

文章编号: 1006—2610(2023) 06—0129—06

基于 BIM 和 WebGL 技术的异步场景漫游方法

邢洁莹 栗煜 韩娅娜

(中国电建集团西北勘测设计研究院有限公司, 西安 710065)

摘要: 随着页面渲染技术的快速发展, 云端 BIM 应用不断涌现, 由于浏览器处理能力和计算资源的限制, 为处理大体量 BIM 模型的流畅浏览与操作提出了新的挑战。为了解决对大型 BIM 模型的顺畅浏览和操作问题, 通过 WebGL 技术提出了一种 BIM 模型异步加载模式下进行场景路径漫游的方法, 实现了在网页端三维轻量化 BIM 模型的异步加载, 提高了模型帧率; 同时, 通过样条曲线算法, 使用关键点指定漫游路径, 提高了创建漫游路径操作的交互性; 最后, 通过视点的动态剔除、遮挡剔除等算法, 保证渲染漫游动画的流畅性。验证表明, 该方法可有效提高了大体量 BIM 模型路径漫游的流畅性与稳定性。

关键词: 轻量化模型; BIM; 路径漫游动画; WebGL; 固定帧率

中图分类号: TP391

文献标志码: A

DOI: 10.3969/j.issn.1006-2610.2023.06.022

Asynchronous Scene Roaming Method based on BIM and WebGL Technology

XING Jieyun, LI Yu, HAN Yana

(PowerChina Northwest Engineering Corporation Limited, Xi'an 710065, China)

Abstract: With the rapid development of page rendering technology, cloud BIM applications are emerging, and due to the limitation of browser processing power and computing resources, it poses new challenges for the smooth browsing and operation of large-scale BIM models. In order to solve the problem of smooth browsing and operation of large-scale BIM models, a method of scene path roaming in the asynchronous loading mode of BIM models is proposed through WebGL technology, which realized the asynchronous loading of 3D lightweight BIM models on the web side and improved the frame rate of the model. At the same time, through the spline algorithm, the roaming path is specified by key points, which improves the interactivity of creating the roaming path. Finally, algorithms such as dynamic culling and occlusion elimination of viewpoints are used to ensure the smoothness of rendering roaming animations. The verification results show that the proposed method can effectively improve the fluency and stability of the path roaming of large-scale BIM models.

Key words: lightweight model, BIM, path roaming animation, WebGL, fixed frame rate

0 前言

BIM (Building Information Modeling) 是一种基于数字化模型的建筑信息管理方法^[1], 与三维设计软件之间存在着紧密联系。由于三维设计软件的发展, 模型的复杂度随着模型的精度不断提升, 模型的规模与数据量日益扩展^[2], 导致在后续模型浏览

与操作阶段难以保证三维模型的快速与流畅渲染。三维场景漫游^[3]是指在虚拟三维环境中通过用户的操作, 以自由、流畅的方式浏览和移动, 仿佛身临其境地探索三维场景的过程。由于浏览器内存的局限, 对于大体量模型难以一次性全部加载, 在进行模型漫游展示时, 场景模型过大容易造成漫游卡顿、浏览效果差等问题。

针对模型传统加载慢、漫游方法卡顿的问题, 本文基于 BIM 和 WebGL 技术, 通过异步加载、样条曲线和视点剔除, 提出一种异步场景漫游方法, 实现三维模型场景漫游路径的可视化编辑和效率提升。

收稿日期: 2023-10-11

作者简介: 邢洁莹(1996-), 女, 陕西省安康市人, 工程师, 主要从事工程三维数字化研究工作。

基金项目: 陕西省重点研发计划(2023-ZDLGY-26)。

1 三维模型场景漫游现状

目前常规的场景漫游实现方法是设定摄像机漫游路径的关键点,通过样条曲线插值后的点作为路径,从而生成动画来进行场景漫游,或者基于 VR 技术,实现虚拟现实场景漫游^[4]。如果场景内的模型复杂程度较高,就会加长动画每帧所需的绘制时间,使得场景漫游动画的帧数下降,甚至出现卡顿的问题;即使在异步加载的环境模式下,镜头朝向一个尚未加载的模型区域时,由于操作耗时,也很容易导致漫游卡顿。为此,本文致力于探索一种基于 BIM 和 WebGL 的异步加载场景漫游方式,既可以在漫游时实现模型的异步加载,也可以保证漫游时的流畅性。

2 三维模型轻量化

设计模型是指在设计过程中创建的一种精确的包含大量几何信息的边界描述(B-rep)模型,在现有的计算机软硬件条件下,使用设计软件无法装配全专业不同格式的模型,且模型体量巨大、维护成本极高。因此需要使用轻量化的方法来优化仿真模型,摆脱设计软件对模型的桎梏,以达到对模型进行快速交互与实时渲染的目的。

2.1 数据压缩存储

BIM 模型文件包含了三维模型的几何数据、设计属性、空间分布与细节构造等信息,模型体量庞大,包含的建筑模型数量较多,严重影响了模型数据的网络传输效率与解析效率。

由于 BIM 模型的几何数据庞大,在 Web 端,难以一次性将数据全部加载。因此,使用数据分割的方法,异步地应用于 BIM 模型的加载以及优化。在异步数据加载的状态下,BIM 模型几何数据在前端与服务器的交互频率极高,对几何数据的优化存储就非常有必要。

本文提出一种新型文件存储结构,以模型结构树节点为基础,对于交互频率极高的几何数据,使用 LZMA 压缩算法^[5]进行高效的压缩,再存储至数据库,客户端在显示模型时会从服务器请求该压缩数据,从而大大减少几何数据的网络传输耗时。轻量化模型数据结构如图 1 所示。

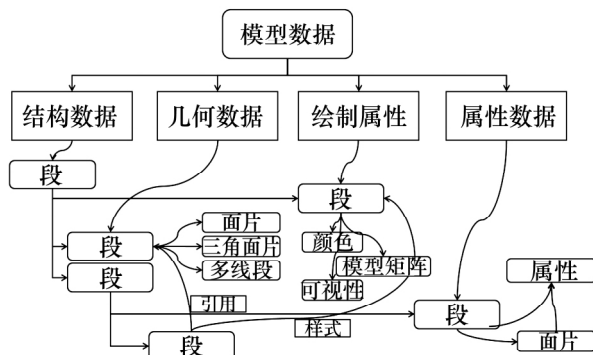


图 1 轻量化模型数据结构

2.2 模型结构优化

通过对多种大型模型设计软件的数据结构进行研究,对场景中大量重复出现的模型仅保留一份真实的几何实体数据,其他相同的模型则以引用的形式存在。这种引用和复用的机制有效地减少了存储大量重复物体的数据量,并提高了数据解析的效率。类似地,当场景中多数物体使用相同的材质时,该材质的数据也只保存一份,其他物体对该材质数据进行引用,从而提升了模型的渲染效率。

2.3 三角面片简化

网格简化是指在尽可能保持原始模型形状变化最小的情况下,降低模型复杂度的过程^[5]。当计算机需要渲染具有庞大三角面片数量的模型时,会消耗大量的计算资源,并且数据的网络传输和存储也会消耗巨大的资源。为了应对这些问题,目前的研究热点之一是对模型进行三角面片简化处理。

通过应用网格简化技术,可以降低模型中三角面片的数量,从而达到减轻计算负担、优化性能以及节省存储空间的目的。常见的网格简化近似方法分为以下三类:

(1) 顶点聚类算法:基本思想是,将模型顶点,无论按照晶格划分还是相近关系,将顶点进行聚类,属于同一类的顶点,由一个新的顶点进行替换。最常见的顶点聚类算法为 K-means 算法^[6],其是将距离作为度量,迭代地将顶点分配到聚类中心。顶点聚类算法拥有很高的效率和鲁棒性,算法的复杂度是线性的,但其缺点在于生成网格的质量不是特别令人满意。

(2) 重采样算法:基本方法是在网格曲面上选择新的采样顶点,通过连接这些顶点,能够构建出一个新的网格,进而减少原始网格的数据量。通过重采样我们能够获得想要的网格连接结构,其主要的

缺点是,如果采样模式与网格区域没有对齐,就会出现走样,拓扑结构可能发生变化,且不同的采样方式对于模型的拓扑保持与细节保持两个方面的侧重点不同。

(3) 增量式删除算法:典型的的就是 QEM(Quadratic Error Matrices,二次误差测度)模型简化算法。具体的操作方法有顶点删除(Vertex removal)法、边坍塌(Edge Collapse)法和三角面片坍塌(Triangle collapse)法等。

由于 K-means 算法生成的网格质量较差,重采样算法会修改模型的拓扑结构,本文采用 QEM 算法减少模型三角面片的数量,从而实现模型结构的优化。

3 场景的异步加载

3.1 WebGL 技术

WebGL(Web Graphics Library)最早是由 Mozilla 工程师所提出,创建一套用于 2D 和 3D 图形交互的 API^[6]。WebGL 是一种符合绝大多数浏览器 Web 标准的技术,它利用 GPU 加速渲染,使得几何和图像处理效果能够作为网页界面的一部分。作为一项广泛应用的技术,WebGL 支持在网页上绘制和渲染复杂三维图形,并且用户可以与之进行交互。

通过推广 WebGL 技术,开发者只需要在已经熟悉的 HTML 和 JavaScript 代码中添加一些额外的三维图形学接口,就能够在网页上展示三维图形,从而极大地降低了开发的复杂度。由于 WebGL 符合通用的 Web 标准,它兼容绝大多数现代浏览器,并且能够充分利用 GPU 来进行渲染,以实现更高效的图形处理和渲染效果。

WebGL 技术解决了现有 Web 交互式三维动画面临的两个问题。首先,它通过使用 HTML 脚本自身来制作 Web 交互式三维动画,无需依赖于任何浏览器插件的支持。其次,WebGL 技术利用底层的图形硬件加速功能进行图形渲染,这是通过统一、标准、跨平台的 OpenGL 接口实现的。

WebGL 技术不仅使得制作 Web 交互式三维动画变得更加便捷和无需插件支持,还通过底层图形硬件加速和标准化的 OpenGL 接口,实现了高效、统一和跨平台的图形渲染^[7]。

3.2 模型异步加载

对于大体量的模型而言,传统的直接加载方式

无法满足需求。因此,需要对模型进行预处理,先将模型以结构树节点为粒度进行分块,再将分块后的模型数据存储到服务器。在加载模型时,以摄像机视椎体扫描到的空间为基础,从服务器异步获取需要加载的模型的数据,异步的加载整个模型。通过这种机制,可以快速地打开大体量的模型,并且保证视椎体内观察到的模型始终是完整的。

3.3 固定帧率更新绘制

在某一特定的镜头内,可能存在大量几何网格数据模型,这导致每次更新和绘制视椎体内所有几何数据时需要耗费较长时间,从而降低了场景更新的帧率。由于显卡性能是主要影响场景更新绘制的因素,不同的显卡在相同传输帧数(FPS)的情况下,其所能承载的渲染压力也不同,因此进行绘制速度测试是达到最佳效果的先前条件。此后,在动画播放的同时,需要进行场景渲染,并且单次场景更新绘制的对象数量不能超过预设的上限。

如果没有使用固定帧率绘制机制,当某个镜头下需要显示的图形数据非常庞大时,用户进行漫游操作,系统需要循环调用绘制更新函数来不断更新和显示场景的图形数据。然而,每次更新绘制函数的耗时可能超过 0.05 s 甚至更长,这将导致帧率明显下降,甚至造成明显的卡顿感。

在使用固定帧绘制机制时,当需要连续进行场景更新时,系统会根据物体包围盒大小、物体之间的遮挡关系以及物体的像素大小等各种因素,暂时不绘制超过预设上限的几何数据,以保证连续场景更新时的帧率稳定。

在异步加载的策略下,通过应用多种加载机制,可以确保场景的更新效率,并保持场景的连贯性。

3.4 实验对比

本文通过实验统计了 3 个不同体量的模型在:① 使用异步加载策略;② 使用异步加载和固定帧绘制策略;③ 不使用策略这三种场景下的数据加载量。

表 1 实验对比

模型	使用异步加载	使用异步加载和固定帧绘制	不使用
模型 1	50Mb	49Mb	67Mb
模型 2	198Mb	143Mb	236Mb
模型 3	617Mb	423Mb	812Mb

实验结果表明在使用异步加载和固定帧绘制策略后,同一个模型在同样的场景下,需要加载和绘制

的数据量显著减少,提高了模型的加载效率和场景更新的流畅性。

同时,在同一场景漫游下,分别开启固定帧绘制和不开启固定帧绘制,测试显示帧率,结果如图 2 和 3 所示。

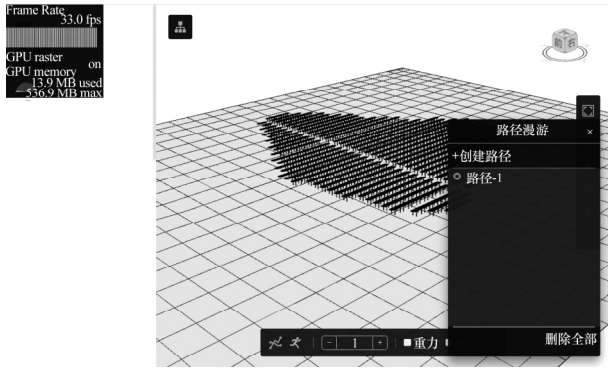


图 2 不开启固定帧绘制帧率

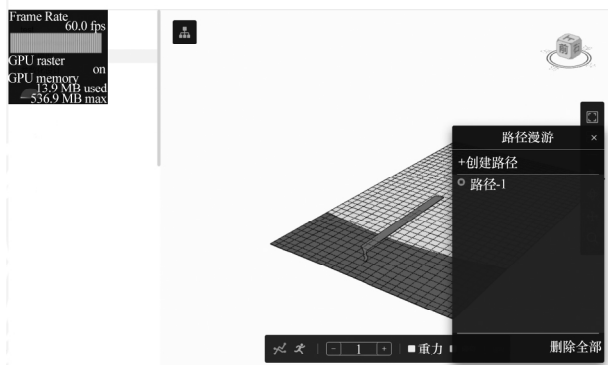


图 3 开启固定帧绘制帧率

实验结果表明在使用固定帧绘制策略后,场景的流畅度明显提升,保证了漫游显示动画的连贯性。

4 场景漫游算法及实现

4.1 路径规划算法

样条曲线是一种基于一组控制点而生成的一条曲线,通过这些点来控制曲线的整体形状。样条曲线主要可以分为两种类型:插值样条和逼近样条。插值样条通常用于数字化绘图或动画的设计,逼近样条一般用来构造物体的表面。Catmull-Rom 样条曲线是一种插值样条曲线,通过给定的控制点来精确地定义曲线。B 样条曲线(也称为基础样条曲线)是一种逼近样条曲线,它通过调整曲线的参数来逼近给定的控制点形状。

B 样条曲线通常使用 Bézier 曲线或 B 样条基函数来表示。B 样条是基于多项式的插值方法,使用多个控制点和基函数来计算曲线上的点。B 样条曲

线通过调整控制点的位置和权重,可以实现局部或全局的平滑性。一阶 B 样条线是直线,二阶及以上的 B 样条线是曲线但不过任何控制点。

Catmull-Rom 样条与 B 样条曲线很相似,不同在于 Catmull-Rom 样条是一种基于基函数的插值方法,使用 4 个相邻的控制点来计算曲线上的点。它是通过多个三次 Hermite 曲线的连接来定义的。Catmull-Rom 样条曲线通过保持控制点的一阶导数连续,具有局部平滑性,但不一定是全局平滑的。两者对比见图 4。

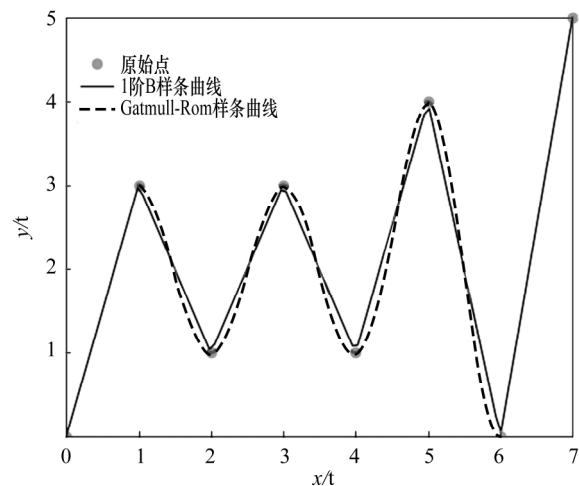


图 4 CatmullRom 样条方法

对于路径规划问题,需要保证在每一个控制点的平滑性,同时要保证路径经过每个控制点。因此本文选择了 Catmull-Rom 样条作为生成路径规划的算法。

4.2 可视化编辑算法

添加路径关键点后,系统会自动根据这些关键点在三维空间场景内生成相应的样条曲线。为了增强用户操作的交互性,每个路径关键点都配备了一个操作手柄,通过操作手柄可以自由调节调整关键点的位置,当关键点位置发生变化时,系统将实时更新并重新绘制与场景漫游路径相对应的新样条曲线。

本文使用 Catmull-Rom 算法,通过一组控制点来生成路径曲线,曲线中两个控制点之间的点坐标 $p(t)$ 计算式如下:

$$p(t) = [1 \ t \ t^2 \ t^3] \begin{bmatrix} 0 & 1 & 0 & 0 \\ -\tau & 0 & \tau & 0 \\ 2\tau & \tau-3 & 3-2\tau & -\tau \\ -\tau & 2-\tau & \tau-2 & \tau \end{bmatrix} \begin{bmatrix} p_0 \\ p_1 \\ p_2 \\ p_3 \end{bmatrix} \quad (1)$$

这里的 p_0 p_1 p_2 p_3 都是曲线上的点,但是要知道,上式只能计算出从点 p_1 到点 p_2 之间的点。式中的 t 取值范围为 $[0, 1]$,当 t 从 0 到 1 线性变化的时候,曲线就会从点 p_1 (此时 $t=0$) 慢慢移动到 p_2 (此时 $t=1$)。式中 τ 为影响扭曲程度,取 0.2。这个曲线另外一个特点就是,我们计算出来的点 p 的切向量和这个点的周围两个起点和终点的切向量是平行的。

当需要处理更多的点时,只需构造一对起始点、终点来构造这条关键帧曲线即可。

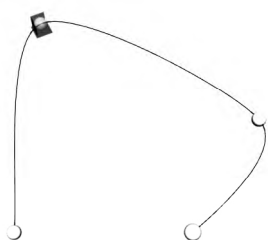


图5 路径的关键点可视化编辑

4.3 场景漫游算法和实现

本文所提出的漫游路径操作流程如下:首先,在三维场景中创建一个透视投影摄像机,并加载所需的模型,在模型加载完成后,对摄像机进行平移、缩放和旋转操作,以获得所需的关键帧;然后,将当前摄像机数据插入到路径漫游的关键帧序列中,重复以上步骤,直至获得完整的关键帧序列,包含了所有需要进行漫游的位置和朝向;接下来,对得到的关键帧序列中的相机位置点和朝向进行插值处理,以生成样条曲线,并对样条曲线进行采样,得到插值后的序列,通过这种方式,可以使漫游路径更加平滑和连

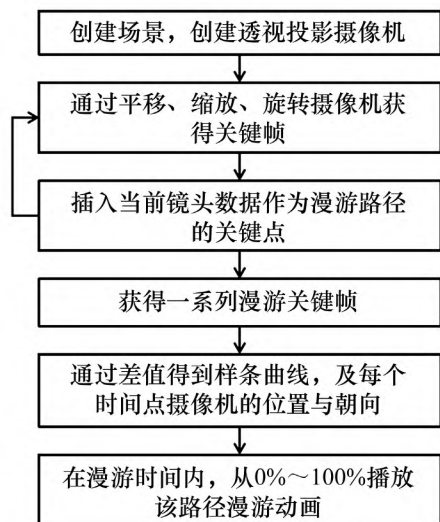


图6 路径漫游流程

贯;最后,设置漫游动画的时间参数,对漫游路径进行动画播放。根据设定的时间参数,系统会按照预定的速度和插值序列,自动推进摄像机沿着漫游路径进行运动,从而实现漫游效果。

4.4 路径漫游的实现和优化策略

在具体实现的过程中,主要分为以下几个步骤:

- (1) 通过工具条进入路径漫游模式;
- (2) 添加新的路径漫游,并定义该漫游的名称和总时间长度;
- (3) 鼠标操作模型使其移动到合适的位置后,插入漫游点;
- (4) 当漫游点总数大于等于 2 个时,即可完成路径设置,路径漫游如图 7 所示。

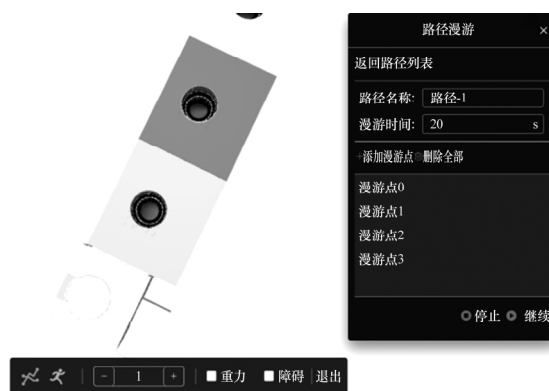


图7 路径漫游实现

通过选择指定漫游点,可以设置漫游的播放起始点和漫游点的删除。如图 8 所示,黑色的线是漫游的指示线,表示漫游的路径。

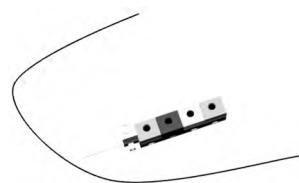


图8 漫游路径可视化

在确定好漫游路径后,就可以点击进行漫游动画的播放。通常情况下,系统会以 60 帧每秒的速率来更新摄像机的位置与朝向,完成路径漫游动画的效果。

本文的路径漫游效果是建立在异步加载三维模型的情况下实现的。在漫游过程中,视锥体可能会扫描到尚未加载的几何数据。为了解决这个问题,在每一帧中需要执行以下一系列步骤:

- (1) 获取几何数据;
- (2) 在主线程中加载几何数据;

(3) 创建绘制数据;

(4) 进行绘制。

然而,与已加载的几何数据相比,对于未加载的几何数据,额外的时间消耗和主线程阻塞很容易导致卡顿现象。因此,需要限制每一帧最多可以加载的几何数量。这个数量限制根据计算机硬件性能来进行动态调整。

同时,系统需要每帧对视锥体内的物体进行重渲染。如果视锥体内的几何数据过多,导致在主线程内执行的渲染操作消耗过多的时间,同样会导致主线程阻塞和卡顿现象的出现。因此,也需要限制每一帧最多可以渲染的几何数量,并根据计算机硬件性能进行动态调整。当同一帧要渲染的几何数量超过限制值时,系统会按照包围盒大小对要渲染的几何数据进行排序。然后,丢弃掉包围盒值较小的那部分几何数据,以确保在每一帧内渲染操作的效率和流畅度。

除此之外,还使用到了其他一些常用的优化策略:

(1) 基于视点的动态剔除

当物体离摄像机比较远时,部分模型在观察状态下会显得很很小。因此,在操作图形平台的时候可以不用渲染这些小的物体。

(2) 遮挡剔除

将三维空间的物体放到二维屏幕上显示时,场景中的各类物件会互相重叠,被遮挡的物体即便是在视锥体范围内,也不代表其是可见的。可以利用 DX11 的新特性,在 GPU 中并行执行计算每个物体的 Z-Buffer,得出三维空间内物体间的遮挡关系。

(3) 多细节层侧(LOD)

在三维模型轻量化时,可以进行数个精细度级别的转换。对于距离视点较远的物体,可以显示较低的精度,对于距离视点较近的物体,才显示最高的精度。如此可以减少模型渲染的压力,提高模型渲染的效率。

5 总结

本文基于 WebGL 技术,在网页端实现了三维

BIM 模型的异步加载,提高了模型帧率、确保模型加载性能;通过样条曲线算法,使用关键点指定漫游路径,实现了漫游路径创建过程中的交互操作;最后,通过视点的动态剔除、遮挡剔除算法,剔除被遮挡物体,减少了渲染几何数量,保证了渲染漫游动画的流畅性。对比实验结果表明,基于 BIM 和 WebGL 技术的轻量化模型异步场景漫游方法,提高了三维轻量化模型在网页端的加载效率,实现了路径漫游的可视化编辑。

参考文献:

- [1] 陈前,王玮.BIM 模型轻量化及三维显示关键技术研究[J].广东土木与建筑,2022,29(5):1-5.
- [2] 冯雨晴,奚雪峰,崔志明.基于 WebGL 的 BIM 模型轻量化研究[J].苏州科技大学学报(自然科学版),2021,38(4):72-78.
- [3] 郭稳稳,仲梁维.网页环境下无插件虚拟场景漫游的研究[J].软件工程,2021,24(6):46-49.
- [4] 叶玉萍.基于虚拟现实技术的三维校园漫游系统研究[J].电脑与信息技术,2020(2):14-16.
- [5] 易军凯,王雪峰,万静.代码变形的增强 LZMA 算法[J].计算机应用与软件,2015,32(5):302-306.
- [6] Wenbin Wu,Houkuan Huang."Adaptive grid k-means algorithm based on partial density peak clustering." Soft Computing 2021.
- [7] 戴佳佳,范丽鹏,庞明勇.特征驱动的三维网格模型自适应重采样算法[J].系统仿真学报,2019,31(5):853-860.
- [8] 郭敏鹏,王剑,杨少雄,等.基于 BIM+GIS 的水库下游洪水模拟与可视化方法[J].西北水电,2023(3):7-14,20.
- [9] 杨东升,费秉宏,孙春华.基于 BIM 技术的水电站地下洞室定位关键块体分析方法[J].西北水电,2022(2):81-85.
- [10] 李梦,孙阳.大石峡水利枢纽工程 BIM+GIS 平台建设[J].西北水电,2021(6):146-150,167.
- [11] 黄文钰,尚海兴.三维实景 GIS 平台在水电站设计中的应用探索[J].西北水电,2020(4):107-109.
- [12] 李鑫,蒋绮琛,于鑫,等.基于 BIM 轻量化技术的协同管理平台研究与实践[J].土木建筑工程信息技术,2020,12(3):59-64.
- [13] BAI S. Secondary developments based on BIM virtual technology [C]//International Conference on Information Computing and Applications.Berlin,Heidelberg: Springer,2013:165-173.