

The Fingerprint Positioning Algorithm Based on Gaussian Distribution for Wireless Sensor Network

Jiangyu Yan^{1, a}

¹School of Electrical and Electronic Engineering, North China Electric Power University, Beijing, 102203, China

^ayabjy@ncepu.edu.cn

Abstract

It is important to determine the location of nodes in a wireless sensor network for a location-based service. In this paper, we proposed a fingerprint positioning algorithm based on Gaussian distribution. The algorithm establishes a fuzzy decision matrix between the reference node and the unknown node according to the multi-dimensional Gaussian distribution characteristics of the wireless signal in the positioning area, then constructs the ideal node, calculates the proximity between the ideal node and the different reference nodes based on the Mahalanobis distance, and finally uses the centroid method to estimate the unknown node location. The simulation experiment shows that the relative positioning error of the algorithm is low, and the positioning accuracy is high.

Keywords

Wireless sensor network; Fingerprint positioning; Multi-dimensional Gaussian distribution.

1. Introduction

As an important support network for physical information perception and collection, Wireless Sensor Network (WSN) is widely used in target tracking [1], smart home [2], and other fields. As one of the key issues in wireless sensor network research, sensor location information acquisition has received extensive attention. Existing WSN positioning algorithms are mainly based on ranging, including Angle of Arrival (AOA) [3], Time of Arrival (TOA), and Time of Arrival Difference (TDOA) [4]. In recent years, the rapid development of wireless communication technology makes RSSI (Received Signal Strength Indication) measurement more convenient and accurate. As a result, many RSSI-based fingerprint positioning algorithms have emerged [5].

RSSI fingerprinting positioning typically divides the positioning process into two stages [6]: offline learning and online positioning. During the offline learning stage, the anchor node sends wireless signals. We measured the RSSI value from multiple anchor nodes on different reference nodes to establish the original fingerprint database. In the online positioning stage, the RSSI information measured by the unknown node is matched with the fingerprint database to predict the location of the unknown node. In this paper, we proposed a fingerprint positioning algorithm based on Gaussian distribution according to the spatial distribution characteristics of wireless positioning signals.

2. Build an Offline Fingerprint Database

The foundation of the fingerprint positioning algorithm is to establish the mapping relationship between RSSI and known coordinates, namely, the fingerprint database. First, we install m

anchor nodes in the positioning area and accurately measure the coordinates of the anchor nodes. The anchor node is a physical node that can transmit wireless signals. Then we set n ($n > m$) reference nodes with the middle-spacing grid in the positioning area. During the offline learning stage, the RSSI of the different anchor nodes is measured at a reference node, forming a signal fingerprint of the coordinates in which the node is located.

In order to reduce interference in the indoor environment, we use multiple sampling and compute the mean value to obtain a more accurate RSSI. Suppose the number of samples is M at each reference node. Then, the RSSI vector of the m anchor nodes measured at the k th by reference node j is $rssi_{jk} = (rssi_{jk}^1, rssi_{jk}^2, \dots, rssi_{jk}^m)$. The mean of RSSI vector for anchor node j for anchor node i measured by reference node j is:

$$rssi_j^i = \frac{1}{M} \sum_{k=1}^M rssi_{jk}^i \quad j \in n, \quad i \in m \quad (1)$$

So, we get the RSSI fingerprint vector for reference node j :

$$rssi_j = (rssi_j^1, rssi_j^2, \dots, rssi_j^m) \quad (2)$$

The coordinate of reference node j is (x_j, y_j) . RSSI measurements are made at each reference node to obtain a fingerprint data matrix for the positioning area A .

$$A = (x_j, y_j, rssi_j)_{n \times (m+2)} = \begin{bmatrix} x_1, y_1, rssi_1^1, rssi_1^2, \dots, rssi_1^m \\ x_2, y_2, rssi_2^1, rssi_2^2, \dots, rssi_2^m \\ \vdots \quad \quad \quad \vdots \quad \quad \quad \vdots \\ x_n, y_n, rssi_n^1, rssi_n^2, \dots, rssi_n^m \end{bmatrix}$$

3. Online Position Algorithm

After establishing the RSSI fingerprint database, the unknown node can be located online. During the online positioning stage, the unknown node measures the RSSI of each anchor node, and fuses it by the fuzzy method with the RSSI of the reference node in the fingerprint database. Then, the coordinate differences between its position and the reference node are computed. Based on these differences, we estimate the coordinate of the unknown node, finally.

Unknown node x measures the signal strength of each anchor node and obtains a vector $rssi_x = (rssi_x^1, rssi_x^2, \dots, rssi_x^m)$. For anchor node i , the signal strength measured by unknown node x is $rssi_x^i$, the signal strength at reference node j in the fingerprint database $rssi_j^i$. Then, the fuzzy approximation of the two values is called proximity v_j^i , which expressed as:

$$v_j^i = \frac{\min(rssi_x^i, rssi_j^i)}{\max(rssi_x^i, rssi_j^i)} \quad (3)$$

If $v_j^i \rightarrow 1$, this dictates that the distance between unknown node x and reference node j to anchor node i is approximately equal. If $v_j^i \rightarrow 0$, this dictates that the distance between node x and reference node j is far apart.

For anchor node i , we calculate the proximity vector V_i of unknown node x to all reference nodes based on formula (3):

$$V_i = (v_1^i, v_2^i, \dots, v_n^i)^T \quad (4)$$

V_i reflects how close the unknown node x is to all reference nodes under the signal of anchor node i . For example, there are two reference nodes, k and j . If $v_j^i > v_k^i$, this dictates that the unknown node x is closer to the reference node j . We calculate the proximity of unknown node x for m anchor nodes and establish fuzzy positioning decision matrix D :

$$D = (V_1, V_2, \dots, V_m) = \begin{bmatrix} v_1^1 & v_1^2 & \dots & v_1^m \\ v_2^1 & v_2^2 & \dots & v_2^m \\ \vdots & \vdots & \ddots & \vdots \\ v_n^1 & v_n^2 & \dots & v_n^m \end{bmatrix} \quad (5)$$

3.1. The Multi-dimensional Gaussian Positioning Algorithm

In the indoor environment, the anchor node position remains unchanged and transmits wireless signals with constant power. Because the noise and electromagnetic interference are relatively stable, the RSSI from the same anchor node in the same location meets the Gaussian distribution [7]. Therefore, the RSSI from the same anchor node measured at all reference nodes can be considered to meet the one-dimensional Gaussian distribution. Moreover, all anchor nodes are independent of each other, and there is no data exchange. Therefore, according to the central limit theorem, we consider the positioning environment subjects to multi-dimensional Gaussian distribution.

3.2. Multi-dimensional Gaussian Distribution

Set the multi-dimensional Gaussian distribution variable $X = (x_1, x_2, \dots, x_d)^T$, and $X \sim N(\mu, \Sigma)$, so the probability density function of X is

$$f(X) = \frac{1}{(2\pi)^{d/2} |\Sigma|^{1/2}} \exp\left[-\frac{1}{2}(X - \mu)^T \Sigma^{-1}(X - \mu)\right] \quad (6)$$

$\mu = (\mu_1, \mu_2, \dots, \mu_d)^T$ is the mean of each dimension variable, Σ is a covariance matrix that describes the correlation between dimensions.

Mahalanobis distance describes the differences in sample data to eliminate the scale differences. For example, assuming that x and y are two samples in total X , the difference between them can be measured by Mahalanobis distance:

$$d_M(x, y) = \sqrt{(x - y)^T \Sigma^{-1}(x - y)} \quad (7)$$

Σ the covariance matrices of x and y :

$$\Sigma = \frac{1}{d} \sum_{i=1}^d (x_i - y_i)(x_i - y_i)^T \quad (8)$$

3.3. The Deviation between the Unknown Node and the Reference Node

In the decision matrix D , the dimension is the number of anchor nodes. A column vector $V_i = (v_1^i, v_2^i, \dots, v_n^i)^T$ represents how close an unknown node under the i dimension is to the reference node. A row vector $f_j = (v_j^1, v_j^2, \dots, v_j^m)$ $1 \leq j \leq n$ indicates how close the reference node j is to an unknown node under all dimensions. Thus, estimating the location of the unknown node requires selecting the closest reference nodes.

Construct an optimal ideal node that should take the optimal value on each column vector, i.e., the closest unknown node under each dimension. The ideal node is $f^* = (v_j^*, v_j^*, \dots, v_j^*)$, where

$$v_k^* = \max(v_1^k, v_2^k, \dots, v_n^k) \quad (9)$$

The decision matrix D is a multi-dimensional Gaussian matrix composed of n m -dimensional samples.

The distance between each sample and the ideal node is calculated using the formula (7), i.e., the deviation value λ_j of the ideal node from the reference node

$$\lambda_j = \sqrt{(f_j - f^*)^T \mathbf{B}^{-1} (f_j - f^*)} \quad (10)$$

Wherein, $\mathbf{B}$ is the covariance matrix of the sample, calculated from the formula (8).

3.4. Estimate Unknown Node Coordinate

The deviation value λ_j is arranged in descending order, and the reference nodes for the first K items are connected into a polygon. Its centroid is the estimated location of an unknown node [8]. Thus, the estimated coordinates are calculated as follows:

$$(x^*, y^*) = \left(\frac{\sum_{j=1}^K \lambda_j x_j}{\sum_{j=1}^K \lambda_j}, \frac{\sum_{j=1}^K \lambda_j y_j}{\sum_{j=1}^K \lambda_j} \right) \quad (11)$$

Wherein, (x^*, y^*) is the estimated coordinates of the unknown node, the coordinates (x_j, y_j) of the reference node j are obtained from fingerprint database A .

4. Simulation Results

In order to evaluate the positioning performance of the proposed algorithm, we simulated and verified the algorithm on MATLAB 2018. The positioning simulation area is shown in Figure 1, with a size of 120m × 120m. Reference nodes are distributed in a grid. Anchor nodes and unknown nodes are set randomly, and the anchor node location is known.

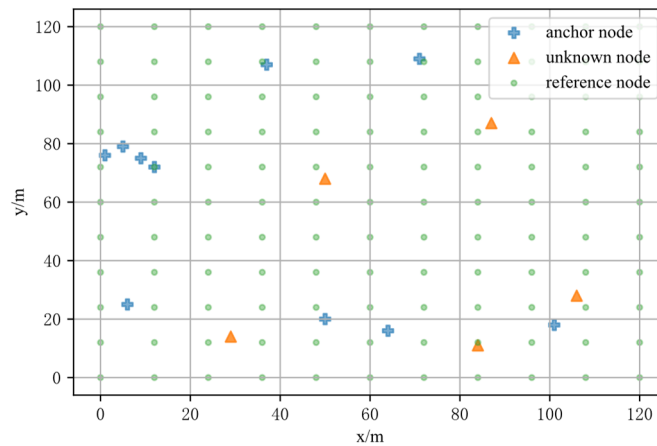


Figure 1. The positioning area diagram

The distance between the actual coordinates (x_k, y_k) of the unknown node k and the estimated coordinates (x_k^*, y_k^*) is defined as the positioning error.

$$err_k = \sqrt{(x_k - x_k^*)^2 + (y_k - y_k^*)^2} \quad (12)$$

The ratio of the average positioning error of l unknown nodes to the maximum communication radius of the anchor node is treated as a relative positioning error (RPE):

$$RPE = \frac{\sum_{k=1}^l err_k}{Ml} \quad (13)$$

M is the maximum communication radius of the anchor node.

4.1. Reference Node Interval

Reference nodes are evenly deployed in the positioning area. Changing their intervals affects positioning accuracy. Figure 2 shows the relationship between the RLE and the reference node interval. The reference node interval expansion will reduce the number of reference nodes around the unknown node, result in lower positioning accuracy.

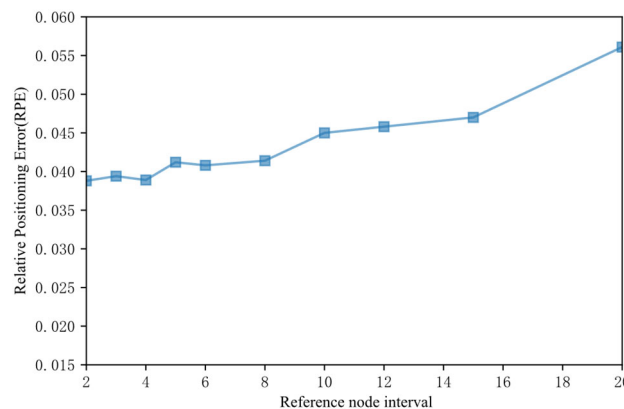


Figure 2. The reference node interval with RPE

From Figure 2, it can be obtained that when the interval is 5 to 20m, the RPE of the algorithm increases with the increase of the reference node interval, which means that the positioning accuracy is reduced. On the other hand, when the interval is less than 5m, the adjustment of the reference node interval has little effect on the RPE. Therefore, considering the accuracy and deployment cost, we think that the reference node interval should select 10m in practical applications.

4.2. The Scale of the Anchor Node

The anchor node ratio is the ratio of the number of anchor nodes to the total number of nodes. Adjusting the anchor node scale affects positioning accuracy. In addition, increasing the ratio of anchor nodes will increase deployment costs.

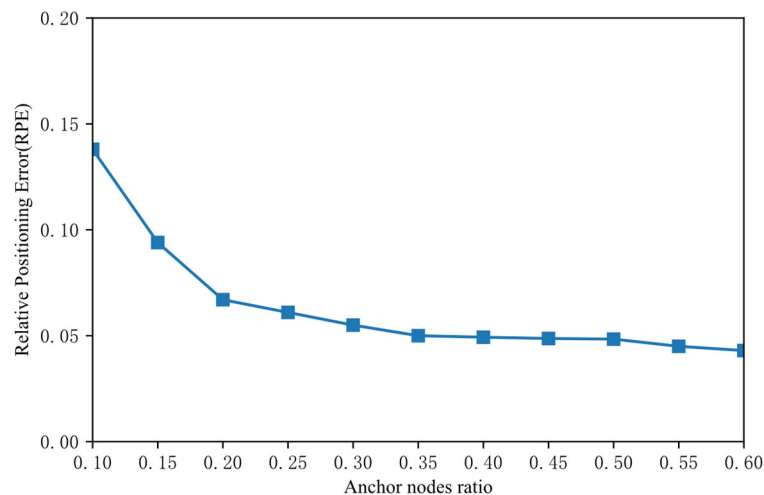


Figure 3. RPE with anchor nodes ratio

As shown in Figure 3, when the anchor node ratio is 0.45, the RPE of the proposed algorithm is 0.05. On the other hand, when the anchor node ratio exceeds 0.45, the continuous increase in the number of anchor nodes has little effect on reducing positioning error.

5. Conclusion

From the above simulation results, it can be concluded that the algorithm proposed in this paper has improved positioning accuracy in WSN. Furthermore, the relative positioning error of the algorithm is small, so it has good practicability.

References

- [1] Calafate C T, Lino C, Diaz-Ramirez A, et al. An integral model for target tracking based on the use of a WSN[J]. *Sensors*, 2013, 13(6): 7250-7278.
- [2] Huiyong W, Jingyang W, Min H. Building a smart home system with WSN and service robot[C]//2013 Fifth International Conference on Measuring Technology and Mechatronics Automation. IEEE, 2013: 353-356.
- [3] Maddumabandara A, Leung H, Liu Minxiang. Experimental evaluation of indoor localization using wireless sensor networks[J]. *IEEE Sensors Journal*, 2015, 15(9): 5228-5237.
- [4] Yin Jihao, Wan Qun, Yang Shiwen, et al. A simple and accurate TDOA-AOA localization method using two stations[J]. *IEEE Signal Processing Letters*, 2016, 23(1): 144-148.
- [5] Li G, Geng E, Ye Z, et al. Indoor positioning algorithm based on the improved RSSI distance model[J]. *Sensors*, 2018, 18(9): 2820.

- [6] Fang X, Nan L, Jiang Z, et al. Fingerprint localisation algorithm for noisy wireless sensor network based on multi-objective evolutionary model[J]. Iet Communications, 2017, 11(8): 1297-1304.
- [7] Xue W, Qiu W, Hua X, et al. Improved Wi-Fi RSSI measurement for indoor localization[J]. IEEE Sensors Journal, 2017, 17(7): 2224-2230.
- [8] GU G, YOU F. Improved centroid localization algorithm based on RSSI[J]. Transducer and Microsystem Technologies, 2018, 37(2): 147-153.