

双层协作的无线传感器网络目标跟踪方法

于涛¹, 孙燕^{1,2}

(1. 南京师范大学计算机科学与技术学院, 南京 210046; 2. 江苏省信息安全保密技术工程研究中心, 南京 210097)

摘要: 通过在无线传感器网络中引入双层监测机制, 提出一种基于象限定位的目标跟踪方法。该方法依据位置关系选取相关性高的内外层节点, 利用外层节点的快速侦测和内层节点的准确定位对移动目标进行定位跟踪, 同时向汇聚节点快速上传移动目标轨迹。理论分析和实验结果表明, 该方法可以降低无线传感器网络功耗, 提高目标定位精度。

关键词: 无线传感器网络; 协作感知; 双层监测机制; 目标跟踪; 外层侦测机制; 内层定位机制

Double-circle Cooperative Target Tracking Method in Wireless Sensor Network

YU Tao¹, SUN Yan^{1,2}

(1. School of Computer Science and Technology, Nanjing Normal University, Nanjing 210046, China;

2. Jiangsu Research Center of Information Security & Privacy Technology, Nanjing 210097, China)

【Abstract】 By introducing the double-circle monitoring mechanism into Wireless Sensor Network(WSN), this paper proposes a target tracking method based on quadrant location. It selects the high-relative nodes based on the relation of the nodes positions, locates the mobile targets which are rapidly detected by the outer circle and accurately located by the inner circle, and uploads the mobile target track to the sink simultaneously. Theoretical analysis and experimental results show that the method can reduce the power consumption of WSN and increase the locating precision.

【Key words】 Wireless Sensor Network(WSN); cooperative sensing; double-circle monitoring mechanism; target tracking; outer layer detection mechanism; inner layer location mechanism

DOI: 10.3969/j.issn.1000-3428.2011.17.022

1 概述

目标跟踪是无线传感器网络(Wireless Sensor Network, WSN)最具代表性的应用之一, 在环境监测、生境监测等诸多领域有广阔的应用前景^[1-3]。无线传感器网络的目标跟踪监测机制是指: 在监测区域内随机部署大量的传感器节点, 移动目标进入监测区域后, 周围的传感器节点通过一定的唤醒机制使在移动目标运动方向上的节点及时加入跟踪过程, 将目标移动轨迹报告给远处的监控中心; 当移动目标远离节点或无移动目标出现时, 多数传感器节点处于休眠状态。

传感器网络的目标跟踪实质上是传感器节点协作跟踪的过程^[1], 传感器网络的目标跟踪过程主要包括侦测、定位和通告 3 个阶段^[2]。由于节点数量众多, 数据存储量较大^[3], 通常在已知位置情况下, 采用分布式方法处理^[4], 国内外学者对此开展了大量研究。文献[5]提出了信息驱动的方法, 该方法采用角度和距离传感器协作计算, 尽管精度较高, 但传感器成本过高, 不利于使用推广。文献[6]提出了无线传感器网络动态护送树协同目标跟踪方法 DCTC, 该方法考虑了节点之间的协作, 但由于协作树的所有节点长时间处于工作状态, 且树的外边缘存在大量对精度贡献较小的节点, 因此网络的整体消耗过大。目标跟踪除了完成基本目标跟踪外, 还面临着监测系统低能耗运行的挑战。

本文提出一种新的无线传感器网络目标跟踪方法, 通过设计双层协作结构和方位逼近定位优化算法, 引入象限逼近点作为定位节点, 使双层圆环配合达到外层监测和内层定位的效果, 相互协作完成目标跟踪定位。

2 双层协作模型

本节介绍无线传感器网络的监测对象和环境, 并在基本网络模型的基础上引入双层协作机制, 用于分析移动目标的发现方法以及移动目标的位置度量。

2.1 基本假设

本模型的网络基本假设如下: (1)移动目标具有与无线传感器节点通信的能力、少量的存储空间和计算能力。(2)检测区域内由大量同类传感器节点组成, 所有节点通过定位技术已知自己的位置。节点感知范围是以 R 为半径的圆形区域, 节点通信范围是以 $2R$ 为半径的区域。(3)分布式跟踪。对移动目标在时间或距离间隔上做离散化处理, 将连续移动的目标轨迹转换为离散的踪迹点, 如图 1 所示的双层圆环跟踪簇。

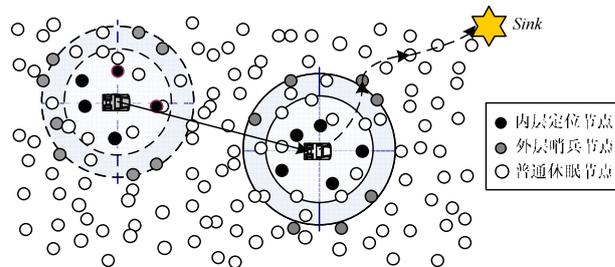


图 1 基于双层监测机制的目标跟踪 WSN

作者简介: 于涛(1985—), 男, 硕士研究生, 主研方向: 无线传感器网络, 嵌入式系统; 孙燕, 副教授

收稿日期: 2011-01-21 E-mail: fish_yt@126.com

2.2 跟踪区域的移动模型

典型的移动模型需要明确 3 个基本要素：移动目标，监测区域和跟踪区域，本文对此作如下定义。

定义 1(移动目标 $Target$) 移动目标指在监测区域中移动的物体，可被描述为离散化的时间点 $Timing_i$ 、离散踪迹点坐标 P_x, P_y 、与前踪迹点之间距离表明速度概念的 x, y 分量 V_x, V_y ，记作： $Target\{Timing_i, P_x, P_y, V_x, V_y\}$ 。

定义 2(监测区域 MA_i) 监测区域部署大量感知节点 N_i 和汇聚节点 $Sink_j$ ，用于跟踪和定位目标位置的整体区域。

定义 3(跟踪区域 TA_i) 跟踪区域是移动目标踪迹离散化后的簇区域。跟踪区域是以 $Target$ 为圆心、以 N_i 检测半径为半径的圆区域。

观察者用时间间隔 Δt 获得运动踪迹，假设移动目标做匀速直线运动，速度为 V ，2 个跟踪区域的间隔是 d_i ，构成的踪迹集合为 $\{d_i, 2d_i, \dots, Md_i\}$ ，则有：

$$d_i = \frac{V \times \Delta t}{k \times R_{max}} \tag{1}$$

其中， R_{max} 是节点最大通信半径； k 是可调功率因子。监测的跟踪区域面积为可调整的，计算公式如下：

$$S_{TA(T)} = \pi \times R^2 \approx \pi \times d_i^2 \tag{2}$$

为了达到定位和监测的目的，跟踪区域的节点数量用 $NodeNumber$ 表示，按照“监测 4 个方位和定位节点 3 个”的要求，监测区域节点数量至少要达到以下要求：

$$\min(NodeNumber) \geq 4 \tag{3}$$

2.3 象限逼近点的内层定位机制

本文提出的象限逼近点的内层定位算法其思路是优选了参与定位的节点，下面将详细介绍其定位机制。

定义 4(监测节点集合 NM_i) 在跟踪阶段的 t 时刻，监测节点集 NM_i 是跟踪区域 TA_i 内的节点总和。每个跟踪区域 TA_i 内的节点包括用于监视的外层节点 NO_i 、定位的内层节点集合 NI_i 和其他休眠节点 ND_i ，即 $NM_i = \{NI_i, NO_i, ND_i\}$ 。

定义 5(内层定位节点集 NI_i) 内层节点用于质心定位。内层定位节点满足下面 2 个条件：(1)以 $Target$ 为圆心、 r 为半径($r < R_{max}$)的内圆中的所有节点。(2)以 $Target$ 为圆心划分 4 个象限，分别取 4 个象限中最靠近移动目标的传感器节点为内节点。若其中一个象限中没有节点存在，继续扩大半径，选择逼近圆心的节点。

引理 内层定位节点定位误差小于等于移动目标到最远点位置的一半。

证明：如图 2 所示，构造 4 个内层定位节点的 Voronoi 图，Voronoi 图的边是内层定位节点与移动目标连线中垂线，因此，Voronoi 图内部点都满足定位误差小于等于移动目标到最远点一半的要求。证毕。

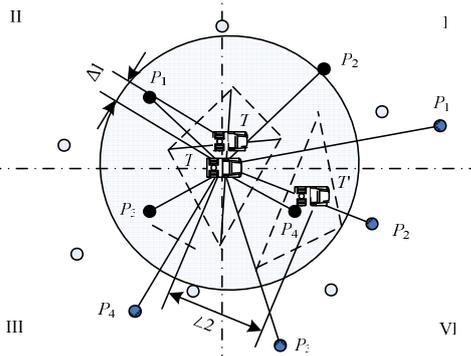


图 2 定位参考节点的选择

内层节点选择好之后，定位误差可由引理确定范围。移动目标的位置用式(4)计算得到：

$$Target(P_x, P_y) = \sum_{i=1}^4 \frac{w_i P_i(x_i, y_i)}{4} \tag{4}$$

其中， P_1, P_2, P_3, P_4 为经过选择而获得的内层定位节点； w_i 是由接收移动目标信号强度 RSSI 得到偏差权值。如图 2 所示的质心位置 T 就是本文方法的定位结果。

换个角度分析定位问题：(1)若对跟踪集合 TA_i 的定位节点不采取象限逼近方法，而是随机选择节点进行定位，定位结果有较大的不确定性，如随机选择 P_1', P_2', P_3' 和 P_4' ，则误差较大。(2)跟踪区域内参与定位的节点越多，定位精度越高，但精度的提高有限，能耗却会大大增加。

2.4 方位识别的外层侦测机制

定义 6(外层侦测节点集 NO_i) 外层侦测节点用于发现移动目标进入或离开监测区，其需要满足下面 2 个条件：(1)以移动目标 $Target$ 为圆心、 R_{max} 为感知最大半径、 r 为内层圆半径、半径大于 r 同时又小于 R_{max} 的圆环区域。(2)以移动目标 $Target$ 为圆心，将满足条件(1)中的节点划分成坐标象限的四方位，并将其区域的节点构成外层侦测节点集合 NO_i 。

定义 7(重构条件) NO_i 中任意象限中所有节点同时检测不到 $Target$ ，则向对应象限的内层定位节点发送重构信息，重构该网络。

图 3 为监测重构示意图。基于外层哨兵重构的测距范围如下：

$$d \leq R_{max} - \epsilon_p, \epsilon_p = D_RSSIMax(SP_1, SP_2, SP_3, SP_4) \tag{5}$$

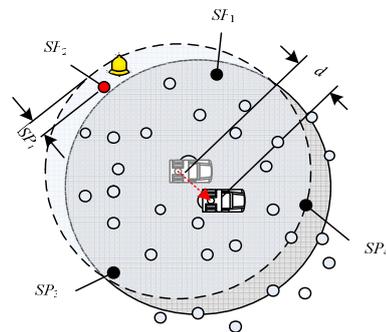


图 3 监测重构示意图

2.5 具有休眠机制的监测网络

定义 8(通告根节点 N_{root}) 通告是将跟踪定位信息传输给 $Sink$ 节点， N_{root} 节点是跟踪区域的发送信息的初始节点，从 N_{root} 节点开始形成一条通往汇聚节点 $Sink$ 的路径。

当监测网络重构之后，跟踪区域内的所有节点被唤醒，休眠机制要完成 3 项工作：

- (1)以跟踪传输节点 N_{root} 为中心，唤醒广播，使 R_{max} 内的所有节点处于跟踪区域，跟踪区域外的节点以周期实施监听。
- (2)以跟踪传输节点 N_{root} 为源点，寻找到达最近的汇聚节点 $Sink$ 的路径。
- (3)对处于监测区域中的各种节点进行状态更新，除外层、内层节点和转发节点以外的所有节点转为休眠状态。

为了明确节点能耗情况，下面对休眠节点情况作进一步考量。假设以 d 为最小监测距离，对于均匀部署的情况，网络节点单位面积覆盖率 ρ 关系如下：

$$\rho = k \times \frac{(R_{max} + 1)^2}{R_{max}^2} \geq M_{Thr} \tag{6}$$

其中， k 为节点平均距离的系数； M_{Thr} 为密度阈值。

设 Num_{mid_Node} 表示上传定位信息的激活节点, 跟踪区域休眠节点的数量为:

$$Num_{Dro-Node} = \rho \times \pi R_{max}^2 - (Num_{mid-Node} + 8) \quad (7)$$

ρ 值越大, 对移动目标的识别率越高, $Num_{dro-Node}$ 越大, 休眠节点越多, 因此, 带这种休眠机制的监测网路具有很好的扩展性。

3 双层协作算法

双层协作跟踪机制要经历外层侦测、内层定位和通告路径树 3 个基本步骤, 监测区域内的传感器节点将经历休眠—工作状态变化。

3.1 双层协作监测机制

对整个网络而言, 双层协作监测机制包括以下 5 个步骤:

步骤 1 系统初始化。每个节点得到全网唯一身份编号、位置、能量以及从每个节点通向最近汇聚节点的通路、移动目标初始化时间间隔或与运动速度相关的距离。

步骤 2 全网侦测部署。由汇聚节点部署最外层的节点轮流进行侦测, 设置轮流侦测的时间段和节点顺序, 每个节点获得“休眠”与“工作”的时间表。

步骤 3 跟踪区域唤醒。在 t 时刻, 移动目标发出唤醒“跟踪关联”信息, 跟踪区域内的节点收到定位信号后, 由“休眠”转入“待命”状态。

步骤 4 计算跟踪集合及定位。移动目标根据跟踪区域的位置, 计算出跟踪簇的根节点、内层定位节点 NI 和外层的侦测节点 NO , 移动目标再次发出“跟踪信息”的信息, 跟踪区域内节点计算出移动目标当前的位置, 并通告汇聚节点。

步骤 5 判断重构条件。当移动目标移出了跟踪区域且在时间间隔内时, 满足重构条件, 移动目标发出“跟踪取消”, 转步骤 3。

3.2 内层象限逼近定位算法

移动轨迹点的坐标是每个象限逼近点的几何中心, 具体算法如下:

算法 1 内层象限逼近定位算法

输入 跟踪区域内所有节点集合 NM_t

输出 移动目标的位置 $Target(P_x, P_y)$, 跟踪簇根节点 N_{root}

```

NI-localization()
for i=1 to Len(NMt)
Calc Node-quadrature()
Sort According to the quadrant sequence
End for
Define NI&Not-aggre()
Centroid-localization()
Select Nroot
Return Target(Px, Py), Nroot
    
```

3.3 外层节点侦测识别算法

移动目标移出外层监测节点的范围时, 外层节点可判断移动目标的方位及是否需要重构计算轨迹。具体算法如下:

算法 2 外层侦测节点识别算法

输入 跟踪区域内所有节点集合 NM_t , 移动目标原先位置接收到的场强 RSSI 信号

输出 移动目标重构信息 Recon-Info

```

NO-Recognition()
Broadcast("Trace Message")
New-NMt; Receive from neighbor ID
If (Not ∩ New-NMt) <> Not
Return Recon-Info=T
    
```

```

Else
Return Recon-Info=F
    
```

4 仿真实验与结果分析

为了验证双层协作机制的有效性, 本文用 VC++ 模拟移动目标在不同监测环境下的运动轨迹, 实验场景设置如下: 监测区域为大小可调的矩形, 长 100 m~1 000 m, 宽 50 m~700 m; 实验的起始位置通过鼠标随机产生, 通信半径在 150 m~250 m 之间可调, 汇聚节点坐标为 (1 000, 50), 移动目标按照光标指定的任意位置做定位计算。

4.1 适应性分析

适应性是指对监测区域节点数量的要求程度与定位精度关系。实验结果如表 1 所示。从中可知, 当节点数量增加到一定时刻, 本文方法与 DCTC 的定位精度基本相同, 但本文方法在计算时间和能耗方面具有明显优势。

表 1 不同节点总数下的定位精度比较 m

方法	50 个节点	70 个节点	90 个节点	100 个节点	500 个节点	800 个节点
本文方法	11	10	8	6	5	5
DCTC	10	7	5	5	4	4

4.2 能耗分析

假设在 100 m×100 m 的区域内随机部署 200 个节点, 移动速度为 10 m/s, 测试 10 次取平均能耗。假设休眠节点能耗为 0.001 μJ, 工作节点能耗为 0.3 μJ, 收发节点的能耗为 1.12 μJ。实验结果如表 2 所示, 从全网节点工作状态分析, 本文方法的工作能耗只有 DCTC 的 28.9%。

表 2 目标跟踪时的节点参与度

方法	休眠节点总数	工作节点总数	收发节点总数	平均能耗/μJ
本文方法	192	8	8	11.55
DCTC	172	28	28	39.93

平均重构时间是指随着移动目标的移动, 需要重新构建网络实现对移动目标跟踪的时间, 重构次数越频繁, 网络存活时间越短。2 种方法的比较结果如图 4 所示。可以看出, DCTC 的平均重构时间为 8.5 s, 本文方法为 12.4 s, 按速度归一化计算可得, 由于重构时间间隔的延长, 本文方法比 DCTC 的能耗少 31.3%。

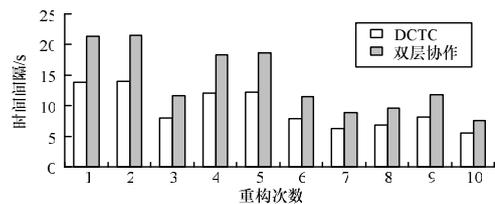


图 4 重构跟踪簇时间间隔比较

4.3 轨迹定位精度分析

在上述实验条件下, 对轨迹定位精度采用 DCTC、本文方法和随机定位方法作定位精度比较, 实验结果如图 5 所示。可以看出, 本文方法与随机定位方法相比具有明显的优势。

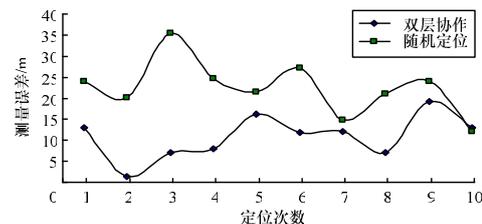


图 5 轨迹定位精度比较