

An Improved Centroid Localization Algorithm Based on Received Signal Strength

Han Wang¹, Wangdong Qi

1. Department of Computer Engineering, PLAUST, Nanjing, China

1.e-mail wanghanxuzhou@126.com

Abstract: Centroid localization algorithm(CLA) is a distributed range-free localization method, with the advantages such as low-cost and simple to implement, whereas its localization error is big. APIT is an improvement of CLA. Location estimation of APIT is precise when anchors are deployed uniformly and densely whereas its computational cost is very high, moreover, the performance of APIT depends highly on the success of point in triangle test. When anchors and unknown nodes are deployed randomly, if some unknown nodes can not be covered by any anchor triangle, APIT will not perform on those nodes which make the average localization precision worse. In this paper, we design an improved centroid localization algorithm based on received signal strength. The core idea of our algorithm is, unknown nodes sort neighbor anchors according to corresponding RSS value, the estimation of location is then calculated as the centroid of the three closest anchors which satisfy unknown node in triangle test. A large number of simulations show that, when wireless sensor network is deployed randomly, the localization precision of RSSC is better than APIT; when WSN is deployed uniformly, the localization precision of RSSC is a little trivial. On average, computational cost of RSSC is less than a quarter of APIT.

Keywords: centroid localization; radio signal strength; localization precision; computational cost

基于接收信号强度的改进质心定位算法

汪晗¹, 齐望东

1. 解放军理工大学计算机系, 南京, 中国, 210007

1. E-mail wanghanxuzhou@126.com

【摘要】质心定位算法是一种不基于测距的分布式定位方法, 具有耗费低、实现简单等优点, 但其定位误差大。APIT 算法是对质心定位算法的改进, 在锚节点均匀、密集布设的情况下, 定位精度很高, 但计算开销很大, 而且算法的有效执行依赖于三角形内点测试能否成功。当锚节点和未知节点完全随机布设时, APIT 算法会在许多未知节点上失效, 此时的平均定位精度将会急剧下降。本文设计了基于 RSS 的改进质心定位算法, 其核心思想是, 未知节点依据 RSS 值对邻居锚节点进行排序, 寻找距离最近的且满足三角形内点测试的锚节点组作为参考锚节点, 将其质心作为节点的估计位置。大量仿真实验表明, 在网络随机分布的情况下, RSSC 算法的定位精度高于 APIT; 在网络均匀布设的情况下, RSSC 的定位精度略小于 APIT。在平均情况下, RSSC 的计算开销至多仅为 APIT 的 1/4。

【关键词】质心定位; 接收信号强度; 定位精度; 计算开销

1 引言

节点定位是无线传感器网络的一项关键技术, 微型传感器节点监测、感知和采集的各种环境或对象信息只有与特定的节点位置相结合才能为相关应用提供有效的决策支持。定位算法可以分为基于测距的定位算法和不基于测距的定位算法, 前者具有更高的定位精度, 但在硬件成本和实现复杂性上具有更大的开销; 后者虽然定位误差较大, 但无需额外的硬件支持, 更易于实现。

最经典的不基于测距的定位算法是质心法[1]。在该算法中锚节点周期性地向邻居节点广播一个信标信号, 信号中包含自身 ID 和位置信息。当未知节点接收到来自不同锚节点的信标信号数量超过某一个预设门限或接收一定时间后, 该节点就估计自身位置为这些锚节点所组成的多边形的质心。质心法实现简单, 完全基于网络连通性, 无需锚节点和未知节点间进行协调, 但是该算法要求锚节点在定位区域内较均匀地布设, 否则位置的估计会向区域的中央严重地偏移。APIT

定位算法[6]是对质心定位算法的改进,其核心思想是三角形内点测试,一个待定位节点任选三个相邻锚节点,判断自己是否位于他们所组成的三角形中。使用不同的锚节点组合重复测试,直到穷尽所有的组合,最后计算包含待定位节点的所有三角形交集的质心,并以质心的位置作为待定位节点的估计位置。APIT 算法的定位精度依赖于锚节点布设的密度,计算复杂性非常大。

本文提出了改进的质心定位算法 RSSC,在不增大任何通信开销的情况下,平均定位精度较质心法有很大提高,在网络完全随机布设的情况下可以与 APIT 算法相媲美,但计算复杂度较 APIT 算法有很大降低。

本文其余部分的组织如下:第二部分详细分析了质心法误差大的原因及 APIT 的计算复杂度;第三部分介绍了基于信号接收强度的改进质心定位算法 RSSC;第四部分在多个场景下对 RSSC 算法的定位误差和计算复杂度进行了大量仿真,并与质心法和 APIT 算法相比较;第五部分为总结和下一步的研究方向。

2 质心法和 APIT 分析

2.1 布设需求

现有的不基于测距的定位方法,如凸规划[2]、DV-hop[3]、质心定位、APIT 等方法,均受锚节点分布位置的约束,为减小定位误差,需要锚节点在定位区域内较均匀、密集地布设。如果锚节点在区域边缘 D 中的分布很稀疏,大部分锚节点集中在区域中央,那么使用凸规划和质心法定位时,D 中未知节点的估计位置将会向区域的中央偏移。用 APIT 算法[6]定位时,D 中会存在大量无法被任何锚节点三角形覆盖的未知节点,在这些节点上 APIT 算法将失效。

在一个定位系统中,锚节点的位置通常由 GPS 等手段获得,其成本较普通节点(位置未知的节点)要高很多,因此,为降低定位系统中锚节点的比例,而又达到较好的锚节点覆盖率,可通过增大锚节点和待定位节点的无线射程比例 ANR(anchor node radio range ratio)来达到。ANR 越大,锚节点信号的传播距离会越远,定位系统所需要的锚节点数就越少,相应的布设成本也越小。

2.2 APIT 的计算复杂度

在三角形内点测试成功,APIT算法可以对未知节点定位时,算法的定位精度取决于锚节点的布设密度,密度越大,未知节点在定位过程中所使用的锚节点个数就越多,可组合的三角形个数就会越多,重叠的区

域越小,取重叠区域的质心作为节点的位置估计,其精度就越高。APIT定位过程的计算复杂度很高,假设未知节点的邻居未知节点数的平均值为 α ,邻居锚节点的平均值为 β ,在最差情况下,一次三角形内点测试需要进行 3α 次RSS值的比较;而三角形内点测试的次数为 C_{β}^3 ;在计算三角形的重叠区域时,虽然提出了APIT聚合算法,但计算复杂度仍为 $O(L)$,L是进行三角形内点测试的次数 C_{β}^3 。APIT聚合算法需要频繁地扫描栅格,定位精度和栅格划分的粒度相关,粒度越细,定位精度越高,所带来的计算开销也越大,成 $O(G^2)$ 增加,G是区域划分成栅格的个数。

3 RSSC 算法

本文提出了基于 RSS 的改进质心定位算法 RSSC,其核心思想是,未知节点根据 RSS 值定性判断锚节点的远近,选择满足三角形内部测试、并且最近的三个锚节点作为参考锚节点,计算其质心作为未知节点的位置估计,如果没有邻居锚节点三角形覆盖未知节点,那么就将未知节点的位置确定为最接近的三个锚节点的质心。

3.1 主算法

算法 RSSC 可以分为三步:1) 锚节点周期性发送定位信标报文;2) 未知节点根据接收到的信标报文将邻居锚节点排序,生成序列 $\prec_{AL}$;3) 未知节点根据 $\prec_{AL}$ 进行位置估计。

IC 在未知节点 u 上的主要流程:

1. Receive location beacon(Xi,Yi) and anchor IDi from neighbor anchors.
2. Sort neighbor anchors, generate a reference anchor list $\prec_{AL}$
3. Estimate location:
 Bool PIT = FALSE;
 num=sizeof($\prec_{AL}$)
 For(i = 0; i < num;i ++)
 For(j = 1; j < num;j ++)
 For(k = 2; k < num;k ++)
 aI = $\prec_{AL}$ (i);aJ = $\prec_{AL}$ (j);aK = $\prec_{AL}$ (k);
 /*there is a triangle in $\prec_{AL}$ satisfying PIT */
 If(PITtest(Triangle(aI,aJ,aK)) == TRUE){
 Estimated Position =
 center of gravity(Triangle(aI,aJ,aK))
 PIT = TRUE

```

        return;
    End for
    /*there is no triangle in  $\prec_{AL}$ , satisfying PIT, estimate position by three nearest anchors*/
    If(! PIT)
        aI =  $\prec_{AL}(0)$ ; aJ =  $\prec_{AL}(1)$ ; aK =  $\prec_{AL}(2)$ ;
        Estimated Position =
            center of gravity(Triangle(aI,aJ,aK))
    End if

```

3.2 邻居排序算法

3.2.1 算法流程

RSS 作为距离测量的手段具有硬件成本低、实施方便等优点，本文的邻居排序算法使用 RSS 值来定性判断距离的远近，据此对邻居锚节点排序。

算法在未知节点上的工作流程如下：

1. Receive ID and location beacons(X_i, Y_i) from neighbor anchors
2. Measure RSS value while receiving neighbor anchor's signals
3. Calculate average RSS value for neighbor anchors
4. Sort neighbor anchor according to average RSS, generating a sorted anchor list $\prec_{AL}$:

In PAL, if ($RSS_u > RSS_v$) then $u \prec_{AL} v$.

3.2.2 算法性能的评价

由于 RSS 受环境因素的影响，实际的硬件实验发现，多次测量取平均的方式可以消除 RSS 测量中的噪声信号，但无法消除多径的影响，因此使用 RSS 值对邻居锚节点的远近进行排序时，会出现错误，为了定量刻画该算法的性能，提出下述定义：

设节点 u 有 k 个邻居， $\prec_u$ 是 u 按照邻居节点到 u 的欧式距离进行排序产生的序列，称为参照序列； $\prec_{AL}$ 是 u 按照接收到的邻居信号的 RSS 值进行排序产生的序列。

定义 1： 节点的乱序度

假设 u 的某个邻居锚节点 v 在参照序列 $\prec_u$ 中处于位置 i ，在 $\prec_{AL}$ 中的位置为 j ，如果 $i \neq j$ ，称为乱序，记节点 v 的乱序度为 $|i - j|$ 。

定义 2： 序列的乱序度

序列 $\prec_{AL}$ 相对于参照序列 $\prec_u$ 的乱序度 $N_{\text{deg ree of disorder}}$ 定义为 $\prec_u$ 中所有元素乱序度的总和。

$$er = \frac{N_{\text{deg ree of disorder}}}{N_{\text{max deg ree of disorder}}} \quad (1)$$

定义 3： 排序算法的错误率 er 分母表示序列 $\prec_{AL}$ 的最大乱序度，取值可表示为：

$$N_{\text{max deg ree of disorder}} = \begin{cases} \frac{k^2 - 1}{2} & k \text{ is odd} \\ \frac{k^2}{2} & k \text{ is even} \end{cases} \quad (2)$$

k 是 $\prec_{AL}$ 的长度，当 $\prec_{AL}$ 是参照序列 $\prec_u$ 的逆序排列时， $\prec_{AL}$ 的乱序度达到最大。显然， $0 \leq er \leq 1$ 。

4 仿真结果

4.1 建立仿真模型

4.1.1 节点布设模型

在仿真中，锚节点和待定位节点按指定的平均密度分布在一个正方形区域中，采用随机布设和均匀布设两种方式。

均匀布设：整个区域划分成大小相等的格子，锚节点和待定位节点在格子内部随机布设，在格子之间均匀布设，即每个格子内的锚节点数和待定位节点数是固定的，但在格子内所处的位置是随机指定的；

随机布设：锚节点和待定位节点在整个区域内随机布设。

4.1.2 无线信道模型

由于各个环境的特征不同，我们将解析方法和具体实验相结合以确定 RSS 随距离衰减的模型。采用最常用的对数正态阴影衰落模型[7]，如式(3)所示。

$$P_r = P_t - Pl(d_0) - 10n \log_{10}\left(\frac{d}{d_0}\right) - X_\sigma \quad (3)$$

d 是发送方和接收方之间的距离， d_0 是一个参考距离， n 是路径损耗指数， X_σ 是一个零均值的高斯分布，标准差为 σ ，表征了阴影效应。在室内/外环境进行大规模测量，通过曲线拟合的方法，确定不同环境下的参数，见表 1[8]：

表 1. 无线信道模型参数

environment	n	σ
outdoor	4.7	4.6
indoor	3.0	3.8

4.2 仿真场景及结果分析

4.2.1 RSSC 的定位效果

为了对 RSSC 的定位效果有一个直观上的体现，做如下仿真：

仿真场景：20 个锚节点、180 个未知节点在 120×120 的区域内随机布设，未知节点的无线射程 (node radio range) 为 10 米，锚节点和未知节点的无线电射程比值 ANR 等于 10。

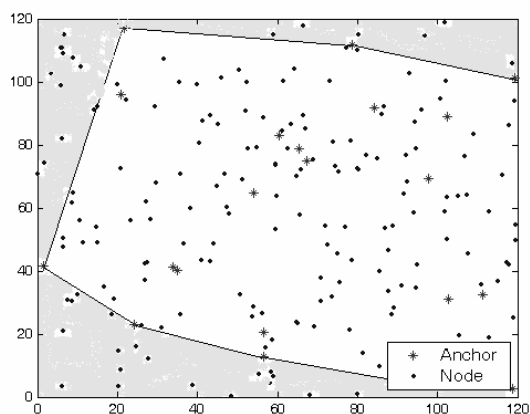


图 1. 布设场景 Random

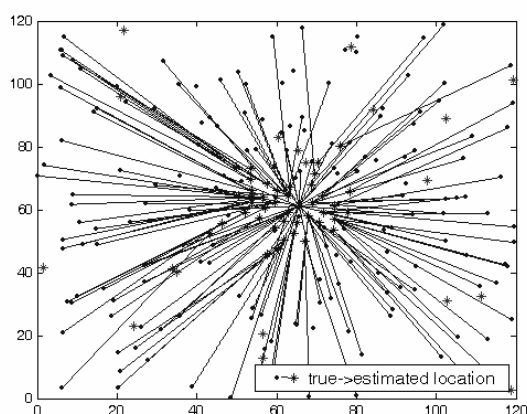


图 2. 质心定位法的抱团现象

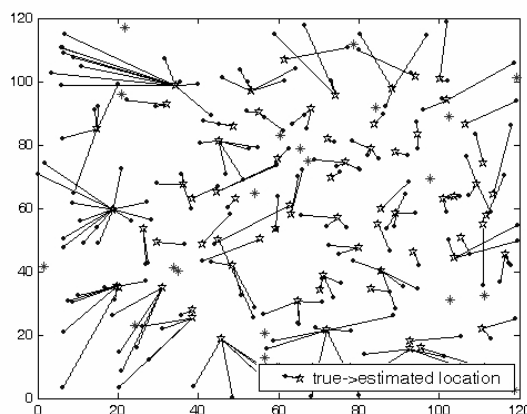


图 3. RSSC 算法的定位效果图

随机生成的场景如图 1 所示，由于锚节点完全随机布设因而容易出现未知节点无法被任何锚节点三角形覆盖的情况，APIT 算法在图 1 蓝色区域内的未知节点上运行失效，只能通过质心法进行位置的估计。由于 $ANR=10$ ，因而锚节点的无线射程为 100 米，这导致未知节点在定位过程中所使用的参考锚节点几乎相同，因此质心法估计的位置出现严重的抱团现象，定位误差很大，如图 2 所示。图 3 表示 RSSC 算法对图 1 场景进行定位的效果，与图 2 相比，节点位置估计的抱团现象大大减轻；对图 1 蓝色区域中的未知节点进行定位，APIT 算法的平均定位的误差为 $3.7R$ ；RSSC 算法的平均定位误差小于 $1.1R$ ，定位精度得到很大提高。

4.2.2 er 对 RSSC 的影响

由于无线电的接收信号强度易受环境的影响，RSS 值的衰减与距离并不呈现单调的递减关系[4]，为分析邻居排序算法在不同信道模型下的排序错误率 er 及其对定位误差的影响，进行如下仿真。

仿真场景：20 个锚节点、180 个待定位节点随机布设在一个正方形区域 A 内， $ANR=7$ ，通过改变 A 的大小进而改变未知节点的平均邻居锚节点数 AH，室内、室外信道模型的参数如表 1 所示。

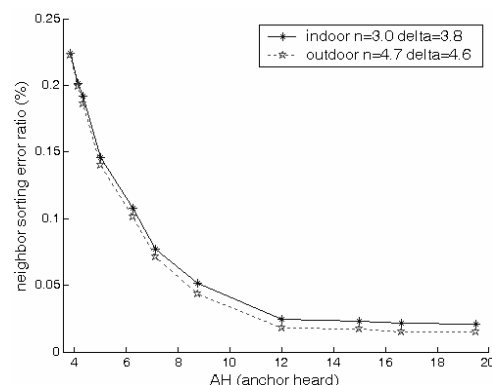


图 4. 不同信道下的 er 值比较 Random

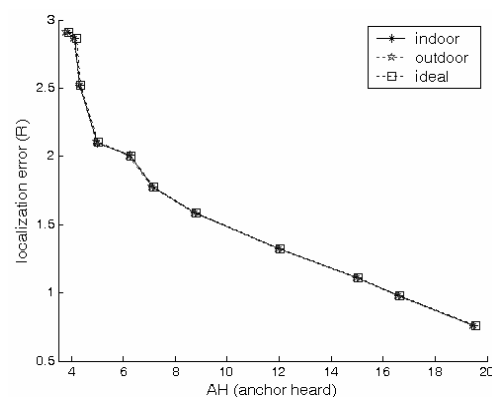


图 5. 不同信道下的定位误差比较 Random

从仿真结果图 4 可知, 在室内、室外两种信道下, er 随着 AH 的增大而减小, 当 $AH < 4$ 时, er 值最大, 但小于 0.25, 这说明使用 RSS 值对邻居锚节点进行排序, 错误率较低。图 5 中曲线 “ideal” 表示理想信道下的定位误差, $er = 0$; 曲线 “indoor” 和 “outdoor” 表示在室内和室外两种信道环境下的定位误差。图 5 说明当 er 值小于 0.25 时, 对定位误差几乎没有影响。

4.2.3 ANR 对定位误差的影响

ANR (anchor node radio range ratio) 越大, 定位系统布设所需要的锚节点数就越少, 相应的耗费也越小。为分析 ANR 对定位误差的影响, 做如下仿真:

仿真场景: 20 个锚节点和 180 个待定位节点随机布设在 120×120 的矩形区域内, 节点布设密度不变, 未知节点的信号传播半径为 10, 采用室外信道模型。

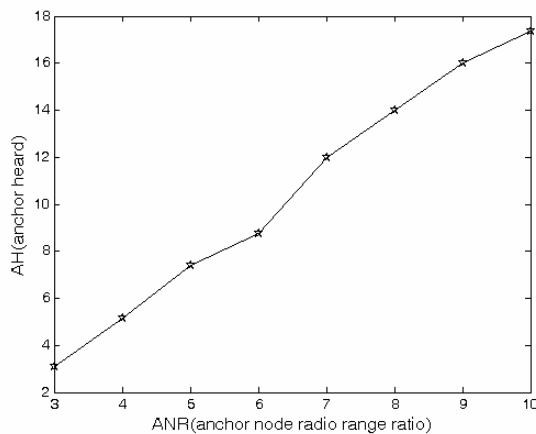


图 6. AH 与 ANR 的关系图 Outdoor, Random

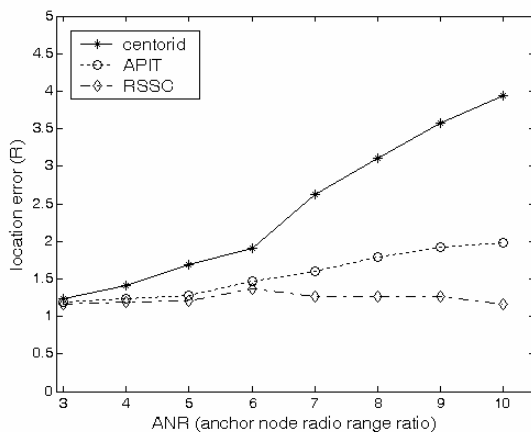


图 7. 定位误差与 ANR 的关系图 Outdoor, Random

增大 ANR 的值, 相应地 AH 值(待定位节点在定位过程中所使用的锚节点数)也会增大, 如图 5 所示。图 6 说明了随 ANR 的增大, 质心法、APIT 的定位误

差都会增大, 质心法的误差增加最剧烈, 这是因为锚节点信号传播更远会导致更大的累积误差。RSSC 的定位误差最小, 误差增长最缓慢, 当 $ANR > 6$ 时, 定位误差甚至出现了下降的趋势。

4.2.4 节点布设场景对定位误差的影响

仿真场景: 20 个锚节点、180 个待定位节点布设在一个正方形区域内, $ANR=7$, 采用室外信道模型, 通过改变布设区域的大小改变未知节点的平均邻居锚节点数 AH 。

在随机布设和均匀布设两种场景下, 比较三种定位方法的性能, 结果见图 8、9。

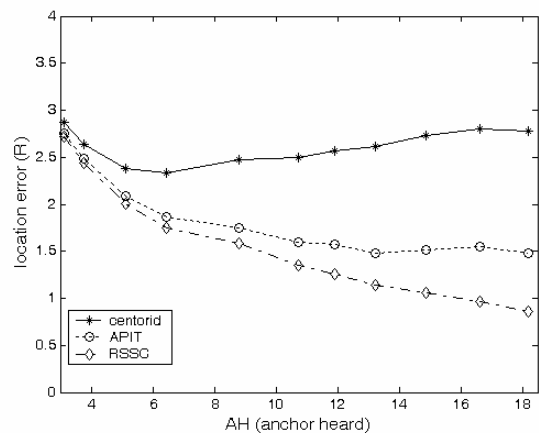


图 8. 定位误差 $ANR=7$, Outdoor, Random

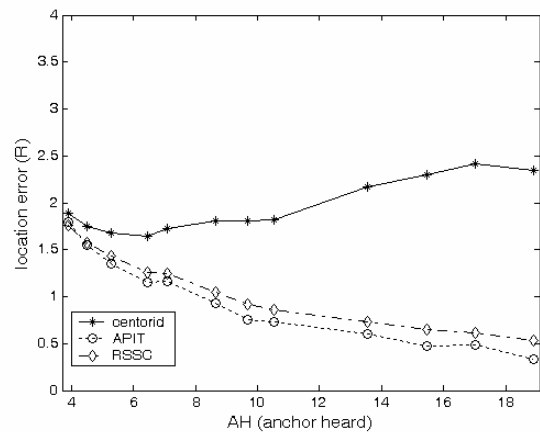


图 9. 定位误差 $ANR=7$, Outdoor, Uniform

4.2.5 RSSC 的计算复杂度

APIT 算法的计算复杂度仅取决于邻居锚节点的个数, 而与邻居锚节点的位置无关。RSSC 算法的复杂度与之不同, 由于 RSSC 算法将邻居锚节点按距离远近进行了排序, 因而在三角形内点测试阶段, 并不是穷尽所有的锚节点组合, 而是从距离最近的锚节点开

始,依次判断锚节点组合能否覆盖未知节点,一旦条件满足,立即停止,并将距离最近且能覆盖未知节点的三角形质心作为待定位节点的位置估计。在最差情况下,RSSC算法仍需要穷尽所有的锚节点组合。为比较APIT和RSSC在平均情况下的计算复杂度,进行如下仿真:

仿真场景: 20个锚节点、180个待定位节点随机布设在一个正方形区域A内,ANR=7,采用室外信道模型,通过改变A的大小进而改变未知节点的平均邻居锚节点数AH。

结果见图10,横坐标AH表示未知节点在定位过程中使用的平均锚节点数,纵坐标的单位是一次内点测试。从图10可知,在平均情况下,RSSC的计算复杂度不大于APIT计算复杂度的1/4。

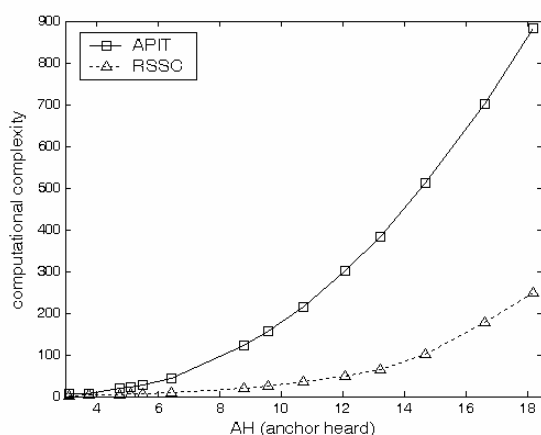


图 10. APIT 和 RSSC 的计算复杂度比较

5 结束语

RSSC算法是不基于测距的、健壮的、分布式定位算法,它结合了质心定位算法实现简单和APIT算法定位精度高的优点,使用易于测量的接收信号强度RSS值将邻居锚节点进行排序,在定位过程中从距离最近的锚节点组开始进行三角形内点测试,一旦找到通过测试的锚节点组就立即停止搜索,并使用该锚节点组

估计未知节点的位置。邻居节点按照距离远近排序对定位误差的改进是显著的,在锚节点和未知节点完全随机布设的情况下,RSSC的平均定位误差甚至小于APIT算法,而平均情况下的计算开销小于APIT算法的1/4。

本文使用了RSS值作为邻居锚节点距离远近的度量,并在室内、室外两种特定信道环境下进行了大量仿真实验,当邻居排序错误率 er 小于0.25时,对RSSC算法的平均定位误差没有影响。信道环境改变后, er 值大于0.25将会怎样影响算法的定位精度,以及在多径影响严重的环境下如何采用其他手段或算法提高邻居排序的正确率,将成为RSSC算法的下一步研究方向,并为RSSC算法的实际应用提供重要支撑。

致谢

本文在研究过程中还得到了威力博士和刘鹏教员的支持和帮助,在此一并提出感谢!

References(参考文献)

- [1] N. Bulusu, J. Heidemann and D. Estrin, GPS-less Low Cost Outdoor Localization for Very Small Devices[J], IEEE Personal Communications Magazine, 2000,7(5),P28-34.
- [2] L. Doherty, K. Pister, and L. Ghaoui, Convex position estimation in wireless sensor networks[C], in Proc. IEEE Infocom, Anchorage, AK, 2001, P1655-1663.
- [3] D.Niculescu and B.Nath, DV Based Positioning in Ad Hoc Networks[J],Netherlands:Telecommunication Systems,2003, 22:1-4, P267-280.
- [4] D. Niculescu and B. Nath, Ad-hoc positioning system[C], in Proc. IEEE Globecom, San Antonio, TX, 2001,P2926-2931.
- [5] T. He, C. Huang, B. Lum, J. Stankovic, and T. Adelszner, Range-free localization schemes for large scale sensor networks[C], in Proc. ACM MobiCom, San Diego, CA, 2003, P81-95.
- [6] Theodore S. Rappaport. Wireless Communications: Principles and Practice[M],Prentice Hall,1996.
- [7] Marco Zuniga and Bhaskar Krishnamachari, Analyzing the Transitional Region in Low Power Wireless Links[C], in Proc. ACM SECON, 2004, P517- 526.