

# 一种改进的无线自主传感器网络定位算法<sup>①</sup>

殷 琪, 徐 巍

(湖北工业大学 机械工程学院, 武汉 430068)

**摘 要:** 节点定位技术是无线自主传感器网络中的关键技术之一。为了提高定位精度, 提出一种基于几何斜率的无线传感器网络(WSN)定位算法。网络区域中的节点分为锚节点和未知节点, 利用几何学斜率的方法选取合适的锚节点, 能够更精确地确定未知节点的位置。在三边测量法上运用最小平方误差方法求解, 能够提高算法的精度。在新算法的基础上建立 Matlab 仿真。仿真结果表明改进的 DV-HOP 算法, 在相同的锚节点数量的情况下, 节点定位精度有明显的提高。

**关键词:** 无线自主传感器网络; 定位节点; 锚节点; 定位误差

## Improved Location Algorithm for Wireless Sensor Networks

YIN Qi, XU Wei

(School of Mechanical and Engineering, Hubei University of Technology, Wuhan 430068, China)

**Abstract:** The location information of the sensor node is one of the key technologies in Wireless sensor network(WSN). To improve the location accuracy, a geometric localization algorithm is proposed for wireless sensor networks. Nodes in the network area is divided into anchor nodes and unknown nodes, using the method of slope geometry, select the appropriate anchor node, to determine the location of unknown nodes more accurately. This paper proposed the least square error method based on the trilateration to improve the algorithm accuracy. A Matlab simulation model based on the new algorithm is established. The results show that the proposed algorithm improves node location accuracy a lot with same amount of anchor nodes under same radio range condition.

**Key words:** wireless sensor networks; node localization; anchor nodes; localization error

## 1 引言

在传感器网络中, 位置信息对传感器网络监测活动至关重要, 事件发生的位置或获取信息的节点位置是传感器节点监测消息中所包含的重要信息, 没有位置信息的监测消息往往毫无意义。因此, 确定事件发生位置或获取信息的节点位置是无线自主传感器网络最基本的功能之一, 对无线自主传感器网络应用的有效性起着关键作用<sup>[1]</sup>。

在 WSN 中, 传感器节点是随机播撒的, 只有少数带有 GPS 定位装置或者其它方式的节点可以确定自身位置, 其它节点只能根据已知节点位置按照某种定位机

制估算出自身位置。其中位置已知节点为锚节点(anchor node), 其它节点为未知节点(unknown node)。WSN 定位根据定位过程是否测量实际节点的距离, 分为基于测距(range-based)和测距无关(range-free)两种算法<sup>[4]</sup>。近年, 为了得到更高的定位精度, 提出了很多解决办法, 在定位精度和算法复杂度上寻求平衡。基于测距的定位算法虽然定位精度较高, 但对节点硬件设施要求很高, 同时在定位过程中要产生能耗。与测距无关的定位算法中, 以 DV-Hop 定位算法为例, 其算法简单、覆盖度高和可行性好, 对节点硬件要求低, 受环境因素影响小, 应用最为广泛, 是目前大家普遍关注的

① 基金项目: 武汉市晨光计划(200750731305); 湖北省教育厅项目(B200714003)

收稿时间: 2011-05-17; 收到修改稿时间: 2011-06-15

定位机制。但是它的主要缺点是定位精度不是很高,因此需要对算法作出改进<sup>[4]</sup>。

## 2 DV-Hop定位原理

距离向量-跳段 DV-Hop 定位机制和传统网络中的距离向量路由机制十分类似。在距离路由机制中,未知节点首先计算与信标节点的最小跳数,然后估算平均每跳的距离,利用最小跳数乘以平均每跳距离,得到未知节点与信标节点之间的估计距离,再利用三边测量法或极大似然估计法计算未知节点的坐标<sup>[8-10]</sup>。

这个定位算法需要在网络中部署一些位置信息已知的节点,称这些节点为锚节点。网络中其他位置信息未知的节点就是普通节点,简称为节点。DV-Hop 定位算法由 3 个步骤组成:信息广播、距离计算、非锚节点定位计算<sup>[10]</sup>。

### 2.1 信息广播

每个锚节点将其位置信息传递给其所有的邻居节点。广播的信息格式为  $\{ID_i, x_i, y_i, Hopsiz_e_i\}$  位置坐标  $(x_i, y_i)$  以及跳数  $Hopsiz_e_i$  信息。初始化  $Hopsiz_e_i$  为 0。接收到此数据的每个节点将  $Hopsiz_e_i + 1$  并记录到一张表中,然后继续向新的邻居节点广播,可计算出锚节点的每跳平均距离。

### 2.2 距离计算

在第一阶段信息广播结束后,将每个锚节点的每跳平均距离进行平均,得到整个网络的每跳平均距离,用  $aver$  (average of hop) 表示。然后每个普通节点可计算自己到每个锚节点间的距离  $d = Hopsiz_e_i \times aver$ , 存入表中。

其中,  $(x_i, y_i), (x_j, y_j)$  是锚节点  $i, j$  的坐标,  $h_j$  是锚节点  $i$  与  $j$  ( $i \neq j$ ) 之间的跳断数。

$$Hopsiz_e_i = \frac{\sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}}{\sum_{j \neq i} h_j} \quad (1)$$

$$aver = R/h \quad (2)$$

### 2.3 定位计算

采用三边或极大似然估计法进行计算。我在这里采用的是三边测量法。

假设有三个锚节点,坐标为  $(x_i, y_i), i=1,2,3$ ,未知节点的坐标为  $(x, y)$ ,与三个锚节点的距离为  $r_i, i=1,2,3$ 。根据欧氏距离计算公式,下列方程组成立:

$$(x_i - x)^2 + (y_i - y)^2 = r_i^2, (i=1,2,3) \quad (3)$$

为求解这个方程组,把它转换成关于  $x$  和  $y$  的线性方程组。因此需要消去二次项,从前面两个方程减去第三个方程,得到两个新方程为:

$$(x_1 - x)^2 - (x_3 - x)^2 + (y_1 - y)^2 - (y_3 - y)^2 = r_1^2 - r_3^2 \quad (4)$$

$$(x_2 - x)^2 - (x_3 - x)^2 + (y_2 - y)^2 - (y_3 - y)^2 = r_2^2 - r_3^2 \quad (5)$$

整理后得到:

$$2(x_3 - x_1)x + 2(y_3 - y_1)y = (r_1^2 - r_3^2) - (x_1^2 - x_3^2) - (y_1^2 - y_3^2) \quad (6)$$

$$2(x_3 - x_2)x + 2(y_3 - y_2)y = (r_2^2 - r_3^2) - (x_2^2 - x_3^2) - (y_2^2 - y_3^2) \quad (7)$$

写成线性矩阵方程得到:

$$2 \begin{bmatrix} x_3 - x_1 & y_3 - y_1 \\ x_3 - x_2 & y_3 - y_2 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} (r_1^2 - r_3^2) - (x_1^2 - x_3^2) - (y_1^2 - y_3^2) \\ (r_2^2 - r_3^2) - (x_2^2 - x_3^2) - (y_2^2 - y_3^2) \end{bmatrix} \quad (8)$$

求解以上方程即可得到未知节点的坐标为  $(x, y)$ 。

DV-Hop 算法在网络平均连通度为 10,锚节点比例为 10%的各向同性网络中定位精度约为 33%。这种方法的优点是对节点的硬件要求低,实现简单,但也有一定的局限性,它在测量的过程中利用跳段距离代替直线距离,存在一定的误差,而且锚节点密度,仅在各向同性的密集网络中,校正值才能合理地估算平均每跳距离。

## 3 改进算法

DV-Hop 定位算法算法简单、覆盖度高和可行性好,应用最为广泛。但是它的主要缺点是定位精度不是很高,因此对算法作出改进。

针对 DV-Hop 定位算法的缺陷,我们提出了一种改进的方案即利用几何学中斜率法选取合适的锚节点组合,其定位精度高,并且算法复杂度低。首先使用几何学中的斜率方法,即用斜率来判断锚节点间的分布关系。其次,当锚节点个数  $n$  等于或大于 3 时,分别采用不同方法求解位置节点的位置。具体步骤如下:

① 计算未知节点与锚节点的间的跳数。舍弃锚节点接收到邻近锚节点跳数中过大或过小的值,然后求其平均值。

② 判断锚节点的个数。当  $n < 3$  时,无法执行三边或极大似然估计法定位;当  $n = 3$  时,判断 3 点是否接近共线。如图 1 所示,首先计算直线  $L_1$  的斜率为  $k_1$ ,再计算直线  $L_2$  的斜率  $k_2$ ,当  $k_2 - k_1$  值越接近 0 时,三点越接近共线,这时利用三点计算未知节点的位置就不准确了,就需要重新选择锚节点;当  $n > 3$  时,首先选取最先接收到锚节点的信息作为参考节点,从其

余锚节点中选取 2 个锚节点进行定位。当 1 个未知节点周围多余 3 个锚节点时，把离未知节点最近的锚节点作为参考节点，分别从中选出 2 个锚节点进行最小二乘定位，进行多次求出未知节点的位置，并把求出的位置进行求平均。当 A, B, C, D 四个锚节点同时向未知节点 M 发送信息包时，M 将最先接收到 B 的消息，把 B 作为参考节点，分别从其它锚节点中选出 2 个锚节点，构成 3 个锚节点，然后判断 3 个锚节点是否共线。

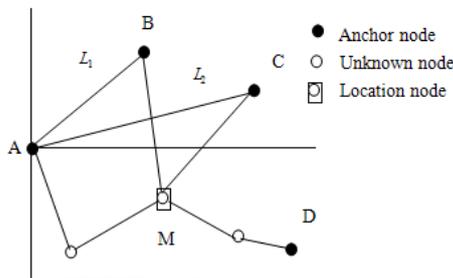


图 1 锚节点与未知节点关系分布图

③ 根据改进的三边测量法来计算未知节点 A 的位置。假设有  $i$  个锚节点，坐标为  $(x_i, y_i), (i=1,2,3, \dots)$ ，未知节点的坐标为  $(x, y)$ ，与  $i$  个锚节点的距离为  $r_i (i=1,2,3, \dots)$ 。根据欧氏距离计算公式，下列方程组成立：

$$(x - x_i)^2 + (y - y_i)^2 = r_i^2 \quad (i=1,2,3, \dots, n) \quad (9)$$

为求解这个方程组，把它转换成关于  $x$  和  $y$  的线性方程组。因此需要消去二次项，从前面两个方程减去第三个方程，得到两个新方程为：

$$(x_1 - x)^2 - (x_3 - x)^2 + (y_1 - y)^2 - (y_3 - y)^2 = r_1^2 - r_3^2 \quad (4)$$

$$(x_2 - x)^2 - (x_3 - x)^2 + (y_2 - y)^2 - (y_3 - y)^2 = r_2^2 - r_3^2 \quad (5)$$

.....  
写成线性矩阵方程：

$$2 \begin{bmatrix} x_1 - x_3 & y_1 - y_3 \\ \dots & \dots \\ x_n - x_3 & y_n - y_3 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} (r_1^2 - r_3^2) - (x_1^2 - x_3^2) - (y_1^2 - y_3^2) \\ \dots \\ (r_n^2 - r_3^2) - (x_n^2 - x_3^2) - (y_n^2 - y_3^2) \end{bmatrix} \quad (10)$$

对于上面的超定方程，可以采用最小平方误差方法求解<sup>[8]</sup>，即选择一个  $(x, y)$  使得  $\|AX - b\|_2$  最小。这里  $2A$  是上式左边的矩阵(是  $(n-1)*2$  的矩阵)， $X=(x, y)$  是未知节点的坐标值， $b$  是式(2-10)右边的矢量(是一个  $n-1$  行的矢量)。 $\|\cdot\|_2$  是 2 范数，最小化 2 范数表示未知节

点的坐标以最小平均误差的方式满足所有  $n-1$  个锚节点对它的约束。

为了对这个最小化问题求解，先看上面表达式的 2 范数的平方，注意到对矢量  $v$ ，有  $\|v\|_2^2 = v^T v$ 。这样

$$\|AX - b\|_2^2 = (AX - b)^T (AX - b) = X^T A^T AX - 2X^T A^T b + b^T b \quad (11)$$

使上式最小化等价于最小化均方误差。把上式当作是  $x$  的函数，令梯度等于 0，得到

$$2A^T AX - 2A^T b = 0 \Leftrightarrow A^T AX = A^T b \quad (12)$$

上式被称为线性最小平方问题的正则方程 (Normal equation)。在一定的假设条件下，方程有唯一的解(矩阵  $A$  是满秩的)，有几种方法来求解该方程，比如 Cholesky 分解或 QR 分解(把  $A$  分解为  $A=QR$ ， $Q$  是正交的， $R$  是上三角矩阵)，这两种分解的差别在于计算未知数值的稳定性。

可见，对于锚节点  $n$  大于等于 3 存在距离测量误差的情况下，该方法能够得到具有最小均方误差的坐标估计。

#### 4 仿真实验

为了验证所提出的改进算法的可行性，利用 Matlab7.0 对 DV-Hop 算法及本文提出的改进后的算法进行了仿真实验，并针对相关实验结果进行分析。在实验仿真的过程中，在  $200\text{m} \times 200\text{m}$  的矩形区域布置 300 个网络节点，其中锚节点为 50 个，通信距离  $R$  为 40，网络连通度为 33.3。产生的随机拓扑图 2。

##### 1) 节点分布图

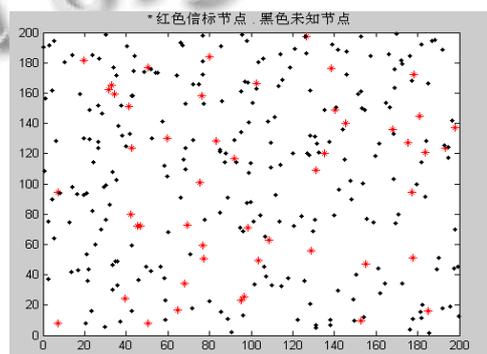


图 2 网络节点分布图

如图所示，红色星节点为锚节点，如图可以看到很多锚节点排列成直线分布，这样用原始的 DV-Hop 算法定位，误差会很大。

##### 2) 节点数量与平均定位误差的关系

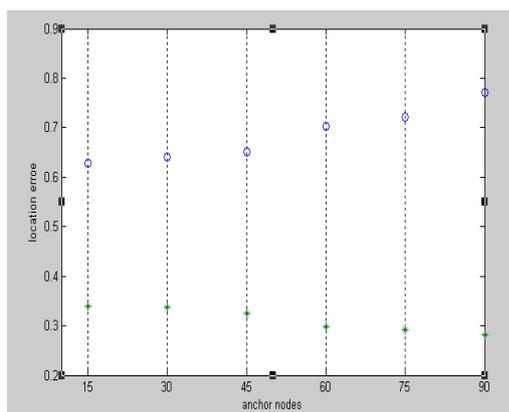


图3 节点数量与平均定位误差的关系

如上图所示在锚节点数从 15,30, ...,90 的变化过程中, 可以看到这两种算法的定位误差与锚节点数的关系图。蓝色圆圈代表的是 DV-HOP 定位算法的误差图, 绿色星形代表的是改进后的 DV-HOP 定位误差图。从图中, 不难可以发现, 改进后的算法的定位误差明显低于 DV-HOP 算法, 可达到 28%, 并且会随着锚节点的增加误差值会逐渐缩小。

## 5 结论

通过对 DV-Hop 算法分析, 总结 DV-Hop 算法的特点, 在此基础上提出了改进算法。改进算法是基于一种几何斜率的定位方法, 来选取合适的锚节点, 从而提高定位精度。经过仿真实验表明在锚节点数量一定的

前提下, 其定位误差明显好于 DV-Hop 算法, 增强了系统的鲁棒性。因此, 改进后的算法是有效和可行的。

## 参考文献

- 1 任丰原, 黄海宁, 林闯. 无线传感器网络. 软件学报, 2003, 14(7):1282-1291.
- 2 邱岩. 无线传感器网络节点定位技术研究. 计算机科学, 2008, 35(5):47-50.
- 3 段渭军, 王建刚, 王福豹. 无线传感器网络节点定位系统与算法的研究和发展. 信息与控制, 2006, 35(2):239-245.
- 4 王福豹, 史龙, 任丰原. 无线传感器网络中的自身定位系统和算法. 软件学报, 2005, 16(5):857-868.
- 5 He T, Huang CD, Blum BM, Stankovic JA, Abdelzaher T. Range-free localization schemes in large scale sensor networks. Proc. of the 9th Annual Int'l Conf. on Mobile computing and Networking. San Diego: ACM Press, 2003.81-95.
- 6 B WHL, H CJ. Global positioning system. 1997.
- 7 Bahl P, Padmanabhan VN. RADAR: An inbuilding RF-based user location and tracking system. Proc. of IEEE Infocom 2000, Tel-Aviv, Israel. 2000.2:775-784.
- 8 李善亮, 黄刘生, 吴俊敏. 基于连通性的传感器节点定位算法研究. 计算机工程, 2008, 34(7):115-117.
- 9 孙利民, 等. 无线传感器网络. 北京: 清华大学出版社, 2005.
- 10 杨旻. 传感器网络节点定位技术研究. 杭州: 浙江大学, 2006.

(上接第 16 页)

化对虚拟化存储系统进行了扩展和丰富, 并在石油物探的深水海洋地震资料处理解释系统的存储子系统中进行了应用, 取得了较好的效果。

## 参考文献

- 1 杨光年, 郭荣亮, 张国政. 存储技术的发展及整合与虚拟化应用. 计算机与数字工程, 2008, 36(3):140-144.
- 2 谢长生, 金伟. SAN 网络存储虚拟化实现方式研究与设计.

计算机应用研究, 2004, 4:191-193.

- 3 谭生龙. 存储虚拟化技术的研究. 微计算机应用, 2010, 21(1): 33-38.
- 4 Hitachi Universal Volume Manager User's Guide. Hitachi Data Systems. 2010.
- 5 StorNext File System. ADIC Educational Services. 2006.
- 6 Hitachi Dynamic Link Manager Software User's Guide for Linux. Hitachi Data Systems. 2010.