引文格式：刘刚,陈根深,陈麒玉,等. 基于Web的大规模三维空间模型高效可视化关键技术[J]. 地理空间信息, 2023, 21(12): 8-13. doi:10. 3969/j.issn. 1672-4623.2023.12.002

基于Web的大规模三维空间模型高效可视化关键技术

刘　刚[1,2,3]，陈根深[1,2,3]，陈麒玉[1,2,3]，吴雪超[1,2,3]
（1. 中国地质大学（武汉）计算机学院，湖北 武汉 430074；2. 智能地学信息处理湖北省重点实验室，湖北 武汉 430074；
3. 智慧地质资源环境技术湖北省工程研究中心，湖北 武汉 430074）

摘　要：在网络环境下客户端对大规模的三维空间模型进行可视化时仍存在着渲染时间长、计算量大等诸多问题，表现为模型遮挡剔除计算效率低、模型数据传输效率低以及渲染不同模型时存在性能和质量不平衡问题等方面。针对上述问题，提出了针对不同的空间模型剔除粒度的优化选择策略、基于WebGPU计算管线的遮挡剔除算法和基于多线程的大规模三维空间模型数据快速调度方案，降低了离线遮挡剔除方法的预处理时间消耗，提高了模型切换的效率和场景渲染的速率。开发了基于Web的大规模三维空间模型可视化原型系统，开展了不同尺度的三维模型加载与可视化实验，结果表明，本文方法可以有效减少网络环境下大规模三维空间模型数据可视化的加载和渲染时间，增强用户体验。

关键词：大规模三维空间模型；可视化；WebGPU；场景调度；遮挡剔除
中图分类号：P208　　**文献标志码：**B　　**文章编号：**1672-4623（2023）12-0008-06

Key Technologies for Efficient Visualization of Large Scale 3D Spatial Models Based on Web

LIU Gang[1,2,3], CHEN Genshen[1,2,3], CHEN Qiyu[1,2,3], WU Xuechao[1,2,3]
(1. School of Computer Science, China University of Geosciences, Wuhan 430074, China;
2. Hubei Key Laboratory of Intelligent Geo-information Processing, Wuhan 430074, China;
3. Hubei Provincial Engineering Research Center of Intelligent Geological Resources and Environment Technology, Wuhan 430074, China)

Abstract: In the network environment, there are still many problems when clients visualize large-scale 3D spatial models, such as long rendering time and high computational complexity. These problems are manifested in low computational efficiency of model occlusion culling, low data transmission efficiency of models, and imbalanced performance and quality when rendering different models. To solve the above problems, we proposed optimization selection strategies for different spatial model removal granularity, an occlusion culling algorithm based on WebGPU computing pipeline and a large-scale 3D spatial model data fast scheduling scheme based on multithreading, which reduced the preprocessing time consumption of offline occlusion removal methods and improved the efficiency of model switching and scene rendering speed. We developed a web-based large-scale 3D spatial model visualization prototype system, and conducted experiments on loading and visualization of 3D models at different scales. The results show that the proposed method can effectively reduce the loading and rendering time of large-scale 3D spatial model data visualization in network environment, and enhance user experience.

Key words: large-scale 3D spatial model, visualization, WebGPU, scenario scheduling, occlusion culling

对于一个城市级或以上的基于网络的三维空间模型可视化系统，其渲染数据量非常巨大，可以达到TB以上的数量级。在网络环境下进行大规模场景渲染的过程中，模型的渲染部分会随着用户视角的改变而改变，浏览器需要快速地切换渲染场景，释放上一帧的同时需要加载当前帧[1-2]。网络浏览器客户端比较常用的模型可视化技术有Construct 2、WebGL、WebGL2、WebGPU等，可以直接通过浏览器加载模型，不需要安装其他的插件。但是面对大规模的三维空间模型时，进行实时交互可视化仍面临着一些技术难点，主要体现在以下几个方面：浏览器绘制性能弱、交互显示延迟、数据传输速度瓶颈等，导致数据加载慢、渲染时间长等诸多问题。为此，前人引入了一系列的渲染加速方法，包括遮挡剔除、隐面剔除和视锥体剔除、空间分割、动态加载[3-10]等。

在基于网络的模型可视化技术方面，2011年2月

收稿日期：2023-11-03。
项目来源：国家自然科学基金联合基金重点项目（U1711267，41942039）；地学长江计划面上项目（CUGCJ1810）。
第一作者简介：刘刚（1967—），博士，教授，博士生导师，主要从事空间信息工程、时空数据模型、三维可视化与可视分析方面的教学和科研工作，E-mail：liugang@cug.edu.cn。

首次发布了Construct 2，元素只能支持2D，绘制功能非常有限，其速度主要取决于CPU向GPU发出绘图命令的效率[11]。WebGL首次出现在2011年初发布的Chrome9和Firefox4中。WebGL2最早出现在2017年的Chrome56和Firefox51，WebGL2是将WebGL的功能提高到与OpenGL ES 3.0同等水平的更新[12]。WebGPU是一种新的图形渲染技术，被认为是当前的图形渲染技术中的一种高性能、跨平台的API[13]。相比于WebGL等上层API，WebGPU是一个更加精确的图形渲染API，完全开放了整个显卡的能力，使其具有更高的性能和效率。此外，WebGPU可以减少CPU的开销、对多线程有良好的支持、使用计算着色器可以将通用计算的功能引入Web、以及拥有全新的着色器语言等。

网络环境下遮挡剔除技术的研究起步较晚，研究重点主要放在对特定的遮挡剔除技术进行改进，以提高其性能、可用性和正确性[14-18]。近年来得益于GPU在图形计算方面的出众表现，使得将图形计算任务从CPU转移至GPU成为可能。Schütz[19]等就基于GPU进行点云渲染计算，这样既减轻了CPU繁重的计算负担，又提高了整体的渲染效率，其性能相较于硬件管道要高出一个数量级。网络环境下遮挡剔除有以下主流方案：①基于视点的遮挡剔除；②基于场景图的遮挡剔除；③基于图像处理的遮挡剔除；④基于深度排序的遮挡剔除；⑤基于矢量化的遮挡剔除；⑥基于面向对象的遮挡剔除；⑦基于光线追踪的遮挡剔除等。

对于局部小范围空间模型的可视化，地上建筑、地质模型等模型数据的数据量较小，一般不超过1 GB。此时，计算机可以一次性地载入并处理全部数据，使其得到更好的渲染效果。但对于海量模型，渲染数据会达到几十GB到几百GB[20-23]。因此，如何利用计算机有限的处理能力实现海量模型数据实时调度和渲染已成为面临的一个重要问题。杜莹[24]、Zhang L[25]和Ren Z[26]等对全球多分辨率虚拟地形仿真的关键技术进行了研究，初步实现了全球大规模地形数据的实时调度和渲染。Lebiedz J[27]等提出了基于计算机集群的虚拟地形并行化方案，有效提高了地形数据调度和渲染的性能，并实现了地形的网络化调度和渲染。赵建斌[28]根据Win32线程库和DirectX的多线程支持，提出了一种渲染帧的线程级并行化方法。李喆[29]根据线程数等于CPU核数的原则，提出了一种设置线程数量的方法等。虽然在基于网络环境下二维、三维空间信息可视化的方面已有较多的成果[23-26, 30-32]，目前多线程异步调度方法能提高CPU的利用率，但是该方法存在着对复杂场景渲染效率优化效果不明显，或者能提升三维场景的并行渲染效率，但却未提出相应的策略调度来保证负载平衡等问题。

综上所述，针对网络环境下大规模三维空间模型应用需求，本文的主要研究目标是使得网络浏览器客户端的大规模三维空间模型可视化系统能够快速、高质量地渲染模型，采用面向网络的三维模型遮挡剔除、渲染优化、并行加载等策略，解决网络环境下大规模三维空间模型可视化高效渲染及数据调度效率提升等相关问题。

1　总体技术路线

针对网络环境下浏览器客户端大规模三维空间模型高效可视化需求，主要开展3个方面的研究：

1）网络环境下多类型三维模型遮挡剔除策略。针对不同三维模型的结构和数据差异性，可以采用不同粒度的剔除策略，包括对象级剔除（Object-level Culling）、面级剔除（Face-level Culling）和像素级剔除（Pixel-level Culling）等。对于具有复杂结构的三维模型和结构简单的三维模型，需要开展相应的不同规模的剔除与渲染实验，通过网络环境下的实际测试确定合适的遮挡剔除策略，以提升整体剔除计算效率。

2）基于WebGPU计算管线的渲染优化方法。传统遮挡剔除算法（TOCA）中，视点发射的每条光线都需要与场景中的每个包围盒BVH（层次包围体，Bounding Volume Hierarchy）进行一次相交测试并计算该包围盒到视点的距离，从而导致遮挡剔除原始方法的计算量十分的巨大。将遮挡剔除原始方法中大量重复的计算单元由CPU移入GPU中做并行计算，通过并行计算的方式来减少时间消耗。随着WebGPU的计算管线API的开放，运用GPU在网络浏览器客户端进行并行计算成为可能，可以直接把所有的计算单元放入GPU中进行计算。WebGPU提供的管线技术，大大降低了CPU和GPU之间频繁通信带来的时间损耗，使得GPU算力能够更好地体现出来。

3）基于多线程的大规模数据异步并行加载和渲染方法。针对大规模三维空间模型的在可视化和人机交互场景中的计算效率问题，通过创建渲染线程的方法，进一步提升渲染和交互的流畅性（图2中PVS为潜在可见集potentially visible sets）。此外，可在内存中开辟多级缓存区，以减少硬盘读取，降低加载时间，并将异步并行加载数据量大的三维空间模型数据添加到渲染队列中等待渲染。

2 关键技术方法

2.1 大规模三维空间模型剔除粒度的选择方法

三维模型渲染剔除粒度指的是在三维渲染过程中，用于减少不必要的渲染计算的一种优化技术，其粒度表示为3种不同的级别：对象级剔除、面级剔除和像素级剔除。不同的剔除粒度可以通过结合使用来达到更好的优化效果，减少不必要的渲染操作，提高渲染效率。对象剔除和面剔除在三维模型渲染使用中，对象剔除通过检测模型包围盒与相机视锥体的相交来判断模型是否需要被渲染，从而减少不必要的渲染操作。面剔除则通过检测模型的每个面与相机视锥体的相交来判断面是否需要被渲染，同样也能减少不必要的渲染操作。对象剔除和面剔除的结合使用可以更有效地剔除不需要渲染的对象和面，从而提高渲染性能和效率。

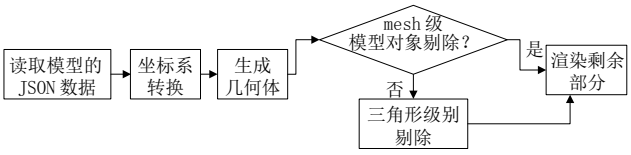


图1 三维模型剔除粒度选择流程

2.2 基于WebGPU计算管线的三维模型遮挡剔除优化方法

对于Web端的遮挡剔除算法，传统的CPU离线预处理方法计算缓慢，占用了大量的资源，渲染时间消耗较大。本文将传统离线预处理中的计算部分从CPU端移植到GPU端执行，提出了基于WebGPU计算管线的改进算法（improved algorithm based on the computing pipeline of WebGPU，IACPW）。并针对IACPW中计算管线创建开销过大的问题引入了离线遮挡剔除的思想和BVH场景树的改进算法（introduced BVH tree improvement algorithm，IBTIA），提出了调整计算策略的改进算法（Adjusted Computing Improvement Algorithm，ACIA）。3种方法的处理流程图见图2，基于WebGPU计算管线的改进算法主要将所有包围盒与射线的计算过程放入GPU中进行并行计算，引入BVH场景树的改进算法，在CPU中剔除掉了一部分不必要的计算过程，随后将剔除结果传入GPU中进行后续的计算，调整计算策略的改进算法在引入BVH场景树的改进算法的基础上将在CPU中剔除部分不必要计算过程的也放入GPU中进行剔除。由于相交测试计算是三维模型遮挡剔除过程中计算量最多的环节之一，这种调整策略可显著降低CPU的负载，提高了GPU的利用率，整体计算效率将得到有效提升。

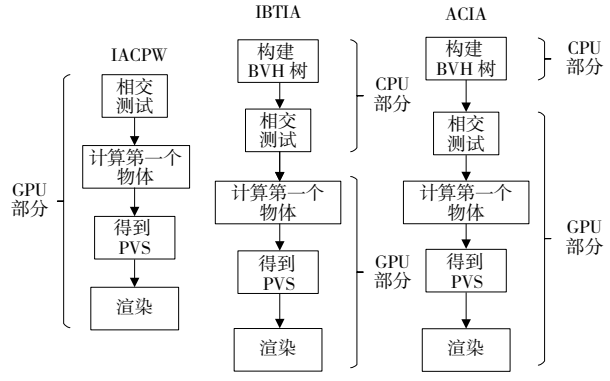


图2 三种模型遮挡剔除方法的处理流程图

2.3 基于多线程的三维模型数据动态调度策略

大规模模型数据读取时，通过单线程对大量的数据读写然后建立包围盒使得速度缓慢。本文提出基于多线程的大规模空间模型数据快速调度的方法，引入WebWorker的方法来实现多线程的功能，能够在JavaScript中创建多线程的机制。在多线程的机制中，主线程正在运行时，WebWorker线程在后台运行，并且两者可以并行工作不会相互受到干扰，等到WebWorker线程完成计算任务后将结果返回给主线程。主线程可以创建WebWorker线程并为其分配一些任务。这样可以把一些高计算量型或延迟较高的任务由WebWorker线程来负担，让主线程来处理一些界面上的交互功能，以免受到其他线程的影响。

在模型加载的过程中主要分为主线程和渲染线程。主线程主要完成场景数据的读取，对应用户的操作进行响应，根据相机的变化获取相应的PVS。子线程（渲染线程）主要是完成数据的加载和模型的渲染，以及模型数据资源的释放。在主线程向渲染线程的数据传递过程中，本文采用了一种新的数据传输方法，即WBufferC（WBuffer-based Conveying），来减少时间消耗。WBufferC会在内存中开辟一块buffer，且其长度可以动态增长，利用buffer自动管理内存长度的优点，可以对其重复利用以及解析不同长度的数据，而不需要自行申请并管理这些内存。

在数据的渲染过程中为防止数据的二次读取和二次渲染，在加载当前视角的可见集后释放上一视角的缓存数据，在确定当前视角的可见集合后计算场景中已有模型与前视角的可见集合的差集，然后剔除不可见的部分。这样做可以先行清空缓存区的数据，有足够的空间对后面的数据进行加载。同时对于场景中已有部分，计算当前视角的可见集与场景中已有模型的差集，在渲染进程中只需要加载计算后的差集，实现

模型数据的动态调度。

3　实验与结果

实验系统总体框架如图3，开发测试环境为Windows 11和Node16，硬件环境是核心处理器Intel Core-i7，显卡3090 Ti，内存64.0 GB。存储模块由MinIO、MySQL及Redis组成，在Windows 11上面部署MinIO用以存储模型的原始文件。采用MySQL存储模型的元数据并建立索引，引入Redis作为缓存。用户可以通过不同的端口访问该系统，该系统目前推荐的浏览器是Google Chrome Canary和Microsoft Edge Canary。

我们分别使用5种尺度三维地质体模型数据进行

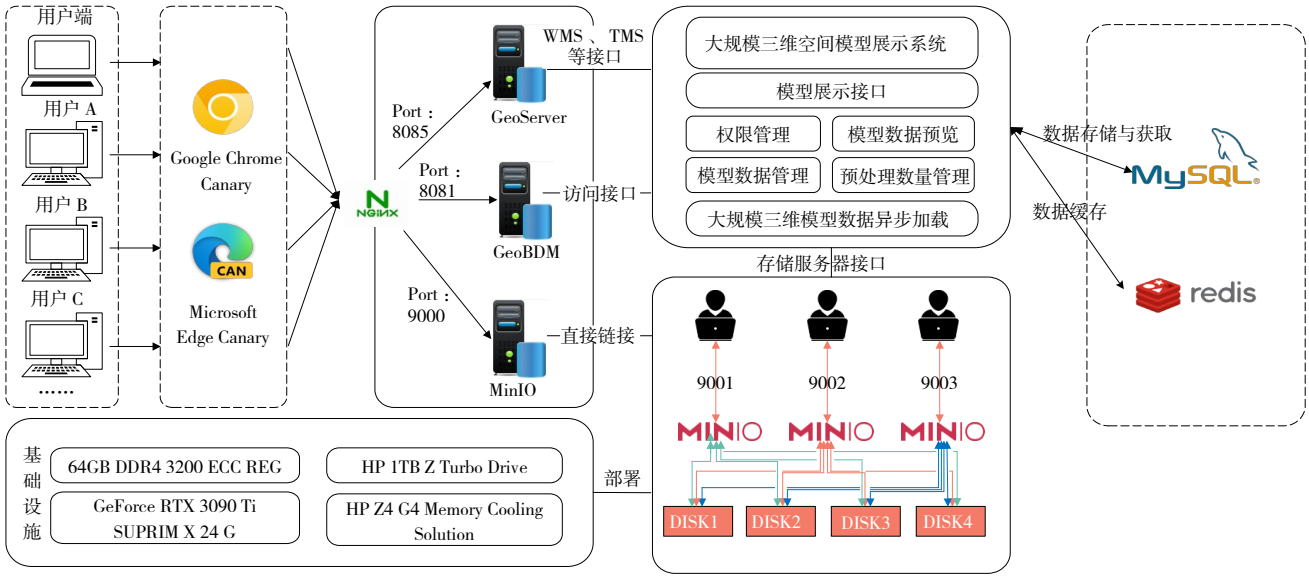


图3　实验系统总体框架

测试，设置从小到大的模型梯度，5种模型的比例尺分别为1∶5 000、1∶25 000、1∶50 000、1∶250 000、1∶500 000，其基本信息如表1所示。

表1　三维地质模型数据总体情况

模型名称	三角面数量/面	网格数/个	模型大小	比例尺
五指山地质模型	9 234 330	1 371	406 MB	1∶5 000
五指山地层体	27 512 307	2 684	1.22 GB	1∶25 000
普孟幅模型	46 731 152	4 433	1.88 GB	1∶50 000
黔东北模型	263 103 774	6 744	10.6 GB	1∶250 000
贵州省整体模型	90 415 179	17 773	8.12 GB	1∶500 000

在遮挡剔除粒度选择策略实验中，本文结合典型的三维地上建筑模型和三维地质模型进行了实验，选择适合的剔除方式。总体模型剔除粒度选择流程如图1所示，以实验区5 000栋，10 000栋，50 000栋和100 000栋三维建筑模型以及五指山地质模型（406 MB），五指山地层体（1.22 GB）和黔东北地质模型（10.6 GB）为实验对象。通过不同的渲染方法对不同数量的建筑物渲染后以mesh级别为剔除对象的结果和三角形级别进行剔除结果对比，发现以mesh级别为剔除对象时CPU、GPU利用率低，内存的使用量会大，加载时间长。在渲染不同规模大小地质模型后，三角形级别剔除的CPU、GPU利用率比mesh级别剔除结果高，内存的使用量和加载时间减小。这是由于建筑模型结构相对规范和简单，加载后以mesh级模型对象为剔除对象总体效率较高。对于地质体模型，由于其形态随机变化、结构复杂，加载后以三角形面片级别进行剔除总体效率较高。

在遮挡剔除实验中，表2为不同方法及不同大小模型实验的时间统计结果，其中-表示算法时间消耗过长。从中可以看出，基于WebGPU计算管线的改进算法的效果相较于原始方法有很大的提升，但是略差于离线遮挡剔除方法，引入BVH场景树的改进算法将基于WebGPU计算管线的改进算法与离线遮挡剔除方法结合后效果较好，再次对引入BVH场景树的改进算法进行改进的调整计算策略的改进算法效果最好。虽然实验模型的规模从小到大设置了一个梯度，但是基于WebGPU计算管线的改进算法的总耗时基本不变。总体而言，从计算速率来看，调整计算策略的改进算法在渲染大体量三维模型时为速率最快的遮挡剔除方法，而基于WebGPU计算管线的改进算法则为最为稳定的改进方法。

图4为表2中的数据柱状图显示，展示了不同方法对表中模型的计算时间消耗。其中TOCA为未经过任何优化的遮挡剔除原始方法，BOCA指经前人优化过后的离线遮挡剔除方法，而IACPW、IBTIA、ACIA为本文依次提出的3种改进遮挡剔除算法。从中我们

表2 遮挡剔除实验的不同方法时间消耗数据/ms

模型名称	遮挡剔除原始方法	离线遮挡剔除方法	基于WebGPU计算管线的改进算法	引入BVH场景树的改进算法	调整计算策略的改进算法
五指山地质模型	919.8	170.5	78.2	87.8	19.2
五指山地层体	1 434.9	537.2	68.5	513.4	32
普孟幅模型	2 180.1	616.3	67.5	777.6	40.1
黔东北模型	–	2 591.6	70.7	2 027.1	87.1
贵州省整体模型	–	782.2	74.4	785.9	88.8

可以直观地看出，IACPW渲染时间消耗较为平稳，IBTIA效果较差，甚至在部分模型的渲染上较前人提出的方法时间消耗更长，ACIA总体上而言计算时间消耗最小，效果最好。

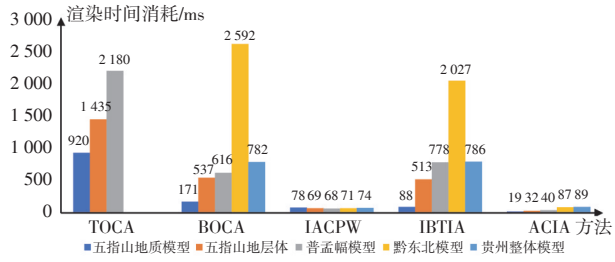


图4 遮挡剔除实验的计算时间消耗柱状图

在动态调度实验中，通过上述方法完成对5种不同尺度和不同大小的模型加载，本文方法均获得更高的加载速度（如图5）。以贵州省1∶500 000整体模型（8.12 G）为例，用传统方法的加载帧率为13 fps，使用本文方法的加载帧率达到40 fps。

采用实验区的100 000余栋三维建筑模型和典型三维地质模型（930 MB，面片数19，246，398个）为实验对象，进行地上下三维模型的渲染实验。较之传统方法，本文方法均取得了可视化性能的提升，加载帧率60 fps，总加载时间为2.6 s，达到预期效果。

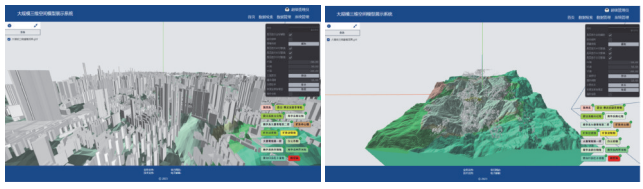


图5 地上建筑及地下地质体的联合实验

4 结 论

针对网络浏览器端大规模三维空间模型绘制性能弱、交互显示延迟等问题，在模型遮挡剔除粒度选择方法、遮挡剔除算法及动态调度等方面进行了优化设计，以提升三维模型的渲染效率和交互体验。

1）针对不同类型的空间模型选择的适合剔除粒度。我们根据建筑模型和地质模型的特点，分别优化确定了对象级和面片级的剔除粒度，在保证显示细节的前提下提高了整体三维模型渲染的效率。

2）提出了基于WebGPU计算管线的遮挡剔除算法。在CPU中构建BVH场景树，用WebGPU渲染管线进行相交计算并计算视点到筛选后包围盒的距离，有效减轻CPU的计算负担，充分利用GPU的计算能力，降低了离线遮挡剔除方法的预处理时间消耗。

3）基于多线程技术提出了一种大规模三维空间模型数据快速调度方案。引入WebWorker线程技术，使用主线程和渲染进程分别处理数据加载、剔除过程和场景资源的更新遍历，可以充分利用系统的多核资源，从而提高了渲染效率。

开发了基于WebGPU的大规模三维空间模型展示系统，系统综合利用了上面描述的关键技术，实现了不同尺度和规模的地质模型和地上建筑模型的快速加载和渲染，在实际工程中得到了较好的表现。

未来工作中，可以在硬件层面上继续发展，优化配置算法来提升网络环境下的数据读取和传输性能。目前WebGPU的管线配置存在着大量的参数，配置过程较为复杂，其计算时间消耗也部分依赖于场景树的深度，未来可以对场景树的结构进行优化，或者引入机器学习的方法对视角所见的模型部分进行预测。

参考文献

[1] 刘刚,吴冲龙,何珍文,等. 地上下一体化的三维空间数据库模型设计与应用[J]. 地球科学(中国地质大学学报),2011,36(2):367-374

[2] 黄科佳,李少杰,左尧,等. 基于四叉树索引和双缓存机制的GIS动态可视化模型设计与实现[J]. 测绘与空间地理信息,2018,41(4):102-105

[3] 莫剑,吴斌晖,俞友. 基于三维Web GIS的多模型地质灾害监测预警系统的设计与实现:以平江县加义镇潭湾村黄子口公路北侧山体滑坡为例[J]. 工程建设与设计,2022(17):156-159

[4] 占伟伟,黄露超,王辉,等. 一种面向大规模动态图形场景的渲染优化方法[J]. 信息技术与信息化,2021(8):37-40

[5] Genova K, Cole F, Vlasic D. Learning Shape Templates with Structured Implicit Functions[C]//Proceedings of the IEEE/CVF International Conference on Computer Vision. 2019:7 154-7 164

[6] 何虎,陈思. 晶体三维模型Web端展示[J]. 四川地质学报,

2021,41(4):670-677

[7] 田勇. 计算机图形图像处理关键技术应用研究[J]. 信息记录材料,2021,22(12):168-170

[8] 徐玉明,高海忠. 基于Web技术下的三维动画模型实时交互研究[J]. 电脑知识与技术,2021,17(26):118-120

[9] Choquette J, Gandhi W, Giroux O. NVIDIA A100 Tensor core GPU:Performance and Innovation[J]. IEEE Micro, 2021,41(2):29-35

[10] 彭燕青,董文煜,吴德聪,等. 三维可视化系统应用[J]. 电子技术与软件工程,2019(16):176-177

[11] Subagio A. Learning Construct 2[M]. Street Birmingham: Packt Publishing Ltd,2014

[12] Ghayour F,Cantor D. Real-time 3D Graphics with WebGL 2: Build Interactive 3D Applications with JavaScript and WebGL 2 (OpenGL ES 3. 0)[M]. Street Birmingham: Packt Publishing Ltd,2018

[13] Nguyen N, WebGPU the new Powerful Successor, the Future of Web API[D]. South-eastern Finland University of Applied Sciences,Finland,2022

[14] 刘一明,何晓曦,黄世贤. 大规模三维模型加速渲染技术的研究与应用[J]. 成都信息工程大学学报,2021,36(3):300-304

[15] Stein C, Limper M, Kuijper A. Spatial Data Structures for Accelerated 3D Visibility Computation to Enable Large Model Visualization on the Web[C]//Proceedings of the 19th International ACM Conference on 3D Web Technologies,2014:53-61

[16] 葛海明,张维,王小龙,等. 基于SfM的城市电缆隧道三维重建方法优化研究[J]. 图学学报,2023(2):1-12

[17] Compieta P, Di Martino S, Bertolotto M, et al. Exploratory Spatio-temporal Data Mining and Visualization[J]. Journal of Visual Languages & Computing,2007,18(3):255-279

[18] 郭向坤,林浒,刘继申. 一种CPU-GPU协同计算的三维地形实时渲染算法[J]. 小型微型计算机系统,2018,39(4):825-829

[19] Schütz M, Kerbl B, Wimmer M. Rendering Point Clouds with Compute Shaders and Vertex Order Optimization[J]. Computer Graphics Forum,2021,40(4):115-126

[20] 单杰,李志鑫,张文元. 大规模三维城市建模进展[J]. 测绘学报,2019,48(12):1 523-1 541

[21] 邵泽凤,邹青青,张荣华. 基于CGA的大规模三维城市模型构建方法研究[J]. 现代计算机,2020(12):44-47

[22] 王永志,刘鹏彧,杨路生,等. 盐岩地质三维建模及分析平台研发与实现[J]. 地下空间与工程学报,2016,12(6):1 586-1 592

[23] 李志祥,杨育林. 铁路线路三维模型可视化建设研究[J]. 电子世界,2021(22):20-21

[24] 杜莹. 全球多分辨率虚拟地形环境关键技术的研究[D]. 郑州:信息工程大学,2005

[25] Zhang L, Yang C, Liu S. Effective Techniques for Interactive Rendering of Global Terrain Surfaces[J]. IEEE Geoscience and Remote Sensing Letters,2005,2(2):215-219

[26] Ren Z, Ma C, Huo P, et al. Real-time Rendering of Global Multiple-resolution Inlaid Terrain Based on Tessellation[J]. Journal of Computational Information Systems, 2015, 10(16):5 899-5 909

[27] Lebiedz J, Mieloszyk K, Wiszniewski B. Real Terrain Visualization with a Distributed PC-Cluster[C]// International Conference on Parallel Processing and Applied Mathematics. 2005:349-356

[28] 赵建斌,李灵巧,杨辉华. 线程级并行计算在图形渲染引擎中的研究[J]. 计算机工程与设计,2011,32(12):41-43

[29] 李喆,郑晓薇. 多核并行技术在三维场景加载中的应用[J]. 计算机工程,2011,37(6):245-246

[30] 闫金珠,李少梅. 港口经济信息地图可视化方法应用研究[J]. 地理空间信息,2023,21(8):45-48,57

[31] 黄丙湖,王福源. 基于WebGIS的供需地图系统设计与实现[J]. 地理空间信息,2023,21(8):89-91,125

[32] 刘刚,吴冲龙,何珍文,等. 面向地质时空大数据表达与存储管理的数据模型研究[J]. 地质科技通报, 2020, 39(1): 164-174

（上接第7页）

[19] 刘警鉴. 基于SAR影像数据提取水稻面积的方法研究[D]. 南宁:南宁师范大学,2019

[20] Phan H, Le Toan T, Bouvet A. Understanding Dense time Series of Sentinel-1 Backscatter from Rice Fields:Case Study in a Province of the Mekong Delta, Vietnam[J]. Remote Sensing,2021,13(5):1-23

[21] Zhang R, Tang Z, Luo D, et al. Combined Multi-Time Series SAR Imagery and InSAR Technology for Rice Identification in Cloudy Regions[J]. Applied Sciences,2021,11(15):1-20

[22] 何泽,李世华. 水稻雷达遥感监测研究进展[J]. 遥感学报, 2023,27(10):2 363-2 382

[23] Otsu N. A Threshold Selection Method from Gray-Level Histograms[J]. IEEE Transactions on Systems, Man, and Cybernetics,1979,9(1):62-66