

Research Article

Improved DV-Hop Node Localization Algorithm in Wireless Sensor Networks

Xiao Chen and Benliang Zhang

School of Electronic and Information Engineering, Nanjing University of Information Science and Technology, Nanjing 210044, China

Correspondence should be addressed to Xiao Chen, chenxiao@nuist.edu.cn

Received 3 May 2012; Revised 26 June 2012; Accepted 19 July 2012

Academic Editor: Shashi Phoha

Copyright © 2012 X. Chen and B. Zhang. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

An improved algorithm is put forward to improve the poor locating performance of the DV-Hop algorithm, which is one of the range-free algorithms in wireless sensor networks. Firstly, we set some anchor nodes at the border land of monitoring regions. Secondly, the average one-hop distance between anchor nodes is modified, and the average one-hop distance used by each unknown node for estimating its location is modified through weighting the received average one-hop distances from anchor nodes. Finally, we use the particle swarm optimization to correct the position estimated by the 2D hyperbolic localization algorithm, which makes the result closer to the actual position. The simulation results show that the proposed algorithm has better localization performance in the localization precision and stability than the basic DV-Hop algorithm and some existing improved algorithms.

1. Introduction

Wireless sensor networks (WSNs) are composed of plenty of sensor nodes that are deployed in the monitoring field, and they form a multihop self-configured network by means of wireless communication. The sensor nodes have perception, processing, and communication ability [1].

In many applications of WSNs, the monitoring data is of no significance if the locations of sensors are unknown because the position is needed to locate events that occurred in WSNs. Positioning information can be used not only to report events place, but also for the target tracking, the target track prediction, the assist routing and the topology of the network management, and so forth.

On the basis of whether it is required to measure the actual distances between nodes, the nodes localization algorithms can be classified into two categories, that is, range-based and range-free [2]. The former exploits the distance or angle information between neighbor nodes and then uses the trilateral or multilateral localization methods to locate the unknown nodes, such as RSSI [3], TDOA [4], TOA [5], and AOA [6]. The latter uses the estimated distance instead of the metrical distance to locate the unknown nodes,

such as the centroid algorithm [7], the DV-Hop algorithm and the convex programming algorithm [8]. Due to the high cost of hardware facilities and energy consumption required by range-based approaches, the range-free algorithms attract more and more attention.

In this paper, we present an improved DV-Hop algorithm, which can reduce the localization error. At first few anchor nodes are properly distributed. The average hop distance is calculated in a different way, in which not only the global property of average hop size and anchor nodes are considered. Finally, the particle swarm optimization algorithm is used to correct the position estimated by the 2D hyperbolic location algorithm. The rest of the paper is organized as follows. Section 2 describes the basic DV-Hop algorithm and some existing improved DV-Hop algorithms. In Section 3, we propose the modified DV-Hop algorithm. In Section 4, simulation results are presented and analyzed. Section 5 analyses communication cost and computational efficiency. Finally, we conclude the paper in Section 6.

2. Related Works

2.1. DV-Hop Localization Algorithm. DV-Hop (distance vector-hop) localization algorithm [9], which was proposed

by Dragos Niculescu, is one of the typical representatives of range-free localization algorithm. Its basic idea is that the distance between the unknown nodes and the reference nodes is expressed by the product of average hop distance and the hop count.

The algorithm implementation is comprised of three steps. In the first step, each anchor node broadcasts an anchor to be flooded throughout the network containing the anchors location with a hop count value initialized to one. Each receiving node maintains the minimum hop count value. Anchors with higher hop count values to a particular anchor are defined as the useless information and ignored. Then, rest of the anchors are flooded outward with hop count values incremented at every intermediate hop. Through this process, all nodes in the network get the minimal hop count to every anchor node.

In the second step, once an anchor gets hop count value to other anchors, it estimates an average size of one hop, which is then flooded to the entire network. After receiving the hop size, the unknown nodes multiply the hop size by the hop count value to derive the estimated physical distance from the anchor. The average hop size is estimated by the anchor node i using the following formula:

$$\text{Hop Size}_i = \frac{\sum_{j \neq i} \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}}{\sum_{j \neq i} \text{hop}_{ij}}, \quad (1)$$

where (x_i, y_i) and (x_j, y_j) are coordinates of anchor nodes i and j , respectively, and hop_{ij} is the hop count between the anchor nodes i and j .

Each anchor node broadcasts its hop size to the network using controlled flooding. Unknown nodes receive the hop size information and save the first one. Meanwhile, they transmit the hop size to their neighbor nodes. This scheme ensures that most nodes receive the hop size from the anchor node that has the least hops between them.

In the last step, when unknown nodes get three or more distance information from anchor nodes, the trilateral measuring method or the maximum likelihood estimation method is used to calculate their locations.

It is assumed that (x_u, y_u) are the location of the unknown node u and (x_i, y_i) are the known location of the i th anchor node. d_{ui} is the distance between them. Then, we have

$$\begin{aligned} (x_u - x_1)^2 + (y_u - y_1)^2 &= d_{u1}^2 \\ (x_u - x_2)^2 + (y_u - y_2)^2 &= d_{u2}^2 \\ &\vdots \\ (x_u - x_n)^2 + (y_u - y_n)^2 &= d_{un}^2. \end{aligned} \quad (2)$$

The coordinates of u are computed by the following formula:

$$A = -2 \times \begin{bmatrix} x_1 - x_n & y_1 - y_n \\ x_2 - x_n & y_2 - y_n \\ \vdots & \vdots \\ x_{n-1} - x_n & y_{n-1} - y_n \end{bmatrix}, \quad (3)$$

$$B = \begin{bmatrix} d_{u1}^2 - d_{un}^2 - x_1^2 + x_n^2 - y_1^2 + y_n^2 \\ d_{u2}^2 - d_{un}^2 - x_2^2 + x_n^2 - y_2^2 + y_n^2 \\ \vdots \\ d_{u(n-1)}^2 - d_{un}^2 - x_{n-1}^2 + x_n^2 - y_{n-1}^2 + y_n^2 \end{bmatrix}, \quad (4)$$

$$P = \begin{bmatrix} x_u \\ y_u \end{bmatrix}, \quad (5)$$

and $P = (A^T A)^{-1} A^T B$.

2.2. Existing Improved DV-Hop Algorithms. When the nodes are uniform and isotropic in WSNs, it is reasonable to get every average hop distance so that the positioning accuracy is proper. But when the network is sparse or nodes are distributed unevenly, nodes often receive inaccurate hop information and calculate the every average hop distance with bigger errors, which affects the network node location accuracy.

To solve this, many scholars improved the DV-Hop algorithm from all aspects. Zhang fixed the anchor nodes at the border land of the network, which saved the number of anchor nodes [10]. Shi used the cluster strategy to reduce the communication cost and group the conflict probability in the first stage and used the quasi-Newton optimization algorithm to estimate the position of nodes, which improved the localization accuracy without increasing communications traffic [11]. Gao et al. formulated the hop-distance relationship for the quasi-UDG model in WSNs [12]. They derived an approximated recursive expression for the probability of the hop count with a given geographic distance. The border effect and dependence problem are also taken into consideration. Furthermore, they gave expressions describing the distribution of distance with known hop counts for inner nodes and those suffering from the border effect. Lee proposed a novel localization scheme based on the DV-Hop algorithm to improve the positioning accuracy with the least error hop sizes of anchors and average hop sizes of unknowns, each unknown node calculated its average hop size with hop sizes from anchors [13]. Lin put forward three approaches to improve the poor locating performance [14]. Firstly, the average one-hop distance between anchor nodes was refined by means of the least square method. Secondly, the average one-hop distance was modified through weighting the received average one-hop distances from anchor nodes. Finally, the iterative numerical method with the initial values of the estimated node location was presented by setting proper threshold. Shi corrected the average hop distance of anchor nodes by the total error between the calculated distances and the estimated distances and used 2D hyperbolic location algorithm, which improved the localization accuracy effectively [15]. Sezaki used a differential error correction scheme designed to reduce the location error accumulated over multiple hops [16]. Gao put forward a WSN localization algorithm based on the adaptive particle swarm optimization [17].

3. Improved DV-Hop Algorithm

In this paper, we propose an improved DV-Hop algorithm, which is divided into the following four steps. (1) Some anchor nodes are fixed at the border land of the monitoring regions. (2) The average one-hop distance among anchor nodes is modified, and the average one-hop distance used by each unknown node for estimating its location is modified. (3) The unknown nodes' positions are obtained by the 2D hyperbolic location algorithm. (4) We use the particle swarm optimization algorithm to correct the estimated position.

3.1. Deploy Anchor Nodes. Because anchor nodes are randomly distributed in DV-Hop algorithm, some anchor nodes may concentrate in an area. If we use those anchor nodes to estimate unknown nodes positions, it will produce larger error and bad stability unless the network has high similarity. Therefore, we fix some anchor nodes at the borderland of the monitoring regions firstly, and the rest of the anchor nodes are distributed randomly [10].

3.2. Calculation of the Average Hop Distance of Unknown Nodes. We modify the network average hop distance to improve the localization accuracy. The average hop distance of every anchor node Hop Size_i is revised based on the least square error criterion mentioned previously [14]:

$$\text{Hop Size}_i = \frac{\sum_{j \neq i} (\text{hop}_{ij} d_{ij})}{\sum_{j \neq i} \text{hop}_{ij}^2}, \quad i \neq j, \quad (6)$$

where d_{ij} is the straight-line distance between the anchor nodes i and j .

Then, we modify the average one-hop distance of unknown nodes Hop Size_u through weighting the N received average one-hop distances from the anchor nodes by the following formula:

$$\text{Hop Size}_u = \sum_{i=1}^n \lambda_i \text{Hop Size}_i, \quad (7)$$

$$\lambda_i = \frac{\text{hop}_i}{\sum_{k=1}^n \text{hop}_k},$$

where hop_i is the hop segment number between the unknown node u and the anchor node i in its routing table and n is the number of all anchor nodes.

The total of weighting value of all anchor nodes in the routing table of unknown node is 1. By the weighted average, the average hop distance of unknown nodes is reflected from the one-hop distance of all anchor nodes.

3.3. Position Determination of Unknown Nodes. We use the 2-D Hyperbolic location algorithm instead of the traditional trilateral or the multilateral position method to estimate the position of unknown nodes, which improves the estimated location precision.

It is assumed that (x_i, y_i) are the coordinates of anchor node i and (x_u, y_u) are the coordinates of unknown node u . The estimated distance d_{iu} between them is

$$d_{iu}^2 = (x_i - x_u)^2 + (y_i - y_u)^2. \quad (8)$$

If $A_i = x_i^2 + y_i^2$ and $B_u = x_u^2 + y_u^2$, (4) takes the form

$$d_{iu}^2 - A_i = -2x_i x_u - 2y_i y_u + B_u. \quad (9)$$

Equation (9) in the matrix form is

$$GZ = h, \quad (10)$$

where

$$Z = [x_u, y_u, B_u]^T, \quad G = \begin{bmatrix} -2x_1 & -2y_1 & 1 \\ -2x_2 & -2y_2 & 1 \\ \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots \\ -2x_n & -2y_n & 1 \end{bmatrix}, \quad (11)$$

$$h = \begin{bmatrix} d_{1u}^2 - A_1 \\ d_{2u}^2 - A_2 \\ \vdots \\ \vdots \\ d_{nu}^2 - A_n \end{bmatrix}.$$

According to (10), Z can be obtained by using the least mean square estimation method:

$$Z = (G^T G)^{-1} G^T h. \quad (12)$$

Therefore, the coordinates of unknown node u are

$$x_u = Z(1), \quad (13)$$

$$y_u = Z(2).$$

3.4. Particle Swarm Optimization. The particle swarm optimization (PSO) [18], one branch of heuristic methods, is especially useful in dynamically changing domains. PSO algorithm initializes a set of random solutions for the objective function and takes each individual solution as one particle. Each particle has an objective function determined by the fitness value, and it also has a rate determined by the distance and direction of particle search. We can determine the pros and cons of the particle based on the fitness value. According the two extremes, the individual extreme (p best) is the optimal position of the current search of the particle and the global minimum (g best) is the optimal position of the current search of all group particles. Each particle can update its velocity and position by

$$v_u^{k+1} = \omega v_u^k + c_1 \text{rand}_1^k (p \text{ best}_u^k - X_u^k) + c_2 \text{rand}_2^k (g \text{ best}^k - X_u^k), \quad (14)$$

$$X_u^{k+1} = X_u^k + v_u^{k+1},$$

where v_u^k is the velocity vector, X_u^k is the current position vector of the particle and its initial value is obtained by using (13), $p_{best}_u^k$ is the individual extreme of the particle u by k -cycles, and $g_{best}_u^k$ is the global extreme of all particles by k -cycles. In the typical PSO implementation, c_1 and c_2 are the acceleration factors and $rand_1^k$ and $rand_2^k$ are both random of $[0, 1]$. k is the current number of iterations.

The inertia weight, ω , is used to maintain the global and local search for the balance. The value of ω needs to be set large to jump out of the local minimum on the contrary to be set smaller to the convergence algorithm. In consideration of the global search, it is better to set large value for ω in the beginning while a small value for ω at the last stage. We take ω as the function of iterations which can linearly decrease by iterations

$$\omega = \omega_{\max} - \frac{\omega_{\max} - \omega_{\min}}{\text{iter}_{\max}} \times k, \quad (15)$$

where ω_{\max} and ω_{\min} are the initial inertia weight and the termination inertia weight, respectively, iter_{\max} is the maximum number of iterations, and k is the current number of iterations.

The fitness in our paper is designed as

$$\text{fitness}(X_u) = \sum_{i=1}^n \alpha_i^2 (d_i - |X_u - A_i^T|)^2, \quad (16)$$

where A_i^T is the position of the anchor node and α_i is inversely proportional to the hop count between the unknown node u and the anchor node i .

4. Simulation Results and Discussion

To evaluate the performance of our proposed algorithm, we use Matlab2009a as the simulator to implement the network scenario and determine the localization results. The simulation experiment environment is set as follows. Firstly four anchor nodes are fixed at the vertex of the sensor field of $100 \times 100 \text{ m}^2$, and then unknown nodes and the rest of anchor nodes are distributed randomly.

The parameters in the PSO algorithm are $c_1 = c_2 = 2.05$, $\omega_{\max} = 0.9$, $\omega_{\min} = 0.4$, $\text{iter}_{\max} = 20$, the number of particles is 20, and $V_{\max} = 10 \text{ m}$, which have been tested and verified by researchers before [18]. The network environment parameters of every simulation, such as the network area, the number of anchor nodes, the number of unknown nodes, and the radio range of sensor nodes, are the same. The algorithm localization accuracy and stability are, respectively, evaluated by the localization error and localization error variance, which are shown by

$$\text{Average Error} = \frac{\sum_{k=1}^n |\hat{X}_u - X_u|}{n \times R},$$

Localization Error Variance

$$= \sqrt{\frac{\sum_{i=1}^n (|\hat{X}_u - X_u| - \text{Average Error} \times R)^2}{n \times R^2}}, \quad (17)$$

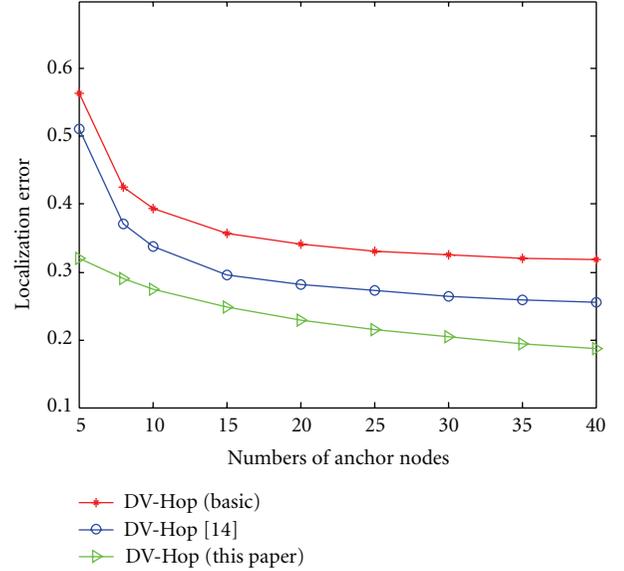


FIGURE 1: Localization error with different number of anchor nodes.

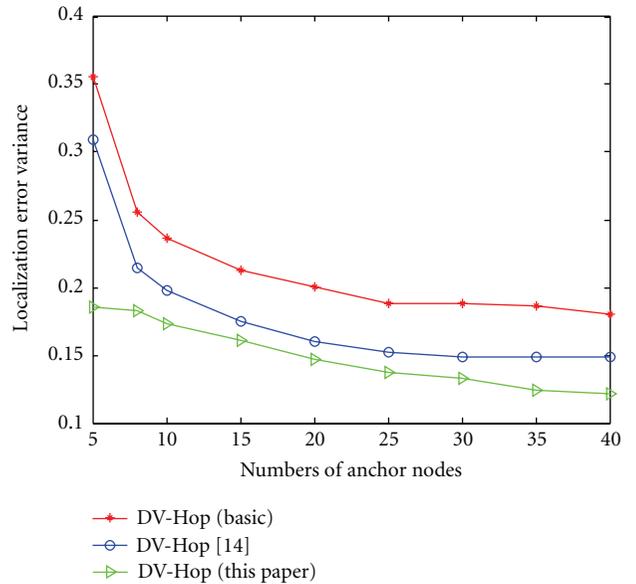


FIGURE 2: Localization error variance with different number of anchor nodes.

where n is the number of unknown nodes, X_u is the unknown node u evaluated position, \hat{X}_u is the unknown node u actual position, and R is the radio range of sensor nodes.

The result data is the average of 100 experiments.

The localization error and error variance with different numbers of anchor nodes are shown in Figures 1 and 2. It is clear that the proposed algorithm is obviously better than the basic DV-Hop algorithm and the improved algorithm in the [14]. The localization error in our algorithm is about 13% lower than that in the basic DV-Hop algorithm and about 7% lower than that of [14]. Localization error variance decreases by about 7% and 3% correspondingly. The use of the PSO

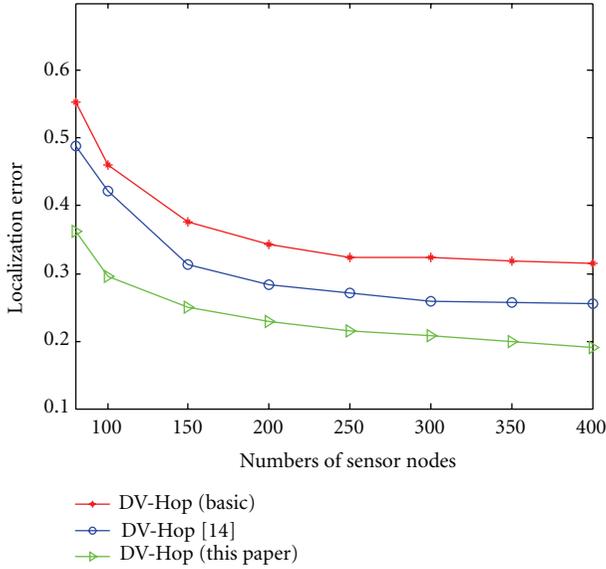


FIGURE 3: Localization error with different number of sensor nodes.

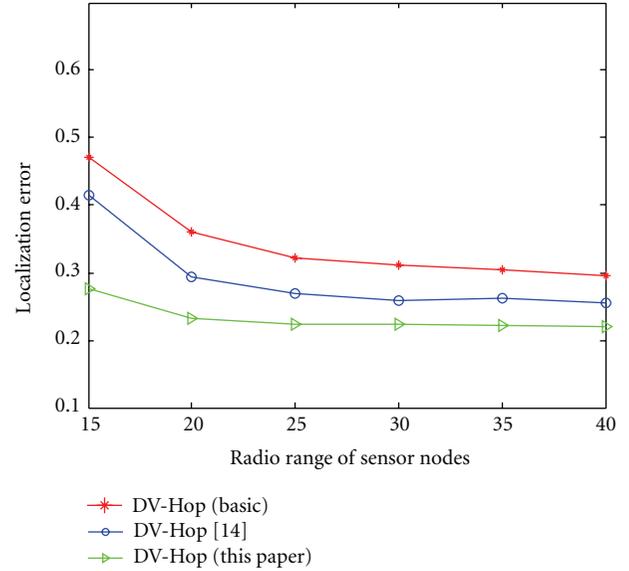


FIGURE 5: Localization error with different radio ranges of sensor nodes.

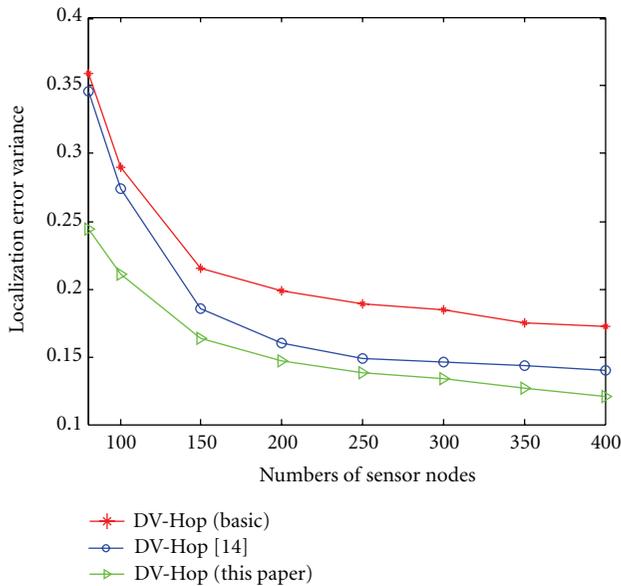


FIGURE 4: Localization error variance with different number of sensor nodes.

algorithm has fastened convergence speed apparently (the number of sensor nodes is set to 200 and R is set to 22 m).

The localization error and error variance with different numbers of sensor nodes are shown in Figures 3 and 4. It is shown that our proposed algorithm is obviously better than the basic DV-Hop algorithm and the improved algorithm in [14]. The localization error in our algorithm is about 14% lower than that in the basic DV-Hop algorithm and about 8% lower than that of [14]. Localization error variance decreases by about 6% and 3% correspondingly (the proportion of anchor nodes is set to 10% and R is set to 22 m).

The localization error and error variance with different radio ranges of sensor nodes are shown in Figures 5 and 6.

It is clear that our improved algorithm is obviously superior to the basic DV-Hop algorithm and the improved algorithm proposed in [14]. The localization error in our algorithm is about 11% lower than that in the basic DV-Hop algorithm and about 6% lower than that of [14]. Localization error variance decreases by about 5% and 2% correspondingly (the number of sensor nodes is set to 200 and the proportion of anchor nodes is set to 10%).

5. Communication Cost and Computational Efficiency

The algorithm communication cost is presented by the number of transmitting packets in process of localization. In this paper, we refine the average one-hop distance between anchor nodes and modify the average one-hop distance used by each unknown node for estimating its own location through weighting N received average one-hop distances from anchor nodes, and the data is saved by unknown nodes in the basic DV-Hop algorithm. So we do not need the extra communication cost.

The process used by the PSO algorithm for correcting the position estimates computed by the 2D hyperbolic location algorithm increases the computation time. Figure 7 shows the localization time comparison. The proposed algorithm is slower about 0.01 second than the original algorithm; meanwhile localization precision is improved. Therefore, we need to keep balance between the precision and the calculation in the practical application.

6. Conclusions

We have presented a modified DV-Hop localization algorithm to improve the poor locating performance of the DV-Hop algorithm, mainly focusing on the following steps.

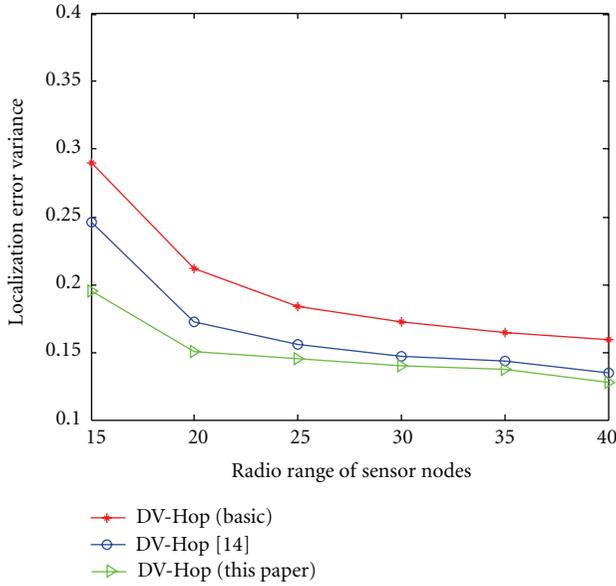


FIGURE 6: Localization error variance with different radio ranges of sensor nodes.

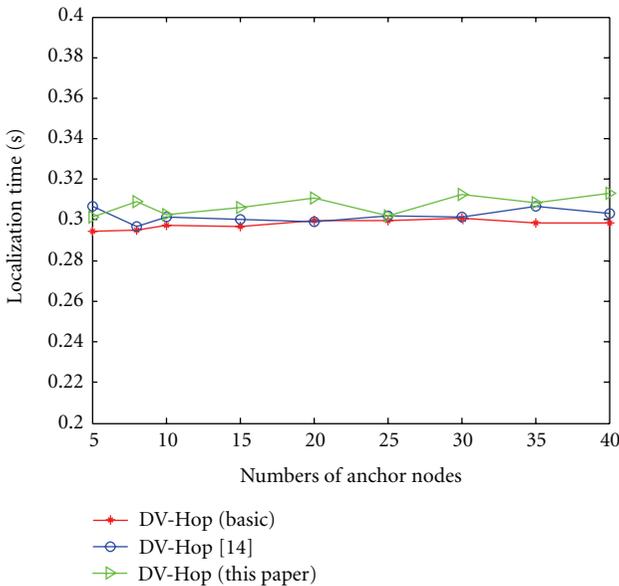


FIGURE 7: Localization time with different numbers of anchor nodes.

Firstly, some anchor nodes are distributed at the border land of the monitoring regions. Secondly, the average one-hop distance of anchor nodes is modified. Moreover, the average one-hop distance, which is used by each unknown node for estimating its own location, is modified by weighting N received average one-hop distances from anchor nodes. Finally, we use the PSO algorithm to correct the position estimated by the 2D hyperbolic location algorithm. It is shown that our proposed algorithm obviously improves the localization precision and stability. But it has very little increase in the computation time with respect to PSO.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (no. 10904073) and the Qing Lan Project.

References

- [1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor networks: a survey," *Computer Networks*, vol. 38, no. 4, pp. 393–422, 2002.
- [2] M. X. Chen and Y. D. Wang, "An efficient location tracking structure for wireless sensor networks," *Computer Communications*, vol. 32, no. 13-14, pp. 1495–1504, 2009.
- [3] P. Bahl and V. N. Padmanabhan, "RADAR: an in-building RF-based user location and tracking system," in *Proceedings of the 19th Annual Joint Conference of the IEEE Computer and Communications Societies (IEEE INFOCOM '00)*, pp. 775–784, March 2000.
- [4] N. B. Priyantha, A. Chakraborty, and H. Balakrishnan, "Cricket location-support system," in *Proceedings of the 6th Annual International Conference on Mobile Computing and Networking (MOBICOM '00)*, pp. 32–43, August 2000.
- [5] G. Mao, B. Fidan, and B. D. O. Anderson, "Wireless sensor network localization techniques," *Computer Networks*, vol. 51, no. 10, pp. 2529–2553, 2007.
- [6] D. Niculescu and B. Nath, "Ad hoc positioning system (APS) using AOA," in *Proceedings of the 22nd Annual Joint Conference on the IEEE Computer and Communications Societies*, pp. 1734–1743, April 2003.
- [7] N. Bulusu, J. Heidemann, and D. Estrin, "GPS-less low-cost outdoor localization for very small devices," *IEEE Personal Communications*, vol. 7, no. 5, pp. 28–34, 2000.
- [8] L. Doherty, K. S. J. Pister, and L. El Ghaoui, "Convex position estimation in wireless sensor networks," in *Proceedings of the 20th Annual Joint Conference of the IEEE Computer and Communications Societies*, pp. 1655–1663, April 2001.
- [9] J. F. Kurose and K. W. Ross, *Computer Networking: A Top Down Approach Featuring the Internet*, Addison Wesley, Longman, Reading, Mass, USA.
- [10] Z. Pin, X. Zhifu, and S. Yan, "Improved DV-hop algorithm based on WSN," *Chinese Journal Of Electron Devices*, vol. 32, no. 6, pp. 1091–1093, 2009.
- [11] S. Hongyu and Y. Sha, "An improved DV-Hop localization algorithm for wireless sensor network," *Electronics Optics & Control*, vol. 18, no. 4, pp. 93–96, 2011.
- [12] D. Gao, P. Chen, C. H. Foh, and Y. Niu, "Hop-distance relationship analysis with quasi-UGD model for node localization in wireless sensor networks," *EURASIP Journal on Wireless Communications and Networking*, vol. 2011, article 99, 2011.
- [13] S. W. Lee, D. Y. Lee, and C. W. Lee, "Enhanced DV-Hop algorithm with reduced hop-size error in ad hoc networks," *IEICE Transactions on Communications*, vol. 94, no. 7, pp. 2130–2132, 2011.
- [14] J. Z. Lin, X. B. Chen, and H. B. Liu, "Iterative algorithm for locating nodes in WSN based on modifying average hopping distances," *Journal on Communications*, vol. 30, no. 10, pp. 107–113, 2009.
- [15] W. Shi, C. Jia, and H. Liang, "An improved DV-Hop localization algorithm for wireless sensor networks," *Chinese Journal of Sensors and Actuators*, vol. 24, no. 1, pp. 83–87, 2011.
- [16] H. Chen, K. Sezaki, P. Deng, and H. C. So, "An improved DV-Hop localization algorithm with reduced node location

- error for wireless sensor networks,” *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, vol. 91, no. 8, pp. 2232–2236, 2008.
- [17] Y. Gao, W. S. Zhao, C. Jing, and W. Z. Ren, “WSN node localization algorithm based on adaptive particle swarm optimization,” *Applied Mechanics and Materials*, vol. 143-144, pp. 302–306, 2012.
- [18] Y. Liu, X. F. Tian, and Z. H. Zhan, “Research on inertia weight control approaches in particle swarm optimization,” *Journal of Nanjing University*, vol. 47, no. 4, pp. 364–371, 2011.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

