

# 基于 WebGL 的室内外智能感知可视化系统设计\*

周文超 张 鹏 李孟委

(中北大学仪器与电子学院 太原 030051)

**摘 要:**当前传统园区的二维平面技术表现形式单一、交互功能匮乏,难以满足多样化的用户需求。为适应智慧园区的发展模式,利用新一代 Web 三维绘图标准 WebGL 的优势,采用 GPU 硬件加速的 3D 图形库 Three.js 框架,设计了基于 WebGL 的室内外智能感知可视化交互系统。该系统架构于 B/S 模式下,借助 Three.js 技术实现场景的 3D 模型加载,融合最小包围球算法与基于四叉树分割的 LOD 算法,使三维场景的渲染速率提高了 2.2 倍;融合 FFmpeg、JSMpeg.js 及 Node.js 技术对视频流实时传输进行优化,使视频流延迟降低了 3.2 倍。对该系统三维模型的渲染速率、旋转、缩放及视频流帧率进行了验证,证明了基于 WebGL 的智能感知三维可视化方法的可行性与有效性。

**关键词:**三维可视化;WebGL;Three.js;智能感知

**中图分类号:** TN92      **文献标识码:** A      **国家标准学科分类代码:** 510.5025

## Design of WebGL-based indoor and outdoor intelligent sensing visualization system

Zhou Wenchao Zhang Peng Li Mengwei

(School of Instruments and Electronics, North University of China, Taiyuan 030051, China)

**Abstract:** The current traditional park's two-dimensional technology features a single form of representation and lacks interactive functions, making it difficult to meet the diverse needs of users. To adapt to the development model of smart parks, this paper leverages the advantages of the new generation Web 3D drawing standard WebGL and employs the GPU-accelerated 3D graphics library Three.js framework to design an indoor and outdoor intelligent sensing visualization interaction system based on WebGL. The system is built on a B/S architecture, using Three.js technology to achieve 3D model loading, and integrates the minimum bounding sphere algorithm and the quadtree-based LOD algorithm, which increases the rendering speed of the 3D scene by 2.2 times. By integrating FFmpeg, JSMpeg.js, and Node.js technologies to optimize real-time video stream transmission, the video stream latency is reduced by 3.2 times. This paper verifies the rendering speed, rotation, scaling, and video stream frame rate of the system's 3D models, demonstrating the feasibility and effectiveness of the WebGL-based intelligent sensing 3D visualization method.

**Keywords:** 3D visualization; WebGL; Three.js; intelligent perception

### 0 引 言

随着大数据时代的高速发展,Web 图形库<sup>[1]</sup>(web graphics library, WebGL)的出现为 Web 平台上的三维图形渲染带来了巨大的改变和优势。其基于 Web 标准,能在几乎所有兼容第 5 版本超文本语言(hyper text markup language 5, HTML5)的现代浏览器上顺畅运行,包括服务器端、客服端和嵌入式设备,实现了跨平台和跨设备的

兼容性;WebGL 运用图形处理单元(graphics processing unit, GPU)进行图形渲染<sup>[2]</sup>,利用硬件加速来实现高效的图形处理,针对复杂的三维场景实现更流畅、更快速的图形渲染。董志成等<sup>[3]</sup>基于 WebGL 开发了地质三维数字露头可视化平台,极大提升野外露头表征的分析效率与易用性。朱颖<sup>[4]</sup>基于 WebGL 技术开展了 3D 智慧园区可视化交互的初步研究,体现三维园区交互的可用性。高壮<sup>[5]</sup>基于 WebGL 技术开发了自动驾驶可视化系统,验证了

收稿日期:2024-05-31

\* 基金项目:测试仪器项目(2009ZCHF00102)资助

WebGL在3D可视化交互系统中的能力与优势。翟月昊等<sup>[6]</sup>应用Web GIS技术开发了室内外一体的三维GIS原型系统,实现了三维漫游等功能;宋关福等<sup>[7]</sup>介绍了基于WebGL技术的Web三维GIS技术,三维GIS技术与WebGL的融合推动了Web 3D应用的发展。各类基于WebGL技术的系统正迅猛发展,频繁展现出WebGL技术引人入胜的视觉奇观。

本文利用WebGL的优势,结合使用HTML5与层叠样式表(cascading style sheets, CSS)<sup>[8]</sup>等前端技术、传感器技术、实时视频流传输技术<sup>[9]</sup>、WebSocket技术<sup>[10]</sup>与数据库技术,通过ArcScene软件、Blender软件与Three.js构建三维场景,并融合最小包围球算法与层次细节模型(level of detail, LOD)算法<sup>[11]</sup>,提高了三维场景的渲染速率,实现了通过浏览器实时查看三维园区、三维办公室的空间化场景,环境监测数据以及员工考勤数据的数字化、图形化信息,推动了WebGL为大数据三维可视化<sup>[12]</sup>管理的深度应用。

## 1 相关技术

### 1.1 浏览器/服务器(B/S)结构

B/S结构模式<sup>[13]</sup>是继客户端/服务器结构(client/

server, C/S)模式之后兴起的一种网络结构模式。与C/S模式相比,将客户端功能转移至网页端的B/S架构,极大简化了系统设计、部署及维护流程。

对于室内外智能感知可视化平台而言,采用B/S架构具有显著优势。用户只需打开浏览器即可与智能感知系统实时交互,查看和分析监测数据。这不仅简化了用户操作,还确保了数据的实时性和准确性。当系统面临更新或修复任务时,只需开发人员在服务器端进行相应操作,即可将最新版本推送给所有用户,极大地提升了系统的维护效率和用户体验。

### 1.2 WebGL与Three.js

标准Web页面构成涉及HTML、CSS及JavaScript文件,而WebGL应用在此基础上,增添了三维模型数据与着色器,通过与WebGL API交互,输入到WebGL的渲染流程中,该流程对输入的顶点进行预处理,将变换后的顶点进行图元装配与光栅化,转化为像素点进行附加处理生成像素颜色和其他属性输入到帧缓冲区,最终在Web平台上实现三维内容的高速渲染与可视化,三维可视化<sup>[14]</sup>渲染管线如图1所示。

Three.js<sup>[15]</sup>是一个基于WebGL的JavaScript开源库,简化了WebGL的API复杂接口,能在浏览器中直接

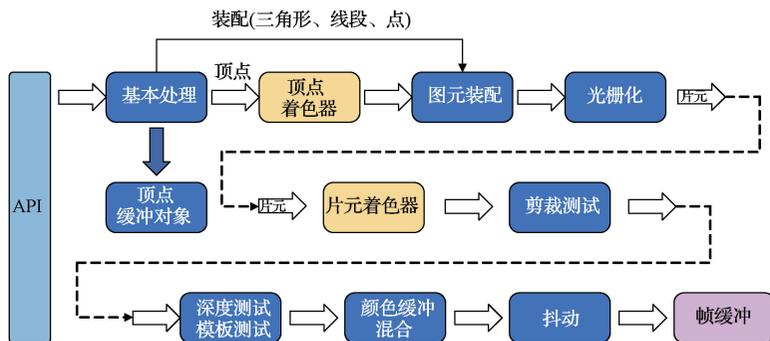


图1 WebGL渲染管线

Fig.1 WebGL rendering pipeline diagram

运行3D引擎,使得开发者更方便地导入和使用各种3D模型。Three.js的核心组成部分包括场景(scene)、相机(camera)和渲染器(renderer),将构建完成的三维模型加载到Web端界面中进行渲染,进而得到三维可视化<sup>[16]</sup>场景。

### 1.3 LOD算法

LOD算法是1976年提出的一种用于渲染大型场景的技术<sup>[17]</sup>。LOD算法的核心思想源自于人眼成像的特性,即人眼在观察物体时会关注近处的细节,而对远处的物体关注较少。因此,LOD算法在渲染过程中,依据物体离摄像头的远近,动态选取不同细致程度的模型——远方物体采用简模以提高效率,而近处物体则加载高细节模型以保证视觉效果。

这种方法带来的主要好处是显著提高了三维场景的

加载和渲染速度。在部署LOD算法前,需先对3D模型实施多级简化,创造多样精细度的模型变体。运行时,依据模型与摄像机间距,匹配最适细节级别模型加载,缩减加载及渲染周期,加速整体渲染进程。

### 1.4 视频流实时传输

实时视频流传输技术是现代监控系统的核心组成部分,涵盖了从视频信号的采集、高效编码、稳定传输到最终在Web端的高质量解码播放的全过程,确保用户能够近乎实时地观察和分析监控场景。这一过程首先通过高灵敏度的摄像头进行视频采集,捕捉现场的每一处细节。紧接着,视频编码技术如H.264、H.265被应用于压缩视频数据,既保证了视频质量,又显著降低了所需带宽。传输环节利用了诸如实时消息协议(real-time messaging protocol, RTMP)、网页实时通信(web real-time communica-

tions, WebRTC)等协议,这些协议在确保低延迟的同时,还具备适应不同网络条件的能力,使得视频流能够在不稳定或带宽受限的网络环境中依然流畅传输。在接收端,高效的解码技术配合智能缓冲策略,即使在网络波动时也能平滑播放视频,避免了卡顿和画面断裂现象,确保了观看体验的连续性和一致性。

此外,现代 Web 端的视频播放器不仅支持 4 K 视频的播放,还融入了用户交互设计,如时间轴快进倒退、截图保存、多画面同时监控等功能,进一步丰富了用户的监控操作体验。

## 2 室内外智能感知系统设计

### 2.1 系统架构设计

本文设计的系统针对校园建筑、办公室等模型及环境监测、员工考勤等数据,集成多种交互功能进行可视化展示。系统采用 B/S 模式下的三层架构<sup>[18]</sup>。在 B/S 模式中,3 层架构为前端视图层、服务器层和数据库层,系统总体架构如图 2 所示。

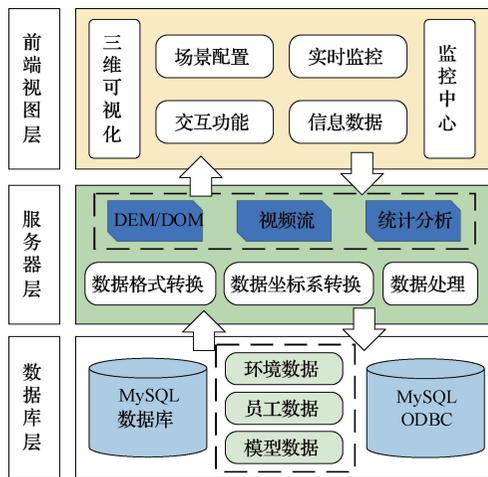


图 2 系统总体架构

Fig. 2 System overall architecture

前端视图层为系统架构的最顶层部分,主要是用户感知交互的界面,运用 Three.js 技术搭建了校园主楼建筑及办公室两个三维可视化场景,将三维校园建筑、三维办公室、摄像头视频流实时传输以及信息数据动态展示给用户,可以通过交互按钮进行两个场景的切换,对单个场景的三维模型可以进行移动、旋转及缩放操作,构建了满足人机交互的三维环境。

服务器层位于前端视图层与数据库层之间,主要负责前端视图层与数据库层之间的数据交互。服务器端利用 JavaScript 链接数据库与前端界面,确保传感器数据经处理后存入数据库,随即通过服务器传递至 Web 端展示。在室内外智能感知系统中,核心处理涉及两个场景的三维模型加载与渲染,以及视频流与数据的同步展示。

数据库层是信息系统的管理层,主要是针对环境数据、员工考勤数据以及模型数据进行存储与维护。通过 MySQL 来存储关于用户操作、员工信息、权限等业务数据。针对上传的和已经发布的实时数据进行存储、查询、修改、删除等。

### 2.2 系统功能设计

室内外智能感知系统由数据采集与处理模块、三维建模与渲染模块、实时交互用户界面和数据分析可视化展示 4 大模块组成,具体功能模块如图 3 所示。

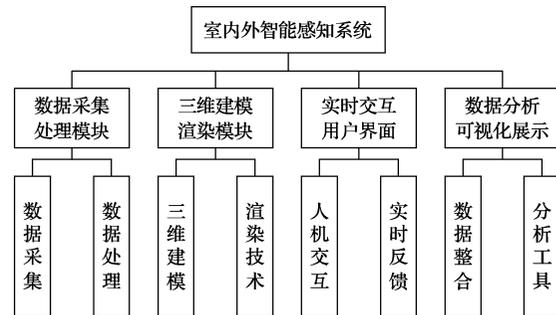


图 3 系统功能模块

Fig. 3 System functional modules

数据采集与处理模块主要包含数据采集和数据处理。数据采集主要有七合一空气质量传感器数据、BIM 模型数据、GIS 数据。数据处理即对采集的数据进行清洗、格式转换、坐标系统一、地图配准、模型优化等工作,以便于后续的三维模型重建和实时数据展示。

三维建模与渲染模块主要包含三维模型创建和渲染技术。三维模型利用 3D 建模软件或编程技术(如 3ds Max<sup>[19]</sup>、Maya、Blender 或基于 Unity/Unreal Engine 的程序化建模)进行创建,并由 Three.js 导入 Web 端进行显示。渲染技术应用光照、纹理、阴影等效果,提升模型的真实感和视觉效果,如使用物理渲染(physically-based rendering, PBR)技术。

实时交互用户界面模块主要包含了人机交互与实时反馈。该界面提供用户友好的界面,允许用户通过鼠标、键盘等设备与三维场景互动。并且系统能即时响应用户的操作,更新视图并提供相关信息,如传感器状态、系统提示等。

数据分析与可视化展示模块主要通过数据整合与分析工具,对数据进行实时分析,将业务数据、监控数据与三维模型结合,通过图表、动画等形式展示实时变化情况。

## 3 室内外智能感知系统实现

### 3.1 三维模型创建与加载优化

本文选用了 Three.js 的 GLTFLoader 加载器去加载 GLB 模型,选择 ArcScene、Blender 作为园区建筑模型、办公室建筑模型制作工具,制作三维模型的步骤如下:

1)将中北大学校区矢量数据导入 ArcScene,将矢量

- 数据拉伸为对应的三维建筑模型,并导出 wrl 格式文件;
- 2) 利用 Blender 对三维校园主楼模型、办公室模型进行模型细节、纹理材质的绘制;
  - 3) 运用 Blender 将模型导出为适合 Three.js 使用的 GLTF 或 GLB 文件格式;
  - 4) 模型压缩,选用基于 node.js 的 gltf-pipeline 插件压缩 GLTF 或 GLB 文件。

在 Three.js 中通过 GLTFLoader 模型加载器加载 GLTF 或 GLB 模型,由于已经将模型进行压缩处理,无法直接将压缩后的模型文件通过 GLTFLoader 模型加载器加载,因此需要经过 DRACOLoader.js 进行模型转换,转换完成后,由 Three.js 创建模型加载事件,同时为模型加载事件添加监听器,首先进行模型、材质、纹理的加载,当加载完毕时调用监听器,通过该工具把更复杂的建模时间直接添加到 Three.js 创建的三维场景中,最后在 Web 浏览器中进行渲染显示。

### 3.2 LOD 算法优化

在处理三维建筑模型时,应用了基于四叉树的多级 LOD 技术,该技术能根据观察者视角自动调节模型的精细度。主要结构特征始终保持高清展示,而随着视角接近,非核心细节渐进式增强,实现了视觉效果的连贯与细节层次的丰富。基于四叉树的多级 LOD 算法过程分为节点评价、裂缝处理和渲染。流程如图 4 所示。

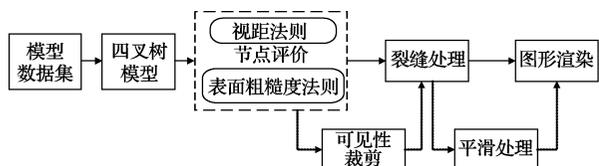


图 4 LOD 算法流程

Fig. 4 LOD algorithm process

LOD 优化技巧常以二维地面为基准,根据其至观察点的直接距离与预设界限设定分辨率,此原理同样适用于三维建筑呈现优化。区别于二维,三维模型因空间形态多样性,直接测定各模型至视点距离  $d$  较为复杂。引入最小外接球算法,可求得三维模型的最小外接球中心及半径,进而确定球心至视点距离。将此半径与距离的比率作为 LOD 算法的判断标准,通过与预设阈值的对比,来调整模型的可见性状态。

- 1) 预先设置阈值  $t$ ;
  - 2) 遍历 JSON 内包含的对象集  $n$ ,并针对每个对象应用最小包围球算法来获取其包围球  $s[j]$ ;
  - 3) 计算包围球半径  $r$  和球心到视点距离  $d$  的比值  $x$ ;
  - 4) 比较  $x$  与  $t$  的大小,设置 object3D.visible 属性值。
- 如式(1)所示,  $s[j].r$  为最小包围球的半径长度,  $(x_v, y_v, z_v)$  为视点的三维坐标,  $(x_s, y_s, z_s)$  为最小包围球的球心三维坐标。定义阈值  $m$  判断条件,当  $m > 0$ ,激活模型渲染过程,当  $m < 0$ ,禁止模型渲染。

$$m = \frac{s[j].r}{\sqrt{(X_v - X_{s[j]})^2 + (Y_v - Y_{s[j]})^2 + (Z_v - Z_{s[j]})^2}} - t \quad (1)$$

针对校园主楼模型,特别定制适宜的临界值以优化显示。通过遍历其子集并算取主楼至摄像机的最小包围球心距,依据此距离来调配对象的可视属性——贴近限定值则设定 visible 为真,予以展示并执行渲染;超出门槛则置 visible 为假,隐藏且避免渲染。此法依据建筑模型实体实行 LOD 场景优化,旨在减轻 GPU 负荷。算法伪代码描述如下。

算法 1: LOD 优化算法

```

Input: Three.js, camera, thresHold
Output: visible
1. var thresHold = setThresHold();
2. for each object in Three.js render Array{
3.   var distance = calculateDistance(camera, position, object, boundingSphere, center);
4.   var radius = getRadius(object, boundingSphere, radius);
5.   radius/distance <= thresHold ? visible = false : visible = true; }
    
```

Three.js 中引入上述 LOD 优化算法,Three.js 根据模型与相机的距离来显示三维模型对象。在 LOD 配置文件中,初始化一个 LOD 对象,并通过调用 insertStage 方法来配置不同距离下的模型展示。这里,依据模型至摄像头的间距 meshRange 进行层次安排,并确保各阶段按间距正确排序。随后,利用 selectModelForRange 函数,依据当前观测的 range 值,从 LOD 系统中取出适配的模型实例。该过程伴随着一个 refreshView 操作,它接受摄像头作为参数,确保根据摄像机的实时位置更新显示内容。每当摄像头位置变动,重新执行 refreshView 过程,即可动态获取并展示与视距相符的模型等级。其流程如图 5 所示。

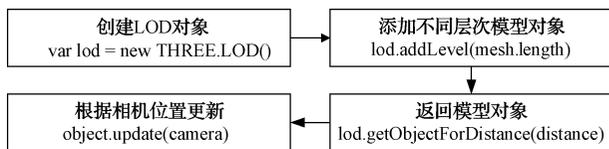


图 5 LOD 模型对象显示流程

Fig. 5 LOD model object display process

Three.js 引入 LOD 优化算法后,主楼模型层次分为 LOD0, LOD1。渲染结果如图 6 所示。

### 3.3 视频流实时传输方案及优化

基于实时流协议 (real-time streaming protocol, RTSP)、RTMP、超文本传输协议 (hypertext transfer protocol, HTTP) 的 Web 端视频流实时传输播放,都存在着一定程度的缺陷,其中 RTSP 协议需要 IE 浏览器安装

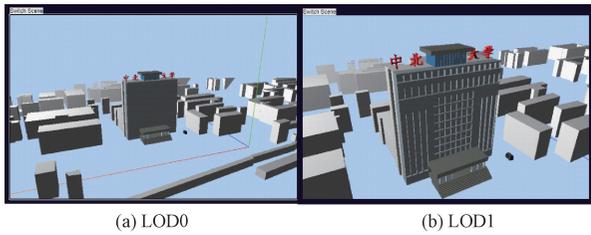


图6 主楼模型层次渲染结果

Fig. 6 Hierarchical rendering results of the main building model

COM 插件或特定视频播放器实现, RTMP 协议需要 Flash 的支持, HTTP 协议则是存在 10 s 以上的较大延迟, 均并不能满足“无插件”和“实时监控”的要求。因此, 借助 FFmpeg 这一开源工具, 实现数字音视频的录制、转换及流化处理, 设计了一种无需额外插件的实时监控方案, 该方案核心融合了 FFmpeg、JSMpeg.js 及 Node.js 技术。其工作流程为使用 FFmpeg 读取摄像头视频源, 并将其编码为 MPEG1 视频流; 使用 Node.js 创建一个 Web-Socket 服务器, 用于接收 FFmpeg 发送的编码视频流, 并将其转发给客户端; 在客户端中使用 JSMpeg.js 连接到 WebSocket 服务器, 并在 Web 端播放接收到的视频流。视频流实时传输流程如图 7 所示。

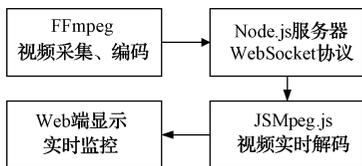


图7 视频流实时传输流程

Fig. 7 Real-time video streaming process

上述方案仍然存在 1~2 s 的延迟问题, 本文采用了如下优化措施: 首先, 通过 NVIDIA NVENC 硬件加速技术降低视频编码的延迟; 其次, 采用视频数据分块传输的方法, 减少单次大数据传输的阻塞风险; 此外, 配置 Web Assembly 解码器对视频流进行高效的实时解码; 最后, 根据网络状态动态调整缓冲区大小。在网络状况较好时, 减小缓冲区以降低延迟; 在网络状况较差时, 增大缓冲区以避免卡顿。通过这些优化操作, 有效地减少了视频流的延迟, 确保了视频流的平滑播放。

## 4 系统测试

### 4.1 软硬件环境配置

系统的使用环境为 13th Gen Intel(R) Core(TM) i7-13700HX 2.10 GHz 的处理器, 运行内存为 16 GB, 显卡为 NVIDIA GeForce RTX 4070 Laptop GPU 的 64 位操作系统的 PC, 选用 IntelliJ IDEA 2023.1.1 开发工具, 运行环境为 Windows11 平台下的 Google Chrome 浏览器。

### 4.2 系统功能测试

当用户打开室内外智能感知可视化浏览器时, 摄像头自动完成视频推流, 传输速度为 30 fps, 在 Web 端实时显示视频流监控画面。同时主体部分展示园区三维场景, 包括了室内外温度、湿度、CO<sub>2</sub>、PM2.5、PM10、CH<sub>2</sub>O 和 TVOC 等实时环境监测数据、员工考勤数据、摄像头实时监控、设备状态等, 如图 8 所示。



图8 园区三维场景可视化界面

Fig. 8 Visualization interface of the park's 3D scene

当用户点击转换场景交互按钮时, 将会跳转到办公室场景, 摄像头监控切换至对应的视频流, 如图 9 所示。



图9 办公室三维场景可视化界面

Fig. 9 Visualization interface of the office 3D scene

对室内外智能感知可视化平台进行功能测试, 测试结果表明, 室内外智能感知可视化平台系统能够实现摄像头切换、视频流暂停回放、图像数据自动化显示、三维场景切换、人机交互<sup>[20]</sup>等功能, 提高了环境监测、员工考勤信息查看效率, 满足了智能感知可视化平台数字化、智能化的基本要求。室内外智能感知可视化平台功能测试内容如表 1 所示。

### 4.3 系统性能测试

采用 3 种不同的视频流传输方案, 对视频流实时传输性能、延迟时间进行对比测试, 方案 1: EasyDarwin + FFmpeg + VLC; 方案 2: FFmpeg + JSMpeg.js + Node.js; 方案 3: FFmpeg + JSMpeg.js + Node.js 优化, 实验结果如表 2 所示。

表 1 室内外智能感知可视化平台功能测试

Table 1 Function testing of indoor and outdoor intelligent perception visualization platform

测试内容	测试操作	预期结果	测试结果
人机交互	随机点击交互按钮	界面交互操作正常、无操作延迟	通过
摄像头切换	点击室内外切换按钮	视频流正确切换,播放正常	通过
三维场景切换	点击 Switch Scene 按钮	三维场景正确切换模型显示正常	通过
数据查询	鼠标在数据图表中移动	鼠标经过环形图、柱状图、曲线图时显示数据	通过
三维场景界面控制	在三维场景中进行鼠标点击、拖拽等	用户通过鼠标控制三维场景变换	通过
数据采集实时显示	观察数据	数据随采集变化而变化	通过
视频流暂停、回放	点击监控界面暂停按钮、进度条任意位置	视频流暂停、回放正常	通过

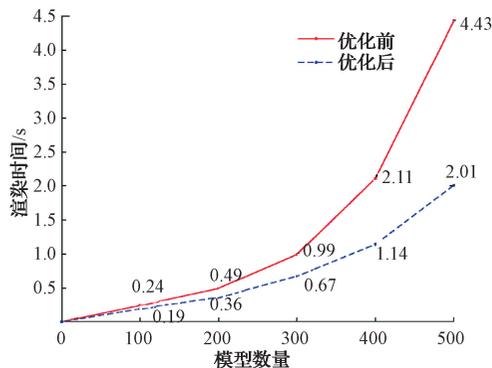
表 2 视频流延迟时间对比

Table 2 Comparison of video streaming latency (s)

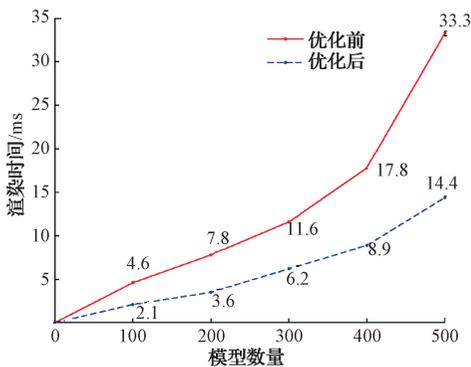
视频类型	方案 1	方案 2	方案 3
本地视频 D:/dd.mp4	1.89	1.43	0.47
摄像头视频 HP Wide Vision HD Camera	2.51	1.98	0.61

基于 FFmpeg、JSMpeg.js、Node.js 的视频流实时传输方案优化后,视频流播放延迟得到明显优化。

通过校园、办公室等模型渲染来验证本文融合最小包围球算法与 LOD 算法方案的优势,实验结果如图 10 所示。



(a) 校园模型渲染对比结果  
(a) Comparison results of campus model rendering



(b) 办公室模型渲染对比结果  
(b) Comparison results of office model rendering

图 10 模型渲染时间对比

Fig. 10 Comparison chart of model rendering times

在相同三维模型场景中,随着模型数量的增加,使用最小包围球算法与 LOD 算法进行优化后的渲染速率明显快于优化前的渲染速率,并且随着模型数量的持续增加,优化后的渲染效率更高,使三维场景的渲染速率提高了 2.2 倍,提高了三维场景的渲染速度与平台可视化的效率。

## 5 结论

本文基于 WebGL、Three.js 3D 引擎,使用 B/S 框架,设计并实现了三维可视化的室内外智能感知系统。根据园区实际环境,通过 ArcScene、blender 软件构建模型,融合最小包围球算法与 LOD 算法,加快了三维模型场景的渲染速度;融合 FFmpeg、JSMpeg.js 及 Node.js 技术优化视频流传输,降低了视频延迟。与传统的三维可视化系统相比,该系统实现了三维园区的网页可视化,解决了传统可视化管理形式单一的问题,为用户查看环境信息提供了便利,同时大模型渲染速度提升了 2.2 倍,视频流延迟降低了 3.2 倍,具有较高的应用价值。

## 参考文献

- [1] 范鹏飞,贺鹏飞,张颖,等. 基于 Web3D 的车间可视化系统设计与实现[J]. 信息技术,2024(3):43-47.  
FAN P F, HE P F, ZHANG Y, et al. Design and implementation of a workshop visualization system based on Web3D [J]. Information Technology, 2024(3):43-47.
- [2] 张羽西. 基于 Web 的隧道三维可视化设计[D]. 成都: 电子科技大学,2021.  
ZHANG Y X. Web-based 3D visualization design of tunnels [D]. Chengdu: University of Electronic Science and Technology of China, 2021.
- [3] 董志成,陈恭洋,印森林,等. 基于 WebGL 的数字露头可视化表征与识别系统:以无人机倾斜摄影依奇克里克剖面为例[J]. 科学技术与工程,2024,24(11):4633-4642.

- DONG Z CH, CHEN G Y, YIN S L, et al. WebGL-based digital outcrop visualization and identification system: A case study of the Yiqikelike section using UAV oblique photography [J]. Science, Technology and Engineering, 2024, 24(11): 4633-4642.
- [4] 朱颖. 基于 WebGL 的智慧园区三维可视化设计与实现[D]. 上海: 华东师范大学, 2022.
- ZHU Y. Design and implementation of 3D visualization for smart parks based on WebGL [D]. Shanghai: East China Normal University, 2022.
- [5] 高壮. 基于 Web 的自动驾驶可视化系统的开发与应用[D]. 广州: 广东工业大学, 2022.
- GAO ZH. Development and application of a web-based autonomous driving visualization system [D]. Guangzhou: Guangdong University of Technology, 2022.
- [6] 翟月昊, 苗立志, 张朋东. 室内外一体化的三维 GIS 原型系统设计与开发[J]. 计算机与数字工程, 2023, 51(11): 2535-2540.
- ZHAI Y H, MIAO L ZH, ZHANG P D. Design and development of an integrated indoor and outdoor 3D GIS prototype system [J]. Computer and Digital Engineering, 2023, 51(11): 2535-2540.
- [7] 宋关福, 钟耳顺, 吴志峰, 等. 新一代 GIS 基础软件的四大关键技术[J]. 测绘地理信息, 2019, 44(1): 1-8.
- SONG G F, ZHONG ER SH, WU ZH F, et al. Four key technologies of new generation GIS basic software [J]. Geomatics and Information Science of Wuhan University, 2019, 44(1): 1-8.
- [8] 吴怡, 李中, 庞照宇, 等. 基于 Web 端的三维井筒几何模型及信息可视化方法[J]. 科学技术与工程, 2024, 24(10): 4070-4083.
- WU Y, LI ZH, PANG ZH Y, et al. Web-based 3D wellbore geometric model and information visualization method [J]. Science, Technology and Engineering, 2024, 24(10): 4070-4083.
- [9] 朱骏侠. 基于 WebGL 的三维智慧管网系统的设计与实现[J]. 测绘通报, 2021(4): 146-150.
- ZHU J X. Design and implementation of a 3D smart pipeline system based on WebGL [J]. Surveying and Mapping Bulletin, 2021(4): 146-150.
- [10] 唐群, 张星刚. 利用 WebSocket 技术实现 B/S 端与 CAD 的即时通信[J]. 信息与电脑(理论版), 2024, 36(3): 30-33.
- TANG Q, ZHANG X G. Real-time communication between B/S end and CAD using websocket technology [J]. Information and Computer (Theoretical Edition), 2024, 36(3): 30-33.
- [11] 丁田妹, 徐发达. 三维模型中多层次细节 LOD 技术研究[J]. 电子制作, 2021(18): 35-37.
- DING T M, XU F D. Research on multilevel detail LOD technology in 3D models [J]. Electronics Manufacturing, 2021(18): 35-37.
- [12] 冯登超, 秦焕禹, 曾湧, 等. 基于三维可视化空中走廊体系的低空空域航图绘制研究[J]. 电子测量与仪器学报, 2018, 32(4): 58-64.
- FENG D CH, QIN H Y, ZENG Y, et al. Research on urban low altitude airspace charting based on three-dimensional visualization skywalk system [J]. Journal of Electronic Measurement and Instrumentation, 2018, 32(4): 58-64.
- [13] 程志广. 基于 WebGL 的设备三维可视化系统研究与实现[D]. 南昌: 南昌大学, 2022.
- CHENG ZH G. Research and implementation of 3D equipment visualization system based on WebGL [D]. Nanchang: Nanchang University, 2022.
- [14] 宁泓淘, 金永, 刘静静. 固体火箭发动机绝热层三维可视化技术研究[J]. 国外电子测量技术, 2022, 41(10): 80-84.
- NING H T, JIN Y, LIU J J. Research on 3D visualization technology of thermal insulation layer in solid rocket engines [J]. Foreign Electronic Measurement Technology, 2022, 41(10): 80-84.
- [15] 秦治贤. 基于 Three.js 的虚拟实验室设计与实现[D]. 桂林: 桂林电子科技大学, 2022.
- QIN ZH X. Design and implementation of a virtual laboratory based on Three.js [D]. Guilin: Guilin University of Electronic Technology, 2022.
- [16] 徐志远. 基于高光谱影像数据库的海洋地质三维可视化测试仿真[J]. 国外电子测量技术, 2019, 38(5): 17-23.
- XU ZH Y. 3D Visualization test and simulation of marine geology based on hyperspectral image database [J]. Foreign Electronic Measurement Technology, 2019, 38(5): 17-23.
- [17] 黄若思, 李传荣, 冯磊, 等. 基于几何的 WebGL 矢量数据三维渲染技术研究[J]. 遥感技术与应用, 2014, 29(3): 463-468.
- HUANG R S, LI CH R, FENG L, et al. Research on geometry-based WebGL 3D rendering technology

- for vector data [J]. Remote Sensing Technology and Application, 2014, 29(3): 463-468.
- [18] 刘天财,杨宇青,马继承,等. 基于 WebGL 的智慧社区三维可视化应用研究[J]. 科技资讯, 2023, 21(24): 233-238.
- LIU T C, YANG Y Q, MA J CH, et al. Research on 3D visualization application of smart communities based on WebGL [J]. Science & Technology Information, 2023, 21(24): 233-238.
- [19] 张贝贝. 基于 3DsMax/Unity/Vuforia SDK 的高校三维定位 AR 导览系统设计[J]. 电子测量技术, 2021, 44(3): 77-82.
- ZHANG B B. Design of a 3D positioning AR navigation system for universities based on 3Ds Max/Unity/Vuforia SDK [J]. Electronic Measurement Technology, 2021, 44(3): 77-82.
- [20] 陈仁钧,费敏锐,杨傲雷. 面向人机交互的手势指向估计方法[J]. 仪器仪表学报, 2023, 44(3): 200-208.
- CHEN R J, FEI M R, YANG AO L. Gesture pointing estimation method for human-computer interaction [J]. Chinese Journal of Scientific Instrument, 2023, 44(3): 200-208.

## 作者简介

周文超, 硕士研究生, 主要研究方向为大数据三维可视化、三维渲染、智能感知。

E-mail: 2278511010@qq.com

张鹏, 副教授, 博士生导师, 主要研究方向为智能感知与系统集成、自动控制、嵌入式系统等。

E-mail: sxyczhangpeng@126.com

李孟委, 教授, 博士生导师, 主要研究方向为石墨烯特种传感技术、新结构射频 MEMS 器组件技术、新原理 MEMS 导航器组件及微系统技术等。

E-mail: lmwprew@163.com