

• 网络与通信 •

文章编号:1000-3428(2010)23-0092-03

文献标识码:A

中图分类号:TP393.03

## 基于运动向量的无线传感器网络节点定位

李东岳<sup>1,2</sup>, 王英龙<sup>1</sup>, 刘颖慧<sup>1,3</sup>, 郭强<sup>1</sup>, 魏诺<sup>1</sup>

(1. 山东省计算中心, 济南 250000; 2. 山东轻工业学院, 济南 250000; 3. 山东科技大学, 山东 青岛 266510)

**摘要:**介绍无线传感器的常用分类,提出一种基于无需测距的定位方法,即基于运动向量的无线传感器网络移动节点定位。通过信号强度得到未知节点的大体区域,利用节点运动中向量的变化减小定位范围,根据向量的移动提高估计定位的精确度。对该算法进行仿真和总结,结果表明,与凸规划法相比,该算法有更高的定位精度。

**关键词:**无线传感器网络; 移动节点; 定位

## Node Location of Wireless Sensor Networks Based on Motion Vector

LI Dong-yue<sup>1,2</sup>, WANG Ying-long<sup>1</sup>, LIU Ying-hui<sup>1,3</sup>, GOU Qiang<sup>1</sup>, WEI Nuo<sup>1</sup>

(1. Shandong Computer Science Center, Jinan 250000, China; 2. Shandong Institute of Light Industry, Jinan 250000, China;  
3. Shandong University of Science and Technology, Qingdao 266510, China)

**【Abstract】**The paper introduces the common classification of wireless sensor, and proposes a range-free algorithm: motion node location algorithm of Wireless Sensor Networks(WSM) based on Motion Vector(MV). It uses received signal strength getting unknown nodes region, uses node movements orientation vector to reduce the scope of the region, and estimates location based on MV to improve accuracy. The algorithm is simulated and concluded. With the convex programming method for comparison, simulation results show that it has higher positioning accuracy.

**【Key words】**Wireless Sensor Networks(WSN); mobile node; localization

### 1 概述

随着网络通信技术、嵌入式技术、无线传输技术、分布式信息技术的发展,作为获取信息的重要途径——无线传感器网络得到飞速发展。无线传感器网络被应用到各个领域,如军事国防、生物医疗、环境监测、工农业等,且具有鲜明的跨学科特点,并已成为各发达国家研究的热点<sup>[1-2]</sup>。

节点定位是无线传感器网络的一个关键问题,不携带位置信息的节点通常是无效的<sup>[3]</sup>。无线传感器网络的自身定位一般分为2类:基于测距技术的定位算法(range-based)和无需测距的定位算法(range-free)。前者通过测量节点间点到点的距离或角度信息来计算节点位置,其中常用的算法有RSSI、TDOA、TOA等。基于测距技术的定位算法会受到反射、多径传播、非视距等问题的影响,从而产生较大的误差。无需测距的定位算法的网络模型都是由锚节点和未知节点组成的异构网络,无需任何基础设施,其中常用的算法有质心定位算法、凸规划定位算法<sup>[4]</sup>、DV-Hop定位算法等。本文提出了一种无须测距的新的定位算法:基于运动向量的无线传感器网络移动节点定位,简称MV。

### 2 MV 算法

将4个已知锚节点 $N_1$ 、 $N_2$ 、 $N_3$ 、 $N_4$ 置成正方形,如图1所示,未知节点为 $U$ ,在其中移动。其中, $N_1$ 与 $N_2$ 相距 $D_1$ , $N_1$ 的通信半径为 $D_2$ ;  $D_1 \geq D_2 \geq (N_2, N_4)/2$ (若 $D_2$ 小于这个区域,则无法保证 $U$ 在正方形中心时能接收到所有的信号;若 $D_2$ 大于等于这个区域,则精确度下降);通信时间相隔 $t$ 。

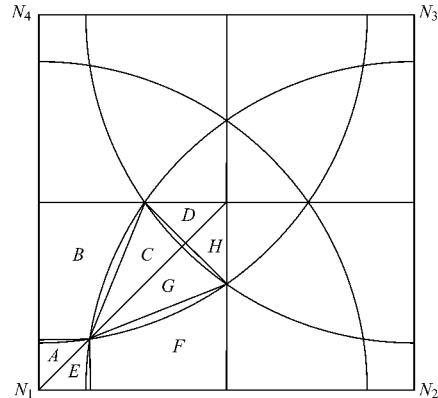


图1 根据信号强度的区域划分

#### 2.1 MV 算法原理

(1) 根据信号强度将 $N_1 N_2 N_3 N_4$ 所组成的正方形划分成若干个区域,然后确定未知节点 $U$ 所在的区域组成的多边形 $P$ 。

(2) 点 $U$ 在正方形中移动时, $U$ 会判断它所能接收到的

**基金项目:**国家自然科学基金资助项目(60802030);山东省中青年科学基金奖励基金资助项目(2007BSC01002);山东省科技攻关计划基金资助项目(2007GG2QT01007)

**作者简介:**李东岳(1982—),男,硕士研究生,主研方向:无线传感器网络;王英龙,研究员,博士生导师;刘颖慧,硕士研究生;郭强,副研究员,博士;魏诺,硕士

**收稿日期:**2010-02-27   **E-mail:**lidy@keylab.net

信号强弱,若  $U$  接收到的信号强度越强,则  $U$  记录其锚节点的位置;接收到的信号强度越弱,设  $U$  从上一个位置到现在的位置的向量为  $\mathbf{S}_1\mathbf{S}_2=(x_a, y_a)$ <sup>[5-6]</sup>, 则  $U$  记录上一个位置的锚节点位置加上向量  $\mathbf{S}_1\mathbf{S}_2$ , 即  $\mathbf{N}\mathbf{N}'=\mathbf{S}_1\mathbf{S}_2=(x_a, y_a)$ , 可以得到  $\mathbf{N}'=(x_a', y_a')=(x_a+x_a, y_a+y_a)$ 。

(3)在得到 3 个以上的锚节点位置时,会组成一个三角形  $T$  或一个四边形  $Q$ 。最后由  $P$  和  $T$  或  $Q$  相交得到一个多边形  $Z$ , 则  $Z$  的中心为  $U$  所在位置;若  $T$  和  $Q$  没有形成,或  $P$  和  $T$  或  $Q$  不相交,则多边形  $P$  的中心为  $U$  所在的位置。

(4)如果节点继续运动,点  $U$  运动经时间  $t$  运动到下一位,设时间  $t$  内,  $U$  的向量改变为  $\mathbf{T}_1\mathbf{T}_2=(x_t, y_t)$ ,  $U$  原来的估算位置为  $T(x_e, y_e)$ , 则现在的估算位置为  $T'(x_e+x_t, y_e+y_t)$ 。再判断  $T'$  是否在多边形  $Z$  或多边形  $P$  中,若在其中,则估算出的  $T'$  的位置不改变;若  $T'$  不在多边形  $Z$  或多边形  $P$  中,则重新计算  $T'$  位置,  $T'$  为多边形  $Z$  或多边形  $P$  的中心位置。

## 2.2 MV 算法步骤

以图 1 中  $N_1$  所在的 1/4 区域为例,算法步骤如下:

(1)根据信号强度确定未知节点  $U$  所在的一个多边形  $P$ 。

1)如果  $U$  与  $N_1, N_2, N_3, N_4$  之间信号强度相等,则  $U$  在中心位置。

2)只接收 1 个信号时,则  $U$  一定在与其相邻的  $N_2, N_3, N_4$  通信区域外,  $U$  在 AE 区域,  $P$  为 AE 区域所组成的四边形。

3)只接收 2 个信号时,假设接收信号的是  $N_1, N_2$ 。则  $U$  在 F 区域,多边形  $P$  为 F 区域的边所组成的梯形。假设接收信号的是  $N_1, N_4$ 。则  $U$  在 B 区域,多边形  $P$  为 B 区域的边所组成的梯形。

4)只能接收 3 个信号时,假设接收信号的是  $N_1, N_2, N_3$ 。如图 1 不可能出现这种情况,因为在  $N_4$  以外的区域,只有  $EF$ ,而在  $EF$  都不在  $N_3$  的范围内。假设接收信号的是  $N_1, N_2, N_4$ 。在 CG 区域,若  $N_2 > N_4$ , 则  $U$  在 G 区域,  $P$  为 G 区域所组成的三角形;若  $N_2 < N_4$ , 则  $U$  在 C 区域,  $P$  为 C 区域所组成的三角形;若  $N_2 = N_4$ , 则  $U$  在 CG 的中心,  $P$  为 CG 区域所组成的三角形。

5)接收 4 个信号时,假设接收信号的是  $N_1, N_2, N_3, N_4$ , 则  $U$  在 DH 区域。若  $N_2 > N_4$ , 则  $U$  在 H 区域,  $P$  为 H 区域所组成的三角形;若  $N_2 < N_4$ , 则  $U$  在 D 区域,  $P$  为 D 区域所组成的三角形;若  $N_2 = N_4$ , 则  $U$  在 DH 的中心,  $P$  为 DH 区域所组成的三角形。

(2)根据  $U$  在移动过程中所接收到的信号,确定一个三角形  $T$  或一个四边形  $Q$ 。

假设  $U$  的位置坐标为  $M_1(x, y)$ , 如图 2 所示,若  $U$  收到的  $N_3$  的信号强度越大,即  $U$  与  $N_3$  越近,  $U$  由  $M_1$  运动到  $M_2(x+x_a, y+y_a)$ ,  $U$  在  $M_2$  收到的信号强度大于在  $U$  在  $M_1$  收到的信号强度,则  $U$  仍记录  $N_1$  相对于  $U$  的位置为  $(100, 100)$ , 当  $U$  运动到  $M_3(x+x_a+x_b, y+y_a+y_b)$ ,  $U$  收到信号强度与以前所接收到的信号强度做相比较,信号强度弱了,则  $U$  记录  $N_1$  的位置跟随  $U$  的向量移动,  $U$  从  $M_2$  到  $M_3$  向量的改变为  $(x_b, y_b)$ , 则记  $N_3'$  的位置为  $(x_b+100, y_b+100)$ 。当  $U$  记录的信号来自 3 个节点时,生成一个三角形  $T$ , 即  $\Delta N_1 N_2' N_3'$ , 如图 2 所示,当  $U$  记录 3 个或 4 个节点的信号

时,则生一个三角形  $T$  或四边形  $Q$ 。当  $U$  越接近 4 个锚节点时,所形成的三角形  $T$  或四边形  $Q$  会越小。若  $U$  在  $N_2, N_3, N_4$  区域里,依此类推。

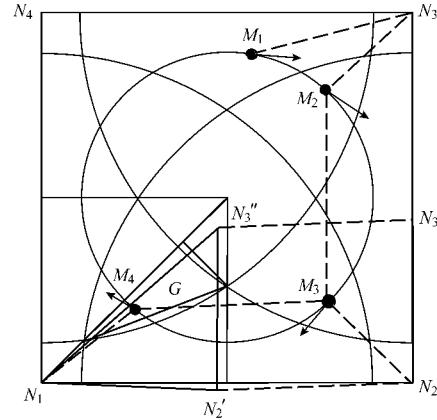


图 2 MV 算法示意

(3)若  $U$  没有接收到 3 个信号,则  $U$  的位置定位为  $P$  的中心;若  $U$  接收到 3 个或 4 个信号,则  $U$  的位置定位为:以  $P$  与  $T$  或  $Q$  相交后所形成的多边形  $Z$  的中心。如图 2 所示,  $U$  的位置为  $G$  与  $\Delta N_1 N_2' N_3'$  相交后多边形的中心。

(4)在第 1 次估算出未知节点的位置后,未知节点移动多少向量,则估算出的位置也加上相应的向量,得到新估算出的位置。用  $P$  或  $Z$  来判断新估算出的位置是否在其中,若不在,则前面的 3 个步骤重新定位;若在  $P$  或  $Z$  中,则不需要重新定位。

## 3 仿真分析

以 Matlab 平台进行仿真实验,并分析算法的性能,最后在相同的环境下与凸规划算法进行比较,质心算法需要的锚节点数较多,DV-hop 算法利用距离矢量跌幅的方法来测距,并与 MV 算法进行比较,所以,本文选用与凸规划算法进行比较。设锚节点之间距离为  $D_1=100$  m, 锚节点的通信半径为  $D_2=80$  m。如图 1 所示,节点随机分布在  $100$  m  $\times$   $100$  m 区域内,所有未知节点为匀速随机直线运动  $1$  m/s。节点的误差  $error = \sqrt{(x_e - x_r)^2 + (y_e - y_r)^2}$ , 节点的平均误差

$$mean\_error = \frac{1}{n} \sum_{i=1}^n \sqrt{(x_e - x_r)^2 + (y_e - y_r)^2},$$

其中,  $x_e, y_e$  表示节点的估计定位;  $x_r, y_r$  代表节点的真正位置;  $n$  表示未知节点个数。单个节点在运动中的平均定位误差如图 3 所示,MV 算法和凸规划算法所产生的误差分布比较明显,用 MV 算法所产生的误差基本上都在用凸规划算法所产生的误差之下。大部分分布在  $7$  m  $\sim$   $13$  m 之间。

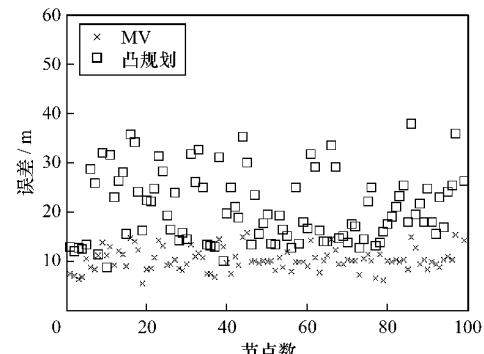


图 3 单个节点的平均定位误差

随着节点数的增加,节点的平均定位误差如图 4 所示,刚开始时定位误差相差不大,随着节点数的增加,误差逐渐拉大,MV 算法所产生的误差为 9.61 m,凸规划算法所产生的误差为 21.38 m。MV 算法所产生的误差是凸规划算法的一半。

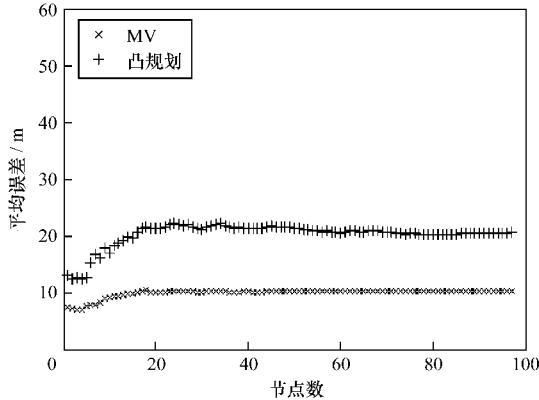


图 4 随着节点数的增加定位误差的变化

为了充分说明本算法的优越性,单个未知节点在  $100 \times 100$  内以螺旋曲线的方式由内向外运动。节点在运动中的误差分布如图 5 所示,节点在刚开始运动时,MV 算法与凸规划算法所得到的节点定位误差相差不大,甚至凸规划算法占优,因为节点在刚运动时是在中心区域,凸规划算法占有一定的优越性,而 MV 算法会因为信号强度不同,在不同的区域内运动会产生较大的误差。随着节点的运动,MV 算法突显出优越性,按照算法的步骤(4),节点的估计定位随着节点的移动而移动,当节点的估计定位与节点的实际位置相近或重合时,节点的估计定位会越来越准确,而且在以后的运动中,节点的定位也会是准确的。在节点运动过程中,节点的估计定位因为靠近节点的实际位置,所以不需重新定位,从而可以保持定位精度。当节点跳出多边形 Z 范围时,就需要重新定位,如图 5 所示,图中有产生 5 次较大的误差。

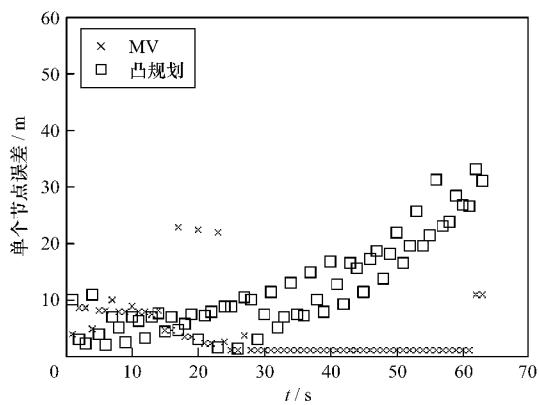


图 5 节点误差的变化

以螺旋曲线的方式,随着节点的运动单个节点的平均定位误差如图 6 所示。刚开始时,MV 算法和凸规划算法定位所产生的误差相差不大,结合图 5 所示,误差最小在 2 s~3 s

左右分布,在  $t=25$  s 时,MV 算法定位所产生的误差减小,在  $t>40$  s 时,MV 算法逐渐与凸规划算法定位所产生的误差拉开距离。MV 算法与凸规划算法的平均误差分别为 5 m 和 12 m。

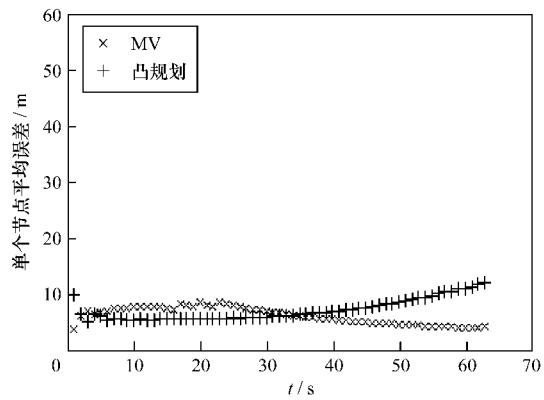


图 6 节点平均误差的变化

#### 4 结束语

本文提出了一种适用于无须测距的新的定位算法,且算法具有较高的定位精度。如果想提高定位精度或是不计成本,可以在四方形区域内抛洒几个锚节点,当未知节点经过锚节点时,可以有效而快速地提高定位精度。如果是大范围地布节点的话,可以考虑以多个四边形做成矩阵的方式来定位。

#### 参考文献

- [1] Wang Wei, Srinivasan V, Wang Bang, et al. Coverage for Target Localization in Wireless Sensor Networks [J]. IEEE Transactions on Wireless Communications, 2008, 7(2): 667-676.
- [2] Zhao Jijun, Li Hua, Sun Xiang. Research on Signal Random Attenuation Coefficient Based on RSSI in WSN to Localizing Technology[C]//Proc. of the 5th International Conference on Wireless Communications, Networking and Mobile Computing. Beijing, China: [s. n.], 2009: 1-4.
- [3] Younis M F, Ghuman K, Eltoweissy M. Location-aware Combinatorial Key Management Scheme for Clustered Sensor Networks[J]. IEEE Transactions on Parallel and Distributed Systems, 2006, 17(8): 865-882.
- [4] 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[J].信息与控制,2006,35(2): 239-345.
- [5] Ou C H. Range-free Node Localization for Mobile Wireless Sensor Networks[C]//Proc. of the 3rd International Symposium on Wireless Pervasive Computing. Santorini, Greece: [s. n.], 2008.
- [6] Van Der P B A, Marshak S. Earth Structure: An Introduction to Structural Geology and Tectonics[M]. New York, USA: [s. n.], 2003.

编辑:任吉慧