

基于云边页协同的 WebBIM 大场景多粒度兴趣加载调度算法

李柯, 张乾, 贾金原¹

(同济大学软件学院, 上海市 201804)

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 BIM 场景的诉求越来越浮上水面, 但因浏览器本身性能、网络延迟等因素的限制, 仍存在诸多问题。本文提出了一种基于云边页 (Cloud-Edge-Browser, CEB) 协同的 WebBIM 大场景多粒度兴趣加载调度算法, 使用了边缘计算技术, 设计了针对 WebBIM 场景在线可视化的云端、边缘端、浏览器端协同运作框架, 给出了基于兴趣度的多粒度化调度传输机制, 并选取了场景的最优初始加载视点。经实验结果表明, 该方案能够有效降低 WebBIM 网络传输时延, 减少大规模 BIM 场景传输的数据量, 提升场景的加载速度, 优化用户体验, 为 WebBIM 在线可视化以及边缘计算的发展提供了一个重要的研究思路。

关键词: WebBIM; 边缘计算; 云边页协同; 多粒度化调度; 最优初始加载视点选取

文章编号: 1004-731X (2002) xx-xxxx-x

中图分类号: TP391.4

文献标识码: A

Multi-granularity interest loading scheduling algorithm for large WebBIM scene based on CEB collaboration

Ke Li, Qian Zhang, Jinyuan Jia

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

Abstract: With the increasing emergence of 5G and VR, the demand for displaying large-scale BIM scene on mobile web browsers is increasingly urgent. However, due to the limitations of browser performance, network delay and other factors, there are still many problems. This paper proposes a multi-granularity interest loading scheduling algorithm for large WebBIM scene based on cloud, edge and browser collaboration. It uses edge computing and presents a CEB collaboration framework for online WebBIM scene visualization. The multi-granularity scheduling based on interest degree is introduced and the optimal initial loading view is selected. The experimental results show that this scheme can effectively reduce the WebBIM network transmission delay, reduce the amount of data transmitted by large-scale BIM scene, improve the loading speed of scene, optimize user experience, and provide an important research idea for the development of WebBIM online visualization and edge computing.

Keywords: WebBIM; edge computing; CEB collaboration; multi-granularity scheduling; optimal initial loading viewpoint selection

0 引言

当前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, 建筑行业越来越朝着数字化、信息化、智能化发展。建筑信息模型 (Building Information Modeling) 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以及计算机 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 虚拟场景的要求愈来愈强烈, 而 5G 时代的到来,

WebBIM 受到日益广泛的关注, 进一步促进了 WebBIM 的应用。尤其值得指出的是, 在 2020 年疫情期间, 国内大部分行业均处于停滞状态, 而 WebBIM 研究极大地促进了 BIM 的发展, 使得 WebBIM 真正成为了刚需。

然而, 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 在线可视化主要面临如下问题:

(1) **问题一: 粒度问题。**粒度指的是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

收稿日期: xxxx-xx-xx

修回日期: xxxx-xx-xx

基金项目: 国家自然科学基金重点项目(U19A2063)

作者简介: 李柯(1994 -), 男, 硕士研究生, 研究方向为虚拟现实与计算机图形学; 张乾(1985 -), 男, 博士研究生, 研究方向为虚拟现实与计算机图形学; 贾金原(1963 -), 男, 教授, 博士, 研究方向为虚拟现实与计算机图形学。

刻仅能观察到场景的一部分区域,比如在一个建筑场景中,当用户视点在某室内区域时,其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。通过多粒度化调度传输算法,降低了场景传输和计算处理的数据粒度,提升了WebBIM可视化效率。

(2) 问题二:架构问题。在WebBIM可视化模式中,架构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模式架构等,但是其各自存在着一定问题。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计算和渲染全部放置云端进行,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则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延迟制约了其进一步发展。本文提出的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,有效提升WebBIM可视化效果。

(3) 问题三: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,一般而言,用户愿意等待的时间相当有限,较长的等待时间可能直接导致用户失去等待的耐心,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,降低了初始加载延迟时间,同时选择了对于用户而言体验较好的初始观瞻角度,提升了用户体验。

本文后续各部分安排如下:第一部分介绍本文的相关工作。第二部分介绍本文的系统架构设计。第三部分介绍本文的技术路线。第四部分介绍基于兴趣度的多粒度化调度传输机制。第五部分介绍最优视点选取算法。第六部分介绍相关实验设计、实验结果及对结果的分析。最后,第七部分总结了本文的工作,并对未来的工作进行了展望。

1 相关工作

针对上述提到的技术与问题,本节将对边缘计算以及WebBIM在线可视化的当前研究进展进行探讨。

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维不同,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,使得服务器的处理响应速度和可靠性都高于云计算。2014年,ETSI提出对移动边缘计算^[2]术语的标准化^[3]。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工作组将Mobile Edge Computing正式更名为Multi-access Edge Computing^[5],相较于之前,除了强调无线网络的接入方式外,也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,对于Web3D可视化的实时计算和处理,也更加高效。2017年,Hou等人初步实现了轻量级的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了比较^[6]。其后,Jacob发表了一篇基于边缘计算的VR/AR沉浸式交互的论文,其在5G基站上进行缓存和计算,实现了一个5G无线360度VR/AR视频展示^[7],探讨了该框架的缓存机制、传输以及边缘计算方式。2018年,Erol-Kantarci进一步探讨了5G网络中基于移动边缘计算的VR/AR缓存与计算机制^[8]。2019年,Xu等人提出了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^[9],推进了VR与边缘计算的研究。

1.2 Web3D在线可视化模式

由于虚拟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计算、渲染任务放到浏览器端,在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。王珩玮等使用IFC Optimizer工具对BIM文件进行了语义轻量化处理,并使用Three.js框架在浏览器上将BIM模型可视化出来^[10]。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场景时的延迟均会变得十分严重。张东培等针对BIM虚拟场景,建立了适用于Web3D现实的轻量级场景结构,并提出了渐进式兴趣区域网格化场景管理策略(PAOI, Progressive Area Of Interest),对场景室内/外场景实现按需加载,降低了内存开销^[11]。刘小军、胡永豪等进行了进一步拓展,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渲染对应区域^{[12][13][14]},其提供了一种有效的针对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

纯云端渲染模式则将所有计算和渲染等任务放在云端进行，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 提出了 Cloud-BIM 方法^[15]，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 客户端进行显示，用户可以随时随地通过 Web 浏览器对 BIM 模型进行交互操作。Moumita 提出了基于云的 BIM 服务器框架^[16]，能够对建筑模型进行动态合并以及拆分。Chen 提出了更为系统全面的 Cloud-BIM 解决方案^[17]。刘强开发了基于云计算的 BIM 数据集成和管理平台^[18]，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，不可避免的会带来严重的渲染延迟和传输延迟。

1.2.3 云页协同可视化模式

协同可视化模式则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两种模式的矛盾，降低了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 等人提出了 CloudLight 系统^[19]，使用三种不同策略的光照和传输方法使前端支持全局光照。刘畅、邵威等人也提出了协作式云烘焙系统，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^{[20][21]}。

表 1 给出了三种模式的具体对比。云端、浏览器端协同渲染模式相对其他两种模式存在一定优势，但是其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会不太理想。

表 1 三种模式对比

	纯网页端 模式	纯云端 模式	云端、浏览器端 协同模式
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	长	中
漫游延迟	低	高	中

本文提出了一种基于云边页协同的 WebBIM 大场景多粒度兴趣加载调度算法，有如下创新点：

- (1) 使用了边缘计算进行 WebBIM 数据的就近处理，提出了基于边缘计算的 WebBIM 服务框架。通过将对时延要求较高的密集型工作转移至网络边缘侧的邻近服务器上，解决传统云计算在时延和带宽方面的限制，新兴的边缘计算提供了一种新的解决方案。本文将 WebBIM 与边缘计算进行了结合，提出了 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，有效提升 WebBIM 可视化效果。
- (2) 使用了基于兴趣度的多粒度调度传输机制。对于完整的大规模场景，由于用户任意时刻都仅能观察到场景的部分构件，传输和渲染其全部的数据会带来严重的负担，导致加载和渲染时间过长，影响用户体验。本文通过在边缘端实时同步用户视点，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，提升了传输和渲染效率。
- (3) 使用了最优初始视点选取算法选取 WebBIM 场景的初始加载视点位置。初始加载等待时间是 Web BIM 场景可视化效果的重要指标之一，用户愿意等待的时间一般相当有限，尽可能降低初始加载等待时间对于提升用户体验极其重要，而 WebBIM 场景每次初始加载的视点位置总是一致的。最优初始视点选取主要通过选择合适的视点位置和观察角度，使得初始加载的数据量在尽可能少的同时选择了对于用户而言体验较好的初始观瞻角度，降低了初始加载等待时间，提升了用户体验。

2 WebBIM 系统架构

2.1 设计原理

系统主要包含三层：Cloud、Edge 以及 Browser 层，三层在处理性能、访问频度以及存储能力均存在较大差异，具体如下：

- (1) Cloud：处于云中心位置，其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，但是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，处理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调度传输处理以及初始加载优化。
- (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、渲染以及用户的交互控制等。

2.2 系统架构

本文的系统架构如图 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 端则主要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调度传输处理，将场景的细粒度数据发送至 Browser 端进行可视化。

相较于传统的浏览器-服务器 BS 模式，CEB 协同模式引入了边缘端，将 BS 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 端进行处理，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，实现了系统的高效率运作。

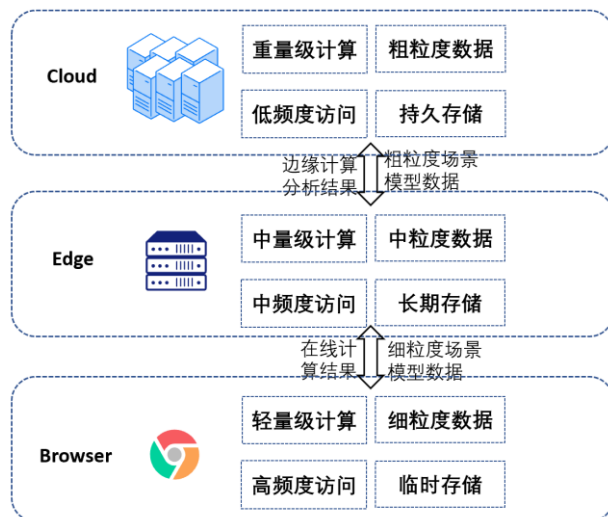


图 1 系统架构

3 技术路线

基于上述的系统架构，本文的技术路线设计大致如图 2 所示，具体为：

- (1) **Cloud:** 主要是对原始 BIM 场景进行轻量化预处理，以降低场景的数据量。基于我们的前期工作(参考文献 12 和 13)的基本思想，本文处理过程为：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室外；然后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:** 主要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，主要包含场景视点同步、计算可视增量列表、兴趣度计算、自适应打包以及按优先级传输。而初始加载优化主要是针对初始加载进行的预处理，包含最优初始视点选取、计算可视增量列表、在线打包以及本地持久存储，其中最优初始视点选取是核心部分。
- (3) **Browser:** 主要负责场景在网页上的轻量级在线可视化，主要包含实时视点更新、按序加载、实例化渲染、用户交互控制等。

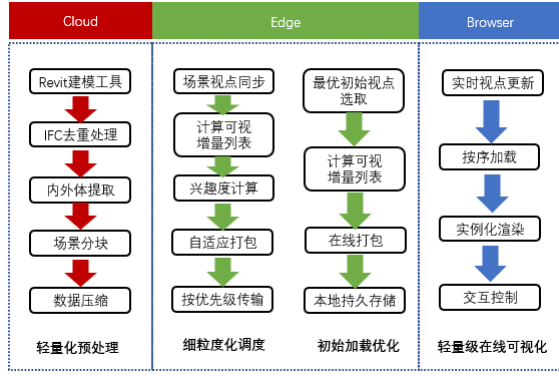


图2 技术路线

4 基于兴趣度的多粒度调度传输机制

对于大规模场景，在漫游或显示中，用户任何时刻都仅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，通过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，比如体积大、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，本文引入了兴趣度的概念。

定义 1: 填充度 (Fill Degree, FD): 模型所占的包围盒体积与其文件大小的比值，计算公式为：

$$FD(m_i) = \frac{Volume(BBox(m_i))}{Size(m_i)} \quad (1)$$

填充度反应了模型单位传输消耗下于 3D 空间中的所占空间大小，如图 3 所示，相较于 Model B，Model A 的包围盒体积更大，而文件更小，对用户视觉影响更大，因此填充度更大，兴趣度（反应了传输的优先级）也就更高。

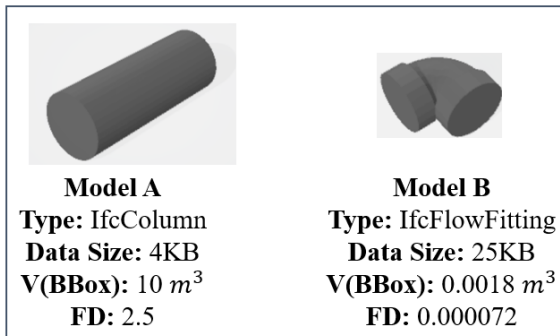


图3 模型填充度

定义 2: 重用度(Reuse Degree, RD): 模型的实例构件数量

与其文件大小的比值，计算公式为：

$$RD(m_i) = \frac{Num(instance(m_i))}{Size(m_i)} \quad (2)$$

重用度反应了模型在单位传输消耗下于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，由于使用了实例化渲染技术，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，重用度也就越大，传输优先级就越高。

Model C	Model D
Type: MechanicalFastener	Type: MechanicalFastener
Data Size: 20KB	Data Size: 18KB
Num(instance): 5	Num(instance): 3
RD: 0.25	RD: 0.17

图4 模型重用度

定义 3: 关注度 (Attention Degree, AD): 反应了模型在单位传输消耗下于用户对其可能的关注程度高低，主要由该模型的类型系数TypeLevel、交互次数统计 N_i 综合得到，公式为：

$$AD(m_i) = \frac{TypeLevel(m_i) + N_i(m_i)}{Size(m_i)} \quad (3)$$

关注度主要是考虑到用户对场景中模型的关注程度，其中模型的类型系数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瞻影响较大，而建筑物内部的管道、螺丝、柱子等相对影响较小。其次，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关注度。当模型关注度越高，传输优先级越高，用户体验越好。

定义 4: 兴趣度(Interest Degree, ID): 填充度、重用度以及关注度的加权结果，综合反应了该模型对于用户视觉影响的贡献大小以及其文件传输的代价，计算公式为：

$$ID(m_i) = \alpha_1 \times FD(m_i) + \alpha_2 \times RD(m_i) + \alpha_3 \times AD(m_i) \quad (4)$$

其中 $\alpha_1 + \alpha_2 + \alpha_3 = 1$ ， $\alpha_1, \alpha_2, \alpha_3$ 为各自的权值，其中权值的确定需要在实验中多次重复调整得到。本文中选取方法是使用控制变量法，权值以 0.1 为递加/减单位。由于兴趣度主要用于量化模型在传输过程中的优先级，因此选取的依据主

要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)。

为了更好地阐述兴趣度在模型传输中的作用,图 5 给出了一套模型分别在两种加载序列下(兴趣度从高到低和兴趣度从低到高)的完整加载过程对比,可以看出当模型加载序列为兴趣度从低到高时,开始加载的模型对用户视觉影响很小且加载速度慢,直到模型基本全部加载完毕用户才能观瞻场景的全貌,而当模型加载序列为兴趣度从高到低时,实际上场景加载一半左右时用户就可以观瞻到场景的全貌,视觉影响较小的细节随后逐步加载,用户体验明显更优。

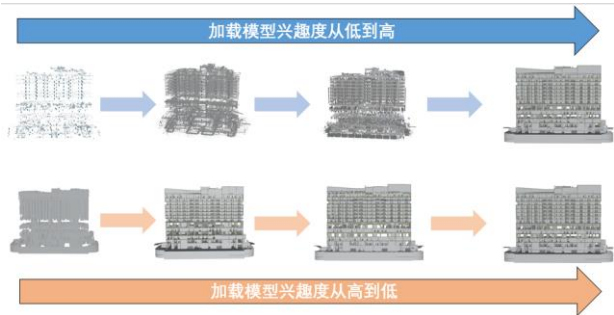


图 5 模型加载序列对比

表 2 给出了某测试场景中的部分模型的兴趣度计算案例,其中权值 $\alpha_1, \alpha_2, \alpha_3$ 在实验中取值分别为 0.3, 0.5, 0.2。

表 2 兴趣度计算

Type	Size (kb)	FD	RD	AD	ID	Priority
Slab	0.35	34.29	11.43	5.71	17.14	1
Covering	1.21	3.31	4.16	1.65	3.39	2
Column	3.92	2.55	1.53	0.51	1.64	3
Door	4.48	0.02	0.67	0.67	0.47	4
Window	6.34	0.01	0.47	0.47	0.33	5
Railing	6.14	0.00	0.49	0.16	0.28	6
Mechanical Fastener	19.23	0.00	0.26	0.05	0.14	7

4.2 多粒度调度传输机制流程

本文使用了基于兴趣度的多粒度化调度传输机制,流程图如图 6 所示,具体过程为:

- (1) 基于我们的前期工作(参考文献 12 和 13)的基本思想,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。
- (2) 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,颜色与模型对照表中存储了渲染模型时所使用的颜色信息,主要用于模型的辨识;

- (3) 用户打开浏览器,Web 端建立与 Edge 端的数据连接;
- (4) Edge 端实时获取用户的视点信息(相机位置、朝向等),以用于在 Edge 端同步用户的视点;
- (5) 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当前视点下所需传输的最小模型集合;
- (6) 计算可视增量模型列表中所有模型的兴趣度,然后依据用户的带宽大小进行自适应打包传输。兴趣度反应了模型对用户视觉的影响贡献大小,其决定了该模型在场景传输中的先后顺序;
- (7) 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,实现了在一次视点更新中场景的完整加载过程。

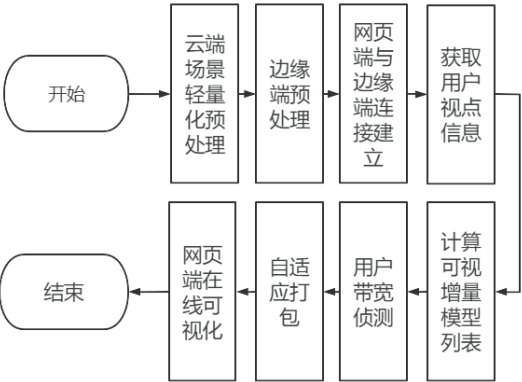


图 6 多粒度调度传输机制流程图

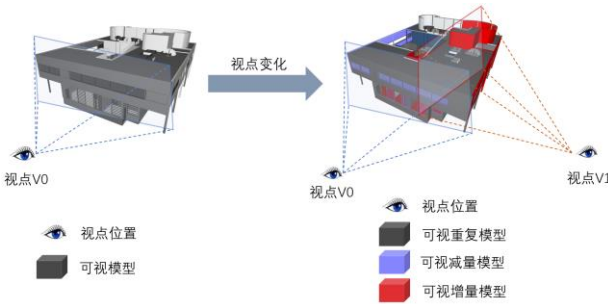


图 7 场景可视增量模型示意图

4.3 自适应打包和传输策略

每次网络请求和传输均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 浏览器得以优先渲

染，获得传输效益最大化。因此，本文采用了自适应打包机制，具体流程如下：

- (1) 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 BW；
- (2) 设定包的最大传输时间 T（本文中使用时 T=1S），将文件包的大小 PS 设定为 BW*T；
- (3) 将之前得到的可视增量列表中的构件模型按照其兴趣度的高低进行排序，依次进行打包合并；
- (4) 将得到的文件包按照次序进行传输。

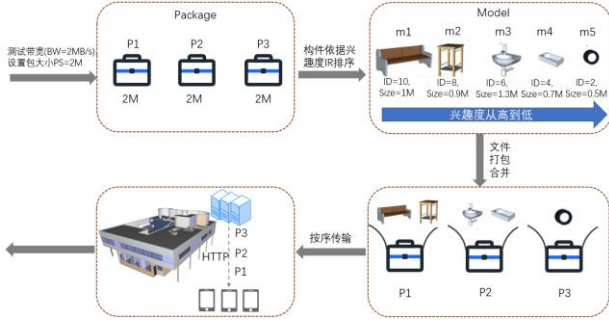


图 8 自适应打包示意图

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。最优初始视点选取主要通过选择合适的视点位置和观察角度，使得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见模型的文件数据量的比值，其计算公式分别为：

$$S_v = \frac{\sum_{i=1}^m S_i + \sum_{i=1}^{N-m} S'_i}{\sum_{i=1}^m S_i} \quad (5)$$

式中 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， m 为该视点下可视模型的数量。

可视规模度反应当前视点下可视模型文件数据量的比例，可视模型数据量越小，可视规模度越大，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 R_v 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，其计算公式为：

$$R_v = \frac{\sum_{i=1}^N s_i}{\sum_{i=1}^m s'_i + \sum_{i=1}^{N-m} s_i} \quad (6)$$

s_i 为各个模型的原始文件大小， s'_i 是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， N 为该视点下的可视模型总数。

由于实例化模型在预处理阶段经过过去重处理，其实例化文件相对极小，此外由于使用了实例化渲染技术，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 F_v 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，用户的观瞻细节就越多，屏幕空间利用越充分，其计算公式为：

$$F_v = \frac{V + V_0}{V} \quad (7)$$

V_0 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

根据以上指标，给出如下视点优先级 P 的计算公式：

$$P = w_0 S_v + w_1 R_v + w_2 F_v \quad (8)$$

其中， $w_0 + w_1 + w_2 = 1$ ，与兴趣度相似，权值使用控制变量法进行选取，以 0.1 为递加/减单位。由于视点优先级主要是用于选择合适的初始加载视点位置，以降低初始加载时间的同时加载更多的对用户视觉影响更大的模型，因此可以依据初始加载完成后场景模型对用户的视觉影响（通过计算初始加载完成后场景占用的屏幕像素数量）和初始加载时间之比作为衡量标准进行选取。

图 9 给出了某测试场景中的视点优先级与视点位置的实例，其中相应的权值 w_0 、 w_1 、 w_2 在实验中分别为 0.3、0.4 和 0.3，View C 的视点优先级高于 View A 和 View B，View C 为相对更优的初始视点。

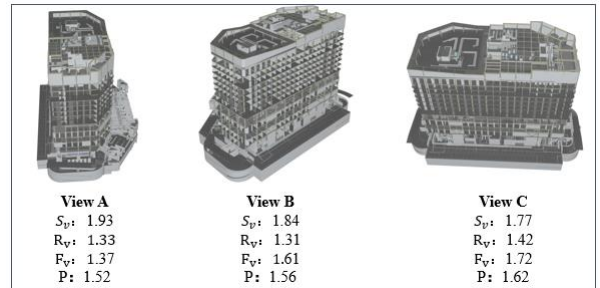


图 9 视点优先级计算

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进行选取，过程如图 10 所示，具体为：

- (1) 计算场景的 AABB 包围盒，并以包围盒中心为球心，计算出包围盒的外接球。
- (2) 为了获得比较可靠的视角，把视角集中在一个有效区域内，减少搜索所消耗的时间，本文取上半球上距离 XOZ 平面 $15^{\circ}\sim 60^{\circ}$ 并且距离 YOZ 平面（Z 轴方向为场景正面朝向，具体参见图 11） $-60^{\circ}\sim 60^{\circ}$ 之间的曲面作为场景的最优初始视点待选区域。
- (3) 使用最优初始视点搜索算法计算出场景的最优初始视点，算法具体如表 3 所示。
- (4) 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。

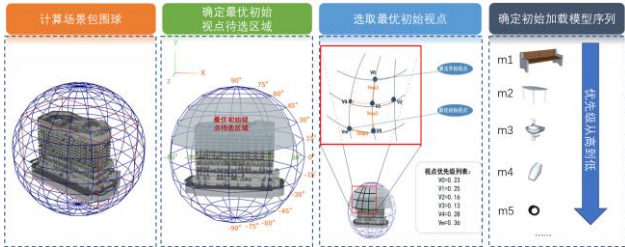


图 10 最优初始视点选取

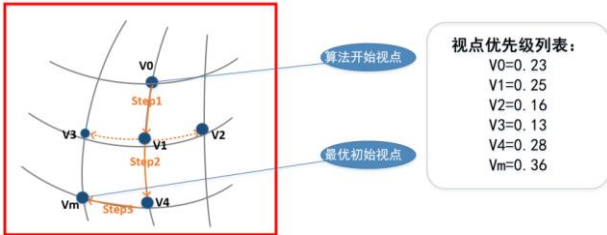


图 11 步骤 3 最优初始视点搜索示意图

表 3 最优初始视点搜索算法

算法： 生成最优初始视点	
输入： 算法搜索开始视点	
输出： 场景加载最优初始视点	
1.	Begin
2.	输入算法搜索开始视点；
3.	将该视点置为当前视点，计算当前视点优先级；
4.	If 当前视点的左方邻点未搜索
5.	计算左方邻点的优先级；
6.	If 左方邻点的优先级 > 当前视点优先级
7.	将左方邻点置为当前视点，跳转至步骤 3；
8.	If 当前视点的上方邻点未搜索
9.	计算上方邻点的优先级；
10.	If 上方邻点的优先级 > 当前视点优先级
11.	将上方邻点置为当前视点，跳转至步骤 3；
12.	If 当前视点的右方邻点未搜索

13. 计算右方邻点的优先级；
14. If 右方邻点的优先级 > 当前视点优先级
15. 将右方邻点置为当前视点，跳转至步骤 3；
16. If 当前视点的下方邻点未搜索
17. 计算下方邻点的优先级；
18. If 下方邻点的优先级 > 当前视点优先级
19. 将下方邻点置为当前视点，跳转至步骤 3；
20. 当前视点即为最优初始视点，输出当前视点。
21. End

6 实验结果

为了验证提出方法的有效性，本文依托阿里云搭建了边缘计算平台，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，并与位于北京的云服务器作为测试对比。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

本文使用了多个测试场景进行测试，具体如下：

- (1) 住宅楼 BIM，数据量 120MB，面片数量为 198 万，效果如图 12 所示。
- (2) 地下车库 BIM，数据量 230MB，面片数量为 310 万左右，效果如图 13 所示。
- (3) 地铁站 BIM，其数据量为 920MB，面片数量为 1200 万左右，其场景效果如图 14 所示。
- (4) 双子塔 BIM，其数据量为 1.61GB，面片数量为 2000 万左右，其场景效果如图 15 所示。
- (5) 大型医院 BIM。数据量 2.2GB，面片数量为 3300 万左右，其场景效果如图 16 所示。

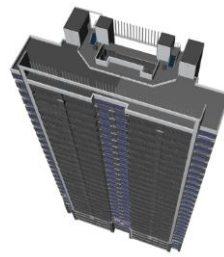


图 12 某住宅楼 BIM (120MB)

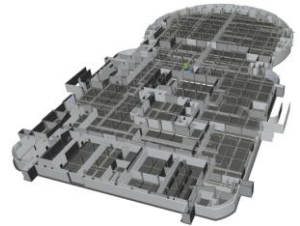


图 13 某地下车库 BIM(230MB)

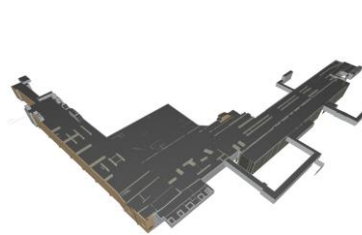


图 14 某地铁站 BIM (920MB)

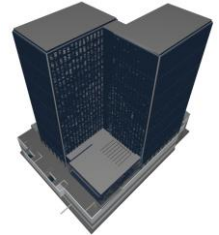


图 15 双子塔 BIM (1.61GB)

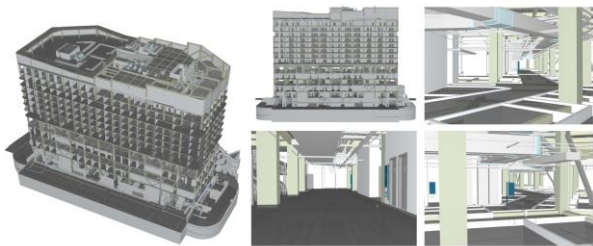


图 16 某大型医院 BIM (2.2GB)

6.2 测试环境

- (1) Master 节点配置:
OS: Ubuntu 16.04
CPU: Intel Xeon Platinum 8269
内存: 16GB
- (2) 边缘(云)服务器配置:
OS: Ubuntu 16.04
CPU: Intel Xeon Platinum 8163
GPU: Nvidia Tesla T4
内存: 23GB
- (3) 网络带宽: 服务器带宽 40Mbps, 客户端 4G 移动网络



图 17 边缘节点分布图

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构件模型文件开始到边缘端或云端接收到相应数据的时间间隔,测试文件大小分别设为 30KB、50KB、100KB、200KB 以及 400KB 左右,测量了引入边缘计算前后的网络时延,表 4 为测试结果。从实验结果大致可以看出,在引入边缘计算之后,网络时延得到大幅降低,存在明显优势。

表 4 网络时延测试结果

文件大小/平均	30KB	50KB	100KB	200KB	400KB
---------	------	------	-------	-------	-------

时 延 (m s)					
云 服 务 器	31	43	113	173	308
边 缘 服 务 器	19	23	29	35	51

6.3.2 初始加载时间

初始加载时间即从用户打开网页到出现模型所间隔的时间,初始加载时间越短,用户体验越好。为了更全面地测试出各部分的性能,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 5 为初始加载等待时间测试结果,从实验结果可以看出,在分别引入多粒度化调度、最优视点选取、边缘计算后,均带来了较为明显的性能提升,其中在引入边缘计算后,其速度提升效果尤为明显。

图 18 展示了在各个场景下纯网页端模式(网页端直接加载渲染完整场景)、CB 协同架构方案(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 协同方案三种模式的初始加载速度对比结果,可以看出 CB 协同架构方案效果优于纯网页端模式,而 CEB 协同方案加载速度明显快于上述二者。

表 5 初始加载时间测试结果

场景/平均加载时间(s)	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

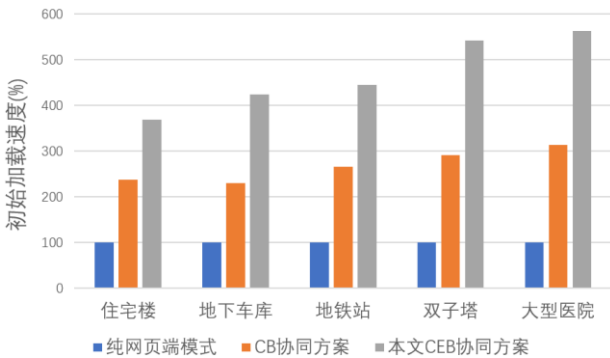


图 18 初始加载速度对比结果

6.3.3 帧率

帧率反应了用户在场景漫游时的流畅度,帧率越高,

用户体验感越好。本文对比测试了多个场景下纯网页端模式、CB 协同架构方案与本文提出的 CEB 协同方案场景加载及漫游中的平均帧率，图 19 为对比测试结果。从图中可以看出，在各个场景的加载中，纯网页端加载模式的帧率最低，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，存在较为明显的优势。

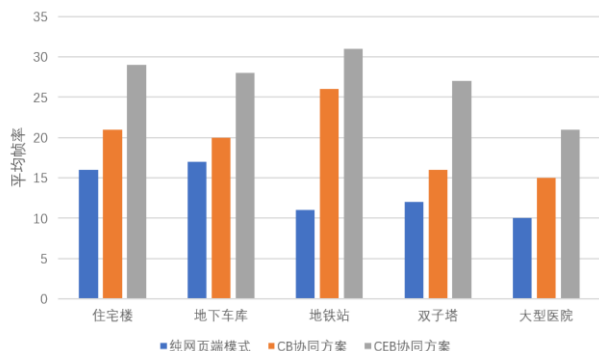


图 19 帧率对比结果

6.3.4 漫游加载延迟

漫游加载延迟指的是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，因此这里不再作比较。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 20 给出了 CB 协同方案与本文提出的 CEB 协同方案的漫游加载延迟对比结果，可以看出 CEB 协同方案存在一定优势，但在不同场景中存在一定差异，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完了大部分模型，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。

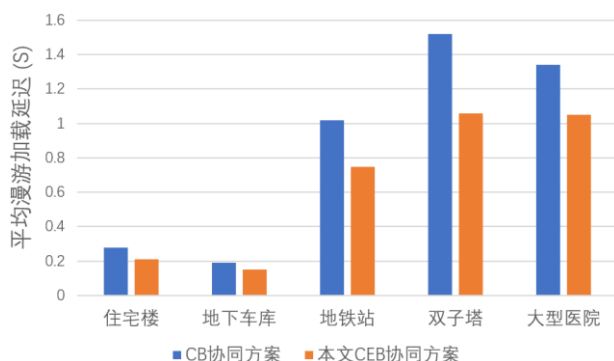


图 20 帧率对比结果

7 结论与展望

本文提出了基于云边页协同的 WebBIM 大场景多粒度

兴趣加载调度算法，在传统的 BS 模式中引入边缘计算，内容主要体现在如下三个方面：首先，使用了边缘计算进行 WebBIM 数据的就近处理，提出了针对 WebBIM 场景在线可视化的云端、边缘端、浏览器端协同运作框架；其次，使用了基于兴趣度的多粒度调度传输机制，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，提升了可视化效率；然后，使用最优初始视点选取算法选择场景加载的初始视点，进一步提升了场景初始加载的速度。最后，设计了相关实验进行验证，实验结果表明该方案能够有效降低 WebBIM 网络传输时延，提升场景的加载速度，优化用户体验。

当然，本文还有很多方面值得进一步探索和完事，接下来会从如下方面进行：

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

(2) 高级渲染、纹理：本文更加注重于大规模 BIM 场景的加载、调度传输等，暂且还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，针对这方面的内容也是本文接下来的工作之一。

(3) 权值的确定：本文在兴趣度和视点优先级的计算中均是各个相关部分进行加权，其中权值的确定相对较为繁琐，需要在实验中多次重复调整得到。本文中选取方法是使用控制变量法，权值以 0.1 为递加/减单位。选取依据二者的目标有所差异，由于兴趣度主要用于量化模型在传输过程中的优先级，因此选取的依据主要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场景占用的屏幕像素数量进行量化）。而视点优先级主要是用于选择合适的初始加载视点位置，降低初始加载时间的同时加载更多的对用户视觉影响更大的模型，因此可以依据初始加载场景占用的像素数量和初始加载时间进行选取。在接下来的工作中，会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：

- [1] W. Shi, J. Cao, Q. Zhang, Y. Li and L. Xu, "Edge Computing: Vision and Challenges," in IEEE Internet of Things Journal, vol. 3, no. 5, pp. 637-646, Oct. 2016.
- [2] ETSI. Executive Briefing - Mobile Edge Computing (MEC) Initiative, 2014.
- [3] Etsi M. Mobile-edge computing[J]. Introductory Technical White Paper, 2014.
- [4] Zhao T, Zhou S, Guo X, et al. A cooperative scheduling scheme of local cloud and internet cloud for delay-aware mobile cloud computing[C]//2015 IEEE Globecom Workshops (GC Wkshps). IEEE, 2015: 1-6.

- [5] European Telecommunication Standards Institute (ETSI). Multi-access Edge Computing(MEC)[OL].[2018-09].
<https://www.etsi.org/technologies/multi-access-edge-computing>
- [6] Hou X, Lu Y, Dey S. Wireless VR/AR with edge/cloud computing[C]//2017 26th International Conference on Computer Communication and Networks (ICCCN). IEEE, 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. ACM, 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[M]//Ad Hoc Networks. Springer, Cham, 2018: 169-177.
- [9] T. Xu, Y. Sun, S. Xia, H. Li, L. Luo and Z. Chen, "Optimal Bandwidth Allocation with Edge Computing for Wireless VR Delivery," 2019 IEEE/CIC International Conference on Communications in China (ICCC), Changchun, China, 2019, pp. 903-907.
- [10] 王珩玮, 胡振中, 林佳瑞. 面向 Web 的 BIM 三维浏览与信息管理[J]. 土木建筑工程信息技术, 2013 (3).
- [11] 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] Xiaojun Liu, Ning Xie, Kai Tang, Jinyuan Jia. Lightweight for Web3D Visualization of Large-scale BIM Scenes in Real-time[J]. Graphical Models, vol. 88, pp. 40-56, November, 2016.
- [13] 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. 中国科学: 信息科学, 2018, 48: 274-292.
- [14] Hu Y.H., Chen C.H., Liu X.J., Huang F., Jia J.Y.. WebTorrent Based Fine-grained P2P Transmission of Large-scale WebVR Indoor Scenes. ACM Web3D, Brisbane, Australia, 2017.
- [15] Chuang T H, Lee B C, Wu I C. Applying cloud computing technology to BIM visualization and manipulation[C]. International Symposium on Automation and Robotics in Construction. 2011. pp: 144-149
- [16] Moumita Das, Jack.C.P Cheng, and S. S. Kumar. "Social BIMCloud: a distributed cloud-based BIM platform for object-based lifecycle information exchange." Visualization in Engineering 3.1(2015):1-20.
- [17] Hung-Ming Chen, Kai-Chuan Chang, Tsung-Hsi Lin. A cloud-based system framework for performing online viewing, storage, and analysis on big data of massive BIMs. Automation in Construction. Vol. 71:34-48. 2016.
- [18] 刘强. 基于云计算的 BIM 数据集成与管理技术研究[D].清华大学, 2017.
- [19] Crassin C, Luebke D, Mara M, et al. Cloud Light: A system for amortizing indirect lighting in real-time rendering[J]. Journal of Computer Graphics Techniques Vol, 2015, 4(4).
- [20] Chang Liu, Lei Zhao, Wei Tsang OOI, Jinyuan Jia. Cloud Baking: Collaborative Scene Illumination for Dynamic Web3D Scenes". ACM Transactions on Multimedia Computing Communication and Applications, April, 14(3s): 59:1-59:20, 2018.
- [21] 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.

修改答复

Reviewer A: BIM 系统的普及应用带来了相关内容的爆发性增长，而 web 平台（浏览器或 web 渲染引擎）是 BIM 内容的重要呈现方式之一，针对 WebBIM 大场景的有效渲染进行研究具有显著意义。本文对 WebBIM 的加载优化研究现状进行了系统全面的总结综述，提出了基于多种粒度的 WebBIM 大场景加载策略。所提出的策略和技术实现通过多个应用场景数据集进行了实验验证。论文结构合理，文字流畅，参考文献引用全面。作者团队在这方面的工作时间很长，积累成果也非常丰富，建议在本论文中，对工作的延续性和增量细节做更多详细强调。

Answer A: 本文接下来的工作主要有以下三个方面：

- (1) 使用基于 5G 的移动边缘计算来进一步扩展，随着 5G 技术的成熟和规模应用，基于 5G 的 MEC 计算框架将成为可能，相较于本文部署的若干割裂独立的边缘域，MEC 服务器可以提供更加普及、高效的服务。
- (2) 使用高级渲染、纹理等进一步完善场景，本文当前注重于大规模 BIM 场景的加载、调度传输等，暂且还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，针对这方面的内容也是本文接下来的工作之一。
- (3) 优化本文的部分概念及计算，本文在兴趣度和视点优先级的计算中均是各个相关部分进行加权，其中权值的确定本文中选取方法是使用控制变量法，权值以 0.1 为递加/减单位，需要在实验中多次重复调整测试得到，相对较为繁琐，在接下来的工作中，会尝试优化合并各自的计算方式，简化权值的选取过程。

Reviewer B: 论文围绕基于云边页协同的 Web BIM 大场景多粒度兴趣加载调试算法展开研究，选题具有较好的学术创新性和应用价值，论文结构清晰、框架合理、表述明确、实验扎实，建议录用并推荐期刊。具体意见包括：

(1) 作者在 4.1 中提出了兴趣度的概念与计算方法，可进一步设计实验验证这一概念和计算方法的科学性与适用性，并解释权重是如何选择的？

(2) 第 5 部分的最优初始视点选取同样需要分析计算公式和权重选择的科学性，并在实验部分对其分析结果进行支撑。

Answer B1: 在本文的第 4.1 和 5.1 节中已增加相关部分内容，并己用红色标出。

本文已经在实验中通过多个测试场景对其进行了测试，其具有普遍适用性，已在 4.1 节部分增加更详细的解释以及部分计算和实验结果。其中权值的确定是在实验中多次重复测试并取最优结果得到的，本文中选取方法是使用控制变量法，每个权值变化以 0.1 为递加/减单位。由于兴趣度主要用于量化模型在传输过程中的优先级，本文选取的依据主要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（通过计算固定时间如 5S 内渲染完成后所有模型占用的屏幕像素数量进行量化，该值最大者相应的权值即为确定的权值）。

Answer B2: 本文已在 5.1 节部分增加更详细的解释以及部分计算和实验结果，其权值的确定基本与兴趣度相似，仅选取标准存在差别，由于视点优先级主要是用于选择合适的初始加载视点位置，降低初始加载时间的同时加载更多的对用户视觉影响更大的模型，因此使用初始加载场景占用的总像素数量与初始加载时间之比作为衡量标准，该值最大者时计算最优初始视点位置使用的权值即为确定的权值。

Reviewer C: 论文研究有意义，方法正确，整体描述清晰，存在的一些问题是：（1）论文提高 5G 在移动网页浏览器展示 WebBIM 场景的需求，相关内容描述不多，建议增加这方面的描述和体现；包括实验部分是否可以增加相关实验内容？

（2）采用的一些方法如边缘计算等在一些已有工作中出现，本文工作中的区别和创新点需要增加和突出。（3）实验结果比较中，除了网络时延等一些性能比较外，还可增加与一些现有方法进行比较。

Answer C1: 使用基于 5G 的 MEC 来进一步完善 WebBIM 于移动浏览器上的展示是本文接下来的研究方向之一，受部分因素限制，本文还不具备进一步研究的条件：

- (1) 5G 在全国的部署还处于初步阶段，暂时还未普及，WebBIM 使用区域受限（仅能在测试区域范围内）；
- (2) 基于 5G 的 MEC 市面上暂时缺乏直接的服务提供，相对较为依赖于与移动、联通等运营商的合作。

Answer C2: 本文的创新点主要为三个方面：

- a) 提出并设计了针对 WebBIM 在线可视化的 CEB 协同架构，并给出了具体实现过程，在此之前很少人研究针对 WebBIM 的边缘计算；
- b) 对 WebBIM 的初始加载进行研究，并给出了 WebBIM 场景的最优初始加载视点选取方法，目前为止较少有针对 WebBIM 初始加载的研究工作，而初始加载效果则是衡量 WebBIM 在线可视化质量最重要的指标之一；
- c) 提出了基于兴趣度的多粒度传输调度机制。

Answer C3: 在本文的第 6.3 节实验部分，已增加纯网页端加载模式、CB 协同架构方案与本文提出的 CEB 协同方案三种模式各个测试指标的对比结果。