

基于多维度融合网格划分的遥感大数据时空查询方法

陈俊哲^{1,2,3}, 刘杰^{1,2,3}, 覃驭楚^{4,5}, 曹宇³, 王帅^{1,3}, 叶丹¹, 钟华¹

1. 中国科学院软件研究所, 北京 100190;
2. 中国科学院大学 南京学院, 南京 211135;
3. 中科南京软件技术研究院, 南京 210000;
4. 可持续发展大数据国际研究中心, 北京 100094;
5. 中国科学院空天信息创新研究院, 北京 100094

摘要: 云平台支撑的遥感大数据在线处理系统为遥感影像的多领域创新应用提供了高效支持。然而, 现有数据库系统在处理百万级遥感影像元数据的时空查询时响应缓慢, 难以满足在线处理系统的时效要求, 复杂的大规模时空查询已经成为遥感影像分析的性能瓶颈。本文提出一种融合二维与三维网格划分的高效时空索引与查询优化方法。该方法首先基于二维网格划分进行初始查询范围匹配, 并在此基础上结合时间条件, 将部分网格单元转换为三维单元, 以提升查询执行效率。递归构建全局网格单元, 并利用Z-order编码将网格单元映射为一维整型索引, 实现时空查询到编码范围查询的转换。同时基于网格的时空局部性设计查询划分机制, 支持查询任务的并行执行。本方法在MongoDB与PostGIS中实现, 并与系统原生索引进行了性能对比实验, 该方法可将查询耗时分别减少93.11%与88.02%。目前该技术方案已集成至可持续发展大数据国际研究中心发布的SDG大数据平台EarthDataMiner系统中并提供服务。

关键词: 遥感大数据, 遥感影像, 时空查询, 时空网格划分, 查询优化, 索引优化, 遥感云计算平台, 时空数据库
中图分类号: TP701/P2

引用格式: 陈俊哲, 刘杰, 覃驭楚, 曹宇, 王帅, 叶丹, 钟华. 2025. 基于多维度融合网格划分的遥感大数据时空查询方法. 遥感学报, 29(12): 3679-3691

Chen J Z, Liu J, Qin Y C, Cao Y, Wang S, Ye D and Zhong H. 2025. A spatiotemporal query method for remote sensing big data based on multi-dimensional fusion grid division. National Remote Sensing Bulletin, 29(12): 3679-3691[DOI:10.11834/jrs.20254256]

1 引言

自1972年Landsat 1卫星开始获取地表遥感影像以来, 世界各国发射的卫星持续采集了全球海量的遥感影像, 全球遥感数据归档量正在持续快速增长, 遥感大数据时代已经到来(李德仁, 2016)。这些数据为地表信息提取提供了丰富的数据来源, 随着以GEE (Google Earth Engine) (Gorelick等, 2017)为代表的遥感云计算平台发展成熟, 遥感大数据挖掘迎来了前所未有的机遇(付东杰等, 2021)。海量遥感数据的快速高效处理有效支撑了农业与粮产(Khanal等, 2020)、土地发展与城市化(Shao等, 2020)、森林覆盖(Negassa等, 2020)、

可持续发展指标(Allen等, 2021)等领域遥感创新应用的不断涌现。

遥感大数据作为一种典型的时空大数据, 其访问过程高度依赖基于时空元数据的查询。因而在大规模遥感数据管理中, 通常将元数据独立提取并依托时空数据库进行管理。一般而言, 遥感影像访问与加载流程可归纳为图1所示的3个步骤: (1) 基于元数据对数据集进行筛选; (2) 针对筛选出的数据划分任务; (3) 将任务分发到计算集群, 各节点相对独立加载数据并执行分析运算。

然而, 查询频次高、代价大的时空元数据检索已成为遥感影像处理中的性能瓶颈。尽管合适的索引结构能够显著提升查询效率, 但是当前主流

收稿日期: 2024-07-17; 预印本: 2025-07-04

基金项目: 国家自然科学基金(编号:42361144884)

第一作者简介: 陈俊哲, 研究方向为分布式软件理论与技术。E-mail: chenjunzhe21@otcaix.iscas.ac.cn

通信作者简介: 刘杰, 研究方向为地球大数据、分布式系统。E-mail: ljie@otcaix.iscas.ac.cn

时空数据库中的时空索引性能均存在较大提升空间：以 PostGIS 为代表的多维树状时空索引结构在海量数据下面临严重的性能退化问题，主要原因用于划分数据的最小外接矩形存在严重重叠，导致树结构变得不合理；以 MongoDB 为代表的网格索引结构具有比多维树状时空索引更优秀的性能表现，但无法对网格划分进行细粒度的控制，在遥感场景下具有很大的性能提升空间。例如，在一组包含 300 万影像记录的遥感数据集上筛选北京地区 2023 年内数据的时空查询中，PostGIS 与 MongoDB 分别需要约 20 s 与 150 s 完成响应，显然无法满足遥感大数据分析任务对时效性的要求。

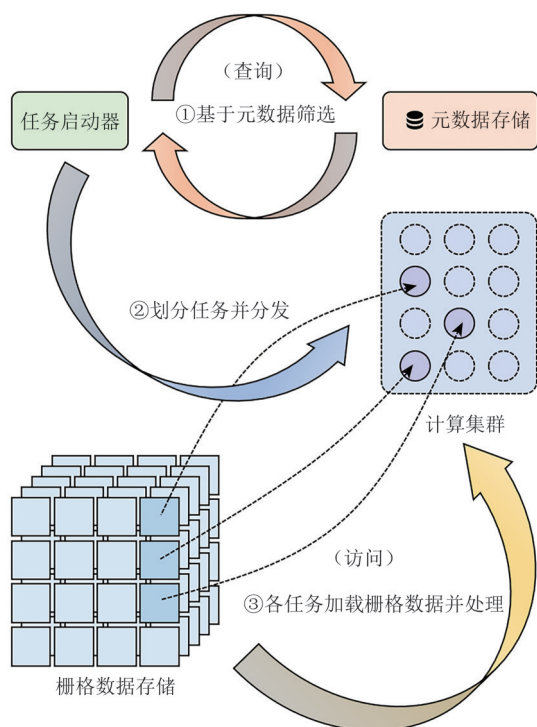


图1 遥感影像数据访问流程

Fig. 1 Data access process for remote-sensing image

面向上述问题，本文提出了一种适用于遥感大数据的多维度融合网格划分时空查询优化方案，主要贡献包括：(1) 提出了一种时空网格划分索引方案，将全球时空范围递归划分成一系列网格单元，并对网格单元编码构建一维索引为时空查询加速提供支持。该索引方案具有易于与现成数据库系统集成的优点；(2) 提出了一种基于多维度融合时空网格划分方案的时空范围查询方法，该方法在二维网格匹配基础上，结合时间约束条件，将部分二维网格单元扩展为三维单元，并统一编码

为一维索引范围，融合了二维与三维索引结构的优势。同时查询方法支持将结果分区加载到不同节点以支持后续对数据的分布式处理。(3) 针对 MongoDB 与 PostGIS 实现了上述方案，并在截止至 2023 年 12 月 1000 万数量级 Landsat 遥感影像元数据集上进行与数据库原生时空索引比对的基准实验中缩减了至多 93.11% 与 88.02% 查询耗时。该技术方案已集成至可持续发展大数据国际研究中心发布的 SDG 大数据平台 EarthDataMiner 系统 (Liu 等, 2020)。

通常而言，针对数据存储与访问特点设计并构建合适的索引结构能够大幅减少查询耗时 (Li 和 Tang, 2012)。现有时空数据索引方法大致可以分为两类：基于多维树状结构的时空索引方案、基于网格划分的时空索引方案。下文将结合典型相关研究工作，综述大规模遥感数据查询优化的主要技术路径。

基于多维树状结构的时空索引方案围绕数据时空属性构建索引，其结构会随着数据的空间分布变化发生显著的调整，在海量数据的场景下存在性能退化的问题。此类方案通常针对时空数据本身构建 R-Tree (Guttman, 1984) 及类似的扩展或变种索引结构。R-Tree 是 B+-Tree 在高维度上的自然扩展，是一种根据最小外接矩形 MBR (Minimal Bounding Rectangle) 递归划分数据组的平衡树形结构，在大多数场景下具有优秀的查询性能。但随着数据量快速增长，R-Tree 内各节点的 MBR 之间开始出现大量重叠覆盖，使得索引维护效率与查询性能显著降低 (Li 和 Tang, 2012)。为解决以上问题，大量基于原始版本 R-Tree 的变种被提出。R*-Tree (Beckmann 等, 1990) 优化了树节点拆分算法，引入节点溢出时的重新插入修正机制，该方法通过额外的索引维护开销换取了更优秀的密集数据查询性能；R+-Tree (Sellis 等, 1987) 在必要时将对象插入到多个叶子节点以避免内部节点 MBR 重叠；此外，Qi 等 (2020) 依据 Hilbert 空间填充曲线网格对数据进行划分，使得生成的 MBR 具有尽可能小的周长，以缓解密集数据场景下的 MBR 重叠；以上研究通过减小 R-Tree 节点 MBR 之间的重叠试图提升查询效率，但并未解决 R-Tree 维护时索引结构调整开销较大的问题，在数据更新频繁时仍然存在性能大幅下降的问题。

目前有一些利用数据集在时空上连续性特征

构建的基于网格划分的时空索引方案，此类方案将整个时空范围划分为一系列网格单元，并面向这些时空网格单元构建索引。基于网格划分的索引方法生成的节点仅与空间划分有关，不随数据量增加。因此，在海量数据的场景下，基于此类方法构建的索引结构具有更小的体积和更小的维护开销。当新增的数据落入非空网格单元或对数据进行更新时，无需对索引结构进行任何调整，因此该类索引方法在频繁变更的海量数据场景下具有出色的性能表现。海量遥感数据存储管理系统通常在关系型数据库中增加数组数据类型的剖分网格编码列，来存储遥感影像元数据中空间信息，对数据进行逻辑剖分索引，从而实现影像数据的统一存储与空间区域检索（李爽等，2016）。RSIMS（Zhou等，2021）与大规模遥感影像调度系统（Zhu等，2023）基于全局空间范围的二维网格划分，并在此划分的基础上使用Hilbert空间填充曲线构建网格索引结构，极大提升了大规模遥感时空数据下的空间查询性能。基于GeoSOT的时空轨迹索引（Qian等，2019）发现在仅考虑空间划分的网格索引方式下，查询时需要引入复合索引，进而导致额外的扫描轮次，增加查询开销。基于以上发现，Qian等引入了时间维度，使用Z空间填充曲线构建了三维网格划分索引方案，在时间与空间范围接近的查询条件下具有更好的性能表现。尽管以上方案很好的解决了复合索引导致的额外开销，但导致查询区域边界处会发生结果假阳性问题，无法严格保证结果正确性。HGST（Liu等，2023）认为基于GeoSOT的时空轨迹索引仍有提升空间，将Z顺序曲线替换空间连续性更优秀的Hilbert曲线，同时引入Spark分布式计算框架进行查询后处理，在解决查询结果正确性问题的同时进一步提升了查询性能。

以上基于二维与三维网格划分的索引方案具有相似的核心思想：首先通过对全局空间或时空范围递归划分构建多级网格，并通过时空填充曲线对产生的网格单元编号进行降维，生成一维标识符；随后，在查询时将输入的时空范围近似匹配为一系列网格单元，利用网格单元的编码生成整型数值范围条件替代时空范围查询条件，以生成更高效率的查询代码。值得注意的是，三维网格索引在规避复合索引开销的同时，引入了更高的网格匹配计算开销（额外时间维度的引入使得计算时间

复杂度从 $O(N^4)$ 提升为 $O(N^8)$ ），在查询范围较大或时空尺度不平衡的情况下具有严重的性能退化，只能进行较低精度的网格匹配，生成不准确的查询语句，通过后处理保证结果正确性。

综上所述，基于网格划分的索引方法有效提升了大规模数据下的时空查询性能。基于二维网格的方法在时间范围较小，空间范围较大的情况下具有更好表现；而基于三维网格的方法在时空范围相近，且对精度要求不高的情况下具有更好的性能。两种索引方法在应用优势上具有互补性，同时在模型构建与查询机制方面也存在一定的共性，为本文开展融合优化提供了可能性。

2 多维度融合网格索引与查询方法

本研究聚焦于融合二维与三维网格划分时空索引方案的各自优势，设计了一种基于融合网格划分的时空索引模型。在索引构建时，使用三维网格划分与编码方法提取遥感影像的时空特征并构建索引；在查询时，将时空查询范围条件转换为一系列网格单元，随后利用网格单元编码生成一系列整形范围查询条件，旨在用更快速地整型范围查询替代部分缓慢的时空范围查询，提升查询效率。

针对传统三维网格索引在匹配精度与计算效率上的不足，本文进一步设计了一种多维融合的网格匹配方法。该方法首先运用二维匹配方法实现较高精度的转换，然后进行三维替换，生成接近三维匹配结果质量的查询代码。此外，本文还基于以上工作提出了一种允许指定粒度的查询分区方案，以支持分布式计算场景下多节点并发查询，将元数据直接加载至各节点以供后续计算，避免集中式查询中数据分发的I/O开销。方法概览如图2所示。

2.1 时空划分与网格单元编码

全局时空网格划分模型是构建基于网格划分时空索引方法中不可或缺的重要环节。本文选择EPSG: 4326作为时空网格划分模型中所有地理空间位置的坐标系参考系统CRS（Coordinate Reference System），任何非此坐标系的输入将被转换至此坐标系下参与运算。在EPSG: 4326坐标系下，地理位置信息以经纬度的形式表示。不同于地理空间坐标，时间属性并没有公认的上下界。因此，本模型采用Unix系统Epoch时间作为时间下

界。时空网格划分模型下的经度范围为 180°W 到 180°E ，纬度范围为 90°S 到 90°N ，时间范围为

1970年1月1日0时0分0秒至2070年1月1日0时0分0秒。

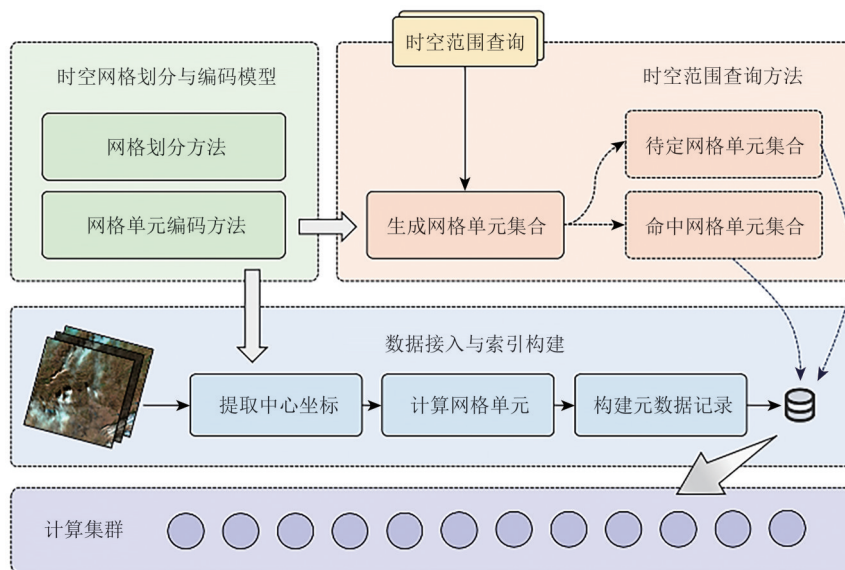
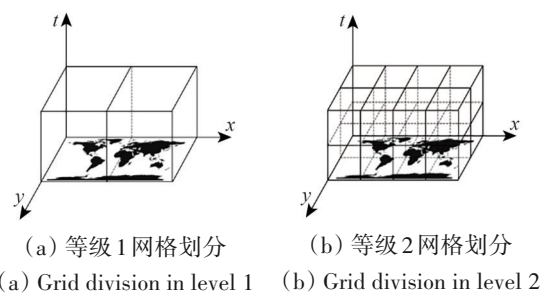


图2 时空查询优化方法概览

Fig. 2 Overview of optimized spatial-temporal query method

本方法对全局时空范围进行递归划分（图3），记网格划分等级为 l ，经度范围上下界分别为 ub_x 、 lb_x ，纬度范围上下界分别为 ub_y 、 lb_y ，时间范围上下界分别为 ub_t 、 lb_t ；则网格单元在经度、纬度、时间3个维度的步长计算公式为 $(sz_x, sz_y, sz_t) = (\frac{ub_x - lb_x}{2^l}, \frac{ub_y - lb_y}{2^{l-1}}, \frac{ub_t - lb_t}{2^l})$ 。记经度、纬度、时间3个维度上网格单元的下标分别为 i, j, k ，全局划分网格单元的起始坐标偏移量计算公式为 $(ofs_x, ofs_y, ofs_t) = (lb_x + i \times sz_x, lb_y + j \times sz_y, lb_t + k \times sz_t)$ 。根据起始坐标偏移量以及各维度步长大小可以计算出网格单元的顶点坐标，从而严格确定网格单元的位置。记经度、纬度与时间3个维度上具有最小值的点为网格单元起始点，其三维坐标计算公式为 $(x_{\min}, y_{\min}, t_{\min}) = (ofs_x, ofs_y, ofs_t)$ ；记经度、纬度与时间3个维度上具有最大值的点为网格单元结束点，其三维坐标计算公式为 $(x_{\max}, y_{\max}, t_{\max}) = (ofs_x + sz_x, ofs_y + sz_y, ofs_t + sz_t)$ 。

本研究旨在融合二维网格划分与三维网格划分的优势，因此采用了一种能够同时保留二维空间信息和三维时空信息的网格单元编码方式，且具备快速进行低维投影的能力。以下对该编码方式进行详细介绍。



(a) 等级1网格划分 (b) 等级2网格划分
(a) Grid division in level 1 (b) Grid division in level 2

图3 全局网格划分示意

Fig. 3 Global grid division

PH-Tree (Zäschke等, 2014) 是一种支持高维索引数据结构，可用于沿各低维分量进行查询。本文受其查询过程启发，发现满足莫顿顺序 (Morton Order) 的Z空间填充曲线可以通过二进制掩码快速构建低维投影（图4）。因此本研究基于64位整型莫顿编码构建三维Z空间填充曲线，通过将3个维度上下标的二进制位交叉排列组合生成编码，并将编码作为网格单元唯一标识，编码的生成过程如图4所示。为了表示最多3个维度的信息，本模型将最高网格划分层级限定为21，每个维度占用21比特，从而最大化利用位空间。本方法将以上流程生成的二进制串以64位整型的方式进行存储，64位整型数据在绝大多数的数据库管理系统中都有支持，具有易于集成且易于计算的优点。

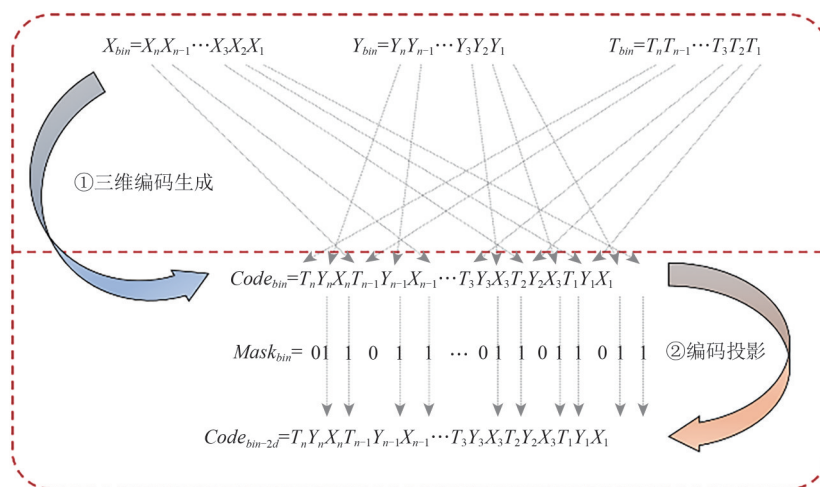


图4 三维莫顿编码的构建二维投影过程

Fig. 4 Procedure of constructing two-dimensional code with three-dimensional morton codec

2.2 索引模型

不同于基于多维树状时空索引结构的方法，本方法采用基于时空网格递归划分的方式构建索引。在本文提出的网格划分方案中，每个网格单元代表一个全局时空范围内唯一的三维时空立方体，该立方体不与任何同层级内的其余单元发生重叠或相交，用于标识特定的时空区域（图3）。遥感影像通常采用瓦片切分的方式存储，因此具有相对固定且规则的时空范围，其时空中心点能够以尽可能低的误差保留其时空信息。本研究在元数据导入时提取其时空中心点，并记录该点所处的网格单元，随后针对该网格单元字段构建B+-Tree索引。该模型下的索引节点数量不会超过网格划分单元的数量，当新插入的数据落入已有数据存在的网格单元时，索引结构不会发生任何改动，十分适合频繁更新的遥感大数据场景。该索引模型下的数据查询过程如下：优先使用网格单元二级索引快速缩减搜索空间，随后使用时空范围索引对部分数据进行扫描，从而大幅减少计算量。该索引模型如图5所示，详细的查询流程将在2.3节中展开介绍。

2.3 时空范围查询方法

优化后的时空查询处理流程旨在通过将时空查询范围转换成一系列时空网格单元，并利用时空网格单元上构建的二级索引快速缩小搜索空间。其中，最主要的工作是将不规则的时空查询范围匹配为时空网格单元集合。由于网格匹配流程的

复杂度与维度数量相关，本研究采用仅利用空间信息的二维匹配过程，以快速达到较高精度。随后，将生成的二维网格单元替换成三维网格单元，以提升生成查询语句的质量。最后，在查询语句生成阶段，允许通过指定分区数量对查询进行拆分。多维度二级网格索引方案下的查询流程可以概括为以下步骤：时空查询范围网格匹配、网格单元集合调整、网格单元三维替换、网格集合分区与查询语句生成，以下将对这些步骤进行详细阐述。

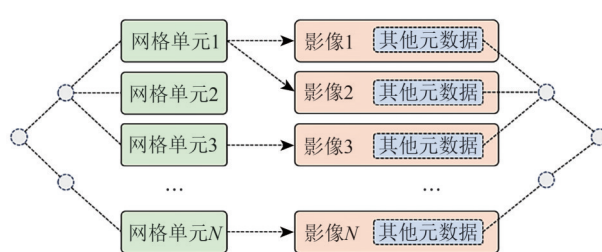


图5 索引模型示意

Fig. 5 Index model

(1) 时空查询范围网格匹配。记 R 为输入的时空查询范围， C 为恰好能容纳 R 的最小网格单元， R （蓝色）与 C （黄色）的空间位置关系如图6所示。查询范围网格匹配具体步骤如下：(1) 将 C 划分为4个子网格单元，得到子单元集合；(2) 对子单元集合进行迭代，判断其中各单元与 R 的空间关系，若子单元被 R 包含则将其加入命中集合，若子单元与 R 相交则将其加入待定集合并重新进行(1)步骤，若子单元与 R 不相交则将其丢弃。

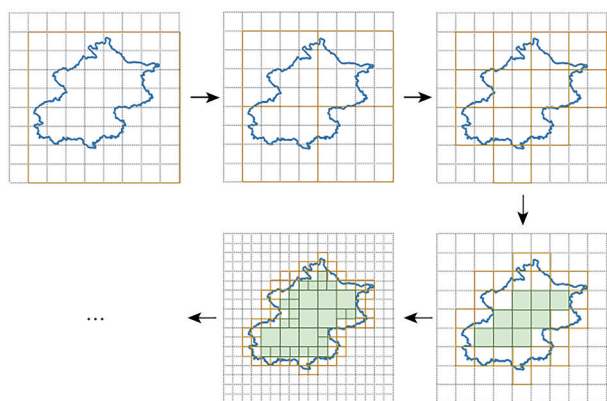


图6 时空查询范围的网格匹配流程

Fig. 6 Procedure of grid matching in spatial-temporal query

时空查询范围网格匹配流程的伪代码描述参见算法1:

算法1 时空范围网格匹配算法

输入: 查询目标范围 R , 初始网格单元 C , 最大层级限制 lv_{max} ;

输出: 命中网格集合 $cells_c$, 待定网格集合 $cells_p$;

```

1  function MATCHTOGRIDCELLS ( $R$ ,  $C$ ,  $lv_{max}$ )
    ▷ 初始化命中集合与待定集合
2  ( $cells_c$ ,  $cells_p$ )  $\leftarrow$  ( $\emptyset$ ,  $\emptyset$ );
3  current_level  $\leftarrow$  C.level;
    ▷ 重复匹配直至  $lv_{max}$ 
4  while current_level < levelmax do
5      sc  $\leftarrow$  CHILDRENOFALLCELLS ( $C$ );
6      contained  $\leftarrow$  { $x$  for  $x$  in sc if  $r.contains(x)$ };
7      intersected  $\leftarrow$  { $x$  for  $x$  in sc - contained if
 $R.intersects(x)$ };
        ▷ 将被包含单元添加至命中集合
8       $cells_c \leftarrow cells_c + contained$ 
        ▷ 将相交单元替换为新的待定集合
9       $cells_p \leftarrow intersected$ ;
10     current_level  $\leftarrow$  current_level + 1;
11  return ( $cells_c$ ,  $cells_p$ );

```

(2) 网格单元集合调整。在部分数据集中, 遥感影像数据可能具有不同的时空范围大小。因此, 时空查询范围匹配过程产生的时空网格单元集合并不能直接用于后续处理流程, 需要针对待定单元进行一些处理才能使用。

由于索引模型仅记录对象中心点, 当数据集中存在大时空体积数据时, 可能出现对象中心所在网格单元与匹配结果集合不相交的情况 (如图7

所示), 导致查询结果正确性的丢失。因此, 本方法需要对待定结果集进行一轮外扩 (图8)。对于执行外扩流程后仍然无法保证查询结果正确性的超大体积对象, 本方法通过将其存入数据库中的单独表或集合中, 依托数据库原生的空间索引进行查询以保证查询结果的正确性。其中, 外扩后无法保证正确性的大对象, 指经、纬、时间3个维度中跨度最大者超过网格单元对应维度尺寸的对象。

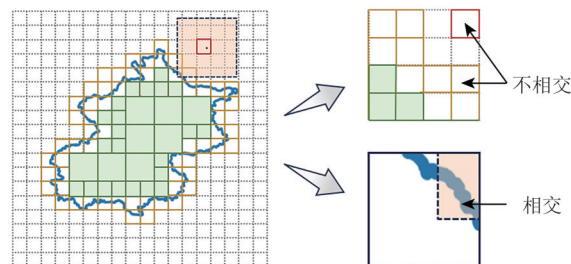


图7 大对象出现时的空间关系误判

Fig. 7 Misjudge of spatial-temporal relation for huge object

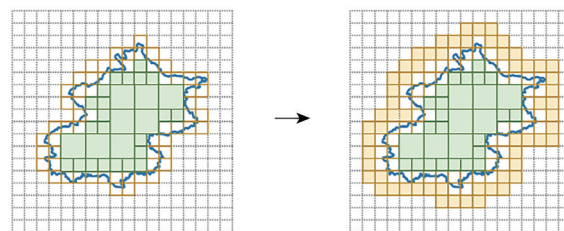


图8 网格集合外扩

Fig. 8 Dilation of grid cell collection

在经过上述外扩流程修正后, 网格单元集合仍然不能直接用于查询语句生成。原因为数量较多的网格单元会导致生成的查询代码中包含冗余的范围条件, 从而影响查询代码的执行效率。因此, 本方法引入了网格单元合并阶段, 以减少集合中网格单元的数量, 提升后续查询语句生成的质量。网格合并具体步骤如下: 1) 遍历集合中的所有网格单元, 对于其父单元的所有子单元 (即兄弟单元) 同时存在于集合中的网格单元, 使用父单元替换所有子单元; 2) 重复步骤 (1), 直至集合中的元素不再发生变动。网格单元合并流程如图9所示。

(3) 网格单元三维替换。在生成可用的二维网格单元集合后, 本方法对二维网格单元集合进行筛选, 选取适合的部分, 结合查询范围的时间条件生成三维网格单元。这里的条件为: 替换网格在时间维度的上下界不允许与查询输入范围有交叉, 是一种“宁缺毋滥”的替换不会在时间维

度引入“假阳性”问题。查询对象3个维度跨度均匀的情况下，三维网格替换能够替换较大范围的网格单元，生成的查询代码更少使用复合索引，相比于二维网格单元生成的查询代码扫描更少的数据，具有更好的执行效率。而在时间维度较为狭小的范围下，替换步骤会退化，但此时时间维度区分度很高，复合索引中的时间维度索引能够很好的缩小扫描范围，性能优于使用三维索引。网格单元三维替换的具体步骤如下：(1) 寻找与时间跨度小于时间范围的最低层级网格单元；(2) 将结果集合中对应层级的二维网格与时间查询范围组合成三维网格单元并进行替换。网格单元三维替换流程描述见算法2，结果如图10所示。

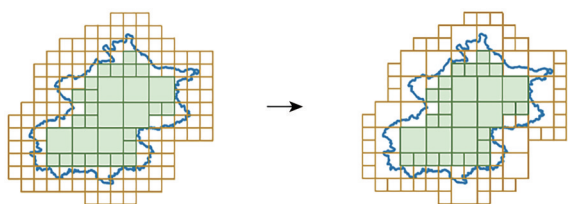


图9 网格单元合并

Fig. 9 Grid cell merging

算法2 网格单元三维替换算法

输入：待替换二维网格集合 Cells，查询范围时间上下界 ($Time_{ub}$, $Time_{lb}$)；

输出：维度替换后二维与三维网格集合 ($Cells_{2d}$, $Cells_{3d}$)；

```

1  function DIMENSIONSUBSTITUTION (Cells,
   Timeub, Timelb)
   ▷输入为空时直接返回，无需替换
2  if Cells.isEmpty ()
3      return (∅, ∅);
   ▷记录 Cells 中最低划分层级
4  min_level ← MINLEVELFROM (Cells);
   ▷记录该层级的时间步长
5  time_size ← TIMESIZEOF (min_level);
6  min_tid ← (Timelb - GLOBAL_TIME_LB) /
time_size;
7  max_tid ← (Timeub - GLOBAL_TIME_LB /
time_size;
8  ( $Cells_{2d}$ ,  $Cells_{3d}$ ) ← (∅, ∅);
   ▷遍历所有二维网格单元
9  for cell in Cells do
   ▷处于目标层级，进行替换
10     if cell.level = min_level

```

```

11         for tid in RANGE (min_tid, max_tid)
do
12             Cells3d.add (Cell (cell.xid, cell.
yid, tid))
   ▷非目标层级，不进行替换
13     else
14         Cells2d.add (cell);
15 return (Cells2d, Cells3d);

```

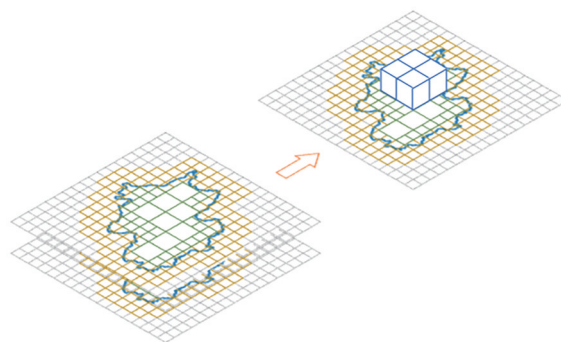


图10 三维网格单元替换

Fig. 10 Three-dimensional substitution of grid cells

(4) 网格集合分区。通过以上步骤处理，可以得到一个保留查询范围时空信息的网格子单元集合。大规模遥感影像数据范围查询的结果中通常包含大量的记录，不适合单节点后续处理。因此本方法引入了查询代码分区阶段，将网格子单元集合划分成指定数量的分区 (图11)，从而支持将查询结果发送到不同的计算节点进行后续分布式处理。网格集合分区算法的步骤描述如下：(1) 寻找合适的分割掩膜单元集合；(2) 将待分区集合内部的网格单元继续划分，直至掩膜单元所在等级；(3) 按照掩膜单元的边界对待分区集合进行划分。

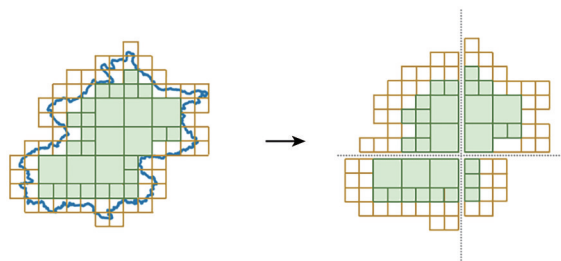


图11 网格单元集合分区

Fig. 11 Partition of grid cell collection

(5) 查询语句生成。经过上述处理后，可以得到一组经过分区且高度优化的网格划分子单元集合，包含若干三维网格单元与二维网格单元，集

合中的每个网格单元具有唯一的 (*Code*, *Level*) 二元组形式, 其中 *Code* 代表网格单元编码, *Level* 为单元所处的网格划分等级。通过算法 3 可以将每个二元组转换为一系列连续的最高层级网格子单元编码, 进而生成一个整形范围条件, 用于生成查询代码。

算法 3 编码查询范围生成算法

输入: 网格单元集合 *Cells*, 最大划分层级 lv_{max} ;

输出: 用于组装查询语句条件的编码范围集合 *CodeRanges*;

```

1  function GENERATECODERANGE (Cells,
 $lv_{max}$ )
2  CodeRanges ← ∅;
   ▷ 对每个单元生成编码整型范围
4  for cell in cells do
5  lower_bound ← cell.code << ( $lv_{max}$  - cell.level) * 3;
6  mask ←  $1 << (lv_{max} - cell.level) * 3 - 1$ ;
7  code_range ← RANGE (lower_bound,
lower_bound + mask);
8  CodeRanges.add (code_range);
9  return CodeRanges;

```

该算法利用莫顿编码的共享前缀特性, 通过在低位填充全 0 与全 1 生成指定子层级的网格集合范围, 如图 12 所示。查询条件的构建方式根据网格单元类型略有不同: 对于三维命中网格单元集合, 可以直接使用编码范围生成查询条件; 对于三维待定网格单元集合, 需要与查询时空范围条件共同组成约束生成查询条件; 对于二维命中网格单元集合, 使用编码范围与时间约束共同生成查询条件; 对于二维待定网格单元集合, 需要与时空约束共同组成查询条件。

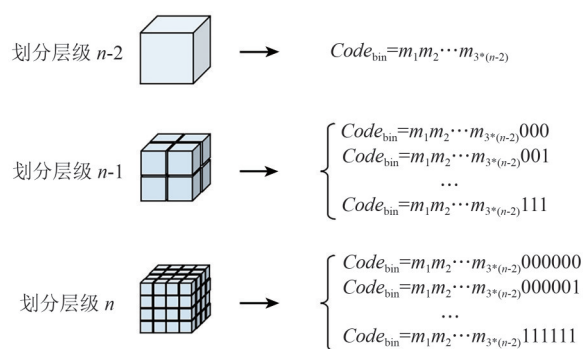


图 12 编码查询范围生成

Fig. 12 Generation of code query range

3 性能评估

本节将对本文提出的时空查询优化方法进行实验验证与结果分析。实验基于典型公开遥感影像数据集, 评估所提出方法在大规模时空范围查询中性能提升效果。首先, 构造覆盖不同数量级的时空范围查询案例, 统计整体查询响应时间, 验证本方法的可行性以及提升效果; 其次, 针对所提出方法中的各项关键优化策略, 分别分析其对性能提升的具体贡献。

3.1 实验设计

(1) 实验数据集。实验采用 Landsat 系列卫星的遥感影像元数据作为查询目标集合。Landsat 系列卫星是美国航空航天局 (NASA) 与美国地质调查局 (USGS) 共同管理的一组卫星。自 1972 年第首颗 Landsat 卫星升空, 该系列卫星持续收集对地观测遥感影像数据, 提供了世界上最长的连续空基观测记录。Landsat 卫星数据具有较为丰富的元数据, 且具有较为完整的地理空间覆盖与时间范围覆盖。

本实验使用的数据集来源于 USGS 提供的 Landsat 卫星元数据服务 (U.S. Geological Survey, 2024), 详细信息参见表 1。其中 Landsat-MSS-C2-L1 数据集包含了 1972 年—1999 年 Landsat 系列卫星收集的多光谱扫描仪 MSS (Multi-Spectral Scanner) 数据; Landsat-TM-C2-L1 数据集包含了 1982 年—2012 年 Landsat 系列卫星收集的专题制图仪 TM (Thematic Mapper) 数据; Landsat-ETM-C2-L1 数据集包含了 1999 年—2023 年 Landsat 系列卫星收集的增强主题制图仪 ETM+ (Enhanced Thematic Mapper Plus) 数据; Landsat-OT-C2-L1 数据集包含了 2013 年—2023 年 Landsat 卫星收集的地表成像仪 OLI (Operational Land Imager) 与热红外传感器 TIRS (Thermal Infrared Sensor) 数据。

表 1 Landsat 数据集信息

Table 1 Landsat dataset details

数据集名称	采集年限	影像数量
Landsat-MSS-C2-L1	1972—1999	1842292
Landsat-TM-C2-L1	1982—2012	2923623
Landsat-ETM-C2-L1	1999—2023	3355632
Landsat-OT-C2-L1	2013—2023	3271868

(2) 查询案例。为了验证本文提出的大规模遥感影像元数据时空范围查询优化方法的性能提

升效果，实验设计了4个具备代表性的地理空间范围： R_1 （17，356 km²）、 R_2 （99，660 km²）、 R_3 （1，142，367 km²）、 R_4 （9，652，782 km²），对应的时间查询范围如表2所示。实验在此基础上构造不同时空组合的查询任务，用于评估方法在多尺度范围下的适应性与查询效率。

表2 查询时空范围

Table 2 The query temporal ranges

范围名称	起始时间	结束时间
T_n	—	—
T_0	1983-01-01	2023-01-01
T_1	2013-01-01	2023-01-01
T_2	2020-01-01	2023-01-01

（3）实验环境与设置。实验在3台Linux服务器节点组成的数据库集群上进行，其软硬件环境配置见表3。中的4个数据集被导入同一个文档集合中，针对地理空间信息属性构建2dSphere索引。

表3 实验软硬件环境配置

Table 3 Experimental software and hardware configuration

环境	详细信息
操作系统	CentOS 7 (Linux Kernel 3.10)
CPU	32核
内存/GB	32
网络带宽/Mbps	1000
MongoDB	v6.0.1 (分片集群部署)

本实验主要评估本文提出的时空查询优化方案在以上查询案例上的执行性能表现，以及各优化策略对性能的具体影响。考虑到网络请求波动以及实验执行期间各类干扰因素，实验均重复运行8次，使用四分位数范围（IQR）方法剔除异常值，计算平均耗时作为结果。

3.2 实验结果与分析

本研究在MongoDB的2dSphere索引和PostGIS的GiST索引上均实现了所提出的方法。由于所提方法的优化效果与数据库内部实现无关，且两者实验中表现结果相近，后续分析选取性能更优的MongoDB作为基线，集中探讨所提方法的有效性与性能表现。

（1）时空范围查询性能评估。为了验证基于二级网格索引的遥感影像元数据查询优化方法在不同查询案例下的性能表现，本实验根据3.4.1节

中的时空范围组合构造了一系列时空范围查询请求。分别使用MongoDB内置的2dSphere索引、二维网格索引、三维网格索引以及本文提出的多维度匹配网格索引方法执行以上查询对比验证性能。实验结果如表4所示。

表4 不同索引的时空查询耗时

Table 4 Spatiotemporal query performance of different indexes

查询范围	2dSphere/ms	2d Grid/ms	3d Grid/ms	本文 方法/ms	耗时变化/%
T_1+R_1	397	453	1230	106	-73.05
T_1+R_2	751	905	3175	469	-37.55
T_1+R_3	3957	2830	6291	812	-79.50
T_1+R_4	135210	32374	65571	9317	-93.11
T_2+R_1	130	449	286	65	-50.00
T_2+R_2	259	312	523	245	-5.41
T_2+R_3	1550	1325	4664	407	-73.74
T_2+R_4	51449	18182	22878	6820	-86.74
T_3+R_1	50	54	105	57	+14.00
T_3+R_2	109	119	182	95	-12.84
T_3+R_3	471	1060	932	384	-18.47
T_3+R_4	15669	21696	322972	3552	-77.33
$+R_1$	405	516	2089	114	-71.85
T_n+R_2	804	982	5998	189	-76.49
T_n+R_3	3967	2176	35259	763	-80.77
T_n+R_4	83921	28940	74687	8535	-89.83

与3.2节中的分析相符合，三维网格索引在较为复杂的大范围查询下产生了过于复杂的条件，导致数据库优化器并不能很好的对其优化，查询性能开销甚至比2dSphere更差。图13中给出了除三维网格索引外的查询耗时的可视化对比，由于不同方法的查询执行开销具有较大的出入，时间坐标轴以对数坐标的方式呈现，耗时单位为毫秒。实验结果表明，本文提出的大规模时空查询优化方案在大部分查询案例上相较于基线（MongoDB的2dSphere索引）、二维网格索引以及三维网格索引均有明显性能提升。值得注意的是，该方案在较小的时空查询范围下提升不大。相反，网格匹配流程的引入可能导致额外开销，从而使得少部分小范围查询案例下与基线方法相比性能稍有倒退。但小查询范围小的性能倒退问题可以通过设置启发式规则来避免。综合而言，本文提出的优化方法对于大规模时空范围查询具有较为显著的性能提升。

(2) 时空范围查询结果正确性评估。3.3 节中讨论了本方法中的空间关系误判情况, 为验证方法针对该情况的正确性修正策略有效性, 本文进行了时空范围查询正确性评估实验。实验结果如表 5 所示, 其中查询结果项数量预期值由 MongoDB

的 2dSphere 索引查询结果项数数量确定。实验结果中各时空查询区域下的结果项数量与预期值严格一致, 证明了本方法提出的查询结果正确性修正策略有效性。

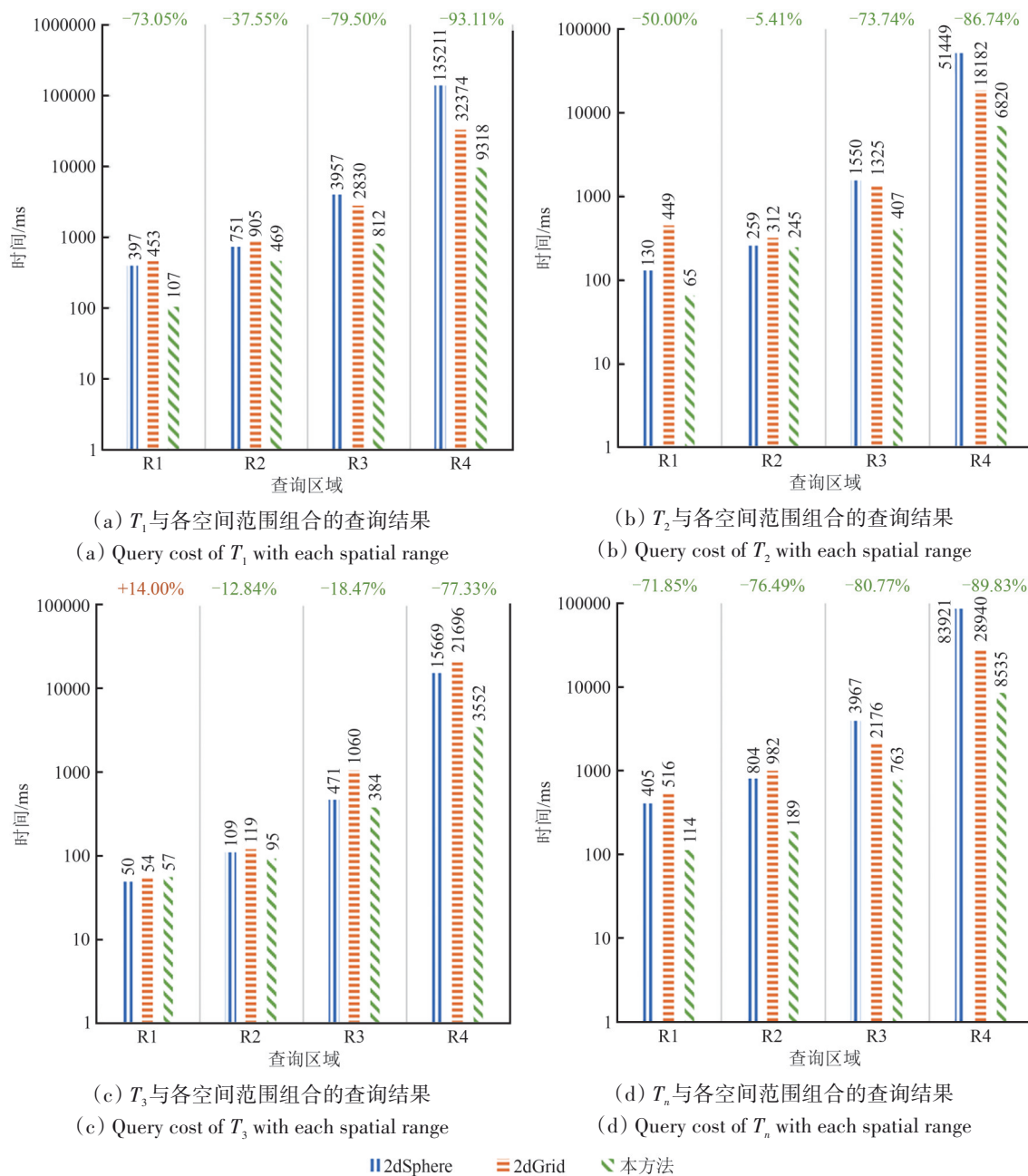


图 13 不同时空范围下的查询耗时

Fig. 13 Query cost for each spatial-temporal range

(3) 网格合并对性能的影响。在所提出的时空范围查询方法中, 本文引入了一种网格单元合并算法, 以提升生成的查询语句质量。为了评估网格合并策略对查询性能的影响, 从 3.1 节中构造的时空查询范围中选取了若干具有代表性的案例,

对比了网格合并策略融入前后的时空查询响应时间差异, 结果见表 6。

在 T_2+R_1 与 T_2+R_2 两组较小范围案例中, 生成的网格单元数量较少, 合并优化空间较小, 合并算法的引入并未带来性能提升, 同时也未引入明显额外

开销；在 T_2+R_3 以及 T_2+R_4 两组中大范围案例下，算法能够有效将参与查询语句生成的网格单元数量减少，其中 T_2+R_3 案例中网格数量缩减了 32.81%，查询响应时间减少了 19.03%， T_2+R_4 案例中网格数量缩减了 40.80%，查询响应时间减少了 10.99%。结果表明，在网格单元数量较多的情况下，网格合并算法能够有效的减少参与查询语句生成的网格单元数量，并相应缩减查询响应时间。

表5 各时空查询返回结果项数量

Table 5 Result items of different spatial-temporal query

范围名称	结果项预期值	查询结果项
T_1+R_1	11496	11496
T_1+R_2	21818	21818
T_1+R_3	82509	82509
T_1+R_4	699742	699742
T_2+R_1	3737	3737
T_2+R_2	7419	7419
T_2+R_3	40173	40173
T_2+R_4	281501	281501
T_3+R_1	1111	1111
T_3+R_2	2240	2240
T_3+R_3	12241	12241
T_3+R_4	85296	85296
T_n+R_1	12789	12789
T_n+R_2	24862	24862
T_n+R_3	89824	89824
T_n+R_4	767331	767331

表6 网格合并算法对开销的影响

Table 6 Effect of grid merging algorithm

时空范围	合并前/后网格数	合并前/后开销/ms
T_2+R_1	64/64	102/101
T_2+R_2	34/34	144/146
T_2+R_3	128/86	599/485
T_2+R_4	603/357	5615/4998

(4) 维度替换对性能的影响。本方法提出了一种网格单元维度替换策略，旨在尽可能使用三维网格对二维网格进行替换，从而提升查询效率，以下实验对三维网格与二维网格生成的查询代码执行效率进行了对比和分析，为维度替换策略提供依据。实验构造了若干以 90°N ， 45°E ，2020-01-01T12:00:00 为起点的不同规模查询区域，分别使用二维网格编码配合时间复合索引的查询策略与三维网格编码单字段索引的策略生成查询语句，并在实验环境上执行，结果见表 7。

表7 维度替换对整体开销的影响

Table 7 Effect of dimension substitution

时空范围	二维开销/ms	三维开销/ms	耗时缩减/%
$45^{\circ}\times 45^{\circ}\times 315576000\text{ s}$	24.24	1.58	93.48
$22.5^{\circ}\times 22.5^{\circ}\times 157788000\text{ s}$	1.58	1.08	31.65
$11.25^{\circ}\times 11.25^{\circ}\times 78894000\text{ s}$	1.05	1.04	0.95
$5.625^{\circ}\times 5.625^{\circ}\times 39447000\text{ s}$	1.05	1.03	1.91

实验结果表明对于较大的时空查询范围，三维网格生成的代码查询效率远高于二维网格生成的查询代码效率。但值得注意的是，在较小的时空范围下，二者性能差距不大。该结论与本方法中对较大的二维网格进行三维替换的策略相符合。

4 结 论

本文针对现有数据库管理系统在大规模复杂时空查询场景下性能不足、难以满足遥感大数据近实时处理需求的问题，提出了一种基于多维度融合网格划分的时空查询加速方案，并在 MongoDB 与 PostGIS 上完成实现与验证。

实验表明，本文提出的技术方案在 MongoDB 上的复杂时空范围查询案例耗时减少最多可达到 93.11%，在 PostGIS 上的复杂时空查询案例耗时减少最多可达到 88.02%，显著提升了大规模复杂时空查询的性能表现。需要指出的是，在小规模遥感数据时空查询的任务中，该技术方案可能存在一定的性能退化，原因为网格索引边缘区域正确性无法保证，需要额外的请求修正结果，多次请求的通信开销在此时反而大于计算开销。对此，可以根据启发式规则直接在小规模查询时使用原生索引，回避性能退化问题。后续针对小规模时空范围查询任务的进一步优化可考虑从数据库存储引擎层面入手。同时，本方案亦适用于其他大规模时空检索场景，如轨迹数据查询。

总体而言，本文提出的多维度融合网格划分查询加速方案在提升大规模复杂时空查询性能的同时，具备良好的分布式友好性与系统集成性，已在可持续发展大数据国际研究中心发布的 SDG 大数据平台 EarthDataMiner 系统 (Liu 等, 2020) 中获得实际部署应用。

参考文献 (References)

Allen C, Smith M, Rabiee M and Dahmm H. 2021. A review of scien-

- tific advancements in datasets derived from big data for monitoring the sustainable development goals. *Sustainability Science*, 16(5): 1701-1716 [DOI: 10.1007/s11625-021-00982-3]
- Beckmann N, Kriegel H P, Schneider R and Seeger B. 1990. The R*-Tree: an efficient and robust access method for points and rectangles. *ACM SIGMOD Record*, 19(2): 322-331 [DOI: 10.1145/93605.98741]
- Fu D J, Xiao H, Su F Z, Zhou C H, Dong J W, Zeng Y L, Yan K, Li S W, Wu J, Wu W Z and Yan F Q. 2021. Remote sensing cloud computing platform development and Earth science application. *National Remote Sensing Bulletin*, 25(1): 220-230 (付东杰, 肖寒, 苏奋振, 周成虎, 董金玮, 曾也鲁, 闫凯, 李世卫, 吴进, 吴文周, 颜凤芹. 2021. 遥感云计算平台发展及地球科学应用. *遥感学报*, 25(1): 220-230) [DOI: 10.11834/jrs.20210447]
- Gorelick N, Hancher M, Dixon M, Ilyushchenko S, Thau D and Moore R. 2017. Google earth engine: planetary-scale geospatial analysis for everyone. *Remote Sensing of Environment*, 202: 18-27 [DOI: 10.1016/j.rse.2017.06.031]
- Guttman A. 1984. R-Trees: a dynamic index structure for spatial searching//*Proceedings of the 1984 ACM SIGMOD International Conference on Management of Data*. Boston: ACM: 47-57 [DOI: 10.1145/602259.602266]
- Khanal S, KC K, Fulton J P, Shearer S and Ozkan E. 2020. Remote sensing in agriculture—accomplishments, limitations, and opportunities. *Remote Sensing*, 12(22): 3783 [DOI: 10.3390/rs12223783]
- Li D R. 2016. Towards geo-spatial information science in big data era. *Acta Geodaetica et Cartographica Sinica*, 45(4): 379-384 (李德仁. 2016. 展望大数据时代的地球空间信息学. *测绘学报*, 45(4): 379-384) [DOI: 10.11947/j.AGCS.2016.20160057]
- Li G B and Tang J E. 2012. A new R-Tree spatial index based on space grid coordinate division//*Proceedings of 2011 International Conference on Informatics, Cybernetics, and Computer Engineering*. Melbourne: Springer: 133-140 [DOI: 10.1007/978-3-642-25188-7_16]
- Li S, Cheng C Q, Tong X C, Chen B and Zhai W X. 2016. A study on data storage and management for massive remote sensing data based on multi-level grid model. *Acta Geodaetica et Cartographica Sinica*, 45(S1): 106-114 (李爽, 程承旗, 童晓冲, 陈波, 翟卫欣. 2016. 基于多级信息网格的海量遥感数据存储管理研究. *测绘学报*, 45(S1): 106-114) [DOI: 10.11947/j.AGCS.2016.F013]
- Liu H, Yan J N, Wang J L, Chen B, Chen M and Huang X H. 2023. HGST: a hilbert-GeoSOT spatio-temporal meshing and coding method for efficient spatio-temporal range query on massive trajectory data. *ISPRS International Journal of Geo-Information*, 12(3): 113 [DOI: 10.3390/ijgi12030113]
- Liu J, Wang W and Zhong H. 2020. EarthDataMiner: a cloud-based big earth data intelligence analysis platform. *IOP Conference Series: Earth and Environmental Science*, 509(1): 012032 [DOI: 10.1088/1755-1315/509/1/012032]
- Negassa M D, Mallie D T and Gemeda D O. 2020. Forest cover change detection using Geographic Information Systems and remote sensing techniques: a spatio-temporal study on Komto Protected forest priority area, East Wollega Zone, Ethiopia. *Environmental Systems Research*, 9(1): 1 [DOI: 10.1186/s40068-020-0163-z]
- Qi J Z, Tao Y F, Chang Y C and Zhang R. 2020. Packing R-Trees with space-filling curves: theoretical optimality, empirical efficiency, and bulk-loading parallelizability. *ACM Transactions on Database Systems*, 45(3): 14 [DOI: 10.1145/3397506]
- Qian C Y, Yi C, Cheng C Q, Pu G L, Wei X F and Zhang H C. 2019. GeoSOT-based spatiotemporal index of massive trajectory data. *ISPRS International Journal of Geo-Information*, 8(6): 284 [DOI: 10.3390/ijgi8060284]
- Sellis T K, Roussopoulos N and Faloutsos C. 1987. The R+-tree: a dynamic index for multi-dimensional objects//*Proceedings of the 13th International Conference on Very Large Data Bases*. San Francisco: ACM: 507-518
- Shao Z F, Ding L, Li D R, Altan O, Huq M E and Li C M. 2020. Exploring the relationship between urbanization and ecological environment using remote sensing images and statistical data: a case study in the Yangtze River Delta, China. *Sustainability*, 12(14): 5620 [DOI: 10.3390/su12145620]
- U.S. Geological Survey. Bulk metadata service. <https://www.usgs.gov/landsat-missions/bulk-metadata-service> [2024-04-15]
- Zäschke T, Zimmerli C and Norrie M C. 2014. The PH-tree: a space-efficient storage structure and multi-dimensional index//*Proceedings of 2014 ACM SIGMOD International Conference on Management of Data*. Snowbird: ACM: 397-408 [DOI: 10.1145/2588555.2588564]
- Zhou X H, Wang X Z, Zhou Y C, Lin Q H, Zhao J H and Meng X H. 2021. RSIMS: large-scale heterogeneous remote sensing images management system. *Remote Sensing*, 13(9): 1815 [DOI: 10.3390/rs13091815]
- Zhu J K, Zhang Z, Zhao F, Su Z, Gu Z N and Wang L L. 2023. Efficient management and scheduling of massive remote sensing image datasets. *ISPRS International Journal of Geo-Information*, 12(5): 199 [DOI: 10.3390/ijgi12050199]

A spatiotemporal query method for remote sensing big data based on multi-dimensional fusion grid division

CHEN Junzhe^{1,2,3}, LIU Jie^{1,2,3}, QIN Yuchu^{4,5}, CAO Yu³, WANG Shuai^{1,3}, YE Dan¹, ZHONG Hua¹

1. Institute of Software of Chinese Academy of Sciences, Beijing 100190, China;

2. University of Chinese Academy of Sciences, Nanjing 211135, China;

3. Nanjing Institute of Software Technology, Nanjing 210000, China;

4. The International Research Center of Big Data for Sustainable Development Goals, Beijing 100094, China;

5. Aerospace Information Research Institute of Chinese Academy of Sciences, Beijing 100094, China

Abstract: With the rapid development of large-scale remote sensing data acquisition technologies, an enormous volume of Earth observation data are generated every day. Online remote sensing analysis platforms, such as Google Earth Engine and Sentinel Hub, have significantly lowered the barrier to conducting large-scale environmental and geospatial studies. Consequently, remote sensing data have been widely applied in agricultural and grain yield monitoring, land-use planning, environmental change detection, forest coverage assessment, and oceanographic research. These applications have provided essential technical support for sustainable development, climate monitoring, and natural resource management. However, as the scale, resolution, and update frequency of remote sensing metadata continue to grow, conventional database systems are increasingly unable to meet the timeliness and responsiveness required for interactive online analysis. A typical spatiotemporal range query over million-level or even billion-level metadata records may take tens of seconds to execute, which severely limits user experience, disrupts data analysis workflows, and limits real-time visualization capabilities. Large-scale and complex range queries have thus become a critical performance bottleneck in current remote sensing analysis systems.

To address these challenges, we propose an efficient spatiotemporal query optimization method based on the fusion of two-dimensional (2D) and three-dimensional (3D) grid indexing. The proposed method first performs spatial range matching by employing an adaptive 2D grid partitioning strategy that divides the spatial domain into high-precision grid cells. According to the temporal constraints in the query, selected 2D cells are dynamically converted into 3D grid units, forming a hybrid 2D-3D hierarchical index. This design allows the query to focus on a significantly smaller subset of the data space, thereby reducing computational complexity while maintaining accuracy. Once the hierarchical grid structure has been constructed, an efficient encoding scheme is required to enable fast indexing and range searches across the entire spatiotemporal domain. All grid units are therefore encoded using Z-order space-filling curves, which map multidimensional coordinates into one-dimensional integer values. As a result, spatiotemporal queries can be efficiently transformed into range queries over these encoded keys, which are naturally supported by most NoSQL and spatial databases. Furthermore, a query partitioning and scheduling mechanism is developed based on the spatiotemporal locality of grid cells. This mechanism supports parallel and distributed query execution, enabling the method to scale efficiently with increasing data volumes and computing capacity.

We implemented the proposed method on both MongoDB and PostGIS platforms to evaluate its generality and performance. Experimental results demonstrate that the optimized index structure and query partitioning mechanism significantly reduce query time—by up to 93.11% on MongoDB and 88.02% on PostGIS—compared with their native indexing approaches. The proposed method effectively improves system responsiveness, enabling near-real-time spatiotemporal query processing even on million-scale datasets.

In summary, this work provides a practical and high-performance solution for accelerating spatiotemporal queries in large-scale remote sensing analysis systems. The method is lightweight, easy to integrate into existing database frameworks, and adaptable to distributed and cloud-based environments. It provides not only technical insights into optimizing spatiotemporal database performance but also a reference framework for future applications involving intelligent remote sensing data management, multi-source data fusion, and dynamic environmental monitoring.

Key words: remote sensing big data, remote sensing imagery, spatiotemporal query, spatiotemporal grid partitioning, indexing optimization, spatiotemporal locality, remote sensing cloud computing platform, spatiotemporal database

Supported by National Natural Science Foundation of China (No. 42361144884)