

A Distributed Node Location Algorithm of Wireless Sensor Network*

TIAN Zeng-shan*, LUO Lei, HE Wei, HAN Ling-jun

(Institute of Wireless Location & Space Measurement, Chongqing University of Posts and Telecommunications, Chongqing 400065, China)

Abstract: The node location algorithm of wireless sensor network based on the receiver signal strength (RSSI) get a wide applications, which no additional hardware supporting. However, the environment impact the wireless signal transmission which making the positioning algorithm has large errors. In order to reduce the error impact of environment, this paper presents a probability positioning methods. According to the sending signal strength of beacon and receive signal strength to calculate one jump estimated distance of unknown nodes, the location of unknown nodes can be found by using the maximum probability density function. The simulation shows that this method can get high positioning accuracy and coverage at low density of the beacon and communication radius.

Key words: wireless sensor networks; received signal strength indicator; probability positioning; positioning accuracy; coverage

EEACC:6150P

一种分布式无线传感器网络节点定位算法*

田增山*, 罗磊, 何维, 韩令军

(重庆邮电大学无线定位与空间测量研究所, 重庆 400065)

摘要: 基于接收信号强度(RSSI)的无线传感器网络节点定位算法由于无需额外测距硬件的支持而受到广泛的应用。但无线信号传输受到环境的影响,使得基于RSSI的定位算法存在较大的定位误差。针对上述不足提出了一种基于概率的无线传感器网络节点定位方法。首先根据信标节点发送信号强度与未知节点接收信号强度差来计算未知节点到其一跳范围内所有信标节点的估计距离,然后根据这些估计距离,通过计算未知节点在某个位置概率密度函数的最大值来确定未知节点自身的位置坐标。通过仿真表明,该方法在较低的信标节点密度和节点通信半径条件下,具有较高的节点定位精度和定位覆盖率。

关键词: 无线传感器网络;接收信号强度;概率定位;定位精度;覆盖率

中图分类号:TN953

文献标识码:A

文章编号:1004-1699(2009)03-0387-04

无线传感器网络是由部署在检测区域内大量的廉价的微型传感器节点组成,通过无线通信方式形成的一个多跳自组织网络系统,其目的是协作地感知、采集和处理网络覆盖区内感知对象的信息,并发送给观测者。在传感器网络中,位置信息对传感器网络的监测活动非常重要,事件发生的位置或获取信息的节点位置是传感器网络节点监测信息中所包含的重要信息^[1]。对于大多数应用而言,不知道位置信息的监测消息往往毫无意

义。传感器节点必须在明确自身位置的情况下才能实现对外部目标的定位和跟踪。因此确定事件发生的位置或获取消息的节点位置是无线传感器网络最基本的功能之一,对传感器网络应用的有效性起着关键性的作用^[2]。

现有定位技术中,一般是利用无线传感器网络中少量已知位置的节点来获取其它未知节点的位置信息。已知位置的节点称为信标节点,未知位置的节点称为未知节点。现有的无线传感器网络节点自

基金项目:教育部重点项目资助(207097)

收稿日期:2008-11-10 修改日期:2008-12-01

身定位方法主要有两种:即基于测距的定位方法和无需测距的定位方法。无需测距的定位方法不需要测量角度和距离信息,仅根据网络的连通性和信标节点的信息来定位。这种方法的定位精度和收敛速度在一定程度上依赖于网络平均每跳距离的估计精度,而且当网络的拓扑结构不规则时,算法的性能急剧下降^[3]。基于测距的定位算法通过测量节点间点到点的距离或角度信息计算出未知节点的位置,这种方法虽然定位精度较高,但对硬件的要求苛刻。文献[4]利用测量三个锚节点到待测未知节点的信号强度差来计算估计距离,然后通过 Chan 算法来计算待测未知节点的位置坐标;文献[5]首先将待定位节点可能存在的区域划分为若干栅格,再根据无线信号在空间传输的概率模型,计算出待定位节点处于各个栅格的置信度,最后以置信度较高的栅格的形心为基准点,计算出待测定位节点的坐标位置。文献[4]和[5]都利用了接收信号强度(RSSI)来实现定位,减少了终端的复杂度,但文献[4]需要三个锚节点的信息才能实现定位,定位覆盖率不高,而且 Chan 算法不能克服 RSSI 测量值受环境噪声的影响,定位精度不高;文献[5]在 RSSI 测距的基础上利用无线信号在空间传输的概率模型,一定程度上提高了待测节点的定位精度,但是定位的精度取决于栅格划分的大小,而栅格划分的越小,计算越复杂,因而待测节点的定位精度和算法的复杂度相互冲突,而且定位的结果限定在已设定的栅格上,部分节点将被定在同一个位置上。

本文在基于 RSSI 测距优势的前提下,利用计算未知节点在某一位置上具有最大概率密度来实现未知节点的定位。该方法在未知节点一跳范围内具有两个信标节点就可以实现其快速定位,定位的精度随着一跳范围内信标节点的增多而快速提高,且计算量没有明显的增大;同时不会出现多个未知节点定位于一点的情况;此外该方法采用分布式计算,由未知节点自行计算自身的位置坐标,一定程度上减少了节点的能耗,增加了整个无线传感器网络的生存时间,因而可广泛应用于无线传感器网络节点自身定位中。

1 基于 RSSI 的距离估计

由于无线传播信号受到阴影衰落,经过障碍物的反射、衍射等影响,接收的信号强度指示(RSSI)结果受到了较为严重的噪声污染。大量实验的统计分析结果表明,信号传输的估计距离 w_i 与接收信号强度差 $L(d_i)$ 之间存在如下的函数关系:

$$w_i = a \frac{L(d) - X_i - L(d_i)}{b \cdot N} \times d$$

其中, w_i 表示未知节点到一跳范围内第 i 个信标节点的估计距离; X_i 是标准差为 σ 的随机变量; $L(d)$ 为信标节点 i 的参考点传输路径损耗; d 为信标节点 i 的参考点距离; $L(d_i)$ 为未知节点接收到信标节点 i 的信号强度差; a, b 为常数,取值为 10 ± 2 ; N 为路径损耗指数,通常介于 2 和 3 之间。

2 概率定位算法设计

假定测量的误差是由多种因素引起的,它们的综合效果形成了实际的误差,用统计的观点看,可以认为这样的误差分布呈正态分布。这里我们假定测量误差的标准差为 σ_0 。

2.1 建立未知节点存在某一位置的概率密度函数

未知节点利用获得的其一跳范围内所有信标节点的估计距离,分别计算未知节点可能落在某个位置的概率密度函数 $p_i(d)$, $i = 1 \cdots n$, n 为未知节点一跳范围内的信标节点总个数,过程如下^[6-7]:

对于某个距离测量值 w_i ,我们可以用 d 的概率密度函数来描述,即

$$p_i(d) = \frac{1}{\sqrt{2\pi}\sigma_0} \exp\left[-\frac{(d-w_i)^2}{2\sigma_0^2}\right] \quad (1)$$

设未知节点的坐标为 (x, y) , 该未知节点在一跳范围内接收到的第 i 个信标节点的坐标为 (x_i, y_i) , 其中 $i = 1 \cdots n$, 则对于距离测量 d , 它所给出的坐标点与测量值之间的关系为:

$$d = \sqrt{(x-x_i)^2 + (y-y_i)^2} \quad (2)$$

对在 (x_0, y_0) 附近的区域,用泰勒级数展开,只取一阶,将复杂的非线性问题线性化近似,即:

$$d = d(x_0, y_0) + \frac{\partial d}{\partial x}(x-x_0) + \frac{\partial d}{\partial y}(y-y_0) \quad (3)$$

于是式(1)成为

$$\begin{cases} p_i(d) = \frac{1}{\sqrt{2\pi}\sigma_0} \exp[F_i(x, y)] \\ F_i(x, y) = -\frac{\left[\frac{\partial d}{\partial x}(x-x_0) + \frac{\partial d}{\partial y}(y-y_0) + d(x_0, y_0) - w_i\right]^2}{2\sigma_0^2} \end{cases} \quad (4)$$

其中, (x_0, y_0) 的取值为未知节点在一跳范围内可以通信的信标节点的质心,即 $x_0 = \frac{1}{n} \sum_{i=1}^n x_i$,

$y_0 = \frac{1}{n} \sum_{i=1}^n y_i$, $F_i(x, y)$ 是一个二次多项式,其标准形式为:

$$F_i(x,y) = a_{1i}x^2 + a_{2i}y^2 + a_{3i}xy + a_{4i}x + a_{5i}y + a_{6i} \quad (5)$$

其中:

$$a_{1i} = -\frac{1}{2\sigma_0^2}(\frac{x_0 - x_i}{d_0})^2$$

$$a_{2i} = -\frac{1}{2\sigma_0^2}(\frac{y_0 - y_i}{d_0})^2$$

$$a_{3i} = -\frac{1}{2\sigma_0^2}(\frac{x_0 - x_i}{d_0})(\frac{y_0 - y_i}{d_0})$$

$$a_{4i} = -\frac{1}{\sigma_0^2}(\frac{x_0 - x_i}{d_0})v_0$$

$$a_{5i} = -\frac{1}{\sigma_0^2}(\frac{y_0 - y_i}{d_0})v_0$$

$$a_{6i} = -\frac{v_0^2}{2\sigma_0^2}$$

$$d_0 = \sqrt{(x_0 - x_i)^2 + (y_0 - y_i)^2}$$

$$v_0 = d_0 - w_0 - \frac{x_0^2 + y_0^2 - x_ix_0 - y_iy_0}{d_0}$$

假定未知节点对一跳范围内所有信标节点的距离估计都是相互独立的, 则所有对于该未知节点其它的测量值 $w_j (i \neq j)$, 即未知节点在一跳范围内接收到的第 j 个信标节点的距离测量值, 可以得到其概率密度函数 $p_j(d)$, 一共可以得到 n 个这样的概率密度函数^[8]。总的概率密度函数由单个测量的概率密度函数相乘产生。所有的单个测量的概率密度函数都是指数函数, 总的概率密度函数也将是一个指数函数, 它的指数是各单个概率密度函数的指数的和^[9]。也就是说, 总概率密度函数的指数为:

$$F(x,y,z) = \sum_{i=1}^n F_i(x,y) \quad (6)$$

各 F_i 是 x, y 的二次多项式, 那么, 它们的和 F 也是 x, y 的二次多项式, 可以用下式来表示。即

$$F(x,y) = a_1x^2 + a_2y^2 + a_3xy + a_4x + a_5y + a_6 \quad (7)$$

其中, $a_1 = \sum_{i=1}^n a_{1i}; a_2 = \sum_{i=1}^n a_{2i}; a_3 = \sum_{i=1}^n a_{3i};$

$$a_4 = \sum_{i=1}^n a_{4i}; a_5 = \sum_{i=1}^n a_{5i}$$

2.2 未知节点求解自身的位置坐标

从物理意义上讲, 求解未知节点坐标的过程就是求解概率密度函数最大值点的过程。以 e 为底的指数函数是单调递增的, 求它的最大值完全可以归结为求它的指数的最大值。

对式(7)中 x, y 分别求导, 并令导数为零, 就可以得到求解定位点的方程式为:

$$\begin{cases} 2a_1x + a_3y = -a_4 \\ a_3x + 2a_2y = -a_5 \end{cases}$$

即

$$\begin{pmatrix} 2a_1 & a_3 \\ a_3 & 2a_2 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} -a_4 \\ -a_5 \end{pmatrix}$$

只要 $\begin{vmatrix} 2a_1 & a_3 \\ a_3 & 2a_2 \end{vmatrix} \neq 0$, 方程就有唯一的解。也就

是说在 (x_0, y_0) 附近的区域, 概率密度函数有一个确定的峰值。如果求解的起始点在实际的目标位置附近, 这个峰值就是所要求的目标位置。如果求解的起始点离实际的目标位置有一定距离, 可以采用迭代的方法, 即从起点 (x_0, y_0) 开始, 先计算一次, 得到一个更好的起点, 再做定位计算。

可以得到未知节点的坐标解为:

$$\begin{cases} x = \frac{2a_2a_4 - a_3a_5}{a_3^2 - 4a_1a_2} \\ y = \frac{2a_1a_5 - a_3a_4}{a_3^2 - 4a_1a_2} \end{cases}$$

3 仿真分析

本文以基于距离的定位算法中有代表性的最小二乘定位算法为代表来对比分析基于概率定位算法的性能。信标节点和普通节点的坐标随机生成。当两节点间的实际距离 d 小于节点通信半径 R 时, 默认两节点已经实现通信, 接收信号强度取与节点距离相关的正态分布随机变量。为了使仿真的结果趋于真实, 取 100 次仿真的均值作为最终的仿真结果。仿真参数的设置见表 1。

表 1 仿真参数设置

参数	值	参数	值
网络覆盖面积	500 m×500 m	信标节点个数	20~60
节点总数	120	节点通信半径 R	40 m~100 m

仿真结果如下各图所示。从图 1 和图 2 可以看出节点的通信半径对无线传感器节点的定位性能有较大的影响, 从整体上来看基于概率的定位算法其定位性能要明显高于最小二乘算法; 当节点的通信半径大于 60 m 时, 基于概率的定位, 其定位误差逐渐趋于平稳, 大约为 30 m。这是由于随着节点通信半径的加大, 未知节点收到的信标节点个数增大, 消除了概率定位的随机性和偶然性, 从而提高了定位的精度。从图 3 和图 4 可以看出信标节点的个数也对无线传感器节点的定位性能有较大影响, 但概率定位无论在定位误差还是定位覆盖率上都具有较大的优势。特别是随着信标节点个数的增多, 基于概率定位算法的定位误差急剧下降; 而且即使在较少信标节点的情况下, 基于概率的定位仍然有较高的定位覆盖率。

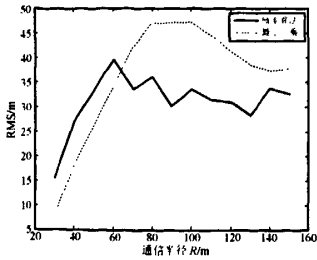


图1 节点通信半径对定位误差的影响(40个信标节点)

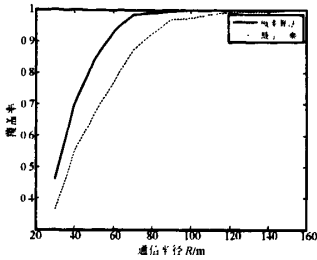


图2 节点通信半径对定位覆盖率的影响(40个信标节点)

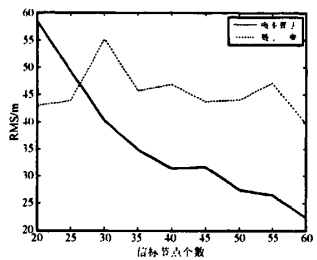


图3 信标节点个数对定位误差的影响(节点通信半径 $R=90$ m)

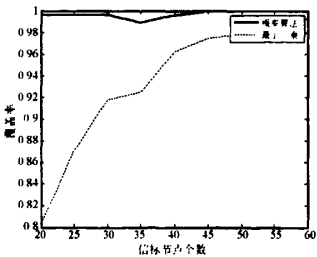


图4 信标节点个数对定位覆盖率的影响(节点通信半径 $R=90$ m)

4 结论

节点的定位技术是无线传感器网络应用的重要支撑,是网络拓扑、覆盖控制、路由选择等上层技术的重要基础。本文在基于接收信号强度的基础上,采用统计学的方法在一定程度上减弱了 RSSI 测量

值噪声污染对定位误差的影响,提高了定位精度和定位覆盖率。通过仿真我们可以看出信标节点的数量仍然是影响节点定位精度的主要因素,而且在仿真的过程中,我们发觉不同的信标节点分布对未知节点的定位精度也有一定程度的影响,这也是我们今后研究的重点。

参考文献:

- [1] 史龙. 无线传感器网络自身定位算法研究[D]. 硕士论文. 西北工业大学, 2005.
- [2] 尚志军, 曾鹏, 于海斌. 无线传感器网络节点定位问题[J]. 计算机科学, 2004, 31(1): 35-37.
- [3] 于宁, 万江文. 无线传感器网络抽样定位和求精算法[J]. 计算机工程, 2008, 34(3): 15-17.
- [4] 一种基于测距的无线传感器网络节点定位方法[P]. 申请(专利)号: 200710164581. 2, 公开号: CN 101004448.
- [5] 蒋文涛, 李连, 孙利民. 无线传感器网络中一种基于栅格划分的概率定位算法[J]. 计算机应用, 2008, 28(4): 841-855.
- [6] 用于定位技术的概率模型[P]. 申请(专利)号: 03815408. 0, 公开号: CN 1666111A.
- [7] 胡来招. 无源定位[M]. 国防工业出版社, 2004: 31-70.
- [8] Doherty L, Ghaoui L E, and Pister K S J. Convex Position Estimation in Wireless Sensor Networks[C]// IEEE INFOCOM, Anchorage, AK, Apr. 2001.
- [9] NLCULESCU D, NATH B. DV Based Positioning in Ad Hoc Networks[J]. Journal of Telecommunication Systems, 2003, 22(1/4): 267-280.



田增山(1968-), 男, 河南固始人, 博士, 重庆邮电大学教授, 主要研究方向为移动通信、个人通信、GPS及蜂窝网定位系统及其应用技术研究, tianzs@cqupt.edu.cn



罗磊(1983-), 男, 河南邓州人, 重庆邮电大学在读硕士研究生, 主要研究方向为移动通信与无线定位, luolei88011743@163.com