

显示导向型的大规模地理矢量实时可视化技术



马梦宇 吴 焯 陈 萃 伍江江 李 军 景 宁

国防科技大学电子科学学院 长沙 410073

(mamengyu10@nudt.edu.cn)

摘 要 对大规模地理矢量要素进行实时可视化是当今地理信息科学领域面临的一个严峻挑战。在现有地理矢量要素可视化方法中,随着数据规模的增长,计算规模也急剧扩大,这导致尽管使用了高性能计算技术,仍很难应对大规模地理矢量要素的实时可视化。基于此,文中提出了一种基于显示导向型计算的地理矢量要素可视化技术。该技术从显示角度出发,将每个用于屏幕显示的像素点作为独立的计算单元,根据用户浏览地理矢量要素时屏幕显示的区域及分辨率确定待计算的像素点范围,通过直接计算每个像素点的值来生成最终的显示结果。该技术使得可视化的计算规模仅依赖于屏幕显示的像素数量,具有对数据规模不敏感的优点,可用于支持大规模地理矢量要素的实时可视化。实验结果表明,显示导向型地理矢量可视化技术可用于支持亿级矢量数据的实时可视化绘制。

关键词: 数据可视化;空间大数据;显示导向型计算;并行计算;实时计算

中图法分类号 TP391

Display-oriented Data Visualization Technique for Large-scale Geographic Vector Data

MA Meng-yu, WU Ye, CHEN Luo, WU Jiang-jiang, LI Jun and JING Ning

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

Abstract Rapid visualization of large-scale geographic vector data remains a challenging problem in geographic information science. In existing visualization methods, the computational scales expand rapidly with data volumes, leading to the result that it is difficult to provide real-time visualization for large-scale geographic vector data, though parallel acceleration technologies are adopted. This paper presents a display-oriented data visualization method for large-scale geographic vector data. Different from traditional methods, the core task of the display-oriented method is to determine the pixel range according to the screen display and then calculate the value of each pixel in the range. As the number of pixels for display is stable, the display-oriented data visualization method is less sensitive to data volumes and can be used to provide real-time data visualization for large-scale geographic vector data. Experiments show that our approach is capable of handling 100-million-scale geographic vector data.

Keywords Data visualization, Spatial big data, Display-oriented computing, Parallel computing, Real-time

1 引言

近年来,人、机、物三元世界的高度融合引发了数据规模的爆炸式增长和数据模式的高度复杂化,世界已进入网络化的大数据时代。由于空间数据采集技术的飞速发展,以交通轨迹数据为代表的基于位置的数据,以微信、微博、Twitter等签到数据为代表的社交媒体数据,以及以环境监测、气象监控、智能家居、可穿戴设备等传感器数据为代表的物联网数据正以前所未有的速度增长,急剧膨胀的地理空间数据已成为大数据流的重要组成部分。与此相对应,在多样化需求的牵引下,大数据分析应用需应对空前的空间数据规模。例如,Blue Brain计划^[1]通过将网络神经元建模为空间数据来分析大脑的结构和功能;气象学家基于空间分析进行天气预测^[2];

新闻播报员利用带地理标签的Twitter数据进行时间预测和分析^[3]。为便于计算机进行处理,通常将空间中的地理实体抽象为点、线、面等地理矢量要素,这些分析应用需要解决的首要问题是如何对大规模地理矢量进行快速可视化。在大数据时代,对大规模地理矢量进行实时可视化是当今地理信息科学领域面临的一个严峻挑战。

随着计算机技术的发展,以分布式、并行化为代表的高性能计算技术正逐渐融入地理信息领域,如何利用具有高性能计算能力的新型硬件体系结构带来的计算性能提升,解决现有数据密集、计算密集的大规模地理矢量分析问题成为地理信息科学领域一个研究热点。目前,已有很多学者针对大规模矢量可视化问题提出了一系列并行处理方法,与原有基于单一进程处理的串行方法相比,性能取得了较为明显的提升。

收稿日期:2019-08-23 返修日期:2019-10-10 本文已加入开放科学计划(OSID),请扫描上方二维码获取补充信息。

基金项目:国家自然科学基金(41471321,41871284)

This work was supported by the National Natural Science Foundation of China (41471321,41871284).

通信作者:景宁(ningjing@nudt.edu.cn)

此外,随着 MPI (Message Passing Interface), OpenMP, Hadoop, Storm, Spark 等多种并行计算模型的不断涌现,国内外还出现了一系列性能较好的大规模地理矢量可视化框架,如 HadoopViz^[4], GeoSparkViz^[5]。高性能计算技术的应用为面向大规模矢量的快速可视化提供了很好的解决途径。

然而,尽管采用了多样化的并行加速技术,并且设计了多种具有良好负载均衡特性的算法,使得矢量可视化性能取得明显提升,但在地理矢量要素规模较大的情况下,现有方法仍很难支持地理矢量的实时可视化,其主要原因是现有地理矢量可视化的相关研究均采用基于数据导向型计算的算法。如图 1 所示,基于数据导向型的算法首先分别对数据集中的每个元素进行绘制,然后将绘制结果进行合并,输出最终的可视化结果。在这种方式中,并行策略主要表现为地理矢量在多进程中的不同任务划分及绘制结果的合并策略。基于数据导向型计算的地理矢量可视化方式是比较直观的计算方式,随着矢量规模的增大,对矢量数据进行可视化的计算规模也急剧扩大,这导致当数据规模较大时,很难达到实时可视化的效果。

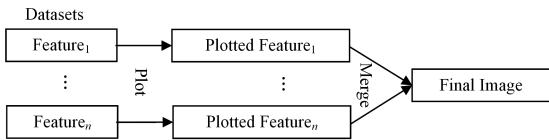


图 1 基于数据导向型计算的地理矢量可视化

Fig. 1 Data-oriented geographic vector data visualization

基于此,本文设计了一种基于显示导向型计算的地理矢量可视化技术,其具体处理流程如图 2 所示。不同于基于数据导向型的计算方式,该方式从显示角度出发,将每个用于屏幕显示的像素点作为独立的计算单元,根据用户浏览地理矢量时屏幕显示的区域及分辨率确定待计算的像素点范围,通过直接计算每个像素点的值来生成最终的显示结果。这种方式的计算规模仅依赖屏幕显示的像素数量,具有对数据规模不敏感的优点,尤其适用于解决大规模地理空间矢量的可视化问题。

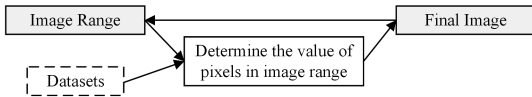


图 2 基于显示导向型计算的地理矢量可视化

Fig. 2 Display-oriented geographic vector data visualization

在我们的前期工作中,已成功将基于显示导向型计算的思想运用于解决缓冲区分析、叠加分析等空间分析问题。本文将基于显示导向型计算的思想应用于大规模地理要素可视化领域,同样取得了较好的效果。

2 相关工作

2.1 空间大数据

随着空间数据规模的不断增长,国内外涌现出了许多针对空间大数据的相关研究。这些研究大多集中于解决空间分析中某些特定的科学问题,如空间范围查询、空间连接运算^[6]、kNN 查询^[7]等。此外,还有一些学者致力于设计专门

针对空间大数据的分析处理系统,如 Hadoop-GIS^[8] 和 MD-HBase^[9],但这些系统均没有提供较为高效的空数据可视化方法。

在我们的前期研究中,已成功将基于显示导向型计算的思想用于解决缓冲区分析、叠加分析等基础空间分析问题,设计了 HiBuffer、HiBO 等系统用于支持大规模空间数据的实时缓冲区和叠加分析。与传统基于数据导向型计算的方法不同,HiBuffer 和 HiBO 具有对数据规模不敏感的特性。表 1 列出了 HiBuffer 与传统方法的性能对比,通过对比可知,传统方法中需花费数小时才可完成的分析任务,在采用基于显示导向型计算的方式后均能够实现实时响应的效果;同时,对于传统方法无法完成的千万规模的数据缓冲区分析,HiBuffer 仍可达到实时的效果。

表 1 HiBuffer 与传统方法的性能对比

Table 1 Performance of HiBuffer and other methods

Algorithm	40927	208067	597829	21898508
	linestrings	linestrings	linestrings	linestrings
Parallel Method 1 ^[10]	9.0 s	38.8 s	332.3 s	Failed
Parallel Method 2 ^[11]	15.4 s	128.2 s	936.9 s	Failed
Parallel Method 3 ^[12]	12.3 s	75.5 s	661.9 s	Failed
Parallel Method 4 ^[13]	17.2 s	220.8 s	2 813.4 s	Failed
PostGIS	34.9 s	295.8 s	2 380.2 s	Failed
QGIS	129 s	2 788 s	Failed	Failed
ArcGIS	139 s	2 365 s	Failed	Failed
HiBuffer	<1 s	<1 s	<1 s	<1 s

在大数据时代,空间大数据的分析应用是研究热点,作为空间大数据分析的基础支持,研究空间大数据的可视化具有重要的意义。

2.2 大数据可视化

近年来出现了很多针对非空间大数据进行可视化的系统^[14-18],这些系统通常通过采样或聚集的方式减小数据规模,然后将处理后的数据以图表的方式绘制在一台单机上。Ermac^[18] 提出将数据可视化算法嵌入数据库中,其可以在数据库生成查询计划的过程中针对数据的采样和聚集方法进行专门的优化。同样地,M4 根据可视化结果图片大小重写了 SQL (Structured Query Language) 查询,通过在数据库中对数据进行聚集操作减小了用于可视化的数据规模。Bin-summarise-smooth^[17] 通过一系列操作 (Binning, Summarizing, Smoothing) 来减小数据规模,从而实现在单机上对大规模数据进行可视化的效果。

在大数据分析领域,数据的可视化分析具有重要的研究价值。虽然大规模地理矢量属于空间大数据范畴,其可视化的相关研究成果对非空间数据可视化也有一定的借鉴意义。

2.3 空间大数据可视化

空间大数据的可视化是大数据领域中的一个重要的分析方法。为了对大规模空间数据进行可视化显示,开放地理空间信息联盟设计了一个标准网络地图服务规范^[19]。通过将网络地图服务组织为瓦片金字塔模型(见图 3),可以实现在浏览器端浏览大规模的空间数据^[20]。

在空间数据可视化和地图制图领域,有很多对空间数据进行可视化分析的研究,同时也涌现出了许多性能较好的空

间数据可视化工具,如 Mapnik^[21],GeoServer^[22],MapServer^[23]等。但现有针对大规模地理矢量进行可视化的解决方法均基于数据导向的计算方式,效果不是很理想^[24-25]。一方面,当要素规模很大时,数据分割切片过程可能耗时很长甚至无法完成;另一方面,将切割好的要素绘制成可显示的图片耗时也较长。为了让用户能在客户端对大规模地理矢量进行快速浏览,往往需要消耗较长时间来预先生成矢量的瓦片金字塔,并且采用这种方式一旦改变矢量数据的显示样式,又需要重新生成瓦片金字塔。为了提升生成瓦片的速度,研究者将并行计算技术引入到地理矢量显示领域^[26-28]。目前已有一些大规模地理矢量可视化框架,Eldawy等^[4]基于MapReduce设计了一个面向大规模矢量数据的分布式可视化系统HadoopViz。Jia等^[5]基于分布式共享内存计算架构Spark,提出了一种面向矢量大数据的显示系统GeoSparkViz,该系统通过采用基于地图瓦片的数据划分策略取得了较好的负载均衡效果。Ghosh等^[29]为了提升生成金字塔瓦片的速度、减小金字塔瓦片集的大小,提出了一种矢-栅结合的瓦片组织方式,对于一些矢量要素较少的区域,由于其绘制速度较快,不需要预先生成栅格瓦片,而采用占空间较小的矢量瓦片进行存储。

在现有空间数据可视化方法中,对地理矢量进行可视化的方式均为基于数据导向型计算的方式。这种方式下,随着矢量规模的增长,计算规模也急剧扩大,导致当数据规模较大时,即使采用并行加速技术,也很难达到实时可视化的效果。

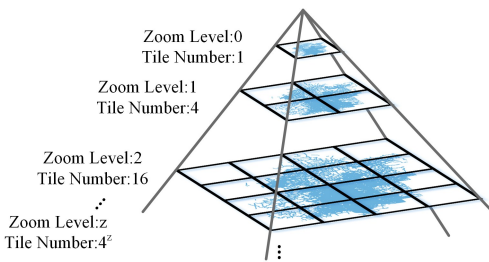


图3 地图瓦片金字塔模型
Fig. 3 Tile-pyramid structure

3 显示导向型可视化方法

面向显示导向型计算的地理矢量要素组织方法是可视化技术的基础。显示导向型计算的地理矢量可视化的核心是根据矢量直接判断屏幕像素的显示值,为此矢量组织方法除了要考虑空间要素属性信息的管理外,还需要构建索引用于支持像素点与矢量元素空间关系的快速判断。R树作为一种空间数据存储的树状数据结构,被广泛应用于空间数据管理,众多学者对R树进行了充分的改进,同时基于R树的空间查询,包括矩形框查询和最近邻查询等也被充分研究。基于此,本文采用R树索引对地理矢量进行组织管理。

本文对地理矢量采用图4所示的方式进行组织,分别为点、线、面等不同要素构建不同的空间索引,用于支持多样化的可视化查询。对于点要素,直接将每个点作为R树中的一个节点;对于线要素,由于每个线要素通常由多个线段组成,本文将每个线段作为R树中的一个节点;对于面要素,为实

现快速判断像素点是否在面要素内部的功能,本文对面要素构建两级R树索引,第一级索引将面要素的边界线段作为R树节点,并用 $IsLevel$ 标识线段是否平行于 x 轴,第二级索引存储每个面要素的最小包围外包框(Minimum Bounding Rectangle, MBR)。由于用于管理的地理要素规模可能较为庞大,相应地,构建的空间索引占用的存储空间也会较大,为避免查询时加载索引占用过高的系统内存,本文将规模较大的地理矢量集的空间索引存储为内存映射文件,并通过内存映射的方式对空间索引进行访问。

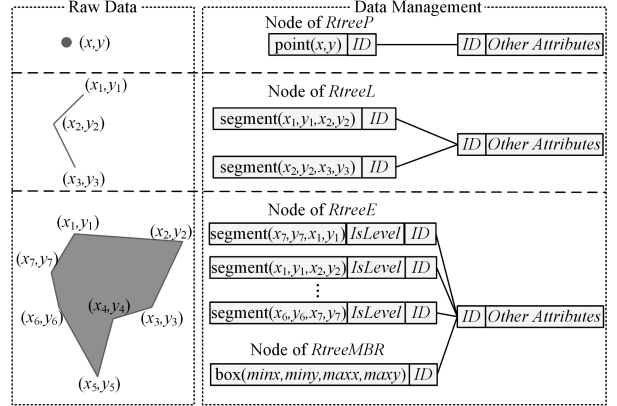


图4 显示导向型地理矢量可视化数据组织模型
Fig. 4 Data organization of display-oriented visualization

接下来,基于地理矢量的组织模型研究其可视化算法。对地理矢量进行可视化,首先需要确定像素点与地理要素的空间拓扑关系,进而根据样式对矢量进行可视化绘制。下面给出了地理矢量可视化流程中的两种典型空间拓扑关系判别问题及其相应的处理算法。

问题1 判断像素点是否在点要素、线要素、面要素边界线的一定像素宽度范围内。为了实现更好的显示效果,对点要素、线要素、面要素边界线进行可视化,须将其可视化带有一定像素宽度的点或线,即在可视化过程中须判断像素点是否在点要素、线要素、面要素边界线一定半径范围内,而非仅仅判断像素点是否与点要素、线要素、面要素的边界线相交。矢量要素边界绘制算法如算法1所示。

算法1 矢量要素边界绘制算法

Input: Pixel P, zoom level Z, radius N (pixels), Rtree (RtreeP, RtreeL or RtreeE)

Output: True or False (whether P is within R from points, linestrings, or polygon boundaries)

1. $R \leftarrow N * RESOLUTION(Z)$
2. $r \leftarrow R * \sqrt{2}/2$
3. $InnerBox \leftarrow box(P, x-r, P, y-r, P, x+r, P, y+r)$
4. $Tmp \leftarrow satisfy Rtree.intersect(InnerBox).limit(1)$
5. if Tmp is not null then return True
6. else
7. $OuterBox \leftarrow box(P, x-R, P, y-R, P, x+R, P, y+R)$
8. $Tmp \leftarrow satisfy Rtree.intersect(OuterBox).nearest(P)$
9. if Tmp is not null & distance(Tmp, P) $\leq R$ then return True
10. return False

算法 1 判断给定像素点 P 是否在点要素、线要素、面要素边界线一定像素宽度范围内,可转换为判断 P 到最近的空间要素的距离是否小于 R ,其中 R 为根据显示分辨率 Z 和像素宽度 N 确定的空间绘制半径。对于此问题,一个较直观的解决办法是以 P 为中心,以 R 为半径画一个圆 C ,然后以 C 为输入,对 R 树索引执行空间交查询,如果有要素相交则认为 P 在要素一定像素宽度范围内。然而,由于 R 树索引的结构特征,基于 R 树的空间交查询更适用于采用矩形框作为输入的查询,而非其他形状如圆形做为查询框的查询,此外基于 R 树的最近邻查询的复杂度远高于矩形框查询,因此本文引入了 $InnerBox$ 和 $OuterBox$ 用于优化查询性能。 $InnerBox$ 用于像素点附近空间要素比较密集的场景,而 $OuterBox$ 用于像素点附近空间要素比较稀疏的场景。像素 P 在绘制宽度 R 下的 $InnerBox$ 和 $OuterBox$ 如图 5 所示。

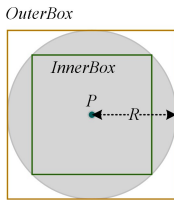


图 5 像素 P 在绘制宽度 R 下的 $InnerBox$ 和 $OuterBox$

Fig. 5 $InnerBox$ and $OuterBox$ of pixel P with radius R

问题 2 判断像素点是否在面要素内部。对于面要素,除了对其边界进行绘制外,还需要考虑其填充问题,因此需要设计算法用于快速判断像素点是否在面要素内部。

对于问题 2,本文基于面要素的两级空间索引采用算法 2 进行判断。该算法采用射线算法的思想,具体为:给定一个像素点和一个多边形,连接像素点和多边形边界外一点画一条水平于 x 轴的线段($QuerySegment$),然后利用多边形的边界索引判断其与多边形的交点个数,如果交点个数为奇数,则像素点在多边形内部,反之,则在多边形外部。此外,为了避免 $QuerySegment$ 过长导致与较多属于其他多边形的边相交进而影响查询性能,本文算法提出了两点优化用于减少 $QuerySegment$ 的长度:1)算法 2 第 2 行将 x 轴方向跨度较小的多边形放在队列前面用于优先判断;2)算法 2 第 6-9 行将像素点向距离较近的 MBR 边做垂线,从而形成 $QuerySegment$ 。

算法 2 面要素内部填充算法

Input: Pixel P , $RtreeE$ and $RtreeMBR$

Output: True or False (whether P is in polygons)

1. $TmpMBR \leftarrow$ satisfy $RtreeMBR.intersect(P)$
2. $sort(TmpMBR) \leftarrow$ Smaller x span polygon ranks ahead.
3. for $v \in TmpMBR$ do
4. $EdgeCount \leftarrow 0$
5. $vMinx \leftarrow v.Box.minx, vMaxx \leftarrow v.Box.maxx$
6. if $P.x - vMinx < vMaxx - P.x$ then
7. $QuerySegment \leftarrow segment(vMinx, P.y, P.x, P.y)$
8. else
9. $QuerySegment \leftarrow segment(P.x, P.y, vMaxx, P.y)$
10. $TmpS \leftarrow$ satisfy $RtreeE.intersect(QuerySegment)$
11. for $s \in TmpS$ do

12. if $(s.ID = v.ID) \&\&. (not s.IsLevel)$ then
13. $EdgeCount++$
14. if $EdgeCount$ is odd then return True
15. return False

随后,根据像素点与地理矢量的空间拓扑关系对地理矢量进行可视化绘制,具体地,可直接根据样式文件中描述的线宽、色彩模式等样式对像素点进行绘制。

4 并行处理框架

本文采用图 6 所示的并行计算框架来支持大规模地理矢量的实时可视化。可视化结果将采用瓦片金字塔的方式进行组织管理,并将结果发布为标准的网络地图瓦片服务(Web Map Tile Service, WMTS),用户可通过浏览器对地理矢量进行快速浏览。该计算框架只生成屏幕视野范围内的瓦片,用户对数据进行缩放显示,具有更高分辨率的瓦片将被生成用于可视化显示,可实现矢量的无失真缩放。该计算框架主要包括多线程可视化服务(Multi-Thread Visualization Service MTVS)、基于内存的消息传递框架(In-Memory Messaging Framework IMMF)和混合并行可视化引擎(Hybrid-Parallel Visualization Engine HPVE)。

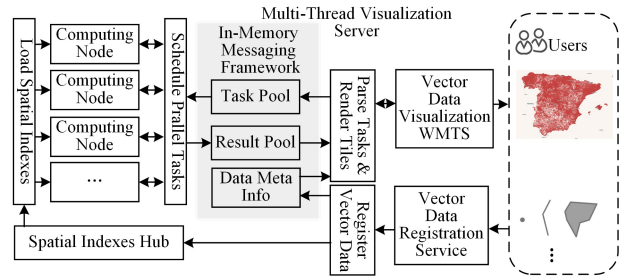


图 6 显示导向型地理矢量并行可视化的计算框架

Fig. 6 Architecture of display-oriented visualization

MTVS 实现可视化分析任务的解析和地理矢量注册服务,其将可视化分析结果封装为 WMTS。该计算框架将生成的每个可视化瓦片视作一个独立任务,对于过去任务中曾生成过且未被清除的瓦片,MTVS 将不会产生新的可视化分析任务请求。

IMMF 是一个基于 Redis 内存数据库实现的消息服务框架。在此框架中,分析结果、分析任务以及地理矢量数据集的源信息将被保存在内存中用于实现 HPVE 和 MTVS 间快速的消息传递。待处理的计算任务将被存储为一个先进先出的消息队列。系统长时间运行将积累大量结果数据,为保证系统长期稳定地运行,同时避免多次相同可视化请求重复消耗计算资源,IMMF 将会在内存占用达到上限时清理过期的分析结果。

HPAE 是并行可视化的核心,本文采用基于 MPI-OpenMP 混合并行的数据可视化分析模型。该模型中,每个瓦片作为一个独立分析任务被分配给一个 MPI 进程,同时,在每个 MPI 进程内,将启动多个 OpenMP 线程,用于提升单个瓦片的生成速度。可视化瓦片请求是以一种流的方式产生的,因此 HPAE 采用动态负载均衡策略。

5 实验与结果

本节通过实验验证了显示导向型可视化技术可用于支持实时可视化的性能。实验硬件环境如表 2 所列。实验采用的数据集如表 3 所列,P1,L1,A1 分别为中国大陆地区的兴趣点、道路和农田数据,其中道路和农田数据绘制的边数均达到亿级,这部分数据由生产商提供。实验软件环境为 MPICH 3.4,Boost C++ 1.64,GDAL 2.1.2,Redis 3.2.13。由于系统硬件环境为 32 核心,且每个核心有两个线程,因此显示导向型矢量可视化方法采用 32 个 MPI 进程且每个 MPI 进程中启动 2 个 OpenMP 线程。

表 2 实验环境

Table 2 Experimental environments

Item	Description
CPU	32cores * 2, Intel(R)Xeon(R)E5-4620@2.60 GHz
Memory	256 GB
Operating System	Centos 7

表 3 实验数据集 1

Table 3 Experimental datasets 1

Dataset	Abbr	Type	Records	Size
China points	P1	Point	20258450	20258450 points
China roads	L1	Linestring	21898508	163171928 segments
China farmland	A1	Polygon	10520644	133830561 edges

为了模拟实际使用过程,我们编写测试程序,分别对每个数据集随机产生 5000 个不同层级的可视化瓦片请求,然后对可视化瓦片的计算时间进行统计,结果如图 7 和图 8 所示。为了准确地测试矢量要素实时绘制的性能,实验不对中间结果进行缓存,即实验中每次请求的瓦片均为重新绘制的。由图 7 可知,对于所有数据集,显示导向型可视化技术均表现出良好的性能(每秒生成的瓦片数量均超过 800),考虑到对于主流 2000 及以下分辨率的显示屏浏览数据时屏幕内的瓦片数量通常不超 50 个,因此显示导向型可视化技术可用于支持大规模矢量要素可视化的多用户并发实时浏览。图 8 展示了每个瓦片生成时间的统计结果,由图 8 可以看出,对于所有实验数据集,大部分可视化瓦片均在 0.1 s 内生成可视化结果,这说明用户浏览可视化结果时具有较高的响应速度。综上,显示导向型地理矢量可视化技术可用于支持亿级矢量数据的实时可视化绘制。

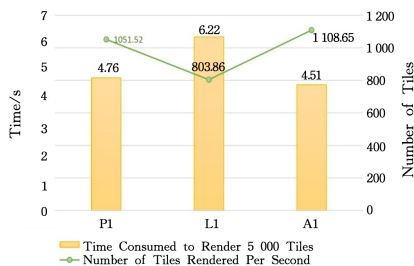


图 7 生成 5000 个瓦片的时间

Fig.7 Rendering time of 5000 tiles

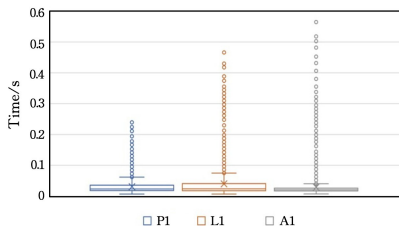


图 8 生成每个瓦片的耗时

Fig.8 Rendering time of each tile

下面通过实验将本文方法(用 HiVision 表示)与 HadoopViz 和 GeoSparkViz 的性能进行了对比。实验采用 4 台服务器(见表 2)构成的集群。实验数据如表 4 所列,均为来自公开地图 OpenStreetMap 的数据,且 D1 至 D7 数据量依次增加。实验中针对每个数据集,分别统计不同方法生成 1,3,5,7,9 级所有瓦片的总时间,实验结果如图 9 所示。通过观察可知,当数据规模较大时,本文方法 Hivision 生成全部瓦片的耗时比 HadoopViz 和 GeoSparkViz 要少,性能较好;此外,随着数据规模的增大,HadoopViz 和 GeoSparkViz 生成全部瓦片的时间都呈急剧上升的趋势,而本文方法则没有较为明显的上升趋势,这说明显示导向型可视化技术具有对数据集不敏感的特性。

表 4 实验数据集 2

Table 4 Experimental datasets 2

Dataset	Abbr	Type	Records	Size
OSM postal code areas boundaries	D1	Linestring	171226	65334342 segments
OSM boundaries of cemetery areas	D2	Linestring	193076	1800980 segments
OSM sporting areas boundaries	D3	Linestring	1767137	18969047 segments
OSM water areas boundaries	D4	Linestring	8419324	376208235 segments
OSM parks or green areas boundaries	D5	Linestring	9961896	454636308 segments
OSM roads and streets	D6	Linestring	72339945	717048198 segments
OSM all lines on the planet	D7	Linestring	106269321	1578947752 segments

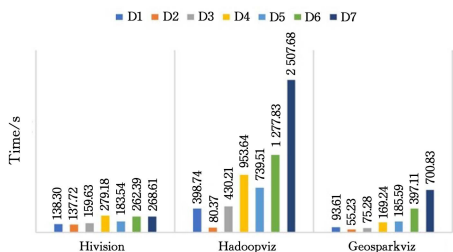


图 9 不同方法生成瓦片的耗时

Fig.9 Rendering time of tiles with different methods

结束语 本文针对地理矢量可视化这一计算机领域的热点和难点问题,设计了一种基于显示导向型计算的地理矢量可视化技术,该技术从屏幕显示像素出发,直接计算用于显示的像素值来生成最终结果。这种方式具有对数据规模不敏感的优点,尤其适用于解决大规模地理空间矢量的可视化问题。该研究在数据可视化领域和地图制图领域都有很大的借鉴意义。

参 考 文 献

- [1] HILL S, MARKRAM H. The Blue Brain Project [C]//International Conference of the IEEE Engineering in Medicine & Biology Society. IEEE, 2006.
- [2] FAGHMOUS J H, KUMAR V. Spatio-temporal Data Mining for Climate Data: Advances, Challenges, and Opportunities [M]//Data Mining and Knowledge Discovery for Big Data. Springer Berlin Heidelberg, 2014.
- [3] ATEFEH F, KHREICH W. A Survey of Techniques for Event Detection in Twitter [J]. Computational Intelligence, 2013, 31(1): 132-164.
- [4] ELDAWY A, MOKBEL M F, JONATHAN C. HadoopViz: A MapReduce framework for extensible visualization of big spatial data [C]//IEEE International Conference on Data Engineering. IEEE, 2016: 601-612.
- [5] JIA Y, ZONGSI Z, MOHAMED S. GeoSparkViz: a scalable geo-spatial data visualization framework in the apache spark ecosystem [C]//ACM International Conference on Scientific and Statistical Database Management. ACM, 2018: 15.
- [6] ZHANG S, HAN J, LIU Z, et al. SJMR: Parallelizing spatial join with MapReduce on clusters [C]//2009 IEEE International Conference on Cluster Computing and Workshops. IEEE, 2009.
- [7] LU W, SHEN Y, CHEN S, et al. Efficient processing of k nearest neighbor joins using MapReduce [J]. Proceedings of the VLDB Endowment, 2012, 5(10): 1016-1027.
- [8] AJI A, WANG F, VO H, et al. Hadoop-GIS: A High-Performance Spatial Data Warehousing System over MapReduce [J]. Proceedings Vldb Endowment, 2013, 6(11): 1009-1020.
- [9] NISHIMURA S, DAS S, AGRAWAL D, et al. MD-HBase: design and implementation of an elastic data infrastructure for cloud-scale location services [J]. Distributed and Parallel Databases, 2013, 31(2): 289-319.
- [10] SHEN J X, CHEN L, WU Y, et al. Approach to Accelerating Dissolved Vector Buffer Generation in Distributed In-Memory Cluster Architecture [J]. ISPRS International Journal of Geo-Information, 2018, 7(1): 26.
- [11] FAN J, JI M, GU G, et al. Optimization approaches to mpi and area merging-based parallel buffer algorithm [J]. Boletim de Ciências Geodésicas, 2014, 20: 237-256.
- [12] HUANG X. Parallel Buffer Generation Algorithm for GIS [J]. J. Geol. Geosci. , 2013, 2: 115.
- [13] WANG T D, ZHAO L J, WANG L Z, et al. Parallel research and optimization of buffer algorithm based on equivalent arc partition [J]. Remote Sensing Information, 2016: 147-152.
- [14] BATTLE L, STONEBRAKER M, CHANG R. Dynamic reduction of query result sets for interactive visualization [C]//IEEE International Conference on Big Data. IEEE, 2013.
- [15] JUGEL U, JERZAK Z, HACKENBROICH G, et al. M4: A Visualization-Oriented Time Series Data Aggregation [J]. Proceedings of the Vldb Endowment, 2014, 7(10): 797-808.
- [16] WESLEY R, ELDRIDGE M, TERLECKI P T. An analytic data engine for visualization in tableau [C]//ACM Sigmod International Conference on Management of Data. ACM, 2011.
- [17] WICKHAM H. Bin-summarise-smooth: a framework for visualizing large data. Technical report [R]. RStudio, 2013.
- [18] WU E, BATTLE L, MADDEN S R. The case for data visualization management systems [J]. PVLDB 2014, 7(10): 903-906.
- [19] YAO X, ZHU D, YUN W, et al. A WebGIS-based decision support system for locust prevention and control in China [J]. Computers and Electronics in Agriculture, 2017, 140: 148-158.
- [20] YAO X, YANG J, LI L, et al. LandQv1: A GIS cluster-based management information system for arable land quality big data [C]//International Conference on Agro-geoinformatics. IEEE, 2017.
- [21] PAVLENKO A. Mapnik [OL]. <https://mapnik.org>.
- [22] YOUNGBLOOD B. GeoServer [OL]. <http://geoserver.org>.
- [23] KROPLA B. MapServer [OL]. <https://mapserver.org>.
- [24] 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.
- [25] LIN W, ZHOU H, XIA P. An effective NoSQL-based vector map tile management approach [J]. ISPRS International Journal of Geo-Information, 2016, 5(11): 1-25.
- [26] JINZHU G A, CHAOLI W B, LIYA L B, et al. A Parallel Multi-resolution Volume Rendering Algorithm for Large Data Visualization [J]. Parallel Computing, 2008, 31(2): 185-204.
- [27] LI J, WU H, YANG C, et al. Visualizing dynamic geosciences phenomena using an octree-based view-dependent LOD strategy within virtual globes [J]. Computers & Geosciences, 2011, 37(9): 1295-1302.
- [28] TANG W. Parallel construction of large circular cartograms using graphics processing units [J]. International Journal of Geographical Information Science, 2013, 27(11): 2182-2206.
- [29] GHOSH S, ELDAWY A, JAIS S. AID: An Adaptive Image Data Index for Interactive Multilevel Visualization [C]//IEEE International Conference on Data Engineering. IEEE, 2019: 1594-1597.



MA Meng-yu, born in 1992, Ph.D. His main research interests include GIS, geo-computation methods, high performance parallel computing, spatial data analysis and visualization.



JING Ning, born in 1962, Ph.D, professor, Ph.D supervisor, is a senior fellow of China Computer Federation. His main research interests include geographical information systems, database systems, spatial data analysis, and visualization.