

Available online at <http://www.meecspress.net/ijwmt>

Intelligent and Distributed Localization of Nodes in Wireless Sensor Networks

Ibrahim S. I. Abuhaiba^{a,*}, Amina Y. Al-Sallut^a

^aComputer Engineering Department, Islamic University, P. O. Box 108, Gaza, Palestine.

Abstract

In wireless sensor networks, the issue of nodes localization has taken a wide area of research. Most applications need to know the position of sensor nodes for reasons of optimal and fast data routing. In this paper, a new distributed localization algorithm based on Self Organizing Maps (SOMs) is proposed to determine the location of a node in a wireless sensor network.

The proposed algorithm is classified as a range-free algorithm which uses only the connectivity information between nodes without the need to measure the time of arrival or signal strength as range-based algorithms require. It utilizes the neighborhood information and the well-known anchors' positions to calculate the estimated locations of nodes. Our algorithm is made up of two main stages. The initial estimated locations of nodes are calculated in the initialization stage, and fed to the learning stage in which a SOM is used to calculate the final estimated locations of nodes.

By using the neighborhood information at the first stage, the algorithm has significantly reduced the SOM learning time and the number of iterations to converge. On the other hand, starting with real data rather than random data maximized the accuracy of the resulted locations. Furthermore, the distributed implementation of the algorithm highly alleviated the pressure on the wireless nodes which are characterized with low power and limited capabilities.

The proposed algorithm has been implemented using MATLAB software and experimented by deploying different number of nodes in a specific area with different communication radio ranges. Extensive simulations evidently verified the performance of the algorithm and achieved a very good accuracy. Moreover, the algorithm proved its effectiveness with a lower average error and lower number of iterations compared to other related algorithms.

Index Terms: Wireless sensor networks, Localization, Anchor nodes, Self organizing maps.

© 2015 Published by MECS Publisher. Selection and/or peer review under responsibility of the Research Association of Modern Education and Computer Science

* Corresponding author.

E-mail address: isiabuhaiba@gmail.com

1. Introduction

A wireless sensor network (WSN) is usually a relatively large-scale network of inexpensive energy efficient devices [1, 2]. For a node in a WSN, awareness of its location and maybe the location of some other nodes is crucial for a successful operation. As a case in point, routing data in sensor networks requires a fine cooperation among nodes in order to use small amount of energy and to deliver data as fast as possible. A node can choose a proper way to the destination, usually a sink node, if it knows geographic location of itself and its neighbor nodes [3]. Furthermore, most applications of sensor networks need to know the position of sensor nodes. For instance, a jungle watching WSN must find out and report the location of a probable fire [4].

Using Global Positioning System (GPS) devices is the simplest way to determine the location of a sensor node [5]. Nevertheless, because of some traits of GPS devices which are in contrary with sensor networks demands, using them in all sensor nodes is not justifiable. These traits include relatively high cost, high weight, and debatable accuracy of GPS equipment in some situations. To overcome GPS limitations, many localization techniques have been developed for sensor networks which do not depend on GPS devices merely. In these localization methods, a few nodes, called anchors or seeds, are equipped with GPS devices and help other nodes to determine their positions.

Many algorithms have been proposed for localization of static WSNs [6-8]. Nodes in static WSNs do not have movement; in consequences, if a node of these networks could estimate its location once, it would not have to repeat the localization process again. Nodes in mobile WSNs may move by an external agent like wind, animal's movements, stream of a river, or by internal movement agents like wheels and continuous track [9, 10]. Mobility has two contrary effects on localization process. In one hand, as previous works indicate, mobility can help localization of static sensor networks [11]. In that, more nodes can get information from mobile anchor nodes. On the other hand, mobile sensor networks may suffer from rapidly changing situations which lead to less validation time for observed information. In general, the previous localization algorithms concentrated on static WSNs due to the high importance of this issue. Under mobility conditions, a static localization algorithm is supposed to be applicable with some superficial modifications and periodic mobility parameters tracking.

In this paper, we consider the problem of locating and orienting a wireless ad-hoc network of unattended sensor nodes that have been deployed in an area at unknown locations. The basic concept is to deploy a large number of low-cost and self-powered sensor nodes that acquire and process data. The sensor nodes may include one or more acoustic microphones as well as seismic, magnetic, or imaging sensors. We consider location estimation in networks where a small proportion of devices, called reference devices or anchors, have a priori information about their coordinates. All devices, regardless of their absolute coordinate knowledge, estimate the range between themselves and their neighboring devices. Such location estimation is called relative location because the range estimates collected are predominantly between pairs of devices of which neither has absolute coordinate knowledge. We intend to implement a range-free localization algorithm with the consideration of power limitations of sensor nodes, the need for accurate results, and the time required to execute the algorithm.

The rest of the paper is organized as follows. Section 2 briefly describes some of the most related works that have been done in the wireless nodes localization problem. The advantages and drawbacks of every work are mentioned and discussed. Section 3 describes accurately the proposed technique to localize the nodes in a given WSN. In Section 4, the experimentation environment, simulations, and results are demonstrated and analyzed. Comparisons to other related works have been done and analyzed to verify effectiveness of the proposed algorithm. Finally, the concluding remarks and future work are presented in Section 5.

2. Related work

Recently, mobile ad-hoc network localization has received attention from many researchers [12]. Many algorithms and solutions have been presented so far. These algorithms are ranging from simple to complicated

schemes, but they can be categorized as range-based and range-free algorithms. Range-based algorithms utilize the distance measured between nodes by either using the Time-Of-Arrival (TOA) [13], Time-Differential-Of-Arrival (TDOA) [14], Angle-Of-Arrival (AOA) [15], or Received-Signal-Strength-Indicator (RSSI) [16] technologies. However, they usually need extra hardware to achieve such measurement. When calculating the absolute location, most schemes need at least three anchors (nodes that are equipped with GPS or know their location in advance).

Range-free algorithms utilize only connectivity information and the number of hops between nodes. They are widely used due to the observable advantages over the range-based algorithms especially the conservation in power consumption in wireless devices. Many range-free algorithms utilized different types of artificial neural networks in the localization process. One of the most recent used neural networks is the Self Organizing Maps (SOMs), which showed its effectiveness in the localization process over other algorithms.

2.1. Range-based localization algorithms

The traditional ranging methods based on received signal strength (RSSI), time of arrival (TOA), angle of arrival (AOA), time difference of arrival (TDOA), etc., have several shortcomings from the point of view of the sensor networks. RSSI is usually very unpredictable since the received signal power is a complex function of the propagation environment. Hence, radios in sensors need to be well calibrated otherwise sensors may exhibit significant variation in power to distance mapping. TOA using acoustic ranging will require an additional ultrasound source. TOA and RSSI are affected by measurement as well as non-line of sight errors. TDOA is not very practical for a distributed implementation. AOA sensing will require either an antenna array or several ultra-sound receivers [17].

The Active Badge Location System is often credited as one of the earliest implementations of an indoor sensor network used to localize a mobile node [18, 19]. Although this system, utilizing infrared, is only capable of localizing the room that the mobile node is located in, many other systems based on this concept have been proposed. The Bat system, much like the Active-Badge System, also utilizes a network of sensors [20, 21]. This system features a central controller that emits a query which the mobile node responds to with an ultrasonic pulse. This pulse is picked up by a network of receivers at varying times due to their different locations. These times can be used to compute the distances and hence the location of the mobile node. Researchers at Massachusetts Institute of Technology (MIT) have utilized similar concepts from the Bat System in their Cricket sensors, using a more decentralized structure. This system requires less of a support infrastructure than the Bat system. The Cricket location system uses a hybrid approach consisting of an extended Kalman filter, least square minimization to reset the Kalman filter, and outlier rejection to eliminate bad distance readings [22]. Other researchers at MIT have proposed localization by exploiting properties of robust quadrilaterals to localize an ad-hoc collection of sensors without the use of beacons [23].

It is also possible to localize optically as shown in the HiBall head tracking system [24]. Arrays of LEDs flash synchronously, and cameras capture the position of these LEDs. The system utilizes information about the geometry of the system and computes the position. Localization using signal strength of RF signals has been studied extensively [25-28].

Monte Carlo Localization (MCL) is also one of the first practical methods for localization of mobile WSNs. Sequential Monte Carlo method had been used for localization of mobile robots [29]. This technique has been adapted to produce a practical method for localization of mobile sensor networks [30]. In sequential Monte Carlo methods, the current state of a system can be obtained by using its current observations and its posterior state. In MCL, the time is divided into discrete intervals. A sensor node moves during a time interval and localizes at the beginning of the next time interval. The main idea of Monte Carlo Localization Boxed (MCB) method is to limit the area in which the samples are drawn in MCL [31]. Unlike the MCL, MCB uses the information obtained from anchor nodes both before and after generation of samples. This can lead to faster and more efficient sample generation.

All of these approaches complain from the use of range-based (signal strength) and connectivity information, that are hard and slow to be collected, need hardware support of network nodes, and are rapidly changing especially in mobile networks.

2.2. Range-free localization algorithms

Some other schemes are range-free and use connectivity information only. One of the first examples of such techniques is the “GPS-Less” positioning system, where nodes use a centroid approach to estimate their position by averaging the coordinates of nearby anchor nodes [32]. Distance Vector-HOP (DV-HOP) is a typical range-free algorithm, where anchor nodes flood the network with message beacons that are used by each node to determine the minimal hop count distances [33]. An estimate of the average hop length is used to obtain distance values and perform multilateration. A similar approach is proposed in [34] and in [35] as an Ad-hoc Positioning System (APS). It uses distance-vector forwarding technique to get the minimum hop count from a node to heard anchors. By using corrections calculated by anchors (average hop-distance between anchors), nodes estimate their location by using lateration (triangulation) method.

Compared to DV-HOP, some other algorithms seem to be more complicated, but have better accuracy. The Multidimensional Scaling Map (MDS-MAP) proposed in [36] is an example. MDS-MAP originated from a data analytical technique by displaying distance-like data in geometrical visualization. It computes the shortest paths between all pairs of nodes to build a distance matrix and then applies the classical Multidimensional Scaling (MDS) to this matrix to retain the first two largest eigenvalues and eigenvectors to a 2D relative map. After that, with three given anchors, it transforms the relative map into an absolute map based on anchors’ absolute locations. There are some variances of MDS-MAP such as centralized method: MDS-MAP(C), and distributed one: MDS-MAP (P). But, in the distributed method, to get the absolute location, nodes need global information about the sub-network’s map that contains at least three anchors.

2.3. Localization using neural networks and self-organizing maps

Neural networks have not been used extensively in this area. There has been some research conducted in [37]. However, in that research, the authors restricted themselves to comparing Recurrent Neural Networks (RNN) to the Kalman Filter. In [38], the authors showed that a Multi-Layer Perceptron (MLP) neural network can be used for localization, and that its performance exceeds that of the Position-Velocity (PV) and Position-Velocity-Acceleration (PVA) variants of the Kalman Filter. In [39], a new localization scheme is proposed based on a Support Vector Machine (SVM). The authors have contributed another machine learning method to the localization problem, and proved the upper bound error of this method. These first approaches that used neural networks in the localization process in WSNs are promising and may lead to faster and more accurate localization.

A Self-Organizing Map (SOM) [40] or Self-Organizing Feature Map (SOFM) is a type of artificial neural network [41] that is trained using unsupervised learning to produce a low-dimensional, typically two-dimensional, discretized representation of the input space of the training samples, called a map. SOMs are different from other artificial neural networks in the sense that they use a neighborhood function to preserve the topological properties of the input space. This makes SOMs useful for visualizing low-dimensional views of high-dimensional data, akin to multidimensional scaling [42].

Regarding the localization based on SOMs, some researchers have employed them directly or with some modification. SOMs have been used to implement localization schemes for mobile robots in unknown environments [43, 44]. The SOM, initially trained with information collected by on-board sensors during the exploration phase, is then used as a virtual map to translate new sensor readings into grid positions or to recognize different environments. The method presented in [45] employed the classical SOM to the localization problem. This method uses centralized implementation and requires thousands of learning steps for convergence to network topology. The authors also realized that this method is good for small and medium size

networks of up to 100 nodes. In [46], a distributed localization scheme is proposed based on SOM. In [47], the authors proposed another version of distributed localization based on SOM. In [48], they employed another deduced SOM version. But, this method still needs too many iterations, at least 4000, to converge to the actual topology with a relatively low accuracy.

In another work [49], the authors use SOM to track a mobile robot with the utilization of surrounding environments using readings of sensor data. In [50], another version of SOM is used to implement the localization in wireless sensor networks. Their model is based on the assumption that network nodes can sense a common phenomenon, e.g., acoustic or seismic, at synchronized time steps. A further assumption is that the correlation between sensor readings is a function only of the distance between nodes. Under these conditions, sensor readings from all nodes are first accumulated to form the training set, and then, after the SOM model has been trained, are used to sort the nodes according to their proximity to a set of virtual sensors placed on a regular grid.

These SOM-based algorithms led to acceptable error range and accuracy, but still need a huge number of iterations to converge. Also, they used the classical SOM update method, which may be not suitable in some cases, and is applicable only to small networks.

In this paper, we also use SOMs in the localization process, but enhance the existing techniques and get benefit from the connectivity and neighboring nodes' information to start the learning stage with meaningful initialization, and thus fasten the process. Moreover, the SOM update function will be modified to be suitable for different network topologies as well as different network sizes.

3. The proposed algorithm: Distributed Localization using Self Organizing Maps (DLSOM)

3.1. Anchors number and placement

The number and placement of anchors affect the accuracy of node localization algorithms to a certain extent. Substantial amount of anchors are required to maintain the accuracy for distributed algorithms based on multilateration, in which nodes estimate their positions as the average of the received positions from anchors and neighbors. Theoretically, more anchors bring higher location accuracy. However, too many anchors cause high calculation complexity and energy consumption.

Many previous studies found that the optimal number of anchors to be selected in most distributions of WSNs ranges from 3 to 6. In multilateration-based algorithms, like our algorithm, using four anchors gives satisfying accuracy with very slight difference of using more anchors, taking into account the calculation complexity and energy consumption.

In [51], the number of anchors is optimized through simulation with Matlab. The localization algorithm is simulated with different number of anchors (3, 4, 6, 8, 10, 20, and 50). The correction quality of multilateration stagnates when more than six anchors are used. To keep low complexity and low energy consumption, four is chosen as the optimal number of anchors. Also, in the experiments of [52], they tried from 3 to 10 anchors on anisotropic networks and found that four anchors usually give the best results. Another benefit of using just four anchors for multilateration is that the communication cost is much lower than using many anchors. The delay in getting the distance information is also smaller. The information locality enables the method to scale to large-size networks.

For the selection of anchors' positions, also many studies showed that the performance is better when anchors are uniformly distributed along the perimeter of the network. The nodes with high correlations with other nodes should be selected [53-55]. In [52], experiments showed that placing the four anchors randomly gives slightly worse solutions than using the four outer anchors at the network perimeter. Using the four inner anchors at the four centers gives the worst solution. In this paper, we tried to select the anchors at the perimeter and at the four centers of the network. Extensive simulations using the two distributions showed that selecting the anchors at the network perimeter gives better accuracy. Thus, four anchors distributed at the network perimeter are selected in the simulations of our algorithm.

3.2. DLSOM methodology

Given a wireless network with randomly distributed nodes with N nodes in a specified area, $L \times L$, and a determined radio range, R , also called communication range, in which the nodes can maximally communicate with each other, the anchor nodes are selected to be at the four ends, i.e., the perimeter, of the network topology. The algorithm is divided into two main stages, the initialization stage and the SOM learning stage.

In the initialization stage, the anchors begin to broadcast their well-known locations to their neighbors, nodes within their communication range, as well as the position of these anchors with regard to the network topology (top-right, top-left, bottom-right, bottom-left). This information can be usefully used by the neighbors to estimate their initial locations.

The x -coordinate, x_{anchTR} , and the y -coordinate, y_{anchTR} , of the top-right anchor are initialized to the maximum x -coordinate and the maximum y -coordinate for the whole network nodes, respectively. For every neighbor node of this anchor, its x -coordinate and y -coordinate are initialized to random values in the intervals $[x_{anchTR} - R, x_{anchTR}]$ and $[y_{anchTR} - R, y_{anchTR}]$, respectively.

The x -coordinate, x_{anchTL} , and the y -coordinate, y_{anchTL} , of the top-left anchor are initialized to the minimum x -coordinate and the maximum y -coordinate for the whole network nodes, respectively. For every neighbor node of this anchor, its x -coordinate and y -coordinate are initialized to random values in the intervals $[x_{anchTL}, x_{anchTL} + R]$ and $[y_{anchTL} - R, y_{anchTL}]$, respectively.

The x -coordinate, x_{anchBR} , and the y -coordinate, y_{anchBR} , of the bottom-right anchor are initialized to the maximum x -coordinate and the minimum y -coordinate for the whole network nodes, respectively. For every neighbor node of this anchor, its x -coordinate and y -coordinate are initialized to random values in the intervals $[x_{anchBR} - R, x_{anchBR}]$ and $[y_{anchBR}, y_{anchBR} + R]$, respectively.

The x -coordinate, x_{anchBL} and the y -coordinate, y_{anchBL} , of the bottom-left anchor are initialized to the minimum x -coordinate and the minimum y -coordinate for the whole network nodes, respectively. For every neighbor node of this anchor, its x -coordinate and y -coordinate are initialized to random values in the intervals $[x_{anchBL}, x_{anchBL} + R]$ and $[y_{anchBL}, y_{anchBL} + R]$, respectively.

Now, the estimated initial locations of the anchors' neighbors are transmitted to the unknown inner neighbor nodes. These neighbor nodes will actually receive many estimated locations according to the number of neighbors. Each inner node, v , estimates its initial location by averaging the received locations according to the following equation:

$$(x_v, y_v) = \left(\frac{\sum_{i=1}^{N_v} x_i}{N_v}, \frac{\sum_{i=1}^{N_v} y_i}{N_v} \right) \quad (1)$$

where (x_v, y_v) are the coordinates of node v , and N_v is the number of its neighbors.

As a final step of the initialization stage, the rest of the network nodes that did not receive any estimated location, due to their wide distance from the network nodes, i.e., out of range of any network node, will estimate their initial locations randomly since there is no knowledge of any neighbor information.

After completing the initialization stage, each node forwards its estimated location to all of its neighbors in preparation to the learning stage. Let the actual and estimated locations of a wireless node be denoted by w_i^a and w_i^e , respectively, $i = 1, 2, 3, \dots, N$. As each node forwarded its estimated initial location to all of its neighbors, it also knows the estimated locations of its neighbors, denoted by $w_{i,j}^e$, $j = 1, 2, 3, \dots, N_i$, where N_i is the number of neighbor nodes to the node with location w_i^e . Now, the node with the location w_i^e plays as the winning neuron to the region formed by the neighbors of that node. Because each node knows its neighbors

within the communication range and due to the distributed nature of the algorithm, each node can be considered as the best matching unit to the set of its neighbors and performs the location updates of its neighbors. The following steps represent the learning stage of the algorithm:

Step 1. Based on SOM, the node with weight (location) $w_i^e(t)$ updates every weight (location), $w_{i,j}^e(t)$, of a neighbor node according to the following formula:

$$w_{i,j}^e(t+1) = w_{i,j}^e(t) + \Delta w_{i,j}^e(t) \quad (2)$$

$$\text{where } \Delta w_{i,j}^e(t) = \alpha(t)(w_i^e(t) - w_{i,j}^e(t)) \quad (3)$$

and α is a learning rate exponential decay function defined as:

$$\alpha(t) = \exp\left(\frac{-(t+1)}{T}\right) \quad (4)$$

where t is t^{th} time step of the total number of learning steps, T .

Step 2. The node with location $w_i^e(t)$ now transmits its neighbor location updates to all of its neighbors, and on the other hand, receives similar updates from its neighbors as $w_{j,i}^e(t+1)$, $j = 1, 2, 3, \dots, N_i$.

Step 3. The node with location $w_i^e(t)$ calculates its new estimated location by averaging the current location and the received updates from neighbors according to the following equation:

$$w_i^e(t+1) = \frac{1}{N_i + 1} (w_i^e(t) + \sum_{j=1}^{N_i} w_{j,i}^e(t+1)) \quad (5)$$

Step 4. The node re-forwards its new estimated location, $w_i^e(t+1)$, to its neighbors.

The previous four steps are repeated T times, or until convergence.

4. Experimentation

4.1. Experimental environment

Our DLSOM algorithm is implemented and tested using Matlab 7.9 software. The code is run on a desktop PC with Intel Pentium 4.0, 2.6 GHz CPU, and 512 KB RAM. The algorithm is run with the following settings:

- Wireless network topology generation and deployment: The network is generated randomly based on the following varying parameters:
Number of nodes, N : varies from 10 to 100 nodes.
Communication range, R : varies from 1 km to 4 km.
Deployment area, $L \times L$: 10 km \times 10 km is used.
- Initialization stage: The initial estimated locations of nodes are calculated by assuming that four anchors are selected at the perimeter of the network topology (top-right, bottom-right, top-left, bottom-left).

- SOM learning stage: The final estimated locations are calculated according to the following SOM parameters:
 Maximum number of iterations, T : varies from 25 to 50.
 Learning rate, α : a decay exponential function, of the current iteration and the maximum number of iterations, is used.
 Weight updating formulae, see Eqns. 2 to 5.

The evaluation parameter is the average error between the estimated locations resulted from the DLSOM algorithm and the actual locations, defines as:

$$Error(R) = \frac{1}{N \times R} \sum_{i=1}^N |w_i^e - w_i^a| \quad (6)$$

where R is the communication range, N is the number of nodes, w_i^e and w_i^a are the i^{th} node estimated and actual locations, respectively.

4.2. A 10-nodes wireless network

For the communication range 1 km, due to the low connectivity, the anchors almost have no neighbors. Most of the nodes have no neighbors; this forces the nodes to estimate their locations randomly since no neighborhood information is available. The difference between the actual and initial locations of network nodes is very high. Almost all nodes have estimated their locations randomly. As a result of the DLSOM learning, the accuracy of the final estimated locations is not very high. Also, the number of iterations needed to reach convergence is considerably high; took about 50 iterations.

For the communication range 2 km, due to the low connectivity and low number of nodes, the anchors almost have no neighbors. Here, the difference between the actual and initial locations of nodes is much smaller. This is because some of the inner nodes have neighbors and get advantage of the estimated locations. As a result of the DLSOM learning, the accuracy of the final estimated locations is also much higher, about twice, and also, the number of iterations needed to reach convergence is lower; took about 35 iterations.

For the communication range 4 km, all anchors have neighbors. Most of the nodes have neighbors. The difference between the actual and initial locations of network nodes is considerably small. As a result of the DLSOM learning, the accuracy of the final estimated locations is considerably high. Most of the nodes have estimated locations with low difference from the actual locations. Also, the number of iterations needed to reach convergence is very low; took about 25 iterations.

4.3. A 50-nodes wireless network

For the communication range 1 km, the anchors almost have no neighbors due to low radio range. Most of the nodes estimated their initial locations randomly due to low connectivity. Also, some of the inner nodes that have neighbors shared the same estimated location. As a result of the DLSOM learning, the accuracy of the final estimated locations is relatively low. The final estimated locations for most of the nodes have noticeable difference from the actual locations. Also, the number of iterations needed to reach convergence is relatively high in contrast with the 10-nodes wireless network with $R = 4$ km. Here, it took about 35 iterations to converge.

For the communication range 2 km, all four anchors have considerable number of neighbors that can estimate their initial locations easily. For this communication range, most of the nodes have neighbors. This helps the nodes to estimate their locations based on neighborhood information and totally get benefit from the

proposed algorithm. As a result of the DLSOM learning, the accuracy of the final estimated locations is higher, and also, the number of iterations needed to reach convergence is much lower; took about 25 iterations. This is due to the higher connectivity and the correlation between nodes.

For the communication range 4 km, all anchors have large number of neighbors that can estimate their initial locations based on received anchors' locations. All of the nodes have neighbors, which helps the nodes to estimate their locations based on neighborhood information. But, because of the existence of the averaging step during the initialization stage, and because of the similarity of neighbors for some nodes, the final estimated locations for these nodes are nearly the same. This leads to agglomeration which is a problem that occurs in most of multiteration-based methods, in which a node can estimate its location by averaging the received locations. This is due to the accumulation of the error in previous estimations. As a result of the DLSOM learning, the accuracy of the final estimated locations is lower than the previous case, contrary to the expectations. This is due to the agglomeration happening during the initialization stage. The number of iterations is about 28.

4.4. A 100-nodes wireless network

For the communication range 1 km, anchors have low number of neighbors due to low radio range. Some of the inner nodes, that have neighbors, estimate their locations based on received data from adjacent nodes. As a result of the DLSOM learning, the final estimated locations showed that the accuracy of the result is relatively low; many nodes shared the same location, and, also, the number of iterations needed to reach convergence is high; took about 45 iterations. Although the communication range is very low, the problem of agglomeration slightly occurred and the low communication range influenced the accuracy of the algorithm. This is due to the large number of nodes deployed in a limited area.

For the communication range 2 km, most anchors have neighbors that can estimate their locations based on anchors' sent data. Here, the problem of agglomeration occurred; this is due to the large number of nodes. As a result of the DLSOM learning, the final estimated locations showed that the accuracy of the result is higher than the previous case and, also, the number of iterations needed to reach convergence is lower; took about 36 iterations.

For the communication range 4 km, anchors have high number of neighbors that can estimate their initial locations making advantage of received data. Again, the agglomeration problem occurred with a high degree due to the high communication range and the large number of nodes. As a result of the DLSOM learning, the accuracy of the result is relatively low; many nodes share the same location and have noticeable difference from the actual locations, and the number of iterations is about 40.

Table 1. Simulation results for different sets of parameters

Number of nodes	Radio range	Number of iterations	Avg. error
10	1	50	0.450
10	2	35	0.341
10	4	25	0.264
50	1	35	0.402
50	2	25	0.228
50	4	28	0.296
100	1	45	0.390
100	2	36	0.277
100	4	40	0.317

Table 1 summarizes the results of the three network types that have been used in simulations. For radio range 1 km, by increasing the number of nodes, the average error decreases. This can be explained as: by increasing the number of nodes with a low radio range, the neighborhood information increases slightly, and thus, the DLSOM algorithm performs better. For radio range 2 km, by increasing the number of nodes, the average error decreases and then increases. This can be explained as: by increasing the number of nodes with a medium radio range, the neighborhood information increases, and thus, the DLSOM algorithm performs better. But, by deploying a large number of nodes in a limited area with this radio range, the agglomeration problem occurs and the DLSOM performance decreases slightly. For radio range 4 km, by increasing the number of nodes, the average error increases. This can be explained as: by increasing the number of nodes with a large radio range in a limited area, the agglomeration problem occurs and the DLSOM performance decreases slightly. The best case is when the number of nodes is 50 with radio range of 2 km, in which case the average estimated error is about 0.228 whereas the worst case is when the number of nodes is 10 with radio range of 1 km, in which the average estimated error is about 0.45.

The effect of radio range variance can be illustrated by results of Table 1, also. For low number of nodes, by increasing the radio range, the average error decreases. This is due to the neighborhood information support that increases with high radio ranges. For medium number of nodes, by increasing the radio range, the average error decreases and then increases. This is due to the occurrence of the agglomeration problem. Also, for large number of nodes, by increasing the radio range, the average error decreases and then increases because of the occurrence of the agglomeration problem.

4.5. Placing the anchors at the four centers

To see the effect of changing the four anchors placement on DLSOM, we tried to put the anchors at the four centers of the network. Table 2 shows the average error results by applying this change, placement, on the same experimental cases described earlier. In this case, the average error is much larger compared to the results of Table 1. This can be explained as: the method used to estimate the locations of anchors' neighbors, explained earlier, is more accurate. For all the anchors at the centers of the network, the estimation equation is:

$$(x, y) = (\text{random}([x_{anch} - R, x_{anch} + R]), \text{random}([y_{anch} - R, y_{anch} + R])) \quad (7)$$

where (x, y) are the coordinates of an anchor neighbor, (x_{anch}, y_{anch}) are the coordinates of any of the four anchors, and R is the communication range. Obviously, the interval of randomization for location estimation is unified and larger in contrast with the four different intervals used before. Thus, based on the results of this experiment and previous studies, the anchors placed at the perimeter of the network yields better accuracy than that at the four centers of the network.

Table 2. Simulation results for placing anchors at the four centers

Number of nodes	Radio range	Avg. error
10	1	0.471
10	2	0.318
10	4	0.683
50	1	0.721
50	2	0.528
50	4	0.693
100	1	0.689
100	2	0.657
100	4	0.812

4.6. Time analysis

The time required to execute the DLSOM algorithm has been measured for each of the simulation experimental cases. Execution time increases linearly by increasing the number of nodes. The time spent in the initialization stage is constant and short. This is because this stage executes for one time (not iterative) and the calculations are fast and simple. Most of the time is spent in the learning stage (iterative process), in which the nodes transmit and receive location updates to neighbors at each iteration. This time, of course, increases by increasing the number of nodes. For example, for radio range 1 km, the time is about 56 seconds. About 12 seconds only are spent in the initialization stage and 44 seconds in the learning stage. Also, the time increases linearly by increasing the radio range. This is because by increasing the radio range, more transmission and receiving processes are done by the increment of neighboring nodes. These processes run most of the time.

Although the number of iterations decreases by number of nodes and radio range increment, the time spent increases. This slight increment does not affect the performance of the algorithm; DLSOM is considered to be fast in all cases.

4.7. Comparisons with other algorithms

The DLSOM algorithm is compared to many similar algorithms that are used to localize the nodes in ad-hoc wireless networks. The comparison parameter chosen is the average error in a unified set of simulation parameters.

For 50 wireless nodes randomly deployed in an area of $10 \text{ km} \times 10 \text{ km}$ with a radio range of 2 km and four anchors selected at the network perimeter, the average error resulted by applying the SOM, MDS-MAP, and DLSOM is 0.46, 0.38, and 0.27, respectively. These results are the average of 30 different trials of the same set of parameters. Obviously, the DLSOM algorithm has the least average error with a noticeable difference. It is worth mentioning that especially in this case, 50 nodes and range of 2 km, the DLSOM performs the best.

The average error calculated after applying DV-HOP, SOM, and DLSOM, is 0.50, 0.35, and 0.30, respectively, where 100 wireless nodes are randomly deployed in an area of $10 \text{ km} \times 10 \text{ km}$ with a radio range of 2 km and four anchors selected at the ends of the network. These results are the average of 50 different trials of the same set of parameters. Again, the DLSOM algorithm has the least average error over the other algorithms with a noticeable difference.

For low radio ranges, low number of nodes, and irregular network topology (random), some algorithms proved their high performance over the others. In [45], a classical SOM localization algorithm is used and produced accurate results in comparison with the others. We have implemented the classical SOM algorithm used in [45] on the same hardware/software environment that we used to implement our algorithm. The tested original network has 10 wireless nodes deployed randomly in an area of $10 \text{ km} \times 10 \text{ km}$ with a radio range of 2 km. The average error rates of localized SOM and DLSOM are 0.36 and 0.38, respectively. Moreover, the classical SOM algorithm is centralized while the DLSOM is distributed. Hence, the wireless network nodes' computation overhead is reduced significantly, and also the number of iterations taken in the DLSOM is about 35 which is very small compared to the thousands of iterations taken by the classical localized SOM. These results are the average of 50 different trials of the same set of parameters.

5. Conclusion

In this paper, a SOM-based distributed localization algorithm is proposed (DLSOM). The main objective of this algorithm is to calculate the locations of nodes in wireless sensor networks. The intelligent SOM neural networks are selected due to their multiple characteristics over other types of neural networks. One of the most important characteristics of SOMs is their unsupervised training fashion, in which no reward or cost functions are needed. Moreover, the arrangement of neurons into a grid increases the accuracy of the results.

The proposed localization algorithm aimed to get benefit from the neighborhood information that can be gathered fast and easy by the wireless nodes; each node knows its neighbors based on the communication radio range. Thus, no communication overhead occurs and the usage of this information in the initialization stage of the algorithm significantly helped DLSOM to begin with useful initial data to be used in the learning stage. Hence, the learning time and the number of iterations took by DLSOM to reach stable results have been noticeably decreased. Also, the accuracy of the results increased compared to results of previous related algorithms.

The experimentation and simulation results proved the effectiveness of the DLSOM algorithm in different simulation parameters. The criterion that has been used to evaluate and compare the performance of the algorithm against other algorithms is the average error. The average error of the DLSOM is found to be the least in most cases compared to some related algorithms that proved their accuracy in this field. The total average error, calculated by averaging errors using all experiments, is about 30%. This percentage is much lower than that of other algorithms. Moreover, the number of iterations needed in DLSOM ranges from 25 to 50. This number is significantly much smaller in comparison with the hundreds to thousands of iterations needed by other algorithms which reduces the learning time substantially.

Experimentations showed that the case in which the DLSOM performs the best is with 50 wireless nodes and radio range of 2 km. The performance gets worse and worse by increasing the number of nodes in a limited area and also increasing the radio range with this large number of nodes. But in all cases, the algorithm proved its effectiveness over the previous contributions either in low or high number of nodes. Referring to the best case mentioned before, this can be considered as the most important advantage of the algorithm because most of the real wireless networks deployed in a limited area (10 km \times 10 km) usually contain no more 50 nodes with a communication range of 2 km as an average.

During the experimentation and validation of our proposed algorithm, the problem of agglomeration has occurred in some cases, in which the nodes have approximately the same initial estimated location due to the averaging step in the initialization stage. These cases include the high number of network nodes deployed in a limited area and have a relatively high communication radio range. This may be solved by selecting the nearest three, or more based on the total number of nodes, neighbors to be averaged to get the estimated location. In this way, the nodes that have the same set of neighbors will be enforced to choose the nearest subset and hence will get a distinct estimated location. Another suggested solution is to replace the averaging process step with a more complex and distinguishing mathematical process, such as trilateration method used in GPS systems to locate some wireless node, in which three known-location nodes are used to estimate the unknown node location with a series of complex mathematical equations. On the other hand, trilateration could be more accurate to calculate the initial estimated locations of wireless nodes.

In this paper, we investigated a proposed distributed localization algorithm that has been applied on static wireless sensor networks, in which the nodes have no movement. For mobile wireless networks, in which the nodes move with specific parameters such as motion speed and motion direction, the proposed localization algorithm can be modified to be applied on these networks. This can be done by refreshing the set of nodes' neighbors periodically and using the most updated neighbors list in the learning stage to get the most approximate location of the node. The period of refreshing is determined based on the motion speed of the wireless nodes in the network. In mobile networks, the accuracy of the algorithm is supposed to be less than static networks due to the continuous movement and different neighborhood information that change rapidly, and hence influence the principle of the algorithm.

Acknowledgements

We thank anonymous referees for their constructive comments.

References

- [1] K. Delin, "The Sensor Web: A Macro-Instrument for Coordinated Sensing", *Sensors*, vol. 2, pp. 270-285, July 2002.
- [2] J. Yick, B. Mukherjee, and D. Ghosal, "Wireless sensor network survey", *Computer Networks*, vol. 52, no. 12, pp. 2292-2330, April 2008.
- [3] B. Karp and H. Kung, "GPSR: greedy perimeter stateless routing for wireless networks", in *Mobile computing and networking conference*, Boston, MA, USA, 2000, pp. 243-254.
- [4] M. Hefeeda and M. Bagheri, "Wireless sensor networks for early detection of forest fires", in *IEEE International Conference on Mobile Adhoc and Sensor Systems*, Pisa, 2008, pp. 1-6.
- [5] S. Pace, G. Frost, I. Lachow, D. Frelinger, D. Fossum, D. K. Wassem, and M. Pinto, "GPS history chronology and budgets", in *The global positioning system*, Santa Monica, RAND Corporation, 1995, Appendix B.
- [6] S. Pandey and P. Agrawal, "A Survey on Localization Techniques for Wireless Networks", *Journal of the Chinese Institute of Engineers*, vol. 29, no. 7, pp. 1125-1148, 2006.
- [7] L. Mo, and L. Yunhao: Rendered path, "Range-free localization in anisotropic sensor networks with holes", *IEEE/ACM Trans. Netw.*, vol. 18, no. 1, pp. 320-332, 2010.
- [8] Zhetao Li, Renfa Li, Yehua Wei, and Tingrui Pei, "Survey of Localization Techniques in Wireless Sensor Networks", *Information Technology Journal*, vol. 9, pp. 1754-1757, Sep. 2010.
- [9] P. Juang, H. Oki, Y. Wang, M. Martonosi, and L. S. Peh, "Energy-efficient computing for wildlife tracking: Design tradeoffs and early experiences with ZebraNet", *ACM*, vol. 37, pp. 96-107, 2002.
- [10] L. Yihan, S. S. Panwar, M. Shiwen, S. Burugupalli, and L. Jong Ha, "A mobile ad hoc biosensor network", in *IEEE International Conference on Communications*, NY, USA, 2005, vol. 2, pp. 1241-1245.
- [11] R. Peng and M. L. Sichitiu, "Localization of wireless sensor networks with a mobile beacon", in *IEEE Conference on Mobile Ad-hoc and Sensor Systems*, NC, USA, 2004, pp. 174-183.
- [12] R. Poovendran, C. L. Wang, and S. Roy, "Secure Localization and Time Synchronization for Wireless Sensor and Ad Hoc Networks", Springer, 2007.
- [13] B. H. Wellenhoff, H. Lichtenegger, and J. Collins, "Global Positioning System: Theory and Practice", 5th edition, Springer, 2001.
- [14] A. Savvides, C. C. Han, and M. B. Strivastava, "Dynamic finegrained localization in ad-hoc networks of sensors", in *Mobile Computing and Networking conference*, New York, NY, USA, 2001, pp. 225-246.
- [15] D. Niculescu and B. Nath, "Ad hoc positioning system (APS) using AOA", *IEEE Computer*, vol. 3, pp. 1734-1743, April 2003.
- [16] N. Patwari, A. O. Hero III, M. Perkins, N. S. Correal, and R. J. O'Dea, "Relative location estimation in wireless sensor networks", *IEEE Transactions on Signal Processing*, pp. 2137-2148, Aug. 2003.
- [17] Nalini Vasudevan, "Node Localization in Wireless Ad-Hoc Sensor Networks", R.V. College of Engineering, Bangalore, Tech. Rep. 52, June 2005.
- [18] R. Want, A. Hopper, V. Falcao, and J. Gibbons, "The active badge location system", Cambridge University, Technical Report 92.1, 1992.
- [19] N. Priyantha, "Providing precise indoor location information to mobile devices", M.S. thesis, Massachusetts Institute of Technology, MIT, 2001.
- [20] A. Ward, A. Jones, and A. Hopper, "A new location technique for the active office", *IEEE Personnel Communications*, vol. 4, pp. 42-47, Oct. 1997.
- [21] A. Harter, A. Hopper, P. Steggles, A. Ward, P. Webster, "The anatomy of a context-aware application", in *Proceedings of the 5th Annual ACM/IEEE International Conference on Mobile Computing and Networking*, New York, NY, USA, 1999, pp. 798-815.

- [22] A. Smith, H. Balakrishnan, M. Goraczko, and N. Priyantha, "Tracking moving devices with the cricket location system", in ACM conference on Mobile systems, New York, NY, USA, 2004.
- [23] D. Moore, J. Leonard, D. Rus, and S. Teller, "Robust distributed network localization with noisy range measurements", ACM Proceedings of the 2nd international conference on Embedded networked sensor systems, New York, NY, USA, 2004.
- [24] Greg Welch, Gary Bishop, Leandra Vicci, Stephen Brumback, Kurtis Keller, and D'nardo Colucci, "High-performance wide-area optical tracking", *Teleoper. Virtual Environ*, vol. 10, pp. 1-21, Feb. 2001.
- [25] C. Alippi, A. Mottarella, and G. Vanini, "A RF map-based localization algorithm for indoor environments", *IEEE International Symposium on Circuits and Systems*, vol. 1, pp. 652-655, May 2005.
- [26] J. Hightower, G. Borriello, and R. Want. SpotON, "An indoor 3D location sensing technology based on RF signal strength", CSE, Technical Report, 2000.
- [27] K. Lorincz and M. Welsh, "A robust decentralized approach to RF-based location tracking", Technical Report TR-04-04, 2004.
- [28] L. M. Ni, Y. Liu, Y. C. Lau, and A. P. Patil Landmarc, "Indoor location sensing using active RFID", in *IEEE International Conference on Pervasive Computing and Communications*, 2003, pp. 407-415.
- [29] F. Dellaert, D. Fox, W. Burgard, and S. Thrun, "Monte Carlo Localization for Mobile Robots", *IEEE International Conference on Robotics and Automation*, 1999, vol. 2, pp. 1322-1328.
- [30] L. Hu and D. Evans, "Localization for mobile sensor networks", *International Conference on Mobile Computing and Networking, USA*, 2004, pp. 1167-1180.
- [31] A. Baggio and K. Langendoen, "Monte Carlo localization for mobile wireless sensor networks", *Ad Hoc Networks*, vol. 6, pp. 718-733, July 2007.
- [32] N. Bulusu, J. Heidemann, and D. Estrin, "GPS-less low-cost outdoor localization for very small devices", *IEEE Personal Communications Magazine*, vol. 7, pp. 28-34, Oct. 2000.
- [33] D. Niculescu and B. Nath, "DV Based Positioning in Ad Hoc Networks", *Journal of Telecommunication Systems*, vol. 22, pp. 267-280, 2003.
- [34] R. Nagpal, H. Shrobe, and J. Bachrach, "Organizing a global coordinate system from local information on an ad hoc sensor network", In *Proceedings of the 2nd international conference on Information processing in sensor networks (IPSN'03)*, Springer-Verlag, Berlin, Heidelberg, 2003, pp. 333-348.
- [35] D. Niculescu and B. Nath, "Ad-hoc positioning systems (APS)," in *Proceedings of IEEE Global Telecommunications Conference, San Antonio, USA, 2001*, vol. 5, pp. 2926-2931.
- [36] Y. Shang, W. Ruml, Y. Zhang, and M. Fromherz, "Localization from connectivity in sensor networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 15, pp. 961-974, 2004.
- [37] S. K. Chenna, Y. Kr. Jain, H. Kapoor, R. S. Bapi, N. Yadaiah, A. Negi, V. Seshagiri Rao, and B. L. Deekshatulu, "State estimation and tracking problems: A comparison between Kalman filter and recurrent neural networks", *International Conference on Neural Information Processing, Berlin, 2004*, pp. 275-281.
- [38] A. Shareef, Yifeng Zhu, Mohamad Musavi, and Bingxin Shen, "Comparison of MLP Neural Networks and Kalman Filter for Localization in Wireless Sensor Networks", in *Proceedings of 19th IASTED International Conference on Parallel and Distributed Computing Systems*, November, 2007, Cambridge, Massachusetts, USA, pp. 323-330.
- [39] D. A. Tran and T. Nguyen, "Localization in wireless sensor networks based on support vector machines", *IEEE Transactions on Parallel and Distributed Systems*, vol. 7, pp. 1278-1290, 2008.
- [40] T. Kohonen, "Self-organized formation of topologically correct feature maps", *Biological Cybernetics*, vol. 43, pp. 59-69, 1982.
- [41] S. Haykin, "Neural Networks", Prentice Hall, 2nd edition, 1998.
- [42] T. Kohonen, "Self-Organizing Maps", Springer, 3rd edition, 2001.
- [43] J. Janet, R. Gutierrez, T. Chase, M. White, and J. Sutton., "Autonomous mobile robot global self-localization using Kohonen and region-feature neural networks", *Journal of Robotic Systems*, vol. 3, pp. 504-509, 1997.

- [44] U. Gerecke and N. Sharkey, "Quick and dirty localization for a lost robot: Computational Intelligence in Robotics and Automation", IEEE International Symposium, 1999, vol. 2, pp. 15-26.
- [45] G. Giorgetti, S. K. S. Gupta, and G. Manes, "Wireless localization using self-organizing maps", In Proceedings of the 6th international conference on Information processing in sensor networks (IPSN '07), ACM, New York, NY, USA, 2007, pp. 293-302.
- [46] S. Asakura, D. Umehara, and M. Kawai, "Distributed location estimation method for mobile terminals based on SOM algorithm", IEICE Transactions on Communications, vol. 85, pp. 1042-1050, 2002.
- [47] J. Hu and G. Lee, "Distributed localization of wireless sensor networks using self-organizing maps", IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems, Aug. 2008, pp. 284-289.
- [48] J. Sum, C. S. Leung, L.W. Chan, and L. Xu, "Yet another algorithm which can generate topograph map," IEEE Transactions on Neural Networks, vol. 6, pp. 548-559, 1997.
- [49] J. A. Janet, R. Gutierrez, T. A. Chase, M. W. White, and J. C. Sutton III, "Autonomous mobile robot global self-localization using Kohonen and region-feature neural networks", Journal of Robotic Systems, vol.14, pp. 263-282, 1997.
- [50] E. Ertin and K. L. Priddy, "Self-localization of wireless sensor networks using self organizing maps", Intelligent Computing: Theory and Applications III, vol. 5803, pp. 138-145, March 2005.
- [51] Xinwei Wang, Ole Bischoff, Rainer Laur, and Steffen Paul, "Localization in Wireless Ad-hoc Sensor Networks using Multilateration with RSSI for Logistic Applications", Proceedings of the EuroSensors XXIII conference, 2009, pp. 461-464.
- [52] Yi Shang, Hongchi Shi, A. A. Ahmed, "Performance study of localization methods for ad-hoc sensor networks", Mobile Ad-hoc and Sensor Systems, IEEE International Conference, Oct. 2004, pp. 184-193.
- [53] Mehdi Vojdani and Mehdi Dehghan, "Localization in Anchor Less Wireless Sensor Network", International Association of Computer Science & Information Technology (IACSIT), vol. 2, Jan. 2012.
- [54] Zhiheng Xie, Mingyi Hong, Hengchang Liu, Jingyuan Li, Kangyuan Zhu, and John Stankovic, "Quantitative uncertainty-based incremental localization and anchor selection in wireless sensor networks", In Proceedings of the 14th ACM international conference on Modeling, analysis and simulation of wireless and mobile systems (MSWiM '11), ACM, New York, NY, USA, 2011, pp. 417-426.
- [55] G. Han, J. Jiang, L. Shu, Y. Xu, and F. Wang, "Localization Algorithms of Underwater Wireless Sensor Networks: A Survey", Sensors, Vol. 12, no. 2, pp. 2026-2061, 2012.

Authors' Profiles

Ibrahim S. I. Abuhaiba is a professor at the Islamic University of Gaza, Computer Engineering Department. He obtained his Master of Philosophy and Doctorate of Philosophy from Britain in the field of document understanding and pattern recognition. His research interests include artificial intelligence, computer vision, image processing, document analysis and understanding, pattern recognition, information security, and computer networks. Prof. Abuhaiba published tens of original contributions in these fields in well-reputed international journals and conferences.

Amina Y. Al-Sallut received her B.Sc. degree in computer engineering, Islamic University of Gaza, in 2006, and master degree in computer engineering, Islamic University of Gaza, in 2012. Her research interests include information security, computer networks, and artificial intelligence.