User-Trained, Zero-Configuration, Self-Adaptive Opportunistic Wi-Fi Localization for Room-Level Accuracy

Jef Lathouwers
Maarten Weyn
Charles Vercauteren
dept. of Applied Engineering: Electronics-ICT
Artesis University College
Antwerp, Belgium
Email: jef.lathouwers@student.artesis.be
{maarten.weyn,charles.vercauteren}@artesis.be

***Abstract*—In this paper the possibility of room-level localization through Wi-Fi by using user collaboration and zero-configuration is investigated. User collaboration and zero-configuration means that it avoids the time-consuming training phase known by other systems such as fingerprinting and entering the floor plan. Fingerprints are created as soon as the users start to collaborate by providing their location and corresponding Wi-Fi data. A floor plan is not necessary as fingerprints are simply assigned to rooms without using coordinates. It is called opportunistic localization in the way that it relies on the already available infrastructure, thus no additional hardware needs to be installed. Using these methods, simulations show that a localization success rate of about 90% can be reached and that the system is able to cope with collaboration errors and a changed environment.**

***Keywords*-Positioning Systems; Indoor Wi-Fi Localization; Zero-Configuration; User Collaboration; Adaptive Algorithms.**

I. INTRODUCTION

Due to the popularity of mobile devices with a wireless interface and the increasing presence of wireless access points, indoor localization using Wi-Fi signal strengths is becoming more and more popular. The knowledge of the location of persons and assets is essential for context-aware computing in ambient intelligent environments.

A common technique that utilizes these signal strengths is pattern matching. This approach of matching radio frequency patterns, which are called fingerprints, was initially proposed by Bahl and Padmanabhan [1] in relation to their system RADAR. However, most current systems that apply fingerprinting require a so-called offline phase where the fingerprints are constructed before localization in the online phase can be performed. In this case, a fingerprint consists of the received signal strength of all received access points combined with the current position of the measurement.

Such systems have some implications. First of all, these systems rely on the data gathered in the offline training phase where the properties of the environment can be different than in the online localization phase. A changing environment can change the radio signal strength pattern dramatically because of multipath and shadowing. In addition, Wi-Fi access points can be replaced, removed or added also resulting in a different radio signal strength pattern. Consequently, the fingerprints become inaccurate and recalibration of the fingerprints is required. Secondly, making a radio map of the environment requires the environment to be known, thus a floor plan is needed.

In this paper, the following research question is addressed:

How is it possible to achieve room-level accurate localization using Wi-Fi through user collaboration and zero-configuration?

A system is proposed, similar to Redpin [2], which does not require a training phase. In addition, it does not initially need any information about the environment and adapts to changes in the environment. It is called opportunistic localization as it relies on the existing infrastructure. Hence no additional hardware needs to be installed. The key concepts for achieving this are user collaboration and a room-level approach. The system lets the users create and update the locations and fingerprints in order to achieve room-level accuracy. This is accomplished by giving the users the ability to tell the system where they are. Locations are identified by a name and ID and fingerprints are simply assigned to these IDs. No coordinates are used.

However, new problems arise when using user collaboration and zero-configuration. User collaboration comes along with a security issue as users can mislead the system. This can be accidentally or on purpose. As the system completely relies on the input of the users to build fingerprints and aims to be adaptive to the environment, it is hard to distinguish between collaboration errors or a changed environment. Furthermore, because neither a floor plan or coordinates are used, it is hard to link rooms as neighbors. When the system is able to detect motion patterns between measurements, the localization can be improved.

Although the general idea is the same as in Redpin, the

 ISBN: 978-1-61208-235-6

implementation is quite different. Firstly, Redpin focuses on mobile phones and tries to combine Bluetooth with GSM and Wi-Fi signal strengths. In contrast, our system aims to achieve the same results by only using Wi-Fi signals and using a probabilistic approach with a motion and measurement model. Secondly, Redpin relies on the fact that users always collaborate correctly, which will not be the case in real life.

The remainder of this paper is organized as follows. In Section 2 the architecture of the system is presented. Section 3 and 4 go more deeply into the fingerprinting process. Section 5 covers the results of experiments conducted at our Department of Applied Engineering and the conclusions are presented in Section 6.

II. Architecture

Figure 1 shows the architecture of the system. It consists of two basic components: a client application acting as a sniffer tool that gathers information about all Wi-Fi access points in range, and a server that stores fingerprints in a database and runs a localization algorithm. The system can be divided into three parts that are described below: user collaboration, fingerprinting and localization.

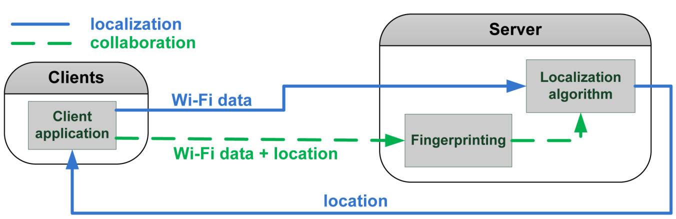


Figure 1. Architecture

A. User Collaboration

As noted before, fingerprints are constructed based on user collaboration. This consists of two tasks. Firstly, as the system initially does not know anything about any location, rooms need to be added to the system. This can easily be done in the client application by adding the name of the room. Next, fingerprints need to be constructed in order to train the localization algorithm. When connecting to the server a user can tell the system where he or she is by selecting the right location from a list. This is called a collaboration and is represented in Figure 1 by the dotted line. The system starts to perform localization as soon as there is one room added and one collaboration is made. Of course, the system will perform better as more rooms are added and more collaborations are made.

As our system aims to be able to cope with user collaboration errors, each collaboration will be rated and this rating will affect the impact of this collaboration to the fingerprints.

B. Fingerprinting

Fingerprinting for room-level accuracy needs a special approach. As rooms can be large, measurements from different positions in the same room can be totally different. Averaging all those measurements will lead to an artificial fingerprint. Thus, multiple fingerprints per room are needed. This is accomplished by clustering the different measurements. Data clustering is the process of grouping together similar multi-dimensional data vectors into a number of clusters. The multi-dimensional data in our case is the Wi-Fi measurement that is sent by the user in combination with a location (i.e., a collaboration). Each cluster will eventually lead to one fingerprint.

1) K-means Clustering: A widely used clustering method is k-means clustering [3]. Because there is no prior knowledge from training data, it is called unsupervised clustering. It aims to partition n observations into k clusters where each observation belongs to the cluster with the nearest mean. This mean is called the centroid of the cluster and will become a fingerprint in our case. Therefore a measure of distance is needed to compare observations (i.e., Wi-Fi measurements) to each other. A simplified version of a measurement model proposed by Weyn in [4] is used to accomplish this. It is developed to calculate the similarity between a measurement and a fingerprint. Besides the comparison between matching access point measurements, it takes into account, and also makes a difference between, an access point measurement from the fingerprint that is missing in the measurement and an access point measurement that is missing in the fingerprint in comparison to the measurement. This is not relevant when comparing two measurements in between and thus is not used. The algorithm returns a probability p, which represents the likelihood of similarity between two observations. The distance between those two observations is defined as $-log(p)$.

2) Adaptive: As noted before, fingerprints are constructed from the centroids of the different clusters. Also stated before, each collaboration will have a rating. This rating represents the likelihood that the collaboration is a correct one and affects the impact of that collaboration to the centroid. Therefore a weighted mean is used in the clustering process instead of a normal mean. The weight will not only be the collaboration rating but also includes a time and user rating as in Equation 1:

$$W_c = R_c^y \cdot R_t \cdot R_u \tag{1}$$

R_c and R_t are parameters representing the collaboration and time rating while R_u represents the user rating. R_c is powered to a value y, which will determine the influence of the collaboration rating on W_c. This is explained in more detail in Section 5.

These three ratings all have a different goal. The collaboration rating will filter collaboration errors, the time

 ISBN: 978-1-61208-235-6

rating will make the system adaptive to the environment and the user rating will filter collaborations from users that consistently collaborate faulty. These are called bad users.

The collaboration rating is calculated using the standard score of the collaboration. The standard score of a sample is the distance of a sample to the mean of the distribution divided by the standard deviation of the distribution [5]. As a result, it is possible to compare samples from different distributions and thus to rate collaborations belonging to different clusters. The collaboration rating is equal to $2p$ where p is the probability that a random sample has a greater standard score than the one being rated.

The time rating is calculated using Equation 2:

$$R_t = \begin{cases} 1 - \frac{1}{a}x & \forall x \in [1, a] \\ 0 & \forall x \notin [1, a] \end{cases} \tag{2}$$

All collaborations from a room are, starting with the most recent one, ordered in time and given a sequence number x. The linear function in Equation 2 will result in a decreasing time rating for older collaborations ending in a time rating of zero for the a^{th} collaboration. Consequently, value a determines how many collaborations are included in the sample space.

Equation 3 is used to calculate the user rating:

$$R_{u_1} = \frac{w_1 \cdot R_{u_0} + w_2 \cdot \overline{R_c*}}{w_1 + w_2} \tag{3}$$

The new user rating R_{u_1} is the weighted mean of the previous user rating R_{u_0} and the average collaboration rating from all collaborations from that user during the last a collaborations. This is denoted as $\overline{R_c*}$. When a user rating falls below a threshold t, all collaborations from that user are expelled from the sample space.

C. Localization

Localization is done when a user connects to the server and sends the Wi-Fi data gathered by the client application to the server. A Bayesian approach, as discussed in [6], can be used for computing a probability for each room. This is shown in Equation 4.

$$p(l|o) = \frac{p(o|l) \cdot p(l)}{p(o)} = \alpha \cdot p(o|l) \cdot p(l) \tag{4}$$

$p(o)$ is the prior probability of an observation and can be replaced by a normalizing constant α since it is independent of the location. $p(l)$ is the prior probability for a location l and is calculated by the motion model. This is multiplied by $p(o|l)$, which is the posterior probability of an observation given a location. This is determined by the measurement model.

1) Motion model: The motion model can be denoted as in Equation 5.

$$p(l_t) = p(l_t|l_{t-1}) \cdot p(l_{t-1}) \tag{5}$$

$p(l_t)$ is the probability for a location at time t, which is the probability of that location at time $t-1$ multiplied by the motion probability of the location at time t given the location at time $t-1$.

These motion probabilities need to be learned automatically. This is accomplished by analyzing users their consecutive measurements to detect transitions between locations. Laplacian smoothing, also known as add-one smoothing [7], is used to calculate motion probabilities out of the number of transitions detected. Using this method there will always be a minimum probability for locations that have no detected transitions. For each combination of locations a probability is calculated using Equation 6.

$$p(l_1|l_0) = \frac{T_{l_1} + 1}{T_{l_x} + L} \tag{6}$$

$p(l_1|l_0)$ is the probability that someone will be at location l_1 after being at location l_0. The numerator consists of two parts. T_{l_1} represents the number of transitions detected between l_0 and l_1. This is added to 1, which will prevent a probability of zero in case of zero detected transitions. The denominator is the addition of the total number of transitions counted from location l_0 to any other location, represented by T_{l_x}, and the number of locations L that exists in the database. Using this equation the sum of probabilities for each room equals one.

The algorithm for the transition detection is shown in Algorithm 1. Because the system will not be 100% reliable in its localization, some filtering is used to detect transitions. l_t represents the calculated location of a certain user at time t. At this time t, a possible transition from $t-2$ to $t-1$ is investigated. Checking on l_{t-2} and l_t prevents a transition detection when the system returned one aberrant location in between other similar locations.

Algorithm 1 Transition detection

```
if l_{t-2} = l_{t-1} or (l_{t-3} ≠ l_{t-1} and l_{t-2} ≠ l_t) then
    Transition_{l_{t-2} → l_{t-1}} detected
else
    No transition detected
end if
```

2) Measurement model: An algorithm proposed by Weyn in [4] is used for computing the probability between a measurement and a fingerprint. The problem of hardware variance between training and localization devices, as discussed by Tsui *et al.* [8], is also tackled by this algorithm.

III. Optimal Number of Clusters

Another issue is the detection of the optimal number of clusters per room. Each cluster will lead to one fingerprint by calculating a weighted average. If too less clusters exist for a large room, averaging those Wi-Fi measurements will result in incorrect fingerprints and higher localization error rates.

By running the k-means algorithm multiple times with a different number of clusters, starting with one, the number of clusters will vary per room if this iteration stops when the average distance from all samples to their centroids does not exceed a threshold i. As a result, the higher the value for this parameter is chosen, the less clusters and thus fingerprints are created. Using this method, more fingerprints will be formed in larger rooms or in rooms where the radio frequency patterns differ a lot.

A value of 0.75 for i was chosen for the simulations described in Section 5 as this minimized the localization error rate. This value corresponds to an average of 2.2 fingerprints per room. This test is not executed in other environments so no general conclusion for the optimal value of i can be drawn yet.

IV. A Changing Environment

Samples are defined as outliers when they are beyond the inner fence of the box plot of the sample space as explained in [5]. Because it is avoidable that collaboration errors become a new cluster and thus a new fingerprint, all collaborations that are marked as outliers are expelled from the sample space when trying to find the optimal number of clusters in the way that is explained above. If the number of clusters is found, the k-means algorithm is run one more time, this time including the outliers. Accordingly, the new centroid of the cluster where the outlier is assigned to will move towards the outlier with the result that the distance from the outlier to the centroid will decrease. This will especially happen if more collaborations similar to the outlier will be added. If this happens there is possibly a change in the environment and because the centroid is moved to the outliers, there is a chance that the former outliers will not be classified as an outlier anymore. This is because all collaborations are rated again after each new collaboration. Consequently, they are added to the sample space for finding the optimal number of clusters and possibly a new cluster will be formed.

V. Results

The 3rd and 4th floor of the Department of Applied Engineering at the Artesis University College of Antwerp were used as a test-bed to analyze the accuracy of the system. A floor plan can be seen in Figure 2. Our system aims to be both adaptive to the environment and being able to filter collaboration errors, but it will not be possible to achieve maximum results on both goals. As a result, two configurations of the system are proposed. These are called the adaptive configuration and the filter configuration. Each of them will perform better on one of the goals and worse on the other. Extremely put, the adaptive configuration will accept all collaborations whether they fit with the existing fingerprints or not, while the filter configuration will be suspicious to aberrant collaborations. As a result, the adaptive configuration will be sensitive for collaboration errors, but also will adapt faster to a changed environment (e.g., a replaced access point). In contrast, the filter configuration ignores collaboration errors but the downside for this configuration is the slower adaption to a new environment. This distinction is made as one can choose what configuration fits best his requirements.

The difference between both configurations is the value for parameter y in Equation 1, which is the power of the collaboration rating R_c. This power defines the influence of the collaboration rating on the collaboration weight in the k-means clustering process. The higher the value for y is chosen, the more the collaboration rating will affect the total collaboration weight and thus the impact of that collaboration to the weighted average of a cluster (i.e., fingerprint). A collaboration with a low collaboration rating will thus affect the fingerprint less in the filter configuration than in the adaptive configuration.

Several simulation tests are performed, each testing a different aspect of our system. Because of the difficulty of simulating motions from users, tests A, B, C and D are performed without using the motion model explained in Section 2. Value y in Equation 1 is chosen 0.8 for the filter configuration and 0.25 for the adaptive configuration, a value of 50 is used for a in Equation 2. These y-values are chosen this way in order to show the difference between the two configurations clearly. Choosing them more closely to each other will give similar results, but more moderate. The influence of value a is explained in test D.

A. Localization Rates

Figure 2 shows the localization success rates for each room. Averaging these rates taking into account the surface area of the rooms, an overall success rate of 89.19% is reached. These results are obtained using the adaptive configuration and during optimal circumstances where collaborations were done correctly and performed at different positions in each room.

B. Random Collaboration Errors

Figure 3 shows the localization success rates in case of random collaboration errors. The two bars at the left represent the localization success rates in perfect conditions where all collaborations were done correctly. Both configurations reach a localization success rate of about 90%.

The localization success rates in case collaboration errors were made are shown more to the right. The filter config-

 ISBN: 978-1-61208-235-6

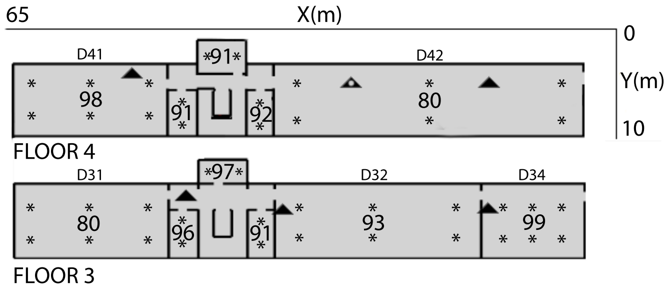


Figure 2. Floor plan Department of Applied Engineering. Access points are represented by a triangle, collaboration points by an asterisk. The numbers in the rooms are the localization success rates.

uration remains more stable than the adaptive configuration when the collaboration error rate increases. This is because the collaboration errors affect the fingerprints more in the adaptive configuration than in the filter configuration. As a result, the fingerprints will move more towards the collaboration errors in the adaptive configuration and the localization success rate will decrease faster in case of collaboration errors. In contrast, the filter configuration is able to filter those collaboration errors better and remains more stable.

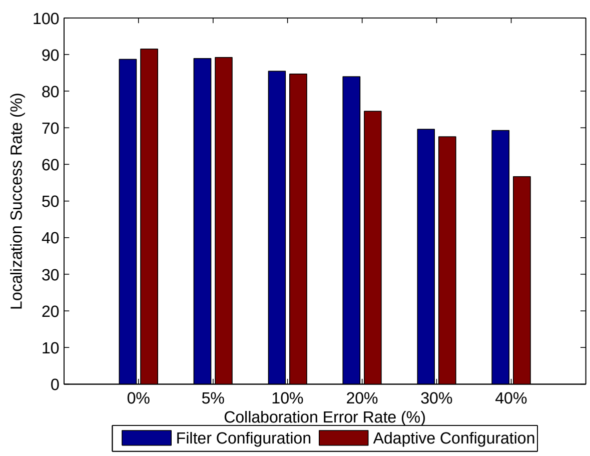


Figure 3. Localization success rates in case of random collaboration errors.

C. Bad User Collaboration Errors

Figure 4 shows the effect on the localization success rates in case of a bad user. The purpose of the system is to filter his or her collaborations. This is tested by comparing two simulations of the same configuration. Both simulations start with the same fingerprints and during the simulation the same collaboration errors are added. The line with triangles shows the localization success rate if each of those collaboration errors are made by a different user. As more collaboration errors are done, the localization success rate decreases. In contrast, the localization success rate of the line with circles remains stable. The same collaboration errors are now done by only one user (i.e., a bad user). As a result, his or her user rating falls below threshold value t, which means all of his or her collaborations are expelled and only the correct localizations from the other users remain in the sample space. Threshold value t is chosen 0.25 during this simulation and for the weighted mean in Equation 3, values 10 and 1 are used for w_1 and w_2.

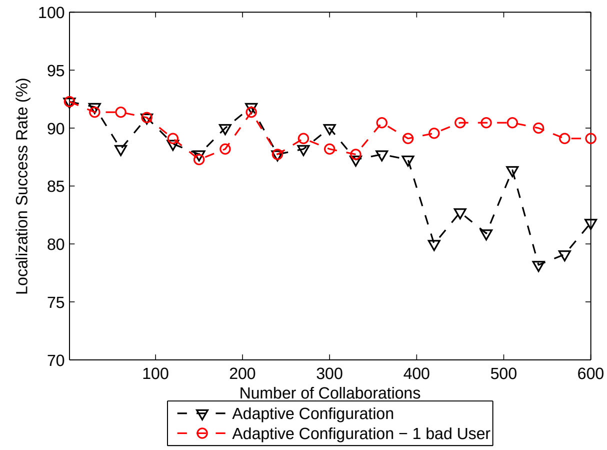


Figure 4. Localization success rates with a collaboration error rate of 20% contributed by 1 user (line with circles) or by multiple users (line with triangles).

D. Changed Environment

Figure 5 shows the localization success rates in case of a replaced Wi-Fi access point. In our simulation the access point from room D42 that is marked with a white dot in Figure 2 was replaced by another access point after 20 collaborations. The filter configuration will perform worse on this because the collaborations after the change differ from the fingerprints in the database and thus will be filtered. The adaptive configuration reaches the former localization success rate after about 200 new collaborations (i.e., approximately 18 collaborations per room). In contrast, the filter configuration needs about 580 new collaborations to recover. This corresponds to approximately 50 collaborations per room. As a value of 50 for parameter a was used during these simulations, one can conclude that the filter configuration filters all new collaborations until no old collaborations exist in the sample space anymore.

E. Motion Filtering

The simulation results above were obtained without the use of a motion model. Figure 6 shows the calculated motion probabilities, using the method explained in Section 2, for the rooms of floor 3 after a user walked around the building

 ISBN: 978-1-61208-235-6

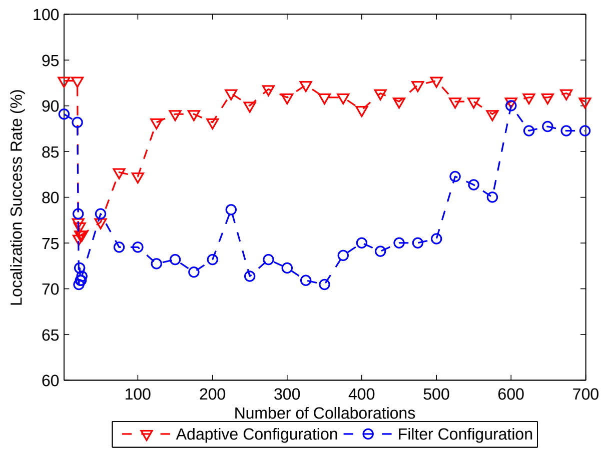


Figure 5. Localization success rates in case of a replaced access point after 20 collaborations.

from Figure 2. The user walked ten times from each room to each of its neighbors, each time staying one minute in each room.

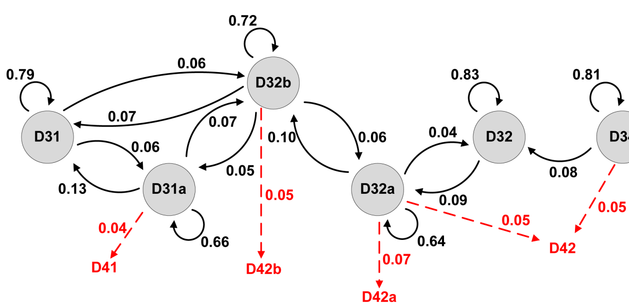
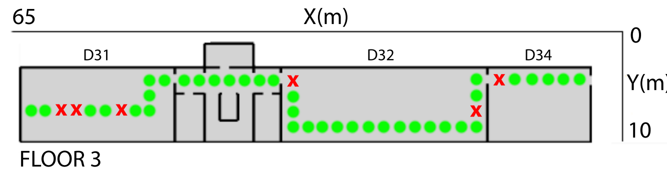


Figure 6. Motion probabilities for the rooms on floor 3. All probabilities smaller than 0.04 are not shown to improve the visibility of the figure.

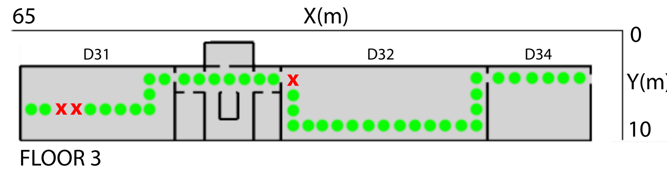
The dotted arrows represent motion probabilities from a room of floor 3 to a room of floor 4, which ideally would not exist if the localization and the transition detection algorithm would be working perfectly.

Using the probabilities from above, a new test was performed simulating a user that walks through the rooms of floor 3 from right to left at a steady pace. The localization results can be seen in Figure 7(a) and Figure 7(b), where a dot and a cross respectively represents a correct and a faulty localization.

Figure 7(a) shows the localization results without the use of the motion model. In contrast, Figure 7(b) shows the localization results with the same measurements, using the motion model with the motion probabilities from Figure 6. As can be seen, three localization errors from the simulation without the motion model are disappeared in the simulation with the use of the motion model. As expected, the motion



(a) Localization test without the motion model.



(b) Localization test using the motion model.

Figure 7. Localization test of a user that walks from right to left. A dot represents a correct localization and a cross represents a localization error.

model acts as a sort of filter on the localization process.

VI. CONCLUSION

As a general conclusion, we can state that this research has reached its goals. Using the aforementioned methods, a localization success rate of about 90% is reached on our department in case of 100% correct collaborations. This is comparable to the results of Redpin. Additionally, simulations show the ability of the system to cope with collaboration errors and to be adaptive to a changed environment. Depending on the adaptive nature of the environment and thrust of the users, different parameters to configure or adapt the system are proposed. Lastly, an experiment testing the motion model is presented, which shows the promising ability to filter localization errors.

The proposed systems enables context-aware applications without the need for an offline fingerprinting phase to calibrate the system. It uses the input of the users to create a trustworthy system, which is created during the use of the system itself.

REFERENCES

[1] P. Bahl and V. Padmanabhan, "RADAR: an in-building RF-based user location and tracking system," *INFOCOM 2000. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, vol. 2, 2000.

[2] P. Bolliger, "Redpin – adaptive, zero-configuration indoor localization through user collaboration," in *Proceedings of the First ACM International Workshop on Mobile Entity Localization and Tracking in GPS-less Environment Computing and Communication Systems*, San Francisco, USA, Sep. 2008.

[3] A. K. Jain, M. N. Murty, and P. J. Flynn, "Data clustering: a review," *ACM Comput. Surv.*, vol. 31, no. 3, pp. 264–323, Sep. 1999.

[4] M. Weyn, "Opportunistic seamless localization," Ph.D. dissertation, University of Antwerp, Mar. 2011.

 ISBN: 978-1-61208-235-6

[5] N. e-Handbook of Statistical Methods. (2012) http://www.itl.nist.gov/div898/handbook/ [May. 21, 2012].

[6] Widyawan, M. Klepal, and D. Pesch, "A Bayesian approach for RF-based indoor localisation," in *IEEE ISWCS*. Trondheim, Norway, October 2007.

[7] S. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*, 3rd ed. Prentice Hall, 2009, p. 863.

[8] A. Tsui, Y. Chuang, and H. Chu, "Unsupervised learning for solving RSS hardware variance problem in WiFi localization," *Mobile Networks and Applications*, vol. 14, pp. 677–691, 2009.