交通数据流的聚集查询方法及系统

技术领域

本发明涉及交通数据流的聚集查询方法及系统，属于信息技术处理领域。

背景技术

随着物联网、社交网络和云计算技术等的蓬勃发展，大量的业务应用产生了海量数据，并且数据流量呈指数级别快速增长。对于这些海量数据的分析和挖掘，以期发现其中蕴含的各种规律和信息，已成为大数据处理技术领域的研究重点。数据分为静态数据和动态数据。动态数据又称为数据流，具有实时性、无限性、瞬时性、流速不定性以及元数据无穷性等特点。交通数据流就是一种典型的动态数据流。例如，为了分析和控制交通流、缓解交通压力，交通监控系统经常关注特定时段内特定路段上移动车辆的近似概要信息，例如，监控和分析南京市新街口上下班高峰期的车流量。

传统的数据查询方法，例如，轮廓查询方法，无法适用于动态数据的变化，对于数据流的查询使无效的。因此，在数据查询技术领域，又发展出一种聚集查询方法，通过扫描大量数据元组获取统计和概要信息。但是，由于交通数据流具有实时性、无限性、瞬时性、流速不定性以及元数据无穷性等特点，即使拥有并行计算能力的云计算技术也难以对整个数据集进行聚集查询，以在较短时间内获取精确的查询结果。因此，在实际应用中，往往利用高质量的近似聚集查询结果代替精确的查询结果。面对业界对查询精度要求的逐步提高，发展出多种近似聚集查询方法，例如，滑动窗口技术、随机采样技术、小波技术、草图索引结构、直方图技术等。然而，现有的近似聚集查询方法均以平均查询误差的大小去衡量算法的优劣，使得现有近似聚集查询方法的精度无法有效替代精确查询。

因此，针对现有近似聚集查询方法无法有效替代传统精确查询方法的技术问题，有必要对现有聚集查询方法进行改进，并提出一种新的聚类查询方法，以提高聚类查询的精度，从而有效替代传统精确查询方法。

发明内容

针对现有技术的上述缺陷，本发明的目的是要提供一种改进的聚集查询方法及系统，采用运用卡尔曼滤波器原理通过校正状态先验估计以获得后验估计的方法，利用桶的频率最优估计计算异常点聚集值，提出了本发明所要解决的技术问题是针对现有近似聚集查询技术忽略了能够产生最大相对误差的不足。

本发明提供一种交通数据流的聚集查询方法，包括如下步骤：

步骤 1，采集移动对象信息，将移动对象信息转化为计算机可处理的数据形式，在系统时间戳到来时形成数据流；

步骤 2，在系统时间戳到来时生成、更新索引文件；

步骤 3，在生成更新索引文件的同时，根据用户的查询请求 $SUM\left(r,ts,te\right)$ 对空间区域 r 进行交通数据流查询，提取查询时间区间 $\left[ts,te\right]$ 的系统时间戳，对于每个时间戳执行空间聚集查询：遍历当前时间戳 t 的索引文件，利用如下表达式求得当前时间戳下空间区域 r 在各桶的聚集查询值：

$$SUM=\begin{cases} f_{bi}\bullet S_{intr}\bullet\omega^2 & ,v_{bi}\le\left(f_{bi}-f_{bi}^{-}\right)^2 \\ \left[f_{bi}^{-}+Kg_{bi}(f_{bi}-f_{bi}^{-})\right]\bullet S_{intr}\bullet\omega^2 & ,v_{bi}>\left(f_{bi}-f_{bi}^{-}\right)^2 \end{cases},$$

SUM 为空间区域 r 在第 bi 个桶的聚集查询值，S_{intr} 为空间区域 r 与第 bi 个桶相交区域的面积，f_{bi} 为当前时间戳第 bi 个桶的平均频率，f_{bi}^{-} 为前一时间戳 t^{-} 第 bi 个桶的平均频率，v_{bi} 第 bi 个桶的方差，

将每个时间戳的查询值求和形成最终的聚集查询值。

进一步地，所述步骤 2 包括：

步骤 A，初始化第一个系统时间戳的数据流生成的索引文件：采用合理直方图将数据空间分割为 $\omega\bullet\omega$ 的单元，以当前时间戳内单元内的移动对象数量表示该单元的频率，再将频率相似的邻近单元组成一个桶，形成 n 个桶，$0<n\le B$，ω 为分辨率，B 为桶数目的上限 ，

对于每个桶：以桶中所有单元的平均频率作为该桶的频率，计算该桶中各单元平均频率方差的平均值、该桶的方差以及卡尔曼增益；

步骤 B，在下一系统时间戳到来时，利用卡尔曼滤波原理更新索引文件：

步骤 B-1，当第 c 单元中的数据变化时，记数据变化量为 d，更新第 c 单元的频率 F_c：$F_c^{-}=F_c, F_c=F_c+d$，其中：F_c^{-} 为前一时间戳单元 c 的频率，$1\le c\le\omega^2$，d 为任意实数；

步骤 B-2，遍历当前时间戳的索引文件找到包含数据量变化单元的桶，对于第 b 个桶，第 b 个桶包含有 n_b 个单元，$b<n$，$n_b<\omega\bullet\omega$：

更新第 b 个桶的频率 f_b：，$f_b = f_b^- + d, \Delta g = f_b^2 - f_b^{-2}$，

更新第 b 个桶中各单元频率平方的平均值 g_b、方差 v_b：$g_b = (n_b \cdot g_b^- + \Delta g) / n_b, v_b = g_b - f_b^2$，

更新第 b 个桶中第 i 单元的卡尔曼增益，F_i 为第 i 单元的频率：

当 $F_i > f_b$ 时，第 i 单元的卡尔曼增益为：$Kg_i^+ = \frac{f_b + \sqrt{v_b} - f_b^-}{f_b - f_b^-}$，

当 $F_i \le f_b$ 时，第 i 单元的卡尔曼增益为：$Kg_i^- = \frac{f_b - \sqrt{v_b} - f_b^-}{f_b - f_b^-}$，

计算出第 b 个桶中频率大于平均频率的单元数目 n_{b1}，更新第 b 个桶的卡尔曼增益 Kg_b：$Kg_b = n_{b1} \bullet Kg_i^+ + (n_b - n_{b1}) \bullet Kg_i^-$，$1 \le i \le n_{b1} \le n_b$；

步骤 B-3-A，对于需要分裂的桶，计算每个需要分裂桶的最高分割利益和最优划分位置，按照最优划分位置将需要分裂的桶分为两个子桶，并且设置前一系统时间戳内两个子桶的频率均与分裂前桶的频率相等，重复步骤 B-1；

步骤 B-3-B，对于不需要分裂的桶，在索引中桶的数量达到上限时，利用最小合并惩罚原理将频率集中的多个桶合并为一个桶，重复步骤 B-1；

步骤 B-3-C，对于不需要分裂的桶，在索引中桶的数量未达上限且当前系统时间戳尚未结束的情况下，返回步骤 1；

步骤 B-3-D，对于不需要分裂的桶，在索引中通的数量未达上限且当前时间戳结束的情况下，提取当前时间戳的索引文件生成历史索引；

进一步地，所述步骤 2 采用二叉划分树结构来索引桶。

进一步地，步骤 B-3-A 利用贪心算法计算每个需要分裂桶的最高分割利益和最优划分位置。

进一步地，步骤 2 中生成的历史索引序列化存储在索引存储模块中。

进一步地，步骤 1 中采集的移动对象信息包括编号、经纬度坐标。

本发明还提供一种交通数据流的聚集查询系统，包括：信息收集模块，采集移动对象的信息，将移动对象信息转化为计算机可处理的数据形式，在系统时间戳到来时形成数据流并发送给数据处理引擎；数据处理模块，将系统时间戳内的空间数据划分为子单元，把包含移动对象数量相近的单元组成一个桶，生成桶的

索引文件，计算桶的频率、各单元平均频率方差的平均值、方差以及卡尔曼增益，在新的系统时间戳到来时利用卡尔曼滤波原理更新索引文件；索引存储模块，用以存储索引生成和更新模块生成的索引文件；应用服务模块，调用索引存储模块查找符合查询请求的索引文件，并反馈聚值查询值。

进一步地，所述处理模块用于接收当前时间戳的索引，在当前时间戳结束后将当前时间戳的索引与历史索引序列化处理后输出至引存储模块。

本发明还提供一种基于卡尔曼滤波的交通流的索引创建方法，包括：步骤 A，信息收集模块获取移动对象信息并进行预处理；步骤 B，数据处理模块判断索引存储模块中是否有当前时间戳的 BPT 索引，若有转向步骤 D，否则转向步骤 C；步骤 C，创建一个当前时间戳的新 BPT 索引，其中：Kg 和 f^- 均为 0，转向步骤 D；步骤 D，基于卡尔曼滤波原理更新 BPT 索引；步骤 E，根据桶的方差判断 BPT 索引中的桶是否需要分裂，若其方差大于事先预设的阈值，则需要进行分裂，转向步骤 F，否则转向步骤 G；步骤 F，通过穷举的方式计算每个需要分裂桶 b 的每种分裂情况的分割利益 SB 和对应划分位置，记 b 分割前的 cell 数量为 n_b 和方差为 v_b，分隔成为 b1 和 b2，其中：对应的 cell 和方差分别为：n_{b1}、n_{b2} 和 v_{b1}、v_{b2}，则分隔利益 $SB = n_b \bullet v_b - \left(n_{b1} \bullet v_{b1} + n_{b2} \bullet v_{b2}\right)$，SB 值最高的记为最高分割利益（BSP），按照 BSP 将桶 b 分为两个子桶 b1 和 b2，并且设置 $f_{b1}^- = f_{b2}^- = f_b^-$ 分割后的初始值，返回步骤 D 更新 BPT 索引并计算相应的卡尔曼增益；步骤 G，判断索引中桶的数量是否达到系统设置的桶数量最大上限（B），若是转向步骤 H，否则转向步骤 I；步骤 H，对索引中桶进行合并操作，计算出所有可能合并桶的合并惩罚 MP，其中：该桶在 BPT 中必须是非叶子节点，并且其两个儿子节点都是叶子节点，选取合并惩罚最小的多个桶合并为一个桶；合并惩罚的计算规则与分隔利益类似，由合并后形成桶的 cell 数量乘以方差减去合并前两个桶的 cell 数量和方差积之和，则合并惩罚计算公式为：$MP = n_b \bullet v_b - \left(n_{b1} \bullet v_{b1} + n_{b2} \bullet v_{b2}\right)$，并利用步骤 D 的方法更新 BPT 索引；步骤 I，判断当前时间戳是否结束，若结束则转向步骤 J，否则转向步骤 A； 步骤 J，索引处理模块从索引存储模块中取出当前时间戳的 BPT 索引，生成历史索引，并将其序列化存储到索引存储模块。

进一步地，所述步骤 A 进一步包括：步骤 A-1：信息收集模块获取移动对象

的信号，并转换成为数据流形式；步骤 A-2：当达到一个系统时间戳时，信息收集模块将移动对象数据流发送给数据处理模块。

进一步地，所述步骤 D 进一步包括：步骤 D-1，对于每一个变更的单元 c，计算其变化量 d，更新其频率信息：$F_c^- = F_c, F_c = F_c + d$（$F_c^-$ 表示前一个时间戳的频率）；步骤 D-2，对于每一个变更的单元 c，遍历 BPT 索引找到包含该变化单元的桶 b，更新桶 b 的频率和卡尔曼增益：$f_b^- = f_b, f_b = f_b + d, \Delta g = f_b^2 - f_b^{-2}$，$g_b = (n_b \cdot g_b + \Delta g)/n_b, v_b = g_b - f_b^2$；步骤 D-3，找出桶中频率大于平均频率的单元的数量 n_{b1}，计算桶的卡尔曼增益：$Kg_b = n_{b1} \bullet Kg^+ + (n_b - n_{b1}) \bullet Kg^-$。

本发明还提供一种基于卡尔曼滤波的交通流聚集查询方法，包括：步骤 A，解析用户的查询请求 $Q(r, ts, te)$，提取查询时间区间 $[ts, te]$ 的系统时间戳集合 $\{t_1, t_2, \cdots, t_k \mid t_s \le t_1 < t_2 < \cdots < t_k \le t_e\}$，将集合中的这些时间戳按时间序置入栈 T; 其中：r 表示查询区域，ts 表示起始时间，te 表示终止时间；t1…tk 表示起始时间与终止时间之间的 1-K 个时间戳；步骤 B，设置聚集查询值 AR=0；步骤 C，对于 T 中的每一个时间戳，执行聚集查询，其中：针对单一时间戳的空间聚集查询，对于每一个与查询区域 r 相交的桶 b：首先，计算桶 b 查询区域 r 相交部分的面积 S_{intr}；然后，计算这个桶 b 的中间结果 $f_b \bullet S_{intr} \bullet \omega^2$；最后，将所有的中间结果累加以得到单一时间戳的空间聚集查询值；对于多个时间戳的查询结果是对单一时间戳的查询结果的累加；步骤 D，返回聚集查询值 AR，查询结束。

进一步地，所述步骤 C 包括：步骤 C-1，遍历 BPT 找到所有与查询区域 r 在空间上相交的桶 b，计算 r 与桶 b 的相交面积 S_{intr}，并将这些桶以<b, S_{intr}>的形式置入栈 S；步骤 C-2，对于 S 中的每一个桶 b，计算其聚集值；如果桶 b 满足条件：$v_b \le \left(f_b - f_b^-\right)^2$，则 $AR+ = f_b \bullet S_{intr} \bullet \omega^2$，其中：$\omega$ 为分辨率，ω^2 为分辨率的平方，否则，$AR+ = \left[f_b^- + Kg_b(f_b - f_b^-)\right] \bullet S_{intr} \bullet \omega^2$。

本发明采用上述技术方案，具有以下有益效果： 解决了现有数据流聚集查询方法忽略最大相对误差的不足，提高了查询的精度和方法的可用性。

附图说明

图 1 为根据本发明一实施例的交通数据流聚集查询系统的结构示意图；

图 2 为根据本发明一实施例的交通数据流聚集查询方法的流程示意图；

图 3 为根据本发明一实施例的聚集查询方法中的二维交通路网在时间戳 0 的数据分布示意图；

图 4 为根据本发明一实施例的聚集查询方法中的二维交通路网在时间戳 1 的数据分布示意图；

图 5 为根据本发明一实施例的聚集查询方法中的 RH 结构中桶划分示意图；

图 6 为根据本发明一实施例的聚集查询方法中的 RH 结构中 BPT 的示意图；

图 7 为根据本发明一实施例的聚集查询方法中的 RH 桶分裂过程中桶数据更新情况示意图；

图 8 为根据本发明一实施例的聚集查询方法中的 RH 桶分裂后的桶划分示意图；

图 9 为根据本发明一实施例的聚集查询方法中的 RH 桶分裂后 BPT 索引的示意图；

图 10 为根据本发明一实施例的聚集查询方法中的 RH 方法与 AMH 的最大误差对比分析图。

具体实施方式

为使本领域技术人员更好地理解本发明的技术方案，下面结合附图和具体实施方式对本发明作进一步详细描述。下文中将详细描述本发明的实施方式，所述实施方式的示例在附图中示出，其中自始至终相同或类似的标号表示相同或类似的元件、组件、模块或具有相同或类似功能的元件、组件、模块。下面通过参考附图描述的实施方式是示例性的，仅用于解释本发明，而不能解释为对本发明的限制。

本技术领域技术人员可以理解，除非特意声明，这里使用的单数形式“一”、“一个”、“所述”和“该”也可包括复数形式。应该进一步理解的是，本发明的说明书中使用的措辞“包括”是指存在所述特征、整数、步骤、操作、元件、组件和/或模块，但是并不排除存在或添加一个或多个其他特征、整数、步骤、操作、元件、组件、模块和/或它们的组。这里使用的措辞“和/或”包括一个或更多个相关联的列出项的任一单元和全部组合。

本技术领域技术人员可以理解，除非另外定义，这里使用的所有术语（包括技术术语和科学术语）具有与本发明所属领域中的普通技术人员的一般理解相同的意义。还应该理解的是，诸如通用字典中定义的那些术语应该被理解为具有与现有技术的上下文中的意义一致的意义，并且除非像这里一样定义，不会用理想化或过于正式的含义来解释。

如下文所述，通过理论推导和大量试验的验证，本发明创造性地发现聚集查询方法的可用性是由最大相对误差决定的，但是现有的近似聚集查询方法均以平均查询误差的大小去衡量算法的优劣忽略了能够产生最大相对误差或者较大相对误差的异常点对聚集查询方法本身性能或适用性的影响。针对这种情况，本发明率先提出运用卡尔曼滤波器原理对现有聚集查询方法进行改进，通过校正状态先验估计以获得后验估计的方法，利用桶的频率最优估计计算异常点聚集值，有效地抑制异常点的最大相对误差，为聚集查询方法的可用性提供可靠保障。优选地，本发明以交通数据作为应用场景，从而本发明将改进的聚类查询方法应用于交通流数据查询和分析中。

图 1 为基于卡尔曼滤波的交通数据流聚集查询系统的结构示意图。如图 1 所示，本发明提供一种交通数据流的聚集查询系统可以包括：信息收集模块、数据处理模块、索引处理模块、应用服务模块和索引存储模块。

信息收集模块用以接收各种移动对象的信号，其中包括：对象编号 OID、位置信息 LOC、时间 t，并将接收的移动对象的信号发送给数据处理模块。

数据处理模块用以将接收到的信号编码并转变成数据处理引擎可以处理的数据形式。当编码数据达到一个系统时间戳后，将编码后的数据以数据流的形式发送给信息处理引擎。

数据处理引擎用以对接收到的数据流进行索引处理，其可以包括：索引处理模块和索引存储模块。索引可以包括当前索引和历史索引，当前索引是指在当前系统时间戳内的索引，历史索引是指索引的时间戳小于当前系统时间戳的索引；为了提高检索效率和存储效率，系统利用索引处理模块对当前索引进行序列化，生成历史索引。如果在索引存储模块中没有当前时间戳的索引，就创建新的索引文件；如果当前时间戳的索引已经存在，就根据接收到的数据流更新索引文件。当系统时间戳结束后，索引存储模块将索引输出给索引处理模块。索引处理模块

用以接收当前索引，进行序列化后生成历史索引并存储到索引存储模块。索引存储模块用以存储整个系统的索引，包括当前索引和历史索引（序列化存储）。在当前时间戳结束后，索引存储模块将当前时间索引（BPT）输出给索引处理模块。在有查询请求的时候，索引存储模块将全部索引输出给应用服务模块。

应用服务模块用以接收用户的查询请求，对索引存储模块中的索引进行聚集查询，生成聚集查询的结果。

由于移动对象信息是具有时空属性的数据流，为了便于处理，系统将索引的时间离散化成一系列等间隔的时间点，连续的两个时间点形成左开右闭的区间。对于时间在该区间内的移动对象，其时间戳均以该区间的右端时间点表示。

图 2 为基于卡尔曼滤波的交通数据流聚集查询方法的流程示意图。如图 2 所示，通过本发明提供的交通数据流的聚集查询系统，本发明提供的交通数据流的聚集查询方法可以包括：首先，移动对象信息获取与预处理；接着，判断存储索引模块中是否有 BPT 索引；如果判断为“是”，则基于卡尔曼滤波原理更新当前时间戳 BPT 索引；如果判断为“否”，则创建当前时间戳的新 BPT 索引，随后再基于卡尔曼滤波原理更新当前时间戳 BPT 索引；接着，当前时间戳 BPT 索引更新之后，判断当前时间戳是否结束；如果判断为“是”，则生成历史时间戳索引；如果判断为“否”，则返回移动对象信息获取与预处理步骤；接着，生成历史时间戳索引之后，基于卡尔曼滤波原理进行交通流聚集查询。

作为一种实施例，本发明提供的聚集查询方法可以采用求和聚集查询 $SUM(r,t)$，其中：t 代表时间戳（timestamp），r 代表查询区域。聚集查询 $SUM(r,t)$ 的返回值是 t 时刻查询区域 r 中所包含的对象总数。如果 $t=0$，表示的是当前时间戳的一个查询；如果 $t<0$，表示的是过去历史时间戳的查询。

图 3 和图 4 为根据本发明一实施例的聚集查询方法中的二维交通路网在时间戳 0 和时间戳 1 的数据分布示意图。如图 3 和图 4 所示，将整个数据空间划分为 5•5 的单元网格，时间戳 0 的数据分布在图 3 中的深色区域中，时间戳 1 的数据分布在图 4 中的深色区域中。令 q_r 表示为图中阴影部分的面积，若当前时间戳 1，那么 $SUM(r,0)$ 值为 50，即阴影部分的显示数字求和 9+11+5+10+9+6=50，那么 $SUM(r,-1)$ 的值 7+8+3+7+7+3=35。若需要查询一个时间段的聚集值

$SUM(r,ts,te)$，则只需从时间段$[ts,te]$提取出相应时间戳，分别在这些时间戳上执行上述聚集查询，将查询结果求和即可，例如，$SUM(r,-1,0)$的值为50+35=85。

图5为根据本发明一实施例的聚集查询方法中的RH结构中桶划分示意图。如图5所示，为本发明的合理直方RH（Reasonable Histogram）结构图，RH通过2维的网格把数据空间分割为$\omega \bullet \omega$（ω为“分辨率”）单元，每个单元$c(1 \le c \le \omega^2)$对应有一个频率F_c，即频率F_c是当前时间戳单元c内移动对象的数量。

图6为根据本发明一实施例的聚集查询方法中的RH结构中BPT的示意图。如图6所示，RH将频率相似的邻近单元分组成一个桶，采用二叉划分树BPT（Binary Partition Tree）来索引桶。如图6所示，最初，BPT只有一个叶子节点，直方图只有覆盖整个数据空间的桶，通过桶的分裂来产生新的桶（叶子节点），但是桶的个数(n)不能超过系统设定的上限（记为B）。如图6所示；对于每个桶b需要存储桶的矩形范围RE，桶中所有单元的平均频率f_b（记为桶的频率），以及桶内单元频率的平方的平均值g_b以及桶的方差v_b和卡尔曼增益Kg_b。

对于RH的每个桶b，n_b表示桶中的单元数量，则有：

单元的平均频率（均值）：

$$f_b = (1/n_b) \sum_{\forall c \ in \ b} F_c \tag{1}$$

桶的变化量（方差）：

$$v_b = (1/n) \sum_{\forall c \ in \ b} (F_c - f_b)^2 \tag{2}$$

RH的目标是所有桶的变化量总和（Weighted Variance Sum, WVS）最小。

$$WVS = \sum_{i=1\sim n} (n_i \bullet v_i) \tag{3}$$

对于b需要存储的信息包括：桶的矩形范围RE，RE中所有单元的平均频率f_b，以及这些单元平均频率平方的平均值。

$$g_b = (1/n) \sum_{\forall c \ in \ RE} F_c^2 \tag{4}$$

因此，对RH的查询过程就是遍历BPT找出每个与查询区域相交的桶，计算出相交的面积S_{intr}，然后计算出与查询区域相交面积所覆盖的单元数量$S_{intr} \bullet \omega^2$，则对于的求和聚集查询值为：

$$SUM = f_b \bullet S_{intr} \bullet \omega^2 \tag{5}$$

分析公式(5)可以发现，若直接采用公式(5)计算桶聚集值，利用桶的平均频率 f 去代替 F_c，当桶的方差较大时，将会形成较大相对误差，影响聚集查询方法的可用性。因此，本发明采用卡尔曼滤波方法对聚集查询值进行优化，利用桶频率的最优估计代替当前桶的平均频率 f。具体方法如下：

将卡尔曼滤波器的状态估计方程应用于 RH，则当前时刻桶频率的最优估计值 $\hat{f}_b$ 为：

$$\hat{f}_b = \left[f_b^- + Kg_b(f_b - f_b^-) \right] \tag{6}$$

其中 f_b^- 表示前一时刻的桶的平均频率，f_b 表示当前时刻桶的平均频率，Kg_b 为桶的卡尔曼增益。需要说明的是，这里采用符号 Kg_b 表示本发明中桶的卡尔曼增益，后文将采用符号 Kg 表示本发明桶中的单元 c 的卡尔曼增益。

利用 $\hat{f}_b$ 代替当前的桶频率，以抑制异常点聚集查询的最大相对误差，则公式(5)变化为：

$$SUM_{intr} = \left[f_b^- + Kg_b(f_b - f_b^-) \right] \bullet S_{intr} \bullet \omega^2 \tag{7}$$

根据卡尔曼滤波器的原理可知，卡尔曼增益可以由系统的协方差矩阵计算而来（式(8)），由于 P 和 R 是协方差矩阵，即 P≥0，R≥0，$Kg_b \geq 0$。

$$Kg = P^- \left(P^- + R \right)^{-1} \tag{8}$$

对公式(6)分析，当 $f_b \neq f_b^-$ 时，可得：

$$Kg_b = \frac{\hat{f}_b - f_b^-}{f_b - f_b^-} \tag{9}$$

对于 RH 中的桶 b，其变化量 v_b 可以定义为：

$$v_b = \frac{1}{n} \sum_{\forall c \, in \, b} (F_c - f_b)^2 \tag{10}$$

由于采用公式(5)的误差来源于 F_c 与 f_b 之间的差值，当利用 f_b 代替 F_c 进行聚集查询时，为了能够有效的抑制最大相对误差，需要使桶中所有单元具有相同方差，公式(10)可以转换为：

$$v_b = (1/n)\bullet n\bullet\left(f_b - \overline{f}_b\right)^2 = \left(f_b - \overline{f}_b\right)^2 \tag{11}$$

这样 $\overline{f}_b$ 相对于 f_b 更加趋近 F_c，这里的 $\overline{f}_b$ 对应于公式(9)中的 $\hat{f}_b$，所以：

$$\hat{f}_b = f_b \pm \sqrt{v_b} \tag{12}$$

将公式(12)带入公式(9)，可得：

$$Kg_b = \frac{f_b \pm \sqrt{v_b} - f_b^-}{f_b - f_b^-} \geq 0 \tag{13}$$

分析公式(13)可得：

$$v_b \leq \left(f_b - f_b^-\right)^2 \tag{14}$$

若 $f_b - f_b^- = 0$，则无需计算卡尔曼增益，直接利用公式(5)计算。

证明如下：

1） 当 $f_b - f_b^- > 0$ 时

因为 $\frac{f_b \pm \sqrt{v_b} - f_b^-}{f_b - f_b^-} \geq 0$，那么 $f_b \pm \sqrt{v_b} - f_b^- \geq 0$

所以 $\pm\sqrt{v_b} \leq f_b - f_b^-$ 成立，即 $v_b \leq \left(f_b - f_b^-\right)^2$。

2） 当 $f_b - f_b^- < 0$ 时

因为 $\frac{f_b \pm \sqrt{v_b} - f_b^-}{f_b - f_b^-} \geq 0$，那么 $f_b \pm \sqrt{v_b} - f_b^- < 0$，

所以 $\pm\sqrt{v_b} \leq f_b^- - f_b$ 成立，即 $v_b \leq \left(f_b - f_b^-\right)^2$。

综上所述，$v_b \leq \left(f_b - f_b^-\right)^2$。

当 RH 中的某个桶不满足公式(14)时，说明这个桶的变化量很大，将有很大的概率产生极大的误差，需要通过卡尔曼滤波方法进行最优估计处理，并通过 $\hat{f}_b$ 代替 f_b，以抑制误差，此时该桶的聚集查询值应采用公式(7)计算；当桶满足公式(14)时，说明桶的变化量不大，直接利用公式(5)，因此对于特定的桶，聚集查询值可以定义为：

$$SUM = \begin{cases} f_b \bullet S_{intr} \bullet \omega^2 & v_b \leq \left(f_b - f_b^-\right)^2 \\ \left[f_b^- + Kg_b(f_b - f_b^-)\right] \bullet S_{intr} \bullet \omega^2 & v_b > \left(f_b - f_b^-\right)^2 \end{cases} \tag{15}$$

BPT 索引存储每个桶的平均频率（f）、方差(v)和每个单元的移动对象数据（F_c）以及 Kg，公式(15)给出 RH 的聚集值计算公式，因此还需要计算 RH 中单元和桶的卡尔曼增益。分析公式(13)可得，Kg 两个取值，分别记为：

$$Kg^{+}=\left(f+\sqrt{v}-f^{-1}\right)/\left(f-f^{-1}\right) \tag{16}$$

$$Kg^{-}=\left(f-\sqrt{v}-f^{-1}\right)/\left(f-f^{-1}\right) \tag{17}$$

下面继续分析 Kg 的取值，RH 的目标是 $\hat{f}^{+}$ 尽可能的逼近 F_c，这样才能保证误差最小，所有对于桶中的每个单元：

当 $F_c > f$ 时，$\hat{f}^{+}$ 更接近 $f+\sqrt{v}$，这个单元的卡尔曼增益为 Kg^{+}；

当 $F_c \le f$ 时，$\hat{f}^{+}$ 更接近 $f-\sqrt{v}$，这个单元的卡尔曼增益为 Kg^{-}。

设一个桶中单元频率大于桶的平均频率的单元数为 n_{b1}，则这个桶的卡尔曼增益为：

$$Kg_b = n_{b1}\bullet Kg^{+}+\left(n_b-n_{b1}\right)\bullet Kg^{-} \tag{18}$$

如图 2 所示，一种基于卡尔曼滤波的交通流聚集查询方法，其索引创建过程包括步骤如下：

步骤 A，移动对象信息获取与预处理，具体包括如下步骤：

步骤 A-1：信息收集模块获取移动对象的信号，并转换成为数据流形式；

步骤 A-2：当达到一个系统时间戳时，信息收集模块将移动对象数据流发送给数据处理模块；

步骤 B，数据处理模块判断索引存储模块中是否有当前时间戳的 BPT 索引，若有转向步骤 D，否则转向步骤 C；

步骤 C，创建一个当前时间戳的新 BPT 索引（Kg 和 f^{-}均为 0），转向步骤 D；

步骤 D，基于卡尔曼滤波原理更新 BPT 索引，例如，下表算法 1 所示步骤：

步骤 D-1，对于每一个变更的单元 c，计算其变化量 d，更新其频率信息：

$F_c^{-}=F_c, F_c=F_c+d$（F_c^{-} 表示前一个时间戳的频率）；

步骤 D-2，对于每一个变更的单元 c，遍历 BPT 索引找到包含该变化单元的桶 b，更新桶 b 的频率和卡尔曼增益：

$f_b^{-}=f_b, f_b=f_b+d, \Delta g=f_b^{2}-f_b^{-2}$，$g_b=(n_b.g_b+\Delta g)/n_b, v_b=g_b-f_b^{2}$；

步骤 D-3，找出桶中频率大于平均频率的单元的数量n_{b1}，计算桶的卡尔曼增益：$Kg_b = n_{b1} \bullet Kg^+ + (n_b - n_{b1}) \bullet Kg^-$；

算法 1 桶更新算法

Algorithm 1 Update()

1. 找到变更的单元 c; (d 是单元 c 的变化量)
2. $F_c^- = F_c$ ， $F_c = F_c + d$;
3. 将单元 c 置于栈 S_c 中，数据格式为<单元 c, 变化量 d>;
4. While(S_c 不为空){
5. $c_s = S_c.pop()$;
6. 遍历索引 BPT 找出包含单元 c_s 桶 b;
7. $f_b^- = f_b, f_b = f_b + d, \Delta g = f_b^2 - f_b^{-2}$;
8. $g_b = (n_b.g_b + \Delta g)/n_b, v_b = g_b - f_b^2$;
9. 将桶 b 加入到哈希表 H 之中;}
10. While(*H.hasNext()*){
11. $b = H.next()$;
12. 找出 b 中频率大于平均频率的单元的数量（记为 n_b）;
13. 计算 b 的卡尔曼增益：$Kg_b = n_{b1} \bullet Kg^+ + (n_b - n_{b1}) \bullet Kg^-$； }
14. Return;

End Algorithm 1 Update()

其中：pop() 方法用于删除并返回数组的最后一个元素，具体而言，pop() 方法将删除 arrayObject 的最后一个元素，把数组长度减 1，并且返回它删除的元素的值；如果数组已经为空，则 pop() 不改变数组，并返回 undefined 值。*hasNext()*方法用于返回数组是否还有下一个元素，若有则返回 *TRUE*，否则，返回 *FALSE*；

步骤 E，根据桶的方差判断 BPT 索引中的桶是否需要分裂，若其方差大于事先预设的阈值，则需要进行分裂，转向步骤 F，否则转向步骤 G;

步骤 F，通过穷举的方式计算每个需要分裂桶 b 的每种分裂情况的分割利益（Split Benefit, SB）和对应划分位置，记 b 分割前的 cell 数量为 n_b 和方差为 v_b，

分隔成为 b1 和 b2（对应的 cell 和方差分别为：n_{b1}、n_{b2} 和 v_{b1}、v_{b2}），则分隔利益 $SB = n_b \bullet v_b - \left(n_{b1} \bullet v_{b1} + n_{b2} \bullet v_{b2}\right)$，SB 值最高的记为最高分割利益（BSP）；

按照 BSP 将桶 b 分为两个子桶 b1 和 b2，并且设置 $f_{b1}^- = f_{b2}^- = f_b^-$ 分割后的初始值，利用步骤 D 的方法更新 BPT 索引（即，返回步骤 D 进行处理）计算相应的卡尔曼增益；

桶的分裂就是对变化量很大的桶的区域产生分割，它能提高 RH 的查询精度，具体处理步骤如下表算法 2 所示：

算法 2 桶分割算法

Algorithm2 Split()

1. 遍历 BPT，计算每个桶的最高分割利益以及最优划分位置;

2. 通过函数 SortedMap()将这些计算好的最高分割利益、最优划分位置以及相应的桶放入集合 *SM* 之中，数据的形式为<最高分割利益, 包含最优划分位置以及相应桶的信息的对象>;

3. While(没有新的数据或者查询到达，并且桶的数量 $n < B$ 以及集合 *SM* 有正的最高分割利益){

4. 从集合 *SM* 中删除具有最大的最高分割利益的键值对，其删除的最高分割利益所对应的桶为 *b*;

5. 按照最优划分位置将桶 *b* 划分为两个子桶 b1 和 b2，并设置 $f_{b1}^- = f_{b2}^- = f_b^-$，利用算法 1 更新 BPT 索引;

6. 分别计算桶 b1 和 b2 的最高分割利益以及最优划分位置;

7. 将 b1 和 b2 的相关信息置入 *SM*;}

8. Return ;

End Algorithm Split()

SortedMap*()*方法用于向 Hash 数据 SM 存储一条数据，添加到数组的尾部，并将数据长度增加 1；

如图 7 所示，单元数据的变化以较大的粗体字表示（相对于图 4），三种方式划分桶 b_1（一个在 x 轴，两个在 y 轴），分别计算其分隔利益，x 轴上划分方

式的分割利益 SB 的计算公式为$n_1 \bullet v_1 - \left(n_{12} \bullet v_{12} + n_{13} \bullet v_{13}\right)$，同理其他桶的分割利益也要进行计算（即 y 轴上的划分），最终 WVS 的降幅最大的分隔是 x 轴上的划分。新的桶划分及二叉树如图 8 和图 9 所示。

步骤 G，判断索引中桶的数量是否达到系统设置的桶数量最大上限（B），若是转向步骤 H，否则转向步骤 I；

步骤 H，对索引中桶进行合并操作，计算出所有可能合并桶（该桶在 BPT 中必须是非叶子节点，并且其两个儿子节点都是叶子节点）的合并惩罚（Merge Penalty, MP），选取合并惩罚最小的多个桶合并为一个桶；

合并惩罚的计算规则与分隔利益类似，由合并后形成桶的 cell 数量乘以方差减去合并前两个桶的 cell 数量和方差积之和，例如：BPT 中叶子节点 b1 和 b2（对应的 cell 和方差分别为：nb1、nb2 和 vb1、vb2）可能合并成为其父节点 b（cell 数量为 nb 和方差为 vb），则合并惩罚计算公式为：$MP = n_b \bullet v_b - \left(n_{b1} \bullet v_{b1} + n_{b2} \bullet v_{b2}\right)$；

并利用步骤 D 的方法更新 BPT 索引；

步骤 I，判断当前时间戳是否结束，若结束则转向步骤 J，否则转向步骤 A；例如，系统会将处于特定时间区域内的例如 08:00:00-08:00:09 内的时间利用时间戳 08:00:00 表示（每 10 秒用一个时间戳表示），当数据流中数据的时间为 08:00:10 时即表示进入下一个时间戳；

步骤 J，索引处理模块从索引存储模块中取出当前时间戳的 BPT 索引，生成历史索引，并将其序列化存储到索引存储模块。

如图 2 所示，一种基于卡尔曼滤波的交通流聚集查询方法，其聚集查询包括如下步骤：

步骤 A，解析用户的查询请求$Q\left(r, ts, te\right)$，提取查询时间区间$\left[ts, te\right]$的系统时间戳集合$\left\{t_1, t_2, \cdots, t_k \mid t_s \le t_1 < t_2 < \cdots < t_k \le t_e\right\}$，将集合中的这些时间戳按时间序置入栈 *T*;

其中：r 表示查询区域，ts 表示起始时间，te 表示终止时间；t1…tk 表示起始时间与终止时间之间的 1-K 个时间戳。

步骤 B，设置聚集查询值 AR=0；

步骤 C，对于 T 中的每一个时间戳，执行聚集查询，例如，具体子步骤如下：

步骤 C-1，遍历 BPT 找到所有与查询区域 r 在空间上相交的桶 b，计算 r 与桶 b 的相交面积 S_{intr}，并将这些桶以<b, S_{intr}>的形式置入栈 S；

步骤 C-2，对于 S 中的每一个桶 b，计算其聚集值；

如果桶 b 满足条件：$v_b \le \left(f_b - f_b^-\right)^2$，则 $AR+ = f_b \bullet S_{intr} \bullet \omega^2$，（$\omega$ 为“分辨率”，ω^2 为“分辨率的平方”）

否则，$AR+ = \left[f_b^- + Kg_b(f_b - f_b^-)\right] \bullet S_{intr} \bullet \omega^2$;

步骤 D，返回聚集查询值 AR，查询结束；

针对单一时间戳的空间聚集查询，如下表算法 3 所示，对于每一个与查询区域 r 相交的桶 b：首先，计算桶 b 查询区域 r 相交部分的面积 S_{intr}；然后，要检测这个桶 b 是否满足预设条件（即是否满足公式(14)），进而做相应的处理（参照下面罗列的算法），得到相应的中间结果 $f_b \bullet S_{intr} \bullet \omega^2$；最后，将所有的中间结果累加以得到单一时间戳的空间聚集查询值。对于多个时间戳的查询结果是对单一时间戳的查询结果的累加。

算法 3 单一时间戳的空间聚集查询

Algorithm3 Query()

1. 根据查询区域条件，遍历 BPT 找到与查询区域相交的桶 b，并将这些桶以<b, S_{int}>的形式置入栈 S 中;
2. 设置聚集结果 AR=0.0;
3. While(栈 S 不为空){
4. <b, S_{intr}> = $S.pop()$;
5. If(桶满足条件：$v_b \le \left(f_b - f_b^-\right)^2$)
6. $AR+ = f_b \bullet S_{intr} \bullet \omega^2$;
7. else
8. $AR+ = \left[f_b^- + Kg_b(f_b - f_b^-)\right] \bullet S_{intr} \bullet \omega^2$;
9. }

10. Return *AR*;

End Algorithm Query()

本发明可以采用计算机实现，本实施例选取了宁波公共交通路网所采集的数据进行实验（数据量 100 万条），数据流的格式为<T, LOC, OID >，即<时间、位置信息和车辆标识>，共记录了 10 个时刻的车辆信息（时间间隔为 10s）。为了对比分析本发明的抑制最大相对误差的效果，我们与经典的交通流聚集查询方法 AMH 方法进行对分析（本发明方法记为 RH）。

我们分别对 RH 和 AMH 进行了 100 次聚集查询（RH 和 AMH 的桶上限 B=350，r=12，查询的时间区间包含 3 个时间戳），实验结果如附图 6 所示， AMH 最大相对误差为 71.875%，平均相对误差为 3.028%，最大的 5 个相对误差的平均值为 38.067%。（最大的 5 个相对误差从大到小分别为 71.875%，40.946%，36.975%，20.557%，20.000%，一共有 5 个异常点。）RH 最大相对误差为 36.957%，平均相对误差为 2.461%，最大的 5 个相对误差的平均值为 26.853%。（最大的五个相对误差从大到小分别为 36.975%，34.247%，33.083%，15.788%，14.189%，其中有 3 个异常点）。不难发现，与 AMH 相比，RH 在抑制异常点从而降低最大相对误差方面具有明显的优势。

综上所述，本发明所涉及的基于卡尔曼滤波的交通数据流聚集查询系统及方法，解决了现有聚集查询方法忽略最大相对误差的问题，提高了聚集查询方法的可用性性。

本技术领域技术人员可以理解，本发明可以涉及用于执行本申请中所述操作中的一项或多项操作的设备。所述设备可以为所需的目的而专门设计和制造，或者也可以包括通用计算机中的已知设备，所述通用计算机有存储在其内的程序选择性地激活或重构。这样的计算机程序可以被存储在设备（例如，计算机）可读介质中或者存储在适于存储电子指令并分别耦联到总线的任何类型的介质中，所述计算机可读介质包括但不限于任何类型的盘（包括软盘、硬盘、光盘、CD-ROM、和磁光盘）、随即存储器（RAM）、只读存储器（ROM）、电可编程 ROM、电可擦 ROM（EPROM）、电可擦除可编程 ROM（EEPROM）、闪存、磁性卡片或光线卡片。可读介质包括用于以由设备（例如，计算机）可读的形式存储或传输信息的任何机构。例如，可读介质包括随即存储器（RAM）、只读存储器（ROM）、

磁盘存储介质、光学存储介质、闪存装置、以电的、光的、声的或其他的形式传播的信号（例如载波、红外信号、数字信号）等。

本技术领域技术人员可以理解，可以用计算机程序指令来实现这些结构图和/或框图和/或流图中的每个框以及这些结构图和/或框图和/或流图中的框的组合。可以将这些计算机程序指令提供给通用计算机、专业计算机或其他可编程数据处理方法的处理器来生成机器，从而通过计算机或其他可编程数据处理方法的处理器来执行的指令创建了用于实现结构图和/或框图和/或流图的框或多个框中指定的方法。

本技术领域技术人员可以理解，本发明中已经讨论过的各种操作、方法、流程中的步骤、措施、方案可以被交替、更改、组合或删除。进一步地，具有本发明中已经讨论过的各种操作、方法、流程中的其他步骤、措施、方案也可以被交替、更改、重排、分解、组合或删除。进一步地，现有技术中的具有与本发明中公开的各种操作、方法、流程中的步骤、措施、方案也可以被交替、更改、重排、分解、组合或删除。

以上所述仅是本发明的部分实施方式，应当指出，对于本技术领域的普通技术人员来说，在不脱离本发明原理的前提下，还可以做出若干改进和润饰，这些改进和润饰也应视为本发明的保护范围。

权 利 要 求 书

1、一种交通数据流的聚集查询方法，其特征在于，包括如下步骤：

步骤 1，采集移动对象信息，将移动对象信息转化为计算机可处理的数据形式，在系统时间戳到来时形成数据流；

步骤 2，在系统时间戳到来时生成、更新索引文件；

步骤 3，在生成更新索引文件的同时，根据用户的查询请求 $SUM\left(r,ts,te\right)$ 对空间区域 r 进行交通数据流查询，提取查询时间区间 $\left[ts,te\right]$ 的系统时间戳，对于每个时间戳执行空间聚集查询：遍历当前时间戳 t 的索引文件，利用如下表达式求得当前时间戳下空间区域 r 在各桶的聚集查询值：

$$SUM=\begin{cases} f_{bi}\bullet S_{intr}\bullet\omega^2 & ,v_{bi}\le\left(f_{bi}-f_{bi}^{-}\right)^2 \\ \left[f_{bi}^{-}+Kg_{bi}(f_{bi}-f_{bi}^{-})\right]\bullet S_{intr}\bullet\omega^2 & ,v_{bi}>\left(f_{bi}-f_{bi}^{-}\right)^2 \end{cases},$$

SUM 为空间区域 r 在第 bi 个桶的聚集查询值，S_{intr} 为空间区域 r 与第 bi 个桶相交区域的面积，f_{bi} 为当前时间戳第 bi 个桶的平均频率，f_{bi}^{-} 为前一时间戳 t^{-} 第 bi 个桶的平均频率，v_{bi} 第 bi 个桶的方差，

将每个时间戳的查询值求和形成最终的聚集查询值。

2、根据权利要求 1 所述的交通数据流的聚集查询方法，其特征在于，所述步骤 2 包括：

步骤 A，初始化第一个系统时间戳的数据流生成的索引文件：采用合理直方图将数据空间分割为 $\omega\bullet\omega$ 的单元，以当前时间戳内单元内的移动对象数量表示该单元的频率，再将频率相似的邻近单元组成一个桶，形成 n 个桶，$0<n\le B$，ω 为分辨率，B 为桶数目的上限 ，

对于每个桶：以桶中所有单元的平均频率作为该桶的频率，计算该桶中各单元平均频率方差的平均值、该桶的方差以及卡尔曼增益；

步骤 B，在下一系统时间戳到来时，利用卡尔曼滤波原理更新索引文件：

步骤 B-1，当第 c 单元中的数据变化时，记数据变化量为 d，更新第 c 单元的频率 F_c：$F_c^{-}=F_c, F_c=F_c+d$，其中：F_c^{-} 为前一时间戳单元 c 的频率，$1\le c\le\omega^2$，d 为任意实数；

步骤 B-2，遍历当前时间戳的索引文件找到包含数据量变化单元的桶，对于

第 b 个桶，第 b 个桶包含有 n_b 个单元，$b<n$，$n_b<\omega\bullet\omega$：

更新第 b 个桶的频率 f_b：，$f_b=f_b^-+d,\Delta g=f_b^2-f_b^{-2}$，

更新第 b 个桶中各单元频率平方的平均值 g_b、方差 v_b：$g_b=(n_b.g_b^-+\Delta g)/n_b, v_b=g_b-f_b^2$ ，

更新第 b 个桶中第 i 单元的卡尔曼增益，F_i 为第 i 单元的频率：

当 $F_i>f_b$ 时，第 i 单元的卡尔曼增益为：$Kg_i^+=\dfrac{f_b+\sqrt{v_b}-f_b^-}{f_b-f_b^-}$，

当 $F_i\le f_b$ 时，第 i 单元的卡尔曼增益为：$Kg_i^-=\dfrac{f_b-\sqrt{v_b}-f_b^-}{f_b-f_b^-}$，

计算出第 b 个桶中频率大于平均频率的单元数目 n_{b1}，更新第 b 个桶的卡尔曼增益 Kg_b：$Kg_b=n_{b1}\bullet Kg_i^+ +\left(n_b-n_{b1}\right)\bullet Kg_i^-$，$1\le i\le n_{b1}\le n_b$；

步骤 B-3-A，对于需要分裂的桶，计算每个需要分裂桶的最高分割利益和最优划分位置，按照最优划分位置将需要分裂的桶分为两个子桶，并且设置前一系统时间戳内两个子桶的频率均与分裂前桶的频率相等，重复步骤 B-1；

步骤 B-3-B，对于不需要分裂的桶，在索引中桶的数量达到上限时，利用最小合并惩罚原理将频率集中的多个桶合并为一个桶，重复步骤 B-1；

步骤 B-3-C，对于不需要分裂的桶，在索引中桶的数量未达上限且当前系统时间戳尚未结束的情况下，返回步骤 1；

步骤 B-3-D，对于不需要分裂的桶，在索引中通的数量未达上限且当前时间戳结束的情况下，提取当前时间戳的索引文件生成历史索引；

3、根据权利要求 1 所述的交通数据流的聚集查询方法，其特征在于，所述步骤 2 采用二叉划分树结构来索引桶。

4、一种交通数据流的聚集查询系统，其特征在于，包括：

信息收集模块，采集移动对象的信息，将移动对象信息转化为计算机可处理的数据形式，在系统时间戳到来时形成数据流并发送给数据处理引擎；

数据处理模块，将系统时间戳内的空间数据划分为子单元，把包含移动对象数量相近的单元组成一个桶，生成桶的索引文件，计算桶的频率、各单元平均频

率方差的平均值、方差以及卡尔曼增益，在新的系统时间戳到来时利用卡尔曼滤波原理更新索引文件；

索引存储模块，用以存储索引生成和更新模块生成的索引文件；

应用服务模块，调用索引存储模块查找符合查询请求的索引文件，并反馈聚值查询值。

5、根据权利要求 4 所述的系统，其特征在于，所述处理模块用于接收当前时间戳的索引，在当前时间戳结束后将当前时间戳的索引与历史索引序列化处理后输出至引存储模块。

6、一种基于卡尔曼滤波的交通流的索引创建方法，其特征在于，包括：

步骤 A，信息收集模块获取移动对象信息并进行预处理；

步骤 B，数据处理模块判断索引存储模块中是否有当前时间戳的 BPT 索引，若有转向步骤 D，否则转向步骤 C；

步骤 C，创建一个当前时间戳的新 BPT 索引，其中：Kg 和 f^- 均为 0，转向步骤 D；

步骤 D，基于卡尔曼滤波原理更新 BPT 索引；

步骤 E，根据桶的方差判断 BPT 索引中的桶是否需要分裂，若其方差大于事先预设的阈值，则需要进行分裂，转向步骤 F，否则转向步骤 G；

步骤 F，通过穷举的方式计算每个需要分裂桶 b 的每种分裂情况的分割利益 SB 和对应划分位置，记 b 分割前的 cell 数量为 n_b 和方差为 v_b，分隔成为 b1 和 b2，其中：对应的 cell 和方差分别为：n_{b1}、n_{b2} 和 v_{b1}、v_{b2}，则分隔利益 $SB = n_b \bullet v_b - \left(n_{b1} \bullet v_{b1} + n_{b2} \bullet v_{b2}\right)$，SB 值最高的记为最高分割利益（BSP），

按照 BSP 将桶 b 分为两个子桶 b1 和 b2，并且设置 $f_{b1}^- = f_{b2}^- = f_b^-$ 分割后的初始值，返回步骤 D 更新 BPT 索引并计算相应的卡尔曼增益；

步骤 G，判断索引中桶的数量是否达到系统设置的桶数量最大上限（B），若是转向步骤 H，否则转向步骤 I；

步骤 H，对索引中桶进行合并操作，计算出所有可能合并桶的合并惩罚 MP，其中：该桶在 BPT 中必须是非叶子节点，并且其两个儿子节点都是叶子节点，选取合并惩罚最小的多个桶合并为一个桶；

合并惩罚的计算规则与分隔利益类似，由合并后形成桶的cell数量乘以方差减去合并前两个桶的 cell 数量和方差积之和，则合并惩罚计算公式为：$MP = n_b \bullet v_b - \left(n_{b1} \bullet v_{b1} + n_{b2} \bullet v_{b2}\right)$，并利用步骤D的方法更新BPT索引；

步骤I，判断当前时间戳是否结束，若结束则转向步骤J，否则转向步骤A；

步骤J，索引处理模块从索引存储模块中取出当前时间戳的BPT索引，生成历史索引，并将其序列化存储到索引存储模块。

7、根据权利要求6所述的方法，其特征在于，所述步骤A进一步包括：

步骤A-1：信息收集模块获取移动对象的信号，并转换成为数据流形式；

步骤A-2：当达到一个系统时间戳时，信息收集模块将移动对象数据流发送给数据处理模块。

8、根据权利要求6所述的方法，其特征在于，所述步骤D进一步包括：

步骤 D-1，对于每一个变更的单元 c，计算其变化量 d，更新其频率信息：

$F_c^- = F_c, F_c = F_c + d$ （F_c^-表示前一个时间戳的频率）；

步骤D-2，对于每一个变更的单元c，遍历BPT索引找到包含该变化单元的桶 b，更新桶 b 的频率和卡尔曼增益：

$f_b^- = f_b, f_b = f_b + d, \Delta g = f_b^2 - f_b^{-2}$， $g_b = (n_b.g_b + \Delta g)/n_b, v_b = g_b - f_b^2$；

步骤D-3，找出桶中频率大于平均频率的单元的数量n_{b1}，计算桶的卡尔曼增益：$Kg_b = n_{b1} \bullet Kg^+ + \left(n_b - n_{b1}\right) \bullet Kg^-$。

9、一种基于卡尔曼滤波的交通流聚集查询方法，其特征在于，包括：

步骤A，解析用户的查询请求$Q(r,ts,te)$，提取查询时间区间$[ts,te]$的系统时间戳集合$\{t_1,t_2,\cdots,t_k \mid t_s \le t_1 < t_2 < \cdots < t_k \le t_e\}$，将集合中的这些时间戳按时间序置入栈 T;

其中：r表示查询区域，ts表示起始时间，te表示终止时间；t1…tk表示起始时间与终止时间之间的1-K个时间戳；

步骤B，设置聚集查询值 AR=0；

步骤C，对于 T 中的每一个时间戳，执行聚集查询，其中：

针对单一时间戳的空间聚集查询，对于每一个与查询区域r相交的桶b：首

先，计算桶 b 查询区域 r 相交部分的面积 S_{intr}；然后，计算这个桶 b 的中间结果 $f_b \bullet S_{intr} \bullet \omega^2$；最后，将所有的中间结果累加以得到单一时间戳的空间聚集查询值；

对于多个时间戳的查询结果是对单一时间戳的查询结果的累加；

步骤 D，返回聚集查询值 *AR*，查询结束。

10、根据权利要求 9 所述的方法，其特征在于，所述步骤 C 进一步包括：

步骤 C-1，遍历 BPT 找到所有与查询区域 *r* 在空间上相交的桶 *b*，计算 *r* 与桶 *b* 的相交面积 S_{intr}，并将这些桶以<*b*, S_{intr}>的形式置入栈 *S*；

步骤 C-2，对于 *S* 中的每一个桶 b，计算其聚集值；

如果桶 *b* 满足条件：$v_b \le \left(f_b - f_b^-\right)^2$，则 $AR+ = f_b \bullet S_{intr} \bullet \omega^2$，其中：$\omega$ 为分辨率，ω^2 为分辨率的平方，

否则，$AR+ = \left[f_b^- + Kg_b(f_b - f_b^-)\right] \bullet S_{intr} \bullet \omega^2$。