文章编号:1002-1175(2011)03-0382-07

# Localization algorithm without distance information for wireless sensor networks *

JIANG Zhi-Peng[1†], GAO Sui-Xiang

(*Department of Mathematical Sciences*, *Graduate University*, *Chinese Academy of Sciences*, *Beijing* 100049, *China*)

(Received 14 June 2010; Revised 10 August 2010)

**Jiang Z P, Gao S X. Localization algorithm without distance information for wireless sensor networks[J]. Journal of the Graduate School of the Chinese Academy of Sciences, 2011, 28(3):382 – 388.**

**Abstract** Most of existing localization algorithms of wireless sensor networks need the distance information between nodes and have many drawbacks. We propose a localization algorithm that does not need the distance information between nodes, and those drawbacks can be avoided. This algorithm only needs two anchors, and the locations of the anchors can be arbitrary. Simulation results show that this algorithm has good performance.

**Key words** wireless sensor networks, localization approach, rotation beacon

**CLC** TN929.5, TP212.9

Wireless sensor network (WSN) has received tremendous attention from both academia and industry in recent years because of their promise of numerous potential applications in both civil and military areas. Localization is one of the key technologies of WSN. In many applications (e. g., environment monitoring, habitat monitoring, battle-field surveillance and enemy tracking) it is necessary to accurately locate the nodes in a global coordinate system in order to report data that are geographically meaningful. Furthermore, some network layer protocol such as geographic routing often rely on location information.

Previous work shows that most localization algorithms are based on the distance between nodes. The methods to get the distance between two nodes can further be divided into distance-measured and distance-estimated. RSSI (received signal strength indication)[1] is a typical distance-measured method to get the distance between two nodes. In theory, the energy of a radio signal diminishes with the square of the distance from the signal's source. As a result, a node listening to a radio transmission should be able to use the strength of the received signal to calculate its distance from the transmitter. The accuracy of the RSSI range measurements is highly sensitive to multi-path, fading, non-line of sight scenarios, and other sources of interference, this method may result in large errors. These errors can propagate through all subsequent triangulation computations, leading to large localization error. Thus, the RSSI ranging measurement is not suitable for wireless sensor networks in many applications. Another range-based distance-estimated method

---

* Supported by the National Natural Science Foundation of China (10831006, 10671024), Important Knowledge Innovation Project of Chinese Academy of Sciences (kjcx-yw-s7)

†E-mail: lstjzp07@yahoo.com.cn

TDOA[1] (time difference of arrival) make use of the relative time differences among sensors. To obtain the relative time difference, each sensor needs to first get the dominant frequency of acoustic spectrum, and then broadcast this information. This may involve collaborative signal processing, such as FFT, which will increase the computational overhead. On the other hand, the audio data exchanged among sensors requires more network bandwidth, a very precious resource in wireless sensor network. So it's also not suitable for the WSN in many applications[2].

Distance-estimated methods use Hop count, network connectivity and other properties of the network and nodes to estimate the distances between nodes, they are more suitable for many WSN applications. There have been many works on this sort of methods.

The authors of Ref. [3] proposed the Amorphous algorithm, this algorithm uses the communication radius as the average hop distance. The product of the hop counts between two nodes and the average hop distance is as the estimated distance between the two nodes. In the DHL algorithm[4], the average hop distance is approximately equal to the multiply product of the communication radius and a coefficient $\mu$. The node density is presented by the node's local connectivity which is the number of neighbors of the nodes. The coefficient $\mu$ is relevant to the node density. The greater the node density is the greater $\mu$ is. In Ref. [5], the author showed that the error of the estimated distance is not only relevant to the node density but also relevant to the distance between the unknown node and the anchors (Here the distance refers to the hops). The farther they are apart from each other the greater the error is. HCRL algorithm[6] uses the ratios of anchor-to-node1 hop-count and anchor-to-node 2 hop-count as the ratios of anchor-to-node 1 distance and anchor-to-node 2 distance. Through the ratio information, we can get a circle. The intersection of all circles is the estimated position of the unknown nodes.

All the methods and algorithms mentioned above and some other methods and algorithms[7-11] are about how to improve the distance-estimated methods. Nevertheless, they can not guarantee the accuracy in theory though sometimes we can get a fine result by them. There are some reasons that can generate a poor estimation result, such as, the network scale is so large, the network is so sparse and the connectivity of the network is poor. The amount and density of anchors can also affect the result. In many applications, people often improve the accuracy of the result by increase and density of anchors. In addition, in the localization process almost every sensor node has to send and receive messages many times which may cost much battery energy.

Both the two kinds of methods to get the distance between nodes mentioned above have some drawbacks which make them have some limitations in practice. We propose a localization algorithm that does not need distance information between nodes.

This paper is organized as follows: In section 1, we introduce a method of calculating polar angle which we will use in our algorithm; In Section 2, we elaborate the basic idea of our localization algorithm; In Section 3, we do analysis and performance evaluation; In Section 4 we give the conclusion.

## 1 Related work

In Ref. [12] the authors proposed a simple localization algorithm which used the polar angle of nodes in a polar system. The method to calculate polar angle is as follows:

There is only one anchor TA in the network. TA transmits a rotation beacon which is uniform with a period of $T$ time units on a channel $\mu$. Every time the beacon coincides with the polar axis, the TA transmits a synchronization signal on a channel $\lambda$. Every sensor node listens to channel $\lambda$ for $T$ time units. Let $t_0$ be the moment at which it hears the synchronization beacon. At that point it switches to channel $\mu$, on which the rotating beacon is transmitted. Assume that the rotating beacon is received by it at time $t_1$. Then its polar angle

$\theta$ is: $\theta = \frac{2\pi(t_1 - t_0)}{T}$.

Our localization algorithm is based on this method, its basic idea is: There are two anchors like TA set in the network. And the locations of the two anchors can be arbitrary. Every sensor node calculates a polar angle in the polar systems centered on one of the anchors and a polar angle in the polar systems centered on the other anchors using the method above. The polar axis is on the line that through both the two anchors. Once a sensor node has calculated a polar angle, it can generate an equation of straight line passing through the corresponding anchor and itself by the polar angle and the coordinate of the anchor. Then it can generate another equation of straight line by the polar angle and the other anchor's coordinate. If the sensor node is not collinear with the two anchors, it can generate two different equations. Then it estimates its position at the intersection point of the two straight lines. If the sensor node is collinear with the two anchors, it can use an auxiliary method to get the ratio of $d_1$ and $d_2$ where $d_1$ is the distance between itself and one anchor and $d_2$ is the distance between itself and the other anchor. Then the sensor node can calculate $d_1$ and $d_2$ respectively. As the sensor node has known that it is collinear with the two anchors, it can estimate its position.

The anchors in our algorithm are equipped with directional antenna and can rotate with a constant angular speed. We assume that the each anchor is not energy constrained so that its transmission range is sufficient for the beacon signals to be received by all sensor nodes in the network. The locations of the anchors can be arbitrary.

Each anchor can be a collocated sink which is set on the fixed position near the area where the sensor nodes are deployment. It also can be a special node which has all the ability above and are deployment with all the sensor nodes in the same time.

Every sensor node determines the time that it receives the rotation beacon by estimating the time at which the received signal strength reaches a maximum. In order to reduce the estimated error, the rotation speed of the anchor is much smaller than the transmission speed of the signal.

## 2 Localization algorithm

First of all, the two anchors build a new coordinate system as follows.

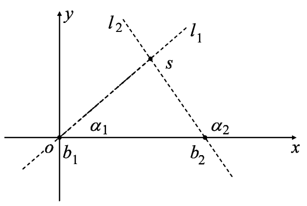



**Fig. 1 Intersection of two straight lines**

We set that the global coordinate of the two anchors ($b_1$ and $b_2$) are $(x_1^a, y_1^a)$, $(x_2^a, y_2^a)$ respectively. The two anchors communicate with each other to determine which one of them to be the origin of the new coordinate system. Without losing generality, we assume $b_1$ as the origin of the new coordinate system in this paper. Then we set the coordinate of $b_2$ in the new coordinate system as $(\sqrt{(x_1^a - x_2^a)^2 + (y_1^a - y_2^a)^2}, 0)$, and set the straight line passing through $b_1$ and $b_2$ as the $x$-axis of the new system. The new coordinate system is indicated in Fig. 1.

After the new coordinate system has been set, every sensor node locates itself by the following steps.

1) A sensor node $s$ calculates its polar angle in the polar system centered at $b_1$, and the polar axis is the $x$-axis of the new coordinate system. Then it calculates its polar angle in the polar system centered at $b_2$, and the polar axis is also the $x$-axis of the new coordinate system. We set the two polar angles are $\alpha_i$ corresponding to $b_i$ respectively ($i = 1, 2$), and the coordinate of $s$ in the new coordinate system is $(x, y)$;

2) There are two different cases for the angle $\alpha_i$:

case 1 $\alpha_i \neq 0$ and $\alpha_i \neq \pi$ ($i = 1, 2$) (the sensor node $s$ is not collinear with the two anchors), Then $s$ can generate a equation of straight line passing through anchor $b_i$ and itself as $l_i: x - x_i = k_i(y - y_i)$, and

$$k_i = \begin{cases} \cot\alpha_i & 0 < \alpha_i < \pi \\ \cot(\alpha_i - \pi) & \pi < \alpha_i < 2\pi \end{cases}.$$

As $s$ is not collinear with the two anchors, it can generate two different equations as $\begin{cases} l_1: x - x_1 = k_1(y - y_1) \\ l_2: x - x_2 = k_2(y - y_2) \end{cases}$. Then $s$ can calculate the coordinate of the intersection point of the two straight lines and estimates its position at that point, as shown in Fig. 1. Then the coordinate of $s$ is $(x, y) = \left(\frac{k_1 x_2}{k_1 - k_2}, \frac{x_2}{k_1 - k_2}\right)$, where $x_2 = \sqrt{(x_1^a - x_2^a)^2 + (y_1^a - y_2^a)^2}$.

case 2　$\alpha_i = 0$ or $\alpha_i = \pi$ $(i = 1, 2)$ (the sensor node $s$ is collinear with the two anchors), then $s$ can locate itself by the following method.

After every sensor node has got its two polar angles, all the sensor nodes that is collinear with the two anchors do time synchronization with the two anchors, and then the two anchors $b_1$ and $b_2$ transmit signals along the $x$-axis of the new coordinate system to the positive and negative direction respectively, as shown in Fig. 2. The signal contains the send time. When a sensor node receives the signal from $b_i$, it can calculate the transmission time $t_i$ of the signal from $b_i$ to itself according to the send time and the receive time.

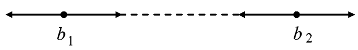



**Fig. 2　Two anchors $b_1$ and $b_2$ transmit signals along the $x$-axis**

Then there are three probably results:

(a) $\alpha_1 = 0, \alpha_2 = \pi$, as shown in Fig. 3 (a), then the coordinate of $s$ in the new coordinate system is $\left(\frac{t_1}{t_1 + t_2} x_2, 0\right)$;

(b) $\alpha_1 = \pi, \alpha_2 = \pi$, as shown in Fig. 3 (b), then the coordinate of $s$ in the new coordinate system is $\left(-\frac{t_2 - t_1}{t_1} x_2, 0\right)$;

(c) $\alpha_1 = 0, \alpha_2 = 0$, as shown in Fig. 3 (c), then the coordinate of $s$ in the new coordinate system is $\left(\frac{t_1 - t_2}{t_2} x_2, 0\right)$.

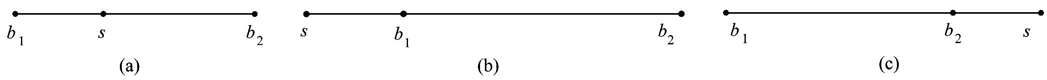



**Fig. 3　The three situations when the sensor node $s$ is collinear with the two anchors**

3) When a sensor node has calculated its coordinate $(x, y)$ in the new coordinate system, it can calculate its global coordinate $(x_a, y_a)$ by coordinate transformation. $(x_a, y_a) = (R\cos\alpha + x_1^a, R\sin\alpha + y_2^a)$, in which $R = \sqrt{x^2 + y^2}$, $\alpha = \arctan\frac{y}{x} + \arctan\frac{y_2^a - y_1^a}{x_2^a - x_1^a}$.

# 3　Analysis and performance evaluation

The localization algorithm we proposed has some advantages which most existing localization algorithms don't have:

1) None sensor node require additional hardware except the two anchors.

2) Every sensor node only needs to receive signals and do some simple calculation, thus the energy consumption is small.

3) It is robust because the result is irrelevant to some characters of the network, including the topology of the network, the node density of the network, and the connectivity of the network.

4) It is a distributed localization algorithm. Every node only needs the information from the anchors to calculate its position and the result is irrelevant to the information of other sensor nodes.

5) The algorithm is easy. Every node only needs to do time synchronization and some simple calculation.

The resulting precision of our algorithm is mainly relevant to the accuracy of estimating the polar angle. The main reason to cause error is the "Non-zero beam width of the directional beam". A directional beam from a wireless antenna has a finite beam width, no matter how small, which will make it difficult for a sensor node to estimate the exact time at which the center of the directional beam passes through it. Though the transmit power strength reaches a maximum at the center of the directional beam, the variety of environmental interferences or the multi-path propagation of the signal will make the receive power of the sensor node is not the strongest at the exact time at which the center of the directional beam passes through the sensor node. As illustrated in Fig. 4.

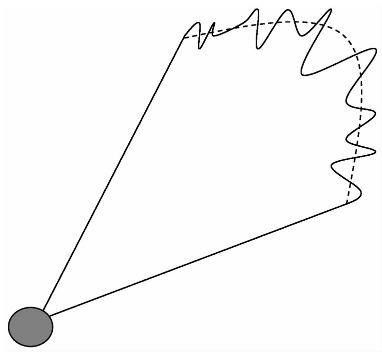


**Fig. 4 Waveform of a directional beam**

In Fig. 4, the dotted line represent the ideal waveform, the solid line represent the waveform in the real environment.

A sensor may perform several determinations of a polar angle and use the averages to improve the accuracy. The improvement of the hardware of directional antenna can reduce the beam width which also can improve the precision.

**Simulation result**

In this section we present some results obtained from computer simulations demonstrating the performance of the proposed algorithm. The simulation environment is: There are 1000 sensor nodes randomly deploying in a $100 \times 100$ square area. The coordinate of the lower-left corner is (0, 0). As the position of the anchors can not affect the result of this algorithm, for operating easily, we use two anchors, one of which is set at the point (0, 0) and the other is set at the point (100, 0). As illustrated in Fig. 5. The main evaluation criterion is the mean error of all the sensors. Then we define the mean error as follow:

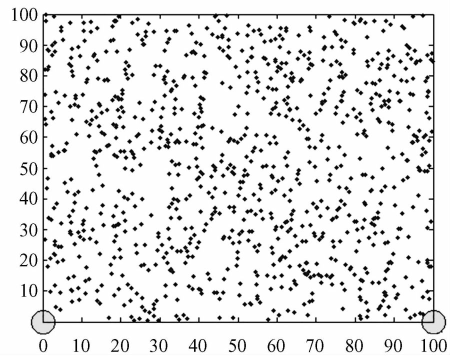



**Fig. 5 Simulation environment**

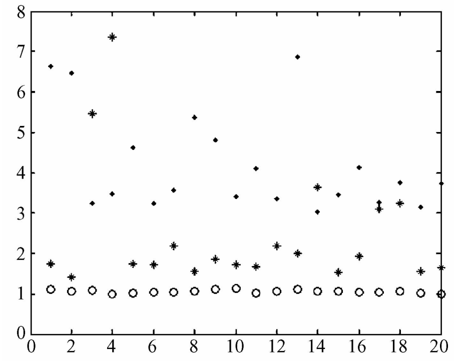
$$\text{meanerror} = \frac{\sum_{i=1}^{N} \sqrt{(x_i - x_i^r)^2 + (y_i - y_i^r)^2}}{N},$$

in which $(x_i, y_i)$ is the estimated position of $\text{sensor}_i$ and $(x_i^r, y_i^r)$ is the real position of $\text{sensor}_i$, and $N$ is the total number of sensor nodes. Our simulation platform is Matlab.

For performing several determinations of a polar angle and using the averages may improve the precision, we first analyze how the precision is affected by the number of times of one sensor node determining the polar angle. We do experiments of the polar angle determination times equal to 1, 5, and 10 respectively, and each do 20 independent experiments. The experiment result is shown in Table 1 and Fig. 6.

**Table 1　Relationship between the number of times of one sensor node determining the polar angle and the result precision**

| number of times of polar angle determination | mean error |
|---|---|
| 1 | 4.1810 |
| 5 | 2.4643 |
| 10 | 1.0536 |



$x$-axis: the 20 times experiments; $y$-axis: the mean error of each experiment.

**Fig. 6　Relationship between the number of times of one sensor node determining polar angle and precision**

In Fig. 6, the $x$-axis represents the 20 experiments and the $y$-axis represents the mean error of each experiment. The mark ' · ', ' * ' and ' o ' represents the number of polar angle determination times equal to 1, 5, and 10 respectively. From Table 1 and Fig. 6 we can see that as the number of times of polar angle determination increasing, the localization precision is improved and the result is more stable. So if the number of times of the polar angle determination is enough, our localization algorithm can guarantee the result precision.

Figure 7 illustrates the localization result of a single sensor node. We do 20 experiments for the node. The mark ' o ' represents its real position, and the mark ' · ' and ' * ' represents its estimated position by our algorithm and the number of times of polar angle determination is 1 and 10 respectively. From Fig. 7, we also can see that, the localization precision is improved as the number of times of polar angle determination increasing.

Then we analyze the relationship between the beam width of the directional antenna and the precision. We do experiments of the beam width equal to 5°, 10°, and 20° respectively, and each also do 20 experiments. The experiment result is shown in Table 2 and Fig. 8. In each experiment, the number of times of polar angle determination is 1.

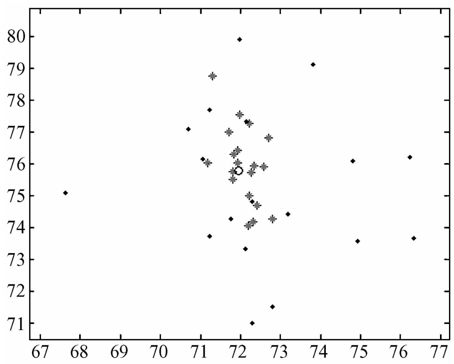
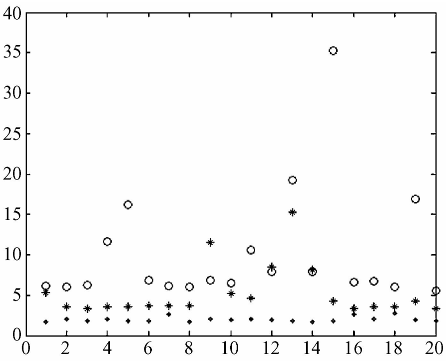


**Fig. 7　Localization result of a single sensor node**



$x$-axis: the 20 times experiments; $y$-axis: the mean error of each experiment.

**Fig. 8　Relationship between the beam width of the directional antenna and the precision**

In Fig. 8, the $x$-axis represents the 20 experiments and the $y$-axis represents the mean error of each experiment. The mark ' · ', ' * ' and ' o ' represents the beam width equal to 5°, 10°, and 20°

respectively. From Table 2 and Fig. 8 we can see that as the beam width decreasing, the localization precision is improved. So if the beam width directional antenna is narrow enough, our localization algorithm can get an enough good result.

**Table 2 Relationship between the beam width of directional antenna and the result precision**

| beam width of the directional antenna | mean error |
|---|---|
| 5° | 1.9873 |
| 10° | 5.3018 |
| 20° | 10.0589 |

## 4 Conclusion

We propose a localization algorithm without distance information in this paper. Compared to most existing localization algorithms, our algorithm has some advantages elaborated in part 3. For these advantages, this localization algorithm is more suitable for the wireless sensor networks than most existing localization methods in many applications. What need to be studied in the future are the special anchors used in this algorithm and how to improve the precision of this algorithm.

### References

[1] 孙利民,李建中,陈渝,等. 无线传感器网络[M]. 北京:清华大学出版社,2005.

[2] Lim A, Qing Y, Casey K, et al. Real-time target tracking with CPA algorithm in wireless sensor networks[C]//Sensor, Mesh and Ad Hoc Communications and Networks, 2008. SECON apos 5th Annual IEEE Communications Society Conference. 2008, 16-20:305-313.

[3] He T, Huang C, Blum B M, et al. Range-free localization scheme for large scale sensor networks[C]//Proc 9th Annual Int Conf on Mobile Computing and Networking. San Diego, CA ,2003.

[4] Wong S Y, Lim J G, Rao S V. Density-aware hop-count localization (DHL) in wireless sensor networks with variable Density[C]//Wireless Communications and Networking Conference, 2005(3):1848-1853.

[5] Wong S Y, Lim J G, Rao S V, et al. Multihop localization with density and path length awareness in non-uniform wireless sensor networks [C]//Proceedings of the 61st IEEE Vehicular Technology Conference, Stockholm. Sweden, 2005:2551-2555.

[6] Yang S W, Yi J Y, Cha H J. HCRL: a hop-count-ratio based localization in wireless sensor networks[J]. Sensor, Mesh and Ad Hoc Communications and Networks, 2007, 18(21):31-40.

[7] Wang Y, Wang X D, Wang D M, et al. Localization algorithm using expected hop progress in wireless sensor networks[J]. Mobile Adhoc and Sensor Systems, 2006:348-357.

[8] Zhao D L, Men Y H, Zhao L L. A hop-distance algorithm for self-localization of wireless sensor networks[J]. Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing, 2007, 2:108-112.

[9] Girod L, Estrin D. Robust range estimation using acoustic and multimodal sensing[C]//Proc IEEE/RSJ Int'l Conf Intellingent Robots and System (IROS'01). Maui, Hawaii, USA, 2001.

[10] Ananthasubramaniam B, Madhow U. Detection and localization of events in imaging sensor nets[C]//Proc 2005 IEEE International Symposium on Information Theory (ISIT), Adelaide. Australia, 2005.

[11] Nasipuri A, Li K. A directionality based location discovery scheme for wireless sensor networks[C]//Proceedings of ACM WSNA'02. 2002.

[12] Stojmenovic I. Handbook of sensor networks—algorithms and architectures[M]. John Wiley & Sons, 2005.

# 一个无需知道节点间距离的无线传感器网络定位方法

姜志鹏, 高随祥

(中国科学院研究生院数学科学学院, 北京 100049)

**摘 要** 目前无线传感器网络中,大多数定位算法需要节点间的距离信息,这些方法有很多弊端.提出一种不需要节点间距离的定位方法,可以避免上述弊端.该方法只需要 2 个锚节点,且锚节点的位置任意.仿真结果显示该方法有较好的性能.

**关键词** 无线传感器网络,定位技术,旋转信标