

无线传感器网络定位的分布式求精算法

姚英彪¹, 姜男澜^{1,2}

(1. 杭州电子科技大学 通信工程学院, 浙江 杭州 310018; 2. 东南大学 移动通信国家重点实验室, 江苏 南京 211189)

摘 要: 通过对 WSN 中基于测距的定位进行误差分析提出一种求精算法。该算法每次迭代中首先根据“磁极”思想确定“误差节点”和“有效节点”; 然后在误差节点的邻居节点中选择 2 个相对偏差最小的节点作为圆心, 以它们到误差节点的测距值为半径分别作圆, 得到 2 个交点; 最后在它的当前定位位置和这 2 个交点之中选择误差较小的作为本轮的求精位置。仿真结果表明, 该算法能够降低多边定位模型产生的节点位置误差, 有效提高网络的定位精度。

关键词: 无线传感器网络; 定位; 误差分析; 求精算法

中图分类号: TP393

文献标识码: A

Distributed refinement algorithm for WSN localization

YAO Ying-biao¹, JIANG Nan-lan^{1,2}

(1. College of Communication Engineering, Hangzhou Dianzi University, Hangzhou 310018, China;

2. National Mobile Communications Research Laboratory, Southeast University, Nanjing 211189, China)

Abstract: A distributed refinement algorithm was proposed based on the error analyzing of range-based WSN localization. In each iteration, the proposed algorithm identified “error nodes” and “effective nodes” according to the idea of “magnetic pole” at first; then, it choosed two nodes with the minimum relative deviations in the 1-hop neighbors set of error node, and getted the intersection points of two circles whose centers were the positions of above two nodes and radiuses are the range distances from above two nodes to the error node respectively; at last, it choosed the position with minimum error, from its current position and above two intersection points as refinement position of error node after this iteration. Simulation results show that the proposed algorithm can reduce the localization error of multilateral positioning model and improve the localization accuracy efficiently.

Key words: wireless sensor networks; localization; error analysis; refinement algorithm

1 引言

无线传感器网络(WSN, wireless sensor network)通常由大量具有无线通信能力的传感器节点组成^[1], 这些节点具有通信、计算以及对环境进行测量的能力^[2]。WSN 主要应用于目标跟踪、环境监测、智能建筑等^[3], 在这些应用中收集的数据通常需要知道对应的采集位置, 这就要求确定网络中节点的位置, 定位技术是确定各节点位置的常用方法。由于 WSN 的能量和带宽受限, 所以不能对每个节点都采用 GPS 等成本和功耗较大的技术^[4,5], 而需要依

靠网络自身来定位。

定位算法包括基于测距的算法和无需测距的算法^[6,7]。基于测距的算法首先利用接收信号强度指示(RSSI)等技术测量节点间的距离, 再根据几何关系算出节点的位置。典型的基于测距的算法^[8-10]包括多边定位、MDS-MAP、边界盒定位等。无需测距的算法^[11, 12]通常利用网络连通性、节点分布、跳数等来实现定位, 典型的无需测距的算法^[13, 14]包括: 质心算法、DV-Hop、APIT 等, 和基于测距的算法相比, 无需测距的算法通常能节省更多的开销和能耗, 但基于测距的算法通常具有更高的定位精度。

收稿日期: 2013-07-25; 修回日期: 2014-09-11

基金项目: 国家自然科学基金资助项目(61100044)

Foundation Item: The National Natural Science Foundation of China(61100044)

为进一步提高定位性能,针对不同定位算法的求精算法被提出,这些求精算法能够提高原有定位算法的定位性能。现有求精算法通常包括2类,第一类是根据具体的定位算法过程对结果进行优化,包括 NCSG^[6]、基于梯度法的求精算法^[15]、基于 RLSM 的求精算法^[16]等;第二类是根据测距结果进行求精,包括基于测距值和定位结果的求精算法^[17]、基于 LQI 的求精算法^[18]等。与以上2类算法不同的是,本文在对定位结果进行误差分析的基础上实现求精。分析结果表明,当没有到达角度信息时,多边定位结果中会出现“镜像误差”和“孤邻节点误差”,并且这些误差还可能扩散,使整个网络的定位性能下降严重。针对这一情况,本文在1跳和2跳邻居节点信息的基础上,提出了一种基于“磁极”思想的求精算法(RAMP, refinement algorithm based on the idea of magnetic pole),该算法能够降低多边定位的误差,从而有效提高网络的定位精度。

2 定位误差分析

2.1 二维平面定位模型

多边定位^[1](MDS, multidimensional scaling)是一种经典的 WSN 二维平面定位模型,许多基于测距的定位算法都由多边定位演变而来,如半定规划(SDP)算法^[19]、SOCP 算法以及它们的变形^[20]。多边定位是求解以最小平方和为目标函数的非线性规划问题。假设网络中共有 N 个节点,布置在 $l \times l$ 大小的平面区域内; i 为其中某个节点, j 是可以和 i 直接通信的节点,一般称 i 和 j 互为“1跳邻居节点”;假设网络中有 m 个锚节点,它们的测量位置和已知位置分别用 \mathbf{n}_k 和 \mathbf{a}_k 表示($k=1, 2, \dots, m$),未知节点的定位位置用 \mathbf{n}_i 表示($i=m+1, m+2, \dots, N$)。用 δ 表示锚节点测量位置的误差上限, \mathcal{A} 表示锚节点的集合,这样就可以得到 WSN 多边定位的一般形式。

$$\begin{aligned} \min \sum_{(i,j) \in \mathcal{N}^1} \left(\|\mathbf{n}_i - \mathbf{n}_j\| - \hat{d}_{ij} \right)^2 + \sum_{k \in \mathcal{A}} \|\mathbf{n}_k - \mathbf{a}_k\|^2 \\ \text{s.t. } \|\mathbf{n}_k - \mathbf{a}_k\| \leq \delta, k \in \mathcal{A} \\ \mathbf{X} \preceq \mathbf{I} \\ \mathbf{Y} \preceq \mathbf{I} \end{aligned} \quad (1)$$

其中, $\|\cdot\|$ 表示 2-范数, \hat{d}_{ij} 是节点 i 和 j 之间的测量距离, \mathcal{N}^1 是网络中所有 1 跳邻居节点对构成的集

合, \mathbf{X} 为各节点的 x 坐标组成的列向量, \mathbf{Y} 为各节点 y 坐标组成的列向量, \mathbf{I} 是所有元素都为 1 的 N 维列向量,分量不等式 $\mathbf{X} \preceq \mathbf{I}$ 表示 \mathbf{X} 中每一个元素都小于 \mathbf{I} 中对应的元素。式(1)可用 Newton-Raphson 方法^[1]、内点法^[6]等求解。

式(1)还可以用分布式的形式表示。如果忽略锚节点位置误差,用 \mathcal{N}_i^1 表示经过路由表得到的节点 i 的所有 1 跳邻居节点的集合,则对于每一个节点 i 都应满足

$$\begin{aligned} \min_{\mathbf{n}_i} \sum_{j \in \mathcal{N}_i^1} \left(\|\mathbf{n}_i - \mathbf{n}_j\| - \hat{d}_{ij} \right)^2 \\ \text{s.t. } \mathbf{X} \preceq \mathbf{I} \\ \mathbf{Y} \preceq \mathbf{I} \end{aligned} \quad (2)$$

2.2 误差分析

对以式(1)或式(2)描述的定位问题,发现以下2种误差在其结果中经常出现,是造成多边定位算法性能下降的关键因素之一。

2.2.1 孤邻节点误差

本文将只有一个 1 跳邻居的节点称为“孤邻节点”,如图 1 所示, A 是孤邻节点, B 是其唯一的 1 跳邻居节点。图 1 中, R 是各节点的通信半径,本文假设所有节点通信半径相等。以 A 为圆心, AB 为半径的整个圆上存在无数个点满足式(2),所以孤邻节点的定位结果很可能存在较大的误差。在没有角度等信息的情况下,孤邻节点是很难准确定位的,并且这种误差还可能扩散。

通过 2 跳邻居节点信息可以降低这类误差,但前期仿真实验发现孤邻节点出现概率很低,例如,根据 1 000 次独立随机节点生成实验的结果,连通度为 2 的稀疏网络中,孤邻节点出现的比例为 1.32% 左右,连通度大于等于 5 时,孤邻节点出现的比例不大于 0.08%,而当连通度达到一种较常见的值——15 左右时,孤邻节点出现的比例仅为 0.01%。基于上述结果, RAMP 对孤邻节点进行屏蔽处理,即 RAMP 算法并不处理孤邻节点,其位置由定位算法确定。

实际操作时,各节点在运行 RAMP 算法前,首先根据 1 跳邻居节点数量判断自己是否为孤邻节点,如果是孤邻节点,则将这一信息发送给邻居节点,且不运行 RAMP 算法。通过这样的机制, RAMP 既能阻止这些孤邻节点误差扩散,又能限制自身的开销。

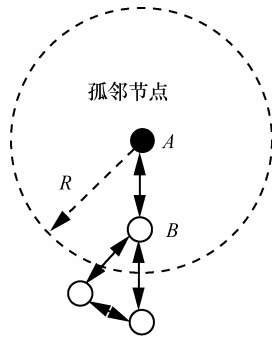


图 1 孤邻节点

2.2.2 镜像误差

本文将节点实际位置和定位位置关于 1 跳邻居节点连线对称或近似对称的情况称为“镜像误差”，如图 2 所示。

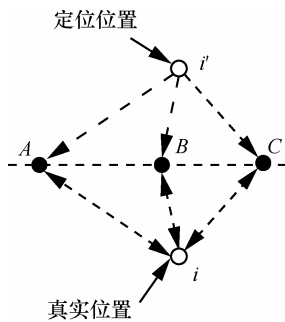


图 2 镜像误差

在图 2 中， i 的全部 3 个邻居节点 A 、 B 、 C 近似排列成一条直线，并且 i 不在这条直线上。在有测距误差的情况下， i 和 i' 的附近位置都可能使式(2)取极小值，如果定位算法将 i' 或其附近的位置作为定位位置则会造成镜像误差。镜像误差的产生并不局限于某一种定位算法，只要在定位过程中没有加入相关的约束条件，定位结果就可能存在镜像误差。这类误差通常较大，并且还可能扩散，使得其他节点也产生较大的定位误差，RAMP 主要考虑对这类误差的求精问题。

3 RAMP 求精算法

RAMP 是基于磁极思想的求精算法，包括误差判断和位置求精 2 个部分。假设 i 和它的 2 跳邻居节点具有相同的极性， i 和它的 1 跳邻居节点具有不同极性，磁极的有效作用距离就是节点的通信半径 R ；在对 i 进行求精时，其他节点位置固定，则一旦 i 的定位位置与其 2 跳邻居节点的距离在 R 以内就会受到“排斥”，从而达到求精目的。

3.1 孤邻节点判断

从孤邻节点的定义可以看出，通过路由表可以判断某个节点是否为孤邻节点。因此，在运行修正算法前，各个节点应首先确定本地路由表，如果路由表中只有一个 1 跳邻居节点，则该节点将自己设置为孤邻节点并通知其邻居节点。

3.2 误差判断

RAMP 借鉴了经典的无需测距定位思想^[6]，如式(3)所示。该思想实质上是利用了节点间的“距离—跳数矛盾”。具体而言，如果节点 i 和 j 是 1 跳邻居节点，则它们定位位置之间距离应该小于 R ；节点 i 和 j 是 2 跳邻居节点，则它们定位位置之间距离应该在 R 和 $2R$ 之间；如果不满足上面规律，则节点 i 或 j 的定位位置存在误差；在定位过程中，要使这个误差最小。

$$\begin{aligned} \min & \sum_{j \in N_i^1} (\|n_i - n_j\| - R)_+^2 + \sum_{j \in N_i^2} (R - \|n_i - n_j\|)_+^2 + \\ & \sum_{j \in N_i^2} (\|n_i - n_j\| - 2R)_+^2 \end{aligned} \quad (3)$$

$$\text{s.t. } \begin{aligned} X & \leq l \\ Y & \leq l \end{aligned}$$

其中， N_i^2 是 i 的所有 2 跳邻居节点的集合，运算符 $(\cdot)_+$ 的定义如下。

$$(\cdot)_+ = \max(\cdot, 0)$$

直观上来看，完全由式(3)中的距离—跳数矛盾进行误差大小的判断更为精确，但式(3)的第 2 项和第 3 项都是对 2 跳邻居节点距离—跳数矛盾的计算，这样的计算很容易造成误判，这是因为如果某个节点自身误差较小，而其 2 跳邻居节点存在较大误差，则该节点通过式(3)计算得到的结果也可能出现较大的值，从而使该节点被误认为存在较大的误差。前期仿真实验也证明了式(3)很容易造成上述误判，并且这种误判没有明显的规律，难以被 RAMP 算法识别，从而可能使求精后误差更大。为了降低误判的概率，RAMP 算法将磁极思想和距离—跳数矛盾思想相结合，把每个节点假想为一个磁极，则节点 i 应和 1 跳邻居节点相互吸引，和 2 跳邻居节点相互排斥，磁力的有效作用范围就是 R ，这样，式(3)的第 3 项就可以忽略不计，这样改进后虽然仍可能在开始的几次迭代中产生误判，但因为 RAMP 对位置更新有较为严格的限制条件，从而可以防止因误判使求精失败。

根据上述分析,如果忽略式(3)最后一项,然后令目标函数中每一项的误差权重相同,在考虑孤邻节点存在的条件下,基于磁极思想的误差水平判断关系式如下

$$E_i(k) = \begin{cases} \sum_{j \in (N_i^1 - \mathcal{L}_i^1)} \alpha_{ij}(k) + \sum_{j \in (N_i^2 - \mathcal{L}_i^2)} \beta_{ij}(k), & i \notin (\mathcal{L} + A) \\ N\mu & , i \in \mathcal{L}, i \notin A \\ 0 & , i \in A \end{cases} \quad (4)$$

其中, \mathcal{L} 是网络中所有孤邻节点的集合, \mathcal{L}_i^1 、 \mathcal{L}_i^2 分别是节点 i 的 1 跳和 2 跳邻居节点中孤邻节点的集合。 $\alpha_{ij}(k)$ 和 $\beta_{ij}(k)$ 的定义如下

$$\alpha_{ij}(k) = \begin{cases} \mu, & \|\mathbf{n}_i^{(k)} - \mathbf{n}_j^{(k)}\| > R \\ 0, & \text{其他} \end{cases} \quad (5)$$

$$\beta_{ij}(k) = \begin{cases} \mu, & \|\mathbf{n}_i^{(k)} - \mathbf{n}_j^{(k)}\| \leq R \\ 0, & \text{其他} \end{cases} \quad (6)$$

其中, $\mathbf{n}_i^{(k)}$ 是第 k 次迭代求精后节点 i 的位置($k \geq 1$), $\mathbf{n}_i^{(0)}$ 是 i 的初始定位位置, 将由式(4)计算得到的结果称为“误差水平”。RAMP 需要迭代运行多次, $E_i(k)$ 为第 k 次迭代求精后节点 i 的误差水平($k \geq 1$), $E_i(0)$ 表示定位算法运行结束后, 第一次迭代求精前(下称“初次定位后”)节点 i 的误差水平。通常情况下, 误差水平越高表明该节点出现较大定位误差的概率越大。

式(4)第一行是除孤邻节点和锚节点以外的节点的误差水平; 第二行是孤邻节点的误差水平, 它保证了网络中孤邻节点的误差水平始终最大; 最后一行是锚节点的误差水平, 因为锚节点误差通常较小, 所以将锚节点误差水平固定为 0。

式(4)~式(6)中的 μ 是由于 i 、 j 的当前定位位置之间的距离与 i 、 j 之间最短路径的跳数发生矛盾而产生的误差权重, 例如, i 、 j 的当前定位位置之间的距离为 $1.5R$, 则它们之间的最短路径至少为 2 跳, 而如果 i 、 j 在路由表上是 1 跳邻居节点, 就会产生矛盾。不妨以发生上述矛盾的邻居节点数量作为误差水平, 即令 $\mu = 1$ 。

图 3 是某网络初次定位后部分节点的分布情况, 其中, U 是某个未知节点的实际位置, U' 是 U 的初始定位位置, 节点 1、2、3 都是 U 的 1 跳邻居节点, 节点 4、5 都是 U 的 2 跳邻居节点, 因为 U' 不在节点 1 和节点 3 的通信范围内, 但在节点 4、节点 5 的通信范围内。根据式(4), 这时 U 的误差水平为 $E_U(0)=4$ 。

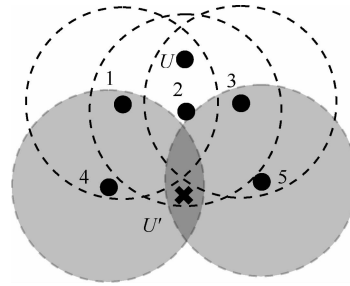


图 3 误差水平的计算

根据误差水平可以判断某个节点是否需要求精。通常一个节点要实现分布式定位, 至少需要 3 个位置较准确的 1 跳邻居节点。设节点 i 的 1 跳邻居节点数量是 W_i , 第 k 次迭代求精后或初次定位后与节点 i 发生距离—跳数矛盾的 1 跳邻居节点数量是 $W'_i(k)$, 则如果 $W'_i(k)$ 满足

$$W_i - W'_i(k) < 3, i \notin \mathcal{L} \quad (7)$$

则可以认为节点 i 需要求精, 本文称 i 为“误差节点”, 除了误差节点和孤邻节点以外的节点都称为“有效节点”。特别地, 若 i 只有 2 个 1 跳邻居节点, 则其任意邻居节点都不应与其发生距离—跳数矛盾, 考虑这种特殊情况后可将式(7)改写为比值的形式。

$$\frac{W'_i(k)}{W_i} > T_e, i \notin \mathcal{L} \quad (8)$$

其中, T_e 是判断误差节点的门限值, 且有

$$T_e = \begin{cases} 1 - \frac{3}{W_i}, & W_i \geq 3 \\ 0, & W_i = 2 \end{cases}$$

当 $W_i \geq 3$ 时, 上式中的 $T_e = 1 - 3/W_i$ 可以松弛为

$$1 - \frac{3}{W_i} \leq T_e < 1 - \frac{2}{W_i}$$

显然, 松弛后并不影响式(8)对误差节点的判断。为了用统一的门限值判断误差节点, 将 T_e 的取值范围进一步松弛为 $[0, 1 - 2/W_i]$ 。类似地, 当 $W_i = 2$ 时, T_e 的取值范围可以松弛为 $[0, 0.5]$ 。可以证明, 松弛后对于所有的 $W_i \geq 2$, T_e 的取值范围交集为 $[0, 1/3)$, 因此, 可以在这个交集选择一个值作为固定门限来判断误差节点。第一次松弛后, 对于 $W_i = 2, 3, 4, 5, \dots$, T_e 的取值范围下限依次为 0, 0, 0.25, 0.4, \dots , 因此, 不妨统一取 $T_e = 0.25$ 。

若再将式(8)中的 $W'_i(k)$ 松弛为 $E_i(k)$, 就可以通

过误差水平来判断误差节点, 如果 $E_i(k)$ 满足

$$\frac{E_i(k)}{W_i} > 0.25, i \in \mathbb{L} \quad (9)$$

则节点 i 就是误差节点, RAMP 只对误差节点进行求精。这样松弛后可能将有效节点误判为误差节点, 但不会将误差节点误判为有效节点, 实验表明, 随着迭代求精的进行, 出现误判的概率将逐渐降低。此外, RAMP 在求精过程中引入了位置更新的限制条件, 以防止有效节点被错误地调整位置。

3.3 位置求精

位置求精分为计算节点位置和更新节点位置 2 部分, RAMP 将计算得到的位置(下称“计算位置”)和当前位置比较, 然后根据位置更新规则来更新。在讨论求精算法前, 先介绍几个相关概念。

3.3.1 子偏差

在式(2)中求和项的基础上考虑孤邻节点的存在, 即可得到子偏差。用 $\sigma_i(k)$ 表示第 k 次($k \geq 1$)位置更新后节点 i 的子偏差, $k=0$ 时 $\sigma_i(k)$ 是初次定位后的子偏差, 则有

$$\sigma_i(k) = \begin{cases} \sum_{j \in (\mathbb{N}_i^1 - \mathbb{L}_i^1)} \left\| \mathbf{n}_i^{(k)} - \mathbf{n}_j^{(k)} \right\| - \hat{d}_{ij}, & i > m \\ 0, & 1 \leq i \leq m \end{cases} \quad (10)$$

假设第 k 次迭代求精中, 节点 i 的计算位置为 $\mathbf{t}_i^{(k)}$, 则位置更新前节点 i 的子偏差为

$$\sigma_i'(k) = \begin{cases} \sum_{j \in (\mathbb{N}_i^1 - \mathbb{L}_i^1)} \left\| \mathbf{t}_i^{(k)} - \mathbf{n}_j^{(k-1)} \right\| - \hat{d}_{ij}, & i > m \\ 0, & 1 \leq i \leq m \end{cases} \quad (11)$$

子偏差实际上是对各节点相对定位误差的近似计算, RAMP 根据子偏差的大小来决定是否更新节点位置。

3.3.2 有效节点相对偏差

假设 i 的 1 跳邻居节点 j 是有效节点, 则第 k 次迭代时, 有效节点 j 相对 i 的偏差为

$$\xi_{ij}(k) = E_j(k-1) + \sigma_j(k-1) \quad (12)$$

RAMP 选择 1 跳邻居有效节点中相对偏差最小的 2 个节点辅助误差节点进行求精。有效节点相对偏差将子偏差和误差水平相加, 使在多个有效节点子偏差相同或相近时发生距离—跳数矛盾较少的被选作辅助求精的节点, 从而提高求精的有效性。

3.3.3 定位点偏差

假设误差节点 i 可能的位置是网络区域中的

一点 v , 用 $L_{iv}(k)$ 表示第 k 次迭代时点 v 的定位点偏差, 则

$$L_{iv}(k) = \sum_{j \in \mathbb{E}_i^1} \left\| \mathbf{t}'_v - \mathbf{n}_j^{(k-1)} \right\| - \hat{d}_{ij} - \sum_{u \in \mathbb{E}_i^2} \gamma_{vu}(k) \quad (13)$$

其中,

$$\gamma_{vu}(k) = \begin{cases} \mu, \left\| \mathbf{n}_i^{(k-1)} - \mathbf{n}_u^{(k-1)} \right\| \leq R, \\ \left\| \mathbf{n}_i^{(k-1)} - \mathbf{n}_u^{(k-1)} \right\| \leq \left\| \mathbf{t}'_v - \mathbf{n}_u^{(k-1)} \right\| \\ 0, \text{其他} \end{cases} \quad (14)$$

\mathbf{t}'_v 表示网络区域中一点 v 的位置。 \mathbb{E}_i^1 、 \mathbb{E}_i^2 分别是节点 i 的 1 跳和 2 跳邻居节点中有效节点的集合。RAMP 以辅助求精的 1 跳邻居节点为圆心, 它们到误差节点 i 的测距值为半径作 2 个圆, 从而得到 2 个交点, 将其中定位点偏差较小的一个作为 i 的计算位置。

3.3.4 迭代误差水平

式(4)中的 $E_i(k)$ 是第 k 次($k \geq 1$)位置更新后 i 的误差水平, 而 i 的第 k 次迭代中计算位置的误差水平(下称“迭代误差水平”) $D_i(k)$ 用式(15)来确定。

$$D_i(k) = \begin{cases} \sum_{j \in (\mathbb{N}_i^1 - \mathbb{L}_i^1)} \eta_{ij}(k) + \sum_{j \in (\mathbb{N}_i^2 - \mathbb{L}_i^2)} \lambda_{ij}(k), & i \notin (\mathbb{L} + A) \\ N\mu, & i \in \mathbb{L}, i \notin A \\ 0, & i \in A \end{cases} \quad (15)$$

$$\eta_{ij}(k) = \begin{cases} \mu, \left\| \mathbf{t}_i^{(k)} - \mathbf{n}_j^{(k-1)} \right\| > R \\ 0, \text{其他} \end{cases} \quad (16)$$

$$\lambda_{ij}(k) = \begin{cases} \mu, \left\| \mathbf{t}_i^{(k)} - \mathbf{n}_j^{(k-1)} \right\| \leq R \\ 0, \text{其他} \end{cases} \quad (17)$$

3.3.5 位置更新规则

在得到子偏差和迭代误差水平后, RAMP 按照如下规则更新节点位置。

$$\mathbf{n}_i^{(k)} = \begin{cases} \mathbf{t}_i^{(k)}, D_i(k) < E_i(k-1) \text{ 或 } \sigma_i'(k) < \sigma_i(k-1) \\ \mathbf{n}_i^{(k-1)}, \text{其他} \end{cases} \quad (18)$$

式(18)表明, 只要本次迭代计算得到的位置使节点的误差水平或子偏差有所下降则更新节点位置。

3.3.6 迭代阈值和最大迭代次数

为防止算法无限迭代下去, 可以设置一个迭代阈值 T_r 和最大迭代次数 I , 当连续 T_r 次迭代定位位置没有改变或达到最大迭代次数 I , 则节点 i 停止求精过程。

3.4 基于磁极思想的分布式求精算法

基于磁极思想的分布式求精算法具体步骤如下。

Step1 各节点判断自己是否为孤邻节点。

Step2 根据式(4)计算各节点的误差水平。

Step3 根据式(9)确定误差节点和有效节点。

Step4 对于第 k 次迭代求精($k \geq 1$), 如果 i 是误差节点, 则根据式(12)计算 i 的各 1 跳邻居有效节点相对 i 的偏差, 选出相对偏差最小的 2 个节点 A 和 B 。若邻居节点中有效节点个数不足 2 个, 则本次不进行求精。

Step5 分别以 A 和 B 的当前定位位置为圆心, 它们到误差节点的测距值为半径作 2 个圆 C_1 和 C_2 。

Step6 根据式(13)计算 C_1 和 C_2 的 2 个交点 p 、 q 的定位点偏差, 将定位点偏差较低的点确定为误差节点的计算位置 r , r 应满足

$$L_{ir}(k) = \min[L_{ip}(k), L_{iq}(k)]$$

Step7 根据式(11)计算其子偏差 $\sigma'_i(k)$, 根据式(15)计算节点的迭代误差水平 $D_i(k)$ 。

Step8 根据式(18)更新各节点位置。

Step9 重复 Step2 到 Step8, 直到连续 T_r 次迭代定位位置没有改变或达到最大迭代次数 I 。

图 4 是对图 3 中的 U 进行求精的示意, 其中, 相对 U 的偏差最小的 2 个 1 跳邻居有效节点是 1 和 3, 它们到 U 的测距结果分别为 \hat{d}_{u1} 和 \hat{d}_{u3} , 则以节点 1 为圆心、 \hat{d}_{u1} 为半径画出圆 C_1 , 以节点 3 为圆心、 \hat{d}_{u3} 为半径画出圆 C_2 , 两圆交于 p 、 q 。由于 q 的定位点偏差比 p 要低, 因此将 q 作为节点 U 的新位置, 受测距误差的影响, q 和 U 的实际位置并未重合, 但已经远优于 U' 。

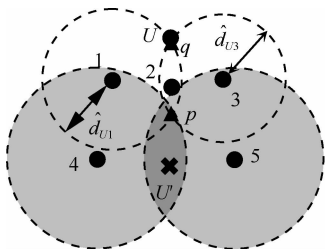


图 4 RAMP 算法

4 仿真结果及分析

本节通过 Matlab 仿真测试 RAMP 的性能。以 1×1 的矩形区域作为网络范围, 即 $x, y \in [0, 1]$, 设

置 100 个节点, 其中 8 个锚节点坐标分别为 $[0.1, 0.1]$, $[0.1, 0.5]$, $[0.1, 0.9]$, $[0.5, 0.1]$, $[0.5, 0.9]$, $[0.9, 0.1]$, $[0.9, 0.5]$, $[0.9, 0.9]$, 其他 92 个节点随机分布于网络中, 它们的坐标在网络范围内服从 $U(0,1)$ 分布, 通信半径 $R = 0.22$ 。测距模型为

$$\hat{d}_{ij} = \begin{cases} d_{ij} \max(|1 + nf \cdot randn|, 1 - s), & |1 + nf \cdot randn| \leq 1 \\ [d_{ij} \min(|1 + nf \cdot randn|, 1 + s)]_{R^-}, & |1 + nf \cdot randn| > 1 \end{cases} \quad (19)$$

其中, i, j 是 1 跳邻居节点, $[\cdot]_{R^-}$ 的定义如下

$$[\cdot]_{R^-} = \min(\cdot, R)$$

上式中的 d_{ij} 是节点 i 和 j 之间的真实距离, nf 是噪声因子, s 是误差限, $randn$ 是服从 $N(0,1)$ 分布的随机数。式(19)限制了 RAMP 使用的测距值不能大于通信半径, 实际上, 当测距值大于通信半径时可以将其修正为通信半径的值。此外, 以 nfa 作为测量锚节点位置的噪声因子, 当 $nfa = 0$ 时表明锚节点没有位置误差。

实验中定位算法选择内点法求解式(1), 并忽略该式目标函数中的第 2 项, 以此得到各节点的初始定位位置, 然后再运行本文提出的 RAMP 求精算法。仿真中, 根据前期实验和文献[18,21]的结果, 将 RAMP 的迭代阈值 T_r 设置为 10, 最大迭代次数 I 设置为 50, 测距误差上限 s 设置为 0.5。

4.1 无锚节点位置误差时的求精

本节对没有锚节点位置误差的情况进行分析, 这时锚节点的测量位置就是其实际位置。对 nf 取不同的值各进行 30 次实验。

图 5 是随机挑选的几次实验中误差节点数量的变化情况。图 6 是图 5 中对应实验的全网相对定位误差的变化情况, 全网相对定位误差的定义如下

$$\varepsilon = \frac{1}{N} \sum_{i=1}^N \frac{\sqrt{(\hat{x}_i - x_i)^2 + (\hat{y}_i - y_i)^2}}{R} \quad (20)$$

其中, (\hat{x}_i, \hat{y}_i) 是节点 i 的当前定位位置, (x_i, y_i) 是其实际位置。

结合图 5 和图 6 可以看出, 当 $nf = 0$ 时, 第 2 次迭代后误差节点数量和全网相对定位误差都不再变化, 将第 2 次迭代和之前的迭代都称为有效迭代, 后面的迭代都称为无效迭代, 通过实验得到有效迭代次数能在实际应用中减少不必要的迭代以节省开销。

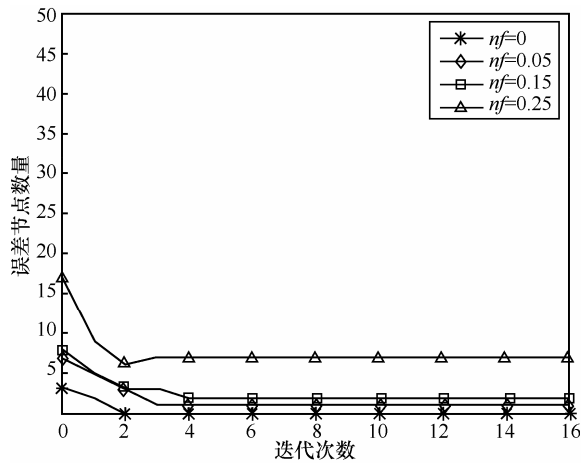


图 5 误差节点数量, nfa=0

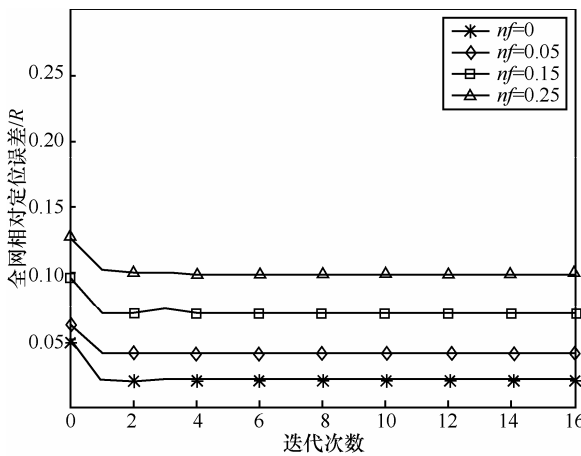


图 6 全网相对定位误差, nfa=0

在没有锚节点误差时, nf 取不同值的有效迭代次数、求精后误差节点下降比例、全网相对定位误差下降比例如表 1 所示, 所有结果均是 30 次独立实验的平均值。

表 1 求精算法的性能, nfa=0

噪声因子	有效迭代次数	误差节点数量下降比	全网相对定位误差下降比
$nf=0$	2.90	79.25%	41.05%
$nf=0.05$	3.37	57.93%	23.87%
$nf=0.10$	4.07	53.51%	18.77%
$nf=0.15$	4.57	50.17%	14.28%
$nf=0.20$	4.97	49.52%	13.36%
$nf=0.25$	5.30	40.61%	11.40%

从表 1 中可以看到, 无锚节点位置误差时, 求精后误差节点数量平均降低 40% 以上, 全网相对定位误差平均降低 11% 以上, RAMP 的平均有效迭代次数均不超过 6 次。

图 7 是图 5 中对应实验的节点分布情况, 图中“×” (叉号) 表示锚节点的实际位置, 菱形表示锚节点的测量位置, “+” (加号)、三角形和圆环分别表示未知节点的实际位置、初始定位位置和最终求精位置, 从未进行求精的节点的最终求精位置没有单独标注。

从图 7 可以看出, 网络中定位误差较大的节点基本都得到了较好的求精结果, 部分节点的求精位置和初始定位位置之间呈现较明显的镜像关系, 由于测距误差的存在, 图 7 中某些节点的求精位置仍然存在一定误差, 但多数节点的误差都明显小于初始定位位置的误差。

4.2 存在锚节点位置误差时的求精

本节引入最大 $\sqrt{2}R$ 的锚节点测量位置误差, 误差大小随机产生, 其他参数和上一小节相同。锚节点的测量位置按下式产生。

$$\hat{x}_i = \begin{cases} [x_i + R\min(nfa \cdot randn, 1)]_{\Gamma} & , randn \geq 0 \\ [x_i + R\max(nfa \cdot randn, -1)]_{0^+} & , randn < 0 \end{cases}$$

$$\hat{y}_i = \begin{cases} [y_i + R\min(nfa \cdot randn, 1)]_{\Gamma} & , randn \geq 0 \\ [y_i + R\max(nfa \cdot randn, -1)]_{0^+} & , randn < 0 \end{cases}$$

上式中的 $[\cdot]_{0^+}$ 的定义如下

$$[\cdot]_{0^+} = \max(\cdot, 0)$$

在前述仿真环境下, 10 组重复 1 000 次的锚节点位置测量实验表明, 将 nfa 设置为 0.1 时, 锚节点测量位置的平均误差在 $10.88\%R \sim 11.82\%R$ 之间。实际场景下, 根据文献[22]的结果, 大多数室外环境中通信半径都可以达到 30 m 或更高; 如果节点采用 GPS 单点定位, 其定位误差在可以控制在 3 m 以内^[23], 即位置误差在 $10\%R$ 以内, 这就是说, nfa 设置为 0.1 能满足大多数室外环境的锚节点测量位置误差要求。因此本节将 nfa 设置为 0.1 来测试存在锚节点位置误差时 RAMP 算法的求精性能。

对 nf 取不同的值各进行 30 次实验, 图 8 是随机挑选的几次实验中误差节点的变化情况; 图 9、图 10 分别是图 8 中对应实验的全网相对定位误差变化情况和节点分布情况, 图 10 中各类符号的含义和图 7 相同。

从图 8~图 10 中可以看出, 锚节点存在一定位置误差时, 多数误差较大的节点都能经过较少次数的迭代有效降低定位误差。

nf 取不同值时的有效迭代次数、求精后误差

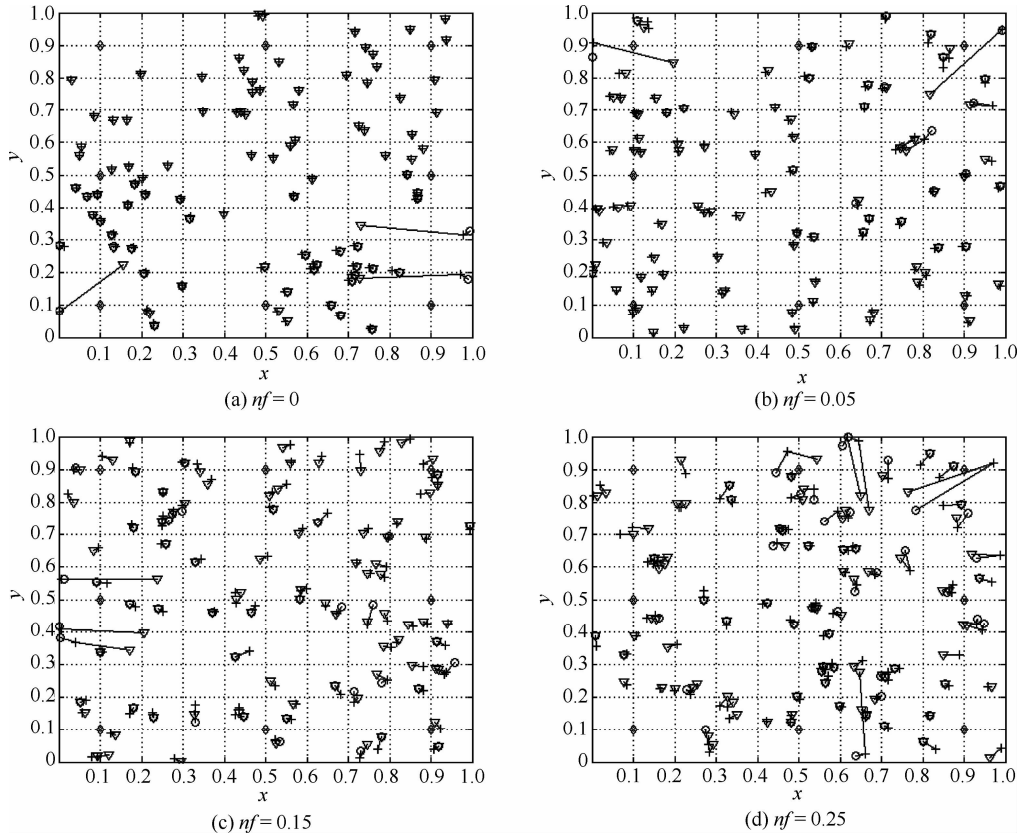


图 7 节点分布, $nfa=0$

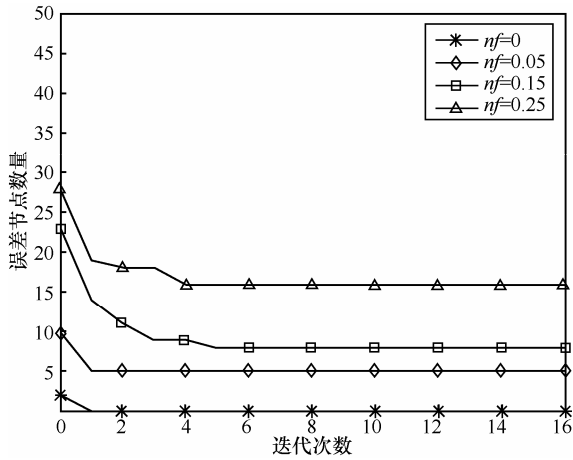


图 8 误差节点数量 $nfa=0.10$

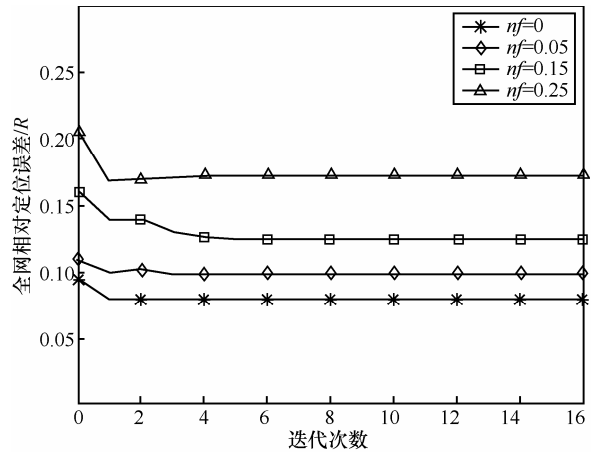


图 9 全网相对定位误差, $nfa=0.10$

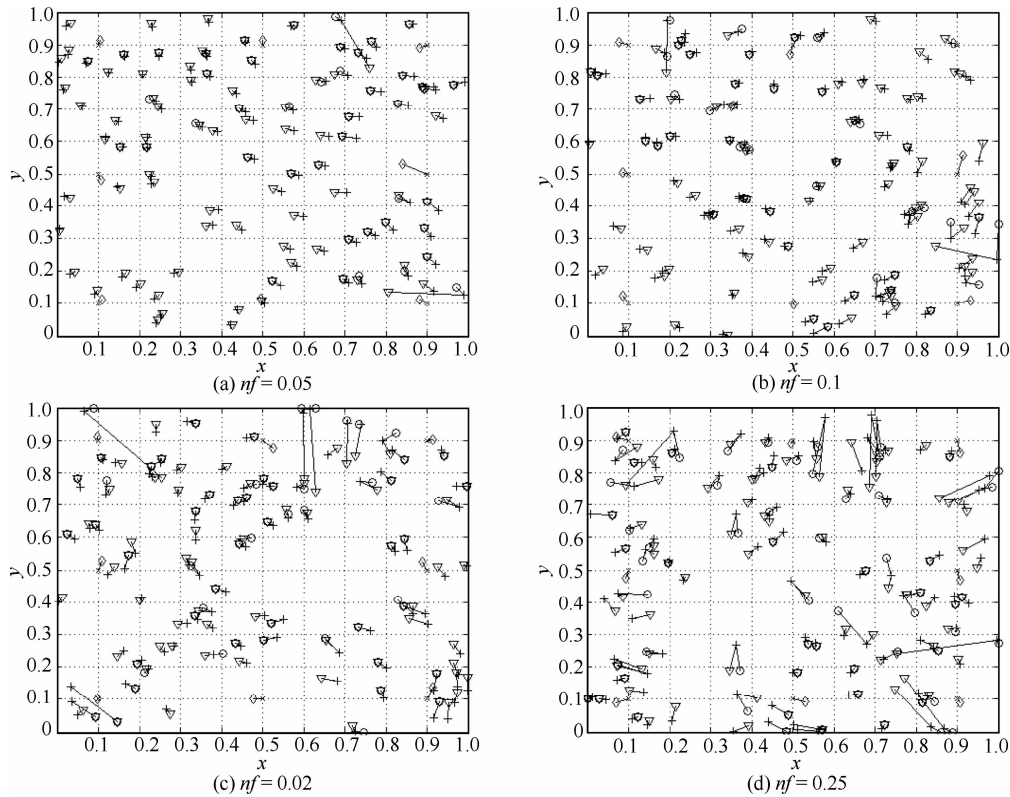
节点下降比例、全网相对定位误差下降比例如表 2 所示, 所有结果均是 30 次独立实验的平均值。从表 2 中可以看出, 存在锚节点位置误差时, RAMP 求精后误差节点数量平均降低 40% 以上, 全网相对定位误差平均降低 7% 以上, 平均有效迭代次数均不超过 5 次。

4.3 对网络性能的影响

对于 WSN 而言, 定位精度和包括算法能耗、

时延在内的开销都对网络性能有重要影响, 而精度和开销在一定程度上相互矛盾, 又相互关联。获得较高的精度通常需要更多的开销, 降低网络寿命, 而精度较低则可能使网络工作效率低下, 如何对这 2 个指标进行权衡要根据实际应用来确定。

一方面, RAMP 算法在提高定位精度的同时会增加整个定位过程的能耗开销, 但由于求精过程通常是渐进的, 所以可以根据应用需求控制迭

图10 节点分布, $nfa=0.1$

代次数,从而用尽量少的能耗开销实现所需的定位精度。例如,对于定位精度非常重要的网络,RAMP采用本文的设置能有效提升网络定位性能;对于精度和能耗开销同等重要的网络,可以修改本文RAMP中的一些设置,如迭代次数或误差节点判断门限,实现用较少能耗开销换取一定程度的定位精度提升;而对于更注重能耗开销的网络,则可能需要设计其他更简单的求精算法。

表2 求精算法的性能, $nfa=0.10$

噪声因子	有效迭代次数	误差节点数量下降比	全网相对定位误差下降比
$nf=0.05$	4.17	60.44%	16.40%
$nf=0.10$	4.63	51.71%	14.32%
$nf=0.15$	4.63	46.88%	10.10%
$nf=0.20$	4.77	40.45%	10.50%
$nf=0.25$	5.00	44.25%	7.70%

另一方面,RAMP算法在提高定位精度的同时会增加整个定位过程的时延开销,但由于它的迭代次数、每次迭代过程中的计算量都远少于定位算法相应的量,因此其延时开销和定位算法相比是有限的。

5 结束语

在基于测距的WSN定位算法中难免出现镜像误差、孤邻节点误差以及由这些误差引起的扩散误差,本文提出的RAMP算法首先确定孤邻节点、误差节点及有效节点,然后根据有效节点相对偏差及测距值反复计算误差节点的新位置,最后利用更新规则更新节点位置。仿真结果表明,不论是否存在锚节点位置误差,RAMP都能以较少的迭代次数显著降低多边定位中的误差节点数量,有效降低全网相对定位误差,提高定位精度,同时,RAMP算法的思想也能应用于其他基于测距的定位算法,降低相应算法的定位误差。今后的工作应可以从定位性能、能耗、时延三者联合的角度改进RAMP算法的参数设置和求精过程。

参考文献:

- [1] STOJIMENOVIC I. Handbook of Sensor Networks Algorithms and Architectures[M]. New Jersey: John Wiley & Sons, 2005.
- [2] WANG L, WANG X P. A new WSN localization algorithm based on regularization method[A]. 2011 IEEE Instrumentation and Measurement Technology Conference (I2MTC)[C]. Hangzhou, China, 2011. 1-4.
- [3] 王洁, 王洪玉, 高庆华等. 一种适用于移动传感器网络的增强型蒙

- 特卡罗定位跟踪算法[J]. 电子与信息学报, 2010, 32(4): 864-868.
WANG J, WANG H Y, GAO Q H, *et al.* Enhanced Monte Carlo localization and tracking algorithm for mobile wireless sensor network[J]. Journal of Electronics & Information Technology, 2010, 32(4): 864-868.
- [4] CHEN Y F, SHU L, LI M C, *et al.* The insights of DV-based localization algorithms in the wireless sensor networks with duty-cycled and radio irregular sensors[A]. 2011 IEEE International Conference on Communications (ICC)[C]. Kyoto, Japan, 2011. 1-6.
- [5] 王福豹, 史龙, 任丰原等. 无线传感器网络中的自身定位系统和算法[J]. 软件学报, 2005, 16(5): 857-866.
WANG F B, SHI L, REN F Y, *et al.* Self-localization systems and algorithms for wireless sensor networks[J]. Journal of Software, 2005, 16(5): 857-866.
- [6] SHI Q J, HE C, CHEN H Y, *et al.* Distributed wireless sensor network localization via sequential greedy optimization algorithm[J]. IEEE Transactions on Signal Processing, 2010, 58(6): 3328-3340.
- [7] XU Y X, GAO X, SUN Z Y, *et al.* WSN node localization algorithm design based on rssi technology[A]. 2012 Fifth International Conference on Intelligent Computation Technology and Automation (ICICTA)[C]. Zhangjiajie, China, 2012. 556-559.
- [8] SUN X L, CHEN T, LI W Q, *et al.* Performance research of improved MDS-MAP algorithm in wireless sensor networks localization[A]. 2012 International Conference on Computer Science and Electronic Engineering (ICCSEE 2012)[C]. Hangzhou, China, 2012. 587-590.
- [9] PENG A P, GUO X S, CAI W, *et al.* A distributed localization scheme for wireless sensor networks based on bounding box algorithm[A]. The Ninth International Conference on Electronic Measurement & Instruments[C]. Beijing, China, 2009. 984-988.
- [10] 王行甫, 刘志强, 黄秋原等. WSN中一种改进的边界盒定位算法[J]. 计算机工程, 2011, 37(20): 57-59.
WANG X F, LIU Z Q, HUANG Q Y, *et al.* Improved bounding-box localization algorithm in WSN[J]. Computer Engineering, 2011, 37(20): 57-59.
- [11] 曾凡仔, 孙正章, 罗娟等. 无线传感器网络的节点定位方法[J]. 通信学报, 2008, 29(11): 62-66.
ZENG F Z, SUN Z Z, LUO J, *et al.* Improved node localization algorithm for wireless sensor network[J]. Journal on Communications, 2008, 29(11): 62-66.
- [12] LI N Q, LI P. A range-free localization scheme in wireless sensor networks[A]. IEEE International Symposium on Knowledge Acquisition and Modeling Workshop[C]. Wuhan, China, 2009. 525-528.
- [13] ZHANG Y J, WANG K, YUAN S F, *et al.* Research of WSN node localization algorithm based on weighted DV-HOP[A]. 2012 24th Chinese Control and Decision Conference (CCDC)[C]. Taiyuan, China, 2012. 3826-3829.
- [14] TAN H B, LIU F. Research and implementation of APIT positioning algorithm in WSN[A]. 9th International Conference on Fuzzy Systems and Knowledge Discovery (FSKD 2012)[C]. Sichuan, China, 2012. 2212-2215.
- [15] SHI Q Q, HUO H, FANG T, *et al.* A distributed node localization scheme for wireless sensor networks[J]. Wireless Personal Communications, 2010, 53(1): 15-33.
- [16] REICHENBACH F, TIMMERMANN D. On improving the precision of localization with minimum resource allocation[A]. 16th International Conference on Computer Communications and Networks[C]. Honolulu, United States, 2007. 1093-1098.
- [17] LIU Y Y, CHENG W, ZHANG L Y. A distributed location correction algorithm in the WSN[A]. 2006 International Symposium on Communications and Information Technologies[C]. Bangkok, Thailand, 2006. 166-170.
- [18] SHU J, GONG J J, CHEN Y B, *et al.* BR²OM: RSSI-based refinement and optimization mechanism for wireless sensor networks[J]. Journal of Networks, 2009, 4(10): 1017-1025.
- [19] BISWAS P, LIANG T C, TOH K C, *et al.* Semidefinite programming approaches for sensor network localization with noisy distance measurements[J]. IEEE Transactions on Automation Science and Engineering, 2006, 3(4): 360-371.
- [20] SRIRANGARAJAN S, TEWFIK A H, LUO Z Q. Distributed sensor network localization using SOCP relaxation[J]. IEEE Transactions on Wireless Communications, 2008, 7(12): 4886-4895.
- [21] ZHANG J W, ZHANG L. Research on distance measurement based on RSSI of ZigBee[A]. 2009 Second ISECS International Colloquium on Computing, Communication, Control, and Management[C]. Sanya, China, 2009. 210-212.
- [22] 徐兴元, 章玥, 季民河等. 农业生态环境监测中无线传感节点信号有效传输距离的确定[J]. 农业工程学报, 2013, 29(14): 164-170.
XU X Y, ZHANG Y, JI M H, *et al.* Determination of effective transmission distances of wireless sensor network nodes for agro-ecological environment monitoring[J]. Transactions of the Chinese Society of Agricultural Engineering, 2013, 29(14): 164-170.
- [23] 孙罡, 王昌明, 张爱军. GPS 静态单点定位的滤波算法比较[J]. 南京理工大学学报(自然科学版), 2011, 35(1): 80-85.
SUN G, WANG C M, ZHANG A J. Comparison of filtering algorithms for GPS Static point positioning[J]. Journal of Nanjing University of Science and Technology (Nature Science), 2011, 35(1): 80-85.

作者简介:



姚英彪 (1976-), 男, 湖北松滋人, 博士, 杭州电子科技大学副教授, 主要研究方向为嵌入式系统设计、媒体信号处理和无线传感器网络等。



姜男澜 [通信作者] (1988-), 男, 重庆人, 东南大学博士生, 主要研究方向为无线传感器网络、车联网等。E-mail: jiangnanlan@qq.com。