

## 移动传感器网络中定位算法的性能评测\*

张士庚<sup>1,2</sup>, 曾英佩<sup>1,2</sup>, 陈力军<sup>1,2+</sup>, 陈道蓄<sup>1,2</sup>, 谢立<sup>1,2</sup>

<sup>1</sup>(南京大学 计算机软件新技术国家重点实验室, 江苏 南京 210093)

<sup>2</sup>(南京大学/香港理工大学 无线与移动传感器网络联合实验室, 江苏 南京 210093)

### Performance Evaluation of Localization Algorithms for Mobile Sensor Networks

ZHANG Shi-Geng<sup>1,2</sup>, ZENG Ying-Pei<sup>1,2</sup>, CHEN Li-Jun<sup>1,2+</sup>, CHEN Dao-Xu<sup>1,2</sup>, XIE Li<sup>1,2</sup>

<sup>1</sup>(State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing 210093, China)

<sup>2</sup>(NJU-POLYU Cooperative Laboratory for Wireless Sensor Networks, Nanjing University, Nanjing 210093, China)

+ Corresponding author: E-mail: chenlj@nju.edu.cn

**Zhang SG, Zeng YP, Chen LJ, Chen DX, Xie L. Performance evaluation of localization algorithms for mobile sensor networks. *Journal of Software*, 2011, 22(7): 1597-1611. <http://www.jos.org.cn/1000-9825/3864.htm>**

**Abstract:** This paper makes three contributions. First, experiments have shown that simulation procedures used in existing localization algorithms for mobile sensor networks cannot output stable statistical data. This paper discusses the reasons for this and proposes a quantitative method to set up a simulation procedure that can output stable statistical data. Then, the paper evaluates and compares the accuracy of typical localization algorithms for mobile sensor networks in both obstacle-free and non-free environments. Results show that in environments with obstacles, many techniques that have been proposed, in the past, to improve localization accuracy in existing algorithms are useless and inversely decrease the algorithm's accuracy. At last, this paper proposes several metrics that can be used by a single node to evaluate the accuracy of its location estimate. Results show that the "possible maximum localization error" metric, which was proposed in previous works, performs best by indicating the accuracy of location estimate for a single node.

**Key words:** localization; mobile sensor network; performance evaluation; simulation

**摘要:** 主要有3个方面的贡献:首先,发现在已有的移动传感器网络定位算法中所使用的仿真过程不能产生稳定的性能统计数据.讨论了这种现象的原因,并且提出一种定量的方法来设置仿真过程,以使得所设置的仿真过程能够产生稳定的性能统计数据.然后,测定和比较了几种典型的移动传感器网络定位算法在无障碍物部署和有障碍物部署的环境中的性能.发现在有障碍物部署的环境中,很多已有算法中提出的用以提高定位精度的技术是无效的;相反地,它们反而会降低算法的定位精度.最后,提出了几种节点可以借以评价自身位置估计的精度度量.发现以前工作中提出的"最大可能定位误差"度量在指示单个节点的位置估计的精度时,其表现好于其他几种所提出的度量.

**关键词:** 定位;移动传感器网络;性能评价;仿真

\* 基金项目: 国家自然科学基金(60873026, 60573132, 60721002); 国家重点基础研究发展计划(973)(2006CB303000, 2009CB320705); 江苏省科技支撑计划(工业)项目(BE201078); 江苏省高校科研成果产业化推进工程项目(JH10-3); 中南大学博士后基金资助项目

收稿时间: 2009-04-02; 修改时间: 2010-01-04; 定稿时间: 2010-03-30

中图法分类号: TP393

文献标识码: A

随着传感器技术、嵌入式计算技术和低功耗无线通信技术的飞速进展,由大量的传感器节点通过自组网技术构成的大规模无线传感器网络已经开始广泛地应用于多种行业,如数据收集、环境监测、野生动物监测等<sup>[1,2]</sup>.在这些应用中,节点需要知道自己的位置来标记数据收集的地点,以帮助研究人员进行相应的数据分析.此外,在一些无线传感器网络的协议中,节点也需要知道自己的准确位置信息.比如,在基于地理位置的路由协议中,节点需要知道邻居节点的位置信息以选择下一跳的中继节点.为了解决无线传感器网络中的定位问题,已经提出了很多的定位算法<sup>[3-12]</sup>.这些定位算法中,未知自己位置的节点(称为待定位节点)借助于一些已知自己位置的节点(称为锚节点)来计算自己的位置.

大部分已经提出的定位算法仅适用于静态传感器网络.近年来,针对移动传感器网络的研究开始兴起.移动传感器网络可以用于收集野生动物的相关数据以供科学家作相应研究,比如监测斑马的生活习性(ZebraNet)<sup>[13]</sup>或者监测并控制公牛的行为<sup>[14]</sup>等.已经提出了一些专为移动传感器网络设计的定位算法<sup>[3-9]</sup>,它们大都基于序列蒙特卡罗(sequential Monte Carlo,简称 SMC)方法.由于缺少实际部署的大规模移动传感器网络,目前这些算法的定位精度都是通过仿真来进行评测的.然而,已有工作中所采用的仿真过程存在着若干缺点:首先,已有工作中所使用的仿真过程所得到的性能统计数据是不稳定的.这主要有如下两个原因:1) 已有工作的仿真过程使用的 Random Waypoint(RWP)移动模型存在着所谓“初始化问题”<sup>[15,16]</sup>.该问题是指在 RWP 移动模型中的最初几百个时间片中,节点的平均邻居数目存在着很大的变化,特别是在节点移动速度比较慢的情况下.由于大部分移动传感器网络定位算法中节点的平均定位误差与节点的平均邻居(准确地说,是待定位节点在计算自身位置的过程中所使用的邻居)数目密切相关,所以节点的平均定位误差也存在着类似的“初始化问题”,而已有的工作均没有考虑这一点;2) 已有工作没有考虑如何确定合适的仿真过程参数,如随机网络的个数和性能数据的统计区间,以得到稳定的性能统计数据.在已有工作所使用的仿真过程中,仅仅在几十个时间片上对算法的定位精度进行平均.这样的仿真过程中所得到的数据是不稳定的(即对同样的算法,使用同样的参数进行仿真,在不同仿真过程中所得到的算法的定位精度可能是不同的),因此不能准确地反映算法的真实性能.其次,已有的仿真过程仅考虑了没有障碍物的部署环境.然而,无线传感器网络的部署环境是多种多样的.当部署环境中存在障碍物时,同样的参数设置条件下,算法的定位精度与在没有障碍物的部署环境中相比会有很大不同.实际上,很多在已有算法中采用的并已被证实为有效的提高定位精度的方法,比如利用二跳锚节点邻居的信息来提高定位精度,在有障碍物的环境中根本不起作用.相反地,它们甚至会降低算法的定位精度.我们的仿真结果证实了这一点.此外,迄今为止,当在移动传感器网络中对算法的定位精度进行考察时,均是从整个网络的观点来考察,即用网络中所有节点的平均定位误差来表示算法的定位精度;节点自身并没有任何办法对自身位置估计的精度做出评估.本文讨论了单个节点如何判断自身位置估计的精度.我们给出了几种节点可以借以判断自身位置估计的精度的度量并比较了它们的效果.

本文第 1 节简单介绍文中将要考虑的几种定位算法,包括 MCL(sequential Monte Carlo localization)<sup>[3]</sup>, MCB(Monte Carlo localization boxed)<sup>[4]</sup>,MSL\*(mobile and static sensor network localization)<sup>[6]</sup>以及针对我们前期工作 WMCL(weighted MCL)<sup>[7,8]</sup>的一种改进算法.第 2 节介绍已有移动传感器网络中所使用的移动模型(RWP 移动模型的一种变体)中存在的初始化问题和该问题对移动传感器网络中定位算法的定位精度评测的影响.第 3 节首先分析不同仿真过程所得到的性能统计数据的不同是如何影响算法性能的评价和比较的,然后提出一种定量的方法来设置仿真过程以使得所设置的仿真过程能够产生稳定的性能统计数据.利用所得到的结果,第 4 节和第 5 节通过仿真分别对相关算法在无障碍物部署和有障碍物部署的环境中的性能进行评测和比较.我们发现,一些在已有算法中采用的并已在无障碍物的部署环境中被证明可以有效提高算法定位精度的技术在有障碍物的部署环境中不再有效.相反地,这些技术可能会降低算法的定位精度.第 6 节提出几种节点可以借以评价自身位置估计的精度的度量并比较它们的效果.通过比较发现,我们以前工作中提出的最大可能定位误差是一个较好的衡量标准.最后,第 7 节对文章进行总结并指出将来的研究方向.

## 1 背景和相关工作

### 1.1 移动传感器网络中的定位算法简介

考虑一个锚节点和待定位节点都可以移动的移动传感器网络,并且假设每个节点都知道自己的最大移动速度  $v_{\max}$ (用节点每个时间片内可以移动的最大距离来表示.例如,记  $r$  为节点的通信半径,则  $v_{\max}=0.2r$  表示节点每个时间片内可以移动的最大距离为 0.2 个通信半径).假设时间被划分为离散的时间片  $t=0,1,2,\dots$ ;在每个时间片内每个待定位节点都重新计算自己的位置.一个待定位节点在各个时间片内的位置可以用一个马尔可夫过程  $\{l_t, t=0,1,2,\dots\}$  来描述.其初始状态  $l_0$  的分布由节点的初始位置分布决定,状态转移概率  $p(l_t|l_{t-1})$  由节点的移动模型决定.在每个时间片内,一个待定位节点可能会观测到若干个其他节点;记节点在时间片  $t$  内观测到的其他节点的集合为  $o_t$ ,则  $l_t$  的分布由以下公式决定:

$$\begin{aligned} & p(l_0), \\ & p(l_t|l_{t-1}), \text{ for } t \geq 1, \\ & p(o_t|l_t), \text{ for } t \geq 1. \end{aligned}$$

我们的目的是计算  $p(l_t|o_t, \dots, o_1)$ ,然而如文献[3]中所述,一般情况下很难得到  $l_t$  分布的解析表达式.序列蒙特卡罗方法提供了一种易于实现的基于仿真的方法来得到  $l_t$  分布的一些样本.在序列蒙特卡罗方法中,我们使用  $N$  个带权样本来近似表示  $l_t$  的分布  $p(l_t|o_t, \dots, o_1) \approx \{(l_t^{(i)}, w_t^{(i)})\}_{i=1, \dots, N}$ ,其中,  $w_t^{(i)} = p(o_t|l_t^{(i)})$  表示样本  $l_t^{(i)}$  对应的权值.通过计算这  $N$  个样本的均值,我们可以估计  $l_t$  的期望并以此作为节点在时间片  $t$  内的位置估计  $l_{est}$ .

Hu 和 Evans 提出了第一个专为移动传感器网络设计的定位算法 MCL<sup>[3]</sup>.该算法直接应用了 SMC 方法,使用一个待定位节点两跳之内的锚节点邻居来计算待定位节点的位置.MCL 算法存在着一些不足,如仅仅能够依靠增加锚节点的密度来提高定位精度、采样效率低下等.针对这些问题,一些研究人员提出了对 MCL 的改进算法,典型的有 Rudafshani 和 Datta 提出的 MSLS\*算法<sup>[6]</sup>、Baggio 和 Langendoen 提出的 MCB 算法<sup>[4]</sup>以及我们在前期工作中所提出的 WMCL 算法<sup>[7,8]</sup>等.这几种算法的具体描述均可在相关文献中找到,这里我们仅简要介绍一下其不同之处.

MCL 算法将移动传感器网络中的定位问题抽象为节点位置的后验概率估计问题,并利用序列蒙特卡罗方法来对这个问题进行求解.MCL 算法存在着两个不足:(1) 采样效率较低(特别是当节点移动速度较大时);(2) 仅仅依靠提高锚节点密度来提高定位精度.MCB 算法针对 MCL 算法的第 1 个不足进行了改进,利用一种称为 Bounding-box 的技术来缩小采样空间,以提高采样效率.MSLS\*算法针对 MCL 算法的第 2 个不足进行了改进,提出了两种利用待定位节点邻居的估计位置来提高定位精度的方法.但是,该算法中所使用的方法会扩大采样空间,从而降低了采样效率.为了不显著降低采样效率,MSLS\*算法采用了与其他算法不同的重采样策略.然而,MSLS\*算法在节点移动速度较大时定位精度很低,甚至比 MCL 还要低.在我们以前的工作 WMCL 算法中,同时改进了 MCB 算法中所使用的 Bounding-box 方法和 MSLS\*算法中所使用的提高定位精度的方法.WMCL 算法可以同时达到提高采样效率和提高定位精度的效果.在我们前期工作所提出的 WMCL 算法中,节点仅在构建好的 Bounding-box 中进行采样.后来的研究发现,在节点移动速度较慢时,利用节点上一个时间片的样本信息来进行采样,可以进一步提高定位精度.所以在本文中,我们对 WMCL 算法的采样过程进行了改进:当节点移动速度不大于  $0.2r$ ( $r$  为节点的通信半径)时,使用 MCB<sup>[4]</sup>提出的采样算法;当节点移动速度大于  $0.2r$  时才使用 WMCL 自身的采样算法,并将这种改进算法称作 EMCL(enhanced WMCL).具体的算法描述可参考文献[4,8].

### 1.2 移动传感器网络定位算法的误差分析

本文所考虑的几种定位算法都属于基于邻接关系的定位算法.在这类算法中,一个待定位节点的位置估计仅仅依赖于它所观察到的邻居节点,即  $o_t$ .Nagpal 等人提出了一种方法来估计这类算法中的节点平均定位误差的下界<sup>[12]</sup>,其思想如下所述:考虑两个待定位节点  $P$  和  $Q$ ,如果它们所观察到的邻接关系是相同的,即  $o_t^P = o_t^Q$ ,则  $P$  和  $Q$  的位置估计结果应该是相同的.所以在这类算法中,一个节点在不影响自己的邻接关系的情况下,平均

可以移动的最大距离就给出了节点平均定位误差的下界.假设节点在部署区域中是均匀分布的,则这类算法中节点的平均定位误差主要取决于网络中节点的密度.记网络中一个节点通信范围内平均的锚节点邻居数和邻居数(包括锚节点和待定位节点)分别为  $s_d$  和  $n_d$ .在 MCL 算法和 MCB 算法中,  $o_i$  仅包括两跳之内的锚节点邻居.利用这种方法,文献[3]推导出这两种算法中节点平均定位误差的下界为  $\pi r/18s_d$ .在 MSL\*算法和 EMCL 算法中,  $o_i$  中包括了二跳之内的锚节点邻居和一跳之内的待定位节点邻居,其定位精度的下界为  $\pi r/(8s_d+4n_d)$ ,具体的推导过程可参见文献[3,8].

从理论上来说,节点的最大移动速度  $v_{\max}$  并不会影响算法的定位精度.然而,如同我们在前期工作中所指出的<sup>[8]</sup>,  $v_{\max}$  会影响算法选取到的有效样本的质量,从而影响算法的定位精度.所以在比较各种算法的定位精度时,我们也考虑节点移动速度的影响.

## 2 已有仿真过程存在的问题

在我们实现各种算法并对它们的定位精度进行比较的过程中发现,在不同的仿真过程中,对同一算法使用同样的参数进行仿真,所得到的性能统计数据有时候会存在着很大的不同.在已有的研究工作中也存在着类似的情况.比如,其他参数设置相同的情况下,在文献[3]中给出的当锚节点密度为 3 时 MCL 的定位精度已经小于  $0.2r$ ,而在文献[6]中给出的当锚节点密度为 3 时 MCL 的定位精度约为  $0.3r$ .当不同的仿真过程中所得到的算法性能统计数据不一致时,对算法性能的评价和比较就不够准确.我们发现,这种数据的不一致性是由于仿真过程的不合理设置所导致的.在这一节,我们讨论在仿真移动传感器网络定位算法时,仿真过程的不同设置对所得到的算法性能统计数据的影响.

### 2.1 RWP移动模型的初始化问题

在已有的工作中,仿真基于序列蒙特卡罗方法的移动传感器网络定位算法<sup>[3-9]</sup>时所采用的移动模型均是在文献[3]中提出的 RWP 移动模型的变体,即在 RWP 模型中,在每个时间片内重新选择节点的移动速度以防止节点的速度退化<sup>[16]</sup>.然而,如同文献[15]中所指出的,RWP 移动模型中存在着“初始化问题(initialization problem)”.该问题是指在仿真的最初几百个时间片内,节点的平均邻居数的变化情况很大,特别是在节点移动速度比较慢的情况下.在文献[3]中提出的 RWP 移动模型的变体中同样存在着类似的问题.而根据我们在第 1.2 节所讨论的,已有移动传感器网络定位算法的定位精度与节点的平均邻居数目密切相关.在已有工作所采用的仿真过程中,通常仅在几十个时间片上对性能指标进行平均.这样所得到的算法性能统计数据不够稳定(即在不同的仿真过程中,对同一算法使用同样的算法参数设置进行仿真,所得性能统计可能有很大不同),因此不能准确地对算法的性能进行评价和比较.

图 1 给出了在一个采用文献[3]中提出的移动模型的网络中,在每个时间片内,节点平均观察到的锚节点邻居数目和待定位节点邻居数目的变化情况.可以看出,如同文献[15]中所指出的,在整个仿真过程中,节点平均观察到的锚节点邻居数目和待定位节点邻居数目变化很大.图 2 给出了在 50 个随机产生的网络中,采用文献[5]中提出的移动模型进行仿真,所得出的在每个时间片内节点平均观察到的锚节点邻居数目和待定位节点邻居数目的变化情况.可以看出,初始化问题仍然存在.同时,我们还可以看出,节点的移动速度越快,每个时间片内观察到的平均锚节点(节点)邻居数目就会越快地趋于稳定.当节点最大移动速度为  $0.2r$  时,节点的平均锚节点数目很快地就会趋于稳定.然而,当节点的移动速度比较慢时,节点的平均锚节点数目需要通过很长一段之后才趋于稳定.如图 2 所示,当节点的移动速度为  $0.05r$  时,大概 200 个时间片之后,节点的平均锚节点邻居数目才趋于稳定.当节点的移动速度为  $0.02r$  时,节点的平均锚节点邻居数目在大概 500 个时间片之后才趋于稳定.

由于已有的移动传感器网络定位算法的定位精度与节点的平均锚节点邻居数目密切相关,节点的平均定位误差在整个仿真过程中也会存在很大的变化.图 3 给出了在一个随机产生的网络中执行 MCL 算法,在每一个时间片内节点的平均定位误差情况.对照图 1,可以清楚地看到,节点的平均定位误差与平均锚节点邻居数目密切相关:当平均锚节点邻居数目大时,平均定位误差就小;当平均锚节点邻居数目小时,平均定位误差就大.所以,对应于平均节点邻居数目的“初始化问题”,节点的平均定位误差也存在同样的问题.图 4 给出了 MCL 算法在

50 个随机产生的网络中、每一个时间片内节点的平均定位误差情况.可以看到,与图 2 对应,节点的平均定位误差趋向于稳定所需要的时间片数与节点的最大移动速度有关.与节点的平均锚节点邻居数目类似,当节点的移动速度较大时,节点的平均定位误差很快趋于稳定.当节点的移动速度较小时,节点的平均定位误差在很长一段时间后才会趋于稳定.比如,当节点的最大移动速度分别为  $0.05r$  和  $0.02r$  时,节点的平均定位误差分别在大概 200 个时间片和 500 个时间片之后才趋于稳定.

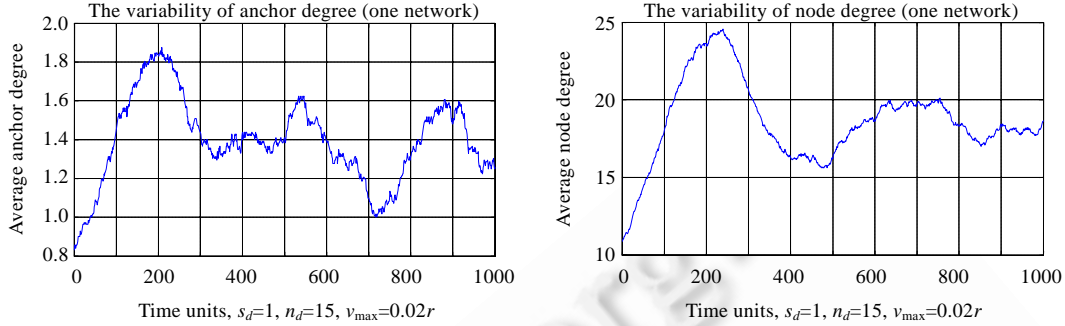


Fig.1 Average number of anchor/node neighbors in each time unit (one network)

图 1 在每个时间片内节点平均观察到的锚节点/节点邻居数目(一个网络)

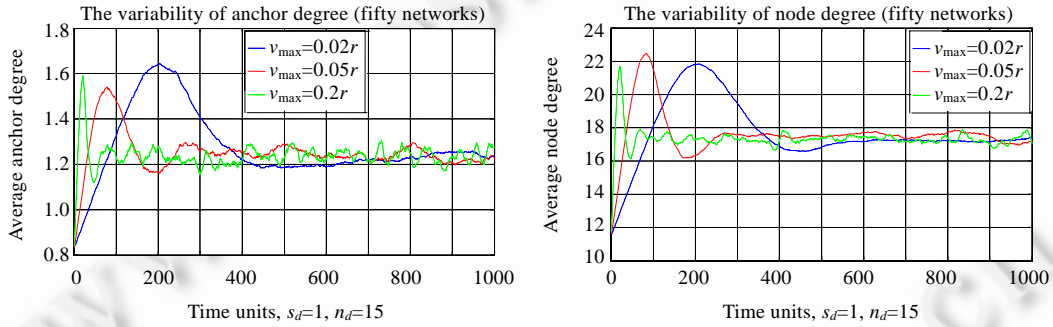


Fig.2 Average number of anchor/node neighbors in each time unit (fifty networks)

图 2 在每个时间片内节点平均观察到的锚节点邻居数目和节点邻居数目(50 个网络平均)

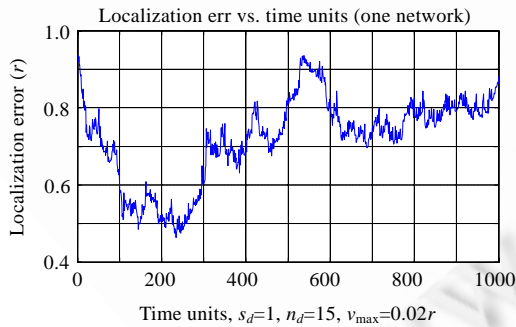


Fig.3 Localization error vs. time unit (one network)

图 3 一个网络中节点定位误差的变化情况

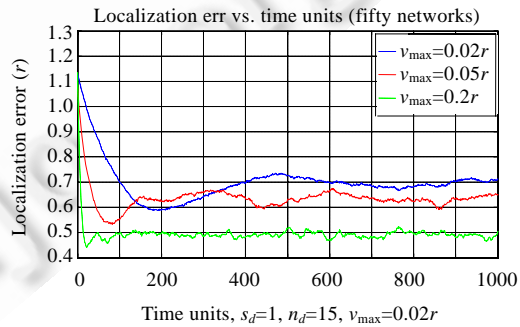


Fig.4 Localization error vs. time unit (fifty networks)

图 4 50 个网络中节点的平均定位误差变化情况

2.2 仿真过程对算法性能评价和比较的影响

表 1 列出了以前工作中所使用的仿真过程中的随机网络数目和在每个网络中对定位精度进行统计的时间

片区间.需要说明的是,这些文献并没有解释为何选择这样的参数.

**Table 1** Parameters used in existing simulation procedures  
表 1 已有仿真过程中使用的参数

Algorithm name	Number of networks	Number of simulated time steps	Statistical interval
MCL <sup>[3]</sup>	10	80	51~80
MCB <sup>[4]</sup>	20	200	101~200
MSL <sup>*[6]</sup>	50	200	36~200
WMCL <sup>[10]</sup>	30	100	51~100

我们从同一算法的性能评价和不同算法的性能比较两个方面,来说明仿真过程对移动传感器网络定位算法的性能评测和比较的影响.首先,在不同的仿真过程中以同样的参数运行 MCL 算法,来考察不同的仿真过程对同一算法定位精度的统计数据的影响.两个仿真过程分别设置如下:1) 部署区域为  $10r \times 10r$ ,产生 10 个网络,每个网络仿真 80 个时间片,在每个网络中,在时间片 51~80 上对节点的定位误差进行平均(即文献[3]中使用的仿真过程设置);2) 部署区域为  $5r \times 5r$ ,产生 10 个网络,每个网络仿真 215 个时间片,在每个网络中,在时间片 185~215 上对节点的定位误差进行平均.所使用的仿真程序是由文献[3]的作者提供的在移动传感器网络定位算法仿真中常用的仿真程序.图 5 给出,当节点最大移动速度发生变化时,MCL 算法在这两个仿真过程中所得到的平均定位误差的数据.可以看出,即使对同一种算法,用同样的参数设置,不同的仿真过程所得到的性能统计数据也相差很大.例如,当节点移动速度较慢时,在同样的参数设置下,同一种算法在不同仿真过程中所得到的性能统计数据甚至会相差超过  $0.2r$ .

其次,我们考察不同的仿真过程对算法性能的比较造成的影响.在前述两个仿真过程中同时仿真了 MCB 算法和 MCL 算法.图 6 给出,在这两个仿真过程中,当节点的最大移动速度发生变化时,相对于 MCL 算法,MCB 算法的定位精度改进幅度的变化情况.从图中可以看出,在不同的仿真过程中,MCB 算法相比 MCL 算法的改进情况相差很大.当节点速度较小时,在不同仿真过程中所得改进幅度的数值甚至相差超过 20%.有些工作中给出的定位精度的改进幅度甚至小于 10%<sup>[4,5]</sup>,这样的数据并不能可靠地反映算法的比较结果.

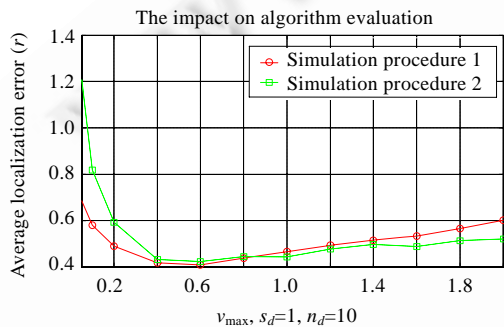


Fig.5 Impact of the setup of simulation procedures on the evaluation of MCL

图 5 仿真过程的设置对 MCL 性能评价的影响

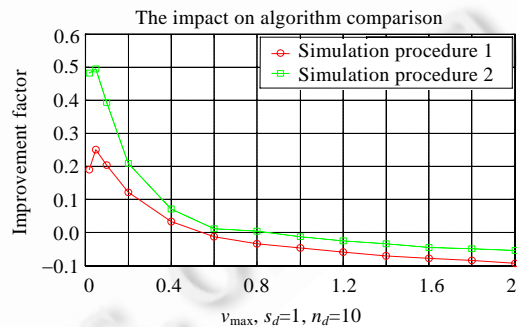


Fig.6 Impact of the setup of simulation procedures on the comparison of MCB and MCL

图 6 仿真过程的设置对 MCL 和 MCB 性能比较的影响

### 3 设置合理的仿真过程参数

在这一节,我们讨论如何设置仿真过程以使所得到的算法性能统计数据是稳定的.首先讨论“稳定”的性能统计数据的含义,然后分两步来确定仿真过程中所使用的参数,以获取稳定的性能统计数据.第 1 步,确定合适的网络数目,使得对每一个时间片来说,对这些网络中所得到的数据进行平均后的值比较稳定.由于节点的初始位置是随机分布的,对多个网络所得的数据进行平均可以消除不同的随机网络造成的影响,从而能够较为准确地反映算法的性能随着时间片变化的情况.第 2 步,计算在不同长度的统计区间上得到的性能统计数据的差异.

当得到的统计数据的差异足够小时,我们就认为这时候所得到的数据能够较为准确地反映算法的平均性能.这两个步骤都是必须的.单纯地增加网络数目不能消除算法性能随时间变化的影响,而单纯地扩大统计区间也不能消除(节点初始位置分布)不同的随机网络带来的影响.根据图 3,当节点的移动速度较小时,初始化问题存在的时间比较长,算法的定位精度趋于稳定状态所需要的时间片数比较多;而当节点的移动速度较大时,算法的定位精度趋于稳定状态所需要的时间片数比较少.所以,我们在节点移动速度较小的情况下( $0.02r$ )来确定这些参数.我们的仿真数据表明,这样所获得的参数在节点移动速度较大时同样适用.在所考虑的算法中,MCL 算法是收敛得最慢的,所以我们使用 MCL 算法来确定这些参数.仿真结果表明,在利用所得参数设置的仿真过程中,其他算法的性能统计数据也是稳定的.

### 3.1 稳定的性能统计数据

首先,我们讨论什么样的性能统计数据才是稳定的.当在两个仿真过程中对同一算法进行仿真所得的定位精度的差值小于一个给定的阈值时,则称这两个仿真过程中得到的性能统计数据是稳定的.设有算法  $A$ ,假设在两个仿真过程——仿真过程 1 和仿真过程 2——中所得  $A$  的定位精度分别为  $a_{A,1}$  和  $a_{A,2}$ ,则当  $|a_{A,1}-a_{A,2}| \leq d$  时,称这两个仿真过程所得到的性能统计数据都是稳定的.根据不同的精度要求, $d$  可以取不同的值.

下面我们从不同算法性能比较的角度来看应该如何选取合适的阈值  $d$ .假设原有原始算法  $A$  和改进算法  $B$ .假设在两个不同的仿真过程中得到的两种算法的平均定位误差分别为  $a_{A,1}, a_{B,1}$  和  $a_{A,2}, a_{B,2}$ .假设算法  $B$  对于算法  $A$  的改进是有效的,即有  $a_{A,1} > a_{B,1}, a_{A,2} > a_{B,2}$ (表示在每个仿真过程中算法  $B$  的定位误差都要小于算法  $A$  的定位误差).那么,在两个仿真过程中,算法  $B$  相比于算法  $A$  在定位精度上的改进幅度(即定位误差的减小幅度)分别为

$$\sigma_1 = \frac{a_{A,1} - a_{B,1}}{a_{A,1}} = 1 - \frac{a_{B,1}}{a_{A,1}}, \quad \sigma_2 = \frac{a_{A,2} - a_{B,2}}{a_{A,2}} = 1 - \frac{a_{B,2}}{a_{A,2}}.$$

在两个仿真过程中所得的改进幅度的差为

$$\sigma = \sigma_1 - \sigma_2 = \frac{a_{B,2}}{a_{A,2}} - \frac{a_{B,1}}{a_{A,1}}.$$

假设两个仿真过程中所得算法性能统计数据都是稳定的,即有  $|a_{A,2}-a_{A,1}| \leq d, |a_{B,2}-a_{B,1}| \leq d$ ,则我们有(由于  $d$  的值很小,对小的  $k \in N$ ,一般情况下,有  $a_{A,1}-kd > 0, a_{B,1}-kd > 0$ .我们假设至少当  $k=1$  时这种关系成立):

$$\frac{a_{B,1} - d}{a_{A,1} + d} \leq \frac{a_{B,2}}{a_{A,2}} \leq \frac{a_{B,1} + d}{a_{A,1} - d} \quad (1)$$

如果算法的改进是有效的,由于  $d$  的值很小,可以假设  $a_{A,1}-d > a_{B,1}+d$ .注意到,当  $0 < d < B < A$  时,我们有  $\frac{B-d}{A-d} \leq \frac{B}{A} \leq \frac{B+d}{A+d}$ .利用公式(1)中左边的不等式,有

$$\frac{a_{B,2}}{a_{A,2}} - \frac{a_{B,1}}{a_{A,1}} \geq \frac{a_{B,1} - d}{a_{A,1} + d} - \frac{a_{B,1}}{a_{A,1}} \geq \frac{a_{B,1} - d - d}{a_{A,1} + d - d} - \frac{a_{B,1}}{a_{A,1}} = \frac{-2d}{a_{A,1}}.$$

利用公式(1)中右边的不等式,则有

$$\frac{a_{B,2}}{a_{A,2}} - \frac{a_{B,1}}{a_{A,1}} \leq \frac{a_{B,1} + d}{a_{A,1} - d} - \frac{a_{B,1}}{a_{A,1}} \leq \frac{a_{B,1} + d + d}{a_{A,1} - d + d} - \frac{a_{B,1}}{a_{A,1}} = \frac{2d}{a_{A,1}}.$$

所以,在两个仿真过程中所得的算法  $B$  相对算法  $A$  的改进幅度的差值为

$$-\frac{2d}{a_{A,1}} \leq \sigma \leq \frac{2d}{a_{A,1}} \quad (2)$$

公式(2)给出了  $d$  和  $\sigma$  之间的关系,利用这个关系,我们可以选取合适的  $d$  值以使  $\sigma$  足够小.本文中所涉及到的定位算法的定位误差(当节点移动速度比较慢时)都大于  $0.2r$ ,所以这里我们取  $d=0.01r$ .这样可以保证  $|\sigma| \leq 10\%$ ,即在两个不同的仿真过程中所得的算法  $B$  相比于算法  $A$  的定位精度的改进幅度的差值小于  $10\%$ .这样,就可以保证图 6 所示的情况不会出现.

### 3.2 网络数目

我们首先考虑使用不同数目的随机网络进行仿真对所得算法性能统计数据的影响.当节点的最大移动速度分别为  $0.02r$  和  $0.05r$  时,分别产生了 200 个随机网络,并在图 7 中给出了对不同数目的随机网络中每一个时间片内节点的平均定位误差进行平均所得到的结果.为了尽可能地避免初始化问题的发生,对每个网络我们都进行了足够长时间的仿真,每个网络中均仿真了 3 000 个时间片.由图 7 可以看出:当所使用的随机网络的数目较少时,比如仅使用 10 个随机网络,所得到的性能统计数据并不稳定,仍然会有较大的抖动;而当采用 40 个随机网络进行平均时,所得到的数据与采用 200 个网络进行平均所得的数据已经非常接近.

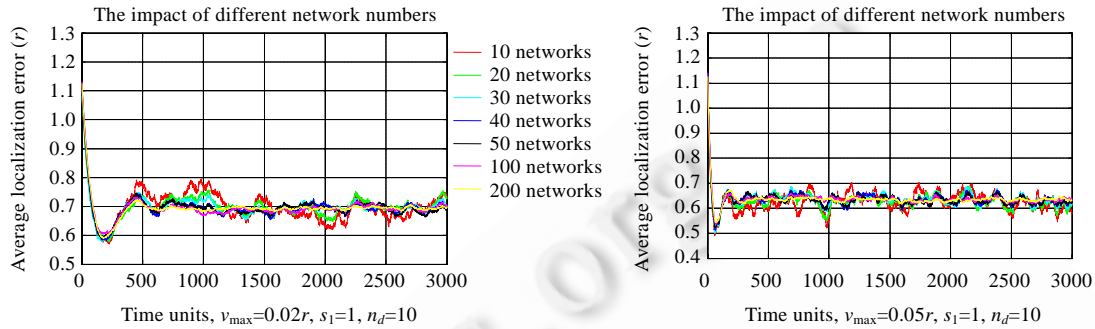


Fig.7 Impact of the number of random networks on the stability of obtained data

图 7 随机网络的数目对所获得数据的稳定性的影响

这里我们给出一种方法来定量地比较两个不同的仿真过程所得到的算法性能统计数据的不同.假设有两个仿真过程  $A$  和  $B$ ,所采用的随机网络的个数分别为  $m$  和  $n$ ,每个网络中仿真的时间片数为  $T$ .以两个向量  $V_A$  和  $V_B$  表示两个仿真过程中所得到的性能统计数据(每一个时间片内节点的平均定位误差).定义这两个仿真过程所获取的性能统计数据的差值如下:

$$diff(A, B) = \frac{\sum_{i=1}^T |V_A(i) - V_B(i)|}{T}$$

其中,  $V_A(i)$  ( $V_B(i)$ ) 表示向量  $V_A$  ( $V_B$ ) 的第  $i$  个分量.当两个仿真过程所获取的平均数据的差值小于一定阈值时,就认为仿真过程所获取的数据已经比较稳定,继续增加随机网络的个数并不能较大地影响所获取数据的值.

图 8 给出,当  $2 \leq k \leq 20$  时,分别使用前述仿真过程中的前  $10k$  个网络和前  $10(k-1)$  个网络算出的性能统计数据的差值变化情况.可以看出,使用 40 个随机网络所获得的数据与使用 30 个网络所获得的数据的差值已经小于  $0.01r$ .这意味着使用 40 个随机网络的仿真过程已经能够获得比较稳定的性能统计数据.

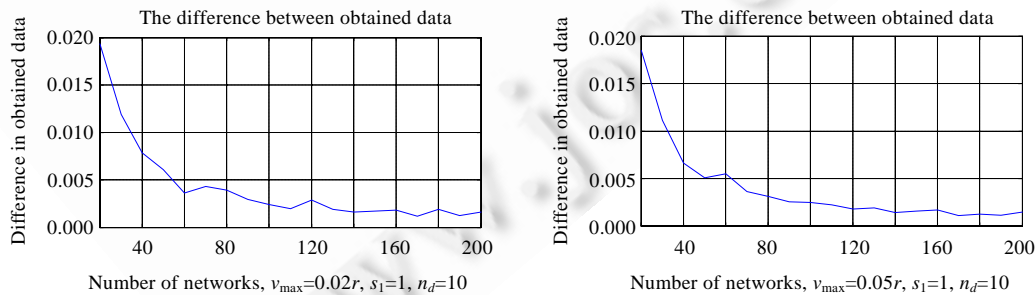


Fig.8 Difference between obtained data from two simulation procedures with  $10k$  networks and with  $10(k-1)$  networks respectively ( $k=2,3,\dots,20$ )

图 8 使用  $10k$  个网络的仿真过程和使用  $10(k-1)$  个网络的仿真过程所得数据的差值( $k=2,3,\dots,20$ )



为了验证这个结论,图 9 给出,当  $4 \leq i, j \leq 20$  时,在两个相互独立的使用  $10i$  个随机网络和使用  $10j$  个随机网络的仿真过程中所得到的性能统计数据之差值.由图 9 可以看出,使用 40 个网络的仿真过程和使用 200 个网络的仿真过程所得到的性能统计数据稍稍大于  $0.01r$ .我们推测,这是因为在前一个实验中,仿真过程中的随机网络不是相互独立的,使得所估计的最少网络个数稍稍少于实际所需的网络个数.图 10 给出,  $5 \leq i, j \leq 20$  时,在两个相互独立的使用  $10i$  个随机网络和使用  $10j$  个随机网络的仿真过程中所得到的性能统计数据的差值.可以看出,在所有仿真过程中,所得到的统计数据的差值均已小于  $0.01r$ .这说明,使用 50 个随机网络的仿真过程所得到的性能统计数据已经相当稳定.在接下来的实验中,仿真过程中我们所使用的随机网络的个数为 50.

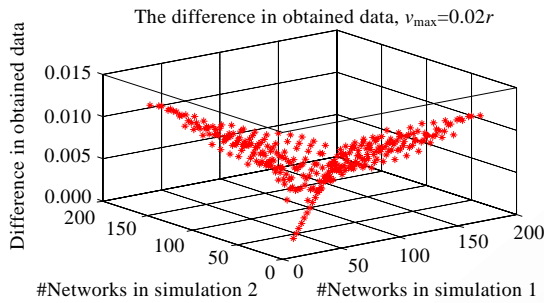


Fig.9 Difference between obtained statistical data in two independent simulation procedures ( $4 \leq i, j \leq 20$ )  
图 9 两个独立的仿真过程中所得到的性能统计数据的差值( $4 \leq i, j \leq 20$ )

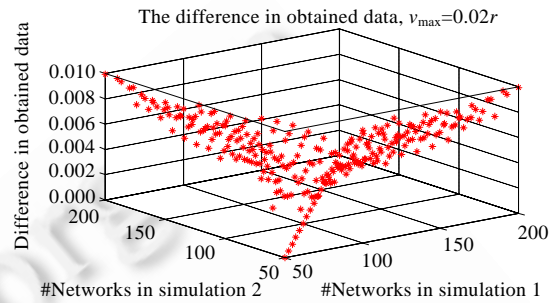


Fig.10 Difference between obtained statistical data in two independent simulation procedures ( $5 \leq i, j \leq 20$ )  
图 10 两个独立的仿真过程中所得到的性能统计数据的差值( $5 \leq i, j \leq 20$ )

### 3.3 统计区间

下一步我们确定合适的统计区间,使得不同仿真过程中所得到的性能统计数据的差值小于阈值  $0.01r$ .首先,为了避免 RWP 模型中的初始化问题的发生,我们从 600 个时间片之后才开始统计数据(根据文献[15]和图 4 中给出的数据,我们认为在 600 个时间片之后统计足以避免 RWP 移动模型中初始化问题的发生).图 11 中给出了在 600 个时间片之后,当统计区间的长度分别为 50,100,200,300,400,600,800,1 200 时得到的统计数据.可以看出:当  $v_{\max}=0.02r$ ,统计区间的长度分别为 100,200,400 时,对不同的区间进行统计所得数据的最大差值分别为  $0.055r, 0.043r, 0.023r$ ;而当统计区间的长度为 800 时,对不同的区间进行统计所得到的数据最大差值为  $0.007r$ ,已经小于  $0.01r$ .当  $v_{\max}=0.2r$  时有类似的结果.

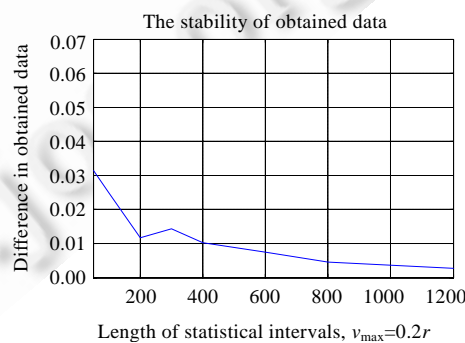
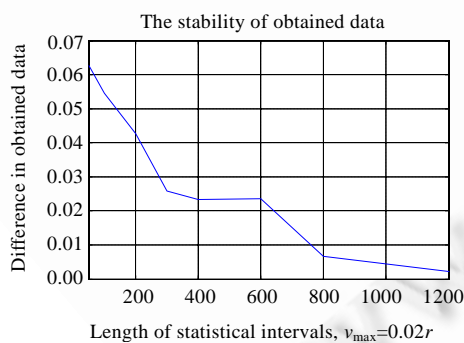


Fig.11 Impact of different statistical intervals,  $v_{\max}=0.02r$  and  $v_{\max}=0.2r$   
图 11 统计区间的长度对性能统计数据的影响,  $v_{\max}=0.02r$  和  $v_{\max}=0.2r$

由以上讨论可知,为了得到稳定的统计数据,至少要随机生成 50 个网络,然后在避过初始化问题(600 个时间

片)之后对至少 800 个时间片上的数据进行统计.我们对节点以其他速度进行移动的情况进行的分析表明,这样的仿真设置可以保证我们在不同的仿真过程中得到的统计数据之间的差值小于  $0.01r$ .然而实际上,当节点移动速度较快时,以较短的统计区间进行统计也可以得到稳定的统计数据.如图 11 所示,当节点最大移动速度为  $0.2r$  时,对 400 个时间片上的数据进行统计已经能够获得较为稳定的数据.在我们接下来的仿真中,当移动速度为  $0.02r$  时,采用的是长度为 800 个时间片的统计区间;其他情况下,采用的是长度为 400 个时间片的统计区间,即从第 601 个时间片统计到第 1000 个时间片.仿真过程中使用的随机网络的个数是 50 个.

#### 4 在无障碍物部署环境中的性能比较

根据在第 1 节所得到的结果,我们对几个移动传感器网络中的典型定位算法在没有障碍物环境中的性能进行了评测和比较.考虑的算法包括 MCL,MCB,MSL\*以及第 1 节所提到的 EMCL.考虑的主要参数是锚节点度数  $s_d$ 、节点度数  $n_d$  和节点最大移动速度  $v_{\max}$ .所得数据如图 12 所示.

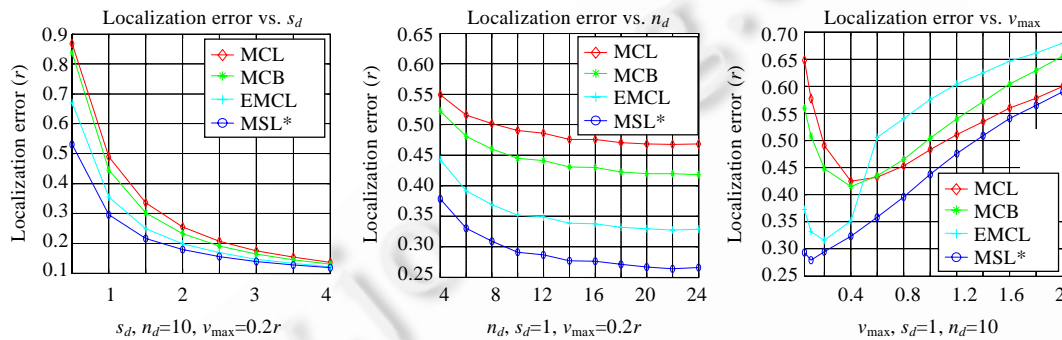


Fig.12 Effects of  $s_d$ ,  $n_d$  and  $v_{\max}$  on localization accuracy

图 12  $s_d$ ,  $n_d$  和  $v_{\max}$  对定位精度的影响

可以看到,当  $s_d$  增加时,MCB,MSL\*和 EMCL 相比于 MCL 的定位精度均有所提高.MCB 的改进效果最小,在 4%~10%之间,平均为 7%;MSL\*的改进效果最大,在 13%~40%之间,平均为 27%;而 EMCL 的改进效果在两者之间,即在 11%~27%之间,平均为 20%.当  $n_d$  增加时,相比于 MCL 算法,MCB 的改进效果在 5%~10%之间,平均为 9%;MSL\*的改进效果在 31%~44%之间,平均为 40%;EMCL 的改进效果在 19%~30%之间,平均为 28%.我们还可以看到,随着  $n_d$  的增加,MSL\*和 EMCL 改进的幅度越来越大.当  $v_{\max}$  增加时,可以看到,在某些情况下,3 种改进算法的精度甚至不如 MCL.3 种定位算法的精度提高幅度分别为:MCB, -9%~14%,平均为 0.0%;MSL\*, 2%~55%,平均为 19%;EMCL, -19%~43%,平均为 0.0%.文献[4]指出,当  $v_{\max}$  变化时,MCB 相对于 MCL 的精度提高幅度为 4%~73%之间,平均为 30%.而在我们的实验中,数据并不是这样.我们猜测可能有如下几个原因:1) 在文献[4]所采用的移动模型中,限制了节点移动的最小速度,而这会降低 MCL 的定位精度(我们的仿真结果证实了这一点)并提高 MCB 的定位精度;2) 在文献[6]中所采用的通信半径为 100m,而在 MCL 原来仿真中所采用的通信半径为 50m.这样,原来 MCL 仿真程序中所使用的松弛参数  $\alpha$  的值应该由 5( $0.1r$ , 当  $r=50m$  时)改成 10,仍然使用 5 会降低 MCL 的定位精度.总之,与 MCL 相比,MCB,MSL\*和 EMCL 的定位精度分别提高了 5%,29%和 16%.

#### 5 在有障碍物部署环境中的性能比较

在有障碍物部署的环境中,已有算法的定位精度会受到很大影响.如图 13 所示,节点 A 和节点 B 是锚节点,而节点 C 是待定位节点.节点 A 和节点 C 之间的距离小于通信半径  $r$ ,在无障碍物的部署环境中,由 UDG(unit disk graph)通信模型可知,节点 C 应该是节点 A 的一跳邻居.然而,由于存在障碍物,节点 C 和节点 A 不能直接通信.由于节点 B 可直接与节点 A 和节点 C 通信,于是有节点 A 是节点 C 的二跳锚节点邻居.这样,C 就认为自己

的位置落在白色的圆外(而  $C$  的实际位置在白色的圆内),从而将自己的位置估计错误,造成较大的定位误差.文献[4]中也提到了类似的场景,但是并没有给出在这种情况下各种算法的定位性能所受到的影响.

为了评测和比较已有定位算法在有障碍物的部署环境中的性能,我们修改了 RWP 移动模型,并在这种模型下对这 4 种算法进行了仿真,以考察在有障碍物部署的环境中它们的定位精度是如何受到影响的.我们对 RWP 移动模型的修改如图 14 所示.在原始的 RWP 移动模型中,节点到达目的地后随机地在部署区域内选择一个点作为自己的新目的地.在有障碍物的情况下,节点不能穿过障碍物而必须绕过障碍物.这里,我们如下计算节点为绕过障碍物所选取的路径,如图 14 所示.节点  $A$  和节点  $B$  选择的目的地分别是  $A'$  和  $B'$ .由于从  $A$  移动到  $A'$  和从  $B$  移动到  $B'$  的路径上并没有障碍物, $A$  和  $B$  的目的地仍然分别是  $A'$  和  $B'$ .现在来看节点  $C$ . $C$  选择的目标位置为  $C'$ ,由于在  $C$  到  $C'$  的路径上存在着障碍物,我们首先计算在线段  $CC'$  上而不越过障碍物的距障碍物最近的点  $C_1$  作为  $C$  的第 1 个中转点.然后,我们计算  $C_1$  所靠近的障碍物的两个端点中距目的地  $C'$  较近的那个点  $C_2$  作为下一个中转点.如果  $C_2$  和  $C'$  之间没有其他障碍物,则将  $C'$  作为最后的目的地;否则,假设  $C_2$  为起始点继续计算剩下的路径.在图 14 中,从  $C$  到  $C'$  的路径上存在着一个障碍物,计算出的路径为  $C-C_1-C_2-C'$ ;从  $D$  到  $D'$  的路径上存在着两个障碍物,计算出来的路径为  $D-D_1-D_2-D_3-D'$ .

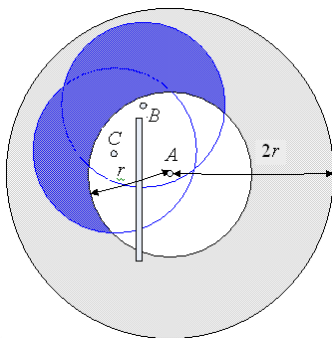


Fig.13 Connectivity status with an obstruct

图 13 当有障碍物存在时的通信情况

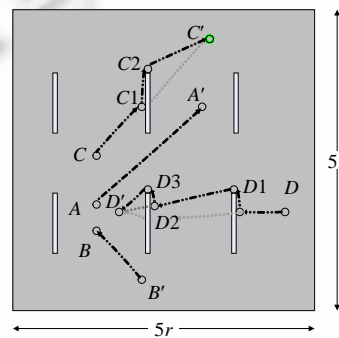


Fig.14 Simulation scenario

图 14 仿真场景

图 14 给出了仿真所使用的场景,图中每个障碍物的长度为 100m,宽度为 2m.我们计算了在同样的参数设置情况下,所考察的算法在有障碍物部署情况下的平均定位误差相比于在没有障碍物部署情况下的平均定位误差的增加情况,所得到的数据如图 15 所示.可以看出,当锚节点的密度增加时,各种算法的平均定位误差都会大幅度增加.当锚节点密度增大时,受图 13 所示情况的影响,相比于没有障碍物部署的情况,在有障碍物部署的情况下算法的平均定位误差增加的幅度越来越大.相比于没有障碍物部署的情况,算法的定位误差增加的平均幅度分别为:MCL 为 97%,MCB 为 108%,EMCL 为 117%,MSL\*为 128%.可以看出,一种算法越多地使用了 2 跳锚节点的信息来减小定位误差,在有障碍物部署的情况下,它的定位精度下降得就越多,如 EMCL 算法和 MSL\*算法.当节点的密度增加时,所考察算法的平均定位误差也会增加,增加的平均幅度如下:MCL 为 31%,MCB 为 34%,EMCL 为 43%,MSL\*为 58%.可以看出,EMCL 和 MSL\*所受的影响显著大于 MCL 和 MCB 所受的影响.这是因为在 EMCL 和 MSL\*中使用了待定位节点邻居的估计位置信息来减小定位误差.然而,由于受如图 13 所示情况的影响,在有障碍物部署的环境中所得到的待定位节点邻居的估计位置信息变得不准确,从而影响了这两种算法的定位精度.当节点的移动速度增加时,各种算法的定位误差增加的平均幅度分别如下:MCL 为 23%,MCB 为 23%,EMCL 为 29%,MSL\*为 33%.这种情况下,各种算法所受的影响是最小的,因为节点速度的变化既不影响锚节点的密度,也不影响待定位节点的定位精度.同时可以看出,当节点的最大移动速度为  $0.4r$  左右时,MCL 算法和 MCB 算法的定位误差增加的最多.这是因为,在没有障碍物部署的情况下,这两种算法在这个速度下的定位误差最小,如图 12 中右图所示.对于 EMCL 算法和 MSL\*算法,可以得出类似的结论.

综上所述可以看出,已有工作中提出的在没有障碍物部署的情况下可以有效地提高定位精度的技术,如使用 2

跳锚节点的信息来提高定位精度,在有障碍物的部署环境中不再有用.相反地,在这种情况下,它们会显著地降低算法的定位精度.我们需要研究新的方法来在这种情况下有效地提高定位精度.

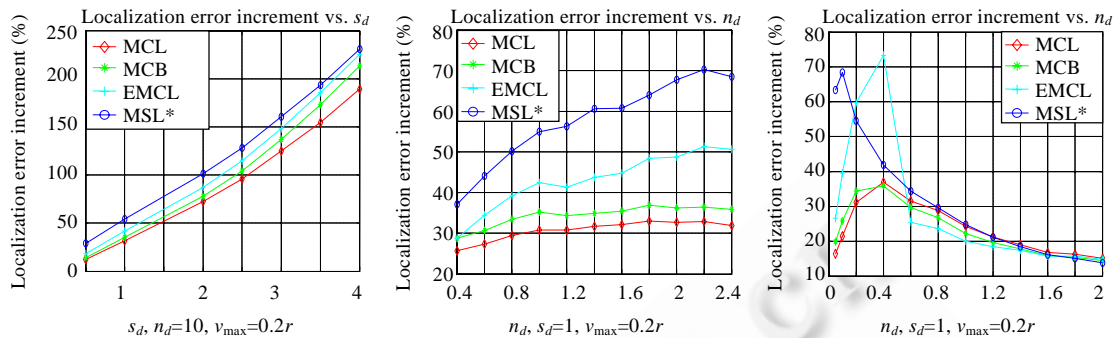


Fig.15 Increment in location error in different algorithms in deployment environments with obstructs

图 15 不同算法在有障碍物的部署环境中的定位误差增加情况

## 6 节点如何判断自身定位精度

本文所讨论的算法均基于序列蒙特卡罗方法.在这类算法中,一个待定位节点用一组带权样本来表示自己的位置分布.但是在这类算法中,单个节点无法确定自己位置估计的精度.尽管我们可以从整体上来说某个定位算法有多精确,但是没有办法判断单个节点的位置估计信息是否准确.文献[11]中指出,当节点的定位误差在  $0.4r$  以上时,会对依赖于节点位置估计的协议,如路由协议、聚类协议等造成较大影响.所以,提供一种节点能够借以判断自身位置估计精度的度量标准是很有意义的.

在以前的工作中<sup>[3-9]</sup>,都没有关注过如何给出一个度量,使得节点可以借以衡量自身位置估计的精度.在我们以前的工作中,提出了使用节点的最大可能定位误差来判断一个节点自身位置估计的精度.在仿真过程中发现,在 MCL<sup>[3]</sup>的仿真程序中给出的 *Sample\_range* 和在 MSL\*<sup>[6]</sup>中提出的 *closeness* 均可以作为节点借以判断自身位置估计的精度度量标准.除此之外,本文提出另外两个可以衡量单个节点位置估计的精度度量.其定义如下:

- 1) *Sample\_range*.即是一个节点的估计位置与它的所有样本距离之中的最大值.根据第 1 节提出的相关定义,*Sample\_range* 的具体定义如下:

$$Sample\_range = \max\{dis(l_{est}, l_t^{(i)}) | 1 \leq i \leq N\},$$

其中,  $dis(x,y)$  表示两个点  $x$  和  $y$  之间的欧氏距离.

- 2) *closeness*.文献[6]中提出的 *closeness* 可以用来比较节点之间位置估计的相对精度.其定义如下(该工作<sup>[6]</sup>仿真程序中定义的 *closeness* 与文章中定义的不一致,这个地方我们取程序中的定义):

$$closeness = \frac{\sqrt{\sum_{i=1}^N w_t^{(i)} (l_{est}(x) - l_t^{(i)}(x))^2} + \sqrt{\sum_{i=1}^N w_t^{(i)} (l_{est}(y) - l_t^{(i)}(y))^2}}{2},$$

其中,  $l_{est}(x)$  和  $l_{est}(y)$  分别代表点  $l_{est}$  的  $x$  分量和  $y$  分量.

- 3) 最大可能定位误差 possible maximum localization error(简称 err).在我们以前的工作<sup>[7,8]</sup>中提出的最大可能定位误差定义如下:假设节点的 Bounding-box 为  $B$ ,则 err 定义为

$$err = \max\{dis\{l_{est}, p\} | p \in B\}.$$

- 4) *Avg\_range*.与 *Sample\_range* 表示节点估计位置与所有样本之间距离的最大值不同,我们提出 *Avg\_range*,表示节点估计位置与所有样本之间的带权平均距离:

$$Avg\_range = \sum_{i=1}^N w_t^{(i)} dis(l_{est}, l_t^{(i)}).$$

- 5)  $p(l_{est}|o_t)$ . 在 MSLS\* 和 EMCL 算法中, 每个样本可以带有不同的权值, 那么很自然地可以使用  $p(l_{est}|o_t)$  来比较节点之间位置估计的相对精确程度。

上面 5 个度量可以分为两类: 第 1 类, 如度量 2) 和度量 5); 它们只能衡量节点位置估计的相对精度, 而不能衡量节点位置估计的绝对精度。通过比较这样的度量, 可以推出一个节点的位置估计可能比另外一个节点的位置估计更精确, 但是并不能知道节点的绝对定位误差; 第 2 类, 如度量 1)、度量 3) 和度量 4), 它们既可以用作节点位置估计的相对精度的比较, 也可用作节点位置估计的绝对精度的比较。

我们通过两种方式来衡量这些度量的效果。针对所有的度量, 利用相关分析来计算它们的值与节点的实际定位误差之间的相关性。对可以衡量节点的绝对定位精度的度量值, 我们计算该值大于节点实际定位误差的比例  $f_{\geq}$ 。因为只有当这样的度量值大于节点的实际定位误差时, 才可以作为定位误差的估计。

图 16 中的左图给出了对 358 427 个节点定位误差的记录中不同度量的效果对比。可以看到, 度量 1)、度量 3) 和度量 4) 与节点实际定位误差的相关系数均大于 0.5, 即它们均与节点实际定位误差显著相关。而 *closeness* 在这方面表现较差, 相关系数仅有 0.457 6, 表现为低度直线相关。令人惊讶的是, 度量 5) 与节点实际定位误差相关度极低, 表明使用这种度量来衡量节点的位置估计的定位精度并不可行。我们猜想, 这是由于节点的位置估计是所有合理样本的加权平均, 故  $p(l_{est}|o_t)$  均比较大, 接近 1。这就使得各个度量值之间无法拉开距离, 进而无法有效地比较节点的位置估计的相对精度。我们统计了所得到的数据并画出了对应的累积分布函数, 如图 16 中的右图所示。我们发现, 在多达 71% 的记录中,  $p(l_{est}|o_t)$  值位于 0.7~1 之间; 而在 55% 的记录中, 该度量值位于 0.9~1 之间, 从而证实了我们的猜测。

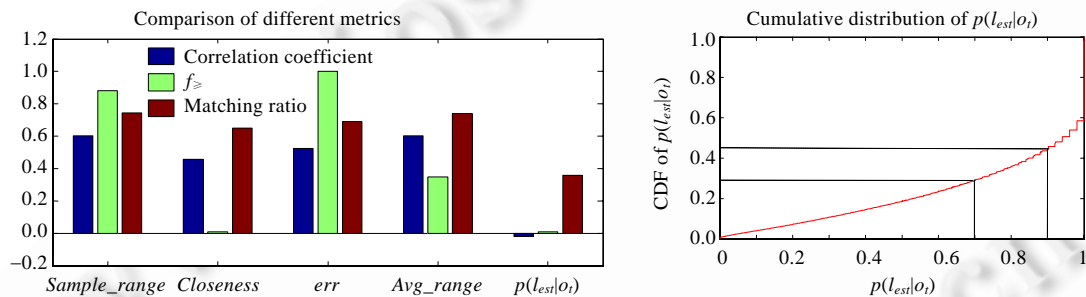


Fig.16 Comparison of different metrics

图 16 不同度量的效果比较

除了使用相关系数, 我们还提出了另外一个值, 即匹配比例, 来比较 5 种度量在衡量节点的位置估计的相对精度时的效果。对两个节点来说, 如果它们的实际定位精度的比较情况与相应度量的比较情况相同, 即均大于或均小于, 则可以认为这个度量与实际定位精度匹配。图 16 中的左图画出了在 10 000 个记录中各个度量的匹配比例。可以看到, 比较的结果与用相关系数比较的结果相同, 说明这两种比较均是符合实际的。

对于那些可以衡量节点位置估计的绝对精度的度量, 表现最好的依次是度量 3)、度量 1) 和度量 4)。度量 3) 和度量 1) 表现得相当好, 节点的实际定位精度小于度量给出的值的比例分别为 100% 和 88%。而对于度量 4), 只有在 35% 的情况下, 节点的实际定位精度才小于度量 4) 给出的误差。

综上所述, 我们认为, *err* 和 *Sample\_range* 是较好的两个节点可以借以衡量自己的位置估计精度的度量。由 *err* 给出的定位误差总是有效的, 所以, 我们认为 *err* 在节点判断自身位置估计的绝对精度时会更有效。

## 7 总结和展望

在已有的关于移动传感器网络中定位算法的工作中, 通过仿真所得到的性能数据往往存在较大差异。我们分析了这种不一致性的原因, 发现是因为仿真过程中使用的参数, 包括所使用的随机网络的个数、仿真的时间片数、性能数据的统计区间等存在着不一致性所致。通过定量分析, 我们确定了仿真过程的参数, 以使所获取的

统计数据较为稳定.使用这些参数,我们对已有的典型移动传感器网络定位算法在不同场景下的定位精度进行了评测和比较.我们发现,一些在无障碍物的部署环境中可以有效提高定位精度的技术在有障碍物的部署环境中不再有效,反而会降低算法的精度.我们还提出了几种节点可以借以衡量自身位置估计精度的度量,并比较了它们的效果.

考虑到待定位节点的密度要比锚节点密度高得多,使用待定位节点的估计位置来提高移动传感器网络定位算法精度的效果并不如预想的那样好,特别是在有障碍物部署的环境中.在以后的工作中,我们计划从理论和仿真两个方面来研究这个问题.已有的提高定位精度的算法在有障碍物部署的环境中不再有效,在这种情况下,新的提高定位精度的技术也将作为我们将来的一个研究方向.

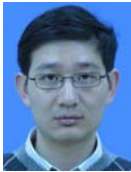
## References:

- [1] Li JZ, Li JB, Shi SF. Concepts, issues and advance of sensor networks and data management of sensor networks. *Journal of Software*, 2003,14(10):1717–1727 (in Chinese with English abstract). <http://www.jos.org.cn/1000-9825/14/1717.htm>
- [2] Akyildiz IF, Su W, Sankarasubramaniam Y, Cayirci E. Wireless sensor network: A survey. *Computer Networks*, 2002,38(4): 393–422. [doi: 10.1016/S1389-1286(01)00302-4]
- [3] Hu LX, Evans D. Localization for mobile sensor networks. In: Haas ZJ, ed. *Proc. of the 10th Annual Int'l Conf. on Mobile Computing and Networking (Mobicom 2004)*. Philadelphia: ACM, 2004. 45–57. [doi: 10.1145/1023720.1023726]
- [4] Baggio A, Langendoen K. Monte Carlo localization for mobile wireless sensor networks. *Ad Hoc Networks*, 2008,6(5):718–733. [doi: 10.1016/j.adhoc.2007.06.004]
- [5] Dil B, Dulman S, Havinga PJM. Range-Based localization in mobile sensor networks. In: Rome K, ed. *Proc. of the 3rd European Workshop on Wireless Sensor Networks*. Zurich: Springer-Verlag, 2006. 164–179. [doi: 10.1007/11669463\_14]
- [6] Rudafshani M, Datta S. Localization in wireless sensor networks. In: Abdelzaher TF, ed. *Proc. of the 6th Int'l Conf. on Information Processing in Sensor Networks (IPSN 2007)*. Cambridge: ACM, 2007. 51–60.
- [7] Zhang SG, Cao JN, Chen LJ, Chen DX. Locating nodes in mobile sensor networks more accurately and faster. In: Banerjee S, ed. *Proc. of the 5th Annual IEEE Communications Society Conf. on Sensor, Mesh and Ad Hoc Communications and Networks*. San Francisco: IEEE, 2008. 37–45. [doi: 10.1109/SAHCN.2008.15]
- [8] Zhang SG, Cao JN, Chen LJ, Chen DX. Accurate and energy-efficient range-free localization for mobile sensor networks. *IEEE Trans. on Mobile Computing*, 2010,9(6):897–910. [doi: 10.1109/TMC.2010.39]
- [9] Yi JY, Yang SW, Cha HJ. Multi-Hop-Based Monte Carlo localization for mobile sensor networks. In: *Proc. of the 4th Annual IEEE Communications Society Conf. on Sensor, Mesh and Ad Hoc Communications and Networks*. San Diego: IEEE, 2007. 162–171. [http://ieeexplore.ieee.org/xpls/abs\\_all.jsp?arnumber=4292828](http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=4292828) [doi: 10.1109/SAHCN.2007.4292828]
- [10] Wang FB, Shi L, Ren FY. Self-Localization systems and algorithms for wireless sensor networks. *Journal of Software*, 2005,16(5): 857–868 (in Chinese with English abstract). <http://www.jos.org.cn/1000-9825/16/857.htm> [doi: 10.1360/jos160857]
- [11] He T, Huang CD, Blum BM, Stankovic JA, Abdelzaher T. Range-Free localization schemes for large scale sensor networks. In: Johnson DB, ed. *Proc. of the 9th Annual Int'l Conf. on Mobile Computing and Networking*. San Diego: ACM, 2003. 81–95. [doi: 10.1145/938985.938995]
- [12] Nagpal R, Shrobe H, Bachrach J. Organizing a global coordinate system from local information on an ad hoc sensor networks. In: Zhao F, Guibas L, eds. *Proc. of the 2nd Int'l Workshop on Information Processing in Sensor Networks (IPSN 2003)*. Berlin, Heidelberg: Springer-Verlag, 2003. 333–348.
- [13] Juang P, Oki H, Wang Y, Martonosi M, Peh LS, Rubenstein D. Energy-Efficient computing for wildlife tracking: design tradeoffs and early experiences with ZebraNet. In: Wood DA, ed. *Proc. of the 10th Int'l Conf. on Architectural Support for Programming Languages and Operating Systems (ASPLOS-X)*. San Jose: ACM, 2002. 96–107. [doi: 10.1145/605397.605408]
- [14] Wark T, Crossman C, Hu W, Guo Y, Valencia P, Sikka P, Corke P, Lee G, Henshall J, Prayaga K, O'Grady J, Reed M, Fisher A. The design and evaluation of a mobile sensor/actuator network for autonomous animal control. In: Abdelzaher TF, ed. *Proc. of the 6th Int'l Conf. on Information Processing in Sensor Networks (IPSN 2007)*. Cambridge: ACM, 2007. 206–215. [doi: 10.1145/1236360.1236389]

- [15] Camp T, Boleng J, Davies V. A survey of mobility models for ad hoc network research. *Wireless Communications and Mobile Computing*, 2002,2(5):483-502. [doi: 10.1002/wcm.72]
- [16] Yoon J, Liu M, Noble B. Random waypoint considered harmful. In: *Proc. of the 22nd Annual Joint Conf. of the IEEE Computer and Communications*. 2003. 1312-1321. [http://ieeexplore.ieee.org/xpls/abs\\_all.jsp?arnumber=1208967](http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1208967) [doi: 10.1109/INFCOM.2003.1208967]

附中文参考文献:

- [1] 李建中,李金宝,石胜飞.传感器网络及其数据管理的概念、问题与进展. *软件学报*,2003,14(10):1717-1727. <http://www.jos.org.cn/1000-9825/14/1717.htm>
- [10] 王福豹,史龙,任丰原.无线传感器网络中的自身定位系统与算法. *软件学报*,2005,16(5):857-868. <http://www.jos.org.cn/1000-9825/16/857.htm> [doi: 10.1360/jos160857]



张士庚(1981-),男,山东滕州人,博士,讲师,主要研究领域为分布式计算,传感器网络技术.



陈道蓄(1947-),男,教授,博士生导师,CCF高级会员,主要研究领域为分布式计算.



曾英佩(1984-),男,博士生,主要研究领域为传感器网络技术.



谢立(1942-),男,教授,博士生导师,CCF高级会员,主要研究领域为分布式计算,网络安全.



陈力军(1961-),男,博士,教授,主要研究领域为移动网络,分布式计算.