A Localization Algorithm with Learning-based Distances

DuyBach Bui, Daeyoung Kim
Auto-ID Lab Korea
Information and Communications University
Daejeon, South Korea
bduybach@cs.uiuc.edu, kimd@icu.ac.kr

***Abstract*—Existing range-based localization algorithms are superior only when a high accuracy node-to-node measured distance exists. This assumption is actually difficult to satisfy with current ranging techniques used in tiny sensor nodes. Meanwhile, range-free localization algorithms work independently of ranging error but can only produce limited node accuracy. In this paper we propose a novel localization scheme that uses a learning-based distance function to estimate distances. The adaptation of distance function to ranging error and other network conditions, i.e. network density, number of anchor, results in better estimated distances. This leads to more accurate positions calculation comparing to existing works, especially when ranging error is high.**

I. INTRODUCTION

Feasibility of many wireless sensor networks (WSN) applications and various location-aided network protocols depends on the availability of sensor node positions. A sensed data with position information is more meaningful for various applications, such as smart environment [1], natural habitat monitoring [2], and disaster relief. A localization algorithm is used to find the position of each sensor node.

Basically, node position calculation requires the knowledge of physical distances between nodes. A distance is measured by using sensors such as ultrasound transceiver and received radio signal strength indicator, or/and geometrically estimated. Both methods are subject to error. Given inaccurate measured and estimated distances, one is incapable of calculating node positions to a certain desired degree of accuracy.

The challenges have attracted many studies, reflecting broad ranges of approaches in developing the technology needed to find node position [3, 4, 5, 6, 7, 8, 9, 10, 12]. Based on the mechanism used for distance estimate, these approaches are divided into two categories: range-based and range-free. The former relies on measured distance to calculate node position. It works well based on the assumption of perfect ranging. This assumption is actually difficult to satisfy with current ranging techniques in sensor networks. On the other hand, the latter makes it robust in terms of ranging error by not using measured distances. However its estimated position accuracy is inadequately low.

Our paper aims at proposing a better localization algorithm, which is supposed to provide higher estimated node position accuracy than existing works especially in the condition of unreliable ranging. The main theme to achieve the goal is effectively using information in estimating node-to-node distances. With well estimated distances, position estimates, in turn, are better. To do that we firstly propose a parameterized distance model, representing node-to-node distances. An algorithm is then introduced to learn the coefficients of the function. The learning process takes into account the conditions of the network so that the learned function is better representing the real distances.

Figure 1 draws the wireless sensor networks to which LDL applies.

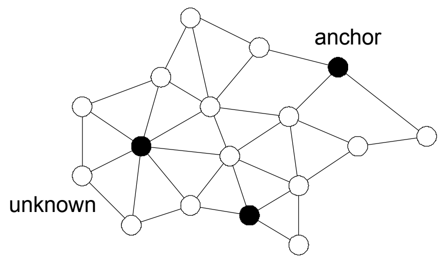

Figure 1. Wireless Sensor Networks

This network includes a small number of anchor nodes, knowing their own positions a priori by either using GPS or being manually configured. The other majority of nodes are unknown position nodes, called unknowns. Both anchors and unknowns are equipped with a low power measuring device, which is used for measuring distances between it and its neighbors. A measured distance is subject to error because of both intrinsic technological constraints and extrinsic environmental conditions. Some unknowns, neighbors of anchors, can measure distances to anchors directly, but most of them are multi-hops far away from anchors and must estimate distances to anchors relying on others' distances. We assume that every node has the same radio range and can measure distance within the radio range. This is a practical assumption if a sensor node uses RSSI for ranging.

The paper is structured as follows. Section II introduces related works, including available ranging technologies and localization algorithms. In Section III, the node-to-node distance model is proposed. Section IV introduces the main algorithm including the distance learning algorithm. We describe simulation setup and evaluation, comparing with both range-base and range-free approaches in Section V. Finally, we conclude in Section VI.

This work was supported in part by Samsung and ITRC program of the Korean Ministry of Information and Communication.

II. RELATED WORKS

A. Ranging Technology

In this subsection we classify the ranging technologies in terms of the devices which are used for measuring distance. Currently, there are three main ranging technologies applicable to WSN such as acoustic device with ultrasound or audible frequency and radio signal strength. Of three approaches, ultrasound is the most stable technique, giving a reasonable and small ranging error, in an ideal environment. 0.5-centimeter measurement accuracy with 5 meters distance is reported, using ultrasound [5]. However, this technique is not able to perform well in an environment with obstruction. Moreover, a node, equipped with ultrasound transducer, can most accurately measure distance only if receiving and transmitting transducer are face-to-face with each other, which is usually difficult to satisfy in most real applications. Authors in [3] report that despite being extensively calibrated, ultrasound devices still have an average ranging error from 10% to 20% in a laboratory environment. Other technologies using audible frequency or received radio signal strength, do not have strict arrange requirement, but this hardware is highly variable with low ranging accuracy. Distance errors of up to 100% of measurement range with radio signal strength and 300% with acoustic hardware are also reported in [3].

B. Localization Algorithms

1) Range-free algorithms

Avoiding using measured distance in estimating positions, range-free approaches are robust in terms of ranging error. In DV-Hop [4], an *unknown* position node estimates distance to an *anchor,* by multiplying the number of hops between them with the average distance of each hop, which is computed by *anchors*. Amorphous system [7], also uses the same mechanism as DV-Hop, i.e. multiplying hop count with the average hop distance to estimate distance. This algorithm, however, uses an offline average hop distance, which is estimated using a formula, introduced in [11]. Given the node density of a wireless ad-hoc network, one can estimate the expected hop distance a priori by using this formula. Other proposed schemes such as APIT [8] also work as a range-free scheme with some different assumptions from the above algorithms and our algorithm. First, although it can work with static *anchors*, this algorithm is designated for systems with mobile *anchors*. In addition this algorithm considers an *anchor* with radio range longer than other nodes. Because of these differences in assumption, we exclude it from comparison. APIT shares the same inherent disadvantage as previous range-free approaches in that, without using measured distance, they are incapable of providing fine-grain node positions.

2) Range-based algorithms

In range-based approaches, distances between nodes are measured. This information is then used for estimating node positions. In AHLos [5], the distance from a node to an *anchor* is estimated by adding up measured distances on each hop on the way from the node to the *anchor*. Although this algorithm reports a low position error in an environment of low ranging error, its performance dramatically reduces once the ranging error exceeds 10%. Another range-based scheme, Euclidean algorithm [11] and Hop-Terrain [12] also work well only with perfect ranging.

MDS algorithm [9] can work with either connectivity only (range-free) or measured distance (range-based). The advantage of MDS is it performs well even with as small number of anchors as 3. However, MDS either range-based or range-free has the same disadvantages of aforementioned approaches.

Robust-Quadrilaterals algorithm [10] avoids large position estimate error caused by ranging error by not including nodes that may have large error in estimate. The result is many nodes are not located when ranging error is high or node density is low.

The fundamental difference between our approach and existing approaches is that instead of assuming small ranging error or avoiding ranging error by not using measured distances, our algorithm adapts estimated distances with the ranging error and other network characteristics to produce the best possible position accuracy.

III. NODE-TO-NODE DISTANCE MODEL

In this section, we are introducing a parameterized distance model used to represent a node-to-node distance in our algorithm. The followings are definitions related to our model.

Def 1. *Distance-related variable between two nodes in a network is any knowledge, taken from the network, which stores information about the real distance between the nodes and can be used to estimate the distance.*

For example, the hop count of the shortest path between any two nodes is a distance-related variable.

Def 2. *A node-to-node distance function of a network is a mapping from distance-related variables between any two nodes in the network to the real distances between them.*

$$d_W(X) = f(X,W) \quad (1)$$

where $X=(x^1,\ldots,x^N)$, the argument of the distance function, is the set of N distance-related variables. W is a vector of parameters, whose value may be different in different networks.

The followings are the two proposed distance-related variables belonging to the distance between node i and node k; that we use in this paper:

Cost in distance of the shortest path (r): sum of distances on the shortest path from node i to node k.

Cost in hop of the shortest path (h): the number of hops on the shortest path from node i to node k

The relationship between a real distance and its distance-related variable will decide the form of $f(X,W)$. These relationships can be seen in Figure 2 and Figure 3, which are drawn for a WSN with 100 sensor nodes, randomly deployed in an area of 30×30, 10% of *anchors* and 10% of ranging error. For each pair of a node and an *anchor*, we computed their real distance and collected the corresponding r and h. Each point in the two figures shows the relationship of a real distance of a node to an *anchor* and the corresponding r and h. Approximately, the relationships between a real distance and its r and h are linear.

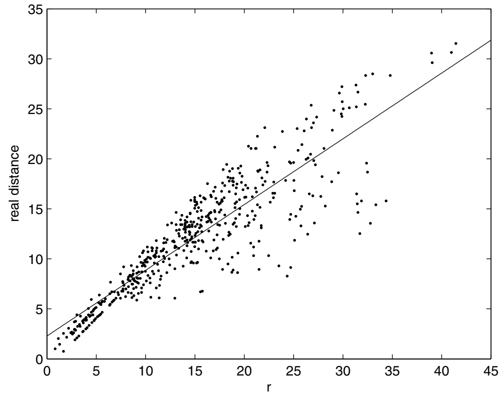

Figure 2. Real distance and r

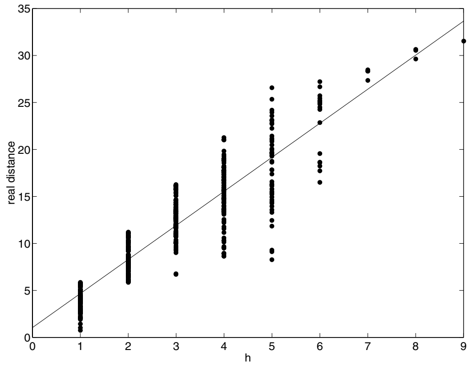

Figure 3. Real distance and h

With this observation, we propose the following distance function:

$$d_w = WX^T \tag{2}$$

where $X = [r \ h \ 1]$. To estimate the parameter W of the distance function in each network we use a supervising learning technique. The detail algorithm is addressed in the next section.

IV. ALGORITHM

A. Learning Distance Function

To learn the distance function by using supervised learning, one needs a number of learning samples. In the problem at hand, a learning sample is defined as a pair of distance-related variables X between two nodes and their real distance d. Because a distance between two *anchors* can be exactly calculated using their known coordinates, 2-tuple *(X,d)* of pairs of *anchors* can be used as samples. Using these samples and a learning algorithm such as least square error gradient decent, we can estimate the distance function. The role of learning process is to reveal the correlation between each distance related variable and the real distance then adapts the function coefficients so that the variables which are more correlated to the real distance have more weight in the function.

B. Learning-based Distance Localization Algorithm

Our Learning-based Distance Localization (LDL) algorithm is distributed and consists of three steps

1. Every node finds the shortest paths between it and *anchors*, collecting the distance-related variables $[r \ h]$ belonging to these shortest paths and the anchors' coordinates.
2. Each *anchor* uses collected information to learn the distance function's coefficients as mentioned in previous subsection. *Anchors* then broadcast the estimated function's coefficients.
3. Upon receiving the broadcasted distance function coefficients, *unknowns* calculate distances to *anchors* using the received distance function. With these estimated distances and *anchors'* coordinates, the nodes use multi-lateration to calculate their positions.

In Step 1, each node in the network runs the following algorithm

Algorithm: distributed algorithm to find the shortest path between nodes and anchors. Each node in the network maintains three arrays which store parameters to anchors: hop count H and cost in distance R to anchors, and coordinates C of anchors. Distance measurements to direct neighbors have been gathered in array M e.g. M[i] is the measured distance to neighbor i. A is the number of anchors. BROADCAST function and RECEIVE function have four arguments *(h,r,c,a)*, where h and r are hop count and cost in distance to anchor a, and c is coordinates of anchor a

$R[1..A] = \infty; H[1..A] = \infty; C[1..A] = \infty$
if isAnchor **then**
  BROADCAST$(0, 0, myCoordinates, myID)$
end
while $timeout == 0$
  if RECEIVE(h, r, c, a) from a neighbor v **then**
    if $h + 1 < H[a]$ **then** //update anchor a parameters
      $H[a] = h + 1$
      $R[a] = r + M[v]$
      if $C[a] = \infty$ **then**
        $C[a] = c$
      end
    end
    BROADCAST$(H[a], R[a], C[a], a)$
  end
end

In step 2, anchors may share their collected information and estimate a unique set of function coefficients in the whole network or they estimate with their collected information only. The latter is preferred because it requires less communication and computation. In this case, unknowns may receive different coefficients from different anchors. The distance function coefficients of an unknown are computed as follow $\sum_{i=1}^{n} W_i / A$, where A is the number of anchors, W_i is coefficients received from anchor i. Our simulations show that there is no significant difference in the performance of the two approaches.

C. *Communication Overhead*

The algorithm requires two broadcast waves. The first wave is used to find the shortest paths between nodes and anchors. In this step, each node receives and forwards at least one packet for each anchor. However, because of the distributed nature of the algorithm, a node may receive more than one version of a packet originating from an anchor which are forwarded by different neighbors of the node. If the packet containing the smallest hop count from an anchor comes later than other packets from the same anchor, a node will forward more than one packet for that anchor. To avoid these redundant communications, each node may use a timer for each anchor. For a node, an anchor timer is set when the node receives the first packet originating from that anchor. The forwarding of the anchor's parameters (step **14**) is delayed until the timer expires. During this delay the node may receive other packets originating from the same anchor and updating the anchor's parameters. In our simulation we set the delay timer is larger than the maximum value of the MAC layer back-off timer. With a proper delay timer, the communication overhead of each node in the first broadcast wave is reduced to as low as *A* broadcast packets, where *A* is the number of anchors. The communication overhead of the second broadcast wave is also *A* broadcast packets for each node. This wave is used to transfer the distance function coefficients computed by anchors to unknowns.

V. EVALUATION

A. *Simulation Setup*

We use ns2 with 802.11 protocol at the MAC layer to simulate the new algorithm. The distance measurement is corrupted by a normal distribution error with zero mean and varied standard deviation. All scenarios include 100 sensor nodes randomly deployed in an area of 30×30. All nodes have the same radio range R. *Anchors* are randomly selected. The network density is controlled by varying radio range, i.e. a smaller radio range is equivalent with a smaller network density.

In the following subsections we evaluate our algorithm, comparing it with two typical range-based and range-free algorithms i.e. AHLos and DV-Hop.

B. *Average Position Error*

Three different network parameters are considered: ranging error, *anchor* fraction and connectivity degree.

1) *Ranging error*

To evaluate the effect of ranging error, we used 200 different scenarios with the *anchor* fraction of 10%, the connectivity degree of 14, which means each sensor node has 14 neighbor nodes on average. Figures 4 shows the mean average position error of LDL, AHLos, and DVHop, when the standard deviation of ranging error changes from 0 to R.

An important observation is that LDL shows a significant performance improvement, in terms of the accuracy of node positions, over both range-based and range-free approaches in a broad range of the ranging error. In the absence of ranging error, LDL gives 5% and 25% higher accuracy than AHLos and DV-Hop, respectively. Different from range-based techniques, LDL is less sensitive to ranging error. With extremely high ranging error, i.e. 100%, its average position accuracy is 60% and 5% better than AHLos and DV-Hop, respectively. While range-free approaches inherently can not take the advantage of the accuracy of ranging information, LDL does. Given a smaller ranging error, LDL can produce appropriately higher node position accuracy.

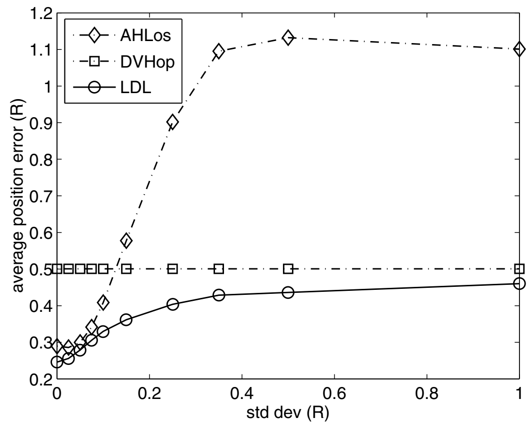

Figure 4. Average position error of three algorithms with different ranging error. Anchor fraction and connectivity degree are fixed to 10% and 14, repectively.

In terms of information usage, we can explain this fact as follows. When ranging error is low, the amount of information about the real distance contained in *r* is high; the learning process adapts the distance function such that weight of *r* is higher than other information. Vice versa, when high ranging error causes the correlation between *r* and the real distance decreased, the learning process reduces the weight of *r* and increases the weight of *h* in the distance function. The fact that LDL outperforms in a broad range of ranging error shows that flexible use of more than one available information can improve the robustness and computational quality.

2) *Anchor fraction and connectivity degree*

Figure 5 and 6 demonstrate the effect of percentage of *anchors* and connectivity degree on estimated position error, respectively.

When the number of anchors decreases, accumulated error in data increases because of the longer path from an anchor to a destination, resulting in a lower accuracy. When the connectivity degree increases the average errors of all algorithms reduce. Note that the increase in connectivity degree, i.e. a node has a larger number of neighbors, results in the higher probability of finding more appropriate values of distance-related variables, which represent the real distances better. Position estimates, in turn, have higher accuracy. It is shown that the better performance of LDL is consistent in all conditions.

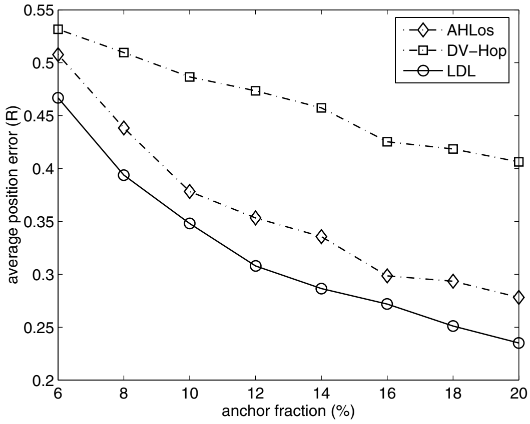

Figure 5. Average position error of three algorithms with different anchor fraction. Connectivity degree and ranging error are fixed to 0.1R and 14, respectively.

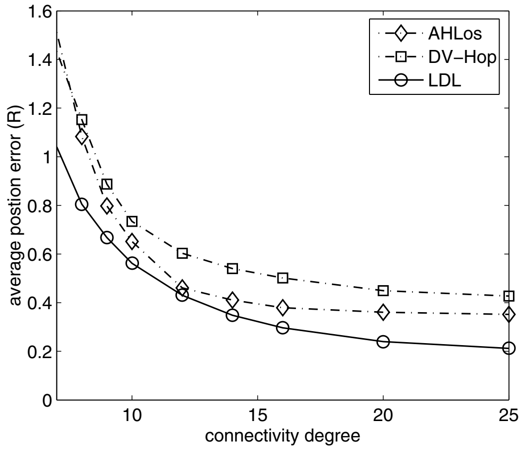

Figure 6. Average position error of three algorithms with different connectivity degree. Anchor fraction and ranging error are fixed to 10% and 0.1R

C. Distance function adaptation

In this subsection, we analyze the behavior of LDL algorithm in scenarios where sensor nodes are deployed in areas with different ranging error.

We do simulations with 100 nodes randomly deployed in an area of 40×20, with the connectivity degree of 10 and anchor fraction of 10%. In area A (0≤X≤20; 0≤Y≤20) the ranging error is 0.9R, whereas in area B (20≤X≤40; 0≤Y≤20) the ranging error is 0.1R. Each node only collects the information from 6 closest anchors.

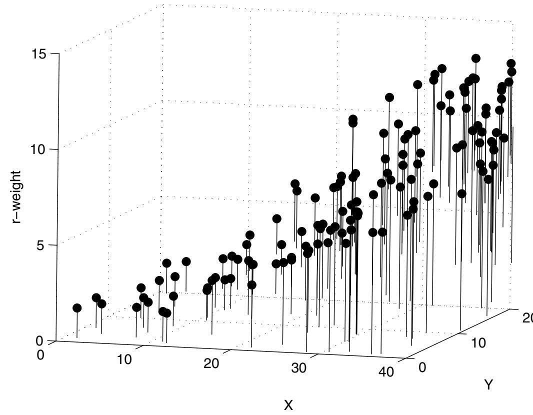

Figure 7. Distance function coefficient belonging to variable *r*

Figure 7 and 8 draw the distance function coefficients belonging to distance-related variable *r* and *h* of each node, respectively. The figures show that the distance function is able to adapt to the ranging error in each area. In area A where ranging error is high the weight of range-based variable *r* is lower than in area B, whereas the weight of range-free variable *h* in A is higher than that in B. This adaptation allows nodes in each area to better estimate distances to anchors. This characteristic is especially important for large-scale sensor networks in which sensor nodes are deployed in a large area with different conditions.

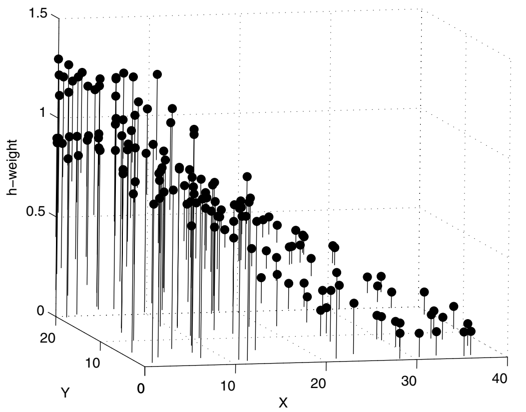

Figure 8. Distance function coefficient belonging to variable *h*

VI. Conclusion

Given the need of a more robust localization algorithm with reasonable node position accuracy in the condition of high ranging error, we proposed a novel approach for WSN localization to meet that requirement. Using available information more efficiently, the LDL algorithm shows that it can work better and more robustly than existing ones, i.e. producing higher node position accuracy in a wide range of working environments. In addition, LDL algorithm is able to estimate the appropriate distance function for each area in a large scale sensor network which comprises areas with different condition.

For future works, we are interested in implementing LDL algorithm in our sensor network system. Also we want to extend the node-to-node distance model to include other distance-related variables.

[1] M.B Srivastava, R.Muntz and M.Potkonjak, Smart Kindergarten: Sensor-based Wireless Networks for Smart Developmental Problem-solving Environments, In *Proceedings of the seventh annual international conference on Mobile computing and networking, Mobicom 2001, 132-139*.

[2] http://www.greatduckisland.net/

[3] K. Whitehouse, The design of Calamari: an ad-hoc localization system for sensor networks, *Master's Thesis, University of California at Berkeley (2002)*.

[4] D. Niculescu and B. Nath. Ad hoc positioning system. In IEEE Globecom 2001, volume 5, pages 2926 –2931, April 2001.

[5] A. Savvides, H. Park and M. Srivastava, The Bits and Flops of the N-Hop Multilateration Primitive for Node Localization Problems, In First ACM International Workshop on Wireless Sensor Networks and Application, Atlanta, GA, September 2002.

[6] K. Langendoen, N. Reijers, Distributed Localization in Wireless Sensor Networks: A Quantitative Comparison, In Computer Networks: The International Journal of Computer and Telecommunications Networking, Volume 43, Issue 4 (November 2003), page 499 - 518.

[7] R. Nagpal, H. Shrobe, J. Bachrach, Organizing a Global Coordinate System from Local Information on an Ad Hoc Sensor Network, In the 2nd International Workshop on Information Processing in Sensor Networks (IPSN '03), Palo Alto, April, 2003.

[8] T. He, C. Huang, B.M. Blum, J. A. Stankovic, T. Abdelzaher, Range-Free Localization Schemes for Large Scale Sensor Networks, In The Ninth Annual International Conference on Mobile Computing and Networking (MobiCom 2003).

[9] Y. Shang, W. Ruml, Y. Zhang, M. P. J. Fromherz, Localization From Mere Connectivity, In The Fourth ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc'03).

[10] D. Moore, J. Leonard, D. Rus, S. Teller, Robust Distributed Network Localization with Noisy Range Measurements, In ACM SenSys'04.

[11] L. Kleinrock and J. Silvester, Optimum Transmission Radii For Packet Radio Networks Or Why Six Is A Magic Number. Proc. Natnl. Telecomm. Conf., pages 4.3.1-4.3.5, 1978.

[12] C. Savarese, J. Rabaey, and K. Langendoen, "Robust positioning algorithm for distributed ad-hoc wireless sensor networks," In USENIX Technical Annual Conf., Monterey, CA, June 2002.