RSS and LEA Adaptation for Indoor Location using Fingerprinting

Pedro Mestre, *Member, IAENG,* Luís Reigoto, Luis Coutinho, Aldina Correia, *Member, IAENG,* Joao Matias, *Member, IAENG,* Carlos Serodio, *Member, IAENG*

Abstract—**Fingerprinting is an indoor location technique, based on wireless networks, where data stored during the offline phase is compared with data collected by the mobile device during the online phase. In most of the real-life scenarios, the mobile node used throughout the offline phase is different from the mobile nodes that will be used during the online phase. This means that there might be very significant differences between the Received Signal Strength values acquired by the mobile node and the ones stored in the Fingerprinting Map. As a consequence, this difference between RSS values might contribute to increase the location estimation error. One possible solution to minimize these differences is to adapt the RSS values, acquired during the online phase, before sending them to the Location Estimation Algorithm. Also the internal parameters of the Location Estimation Algorithms, for example the weights of the Weighted k-Nearest Neighbour, might need to be tuned for every type of terminal. This paper focuses both approaches, using Direct Search optimization methods to adapt the Received Signal Strength and to tune the Location Estimation Algorithm parameters. As a result it was possible to decrease the location estimation error originally obtained without any calibration procedure.**

Index Terms—**Fingerprinting Location, IEEE802.11, Direct Search Optimization Methods, LEA calibration, RSS adaptation.**

I. Introduction

Fingerprinting location estimation is a wireless network based technique on which the values of a given property of the wireless signals, received by a mobile device, are compared with a set previously stored values, called the Fingerprinting Map (FM). This location estimation technique belongs to the scene analysis methodologies [1], and it comprises two distinct phases: one on which data to generate the Fingerprinting Map is collected, called the offline phase; and a second phase, called the online phase, on which the estimation of the node location is made by comparing the collected data with the data stored in the Fingerprinting Map [2], [3], [4].

Typically, the property of the wireless signal used in Fingerprinting is the Received Signal Strength (RSS). During the offline phase, for each point of the spatial domain that will be mapped on the Fingerprinting Map, are collected RSS values from the wireless base stations that will be used as references. In a WiFi network those references are the infrastructure Access Points.

The signal received by the wireless node from each Access Point can be expressed as in Eq.1:

$$P_r = P_t + G_t + G_r - L_{sum} \tag{1}$$

where P_r is the received power (in dBm), P_t is the output power of the AP (in dBm), G_t and G_r the gains of the transmitting and the receiving antennas (in dB or dBi) and L_{sum} is the sum of all losses in the path between the transmitter and the receiver (in dB).

It is obvious that different types of mobile terminals migth have different types of antennas that consequently might have different values of gain and directivity, therefore, under the same operating conditions they will have different values for the RSS.

These differences can have a negative impact on the performance of Location Estimation Algorithms (LEA), used in the online phase, whenever the mobile node to be located is not the same that was used to collect data from which the Fingerprinting Map was generated. When different types of mobile terminals are used during the offline and online phases, it is expected that the values of RSS used to locate the mobile device, are different from those that were previously stored in the Fingerprinting Map.

Collecting wireless network data to generate the FM using every type of mobile terminal that could be used during the online phase it not a feasible solution because of the large number of existing mobile terminals that could potentially be used in such applications.

Restricting the access to the location estimation application only to a subset of mobile terminals, with which data was collected to generate the FM, might not be also the best solution because it would exclude many mobile terminals from the location system.

In this paper are presented two techniques to overcome the differences between mobile terminals, and minimize the impact of using different terminals during the offline and online phases. One of these techniques consists on adapting the RSS values before trying to locate the mobile device

Manuscript received March 05, 2012.

P. Mestre is with CITAB - Centre for the Research and Technology of Agro-Environment and Biological Sciences, University of Trás-os-Montes and Alto Douro, Vila Real, Portugal, email: pmestre@utad.pt

L. Reigoto is with UTAD - University of Trás-os-Montes and Alto Douro, Vila Real, Portugal, email: luisreigoto@gmail.com

L. Coutinho is with UTAD - University of Trás-os-Montes and Alto Douro, Vila Real, Portugal, email: luis_ coutinho_86@hotmail.com

A. Correia is with ESTGF-IPP, School of Technology and Management of Felgueiras Polytechnic Institute of Porto, Portugal, aldinacorreia@eu.ipp.pt

J. Matias is with CM-UTAD - Centre for the Mathematics, University of Trás-os-Montes and Alto Douro, Vila Real, Portugal, email: j_matias@utad.pt

C. Serodio is with CITAB - Centre for the Research and Technology of Agro-Environment and Biological Sciences, University of Trás-os-Montes and Alto Douro, Vila Real, Portugal, email: cserodio@utad.pt

and the second technique consists on adapting the internal parameters of the LEA to match the characteristics of mobile terminal in use.

The main objective of this work is to find a calibration procedure that enables the location estimation system to work correctly for a given mobile terminal, without the need to collect data in all (or too many) points of the scenario with it. The mobile terminal should be placed at some predefined calibration points where the calibration would be executed.

For the calibration procedure to be feasible in a real-life location estimation application, the calibration points that were used in this paper are located at the entrance hall of the building where data to do the calibration and location estimation tests was collected.

II. RSS and LEA Adaptation

In this section are presented the two techniques proposed to minimize the effect of using different types of mobile terminals. In both cases, adaptation of the acquired signals and LEA adaptation, Direct Search Methods are used. These methods are useful when derivative-based methods cannot be used [5], [6], for example when the values of the objective function are data collected from experiments, which is the case.

A. RSS Adaptation

During the online phase, the acquired values of RSS must be sent to the LEA. In the approach here presented, the raw values of the RSS acquired from the wireless network interface are not fed directly to the LEA. These values are adapted before being sent to the location algorithm. This adaptation, as in [7] and [8], consists in adding a calibration offset (c), Eq. 2, to the RSS value read from the wireless network interface:

$$RSS_{LEA} = RSS_{acquired} + c \tag{2}$$

where RSS_{LEA} is the value of the RSS sent to the LEA, $RSS_{acquired}$ is the raw value of RSS acquired from the wireless network interface and c is the calibration offset.

To determine the optimal value for the calibration offset two strategies are presented in this work. One uses the average error between the values acquired by the wireless node and the values stored in the FM, and the other uses Direct Search Optimization Methods.

1) Using the average error: In this first method the average error between the expected values of RSS, which are stored in the FM, and the values acquired by the mobile node, as in Eq. 3, are used as the calibration parameter, c, of Eq. 2:

$$c = \frac{1}{K}\sum_{i=0}^{K-1}\left(\frac{1}{N}\sum_{j=0}^{N-1}\left(FM_{i,j} - RSS_{i,j}\right)\right) \tag{3}$$

where K is the number of points used in the calibration process, N is the number of Access Points detected at the calibration point i, $RSS_{i,j}$ is the RSS value of Access Point j at point i and $FM_{i,j}$ is the value of the RSS stored in the Fingerprinting Map for Access Point j at point i.

2) Using Direct Search Optimization Methods: This second procedure is based on the use of Direct Search Optimization Methods. These methods can be used in the optimization of both constrained and unconstrained optimization problems.

In this case we have an unconstrained optimization problem of the form:

$$\min_{x\in\mathbb{R}^n} \quad f(x) \tag{4}$$

where:

- $f : \mathbb{R}^n \to \mathbb{R}$ is the *objective function*.

The objective function, $f(x)$, for this particular problem is the average location error for the points of the scenario under test that were used during the calibration phase. The dimension of the problem is 1 because the only input value to the objective function is the value of the offset.

An API (Application Programming Interface) built using Java Technology, that implements the used optimization methods, developed by the authors and presented in [5] was used. For the optimization of this particular problem it was used the Nelder and Mead algorithm, which was implemented in the API as in [9], [10] and [11].

B. LEA Adaptation

The LEA adaptation consists in tuning the internal parameters of the location algorithm using Direct Search Optimization Methods, to adapt it to the mobile terminal characteristics.

In this case the values of the internal parameters of the algorithm might be subject to constraints, so we have a constrained problem.

A constrained problems has the form of 5:

$$\begin{array}{ll} \min\limits_{x\in\mathbb{R}^n} & f(x) \\ s.t. & c_i(x) = 0, i \in \mathcal{E} \\ & c_i(x) \leq 0, i \in \mathcal{I} \end{array} \tag{5}$$

where:

- $f : \mathbb{R}^n \to \mathbb{R}$ is the *objective function*;
- $c_i(x) = 0,\ i \in \mathcal{E}$, with $\mathcal{E} = \{1, 2, ..., t\}$, define the problem *equality constraints*;
- $c_i(x) \leq 0,\ i \in \mathcal{I}$, with $\mathcal{I} = \{t+1, t+2, ..., m\}$, represent the problem *inequality constraints*;
- $\Omega = \{x \in \mathbb{R}^n : c_i = 0,\ i \in \mathcal{E} \wedge c_i(x) \leq 0,\ i \in \mathcal{I}\}$ is the set of all feasible points, i.e., defines the *feasible region*.

As a case study, the optimization of the weights of the Weighted k-Nearest Neighbour algorithm is presented in this work. The following conditions were used:

- The objective function, f, is the average location error for the points of the testing scenario that were used during the calibration procedure;
- The input parameters of the objective function, x_i are the k weights of the k nearest neighbours (x_0 is the weight for the nearest neighbour and x_{k-1} is the weight of the further neighbour);
- Each weight, W_i, must satisfy the following constraint: $W_i \geq 0$.

To solve the optimization problem it was used an implementation of the Penalty and Barrier Methods [12] available

ISBN: 978-988-19252-1-3
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

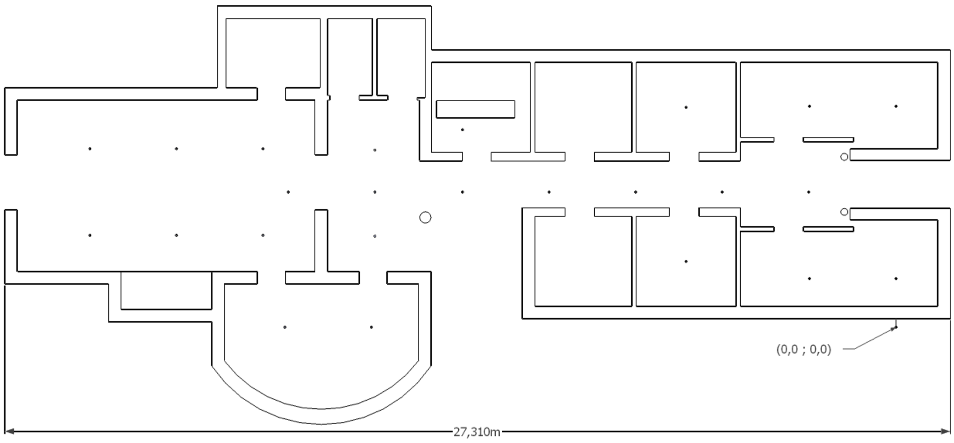

Figure 1. Map of the building where the tests were made.

on the above mentioned Java API. In Penalty and Barriers Methods the optimization problems are transformed into a sequence of unconstrained problems. These problems are then solved using the same algorithms that are used to solve unconstrained problems. This new sequence of unconstrained problems, that replaces the original problem, is defined by:

$$\Phi(x_k, r_k) : \min_{x_k \in \mathbb{R}^n} f(x_k) + r_k p(x) \qquad (6)$$

where Φ is the new objective function, k is the iteration, p is a function that penalises (penalty) or refuses (barrier) points that violates the constraints and r_k is a positive parameter.

In this work the Penalty and Barrier Methods were used with the Hooke and Jeves algorithm [13] as internal method. As penalty function it was used a Non-stationary Penalty function [14].

III. Testing scenario and conditions

To test the feasibility of the above presented procedures, data was collected in the scenario depicted in Fig. 1, which are the the headquarters of a local scouts group. These data were used to generate the Fingerprint Maps, calculate the calibration offset, tune the LEA and to make the location estimation tests.

Data was collected using two Android smartphones, with different sizes and from two different manufacturers, that will be referred as Smartphone 1 and Smartphone 2.

In the scenario a total of 24 points were used to collect data with both terminals. For each point of the scenario, which are marked in Fig. 1, a total of 20 samples was taken. Those samples were acquired using an application developed for Android, Fig. 2.

Since data collected with this application are going be used to test the several calibration procedures, this application does not make any location estimation. It only stores data in XML (eXtensible Markup Language) files for later processing. It is then possible to test offline all LEA and calibration procedures exactly with the same data.

Although Android smartphones were used in these experiments, any other type of smartphone and Operating System could be used, as long as the acquired values can be expressed in dBm.

For the tests presented in the work, three of the classic LEA were used:

- Nearest Neighbour (NN) – which considers that the coordinates (in the spatial domain) of the nearest point (in the signal domain) are the coordinates of the current location of the mobile terminal;
- k-Nearest Neighbour (kNN) – where the average of the coordinates (in the spatial domain) of the k nearest points (in the signal domain) are considered as the current coordinates of the mobile terminal;
- Weighted k-Nearest Neighbour (WkNN) – this LEA is similar to the previous, however a weighted average is used instead.

For the tests presented in this work it was considered for kNN and WkNN that $k = 3$, and the weights 0.7, 0.2, 0.1, as in [15], were used for WkNN (from the nearest to the further neighbour). There is an exception for the weights on those tests where the LEA parameters were calibrated for the mobile terminal under test.

For the calibration procedures, whose results are presented in Section IV, there were used different number of calibration points. Although the use of all points, belonging to the scenario, in the calibration procedure is not a feasible solution in a real-life application, results obtained in such conditions are presented.

Figure 2. Android application developed to acquire the data.

ISBN: 978-988-19252-1-3
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

Table I
COMPARISON OF THE USED LEA, USING BOTH SMARTPHONES AND THE FM GENERATED WITH DATA FROM SMARTPHONE 1

	Smartphone 1			Smartphone 2		
	NN	kNN	WkNN	NN	kNN	WkNN
Prec. (m)	0.27	1.68	0.91	1.42	1.83	1.56
StDev (m)	1.01	0.87	0.72	1.70	1.09	1.15
MaxErr (m)	8.67	7.65	8.12	6.32	6.32	5.25
MinErr (m)	0.00	0.00	0.08	0.00	0.00	0.13

Table II
COMPARISON OF THE USED LEA, USING BOTH SMARTPHONES AND THE FM GENERATED WITH DATA FROM SMARTPHONE 2

	Smartphone 1			Smartphone 2		
	NN	kNN	WkNN	NN	kNN	WkNN
Prec. (m)	1.38	1.83	1.52	0.34	1.68	0.94
StDev (m)	1.58	0.94	1.02	1.00	0.90	0.65
MaxErr (m)	8.67	6.77	8.03	5.77	3.57	5.00
MinErr (m)	0.00	0.00	0.15	0.00	0.00	0.15

Table III
COMPARISON OF THE USED LEA USING SMARTPHONE 2, FM GENERATED WITH SMARTPHONE 1, AND CALIBRATION VALUES CALCULATED BY ERROR AVERAGE.

	All Points			3 Points			1 Point		
	NN	kNN	WkNN	NN	kNN	WkNN	NN	kNN	WkNN
Prec. (m)	1.27	1.83	1.49	1.32	1.85	1.54	1.59	1.94	1.71
StDev(m)	1.66	1.10	1.13	1.71	1.11	1.17	1.96	1.26	1.42
MaxErr(m)	6.32	5.33	5.25	6.32	5.33	5.25	8.30	5.89	6.83
MinErr(m)	0.00	0.00	0.13	0.00	0.00	0.13	0.00	0.00	0.13

Table IV
COMPARISON OF THE USED LEA USING SMARTPHONE 1, FM GENERATED WITH SMARTPHONE 2, AND CALIBRATION VALUES CALCULATED BY ERROR AVERAGE.

	All Points			3 Points			1 Point		
	NN	kNN	WkNN	NN	kNN	WkNN	NN	kNN	WkNN
Prec. (m)	1.02	1.84	1.31	1.05	1.87	1.34	1.17	1.78	1.38
StDev(m)	1.55	1.05	1.08	1.68	1.05	1.16	1.50	0.92	0.98
MaxErr(m)	8.67	6.77	7.60	8.67	6.77	7.60	8.67	6.77	7.60
MinErr(m)	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

These results are presented as reference values for comparison purposes, and will be considered as the best results that could be achieved with the location system. The calibration procedures were also made using three and one calibration points. These points are located at the entrance hall of the building.

IV. NUMERICAL RESULTS

In this section are presented the results obtained for both smartphones with the above presented LEA and calibration procedures.

A. Reference Values

Table I and Table II present the values for the location precision (Prec.), standard deviation (StDev), maximum location error (MaxErr) and minimum location error (MinErr) for the Nearest Neighbour, k-Nearest Neighbour and Weighted k-Nearest Neighbour algorithms, using both smartphones. For Table I the FM was generated using data acquired with Smartphone 1 and for Table II data acquired with Smartphone 2.

As it was already expected, better values are obtained when the mobile device used during the online phase is the same that was used during the offline phase.

B. RSS Adaptation using the Average Error

The numerical results of the tests made using the average error as the calibration value, are presented in Table III and Table IV. Table III presents the values for Smartphone 2 using the FM generated with Smartphone 1 and Table IV presents the values for Smartphone 1 using the FM generated with data from Smartphone 2.

Comparing the values in Tables I and III (Smartphone 2) it can be observed that when all points are used in the calibration procedure there is a boost in the precision and standard deviation values for the Nearest Neighbour and Weighted k-Nearest Neighbour algorithms, and a better value for the maximum error in k-Nearest Neighbour. Similar behaviour can be observed for the value of precision and maximum error, when three points are used in the calibration procedure. When the calibration procedure is made using only one calibration point, worse results are obtained.

For Smartphone 1 the absolute values are slightly better than for Smartphone 2, when comparing results from Tables II and IV. With this smartphone also more precision values were better than the original ones, without calibration. As expected the absolute values for the precision are better when more points are used during the calibration process.

C. RSS Adaptation using Direct Search Methods

Numerical results of the tests made with the optimization methods are presented in Table V and Table VI, where are presented the values obtained for Smartphone 2 and Smartphone 1, respectively.

For Smartphone 2 (Table V), all values of precision and standard deviation, using the calibration procedure with all points and three points, are better than the original ones. Even with a single calibration point, the values of the precision and standard deviation for the Nearest Neighbour and k-Nearest Neighbour are better.

When comparing the results obtained with Smartphone 1, it can be concluded that all values for the precision are better than the original ones, except for one that had the same value. However the same is not true for the obtained standard deviation values. Nevertheless, those difference in the standard deviation values can be considered small.

D. LEA Adaptation using Direct Search Methods

Table VII presents the values of the weights obtained for Smartphone 1 (SP1) and Smartphone 2 (SP2), using all points of the scenario, three points and one point in the calibration procedure.

Using these weights a new set of tests was made using data of both smartphones and the Weighted k-Nearest Neighbour algorithms, whose results are presented in Tables VIII and IX.

ISBN: 978-988-19252-1-3
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

Table V
COMPARISON OF THE USED LEA USING SMARTPHONE 2, FM GENERATED WITH SMARTPHONE 1, AND CALIBRATION VALUES CALCULATED USING DIRECT SEARCH METHODS.

	All Points			3 Points			1 Point		
	NN	kNN	WkNN	NN	kNN	WkNN	NN	kNN	WkNN
Prec. (m)	1.22	1.75	1.44	1.27	1.76	1.49	1.31	1.75	1.65
StDev(m)	1.57	1.01	1.02	1.66	1.00	1.13	1.59	1.01	1.35
MaxErr(m)	6.32	4.51	5.25	6.32	4.51	5.25	6.32	4.51	6.83
MinErr(m)	0.00	0.00	0.13	0.00	0.00	0.13	0.00	0.00	0.13

Table VI
COMPARISON OF THE USED LEA USING SMARTPHONE 1, FM GENERATED WITH SMARTPHONE 2, AND CALIBRATION VALUES CALCULATED USING DIRECT SEARCH METHODS.

	All Points			3 Points			1 Point		
	NN	kNN	WkNN	NN	kNN	WkNN	NN	kNN	WkNN
Prec. (m)	0.97	1.75	1.28	1.06	1.78	1.34	1.06	1.83	1.35
StDev(m)	1.51	0.92	1.06	1.73	1.01	1.18	1.73	0.94	1.19
MaxErr(m)	8.67	6.77	7.60	8.67	6.77	7.60	8.67	6.77	7.60
MinErr(m)	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

For the calibration procedure using all points, the obtained weights confirm that, from the used algorithms, Nearest Neighbour is the one with better precision values. Consequently, for this case better precision values (comparing to Table I and II) were obtained for both smartphones (using WkNN). Similar values were also obtained for Smartphone 2 when one calibration point was used.

However the results obtained with Smartphone 2 using three calibration points, and for Smartphone 1 using three and one calibration points, are worse than the original values. Also to be noticed that except for Smartphone 2 with all calibration points, none of the other values for the precison are better than those obtained for the previous calibration procedure (RSS adaptation).

Even though the use of direct search methods to optimize the LEA appears to have no positive impact in the performance of the location system, a new set of tests was made. While in the previous tests the LEA was calibrated using the raw RSS values acquired by the wireless network interface, in this new set of tests the two calibration procedures were used together, i.e., the new calibration procedure consists in using Direct Search Methods to adapt the RSS values and then do the optimization of the LEA, also using Direct Search Methods.

Table VII
WEIGTHS OBTAINED WITH THE PENALTY AND BARRIER METHODS FOR BOTH SMARTPHONES.

	All Points	3 Points	1 Point
SP1	(1.00;0.00;0.00)	(0.31;0.29;0.40)	(0.39;0.55;0.15)
SP2	(1.00;0.00;0.00)	(0.52;0.24;0.24)	(1.00; 0.00;0.00)

Table VIII
RESULTS OBTAINED WITH SMARTPHONE 2 AND WEIGHT OPTIMIZATION OF WEIGHTED K-NEAREST NEIGHBOUR USING DIRECT SEARCH METHODS.

	All Points	3 Points	1 Point
Prec. (m)	1.42	1.66	1.42
StDev(m)	1.70	1.01	1.70
MaxErr(m)	6.32	5.49	6.32
MinErr(m)	0.00	0.00	0.00

Table IX
RESULTS OBTAINED WITH SMARTPHONE 1 AND WEIGHT OPTIMIZATION OF WEIGHTED K-NEAREST NEIGHBOUR USING DIRECT SEARCH METHODS.

	All Points	3 Points	1 Point
Prec. (m)	1.38	1.92	1.87
StDev(m)	1.58	0.98	0.91
MaxErr(m)	8.67	6.47	7.60
MinErr(m)	0.00	0.15	0.36

Table X
WEIGTHS OBTAINED WITH THE PENALTY AND BARRIER METHODS FOR BOTH SMARTPHONES.

	All Points	3 Points	1 Point
SP1	(1.00;0.00;0.00)	(1.00;0.00;0.00)	(1.00;0.00;0.00)
SP2	(1.00;0.00;0.00)	(1.00;0.00;0.00)	(1.00;0.00;0.00)

The weights obtained for these new set of tests are presented in Table X, and for all tests it is confirmed that the best values for precision are obtained using the Nearest Neighbour algorithm.

Table XI and Table XII present the results obtained with both smartphones using these weights. The obtained values for the precision are better than the original values (Tables I and II) and the values obtained using the RSS calibration (Tables V and VI). However the same is not true for the obtained values of standard deviation and the maximum location error.

Table XI
RESULTS OBTAINED WITH SMARTPHONE 2 AND WEIGHT OPTIMIZATION OF WEIGHTED K-NEAREST NEIGHBOUR USING DIRECT SEARCH METHODS.

	All Points	3 Points	1 Point
Prec. (m)	1.25	1.17	1.50
StDev(m)	1.57	1.66	1.90
MaxErr(m)	6.32	6.32	8.30
MinErr(m)	0.00	0.00	0.00

Table XII
RESULTS OBTAINED WITH SMARTPHONE 1 AND WEIGHT OPTIMIZATION OF WEIGHTED K-NEAREST NEIGHBOUR USING DIRECT SEARCH METHODS.

	All Points	3 Points	1 Point
Prec. (m)	0.97	1.04	1.06
StDev(m)	1.51	1.72	1.73
MaxErr(m)	8.67	8.67	8.67
MinErr(m)	0.00	0.0	0.00

ISBN: 978-988-19252-1-3
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

V. Conclusion and Future Work

In this paper two techniques to minimize the location estimation error, in indoor environments, using wireless networks and fingerprinting were presented. These techniques were divided into two main categories:

- RSS adaptation – where the RSS values acquired during the online phase are adapted before being sent to the LEA;
- LEA adaptation – on which the internal parameters of the LEA are tuned to suite the characteristics of the mobile terminal used during the online phase.

The objective of these techniques is to overcome the differences between mobile terminals and consequently try to minimize location estimation error.

Although the used optimization methods require some computational power, usually not available in mobile platforms such as smartphones, the used API was designed for remote access [5]. This means that the calibration procedure is not executed locally on the CPU of the mobile node but remotely on a server.

For all tests, although the use of all points of the scenario in the calibration procedure had the best results, it is not feasible in a real life scenario. So, tests were also made using one and three calibration points located at the entrance hall of the building used during the tests. These tests also had interesting results, especially those made using three calibration points. As it was expected, when more calibration points are used, better are the achieved results.

In the case of RSS adaptation, from the two proposed techniques, the one with better results was the use of direct search methods. Considering only the tests with three and one calibration points, the best value obtained with Smartphone 1 was a reduction of the location error by 10.56% and 23.91% for Smartphone 2 (both using three calibration points).

Using direct search methods to calibrate the values of RSS and tune the internal parameters of the LEA, it was possible to increase even further the precision of the location estimation. If used with Weighted k-Nearest Neighbour, Direct Search Optimization Methods can be used to adapt the algorithm to the mobile terminal, and it can be used for automatic LEA selection (choosing between Nearest Neighbour, k-Nearest Neighbour and Weighted k-Nearest Neighbour) by adjusting the algorithm weights. In the presented tests it was confirmed that, for the test scenario and the used smartphones, Nearest Neighbour was the algorithm with the best precision values.

Considering only the tests made with three and one calibration points, the best values obtained when the two optimization procedures were used together are 25.00% for Smartphone 2 and $31,58\%$ for Smartphone 1 (using three calibration points).

Despite the fact that precision was used in the objective function of the optimization problems, other parameters such as the standard deviation, maximum error, minimum error or a combination of some of them could be used. This is an option to be explored in future developments of this work.

As future work, also the use of the proposed calibration procedures together with alternative techniques to build the Fingerprinting Map, such as the use of propagation models [8], will be considered.

Furthermore, other types of LEA, such as the ones based on Fuzzy Logic, e.g. [7] and [16], could benefit from the calibration procedures presented in this paper.

Acknowledgement

This work is supported by European Union Funds (FEDER/COMPETE - Operational Competitiveness Programme) and by national funds (FCT - Portuguese Foundation for Science and Technology) under the project FCOMP-01-0124-FEDER-022696.

References

[1] J. Hightower and G. Borriello, "Location sensing techniques," University of Washington, Department of Computer Science and Engineering, Seattle, Tech. Rep., July 2001.

[2] A. Taheri, A. Singh, and E. Agu, "Location fingerprinting on infrastructure 802.11 wireless local area networks," in *LCN '04: Proceedings of the 29th Annual IEEE International Conference on Local Computer Networks*. Washington, DC, USA: IEEE Computer Society, 2004, pp. 676–683.

[3] C. Komar and C. Ersoy, "Location tracking and location based service using IEEE 802.11 WLAN infrastructure," in *European Wireless*, Barcelona Spain, February 2004, pp. 24–27.

[4] 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, pp. 775–784 vol.2, 2000.

[5] P. Mestre, J. Matias, A. Correia, and C. Serodio, "Direct Search Optimization Application Programming Interface with Remote Access," *IAENG International Journal of Applied Mathematics*, vol. 40, no. 4, pp. 251–261, Nov 2010.

[6] A. Correia, J. Matias, P. Mestre, and C. Serodio, "Derivative-free Nonlinear Optimization Filter Simplex, International," *Journal of Applied Mathematics and Computer Science (AMCS)*, vol. 4029, no. 4, pp. 679–688, Dec 2010.

[7] C. Serodio, L. Coutinho, H. Pinto, and P. Mestre, "A comparison of multiple algorithms for fingerprinting using IEEE802.11," in *Lecture Notes in Engineering and Computer Science: Proceedings of The World Congress on Engineering 2011, WCE 2011, 6-8 July, 2011, London, U.K.*, 2011, pp. 1710–1715.

[8] P. Mestre, C. Serodio, L. Coutinho, L. Reigoto, and J. Matias, "Hybrid technique for fingerprinting using IEEE802.11 wireless networks," in *Indoor Positioning and Indoor Navigation (IPIN), 2011 International Conference on*, sept. 2011, pp. 1 –7.

[9] J. Dennis and D. Woods, "Optimization on microcomputers: The nelder-mead simplex algorithm," *New Computing Environments: Microcomputers in Large-Scale Computing,*, vol. Wouk, A., ed., pp. 116–122, (1987).

[10] C. Kelley, *Iterative Methods for Optimization*. Philadelphia, USA: Number 18 in Frontiers in Applied Mathematics, SIAM, (1999).

[11] J. Lagarias, J. Reeds, M. Wright, and P. Wright, "Convergence properties of the nelder–mead simplex method in low dimensions," *SIAM Journal on Optimization*, vol. 9, no. 1, pp. 112–147, (1998).

[12] A. Correia, J. Matias, P. Mestre, and C. Serodio, "Direct-search penalty/barrier methods," in *Lecture Notes in Engineering and Computer Science - World Congress on Engineering 2010*, vol. 3. London, UK: IAENG, (2010), pp. 1729–1734.

[13] R. Hooke and T. Jeeves, "Direct search solution of numerical and statistical problems," *Journal of the Association for Computing Machinery*, vol. 8, no. 2, pp. 212–229, (1961).

[14] F. Y. Wang and D. Liu, *Advances in Computational Intelligence: Theory And Applications (Series in Intelligent Control and Intelligent Automation)*. River Edge, NJ, USA: World Scientific Publishing Co., Inc., (2006).

[15] P. Mestre, H. Pinto, C. Serodio, J. Monteito, and C. Couto, "A multi-technology framework for LBS using fingerprinting," in *Industrial Electronics, 2009. IECON '09. 35th Annual Conference of IEEE*, Nov. 2009, pp. 2693 –2698.

[16] P. Mestre, L. Coutinho, L. Reigoto, J. Matias, A. Correia, P. Couto, and C. Serodio, "Indoor location using fingerprinting and fuzzy logic," in *Advances in Intelligent and Soft Computing*, no. 107. Springer-Verlag Berlin and Heidelberg GmbH & Co. K, 2011, pp. 363–374.