

无线传感网中基于成对节点探测不确定性的目标容错跟踪方法

谢 羿 黄奇珊 张汇川

(西南电子电信技术研究所 成都 610041)

摘要 针对无线传感器网络(Wireless Sensor Networks, WSN)中的节点在真实环境中的不可靠感知现象及其对目标跟踪精度的影响,首先分析计算了成对传感器节点感知存在的不确定区域及其边界,在此基础上,提出了一种基于成对节点探测不确定性的目标容错跟踪方法(Tracking with Pairwise Uncertainty of RSSI, TPU-RSSI),即通过匹配分组感知采样得到的感知向量(sampling vector)和跟踪区域划分面(face)的特征向量(signature vector)来进行移动目标容错跟踪。该方法在保持跟踪方法灵活性的基础上,能够减小由环境因素带来的跟踪误差。为了降低计算复杂度,提出了一种基于邻居面连接的启发式匹配算法。大量的仿真实验结果均表明,所提方法相比同类的其他方法具有更强的灵活性和更高的定位精度。

关键词 目标跟踪,不可靠感知,不确定区域,无线传感器网络

中图分类号 TP301 **文献标识码** A

Tracking with Pairwise Uncertainty of RSSI Based on Pairwise Sensing Uncertainty in Wireless Sensor Networks

XIE Yi HUANG Qi-shan ZHANG Hui-chuan

(Southwest Electronics and Telecommunication Technology Research Institute, Chengdu 610041, China)

Abstract Focusing on the unreliable sensing phenomenon in wireless sensor networks and its impact on target-tracking accuracy, this paper first analyzed the uncertain area caused by the uncertainty of pairwise sensing results and its boundaries. Then the tracking with pairwise uncertainty of RSSI (TPU-RSSI) strategy was proposed. Hence, the tracking problem is transformed into a vector matching process, which matches the signature vector of divided faces with the sampling vector formed by the burst grouping samplings, in order to improve the tracking flexibility, increase the tracking accuracy and reduce the influence of in-the-filed factors. In addition, a heuristic matching algorithm was introduced to reduce the computational complexity. The experiment results show that TPU-RSSI is more flexible and has higher tracking accuracy than related methods.

Keywords Target-tracking, Unreliable sensing, Uncertain area, Wireless sensor networks

1 引言

无线传感器组成的自组织网络——无线传感器网络(Wireless Sensor Networks)已成为信息采集、存储、传递的重要手段^[1]。无线传感器网络的一种典型应用是移动目标的跟踪,例如在偏远地区(remote area)或重要军事区域,人为或随机地部署若干传感器节点,通过传感器节点感知非合作目标的物理信号和事件,协同地进行入侵目标检测、实时位置获取以及目标轨迹跟踪等,完成如抢险救灾、入侵监测、目标追踪等特定任务。

近年来,出现了许多用于无线传感器网络目标跟踪的方法^[2-7],但它们都有各自的缺点和不足。基于测量的目标定位跟踪方法^[8-10],主要通过测量目标与传感器节点的距离或者传感器节点接收到目标发射信号的角度来进行跟踪,这些方法通常都要求附加的硬件设备^[9]或者需要对环境的配置进行

详细定义^[8,10],因此较难以实现。另一类比较普遍的跟踪方法是通过连续的位置估测^[3]、速度估测^[5,6]或轨迹估测^[11],利用目标的移动模型、估测模型或滤波模型^[12,13](例如,卡尔曼滤波^[14]、粒子滤波^[15]、贝叶斯网络^[16]、变分滤波^[17]等)进行目标跟踪。然而,这类方法通常需要对移动目标的移动模型进行假设和详细描述,因此在实际应用中显得比较复杂且缺乏灵活性。最近,出现了基于平面划分和节点序列匹配的目标定位跟踪方法^[18-20],这些方法将跟踪区域划分为若干面(faces/patches)作为候选的目标位置,通过反映目标同传感器节点间相对位置关系的节点序列来实现目标跟踪。这样,目标定位跟踪过程就转化为一个序列匹配的过程,而不需要对目标的移动模型进行假设或者精确地测量距离或角度。虽然这类方法显得更加灵活、普适,但其在真实环境的实际应用中,会出现两个无法回避的问题,即:1)对于单次定位,定位误差较大;2)对于连续定位,目标的位置来回跳动,不能得到

收稿日期:2013-04-17 返修日期:2013-07-22

谢 羿(1984—),男,博士,工程师,主要研究方向为无线传感器网络、目标跟踪, E-mail: xieyi310@163.com; 黄奇珊(1979—),女,博士,高级工程师,主要研究方向为宽带信号处理; 张汇川(1988—),男,硕士,助理工程师,主要研究方向为统计信号处理、目标定位。

一个平滑的移动轨迹。

文献[21,22]对真实环境中不可靠感知和不确定区域的问题进行了详细阐述,很好地揭示了以上两个问题的本质,即,目标和传感器节点之间的真实距离和探测模型计算出的距离存在差异。因此对于成对的节点,存在一个不确定区域,当目标出现在这对节点的不确定区域中时,目标与这两个节点距离的相对关系(目标距离哪个传感器节点更近)无法确定。以此为出发点,论文提出了一种基于成对节点探测不确定性的目标容错跟踪方法(Tracking with Pairwise Uncertainty of RSSI, TPU-RSSD),其核心思想是:在分析计算成对传感器节点探测不确定区域及其边界线的基础上,利用其相互交叠将跟踪区域划分为若干个具有唯一特征向量(signature vector)的划分面(faces);当进行移动目标的定位时,通过感知节点进行突发的分组感知,通过比较得到成对节点的感知结果的大小关系是有序还是发生翻转,构建反映目标位置的感知向量(sampling vector);然后,通过匹配这两个包含不确定信息的向量来进行移动目标定位跟踪,以在保持跟踪方法灵活性的基础上,大大减小由环境因素带来的跟踪误差,提高跟踪精度。此外,该方法通过构建邻居面连接集合,设计了一个启发式的匹配算法,有效地控制了算法的时间和空间复杂度。最后通过仿真实验验证了 TPU-RSSI 算法能够在宽松的运行条件和有限的复杂度下实现移动目标的高效跟踪。

2 相关工作

如前所述,距离相关和模型相关的目标跟踪方法和本文的方法并非紧密相关。鉴于文章篇幅,这里仅对与本文方法直接相关的基于平面划分和序列匹配的目标跟踪方法和面向不确定性的目标跟踪方法进行比较和综述。

典型的基于平面划分和序列匹配的目标跟踪方法包括:基于序列的目标定位 Sequence-based methods 方法^[18];多序列匹配目标定位 Multi-Sequence Positioning, MSP 方法^[19];直接序列最大似然匹配 Direct MLE 方法^[20]和基于路径的最大似然匹配 Path-based MLE 方法^[20]。Sequence-based 算法^[18]利用参与跟踪传感器节点的空间关系将整个跟踪区域划分为多个区域,每个划分区域都可以通过一个序列唯一表示目标与跟踪节点之间的距离的排序,在目标跟踪时,通过某一方向的扫描实现目标的定位和跟踪。在此基础上,MSP 定位算法^[19]为了提升 Sequence-based 算法的跟踪性能,利用不同方向的扫描确定目标和各个感知节点的相对位置关系,每次扫描都能够得到一个序列,通过对多次扫描的序列集合的序列进行迭代计算,逐步将目标确定在一个很的小区域内,并将该区域中心作为目标位置。Direct MLE 方法^[20]通过采样得到每个传感器节点对目标的感知强度大小,将其排序得到一个感知强度节点序列;通过利用节点两两之间的中垂线,将平面划分为若干个具有唯一特征节点序列的面(face);将每一次采样的感知强度节点序列(目标信息)与面的节点序列(位置信息)进行匹配,即可实现对目标的跟踪。Path-based MLE 方法^[20]是在 Direct MLE 方法的基础上提出的,通过假设目标的最大移动速度,确定各个划分面的递推关系,并在整个跟踪区域建立邻居图,以提高定位跟踪的可靠性。

基于平面划分和序列匹配的目标跟踪方法不需要定量刻画目标和传感器节点的绝对距离,也不需要设置目标和环境的模型;它通过传感器之间对移动目标的感知的比较来进行定位,具有很高的灵活性和实用性,但以上方法都存在各自的问题。Sequence-based 和 MSP 方法只针对静止的传感器节点或目标进行讨论,其定位跟踪效果受到复杂环境不确定性的强烈影响。Direct MLE 方法没有考虑到传感器单次感知的误差,其每个感知序列并不一定能真实准确地反映目标和各个传感器节点之间的相对位置关系,从而导致对相对距离比较结果的误判。Path-based MLE 虽然通过假设目标的最大移动速度来提高跟踪方法对不确定性的容忍能力,但在真实系统运行时目标的最大移动速度很难被估计,使得该方法实现比较困难;同时,邻居图的建立也大大地增加了目标定位跟踪的空间复杂度。

为了处理现有的大多数跟踪方法面临的目标跟踪不确定性影响的问题,近年来也出现了一些面向不确定性的目标定位跟踪方法。例如:CDL^[23]借助邻居跳数,采用匹配和邻居序列匹配的办法实现本地过滤,以保证较高的定位感知质量,有效地降低不确定性对目标定位跟踪的影响;PkNN 算法^[24]考虑到目标跟踪中存在不确定性的情况,通过建立机会 k 近邻模型(probabilistic k -nearest neighbors)来提高目标定位的精度和效能,并利用 PkNN 回溯算法选取参与目标定位的活动节点,以达到能量高效的目的。事实上,上述方法主要是通过剔除、修正低质量、不可靠的探测(感知)结果来提高目标定位跟踪的性能,却忽略了不确定、不可信的感知结果也能够作为提高定位跟踪精度的有用附加信息。受到近年来关于不可靠感知^[21]和不确定区域^[22]问题相关研究成果的启示,论文提出的 TPU-RSSI 方法通过多次分组感知采样来克服真实感知环境下由各种环境因素带来的探测不确定性,并利用成对节点的不确定区域边界对感知区域进行划分。因此,该方法能够健壮、准确地在不可靠感知环境下对移动目标实施高精度的定位跟踪。另外,与上述方法的条件约束相比,TPU-RSSI 方法不需要任何假设条件作为约束,体现出更强的灵活性和普适性。

3 问题描述

本节首先针对给定的节点对讨论是否存在成对节点的不确定区域;在此基础上,以对数噪声模型为例,计算不确定区域的边界。

3.1 不确定区域假设

在真实环境中,由于跟踪环境中的网络延迟、离散采样、感知分辨率和环境噪声等各种不确定因素,节点的感知信息总是不可靠的,完全精确的感知只能在理想状态下得到。因此,节点的接收信号强度并不能精确反映目标与节点之间的距离。可以把单个传感器节点对移动目标的感知的不确定性作如下描述:假设目标距离传感器节点的距离为 d ,接收信号强度 RSS 存在一个固有误差 ϵ ,此时传感器节点的 RSS 将分布于 $[rs_s^{\wedge} - \epsilon, rs_s^{\wedge} + \epsilon]$ 中,其中 rs_s^{\wedge} 为理想情况下传感器节点的 RSS 值。因此,当传感器节点的 RSS 值为 rs_s^{\wedge} 时,传感器

节点与移动目标同样分布在 $[d-\delta, d+\delta]$ 区间内。为了消减不确定因素的影响,可以对单个节点接收信号强度的读值进行校正,但这类方法通常比较复杂,且校正模型难以真正地刻画探测的不确定性^[21]。

通过节点间的 RSS 比较并确定其相对大小关系,能够有效抑制不确定性对目标感知的影响^[22]。通过实验比较两节点($node_1, node_2$)间的接收信号强度(RSS),并通过降序排列两节点对的 RSS 值,可建立一个探测节点序列。如图 1 所示,当目标在两节点的垂直平分线附近出现(但事实上更靠近其中一个节点)时,返回的节点序列会随时间变化而不断变化。因此,诸如文献[20]中垂直平分线的划分和单次感知信号强度比较的方法无法得到确定的节点探测序列,这类跟踪算法在引入不确定性后存在严重的问题。

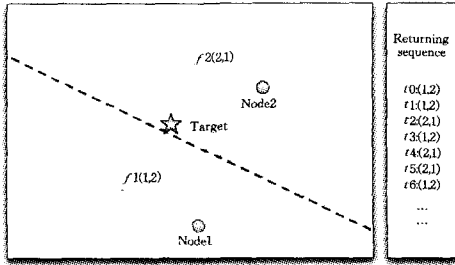


图 1 在不同时间返回节点排序翻转

定义 1(不确定区域, Uncertain Area) 对任意一对传感器节点 $n_1:(x_1, y_1)$ 和 $n_2:(x_2, y_2)$, 存在一个区域, 当目标出现在这个区域时, 无法判别目标更靠近哪一个节点, 这个区域被称为不确定区域。

由定义 1 可知, 对每一个采样时刻, 传感器节点间的接收信号强度关系排序同样不能准确地获得。因此基于相对距离定位的目标跟踪方法^[18-20] 存在较大的误差就是由于其过分依赖根据偏差的探测结果确定的节点感知结果排序。下一小节将基于对数信道衰减模型, 计算不确定区域的几何形式及其边界, 并通过部署的成对传感器节点的不确定区域的相互交叠对跟踪区域进行划分。

3.2 不确定区域边界

根据基于对数信道衰减强度的检测模型, 传感器节点接收到的移动目标的信号强度可以表示为:

$$PL(d_k^i) = PL(d_0) + A - 10\beta \log\left(\frac{d_k^i}{d_0}\right) + X_i \quad (1)$$

式中, $PL(d_k^i)$ 表示第 i 个节点在 k 时刻的感知强度大小, β 为路径损耗指数, d_k^i 表示在 k 时刻移动目标和第 i 个节点的距离, $PL(d_0)$ 表示对于参考距离 d_0 的经验路径损耗值。路径损耗指数 $\beta=2$ 表示自由空间, $\beta=3, 4$ 时代表具有反射和折射的实际环境^[21]。 X_i 表示第 i 个节点在 k 时刻的随机噪声, 它服从均值为 0、方差为 σ_x^2 的正态分布。如果把传感器节点接收信号强度的误差容忍的范围定义为传感器的感应分辨率, 假设传感器的感知分辨率均为 $\epsilon/2$, 可以通过感知分辨率来描述两个节点 m 和 n 在 k 时刻的接收信号强度差的上界, 有:

$$\epsilon = PL(d_k^m) - PL(d_k^n) = 10\beta \log\left(\frac{d_k^m}{d_k^n}\right) + (X_m - X_n) \quad (2)$$

那么移动目标与这两个节点的距离比值为:

$$\frac{d_k^m}{d_k^n} = 10^{\left(\frac{\epsilon - (X_m - X_n)}{10\beta}\right)} = e^{\left(\frac{\ln 10[\epsilon - (X_m - X_n)]}{10\beta}\right)} \quad (3)$$

由于对数信道衰减模型中, $X_m, X_n \sim N(0, \sigma_x^2)$, 因此可以得到:

$$\epsilon - (X_m - X_n) \sim N(\epsilon, (\sqrt{2}\sigma_x)^2) \quad (4)$$

$$\text{设 } y = \frac{\epsilon - (X_m - X_n)}{10\beta}, y \sim N\left(\frac{\ln 10}{10\beta}\epsilon, \left(\frac{\ln 10}{10\beta}\sqrt{2}\sigma_x\right)^2\right)。$$

取式 (4) 的期望可得:

$$\frac{d_k^m}{d_k^n} = E(e^y) = e^{\left(\frac{\ln 10}{10\beta}\epsilon + \frac{1}{2}\left(\frac{\ln 10}{10\beta}\sqrt{2}\sigma_x\right)^2\right)} = C > 1 \quad (5)$$

可以看到, 对于给定的传感器节点感知分辨率 $\epsilon/2$, C 是一个大于 1 的常数, 并且如果两个节点在时间 k 的接收信号强度过于接近(两者的差小于 ϵ), 则无法对这两个节点的 RSS 进行排序比较。当两个节点的 RSS 无法排序且其 RSS 之差等于 ϵ 时(两个节点无法排序时 RSS 可能出现的最大差异), 目标与两个节点的距离的上界则遵从式 (5), 其点的轨迹刻画了成对节点探测不确定区域的边界。

假设两个节点(见图 2)的距离是 $2d$, 它们的坐标分别是 $(d, 0)$ 和 $(-d, 0)$ 。根据等式 (5) 得到点 (x, y) 的轨迹:

$$\left(x - \frac{C^2 + 1}{C^2 - 1}d\right)^2 + y^2 = \frac{4C^2 d^2}{(C^2 - 1)^2} \quad (6)$$

式 (6) 是一个圆的轨迹方程, 因此给出不确定区域边界的定义如下。

定义 2(不确定区域边界, Uncertain Boundary) 对任意两个节点组成的节点对, 定义围成其不确定区域的两个轴对称圆(阿波罗尼奥斯圆, Circles of Apollonius^[25]) 为这个节点对的不确定区域边界, 如图 2 所示, 节点对的垂直平分线为其对称轴。

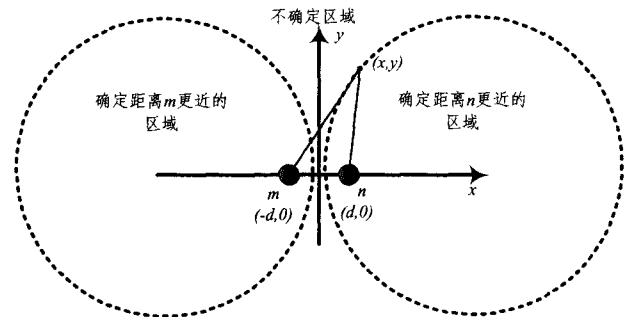


图 2 节点对的不确定区域边界

如果将 4 个传感器节点部署成栅格状, 按照文献[20]的不确定节点序列匹配方法, 整个平面将被 4 个节点的两两的中垂线划分成 8 个面, 如图 3(a) 所示。在理想情况下, 所有返回的接受信号强度能够精确反映目标与传感器的距离, 因此基于平面划分和序列匹配的目标跟踪方法^[18-20] 体现出高效、灵活的优点。然而, 根据不确定区域的定义以及不确定区域边界的分析, 如图 3(b) 所示, 当 d_k^m 从 0 变化到 ∞ 时, 节点对的不确定区域边界为目标与这两个节点距离比值 C 的点轨迹, 形成一对阿波罗尼奥斯圆。节点对两两的不确定区域相交, 将平面划分成若干各面。在这样的划分下, 能够确定反映节点接收信号强度排序的区域的面积(见图 3(a) 中划分的 8 个面), 其在图 3(b) 中已经缩至很小了, 如 $f_1 \rightarrow f_1', f_2 \rightarrow f_2'$, 而大部分的划分面都包含至少一个节点对相对 RSS 排

序的不确定性。这就解释了为什么在受到不确定性影响的应用场景中,基于平面划分和序列匹配的方法的目标跟踪性能会受到巨大影响。此外,当两节点间距离增加到一定值时,将不存在可以获得确定探测节点序列的面,如图 3(c)所示,这种情况下,在真实环境中利用单次目标采样以及以确定节点序列匹配方法进行连续跟踪的误差将会更为巨大。

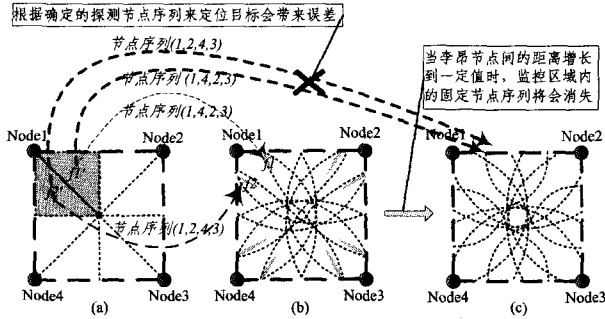


图 3 基于垂直平分线(a)和不确定区域边界线(b)、(c)的区域划分

4 跟踪方法详细设计

本小节将详细讨论基于成对节点 RSSI 探测不确定性的目标跟踪(Tracking with Pairwise Uncertainty of RSSI, TPU-RSSI)算法。

4.1 算法概述

上一节从节点感知模型和区域划分的角度说明了基于平面划分和序列匹配的目标跟踪方法会受到不确定性的巨大影响。从目标感知的角度来看,传感器节点对目标的单次采样具有一定的偶然性,其感知结果的排序并不能准确反映真实的目标与各个传感器节点的相对距离关系。因此,基于成对节点 RSSI 探测不确定性的目标跟踪(TPU-RSSI)算法将从平面划分和目标感知两个方面分析 RSSI 目标探测模型的不确定性,抑制不确定性对跟踪性能的不良影响,同时通过获取和利用不确定性作为附加信息来提高目标跟踪精度,其基本思想如图 4 所示。

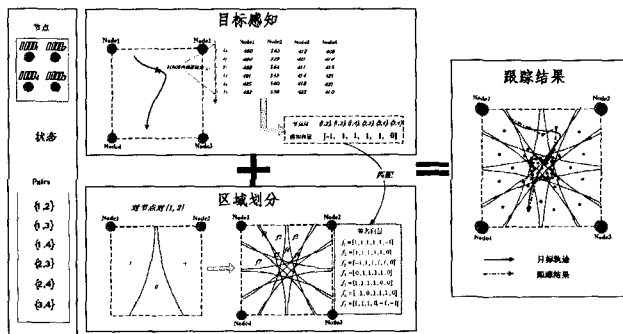


图 4 基于成对节点 RSSI 探测不确定性的目标跟踪算法示意图

一方面,每一次进行目标定位的各个传感器节点都能够近似同步地在一个非常短的时间间隔内,突发、连续地采样若干次,生成该节点的一个分组感知结果,再通过对分组感知结果的分析,构建反映各个传感器对移动目标感知的相对关系的感知向量,作为本次目标定位的依据。具体方法为:对于参与定位的每一个节点对,如果两个节点的感知结果的排序在不同的时刻发生了翻转,将这个节点对赋值为 0;如果在所有时刻,节点对两个节点的感知结果的排序相同,根据节点 id

的升序或降序将节点赋值为 1 和 -1。将所有的节点对的赋值根据节点 id 从小到大枚举排列,即可得到用于本次目标定位的感知向量。

另一方面,参与目标跟踪的传感器节点的两两不确定区域相互交叠,将跟踪区域划分为若干面,每一个划分面的质心作为目标定位的候选位置。每个划分面通过一个反映该划分面与各个传感器节点的距离关系的特征向量唯一标识。具体方法为:每一节点对的不确定区域线将跟踪区域分成 3 部分,将该节点对不确定区域赋值为 0,更接近于较小节点号的传感器的区域赋值为 1,剩余的部分为 -1。任何一个划分面对每一节点对来说,只能属于该节点对 3 个区域之一,因此对每一个划分面,其特征向量可以通过从小到大枚举排列该划分面所属的各个节点对的区域赋值得到,且每一个划分面拥有唯一特定的特征向量。

因此,通过将每一次基于分组感知的目标定位所生成的感知向量与各个划分面的特征向量进行匹配,就能够获得移动目标的实时位置。在整个目标跟踪过程中,感知向量和区域划分不但包含了不确定性的影响,而且巧妙利用了传感器节点和移动目标之间的相对位置关系。通过相关传感器节点周期性地同步分组感知和向量匹配,就能够实现高效灵活的移动目标跟踪算法。

4.2 分组感知与分组感知向量

在传感器节点感知到移动目标时,由于它和各个传感器节点间的地理距离的差异会使各传感器节点具有不同的采样结果,因此当传感器节点对目标进行每一次定位感知时,可以在一个很短的时间区间连续对目标进行多次采样。在传感器的采样频率较高时,认为这个时间区间内目标是相对静止的。因此,这种分组、突发的采样可得到一组分组感知结果。通过对分组感知结果中节点对采样结果在不同时刻的大小进行排序(分析节点对的采样大小是有序还是发生了翻转),就可以有效地捕获目标跟踪过程中感知的不确定性,提高目标跟踪的可靠性。首先给出分组感知的形式化定义如下。

定义 3(分组感知) 假设跟踪区域内部署 n 个传感器节点,在一个很小的时间区间 Δt 内,各个传感器节点几乎同步地对目标突发地进行 m 次采样,即,从 t_0 到 t_{m-1} 各采样一次,这些感知结果被称为一个分组感知。一个分组感知能够通过如下矩阵表示:

$$\begin{bmatrix} r_{SS1,t_0} & r_{SS2,t_0} & r_{SS3,t_0} & \cdots & r_{SSN,t_0} \\ r_{SS1,t_1} & r_{SS2,t_1} & r_{SS3,t_1} & \cdots & r_{SSN,t_1} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ r_{SS1,t_{m-1}} & r_{SS2,t_{m-1}} & r_{SS3,t_{m-1}} & \cdots & r_{SSN,t_{m-1}} \end{bmatrix}$$

每一个采样时刻,所有传感器的感知信号强度构成了分组感知矩阵的每一行,将所有的传感器节点的接收信号强度降序排列,能构成一个感知结果排序。对任意传感器节点 n_i ,不同时刻的采样结果的集合定义为 r_{SSi} ,同时它也是矩阵的每一列。对任意节点对(n_1 和 n_2)及其感知结果(n_1 节点的采样值为 r_{SS1} , n_2 节点的采样值为 r_{SS2}),有 3 种情况:1. 若 r_{SS1} 中的所有元素都大于 r_{SS2} ,则定义为 $r_{SS1} > r_{SS2}$; 2. 若 r_{SS1} 中的所有元素都小于 r_{SS2} ,则定义为 $r_{SS1} < r_{SS2}$; 3. 若 r_{SS1} 中并非所有元素都大于(小于) r_{SS2} ,则定义为 $r_{SS1} \approx r_{SS2}$ 。所以根

据分组感知节点对的采样结果的3种可能性,分组感知中节点对的关系可能为顺序节点对 (n_1, n_2) 、逆序节点对 (n_2, n_1) 及翻转节点对 $\{n_1, n_2\}$ 。

定义4(分组感知节点对赋值) 对任意传感器节点 n_i 和 n_j ,设 $i < j$,通过比较其感知结果(如RSS),可以得到节点对的值,定义为 V_{n_i, n_j} , $\forall i, j, V_{n_i, n_j} \in \{-1, 0, 1\}$;如果 $r_{SS1} > r_{SS2}$,则 $V_{n_i, n_j} = 1$;如果 $r_{SS1} < r_{SS2}$,则 $V_{n_i, n_j} = -1$;若 $r_{SS1} \approx r_{SS2}$,则 $V_{n_i, n_j} = 0$ 。

在分组感知节点对赋值的基础上,通过有序地枚举排列部署的传感器节点所有的节点对的赋值,形成包含不确定因素和目标跟踪信息的感知向量。

定义5(感知向量) 假设在跟踪区域部署了 n 个传感器节点,容易得到总的节点对的数目为 $\mathcal{N} = C_N^2$ 对。按照从小到大的顺序将这些节点对的值进行排列,即 $(n_1, n_2), (n_1, n_3), \dots, (n_1, n_N), (n_2, n_3), (n_2, n_4), \dots, (n_2, n_N), \dots, (n_{N-1}, n_N)$,根据节点对的赋值形成感知向量 $\vec{V}_d = [v_1, v_2, \dots, v_N]$,其中 $V_i \in \{-1, 0, 1\}, i = 1, 2, \dots, \mathcal{N}$ 。

综上所述,可以将移动目标的分组感知向量的构建归纳为4个步骤:

1. 获得分组感知结果:收集所有传感器在感知时间区间 Δt 内不同时刻的接收信号强度信息,构建分组感知矩阵。
2. 降序排列以找出每个感知时刻的感知节点采样结果排序:将分组感知矩阵中的每一行数据进行降序排序,得到不同时刻各个节点的接收信号强度序列。
3. 查找每对节点对是有序还是翻转:两两比较接收信号强度序列中所有感知节点的值,得到有序对集合和翻转对集合这两个集合。
4. 通过组感知结果获取感知向量:按照步骤3的计算结果,给每个节点对赋值,并通过节点对的排列形成感知向量。

分组感知向量构建如算法1所示,该算法给出了如何根据 N 个传感器节点的分组感知结果构建感知向量。

算法1 构建感知向量

```

输入: 传感器在不同时间段的感知结果,感知次数为k
输出: 感知向量  $\vec{V}_d$ 
1: while count  $\neq$  k do
2:   TempArray = [RSS1k, RSS2k, ..., RSSNk];
3:   SamplingArray = Descending(TempArray);
4:   Matrix[count]  $\leftarrow$  SamplingArray;
5:   count = count + 1;
6: end while;
7: count = 0;
8: for i = 0 to N do
9:   for j = i to N do
10:    for w = 0 to k do
11:     if Matrix[i][k]  $\geq$  Matrix[j][k] then
12:      if Vd[count] = -1 then
13:       Vd[count] = 0;
14:      break;
15:     else
16:      Vd[count] = 1;
17:     end if;
18:     else if Matrix[i][k]  $\leq$  Matrix[j][k] then
19:      if Vd[count] = 1 then
20:       Vd[count] = 0;
21:      break;
22:     else
23:      Vd[count] = -1;
24:     end if;
25:     end if;
26:   end for;
27:   count = count + 1;
28: end for;
29: end for;

```

通过一个例子来说明分组感知以及分组感知向量的构建。如图5所示,4个节点在一个很短的时间区间内几乎同步地在不同时刻进行采样,例如,在一个非常短的时间区间 Δt 内,在 t_0 到 t_5 时刻分别进行采样。这4个传感器节点在不同时刻的6次感知结果构成分组感知矩阵。通过对每个时刻的感知结果降序排列(分组感知矩阵的每一行),可以很容易地得知4个节点总共有 C_4^2 个节点对,其中有有序对为 $(2, 1), (1, 3), (1, 4), (2, 3), (2, 4)$,翻转对为 $\{3, 4\}$ 。故得到该分组

感知的感知向量为 $[-1, 1, 1, 1, 1, 0]$ 。

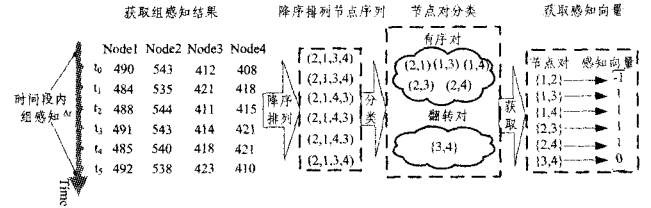


图5 构建感知向量示意图

算法的输出向量 \vec{V}_d 即为该分组感知结果的感知向量。算法1中的第1到第6行表示获取组感知矩阵并对每一行进行降序排列,第8到第29行表示构建感知向量。易知该算法的空间复杂度为:

$$O(N \cdot k + \frac{N(N-1)}{2}) = O(N^2) \quad (5)$$

而该算法的时间复杂度为 $O(N^2 \cdot k)$ 。其中分组感知次数 k 为有限的常数,因此系统的时间复杂度也为 $O(N^2)$ 。

4.3 区域划分与划分面特征向量

为了利用分组感知得到的包含不确定因素的目标信息来进行目标跟踪,需要利用两两节点的不确定区域边界线将跟踪区域划分成若干个面,每一个面不仅需要刻画该面与所有传感器节点的空间位置关系,还要反映成对节点的不确定性,并将其作为确定目标位置待选的候选面,如图6(a)所示。

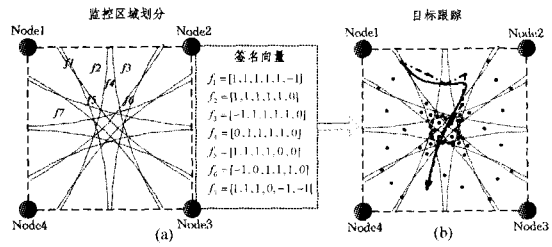


图6 划分区域面和跟踪过程示意图

对于任意给定的节点对,对跟踪区域内的任一位置点可分为以下3种情况:离其中一个传感器节点 n_i 更近,离另一个传感器节点更近或者处于这两个节点对的不确定区域中。按照3.2节中不确定区域的讨论,跟踪区域的每个位置点对每一个特定的节点对都有一个唯一的特征赋值。将该位置点的参与目标跟踪的所有节点对的特征赋值进行有序排列,可以得到该位置点特征向量,用于描述位置点在不确定因素下与各个传感器节点位置的空间相对关系。故给出跟踪区域中任一位置点的特征向量如下。

定义6(特征向量 Signature Vector): N 个部署在跟踪区域的传感器节点,共有 $\mathcal{N} = C_N^2$ 个节点对。顺序排列该位置点对所有的节点对的特征赋值,可获得一个节点序列 $(n_1, n_2), (n_1, n_3), \dots, (n_1, n_N), (n_2, n_3), (n_2, n_4), \dots, (n_2, n_N), \dots, (n_{N-1}, n_N)$ 。跟踪区域内的任意一个给定位置点 p ,对任意两个节点 $(n_i, n_j), i < j, p$ 所分配的特征值的规则为:若 p 离 n_1 更近,则值为1;若 p 离 n_2 更近,则值为-1;若位置点 p 位于节点和 n_j 的不确定区域,则值为0。容易得到每个节点对的值并形成 p 点的特征向量,即 $\vec{V}_s = [v_1, v_2, \dots, v_N]$,其中 $v_i \in \{-1, 0, 1\}, i = 1, 2, \dots, \mathcal{N}$ 。

引理1(划分面特征向量的唯一性) 所有在同一个划分

面的位置点具有相同的特征向量且具有相同的特征向量的位置点也属于同一个划分面,即,若 P 为位置点的集合且 F 为划分面的集合, $\forall_{p_1, p_2} \vec{V}_s(p_1) = \vec{V}_s(p_2) \Leftrightarrow p_1, p_2 \in f$, 其中, $p_1, p_2 \in P, f \in F$ 。

证明:利用反证法进行证明。设存在不同两个位置点 p_1, p_2 , 且 $\vec{V}_s(p_1) = \vec{V}_s(p_2)$, 若 p_1, p_2 属于不同的划分面, 即, $p_1 \in f_1, p_2 \in f_2$, 且 $f_1 \neq f_2$, 易知划分面 f_1, f_2 至少被某一个节点对的不确定区域边界线所隔开(如果不是, f_1, f_2 必为同一划分面), 那么 p_1, p_2 对该节点对的特征值必然不同, 与假设 $\vec{V}_s(p_1) = \vec{V}_s(p_2)$ 矛盾。因此 p_1, p_2 属于同一个划分面。另一方面, 若 p_1, p_2 属于同一个划分面, 即, $p_1, p_2 \in f$, 若 $\vec{V}_s(p_1) \neq \vec{V}_s(p_2)$, 则 $\vec{V}_s(p_1), \vec{V}_s(p_2)$ 两个向量中必然存在至少一个分量节点对 p_1, p_2 的特征值不同, 那么, p_1, p_2 一定被这对节点对的不确定区域边界线分隔, 所以 p_1, p_2 一定属于不同的面, 这与假设 $p_1, p_2 \in f$ 矛盾, 因此 $\vec{V}_s(p_1) = \vec{V}_s(p_2)$ 。综上, 得到结论 $\forall_{p_1, p_2} \vec{V}_s(p_1) = \vec{V}_s(p_2) \Leftrightarrow p_1, p_2 \in f$, 其中, $p_1, p_2 \in P, f \in F$, 证毕。

引理 1 说明了划分面的特征向量的唯一性, 因此对移动目标的每次定位的分组感知将返回一个感知向量, 它能够通过向量匹配的方法将目标唯一确定在最有可能出现的划分面中, 如图 6(b) 所示。

4.4 最大似然匹配

在理想情况下, 每一次目标定位的分组感知向量 \vec{V}_d 总能找到一个唯一的特征向量为 \vec{V}_s 的划分面, 满足 $\vec{V}_s = \vec{V}_d$, 因此目标的实时位置即为 \vec{V}_s 对应的划分面的质心。然而, 在实际目标跟踪场景中, 由于受到不确定性的影响, 传感器节点的感知的可靠性降低, 很可能某一次分组感知的感知向量找不到某一个与之相等的特征向量及其对应的划分面。这是由于跟踪区域将被部署的 N 个传感器节点划分为 $O(N^4)$ 个面, 即目标可能出现候选位置的数量为 $O(N^4)$ 个; 而 N 个传感器返回的分组感知结果可能出现的感知向量的数量则远远大于 $O(N^4)$ 。那么, 当出现找不到匹配的划分面的情况时, 就需要采用极大似然匹配的方法, 通过相似计算来确定移动目标应该出现在哪一个划分面。

为了进行基于极大似然匹配的目标定位, 我们需要首先给出向量相似度的定义, 这里采用向量的欧几里得距离来定义向量间的相似度。

定义 7(向量的相似度) 对任意两个不同的向量 \vec{V}_1 和 \vec{V}_2 ($\vec{V}_1 \neq \vec{V}_2$), 这两个向量的相似度 $S_{\vec{V}_1, \vec{V}_2}$ 定义为:

$$S_{\vec{V}_1, \vec{V}_2} = \frac{1}{\|\vec{V}_1 - \vec{V}_2\|} \quad (7)$$

式中, $\|\vec{V}_1 - \vec{V}_2\|$ 为向量 \vec{V}_1 和 \vec{V}_2 的欧几里得距离。根据感知向量和特征向量的定义易知, 假设两者的相似度是合理的。对于一个感知向量, 若存在某一面的特征向量与之具有最大相似度, 则认为该面所处位置为目标最可能存在的位置。

为了证明相似度假设的合理性, 需要分析向量的相似度和移动目标与传感器节点的地理距离之间的关系。假设有 N 个传感器节点部署在跟踪区域内, 并存在由不确定区域边界线划分的两个面 f_1 和 f_2 , 其特征向量分别为 $\vec{V}_s(f_1)$ 和 $\vec{V}_s(f_2)$, 向量中每个分量都代表目标处在其中某一对特定节点

对的不确定区域的边界线划分区域的位置。对第 i 个分量的节点对, 其划分区域的赋值分别对应划分面 f_1 特征向量的分量 $v_{1,i}$ 以及划分面 f_2 的分量 $v_{2,i}$, 两个分量的差就意味着这两个划分面相对于该节点对感知排序的不同特征。如果把感知向量作为一个未知的划分面, 那么在无法直接找到与感知向量相等的特征向量所在划分面的情况下, 通过寻找具有与感知向量最大相似度的划分面的方法对目标进行定位是合理的。

例如如图 4 所示, 跟踪区域里部署了 n_1, n_2, n_3 和 n_4 4 个节点, 根据其分组感知矩阵可以计算得到的感知向量为 $\vec{V}_d = [-1, 1, 1, 1, 1, 0]$, 同时可以看出面 f_3 的特征向量亦为 $[-1, 1, 1, 1, 1, 0]$, 那么当前移动目标的位置就确定在划分面 f_3 中。但是, 若感知向量为 $\vec{V}_d = [-1, 1, 1, 1, 1, 1]$, 则不存在任何一个划分面的特征向量与之完全匹配, 所以需要用最大似然匹配的方法来进行目标定位。通过计算感知向量与每个面的特征向量间的相似度, 我们可以发现感知向量 \vec{V}_d 与面 f_3 的特征向量间的相似度 $S_{\vec{V}_d, \vec{V}_s(f_3)} = 1$, 是 \vec{V}_d 与所有划分面特征向量相似度的最大值。因此, 目标同样应被定位到 f_3 上。

4.5 启发式匹配算法

TPU-RSSI 跟踪算法的核心是分组感知向量与划分面特征向量的匹配。因为 N 个参与目标跟踪的传感器节点会将跟踪区域划分为 $O(N^4)$ 个面, 所以必须对每一个划分面的 ID 及其特征向量进行存储, 其空间复杂度 $O(N^4)$ 。此外, 通常情况下感知向量与特征向量匹配需要遍历所有的面, 这就意味着遍历匹配过程的时间复杂度为 $O(N^4)$ 。由于移动目标的实时性要求较高、传感器能力较弱, 这样的计算时间复杂度是不能接受的。由于划分面之间并不是毫无联系的, 相反它们的位置和边界存在很强的空间关联特性, 因此可以通过引入邻居面连接关系来降低匹配的难度, 从而改进算法。首先我们给出邻居面连接的定义。

定义 8(邻居面连接) 对任意划分面 f , 如果一个相邻的划分面 f' 与 f 有公共边, 则称 f' 和 f 互为邻居面, 该关系即为 f' 和 f 的邻居面连接(Neighbor face link), 表示为 $l_{f-f'}$ 。如图 6 所示, f_1, f_4 和 f_5 都是 f_2 的邻居面。

定理 1(邻居面及其特征向量关系) 假设 f 和 f' 是邻居面, f 的特征向量是 $\vec{V}_s(f)$, f' 的特征向量是 $\vec{V}_s(f')$, 则 $\vec{V}_s(f)$ 和 $\vec{V}_s(f')$ 满足 $\|\vec{V}_s(f) - \vec{V}_s(f')\| = 1$ 。

证明: 由于 f 和 f' 是邻居面, 它们之间必定存在一条公共边。设 p 与 p' 点的特征向量分别是 $\vec{V}_s(p)$ 和 $\vec{V}_s(p')$, 并且这两点分别属于 f 和 f' 。假设 p 非常接近于 p' 而且只该共同边分割, 由于该边处在一个节点对的不确定区域边界线, 使得 p 与 p' 点的特征向量之间至少存在一组不同的分量, 假设为第 i 组分量, 表示为 v_i 和 v_i' , 容易看出, $|v_i - v_i'| = 1$; 由于 p 非常接近于 p' , 因此并不存在其它不确定区域边界线穿过这两点连线, 这意味着 p 与 p' 点的特征向量除了第 i 组分量外, 其他分量都相等。所以这两点的特征向量满足 $\|\vec{V}_s(f) - \vec{V}_s(f')\| = 1$ 。据引理 1, 定理得证。

跟踪区域划分面的所有邻居面连接构成邻居面连接集合 L , 可定义为: $L = \{(f, f') \mid \|\vec{V}_s(f) - \vec{V}_s(f')\| = 1\}$ 。该集合将存储在簇头或基站中, 作为跟踪的辅助信息, 它旨在描述

划分面之间的内在关系。如图 7 所示,目标跟踪过程在加入邻居面连接信息后,具有两个优点:1)每一次目标定位过程都被分成若干轮的向量匹配,基于邻居面连接集合,每轮匹配转换为寻找与感知向量拥有最高相似度的特征向量的邻居面,以确保匹配具有最高的收敛效率。因此,原本需要进行遍历的定位过程被转换为寻找匹配的收敛路径,大大降低了计算复杂度。2)考虑到目标跟踪是一个连续定位的过程,上一次确定的目标位置的划分面被选为下一次定位匹配的初始划分面。通过综合历史定位信息,每次的定位匹配将在很少的搜索轮中完成,进一步降低了跟踪的复杂度。

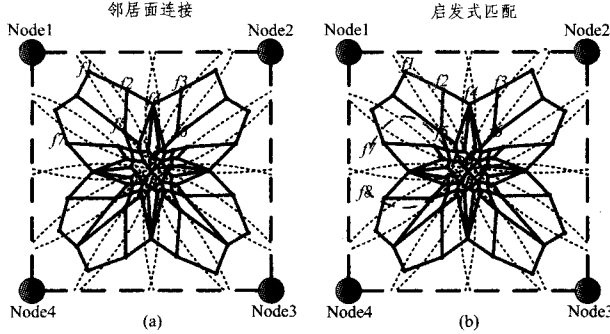


图 7 构建邻居面连接集合的启发式匹配算法示意图

算法 2 给出了通过向量匹配目标跟踪的具体算法。可以看出,在引入邻居面连接集合之前,目标跟踪需要存储各个划分面的 ID 以及特征向量,其空间复杂度为 $O(N^4)$ 。而启发式匹配算法需要对邻居面连接集合进行存储,邻居面之间的连接边的条数等于由不确定区域线相交分割的线段的数目,即 $O(N^4)$,这意味着建立邻居面连接集合最终不会增加算法的空间复杂度。然而,通过构建邻居面连接集合,基于邻居面之间的相互联系有序地组织划分面,并采用基于相邻关系的启发式匹配方法,将目标跟踪方法的时间复杂度从 $O(N^4)$ 降低为 $O(N^2)$ 。

算法 2 通过向量匹配法跟踪

```

输入: 面ID及签名向量 $V_i$ , 节点对  $(f_1, V_1(f_1)), (f_2, V_2(f_2)), \dots, (f_n, V_n(f_n))$ 
      邻居面连接集合  $L$ 
      感知向量  $V_{d1}, V_{d2}, \dots, V_{dm}$ 
输出: 目标跟踪轨迹  $P$ 

1: for  $i=0$  to  $m$  do
2:   if  $f_0 == null$  then
3:      $f_0 = \text{Initialization}()$ ;
4:   end if;
5:    $f_n = f_0$ ;
6:   while  $S_{neighbor} > S_{current}$  do
7:      $f_c = f_n$ ;
8:      $S_{current} = S_{V_c, V_c(f_c)}$ ;
9:      $S_{neighbor} = 0$ ;
10:    while  $f_c.next\_neighbor \neq null$  do
11:      if  $S_{neighbor} < S_{V_c, V_c(f_c.next\_neighbor)}$  then
12:         $S_{neighbor} = S_{V_c, V_c(f_c.next\_neighbor)}$ ;
13:         $f_n = f_c.next\_neighbor$ ;
14:      end if;
15:    end while;
16:  end while;
17:   $f_0 = P[i] = f_c$ ;
18:   $S_{current} = -1$ ;
19: end for;
```

综合算法 1 可知,跟踪算法最终的空间复杂度为 $O(N^4)$ 而时间复杂度是 $O(N^2)$ 。因此采用基于相邻关系的启发式匹配方法将降低算法的计算复杂度,增加定位跟踪的效能。

5 仿真实验

为了验证 TPU-RSSI 的性能,使用仿真平台 OPNET 14.5 构建用于目标跟踪的传感器网络,网络中存在两种节点模型,一是参与目标跟踪的传感器节点,二是移动目标。

在仿真实验中,我们按照文献[26]提出的方法将跟踪区域分割成栅格以实现近似区域划分。移动目标的运动轨迹服从文献[23]提出的随机路点运动模型。传感器节点分别按照

规则部署和不规则均匀部署进行实验,其数目在 5~40 之间变化,每个传感器节点基于 RSSI 感知模型,设定其感知半径为 4m(当目标与传感器的距离超过其感知半径时,其感知结果不参与目标定位计算),采样频率为 10Hz。由于目标跟踪主要考察目标跟踪的误差,因此,定义在目标跟踪中某一点的误差为估计位置与实际位置的地理距离。同时,为了刻画跟踪算法得到的目标运动轨迹的平滑性,定义目标跟踪误差的偏差为目标跟踪轨迹上每次定位的误差的平均值。为了描述环境不确定性,环境噪声模型参数中的路径损耗指数 $\beta=4$,噪声分布的标准差 $\sigma_x=6$ 。在实验场景中设置每一次仿真持续 60s。仿真实验参数的设置如表 1 所列。

表 1 仿真参数设置

参数	取值
部署区域面积	$10 \times 10 \text{m}^2$
噪声模型参数	$\beta=4, \sigma_x=6$
传感器节点数量(N)	5~40
感知半径(R)	4m
感知分辨率(ϵ)	0.5~3dBm
感知频率(λ)	10Hz
目标移动速度	1~5m/s
感知次数	3~9times

5.1 目标跟踪仿真概况

本小节通过仿真实验对基于成对节点探测不确定性的目标容错跟踪方法(TPU-RSSI)和基于路径的最大似然匹配目标跟踪算法(Path-based MLE, PM)的跟踪性能进行了比较。仿真中,目标将从左向右按“┌”移动。跟踪区域部署了 9 个传感器节点,设置感知分辨率 $\epsilon=1$,且分组感知的采样次数 $k=5$,分别在传感器节点的规则部署(9 个参与跟踪的节点部署为交叉十字即“+”形)和非规则均匀部署的情况下对两种方法的目标跟踪的轨迹进行标绘,如图 8 所示。根据仿真结果,可以得到如下结论:

1)图 8(a)和 8(b)给出了当传感器节点规则部署在 $10 \times 10 \text{m}^2$ 的跟踪区域中时,TPU-RSSI 算法和 PM 算法所获得的跟踪结果。图 8(c)和 8(d)为传感器节点按照二维正态分布随机部署下两种方法的跟踪结果。可以看出,与基于平面划分和序列匹配的跟踪方法中性能最好的 PM 算法相比,TPU-RSSI 方法不论在跟踪精度还是轨迹平滑度上都更好。

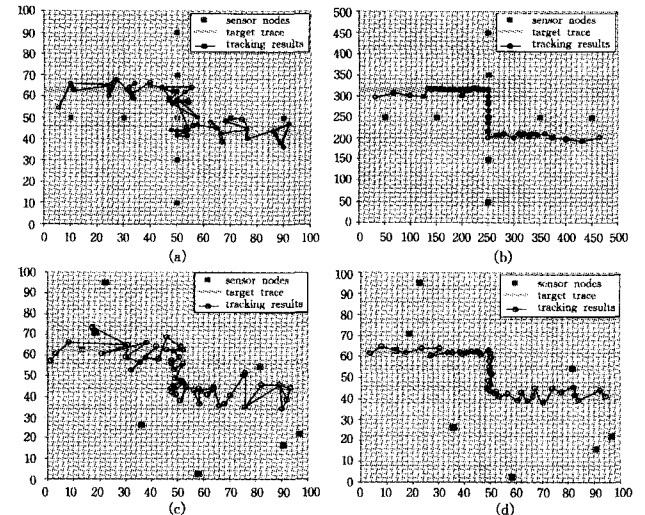


图 8 TPU-RSSI 和 PM 在规则以及随机部署的节点下的目标跟踪示意图($k=5, \epsilon=1$)

2)图8(a)和图8(c)是PM算法在跟踪节点规则和非规则的情况下的跟踪结果,可以看出,两者的跟踪精度和轨迹平滑度不高并且相近,这说明PM算法无论在节点部署规则和非规则的情况下,都受到不确定性的较大影响。图8(b)和图8(d)是TPU-RSSI算法在这两种相同条件下的跟踪结果,总的跟踪效果较好;节点部署得越规则,跟踪的平均误差和误差偏差就越小,这是因为部署越规则,各划分面大小和位置越均匀,导致目标候选位置分布得越均匀。

5.2 仿真结果性能分析

图9给出了在分组感知采样次数 $k=5$ 、感知分辨率 $\epsilon=1$ 、节点数目 $N=10$ 时的仿真结果,对TPU-RSSI算法与PM方法以及Direct MLE方法跟踪不同时刻动态误差的情况进行了比较,从直观上更验证了TPU-RSSI目标跟踪算法的误差更小且更加稳定。

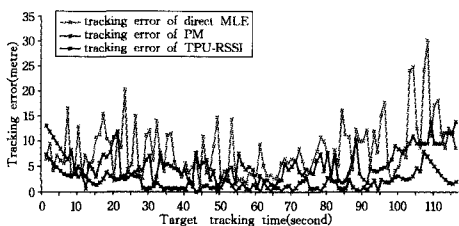


图9 动态误差比较

图10比较了TPU-RSSI目标跟踪算法与PM方法以及Direct MLE方法在不同数目的传感器节点随机分布下的跟踪性能,其中分组感知采样次数 $k=5$ 、感知分辨率 $\epsilon=1$,使参与跟踪的节点数目从10到40变化。图10(a)和图10(b)分别给出了随着节点数目的增大,TPU-RSSI、PM以及Direct MLE 3种目标跟踪算法的平均误差和误差偏差。实验证明:

1)对3种跟踪方法,随着参与目标跟踪的传感器节点数目的增加,跟踪平均误差及其偏差将会减小,尤其在节点数小于25时,误差减小十分显著。

2)TPU-RSSI方法相对于PM方法以及Direct MLE方法不论是在平均误差还是误差偏差上都具有更好的性能。

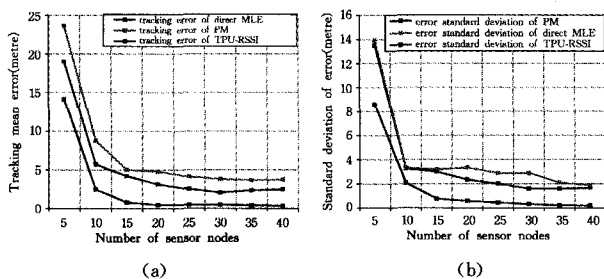


图10 目标跟踪的平均误差和误差的偏差随传感器节点数目的变化($k=5, \epsilon=1$)

如图11所示,当有10、15、20和25个感知节点参与进行目标跟踪且分组感知的采样次数 $k=5$ 时,TPU-RSSI方法的跟踪平均误差随感知分辨率 ϵ 而变化。实验证明:

1)在同样的噪音模型参数下(β 和 σ_x),更高的感知分辨率 ϵ 值可提供更高的跟踪精度。这个结论是违反直觉的,因为 ϵ 值越大表示成对传感器节点的采样结果大小越难以分辨和排序(因为 $\epsilon=0$ 的时,是没有不确定性的);然而,正是由于基于成对节点RSSI探测不确定性的目标跟踪算法将捕获的成对节点感知的不确定性作为目标跟踪的附加信息,使得感

知分辨率 ϵ 值的增加反而提高了定位精度。

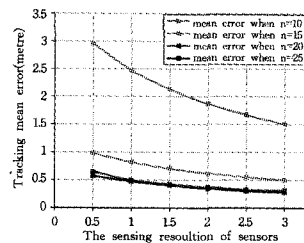


图11 跟踪平均误差随传感器节点感知分辨率的变化

2)对于固定的分组感知采样次数(如 $k=5$),参与跟踪的传感器数目越少,伴随着感知分辨率的下降,跟踪误差的增加越快,因此当传感器节点数大于20时,感知分辨率对跟踪平均误差的影响就非常小了。这说明,对任意的感知分辨率 $\hat{\epsilon}$,都存在特定的采样次数 \hat{k} 和参与跟踪的传感器数目 \hat{N} ,当满足这样的采样和部署时,目标跟踪的精度对感知分辨率是不敏感的。

图12给出了在确定感知分辨率后,不同的分组感知次数 $k=3,5,7,9$ 且传感器节点数从10到40变化时TPU-RSSI方法的平均误差。实验证明:

(1)增加分组感知的采样次数 k 能减少跟踪误差。这是显而易见的,因为采样次数 k 的增加能够提高分组感知中获得成对节点不确定性的概率。

(2)在采样次数非常小而传感器节点数很大时,跟踪误差反而会随着参与跟踪传感器节点数目的增加而增加。这是因为当有许多传感器参与跟踪时,成对节点的不确定性难以通过有限的采样次数获得。

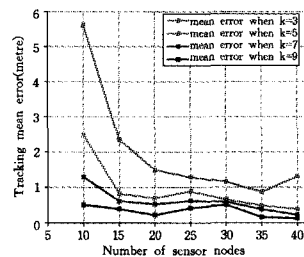


图12 跟踪平均误差随分组感知采样次数的变化

结束语 本文针对成对节点的不确定区域及其边界线做区域划分,通过节点探测的多次采样,有效地降低了传感器节点不可靠感知对目标跟踪性能的影响,将目标跟踪过程转化为一个向量匹配的过程,以此提高目标跟踪的灵活性、减小环境因素的影响并提高跟踪精度。大量的实验证明,在不确定性影响下的目标跟踪环境中,TPU-RSSI方法与同类其他方法相比,体现出更好的跟踪性能。

参考文献

- [1] Akyildiz I F, Su W, Sankarasubramaniam Y, et al. Wireless sensor networks: A survey[J]. Computer Networks, 2002, 38(4): 393-422
- [2] Li D, Wong K, Hu Y H, et al. Detection, Classification and Tracking of Targets in Distributed Sensor Networks[J]. IEEE Signal Processing Magazine, 2002, 19(2)
- [3] Smith A, Balakrishnan H, Goraczkoet M, et al. Tracking Moving Devices with The Cricket Location System[C]//MobiSys'04.

- [4] Gustaffsson F, Gunnarsson F. Mobile Positioning Using Wireless Networks[J]. IEEE Signal Processing Magazine, 2005, 22(4)
- [5] Kusy B, Ledecz A, Koutsoukos X. Tracking Mobile Nodes Using RF Doppler Shifts[C]//Sensys'07. 2007;29-42
- [6] Mohanty S. VEPSD; A Novel Velocity Estimation Algorithm for Next-Generation Wireless Systems[J]. IEEE Trans. on Wireless Com. ,2005,4(6)
- [7] Terzis A, Anandarajah A, More K, et al. Slip Surface Localization in Wireless Sensor Networks for Landslide Prediction[C]//IPSN'06. 2006;109-116
- [8] Gaddi B, Bracha H, Tal A, et al. Continuous Close-Proximity RSSI-Based Tracking in Wireless Sensor Networks[C]//Proc. of 2010 Int'l Conf. on Body Sensor Networks. 2010;234-239
- [9] Moore D, Leonard J, Rus D, et al. Robust distributed network localization with noisy range measurements[C]//Proc. SenSys'04. 2004
- [10] Whitehouse K, Karlof C, Culler D. A Practical Evaluation of Radio Signal Strength for Ranging-based Localization[J]. SigMobile'07, 2007, 11(1)
- [11] Wang Z J, Bulut E, Szymanski B K. Distributed Energy-Efficient Target Tracking with Binary Sensor Networks[J]. TOSN'10, 2010, 6(4)
- [12] Li X R, Jilkov V P. A Survey of Maneuvering Target Tracking; Approximation Techniques for Nonlinear Filtering[C]//SPIE'04. 2004
- [13] Zhang P, Martonosi M. LOCALE; "Collaborative Localization Estimation for Sparse Mobile Sensor Networks[C]//IPSN'08. 2008
- [14] Kalman R E. A New Approach to Linear Filtering and Prediction Problems[J]. Trans. ASME, Journal of Basic Engineering, 1960, 82(1); 35-45
- [15] Ristic B, Arulampalam S, Gordon N. Beyond the Kalman Filter; Particle Filters for Tracking Applications[M]. Artech House, 2004
- [16] Liu J, Reich J, Zhao F. Collaborative In-Network Processing for Target Tracking[J]. EURASIP Journal on Applied Signal Processing, 2003; 378-391
- [17] Ting J, Snoussi H, Richard C. Decentralized Variational Filtering for Target Tracking in Binary Sensor Networks [J]. IEEE Trans. on Mobile Computing, 2010, 9(10); 1465-1477
- [18] Yedavalli K, Krishnamachari B. Sequence-Based Localization in Wireless Sensor Networks[J]. IEEE Trans. on Mobile Computing, 2008, 7(1); 81-94
- [19] Zhong Z, He T. MSP; Multi-Sequence Positioning of Wireless Sensor Nodes[C]//Sensys'07. 2007
- [20] Zhong Z, Zhu T, Wang D, et al. Tracking with Unreliable Node Sequences[C]//IEEE InfoCom'09. April 2009; 1215-1223
- [21] Zanca G, Zorzi F, Zanella A, et al. Experimental Comparison of RSSI-based Localization Algorithms for Indoor Wireless Sensor Networks[C]//RealWSN'08. 2008; 1-5
- [22] Wang W, Srinivasan V, Wang B, et al. Coverage for Target Localization in Wireless Sensor Networks [J]. IEEE Trans. on Wireless Comm. , 2008, 7(2); 667-676
- [23] Xi W, He Y, Liu Y H, et al. Locating Sensors in the Wild; Pursuit of Ranging Quality[C]//Sensys'10. 2010; 295-308
- [24] Ren Q Q, Li J Z, Cheng S Y. Target Tracking under Uncertainty in Wireless Sensor Networks[C]// Mobile Ad hoc and Sensor Systems (MASS), 2011 IEEE 8th International Conference on. 2011; 430-439
- [25] Ogilvy C S. Excursions in Geometry[M]. Dover, 1990

(上接第 181 页)

结束语 本文基于多核处理器提出了多线程 XML 节点查询执行框架,该框架基于 VTD-XML 开展研究工作。该框架通过并行执行提高节点查询执行的性能,同时通过预读策略将查询执行线程将要访问的数据提前取至处理器 Cache 中,提高了查询执行线程的性能。在实验中,基于 VTD-XML 开源代码实现了多线程 XML 节点查询执行框架和本文提出的各种性能优化措施。实验结果表明,本框架能够充分利用多核处理器的计算资源,大大提高了 XML 节点查询执行的性能。

参 考 文 献

- [1] Zhao L, Bhuyan L. Performance Evaluation and Acceleration for XML Data Parsing[C]//Proc. 9th Workshop Computer Architecture Evaluation Using Commercial Workloads (CAECW 06). 2006
- [2] VTD-XML; The Future of XML Processing[OL]. <http://vtd-xml.sourceforge.net/>
- [3] Buneman P, Cong G, Fan W, et al. Using Partial Evaluation in Distributed Query Evaluation[C]//VLDB (2006). 2006; 211-222
- [4] Cong G, Fan W, Kementsisetsidis A. Distributed Query Evaluation with Performance Guarantees[C]//SIGMOD Conference 2007. 2007; 509-520
- [5] Suci D. Distributed Query Evaluation on Semistructured Data [J]. ACM Trans. Database Syst. , 2002, 27(1); 1-62
- [6] Lu W, Gannon D. Parallel XML Processing by Work Stealing [C]//SOCP'07; Proceedings of the 2007 workshop on Service-oriented computing performance; aspects, issues, and approaches 2007. 2007; 31-38
- [7] Bordawekar R, Lim L, Shmueli O. Parallelization of XPath Queries using Multi-core Processor [C]// Challenges and Experiences. EDBT 2009. 2009
- [8] Zhou Jing-ren, Cieslewicz J, Ross K A, et al. Improving Database Performance on Simultaneous Multithreading Processors[C]// Proceeding of the 31nd International Conference on Very Large Databases. VLDB Endowment, 2005; 49-60
- [9] Hardavellas N, Pandis I, Johnson R. Database servers on chip multiprocessors limitations and opportunities[C]//3rd Biennial Conference on Innovative Data Systems Research (CIDR). 2007
- [10] He Bing-sheng, Luo Qiong. Cache-oblivious Database; Limitations and Opportunities [J]. ACM Transactions on Database Systems, 2008, 33(2); 1-42
- [11] Hennessy J L, Patterson D A. Computer Architecture (4th edition)[M]. Morgan Kaufman, 2007