An Experimental Study of RSS-Based Indoor Localization Using Nonparametric Belief Propagation Based on Spanning Trees

Vladimir Savic, Adrian Poblacion, Santiago Zazo and Mariano Garcia

Post Print

N.B.: When citing this work, cite the original article.

©2010 IEEE. Personal use of this material is permitted. However, permission to reprint/republish this material for advertising or promotional purposes or for creating new collective works for resale or redistribution to servers or lists, or to reuse any copyrighted component of this work in other works must be obtained from the IEEE.


http://dx.doi.org/10.1109/SENSORCOMM.2010.44

Postprint available at: Linköping University Electronic Press

http://urn.kb.se/resolve?urn=urn:nbn:se:liu:diva-81370
An Experimental Study of RSS-based Indoor Localization using Nonparametric Belief Propagation based on Spanning Trees

Vladimir Savic, Adrián Población, Santiago Zazo and Mariano García
Signal Processing Applications Group, Polytechnic University of Madrid, Spain
{vladimir, santiago, mariano}@gaps.ssr.upm.es, adrian.poblacion.hernandez@alumnos.upm.es

Abstract—Nonparametric belief propagation (NBP) is the well-known method for cooperative localization in wireless sensor networks. It is capable to provide information about location estimation with appropriate uncertainty and to accommodate non-Gaussian distance measurement errors. However, the accuracy of NBP is questionable in loopy networks. Therefore, in this paper, we propose a novel approach, NBP based on spanning trees (NBP-ST) created by breadth first search (BFS) method. In addition, we propose a reliable indoor model based on obtained received-signal-strength (RSS) measurements in our lab. According to our experimental results, NBP-ST performs better than NBP in terms of accuracy and communication cost in the networks with high connectivity (i.e., highly loopy networks).

Keywords—Indoor localization; nonparametric belief propagation; spanning tree; breadth first search; sensor networks.

I. INTRODUCTION

The belief propagation (BP) algorithm, proposed by Pearl [1], is a way of organizing the global computation of marginal beliefs in terms of smaller local computations within the graph. It is one of the best-known graphical model for distributed inference in statistical physics, artificial intelligence, computer vision, error-correcting codes, positioning, etc. The whole computation takes a time proportional to the number of links in the graph, which is significantly less than the exponentially large time that would be required to compute marginal probabilities naively.

Due to the presence of nonlinear relationships and highly non-Gaussian uncertainties, the standard BP is undesirable. In addition, in order to obtain acceptable spatial resolution for the sensors, the discrete space (grid) in the deployment area must be made too large for BP to be computationally feasible. However, particle-based approximation via nonparametric belief propagation (NBP), proposed by Ihler et al. [2], makes BP acceptable for inference in sensor networks. The main features of this approach are easy implementation in a distributed fashion and sufficiency of a small number of iterations to converge. Furthermore, NBP is capable to provide information about location uncertainties and to accommodate non-Gaussian measurement errors. This is the main advantage of NBP comparing with well-known deterministic methods, e.g., [3], [4]. In our application (indoor positioning), the distance error model is not even close to Gaussian model, thus this is our motivation for choosing NBP.

However, BP convergence is not guaranteed in a network with loops [1], [5] or even with convergence, it could provide us less accurate estimates. Regarding localization using NBP, the convergence is usually sufficient, but the accuracy is questionable. In the current state of the art, there are few solutions for networks with loops, but mostly they have not been used for the localization. The well-known solutions based on generalized belief propagation (GBP) [5]–[7], which are based on clusters or cliques, are still very complex for the large-scale ad-hoc/sensor networks. Therefore, in this paper, we propose NBP based on spanning trees (NBP-ST) created by breadth first search (BFS) method [8] which is optimal for the unweighted graphs. NBP-ST algorithm represents two (or more) independent runnings of the NBP algorithm based on formed spanning trees. In order to obtain realistic distance measurements for indoor scenario, we performed experiments in our lab using IRIS wireless sensor nodes equipped with AT86RF230 transceiver [9], [10]. According to our experimental results, NBP-ST performs better than NBP in terms of accuracy and communication cost in the networks with high connectivity (i.e., highly loopy networks). Furthermore, the communication cost is nearly constant with respect to the transmission radius. However, the drawbacks of proposed method are slightly higher (10-40%) computational cost and poor performance in low connected network. Anyway, the latter is not a problem since for the low-connected networks we can keep using NBP.

The remainder of this paper is organized as follows. In Section II, we provide a short review on cooperative localization using NBP. In Section III, we propose NBP method based on spanning trees. Experimental results for indoor scenario are presented in Section IV. Finally, Section V provides some conclusions and future work perspective.

II. COOPERATIVE LOCALIZATION USING NONPARAMETRIC BELIEF PROPAGATION

We consider the case in which some small number of anchor nodes, obtain their coordinates via GPS or by installing them at points with known coordinates, and the rest, unknown nodes, must determine their own coordinates. We suppose that all sensors with unknown positions obtain noisy
distance measurements of nearby subset of the other sensors in the network. This measurements can be obtained using a broadcast transmission from each sensor as all other sensors listen. Typical measurements techniques [11] are time of arrival (TOA), time difference of arrival (TDOA), receive signal strength (RSS) and angle of arrival (AOA). In this paper, we use RSS measurements.

Let us denote the received power by \( P_r(d) \). It is empirically accepted to model \( P_r(d) \) [mW] as log-normally distributed random variable with a distance dependent mean value. Thus, converted in dBm, it is given by:

\[
P_r(d)[dBm] = P_0(d_0)[dBm] - 10\sigma \log_{10} \left( \frac{d}{d_0} \right) + X \sigma
\]

where \( P_0(d_0) \) is known reference power value in dB milliwatts at a reference distance from the transmitter, \( \sigma \) is the path loss exponent that measures the rate at which the RSS decreases with distance, typically between two and four depending on the environment, \( X \sigma \) is zero mean Gaussian distributed random variable with standard deviation \( \sigma \) which accounts for the random effects of shadowing. It is trivial to conclude from (1) that, given \( P_r(d)[dBm] \), the estimated distance between a transmitter and receiver is:

\[
d = d_0 \cdot 10^{\frac{P_r(d)[dBm]-P_0(d_0)[dBm]}{10\sigma}} \cdot 10^\frac{X_\sigma}{\sigma}
\]

As we can see, the distance error is multiplicative (i.e., log-normally distributed) which means that RSS-based distance estimates have variance proportional to their true distance. Therefore, RSS is most valuable in high-density sensor networks which we target in this paper.

Regarding detailed statistical framework for localization using BP/NBP, we refer the reader to our previous results [12]. We define here only BP and appropriate potential functions. Since the single-node potential (prior information about position) is equal to one (no prior information), we only need to define pairwise potential function. The pairwise potential (probabilistic information about distance) between nodes \( t \) and \( u \) is given by:

\[
\psi_{tu}(x_t, x_u) = \begin{cases} 
P_d(x_t, x_u)p_u(d_{tu} - \|x_t - x_u\|), & \text{if } o_{tu} = 1 \\
1 - P_d(x_t, x_u), & \text{otherwise.} \end{cases}
\]

where \( d_{tu} \) is estimated distance, \( o_{tu} \) is a binary variable which indicates if there is a detection or not, and \( P_d(x_t, x_u) \) is probability of detection. We can now estimate the sensor positions by applying the BP algorithm. Each node \( t \) computes its belief \( M_t^i(x_t) \), the posterior marginal distribution of 2D position \( x_t \) at iteration \( i \), by taking a product of its local potential \( \psi_t \) with the messages from its set of neighbors \( G_t \):

\[
M_t^i(x_t) \propto \psi_t(x_t) \prod_{u \in G_t} m_{tu}^i(x_t)
\]

The messages \( m_{tu} \), from node \( u \) to node \( t \), are computed by:

\[
m_{tu}^i(x_t) \propto \sum_{x_u} \psi_{ut}(x_t, x_u) \frac{M_{tu}^{i-1}(x_u)}{m_{tu}^{i-1}(x_u)}
\]

In the first iteration of this algorithm, it is necessary to initialize \( m_{tu}^1 = 1 \) and \( M_t^0 = 1 \) (in case of no prior information) for all \( u, t \), and then repeat computation using (4) and (5) until sufficiently converge.

Since we use NBP, the belief and message update equations, (4) and (5), are performed using stochastic approximations, in two stages: first, drawing \( N \) weighted particles \( \{W_t^{j,i}, X_t^{j,i}\} \) from the belief \( M_t^i(x_t) \), then using these particles to approximate each outgoing message \( m_{tu}^i \). The detailed description of the algorithm can be found in [2].

III. NBP BASED ON SPANNING TREES

If we ignore the existence of loops and permit the nodes to continue communicating with each other, messages may circulate indefinitely around these loops, and the process may not converge to a stable equilibrium, or even if it converges, it could provide us less accurate estimates [1], [5], [6]. That is the main reason why we are going to implement NBP based on spanning trees (NBP-ST).

We start by describing the basics of graphical models. An undirected graph \( G = (V,E) \) consists of a set of nodes \( V \) that are joined by a set of edges \( E \). A loop is a sequence of distinct edges forming a path from a node back to itself. A tree is a connected graph without any loops. A spanning tree is an acyclic subgraph that connects all the nodes of the original graph. A root node is a node without parent and leaf node is a node without children. In order to define a graphical model, we place at each node a random variable \( x_s \) taking values in some space. In case of localization, this random variable represents the 2D or 3D position and each edge indicate that measurement is available. If we exclude anchors, the graph is undirected, but only for the first phase (spanning tree formation) we assume that it is directed (starting from chosen root node).

The optimal method for spanning tree formation for unweighted graphs is breadth first search (BFS) method [8]. It begins at the root node and explores all the neighboring nodes. Then each of those neighbors explores their unexplored neighbor nodes, and so on, until all nodes are explored. In this way, there will not be a loop in the graph because all nodes will be explored just once. The worst case complexity is \( O(v+e) \), where \( v \) is the number of nodes and \( e \) is the number of edges in the graph, since every node and every edge will be explored in the worst case.

In case of NBP localization, we exclude all the anchors from the BFS algorithm since they do not form the loops in the graph (they just send, and never receive the messages). A graph generally has a large number of spanning trees, but since our graph is unweighted we choose few (minimum
of them in a partly random way. In order to choose spanning tree, it is sufficient to choose root nodes for all spanning trees, then the algorithm will automatically set the spanning tree. Taking into account that we want to maximize the difference between two spanning trees, the root nodes can be chosen in two ways:

- The first root node we choose randomly from the set of all unknown nodes. The second root node has to be as far as possible from the first root node. Thus, it should be one of the leaf nodes which is the maximum-hop away from the root.
- We choose two (or more) anchor nodes which are far away from each other. The closest unknown nodes to chosen anchor nodes will be chosen as roots.

If we want to form more spanning trees, the analog procedure can be used. Since we will apply this algorithm for the indoor scenario, where the anchor nodes are usually fixed, we will use the second option. Note that, using BFS, it is not possible to form two spanning trees with completely different edges, and that usually some of the edges will be out of both spanning trees. Thus, if we want to ensure that all edges are used, we have to add more spanning trees, but it is usually not necessary since it would only provide us a redundant information. It is especially the case in highly connected networks (i.e., with high transmission radius) which we target in this paper.

The NBP method is naturally distributed through the network which means that there is no central processor which will handle all computations. Therefore, the proposed BFS method has to be done in a distributed fashion. This can be simply done if each unknown node initially broadcast its ID to all neighbors, which will continue to broadcast to others, and so on, until each unknown node has a list of all unknown nodes in the graph. One anchor node (e.g., with lowest ID) has to be assigned to choose the root node from that list and give him the permission to start BFS algorithm. Then, the chosen root node has all initial data to start the BFS algorithm, and, when it is necessary, has only to broadcast all data to all its neighbors. In the end, the last visited node has the output result (edges in spanning tree) and it just has to start multihop broadcast until each unknown node receives this result.

Finally, NBP-ST algorithm represents two (or more) independent runnings of the NBP algorithm based on formed spanning trees. Each running will provide us weighted particles of the node beliefs computed by (4). The collection of particles from all spanning trees represents our final output, from which we can easily extract any parameter that we need (e.g., mean value for location estimate). The pseudocode in Alg. 1 illustrates the NBP-ST method.

### IV. Experimental Results

In this section we start with the description of the setup used for the experiments performed in our lab. We then create reliable indoor model using obtained measurements and import all data into Matlab in order to check the performance of the proposed method in high-density sensor network.

#### Algorithm 1 NBP-ST method for localization

```
1: for all nodes do
2:   Take sensing actions
3:   Set all parameters to the initial values
4:       Broadcast own and all received IDs and listen for other sensor broadcasts (until receive all IDs)
5: end for
6: Set a list of nodes for BFS (excluding anchors)
7: Choose root node
8: for all spanning trees do
9:       Run BFS
10: Run NBP on defined spanning tree
11: Choose root node as far as possible from the previous roots
12: end for
13: Fuse all beliefs into one and compute location estimates
```

Figure 1: (a) Crossbow’s IRIS wireless sensor node, (b) Illustration of the experiment in our lab

#### A. Experimental Setup

For our experiments, we use Crossbow’s IRIS wireless sensor nodes (Fig. 1a) equipped with AT86RF230 transceiver. The AT86RF230 is high performance RF-CMOS 2.4 GHz radio transceiver specially targeted for low cost ZigBee/IEEE802.15.4 applications. The transmitter provides programmable output power: -17 dBm up to 3 dBm. The receiver, with -101 dBm sensitivity, generates digital signal with 3 dB granularity. The data is stored in a 128-byte dual port SRAM, from which 8 bytes are reserved. More details in [9], [10].

In order to estimate the distance between sensors, we placed two sensors, 2m from the floor, in our 5m x 10m lab (Fig. 1b) and set the transmission power to 3 dBm. Then, we obtained RSS measurements at 8 equidistant intersensor distances \((k \cdot 1.2m, k = 1, ..., 8)\). For each of them, we obtained 1000 measurements. Because of the 3 dB
granularity of RSS, we assume that the real power is a random variable uniformly distributed within the interval (RSS - 1.5 dB, RSS + 1.5 dB).

B. Indoor Modeling using Distance Measurements

Using obtained RSS measurements, our goal is to obtain all necessary parameters for indoor model: path-loss exponent, reliable distance estimation, probability of detection and potential functions.

Path-loss exponent. First, we define a reference point ($P_0(d_0 = 2.4m) = -61$ dBm). The path-loss exponent ($n_p$) could be easily obtained using another reference point, but this is not an optimal way. The better option is to, using all measured data, minimize the root mean square (RMS) error with respect to $n_p$:

$$e_{\text{rms}}^d(n_p) = \sqrt{\frac{1}{n} \sum_{i=1}^{n} (d^i_{\text{measured}}(n_p) - d^i_{\text{true}})^2}$$  \hspace{1cm} (6)

where $n$ is the number of inter-sensor distances (in our case, $n = 8$) and $d^i_{\text{measured}}(n_p)$ is given by:

$$d^i_{\text{measured}}(n_p) = d_0 \cdot 10^{-\frac{P_0\left[dBm\right] - P_i\left[dBm\right]}{10n_p}}$$  \hspace{1cm} (7)

Note that equations (2) and (7) are equivalent since the measured power includes the noise which accounts for the random effects of shadowing. According to Fig. 2a (dashed line), the optimal value of the path-loss exponent is $n_p = 2.7$.

Reliable distance estimation. Using obtained measurements and estimated $n_p$, we can estimate the distance. As we expected our indoor model is not similar to the ideal one (Fig. 2b), so the distance cannot be always trustfully estimated using (7). For instance, the averaged received power of -66 dBm corresponds to three different distances (4.6m, 7m and 9.6m), so the sensor has no other option, but to guess. This is because the power is not monotonically decreasing function of the distance. Therefore, we have to cut out the area below the threshold power (-64 dBm) because this area corresponds to the non-monotonic part of the function. Above the threshold, each received power corresponds to the unique distance, which makes this model reliable for our scenario. In addition, since we excluded data below the threshold, we must re-estimate $n_p$ using only the remaining data. According to Fig. 2a, $n_p = 1.2$.

We illustrated in Fig. 2c, the distance estimation which corresponds to the true value of 1.2m. As we can see, the error model ($d^i_{\text{measured}} - d^i_{\text{true}}$) is not similar to the log-normal distribution, so we will not use any parametric form of the error distribution. Moreover, we have three different error models (for 1.2m, 2.4m, and 3.6m). Thus, in order to import this error model into Matlab, we will simply draw the samples from the nearest error model and add it to the true distance (i.e., this is nearest neighbor interpolation, so for the true value of e.g., 2.9m, we use error model for 2.4m).

Probability of detection. For each inter-sensor distance, we found that RSS is above the power defined by sensitivity. This is expected because we set the transmission power to the maximum which could even provide us around 75m radius, according to ZigBee standard. Anyway, we have to follow defined reliable model, so we assume that, if the power is less than threshold (-64 dBm), there is no communication between nodes. This could be easily forced by software. As we can see in Fig. 2b, the corresponding distance is 4m, so this will be the maximum value of transmission radius. Note that, in our case, we didn’t detect communication failures (link quality indicator is always maximum), so we set $P_d = 1$ in the transmission range, and otherwise $P_d = 0$. This is expected due to the very small distance between nodes.

Potential functions. Since we don’t have any a priori information about positions of unknown nodes, we have to define only pairwise potential. According to (3), given anchor node (or particle of unknown node), the position of other node is shifted in the random direction by mea-
Figure 3: Pairwise potential function $\psi_{ut}(x^*_t, x_u)$ ($x^*_t$ - anchor, $x_u$ - unknown) using (a) log-normal model, (b) indoor model from our lab.

Figure 4: (a) Original network, (b),(c) two corresponding spanning trees. Connections between anchors (marked by red squares) and unknowns (marked by black circles) are not shown.

Figure 5: Comparison of accuracy

---

C. Simulations

We placed 50 unknowns and 10 anchors in a 5m x 10m area (Fig. 4). Unknown nodes are deployed randomly within this area and anchor nodes are fixed (8 along the edges and 2 in the center area). This constraint, realistic for indoor scenarios, helps the unknown nodes near the edges which suffer from low connectivity. The number of iteration is set to $N_{iter} = 3$. According to our analysis, this number is sufficient for good convergence. All simulations are done for $N = 50$ particles with respect to the transmission radius ($R = 2m - 4m$). Finally, each point in the simulations represents the average over 30 Monte Carlo trials.

Using the defined scenario, we compared NBP and NBP-ST algorithms. For NBP-ST, we used 2 and 3 spanning trees. The error is defined as Euclidean distance between true and estimated location. As we can see in Fig. 5, NBP-ST performs better than NBP starting from some value of $R$, which controls the connectivity. We can conclude the same for the coverage (Fig. 6), which represents the percentage of located nodes with error less than predefined tolerance. Obviously, for higher values of $R$, there is a large number of loops in the network (hundreds, in our case) which decreases the performance of the NBP method. For lower values of $R$, we could expect that NBP-ST performs with higher (or same) accuracy, but we cannot forget that, by using only 2 or 3 spanning trees, we didn’t include all information (i.e., removed edges) that we have. Thus, in this case the NBP overperforms NBP-ST.

To measure the communication cost, we count elementary messages, where one elementary message is defined as simple scalar data. We assumed that this data is represented in single precision floating-point format that occupies 4 bytes in the memory. As we have already mentioned, 8 bytes are already reserved, so the size of an elementary message is 12 bytes. According to Fig. 7, NBP-ST performs better than NBP for $R > 3.3m$ only if we use 2 spanning trees. In order to explain this we have to remember two main things we have taken into account: removing the edges in order to form the spanning trees and running NBP two times in these spanning trees. First operation decreases the communication, but the second one increases it. Therefore, in low connected networks the second operation predominates, but in high connected networks the first one predominates.

Regarding computational cost, NBP (0.15-0.25 MFlops per node) slightly overperforms NBP-ST (0.21-0.28 MFlops per node). Furthermore, it is important that the communication cost is nearly constant with respect to the transmission radius. This feature provides us more precise information about battery life. Finally, if we use 3 or more spanning trees, both computational and communication cost will be obviously significantly higher.

The final conclusion is that NBP-ST (with 2 spanning trees) algorithm performs better than NBP in terms of accuracy and communication cost, for $R > R_{min}$. In our case $R_{min} = 3.4m$, but this parameter depends on the density in the network (i.e., average connectivity). On the other hand, if the unique goal is accuracy, user should increase the number of spanning trees.
V. CONCLUSIONS AND FUTURE WORK

As presented in this paper, NBP localization algorithm has poor performance in highly loopy networks. Moreover, the connectivity in these networks is very high which makes communication burden for low-power applications. Therefore, we proposed NBP-ST method based on spanning trees created by the BFS method which is optimal for the unweighted graphs. The BFS method is done in a distributed way which makes the algorithm applicable in ad-hoc/sensor networks. We can conclude that NBP-ST method performs better than NBP in terms of accuracy, and communication cost in highly connected networks. However, the drawbacks of proposed method are slightly higher computational cost and poor performance in low connected network. There remain few open directions for the future work. One possible future line is the implementation of localization algorithm based on nonparametric generalized belief propagation (NGBP). Some versions already exist, but they are still very complex for the large-scale sensor networks. Furthermore, real-time tracking using these methods could be an interesting direction. This will be a part of our future research.

ACKNOWLEDGMENT

This work is supported by the FPU fellowship from Spanish Ministry of Science and Innovation. Furthermore, we thank partial support by project ICT-217033 WHERE, program CONSOLIDER-INGENIO 2010 CSD2008-00010 COMONSENS and National Project M3HF.

REFERENCES


