



引文格式:刘泽邦,陈萃,杨岸然,等.支持大规模地理矢量数据实时可视化的高效索引技术[J].武汉大学学报(信息科学版), 2023, 48(9):1512-1521. DOI:10.13203/j.whugis20210445

Citation: LIU Zebang, CHEN Luo, YANG Anran, et al. Efficient Indexing Technology for Real-Time Visualization of Large-Scale Geographic Vector Data[J]. Geomatics and Information Science of Wuhan University, 2023, 48(9): 1512-1521. DOI: 10.13203/j.whugis20210445

支持大规模地理矢量数据实时可视化的高效索引技术

刘泽邦¹ 陈萃¹ 杨岸然¹ 马梦宇¹ 曹竞之¹

¹ 国防科技大学电子科学学院, 湖南 长沙, 410073

摘要:对大规模地理矢量数据进行可视化是当前地理信息科学领域研究的热点与难点,在当前研究中,显示导向型计算方法具有对数据规模不敏感的特性,可支持大规模地理矢量数据实时可视化。但随着数据规模增长,该方法在数据预处理时索引构建时间和生成索引尺寸急剧增长,这大幅影响方法的实用性。基于此,提出一种基于显示导向型计算的高效索引技术:在预处理阶段,基于四叉树递归划分全球地理空间范围快速构建瓦片四叉(tile-quad, TQ)树索引,在TQ树中通过编码实现结点和瓦片/像素空间范围的对齐;在可视化阶段,以屏幕显示的像素点为计算单元,判断TQ树中像素对应结点是否存在,可快速计算像素值来生成最终的显示效果。实验结果表明,该技术在数据预处理时的索引构建时间短,索引尺寸小,同时可视化效率远优于现有显示导向可视化方法,可更快地支持亿级矢量要素的实时可视化。

关键词:大规模地理矢量数据;数据可视化;索引技术;显示导向计算;实时计算

中图分类号:P208

文献标识码:A

收稿日期:2021-09-27

DOI:10.13203/j.whugis20210445

文章编号:1671-8860(2023)09-1512-10

Efficient Indexing Technology for Real-Time Visualization of Large-Scale Geographic Vector Data

LIU Zebang¹ CHEN Luo¹ YANG Anran¹ MA Mengyu¹ CAO Jingzhi¹

¹ College of Electronic Science and Technology, National University of Defense Technology, Changsha 410073, China

Abstract: Objectives: Real-time visualization of large-scale geographic vector data is a hot and difficult topic in the field of geographic information science. In the current research, display-driven computing method (DisDC) is insensitive to data scale and can support real-time visualization of large-scale geographic vector data. However, with the growth of data scale, the index construction time and the index size of the DisDC visualization method greatly increase in data preprocessing, which greatly affects the practicality of the method. **Methods:** To fill the gap, a fast indexing technique based on DisDC is proposed. Rapid construction of tile-quadtrees (TQ-trees) based on quadtree recursive division of global geographic range in the pre-processing stage, in TQ-tree, the alignment of nodes and tiles/pixels are realized by encoding. In the visualization stage, according to the process of DisDC, the pixel is taken as the calculation unit to determine whether the corresponding node of the pixel in TQ-tree exists, and the pixel value can be quickly calculated to generate the final display effect. **Results:** Experimental results show that the proposed technique has shorter index construction time and smaller index size, and the visualization efficiency outperforms the existing DisDC visualization methods. **Conclusions:** The Method can support real-time visualization of multi-billion vector elements more quickly.

基金项目:国家自然科学基金(41971362);湖南省自然科学基金(2019JJ50718);中国人民解放军国防科技大学青年科学基金(42101432)。

第一作者:刘泽邦,博士生,研究方向为高性能地理计算。liuzebang19@nudt.edu.cn

通讯作者:陈萃,博士,教授。luochen@nudt.edu.cn

Key words: large-scale geographic vector data; data visualization; index technology; display-driven computing; real-time computing

在地理信息系统(geographic information system, GIS)领域,随着国家基础测绘、对地观测系统、位置服务系统、社交网络及物联网等空间数据产源技术的发展与进步,空间数据的规模呈现爆炸式的增长^[1]。地理矢量数据作为空间数据的重要组成部分^[2],在城市规划、土地利用、气象预测等领域发挥着重要作用。数据可视化可以让用户对未知的数据集有更为直观的理解和认识^[3],对地理矢量数据进行可视化是对地理矢量数据进行挖掘分析的第一步,也是对地理矢量数据进行下一步复杂空间分析的基础。在大数据时代,如何对大规模地理矢量数据进行快速的可视化是当前研究面临的一个严峻挑战^[4]。

1 相关研究现状

1.1 矢量数据可视化方法

大规模地理矢量数据的可视化是空间大数据领域中的一个重要的分析方法,为了更好地在互联网中展现空间大数据,标准的网络地图瓦片服务(web tile map service, WTMS)被提出^[5]。该服务通过将空间数据的可视化结果按照瓦片金字塔^[6]的结构进行组织,可以实现在浏览器端多尺度浏览大规模地理矢量数据。但当数据规模很大时,该方法存在着瓦片生成耗时长、瓦片样式不灵活等问题^[7]。随着计算机技术的发展,许多研究引入高性能计算技术,通过并行计算减少可视化瓦片的绘制时间^[8-12],同时出现了HadoopViz^[13]、GeoSparkViz^[14]等一系列性能较好的大规模地理矢量可视化平台,相比传统串行方法在性能上有明显的提升。但在上述地理矢量可视化方法中,均采用基于数据导向型计算的方式,随着数据规模的增加,即使采用并行技术也难以达到实时的性能^[15]。在前期的研究中已提出显示导向型计算框架^[15]和基于显示导向计算的可视化方法^[4,16],将用于屏幕显示的像素点作为独立的计算单元,按照像素点空间范围对矢量数据进行空间检索,得到影响像素显示值的矢量对象,从而生成像素值来生成最终的显示结果,该方法具有对数据不敏感的特性,其可视化性能远强于HadoopViz、GeoSparkViz等平台,非常适用于大规模地理矢量可视化。

1.2 空间索引技术

在显示导向型计算中,需对矢量数据进行转换使其具备快速检索能力,但随着数据规模激增,当前显示导向型可视化方法在数据预处理时索引构建时间和生成索引尺寸急剧增长,无法同时兼顾数据组织和可视化的效率。空间索引技术是解决海量数据快速检索、查询和访问的重要手段^[17],当数据规模激增时,传统空间索引如格网索引^[18]、四叉树索引^[19]、R树索引^[20]难以维持良好性能。而分布式环境下的索引技术成为研究热点:文献[21]提出基于Hadoop的HQ树空间数据索引。在HQ树中使用线性四叉树来解决数据插入顺序和空间重叠导致的并行处理效率低的问题;文献[22]通过希尔伯特编码将空间数据集进行排序,并采用并行计算框架MapReduce构建索引,对R树进行批量加载实现了希尔伯特R树的并行构建;文献[23]提出一个动态分布式数据结构DD-R树索引,DD-R树构建了两个层次的R树,插入数据时先由主结点进行缓存,当数据量达到缓冲区容量,则将缓冲区的数据批量写入相应的结点中。分布式索引设计时多侧重于提升对数据的检索效率,欠缺对索引构建效率和索引尺寸方面的考虑^[1]:一方面增加了集群内部结点间的通信开销,另一方面索引机制必须充分考虑分布式系统的整体框架和数据的组织存储方式,索引构建的开销较大。基于此,为兼顾数据预处理和可视化效率,本文提出一种支持矢量数据实时可视化的高效索引技术。

2 显示导向型计算模型

对大规模地理矢量数据进行可视化,通常方式是将矢量数据转换为便于屏幕显示的栅格对象,即矢量栅格化。采用传统数据导向的计算方式,是以矢量要素为计算单元,需要将屏幕范围内的矢量要素逐一栅格化,再将所有结果合并生成最终的可视化结果。当屏幕范围内的矢量数据规模较大时,其可视化处理的计算量大,即使采取了高性能计算技术也难以支持大规模地理矢量数据的快速可视化^[16]。

显示导向型计算模型(display-driven computing, DisDC)是一个非常适合于GIS中数据密集型计算问题的计算模型^[15],如图1所示。

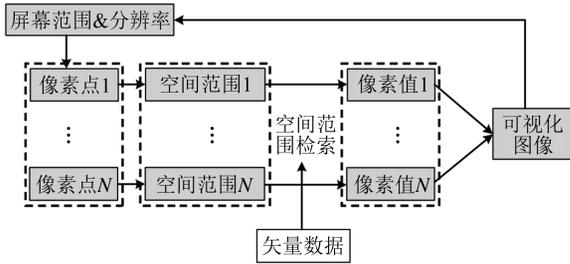


图1 基于显示导向型计算的矢量数据可视化流程
Fig. 1 VectorData Visualization Process Based on DisDC

不同于数据导向型方法,在 DisDC 中,计算单元不再是矢量要素,而是以像素点为最基本的计算单元,根据用户浏览数据时屏幕显示的区域及分辨率确定待计算的像素点范围,检索出决定像素值的空间对象,进而生成像素值,最终将可视化结果的屏幕显示效果快速生成。最终的可视化结果通过瓦片金字塔进行组织,实现在浏览器端对数据的多尺度浏览。DisDC 的计算复杂度仅依赖屏幕显示的像素数量,具有对数据规模不敏感的特性,其在大规模地理矢量数据的可视化分析上有着广阔的应用前景。

在前期工作中,科研人员已将 DisDC 成功应用并解决 GIS 中一些基础的空间分析问题,并提出 HiVision^[4]、HiBuffer^[24]、HiAccess^[25] 用于实现大规模矢量数据的交互式的实时的可视化、空间缓冲区分析、空间可达性分析,这些方法在处理

大规模地理矢量数据时的计算效率远高于传统方法,且均能实现实时可视化的效果,验证了 DisDC 相比传统数据导向方法具有巨大的优势。在 DisDC 中,为支持快速空间范围检索操作,需要对数据集进行转换使其具备应对快速检索的能力。然而在现有显示导向型可视化方法 HiVision 中,仅通过 R 树实现对数据的组织,当数据急剧增长时,导致数据预处理阶段非常耗时且生成索引尺寸大,严重影响其性能和实用性。

3 TQ 树索引结构设计

在 DisDC 中,矢量数据的可视化结果通过瓦片金字塔进行组织。瓦片金字塔是一种多分辨率多层次的图片组织模型^[6],在瓦片金字塔第 0 层,通过一张瓦片在全球范围内对整个可视化结果进行显示,随着层级的升高,瓦片数量逐渐增加,可视化结果的分辨率也依次升高。通常瓦片金字塔中相邻级别比例尺关系为 1:2,瓦片总数量关系为 1:4。瓦片金字塔中每个瓦片具有相同且规律的像素规模,即 256×256 像素。在 DisDC 可视化流程中即以瓦片金字塔中瓦片的像素点为计算单元,计算像素值生成最后的可视化结果。

基于 DisDC 的计算特点,本文设计了适应 DisDC 的 TQ 树索引结构,如图 2 所示。

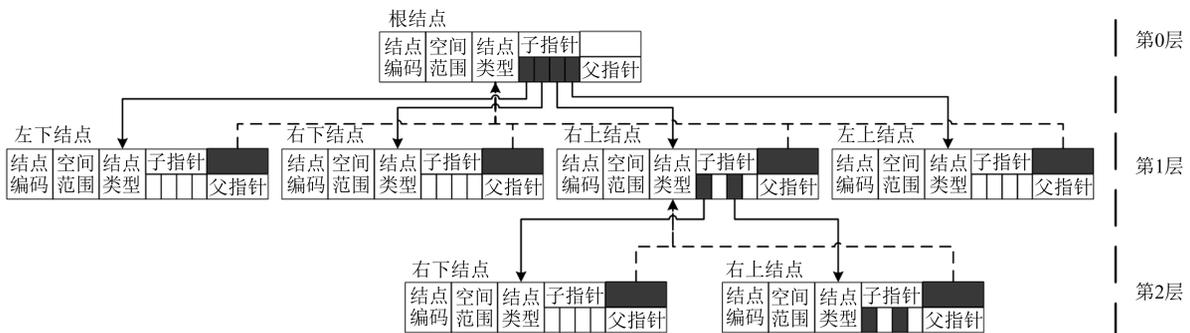


图2 TQ 树索引结构设计
Fig. 2 Structure Design of TQ-Tree Index

由图 2 可知,TQ 树以结点作为最小存储单元,结点分为 5 种类型:根结点、左下结点、右下结点、右上结点、左上结点。每个结点中记录编码,空间范围、结点类型、子结点指针、父结点指针等属性信息,父子结点之间通过指针相连,当子结点或父结点不存在时指针为空。不同于四叉树中将数据集的最小外包矩形(minimum bounding rectangle, MBR)作为根结点空间范围属性,TQ

树将球面墨卡托投影下的全球空间范围作为 TQ 树根结点的空间范围,实现根结点与瓦片金字塔 0 级瓦片的对齐,再依次不断递归四分设置为子结点的空间范围,递归的同时通过 Geohash 编码方法对结点进行编码。Geohash 是常用的地址编码方法,通过二分把二维的空间位置编码成一个二进制字符串^[26],如图 3 所示。

由图 3 可知,采用 Geohash 的划分方式对将

全球空间范围递归编码, L 表示赤道周长的一半。从 TQ 树根结点向下编码: 每向下一层, 将“00”设置为左下结点的编码, “01”设置为左上结点的编码, “10”设置为右下结点的编码, “11”设置为右上结点的编码。由此 TQ 树中每个结点具有唯一的编号, 并且父子结点的包含关系通过编码得以体现。

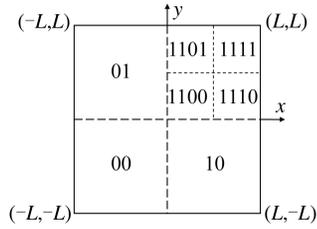


图 3 TQ 树结点编码方式

Fig. 3 Coding Method of TQ-Tree Nodes

TQ 树的设计中实现了结点与瓦片/像素空间范围的对齐: TQ 树中根结点空间范围与瓦片金字塔第 0 层级瓦片范围保持一致, 且两者均采用递归四分的方式, 所以 TQ 树中 n 层级中结点的空间范围与瓦片金字塔中 n 层级中对应瓦片空间范围一致, 并且由于瓦片规格为规则的 256×256 像素, 瓦片像素点空间范围可认为将该瓦片空间范围再递归向下划分 8 次得到, 所以从 TQ 树中 8 层级开始, n 层级中结点空间范围不仅与瓦片金字塔中 n 层级中对应瓦片空间范围一致, 而且也与瓦片金字塔中 $n-8$ 层级中瓦片的像素点空间范围一致。

4 基于 TQ 树的显示导向型可视化方法

大规模矢量数据的可视化实现的核心是高效地确定像素值, 即如何快速检索与像素值有空间关系的矢量要素。这需对数据集进行转换, 要求索引满足以下两点: 第一, 要实现索引的快速构建, 减少数据预处理时间, 同时减小索引尺寸; 第二, 要具备响应快速计算像素值的能力, 可支持像素点与矢量元素空间关系的快速判断。而 TQ 树结构可满足上述需求, 本文首先设计 TQ 树构建方法, 实现矢量数据集的快速转换和索引尺寸的减小, 并根据构建好的 TQ 树设计矢量数据快速可视化方法, 通过判断 TQ 树中相应结点是否存在即可生成像素值, 进一步提升了显示导向型可视化的可视化效率。同时设计并行计算框架实现矢量数据的实时可视化。

4.1 基于矢量数据的 TQ 树构建方法

TQ 树构建方法采用四叉树递归划分方法, 采取自上而下的方式构建 TQ 树, 创建根结点, 因为 TQ 树均从全球空间范围开始向下索引, 所以基于不同数据集构建的 TQ 树的根结点是一致的, 不同数据集所构建 TQ 树从根结点向下的树结构则不同。其构建过程主要分为 3 步:

1) 创建根结点并设置属性, 其 4 个子结点指针值为空; 同时设置 TQ 树的最大层级 n 。

2) 开始插入矢量对象, 将根结点空间范围均分为 4 个象限, 当象限与矢量对象存在空间包含或相交关系时, 则向下创建新结点。按此法从新结点向下递归创建新结点直至新结点的层级等于 n 。当所有矢量对象判断完毕, 根结点同所有新创建结点组成 TQ 树。

3) 将 TQ 树中所有结点属性信息存储到磁盘空间中, 将索引输出至外存。

在判断象限与矢量对象的空间关系时, 将矢量对象拆分为细粒度项进行判断^[4,16], 对于点要素, 将点对象作为细粒度项; 对于线要素, 将其每个线段作为细粒度项; 对于面要素, 将其每条边作为细粒度项。同时在判断时首先利用细粒度项的 MBR 进行粗筛, 当满足条件时再具体判断细粒度项与空间范围是否真实相交或包含。

对矢量点数据构建 TQ 树流程如图 4 所示。对矢量线/面数据构建 TQ 树流程如图 5 所示。

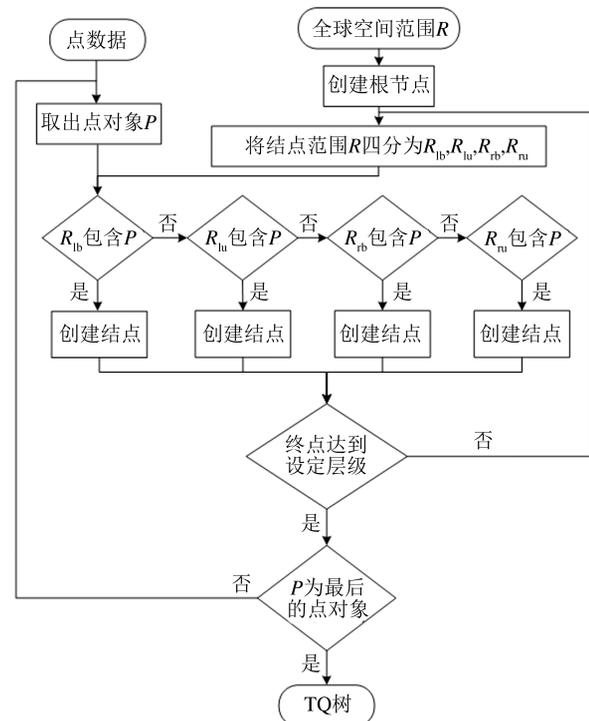


图 4 矢量点数据的 TQ 树构建流程

Fig. 4 TQ-Tree Building Process of Vector Point Data

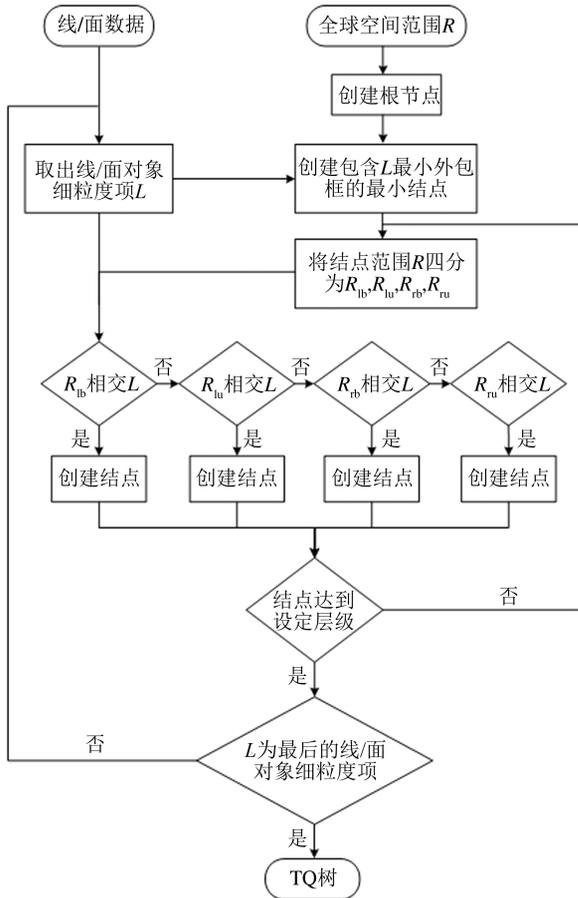


图5 矢量线/面数据的TQ树构建流程
 Fig. 5 TQ-Tree Building Process of Vector Line/Polygon Data

当插入点对象时,按步骤从根结点向下递归创建新结点,当插入线/面对象时,先根据一个线/面对象的MBR从根结点递归向下创建至空间范围包含该MBR的最小结点,再按步骤根据该线/面对象的细粒度项从该结点递归向下创建新结点。这样无需每次判断均从根结点向下递归,以缩短索引构建时间。

TQ树构建采用四叉树递归可快速实现索引构建,所得索引为非满四叉树,且结点中无需储存原始矢量数据,索引的尺寸大大减小。同时索引中结点具有特殊含义:即结点对应瓦片/像素空间范围与矢量要素存在空间包含或相交关系,这可为后续快速可视化提供支持。

4.2 基于TQ树索引的快速可视化方法

在可视化过程中,为了更好的显示效果,矢量要素在屏幕上被绘制为有一定像素宽度的空间对象。采用显示导向的计算方式,以像素为计算单元,需要判断像素点对应空间位置是否在矢量要素一定像素宽度范围内,即判断以像素点为中心的一定像素宽度范围内是否与矢量要素存

在空间包含或相交关系。该判断是规律的并只需关注“有没有矢量要素”。

在TQ树结构中结点与瓦片/像素空间范围是对齐的,且已构建的TQ树实质类似于数据集的瓦片/像素索引,索引中存在的结点代表该结点对应瓦片/像素空间范围与矢量要素存在空间包含或相交关系,即满足“有没有矢量要素”条件。基于此特点设计基于TQ树的可视化流程如图6所示。

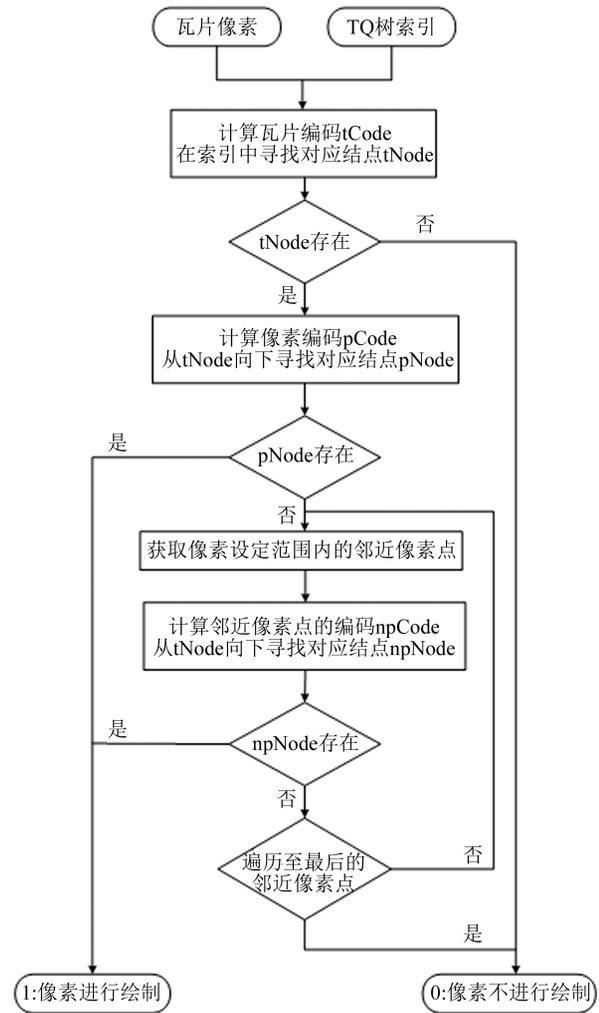


图6 基于TQ树的可视化流程
 Fig. 6 Process of TQ-Tree Based Visualization

首先绘制可视化瓦片时,判断瓦片对应结点在索引中是否存在以此决定瓦片是否绘制,若无需绘制则不生成可视化瓦片。其次,当瓦片需要绘制时,可将图1中空间检索操作进行转换:获取以像素点为中心的一定像素宽度范围内的像素点集合,并判断集合中像素对应结点在索引中是否存在,在瓦片中像素点是均匀离散的,某一像素点周围的像素点易通过瓦片分辨率计算得到。

同时在判断瓦片/像素对应结点是否存在时,根据其 Geohash 编码从已构建的 TQ 树根结点开始解码判断,以两个字符为单位从头至尾开始解码:其中“00”代表向下定位至结点的左下子结点;“10”代表定位至结点的右下子结点;“01”代表定位至结点的左上子结点;“11”代表定位至结点的右上子结点。当解码向下定位时的子结点为空,说明瓦片/像素对应结点不存在;当解码完毕且瓦片/像素编码与结点编码一致,说明瓦片/像素对应结点存在。

像素值计算流程如图 7 所示。 P 为待计算像素点, R 为指定像素宽度范围, R_c 为像素分辨率,橙色网格表示像素点对应结点存在。当计算像素值时:首先判断 P 点对应结点是否存在,如存在则生成像素值;如不存在,逐一计算 P 点外围第一层橙色像素点,判断像素点与 P 点距离是否不超过 R ,同时判断该像素对应结点是否存在,如均满足条件,则生成像素值;如外围第一层像素点均不满足条件,再向外一层继续判断,直至外层像素点不在像素范围距离内。

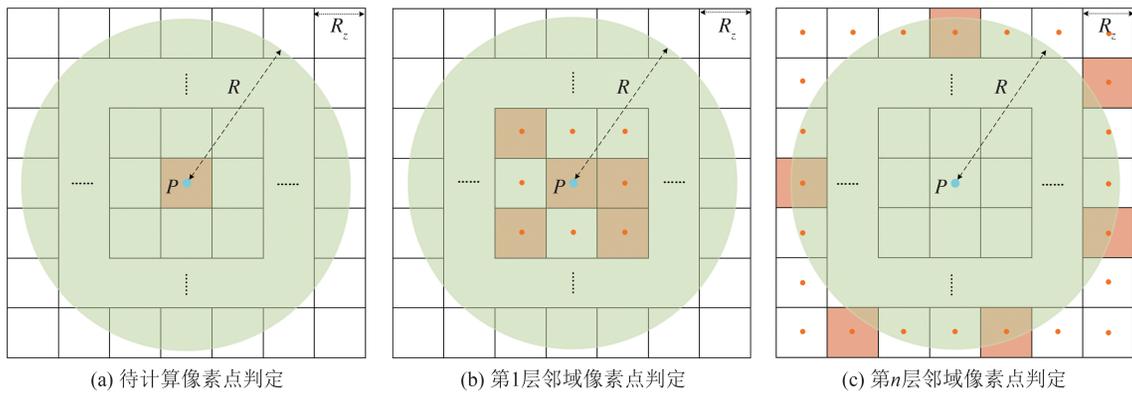


图 7 像素值计算流程

Fig. 7 Process for Calculating Pixel Value

基于 TQ 树索引的快速可视化方法具有两点优势:首先预先判断瓦片是否绘制,减少了空白瓦片的绘制;其次在像素计算时将复杂耗时的空间检索操作转换为索引结点存在判断操作,大幅

缩短了单张瓦片的绘制时间。

4.3 并行实时计算框架

本文采取图 8 所示并行可视化实时计算框架实现大规模矢量数据的实时可视化。

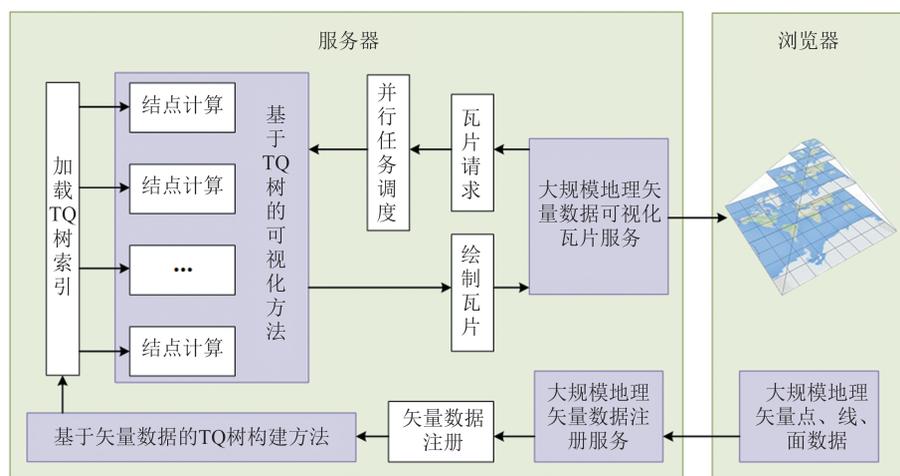


图 8 并行可视化实时计算框架

Fig. 8 Real-Time Computing Framework of Parallel Visualization

矢量数据可视化结果被封装成标准的网络地图瓦片服务,用户可通过浏览器对数据进行浏览。该框架包含大规模地理矢量数据注册服务和可视化瓦片服务:在矢量数据注册服务中,“矢

量要素注册”指采用 TQ 树构建方法构建索引,其是可视化瓦片服务的基础;在矢量数据可视化瓦片服务中,“瓦片请求”指发送屏幕范围内的瓦片,同时“加载 TQ 树索引”至结点中,“结点计算”

指采用基于TQ树的可视化方法生成像素值,最终“绘制瓦片”得到最终结果。在结点计算中基于MPI-OpenMP混合并行方法,将每个瓦片作为独立分析任务分配至一个MPI进程,同时每个MPI进程内启动多个OpenMP线程,用于提升单

张瓦片生成速度。

5 实验与结果分析

本文以不同规模的矢量数据为实验对象,实验数据如表1所示。

表1 实验数据集

Tab. 1 Experiment Datasets

数据集	类型/单位	要素数量	数据集规模
L_1 : OSM 全球邮编边界	线/条线段	210 054	65 334 342
L_2 : OSM 全球墓地边界	线/条线段	193 339	1 800 980
L_3 : OSM 全球运动区域边界	线/条线段	1 783 952	18 969 047
L_4 : OSM 全球水域边界	线/条线段	9 211 000	376 208 235
L_5 : OSM 全球绿地边界	线/条线段	11 634 551	396 527 008
L_6 : OSM 全球道路	线/条线段	72 336 396	717 048 198
L_7 : OSM 全球线要素	线/条线段	106 268 554	1 573 469 984
P_1 : OSM 全球点要素	点/个	2 682 401 763	2 682 401 763
A_1 : OSM 全球建筑物	面/条边	114 839 692	804 028 282
A_2 : OSM 全球面要素	面/条边	181 772 692	2 077 524 465

表1中,数据集包括全球尺度上的矢量点、线、面数据集。

所有数据集来自OSM(OpenStreetMap)^[27],其是一个包含矢量地理数据的开源志愿者网络地图服务。从 $L_1 \sim L_7$,数据规模逐渐增大, L_7 、 P_1 和 A_2 分别包含OSM全球所有点、线、面数据,其中点对象、线对象、面对象的规模均达到十亿级别。实验环境为:内存为512 GB,核数为32(每个核数中含8个线程)的2.6 GHz的Inter处理器,且处理系统为Ubuntu 18.04。

为验证基于TQ树的显示导向型可视化方法在支持快速数据预处理与实时可视化上的性能,与现有显示导向型可视化方法HiVision进行比较,HiVision在处理大规模矢量数据时的可视化性能大幅超过了HadoopViz^[13]、GeoSparkViz^[14]等流行的可视化平台^[4]。通过实验验证该方法在索引构建时间、输出索引尺寸上、可视化瓦片生成时间和瓦片绘制速率上的优势。实验一对比了本文方法与HiVision的索引构建时间和索引尺寸大小,验证了该方法在数据预处理上的优势;实验二对比了本文方法与HiVision的可视化瓦片生成总时间,验证了该方法在可视化效率上的优势。

5.1 大规模矢量数据索引快速构建实验

本文分别采用TQ树构建方法和HiVision对数据集构建索引,在HiVision中基于Quadratic算

法构建R树索引^[28],本文方法构建总层数为16的TQ树。两种方法生成索引均输出至外存。两者的索引构建时间和索引尺寸大小的对比,实验结果见表2,验证了基于矢量数据的TQ树构建方法的优势。

表2 索引构建时间与索引尺寸对比

Tab. 2 Comparison of Index Build Time and Index Size

数据集	HiVision		TQ树	
	t_1/s	s_1/GB	t_2/s	s_2/GB
L_1	247.57	6.42	24.77	0.39
L_2	2.45	0.16	0.41	0.03
L_3	118.58	1.76	6.09	0.07
L_4	1 487.52	36.07	118.52	1.54
L_5	1 097.64	37.53	130.46	1.23
L_6	2 407.65	69.06	297.22	2.47
L_7	2 468.21	150.61	492.01	4.09
P_1	6 916.73	244.28	685.92	4.33
A_1	1 118.39	60.69	148.09	0.22
A_2	3 378.77	197.98	437.76	3.06

表2中, t_1 和 s_1 分别表示HiVision的索引构建时间和索引尺寸大小, t_2 和 s_2 分别表示TQ树的索引构建时间和索引尺寸大小。首先对比索引构建时间:对于所有数据集,TQ树构建方法的索引构建时间远小于HiVision,尤其对于亿级规模数据集 $L_4 \sim L_7$ 、 P_1 、 $A_1 \sim A_2$,采用HiVision构建索引的时间非常漫长,原因在于R树构建是不停调整

的动态过程,当数据量激增时,需要对结点内的数据甚至整个树层次结构进行调整,使得索引构建非常耗时;对于 L_7 、 P_1 和 A_2 , TQ 树索引的构建时间分别仅为 HiVision 的 19.94%、9.92% 和 12.96%。其次对比索引输出至外存中的尺寸大小:同样对于所有数据集, TQ 树构建方法仅需存储结点信息,从而节省大量存储空间,其索引尺寸远远小于 HiVision。对于 L_7 、 P_1 和 A_2 , 索引大小分别仅为 HiVision 的 2.72%、1.77% 和 1.55%。

综上可知,基于 TQ 树的索引构建方法在数据预处理上具有更好的性能,可为后续数据可视化提供更快数据预处理支撑。

5.2 大规模矢量数据实时可视化实验

本文分别采用 HiVision 和基于 TQ 树的可视化方法对矢量数据集进行可视化,对于每个数据集,分别使用两种方法生成 0~8 层级的可视化瓦片数据。统计两者生成瓦片数据所需生成瓦片数、总耗时和在不同层级上的瓦片生成速率。在实验环境中均启用 32 个进程,每个进程中启用 8 个线程,用于生成可视化瓦片。

图 9 展示了生成 0~8 层级瓦片数据需生成的瓦片数,基于 TQ 树的可视化方法在绘制瓦片时通过预先判断减少了空白瓦片生成,其所需生成瓦片数远小于 HiVision,这有助于减少生成瓦片的总耗时。图 10 展示了生成 0~8 层级瓦片数据耗时对比。

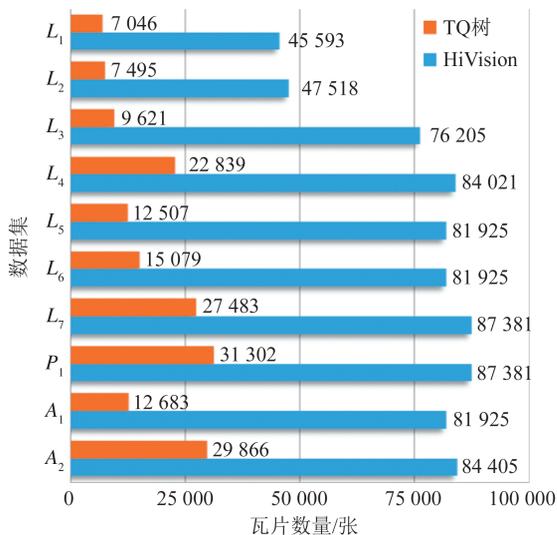


图 9 生成 0~8 层级瓦片数据所需生成瓦片数

Fig. 9 Number of Tiles Required to Generate Tile Data of Zoom Levels 0-8

从图 10 结果知,基于 TQ 树的可视化方法表现出的瓦片生成性能远好于 HiVision。对于

20 亿规模的数据集 P_1 仅耗时 36.27 s,对于 L_7 、 P_1 和 A_2 , 该方法的总耗时分别仅为 HiVision 的 8.22%、4.53% 和 3.16%,同时随着数据规模的增长,采用该方法瓦片耗时增长趋势远小于 HiVision,其对数据规模不敏感,表现出非常优异的瓦片生成性能,非常适用于大规模矢量数据的可视化。同时为验证支持矢量数据实时绘制的性能,对亿级规模数据集 L_4 ~ L_7 、 P_1 、 A_1 、 A_2 , 计算两种方法在各层级的瓦片生成速率,结果如图 11 所示。



图 10 生成 0~8 层级可视化瓦片数据耗时

Fig. 10 Total Time of Generating Tiles at Zoom Levels 0-8

对于所有数据集,基于 TQ 树的可视化方法相比 HiVision 表现出更高效的瓦片生成速率,在 P_1 的最慢速率也达到 438.05 张瓦片/s。在实际应用中,用户窗口范围内的瓦片数量通常不超过 50 张,所以用户在浏览过程中变换显示层级时,在 1 s 以内即可完成可视化结果的计算。同时当更改颜色参数时,进行重新计算的结果也均可在 1 s 内返回,所以该方法可以支持对所有数据同时进行实时的颜色渲染。由此可见,该方法在处理亿级规模矢量数据时具有优异的实时可视化性能。

综上可知,基于 TQ 树的可视化方法可以很好地支持大规模矢量数据可视化:相比传统方法生成本地瓦片非常耗时,更不具备实时可视化能力,该方法不仅可用于快速全部生成可视化瓦片存储在外存,而且具备优异的实时可视化能力,便于用户进行交互式浏览和分析。

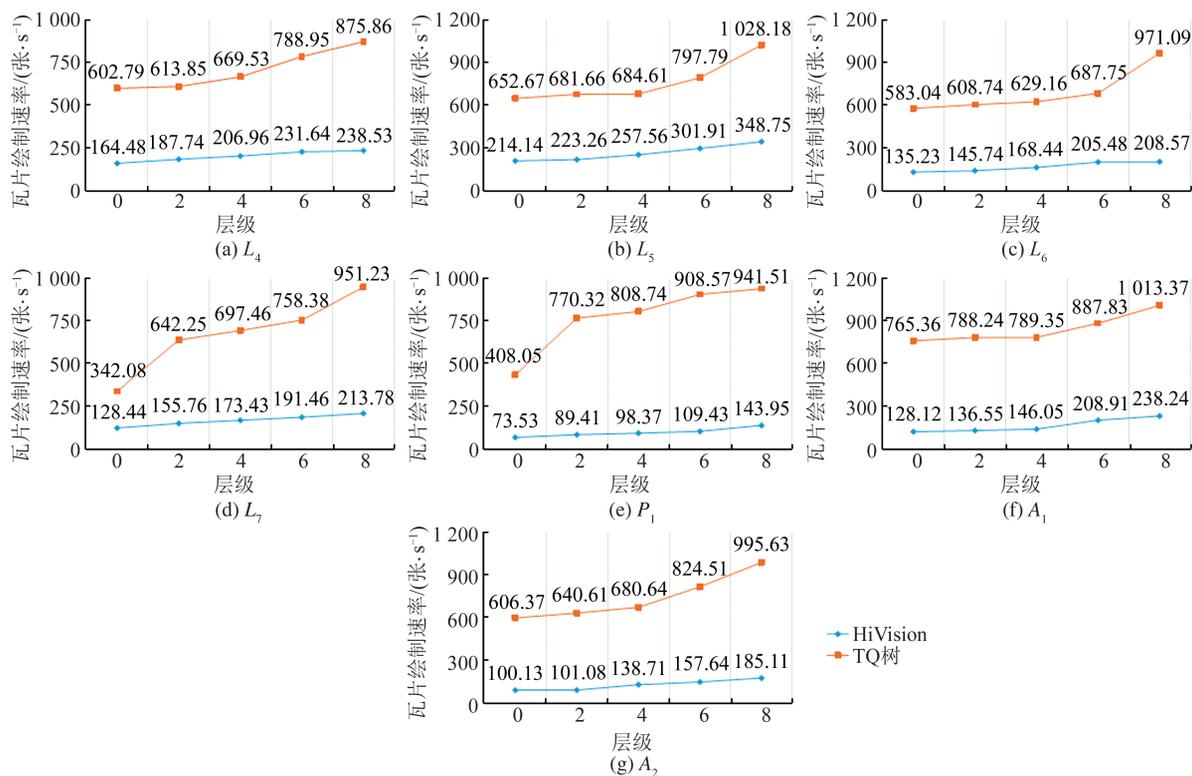


图 11 各层级瓦片生成速率

Fig. 11 Tile Generation Speed at Each Zoom Level

6 结 语

本文针对大规模地理矢量数据可视化这一当前GIS领域研究的热点问题,在显示导向计算模型基础上,设计了一种支持大规模矢量数据实时可视化的高效索引技术:在数据预处理时采用递归划分全球空间范围实现索引快速构建,索引结点与瓦片/像素的对齐;在可视化时,以像素点为计算单元,判断TQ树索引中结点是否存在,快速计算像素值生成显示效果。相比现有显示导向型可视化方法,该方法不仅进一步提升了显示导向型计算的速度,而且兼顾了数据快速预处理的效率,具备更优异的可视化性能和更强的实用性,非常适用于大规模矢量数据的实时可视化。

参 考 文 献

- [1] Yao Xiaochuang. Research on Key Technologies of Vector Big Data Management[D]. Beijing: China Agricultural University, 2017. (姚晓闯. 矢量大数据管理关键技术研究[D]. 北京: 中国农业大学, 2017.)
- [2] Tong X, Ben J, Liu Y, et al. Modeling and Expression of Vector Data in the Hexagonal Discrete Global Grid System[J]. *The International Archives of the Photogrammetry, Remote Sensing and Spa-*

tial Information Sciences, 2013, 4: 15-25.

- [3] MacEachren A M, Gahegan M, Pike W, et al. Geovisualization for Knowledge Construction and Decision Support[J]. *IEEE Computer Graphics and Applications*, 2004, 24(1): 13-17.
- [4] Ma M, Wu Y, Ouyang X, et al. HiVision: Rapid Visualization of Large-Scale Spatial Vector Data[J]. *Computers & Geosciences*, 2021, 147: 104665.
- [5] Gou Limei, Zhu Meizheng, Li Yanming. Study on Web Map Tile Service Based on RESTful[J]. *Computer Engineering and Design*, 2012, 33(9): 3609-3616. (苟丽美, 朱美正, 李艳明. RESTful风格地图瓦片服务的研究[J]. 计算机工程与设计, 2012, 33(9): 3609-3616.)
- [6] Yin Fuzhong, Sun Limin. The Research of Map Publishing Platform Development Based on the Tile Pyramid Technology[J]. *Geomatics & Spatial Information Technology*, 2010, 33(5): 16-17. (殷福忠, 孙立民. 基于瓦片金字塔技术的地图发布平台开发研究[J]. 测绘与空间地理信息, 2010, 33(5): 16-17.)
- [7] Guo M, Guan Q, Xie Z, et al. A Spatially Adaptive Decomposition Approach for Parallel Vector Data Visualization of Polylines and Polygons[J]. *International Journal of Geographical Information Science*, 2015, 29(8): 1419-1440.

- [8] Gao J, Wang C, Li L, et al. A Parallel Multiresolution Volume Rendering Algorithm for Large Data Visualization[J]. *Parallel Computing*, 2005, 31(2): 185-204.
- [9] Hughes J N, Annex A, Eichelberger C N, et al. GeoMesa: A Distributed Architecture for Spatio-temporal Fusion[C]// Geospatial Informatics, Fusion, and Motion Video Analytics V, Baltimore, Maryland, USA, 2015.
- [10] Eldawy A, Mokbel M F, Alharthi S, et al. SHAHED: A MapReduce-Based System for Querying and Visualizing Spatio-Temporal Satellite Data[C]//The 31st International Conference on Data Engineering, Seoul, Korea, 2015.
- [11] Guo M, Huang Y, Guan Q, et al. An Efficient Data Organization and Scheduling Strategy for Accelerating Large Vector Data Rendering [J]. *Transactions in GIS*, 2017, 21(6): 1217-1236.
- [12] Pahins C A L, Stephens S A, Scheidegger C, et al. Hashedcubes: Simple, Low Memory, Real-Time Visual Exploration of Big Data[J]. *IEEE Transactions on Visualization and Computer Graphics*, 2017, 23(1): 671-680.
- [13] Eldawy A, Mokbel M F, Jonathan C. HadoopViz: A MapReduce Framework for Extensible Visualization of Big Spatial Data[C]//The 32nd International Conference on Data Engineering, Helsinki, Finland, 2016.
- [14] Yu J, Zhang Z, Sarwat M. GeoSparkViz: A Scalable Geospatial Data Visualization Framework in the Apache Spark Ecosystem[C]// The 30th International Conference on Scientific and Statistical Database Management, Bozen-Bolzano, Italy, 2018.
- [15] Ma M Y, Yang A R, Wu Y, et al. DiSA: A Display-Driven Spatial Analysis Framework for Large-Scale Vector Data[C]// The 28th International Conference on Advances in Geographic Information Systems, Seattle, WA, USA, 2020.
- [16] Ma Mengyu, Wu Ye, Chen Luo, et al. Display-Oriented Data Visualization Technique for Large-Scale Geographic Vector Data [J]. *Computer Science*, 2020, 47(9): 117-122. (马梦宇, 吴烨, 陈萃, 等. 显示导向型的大规模地理矢量实时可视化技术[J]. *计算机科学*, 2020, 47(9): 117-122.)
- [17] Zhou Jingwei. Research on High Performance Computing Model and Key Technologies of Vector Big Data [D]. Hangzhou: Zhejiang University, 2016. (周经纬. 矢量大数据高性能计算模型及关键技术研究[D]. 杭州: 浙江大学, 2016.)
- [18] Nievergelt J, Hinterberger H, Sevcik K C. The Grid File: An Adaptable, Symmetric Multikey File Structure[J]. *ACM Transactions on Database Systems*, 1984, 9(1): 38-71.
- [19] Finkel R A, Bentley J L. Quad Trees a Data Structure for Retrieval on Composite Keys[J]. *Acta Informatica*, 1974, 4(1), DOI:10.1021/ja01172a501.
- [20] Guttman A. R-Trees: A Dynamic Index Structure for Spatial Searching[J]. *ACM SIGMOD Record*, 1984, 14(2):47-57
- [21] Feng J, Tang Z X, Wei M, et al. HQ-Tree: A Distributed Spatial Index Based on Hadoop[J]. *China Communications*, 2014, 11(7): 128-141.
- [22] Li Xun. Research on Parallel Vector Data Indexing Algorithm Based on Hilbert Partition [D]. Chengdu: University of Electronic Science and Technology of China, 2013. (李勋. 基于 Hilbert 划分的并行矢量数据索引算法研究[D]. 成都: 电子科技大学, 2013.)
- [23] Challa J S, Goyal P, Nikhil S, et al. DD-Rtree: A Dynamic Distributed Data Structure for Efficient Data Distribution Among Cluster Nodes for Spatial Data Mining Algorithms[C]// IEEE International Conference on Big Data (Big Data), Washington, USA, 2017.
- [24] Ma M Y, Wu Y, Luo W Z, et al. HiBuffer: Buffer Analysis of 10-Million-Scale Spatial Data in Real Time[J]. *ISPRS International Journal of Geo-Information*, 2018, 7(12): 467.
- [25] Ma M Y, Wu Y, Guo N, et al. A Parallel Processing Model for Accelerating High-Resolution Geo-Spatial Accessibility Analysis [J]. *IEEE Access*, 2018, 6: 52936-52952.
- [26] Moussalli R, Srivatsa M, Asaad S. Fast and Flexible Conversion of Geohash Codes to and from Latitude/Longitude Coordinates[C]// The 23rd Annual International Symposium on Field-Programmable Custom Computing Machines, Vancouver, Canada, 2015.
- [27] OpenStreetMap [EB/OL]. [2021-08-23]. <https://openstreetmap.org>.
- [28] Fernández, F. Boost [EB/OL]. [2021-08-23]. https://www.boost.org/doc/libs/1_76_0/libs/geometry/doc/html.