

基于网格的传感器网络 K 近邻查询处理算法

刘宇雷 秦小麟 沈佳佳

(南京航空航天大学信息科学与技术学院 南京 210016)

摘 要 综合考虑了能量消耗、查询延迟、查询结果正确性等因素,提出了一种基于网格的传感器网络 K 近邻查询处理算法 GKNN。它优化现有的查询区域估计方法以减少算法的能量消耗。利用网格对节点进行管理,将查询区域中的网格划分成多个网格区,由各个网格区并行处理查询从而减少延迟。另外,GKNN 利用节点冗余降低了节点失效对查询结果的影响,提高了查询结果的正确性。仿真实验结果表明,GKNN 优于现有的算法。

关键词 无线传感器网络,查询处理,K 近邻查询,节点失效

中图法分类号 TP393 **文献标识码** A

Grid-based K Nearest Neighbor Query Processing Algorithm in Wireless Sensor Networks

LIU Yu-lei QIN Xiao-lin SHEN Jia-jia

(College of Information Science & Technology, Nanjing University of Aeronautics and Astronautics, Nanjing 210016, China)

Abstract A grid-based KNN query processing algorithm called GKNN was proposed in this paper which takes energy consumption, query latency, query result correctness and etc into consideration in an integrated way. It optimizes the existing query area estimation methods in order to reduce the energy consumption of the algorithm. GKNN takes advantage of grids to manage the nodes and divides the query region into several grid zones. Each grid zone processes query parallel to reduce query latency. Furthermore, GKNN takes advantage of node redundancy to reduce the influence of node failures on query result correctness which improves the accuracy of query result. Experimental results show that GKNN outperforms the existing algorithms.

Keywords Wireless sensor network, Query processing, KNN query, Node failures

1 引言

无线传感器网络综合了传感器技术、嵌入式技术、分布式信息技术和无线通信技术。通过发送一个查询,可以使用户在任何时间、地点和任何环境条件下获取大量详实而可靠的信息。通常情况下,用户感兴趣的是由传感器网络监测得到的与时间、空间相关的事件信息,而不是与传感器本身无关的大量观察数据,因此,时空查询成为传感器网络技术中的一个重要的研究内容。在各种时空查询中,K 近邻查询是一个重要而典型的查询。K 近邻查询可以形式化为:给定一个包含有 N 个节点的传感器网络 M 和一个已知的物理地址位置 p (查询点),寻找一个含有 K 个节点的子集 M' ,其中 $M' \subseteq M$, $K \leq N$,并且 $\forall n_1 \in M'$ 和 $\forall n_2 \in M - M'$,满足 $dist(n_1, q) \leq dist(n_2, q)$ 。其语义是:在整个传感器网络环境中,查找距离某个指定位置最近的 K 个传感器节点。

现有的传感器网络环境下的 K 近邻查询处理算法主要分为 3 类:(1)集中式算法。它将所有传感器节点的位置和感知的数据收集至基站并存储于集中式的时空数据库中,利用空间索引技术^[1]减少磁盘 I/O 的访问次数,以提高处理查询的速度^[2]。考虑到传感器节点的能量由电池供应且通常无法

更换,能量十分有限,集中式算法势必造成大量的能量消耗,因而不适用于传感器网络。(2)基于树的算法。文献[3,4]利用节点的位置信息将传感器网络组织成一棵分布式的 R 树^[1],节点维护以其为根的子树中的所有节点的 MBR(最小边界矩形)。节点收到查询消息后,仅将该消息发送到可能包含查询结果的子结点,这样无需访问不包含查询结果的子树,减少了能量的消耗。KPT^[5]算法首先利用位置路由协议^[6]将查询消息发送至距离查询点最近的节点(主节点)。到达主节点后,利用位置路由由查询消息过程中收集的节点信息估计出包含 K 个最近邻节点的区域边界大小,然后将估计区域中的节点组织成树形拓扑结构,并在该区域中查找距查询点最近的 K 个节点。KPT 算法通过对查询区域进行估计,减少了查询处理过程中被处理节点的个数,降低了能量消耗。(3)基于路线的算法。在 KPT 算法中,如果不知道当前网络的拓扑信息,则无法估算出准确的 KNN 查询区域大小从而影响整个查询的能量消耗及查询的准确性。另外,若查询的发起节点距离查询点很近,则估算查询区域的算法将失效,无法估算出准确的查询区域。针对上述问题,IKNN^[7]采用基于路线的方法^[8],根据阿基米德螺旋线向外传播进行查询。查询过程中,节点被查询访问的次序可近似地认为是节点距离查询点

到稿日期:2010-06-24 返修日期:2010-11-12 本文受国家自然科学基金资助项目(60673127),国家高技术研究发展计划(“863”计划)基金资助项目(2007AA01Z404),江苏省科技支撑计划(BE2008135),工信部电子信息产业发展基金项目资助。

刘宇雷(1974—),男,博士生,主要研究方向为传感器网络数据库,E-mail:liu_yulei@nuaa.edu.cn;秦小麟(1953—),男,教授,博士生导师,主要研究方向为传感器网络数据库、空间信息处理与GIS、数据库安全技术;沈佳佳(1974—),男,硕士,主要研究方向为传感器网络数据库。

远近的顺序。因此,当查询访问到第 K 个节点时,查询结束。相比于 KPT 算法,其能量消耗明显较少。但是当 K 值很大时,查询路线将会变得很长,增加了查询的延迟。解决办法是设计两条或者更多的路线同时开始查询处理^[9-12]。当多条路线同时进行查询时,为保证查询到第 K 个节点时查询能停止,设计的路线应该周期性相遇,彼此互换信息以确定当前已经查询的节点数目,更新查询停止条件。多路线并行查询的方法,有效减少了查询延迟,但也增加了节点间通讯碰撞的可能性。

现有的 K 近邻查询处理算法没有综合考虑能量消耗、查询延迟、查询结果正确性等因素,无法满足应用需求。本文提出的基于网格的 KNN 查询算法 GKNN(Grid-based KNN)权衡了这三方面因素。它利用网格对节点进行管理并优化了现有的查询区域估计方法,以减少能量的消耗。GKNN 把查询区域中的网格划分成多个网格区并将查询发送给各个网格区,由各个网格区并行处理查询从而减少延迟。另外,GKNN 利用节点冗余降低了节点失效对查询结果的影响,提高了查询结果的正确性。仿真实验结果表明,在绝大多数情况下,GKNN 优于现有的算法。

2 预备知识

KPT^[5]算法和基于路线的算法均需估计包含离查询点 q 最近的 K 个传感器节点的查询区域边界大小。假设离查询点 q 最近的点为主节点。查询从出发点开始,采用 GPSR 路由算法发送至主节点,由主节点计算查询区域大小。主节点估计查询区域大小的算法主要有:

(1) SUMDIST^[5]:在 GPSR 路由查询至主节点过程中,将处于此路线上的每一个节点的位置信息按顺序添加至查询消息中。当主节点接收到查询信息后,提取查询信息中倒数第 K 个位置的信息,将此节点距离查询点 q 的距离作为查询区域的半径。

(2) MHD^[5]:SUMDIST 算法中传送整个路径信息表带来了很大的能量消耗,因此 MHD 算法中只传送路径上任意两跳之间的最大距离值(maximum hop distance)。即将变量 MHD 的值与查询一起传送到路径上的下一个节点,接收节点重新计算与上个节点之间的距离,并与收到的 MHD 值比较,若大于,则更新 MHD 值,否则继续向前发送直到到达主节点。假设主 node 有 m 个邻居节点并且 $m < K$,则主节点计算的查询区域的半径大小为 $(K-m) * MHD$ 。相比于 SUMDIST,此算法有效地降低了传输数据量的大小,减少了能量的消耗,但同时也带来一个问题,即由此计算而来的半径很有可能比 SUMDIST 中计算的半径大很多,导致在下一步在区域中寻找 KNN 时消耗较多的能量。

(3) NeighborClass^[7]:一个节点含有 m 个邻居节点,取其中距离查询点 q 最近的 $\text{Min}\{m, K\}$ 个节点,存入当前节点的 NeighborClass 中。NeighborClass 的距离是表中离查询点最近的节点的距离。路线上的每一个节点都将产生一个 NeighborClass。当一个节点收到查询信息及前面节点计算得来的 NeighborClass 后,当前节点寻找其最近的 $\text{Min}\{m, K\}$ 个且不在之前的所有 NeighborClass 中的邻居节点作为当前节点的 NeighborClass。最后将结果一同传给下一个相关节点。当主节点收到查询后,计算自身的 NeighborClass,开始向前遍历收到的 NeighborClass,并相加各个 NeighborClass 中的

节点数目,当节点数大于 K 时取当前遍历到的 NeighborClass 的距离作为查询区域的半径。

(4) NeighborClass2^[7]:NeighborClass 算法相比于前两种算法,其半径的计算精确了很多。但同时也带来了较大的能量消耗,即路径上的每个点都要传输已经收到的 NeighborClass 表及当前节点自身的 NeighborClass。而 NeighborClass2 算法中每个路径上的节点只要传送节点数目和当前 NeighborClass 的最大距离即可。最后主节点通过对接收到的信息按 NeighborClass 的距离排序,从小到大累加 NeighborClass 的节点数目,达到 K 时停止,选择当前的 NeighborClass 的距离作为查询区域的半径。此算法存在的一个缺点是路径上的节点计算出的 NeighborClass 只是简单记录了数目,下一个相关节点无法根据之前的信息准确计算自己的 NeighborClass,最后计算的是其邻居节点的数目,因此所有的 NeighborClass 中存在很多重复,有可能导致最后计算的查询区域无法包含 K 个节点数目。

3 基于网格的 K 近邻查询算法 GKNN

本文提出的基于网格的 KNN 查询算法 GKNN(Grid-based KNN)权衡了能量消耗、查询延迟以及结果正确性等因素。它分为 3 个阶段:

(1) 将查询消息发送至离查询点最近的传感器节点(主节点)上,同时计算查询区域大小;

(2) 计算网格覆盖的查询区域,把网格划分为网格区,将查询由主节点分别发送到各个网格区中,由各网格区并行处理查询;

(3) 返回查询结果。

算法第一阶段类似于 KPT,不同的是对查询区域边界的大小进行优化,计算出尽可能小且包含最近 K 个节点的区域,避免不包含查询结果的节点参与查询,以减少能量消耗。在算法第二阶段中,由于整个查询处理过程转交由各个网格区自行完成,因此提高了算法的并行性,减小了查询延迟。另外,采用网格管理网络中的节点,使算法具有较好的鲁棒性,保障了查询结果的准确性。

3.1 传感器节点组织结构

假设传感器节点的通讯半径为 R ,将区域划分为一系列网格,其边长为 L 。如图 1 所示,节点可以和网格内的任何其他节点直接通讯,且相邻网格中的任意两个节点可以直接通信,因此,网格边长 L 满足 $L^2 + (2L)^2 = R^2$,即 $L = R/\sqrt{5}$ 。每个网格有其唯一 ID 编号,编号按网格位置顺序分配。每个传感器节点根据自身位置可计算出具体位于哪个网格之中。由于初始划分的网格比较小,查询处理过程中,根据需要可将多个网格合并成一个网格区,查询由各网格区处理,3.3 节将详细介绍。

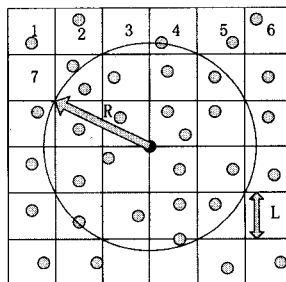


图1 网格设计

3.2 查询区域边界计算

考虑到查询消息路由到主节点后, GPSR 协议会进入 perimeter 模式, 即消息绕主节点传递一周, 以确定当前节点是否最靠近查询节点。确认后, 由主节点计算查询区域大小。perimeter 模式中访问到的节点形成一个周界, 环绕主节点。利用上述特性, 提出一种基于 NeighborClass 的改进的查询区域计算方法 NeighborClass3。即当 GPSR 将查询及路线上的 NeighborClass 发送到主节点后, GPSR 协议进入 perimeter 模式, 并将所有 NeighborClass 信息继续向前发送并计算周界点的 NeighborClass 值。当绕回到主节点时, 确定当前节点是最节点后, 根据所有 NeighborClass 计算查询区域半径。当 K 值较大时, NeighborClass3 对查询区域半径的估计明显优于其他几个算法。如图 2 所示, 内圆包含的区域为 NeighborClass3 计算的查询区域, 外圆区域为 NeighborClass 计算的查询区域大小。当 K 值增大时, NeighborClass3 减小了估计边界的大小, 因而减少了查询过程中被访问的节点数量, 降低了能量消耗。3.5 小节将分析 K 值大小对边界半径大小的影响。

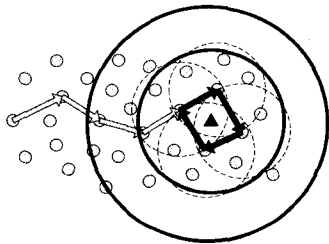


图 2 查询区域大小(其中 K=12)

3.3 查询处理过程

查询围绕查询点完成周界传递, 同时计算出查询区域大小并将当前结果返回给主节点。简单的做法是直接查询区域内进行泛洪查询, 由于新算法的区域边界计算更准确, 显然其查询效率将好于 KPT 算法。直接泛洪法的优点是查询快, 每个节点执行查询后直接将结果返回给主节点, 无需等待其他节点的处理情况, 然而, 对于具有聚集操作的查询采用泛洪法明显效率不高, 数据之间缺少必要的聚集操作, 造成了一些无谓的能量消耗。

针对聚集查询, 采用树形拓扑结构是一种常用的手段。主节点根据查询区域大小, 计算查询区域覆盖的网格的相关信息, 如网格的 ID、位置等。通常情况下划分的网格比较小, 目的是为了减小覆盖查询区域的冗余, 查询过程中主节点根据实际情况如节点密度、区域大小等信息将相邻的几个网格合并成一个大的网格区, 如图 3 中将查询覆盖区域划分成 4 个网格区。最后主节点将查询通过 GPSR 分别发送到每个网格区中, 由每个网格区单独处理各自的查询。

如图 3 所示, o 为查询点; s 为主节点, 圆形区域为主节点计算出的查询区域大小; 包含圆形区域的矩形为查询覆盖区域, 其内部的虚线将整个查询覆盖区域划分成 4 个网格区。主节点将查询、当前区域半径大小和目的网格区的信息通过 GPSR 协议发送到对应网格区之中, 网格区中最先收到查询的节点标记为 master, master 节点以自身为根在本网格区中建树, 负责处理本网格区内的查询, 收集网格区内其他节点的数据, 进行相应处理后将结果返回给主节点。网格区中其他节点收到查询后, 判断自身是否位于当前网格区及查询区域

范围内, 如果满足, 则将结果返回给 master。

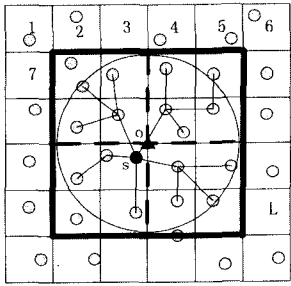


图 3 查询处理过程

算法将查询分别交给位于查询区域内的各个网格区, 由网格区组织查询处理过程, 并将结果返回。查询覆盖区域划分的网格区越多, 生成的查询子树越多, 从而加大了查询的并行性, 相比于 KPT 及二路 IKNN 算法, 其具有较小的查询延迟。同样地, 划分的子树越多, 带来的问题是子树与子树之间的数据缺少必要的操作, 降低了聚集查询效率。

3.4 返回查询结果

当前查询不包含聚集操作, 则网格区 master 收到网格中其他节点的查询结果后, master 直接将结果返回给主节点。主节点收到所有网格区的数据后提取满足要求的 K 个查询结果发送给查询发起者。同样地, 对于每个网格区中的节点也可单独将结果返回给主节点, 相当于在整个区域中进行泛洪处理; 若查询包含聚集操作, 则网格内节点首先将结果返回给 master, 每个 master 将结果进行聚集操作, 最后将每个 master 的结果返回给主节点, 主节点进行聚集操作后将最终结果返回。

3.5 性能分析

对本节算法 GKNN 的性能进行理论分析, 包括查询区域边界半径的估计、算法的能量消耗、查询的延迟及鲁棒性等。GKNN 算法是基于 KPT 算法的改进算法, 因而主要分析比较 GKNN 与 KPT 算法, 用到的变量定义如表 1 所列。

表 1 传感器网络变量定义

定义	变量
传感器节点总数目	n
传感器通信半径	r
节点平均邻居数	n_r
查询起点到主节点的跳数	h
环绕查询位置周界上的节点数	n_p

实验分析中将给出 GKNN 与 IKNN 算法性能之间的比较。

(1) 区域边界半径: 现有的估算查询边界算法中 SUM-DIST 和 MHD 只考虑查询路由过程中路线上的节点, 由此估算查询边界大小, 就会产生较大的误差。而 NeighborClass2 算法只发送路线中邻居点数目, 较多的节点将被重复计算, 在一个节点分布不均匀的环境中, 算法的效率很差, 可能导致计算的查询区域实际包含的节点个数少于 K 值。这里将主要分析算法 NeighborClass 及本文中改进后的 NeighborClass3。为简化计算复杂度, 考虑图 4 中的情况, 计算查询路线上的一个节点的 NeighborClass 集时, 当前节点的邻居节点大约有一半的节点也属于路线上前一个节点的邻居, 即可以假设除起始节点外, 每个节点的 NeighborClass 集中包含有其所有邻居节点数一半的节点 $n_r/2$ 。从查询起始点到主节点, 经过 h 跳,

算法 NeighborClass 计算得到路线上总共有 $n_r \times (h+2)/2$ 个节点,而算法 NeighborClass3 得到 $n_r \times (h+2+n_p)/2$ 个节点。因此,对于 K 近邻查询,算法 NeighborClass 计算的边界半径大小约为 $\frac{K-n_r/2}{n_r/2} \times r+r$,类似地,算法 NeighborClass3 计算的边界半径约为 $\frac{K-n_r \times n_p/2}{n_r/2} \times r+r$ 。两者的半径差约为 $(n_p-1) \times r$ 。显然,当 K 值较小时,两者计算的结果基本一致;当 K 值增大时,算法 NeighborClass 计算结果开始增长,而算法 NeighborClass3 将出现一个稳定的半径大小;当 K 值继续增大时,两个算法计算的半径结果都将变大,两者间的差值基本是一个常数。可见改进后的算法 NeighborClass3 相比其他边界计算算法更准确,计算的查询边界区域半径更小。边界半径的大小直接影响算法的执行效率,越小的半径包含越少的节点,也即通信消耗的能量及数据传送的延迟将更小。

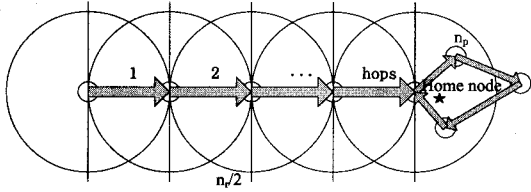


图4 查询路由过程

(2) 能量消耗:算法总的通信及数据传输的能量大小与执行过程中参与查询的节点的个数有关。GKNN 算法第一阶段中计算的查询区域边界半径较 KPT 算法更优,因此,其参与查询的总节点个数少于 KPT 算法,相应的能量消耗也少。当 K 值较小时,由于两个算法计算的区域边界半径大小基本一致,因此能量消耗基本相当;当 K 值增大时,KPT 算法半径开始增长,而 GKNN 半径相对稳定,因此 KPT 算法的能量消耗开始增大,而 GKNN 算法的能量消耗会出现一个稳定阶段;当 K 值继续增大时,两个算法计算的半径都变大,且两者间的差值基本是一个常数,然而随着半径的增大,查询区域的面积差值增大,即 KPT 算法中参与查询的节点数目的增量大于 GKNN,因此 KPT 能量消耗的增幅将大于 GKNN。节点密度较小时,KPT 计算查询区域边界的半径大于 GKNN 算法,相应的包含的节点个数较多,查询消耗能量也相对较大,而当节点密度增大时,两者的计算半径都开始减小,同时 KPT 接近于 GKNN 算法,因此两个算法的能量消耗相当。

(3) 查询延迟:算法 KPT 与算法 GKNN 都依赖于树形结构,因此查询算法延迟时间与树的高度及树中节点个数有关。KPT 算法采用周界树的思想,即每个周界节点是一个子树树根,因此查询区域中,子树的数目与周界节点个数 n_p 有关。GKNN 算法根据查询覆盖区域来划分网格区,每个网格区各自建树并进行查询处理,因此 GKNN 中子树的个数与网格区的多少有关。子树个数越多,算法执行的并行程度越大,每颗子树的节点数越少,树的高度越低,因此查询延迟越少,同时也带来一个问题即若查询包含聚集操作,则在子树与子树之间缺少对数据进行必要的聚集操作,导致多一份的数据传输,消耗相对较多的能量。GKNN 算法中网格区的设置可根据当前计算的区域边界半径大小动态调整,权衡查询的延迟与能量的消耗,使得查询在满足实时要求的同时具有较少的能量损耗。

(4) 查询准确性:KPT 算法采用在查询区域内建树的方

式寻找最近 K 个节点数据,同时完成数据聚集。算法过分依赖这种树形的下层结构,若网络中部分节点无法正常工作,造成节点之间通信链路频繁失效,将导致以这些节点为根的子树数据全部丢失,严重影响查询的准确性。改进后的算法利用围绕查询位置的周界节点,将整个查询区域划分为若干部分,分别以这些周界节点为树根建树,进行查询处理。算法在一定程度上降低了树的高度和树中节点的个数,然而当 K 值增大时,查询区域变大,而环绕查询位置的周界节点个数基本不变,相应增加子树高度及树中节点个数,当有节点失效时同样对查询准确率带来较大影响。IKNN 算法采用基于路线的方法,查询按预先设定的路线向前传播。由于具体查询路线在查询过程中形成,因此能有效消除通信链路失效等问题带来的影响。然而,基于路线的方法,不能保证路线的联通性,在一个节点分布不均匀的网络中,查询可能因找不到路线上的下一节点而中止,影响结果的正确性。GKNN 算法将覆盖查询区域的网格划分成多个网格区,每个网格区各自建树并进行查询处理。由于每个网格区较小,其中的节点个数较少。对于查询区域中的绝大多数节点,GKNN 算法中以该节点为根的树中的节点数目通常小于 KPT 算法,因而当节点失效时,GKNN 丢失的数据较少,具有更好的鲁棒性。

可见,GKNN 算法相对于 KPT 各方面都具有一定优势,算法最差情况下效率基本与 KPT 一致。在下一节的实验中 will 看到,GKNN 在大多数情况下其执行效率也优于 IKNN 算法。

4 实验

在文献[13]中的模拟器上实现上述的几个算法,分析比较算法的执行效率,主要考察算法的能量消耗、查询的延迟及查询结果的准确性。

缺省情况下传感器网络的实验环境配置如表 2 所列。假设传感器节点均匀分布在整个传感器区域中,每个节点平均含有 30 个邻居节点,查询 K 取值范围从 10 到 200。同时假定节点固定且不失效,MessageDelay 值为 30ms。网格边长 L 为 17 米,平均一个网格内含有 2 个节点。为简化模拟实验,实验中网格区采用统一划分方法,即类似于图 3,划分相邻 4 个网格为一个网格区,即平均一个网格区包含有 8 个节点。

表2 KNN 查询网络缺省配置

参数	初始值
传感器网络覆盖面积	500x500 米
传感器通信半径	40 米
整个网络传感器节点数目	1500 个
K	50

对于不包含聚集操作的查询,由于 KPT 算法计算的查询区域边界大于 GKNN 算法,其能量消耗明显较大。而在 IKNN 算法中,路线上的节点需发送当前所有的数据信息,造成单节点的能量消耗偏大,影响整个网络的生命周期,因此算法不适用于不包含聚集操作的查询。实验主要考察带聚集操作的几个 KNN 查询算法执行效率,包括 KPT SR,IKNN 和 GRT。

实验中考察的数据的计算:

能量消耗 (Transmission energy consumption) (J): 算法

执行过程中总的通信能量消耗。

查询延迟(Query latency)(ms):从查询发起收到查询结果的平均时间。

查询的准确性(Query accuracy)(%):离查询最近的 K 个节点中被访问的节点百分比。

4.1 查询区域边界估计

查询区域边界的估计直接影响 KPT 算法及网格下 KNN 算法的能量消耗。一个最优边界是只包含 K 个节点的区域,实验中将被用来作为参照对象。

图 5 与图 6 显示了 K 值大小对不同算法估算区域边界半径产生的影响。其中 Y 轴表示边界半径大小, X 轴表示 K 的大小。图 5 显示了当 K 值很小(小于 10)时, SUMDIST 和 MHD 计算的边界半径最接近最优解, NeighborClass3 算法计算的半径结果优于 NeighborClass 算法;而当 K 值增大时,查询发起点离查询点不足 K 跳路由,导致 SUMDIST 和 MHD 无法获得足够信息,计算的值显著增大,如图 6 所示。其中, NeighborClass3 算法计算的边界半径最接近最优解。

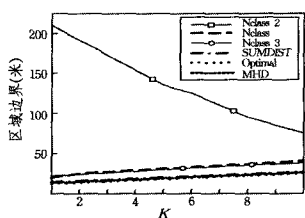


图 5 边界半径($K < 10$)

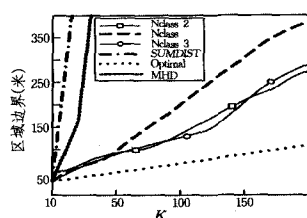


图 6 边界半径($K > 10$)

4.2 传感器网络节点密度对算法的影响

本小节通过调整网络中分布的节点数目来考察网络节点密度对 KNN 查询算法的影响。节点数目从 500 调整到 1500,即其中每个节点的平均邻居数从 10 到 30。

图 7 显示了节点密度大小对算法能量消耗的影响。可见,随着节点密度增大, GRT^[4], KPT 及 GKNN 的能量消耗均变大。其中 GKNN 的能量消耗略优于 KPT 算法。IKNN 算法的能量消耗基本不受节点密度大小的影响。这主要是因为当节点密度增大时,位于 MBR 及查询区域边界范围内的节点数增多,查询过程中参与查询的节点数增加,导致查询总能量消耗变大,而 IKNN 查询根据预设路径进行访问,当查询到第 K 个节点后查询结束,因此总消耗能量基本不受网络节点密度影响。

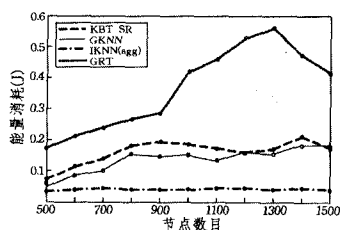


图 7 节点密度-能量消耗

图 8 反映了节点密度大小对查询结果的准确性的影响。可以看到,节点密度的大小对各个算法都有一定影响,其中对 IKNN 算法影响最大。当网络内节点密度增大,查询路由的失败的可能性将减小。IKNN 算法中 Q-node 在查询选路过程中,必须按照其预设路线选择下一个合适的节点,当节点密度较小时,很容易发生查询路由失败的情况,而 KPT 和

GKNN 等没有查询路由限制,相对影响较小。正常情况下, IKNN 的结果准确性接近于 KPT, GKNN 算法。

图 9 显示了节点密度大小对查询延迟的影响。IKNN, KPT, GKNN 算法基本不受节点密度的影响。其中 IKNN 算法的延迟时间最大,而 GKNN 的延迟最小。这主要是由于本实验中 IKNN 算法采用了 2 路并行查询方式,而 KPT 算法则按树状方式查询,相当于多路查询, GKNN 相当于多颗树进行并行查询,因此延迟时间最小。

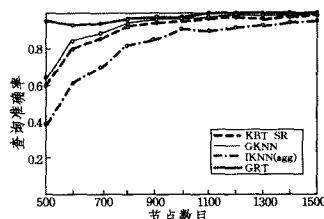


图 8 节点密度-查询结果的准确性

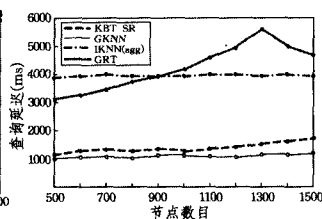


图 9 节点密度-查询延迟

从实验中看出,节点密度的大小对算法 KPT, GRT 和 GKNN 的能量消耗影响较大,因此,当用户查询没有实时要求且查询带聚集操作时,算法 IKNN 具有较高的效率;其他情况下,尤其是处理实时查询时,本文的算法 GKNN 优于其他几个 K 近邻算法。另外,不同于 IKNN 算法对网络环境具有较强依赖, GKNN 算法具有较好的适应性,对于网络中节点分布不均匀等情况,算法同样能进行有效的查询处理,因此在大多数情况下,算法 GKNN 具有较好的查询执行效率。

4.3 查询 K 值对算法的影响

本节将考察查询 K 值对算法执行效率的影响。实验中 K 的取值范围从 50 到 400。

图 10 中,当 K 取值不大,即不大于 250 时, IKNN 的能量消耗小于其他几个算法。不包含聚集操作的 IKNN 算法在沿路线进行查询的过程中,每次都发送之前所有的查询结果,明显地,当 K 值很大并当查询到达路线的后半部分时,每次向前传播将消耗大量能量。因此当 K 值大于 250 时, IKNN 算法的能量消耗大于其他算法。GRT 算法中,计算和传输节点的 KTable 依赖于 K 与当前节点的邻居节点数目 m 的值,邻居节点数目与当前网络中节点密度有关,因此当 K 大于 m 时,传送 KTable 消耗能量稳定,即 GRT 算法在 K 较大时,其能量消耗基本不受 K 值影响。GKNN 算法由于其查询边界计算优于 KPT,因此能量消耗小于 KPT。

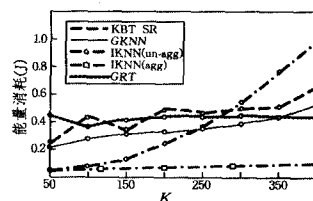


图 10 K -能量消耗

图 11 显示了 K 值大小对查询准确性的影响。图中可以看出 K 值对各个算法的查询结果准确性影响较小,当 K 值增大时,算法的准确性有下降趋势。

图 12 显示了 K 值大小对算法的延迟时间的影响。实验中只考察 KPT, GKNN 和 IKNN 算法的查询时间延迟。

IKNN 算法的时间延迟直接跟 K 值有关,呈线性增长趋势, K 值较大时,其时间延迟较大。KPT 算法中,随着 K 值的增大,算法计算的查询边界半径大小呈线性增长,查询区域面积大小将出现指数级增长,也即查询区域中树的高度及节点个数呈指数增长,因此 KPT 算法的查询延迟随着 K 值增大增长较快。GKNN 算法的查询延迟时间与查询区域中的网格区的数量有关,两者为线性关系。因此,当 K 值增大时,网格区数量增多,延迟时间增大,其增长量小于 KPT 算法。

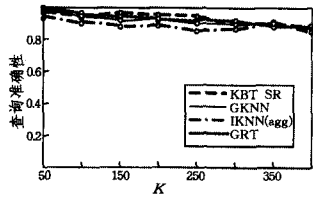


图 11 K -查询结果准确率

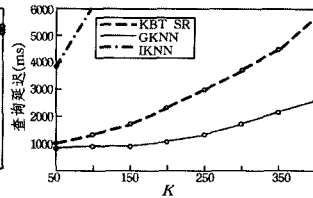


图 12 K -查询时间延迟

K 值大小对各个算法的能量消耗及查询时间延迟具有较大的影响。当查询包含聚集操作且没有实时性要求时,IKNN 算法优于其他几个算法;其他情况下,尤其当处理实时查询时,GKNN 具有较大优势,同时其能量消耗相对较少,查询准确率较高。

结束语 本文提出了一种基于网格的 KNN 查询算法 GKNN。算法优化了现有的查询区域边界计算方法,使得计算出的边界尽可能小,同时包含距离查询点最近的 K 个节点,减少了能量消耗。节点采用基于网格的方式进行管理,算法把查询区域划分成多个网格区,将查询发送给各个网格区,由各个网格区并行执行最后的查询及结果的返回,增加了查询处理的并行性从而减少了延迟;算法划分多个小子树,降低了节点失效对查询结果的影响,提高了查询结果的正确性。最后通过实验验证 GKNN 算法较其他几个算法具有最少的时间延迟,且具有较少的能量消耗和较高的查询准确率。

由于传感器网络的动态性,在实际应用中 GRT 算法较少被采用。若查询不包含聚集操作,相对于其他 KNN 查询算法,无论从能量消耗及查询延迟等方面考虑,GKNN 具有较好的查询性能和效率;若查询包含聚集操作,当 K 值较小时,IKNN 相对更有效,随着 K 值变大时,IKNN 算法的查询延迟时间增大,算法的选择应根据当前网络状况和实际需求如能量消耗或实时性等方面决定,权衡能量消耗和查询延迟,GKNN 更优。同时 IKNN 算法由于基于预先设定的路线进行查询,要求网络中的节点分布均匀,因此不适合于动态变化网络环境。而 GKNN 受节点分布影响较小,在不均匀分布的网络中同样具有较好的查询效率。因此在多数情况下,GKNN 算法优于其他算法。

参考文献

- [1] Guttman A. R-Trees: A Dynamic Index Structure for Spatial Searching [C]//Proceeding of the 1984 ACM SIGMOD international conference on Management of data. New York: ACM Press, 1984: 47-57
- [2] Roussopoulos A, Kelley S, Vincent F. Nearest Neighbor Queries [C]//Proceeding of the 1984 ACM SIGMOD international conference on Management of data. New York: ACM Press, 1995: 71-79
- [3] Demirbas M, Ferhatosmanoglu H. Peer-to-Peer Spatial Queries in Sensor Networks [C]//Proceeding of the IEEE International Conference on Peer-to-Peer Computing Peer-to-Peer Computing. Washington: IEEE Computer Society, 2003: 32-39
- [4] Goldin D, Song M, Kutlu A, et al. Georouting and delta-gathering: Efficient data propagation techniques for geosensor networks [C]//Proceeding of GeoSensor Networks Workshop. Portland, Maine, 2003
- [5] Winter J, Lee W K. A dynamic knn query processing algorithm for location-aware sensor networks [C]//Proceeding of the 1st International Workshop on Data Management for Sensor Networks in Conjunction with VLDB 2004. New York: ACM Press, 2004: 119-124
- [6] Karp B, Kung H T. GPSR: greedy perimeter stateless routing for wireless networks [C]//Proceeding of the 6th annual international conference on mobile computing and networking. New York: ACM Press, 2000: 243-254
- [7] Xu Y, Fu T Y, Lee W C, et al. Itinerary-based Techniques for Processing K Nearest Neighbor Queries in Location-aware Sensor Networks [J]. Signal Processing, 2007, 87(12): 2861-2881
- [8] Xu Yingqi, Lee Wang-chien, Xu Jianliang, et al. Processing Window Queries in Wireless Sensor Networks [C]//Proceeding of the 22rd International Conference on Data Engineering. Washington: IEEE Computer Society, 2006: 70-70
- [9] Yao Yuxia, Tang Xueyan, Lim Ee-peng. Localized monitoring of kNN queries in wireless sensor networks [J]. Journal of Very Large Database, 2009, 18(1): 99-117
- [10] Wu Shan-hung, Chuang Kun-ta, Chen Chung-mi, et al. DIKNN: An Itinerary-based KNN Query Processing Algorithm for Mobile Sensor Networks [C]//Proceeding of the 23rd International Conference on Data Engineering. Washington: IEEE Computer Society, 2007: 456-465
- [11] Wu Shan-hung, Chuang Kun-ta, Chen Chung-min, et al. Toward the Optimal Itinerary-Based KNN Query Processing in Mobile Sensor Networks [J]. IEEE Transactions on Knowledge and Data Engineering, 2008, 20(12): 1655-1668
- [12] Fu Tao-young, Peng Wen-chih, Lee Wang-chien. Optimizing parallel itineraries for knn query processing in wireless sensor networks [C]//Proceeding of the sixteenth ACM conference on Conference on information and knowledge management. New York: ACM Press, 2007: 391-400
- [13] Coman A, Nascimento A M, Sander J. A framework for spatiotemporal query processing over wireless sensor networks [C]//Proceeding of the 1st Int'l Workshop on Data Management for Sensor Networks in Conjunction with VLDB 2004. Washington: IEEE Computer Society, 2004: 104-110