无线传感器网络中节点定位算法①

刘胜辉，董晓瑞，高俊峰

(哈尔滨理工大学 计算机科学与技术系，哈尔滨 150080)

摘 要：近年来，无线传感器网络作为一种新型的数据采集技术得到了空前的发展，研究能够满足实际应用的定位算法成为一个新的热点。考虑到无线传感器节点的成本、性能、功耗和应用领域等不同因素的影响，已经出现了许多实用的定位算法。基于锚点的质心算法定位精度高，而与锚点无关的 ABC 算法的成本低。通过结合这两种定位算法的优点，提出了一种改进算法。在减少质心算法锚点数目的情况下控制 ABC 算法的误差在可接受的范围内，从而来满足实际应用对精确度的要求。最后讨论了使用改进算法需要注意的两方面因素和如何根据实际应用来进行选择。

关键词：无线传感器网络；定位算法；锚点；基于锚点定位算法；与锚点无关定位算法

Localization Algorithm in Wireless Sensor Networks

LIU Sheng-Hui, DONG Xiao-Rui, GAO Jun-Feng

(College of Computer Science and Technology, Harbin University of Science and Technology, Harbin 150080, China)

Abstract: Wireless sensor networks becomes a new data collection technology of autonomous sensor nodes. The research on localization algorithm in the actual applications has been very hot. In this situation, a number of excellent localization algorithm appeared according to various factors, such as the cost, power and applications areas. As we know, the anchor-based mass algorithm has higher precision while the anchor-free ABC algorithm has lower cost. In order to take advantages of them, a new localization algorithm is presented. It does not only control the precision within endure ranges of ABC algorithm, but also reduces the number of anchors of the mass algorithm. So it can meet the needs of the actual applications. Finally, the attentions of the new localization algorithm is indicated.

Keywords: wireless sensor networks; localization algorithm; anchor; anchor-based algorithm; anchor-free algorithm

1 引言

无线传感器网络(Wireless Sensor Networks, WSN)是由部署在监测范围内大量的微型传感器节点构成，通过无线通信方式形成的多跳自组织网络[1]。其中的传感器节点一般是廉价的，低功耗的，具有分布式和自组织等特点。因此，无线传感器网络作为一种新型的数据采集技术得到迅速发展，有广泛的应用[2]。

无线传感器空间位置的确定过程通常被称为定位[3]。无线传感器的定位可应用于儿童监管、无人看守、物流管理、矿工监视、智能交通、军事定位等领域[4]。定位算法常见的分类方式有以下几种：按与距离有无关系分类[5]；按与锚点有无关系分类；按集中式与分布式分类；按递增式与并发式分类；按紧耦合与松耦合分类[6,7]。在无线传感器定位算法中经常使用到的计算方法有三种：三边测量、三角测量和极大似然估计[8,9]。

目前应用最广泛的定位方法就是全球定位系统(GPS, Global Position System)，它是利用多颗 GPS 卫星来完成对全球的信号覆盖，主要由空间 GPS 卫星、地面控制系统和用户设备三部分配合实现定位功能。由于成本、功耗等因素的制约，在工业现场中为每个无线传感器节点都配备 GPS 是一种不现实的解决方

① 收稿时间:2010-09-10;收到修改稿时间:2010-10-16

© 中国科学院软件研究所 http://www.c-s-a.org.cn

案。特别在室内的定位应用中，由于室内环境中的障碍物较多，会使 GPS 信号大大衰减，不足以实现精确定位[10]。因此，在实际应用领域中，考虑到不同的影响因素，寻找满足性能要求的定位算法成为一种合适的选择。

本论文主要是在对已有的基于锚点的质心算法和与锚点无关的 ABC 算法进行分析和研究的基础上提出了一种改进算法，最后通过实验验证了该算法的可行性。

2 现有定位算法分析

2.1 基于锚点的质心算法

质心算法是最典型的一种基于锚点的定位算法[11]。它的算法原理是所有锚点不间断的广播信标信号，其中包括锚点的 ID 和锚点的位置信息。待测节点 X 能接收到锚点 Y_1、Y_2、Y_3 …Y_n 的信标信号。n 个锚点的坐标分别为(x_1,y_1)、(x_2,y_2)、(x_3,y_3) …(x_n,y_n)，点 X 坐标为(x,y)。节点 X 与节点 Y 的连通度 CM(X,Y)定义如公式(1)所示。

$$CM_{(X,Y)} = \frac{N_r(Y,t)}{N_s(Y,t)} \tag{1}$$

其中 $N_r(Y,t)$为时间 t 内 X 节点接收到 Y 节点信标信号的个数，$N_s(Y,t)$为时间 t 内 Y 节点广播的信标信号的个数。

若 X 与锚点 Y_1、Y_2、Y_3 …Y_j 的连通度满足阈值(如 $CM_{(X,Y)}>95\%$)要求，则点 X 的位置为锚点 Y_1、Y_2、Y_3 …Y_j 所组成的多边形的质心，则 X 的坐标如公式(2)所示：

$$(x,y) = \left(\frac{x_1+x_2+\ldots+x_j}{j}, \frac{y_1+y_2+\ldots+y_j}{j}\right) \tag{2}$$

质心算法实现简单，90%待测节点的定位精度小于锚点间距离的 1/3。质心算法要求锚点的密度要大，这就意味着成本的增加。

2.2 与锚点无关的 ABC 算法

ABC(assumption based coordinates)算法属于与锚点无关的一种定位算法[12]。它的算法原理是在可连通范围内，该算法一次计算得出一个待测节点的坐标，在下次计算中利用这个坐标来计算临近节点的坐标，循环计算最终得出所有待测节点的坐标。计算过程如下：

d_{ij} 表示节点 i 到节点 j 的距离，令节点 X0 的坐标为(0,0,0)，其临近节点为 X_1，假设 X_1 坐标为$(d_{01},0,0)$，X_1 的临近节点 X_2 坐标为(x_2,y_2,z_2)，假设 X_2 满足：$y_2>0$；$z_2=0$，则 X_2 的坐标的计算方法如公式(3)所示：

$$\begin{cases} x_2 = \dfrac{d_{01}^2 + d_{02}^2 + d_{12}^2}{2d_{01}} \\ y_2 = \sqrt{d_{02}^2 - x_2^2} \\ z_2 = 0 \end{cases} \tag{3}$$

下一节点 X_3 坐标确定方法与 X_2 类似，但假设 $z_3>0$，则 X_3 的坐标的计算方法如公式(4)所示：

$$\begin{cases} x_3 = \dfrac{d_{01}^2 + d_{03}^2 + d_{13}^2}{2d_{01}} \\ y_3 = \dfrac{d_{03}^2 - d_{23}^2 + x_2^2 + y_2^2 - 2x_2x_3}{2d_{01}} \\ z_3 = \sqrt{d_{03}^2 - x_3^2 - y_3^2} \end{cases} \tag{4}$$

这样便有了 4 个已知坐标的节点，再利用这 4 个节点通过测量距离反复迭代求解，把这些坐标已知的节点再做为中心节点来使用，逐步计算得出所有待测节点的坐标。

ABC 算法在允许测量误差的条件下提高了局部定位的精度，但递增式的计算方法会导致误差的不断累计，使全局定位精度降低。

3 改进算法的原理

质心算法对锚点数量的要求导致实现成本高，而 ABC 算法会导致误差的不断累计，使定位结果不能满足实际的应用。改进算法是利用基于锚点的质心算法和与锚点无关的 ABC 算法相结合的思想来设计的，可以在减少质心算法锚点数目的情况下控制 ABC 算法的误差在可接受的范围内。具体实现方法如图 1 所示。

将每三个锚点组成一个等边三角形，待测节点在这些三角形区域内。每一个三角形内的待测节点利用 ABC 算法的迭代思想进行计算。

© 中国科学院软件研究所 http://www.c-s-a.org.cn

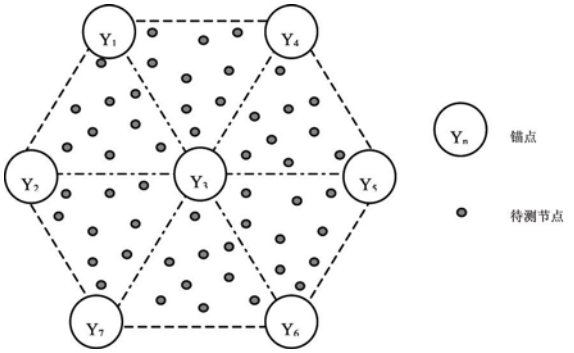

图 1　改进算法示意图

3.1 改进算法的计算策略

1) 确定待测节点属于哪个三角形区域。所有锚点广播的信标信号，待测节点接收信标信号后计算连通度，最简单的方法为选取连通度最高的两个锚点为三角形区域的两个顶点，如图 2 中的 Y_2 和 Y_3，选取连通度排在第三位的锚点做为三角形的第三个顶点，如果这样的点有两个则说明待测点在边 Y_2Y_3 上，这两个点应该为 Y_1 和 Y_7 那么选取任个做为三角形的第三个顶点即可。

2) 确定每个三角形区域内迭代计算的第一个节点。锚点广播的信标信号，待测节点读取信标信号后选取信标信号中锚点 ID 为自身所在三角形的三个顶点的信标信号进行连通度计算，当三个连通度满足阈值(如 $CM_{(X,Y)}$>95%)要求时利用质心算法确定自身坐标。如果有多个待测节点的三个连通度满足阈值要求时选取阈值满足度最高的待测节点(满足度可按平均数来计算)，如相等则随机选择一个节点。最坏一种情况就是没有待测节点的三个连通度能满足阈值的要求，即三角形质心附近没有待测点。这时的解决方法为：选取阈值满足度最高的一个待测节点，利用三边或三角测量法确定这个节点的坐标。这样做可以使这个节点尽可能的接近等边三角形的质心即外接圆圆心。

3) 每个三角形区域内的迭代计算。这里有以下 2 种计算方式：

① 把第 2 步中得到的节点做为第一个节点，直接利用 ABC 算法进行迭代计算。要注意的是使用 ABC 算法产生相对坐标，因此要进行一次坐标的转换才能得到绝对坐标，坐标的转换方法有很多，这里不再赘述。

② 把第 2 步中得到的节点做为一个锚点来使用，这样锚点增加到 4 个。其余待测节点按照距离这个节点的跳数逐步进行定位。待测节点定位时首先把锚点的连通度进行排序，然后取连通度最大的一些锚点利用极大似然估计法进行坐标的计算。计算时取的锚点的个数可以按跳数的增加而增加，第 1 跳时取 4 个、第 2 跳时取 5 个…第 5 跳时取 8 个，在第 5 跳之后的也 8 个。这样做是因为随着迭代过程的进行锚点的数目不断增加，虽然取的锚点个数多对精度有好处，但会增加计算过程所需的时间。假设三角形三个顶点 R_1、R_2、R_3 坐标为(x_1,y_1)、(x_2,y_2)、(x_3,y_3)，第 2 步中得到的节点 R_4 的坐标为(x_4,y_4)，距节点 R4 跳数为 1 的待测节点 R_5 的坐标为(x_5,y_5)，它距 R_1、R_2、R_3、R_4 这 4 点的距离分别为 d_1、d_2、d_3、d_4，则利用极大似然估计法计算可得公式(5)。

$$\hat{X} = (A^T A)^{-1} A^T b \tag{5}$$

其中，

$$A = \begin{bmatrix} 2(x_1 - x_4) 2(y_1 - y_4) \\ 2(x_2 - x_4) 2(y_2 - y_4) \\ 2(x_3 - x_4) 2(y_3 - y_4) \end{bmatrix},$$

$$b = \begin{bmatrix} x_1^2 - x_4^2 + y_1^2 - y_4^2 + d_4^2 - d_1^2 \\ x_2^2 - x_4^2 + y_2^2 - y_4^2 + d_4^2 - d_2^2 \\ x_3^2 - x_4^2 + y_3^2 - y_4^2 + d_4^2 - d_3^2 \end{bmatrix},$$

$$X = \begin{bmatrix} x_5 \\ y_5 \end{bmatrix} \tag{6}$$

把 R_5 做为一个新的锚点来使用，如此递推得所有待测节点的坐标。

3.2 算法详细分析

通过以上 3 步可实现对所有待测节点的定位，这里把锚点布置成等边三角形的原因主要有以下 3 条：

1) 把锚点布置成三角形一方面是因为最少要利用三个锚点才可计算出一个待测点的坐标(例如使用三边测量法或三角测量法)，这样布置可以使锚点的数目尽可能的少，降低成本。另一方面，可以利用三角形组合成其它规则图形以适应不同的应用环境。

2) 布置成等边三角形是因为利用质心算法的得

© 中国科学院软件研究所　http://www.c-s-a.org.cn

到的第一个节点即是质心也是等边三角形外接圆的圆心，由于在第 3 步的计算中待测节点是以第一个节点为中心逐步定位的，所以选择第一个节点时要尽可能的在三角形区域的外接圆圆心，这样可以达到减小误差累积的效果。

3) 当三角形质心附近没有待测节点时，解决方法为选取连通度阈值满足度最高的一个待测节点做为第一个节点，这只能保证这个节点是与质心最接近的一个节点，如果是任意的三角形，质心和外接圆圆心并不重合，那么就不能保证这个节点是最接近外接圆圆心的节点了。

使用这个改进算法时需要注意以下三点：

1) 无线网络丢包问题。无线传感器网络采用无线通信，必然存在丢包问题，这会导致节点接收信标信号的丢失。可利用不同的时间数量极来解决这个问题。设置各节点每 1ms 发送一次信标信号，节点在接收到信标信号后每 100ms 进行一次信标计数。由此可知用于计数的信标个数是接收到的信标个数的 1/100。通过这种方法可以保证即使丢失一定数量的信标信号也不会影响最后的计算。

2) 连通度阈值的设定。阈值过高会导致不能找到的质心附近的第一个节点，虽然可以通过解决方法找到与质心最接近的一个节点，然后再用三边或三角测量法来确定这个节点的坐标，但计算量的增加会导致定位时间的增加；阈值过低会产生很大的误差，特别在使用 ABC 算法时会将这个误差累积放大。

(3) 三角形区域面积的设定。三角形区域的面积控制着 ABC 算法的使用范围，过大时 ABC 算法的使用范围过大，产生的误差会使定位结果不能满足实际应用；过小时锚点的密度大，不能够起到降低成本的作用。

通过以上两点可以得出连通度阈值与三角形区域面积的设定需要根据具体应用确定，在具体应用中可按如下步骤反复迭代来确定这两个参数值：

1) 先采用高的连通度阈值和小的三角形区域面积来进行第一次实验。由于连通度阈值最高为 100%，而无线网络存在丢包问题导致这只能是一个理论值，实际的连通度阈值要小于 100%，根据大量的实验证明，可先采用连通度阈值为 95%，三角形区域边长为 10 米来进行第一次实验。放置待测节点时要考虑在三角形质心附近没有待测节点的情况。实验结果如不能满足精度需求，则以 1%以量度提高连通度阈值，以 2 米为量度减小三角形区域边长，再进行反复迭代实验。需要注意的是：如果应用中对计算时间要求高时应优先选择减小三角形区域的边长，因为提高连通度阈值会导致计算时间的增加；如果应用中对成本的要求高时应优先选择提高连通度阈值，因为减小三角形区域的边长会增加锚点的数量，导致成本的增加。除对精度要求极高的应用外一般工业应用中设置连通度阈值为 95%，三角形区域边长为 10 米都可满足精度要求，这时则进行步骤 2。

2) 以 3%为量度降低连通度阈值，以 5 米为量度加长三角形区域边长。再进行反复迭代实验，直到得出能够满足应用精度需求的最小连通度阈值和最大三角形区域面积为止。需要注意的是：如果应用中对计算时间要求高时应优先选择降低连通度阈值，因为降低连通度阈值会导致计算时间的减少；如果应用中对成本的要求高时应优先选择加长三角形区域边长，因为加长三角形区域的边长会减少锚点的数量，导致成本的降低。如此得出的最小连通度阈值和最大三角形区域面积可在降低了成本的同时保证了一定的定位精度。

3.3 算法复杂度分析

设待测节点个数为 n，锚节点个数为 m，锚节点所组成的三角形区域的个数为 p，第 p 个三角形区域内有待测节点 S_p 个。可知 $S_1+S_2+S_3+...+S_p=n$，由锚节点是人为布置的，一旦布置完成，所形成的三角形区域的个数就确定的，因此可将 m 和 p 视为常数。根据 3.1 节中的计算步骤可进行算法的时间复杂度和空间复杂度分析。

1) 时间复杂度。在第一步中要每一个待测节点与 m 个锚点的连通度进行比较判断后确定每个节点属于哪个三角形区域。时间复杂度为 O(n*m)。第二步确定每个三角形区域中迭代计算的第一个节点，由于有 p 个三角形区域就有 p 个这样的点，可知时间复杂度为常数 O(p)。第三步其余的待测节点采用极大似然估计法进行迭代计算，由极大似然估计的时间复杂度为 $O(S_p)$，而这样的点有 S_p 个可知 p 个三角形区域所有待测节点的时间复杂度为 $O(S_1^2+S_2^2+S_3^2+...+S_p^2)$，可近似认为是 $O(n^2)$。综合得出改进算法的时间复杂度为

© 中国科学院软件研究所 http://www.c-s-a.org.cn

$O(n^2)$。

2) 空间复杂度。由所有的定位算法的目的是得出所有节点的坐标以便计算使用，可知每个节点都有要有一个存贮空间来存放该节点的坐标，且不能交差利用。因此算法的空间复杂度为 O(n+m)。具体为首先为每个节点分配一个存贮空间，锚节点直接存贮其坐标值。进行第一步计算后将每个待测节点属于哪个三角形区域存入待测节点的贮存空间中。第二步和第三步后不再需要待测节点属于哪个三角形区域，因此可以将每个待测节点计算后的坐标存入到对应的存贮空间中。因此改进算法的空间复杂度为 O(n+m)。

4 改进算法的验证

4.1 实验设置与误差定义

实验环境为一地下停车场，在该停车场内随机设置了 25 个待测节点，这些节点的分布情况如图 2 和图 3 中的圆形节点所示。在保持这 25 个待测节点位置不变的情况下分别利用 ABC 算法，质心算法和改进算法对这 25 个待测节点进行定位，对其结果进行比较。在改进算法中根据上节中确定两个参数值的步骤，先采用连通度阈值为 95%，三角形区域边长为 10 米来进行第一次实验，之后以 3%为量度降低连通度阈值，以 5 米为量度加长三角形区域边长进行迭代。最后确定连通度阈值为 89%(迭代 2 次)，每个三角形区域的边长为 50 米(迭代 8 次)。改进算法的锚点分布情况如图 3 中菱形节点所示。

定义 E_m 为第 m 个待测节点的定位误差，定位得到的待测节点的坐标为(x_m', y_m')，其真实坐标为(x_m, y_m)，则各待测节点的定位误差的计算方法如公式(7)所示，即定位坐标与真实坐标间的距离。

$$E_m = \sqrt{(x_m' - x_m)^2 + (y_m' - y_m)^2} \tag{7}$$

定义 n 为第待测节点的个数，则整个系统的定位误差 E 的计算方法如公式(8)所示，即各待测节点定位误差的平均数。

$$E = \frac{E_1 + E_2 + \ldots + E_n}{n} \tag{8}$$

4.2 实验过程与结果

1) ABC 算法。图 2 为使用 ABC 定位算法的实验结果。其中圆形节点是待测节点的真实坐标，正方型节点是定位后的坐标，标号相同的正方型节点与圆形节点间的距离就是定位过程中产生的误差。使用节点 1 做为 ABC 算法的第一个节点。可以看出距离节点 1 近的待测节点的定位误差小，距离这个节点远的待测节点的定位误差大。

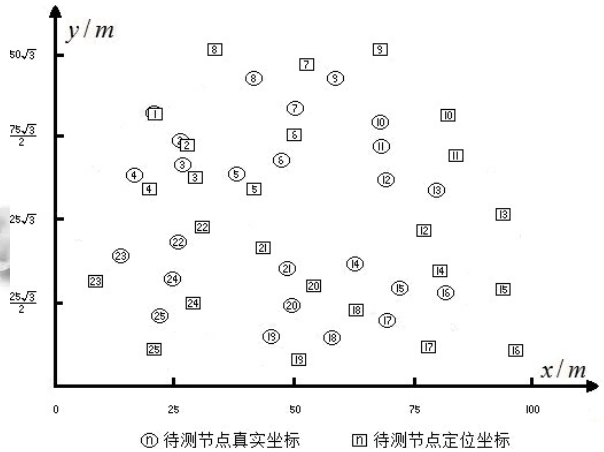

图 2　ABC 算法实验结果

2) 质心算法。如采用单纯的质心算法对图 2 中的 25 个待测节点进行定位，那么最理想的情况下也需要有 50 个以上的锚点才能使定位后的误差控制在能满足实际需求的范围内。这会导致应该成本的急剧加大，根本不能满足实际应用的要求。

3) 改进算法。图 3 为使用改进算法的实验结果。其中菱形节点为锚点，圆形节点是待测节点的真实坐标，正方型节点是定位后的坐标，标号相同的正方型节点与圆形节点间的距离就是定位过程中产生的误差。通过上节中讲到的参数设置方法，首先设置连通度阈值为 95%，三角形区域边长为 10 米进行第一次实验，之后以 3%以量度降低连通度阈值，以 5 米为量度加长三角形区域边长，再进行反复迭代实验。经实验得出按每三个锚点组成一个等边三角形的方式在该停车场内设置了 7 个锚点，每个三角形区域的边长为 50 米(迭代 8 次)，其具体分布情况如图 3 中菱形节点所示，设置连通度阈值为 89%(迭代 2 次)。每个三角形质心附近的那个节点做为迭代计算的第一个节点，它的定位误差最小，其余待测节点的定位误差按距离这个节点由近到远逐步增大。

© 中国科学院软件研究所　http://www.c-s-a.org.cn

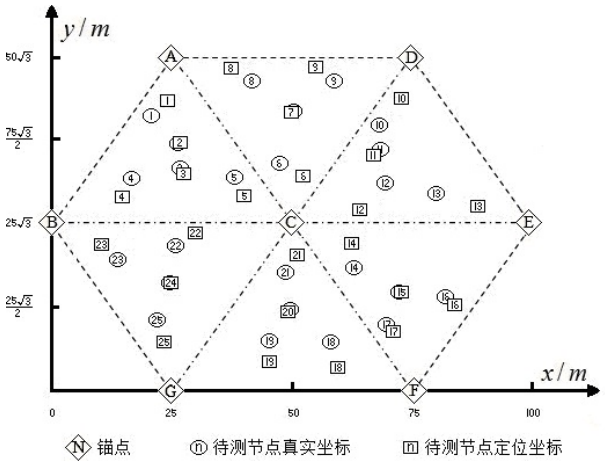

图 3 改进算法实验结果

对成本的要求。

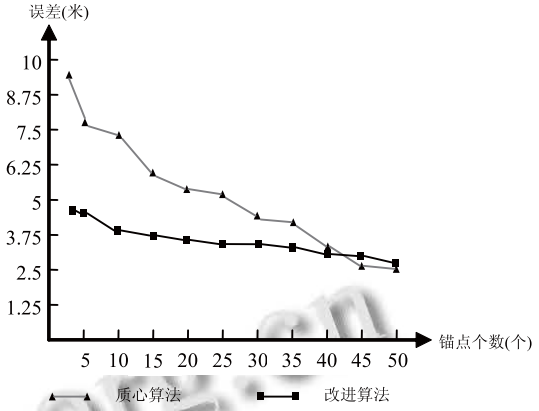

图 4 质心算法与改进算法误差对比

4.3 实验结果分析

通过图 2 与图 3 的对比可以得出以下结果：

1) 在图 2 中，节点 1 做为迭代计算的第一个节点，这个节点的定位误差是最小的。而在图 3 中这样的节点有 6 个，分别是节点 2、7、11、15、20、24，这就相对提高了(6-1)/25 个节点的定位精度。

2) 在图 2 中，节点 2、3、4、5、6、21、22、23 距离节点 1 较近，这些节点的定位误差尚可接受，但距离节点 1 较远的待测节点 7、8、9、10、11、12、13、14、15、16、17、18、19、20、24、25 的定位坐标与实际坐标的误差很大，特别是节点 16 的定位坐标根本不据有参加价值。而在图 3 中由于每一个三角形限制了其余待测节点距离迭代计算的第一个节点的距离，因此大大提高了定位的精度，使所有待测节点的定位误差都控制在可接受的范围内。

3) 在图 3 中△CDE 的质心附近没有待测节点，所以选择距离质心最近的节点 11 做为迭代计算的第一个节点，明显可以看出这个三角形内所有待测节点的定位误差都比其它三角形内的节点要大的多。

质心算法与改进算法主要是在锚点个数对误差影响方面的对比，图 4 为质心算法与改进算法误差的对比图。可以看到由单纯质心算法需要 40 到 50 个锚点才能使定位误差控制在可接受范围内，这根本不能满足实际应用中对成本的要求。改进算法只需要 7 个锚点即可保证定位误差控制在可接受范围内，由锚点个数的比例可知改进算法在可允许误差范围内大大的减少了质心算法所需的锚点个数，更能满足实际应用中

ABC 算法与改进算法主要是在待测节点个数对误差影响方面的对比，图 5 为 ABC 算法与改进算法误差的对比图。可以看到 ABC 算法在待测节点个数增加时导致的误差累积很严重，呈指数型增长。ABC 算法只可控制 10 个左右待测节点的误差在可接受范围内，当待测节点数继续增加时定位误差急剧增加，当有 25 个待测节点时定位误差更是大到了 10 米以上，这根本不能满足实际应用中对定位精度的要求。改进算法的定位误差随着待测节点个数的增加在小范围内波动，基本稳定在可接受范围内。由定位误差随待测节点个数增加的变化情况可知改进算法更能满足实际应用中对定位精度的要求。

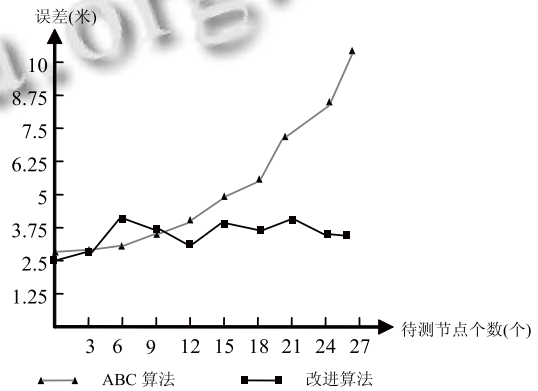

图 5 ABC 与改进算法算法误差对比

实验结果表明使用质心算法需 40 到 50 个锚点才能控制定位误差在可接受范围内，不能满足实际应用中对成本的要求。而改进算法只需 7 个锚点，可以满

© 中国科学院软件研究所 http://www.c-s-a.org.cn

足实际应用中对成本的要求。使用 ABC 算法在待测节点个数个超过 10 个时定位误差急剧增加，不能满足实际应用中对定位精度的要求。而改进算法的定位误差基本稳定在可接受范围内，能满足实际应用中对定位精度的要求。因此得出结论改进算法确实在降低了成本的同时保证了一定的定位精度。

5 结语

本文在对基于锚点的质心算法和与锚点无关的 ABC 算法进行分析的基础上，提出了一种改进的定位算法，它根据实际应用设定的连通度阈值和三角形区域的面积，从而制定出最可行的解决方案。当对定位速度要求高，而对精度的要求并不高时可采用较小的连通度阈值，相反如果对定位速度要求不高，而对精度的要求高时可采用较大的连通度阈值；当要求低成本，而对精度要求不高时可采用较大的三角形面积，相反如果应用中不要求低成本，而对精度要求高时可采用较小的三角形面积。

定位算法的评定要从定位精确度、规模、锚点密度、容错性、自适应性、功耗、节点密度等多方面考虑。在不同的应用中对无线传感器的成本，功耗，使用环境以及定位时间和定位精度的要求各不相同，能够对各项要求进行平衡优化的定位算法必将成为今后研究的热点。

参考文献

1 刘智武,吴威.认知增强型无线传感器节点设计.计算机研究与发展, 2009,46(12):1963−1970.

2 李建中,高宏.无线传感器网络的研究发展.计算机研究与发展, 2008,45(1):1−5.

3 崔莉,鞠海玲,苗勇.无线传感器网络研究进展.计算机研究与发展, 2005,42(1):163−174.

4 高健.无线传感器及其网络研究与实现[硕士学位论文].北京:北京航空航天大学, 2007.

5 裴忠民,邓志东,徐硕,许萧.一种基于 N-最优阶次序列的无线传感器网络节点定位方法.自动化学报, 2010,36(2):199−207.

6 李连,朱爱红.无线传感器网络中的定位技术研究.微计算机信息, 2005,21(9):133−135.

7 陈寒,李仁发,乐光学.无线传感器网络中的一种分布式相对定位算法的分析与改进.计算机应用,2006,26(2):310−313.

8 Michel D, Toggenburger D. Sensor Network Soccer [Ph.D. Dissertation]. University of Applied Sciences Rapperswil HSR, Switzerland, 2006.

9 Zhou ZD, Hu P, Liu Q, Li FM. MDS-based fast localization algorithm for wireless sensor networks. Journal of Ransduction Technology, 2007,20(10):2303−2307.

10 胡圣.室内定位技术的研究[硕士学位论文].重庆:重庆邮电大学,2006.1−53.

11 Bulusu N, Heidemann J, Estrin D. GPS-less low cost out-door localization for very small devices. IEEE Personal Communications Magazine, 2000:7(5):28−34.

12 Savarese C, Rabaey JM, Beutel J. Locationing in Distributed Ad-Hoc Wireless Sensor Networks. Proc. of the 2001 IEEE Int'l Conf Acoustics, Speech and Signal Processing (ICASSP 2001), Piscataway, NJ, USA. 2001: 2037−2040.

（上接第 95 页）

参考文献

1 Patzer A.标定和测量应用的焦点—XCP. [2009-07-30]. http://www.vogel.com.c-n/AUTO/

2 Schuermans R, Zaiser R, Hepperle F, et al. XCP-Part 1-Overview-1.0.Association for Standardization of Automation and Measuring System, 2003.

3 Schuermans R, Zaiser R, Hepperle F, et al. XCP -Part 2-Protocol Layer Specification-1.0. Association for Standardization of Automation and Measuring System, 2003.

4 丁圣彦,罗峰,孙泽昌.基于CCP协议利用CANape进行电控单元标定.电子技术应用,2005,12:5−10.

5 Semiconductor F. MPC563XM Microcontroller Reference Manual,Rev.3. [2009-12]. http://www. freescale.com.cn/

6 苏瑜,周文华,竺春狄.一种适用不同通信方式基于 XCP 协议的 ECU 标定工具开发.汽车工程,2010,32(1):81−85.

7 杨时威,杨林,卓斌.基于 XCP 协议车用标定系统的研发.中国内燃机学会.中国内燃机学会第四届青年学术年会论集,哈尔滨:哈尔滨工程大学出版社,2006.

© 中国科学院软件研究所 http://www.c-s-a.org.cn