

# 基于跳数的无线传感器网络定位求精算法

郭永红<sup>1</sup>, 万江文<sup>2</sup>, 于 宁<sup>1</sup>, 冯仁剑<sup>2</sup>

(1. 北京邮电大学自动化学院, 北京 100876; 2. 北京航空航天大学仪器科学与光电工程学院, 北京 100083)

**摘 要:** 节点本身的定位误差和节点间距离测量(或计算)误差的积累限制了无线传感器网络节点定位循环求精的功效。为有效提高定位精度, 在继承加权最小二乘法优点的基础上, 提出基于跳数的求精和与其相对应的节点相对置信度方法。增加节点定位计算的冗余度, 可有效减小误差积累。将该方法应用于 Robust Positioning 定位算法中, 构成一种新的 RP-Hop 定位算法。仿真实验结果表明, 节点的定位精度可提高约 10%。

**关键词:** 无线传感器网络; 节点定位; 循环求精; 加权最小二乘法

## Hop-based Refinement Algorithm for Localization in Wireless Sensor Networks

GUO Yong-hong<sup>1</sup>, WAN Jiang-wen<sup>2</sup>, YU Ning<sup>1</sup>, FENG Ren-jian<sup>2</sup>

(1. Automation School, Beijing University of Posts and Telecommunications, Beijing 100876;

2. School of Instrument Science & Opto-electronics Engineering, Beijing University of Aeronautics and Astronautics, Beijing 100083)

**【Abstract】** Accumulation of node localization error and distance measurement(or calculation) error between nodes restrict the benefit of iterative refinement of node localization in Wireless Sensor Networks(WSN). In order to improve the precision of node localization effectively, this paper proposes a hop-refinement and relative confidence based on Weight Least Square(WLS) algorithm. Hop-refinement increases redundancy for node location. Relative confidence declines accumulation of error. Application of hop-refinement in robust positioning algorithm brings forward RP-Hop algorithm. Simulation results show that localization precision increases by 10% compared with Robust Positioning algorithm.

**【Key words】** Wireless Sensor Network(WSN); node localization; iterative refinement; Weight Least Squares(WLS)

### 1 概述

由于受到成本、功耗、扩展性等问题的限制, 在大中型无线传感器网络(Wireless Sensor Network, WSN)中, 往往只有少数节点配置 GPS 接收器, 也不可能人工部署每个节点, 因此, 必须采用一定的机制与算法解决节点的定位问题。

根据定位过程中是否需要测量节点间的距离, 定位机制可分为基于距离(range-based)和与距离无关(range-free)的 2 类定位算法。为减小无线传感器网络的能量开销, 在大规模随机布撒的网络中多采用与距离无关的定位算法来计算节点的位置。循环求精是提高定位精度的有效手段, 是距离无关定位算法的重要后续步骤<sup>[1]</sup>, 并且, 也适用于基于距离的定位算法<sup>[2]</sup>。循环求精也可在一定程度上解决由于网络中信标节点较少而带来的节点定位精度低等问题。但节点本身的定位误差和节点间距离测量(或计算)误差的积累, 限制了无线传感器网络节点定位循环求精的功效<sup>[3]</sup>, 解决这一问题就能进一步提高定位精度, 对无线传感器网络定位技术具有重要意义, 因此, 本文提出基于跳数的求精算法。

### 2 定位算法分析

无线传感器网络中现有的定位算法通常包括距离(或角度)测量、坐标计算以及可选的循环求精 3 个阶段。

(1)距离测量: 主要有基于达到时间(TOA)、基于到达时间差(TDOA)、基于接收信号强度指示(RSSI)等。

(2)坐标计算: 主要有 3 边测量法和极大似然法估计法。

(3)循环求精: 根据未知节点的定位方式, 循环求精可以

分为 2 种: 1)利用相关定位算法得到未知节点初始定位结果, 将其邻居节点作为参考节点, 重新计算未知节点的位置, 并将符合限定条件的结果作为未知节点新的位置估计, 进入下一次循环过程, 直到满足循环停止的条件<sup>[1, 4]</sup>; 2)部分未知节点确定自身位置后, 将其升级为信标节点, 进入下一次循环, 直到所有满足定位条件的未知节点确定位置后定位结束<sup>[5]</sup>。

在循环求精过程中, 将已定位的未知节点引入到其他未知节点的定位计算, 在一定程度上可解决信标节点数量不足的问题, 但由于节点本身的定位和节点间距离的测量都存在误差, 导致误差在循环过程中积累。

本文提出的基于跳数的求精算法(hop-refinement)将求精条件扩展到多跳, 增加了节点计算的冗余度, 以解决一跳范围内求精因计算条件不足导致节点无法求精的问题, 提高了求精的覆盖率。多跳的引入减小了因节点间距离测量误差较大而带来的节点定位误差, 提高了节点的定位精度。在每次迭代中采用加权最小二乘法来计算节点的位置, 并将相对置信度作为加权值, 有效地抑制了循环求精过程中误差的积累。

**基金项目:** 国家“863”计划基金资助项目(2006AA01Z222); 国家自然科学基金资助项目(60873240)

**作者简介:** 郭永红(1982—), 女, 硕士研究生, 主研方向: 无线传感器网络; 万江文, 教授、博士生导师; 于 宁, 博士研究生; 冯仁剑, 副教授

**收稿日期:** 2008-05-16 **E-mail:** sensory@buaa.edu.cn

### 3 基于跳数的求精

#### 3.1 存在的问题

一跳范围内的求精指在未知节点求精过程中只利用其一跳范围内的邻居节点的信息来更新自身的位置信息。在 Robust Positioning 定位算法<sup>[1]</sup>中主要采用该方法。其存在的问题如下:

(1)某些节点由于在初始阶段中所处的条件相同而导致节点得到的初始位置信息相同。在 100 m×100 m 的区域内随机布置 30 个节点。信标节点的 ID 为 0, 1, 2, 3, 4。表 1 是未知节点接收到的信标节点的 ID, 未知节点距离信标节点的跳数以及平均每跳的距离估计。可以看出, 在 5 组数据(13, 14, 23, 29; 10, 12, 20; 16, 26; 17, 24; 8, 18)中, 每组各个节点的信息完全相同, 其相应的初始位置坐标也相同。

表 1 未知节点接收到的信标节点的 ID

节点 ID	信标节点	跳数	每跳的距离/m
8	2, 3, 4	3, 3, 1	14.571 357 87
10	1, 2, 3, 4	5, 3, 5, 6	17.019 579 76
12	1, 2, 3, 4	5, 3, 5, 6	17.019 579 76
13	1, 2, 3, 4	1, 2, 1, 2	17.019 579 76
14	1, 2, 3, 4	1, 2, 1, 2	17.019 579 76
16	2, 3, 4	4, 4, 1	14.571 357 87
17	1, 2, 3, 4	3, 1, 3, 4	17.998 662 01
18	2, 3, 4	3, 3, 1	14.571 357 87
20	1, 2, 3, 4	5, 3, 5, 6	17.019 579 76
23	1, 2, 3, 4	1, 2, 1, 2	17.019 579 76
24	1, 2, 3, 4	3, 1, 3, 4	17.998 662 01
26	2, 3, 4	4, 4, 1	14.571 357 87
29	1, 2, 3, 4	1, 2, 1, 2	17.019 579 76

在求精阶段, 按节点坐标的计算方法, 当未知节点的邻居节点数目大于或等于 3 时(对于二维空间而言), 可利用极大似然法或三边法计算未知节点的坐标。但是, 由于初始阶段中出现的某些节点的初始坐标相同, 虽然某些未知节点其邻居节点的数目达到了位置计算的要求, 由于其邻居节点中的某些节点的坐标相同, 使可参加计算的邻居节点有效数目减少, 未知节点最终无法进行求精。仿真中节点 19 的邻居节点为 10, 12, 20, 24, 但是 10, 12, 20 的初始条件完全相同, 只能作为 1 个有效节点来看待, 因此, 节点 19 的有效邻居节点只有 2 个, 无法进行迭代求精。

(2)存在不良节点。不良节点<sup>[1, 6]</sup>分 2 种: 1)不可到达节点; 2)单个节点的一跳内只有 1 个或 2 个位置确定节点。单跳范围内的求精主要受后一种情况的影响, 由于不满足节点位置计算的条件, 因此不良节点无法再进行求精计算。

#### 3.2 基于跳数的求精

未知节点得到初始位置信息后, 根据预先设定的条件, 收集指定跳数范围内的有效参考节点的信息, 利用加权最小二乘法重新计算未知节点的位置, 若所得结果符合更新的条件, 则更新节点的位置信息, 进入下一轮循环求精, 直到满足迭代停止的条件。

求精可以分为 3 步:

(1)设定求精跳数(即阈值  $TTL$ ), 假设  $TTL=n, n>1$ 。

(2)在指定条件下, 未知节点根据收集的有效信息, 建立参考节点列表。列表中包括参考节点的 ID、坐标、未知节点到参考节点的最小跳数, 以及参考节点的相对置信度。在循环求精过程中, 未知节点接收到新的参考节点的信息, 对于新加入的参考节点, 将其信息添加到列表中, 如果不是新加入节点, 则更改相应参考节点的信息。

(3)未知节点根据列表的信息计算坐标, 如果新坐标符合

节点坐标更新的条件, 则更新节点的坐标, 并将新的坐标信息发送给其邻居节点, 进入下一次循环过程, 直到满足迭代终止的条件。迭代终止的条件如下:

1)在设定跳数内, 未知节点的初始列表中有效邻居节点的数目若少于 3 个, 则无法进行定位计算, 在迭代过程中舍弃该未知节点。

2)更新后节点的坐标与原坐标相比的差值小于节点通信半径的 1%。

3)达到限定的迭代次数。

#### 3.3 相对置信度

在求精计算中, 采用了加权最小二乘法计算节点的坐标。节点的置信度(即加权值)是对节点定位精度的评估, 节点置信度的值越大, 节点的定位精度越高。参考节点的置信度代表了参考节点在计算中决定权的大小, 定位精度高的参考节点, 对未知节点坐标计算产生的影响大, 反之影响小。

相对置信度由节点本身的定位精度和节点间的距离测量精度共同决定。节点本身的定位精度对于其他节点而言是一定的, 但是不同节点间的距离测量误差却有很大的不同。

假设节点  $X$  收到  $n$  个参考节点  $S_i(i=1, 2, \dots, n)$  的信息, 参考节点  $S_i$  的定位精度为  $\delta_i$ ,  $S_i$  与节点  $X$  间距离测量的精度为  $\mu_i$ , 则参考节点  $S_i$  相对于节点  $X$  的置信度为

$$w_i = \frac{1}{\sqrt{\delta_i^2 + \mu_i^2}}$$

#### 4 RP-Hop 定位算法

PR-Hop 算法基于 Robust Positioning(RP)定位算法, 利用 Hop-TERRAIN 得到节点初始位置, 用基于跳数的求精来代替 RP 定位算法中的求精算法, 以达到进一步提高定位精度的目的。

PR-Hop 算法描述如下:

(1)初始阶段: 利用 Hop-TERRAIN 定位算法得到未知节点的初始定位信息。

(2)循环求精阶段:

1)确定未知节点的初始置信度。RP 算法的求精阶段将信标节点的置信度设置为 1.0, 所有未知节点的置信度设置为 0.1, 即所有未知节点在初始阶段具有相同的定位精度。但实际上未知节点的初始定位精度有很大不同, 如图 1 节点误差分布所示, 在网络连通度为 3、信标节点比率为 30%、网络中共有 100 个节点的情况下, 未知节点的定位误差最高可达 458.8%, 最低也达到 12.7%, 多数节点的定位误差在 90%上下浮动。

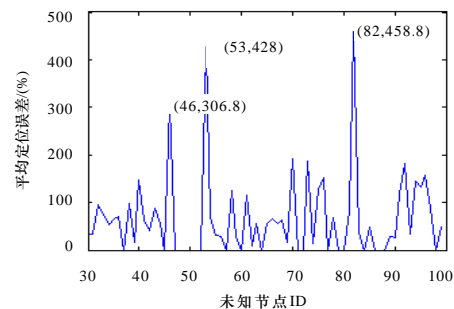


图 1 节点误差分布

图 1 中的节点 46、节点 53 和节点 82 等定位误差较大, 主要原因是连通度较差, 在信标节点到达未知节点的路径中, 未知节点的邻居节点个数少于 3。

根据未知节点的坐标计算方法, 未知节点的定位误差主

要由未知节点与相关信标节点间的距离误差决定。在网络中随着跳数的增加，节点间的测距误差也不断增加。

假设信标节点到达未知节点的路径中，未知节点的邻居节点个数为  $N$ ；未知节点接收到的信标节点个数为  $n$ ；信标节点到未知节点的跳数为  $hop_1, hop_2, \dots, hop_n$ ；则未知节点的初始置信度  $w_0$  的计算方法如式(1)：

$$w_0 = \begin{cases} 0.01 & N < 3 \\ \frac{\sum_{i=1}^n \sqrt{hop_i}}{N \times n} & N \geq 3 \end{cases} \quad (1)$$

2)得到未知节点所设定跳数内的节点的信息。

3)确定节点的相对置信度。节点本身的定位精度可由节点的置信度来衡量。由图 2 和图 3 可知，网络的连通度和网络中信标节点的比例对节点间平均距离的测量误差影响不大。其中，图 2 是在网络连通度为 8、不同信标节点比的情况下得到的，图 3 是在信标节点比为 20%、不同连通度的情况下得到的。因此，节点间的距离测量误差主要由节点间的跳数决定。

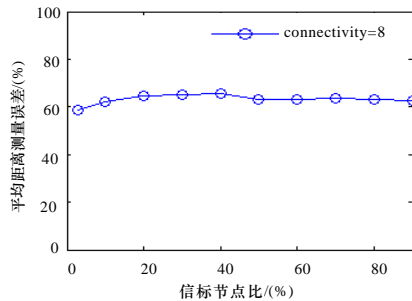


图 2 不同信标节点比下平均距离的测量误差

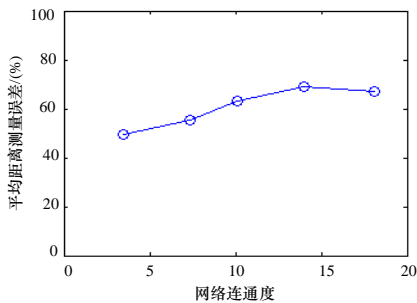


图 3 不同连通度下平均距离的测量误差

假设未知节点  $X$ ，其参考节点  $S_i (i=1, 2, \dots, n)$  到  $X$  的跳数为  $hop_i (i=1, 2, \dots, n)$ ，置信度为  $w_i$ ，则  $S_i$  相对  $X$  的相对置信度如下：

$$w_{xi} = \frac{1}{\sqrt{\left(\frac{1}{w_i}\right)^2 + hop_i^2}}, i = 1, 2, \dots, n \quad (2)$$

4)利用加权最小二乘法计算未知节点的坐标。如未知节点坐标计算成功，将所得坐标与原坐标相比，一旦满足迭代的条件，将其参考节点相对置信度的平均值作为未知节点的置信度，进入下一次迭代，如不满足则停止迭代。

## 5 RP-Hop 仿真实现

为检验 RP-Hop 算法的性能，采用 Ns-2 网络模拟器进行仿真比较。节点随机分布在  $100\text{ m} \times 100\text{ m}$  的区域内，节点个数为 100。

图 4 为在网络连通度为 3、不同信标比的情况下，RP 算法和 RP-Hop 算法的定位误差比较，其中，RP-Hop 的阈值分

别为 2, 3, 4。可见，RP-Hop 比 RP 算法定位精度有所提高。随着阈值的增加，RP-Hop 算法的定位精度也有提高，这表明在网络连通度较低时，节点的定位精度与阈值的大小相关。当信标节点较低时，由于未知节点的初始定位误差较大，RP-Hop 算法与 RP 算法相比节点的定位精度改变很小。

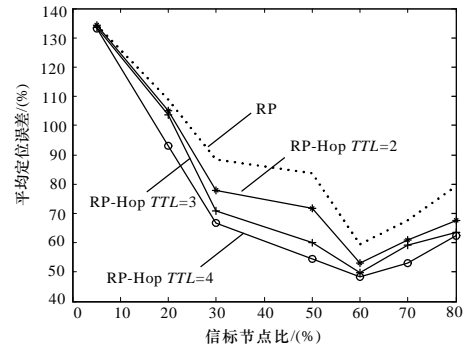


图 4 连通度为 3 时的平均定位误差

图 5 为在网络连通度为 8、不同信标比的情况下，RP 和 RP-Hop 的定位误差的比较。RP-Hop 的阈值分别为 2, 3, 4，RP-Hop 与 RP 算法相比，未知节点的定位精度提高了大约 10%~20%。与网络连通度为 3 时不同的是，随着阈值的增加，RP-Hop 算法的节点定位精度相差不大。这主要是网络连通性较高时，在不同阈值条件下，未知节点的有效邻居节点数目相差不大，则阈值设定的变化对节点的定位精度影响较小。另一方面在网络连通度为 8，信标节点比较小时 RP-Hop 与 RP 算法相比节点定位精度明显提高。

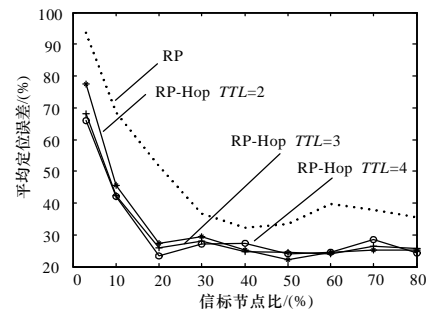


图 5 连通度为 8 时的平均定位误差

图 6 显示了当网络连通度为 8、RP-Hop 的阈值为 2 时，在不同信标节点比下，RP 和 RP-Hop 求精阶段节点求精的覆盖率比较。可以看出，在信标节点所占百分比比较低时，节点的覆盖率明显改善。

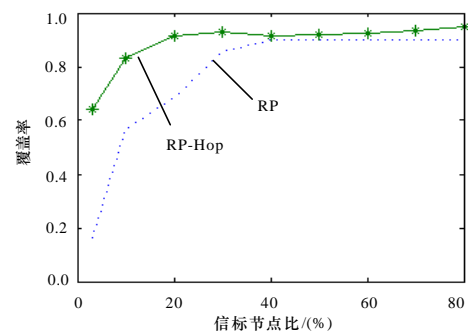


图 6 不同信标节点比下节点求精覆盖率

RP-Hop 与 RP 定位算法相比，解决了一跳范围内节点因有效邻居节点不足而无法求精的问题，提高了求精节点覆盖率和节点求精计算冗余度，同时相对置信度的应用有效地抑

(下转第 151 页)