

# 无线传感器网络中一种精细距离控制定位算法

张松涛 蒋洪波 唐振华 刘文予

(华中科技大学电子与信息工程系 武汉 430074)

**摘要** 在 Hop-terrain 定位算法的基础上,提出了一个新的改进定位算法。新算法先利用由距离向量交换获得的普通节点与信标节点间的跳数距离进行传感器节点的初始位置估计;然后利用节点初始估计位置和节点直接邻居信息进行节点位置的迭代更新。在更新阶段,为了降低误差,引入了一种精细距离控制机制。该距离控制机制参考到高可信度权值邻居节点的距离约束关系,选择可能的最好节点位置作为最新迭代定位结果。仿真结果表明,与原算法相比,该算法能提高定位节点的比例,降低节点定位误差。

**关键词** 无线传感器网络,定位,距离控制

## Fine-grained Distance Controlled Localization Algorithm in Wireless Sensor Networks

ZHANG Song-tao JIANG Hong-bo TANG Zhen-hua LIU Wen-yu

(Department of Electronics and Information Engineering, Huazhong University of Science and Technology, Wuhan 430074, China)

**Abstract** An improved location discovery algorithm upon previous Hop-terrain localization algorithm was proposed in this paper. At first, the algorithm used distance vector exchange to get hop counts from sensors to anchors and estimate the initial positions of sensor nodes. Then it iteratively updated the sensor nodes' positions by utilizing the initial position estimates and information of immediate neighbors of sensors. A fine-grained distance control mechanism was introduced in the position iteration update phase to decrease the position error. The distance control mechanism chose the possible position with great accuracy as iteration result based on the constraints of the distance to those neighbors with high weight value calculated by our algorithm. Simulation results show that the fraction of located nodes is increased and the position error is decreased compared with Hop-terrain algorithm.

**Keywords** Wireless sensor networks, Localization, Distance controlled

## 1 引言

随着微机电系统技术和无线通信技术的发展,由分布在不同环境的众多小尺寸低功耗多功能的传感器节点组成的无线传感器网络<sup>[1-3]</sup>在诸多方面具有潜在的应用价值。在众多的应用中,如果传感器节点收集到的数据没有相应的位置信息,该数据将失去意义。此外,在一些大型无线传感器自组织网络中,节点位置能为路由<sup>[4]</sup>、边界检测<sup>[5]</sup>、覆盖<sup>[6]</sup>、拓扑控制<sup>[7]</sup>等提供有效帮助。

传感器网络定位技术通常是在少部分信标节点位置已知的情况下,估算网络中绝大多数普通节点的位置。这里的信标节点指事先通过其它手段获得的自身位置的节点。信标节点获得自身位置的主要途径有:信标节点上装有全球定位系统 GPS 或事先将信标节点安装在指定的位置上等。

在基于距离的定位方法中,需要借助一些可选择的测量技术,如信号到达时间、到达时间差、到达角度或接收信号强度等。根据不同的硬件配置及测量复杂度,每种测量技术会有不同的测量精度。如果不采取必要的措施,过大的测量误

差会使节点定位误差太大而失去意义。因此,测量误差是传感器网络节点定位技术需要解决的难题之一。

另一方面,信标节点占所有节点的比例也直接影响节点定位性能的优劣。在平面上,要唯一确定一个未知节点的坐标,需要借助 3 个已知位置的参考节点。在运行定位算法时,通常只有少部分已知自身位置的信标节点可以作为参考节点。由于 GPS 设备比较昂贵、能量消耗较高、在室内环境不能工作等原因,在每个节点上安装 GPS 设备不是一个切实可行的方法。因此,稀疏信标节点问题是传感器网络定位技术中另一个具有挑战性的工作。

在经典的鲁棒性定位算法 Hop-terrain<sup>[8]</sup>基础上,本文提出一种精细距离控制定位算法(fine-grained distance controlled localization algorithms, FDC)。该算法能较好地解决上述测量误差及稀疏信标节点问题。而且,与以前的算法相比,该算法能提高定位节点的比例,降低全网的定位误差。

## 2 相关研究工作

文献[3]对最新传感器网络定位方法进行了较全面的描

到稿日期:2009-06-08 返修日期:2009-08-27 本文受国家 863 计划(2007AA01Z223),国家自然科学基金(60803115,60873127)资助。

张松涛 博士生,主要研究方向为无线传感器网络等,E-mail:stzhang2000@gmail.com;蒋洪波 博士,副教授,主要研究方向为计算机网络、无线传感器网络等;唐振华 博士生,主要研究方向为无线通信、多媒体通信等;刘文予 博士,教授,博士生导师,主要研究方向为计算机视觉、无线通信等。

述。按照是否使用测量距离或角度信息,可将定位方法粗略地分为基于距离定位方法和距离无关定位方法。

基于测距定位通常是用基本的三角测量定位算法来完成对节点的定位。在一个2(3)维网络中,对一个未知节点而言,要完成一个基本的三角测量定位算法,3(4)个已知坐标的参考节点和到相应参考节点的距离是必需的。由于有测距硬件的辅助,基于测距的定位技术能提供相对准确的节点定位估计。Savarese<sup>[9]</sup>等人提出的ABC(assumption based coordinates)算法、Savvides<sup>[10]</sup>等人提出的原子多边形算法、Moore<sup>[11]</sup>等人提出的分簇算法以及Zhou<sup>[12]</sup>等人提出的三维欧氏距离定位算法都是利用直接邻居信息估算自身位置进而得到全局拓扑。这些算法能够实现较准确的节点定位,而且适用于各向异性网络。其主要缺点是需要较多的迭代次数以及迭代过程中存在误差累积。文献[10,12]提出的算法里面还蕴含着较高的节点连通度及较高的信标节点比例的需求。这里,节点连通度指网络中节点的平均邻居个数。

在定位要求不高的情况下,距离无关的定位方法也是一个不错的选择。APIT(approximate point-in-triangulation test)算法<sup>[13]</sup>先以某节点周围可感知信标节点中的任意3个作为顶点构成若干三角形,然后确定该节点是其内点的三角形的交集,得到一个多边形,以该多边形的质心作为该节点的估算位置。该算法受无线信号的不规则性(degree of irregularity,DOD)影响较小。不过,该算法要求有较多的普通节点可感知信标节点数量。REP(Rendered Path)算法<sup>[14]</sup>通过增加虚拟圆解决有洞网络中的节点定位问题。该算法要求,除洞以外的网络,其它部分具有各向同性。

Savarese<sup>[8]</sup>等人提出的鲁棒性定位算法Hop-terrain是一种典型的方法,该方法分为两步:起始阶段和迭代求精阶段。在起始阶段算法利用距离向量消息交换得到各节点之间的跳数,然后结合信标节点坐标估计网络平均单跳距离,进一步估算普通节点初始坐标。迭代求精阶段利用起始阶段得到的节点初始坐标,结合到邻居节点的测量距离运行带权值的三角测量定位算法。对成功运行三角测量定位算法的节点进行到信标节点和到邻居节点的距离检测,如果两项检测都通过,则保留本次迭代结果;否则用在起始阶段得到的估计坐标作为本次迭代结果。上述过程反复迭代以尽量减小定位误差。

Hop-terrain算法在网络连通度比较低时定位节点比例很低。本文提出的FDC算法通过对Hop-terrain算法进行改进,较大地提高了低连通度网络条件下的定位节点比例。同时,网络所有普通节点平均定位精度也有提高。

### 3 FDC 定位算法

FDC定位算法对Hop-terrain的改进源于以下思路。为了提高定位节点比例和定位精度,需尽量保留较好的迭代结果。首先,如果某普通节点在上一次迭代中成功运行三角测量定位计算并通过到信标节点和到邻居节点的距离检测,而本次迭代无法成功运行三角测量定位计算,本文将用上一次迭代结果作为本次迭代结果。然后对每次成功运行三角测量定位计算并通过两项距离检测的节点,用到邻居节点测量距离与计算距离的差值作为基准,选择可能的较好迭代结果作为节点最新迭代位置坐标。因为,当邻居节点定位准确性很高时,比如邻居节点为信标节点,到邻居节点的测量距离与计算

距离的差值能较好地反映节点定位准确性。基于这个原因,本文只选择可信度权值较高的部分邻居节点进行到邻居节点的距离检测。

与Hop-terrain算法相仿,FDC算法也包含两个主要步骤:节点初始坐标估算阶段和采用距离控制的节点坐标迭代更新阶段。两个阶段分别改善稀疏信标节点问题和测量误差问题对节点定位的影响。节点初始坐标估算只在算法开始时运行一次,为节点坐标迭代更新阶段提供一个有效起始点。采用距离控制的节点坐标迭代更新只利用包含信标节点和普通节点在内的直接邻居的相关信息按上述思路进行,该方法提高了节点定位的准确性。下面论述FDC算法的两个主要步骤。

#### 3.1 节点初始坐标估算

节点初始坐标估算总体思路如下:各节点通过距离向量交换得到各信标节点坐标和到相应信标节点的最小跳数。信标节点利用自身坐标和到其它信标节点的最小跳数估算网络平均单跳距离,并将该值在网络中传播。各普通节点收到网络平均单跳距离后将该值乘以到信标节点的最小跳数,以获得到信标节点的估算距离。有了到足够信标节点的估算距离和相应信标节点坐标,各普通节点运行基本三角测量定位算法,得到自身初始坐标估算值。

相对于传感器节点较小的通信半径,整个无线传感器网络的覆盖区域通常要大得多。定位算法开始前,一般只有小部分节点是信标节点。在节点任意分布的网络中,只有极少数普通节点有可能有足够的邻居信标节点,从而有机会运行基本的三角测量定位算法来估算自身位置。

为了克服上述困难,节点初始坐标估算阶段采用普通节点到信标节点的最小跳数乘以网络平均单跳距离来估算普通节点到信标节点的距离。一旦某普通节点获得到足够多信标节点的估算距离,就可以利用这些估算距离及相应信标节点的坐标运行基本的三角测量定位算法来估算自身位置。由普通节点到若干信标节点的距离形成的二次方程组可以整理成形如 $AX=b$ 的线性方程组,然后采用标准的最小均方差估计方法可以得到方程组的解,即普通节点的坐标估计<sup>[15]</sup>。

网络通过消息广播使各节点知晓到各信标节点的跳数,从而估算网络平均单跳距离。节点初始坐标估算开始时,每个信标节点广播一条包含自身位置和跳数为0的消息。该信标节点的直接邻居节点(包括普通节点和信标节点)收到该广播消息后记录该信标节点的坐标及相应跳数值1;然后转发该消息,此消息中的跳数值为1。转发该广播消息的信标节点的二跳邻居,此时一般会收到该转发消息而没有收到该信标节点的直接消息。这些二跳邻居将记录该信标节点的坐标及相应跳数值2并转发该消息。这个过程一直进行下去,直到网络中尽可能多的节点收到尽可能多的信标节点的广播消息。注意,如果某节点是孤立节点,则它可能收不到任何广播消息;类似地,如果整个网络在地理位置上被分成几个簇,那么簇内节点只能收到本簇信标节点的消息,而无法收到其它簇信标节点的消息。通常情况下,对于一个连通网络来说,网络中所有节点均能收到所有信标节点的广播消息。收到信标节点广播消息的节点需要发现到该信标节点的最小跳数,用于估算网络平均单跳距离和节点之间的距离。网络平均单跳距离 $h$ 由式(1)估算:

$$h = \frac{\sum mdis_{ij}}{\sum k_{ij}}, i \neq j \quad (1)$$

式中,  $mdis_{ij}$  和  $k_{ij}$  分别表示信标节点  $i$  和  $j$  之间的测量距离和最小跳数。

网络平均单跳距离同样在网络上广播,各普通节点收到网络平均单跳距离  $h$  后,由式(2)估算到相应信标节点的距离:

$$d_{ij} = h \times k_{ij} \quad (2)$$

式中,  $d_{ij}$  表示节点  $i$  和  $j$  之间的估算距离。

在一个 2(3) 维网络中,如果某普通节点知晓到至少 3(4) 个信标节点的估算距离和相应信标节点的坐标,该节点就可以运行基本的三角测量定位算法来估算自身位置。

采用无效消息机制,一方面可以发现某节点到某信标节点的最小跳数,另一方面可以阻止大量无效消息的传输。对某节点新收到的来自某信标节点的消息,如果在该节点记录有相应信标节点的消息,并且其跳数小于或等于刚收到的消息跳数,则认为新收到的消息为无效消息。如果某节点收到无效消息,则丢弃该消息,不作其它处理。无效消息机制有效地减少了消息传输数量。

在节点初始坐标估算阶段,两个或多个地理位置比较接近的普通节点到相应信标节点很可能有对应相同的跳数,从而有相同的距离估计,使这些节点得到相同的初始坐标估计。这对后续节点坐标迭代更新是不利的。我们可以对初始坐标估计结果增加一个较小的随机噪声,从而有效地避免上述情况的发生,提高节点最后定位的准确性。

节点初始坐标估算完成后,网络中所有或绝大多数普通节点得到了一个初始位置估计,为节点坐标迭代更新提供了一个有效起始点。

### 3.2 节点坐标迭代更新

在节点坐标迭代更新阶段,FDC 算法结合各节点直接邻居测量距离及从节点初始坐标估算阶段得到的节点初始坐标,采用有效距离控制手段,迭代更新各节点坐标,减小各节点定位误差。每一步坐标更新过程如下:

各普通节点广播其在节点初始坐标估算阶段或上一迭代过程中得到的位置估计,接收邻居节点的位置估计以及到相应邻居节点的距离测量,用标准的最小均方差估计方法完成带权值的三角测量定位计算,从而得到一个坐标估计。对于成功运行三角测量定位的节点,进行到信标节点和到邻居节点的距离检测,根据不同情况选择本次迭代坐标、上次迭代坐标或初始位置估算阶段坐标作为最新迭代坐标。其中到邻居节点距离检测只选择有高性价比权值的部分邻居节点,并且采取一定的距离控制策略,以提高定位精度。

在许多情况下,由于到邻居节点的距离约束,新坐标位置将不断趋向于该节点的真实位置。如果定位误差足够小或迭代次数达到某一阈值,迭代终止,最后的定位结果将作为整个算法的定位结果。

下面 4 个因素直接影响着迭代的收敛性和最后定位结果的准确性:初始位置估计的准确性、距离测量误差、信标节点的比例和网络节点连通度。在一个已部署的网络中,后面 3 个因素是由网络决定而不能改变的。所以,提高节点初始位置估算的准确性成为改善最后定位结果的有效手段。而且,坐标更新是一个迭代过程;每一次迭代结果都成为节点下一

次迭代的初始位置。也就是说,每次迭代结果都直接关系到最后定位的准确性。另外,对于已部署网络,虽然硬件测量误差不可改变,但是可以从软件上改变测量误差对定位结果的影响。我们的算法正是从提高初始位置估计的准确性和减小测量误差对迭代的影响两方面来进行迭代控制,提高定位准确性的。

在一个 2(3) 维网络中,如果有超过 3(4) 个信标节点作为参考节点,在运行三角测量定位算法时,测量误差的影响将减小,定位准确性也将提高。

为了减小误差传播,在三角测量定位算法中引入基于节点定位准确性的可信度参数,把线性方程组  $AX=b$  改成:

$$WAX = Wb \quad (3)$$

式中,  $W$  是各节点定位可信度权重向量,各元素取值在 0 和 1 之间。信标节点由于有较高的定位可信度而被分配一个较高的可信度权值;邻居数比较少的普通节点则被分配一个较低的可信度权值,以减小对其邻居节点定位的影响。通过引入可信度权值,坐标更新迭代过程的收敛性和定位结果的准确性都将得到进一步改善。

节点坐标迭代更新开始时,信标节点的可信度权值直接设置为 1;普通节点的权值则设定为接近于 0 的某个值,如 0.1。普通节点设两个标志位  $flag$  和  $flgs$ ,坐标迭代更新前都设置为 0。如果某节点成功运行一次三角测量定位算法,得到一个坐标估计,则将对节点进行两项距离约束检测:到信标节点的距离检测和到邻居节点的距离检测。

到信标节点距离检测先用某普通节点的最新估计位置计算到某信标节点的距离  $cdis_{0i}$ :

$$cdis_{0i} = \sqrt{(x_0 - x_i)^2 + (y_0 - y_i)^2} \quad (4)$$

式中,  $(x_0, y_0)$  和  $(x_i, y_i)$  分别表示待求节点最新估计坐标和某信标节点坐标。然后计算该普通节点到相应信标节点按通信半径的最大约束距离  $kdis_{0i}$ :

$$kdis_{0i} = R \times k_{0i} \quad (5)$$

式中,  $R$  为传感器节点通信半径,  $k_{0i}$  为相应普通节点和信标节点间的最小跳数。如果某普通节点到能收到其广播消息的所有信标节点均满足:

$$cdis_{0i} \leq kdis_{0i} \quad (6)$$

则认为该普通节点到信标节点的距离检测成功;否则认为检测失败。

到邻居节点的距离检测算法伪码如图 1 所示。这里要用到两个阈值:可靠邻居数量阈值,设为  $cnnt$ (图 1 中第 8, 14 和 21 行);高性价比权值邻居数量阈值,设为  $hvwnt$ (图 1 中第 6, 12 和 18 行)。对某普通节点而言,图 1 中 1 到 19 行选择最多  $hvwnt$  个高性价比权值邻居节点,并更新该普通节点到相应邻居节点间的测量距离和计算距离。图 1 中第 20 行按如下方法计算该普通节点到相应邻居节点的测量距离与计算距离的归一化差值  $normd$ :

$$normd = \frac{|\sum mdis_{0i} - \sum cdis_{0i}|}{hwnt} \quad (7)$$

式中,  $\sum mdis_{0i}$  表示该普通节点到相应邻居节点的测量距离之和,  $\sum cdis_{0i}$  表示该普通节点到相应邻居节点按最新估计位置坐标计算的距离之和,  $hwnt$  为选取的邻居节点个数。图 1 中 21 至 28 行判断当选取用于比较的邻居数大于或等于可靠邻居数量阈值  $cnnt$ (比如 2 或者 3) 并且  $normd$  小于某一阈值

(比如传感器节点通信半径  $R$ )时,我们认为到邻居节点的距离检测成功;否则认为检测失败。

```

0 hwnn=0;mdis=0;cdis=0;
1 for all neighbors
2   if neighbor is an anchor //邻居是信标节点
3     更新 hwnn,mdis 和 cdis;
4   if(neighbor is a sensor && flag==1 && flgs=1)//邻居是普通节点
5     更新 hwnn,mdis 和 cdis;
6   if hwnn≥hwnt
7     break;//可用邻居数量足够,停止搜索邻居
8   if hwnn<cnnt
9     for all neighbors
10      if (neighbor is a sensor && flag==1 && flgs!=1)//邻居是普通节点
11        更新 hwnn,mdis 和 cdis;
12      if hwnn≥hwnt
13        break;//可用邻居数量足够,停止搜索邻居
14  if hwnn<cnnt
15    for all neighbors
16      if (neighbor is a sensor & flag!=1)
17        更新 hwnn,mdis 和 cdis;
18      if hwnn≥hwnt
19        break;//可用邻居数量足够,停止搜索邻居
20 按式(7)计算 normd;
21 if(hwnn>cnnt && normd<R)//R 为无线通信半径
22  if (normd<normdt && 用于比较的普通邻居节点 flag 和 flgs 均为 1)
23    flgs=1;
24  if (normd<minnormd)
25    更新节点坐标为本次迭代结果;
26  minnormd=normd;
27 else
28  到邻居节点距离检测失败。

```

图 1 到邻居节点距离约束检测算法

当某普通节点到信标节点的距离检测和到邻居节点的距离检测都成功时,将节点  $flag$  标志置为 1,表示节点定位成功;然后更新该节点定位可信度权值为其邻居节点可信度权值的均值,以此来更新式(3)中的  $W$  值。同时,在节点坐标迭代更新前,算法先保存了一个最小归一化差值  $minnormd$ 。如果最新的归一化差值  $normd$  满足:

$$normd < minnormd \quad (8)$$

则认为本次节点坐标位置估计比以前的好,将节点定位坐标更新为本次迭代估计的坐标,同时将最小归一化差值更新为本次归一化差值,作为后续迭代更新用于比较的基准值(图 1 中 24 到 26 行);否则,不更新节点定位坐标和最小归一化差值。也就是说,此时依然用上次迭代估计坐标作为本次迭代估计坐标。

这里还要解释一下,在到邻居节点的距离检测中部分邻居节点的选取问题(图 1 中 1 到 19 行)。选取的邻居节点定位的准确性越高,最后计算出来的归一化差值越能反映节点定位准确性。所以,对到邻居节点的距离检测,本文算法并不选择所有邻居,而只选择可信度权值比较高的部分邻居。我们用高可信度权值邻居数量阈值  $hwnt$ (比如,网络连通度的

一半)表示每次到邻居节点的距离检测最多需要选取的邻居节点个数。可靠邻居数量阈值  $cnnt$  表示成功进行一次到邻居节点的距离检测需要选择的最少邻居个数。当已选择邻居节点个数小于  $hwnt$  时,按邻居节点可信度权值高低依次选择用于距离比较的邻居节点。信标节点的可信度权值 1 为最高,按上述思路信标邻居节点必然会先被访问,且被直接选取。当邻居节点是普通节点时,如果节点  $flag$  和  $flgs$  均为 1,该普通邻居节点会被选取(图 1 中 1 到 7 行)。如果按此条件把所有邻居节点搜索完后被选取的邻居节点个数不足  $cnnt$  时,重新搜索邻居节点,当邻居节点是普通节点、 $flag$  为 1 而  $flgs$  不为 1 时,该普通邻居节点被选取(图 1 中 8 到 13 行)。如果按此条件把所有邻居节点搜索完后被选取的邻居节点个数仍不足  $cnnt$  时,再一次重新搜索邻居节点,当邻居节点是普通节点并且  $flag$  不为 1 时,该普通邻居节点被选取(图 1 中 14 到 19 行)。如果此时被选取的邻居节点个数还不足  $cnnt$ ,则终止本次到邻居节点的距离检测并标记检测失败。如果只需选取信标邻居节点和  $flag, flgs$  均为 1 的普通邻居节点就能成功完成到邻居节点的距离检测并且检测中计算出的归一化差值  $normd$  小于某一阈值  $normdt$ (比如 0.1 或 0.05) 时,则将该节点的  $flgs$  值置为 1(图 1 中 22 到 23 行)。

最后,对某普通节点而言,有可能在某次坐标迭代更新过程中有足够的可用邻居而成功运行基本的三角测量定位算法,但在下次迭代时又因可用邻居节点不足而无法完成基本的三角测量定位算法。此时,考虑该普通节点的  $flag$  标志。若该标志为 1,则用该节点上次定位坐标作为本次定位结果;若该标志不为 1,则用节点初始坐标估算阶段的估计坐标作为本次定位结果。上次定位坐标的选择性采用,改善了节点定位结果。

随着坐标更新迭代不断进行,越来越多的普通节点将被成功定位,相应的定位可信度权值也不断升高。如果少数节点一直无法成功运行三角测量定位算法或无法成功通过两项距离检测,则给定一个迭代次数阈值以使算法有效终止。

## 4 性能评估

本文搭建实验平台实现了 FDC 算法,并进行了广泛的实验仿真。将 FDC 算法的相关性能,如定位节点的比例、定位准确性,定位误差累积分布等几个方面与 Hop-terrain 算法<sup>[8]</sup>进行了对比分析。

### 4.1 仿真环境设置

本文考虑在一个  $100m \times 100m$  的正方形区域中随机布撒 400 个传感器节点,并在其中任意选择一部分作为信标节点。由于节点定位性能跟信标节点比例和网络连通度有较大关系,我们通过改变网络中信标节点比例和网络连通度——进行了大量的实验。网络连通度的调整通过改变节点无线通信半径来实现。节点之间的距离测量误差按高斯分布考虑,均值为实际距离,标准差取实际距离的 5%。定位误差为节点定位位置与节点真实位置之间的距离。定位节点比例指成功定位的普通节点占全部普通节点的比例。为便于比较,定位误差按节点无线通信半径归一化。如节点通信半径为 8 米,定位误差为 0.8 米,则按通信半径归一化的定位误差为 10%。所有实验结果为 100 次实验的平均值。

### 4.2 仿真结果分析

图 2 是在 5% 的测量误差情况下, 定位节点比例随连通度变化的情况。定位节点比例总的趋势是随着网络连通度的增加而增加。从图 2 中可以看出, FDC 算法的定位节点比例比 Hop-terrain 算法有较大的提高, 尤其在网络连通度比较低即取值为 6 和 8 左右时, 定位节点比例比 Hop-terrain 算法分别提高 50% 和 20% 以上。

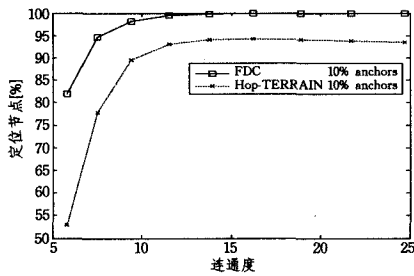


图 2 定位节点比例随连通度变化情况(5%测量误差)

图 3 是在 5% 的测量误差情况下, 定位误差随连通度变化的情况。随着网络连通度和信标节点比例的增加, 普通节点定位误差越来越小。在信标节点比例为 10% 和 20%、连通度在 10 以上时, 定位误差在 15% 以下。从图 3 中可以看出, 在信标节点比例为 10%、连通度在 15 以下时, FDC 算法定位误差比 Hop-terrain 算法略高。这一点可解释如下: 图 3 描述的不是所有普通节点的平均定位误差, 而是网络中成功定位节点的定位误差。但在相同网络连通度和相同信标节点比例情况下, 尤其在连通度比较低的时候, FDC 算法的定位节点比例比 Hop-terrain 算法高很多(图 2 的主要结论)。所以, 只从图 3 中并不能反映在信标节点比例为 10%、连通度在 15 以下的网络条件下, FDC 和 Hop-terrain 算法全网定位误差的优劣。

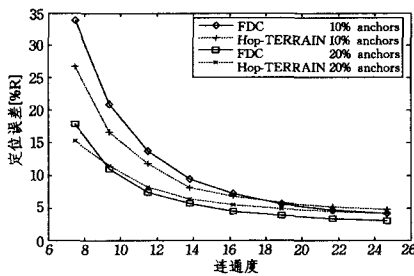


图 3 定位误差随连通度变化情况(5%测量误差)

图 4 和图 5 为累积误差分布图。两个图除信标节点比例分别为 10% 和 20% 外, 其它实验参数都一样。图 5 中, 当信标节点比例为 20%、连通度为 5.8 时, FDC 算法定位误差小于 50% 的节点占有普通节点的比例为 71.3% (Hop-terrain 算法为 61.4%), 比 Hop-terrain 算法高 16%。类似地, 当信标节点的比例为 20%、连通度为 5.8 时, FDC 算法定位误差小于 10% 和 100% 的节点占有普通节点的比例比 Hop-terrain 算法相应要高 42.6% 和 8.6%。图 4 中, 当信标节点比例为 10%、连通度为 5.8 时, FDC 算法定位误差小于 10%, 50%, 100% 的节点占有普通节点比例比 Hop-terrain 算法分别高 53.6%, 20% 和 13%。从图 4 和图 5 可以看到, FDC 算法的定位结果中, 具有较小定位误差的节点比例比用 Hop-terrain 算法获得的具有相同定位误差范围的节点比例高。换言之, FDC 算法定位准确性比 Hop-terrain 算法好。FDC 算法定位准确性的提高, 主要贡献来自于采用精细距离控制的

节点坐标迭代更新阶段。在到邻居节点的距离检测中, 我们并不选择所有邻居节点, 而只选择具有高可信度权值的部分邻居节点, 从而使到邻居节点的测量距离与按估计坐标计算的距离之差更能反映定位的准确性。根据该差值的大小, 选择可能的最好结果作为节点估计坐标, 提高了节点定位准确性。

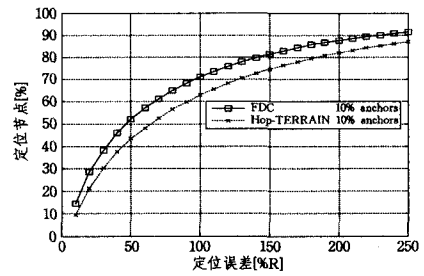


图 4 累积误差分布(5%测量误差, 连通度 5.8)

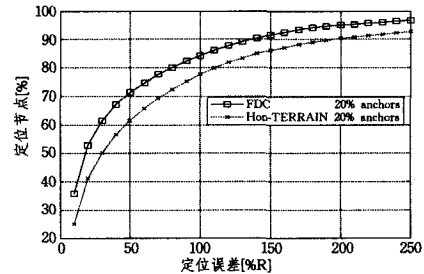


图 5 累积误差分布(5%测量误差, 连通度 5.8)

**结束语** 本文提出一种采用精细距离控制的无线传感器网络节点定位方法, 利用到高可信度权值邻居节点的距离约束关系, 对定位成功并满足一定距离约束的节点, 选择可能的最好结果作为最新定位结果, 能较好地克服测量误差和稀疏信标节点问题给节点定位带来的影响。与以前的算法相比, 定位节点比例和全网节点定位准确性都有提高。

本文选择具有高可信度权值的部分邻居节点, 使到邻居节点的测量距离与按估计坐标计算的距离之差较好地反映了节点定位的准确性。下一步工作将选择更好的反映节点定位准确性的评价体系, 利用该评价体系进行定位控制, 进一步提高节点定位的准确性。

## 参考文献

- [1] 孙利民, 李建中, 陈渝, 等. 无线传感器网络[M]. 北京: 清华大学出版社, 2005
- [2] 任丰原, 黄海宁, 林闯. 无线传感器网络[J]. 软件学报, 2003, 14(7): 1282-1291
- [3] Mao G Q, Fidan B, Anderson B D O. Wireless sensor network localization techniques[J]. Computer Networks, 2007, 51(10): 2529-2553
- [4] Kuruvila J, Nayak A, Stojmenovic I. Hop count optimal position-based packet routing algorithms for ad hoc wireless networks with a realistic physical layer[J]. IEEE Journal on Selected Areas in Communications, 2005, 23(6): 1267-1275
- [5] Ding M, Chen D C, Xing K, et al. Localized fault-tolerant event boundary detection in sensor networks[C] // Proceedings of IEEE INFOCOM, Miami, FL, USA, Mar. 2005

(下转第 54 页)

**推论 1** 设  $f(x) \in B_n$ , 若  $f$  的代数正规型中含有所有形如  $x_{i_1} x_{i_2} \cdots x_{i_t}$  ( $t > 1$ ) 的项, 其中  $i_1, i_2, \dots, i_t \in I \subset \{1, 2, \dots, n\}$ , 则  $NN(f^c) \geq k$ , 其中  $k = \# I$ 。

事实上, 在寻找具体的  $I$  时, 可以利用下面推论的结论。

**推论 2** 设  $f(x) \in B_n, J \subset \{1, 2, \dots, n\}$ , 且对  $f$  的代数正规型中任意项  $x_{i_1} x_{i_2} \cdots x_{i_t}$ , 都有  $\{i_1, i_2, \dots, i_t\} \cap J \neq \emptyset$ , 则  $NN(f) \geq n - k$ , 其中  $k = \# J$ 。

其实只需取定理 3 中的  $I = \{1, 2, \dots, n\} - J$  即可。比如在上例中就可取  $J = \{3, 5\}$ 。

从上面的定理及推论可以看出, 只需从函数代数正规型的每一个单项中取一个下标就可构成  $J$ , 且  $J$  越小越有利于取得正规阶上界。

结合定理 2 与定理 3, 可得到代数正规型与代数免疫阶之间的如下关系。

**推论 3** 设  $f(x) \in B_n$ , 若  $f$  的代数正规型中不含有任意形如  $x_{i_1} x_{i_2} \cdots x_{i_t}$  ( $t > 1$ ) 的项, 其中  $i_1, i_2, \dots, i_t \in I \subset \{1, 2, \dots, n\}$ , 则  $AI(f) \leq n - k$ , 其中  $k = \# I$ 。

如在上面的例 2 中, 就有  $AI(f) \leq 2$ 。类似地还可以得到代数正规型与扩展的正规性及扩展代数免疫阶之间的关系。显然, 据此结论在计算正规阶上界以及代数免疫阶时可以节省大量的计算。比如在上面的例子中, 用我们的方法只需简单计算就可以得出函数的正规阶上界不小于 10 的结论, 而采用常规的正规性判定方法仅寻找所有的 10 维子空间和仿射子空间就有约  $2^{23}$  个。对代数免疫的计算量可类似分析, 但本文的方法可能对某些函数不能取得较好的结果。

**结束语** 本文从正规性这一密码函数的复杂性指标出发, 考虑到在对密码函数作一个小改动时是否仍然具有良好的复杂性, 定义了扩展的正规性。同时研究了扩展正规性这一指标和扩展代数免疫阶之间的关系, 并从代数正规型的角度对密码函数的正规性和代数免疫阶进行了分析, 提供了一条方便易行的思路。

## 参 考 文 献

[1] Dobbertin H. Constructions of bent functions and balanced Boo-

lean functions with high nonlinearity [C] // Fast Software Encryption, Lecture Notes in Computer Science. Springer-Verlag, 1994, 1008: 61-74

[2] Carlet C. On the complexity of cryptographic Boolean functions [C] // Sixth International Conference on Finite Fields and Applications, Lecture Notes in Computer Science. Berlin: Springer, 2002: 53-69

[3] Zhang X M, Josef P, Zheng Y. On Algebraic Immunity and Annihilators [C] // ICISC 2006, LNCS 4296. Berlin Heidelberg: Springer-Verlag, 2006: 65-80

[4] Daum M, Dobbertin H, Leander G. An algorithm for checking normality of Boolean functions [C] // Proceedings of the 2003 International Workshop on Coding and Cryptography (WCC 2003). Versailles, France, 2003: 133-142

[5] Braeken A, Wolf C, Preneel B. A randomised algorithm for checking the normality of cryptographic Boolean functions [C] // IF-IP TCS. Kluwer, 2004: 51-66

[6] Courtois N, Meier W. Algebraic attacks on stream cipher with linear feedback [C] // Advances in Cryptology-EUROCRYPT 2003, LNCS 2656. Springer Verlag, 2003: 345-359

[7] Meier W, Pasalic E, Carlet C. Algebraic attacks and decomposition of Boolean functions [C] // Advances in Cryptology-EUROCRYPT 2004, LNCS 3027. Springer-Verlag, 2004: 474-491

[8] 张卫国, 丁勇, 张宁, 等. 代数免疫布尔函数的一个特征[J]. 北京邮电大学学报, 2007, 30(5): 55-57

[9] Dalai D K, Gupta K C, Maitra S. Results on Algebraic immunity for cryptographically significant Boolean functions [C] // INDOCRYPT 2004, LNCS 3348. Springer-Verlag, 2004: 92-106

[10] Dubuc S. Etude des proprietes de degenerescence et de normalite des fonctions booleennes et construction de fonctions q-aires parfaitement non-lineaires [D]. Universite de Caen, 2001

[11] Zheng Y, Zhang X M, Imai H. Restriction, terms and nonlinearity of Boolean functions [J]. Theoretical Computer Science, 1999, 226(1): 207-223

(上接第 40 页)

[6] Huang C, Tseng Y. The coverage problem in a wireless sensor network [J]. Mobile Networks and Applications, 2005, 10(4): 519-528

[7] Li N, Hou J C, Sha L. Design and analysis of an MST-based topology control algorithm [J]. IEEE Transactions on Wireless Communications, 2005, 4(3): 1195-1206

[8] Savarese C, Rabaey J, Langendoen K. Robust positioning algorithms for distributed ad-hoc wireless sensor networks [C] // Proceedings of the USENIX Technical Annual Conference. Monterey, CA, USA, June 2002

[9] Savarese C, Rabaey J M, Beutel J. Locationing in distributed ad-hoc wireless sensor networks [C] // Proceedings of IEEE ICASSP. Salt Lake City, UT, USA, May 2001

[10] Savvides A, Han C C, Strivastava M B. Dynamic fine-grained localization in ad-hoc networks of sensors [C] // Proceedings of

ACM MobiCom. Rome, Italy, July 2001

[11] Moore D, Leonard J, Rus D, et al. Robust distributed network localization with noisy range measurements [C] // Proceedings of ACM SenSys. Baltimore, MD, USA, Nov. 2004

[12] Zhou Z, Cui J H, Zhou S L. Localization for large-scale underwater sensor networks [R]. UbiNet-TR06-04. Computer Science & Engineering dept, University of Connecticut, 2006

[13] He T, Huang C D, Blum B M, et al. Range-free localization schemes for large scale sensor networks [C] // Proceedings of ACM MobiCom. San Diego, CA, USA, Sep. 2003

[14] Li M, Liu Y H. Rendered path: range-free localization in anisotropic sensor networks with holes [C] // Proceedings of ACM MobiCom. Montreal, Quebec, Canada, Sep. 2007

[15] Golub G. Matrix computations (3<sup>rd</sup> edition) [M]. The Johns Hopkins University Press, 1996