

Deploying a BTLE Positioning System: Practical Issues on Calibration

Ana M. Bernardos, Luca Bergesio, Eduardo Metola, Daniel Ortiz and José R. Casar

Abstract—Nowadays, the Bluetooth Low-Energy (BTLE) technology is integrated in numerous smartphones and beacons, but there is still an open challenge on delivering a presence-into-zone positioning system that facilitates the design and reconfiguration of the service zones and offers stable target tracking. In this paper, we report the experience gained from the deployment of a BTLE zone-based positioning system in a retail setting with defined service objectives. The algorithm powering the positioning system requires previous calibration, which comes to be a tedious and long invasive process when it is accomplished in a space with high people flow. The paper describes our work to minimize the calibration effort, by analyzing the effect of calibration samples reduction and proposing the application of a on-the-move calibration strategy that facilitates the calibration dynamics preserving the system’s correct recognition rate.

Index Terms—Indoor location system, location-based services, positioning technologies, context-aware services, calibration

I. INTRODUCTION

Indoor positioning systems are becoming popular in different environments, in particular for retailers, who are interested in their potential for inventory, customer/vendor behavioral analysis [1], physical space performance or notifications handling (information, *proximity marketing*, etc.). Delivering an operational indoor location system, capable of working with standard smartphones, requires to use the communication technologies available on these devices (i.e. Wi-Fi, Bluetooth). Since version 4.0, the Bluetooth specification enables the implementation of a low energy version. On top of the Bluetooth Low Energy (BLTE) stack, three main commercial location-oriented protocols have been developed and integrated in the smartphones: iBeacon [2], Eddystone [3] and AltBeacon [4]. They all provide a basic location method that uses the received signal strength (RSS) to determine when a smartphone is close to a beacon, i.e. a small electronic device that periodically broadcasts a BLTE signal.

Combining the RSS coming from different spatially-distributed beacons, it is possible to estimate the position a moving target (i.e. smartphone or moving beacon)

through the use of traditional positioning techniques. Results are similar to the ones obtained from other wireless technologies for *accurate positioning*, when the objective is to determine the coordinates of the moving target with respect to a reference system. Fingerprinting and channel modeling techniques have been widely explored and their limitations, documented in literature [5] [6] [7]. Due to the high variation of signal strength in indoor environments (because of reflections and absorption at surfaces), accurate location estimation (<1-2 m) is practically unfeasible with RSS-based techniques. Previous works such as [8] or [9] include systems and algorithms benchmarkings in semi-controlled scenarios.

But, for many services, positioning systems only need to deliver basic *proximity* discovery, i.e. a digital estimation which is activated when a smartphone detects a visible beacon, or *in-zone presence* detections, i.e. the zone where the moving target is, usually computed by merging information from different visible beacons. Most commercial solutions from beacons’ vendors offer off-the-shelf tools that enables proximity discovery, but with them, it is not feasible to build a system that provides continuous and reliable estimations for non-walled zones with sufficient reliability. In this paper, we describe our experience on the deployment of a BLTE positioning system in a real setting (a supermarket). Our positioning system is capable of providing both presence and proximity location to deliver notification and behavior analysis services on top of it. Our focus on the paper is not to go deep on the evaluation of the positioning strategy, but to provide some hints on how to improve and accelerate deployments in non-controlled scenarios, in particular by making easier and faster the calibration procedure.

II. BTLE INDOOR POSITIONING SYSTEM

A. System Requirements

In our experience with indoor location systems, we have found that practical deployments may impose restrictions that are not necessarily considered in the theoretical sphere. In this case, we are going to present our conclusions from a deployment of a location system

in a supermarket during work days. The design of the location system has been done following several requirements and restrictions, imposed by the deployment environment and basically dictated by both communication constraints and the need to guarantee customer privacy; the most relevant ones are:

- 1) For service purposes, the system must deliver both presence (zone-based) and proximity location estimations. As it is explained in Section III, the owner of the setting has identified 10 different zones of interest meaningful for the location service, which have a variable size and include different segments of corridors and open areas.
- 2) Position has to be computed in the mobile device, so the user has full control over it (device-centric system).
- 3) Information about the beacons composing the positioning infrastructure (in particular, their location, identifier and configured transmission power) has to be provided to the positioning algorithm in the mobile device, so it has to be available for download the first time that the user triggers the service and, in update mode, when modifications are done.
- 4) Location estimates have to be managed in an anonymized way when sent to the server managing history and support services.
- 5) The response time of the location must be sufficiently low to permit to locate a user walking through the smallest subzones.
- 6) The system has to be set with minimum effort, so deployment and calibration procedures have to be fast and automated.

B. Positioning algorithm

Our positioning system relies on an algorithm that estimates the zone where the mobile target is. The algorithm is referred within the text as *threshold algorithm* and requires an array of RSS thresholds to be calculated for each beacon visible from a given zone. To be considered inside the zone, the RSS received in the smartphone needs to fulfill all the thresholding conditions with respect to each visible beacon. As many algorithms based on fingerprinting or channel modeling techniques, this algorithm needs calibration to estimate the thresholds between zones. Assuming that $r_{m,n}$ is the RSS from the beacon m in the area n , being M the number of beacons per zone, the average RSS in a zone is:

$$r_n = \frac{1}{M} \sum_{m=1}^M r_{m,n} \quad (1)$$

The first step for the algorithm is to find the area $A_n(max)$ in which the computed in-zone average RSS $r_n(max)$ is the highest. Considering that each element



Fig. 1. Views exemplifying the distribution of beacons in an aisle.

of $D = [d_1, d_2, \dots, d'_n]$ is the difference of $r_n(max)$ and $r'_{n'}$, for every area $A_{n'}$ where $n \neq n'$, the algorithm will only provide a solution (L) if the difference of RSS with respect to the average RSS in the rest of the areas is over a certain threshold, to avoid false positives:

$$L = \begin{cases} A_n(max) & \text{if } d_{n'} > th_{n,n'}, \forall n \neq n' \\ out - of - zone & \text{otherwise} \end{cases} \quad (2)$$

As said before, each threshold th_{AB} is computed from a number of simultaneous RSS sample pairs taken from beacons both in zones A and B, when the user is physically in zone A. Depending on the size and shape of zone A, the calibration will require to repeat the sampling procedure in different positions distributed within the zone. Then, for every set of simultaneous measurements, the lower and upper bounds will be computed $[th_{min}, th_{max}]$, being the upper one the most restrictive. Representing the histogram of differences also enables to set different confidence levels different to 100% (which may be assumable for specific use cases), avoiding the number of *out - of - zone* situations. Although other alternative algorithms are available (e.g. a bayesian algorithm that includes information about the possible transitions among zones) [10], the threshold algorithm has been chosen for the real deployment due to the trade-off between complexity and accuracy.

C. System Components

The delivery of a functional positioning system that fulfills the requirements above implies the deployment of a support infrastructure (beacons and central processing unit - a local server, in this case, although the solution is also available in cloud mode) and to include the necessary logic in the mobile device. The system is prepared to work with Estimote Proximity Beacons for in-zone presence estimation (Fig. 1) and RadBeacon Dot for proximity location (although it is easy to include new hardware). As required, the positioning algorithm has been coded both for Android and iOS (APIs are provided

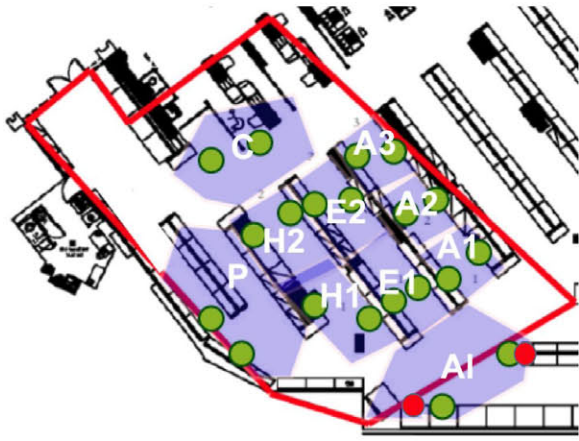


Fig. 2. Area to cover with the location system. PoI (point of interest) for proximity zones are in red, while green dots represent the position of the beacons.

for integration with existing client applications). Both infrastructure and calibration data are downloaded to each smartphone running the positioning functionality when the application managing location is launched for first time in the area of coverage of the system, and updated when the infrastructure data has changed. Bidirectional communication with the server is required, not only to download the beacons' information, but also to receive the results of the positioning procedure. In practical terms, this need requires the configuration of the setting's network to enable the communication between mobile devices and server.

III. BEACON DEPLOYMENT AND CALIBRATION

The map showing the area to cover is represented in Fig. 2 (red line). The zones of interest comprise six master zones (Pets, Household, Cleaning, Beauty, Beverages and Cashiers), three of them with two or three sub-zones inside (ten subzones in total). In particular, zones are as follows:

- 1 aisle, divided into three zones (2 x 1.7 m approximately), zones A1, A2 and A3 in Fig. 2.
- 2 aisles, each of them divided into two zones (3 x 1.7 m approximately). Respectively E1, E2 and H1, H2 in Fig. 2.
- Three open zones, C, P and AI respectively in Fig. 2.

For proximity location, the supermarket's owner defined two points of interest in a pop-up set and in a section of permanent shelves that were located at zone A.

The design phase included the design of the physical distribution for beacons (depending on the materials of the aisles and the products around), the number of beacons per zone (depending on the size and shape of the zone), the distribution of the calibration positions per zone and the number of samples to take per position,

together with the sampling orientation. Prior to proceed with the deployment, information about the physical distribution of the setting and the building materials was gathered, but the retrieved data were too general so it was impossible to create a propagation model of the Bluetooth signal to determine the number and setting of the beacons to deploy beforehand. For this reason, the final deployment design was decided after a on-site visit and a basic test.

So the initial step of the deployment was a propagation test to evaluate the signal fading due to the absorption of the aisles' materials. Several beacons were first installed between the aisles in a comfortable position. In-zone presence beacons were all of them configured to transmit at their maximum power. For calibration and testing, two different models of smartphones were used (two Apple iPhones 6S and three Samsung Galaxy S5). Some short walks carrying the two mobile devices (on hands and inside a pocket) enabled to store a significant number of RSS samples that served to evaluate the propagation scenario. The signal fading induced by the aisles' materials was favorable, so it was decided to deploy two beacons per zone (which is the minimum that the location algorithm needs to work in an stable way in laboratory tests).

The second step was the physical installation of the beacons. For the presence beacons, they were located at the top height of the shelves (at the maximum 2.3-2.5 m), at mid-distance of the zone's length and approximately one in front of the other in the case of the aisles (Fig. 2 shows a pair of beacons). The two proximity beacons were installed in two totems in the zone A, to detect when the customer approaches to a promotional product to deliver tailored notifications. The final deployment is shown in the Fig. 2.

The third step was the calibration of the presence location algorithms (proximity beacons did not require any calibration). Calibration requires the tester remains in a certain position and orientation, holding a smartphone to take the predefined number of RSS samples (the client application scans the RSS from all the visible beacons up to take a pre-defined number of samples). The process is repeated for each pre-established position and orientation. We have discarded the idea of building a dense fingerprint map for the deployment of the system, as the trade-off between time and accuracy benefits is not favorable. So the objective is to select few meaningful calibration positions inside each zone. Regarding the number of samples per position, the system's performance is supposed to increase with the samples volume up to a limit. In this unknown environment we opted by oversizing the sampling, so we were able to configure a sufficiently large dataset also for experimentation purposes. The calibration design finally included four positions in each zone; samples

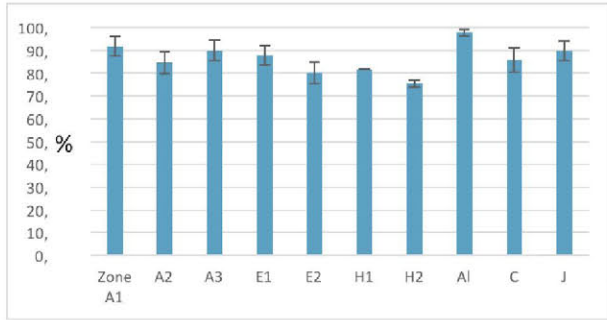


Fig. 3. Average Correct Recognition Rate (%) per zone with error bar of one standard deviation.

in four different orientations were taken (North-East-South-West), so 300 samples were finally stored for each calibration position. The entire process would have lasted more than three hours ($300 \text{ samples} \times 4 \text{ positions} \times 10 \text{ subzones} = 12\,000 \text{ seconds}$) if three testers were not have been simultaneously involved. The calibration was completed in about one hour and twenty minutes; as expected, it was a long task that was interrupted in several occasions by the people flow, customers willing to grab products from the shelves or to move around with their shopping carts.

Calibration data were off-line processed and stored in the database to enable the dynamic download of the algorithm parameters to the clients. Some network configurations were needed to set the final working system, due to the NSP (network security policy) in the supermarket. For testing, an application enabling path recording was used. The application requires that the tester identifies the zone in which she will be moving before starting the recording process. Then, a log file stores the RSS raw measurements and the location estimation results, with the timestamps. Testers were moving following a predefined path covering all the zones.

When in central positions, the obtained correct recognition rate (CRR) varies between 76% and 99%, depending on the zone (Fig. 3). The lowest CRR was obtained in zone H2, which was mainly confused with H1. The estimates were fairly stable, as it can be derived from the static deviation.

IV. OPTIMIZATION OF THE CALIBRATION PROCESS

The data collection for calibration was a time-consuming and uncomfortable phase which is not usually optimized for in-lab experiments. We next describe how to reduce the necessary time to calibrate the system.

A. Reducing the Number of Samples and the Sampling Positions

Key issues for calibration are the reference positions and the number of samples to take in each of them. In each reference position, the goal is to reduce the number of samples to a minimum that guarantees a

TABLE I
CLASSIFICATION OF CALIBRATION POSITION.

	Acc.	# measures	# positions	Necess. time
Type A	> 90%	20-40	25	500 s
Type B	> 90%	100-260	4	660 s
Type C	60% – 90%	120-260	6	1200 s
Type D	< 60%	120-260	5	950 s
Total				55 min

stable CRR. The stability requirement is relevant: the obtained CRR may be higher or lower, but it still needs to be converge towards a stable one in any case. To evaluate the influence of the number of samples, the set of samples taken during the test phase has been divided into groups of multiples of 20 measurements (20, 40, 60, ...). For each group, the thresholding conditions were calculated and finally we used the data obtained during the test as the input for the location algorithms. For the test, stable CRR meant that adding new samples, the CRR did not vary more than 3%. Analyzing the variation of the CRR in each zone, it was noted that 75% was reached with at most 40 samples, and that in 13 out of 40 positions the CRR was 90% with the same number of measures.

A classification of types for sampling position is gathered in TABLE I. Sampling positions of type A reach a CRR above 90% and achieve a stable CRR after a very low number of samples (20-40). B type-sampling positions also achieve very good CRR, but they need 100-260 calibration samples enter in the stable part of the curve. C type-sampling positions obtain a CCR between 60 and 90% with 120-260 samples and D type-sampling positions have a lower CRR with the same number of calibration samples. Fortunately, the most populated group is A-type sampling positions, gathering more than 62% of positions in the deployment. Type A positions are those located well inside the zones under analysis, while D positions are close to the zone boundaries. B and C positions are in the middle, between A and D, with B close to A and C close to D. As it is show in TABLE I, the prior classification of sampling positions may significantly reduce the necessary calibration time by more than 72%. In our case, the full calibration process could be completed by a single person in 55 minutes.

Regarding the number of calibration positions, we were willing to check the effect of removing some A-type and D-type points to reduce the calibration time even more. Results show that removing the A-type positions obtaining the lowest CRR have a general effect on the global CRR, which drastically decreases. Removing the A-type position with the highest CRR makes the CRR in zone to slightly improve. Removing the worst D-type position makes the CRR for that zone decrease. Removing the best D-type point, the CRR for

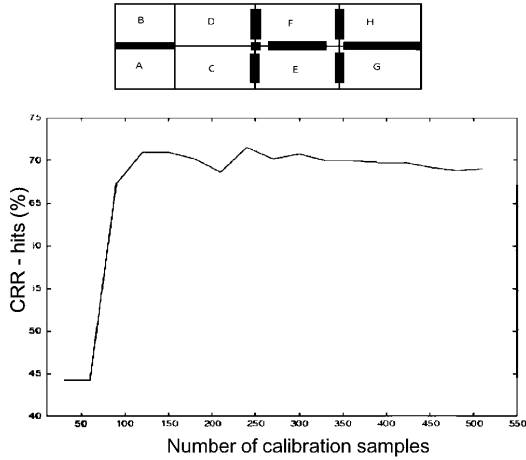


Fig. 4. a) Lab environment. b) Evolution of the Correct Recognition Rate in zone B with the strategy on-the-move.

TABLE II
COMPARISON OF STATIC AND CALIBRATION ON-THE-MOVE STRATEGIES.

Zone	CRR (%) for static cal.	CRR (%) for cal. on-the-move
A	63,11	61,59
B	61,73	70,95
C	56,24	58,67
D	27,57	82,26
E	30,25	63,15
F	75,43	46,59
G	36,63	53,21
H	41,73	72,71

that zone slightly improves. These results show that it is probably not worthy to remove positions to reduce the calibration time. The positions in which the lowest CRR are obtained provides important information to the system: removing them, not only deletes that information from the parameters, but it also gives more weight to the rest of points that do not provide that kind of information. This test suggests that it is worthy to have enough spatial diversity, while adjusting the number of samples in each calibration position may be helpful to reduce the calibration time.

B. Introducing Calibration On-the-move

The spatial diversity requirement suggests that an alternative to static calibration could be to take RSS samples when walking around within the areas where users will be wandering, depending on the space configuration and user's practices. So after the in-shop deployment, we performed additional tests in our lab to compare the static calibration procedure with a strategy on-the-move. The lab space is divided into 8 different non-walled zones (Fig. 4a). We took 10-minutes walks of samples within each zone carrying two smartphones simultaneously, so 1000 samples were gathered for each area. When analyzing the effect of using an increasing

number of samples for calibration on-the-move, it can be seen that the CRR converges to a value when a number of samples are considered; for example, in Fig. 4b, convergence is appreciated from around 120 samples on. A similar behavior has been obtained for the rest of zones. TABLE II shows the CRR for each calibration technique in the different zones, while using a) 6 calibration positions and 150 samples per position per zone for static calibration, b) 150 samples of the 10-minutes walk (randomly sampled from the trajectory) and c) 500 samples for evaluation both for static and strategies on-the-move. Results with 25 samples per position for static calibration has also been computed (150 samples per zone), providing very similar results to the ones shown in TABLE II. From these data, it can be inferred that results with calibration on-the-move are better or at least similar to those obtained with the static procedure for almost all the zones under analysis, with the exception of zone F. This can be due to the fact that the walk was not including enough spatial diversity. This experiment shows that calibration on-the-move may achieve satisfactory results, reducing the calibration time to 2.5 minutes in this case (as CRR is not getting much better with additional samples).

V. CONCLUSIONS

Many RSS-based indoor location algorithms need to be tuned to the specific physical and electromagnetic characteristics of the space through calibration. The calibration procedure is usually tedious and may consume a good part of the deployment time. For this reason, any mechanism that may alleviate the task is welcome. In this paper, we have presented our experience during a real deployment and shown how to deal with calibration to optimize it, for the needs of our zone-based location algorithm. The optimization of the number of samples to be gathered on each position may significantly reduce the calibration time. Additionally, the correct choice of the calibration reference positions to guarantee spatial diversity may help to get better performance in terms of correct recognition rate.

We have also described and shown the potential of calibration on-the-move which, from our experiments in the lab, may achieve better CRR than motionless approaches while optimizing time and calibration dynamics (it is always easier for a tester to take measurements in not so restrictive conditions). Calibration on-the-move enables to include a increased spatial diversity adapted to the service needs (i.e. particularly tuned for the areas where the user is really moving around in the space). It also makes possible to design strategies for permanent updating of the calibration, as any user may be gathering measurements that may be used for this purposes. We are already working on better defining how to perform calibration on-the-move for continuous update of the algorithm's parameters (following the line

of previous works such as [11]). Finally, the deployment experience has demonstrated the need to define a much more guided calibration methodology, as it is evident that it is necessary to avoid unnecessary deployment delays and even the presence of technicians in the deployment process. If sufficiently standardized, any location system administrator may perform its own calibration process, so we are working on tools to achieve this goal.

ACKNOWLEDGMENTS

This work has been supported by the Spanish Ministry of Economy and Competitiveness under grant TEC2014-55146-R and by UPM under grant RP150955017.

REFERENCES

- [1] A. Yaeli, P. Bak, G. Feigenblat, S. Nadler, H. Roitman, G. Saadoun, H. J. Ship, D. Cohen, O. Fuchs, S. Ofek-Koifman, and T. Sandbank, "Understanding customer behavior using indoor location analysis and visualization," *IBM Journal of Research and Development*, vol. 58, no. 5/6, pp. 3:1–3:12, Sept 2014.
- [2] Apple Inc., "iBeacon for Developers," 2013. [Online]. Available: <https://developer.apple.com/ibeacon/>
- [3] Google Inc., "Google Beacons," 2015. [Online]. Available: <https://developers.google.com/beacons/>
- [4] AltBeacon, "AltBeacon. The Open and Interoperable Proximity Beacon Specification," 2014. [Online]. Available: <http://altbeacon.org/>
- [5] P. Tarrío, A. M. Bernardos, and J. R. Casar, "Weighted least squares techniques for improved received signal strength based localization," *Sensors*, vol. 11, no. 9, pp. 8569–8592, 2011. [Online]. Available: <http://www.mdpi.com/1424-8220/11/9/8569>
- [6] R. Faragher and R. Harle, "An analysis of the accuracy of bluetooth low energy for indoor positioning applications," in *Proceedings of the 27th International Technical Meeting of the Satellite Division of the Institute of Navigation (ION GNSS+ '14)*, 2014, pp. 201–210.
- [7] Z. Jianyong, L. Haiyong, C. Zili, and L. Zhaohui, "Rssi based bluetooth low energy indoor positioning," in *2014 Intl. Conference on Indoor Positioning and Indoor Navigation (IPIN)*, Oct 2014, pp. 526–533.
- [8] D. Lymberopoulos, J. Liu, X. Yang, R. R. Choudhury, V. Handziski, and S. Sen, "A realistic evaluation and comparison of indoor location technologies: Experiences and lessons learned," in *Proceedings of the 14th International Conference on Information Processing in Sensor Networks*, ser. IPSN '15. New York, NY, USA: ACM, 2015, pp. 178–189. [Online]. Available: <http://doi.acm.org/10.1145/2737095.2737726>
- [9] S. He and S. H. G. Chan, "Wi-fi fingerprint-based indoor positioning: Recent advances and comparisons," *IEEE Communications Surveys Tutorials*, vol. 18, no. 1, pp. 466–490, Firstquarter 2016.
- [10] E. Metola, "Contributions to Beacon-Based Applications in Smart Spaces," Ph.D. dissertation, Universidad Politécnica de Madrid, Signal, Systems and Radiocommunications Department, Madrid, Spain, 2017.
- [11] A. M. Bernardos, J. R. Casar, and P. Tarrío, "Real time calibration for rss indoor positioning systems," in *Indoor Positioning and Indoor Navigation (IPIN), 2010 Intl. Conference on*. IEEE, 2010, pp. 1–7.