

1-3-2019

Anchor Adaptive Movement Algorithm for Mobile Nodes Localization

Zhou Fei

Chongqing Key Laboratory of Optical Communication and Networks, Chongqing University of Posts and Telecommunications, Chongqing 400065, China;

Wennan Liu

Chongqing Key Laboratory of Optical Communication and Networks, Chongqing University of Posts and Telecommunications, Chongqing 400065, China;

Follow this and additional works at: <https://dc-china-simulation.researchcommons.org/journal>



Part of the [Artificial Intelligence and Robotics Commons](#), [Computer Engineering Commons](#), [Numerical Analysis and Scientific Computing Commons](#), [Operations Research, Systems Engineering and Industrial Engineering Commons](#), and the [Systems Science Commons](#)

This Paper is brought to you for free and open access by Journal of System Simulation. It has been accepted for inclusion in Journal of System Simulation by an authorized editor of Journal of System Simulation.

Anchor Adaptive Movement Algorithm for Mobile Nodes Localization

Abstract

Abstract: Location awareness is very important in mobile sensor networks. The existing methods mainly focus on the improvement based on the MCL (Monte Carlo Localization), few of them study the anchor path planning. However, anchors random movement usually causes the waste of energy and resource. Aiming at this deficiency, this paper adopts the grid benefit criterion and achieves anchors adaptive movement. *Threshold instead of traversal method is used to reduce the amount of calculation and positioning time. In order to solve the anchors aggregation problem during adaptive movement, we divide the target area into four subareas and constrain part of the anchors moving in the subarea. By this method we improve the amount of localized unknown nodes obviously.*

Keywords

mobile sensor networks, localization, anchor path planning, energy-efficient

Recommended Citation

Zhou Fei, Liu Wennan. Anchor Adaptive Movement Algorithm for Mobile Nodes Localization[J]. Journal of System Simulation, 2018, 30(5): 1885-1892.

动态节点定位过程中锚节点的自适应移动方案

周非, 刘文楠

(光通信与网络重点实验室, 重庆邮电大学, 重庆 400065)

摘要: 移动传感器网络中节点的位置信息是至关重要的。现有的定位方法多是基于 MCL (Monte Carlo Localization) 进行改进, 鲜有对锚节点路径优化的算法。锚节点的随机移动往往造成资源以及能量的浪费。针对这种不足, 借鉴节点效益最大化准则对锚节点路径进行优化, 实现其自适应移动; 在自适应移动的基础上对算法进行改进, 采用阈值代替遍历的方法减少了计算量和定位时间; 针对自适应移动出现的锚节点聚集的情况, 对锚节点进行分区置点, 较大程度地对未知节点进行覆盖, 提高了未知节点的定位数量。

关键词: 移动传感器网络; 节点定位; 锚节点路径优化; 能量节省

中图分类号: TP212.9 文献标识码: A 文章编号: 1004-731X (2018) 05-1885-08

DOI: 10.16182/j.issn1004731x.joss.201805034

Anchor Adaptive Movement Algorithm for Mobile Nodes Localization

Zhou Fei, Liu Wenna

(Chongqing Key Laboratory of Optical Communication and Networks, Chongqing University of Posts and Telecommunications, Chongqing 400065, China)

Abstract: Location awareness is very important in mobile sensor networks. The existing methods mainly focus on the improvement based on the MCL (Monte Carlo Localization), few of them study the anchor path planning. However, anchors random movement usually causes the waste of energy and resource. Aiming at this deficiency, this paper adopts the grid benefit criterion and achieves anchors adaptive movement. *Threshold instead of traversal method is used to reduce the amount of calculation and positioning time. In order to solve the anchors aggregation problem during adaptive movement, we divide the target area into four subareas and constrain part of the anchors moving in the subarea. By this method we improve the amount of localized unknown nodes obviously.*

Keywords: mobile sensor networks; localization; anchor path planning; energy-efficient

引言

随着无线传感器网络的飞速发展和广泛应用, 对传感器中节点的位置信息要求也越来越高。传统

的定位算法主要针对于静态网络, 而当前静态节点定位技术已无法满足新增定位的需要, 移动定位技术的发展扩展了无线传感器网络的应用领域, 进而使事件检测更灵活^[1]。

Hu 和 Evans 最早提出了适用于动态网络节点定位的 MCL 算法^[2], 算法利用加权粒子表示节点位置的后验分布, 实现了对动态节点的定位。为了简化算法, Baggio 等提出了 MCB (Monte Carlo Boxed) 算法^[3], 提高了采样效率。除此之外, 还有



收稿日期: 2016-07-04 修回日期: 2016-08-24;
基金项目: 国家自然科学基金(61471077), 重庆市教委科学技术研究项目(KJ1400413);
作者简介: 周非(1977-), 男, 湖北浠水, 博士, 教授, 研究方向为无线定位、信号处理、网络安全及图像处理; 刘文楠(1991-), 女, 河北邢台, 硕士, 研究方向为无线定位。

<http://www.china-simulation.com>

• 1885 •

很多基于 MCL 的改进算法^[4-7], 这些算法有的结合了最大似然估计、最小二乘对未知节点坐标进行估计^[4], 有的对权值进行优化^[5], 还有的重新限制滤波条件, 把锚节点移动距离考虑进去, 从而达到提高定位精度的目的^[6]。然而这些算法均没有考虑锚节点的路径规划, 因此不可避免地面临能量浪费, 锚节点利用率低等问题。

文献[8-9]提出并比较了几种锚节点按照固定路线移动的路径规划方案, 实现了单个锚节点对整个监测区域的全覆盖, 然而当未知节点处于移动状态时就不适用了。蒲实在文献[10]中提出了一种适用于动态网络的锚节点路径规划算法, 首先选择邻节点数量最多的节点作为初始节点, 然后选择节点转移距离最小的能够遍历初始节点的路径作为最优路径。然而这种算法并未考虑锚节点的移动速度, 且定位精度不够理想。文献[11]提出一种未知节点移动状态下锚节点的自适应路径优化方案, 方案中提出节点效益作为锚节点移动位置的判断指标。文献[12]对节点效益最大化的算法进一步提高, 把锚节点移动距离考虑进去, 减少了锚节点移动过程中的能量消耗, 这两种方案虽然解决了锚节点穿越节点稀疏区域的问题, 但计算量很大, 定位时间较长, 且随着定位时间的增加, 锚节点会逐渐聚集到未知节点密度大的地方, 造成锚节点分布不均匀, 未知节点定位覆盖率低的问题, 本文主要针对这些问题进行改进。

1 相关工作

假设监测区域内分布有 n 个未知节点, 个锚节点, 均处于移动状态。在定位过程中时间是离散的, 令 $\mathbf{G}_t = \{\chi_t^1, \chi_t^2, \dots, \chi_t^n\}$, $\mathbf{A}_t = \{\xi_t^1, \xi_t^2, \dots, \xi_t^m\}$, 其中 χ_t^i 与 ξ_t^j 分别表示 t 时刻第 i 个未知节点 m 以及第 j 个锚节点的坐标^[12]。算法的目的是先确定锚节点下一时刻的最优坐标 \mathbf{A}_{t+1} , 然后利用 MCB 定位算法来计算下一时刻未知节点的估计坐标 \mathbf{G}_{t+1} 。

在仿真环境中节点的移动模型表示如下:

$$\begin{aligned} V_{t+1} &= \min \left\{ \max [V_t + \delta_v, V_{\min}], V_{\max} \right\} \\ \alpha_{t+1} &= \alpha_t + \delta_\alpha \\ x_{t+1} &= x_t + V_t \times \cos(\alpha_t) \\ y_{t+1} &= y_t + V_t \times \sin(\alpha_t) \end{aligned} \quad (1)$$

式中: V_{\max}, V_{\min} 分别表示节点移动的最大最小速度; α_t 表示节点移动的方向; δ 为随机量; $\delta V_t, \delta \alpha_t$ 分别表示节点速度与方向的改变量。

节点在单位时间内移动的距离不可能大于其最大速度, 因此, 我们以节点坐标为中心, 以其最大移动速度为边界构建其边界盒子: 假设节点 t 时刻的坐标 (x_t, y_t) , 最大移动速度 V_{\max} , 则其边界盒子为,

$$P = \left((x_{\min}^p, y_{\min}^p), (x_{\max}^p, y_{\max}^p) \right) \quad (2)$$

式中:

$$\begin{aligned} x_{\min}^p &= (x_t - V_{\max}), y_{\min}^p = (y_t - V_{\max}) \\ x_{\max}^p &= (x_t + V_{\max}), y_{\max}^p = (y_t + V_{\max}) \end{aligned}$$

MCB 算法中未知节点需要根据锚节点的广播信息来估计 $t+1$ 时刻自身位置, 故若锚节点 $t+1$ 时刻坐标为 (x_{t+1}, y_{t+1}) , 则定义其 t 时刻通信盒子为:

$$R = \left((x_{\min}^R, y_{\min}^R), (x_{\max}^R, y_{\max}^R) \right) \quad (3)$$

式中:

$$\begin{aligned} x_{\min}^R &= (x_{t+1} - r), y_{\min}^R = (y_{t+1} - r) \\ x_{\max}^R &= (x_{t+1} + r), y_{\max}^R = (y_{t+1} + r) \end{aligned}$$

根据锚节点的通信盒子以及未知节点的边界盒子构建节点效益模型如图 1 所示。

公式定义为:

$$E_{t+1}^i = \left((x_{\min}^E, y_{\min}^E), (x_{\max}^E, y_{\max}^E) \right) \quad (4)$$

式中:

$$\begin{aligned} x_{\min}^E &= \max(x_{\min}^p, x_{\min}^R), y_{\min}^E = \max(y_{\min}^p, y_{\min}^R) \\ x_{\max}^E &= \min(x_{\max}^p, x_{\max}^R), y_{\max}^E = \min(y_{\max}^p, y_{\max}^R) \end{aligned}$$

定义未知节点 n_j 对锚节点 s_i 的节点效益为:

$$G_j^i = P_{t+1}^n - E_{t+1}^n \quad (5)$$

在仿真模型中, 我们用面积表示节点效益大小, 即 $G = S_p - S_E$

其中:

$$S_E = \sqrt{(x_{\max} - x_{\min})^2 + (y_{\max} - y_{\min})^2}$$

$$S_p = (2 \times V_{\max} + 1)^2$$

而锚节点在 i 处的节点效益为 s_i 周围所有未知节点的节点效益之和:

$$G^i = \sum_{j=1}^m G_j^i \quad (6)$$

式中: m 表示 s_i 附近未知节点数量。

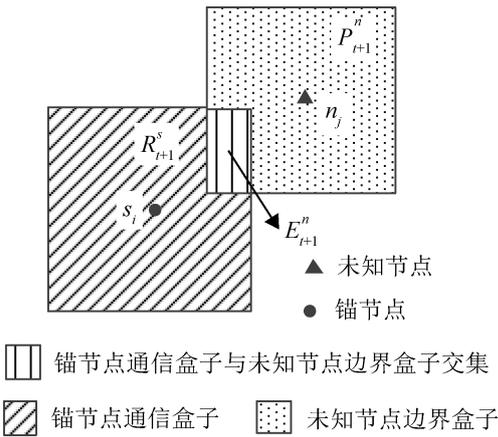


图 1 锚节点的节点效益模型
Fig. 1 Anchor grid benefit model

算法一具体步骤如下:

1. 初始化未知节点与锚节点的坐标 χ_t^n, ξ_t^s 。
2. 构建 t 时刻锚节点的边界盒子 P_t^s 。
3. 按照图 2 所示方法, 将 P_t^s 划分为 $n \times n$ 个大小相等的小正方形, 每个小正方形标记为 $r_{i,j}$ 。

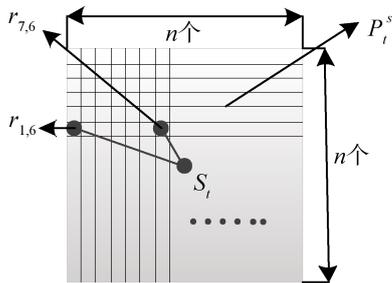


图 2 锚节点边界盒子分割示意图
Fig. 2 Partition diagram of anchor boundary box

4. 从 $i=1; j=1$ 开始, 依次计算每个 $r_{i,j}$ 的节点效益 $G_{i,j}$ 放入集合 Π 中。
5. 从集合 Π 中选取 $G_{i,j}$ 最大的点作为锚节点下一时刻坐标。

6. 更新锚节点坐标 ξ_{t+1}^s 。

7. 利用 MCB 算法, 求得未知节点的坐标 χ_{t+1}^n , 并更新未知节点坐标。

2 优化的锚节点路径规划方案

2.1 定位贡献因子

理论上, 锚节点可以移动到其边界盒子的任意位置, 但是不同的位置对未知节点的定位贡献不同, 因此, 我们需要在锚节点的边界盒子内找到一个对未知节点贡献最大的点作为锚节点下一时刻的坐标。为了衡量不同位置的锚节点的贡献值大小, 我们定义衡量定位贡献大小的因子 I 。

$$I = \begin{cases} \frac{w_1 G + w_2 H}{w_3 d}, & H \neq 0 \\ 0, & \text{其他} \end{cases} \quad (7)$$

式中: w_1, w_2, w_3 分别表示权重; G 表示节点效益; H 表示锚节点附近未知节点数量; d 表示锚节点移动距离。为了防止锚节点距离过小造成贡献值较大的情况, 我们规定当 H 为 0 时, 定位贡献值为 0。

首先节点效益 G 越大, 定位阶段采样区域越小, 相应的定位精度也会越高^[14], 因此 G 是影响定位精度的因子。

其次, 定位过程中, 衡量定位性能的第二个因素是未知节点定位数量, 即定位覆盖率。定位覆盖率也指 t 时刻监测区域内定位的未知节点数量与所有的未知节点数量之比。假设监测区域内共有 n 个未知节点, 锚节点位于位置 a 时, 其通信盒子与 σ_a 个未知节点边界盒子有交集, 那么在定位阶段有未知节点接收到锚节点广播信息的概率为 σ_a/n^2 , 我们定义:

$$H = \frac{\sigma_a}{n^2} \quad (8)$$

H 越大, 定位阶段未知节点接受到锚节点广播信息的概率越大, 相对应的定位覆盖率越高。

最后, 能量消耗也是评价定位性能的因素之一, 在移动传感器网络中, 锚节点自身所携带的能量是有限的, 而锚节点的移动是能量消耗的主要来

源之一，定义锚节点单位时间内移动的距离为 d ， d 越大，能量消耗越多，如图 2 所示，锚节点移动到位置 $r_{7,6}$ 消耗的能量要小于位置 $r_{1,6}$ 。

$$d = \|\xi_t^s - \xi_{t+1}^s\| \quad (9)$$

2.2 减少计算复杂度的算法优化

在移动网络中，定位速度是衡量定位性能的一个重要指标，定位速度主要是由算法复杂度决定的。原有的节点效益最大化算法虽然实现了锚节点的自适应移动，使得能量和资源得到了更加充分的利用，然而在确定锚节点位置时，计算量较大。

原始算法的复杂度主要存在于对锚节点的边界盒子进行遍历。假设监测区域内共有 m 个锚节点，每个锚节点都需要通过构建边界盒子选择最佳移动位置，如果把边界盒子分为 $n \times n$ 个小正方形，那么就需要进行 $m \times n^2$ 次计算。针对这个问题，文章利用阈值代替遍历求最大值的方法来优化原有的算法。

边界盒子的不同位置对未知节点定位的贡献量不同。2.1 节我们总结了衡量锚节点定位贡献量的参数，并用 I 代替节点效益来最终衡量不同位置的定位贡献值。

首先预先设定阈值 λ ，然后将锚节点的边界盒子划分为 $n \times n$ 个小正方形，依次计算每个小正方形的贡献值 I ，一旦出现 I 大于阈值 λ 。那么锚节点位置选择结束，并输出该点坐标作为锚节点下一时刻最佳位置。若遍历锚节点的边界盒子之后仍未找到大于 λ 的点，则选取 I 最大的点作为锚节点下一时刻最佳位置。

算法二具体步骤如下：

1. 初始化未知节点与锚节点的坐标 χ_t^n, ξ_t^s 。
2. 构建 t 时刻锚节点的边界盒子 P_t^s 。
3. 将 P_t^s 划分为 $n \times n$ 个大小相等的小正方形，每个小正方形记为 $r_{i,j}$ 。
4. 从 $i=1; j=1$ 开始，依次计算每个 $r_{i,j}$ 的定位贡献值 $I_{i,j}$ 。若出现 $I_{i,j} \geq \lambda$ ，则输出 $r_{i,j}$ 中心位置对应的坐标作为锚节点 $t+1$ 下一时刻的移动坐标。

5. 若遍历锚节点边界盒子仍未出现 $I_{i,j} \geq \gamma$ 则输出 $I_{i,j}$ 最大的点作为锚节点 $t+1$ 下一时刻的移动坐标。

6. 更新锚节点坐标 ξ_{t+1}^s 。

7. 利用 MCB 算法，求得未知节点的坐标 χ_{t+1}^n ，并更新未知节点坐标。

对于阈值的选择，我们对算法进行 N 次仿真，求得每次锚节点最大贡献值 I_{\max}^i 并对其求均值：

$$\lambda = \frac{1}{N} \sum_{i=1}^N I_{\max}^i \quad (10)$$

我们通过仿真数据进一步证明该算法的优势。在 400×400 的监测区域内布置 20 个锚节点，其中未知节点个数为 100，节点最大移动速度 $V_{\max} = 20 \text{ m/s}$ ，构建的锚节点边界盒子大小为 $41 \text{ m} \times 41 \text{ m}$ ，为计算方便我们以 1 m^2 为单位，将边界盒子划分为 $41 \times 41 = 1681$ 个小正方形。

统计数据如表 1 所示。

表 1 算法计算次数比较

Tab. 1 Comparison of two algorithms on calculation amount

序号	遍历 算法	阈值 算法	序号	遍历 算法	阈值 算法
1	1 681	56	11	1 681	101
2	1 681	320	12	1 681	1 681
3	1 681	1 681	13	1 681	992
4	1 681	982	14	1 681	1 681
5	1 681	1 012	15	1 681	624
6	1 681	76	16	1 681	620
7	1 681	1 681	17	1 681	1 006
8	1 681	1 681	18	1 681	1 681
9	1 681	867	19	1 681	265
10	1 681	1 681	20	1 681	1 681

由表 1 可知，阈值算法平均计算次数为 1 018.45，小于遍历算法的 1 681。计算次数减少了 39.4%。其中小于 1 681 的锚节点有 8 个，占总数的 40%。

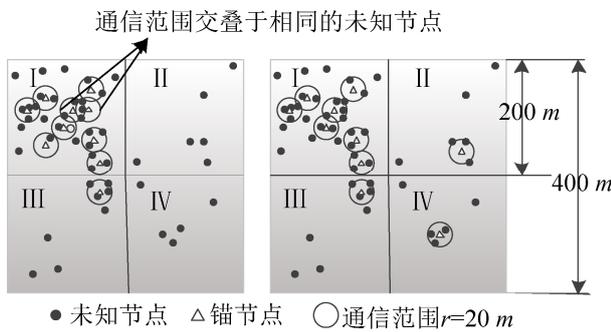
除此之外，阈值算法中，对其中的一个锚节点进行 1 000 步移动，平均每步计算次数为 998.23，相较于遍历算法，计算次数减少 40.6%。

2.3 基于分区置点的算法优化

节点效益最大化算法虽然解决了随机移动造成的能量和资源浪费问题,但是随着时间增加,锚节点会逐渐向未知节点密度较高的区域移动,使得对未知节点的定位覆盖率降低。

根据公式(5)可知,定位贡献值 I 与未知节点定位数量 H 成正比,故为了确保较大的 I 值,锚节点会逐渐向未知节点密度大的区域聚集。

如图 3 为某一时刻节点密度分布图,在 400×400 的监测区域内分布 50 个未知节点,10 个锚节点,锚节点通信半径 $r = 20$ m。图 3(a)表示未分区置点情况。I 区未知节点密度较高,锚节点逐渐向 I 区聚集,多个锚节点的通信范围交叠于相同未知节点,虽然可以提高该节点的定位精度,却在一定程度上造成了锚节点资源的浪费,II、III、IV 三个子区域的未知节点没有被覆盖,整体未知节点定位覆盖率不高。



(a) 锚节点分区置点示意图 (b) 锚节点非分区置点示意图

图 3 节点非均匀分布示意图

Fig. 3 Non-uniform distribution of nodes

针对这个问题,我们对锚节点进行分区置点。将整个监测区域分为 4 个子区域,每个子区域 $D_k (k=1,2,3,4)$ 内布置固定数量的锚节点 S_p^k ,固定锚节点限制在对应子区域内移动。其余锚节点不受区域限制,称为自由锚节点 S_f 。 S_p^k 的存在保证了每个子区域内至少有一个锚节点对该区域内的未知节点进行定位,避免了在某时刻子区域未知节点密度较小时没有锚节点分布的情况。

图 3(b)表示分区置点的节点分布情况。将四个子区域限定一个固定锚节点,在一定程度上解决了锚节点聚集造成的覆盖率低的问题。 S_f 移动方案与算法二相同,这里不多做赘述, S_p^k 移动方案如算法三所示。

算法三实现步骤如下:

1. 构建子区域

$$D_1 = (0 < x < Xrange / 2, Yrange / 2 < y < Yrange)$$

$$D_2 = (Xrange / 2 < x < Xrange, Yrange / 2 < y < Yrange)$$

$$D_3 = (0 < x < Xrange / 2, 0 < y < Yrange / 2)$$

$$D_4 = (Xrange / 2 < x < Xrange, 0 < y < Yrange / 2)$$

其中: $Xrange, Yrange$ 表示整个观测区域范围。

2. 初始化未知节点与锚节点坐标 $\chi_t^n, \xi_t^{S_p}$ 。

3. 构建 t 时刻锚节点 S_p^k 的边界盒子 $P_t^{S_p^k}$ 。

4. 将 $P_t^{S_p^k}$ 划分为 $n \times n$ 个大小相等的小正方形,记为 $r_{i,j}$, 其中心位置坐标 $(x_{i,j}^k, y_{i,j}^k)$ 。

5. for $i=1$ to n do;

for $j=1$ to n do;

计算 $r_{i,j}$ 的定位贡献值 $I_{i,j}$

if $(x_{i,j}^k, y_{i,j}^k) \in D_k (k=1,2,3,4)$ 且 $I_{i,j} \geq \lambda$

输出 $I_{i,j}$ 作为 S_p^k 下一时刻的坐标

else if $(x_{i,j}^k, y_{i,j}^k) \in D_k$ 且 $I_{i,j} < \lambda$

输出坐标存入集合 Ω

end if

$j=j+1$;

end for

$i=i+1$;

end for

if $i=n$ 且 $j=n$

输出 Ω 中 $I_{i,j}$ 最大的值为 S_p^k 下一时刻的坐标。

6. 更新锚节点坐标 $\xi_{t+1}^{S_p}$

7. 利用 MCB 算法,求得未知节点的坐标 χ_{t+1}^n , 并更新未知节点坐标

为了观察固定锚节点数量对定位性能的影响,我们将 400×400 的观测区域平均分割成 4 个子区域,并在整个监测区域内随机分布 100 个未知节点,锚节点数量总共为 20 个,每个子区域内的固

定锚节点数量比例由 0.05 逐渐增加到 0.25。从图 4 我们可以看出,随着固定锚节点数量的增加,定位平均误差有上升的趋势,而定位的未知节点比例却逐渐增加,这与之前分析的结果是一致的,从实验结果来看,定位误差曲线变化比较缓慢,也就是说固定锚节点数量对定位精度的影响不会很大。而对于定位的未知节点比例曲线来说,当固定锚节点数量高于 2 时,曲线的趋势也相对变得平缓,所以固定的锚节点数量取总数的 10% 为宜。

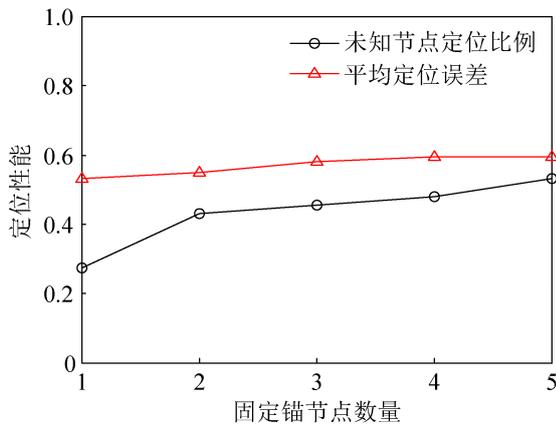


图 4 固定锚节点数量对定位性能的影响

Fig. 4 Influence of fixed anchor node number on positioning performance

3 仿真实验验证

仿真环境如下: Matlab7.0, 监测区域 400×400 锚节点通信半径为 $r=20$ m, 节点最大动速度为 20 m/s, 最小移动速度为 0 m/s。

我们用 GBM (Grid Benefit Maximum 表示节点效益最大化定位算法, 用 OGBM (Optimal Grid Benefit Maximum) 表示 2.2 提出的减少计算复杂度的优化算法, 用 S-OGBM (Optimal Grid Benefit Maximum Based on Subarea) 表示 2.3 提出的基于分区置点的算法优化方案。

仿真性能评价指标:

1. 未知节点定位比例: 我们用接收到锚节点广播信息的未知节点数量 $n_{\text{localized}}$ 比上监测区域内所有的未知节点数量 N 来衡量未知节点定位比例。

$$LP = \frac{n_{\text{localized}}}{N} \quad (11)$$

2. 定位精度: 为避免通信半径对定位精度的影响, 定位精度用估计位置坐标与节点真实坐标之间的欧氏距离与通信半径的比值来衡量。

$$LE = \frac{\sum_{i=1}^{n_{\text{localized}}} \|\chi_i - \hat{\chi}_i\|}{n_{\text{localized}}} \times \frac{1}{r} \quad (i=1, 2, 3, \dots, n_{\text{localized}}) \quad (12)$$

式中: $\hat{\chi}_i$ 为第 i 个未知节点的估算位置; χ_i 为未知节点的实际位置; r 为锚节点的通信半径。

3.1 定位性能随时间变化曲线

本节仿真参数设置如下: 未知节点数量 100 开始, 锚节点数量为 20, 每个子区域 S_p^k 的数量为 2。

根据 MCB 算法的定位原理, 未知节点附近锚节点数量越多, 其定位精度越高, 这是由于锚节点数量越多, 定位时的采样盒子越精确, 使得采取的样本值更接近未知节点的实际位置。基于 GBM 算法的定位算法, 锚节点实现了自适应移动, 随着时间增加, 锚节点逐渐向未知节点密度较高的区域聚集, 造成定位阶段未知节点附近锚节点数量增多, 相应定位精度会略有提高, 如图 5 所示, GBM 与 OGBM 算法的定位误差会逐渐减小。S-OGBM 的定位精度略小于 GBM 和 OGBM, 这是因为部分锚节点分区分布, 密度分布比较均匀, 定位精度曲线相对平稳。OGBM 的定位精度虽然略高于 GBM, 但是相差无几。这说明经过算法优化的 OGBM 算法能够达到较好地定位精度。

图 6 是未知节点定位比例随时间变化曲线, 图中基于 GBM 算法的曲线要远高于锚节点随机移动的 MCB 算法。OGBM 略大于 GBM, 这是由于算法在计算贡献值时将未知节点的数量考虑进去, 从而提升了未知节点定位比例。初始时刻, OGBM 定位比例略高 S-OGBM, 但随着时间增长 OGBM 与 GBM 算法表现出了明显的下降趋势, S-OGBM 逐渐开始高于 OGBM。这是由于随着时间增加, 锚节点会逐渐向未知节点聚集区域移动, 造成锚节

点资源分布不均匀。除此之外 S-OGBM 算法定位数量曲线相对平稳, MCB 算法由于锚节点移动的随机性和不确定性, 定位的未知节点数量也相对起伏比较大。

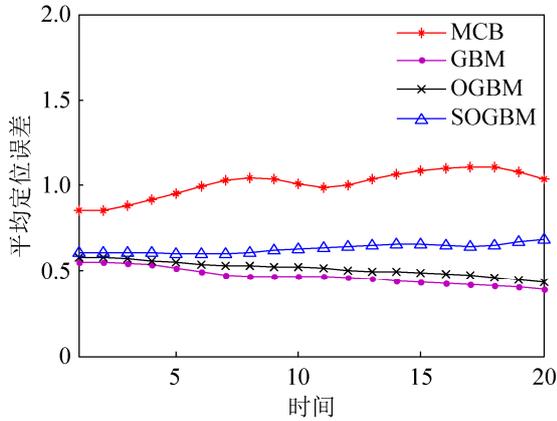


图 5 定位精度随时间变化曲线

Fig. 5 Variation of localization error curves over time

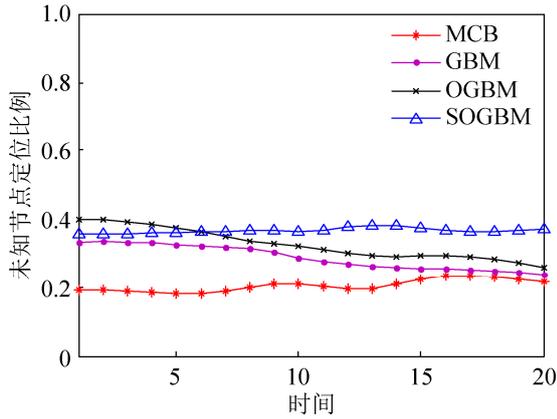


图 6 定位的未知节点比例随时间变化曲线

Fig. 6 Fraction of localized sensor nodes curves over time

3.2 锚节点数量对定位性能的影响

本节仿真参数设置如下: 未知节点数量为 100。锚节点数量从 10 开始以 10 为间隔逐渐增加到 100, 基于分区置点的算法, 每个子区域固定锚节点数量设置为总锚节点数量的百分之十。

图 7 与图 8 分别表示平均定位误差与未知节点定位数量同锚节点的数量变化之间的关系。四种算法的定位精度与未知节点定位比例均随着锚节点的数量增加而增加。

对于定位精度, S-OGBM 算法变化趋势相对

平缓, 而锚节点随机移动的 MCB 算法对锚节点数量变化最为敏感。对于未知节点定位比例曲线, S-OGBM 算法的定位数量相对最高, 并且随着锚节点数量的增多, 定位的未知节点数量优势越来越明显。这说明 S-OGBM 算法在定位性能上是优于其他三种算法的。而 OGBM 算法与 GBM 算法在定位性能上总体相差不多, 说明, 算法简化以后的 OGBM 算法对定位性能影响不大。

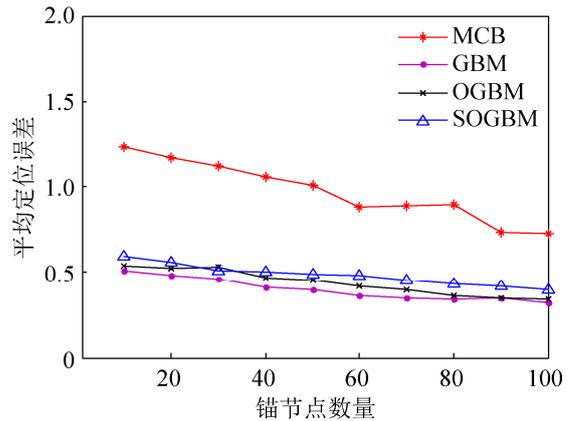


图 7 定位精度随锚节点数量变化曲线

Fig. 7 Variation of localization error with anchor number

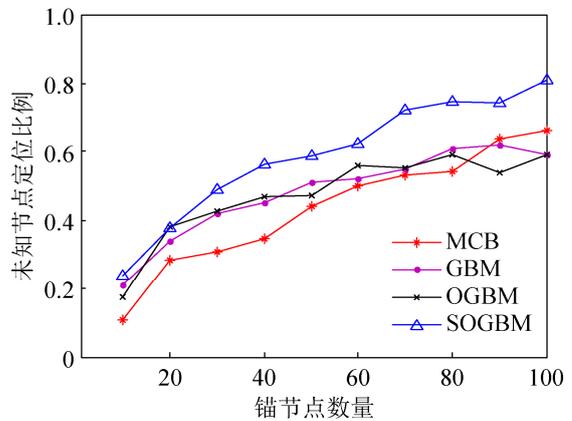


图 8 定位的未知节点比例随锚节点数量变化曲线

Fig. 8 Fraction of localized sensor nodes with varying anchor number

4 结论

本文针对现有的动态节点定位算法存在的不足, 在节点效益最大化算法的基础上提出了 OGBM 与 S-OGBM 两种改进的算法。OGBM 利用阈值代替遍历的方法减少了原有算法的复杂度, 缩

短了定位时间；S-OGBM 算法采用分区置点方法，解决了由于锚节点向未知节点密度高的区域聚集造成的监测区域覆盖率较低的问题；实验结果表明，改进的两种算法是有效的。

参考文献：

- [1] 董齐芬, 俞立, 陈友荣, 等. 移动无线传感网中的迭代蒙特卡罗定位算法研究[J]. 传感技术学报, 2010, 23(12): 1803-1809.
Dong Q F, Yu L, Chen Y R, et al. Iterative Monte Carlo Localization for Mobile Wireless Sensor Networks[J]. Chinese Journal of Sensors and Actuators, 2010, 23(12): 1803-1809.
- [2] Hu L X, Evans D. Localization for mobile sensor networks[C]// Tenth International Conference on Mobile Computing and Networking, Philadelphia, Pennsylvania, USA2004: 45-57.
- [3] Baggio A, Langendoen K. Monte carlo localization for mobile wireless sensor networks[J]. Ad Hoc Networks (S1570-8705), 2008, 6(5): 718-733.
- [4] Wang G, Yang K. A New Approach to Sensor node Localization Using RSS Measurements in Wireless Sensor Networks[J]. IEEE Transactions on Wireless Communications (S1536-1276), 2011, 10(5): 1389-1395.
- [5] Teng G, Zheng K, Dong W. MA-MCL: Mobile-Assisted Monte Carlo Localization for Wireless Sensor Networks[J]. International Journal of Distributed Sensor Networks (S1550-1477), 2011, 2011(4): 235-245.
- [6] Zhang S, Cao J, Chen L J, et al. Accurate and Energy-Efficient Range Free Localization for Mobile Sensor Networks[J]. IEEE Transactions on Mobile Computing (S1536-1233), 2010, 9(6): 897-910.
- [7] Wang Z, Wang Y, Ma M, et al. Efficient localization for mobile sensor networks based on constraint rules optimized Monte Carlo method[J]. Computer Networks (S1389-1286), 2013, 57(14): 2788-2801.
- [8] Koutsonilas D, Das S M, Hu Y C. Path Planning of mobile landmarks for localization in wireless sensor networks[J]. Computer Communications (S0140-3664), 2007, 30(13): 2577-2592.
- [9] Huang R, Zaruba G V. Monte Carlo localization of wireless sensor networks with a single mobile beacon[J]. Wireless Networks (S1022-0038), 2009, 15(8): 978-990.
- [10] Pu S, Qi R B, Jin W H, et al. Path Planning for Mobile Anchor Node in Localization Based on Ad-Hoc Localization Algorithm[J]. Journal of Donghua University (S1672-5220), 2014, 31(2): 212-216.
- [11] Kumar S, Tiwari S N, Hegde R M, et al. Optimal Anchor Guiding Algorithms for Maximal Sensor node Localization in Mobile Sensor Networks[C]// Wireless Sensor, 2013 IEEE Conference on. USA: IEEE, 2013: 13-18.
- [12] Kumar S, Tiwari S N, Hegde R M. An adaptive anchor navigation algorithm for localization in MANET[C]// Signal Processing and Communications, 2014 International Conference on. USA: IEEE, 2014.
- [6] Portilla E, Fung A, Chen W, et al. Sense And Avoid (SAA) and Traffic Collision Avoidance System (TCAS) Integration for Unmanned Aerial Systems (UAS)[C]// Proceedings of the AIAA Infotech@Aerospace Conference, Rohnert Park, California, USA, May 2007, AIAA 2007-3004.
- [7] McCalmont J, Utt J, Deschenes M. Detect And Avoid Technology Demonstration[C]// Proceedings of the AIAA's 1st Technical Conference and Workshop on Unmanned Aerospace Vehicles, Portsmouth, Virginia, USA, May 2002, AIAA 2002-3419.
- [8] Shakernia O, Chen W, Raska V. Passive Ranging for See and Avoid Applications[C]// Proceedings of the AIAA Infotech@Aerospace Conference, Arlington, Virginia, USA, September 2015, AIAA 2015-7179.
- [9] Utt J, McCalmont J, Deschenes M. Development of a Sense and Avoid System[C]// Proceedings of the AIAA Infotech@Aerospace Conference, Arlington, Virginia, USA, September 2005, AIAA 2005-7177.
- [10] Hendrickson A. AN INVESTIGATION OF ALERTING AND PRIORITIZATION CRITERIA FOR SENSE AND AVOID (SAA)[R]. AD report: ADA586852, October 2013.
- [11] Zhang Y, Fairley G. MULTIPLE-TRAJECTORY-PREDICTION (MTP) ALGORITHM FOR UAS'S SENSE AND AVOID (SAA) OPERATION[C]// Proceedings of 2013 Integrated Communications Navigation and Surveillance (ICNS) Conference, 2013.

(上接第 1884 页)