

Iteration-Based Localization Algorithm for Wireless Sensor Network in Three-Dimensional Space *

QI Rongbin^{1*}, LI Sijin¹, MA Tianyi², ZHONG Weimin¹, QIAN Feng¹

(1. Key Laboratory of Advanced Control and Optimization for Chemical Processes, Ministry of Education, East China University of Science and Technology, Shanghai 200237, China; 2. Department of Computer Science and Technology, Shanghai Business School, Shanghai 200235, China)

Abstract: There exist low precision, high computational complexity and large amount of data transmission for the localization problem of wireless sensor network in three-dimensional space. To overcome these shortcomings, a distributed algorithm based on distance is proposed, which is called iterative localization algorithm and with high-precision (namely ILAH-3D). This method extends the two-dimensional, classical algorithm AHLos to be applied to the three-dimensional space. The cumulative errors can be reduced and localization precision can be improved by using the weighted least squares method and adding the upgraded validation condition of localized node. And the performing conditions of the constrained cooperation algorithm are given according to the various position relations of those nodes. Then localization algorithm of the new iteration is carried on by combining with the upgraded anchor nodes. The computing complexity of ILAH-3D is simple and its amount of communication is small. Compared with the existing three-dimensional localization algorithms, this proposed algorithm can achieve very good accuracy even though there are some errors in the ranging.

Key words: wireless sensor network; three-dimensional localization; iteration; ranging

EEACC: 6150P

doi: 10.3969/j.issn.1004-1699.2012.05.018

基于迭代的无线传感器网络三维定位算法 *

祁荣宾^{1*}, 李思瑾¹, 马天义², 钟伟民¹, 钱 锋¹

(1. 华东理工大学化工过程先进控制和优化技术教育部重点实验室, 上海 200237; 2. 上海商学院计算机科学与技术系, 上海 200235)

摘 要: 针对目前无线传感器网络三维空间定位存在精度不高、计算复杂、传输数据量大等问题, 提出了一种基于测距的分布式算法——高精度迭代三维定位算法 (ILAH-3D)。该方法将经典二维算法 AHLos (Ad-Hoc Localization System) 扩展到三维空间, 通过采用加权最小二乘法和加入定位节点升级锚节点的验证条件来减少累计误差, 提高节点定位精度, 并根据三维空间中节点的各种位置关系, 给出了约束协作算法的执行条件, 然后结合升级的锚节点再进行新一轮的定位运算。该方法计算简单、通信量小, 与已有三维定位算法相比, 在测距存在一定误差的情况下, 依然可以达到很好的精度。

关键词: 无线传感器网络; 三维定位; 迭代; 测距

中图分类号: TP393

文献标识码: A

文章编号: 1004-1699(2012)05-0644-07

无线传感器网络 (WSN, Wireless Sensor Network) 是带有无线收发装置的传感器节点以自组网形式构建的无线网络^[1]。该技术发展迅速, 广泛应用于环境、军事、安全、工业及家居等领域。实际使用中, 传感器节点在布设时一般采用随机投放的方式, 大多数节点的位置不能事先确定^[2]。而节点的位置信息对于目标追踪、环境监测、基于位置信息的路由等是不可或缺的。与此同时, 了解节点的位置信息还可以提高网络的路由效率、覆盖质量, 实现网络的负载均衡

和拓扑自配置^[3]。因此, 节点定位技术作为无线传感器网络研究中的核心技术之一显得尤为重要。

为传感器节点提供 GPS 模块是最直接最精确的定位方法, 但是附带 GPS 模块会使节点的成本增加两个数量级以上^[4], 所以 GPS 模块通常只用在锚节点 (Anchor Node) 上, 锚节点即可以获得绝对位置信息的节点。定位算法即采用一定的逻辑运算方法, 利用锚节点位置信息得到未知节点的位置。

目前无线传感器网络节点的二维平面定位算法

项目来源: 国家自然科学基金项目 (61134007); 中央高校基本科研业务费专项资金项目; 上海市教育委员会科研创新基金资助项目 (11YZ262); 上海高校选拔培养优秀青年教师科研专项基金资助项目 (Sxy09007)

收稿日期: 2011-12-15 **修改日期:** 2012-04-16

已经趋于成熟。当传感器节点投放在水下、空中、山区等复杂环境中时,受到地理空间因素限制,节点呈现三维空间中的立体分布,因此对节点三维定位的研究非常必要^[5]。但是由于三维空间运算更为复杂,相关三维定位技术^[6-7]的研究还有待深入,存在定位精度不高,计算量大,节点传输数据量大等不足。其中,直接使用大量锚节点位置信息对所有的未知节点进行定位可以得到较高的定位精度,但是成本过高;而减少锚节点的数量,往往需要大幅增加算法的运算量和节点的通讯量。

本文针对无线传感器网络在三维空间上的定位问题,将经典的 AHLos (Ad-Hoc Localization System)^[8] 二维定位算法扩展到三维,并进行改进,提出了一种基于测距的分布式定位算法——高精度迭代三维定位算法 ILAH-3D。该算法具有以下特点:(1)基于测距,使用锚节点信息对未知节点进行定位,精度高。(2)利用加权思想对最小二乘法进行改进,削减测距误差带来的影响。(3)节点计算方法复杂度低,降低了算法的运算量,采用分布式运算,降低了节点的通讯量。(4)采用协作和迭代的方式提高锚节点的利用率,在保证精度的同时降低对初始锚节点数量的需求。

1 AHLos 定位算法

AHLos 算法是一种基于测距的二维定位算法,它是由加州大学洛杉矶分校的 Andreas Savvides 等人提出的。在 AHLos 算法中,锚节点的位置是已知的,利用测距方法测量每个节点与其邻居节点间的距离。根据每个待定位节点的邻居锚节点的分布情况,利用相应的子算法计算坐标。该算法的核心是迭代思想,将已定位的节点“升级”为锚节点投入新一轮的运算。

AHLos 定义了3种定位子算法——原子多边算法 AM (Atomic Multilateration)、协作多边算法 CM (Collaboration Multilateration) 和迭代多边算法 IM (Iterative Multilateration)。

原子多边算法就是最小二乘法。如图1(a)所示,当未知节点邻居节点中有3个及3个以上的锚节点时,可以直接通过最小二乘法计算出坐标。

协作多边算法是假如一个未知节点可获得足够的信息,组成一个具有多个方程式并拥有唯一解的超定(Over-Determined)或适定(Well-Determined)系统,那么就可以通过使用跨越多跳的一组节点(包括未知节点)来实现定位。在图1(b)所示的拓扑结构中,位置节点2和4都有三个邻居节点,其中1,3,5,6是锚节点,根据拓扑结构中的5个连接信息可以建立拥有4个未知数的5个二次方程,通过节

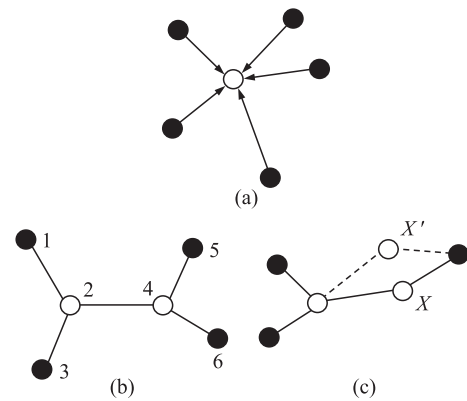


图1 节点连接关系图

点协作计算可以得出节点2和4的位置信息。

迭代多边算法是在使用 AM 和 CM 算法的基础上,将成功定位的节点升级为锚节点,进入下一轮循环,更新节点的邻节点信息,对符合条件的未知节点使用 AM 和 CM 算法。如此迭代循环,直至所有的节点都完成定位,或者没有满足 AM 和 CM 算法成立条件的节点存在时结束。

AHLos 算法将迭代思想引入节点定位,在获得锚节点带来的高计算精度的同时,从一定程度上缓解了锚节点密度较低时的节点定位问题。但将已定位的未知节点直接升级为锚节点可能会造成误差累计,降低整体的定位精度。其次,在 AHLos 算法中并没有给出严格的执行协作多边算法的充分条件,如图1(c)所示,虽然存在4个限定方程和4个未知数,但是如果执行算法,节点位置的解并不唯一。

2 ILAH-3D 算法

ILAH-3D 算法继承 AHLos 算法的思想,并进行改进和扩展。“三维原子算法”将原子多边算法扩展到三维空间,并引入加权最小二乘法,减小测距误差带来的影响;“约束协作算法”针对协作多边算法的缺陷,改进协作的方式,完善了算法执行条件和约束条件,并提出一种低复杂度的计算三维坐标方法,保证精度的同时降低了三维空间求解的运算量;“条件迭代算法”在迭代多边算法的基础上,加入了节点升级为锚节点的验证条件,一定程度上消减迭代误差的来源。

2.1 三维定位问题描述

假设一组传感器节点 $V = \{v_i | i=1, \dots, N\}$, 其中位置坐标 $X_A = \{x_1, y_1, z_1, \dots, x_m, y_m, z_m\}$ 已知的锚节点 m 个,其余节点是位置待求的未知节点。假设所有节点均匀分布在一个三维空间 R^3 的第一象限内,坐标为正实数,已经完成了自组网工作。节点的信号传输模型为理想球体,半径为 R 。即节点间距离小于等于 R 时,节点连通,为邻居节点。测距模块

的测量范围大于 R , 即邻居节点之间均完成测距。本算法通过节点连通信息, 距离信息, 锚节点坐标信息求出未知节点的位置坐标。

2.2 节点测距及分类

完成自组网工作, 并在组网过程中将所有锚节点的位置信息传输到每个可以加入网络连接的节点。每个节点与其通讯范围内所有邻居节点通过测距模块完成测距工作, 获得距离信息。将未知节点进行分类, 邻居节点中拥有不少于 4 个锚节点的执行“三维原子算法”, 拥有 3 个锚节点的执行“约束协作算法”。

2.3 三维原子算法

三维原子算法使用最小二乘法来计算节点坐标。因为该方法在邻居锚节点大于 4 个的情况下, 可以充分利用多余的锚节点信息提供更高的精度; 其次, 当测距在一定误差范围内时, 依然可以达到较高的精度, 执行三维原子算法是未知节点升级为新锚节点的最好选择。

2.3.1 最小二乘法

在三维空间中, 对于邻居节点中拥有不少于 4 个锚节点的未知节点, 可以直接使用最小二乘法来计算其坐标位置信息。

假设节点 $A(x_a, y_a, z_a)$ 的邻居节点中有 n 个锚节点, 坐标为 $(x_1, y_1, z_1), \dots, (x_n, y_n, z_n)$, 它们到节点 A 的距离分别为 d_1, d_2, \dots, d_n 。则有公式:

$$\begin{cases} (x_1 - x_a)^2 + (y_1 - y_a)^2 + (z_1 - z_a)^2 = d_1^2 \\ \vdots \\ (x_n - x_a)^2 + (y_n - y_a)^2 + (z_n - z_a)^2 = d_n^2 \end{cases} \quad (1)$$

每个方程分别减去最后一个方程, 得到一组线性方程, 可表示为: $\mathbf{A}\mathbf{X} = \mathbf{b}$

$$\mathbf{A} = \begin{bmatrix} 2(x_1 - x_n) & 2(y_1 - y_n) & 2(z_1 - z_n) \\ \vdots & \vdots & \vdots \\ 2(x_{n-1} - x_n) & 2(y_{n-1} - y_n) & 2(z_{n-1} - z_n) \end{bmatrix} \quad (2)$$

$$\mathbf{b} = \begin{bmatrix} x_1^2 - x_n^2 + y_1^2 - y_n^2 + z_1^2 - z_n^2 - d_1^2 + d_n^2 \\ \vdots \\ x_{n-1}^2 - x_n^2 + y_{n-1}^2 - y_n^2 + z_{n-1}^2 - z_n^2 - d_{n-1}^2 + d_n^2 \end{bmatrix} \quad (3)$$

通过最小二乘估计法, 可得到节点 A 的坐标为

$$\mathbf{X} = (\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T \mathbf{b}, \mathbf{X} = \begin{bmatrix} x_a \\ y_a \\ z_a \end{bmatrix} \quad (4)$$

在测距存在误差的情况下, 虽然最小二乘法可以一定程度上减弱误差, 但是由于普通最小二乘法的特性, 不再是无偏估计。另外, 根据文献[9]的证明, 上述方法求解的精确性受制于最后一个被减方程中的误差, 如果最后一个方程误差较大, 那最后结

果就会受到很大影响。

2.3.2 加权最小二乘法

为了得到更加精准的结果, 抑制误差, 对上述最小二乘法进行改进。本文假定, 在测量手段相同的情形下, 测距误差会随着节点间距离的增加而增大, 且增长呈现非线性特性。

首先, 将测距误差最小的方程作为被减方程, 可以最大程度减小对计算的影响。根据假设可知, 测量距离最短即测距误差最小。

其次, 采用加权最小二乘法进行改进。根据每段距离的测距误差, 在计算中加入不同的权值进行结果的校正, 通过加权尽可能使结果实现无偏估计, 提高定位精度, 计算公式如下:

$$\mathbf{X} = (\mathbf{A}^T \boldsymbol{\omega} \mathbf{A})^{-1} \mathbf{A}^T \boldsymbol{\omega} \mathbf{b}$$

加权矩阵 $\boldsymbol{\omega}$ 应为正定矩阵。由许瓦兹不等式可以证明, $\boldsymbol{\omega} = \mathbf{P}^{-1}$ 时 \mathbf{X} 的估计均方差最小, 其中 \mathbf{P} 为测距误差的方差矩阵。但实际中, 测距误差不容易求得, 因此我们根据不同的测距误差模型设计相应的加权函数。本文定义一个对角加权矩阵 $\boldsymbol{\omega}$, 其对角线上的元素 $\omega_i = 1 - d_i/R$, 其中 R 为节点通信半径, 距离越近误差越小, 权重越高。如果测距误差的模型改变, 只需将加权矩阵 $\boldsymbol{\omega}$ 改成相适应的函数, 或者根据实际情况指定相应的权值即可。

2.4 约束协作算法

AHLos 的协作多边算法是利用一组关联节点的连接信息建立方程组。但是在三维空间中, 节点的关联情况更多, 连接信息更为复杂, 而且也可能出现解不唯一的情况, 因此直接根据连接信息列方程组求解会使计算非常复杂。约束协作算法改进了二维协作多边算法, 以适应三维定位。

在三维空间中, 直接对邻居节点中有 3 个锚节点的未知节点进行求解, 可以求得两组解。如图 2 所示未知点 $A(x_a, y_a, z_a)$ 与锚节点 $B(x_b, y_b, z_b)$, $C(x_c, y_c, z_c)$, $D(x_d, y_d, z_d)$ 距离确定的情况下, 可以求出 A, A' 两个位置都满足已知条件, A, A' 关于平面

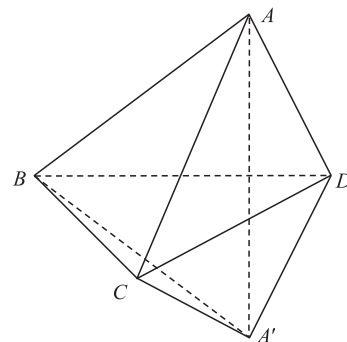


图2 未知节点 A 的邻居节点中有 3 个锚节点

BCD 对称。根据这个性质,以这种节点为局部核心节点,先求出 2 个待定结果,然后利用周围节点的情况协作进行约束就可以得到该节点正确的位置坐标。在求解中,由于已知方程数少,使用上述加权最小乘法不能计算得到相应的两组解,而直接对复杂的非线性方程组进行求解运算量较大。这里提出一种新的基于向量的求解方法。

2.4.1 向量求解算法

根据节点间的距离可以得到方程组(5):

$$\begin{cases} \cos \angle CBA = \frac{|\vec{BA}|^2 + |\vec{BC}|^2 - |\vec{CA}|^2}{2|\vec{BA}||\vec{BC}|} \\ \cos \angle BDA = \frac{|\vec{DB}|^2 + |\vec{DA}|^2 - |\vec{BA}|^2}{2|\vec{DB}||\vec{DA}|} \\ \cos \angle DAC = \frac{|\vec{AD}|^2 + |\vec{AC}|^2 - |\vec{CD}|^2}{2|\vec{AD}||\vec{AC}|} \end{cases} \quad (5)$$

其中向量 $\vec{BC} = (x_c - x_b, y_c - y_b, z_c - z_b)$, 同理可得其它向量。所有向量的模已知,即为节点间测得的距离。由向量的性质可得方程组(6):

$$\begin{cases} \cos \angle CBA = \frac{\vec{BC} \cdot \vec{BA}}{|\vec{BC}||\vec{BA}|} \\ \cos \angle BDA = \frac{\vec{DB} \cdot \vec{DA}}{|\vec{DB}||\vec{DA}|} \\ \cos \angle DAC = \frac{\vec{AD} \cdot \vec{AC}}{|\vec{AD}||\vec{AC}|} \end{cases} \quad (6)$$

联合方程组(5),(6)可得方程组(7),求解可得关于点 $A(x_a, y_a, z_a)$ 的两组解。

$$\begin{cases} (x_c - x_b)(x_a - x_b) + (y_c - y_b)(y_a - y_b) + (z_c - z_b)(z_a - z_b) = \cos \angle CBA (|\vec{BC}||\vec{BA}|) \\ (x_b - x_d)(x_a - x_d) + (y_b - y_d)(y_a - y_d) + (z_b - z_d)(z_a - z_d) = \cos \angle BDA (|\vec{DB}||\vec{DA}|) \\ (x_d - x_a)(x_c - x_a) + (y_d - y_a)(y_c - y_a) + (z_d - z_a)(z_c - z_a) = \cos \angle DAC (|\vec{AD}||\vec{AC}|) \end{cases} \quad (7)$$

方程组(7)中的第一个和第二个方程为线性方程,第三个方程为二次方程,求解该方程组运算简单,计算量小,同时运算精度有一定保证。该计算方法误差大小主要取决于测距误差,在没有测距误差的情况下可以准确地得出两个对称坐标,再通过约束协作算法可以对这两组解进行判别,选出符合实际情况的解。

2.4.2 核心节点分类

将拥有 3 个邻居锚节点的未知节点称为核心节点,核心节点的邻居节点中的其他未知节点称为辅

助节点。根据辅助节点的状态对核心节点进行分类,没有辅助节点的核心节点为 3-0 节点;辅助节点拥有 1 个和 2 个邻居锚节点的核心节点分别称为 3-1 节点和 3-2 节点;辅助节点同样拥有 3 个锚节点的,则这两个节点互为 3-3 节点。

对于 3-1,3-2,3-3 节点,核心节点和辅助节点可能公用锚节点,即辅助节点的邻居锚节点同样也是核心节点的锚节点,因此其连接关系存在多种情况。如图 3 所示,3-1 节点存在 2 种情况,a 图中没有公共锚节点,称为 3-1-a 节点,b 图中存在公共锚节点,称为 3-1-b 节点。如图 4 所示,3-2 节点存在 3 种情况,a 图中没有公共锚节点,称为 3-2-a 节点,b 图中存在一个公共锚节点,称为 3-2-b 节点,c 图中存在两个公共锚节点,称为 3-2-c 节点。3-3 节点的情况较为类似,不再一一列出。

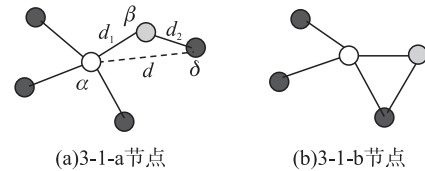


图 3 3-1 节点示意图

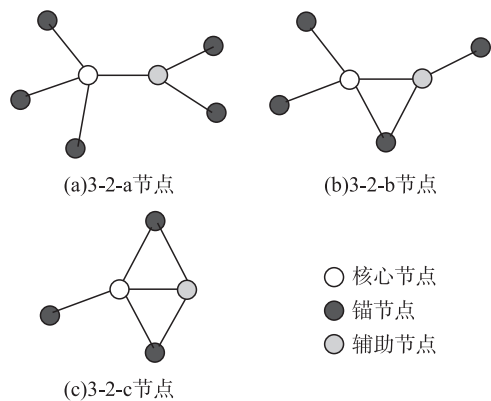


图 4 3-2 节点示意图

○ 核心节点
● 锚节点
● 辅助节点

2.4.3 约束协作算法

假设在组网阶段,核心节点已经获得所有可以与该节点多跳连接的锚节点坐标信息。对核心节点使用向量求解方法得到两个解。

约束 1 假设整个网络空间为 D^3 , D 为实数域上的某一区间,每个节点坐标都在 D^3 之内。考虑定位节点测距误差区间为 $[0, \varphi R]$, φ 为最大测距误差与节点通讯半径 R 的比例。那么将空间 D^3 每个坐标方向双向扩展 φR 为 D'^3 ,计算出的节点坐标在 D'^3 内才符合实际情况,否则与实际不符,排除该解。约束 1 适合所有的核心节点。

约束 2 将核心节点的两个解与其所有的非邻居锚节点通过距离公式(8)求得距离 d ,如果出现 d 小于 R 的情况,即该非邻居锚节点与核心节点在一

跳范围之内,与实际不符,则可以排除该解。

$$d = ((x_1 - x_2)^2 + (y_1 - y_2)^2 + (z_1 - z_2)^2)^{1/2} \quad (8)$$

约束 2 适合于所有的核心节点。

约束 3 假设节点的通讯半径为 R ,未知节点 β 的邻节点中有未知节点 α 和锚节点 δ ,并且 α 和 δ 不是邻节点,如图 3(a)。 β 和 α 之间的距离为 d_1 , β 和 δ 之间的距离为 d_2 ,则 α 与 δ 的距离 d 小于 $d_1 + d_2$,且大于 R ,小于 $2R$ 。

符合上述 α 点情形的节点,可以通过距离公式(8)分别计算两个解与锚节点 δ 的距离 d ,如果 d 大于 $d_1 + d_2$ 或者小于 R ,则该解被排除。

3-1-a,3-2-a,3-2-b 节点及大多数 3-3 节点都可以使用约束 3 进行判定。

约束 4 对于 3-2 节点,核心节点求出两个解后,暂时升级为锚节点,分别代入辅助节点进行计算,可以求出对应的 4 个从属解。辅助节点肯定满足约束 2 和约束 3 的条件,如果将核心节点一个解对应的一组两个从属解都排除,则该解被排除。约束 4 适用于 3-2 节点。

约束 4-1 对辅助节点执行约束 2。

约束 4-2 对辅助节点执行约束 3。

约束 5 符合 3-3 节点条件的一对核心节点都可以使用向量求解方法求出各自的两个解。两个节点间的距离 d' 是测量已知的,因此直接将四个解代入距离公式(8)求得 d ,进行验证 d 与 d' 是否相等,可以得出正确的一组解。

因为测距可能存在误差,所以不能简单的判定 $d = d'$ 。引入一个阈值 ε_1 ,验证 $|d' - d| < \varepsilon_1$ 成立时为正确的解。其中 ε_1 根据测距设备精度以及定位精度要求进行选取。

约束 5 适合于所有的 3-3 节点。

2.4.4 核心节点约束算法的选择

5 种约束算法的计算量不同,约束 1 计算量 V_1 最小,约束 3 和约束 5 的计算量 V_3 、 V_5 基本在同一级别,小于约束 2 计算量 V_2 。而约束 4 基于约束 2、3,其计算量 V_{4-1} 、 V_{4-2} 取决于具体情况。总体来讲, $V_{4-1} > V_2 > V_{4-2} > V_3 > V_5 > V_1$,由于这种计算量上的差别,对于不同的核心节点,优化执行约束算法的顺序,可以降低整体的计算量和计算复杂度。

约束 1 适合所有的核心节点,不需要额外信息,而且计算量最小,因此对所有的节点首先执行约束 1。如果约束不能得到结果,则按照下表中的步骤对不同类型的节点依次执行不同的算法。节点如果满足第一个约束的执行条件则直接执行,不满足约束的执行条件或者该约束失效,则跳转到下一个约束

继续考察,直至得到实际解。如果所有约束对该节点都失效,则放弃节点在本轮运算中的求解。

表 1 不同节点约束条件执行流程

3-0 节点	3-1 节点	3-2 节点	3-3 节点
约束 1	约束 1	约束 1	约束 1
约束 2	约束 3	约束 3	约束 5
	约束 2	约束 4-2	约束 3
		约束 2	约束 2
		约束 4-1	

其中,对 3-3 节点执行约束 5,一般都可以得到结果。在极为特殊的情况下,会存在两组解都可以满足 $d = d'$,不能确定结果,则转而根据情况执行约束 3 和约束 2。

结合上述各种节点的执行步骤,一个核心节点如果同时拥有多个辅助节点,符合多种节点情况时,算法执行的顺序为①约束 1②约束 5③约束 3④约束 4-2⑤约束 2⑥约束 4-1。节点满足约束执行条件则执行,不满足条件,则考察下一个约束,直到计算出结果或者全部执行完毕无法确定结果。

2.5 条件迭代算法

条件迭代算法是通过三维原子算法和约束协作算法求解后,将符合精度要求的被定位的未知节点升级为锚节点,重新执行三维原子算法和协作算法,进入迭代运算,直到所有的点都被定位,或者剩下所有的未知节点都已不符合进行定位运算的条件为止。

虽然三维原子算法和条件迭代算法已经对误差进行了抑制,但是由于测距误差存在不确定性,不能保证计算结果一定达到某种精度,并且迭代误差也不可避免。因此我们采取一定的措施来尽可能减少迭代误差,在被定位的节点升级为锚节点前,验证其是否符合精度要求。

条件 1 将求得的节点坐标与相邻锚节点坐标通过距离公式(8)计算距离为 d , d' 为测得的距离。设定一个精度阈值 ε_2 ,如果 $|d' - d| > \varepsilon_2$,则该节点状态设为待定,不升级为锚节点,待迭代之后,邻节点状态更新,进行重新计算。其中 ε_2 根据测距设备精度以及定位精度要求进行选取。

条件 2 计算定位节点的升级权值:将与定位节点相邻锚节点的各自权值分别乘以定位节点与其对应的锚节点的测距误差权值 $\omega_i = 1 - d_i / R^2$ 之后相加取平均值,即为定位节点的升级权值。如果升级权值小于一定的阈值 ε_3 ,则说明测距误差较大,该节点暂不升级为锚节点。其中 ε_3 根据测距设备精度以及定位精度要求进行选取。该条件可以有效抑制误差在节点定位过程中的传播。

满足以上两个条件的节点可以升级为锚节点,进入下一轮的计算。此时该锚节点的权值为升级权值,而网络中原有的锚节点权值设为 1。

3 分析与仿真

为了检验 ILAH-3D 算法的性能,本文使用 Matlab2010b 软件进行仿真实验。与经典的 MDS-MAP (P)算法^[10]、Landscape-3D 算法^[11]、Constrained 3-D 算法^[12]进行比较。MDS-MAP(P)是一种分布式多维定位算法,根据节点间的连接关系进行定位,对锚节点需求小,在测距和非测距环境下都可以进行计算,在测距环境下精度较高。Landscape-3D 和 Constrained 3-D 都是基于测距的算法,精度较高。前者采用“移动”锚节点进行定位,未知节点间无需通信,后者与本文相似,利用一般锚节点信息进行定位。

仿真实验区域为 100 m×100 m×100 m 的三维空间,按均匀分布随机部署 200 个传感器节点,传感器节点的传输半径为 R 。初始参数设定为锚节点比例 10%,平均节点连接度(节点的平均邻节点数量)为 15.52,测距误差为随机误差,范围为 $[0,0.1]R$,分布符合随节点间距离 d 的增大非线性增大,所有误差均表示为误差数值与 R 的百分比。在上述初始参数的基础上,根据实验需要调整锚节点比例、测距误差、平均节点连接度(通过改变 R 来调整),来观察算法性能的变化。

3.1 定位节点比例

图 5 显示了在不同的锚节点密度和节点平均连接度的环境下,ILAH-3D 算法成功定位的节点占所有未知节点的比例。在锚节点比例 5%,平均连接度 8 以上可以取得接近或者达到 100% 定位的结果。

分析实验结果,算法定位结果受到锚节点密度和节点平均连接度的影响,两个参数越大,定位节点数量越多。原因是 ILAH-3D 算法是一种基于节点

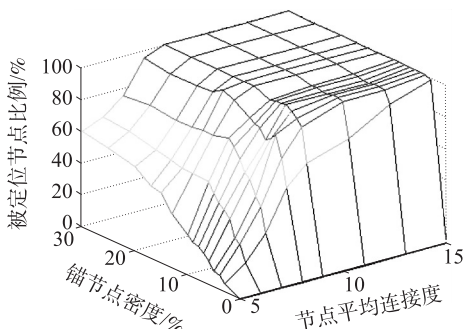


图 5 锚节点密度和节点平均连接度对定位节点比例的影响

数据的计算方法,虽然使用协作和迭代算法,对锚节点的需求已经大大降低,不过执行三维原子算法和约束协作算法还是要依赖于一定的锚节点,初始锚节点数量需要超过一定的阈值。而节点连接度较低时,节点的邻节点过少,不满足算法的要求。提高节点平均连接度可以使更多节点与锚节点连接,相当于间接提高了锚节点数量,因此算法对锚节点数量的依赖可以通过提高节点平均连接度来缓解。总的来说,算法对锚节点的依赖性适中,初始锚节点数量超过一定的阈值即可,而对节点连接度的要求较低。

3.2 加权最小二乘法和迭代误差

在三维原子算法中,采用加权最小二乘法代替了最小二乘法。如图 6 所示,此方法在误差模型符合估计时效果好于一般的最小二乘法。

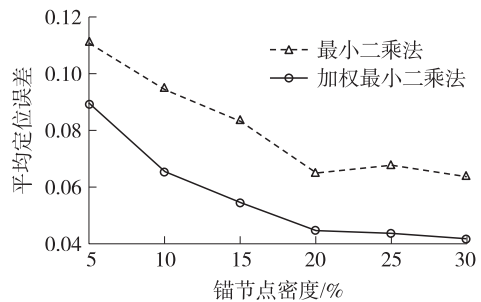


图 6 最小二乘法和加权最小二乘法对比

作为一种迭代算法,迭代中累积误差的大小非常重要。图 7 显示了在不同锚节点密度的环境下,算法每一代的计算误差。从实验结果可以看出,ILAH-3D 有一定的迭代误差,并受到锚节点密度的影响。锚节点密度越小则迭代代数越多,累积误差越大,这与算法原理是相符的。

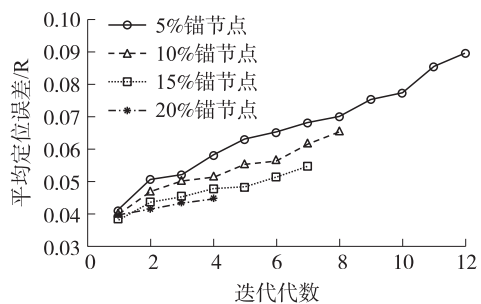


图 7 误差随迭代代数的变化

3.3 定位精度

图 8 显示了在不同锚节点数量下算法的定位精度。本文提出的 ILAH-3D 算法定位精度随着锚节点密度的增大而提升,在锚节点比例达到 10% 以上时,定位精度优于其余 3 种算法,且锚节点密度越高,这种优势越明显。

图 9 所示为在不同的节点平均连接度下算法的

定位精度。可以看出,由于提高节点连接度相当于间接提高了锚节点密度,因此 ILAH-3D 算法定位精度随节点连接度的增大而提升,整体来看平均误差低于其他3种算法,达到了较高精度,这也与图8的结论相一致。

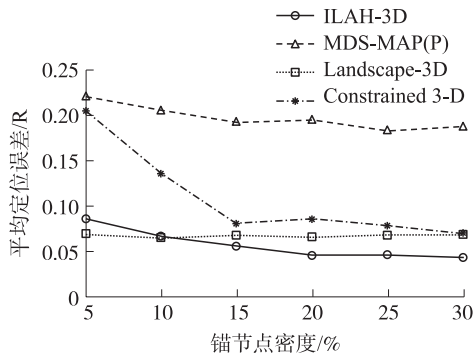


图8 锚节点密度对定位精度的影响

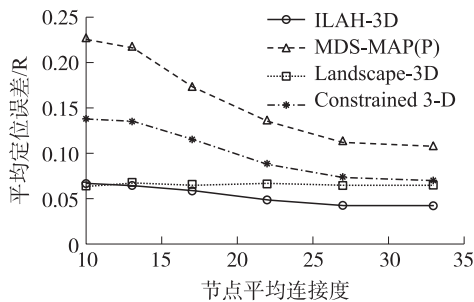


图9 节点平均连接度对定位精度的影响

图10给出了不同的测距误差环境下算法的定位精度。4种算法的精度都会受到测距误差的影响。ILAH-3D算法由于其计算依赖测距的特点,在测距误差增大到一定程度后,定位误差急速增大,不过依然能够保持精度与别的算法在同一级别。

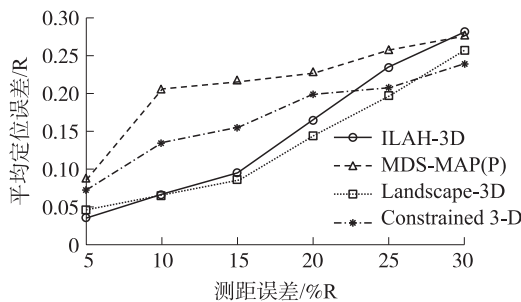


图10 测距误差对定位精度的影响

4 结论

本文针对无线传感器网络在三维空间内的定位问题,继承和扩展经典二维定位算法 AHLos 的迭代和节点协作思想,提出了一种新的基于测距的分布式 ILAH-3D 算法。从实验仿真可以看出,ILAH-3D 降低了对锚节点密度的依赖性,对节点连接度要求较低,定位精度高,并且对测距误差有一定的抵抗能力。进一步提高锚节点的利用效率,设法继续减小测距误差对算法的影响是下一步研究的重点。

参考文献:

- [1] Schurgers C, Kulkarni G, Srivastava M B. Distributed on-Demand Address Assignment in Wireless Sensor Networks [J]. IEEE Transactions on Parallel and Distributed Systems, 2002, 13(10): 1056-1056.
- [2] Zhao F, Shin J, Reich J. Information-Driven Dynamic Sensor Collaboration [J]. IEEE Signal Processing Magazine, 2002, 19(2): 68-77.
- [3] Mao G, Fidan B, Anderson B. Wireless Sensor Network Localization Techniques [J]. Computer Networks, 2007, 51(10): 2529-2553.
- [4] Heidemann B N J, Estrin D. GPS-Less Low Cost Outdoor Localization for Very Small Devices [J]. IEEE Personal Communications, 2000, 7(5): 28-34.
- [5] Ou C, Su K F. Sensor Position Determination with Flying Anchors in Three-Dimensional Wireless Sensor Networks [J]. IEEE Transactions on Mobile Computing, 2008, 7(9): 1084-1097.
- [6] 毛科技,赵小敏,邵奔,等.无线传感器网络中基于共度面的三维定位算法研究与设计[J].传感技术学报,2011,24(10):1484-1488.
- [7] 李辉,熊盛武,刘毅,等.无线传感器网络 DV-HOP 定位算法的改进[J].传感技术学报,2011,24(12):1782-1786.
- [8] Savvides A, Han C C, Srivastava M B. Dynamic Fine-Grained Localization in Ad-Hoc Networks of Sensors [C]//Proceedings of the 5th Annual International Conference on Mobile Computing and Networking, Rome: ACM Press. 2001:166-179.
- [9] 吕睿,阳宪惠.减少无线传感器网络节点定位误差的方法[J].清华大学学报(自然科学版),2008,48(10):1839-1843.
- [10] Shang Y, Ruml W, Zhang Y. Improved MDS-Based Localization [C]//Proceedings of IEEE Infocom 2004, Hong Kong. 2004:2640-2651.
- [11] Zhang L Q, Zhou X B, Chen Q. Landscape-3D: A Robust Localization Scheme for Sensor Networks over Complex 3D Terrains [C]//Proceeding of 31st IEEE Conference on Local Computer Networks, Tampa, 2006:239-246.
- [12] Liang J L, J Shao J, Xu Y, et al. Sensor Network Localization in Constrained 3-D Space [C]//Proceeding of IEEE International Conference on Mechatronics and Automation, Luoyang, 2006:19-54.



祁荣宾(1974-)女,汉族,华东理工大学信息科学与工程学院副研究员、硕导,主要从事智能计算、工业过程控制与优化、无线传感器网络的研究, qirongbin@ecust.edu.cn;



李思瑾(1987-)男,汉族,华东理工大学信息科学与工程学院硕士研究生,主要从事无线传感器网络与智能算法的研究, happyblsj@gmail.com.