A Localization Method Based on Detection of Beacon Node Drift in Wireless Sensor Networks

Jun Wang*,[1], Jiajia Wang[1] , Tiansi Ren[1],Xun Chen[1], Ruifang Li[2], Gang Liu[3]

[1.]College of Agriculture Equipment Engineering, Henan University of Science and Technology, Luoyang, Henan 471003, China

[2.]College of Electrical and Mechanical Engineering, Zhengzhou University of Industrial Technology, Zhengzhou, Henan 451100, China

[3.]Key Laboratory for Modern Precision Agriculture System Integration Research, Ministry of Education, China Agricultural University, Beijing 100083, China

***Abstract* — A distributed detection method for beacon node drift is proposed to address the node localization problem concerning beacon node drift. This method automatically identifies possible drift beacon nodes by calculating the degree of variation in RSSI similarity at different times. A localization algorithm combining Kernel Principal Component Analysis (KPCA) and PSO-BP neural network is also proposed to solve the high dimensionality problem concerning localization data. First, KPCA is used to eliminate data dependency, extract the principal component containing localization information and reduce dimensionality of sample space. Then the PSO-BP neural network is trained using the extracted eigenvectors of nonlinear principal component as input sample and position coordinate of grid vertex as output sample to create a localization model. The simulation result shows that this algorithm is highly practical and outperforms traditional method in terms of drift detection and localization error.**

Keywords - Wireless sensor network; Beacon node drift; Similarity; Node Localization

I. INTRODUCTION

A wireless sensor network is a network system composed of a large number of static or moving sensor nodes deployed within sensor field in an autonomous or multi-hop manner, where the sensor nodes collaboratively sense, collect and process the information about the monitored object in the area covered by the network and send it to an observer[1,2]. Featuring rapid deployment, high invulnerability, long life cycle etc., a wireless sensor network does not need support from any fixed network[3]. It has a wide range of applications in complicated monitoring and tracking tasks.

In practice, since the number of nodes in wireless sensor networks is usually huge, and the nodes are often deployed randomly, it is difficult to measure the position of each node one by one at the time of deployment. However, node position is crucial for access to monitored information, because 1) the accuracy of node position information directly affects the validity of the data collected; 2) the prerequisite for implementing route discovery, maintenance and data forwarding based on geographic routing protocols is to obtain node position information. The method to directly obtain node position is to use the Global Positioning System (GPS), but due to many restrictions such as cost, size, power consumption and deployment environment, it is not realistic to provide a GPS receiver for each node, which necessitates the study of node localization techniques in wireless sensor networks.

According to localization mechanism, localization algorithms for wireless sensor networks can be classified as range-free or range-based. Range-free localization algorithms provide fuzzy localization by estimation only according to network connectivity, so they are less subject to environmental factors, but they provide lower localization accuracy and have higher requirements on anchor node density[4,5]. In contrast, range-based localization algorithms need to measure the actual distance between adjacent nodes or their orientation in order to calculate unknown node positions, which provide higher localization accuracy. Hence, for applications with high requirements on node position accuracy, range-based localization algorithms are commonly used, and their localization accuracy largely depends on the estimated distance between beacon nodes and unknown nodes.

In a traditional static wireless sensor network, beacon nodes as the basis for localization are all assumed to be static with stable localization performance. But in actual applications, due to various uncertain natural, human factors, or malicious localization attacks, beacon nodes are likely to move unexpectedly, or their localization performance may fluctuate dramatically, which is known as "drift". For such applications, repositioning the beacon nodes in cycles can correct the localization deviation caused by drift. Beacon nodes within the actual monitoring area are usually preset. After deployment, a beacon node uploads data packets containing its own node ID and position and communicates with unknown nodes. In this case, the repositioning process further spreads the position deviation once the beacon node drifts, thus affecting the quality of service throughout the network. Therefore, the study of localization problem concerning beacon node drift in wireless sensor networks are of great theoretical and application values.

II. RELATED WORK

To address the problem of node drift detection, Kuo et al. proposed a Beacon Movement Detection (BMD) algorithm to identify which beacon nodes in the network have their position passively changed[6]. The basic idea behind this is to set a BMD engine in the network to collect and process the RSSI (received signal strength indication) information throughout the network. This method can identify beacon node movement within a certain range of fault tolerance. BMD model is essentially a centralized algorithm for solving NP-complete problem, and for a heuristic algorithm used to solve NP-complete problem, a contradiction exists between computing speed and result accuracy. Ravi Garg et al. improved localization reliability by excluding the beacon nodes that provide larger downward gradient during node position computation, but without considering the reference function of ordinary node position, this method is not suitable for networks with sparse beacons and requires heavy computation[7]. Literature [8] proposed a distributed lightweight method based on reliability model to verify node position. By combining the observations of the relative positions of ordinary and beacon nodes, this method fully takes into account the observed reliability of different node types and utilizes reliability indicators to identify drift ordinary nodes and unreliable beacon nodes. However, this algorithm is more complex than those that merely consider beacon nodes in identification. Literature [9] proposed a beacon node drift detection algorithm based on negotiation scoring system to automatically find possible drift nodes. The distributed computing feature of this algorithm reduces communication overhead and node energy consumption, but its strong randomness during computation directly affects the accuracy of drift detection.

At present, there is little study directly on node repositioning in case of beacon node drift. Literature [10] proposed a localization algorithm to restore localization accuracy based on the localization technique of maximum likelihood estimation. This algorithm uses optimal noise error to check beacon nodes involved in localization estimation, and is able to effectively reduce incorrect localization caused by drift and restore the original localization technique of maximum likelihood estimation. Literature [11] proposed a secure localization algorithm, AtLoc, which uses unbiased estimate of variance as the basis for security check to find a minimum set of security reference with stochastic method. Then the predicted residuals based on this minimum set of security reference are used to check whether the remaining reference points are abnormal, thus improving the ability of the positioning system to tolerate drift. Literature [12] uses gradient descent method and anomaly detection techniques to realize high accuracy localization by filtering out drift data. The simulation result shows that this algorithm performs as expected and is able to achieve better localization performance with less computational resources. The above studies promote the exploration of node repositioning problem to some extent and can be used as a useful reference.

III. WIRELESS SIGNAL TRANSMISSION MODEL

Wireless signal is a kind of electromagnetic wave. Considering the received signal strength indicator of isotropic spherical wave, expressed in dBm, part of its energy is absorbed by the medium during propagation and its strength decays exponentially as distance increases[13]. Common wireless signal propagation path loss models include: free-space propagation model, two-ray ground-reflection model and log-distance path loss model. The log-distance path loss model is the most widely used one. A log-distance path loss model is composed of two parts. First is a pass loss model, which predicts the received signal power, expressed as $\overline{P_r(d)}$, when distance is equal to d. The close-to-center distance d_0 is used as reference. $P_r(d_0)$ is the received power at reference distance d_0, which is either measured or known. $\overline{P_r(d)}$ relative to $P_r(d_0)$ is calculated as follows:

$$\frac{P_r(d_0)}{\overline{P_r(d)}} = (\frac{d}{d_0})^{\beta} \tag{1}$$

Where β is the path loss exponent (pass loss index), which is usually an empirical value obtained from actual measurement and reflects the rate of path loss as distance increases. β depends mainly on the environment of wireless signal propagation, i.e. complex jamming in the air such as attenuation, reflection and multipath effect. Pass loss model normally uses dB as the unit of measurement, which is expressed as follows:

$$\frac{\overline{P_r(d)}}{P_r(d_0)} = -10\beta\log(\frac{d}{d_0}) \tag{2}$$

The second part of a log-distance path loss model is a Gaussian-distributed random variable $X_{\mathrm{dB}}(0,\sigma^2)$, which reflects the change of received power caused by noise jamming for a given distance. Thus the log-distance path loss model is expressed as follows:

$$\overline{P_r(d)} = P_r(d_0) - 10\beta\log(\frac{d}{d_0}) + X_{\mathrm{dB}} \tag{3}$$

Now RSSI has normal distribution with actual value as expected and σ as standard deviation, i.e. $\overline{P_r(d)} \sim N(P_r(d_0) - 10\beta\log(\frac{d}{d_0}), \sigma^2)$.

IV. BEACON NODE DRIFT DETECTION METHOD

A. Network model

Figure 1 is the schematic diagram of a network model, where a set of wireless sensor nodes $S=\{S_i|i=1,2,...,M\}$ are randomly deployed in a three-dimensional area ($a\times b\times c$). All nodes are isomorphic and their propagation range is a circle centered at their actual position having a radius of R, i.e. the communication radius of the nodes is R (R is larger than the area's diagonal line L). All nodes are classified as beacon or unknown according to their function in the localization system. The first n nodes $S_1(x_1,y_1)$, $S_2(x_2,y_2)$,..., $S_n(x_n,y_n)$, whose position can be obtained in advance through external devices such as GPS or actual known arrangement, are used as beacon nodes; nodes $S_i(x_i,y_i)$ ($n<i\le M$), whose position in the network is unknown and there is no special device available to obtain their information, are used as unknown nodes. For ease of discussion, this paper makes the following assumptions:

- Each sensor node has a unique ID;
- Spatial wireless signal transmission model is a perfect sphere;
- All sensor nodes are isomorphic having the same power and computing capability;
- All nodes are time-synchronized and can communicate with each other directly.

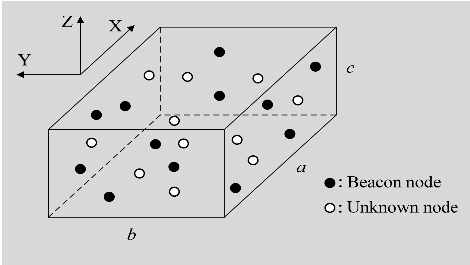


Figure 1. Schematic diagram of network model

The drift process of beacon nodes in network is shown in Figure 2, where beacon node A drifts after localization for some time, and the neighbor relationship between nodes changes accordingly, but the broadcast position information of beacon node *A*' remains unchanged.

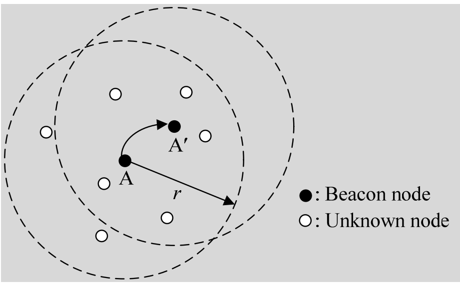


Figure 2. Drift process

The reliability of position information of a beacon node can be described as its degree of RSSI variation relative to other beacon nodes. The greater variation is, the more intense its relative movement to other beacon nodes will be, and the more likely it has drifted. When network deployment is completed, the beacon nodes communicate with each other and measure themselves according to the node identification mechanism to determine their possibility of drift. If deviation is less than threshold, they are marked as non-drift beacons, or otherwise as drift beacons.

Drift beacon nodes are excluded from the beacon set involved in localization. As time goes on, more and more beacon nodes in the wireless sensor network may start to drift and there will be less and less beacon nodes available, which directly affects the localization accuracy of unknown nodes. To avoid this, the position of drift beacon nodes should be updated after each node identification. These beacon nodes may be repositioned as unknown nodes, using estimated position as their new position.

B. Node identification mechanism

This paper employs an identification mechanism for beacon node drift based on RSSI similarity. The identification criteria for the reliability of current position of beacon nodes depend on their RSSI similarity relative to other beacon nodes over time. A higher similarity indicates more beacon nodes without significant RSSI variation and thus a higher reliability, whereas a lower similarity indicates a lower reliability.

Definition 1 A set of RSSI data for beacon nodes that communicate with other beacon nodes in a wireless sensor network, sampled in their ID sequence at t_1 is denoted as $(S,t_1)=[r_1,r_2,...,r_n]$, where n is the total number of beacon nodes, and the RSSI to which the ID of beacon node corresponds is denoted as 1.

Definition 2 $(S,t_1),(S,t_2), ...,(S,t_n)$ is a set of RSSI data with a fixed time interval, where (S,t_i) indicates a set of RSSI data S for beacon nodes that communicate with other beacon nodes sampled at t_i.

Definition 3 The function expression for beacon node RSSI similarity at adjacent sampling times is given as follows:

$$Gsim[(S,t_i),(S,t_{i+1})] = \sum_{j=1}^{n}\left(1-\frac{\left|(S,t_i)_j-(S,t_{i+1})_j\right|}{\left|(S,t_i)_j-(S,t_{i+1})_j\right|+m_j}\right)\Bigg/ n \quad (4)$$

Where $(S,t_i)_j$ is dimension j of the RSSI data set for beacon nodes that communicate with other beacon nodes at t_i; $(S,t_{i+1})_j$ is dimension j of the RSSI data set for beacon nodes that communicate with other beacon nodes at t_{i+1}; m_j is the absolute average of (S,t_i) and (S,t_{i+1}) on dimension j; $G_{sim} \in [0,1]$; n is the total number of beacon nodes.

The RSSI similarity time series of beacon nodes in a wireless sensor network may be expressed as a function with sampling interval t as independent variable and RSSI similarity G_{sim} as dependent variable. If RSSI similarity values at various time points are linearly distributed around a line, and this line segment is calculated from a simple linear regression model, then it is called a simple linear fitting regression line of RSSI similarity time series.

Considering regression function with the linear function of t,

$$d' = \beta_0 + \beta_1 t \tag{5}$$

Where t is the sampling time point, β_0 is the intercept of the line, β_1 is the slope of the line, d' is the fitted value at sampling time point t. By analyzing the dynamic linear relationship contained in the time series, the parameters of the simple linear regression model are obtained by least square fitting. The value of β_1 is obtained through expression (6).

$$\beta_1 = \frac{\sum_{i=1}^{n}(t_i - \overline{t})(d_i - \overline{d})}{\sum_{i=1}^{n}(t_i - \overline{t})^2} \tag{6}$$

Where $\overline{t}$ represents the average of the time period, $\overline{d}$ represents the average RSSI similarity of beacon nodes during the period. The value of β_0 can be further calculated from the value of β_1.

$$\beta_0 = \overline{d} - \beta_1 \overline{t} \tag{7}$$

According to the value of β_1, the intensity of data variation can be determined. If the difference between the fitted value d_i on the regression line fitted to the time series of RSSI similarity degree at $t_i(i=1,2,\ldots,n-1)$ and $d_i^{'}$ is smaller than the given threshold ε, the beacon node is considered to be a drift beacon, or it is not a drift beacon. For different networks, threshold ε can be set from an empirical value or actual measurement.

V. METHOD FOR LOCALIZING UNKNOWN NODES

A. Algorithm description

Unknown nodes obtain their estimated position using a specific calculation method by communicating with and estimating the distance between them and various beacon nodes. Due to the generally high deployment density of beacon nodes, localization data tend to have a high number of dimensions. This paper employs the Kernel Principal Component Analysis (KPCA) method to reconstruct original RSSI localization data. It removes some of the principal component and extracts major localization characteristic data through data dimension reduction and decorrelation, thus effectively eliminating redundant information and the effect of multicollinearity on regression accuracy and stability. For nonlinear relationship between localization characteristic information and unknown node coordinate, a localization model is constructed in conjunction with PSO-BP neural network.

Each time node identification is completed, the beacon set randomly selects a beacon node as sink node to divide the cubic area into $\frac{a}{L}\times\frac{b}{L}\times\frac{c}{L}$, $L \in Z^{+}$ virtual grids, the vertex of which is denoted as K_j ($j=1,2,\ldots,(L+1)^3$). Suppose that the actual distance between various beacon nodes and a certain unknown node is d_i, then d_i can form a distance vector $T=[d_1,d_2,\ldots,d_n]$. It can be demonstrated that a one-to-one nonlinear mapping relationship exists between the unknown node and distance vector T. This algorithm first obtains the theoretical distance from the grid vertex to various beacon nodes to form a corresponding distance vector; then employs KPCA to analyze the distance vector group, eliminate data dependency, extract the principal component containing localization information, and reduce the dimensions of the sample space; finally, it trains the PSO-BP neural network using the extracted non-linear characteristic vector of the principal component as input sample and the position coordinate of the grid vertex as output sample to obtain a localization model for further use in estimating the position of unknown nodes within the localization area.

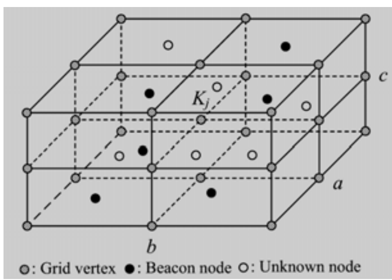


Figure 3. Virtual grid model

B. Extraction of kernel principal component

DOI 10.5013/IJSSST.a.17.49.12 ISSN: 1473-804x online, 1473-8031 print

characteristic

Kernel principal component analysis is a nonlinear characteristic extraction method, which maps input vector into a high-dimensional characteristic space by selecting a nonlinear mapping in advance to obtain better linear separability, and then performs linear principal component analysis on the mapping data in the high-dimensional space to obtain the nonlinear principal component of the data. For a given set of positioning sample data $x_k \in R^m$ (k=1,2,…,N and N=(L+1)^3), a nonlinear function $\varphi(g)$ maps the input data from the original space into the high-dimensional characteristic space F, the dimensions of which can be arbitrarily large, and it is supposed that in F, $\sum_{i=1}^{N} \varphi(x_i) = 0$, so the covariance matrix for the mapped data is expressed as follows:

$$C^F = \frac{1}{N}\sum_{i=1}^{N} \varphi(x_i)\varphi(x_i)^T \tag{8}$$

A principal component analysis on the mapped data is equivalent to a characteristic vector analysis on the matrix, i.e.:

$$\lambda v = C^F v \tag{9}$$

Where eigenvalue $\lambda \geq 0$, so Equation (8) can be expressed as follows:

$$C^F v = (\frac{1}{N}\sum_{i=1}^{N} \varphi(x_i)\varphi(x_i)^T)v = \frac{1}{N}\sum_{i=1}^{N} \langle \varphi(x_i), v\rangle \varphi(x_i) \tag{10}$$

Since characteristic vector v can be composed by vectors of the characteristic space, so ai exists so that:

$$v = \sum_{i=1}^{N} a_i \varphi(x_i) \tag{11}$$

From Equations (9)-(11), the following can be obtained:

$$\lambda \sum_{i=1}^{N} a_i \varphi(x_i) = \frac{1}{N}\sum_{i=1}^{N} a_i \sum_{j=1}^{N} \langle \varphi(x_j), \varphi(x_i)\rangle \varphi(x_j) \tag{12}$$

Left-multiplying both sides of Equation (12) to obtain:

$$\lambda \sum_{i=1}^{N} a_i \langle \varphi(x_k), \varphi(x_i)\rangle = \frac{1}{N}\sum_{i=1}^{N} a_i \sum_{j=1}^{N} \langle \varphi(x_k), \varphi(x_j)\rangle \langle \varphi(x_j), \varphi(x_i)\rangle \tag{13}$$

Where K=1,2,…,N.

Matrix K of $N \times N$ is defined as follows:

$$K_{ij} = K(x_i, x_j) = \langle \varphi(x_i), \varphi(x_j)\rangle \tag{14}$$

Then Equation (13) is simplified as follows:

$$N\lambda a = Ka \tag{15}$$

Where a=[a_1,a_2,…,a_N]^T. It can be seen from the above analysis that performing a principal component analysis in the characteristic space is equivalent to solving Equation (15), where eigenvalue $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_N$ and eigenvector $a_1, a_2, \ldots, a_N$ are used to obtain eigenvector v of the covariance matrix of the mapped data from Equation (11).

Principal component selection criteria

$$\left(\sum_{k=1}^{m} \lambda_k \Big/ \sum_{i=1}^{N} \lambda_i\right) \geq E \tag{16}$$

Where E is the cumulative rate of contribution, i.e. the ratio of the sum of the first m eigenvalues to the overall sum of the eigenvalues is greater than E. To ensure more localization characteristic information is obtained after characteristic extraction, the value of E is taken as ≥ 0.95.

C. *PSO-BP neural network localization algorithm*

PSO algorithm features an optimal solution for globally random search and a gradient descent for locally detailed search, as well as fast convergence speed[14]. This paper uses PSO algorithm to optimize the weights and thresholds of a BP neural network[15,16]. This algorithm is based on swarm iteration where a swarm searches by following the optimal particles in the solution space. The extracted principal component eigenvector is used as input sample and the position of grid vertex as out sample to train the PSO-BP neural network and create a localization model.

The rules for updating particle velocity and position are expressed as follows:

$$\begin{aligned} v_{id}(t+1) = w \cdot v_{id}(t) + c_1 rand()(p_{id}(t) - x_{id}(t)) \\ + c_2 rand()(p_{gd}(t) - x_{id}(t)) \end{aligned} \tag{17}$$

$$x_{id}(t+1) = x_{id}(t) + v_{id}(t+1) \tag{18}$$

Where v_{id} (t+1) is the velocity on dimension d at iteration t+1 of particle i; $P_{id}(t)$ is the individual optimal solution at iteration t of particle i; $P_{gd}(t)$ is the optimal solution of the entire particle swarm at iteration t; x_{id} is dimension d of particle i; c_1 and c_2 are acceleration constants; *rand*() is a random number between 0-1; w is inertia weight.

The algorithm for a PSO-BP neural network model is as follows:

- Initialize the weights and thresholds of the BP neural network. $x_i =(x_{i1},x_{i2},\ldots,x_{id})$ represents one particle; each dimension in the vector represents the size of a weight or threshold; d is the sum of all weights and thresholds in the BP neural network; $d=r\times p+p+p\times s+s$, where r, p and s are the number of nodes in respectively the input, hidden and output layers of the BP neural network.
- Set the parameters of the particle swarm. Initialize the maximum and minimum values of inertia weight w and the values of acceleration constants c_1 and c_2; give parameters such as particle swarm size R and maximum number of iterations M;
- Calculate the fitness of each particle, i.e. for each particle, calculate the actual output values of all samples in the forward direction of the BP network to obtain the mean square error.
- For each particle, compare its fitness value with the one at the optimal position P_{id} it passes through; if the former is better, then P_{id} is updated; for each particle, compare its fitness value with the one at the optimal position P_{gd} the swarm passes through; if the former is better, then P_{gd} is updated;
- Update the velocity and position of each particle according to Equations (17) and (18);
- If the algorithm satisfies the convergence criteria or reaches the maximum number of iterations, exit PSO algorithm and go to Step 7, or return to Step 3;
- Continue to train the neural network using BP algorithm. If the training result is better than PSO training result, the BP neural network is outputted, or the PSO-trained neural network is outputted.

VI. EXPERIMENT AND SIMULATION

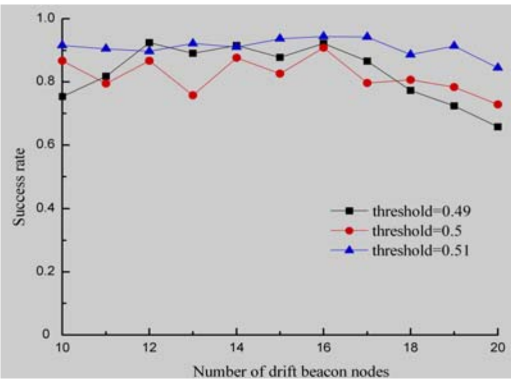
To test the performance of the algorithm, a simulation test is conducted on the localization algorithm proposed. The simulation scenario is set as follows: 1) three dimensional test area: 100 m ×100 m ×50 m; 2) total number of nodes: 100; maximum drift distance of beacon node: 20 m; 3) transmitting signal strength Pt of unknown and beacon nodes: 30 dBm; reference distance d_0: 20 m; transmitting antenna gain G_t and receiving antenna gain G_r: 1dBi; path loss exponent n: 2.

The two performance indicators for the beacon node drift identification algorithm are success rate $\left(\mathrm{Num}(B_M \cap B_{MD}) / \mathrm{Num}(B_M)\right)$ and error rate $\left(\mathrm{Num}((U-B_M)\cap B_{MD}) / \mathrm{Num}(B_M)\right)$, where B_M is the set of actual drift beacon nodes, B_{MD} is the set of identified drift beacon nodes, and U is the set of all beacon nodes. The success rate is the ratio of the number of correctly identified drift beacon nodes to the number of actual drift beacon nodes; the error rate is the ratio of the number of correctly identified drift beacon nodes to the number of actual drift beacon nodes.

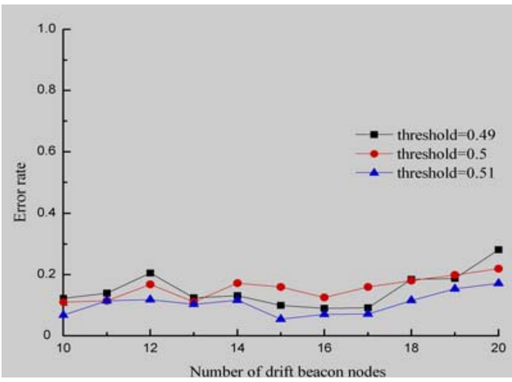
Localization error is the criteria for measuring the accuracy of localization algorithm, which is defined as the distance between the coordinate of an unknown node estimated by the localization algorithm and its actual coordinate: $\left(\sqrt{(x_i-x_e)^2+(y_i-y_e)^2+(z_i-z_e)^2}\right)$, where the estimated coordinate is (x_e, y_e, z_e), and the actual coordinate is (x_i, y_i, z_i).

A. Simulation test for beacon node drift identification algorithm

Identification of a drift beacon node is done by analyzing its RSSI similarity relative to other beacon nodes within a certain period, and only if the difference between the fitted value on the regression line fitted to the time series of RSSI similarity and the actual value is greater than the threshold will it be identified as a drift beacon node. For this reason, it is crucial to select a proper threshold. After many experiments, the range of threshold is determined at about 0.5, and the values around 0.5 are used for simulation. There are 10 elements in the RSSI data set. As can be seen from Figure 4, with decreasing threshold, the success rate of beacon node drift identification algorithm continues to rise, while the error rate is continuously increasing. The smaller threshold is, the more sensitive it will be to RSSI variation, and the easier it will be to incorrectly identify a beacon node with smaller drift. When the threshold is 0.51, the success rate is 91.09%, and the error rate is 10.51%, which gives good overall result, so threshold ε is selected as 0.51.



(a)



(b)

Figure 4. Effect of threshold on identification of drift beacon node

An appropriate number n of elements in RSSI data set helps shorten the running time of the algorithm and improve the accuracy of identification. A too small or large number of elements often leads to sensitive or sluggish response to the fluctuation in RSSI similarity, thus increasing the likelihood of incorrect identification and reducing the robustness of the algorithm. The number of elements is set to vary from 5 to 15 continuously to reflect its effect on the performance of the algorithm by comparing success rate and error rate. It can be seen from Figure 5 that as the number of elements increases, at first the success rate of the algorithm increases continuously while the error rate declines continuously, but as the number of elements continues to increase, the success rate of the algorithm begins to decline, while the error rate begins to rise. Either a too small or large number of elements can easily lead to incorrect identification, so it can be observed that when the number of elements n in the RSSI data set is selected as 9, the performance of the algorithm is most stable.

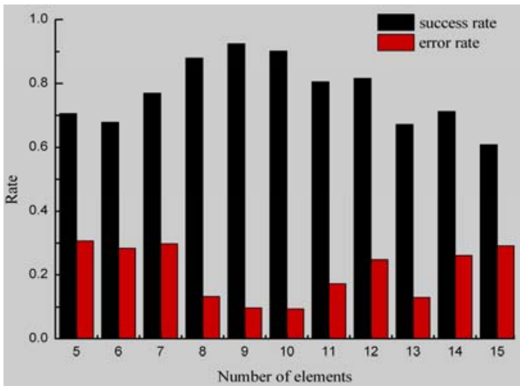


Figure 5. Effect of number of elements on identification of drift beacon nodes

Threshold ε is taken as 0.51; the number of elements n in the RSSI data set is taken as 9; the number of drift beacon nodes is taken from 0 to 20. With other conditions unchanged, the drift beacon node identification algorithm proposed in this paper is compared with BMD algorithm, and the average success rates of these two algorithms are respectively 92.23% and 80.05%, average error rates are respectively 9.95% and 12.47%. This shows that the algorithm proposed in this paper outperforms BMD algorithm in terms of success rate and error rate.

DOI 10.5013/IJSSST.a.17.49.12 ISSN: 1473-804x online, 1473-8031 print

B. Simulation test for localization algorithm

To verify the performance of the algorithm, the simulation test is divided into two parts. 1) Drift beacon node identification is not performed, and localization is performed directly using the unknown node localization method, i.e. unreliable beacon node positions are possibly used to localize unknown nodes, which is called the traditional method. 2) Drift beacon node identification is performed, and drift beacon nodes are excluded from the node set, which is called the method of discarding drift beacons. Simulation environment settings: Particle size R=50, acceleration constants $c_1=c_2=1.4962$, max. inertia weight $w_{max}=1$, minimum inertia weight $w_{min}=-1=-1$, maximum number of iterations M=50, convergence accuracy $\varepsilon=10^{-6}$, virtual grid size 5 m × 5 m. The localization result is the average value from 100 simulations using the same parameters. As can be seen from Figure 6, the traditional method results in a large localization error while the method of discarding drift beacons can effectively improve localization accuracy and perform well in error compensation as well as environment adaptability.

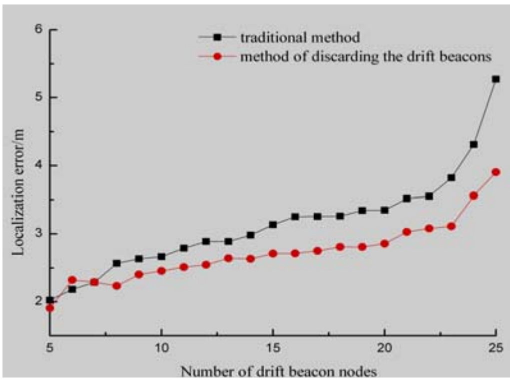


Figure 6. Comparison of traditional method and method of discarding drift beacons

Under the same conditions of simulation, the localization algorithm proposed in this paper is compared with the cross particle swarm localization algorithm described in Literature [17]. The simulation result is shown in Table 1. It can be seen that the localization algorithm proposed in this paper has better performance and it can properly handle the effect of drift beacon nodes, give better localization result, and have some practical value in wireless sensor networks.

TABLE I. Comparison of localization result

Localization error	Method of discarding the drift beacons	cross particle swarm optimization algorithm
Maximum/m	3.9042	5.2373
Minimum/m	1.9053	2.5615
Average value/m	2.7272	3.7546

VII. CONCLUSION

During localization in a wireless sensor network, unexpected movement of beacon nodes or violent fluctuation in localization performance leads to inconsistency between position reference information and actual position, thus affecting the quality of localization service throughout the network. This paper proposes a drift beacon node identification mechanism based on RSSI similarity, where the beacon nodes communicate with each other and determine the possibility of drift according to the degree of RSSI variation. In addition, a localization algorithm combining KPCA and PSO-BP neural network is also proposed to solve the high dimensionality problem concerning localization data. First, the principal localization characteristic data is extracted through data dimension reduction and decorrelation. Then the PSO-BP neural network is used for modeling to create a localization model based on the nonlinear relationship between the localization characteristic information and unknown node coordinate. The simulation test shows that the drift beacon node identification mechanism and node localization algorithm proposed in this paper are feasible and outperform other algorithms to some extent. The next step is to introduce auxiliary algorithms to adapt to range-free localization scenarios.

VIII. ACKNOWLEDGEMENTS

This work is supported in part by Key Project of Henan

Tobacco Company (HYKJ201316)，Basic research project of the Education Bureau of Henan Province (17A416002) and Innovation Ability Foundation of Natural Science (Grant No. 2015GJB008) of Henan University of Science and Technology.

REFERENCES

[1] Patwari N, Ash J N, Kyperountas S, et al. Locating the nodes: cooperative localization in wireless sensor networks[J]. IEEE Signal Processing, 2005, 22(4): 54-69.

[2] Biswas P, Lian T C, Wang T C. Semi-definite programming based algorithms for sensor localization[J]. ACM Trans Sensor Networks, 2006, 2(2): 188-200.

[3] Kushki A, Plataniotis K, Venetsanopoulos A. Intelligent dynamic radio tracking in indoor wireless local area networks[J]. IEEE Transactions on Mobile Computing, 2010, 9(3): 405-419.

[4] Ahn H S, Yu W. Environmental adaptive RSSI based indoor localization[J]. Automation Science and Engineering, 2009, 6(10): 626-633.

[5] Gurrieri L E, Willink T J, Petosa A, et al. Characterization of the angle, delay and polarization of multipath signals for indoor environments[J]. Antennas and Propagation, 2008, 56(8): 2710-2719.

[6] KUO S P, KUO H J, TSENG Y C. The beacon movement detection problem in wireless sensor networks for localization applications[J]. IEEE Transactions on Mobile Computing, 2009, 8(10): 1326-1338.

[7] GARG R, VARNA A L, WU M. An efficient gradient descent approach to secure localization in resource constrained wireless sensor networks[J]. Information Forensics and Security, IEEE Transactions on, 2012, 7(2): 717-730.

[8] MAO K J, JIN H B, MIAO C Y, et al. Sensor location verification scheme in WSN[J]. Chinese Journal of sensors and actuators, 2015, 28(6): 850-857.

[9] ZHAO X M, ZHANG H Y, JIN Y, et al. Node localization scheme in wireless sensor networks under beacon drifting scenes[J]. Journal on Communications, 2015, 36(2): 2015032-1 -2015032-10.

[10] JIN H X, CAO J, WU D. Localization accuracy restoring algorithm under accuracy deterioration for WSN[J]. Chinese Journal of Scientific Instrument, 2011, 32(7): 1590-1597

[11] YE A Y, MA J F. Attack-tolerant node localization in wireless sensor networks[J]. Journal of Wuhan University of Technology, 2008, 30(7): 111-115.

[12] LUO Z, LIU H L, XU K. A secure localization algorithm against non-coordinated attack of malicious node [J]. Chinese Journal of sensors and actuators, 2013, 26(12): 1724-1727.

[13] CAPKUN S, HUBAUX J P. Secure positioning in wireless networks[J]. IEEE Journal on Selected Areas in Communications, 2006, 24(2): 221-232.

[14] BAO H, ZHANG B X, LI C. Mobile anchor assisted particle swarm optimization (PSO) based localization algorithms for wireless sensor networks[J]. Wireless Communications & Mobile Computing, 2012, 12(15): 1313-1325.

[15] GHOLAMI M, CAI N, BEMNAN R W. An artificial neural network approach to the problem of wireless sensors network localization[J]. Robotics and Computer-Integrated Manufacturing, 2012, 29(1): 96-109.

[16] HACKMANN G, SUN F, CASTANEDA N, et al. A holistic approach to decentralized structural damage localization using wireless sensor networks[J]. Computer Communications, 2012, 36(1): 29-41.

[17] Jun Wang, Fu Zhang, Xin Jin, et al. Localization method of agriculture wireless sensor networks based on rough set and artificial fish swarm algorithm[J]. International Agricultural Engineering Journal, 2015, 24(2): 95-103.