing dynamics and mobility to the sensors, instead of having a static dataset for all experiments.
4.1.2 Sensor Malfunctioning and Lack of Accuracy

The original measurements of the sensors are modified in two ways, due to lack of accuracy and malfunctioning. In order to model the lack of accuracy of the sensors, once a location is selected, some noise is added to all six variables using an uniform distribution with parameters given by the accuracy of each sensor. In order to model a malfunctioning sensor, a set of locations is randomly selected, and one of the sensor’s measurements is changed to a random value chosen from an uniform distribution with parameters given by the range of the sensor. The values for the accuracy and range are taken from the manufacturers’ datasheets, after the characterization of the sensor, or by considering the natural range of the variables (see Table 4.1).

It is assumed that when an user participates in the application, the cellphone collects more than one measurement per sensor and sends this packet to the central server. Also, more than one user might be participating at the same time in the same area. Considering this, for every selected location, several new noisy measurements are generated (15 for the experiments presented here).

Table 4.1: The accuracy and range of the pollution sensors.

<table>
<thead>
<tr>
<th>Variable</th>
<th>Range</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Temperature</td>
<td>[0 : 150]</td>
<td>±0.54°F</td>
</tr>
<tr>
<td>RH</td>
<td>[0 : 100]</td>
<td>±1.8%</td>
</tr>
<tr>
<td>Air Quality</td>
<td>[0 : 31]</td>
<td>±0.2</td>
</tr>
<tr>
<td>CO₂</td>
<td>[200 : 9000]</td>
<td>±5ppm</td>
</tr>
<tr>
<td>CO</td>
<td>[0 : 1000]</td>
<td>±1ppm</td>
</tr>
<tr>
<td>Comb Gases</td>
<td>[0 : 14000]</td>
<td>±80ppm</td>
</tr>
</tbody>
</table>

After all locations are selected and the noise model is applied to the sensors, a process takes place in which an imaginary grid divides the area in small cells and an aggregation mechanism is applied. The aggregation process is essential to reduce data redundancy, to improve data accuracy, and to reduce the computational requirements of posterior processing [99, 51]. After the aggregation process, a local outlier removal algorithm is executed.
(see Section 4.2) and the average for each variable for each cell is left. For the test scenarios presented here, the size of the cell is approximately $50 \times 50$ meters.

### 4.1.3 Malicious Attacks

In this dissertation, an adversary model is also created, where users maliciously change the values of their measurements. Two cases are considered: one completely random, in which users do not know if anyone else is forging the data, and one scenario in which users attack the system in a cooperative manner.

#### 4.1.3.1 Random Attacks

In order to model random attacks, two parameters are used: *conspiracy percentage* and *conspiracy level*, both in a range of $[0 : 1]$. The conspiracy percentage is the number of malicious users compared with the total number $N$ of participants in the system. A value of 0 represents a system with no malicious users and a value of 1 means that all users are conspiring against the system. Therefore, a conspiracy ratio of 0.3 means that thirty percent of the users are modifying their original measurements. After the data have been aggregated, a set of locations is selected to comply with the ratio defined by the conspiracy percentage parameter, and then the corresponding measurements are reduced by an amount defined by the conspiracy level.

The conspiracy level determines the amount of reduction that the users apply to the acquired measurements. For instance, a value of 0 implies no change to the original measurements whereas a value of 0.4 implies that the measurements of all sensors are reduced by 40% their original values. Only a reduction of the values is considered because this is what makes sense in a pollution application, i.e., no one would like to cheat increasing the pollution level. The data verification mechanisms presented here do not know of the exact value of these parameters or how the data was manipulated, and therefore, the outliers detection process is not trivial.
4.1.3.2 Cooperative Attacks

The previous case presents a simple scenario where users are blind, in the sense that they do not know how the rest of the users are modifying their values. In the previous scenario, due to its random nature, in some occasions these users will cluster and make the data verification a difficult task, but in some other cases they will appear isolated and surrounded by trusted users, making the detection process simpler.

In cooperative attacks, a more complex case is considered where users forge their data in a more intelligent way, cooperating in order to confuse the data verification system and succeed in their objective. In order to model this scenario, two new concepts are introduced: the center of conspiracy, as the point with the maximum level of conspiracy that the attackers will use, and the area of influence, as the region where the malicious users are conspiring. For example, the center of conspiracy might be the location of a factory that would like to fake the pollution values around it in order to avoid penalty fees.

In order to confuse the verification system, the malicious users surrounding this center will cooperate, modifying their values in a decreasing way with respect to the distance to the center of conspiracy. In other words, the closer a user is to the center, the higher the conspiracy value she uses, considering the conspiracy level of the center, as the maximum allowable value. In this way, the closest users will decrease their values in an amount similar to the center of conspiracy, while the furthest users will decrease their value in a really small amount.

Additionally, in this cooperative case, two possible scenarios are created. In the first one, some of the surrounding users of the center conspire, while others will not. This implies that some trusted users will be mixed with malicious users. This scenario is shown in Figure 4.2(a), where crosses represent the locations of the users participating in the system, the centers of conspiracy are represented with a circle, and the malicious neighbors with squares. For the data verification process, this is a simpler condition since the values reported by the trusted users will help detecting the malicious users. The second scenario
4.2 The Hybrid Algorithm for Data Verification

This section describes the new hybrid algorithm proposed in this dissertation for the detection and removal of invalid data in a computational efficient manner. The steps of the entire algorithm are shown in Figure 4.3.

Initially, the data are aggregated in space using an imaginary grid, as explained before. On this new dataset, a local outlier detection algorithm is applied first in order to remove the outliers due to sensor malfunctioning. With the resulting measurements, the average per cell is calculated, leaving one valid value per variable per cell.

The next stage is crucial. Here, a new technique is proposed to classify each location as Dense or Sparse. A Sparse location implies either distant neighbors or a strongly heterogeneous neighborhood, i.e., neighbors with very different measurement values of the same variables. On the other hand, a Dense location implies either nearby neighbors or an homogeneous neighborhood. This classification is achieved using the Delaunay triangulation and Gaussian Mixture Models.
Using this new classification of the locations, the hybrid algorithm calculates the neighborhood function, which estimates the value of a location based on its neighbors. This is one of the fundamental contributions of this dissertation. If the location is classified as Dense, a simple algorithm, such as the median, is used to detect and remove the outliers. Otherwise, multivariate local kriging is used. Finally, after the hybrid algorithm removes the suspicious locations, the remaining locations are used to interpolate the data and measure the quality of the interpolation using the coefficient of determination ($R^2$):

$$R^2 = 1 - \frac{\sum_i (y_i - f_i)^2}{\sum_i (y_i - \bar{y})^2}$$  \hspace{1cm} (4.1)

where $y_i$ are the known values, $f_i$ are the predicted values, and $\bar{y}$ is the mean of the known data. A $R^2$ value close to 1 implies an accurate fit of the estimations with the original variables, and a value of zero, or negative, implies a poor estimation.
4.2.1 Local Outlier Detection Algorithm

Traditional statistical outlier detection methods can be either univariate or multivariate. Multivariate statistical methods are more robust and selective than univariate methods [17]. As presented in [16, 17], a widely used multivariate method is based on the Mahalanobis distance. The Mahalanobis distance takes into account the covariance among the variables in calculating distances. With this measure, the problems of scale and correlation inherent in the Euclidean distance are no longer an issue. The square root $\Delta$ of the quadratic form:

$$\Delta^2 = (\vec{x} - \vec{\mu})'\Sigma^{-1}(\vec{x} - \vec{\mu}) \quad (4.2)$$

is the Mahalanobis distance from the $(d \times 1)$ vector $\vec{x}$ to the $(d \times 1)$ vector $\vec{\mu}$ representing the expected value of the random vector $\mathbf{X}$, and the $(d \times d)$ matrix $\Sigma$ is the variance-covariance matrix of $\mathbf{X}$, $\mathbf{X}$ being the collection of all observations $\vec{x}$. For the density of a $d$-dimensional normal variable, the paths of $\vec{x}$ values yielding a constant height for the density are ellipsoids. That is, the multivariate normal density is constant on surfaces where the Mahalanobis distance is constant. These paths are called contours:

$$(\vec{x} - \vec{\mu})'\Sigma^{-1}(\vec{x} - \vec{\mu}) = c^2 \quad (4.3)$$

These ellipsoids are centered at $\vec{\mu}$ and have axes $\pm c\sqrt{\lambda_i} \vec{e}_i$, where $\Sigma \vec{e}_i = \lambda_i \vec{e}_i$ for $i = 1, 2, \ldots, d$. In other words, $(\lambda_i, \vec{e}_i)$ is one of the $d$ eigenvalue-eigenvector pairs for $\Sigma$. For multivariate normally distributed data the values are approximately chi-square distributed with $d$ degrees of freedom ($\chi^2_d$). Multivariate outliers can now simply be defined as observations having a large Mahalanobis distance. In other words, the solid ellipsoid of $\vec{x}$ values satisfying

$$(\vec{x} - \vec{\mu})'\Sigma^{-1}(\vec{x} - \vec{\mu}) \leq \chi^2_d(\alpha) \quad (4.4)$$

has probability $(1 - \alpha)$. 

47
Using the data aggregation process explained before, the algorithm detects abnormal correlations among the measurements in the same cell. The correlation among all random variables in the same cell should follow the same behavior. Therefore, the Mahalanobis distance is calculated for each measurement in the same cell, and consequently, the algorithm removes the measurements that do not satisfy $\chi^2_d(\alpha)$, the upper 100th percentile of a chi-square distribution with $d$ degrees of freedom. For this case $d = 6$ (the number of variables), and after exhaustive tests, $\alpha = 0.05$ has been found as the optimal parameter to detect around 90% of the local outliers because of sensors malfunctioning. The remaining local outliers get caught by the second processing stage (spatial outliers detection) or they do not affect significantly to the quality of the final estimations.

A graphical representation of this method is shown in Figure 4.4(a), where only CO and combustible gases are used. Points with the same Mahalanobis distance create ellipses (ellipsoids in high-dimensional data), which become the thresholds to detect abnormal measurements, i.e., points outside the ellipse are considered local outliers. Using only the points inside the ellipse, the arithmetic average is calculated, leaving one valid value per variable per cell.

![Ellipses and spatial outlier](image)

(a) Ellipse for the two variable case (CO and combustible gases).
(b) Spatial outlier on the CO data for the USF campus.

Figure 4.4: The local and spatial outlier detection methods.
In the second stage, a spatial outlier detection algorithm takes care of the invalid data introduced by malicious users. If all measurements in the cell come from the same user, the previous method will not be able to detect measurements that have been consistently modified by a malicious user. As it can be seen from Figure 4.4(b), the insertion of a spatial outlier by a malicious user could generate wrong conclusions about the levels of CO in the USF campus. The proposed method is based on the observation that spatial outliers can be detected if their values are notably different compared to the values from close-by locations, or nearby users. Spatial outliers are removed in this second stage using spatial data mining techniques and a new hybrid method based on the classification of the data in neighborhoods.

4.2.2 Neighborhood Definition

The input data for a spatial data mining algorithm have two different types of attributes: spatial, i.e., locations, coordinates, and the like, and non-spatial attributes, e.g., the values of the measured variables of interest. In the proposed approach, both attributes are considered to define a neighborhood and calculate the neighborhood function for the outlier detection algorithm. It is important to emphasize that similar approaches found in the literature consider a fixed size of K-closest neighbors for each location. The algorithm presented here is more realistic since it does not assume an a priori known number of neighbors per location.

To classify each location as dense or sparse according to its neighborhood, the neighborhoods must be defined or created first. This technique executes in two steps: First, the Delaunay triangulation is used to cluster the data based on the spatial attributes. Second, Gaussian Mixture Models are applied to cluster the data based on the non-spatial attributes of the dataset.

The use of these two clustering methods is very important since the classification of the locations is not straightforward. A location can have really close neighbors, but if the measurements of the neighbors are strongly heterogeneous, i.e., very dissimilar values of
the variables of interest, the use of a median algorithm might not be the right choice to
detect the outliers. This can be the result when using only a distance-based classification,
such as the Delaunay triangulation. To solve this problem, an additional classification stage
that considers the behavior of the measurements in the neighborhood, is required. This is
performed by the Gaussian Mixture Models.

4.2.2.1 Delaunay Triangulation

The Delaunay triangulation (a dual graph of the Voronoi diagram) has been used before
to determine the natural neighbors [32, 33]. A Delaunay triangulation is the set of edges
connecting a set of points such that each location is joined to its nearest/natural neighbors.
Here, the Delaunay triangulation is used as the first estimate to classify each location, which
is based only on the spatial attributes of the dataset.

Using this triangulation, for each location the average distance to all neighbors is calcu-
lated, resulting in $N$ averages, one per location. A location is classified $i$ as having a Sparse
neighborhood if its average distance to its neighbors $\mu_{di}$ complies with $\mu_{di} > (\mu_{avg} + \sigma_{avg})$,
where $\mu_{avg}$ and $\sigma_{avg}$ are the mean and standard deviation of the average distance of all
locations. Figure 4.5(a) shows the initial classification of the locations as Sparse or Dense
after using this technique.

4.2.2.2 Gaussian Mixture Models

The second step clusters the data based on the non-spatial attributes of the dataset. For
this purpose, a Gaussian Mixture Model (GMM) is trained on the variables without using
the location information. This clusters the locations based upon how similar their measure-
ments are. Normally, is is expected nearby locations to have similar measurements. The
Expectation-Maximization (EM) algorithm [100, 101] is utilized with an initial estimation
of the means and covariance matrices using K-means.

As any other clustering technique, the number of components $k_c$ to be used needs to be
defined. Considering this, the distortion metric and the jump criterion presented in [102] are
utilized. Applying this technique on different scenarios always generated consistent results, finding that 5 or 6 components for the GMM creates a good clustering of the locations, in the sense that it is possible to classify the neighbors into homogeneous or heterogeneous. The result of the GMM clustering technique can be seen in Figure 4.5(b).

The homogeneity of each neighbor is determined by a metric based on the mode component of each neighborhood. The mode component (the component with the highest frequency of appearance) of the neighborhood of location $i$ is defined as $m_{Bi}$, the number of neighbors as $n_{ei}$, and the number of neighbors that belong to component $m_{Bi}$ as $n_{mi}$. If $n_{mi} < (0.4 \cdot n_{ei})$, location $i$ is classified as Sparse. The previous criterion establishes that at least 40% of the locations must belong to the mode component to classify a location as Dense. After experimenting, it was found that using a threshold greater than 0.5 (the existence of a majority component) is a very strong condition and most locations would end up being classified as Sparse. After applying this additional classification criterion, those additional Sparse locations are included to the previous classification shown in Figure 4.5(a). The final classification (Figure 4.5(c)) clearly shows how after applying this additional density metric, more locations are classified as Sparse. Both in Figure 4.5(a) and 4.5(c), Dense locations are represented as blue triangles and Sparse locations as red dots.

Figure 4.5: The two steps of the Hybrid approach.
4.2.2.3 Selection of the Neighbors for Each Location

Up to this point, each location has been classified as Dense or Sparse. Now the neighbors for each location need to be selected to calculate the neighborhood function. In the case of Sparse locations, local kriging is utilized to calculate the neighborhood function. For this, the K-closest neighbors to each Sparse location are considered, where \( K = \frac{N}{10} \) has been found as a minimum value to create a good regression using a spherical model for the semi-variogram. The interested reader will find more details about the kriging technique in Chapter 5.

In the case of Dense locations, a location \( j \) is a neighbor of location \( i \) if the following two conditions hold:

1. \( d_{ji} < (\mu_{avg} - \sigma_{avg}) \), where \( d_{ji} \) is the distance between \( j \) and \( i \).
2. \( \gamma(j, m_{Bi}) > 1 \times 10^{-3} \), where \( \gamma(j, m_{Bi}) \) is the probability that the location \( j \) belongs to component \( m_{Bi} \).

Even though the value \( 1 \times 10^{-3} \) seems small, it was found that it is a good threshold to consider nearby locations having “similar” measurements but that do not belong to the same component. Figure 4.6 presents the neighbors for each location, Sparse and Dense.

Figure 4.6: The neighborhood for the Sparse (left) and Dense (right) locations.

Even though the value \( 1 \times 10^{-3} \) seems small, it was found that it is a good threshold to consider nearby locations having “similar” measurements but that do not belong to the same component. Figure 4.6 presents the neighbors for each location, Sparse and Dense. It
can be seen how the Dense neighborhoods cluster following the information from the GMM components (see Figure 4.5(b)).

The final purpose of this two-stage procedure is to have a classification that is sensitive to the level of conspiracy in the system. Figure 4.7 shows the ratio between Dense and Sparse locations versus the conspiracy percentage. When there is no conspiracy, the ratio depends only on the distance, and ultimately, on the Delaunay triangulation. As expected, when the conspiracy percentage is incremented, the number of Sparse locations also increments since the heterogeneity of the locations in the same neighborhood also increases. This special behavior is what allows this algorithm to adapt to an increasing number of malicious users.

![Graph showing the ratio between dense and sparse locations versus the conspiracy percentage.](image)

Figure 4.7: The ratio between dense and sparse locations versus the conspiracy percentage. The conspiracy level is fixed to 0.4.

All these previous parameters, such as the definition of 40% as the threshold to classify a location as Dense instead of the majority criterion, have been selected after a training stage on different datasets. In the same way, filtering data outside one or two standard deviations, is a known technique to remove outliers and non-significant data.
4.2.3 Spatial Outlier Detection for Multiple Attributes

After the neighborhood for each location is defined, a spatial outlier removal algorithm is needed. There are two alternatives to detect outliers using spatial statistics: a quantitative or a graphical approach [12]. The graphical approach basically highlights the outliers for a later interpretation of the user. Some examples of these methods are variogram clouds and Moran scatterplots. The graphical tests lack a precise criterion to decide on the outliers and it strongly depends on the perception of the analyst, that is why a quantitative approach is considered here.

4.2.3.1 General S-Outlier Detection and Removal

The proposed hybrid algorithm is based upon the work of Lu et al. in [16, 43, 45] about local differences. The basic general algorithm is presented first and the hybrid modification is presented afterwards. Given a set of points \( X = \{x_1, x_2, \ldots, x_N\} \) in a space with dimension \( p \geq 1 \), an attribute function is defined as a mapping of \( X \) to \( \mathbb{R} \) (the set of all real numbers). The attribute function \( f(x_i) \) is the value of that attribute in location \( x_i \). For each point \( x_i \), \( NN_k(x_i) \) is denoted as the set of the k-closest neighbors of \( x_i \), where \( k \) can be location-dependent, but for simplicity in this case, \( k \) is a constant for all locations.

A neighborhood function \( g(x_i) \) is a mapping of \( X \) to \( \mathbb{R} \), such that for each location, it returns a summary statistic of the attribute values in \( NN_k(x_i) \). To detect s-outliers, the attribute function \( f(x_i) \) is compared to the neighborhood function \( g(x_i) \). Such comparison is done through a function \( h \), which can be a ratio \( (f/g) \), a difference \( (f - g) \), etc. Given \( y_i = h(x_i) \), a point \( x_i \) is an s-outlier if \( y_i \) is an extreme value in \( \{y_1, y_2, \ldots, y_N\} \).

This definition can be very general, and there are multiple variations depending on how the neighborhood function \( g(x_i) \) is calculated, or which comparison function \( h \) is used. Three basic algorithm are: the iterative \( r \) (\( g \) is the average of the attribute in \( NN_k(x_i) \), and \( h = f/g \)), the iterative \( z \) (\( g \) is the average, but \( h = f - g \)) and the median algorithm (\( g \) is the median and \( h = f - g \)) [16]. After extensive testing, the median algorithm was found to give the best results among these three. This performance is because the median is a
robust estimator of the “center” of a sample. Algorithm 1 presents a general pseudo-code of the median algorithm.

**Algorithm 1**: The Median Algorithm.

**Input**: $X, f$

**Output**: Spatial outlier-free data

**foreach** spatial point $x_i$ **do**

- Compute the $k$ neighbor set $NN_k(x_i)$, the neighborhood function $g(x_i) = \text{median}$ of the data set $\{f(x) : x \in NN_k(x_i)\}$, and the comparison function $h_i = h(x_i) = f(x_i) - g(x_i)$;

**end**

Let $\mu$ and $\sigma$ denote the mean and standard deviation of the dataset $\{h_1, h_2, \ldots, h_N\}$.

Standardize the dataset and compute the absolute values: $y_i = |\frac{h_i - \mu}{\sigma}|$.

For a given positive integer $m$, let $\{i_1, i_2, \ldots, i_m\}$ be the indexes such that their $y$ values are the $m$ largest;

Locations $\{x_{i_1}, x_{i_2}, \ldots, x_{i_m}\}$ are the $m$ s-outliers of the dataset $X$.

As it can be seen, this approach presents several problems. First, the number of neighbors $k$ is fixed for every location, and the selection of these neighbors is only based on distance, without considering the non-spatial attributes. Second, it is necessary to know beforehand the number $m$ of outliers in the dataset (which is not feasible) and that also fixes the stopping point of the algorithm regardless of the dataset. And third, it does not consider the possibility of a multivariate attribute function. A multivariate extension has been made in [45], but after testing, the results were really poor.

### 4.2.3.2 Hybrid S-Outlier Detection and Removal

In order to detect spatial outliers, a multivariate (multiple non-spatial attributes) quantitative approach is considered here. Given a set of points $X = \{x_1, x_2, \ldots, x_N\}$, a multivariate attribute function is defined as $F = \{F_1, F_2, \ldots, F_6\}$, representing the measurements for all 6 variables in all $N$ locations. A neighborhood function $g_i(j)$ and a comparison function $h_i(j) = F_i(j) - g_i(j)$ are also defined for each attribute $i$ and every location $j$.

Almost all previous work in the area of spatial outlier detection considers only one neighborhood function. However, this approach is different, as the proper estimation is selected based on the classification of the neighborhood. Proper in the sense of finding a
good trade-off between computational requirements and quality of the final fit of the spatial interpolation. The idea is to use a simple local method (iterative r, iterative Z, or median algorithms) [16, 45] when a location has a Dense neighborhood, and use a linear estimation of the local value using the kriging technique, which is considered the best linear unbiased spatial predictor [19], for the Sparse case. This combination improves the accuracy and execution time compared to existing methods, as the most appropriate detection algorithm is applied for each neighborhood, i.e., a simple method in dense neighborhoods and a costly but more accurate method in sparse neighborhoods only.

For the case of locations in dense neighborhoods, after initial tests, the median algorithm was chosen over the iterative r or z algorithms due to its higher detection ratio, simplicity, and lower computational complexity. The reader may find and use similar algorithms and probably obtain similar results; however, the main interest here is to show that our hybrid algorithm improves the final results of a single $g_i(.)$ approach.

For the sparse case, a global estimation using kriging is avoided, but instead local kriging is utilized. Local kriging is a kriging estimation that only uses a subset (nearby locations) of the total number of locations to reduce the computational complexity. For this local estimation, Ordinary kriging is preferred since Simple kriging does not adapt well to local trends [103]. Ordinary kriging uses a local mean $\mu_z$ to re-estimate the mean at each grid node from the data within the search neighborhood. In order to have a multivariate criterion, either PCA or ICA is utilized (depending on the variable of interest) as a previous stage to the interpolation process [24]. The complete details about the use of kriging along with PCA o ICA can be found in Chapter 5.

Algorithm 2 presents the proposed hybrid approach in pseudo-code. As it can be seen, the hybrid approach removes a location and all its attributes if the estimation of at least one of the attributes (either using the median or local kriging) is considered a spatial outlier. The parameter $\theta$ can vary between 2.5 and 3.0, corroborating the results presented in [16].

Some approaches consider data modification instead of removal. In that case, the location is not removed but the attribute function for that location is changed to the value
Algorithm 2: The Hybrid Algorithm.

Input: Aggregated data
Output: Spatial outlier-free data

foreach attribute $F_i$ do
  foreach location $j$ do
    Calculate $g_i(j)$ using the median or the local kriging estimation depending on the neighborhood classification;
    Calculate $h_i(j) = F_i(j) - g_i(j)$;
  end
  $h_n(.) = \text{Normalize}(h_i(.))$;
  find a location $b$ with maximum $h_n(.)$;
  while $h_n(b) > \theta$ do
    Remove location $b$ and all its attributes $F_i(b)$;
    Update $g_i(x)$ and $h_i(x)$ for location $x$ that has $b$ in its neighborhood;
    Find a new location $b$ with maximum $h_n(.)$;
  end
end

predicted by the neighborhood function. After testing, it was found that when having a low number of outliers and a low density of malicious users per neighborhood, this approach works well. On the other hand, when there are a large number of malicious users, the s-outliers affect the estimation process on their neighbors and the final performance is poor. The removal of the locations classified as s-outliers lead to a better quality of the final estimations, and that is the reason why that approach is used here.

4.2.4 Computational Complexity

The hybrid algorithm will be compared against the median algorithm, and a pure local Ordinary kriging approach. To ease the mathematical analysis, $C$ is denoted as the number of neighbors for each location, and the number of spatial outliers as $m_s = \text{cons perc} \times N$. For a dataset of $l$ points, the median requires $O(l)$ time and Ordinary kriging requires $O(l^2)$ [69]. For the general spatial outlier removal algorithm, the strongest computational requirement consists of the initial calculation of the $g_i(.)$ function for all $N$ locations. Using the median algorithm for this stage requires $O(CN)$ time, while the kriging approach takes $O(C^2N)$. Ideally, the algorithm will remove one spatial outlier per cycle, therefore, exe-
cuting additional $m_s$ cycles. When removing one spatial outlier, the algorithm recalculates $g_i(.)$ for $C$ locations, requiring $O(Cm_s)$ for the median and $O(C^2m_s)$ for the kriging case.

The Delaunay triangulation requires $O(N\log N)$ [34], the training of a GMM using the EM algorithm requires $O(kcNd)$ [41], and applying the classification criteria (spatial and non-spatial) requires $O(N)$. Therefore, the complete neighborhood definition stage executes in $O(N\log N)$ time.

As explained before, the hybrid algorithm has the median and the kriging algorithm as lower and upper execution bounds, respectively. Ideally, when the system has no malicious users ($m_s \approx 0$), the hybrid algorithm depends on the initial calculation of $g_i(.)$ for all $N$ using the median ($O(CN)$), but no outliers are removed. Therefore, the total complexity would be $O(N\log N + CN) = O(N\log N)$. On the other hand, when all users are conspiring ($m_s \rightarrow N$), the hybrid algorithm would calculate the $g_i(.)$ for all $N$ using kriging ($O(C^2N)$) and additionally would remove $N$ outliers ($O(C^2N)$). The total complexity would be $O(N\log N + C^2N + C^2N) = O(C^2N)$. In practice, the hybrid algorithm would be somewhere in between these two computational bounds, as it will be corroborated with the experimental results in the following section.

4.3 Experiments and Analysis

In this section, the experiments will compare the performance of three algorithms for spatial outlier detection: the median algorithm alone (refer to Algorithm 1), the local kriging algorithm alone, and the proposed hybrid algorithm (refer to Algorithm 2). For the local kriging algorithm, a structure similar to Algorithm 1 is utilized, but the local kriging estimation is calculated for the $g(.)$ neighborhood function, regardless of the classification of the neighborhood, in a similar way as presented in [16, 43].

Initially, the results for the malicious random attacks are presented. After this, the execution time of these three algorithms is measured, and the results are compared. Finally, the case for cooperative attacks is presented and analyzed.
4.3.1 Random Attacks

The evaluation consists of two different scenarios. The first scenario fixes the conspiracy level and increases the conspiracy percentage. The second scenario uses a fixed conspiracy percentage and increases the conspiracy level. In both cases the execution time was measured to show that the hybrid algorithm is faster than a pure kriging approach. Table 4.2 shows the parameters used in the experiments.

Table 4.2: Parameters used in the experiments.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of Locations (N)</td>
<td>100</td>
</tr>
<tr>
<td>Conspiracy Level</td>
<td>[0.1 : 0.70]</td>
</tr>
<tr>
<td>Conspiracy Percentage</td>
<td>[0.0 : 0.25]</td>
</tr>
<tr>
<td>Local Outliers</td>
<td>10</td>
</tr>
<tr>
<td>Repetitions</td>
<td>4</td>
</tr>
</tbody>
</table>

For each instance of the experiment, the coefficient of determination is calculated, and since every experiment is repeated 4 times, the average of $R^2$ is reported as the final performance metric. As explained before, the detection ratio for the local outliers is around 90%, and no more results are presented here since the final goal is the quality of the final interpolation, measured through $R^2$. Due to formatting restrictions, the results for four variables (temperature, relative humidity, carbon monoxide (CO) and carbon dioxide (CO$_2$)) are reported, but the performance of the algorithms is similar for the remaining variables (combustible gases and air quality).

4.3.1.1 Conspiracy Level

For this first experiment, the conspiracy level is varied while maintaining a fixed value of the conspiracy percentage (0.15). The results of these experiments are shown in the left part of Figure 4.8.

In general, a negative value of $R^2$ represents a model-fitting procedure for a completely different data. One of the most notable results is the behavior of the median algorithm for this conspiracy percentage. Regardless of the conspiracy level, the median algorithm
Figure 4.8: $R^2$ versus the conspiracy level (left) and the conspiracy percentage (right) for all four variables and the three algorithms.
always produces a negative coefficient of determination, which implies a poor result for the final spatial interpolation process. The only exception is in the case of the carbon monoxide (CO), which presents similar results to local kriging and the hybrid algorithm. This is because CO presents really low values across the map and the conspiracy of the user does not affect this variable much.

On the other hand, the local kriging and the hybrid algorithms generate similar and almost constant performance for the complete range of the conspiracy level. This clearly proves how robust and resilient the local kriging algorithm is, but also that the new hybrid approach matches this performance. The reader must remember that kriging is, statistically speaking, the best linear unbiased spatial predictor [19]. Achieving a performance close to kriging is optimal, and above this value could imply overfitting of the data. Nevertheless, as it will be shown later, the hybrid algorithm not only matches this accuracy, but also outperforms the local kriging approach in the execution time.

4.3.1.2 Conspiracy Percentage

In this set of experiments, the conspiracy level is fixed at 0.5 and the conspiracy percentage is varied. The results of these experiments are shown in the right part of Figure 4.8. The median algorithm achieves a poor performance when the conspiracy percentage is greater than 0.10 (1 out of 10 users is modifying the measurements). Even though its performance is really good before this point, this condition should not be assumed as the worst case scenario for a real PS system.

Nevertheless, the local kriging and hybrid algorithms again outperform the median algorithm. As expected, the accuracy of the final spatial interpolation suffers with the number of malicious users. However, both algorithms present a good coefficient of determination up to a conspiracy level of 0.22, a notable result compared to the case for the median algorithm. As in the previous experiment, the hybrid algorithm has a similar performance compared to the local kriging, but with a faster execution time.
4.3.2 Execution Time

Finally, it is important to compare the execution time of the three algorithms. Figure 4.9 plots the different execution times for both experiments: on the left, the conspiracy level is varied while the conspiracy percentage is fixed at 0.20, and on the right, the conspiracy percentage is varied while the conspiracy level is fixed at 0.40. As it can be seen from the figure, the execution times are roughly constant for all instances of the experiments. As expected, the median algorithm is the fastest of all three because of the simplicity in the calculation of the median; however, the limitations of its performance have been presented already.

![Figure 4.9: The execution time of the three algorithms for data verification.](image)

On the other hand, the figure shows how the hybrid algorithm executes in less than half the time of the local kriging, but still matching its performance. Under the simulated conditions and after extensive experiments, it was found that the neighborhood generation is quite fast, and the bulk of the computation comes from the spatial outlier detection and removal. These results corroborate the theoretical analysis of the execution time of these three algorithms.
4.3.3 Cooperative Attacks

This section evaluates the proposed algorithms in the case of cooperative attacks. Two different conditions are considered: one where all of the users surrounding the center of conspiracy cooperate, and another where only some of these users cooperate.

Figure 4.10: The $R^2$ for the cooperative attacks where only some of the users surrounding the center cooperate.
Two centers of conspiracy are considered and 5 surrounding users cooperate, as shown in Figure 4.2. The experiments consider values of the conspiracy level of the center between 0.1 and 0.8. For all three algorithms local kriging (left), Hybrid (middle) and median (right), the verified case is compared with the non-verified result.

Figure 4.10 presents the coefficient of determination corresponding to the case where only some of the users surrounding the centers of conspiracy cooperate. As expected, the hybrid algorithm achieves a better performance compared with the median case and really close to the optimal estimator (local kriging). Also, the median algorithm decreases its accuracy as the conspiracy level increases, something that does not happen to the hybrid and local kriging cases, as they exhibit an almost stable performance. Although not shown here, the quality of the spatial estimation for the rest of the variables without data verification is also very poor for conspiracy levels above 0.1.

Figure 4.11 now presents the case where all users surrounding the centers of conspiracy are cooperating to confuse the verification system even further. For all three algorithms local kriging (left), Hybrid (middle) and median (right), the verified case is compared with the non-verified result. Again, local kriging achieves the best results, followed really close by the hybrid algorithm, while the median approach has a mediocre performance. If these results are compared with the ones shown in Figure 4.10, it is easy to see that this case is more demanding for the data verification system than the case where only some of users conspire. In addition, it can be observed how this cooperative attack affects more as the conspiracy level increases.

Finally, Figure 4.12 shows the effect and importance of the verification process. On the left you can see the normal interpolation when no malicious users are attacking the system; in the middle the resulting spatial estimation after all the surrounding users have cooperatively conspired, without data verification; and on the right the interpolation when the Hybrid algorithm is applied. In this case, there are 5 cooperative users and two centers of conspiracy, each with a conspiracy level of 40%. For the non-verified case, it is clear how the spatial estimation is very poor and the visualization can be misleading. On the other
Figure 4.11: The $R^2$ for the cooperative attacks where all of the users surrounding the center cooperate.

On the other hand, the quality of the spatial estimation after the verification process is quite good and reconstructs the original variable in such a way that it can extract the details and most of the meaningful information.
4.4 In Conclusion: Data Verification

The main conclusions that can be drawn from this chapter about data verification are:

- The adversary model that as been designed and implemented, successfully simulates complex scenarios in which the users attack the system in a random and a cooperative way, also considering the low accuracy and the high failure rate of the sensors.
• The neighborhood-aware mechanism properly classifies each location into Dense and Sparse, based on the spatial and non-spatial attributes. This was achieved through the use of the Delaunay triangulation (applied to the spatial attributes) and the Gaussian Mixture Models (applied to the non-spatial attributes).

• By using this classification it is possible to apply computational expensive estimation techniques only in the cases where is necessary, while still matching the performance of kriging (the best spatial estimator) under different test scenarios. The experimental results show that the proposed hybrid algorithm executes in less than half of the time when compared to the kriging technique.

• It has been shown how the data verification module is fundamental in the appropriate analysis of the data, since it removes invalid measurements that can affect the conclusions that are drawn from the complete spatial interpolation.
CHAPTER 5

DATA VISUALIZATION

This chapter studies the problem of data interpolation in Participatory Sensing (PS) systems using the data collected by P-Sense as a validation example. While traditional environmental monitoring systems consist of very few static measuring stations, PS systems rely on the participation of many mobile stations. As a result, the structure of the data provided by each system is different and instead of a multivariate time series with a few gaps in the same space, it consists of a multivariate time-space series with many gaps in time and space. First, two data interpolation techniques, Markov Random Fields and kriging, are analyzed. After showing the trade-offs and superiority of kriging, this technique is used to perform a one-variable data interpolation. The problems of cokriging for multivariate interpolation are introduced, and then, a proposed new method based on Principal Component Analysis and Independent Component Analysis is presented. Finally, a new method to interpolate data in time and space is proposed, which is more appropriate for PS systems. The results indicate that the accuracy of the estimates improves with the amount of data, i.e., one variable, multiple variables, and space and time data. Also, the results clearly show the advantage of a PS system compared with a traditional measuring system in terms of the precision and granularity of the information provided to the users.

The chapter is organized as follows. Section 5.1 compares Markov Random Fields versus kriging, two important techniques for spatial interpolation. Section 5.2 shows the use of kriging to interpolate one variable in space. The work on multivariate spatial interpolation and the corresponding results are presented in Section 5.3, and the proposed interpolation method for space and time is presented in Section 5.4.
5.1 MRF vs Kriging for PS

First, a comparative analysis between two techniques for spatial interpolation (MRF and kriging) is presented. After introducing the basic theory of both techniques, a PS scenario is simulated to compare the performance of these approaches.

5.1.1 Markov Random Fields and Gibbs Sampling

The theory of Markov Random Fields (MRF) applied to image restoration is used here for spatial interpolation. Every location in the map is considered equivalent to a pixel in the image, and the idea is to estimate and fill the missing locations in the map, or the noisy pixels in the image.

In [23], an analogy between images and statistical mechanics is made. The pixel gray levels and the orientation of the edges are viewed as states of atoms or molecules in a lattice-like physical system. The assignment of an energy function in the physical system determines the Gibbs distribution [104]. The algorithm for image restoration adopts a Bayesian approach and introduces a hierarchical stochastic model for the original image based on the Gibbs distribution, leading to a procedure that allows the computation of the maximum a posteriori (MAP) estimate. The restoration algorithm also uses stochastic relaxation and annealing scheduling. The stochastic relaxation algorithm can be described as follows:

1. A local change is made in the image based upon current values of pixels in the immediate neighborhood. This change is generated by sampling from a local conditional probability distribution.

2. The local conditional distributions are dependent on a global control parameter $T$, known as temperature. At low temperatures, the local conditional distributions concentrate on states that increase the objective function, whereas at high temperatures, the distribution is essentially uniform.
3. The algorithm avoids local maximum by using the “annealing schedule”. It begins by using high temperatures where many of the stochastic changes will actually decrease the objective function, and then the temperature is gradually decreased.

The energy function is relaxed in the local environment of the variable, conformed by the cliques of the MRF, or fully connected subsets of variables. The Markovian assumption makes the value of the variables to depend only on the vicinity instead on the values of all pixels.

Since the conditional distribution of every location is unknown, $N$ replicated images (maps) with noisy and missing pixels are generated. At each iteration, the energy of each pixel, which depends on its value and its neighborhood, is compared with the energy of replacing that pixel with a new pixel drawn from one of these generated images. The new value for the pixel is randomly assigned to the current image based on an uniform distribution and the energy exponential ratio. Finally, the “annealing schedule” updates the current temperature on each iteration using:

$$T = T_0 / \log(1 + K)$$

where $T_0$ is the initial temperature, and $K$ is the current iteration.

The proposed model for the MRF is an 8-pixel/location neighborhood (see Figure 5.1), as follows: Horizontal cliques $= \{(S, 4), (S, 5)\}$, vertical cliques $= \{(S, 2), (S, 7)\}$, and diagonal cliques $= \{(S, 1), (S, 3), (S, 6), (S, 8)\}$. For the energy function, 2-pixel cliques are utilized in three different configurations: horizontal, vertical, and diagonal. The energy function (based on the Ising model [105]) is the following:

$$U(s) = \alpha V(s) + \beta_h \sum_{s_H} |V(s) - V(s')| + \beta_v \sum_{s_V} |V(s) - V(s')| + \beta_d \sum_{s_D} |V(s) - V(s')|$$

where $V(s)$ is the value for the pixel for location $s$, $\alpha$ measures the external field, and $(\beta_h, \beta_v, \beta_d)$ measure the bonding strengths for horizontal, vertical, and diagonal cliques, and
Figure 5.1: Cliques definition.

\((S_H, S_V, S_D)\) are the sets of horizontal, vertical and diagonal cliques, respectively. The definition of these parameters depends on the characteristics of the variable to be estimated (or the image to be restored). Since the idea here is to prove the concept of MRF versus kriging interpolation, the parameters of the energy function \((\alpha \text{ and } \beta_{h,v,d})\) were optimized for the test map.

5.1.2 The Kriging Technique

Kriging is one of the most used techniques in geostatistics (a branch of statistics that focuses on spatio-temporal datasets). Kriging is a BLUE (Best Linear Unbiased Estimator) interpolator, i.e., it is a linear estimator that matches the correct expected value of the population (unbiased), also minimizing the variance of the observations (Best). The basic theory on kriging to spatially interpolate one variable will be presented. Sections 5.2 and 5.3 present the use of kriging along with some modifications to spatially interpolate the dataset of air pollution measurements collected by P-Sense. For the interested reader, a detailed explanation of the kriging technique can be found in [22, 53]. Here, a brief introduction to the theory of this technique is presented based on these references.
Let $Z(\vec{x},\xi)$ be a random function representing any of the variables of interest (temperature, RH, CO, etc.). In this notation $\vec{x}$ implies a point in the space (for this case a 2D or 3D space), while $\xi$ denotes the state variable in the space of realizations. Therefore, given a set of sample values $Z(\vec{x}_i,\xi_1)$, with $i = 1, 2, \ldots, N$, the main idea is to reconstruct all possible locations, i.e., $Z(\vec{x},\xi_1)$ a realization of $Z(\vec{x},\xi)$.

One first assumption about the random function is that they are second order stationary, implying that the two first statistical moments (mean and covariance) are translation invariant:

1. Constant mean: $E[Z(\vec{x},\xi)] = m$

2. The autocovariance is a function of the distance between the reference points $\vec{x}_1$ and $\vec{x}_2$: $\text{cov}[x_1, x_2] = E[(Z(\vec{x}_1,\xi) - m)(Z(\vec{x}_2,\xi) - m)] = C(\vec{h})$

The normal definition of variance (also called dispersion variance) is achieved when $\vec{h} = 0$: $C(0) = \text{var}[Z] = \sigma^2$. For simplicity, the variable $\xi$ and the vector notation for $x$ and $h$, are dropped. A new variable $Y(x)$ can now be defined having zero mean:

\begin{align*}
Y(x) &= Z(x) - m \\
E[Y(x)] &= 0 \\
Y_i &= Y(x_i)
\end{align*}

(5.3) \hspace{1cm} (5.4) \hspace{1cm} (5.5)

It is desired now to find a linear estimator $Y_0^*$ of $Y_0$ at point $x_0$ using the observed values. The general form of this estimator is given by:

\[ Y_0^* = \sum_{i=1}^{n} \lambda_i Y_i \]

(5.6)

The weights $\lambda_i$ are calculated by imposing the restriction that the statistical error $\epsilon(x_0) = (Y_0 - Y_0^*)$ has zero expected value and minimum variance. After using these
restrictions, a linear system of equations can be found:

\[
C = \lambda b
\]  

(5.7)

where the matrix \( C \) is defined as:

\[
C = \begin{pmatrix}
C(0) & C(x_1 - x_2) & \ldots & C(x_1 - x_n) \\
& \ddots & \ddots & \ddots \\
& & \ddots & \ddots \\
C(x_n - x_1) & C(x_n - x_0)
\end{pmatrix}
\]  

(5.8)

and the vector \( b \) is given by:

\[
b = \begin{pmatrix}
C(x_1 - x_0) \\
& \ddots \\
& & \ddots \\
C(x_n - x_0)
\end{pmatrix}
\]  

(5.9)

Once the weights are calculated, the estimation of \( Y_0 \) at point \( x_0 \) can be found. Nevertheless, the second order stationary property not always holds and in that case is necessary to use the intrinsic hypothesis in which it is assumed that the first order increments:

\[
\delta = Y(x + h) - Y(x)
\]  

(5.10)

are second order random functions with:

\[
E[Y(x + h) - Y(x)] = m(h) = 0
\]  

(5.11)

and

\[
\text{var}[Y(x + h) - Y(x)] = 2\gamma(h)
\]  

(5.12)
The function $\gamma(h)$ is the variogram, and $2\gamma(h)$ is called the semi-variogram. The variogram is related to the covariance function in the following way:

$$\gamma(h) = C(0) - C(h)$$ (5.13)

It is clear that by calculating the variogram, the covariance can also be calculated, and therefore, the weights $\lambda$. In order to calculate a robust experimental semivariogram $2\bar{\gamma}(h)$, the work in [106] defines:

$$2\bar{\gamma}(h) = \left( \frac{1}{N} \sum_{i=1}^{N} |Z(x_i) - Z(x_j)|^{1/2} \right)^4 / (0.457 + 0.494/N)$$ (5.14)

Given this experimental semivariogram, these values need to be fitted to a theoretical model, being the spherical, Gaussian, and exponential the most commonly used ones. For the dataset that was used, the one that worked better is the spherical model:

$$\gamma(h) = (s - n) \left( \frac{3h}{2r} - \frac{h^3}{2r^3} \right) 1_{(0,r)}(h) + 1_{(r,\infty)}(h) + n 1_{(0,\infty)}(h)$$ (5.15)

where $n$ is the nugget or the height of the jump of the semivariogram at the discontinuity of the origin, $s$ is the sill or the limit of the variogram tending to infinity lag distances ($h$), and $r$ is the range or the distance in which the difference of the variogram from the sill becomes negligible. The $1_A(h)$ function is 1 if $h \in A$, and 0 otherwise. Nevertheless, the use of any of the other two models will not change the methods presented here, but just the fitting process of the semivariogram.

For the semivariograms, it is assumed that the variability of the random function does not depend on the direction in the space (isotropic semivariograms). In the future, for the case of air pollution, if wind speed and direction data are available, anisotropic semivariograms could be considered. As mentioned before, the isotropic spherical model for the semivariogram fits well the calculated experimental semivariograms. Figure 5.2 shows the semivariogram for the simulated PS scenario (presented next), which includes the fitted
spherical model. Even though the model does not seem to do a perfect job due to the dispersion of the points, this fit is a normal result when using noisy data.

Figure 5.2: Experimental semivariogram and the fitted curve of an isotropic spherical model. Range = 13.628, Sill = 4866, Nugget = 0.0.

A software application has been implemented to automate the entire interpolation process. The application generates the experimental semivariogram, fits the values to the spherical model, and makes use of mGstat [107] to calculate the interpolated data by using three different types of kriging, simple, ordinary, and trend. The use of kriging using trends leads to the best results.

5.1.3 Modeling a PS Scenario for Data Interpolation

A squared \((L \times L)\) map using a mixture of 2-dimensional gaussians in the space was generated, where \(L\) is the size of the side of the map (or number of pixels per side). For each gaussian, the mean \((\mu)\), the covariance \((\Sigma)\), and a weight \((w)\) are defined. This map is scaled in such a way that the final map has a maximum of 255 and a minimum of 0 (following the normal convention for gray values in an image). As explained before, this original map is replicated \(N\) times, one per iteration, representing different snapshots of the system. All maps have the same basic configuration for the mixture of gaussians \((\mu\) and
In order to model the characteristics of a real PS application, two sources of disturbance were added: Gaussian noise per location and missing locations per snapshot. The Gaussian noise, with zero mean and a standard deviation $\sigma$ that is varied during the experiments, models the lack of accuracy (noise) of the sensors in the cellular phones. On the other hand, due to the mobility of the users, only some locations in the map will be available at every time. To model this, randomly selected locations were removed on each iteration.

A ratio ($n_r = \frac{l_{un}}{l_{kn}}$, from 0 to 1) between the number of missing ($l_{un}$) and known ($l_{kn}$) locations is defined. In this case, a ratio $n_r = 0$ represents the case where no locations are removed, but a ratio $n_r = 1$ means that all ($L \times L$) locations are deleted. The locations to be removed are randomly chosen (following an uniform distribution) from the set of known locations, until the desired ratio is met.

When using kriging, only the known data are considered, but for MRF, all locations are required to be defined, since they are supposed to be pixels from an image. For the locations that are selected to be removed, the known value is replaced by a random value following an uniform distribution between 0 and 255. If the removed data were replaced by a fixed value, say the mean of the map, when the ratio $n_r$ is low the final estimates by using the MRF-Gibbs algorithm tend to this fixed value.

5.1.4 Results and Comparison

In order to measure the quality of the estimations, the coefficient of determination ($R^2$) is utilized. By calculating the coefficient of determination, the performance of both techniques (MRF and kriging) can be compared under different noise levels and ratios $n_r$.

In this section the results and corresponding analysis for both approaches, MRF and kriging, are presented. The variables to be estimated are assumed to be continuous and smooth in the space, i.e. temperature does not change abruptly between two close-by loca-
tions in the map. Using this, a low-pass smoothing filter is applied to the final estimations in both cases, MRF and kriging.

First, a qualitative analysis is given. In Figure 5.3 you can see, from left to right, the original map, a modified noisy map, and the restored version after running the MRF-Gibbs algorithm. The generated map will be the base of all the experiments. The side of the grid is \( L = 35 \) pixels and 4 Gaussians are used to generate the map, and the noise has a standard deviation \( \sigma = 6 \), the missing ratio is \( n_r = 0.6 \). The MRF algorithm is tuned for this specific map, and the optimal initial temperature for all experiments presented here is \( T_0 = 3.5 \) (the literature [23] reports values in the range of \( T_0 = [2 : 4] \)). After initial tests, it was found that 50 is an appropriate number of iterations for the MRF case.

![Figure 5.3](image)

Figure 5.3: The map (top row) and surface (bottom row) representation of the synthetic data created for the experiments.

Figure 5.3 gives an idea of the performance of MRF for estimating the missing and noisy locations. Figure 5.4 now presents a comparative, but still qualitative, result between MRF
and kriging. While having a fixed noise standard deviation ($\sigma = 6$), the missing ratio is increased in order to analyze the quality of the estimations. For low missing ratios, such as $n_r = 0.5$ (left column), both approaches have a good performance, but when the ratio is increased to $n_r = 0.8$ (middle column) and $n_r = 0.95$ (right column), it is clear that kriging generates a smoother and more accurate estimation of the original map, compared to the MRF case.

Figure 5.4: A comparison between the results for the spatial interpolation using kriging (top row) and MRF (bottom row).

Nevertheless, by only looking at the surface map, it is not possible to quantify and compare the performance of MRF and kriging. It is necessary to quantify the accuracy of the estimations while varying the noise $\sigma$ and the missing ratio $n_r$. For this purpose, the coefficient of determination $R^2$ is used. It is worth mentioning that the additional stage of filtering (low-pass filter) always improved the $R^2$ since it removes high frequency components not present in the original map. This smoothing filter uses a $5 \times 5$ averaging window.

Experiments were run with different values for the noise $\sigma$ and varying the missing ratio $n_r$ between 0 and 1. Figure 5.5 shows the performance, as given by $R^2$, for the MRF case.
Three different values for the standard deviation of the noise are considered: 6, 15 and 30. On the left, the range for the missing ratio is [0.0 : 1.0], and on the right, the range is only [0.8 : 1.0] in order to have a closer look at the conditions for a typical PS application. As it can be expected, the higher the noise and the higher the missing ratio on the signal, the lower the accuracy of the estimates. For a missing ratio \( n_r < 0.7 \), the performance of MRF case is still acceptable \( (R^2 > 0.6) \).

![Figure 5.5: R^2 for MRF under different conditions of noise and missing ratios.](image)

The results of the performance of the kriging technique in the spatial interpolation process are shown in Figure 5.6. Again, three different values for the standard deviation of the noise are considered: 6, 15 and 30. On the left, the range for the missing ratio is [0.0 : 1.0], and on the right, the range is only [0.8 : 1.0]. As it is shown in the figure, kriging has a great performance for almost all the different missing ratios. Although the performance also decreases with the increase of the noise \( \sigma \), the performance is much better than the one observed in the MRF case.

Nevertheless, the PS case needs to be considered now. For a typical PS application, the number of known locations, compared to the number of points in the space to be estimated, is really low. For the scenario proposed here, that would imply a missing ratio always higher...
than 0.9. In Section 3.1, a real PS application for air pollution monitoring is presented. For that specific dataset, the missing ratio is always greater than 0.9, and this is a typical value for real PS applications. For this case, a closer look to the condition $n_r = [0.8 : 1.0]$ is considered.

For the condition where $n_r = [0.8 : 1.0]$, Figures 5.5 and 5.6 present a second graph on the right. There is a big difference between MRF and kriging in this range of $n_r$. Kriging clearly outperforms MRF for missing ratios greater than 0.8, and the strong fall of kriging only occurs for values greater than 0.975, and even for that value, the coefficient of determination is around 0.8.

Several conclusions can be drawn from this quantitative analysis. Based on the coefficient of determination, even though MRF has a good performance for high densities (low missing ratios), for the case of low densities (high missing ratios) kriging clearly outperforms MRF. However, this conclusion does not consider the computational requirements of the techniques. Because of the calculation of the semi-variogram and the estimation of each location based on all other known locations, kriging is computationally more expensive than MRF, for which the estimation happens in a small neighborhood. Therefore, it can
be concluded that MRF is a good choice for spatial interpolation when a high density of measurements is available, but for small densities, kriging is clearly the right technique.

One of the possible solutions for the poor performance of MRF for large values of \( n_r \) is to increase the size of the neighborhood (see Figure 5.1). By doing this, the estimation of each pixel/location will now depend on farther locations, reducing the effect of gaps in the space. The downside of this approach is that the computational requirements will increase as well. For example, if the size of the neighborhood is increased continuously, at the end, the energy function will include all possible known locations in the map, arriving to a case similar to kriging. Kriging determines the weights by using the semi-variogram, and the MRF with a neighborhood as big as the map/image should use a distance criterion to define the parameters of the energy function.

As explained before, Section 3.1 presents a PS application for air pollution, and due to the density of the measurements, kriging will be use for the spatial interpolation process. Nonetheless, the work will be focused on the extreme missing ratio conditions, and some modifications for kriging using multiple variables and time are presented and utilized.

5.2 One Variable Spatial Interpolation for PS

From this point, the dataset will be the one generated after the collection of real pollution data with the use of P-Sense. This creates a complex and interesting test scenario for the presented algorithms. To measure the quality of the interpolated data, the coefficient of determination \( (R^2) \) is calculated. This calculation is computationally expensive since it has to interpolate every known location in the map without using the information for that location. Once the estimation is finished, the coefficient of determination using the known values is calculated. Figure 5.7 shows an example of the mesh and contour maps of two interpolated variables, temperature and combustible gases.

As it can be seen from the figure, the interpolated data follows a smooth function in space. Nonetheless, there is a really interesting result: the map for combustible gases shows an almost constant behavior across most of the campus, but there is a lump in one of
the corners. The location on the campus where this increase on the combustible gases is occurring is filled with big buildings that require powerful air conditioning plants and parking facilities. These conditions might be the cause of this unexpected behavior, and thanks to P-Sense it is now possible to alert the corresponding agencies to take care of these situations.

This previous result is one of the most important contributions of this dissertation. By using traditional air pollution stations, it would be impossible to generate spatial estimations with this resolution and accuracy because this behavior occurs on small localities. For a governmental agency it would be crucial to know these details and behaviors of the pollution variables to decide where to build new schools, hospitals, or similar important buildings. For an individual, this information could be very useful to avoid allergic or respiratory situations. For instance, a PS user could set threshold values in the mobile application to specify his or her tolerance to a particular variable of interest and receive automatic alert messages when the thresholds are exceeded. This feedback mechanism can be used as an incentive mechanism to promote user participation.
5.3 Multivariate Spatial Interpolation for PS

An extension of the univariate case is necessary to consider the correlation information of multiple variables, as the use of correlation information should improve the accuracy of the prediction process. For such a scenario, the cokriging technique considers the variability in space of multiple variables to interpolate every variable.

Nonetheless, one of the problems of using cokriging is that it not only needs to calculate a semivariogram for each variable (which is the same case for kriging), but also the cross-semivariograms (a function that represents the cross-spatial dependence between two different variables) for each pair of variables. In this case, the number of regressions for the semivariogram model increases exponentially with the number of variables, the complexity of the system increases, and the estimates depend on many different fitted parameters, compromising the final accuracy. As a result, a modification of kriging is proposed instead of the cokriging technique.

The proposed modification works as follows. It first pre-processes the PS pollution measurements by using principal component analysis or independent component analysis. After that, it uses the regular one-variable kriging technique for each of the components, and then the original variables are reconstructed. It will be shown, that this simple procedure improves the accuracy of the system by using the correlation between the variables.

5.3.1 Kriging and PCA

The objectives of the principal components analysis (PCA) are data reduction and interpretation [17]. The selected $k$ principal components can replace the original $p$ variables, reducing the dimensionality of the input data. Furthermore, the principal components are uncorrelated linear combinations, which help reducing the redundancy of the input data.

PCA is based on the analysis of the covariance (or correlation in other cases) matrix $\Sigma$. Let $\Sigma$ be the covariance matrix associated with the random vector $\mathbf{X}' = [X_1, X_2, \ldots, X_p]$. Let $\Sigma$ have the eigenvalues-eigenvector pairs $(\lambda_1, \vec{e}_1), (\lambda_2, \vec{e}_2), \ldots, (\lambda_p, \vec{e}_p)$, where $\lambda_1 >, \lambda_2 > \ldots > \lambda_p$. After having the sorted pairs of eigenvalues and eigenvectors, the $i_{th}$ principal
component is given by:

\[ Y_i = \tilde{e}_i^T X = \tilde{e}_{i1} X_1 + \tilde{e}_{i2} X_2 + \cdots + \tilde{e}_{ip} X_p, \text{ for } i = 1, 2, \ldots, p \]  \hspace{1cm} (5.16)

If more principal components are included, more of the variability of the system will be included, but it will not reduce much its dimensionality. For this specific application, dimensionality reduction is not the objective. The fact that all principal components are uncorrelated linear combinations is the key fact of this approach.

Because of the additional complexity of having to model the cross-semivariograms, it is suggested to first apply PCA to the data. The use of the principal components instead of the original variables provides uncorrelated new variables for the kriging process. Since the variables are uncorrelated, the use of cokriging brings no improvement compared to simple kriging.

The idea now is to use the principal components for the regular one-variable kriging. To reconstruct the interpolated original variables, the inverse transformation (using the eigenvectors from the correlation matrix of the original data) is applied to the kriged values. In this approach, if there are \( K \) original variables, all \( K \) principal components will be used, therefore, there will be no dimensionality reduction. A pictorial representation of this method versus cokriging is shown in Figure 5.8, and the red arrow shows the flow of the processing for the spatial interpolations.

![Figure 5.8: The proposed kriging plus PCA/ICA method, versus multivariate cokriging.](image-url)
By using kriging and PCA the complexity of the spatial-interpolation process is reduced compared to the cokriging case. Nevertheless, by using cokriging a cokriging variance for each variable can be generated, which is not the case for the kriged values on the principal components. The generated variance for that case will correspond to the variance of the principal components, and not the original variables under analysis. For the interested reader, an illustrative example on the use Principal Component Analysis (PCA) to analyze air quality variables is included in [47].

For the kriging and PCA combination, the accuracy of the system improves when compared with the case of one-variable kriging. Figure 5.9 shows the results of applying kriging on the principal components of the original pollution variables, and including time information, as it will be presented later in Section 5.4. As it can be seen, the smoothness of the map is maintained, and as it is shown in Table 5.1, the quality of the interpolation, measured using the coefficient of determination ($R^2$), is higher than in the previous case when using kriging independently for each variable. The meaning of grid size and delta time in this model, and the corresponding choice for the optimal values, will be presented in Section 5.4.

5.3.2 Kriging and ICA

Independent component analysis (ICA) is a recently developed method in which the goal is to find a linear representation of non-gaussian data so that the components are statistically independent, or as independent as possible [108].

To define the concept of independence, let us first consider two scalar-valued random variables $y_1$ and $y_2$. The variables $y_1$ and $y_2$ are said to be independent if information on the value of $y_1$ does not give any information on the value of $y_2$, and vice versa. This is the most important reason why using ICA along with kriging is a good idea, and a feasible solution versus cokriging.
To define ICA a statistical “latent variables” model can be used. Assume that $n$ linear mixtures $x_1, x_2, \ldots, x_n$ of $n$ independent components $s_1, s_2, \ldots, s_n$, are observed:

$$x_j = a_{j1}s_1 + a_{j2}s_2 + \ldots + a_{jn}s_n \text{ for all } j \quad (5.17)$$

Without loss of generality, it is assumed that both the mixtures and the independent components have zero mean. If this condition does not hold, the means can be subtracted from them, having zero mean mixtures or components. The independent components are latent variables, meaning that they cannot be directly observed. In this case, the mixtures are the measures collected by P-Sense (6 variables in total) and the independent components are unknown variables to be estimated.

It is convenient to have a matrix notation of the ICA model [109, 110]. Denoting $\mathbf{X}$ the random vector whose elements are the mixtures $x_1, x_2, \ldots, x_n$, and likewise $\mathbf{S}$ the random
vector with elements \( s_1, s_2, \ldots, s_n \), the ICA model can be formulated as:

\[
X = AS
\]  

(5.18)

where \( X \) is an observed multidimensional vector, \( S \) is a multidimensional random vector in which its components are assumed mutually independent, and \( A \) is a mixing matrix to be estimated. After estimating \( A \), its inverse can be computed, say \( W \), and the independent components are simply:

\[
S = WX
\]  

(5.19)

An important measure of non-gaussianity is given by negentropy. Negentropy is based on the information theoretic quantity of (differential) entropy. The entropy of a random variable can be interpreted as the degree of information that the observation of the variable gives. The more random, i.e. unpredictable and unstructured the variable is, the larger its entropy. More rigorously, entropy is closely related to the coding length of the random variable, in fact, it can be said that the entropy is the coding length of the random variable. Considering this, negentropy \( J \) is defined as follows:

\[
J(y) = H(y_{\text{gauss}}) - H(y)
\]  

(5.20)

where \( H(X) \) denotes the entropy of a random variable \( X \), \( y_{\text{gauss}} \) is a Gaussian random variable of the same covariance matrix as \( y \). The negentropy is always non-negative, and it is zero if and only if \( y \) has a Gaussian distribution.

Similar to PCA, ICA finds a linear subspace transformation, but with different constraints. The goal of ICA is to minimize the mutual information. Using the concept of differential entropy, the mutual information \( I \) between \( m \) scalar random variables \( y_i \) can be found as follows:

\[
I(y_1, y_2, \ldots, y_m) = \sum_{i=i}^{m} H(y_i) - H(y)
\]  

(5.21)
This previous equation can be interpreted using the definition of code length. The terms $H(y_i)$ represent the lengths of the code for each $y_i$ when they are coded separately, and $H(y)$ is the code length when $y$ is coded as a random vector, in other words, all the components are coded in the same code. Mutual information is a natural measure of the dependence among random variables. Given an invertible linear transformation $S = WX$, an important property of mutual information is:

$$I(s_1, s_2, \ldots, s_m) = \sum_i H(s_i) - H(X) - \log|\det W|$$

$$= C - \sum_i J(s_i)$$

(5.22)

(5.23)

considering that all independent components $s_i$ are uncorrelated. This shows the fundamental relation between negentropy and mutual information.

FastICA [111], a popular ICA algorithm based on a fixed-point iteration scheme, is utilized in this project. Again, after applying ICA to the original variables, the univariate version of kriging can be utilized to interpolate the independent components and then return to the original variables, as shown in 5.8.

Figure 5.9 shows some interpolated maps using kriging along with ICA. A comparison against the other techniques for spatial interpolation (see Table 5.1) shows that ICA improves the accuracy of the system in a similar way that PCA does. Moreover, PCA and ICA complement each other in the sense that the maximum $R^2$ is achieved by using PCA for some variables and using ICA for the other variables. Again, the meaning of grid size and delta time, and the choice for the optimal values, will be presented in Section 5.4.

5.4 Interpolation in Time and Space

Suppose that for a given day, consecutive snapshots of the university campus are available, say at 9AM, 10AM, and 11AM (see Figure 5.10). It is clear that the information contained in the snapshots at 9AM and 11AM can give an idea of what the state of the variables were at 10AM.
So far, the performance of the inference process has been improved by using the correlation information between the given variables, and that is as far as most of the geostatistical methods go. Nevertheless, given the nature of the PS application, the correlation information from consecutive snapshots of the system can be used. Here, a modification of the kriging technique to make use of this available correlation is proposed.

5.4.1 The Grid and Delta Parameters

The environmental conditions, and also the nature of the variables, affect the way how each variable changes in space. Not all variables change in the same manner under the same environmental conditions, therefore some important characteristics of the pollutants can happen on different scales. Because of this, the resolution required to characterize each variable can be different. Here, a broad range of values for the grid size, which defines the resolution of the grid, is considered. For the case in which only spatial information is considered, the application interpolates the variables using each grid size value, and the value for which the highest $R^2$ is achieved determines the optimal size. Due to the different characteristics of the pollutants and the quality and density of the measurements, this optimal parameter is not the same for all variables.

The spatial interpolation using kriging can be seen as the process of finding the unknown values of a variable in a 2D grid with some known locations. If this concept is extended
by adding one dimension to model time, information of consecutive snapshots, as having adjacent layers, can be included. Figure 5.10 gives a pictorial representation of this concept of interpolating the values for 10AM by using the information from the adjacent snapshots at 9AM and 11AM. An additional parameter $delta_{time}$ is used here to model the distance between consecutive grids. A snapshot will affect more if it is closer to the target grid.

Since each variable changes differently in time, i.e., the temperature does not change as fast as the carbon monoxide or the combustible gases, it might be necessary to have a different “distance in time” for each variable. The experiments for this case will again consider a broad range of values for the parameter $delta_{time}$ to model the different dynamics for the variables.

If a model with the dynamics of the air pollutants was available, it could be used to determine the optimal parameters $delta_{time}$ and $grid_{size}$ in a more elegant way. Since this kind of model is not available, the developed application interpolates the variables for all possible combinations of $grid_{size}$ and $delta_{time}$, using only kriging for one variable and using kriging along with PCA or ICA. The application picks the duple ($grid_{size}, delta_{time}$) that achieves the maximum $R^2$. The results show that for different pollutants, the optimal $grid_{size}$ and $delta_{time}$ can be different (see Table 5.1), which is normal due to the differences in the nature of the variables being estimated.

5.4.2 Multivariate Kriging in 3D

Part of the dataset that was collected using P-Sense in the university campus includes sets of three consecutive snapshots. Using this dataset and the modifications presented before, the given pollution variables were interpolated. Figure 5.11 shows the coefficient of determination as a function of the $grid_{size}$ and the $delta_{time}$ parameters for two selected variables. As expected, the behavior of $R^2$ depends on the variable under analysis. In general, really good results can be achieved ($R^2 > 0.8$) for $grid_{size} > 100$ and $delta_{time} > 1x10^4$ for these two specific variables.
Table 5.1 shows the complete comparison of all variables for all the interpolation techniques presented in this dissertation. The table gives the maximum coefficient of determination when using only kriging for one variable and when using kriging along with PCA or ICA. Both conditions, using only spatial information and using spatio-temporal information (consecutive snapshots) are tested.

From Table 5.1 it can be seen that, in general, the use of time information always improves the accuracy of the inference process. As expected, the best performance of the system comes when using the correlation in time along with PCA or ICA. The comparison between ICA and PCA is very interesting, because the combination of both techniques gives a robust inference process with coefficients of determination in the range of [0.77 : 0.98] for almost all variables.
Figure 5.12: Known and predicted values after using kriging interpolation along with PCA and time information.

Table 5.1: Maximum $R^2$ for each variable with the corresponding grid size and delta time parameters.

<table>
<thead>
<tr>
<th>Method</th>
<th>Only Space</th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Krig</td>
<td>$R^2$</td>
<td>Grid</td>
<td>$R^2$</td>
<td>Grid</td>
<td>Delta</td>
</tr>
<tr>
<td>Temp</td>
<td>0.6407</td>
<td>100</td>
<td>0.7294</td>
<td>180</td>
<td>30000</td>
</tr>
<tr>
<td>RH</td>
<td>0.6607</td>
<td>80</td>
<td>0.7078</td>
<td>40</td>
<td>400</td>
</tr>
<tr>
<td>CO</td>
<td>0.6541</td>
<td>160</td>
<td>0.8381</td>
<td>160</td>
<td>1000</td>
</tr>
<tr>
<td>CO2</td>
<td>0.4832</td>
<td>80</td>
<td>0.5561</td>
<td>80</td>
<td>25</td>
</tr>
<tr>
<td>Comb Gas</td>
<td>0.5903</td>
<td>80</td>
<td>0.6947</td>
<td>100</td>
<td>300</td>
</tr>
<tr>
<td>Krig-PCA</td>
<td>$R^2$</td>
<td>Grid</td>
<td>$R^2$</td>
<td>Grid</td>
<td>Delta</td>
</tr>
<tr>
<td>Temp</td>
<td>0.6812</td>
<td>180</td>
<td>0.7708</td>
<td>180</td>
<td>50000</td>
</tr>
<tr>
<td>RH</td>
<td>0.7308</td>
<td>60</td>
<td>0.7978</td>
<td>80</td>
<td>10</td>
</tr>
<tr>
<td>CO</td>
<td>0.8485</td>
<td>140</td>
<td>0.9868</td>
<td>180</td>
<td>500</td>
</tr>
<tr>
<td>CO2</td>
<td>0.5819</td>
<td>160</td>
<td>0.7367</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>Comb Gas</td>
<td>0.7109</td>
<td>80</td>
<td>0.8273</td>
<td>140</td>
<td>500</td>
</tr>
<tr>
<td>Krig-ICA</td>
<td>$R^2$</td>
<td>Grid</td>
<td>$R^2$</td>
<td>Grid</td>
<td>Delta</td>
</tr>
<tr>
<td>Temp</td>
<td>0.6900</td>
<td>120</td>
<td>0.7758</td>
<td>180</td>
<td>500</td>
</tr>
<tr>
<td>RH</td>
<td>0.7580</td>
<td>120</td>
<td>0.7656</td>
<td>100</td>
<td>500</td>
</tr>
<tr>
<td>CO</td>
<td>0.8725</td>
<td>180</td>
<td>0.9867</td>
<td>180</td>
<td>1000</td>
</tr>
<tr>
<td>CO2</td>
<td>0.5716</td>
<td>160</td>
<td>0.7507</td>
<td>160</td>
<td>5</td>
</tr>
<tr>
<td>Comb Gas</td>
<td>0.7063</td>
<td>100</td>
<td>0.8199</td>
<td>140</td>
<td>500</td>
</tr>
</tbody>
</table>
Figure 5.12 plots the known values and the predicted ones. This figure was generated using the optimal values shown in Table 5.1 for PCA and time information. As the figure shows, the predicted values follow very closely the known values, leading to the high coefficients of determination.

As explained before, and as it was proven now, the parameters grid size and delta time are different for each variable and strongly determine the quality of the spatial interpolations. It is clear that one configuration (only kriging, or kriging with PCA, or kriging with ICA) will not work properly on estimating all possible variables under the restrictions of a PS system such as P-Sense, and that is the reason why expert predictors should be used to interpolate the variables.

An expert predictor per variable can be used to consider these different configurations for the grid size and delta time for the kriging technique. Using the optimal values of grid size and delta time, the expert predictors will use these same values for all future estimations in the process of aggregation and 3-dimensional estimation. These optimal values can be found by looking at the maximum $R^2$ for each variable in Table 5.1. The configuration for each expert predictor is summarized in Table 5.2. It is worth noticing that all expert predictors, either using kriging with PCA or ICA, will always consider the spatio-temporal information to interpolate the variables.

Table 5.2: Configuration of the expert predictors for all the variables.

<table>
<thead>
<tr>
<th>Variable</th>
<th>Maximum $R^2$</th>
<th>Grid Size</th>
<th>Delta Time</th>
<th>Method Used</th>
</tr>
</thead>
<tbody>
<tr>
<td>Temp</td>
<td>0.7758</td>
<td>180</td>
<td>500</td>
<td>Krig-ICA</td>
</tr>
<tr>
<td>RH</td>
<td>0.7978</td>
<td>80</td>
<td>10</td>
<td>Krig-PCA</td>
</tr>
<tr>
<td>CO</td>
<td>0.9868</td>
<td>180</td>
<td>500</td>
<td>Krig-PCA</td>
</tr>
<tr>
<td>CO2</td>
<td>0.7507</td>
<td>160</td>
<td>5</td>
<td>Krig-ICA</td>
</tr>
<tr>
<td>Comb Gas</td>
<td>0.8273</td>
<td>140</td>
<td>500</td>
<td>Krig-PCA</td>
</tr>
</tbody>
</table>
5.5 In Conclusion: Data Visualization

The main conclusions that can be drawn from this chapter about data visualization are:

- By using kriging (the best spatial estimator) along with Principal Components Analysis (PCA) or Independent Components Analysis (ICA), a multivariate approach was created in order to improve the quality of the spatial interpolations in space. The experimental results show that by considering the correlation among multiples variables, the quality of the spatial interpolation improves when compared to the univariate case.

- Through the use of distance to model time, the temporal correlation of the variables has been effectively included in the spatial interpolation process. The experimental results also show that by considering not only multiple variables but also the temporal correlations of adjacent snapshots of the system, the quality of the final spatial interpolation improves when compared to the multivariate approach alone.

- The final configuration of the interpolation parameters strongly depends on the variable under analysis. Taking this into consideration, the use of expert predictors is necessary in order to achieve the best quality of the spatial interpolations.
CHAPTER 6

DENSITY MAPS

One key challenge in PS systems is that of the determination of the locations and number of users where to obtain samples from in order to accurately represent the variable of interest with a low number of participants. This chapter presents the use of density maps, based on the current estimations of the variable, to address this challenge. The density maps are then utilized by the incentive mechanism in order to encourage the participation of those users indicated in the map. The results show how the density maps greatly improve the quality of the estimations while maintaining a stable and low total number of users in the system [26].

The chapter is organized as follows. Section 6.1 details the proposed technique, both for the univariate and the multivariate cases, and Section 6.2 presents the experiments and analyzes the results.

6.1 Density Map of Users

In the same way of the previous chapters, the information collected by P-Sense is used here as the dataset to validate the proposed techniques and mechanisms. As an example, Figure 6.1 shows the contour maps of the spatial interpolation for CO and CO$_2$ across the campus. This resulting interpolation is the dataset that will be used in the experiments presented here. The idea is to characterize the system depending on how good these variables can be reconstructed by sampling the maps with a reduced number of users.

Through the generation of a density map (a concept that is first introduced in the PS context), the desired locations where to obtain the next measurements will be defined.
Figure 6.1: Contour maps for all variables (temperature, RH, CO$_2$, CO, air quality, and combustible gasses) in the USF campus.

As explained before, this mechanism provides feedback and affects the process of data collection, both for the incentives strategies and the privacy protection mechanisms. The incentives strategies must encourage the displacement of the users to the locations where is more convenient to sample the variable of interest. Also, the privacy mechanisms should
guarantee that not only the identification of the user is not possible, but also that his location and displacement cannot be tracked.

The idea behind the density maps is based on the observation that if a variable is fairly constant in a region of the map, a low number of locations would be required to reconstruct it. On the other hand, if the pollutant has high variability in a region, more locations (measurements) would be needed to better characterize it.

This study analyzes two cases. The first case is the univariate case, where only one variable, e.g, CO or CO$_2$, or any other, is used to generate the desired location of the users. The second case is the multivariate case, in which now a set of control variables is used to create the desired density map. These cases are discussed next.

6.1.1 The Univariate Case

In order to measure the level of variability of the variable under investigation, a gradient-based metric has been defined. The general definition of the gradient is that of a vector field that points in the direction of maximum change and its magnitude is the largest rate of change. Considering a two-dimensional scalar function $f(x, y)$ (for the pollution application case, $f$ is either temperature, RH, CO, CO$_2$, air quality, or combustible gases), the gradient is calculated by the partial derivatives of $f$ by:

$$\nabla f = \frac{\partial f}{\partial x} \hat{i} + \frac{\partial f}{\partial y} \hat{j}$$  \hspace{1cm} (6.1)

where $\frac{\partial f}{\partial x}$ is the partial derivative of $f$ along the $x$ axis, e.g., longitude, and $\frac{\partial f}{\partial y}$ is the partial derivative of $f$ along the $y$ axis, e.g., latitude. Therefore, the magnitude of the gradient is:

$$|\nabla f| = \sqrt{\left( \frac{\partial f}{\partial x} \right)^2 + \left( \frac{\partial f}{\partial y} \right)^2}$$  \hspace{1cm} (6.2)

Since only discrete points of the functions are available, due to the resolution used during the spatial interpolation process (38×38 points for Figure 6.2(a)), the gradient is also calculated in a discrete way. In other words, the gradient and the magnitude of the gradient
Figure 6.2(a) shows how every point in the gradient map has a corresponding magnitude and direction. Nonetheless, in this approach, only the magnitude of the gradient is considered for each point.

![Gradient field and contour maps for CO2 in the USF campus.](image)

(a) Gradient field for CO2.  
(b) Contour map and corresponding locations for CO2.

Figure 6.2: Gradient field and contour maps for CO2 in the USF campus.

The density map is generated as follows. First, as shown in Figure 6.2(a), the area of interest is divided into regions using a virtual grid. For the pollution dataset, the area was divided in 25 \((5 \times 5)\) regions. Each region will be identified with the duple \((i, j)\), where \(i\) is the coordinate along the \(x\) axis and \(j\) along the \(y\) axis, each varying from 1 to 5. Then, the Mean Magnitude of Change, \(MMC_{f,ij}\), is calculated for each region \((i, j)\) of the scalar function \(f\), defined as the mean of the magnitude of the gradient for each region, as follows:

\[
MMC_{f,ij} = \frac{1}{|\Delta f_{ij}|} \sum_{x,y} |\nabla f_{ij}(x,y)|
\]  

Remember that this is a discrete case and the magnitude of the gradient is calculated on a per point basis. In this case, \(MMC_{f,ij}\) implies the mean of the magnitude of the gradient.
for the points that belong to the region \((i, j)\). For instance, since \(38 \times 38\) points and \(5 \times 5\) regions are used, each region has approximately \(7 \times 7\) points to calculate the average.

Finally, the number of locations per region to take samples from during the next round is determined using the \(MMC\) per region and two input parameters (\(min\_loc\) and \(max\_loc\)) that limit the minimum and maximum number of locations required for each region. These are input parameters given by the administrator of the PS system because they are application dependent and vary according to the dynamics of the variable, the type of events to be detected, the total budget for the data collection process, etc. In Figure 6.2(b) the minimum and maximum number of users per region are 0 and 3, respectively, and consequently, 0 users are assigned to the region with the lowest \(MMC\) and 3 users to the region with the highest \(MMC\). The number of locations for the other regions is assigned using a linear mapping between these two limits. Figure 6.3 gives you a pictorial representation of this linear mapping. Again, from Figure 6.2(b), it can be seen how after 8 iterations, the density map clearly follows the regions of the map where the \(CO_2\) behaves with a higher variability and reduces the density (number of users) in the regions of the map with an almost constant behavior.

It is important to explain the reasons and implications of defining regions and not the exact location of the users. A desired density of users is generated on a per region basis because the exact location of the users cannot be controlled due to their natural mobility. Also, some of the privacy mechanisms guarantee the protection of the user’s information through the use of anonymization or obfuscation techniques that modify the users’ locations, making impossible to assume that the users will be located on specific points in the map. That is why regions are created to encourage users to be in an area of the map, but not exactly in a certain position at a certain time.

The PS system administrator, then, has two important decisions to make. First, she has to decide the number of regions in which the area is going to be divided. Then, she has to decide the maximum and minimum number of users per region. Increasing the size of the regions would reduce the total number of users in the system, but it might
imply that some events could be missed or not detected. On the other hand, small regions would guarantee the detection and visualization of small events in the map, but would imply a generous budget for the data collection process. Therefore, it is necessary to find the optimal trade-off between these two extremes considering parameters such as, budget, speed of stabilization, and dynamics of the variables.

### 6.1.2 A Multivariate Extension

So far a gradient-based metric has been presented, but it only considers the dynamics of a single variable to generate the desired locations of the users. Unless all variables are highly correlated, this density map might not properly characterize the other variables of interest, as it will be presented in Section 6.2. Taking this into account, the metric is now extended to the multivariate case. Having this in mind, control variables must be defined. These control variables are the set of variables that will influence and that will be considered in the generation of the density map. The new multivariate MMC ($MMC_{m_{ij}}$) is define...
as the average of the univariate MMC for the set of control variables \((f, g, \ldots, n)\) for each region \((i, j)\):

\[
MMC_{m_{ij}} = \frac{MMC_{f,ij} + MMC_{g,ij} + \ldots + MMC_{n,ij}}{n}
\]  \hspace{1cm} (6.4)

In a similar way to the univariate case, \(\text{min loc}\) users are assigned to the region \((i, j)\) with the lowest \(MMC_{m_{ij}}\), and \(\text{max loc}\) users to the region with the highest \(MMC_{m}\). The number of locations for the other regions is assigned using a linear mapping between these two limits.

### 6.2 Experiments and Analysis

In this section the experiments and the corresponding analysis are presented, both for the univariate and the multivariate cases. To determine the number of users for the first iteration on a real application, a default value per region can be calculated following temporal records about the behavior of that variable for that area in the map. Nevertheless, in order to create a good initial point for the experiments, the initial location of the users are predefined to be on the surroundings of the campus, initially guaranteeing the quality of the estimation to be very poor. Based on these measurements, a complete map of the area can be generated using known interpolation techniques, such as multivariate kriging [24, 68]. Once the complete status of the variables has been estimated, it is possible to generate the density maps for the next round of the data collection process.

Using the number of users defined by the density map from the previous iteration, the locations are selected from each region following a uniform distribution. When sampling each region, if a location is selected twice, an additional random location should be necessary to achieve the desired density of users for that region. Nevertheless, in that case, a new location is not selected and that region is simply left with a lower density than expected. With this decision, the case where the incentive mechanism is not successful obtaining the desired number of users for a region, is being modeled. In order to measure the quality of
Figure 6.4: $R^2$ for all variables in the univariate case using CO as the variable of control. The mean number of users for the 5 last iterations is 23.

In the estimations, the coefficient of determination ($R^2$) is utilized, in the same way that has been used in the previous chapters.
6.2.1 The Univariate Case

The results for the univariate case consider the generation of the density map using only one control variable, and sampling all variables following the desired density per region. First, the evolution of the system is presented, during 10 iterations of the whole process of collecting, estimating, and generating maps. The mean value of the coefficient of determination ($R^2_{mean}$) and the total number of users in the system ($N_{mean}$) for the last 5 iterations are used to characterize the performance of the experiment. Figure 6.4 presents the coefficient of determination for all the environmental data by using carbon monoxide (CO) as the control variable. The parameters max_loc and min_loc have been fixed to 3 and 0, respectively. It is clear how after 4 to 5 iterations, the system achieves its maximum performance.

It is also important to notice how the control variable (CO) maintains a stable behavior after 3 iterations and its coefficient of determination never decreases, while the quality of the estimations for the other variables is very poor in some cases. This is an expected behavior since the system is tracking the characteristics of this single variable in order to improve the quality of the estimations. Figure 6.5 presents the final location of the users after 10 iterations. The quality of these estimations should be compared against the original variables in Figure 6.1. The system clearly generates optimal locations for CO, but these locations might not be appropriate to estimate the rest of the variables, such as CO$_2$ in the graph, which has a low value for $R^2$. Once the system achieves its maximum performance, the mean total number of users in the system is stable as well, and for this specific case was 23.

6.2.2 Increasing the Number of Users per Region

As it was mentioned previously, the parameters max_loc and min_loc are application-dependent. Assuming that the system administrator has a generous budget, the maximum number of users per region could be increased, but also an increase in the quality of the estimations is expected. In order to simulate this, min_loc is fixed to zero and max_loc
Figure 6.5: Location of the users after 10 iterations for all variables, when using only CO as the control variable.

is gradually increased. For instance, Figure 6.6 shows the results of this experiment when using relative humidity (RH) as the control variable, but this behavior is similar for the rest of the variables. It can be seen from Figure 6.6(a) that not only the quality of the estimations improves with the number of users, but also the number of iterations necessary to achieve the maximum performance decreases. Also, from Figure 6.6(b), it can be seen how the total number of users in the system increases when incrementing $max\_loc$ and it
Figure 6.6: Increasing the number of users ($max_{loc}$) per region. The $min_{loc}$ value is zero in all cases.
maintains a stable value after a small number of iterations. Not only that, the speed at which the stable state is reached also increases with the maximum number of users per region.

Nevertheless, while changing $\text{max}_{\text{loc}}$ from 2 to 4 improves the accuracy from 0.6012 to 0.8654 (with a total of 39 users), when incrementing $\text{max}_{\text{loc}}$ from 4 to 6 the accuracy only achieves 0.9456 (with a total of 62 users). It is clear how after some point, increasing the maximum number of users per region does not bring a notorious benefit to the system, but it will significantly affect the available budget. Table 6.1 summarizes these results when using different control variables. The system administrator can use these results to define the optimal parameters based upon the desired estimation quality and the available budget.

Table 6.1: Coefficient of determination $R^2$ and total number of users.

<table>
<thead>
<tr>
<th>Variable</th>
<th>$\text{max}_{\text{loc}}$</th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>2</td>
<td>4</td>
<td>6</td>
</tr>
<tr>
<td>$R^2_{\text{mean}}$</td>
<td></td>
<td>0.9027</td>
<td>0.9431</td>
<td>0.9731</td>
</tr>
<tr>
<td>$N_{\text{mean}}$</td>
<td></td>
<td>16.2</td>
<td>31.4</td>
<td>49.8</td>
</tr>
<tr>
<td>Temperature</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Relative Humidity</td>
<td></td>
<td>0.6012</td>
<td>0.8654</td>
<td>0.9456</td>
</tr>
<tr>
<td>$N_{\text{mean}}$</td>
<td></td>
<td>18.0</td>
<td>39.6</td>
<td>62.4</td>
</tr>
<tr>
<td>Air Quality</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$R^2_{\text{mean}}$</td>
<td></td>
<td>0.9243</td>
<td>0.9712</td>
<td>0.9324</td>
</tr>
<tr>
<td>$N_{\text{mean}}$</td>
<td></td>
<td>21.6</td>
<td>35.2</td>
<td>44.2</td>
</tr>
<tr>
<td>CO$_2$</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$R^2_{\text{mean}}$</td>
<td></td>
<td>0.8257</td>
<td>0.9078</td>
<td>0.9380</td>
</tr>
<tr>
<td>$N_{\text{mean}}$</td>
<td></td>
<td>20.4</td>
<td>38.0</td>
<td>58.0</td>
</tr>
<tr>
<td>CO</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$R^2_{\text{mean}}$</td>
<td></td>
<td>0.7891</td>
<td>0.9592</td>
<td>0.9707</td>
</tr>
<tr>
<td>$N_{\text{mean}}$</td>
<td></td>
<td>15.6</td>
<td>33.8</td>
<td>44.6</td>
</tr>
<tr>
<td>Combustion Gases</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$R^2_{\text{mean}}$</td>
<td></td>
<td>0.7951</td>
<td>0.9582</td>
<td>0.9725</td>
</tr>
<tr>
<td>$N_{\text{mean}}$</td>
<td></td>
<td>15.6</td>
<td>37.0</td>
<td>50.0</td>
</tr>
</tbody>
</table>
Figure 6.7: $R^2$ for all variables for the multivariate case, using RH and CO are the control variables.

6.2.3 The Multivariate Case

The multivariate case is presented now, considering the results when using two control variables. Figure 6.7 shows the coefficient of determination for all variables when using
relative humidity (RH) and carbon monoxide (CO) as the control variables of the system. When these results are compared to the univariate case presented in Figure 6.4 where CO was the only control variable, it can be seen that while for CO the $R^2_{\text{mean}}$ remains almost at the same value (from 0.89 to 0.90), the performance significantly improves for the rest of the variables. Temperature improves from 0.86 to 0.95, RH from 0.63 to 0.82, air quality from 0.75 to 0.95, CO$_2$ from 0.52 to 0.73, and combustible gases from 0.84 to 0.92. The most impressive result is that this improvement is gained in all variables while only incrementing the total number of users from 23 to 27.

After experimenting, it has been found that including a large number of control variables requires an increment on the number of users, and also generates a density map with an almost uniform distribution of the users in the space. Even thought this is the ideal case, complete coverage of the area by the users, this also implies a larger or unlimited budget for the PS system.

This previous behavior is the result of a wise selection of the control variables. If two variables are highly correlated, it is not necessary to use both since selecting only one of them will bring enough information to effectively generate the density map. After experimenting it was possible to identify how RH and CO would bring enough information to track the changes of almost all variables.

6.3 In Conclusion: Density Maps

The main conclusions that can be drawn from this chapter about density maps are:

- Through the use of a gradient-based metric, it is possible to establish the density of users per region in order to properly characterize the variables of interest, while maintaining a low and stable number of users. The experimental results show how this univariate approach successfully determines the optimal locations to characterize only one variable.
• By using a multivariate extension through the use of “control variables”, it is now possible to characterize the dynamics and spatial events of several variables at the same time. The experimental results show how the quality of the estimations for all variables has been improved when using this new multivariate approach.
CHAPTER 7

CONCLUSIONS

In this dissertation different aspects of Participatory Sensing applications have been tackled. A framework for the design and implementation of PS systems has been presented, which addresses the unique characteristics and challenges related to this type of applications. This has been the first attempt to analyze PS applications from a system viewpoint.

The framework proposed here describes the main modules and challenges to design and implement a PS system. However, it provides a systemic view without looking at other issues and challenges that may arise in the first integrator device. For example, if several PS applications are run simultaneously in the same device, they might need to use the same sensors and compete for the same resources. In that case, new mechanisms are needed in the cell phone to use its resources in the best efficient manner, perhaps even sharing the data among applications. This is a non-trivial problem, recently recognized in [112].

Having presented the framework, this dissertation focused on 3 different modules of it: data verification, data visualization and density maps generation. For data verification a new hybrid algorithm for spatial outlier detection and removal has been proposed and implemented. It considers aspects that could be found in real PS systems, such as the uneven spatial density of the users, malicious users, and the lack of accuracy and malfunctioning sensors. While the median algorithm is really fast, it fails to detect the spatial outliers when the number of malicious users are increased. After experimenting, it has been shown how the Delaunay triangulation and Gaussian Mixture Models can be used to build neighborhoods based on the spatial and non-spatial attributes of each location. Using these neighborhoods is is possible to detect and remove the spatial outliers while being computationally efficient.
In order to simulate users attacking the system, a complete adversary model has been
generated. This model not only considers noisy and faulty sensors, but also different scenar-
ios, such as random and cooperative attacks made by malicious users. Under the different
test scenarios the hybrid algorithm always matches the performance of the best spatial es-
timator (kriging) while significantly reducing the total execution time, which is particularly
important given the large amount of data generated by participatory systems.

In the area of data visualization for PS system, a comparative study between the use
of Markov Random Fields and kriging techniques has been created. After showing the su-
periority of kriging in terms of the quality of the estimations, the technique is applied to
make spatial interpolations for one variable. The results are very revealing in the sense
that the technique interpolates data with high granularity, i.e., provides more precise in-
formation about a variable of interest in a particular place than the information provided
when interpolating data from traditional measuring systems. This conclusion is extremely
important from the users and system point of view. Users will be able to obtain more
precise information about a variable that affects them directly. This feature justifies or
somehow proves the usefulness and superiority of participatory sensing systems compared
with traditional measuring systems. This dissertation has introduced a new technique for
the spatial interpolation process (kriging along with PCA or ICA), and presents a solution
using expert predictors for each of the variables of interest.

The use of real pollution data imposes a strong problem due to the high levels of noise
in the data, and the continuing changing conditions of the environment, but the implementa-
tion of PCA and ICA has given the necessary flexibility to achieve high accuracy in all
measured variables. With these spatial estimations, the data coming from the participants
can be validated and visualized for easy analysis.

In order to close the loop in the framework, the use of density maps has been proposed.
These maps determines the number of users and their locations where to take samples from
in order to accurately reconstruct the variable of interest. Density maps are calculated on
a round by round basis and fed back to the incentive mechanism so that only those users
located in those locations are encouraged to participate. The experiments have shown how
the density maps greatly improve the quality of the estimations while maintaining a stable
and low total number of users in the system.

Even though the multivariate extension of the density maps has proven useful, it still
requires more research and experimentation. A mechanism to automatically select the
control variables is still necessary in order to successfully estimate the variables of interest
without greatly increasing the total number of users, and therefore incurring in additional
costs for the system. Some approaches could include metrics based on the correlation matrix
and factor analysis.

Future work on this and similar participatory sensing systems is needed in several di-
rections. Sensor integration and minimization is a very important area. Users will not
participate in the application if they have to carry another (bulky) device. Finally, par-
ticipatory sensing systems will generate huge amounts of data. This presents a scalability
problem that needs to be addressed from different directions. Finding ways to eliminate
redundant or useless data, transmit less data, process and store data efficiently, and similar,
are of paramount importance. Finally, appropriate techniques to merge both systems (PS
and traditional) must be developed in order to exploit the advantages that both systems
have (spatial resolution for PS and traditional stations accuracy).
REFERENCES


[42] E. Miluzzo, M. Papandrea, N. Lane, H. Lu, and A. Campbell, “Pocket, bag, hand, etc. - automatically detecting phone context through discovery,” in *International Workshop on Sensing for App Phones (PhoneSense)*, nov 2010.


APPENDICES
Appendix A: P-Sense - Permission for Reuse

Thesis / Dissertation Reuse

The IEEE does not require individuals working on a thesis to obtain a formal reuse license, however, you may print out this statement to be used as a permission grant:

Requirements to be followed when using any portion (e.g., figure, graph, table, or textual material) of an IEEE copyrighted paper in a thesis:

1) In the case of textual material (e.g., using short quotes or referring to the work within these papers) users must give full credit to the original source (author, paper, publication) followed by the IEEE copyright line © 2011 IEEE.

2) In the case of illustrations or tabular material, we require that the copyright line © [Year of original publication] IEEE appear prominently with each reprinted figure and/or table.

3) If a substantial portion of the original paper is to be used, and if you are not the senior author, also obtain the senior author’s approval.
Appendix B: G-Sense - Permission for Reuse

Thesis / Dissertation Reuse

The IEEE does not require individuals working on a thesis to obtain a formal reuse license, however, you may print out this statement to be used as a permission grant:

Requirements to be followed when using any portion (e.g., figure, graph, table, or textual material) of an IEEE copyrighted paper in a thesis:

1) In the case of textual material (e.g., using short quotes or referring to the work within these papers) users must give full credit to the original source (author, paper, publication) followed by the IEEE copyright line © 2011 IEEE.
2) In the case of illustrations or tabular material, we require that the copyright line © [Year of original publication] IEEE appear prominently with each reprinted figure and/or table.
3) If a substantial portion of the original paper is to be used, and if you are not the senior author, also obtain the senior author's approval.