

基于局部网络拓扑特征的鲁棒节点定位算法^①

赵 方^②* ** 马 严* 罗海勇*** 林 权**** 宋茂强*

(* 北京邮电大学软件学院 北京 100876)

(** 北京邮电大学网络技术研究院 北京 100876)

(*** 中国科学院计算技术研究所 北京 100190)

(**** 北京航空航天大学 北京 100083)

摘 要 针对无线传感器网络的较大测距误差严重影响定位算法精度和鲁棒性的问题,利用节点均匀部署网络的拓扑特征,提出了一种基于局部网络拓扑特征的鲁棒节点定位算法(LFLS 算法)。该算法通过构建节点测距高估粗差阈值参数和测距低估粗差阈值参数,在对未知节点 1 跳测距数据集进行粗差识别及剔除等预处理滤波的基础上,使用高斯加权最小二乘定位算法实现节点定位。仿真结果表明,基于局部网络拓扑特征的鲁棒节点定位算法的定位精度明显优于未采用局部网络拓扑特征进行粗差预处理的加权最小二乘定位算法,其中粗差测距直接相关节点的定位精度改进尤为明显。

关键词 无线传感器网络,节点定位,加权最小二乘,局部网络拓扑特征

0 引 言

受成本、能量限制以及环境干扰等因素影响,无线传感器网络测距数据中不仅包含数量较多、幅度较小的测距误差,而且还包含少量幅度较大的测距误差(即粗差)^[1-3]。当没有测距粗差时,加权最小二乘定位算法具有较高的定位精度,表现出对较小测距误差的良好抑制效果,但当数据中混有少量粗差时,较大方差的粗差数据在最小二乘平方和中的取值偏大,最小二乘估计被拉向方差大的项,导致方差大的项拟合程度提高,而方差小的项拟合程度降低,定位算法性能明显下降^[4,5],尤其是粗差测距边直接相关节点的定位误差明显增大,表现出较差的鲁棒性。为提高节点定位算法鲁棒性和定位精度,需要研究测距粗差探测和识别机制,抑制测距粗差对定位算法的不利影响。

1 相关研究

无线传感器网络节点定位目前有多种测距误差处理方法。一种是采用多次测量求平均方法^[6]消除因环境干扰等偶然因素产生的测距粗差,不过该方法对因节点硬件原因或障碍物阻挡而产生的测距粗差的处理并不有效。另一种方法是利用网络相邻节

点欧式空间的几何约束关系(如三边形边长之间的约束条件)^[6]识别和剔除因系统原因产生的测距粗差。文献[7]综合考虑节点自身定位精度与测距误差,引入节点相对定位误差概念,并基于相对定位误差自适应选择综合性能较好的邻居节点参与定位求精运算,减小较大定位误差的扩散,从而提高定位精度。Costa 等^[8]则采用二步邻居选择法消除噪声环境下接收信号强度指示(RSSI)模型的测距低估效应(2 跳邻居测距估计因测距低估效应被误判为 1 跳邻居),让未知节点选择出真实 1 跳邻居节点进行定位求精,不过该误差处理机制难以有效处理 1 跳邻居范围内的测距粗差。

不同于上述误差处理方法,本文利用节点部署的网络拓扑特征,针对均匀部署网络,提出了基于局部网络拓扑特征的粗差识别算法,在识别和剔除测距粗差后,使用高斯加权最小二乘法进行节点定位,不仅整体上提高了节点定位精度,而且大幅提高了测距粗差相关节点的定位精度,增强了定位算法鲁棒性。

2 基于局部网络拓扑特征的粗差识别

2.1 均匀网络部署条件下的测距粗差

假设网络节点均匀部署,局部网络节点密度不

① 863 计划(2006AA10Z253, 2007AA12Z321)和国家自然科学基金(60772111, 60873244)资助项目。

② 女,1968 年生,博士生,讲师;研究方向:计算机网络,分布式计算;联系人, E-mail: zhaofanghy@yahoo.com.cn
(收稿日期:2008-04-10)

发生突变,即节点 i 的 1 跳邻居局部拓扑结构与以其邻居节点 j 为中心的 1 跳邻居局部拓扑结构基本一致,局部节点密度连续变化。从网络全局看,该假设认为,网络中任意两个节点间均存在通路,整个网络拓扑连通,不存在局部节点密度突然变大或变小情况,该假设与很多传感器网络实际应用情况基本相符。

节点均匀部署情况下,假设每个节点的局部拓扑如图 1 所示,其中黑点表示节点 i 的邻居节点,线表示节点 i 与其直接邻居间的无线链路。受环境干扰与节点硬件精度等因素影响,相邻节点间测距存在粗差^[1-3]。假设节点 i 的实际局部网络拓扑如图 1 所示,因环境干扰等因素,导致节点 i 所感知的局部网络拓扑位置与节点的实际位置相差较大,图 2 显示了三种形式的粗差。一种是邻居节点 A 虽然位于节点 i 的通信半径之内,但由于多径效应或障碍物阻挡,节点 i 并未感知到 A ,即节点 i 判定节点 A 不是自己的 1 跳邻居。第二种是节点 i 到邻居节点 B 的估计距离比实际距离小很多,这类粗差称为测距过度低估。第三种是节点 i 到邻居节点 C 的估计距离比实际距离大很多,即测距过度高估。

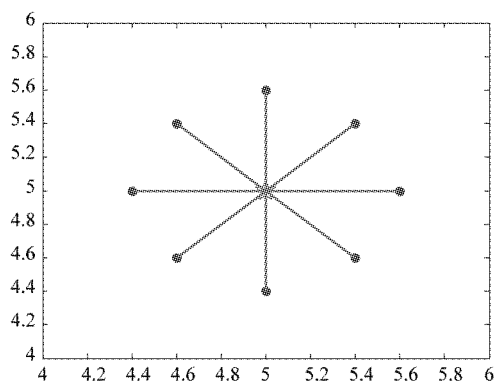


图 1 节点均匀部署条件下的实际局部网络拓扑

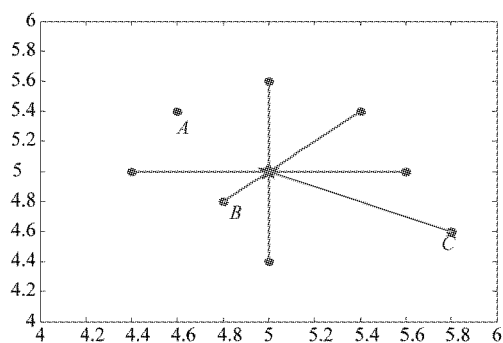


图 2 存在粗差情况下节点所感知的局部网络拓扑

尽管测距粗差在整个网络测距数据中所占比例不大,但给节点定位带来较大不利影响,导致部分测距粗差相关节点出现较大定位偏差。为提高网络所有节点的定位精度,避免部分节点定位误差过大而失去使用意义,有必要剔除网络中的测距粗差,防止粗差在网络内的扩散,提高节点定位算法鲁棒性和定位精度。

上述三种测距粗差,除了第一类粗差因未知节点不能感知到邻居节点的存在而难以处理外,对于测距过度高估和测距过度低估这两类粗差,可采用 2.3 节介绍的局部网络拓扑特征进行识别和剔除。

2.2 局部网络拓扑特征

本文把节点局部网络拓扑特征定义为节点到直接邻居的平均测距(本文称之为节点虚拟 1 跳距离)及标准差,如式(1)和(2)所示。其中 $\hat{d}_{ij}$ 为节点 i 与 1 跳邻居 j 的测距, N_i 为节点 i 的一跳邻居集合, $|N_i|$ 为节点 i 的一跳邻居数目。

$$Avg_i = \frac{\sum_{j \in N_i} \hat{d}_{ij}}{|N_i|} \quad (1)$$

$$Std_i = \left(\frac{\sum_{j \in N_i} (\hat{d}_{ij} - Avg_i)^2}{(|N_i| - 1)} \right)^{1/2} \quad (2)$$

在节点均匀部署网络,局部网络拓扑特征反映了在节点 1 跳局部范围内,节点到直接邻居测距的统计特性。通过适当设置测距粗差阈值判定范围,把那些远远偏离节点虚拟 1 跳距离的粗差予以剔除,为定位算法提供更准确的测距数据集。

本文定义了测距高估粗差阈值参数 ($ThrUpper$) 和测距低估粗差阈值参数 ($ThrLower$),分别用于判定测距过度高估和过度低估,其定义如式(3)所示,其中 N_i 为节点 i 的一跳邻居集合, α 为正值常量,用于控制粗差阈值大小,可根据测距技术及应用需求进行选择(在本文第 4 节仿真实验中 $\alpha = 1.85$)。

$$ThrUpper_i = \frac{1}{|N_i|} \sum_{j \in N_i} Avg_j + \alpha \times \frac{1}{|N_i|} \sum_{j \in N_i} Std_j \quad (3)$$

$$ThrLower_i = \frac{1}{|N_i|} \sum_{j \in N_i} Avg_j - \alpha \times \frac{1}{|N_i|} \sum_{j \in N_i} Std_j$$

如果节点 i 到直接邻居 j 的测距 $\hat{d}_{ij}$ 处于 $[ThrLower_i, ThrUpper_i]$ 区间内,则判定 $\hat{d}_{ij}$ 为正常测距(测距误差处于正常范围内),让 $\hat{d}_{ij}$ 参与节点 i 的定位求精运算;如果 $\hat{d}_{ij}$ 处于 $[ThrLower_i, ThrUpper_i]$ 区间外,则判定为粗差,予以剔除,不让节点 j 参与节点 i 的定位迭代求精运算,抑制其不利影响。

2.3 基于局部网络拓扑特征的粗差剔除

基于局部网络拓扑特征的粗差剔除步骤如下:

(1) 节点 i 估计到 1 跳邻居的所有测距, 然后使用公式(1)和公式(2), 提取节点 i 到直接邻居的局部网络拓扑特征 $LocalInfo_i = [Avg_i, Std_i]$ 。

(2) 节点 i 向一跳邻居发出局部网络拓扑信息请求, 获得直接邻居的局部网络拓扑特征信息列表:

$$NeighborLocalInfo_i = \left\{ \bigcup_{j \in N_i} LocalInfo_j \right\}$$

(3) 节点 i 基于上述直接邻居的局部网络拓扑特征信息列表 $NeighborLocalInfo_i$, 使用式(3)计算测距高估粗差阈值参数及测距低估粗差阈值参数, 并对自身局部范围内测距数据进行检测, 识别并剔除局部网络内存在的粗差测距边, 获得具有正常测距的 1 跳邻居集合。

(4) 节点 i 更新剔除粗差后的局部网络拓扑特征。节点 i 仅保留正常测距对应的直接邻居节点, 用它们构建有效 1 跳邻居集合 N'_i , 而将判定为粗差的测距边并入粗差测距集合 $OutlineSet = OutlineSet \cup [ID_i, ID_j]$, 其中 i, j 为粗差测距边对应节点, 节点 i 的有效 1 跳邻居集合 N'_i 为:

$$N'_i = \{ \forall j, j \in N_i \cap (i, j) \notin OutlineSet \} \quad (4)$$

节点 i 剔除粗差数据后, 使用式(5)及(6), 重新计算局部网络拓扑特征 $LocalInfo'_i = [Avg'_i, Std'_i]$, 并更新自己的局部网络拓扑信息。

$$Avg'_i = \frac{\sum_{j \in N'_i} \hat{d}_{ij}}{|N'_i|} \quad (5)$$

$$Std'_i = \left(\frac{\sum_{j \in N'_i} (d_{ij} - Avg'_i)^2}{(|N'_i| - 1)} \right)^{1/2} \quad (6)$$

上述局部网络拓扑特征剔除算法并不会给节点增加较大的计算和存储开销。每个节点在计算局部网络拓扑特征时, 仅涉及自己的 1 跳邻居集合, 拓扑特征计算量较小。节点需要存储的仅是 1 跳邻居的拓扑特征, 存储量开销也不大。

3 基于局部网络拓扑特征的鲁棒节点定位

基于局部网络拓扑特征的鲁棒节点定位算法 (Robust Node Localization Based on Local Network Parameter, 本文简称 LFLS) 主要包括以下步骤: (1) 每个节点使用局部网络拓扑特征识别和剔除粗差, 得到滤波处理后的测距数据集; (2) 每个节点使用剔除粗

差后的测距数据集构建初始局部相对坐标系; (3) 使用增量贪婪算法构建全局相对/绝对坐标; (4) 使用基于高斯加权的最小二乘法进行节点定位求精。

3.1 节点初始局部相对坐标系构建

本文使用计算和通信开销相对较小的基于假设坐标的定位算法 (Assumption Based Coordinates, ABC)^[9] 构建初始坐标, 其计算量相对较少。

ABC 算法利用简单几何关系构建节点相对坐标定位系。首先由局部区域中某个节点 n_0 首先发起, 假设其坐标为 $(0, 0)$ 。 n_0 根据第一个与它建立连接的节点 n_1 (在 n_0 的通信半径之内), 估计它们之间的距离 r_{01} , 并假设 n_1 在以 n_0 为原点的坐标系的 X 正半轴, 即 n_1 的坐标为 $(r_{01}, 0)$ 。第二个与 n_0 建立连接的节点 n_2 , 假设 n_2 在坐标系 Y 轴的正半轴区域内, 与 n_0 和 n_1 的距离分别为 r_{02} 和 r_{12} , 那么 n_2 的坐标 (x_2, y_2) 可根据公式(7)确定。

$$x_2 = \frac{r_{01}^2 + r_{02}^2 - r_{12}^2}{2r_{01}} \quad (7)$$

$$y_2 = \sqrt{r_{02}^2 - x_2^2}$$

使用 n_0 、 n_1 、 n_2 三个节点确定原点和 X 轴、Y 轴方向后, 其它任何位于 n_0 通信半径内的节点 n_i , 利用式(8)确定它在这个相对坐标系中的坐标。

$$x_i = \frac{r_{01}^2 + r_{0i}^2 - r_{1i}^2}{2r_{01}} \quad (8)$$

$$y_i = \frac{r_{0i}^2 - r_{2i}^2 + r_{02}^2 - 2x_2x_i}{2y_2}$$

每个节点使用这种方法, 建立自己的局部坐标系, 再采用 3.2 节介绍的全局地图融合算法, 构建全局相对坐标系, 最终利用信标节点信息转换成全局绝对坐标系。

3.2 全局地图融合

当每个节点建立起自己的局部相对坐标系后, 使用地图融合策略建立全局相对坐标系。

局部地图可采取串型或并行顺序进行融合, 其中串型方法可随机挑选或按照最适于应用的顺序进行融合。本文采用增量贪婪算法 (Incremental Greedy Algorithm)^[10] 来实现地图融合。该算法首先挑选 1 跳邻居数目较多的 1 个节点, 把它的局部地图作为核心地图, 然后每次选择与核心地图拥有最多相同节点的邻居地图依次进行融合, 直到核心地图覆盖整个网络。

两个局部地图融合的变换矩阵使用最佳线性变换 (Optimal Linear Transformation) 方法获得, 即使用最小二乘法来最小化 2 个地图相同节点的坐标变换误

差,来获得变换矩阵,把其中一个地图变换到另一个地图上,该变换包括平移、旋转、镜像以及缩放。由于局部地图并不是刚体,采用最佳线性变换方法,具有良好的鲁棒性。

设 $LN(p)$ 表示节点 p 的 1~2 跳邻居, $LM(p)$ 表示节点 p 的局部地图,即 $LN(p)$ 的节点坐标。首先随机选择节点 p ,并构建集合 $C = \{p\}$,表示已完成融合处理的节点。集合 $D = LN(p)$ 表示已定位的节点,集合 $M = LM(p)$ 表示当前全局地图,按以下 (1)~(3) 步骤迭代,直到集合 C 包含所有节点为止:

(1) 在集合 $D-C$ 中寻找节点 p' ,使得集合 D 与集合 $LN(p')$ 拥有最大的交集 I ,即搜索 $p' \in D-C$,使得 $I = \{LN(p') \subset D\}$ 最大。

(2) 使用最小二乘法求线性变换 T ,使得集合 I 的节点从坐标系 $M' = LM(p')$ 变换到全局坐标系 M 时,变换误差 $\sum_{k \in I} (M(k) - T[M'(k)])^2$ 最小。该变换包括偏移、正交旋转、镜像和缩放。实验结果显示,在没有缩放操作时,融合后的地图能更好地保留融合前的信息,因此 T 变换只取偏移、旋转和镜像操作。

(3) 融合 M 和 M' 。集合 $C = C \cup \{p'\}$, $D = D \cup M'$, $M = merge(M, M')$,更新 M 及相关数据。节点 p' 在融合后的地图 M 中的坐标更新方法为:

- 如 p' 仅包含在融合前的 M 中,不在 M' ,则使用融合前 M 中的坐标;
- 如 p' 仅包含在融合前的 M' ,不在 M ,则使用 $T(M')$ 中的坐标;
- 当 p' 同时在 M' 与融合前的 M 中,取融合前 M 中的坐标与 $T(M')$ 坐标的平均值。

当网络拥有足够多的信标节点时(2 维网络时最少 3 个,3 维网络时最少 4 个),使用所有信标节点在相对地图和绝对地图中的坐标(不考虑其它节点),求出最佳线性变换,就可计算出整个网络或者子网的绝对坐标。

获得全局初始坐标后,进入坐标迭代求精步骤。

3.3 未知节点定位求精

为方便描述,本文假设网络有 $N = n + m$ 个节点,对应坐标向量为 $X = [x_1, \dots, x_n, x_{n+1}, \dots, x_{n+m}]^T$,其中 $x_i = [x_{i1}, \dots, x_{ik}]$, $k = 2$ 或 3 (分别对应 2 维或 3 维空间)。其中第 1~ n 个节点是未知节点,第 $(n+1) \sim (n+m)$ 个节点是具有先验位置信

息的信标节点, d_{ij} 是节点 i, j 的欧氏距离, $\hat{d}_{ij}$ 是节点 i, j 间的测距,若节点 i, j 是 1 跳邻居, $\hat{d}_{ij}$ 直接用节点间测距表示,若节点 i, j 不是 1 跳邻居, $\hat{d}_{ij}$ 用节点 i, j 的最短路径表示。

基于局部网络拓扑特征的鲁棒节点定位算法在获得未知节点的全局初始坐标后,采用对式(9)所示全局代价函数^[8]进行最小优化,实现定位求精。其中 ω_{ij} 为权值函数,当节点 i, j 为非邻居节点(即无距离测量值)时, $\omega_{ij} = 0$,本文假设 $\omega_{ii} = 0$, $\omega_{ij} = \omega_{ji}$ (即对称加权机制)。

$$S = 2 \sum_{1 \leq i \leq n} \sum_{i \leq j \leq n+m} \omega_{ij} (\hat{d}_{ij} - d_{ij}(X))^2 \quad (9)$$

对式(9)所示全局代价函数进行改写,得到式(10)所示的分布式模型,其中 C 为常数, S_i 为节点 i 的局部代价函数。

$$S = \sum_{i=1}^n S_i + C$$

$$S_i = \sum_{\substack{j=1 \\ j \neq i}}^n \omega_{ij} (\hat{d}_{ij} - d_{ij}(X))^2 + 2 \sum_{j=n+1}^{n+m} \omega_{ij} (\hat{d}_{ij} - d_{ij}(X))^2 \quad (10)$$

当 $m = 0$ 时(即无信标节点), $\partial S / \partial x_i = 2 \partial S_i / \partial x_i$, x_i 变化对全局代价函数的影响与对局部代价函数的影响一致,则全局代价函数 S 的优化问题可转化为局部代价函数 S_i 的优化问题。每个未知节点通过最小化本地局部代价函数实现全局代价函数的最小化,实现分布式节点定位求精。

本文采用高斯函数加权机制,对于处于可接受范围内的测距数据赋予相应权值,其定义如式(11)所示,即根据邻居节点局部网络测距方差进行高斯核赋权。

$$\omega_{ij} = \exp[-(Std'_{ij})^2] \quad (11)$$

式(10)所示局部代价函数为非线性二次函数,本文采用 DFP 算法(Davidon Fletcher Powell,即变尺度法)^[11]进行求解。DFP 算法是一种拟牛顿方法,它使用目标函数梯度构造一系列矩阵来近似 Hessian 矩阵的逆,计算量相对较少,至多经过 2 步迭代就能终止。

LFLS 定位算法步骤如下:

- (1) 算法初始化。设置允许误差 ϵ 。
- (2) 对节点间距离进行多次测量平均(可选),然后使用平均后的数据计算局部网络拓扑特征参数。
- (3) 使用基于局部拓扑信息的粗差识别及剔除

算法,剔除测距粗差,更新剔除粗差后的局部网络拓扑参数,构建有效 1 跳邻居集。

(4) 节点基于有效 1 跳邻居集,使用 ABC 算法构建初始局部坐标系。

(5) 使用 3.2 节介绍的增量贪婪算法,对所有局部坐标系进行融合,生成全局相对坐标系。

(6) 利用网络信标节点,把全局相对坐标变换为初始全局绝对坐标 $X^{(0)} \in R^t$ (t 为空间维数,不失一般性,本文取 $t=2$)。

(7) 节点 i 接收广播包,获得上一轮邻居节点的更新坐标。

(8) 令 $k=1$,置 $H_k = I_2$ (与节点坐标向量维数相同的单位矩阵),计算局部代价函数 S_i 在 $x^{(k)}$ 处梯度值:

$$g_k = \nabla S_i(x^{(k)}) = \left[\frac{\partial S_i(x^{(k)})}{\partial x_i} \quad \frac{\partial S_i(x^{(k)})}{\partial y_i} \right] \quad (12)$$

其中

$$\begin{aligned} \frac{\partial S_i(x^{(k)})}{\partial x_i} &= 2 \sum_{\substack{j=1 \\ j \neq i}}^n \omega_{ij} (\hat{d}_{ij} - d_{ij}(X)) \frac{\partial d_{ij}(X)}{\partial x_i} \\ &\quad + 4 \sum_{j=n+1}^{n+m} \omega_{ij} (\hat{d}_{ij} - d_{ij}(X)) \frac{\partial d_{ij}(X)}{\partial x_i} \\ \frac{\partial d_{ij}(X)}{\partial x_i} &= \frac{\partial}{\partial x_i} ((x_i - x_j)^2 + (y_i - y_j)^2)^{\frac{1}{2}} \\ &= \frac{(x_i - x_j)}{d_{ij}(X)} \end{aligned}$$

$$\text{类似可得 } \frac{\partial S_i(x^{(k)})}{\partial y_i}。$$

(9) 令目标代价函数的搜索方向 $d^{(k)} = -H_k g_k$, 判断 $\|\nabla S_i(x^{(k)})\| \leq \epsilon$ 是否成立,若成立,则算法停止迭代,输出 $x_{p+1} = x^{(k)}$, 执行步骤(13);否则执行步骤(10)。

(10) 从 $x^{(k)}$ 出发沿方向 $d^{(k)}$ 搜索,求步长 λ_k , 使 $S_i(x^{(k)} + \lambda_k d^{(k)}) = \min_{\lambda \geq 0} S_i(x^{(k)} + \lambda d^{(k)})$, 令 $x^{(k+1)} = x^{(k)} + \lambda_k d^{(k)}$ 。

(11) 若 $k=2$ (即节点坐标向量维数),则令 $x^{(1)} = x^{(k+1)}$, 返回步骤(8);否则执行步骤(12)。

(12) 令 $g_{k+1} = \nabla S_i(x^{(k+1)})$, $p^{(k)} = x^{(k+1)} - x^{(k)}$, $q^{(k)} = g_{k+1} - g_k$, 使用 DFP 校正公式得校正矩阵:

$$\Delta H = \frac{p^{(k)} p^{(k)T}}{p^{(k)T} q^{(k)}} - \frac{H_k q^{(k)} q^{(k)T} H_k}{q^{(k)T} H_k q^{(k)}}$$

由此得 $H_{k+1} = H_k + \Delta H$ 。置 $k = k + 1$, 返回步骤(9)。

(13) 节点计算坐标更新后的局部代价函数 $s_i(x_{p+1})$, 并把新坐标 (x_{p+1}) , 局部拓扑信息

$LocalInfo_i$ 及局部代价函数值 $s_i(x_{p+1})$ 广播给邻居节点。

(14) 当全局代价函数 $S^{(p-1)} - S^{(p)} < \epsilon$ 时,节点定位算法停止,否则广播新坐标,返回步骤(7)继续进行下一轮的优化迭代。

3.4 算法计算复杂度及通信开销

计算复杂度采用软件仿真方式进行评估。仿真平台:PC(Pentium(R) 4 CPU 1.50GHz, 448MB), Matlab 7.1。

表 1 比较了不同网络规模下 LFLS 算法的仿真计算时间及迭代次数。可见,随着网络规模的增大, LFLS 算法的总运行时间近似线性增加,而平均每个节点的运行时间以及单个广播周期内每个节点的平均迭代次数基本保持不变,表明 LFLS 算法具有良好的网络可扩展性。表 1 还表明,网络节点坐标更新需要广播的周期数并不会随着网络规模的增大而增加。由于定位算法每进行一轮优化迭代,就需进行一次无线通信,考虑到未知节点每传输 1 比特信息 100m 距离需要消耗的能量相当于执行 3000 条指令所消耗的能量^[12],可见, LFLS 算法能量效率较高。

表 1 均匀网格部署条件下 LFLS 算法运算时间和迭代次数

节点数目	单个节点总的广播次数	单广播周期内每个节点的迭代次数	LFLS 算法总运行时间(s)	单个节点平均运行时间(s)
49	7.02	1.00	81.41	1.81
100	8.13	1.02	180.26	1.88
196	7.04	1.01	363.29	1.89

节点平均连通度 9, 节点接收功率 5 次测量平均。

同计算开销相比,节点无线通信所消耗的能量占主导地位。通信复杂度考虑节点通信量随网络规模增大的变化规律,即节点平均通信总能耗与节点数目 n 间的关系。本文采用公式(13)^[13]所示的通信能耗模型,其中 $b(n)$ 为每个传输数据包的平均比特数, $h(n)$ 为通信平均跳段数, $e(n)$ 为传输数据包内 1 比特数据一个跳段需要消耗的能量。

$$\epsilon(n) = b(n) \times h(n) \times e(n) \quad (13)$$

为简化分析起见,假设节点均匀部署在单位长度的正方形或立方体区域(分别对应 $K=2$ 或 3 维网络)。LFLS 算法要求每个节点向其它能够获得测距估计的节点发送其位置估计。假设每个节点能感知距离自身 d (无线通信半径)距离内的所有其它节点,则能与该节点进行通信的平均邻居数目上限为 $O(cnd^K)$ (单位圆或单位球边沿节点的邻居数相

对较少),其中 c 为与网络部署面积相关的常数, n 为网络节点数。由于 LFLS 算法要求每个节点完成一次迭代计算后,需要把更新的坐标估计及局部网络拓扑参数发送给邻居节点,因此平均发送的比特数为 $O(n^2 L d^K)$ (其中 L 为算法迭代次数),而每次广播通信为 1 个跳段,即 $h(n) = 1$,故 LFLS 算法无线通信的平均能量消耗如公式 (14) 所示,其中 $e_{\text{LFLS}}(n)$ 与节点无线通信半径 d 相关。

$$\varepsilon_{\text{LFLS}}(n) \sim O(n^2 L d^K e_{\text{LFLS}}(n)) \quad (14)$$

4 仿真实验

4.1 截断误差条件下少量粗差剔除

本节主要评测算法在存在少量测距粗差时的定位性能。在 $[0,1] \times [0,1]$ 单位区域网格点上均匀部署 7×7 个节点,网络节点测距误差小于 20%,按一定比例缩放部分测距值,构造测距粗差,如图 3 所示。其中连线表示两节点可互相通信,黑色连线表示测距误差处在正常范围内(测距误差小于 20%),粗黑色虚线为构造粗差(共 4 个)。

使用本文所提的粗差检测算法,可识别出图 3 所示所有粗差。可见,当网络中大部分测距数据较准确时,每个节点利用邻居节点的局部网络拓扑特征信息对本地测距进行检测,可准确识别出所有粗差测距。

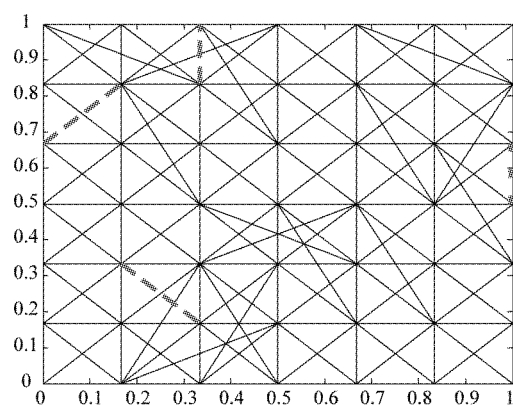


图 3 截断误差条件下少量测距粗差原始数据

在每个节点使用粗差识别算法剔除粗差后,采用高斯加权最小二乘法进行节点定位优化,可明显提高节点定位精度。在没有剔除粗差时,直接使用高斯最小二乘算法进行定位,节点定位误差 RMSE (均方根误差) 为 8.57%,如图 4 所示,而剔除粗差后, RMSE 降为 6.49%,如图 5 所示。

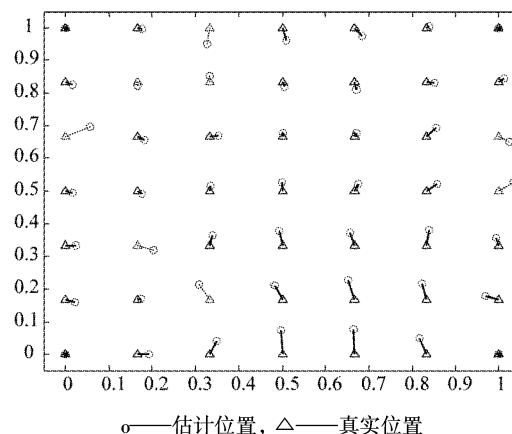


图 4 截断误差条件下未剔除粗差的定位求精性能

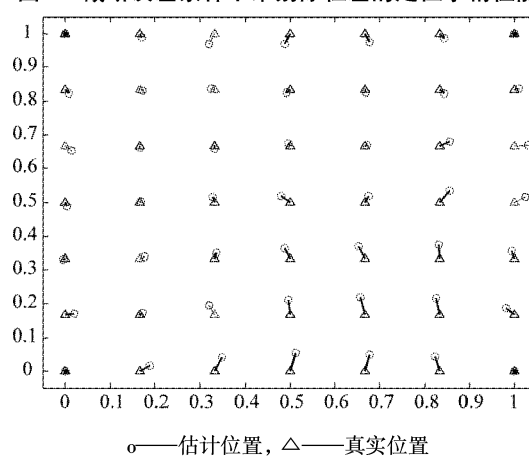


图 5 截断误差条件下剔除粗差后的定位求精性能

剔除粗差不仅整体提高了定位精度,而且能显著提高粗差直接相关节点的定位精度,如图 6 所示。在未剔除粗差时,粗差相关节点的 RMSE 为 11.50%,剔除粗差后,直接相关节点的 RMSE 降为 6.33%,改进 5% 以上。

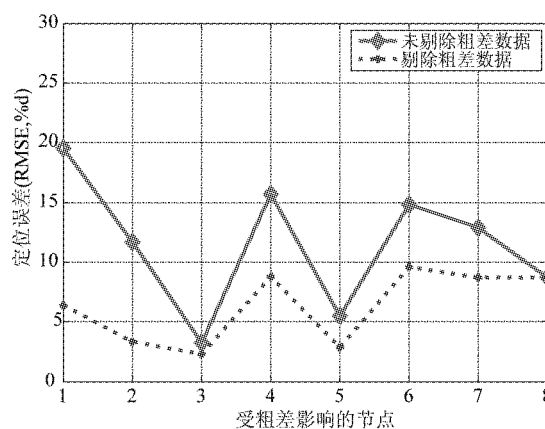


图 6 截断误差条件下少量粗差直接相关节点的定位精度改进

4.2 对数高斯分布条件下的粗差剔除

本节主要评测无截断误差条件下粗差检测算法

性能。对节点间测距误差大小不作任何截断限制,直接使用 RSSI 测距模型(对数高斯分布)^[14],网络中同时存在测距高估和测距低估粗差,粗差数目较多。

图 7 显示使用粗差检测方法的一次实验结果(不同实验产生的粗差数目不同),其中粗黑色虚线是算法检测出的粗差,整个网络共检测出 6 个粗差,影响 12 个节点。

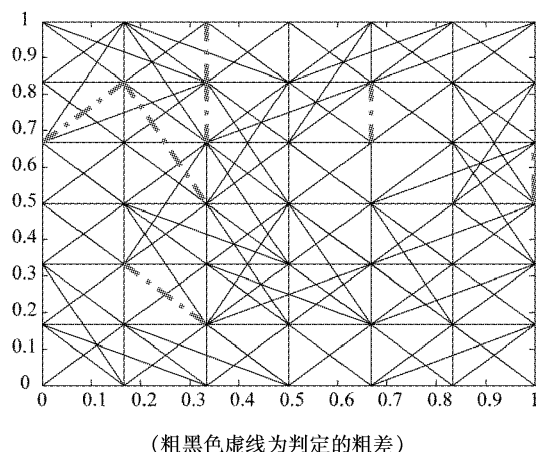


图 7 对数高斯分布条件下测距粗差检测

剔除检测出的粗差测距后,使用基于高斯加权最小二乘定位算法。当没有剔除测距粗差时,网络所有节点的平均定位误差为 25.31%,如图 8 所示,剔除粗差后,所有节点的 RMSE 降为 21.65%,如图 9 所示。粗差边直接相关节点的定位精度改进如图 10 所示。没有剔除粗差时,粗差测距相关节点的 RMSE 为 28.48%,剔除粗差后定位误差降到 22.08%,定位性能改进约 6%。可见,节点使用局部网络拓扑信息,对测距粗差进行检测及剔除,可为优化定位算法提供更准确的测距数据,从而提高节点定位精度。

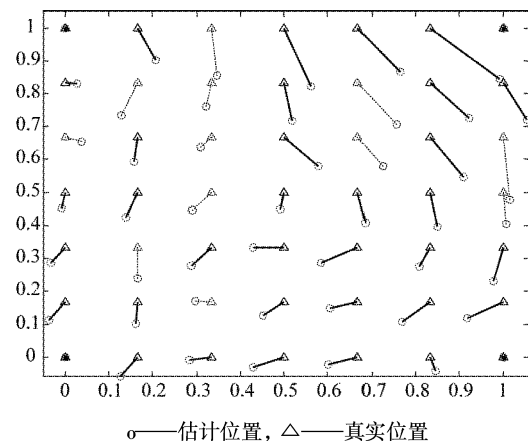


图 8 对数高斯分布条件下粗差未剔除定位求精性能

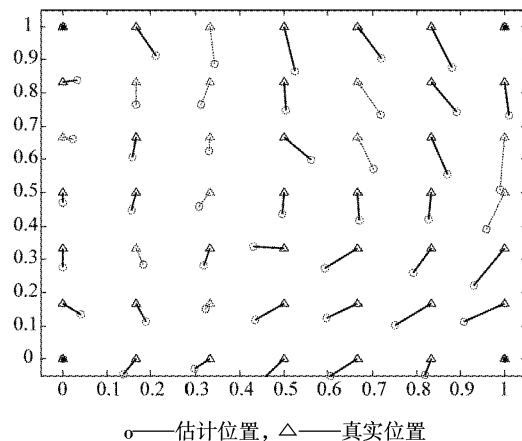


图 9 对数高斯分布条件下剔除粗差后定位求精性能

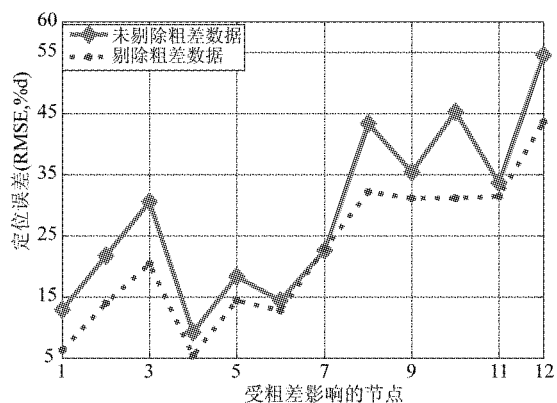


图 10 对数高斯分布条件下粗差相关节点的定位精度改进

5 结论

针对经典最小二乘节点定位算法抗差性较弱,少量较大测量误差影响节点定位精度的不足,基于网络局部拓扑特征的鲁棒定位算法通过构建未知节点局部范围内的网络特征参数,识别和剔除测距粗差,为加权最小二乘定位算法提供更准确的测距数据,不仅提高了整体平均定位精度,而且大幅度提高了粗差测距相关节点的定位精度,表现出良好的定位鲁棒性。

参考文献

- [1] Zhou G, He T, Krishnamurthy S, et al. Models and solutions for radio irregularity in wireless sensor networks. *ACM Transactions on Sensor Networks*, 2006, 2(2): 221-262
- [2] Whitehouse C D. Understanding the prediction gap in multi-hop localization: [Ph.D dissertation]. Berkeley: University of California, 2006. <http://www.cs.virginia.edu/~whitehouse/pubs/whitehouse06dissertation.pdf>
- [3] Whitehouse K, Karlof C, Woo A, et al. The effects of rang-

- ing noise on multihop localization: an empirical study. In: Proceedings of the 4th International Symposium on Information Processing in Sensor Networks (IPSN), Los Angeles, CA, USA, 2005. 73-80
- [4] Shang Y, Shi H C, Ahmed A A. Performance study of localization methods for ad-hoc sensor networks. In: Proceedings of the 1st International Conference on Mobile Ad hoc and Sensor Systems (MASS), Fort Lauderdale, Florida, USA, 2004. 184-193
- [5] Li Z, Trappe W, Zhang Y Y, et al. Robust statistical methods for securing wireless localization in sensor networks. In: Proceedings of the 4th International Symposium on Information Processing in Sensor Networks, Los Angeles, CA, USA, 2005. 91-98
- [6] Kwon Y, Mochitov K, Sundresh S, et al. Resilient localization for sensor networks in outdoor environments. In: Proceedings of the 25th International Conference on Distributed Computing Systems (ICDCS), Columbus, Ohio, USA, 2005. 643-652
- [7] 罗海勇, 李锦涛, 赵方等. 一种基于加权多尺度分析技术的鲁棒节点定位算法. 自动化学报, 2008, 34(3): 288-297
- [8] Costa J A, Patwari N, Hero III A O. Distributed multidimensional scaling with adaptive weighting for node localization in sensor networks. *IEEE/ACM Transactions on Sensor Networks*, 2006, 2(1): 39-64
- [9] Savarese C, Rabaey J, Beutel J. Locating in distributed ad-hoc wireless sensor networks. In: Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), Salt Lake City, Utah, USA, 2001. 2037-2040
- [10] Shang Y, Ruml W. Improved MDS-based localization. In: Proceedings of the 23rd Annual Joint Conference of the IEEE Computer and Communications Societies, Hong Kong, China, 2004. 2640-2651
- [11] 黄红选, 韩继业编著. 数学规划. 北京: 清华大学出版社, 2006. 200-203
- [12] Raghunathan V, Schurgers C, Park S, et al. Energy-aware wireless microsensor networks. *IEEE Signal Processing Magazine*, 2002, 19(2): 40-50
- [13] Rabbat M, Nowak R. Distributed optimization in sensor networks. In: Proceedings of the 3rd International Symposium on Information Processing in Sensor Networks (IPSN'04), Berkeley, California, USA, 2004. 20-27
- [14] Coulson A J, Williamson A G, Vaughan R G. A statistical basis for lognormal shadowing effects in multipath fading channels. *IEEE Transactions on Communications*, 1998, 46(4): 494-502

A robust node localization algorithm based on local network feature in wireless sensor networks

Zhao Fang^{* **}, Ma Yan^{*}, Luo Haiyong^{***}, Lin Quan^{****}, Song Maoqiang^{*}

(^{*} School of Software Engineering, Beijing University of Posts and Telecommunications, Beijing 100876)

(^{**} Researching Academy of Network Technology, Beijing University of Posts and Telecommunications, Beijing 100876)

(^{***} Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100190)

(^{****} Beijing University of Aeronautics and Astronautics, Beijing 100083)

Abstract

This paper presents a robust localization algorithm which employs local networks feature to identify and discard ranging outliers in evenly deployed sensor networks. It first filters out 1-hop ranging outliers of the unknown nodes by using the threshold parameters of extreme ranging overestimates and extreme ranging underestimate, and then employs Gaussian-weighted least squares to position nodes. For received signal strength based range measurements, the simulation results confirm that this localization scheme outperforms the traditional weighted least squares (WLS), which do not employ outlier identification and deletion. Especially, the scheme can remarkably improve the localization accuracy of the unknown nodes which are directly related to the outliers.

Key words: wireless sensor networks, node localization, least square scaling, local network feature