Inferring Motion and Location Using WLAN RSSI

Kavitha Muthukrishnan, Berend Jan van der Zwaag, and Paul Havinga

Pervasive Systems Group, University of Twente, Enschede, The Netherlands
{k.muthukrishnan,b.j.vanderzwaag}@utwente.nl
http://ps.ewi.utwente.nl/

Abstract. We present novel algorithms to infer movement by making use of inherent fluctuations in the received signal strengths from existing WLAN infrastructure. We evaluate the performance of the presented algorithms based on classification metrics such as recall and precision using annotated traces obtained over twelve hours effectively from different types of environment and with different access point densities. We show how common deterministic localisation algorithms such as centroid and weighted centroid can improve when a motion model is included. To our knowledge, motion models are normally used only in probabilistic algorithms and such simple deterministic algorithms have not used a motion model in a principled manner. We evaluate the performance of these algorithms also against traces of RSSI data, with and without adding inferred mobility information.

Key words: Motion inference, Localisation, WLAN, RSSI

1 Introduction

Ubiquitous computing is emerging as an exciting new paradigm with a goal to provide services anytime anywhere. Context is a critical parameter of ubiquitous computing. Ubiquitous computing applications make use of several technologies to infer different types of user context. The context cue that we are interested in is users' *motion* being either *"moving"* or *"still"* and *location*. The desire of using WLAN infrastructure particularly to derive context is very strong, both from the perspective of the availability of the clients device and that of the infrastructure – nearly all smart phones, PDAs, laptops and many other personal electronic devices have a built-in wireless interface.

Looking at the applicability and usefulness of motion detection, WLAN radio by itself can sense motion, and it can potentially be also part of the sensor ensemble to improve recognition performance. Apart from inferencing activity of the user itself, it has been showcased that such motion inference is useful for efficient radio fingerprinting solutions [2]. Recently movement detection was shown to adaptively switch between passive sniffing and active scanning to allow positioning and to minimise the impact on communications [3]. In this paper, we show yet another use of how it can improve localisation accuracy. The applications that are described above do not necessarily benefit from accurate and complete information about the mobility status. For the purposes described above it is sufficient to know whether the user is moving or not.

This paper examines the results of several motion and location sensing algorithms that operate on RSSI data gathered from existing WLAN infrastructure. The main advantages of the proposed algorithms are: *(i)* deducing user's context without a need for additional hardware, and *(ii)* preserving user privacy, as context inference can be performed locally at the client device.

Contributions: The key contributions of this paper are as follows:

- A detailed characterisation of WLAN RSSI data, exposing a rich set of features both in time and frequency domain to gather mobility information. Our analysis in both temporal and spectral domain, results in a conclusion that "when a device is moving, signal strengths of all heard access points vary much greater compared to when a device is still and the number of detectable samples from access points vary considerably when the device is moving".
- We present novel algorithms to infer movement that makes use of inherent fluctuations in the signal strength. We evaluate the performance of the presented algorithms thoroughly based on classification metrics such as recall and precision from annotated traces (typically groundtruth recorded for every second) obtained over twelve hours effectively from different types of environment and with different access point densities.
- We show how a common deterministic location algorithm such as centroid and its variant can improve its accuracy when a motion model is included. To our knowledge, a motion model is normally used only in probabilistic algorithms and such simple deterministic algorithms have not used a motion model in a principled manner. We evaluate the performance of algorithms against traces of RSSI data collected from different environments, with and without adding mobility information inferred from the mobility detection algorithm.

2 Related Work on Motion Sensing

Randell et al. [9] demonstrated the possibility of distinguishing various states of movement such as walking, climbing and running using a 2D accelerometer. Patterson et al. [6] take the velocity readings from GPS measurements and infer the transportation mode of the user, for instance walking, driving, or taking a bus using a learning model. The model learns the traveller's current mode of transportation as well as his most likely route, in an unsupervised manner. It is implemented using particle filters and is learned using Expectation-Maximisation. The learned model can predict mode transitions, such as boarding a bus at one location and disembarking at another.

Krumm et al. [4] classified a user as either moving or still based on the variance of a temporally short history of signal strength from currently the strongest access point. This classification had many transitions, hence it was smoothened over time with a two-state hidden Markov model resulting in an overall accuracy of 87%.

Anderson et al. [1] use GSM cellular signal strength levels and neighbouring cell information to distinguish movement status. The classification of the signal patterns is performed using a neural network model resulting in an average classification accuracy

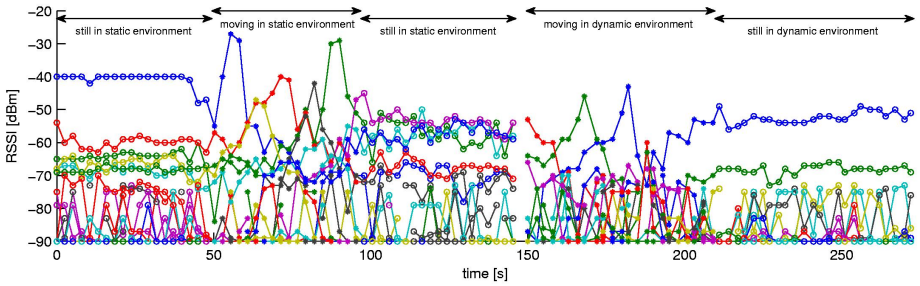


Fig. 1. This figure illustrates two minutes of "still-moving-still" as measured in a static environment and two minutes of "moving-still" measured in a dynamic environment.

of 80%. The authors trained the neural network initially and demonstrated a proof of concept by implementing it at run time on a cell phone. However, the initial training did not work in all the environments as signal strength fluctuations were different in different environments. Sohn et al. [10] published a similar technique for detecting the users' motion using signal traces from GSM network. Their motion detection system yields an overall accuracy of 85%. They extracted a set of 7 features to classify the user state as either still, walking, or driving.

Our work on motion detection algorithm is similar to the work on Anderson et al. [1] and Sohn et al. [10], but we look into variation in the WLAN RSSI observed across several access points as opposed to GSM signals.

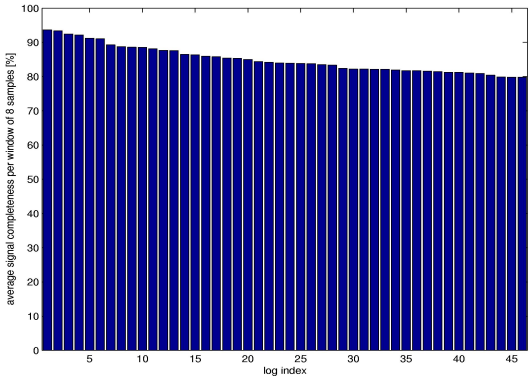
3 Temporal and Spectral Characterisation of Received Signal Strength (RSSI)

In this section we investigate some of the properties showcased by RSSI, particularly how it changes over time when the user is still and moving, both in static and dynamic environments. By static environment, we mean when the device is placed in a relatively quiet environment (e.g., by logging measurements at off-peak hours) and dynamic environment refers to an area affected by people moving about (e.g., canteen during lunch hours).

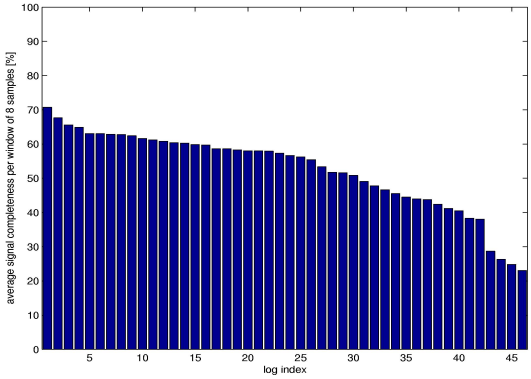
Figure 1 shows an example of temporal characteristics – each of the lines represent signal strength received from a specific access point. We can observe from the figure, only some of the access points show a clear distinction between the "still" and "moving" periods – specifically the weaker signals (RSSI < -75 dBm) do not convey any significant difference for both still and moving, hence we have used only the stronger access points for the analysis presented below.

As for whether the variation in the signal strength is influenced more by the changing environment around a static device or by movement of the device itself, Fig. 1 clearly shows that the signal variation is much more prevalent due to the device movement rather than to the dynamics of its environment.

Apart from signal strength fluctuations, we do observe a lot of variations in the number of samples received within a particular observational window (e.g., window



(a) Still



(b) Moving

Fig. 2. Variations in the number of samples received when (a) Still and (b) Moving. Each log (with an average duration of 7–8 minutes) is split into windows of 8 samples and the results are averaged together.

of 8 samples) as shown in Figure 2. It is particularly interesting to note that, at a fixed location the number of signal strength samples received from the same access point over a window of reading fairly remains closer to 85% on an average. This is reasonable, as in one scan we typically do not hear all the access points, so one or two missing signal values is still relatively acceptable when the device is still. As opposed to this, in the case of moving, the number of signal strength samples received from the access point varies as the number of access points detectable at a place varies greatly as the user moves. Each of the bins in the Figure 2 represents the average result of the number of samples seen from all the detected access points over all windows of 8 samples from one distinct log. In total for still and moving, we collected over 90 different logs with average duration of log spanning for about 7–8 minutes.

Looking at the spectral characteristics (Figure 3(a)) reveals that as a rule of thumb, the more concentrated the time domain, the more spread out the frequency domain. In

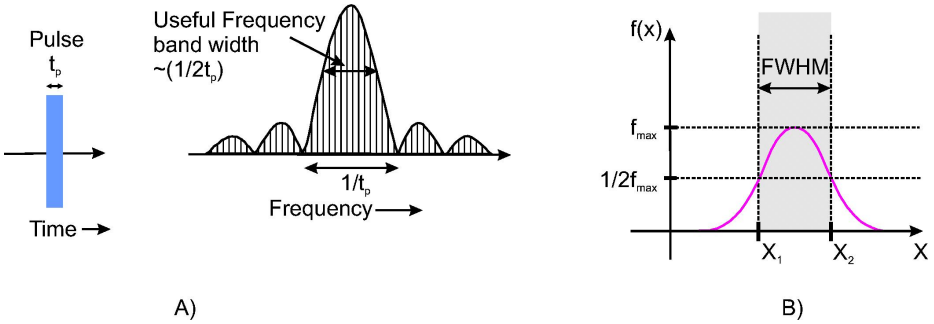
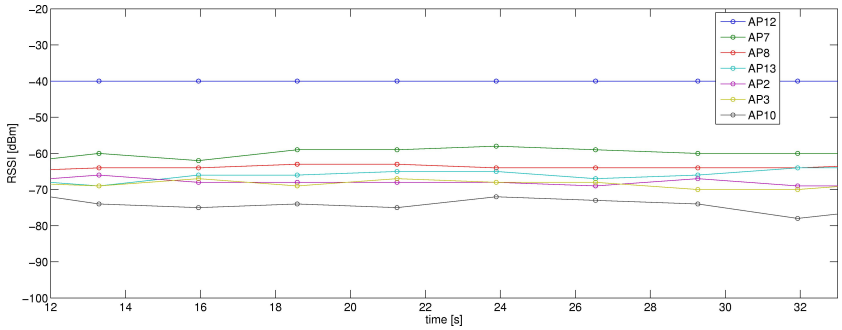
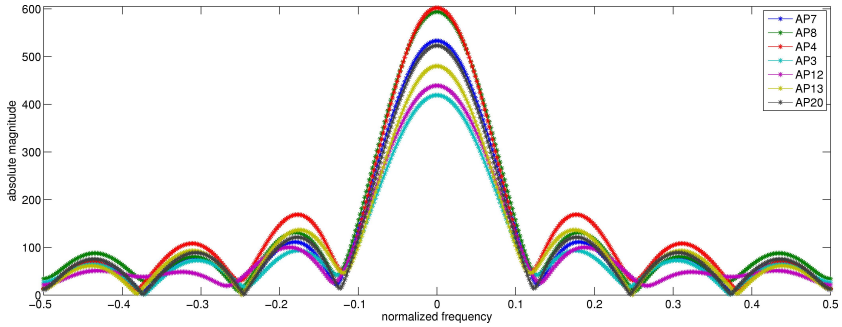


Fig. 3. *(a). Schematic representation of a rectangular pulse in time and frequency domain, short duration pulses produces a large bandwidth (b). Full Width Half Maximum, corresponding to peak width at 50% peak height. t_p is the pulse period, and f_{max} is the peak at maximum. X_1 and X_2 are used for calculating FWHM (explained later in Section 4.2).*

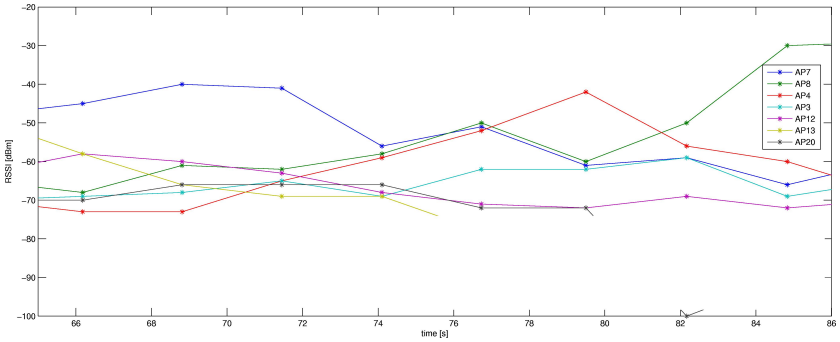


(a) Temporal variations of 8 samples over a window, when the device is still

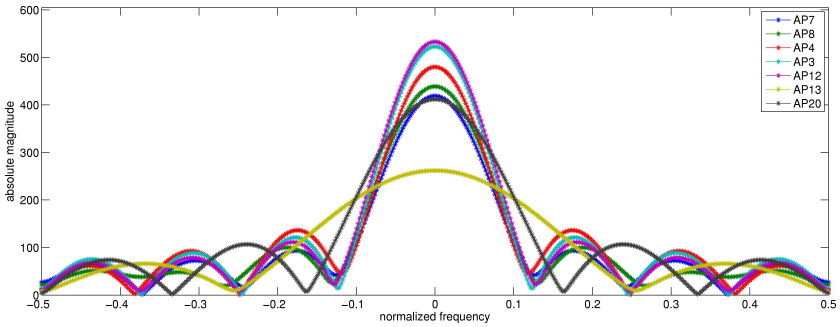


(b) Spectral variations of 8 samples over a window, with a 512-point FFT when the device is still

Fig. 4. *Temporal and Spectral characteristics of a window of 8 samples of the strongest 7 access points for the case of "still". (a) The signal taken is a subset of the signals that are represented in still phase in Figure 1 for time varying between 12–32 seconds, and corresponding FFTs in (b).*



(a) Temporal variations of 8 samples over a window, when the device is moving



(b) Spectral variations of 8 samples over a window, with a 512-point FFT when the device is moving

Fig. 5. *Temporal and Spectral characteristics of a window of 8 samples of the strongest 7 access points for the case of "moving". (a) The signal taken is a subset of the signals that are represented in moving phase in Figure 1 for time varying between 66–86 seconds, and their corresponding FFTs in (b).*

particular, if we "squeeze" a function in time, it spreads out in frequency and vice-versa. Also Figure 3(b) illustrates Full Width at Half Maximum (FWHM) that corresponds to peak width of the FFT signal at 50% peak height.

Figures 4(a) and 5(a) present temporal variations in the signal strength observed over a short window of 8 samples (approximately 20 seconds duration) from the strongest seven heard access points when the device is still and moving and the corresponding frequency domain representation is shown in Figures 4(b) and 5(b). It is evident that although signal strength varies even while the user is still, this variation is reflected in all the heard access points uniformly as there is a well defined peak with a narrow spectral width in the frequency domain from all the access points, despite the fact that there is difference in the Fourier amplitude from each of the heard access point. But when the user is moving, there is no well defined peak from all the access points in the frequency domain indicating that variation in the signal strength happens more often and not in all the heard access points in the same manner. Specifically, we observe the effect of spectral broadening from a significant number of access points when the user is moving,

resulting in a wider full width at half maximum. This phenomenon happens mainly due to two reasons: *(i)* the variation in the signal strength is large in case of a moving user and *(ii)* the number of access points detectable varies with distance resulting in too few received samples from the access points. This confirms that both the temporal and spectral analysis lead to the similar conclusions but give a different view of representation. More detailed background information can be found in our earlier work [7].

4 Algorithms for Sensing Motion

In this section we present algorithms for sensing motion. We base the algorithms on the observations presented in the previous section and categorise them into time domain and frequency domain. All presented algorithms are based on "thresholding" applied to a certain metric. We explain in Sec. 4.3 how these thresholds are obtained automatically.

4.1 Time Domain Algorithms

We evaluate four different metrics that we observe in the temporal domain to infer user movement. As opposed to looking at one RSSI value, all the algorithms presented here use RSSI observed over a window of readings (window size typically 8 samples).

AP Visibility. This is the simplest algorithm as it just uses the proportion of the time that RSSI of a particular access point is observed within the observation window. For classifying "moving" or "still" the observed proportion is calculated for each access point and then averaged together. Depending on a (learned) threshold the algorithm detects the state as either moving or still.

Spearman's Rank Correlation Coefficient. We estimate the correlation coefficient using Spearman's Rank Correlation Coefficient (ρ) [11]. The rank correlation coefficient between any two measurements represents how closely the signals are ranked. It takes values between -1 and 1. Values closer to 1 indicate that the measurements are similar and hence the user is still and when the user is moving the values are lower. The algorithm tracks Spearman's ρ between the first and the last measurement in an observation window as a metric to distinguish between moving and still states. Figure 6 presents how the rank correlation coefficient varies when the device is still and moving.

Standard Deviation. This algorithm uses mean standard deviation (SD) over all the heard access points as a metric to distinguish between still and moving states. Within the observation window we measure the SD between the measurements for each detected access point, and use the average SD over all heard access points for inferring the motion status. Figure 7 presents how the average SD varies when the device is still and moving.

Euclidean Distance. This algorithm determines the Euclidean distance between the first and last measurements within an observation window. It is based on the expectation that the average Euclidean distance between WLAN measurements is proportional

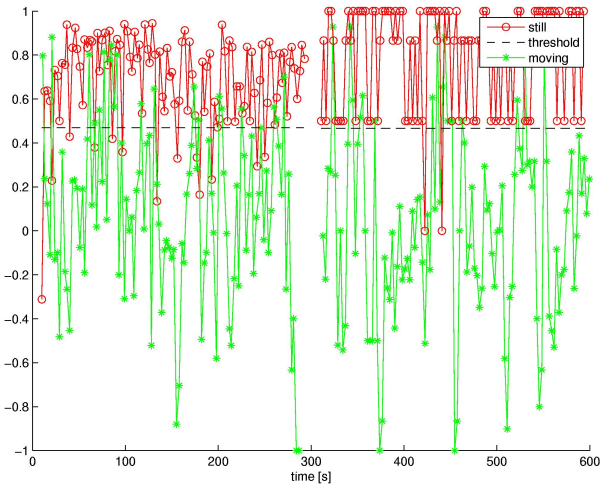


Fig. 6. *Spearman's rank correlation coefficient when still and moving, for outdoor (left) and indoor (right) environments. The difference in the rank correlation coefficient remains the same for both outdoor and indoor.*

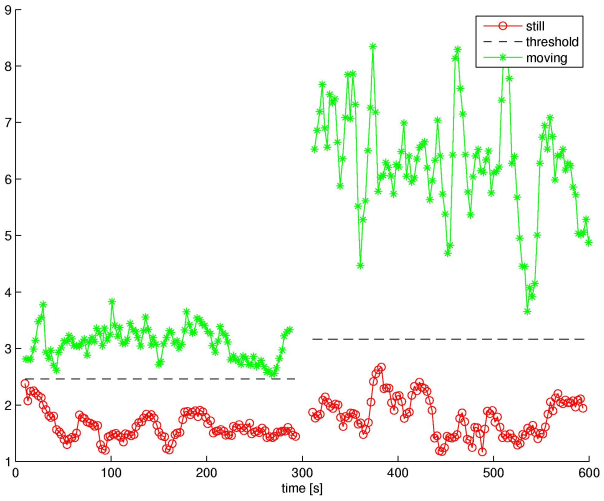


Fig. 7. *Mean standard deviation when still and moving, for outdoor (left) and indoor (right) environments. Here we can observe a considerable difference in the SD values between the measurements logged from an outdoor and indoor environment.*

to the state of the movement. Figure 8 illustrates the average Euclidean distance between WLAN measurements and shows that the average Euclidean distance between WLAN measurements are proportional to the state of the movement.

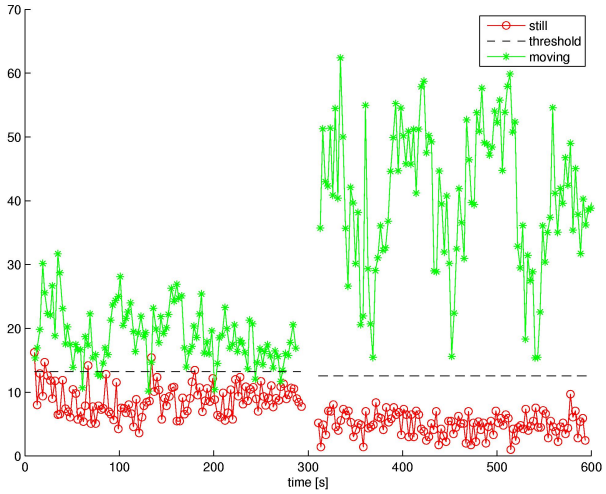


Fig. 8. *Euclidean distance when still and moving, for outdoor (left) and indoor (right) environments. Here we can observe a considerable difference in the ED values between the measurements logged from an outdoor and indoor environment.*

4.2 Frequency Domain Algorithms

We now present three novel motion detection algorithms which are inspired by our observations [7] in the frequency spectrum of the RSSI.

Full Width at Half Height (FWHM). The RSSI in time series is converted into the frequency domain using Fast Fourier Transformation (FFT). For calculating the FWHM we let the algorithm find the width of the main peak in the FFT signal at 50% of the maximum amplitude, for each of the observed access points. Typically, there is no value at exactly this amplitude, hence we linearly interpolate between the two points nearest to it on either side. For classification, this algorithm uses the FWHM of the main peak of the FFT for a given window of samples and takes the median over all the access points observed in that window.

FWHM Count. This algorithm is very similar to the previous algorithm. It essentially tracks how many access points have a spectral width (i.e., FWHM) that is exceeding a certain threshold within the window of readings. The FWHM is calculated as explained above. Whenever an entry (access point) exceeds the FWHM threshold, the algorithm treats this as an outlier and increments a counter. If the counter exceeds a certain threshold relative to the total number of heard access points within the observation window, the algorithm returns the user state as *moving*, otherwise it returns *still*.

Low-Amplitude-Frequency Count (LAFC). A signal that is not varying much in the time domain has a frequency spectrum with a narrow peak around 0 and very low amplitudes at higher frequencies. In contrast, a signal that significantly varies in the time

domain has a broader frequency spectrum, i.e., the peak around 0 is wider and amplitudes at higher frequencies are not as low as for less varying signals.

Based on this observation we use a novel metric, (*low-amplitude-frequency count* or *LAFC*) that distinguishes between "still" and "moving". The algorithm operates on the FFT signal and effectively counts the number of frequencies that have low amplitude (based on earlier experiments we define " low amplitude" as less than 10% of the maximum amplitude in the FFT to achieve the best results). If this number exceeds a certain threshold, the motion status is set as "moving", otherwise as "still". The LAFC is determined for each heard access point within the observation window and then averaged over all heard access points.

4.3 Threshold Learning

Each of the algorithms described above uses a certain threshold to decide whether the user's device is still or moving. As these thresholds are sensitive to several factors (e.g., environment, hardware, operating system), we use part of our data set for learning the respective thresholds and the remaining part of our data set for determining the classification accuracy using the learned threshold. We use five-fold cross validation, where a data set is partitioned into five folds, and five training and testing iterations are performed. On each iteration, four folds form a training set and one fold is used as a testing set. To illustrate our threshold scheme, let us consider finding the right threshold for the LAFC metric described in Sec. 4.2. The threshold is derived automatically from a training data set (containing more than 12,000 samples in total) using the following method.

1. For each observation window in the training set, the LAFC is calculated.
2. Then, the distributions for both classes ("still" and "moving") are determined. See Figure 9 for an example distribution histogram. If a threshold is applied anywhere on the LAFC axis, typically some of the "moving" observations will lie on the "still" side (in this case the right hand side) and some of the "still" observations will lie on the "moving" side (in this case the left hand side); these are the *false negatives* and *false positives*.
3. Our method now places the threshold at a position where the weighted sum of false positives and false negatives is minimal. Figure 10 shows the amount of false positives ("still" classified as "moving") and false negatives ("moving" classified as "still") as well as their weighted sum as a function of the threshold. For the LAFC metric, values below (i.e., to the left of) the threshold are classified as "moving" and values above (i.e., to the right of) the threshold as "still".

   We can see that for this particular part of the data set the best threshold is 1.14 yielding a total classification error of about 9% for the training set.
4. We then use this learned threshold to calculate the classification accuracy for the remaining part of the data set (i.e., for the test set). Classification accuracies for all algorithms are reported in the next section.

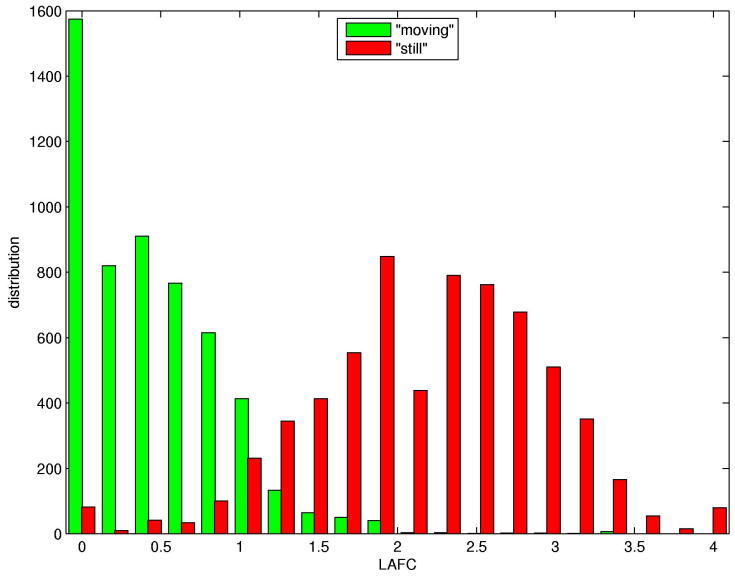


Fig. 9. *Distribution of low-amplitude-frequency count (LAFC) for "still" and "moving" classes for a typical training data set (containing more than 12,000 samples in total).*

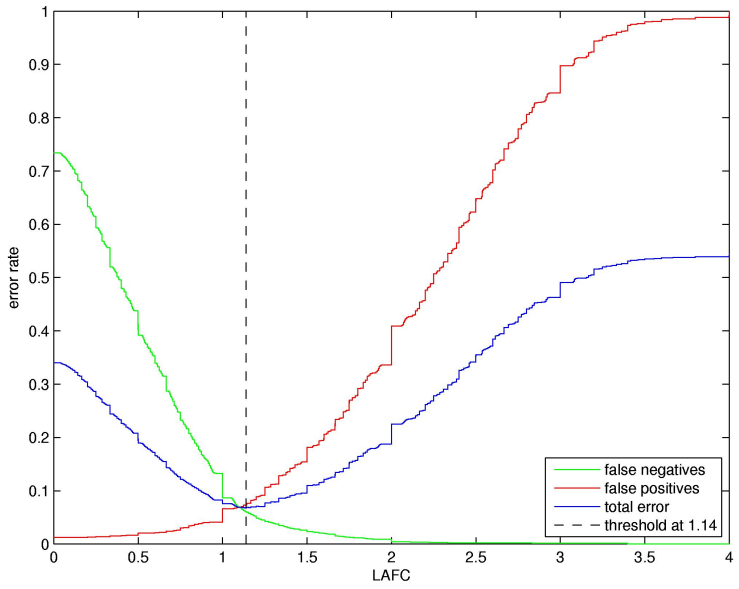
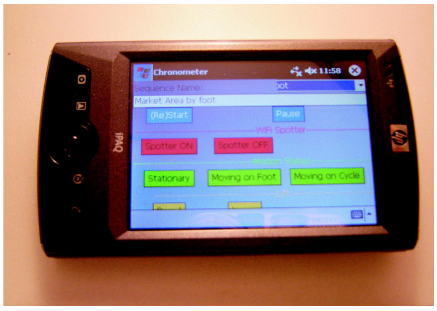


Fig. 10. *The amount of false positives ("still" classified as "moving") and false negatives ("moving" classified as "still") as well as their weighted sum as a function of the threshold for the LAFC metric.*



Measurment Fri May 25 12:14:44 CEST 2007

25-05-2007 12:14:45 149 Start Sequence:Waaier
25-05-2007 12:14:45 149 Motion:Moving
000136079de0,-90
000cf6164f6c,-90
00116b267fd8,-73
0001e3d43a8d,-53
0001e3da0a55,-90
00147f54a4ff,-74

Fig. 11. Snapshot of (*left*) custom diary application and (*right*) logged ground truth with measured RSSI readings.

5 Motion Inference Performance Evaluation

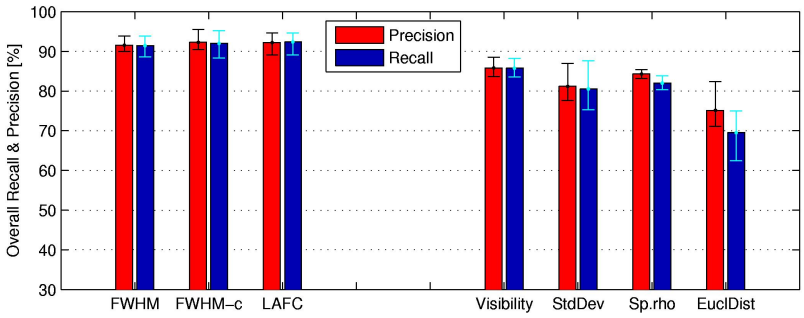
5.1 Data Collection.

Two members of our research team collected WLAN network traces, each data collector carried an HP IPAQ pocket PC running a spotter application for recording readings from nearby access points and logging them. Data collectors recorded their mobility activities using a custom diary application running on the PDA that allowed them to indicate whether they were walking, driving, cycling or staying still (refer Figure 11). Data collection was performed at common places such as city centre, parking lot, university campus and indoor at the office, canteen and home. In all, the spotter logs contained WLAN traces of about 12 hours duration with annotated ground truth. The unlabeled part of the logs were filtered out in order to measure the accuracies of the presented algorithms accurately. Approximately 50% of the logs collected correspond to stationary phase and the remaining 50% correspond to activities performed on the move. Sampling the radio environment at approximately 0.4 Hz, the 12 hours of logs correspond to roughly 16,000 samples. The logs also include different access point densities (the least number of access points in the data collected was 0; this happens when no access point is heard during a particular scan, in this case all our algorithms maintain the last inferred motion status until one or more access points are heard again).

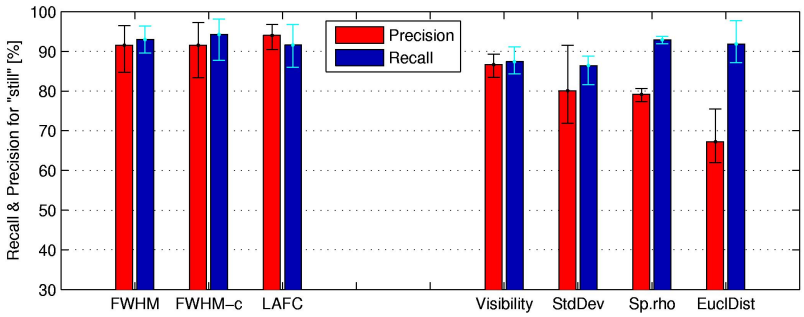
5.2 Results and Discussion.

We evaluate how accurately the presented *time domain* and *frequency domain* algorithms can differentiate between moving and still states. Figure 12 shows a one-to-one comparison of the results obtained from both time and frequency domain algorithms tested against the same data sets.

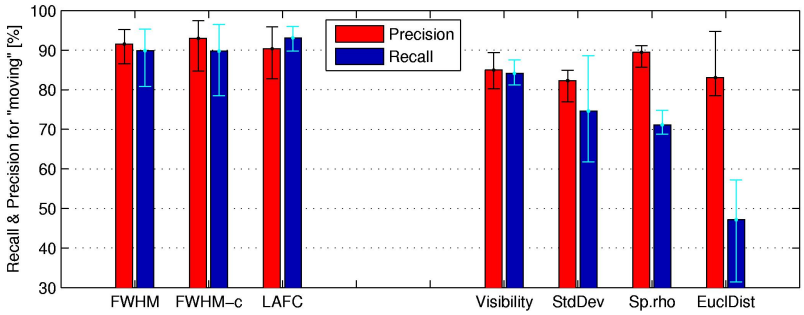
In order to thoroughly characterise the classification performance, we use the metrics *precision* and *recall*. Precision for a class is the number of true positives (i.e., the number of items correctly labelled as belonging to the class) divided by the total number of elements labelled as belonging to the class (i.e., the sum of true positives and false positives, which are items incorrectly labelled as belonging to the class). Recall



(a) Comparison of the overall precision and recall



(b) Comparison of the precision and recall for still



(c) Comparison of the precision and recall for moving

Fig. 12. *Performance of motion detection algorithms achieved by 3 frequency domain and 4 time domain algorithms: FWHM, FWHM-count, Low FFT, AP seen, Std dev, Rank, Euc. dist. over 12 hours of WLAN traces collected. The error bars indicate the variations in the accuracy depending on which training and test sets were used in each iteration.*

in this context is defined as the number of true positives divided by the total number of elements that actually belong to the class (i.e., the sum of true positives and false negatives, which are items which were not labelled as belonging to that class but should have been). Figure 12 shows the precision and recall of all the 7 algorithms that we

discussed in section 4, using five-fold cross validation for the selection of training and test data. The classification results are averaged together for getting the final result.

The error bars in Figure 12 indicate the variations in the accuracy depending on which training and test sets were used in each iteration. Further analysis (not reported here) shows that the sensitivity of a particular algorithm does not only depend on the variations in the learned thresholds observed among different folds, but also on the underlying data itself.

Looking at the results in general leads to the following conclusions – the performance of the frequency domain algorithms show a better precision and recall than the time domain algorithms. Nevertheless, it is interesting to note that simple count of the number of observed access points achieves an overall accuracy of 86%. This accuracy is similar to the one reported by Krumm et al. [4] using an algorithm based on the temporal variation of RSSI. Of course it is hard to make one-to-one comparison among algorithms when the data used for testing is different in both cases. But since we have performed the evaluation for data collected from different environments and settings, we do not expect any drastic difference in the performance, when testing on other data.

Comparing the different frequency domain algorithms, all the three algorithms – FWHM, FWHM-count and LAFC results are comparable (91%–92%). Another important aspect we observed is that generally the thresholds for the frequency domain metrics are not as sensitive to external influences as the ones for the time domain metrics and some of the time domain metrics are particularly more sensitive.

The overall classification accuracies obtained with most time domain algorithms (81%–86%) are comparable to the one reported by Sohn et al. [10] (85%). Although Sohn et al. achieved this accuracy for a three state classification scheme and our results are for two state classification, it is interesting to note that all our presented algorithms use a single feature as opposed to Sohn's work where 7 different features were used in combination to train and test data. We expect that combining features will result in better accuracy, at the cost of higher complexity. This is yet to be investigated.

It typically takes half an observation window for any metric to cross its threshold during transitions between states. This is of course not surprising, because halfway the window half of the samples will have a ground truth of "still" and the other half will be "moving". We can therefore interpret a classification at time t as the estimated motion status for time $t - \frac{1}{2}T_{\mathrm{w}}$, where T_{w} is the length of the observation window.

Our frequency domain algorithms perform very well for all experimental settings by achieving an overall classification accuracy of 92%, clearly outperforming all the other motion inference algorithms. Fine tuning the threshold learning and/or incorporating more features together might even further increase the accuracy. The results show that we are able to distinguish between still and moving states with a high accuracy without having to instrument a person with any additional sensors.

6 Localisation

In this section, we outline the localisation algorithms that operate on RSSI data. The goal of our work is to demonstrate that algorithms which rely on only the location of access points and a coarse estimate of the relative distance to the access points can benefit

from adding motion information that we presented above in a principled manner. Many of the probabilistic approaches like particle or Kalman filtering use an inherent motion model to enable localisation and tracking. These algorithms work in a "predictor-corrector" fashion, by weighing the filter model more heavily when the errors in the raw measurement increase, thereby making the final estimates quite accurate. We have used a similar approach but coupled to a simpler centroid algorithm and its variant. This work is an extension of our earlier work [8], which also includes more background and an overview of other related work on WLAN localisation.

Our localisation algorithms are implemented using *filter chains*, which represent a sequence of calculations performed on the RSSI measurements. We rely on the known locations of the access points and other information such as their MAC addresses and their transmit power settings. These access point data could either be managed as a database residing in the network, or it could be a local configuration file in order to minimise the network dependencies and also keeping in view of privacy. Obtaining this information is easier when an accurate database of network access points is already available. In literature, there exist many other ways such as war driving, stumbling to obtain the neighbouring access point coordinates [5].

We use an exponential moving-average filter to reduce the effect caused due to the noise and smoothen the received signal strength for the analysis presented here.

$$\text{RSSI}_{\text{current}} = \alpha \times \text{RSSI}_{\text{prev}} + (1 - \alpha) \times \text{RSSI}_{\text{current}} \tag{1}$$

Equation 1 states that the current RSSI value is a linear aggregate of the previous RSSI value and an independent weighting factor α ($0 \leq \alpha \leq 1$). The weighting for each older observation decreases exponentially, giving much more priority to recent observations while still not discarding the older observations entirely. We use $\alpha = 0.2$. Automatically determining the optimal α is part of our future work.

The core positioning algorithm is based on a weighted centroid approach. The difference between the normal centroid and weighted centroid is that it introduces variable weights for each access point. Weighted Centroid uses the distance estimates to the strongest access points in relation to each other. This is performed by assigning the location of each of the few strongest access point a weight in the position calculation based on the relative distance between those estimates. Obtaining absolute distances precisely between the access points and the mobile device to be located is harder due to the multi-path reflections that are predominant to indoor environments. Hence algorithms that make use of absolute distances retrieved from WLAN RSSI, such as based on trilateration perform poor, especially when the access points are arranged in a collinear fashion.

We estimate the *relative distances* based on the transmit power of the access points that are available and the RSSI values which typically correspond to the power at the receiving end. Although the distance estimates are not accurate, it will give a cue on which access points are relatively closer and hence to be used in the position estimation. From the RSSI we use motion inference as explained earlier to detect the state of the device as either still or moving. Depending on the state, we use two filter chains:

1. When the motion detection algorithm returns the state of the user as moving, we employ a motion model which smoothens the final location estimates, by preventing

any large movements between two different time steps. The motion model filter uses a maximum allowable distance, depending on the users walking speed (say 1.4 m/s) within a stipulated time frame. If the estimated travelled distance exceeds the limit, the location is updated solely based on the motion model.
2. When the user is still, ideally the user's estimated position must remain still at the same point. But since the signal strength varies even at static location, the estimated location often jumps even when the device is still. We therefore use a smaller value for the maximum allowable distance (say 0.2 m) for the static cases and use a history of measurements to average the results together.

7 Experimental Evaluation

We compare the accuracy of the presented Weighted Centroid with the motion model described above to that of other simple position estimation algorithms with and without adding motion information. In total we have four algorithms to compare: (1) Weighted Centroid with motion model (2) Centroid with motion model (3) Weighted centroid without motion model and (4) Centroid without motion model.

The Centroid algorithm places the user on the geometric centroid of the strongest access points that appeared on the current scan. Weighted centroid as we explained before assigns specific weights to the access points based on the estimated relative distances. Centroid with motion model essentially uses the same principle, but with movement limits depending on the inferred motion status. The metrics we use for the evaluation are the median accuracy, which indicates the accuracy reported by 50% of the readings and the mean horizontal and vertical errors.

7.1 Data Collection.

The experiments to assess localisation accuracy were performed in a five-storied university building. Floors 2–5 have a dimension of 106 m $\times$ 14.5 m and have a similar layout with a long corridor and many rooms and have four access points per floor that are mounted on the ceiling and are placed in a straight line. The ground floor (refer Fig. 14) has a different layout with a few additional access points covering the northern extension of the building and no access points covering the southern extensions. The transmit powers of the access points are either 50 mW or 30 mW. The measurements were taken from walking along all floors (some twice) from one end to the other, including the stair cases at both ends. Data was recorded as one trace lasting approximately twenty minutes, resulting in about 350 RSSI readings at 0.4 Hz. Since the same data was to be tested with different algorithms, we logged the measurements and all the analyses were done offline.

Ground Truth.
Normal map clicking applications are error-prone when used on a small-screen device such as a PDA. Hence, we followed a slightly different approach for registering the ground truth locations. Measurements were logged using the same diary application as used for recording the motion status. At crucial points in the path, such as corners and

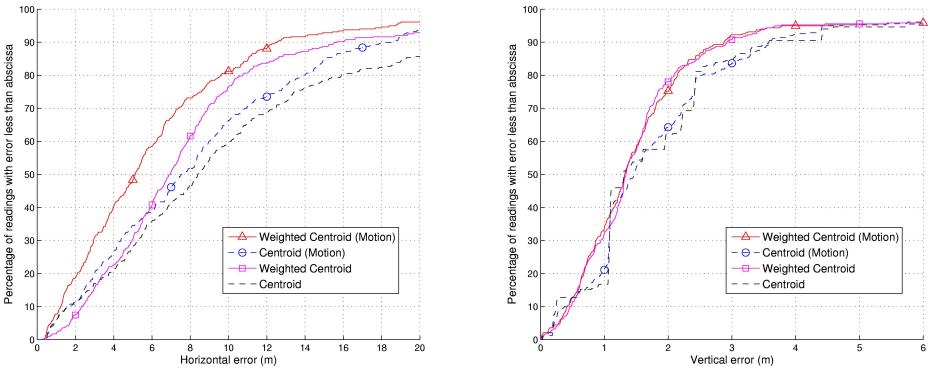


Fig. 13. Cumulative distribution comparing the accuracy of four presented algorithms tested with 20 minutes trace of RSSI measurements.

Table 1. Tracking performance summary. All values shown pertain to the location results of the "walking" data traces collected for about 20 minutes in a five-storied building, covering all the five floors during the measurement period.

Algorithm	**Mean hor. error**	**Mean vert. error**	50% conf. level (m) **Horizontal**	**Vertical**	75% conf. level (m) **Horizontal**	**Vertical**
Centroid	10.77	1.73	8.41	1.30	13.48	2.43
Weighted Centroid	8.53	1.51	6.87	1.34	9.74	1.84
Centroid (with motion)	9.01	1.84	7.49	1.47	12.47	2.42
Weighted Centroid (+ motion)	6.68	1.53	5.14	1.32	8.57	1.98

stairs, the motion status was recorded explicitly so as to log insertion points where we could manually insert the corresponding positions and we used an interpolation script to obtain a ground truth location for each time stamp in the actual measurement log. This worked well and essentially made the comparison easier as we could make a point-to-point comparison between the location estimates and the ground truth.

7.2 Results and Discussion.

We evaluate the four algorithms by computing the median accuracy (accuracy reported by 50% of the readings). Table 1 summarises the overall results of the algorithms. Without adding motion, Centroid and Weighted Centroid report median accuracies of 8.41 m and 6.87 m respectively. Adding motion improves these to 7.49 m and 5.14 m respectively. This is because the motion model filter utilises its predicted estimate for the position of the device, in addition to estimates calculated using current RSSI observations, to produce the new estimate (similar to that of a Kalman filter). The cumulative distribution shows even the 75^{th} percentile error reports less than 9 m error for weighted centroid with motion incorporated, given the fact that we have completely avoided the intensive radio-mapping process which is typically used in fingerprinting algorithms. Figure 14 reveals that a considerable portion of the error occurs when the user is at the

Table 2. Accuracy of floor estimation, represented on a per-floor basis (percentages).

Algorithm	All floors	Floors 2–5	Accuracy per floor				
			Floor 1	Floor 2	Floor 3	Floor 4	Floor 5
Centroid	70.9	79.4	45.2	72.5	80.3	84.4	83.3
Weighted Centroid	70.0	78.3	45.2	77.5	81.8	75.3	80.0
Centroid (with motion)	72.7	79.5	53.6	71.3	89.4	75.3	86.7
Weighted Centroid (+ motion)	75.1	82.2	53.6	80.0	90.0	75.0	86.0

extreme end of the corridor, as typically the extreme ends have much less access point densities. It is to note that the building has no nearby neighbouring buildings and hence the access points are solely used by the installations in the same building. Figure 14 also shows that the ground floor measurements do not report any location estimates in the southern extension of the floor. This is again due to the unavailability of the access points in that region. Considering the fact that 15% of the readings constitute either stair cases or the southern extension on the ground floor, the reported mean accuracy of the weighted centroid with motion, 6.68 m, is reasonable for a calibration-free approach.

Floor Identification. This subsection reports how the presented algorithms detects correct floor information. This is very important for many of the applications to identify at which floor a user is present. Table 2 gives the percentage of the measurement time, each algorithm reporting that the user is in the correct floor. For most of the measurements, the algorithm reports that the user is in the correct floor. It is particularly interesting to note that for the measurements in the southern extension on the ground floor the floor error increases. Here we did not have any access points mapped, hence any access points heard at that point were from higher floors, thereby pulling the floor estimates higher by two floors. Table 2 summarises the error in floor estimates reported by all the four algorithms on a per-floor basis. It is clear that the error in the first floor contributes to the maximum error, as all the algorithms consistently show an average of only 50% correct classification. Excluding the first floor measurements show that in principle we are able to identify the correct floor around 82% of the time. The table also shows that there is a modest improvement between the algorithms that use motion information over the ones without a motion model.

8 Conclusions

This paper addresses how to sense motion and location leveraging the existing infrastructure. Based on the thorough characterisation of RSSI measurements we developed a range of motion detection algorithms. We identified a rich set of features that could be gathered based on either temporal or spectral characterisation. Our motion detection algorithms exploiting the frequency domain characteristics report a precision and recall over 90%. It will be interesting to consider the complexity of each of the algorithms to analyse the tradeoff between accuracy and complexity of the presented algorithms.

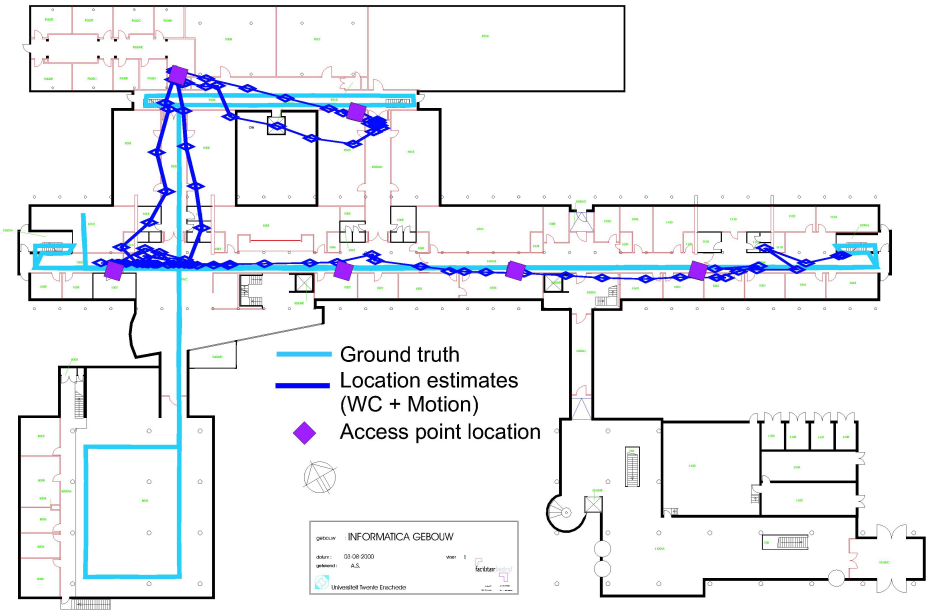


Fig. 14. Access point locations and estimated path overlaid on a floor plan (pertaining to ground floor measurements only). Comparing the trajectory of the ground truth and estimated location, emphasises the error mostly happens on the stairs and towards the extreme end of the corridor.

One possibility of extending this work is to use a combined set of features in a machine learning algorithm, to obtain finer accuracy and also explore the possibility of identifying more states like “cycling” or “driving”.

We have shown the benefit of combining motion information with location algorithms. A median error of approximately 5 m can be achieved without the use of calibration. We have validated our analysis by testing the algorithms in a typical setup used in many office environments, where access points are arranged linearly. However, these results cannot easily be generalised, as the results are very much dependent on the density and topology of the access points in the test area.

The improvements in the algorithms with motion incorporated, suggest that many of the calibration-intensive fingerprinting algorithms could use such a simple scheme for detecting users in the hallways, and restrict the fingerprints to the rooms, as we envisage that our method might not work as well there. This depends on the access point configuration; if there are access points also distributed along spatially separated axis it will result in considerable improvement because it will allow for better trilateration. In general, if the access points are deployed not only to provide good coverage for communication purposes, but if they are deployed keeping in mind that such infrastructures can be used for positioning purposes, we can expect much more improvement.

When incorporating the motion information we assumed the walking speed of the user is known, usage of other sensors which can actually give us the speed and direction,

for instance by using a combination of accelerometers, gyroscopes and magnetometer and incorporating map-matching methods and in combination with probabilistic methods like particle or Kalman filtering might be an suitable venue of future research.

As a general note, we expect that motion information we inferred can also give a cue of what degree of history size must be used for temporal smoothing of the location estimates (i.e., adaptive windowing) – for instance, when the device is moving history size can be set to a smaller value and when the device is still it can be set larger.

References

1. I. Anderson and H. Muller. Context awareness via GSM signal strength fluctuation. 4th International Conference on Pervasive Computing, Late breaking results, May 2006.
2. P. Bolliger, K. Partridge, M. Chu, and M. Langheinrich. Improving location fingerprinting through motion detection and asynchronous interval labeling. In *LoCA 2009, Tokyo, Japan*, LNCS, May 2009.
3. T. King and M.B. Kjærgaard. ComPoScan: Adaptive scanning for efficient concurrent communications and positioning with 802.11. In *MobiSys 2008*, 2008.
4. J. Krumm and E. Horvitz. LOCADIO: Inferring motion and location from wi-fi signal strengths. In *Mobiquitous 2004*, pages 4–13, August 2004.
5. A. LaMarca, J. Hightower, I. Smith, and S. Consolvo. Selfmapping in 802.11 location systems. In *Ubicomp 2005*, LNCS, pages 87–104, 2005.
6. L. Liao, D.J. Patterson, D. Fox, and H. Kautz. Inferring high-level behavior from low-level sensors. In *UBICOMP2003, Seattle, WA*, 2003.
7. K. Muthukrishnan, M.E.M. Lijding, N. Meratnia, and P.J.M. Havinga. Sensing motion using spectral and spatial analysis of WLAN RSSI. In *EuroSSC 2007, Lake District, UK*, volume 4793 of *LNCS*, pages 62–76, October 2007.
8. K. Muthukrishnan, N. Meratnia, M.E.M. Lijding, G.T. Koprinkov, and P.J.M. Havinga. WLAN location sharing through a privacy observant architecture. In *COMSWARE, New Delhi, India*. IEEE Computer Society, January 2006.
9. C. Randell and H. Muller. Context awareness by analysing accelerometer data. In B. MacIntyre and B. Iannucci, editors, *The Fourth International Symposium on Wearable Computers*, pages 175–176, 2000.
10. T. Sohn, A. Varshavsky, A. LaMarca, M.Y. Chen, T. Choudhury, I. Smith, S. Consolvo, J. Hightower, W.G. Griswold, and E. de Lara. Mobility detection using everyday GSM traces. In *UbiComp 2006, Irvine, CA*, pages 212–224, September 2006.
11. E. Weisstein. Spearman rank correlation coefficient. From MathWorld–A Wolfram Web Resource. http://mathworld.wolfram.com/.