Positioning for mobile phones using WLAN and accelerometer data

Viktor Moell och André Horntvedt

Master's thesis
2012:E1

Lund University

Faculty of Engineering
Centre for Mathematical Sciences
Mathematical Statistics

CENTRUM SCIENTIARUM MATHEMATICARUM

TYP AV DOKUMENT ☒ Examensarbete □ Kompendium □ Delrapport □ Rapport	**DOKUMENTBETECKNING** **LUTFMS-3186-2012**

INSTITUTION

Matematikcentrum. Matematisk statistik, Lunds universitet, Box 118, 221 00 LUND

FÖRFATTARE

Viktor Moell och André Horntvedt

DOKUMENTTITEL OCH UNDERTITEL

Positioning for mobile phones using WLAN and accelerometer data

SAMMANFATTNING

In recent years, techniques for using a Wireless LAN for indoor positioning of mobile phones have been developed and commercialized. This thesis analyses the possibilities of using the built-in sensors of "smart" mobile phones for improving the accuracy of position estimates. We've focused on accelerometers due to their widespread prevalence in today's mobile phones.
By using the accelerometer, it is possible to determine whether or not the user is stationary. When stationary, the noise present in the received signal strengths of the nearby wireless access points can be eliminated by averaging them over time. This leads to improved accuracy in position estimates.

NYCKELORD

DOKUMENTTITEL OCH UNDERTITEL - SVENSK ÖVERSÄTTNING AV UTLÄNDSK ORIGINALTITEL

Inomhuspositionering med hjälp av WiFi och accelerometer

UTGIVNINGSDATUM år 2012 mån	**ANTAL SID**	**SPRÅK** □ svenska ☒ engelska □ annat

ÖVRIGA BIBLIOGRAFISKA UPPGIFTER	**ISSN**
	ISBN
	2012:E1

I, the undersigned, being the copyright owner of the abstract, hereby grant to all reference source permission to publish and disseminate the abstract.

Signature ______________________________ Date ______________

Abstract

In recent years, techniques for using a Wireless LAN for indoor positioning of mobile phones have been developed and commercialized. This thesis analyses the possibilities of using the built-in sensors of "smart" mobile phones for improving the accuracy of position estimates. We've focused on accelerometers due to their widespread prevalence in today's mobile phones. By using the accelerometer, it is possible to determine whether or not the user is stationary. When stationary, the noise present in the received signal strengths of the nearby wireless access points can be eliminated by averaging them over time. This leads to improved accuracy in position estimates.

Acknowledgements

We would like to thank our supervisor, professor Andreas Jakobsson at the department of Mathematical Statistics at Lund University, for giving us the opportunity to work on this project, and for his continuous support.

We would also like to thank Brice, Mikael, Giovanni, and Olof for their support regarding the technical aspects of this thesis.

CONTENTS

Chapter 1

INTRODUCTION

Not all those who wander are lost.
- J. R. R. Tolkien

During the last few years the market of mobile telephones have been revolutionized. With the launch of the Iphone, "smart" mobile telephones became available to a much broader audience than ever before. This in turn led to a revolution in how we view software on mobile platforms. The performance of these new types of telephones also enabled developers to use conventional developing languages to make applications for these devices, or "apps", with a much less demanding technical background.

1.1 Wireless Positioning Systems for mobile platforms

Global Positioning System (GPS) is a positioning system based on satellites broadcasting a signal that can be used to triangulate the position of an object on the ground. Today GPS is available in very many different forms, in hand-held GPS devices, in vehicles, and in many advanced mobile telephones. Unfortunately, GPS does not work indoors, something that sometimes is very relevant for pedestrians. Thus the idea of using other signals for triangulation, particularly WiFi was conceived see eg. A. Kushki (2007) [9] and the references therein for further background.

In an area with an active wireless local area network (WLAN), it is possible to measure the received signal strength index (RSSI) from each WLAN access point (AP), thereby obtaining a signal profile or "fingerprint" of a position. Multiple such measurements can be combined into a "map" that can be stored. With these measurements in place, the position of a mobile device can be estimated by comparing the current RSSI values of all available APs with the RSSI values of the stored positions and interpolating the current position. The same method can be used with GSM signals; however, since these signals are transmitted much more seldom they are not as dependable as WLAN signals when it comes to positioning.

This means that for any area in which this system is to be used, one must first obtain location fingerprints by surveying the site, obtaining RSSI values for known locations.

1.2 Issues with RSSI position estimates

Just like any other real-world signal, the measured WLAN signals contain noise. Thus, all position estimates will have errors resulting from this noise. This error is especially noticeable when the mobile phone is stationary, where a user would expect that the position estimate remains constant. In practice however, the random noise part of the RSSI will make the position estimate vary. The aim of this thesis is to alleviate this problem in two steps:

- use available sensor data in order to estimate the movement of the mobile phone.
- apply averaging methods in order to reduce or eliminate the variance of positioning estimates while stationary.

There is currently a lot of research being done to improve these kind of systems; see, for instance, Kushki (2010) [8], Seitz *et al.* [5] and Evennou and Marx (2006) [6].

The second chapter of this report explains the problem and its possible solutions in more detail, along with some theoretical background. The third chapter contains a look at data from a current positioning system, along with analysis of needed improvements. The fourth chapter analyses the application of a Kalman filter. The fifth chapter explains how to determine whether a mobile device is stationary, and an initial approach at improving accuracy, while the sixth chapter investigates other possible means of reducing the position estimate error while stationary. A very basic idea of the implementation on an actual mobile phone is given in Chapter seven. The eighth chapter contains the measured results. Finally we summarize our findings in Chapter nine which contains our conclusions.

Chapter 2

PROBLEM DESCRIPTION AND PREVIOUS WORK

2.1 Problem description

In this thesis, we will examine the possibilities of using sensor data to improve the positioning estimates, specifically:

- Is it possible to ascertain whether or not the device is still, thereby lessening the need for position updates?
- Is it possible to determine in which direction the user is going?
- If we know the direction, can we speed up the algorithm by predicting what position the user will have next?
- Can we use the direction to make the positioning updates less jittery and more accurate?

2.2 Previous work done on positioning based on RSSI readings

There are several commercially available systems using RSSI value averaging for indoor positioning [13–16]. Since our work builds on existing positioning solutions, but poses no constraints on the actual solution, we have chosen not to include the exact ways the RSSI averaging and positioning systems work. However in the latter part of this thesis, we do use the gathered RSSI data instead of rendered positions, which means that there is a requirement to at least understand the basis of the underlying process.

The positioning algorithm works by measuring the signal strength of all available signals within the WLAN, indexing all of them and disregarding any signal that is bellow a certain value, as it is most likely just noise. These values are then forwarded to a positioning algorithm that uses the pre-rendered mapping to estimate the position of the receiver. More about how these systems work can be found in, for example, Fang *et al.* (2010) [7] or Soldi (2011) [1].

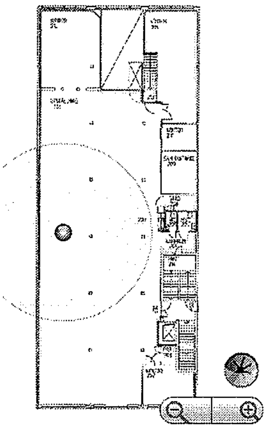


Figure 2.1. A mobile RSSI positioning System

2.3 The Kalman filter

A Kalman filter is a mathematical method that uses previously observed data to help estimate, predict and interpolate the internal state of a state space representation of a process [12]. This is done by weighing previous steps or observations of the process and adding a stochastic noise element. This can be done solitary on previous output data, or with output data from an unseen underlying process, or in some cases with a process with a separate input data. For this particular application a Kalman filter without an input works essentially by determining the current position weighted average of the earlier positions and the current update with a added noise element.

To create a Kalman filter we first require a state space model, i.e. a model of the underlying process that describes the process as whole [2] :

$$X_t = AX_{t-1} + Bu_{t-1} + e1, t$$

$$Y_t = CX_t + e_{2,t}$$

where: X_t is the hidden, "real" process, Y_t is the observations of the process, A, B, C are constant matrices and $e_{n,t}$ is the noise processes where $Var[e_{1,t}] = \Sigma_1, Var[e_{2,t}] = \Sigma_2$

The optimal *reconstruction* $\hat{X}_{t|t}$ and *prediction* $\hat{X}_{t+1|t}$ of the system is obtained by a *reconstruction* (updating) [2]

$$\hat{X}_{t|t} = \hat{X}_{t|t-1} + K_t(C\hat{X}_{t|t-1}),$$

$$\Sigma^{xx}_{t|t} = \Sigma^{xx}_{t|t-1} - K_t \Sigma^{yy}_{t|t} K_t^T = \Sigma^{xx}_{t|t-1} - K_t C \Sigma^{xx}_{t|t-1},$$

where the Kalman gain is

$$K_t = \Sigma^{xx}_{t|t-1} C^T (\Sigma^{yy}_{t|t-1})^{-1},$$

and the *prediction* is

$$\hat{X}_{t+1|t} = A\hat{X}_{t|t} + Bu_t,$$

$$\Sigma^{xx}_{t+1|t} = A\Sigma^{xx}_{t|t} A^T + \Sigma_1,$$

$$\Sigma^{yy}_{t+1|t} = C\Sigma^{xx}_{t+1|t} C^T + \Sigma_2,$$

With initial condition

$$\hat{X}_{1|0} = E[X_1] = \mu_0$$

$$\Sigma^{xx}_{1|0} = Var[X_1] = V_0$$

We will use Kalman filters in Chapter 4.

2.4 Positioning sensors in mobile phones

Modern mobile phones usually contain multiple positioning sensors. Most common are accelerometers, but more advanced models can also contain gyroscopes and compasses, although usually only one of the two.

2.4.1 Accelerometers

Accelerometers are sensors that detect linear acceleration. Most phones have at least three, usually oriented in X, Y, and Z so that the readings form an orthogonal coordinate system. This information can be relatively useful, as it can be used to determine in which direction the device is held; however this only indicates which way is down, and gives no way of knowing which XYZ-direction the device is currently held in, or moves at.

2.4.2 Gyroscopes

Gyroscopes are devices used to determine orientation. In many exact sciences, they are used in place of other methods, like for example compasses, because of their better resolution. Most digital devices use a MEMS gyroscope, which is a chip-mounted electrical gyroscope. They are fairly advanced sensors, and while relatively cheap to build, they require significantly more power to operate than an accelerometer. We have not had any mobile devices with gyroscopes available during our research, but almost any work done with compasses can be done with gyroscopes as well.

The problem discussed above about accelerometers not being able to distinguish coordinate acceleration from gravity could be solved using gyroscopes. Gyroscopes are far from commonplace in today's mobile devices, though, and thus we chose not to rely upon these to provide additional data about orientation.

2.4.3 Compass

Digital compasses work the same way as their analogue counterparts by measuring the earth's magnetic field and can thereby finding true north. By using this and accelerometers all other directions can be calculated. These measurements are fairly exact, despite the fact that they depend on multiple different sensors. This is mainly because of the fact that both gravity and the earth's magnetic field are comparatively strong forces with a very exact orientation. Many modern smart phones have abandoned the use of gyroscopes in favour of compasses for this reason. Even so the compass requires much more battery power than accelerometers, a fact that is something of a limitations in all applications that are built to be used long periods of time.

2.5 What the accelerometer output really means

Even though an accelerometer measures acceleration, it cannot do so directly. In practice, the accelerometer can be viewed as in Figure 2.2; a mass suspended by a spring [10], where acceleration is determined by measuring the displacement of the mass, according to:

$$x = kF = kma \Rightarrow a \propto x$$

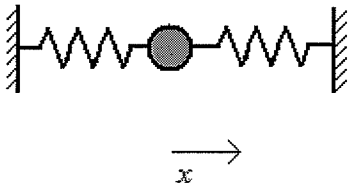


Figure 2.2. A conceptual sketch of an accelerometer

This means that the displacement due to coordinate acceleration and the displacement due to gravity are indistinguishable, and both register as acceleration [3].

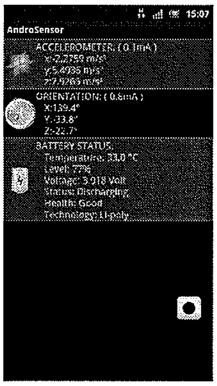
Thus, we conclude that it is impossible to predict movement direction using only the accelerometer. However, there is a connection between accelerometer readings and movement. More on this in chapter 5.

2.6 Sensor software

There is a variety of software available on the market to monitor and record sensor readings on phones. Most software use the built-in sensor recording cycles that comes with all devices that run the Android operating system. These cycles are Normal (4 Hz), UI (10 Hz), Game (20 Hz) and Fastest (40 Hz). The exact frequency of the readings vary a little between manufactures, but is usually somewhat similar. We found that one of the most important parts when choosing what software to use is how the data is saved. The program we used (AndroSensor) saved it's data in a csv format which made data processing very much easier, since it is already in matrix form.

2.7 Inertial navigation in general

There are several different inertial navigation systems based on sensor data available today. Most of these systems work with more or less the same basic idea of inte-



AndroSensor

ACCELEROMETER: (0.1mA)
X:-2.2759 m/s²
Y:5.4936 m/s²
Z:7.9263 m/s²

ORIENTATION: (0.8mA)
X:139.4°
Y:-33.8°
Z:-22.7°

BATTERY STATUS:
Temperature: 33.0 °C
Level: 77%
Voltage: 3.918 Volt
Status: Discharging
Health: Good
Technology: Li-poly

Figure 2.3. The AndroSensor application showing accelerometer and compass readings.

grating sensors [4]. The use of accelerometers along with gyroscopes and compasses for navigation is nothing new. However, since mobile phones do not always contains all these sensors (and those available are far less accurate), and since mobile phones are much more sensitive to battery drain, we have had to factor in this into our work. This means that using accelerometers for positioning with a mobile phone will be very different from much of the previous work done in the field.

Chapter 3

INITIAL DATA ANALYSIS

3.1 Location and accelerometer data from initial tests at Hansa Shopping Center

The initial data given at the start of this thesis work was log files from a walk around the Hansa shopping center in Malm. By preliminary analysis of this data, it would then be possible to examine possibilities and limitations of the current system. Also, it would determine if any change needed to be made regarding sensor input.

In Figure 3.1 and Figure 3.2 are the position estimates compared with the actual walked route. The starting points are marked with a circle. The squares mark stops, where the user has waited for 10-15 seconds before continuing. As can be seen, the position estimates have a large variance. Note especially how the position estimates vary even when standing still.

The initial data set also contained accelerometer data, which can be seen in

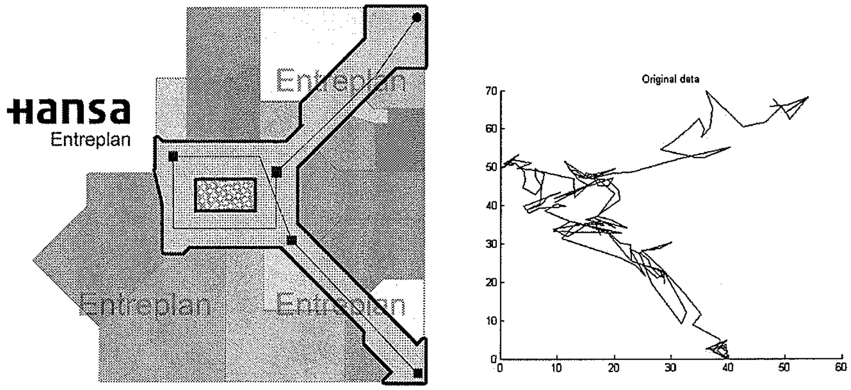


Figure 3.1. Comparison between walked route and estimated positions for the ground floor.

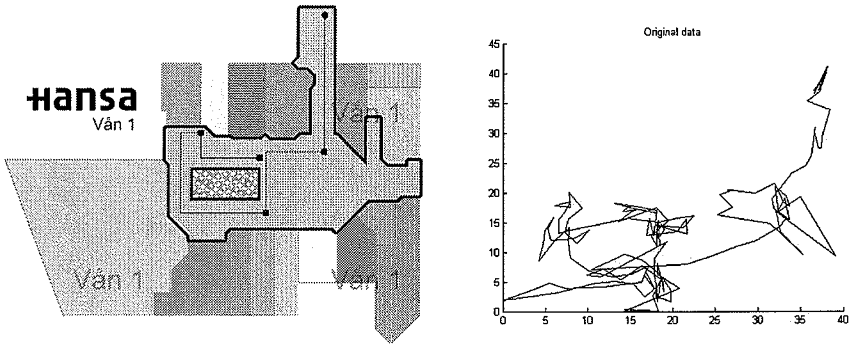


Figure 3.2. Comparison between walked route and estimated positions for the first floor.

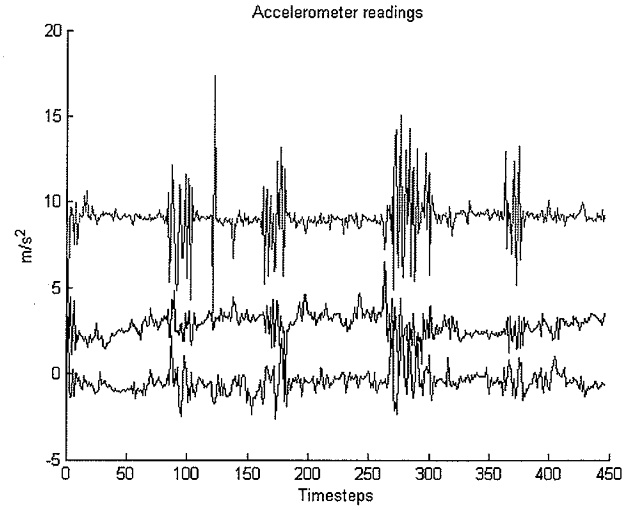


Figure 3.3. The accelerometer readings. x-, y-, and z-direction are represented as blue, black and red, respectively

Figure 3.3.

Unfortunately, since the sampling rate is too low (less than 1Hz for both location data and accelerometer data), it is not possible to draw any conclusions about the correlation using only this initial data. Thus, the need to gather better data arises. It also did not contain a true position of the measurements, so we can only compare our results with the original data, and not with the true positions.

Chapter 4

APPLICATION OF A KALMAN FILTER

Much of our work with constructing a Kalman filter was limited by the fact that for the most part of our research we lacked a true position of reference to calibrate our filter with. This coupled with the fact that the positioning process is very noisy poses yet another problem; while the real process probably has some dependency on previous values, there was enough noise in the process for us not to be able utilize it without noise reduction. With true measurements it would be possible to build a model of the noise and then construct a better model for estimating positions.

Since we lack this, we decided to make a very simple state space model where we assume that all movement lies in the noise. We also denominate the positions that are generated by the Kalman filter as the observed positions (even though we know they are only better estimations), and use the RSSI position estimation to update this process. Finally, we assume that our estimate of the X coordinate is independent of our estimate for the Y coordinate. As the model is rather poor it will only hold up as long as we interpolates our observation data, any attempt to predict future movement will give unsatisfactory results.

4.1 Kalman filter without sensor inputs

Our state space model for each time step, where the RSSI position estimate of the X and Y coordinates are our states $X_{1,t}$ and $X_{2,t}$, and where Y is the "observations" (i.e. estimates) of the real process.

$$X_t = AX_{t-1} + e_{1,t}$$

and where our state spaces are expressed as follows:

$$X_t = \begin{pmatrix} X_{1,t} \\ X_{2,t} \end{pmatrix}$$

$$Y = CX_t + e_{2,t}$$

Since we assume zero correlation between X_1 and X_2, and we lack any knowledge of the correlation between the real and the measured positions, we use 2x2 unit matrices as A and C.

After some testing the variances of both the errors e_1 and e_2 was set to 15, a number that smooths the process enough to make it look better without distorting the average position needlessly.

We then used a standard Kalman filter methodology for obtaining the optimal reconstruction and prediction of the process as described in Chapter 2. The starting value for the covariance of the reconstruction error was set to 1. The initial values of the true position was set to zero, and the choice was made not to start the filter until we measured the second position. This gives us a slightly worse position to begin with, but the algorithm improves quickly, as there isn't very much dependency on older positions.

We built this Kalman filter with the purpose of improving it as we went along, preferably by having data with exact positioning. This would enable us to build a state space model that actually models the process, as opposed to one that models all of the movement as a noise process and nothing else. By the time we got the capability to do this (as we had access to neither smartphones, nor an actual positioning system at the start of the project), we had already built the movement detection system. We had the choice between implementing the movement detection system on a smartphone or building a better and/or a more advanced Kalman filter. We chose the former, and therefore we leave this part for future research.

4.2 Kalman filters with sensor inputs

If it can somehow be predicted how a person has moved when calculating the new position, there is a possibility of creating a Kalman filter that improves the positioning without creating a delay in the process. There are some constraints on how effective this improvement must be in order to be viable, as the current mobile phones have a severely limited battery capacity, and thus there is a limitation on how much calculation that can be done, and how often sensors can be used.

We decided to limit our research on this branch for two reasons. We were uncertain if the resulting filter would be useful, since the computational requirements on the phone would most likely be huge, which is a problem since there is a limitation on both computational power and battery capacity. We also had some problems coming up with a good working model to base our work on. The only model we came up with would be to use gyroscope to determine direction of movement, accelerometer for determine if there is movement and use the previous positions to estimate movement speed.

Chapter 5

ACCELEROMETER CUTOFF: A FIRST STEP

The traditional approach to using accelerometer data is by using dead-reckoning; that is, integrating the acceleration twice in order to get the change in position [4]. Unfortunately, when we tested this on a mobile phone, the error was several orders of magnitude. There are several reasons for why this will not work. First of all, we can make no assumptions regarding the initial orientation and velocity of the mobile device. Secondly, the accelerometer is not sampled often enough to get an accurate representation of the actual acceleration of the device. Thirdly, the error of the accelerometer in the average mobile phone is most likely too great to be reliable.

5.1 Detecting movement with accelerometers

Since it's impossible to determine the exact direction of movement with only an accelerometer, the simplest way to improve positioning estimates with an accelerometer is to determine if a person is moving or not, and change the update depending on this. However, determining if a person is moving is not that simple. Since there is no way of knowing exactly which way the device is oriented, there will always be a component of gravity divided among the three different directional sensors, as seen in Figure 5.1.

To compensate for this we must find a way to generalize the reading of all three sensors. The simplest way of doing this is using the Euclidean norm:

$$Norm = \sqrt{x^2 + y^2 + z^2}$$

This norm is independent of direction, and we can now plot the acceleration as a one-dimensional value. When the mobile device is stationary, the acceleration vector always points in the direction of gravity.

When a person walks with a mobile phone, the value of the accelerometer norm will change due to the way the device is held, since it is close to impossible to walk without causing some kind of lateral shifting, or tilting. It can easily be seen in Figure 5.2 that the norm shifts from a default value, which we will label as the base norm, when moving.

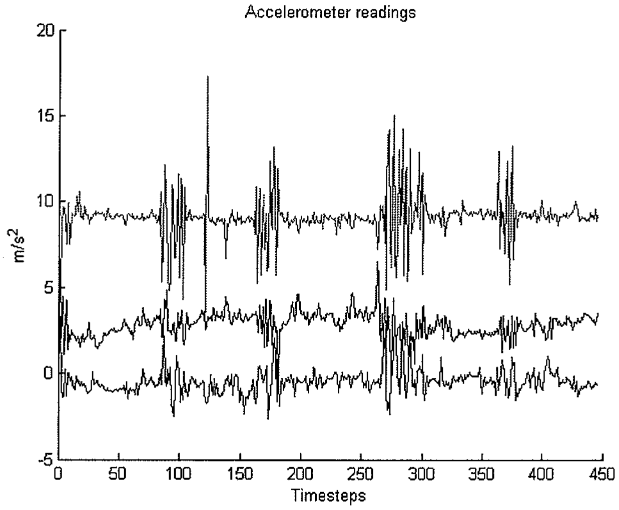


Figure 5.1. The accelerometer readings.

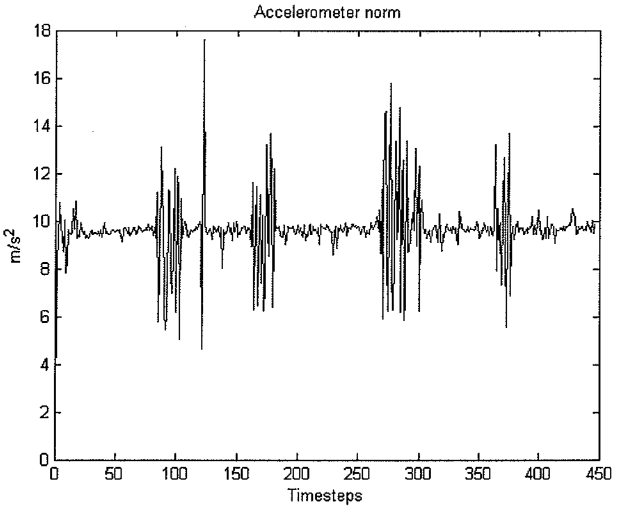


Figure 5.2. The Euclidean norm of the accelerometer readings in Figure 5.1.

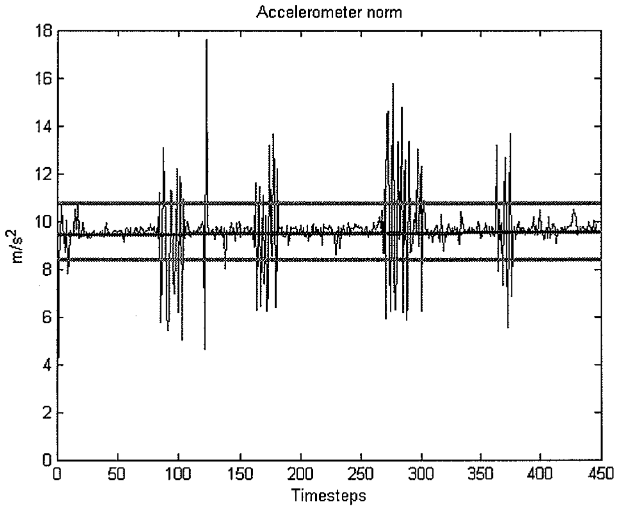


Figure 5.3. The range of movement detection. The movement detection bounds are marked with a purple line, and the norm is marked with a red line.

This base norm however is not exactly what one would expect. From a theoretical standpoint it should be equal to the gravitational constant, 9.82 m/s^2, but for the phones that we have tested, it is closer to 9.5. This is probably due to the inaccuracy of the accelerometers available in the mobile phone.

This makes it possible to define an exact criterion for detecting movement with the accelerometer. Whenever the accelerometer norm differentiates from the base norm more than a certain value, a movement has been detected. This threshold value is very important however, and must be chosen carefully, as the range when actual movement occurs is quite small.

As we can see in Figure 5.3, not every movement is detected, even with the very sensitive 1 m/s^2 threshold, but we can also see that most movement are much larger than this. Since all testing were preformed by two adults, we deliberately put this value as low as possible without incurring erroneous movement detection. Depending on the purpose of the application this value could definitely be recalibrated.

5.2 Accelerometer sampling frequency

By studying the accelerometer norm of a person walking (Figure 5.4), one can deduce that in order to have a resolution high enough to see the entire movement, an accuracy of the order 30 Hz would be required. However, in order to only register the larger peaks of movement, a much lower resolution can be used. It is important

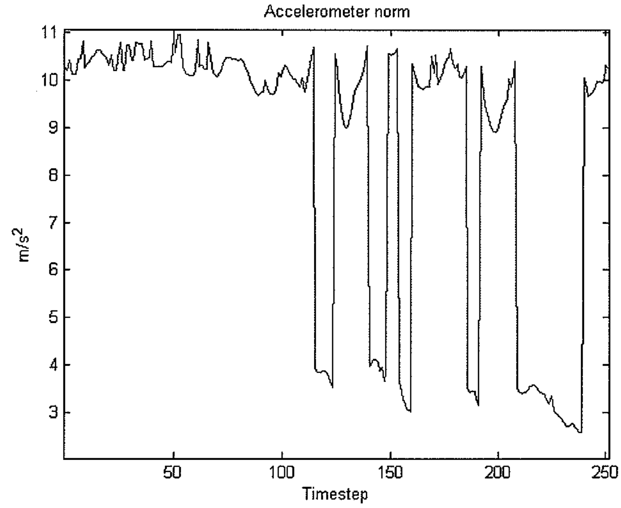


Figure 5.4. Excerpt from the norm of accelerometer readings of one of the authors walking. The sampling rate is 100 Hz.

to avoid an unnecessarily high frequency, as this increases the battery consumption.

5.2.1 Aliasing

Aliasing is the phenomenon that occurs when sampling a signal with a too low sampling rate. If an event occurs too close to the sampling rate, a systematic error can occur. If for example a cosine function is sampled every π seconds, the sampling will yield the same result every time. To avoid this we need a sampling rate of at least double the cycle of the process being sampled. When a person is moving, there is a sinusoidal peak that occurs at roughly 1 Hz. This means that the minimum sampling rate should be 2 Hertz, and preferably somewhat higher.

5.3 Discarding position estimates when no movement is detected

Our first attempt at decreasing the position jitter went as follows: Determine movement using accelerometer, and discard new position estimates unless movement has been detected. We also used a Kalman filter in order to smooth out the movements.

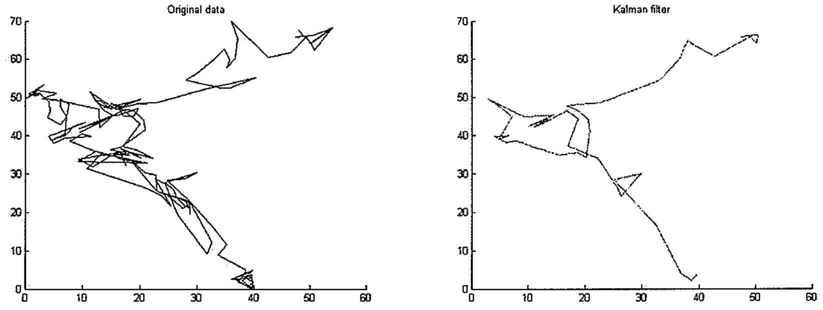


Figure 5.5. Position estimates for the ground floor of the Hansa shopping center.

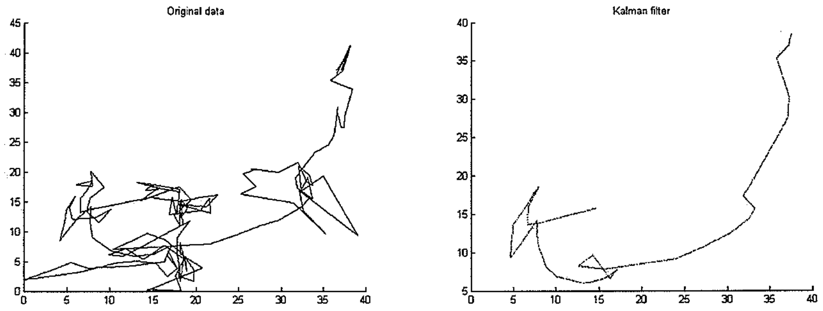


Figure 5.6. Position estimates for the first floor from the Hansa shopping center.

5.4 Applying accelerometer cutoff and Kalman filtering to initial data

The initial data from chapter 3 was subjected to accelerometer cutoff. The result can be seen in Figure 5.5 and Figure 5.6. The path of the position updates are obviously a lot smoother.

This, however, introduced a new problem: position update delay. By using plots with points rather than lines, shown in Figure 5.7 and Figure 5.8, we see that position estimates have been discarded even while moving, resulting in the position updates lagging behind.

5.5 Initial data gathering: Two phones

Initial data gathering was done by using two phones; one for logging position data, and the other for logging accelerometer data. By applying our method of deter-

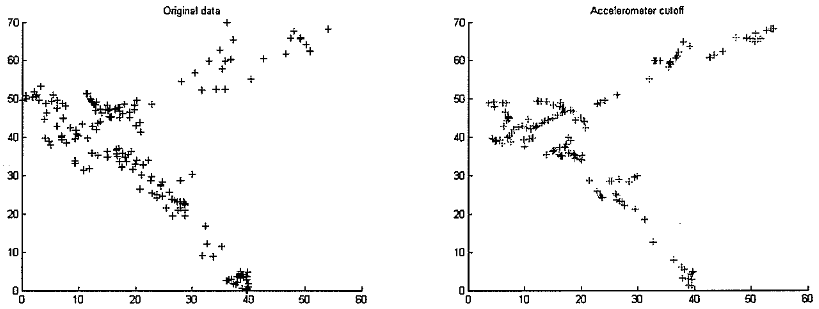


Figure 5.7. Position estimates for the ground floor.

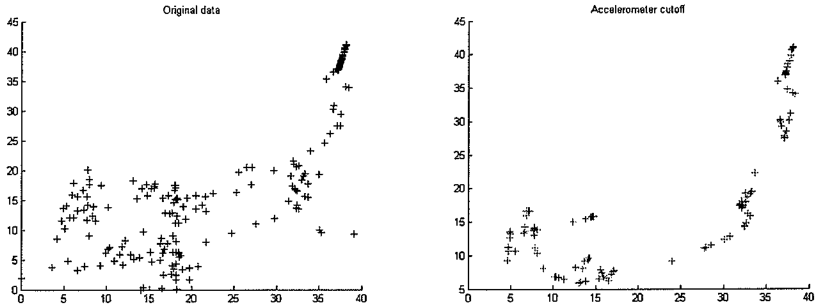


Figure 5.8. Position estimates for the first floor.

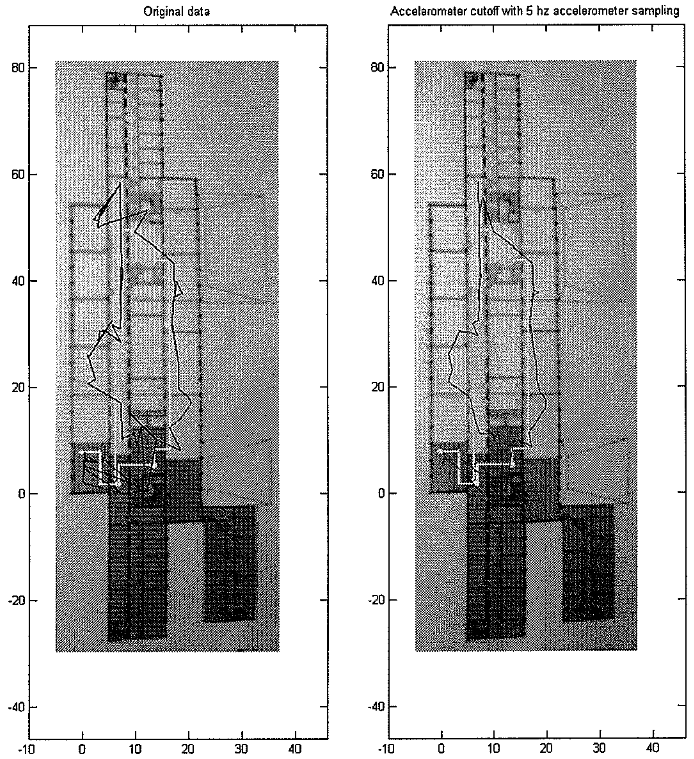


Figure 5.9. Comparison between unfiltered and filtered position estimates.

mining movement to the accelerometer data, and then synchronizing the two logs using time-stamps, we could discard some position estimates in order to simulate accelerometer cutoff in a real mobile application.

The location used was the second floor of the Mathematics Building at LTH. In the figures, the position estimates have been plotted on a map of the second floor, with the real path drawn in yellow. The circles on the path mark stops, where the user has paused for around 10 seconds in order to test the movement detection.

5.5.1 Analysis of path accuracy

In Figure 5.9, we see that the jitter has noticeably decreased after applying accelerometer cutoff.

By discarding some accelerometer log entries, we could also simulate another sampling frequency, shown in Figure 5.10. As can be seen, the 1 Hz movement

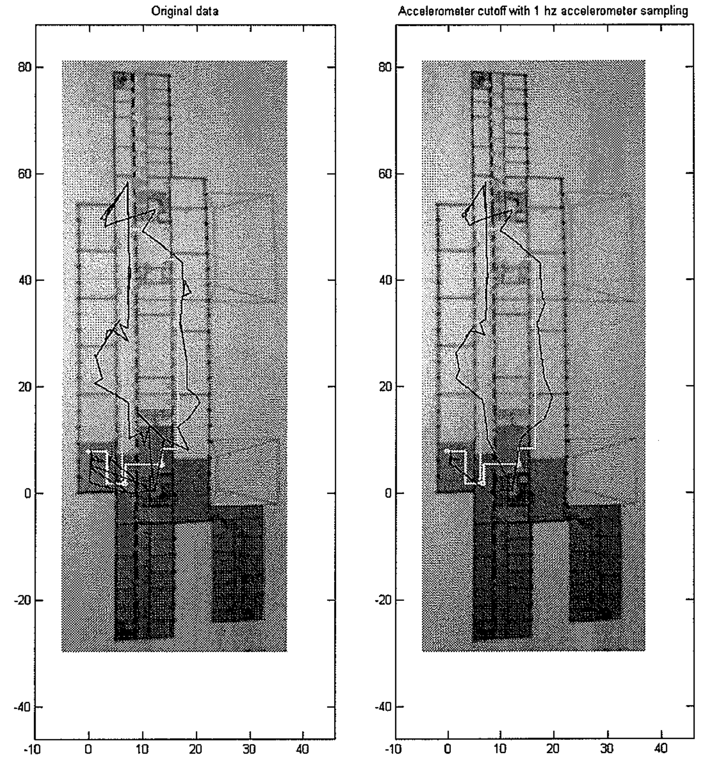


Figure 5.10. Comparison of different sampling rates.

seems even less jittery, but at the cost of missing some actual movement. This is, of course, not acceptable.

5.5.2 Analysis of update frequency

In order to measure the perceived delay of the position estimates, we have also compared these plots in point form, as shown in Figure 5.11. Here we distinguish that there is a very slight delay in position estimates as compared to the original data. This is due to the Kalman filter used for smoothing.

As above, we discarded some accelerometer data to simulate a 1 Hz sampling rate, which is shown in Figure 5.12. The algorithm now misses some of the instances where movement has started, and discards position estimates while moving, which leads to a delay in updates.

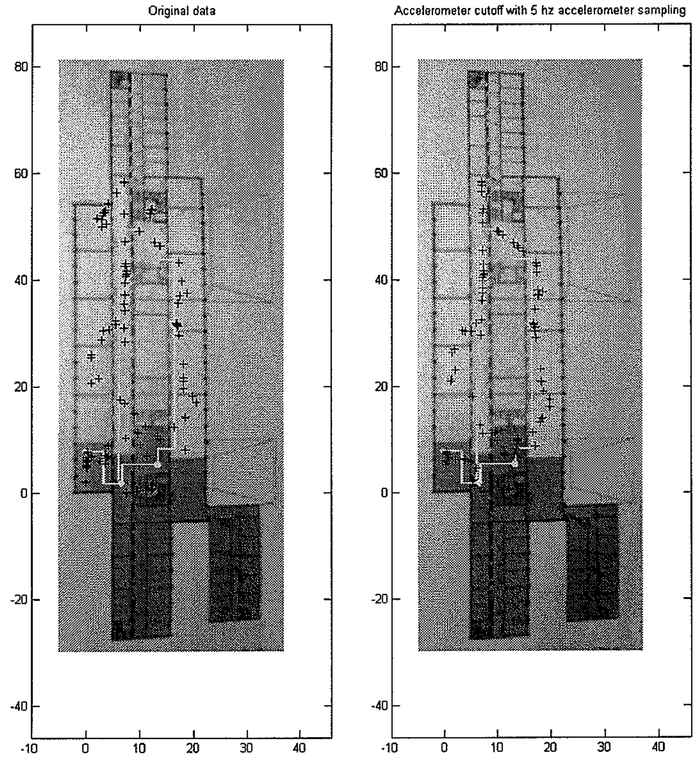
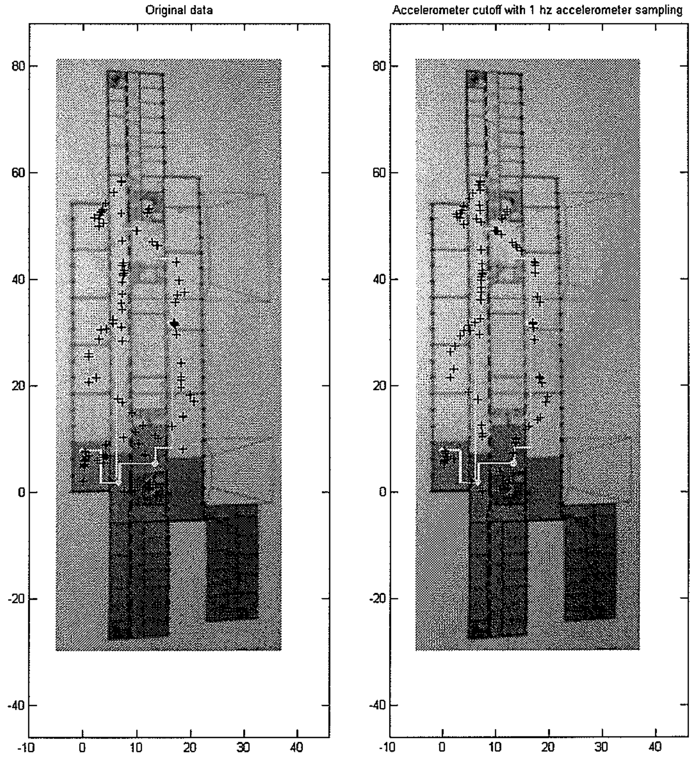


Figure 5.11. Comparison between unfiltered and filtered positions.



Original data

Accelerometer cutoff with 1 hz accelerometer sampling

Figure 5.12. Comparison of different sampling rates.

5.6 Conclusions from initial testing

While no extensive analysis has been done of exactly how high the accelerometer sampling rate needs to be, we can nonetheless conclude that the <1 Hz frequency of the data in chapter 3 is indeed a limiting factor. We also conclude that a 5 Hz sampling rate is good enough for determining movement, or lack thereof. This is also in line with the earlier work of Foster et al. [11], where a 4-5 Hz sampling rate was found sufficient for measuring walking speed using accelerometers strapped to the leg.

We can also note the slight delay in positioning updates originating from the Kalman filter. This is due to the fact that it only relies on older positioning estimates for predicting the new position estimate, and thus the new estimated position is shifted slightly towards the previous. The benefit of this is that the path is smoother. However, from a user's point of view, reducing delay might be of greater importance since only the latest position estimate is viewed in a positioning application.

Chapter 6

IMPROVING STATIONARY ACCURACY

Using the previously mentioned accelerometer cutoff method, it is possible to avoid updates while the device is not moving. Of course, completely discarding data is not an ideal approach, especially since the latest given position estimate might have been wrong. If possible, we'd like to use the discarded information in order to improve the accuracy of the current position over time. The information available is both the RSSI measurements and the resulting position estimates. We have studied different methods of using these in order to find the best solution, as described in more detail below.

The RSSI averaging methods ultimately proved to result in more accurate positioning while stationary. This is probably due to the lessening of the signal-to-noise ratio of the RSSI, which was previously too large for making accurate position estimates.

6.1 Interpolation of position estimates

Instead of discarding position estimates outright, they are stored until movement is detected again. By taking the mean of these values an average position is found. This leads to a positioning algorithm that approaches towards the true value. However, since we make the positioning calculations based on signals with high noise content, we will still get a rather large error. Also, the correlation between RSSI values and positions is not necessary linear, which can add another source of errors. Therefore, it would be advantageous to have some way of verifying that the new position estimate is actually better than the old one. To prevent us from missing any movement we only average over a certain number of recent points. In case the number of previous positions is too small, the shown position might "drift away"; if the number is too large, there will be a delay if the device is moved without it being detected. Therefore, it is important to choose the number of points carefully.

6.2 Finding clusters of position estimates

Another way to find a more correct position when stationary is to examine a few previously obtained position estimates and determine if they are clustered within a certain distance from each other, as seen in Figure 6.1. This is more calculation intensive than averaging, but the likelihood of the shown update being the correct one increases. This makes for a slower update, but the updates themselves will be more likely to be true. This requires stationarity for a much longer period of time before the true position is found, and if the device is moved around too much, even this can be upset. However, since positioning values have a tendency to lag, there is no guarantee that the cluster is in the proper location.

6.3 Averaging over RSSI values

When movement is not detected, instead of calculating the current position, the average between the current and the previous RSSI measurement values are used. Average signals are then used to calculate the position. If the mobile phone continues to be stationary, the averaging is calculated over all values that has been determined to be collected while stationary. This leads to a much more robust estimate; since all values are averaged before they are processed, errors are less likely to accumulate in the positioning estimation. This still gives us problems however, since there no longer is an easy way of determining if there is a drift in the positioning, and we no longer have access to single positioning data points.

When measuring RSSI data, there is two major sources of errors; noise from the signals themselves, and the fact that not every signal is transmitting at every sampling interval. To avoid picking up too much noise, there is also a limit on how low a signal can be before being counted as noise and disregarded entirely. This means that a choice must be made on how to average the signals, either by averaging over the total amounts of samples, or by the number of samples for each signal ID.

6.3.1 Averaging over all WLAN ID's

To average over all WLAN ID's we simply divide the sum the station's signal intensity by the total number of samples for all ID's. This sets all missing measurements to zero, which will discredit them from influencing the final position of the averaging function. If this choice is used the measurements will become slightly distorted over time, and even though the ratio should be the same, the signal intensity will become lower than it was originally. This may not be a problem, but it is an added source of error to a process that is already rather uncertain. It also means that if an ID is suddenly "lost" (for example, due to the access point being turned off), it's averaged value would slowly tend towards zero.

6.3.2 Averaging over each wifi ID individually

If each sample is averaged only over over it's own index, the averaging will more exactly represent the measured values. However, since the signal contains so much noise, it is more likely that a very seldom occurring signal is simply noise and not a real signal. If this is counted as a credible signal it will affect the final stable position of the algorithm and result in a small error.

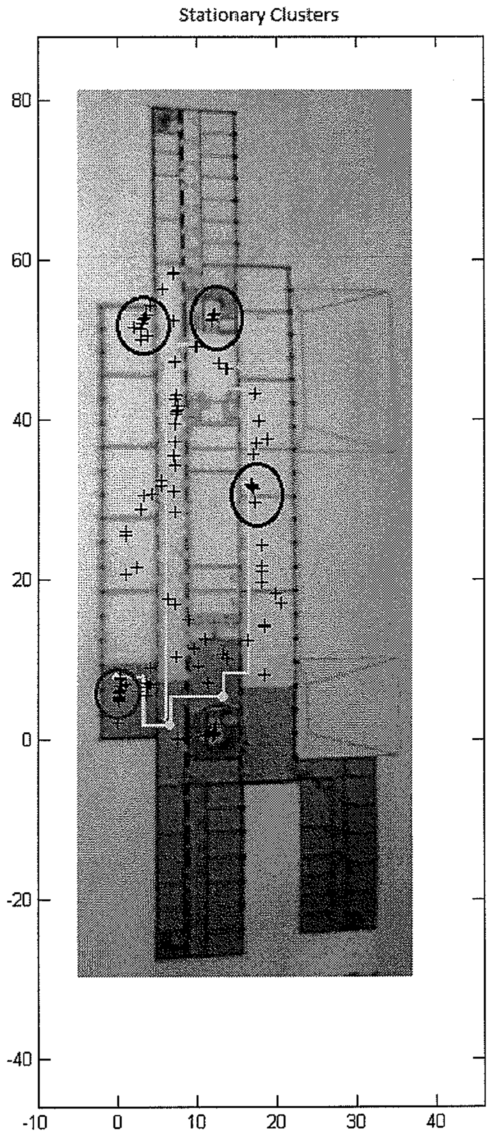


Figure 6.1. Position clustering when stationary

Chapter 7

IMPLEMENTATION ON A MOBILE DEVICE

7.1 Android positioning app

The averaging algorithm can be implemented by adding to an existing Android positioning application that uses RSSI values as input and returns positioning estimates as output. For this thesis, the algorithm was written in Java using Android development tool-kits and the Eclipse software editor. In this final implementation, we did not to use a Kalman filter as it is not required when we are averaging stationary points, and even though we could have implemented it while moving, we did not feel it worked well enough to merit implementing without further improving the filter first.

The implementation step itself was fairly complicated as we had to essentially add a new system to a pre-existing one, especially since the previous system did not always have all the components we required to complete our movement detection algorithm. We did however solve all these problems, and our final outline of the program can be viewed bellow.

When the original positioning application calculates a new position, the averaging algorithm does the following:

1. Use the accelerometer for detecting movement by comparing the Euclidean norm of its axes to the predefined cutoff value.

2. Get the RSSI values of each station.

3. Average the RSSI values using the "Averaging over all wifi IDs" approach described in chapter 6, if the mobile phone is found to be stationary. If the phone is not found to be stationary, simply use the original RSSI values.

4. Use these RSSI values as input to the positioning algorithm.

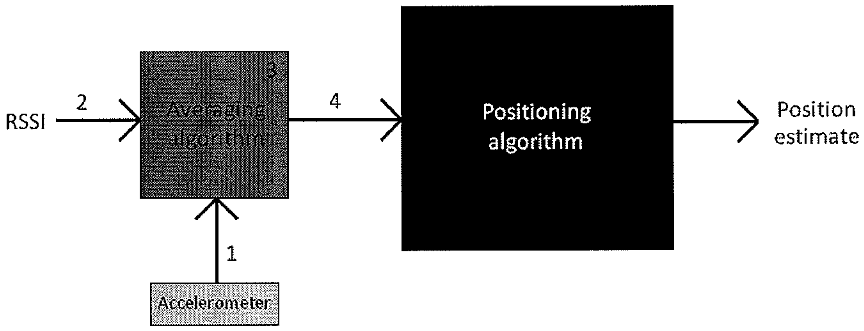


Figure 7.1. Implementation of the averaging algorithm

Chapter 8

RESULTS

8.1 RSSI averaging

This is the difference between the averaged and actual values of some of the measured APs. The averaging is done over every ID individually, one of the options that was discussed in chapter 5. The graphs will look very similar to values averaged over the total numbers of samples, but the amplitude will be lower for those values with fewer measurements.

As we can clearly see in Figure 8.1 the averaging of these signals work very well for stabilizing the individual ID's. Some of the signals move over time, but most of them stay relatively stationary. We can also observe that some of the ID's has remarkably samples than others, such as the top left graph of Figure 8.1 (the signal ID:s with less than five samples were not plotted here, as they were considered only noise). This is probably because the wifi senders are further away, or blocked by building walls or by some other mean weakened. This means that the only way we will get a strong enough reading from this ID is if it is transmitting abnormally strong, or if it is amplified by noise. This gives us a correlation between the number of measurements of a frequency and it's dependability.

8.2 Choice of Averaging function

The positioning data from the two different choices for averaging RSSI values (averaged over own samples or averaged over whole sampling length) can be seen in Figure 8.2 and Figure 8.3, respectively.

As we discussed in last section, there is a correlation between how dependable a frequency is and the number of samples counted. Therefore it is theoretically better to average each ID's samples over the maximum number of samples. This choice proves to be better, as the first figure shows. The inclusion of values with fewer samples gives us a small but uncorrectable error, as even a single erroneous value from an ID not present at the location is not corrected as long as we stand still. Ideally we would find a way to make both methods coexist to some extent, as it is mostly the most uncommon ID's that causes the majority of the error, and that the ID's with half the maximum number of samples are still a valid source

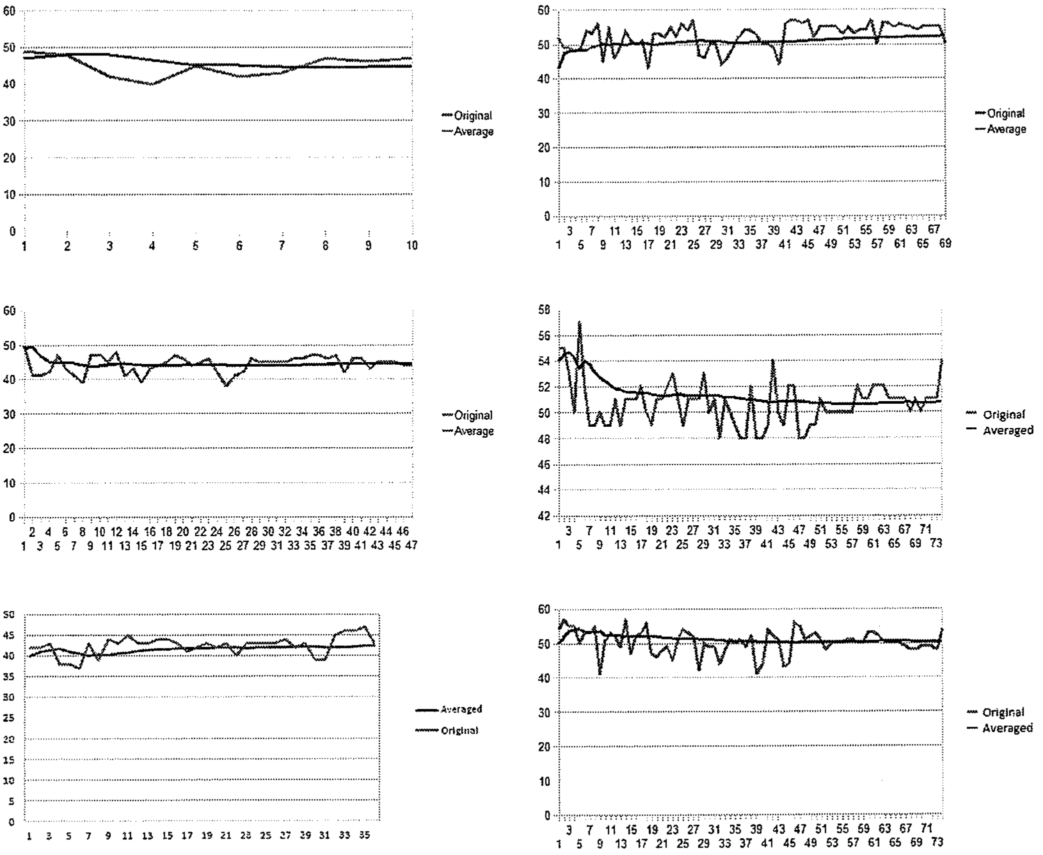


Figure 8.1. RSSI data and their averages from some of the access points. Note that the number of received RSSI values are different for each access point.

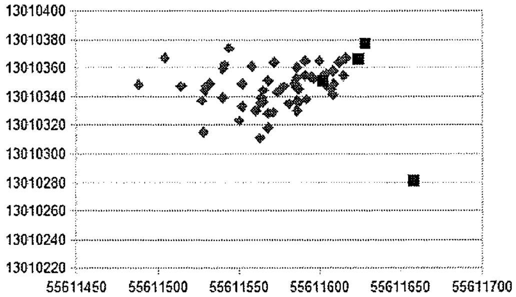


Figure 8.2. Positioning generated from RSSI values averaged over only the samples of its own ID.

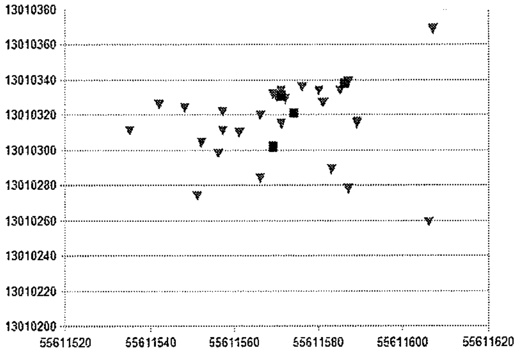


Figure 8.3. Positioning generated from RSSI values averaged over all samples.

of information that now gets mostly ignored. We still choose to use this rather simple method because it's robustness. Our tests were performed in a rather small environment with only 18 ID's present, and we can clearly see that low frequency ID's give us an erroneous contribution when averaging every ID over only it's own values. In a environment with many more WLAN access points this rather small source of errors could potentially become much larger, as more ID's might be given too much weight in the positioning.

8.3 Rate of Convergence

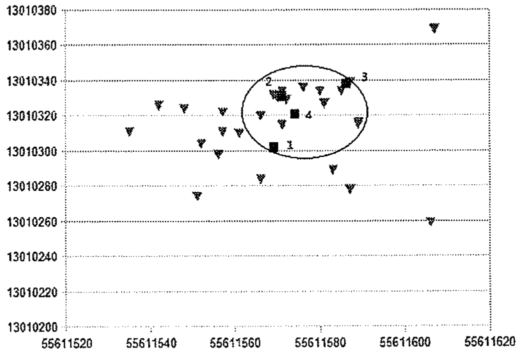
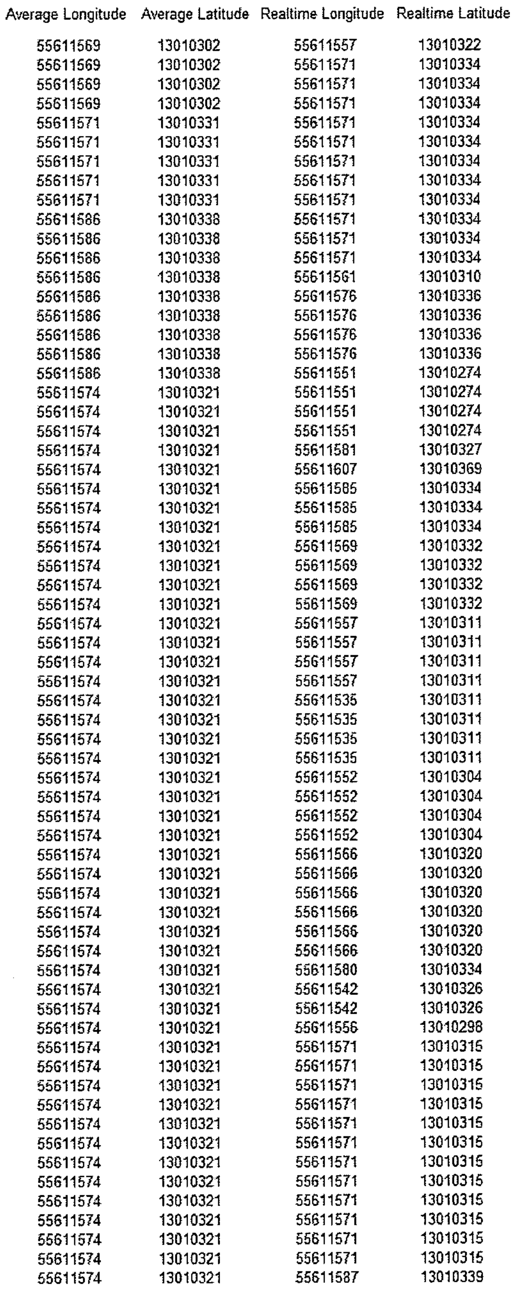


Figure 8.4. Reduced estimation error when using averaged RSSIs.

As we can see in Figure 8.5 the average function becomes stationary after 19 steps. Since the RSSI values are so varying it is prudent to assume that convergence will take some time. We have observed convergence in as little as three steps, but as shown in the table below, sometimes more time is required. The estimate is however still very good even before the final stationary sets in, as we can observe in Figure 8.4. This convergence is also dependant on the positioning algorithm used to interpret the RSSI values, and at least some of the delay in this case comes from this. This is also one of the reason for there being so few points in the initial averaging step.

8.4 Spread of Position

All of the averaged points in Figure 8.4 are within a 5 meter radius. The final stationary position is quite clearly within the center of this radius, and also at the center of all the non-averaged points. One thing we cannot show with graphs is



Average Longitude	Average Latitude	Realtime Longitude	Realtime Latitude
55611569	13010302	55611557	13010322
55611569	13010302	55611571	13010334
55611569	13010302	55611571	13010334
55611569	13010302	55611571	13010334
55611571	13010331	55611571	13010334
55611571	13010331	55611571	13010334
55611571	13010331	55611571	13010334
55611571	13010331	55611571	13010334
55611571	13010331	55611571	13010334
55611586	13010338	55611571	13010334
55611586	13010338	55611571	13010334
55611586	13010338	55611571	13010334
55611586	13010338	55611561	13010310
55611586	13010338	55611576	13010336
55611586	13010338	55611576	13010336
55611586	13010338	55611576	13010336
55611586	13010338	55611576	13010336
55611586	13010338	55611551	13010274
55611574	13010321	55611551	13010274
55611574	13010321	55611551	13010274
55611574	13010321	55611551	13010274
55611574	13010321	55611581	13010327
55611574	13010321	55611607	13010369
55611574	13010321	55611585	13010334
55611574	13010321	55611585	13010334
55611574	13010321	55611585	13010334
55611574	13010321	55611569	13010332
55611574	13010321	55611569	13010332
55611574	13010321	55611569	13010332
55611574	13010321	55611569	13010332
55611574	13010321	55611557	13010311
55611574	13010321	55611557	13010311
55611574	13010321	55611557	13010311
55611574	13010321	55611557	13010311
55611574	13010321	55611535	13010311
55611574	13010321	55611535	13010311
55611574	13010321	55611535	13010311
55611574	13010321	55611535	13010311
55611574	13010321	55611552	13010304
55611574	13010321	55611552	13010304
55611574	13010321	55611552	13010304
55611574	13010321	55611552	13010304
55611574	13010321	55611566	13010320
55611574	13010321	55611566	13010320
55611574	13010321	55611566	13010320
55611574	13010321	55611566	13010320
55611574	13010321	55611566	13010320
55611574	13010321	55611566	13010320
55611574	13010321	55611580	13010334
55611574	13010321	55611542	13010326
55611574	13010321	55611542	13010326
55611574	13010321	55611556	13010298
55611574	13010321	55611571	13010315
55611574	13010321	55611571	13010315
55611574	13010321	55611571	13010315
55611574	13010321	55611571	13010315
55611574	13010321	55611571	13010315
55611574	13010321	55611571	13010315
55611574	13010321	55611571	13010315
55611574	13010321	55611571	13010315
55611574	13010321	55611571	13010315
55611574	13010321	55611571	13010315
55611574	13010321	55611571	13010315
55611574	13010321	55611571	13010315
55611574	13010321	55611587	13010339

Figure 8.5. The positions estimated with averaged RSSIs compared to the original.

that the actual position of the measurement is that of the final stationary position, but as far as we could observe this was the case.

Chapter 9

CONCLUSIONS

Now matter where you go, there you are.
- attributed to Confucius

This thesis concludes that it is possible to detect movement using the accelerometer in a mobile phone.

It has been found impossible to use accelerometer data to directly deduce changes in position, due to lack of knowledge about how the mobile phone is held. However, movement can be detected by taking the Euclidean norm of the accelerometer output, subtracting gravity, and comparing it to a threshold value. If this is done correctly the output will be normal distributed.

It is very important to effectively subtract the correct gravitational norm from the accelerometer readings since doing so incorrectly will skewer the normal distribution on the output, which in turn will lead to imbalance in the movement threshold as the absolute will not be centred around zero. This gravitational norm is slightly lower than the expected value of 9.82 m/s^2, which has been assumed to be caused by inaccurate sensors. To compensate, we chose to build an averaging function that updates the gravitational norm by averaging over all accelerometer readings where movement is not detected.

While our choice of movement threshold works very well, we have basically only tested it ourselves. We deliberately chose a low threshold for movement. A lighter person might have a significantly lower acceleration while moving, which would lead to the system not detecting any movement, and therefore assuming that the person is permanently stationary. This lower threshold makes the estimates slightly less useful since they tend to reset if you move around too much, but this is preferable to having a systems that sometimes misses slow movement. Our final choice of threshold was 1 m/s^2, but this should be evaluated depending on the purpose of the application.

The norm of the three accelerometer readings of a person walking has a frequency of roughly 1 Hz. It is preferable to have a sampling frequency of at least double this, and use the built in Normal (4 Hz) sampling rate as our choice. Implementation-wise, there are some advantages of using the built in sampling rates, as these are handled by the operating system rather than the program and therefore makes it easily translatable across different hardware from different manufacturers.

This thesis also concludes that stationary accuracy can be significantly improved by averaging the measured RSSI values over time.

All WLAN data contain a lot of noise, making a position estimate calculated from a single set of RSSI data rather unreliable. Even when averaging a very large number of position estimates, the average does not remain stationary. Instead, the RSSI of each AP is averaged over the total amount of samples. If a sample is missing from a particular reading, it is counted as zero for that measurement. This means that some data is discarded, but the advantages are paramount as the system becomes much more robust and the impact of measurement errors and noise on the averaged position is minimized.

The intent at the start of this thesis work was to use a Kalman filter for improving the accuracy of position estimates. The purpose was to improve accuracy by reducing noise, which becomes redundant when averaging RSSI values while stationary. The Kalman filter is still viable when not stationary, and even though our Kalman filter is quite primitive, it does still reduce noise while the device is not stationary. It does however slow down the new positioning since it averages over new and the current position, i.e. the new position will be two thirds between the old value and the new input value.

There is currently no system in place to detect if the mobile phone is in fact moving without the accelerometer detecting this. However, since the averaging function is still active, the position will move towards a new position, albeit slowly. A backup system could be created that would set the movement algorithm to a moving state if the current averaged position has moved too far from the first.

Chapter 10

SUGGESTIONS FOR FURTHER RESEARCH

Though we have explored the possibility of detecting stationarity based on previous positioning values and accelerometer readings and found that our method works both well and effectively, we still have not produced any way of improving accuracy while moving.

There are however many ways one could improve this and perhaps even find some suitable way of predicting movement. We have worked a little with some of these ideas, while others are completely theoretical. Each idea has it's own merits and flaws and are discussed in more detail in the following sections.

10.1 Using compass or gyroscopes

By using compasses or gyroscopes it's possible to overcome the basic problem of detecting the orientation of a movement. If used in conjunction with accelerometer detection of movement it is possible to detect which way a person is facing while moving and detect the entire movement acceleration. This could in turn enable us to predict the next position is more likely to be. To capture the entire movement with the accelerometer would however require a very high resolution sampling, of at least 30 Hz.

The downside is that all active sensors use batteries, something that currently is a limited commodity in modern phones. Compared to accelerometers, gyroscopes will drain the batteries faster (eight times as much in our case, 0.1 mA vs 0.8 mA). Our algorithm uses a 4 Hz accelerometer sampling as opposed to 30 Hz, and added to that would be the extra use of gyroscope or compass, which would most likely lead to a battery consumption that would not be acceptable for the purpose of a simple positioning algorithm.

Another issue is the fact that many smart telephones lack sensors of this kind, especially older ones. This venue is mostly hindered by the limitations of current technology on the market; inertial navigation is already possible, but requires very expensive and bulky sensors. It should however be very accurate; by acquiring all possible data and using them, there is no doubt that a good positioning algorithm could be found. It could be an interesting topic of research at a future date when battery life is somewhat less of a concern, and more advanced sensors are readily available.

10.2 Kalman filter

As discussed earlier, though the Kalman filter described above functions, the lack of a true reading on the positions we measured rendered a filtering impossible as we could not filter out the noise in the movement data. This left us without the base for a proper state space model to base a Kalman filter on. If this was done, there would no doubt be improvement in the positioning, and the model might even be so good that you could use it to predict future movement without a new RSSI input. The filter could also be used in conjuncture with other sensors like compass or gyroscope, and could thus become even better at prediction movement. The drawbacks would be increased power consumption, as discussed above.

10.3 Kalman filter on the RSSI readings

It is possible that a Kalman filter working to improve RSSI readings instead of the actual positioning would be more useful, and we have two ideas how to do this.

The first approach involves the assumption that the RSSI signals are not independent processes, but can be modelled with a single Kalman filter. The model would become rather complicated, but this might be circumvented as there is already a modelling algorithm that maps the RSSI values into positions. By using this model in a Kalman filter it would be possible to both improve the current positioning and possibly prediction future positions based on the previous movements. An in-depth study of the noises in the RSSI readings would also yield greater understanding of the problems with positioning as a whole.

BIBLIOGRAPHY

[1] *Wireless positioning using ellipsoidal constraints* by G. Soldi, Master's thesis, Lund University, 2011.

[2] *Time Series Analysis* by H. Madsen, Chapman and Hall / CRC, p. 286-296, 2007. ISBN: 9781420059670

[3] *University Physics* by H. D. Young *et al.*, Addison-Wesley, p. 1219, 1996. ISBN: 0201847698

[4] *Inertial Navigation Forty Years of Evolution* by A.D. King, GEC Review (General Electric Company plc), 13:3, p. 140-149, 1998.

[5] *Sensor Data Fusion for Pedestrian Navigation Using WLAN and INS* by J. Seitz *et al.*, Symposium Gyro Technology 2007, Karlsruhe, Germany, 2007.

[6] *Advance Integration of WiFi and Inertial Navigation Systems for indoor Mobile Positioning* by F. Evennou and F. Marx, EURASIP Journal on Applied Signal Processing, Volume 2006, Article ID 86706, p. 111, 2006.

[7] *A dynamic system approach for radio location fingerprinting in wireless local area networks* by Shih-Hau Fang and Tsung Nan-Lin, IEEE Transactions on Communications, 58:4, p. 1020-1025, 2010.

[8] *Intelligent Dynamic Radio Tracking in Indoor Wireless Local Area Networks* by A. Kushki *et al.*, IEEE Transactions on Mobile Computing, 9:3, p. 405-419, 2010.

[9] *Kernel-Based Positioning in Wireless Local Area Networks* by A. Kushki *et al.*, IEEE Transactions on Mobile Computing, 6:6, p. 689-705, 2007.

[10] *Strapdown Inertial Navigation Technology* by D. Titterton and J. Weston, American Institute of Aeronautics and Astronautics, p. 153-155, 2004. ISBN: 9781563476938

[11] *Optimization of accelerometers for measuring walking* by R Foster, L Lanningham-Foster and J Levine, Proceedings of the Institution of Mechanical Engineers, Vol. 222, Part P: Journal of Sports Engineering and Technology, p. 53-60, 2008.

[12] *A New Approach to Linear Filtering and Prediction Problems* by R. E. Kalman, Transactions of the ASME Journal of Basic Engineering, Vol. 82 (Series D), p. 35-45, 1960.

[13] *www.Skyhookwireless.com*

[14] *www.Ekahau.com*

[15] *www.Qubulus.com*

[16] *www.Combain.com*

Master's Theses in Mathematical Sciences 2012:E1
ISSN 1404-6342

LUTFMS-3186-2012

Mathematical Statistics
Centre for Mathematical Sciences
Lund University
Box 118, SE-221 00 Lund, Sweden

http://www.maths.lth.se/