Localization for Wireless Sensor Networks with Improved Dv-Hop and Spring-Model Refinement

Chunping Tu

College of Information Engineering, East China Jiao Tong University, chunpingtu@sina.com

Abstract

Wireless sensor network (WSN) is a key enabling technology for ambient intelligence, where localization of sensor nodes is a fundamental and essential issue. Aiming at localization deficiency of DV-Based algorithm which are caused by the shortest path distance substituting for Euclidean distance, combing the advantages of error inhibition of MSO (mass-spring model), in this paper we propose a DV-MSO localization algorithm, which not restrains the localization error of DV-Hop algorithm, but also overcomes the defect of MSO algorithm that it is easy to fall into local optimum. Simulation results demonstrate that the localization error of unknown nodes is reduced greatly by the proposed method.

Keywords: Wireless Sensor Networks, DV-Hop Algorithm, mass-spring model, localization

1. Introduction

In recent decade, great advances in MEMS, the miniaturization of microprocessors and electronic devices have enabled the development to low-cost, low-power, multifunctional sensor nodes. A wireless sensor network is composed of a large number of sensor nodes which have the ability to sense, process and communicate that are densely deployed in a field [1]. People have found its significant potential use in various fields which include target tracking [2], monitoring, controlling. Specific applications of wireless sensor networks include disaster management, environmental monitoring, home automation, vehicle surveillance, etc[3]. In all of these WSN applications, sensor network is composed of a large number of sensors, and it is unlikely that the physical positions of each node has been pre-determined. This is because: Nowadays, the most simple, off-the-shelf, mechanism to determine the absolute location of sensor nodes is to use the global positioning system (GPS) [4], but this is currently a costly solution. It has been a challenging task to design a practical algorithm for node localization given the constraints that are imposed on sensors, including limited power, low cost, etc [5].

A great many of localization algorithms have been proposed for wireless sensor networks. These localization protocols are classified into range-based and range-free algorithms [6]. The range-based localization schemes can be further classified into Time of Arrival (TOA) [7], Time Difference of Arrival (TDOA)[8], Angle of Arrival (AOA)[9], and Received Signal Strength Indicator (RSSI)[10] measurements. The range-based algorithm uses absolute point-to-point distance or angle estimates for calculating the location, which are relatively precise but require additional hardware and their cost is relatively high [11]. On the contrary, the range-free algorithms do not need the distance or angle information to the sensor nodes from the anchor nodes for their localization, they provide more economic and simpler estimates than the range-based ones, but their results are not as precise as those of the range-based[12]. The range-free scheme enables sensors to learn their location information without the aid of range estimates, such as Centroid[13], APIT[6], DV-HOP[14], CPE[15], MCL[16], MDS-MAP[17], and so on, it is suitable for sensor positioning due to its cost-effectiveness. In ranged-free schemes, the sensor nodes without location information (called unknown nodes) gather location information from nodes with known locations (called anchors) and estimate their own location according to the location information of the anchors. However, existing ranged-free schemes are either too costly (cause heavy traffic load), not accurate enough or not scalable.

In the trilateration of wireless sensor networks, to overcome the affection of localization error caused by the selection of anchor nodes, a selective strategy of anchor nodes based on the angle information is proposed in [21]. By the comparative analysis of angle, known nodes can choose three anchor nodes which are the most accurate to execute trilateration. In [22], YuanYuan Li proposed an improved DV-HOP location algorithm based on local estimating and correction in location for
wireless sensor networks. They introduce the neighbor anchor nodes as the main participants to estimate the location of unknown node. The local information can better reflect the location of the unknown nodes. Meanwhile, they dynamically adjust the error according to the anchor nodes distribution of the unknown nodes.

To improve previous works, we propose a DV-MSO localization algorithm for static WSNs. The rest of this paper is organized as follows. Section 2 presents DV-Hop localization algorithm. Section 2 presents Spring-Relaxation Technique and mass-spring model. Section 4 provides localization theory and experiments about DV-MSO localization algorithm. Section 5 presents our conclusions.

2. DV-Hop Localization Algorithm

DV-Hop localization algorithm is proposed by Dragos Niculescu and Badri Nath. Anchor nodes generate packets including their position information and a flag which is initialized as 1 to figure the number of hops away from them. These packets are flooded in WSN. When they are transmitted by the relay nodes, the hop number is increased by 1. In this way, any node can determine the hop number from it to a certain anchor node. Similarly the anchor nodes can compute their hops to other anchors as well. The average distance per hop can be determined by a simple formula and then will be broadcast. When an unknown node receives it, the receiver will estimate its distance to the anchor node. After it obtains three or more estimated values form anchor nodes, its location can be figured out. It mainly consists of three steps [18].

Step 1, Computing the least hops and calculating the average hopsize. Each anchor node broadcasts its information throughout the network. The information package contains the location information of each anchor node. Ultimately, all sensor nodes acquire minimum hop count from anchor nodes. Then we estimate the average hopsize of each anchor node using the location information between anchor nodes and the minimum hop count between each other. We compute the hopsize of each beacon node as follows:

$$\text{HopSize}_i = \frac{\sum_{i \neq j} \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}}{\sum h_{ij}}$$

Where \((x_i, y_i)\) and \((x_j, y_j)\) are the coordinates of anchor node \(i\) and \(j\), \(h_{ij}\) is the hop count between anchor node \(i\) and \(j\). Then, each anchor node broadcasts the information with its hopsize throughout the network using flooding.

Step 2, Calculating the estimated distance among nodes. After the step1 is implemented, each node has the information, including coordinates of all anchor nodes and the least hops to them. Unknown node \(k\) keep the average distance of anchor node \(i\) which is received first, then calculate the distance \(d_{kl}\) using hop count \(h_{kl}\) between unknown node \(k\) and anchor node \(l\) by the following formula.

$$d_{kl} = h_{kl} \times \text{HopS}$$

Step 3, Computing the location. Based on the knowledge of plane geometry, coordinate of the unknown node can be computed after knowing the coordinates of the three anchor nodes and the distance to them. Let \((x_1, y_1), (x_2, y_2), (x_3, y_3)\) are the coordinates of three anchor nodes respectively, and the distance to unknown node D are \(d_1, d_2, d_3\), the coordinate of D is \((x, y)\), then there is the following formula:

$$\begin{align*}
\sqrt{(x-x_1)^2 + (y-y_1)^2} &= d_1 \\
\sqrt{(x-x_2)^2 + (y-y_2)^2} &= d_2 \\
\sqrt{(x-x_3)^2 + (y-y_3)^2} &= d_3 
\end{align*}$$
The coordinate of D is computed by the following formula:

\[
\begin{pmatrix}
    x \\
    y
\end{pmatrix} = \left( \begin{pmatrix}
    2(x_1 - x_3) \\
    2(x_2 - x_3)
\end{pmatrix} \right)^{-1} \begin{pmatrix}
    x_1^2 - x_2^2 + y_1^2 - y_2^2 + d_3^2 - d_2^2 \\
    x_1^2 - x_2^2 + y_1^2 - y_2^2 + d_3^2 - d_2^2
\end{pmatrix}
\]


To explain the concept of mass-spring model for the localization, we consider a simple example that consists of five anchor and two sensor whose location is to be determined. In the concept of mass-spring model, the considered example is equivalent to having two moving particle (i.e., unknown sensor) attaching with five springs. For each spring, while its one end attaches to the particle, its another end is nailed by a pin (i.e., anchor node) at a fixed location. Figure 1 depicts the described example.

Phase 1: sensor nodes are deployed in an area. The Wireless equipment denotes the wireless sensor network nodes and the solid lines denote that two nodes are within the radio range.

Phase 2: the wireless sensor network is represented by the spring system. The anchor nodes are represented by the red sensor nodes. The unknown nodes are represented by the green sensor nodes. The distances between neighbor nodes are represented by the springs.

Phase 3: the anchor nodes have their actual absolute positions, so the anchor nodes are fixed in their absolute positions. The unknown nodes don’t know their positions, so they are drawn to random positions. After this, the springs between these unknown are stretched or compressed, which makes the total forces of the unknown nodes not equal to zero. Therefore, these unknown nodes will move according to step 4.

Phase 4: these unknown nodes go back to the stable locations where they are before. The locations can be obtained using step 4. These locations are what we need to localized.

![Figure 1. The simple example of spring model for wireless sensor network](image-url)
node pair $i, j$ where the measuring distance $r_{ij}$ and estimated distance $d_{ij}$. Each particle has attribute $m$ and $v$, where $m$ denotes the mass of the particle, $v$ denotes the velocity. Each spring has two $k$, $l_{ij}$ attributes, where $k$ denotes the spring constant, $l_{ij}$ denotes the original length of the spring.

Step 2: Computing. From Hooke’s law, the magnitude of the force $F$ from each spring is

$$f_{ij} = v_{ij}k_{ij}(d_{ij} - l_{ij}) = v_{ij}k_{ij}(d_{ij} - a_{ij})$$

Step 3: The net force on sensor node $i$, defined as $F_i$ is the vector sum of all forces and $E_i$ is the node $p_i$ energy.

$$F_i = \sum_{i,j} f_{ij}$$

$$E_i = \sum \|f_{ij}\|^2 \approx \sum (d_{ij} - r_{ij})^2$$

Step 4: Updates the locations of sensors in iterations. In each iteration, the algorithm moves node $i$ a small distance in the direction of $F_i$ and then re-computes all the applied forces. Considering a linear relationship between the net force and the displacement, the location of node $i$ is then updated as

$$X[i + 1] = X[i] + \delta F_i$$

Where $\delta$ be the step size of location adjustment, which is related to the speed of the convergence. Based on experience, the value is in the interval of $[1 / m, 1 / (2m)]$.

Step 5: Assume that the threshold value force $F_i$ is $T_{force}$. When $\|F_i\| > T_{force}$, it need to repeated calculation $F_i$ continuously and update node location until $\|F_i\| \leq T_{force}$.

4. DV-MSO Localization Algorithm

4.1. Basic knowledge

In this paper, based on DV-hop algorithm and the advantages of MSO, we proposed an improved localization algorithm. Algorithm procedure is as follows:

Step 1: distance between adjacent nodes is obtained by actual measuring (i.e. Euclidean distance $r_{ij}$), obtain distance matrix $D$ by using the shortest path method.

Step 2: calculating the estimated distance between unknown nodes and each beacon node. Ultimately, unknown nodes in the network compute their locations $a_i$ by trilateration.

Step 3: A particle-spring system is proposed to model a wireless sensor network, where $p_i = a_i$ is a initial coordinate and $l_{ij} = r_{ij}$ is the length of the spring.

Step 4: Repeat theory 4 and 5 of MSO until the total force exerted by neighbors is smaller than a threshold $T_{force}$.

4.2. Simulation Results

In order to verify the effectiveness and the availability of DV-MSO algorithm for improving localization accuracy, we use Matlab to compare and analyze DV-MSO algorithm with the original DV-Hop localization algorithm.

Let $(x_n', y_n')$ be the estimated coordinates of node $n$, and $(x_n, y_n)$ be its real coordinate. We express the localization error in percents, relative to the communication radio $R$. Namely,
In this paper, the average localization error of the sensor network which is composed with N unknown nodes is defined as follows:

\[ \Delta = \frac{1}{N} \sum_{n=1}^{N} \Delta d_n \]

In the initial simulation experiments, we assumed that the network is a C-shaped square area with the size of 1000m×1000m. Due to the C shaped obstacle, signal is blocked so that edge node can not communicate with the adjacent nodes. 240 sensors are randomly scattered with a uniform distribution with in the C-shaped square area. Sensor nodes have the same maximum communication radius R to be 200m. There are 48 anchor nodes (denoted by the red pentagram), which leads to an average neighbor anchor nodes number of network: 6.0417. Fig.2 shows the neighbor relationship graph. The black dot represents the true locations of the nodes. Fig.3 shows DV-hop localization error map. The black dot represent the estimation locations of the nodes, straight lines represent the estimation error, the longer the line represents the error the greater. Simulation result of localization error is 35.73%. Fig.4 shows localization error of DV-MSO is 10.77%. This example demonstrates that DV-MSO works well on C-shaped square area and is better than DV-hop.

In Figure.5, We discuss the influence of localization error based on the traditional DV-HOP algorithm and the DV-MSO algorithm on the ratio of anchor nodes. Each case runs 100 times in the simulation, and simulation results are the average values. The results show that the average location error of the traditional DV-HOP algorithm and our scheme are decreased with the increasing in the number of anchor nodes. the traditional DV-HOP localization errors reach over 30%, which make the algorithm fail. DV-MSO algorithm’s maximum error is less 30%, in line with localization accuracy of the standard algorithm. Under the same conditions, the average location error of our scheme is significantly less than the traditional DV-HOP algorithm.
5. Conclusions

In this paper, we proposed a DV-MSO algorithm based on DV-Based algorithm and the advantages of error inhibition of MSO (mass-spring model). And we point out that between non-neighbor nodes can not communicate directly in DV-hop algorithm. When using the shortest communication path instead of using the Euclidean distance to calculate the distance matrix, there will have some errors and lead to the limitation of node positioning ineffective. On the other hand, the MSO algorithm lacks of restraint in the initialization, which make node coordinate easy trapping local optimum. But MSO has a robust inhibitory effect on the error. Thus, DV-MSO localization algorithm is proposed. Simulation results show that this method can greatly improve the localization accuracy of the unknown nodes.

6. References

Localization for Wireless Sensor Networks with Improved Dv-Hop and Spring-Model Refinement
Chunping Tu


