(12) 按照专利合作条约所公布的国际申请

(19) 世界知识产权组织
国 际 局

(43) 国际公布日
2022 年 3 月 31 日 (31.03.2022)

WIPO | PCT

(10) 国际公布号
WO 2022/061995 A1

(51) 国际专利分类号:
G06F 16/22 (2019.01)

(21) 国际申请号: PCT/CN2020/121521

(22) 国际申请日: 2020 年 10 月 16 日 (16.10.2020)

(25) 申请语言: 中文

(26) 公布语言: 中文

(30) 优先权:
202011026535.8 2020年9月25日 (25.09.2020) CN

(71) 申请人: 浙江大学 (ZHEJIANG UNIVERSITY) [CN/CN]; 中国浙江省杭州市西湖区余杭塘路866号, Zhejiang 310058 (CN)。

(72) 发明人: 张丰 (ZHANG, Feng); 中国浙江省杭州市西湖区余杭塘路866号, Zhejiang 310058 (CN)。

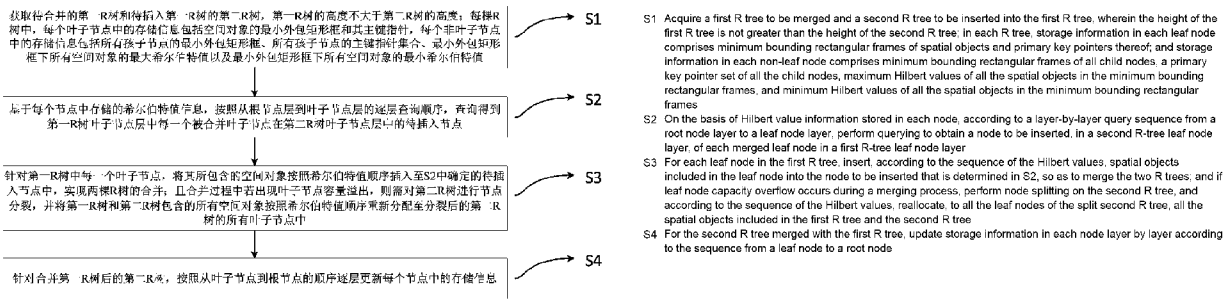
钱伯至 (QIAN, Bozhi); 中国浙江省杭州市西湖区余杭塘路866号, Zhejiang 310058 (CN)。 汪愿愿 (WANG, Yuanyuan); 中国浙江省杭州市西湖区余杭塘路866号, Zhejiang 310058 (CN)。 胡林舒 (HU, Linshu); 中国浙江省杭州市西湖区余杭塘路866号, Zhejiang 310058 (CN)。

(74) 代理人: 杭州求是专利事务所有限公司 (HANGZHOU QIUSHI PATENT OFFICE CO., LTD.); 中国浙江省杭州市玉古路147号黄鸿年科技综合大楼215室/傅朝栋, Zhejiang 310013 (CN)。

(81) 指定国 (除另有指明，要求每一种可提供的国家保护): AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DJ, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, IT,

(54) **Title:** R-TREE INDEX MERGING AND UPDATING METHOD AND APPARATUS BASED ON HILBERT CURVE, AND MEDIUM

(54) **发明名称:** 基于希尔伯特曲线的R树索引合并更新方法、装置及介质



S1 Acquire a first R tree to be merged and a second R tree to be inserted into the first R tree, wherein the height of the first R tree is not greater than the height of the second R tree; in each R tree, storage information in each leaf node comprises minimum bounding rectangular frames of spatial objects and primary key pointers thereof; and storage information in each non-leaf node comprises minimum bounding rectangular frames of all child nodes, a primary key pointer set of all the child nodes, maximum Hilbert values of all the spatial objects in the minimum bounding rectangular frames, and minimum Hilbert values of all the spatial objects in the minimum bounding rectangular frames

S2 On the basis of Hilbert value information stored in each node, according to a layer-by-layer query sequence from a root node layer to a leaf node layer, perform querying to obtain a node to be inserted, in a second R-tree leaf node layer, of each merged leaf node in a first R-tree leaf node layer

S3 For each leaf node in the first R tree, insert, according to the sequence of the Hilbert values, spatial objects included in the leaf node into the node to be inserted that is determined in S2, so as to merge the two R trees; and if leaf node capacity overflow occurs during a merging process, perform node splitting on the second R tree, and according to the sequence of the Hilbert values, reallocate, to all the leaf nodes of the split second R tree, all the spatial objects included in the first R tree and the second R tree

S4 For the second R tree merged with the first R tree, update storage information in each node layer by layer according to the sequence from a leaf node to a root node

(57) **Abstract:** An R-tree index merging and updating method and apparatus based on a Hilbert curve, and a medium. The method comprises the steps of: S1, acquiring a first R tree to be merged and a second R tree to be inserted into the first R tree, wherein upper and lower limit ranges of Hilbert values of included objects are stored in non-leaf nodes of the R trees; S2, by means of a "top-to-bottom" hierarchical query algorithm, querying a node to be inserted, in a second R-tree leaf node layer, of each merged leaf node in a first R-tree leaf node layer; S3, according to a "bottom-to-top" hierarchical adjustment algorithm, for each leaf node in the first R tree, inserting, according to the sequence of the Hilbert values, spatial objects included in the leaf node into the node to be inserted that is determined in S2, so as to merge the two R trees; and S4, for the second R tree after merging, updating storage information in each node layer by layer according to the sequence from a leaf node to a root node. The method has an important practical application value in the field of high-performance storage of geographical spatial-temporal big data.

(57) **摘要:** 一种基于希尔伯特曲线的R树索引合并更新方法、装置及介质。方法步骤包含：S1：获取待合并的第一R树和待插入第一R树的第二R树，R树的非叶子节点中存储所包含对象的希尔伯特值上下限范围；S2：通过"从上至下"层级查询算法查询第一R树叶子节点层中每一个被合并叶子节点在第二R树叶子节点层中的待插入节点；S3：按照"从下至上"层级调整算法，针对第一R树中每一个叶子节点，将其包含的空间对象按照希尔伯特值顺序插入至S2中确定的待插入节点中，实现两棵R树的合并；S4针对合并后的第二R树，按照从叶子节点到根节点的顺序逐层更新每个节点中的存储信息。上述方法在地理时空大数据高性能存储领域具有重要的实际应用价值。

WO 2022/061995 A1

[见续页]

JO, JP, KE, KG, KH, KN, KP, KR, KW, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, WS, ZA, ZM, ZW。

(84) 指定国(除另有指明，要求每一种可提供的地区保护)：ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, ST, SZ, TZ, UG, ZM, ZW), 欧亚 (AM, AZ, BY, KG, KZ, RU, TJ, TM), 欧洲 (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG)。

本国际公布：

— 包括国际检索报告(条约第21条(3))。

基于希尔伯特曲线的R树索引合并更新方法、装置及介质

技术领域

本发明涉及数据检索与更新的技术领域，尤其涉及一种希尔伯特 R 树索引的合并更新技术。

背景技术

空间数据的更新频率随着地理信息数据采集设备的发展而逐渐提高，对空间数据的管理不仅要满足高效的查询需求，还需要兼顾快速的更新需求。结合 LSM 树结构的空间索引能够有效支撑空间数据频繁插入和更新的需求，其中 LSM R 树索引相较于 DBH 树、DHVB 树以及 SHB 树，有更好的普适性和稳定性。但是 LSM R 树的实际应用过程中往往会涉及到大量 R 树的合并操作，R 树在合并过程中，会占用系统的一定的内存以及计算资源，影响在此期间对索引进行查询、插入、删除等操作的效率。因此需要高效的希尔伯特 R 树的合并算法，通过提高 R 树合并效率整体上提升 LSM R 树的可用性。

发明内容

本发明的目的是克服现有技术在高频时空数据插入时索引更新效率与查询效率不足，提供一种基于希尔伯特曲线的改进 R 树索引。

本发明所采用的具体技术方案如下：

一种基于希尔伯特曲线的 R 树索引合并更新方法，其步骤如下：

S1：获取待合并的第一 R 树和待插入第一 R 树的第二 R 树，第一 R 树的高度不大于第二 R 树的高度；每棵 R 树中，每个叶子节点中的存储信息包括空间对象的最小外包矩形框和其主键指针，每个非叶子节点中的存储信息包括所有孩子节点的最小外包矩形框、所有孩子节点的主键指针集合、最小外包矩形框中所有空间对象的最大希尔伯特值以及最小外包矩形框中所有空间对象的最小希尔伯特值；

S2：基于每个节点中存储的希尔伯特值信息，按照从根节点层到叶子节点层

的逐层查询顺序，查询得到第一 R 树叶子节点层中每一个被合并叶子节点在第二 R 树叶子节点层中的待插入节点；

S3：针对第一 R 树中每一个叶子节点，将其所包含的空间对象按照希尔伯特值顺序插入至 S2 中确定的待插入节点中，实现两棵 R 树的合并；且合并过程中若出现叶子节点容量溢出，则需对第二 R 树进行节点分裂，并将第一 R 树和第二 R 树包含的所有空间对象按照希尔伯特值顺序重新分配至分裂后的第二 R 树的所有叶子节点中；

S4：针对合并第一 R 树后的第二 R 树，按照从叶子节点到根节点的顺序逐层更新每个节点中的存储信息。

本发明相对于现有技术而言，具有以下有益效果：

本发明通过“从上至下”层级查询算法和“从下至上”层级调整算法，可以实现希尔伯特 R 树的快速合并更新。在合并过程中，每个待插入的节点只会被读取一次，每个节点的写入和分裂也只会出现一次。相较于逐一插入的方式，大幅度的减少了对磁盘读写的次数，提高了希尔伯特 R 树的合并效率，降低了 R 树合并对整体空间索引操作所带来的影响。同时改进希尔伯特 R 树采用批量插入均匀分裂的方式，也可以在一定程度上提高节点的填充率，提高 R 树的空间使用率，因而在查询效率上也能有一定的提升。

附图说明

图 1 是空间希尔伯特填充曲线；

图 2 是改进希尔伯特 R 树结构；

图 3 是实验数据展示图；

图 4 是 R 树与不同数据量 R 树合并时间对比折线图；

图 5 是 R 树空间范围查询效率对比折线图。

具体实施方式

下面结合附图和具体实施方式对本发明做进一步阐述和说明。本发明中各个实施方式的技术特征在没有相互冲突的前提下，均可进行相应组合。

R 树在合并过程中，会占用系统的一定的内存以及计算资源，会影响在此期间对索引进行查询、插入、删除等操作的效率。因此高效的希尔伯特 R 树的合并方法，可以减少上述的情况发生，整体上提升树空间索引的使用效率。本发明中改进了传统的希尔伯特 R 树结构，并设计了基于改进希尔伯特 R 树的合并方

法，实现希尔伯特 R 树的高效合并。

如图 1 所示，在本发明的一个较佳实施例中，提供了一种基于希尔伯特曲线的 R 树索引合并更新方法，该方法的步骤如下：

S1：首先获取两棵 R 树，分别为待合并的第一 R 树和待插入第一 R 树的第二 R 树。为了便于叙述，下面将第一 R 树记为 R1，将第二 R 树记为 R2。由于索引合并都是从低层往高层合并，因此树 R_2 的高度必然大于或等于树 R_1。

本发明中的树 R_1 和树 R_2 均为改进的希尔伯特 R 树，两棵 R 树的基本结构相同。为了统一名词含义，下面定义 R 树的叶子节点为 R 树中没有孩子节点的节点，即最后一层节点；R 树的根节点为 R 树中的第一层节点，即最顶层的节点；R 树中节点的层次为：从根结点开始，设定根结点的层次为第 1 层，根结点的孩子节点的层次为第 2 层，依此类推，如果某一个节点的层次为第 L 层，则其孩子节点的层次为第 L+1 层；R 树的高度为 R 树中所有节点的层次最大值。

在每棵 R 树中，改进的希尔伯特 R 树叶子节点 C_l 与常规的希尔伯特 R 树的叶子节点结构是一致的，每个叶子节点中的存储信息为空间对象的最小外包矩形框和其主键指针，即 C_l=(R, object_id)。

但对于非叶子节点 C_n，两棵改进的希尔伯特 R 树均在原有动态希尔伯特 R 树非叶子节点的结构基础上，额外存储了最小希尔伯特值，因此每个非叶子节点中的存储信息为该节点的所有孩子节点的最小外包矩形框 R、该节点的所有孩子节点的主键指针集合 ptr、最小外包矩形框 R 中所有空间对象的最大希尔伯特值 maxLHV 以及最小外包矩形框 R 中所有空间对象的最小希尔伯特值 minLHV，即 C_n=(R,ptr,maxLHV,minLHV)。

如图 2 所示，为改进的希尔伯特 R 树结构的一个示意图，从图中可见节点 I、节点 II、节点 III 为三个非叶子节点。以节点 II 为例，其中包含三个空间对象，其希尔伯特值分别为 5010，7671 和 13646，所以节点 II 存储的最大希尔伯特值 maxLHV 为 13646，最小希尔伯特值 minLHV 为 5010。根节点包含节点 I、节点 II、节点 III，因此需要在根节点存放的最大希尔伯特值 LHV 为 14033，最小希尔伯特值 minLHV 为 415。

在该改进的希尔伯特 R 树中，通过记录节点下所有空间对象的希尔伯特值范围，可以确定此节点下的空间对象在待插入的希尔伯特 R 树中非叶子节点的路径，而不需要对每一个空间对象的插入都从根节点开始查询。由此，可以减少

磁盘的读写次数，提高合并效率。

本发明改进的希尔伯特 R 树中，针对最小外包矩形的希尔伯特值计算的方式有多种：1）以矩形的中心点作为二维的希尔伯特值计算；2）以左下角坐标和右上角坐标作为四维点的希尔伯特值计算；3）以矩形中心点和中心到矩形边界的距离 d_x、d_y 作为四维点的希尔伯特值计算。不同希尔伯特值计算方式会产生多种聚集情况。

本实施例中，优选以空间对象的最小外包矩形框中心点二维坐标在希尔伯特曲线上的一维编码值作为每个空间对象的希尔伯特值。希尔伯特值得编码过程可采用任意现有技术，为了便于理解下面提供了一种编码过程：

若希尔伯特曲线阶数为 n，空间数据的坐标为 x 和 y

步骤 1：读取最小外包矩形框中心点二维坐标 x 和 y，并将 x 和 y 值转换为长度为 2^n 的二进制值，用 x^B、y^B 表示；

步骤 2：按顺序分别提取 x^B 和 y^B 中的一位二进制值进行组合，得到二进制值段，第 i 组二进制值段记为$x_i^B y_i^B$；

步骤 3：对所有二进制值段进行转换，其中‘00’转换为 00，‘01’ 转换为 01，‘10’ 转换为 11，‘11’ 转换为 10；

步骤 4：将所有转换后的二进制值段重新按顺序组合，得到的完整的二进制值，再转换为十进制值即为希尔伯特值。

另外为了便于理解，下面对希尔伯特 R 树的构建过程进行展开描述，但需要注意的是 R 树的构建过程并非本发明的必要步骤，本发明可用于对已经构建完毕的 R 树进行合并。希尔伯特 R 树构建过程如下：

S11：计算得到每个空间对象的希尔伯特后，将所有空间对象按照希尔伯特值升序排列；

S12：根据 R 树每个节点的容量 c，提取前 c 个空间对象，构造成一个 R 树的第一个叶子节点，再往后提取 c 个空间对象，构造下一个叶子节点，不断重复直至提取完所有空间对象，得到 k 个叶子节点，并将其按照节点构造顺序加入到先进先出队列中；

S13：按照从叶子节点到根节点的顺序逐层构造非叶子节点；且对于每一层非叶子节点而言，从其孩子节点层的先进先出队列中提取前 c 个节点，构造第一个非叶子节点，再从先进先出队列中提取前 c 个节点，构造下一个非叶子节点，

不断重复直至取完先进先出队列中所有节点，得到当前非叶子节点层并将其中的所有非叶子节点按照节点构造顺序加入到新的先进先出队列中，用于作为下一层非叶子节点的构造。

S2：基于每个节点中存储的希尔伯特值信息，按照从根节点层到叶子节点层的逐层查询顺序，查询得到第一 R 树叶子节点层中每一个被合并叶子节点在第二 R 树叶子节点层中的待插入节点。查询过程中，确定待插入节点的原则是：被合并的节点的希尔伯特值范围尽量完全落入被插入的节点的希尔伯特值范围中。

在本实施例中，S2 中的该查询过程采用了一种“从上至下”的层级查询算法，该层级查询算法的具体实现过程如下：

S21：将第一 R 树中当前待查找节点层的层次 n 设定为 1，将第二 R 树中当前待查找节点层的层次 m 设定为 1，第一 R 树的高度为 H_1，第二 R 树的高度为 H_2；第一 R 树和第二 R 树的第 1 层节点层均为根节点，第二 R 树的根节点作为第一 R 树的根节点对应的待插入节点；

S22：针对第一 R 树中的第 n 层节点，分别获取其中每个节点x_i^n的所有孩子节点集合X_i^{n+1}，同时在第二 R 树中获取节点x_i^n对应的待插入节点的所有孩子节点集合Y_i^{n+1}；针对集合X_i^{n+1}中包含的每个节点x_i^{n+1}，根据节点x_i^{n+1}中存储的最大希尔伯特值 $maxLHV_i$ 和最小希尔伯特值 $minLHV_i$，在集合Y_i^{n+1}包含的所有节点中查找最大希尔伯特值大于 $minLHV_i$ 的第一个节点的指针 A，并在集合Y_i^{n+1}包含的所有节点中查找最大希尔伯特值大于 $maxLHV_i$ 的第一个节点的指针 B，以集合Y_i^{n+1}中指针值在[A,B]区间内的节点作为节点x_i^n的待插入节点。

在执行 S22 时，在集合Y_i^{n+1}包含的所有节点中，可分别采用二分查找的方式查找指针 A 和指针 B 对应的节点。

S23：对 n 和 m 的值分别进行加 1，然后重复 S22；

S24：不断重复 S23，直至 n 和 m 取值均为 H_1，得到第一 R 树中的每一个叶子节点在第二 R 树中对应的待插入节点；

S25：不断执行判断过程，若 $m < H_2$，则保持 n 的值不变并对 m 值进行加 1 后重复 S22；直至 $m = H_2$ 后完成查询，使第一 R 树叶子节点层中每一个被合并叶子节点均在第二 R 树叶子节点层中找到对应的待插入节点。

在该层级查询算法中，第一次执行 S22 步骤时对应于查询根节点的待插入节

点。其首先读取 R_1 的根节点，并获得其孩子节点的集合X_i^2，下标 i 代表第 i 个节点，下同。基于改进的希尔伯特 R 树节点特点，集合X_i^2中的节点存储了该节点下空间对象的希尔伯特值范围。针对集合X_i^2中的每个节点x_i^2，根据其最小希尔伯特值 $minLHV_i$ 和最大希尔伯特值 LHV_i，采用二分查找的方式，计算 R_2 树根节点的孩子节点集合Y_i^2中 maxLHV 第一个大于 $minLHV_i$ 的孩子节点指针，记为$fromPage_i$。同样计算 R_2 树根节点的孩子节点Y_i^2中 maxLHV 第一个大于 LHV_i 的孩子节点指针，记为$toPage_i$。由此可知节点x_i^2中的空间对象需要插入的磁盘页位置page满足$fromPage_i \le page \le toPage_i$中，即插入至$fromPage_i$至$toPage_i$范围内的节点中。计算集合$X_i^2$中每个节点的$fromPage_i$和$toPage_i$，此时 R_1 树第二层中的每个节点，在 R_2 树的根节点中找到了插入位置节点的指针。在后续执行 S23 时，根据指针值找到 R_2 树中第二层的节点，再以上述同样的方式找到 R_1 树第三层节点的对应 R_2 树第三层节点的位置。以此方式进行层级查找，直到层级达到为 R_1 树叶子节点对应的空间对象，则 R_1 树不能再往下查找细分，但是 R_2 树在执行 S24 过程中继续往下查找到叶子节点，将对应的 R_1 树中的空间对象批量插入至该叶子节点中。

S3：针对第一 R 树中每一个叶子节点，将其所包含的空间对象按照希尔伯特值顺序插入至 S2 中确定的待插入节点中，实现两棵 R 树的合并。且合并过程中若出现叶子节点容量溢出，则需对第二 R 树进行节点分裂，并将第一 R 树和第二 R 树包含的所有空间对象按照希尔伯特值顺序重新分配至分裂后的第二 R 树的所有叶子节点中。

对第二 R 树进行节点分裂的方法可按照现有技术中的节点索引分裂方式处理，在本实施例中为了便于实现批量、快速的 R 树合并，提供了一种 “从下至上”的层级调整算法，以便于对第二 R 树进行节点索引分裂，其做法如 S31~S33 所示：

S31：计算第一 R 树和第二 R 树包含的所有空间对象总数相对于第二 R 树的节点容量的溢出对象数量；

S32：对第二 R 树的叶子节点进行分裂，创建最少的额外叶子节点用于容纳所有溢出对象；

S33：将第一 R 树和第二 R 树包含的所有空间对象，按照希尔伯特值升序进行排列，然后重新按序分配至第二 R 树分裂后的每个叶子节点中；

S34：按照从叶子节点层朝向根节点层的逐层调整顺序，对第二 R 树的每一层非叶子节点进行节点容量溢出判断，若一层非叶子节点中出现节点容量溢出，则对该层非叶子节点进行节点分裂，并将该层非叶子节点的所有孩子节点重新分配至分裂后的该层非叶子节点中；最终调整完根节点后完成第二 R 树的节点分裂处理。

根据此方法，首先将最后一层中的所有叶子节点实现空间对象数据的插入和分裂，随后向上调整上一层的非叶子节点。随着叶子节点层中的节点分裂，节点数量会增加，可能会导致倒数第二层中的非叶子节点溢出情况。因此随后调整倒数第二层的节点，以相同的方式进行节点分裂，直至调整到根节点。若根节点出现节点容量溢出，则将根节点分裂成两个节点，并以这两个节点为孩子节点重新构建一个根节点。

需要注意的是，两棵 R 树的合并过程中，第一 R 树的空间对象插入第二 R 树的待插入节点时，其应当按照希尔伯特值顺序插入，以保证最终每个叶子节点中的所有空间对象按照希尔伯特值升序排列，所有叶子节点整体也按照最大希尔伯特值升序排列。

S4：由于节点合并、分裂后，其存储的空间对象会发生变化，因此节点中的存储信息会发生变化，因此需要针对合并第一 R 树后的第二 R 树，按照从叶子节点到根节点的顺序逐层更新每个节点中的存储信息。对于叶子节点，需要更新的存储信息包括空间对象的最小外包矩形框 R、主键指针 object_id，对于非叶子节点，需要更新的存储信息包括空间对象的最小外包矩形框 R、主键指针集合 prt、最大希尔伯特值 maxLHV 和最小希尔伯特值 minLHV。

为了进一步说明本发明的技术效果，下面基于一个具体案例来展示上述基于希尔伯特曲线的 R 树索引合并更新方法在具体空间数据上的效果。

实施例：

本实施例步骤与具体实施方式前述步骤相同，即步骤 S1~S4，且 S2 步骤具体采用 S21~S25 实现，S3 步骤具体采用 S31~S34 实现，在此不再进行赘述。下面就部分实施过程和实施结果进行展示：

本实施例采用面状的空间数据类型作为研究对象，数据采用通过 OpenStreetMap 下载的全中国建筑区数据，总共要素条数为 1000192 条。如图 3 所示为研究对象的空间数据。从图中可以看出，空间数据大小不一，且分布不均

匀，数据的疏密程度存在一定的随机性，可以作为实验样本。

随机选取 1000 到 50000 条记录共 50 组数据，对基于希尔伯特曲线的改进 R 树索引、常规 R 树索引和希尔伯特 R 树索引进行更新操作，验证本发明改进希尔伯特 R 树合并算法的高效性，并且对比合并后的查询效率，证实 LH R 树合并算法可以保证较好的空间聚集性和更高的查询效率。

在图 4 中，横轴表示被合并 R 树的大小，纵轴表示 R 树合并所需的时间。在折线图中，从上至下分别为：黄色折线表示为常规 R 树合并所需的时间，灰色折线表示为动态希尔伯特 R 树合并所需的时间，橙色折线表示为静态希尔伯特 R 树合并所需的时间，蓝色折线表示为本发明改进希尔伯特 R 树合并所需的时间。从图中可以看出，动态希尔伯特 R 树的合并方式优于常规 R 树的合并方式，本发明改进希尔伯特 R 树合并方式优于静态希尔伯特 R 树合并方式，本发明改进希尔伯特 R 树合并方式和静态希尔伯特 R 树合并方式所需的时间明显少于逐一插入合并的两种方式。

在图 5 中横轴表示的为搜索矩形的范围，以经纬度作为单位，纵轴表示为对 R 树磁盘读取的次数，单位为万页。在 R 树查询中，当查询的范围较小时，查询速度较快，获得的查询时间误差值较大。由于对磁盘读写的速度远大于内存的查找，因此可以采用磁盘页面读取的次数代替范围查询时间，本实验采用对磁盘读取的次数作为比较查询效率的依据。图中从上至下分别中常规 R 树空间查询效率折线图，动态希尔伯特 R 树空间查询效率折线图、本发明改进希尔伯特 R 树空间查询效率折线图和静态希尔伯特 R 树空间查询折线图。从图中可以看出，四种 R 树空间查询效率随着空间范围的增加变化趋势大体一致。但是动态希尔伯特 R 树查询效率优于常规的 R 树，可以说明希尔伯特 R 树的空间聚集性好于常规的 R 树。采用批量合并算法得到的 R 树查询效率明显优于上述两种 R 树，静态希尔伯特 R 树的效率略优于本发明改进希尔伯特 R 树。但本发明改进希尔伯特 R 树对比 R 树的平均查询效率提升了 52%，对比动态希尔伯特 R 树平均查询效率提升了 41%，对比静态希尔伯特 R 树查询效率降低了 5%。

另外，在其他实施例中，还可以提供一种基于希尔伯特曲线的 R 树索引合并更新装置，其包括存储器和处理器；

所述存储器，用于存储计算机程序；

所述处理器，用于当执行所述计算机程序时，可实现前述的基于希尔伯特曲

线的 R 树索引合并更新方法。

另外，在其他实施例中，还可以提供一种计算机可读存储介质，该存储介质上存储有计算机程序，当所述计算机程序被处理器执行时，可实现前述的基于希尔伯特曲线的 R 树索引合并更新方法。

以上所述的实施例只是本发明的一种较佳的方案，然其并非用以限制本发明。有关技术领域的普通技术人员，在不脱离本发明的精神和范围的情况下，还可以做出各种变化和变型。因此凡采取等同替换或等效变换的方式所获得的技术方案，均落在本发明的保护范围内。

权 利 要 求 书

1．一种基于希尔伯特曲线的R树索引合并更新方法，其特征在于，步骤如下：

S1：获取待合并的第一R树和待插入第一R树的第二R树，第一R树的高度不大于第二R树的高度；每棵R树中，每个叶子节点中的存储信息包括空间对象的最小外包矩形框和其主键指针，每个非叶子节点中的存储信息包括所有孩子节点的最小外包矩形框、所有孩子节点的主键指针集合、最小外包矩形框中所有空间对象的最大希尔伯特值以及最小外包矩形框中所有空间对象的最小希尔伯特值；

S2：基于每个节点中存储的希尔伯特值信息，按照从根节点层到叶子节点层的逐层查询顺序，查询得到第一 R 树叶子节点层中每一个被合并叶子节点在第二R树叶子节点层中的待插入节点；

S3：针对第一 R 树中每一个叶子节点，将其所包含的空间对象按照希尔伯特值顺序插入至S2中确定的待插入节点中，实现两棵R树的合并；且合并过程中若出现叶子节点容量溢出，则需对第二R树进行节点分裂，并将第一R树和第二 R 树包含的所有空间对象按照希尔伯特值顺序重新分配至分裂后的第二 R 树的所有叶子节点中；

S4：针对合并第一R树后的第二R树，按照从叶子节点到根节点的顺序逐层更新每个节点中的存储信息。

2. 如权利要求1所述的基于希尔伯特曲线的R树索引合并更新方法，其特征在于，所述 R 树中，每个空间对象的希尔伯特值为该空间对象的最小外包矩形框中心点二维坐标在希尔伯特曲线上的一维编码值。

3. 如权利要求1所述的基于希尔伯特曲线的R树索引合并更新方法，其特征在于，所述R树的构建过程如下：

S11：将所有空间对象按照希尔伯特值升序排列；

S12：根据R树每个节点的容量c，提取前c个空间对象，构造成一个R树的第一个叶子节点，再往后提取c个空间对象，构造下一个叶子节点，不断重复直至提取完所有空间对象，得到k个叶子节点，并将其按照节点构造顺序加入到

先进先出队列中；

S13：按照从叶子节点到根节点的顺序逐层构造非叶子节点；且对于每一层非叶子节点而言，从其孩子节点层的先进先出队列中提取前 c 个节点，构造第一个非叶子节点，再从先进先出队列中提取前 c 个节点，构造下一个非叶子节点，不断重复直至取完先进先出队列中所有节点，得到当前非叶子节点层并将其中的所有非叶子节点按照节点构造顺序加入到新的先进先出队列中，用于作为下一层非叶子节点的构造。

4. 如权利要求 1 所述的基于希尔伯特曲线的 R 树索引合并更新方法，其特征在于，所述 S2 的具体实现方法为：

S21：将第一 R 树中当前待查找节点层的层次 n 设定为 1，将第二 R 树中当前待查找节点层的层次 m 设定为 1，第一 R 树的高度为 H_1，第二 R 树的高度为 H_2；第一 R 树和第二 R 树的第 1 层节点层均为根节点，第二 R 树的根节点作为第一 R 树的根节点对应的待插入节点；

S22：针对第一 R 树中的第 n 层节点，分别获取其中每个节点x_i^n的所有孩子节点集合X_i^{n+1}，同时在第二 R 树中获取节点x_i^n对应的待插入节点的所有孩子节点集合Y_i^{n+1}；针对集合X_i^{n+1}中包含的每个节点x_i^{n+1}，根据节点x_i^{n+1}中存储的最大希尔伯特值 $maxLHV_i$ 和最小希尔伯特值 $minLHV_i$，在集合Y_i^{n+1}包含的所有节点中查找最大希尔伯特值大于 $minLHV_i$ 的第一个节点的指针 A，并在集合Y_i^{n+1}包含的所有节点中查找最大希尔伯特值大于 $maxLHV_i$ 的第一个节点的指针 B，以集合Y_i^{n+1}中指针值在[A,B]区间内的节点作为节点x_i^n的待插入节点；

S23：对 n 和 m 的值分别进行加 1，然后重复 S22；

S24：不断重复 S23，直至 n 和 m 取值均为 H_1，得到第一 R 树中的每一个叶子节点在第二 R 树中对应的待插入节点；

S25：不断执行判断过程，若 m <H_2，则保持 n 的值不变并对 m 值进行加 1 后重复 S22；直至 m =H_2 后完成查询，使第一 R 树叶子节点层中每一个被合并叶子节点均在第二 R 树叶子节点层中找到对应的待插入节点。

5. 如权利要求 4 所述的基于希尔伯特曲线的 R 树索引合并更新方法，其特征在于，所述 S22 中，在集合Y_i^{n+1}包含的所有节点中，分别采用二分查找的方式查找指针 A 和指针 B 对应的节点。

6. 如权利要求 1 所述的基于希尔伯特曲线的 R 树索引合并更新方法，其特

征在于，所述 S3 中，若两棵 R 树合并过程中出现叶子节点容量溢出，则需按照 S31~S33 对溢出进行节点分裂处理：

S31：计算第一 R 树和第二 R 树包含的所有空间对象总数相对于第二 R 树的节点容量的溢出对象数量；

S32：对第二 R 树的叶子节点进行分裂，创建最少的额外叶子节点用于容纳所有溢出对象；

S33：将第一 R 树和第二 R 树包含的所有空间对象，按照希尔伯特值升序进行排列，然后重新按序分配至第二 R 树分裂后的每个叶子节点中；

S34：按照从叶子节点层朝向根节点层的逐层调整顺序，对第二 R 树的每一层非叶子节点进行节点容量溢出判断，若一层非叶子节点中出现节点容量溢出，则对该层非叶子节点进行节点分裂，并将该层非叶子节点的所有孩子节点重新分配至分裂后的该层非叶子节点中；最终调整完根节点后完成第二 R 树的节点分裂处理。

7. 如权利要求 6 所述的基于希尔伯特曲线的 R 树索引合并更新方法，其特征在于，所述 S34 中，若根节点出现节点容量溢出，则将根节点分裂成两个节点，并以这两个节点为孩子节点重新构建一个根节点。

8. 如权利要求 6 所述的基于希尔伯特曲线的 R 树索引合并更新方法，其特征在于，所述 S3 中，两棵 R 树的合并后，每个叶子节点中的空间对象按照希尔伯特值升序排列。

9．一种基于希尔伯特曲线的 R 树索引合并更新装置，其特征在于，包括存储器和处理器；

所述存储器，用于存储计算机程序；

所述处理器，用于当执行所述计算机程序时，实现如权利要求 1~8 任一项所述的基于希尔伯特曲线的 R 树索引合并更新方法。

10．一种计算机可读存储介质，其特征在于，所述存储介质上存储有计算机程序，当所述计算机程序被处理器执行时，实现如权利要求 1~8 任一项所述的基于希尔伯特曲线的 R 树索引合并更新方法。

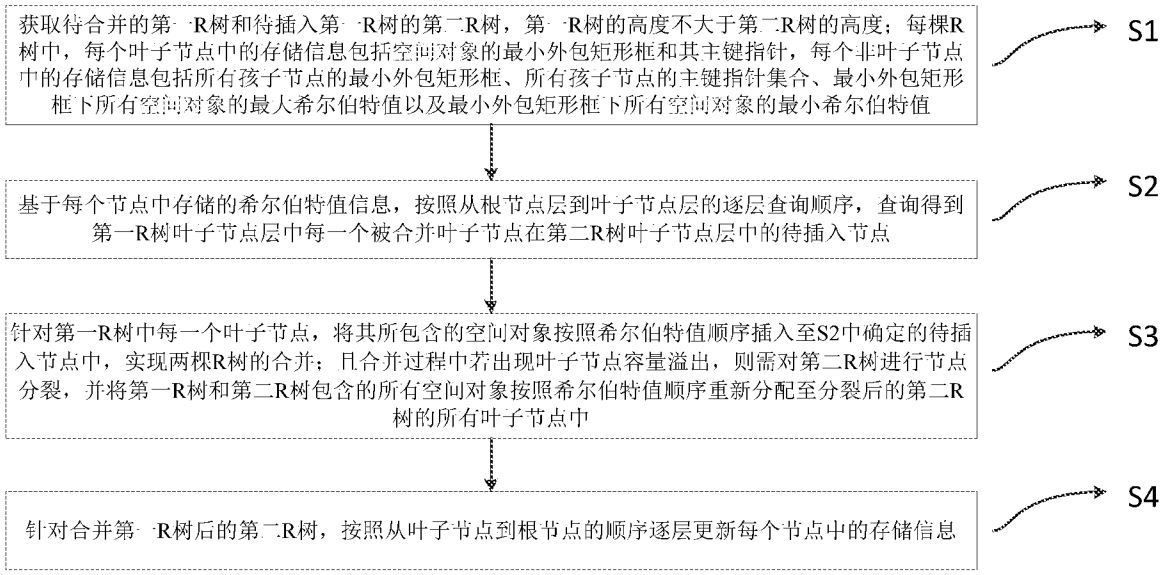

获取待合并的第一R树和待插入第一R树的第二R树，第一R树的高度不大于第二R树的高度；每棵R树中，每个叶子节点中的存储信息包括空间对象的最小外包矩形框和其主键指针，每个非叶子节点中的存储信息包括所有孩子节点的最小外包矩形框、所有孩子节点的主键指针集合、最小外包矩形框下所有空间对象的最大希尔伯特值以及最小外包矩形框下所有空间对象的最小希尔伯特值
S1
基于每个节点中存储的希尔伯特值信息，按照从根节点层到叶子节点层的逐层查询顺序，查询得到第一R树叶子节点层中每一个被合并叶子节点在第二R树叶子节点层中的待插入节点
S2
针对第一R树中每一个叶子节点，将其所包含的空间对象按照希尔伯特值顺序插入至S2中确定的待插入节点中，实现两棵R树的合并；且合并过程中若出现叶子节点容量溢出，则需对第二R树进行节点分裂，并将第一R树和第二R树包含的所有空间对象按照希尔伯特值顺序重新分配至分裂后的第二R树的所有叶子节点中
S3
针对合并第一R树后的第二R树，按照从叶子节点到根节点的顺序逐层更新每个节点中的存储信息
S4

图 1

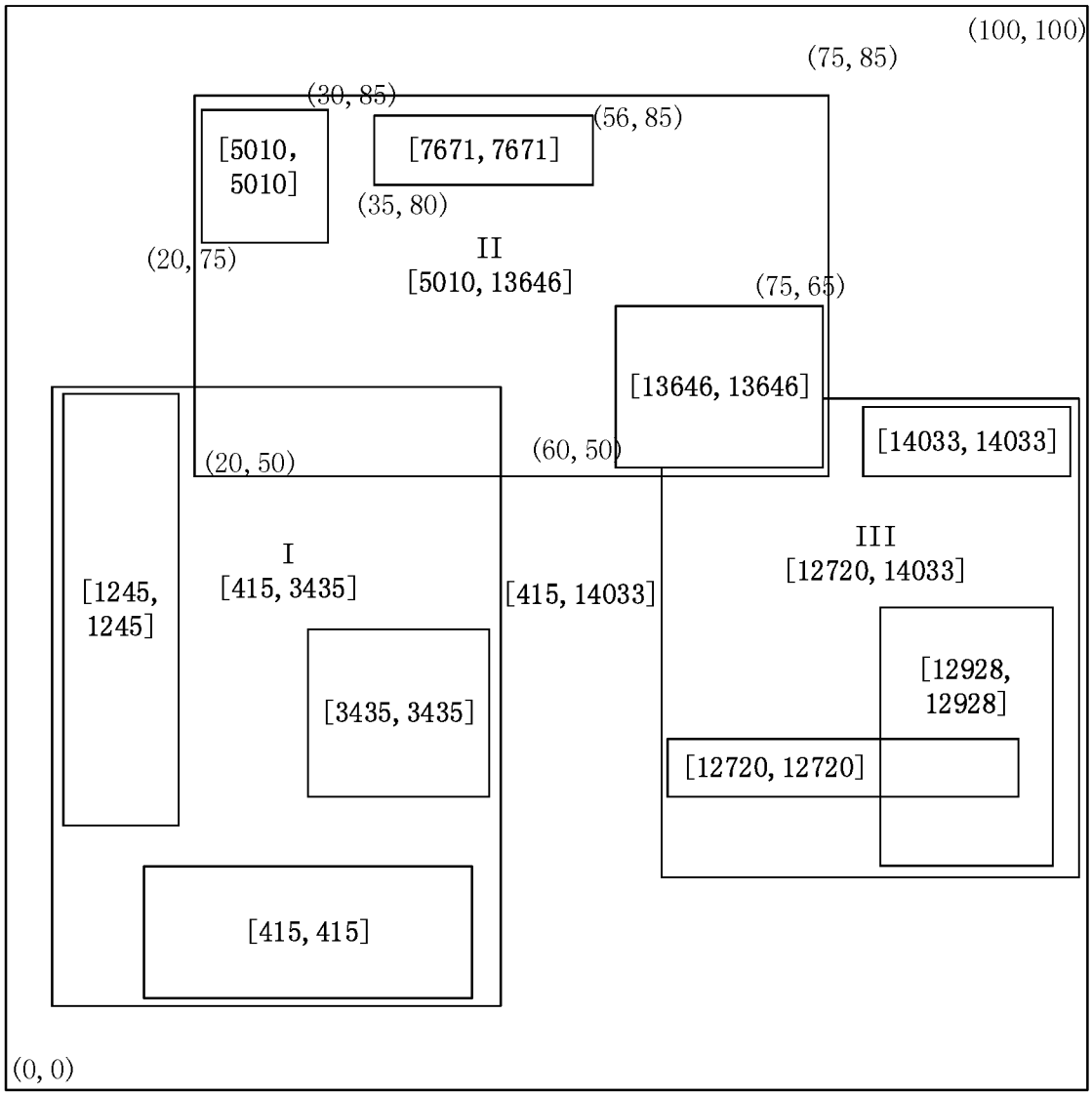

(100, 100)
(75, 85)
(30, 85)
(56, 85)
[5010, 5010]
[7671, 7671]
(35, 80)
(20, 75)
II
[5010, 13646]
(75, 65)
[13646, 13646]
[14033, 14033]
(20, 50)
(60, 50)
[1245, 1245]
I
[415, 3435]
[415, 14033]
III
[12720, 14033]
[12928, 12928]
[3435, 3435]
[12720, 12720]
[415, 415]
(0, 0)

图 2



图 3

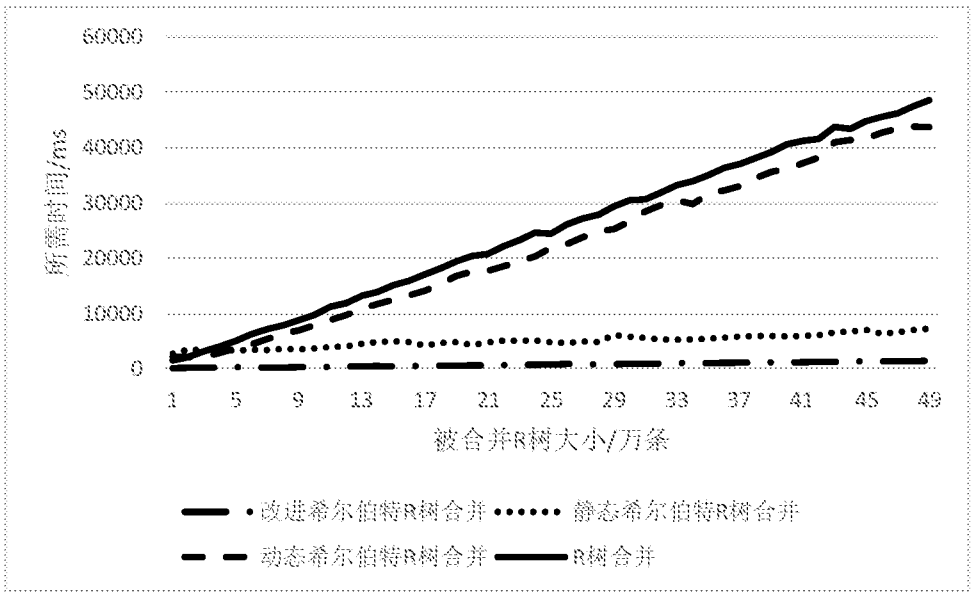

60000
50000
40000
30000
20000
10000
0
所需时间/ms
1
5
9
13
17
21
25
29
33
37
41
45
49
被合并R树大小/万条
改进希尔伯特R树合并
静态希尔伯特R树合并
动态希尔伯特R树合并
R树合并

图 4

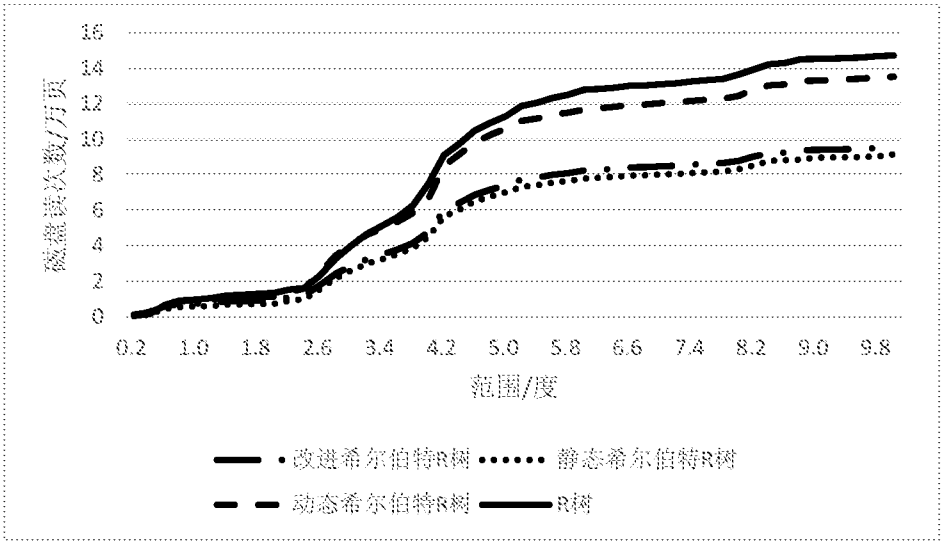

16
14
12
10
8
6
4
2
0
磁盘读次数/万页
0.2
1.0
1.8
2.6
3.4
4.2
5.0
5.8
6.6
7.4
8.2
9.0
9.8
范围/度
改进希尔伯特R树
静态希尔伯特R树
动态希尔伯特R树
R树

图 5

INTERNATIONAL SEARCH REPORT

International application No.
PCT/CN2020/121521

A. CLASSIFICATION OF SUBJECT MATTER

G06F 16/22(2019.01)i

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

G06F

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

CNKI, CNPAT, WPI, EPODOC: 希尔伯特, R树, 合并, 插入, 外包矩形, 包围盒, 节点, 根节点, 叶子, 指针, 高度, 查询, 溢出, 分裂, hilbert, tree, node, query, pointer

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	李勋 (LI, Xun). "基于hilbert划分的并行矢量数据索引算法研究 (Parallel Spatial Index Algorithm Based on Hilbert Partition)" *中国优秀硕士学位论文全文数据库 信息科技辑 (Information Science and Technology, Chinese Master's Theses Full-Text Database)*, No. 01, 15 January 2014 (2014-01-15), pp. 22-23 and 29-37	1-10
A	CN 104731984 A (SHANDONG UNIVERSITY OF TECHNOLOGY) 24 June 2015 (2015-06-24) entire document	1-10
A	CN 104408039 A (NORTHEASTERN UNIVERSITY) 11 March 2015 (2015-03-11) entire document	1-10
A	CN 103617162 A (NANJING UNIVERSITY OF POSTS AND TELECOMMUNICATIONS) 05 March 2014 (2014-03-05) entire document	1-10
A	US 2007198568 A1 (LINKAGE TECHNOLOGY CO., LTD.) 23 August 2007 (2007-08-23) entire document	1-10

☐ Further documents are listed in the continuation of Box C. ☑ See patent family annex.

* Special categories of cited documents:
"A" document defining the general state of the art which is not considered to be of particular relevance
"E" earlier application or patent but published on or after the international filing date
"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
"O" document referring to an oral disclosure, use, exhibition or other means
"P" document published prior to the international filing date but later than the priority date claimed
"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
17 May 2021	**23 June 2021**
Name and mailing address of the ISA/CN **China National Intellectual Property Administration (ISA/CN)** **No. 6, Xitucheng Road, Jimenqiao, Haidian District, Beijing 100088** **China** Facsimile No. **(86-10)62019451**	Authorized officer Telephone No.

INTERNATIONAL SEARCH REPORT
Information on patent family members

International application No.
PCT/CN2020/121521

Patent document cited in search report	Publication date (day/month/year)	Patent family member(s)	Publication date (day/month/year)
CN 104731984 A	24 June 2015	None	
CN 104408039 A	11 March 2015	None	
CN 103617162 A	05 March 2014	None	
US 2007198568 A1	23 August 2007	CN 1838124 A	27 September 2006

国际检索报告

国际申请号
PCT/CN2020/121521

A. 主题的分类

G06F 16/22(2019.01)i

按照国际专利分类(IPC)或者同时按照国家分类和IPC两种分类

B. 检索领域

检索的最低限度文献(标明分类系统和分类号)

G06F

包含在检索领域中的除最低限度文献以外的检索文献

在国际检索时查阅的电子数据库(数据库的名称，和使用的检索词（如使用））

CNKI,CNPAT,WPI,EPODOC:希尔伯特，R树,合并，插入，外包矩形，包围盒，节点，根节点，叶子，指针，高度，查询，溢出，分裂，hilbert，tree，node，query，pointer

C. 相关文件

类 型*	引用文件，必要时，指明相关段落	相关的权利要求
A	李勋. "基于hilbert划分的并行矢量数据索引算法研究" 中国优秀硕士学位论文全文数据库 信息科技辑，第01期，2014年 1月 15日 (2014-01-15)， 第22-23、29-37页	1-10
A	CN 104731984 A (山东理工大学) 2015年 6月 24日 (2015-06-24) 全文	1-10
A	CN 104408039 A (东北大学) 2015年 3月 11日 (2015-03-11) 全文	1-10
A	CN 103617162 A (南京邮电大学) 2014年 3月 5日 (2014-03-05) 全文	1-10
A	US 2007198568 A1 (LINKAGE TECHNOLOGY CO., LTD.) 2007年 8月 23日 (2007-08-23) 全文	1-10

☐ 其余文件在C栏的续页中列出。

☑ 见同族专利附件。

* 引用文件的具体类型：

"A" 认为不特别相关的表示了现有技术一般状态的文件

"E" 在国际申请日的当天或之后公布的在先申请或专利

"L" 可能对优先权要求构成怀疑的文件，或为确定另一篇引用文件的公布日而引用的或者因其他特殊理由而引用的文件（如具体说明的）

"O" 涉及口头公开、使用、展览或其他方式公开的文件

"P" 公布日先于国际申请日但迟于所要求的优先权日的文件

"T" 在申请日或优先权日之后公布，与申请不相抵触，但为了理解发明之理论或原理的在后文件

"X" 特别相关的文件，单独考虑该文件，认定要求保护的发明不是新颖的或不具有创造性

"Y" 特别相关的文件，当该文件与另一篇或者多篇该类文件结合并且这种结合对于本领域技术人员为显而易见时，要求保护的发明不具有创造性

"&" 同族专利的文件

国际检索实际完成的日期	国际检索报告邮寄日期
2021年 5月 17日	2021年 6月 23日
ISA/CN的名称和邮寄地址 中国国家知识产权局(ISA/CN) 中国北京市海淀区蓟门桥西土城路6号 100088 传真号 (86-10)62019451	受权官员 刘宇儒 电话号码 86-(10)-53961294

国际检索报告
关于同族专利的信息

国际申请号
PCT/CN2020/121521

检索报告引用的专利文件	公布日 (年/月/日)	同族专利	公布日 (年/月/日)
CN 104731984 A	2015年 6月 24日	无	
CN 104408039 A	2015年 3月 11日	无	
CN 103617162 A	2014年 3月 5日	无	
US 2007198568 A1	2007年 8月 23日	CN 1838124 A	2006年 9月 27日