Localization in Urban Spaces using a Collaborative WiFi+GSM-fingerprint-based Approach

Kristian Samuel Cugia Peña

Universidad del Cauca
Facultad de Ingeniería Electrónica y Telecomunicaciones
Grupo de Investigación: Ingeniería Telemática
Departamento de Telemática
Popayán, Julio de 2012

Localization in Urban Spaces using a Collaborative WiFi+GSM-fingerprint-based Approach

Kristian Samuel Cugia Peña

Director:
Fernando Aparicio Urbano Molano

Co-director:
Martin Wirz
ETH Zürich, Suiza (Swiss Federal Institute of Technology Zürich)

Universidad del Cauca
Facultad de Ingeniería Electrónica y Telecomunicaciones
Grupo de Investigación: Ingeniería Telemática
Departamento de Telemática
Popayán, Julio de 2012

Preface

I would like to give special thanks to Martin Wirz and Daniel Roggen for friendly welcoming me and for his pleasurable collaboration, support and guidance during the conduction of this work, through their very valuable comments, suggestions and questions, they encouraged me and helped to make my stay at ETH Zürich one of the most valuable and enriching academic and personal experiences I have had.

I am grateful to Prof. Gerhard Tröster for allowing me to participate and conduct this work at the Electronics Laboratory. Thanks to all the members of the group who directly or indirectly welcomed me and/or made valuable comments and inputs.

Special thanks to Rafael Rengifo, Dean of the Faculty of Electronics Engineering and Telecommunications at University of Cauca for his valuable collaboration and help. Same to Fernando Aparicio Urbano, who acts as director of the present work.

I am very grateful to my mother, my family and friends for their continuous personal support and suggestions. I apologize if unintentionally ommited anyone to whom acknowledgement is due, has been under no reason a willful act.

Kristian Cugia

Abstract

State-of-the-art fingerprinting-based localization methods relying on WiFi/GSM information provide sufficient localization accuracy for many mobile applications and work reliably in urban areas and indoors. These methods assume that each location contains a unique combination of signal strength readings. To obtain a location estimation, a mobile devices gathers signal strength readings and with the help of a fingerprinting algorithm, the closest match in a reference database is found. Building this reference database requires a training set consisting of geo-referenced fingerprints. Traditional approaches require manual labelling of the reference locations or GPS information. This work proposes a collaborative, semi-supervised WiFi/GSM-based fingerprinting method where only a small fraction of all fingerprints needs to be geo-referenced. This allows for automatic indexing of areas in the absence of GPS reception as found in urban spaces and indoors without requiring manual labelling of fingerprints. Taking advantage of the characteristic that the similarity between two fingerprints correlates to the distance between their corresponding locations, this method applies multidimensional scaling to generate a topology estimation of the training set. With the help of a subset of geo-referenced fingerprints, the topology estimation is anchored to physical locations now serving as a reference database. Further fingerprints can be used to refine and extend the topology estimation. Hence, the covered space grows gradually. An evaluation of the approach is performed using an urban-scale dataset showing that the method can locate a mobile device with a median accuracy of 30 m. Hereby, only 7% of the fingerprints are geo-referenced. Further, the localization error decreases and converges to a value comparable to related work as new fingerprints are added to the reference database. A promising application of the method is seen by combining it with existing fingerprinting systems to extend their functionality into areas where a GPS-based indexing is not possible.

Keywords

Localization – Mobile Phone – WiFi – GSM – Fingerprints – MDS – GPS Anchor Points

Contents

1. Introduction 1
   1.1. Motivation . . . 1
   1.2. Problem Statement . . . 1
   1.3. Approach and Outline . . . 2

2. Related Work 4
   2.1. Wi-Fi based Positioning . . . 4
   2.2. GSM based Positioning . . . 6
   2.3. Similarity Measurement . . . 8
   2.4. MDS . . . 10

3. Experiments & DataSet 12
   3.1. CoenoSense Extension . . . 12
      3.1.1. Description . . . 12
   3.2. Aim of Experiment . . . 13
      3.2.1. Description & Procedure . . . 14
   3.3. Dataset . . . 15
   3.4. WiFi Scan . . . 15
   3.5. GSM Scan . . . 15

4. Similarity Measurement 16
   4.1. Similarity and Distance Relation . . . 16
      4.1.1. Discussion . . . 17
   4.2. Data Analisys . . . 18

5. MDS 33
   5.1. Brief Introduction . . . 33
   5.2. Topology Reconstruction . . . 38
      5.2.1. Topology Results Phenomena . . . 40

5.2.2. Algorithm Operation Limitations 49
5.3. Proposed Algorithm 50
5.3.1. Generating a reference topology 52
5.3.2. Density Considerations 60

6. Results 61
6.1. Simulation 61
6.1.1. Fingerprint Similarity vs. Distance 61
6.1.2. Topology Estimation 62
6.1.3. Evolution of the reference topology 65

7. Discussion 68
7.1. Summary and Contributions 68
7.2. Findings, Achievements and Conclusions 69
7.3. Limitations 70
7.4. Outlook 70

Bibliography 72

List of Figures 76

List of Tables 78

Appendix A. Excerpts of the Code 79

Appendix B. Similarity Measurement Results 83

CHAPTER 1

Introduction

1.1. Motivation

Knowing the geographical position of a person enables a large number of location-based mobile applications [1]. State-of-the-art mobile phones contain multiple technologies to provide such location information including GPS, WiFi and GSM-based approaches. Despite its largely spread use and commercialization, GPS-based localization faces some limitations as it requires a clear view of the sky, it provides accurate positions in open sky conditions and less accurate ones or none in urban and indoor areas [2]. These, however, are places where people spend most of their time [3].

The localization problem in indoor venues and urban spaces has then became object of research. Thanks to the vast penetration of GSM and WiFi networks, exploiting these existing infrastructures for localization purposes has found great interest. The achievable location accuracy has been found to be sufficient for many mobile phone applications. Additionally, WiFi- and GSM-based approaches have the advantage of performing well in urban areas and indoor venues [1]. However most of the existing approaches need previous indexing of the WiFi and/or GSM information to geographical locations, requiring GPS availability for this process.

1.2. Problem Statement

Recently, so-called fingerprinting approaches have found great interest in the research community for localization purposes. Hereby, a reference database is built where a list of access points (APs) and their corresponding received signal strengths at given locations are called fingerprints.

The assumption is that each fingerprint is unique across the space and thus represents a particular geographical location. To be localized, a mobile device gathers signal strength readings and with the help of a fingerprinting algorithm, the closest match in the reference database can be found revealing a location. Bahl *et al.*'s RADAR localization system [4] was a pioneer effort in that direction. In a more recent work, LaMarca *et al.* [3], achieve $20 - 30$ meters median localization accuracy in urban areas with their Place Lab system.

For such fingerprinting approaches to work, training data is required to build the reference database consisting of geo-referenced -usually using GPS- fingerprints.

For collecting such GPS-referenced fingerprints in urban environments, approaches like wardriving [5] and war-walking [6] became popular. War-walking tends to take more time but provides better accuracy and larger coverage in metropolitan areas as some regions in a city are only accessible by pedestrians [6]. Fingerprinting efforts can be minimized by e.g. geocoded information to bootstrap fingerprinting databases [7]. Following a collaborative approach, fingerprint databases are automatically updated when GPS and WiFi are active on a user's phone [8].

Many of the existing methods assume the availability of accurate GPS signals during the recording of the training set. GPS reception, however, is not always available in many urban areas as well as indoors, limiting the possible indexing space significantly and thus the usefulness of such localization systems. Hence, to provide extensive coverage and high accuracy for urban positioning, methods are required to be able to index urban spaces also in the absence of GPS-based reference information.

While existing approaches rely on manual labeling of the reference locations [9] or require expensive equipment [10], in this work we present a fingerprinting approach which does not require a reference location for each fingerprint in the training set. Only a small number of anchor points is required. Contribution is threefold: 1) A collaborative, semi-supervised WiFi and GSM (termed WiFi+GSM in the following) fingerprinting method that only require geo-referencing of a fraction of the fingerprints is proposed, by taking advantage of the characteristic that the similarity between two fingerprints correlates to the distance between the location of the recordings. By applying Multidimensional Scaling (MDS) [11] on the similarity information, a topology of fingerprints which can be mapped to a geographical coordinate system is obtained using some geo-referenced fingerprints serving as anchor points. This reference topology can be used to locate new fingerprints. 2) The topology can be updated with new fingerprints to increase the localization accuracy and to extend the covered space and is therefore suitable for a collaborative approach. 3) Evaluation results using an extensive urban data set that provides evidence for the feasibility of our approach is presented.

1.3. Approach and Outline

Figure 1.1 gives a general overview of the approach we intend to use. From every set of scans taken with different mobile devices, a fingerprint is builded up, then through a pairwise similarity comparison of these fingerprints in relation with their geographical separation distance, a model of its behaviour is should be obtained, from this model and by making use of an algorithm, we intend to obtain a topology that represents the real geographical locations where the scans were made.

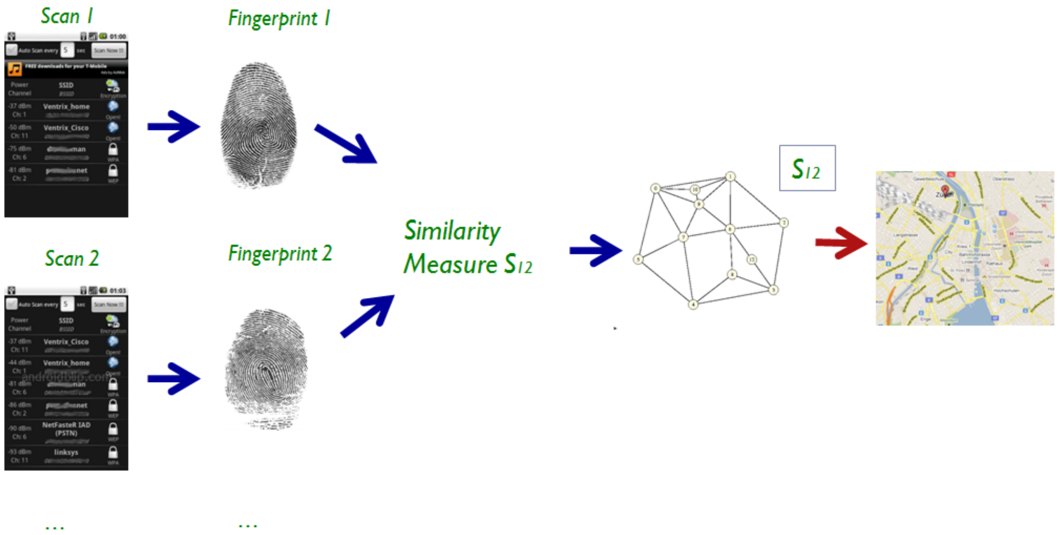


Figure 1.1.: Addressed approach.

This chapter acted as introduction to this thesis. The next chapter will present the state of the art in related literature. Through the thesis we will rely on data, thus we require to obtain a data set through the conduction of an experiment. For obtaining the desired data, the extension of the existing CoenoSense plattform and the conducted experiment are presented in chapter 3.

The analysis of WiFi and GSM data, and its behaviour in relation to spatial distance is presented in chapter 4. An introduction to the basic principles of MDS its expected performance, and the proposed algorithm operation is presented in Chapter 5; Chapter 6 shows the evaluation and performance of the proposed algorithm. The results are discussed and summarized in Chapter 7 by showing the limitations, presenting a conclusion and proposing further work that could be done in the area. Some considerations that should be taken into account for a real life implementation are also suggested.

CHAPTER 2

Related Work

The most commonly available and used localization technology nowadays is the Global Positioning System GPS. Altough highly efective and accurate in open enviroments, GPS does not work well in indoor enviroments, urban canyons or in such areas with limited view of the sky: places where people spend most of their time. A study in [3] shows that GPS is available only 4.5% of the time for a device carried in user's pocket or during a normal day, these values represent the worst case scenario and are elevated when both mobile and stationary times are considered. Some works have been done in localization approaches to work indoors, for example Infrarred [12], Ultrasound [13] and Bluetooth [14] have been explored as alternatives, even though they work well indoors, deploying them in a wide area is either cost prohibitive or dificult from a technical perspective, for example, due to infrarred interference from the sun.

A review of the existing positioning methods for wireless networks is made in [15], authors examinate a broad range of techniques from Angle of Arrival (AoA), to Time of Arrival (ToA), propagation and geometrical models. Most of these methods focus on wireless sensor networks and thus require a complete knowledge of the network and hardware.

2.1. Wi-Fi based Positioning

Several fingerprinting approaches have been used for localization purposes, using either Wi-Fi or GSM information. Bahl *et al.'s* RADAR localization system [4] was a pioneer effort that estimated the distance from a client to each beacon using its received signal strenght indication RSSI, then trilaterated a location estimate. Niculescu *et al.* used relative angles, rather than distances, from clients to beacons [16]. The use of time difference of arrival techniques has been studied for example in [17] and [18] to generate precise distance estimates than RSSI alone could provide. Hightower *et al.* focused on removing the need for fixed infraestructure through rapid and flexible RF deployments [19]. The main difficulty that all of these approaches face is that reflections, difractions, absorptions, multi-path effects and the presence of objects *e.g.*

people, cars, often affect signal propagation models and in turn, angles and distance estimates.

Subsequent work of the RADAR group dealed with the signal modeling and triangulation by proposing the use of RF Signatures, or fingerprints. Due to walls, obstacles, distance and structures in indoor enviroments, the observed signals in a particular area will be different from those in other spaces, even adjacent ones. This also applies to GSM signals, which have been used for localization purposes due to its behavior in urban and indoor enviroments, further Many RF sources are geographically fixed which makes the signals in most spaces fairly consistent over time. Together the RF signals observed in a space form that's space RF Signature or fingerprint. RADAR's approach works as follows: First a database containing the signatures from all the spaces is constructed, this phase is often called *training* or *survey* phase, given this database a user device can gather a current RF signature, then find the closest match in the previously constructed database. The space with the closest match is returned as the result. This is called the *use* phase.

The RADAR effort obtained a medium accuracy of 2-3 meters, which inspired future research in the topic.

An important tradeoff while deploying this kind of location systems is the obtained or desired accuracy versus the training phase or calibration effort involved. Developing such a method for large scale urban areas would require a high effort training phase in order to obtain a good accuracy. The classical fingerprinting algorithm is based on the RADAR mechanism [4], to position a device, the algorithm uses a previously indexed set of fingerprints or RF signatures to compare with the current fingerprint to find the fingerprint that is the closest match to the positioning scan in terms of APs seen and their corresponding signal strengths. The Euclidean distance in signal strength is calculated between the observed signal strengths in the current fingerprint and the recorded ones in the stored fingerprints. Suppose that a positioning scan discovered three APs: A, B, and C with corresponding signal strengths S_a, S_b, S_c. For each match between the actual scan and a fingerprint S'_a, S'_b, S'_c stored in the database, and according to the Euclidean distance definition, the distance is computed as equation 2.1:

$$\sqrt{(S_a - S'_a)^2 + (S_b - S'_b)^2 + (S_c - S'_c)^2} \tag{2.1}$$

It then selects the k fingerprints with the smallest distance to the observed scan as potential indicators of the observed scan. The location of the device is estimated as an average of the latitude and longitude coordinates of the best k matches [5]. Location estimation accuracy is highly dependent on the density of the set of collected fingerprints. Many indoor WiFi and GSM localization methods using fingerprinting approaches as [4] collected fingerprints at a density of around one fingerprint per square meter. Authors in [20] investigated how well fingerprinting works with sparser calibration and less uniformly distributed set of fingerprints at a metropolitan scale.

In [5] the Intel PlaceLab group relied on user-contributed data collected by *war driving*, the process of using software on Wi-Fi and GPS equipped mobile computers and driving or walking through an area collecting traces of Wi-Fi access points. MIT group at [21] developed a system which in an organic, i.e. crowdsourced, way eliminates the necessity of a training phase to save a set of fingerprints of a determinated area, and does not require GPS to build the radio map. This system has an user interface that prompts the user to select his location by indicating on a labeled floorplan his position if there is no match between the current scan and the actually saved data, or if the match is of low confidence.

A similar work by Bolliger in [9], presents an application that in a collaborative way, similar to [21], builds in an incremental and collaborative approach a database with fingerprints. Similar works were done by [5] where the training process is shifted by the so called *war driving*, and by [22] which bases the training process on a simplistic algorithm that relies on known positions of access points on an university campus to find a user's location.

There are many different variations of the fingerprinting localization algorithm, in [5] two variations are shown. When the algorithm is not able to find stored fingerprints with the same set of APs as heard in the scan, the search is expanded to look for fingerprints containing subsets of APs, the algorithm matches fingerprints that have at most p different APs between the stored fingerprints and the scan. Also, if the scan shows an AP that never appears in the database or *radio map*, for example a recently deployed AP, it is ignored. These modifications allowed this approach to improve the matching rate from 70% up to 99%. According to this work, $p = 2$ provides the best matching rate without reducing overall accuracy.

Another adaptation to the fingerprinting localization algorithm is introduced by [5]. It adopts a *ranking*. Fingerprinting is based on the assumption that Wi-Fi devices used for training and positioning measure signal strengths in the same way or scale. If that is not the case, due to differences caused by manufacturing variations, antennas, orientation… One can not directly compare the signal strengths nor derive distances from them. The algorithm proposed by RightSpot in [23] proposed that instead of comparing absolute signal strengths, to compare a list of access points sorted by signal strength in descending order. Then the comparation is made by using the Spearman rank-order correlation coefficient [24]. However, according to [5] the *ranking* algorithm performs quite poorly in low AP densitiy zones.

There are different adaptations for positioning algorithms, either for the training phase or for the test phase. For example the Centroid algorithm [5], which is the simplest positioning method, during the training phase estimates a geographic location for an access point by computing the arithmetic mean of the positions reported in all the readings. Using the radio map, the Centroid algorithm positions the user in the center of all the APs heard during the scan by computing an average of the estimated positions of each of the heard APs. There is also a weightened version of this algorithm that assigns weights according to the reported signal strength during a scan. Statistical positioning algorithms are also shown in [20], where the position is calculated by using probabilistic distributions and assigning probabilities to each position where the user could stay.

Summarizing, Wi-Fi fingerprinting localization methods either require a training phase and GPS availability to build a radio map, or require the user's colaboration to determine his location either by providing GPS information or by selecting his position in a map.

2.2. GSM based Positioning

Global System for Mobile Communication (GSM) is the most widespread cellular telephony standard in the world, with deployments in more than 210 countries by over 676 network operators [25]. In North-America, GSM operates on the 850 MHz and 1900 MHz frequency bands. Each band is subdivided into 200 KHz wide physical channels using Frequency Division Multiple Access (FDMA). Each physical channel is then subdivided into 8 logical channels based on Time Division Multiple Access (TDMA). There are 299 non-interfering physical channels available in the 1900 MHz band, and 124 in the 850 MHz band, totaling 423 physical channels.

A GSM base station is typically equipped with a number of directional antennas that define sectors of coverage or cells. Each cell is allocated a number of physical channels based on the expected traffic load and the operator's requirements. Typically, the channels are allocated in a way that both increases coverage and reduces interference between cells. Thus, for example, two neighboring cells will never be assigned the same channel. Channels are, however, reused across cells that are far-enough away from each other so that inter-cell interference is minimized while channel reuse is maximized. The channel-to-cell allocation is a complex and costly process that requires careful planning, and typically involves field measurements and extensive computer-based simulations of radio signal propagation [26].

Therefore, once the mapping between cells and frequencies has been established, it rarely changes. Every GSM cell has a special Broadcast Control Channel (BCCH) used to transmit, among other things, the identities of neighboring cells to be monitored by mobile stations for handover purposes. While GSM employs transmission power control both at the base station and the mobile device, the data on the BCCH is transmitted at a full and constant power. This allows mobile stations to compare signal strength of neighboring cells in a meaningful manner and choose the best one for further communication.

With GSM several approaches of fingerprinting have been made, they require a training phase where given a set of GPS-stamped GSM traces, the algorithm builds a model of an environment, which it later uses for predicting device's location.

Fingerprinting then matches every measurement in the testing set to one or more measurements observed during the training phase and then averages the true positions of the best matched measurements. Weighting by the signal strength of the best matched measurements provides better results. Authors in [26] argued that for emerging location enhanced applications, client-based GSM localization based on fingerprinting can provide an adequate solution both in terms of coverage and accuracy in a device people already carry. To dispel the notion that location systems using GSM phones are inherently less accurate than systems built for WiFi devices, they presented preliminary results showing that using GSM for indoors, it is feasible to achieve a median error of 2-5 meters, and room-level localization. Using GSM for outdoors, it is feasible to achieve a median error of 70-200 meters, and to detect places people go in their everyday lives. These results are comparable to what has been demonstrated previously for WiFi.

They also consider that onphone localization based on cellular networks is not specific to GSM. Indeed, any cellular technology that transmits stable beacons from the cellular towers (e.g., for the need of hand-off purposes) will make the on-phone localization possible. The main problem of the existing fingerprinting localization systems, whether WiFi or GSM, is the non-trivial training required for the system to become usable.

2.3. Similarity Measurement

Metric Functions The following are common functions used for similarity measurements:

Minkowsky Metric Given an m-by-n data matrix, which is treated as m (1-by-n) row vectors x_1, x_2, ..., x_m, the various distances between the vector x_s and x_t are defined in equation 2.2.

$$d_{st} = \sqrt[p]{\sum_{j=1}^{n} |x_{sj} - x_{tj}|^p} \tag{2.2}$$

For the special case of $p = 1$ the *Minkowski* metric gives the *City Block* metric, for $p = 2$ the *Euclidean* distance, and for $p = \infty$ the *Minkowski* metric gives the *Chebychev* distance.

Similarity Indices The similarity indices often have their origins in Ecology or Botanics, with the efforts to find the distribution of species among geographical zones. They are often used in to compare documents in text mining. In addition, they can be used to measure cohesion within clusters in the field of Data Mining, for example using the Cosine Similarity. In the following equations A and B represent two fingerprints, so the union would represent the information present in A and B, and the intersection represent the common information between those fingerprints.

Jaccard Index Introduced by [27] in 1912 to compare the distribution of plants among the alps, but it can be applied to numerous types of data. We see a WiFi fingerprint as a set that contains the *Basic Service Set Identification* BSSID, i.e. MAC Address, of all WiFi networks that were visible at the time of the WiFi scan. The Jaccard index is defined as the size of the intersection divided by the size of the union of the WiFi fingerprint sets, in other words the number of networks common to the two locations divided by the total number of networks at the two locations. Given by equation 2.3.

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|} \tag{2.3}$$

Sørensen Index is another statistic commonly used to compare sets of samples, developed in 1948 by botanist *Thorvald Sørensen* [28]. It processes the same fingerprint sets as the Jaccard index. Given by equation 2.4

$$S(A, B) = \frac{2|A \cap B|}{|A + B|} \tag{2.4}$$

Cosine Similarity Is a measure of similarity between two vectors by measuring the cosine of the angle between them. The result of the Cosine function is equal to 1 when the angle is 0, and it is less than 1 when the angle is of any other value. Calculating the cosine of the angle between two vectors thus determines whether two vectors are pointing in roughly the same direction.

The cosine can be derived from the dot product of two vectors as equation 2.5.

$$A \cdot B = \|A\| \, \|B\| \cos\theta \tag{2.5}$$

Thus, similarity is given by equation 2.6.

$$\cos\theta = \frac{A \cdot B}{\|A\| \, \|B\|} \tag{2.6}$$

Since the angle θ is in the range of $[0, \pi]$, the resulting similarity yields the value of -1 as meaning exactly opposite, 0 meaning independent and +1 meaning exactly the same, with in-between values indicating intermediate similarities or dissimilarities. The cosine similarity is extended to the Jaccard coefficient in case of binary distributions, this is the Tanimoto coefficient. For the present work, vector-oriented similarity indices are expected to result in better or more addecuate estimations, because they would consider both the networks and their received signal strenghts as a whole. We could think in each fingerprint as a vector, each component of this vector would be each network name and its magnitude would be the respective received signal strength, then we would have fingerprints as vectors of n-dimensions, with n being the size of the biggest fingerprint. This consideration is necessary in order to be able to operate vectors, as they must have the same dimensions. Thus, all the fingerprints would 'know' all the networks, but the networks don't seen in each fingerprint would have a zero magnitude. For the case of GSM information, the network name would be the Cell ID, but further considerations should be taken in mind, as there is a specific order for the GSM cell ditribution.

Tanimoto Coefficient [29] is an extension of the Jaccard index that is used to compare non-binary samples. Most data are not simple binary sets containing or not BSSIDs. There are scalar vectors that contain the signal strength for each network. Given by equation 2.7.

$$T\left(A, B\right) = \frac{A \cdot B}{\|A\|^2 + \|B\|^2 - A \cdot B} \tag{2.7}$$

First Kulczynski Is another similarity index used to compared sets of samples with origins in Ecology. Given by equation 2.8

$$K\left(A, B\right) = \frac{2\left|A \cap B\right|}{\left|A + B\right| - 2\left|A \cap B\right|} \tag{2.8}$$

Second Kulczynski A second version of the previously described similarity index. Given by equation 2.9

$$K\left(A, B\right) = \frac{2\left|A \cap B\right| \left|A + B\right|}{2\left|A\right| \left|B\right|} \tag{2.9}$$

Other Indices in [21] a fixed similarity index is used to compare the BSSIDs between each pair of fingerprints and another index for each pair of RSSIDs of the corresponding fingerprints.

2.4. MDS

Multidimensional Scaling is a collection of statistical techniques which explores the similarities or dissimilarities in data, they have its origins in psychometrics and psychophysics.

The history of MDS techniques begins with the work of *Torgeson* in 1952, who draw the first ideas [30]. In 1962 *Shepard* proposed a quite accurate formulation of MDS when he proved that in a known ordination of the distances between a determinated set of points, it could be possible to find a configuration of points in a low dimensional Euclidian space, which pair-wise distances represent or practically reproduce the initial configuration. Those ideas were refined by *Kruskal* in the 60's and later developed by other authors like *Guttman* and *Lingoes.* The approach of these techniques is to build metric distances by adecuately transforming disimilarities and are often known as *non-metric* multidimensional scaling.

In 1966 *Gower* proposed the *Principal Coordinates Analysis* method, which could be considered as a *metric* MDS which avoids the resolution by iterative processes used in the non-metrical techniques. In the 80's and 90's several researchers countinued the research on finding algorithms able to produce a final configuration of points which distances are as close as possible to the perceived disimilarities. MDS has been applied in many fields ranging from machine learning, computational chemistry to psychologie. For localization purposes MDS takes full advantage of connectivity or distance information between known and unknown nodes, unlike other distance approaches [31].

MDS models are defined by specifying how the given similarity data between two nodes i and j are mapped into distances $\hat{d}_{ij}$. MDS-based works on range-based localization assuming the node-node pairwise distance is measurable, and on range-free localization, only estimating node-node pairwise distance by connectivity information [32].

Despite fingerprinting methods are not novel persé, the use of ordination algorithms or techniques such as MDS for localization purposes has not been explored as much as fingerprinting. Until the extent of literature research suggests, recently there are works that applied MDS and adaptations of MDS algorithms to similarity data for localization purposes, which means that the location resolution should be between 1 and 100 meters, and the timeliness should be in seconds [31]. However these methods make use of either previously recorded information, GPS, or have been applied into Wireless Sensors Networks. Allegedly there is no work that has applied MDS in a collaborative manner to fingerprints either based on GSM information alone or a combination of Wi-Fi and GSM.

Goussevakia *et al.* [33] used MDS to map the *The World of Music* into an Euclidean space with the scope of accelerating the calculation of the shortest path between two musical pieces. MDS is applied in this work under multiple iterations in order to increase the accuracy of the obtained map. Shang *et al.* [34] applied for the first time MDS into a Wireless Sensor Network (WSN). MDS was sucessfully used to derive the position of nodes in a WSN, based exclusevely on connectivity information: who is in the range to who. Its approach initially calculates the shortest path using the *Djkstra* algorithm, then mutiplies it with the average range of radio signals to obtain each pairwise distance, and uses MDS to obtain a relative topology of the network. This relative topology can be approximated into a real topology if the position of at least three nodes is known. This work was extended in [35] to work without knowledge of the entire connectivity.

Authors in [31] developed an algorithm based on MDS and *Received Signal Strength Indication* (RSSI). In this work, the signals with RSSI values among beacon nodes, i.e. nodes with knwon location, helped to construct a real time 2D map of the network by self iterations. Nodes with unknown locations use the map to determine their locations in this network. A simulation was used to test the performance of the algorithm as well as practical experiments showed the localization efficiency and accuracy. The most significant difference of this approach is that it uses RSSI to obtain estimated distances between beacons to build a distance matrix, and then uses a MDS-based algorithm to approximate the mapping, which is shown to be more effective in both simulation and implementation. After this process, the unknown nodes can estimate their location based on the previously built map. The results of this approach, show an acceptable performance: less than 10% of the average distance between beacons.

Schläpfer in [36] applied MDS to a data set of Wi-Fi fingerprints with the aim of mapping the similarities between fingerpringts into a bidimensional space which represents the points where the fingerprints were taken. This approach used fingerprints consisting in the name of the network SSID, the MAC address of the acces point BSSID and the received signal level. According to the author, this process was made in a very short time, however, its results seem promising. Authors in [32] were motivated to improve the accuracy of fingerprinting localization algorithm using MDS. Based on the original fingerprinting location estimation, the proposed location sensing approach uses MDS and *Procrustes* analysis to improve the localization accuracy, by trying to guarantee that the configuration of points obtained by MDS matches the original configuration of points in the least-squares sense. From radio scans authors in [37], extracted dissimilarities between pairs of WiFi APs, then analyzed the dissimilarities to produce a geometric configuration of WiFi APs based on a multidimensional scaling technique. To validate the scheme, they conducted experiments on five floors of an office building with an area of 50 m by 35 m in each floor. WiFi APs were located within a 10m error range, and floors of APs are recognized without error. While they intend to locate WiFi APs, we intend to locate mobile devices based on the available WiFi and GSM information. The work that is most comparable to our effort is presented by Pulkkinen *et al.* in [38] where they generate a topology map, based on the similarity of WiFi fingerprints. They achieve a median localization error of $1.5m$ by using MDS to generate the reference database and a classic fingerprinting algorithm for the positioning part. The approach in this work, on the other hand, also uses MDS for locating fingerprints. Further, their approach was only evaluated on one floor of a building. For urban-scale deployment as envisioned in our work, additional effects have to be taken into consideration mainly due to the non-linear relation between distance estimation and fingerprint similarity. In this work, we address these issues and evaluate a method for urban-scale positioning using both WiFi and GSM readings.

CHAPTER 3

Experiments & DataSet

To investigate the relationship between the similarity of the two fingerprints and their spatial distance as well as to evaluate the accuracy, performance and efficiency of the MDS-based topology estimation, we require a data set. For this we recorded two data sets in a realistic urban environment.

Besides the following reported experiment, several and smaller tests or experiments were conducted with the aim of getting familiar with the devices and the behavior of the existing CoenoSense platform, and to evaluate the correct functionality of the intended extensions and modifications to this platform, in order to be able to collect and process WiFi and GSM data.

3.1. CoenoSense Extension

CoenoSense is a mobile sensing platform that facilitates the development and evaluation of algorithms working with large scale socio technical systems [36]. An existing version of **CoenoSense** which initially enabled the phones to act as data loggers was available, storing raw data locally from different sensors enabled in the mobile device. This initial version was substantially extended by Schläpfer in [36], with the aim of facilitating the centralized analysis of the acquired data, extracting characteristics from raw data obtained in the mobile phone, and sending it to a server where it could be accesed and subsequently analized.

In the present work, the CoenoSense platform was extended to allow recoding of GSM cell-tower information for subsequent analysis, similarity measurement and topology estimation.

3.1.1. Description

GSM information to be collected by the CoenoSense platform for the current and neighboring cells is:

- Cell ID: is a generally unique number used to identify each *Base Transceiver Station* (BTS) or sector of a BTS within a *Location Area Code* (LAC) if not within a GSM network.
- RSSI: The received signal strength in a cell in *Active Service Unit* (ASU) for GSM networks. ASUs are related to dBm by equation 3.1.

$$RSSI(dBm) = -113 + 2 * ASU \tag{3.1}$$

- LAC: Locarion Area Code, Is a number that represents a set of base stations that are grouped together to optimise signalling.
- Network Type: The current network type of the cell i.e. GPRS

However, as later explained, the useful information basically consists in the RSSI and cell IDs, as the LAC, BER and Network types are always the same due to the characteristics and location where experiments were conducted.

3.2. Aim of Experiment

The aim of this experiment is to collect a data set that allows us to evaluate and characterize the approach of the present work.

The obtained data set should allow, after a proper analysis, answer or get a better understanding of:

- Comparison of distance estimation/representation for different fingerprinting approaches and similarity measures.
- Computational complexity of the whole proposed algorithm.
- Evolution of localization error.
- Effect of the density of access points for topology estimation.
- Effect of the density of Fingerprints for topology estimation.
- Influence of fingerprinting measurement errors on the localization estimation.
- Limitations of the approach.
- How to minimize error in localization estimation.

The better case is to obtain a data set that reflects the normal, day to day, user's activity, to evaluate or get an approximated estimation of the performance under real conditions. The problem when intending to acquire data in the a realistic way is how to avoid influencing the data acquisition, and at the same time being able to obtain useful and analizable data that allows answering the intended questions.

The scope of the present work depends on the quality of the data set that should be obtained after completion of this experiment. A good data set is required in order to evaluate and apply the intended method in a proper manner.

3.2.1. Description & Procedure

A controlled experiment with a proper ground truth is very useful when evaluating a specific characteristic or parameter is desired. However, a localization method or system under no circunstances would be useful if its use is bounded or limited to certain conditions. A localization method is supposed to work in as many as possible scenarios, and in the ideal case, anywhere and anytime.

Three *Android Nexus-One* phones [39] were available, as well as the existing *CoenoSense* platform. The following steps compose the experiment:

- During one week, a planned walk through the same paths was taken once during the day and once during the evening through the city center of Zürich, considered to have a high density of WiFi access points. Influence of non-permanent obstacles is expected, such as people walking, and trams or buses passing by the streets.

- Two zones of the city center of Zürich will be selected to run the planned walks. Figure 3.1 shows the areas where the data acquisition took place.

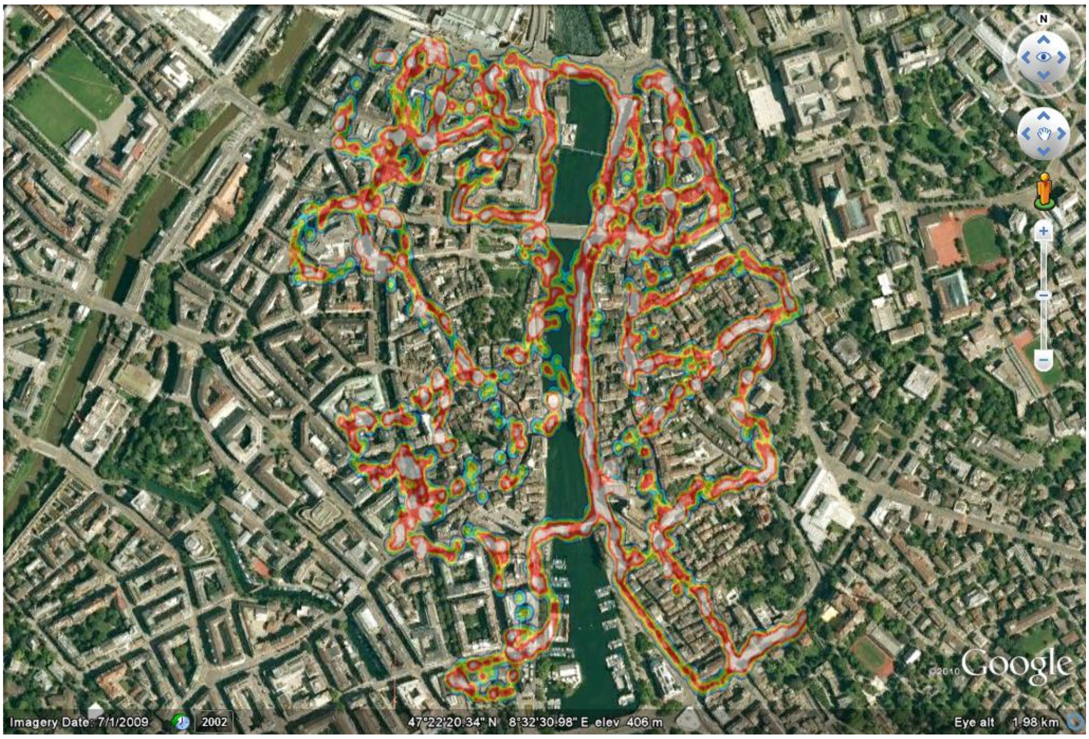


Figure 3.1.: Zones covered during the experiment

3.3. Dataset

Two recordings were taken in the area shown in figure 3.1. In following chapters we will explain that initially one experiment was conducted with a scanning rate of 30 seconds, but we found that in order to be able to test our algorithm, a second recording with a smaller scanning rate of 5 seconds was necessary. The obtained data set allowed to get 1000 fingerprints, which contain GSM, WiFi information, and GPS traces used as ground truth.

To properly make a similarity estimation, in the present part each fingerprint ideally should consist of WiFi information, GSM information, GPS location, time stamp and an user ID. However, there could be locations where for example GPS is not available, or locations with no WiFi signals, or even locations, for example inside a building, where there is no GSM coverage, all of this would result in unaccurate estimations. So, with the aim of getting a better understanding of the similarity behavior of GSM and WiFi information, the recorded and in-server stored fingerprints were filtered by comparing the timestamps of every part of each fingerprint, as each fingerprint has three parts of interest: a WiFi part, a GSM part and a GPS part, each of these parts have a timestamp and an user ID, so a fingerprint is 'built' or 'filtered' by comparing the time stamps of its parts and the user ID, to classify who generated what information and when.

3.4. WiFi Scan

CoenoLogger starts a WiFi scan every 5 seconds and stores the results. The scan returns a list of detected WiFi networks and contains the following information about each of these networks:

- SSID: The network name, not necessarily unique.
- BSSID: The MAC address of the network's access point. This is a unique identifier that should exist only once worldwide.
- Level: The signal level (signal strength) in dBm.

3.5. GSM Scan

CoenoLogger starts a GSM scan every 5 seconds and stores the results. The scan returns the Cell ID, received signal strength, location area code, and network type of the current GSM cell and the neighboring GSM cells:

- Cell ID: generally an unique number used to identify each Base Transceiver Station (BTS) or sector of a BTS within a Location Area Code (LAC), if not within a GSM network.
- RSSI: The Received Signal Strength in a cell in dBm.
- LAC: Locarion Area Code, a number that represents a set of base stations that are grouped together to optimise signalling.
- Network Type: The current network type of the cell i.e. GPRS.

CHAPTER 4

Similarity Measurement

In order to obtain a topology by means of an MDS algorithm, a matrix containing all pairwise distances between the diferent points must be obtained. However, for the present approach, an absolute distance matrix is not built, instead of it, the *points* are represented by fingerprints containing WiFi and GSM information, and the pairwise distances are dissimilarity estimations between each pair of fingerprints. As shown in a previous chapter, several similarity measurements were studied. Basically, two types of similarity measurements were evaluated, those which work for binary samples, and two which consider non-binary samples. It's expected that an estimation based on non-binary samples would provide better results.

4.1. Similarity and Distance Relation

We expected fingerprints recorded in close proximity to each other to be more similar than fingerprints that were recorded further apart. This is a reasonable assumption, because 802.11 WiFi networks have limited range in open space and significantly less range in indoor areas or *Urban Canyons*. GSM signals have a broader range, current cell and the neighboring cells information sent by different mobile stations in a particular moment is expected to be more similar if the mobile stations are close to each other. To review these assumptions, we compared different similarity indices for each pair of fingerprints to their geographical distance, which at the same time were computed from the locations obtained by the GPS measurements taken to serve as ground truth.

- WiFi Fingerprints: contain only WiFi information.
- GSM Fingerprints: contain only GSM information.
- WiFi+GSM information: Containing both WiFi and GSM information.

The plotted results of each similarity measurement applied to the each of the three types of fingerprints described above are presented in Appendix B. The blue points represent the estimated similarities at a certain distance between each pair of fingerprints. The colored curves correspond to different grade polynomial regressions, from one to fourth order, applied to the estimated similarities, this kind of data analysis finds a curve that best fits the behaviour of the points in the figure in a least-square sense.

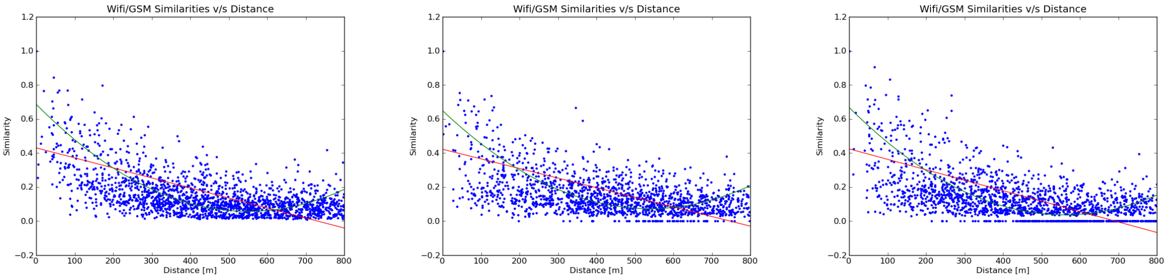
4.1.1. Discussion

According to the results shown in Appendix B, the worst performance is obtained with the First and Second kulzcynski indices. Vectorial approaches show a better performance and lower error. A polynomial modelling shows lower error if made up to second grade, for third and fourth grade the error is larger. The fingerprints consisting on WiFi and GSM information combined show better accuracy, less spreading and a lower error when modeled with a curve of up to second grade, in some cases even third. This behaviour can be observed on the calculated regression coefficients, where the lower grades coeffiecients have more weigth than the higher grades coefficients. It seems interesting that the behaviour in the first 100 meters can be modeled with a rect line or with a parabolic function, both of them presenting very similar accuracy. From all the figures that show similarity evaluations of GSM information, in particular figures B.1c, B.2c, B.3c and B.4c, result of particular interest that through various distances the estimated similarity would seem to be fixed in the same value. This behaviour is probably due to the GSM system itself, because of the broader size of the cellular cells, many fingerprints located inside the same cell would see almost the same cell IDs through long distances, the variation would be present in the received signal strength, it can explain how the non-vectorial approaches show in a stronger manner this behaviour, while the vectorial approaches, by considering the signal strengths can differentiate better the similarity from GSM information. The lower error is present in the vectorial approaches, with a similar performance for the Tanimoto coefficient and Cosine similarity. However, it is important to be aware of a intrinsec error in all the measurements, an error that can not be controlled in an easy way. It mainly consist for example in the accuracy of the GPS coordinates received from satellites, which allow us to obtain the separation distances between each pair of fingerprints. Further, the process of obtaining the separation distance also makes use of the *Haversine Formula* which requires the assumption of the earth as an sphere, which in reality is not true. Despite these intrinsec described errors it is expected that the present results still have consistence and validity.

According to the obtained results, the higher the density of fingerprints the higher the processing complexity, specially for those vectorial approaches. For example, with 500 fingerprints consisting on GSM and WiFi information each one, the amount of operations required is gigantic, it took several hours to our server to process and generate the previously shown results. From now on, it shows up to be compulsory not to recalculate every time the entire matrices, but to update them incrementally, otherwise, the response of a localization system based on this type of approach would be incredibly slow. On the other hand, the error associated to a polynomial modelling of the similarity behaviour seems to be reduced when a high density of fingerprints is used; specially for low order regressions. However, the error obtained in the regressions for each similarity measurement does not mean that a specific measurement is better than another one, it gives an idea about how accurate the current measurement results can be modelled with a polynomial approximation, that means, an idea about the behaviour of the results obtained through the application of a specific similarity. Which similarity measurement is the best one, is still hard to analitically determine out of the present results, as the best similarity would be the one wich provides less dispersion and high differentiability.

4.2. Data Analisys

The previous section showed that the behaviour of our dataset could be modelled with up to second order polynomials with a certain variable error, depending basically on the amount of fingerprints analyzed. However, it could be risky to try to model such data in a deterministic way, as there are many factors that affect the data at the moment it is recorded, factors that in a real situation can not be controlled. Figure 4.1 show the similarity measures obtained from the analysis of the data recorded by three different devices worn at the same time by the same subject. It can be seen that even in this case, the three devices report different records. The reason for the existence of these differences is certainly unknown, but it can be due to differences in the devices' hardware, place and way the devices were carried: for example, inside different pockets etc.



(a) (b) (c)

Figure 4.1.: WiFi+GSM Similarities for three different devices.

All these factors lead to consider the analysis of the similarity behaviour with a probabilistic approach, given a specific similarity between a pair of fingerprints, which is the most probable distance between these pair of fingerprints. In our case, this behaviour can be modelled with a gaussian probability distribution function, that has a different mean and variance for every possible similarity value. Figures 4.2, 4.3, 4.4 and 4.5 show the probability of finding a specific similarity at certain distances, the expected value and variance for different types of similarity measures, and the probability distribution in the form of heatmaps.

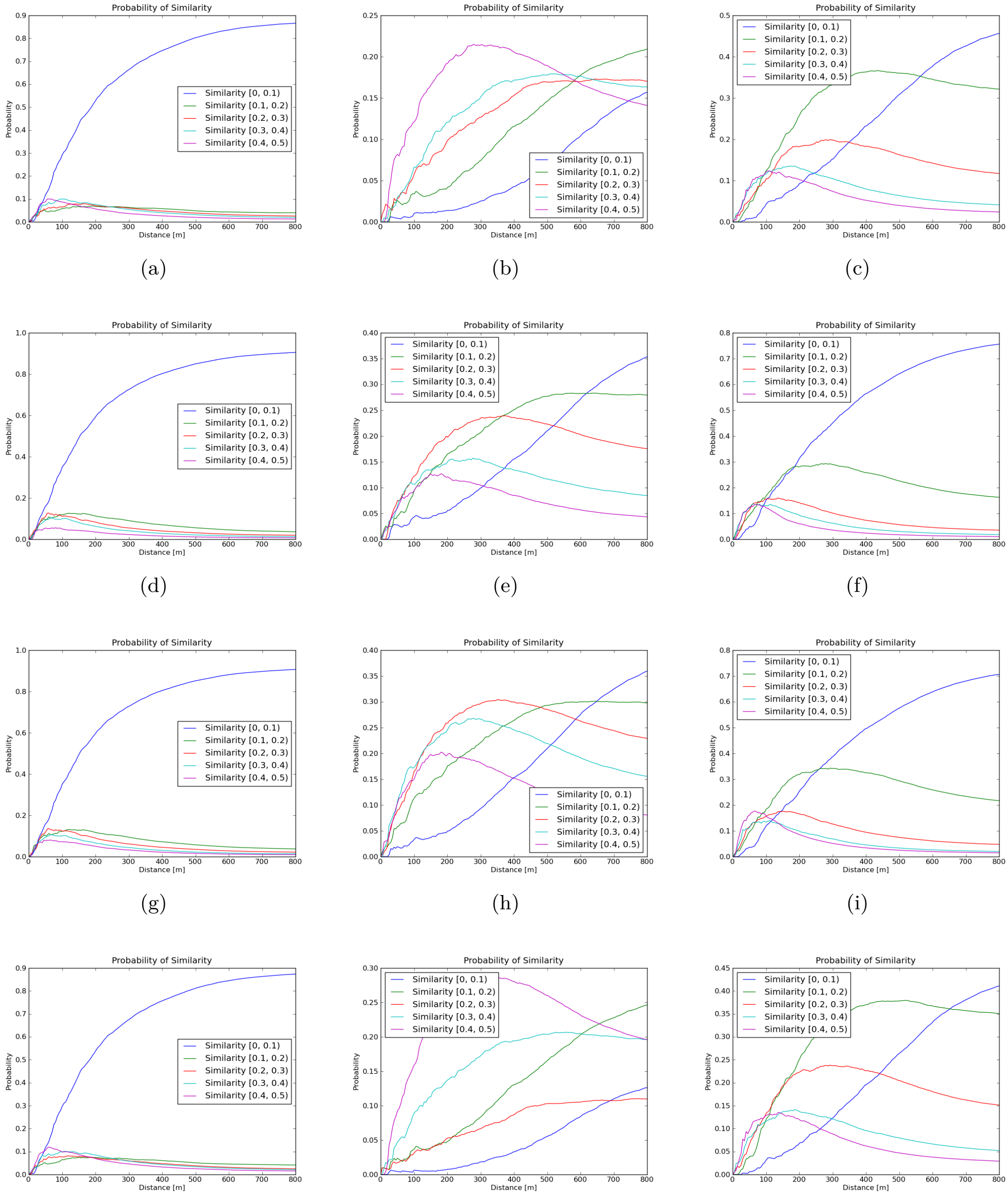
The figures are organized in the following way: the first column corresponds to WiFi fingerprints, the second to GSM fingerprints and the third column to WiFi+GSM fingerprints. The first row correspond to Cosine Similarities, the second to Tanimoto Coefficient, the third to Jaccard Index and the last row corresponds to Sørensen Index. These similarity estimations where choosen among the set of similarity estimations previously introduced due to their acceptable behaviour presented in the last section, where the initial similarity data plots were shown.

Figure 4.2 shows the probability of finding a specific similarity value at certain distances, each curve is a specific similarity range. Low range values are plotted. In figure 4.3, high range values are plotted.

Figure 4.4 shows the expected values of similarity versus distance and the variance. As explained above the first column corresponds to WiFi fingerprints, the second to GSM fingerprints and the third column to WiFi+GSM fingerprints. The first row correspond to Cosine Similarities, the second to Tanimoto Coefficient, the third to Jaccard Index and the last row corresponds to Sørensen Index. It can be seen that WiFi fingerprints present high variance for mid-high similarity values, GSM fingerprints exhibit a constant variance among almost the entire distances, and the expected values are not very differentiable. WiFi+GSM fingerprints present lower variance and better differentiability in the expected values. The following table shows how the figures are presented in the next pages, for example, the element in first row, first column corresponds to a plot containing information for WiFi fingerprints using cosine similarity.

WiFi Cosine	GSM Cosine	WiFi & GSM Cosine
WiFi Tanimoto	GSM Tanimoto	WiFi & GSM Tanimoto
WiFi Sørensen	GSM Sørensen	WiFi & GSM Sørensen
WiFi Jaccard	GSM Jaccard	WiFi & GSM Jaccard

Table 4.1.: Presentation order of the characterization results



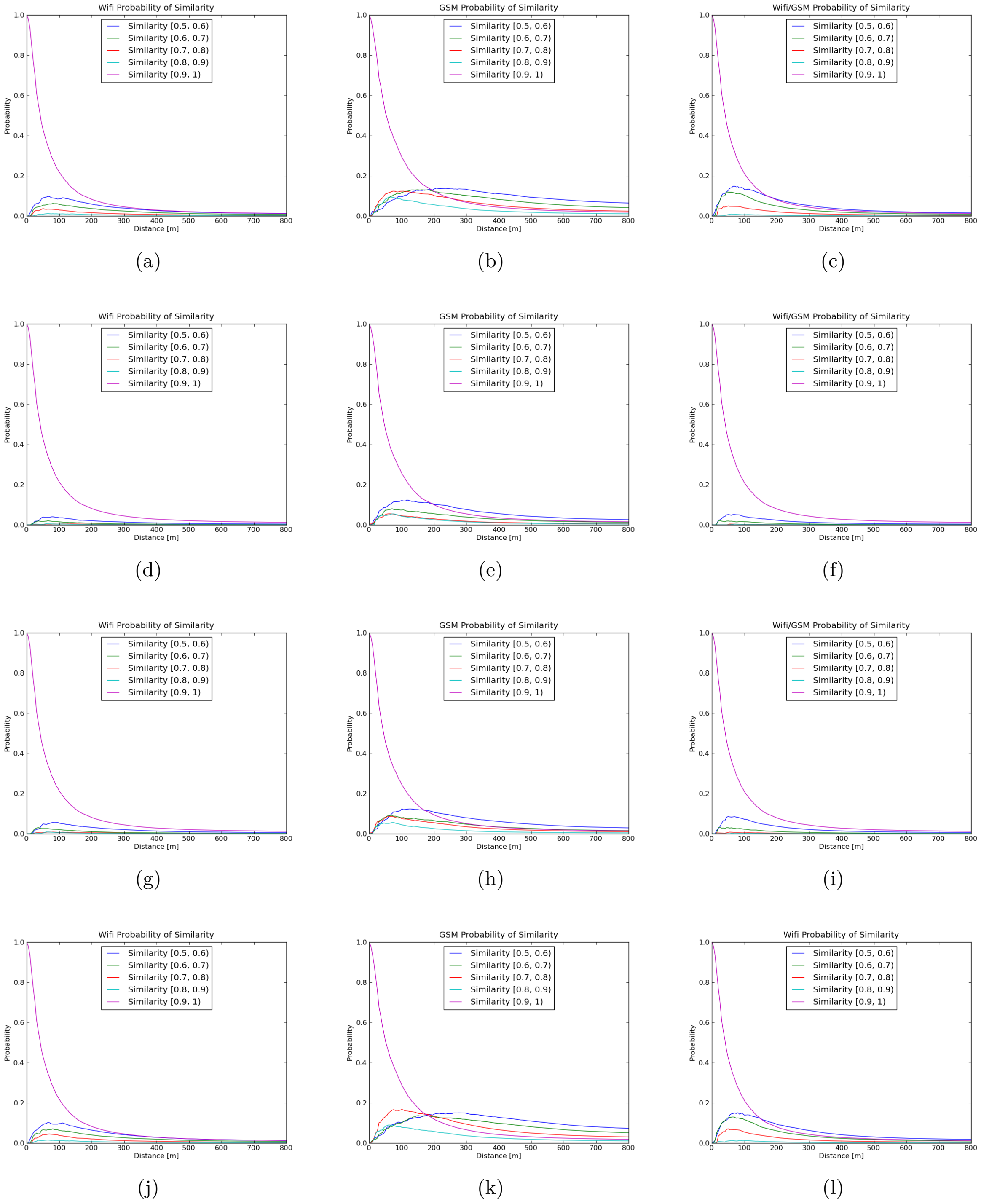
(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

(j) (k) (l)

Figure 4.2.: Probability Behaviour of different Similarity Values for various Indices.



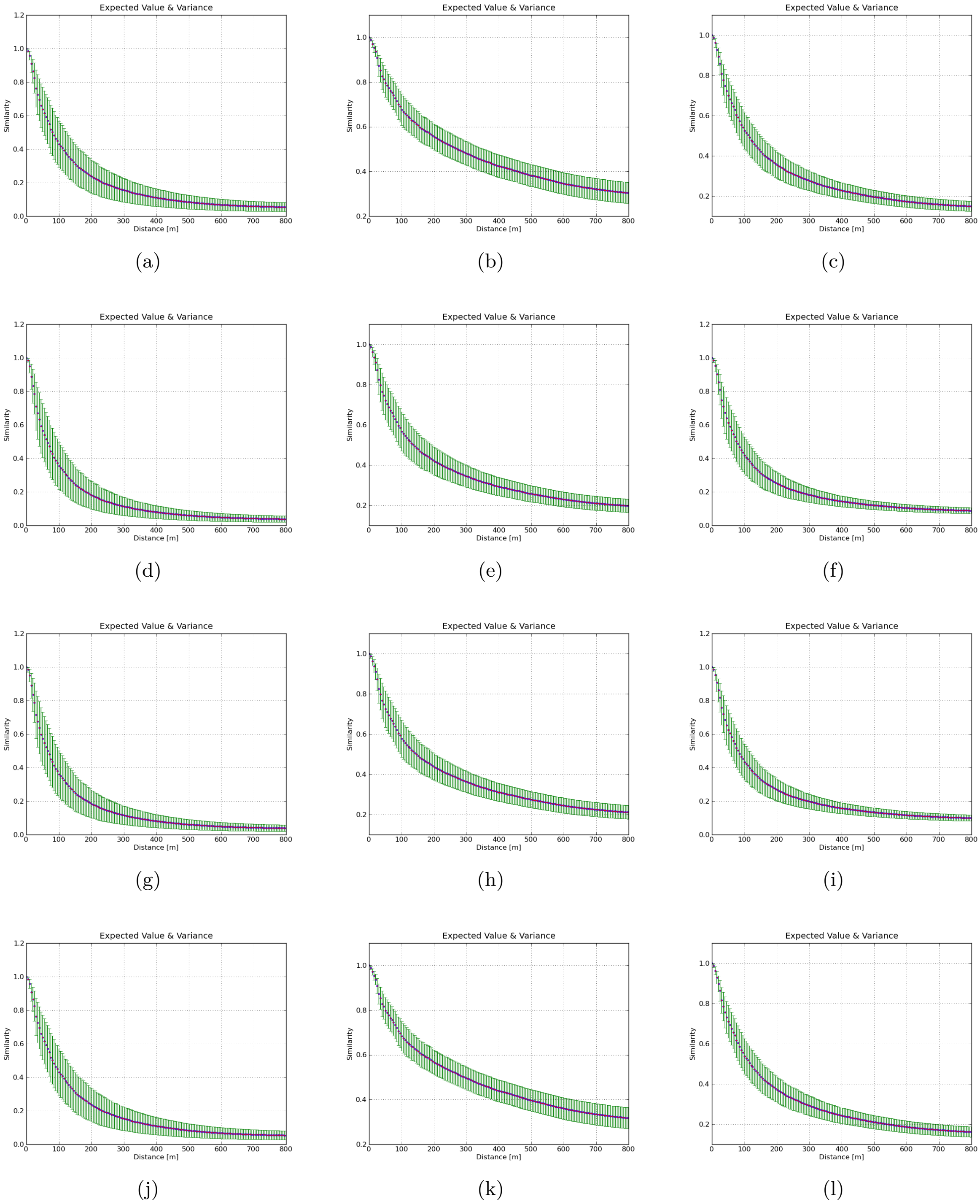
(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

(j) (k) (l)

Figure 4.3.: Probability Behaviour of different Similarity Values for various Indices.



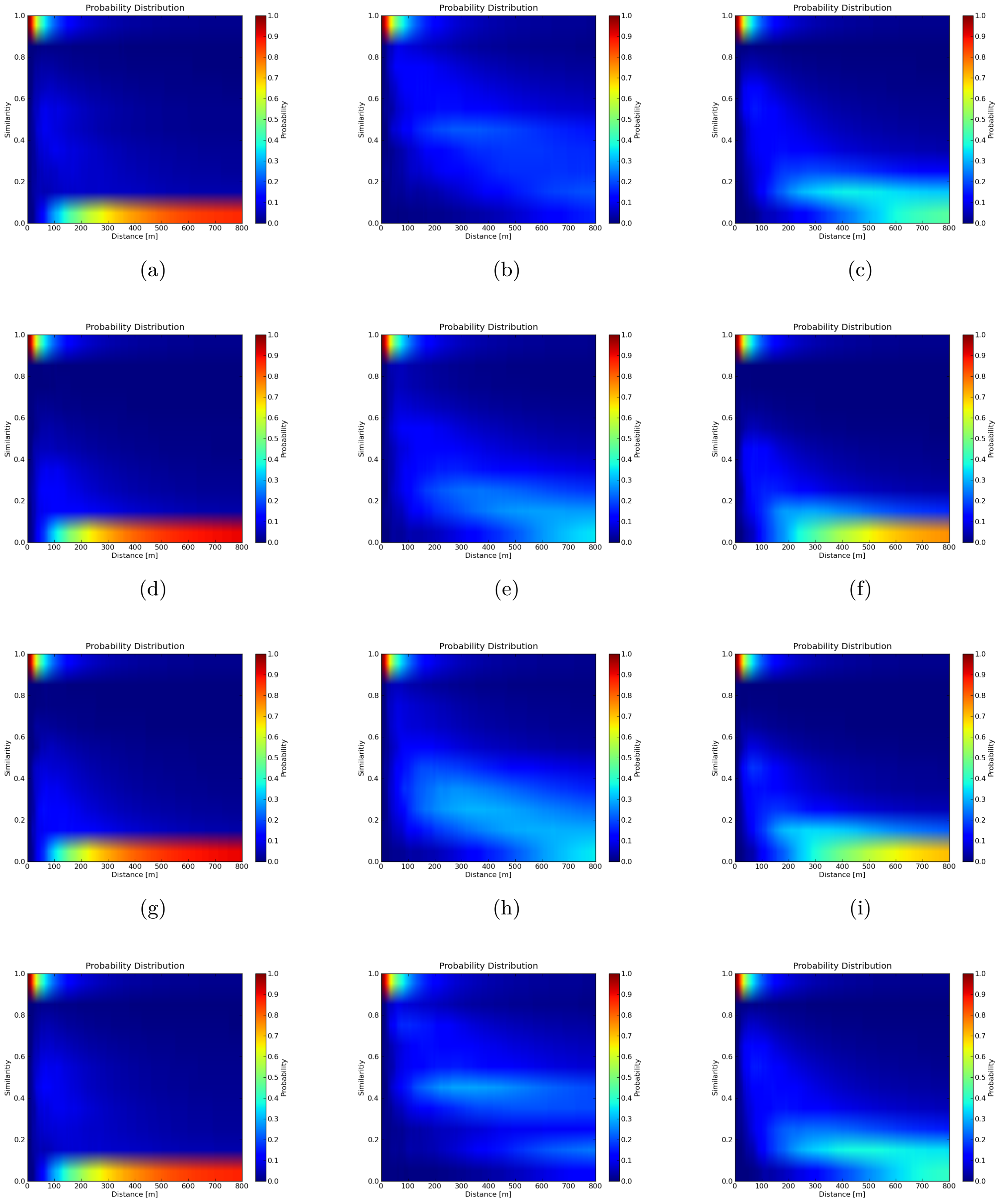
(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

(j) (k) (l)

Figure 4.4.: Expected values and Variance for different Similarity Measures.



(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

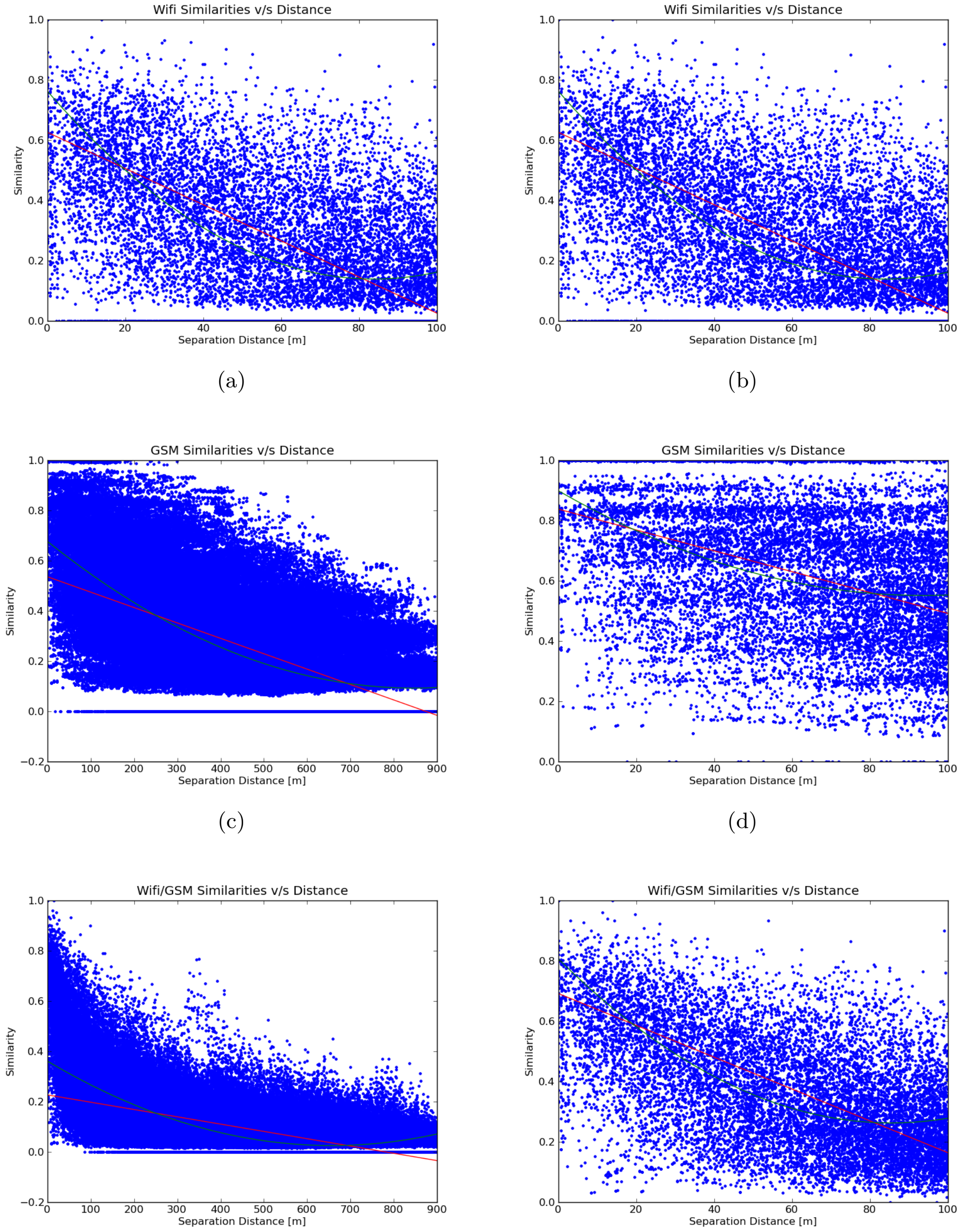
(j) (k) (l)

Figure 4.5.: Probability Distribution for different Similarity Measures.

Figures 4.5 shows how the probability of finding a specific similarity value between two fingerprints is distributed among the distance between them.

From them, it can be seen that the probability distribution is highly concentrated for WiFi fingerprints, it is highly spread or dispersed for GSM fingerprints, and is relatively compensated for WiFi+GSM fingerprints. The aim is to determine which similarity measures provide high differentiability with low dispersion. WiFi fingerprints show high differentiability however this differentiability is found only for high and low similarity values, it is, WiFi fingerprints are expected to be highly differentiable between highly different similarity values, not mid or close values. For example, it results difficult to determine a distance for a specific low similarity. For short ranges, WiFi fingerprints with high similarity values would be easily located and differentiable, however one will have trouble to determine a distance range for mid-high similarity values which are expected for distances between 100-200 meters as shown above. In contrast, WiFi+GSM fingerprints seem to exhibit a good differentiability with acceptable dispersion for high and mid-high ranges. The Tanimoto Coefficient seems to provide the better differentiability and less variance.

Figures 4.6, 4.7, 4.8, 4.9, 4.10, 4.11, 4.12 and 4.13 show the result of the same analysis performed to the second record taken with a scanning window of 5 seconds. We appreciate that due to the increased number of available fingerprints variance gets reduced, and considering the amount of data being analyzed, we assume that the following figures represent more accurate the separation distances through similarities. However, the behaviour for the different types of fingerprints and similarity measurements described above keeps the same.



(a)

(b)

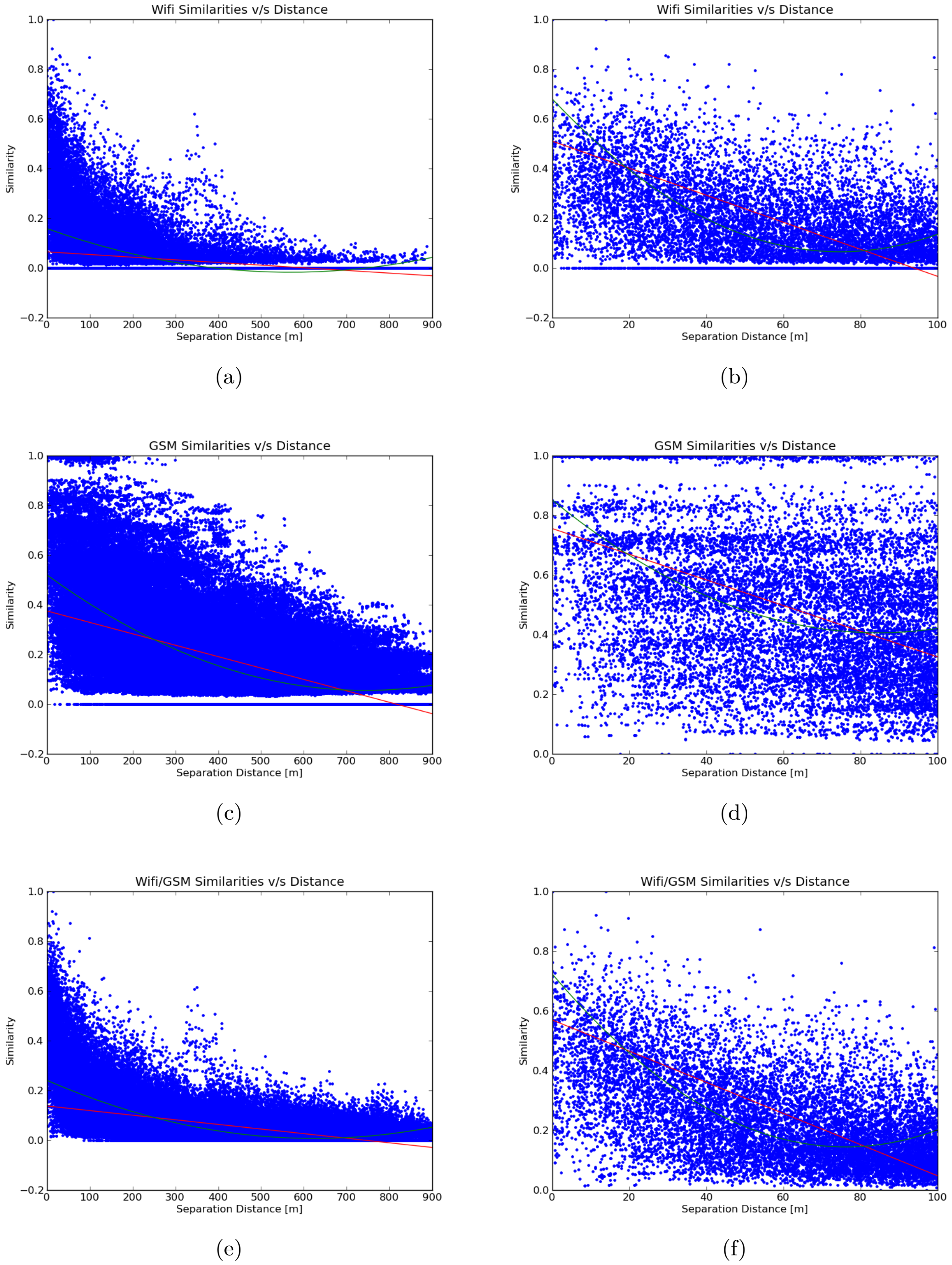
(c)

(d)

(e)

(f)

Figure 4.6.: Cosine Similarity. Coloured curves show different order polynomial regressions. a,b)WiFi, c,d)GSM, e,f)WiFi+GSM. Second recording.



(a)

(b)

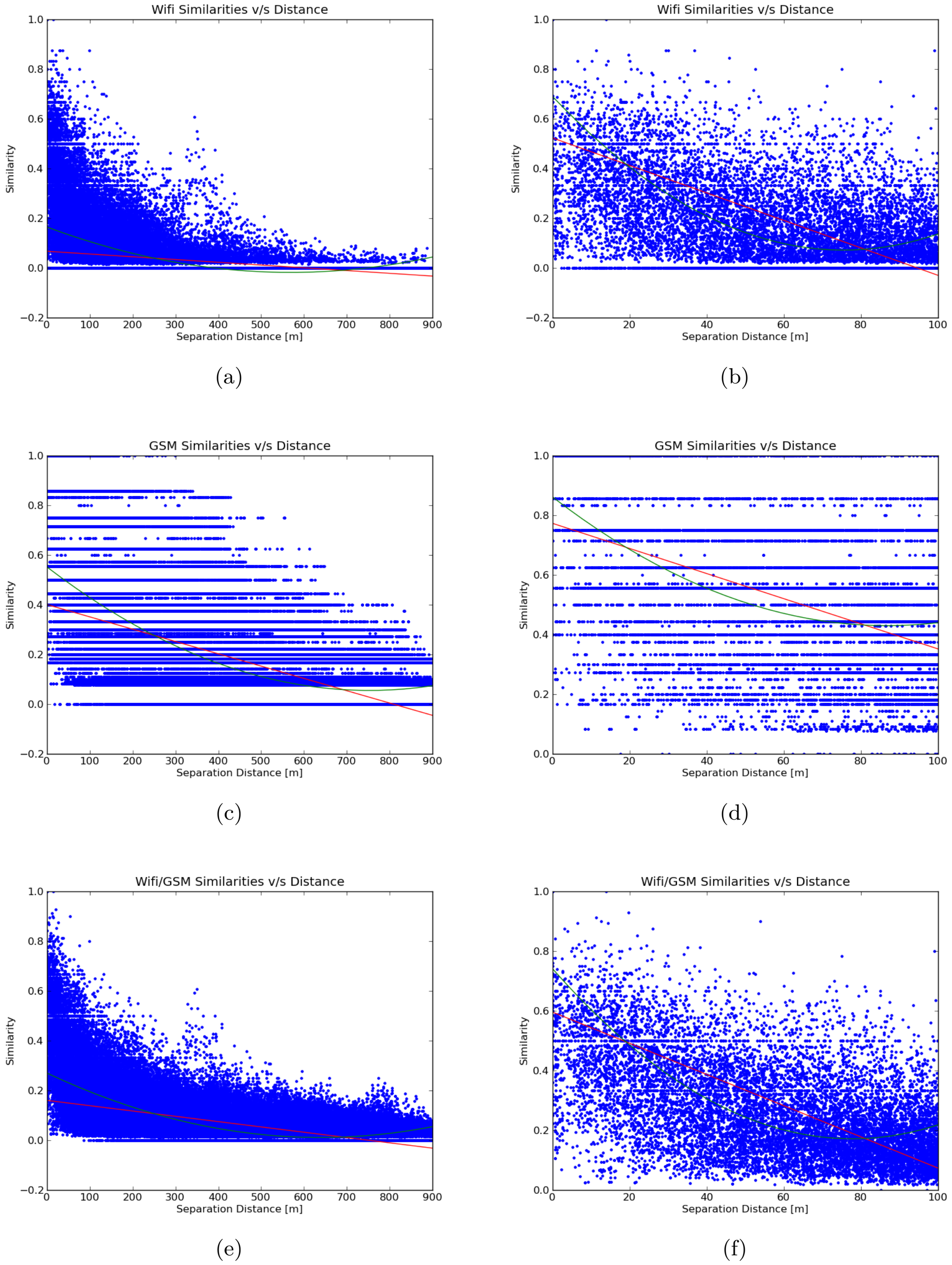
(c)

(d)

(e)

(f)

Figure 4.7.: Similarities using Tanimoto Coefficient. Coloured curves show different order polynomial regressions. a,b)WiFi, c,d)GSM, e,f)WiFi+GSM. Second recording.



(a)

(b)

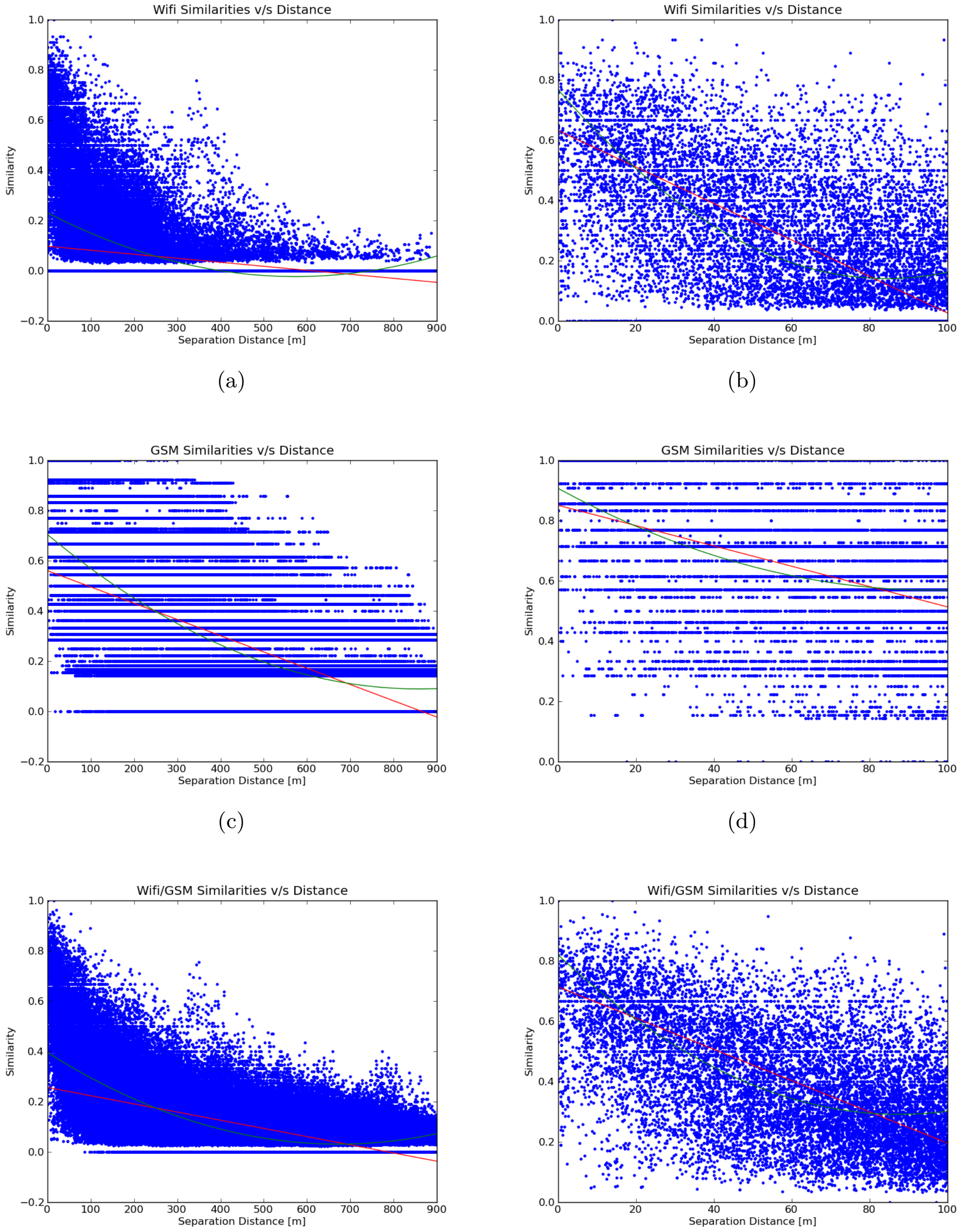
(c)

(d)

(e)

(f)

Figure 4.8.: Similarities using Jaccard Index. Coloured curves show different order polynomial regressions. a,b)WiFi, c,d)GSM, e,f)WiFi+GSM. Second recording.



(a)

(b)

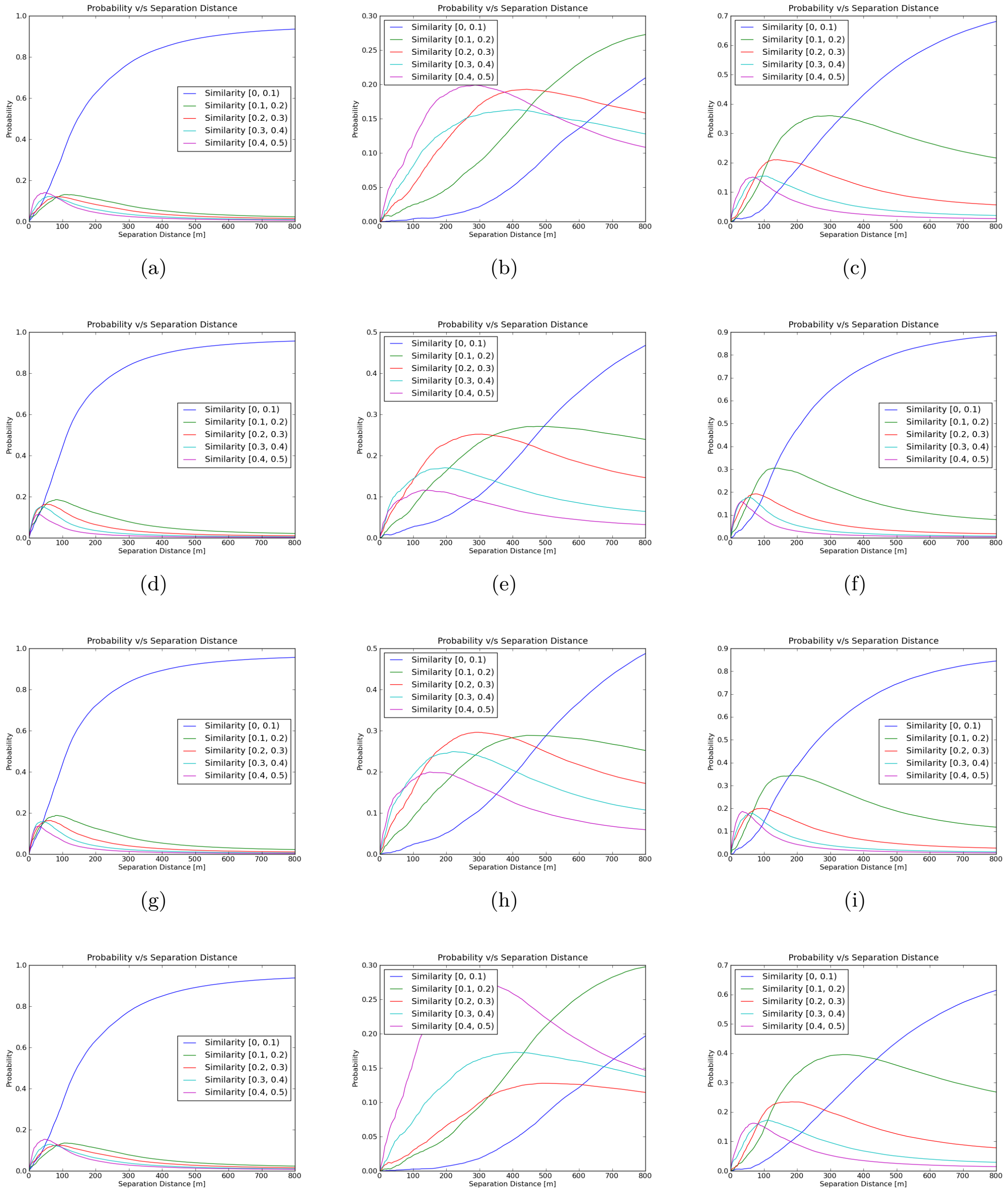
(c)

(d)

(e)

(f)

Figure 4.9.: Similarities using Sørensen Index. Coloured curves show different order polynomial regressions. a,b)WiFi, c,d)GSM, e,f)WiFi+GSM



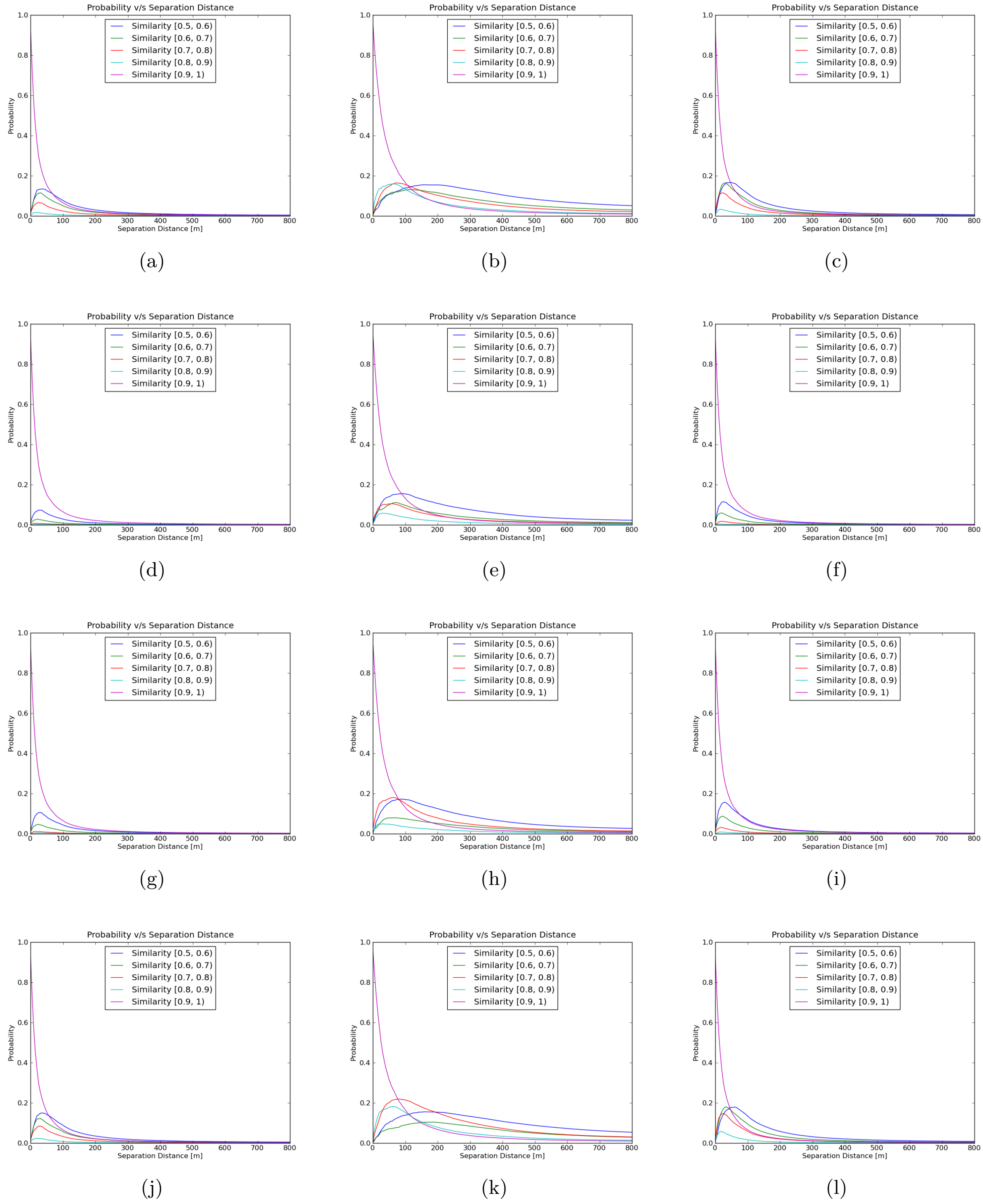
(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

(j) (k) (l)

Figure 4.10.: Probability behaviour of different similarity values, second record.



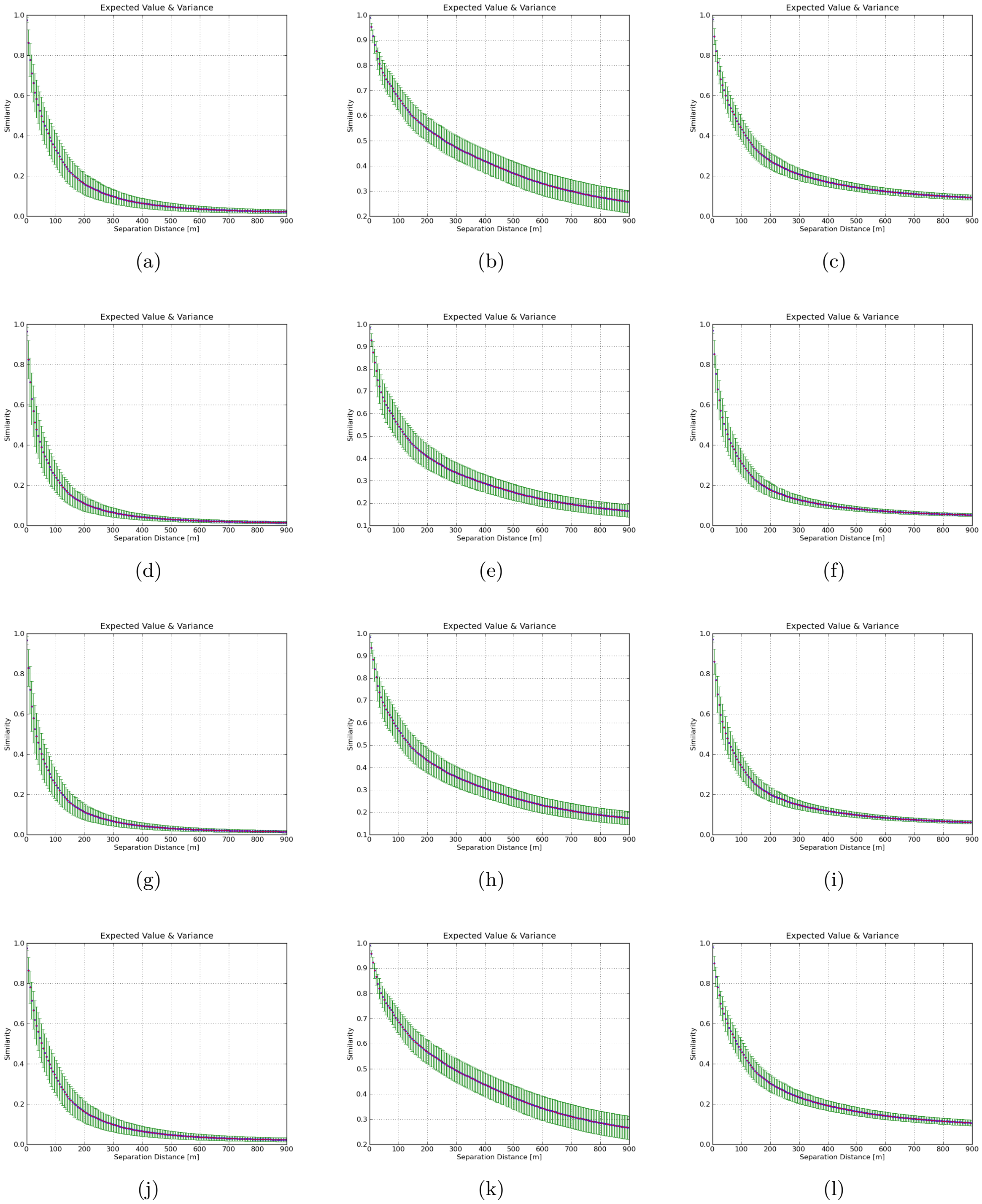
(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

(j) (k) (l)

Figure 4.11.: Probability behaviour of different similarity values, second record.



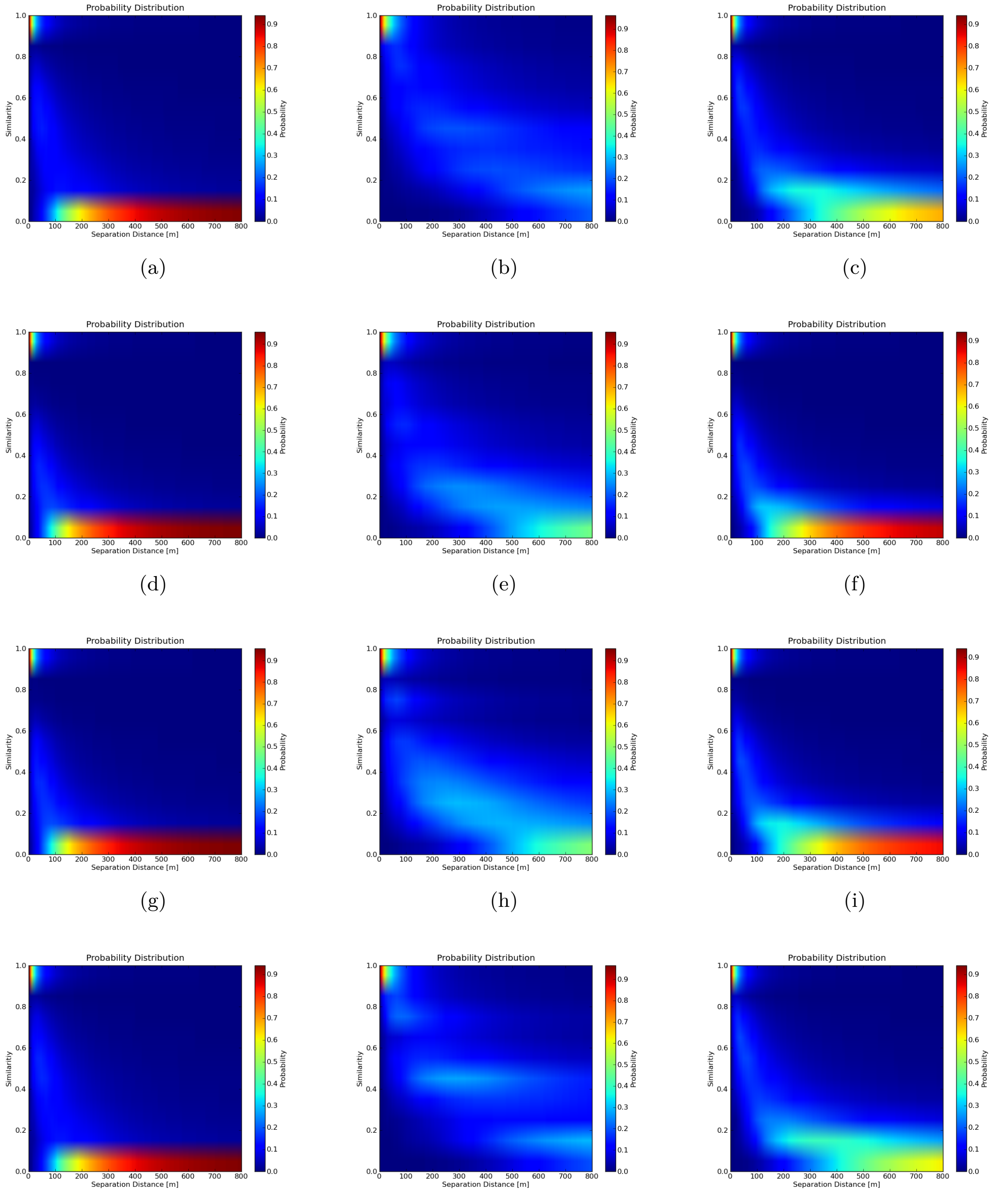
(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

(j) (k) (l)

Figure 4.12.: Expected values and variance for different similarity measures, second record.



(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

(j) (k) (l)

Figure 4.13.: Probability distribution for different similarity measures.

CHAPTER 5

MDS

5.1. Brief Introduction

Multidimensional Scaling is a method that represents measurements of similarity (or dissimilarity) among pairs of objects as distances between points in a low-dimensional multidimensional space. It is a technique for the analysis of similarity or dissimilarity data on a set of objects. MDS attempts to model such data as distances among points in a geometric space.

To give an intuitive idea of what MDS is, figure 5.1 shows schematically an idea of its operation. Suppose that there is a set of n objects, which in our case are fingerprints containing WiFi and GSM information. Assuming that we can estimate the pairwise similarity in this set of fingerprints, we would obtain a matrix S whose element s_{ij} represent the similarity between fingerprint i and j, and in our case all s_{ii} are identical. Now, s_{ij} tell us how "close" is fingerprint i to j. Suppose that somehow we traduce this similarity into a dissimilarity δ_{ij} which would tell us how "far" is fingerprint i from j, mapping all pairwise similarities we would obtain a dissimilarity matrix Δ. We can now look at MDS as a black box, where we introduce our dissimilarity matrix Δ, and it would give us a matrix X containing coordinates in our case in a two dimensional space, for each fingerprint. Now, with these coordinates we can calculate the euclidean distance in this two dimensional space for all pairs of fingerprints, which would be another matrix Δ which elements $\hat{\delta}_{ij}$. The aim of MDS is that each $\hat{\delta}_{ij}$ be as close as possible to δ_{ij}. Ideally, MDS would give us a matrix X such that its pairwise euclidean distance matrix Δ is equal to the initial matrix Δ.

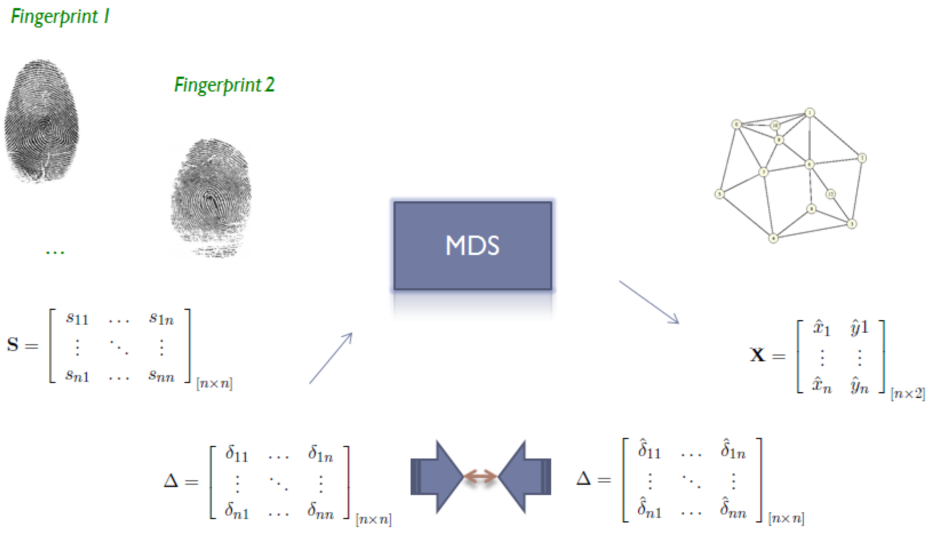

Fingerprint 1
Fingerprint 2
...
MDS
$\mathbf{S} = \begin{bmatrix} s_{11} & \dots & s_{1n} \\ \vdots & \ddots & \vdots \\ s_{n1} & \dots & s_{nn} \end{bmatrix}_{[n\times n]}$
$\mathbf{X} = \begin{bmatrix} \hat{x}_1 & \hat{y}1 \\ \vdots & \vdots \\ \hat{x}_n & \hat{y}_n \end{bmatrix}_{[n\times 2]}$
$\Delta = \begin{bmatrix} \delta_{11} & \dots & \delta_{1n} \\ \vdots & \ddots & \vdots \\ \delta_{n1} & \dots & \delta_{nn} \end{bmatrix}_{[n\times n]}$
$\Delta = \begin{bmatrix} \hat{\delta}_{11} & \dots & \hat{\delta}_{1n} \\ \vdots & \ddots & \vdots \\ \hat{\delta}_{n1} & \dots & \hat{\delta}_{nn} \end{bmatrix}_{[n\times n]}$

Figure 5.1.: MDS scheme

Of course, it was a very intuitive way to understand what MDS does. In practice, some transformations need to be applied to matrix X, such scaling it. In practice, almost never both Δ matrices are equal. The effort consist of finding methods to make them as equal as possible.

There are many MDS methods, most common being the following:

- Metric or Ratio and Ordinal or non-Metric MDS: Metric scaling uses the actual values of the dissimilarities, while nonmetric scaling effectively uses only their rank. Ordinal or non-metric MDS only requires the order of the data to properly reflect the order of the similarities or distances, thus points are located inside zones where they can be located according to the ranking information, because the rank of the distances is not absolute.

  Treating the data as ordinal information only may be sufficient for reconstructing the original map. Ratio and ordinal MDS solutions are almost always very similar in practice. However, there are some instances when an ordinal MDS will yield an undefined solution. The positions of the points in an ordinal MDS are practically just as unique as they are in *metric* MDS, unless one has only very few points. With few points, the solution spaces remain relatively large, allowing for much freedom to position the points [11].

  Ordinal or non-metric MDS is used at all to scale level considerations on data, it is, when the disimilarity information is very relative, dependent or subjective to the way it is measured.

- Classical and Distance Scaling: The main difference between classical MDS and distance scaling, is that classical MDS is performed by a single eigen-decomposition, while distance scaling requires an iterative process in which a MDS result is modified and compared with the ideal dissimilarities or distances between all pair of points, until the result reaches a fitting threshold with the original or ideal map. Distance scaling is performed by optimizing the resulting MDS configuration trying to reduce the cost or stress function until a minimium specified threshold.

Thus, MDS result is a topology: a MDS configuration consisting of a set of points in a space. This configuration can be easier looked by applying transformations, such transformations should be only admissible ones, i.e. those which leave the shape (but not necessarily the size) of a figure unchanged: rotations, translations, reflections, dilations (enlargements or reductions of the entire configuration). [11]

Suppose we have a set of n objects and we are somehow able to estimate the proximity or similarity p_{ij} between object i and j. Lets consider the matrix $\mathbf{X}$ shown in equation 5.1 as the matrix of $n \times m$ size which represents the position of the n objects in a dimensions, so x_{na} represents the position (or coordinate) of object n in dimension a.

$$\mathbf{X} = \begin{bmatrix} x_{11} & \dots & x_{1m} \\ \vdots & \ddots & \vdots \\ x_{n1} & \dots & x_{nm} \end{bmatrix}_{[n \times m]} \tag{5.1}$$

The euclidean distance between any two objects in our representation will be given by equation 5.2:

$$d_{ij} = \sum_{a=1}^{m} \sqrt{x_{ia} - x_{ja}} \tag{5.2}$$

As the information we have is about similarity between pairs of objects, we want to obtain the representation $\mathbf{X}$ such that the distance between any two points matches their dissimilarity as closely as possible. We should then find a way to map proximities into dissimilarities that can be compared with the distances between any two points of a representation. Assuming an error of representation given by equation 5.3:

$$e_{ij}^2 = (d_{ij} - \delta_{ij})^2 \tag{5.3}$$

Being δ_{ij} the mapping of proximity p_{ij} into a dissimilarity. There are many posibilities to do this mapping, from linear mapping which is used in ratio MDS, so $\delta_{ij} = bp_{ij}$, logarithmic, exponential, and more non-linear models, as polynomial mappings, for example a quadratic mapping given by equation 5.4

$$\delta_{ij} = ap_{ij}^2 + bp_{ij} + c \tag{5.4}$$

The interested reader on such mappings is invited to take a look at chapter 9 in [11]. Once the proximities are mapped as disimilarities, summing equation 5.3 over i and j produces equation 5.5 the total error (of approximation) of the MDS representation.

$$\sigma_r(\mathbf{X}) = \sum_{i<j} (d_{ij} - \delta_{ij})^2 \tag{5.5}$$

The relation $i < j$ indicates that it is sufficient to sum over half of the data due to the symmetric nature of the dissimmilarities. In more general cases, some dissimilarity values can not necessary be available, they can be undefined, in that case, the undefined values play no role in any distance in $\mathbf{X}$, so the total error can be written as equation 5.6:

$$\sigma_r\left(\mathbf{X}\right)=\sum_{i<j} w_{ij}\left(d_{ij}-\delta_{ij}\right)^2 \tag{5.6}$$

The weights w_{ij} are generally defined as w_{ij}=1 if δ_{ij} is known and w_{ij}=0 if δ_{ij} is unknown, however other values are also used [11]. The above equation 5.6 is referred as *raw Stress* (Kruskal, 1964b).

For every $\mathbf{X}$, a stress can be computed. There are also many forms of stress functions or 'cost functions' to measure the goodness or badness of fit of a determinated representation. However and according to [11], stress is more a measure of scientific significance. It is important to take into account the degree to which an MDS solution can be brought into a meaningful and replicable correspondence with prior knowledge or with theory about the scaled objects. It is, a particular value of stress can be relatively large, but it does not necessary mean that the representation is a bad one. Sometimes this functions are normalized in order to obtain a bounded value.

It is possible to make himself an idea on what MDS does by looking at the concept of squared distance in some matrix $\mathbf{X}$, the squared distance between any two points i, j would be given by equation 5.7:

$$d_{ij}^2\left(\mathbf{X}\right)=d_{ij}^2=\sum_{a=1}^{m}\left(x_{ia}-x_{ja}\right)^2=\sum_{a=1}^{m}\left(x_{ia}^2+x_{ja}^2-2\left(x_{ia}x_{ja}\right)\right) \tag{5.7}$$

Which in matrices representation would be given by equation 5.8:

$$\begin{aligned}\mathbf{D}^2\left(\mathbf{X}\right)=\begin{bmatrix} d_{11}^2 & \dots & d_{1n}^2 \\ \vdots & \ddots & \vdots \\ d_{n1}^2 & \dots & d_{nn}^2 \end{bmatrix}_{[n\times n]} &= \sum_{a=1}^{m}\begin{bmatrix} x_{1a}^2 & x_{1a}^2 & \dots & x_{1a}^2 \\ x_{2a}^2 & x_{2a}^2 & \dots & x_{2a}^2 \\ \vdots & \vdots & \ddots & \vdots \\ x_{na}^2 & x_{na}^2 & \dots & x_{na}^2 \end{bmatrix}_{[n\times n]} \\ +\sum_{a=1}^{m}\begin{bmatrix} x_{1a}^2 & x_{2a}^2 & \dots & x_{na}^2 \\ x_{1a}^2 & x_{2a}^2 & \dots & x_{na}^2 \\ \vdots & \vdots & \ddots & \vdots \\ x_{1a}^2 & x_{2a}^2 & \dots & x_{na}^2 \end{bmatrix}_{[n\times n]} &-2\sum_{a=1}^{m}\begin{bmatrix} x_{1a}x_{1a} & x_{1a}x_{2a} & \dots & x_{1a}x_{na} \\ x_{2a}x_{1a} & x_{2a}x_{2a} & \dots & x_{2a}x_{na} \\ \vdots & \vdots & \ddots & \vdots \\ x_{na}x_{1a} & x_{na}x_{2a} & \dots & x_{na}x_{na} \end{bmatrix}_{[n\times n]}\end{aligned} \tag{5.8}$$

This leads to equation 5.9:

$$d_{ij}^2\left(\mathbf{X}\right)=\mathbf{c}\mathbf{1}^T+\mathbf{1}\mathbf{c}^T-2\sum_{a=1}^{m}\left(\mathbf{x}_a\mathbf{x}_a^T\right)=\mathbf{c}\mathbf{1}^T+\mathbf{1}\mathbf{c}^T-2\mathbf{X}\mathbf{X}^T \tag{5.9}$$

Where $\mathbf{x}_a$ is column a of matrix $\mathbf{X}$, $\mathbf{1}$ is a $nx1$ vector of ones, and $\mathbf{c}$ is a vector that contains the elements $\sum_{a=1}^{m} x_{ia}^2$, i.e. the diagonal elements of $\mathbf{X}\mathbf{X}^T$. The matrix $\mathbf{B}=\mathbf{X}\mathbf{X}^T$ is called the *scalar product matrix*. So assuming that the matrix $\mathbf{X}$, which is the representation of points, to be unknown, by knowing the squared distances matrix from $\mathbf{X}$ which is in the left side of equation 5.9 one could think that it could be possible to find the matrix $\mathbf{X}$ itself. This is what MDS does, starting from a matrix of quadratic distances or dissimilarities, represented on the left side of equation 5.9, the first two terms on the right are removed by extracting the row and column means, process known as double centering, the constant -2 is removed multiplying by $-1/2$ reducing the matrix of quadratic distances to the matrix $\mathbf{B} = \mathbf{X}\mathbf{X}^T$, from this point matrix $\mathbf{X}$ can be obtained by the process described below.

Now, every nxn matrix can be decomposed into the product of several matrices. There is one useful case called the *eigen-decomposition* which can be constructed for most matrices, but always for symmetric ones, as equation 5.10.

$$\mathbf{A} = \mathbf{Q}\Lambda\mathbf{Q}^T \tag{5.10}$$

With $\mathbf{Q}$ and $\mathbf{Q}^T$ orthonormal, i.e. $\mathbf{Q}\mathbf{Q}^T = \mathbf{Q}^T\mathbf{Q} = I$, being I the identity matrix (matrix with main diagonal full of ones and all other values set to zero) and Λ diagonal. The *eigen-decomposition* is a procedure that can be performed in a computerized way, there exists many methods to do it based on trial and error. At this point, we could suppose that matrix $\mathbf{B} = \mathbf{X}\mathbf{X}^T$ can be eigendecomposed: $\mathbf{B} = \mathbf{X}\mathbf{X}^T = \mathbf{Q}\Lambda\mathbf{Q}^T$, as the scalar product matrices $\mathbf{X}$ and $\mathbf{X}^T$ are symmetric and have non-negative values, one can write equation 5.11:

$$\mathbf{B} = \left(\mathbf{Q}\Lambda^{1/2}\right)\left(\mathbf{Q}\Lambda^{1/2}\right)^T = \mathbf{U}\mathbf{U}^T \tag{5.11}$$

Here we can see that the matrix $\mathbf{U} = \left(\mathbf{Q}\Lambda^{1/2}\right)$ is the matrix that gives the coordinates that reconstruct $\mathbf{B}$. The coordinates in $\mathbf{B}$ differ from those in $\mathbf{X}$ due to the reason that they may be related to different coordinate systems, but it can be solved by rotating the configuration.

On this way, if we refer to 5.7, it can be seen that a configuration of points can be obtained through an eigen-decomposition after relating the similarities to distances, and in fact it is what the classical MDS algorithm does.

Classical MDS algorithm for a set of n points works as follows:

- The matrix of squared distances is computed: $\mathbf{D}^2 = d_{ij}^2\left(\mathbf{X}\right)$.
- Compute the matrix: $\mathbf{J} = \mathbf{I} - \frac{1}{n} \cdot \mathbf{1} \cdot \mathbf{1}^T$.
- Apply double centering to the distance matrix: $\mathbf{B} = -\frac{1}{2}\mathbf{J}\mathbf{D}^2\mathbf{J}$.
- Compute the eigen-decomposition: $\mathbf{B} = \mathbf{Q}\Lambda\mathbf{Q}^T$.
- As the matrix $\Lambda^{1/2}$ is diagonal with decreasing order, and contains the eigenvalues of the eigen-decomposition, according to the theory, the reconstruction is obtained by using the $i \geq 1$ largest eigenvalues. Thus the matrix $\mathbf{U} = \mathbf{Q}_i\Lambda_i^{1/2}$ being i the firs i columns of the matrix $\mathbf{Q}$ represent the coordinate matrix or the representation in i dimensions for the set of points.

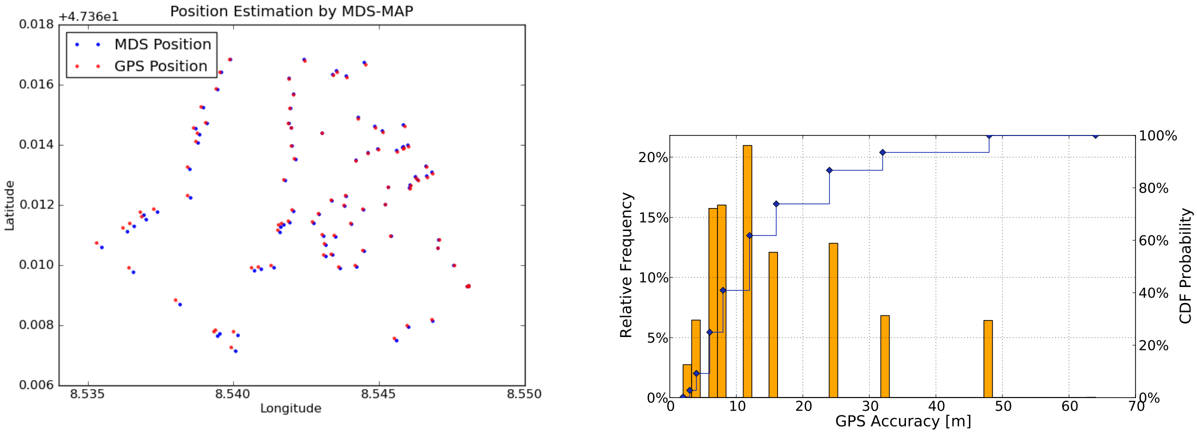
As said before, the resulting coordinate matrix does not necessarily fit the original configuration of points, it is necessary to apply transformations as rotations, translations, reflections and dilations, in order to try to fit to the original topology as close as possible.

5.2. Topology Reconstruction

We are interested in evaluating a localization technique or method without relying on GPS information, except for some anchor points which are required. There are many MDS methods that make use of iterations to reduce the stress, it is, the mean squared error of a configuration matrix such that its interdistances are as equal as possible as the initial distances between all pair of elements. This process is based on comparing the MDS obtained result constantly with the initial distance matrix entered into the MDS algorithm, however the initial distance or dissimilarity matrices that in the present work we use to feed the MDS algorithm are indeed inaccurate, as they are obtained from a similarity estimation that is not necessarily a precise representation of the real separation distance.

Hence, the use of an iterative approach to try to fit the MDS resulting topology to the dissimilarity or distance matrix, would be a computational expensive process that would intend to fit as close as possible the resulting MDS configuration to a final configuration, based on information that has exhibit a random behaviour. For the present work we assume that an iterative stress minimizing approach is still not a good option, as it would require us to rely on GPS information which we consider as ground truth, and that is precisely the information we want to leave aside.

To understand the potential of an MDS-based topology estimation, we first evaluated the accuracy of the approach by using ground truth GPS information as input. Out of the GPS location information we calculated the pairwise spatial distance between all fingerpints and fed them into the MDS algorithm to obtain a topology estimation. The result in Figure 5.2 shows a good matching between the real locations (red) and the estimations (blue).



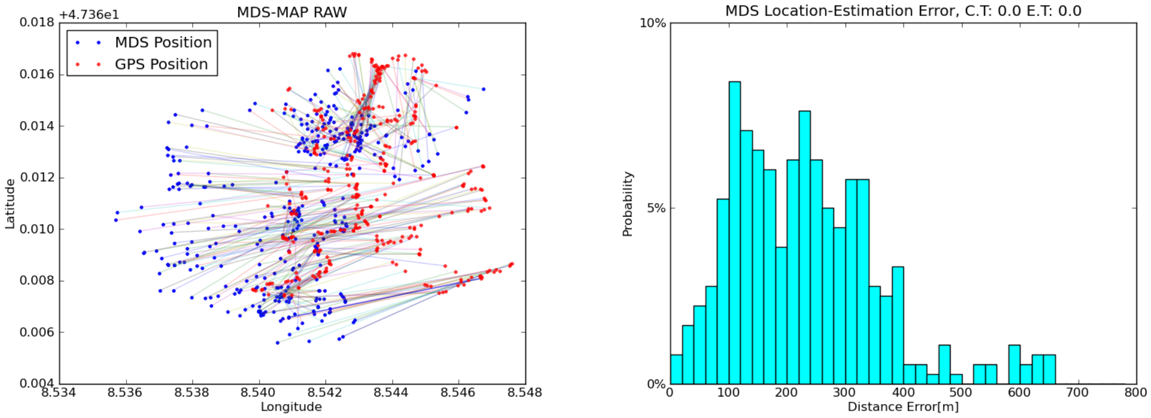
(a)

(b)

Figure 5.2.: Histogram and cumulative distribution function of GPS accuracy over the complete data set. The accuracy value defines the radius of 95% confidence circle.

The error between the resulting topology represented by the blue points and the original GPS information represented by the red points is below 6.58 meters. It means that the accuracy we should expect depends on the precision to estimate the separation distance between each pair of fingerprints, which for the fingerprints we use for localization purposes,means how accurate the dissimilarities represent the real separation distances between fingerprints.

Based on the similarity estimation results, we saw that for high separation distances the estimations are not so differentiable i.e. low similarites are expected to be found in a large set of separation distances while high similarity values are expected to be found in a short set of separation distances, further there are estimations that exhibit a zero similarity. We assume and experienced that a direct use of the whole similarity estimation matrix in the MDS algorithm leads to high resulting error, as the low similarities do not represent in a clear way the real separation distance, and the fact that we need to convert similarities into dissimilarities leads to a matematical problem when a zero similarity value is present. For example: a similarity below 0.1 can be found at any point beyond 200 meters. As the underlying data exhibits some randomness in its link structure, the separation distances are not correctly represented. This randomness leads to a folding effect in the MDS result. Using directly MDS to obtain a topology reconstruction would lead to a result like figure 5.3:



(a) (b)

Figure 5.3.: Raw topology reconstrucion

The problem is how to make a good estimation of the separation distance between fingerprints that are far away. To face this problem, we refer to the previous chapter where we saw that high similarity values are not expected to be found for high separation distances, we assume that high similarities represent with lower error the separation distance between fingerprints, thus based on the initial similarity matrix, we build a dissimilary graph, where each node represents a fingerprint and each edge represents the dissimilarity between two fingerprints. Similarities below a determinated threshold are not transformed into edges, with this process we eliminate those incorrect nodes and links which produce folding in the MDS result. Then a shortest path calculation between all pairs of nodes is performed.

At this point, an issue arises: in order to be able to build a complete graph based on high similarities between fingerprints we require a large set of fingerprints that allow us to link fingerprints that are far away from each other by calculating their shortests paths through the existing set of fingerprints, thus a high scanning rate would be required, should a system based on such an approach intends to be implemented, we noticed this problem after making an inital recording experiment with a 30 seconds scanning rate, which for non-stopping pedestrian speed could represent a 30 meters walked distance, thus a device would report just about 3 fingerprints every 100 meters, and just in case all these three fingerprints pass the filtering process where fingerprints containing inconsistent data are discarded. It leads just to a very short set of available fingerprints to correctly represent their separation distances based on their similarity estimations. Thus a second record was done with 5 seconds scanning rate.

The localization accuracy shows different results based principally on the number of available fingerprints, these fingerprints are obtained out from scans of WiFi, GSM information, thus the number of available fingerprints depends on the scanning rate, because not all scans will be part of a fingerprint, as said in section 3.3. The density of WiFi access points in a determinated zone is not necessarily an indicator of a good or bad estimation, it basically depends on how good the devices detect the available networks. However, a high density of access points is desired if one wants to use WiFi information to estimate long separation distances, because the existence of many access points between two far fingerprints represent more possible links to estimate their separation, considering of course that there are fingerprints in between. WiFi fingerprints exhibit a good performance for short-range localization estimation, as their similarities are good differentiable in short separation distances, for mid and high range localization WiFi information may be used but exhibits problems to estimate long separations. The combination of WiFi/GSM fingerprints reported a good performance for short, mid and high range localization, when a high density of fingerprints is avaliable, when a low density of fingerprints is available the algorithms face the same problem as WiFi fingerprints to correctly estimate separations.

GSM fingerprints are interesting ones. Due to the GSM system characteristics and the GSM cells range they are suitable to estimate long separations even when a small density of fingerprints is available in the considered zone. In fact, when a low density of fingerprints is available they exhibit better behaviour than WiFi and WiFi/GSM fingerprints. The reason that explains the better behaviour of GSM fingerprints over WiFi/GSM fingerprints is that when long distances are considered and a small fingerprints density is available, GSM similarities are better differentiable than WiFi/GSM similarities because WiFi similarities are not good differentiable in long separation distances when mixed with GSM information they mix their behaviour with the GSM data resulting on a reduction of the differentiability that GSM information provides for long separations. However, GSM fingerprints provide no good differentiation for short range separations. The following section presents the mentioned phenomena, its effects and the behaviour of the different type of fingerprints.

5.2.1. Topology Results Phenomena

To ilustrate and get a better understanding of the problems and phenomena that an approach like the studied in the present work faces, the following example is introduced. We take a fingerprint out from our data set and want to locate it supposing it is an unknown fingerprint that arrived to the system. As initial step, an iterative process is performed to locate the closest fingerprints available in our dataset by choosing a specific similarity measure, a minimum threshold (one that we use as minimum similarity value in order to expect good results) is settled, and by continuous iterations we look in our dataset for the closest fingerprints which have a similarity with the fingerprint we want to locate, superior to the settled threshold. As soon the closest fingerprints (more than 2) with highest similarity with the one we want to locate are found, the iterations stop and the set of closest fingerprints is taken as anchor points, either they are the original anchor points or added anchor points through previous MDS reconstructions. Then we add to this set of closeset fingerprints the fingerprint to be located and proceed to build a graph. The first step to build a graph is constructing a new similarity matrix for the set of closest fingerprints and the fingerprint to be located, then based on this similarity matrix a graph is constructed where a minimium threshold is settled, to specify which similarities are going to be mapped into edges by transforming them into dissimilarities or distances either by an inverse transformation of the dissimilarities, or through a distance estimation based on its expected value. With this process we eliminate links and fingerprints that are not reliable for the present localization intent.

The next step is to calculate the shortest paths for our graph and build a matrix of dissimilarites or distances with the shortest paths for each pair of fingerprints. This matrix is passed through the MDS algorithm explained in the first section of this chapter obtaining a matrix of configuration of points whose pairwise euclidean distances are supposed to represent as close as possible the initial dissimilarity or distance matrix. This configuration contains a topology, however, it needs to be fixed using the anchor points. The process of fixing this matrix consists of another iterative process that consists of a series of iterations, the number of iterations is equal to the number of anchor points or close fingerprints encountered.

To fix the encountered MDS topology we count with the locations of the closest fingerprints, so two maps are available: the one for the anchor points and the one encountered through the present MDS process, the idea is to fit the anchor points in MDS map to its known locations which can be considered as another map which from now we will call the anchor points' map. Out from the anchor points or closest fingerprints encountered and filtered through the graph construction one fingerprint is selected to be the center of the MDS map and the anchor points' map, for the following steps the next consideration is taken.

- The anchor points' map is a GPS coordinate map. It is known that the GPS coordinate system has a spherical geometry and the MAP obtained through MDS has a bi-dimensional geometry, however we will assume that the zone in consideration is relatively small compared with the entire surface of the earth, thus we will ignore the earth curvature and the GPS coordinate map will be considered as locally flat.

Then we can proceed to fix the MDS map.

- Both maps are centered at the same point, i.e. one of the anchor points, at this point, as presented in equation 5.12, the center has different coordinates both in the MDS map and in the anchors map.

$$\begin{aligned}
\hat{x}_i &= x_{\text{mds } i} - x_{\text{c}} \\
\hat{y}_i &= y_{\text{mds } i} - y_{\text{c}} \\
\hat{x}_{\text{anchors } i} &= x_{\text{anchors } i} - x_{\text{center}} \\
\hat{y}_{\text{anchors } i} &= y_{\text{anchors } i} - y_{\text{center}}
\end{aligned} \tag{5.12}$$

- There exists the possibility that the MDS map is actually mirrored, thus a second MDS map is considered. We also do not know whether the horizontal axis is the real horizontal axis or the vertical axis is the real vertical axis, however by mirroring either the horizontal axis or the vertical axis the MDS map, the rest of the fit can be obtained through a rotation. We take the mirror MDS map by mirroring in the horizontal axis, the vertical coordinates keep unchanged.
$$\hat{x}_i' = (-1) \cdot \hat{x}_i \tag{5.13}$$

- The MDS map and its mirror are scaled by taking the average ratio between the euclidean distance between each pair of anchor points in the anchors' map and in the MDS map being evaluated i.e. the mirrored or the not mirrored, this scaling represents the mapping from spatial coordinates to geographical coordinates, as it is being scaled to the considered flat GPS coordinate system, thus we can see that this mapping is different for each MDS map being fitted.

- Now both maps are passed through a rotation process that consists on a loop that rotates around their centers both MDS maps (not mirrored and mirrored) previously centered and scaled, from 0 to 2π radians, and finds a rotation angle for the MDS map, which provides the lowest average location error between the position of all anchor points in the anchors' map and in the MDS map. The rotation is performed for the not mirrored and mirrored MDS map, the rotation angles for the not mirrored map and for the mirrored map are generally different. The average location error for all anchor points is estimated by calculating the distance in meters between the location of each anchor point in the anchors' map and in a MDS map. As the MDS map has been previously scaled into a GPS coordinate system, the distances are calculated by using the *Haversine Formula*. Then the average distance between all anchor points in the anchors' map and MDS map is considered as the average error. The rotation process for each point in a MDS map can be done in the following way: First the angle β_i of each point with respect to the positive horizontal axis is calculated by equation 5.14.

$$\beta_i = \begin{cases} \arctan\left(\frac{\hat{y}_i}{x_i}\right) & \text{if } x_i > 0 \text{ and } \hat{y}_i \geq 0 \quad \forall i \in \{1, ..., n\} \\ \pi - \arctan\left(\frac{\hat{y}_i}{-1 \cdot x_i}\right) & \text{if } x_i \leq 0 \text{ and } \hat{y}_i > 0 \quad \forall i \in \{1, ..., n\} \\ \pi + \arctan\left(\frac{-1 \cdot \hat{y}_i}{-1 \cdot x_i}\right) & \text{if } x_i < 0 \text{ and } \hat{y}_i \leq 0 \quad \forall i \in \{1, ..., n\} \\ 2\pi - \arctan\left(\frac{-1 \cdot \hat{y}_i}{x_i}\right) & \text{if } x_i \geq 0 \text{ and } \hat{y}_i < 0 \quad \forall i \in \{1, ..., n\} \end{cases} \tag{5.14}$$

In the previous and next equations, x_i represents the centered and scaled points $\hat{x}_i$ or their mirrors $\hat{x}_i'$ depending on which map is being rotated, the vertical components in the mirrored and not mirrored maps are the same. Subsenquently each point is rotated an angle α by applying equation 5.15.

$$\begin{aligned} x_i &= d_i \cdot \cos(\alpha + \beta_i) \qquad \forall i \in \{1, ..., n\} \\ \hat{y}_i &= d_i \cdot \sin(\alpha + \beta_i) \qquad \forall i \in \{1, ..., n\} \end{aligned} \tag{5.15}$$

Being d_i given by equation 5.16:

$$d_i = \sqrt{x_i^2 + \hat{y}_i^2} \tag{5.16}$$

- Finally the rotated map (mirrored or not mirrored) with the lowest average location error for all anchor points is taken as the fitted MDS map and the initial translation is reverted. At this point, we know that both centers are in the same geographical position, so the translation is reverted with respect to the anchor's point center, as equation 5.17 shows.

$$\begin{aligned} x_i &= x_i + x_{\text{center}} \\ \hat{y}_i &= \hat{y}_i + y_{\text{center}} \\ \hat{x}_{\text{anchors } i} &= x_{\text{anchors } i} + x_{\text{center}} \\ \hat{y}_{\text{anchors } i} &= y_{\text{anchors } i} + y_{\text{center}} \end{aligned} \tag{5.17}$$

The previously mentioned steps describe the fitting process, which is done considering every anchor point as center, and finally the fitted map with lowest average error for the anchor points is taken as the final MDS reconstruction. In case this average error results under a determinated threshold, for example one meter, the currently located fingerprint can be added as a new anchor point in case of a low densitiy of fingerprints and high similarity between the anchor points and the fingerprint being located (conditions that are specified through the minimum thresholds for finding closest fingerprints and adding edges by transforming similarities into dissimilarities in the building graph step), as it would mean that the area being considered is a small one, and the error in the location of the present fingerprint should not be high. A lower error for all anchor points does not necessarily mean that the error in the localization of a fingerprint is low, it just means that the present configuration is the one which best fits the existing information. If a particular fingerprint with high error in its own location is added to the map, it would not necessarily affect the localization of another fingerprint, as it would be discarded as an anchor point in the graph construction steps.

To ilustrate the phenomena of folding and the behaviour of WiFi and GSM information the figures 5.4, 5.5 show the localization of a fingerprint. The full experiment area is considered, the non-desired case of low fingerprint density available and long separation distances are taken to give a better idea of the phenomena. In this scenario we only consider GSM-only and WiFi/GSM information, as according to WiFi-only information there is only one close fingerprint, thus in this scenario we can not locate the desired fingerprint using WiFi-based fingerprints. The similarity measure used in this example was the cosine similarity, similarities are mapped into dissimilarities as its inverse.

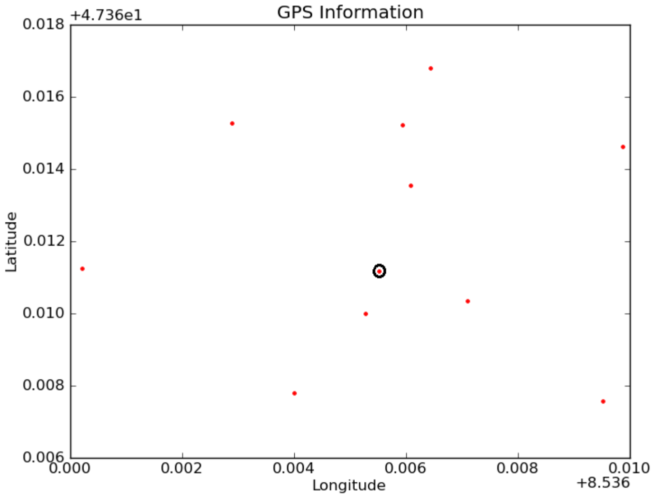
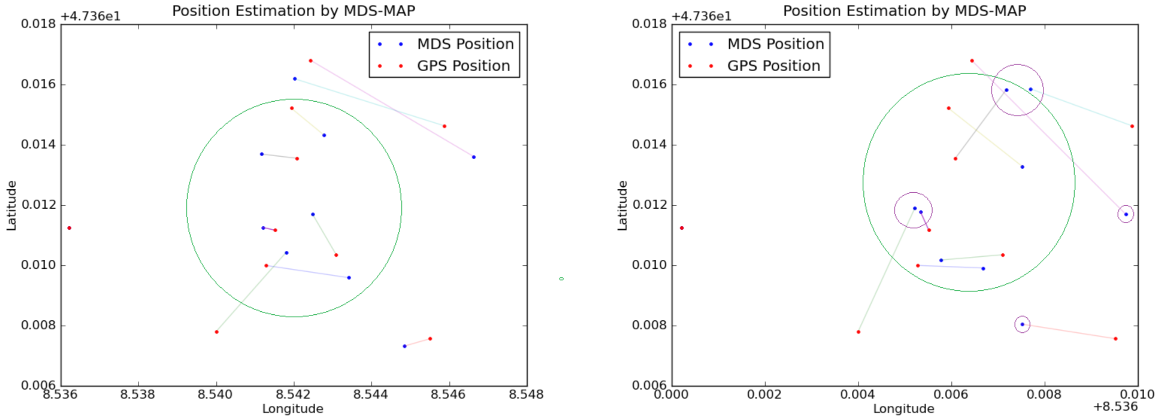


Figure 5.4.: Ground Truth of the fingerprint to be located



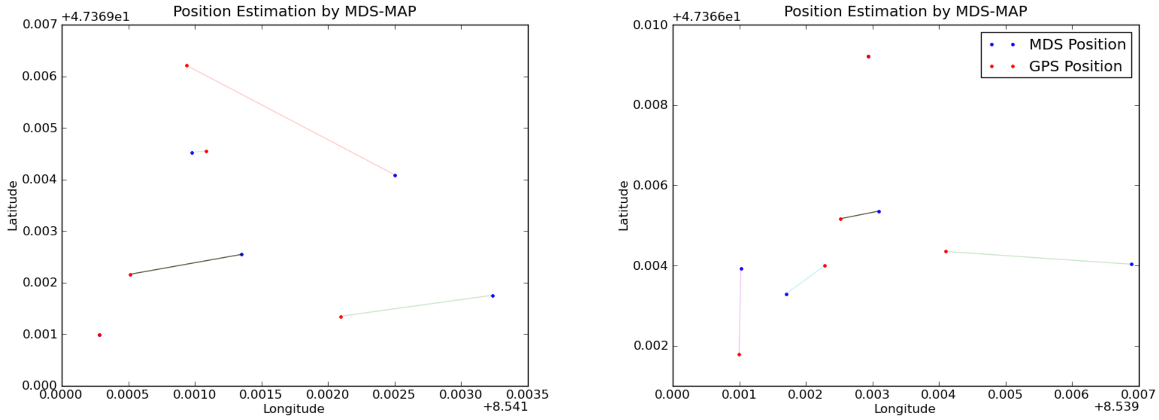
(a)

(b)

Figure 5.5.: Initial Maps: left column GSM Information, right column WiFi/GSM Information.

Figure 5.5 is divided in two columns, the first column corresponds to the mapping using GSM-only information, the second column corresponds to the use of WiFi/GSM information. The red dots represent the position according to GPS information that we consider as ground truth, the blue dots correspond to the estimated locations according to the MDS process. All pair of 'true' and estimated locations (blue and red dots) are connected by lines, the broader line connects the 'true' and estimated locations of the fingerprint we want to locate. The left picture above shows the initial mapping using a standard MDS reconstruction, in this part and for matematical reasons (to avoid by zero divisions) all the close fingerprints that yield a similarity over zero are considered. The fingerprint we want to locate is the one showed in figure 5.4. As we can see, there is a lot of folding in both maps. In the left upper map that corresponds to the reconstruction using GSM-only information, the GSM behaviour can be seen, due to the characteristics of the GSM system and the range of the GSM cells, GSM similarity estimation is unable to differentiate close separations, or separations within the range of a GSM cell. Inside the green circle we have enclose those fingerprints that are more close to the one we want to locate, as it can be seen, they seem to be almost at very similar distance between each other despite geographically it is not like that, it occurs because in separations that are too short in comparison to the range of a GSM cell, all the reported GSM information is very similar, thus the estimated separation distance is also very similar, causing folding effect in the MDS reconstruction. The initial error in this reconstruction is 231 *meters*. Although graphically we can see that the fingerprint we want to locate is actually really close to its real location, in a real situation we do not know anything about the real location of this fingerprint, thus we can only measure distances estimation errors by looking at the error in relation to the anchor points we use.

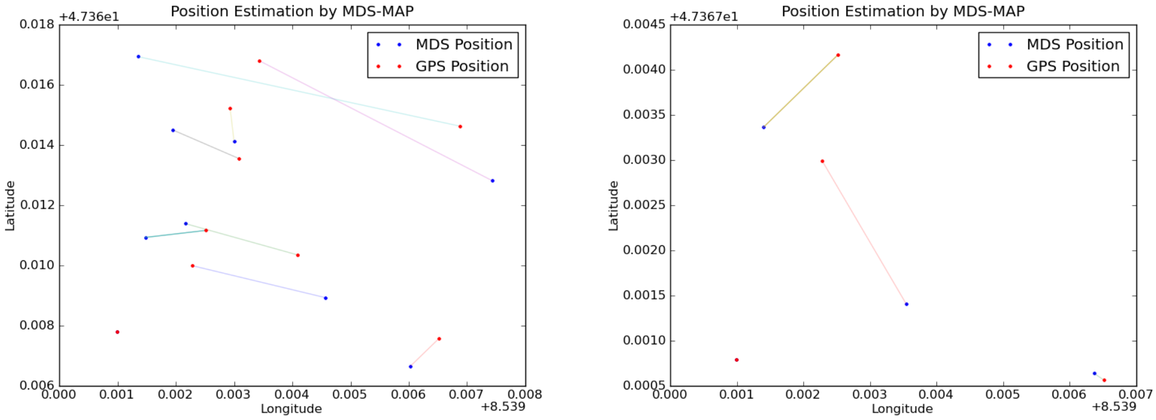
The right side of figure 5.6 shows the reconstruction using WiFi/GSM information. Now the behaviour has changed. Due to WiFi access points range in comparison to the GSM cells size, similarity estimations based on WiFi-only data are highly sensitive to short separation distances, i.e. WiFi information allows a better differentiation between fingerprints close to each other. Even more, the use of received signal strenght values provides a better differentiation. Figure 5.6 shows that the closest fingerprints to the one we want to locate, enclosed by the green circle, are now placed more separated between each other, it is because of the fact that in short ranges WiFi information has more weight in the similarity estimation, as GSM-only data provides very similar values in short range distances. WiFi introduce a differentiation to these very similar values causing a different distance estimation. However, the purple circles show those fingerprints more separated between each other, we can see that they are far away from its real location, it occurs because WiFi provides no differentiation for large separations, thus GSM is the information that actually leads to the similarity estimation, and as said before, it provides very similar values for short separation distances (distances within the range of a GSM cell). There is also further phenomena, when using WiFi/GSM-data a path between two fingerprints is estimated in different ways, when the fingerprints are close enough to each other that exists some WiFi similarity, then WiFi information has more weight in the similarity estimation, however when they are far away and there are no other fingerprints in between, GSM has more weight in the similarity estimation, but if they are not separated enough then from GSM-data one can not differentiate very well the separation. Now, in the case that there are fingerprints in-between the two whose similarity is being estimated, and between those fingerprints some WiFi similarity exists, then WiFi information would have more weight when estimating their similarities. However, the low density of fingerprints causes problems to correctly estimate the similarities, all of these factors lead to the folding effect we see on figure 5.6. The initial error for this estimation was 260 *meters*, it shows under these conditions, despite the unwanted effects GSM provides a better estimation, as the introduction of WiFi information when there is a low density of fingerprints just leads to a bigger folding in the reconstruction.



(a) (b)

Figure 5.6.: Use of Close Fingerprints Threshold: left column GSM Information, right column WiFi/GSM Information.

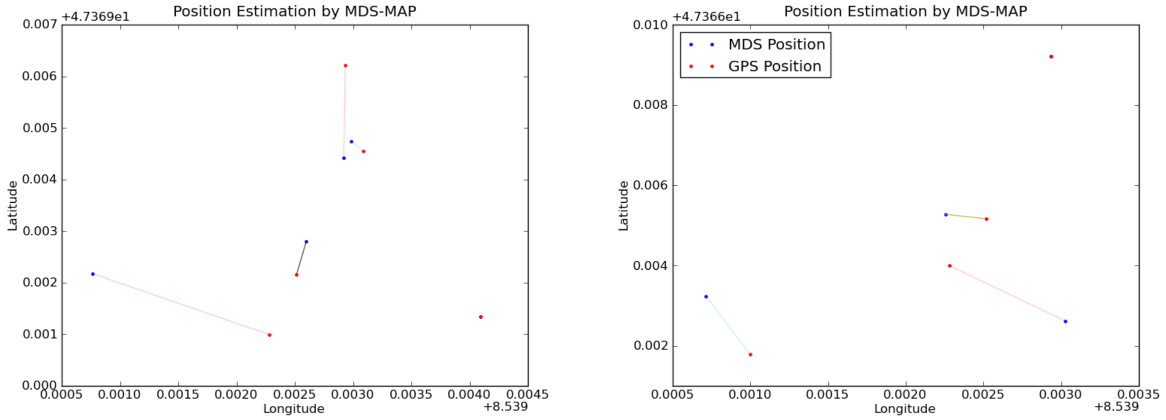
Figure 5.6 shows the effect of incrementing the threshold for detecting close fingerprints, now those far away fingerprints to the one being located are not considered, but as there is no threshold when calculating the shortest paths between fingerprints, then a folding effect is still present. For WiFi/GSM we see the same effect of WiFi information having more weight in the similarity estimation for short separations and GSM having more weight for larger separations, but the same problem of not large enough separations still exists causing problems to differentiate them. For GSM only information the problem is even bigger, as the area being considered is not large enough to differentiate direct similarities between each pair of fingerprints, and no similarity prunning is applied, direct similarities are used as edges. With this procedure the average error for GSM is 108 *meters* and for WiFi/GSM is 160 *meters*



(a) (b)

Figure 5.7.: Use of Edges Threshold: left column GSM Information, right column WiFi/GSM Information.

Figure 5.7 shows the effect of incrementing a threshold for adding edges when building the graph of fingerprints. For WiFi/GSM we now see that some of those fingerprints close to the one we want to locate have been removed, as their similarity with other fingerprints is under the set threshold, it is because WiFi provide a high differentiation for them and thus a low similarity. However for those far fingerprints to the one being located, GSM has the weight in the similarity estimation, then they are reported to be very similar with the one we are locating (as they are not far away enough for GSM purposes), then the still persists, and the initial effect of perceiving them as similarly separated between each other can be seen. This is also caused by insufficient amount of fingerprints close enough to report some good WiFi similarity to the one being located, then the threshold eliminates those who are in the mid-separation affected by low WiFi similarity and keeps those large separated helped by their GSM reported information. The average error in this process is of 217 *meters* for WiFi/GSM information, and 108 *meters* for GSM information. GSM performs very similar as reducing the number of closefingerprints because even though they are geographically distant, it is not large enough separation to report enough difference in GSM information.



(a) (b)

Figure 5.8.: Algorithm Operation: left column GSM Information, right column WiFi/GSM Information.

Figure 5.8 shows the effect of the entire algorithm reducing the number of close fingerprints used as anchors, and removing low similarity edges between fingerprints. Now we see that WiFi/GSM has reconstructed a similar topology, eliminating somehow the folding effect, it is by not considering those fingerprints far away, and considering edges over a threshold, then we see that the folding caused by distant fingerprints reporting similar GSM information is no longer there, only those closest fingerprints are being considered, and as there is an edge removal the estimated dissimilarity between some fingerprints is basically estimated by the sum of the dissimilarites between the fingerprints in between, avoiding GSM folding, but still reporting some folding due to the lack of fingerprints close enough to report good WiFi similarity. Despite the average error is of 112 *meters* we see that the topology is no longer under the extreme folding effects perceived without using the algorithm, and in fact the fingerprint being located is close to its real position. However we do not consider this error because for us the important matter is how good the new map fits with our old map (the one without the new fingerprint that has been located). For the GSM reconstruction, the behaviour has not changed that much, due to the fact already exposed of fingerprints not far away enough to see good differences in their GSM information the average error keeps constant 108 *meters*.

We have now presented the algorithm operation in one of the undesired scenarions, where a low densitiy of fingerprints is present and the separation distances are geographically large, which for our purposes means more than 200 meters. We can also see how the algorithm improves the folding effects and the average error present when a clasical MDS algorithm is used just by building feeding an MDS algorithm with similarities transformed into dissimilarities. The algorithm in this unwanted scenario has shown to improve the average fitting error in about 250 meters, and the final average localization error of around 100 meters in the considered area which size is about 400 meters radius, according to [31] would be still suitable for localization purposes, however it would be on the upper acceptance limit. Despite that in this case we know that the calculated location is in fact really close to the GPS reported location, we just take into account the average error for the selected anchor points, as the real location of a new fingerprint in a live scenario would be unknown.

5.2.2. Algorithm Operation Limitations

Despite we have now seen how the whole approach operates and presented how it reduces the folding effects due to the characteristics of the data being considered and analized, it is important to have in mind that there are some limits that should not be trepassed. There are some limitations ineherent to the algorithm itself that can lead to unaccurate results, the most important of them is how many anchor points which for us are how many close fingerprints we intend to use, and how high should be the threshold for edges removal.

There are two types of answers for the first question, from the mathematical point of view, we need more than one close fingerprints in order to be able to perform the fitting process previously described, however, the matter is not how many is the minimium number of close fingerprints or anchor points we need, but how a certain number leads to better results. From this point of view it would be recommendable to use at least more than 2 close fingerprints or anchor points, it means, finding more than two closest fingerprints preferably close enough to report usable WiFi similarity. In this aspect the computational charge plays an important role, as the more anchor points being considered, the higher the computational effort, due to the iterative processes involved.

The second question is of high importance, as we have found that edge removal must be done in a careful manner, specially when the closest fingerprints detected are actually separated, i.e. closest fingerprints with low similarity had been detected, which is a consecuence of low density of fingerprints in a specific zone. To ilustrate it, figure 5.9 is presented.

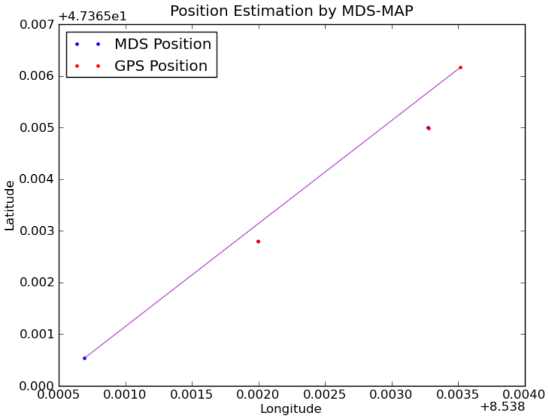


Figure 5.9.: Excessive edge removal effect

It corresponds to the same fingerprint being located previously, but now the edge removal has gone too far, due to the low density of fingerprints and the separation between the fingerprints, we saw that WiFi information has more weight in the closest area, but as the closest fingerprints are not close enough to report good WiFi similarity, the estimation results on a low similarity between the fingerprint being located and the closest one. All other fingerprints have been removed as there are no edges to connect them, however the fingerprint in the low part which is actually the more distant to the one being located is far enough to report no WiFi similarity at all, thus GSM has the influence in this estimation, and as said before, with GSM data they report a higher similarity than the reported for the closest fingerprint, as they are not far away enough to be differentiable.

When the edges removal threshold is setted too high, then the edge to the closest fingerprint is removed and the fingerprint being located seems to be now closer to the distant one than to the really closer fingerprint, it leads to an erroneous localization shown in figure 5.9. Despite the average error for us in this map is really low, the real localization error is high. It would be then recommendable to set a edge removal threshold similar the the closest fingerprints threshold, but not above it. Special care must be taken when a low density of fingerprints is being handled.

We have also found that the density of WiFi access points is not as important as the density of fingerprints, a high density of access points has the potential of providing a better differentiability but if there is not enough density of fingerprints, this potential becomes useless. On the other hand, a high density of fingerprints leads to better estimations and results, due to the fact that we can better estimate separations between fingerprints through the elimination of edges, i.e. it allows the algorithm to set a higher threshold to convert similarities into dissimilarities or separation edges or links.

5.3. Proposed Algorithm

Consider figure 5.10, suppose that we want to locate the blue fingerprint, the red points represent the known map i.e. they are our actual anchor points. As there is a bulding between the fingerprints, very probably fingerprints 1 and 10 will report a low similarity, as the building between blocks the WiFi signals. However, Fingerprint 1 and 2, 2 and 3, and so on will report high similarity values. If we consider all the anchor points to estimate a new map that includes the blue fingerprint, the blue fingerprint that we want to locate will report low similarity with fingerprints in the other side of the building, so if we read these direct similarity estimations, their separation distance would seem to be longer than it really is, causing folding. If we consider all the anchor points, but ignore low direct similarities and consider hops, the blue fingerprint would see the fingerprints on the other side of the building through fingerprints 1, 2, 3, an so on, the summation of these similarities would again estimate longer separation distances than the real geographical separation distances, this wrong estimation of the pairwise separation distances causes folding in the MDS topology result. So, one solution is only to consider the closest vicinitiy to the blue fingerprint, calculate a little submap which can be joined afterwards to the actual known map, updating its size and increasing its coverage. This update should only be done if the estimated locations in the little submap for the subset of anchor points that are in the closest vicinity to the blue fingerprint, do not differ beyond a determinated limit with their actual known locations; in that case the blue fingerprint can be added as a red fingerprint (new anchor point) and its estimated location becomes its known location. That is in short words what our algorithm intends to do, to calculate little maps for each fingerprint that arrives to the system, and then update the actual big map if the addition of a new point does not introduce significant error to the existing big map.

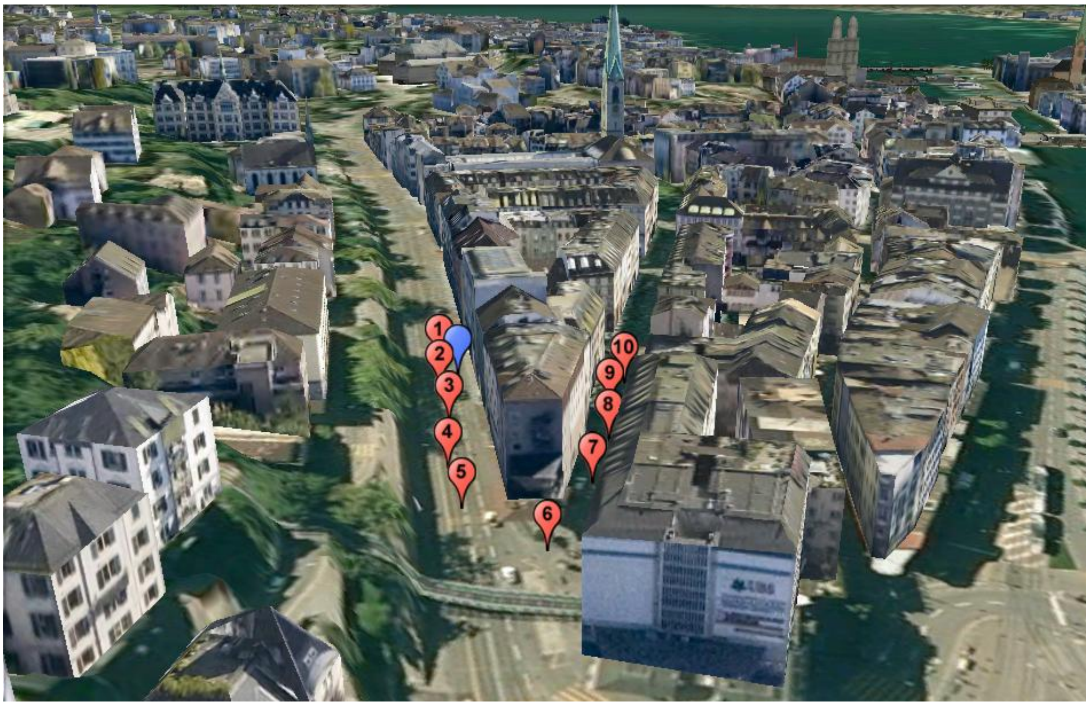


Figure 5.10.: Localization Problem Overview

In the previous subsection we described the operation of the proposed algorithm by introducing a scenario where the folding phenomena was clearly visible. Now we present the proposed algorithm as a diagram in figure 5.11.

Our localization approach consists of three steps: 1) Building a reference topology from a set of training fingerprints. 2) Providing a location estimation for new fingerprints using the reference topology. 3) Including the new fingerprint in the reference topology to refine and extend it.

Figure 5.11 schematically shows the process to build the reference topology (top) and to obtain a location estimation (bottom). To generate a reference topology, WiFi and GSM fingerprints are collected ①. Among all fingerprints, a pairwise similarity measure is calculated ②. Unreliable similarity measures are removed during the pruning process ③. By applying MDS, a topology estimation can be generated ④. Hereby, MDS tries to optimally place the fingerprints into a two-dimensional configuration that retains the similarity relations between fingerprint pairs. Using a minimal set of geo-referenced fingerprints serving as anchor points, a non-linear mapping to geographical locations is determined ⑤.

To obtain a location estimation of a new fingerprint, the same procedure is applied on a subset of the graph. Besides obtaining a location estimation, the fingerprint can also be added to the list of fingerprints of the reference topology which then gets refined and can grow in size.

These steps are described in more detail in the following sections.

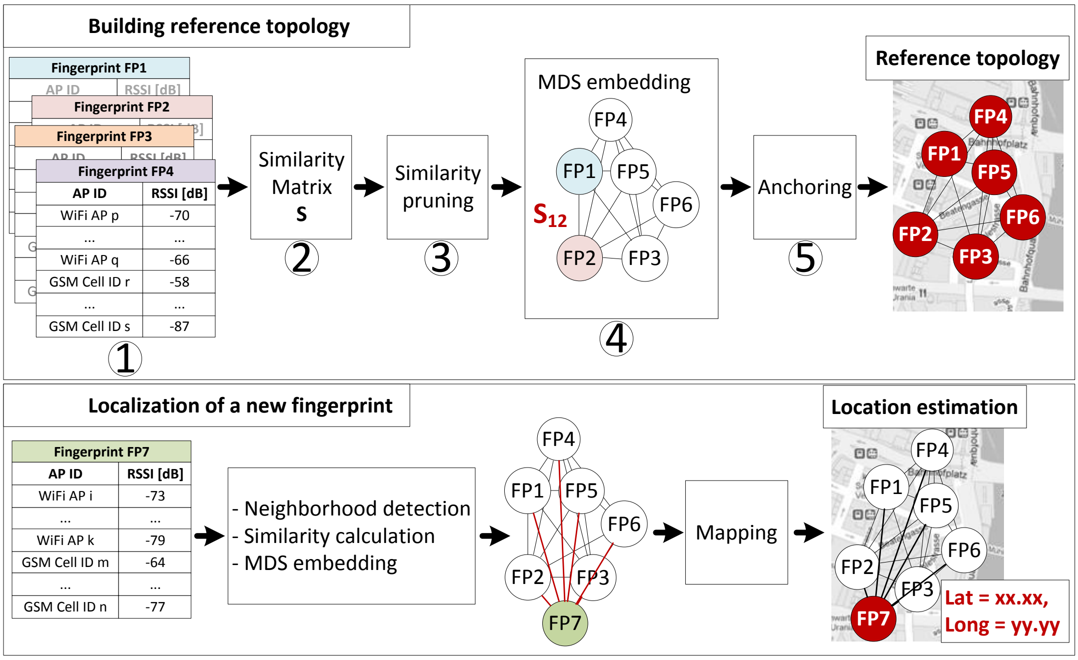


Figure 5.11.: Process to build the reference topology (top) and to obtain a location estimation (bottom).

5.3.1. Generating a reference topology

Collecting fingerprints and generating a similarity matrix A fingerprint contains signal strength readings of detectable APs and base stations referenced by their IDs at a given location. Hereby, the set of fingerprints should ideally have the following properties:

- Each fingerprint should be unique across the space to uniquely reference a geographical location. I.e. if two fingerprints are identical, they stem from the same location.

- For a given location, the fingerprints should not vary over time.

- The similarity between fingerprints should correlate to the distance between their recordings. Close fingerprints should have a higher similarity compared to those far apart.

However, in practice the fingerprints are effected by both multipath and shadow fading. Our approach provides robustness to mitigate their influences.

The last property is of great importance for our method. We used the Tanimoto coefficient [27] as a metric to determine the similarity between two fingerprints. This measure has been used in other works for the comparison of fingerprints [29]. The metric considers each fingerprint as a n-dimensional vector $\vec{F_i}$ with one dimension for each visible access point and the signal strength as the magnitude in the corresponding direction. The Tanimoto coefficient between two fingerprints $\vec{F_1}$ and $\vec{F_2}$ is then calculated according to Equation 5.18.

$$T(\vec{F_1},\vec{F_2}) = \frac{\vec{F_1}\cdot\vec{F_2}}{\left|\left|\vec{F_1}\right|\right|^2 + \left|\left|\vec{F_2}\right|\right|^2 - \vec{F_1}\cdot\vec{F_2}} \tag{5.18}$$

The coefficient is bounded between 0 and 1, with $T(F_1, F_2) = 0$ if F_1 and F_2 have no APs in common and $T(F_1, F_2) = 1$ if F_1 and F_2 being the same fingerprint. The similarity matrix is then given by equation 5.19

$$S = \begin{pmatrix} s_{11} & \cdots & s_{1m} \\ \vdots & \ddots & \vdots \\ s_{n1} & \cdots & s_{nm} \end{pmatrix}; \quad s_{ij} = T\left(F_i, F_j\right) \tag{5.19}$$

Pruning the similarity matrix to increase robustness As we will evaluate in Section 6.1.1, the relation between the Tanimoto similarity measure and the distance between two fingerprints is a monotone, non-linear decaying function. A characteristic of this function is its good discrimination capability within small distances between two fingerprints and its weak performance when the distance is large.

To increase robustness against error introduced by the variance of the similarity estimation from fingerprint-pairs far apart, we disregard similarity values from fingerprint pairs far apart. To do so, we consider our similarity matrix S as a graph G with vertices for each fingerprint and edges for all pairwise similarities. Hereby, we prune the graph by removing all edges with similarity values below a threshold θ. As next step, we find fully connected subgraphs with $n > 3$ nodes. Each subgraph is then fed into the MDS algorithm to obtain an embedding into a two-dimensional space.

Topology estimation using MDS Multi dimensional scaling is a method which represents measurements of dissimilarity among pairs of objects as distances between points in a low dimensional space. Through the analysis of dissimilarities between pairs of objects, MDS estimates a mapping into a geometric configuration in a low dimensional space by trying to keep the pairwise original dissimilarity relations [40]. The MDS method takes dissimilarity values as input. We take each subgraph from the previous step and transform all edges into dissimilarities as follows:

$$\bar{s}_{ij} = 1/s_{ij} \tag{5.20}$$

By calculating all shortest paths, we obtain a dissimilarity matrix $\bar{S}$ for each subgraph which can be fed into MDS. Supposed we have a set of n fingerprints and we are able to estimate the dissimilarity $\bar{s}_{ij}$ between all pairs of fingerprints i and j, MDS finds a configuration represented by a matrix $\mathbf{X}$ of size $n \times m$ where the entries represent the positions of the n fingerprints in m dimensions. So x_{ia} represents the relative position (or coordinate) of fingerprint i in dimension

a. Hence, the output of the MDS method is $\mathbf{X}$ with 5.21

$$\mathbf{X} = \begin{bmatrix} x_{11} & \dots & x_{1m} \\ \vdots & \ddots & \vdots \\ x_{n1} & \dots & x_{nm} \end{bmatrix}_{[n \times m]} \tag{5.21}$$

such that the euclidean distance d_{ij} for any two points is 5.22:

$$d_{ij} = \sum_{a=1}^{m} \sqrt{(x_{ia} - x_{ja})^2} \tag{5.22}$$

As this is an optimization problem and d_{ij} is an estimation, an error of estimation e_{ij} is introduced 5.23

$$e_{ij}^2 = (d_{ij} - \bar{s}_{ij})^2 \tag{5.23}$$

Being $\bar{s}_{ij}$ the dissimilarity, and d_{ij} the euclidean distance in the MDS representation. Averaging e_{ij}^2 over all pairs gives a measure of the error σ_r for the entire MDS representation, called *Raw Stress* [40]. MDS tries to find a configuration $\mathbf{X}$ which minimizes σ_r 5.24.

$$\sigma_r(\mathbf{X}) = \min\left(\sum_{i<j} e_{ij}^2\right) = \min\left(\sum_{i<j} (d_{ij} - \bar{s}_{ij})^2\right) \tag{5.24}$$

An MDS embedding is performed for every subgraph obtained during the pruning process.

Anchoring of the MDS output to geographical coordinates The position of the fingerprints estimated by MDS are relative positions in an arbitrary two-dimensional space. A transformation has to be applied to map the MDS topology to geographical coordinates. Knowing the geographical position of at least three fingerprints included in the MDS topology, such a transformation can be found. Hence, the output of the MDS method is passed through an anchoring process to obtain a transformation into geographical locations. The anchoring process is a regression problem. Our method is comparable to the approach presented in [40]. When all subgraphs have been passed through the anchoring process, a global representation is obtained. We now have a reference topology where all fingerprints are assigned to a geographical location.

Fingerprint localization and updating the reference topology The obtained reference topology can be used for locating new fingerprints, i.e. fingerprints which were not located in the initial topology e.g. coming from new inquiries. To locate such a new fingerprint, the similarity between a new fingerprint and each fingerprint in the current topology is determined. The subset of all fingerprints which yield a similarity value greater than θ with the new fingerprint is selected and the same process as described previously of calculating the dissimilarity matrix, applying MDS and anchoring the map is performed for this subset. The result is a location estimation of the fingerprint. Additionally, this fingerprint can now be included into the reference database and help the topology map to grow. However, we only add the new fingerprint to our reference database if the distortion of the topology is low. To evaluate this, we determine the

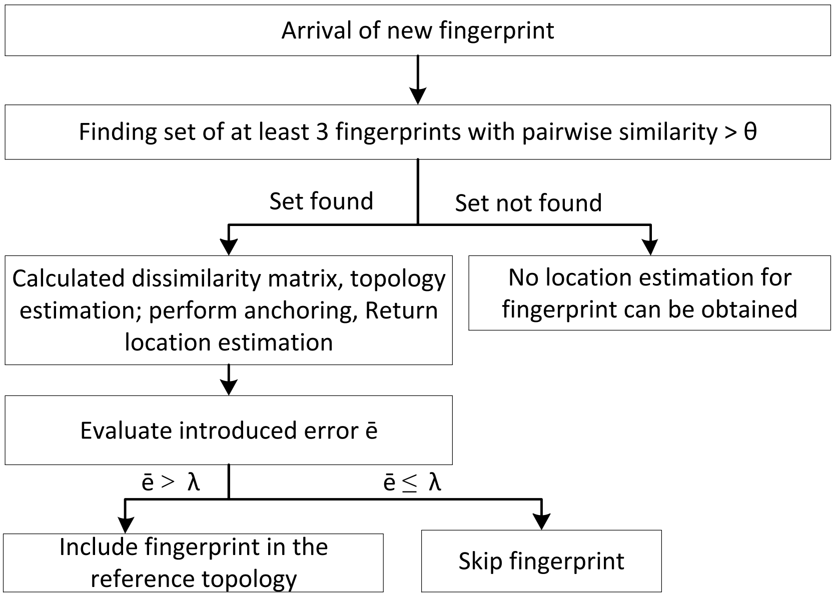

Arrival of new fingerprint
Finding set of at least 3 fingerprints with pairwise similarity > θ
Set found
Set not found
Calculated dissimilarity matrix, topology estimation; perform anchoring, Return location estimation
No location estimation for fingerprint can be obtained
Evaluate introduced error ē
ē > λ
ē ≤ λ
Include fingerprint in the reference topology
Skip fingerprint

Figure 5.12.: Operation scheme to locate a new fingerprint and to consider its inclusion into the reference topology.

fingerprint's influence on the existing topology by calculating the average displacement of the nodes in the subgraph before and after the insertion of the new fingerprint using the *Haversine Formula* [41] with Equation 5.25.

$$\bar{e} = \frac{1}{n}\sum_{i=1}^{n} distance\left(\left[lat_i, long_i\right], \left[lat_{i'}, long_{i'}\right]\right) \tag{5.25}$$

Hereby $[lat_i, long_i]$ and $[lat_{i'}, long_{i'}]$ are the current and proposed locations of fingerprint i in the subgraph, respectively. If $\bar{e} < \lambda$ with λ a given error threshold in meters, the current topology map is updated adding the newly located fingerprint to it. Figure 5.12 summarizes the process.

Figures 5.13, 5.14, 5.15, and 5.16 present in more detail the steps made in the process shown in figure 5.12.

Fingerprints are recorded locally on each mobile device by independent threads, stored in the device's memory as an XML file and then sent to the server. At the server side once this information arrives, it is stored in a database and can be accesed for analysis purposes. However the information that the device sents and that is stored in the server represents the building blocks of a fingerprint, fingerprints must be built up. This building up process is made in two steps, first a filtering process is performed where inconsistent arrived information is discarded eg.: possitive values for received signal strenghts, negative values for cell IDs, or simply empty records. Subsequently WiFi and GSM information is filtered by their time stamps and the user who generated each package, responding in this way to the question: who generated what when?

After building up a consistent fingerprint, process in figure 5.13 is performed.

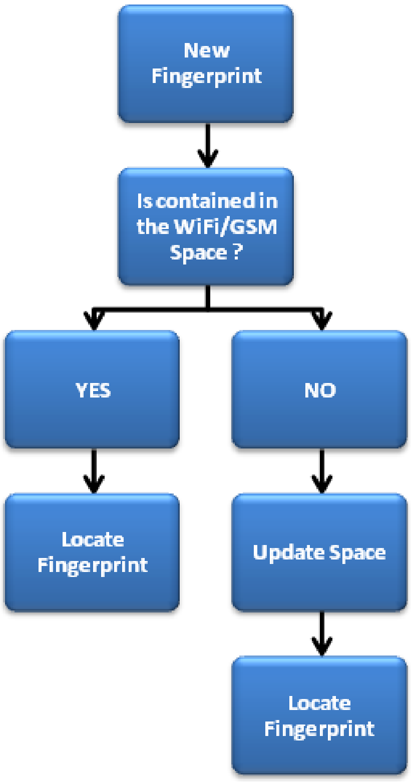


Figure 5.13.: Processing a new Fingeprint

Basically this process evaluates if the fingerprint actually lives in the WiFi/GSM spaces, we previously said that these built spaces are basically the set containing the union of all cell IDs and BSSIDs reported by all fingerprints in the initial map, which is lets say the first initial information whose geographical location is known. If the fingerprint contains a Cell ID or a BSSID not alredy known, the unknown information is added into the respective space. After this process the algorithm starts the localization process.

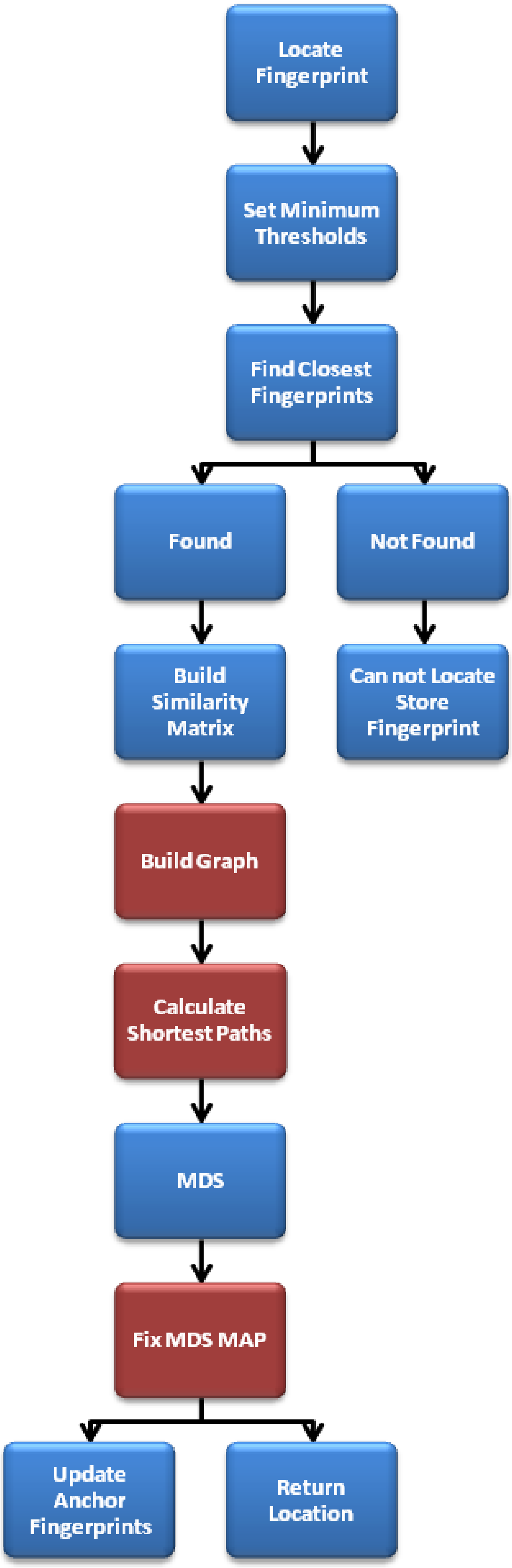


Figure 5.14.: Algorithm operation diagram

When the fingerprint arrives, the minimum thresholds are set, and the closest fingerprints to the fingerprint being located are searched and selected. If no closest fingerprints are found, the fingerprint can not be located

A similarity matrix containing the direct similarity values for the selected set of fingerprints is built, this similarity matrix is then used to build the graph, where direct similarities that are above the threshold are converted into edges, this process eliminates fingerprints that do not report any good connection with the others or those subsets of fingerprints inside the set of closest fingerprints that are isolated by reporting high similarity values between them but have no good link to the fingerprint being located. The fingerprint being located can also be eliminated, but it only occurs when the close fingerprints threshold is higher than the threshold used for prunning edges, as initally a set of closest fingerprints can be found, but if their direct similarities with the fingerprint being located are not above the threshold, all those links are removed, leaving isolated the fingerprint that was intented to be located, if this is the case, the algorithm also returns a reponse that the fingerprint can not be located. Figure 5.15 illustrates this process.

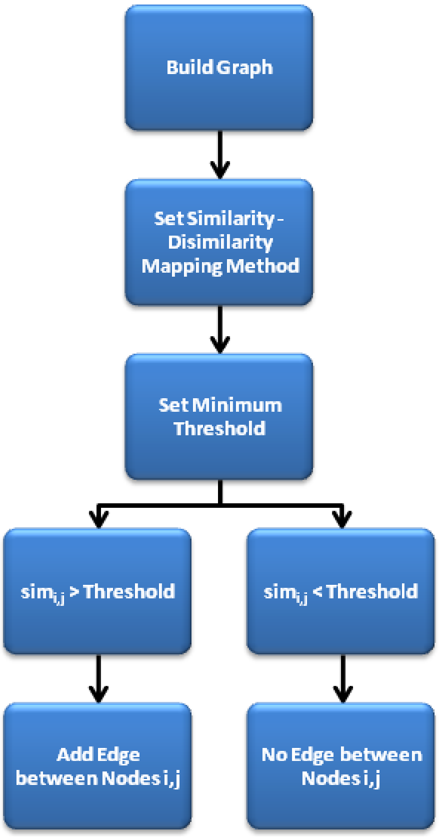

Build Graph
Set Similarity - Disimilarity Mapping Method
Set Minimum Threshold
sim$_{i,j}$ > Threshold
sim$_{i,j}$ < Threshold
Add Edge between Nodes i,j
No Edge between Nodes i,j

Figure 5.15.: Building Graph Process

Then the shortest paths are computed, and the final dissimilarity matrix containing the pairwise distances or dissimilarities between all pairs of closest fingerprints is obtained.

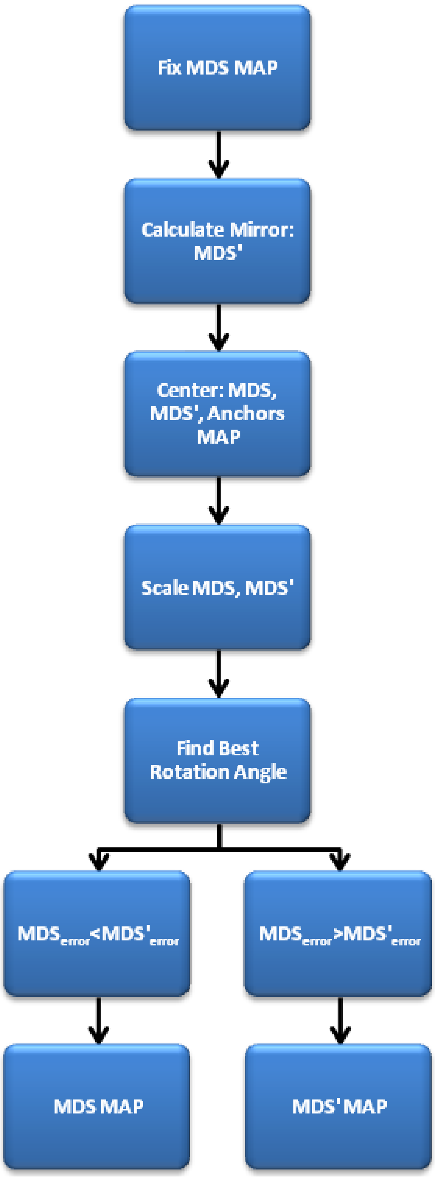


Figure 5.16.: Fixing Map process diagram

5.3.2. Density Considerations

In the last subsections we introduced the algorithm operation and showed its performance for one scenario. We talked about large and short separation distances, about WiFi being dominant in the short range and GSM in the large range, however we have not characterized what we mean about large and short distances. When we refer to separations that allow WiFi differentiability, we are referring mostly to separation distances below 20 or 30 meters, preferring separations up to 10 meters. Why do we do that? If we take a look at the figures in chapter 4, we will see that for all WiFi similarity estimations there is a continuous zero level line beyond 30 meters, that precisely shows the WiFi network access points ranges, and tells that for separations bigger than that, we will have mostly no direct WiFi similarity. Thus, WiFi-based estimations should require a density of about 1 fingerprint every 10 meters to provide good and more or less reliable estimations. It also implies a bigger computational effort, as for a single zone many fingerprints would be needed to estimate a location.

On the other hand GSM figures show that we only start to see zero similarities beyond separations of 250 meters approximately. That would be precisely the range of GSM cells. We can also see constant similarity estimations for a large range of separation distances, which means that GSM does not provide a good differentiation for fingerprints not separated more than 200 meters at least. It means that GSM provides information for large separations, but as showed is not a good option for separation distances in between, as it causes folding in the short range. It also requires a lower density of fingerprints and thus a lower computational effort.

Finally WiFi/GSM is an interesting one, it combines both mentioned characteristics, which under a good density of fingerprints will increment the WiFi influence range and reduce the GSM influence range, requiring a lower densitiy of fingerprints for a localization estimation and thus reducing the computational effort without necessarily reducing the differentiation capability. According to our figures and results, WiFi/GSM shows low similarities beyond 100 meters separation distances, which means that it can reduce the required density of fingerprints and thus the required computational effort (despite the similarity estimation requires a bit more resources) in WiFi-based localization estimations by a factor of 3 or 4, because a smaller set of fingerprints needs to be treated, then a smaller set of iterations needs to be performed.

CHAPTER 6

Results

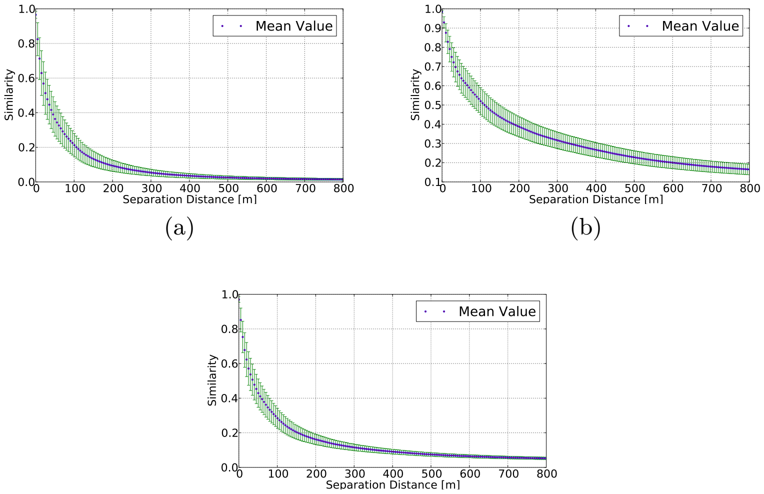
6.1. Simulation

In this section we first evaluate the relationship between the similarity measure and the recording distance between two fingerprints. Afterwards, we investigate the accuracy of our MDS-based topology estimation. Hereby, we investigate the influence of different pruning thresholds θ. For each evaluation step, we compare the approach with fingerprints generated i) with WiFi information, ii) GSM information, and iii) WiFi+GSM information. As our approach is designed for a collaborative system which gradually grows as people use it, we further evaluate the evolution of the location accuracy as new fingerprints are added. We use GPS information as ground truth for the evaluation.

One of the main motivations for the present work has been to get a better understanding of an alternative localization method to GPS. Until this point we have shown and presented an algorithm which by means of MDS can provide localization. To evaluate the algorithm performance a simulation using the recorded dataset was performed. The designed algorithm is intended to serve as a localization method, receiving radio scanning results from lets say, a user wearing a mobile device and running an application who sends a request to know or see its location. The algorithm presented in the last chapter receives a fingerprint, performs its designed process and returns, if available, an estimated location for the received fingerprint as a GPS coordinate. This coordinate can be then sent back to the user which performed the initial request and can easily be displayed in a map on the user's mobile device providing him with a result for his request.

6.1.1. Fingerprint Similarity vs. Distance

We evaluate the relationship between the Tanimoto similarity measure of two fingerprints and the distance between their recordings. To do so, we calculate similarity values between all fingerprint pairs in our data set The relation between similarity measure and distance is illustrated in



(a)

(b)

(c)

Figure 6.1.: Mean and variance valuess for Tanimoto Similarity Measures. a)WiFi, b)GSM, c)WiFi+GSM

Figure 6.1 for the three different fingerprint sets WiFi, GSM and WiFi+GSM. The plots show for each distance value (obtained from GPS information) the mean similarity value together with the variance. The relationship follows a non-linear, monotonic decaying curve. The flattening for smaller similarities or larger distances, respectively, causes an increased error rate in the distance estimation by given similarity due to the non-negligible influence of the variance. For example a similarity below 0.1 can be found at any point beyond $200m$. Thus, no clear discrimination of distances is possible in the low similarity range. This effect is less influential for large similarities or small distances, respectively. By comparing the three relations, the figures show that GSM has the highest variance. WiFi presents a steeper slope in the low distance range than WiFi+GSM which is required for good discrimination. However, WiFi+GSM has a lower variance, providing a larger discrimination range for distances than WiFi and is thus favoured.

6.1.2. Topology Estimation

We are now going to evaluate the localization accuracy of our approach. Table 6.2 to Table 6.4 list the localization accuracy together with additional parameters. Table 6.1 gives a description of the parameters. θ is the pruning parameter as introduced previously, $\widetilde{e}$ is the median localization error in comparison to the GPS ground truth information, α and β are the 25% and 75% error quantiles in meters, respectively. For each threshold, we rerun the localization process 100 times with random starting configurations. σ is the variance of the median error for these iterations, δ represents the percentage of fingerprints out of the data set that were localized (and hence not pruned), ρ represents the percentage of fingerprints out of the data set that were used as anchor points.

Parameter	Description
θ	Pruning parameter, $0 \leq \theta \leq 1$
$\widetilde{e}$	median localization error [m]
α	25% error quantiles [m]
β	75% error quantiles [m]
σ	Variance of the median error
δ	ratio of localized fingerprints [%]
ρ	ratio of anchor points [%]

Table 6.1.: Overview of evaluated parameter

θ	$\widetilde{e}\,[m]$	$\alpha\,[m]$	$\beta\,[m]$	σ	$\delta\,[\%]$	$\rho\,[\%]$
0.6	17	9	31	0.6	9	2.9
0.5	42	18	78	0.4	23	3.6
0.4	33	13	54	0.7	45	8.1
0.3	88	28	175	0.4	72	8.8
0.2	377	190	390	0.6	92	2.2
0.1	383	256	525	0.8	96	0.7
0.0	431	324	615	0.6	100	0.3

Table 6.2.: Summary of the algorithm performance for different thresholds θ using WiFi-based fingerprints.

θ	$\widetilde{e}\,[m]$	$\alpha\,[m]$	$\beta\,[m]$	σ	$\delta\,[\%]$	$\rho\,[\%]$
0.6	286	161	708	0.16	94	1.8
0.5	201	130	309	0.06	98	0.3
0.4	273	160	397	0.08	99	0.3
0.3	300	191	605	0.12	99	0.3
0.2	466	259	859	0.11	100	0.3
0.1	577	389	733	0.09	100	0.3
0.0	640	483	893	0.13	100	0.3

Table 6.3.: Summary of the algorithm performance for different thresholds θ using GSM-based fingerprints.

θ	$\widetilde{e}\,[m]$	$\alpha\,[m]$	$\beta\,[m]$	σ	$\delta\,[\%]$	$\rho\,[\%]$
0.6	26	10	44	1.6	14	3.6
0.5	**30**	**14**	**57**	**1.4**	**36**	**7.3**
0.4	56	18	114	1.7	70	10.6
0.3	201	84	316	1.4	95	5.5
0.2	366	211	511	1.6	99	0.3
0.1	264	158	396	1.8	100	0.3
0.0	574	457	640	1.5	100	0.3

Table 6.4.: Summary of the algorithm performance for different thresholds θ using WiFi+GSM-based fingerprints.

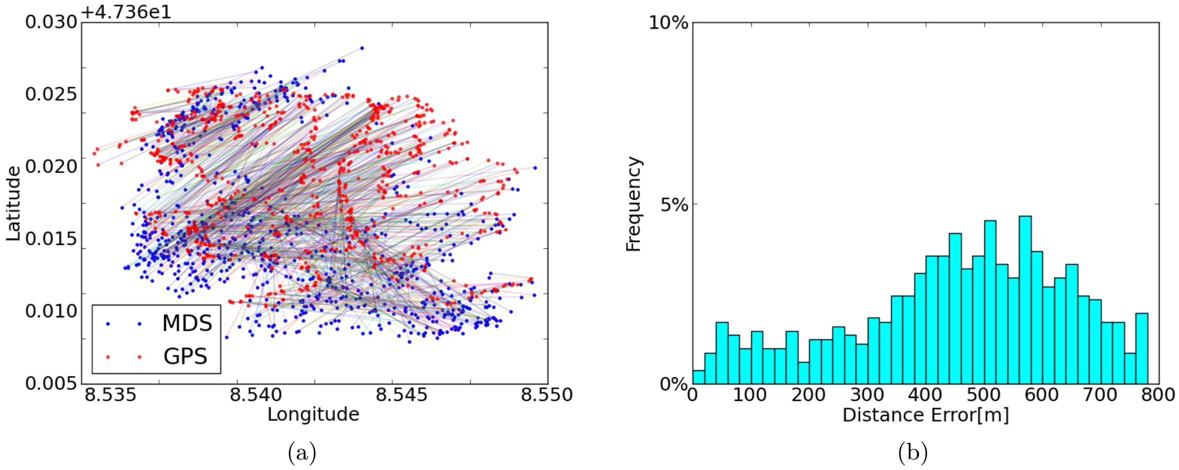
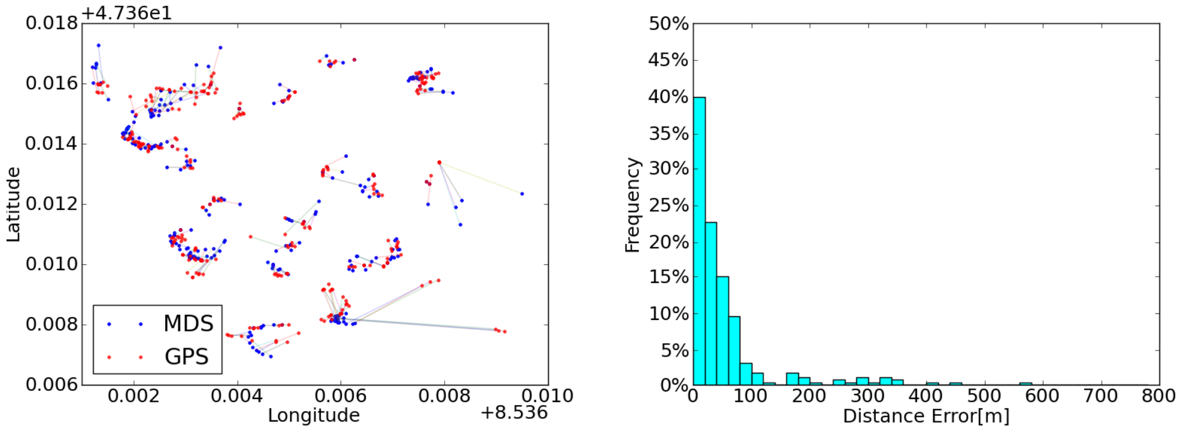


Figure 6.2.: No pruning: Topology reconstruction and localization error for WiFi+GSM fingerprints. Threshold: $\theta = 0.0$

Let us now have a closer look at some of the obtained results. Generally, we obtain better results by considering WiFi+GSM fingerprints compared to using only WiFi or only GSM. By setting $\theta = 0$, the similarity graph is not pruned. Figure 6.2a shows the reference topology results from the WiFi+GSM fingerprints in blue together with the GPS ground truth in red. Ideally, the two graphs completely overlap. Figure 6.2b shows a histogram of the corresponding error distribution. The median error localization is $574m$. By increasing θ, the similarity graph is being pruned. Figure 6.3 shows the MDS-based topology reconstruction by applying a pruning threshold $\theta = 0.5$ on the WiFi+GSM fingerprints. As listed in Table 6.4, of WiFi+GSM, only $\delta = 36\%$ of the fingerprints can be used for the reference topology while the rest of the fingerprints do not fulfill the required similarity criteria. However, the median accuracy is now $30m$. With a pruning threshold $\theta = 0.6$, we achieve a median accuracy of $26m$ while being able to localize 14% of the fingerprints. With this, we see that by removing low similarity values we are not able to locate all fingerprints anymore but, on the other hand, the localization accuracy increases significantly. Hence, our method can automatically detect fingerprints which can not be located reliably and for the others provide a location estimation with accuracy in a similar range as related work [3]. Only 7% of all fingerprints in the reference topology need to be geo-referenced. This is far less than the 100% required in state-of-the-art systems.



(a) (b)

Figure 6.3.: Topology reconstruction and localization error for WiFi+GSM fingerprints. Threshold: $\theta = 0.5$

6.1.3. Evolution of the reference topology

Our approach fits a collaborative approach where the localization estimation starts with a few fingerprints and gradually grows by adding new ones. Hereby, at the beginning, when only a few data points are present, the provided localization is expected to be rather inaccurate or a localization is not possible at all as the majority of similarities stem from long distance measures and hence get pruned. However, gradually, we expect a denser sampling of the region resulting in smaller distances between fingerprints and thus larger similarity values can be expected which remain during the pruning step. With this, we expect the localization method to provide more accurate results over time. To investigate this behavior, we observe the relation between median error rate and the number of considered samples by adding samples. We start with a minimal set of three fingerprints and gradually add new ones. Only fingerprints that can be localized are considered. Figure 6.4 shows that the obtained result follows the expected trend that the localization error decreases by gradually adding new fingerprints arrive. The dotted red line represents the median location accuracy obtained by Place Lab [3]. We see a convergence towards a comparable error rate.

Figure 6.3 shows the MDS-based topology reconstructions for different threshold θ on the WiFi+GSM fingerprints. The effect of our prunning threshold θ can be seen in the error histograms for different values of θ.

According to these results, better accuracy can be expected calculating the expected separation distances instead of using inverse similarities to estimate the separation distance within a set fingerprints in order to calculate a topology. However, in order to be able to use expected separations, several previous information from the area where the system intends to operate is necessary to model the similarity behaviour, thus, we would rely preferably on using dissimilarities despite they show a lower accuracy, but anyway comparable to the achieved through calculating expected separation distances.

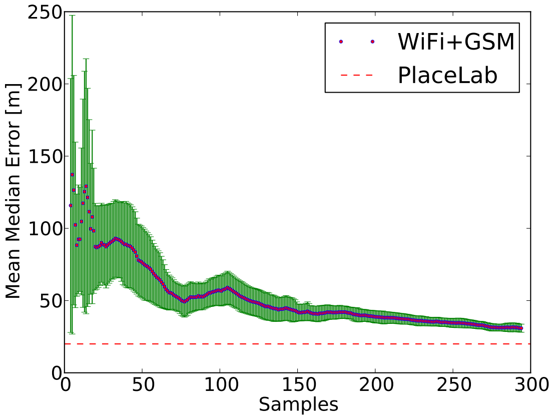
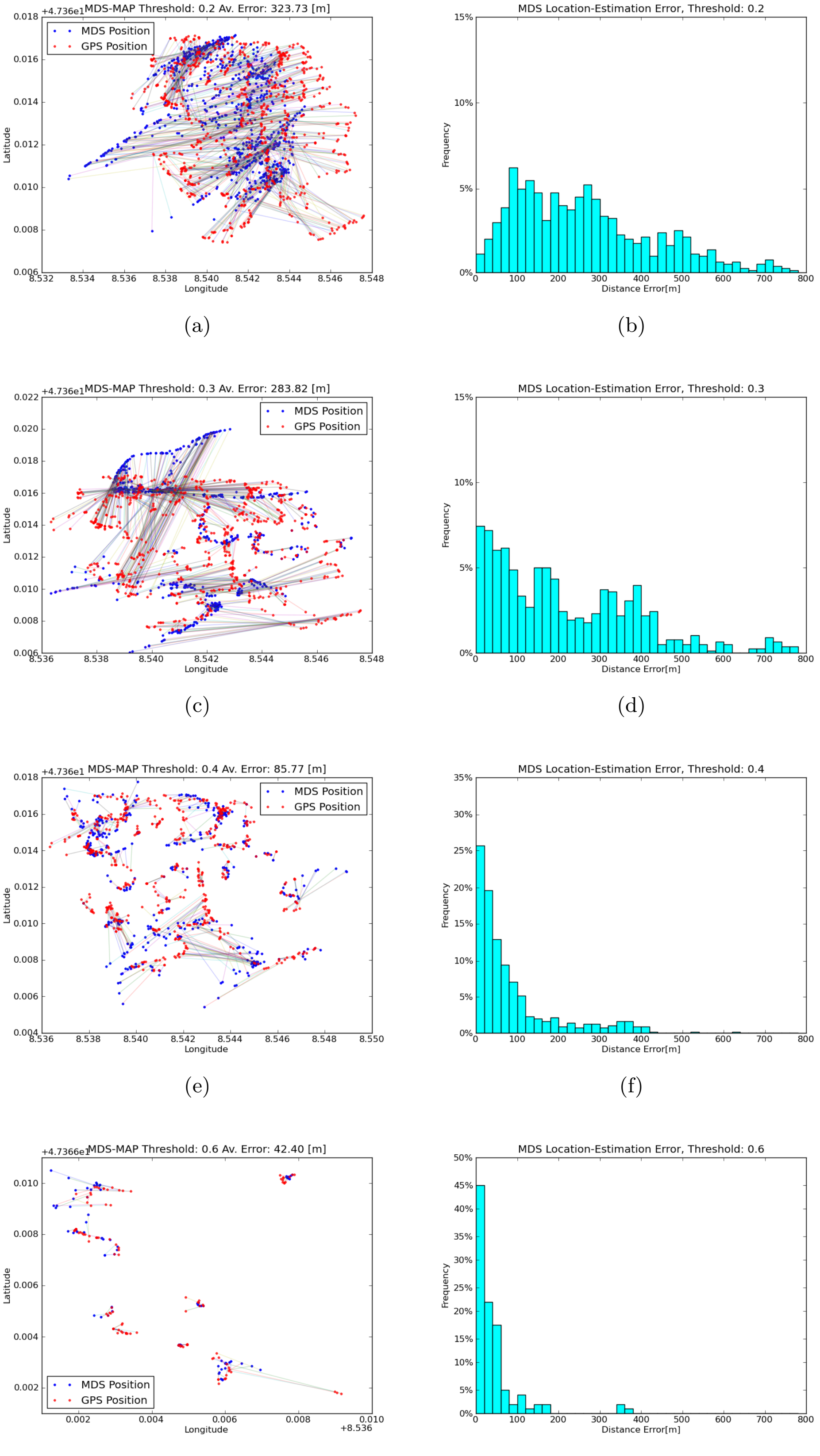


Figure 6.4.: Evolution of the localization error by gradually adding new fingerprints to the reference topology. Threshold $\theta = 0.5$

Another important note is that according to our results, the considered area in the experiment has a WiFi space of 2028 dimensions i.e. access points, and a GSM space of 66 dimensions i.e. GSM cells. That means that the density of WiFi access points is 30.73 access points per GSM cell. In terms of area units, and considering GSM cells as circular areas of 200 square meters size, would be a density of 1.5 access points every 10 square meters, or 0.15 access points per square meter.



(a) (b)

(c) (d)

(e) (f)

(g) (h)

Figure 6.5.: Topology reconstruction and localization error. (a),(b) $\theta = 0.2$, (c),(d) $\theta = 0.3$, (e),(f) $\theta = 0.4$, (g),(h) $\theta = 0.6$

CHAPTER 7

Discussion

The present chapter summarizes the work, presents a review of the major findings and achievements, gives a look at the expected limitations of implementing such an approach, provides a conclusion and propose possible further work that could be done in this direction.

7.1. Summary and Contributions

The aim of this work was to evaluate, to get a better understanding of the potential behind a localization method relying on the existing WiFi, GSM infraestructure, by using a fingerprinting approach without requiring the previous elaboration of a radio map. Fingerprinting is not novel per se, a lot of related work in both with WiFi and GSM information has been done so far, and its is proved that typical fingerprinting approaches are suitable for localization purposes. The main drawback of mostly all the fingerprinting methods is that they do require a training phase, where a lot of fingerprints must be recorded and associated with a geographical location. Normally these training phases can be done through the so called *war driving* process where a radio map containing a access points associated to a geographical position in a determinated area is obtained and stored in a database, then the system can estimate user locations by comparing a fingerprint with unknown location to those previously stored.

The present work intended to evaluate the feasibility of a localization method without requiring an initial radio map construction, avoiding the use of GPS. As potentially useful in the so-called Urban Canyons or inside buildings where GPS is not available i.e. in GPS less enviroments.

The use of multidimensional scaling for localization purposes has been addressed for sensor networks and for WiFi infraestructure in controlled enviroments. A experiment in a non-controlled enviroment with the purpose of obtaining a dataset to develop and evaluate an algorithm was performed. The obtained dataset consists of GPS location traces, WiFi fingerprints, and GSM fingerprints, this data set was examined, processed and analized to learn about its behaviour and how its particular characteristics could allow the design and implementation of localization method.

Based on the findings about the dataset behaviour, and on previous work, an algorithm was designed which starting with a small set of known information can provide localization and at the same time incrementally expand its known information i.e. reconstruct an approximated topology. The algorithm was evaluated on our dataset to assess its accuracy and to answer our research questions.

7.2. Findings, Achievements and Conclusions

This work presents a fingerprinting method for localizing mobile devices in urban spaces using MDS-based embedding of WiFi+GSM fingerprints to obtain a reference topology. The novelty of our method is threefold:

- Only a fraction of the training set's fingerprints needs to be geo-referenced. This allows to include fingerprints into reference databases also in the absence of GPS reception and does not require a manual labeling.
- By removing low similarity values, increased robustness against multipath, shadow fading and other influences that affect similarity estimations can be provided.
- The method is ideal for a collaborative approach: Users provide a fingerprint to receive a location estimation. Simulteneously, this fingerprint can be used to refine and extend the topology estimation. Hence, we can gradually increase the covered space without requiring further efforts by the users.

Our evaluation shows that by increasing the pruning threshold θ, more fingerprints are discarded and cannot be located. However, for the remaining fingerprints, the accuracy of the localization increases. For $\theta = 0.5$, our method could locate 36% of the fingerprints with a median error of $30m$. Only 7% of the fingerprints were geo-referenced and the rest could be positioned witout any corresponding location informat but only considering their similarity. We further show with our data set that the localization error decreases as new fingerprints are added and converges to an accuracy comparable to related work. The reason that a fingerprint cannot be localized is that there are not enough similar fingerprints to be found. A dense, uniform sampling of the space could increase the ratio of fingerprints that can be localized. Further, a minimal density of WiFi access points in an urban area is required so that signal from different networks overlap. The density of access points in our experiment area was on average 1.5 access points every $10m^2$ circular area. We expect this number to be reasonable for many urban areas and indoor venues and hence, comparable results can be expected.

For a real-time implementation, the computational complexity is a key factor: At the core of MDS is an eigen-decomposition on an $n \times n$ symmetric matrix which for classic MDS takes $O\left(n^3\right)$ time. However, it can be reduced to $O\left(n \lg n\right)$ steps and easily parallelized for use with large datasets [42]. When a new fingerprint arrives, it takes $O\left(C \cdot m \cdot n + m(n+1)^3\right)$ time for our algorithm to generate location results. Hereby, n is the number of closest fingerprints, m the number of anchor points, usually $m = 3$, and C the cost of computing and accessing each entry of the dissimilarity matrix built with the new and the closest fingerprints. We expect the method to be scalable to also work with large data sets.

7.3. Limitations

WiFi fingerprints provide differentiation in a very short range, up to 30 meters more precisely. This situation is explained by the area being considered where basically streets are surrounded by high buildings, these structures block WiFi signals, thus the mobile devices will likely record very different WiFi fingerprints beyond a separation distance of 30 to 40 meters. It leads to the problem that WiFi fingerprints can not be used to estimate long separations with proper accuracy, because they can provide no reliable or even no similarity for large separation distances. GSM signals however have a broader and more constant range, but their use as fingerprints fail to differentiate short separation distances, i.e. fingerprints inside the same GSM cell.

The major problem of our approach is to correctly represent the separation distance through similarities. The best we can do to avoid completely undesired results is to estimate separations by hops. But to do this, a high density of data is required. Another limitation is the possible slow growth of covered area, in this case, a collaborative approach could benefit a fast growth, for example, many users providing information at the same time allow the rapid expansion of the topology. However, there is a real and important limitation that arises when the known topology becomes too big, as a larger set of fingerprints must be treated, and the number of known networks becomes bigger, the computational complexity and required amount of calculations, and thus the required processing time to provide results becomes longer. Further, processing multiple requests at the same time could mean a very high computational effort. In this case, a distributed approach could be considered.

7.4. Outlook

Our results suggest that WiFi & GSM based algorithm should be able to operate in areas without reliable GPS reception or indoors, providing simultaneously localization and topological results. However, it is far away from being perfect. For example, the percentage of the initial topology that could be reconstructed is not even a half. This is due to the characteristics of our dataset, as previously said, the higher the fingerprints density, the better and bigger number of expected results.

According to [31] a localization method should have a location resolution between 100 meters, and the timeliness should be in seconds, thus our present work clearly fullfills these conditions, and gives a promising insight into what could be achieved specially for indoor localization. However there is a lot of research ahead, specially in optimization of topology estimations. For example, in the present work we did not considered a iterative MDS approach because we assumed that it would have no sense to increase computational complexity trying to fit a result to a expected result which accuracy we are not sure about. However, this expected result that in our case are dissmilarities, within a certain range, up to some point still represent somehow the real separation distances, thus an evaluation on how an iterative MDS could improve the results would be interesting.

On the other side, the real world applicability of such an approach should require further considerations, specially if a collaborative approach is desired. These considerations are mostly related to the required computational effort which is proportional to the size of the geographical zone, the density of access points and the amount of users that intend to make use of such a system. Further, another measurements that can provide better representation of the geographical separation distances could be considered. Finally, despite the promising results of an approach like the one we have presented, we should say that GPS is still a reliable localization method for those areas where WiFi and GSM networks are not widely spread. Our approach is suitable for providing location estimations in regions where GPS information is either unreliable or not present at all and hence ideal for urban spaces and indoor venues. We see a promising application of our method by combining it with existing systems such as Place Lab [3] to extend their functionality into areas where a GPS-based indexing is not possible. GPS-referenced fingerprints obtained in regions with good reception can serve as anchor points. With our method the covered space can gradually grow as people are using the system without the requirement of manual labeling of fingerprints.

Bibliography

[1] A. Küpper, *Location–based Services — Fundamentals and Operation.* John Wiley & Sons, 2005.

[2] M. B. Kjærgaard, H. Blunck, *et al.*, "Indoor positioning using gps revisited," in *Proc. of the 8th International Conference on Pervasive Computing*, 2010.

[3] A. LaMarca, Y. Chawathe, *et al.*, "Placelab: Device positioning using radio beacons in the wild," *Pervasive Computing*, 2005.

[4] P. Bahl and V. Padmanabhan, "Radar: An in-building rf-based user location and tracking system," in *Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies*, vol. 2, IEEE, 2000.

[5] Y. Cheng, Y. Chawathe, A. LaMarca, and J. Krumm, "Accuracy characterization for metropolitan - scale wifi localization," in *Proceedings of the 3rd International Conference on Mobile Systems, Applications and Services*, ACM, 2005.

[6] A. W. Tsui, W.-C. Lin, W.-J. Chen, P. Huang, and H.-H. Chu, "Accuracy performance analysis between war driving and war walking in metropolitan wi-fi localization," *Trans. Mob. Comput.*, vol. 9, no. 11, 2010.

[7] G. Chandrasekaran, M. A. Ergin, M. Gruteser, and R. P. Martin, "Bootstrapping a location service through geocoded postal addresses," in *Proc. of the Third International Symposium on Location- and Context-Awareness*, 2007.

[8] P. Nurmi, S. Bhattacharya, and J. Kukkonen, "A grid-based algorithm for on-device gsm positioning," in *Proc. of the 12th Int. Conf. on Ubiquitous Computing*, ACM, 2010.

[9] P. Bolliger, "Redpin-adaptive, zero-configuration indoor localization through user collaboration," in *Proc. of the 1st Int. Workshop on Mobile Entity Localization and Tracking in GPS-less Environments*, ACM, 2008.

[10] O. Woodman and R. Harle, "Rf-based initialisation for inertial pedestrian tracking," *Pervasive Computing*, 2009.

[11] I. Borg and P. Groenen, *Modern Multidimensional Scaling: Theory and applications.* Springer Verlag, 2005.

[12] A. Hopper, A. Harter, and T. Blackie, "The active badge system," in *Proc. of the INTERACT'93 and CHI'93 Conference on Human Factors in Computing Systems*, ACM, 1993.

[13] L. Aalto, N. Göthlin, J. Korhonen, and T. Ojala, "Bluetooth and wap push based location-aware mobile advertising system," in *Proc. of the 2nd International Conference on Mobile Systems, Applications, and Services*, ACM, 2004.

[14] N. Priyantha, A. Chakraborty, and H. Balakrishnan, "The cricket location-support system," in *Proc. of the 6th Annual International Conference on Mobile Computing and Networking*, ACM, 2000.

[15] G. Mao, B. Fidan, and B. Anderson, "Wireless sensor network localization techniques," *Computer Networks*, vol. 51, no. 10, 2007.

[16] D. Niculescu and B. Nath, "Ad-hoc positioning system (aps) using aoa," in *Twenty-Second Annual Joint Conference of the IEEE Computer and Communications*, vol. 3, IEEE, 2003.

[17] A. Savvides, C. Han, and M. Strivastava, "Dynamic fine-grained localization in ad-hoc networks of sensors," in *Proc. of the 7th Annual International Conference on Mobile Computing and Networking*, ACM, 2001.

[18] R. Fontana, E. Richley, and J. Barney, "Commercialization of an ultra wideband precision asset location system," in *Ultra Wideband Systems and Technologies*, IEEE, 2003.

[19] J. Hightower, G. Borriello, and R. Want, "Spoton: An indoor 3d location sensing technology based on rf signal strength," *UW CSE 00-02-02*, 2000.

[20] M. Y. Chen, T. Sohn, *et al.*, "Practical metropolitan-scale positioning for gsm phones," in *UbiComp 2006: Ubiquitous Computing*, IEEE, 2006.

[21] S. Teller, J. Battat, B. Charrow, *et al.*, "Organic indoor location discovery," in *Technical Report MIT-CSAIL-TR-2008-075*, Computer Science and Artificial Intelligence Laboratory, MIT, 2008.

[22] W. Grisswold, P. Shanahan, S. Brown, *et al.*, "Activecampus: Experiments in community-oriented ubiquitous computing," in *Computer*, vol. 37, pp. 73–81, IEEE, 2004.

[23] J. Krumm, G. Cermak, and H. E., "Rightspot: A novel sense of location for a smart personal object," in *Proceedings of International Conference on Ubiquitous Computing UBICOMP*, IEEE, 2004.

[24] B. Flannery, W. Press, S. Teukolsky, and W. Vetterling, “Numerical recipes in c,” *Press Syndicate of the University of Cambridge, New York*, 1992.

[25] J. Eberspacher, H.-J. Vogel, and C. Bettstetter, *GSM Switching, Services and Protocols.* John Wiley & Sons Ltd., 2001.

[26] A. Varshavsky, M. Chen, *et al.*, “Are gsm phones the solution for localization?,” in *Proceedings of the Seventh IEEE Workshop on Mobile Computing Systems & Applications*, IEEE, 2006.

[27] P. Jaccard, “The distribution of the flora in the alpine zone,” *New Phytologist*, vol. 11, no. 2, 1912.

[28] H. Wolda, “Similarity indices, dample dize and diversity,” *Öcologia*, vol. 50, no. 3, 1981.

[29] D. Kim, Y. Kim, D. Estrin, and M. Srivastava, “Sensloc: Sensing everyday places and paths using less energy,” in *Proc. of the 8th ACM Conference on Embedded Networked Sensor Systems*, ACM, 2010.

[30] G. Linares, “Escalamiento multidimensional: Conceptos y enfoques,” *Journal of Operational Research*, vol. 22, no. 2, pp. 173–183, 2001.

[31] B. Wei, W. Chen, and X. Ding, “Advanced mds based localization algorithm for location based services in wireless sensor network,” in *Ubiquitous Positioning Indoor Navigation and Location Based Service UPINLBS*, IEEE, Oct. 2010.

[32] W. Ni, W. Xiao, Y. K. Toh, and C. K. Tham, “Fingerprint-mds based algorithm for indoor wireless localization,” in *IEEE 21st International Symposium on Personal Indoor and Mobile Radio Communications*, Sept. 2010.

[33] O. Goussevskaia, M. Kuhn, M. Lorenzi, and R. Wattenhofer, “From web to map: Exploring the world of music,” in *Web Intelligence and Intelligent Agent Technology*, vol. 1, IEEE, 2008.

[34] Y. Shang, W. Ruml, Y. Zhang, and M. Fromherz, “Localization from mere connectivity,” in *Proc. of the 4th ACM International Symposium on Mobile ad-hoc Networking & Computing*, pp. 201–212, ACM, 2003.

[35] Y. Shang and W. Ruml, “Improved mds-based localization,” in *23rd Annual Joint Conference of the IEEE Computer and Communications Societies*, IEEE, 2004.

[36] P. Schläpfer, “Detection of large-scale crowd behavior patterns from mobile phone sensors at the example of flocking,” *Master Thesis, ETH Zürich*, 2010.

[37] J. Koo and H. Cha, “Autonomous construction of a wifi access point map using multidimensional scaling,” *Pervasive Computing*, 2011.

[38] T. Pulkkinen, T. Roos, and P. Myllymäki, “Semi-supervised learning for wlan positioning,” *Artificial Neural Networks and Machine Learning*, 2011.

[39] G. Inc., “Htc nexus one.” http://www.htc.com/www/product/nexusone/specification.html, 2011. [Online; accessed 03-March-2011].

[40] M. Covell and M. Slaney, *Matlab Code to compute Multi-dimensional Scaling (MDS)*. Purdue University, Indiana USA, 2002.

[41] J. Montavont and T. Noel, “Ieee 802.11 handovers assisted by gps information,” in *Int. Conf. on Wireless and Mobile Computing, Networking and Communications*, 2006.

[42] T. Yang, J. Liu, *et al.*, “A fast approximation to multidimensional scaling,” in *Proc. of Workshop on Computation Intensive Methods for Computer Vision*, 2006.

List of Figures

1.1. Addressed approach. 3

3.1. Zones covered during the experiment 14

4.1. WiFi+GSM Similarities for three different devices. 18
4.2. Probability Behaviour of different Similarity Values for various Indices. 20
4.3. Probability Behaviour of different Similarity Values for various Indices. 21
4.4. Expected values and Variance for different Similarity Measures. 22
4.5. Probability Distribution for different Similarity Measures. 23
4.6. Cosine Similarity. Coloured curves show different order polynomial regressions. a,b)WiFi, c,d)GSM, e,f)WiFi+GSM. Second recording. 25
4.7. Similarities using Tanimoto Coefficient. Coloured curves show different order polynomial regressions. a,b)WiFi, c,d)GSM, e,f)WiFi+GSM. Second recording. . 26
4.8. Similarities using Jaccard Index. Coloured curves show different order polynomial regressions. a,b)WiFi, c,d)GSM, e,f)WiFi+GSM. Second recording. 27
4.9. Similarities using Sørensen Index. Coloured curves show different order polynomial regressions. a,b)WiFi, c,d)GSM, e,f)WiFi+GSM 28
4.10. Probability behaviour of different similarity values, second record. 29
4.11. Probability behaviour of different similarity values, second record. 30
4.12. Expected values and variance for different similarity measures, second record. . . 31
4.13. Probability distribution for different similarity measures. 32

5.1. MDS scheme 34
5.2. Histogram and cumulative distribution function of GPS accuracy over the complete data set. The accuracy value defines the radius of 95% confidence circle. . . 38
5.3. Raw topology reconstrucion 39
5.4. Ground Truth of the fingerprint to be located 44
5.5. Initial Maps: left column GSM Information, right column WiFi/GSM Information. 44

5.6. Use of Close Fingerprints Threshold: left column GSM Information, right column WiFi/GSM Information. . . . 46
5.7. Use of Edges Threshold: left column GSM Information, right column WiFi/GSM Information. . . . 47
5.8. Algorithm Operation: left column GSM Information, right column WiFi/GSM Information. . . . 48
5.9. Excessive edge removal effect . . . 49
5.10. Localization Problem Overview . . . 51
5.11. Process to build the reference topology (top) and to obtain a location estimation (bottom). . . . 52
5.12. Operation scheme to locate a new fingerprint and to consider its inclusion into the reference topology. . . . 55
5.13. Processing a new Fingeprint . . . 56
5.14. Algorithm operation diagram . . . 57
5.15. Building Graph Process . . . 58
5.16. Fixing Map process diagram . . . 59

6.1. Mean and variance valuess for Tanimoto Similarity Measures. a)WiFi, b)GSM, c)WiFi+GSM . . . 62
6.2. No pruning: Topology reconstruction and localization error for WiFi+GSM fingerprints. Threshold: $\theta = 0.0$ . . . 64
6.3. Topology reconstruction and localization error for WiFi+GSM fingerprints. Threshold: $\theta = 0.5$ . . . 65
6.4. Evolution of the localization error by gradually adding new fingerprints to the reference topology. Threshold $\theta = 0.5$ . . . 66
6.5. Topology reconstruction and localization error. (a),(b) $\theta = 0.2$, (c),(d) $\theta = 0.3$, (e),(f) $\theta = 0.4$, (g),(h) $\theta = 0.6$ . . . 67

B.1. Similarities using Jaccard Index. Curves show different order polynomial regressions. a,b)WiFi, c,d)GSM, e,f)WiFi+GSM . . . 84
B.2. Similarities using Sørensen Index. Curves show different order polynomial regressions. a,b)WiFi, c,d)GSM, e,f)WiFi+GSM . . . 85
B.3. Similarities using First Kulczynski Index. Curves show different order polynomial regressions. a,b)WiFi, c,d)GSM, e,f)WiFi+GSM . . . 86
B.4. Similarities using Second Kulczynski Index. Curves show different order polynomial regressions. a,b)WiFi, c,d)GSM, e,f)WiFi+GSM . . . 87
B.5. Similarities using Tanimoto Coefficient. Curves show different order polynomial regressions. a,b)WiFi, c,d)GSM, e,f)WiFi+GSM . . . 89
B.6. Similarities using Cosine Similarity. Curves show different order polynomial regressions. a,b)WiFi, c,d)GSM, e,f)WiFi+GSM . . . 90

List of Tables

4.1. Presentation order of the characterization results 19

6.1. Overview of evaluated parameter 63
6.2. Summary of the algorithm performance for different thresholds θ using WiFi-based fingerprints. 63
6.3. Summary of the algorithm performance for different thresholds θ using GSM-based fingerprints. 63
6.4. Summary of the algorithm performance for different thresholds θ using WiFi+GSM-based fingerprints. 63

B.1. Regression coefficients for Jaccard Index, planned walk. 91
B.2. Regression coefficients for Sørensen Index, planned walk. 91
B.3. Regression coefficients for 1_{st} Kulczynski Index, planned walk. 92
B.4. Regression coefficients for 2_{st} Kulczynski Index, planned walk. 92
B.5. Regression coefficients for Tanimoto Coefficient, planned walk. 93
B.6. Regression coefficients for Cosine Similarity, planned walk. 93
B.7. Mean Square Errors for a first order regression. Planned Walk 94
B.8. Mean Square Errors for a second order regression. Planned Walk 94
B.9. Mean Square Errors for a third order regression. Planned Walk 95
B.10.Mean Square Errors for a fourth order regression. Planned Walk 95

APPENDIX A

Excerpts of the Code

```java
package com.coenosense.logger.worker;

import java.util.HashMap;

/**
 * Performs GSM Scans at regular intervals (set by mInterval) and adds
 * the result directly to the Publisher.
 *
 * TODO: Currently, the doWork() function requests a new scan to be performed
 * and the immediately reads the scan results. It also performs GSM information
 * readings when a change in the signal strength or in the cell location is
 * detected, by starting a PhoneStateListener object.
 * The logic steps follows the structure of the previously existing workers
 * in order to keep consistency with the entire application.
 *
 * @author Kristian Cugia
 */

public class GSMWorker extends BackgroundWorker{

    protected static final String TAG = "GSMWorker";
    private ExecutorService mExecutor;
    private TelephonyManager tphoneManager;
    private SignalStrength signalStrength;
    private GsmCellLocation cellLoc;
    private List<NeighboringCellInfo> neighboringCells;
    //private MyPhoneStateListener mPhoneListener;
    //JSONObject jsonEncoder;
    private MyPhoneStateListener mPhoneListener;

    public GSMWorker(Context ctx, BackgroundService bgs, Publisher pub) {
        super(ctx, bgs, pub);
        tphoneManager =
(TelephonyManager)ctx.getSystemService(Context.TELEPHONY_SERVICE);
        mInterval = new Long(5000); //Interval to perform GSM Scans

        /*The SignalStrength information is fetched from the API to a
PhoneStateListener, this
         * runs as another thread and due to the architecture of the application
must be started as a
         * simultaneous thread, to avoid interferences with the GSMWorker thread
and its scanning loops */
        Handler h = new Handler(Looper.getMainLooper());
        Runnable r = new Runnable() {
            public void run() {
                try{
                    mPhoneListener = new MyPhoneStateListener();
                    }catch(Exception ite){
                        Log.i("EXEP",ite.getMessage());
                    }
                    tphoneManager.listen(mPhoneListener,
PhoneStateListener.LISTEN_SIGNAL_STRENGTHS |
PhoneStateListener.LISTEN_CELL_LOCATION);
            }
        };
        h.post(r);
        // TODO Auto-generated constructor stub
    }

    @Override
    void doWork() {
        Log.d(TAG, "doWork()");
        mExecutor = Executors.newSingleThreadExecutor();
        mExecutor.execute(new Runnable() {
            //@Override
            public void run() {
                processGsmInfo();
            }
        });
    }

    public void processGsmInfo(){
        // GSM is "always active, unless some operator unavailability issue, not necessary to query for GSM state"

        cellLoc = (GsmCellLocation)tphoneManager.getCellLocation();
        JSONArray jsonArray;
        JSONObject jsonEncoder;
        if(cellLoc!=null){

            //GSM Information for Neighboring Cells
            neighboringCells= tphoneManager.getNeighboringCellInfo();
            Log.i(TAG, "Found " + (neighboringCells.size()+1) + "Cells");

            jsonArray = new JSONArray();
            jsonEncoder = new JSONObject();
            try {
                //GSM Information for current Cell
                jsonEncoder.put("CELLID", cellLoc.getCid());
                jsonEncoder.put("RSSI", (-113 + 2*signalStrength.getGsmSignalStrength()));
                jsonEncoder.put("LAC", cellLoc.getLac());
                //jsonEncoder.put("PSC", cellLoc.getPsc());
                jsonEncoder.put("NETTYPE", tphoneManager.getNetworkType());
                jsonEncoder.put("BER", signalStrength.getGsmBitErrorRate());
                jsonEncoder.put("NETNAME", tphoneManager.getNetworkOperatorName());
                if(tphoneManager.getSimState() == android.telephony.TelephonyManager.SIM_STATE_READY)
                    jsonEncoder.put("SIMNAME", tphoneManager.getSimOperatorName());
                jsonArray.put(jsonEncoder);


                //Get Information from Neighboring Cells
                for(NeighboringCellInfo ncinfo : neighboringCells){
                    jsonEncoder = new JSONObject();
                    jsonEncoder.put("CELLID", ncinfo.getCid()); //Cell ID
                    Log.d("NCELLID","="+ ncinfo.getCid());
                    jsonEncoder.put("RSSI", (-113 + 2*ncinfo.getRssi()));   //Received Signal Strength
                    jsonEncoder.put("LAC", ncinfo.getLac());    //Location Area Code
                    //jsonEncoder.put("PSC", ncinfo.getPsc());  //Primary Scrambling Code
                    jsonEncoder.put("NETTYPE", ncinfo.getNetworkType());   //Network Type. i.e. GPRS
                    jsonArray.put(jsonEncoder);
            }
                } catch (JSONException e) {
                    Log.w(TAG, "JSONException while building JSON of Scan-Results. " + e.getMessage());
                }


            HashMap<String, Object> toPublisher = new HashMap<String, Object>();

            toPublisher.put(Publisher.INTERNAL_TYPE, "gsm_scan");
            toPublisher.put(Publisher.INTERNAL_TS, Publisher.getSensorTimestamp());
            toPublisher.put(Publisher.INTERNAL_UNIXTS, System.currentTimeMillis());
            toPublisher.put(Publisher.INTERNAL_WINDOW, "");

            toPublisher.put("gsm_info", jsonArray.toString());
            mPublisher.incomingQueue.add(toPublisher);

            Log.d(TAG, "Finished GSM-Scan");
        }else{
            Log.i(TAG, "No Networks found!");
        }
    }

    @Override
    public int getSensorType() {
        return ProcessorManager.TYPE_GSM;
    }

    private class MyPhoneStateListener extends PhoneStateListener{
        @Override
        public void onSignalStrengthsChanged(SignalStrength signalStrength){
             super.onSignalStrengthsChanged(signalStrength);
             GSMWorker.this.signalStrength = signalStrength;    //To Use on the GSM scans
             //processGsmInfo();
             Log.i("RSSI","now" + signalStrength.getGsmSignalStrength());
        }
        @Override
        public void onCellLocationChanged(CellLocation location) {
             super.onCellLocationChanged(location);
             if(signalStrength != null){    //The first activation results on a change in the cellLocation, but it may not have fetched signalChanges yet
            //   processGsmInfo();
             }
             Log.i("Location","Changed");
        }
   };

    @Override
    public void stop() {
        super.stop();
        tphoneManager.listen(mPhoneListener, PhoneStateListener.LISTEN_NONE);
    }

}
```

APPENDIX B

Similarity Measurement Results

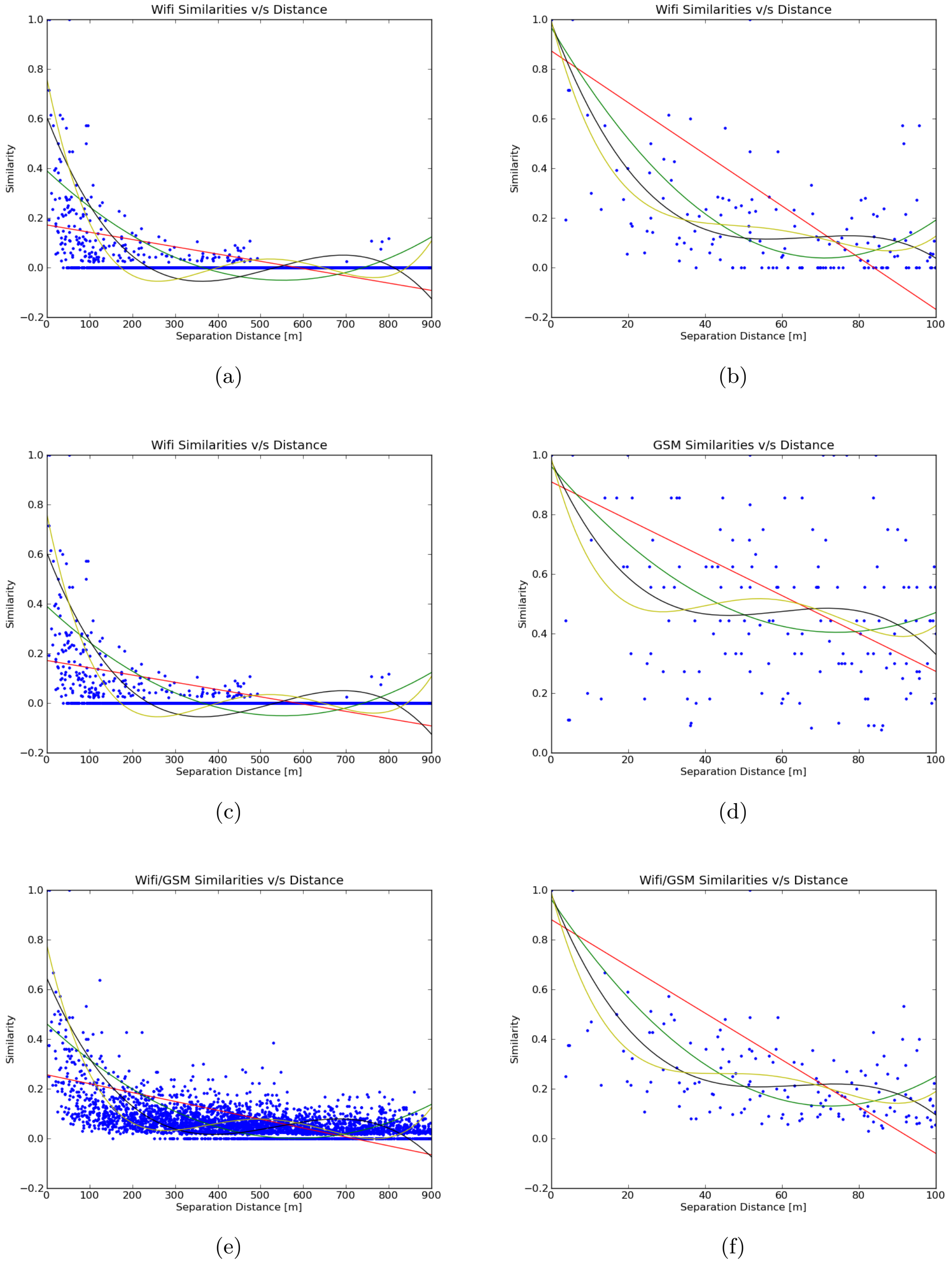
As described in section 4.1 the following similarity measurements were applied to our dataset.

Jaccard Index Given by equation 2.3, it estimates similarity between pair of sets. We used it to compare our fingerprints. For WiFi, it only uses the BSSID of the networks in each fingerprint. For GSM only the Cell IDs of each fingerprint, a WiFi+GSM fingerprint is obtained by merging a WiFi and a GSM fingerprint. If A is a fingerprint and B another, the measurement is made basically by dividing the number of common networks and/or cells between the two fingerprints by the total number of networks and/or cells present in the two fingerprints.

Sørensen Index Given by equation 2.4 works in a similar way as the Jaccard index, it evaluates only binary data. For WiFi it only uses the BSSID of the networks in each fingerprint. For GSM only the Cell IDs of each fingerprint, a WiFi+GSM fingerprint is obtained by merging a WiFi and a GSM fingerprint.

First Kulczynski Index Given by equation 2.8 as the previous indices, it works under binary samples, comparing the common data and the total data in a specific way. The main problem of using this index is that it is not defined for two completely similar fingerprints, in which case results in division by zero, and as it only makes use of simple network names and/or GSM cell's ID, there could be many equal fingerprints, which result in an undefined similarity. We modified the implementation of this index to give a similarity of 1 when two equal fingerprints are compared. But of course, it results in an unaccurate and not reliable index for the similarity estimation purposes of the present approach.

Second Kulczynski Index Given by equation 2.9, it works in a similar way as the previous index, in fact it is a modification of the First Kulczynski Index, but it do not present the undefined result for a two equal fingerprints comparison.



(a)

(b)

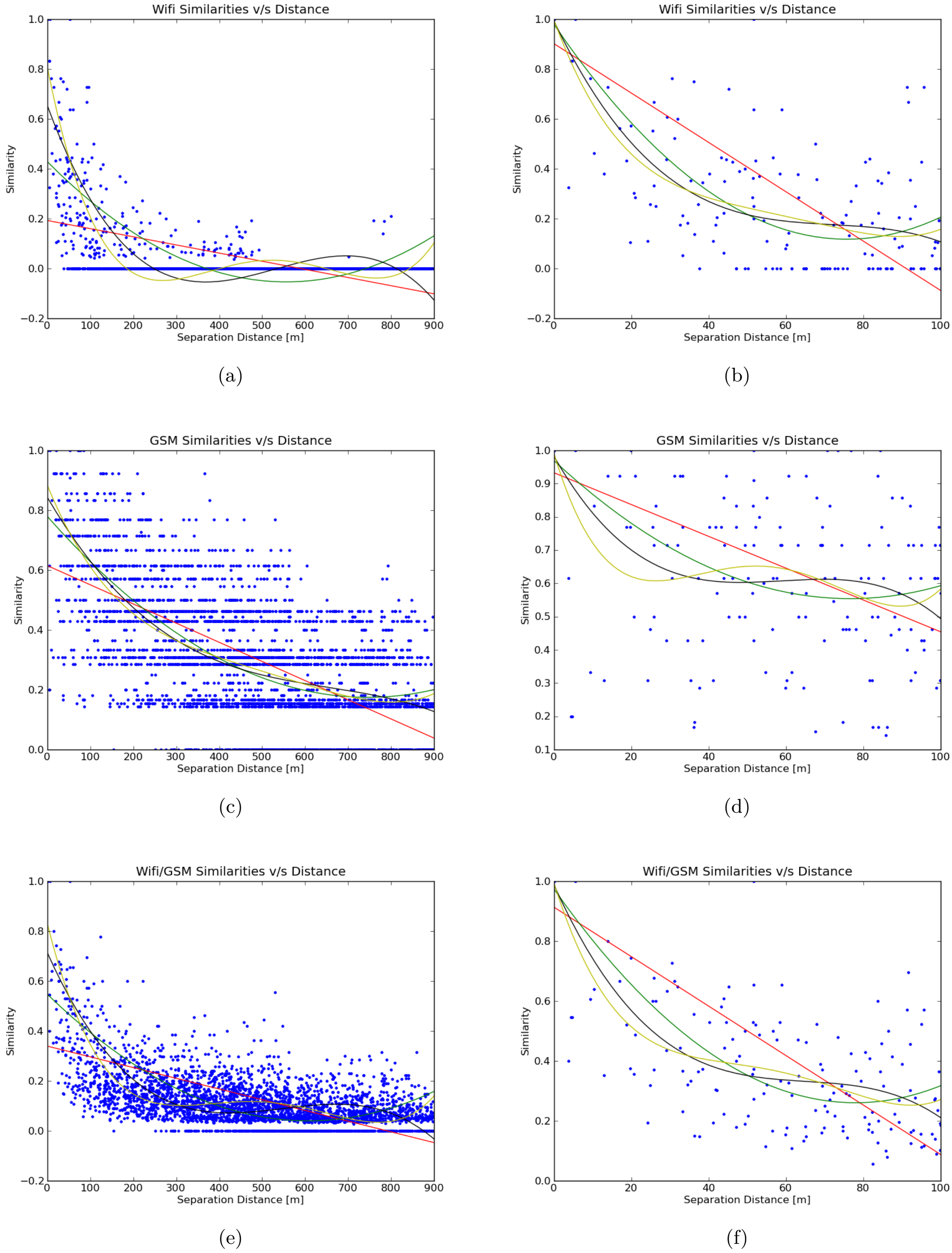
(c)

(d)

(e)

(f)

Figure B.1.: Similarities using Jaccard Index. Curves show different order polynomial regressions. a,b)WiFi, c,d)GSM, e,f)WiFi+GSM



(a)

(b)

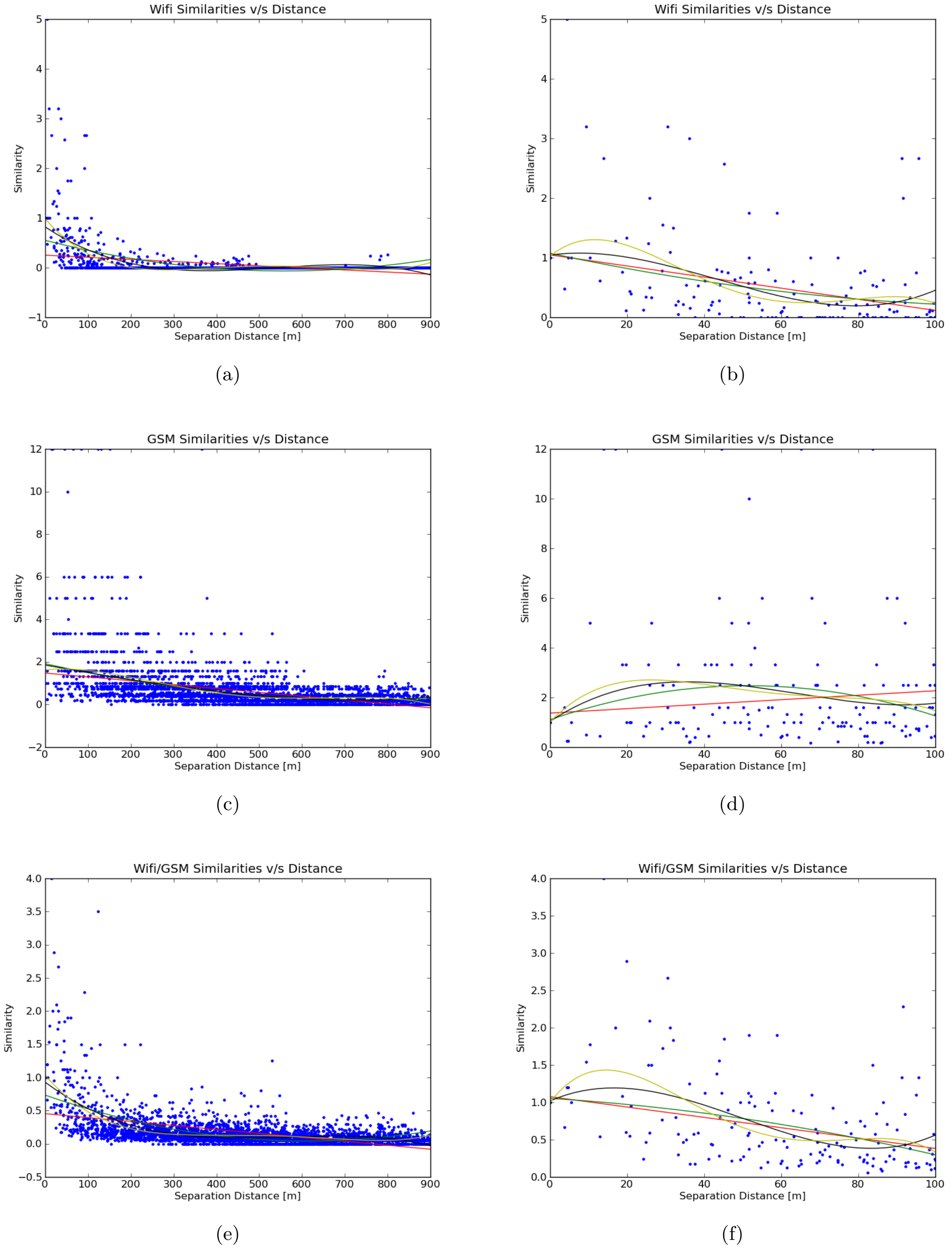
(c)

(d)

(e)

(f)

Figure B.2.: Similarities using Sørensen Index. Curves show different order polynomial regressions. a,b)WiFi, c,d)GSM, e,f)WiFi+GSM



(a)

(b)

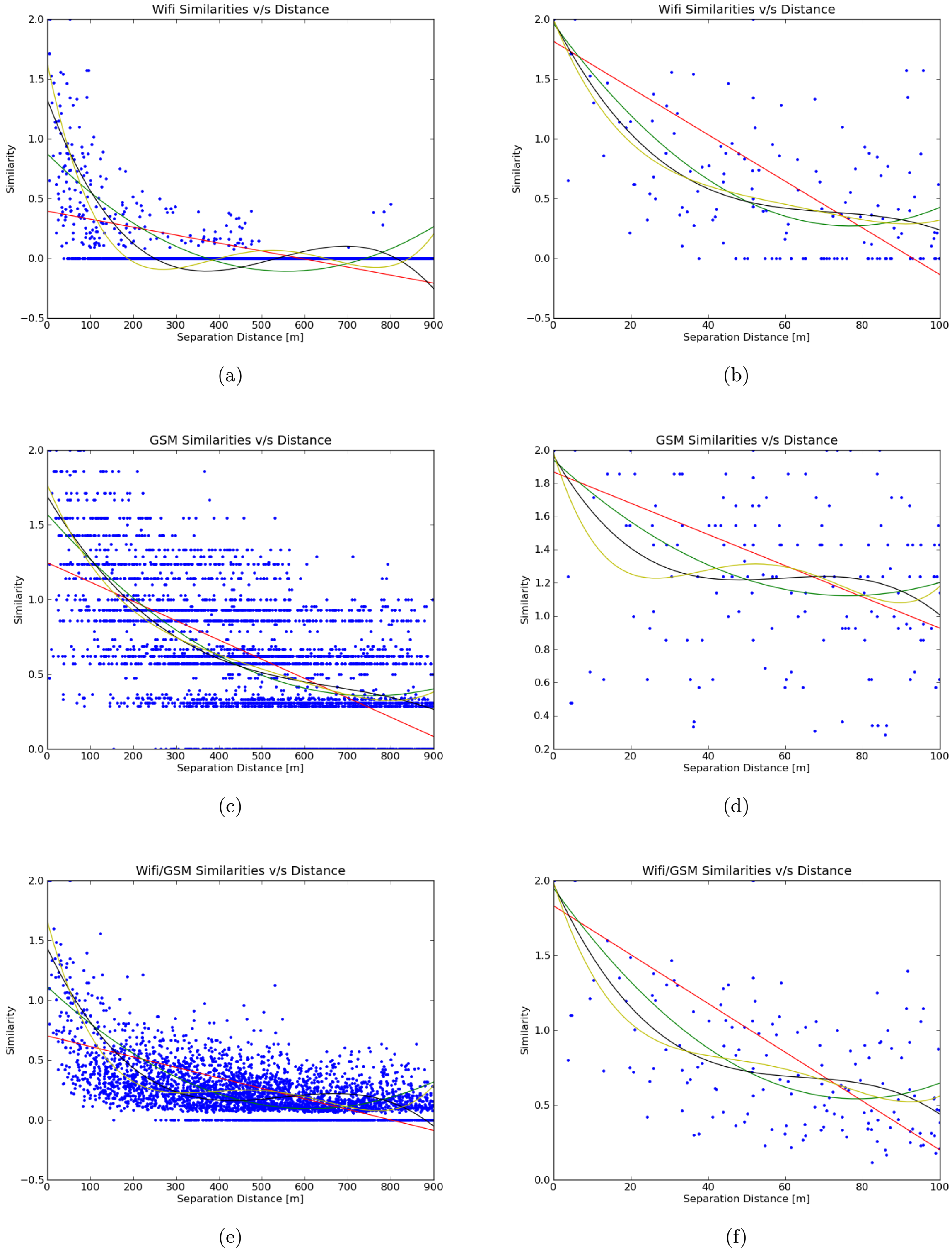
(c)

(d)

(e)

(f)

Figure B.3.: Similarities using First Kulczynski Index. Curves show different order polynomial regressions. a,b)WiFi, c,d)GSM, e,f)WiFi+GSM



(a) (b)

(c) (d)

(e) (f)

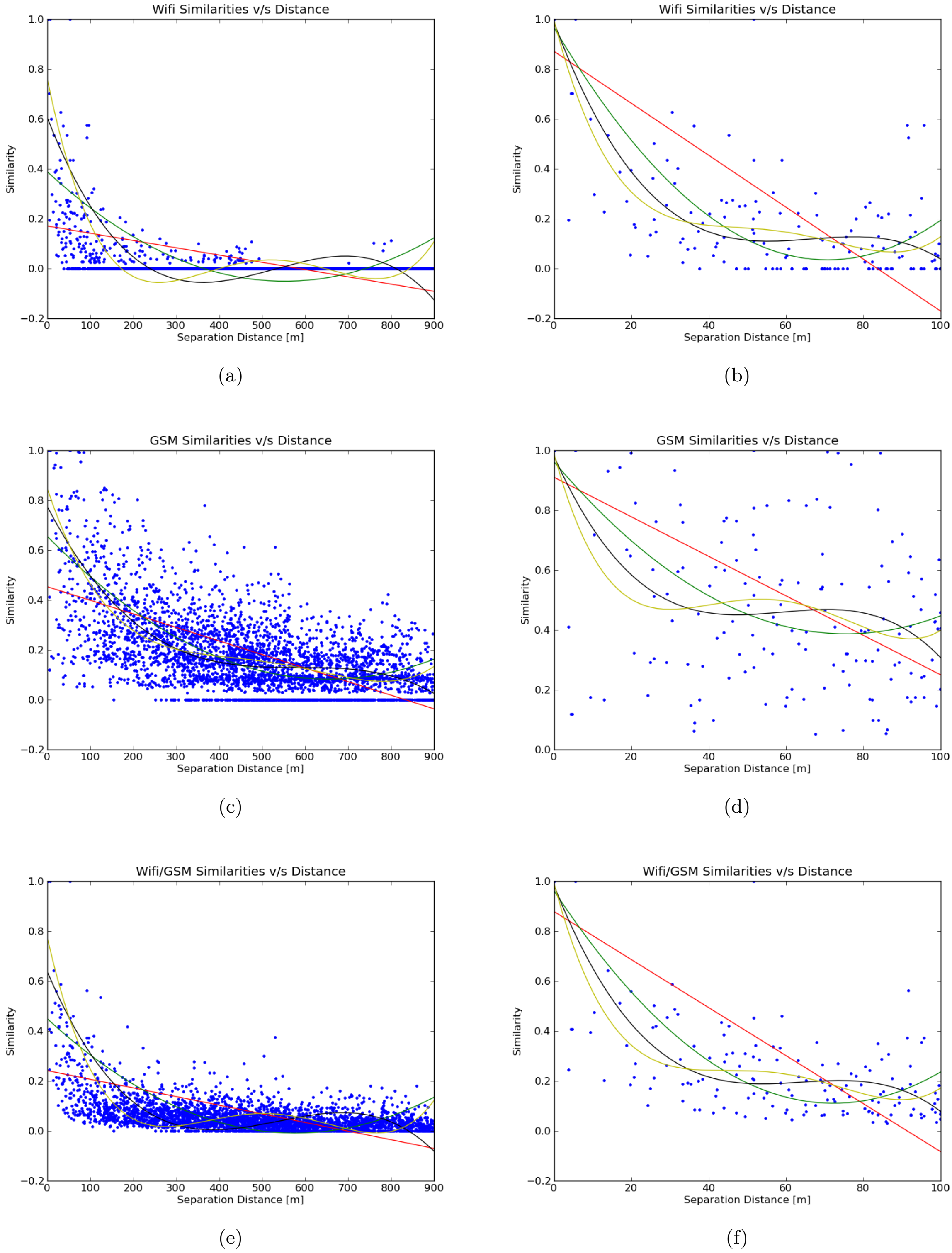
Figure B.4.: Similarities using Second Kulczynski Index. Curves show different order polynomial regressions. a,b)WiFi, c,d)GSM, e,f)WiFi+GSM

Vectorial Approaches

The following two estimations differ as the previously shown ones, in the fact that are statistics comparing non-binary data, in fact comparing vectors with magnitude and direction. We expect this kind of measurements to be more reliable and more accurate, as they evaluates in a broader sense our set of fingerprints. But the use of this measurement required a modification in the approach, as they compute dot vectorial products and these kind of operation is only possible between vectors of equal size. The approach is to build a space, in which all the fingerprints live. So, for the wifi fingerprints we builded a n-dimensional wifi space, where every BSSID of a network is dimension, and the signal level the component of the vector in this dimension. Similarly for the GSM fingerprints, every cellID is a dimension and the associated RSSI is the magnitude of this component. For the WiFi+GSM fingerprints, the two spaces are merged. This approach requires all the fingerprints to 'know' all the network names and/or cellIDs existing in all the fingerprints. Basically the space is the union of all the BSSID's and/or GSM cell IDs. Every fingerprint is extended to have a component in this n-dimensional space, but when originally a fingerprint do not see a particular BSSID and/or GSM cell, the magnitude of this component in the n-dimensional space is zero. With this approach it is expected to obtain better and more accurate similarity estimations.

Tanimoto Coefficient Given by equation 2.7, this statistic compares non-binary samples, it assumes the compared samples as vectors with a magnitude and direction.

Cosine Similarity Given by equation 2.6, this statistic uses a vectorial approach to estimate similarity between data. It is commonly used in information retrieval for example to compare how similar two documents are. The output is the cosine of an angle, which yields the value of -1 as meaning exactly opposite, 0 meaning independent and +1 meaning exactly the same. In our data, the obtained results are to be from 0 to +1, as there are no opposite fingerprints in the a meaningful sense, as it would require an absolute negative received signal strength (or positive dBm value), which is not possible, as the lower possible signal is no signal, not negative values.

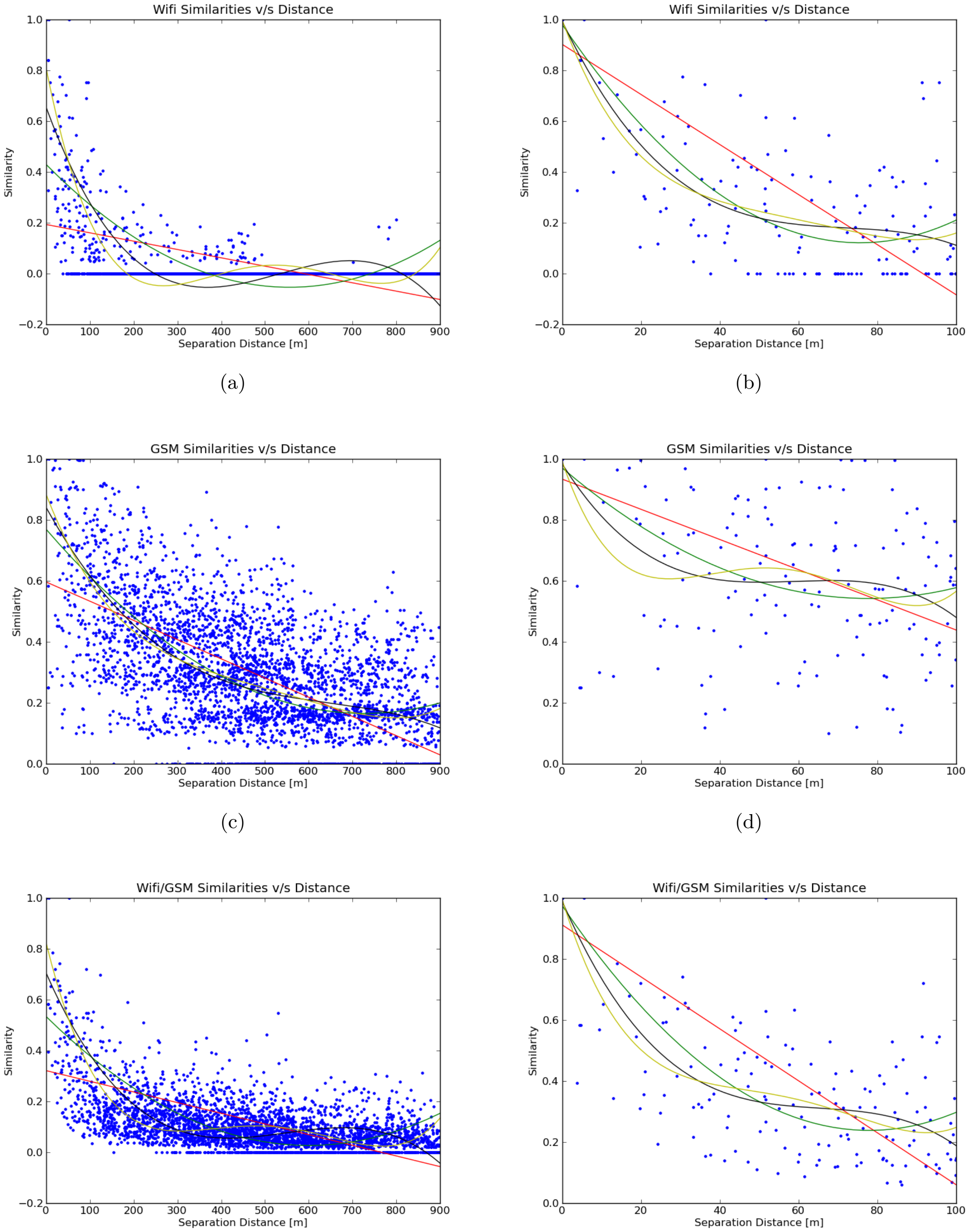


(a) (b)

(c) (d)

(e) (f)

Figure B.5.: Similarities using Tanimoto Coefficient. Curves show different order polynomial regressions. a,b)WiFi, c,d)GSM, e,f)WiFi+GSM



(a)

(b)

(c)

(d)

(e)

(f)

Figure B.6.: Similarities using Cosine Similarity. Curves show different order polynomial regressions. a,b)WiFi, c,d)GSM, e,f)WiFi+GSM

Four polynomial regressions were made to the data shown in each figure. A polynomial regression finds a curve that represents in a least square sense a set of points in a plot. The polynomial form would be like equations B.1 and B.2.

$$f(x) = \sum_{i=0}^{n} a_i x^i \tag{B.1}$$

$$f(x) = a_n x^n + a_{n-1} x^{n-1} + \ldots + a_1 x + a_0 \tag{B.2}$$

With n representing the polynomial order, and a_i representing the coefficients. The following table shows the coefficients derived from each regression for the data set obtained in the first part of the experiment described in chapter 3.

Jaccard Index	a_4	a_3	a_2	a_1	a_0
WiFi				-0.0003	0.1325
			2.29x10^{-6}	-0.0018	0.3157
		-1.38x10^{-8}	1.53x10^{-5}	-0.0051	0.5160
	7.28x10^{-11}	-1.03x10^{-7}	5.11x10^{-5}	-0.0102	0.6753
GSM				-0.0007	0.4867
			1.71x10^{-6}	-0.0018	0.6236
		-7.31x10^{-9}	8.59x10^{-6}	-0.0036	0.7291
	3.37x10^{-11}	-4.88x10^{-8}	2.51x10^{-5}	-0.0059	0.8029
WiFi/GSM				-0.0004	0.2316
			2.10x10^{-6}	-0.0017	0.3997
		-1.23x10^{-8}	1.37x10^{-5}	-0.0047	0.5775
	6.27x10^{-11}	-8.94x10^{-8}	4.45x10^{-5}	-0.0090	0.7146

Table B.1.: Regression coefficients for Jaccard Index, planned walk.

Sørensen Index	a_4	a_3	a_2	a_1	a_0
WiFi				-0.0003	0.1635
			2.61x10^{-6}	-0.0020	0.3725
		-1.46x10^{-8}	1.64x10^{-5}	-0.0056	0.5845
	7.04x10^{-11}	-1.01x10^{-7}	5.10x10^{-5}	-0.0105	0.7383
GSM				-0.0008	0.6359
			1.25x10^{-6}	-0.0016	0.7362
		-5.11x10^{-9}	6.07x10^{-6}	-0.0028	0.8101
	2.29x10^{-11}	-3.33x10^{-8}	1.73x10^{-5}	-0.0044	0.8602
WiFi/GSM				-0.0005	0.3306
			2.15x10^{-6}	-0.0019	0.5028
		-1.16x10^{-8}	1.31x10^{-5}	-0.0047	0.6714
	5.39x10^{-11}	-7.80x10^{-8}	3.96x10^{-5}	-0.0084	0.7893

Table B.2.: Regression coefficients for Sørensen Index, planned walk.

1_{st} Kulczynski Index	a_4	a_3	a_2	a_1	a_0
WiFi				-0.0005	0.2443
			3.79x10^{-6}	-0.003	0.5475
		-1.95x10^{-8}	2.21x10^{-5}	-0.0077	0.8293
	7.79x10^{-11}	-1.15x10^{-7}	6.04x10^{-5}	-0.0131	0.9996
GSM				-0.0029	1.8031
			5.59x10^{-6}	-0.0065	2.2501
		2.83x10^{-9}	2.93x10^{-6}	-0.0058	2.2092
	-1.28x10^{-10}	1.60x10^{-7}	-6.00x10^{-5}	0.0029	1.9292
WiFi/GSM				-0.0008	0.5111
			3.89x10^{-6}	-0.0034	0.8223
		-1.73x10^{-8}	2.02x10^{-5}	-0.0076	1.0732
	5.19x10^{-11}	-8.12x10^{-8}	4.57x10^{-5}	-0.0112	1.1866

Table B.3.: Regression coefficients for 1_{st} Kulczynski Index, planned walk.

2_{st} Kulczynski Index	a_4	a_3	a_2	a_1	a_0
WiFi				-0.0007	0.3421
			5.38x10^{-6}	-0.0042	0.7718
		-2.96x10^{-8}	3.32x10^{-5}	-0.0115	1.1999
	1.38x10^{-10}	-2.00x10^{-7}	0.0001	-0.0210	1.5029
GSM				-0.0016	1.2842
			2.46x10^{-6}	-0.0032	1.4811
		-1.01x10^{-8}	1.19x10^{-5}	-0.0056	1.6271
	4.50x10^{-11}	-6.55x10^{-8}	3.41x10^{-5}	-0.0087	1.7256
WiFi/GSM				-0.0010	0.6840
			4.26x10^{-6}	-0.0038	1.0243
		-2.29x10^{-8}	2.58x10^{-5}	-0.0094	1.3560
	1.057x10^{-10}	-1.53x10^{-7}	7.78x10^{-5}	-0.0167	1.5871

Table B.4.: Regression coefficients for 2_{st} Kulczynski Index, planned walk.

Tanimoto Coefficient	a_4	a_3	a_2	a_1	a_0
WiFi				-0.0003	0.1308
			2.26x10^{-6}	-0.0017	0.3120
		-1.37x10^{-8}	1.52x10^{-5}	-0.0051	0.5109
	7.28x10^{-11}	-1.03x10^{-7}	5.09x10^{-5}	-0.0101	0.6700
GSM				-0.0006	0.4666
			1.83x10^{-6}	-0.0018	0.6134
		-7.74x10^{-9}	9.12x10^{-6}	-0.0037	0.7251
	3.49x10^{-11}	-5.07x10^{-8}	2.62x10^{-5}	-0.0061	0.8015
WiFi/GSM				-0.0003	0.2162
			2.12x10^{-6}	-0.0017	0.3856
		-1.24x10^{-8}	1.38x10^{-5}	-0.0047	0.5649
	6.43x10^{-11}	-9.15x10^{-8}	4.54x10^{-5}	-0.0092	0.7055

Table B.5.: Regression coefficients for Tanimoto Coefficient, planned walk.

Cosine Similarity	a_4	a_3	a_2	a_1	a_0
WiFi				-0.0003	0.1648
			2.62x10^{-6}	-0.0020	0.3741
		-1.46x10^{-8}	1.64x10^{-5}	-0.0056	0.5856
	6.99x10^{-11}	-1.00x10^{-7}	5.07x10^{-5}	-0.0104	0.7384
GSM				-0.0007	0.6207
			1.37x10^{-6}	-0.0016	0.7304
		-5.44x10^{-9}	6.50x10^{-6}	-0.0030	0.8090
	2.33x10^{-11}	-3.41x10^{-8}	1.79x10^{-5}	-0.0046	0.8599
WiFi/GSM				-0.0005	0.3136
			2.21x10^{-6}	-0.0019	0.4897
		-1.18x10^{-8}	1.33x10^{-5}	-0.0048	0.6602
	5.59x10^{-11}	-8.06x10^{-8}	4.07x10^{-5}	-0.0086	0.7824

Table B.6.: Regression coefficients for Cosine Similarity, planned walk.

To have an approximated idea of the behaviour of the obtained and plotted data, the mean square error to each obtained regression was calculated, for all the data observed in each figure, and for the the sections below 100 meters and 200 meters. For the first order regression the results are shown in table B.7.

Similarity	Distance	WiFi	GSM	WiFi/Gsm
Jaccard	<100	0.3949	0.3097	0.3537
	>100	0.1412	0.2301	0.1510
Sørensen	<100	0.4099	0.2622	0.3364
	>100	0.1585	0.2608	0.1784
1_{st} Kulczynski	<100	0.7688	2.1149	0.8034
	>100	0.2713	1.1892	0.3452
2_{nd} Kulczynski	<100	0.8287	0.5207	0.6675
	>100	0.3264	0.5255	0.3629
Tanimoto	<100	0.3937	0.3377	0.3606
	>100	0.1402	0.2289	0.1492
Cosine	<100	0.4095	0.3121	0.3484
	>100	0.1590	0.2627	0.1777

Table B.7.: Mean Square Errors for a first order regression. Planned Walk

For the second order regressions.

Similarity	Distance	WiFi	GSM	WiFi/Gsm
Jaccard	100	0.3716	0.2995	0.3337
	>100	0.1612	0.2450	0.1703
Sørensen	100	0.3788	0.2576	0.3141
	>100	0.1816	0.2710	0.1980
1_{st} Kulczynski	100	0.7270	2.1030	0.7655
	>100	0.3011	1.2254	0.3760
2_{nd} Kulczynski	100	0.7630	0.5118	0.6241
	>100	0.3738	0.5453	0.4009
Tanimoto	100	0.3709	0.3260	0.3405
	>100	0.1599	0.2456	0.1702
Cosine	100	0.3782	0.3064	0.3254
	>100	0.1822	0.2743	0.2000

Table B.8.: Mean Square Errors for a second order regression. Planned Walk

For the third order regression:

Similarity	Distance	WiFi	GSM	WiFi/Gsm
Jaccard	100	0.3903	0.3114	0.3527
	>100	0.1828	0.2501	0.1873
Sørensen	100	0.3989	0.2661	0.3333
	>100	0.2043	0.2735	0.2137
1_{st} Kulczynski	100	0.7404	2.1012	0.7808
	>100	0.3280	1.2245	0.3956
2_{nd} Kulczynski	100	0.8033	0.5287	0.6620
	>100	0.4196	0.5499	0.4290
Tanimoto	100	0.3894	0.3387	0.3593
	>100	0.1814	0.2517	0.1880
Cosine	100	0.3981	0.3154	0.3445
	>100	0.2049	0.2773	0.2153

Table B.9.: Mean Square Errors for a third order regression. Planned Walk

For the fourth order regressions.

Similarity	Distance	WiFi	GSM	WiFi/Gsm
Jaccard	100	0.4192	0.3244	0.3793
	>100	0.1899	0.2499	0.1910
Sørensen	100	0.4279	0.2745	0.3573
	>100	0.2102	0.2728	0.2137
1_{st} Kulczynski	100	0.7618	2.0866	0.7956
	>100	0.3322	1.2320	0.3956
2_{nd} Kulczynski	100	0.8605	0.5451	0.7088
	>100	0.4306	0.5486	0.4316
Tanimoto	100	0.4182	0.3530	0.3865
	100	0.1885	0.2517	0.1919
Cosine	100	0.4268	0.3250	0.3698
	>100	0.2107	0.2767	0.2168

Table B.10.: Mean Square Errors for a fourth order regression. Planned Walk