云边页协同的 WebBIM 大场景多粒度兴趣加载调度算法

李柯, 张乾, 贾金原*

(同济大学软件学院 上海 201804)
(jyjia@tongji.edu.cn)

摘 要: 当今 5G 与 VR 日益兴起, 在移动网页浏览器上展示大规模建筑信息模型(BIM)场景的需求越来越多. 网页浏览器计算性能弱与网络延迟严重等问题, 使大规模 WebBIM 场景的在线可视化具有较大的挑战性. 为此, 提出云边页协同的 WebBIM 大场景多粒度兴趣加载调度算法. 首先, 将边缘计算引入 WebBIM 在线可视化中, 设计云边页三端协同的 WebBIM 多粒度在线可视化框架; 然后, 针对 BIM 构件给出兴趣度的定义, 并提出基于兴趣度的多粒度化传输调度机制; 最后, 根据视角的全局性、轻量性与兴趣度, 给出 BIM 场景的最优初始加载视点的选取方法. 借助于阿里云服务搭建实验平台, 并选取若干大型 BIM 场景进行了网上实测. 实验结果表明, 该算法能够有效地降低加载 BIM 场景的数据量, 显著地提升 WebBIM 场景的初始加载速度, 在线漫游时没有明显的延迟.

关键词: WebBIM; 边缘计算; 云边页协同; 多粒度化调度; 兴趣驱动的渐进式加载; 最优初始加载视点选取

中图法分类号: TP391.41 **DOI:** 10.3724/SP.J.1089.2021.18718

CEB-Collaboratively Multi-Granularity Interest Scheduling Algorithm for Loading Large WebBIM Scene

Li Ke, Zhang Qian, and Jia Jinyuan*

(School of Software Engineering, Tongji University, Shanghai 201804)

Abstract: With the rise of 5G and VR, the demand on visualizing large-scale BIM scene on mobile web browsers is increasing rapidly. However, due to the weakness of browser rendering and networking delay, it is still rather challenging to access huge Web3D scenes smoothly. Firstly, edge computing is introduced into WebBIM online visualization, and a cloud-edge-browser (CEB) collaboratively multi-granularity WebBIM framework is designed. Secondly, the interest degree is defined for each of BIM components and the multi-granularity WebBIM transmission scheduling is proposed based on interest degree. Then, the optimal viewpoint selection for initial loading is presented in terms of globality of viewing scope, light weightness and interest degree. Finally, the experimental platform is deployed using Alibaba Cloud service, and some large-scale BIM scenes are used for practical transmission testing. Experimental results show that the proposed algorithm can effectively reduce the data amount of loading WebBIM scenes. The initial loading speed is also improved significantly, and there is no obvious delay to occur when users move huge WebBIM scenes through freely.

Key words: WebBIM; edge computing; cloud-edge-browser collaboration; multi-granularity scheduling; interest

收稿日期: 2020-10-15; 修回日期: 2020-11-30. **基金项目:** 国家自然科学基金面上项目(6207071897); 国家自然科学基金区域联合基金重点项目(U19A2063). **李柯**(1994—), 男, 硕士研究生, CCF 学生会员, 主要研究方向为 WebBIM、边缘计算、虚拟现实、计算机图形学; **张乾**(1985—), 男, 博士研究生, 主要研究方向为 Web3D、虚拟仿真实验、渲染引擎、轻量级几何造型; **贾金原**(1963—), 男, 博士, 教授, 博士生导师, CCF 高级会员, 论文通讯作者, 主要研究方向为 Web Graphics、分布式虚拟世界、轻量级 3D 建模、数字娱乐、游戏引擎.

driven downloading; optimal viewpoint selection for initial loading

当今社会已经步入大数据时代,各行各业的数据量均呈快速增长趋势,建筑行业也是如此,越来越朝着数字化、信息化、智能化方向发展.建筑信息模型(building information modeling, BIM)是基于三维数字设计所构建的可视化数字建筑模型,其作为建筑信息化与计算机3D可视化交叉新兴技术,已经成为建筑行业信息化发展和计算机3D可视化研究的重要方向.随着互联网+的日益成熟, BIM与Web3D技术相结合,在网页浏览器上实时展示BIM虚拟场景的要求越来越强烈,而5G时代的到来,使得WebBIM受到日益广泛的关注,进一步促进了WebBIM的应用.值得指出的是,在2020年新冠疫情期间,国内大部分行业均处于停滞状态,WebBIM研究极大地促进了BIM的发展,使WebBIM真正成为刚需.

随着BIM场景中建筑物模型的数据量越来越大,在性能低、带宽有限的移动网页端进行快速、高效且流畅的可视化常常存在着诸多问题,这些问题严重影响着用户的使用体验,极大地限制了WebBIM的发展.当前,WebBIM在线可视化面临如下问题.

(1) 粒度问题.粒度是指所有位于可视区域的构件单元集合.对于一个大规模场景,用户在任意时刻仅能观察到场景的一部分区域.如在一个建筑场景中,当用户视点在某室内区域时仅能观察到该室内部分的模型,每当用户视点移动时都只需下载位于内部的BIM场景,可大大减少每次需要下载的数据量.而原始BIM场景粒度较粗,传输所需时间长,计算和渲染效率也较低,导致网页端场景加载慢、易卡顿,严重影响用户的使用体验.多粒度化传输调度算法可以降低场景传输和计算处理的数据粒度,提升WebBIM可视化效率.

(2) 架构问题.在WebBIM可视化架构中,一般采取纯浏览器端架构、纯云端架构、云端与浏览器端协作架构等,但是它们各自存在着一定问题.例如,纯浏览器端架构将所有的计算、处理和渲染任务放置在浏览器端进行,由于浏览器本身性能有限,大规模WebBIM可视化效果较差;而纯云端架构则将所有的计算和渲染全部放置在云端进行,浏览器端则仅显示云端传输的视频帧,这种架构对带宽要求极高,对于带宽较低的用户效果较差;云端与网页端协作模式同时发挥二者的作用,将部分复杂的且对硬件性能需求较高的计算和渲染

任务放到云端进行,但是云端与浏览器端之间的高传输时延制约了其进一步发展.本文提出云边页(cloud-edge-browser, CEB)协同架构模式,同时发挥云端、边缘端与浏览器端的优势,三端协调运作,有效地提升WebBIM可视化效果.

(3) 加载延迟问题.加载延迟时间长是当前WebBIM可视化面临的重要难题.通常,用户愿意等待的时间相当有限,较长的等待时间可能导致用户失去耐心,尽可能降低加载延迟时间对于提升用户体验极其重要.本文通过最优初始视点选取算法选择场景加载的最优初始视点,降低初始加载延迟时间,同时选择对于用户体验较好的初始观瞻角度,提升用户体验.

1 相关工作

1.1 边缘计算与3D可视化

传统云计算采取中心化部署,本地设备数据传输至云端,云端统一进行数据的处理、计算和存储,再将结果返回本地设备,这种处理方式对于时延低、数据量大的处理任务存在明显不足.与云计算的中心化思路不同,边缘计算^[1]采取分布式结构,通过在本地设备端附近部署一定数量的计算节点,本地设备数据直接传输至边缘节点进行计算和处理,使服务器的处理响应速度和可靠性都高于云计算.2014年,欧洲电信标准化协会(European telecommunications standards institute, ETSI)^[2-3]提出对移动边缘计算术语的标准化.利用移动边缘计算,可将部分移动计算任务迁移到附近的移动边缘服务器^[4]进行.2018年,ETSI^[5]将mobile edge computing正式更名为multi-access edge computing,相较于之前,除了强调无线网络的接入方式外,还需要考虑其他的数据接入,完善了边缘计算的模式.

由于数据不需要传输到云端,只需在边缘侧处理,因此对于Web3D可视化的实时计算和处理也更加高效.2017年,Hou等^[6]实现了轻量级的虚拟现实(virtual reality, VR)眼镜与远程执行渲染的边缘/云设备之间的无线连接,并从延迟性等方面对多种方案进行比较;Chakareski^[7]在5G基站上进行缓存和计算,实现了一个5G无线360°VR/AR视频展示,探讨了该框架的缓存机制、传输和边缘计算方式.2018年,Erol-Kantarci等^[8]研究了5G网络中基于移动边缘计算的VR/AR缓存与计算机制.

2019 年, Xu 等^[9]提出边缘计算中依据无线 VR 内容传输进行最优带宽分配的方法, 推进了 VR 与边缘计算的研究.

1.2 Web3D 在线可视化模式

虚拟 3D 场景数据量巨大、结构复杂, 其可视化受移动端浏览器性能与网络带宽速率等方面影响. 当前, Web3D 在线可视化仍处于初步探索阶段.

1.2.1 基于纯网页端可视化模式

纯网页端可视化模式将所有的计算、渲染任务放到浏览器端, 浏览器端完成所有的计算、调度、加载和渲染, 其对浏览器端的硬件性能要求较高, 特别是对于移动端的设备. 王珩玮等^[10]使用 IFC Optimizer 工具对 BIM 文件进行语义轻量化处理, 并使用 Three.js 框架在浏览器上将 BIM 模型可视化; 但是, IFC Optimizer 轻量化能力有限, 当在网页上加载和渲染大规模 BIM 场景时, 会产生高额的延迟. 张东培等^[11]针对 BIM 虚拟场景建立适用于 Web3D 现实的轻量级场景结构, 并提出渐进式兴趣区域(progressive area of interest, PAOI)网格化场景管理策略, 对室内/外场景实现按需加载, 降低了内存开销. 文献[12-14]进行进一步拓展, 使用基于语义与几何分析的轻量化方法对 BIM 模型进行去重处理, 使用基于空间分析方法对区域子空间进行分割并建立索引, 依据漫游视点位置加载和渲染对应区域模型, 其提供一种有效的针对 BIM 场景的在线可视化思路; 但是其要求场景要有较强的封闭性, 对于开阔场景或者无明显外体的场景(如管廊)实用性较低.

1.2.2 纯云端可视化模式

纯云端可视化模式将所有的计算和渲染等任务放在云端, 并将渲染的结果发送到 Web 端进行显示, 其渲染质量优秀且不受 Web 端硬件影响. Chuang 等^[15]提出 Cloud-BIM 方法, 通过运用强大的云计算功能对 BIM 空间进行可视化计算; 然后将渲染计算结果传输到 Web 客户端显示, 用户可以随时通过 Web 浏览器对 BIM 模型进行交互操作. Das 等^[16]提出基于云的 BIM 服务器框架, 能够对建筑模型进行动态合并与拆分. Chen 等^[17]提出更系统全面的 Cloud-BIM 解决方案. 刘强^[18]开发了基于云计算的 BIM 数据集成和管理平台, 支持各参与方按照自身的数据需求分布式地建立、存储、管理和应用 BIM 数据, 实现数据的传递和共享; 但是这种模式对于云端的硬件性能以及带宽的要求都非常高, 其完全忽视了 Web 端设备本身的计算和渲染能力, 造成了极大的资源浪费, 并且由于

完全由云端提供实时渲染, 不可避免地会带来严重的渲染和传输延迟.

1.2.3 云页协同可视化模式

云页协同可视化模式同时发挥云端和浏览器端的性能, 将部分复杂的且对硬件性能需求较高的计算和渲染任务放到云端进行. 这种方式协调了上述 2 种模式的矛盾, 在降低浏览器端的硬件门槛的同时也减轻了云端的负荷, 当前这种方式在图形渲染方面已经有较多的研究和应用. Crassin 等^[19]提出 CloudLight 系统, 使用 3 种不同策略的光照和传输方法使前端支持全局光照. Liu 等^[20]和邵威等^[21]也提出协作式云烘焙系统, 在云端进行光影贴图的烘焙, 并将结果与 Web 端场景进行融合.

表 1 所示为 3 种渲染模式的性能对比. 可以看出, 云端、浏览器端协同渲染模式相对其他 2 种模式存在一定优势, 但是对时延性要求依旧较高, 当存在大量实时数据的传输和处理时, 云端、浏览器端二者间的协同效果不太理想.

表 1 3 种渲染模式性能对比

指标	渲染模式		
	纯网页端	纯云端	云页协同
云端算力需求	低	高	中
浏览器端算力需求	高	低	中
支持场景规模	低	高	高
网络带宽速率要求	低	高	高
用户等待时间	长	短	中
漫游延迟	低	高	中

2 WebBIM 系统架构设计

2.1 设计原理

本文系统包含 cloud, edge 和 browser 共 3 层, 它们在处理性能、访问频度和存储能力方面均存在较大差异.

(1) cloud. 处于云中心位置, 计算和处理能力相对最强, 数据访问频度相对较低, 存储能力较强; 但是其与用户物理位置较远, 延迟与带宽成本较高, 适合进行重量级计算, 处理访问频度较低、粒度较粗的数据, 可以进行持久化存储. 本文主要进行 WebBIM 场景的轻量化预处理, 该处理对硬件性能要求较高, 处理时间长, 数据只需在初始时传输即可.

(2) edge. 相对于 cloud, 其与用户物理位置相对较近, 延迟与带宽成本较低; 但其计算和处理能

力相对较低,数据访问频度相对较高,存储能力较弱,适合进行中量级计算,处理中粒度数据,数据的访问频度相对较高,可以长期存储.本文主要进行场景的多粒度化传输调度处理和初始加载优化.

(3) browser. 浏览器处理性能相对最低,处理的数据粒度相对较细,数据的访问频度相对较高,数据仅能临时存储,适合进行轻量级计算、处理访问频度较高的轻量数据.本文主要进行场景在网页上的加载、渲染和用户的交互控制等.

2.2 WebBIM 系统架构

本文的 WebBIM 系统架构如图 1 所示. cloud 端与 edge 端的数据通信相对较少, cloud 端进行 WebBIM 场景的轻量化预处理,将处理得到的场景模型数据发送至 edge 端; edge 端则将部分边缘计算的结果返回至 cloud 端进行持久化存储; edge 端与 browser 端的通信相对较为频繁, browser 端向 edge 端发送其在线计算结果或可视化请求, edge 端进行场景的多粒度化传输调度处理,将场景的细粒度数据发送至 browser 端进行可视化.

图1 WebBIM 系统架构

与传统的云服务器-浏览器 (cloud-browser, CB) 模式相比, CEB 协同模式引入了边缘端, 将 CB 模式中服务器端的调度与计算放置于边缘端进行, 大幅降低了计算和数据传输延迟. 同时, 将强度高、处理时间长、访问频度低的任务放在 cloud 端处理, 发挥了 cloud 端本身的计算性能优势, 而

browser 端负责接收传输的数据、场景可视化等相对轻量的任务, 三端协同配合, 实现了系统的高效运作.

3 技术路线

基于上述的系统架构, 本文的技术路线如图 2 所示, 具体如下.

(1) cloud. 对原始 BIM 场景进行轻量化预处理, 以降低场景的数据量. 基于参考文献[12-13], 本文处理过程如下: 通过 Revit 建模工具得到的原始 BIM 场景, 首先对其进行语义和几何重复性检测, 去除冗余数据; 然后利用稀疏体素化对场景进行索引, 并利用体素的遮挡关系将场景分为室内和室外; 再利用体素块中构件的邻接关系对场景进行分块处理; 最后对得到的模型进行数据压缩.

(2) edge. 包含多粒度调度处理和初始加载优化. 多粒度调度处理为实时的处理过程, 通过同步用户的视点信息, 计算并传输用户真正可视的模型集合, 大幅减少了场景所需实时加载的数据量, 多粒度调度处理过程包含场景视点同步、计算可视增量列表、兴趣度计算、自适应打包和按优先级传输. 而最优初始视点选取是针对初始加载进行的优化, 通过选取场景的最优初始视点, 减少场景初始可视化所需加载数据量的同时提供良好的初始观察视点, 以改善场景初始加载的用户体验. 最优

图2 本文技术路线

初始视点选取过程包含计算场景包围球、确定视点待选区域、搜索最优视点以及确定模型加载序列。

(3) browser. 负责场景在网页上的轻量级在线可视化, 包含实时视点更新、按序加载、实例化渲染和用户交互控制。

4 基于兴趣度的多粒度化传输调度机制

对于大规模场景, 在漫游或显示中用户任何时刻都仅能观察到场景中的一部分构件, 因此, 加载和渲染全部场景会带来严重的资源浪费。依据用户的视点信息, 实时计算出用户的可视构件集合, 仅加载和渲染真正对用户视觉有影响的构件模型, 可以提升可视化效率。

4.1 兴趣度概念及其计算

由于场景中的不同构件模型对于用户的视觉存在着明显的差异, 如体积大、深度图中占用像素多的构件在屏幕中占据较大空间, 因此在传输过程中应对这些构件模型赋予较高的优先级。为了更好地量化各个模型在传输过程中的优先次序, 本文引入兴趣度的概念。

定义 1. 填充度(fill degree, FD). FD 为模型所占的包围盒体积与其文件大小的比值, 定义为

$$FD(m_i) = \frac{V_b(m_i)}{S(m_i)}.$$

其中, V_b 为模型的包围盒体积; S 为模型的文件大小. FD 反映模型单位传输消耗下在 3D 空间中所占的空间大小, V_b 越大、文件越小, 对用户视觉影响越大, 模型的传输优先级也就更高。

定义 2. 重用度(reuse degree, RD). RD 是模型的实例构件数量与其文件大小的比值, 定义为

$$RD(m_i) = \frac{N_i(m_i)}{S(m_i)}.$$

其中, N_i 为模型的实例数量; S 为模型的文件大小. RD 反映模型在单位传输消耗下场景中重复实例的个数, 在场景的预处理阶段, 通过对模型查重去冗, 已经生成了模型的实例化信息(文件相对极小), 模型及其实例化信息在传输中会一起传输至 Web 端; 在渲染阶段, 由于使用实例化渲染技术, 一次渲染即可渲染出所有实例化模型, 大幅提升模型渲染效率. 因此, 重用实例数量越大、加载和渲染效率越高, RD 就越大, 传输优先级越高。

定义 3. 关注度(attention degree, AD). AD 反映模型在单位传输消耗下用户对其可能的关注程度高低, 计算公式为

$$AD(m_i) = \frac{T(m_i) + N(m_i)}{S(m_i)}.$$

其中, T 为模型的类型系数; N 为模型的交互次数统计值; S 为模型的文件大小. AD 考虑用户对场景中模型的关注程度, 其中, T 是根据模型的构件类型分配的优先级, 由于不同类型的模型对用户的浏览存在着不同程度的影响力, 如建筑物的门、窗等就对用户的有意观察影响较大, 而建筑物内部的管道、螺丝、柱子等相对影响较小. 考虑到用户对场景中模型交互与浏览的相似性, 通过统计历史用户对各个模型的交互次数, 进一步细化各个具体模型的 AD. AD 越高, 传输优先级越高, 用户体验越好。

定义 4. 兴趣度(interest degree, ITD). ITD 是 FD, RD 和 AD 的加权结果, 综合反映该模型对用户视觉影响的贡献大小及其文件传输的代价, 定义为

$$ITD(m_i) = \alpha \times FD(m_i) + \beta \times RD(m_i) + \gamma \times AD(m_i).$$

其中, $\alpha + \beta + \gamma = 1$, α , β 和 γ 为各自的权值, 这些权值需要在实验中多次重复调整才能确定. 本文使用控制变量法, 权值以 0.1 为递加/减单位. 由于 ITD 主要用于量化模型在传输过程中的优先级, 因此选取的依据是模型传输顺序的合理性, 即在一定的时间内传输更多对用户视觉影响更大的模型, 可以通过计算固定时间内渲染完成后所有模型占用的屏幕像素数量进行量化。

为了更好地阐述 ITD 在模型传输中的作用, 图 3 给出了一套模型分别在 2 种加载序列下(ITD 从高到低和 ITD 从低到高)的完整加载过程对比. 可以看出, 当模型加载序列为 ITD 从低到高时, 开始加载的模型对用户视觉影响很小且加载速度慢, 直到模型全部加载完毕用户才能观察到场景的全貌; 而当模型加载序列为 ITD 从高到低时, 当场景加载一半左右时就可以观察到场景的全貌, 视觉影响较小的细节随后逐步加载, 用户体验明显更优。

图 3 模型在 2 种序列下加载过程对比

4.2 多粒度化传输调度机制流程

本文使用基于ITD的多粒度化传输调度机制,步骤如下。

Step1. 基于文献[12-13], cloud端对原始BIM场景进行轻量化预处理(包括场景几何和语义去重、数据压缩等),去除场景的冗余数据,降低传输的数据量。

Step2. edge端服务器获取cloud端场景数据并加载相应场景,准备颜色与模型对照表,该对照表中存储了渲染模型时所使用的颜色信息,用于模型的辨识。

Step3. 用户打开浏览器,Web端建立与edge端的数据连接。

Step4. edge端实时获取用户的视点信息(相机位置、朝向等),用于在edge端同步用户的视点。

Step5. 按照用户视点渲染当前场景,依据颜色与模型对照表提取出当前视点下的真实可见模型列表,并与上一视点的可视模型列表取差值得到可视增量模型列表,该可视增量模型列表即为当前视点下所需传输的最小模型集合。

Step6. 计算可视增量模型列表中所有模型的ITD,然后依据用户的带宽大小进行自适应打包。ITD反映了模型对用户视觉的影响贡献大小,决定该模型在场景传输中的次序。

Step7. Web端加载edge端传输的模型数据包,解析并渲染出相应场景。

通过上述步骤,实现在一次视点更新中场景的完整加载过程。

4.3 自适应打包和传输策略

每次网络请求和传输都会消耗一定的网络资源,由于每个构件文件相对较小,整个场景构件数量庞大,如果每次网络请求和传输仅发送单个构件文件,整个场景的文件传输会带来大量的额外网络资源消耗。为了最大限度地降低用户的等待时间,必须在低带宽网络环境下,以最短的时间传输最多的、最合理的模型到Web浏览器进行优先渲染,获得传输效益最大化。因此,本文采用自适应打包机制,步骤如下。

Step1. 侦测用户的网络可用带宽,即在建立服务器与Web浏览器端的连接之后发送一定大小的文件包,根据传输时间计算出带宽大小 W_b 。

Step2. 设定包的最大传输时间 T (本文中 $T=1s$),将文件包的大小定为 $S_p=W_b \times T$ 。

Step3. 将之前得到的可视增量列表中的构件按照其兴趣度的高低排序,依次进行打包合并。

Step4. 将得到的文件包按照次序进行传输。

5 最优初始视点选取

由于WebBIM场景的每次初始加载视点位置

总是一致的,并且初始加载效果基本直接决定用户的使用体验,因此需要对场景的初始加载位置进行选取。最优初始视点选取通过选择合适的视点位置和观察角度,使初始加载的数据量在尽可能少的同时加载尽可能多的模型,用户在该视点下可以观察到尽可能多的场景细节。

5.1 视点优先级计算

5.1.1 可视规模度

可视规模度为场景的全部模型文件数据量与该视点下所有可视模型的文件数据量的比值,计算公式为

$$S_v = \left(\sum_{i=1}^{N_v} S_i + \sum_{i=1}^{N-N_v} S'_i \right) / \sum_{i=1}^{N_v} S_i$$

其中, S_i 为可视模型文件数据量; S'_i 为不可视模型文件数据量; N 为场景的总模型数量; N_v 为该视点下可视模型的数量。

S_v 反映了当前视点下可视模型文件数据量的比例, S_i 越小, S_v 越大,该视点所需加载的数据量越小,视点的优先级越高。

5.1.2 场景重用度

场景重用度考虑的是位于该视点下可视模型非重用(实例化)与重用情况下文件大小的比例,计算公式为

$$R_v = \sum_{i=1}^{N_v} U_i / \left(\sum_{i=1}^m U'_i + \sum_{i=1}^{N_v-m} U_i \right)$$

其中, U_i 为各个模型的原始文件大小; U'_i 为重用模型的实例化文件大小; m 为该视点下重用的可视模型数量。

实例化模型在预处理阶段经过去重处理,其实例化文件相对极小,且由于使用实例化渲染技术传输和渲染成本较低,因此当前视点下的重用模型占比越高,该视点下的模型加载渲染成本越低,视点优先级越高。

5.1.3 场景填充度

场景填充度考虑的是位于该视点下所有可视模型占用屏幕空间的大小,其影响用户对场景的观瞻细节,场景填充度越大,用户的可观察细节就越多,屏幕空间利用越充分。场景填充度计算公式为

$$F_v = (V + V') / V$$

其中, V' 为模型在屏幕上占据的空间; V 为屏幕总空间。

5.1.4 视点优先级

视点优先级的计算公式为

$$P = \alpha \times S_v + \beta \times R_v + \gamma \times F_v.$$

其中, $\alpha + \beta + \gamma = 1$.

5.2 最优初始视点选取

本文中, 最优初始视点从场景包围球的球面上选取, 过程如图 4 所示, 步骤如下:

Step1. 计算场景的轴对齐包围盒(axis-aligned bounding box, AABB), 并以包围盒中心为球心计算出包围盒的外接球.

Step2. 为了减少不必要的计算时间, 本文取上半球上距离 xoz 平面 $15^\circ \sim 60^\circ$ 且距离 yoZ 平面(z 轴方向为场景正面朝向, 具体参见图 4) $-60^\circ \sim 60^\circ$ 的曲面作为场景的最优初始视点待选区域.

Step3. 使用最优初始视点搜索算法计算出场景的最优初始视点.

Step4. 依据计算得出的最优初始视点, 计算该视点下的可视模型列表, 然后依据模型的兴趣度对列表中的模型从高到低进行排序, 最终得到场景的初始加载模型序列.

图 4 最优初始视点选取

本文的最优初始视点搜索算法步骤如下.

输入. 算法搜索开始视点.

输出. 场景加载最优初始视点.

Step1. 输入算法搜索开始视点.

Step2. 将该视点置为当前视点, 计算当前视点优先级.

Step3. 如果当前视点的左方邻点未搜索:

Step3.1. 计算左方邻点的优先级;

Step3.2. 如果左方邻点的优先级 > 当前视点优先级, 将左方邻点置为当前视点, 转 Step3.

Step4. 如果当前视点的上方邻点未搜索:

Step4.1. 计算上方邻点的优先级;

Step4.2. 如果上方邻点的优先级 > 当前视点优先级, 将上方邻点置为当前视点, 转 Step3.

Step5. 如果当前视点的右方邻点未搜索:

Step5.1. 计算右方邻点的优先级;

Step5.2. 如果右方邻点的优先级 > 当前视点优先级, 将右方邻点置为当前视点, 转 Step3.

Step6. 如果当前视点的下方邻点未搜索:

Step6.1. 计算下方邻点的优先级;

Step6.2. 如果下方邻点的优先级 > 当前视点优先级, 将下方邻点置为当前视点, 转 Step3.

Step7. 当前视点即为最优初始视点, 输出当前视点.

6 实验及结果分析

为了验证本文方法的有效性, 本文依托阿里云服务搭建边缘计算平台, 分别在全国各地分散部署了若干边缘服务器, 测试地点以上海为例, 并以云服务器作为对比. 依据 WebBIM 可视化的性能, 评估测试指标为网络时延、初始加载时间、帧率、漫游加载延迟等.

6.1 测试场景

本文使用多个测试场景进行测试, 具体如下.

(1) 住宅楼 BIM. 数据量为 120MB, 面片数量为 198 万, 其场景效果如图 5a 所示.

(2) 地下车库 BIM. 数据量为 230MB, 面片数量为 310 万左右, 其场景效果如图 5b 所示.

(3) 地铁站 BIM. 数据量为 920MB, 面片数量为 1200 万左右, 其场景效果如图 5c 所示.

(4) 双子塔 BIM. 数据量为 1.61GB, 面片数量为 2000 万左右, 其场景效果如图 5d 所示.

(5) 大型医院 BIM. 数据量为 2.2GB, 面片数量为 3300 万左右, 其场景效果如图 5e 所示.

图 5 本文测试场景

6.2 测试环境

(1) 边缘(云)服务器配置. Ubuntu 16.04 操作系统, Intel Xeon Platinum 8163 CPU, 23GB 内存, NVIDIA Tesla T4 GPU.

(2) 网络带宽. 服务器带宽 40Mb/s, 客户端 4G 移动网络.

6.3 测试结果

6.3.1 网络时延

本文中,网络时延测试的是从发送一个测试文件开始到边缘端或云端接收到相应数据的时间间隔,对不同大小测试文件测量引入边缘计算前后的网络时延,测试结果如表2所示.可以看出,引入边缘计算之后,网络时延得到大幅降低.

表2 不同服务器网络平均时延测试对比 ms

服务器	测试文件大小/KB				
	30	50	100	200	400
云	31	43	113	173	308
边缘	19	23	29	35	51

6.3.2 初始加载时间

初始加载时间指从用户打开网页到浏览器显示模型完成所间隔的时间,初始加载时间越短,用户体验越好.为了更全面地测试各部分的性能,本文从传统的纯网页端加载模式(网页端直接加载渲染完整场景)开始逐步引入各个部分的组件,表3所示为各场景下不同模式初始加载时间测试结果.可以看出,引入多粒度化调度、最优视点选取、边缘计算后,均得到较为明显的性能提升,其中,在引入边缘计算后速度提升效果尤为明显.

表3 各场景下不同模式初始加载时间测试对比 s

模式	场景				
	住宅楼	地下车库	地铁站	双子塔	大型医院
纯网页端加载	19.5	21.6	30.2	45.5	51.2
引入多粒度化调度	14.3	13.1	17.6	23.1	28.3
引入最优视点选取	8.2	9.4	11.4	15.6	16.3
引入边缘计算	5.3	5.1	6.8	8.4	9.1

图6所示为在各个场景下,纯网页端方案(网页端直接加载渲染完整场景)、CB协同方案(即在cloud端进行多粒度调度处理,网页端加载渲染cloud传输的细粒度模型)与本文提出的CEB协同方案3种模式的初始加载速度对比结果,以纯网页端方案的加载速度为100%.可以看出,CB协同方案的效果优于纯网页端方案,而本文提出的CEB协同方案加载速度明显快于上述二者.

图6 各场景下不同模式初始加载速度对比

6.3.3 帧率

帧率反映用户在场景漫游时的流畅度,帧率越高,用户体验越好.本文测试了多个场景下纯网页端方案、CB协同方案与本文提出的CEB协同方案场景加载及漫游中的平均帧率,测试结果如图7所示.可以看出,在各个场景中,纯网页端方案的帧率最低;而与CB协同方案相比,本文提出的CEB协同方案优于CB协同方案,平均帧率高8帧左右,具有明显优势.

图7 各场景下不同模式平均帧率对比

6.3.4 漫游加载延迟

漫游加载延迟指用户在场景中漫游时,从视点变化到场景加载完毕之间的延迟时间,漫游加载延迟越长,卡顿现象越严重,用户体验越差.由于纯网页端加载模式采取一次性全加载的方式,没有漫游加载延迟,因此本文不进行比较.本文测试中将漫游测试间隔固定为10m,即每漫游10m测试一次漫游加载延迟.图8所示为CB协同方案与本文CEB协同方案的漫游加载延迟对比结果.可以看出,CEB协同方案比CB协同方案在延迟方面具有优势,但受场景的差异影响,如在住宅楼、地下车库这些比较开阔的场景中,由于初始加载基本加载了大部分模型,因此漫游加载延迟差距较小,而对于较为封闭的场景,如地铁站、双子塔等,则效果较为明显.

图 8 各场景下漫游加载延迟对比

7 结 语

本文首先提出了基于 CEB 协同的 WebBIM 大场景多粒度兴趣加载调度算法,在继承了云页协同架构的基础上,为了降低网络请求和传输延迟,引入了边缘计算技术,提出了针对 WebBIM 场景在线可视化的 CEB 协同运作框架;其次,使用了基于兴趣度的多粒度化传输调度机制,通过在边缘端实时计算出用户的可视增量构件列表并依据用户带宽进行自适应打包传输,提升可视化效率;使用最优初始视点选取算法选择场景加载的初始视点,进一步提升场景初始加载的速度.实验结果表明,本文算法能够有效地降低 WebBIM 网络传输时延,提升场景的加载速度,优化用户体验.

本文还有很多方面需要完善,未来将从以下方面展开研究:

(1) 基于 5G 的移动边缘计算.随着 5G 技术的成熟和规模化应用,基于 5G 的移动边缘计算框架将成为可能,与本文部署的若干割裂独立的边缘域相比,MEC 服务器可以提供更加普及、高效的服务.

(2) 高级渲染、纹理.本文算法更加注重大规模 BIM 场景的加载、传输调度等,暂未考虑场景的纹理贴图以及场景加载完成后的各种高级渲染,这方面的内容是今后的工作之一.

(3) 权值的确定.本文在 ITD 和视点优先级的计算中均是各个相关部分进行加权,其中权值的确定较为烦琐,需要在实验中多次重复调整得到.今后将尝试优化合并各自的计算方式,简化权值的选取过程.

参考文献(References):

- [1] Shi W S, Cao J, Zhang Q, *et al.* Edge computing: vision and challenges[J]. IEEE Internet of Things Journal, 2016, 3(5):

637-646

- [2] ETSI. Executive briefing - mobile edge computing (MEC) initiative[OL]. [2020-10-15]. <https://portal.etsi.org/portals/0/tbpages/mec/docs/mec%20executive%20brief%20v1%2028-09-14.pdf>
- [3] ETSI. Mobile-edge computing - introductory technical white paper[OL]. [2020-10-15]. https://portal.etsi.org/Portals/0/TBpages/MEC/Docs/Mobile-edge_Computing_-_Introductory_Technical_White_Paper_V1%2028-09-14.pdf
- [4] Zhao T C, Zhou S, Guo X Y, *et al.* A cooperative scheduling scheme of local cloud and Internet cloud for delay-aware mobile cloud computing[C] //Proceedings of the IEEE Globecom Workshops. Los Alamitos: IEEE Computer Society Press, 2015: 1-6
- [5] ETSI. Multi-access edge computing (MEC)[OL]. [2020-10-15]. <https://www.etsi.org/technologies/multi-access-edge-computing>
- [6] Hou X S, Lu Y, Dey S. Wireless VR/AR with edge/cloud computing[C] //Proceedings of the 26th International Conference on Computer Communication and Networks. Los Alamitos: IEEE Computer Society Press, 2017: 1-8
- [7] Chakareski J. VR/AR immersive communication: caching, edge computing, and transmission trade-offs[C] //Proceedings of the Workshop on Virtual Reality and Augmented Reality Network. New York: ACM Press, 2017: 36-41
- [8] Erol-Kantarci M, Sukhmani S. Caching and computing at the edge for mobile augmented reality and virtual reality (AR/VR) in 5G[J]. Ad Hoc Networks, 2018: 169-177
- [9] Xu T Y, Sun Y P, Xia S J, *et al.* Optimal bandwidth allocation with edge computing for wireless VR delivery[C] //Proceedings of the IEEE/CIC International Conference on Communications in China. Los Alamitos: IEEE Computer Society Press, 2019: 903-907
- [10] Wang Hengwei, Hu Zhenzhong, Lin Jiarui. Web-oriented BIM 3D viewing and information management[J]. Journal of Information Technology in Civil Engineering and Architecture, 2013, 5(3): 1-7(in Chinese)
(王珩玮, 胡振中, 林佳瑞. 面向 Web 的 BIM 三维浏览与信息管理的[J]. 土木建筑工程信息技术, 2013, 5(3): 1-7)
- [11] Zhang Dongpei, Xie Ning, Liu Xiaojun, *et al.* Lightweight real-time rendering system for online large scale underground space[J]. Journal of Frontiers of Computer Science and Technology, 2015, 9(9): 1034-1043(in Chinese)
(张东培, 谢宁, 刘小军, 等. 网上大规模地下空间的轻量级实时绘制系统[J]. 计算机科学与探索, 2015, 9(9): 1034-1043)
- [12] Liu X J, Xie N, Tang K, *et al.* Light weighting for Web3D visualization of large-scale BIM scenes in real-time[J]. Graphical Models, 2016, 88: 40-56
- [13] Liu Xiaojun, Jia Jinyuan. Mobile web-based lightweight and real-time roaming algorithm for large-scale WebBIM scenes[J]. Science in China: Information Sciences, 2018, 48(3): 274-292(in Chinese)
(刘小军, 贾金原. 面向手机网页的大规模 WebBIM 场景轻量级实时漫游算法[J]. 中国科学: 信息科学, 2018, 48(3): 274-292)

- [14] Hu Y H, Chen Z H, Liu X J, *et al.* WebTorrent based fine-grained P2P transmission of large-scale WebVR indoor scenes[C] //Proceedings of the 22nd International Conference on 3D Web Technology. New York: Association for Computing Machinery, 2017: Article No.7
- [15] Chuang T H, Lee B C, Wu I C. Applying cloud computing technology to BIM visualization and manipulation[C] // Proceedings of the 28th International Symposium on Automation and Robotics in Construction. Seoul: ISARC, 2011: 144-149
- [16] Das M, Cheng J C P, Kumar S S. Social BIMCloud: a distributed cloud-based BIM platform for object-based lifecycle information exchange[J]. Visualization in Engineering, 2015, 3: Article No.8
- [17] Chen H M, Chang K C, Lin T H. A cloud-based system framework for performing online viewing, storage, and analysis on big data of massive BIMs[J]. Automation in Construction, 2016, 71: 34-48
- [18] Liu Qiang. Research on cloud computing based BIM data integration and management[D]. Beijing: Tsinghua University, 2017(in Chinese)
(刘强. 基于云计算的 BIM 数据集成与管理技术研究[D]. 北京: 清华大学, 2017)
- [19] Crassin C, Luebke D, Mara M, *et al.* CloudLight: a system for amortizing indirect lighting in real-time rendering[J]. Journal of Computer Graphics Techniques, 2015, 4(4): 1-27
- [20] Liu C, Ooi W T, Jia J Y, *et al.* Cloud baking: collaborative scene illumination for dynamic Web3D scenes[J]. ACM Transactions on Multimedia Computing, Communications, and Applications, 2018, 14(3s): Article No.59
- [21] Shao Wei, Liu Chang, Jia Jinyuan. Lightmap-based GI collaborative rendering system for Web3D application[J]. Journal of System Simulation, 2020, 32(4): 649-659(in Chinese)
(邵威, 刘畅, 贾金原. 基于光照贴图的 Web3D 全局光照协作式云渲染系统[J]. 系统仿真学报, 2020, 32(4): 649-659)