

基于移动阅读器的 RFID 概率空间范围查询技术的研究

谷 峪^{1),2)} 郭 娜²⁾ 于 戈^{1),2)}

¹⁾(医学影像计算教育部重点实验室(东北大学) 沈阳 110004)

²⁾(东北大学信息科学与工程学院 沈阳 110004)

摘 要 在智能交通运输系统和人员物品跟踪等基于位置服务的领域中,对于移动对象位置上的索引建立和查询处理已经成为比较热门的研究内容.文中主要研究利用一种新颖的 RFID 系统的框架结构,依靠位置相对固定的标签来定位携带移动式的阅读器监控对象,从而支持高效的移动范围查询.该结构能够缩小监控对象可能的位置区域,但还是存在位置的不确定性,所以文中提出了此场景下的移动对象位置查询的一种概率模型,给出了有效的定位方法,并在此基础之上讨论了基于虚拟网格划分的 m^2 ROvgP-MRQ 优化算法和 CU-grid 索引结构.实验结果验证了此模型和方法的性能及有效性.

关键词 RFID;移动式阅读器;固定式标签;概率;移动范围查询

中图法分类号 TP311 DOI号: 10.3724/SP.J.1016.2009.02052

Study on Processing Probabilistic RFID Spatial Range Query Based on Mobile Readers

GU Yu^{1),2)} GUO Na²⁾ YU Ge^{1),2)}

¹⁾(Key Laboratory of Medical Image Computing (Northeastern University), Ministry of Education, Shenyang 110004)

²⁾(School of Information Science and Engineering, Northeastern University, Shenyang 110004)

Abstract For the applications of location-based services such as intelligent transport systems and object tracking systems, constructing indexes and processing queries about the positions of moving objects are widely studied. This paper proposes to utilize a novel RFID system framework, using relatively fixed tags to locate moving objects with portable readers, to support efficient probabilistic spatial moving range query. Although the probable position region has been greatly narrowed, the location uncertainty of the monitored objects still challenges the efficiency of moving range queries. This paper proposes a probability evaluation model in the RFID-enabled monitoring environments. Efficient locating methods are analyzed. Furthermore, based on virtual grid, an optimization algorithm called m^2 ROvgP-MRQ and an index structure named CU-grid are discussed to improve the performance of the queries. The experimental results verify the effectiveness and efficiency of the proposed model and methods.

Keywords RFID; mobile reader; fixed tag; probabilistic; moving range query

1 引 言

RFID(Radio Frequency Identification)作为一

个新兴的数据收集和传输技术,采用存储在电子标签中的唯一的 ID 标识物体,自动高速地收集识别范围内的标签信息数据.因此,对任何移动对象都可以进行实时的定位、跟踪和监测,进而支持空间范围等

基于移动对象位置的查询. 例如, 在大型展会中, 游客可选择跟随导游, 这样游览虽然方便却不能根据自己的计划和兴趣灵活的分配时间、选择路线. 那么我们可以利用 RFID 技术, 代替导游使游客在自由观赏的同时, 也可以实时地查询到所需信息, 如附近一定范围内的展项、其他的伙伴有哪些等等.

与传统的定位技术如 GPS 相比较, RFID 系统产品在价格方面更占优势, 因而使应用范围可以更加广泛. 而且它可以由用户自己决定对哪些区域进行监控, 例如在关键出入口处安放阅读器实时地检测带有标签的对象, 其原理类似于在关键位置安排人员对过往物品进行登记, 可以根据登记的信息对物品进行定位. 许多不是基于位置的服务领域业, 也已经出现了 RFID 定位系统的身影, 例如英国一些医院利用它对医疗器械进行定位和跟踪, 减少了设备失窃、误用等事件的发生.

2 相关工作

RFID 技术在迅速发展的同时也为数据库领域提出了许多新问题. 目前主要的研究工作包括数据清洗^[2-3]和基于 RFID 事件流的复杂事件处理^[4]等. 在 RFID 不确定数据管理方面, Cascadia^[5]和 Laha^[6]讨论了如何从原始读数中的基本概率信息挖掘出具有丰富语义的复杂事件. 到目前为止, 已有的研究中只是很少一部分如文献^[7-9]涉及到了带概率的 RFID 定位技术. 然而这些 RFID 系统大都采用固定式的阅读器(布置在特殊位置), 将标签贴在待监控的移动对象上. 因为阅读器的价格要远远高于标签(特别是无源标签和只读标签), 所以出于经济效益方面考虑, 阅读器的使用数量要尽可能地少. 那么在一般情况下阅读器之间的距离较远, 阅读器的探测区域不相交. 这种情况下, 只能确定贴有标签的对象在哪个阅读器的探测区域内(一个圆形或者扇形区域), 但具体位置只能根据速度等其它信息进行估计, 其准确程度与阅读器布置的密度有关. 所以对于监控范围特别大的应用, 很难兼顾准确性和经济效益两方面的要求.

以“园艺博览会”这种大型展会场景为例, 我们采用一种不同结构形式的 RFID 定位系统——阅读器的位置动态改变, 而将标签贴在位置固定的物体上, 根据标签的位置(已知)和读数对阅读器进行实时定位, 从而实现移动范围查询. 之前也有少数研究用到了这种组织结构, 但他们的应用原理、目标不

同, 对于我们的问题和场景并不适用. 比如, Tran 等人^[10]首先根据经验得出带有未知参数的模型, 然后通过对训练样本的学习, 确定模型中的参数值; 在定位过程中, 根据阅读器在不同位置的读数, 利用粒子滤波算法确定物体(标签)的位置, 并动态地修正参数的值. 其原理与标签移动、阅读器固定的 RFID 系统相似, 仍然是已知阅读器的位置, 来定位标签, 而且只考虑了单一阅读器的情况. 文献^[11-12]也用到了这种组织结构, 但采用的是可读写式的无源电子标签, 其存储容器用来记录周围电子标签的信息和经过的阅读器携带着的(远处标签的)信息. 用“信息梯度”作为指南针, 寻找目标对象. 其原理是根据标签的“记忆”对极少数位置发生变化的对象(标签)进行搜索. 存在搜索失败的情况, 即使成功也是非常耗时的, 无法实现实时的空间范围查询.

这种 RFID 系统在进行跟踪定位的过程中会产生大量的读数, 形成携有时空信息的大量数据, 高效索引结构支持的移动范围查询需要进一步分析和研究. 与精确定位技术相比, 位置的不确定性决定了这种 RFID 系统面临着两个挑战:

(1) 定位不是精确的点. 若某一时刻, 阅读器探测到了某些移动对象, 阅读器的位置是根据周围标签的位置进行估计的. 虽然与一个圆形区域相比, 我们将移动对象定位得更加精确, 但仍然不是一个精确的点.

(2) 读数是 不完全连续的. 当阅读器路过标签稀少的地区(如路径上), 某些时刻周围没有标签, 没有读数产生, 造成了对该阅读器读数的不连续性.

在传统的移动对象管理方面, 对于移动对象上的静态范围查询, 大多采用 R-tree 作为基本的索引结构提高查询效率. 其中概率范围查询需要比较特殊的索引结构, Cheng 等人^[13]在 R-tree 的结构中融入了数据分布信息, 构造出的 PTI 索引结构可以有效地提高概率限制类查询的执行效率. 而 U-tree^[14]的设计思想来源于 CFB(Conservative Functional Box), 适用于多维空间内满足任意概率分布的数据, 其原理是将数据库中不满足条件数据先过滤掉一部分, 这样可以减少大量的处理时间. 此外, 由 Kalashnikov 等人^[15]提出的 U-grid 与以上那些索引的结构完全不同, 它用来高效地处理基于不确定性用直方图和四叉树的概率范围查询. 以上这些索引并不适用于动态的范围查询. Gedik 等人^[16]提出了根据移动对象当前的位置和速度信息、运动敏感边界盒及查询点位置处理移动范围查询的技术. 其

索引结构是建立在移动对象的运动敏感边界盒上的,而不是精确的位置坐标.然而对于 RFID 应用,由于位置不确定性的特征,RFID 监控对象只能推断出在哪些分散的区域中或者某条路径上的离散点处.所以现有的索引结构对于我们考虑的场景并不会会有显著的性能提升.

本文利用 RFID 技术,采用固定式阅读器和半固定式标签的新型结构,提出了移动对象位置查询的一种概率模型,并在该模型上讨论了基于虚拟网格划分的 m^2 ROvgP-MRQ 优化算法和 CU-grid 索引结构,有效地提高了查询效率.

3 RFID 空间数据建模

3.1 系统结构分析

图 1(a)是某园艺博览会游览示意图,白色的曲线表示游客在园内参观时可选择的路线,带数字的圆形图标指示园内各个展项的中心位置,方形图标用来指示固定的(如凉亭、洗手间)和移动式的服务设施(常用商品提供商等)以及展会活动(如节日活动、文艺演出、展示交易等)的位置.

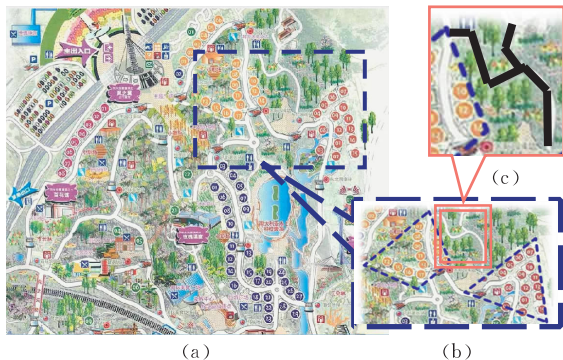


图 1 某园艺博览会游览示意图

虽然展会每日接待的游客可能很多,但 RFID 阅读器并不需要人手一部.如果全家人一起外出或者一个部门的员工组织集体出游,这样路线和行程安排都一致的一个组人只需要一部为其提供服务即可.而且该服务是要收取一定费用的,所以提供给有需要的人即可.阅读器的使用寿命长,循环使用也减少了所需阅读器的数量.园中各种展项、服务设施数目众多,因此相比之下,所需阅读器数量要远远少于标签的数量.这种场景下的特征和各方面需求决定了这种 RFID 查询定位系统适合采用 MR-SM/FT (Mobile Reader Seldom Mobile or Fixed Tag) 结构形式而不是以往的 FR-MT (Fixed Reader Mobile

Tag) 结构. MR-SM/FT 结构的 RFID 定位系统由 4 个部分组成:

(1) 电子标签. 用 o_i 表示. 选用有源只读式电子标签. 因为这种标签的发射功率低、通信距离远、速度快、稳定性较高. 对于园中每个向游客提供服务的实体(包括展项、自然景观、服务设施等各种属性类别),在它们的中心附近都布置一个 RFID 标签,为了下文表述方便,这种实体简单表示为 TItem (Tagged Item). 标签信息的存储格式为 (oid, type, category, position, character). 各属性字段意义是: 标签的唯一标识、分类(固定式和半固定式)、位置(坐标或者函数)、特征信息. 根据标签所属 TItem 的移动性质,将标签分为两类:

① 固定式标签. 位置不发生变化,用一个点坐标 (x, y) 表示.

② 半固定式标签(偶尔移动). 初始位置已知,用一个点坐标 (x, y) 表示;移动之后,位置改变,估计后得出的位置是一个可能区域,区域内部的各点的概率分布情况用概率分布密度函数 $\rho(x, y)$ 表示.

(2) 阅读器. 用 r_i 表示. 选用远距离读卡器. 游客手持,用 rid 作为唯一标识,通过通信网络 (GSM/GPRS/CDMA) 与数据存储中心相连接交换信息: 将每个周期的读数实时的传输给控制中心; 从信息中心获取的 TItem 信息; 接收来自控制中心的查询结果等.

(3) 数据存储中心. 存储每个标签的位置信息及所属 TItem 的其它关键属性信息和概要特征信息(如属性类别是餐饮类、餐厅是法式的不是中式的、特色菜品等等); 根据控制中心的命令,更新移动标签的位置信息; 还要存储静态的路径位置信息.

(4) 控制中心. 建立和维护阅读器、标签和路径的位置数据索引; 根据每个阅读器在移动过程中上报给控制中心的读数,实时地进行推断和重定位,并更新存储记录; 执行用户触发的查询,将结果反馈给用户.

3.2 RFID 定位建模

图 1(b)是对 (a) 部分图的放大,从图中我们可以清晰看出整个展园中 TItem 的分布特点: 包含许多 TItem 比较集中的展区,这些展区之间彼此通过较长的路径连接在一起,路径上的 TItem 分布比较稀疏. 所以在定位时,需要对以下两种情况分析.

(1) 在标签附近的情况

目前,阅读器的种类繁多,不同类型阅读器的探测范围不同(扇形、圆形等),这里我们默认为圆形.

设 R_{read} 为阅读器的最远探测距离,那么探测范围就是以阅读器为中心, R_{read} 为半径的圆形区域. 设 R_{set} 是阅读器 ID 的集合, O_{set} 是标签 oid 的集合, 那么一条 RFID 读数建模为一个三元组 $r_j(t) = \langle j \in R_{\text{set}}, os \subseteq O_{\text{set}}, t \rangle$, 表示在时刻 t 阅读器 r_j 同时探测到了多个标签组成的集合 os . 下面先给出一个基本定义.

定义 1(被探测区域). 一个标签 o_i 出现在某个阅读器 r_j 的读数中, 即被 r_j 探测, 那么这个 r_j 在 o_i 周围可能的位置集, 就叫做 o_i 对于 r_j 的被探测区域 BDR(Being Detected Region).

下面介绍出如何确定某个标签的被探测区域 BDR 的定理, 并给出这个定理的相关证明过程.

定理 1. 若 r_j 的探测区域是一个半径为 r_{read} 的圆, 那么 o_i 对于 r_j 的 BDR 是以 o_i 的坐标为圆心, r_{read} 为半径的圆形区域, 用符号表示为 C_i^j .

证明. 设 o_i 对于 r_j 的 BDR 是 C , $A = C_i^j - C$, $B = C - C_i^j$. 用反证法. 假设 $C \neq C_i^j$, 即 $A \neq \emptyset$ 或 $B \neq \emptyset$.

① 假设 $A \neq \emptyset$, 任取一点 $P \in A$;

那么 $P \in C_i^j \wedge P \notin C$. $P \notin C \Leftrightarrow r_j$ 在 P 处不能探测到 o_i , 可以推断出 r_j 与 P 之间的距离 $dis(r_j, P) = \sqrt{(r_j.x - P.x)^2 + (r_j.y - P.y)^2} > R_{\text{read}}$. 而 $P \in C_i^j \Leftrightarrow \sqrt{(r_j.x - P.x)^2 + (r_j.y - P.y)^2} \leq R_{\text{read}}$, 矛盾! 则 $A \neq \emptyset$ 的假设不成立, 即 $A = \emptyset$.

② 假设 $B \neq \emptyset$, 任取一点 $Q \in B$;

那么 $Q \in C \wedge Q \notin C_i^j$. $Q \in C \Leftrightarrow r_j$ 在 Q 处能够探测到 o_i , 可以推断出 r_j 与 Q 之间的距离 $dis(r_j, Q) = \sqrt{(r_j.x - Q.x)^2 + (r_j.y - Q.y)^2} \leq R_{\text{read}}$. 而 $Q \notin C_i^j \Leftrightarrow \sqrt{(r_j.x - Q.x)^2 + (r_j.y - Q.y)^2} > R_{\text{read}}$, 矛盾! 则 $B \neq \emptyset$ 的假设不成立, 即 $B = \emptyset$.

综上所述 $A = \emptyset$ 且 $B = \emptyset$, 即 $C_i^j - C = \emptyset \wedge C - C_i^j = \emptyset$, 则 $C = C_i^j$. 证毕.

由于标签的价格较阅读器便宜得多, 所以在我们的模型中, 标签布置得比较密集, 从而在某一时刻一个阅读器可以同时探测到多个标签, 那么阅读器应该同时存在于这些标签的 BDR 中, 即通过阅读器的位置可以进一步确定为这些标签的 BDR 重叠区域.

(2) 在路径上距离标签很远的情况

为了提高没有读数时定位的准确度, 将路径抽象为二维空间内的线段, 具体过程如下: 首先根据零星分布的标签的位置将较长的路径划分成多个没有

读数的子路径; 然后将弯度较大的子路径再次分割成可近似看作直线的原子路径, 在没有标签的端点处布置固定式的标签. 随着 RFID 硬件技术的迅速发展, 防碰撞算法不断优化, 阅读器的漏读和错读现象越来越少, 我们这里假设阅读器的读数都是准确的. 下面给出空路径的概念.

定义 2(空路径). 由上述过程得到, 阅读器通过两端时有读数产生, 但在路径途中不会产生读数的原子路径. 忽略这些原子路径的宽度, 得到的二维空间中的线段即为空路径.

可根据保存的最近一次的读数, 判断出游客选择了哪条路径, 根据速度和相对时间可以估计出可能的位置.

4 RFID 概率范围查询基本建模

在本节中我们将讨论基于概率的算法模型. 由于 RFID 技术固有的定位不精确、读数不连续的限制性, 许多位置都是根据有限的信息分析估计得出的, 因此无法给用户一个绝对准确的答案, 那么我们从概率的角度为每种可能性附加一个定量的描述——存在概率, 这个概率值参与中间计算, 为系统提供最终结果的可能性参考依据.

在我们考虑的场景中, 标签布置在每个 TItem 中心附近, 游客在 TItem 范围内可随机移动到各处, 那么阅读器在每个标签的 BDR 中的位置可能性分布可近似看作是服从均匀分布的. 在对查询进行概率建模之前, 首先介绍一个基本定义.

定义 3(可能的位置区域). 由于定位技术的局限性或者其它原因引起的定位偏差, 只能根据已知信息如某些相关对象的坐标等, 确定一个目标的位置在这些相关对象决定的某个平面区域内, 那么这个区域就称为定位目标的一个可能位置区域(Potential Location Region, PLR).

若一个阅读器在某一时刻同时探测到多个标签, 则阅读器的位置可以确定为在这些标签的 BDR 重叠区域内, 那么这个重叠区域就是这个阅读器的一个 PLR. 但它不一定是最小的 PLR, 还有可能根据读数不包含的某个标签子集, 将阅读器排除在某些区域以外, 进一步缩小 PLR. 用元组信息及对应的存在概率 (rid, p) 或 (oid, p) 这种二元组结构作为中间结果形式, 系统根据经验数据或统计学方法等其它依据, 对可能结果进行过滤, 筛选出可能性最高的一个作为结果返回给用户. 那么在每次确定阅读

器或者半固定式标签的 PLR 时,应该尽可能地根据现有的信息找到目前可以确定的最小 PLR,便于系统仲裁出可信度较高的最终结果。

假设所有阅读器的读周期的长度均相等,并且读周期是同步的. 在每个读周期中,系统需要完成的任务和操作步骤如下。

1. 定位每个阅读器的位置:根据当前的读数确定每个阅读器的 PLR;
2. 对位置发生变化的半固定式电子标签的位置重定位或进一步缩小其 PLR;
3. 将步 1 和步 2 中的位置变化更新到信息存储中心和索引结构中;
4. 若有查询被触发,则根据更新后的信息记录和索引内容,确定查询范围,执行查询过程并将结果返回给用户。

4.1 阅读器概率定位算法

4.1.1 滑动窗口机制

滑动窗口机制仅考虑最近的 w 个元组(w 是窗口大小),新元组不断到达,窗口向前滑动,旧元组滑出窗口被淘汰. 根据窗口单位不同可分为以时间为单位的滑动窗口机制和以元组为单位的滑动窗口机制两类. 前者的窗口大小以时间为单位,用一个时间区间表示;后者的窗口大小表示窗口中保持的元组的数目。

采用滑动窗口主要是考虑到元组的作用有一定时效限制,在本研究中的场景下,游客要求在触发查询后实时的得到结果,而且由于读数是不完全连续的,在没有读数的情况下,结果的好与坏在很大程度上取决于对阅读器速度估计的准确程度. 分析在多数情况下速度的变化过程:游客的速度会根据自身状况的改变缓慢变化,比如走了很长的路,随着能量消耗,速度逐渐慢下来,这样用越接近当前时刻的历史速度标量作为没有读数区间内的平均速度估计值就会越准确. 所以在本场景下选择基于以元组为单位的滑动窗口机制,根据需要将窗口的大小取为 1. 也就是说,我们只保留最近的(t_0 时刻的)读数记录在窗口中,但并不保证一定是当前时刻(t_1)的读数记录。

4.1.2 空路径上概率定位算法

在存储读数信息的同时,也将最新的速度估计值记录下来,最新的速度估计值用式(1)计算:

$$v_{t_0} = \text{dis}(S_0 - S_{-1}) / (t_0 - t_{-1}) \quad (1)$$

其中 S_{-1} 和 S_0 分别表示阅读器在 t_{-1} 时刻和 t_0 时刻的 PLR, $\text{dis}(S_0, S_{-1})$ 表示 S_{-1} 和 S_0 的距离. 这里 S_0 和 S_{-1} 不是确定的一点,求 $\text{dis}(S_0, S_{-1})$ 时不能简单地利用两点之间的欧式距离公式计算,需要考虑

S_{-1} 和 S_0 的区域范围,下面给出计算 $\text{dis}(S_0, S_{-1})$ 的定理。

定理 2. 任意两个 PLR S 和 S' 之间的距离 $\text{dis}(S, S')$ 等于对应的平面内的两个区域的质心距。

证明. 设 S 和 S' 的概率密度函数分别为 $\rho(x, y)$ 和 $\rho'(p, q)$:

(1) 用积分计算:

如图 2 所示,在 S' 中任取一点 (p, q) ,它到 S 中任意一点 (x, y) 的距离 $d_{p \rightarrow p} = \sqrt{(x-p)^2 + (y-q)^2}$. 那么 (p, q) 到 S 的距离

$$d_{p \rightarrow r} = \frac{\iint_S d_{p \rightarrow p} \rho(x, y) dx dy}{\iint_S \rho(x, y) dx dy} = \frac{1}{A} \iint_S d_{p \rightarrow p} \rho(x, y) dx dy.$$

设 $A = \iint_S \rho(x, y) dx dy$, $B = \iint_{S'} \rho'(p, q) dp dq$, 则

$$\begin{aligned} d_{r \rightarrow r} &= \frac{\iint_{S'} d_{p \rightarrow r} \rho'(p, q) dp dq}{\iint_{S'} \rho'(p, q) dp dq} \\ &= \frac{1}{B} \iint_{S'} d_{p \rightarrow r} \rho'(p, q) dp dq \\ &= \frac{1}{AB} \iint_{S'} \rho'(p, q) \left[\iint_S d_{p \rightarrow p} \rho(x, y) dx dy \right] dp dq \\ &= \frac{1}{AB} \iint_{S'} \rho'(p, q) \left[\iint_S \sqrt{(x-p)^2 + (y-q)^2} \cdot \rho(x, y) dx dy \right] dp dq \\ &= \frac{1}{AB} \sqrt{|X|^2 + |Y|^2}, \end{aligned}$$

其中,

$$X = B \iint_S x \rho(x, y) dx dy - A \iint_{S'} p \rho'(p, q) dp dq;$$

$$Y = B \iint_S y \rho(x, y) dx dy - A \iint_{S'} q \rho'(p, q) dp dq.$$

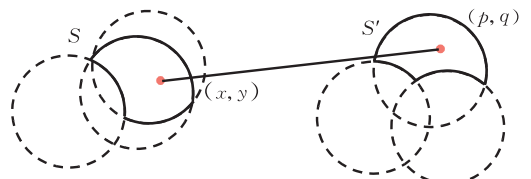


图 2 求两个 PLR 之间距离过程图

(2) 用质心计算:

S 的质心坐标为

$$\left(\frac{\iint_S x \rho(x, y) dx dy}{A}, \frac{\iint_S y \rho(x, y) dx dy}{A} \right);$$

S' 的质心坐标为

$$\left(\frac{\iint_{S'} p\rho'(p,q)dpdq}{B}, \frac{\iint_{S'} q\rho'(p,q)dpdq}{B} \right);$$

则质心距

$$\begin{aligned} d_{G \rightarrow G'} &= \sqrt{(x_{GS} - x_{GS'})^2 + (y_{GS} - y_{GS'})^2} \\ &= \frac{1}{AB} \sqrt{|X|^2 + |Y|^2}. \end{aligned}$$

综上所述 $dis(S, S') = d_{r \rightarrow r'} = d_{G \rightarrow G'}$. 证毕.

根据定理可知,用积分式(2)可以计算出 $dis(S_0, S_{-1})$ 精确的值,这样,在没有读数的情况下仍然可以对阅读器作比较准确的位置估计.假设阅读器在路径 $path(x_1, y_1, x_2, y_2)$ 上匀速前进,则当前时刻(t_1)位置可用下面方程组进行估计

$$\begin{cases} x = x_1 + \frac{v_0 \times (t_1 - t_0)}{\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}} (x_2 - x_1) \\ y = y_1 + \frac{v_0 \times (t_1 - t_0)}{\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}} (y_2 - y_1) \end{cases} \quad (2)$$

这里 $(x_1, y_1), (x_2, y_2)$ 都是已知的.

4.1.3 BDR 内的概率定位算法

当阅读器产生读数时,能够利用的已知信息更多,需要一种不同于在空路径上的定位算法.首先考虑固定式的标签,若一个阅读器在某一时刻同时探测到多个标签,则阅读器的位置可以确定为在这些标签的 BDR 重叠区域内,并根据读数不包含的某个标签子集,将阅读器排除在某些区域以外,从而缩小 PLR.例如,在某一个周期,根据读数和系统当前存储的标签位置信息,有可能被阅读器读到包括出现在读数中的标签以及与这些标签 BDR 有交叉的标签,假设这样的固定式标签共有 3 个: $tag1, tag2$ 和 $tag3$. 如图 3 所示,若当前读数有两种情况: (1) $(rid, \{tag1, tag2, tag3\}, t)$; (2) $(rid, \{tag1, tag2\}, t)$. 所确定的 PLR 分别对应图中两个阴影区域.每个固定式阅读器的位置信息记录中,要增加与自己 BDR 相交的邻近标签的概要信息(如坐标等)和指向该标签的指针.

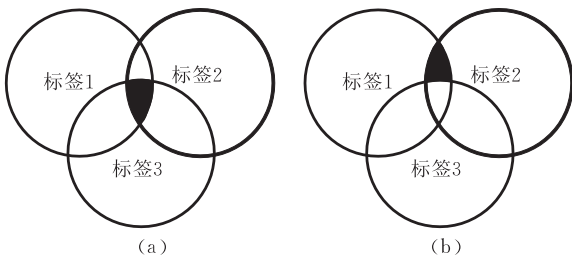


图 3 用固定式标签确定阅读器的 PLR 示意图

再用半固定式标签的位置,进一步缩小 PLR 的范围.首先,根据阅读器已确定出来的 PLR,把半固定式标签的 PLR 中存在概率为零的那部分区域过滤掉.然后用过滤后的区域估计的 PLR 与阅读器的原 PLR 进行 superpose 操作.由于可能的位置是一个区域,所以在叠加的时候需要考虑概率分布密度的变化.如图 4(a) 所示,superpose 操作的具体过程:用 A 表示已过滤的区域,在 B 中任选一点 (a, b) ,以这点为圆心、 R_{read} 为半径的圆形区域与 A 相交部分面积为 $s(a, b, A)$,所以该点的密度可表示为

$$\rho(a, b) = \frac{s(a, b, A)}{\iint_B s(x, y, A) dx dy} \quad (3)$$

然后将密度为零或接近零的点组成的区域删除掉.

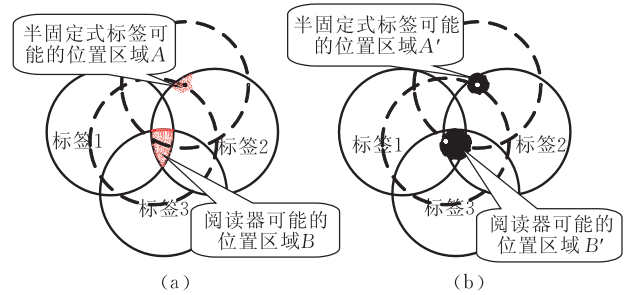


图 4 用半固定式标签确定阅读器的 PLR 示意图

将这个公式化简是一个特别繁琐的过程,以不牺牲太多的准确度为前提,将阅读器的 PLR 近似为以其质心为圆心,以其最小覆盖圆的半径为半径的圆形区域,如图 4(b),这样式(3)涉及到的区域都是规则的圆形区域,相交图形都是等半径的弓形,化简起来容易得多.

用 R_j 表示阅读器 r_j 当前的读数记录, $R_j.Oset$ 是读数记录 R_j 在属性 oid 方向上的投影运算,即表示出现在 R_j 中的 oid 集合.假设当前系统中的标签总数为 M ,阅读器数目为 N ,整个园区的范围用 U 表示.具体描述如算法 1.

算法 1. 基本 BDR 概率估计算法 (Basic BDR Probability Evaluation Algorithm).

Input: 每个 r_j 的当前读数记录;每个 o_i 的位置信息

Output: 每个 r_j 的位置信息

1. for $j=1$ to N do
2. $A = \{oid | oid.type = "fixed" \wedge oid \in R_j.Oset\}$;
3. $B = \{oid | oid.type = "unfixed" \wedge oid \in R_j.Oset\}$;
4. $U \rightarrow S_j$; $D = \emptyset$;
5. for each $i \in A$ do
6. $S_j \cap C_i^j \rightarrow S_j$;
7. for $i=1$ to M do

8. if $o_i.oid \notin R_j.Oset \wedge C_{o_i.oid}^j \cap S_j \neq \emptyset$ then
9. $\{o_i.oid\} \cup C \rightarrow C$
10. for each $i \in C$ do
11. $S_j - C_i^j \rightarrow S_j$;
12. if $B \neq \emptyset$ then
13. for each $i \in B$ do,
14. if $o_i.position$ is a point $\wedge C_{o_i.oid}^j \cap S_j \neq \emptyset$ then
15. $superpose(S_j, C_{o_i.oid}^j, 1)$;
16. if $o_i.position$ is a region $\wedge C_{o_i.oid}^j \cap S_j \neq \emptyset$ then
17. $superpose(S_j, C_{o_i.oid}^j, \rho(x, y))$;
18. return $\{S_j | j=1 \text{ to } N\}$.

4.2 标签概率位置更新算法

假设半固定式标签是以小概率大范围移动的,新位置处的 BDR 与原来位置上的 BDR 没有交叉.那么如果阅读器的读数中包含了本不应该读到的标签(根据当前存储的位置信息判断),我们就可以推断出这个标签的位置一定发生了变化.为了使其能够在以后的周期中参与阅读器的定位,帮助其缩小 PLR、提高精确度,我们需要对这样的标签重定位,覆盖掉存储中心处已过时的位置信息.对半固定式标签的位置更新是一个持续的学习过程,即使本周期中没有查询被触发,也要继续学习,逐渐缩小其 PLR.当 PLR 缩小到一定程度时($\leq S_{\min}$),可取里面任意一点近似作为它的精确坐标(S_{\min} 值的选取与 R_{read} 值有关).

半固定式标签的位置发生变化,被某一阅读器探测到了,那么这个标签的新位置一定在阅读器的探测范围内,将阅读器的位置近似为其 PLR 质心的位置简化计算过程.根据同一周期不同阅读器或不同周期同一阅读器的读数,多重定位,再进行 superpose(定位算法和 superpose 算法同上一小节).

4.3 概率范围查询算法

如果此周期系统收到用户通过阅读器发出查询,计算满足查询条件的结果集: $results = \{(oid, p) | oid.category = query.category \wedge p\{dis(oid, rid) \leq r\} > P_{\min}\}$,并及时返回给用户.例如,游客要查询附近 500m 之内的卖相机电池的移动商户有哪些,解决这个问题的关键是确定 $p\{dis(oid, rid) \leq r\}$ 的值.由前两小节可知,阅读器和标签经过算法定位后,所得的 PLR 都近似成概率分布密度不均匀的圆形区域.如图 5(a) 所示,当前某个阅读器同时探测到了停车场、洗手间、冷饮厅附近布置的标签,经过定位后得到其 PLR 为图 5(b) 右上角圆形区域 A.

图 5(b) 中,左下角的圆形区域 B 是一个可能在查询范围内的目标的 PLR,它与查询范围边界(用虚线表示)的相交情况如图所示,圆形区域内的颜色深浅与概率分布密度成正比.下面给出存在概率的形式化定义和求值定理及证明过程.

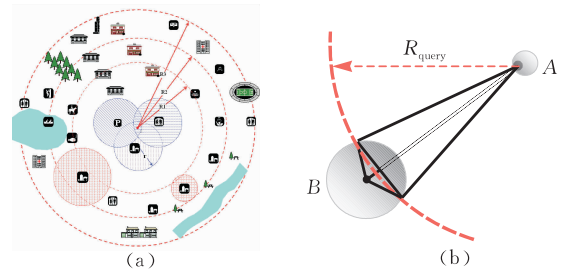


图 5 计算查询结果的存在概率过程示意图

定义 4(存在概率). 若一个查询 q 的查询范围是 R_{query} ,对于触发查询的阅读器 r_j 在其 PLR 内的每个点处,与目标 o_i 的距离小于 R_{query} 的概率平均值为 $p_{\text{avg}} = p\{dis(o_i, r_j) \leq R_{\text{query}}\}$.如果 p_{avg} 大于概率阈值 P_{\min} ,那么 o_i 属于查询结果集.这个与 P_{\min} 作比较的参数 $p\{dis(o_i, r_j) \leq R_{\text{query}}\}$ 就叫做 o_i 对于查询 q 的结果存在概率.

定理 3. 若触发查询 q 的阅读器 r_j 的 PLR 为 A,阅读器在 A 中的概率密度函数为 $\rho_1(x, y)$;某个可能成为结果的目标 o_i 的 PLR 为 B, o_i 在 B 中的概率密度函数为 $\rho_2(x, y)$,并且 q 的查询半径为 R_{query} ,则

$$p\{dis(o_i, r_j) \leq R_{\text{query}}\} = \frac{1}{\pi r^2} \iint_A \rho_1(p, q) \left[\iint_{S_{pq}} \rho_2(x, y) dx dy \right] dp dq,$$

其中 S_{pq} 是与 (p, q) 坐标有关的区域,是 r_j 在点 (p, q) 处时查询范围与 B 的重合区域; r 是 B 的半径.

证明. 如图 5(b) 所示,在 A 中任取一点 (p, q) 作为 r_j 的位置坐标,则此时 o_i 在查询范围内的概率为

$$P(x, y) = \frac{\iint_B \rho_2(x, y) dx dy}{\iint_B \rho_2(x, y) dx dy},$$

对于 A 内所有位置的概率平均值为积分 $\iint_A P(x, y) \cdot \rho_1(x, y) dx dy$ 的值.又 B 内所有位置的概率之和为 1,则 $\iint_B \rho_2(x, y) dx dy = \pi r^2$,那么

$$p\{dis(o_i, r_j) \leq R_{\text{query}}\} = \frac{1}{\pi r^2} \iint_A \rho_1(p, q) \left[\iint_{S_{pq}} \rho_2(x, y) dx dy \right] dp dq. \quad \text{证毕.}$$

5 基于网格划分的移动概率查询优化策略

上一节中我们主要讨论了移动式 RFID 设备监控环境下的基于概率的范围查询的处理模型,对于可移动对象定位的不确定性描述用到了概率密度函数.它是一种连续型的表现形式,当计算在 PLR 的某个子区域内分布概率值时需要积分运算,由于在定位和取查询范围的交集部分所得的子区域都是不规则的,直接积分十分复杂,近似积分会降低准确性也需要做规则图形(圆)上的积分运算,会影响到实时查询的处理速度和响应时间. Kalashnikov 等人^[15]在 2006 年曾提出了利用直方图表示分布不确定性的方法,将连续的概率密度函数离散化,变积分运算简化为少量概率数值的累加运算问题显著降低了计算复杂度和计算开销;特别的,如果是均匀分布,则只是网格单元的个数比计算.下面将上一节中提出的基本算法引入直方图+四叉树算法和 U-grid 索引的结构和网格划分的思想做优化设计,提出适合于本研究场景中的基于移动式 RFID 技术的概率化的移动范围查询的高效算法 m^2 ROvgP-MRQ (RFID mobile reader enabled Moving Range Query on moving objects based on virtual grid partition algorithm)和索引结构 CU-grid.

5.1 基于虚拟网格划分的移动概率查询优化算法

5.1.1 基本结构建立

根据虚拟网格划分的思想,将整个 RFID 监控的二维空间划分成 $\delta \times \delta$ 个面积相等的方形区域,每个方形区域称为网格体系 G 的一个单元,面积为 σ ,用指针 G_{ij} ($i, j \in [1, \delta]$) 表示.下面重新定义一些基本概念.

定义 5(可能的位置网格单元集). 基于网格划分的空间中,PLR 包围的范围不再是任意形状和大小的区域,对于一个网格单元只有两种关系:或者完整的包含,或者完全不包含.根据定义将基于坐标的 PLR 转化成基于虚拟网格划分空间的表示形式,就是可能的位置网格单元集 PLgcS(Potential Location gridding cell Set).

如图 6 所示,引入这种思想会使我们失去 PLR 的边界信息和概率分布的细节信息,因为这种思想是基于在每个网格单元中的概率分布是均匀的,概率值等于

$$p(G_{ij}) = \iint_{G_{ij}} \rho(x, y) dx dy \quad (4)$$

那么网格单元粒度越大,丢失的信息越多,计算量越少,所以对于网格单元粒度的选取应该折中进行考虑.

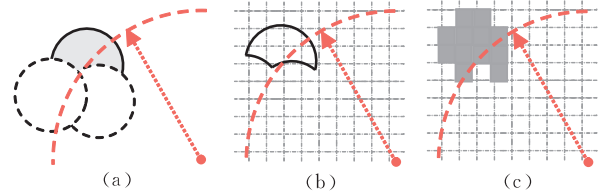


图 6 PLR 和与之对应的 PLgcS

为了提高网格表示方式的信息提取速度,在网格上建立带有聚合信息的四叉树,而且文献^[15]在索引结构上考虑了压缩技术,用合并叶子结点的贪婪算法减小存储和 I/O 开销.由于对半固定式标签的重定位和用半固定式标签定位阅读器,都是 PLgcS 与 PLgcS 的 superpose 过程,需要考虑不均匀的概率分布密度,这样会出现某些边缘“很薄”的现象.所以我们提出了网格单元层面上的压缩技术:对于某个边缘 G_{ij} ,若 $\{G_{ij} | p(G_{ij}) \leq P_{door}\}$,则忽略这个分布单元,其中 P_{door} 是判断某个 PLgcS 是否应该包含 G_{ij} 的概率阈值参数.具体描述如算法 2.

算法 2. PLgcS 压缩算法(PLgcS compression Algorithm).

Input: PLR

Output: 压缩后的 PLgcS

1. $x_{min} = \min\{x | (x, y) \in PLR\}$;
 $x_{max} = \max\{x | (x, y) \in PLR\}$;
2. $y_{min} = \min\{y | (x, y) \in PLR\}$;
 $y_{max} = \max\{y | (x, y) \in PLR\}$;
3. 设 N 为整数集;
4. $n_1 = \max\{n | dn \leq x_{min} \wedge n \in N\}$;
5. $n_2 = \min\{n | dn \geq x_{max} \wedge n \in N\}$;
6. $m_1 = \max\{n | dn \leq y_{min} \wedge n \in N\}$;
7. $m_2 = \min\{n | dn \geq y_{max} \wedge n \in N\}$;
8. $PLgcS = \emptyset$;
9. for $i = n_1$ to n_2 do
10. for $j = m_1$ to m_2 do
11. if $PLR \cap G_{ij} \neq \emptyset$ then
12. $p(G_{ij}) = \iint_{G_{ij}} \rho(x, y) dx dy$;
13. if $p(G_{ij}) > P_{door}$ then
14. $PLgcS = PLgcS \cup \{G_{ij}\}$;
15. return PLgcS.

同四叉树结构的索引层面上的压缩技术一样,这也是一种有损的压缩算法. P_{door} 值越高,损失的信息越多,影响准确性程度越高,但存储和 I/O 开销会越少,所以对于 P_{door} 的选取应该折中进行考虑.也可以根据不同的情况,可以 P_{door} 选取多个不同的值,如 G_{ij} 单独在一个四叉树索引结构的一个空的叶子结点中时, P_{door} 可以相对高一些.

定义 6(被探测网格单元集). 基于网格划分的空间中,(与定义 4 类似)根据定义将基于坐标的 BDR 转化成基于虚拟网格划分空间的表示形式,就是可能的位置网格单元集 BDgcS (Being Detected gridding cell Set).

根据式(4)知每个网格单元的概率同网格单元方形区域与 BDR 相交部分的面积成正比.那么会出现边缘部分可能“较薄”的现象,即概率分布密度在网格单元内均匀,在网格单元间不均匀.将“过薄的边缘”部分忽略掉,可减小计算量.

在本研究的场景中,除了建模 BDgcS、PLgcS 的位置区域外,还需要对路径的位置信息进行建模,而文献[15]完全没有考虑路径的问题.存储中心中在记录每条分割后的原子空路径的起始坐标 (x_1, y_1) 和终止坐标 (x_2, y_2) 的同时,还要记录其起始网格单元 G_s 和终止网格单元 G_e 及路过的网格单元 G_m ; 每个网格单元除了记录其包含的读者和标签的位置信息外,还需要在每个路径的 $G_s(G_e)$ 中存储路径的起始(终止)坐标和与边界的交点坐标,在 G_m 中存储与边界的两个交点坐标.

5.1.2 对基本算法的修改

根据虚拟网格划分得到适合 RFID 技术应用场景下的 m^2 ROvgP-MRQ 算法包括: PLE Algorithm、udO Algorithm、reLocTag Algorithm、CU-gridQ Algorithm. 由于 CU-gridQ Algorithm 的算法过程涉及到索引部分的内容,具体的算法过程将在下一小节中与 CU-grid 索引结构一同阐述.

(1) 路径上的位置估计算法 (PLE Algorithm)

设两个时刻的 PLgcS 分别为 $\{G_i\}$ 、 $\{G_j\}$, 包含的网格单元个数分别为 m 、 n , 那么两个 PLgcS 的质心坐标分别为 $(\sum_1^m G_i.x \cdot p(G_i), \sum_1^m G_i.y \cdot p(G_i))$ 、 $(\sum_1^n G_j.x \cdot p(G_j), \sum_1^n G_j.y \cdot p(G_j))$. 那么两个时刻的 PLgcS 的位置 S 和 S' 之间的距离可根据定理 2 求得. 根据式(1)可计算出最新的速度估计值 v_{t_0} , 则可根据式(2)估计阅读器在某个路径上的位置坐标.

(2) BDgcS 内的位置估计

与基于坐标的 BDR 不同, BDgcS 的原子单位是网格单元, 而且单元之间的分布密度不同, 那么 BDgcS 之间的叠加(减)运算要考虑分布密度. 将固定式标签定位算法(算法 2)中虚线框的部分修改如下, 得到更新后的算法(udO Algorithm)如算法 3 所示, $\bar{\cap}$ 和 \simeq 过程仅第 2 步不同, $/ * * /$ 中为 \simeq 的内容.

算法 3. 更新操作算法 (udO Algorithm).

Input: 操作数 BDgcS: A, B

Output: 操作结果 $\{(G_k, p(G_k))\}$

1. 设 $A = \{(G_i, p_A(G_i))\}$, $B = \{(G_j, p_B(G_j))\}$;
2. 计算 $result = \{G_k | G_k \in A \wedge G_k \in B\}$;
/* 计算 $result = \{G_k | G_k \in A \wedge G_k \notin B\}$ */;
3. $N = Num(result)$;
4. $p_{sum} = 0$;
5. for $k = 1$ to N do
6. $p(G_k) = p_A(G_k) \cdot p_B(G_k)$;
7. $p_{sum} += p(G_k)$;
8. for $k = 1$ to N do
9. $p(G_k) = p(G_k) / p_{sum}$;
10. return $\{(G_k, p(G_k)) | G_k \in result\}$.

用半固定式标签的位置信息可以减少阅读器 BDgcS 中包含的网格单元格的数目, 具体的过滤算法(filter-BDgcS Algorithm)的核心就是修改基本算法中的 superpose 过程. 从 PLE Algorithm 过程中知道如何确定一个网格单元集 $\{G_k\}$ 表示的区域的质心坐标, 那么在 filter-BDgcS 算法中我们用其 PLgcS 的质心坐标来近似半固定式标签的位置. 若某一周期阅读器读到了半固定标签, filter-BDgcS 具体过程描述如算法 4(没读到时的情况用 $/ * * /$ 中的内容替代同一行的语句部分).

算法 4. BDgcS 过滤算法 (filter-BDgcS Algorithm).

Input: 阅读器的 PLgcS: $\{(G_i, p_1(G_i))\}$

半固定标签的 PLgcS: $\{(G_j, p_2(G_j))\}$

Output: 过滤后的阅读器 PLgcS: $\{(G_k, p(G_k))\}$

1. 求得标签质心坐标

$$\left(\sum_1^n G_j.x \cdot p_2(G_j), \sum_1^n G_j.y \cdot p_2(G_j) \right);$$

2. 求得坐标为 (x, y) 处的 BDgcS: $\{(\tilde{G}_j, p_3(\tilde{G}_j))\}$;
3. $G = \{G_i\} \cap \{\tilde{G}'_j\}$; /* $G = \{G_i\} - \{\tilde{G}'_j\}$; */
4. $N = Num(G)$; $p_{sum} = 0$;
5. 设 $G = \{(G_k, p(G_k))\}$;
6. for $k = 1$ to N do
7. $p(G_k) = p_1(G_k) \cdot p_3(G_k)$; /* NOP 空操作 */

8. $p_{\text{sum}} += p(G_k)$;
9. for $k=1$ to N do
10. $p(G_k) = p(G_k) / p_{\text{sum}}$;
11. return $\{(G_k, p(G_k))\}$.

标签的重定位算法(reLocTag Algorithm)与基本算法的过程相同,但会用到上面提到的修改后的算法,由于空间的关系省略重复的过程描述。

5.2 基于虚拟网格划分的移动概率查询索引结构

Kalashnikov 等人^[15]提出了 U-grid(Uncertain grid, 不确定网格)的索引结构,作为移动对象的不确定性位置的索引。首先,他们用直方图表示位置的不确定性,然后在直方图上建立四叉树基本索引结构,最后在四叉树索引之上建立 U-grid 索引。

同四叉树索引的建立类似,以四叉树中叶子节点表示的方形区域为基本单元 L_{ij} 构成网格系统 L ,在其上建立 U-grid 索引。在每个单元中存放不确定区域与这个单元相交的对象的某些聚合信息,用于索引层次上的辅助过滤。此外,每个单元还存放了一个指向列表的指针,列表中存放了这些对象各自的基本信息,作为对象层次上的过滤依据。

经过对移动对象的各类概率查询的特征做分析,我们总结出查询过程在本质上就是对移动对象进行层层过滤的过程。过滤内容主要分为:对不确定位置的过滤、对存在概率值的过滤和对其它具体属性的过滤。比如 MBR、概率上限和概率和等。在现实生活和实际应用中,常常按某种属性将对象分属为几个不同类别范畴(category),例如:我们进入某个论坛,常会根据自己关心的话题,选择一个按内容属性划分出的几个类别选项,缩小搜索空间等。基于这种思想,我们对 U-grid 索引结构进行扩展,提出适合本研究场景或更广泛的实际应用的索引结构 CU-grid(category-based Uncertain grid, 基于类别的不确定网格),将基于类别的概要信息存储在每个单元中。基于类别的概要信息可理解为一种二维的索引概要信息:一维是按照 category 进行分组,另一维是组内所有对象的聚合信息如 MBR、概率上限、概率和等。这种结构体现了聚合的思想和概要存储的思想。

移动范围查询的查询范围 γ_{query} 是以触发查询的阅读器位置为圆心、 R_{query} 为半径的圆形区域,用其 PLgcS 的质心坐标近似阅读器的位置。为了计算方便,不必考虑查询范围的概率分布情况,将 γ_{query} 与每个网格单元相交的扇形区域近似为三角形区域(忽略弓形区域),则重叠部分就是一个梯形。用 λ_{min}

作为面积比例系数阈值,如果重叠部分的面积大于 $\lambda_{\text{min}} \cdot \sigma$,则认为完全包括这个单元,否则表示完全不包含。根据建立的 CU-grid 索引的查询算法(CU-gridQ Algorithm)具体描述如算法 5。

算法 5. CU-grid 索引查询算法(CU-gridQ Algorithm)。

Input: 触发查询的阅读器标识 rid , γ_{query} , 查询类别 c , 查询结果概率阈值 P_{min} , 其它属性条件 $conds$

Output: 查询结果集 $\{(oid, p)\}$

1. 通过 rid 可做 Hash 运算得到其 PLgcS 为 $G_0 \{(G_j, p(G_j))\}$;
2. 标签质心坐标为

$$(x_0, y_0) = (\sum_1^m G_j.x \cdot p(G_j), \sum_1^m G_j.y \cdot p(G_j));$$
3. 查询范围的 PLgcS 为

$$G_1 = \{G_j \mid G_j \in G_0, G_j \cap \gamma_{\text{query}} > \lambda_{\text{min}} \cdot \sigma\};$$
4. 令 $results = \emptyset$; $oids = \emptyset$;
5. $recg = MBR(G_1)$; $ls = \{L_{ij} \mid recg \cap L_{ij} \neq \emptyset\}$;
6. 用 $recg$ 表示的区域范围查询 CU-grid 索引;
7. for $i=1$ to N do
8. if $MBR(ls[i][c]) \cap G_1 = \emptyset \vee ls[i][c].p_{\text{sum}} < P_{\text{min}}$ then
9. $ls = ls - \{ls[i]\}$;
10. 用同样的查询条件过滤四叉树索引至叶子节点层,得到有可能存在查询结果的网格单元集 G_2 , $M = Num(G_2)$;
11. for $i=1$ to M do
12. $M' = Num(G_2[i].oid)$;
13. for $j=1$ to M' do
14. if $p(G_i.oid[j]) \geq P_{\text{min}} \wedge conds(G_i.oid[j]) = \text{true}$ then
15. if $G_i.oid[j] \notin results.oid$ then
16. $results = results \cup \{(G_i.oid[j], G_i.oid[j].p)\}$;
17. else $results[indexof(G_i.oid[j])].p += G_i.oid[j].p$;
18. return $results$.

6 实验

在这节中,我们用实验结果分析一下本篇文章提出的模型和方法的性能及有效性。实验的所有算法使用 C++ 语言实现,执行环境是 Intel Core 2 Duo 2.00 GHz 处理器,2.0GB 内存,120GB SATA 硬盘,Windows XP 操作系统。

6.1 实验设置

我们从执行效率和结果的准确性两方面说明优化算法和索引结构的有效性。表 1 总结了实验涉及

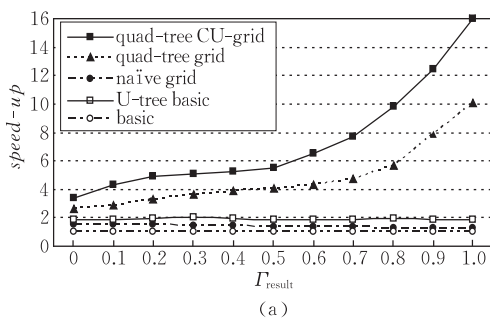
参数的表示符号、意义及参数的可能取值(粗体表示默认情况下的取值). 默认情况下我们将场景模拟为 7.5km^2 的园区, 划分为 400×400 的虚拟网格系统, 每个单元的边长大约为 5m . 根据场景的环境特点, 我们取半固定式标签所占比例为 5% , 移动概率为 0.3 , 移动周期为 1h , 每次移动距离大于 500m . 页面大小设为 1024bytes , 压缩 PLgcS 的概率阈值参数 $P_{\text{door}}=0.1$, 面积比例系数阈值 $\lambda_{\min}=0.5$.

表 1 实验参数表

符号	取值范围	意义
R_{query} (m)	1000, 500~2000	范围查询范围(半径)
R_{read} (m)	250, 100~400	阅读器的探测范围(半径)
Γ_{result}	0.5, 0.1~1	成为结果成员的概率阈值
grid size	400, 200~800	网格空间基数
N_{category}	5, 1~10	分类 category 的数量
N_{tag}	20000, 10000~30000	电子标签总数
N_{reader}	2000, 1000~3000	系统中的阅读器数量

6.2 实验结果分析

我们以利用密度函数计算概率的算法作为参比的基本算法(basic), 与引入 U-tree 索引的基本算法(U-tree basic)、单纯利用网格划分作优化的算法(naïve grid)、引入基于网格的四叉树索引结构的算法(quad-tree grid)和引入 CU-grid 索引结构的优化算法(quad-tree CU-grid)作比较. 为了将优化算法中各个组成部分对性能的影响表现得更加明显, 结果数据将采用与 basic 的各项评估参数值之比(或倒数之比)的形式.

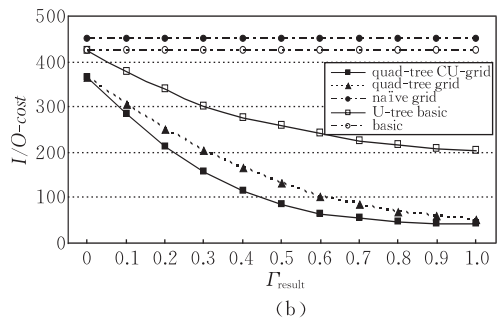
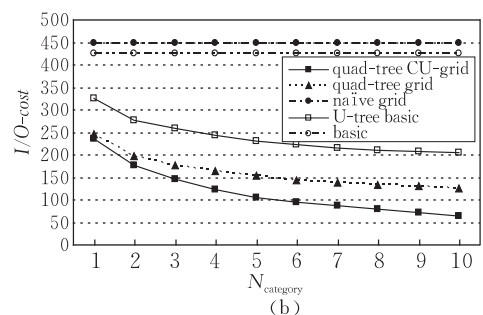
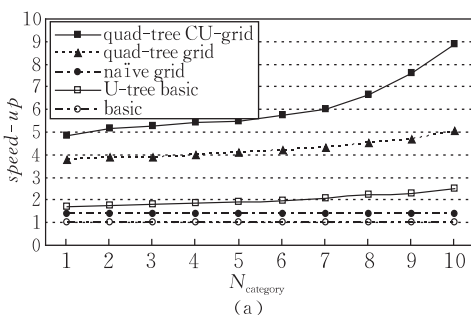


6.2.1 查询效率方面

本节从查询处理速度提高的程度($speed-up$)和 I/O 通信量降低的程度($I/O-cost$)两方面对比 4 种算法的查询效率. $speed-up$ 是其他算法与基本算法对于每个查询的平均执行时间倒数之比, 即处理速度之比; $I/O-cost$ 包括索引和数据两方面内容.

图 7 反映了结果概率阈值 Γ_{result} 与 $speed-up$ 和 $I/O-cost$ 的变化关系. 如图 7(a) 所示, 随着 Γ_{result} 升高, 以 basic 为基准, quad-tree grid 和 quad-tree CU-grid 的 $speed-up$ 变化明显提高幅度大, 其它效果不明显; 如图 7(b) 所示, 随着 Γ_{result} 升高, basic 和 naïve grid 的 $I/O-cost$ 无明显变化, 其他算法的 $I/O-cost$ 均降低. 原因是 quad-tree CU-grid 在 CU-grid 索引层面上开始就通过 Γ_{result} 的值过滤目标对象, 而 quad-tree grid 和 U-tree basic 在 quad-tree 索引层面上进行过滤, naïve grid 在读磁盘之前没有用 Γ_{result} 对目标进行过滤. 虽然 quad-tree grid 和 U-tree basic 也过滤了一部分目标, 但增加的索引 $I/O-cost$ 不能被忽略.

如图 8(a) 所示, quad-tree grid 和 quad-tree CU-grid 的 $speed-up$ 提高很多, U-tree basic 也有少量的提高, 其他算法无明显变化; 如图 8(b) 所示, U-tree basic, quad-tree grid 和 quad-tree CU-grid 的 $I/O-cost$ 减少了很多, 其他算法无明显变化. 原因是 quad-tree CU-grid 在 CU-grid 索引层面上就开始通过 category 的值过滤目标对象, 而 quad-tree grid 和 quad-tree basic 在 quad-tree 索引层面上进

图 7 Γ_{result} 对查询效率的影响图 8 N_{category} 对查询效率的影响

行过滤, naive grid 和 basic 则是在最后对象层面上用 category 对存在概率的属性值加以判断, 没有起到过滤的作用. 由于通过 category 的值可以过滤的目标对象较多, 增加的索引 $I/O\text{-cost}$ 可以忽略.

6.2.2 准确性方面

$recall$ 和 $precision$ 是经常用来评估解决带有不确定性的问题的准确程度的参数. 在本文中, 对 $recall$ 和 $precision$ 的定义如下: 假设用前文某种概率算法得到的查询结果的元素集合用 $result_p$ 表示, 而该查询的准确结果的元素集合用 $result_acc$ 表示, 并且 $tp = result_acc \cap result_p$, $fn = result_acc - result_p$, $fp = result_p - result_acc$, 则 $recall = \frac{tp}{(tp+fn)}$, $precision = \frac{tp}{(tp+fp)}$. 在进行多次实验时, 取每次 $recall$ 值和 $precision$ 值的平局数作为参

数的取值进行比较. 由于 quad-tree grid 与 naive grid 的准确性相同, U-tree basic 和 basic 的准确性相同, 则只考虑 basic、naive grid 和 quad-tree CU-grid 的 $recall$ 和 $precision$ 即可.

图 9 反映了 R_{query} 对各个算法得出的查询结果准确性的影响. 如图 9 所示, 随着查询范围的扩大, 所有算法的 $recall$ 和 $precision$ 都有提高, basic 的 $recall$ 值及总体准确性最高, 但其他算法的 $recall$ 和 $precision$ 都能达到 92% 以上. 原因是 basic 算法在基于概率定位过程中使用积分运算, 对 PLR 和 BDR 的边界信息等保存得较为完整, 因此对位置的可能性估计更准确. 而基于网格的优化算法, 忽略了一些概率小的位置, 用累加计算代替积分计算, 而且在建立索引结构时还使用了有损的压缩存储技术等, 因此准确性要稍低一些.

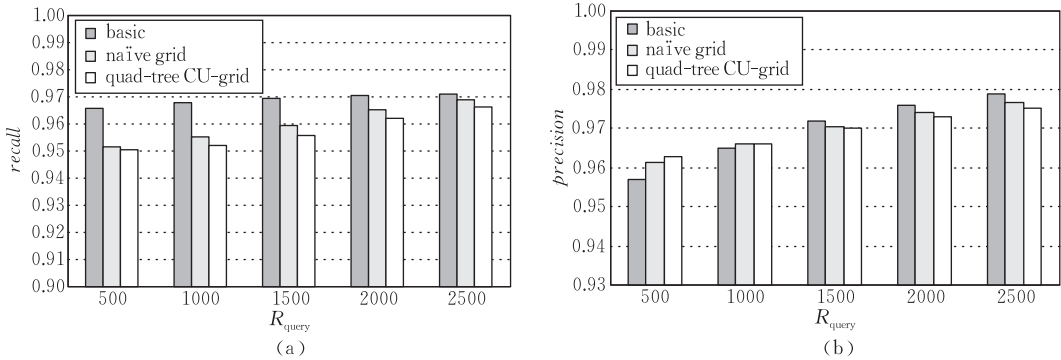


图 9 R_{query} 对查询结果准确性的影响

图 10 反映了 grid size 对各个算法准确性的影响. 如图 10 所示, 随着 grid size 的扩大, 除 basic 外, 所有算法的 $recall$ 和 $precision$ 都有提高, basic 的准确性不受 grid size 的影响, 所有算法的 $recall$ 和 $precision$ 都能达到 93% 以上. 原因是 basic 算法是基于坐标的定位算法没有划分网格单元. grid size

增大, 而整个园区的范围没有改变, 则每个网格单元覆盖的面积减小. 网格单元是基于网格划分的优化算法过程中计算位置不确定性及概率分布的最小单位, 那么网格单元面积越小, 计算的结果就相对越准确.

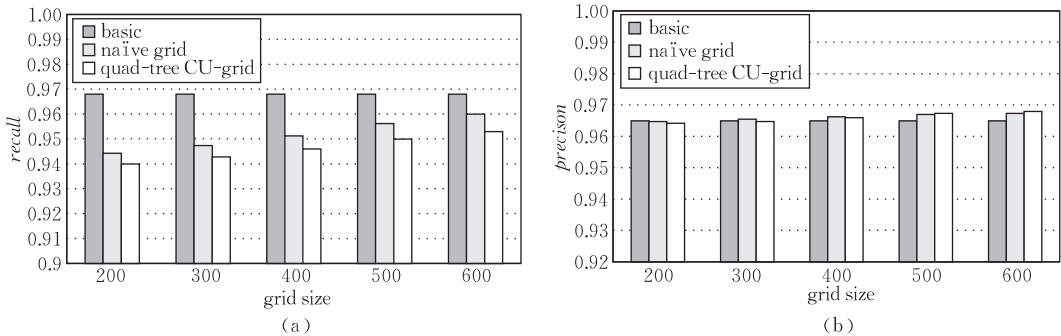
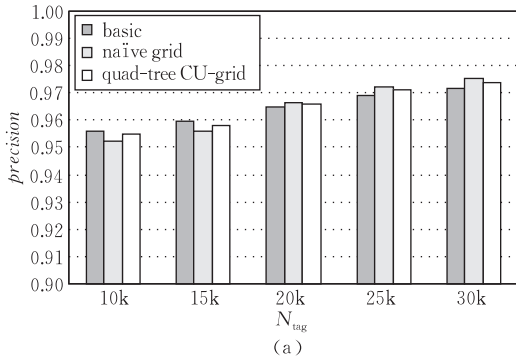


图 10 grid size 对查询结果准确性的影响

如图 11 所示, 随着 N_{tag} 的增大, 所有算法的 $recall$ 和 $precision$ 都有提高. 原因是标签的数量增加, 整个园区中分布在单位面积的标签数量增加, 因

为更多区域叠加后的结果范围会更小, 那么对阅读器的 PLR 范围就会缩小 (PLGcS 包含的网格单元数量减少), 确定比较准确的位置信息需要的周期就会

减少. 虽然半固定式的标签数量也会增多, 会增加定位的不确定性, 但其所占比例少概率小, 而且每个周期都会根据多个阅读器的读数对其位置进行修正, 查询却不是每个周期必有的. 所以增大 N_{tag} 的数量



可以提高算法结果的准确性. 从图中数据我们还可以知道, 标签的数量不需要过多, 算法就可以有比较高的 *recall* 和 *precision*, 也体现了本文提出模型和算法的有效性.

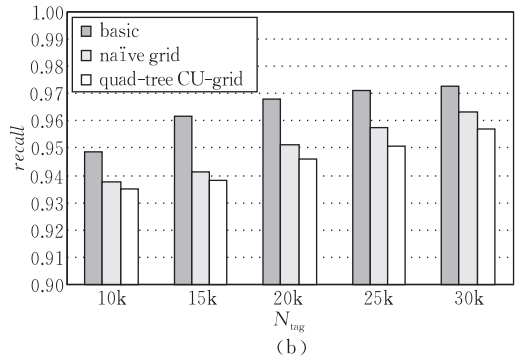


图 11 N_{tag} 对查询结果准确性的影响

综上, 实验结果证明了本文提出的概率模型上的优化算法和索引结构具有良好的性能, 在保证较高准确性的前提下, 有效地提高了查询处理速度, 降低了 I/O 通信代价, 能够有效地支持本研究场景及其它需求类似的应用.

7 结 论

本文利用不同于以往的 RFID 系统的组织结构——依靠位置相对固定的标签来定位携带移动式的阅读器监控对象, 提出了移动范围查询的一种概率处理模型. 并在此基础上, 对 Kalashnikov 等人提出的用直方图表示不确定性的方法和 U-grid 索引结构进行扩展, 讨论了基于虚拟网格划分的 m^2 ROvgP-MRQ 优化算法和 CU-grid 索引结构. 实验结果从查询效率和结果的准确性两方面证明了本文提出的概率模型上的优化算法和索引结构具有良好的性能及有效性.

参 考 文 献

- [1] Pereira D P, Dias W R A, Braga M L et al. Model to integration of RFID into wireless sensor network for tracking and monitoring animals//Proceedings of IEEE-CSE. São Paulo, Brazil, 2008; 125-131
- [2] Jeffery S R, Alonso G, Franklin M J et al. A pipelined framework for online cleaning of sensor data streams//Proceedings of ICDE. Atlanta, GA, USA, 2006; 140-142
- [3] Jeffery S R, Garofalakis M, Franklin M J. Adaptive cleaning for RFID data streams//Proceedings of VLDB. Seoul, Korea, 2006; 163-174
- [4] Wang F S, Liu S, Liu P. Bridge physical and virtual worlds;

Complex event processing for RFID data streams//Proceedings of EDBT. Munich, Germany, 2006; 588-607

- [5] Welbourne E, Khoussainova N, Letchner J et al. Cascadia: A system for specifying, detecting, and managing RFID events//Proceedings of MobiSys. Breckenridge, CO, USA, 2008; 281-294
- [6] Re C, Letchner J, Balazinksa M et al. Event queries on correlated probabilistic streams//Proceedings of SIGMOD. Vancouver, BC, Canada, 2008; 715-728
- [7] Jacobsen R, Nielsen K F, Popovski P, Larsen T. Reliable identification of RFID tags using multiple independent reader sessions//Proceedings of the IEEE RFID 2009. Orlando, FL, USA, 2009; 64-71
- [8] Vorst P, Zell A. Semi-autonomous learning of an RFID sensor model for mobile robot self-localization//Proceedings of EUROS. Prague, Czech Republic, 2008; 273-282
- [9] Vorst P, Schneegans S, Yang B, Zell A. Self-localization with RFID snapshots in densely tagged environments//Proceedings of IROS. Nice, France, 2008; 1353-1358
- [10] Tran T, Sutton C, Cocci R, Nie Y, Diao Y, Shenoy P. Probabilistic inference over RFID streams in mobile environments//Proceedings of ICDE. Shanghai, China, 2009; 1096-1107
- [11] Bapat T, Candan K S, Cherukuri V S, Sundaram H. AURA: Enabling attribute-based spatial search in RFID rich environments//Proceedings of ICDE. Shanghai, China, 2009; 1211-1214
- [12] Bapat T, Candan K S, Cherukuri V S, Sundaram H. Information-gradient based decentralized data management over RFID tag clouds//Proceedings of MDM. Taipei, Taiwan, China, 2009; 693-696
- [13] Tao Y, Cheng R, Xiao X et al. Indexing multi-dimensional uncertain data with arbitrary probability density functions//Proceedings of VLDB. Trondheim, Norway, 2005; 922-933
- [14] Tao Y, Papadias D, Sun J. The tpr*-tree: An optimized spatio-temporal access method for predictive queries//Proceedings of VLDB. Berlin, Germany, 2003; 790-801

- [15] Kalashnikov D V, Ma Y, Mehrotra S, Hariharan R. Index for fast retrieval of uncertain spatial point data//Proceedings of ACM-GIS. Arlington, VA, USA, 2006: 195-202
- [16] Gedik B, Wu K L, Yu P S et al. Processing moving queries

over moving objects using motion-adaptive indexes. IEEE Transactions on Knowledge and Data Engineering, 2006, 18 (5): 651-668



GU Yu, born in 1981, Ph. D. candidate. His main research interests include data stream and RFID data management.

GUO Na, born in 1985, M. S. candidate. Her current major research interests focus on RFID data management.

YU Ge, born in 1962, Ph. D., professor, Ph. D. supervisor. His major research interests include database theory and technology.

Background

For the applications of location-based services such as intelligent transport systems and object tracking systems, constructing indexes and processing queries about the positions of moving objects are widely studied. Compared to the traditional positioning technology such as GPS and GIS, RFID technology is relatively cheap and predominant.

This paper contributes on the field of RFID data management. It proposes an RFID solution for real-time locating the moving objects. Although some researches have studied the organization and structure of this kind of RFID system, the technology and the basic model proposed by this paper is different. The model and method of these works are not suitable for the problem and scenario, which can not support the real-time probabilistic moving range query effectively.

This paper proposes to utilize a novel RFID system framework, using relatively fixed tags to locate moving objects with portable readers, to support efficient probabilistic spatial moving range query.

This research was supported by the National Natural Science Foundation of China under grant No. 60773220 and No. 60873009.

The authors of this paper have been engaged in RFID data management and RFID event stream query processing research, with special attention to articles about cleaning and complex event processing over RFID data. These studies have laid an important theoretical basis and the background knowledge. By these studies, they have published some papers in this area.