ELSEVIER
Available online at www.sciencedirect.com
SciVerse ScienceDirect
Procedia Computer Science 10 (2012) 1106 – 1113
Procedia Computer Science

International Workshop on Cooperative Robots and Sensor Networks (RoboSense)

Simultaneous RSS-based Localization and Model Calibration in Wireless Networks With a Mobile Robot

J.Vallet, O.Kaltiokallio, M.Myrsky, J.Saarinen, M.Bocca

Department of Automation and Systems Technology
Aalto University School of Electrical Engineering
Otaniementie 17, Espoo, 02150, Finland

Abstract

This paper presents a recursive expectation maximization-like algorithm that can be used to simultaneously locate the nodes of a wireless network and calibrate the parameters of received signal strength vs. distance models. The algorithm fine tunes one model for each node accounting for its local environment and small hardware differences with respect other nodes. In contrast with using a common model for all the nodes, it is not required to artificially inflate the standard deviation of the random variable accounting for uncertainties in order to accommodate differences of signal strength measurements from different nodes. As a consequence, the position estimate is more accurate. We conducted a series of experiments in which a mobile robot with known location was used as a mobile beacon in three environments with different propagation characteristics. The results show a significant decrease of the mean error of the position estimates in all environments when using individual models compared to using a common one. Using a model with a third order polynomial and a mixture of two Gaussians, the algorithm was able to locate the nodes within a meter on average in an office and with less than half a meter in more open environments. The estimated potential accuracy is about half a meter in all the environments.

Keywords: RSS, Localization, Automatic Model Calibration, Mobile Robot, Wireless Networks

1. Introduction

The problem of localizing the nodes of a wireless network can be addressed from different perspectives [1]. Very often it is assumed that the location of some of the nodes is known. These nodes can then be used as beacons to localize the other nodes using distance estimates from received signal strength (RSS) measurements. In this paper we take a different approach: we assume that the position of all the nodes is unknown and use a mobile robot capable of simultaneous localization and lapping (SLAM) as a mobile beacon. While the robot moves around, it builds a map of the environment using a laser scanner and odometry information. Thus, its position within the map is known at any moment. As the robot moves, it also collects RSS measurements from the nodes of the network. All this information is then exploited to estimate the position of the nodes.

Another usual practice is to calibrate and use a site-specific model common to all the nodes deployed in a similar environment. The advantage of this approach is that only a few model parameters have to be estimated for all the nodes. In case of the popular log-normal model, these are the absolute mean path loss, the path loss exponent and the standard deviation of the difference between measured and predicted observations [2]. However, a correct and careful calibration procedure requires taking measurements with different nodes placed in many different positions, which

doi:10.1016/j.procs.2012.06.157

is not always feasible. In any case, the values of the model parameters are valid only for the specific environment where the model was calibrated. In our algorithm, each node uses its own site-specific model. The advantage is that individual models can explain better the node's particularities, such as its close environment and small hardware differences among nodes. Thus, as we will show, the localization error decreases significantly.

Finally, many localization algorithms assume that the model parameters are known before the deployment. However, in many real scenarios this prior information is not available. It would be then very beneficial to have localization algorithms that can automatically calibrate the models. The work presented in this paper is a step in this direction. We propose a simple offline iterative algorithm that can simultaneously estimate the position of the nodes and calibrate their individual model parameters using the data provided by a mobile beacon. The localization algorithm is tested in three different environments: a) a large indoor sports hall (basketball field) with clear line of sight among the nodes, b) a partially obstructed lobby of a building, and c) an office environment in which the nodes were located inside rooms and in the corridors. Based on our observations, the model chosen for the RSS is a third degree polynomial with a Mixture of Gaussians (MoG) accounting for the uncertainty. The estimated positions are then compared with the real positions of the nodes. It is shown that the algorithm can potentially reach close to half a meter accuracy in all our tests and environments. With the actual implementation, it was capable of estimating the position of the nodes with an average error of about a meter in the office environment and less than half a meter in the basketball field.

1.1. Related Work

One of the first attempts of using a robot to locate the nodes of a wireless network can be found in [3]. The authors use a remotely controlled truck equipped with a GPS to move a hand-held device outdoors that broadcasts messages containing information about the location of the truck. The nodes receiving these messages construct probabilistic constraints based on the location of the beacon and distance estimates from RSS. The reported localization error was less than three meters.

In [4], a mobile robot is used to estimate the location of the nodes of a network organized in clusters in a large indoor area. The authors use a logarithmic RSS-distance model with a random variable with unknown probability distribution function (PDF). The model is also calibrated before the experiments. Because the PDF of the noise is not known, a robust extended Kalman filter is used to estimate the position of the nodes. The reported localization accuracy is about one meter.

In [5], a robot equipped with DGPS moves in an outdoor parking lot measuring RSS from neighboring nodes. The localization algorithm applies at first a particle filter for an initial approximation of the position of the nodes using measurements from the robot, and then an inverse filter for refining the estimates using measurements among the nodes. The likelihood of the measurements is modeled as a Gaussian whose mean and standard deviation increase with the distance. The model is calibrated before the experiments and, to account for the differences among the nodes, the authors intentionally inflate the standard deviation. A similar particle filter and model for the RSS are used in [6], but this time a helicopter is used as a mobile beacon to estimate the position of the nodes in 3D.

In [7], an extended Kalman filter is used to simultaneously estimate the position of a robot and mapping the nodes using distance estimates from RSS and odometry from the robot. The authors use a log-normal RSS-distance model, which is first calibrated before the experiments and then used to calculate the inverse distance-RSS model. The tests were carried out indoors in a large corridor, and the reported accuracy is less than a meter if no prior information about the location of the nodes is available, and less than half meter with priors close to the true positions.

The approach in [8] uses a MoG defined over the angle of arrival of the signal to represent the likelihood of distance estimates from RSS measurements as in [6]. This model represents a multiple hypothesis over the angle of arrival of the signal from the robot to the node. Upon arrival of the first RSS measurement, the Gaussian components are equally spaced covering the whole angle. As the robot moves new RSS measurements are integrated using a Kalman filter. The parameters of the Gaussians and the mixture coefficients are then updated according to the likelihood of the new measurements and Gaussians with poor weights are pruned.

All the approaches reviewed so far have three characteristics in common: a) they all use one RSS-distance model common to all the nodes, b) the parameters of the model are calibrated before the real experiments take place and c) all the experimental tests are done in similar environments. The algorithm presented in this paper differs from the previous ones in all these three aspects. It simultaneously locates the nodes and estimates the model parameters, and thus can potentially be used in any environment with no prior calibration. In addition, it uses an individual model for each node which the algorithm fine tunes for the local environment of each particular node. As a consequence, the algorithm can make more accurate position estimates, as confirmed in tests carried out in different types of environments.

2. Experimental setup

The nodes used are equipped with TI CC2420 802.15.4 compliant radios (2.4 GHz) and omnidirectional antennas. The nodes were mounted on top of one meter high poles thick enough so that they could be detected by a laser scanner. The robot used was a typical differential driven research platform equipped with a laser scanner and encoders, and capable of SLAM. One node was mounted on top of the robot to collect about 10 RSS measurements per second from each of the rest of the nodes. Each node has a unique id that is used to uniquely identify the origin of the packets, and thus there is no correspondence problem. The data collected were forwarded to the robot and then stored for off line processing. As the poles of the nodes are visible to the laser scanner, the maps generated by the robot show the poles of the nodes as any other obstacle. The recognition of the poles and the estimation of their position was done manually by a human straight from the map. These coordinates were then taken to be the true positions of the nodes, with an estimated accuracy in the order of centimeters. This error is thus negligible with respect RSS based position estimate errors.

In the experiments we used 18 to 20 nodes deployed in random fixed positions in three environments with different propagation characteristics: a) a large indoor sports hall (basketball field) with clear line of sight among the nodes, b) a partially-obstructed lobby of a building, and c) an office area in which the nodes were located inside rooms and in the corridors. In each environment we performed three tests in which the robot explored the area following different trajectories. Each test lasted about 7 minutes. The maps from different tests for the same environment were matched as in [9] to obtain a global localization frame. The resulting maps with the true position of the nodes and a sample trajectory of the robot can be seen in Figure 2.

The RSS measurements were filtered by averaging them every 25 cm traveled by the robot. This filtering decreased the effects of fast fading, made the rate of effective measurements independent of the speed of the robot, and lowered the computational complexity of the problem, as it reduced the size of the data to process.

3. RSS-Distance Model

In order to decide what model to use, we made a preliminary study of the RSS-distance plots using the real distances between the nodes and the robot. These distances were calculated using the true position of the nodes estimated as explained in section 2 and the position of the robot for each measurement. At first we tried the log-normal model from [2], which was observed to fit well to the observations from the the basketball field. However, in some cases it was not able to explain properly the measurements from nodes in the other two more cluttered environments. Figure 1 shows examples of sets of observations from two nodes located in the lobby and the office. These are representative examples from which we can see typical tendencies of the RSS observed in cases in which the log-normal model is not suitable. The left figure shows measurements from a node in the lobby. We can see that the curve is roughly divided in three parts: one with a rapid decrease of the RSS for short distances followed by another in the middle with more constant values (sometimes even increasing), and the last one with the final decay for the larger distances. The log-normal model is constantly decaying and cannot model this properly. The figure in the right shows a case of a node located in the office. The plot is in logarithmic scale, in which log-normally distributed data should follow a straight line. As we can see, the rate of decay of the RSS is larger as the distance increases. Based on these observations it was decided to use polynomial models of third degree, which are more flexible and can cope with these observations. The figures described previously show the fit of third degree polynomials in the distance and logarithm of the distance respectively. In the lower parts of the figures we can see the histograms of the residuals. These are also poorly modeled with Gaussians, which the log-normal model uses. It was decided then to use MoGs with two components, which can accommodate well skewed, multimodal and heavy tailed residuals.

For a generic polynomial of degree m and vector of coefficients $\boldsymbol{p} = [p_1, p_2, \ldots, p_m, p_{m+1}]^t$ and for a MoG with K components, the model for the observed RSS z for a given distance d can be written as

$$z = p_1\rho^m + p_2\rho^{m-1} + ... + p_m\rho + p_{m+1} + \sum_{k=1}^{K} m_k\zeta_k = \boldsymbol{D}\boldsymbol{p} + \sum_{k=1}^{K} m_k\zeta_k \tag{1}$$

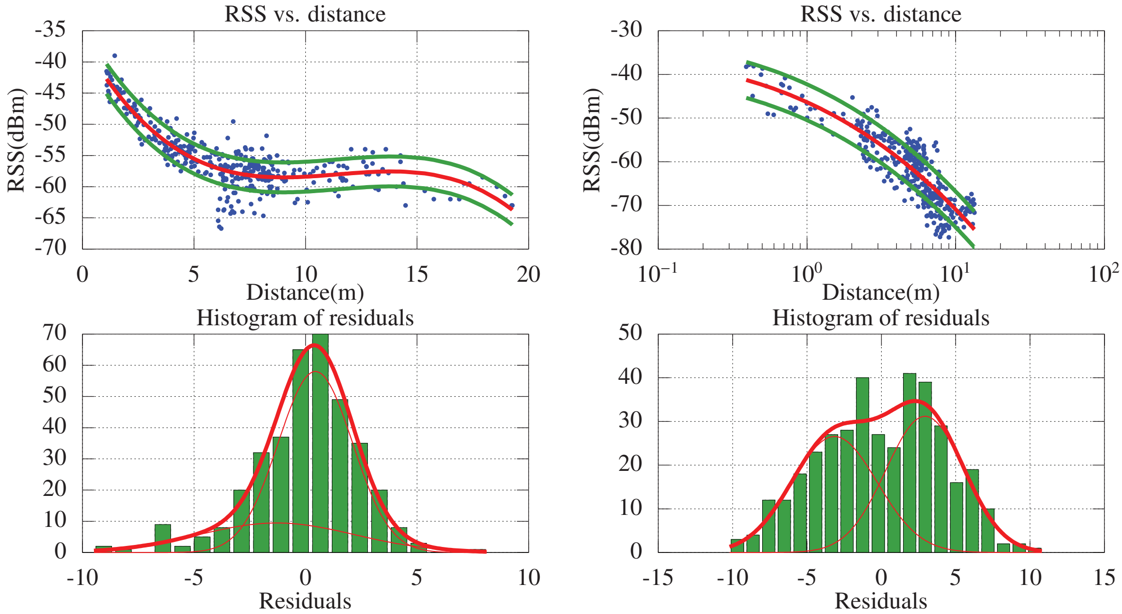


Figure 1: Example of 3^{rd} degree polynomial models with MoG for nodes. In the left fit of a P3M2N model for a node in the lobby. In the right fit of a P3LNM2N model for a node in the office. The red and green lines in the upper plots represent the mean and one standard deviation with respect to the mean respectively. In the lower plots the pdf of the MoGs is represented by the thick red lines, whereas the thin ones represent the weighted component distributions.

where $\boldsymbol{D} = [\rho^m, \rho^{m-1}, \ldots, \rho, 1]$, and $\rho = d$ or $\rho = log(d)$ depending on whether the polynomials depend on the distance or the logarithm of the distance respectively. The MoG is characterized by the weighted sum of K normally distributed random variables $\zeta_k \sim N(\mu_k, \sigma_k)$, with mixture coefficients m_k restricted to $\sum_{k=1}^{K} m_k = 1$. The log-normal model is thus a particular case for which $\rho = log(d)$, $m = 1$ and $K = 1$.

The goodness of the models can be compared using the likelihood of the data. We consider, however, that a more meaningful comparison measure in the context of this research is to use the real error of maximum likelihood (ML) position estimates, that is: the models are used to build ML estimators for the position of the nodes, and the resulting estimates are compared against the true positions. In this paper we compare several models using this criterion, including the classic log-normal. In order to assess the potential gain of using an individual model for each node with respect using a common one for all the nodes, we also extracted common models using all the measurements from all the nodes and all the tests for each environment. Table 1 shows averaged values of the mean error of the ML estimates for all the models considered. The notation is as follows: models depending on the distance are denoted for example P3M2N, to be read as *polynomial of* 3^{rd} *degree with a mixture of 2 normal components*. Models depending on the logarithm of the distance are denoted for example P1LM2N, to be read as *polynomial of* 1^{st} *degree in the log-distance with a mixture of 2 normal components*. The log-normal model is thus denoted P1LM1N, as is a 1^{st} order polynomial that depends on $log(d)$ and with a single normal random variable.

As we can see, the use of individual models instead of a common one improves systematically the localization accuracy. In the office environment, i.e. the most obstructed, the accuracy increases approximately by 30% when using log-normal models. This percentage increases close to 70% when using a third degree polynomial (P3LM1N), with which the mean error is about 60cm. In the basketball field and lobby the mean error decreases by a factor larger than 2 in all cases.

In the basketball field (close to open environment) the use of logarithmic individual models is generally better than third order polynomials in the distance. The increase of degree of the polynomial from log-normal to third degree (P1LM1N to P3LM1N) does not increase significantly the accuracy, which is consistent with the hypothesis that the

Table 1: Comparison of average errors of ML position estimates (distance in meters) when using different RSS-distance models in different environments. CM=common model, IM=individual model.

	P1LM1N			**P1LM2N**		
	CM	IM	Ratio(CM/IM)	CM	IM	Ratio(CM/IM)
Basketball field	1.00	0.37	2.74	1.00	0.34	2.97
Lobby	1.46	0.64	2.30	1.39	0.54	2.54
Office	0.99	0.78	1.28	1.01	0.69	1.46
	P3LM1N			**P3LM2N**		
	CM	IM	Ratio(CM/IM)	CM	IM	Ratio(CM/IM)
Basketball field	0.99	0.36	2.73	0.87	0.33	2.66
Lobby	1.46	0.68	2.14	1.36	0.53	2.57
Office	0.98	0.59	1.67	0.99	0.58	1.69
	P3M1N			**P3M2N**		
	CM	IM	Ratio(CM/IM)	CM	IM	Ratio(CM/IM)
Basketball field	0.88	0.41	2.14	0.83	0.40	2.08
Lobby	1.53	0.70	2.19	1.34	0.57	2.32
Office	1.02	0.59	1.73	1.03	0.58	1.78

log-normal model is well suited for open environments. The benefits can be noticed in the office, where the deviations from the log-normal model are larger.

Finally, the use of MoGs generally decreases the mean error when compared to using a single normal, as they can better approximate the non Gaussian distribution of the observed residuals. The use of third order polynomial models and MoGs is thus justified based on their flexibility and greater ability to model observations from different environments when compared to the log-normal model. The drawback is the increased complexity and number of parameters required.

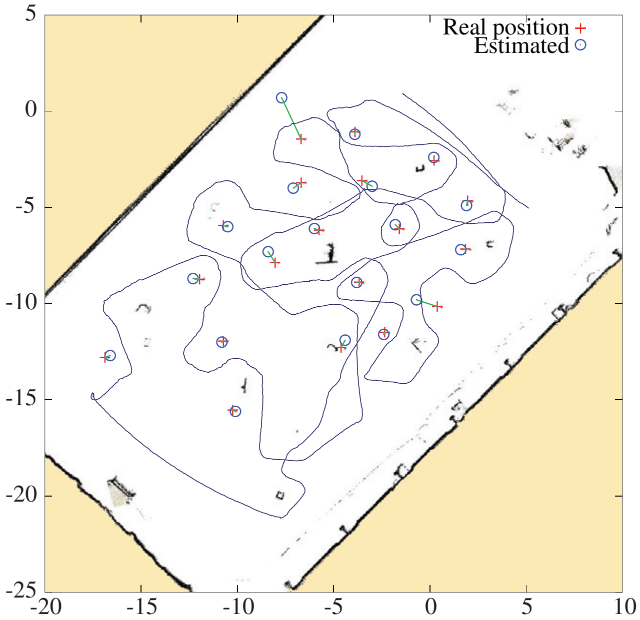
4. Simultaneous Localization and Model Calibration

The proposed algorithm is a recursion consisting of two steps: position estimation and model extraction. It starts with an initial guess of the values of the model parameters, which are refined in each iteration together with the position of the nodes in the direction of increasing the likelihood of the data. The robot acts as a mobile beacon which provides information only about its own position within the reference frame of its own map and the RSS measurements. The laser scanner is then used only for SLAM of the robot itself, and not for direct localization of the nodes.

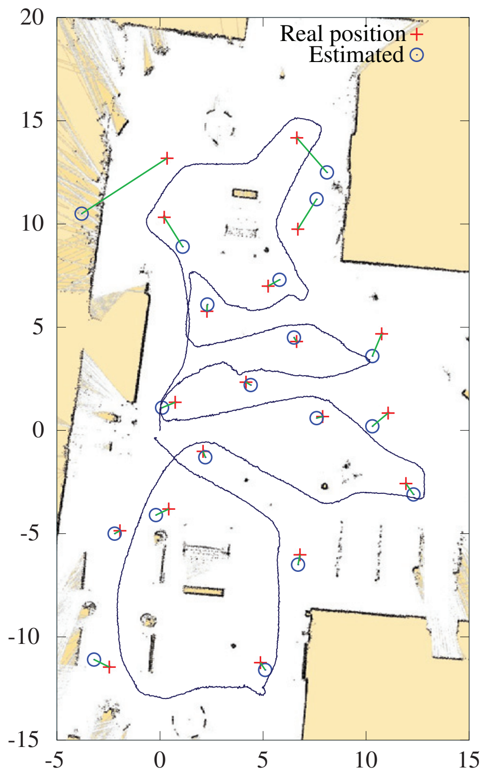
Consider now a robot exploring an area and measuring RSS from only one of the nodes in N points of its trajectory. A complete trajectory contains then N measurement pairs $(z_i, \vec{r_i})$, where z_i is the observed RSS and $\vec{r_i} = [r_{x_i}, r_{y_i}]$ is the position of the robot at the i-th measurement ($i = 1 : N$). Let us also denote $\vec{x} = [x, y]$ the (fixed) position of the node and $d_i = d_i(\vec{x}, \vec{r_i}) = \left\|\vec{x} - \vec{r_i}\right\|_2$ the distance between the robot and the node. Using the generic polynomial model of equation 1, the likelihood of one observation z_i can be written as

$$p_Z(z_i; d_i(\vec{x}, \vec{r_i}), \boldsymbol{\theta}) = \sum_{k=1}^{K} m_k \frac{1}{\sqrt{2\pi\sigma_k^2}} exp\left[-\frac{(z_i - (\boldsymbol{D}_i\boldsymbol{p} + \mu_k))^2}{2\sigma_k^2}\right] \quad / \quad \sum_{k=1}^{K} m_k = 1 \tag{2}$$

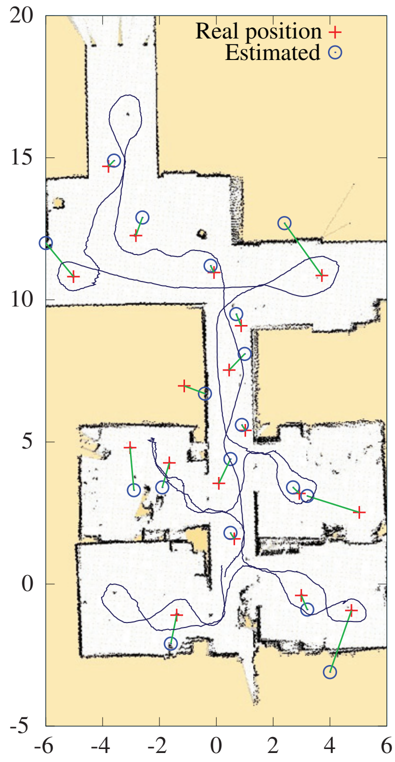
The model parameters are then $\boldsymbol{\theta} = [\boldsymbol{p}^t, \boldsymbol{m}, \boldsymbol{\mu}, \boldsymbol{\sigma}]$, with $\boldsymbol{m} = [m_1, \ldots, m_K]$, $\boldsymbol{\mu} = [\mu_1, \ldots, \mu_K]$ and $\boldsymbol{\sigma} = [\sigma_1, \ldots, \sigma_K]$.



(a) Basketball field



(b) Lobby



(c) Office

Figure 2: Maps generated by the robot, real position of the nodes and position estimates using the algorithm of Table 2 with P3M2N models to sample robot trajectories in three different environments. The trajectories of the robot are represented as dark blue continuous lines. Coordinates are in meters.

Table 2: Algorithm for simultaneous localization and model calibration using a mobile beacon.

```
1 locate_and_calibrate_node(z, r, θ_init)
2   θ = θ_init
3   repeat
4     x⃗=locate_node_ML(z, r, θ)
5     d=calculate_robot_node_distances(x⃗, r)
6     θ=extract_rss_model(z, d)
7   until convergence
8   return x⃗, θ
```

In the position estimation step we assume that $\boldsymbol{\theta}$ is known. Assuming independence among observations the likelihood of all the observations is the product of the likelihoods of individual measurements, and thus the ML estimate of the node position is

$$\hat{\vec{x}}_{MLE} = \underset{\vec{x}}{\operatorname{argmax}}\, L(\boldsymbol{z}; \boldsymbol{d}(\vec{x}, \mathbf{r}), \boldsymbol{\theta}) = \underset{\vec{x}}{\operatorname{argmax}} \sum_{i=1}^{N} ln p_Z(z_i; d_i(\vec{x}, \vec{r_i}), \boldsymbol{\theta}) \tag{3}$$

where $\boldsymbol{z} = [z_1, \ldots, z_N]$, $\boldsymbol{r} = [\vec{r_1}, \vec{r_2}, \ldots, \vec{r_N}]$ and $d = [d_1, \ldots, d_N]$. The position estimation step is thus a standard maximization problem over the coordinates of the node that can be solved using different techniques. In this paper we use a grid search method with a 0.1 m grid resolution. One advantage of the ML formulation of the localization problem is that it does not require to calculate the inverse model of the RSS-distance, which can be difficult. In particular, if the RSS-distance mapping is not bijective, the inverse model (distance-RSS) might contain several distances for the same RSS values. This can be a serious drawback for some algorithms that require a direct estimate of the distance from RSS, and requires additional work to choose between the possible alternatives. The ML formulation of the problem simply does not suffer from this problem, and it can work with any function of the distance, as long as the model is a valid PDF.

Using the position calculated in the position estimation step we can compute estimates of the distances between the node and the robot for all the measurement points. The model extraction step uses then these distances and the set of RSS measurements to calculate a new estimate of the model parameters. To extract the model we estimate first $\boldsymbol{p}$ using a polynomial regression and then $\boldsymbol{m}$, $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}$ fitting the MoG to the residuals using expectation maximization (EM) [10]. In the case of single normal, this simplifies to a normal regression. Finally, the complete algorithm for simultaneous localization and model calibration is schematically shown in Table 2.

This procedure is applied to each node independently to estimate its location and calibrate its individual model. The arguments are the set of observations $\boldsymbol{z}$, the trajectory of the robot $\boldsymbol{r}$ and an initial guess of the model parameters $\boldsymbol{\theta_{init}}$. The position estimation step takes place in line 4, and the model extraction in line 6. When the algorithm enters line 4 for the first time, it calculates the position of the node using ML and $\boldsymbol{\theta_{init}}$ as explained before. In line 5, this position together with the trajectory of the robot are used to calculate the node-robot distances for all the measurement points. Then, in line 6 these distances and the measurements $\boldsymbol{z}$ are used to recalculate the model parameters $\boldsymbol{\theta}$ using EM. This iteration is repeated until successive increments of the likelihood are smaller than a small threshold ϵ.

Table 3 contains the numerical results of using the localization algorithm of Table 2 with log-normal and P3M2N models. The results are averaged over all the tests and nodes. Figure 2 shows the results of a sample individual tests using P3M2N models in the three different environments. Results for P3LM2N are similar to those for P3M2N, and are omitted for brevity. The initial guesses $\boldsymbol{\theta_{init}}$ used were the same for all the nodes within the same environment. The values were the median of the parameters from the individual models extracted for all the nodes using the true position of the nodes. While for most of the nodes the error is small (mean and median below 1 m), for some others it can be significantly larger (maximums above 4 meters), increasing the average mean error. This is due to the sensitivity of the algorithm to the initial values of the parameters, which can get trapped in local maxima of the likelihood. When comparing values for individual models from table 1 and values from table 3, recall that former ones were calculated using ML and the model parameters calibrated using the true position of the nodes. Thus they can be seen as lower bounds for the averaged mean errors of the algorithm for simultaneous node localization and model calibration.

Table 3: Average errors of the position estimates calculated with the algorithm from Table 2 in different environments using log-normal and P3M2N models.

	Min		Mean		Median		Max	
	P1LM1N	**P3M2N**	**P1LM1N**	**P3M2N**	**P1LM1N**	**P3M2N**	**P1LM1N**	**P3M2N**
Basketball field	0.05	0.09	0.64	0.47	0.3	0.32	4.84	2.44
Lobby	0.11	0.12	1.27	1.06	0.79	0.59	6.17	5.04
Office	0.16	0.16	1.18	0.97	0.93	0.76	2.72	2.32

5. Conclusions And Future Work

This paper presents a recursive EM-like algorithm that simultaneously locates the nodes of a wireless network and estimates the RSS-distance model parameters without requiring an a-priori, site-specific calibration. For each node, the algorithm uses an individual model in order to better fit the characteristics of the node's local environment and hardware. This increases significantly the accuracy of the position estimates when compared to using a unique common model to all the nodes. The use of a third order polynomial models with a mixture of two Gaussians further improves the position estimates compared to using a simple log-normal model with a single Gaussian. While the third order polynomial can be shaped to more generic curves, the MoG can better represent skewed, multimodal and heavy tailed residuals.

Tests were performed in three different indoor environments by using fixed nodes and a mobile robot serving as a mobile beacon. The experimental results show that, using the model with the third order polynomial for each node, the mean error of the ML position estimates can potentially decrease to about half a meter in all the environments. These results can be thought of as a lower bound on the performance of the proposed algorithm, as they were obtained using individual models tuned knowing the real nodes-robot distances. Using as an initial guess of model parameters the median values of parameters from models extracted from the experimental data, the mean error was less than a meter in all environments. This difference of performance is explained by the sensitivity of the algorithm to the initialization of the model parameters. In future work, we plan to find more robust initialization procedures and implement an online version of the algorithm.

[1] J. Bachrach, C. Taylor, Localization in sensor networks, in: I. Stojmenovic (Ed.), Handbook of Sensor Networks: Algorithms and Architectures, Wiley Series on Parallel and Distributed Computing, 2005, Ch. 9.

[2] S. Seidel, T. Rappaport, 914 mhz path loss prediction models for indoor wireless communications in multifloored buildings, Antennas and Propagation, IEEE Transactions on 40 (2) (1992) 207–217. doi:10.1109/8.127405.

[3] M. Sichitiu, V. Ramadurai, Localization of wireless sensor networks with a mobile beacon, in: Mobile Ad-hoc and Sensor Systems, 2004 IEEE International Conference on, 2004, pp. 174 – 183. doi:10.1109/MAHSS.2004.1392104.

[4] P. N. Pathirana, N. Bulusu, A. V. Savkin, S. Jha, Node localization using mobile robots in delay-tolerant sensor networks 4 (3) (2005) 285–296. doi:10.1109/TMC.2005.43.

[5] F. Caballero, L. Merino, P. Gil, I. Maza, A. Ollero, A probabilistic framework for entire wsn localization using a mobile robot, Robot. Auton. Syst. 56 (2008) 798–806. doi:10.1016/j.robot.2008.06.003.

[6] F. Caballero, L. Merino, I. Maza, A. Ollero, A particle filtering method for wireless sensor network localization with an aerial robot beacon, in: Proc. IEEE Int. Conf. Robotics and Automation ICRA 2008, 2008, pp. 596–601. doi:.2008.4543271.

[7] E. Menegatti, A. Zanella, S. Zilli, F. Zorzi, E. Pagello, Range-only slam with a mobile robot and a wireless sensor networks, in: Proc. IEEE Int. Conf. Robotics and Automation ICRA '09, 2009, pp. 8–14. doi:.2009.5152449.

[8] F. Caballero, L. Merino, A. Ollero, A general gaussian-mixture approach for range-only mapping using multiple hypotheses, in: Proc. IEEE Int Robotics and Automation (ICRA) Conf, 2010, pp. 4404–4409. doi:.2010.5509704.

[9] J. Saarinen, J. Paanajrvi, P. Forsman, Best-first branch and bound search method for map based localization, in: Proceedings of the IEEE/RJS International Conference on Intelligent Robots and Systems 2011, San Francisco, 2011, pp. 59–64, iEEE/RJS International Conference on Intelligent Robots and Systems, San Francisco, Sep 25-30, 2011.

[10] A. P. Dempster, N. M. Laird, D. B. Rubin, Maximum likelihood from incomplete data via the em algorithm, Journal of the Royal Statistical Society, Series B 39 (1) (1977) 1–38.