International Journal of Grid Distribution Computing
Vol.8, No.2 (2015), pp.79-90
http://dx.doi.org/10.14257/ijgdc.2015.8.2.08

A Novel Distributed Clustering-based MDS Algorithm for Nodes Localization in WSNs

Guoyong Dai[1,2], Chunyu Miao[1], Yidong Li[1], Keji Mao[1] and Qingzhang Chen[1*]

[1]*College of Computer Science and Technology, Zhejiang University of Technology, Hangzhou 310023, China*
[2]*College of Information Science and Technology, Zhejiang Shuren University, Hangzhou 310014, China*

Abstract

Localization of sensors nodes is a key and fundamental issue in wireless sensor networks due to random deployment. In this paper, we propose a tree based clustering (TBC) multidimensional scaling algorithm for wireless sensor networks with the purpose of overcoming the shortage of classical MDS algorithms in its localization accuracy and computing complexity. Clustering is adopted to degrade the problem scale in our approach and a moderate number of common nodes between clusters are kept during clustering. Inner cluster local coordinates are calculated and then mapped into global coordinates according to the tree structure formed by clustering. Simulations on MATLAB are conducted and the results show that the proposed algorithm has better localization coverage and higher accuracy than the traditional MDS based algorithms.

Keywords: *WSN, Multidimensional Scaling, Clustering, Tree*

1. Introduction

Wireless sensor networks (WSNs) have attracted a great deal of research interests due to their potential applications in military, environmental monitoring, medical care, battlefield surveillance, *etc*., [1-4]. Especially with the blossom of the Internet of Things (IOT) and cloud computing in recent years, WSNs have been fundamental infrastructures of IOT and cloud computing for data gathering. The initial locations of sensor nodes are uncertain since nodes are randomly scattered in the target area in many application scenarios [2,4-6]. Besides, many subsequent environmental factors (such as winds, streams *etc*.,) may make the locations of nodes change after deployment. Locations of sensor nodes are uncontrollable to some extent. However, many applications require the sensors to be aware of their geographical position either in absolute or relative form. Data unassociated with its position may make no sense. Equipping all the sensor nodes with a global positioning system (GPS) is a direct but not available way since its considerable costs. So the position of each sensor node must be estimated first. Moreover, some communication protocols in WSNs are location based. Therefore localization of sensor nodes in WSNs is one of the most critical and fundamental tasks.

Localization algorithms in WSNs can be classified as centralized algorithms and distributed algorithms. The centralized approaches, which use a powerful node (sink node or base station) to calculate the position of each node, are disadvantageous due to the computational limitation of sensor nodes. Multi-dimensional Scaling (MDS) is a typical centralized algorithm. However, the centralized algorithms may not be available in large scale WSNs owing to the heavy computation and communication cost. A MDS-based distributed approach is proposed in this paper which employs the concept of clustering to

---

[*] Corresponding author, email: qzchen@zjut.edu.cn

ISSN: 2005-4262 IJGDC

degrade the problem scale. Our contributions are: (1) we devised a competitive mechanism to carry out the cluster heads in which the nodes' capability of recruiting members is used as the main competitive factor. So that very small clusters with only very few nodes can be avoided (2). In our approach, cluster numbers and overlaps between clusters are well controlled and a tree structure is formed to help mapping to global coordinates from local coordinates in cluster.

The rest of the paper is organized as follows. Section 2 discusses the related works. Section 3 presents our motivation. Section 4 describes the proposed tree-based clustering MDS algorithm in detail. Section 5 provides the simulations and result analysis. Finally, Section 6 concludes the paper.

2. Related Works

Localization of sensor nodes in WSNs has been extensively studied. The current localization approaches can be categorized as range-based and range-free [2, 4, 7, 8]. Range estimation hardware is employed to estimate the distance between sensor nodes and key techniques such as received signal strength indicator (RSSI), angle of arrival (AOA), time of arrival (TOA) and time difference of arrival (TDOA) are used in range-based schemes. Range-free algorithms use the connectivity of sensor nodes to estimate their positions. Distance vector (DV) hop, centroid system, and approximate point in triangulation (APIT) are typical range-free algorithms. Y. Liu, *et al.*, have made great effort to the localization and localizability problem in WSN and have proposed many effective localization methods, such as confidence based iterative localization [9], mobile beacon based non-iterative localization[10], component based localization in sparse wireless networks[11], *etc*.

MDS, which has its origins in psychometrics and psychophysics [4] and views similarities between data as distances, is now employed in localization approaches in WSNs. The underlying principle of MDS is to convert distances into an inner product matrix, whose eigenvectors are the unknown coordinates. MDS-MAP [12] is the first MDS-based localization method proposed in 2003 by Shang Yi, *et al.*, The connectivity information is used to compute the shortest paths between all pairs of nodes in order to approximate the missing entries of the distance matrix in this method. Then if a sufficient number of anchor nodes are known, the absolute coordinates of all nodes can be determined. Since the shortest path distance is not the true Euclidean distance, MDS-MAP does not work well in case of irregularly-shaped WSNs. Shang Yi, *et al.*, developed another method called MDS-MAP(P) in 2004 [13], which can be executed in a distributed fashion. MDS-MAP (P) build a local map for each node and its neighbors, then merge these maps together to get a global map. However the precision of MDS-MAP (P) in range-free mode is still to be improved.

Inspired from these classical algorithms, many other derived algorithms based on MDS are proposed. R. Iyengar, *et al.*, proposed a scalable and distributed localization scheme for WSNs [14]. The idea of clustering is adopted in this scheme for the coordinates formation wherein a small subset of the nodes, then the local relative coordinates are used to establish the coordinate system for the whole network. The cluster head is randomly chosen in this scheme. The heavy communication caused by excessive clusters and repetitive computation cause by excessive overlaps between clusters may lead to overhead costs for sensor nodes. HMDS [15] is an MDS based algorithm for hierarchical WSNs which can operate in non-convex network environments. In this algorithm, the hierarchical structure is constructed by clustering and the clustering method in [14] is employed. So it may have similar problems with [14]. G. Latsoudas, *et al.*, proposed a two-stage algorithm combined algebraic initialization and gradient descent with classical MDS [16]. Fast map algorithm borrowed from the database literature is their approach. A range-based weighted MDS algorithm is proposed in reference [17], which can be operated in partially connected WSNs. CMDS [18] is also proposed to improve the accuracy and computational complexity of classical MDS method by using k-hop clustering. The value of k is not clearly introduced. However, it may affect the performance of this algorithm. IMDS [19] improves MDS by distance matrix refinement in order to decrease distance matrix error in classical MDS. Though the accuracy is improved, the computational complexity is still remained.

3. Problem and Motivation

The previous researches showed us the availability of applying MDS into localization in WSN. However, there are still some problems unsolved. The computational complexity is fairly high in traditional MDS algorithm, since it contains many matrix based computation. The distance between any pair of nodes should be estimated for further processing by matrix based computation. And the scale of matrix grows with the amount of sensor nodes in the network. In many applications, there may be hundreds or thousands of sensor nodes which can lead to the high complexity of computation. So degrading the scale of matrix is an intuitive idea to reduce the computational complexity of traditional MDS algorithm. Clustering is a commonly used method to degrade the problem scale. Therefore, we adopted clustering to divide the whole wireless sensor network into several parts with some overlapped nodes between adjacent clusters. Then the in-cluster localization can be done by MDS algorithm. But another problem comes up that how to assure a proper overlap between clusters. If the amount of overlapped nodes is too small, it is impossible to convert local coordinates into global coordinates. Conversely, if the amount of overlapped nodes is too large, it may leads to too much repetitive computation which will cost computational resources and energy. So the cluster head should be carefully chosen and the overlapped area should be well controlled in the clustering algorithm. Sensor nodes compete for the chance to be a cluster head according to their capability of recruiting members. We also consider a threshold of overlapped nodes to ensure there are enough overlapped nodes for the later coordinate transformation. In this way, the computational complexity can be reduced by degrading the problem scale. Cluster heads are dynamically elected to avoid running out of energy.

4. The TBC-MDS Approach

The principle idea of this method is the divide and rule strategy by which a large scales WSN is divided into several small clusters. Then the MDS based localization algorithm is applied in each cluster to calculate the local coordinates which will be mapped into global coordinates. The whole localization procedure can be illustrated by three phases: clustering phase, in-cluster localization phase and global coordinates mapping phase.

4.1. Tree-based Clustering and Inner-cluster Localization

Here we define some states for the sensor nodes during the clustering phase.

CH (Cluster Head): Obviously, it means a sensor node in this state is a cluster header. Only the root node is initiated to this state.

NM (Non-Member): It means a sensor node in this state hasn’t join in any cluster ever. All nodes except the root node are initiated to this state.

CM (Cluster Member): It means a sensor node in this state is a member of a specific cluster.

BN (Border Node): It means a sensor node in this state has joint in more than two different clusters.

CHC (Cluster Head Candidate): It means a sensor node in this state is qualified for competing to be a cluster head. In another word, it is a candidate for cluster head.

END: Simply it means the ending.

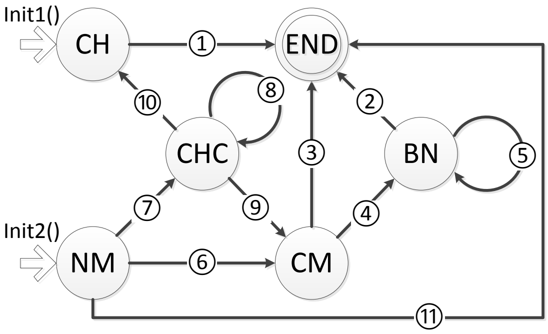

Figure 1. State Transition Diagram During the Clustering Phase

These states can be transited under certain condition. The procedures of state transition are shown in Figure 1. Each number in Figure 1 represents a transition from one state to another. Init1() and Init 2() are two initiation procedures. All the procedures will be described in pseudo code and some symbols are explained in Table 1.

Table 1. Identifiers in the Pseudo Codes

Symbols	*Meaning*
State	A variable to save the current state for the node, its value can be CH, NM, CM, BN, CHC, FAIL.
Hop	A variable to save the number of hops from the current node to the root.
Parent	A variable to save the parent node in the tree structure.
CH[]	A variable to save the cluster heads for the current node.
NB[]	A variable to save the neighbors for the current node.
P_CH[]	A variable to save the candidate for parent node.
b	A parameter for head competing, implicate the number of members that the current node can recruit if it is a cluster head.
B[]	A variable to save the neighbors' b parameters.
Timer1	The primary timer which is started when initiating the algorithm and the algorithm will be force to quit when time is up.
Timer2	The timer for receiving messages which is started when a message is received.
Recruit	It is a message name with the structure of (*Recruit, u, Hop, Mu[]*), where *Recruit* is a tag, *u* is the ID of sender, *Hop* is the value of the variable Hop and *Mu[]* is the set of neighbor nodes. This message is always sent by a cluster head to notify its neighbors to join in the cluster.
Relay	It is a message name with the structure of (*Relay, u, Hop, Mu[]*), where *Reply* is a tag, other parameters is a direct copy from the *Recruit* message it received. This message is always sent by a node who received a *Recruit* message. If a non-member node receives a *Replay* message, value of P_CH[] should be updated and decide if to compete for cluster head.
UpdateB	It is a message name with the structure of (*UpdateB, u, bu*), where *UpdateB* is a tag, *u* is the ID of sender and *bu* is the value of the variable b. A node broadcast this message to its neighbors to notify its value of the variable b.

Init1() is an initialization for a cluster head and Init2() is an initialization for a non-member node.

- Init1

Pseudo code of Init1()

```
State=CH;      //Set the state of the current node as CH
Hop=0;
Parent=&this;   //Set the current node itself as its parent node
```

- Init2

Pseudo code of Init2()

```
State=NM;      // Set the state of the current node as NM
Hop=-1;        //The hops to the root is not available
Parent=0;      //The parent node is not available
CH[]=Ø;
NB[]=Ø;
P_CH[]=Ø;
b=0;
B[]=Ø;
Timer1=T1;
Timer2=-1;
```

The following transition procedures are presented in the order of first executed in the clustering algorithm.

- Transition ①

Pseudo code of Transition ①

```
IF(State==CH)
  IF(NB[]==Ø)
      //Neighbor discovering
      NB[]=Find_My_Neighbor();
  END IF
  Mu[]=NB[];
  Send_Msg=(Recruit,u,Hop,Mu[]);//Build a Recruit Message
  Send_Msg.send();  //Send the Recruit Message
  EXIT();  //Go to the state of END
END IF
```

- Transition ⑥ or ⑨

Pseudo code of Transition ⑥ or ⑨

```
IF(Received_Msg==(Recruit,u,Hop,Mu[]) && (State==NM||CHC))
  CH[]←u;  //Put u into the set CH[]
  State=CM;
  Send_Msg=(Relay,u,Hop,Mu[]);
  Send_Msg.send();
END IF
```

- Transition ⑦ is a stringent conditional procedure, which means the state of current node won't change if not all these conditions are satisfied.

Pseudo code of Transition ⑦

```
IF(Received_Msg==(Relay,u,Hop,Mu[]) && (State==NM))
  IF(NB[]==Ø)
      NB[]=Find_My_Neighbor(); //Neighbor discovering
  END IF
    IF(NB[]∩Mu[]>=3)
      P_CH[]←[1];
      State=CHC;
    END IF
    IF(P_CH[]!=Ø)
      Parent=the node in P_CH[] with the minimum Hop value;
      Hop=P_CH[Parent]+1;
        b=(NB[]-(NB[]∩Mu[])).size; //b is the number of nodes in NB[] but not in Mu[]
```

```
13      Send_Msg=(UpdateB,v,b);  //v is the ID of current node
14      Send_Msg.send();
15      END IF
16    END IF
```

- Transition ⑧ is a reflexive procedure which will not change the state of the current node.

Pseudo code of Transition ⑧

```
IF(Received_Msg==(Relay,u,Hop,Mu[]) && (State==CHC))
  execute Line 2 to Line 14 in Transition ⑦;
ELSE IF(Received_Msg==(UpdateB,u,b) && (State==CHC||NM))
  B[]←{u,b};
END IF
```

- Transition ⑩

Pseudo code of Transition ⑩

```
IF(State==CHC && Timer2 time's up && b>max(B[].b))
  State=CH;
END IF
```

- Transition ④ or ⑤

Pseudo code of Transition ④ or ⑤

```
IF(Received_Msg==(Recruit,u,Hop,Mu[])&&(State==CM||BN))
  CH[]←u;
  State=BN;
  Send_Msg=(Relay,u,Hop,Mu[]);
  Send_Msg.send();
END IF
```

- Transition ② or ③

Pseudo code of Transition ② or ③

```
IF(Timer2 is up && (State==CM||BN))
  EXIT();
END IF
```

- Transition ⑪

Pseudo code of Transition ⑪

```
IF(Timer1 is up)
  IF(State==NM)
    State=FAIL;  //Set the state of current node as FAIL
  END IF
  EXIT();
END IF
```

The whole clustering algorithm can be explained as follows:

(1) A sensor node in the center of the WSN, which may have many neighbors, should be manually chosen as the root node.

(2) Once the root node is identified, it is initiated as a cluster head. Find_My_Neighber() and Recruit() functions are executed sequentially by the root to discover its neighbors and to recruit cluster members.

(3) A node joins in the cluster once received a Recruit message and sends a Relay message to its neighbors. If a node is already a member of a cluster when it receives a new Recruit message, it is then a border node.

(4) Find_My_Neighber() function is executed when a node receives a Relay message to discover its neighbor nodes. The intersection of its neighbors and the neighbors of the cluster head indicated in the Relay messages is carried out. If the elements in the intersection are more than three, the head is recognized as a parent candidate and the current node itself turns into a head candidate.

(5) The head candidate choose the node with the minimum hops to the root from P_CH[] as its parent and calculate its potential cluster member count which will be delivered to other head candidates in an UpdateB message.

(6) One node with the maximum value of b will make itself as a cluster head when Timer2 is up. And then send a Recruit message to recruit members.

(7) All the above operations will be repeated until Timer1 is up.

After the clustering phase is accomplished, the classical MDS method [8] is applied to in each cluster to carry out the local coordinates.

4.2 Clusters merging phase

The tree structure, which is a result of the clustering phase, is the basis of this phase. The local coordinates of each sensor node in cluster are mapped into global coordinates by traversing the tree in the depth-first order. Border nodes, which have more than one local coordinates, play as the intermediary in the mapping process. The affine transformation details are described as follows.

Suppose there are m nodes with their local coordinates $(x_1, y_1), (x_2, y_2) \cdots (x_m, y_m)$, and their corresponding global coordinates are $(x_1^{'}, y_1^{'}), (x_2^{'}, y_2^{'}) \cdots (x_m^{'}, y_m^{'})$. The global coordinates $(x_m^{'}, y_m^{'})$ can be carried out by Formula (1).

$$\begin{bmatrix} x_m^{'} \\ y_m^{'} \\ 1 \end{bmatrix} = \begin{bmatrix} \lambda\cos\theta & \pm\lambda\sin\theta & b_x \\ \lambda\sin\theta & \mp\lambda\cos\theta & b_y \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x_m \\ y_m \\ 1 \end{bmatrix} \tag{1}$$

In order to get the parameters λ, θ, b_x and b_y, the coordinates of the m nodes should be substituted into Formula (1). Then we get the equation set (2) and (3).

$$\begin{cases} x_1^{'} = \lambda\cos\theta\, x_1 \pm \lambda\sin\theta\, x_1 + b_x \\ x_2^{'} = \lambda\cos\theta\, x_2 \pm \lambda\sin\theta\, x_2 + b_x \\ \qquad\vdots \\ x_m^{'} = \lambda\cos\theta\, x_m \pm \lambda\sin\theta\, x_m + b_x \end{cases} \tag{2}$$

$$\begin{cases} y_1^{'} = \lambda\cos\theta\, y_1 \mp \lambda\sin\theta\, y_1 + b_y \\ y_2^{'} = \lambda\cos\theta\, y_2 \mp \lambda\sin\theta\, y_2 + b_y \\ \qquad\vdots \\ y_m^{'} = \lambda\cos\theta\, y_m \mp \lambda\sin\theta\, y_m + b_y \end{cases} \tag{3}$$

MLE (Maximum Likelihood Estimation) is applied to get the value of these parameters in the sense of MMSE (Minimum Mean Square Error) and finally we get the following linear equations in Formula (4).

$$\begin{cases} x^{'} = \lambda\cos\theta\, x \pm \lambda\sin\theta\, y + b_x \\ y^{'} = \lambda\sin\theta\, x \mp \lambda\cos\theta\, y + b_y \end{cases} \tag{4}$$

5. Simulation and Discussion

5.1. Simulation Setting

Computer simulations on Matlab are conducted to evaluate the performance of the proposed approach. We consider that there are 200 sensor nodes randomly deployed in a square of 1000m×1000m (Figure 2). The communication range of each sensor node is variable, and different degrees of connectivity are simulated by setting different communication ranges of sensor nodes.

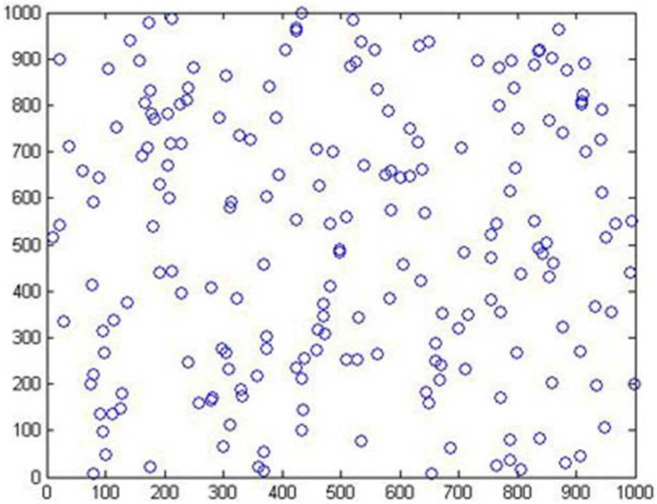

Figure 2. Randomly Deployment of Sensor Nodes

5.2. Evaluation Criteria

The average positioning error ratio (APER) and positioning coverage ratio (PCR) are mainly considered to evaluate the performance of our approach.

APER is defined in Formula (5), where n is the total number of sensor nodes, r is the communication range of sensor nodes, (x_i', y_i') and (x_i, y_i) are the localization result and true coordinates respectively.

$$APER = \frac{\sum_{i=1}^{n} \sqrt{\left(x_i' - x_i\right)^2 + \left(y_i' - y_i\right)^2}}{nr} \times 100\% \tag{5}$$

PCR is defined in Formula (6), where N_P is the number of positioned nodes and n is the total number of sensor nodes. Higher PRC means strong capability of localization.

$$PCR = \frac{N_P}{n} \times 100\% \tag{6}$$

5.3. Results and Discussion

Our proposed TBC-MDS algorithm is compared with SDGPSN [11] algorithm in PCR with different average connectivity degree of nodes (Figure 3). The result shows that PCR increases as the connectivity degree increases. The PCR of TBC-MDS is more than 90% in case of the average connectivity degree reached 12, while the PCR of SDGPSN is only 60%. When the average connectivity degree reached 14, only one node is not positioned with the TBC-MDS algorithm, the PCR of which is much higher than SDGPSN algorithm. But when the average connectivity degree is more than 18, the PCR of the two algorithms tend to be closer.

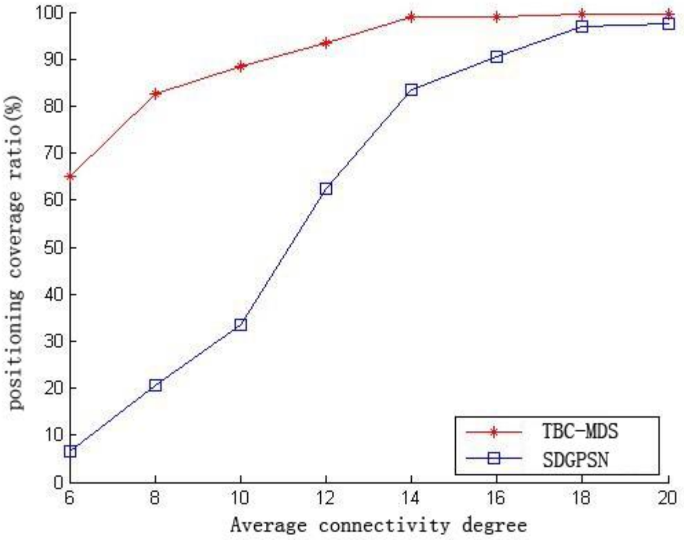

Figure 3. PCR in Different Connectivity Degrees

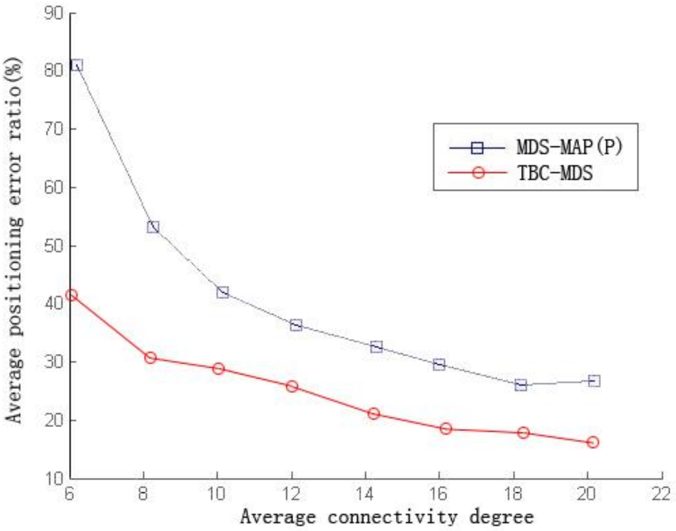

Figure 4. APER Comparison of Different Algorithms

By analyzing the clustering procedure of the two algorithms, it is known that clustering mechanism in SDGPSN is randomly head election which may lead to uneven distribution of clusters and the number of border nodes is uncontrollable. Even isolated clusters or isolated nodes may exist. That will further result in unavailability of mapping local coordinates into global. Therefore some nodes cannot be positioned with SDGPSN. Conversely, the clustering procedure is much more controllable in out scheme. It is ensured that there are enough border nodes unless there are isolated nodes which are not connectable to the root node because of the constraints of communication range. That is why the PCR of our approach is higher than SDGPSN.

The APER accuracy comparison of the well-known MDS-MAP(P) and the proposed TBC-MDS is shown in Figure 4. Obviously, it can be seen that both of the two algorithms have lower average positioning error ratio when the connectivity degree increases. But on the whole, our algorithm has a higher accuracy than MDS-MAP(P), especially in low connectivity degree cases.

6. Conclusion and Future Work

A tree based clustering multidimensional scaling algorithm for node localization in WSN is devised in this paper. The in-cluster local coordinates are carried out by applying traditional MDS method after clusters are formed and then mapped into global coordinates. The simulation results show that TBC-MDS has a high positioning coverage ratio and low average positioning error ratio. It works better than classical MDS methods especially in low connectivity scenarios or sparse networks. As it is a distributed algorithm with high coverage and accuracy, it can be applied in large scale WSNs.

As the sensor nodes are battery-operated, the energy is highly limited. The energy consumption of localization algorithms is crucial though it is not addressed in this paper. As a future work, we will do some further research on energy consumption of node localization.

Acknowledgements

This work is supported by the National Natural Science Foundation of China (Grant No. 61379023) and Provincial Natural Science Foundation of Zhejiang (Grant No. LY12F02036).

References

[1] F. Viani, P. Rocca, G. Oliveri, D. Trinchero and A. Massa, "Localization, tracking, and imaging of targets in wireless sensor networks: An invited review", Radio Sci, vol. 46, **(2011)**.

[2] L. Cheng, C. D. Wu, Y. Z. Zhang, H. Wu, M. X. Li and C. Maple, "A Survey of Localization in Wireless Sensor Network", Int J Distrib Sens N, **(2012)**.

[3] C. Buratti, A. Conti, D. Dardari and R. Verdone, "An Overview on Wireless Sensor Networks Technology and Evolution", Sensors-Basel, vol. 9, **(2009)**, pp. 6869-6896.

[4] G. Q. Mao, B. Fidan and B. D. O. Anderson, "Wireless sensor network localization techniques", Comput Netw, vol. 51, **(2007)**, pp. 2529-2553.

[5] J. Yick, B. Mukherjee and D. Ghosal, "Wireless sensor network survey", Comput Netw, vol. 52, **(2008)**, pp. 2292-2330.

[6] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam and E. Cayirci, "Wireless sensor networks: a survey", Comput Netw, vol. 38, **(2002)**, pp. 393-422.

[7] H. Tian, Y. Ding and S. Yang, "A Survey on MDS-Based Localization for Wireless Sensor Network", Adv Intel Soft Compu, vol. 159, **(2012)**, pp. 399-403.

[8] A. H. Sayed, A. Tarighat and N. Khajehnouri, "Network-based wireless location", Ieee Signal Proc Mag, 22, **(2005)**, pp. 24-40.

[9] Z. Yang and Y. Liu, "Quality of trilateration: Confidence-based iterative localization", Parallel and Distributed Systems, IEEE Transactions on, vol. 21, **(2010)**, pp. 631-640.

[10] Y. He, Y. Liu, X. Shen, L. Mo and G. Dai, "Noninteractive localization of wireless camera sensors with mobile beacon", Mobile Computing, IEEE Transactions on, vol. 12, **(2013)**, pp. 333-345.

[11] X. Wang, J. Luo, Y. Liu, S. Li and D. Dong, "Component-based localization in sparse wireless networks", IEEE/ACM Transactions on Networking (ToN), vol. 19, **(2011)**, pp. 540-548.

[12] Y. Shang, W. Ruml, Y. Zhang and M. Fromherz, "Localization from Mere Connectivity", Proceeding of the 4th ACM international symposium on Mobile ad hoc networking & computing, ACM New York, NY, USA, **(2003)**, pp. 201-212.

[13] Y. Shang, W. Ruml, Y. Zhang and M. Fromherz, "Localization from connectivity in sensor networks", Ieee T Parall Distr, vol. 15, **(2004)**, pp. 961-974.

[14] R. S. Iyengar and B., "Scalable and distributed GPS free positioning for sensor networks", Proceeding of IEEE Int'l Conf. on Communications 2003, IEEE Communications Society, Anchorage, **(2003)**, pp. 338-342.
[15] G. J. Yu and S. C. Wang, "A hierarchical MDS-based localization algorithm for wireless sensor networks", Proceedings of the 16th 1st Mobile and Wireless Communications, vol. 1-3, **(2007)**, pp. 255-259.
[16] G. Latsoudas and N. D. Sidiropoulos, "A fast and effective multidimensional scaling approach for node localization in wireless sensor networks", IEEE Trans Signal Proces, vol. 55, **(2007)**, pp. 5121-5127.
[17] F. K. W. Chan and H. C. So, "Efficient Weighted Multidimensional Scaling for Wireless Sensor Network Localization", IEEE Trans Signal Proces, vol. 57, **(2009)**, pp. 4548-4553.
[18] M. Shon, M. Jo and H. Choo, "An interactive cluster-based MDS localization scheme for multimedia information in wireless sensor networks", Computer Communications, vol. 35, **(2012)**, pp. 1921-1929.
[19] B. R. K. Stojkoska and V., "Improved MDS-based algorithm for nodes localization in wireless sensor networks", Proceeding of EUROCON, 2013 IEEE Zagreb, **(2013)**, pp. 608 - 613.

Authors

Guoyong Dai, received his Master in Computer Science from Zhejiang University of Technology, China, in 2009. He is currently pursuing his PhD in Control Science and Engineering at the same university. He is currently a Lecturer at Zhejiang Shuren University, China. His current research interests include wireless sensor network and information security.

Chunyu Miao, received his Master in Computer Science from East China Normal University, China, in 2006. He is currently pursuing his PhD in Control Science and Engineering at Zhejiang University of Technology, China. He is also an Associate Professor in Zhejiang Normal University, China. His current research interests include wireless sensor network and trusted computing.

Yidong Li, is currently pursuing his Master in Computer Science in Zhejiang University of Technology, China. His research interests include wireless sensor network and distributed systems.

Keji Mao, received his PhD in Control Theory and Engineering from Zhejiang university of Technology, China, in 2014. He is now a Lecturer at College of Computer Science and Technology, Zhejiang university of Technology. His current research interests include wireless sensor network and data mining.

Qingzhang Chen, received his PhD in Computer Science from Hefei University of Technology, China, in 2007. He is currently a Full Professor in Zhejiang University of Technology, China. His current research interests include wireless sensor network, computer supported collaborative work, data fusion and information security.