Date of acceptance Grade

Instructor

Semi-supervised learning of WLAN radio maps

Teemu Pulkkinen

Helsinki December 13, 2010

UNIVERSITY OF HELSINKI
Department of Computer Science

HELSINGIN YLIOPISTO — HELSINGFORS UNIVERSITET — UNIVERSITY OF HELSINKI

Tiedekunta — Fakultet — Faculty Faculty of Science	Laitos — Institution — Department Department of Computer Science

Tekijä — Författare — Author
Teemu Pulkkinen

Työn nimi — Arbetets titel — Title
Semi-supervised learning of WLAN radio maps

Oppiaine — Läroämne — Subject
Computer Science

Työn laji — Arbetets art — Level M. Sc. Thesis	Aika — Datum — Month and year December 13, 2010	Sivumäärä — Sidoantal — Number of pages 54 pages

Tiivistelmä — Referat — Abstract

In this thesis a manifold learning method is applied to the problem of WLAN positioning and automatic radio map creation. Due to the nature of WLAN signal strength measurements, a signal map created from raw measurements results in non-linear distance relations between measurement points. These signal strength vectors reside in a high-dimensioned coordinate system. With the help of the so called *Isomap-algorithm* the dimensionality of this map can be reduced, and thus more easily processed. By embedding *position-labeled* strategic key points, we can automatically adjust the mapping to match the surveyed environment. The environment is thus learned in a *semi-supervised* way; gathering training points and embedding them in a two-dimensional manifold gives us a rough mapping of the measured environment. After a calibration phase, where the labeled key points in the training data are used to associate coordinates in the manifold representation with geographical locations, we can perform positioning using the adjusted map. This can be achieved through a traditional *supervised learning* process, which in our case is a simple nearest neighbors matching of a sampled signal strength vector.

We deployed this system in two locations in the Kumpula campus in Helsinki, Finland. Results indicate that positioning based on the learned radio map can achieve good accuracy, especially in hallways or other areas in the environment where the WLAN signal is constrained by obstacles such as walls.

ACM Computing Classification System (CCS):
C.2.1[Network Architecture and Design]:Wireless communication,
G.1.3[Numerical Linear Algebra]:Eigenvalues and eigenvectors (direct and iterative methods)
I.2.6[Learning]

Avainsanat — Nyckelord — Keywords
manifold learning, wlan positioning

Säilytyspaikka — Förvaringsställe — Where deposited
Kumpula Science Library, serial number C-

Muita tietoja — övriga uppgifter — Additional information

Contents

1 Introduction

The need for *indoor positioning systems* (IPS) arises from the failure of established technologies, such as GPS, to properly locate and track objects in an indoor environment. GPS signals tend to be weak when blocked by building walls and even when a position is triangulated the accuracy is not sufficient for indoor use. Triangulation using GSM base stations suffers from similar problems in addition to not divulging information as easily as other systems. Solutions for indoor positioning thus need to look for infrastructure elsewhere.

Several systems have been proposed that rely on the localized object carrying some kind of transceiver (RFID) or that sensors be built into the environment (IR). Recently, the interest in using WLAN access points for localization has grown greatly. This can be attributed to their wide use and distribution as well as the open standard which allows for requesting of signal strength information without separate authentication. In addition to widespread deployment of access points, wireless network cards are becoming a standard in personal mobile devices. This means that a person could be localized using software alone, with no need for purchase of separate devices.

By performing positioning using WLAN signals we can locate and track people and objects in an indoor environment. This could be used in a factory setting to automatically keep track of equipment without the need to set up auxiliary infrastructure. By being able to locate a shopper in a shopping mall, for instance, we could deliver location-sensitive information such as directions or points of interest.

In this thesis we present a new approach to WLAN positioning wherein a widely used *RSSI* (*Received Signal Strength Indication*)*-fingerprinting* system is augmented with non-linear pre-processing. This process converts a high dimensional *radio map*, where inter-point relations are assumed to be non-linear, into a mapping in a lower dimension. Once the conversion is done, established linear methods can be used to infer an objects position based on its unique RSSI fingerprint. The approach is semi-supervised as only a few labeled (correctly located) *key points* are needed in addition to the signal strength fingerprints to create a simile of the measured environment.

In section 2, we review the concept of *manifold learning*, including the specific non-linear approach used in this thesis. Section 3 surveys the different technologies used for indoor positioning, as well as a few implementations based on them. In section 4 the theoretical framework for empirical testing is laid out, including a presentation of the surveyed environments. Section 5 deals with how this method was implemented, and in Section 6 we present the results of deployment in a real-world setting. We draw conclusions from the results in section 7.

2 Manifold learning

Manifold learning is an umbrella term for a group of methods that purport to find the defining features of a set of data. In most cases this involves reducing the dimensions of the data to a more manageable level. The need for such methods arose from the *curse of dimensionality*; as the amount of data generated rose, so did the features it contained. These features can be equated to dimensions or variables in a data vector. Processing this vast amount of data usually requires a lot of time, most of which is wasted on features that do not vary significantly. By focusing efforts on the underlying source of variance (usually a fraction of the dimensionality in the original set), one can cut down on processing time with little to no loss in accuracy.

In the following sections we present a few methods for dimensionality reduction, building up to the main focus of this thesis: the *Isomap*-algorithm.

2.1 Linear dimensionality reduction

Linear dimensionality reduction assumes a linear relation between points in the data set. As long as the points in the set of data vary linearly, a lower-dimension representation of high-dimensional data can usually be found. Many of the linear methods rely on reducing the data to the vectors that account for the majority of the variance.

2.1.1 PCA

A very popular algorithm for linear dimensionality reduction is PCA (*Principal Component Analysis*), first introduced in [Pea01]. It is based on finding the most significant components that make up a set of data, and does this by finding the top eigenvectors from the covariance matrix of the data. These eigenvectors are ranked based on the amount of variance they explain. The vector that accounts for the highest variance is named the first principal component. One can enumerate components up to the original dimension of the data, though a set limit is often imposed with the goal of capturing the most important features without unduly inflating the representation. Due to its simplistic nature PCA is rarely used as an end-all algorithm on its own; it's usually a part of pre-processing (also known as *whitening*). By rebuilding the data using the significant principal components, one can reduce the dimensions to a given extent. It is up to each implementation to decide a balance between reducing the dimensions and keeping a good level of accuracy.

In the following example, PCA is performed on a sample set of handwritten digits from the MNIST database (a in Figure 1). Since the digits are represented by images of size 28x28, their apparent dimensionality is 784. Before PCA is performed on images, it is customary to remove the DC component (1b) (*Direct current*, considered synonymous with "constant" in electric engineering terms) since it is not considered to contain interesting information [HHH09]. This essentially means dividing the images by their variance to achieve unit norm and subtracting the mean to give them a mean of zero. Next (1c), the image is reduced to its two principal components and rebuilt. Clearly, these two do not carry enough of the variation to properly represent the original images. Using 16 components (1d) results in a better depiction but is still quite ambiguous. In the end it takes 64 components (dimensions) in 1e to convey the content of the original image. The dimensions of the image have thus been reduced from 784 to just 64, though some accuracy has been lost.

2.1.2 MDS

Multidimensional scaling (in this context specifically *metric* MDS, in that Euclidean distances are used) works by creating a mapping of given data on a lower dimension while trying to keep pairwise distances between points intact. The data it uses as input is a dissimilarity matrix, X, of the points in the data. If the dissimilarity is the Euclidean distance between the points, MDS can be said to minimize the quantity [Gho06]

$$min_Y \sum_{i=1}^{t}\sum_{i=1}^{t}(d_X(i,j) - d_Y(i,j))^2, \qquad (1)$$

where d_{ij} is the Euclidean distance between points i and j in the respective matrices: $d_X(i,j) = |x_i - x_j|^2$. That is, MDS tries to find the dimension where the pairwise distance between points is as close as possible to the given distance matrix (X). The approximation is performed through an eigenvalue decomposition of a *double-*

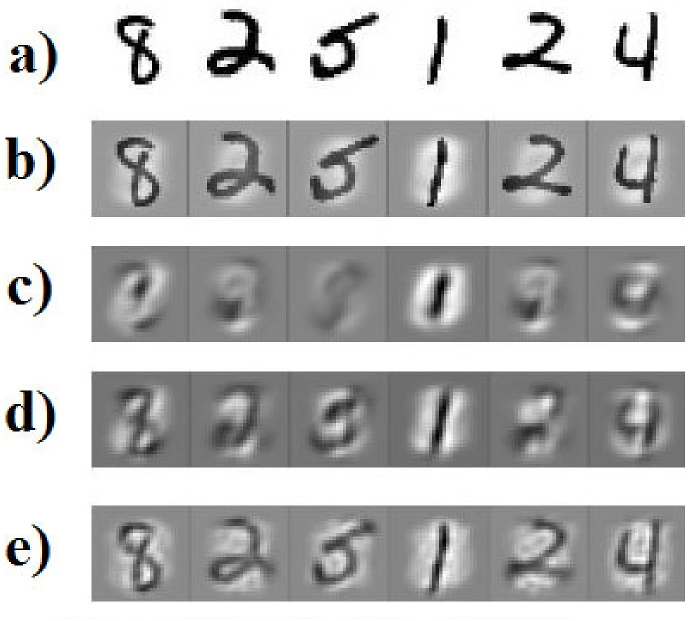

Figure 1: Dimensionality reduction of handwritten digits using PCA.

centered version of the distance matrix. Double-centering entails multiplying the squared distance matrix with $-\frac{1}{2}$ and a *centering matrix* from both sides:

$$\hat{X} = -\frac{1}{2}HXH, \tag{2}$$

where

$$H = I - \frac{1}{n}ee^T, \tag{3}$$

and I is an identity matrix of size n, e is an $[n \times n]$ matrix of ones and n is the size of the distance matrix. By multiplying the distance matrix from both sides we are subtracting the mean from both the rows and columns. We are thus essentially performing the same *whitening* process as we did in the PCA example earlier. By performing an eigenvalue decomposition on this centered distance matrix we can then find the final coordinates for the scaled mapping through

$$Y = \sqrt{\Lambda}V, \tag{4}$$

where Y is a matrix of coordinates, Λ are the eigenvalues and V is the eigenvector. This solution is very similar to that of PCA, except the dissimilarities in classic MDS aren't restrained to just Euclidean distances.

An example of how MDS could be used is to recreate a map of geographical locations based solely on the distances between cities on that map. In Figure 2b, Cox and Cox present an example of a map of British cities created based on the distances between them [CoC01]. Due to the nature of MDS, there is no guarantee that the resulting mapping is geometrically correct. A solution where the cities are translated, rotated or reflected is equally valid. MDS can only guarantee a relative mapping. We have thus chosen to rotate and mirror the mapping for illustrative purposes.

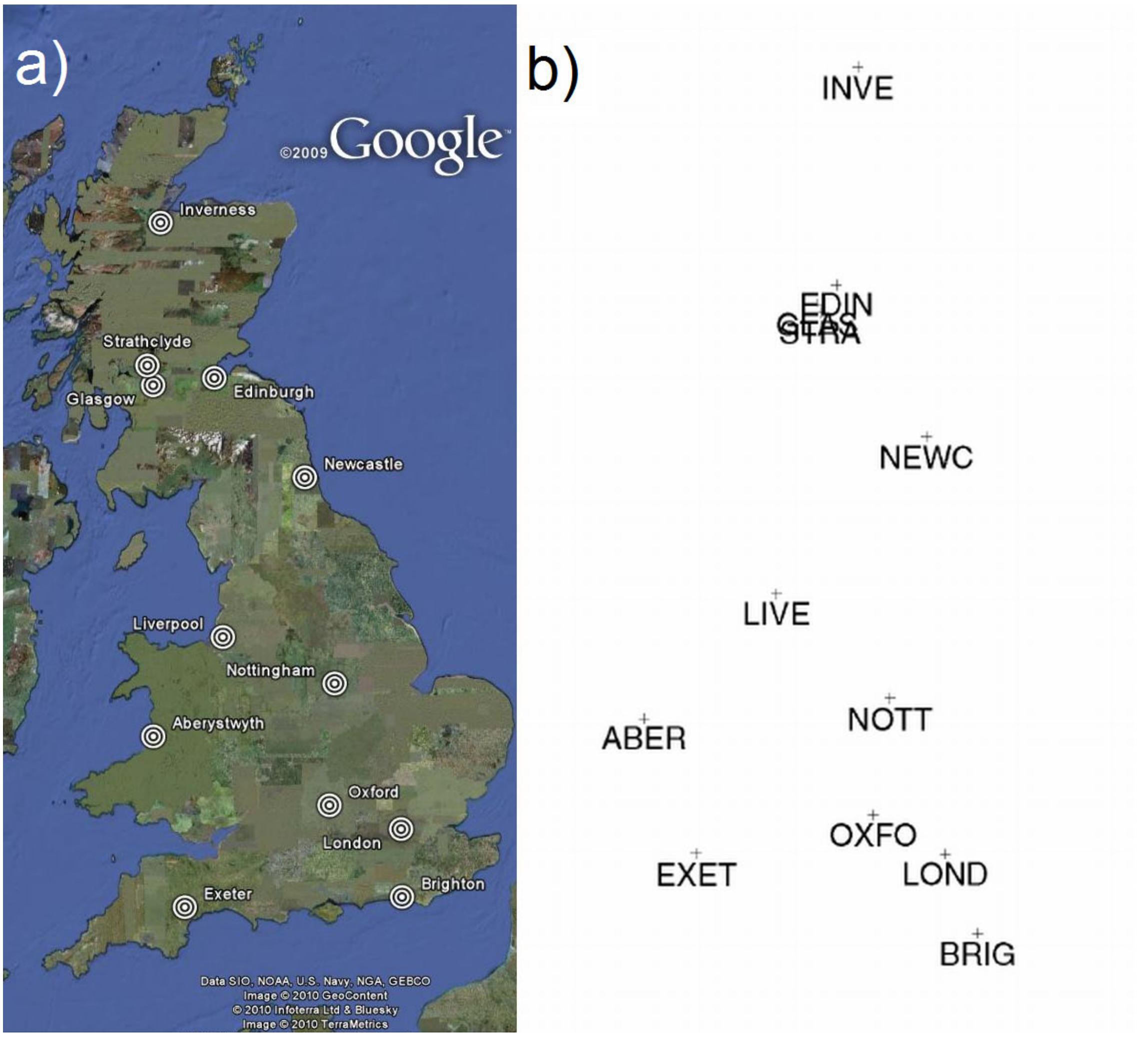

Figure 2: MDS mapping of cities based on the relative distances between them. a)Actual locations b)MDS mapping after transformations (modified from [CoC01])

2.2 Non-linear dimensionality reduction

When inter-point distances in the data set no longer adhere to linear relations, methods like PCA that assume a linear base break down. In a non-linear space

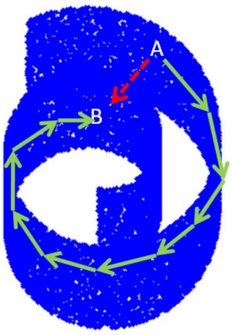

Figure 3: Swissroll manifold with the path between A to B interpreted by a linear (red dotted arrow) and a non-linear (green solid arrows) method.

Euclidean distances are no longer valid dissimilarity measures since any method relying on them will inevitably create shortcuts to deceptively close parts of the manifold. Figure 3 highlights this problem. A linear method would not "respect" the manifold since it is not aware of the embedded structure in the data. Methods that can extract features out of non-linear relations are thus needed. A common feature in non-linear methods is the creation of local neighborhoods where it is assumed linear relations still apply. By combining these neighborhoods a global view of the manifold can be obtained and placed in a lower dimension.

2.2.1 LLE

LLE (*Locally Linear Embedding*), like the linear methods mentioned before also relies on the extraction and manipulation of eigenvectors. Its main goal is to make sure points that are nearby in the original dataset remain close to each other in the reduced space. In addition, the relative positions of the points should remain as similar as possible. It does this by calculating a set neighborhood of each point and then computing the weights (W) that best reconstruct the point using its neighbors. The fitness of the reconstruction is measured by [SaR03]

$$\varepsilon(W) = \sum_i (X_i - \sum_j W_{ij} X_j)^2, \tag{5}$$

where W_{ij} signifies the contribution of data point j to the reconstruction of data point i. Once the neighborhood-preserving mapping is done, the high-dimensional input X_i is mapped to a low-dimensional output Y_i by choosing the coordinates of

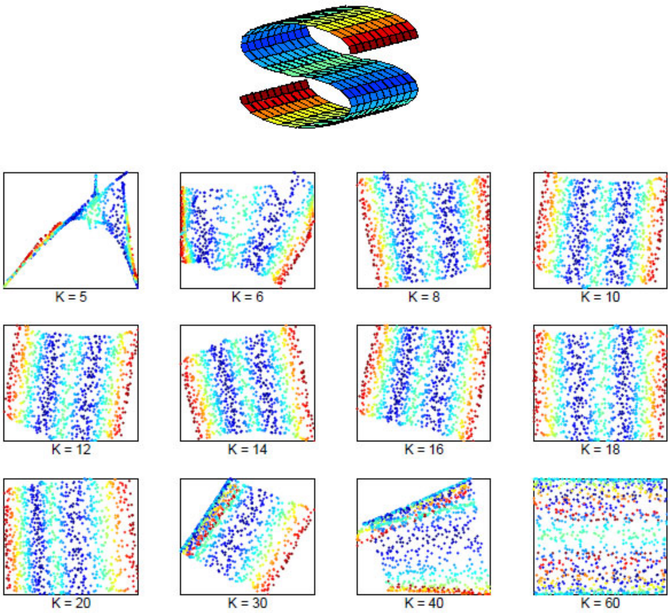

Figure 4: Sensitivity of LLE algorithm to neighborhood size. Modified from [SaR03]

Y_i to minimize

$$\theta(Y) = \sum_i (Y_i - \sum_j W_{ij} Y_j)^2, \tag{6}$$

i.e. the embedding cost function. In their article Saul and Roweis state this can be done by solving a "sparse NxN eigenvalue problem". They show that the resultant embedding coordinates are received from the bottom d (dimensionality) non-zero eigenvectors.

LLE only has one free parameter in the neighborhood size, K. As will become apparent later on, the selection of the neighborhood parameter is delicate and subject to model selection. As can be seen in Figure 4 the two-dimensional S-manifold can be recovered from its three-dimensional representation with several values of K. Too large or small a value will distort the local neighborhood, however, and the embedding is corrupted.

2.2.2 Isomap

Whereas LLE focuses on preserving the local neighborhood relations, Isomap strives for a more global view. In other words, LLE tries to ensure that the local geometry remains linear when embedded in the lower dimensional space, whereas Isomap works to ensure that the distances between distant points remain similar as well.

Since the main focus of this thesis is the implementation and use of Isomap, the algorithm will be presented in finer detail than the preceding approaches.

Isomap was first introduced by Tenenbaum et al. as an aid for what seems to be a popular target for manifold learning algorithms; modeling human vision [TSL00]. The intuition here seems to be that humans are exceptionally adept at reducing high-dimensional visual input to its barest components: the degrees of freedom underlying the seemingly complex imagery.

Isomap can be seen as a continuation of the classical (metric) MDS in that it acts as its non-linear extension. As MDS mostly tries to preserve the pairwise Euclidean distances it cannot function as such in a space with nonlinear relations. Isomap's main contribution is to turn a given dissimilarity matrix into a distance matrix that can be handled by an established MDS implementation.

The algorithm begins by determining the local neighborhood using the given distance matrix as input (X). Points are deemed neighbors based on their relative distances, $d_X(i, j)$, in the matrix. This is done by either selecting a set, K, of nearest neighbors or by defining a fixed radius ϵ from which to choose neighbors. For the purposes of this thesis only K-NN is considered in the future. Once the neighbors are defined the relations are represented as a weighted graph G, where the weights for the edges are defined by the distance between the neighbors.

Next, the geodesic distance between all pairs of points on the manifold, $d_M(i, j)$, are estimated by computing the shortest paths between them in the graph, $d_G(i, j)$. This shortest path calculation could be performed by any established algorithm, but is usually assumed to be Dijkstra's algorithm.

At this point Isomap has essentially turned the given distance matrix into one where linear relations are meaningful. It can thus be turned over to the classic MDS, described earlier. In the simplest implementation Isomap is then allowed to run until the dimensionality is found where the output Y results in the smallest error. In other words, starting from the lowest dimension possible the dimension is increased until the error no longer decreases significantly. A classic example of a manifold where linear methods like PCA and MDS fail is the so called "Swiss roll", the structure displayed in an earlier example. It serves as a good example because linear methods tend to take shortcuts from one part of the manifold to another, thus distorting the resultant embedding beyond use. In Figure 5 Isomap is allowed to run on the sampled manifold. To the human eye it seems somewhat obvious that the three-dimensional figure is simply a plane curled upon itself. Isomap correctly manages to "unfurl" this manifold into its two-dimensional representation of a plane.

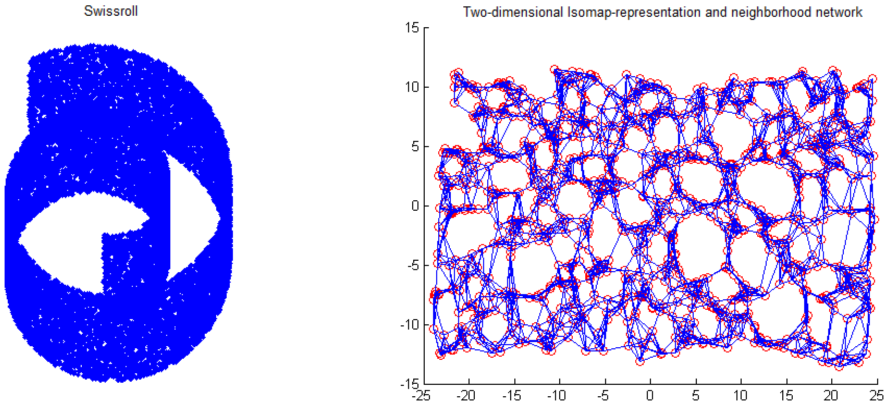

Figure 5: A representation of the "Swiss roll" manifold in the third dimension and resulting Isomap-representation

Isomap, like its linear and non-linear counterparts, is not without its weaknesses. While resulting in a better global view of the manifold, Isomap cannot match LLE's local accuracy. Some areas of the resultant embedding might be distorted locally. Like LLE it only has one free parameter: the neighborhood size K. This parameter needs to be chosen carefully so as to not misrepresent the underlying manifold. Too high a value will result in shortcuts between parts of the manifold. If the parameter is chosen to be too small, however, one runs the risk of creating isolated neighborhood "islands" with poor connectivity.

Since Isomap has only been proven to be "arbitrarily accurate in the limit of infinite data"[TSL00], the input data cannot be too sparse. This carries the same potential pitfalls as choosing the wrong neighborhood parameter. The choice of K is also dependent on the amount of noise in the data. With increasing noise the neighborhood size needs to be enlarged.

Finding the optimal neighborhood parameter has been the focus of some research [CHC07],[SMR06]. Chao et al. suggest constructing the minimal connected neighborhood graph and calculating the cost of it (breadth-first shortest distance). This would sidestep the entire MDS step of the original algorithm until a suitable K has been found. Samko et al. have a similar approach, but restrict the choice of K to a specific interval.

Isomap in its default form is also not equipped to handle curved manifolds, such as that of a "fishbowl". Tenenbaum and de Silva have created an extension for this purpose, called C-Isomap [DeT02]. They overcome Isomap's restrictions by a balance of assumptions. By requiring denser sampling, certain assumptions on the mapping can be loosened. C-Isomap works by weighting the edges by the average distance to a points' neighbors. Roughly speaking, it "shrinks sparse regions and magnifies dense regions". It carries the caveat that it no longer works as well in situations the original Isomap can handle. The results from their tests can be seen in Figure 6

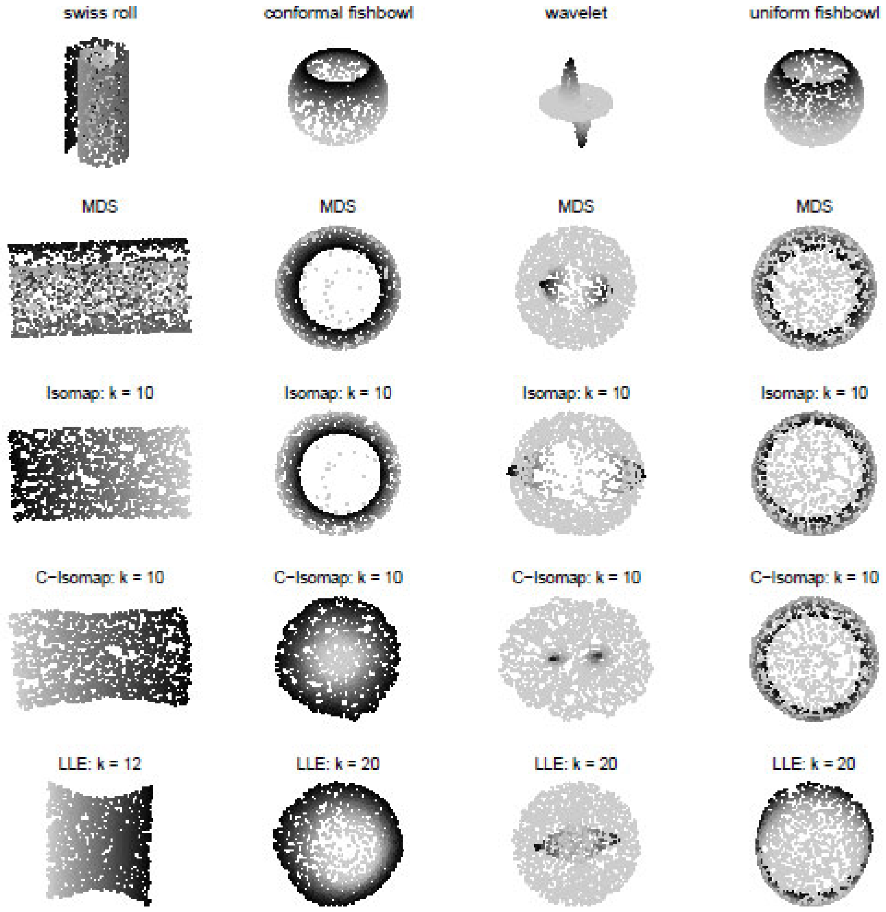

Figure 6: Algorithm behaviour on special manifolds. Modified from [DeT02]

Because measurements based on WLAN signal strengths are thought to be non-linear, but assumed to be part of a two-dimensional manifold, a mapping algorithm like Isomap could be used to discern their placement on the manifold. The specifics of this approach are discussed in greater detail in Sections 4 and 5.

3 Positioning

Positioning methods can roughly be separated into two categories: methods that require infrastructure be built and setup for them, and ubiquitous methods that rely on readily available stations. RFID, IR and WLAN systems represent both ends of this spectrum.

Since common methods for outdoor positioning, including GPS and GSM tower triangulation, due to various reasons are unfit for localization indoors the following sections will deal exclusively with the different ways positioning has been realized within the confines of a building.

3.1 WLAN

In the following sections the WLAN approach to location estimation is presented. In the first section the standard is defined, in addition to a short survey on the nature of Wi-Fi signals. The second section describes different approaches to using WLAN access points as a means to user localization.

3.1.1 Standards and technology

WLAN(*Wireless Local Area Network*) offers the same capabilities as the traditional Ethernet cable connection using wireless signals. Devices wishing to use the network need to conform to the 802.11 WLAN standard (also known "Wi-Fi", *Wireless Fidelity*), regulated by the Wi-Fi Alliance [HBO05]. Later amendments to the standard have mostly improved throughput; the latest standard of 802.11n being capable of handling transmission rates up to 600Mbit/s.

The network works within two unlicensed frequency bands: 2,4 GHz and 5,7GHz. These frequencies are open to the public and do not require licenses to use. Wireless devices are associated with an AP (*Access Point*) that is usually connected to an outer network through a wired connection. Devices in the network are assigned an SSID (*Service Set Identity*), which is a 32-character unique identifier.

For the purposes of positioning the most interesting aspect of the WLAN network is the RSSI devices can detect from the AP(s) they are connected to. RSSI has no standardized format and is thus dependent on the wireless network card and drivers of the device registering them. The most common format of signal strength is dBm (the power ratio in decibels), usually detected in the range of 0...-100dBm, depending on the distance to the AP. Some interfaces also give an arbitrary value between 0,...,255, with no set unit. In an ideal environment signal strength values could be said to indicate the distance a wireless device is from an AP. Due to the multi-path nature of signals (bouncing off walls, reflected off surfaces) and varying wireless card quality this can not be assumed in a real world environment. For the purposes of positioning, however, it is enough that the collected signals differ distinctly in different parts of the measuring environment.

Operating in the unlicensed 2,4GHz frequency band has its problems: other devices use the same frequency. This interference is not limited to cordless telephones and Bluetooth devices; even microwaves can leak radio signals in this frequency. Since this interference is intermittent and usually does not follow a set pattern any testing done in this band requires a broad sampling range under a long period of time. 2,4GHz is also the resonating frequency of water and since human bodies are made up -of up to 70% water, any human presence during measurements is likely to cause distortion. In addition, systems employed indoors suffer from the above-mentioned multi-path problem in that signals are reflected off walls and thus give distorted measurements.

Several studies have been made on the behavior of a wireless signal in an indoor environment, and factors that affect the measurement of RSSI. Farivar et al. study how signals behave indoors, and what causes anomalies [FWC05]. As would seem logical, a higher signal strength results in less error in location estimation. Mainly, trying to do location estimation using weak signals alone will more than likely cause significant anomalies in the position determination. Walls and objects obscuring a direct line-of-sight (*LOS*) also affect measurements.

Farivar et al. suggest two metrics for determining the quality of the environment the measurements are recorded in. NSD (*Neighborhood Signal Distance*) calculates the median of the Euclidean distances of all the sampled points to the center sample. This gives the neighborhood an abstract value that could be regarded as a performance metric. ANSD (*All Neighbors Signal Distance*) is made up of a distribution histogram that details the distribution of the signal distance in a neighborhood between every two points. In well-behaving neighborhoods ANSD tends to center around a specific value.

Kaemarungsi and Krishnamurthy have done a statistical survey on the behavior of RSSI depending on certain variables, including user presence, orientation and variation between APs [KaK04]. User presence seems to cause the standard deviation to rise, though the mean remains around the same value (Figure 7). This mainly suggests that in the learning process the training and testing environments should reflect one another. That is, if the purpose of the localization is to have the user hold the wireless device during positioning, the measurements for the training should also be recorded with a user present.

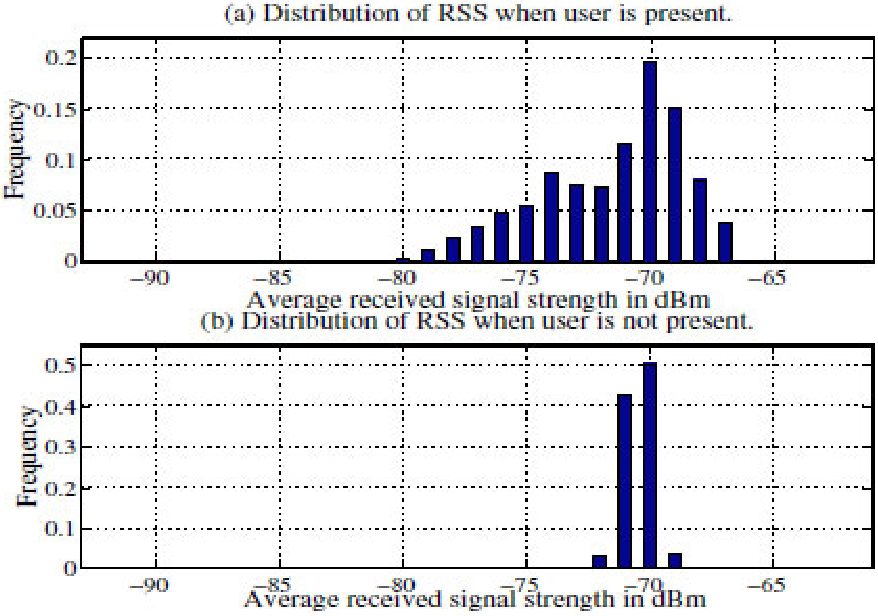

Figure 7: Effect of user presence on signal distribution. From [KaK04]

The direction the user is facing when recording measurements when the device has a LOS to the AP can cause a difference of about 10 dBm in the signal strength. When the device has no LOS the user blocking the signal can completely remove the AP from the devices vision. According to the article, the distribution of the RSSI tends to be left-skewed, in that most of the variation happens below the maximal strength value [KaK04]. An interesting observation is also that the variation tends to increase with the strength of the measured signal, meaning weaker signal strengths tend to fluctuate less.

Kaemarungsi and Krishnamurthy also recorded signal strength values over two separate time periods, to measure the stability of measurements. According to their tests, the signal seems to be stationary within a small scale (within an hour) but can change during a longer time period (a day). They also found that RSSI from different APs are not correlated, even though they operate on the same channel.

This is attributed to the 802.11 standard that is equipped to handle competing signals. Finally, the authors suggest that even two APs cause a unique enough signal pattern that they could be used for rough positioning. Naturally, using more APs results in better accuracy.

3.1.2 Methods of positioning

Wireless positioning using WLAN could further be divided into methods that record a signal strength *fingerprint* to create a database or a radio map, and systems that strive to model the signal propagation itself. With a recorded radio map, positioning is reduced to a simple supervised learning problem where given signal strength fingerprints are matched to those in the database. Most systems designed around modeling the signal space rely on some form of Bayesian modeling. If the behavior of the signal is known several established methods of positioning could be used to localize a wireless device. These could include AOA (*Angle of Arrival*), TOA and TDOA (*Time of Arrival* and *Time Difference of Arrival*) or RSP (*Received Signal Phase*) [PLY00]. One of the pioneering efforts in WLAN positioning is the RADAR system, designed by Bahl and Padmanabhan [BaP00]. RADAR is built around 3 base stations with one mobile wireless station. The system records RSSI, the physical coordinates and the direction the user is standing in when recording the measurements. The actual training and testing process adheres to the fingerprint method mentioned earlier, i.e. vectors of information are stored at every training location. In the testing phase a vector is recorded and then compared to the stored ones based on the Euclidean distance between them. Using this method they achieve an accuracy of about 3 m in a space of size 22.5x43.5 m.

In addition to the fingerprinting method Bahl and Padmanabhan also present a theoretical propagation scheme that maps probable signal strengths based on the room layout. This model gains an accuracy of about 8 m. Tests using a mobile user also give good results. The authors suggest improvements in the form of user profiles that could fine tune the algorithm to specific habits and types of movement.

Roos et al. implemented WLAN positioning based on Baeysian probabilities and compared its performance to the nearest-neighbor approach of the RADAR system [RMT02]. At the base of the algorithm is the estimation of the *posterior distribution* of the location:

$$p(l|o) = \frac{p(o|l)p(l)}{p(o)},$$

where $p(l)$ is the prior probability for location l and $p(o)$ is the likelihood of the observation. By attaching a *likelihood function* to determine $p(o|l)$, the probability of the observation given the location where it was recorded, the most likely location for the given observation, $p(l|o)$, can be calculated. In their testing, both a kernel method and a histogram method were implemented as the practical likelihood estimator. The nearest-neighbor approach then served as a form of control.

Through testing these methods they found that the resulting accuracy varied based on both the amount of test points recorded (or history used for smoothing) and the amount of APs used for fingerprint recording. By adjusting these parameters one could either cover for a weakly covered area by recording more data, or counteract a sparsely recorded environment by installing more APs.

In a testing area of size 16x40m they reached an accuracy of up to 1.56 m, using the histogram method as the likelihood function. When less test observations were used in estimation, in this case only one, the kernel method proved the most accurate at 2.57 m. Though the nearest-neighbor approach suffered in this shorter testing length, managing only a 3.71 m accuracy, it fared comparatively well when 20 test observations were considered, with an accuracy of 1.69 m.

Yeung et al. also extend the RADAR system by measuring the RSSI not only from the AP, but also from the mobile device [YZN07]. This is based on the intuition that even with the same output strengths these two measurements report different RSSI. This would then add another level of distinction to the recorded fingerprint. Though their method proves effective, the authors admit that recording the RSSI from the mobile device is not as accurate as the one received from the AP. They illustrate this with a radio map comparison in another version of their article [YeN07] (Figure 8).

Their method has two variants. One records the composed distance from the uplink and downlink data using the Mahalanobis distance, which is an extension of the Euclidean distance that considers the correlation between data sets. The other method assumes the signal strengths follow multiple normal distributions and are independent. By using the Maximum Likelihood Estimator (MLE), they receive the suggested location. Using these enhancements they record an accuracy upgrade of up to 30%. The probability estimator is more accurate but also more computationally expensive.

In accordance with the discovery in section 3.1.1. that variation actually increases in relation to the strength of the signal, Zhang and Zhang present another positioning

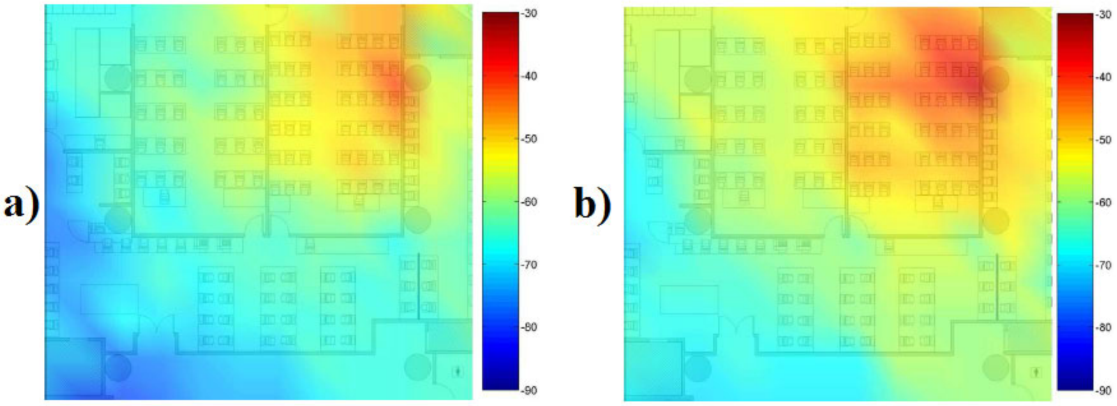

Figure 8: Recorded radio map from the point of view of the a) AP b) mobile device. Modified from [YeN07]

system based on including distant APs in the localization algorithm [ZhZ07]. In addition they employ a maximum matching algorithm that limits the possible choices of positions to the most likely ones right away. This speeds up the localization process. In an environment of size 100x20m and a grid size of 5x2 m they achieve an accuracy of 5 m. Their conclusion is that if possible, all the APs available should be included in the training, not only APs in the immediate environment. Focusing on just the strongest signals does not achieve the same level of accuracy and has negligible improvements on processing speeds.

Papapostolou and Chaouchi's WIFE (*Wireless Indoor positioning based on Fingerprint Evaluation*) system is another fingerprint system whose main contribution is the handling of fingerprints that look alike [PaC09]. Because of signal strength attenuation and the multi-path nature of the signal, certain fingerprints can look similar although they've been collected from two distinct sources. The authors call this effect *aliasing*. To combat this Papapostolou and Chaouchi only consider fingerprints that are physically close to the last determined location.

The WIFE system employs a compass with 8 directions as well as filtering zero-valued signal strength measurements. By doing this accuracy is increased from traditional systems; 3.5 m to less than two meters. Through empirical testing it is discovered that although adding APs to the learning process improves accuracy, this improvement has a limit after which results actually worsen (Figure 9). The authors attribute this to the aliasing effect.

As a more probabilistic view of localization Youssef et al. present a system based

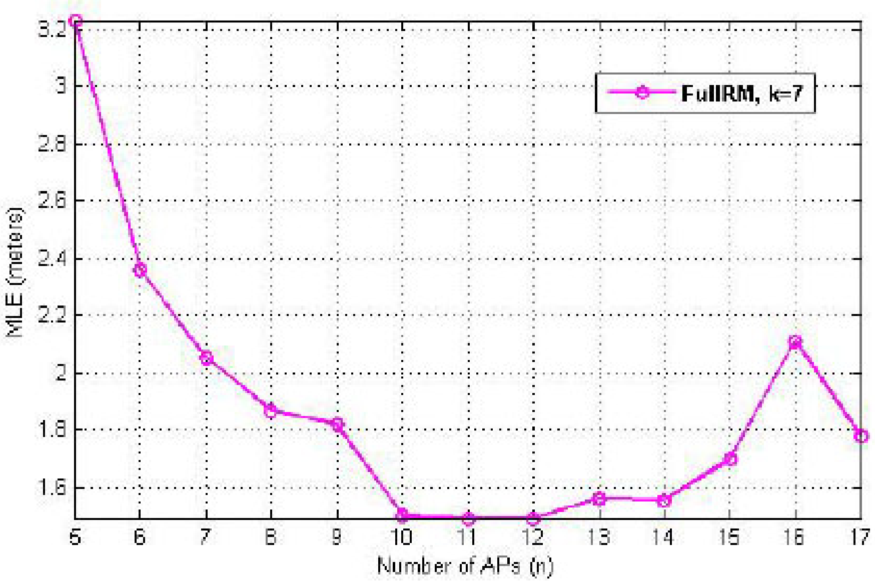

Figure 9: The aliasing effect causes fingerprints to look alike when a certain number of APs are added. From [PaC09]

on MLE, which determines a location based on what set of access points best correspond to the sampled distribution [YAS03]. Locations are clustered based on the access points that cover them. The algorithm strives to find the optimum cluster size (number of APs) and amount. A lot of the work goes to finding a trade-off between the size of the observation vector, the nature of the clusters and computation requirements. On a floor of size 25x70m they reach an accuracy of about two meters.

Youssef and Agrawala later employed this clustering technique in the Horus WLAN location determination system [YoM05]. In addition to the joint clustering technique, whose main goal is to lower the computational requirements of the system, Horus is also designed to handle large- and small-scale variations in signal strength. Using a probability measure it matches a recorded signal to a radio map. Youssef and Agrawala suggests that some of the implemented enhancements could be used independently in other positioning systems.

Finally, Ferris et al. use Gaussian processes (GPs) for location estimation based on RSSI [FHF06]. GPs are said to be "parametric models that estimate Gaussian distributions over functions based on training data". The authors motivate their use by explaining that they don't need to be calibrated in advance, they can approximate a wide range of models, incorporate uncertainty estimates and estimate parameters from the training data. Their main strength thus lies in estimating likelihoods for spots in the environment that have no training data. The algorithm itself involves creating a model of the signal strength measurements and building a graph of the hallway environment where they implement it. In this graph the edges are represented by straight hallways and rooms are represented by vertices. Even when moving between floors and rooms the system achieves an accuracy of up to 1.69 m (Figure 10). The complete system also incorporates an outdoor localization component which is beyond the scope of this thesis.

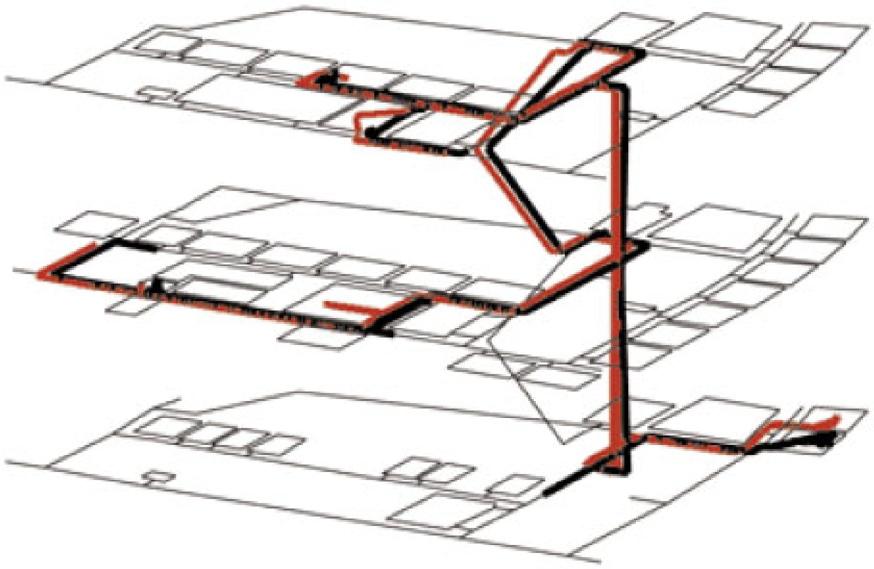

Figure 10: Test run of GP system. Actual route in red, most likely estimate in black. From [FHF06]

3.2 Alternative approaches

Radio-frequency identification (*RFID*) involves a system of tags and readers and the communication between them. Tags are classified as either active or passive, depending on if they are able to send out signals independently. Passive tags have no battery and can thus be installed and operated without replacement for a longer period of time. The tag is energized by the tag reader using inductive coupling between two coils in the tag. This energy is stored until it is sufficient to transmit its ID on a modulated frequency [WFG99]. The main field of use for passive tags has been in replacing the traditional bar codes on products. Because of their passive nature, however, they have a very limited range and are thus hard to implement in a positioning scenario. Active tags contain both a transceiver and a battery to power it. This affords them a greater range, but also makes them heavier, more expensive and perishable. Despite their volatile nature they can operate several years, and can thus be considered for positioning purposes.

In the LANDMARC (*Location Identification based on Dynamic Active RFID Calibration*) system Lionel et al. employ a system of active RFID tags and readers for the purpose of indoor localization (Figure 11) [NLL04]. Like can be seen in WLAN systems presented earlier, they also have to combat the multi-path problem and lack of line-of-sight. Though the authors manage to achieve of accuracy up to one meter (in an environment of size 4x9 m), the setup required optimal placing and density of tags. In addition, the positioning is relatively slow. The authors mention three factors in particular that render the system unfeasible using the current standard of RFID technology. A lacking standard of signal strength reporting means LANDMARC has to scan power levels itself, and a lot of time is wasted on this. In addition, because of the above signal strength scanning and the fact that tags have a set delay between transmissions a long latency is incurred. Finally, the tags do not behave in a uniform way. Differences in circuits and batteries result in different readings in the same location.

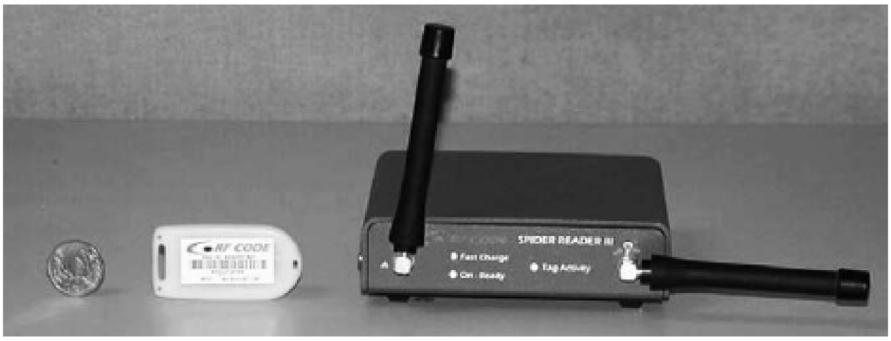

Figure 11: LANDMARC system infrastructure components. From [NLL04]

Lim and Zhang combat the problem of varying signal strengths by deploying passive tags in a uniform grid on the ceiling of an industrial warehouse [LiZ06]. Instead of recording signal strengths, a mobile device moving from grid to grid records the tags it can detect. In testing certain pattern matching algorithms are then used to detect in which of the trained grids the device is currently located. Lim and Zhang narrow the searching down to two algorithms. The "Intersection over Union"- algorithm quantifies the similarities between the training and testing set as the quotient of their common elements over the unified elements. The "Tag-to-Location Mapping Count" version scores a testing set based on how many of the training points it maps.

Through this process the authors report an accuracy of more than 97% with an error of less than one meter. In addition, they receive good results in orientation identification. Since the solution uses passive tags it also incurs lesser costs than with an active solution and is robust against interference.

Petrelli et al. augment a previously deployed *infrared network* to include orientation identification [PKA07]. Their original system depends on installing a pair of infrared transmitting devices that transmit separate sets of patterns, with varying quality. When a pair of receivers are attached to the side of a traceable target, it can be localized depending on the patterns it is detecting. When moving through grids on a plane, the receivers record the *success rate* of detected patterns. The success rate is defined as the number of received patterns of a certain type compared to the expected patterns in a set interval. The set of success rates define a unique identity, which can then be compared to training sets.

The system fails, however, when orientation is changed between measuring points. If the target rotates while remaining otherwise stationary, the patterns recorded are distorted and can no longer be matched to recorded ones. The authors amend this system by adding a third receiver onto the target which helps detect orientation.

4 The Semi-supervised Approach

Though WLAN fingerprinting approaches via different means have achieved relatively good accuracy, as detailed above, they have always suffered from the amount of effort needed to implement them. To cover an environment fingerprints need to be recorded in every location to be mapped. Since the radio map created through this effort needs to be tied to real-world coordinates in order to be useful, the calibration effort includes the recording of the location of every fingerprint. This invariably requires human presence for the entirety of the training run as well as accurate location information for every measured location. Several approaches have been implemented to combat this time-consuming process. The commercially available Ekahau Real-Time Location System [EKA] depends on reducing the environment into a set of rails and areas that fingerprints are constrained to. In addition, the calibration process is eased somewhat by interpolating signal strength measurements between calibrated points. By abstracting locations from specific coordinate points to larger entities, like rooms or hallways, some approaches have turned the positioning process into one of classification [HFL04]. By decreasing granularity, fingerprints are allowed to represent a larger area, improving the robustness of localization.

In the following we present a method for semi-supervised RSSI-fingerprint localization, which removes the need to know the precise location of most of the recorded fingerprints during the training process. By letting a few carefully selected *key points* represent regions of fingerprints, we can calibrate an entire radio map based on the real-world coordinates of only a fraction of the fingerprints.

The presented approach is based on the assumption that distances between fingerprints in the signal space are smooth yet non-linear. By applying the previously presented Isomap-algorithm to a distance matrix created from the pairwise distances between fingerprints, we can map the fingerprints into a two-dimensional manifold representation. Since the main result of the learning is a radio map, the positioning algorithm itself is interchangeable. In our approach we used a simple k-NN comparison, as a proof of concept.

Following examples from literature, we present the localization process in two sections: *training*, during which the environment is mapped and calibrated, and *testing* where signal fingerprints are sampled and compared to gauge the accuracy of this approach as a positioning method.

4.1 Training procedure

The approach in this thesis is based on the similarity of RSSI fingerprints. A *fingerprint* in this context is defined as a vector, s, of RSSI measurements. The length of the vector, p, is defined as the number of APs detected in the current environment. In the example environment of Figure 12, the fingerprint for measurement location A could be described as

$$\vec{s}_i = (s_{i1}, s_{i2}, \ldots, s_{ip}), \tag{7}$$

where i is the index of the measurement point. In this particular example, s_{i1} and s_{i2} would most likely give the most relevant information.

Since signal strengths tend to be volatile, it is best to use as a fingerprint an aggregated measurement of several RSSI values measured over a period of time. We thus chose to average the measured fingerprints for a particular location, in addition to performing some further preprocessing which we discuss in section 5.1. As has

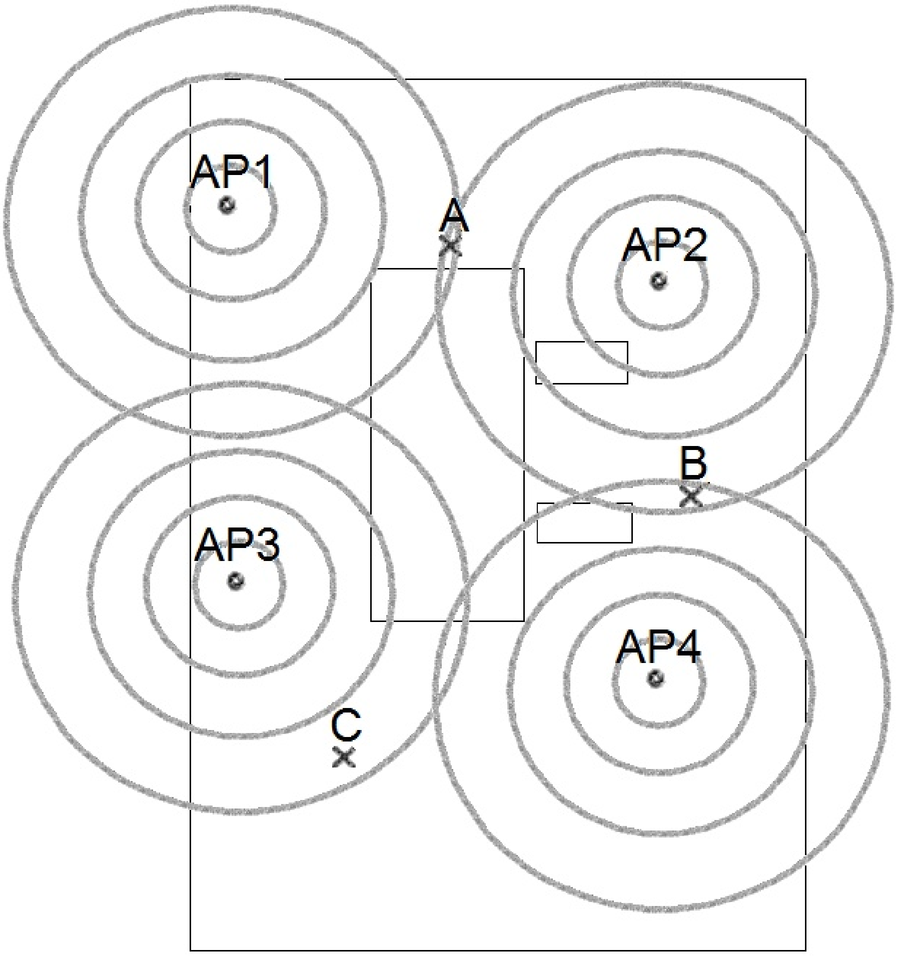

Figure 12: Example environment layout for RSSI measurements with hypothetical AP/measurement location placement

been shown earlier, RSSI tend to be relatively stable over a short period of time [KaK04]. Due to the changing environment (people moving around), attenuation and multi-path traversal some APs might disappear from the measurements entirely. The solution used for this testing was to assign these APs minimal strength values, so as to not corrupt the algorithms handling them. The actual measurements were performed in a rough grid covering the testing environment, resulting in a list of fingerprint vectors of size $[p \times n]$, where n is the number of measurement locations:

$$M = \begin{pmatrix} s_{11} & s_{12} & s_{13} & \dots & s_{1p} \\ s_{21} & \dots & \dots & \dots & \dots \\ s_{31} & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots \\ s_{n1} & s_{n2} & s_{n3} & \dots & s_{np} \end{pmatrix},$$

where each row represents one fingerprint, as seen in (7).

This in itself could be the focus of a supervised learning experiment, where sampled fingerprints in the environment are compared to matrix M and a location is inferred through a nearest-neighbour comparison. This experiment would work under the assumption that neighbors in the matrix are separated by a linear distance. In this thesis, however, it is assumed that distances in signal space do not adhere to a linear relationship. By assuming that the fingerprints constitute the coordinates in signal space and that the distances between them are non-linear, a manifold learning algorithm can be applied to them. This is done in the hope that the algorithm can "flatten" the radio map, making the distances truly Euclidean after which the above supervised learning can be performed in earnest.

As has been detailed earlier, the manifold learning algorithm used was *Isomap.* Since Isomap works on the MDS principle, it needed the data in the form of a distance matrix. This distance matrix was be made up of the Euclidean distance in signal space between the fingerprints gathered earlier. In essence, each fingerprint was compared to each other as if they were Cartesian coordinates in Euclidean n-space. Thus, $d(\vec{s}_i, \vec{s}_k)$, where $\vec{s}_k$ is another fingerprint, in distance matrix X is defined as

$$d(\vec{s}_i, \vec{s}_k) = \sqrt{\sum_{l=1}^{n} (s_{il} - s_{kl})^2}$$

X will thus take the form of an $[n \times n]$ matrix:

$$X = \begin{pmatrix} 0 & d(\vec{s}_1, \vec{s}_2) & d(\vec{s}_1, \vec{s}_3) & ... & d(\vec{s}_1, \vec{s}_n) \\ d(\vec{s}_2, \vec{s}_1) & 0 & d(\vec{s}_2, \vec{s}_3) & ... & ... \\ d(\vec{s}_3, \vec{s}_1) & d(\vec{s}_3, \vec{s}_2) & 0 & ... & ... \\ ... & ... & ... & ... & ... \\ d(\vec{s}_n, \vec{s}_1) & d(\vec{s}_n, \vec{s}_2) & d(\vec{s}_n, \vec{s}_3) & ... & 0 \end{pmatrix},$$

where the diagonal is represented by a zero vector since a point's distance from itself is 0 and $d(i, j) = d(j, i)$ for all i, j.

This final form of the measurements is something that Isomap is equipped to handle. In addition to the distance matrix above, Tenenbaum's own MATLAB implementation requires that the neighborhood size be chosen. The size of this neighborhood is an important parameter to choose well, and it should be tuned to the size of the testing environment. This was done through model selection, by way of running Isomap several times with different values for the neighborhood size so as to achieve maximum accuracy.

The MATLAB version of Isomap outputs three matrices: Y, R and E. Matrix Y contains the coordinates for the d-dimensional embedding in the form of an array of size $[D \times m]$, where D is the dimension the data has been reduced to, and m is the number of points. In our testing, m is defined by the number of training points recorded. In addition to this, Y contains the indices for the points in the embedding, which can be used to find the original data points in the new embedding.

Vector R contains the residual variances for the embeddings in Y. Though these values were initially thought to be indicative of mapping accuracy, using them as a metric was later abandoned due to poor performance. Namely, it was discovered that a low residual variance in the final embedding was not indicative of a small positioning error. Finally, E is the edge matrix for the neighborhood graph and serves no further use in this testing.

Once this two-dimensional representation of the signal space is obtained, it can be matched to the floor plan of the environment in which the tests are running. Due to Isomap relying on classical MDS, however, this mapping is not a 1:1 match. The embedding need thus be the subject of *rotation*, *scaling* and *mirroring* before it can be used for positioning [CoC01]. This process is handled in a calibration phase, which entails adding labeled key points to the set of training points. Since the real-world locations for these key points are known, they can be used to adjust and place the mapping in a representative coordinate space.

The neighborhood size K of the mapping is chosen to be the one that leads to the smallest error between the embedded key points and their actual real-world coordinates. In other words, the parameter that minimizes the average

$$\frac{\sum_{i=1}^{k} d(y_i, \hat{y}_i)}{N},$$

where y_i are the coordinates for the key points in the mapping, $\hat{y}_i$ are the real-world coordinates, d is the Euclidean distance between them and N is the number of key points in the calibration set. The specifics of the calibration phase are discussed in more detail in section 5.

As has been the case with most other fingerprinting studies, training will need recalibration if the trained environment changes radically. As long as enough "familiar" APs remain in the environment old training data should be usable, though, by restricting the fingerprints to using these APs alone.

4.2 Testing procedure

The above training procedure has at this point resulted in a two-dimensional representation of the signal space, where non-linear relations between measuring points have been made linear. After calibration this representation should match the testing environment as well as possible. We can thus now record new fingerprints and place them in the mapping, to gauge the accuracy of positioning based on it.

Testing involves picking random locations in the testing environment and reading the RSSI into a vector, like in the training phase. This testing vector is then compared to the data collected in training, specifically matrix M. Using a *k-nearest neighbors* search the k closest points in the space are gathered. This choice of k is another parameter used for accuracy maximization. Because Isomap outputs not only the embedding coordinates, but also the indices for them, the k neighbors can be found in the two-dimensional mapping. The localized position could then be defined as the average over the neighbors, or the point that is the closest to the average. Choosing the average point results in a position that has never actually been recorded, which could put the localized position inside a wall or another impossible location. Equating the position with the closest neighbor ensures that the position exists, but sacrifices some accuracy. In Figure 13 this choice is visualized. In *a* the finalized location of y_i has been chosen as the average between points A,B and C. In *b*, y_i is placed on point C since it is closest to the averaged location. In our testing we have chosen to rely on the average alone. This was decided based on the fact that we are specifically interested in the accuracy of the method, and not placing the localized point in a sensible location. In the future avoiding illegal positions could be achieved through map-matching or other post-estimate constraints. These efforts are beyond the scope of this thesis, however.

To ensure that the training and testing procedures (such as the choice of K) are not over-fitted to the initial testing environment, they are also run in a completely

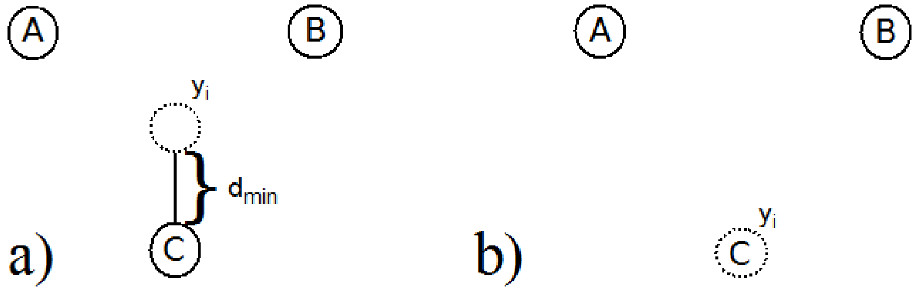

Figure 13: Choosing the final location as a)the average of neighbors b)the point closest to the average

separate environment. This environment has different set (and number) of APs as well as having a different layout of walls and obstacles.

4.3 Testing environments

The testing environments used in the final deployment of the system consisted of the third floor *A-wing* of the Exactum building in Kumpula, Helsinki, as well as the *Science Library* on the campus. Specifically, the area of the A-wing chosen for measurements covered part of the hallway and an adjoining lounge/meeting room, A317, see Figure 14. In total, this environment spanned an area of about 24x7 m. The room was selected as part of the testing locations due to its relatively static configuration and infrequent use. It serves mainly as a meeting room for project gatherings and is thus mostly empty otherwise. The room is dotted with furniture, however, which makes impromptu calibration challenging. The furniture might also be moved around as the situation calls for it. During calibration and testing the furniture remained stationary, and no piece of furniture is taller than chest height. We could thus assume that the configuration of the room itself had minimal effect on the final mapping success.

The main area of interest for testing is the hallway that adjoins this meeting room. It is an interesting measurement space for several reasons. First, because paths cross and have a distinct shape, a proper mapping should begin to display a distinct shape right after calibration. Second, the hallway is devoid of furniture and other objects, meaning the space could be mapped more or less thoroughly. Third, the nearby kitchenette with its electronic appliances could be seen to have a distorting effect. Finally, researchers and students using the nearby facilities, classrooms and halls introduced noise, meaning we were not restricting ourselves to a best-case environment.

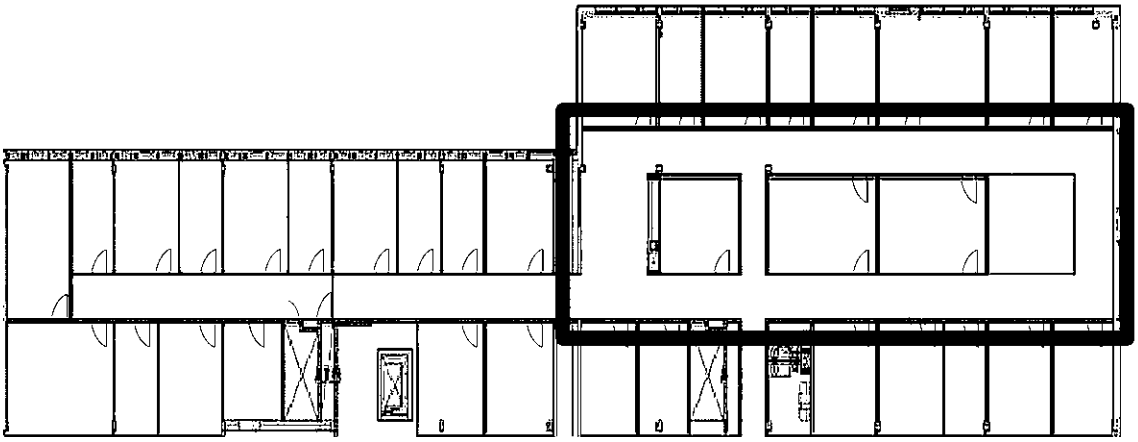

Figure 14: Floorplan of A-wing. Measured area highlighted

To make sure we sampled completely different sets of APs for testing, the second testing environment was the Kumpula Science Library (Figure 15). The size of the section selected for testing was about 24x14 m. Though the library building is physically joined to the Exactum building, the amount of walls and distance between the testing environments ensured we didn't use the same APs for measurements. Before testing proceeded, the environment was considered somewhat challenging due to its configuration. In addition to being somewhat busy throughout operational hours, the placement of APs is less than optimal. Because APs are installed on the ceiling, the signal is allowed to spread unhindered and forms many areas where signal strength is more or less constant. Due to the issue of aliasing discussed in section 3.1.2, this might cause the gathered fingerprints to look alike and not provide good inter-point distance measurements. APs in this environment also change channels dynamically, but because this is thought to happen relatively infrequently (weekly basis at the most) it was not considered a major issue. At most, this might necessitate regular remapping. With a sample rate of 1 Hz, the testing environment (a section of the upper floor) was trained in a matter of hours.

5 Implementation

In the following, we will introduce the intuition and specifics behind the technical implementation of the theoretical framework described above. Following a pro-

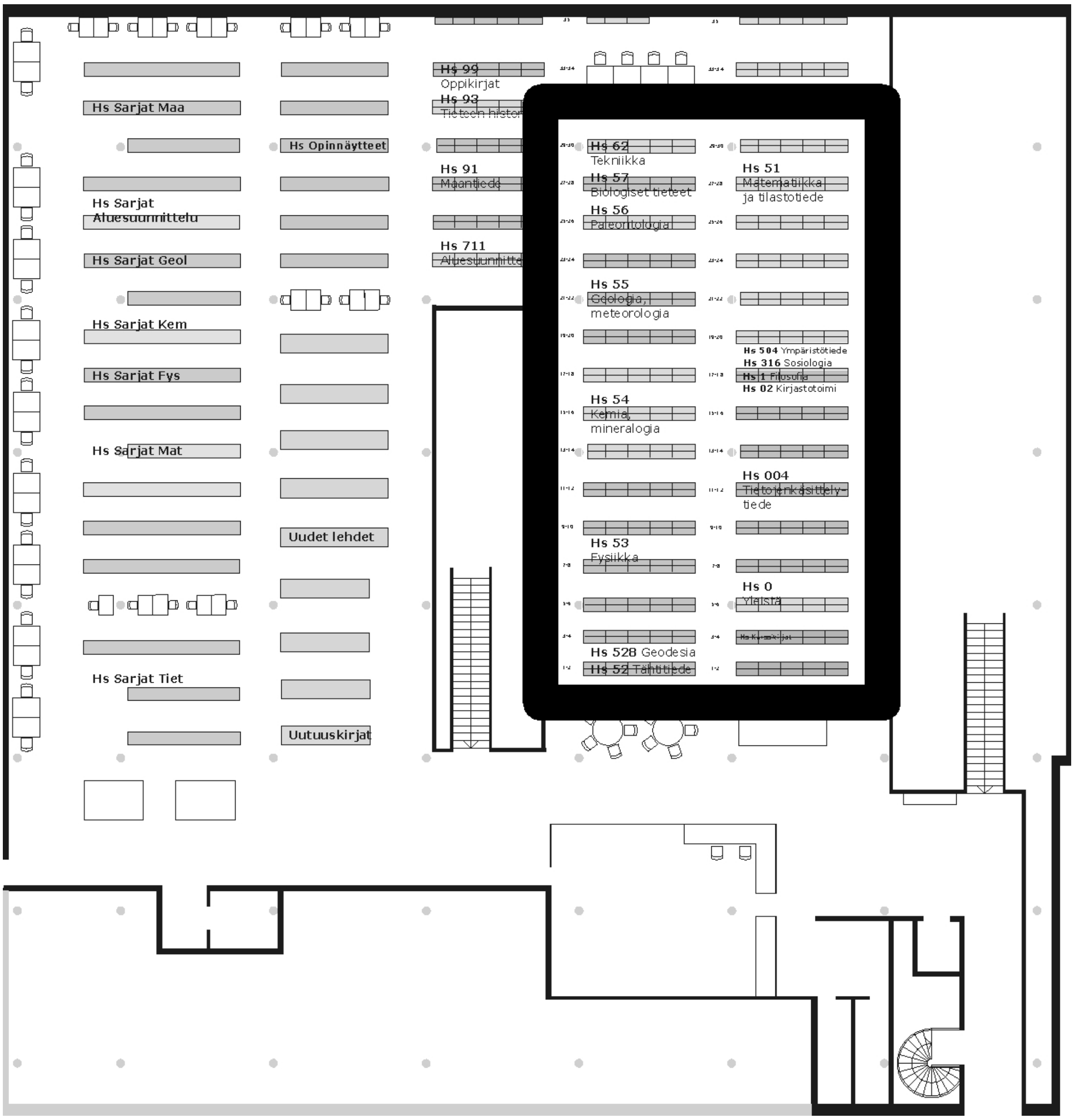

Figure 15: Floorplan of Library. Measured area highlighted

cess displayed in Figure 16, training and testing was implemented separately. The training phase includes the initial survey up until the final embedding Isomap has produced. From here, calibration takes over and key point fingerprints (with known positions in the environment) are used to adjust the mapping. Finally, by sampling test points from the environment and performing a k-NN search, we can judge how well WLAN positioning using the created map works.

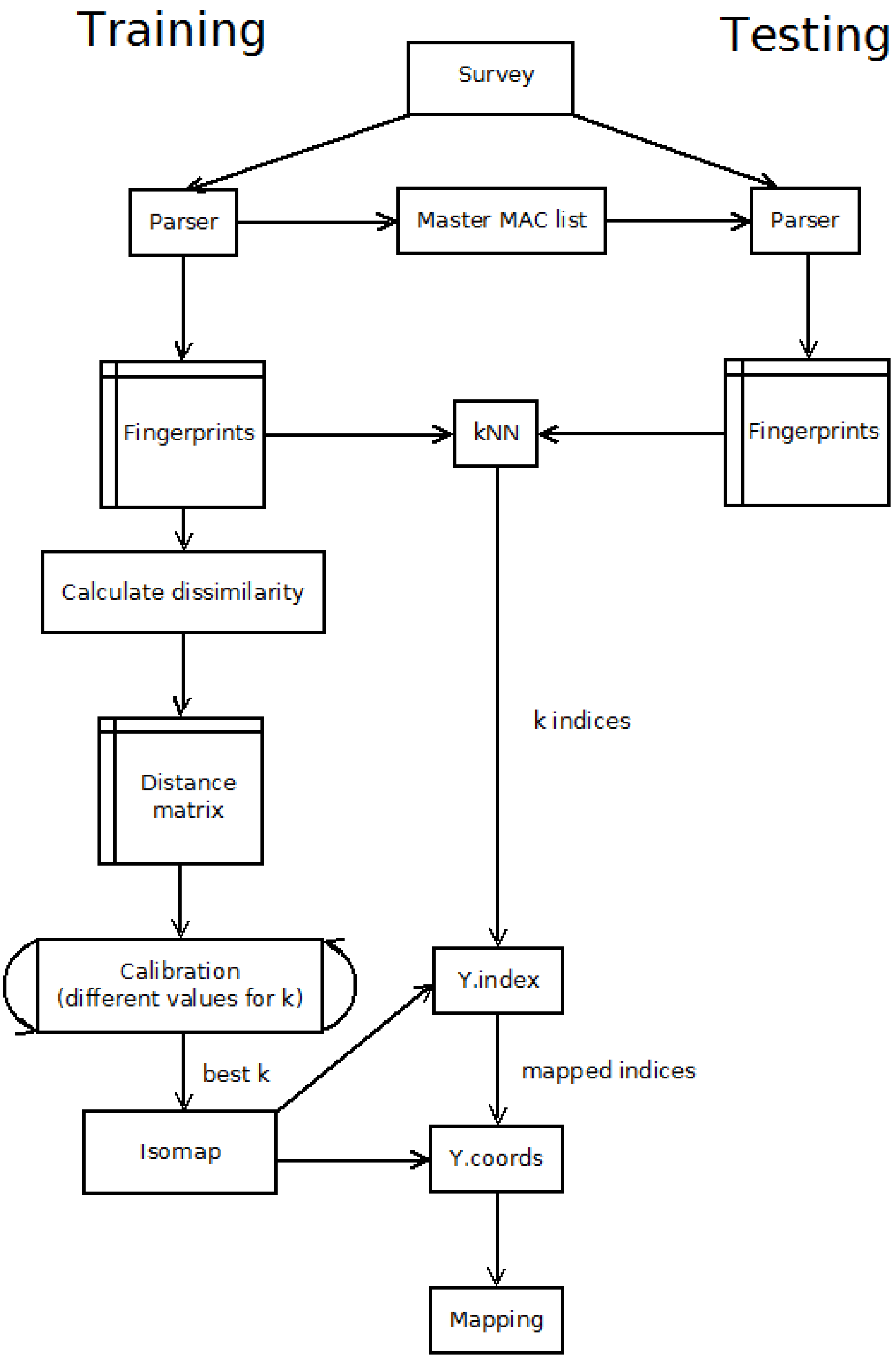

Figure 16: A rough depiction of the implementation

5.1 Training

For the purposes of RSSI recording a Python implementation was used. This implementation used the *pyiw* library to survey the current WLAN signal environment.

Due to either a hardware or software limitation, the source of which were not able to find during development, the wireless library and interface was limited to a scanning speed of 1 Hz. This limitation proved something of a blessing in disguise, however. Efforts to increase the measurement frequency on a HTC Hero smartphone running Android proved fruitless. Though the environments were measured considerably faster, the mapping performance was decidedly worse. Scanning speeds of [0.05,0.1,0.2,0.5] Hz gave improved mapping accuracy as the frequency was decreased. Since the wireless interface and hardware intricacies were out of reach, we were left to draw our own conclusions. A faster scanning speed might simply be an illusion, with the reported signal strength simply repeated for many of the recordings. Since studies have shown that signal strength measurements benefit from recordings over a longer period of time, it is also possible that the signal is not allowed to "settle" with a faster scanning speed. The measurement would thus fluctuate wildly during the recording phase and would not be indicative of the true signal space in the measured location. In the end it was decided that the time spent recording fingerprints would serve the final positioning accuracy well.

Since the performance of APs varies greatly in that those that have a stronger signal tend to output their signals faster, a limit for the collection of fingerprints was imposed. RSSI were collected until either five signals had been collected from the weakest AP, or ten from the strongest one. This ensured the algorithm was not left waiting indefinitely for an AP that might never reappear, but also did not let the strongest signal fingerprint grow too severely in proportion to the weakest one(s). The restriction also contributed to make mapping of the environments possible within a sensible time frame (limited by the charge of the laptop battery, for instance).

As it stands, values in the fingerprint can vary widely due to interference. This means the same AP can be read as outputting several different values, on the order of 20 dBm between the most extreme ones. In an algorithm that is based on the differences between recorded fingerprints this can cause severe distortion in the mapping. It was thus decided that fingerprints should be toned down to minimize the maximum standard deviance. In essence, this meant removing the value which deviated most from the mean until the difference between the maximum and mean value reached a predefined interval (in testing: 5 dBm). Since signal strength values tend to vary the most near the maximum possible value [KaK04], only the weakest values were pruned.

Once the fingerprints had been recorded and pruned, each fingerprint was averaged. Next, a parsing algorithm read the collection of fingerprints in order to organize them in a way that is easier for subsequent MATLAB algorithms to handle. Signal strengths for a particular AP at a specific measuring point might be hidden due to attenuation or environment topography. The collected fingerprints thus needed to be "filled" out to conform to a unified standard. In addition, they needed to be ordered the same way for the fingerprints to be comparable as there is no guarantee the surveying algorithm outputs the APs in any specific order. For this purpose a "master MAC list" was created that contained an instance of each MAC seen in the entire collection of fingerprints. This MAC list was at this point also exported into a file, for later use in test point parsing. Based on the length of this MAC list, a dummy fingerprint list was created with low (-110 dBm) values for each RSSI. Next, the original fingerprints were placed into this dummy list, replacing the dummy values with their own in the order dictated by the "master list". Values not replaced were left in the list, representing a very weak measurement. Before being turned into a compatible distance matrix, the average signal strength for each AP (that is, averaging over all collected fingerprints) was calculated. As a part of the preprocessing of the data APs that consistently supply weak data were removed. Weak data here meant the average for the pruned AP was lower than -100 dBm. This included APs that were only seen in a fraction of measurement points, meaning the majority of their signal strengths were made up of the "dummy" prints created during the parsing phase. Whatever information they potentially supplied to the fingerprints where they were measured was heavily offset by the substitute values.

Finally, using a fully vectorized solution (supplied with Tenenbaum's MATLAB package), we were quickly able to create the needed distance matrix based on the list of fingerprints collected.

5.2 Calibration and testing

Before the final Isomap embedding was finished, a model selection was performed. Because the mapping that was initially created had no guaranteed scale or rotation, we needed to find a set of parameters that minimized the difference between the embedding and the surveyed environment. By adding labeled key points to the training data and embedding them among the rest of the measurement points, we could make adjustments based on the embedded key point coordinates and their known real-world counterparts. The key points were recorded in the same way as the testing points, meaning the fingerprints were ordered to match the MAC list created during training. For small K-values (usually less than 3), Isomap was not able to embed all points in a mapping. We thus restricted ourselves to using mappings where all training points were included. For these points, a transformation was performed based on the embedded key points.

Since we only needed to subject them to a degree of rotation, scaling and mirroring, the relationship between the coordinates of embedded key points and their real-world counterparts could be approximated through a least-squares regression to tune the embedded coordinates to match the measured environment. A strictly linear implementation provided promising results, but due to the assumed pairwise non-linearity of the coordinates we extended the model to contain the squares of the mapped points. In this regression model, we can depict the relationship as

$$X = b\tilde{M} + \epsilon,$$

where X are the real-world coordinates, $\tilde{M}$ are the embedded coordinates extended with their squares, ϵ is an error vector and b is a parameter vector that can be found through the usual least-squares formula

$$\begin{aligned} C_x &= b_x\tilde{M} + \epsilon_x \\ C_y &= b_y\tilde{M} + \epsilon_y, \end{aligned}$$

for the respective embedded coordinates, where b can be solved through

$$\hat{b}_x = (\tilde{M}^T\tilde{M})^{-1}\tilde{M}C_x$$

for the x-coordinates and similarily for the y-coordinates. The final regressed map could then be found through a simple vector multiplication of $\hat{b}$ and the measured Isomap coordinates. This approximation was run for all possible Isomap neighborhood sizes in order to find a K that best matched the measured environment. The average residual distance between the adjusted key points and their real-world coordinates were stored as an error metric. We could then choose the b and K that produced the smallest error and use them to embed and transfer the rest of the manifold to the same coordinate system. This approach assumed the manifold's sparsity matched the measurement point distribution in the surveyed environment and that the distance relations between points were similar. Testing results supported the notion that this was a valid way of transferring the manifold to a coordinate system matching that of the floor plan.

Figure 17 illustrates the calibration process. The map Isomap outputs is centered around the origin, with points in the embedding spread out accordingly. By tuning the entire map based on where the key points have been embedded, we could quickly and accurately transform the entire map to the new coordinate system. We have included a bounding rectangle in the plots to illustrate the transformation the mapped points are subjected to during the calibration process. The non-linearity of the fit is clearly visible in the distortion of the rectangle.

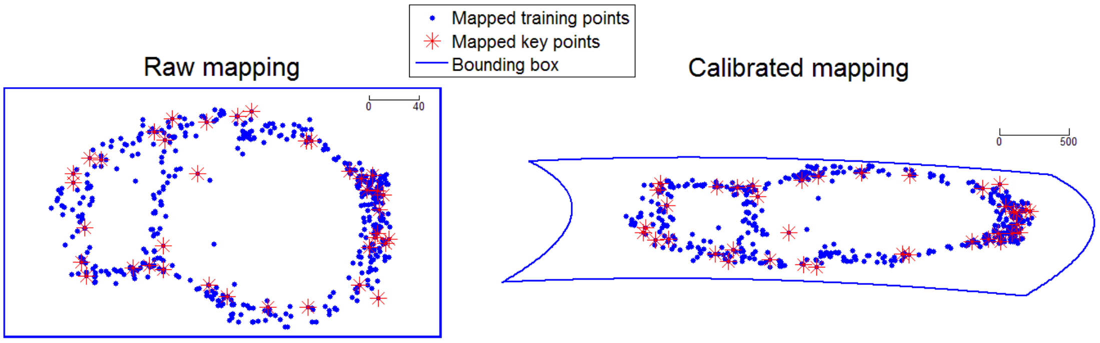

Figure 17: The map as output by Isomap and after calibration, respectively.

The testing procedure could then simply compare new fingerprints recorded to the training set through a *k nearest neighbors* search. Since Isomap by default outputs the indices of the embedded coordinates, we could find the neighbors of the fingerprint in the embedding and give the test point an average of these neighbors' coordinates.

6 Deployment and results

The environment survey, preprocessing and most positioning tests were performed with a Samsung NC10 Netbook, running Ubuntu Linux 9.10. This netbook was equipped with an Atheros AR5007EG Wireless network adapter, complying to the 802.11b/g standard.

The testing was performed in two separate locations with separate sets and configurations of APs in Kumpula, Helsinki. Initial and configuration tests were done in the third floor of the Computer Science department, in a section of the A-wing.

Later, a standard test was run in the Science Library, thought to be a challenging location due to varying architecture and AP placement.

The tests were for the most part unidirectional, meaning training and testing had to be performed facing the same way. The initial intuition might suggest measuring fingerprints under different orientations might prove detrimental to an approach that depends on the dissimilarity of the fingerprints. Some orientation tests were later performed, but they proved mostly inconclusive. As long as the training conditions matched the testing ones, i.e. testing measurements were done in an orientation measured during training, the mapping would succeed. Orientation testing actually showed improvement over strictly unidirectional testing. We attributed this to the fact that orientational measurements simply contained four times the fingerprints compared to measurements done in only one direction. It quickly became apparent throughout testing that this was key to the success of the learning process. Since both fingerprinting approaches in general and the Isomap algorithm specifically benefit from dense measurements, it follows that a positioning approach based on both would do likewise.

The following sections are organized based on the aspects we set the tests up to measure. The tests were run in different environments, during different times as well as under different sparsity conditions. The sparsity testing was performed entirely offline, entailing the simulation of fingerprint as well as measurement sparsity by increasingly truncating already measured datasets.

6.1 Standard tests

The initial and baseline tests included measuring fingerprints in the mentioned environments with no additional constraints or parameters. These tests were set up to give an indication of how different infrastructure and access point placement affected mapping results. The focus of the baseline tests was to gauge the system under what could be assumed to be "normal" conditions. Training and calibration were performed under light crowding conditions and the testing dataset was collected during normal working hours. In addition, the testing sets were collected a few days apart from the calibration sets. Doing this ensured we were not simply over-fitting the data to the prevailing conditions, and made sure they represented the setup the system would actually be deployed in.

Calibration of the A-wing entailed collecting fingerprints after-hours, when the department was otherwise closed and the building was devoid of researchers and students. The testing set was collected during the morning and afternoon, when the hallway and adjoining room was visited by numerous people during the measurements. A completely separate set of data for the A-wing was collected as well, to gauge temporal and crowding effects on fingerprinting. This study is detailed in the next section.

The library was calibrated during the morning, when it was thought most students were attending lessons or otherwise occupied. A completely noiseless calibration was not achieved, however, since the duration of the fingerprint recording meant it was finished in the afternoon. The section of the library measured was judged quite calm throughout in any case. The testing set was recorded a few days later during the afternoon, when more students were around.

During calibration of the A-wing, around 500 fingerprints were collected in total (including key points). Of this amount 38 were key points. Test points were gathered in 66 separate locations in the environment, illustrated in Figure 18. The evening calibration was left with a residual error of about 1.9 m after embedding. The residual error is defined as the average distance that remains between the embedded key points and their predefined real-world location. This set something of a lower bound for the positioning accuracy we could hope to achieve using this mapping in positioning. Though test point accuracy sometimes actually superseded this, this was considered mostly incidental. The residual error of the calibration effort mainly depicts our confidence in the mapping; errors larger than this among test points are probably caused by interference during test measurements.

To achieve this mapping Isomap only needed 15 neighbors out of a total of about 500 possible points. This lends credence to the notion that the relations between points in signal space are non-linear. A larger neighborhood value would suggest at least parts of the space was linear and could be solved with MDS alone. In the extreme case, Isomap would use all neighbors available (all measured points). This would, in essence, amount to using the linear MDS from the start.

Plotting the calibrated points onto the floorplan of the A-wing, we can clearly see the shape of the hallway in the mass of points, Figure 19. A majority of the points mapped to the hallway respect the infrastructure; relatively few points are mapped into the rooms in the center. Considering the embedding was not constrained in any way, this was an interesting result. It is clear that the hallways insulate the WLAN signal and create unique signatures, which Isomap was correctly able to interpret as

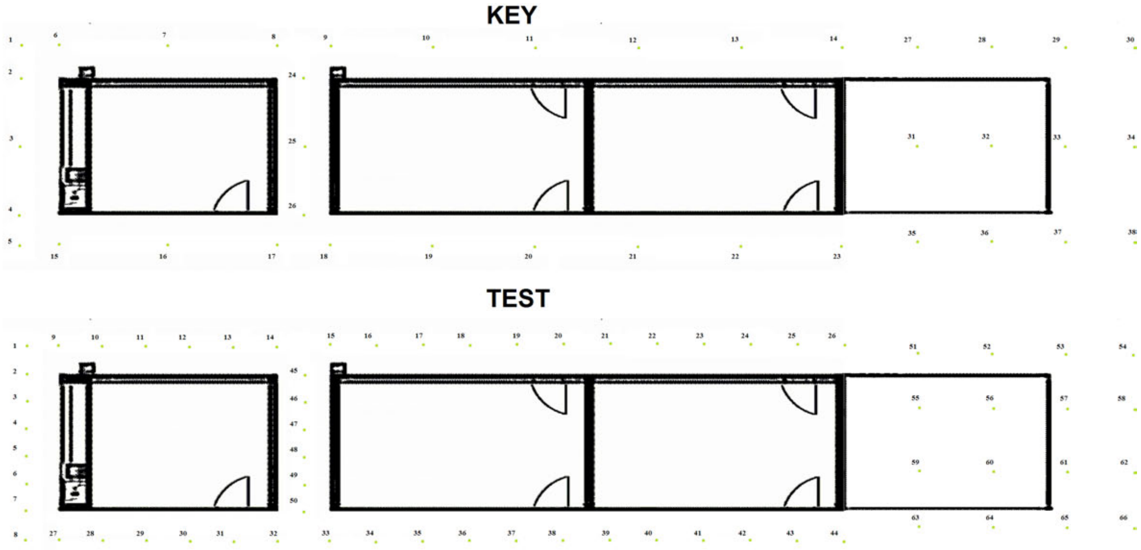

Figure 18: Layout of key and test points in the A-wing.

proximity in the real world. The mapping of fingerprints in A317 was not as distinct, however. Points seemed to congregate around the center and the room was not as distinctly represented as the hallway. This was most likely because, as mentioned in the environment presentation, it was void of attenuating infrastructure. Signals travelling in the room were allowed to spread freely and thus did not contribute to a distinct enough signature. Fingerprints collected from opposite sides of the room were much more alike than the ones collected from different arms of the hallway. The extreme similarity translated to proximity through Isomap. This factor was no doubt also compounded by the hallway fingerprints being embedded in the same space. The hallway fingerprints were allowed to dictate the dissimilarity-to-distance ratio, and the relative similarity of A317 fingerprints suffered from it.

Placing the collected test points in the mapping through a nearest neighbor matching (with 1-10 neighbors), we achieved an average accuracy of about two meters; the smallest average accuracy achieved when the test print was defined as the average coordinates of eight embedded locations. Varying the neighborhood size in the k-NN approach seemed to have meager effect; the average accuracy when using 1-10 neighbors for comparison was 2.02 m, with a standard deviation of about four cm. It is worth noting, however, that though the average accuracy of the test points was about two meters, the individual values varied greatly, see Figure 20. The largest error was considerable, at about 11 m. On the other extreme, the smallest error was only about three cm. All told, the position accuracy had a standard deviation

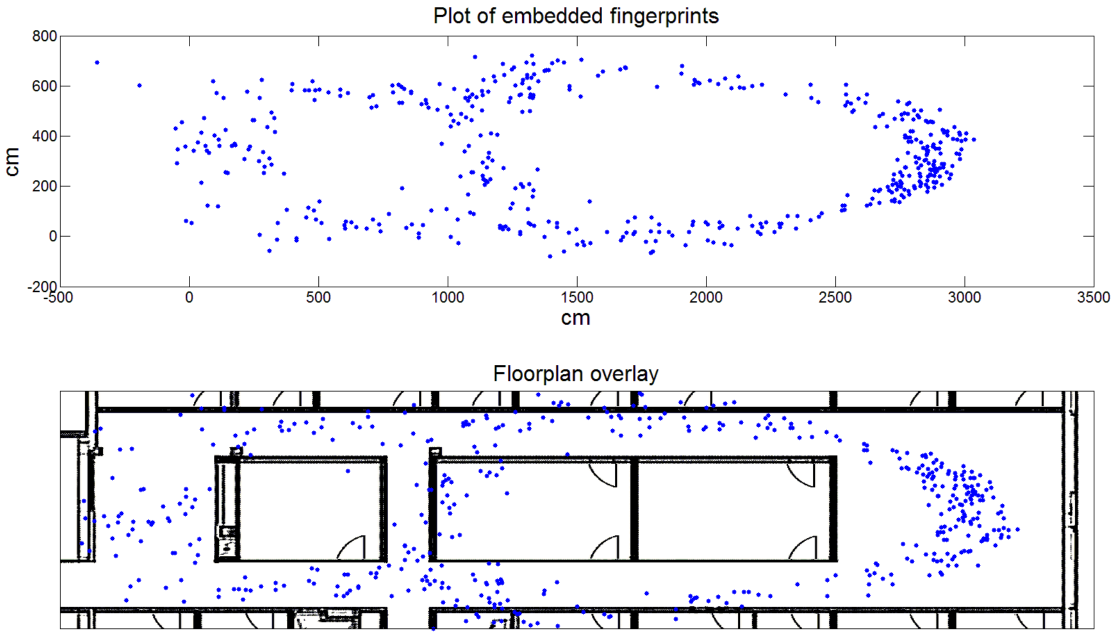

Figure 19: Plot of embedded fingerprints, A-wing

of 1.6 m which is indicative of the fact that the results are highly influenced by outliers. The median accuracy of this run was 1.5 m. Removing the main outlier decreased the average accuracy to 1.83 m and the standard deviation to 1.2 m. The full statistics are shown in the table at the end of this section.

The calibration of the library left us with a residual error of 2.24 m. To achieve this, Isomap needed 92 neighbors (out of a total 533+24 fingerprints collected, see Figure 21). This was a significant increase over the 15 needed for the A-wing, even though the amount of measured points was about the same. The lower sparsity (fingerprint to area ratio) in this environment could be explained by the infrastructure. Though the space in itself is twice as wide, the bookshelves constrain the measurable area to single paths between them. A visual inspection of the mapping, see Figure 22a, made it clear the area hadn't been captured as distinctly as the A-wing. Though obvious borders in the points can be detected, there is clear distortion in the mapping. The bounding box in particular displays the distortion caused by the squared fit. To combat this the calibration was rerun without including the squares of the coordinates detailed in the calibration section, Figure 22b. In other words, in this mapping the points were transformed strictly linearly. This new embedding had a worse residual mapping error at 3.4 m, but much improved test point accuracy. It

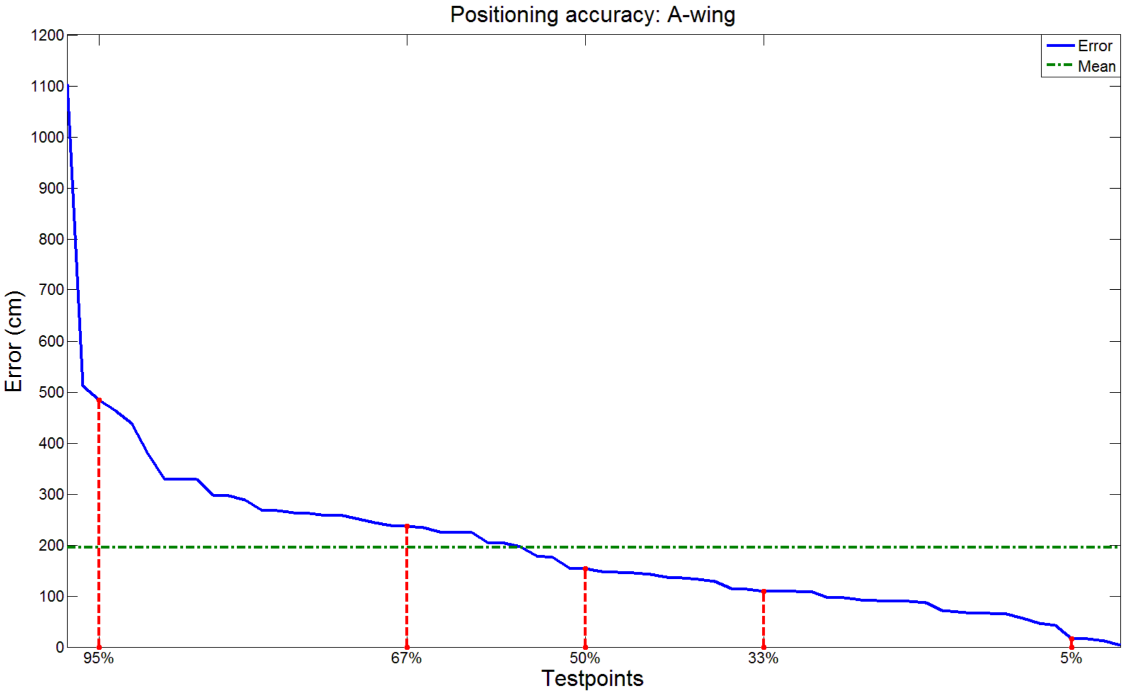

Figure 20: Distance errors for test points in the A-wing

also found the best mapping using only three neighbors, which is usually a good indicator of a proper fit. It seems, for this specific environment at least, adding the squared coordinates to the model lead to over-fitting. The improved mapping also suffers from drift, but otherwise maintains the rectangular shape of the library a lot better.

Using ten neighbors in the nearest neighbor matching for the linear mapping of the library, we arrived at an average error of about four meters for the test points. These results also had a high variance, with a standard deviation of about 1.6 m. The worst error was about ten meters while the smallest error was only 60 cm. In other words, the mean error was very much defined by its extremes, see Figure 23.

In conclusion, the library was a problematic environment partly because of the placement of access points, which in an effort to cover as large an area as possible to optimize efficiency are placed on the ceiling and thus spread signals uniformly. The shelves themselves are not particularly good at blocking signals, either. It was thought before experiments started that even though the AP placement was poor for positioning use, the aisles created by shelves would have unique fingerprint signatures, since the signals from outside would be blocked. The sparse grouping and diverse height of the books, however, allowed for significant gaps in the shelves.

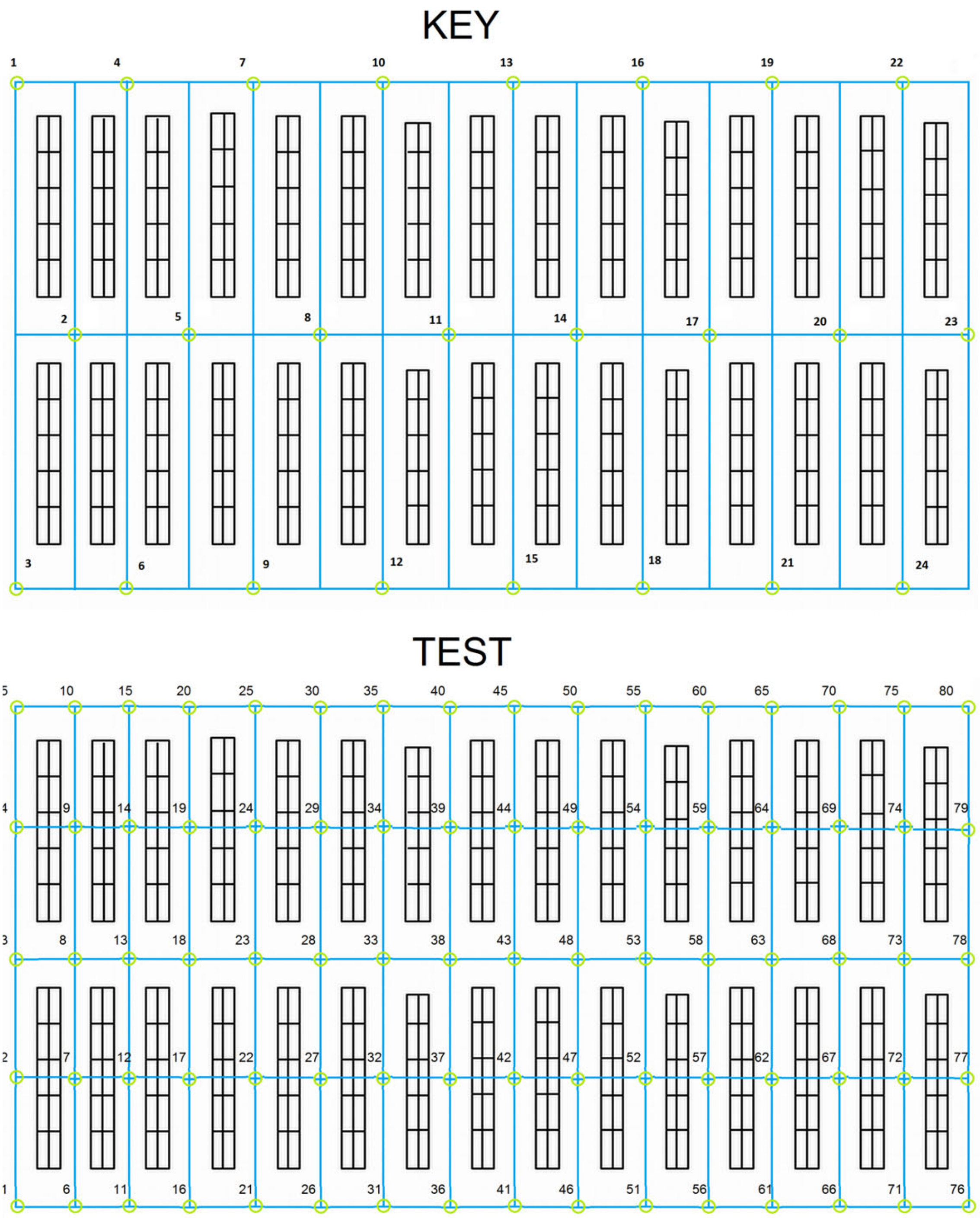

Figure 21: Layout of key and test points in the library. The solid blue line depicts the paths measured for training. The green circles indicate the locations where key- and test points were measured.

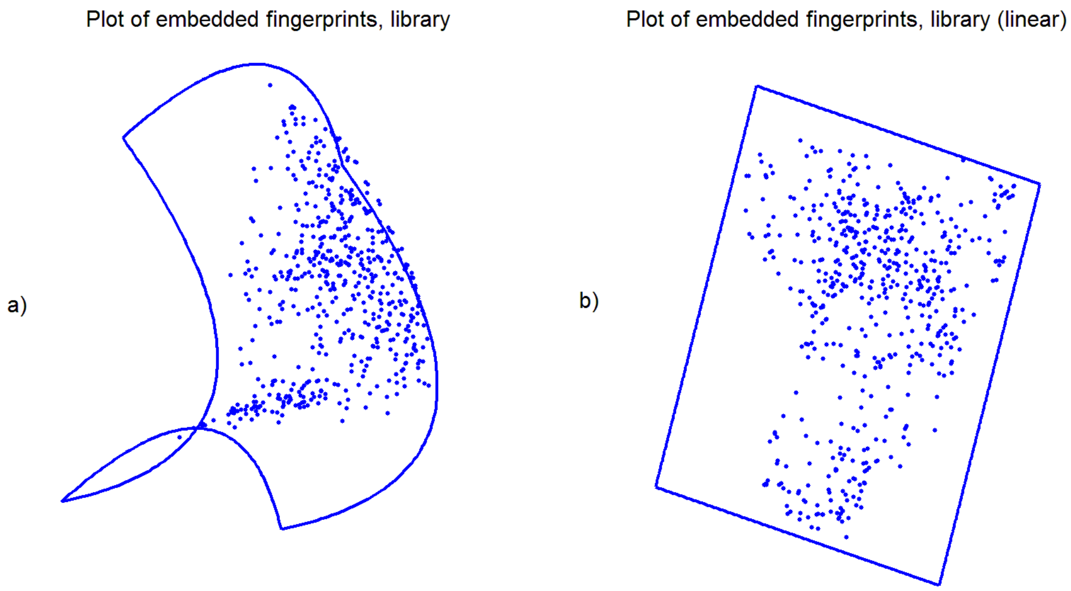

Figure 22: Plot of embedded fingerprints, library. a) Normal b) Linear

This undermined whatever shielding effect they were thought to have, equating the majority of the library area to one enormous open space. This proved severely detrimental to an approach that relied on fingerprints being uniquely distinguishable from each other, especially when the defining feature of the main algorithm was based on the difference between them.

The A-wing proved to be the best environment for mapping under normal conditions. Specifically, the hallway could distinctly be recognised even from the calibrated data itself. The most problematic section of this area was the open lounge at the end of the hallway. It thus seems that our approach like most other WLAN-based methods benefits from an obstructed environment. This open space problem is especially aggravated by the policy dictating AP placement. For the sake of cost-efficiency, APs are placed in a way where they have the best coverage. An approach relying on the ubiquitous nature of WLAN architecture is thus something of a double-edged sword. Not relying on separate infrastructure contributes to fast and wide implementation, but cannot handle open spaces as well as a designed positioning environment would.

The table below lists the final results for the baseline testing.

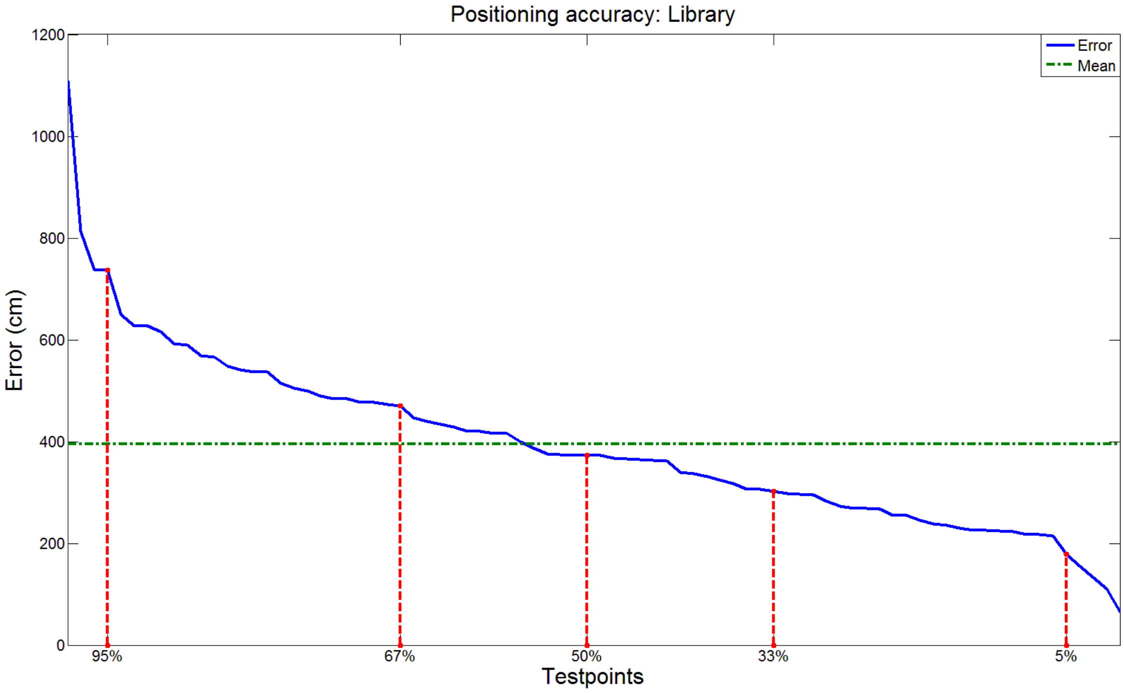

Figure 23: Distance errors for test points in the library

Env.	n_{tot}	n_{key}	n_{test}	K	max	min	mean	σ
A-Wing	437	38	66	15	11 m	0.03 m	2.0 m	1.6 m
Library	533	24	80	92	14.8 m	0.07 m	5.2 m	3.3 m
Library,linear	533	24	80	3	10.4 m	0.60 m	3.9 m	1.7 m

Env. The environment measured

n_{tot} Number of fingerprints collected

n_{key} Key points defined

n_{test} Test points collected

K Isomap neighborhood size

max Maximum error among test points

min Minimum error among test points

mean Average error of test points

σ Standard deviation of error

6.2 Special circumstance testing

In the following sections we present results from testing that deviates from the standard setup detailed previously.

By measuring two distinct datasets for the A-wing environment on two separate days and times of the day, we endeavoured to find the effect of time as a parameter. The environments change not only in terms of furniture placement, but also in pedestrian traffic levels as well as access point configuration. Finally, using data already collected we re-ran old tests with decimated fingerprints and measurement locations. With this we hoped to be able to define a minimum standard for successful positioning using the proposed method.

The A-wing datasets were chosen as the testing environment for these experiments as they had given the best results. It was thus easier to gauge the effect of variables in this dataset.

6.2.1 Time and crowding

A major factor in positioning based on WLAN RSSI is the effect of calibration deterioration. Since location determination relies exclusively on fingerprints gathered previously, any difference between the infrastructure and access point configuration between these two events has an effect on the positioning quality. The introduction of obstacles can block signals from access points that previously provided a significant signature to the measured environment. The adjustment of access point antennas, either deliberately or accidentally, can have a detrimental effect on fingerprint comparison.

In addition to permanent changes to the measured environment, the effect of crowding on fingerprint quality is undeniable. As detailed earlier, the human body is made up of up to 70% water. A busy environment will thus cause a source of noise when access points are obscured from one fingerprint to another. Since our approach currently considers missing access point RSSI as simply extremely weak it is extremely susceptible to this kind of variation. Especially since the mapping considers pairwise differences between fingerprints; a temporarily blocked access point causes a direct shift between measured points.

In light of these issues it was thus in our interest to gauge the short-time effects of time on our positioning quality. In essence, we wanted to know if and how collecting test prints separately from calibration data affects the positioning quality. To facilitate this two complete datasets were collected in the A-wing. One set of calibration and test data collected during the morning (from here on referred to as the *busy* dataset) and one set collected after-hours (*quiet*). By cross-examining these datasets we hoped to gauge the sensitivity of the approach to temporal changes.

The baseline tests already contained a temporal element in that testing sets were measured on a different day than when calibration took place. In the following we present results from testing the *busy case* and *quiet case* scenarios, as well a combination of them. The *quiet case* scenario here means when the calibration and testing sets are not only measured during the same calibration session, but also after-hours. This way the environment doesn't have time to change between measuring sessions and is free from the noise caused by human interference. Though this scenario could be considered optimal in terms of positioning accuracy, it is not very realistic. Calibration is assumed to be performed relatively seldom and can include special measures to ensure a noiseless environment. Testing, however, is supposed to represent a real use case. Measuring test points during quiet hours does not give a realistic view of the robustness of the positioning system. In our case this setup will mainly be compared to the baseline case, thought to depict a realistic scenario.

The *busy case* scenario employs the opposite circumstances. Calibration performed under crowding introduces a lot of noise. Most importantly, recording fingerprints for key points under heavy interference will translate to errors in the final mapping. Measuring test points under these same circumstance only acts to compound on these negative effects.

Using the quiet case scenario we achieved an average accuracy of about 1.8m for test points. This is a definite improvement over the baseline test, though not a significant one. The main difference can be seen in the standard deviation, which shrunk to 1.1m compared to 1.6m of the baseline test. The maximum error was a relatively lower 4.8m. Interestingly, the maximum error occurred in the same test location as in the baseline. This suggests the calibration is not optimal in this location, though the testset used in the baseline tests seemed to compound this error further.

To some surprise, the busy case scenario actually scored an average accuracy of about 1.55m. This is further indication that the quiet case calibration suffered from interference, which translated to all results related to it. This result nevertheless supports the notion that measuring calibration and testing data on separate days decreases the accuracy compared to collecting all data at the same time. Using the busy calibration set with the quiet testing set gives the next-best results at about 1.6m.

In conclusion, accuracy seems to be negatively affected by separating calibration and testing measures, but these effects are marginalized by the success of the calibration. The busy case scenario, which could intuitively be defined as the worst possible condition, scored the highest accuracy even compared to when the data was collected exclusively during quiet hours. This suggest that at moderately short intervals, such as the week of separation of dataset recording reported here, the temporal difference between fingerprints is negligible. The main focus using the presented approach should instead be to ensure the calibration is performed carefully. Whatever effects crowding may have on positioning accuracy, the calibration setup is more important.

Below is listed the final cross-comparisons of average error between datasets.

Calibration/Testing	Quiet	Busy
Quiet	1.8 m	1.95 m
Busy	1.6 m	1.55 m

6.2.2 Sparsity

Since all tests so far have been performed in academic circumstances, and specifically computer science oriented ones, we have been blessed with an abundance of WLAN APs in the environment. Even in the worst cases, we have been able to record fingerprints of length 30 or more. This means RSSI have been subject to pruning if the AP that supplied them has consistently supplied weak data. To ensure our approach isn't limited to areas with dense wireless coverage, we set out to test how accuracy is affected by fingerprint length. This was done by simply increasingly decimating the fingerprints we have already collected.

In addition we endeavoured to measure how measurement point density affects accuracy. Since Isomap is highly volatile when it comes to manifold sparsity, we thought a small study using previously gathered data would be of interest. Like in the fingerprint length study, we simply performed embeddings for different configurations of measurement points. To ensure we were not simply shrinking the manifold, the measurement points were decimated at random, with the intent of creating an area of equal size but increased sparsity.

Since the most consistently good results had come from the tests in the A-wing, we decided the dataset collected from it would service the experimentation the best. In the following we present residual distance errors for different fingerprint lengths. The dataset was pruned as in the original testing, meaning APs that consistently provided weak data were removed. This essentially meant truncating the dataset so that APs had an average signal strength above -100 dBm. This process reduced the amount of APs in the dataset from 37 to 16, meaning over half of the APs in the original set were providing weak data. Excluding these distracting APs from testing removes some variance in accuracy, since the only variable will be the length of the fingerprints.

In addition to truncating the fingerprints, we started with a minimum fingerprint length of two since a single signal strength measurement could obviously not translate to a coordinate point on a two-dimensional map. Isomap was also constrained to consider only up to 50 neighbors in an effort to ensure the tests could be run within a sensible time frame. Since Isomap even in the worst case only needed 15 neighbors for its optimum mapping in the initial tests, we felt confident in using this constraint here.

The set of APs to remove from the dataset was chosen at random, and the process was repeated 100 times. We present the maximum, minimum and average error achieved during these iterations.

From the results (Figure 24) we can tell that increasing fingerprint length indeed decreased the average error of the embedding. The largest jump is seen as we move from a fingerprint length of two to three or four. After this the accuracy stabilizes somewhat, but decreases on average throughout. After the fingerprint length is increased to ten only the maximum error decreases, meaning we are mostly improving the robustness of the calibration at this point. This seems to agree with the findings of Roos et al. in that ten APs (appropriately placed) give fingerprints enough distinction for positioning purposes [RMT02]. They are, however, able to complement fingerprint length with a longer history, achieving moderate to good results even with three APs. In our approach a fingerprint this short seems to be detrimental to the accuracy, most likely because they are not given enough distinction to maintain a pairwise distance relation. We could only achieve good accuracy in this case if the appropriate APs are chosen, as indicated by the minimum error. In the best case we could achieve a residual calibration error of two meters, but a bad set of APs could incur an error of up to almost eight meters as depicted by the maximum error.

Please note that the sudden jump in the graph when we reach fingerprint length 16 is simply because we at this point are using all APs in the dataset. The maximum, minimum and average accuracies of 100 runs are thus exactly the same, since the order of APs has no effect on calibration.

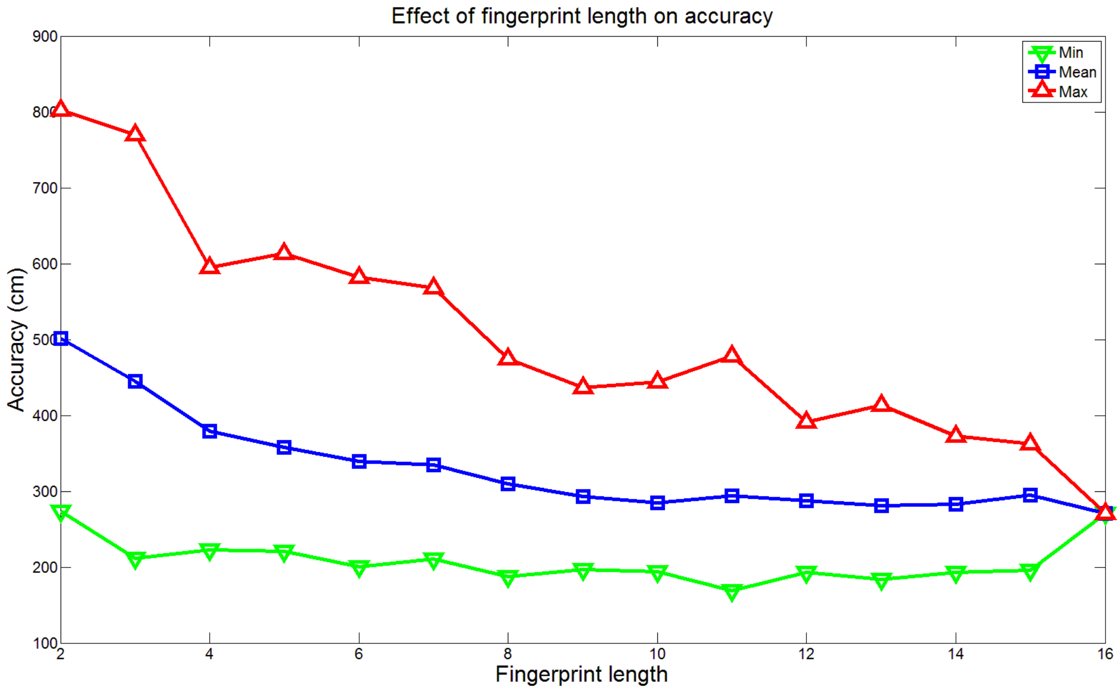

Figure 24: Results from fingerprint length comparison.

Next, the target of decimation were the measurement points themselves. Since there are 476 measurement points in this dataset, running extensive testing on all of them would have been a prohibitively time-intensive process. We thus chose to increment the size of the testing set by five with each iteration (i.e. subsets of 5,10,...,476 of the original measurement points). Isomap also has problems creating a mapping with less than five points, meaning we tested starting from that value. As the choice of measurement points in each testing set was random, the random sampling was performed 100 times for each testing set. This ensured that the values recorded were not coincidental for the particular testing set configuration. For each testing set the maximum, minimum and average value of the embedding error was recorded.

It appears that although the average accuracy remains somewhat stable throughout, there's a clear trend towards improved accuracy as measurement points are increased, as seen in Figure 25. Since the mapping regardless of measurement point density contains 38 key points, the first 40 or so measurement behave at odds with the rest of the data. There is even a sharp increase in the minimum accuracy, which can only be attributed to the fact that the key points controlled the accuracy until this point. Even in the worst cases we only reach a maximum embedding error values of less than two meters. The minimum error graph always picks the best result out of the 100 runs which naturally means its trend is not as clear as the maximum and mean values. Starting from about 100 measurement points we see no distinct improvement in any of the graphs until 300 or so measurement points are used. After this a clear improvement is achieved up to and including calibrating with the entire dataset. As with the fingerprint study, we see a sharp jump at the last iteration. Since all measurements are used at this point, no amount of randomizing is going to separate the maximum, minimum and average values.

Overall, calibrating using the same key points in all runs means the embedding is adjusted well no matter how many measurement points it is surrounded with. Once a sufficient density has been achieved adding measurement points does not improve the accuracy significantly until we reach another peak in density. After this we see more improvement with the inclusion of 100 measurement points than we saw with adding 200 earlier.

Roos et al. performed a similar decimation procedure in their article [RMT02]. Though the general conclusion seems the same, in that more measurement points improve accuracy, their approach entailed making sure remaining points were evenly distributed across the measurement environment. Since our application is completely random in its pruning procedure, however, it seems somewhat natural that the average embedding error remains a similar value throughout most of the data. We can only tell that on average, the amount of measurement points is not absolutely detrimental to the positioning procedure, once we pass a certain limit of density. A "smarter" distribution of subset points would most likely improve the average accuracy, as indicated by the minimum error graph. This could be considered a "best case" distribution, i.e. one where the measurement points were chosen in an optimal (uniform) way.

This barrage of testing has taught us that although fingerprint length has an effect on overall accuracy, we only need them to consist of about ten APs for them to stabilize in average accuracy. Longer fingerprints could be said to increase the accuracy of the embedding further, but only when the worst case is considered. The main effect of increasing fingerprint length seems to be to improve the robustness of the positioning.

Concerning measurement point density we find that as long as the points remaining

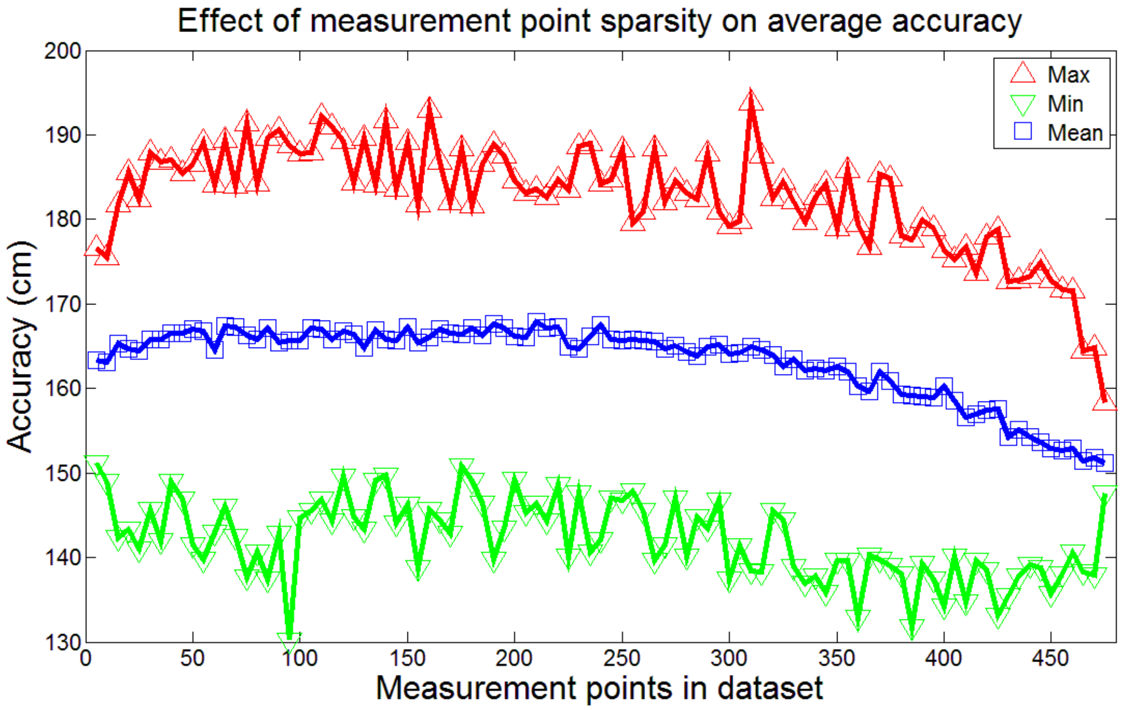

Figure 25: Results from measurement point density comparison.

in the data represent the measured environment well, the key points embedded in them are able to calibrate the embedding to good accuracy. It is thus not sparsity itself that is an issue, but a distribution of sparsity that is not uniform. As long as the measurement points are embedded in a way where they are covering the measured environment in a uniform way, we can maintain a standard accuracy. For optimum accuracy, however, it is recommended that measurement points are measured as densely as possible. A clear benefit was shown when measurement point density increased, though this improvement was not apparent until about 75% of measurement points were included.

7 Conclusions and Future Work

Manifold learning algorithms work to decrease the dimensions of otherwise high-dimensional data by finding local relations and extending them to a global scale. The Isomap algorithm is able to work on a dissimilarity matrix of the points on the manifold and create an embedding in a lower dimension, with distances between neighboring points maintained as well as possible.

Traditional WLAN positioning uses signal strength values gathered from access points in the environment. Whether by probabilistic modelling or supervised learning, algorithms are able to place a sample fingerprint in the trained environment by various proximity measures. Previous approaches have battled with a costly training and calibration process, however. Though some efforts have been made to simplify the training process, the calibration has always been performed under the constraint of perfect world knowledge. Training and calibration have needed careful planning and systematic measurements.

We have presented a WLAN positioning approach that is able to embed recorded RSSI fingerprints in a two-dimensional coordinate space through manifold learning, similar to the manifold-based approach of [PaY07]. By additionally measuring and labelling specific key points and embedding them along with the original dataset, we are able to automatically attach the manifold to a real-world coordinate system associated with the mapped environment. Due to the semi-supervised nature of the approach, requiring only a partially labeled dataset, the process of data collection is made significantly easier than through a fully supervised learning. Though some distortion can be discerned from the resulting mapping and positioning tests, the methodology itself shows a lot of promise. The approach could help especially in areas where determining the exact location of all measured fingerprints is either prohibitively cumbersome (such as large indoor spaces), expensive or intrusive. Since the fingerprint measures themselves could be collected automatically, they could be acquired by an automaton enabling a larger dataset (incurring a higher accuracy). This is especially useful for collecting measurements in an environment that might be disturbed by human involvement.

We performed tests in two separate environments, as well as different conditions and measurement parameters. By comparing two datasets from the same location we were able to discern that recording testing and calibration datasets on different days and times of the day has an effect on positioning accuracy. More importantly, however, we found that this effect was minimal and was usually marginalized by problems in the calibration performance. A calibration that is performed with care and optimized parameters should remain robust even during crowding and under a long period of time.

Positioning using RSSI fingerprints can achieve good accuracy, if the problems inherent to the methodology are considered and handled well. Traditional WLAN positioning approaches all suffer from the noise caused by a real-world environment, whether caused by measuring procedure or sources of distortion like movement and bandwidth crowding. Since our approach depends on the dissimilarity between the recorded fingerprints, it is especially volatile to fluctuations in the signal strength. Errors are compounded if the key points used to adjust the mapping are not representative of their positions. Any distortion caused in embedding them is directly transferred to the complete mapping, since calibration only considers these specific points. A lone fingerprint straying arbitrarily away from the mass of mapped points can skew the map if it's used as a calibration point. Since the calibration is based on a least-squares fit, optimizing the entire map based on this outlier means the bulk of the fingerprints will be mapped with heavy distortion. Improvements would likely include measuring key points several times and performing model selection based on them. Key points could also be determined by involving human interaction in the selection; key point fingerprint measurements could be constrained to try to match the previously measured environment, in that a key point fingerprint should not be allowed to differ too greatly from established fingerprints in the same area. This would naturally entail keeping a record of where previous measurements have been done, which would undermine some of the strengths of the semi-supervised system.

The current implementation is based on measurements performed while standing still. Future improvements could thus include training while in motion, provided the survey frequency could be increased. A successful training in motion carries several benefits. The amount of time required for the complete mapping of an environment is naturally decreased radically. In addition, with a higher sampling speed a form of tracking could be performed. In its current form the slowest component of the positioning system is the initial Isomap embedding. Once this embedding has been successfully implemented and adjusted, the actual comparison and placing of sampled testing points is relatively quick and painless. Finally, training in motion could mean that the environment mapping could be automated. Since Isomap is indiscriminate to the ordering of measurement points the environment could be mapped in arbitrary order. In essence, this could entail delegating this time-consuming and monotonous survey procedure to a moving automaton. A roving vacuum cleaner like the Roomba by iRobot could perform the environment survey with no additional programming needed.

By posing constraints on where fingerprints are allowed to be embedded the resultant mapping could be improved. Specialized environments could find positions constrained to a rail along paths useful. By decreasing the granularity and separating the measured area into a grid of cells, fingerprints could be designed to represent an entire area. This could be useful in areas where point-precision isn't as important as specific sections, such as in a retail environment. Map-matching could ensure that the measured points respect the infrastructure and are not mapped inside walls or outside of the building.

References

BaP00 Bahl, P. and Padmanabhan, V. N., RADAR: An in-building RF-based user location and tracking system. *International Conference on Computer Communications*, volume 2. IEEE, 2000, pages 775–784.

CoC01 Cox, T. F. and Cox, M. A. A., *Multidimensional scaling.* Chapman and Hall, 2nd edition, 2001.

DeT02 de Silva, V. and Tenenbaum, J. B., Unsupervised learning of curved manifolds. *Proceedings of the Mathematical Sciences Research Institute Workshop on Nonlinear Estimation and Classification*, Denison, D. D., Hansen, M. H., Holmes, C. C., Mallick, B. and Yu, B., editors. Springer Verlag, 2002, pages 453–466.

EKA Ekahau, Inc. RTLS. URL `http://www.ekahau.com`.

FHF06 Ferris, B., Hahnel, D. and Fox, D., Gaussian processes for signal strength-based location estimation. *Robotics: Science and Systems*, Sukhatme, G. S., Schaal, S., Burgard, W. and Fox, D., editors. The MIT Press, 2006, pages 1–8.

FWC05 Farivar, R., Wiczer, D., Gutierrez, A. and Campbell, R. H., A statistical study on the impact of wireless signals' behavior on location estimation accuracy in 802.11 fingerprinting systems. *International Symposium on Parallel & Distributed Processing.* IEEE Computer Society, 2009, pages 1–8.

Gho06 Ghodsi, A., Dimensionality reduction a short tutorial. Technical Report, University of Waterloo, 2006.

HBO05 Harte, L., Bowler, D., Ofrane, A. and Levitan, B., *Wireless Systems.* Althos Publishing, first edition, 2005.

HFL04 Haeberlen, A., Flannery, E., Ladd, A. M., Rudys, A., Wallach, D. S. and Kavraki, L. E., Practical robust localization over large-scale 802.11 wireless networks. *Annual International Conference on Mobile Computing and Networking*. ACM, 2004, pages 70–84.

HHH09 Hyvärinen, A., Hurri, J. and Hoyer, P. O., *Natural Image Statistics - A Probabilistic Approach to Early Computational Vision*. Springer, preprint edition, 2009.

KaK04 Kaemarungsi, K. and Krishnamurthy, P., Properties of indoor received signal strength for WLAN location fingerprinting. *Annual International Conference on Mobile and Ubiquitous Systems*. IEEE Computer Society, 2004, pages 14–23.

LiZ06 Lim, A. and Zhang, K., A robust RFID-based method for precise indoor positioning. *International Conference on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems*, Ali, M. and Dapoigny, R., editors, volume 4031, 2006, pages 1189–1199.

NLL04 Ni, L. M., Liu, Y., Lau, Y. C. and Patil, A. P., LANDMARC: Indoor location sensing using active RFID. *Wireless Networks*, 10,6(2004), pages 701–710.

PaC09 Papapostolou, A. and Chaouchi, H., WIFE: Wireless indoor positioning based on fingerprint evaluation. *Networking*, volume 5550. Springer Berlin / Heidelberg, 2009, pages 234–247.

Pea01 Pearson, K., On lines and planes of closest fit to systems of points in space. *Philosophical Magazine*, 2,6(1901), pages 559–572.

PKA07 Petrellis, N., Konofaos, N. and Alexiou, G., A wireless infrared sensor network for the estimation of the position and orientation of a moving target. *International Conference on Mobile Multimedia Communications*. Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering, 2007, pages 1–4.

PLY00 Pahlavan, K., Li, X., Ylianttila, M., Chana, R. and Latva-aho, M., An overview of wireless indoor geolocation techniques and systems. *IFIP-TC6/European Commission International Workshop on Mobile and Wireless Communication Networks*. Springer-Verlag, 2000, pages 1–13.

PaY07 Pan, J. J. and Yang, Q., Co-localization from labeled and unlabeled data using graph Laplacian. *International Joint Conference on Artifical Intelligence*. Morgan Kaufmann Publishers Inc., 2007, pages 2166–2171.

RMT02 Roos, T., Myllymäki, P., Tirri, H., Misikangas, P. and Sievänen, J., A probabilistic approach to WLAN user location estimation. *International Journal of Wireless Information Networks*, 9,3(2002), pages 155–164.

CHC07 Shao, C., Huang, H. and Wan, C., Selection of the suitable neighborhood size for the Isomap algorithm. *International Joint Conference on Neural Networks*. IEEE, 2007, pages 300–305.

SMR06 Samko, O., Marshall, A. D. and Rosin, P. L., Selection of the optimal parameter value for the Isomap algorithm. *Pattern Recognition Letters*, 27,9(2006), pages 968–979.

SaR03 Saul, L. K. and Roweis, S. T., Think globally, fit locally: Unsupervised learning of nonlinear manifolds. *Journal of Machine Learning Research*, 4,1(2003), pages 119–155.

TSL00 Tenenbaum, J. B., de Silva, V. and Langford, J. C., A global geometric framework for nonlinear dimensionality reduction. *Science*, 290,5500(2000), pages 2319–2323.

WFG99 Want, R., Fishkin, K. P., Gujar, A. and Harrison, B. L., Bridging physical and virtual worlds with electronic tags. *SIGCHI Conference on Human factors in computing systems*. ACM, 1999, pages 370–377.

YoM05 Youssef, M. and Agrawala, A., The Horus WLAN location determination system. *International Conference on Mobile Systems, Applications, and Services*. ACM, 2005, pages 205–218.

YAS03 Youssef, M., Agrawala, A. and Shankar, A. U., WLAN location determination via clustering and probability distributions. *Pervasive Computing and Communication*. IEEE Computer Society, 2003, pages 1–8.

YeN07 Yeung, W. M. and Ng, J. K., Wireless LAN positioning based on received signal strength from mobile device and access points. *International Conference on Embedded and Real-Time Computing Systems and Applications*. IEEE Computer Society, 2007, pages 131–137.

YZN07 Yeung, W. M., Zhou, J. and Ng, J. K., Enhanced fingerprint-based location estimation system in wireless LAN environment. *Embedded and Ubiquitous Computing Workshops*, volume 4809. Springer, 2007, pages 273–284.

ZhZ07 Zhang, M. and Zhang, S., An accurate and fast WLAN user location estimation method based on received signal strength. *International Conference on Computational Science*. Springer-Verlag, 2007, pages 58–65.