

基于 Web 的 Revit 模型轻量化和大屏显示研究

汪学清, 董 泽, 陈 博, 赵云猛

(中国矿业大学(北京)力学与建筑工程学院, 北京 100083)

摘要: 实现建筑信息模型在浏览器中的显示功能对于推动 BIM 技术在工程现场的实际应用具有重要价值。通过对 Revit 的二次开发, 将 Revit 模型以 GLTF 格式储存并采用 Draco 压缩算法进行轻量化处理, 在 Three.js 引擎中完成解析和重构模型, 从而实现模型在浏览器中的显示功能。在谷歌 Chrome 浏览器中实现模型优化和交互功能, 之后对多种浏览器的兼容性进行测试, 结果表明交互功能和优化效果良好。采用 Three.js 引擎实现了 BIM 模型在数据大屏系统中的展示, 提高了模型的美观度和模型与数据的关联性。研究结果表明: Revit 模型在浏览器中的轻量化显示具有可行性; BIM 模型轻量化显示与数据大屏技术相结合具有未来研究价值。

关键词: 建筑信息模型; 二次开发; Revit; GLTF; 浏览器

中图分类号: TP391 **文献标志码:** A **文章编号:** 1673-4602(2024)01-0073-09

Research on the lightweight and large screen display of Web-based Revit model

WANG Xueqing, DONG Ze, CHEN Bo, ZHAO Yunmeng

(School of Mechanics and Architectural Engineering, China University of Mining and Technology (Beijing), Beijing 100083, China)

Abstract: Implementing the display function of building information model in the browser is of great value to promote the practical application of BIM technology in engineering field. Through the secondary development of Revit, this paper stores the Revit model in GLTF format, uses the Draco compression algorithm for lightweight processing and completes the parsing and reconstruction of the model in the Three.js engine, so as to realize the display function of the model in the browser. The model optimization and interaction function are implemented in the Google Chrome browser, and then the compatibility of multiple browsers is tested. The results of the test show that the interaction function and optimization work well. The Three.js engine is used to realize the display of BIM model in the data large-screen system, which improves the aesthetics of the model and the correlation between the model and the data. The research results of this paper show that the Revit model is feasible for lightweight display in the browser, and the combination of BIM model lightweight display and data large-screen technology has research value in the future.

Key words: building information modeling; secondary development; Revit; GLTF; browser

收稿日期: 2022-09-01

基金项目: 深海矿产资源开发利用技术国家重点实验室开放课题项目 (2021-HYSDDYB-001)

作者简介: 汪学清 (1971-), 男, 山东成武人。副教授, 主要从事岩土工程方面的研究。E-mail: wxqcase@163.com。

随着计算机技术的进步,建设行业的信息化发展势头不可阻挡。BIM(Building Information Modeling)技术具有可视化、参数化、可模拟性等诸多优势,成为建设行业的首选。美国作为最早应用 BIM 技术的国家,为 BIM 的应用推出了很多标准,这也为 BIM 在全球的推广做出了贡献^[1]。同时我国住房和城乡建设部也颁布了许多政策和标准^[2-4],以鼓励和支持建筑行业应用 BIM 技术。现阶段,BIM 技术在工程应用中依然存在许多问题,施工人员希望在施工现场轻松便捷地浏览三维模型,但施工现场环境往往较难满足 BIM 软件的应用。因此,在 Web 端(浏览器)浏览建筑模型成为研究应用的热点,对于指导工程施工工具有重要意义。

WebGL 技术可被用于创建具有复杂 3D 结构的网站页面,通过 JavaScript 语言编写网页代码即可实现三维模型的展示^[5]。HTML5,WebGL 以及 CSS 技术的出现为在网页浏览三维模型并进行交互提供了可能性。前人做了如下研究:AHYUN 等^[6]基于 WebGL 搭建的三维模型展示平台实现了 3D 空间信息平台展示功能。贺佳佳^[7]将轻量化技术与 Revit 二次开发结合起来,开发出的 BIM 轻量化模型显示平台基本实现了 Web 端的轻量化显示。余宇深^[8]在全桥建模的基础上,提出了以 JSON 形式进行轻量化并实现 Web 显示的方法。阎超^[9]将 Revit 参数化模型工程信息输出为 XML 格式,并实现模型在 Web 端重构与展示。同时 SHIROLE 等^[10]也通过 XML 形式实现了对 BIM 模型信息的访问,为格式转换格式提供了新的方向。王占军等^[11]通过采用 GLTF 格式实现模型数据转换功能,引入 3D Tiles 技术实现海量三维模型基于图形引擎系统的集成与展示功能。吕婧等^[12]验证了 BIM 模型数据转换为 GLTF 格式的可行性,为本文研究提供了参考。综上,GLTF 传输格式作为连接 BIM 模型与 Web 端模型的“桥梁”应用于 Web 端模型显示具有可行性,并且相比于 XML 格式具有巨大优势,因此选择 GLTF 格式作为中间格式。

要在保证模型真实性的前提下实现 Web 端的轻量化显示,必须经过模型格式转换和渲染交互两个关键环节,其中,须对插件导出的 GLTF 文件进行轻量化处理。Web 端显示思路如图 1 所示。

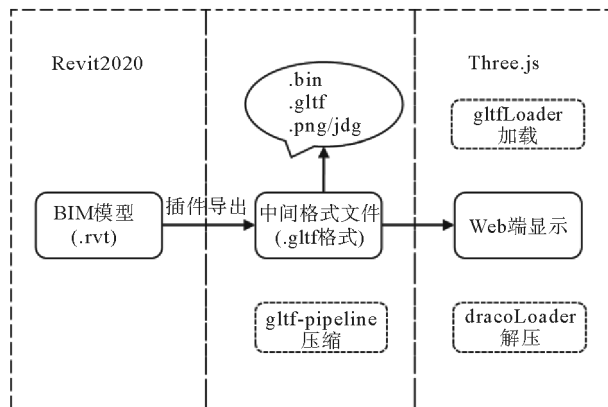


图 1 BIM 模型轻量化显示流程

1 模型轻量化

实现 BIM 模型的轻量化,即实现模型文件体量的压缩。此过程分为两步:①借助二次开发技术开发自定义插件将 Revit 模型导出为 GLTF 格式的中间文件;②通过内置 npm 载入 gltf-pipeline 插件压缩 GLTF 文件,实现模型轻量化。

1.1 插件导出

Revit 模型建立之后,模型几何和属性信息都储存在 .rvt 格式的文件中,浏览器无法进行读取。借助 Revit API 开发插件将 Revit 模型的几何、材质等信息储存在 GLTF 文件中。

插件开发主要分为如下 3 个步骤:①在 Visual Studio2019 内新建类库,命名为 RevitexportGltf;②在类库内引用 RevitAPI.dll,RevitAPIUI.dll,并通过 IExternalCommand 外部命令,加载 Excute()函数,借助 Revit 内提供的 CustomExporter 自定义导出接口;③编程结束,在 Revit 内通过 Add-InManager 加载 .dll 文件,查看插件运行结果情况。开发流程如图 2 所示。

在 CustomExporter 类中,重载 OnPolyMesh(),OnMaterials()等 15 个方法,实现模型中点、边界、材质等元素的提取。几何信息的获取主要是获取三角网格的顶点坐标、顶点索引、法线向量、法线向量索引、几何体中心、唯一标识符 ID;属性信息的获取是在导出材质贴图后,提取出模型的 name、userData、标识符 ID 等,以上几何信息和属性信息最后都以 Array 的形式存储在 GLTF 文件中,获取信息的过程如图 3 所示。

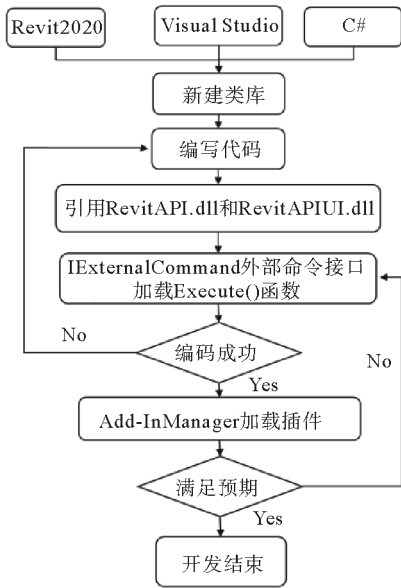


图 2 插件导出二次开发流程

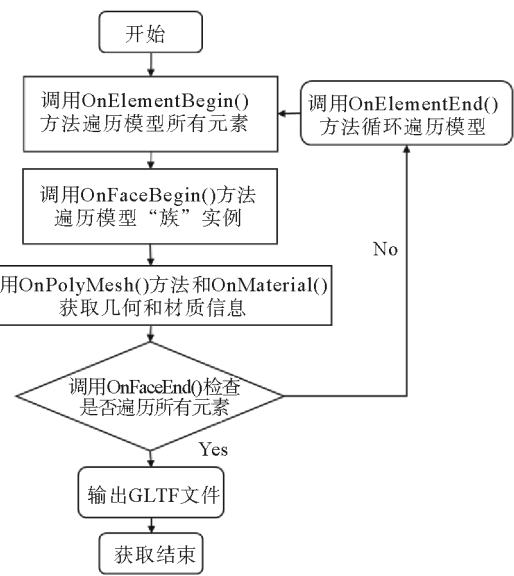


图 3 模型信息提取过程

在信息提取过程中,首先调用 OnElementBegin() 和 OnElementEnd() 循环遍历模型的所有元素和“族”实例对象,之后调用 OnFaceBegin() 函数获取模型的 face 数据,调用 OnPolymesh() 和 OnMaterial() 函数获取模型的 Geometry 与 Material,最后通过 OnFaceEnd() 判断是否遍历完模型中的所有构件,若没有则返回 OnElementEnd() 函数进行继续遍历;若已经遍历完模型的所有元素就导出 GLTF 文件,完成格式转换。

1.2 模型文件数据轻量化

模型成功转换为 GLTF 格式后,文件中产生了许多冗余数据,使导出的文件相比之前的文件体量产生了不同程度的增长。因此,通过 Draco 压缩算法对模型数据进行轻量化处理,减小模型文件体量。

Draco 是谷歌开发的一种大幅加速 3D 数据网格的编码、传输和解码的库,主要应用于压缩和解压 3D 网格和点云数据,能有效改善 3D 图形网格的存储和传输。Draco 的底层压缩基于 Edgebreaker 算法,Edgebreaker 通过一系列步骤遍历网格。对于每一步,算法都需要访问并编码一个尚未访问的三角形网格。在每个阶段,输入网格被划分为不相连的区域,这些区域可以共享一个顶点,但没有边共享。Edgebreaker 通过分析顶点 v 与栅边 g 、边界 B 之间的关系,可以分为下面五种情况^[13]:C,L,E,R,S。五种标识符如图 4 所示。

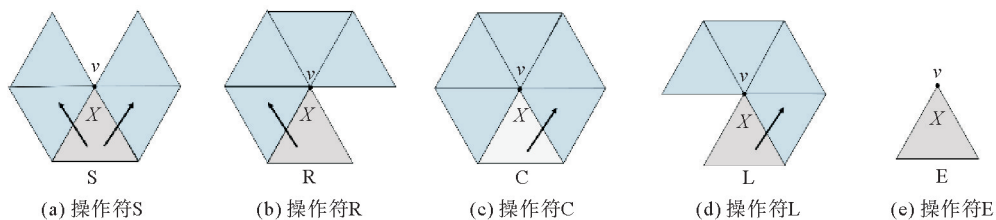


图 4 操作符分类

五种标识符分别代表着五种遍历三角形网格的方式:图 4(a)中操作符 S 表示两边网格都未被遍历;图 4(b)中操作符 R 表示左边网格未被遍历,右边网格已被遍历;图 4(c)中操作符 C 表示左右两边网格都未被遍历;图 4(d)中操作符 L 表示左边网格已被遍历,右边网格未被遍历;图 4(e)中操作符 E 表示两边网格都已遍历。

如图5所示,对于一个有边界的简单网格进行压缩时,Edgebreaker算法起始于绿色的门,跟随蓝色箭头表示的路径依次进行遍历,并借助图4所示的五种操作符方法遍历三角形网格,直到将所有网格遍历。

2 模型 Web 端重构、交互与优化

Three.js 作为一种简化使用 WebGL 技术的方式^[14],近几年被国内外程序员广泛使用。故本文在 Three.js 框架内实现轻量化模型在 Web 端的重构、渲染与显示,渲染流程如图6所示。

首先在 Three.js 框架中构建三维场景(Scene)、相机(Camera)、渲染器(Renderer)和光源(Light)等主要元素,之后导入轻量化后的 GLTF 格式文件,接着在 VScode 中利用 Three.js 框架中的 gltfLoader 和 dracoLoader 对 GLTF 文件内模型数据进行解析,最后通过添加方法等,在浏览器中实现模型重构与交互。

模型重构之后,为了方便浏览和查看其属性信息,可以通过在 VScode 添加控件等为模型增加交互功能。其中包括平移、缩放、旋转、拾取、高亮与属性表格等。

2.1 几何变换: 平移、缩放与旋转

模型首先要通过几何变换中的平移、旋转、缩放或组合变换实现将坐标系由局部坐标系转换到世界坐标系。

1) 平移。在平移操作过程中需要进行矩阵变换。假如场景中存在点 A , 坐标为 $(X, Y, Z, 1)$, 经过平移后的坐标为 A_1 , 坐标为 $(X_1, Y_1, Z_1, 1)$ 。过程中点按照平移矩阵 T 平移, 在各个轴方向的位移分量分别为 X_t, Y_t, Z_t , 则平移前的点 A 和平移后的点 A_1 关系如下:

$$[X_1 \ Y_1 \ Z_1 \ 1] = [X \ Y \ Z \ 1] \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ X_t & Y_t & Z_t & 1 \end{bmatrix} \quad (1)$$

其中平移矩阵 T :

$$T = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ X_t & Y_t & Z_t & 1 \end{bmatrix} \quad (2)$$

2) 旋转。在旋转操作中,三个主要因素是旋转轴、旋转方向和旋转角度。假设以 z 轴为旋转轴,逆时针旋转 θ 后,则旋转前的点 $A(X, Y, Z)$ 和旋转后的点 $A_1(X_1, Y_1, Z_1)$ 关系如下:

$$\left. \begin{aligned} X_1 &= X \cos\theta + Y \sin\theta \\ Y_1 &= Y \cos\theta + X \sin\theta \\ Z_1 &= Z \end{aligned} \right\} \quad (3)$$

3) 缩放。假设三个坐标轴的缩放因子分别为 Z_x, Z_y, Z_z , 缩放矩阵为 R , 则缩放前的点 A 和缩放后的点 A_1 关系如下:

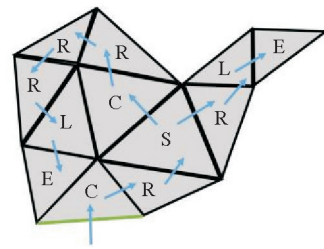


图5 Edgebreaker 网格压缩示意

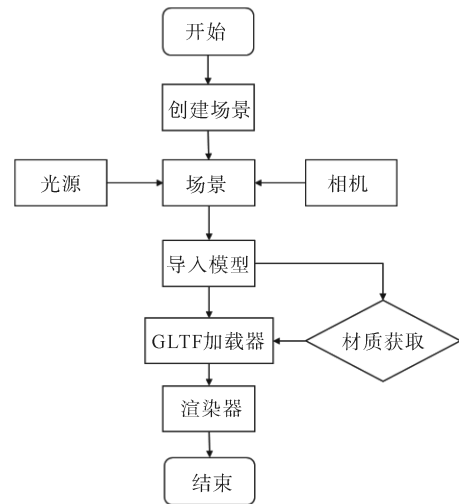


图6 Three.js 渲染流程

$$\begin{bmatrix} X_1 \\ Y_1 \\ Z_1 \\ 1 \end{bmatrix} = \begin{bmatrix} Z_x & 0 & 0 & 0 \\ 0 & Z_y & 0 & 0 \\ 0 & 0 & Z_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \quad (4)$$

其中缩放矩阵 R :

$$R = \begin{bmatrix} Z_x & 0 & 0 & 0 \\ 0 & Z_y & 0 & 0 \\ 0 & 0 & Z_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (5)$$

2.2 坐标系转换

模型坐标系经过几何变换之后,还需要依次经过视图变换、投影变换、视口变换,将世界坐标系转换为标准设备坐标系,再转为屏幕坐标系,以实现模型坐标的获取,转换流程如图 7 所示。

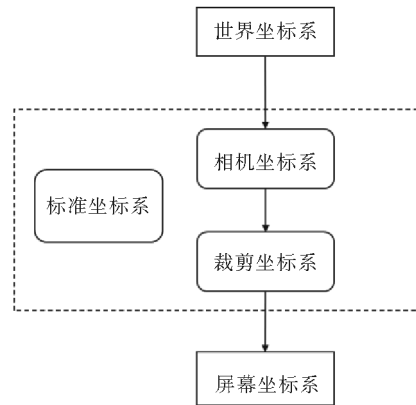


图 7 模型坐标转换流程

2.3 射线法拾取

在交互事件中对模型构件的拾取是通过射线法获取的。射线法是利用一根有箭头的红色射线去和三维场景中的模型构件相交,相交的模型对象就是被拾取到的模型。

红色射线由透视相机发出,穿过蓝色屏幕上的位置 P_0 ,继续延伸,交视棱台近截面于 P_1 ,交视棱台远截面于 P_3 ,得到一条线段 P_1P_3 ,线段 P_1P_3 就是眼睛能看到的射线在透视空间中的部分,并且这条线段相交的模型,就是鼠标点击选中的空间模型;这个交点坐标就是鼠标点击选中模型的交点空间三维坐标。射线法原理如图 8 所示。

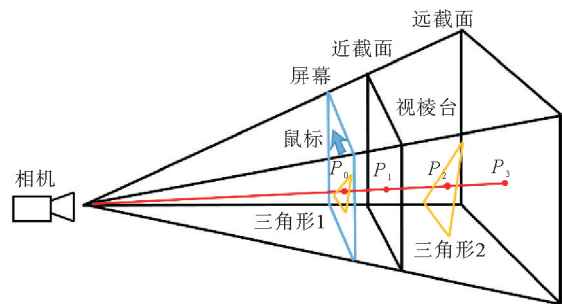


图 8 射线法原理

2.4 属性获取表格

属性信息是通过构件唯一标识符确定的,通过元素 ElementId 获取,并在重构时与模型相关联,然后采用射线法,进行鼠标拾取,当射线与第一个构件相交时触发事件使其高亮并展示构件属性信息。通过这种方法可以实现在浏览器中实时查看具体构件详细属性的功能,属性获取栏如图 9 所示。

高亮显示是通过射线法拾取技术,拾取到三维世界中的某个对象,当使用鼠标点击某个结构时,被选中的结构、颜色会发生改变,同时会弹出对应属性,效果如图 10 所示,其中墙装饰高亮显示为粉色。

属性	属性值
name	体量幕墙大样-门头
UniqueId	436b243b-7338-4
ElementId	2026587
类别	体量
明细表标高	F1
工作平面	标高: F1
主体中的偏移	-300.000 mm
创建的阶段	新构造
拆除的阶段	无
总楼层面积	0 m ²
总面积	0 m ²
总体积	0.00 m ³

(a) Revit属性栏

(b) Web端属性栏

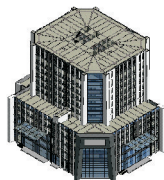
图 9 属性获取栏

3 模型显示与交互实验

本文以某一高层建筑塔楼作为渲染的主要对象,以某办公楼、某教学楼作为对比对象,模型信息见表1。采用的机器型号为 Intel(R) Core(TM) i9-12900H@ 2.70GHz、16 GB 内存以及 64 位 Windows11 操作系统的笔记本电脑,基于 Google Chrome 浏览器实现。

3.1 模型外观

为了更好地还原在 Revit 中的模型外观,在二次开发中导出 GLTF 格式的同时导出模型的材质贴图。在模型重构过程中配上材质贴图后,模型的 Revit 效果图和网页端渲染图的对比如图 11 所示。



(a) Revit模型

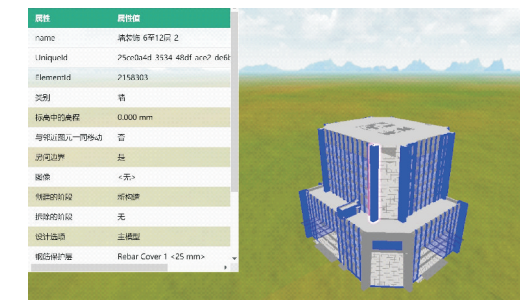
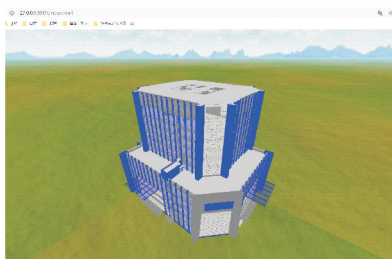


图 10 高亮拾取

表 1 模型信息

模型名称	构件数量/个	模型大小/MB
办公楼	169	7.16
教学楼	8 233	27.70
塔楼	16 023	70.40



(b) 浏览器渲染模型

图 11 模型外观对比

3.2 浏览器兼容性测试与交互事件

在操作过程中,先添加 controls 轨道球控件,检测键盘和鼠标的操作事件并将其定义于浏览器之中。执行操作时会接收到反馈的对象,给该对象添加一个监听事件来监听鼠标、键盘的变化。执行 controls 语句时,相机参数随着用户操作鼠标键盘变化,最终呈现出变化后的渲染结果,鼠标键盘操作事件见表 2。引用代码如下:

```
const controls=new OrbitControls(camera, canvas)
```

事件交互功能在 Chrome 浏览器进行测试,效果如图 12 所示。

浏览器兼容性对模型渲染效果和交互功能有重要影响。因此,对常用的多个浏览器进行兼容性测试,效果如图 13 所示。

在测试的四种浏览器中交互结果见表 3,较好地实现了模型的交互功能,浏览器兼容性测试结果良好。

3.3 Draco 算法压缩效率

Draco 算法压缩模型的效率直接影响到 Web 端轻量化显示的效果。建筑模型在通过插件导出之后,文件体量会发生不同程度的变化。经过 Draco 压缩算法轻量化处理之后,文件体量会大大减小,压缩优化率会稳定在 50%以上,并且随着模型体量的增加,压缩优化率降低,具体数据见表 4。

表 2 鼠标键盘交互事件

交互事件	平移	缩放	旋转
操作	Shift+左键拖动 /右键+拖动	鼠标滚轮	Shift+右键拖动 /左键+拖动

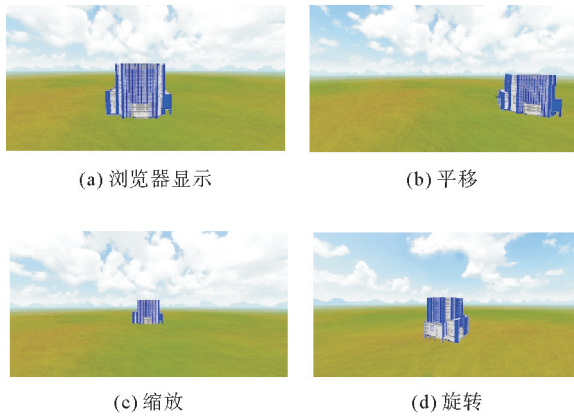


图 12 Chrome 浏览器模型交互

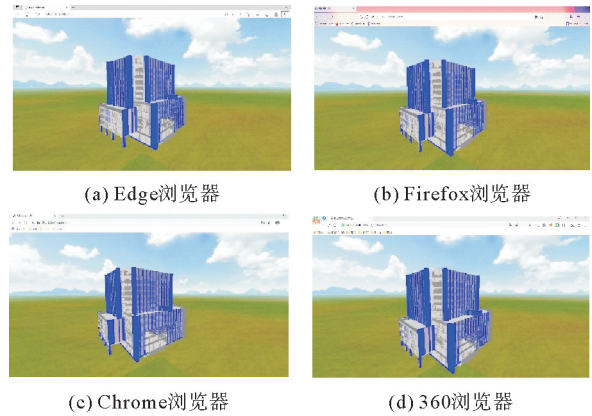


图 13 不同浏览器展示效果

表 3 浏览器兼容性操作事件

浏览器类型	平移	缩放	旋转	高亮
Edge	是	是	是	是
Chrome	是	是	是	是
Firefox	是	是	是	是
360	是	是	是	是

表 4 BIM 整体模型参数化信息

模型	文件类型 (压缩前)	文件类型 (压缩后)	转换前/ MB	转换后/ MB	压缩后/ MB	压缩优化率/ %
办公楼	gltf	gltf	7.16	7.31	1.10	85.0
教学楼	gltf	gltf	27.70	57.50	15.80	72.5
塔楼	gltf	gltf	70.40	64.50	27.80	56.9

4 大屏显示

数据大屏展示系统具有直观化、关联化、艺术化和交互性的特点,可以通过合理的可视化使用户感知大量数据,用可视化方式呈现出来,会更直观体现数据信息^[15]。本文通过使用 iframe 标签,将 Web 端模型轻量化显示的 Index.html 文件内嵌于通过 CSS3 技术和 HTML5 创建的数据大屏系统中,更美观、更细致地展示 BIM 模型。数据大屏系统架构如图 14 所示。

1) 技术层:主要包括在创建数据大屏系统时需要采用的架构技术,如 CSS3, HTML5 以及 Three.js 框架等。

2) 展示层:大屏展示平台,页面包括左侧模型展示和右侧构件、图表等数据展示。

3) 应用层:模型渲染交互功能、登录、模型全景展示。

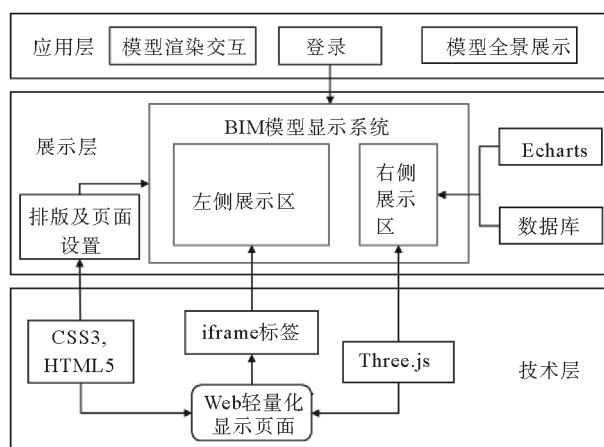


图 14 数据大屏系统架构

为大屏系统设置的登录界面如图 15 所示。

BIM 模型大屏展示系统页面布局如图 16 所示。从登陆界面进入到 BIM 模型展示系统后,左侧为主要信息展示区域,展示 BIM 模型,右侧为次要信息展示区域。为避免长时间直视大屏产生视觉疲劳,让画面保持平稳,主题色、背景色选用低饱和度、低亮度的暗色系中的蓝色、深蓝色为主题颜色,比较符合 BIM 技术的科技感和可视化特点。

为获得更好的操作体验,优先考虑使用 Google Chrome 浏览器浏览页面。大屏展示页面地址为 <http://127.0.0.1:5501/bigscreen.html>。本文设计的数据大屏展示系统,用户可以选择模型和修改系统版式以提供直观、舒适的数据大屏交互体验。塔楼 Revit 模型上传到数据大屏系统的三维视图如图 17 所示。

Revit 模型在 Web 端的重构与交互功能在大屏系统保持正常运行,整体效果如图 18 所示。



图 15 登陆界面

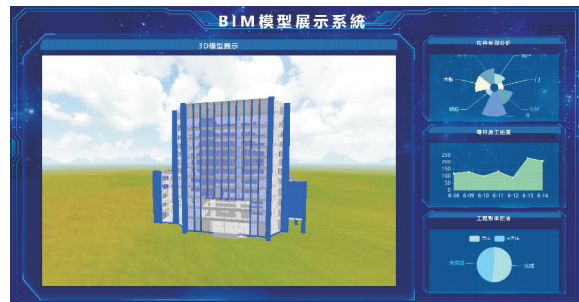


图 16 页面布局效果

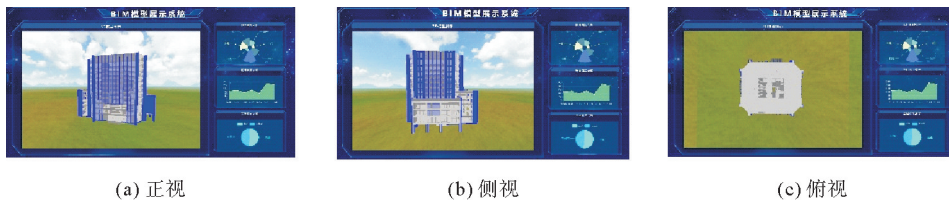


图 17 塔楼模型三维视图



图 18 大屏展示系统中的模型交互

5 结论

1) 通过 Revit 二次开发,实现了模型从 RVT 格式到 GLTF 格式的转换;采用 Draco 压缩算法,实现了模型文件的轻量化处理;选用 Three.js 引擎,实现了 BIM 模型在 Web 端的显示以及模型交互功能。

2) 在 Web 端实现模型显示与交互功能的基础上,增加了数据大屏功能,使模型展示更系统、更美观、

更详细,达到了较好的轻量化显示效果。

参考文献(References):

- [1] SHOCKLEY J, BRUCKER B A, CASE M P. Building information modeling (BIM): A road map for implementation to support MIL-CON transformation and civil works projects within the US army corps of engineers[M]. Washington: Engineering Research and Development Center, 2019.
- [2] GB/T 51212—2016, 建筑信息模型应用统一标准[S].
GB/T 51212—2016, Unified standard for building information modeling[S].
- [3] GB/T 51269—2017, 建筑信息模型分类和编码标准[S].
GB/T 51269—2017, Standard for classification and coding of building information model[S].
- [4] GB/T 51447—2021, 建筑信息模型存储标准[S].
GB/T51447—2021, Standard for storage of building information model[S].
- [5] 俞铭涛. 基于 ASP.NET 的大学图书馆网站的设计与实现[D]. 杭州: 浙江工业大学, 2020.
YU Mingtao. Design and implementation of university library website based on ASP.NET[D]. Hangzhou: Zhejiang University of Technology, 2020.
- [6] AHYUN L, INSUNG J. Implementation of an open platform for 3D spatial information based on WebGL[J]. ETRI Journal, 2019, 41(3): 277-288.
- [7] 贺佳佳. Revit 模型的轻量化研究[D]. 西安: 西安建筑科技大学, 2019.
HE Jiajia. Lightweight research of Revit model[D]. Xi'an: Xi'an University of Architecture and Technology, 2019.
- [8] 余宇深. 基于 BIM 技术的桥梁参数化建模及轻量化研究[D]. 北京: 北京交通大学, 2021.
SHE Yushen. Research on parametric modeling and lightweight of bridge based on BIM technology[D]. Beijing: Beijing Jiaotong University, 2021.
- [9] 阎超. 基于 Revit 的建筑模型信息二次开发应用[D]. 太原: 中北大学, 2020.
YAN Chao. Revit-based development and application of building model information[D]. Taiyuan: North University of China, 2020.
- [10] SHIROLE A M, CHEN S S, GAO Q, et al. BrIM based bridge operations and management[M]. Philadelphia: Bridge Maintenance, Safety, Management and Life-Cycle Optimization, 2010.
- [11] 王占军, 王鹏, 吕婧, 等. 浅析基于 3D Tiles 的 BIM 模型网络共享方案[J]. 科学技术创新, 2020(8):91-93.
WANG Zhanjun, WANG Peng, LÜ Jing, et al. Analysis of BIM model network sharing scheme based on 3D Tiles[J]. Science and Technology Innovation, 2020(8):91-93.
- [12] 吕婧, 金浩然, 谭军, 等. glTF 在 BIM 模型轻量化中的应用[J]. 科技创新与应用, 2020(6):174-176.
LÜ Jing, JIN Haoran, TAN Jun, et al. Application of glTF in BIM model lightweight[J]. Technology Innovation and Application, 2020(6):174-176.
- [13] JAREK R. Edgebreaker: Connectivity compression for triangle meshes[J]. IEEE Transactions on Visualization and Computer Graphics, 1999, 5(1):47-61.
- [14] 谯慧琼, 刘力溯. 基于 Three.js 的在线三维模型展示方案[J]. 电脑编程技巧与维护, 2021(8):139-141.
QIAO Huiqiong, LIU Lisu. Online 3D model display scheme based on Three.js[J]. Computer Programming Skills and Maintenance, 2021(8):139-141.
- [15] 徐彤. 运用数据分析和可视化技术的数据大屏交互体验系统研究[J]. 河南科技, 2019(7):25-27.
XU Tong. Data large screen interactive experience system using data analysis and visualization technology[J]. Henan Science and Technology, 2019(7):25-27.

(责任编辑 姜锡方; 英文校审 程文华)