

一种面向 Web 的大规模点云可视化方法

任以燊, 乔秀全

(北京邮电大学网络与交换技术国家重点实验室, 北京 100876)

5

摘要: 在目前的导航领域中, 大多是基于二维平面地图的导航, 但随着我国社会发展, 城市规模的扩大化, 各种集群式建筑的兴起导致人们在实际导航中会遇到诸多问题, 在这样的背景下, 探索 AR 技术的导航系统能够发挥出其价值。WebAR 作为一种轻量化、跨平台的新兴技术为 AR 导航的发展提供了新的方向。然而在目前的 WebAR 导航应用中, 渲染大规模点云会遇到内存限制、实时渲染性能需求等挑战, 缺乏一种面向 Web 的大规模点云可视化方法。本研究针对这些挑战, 提出了一个面向 Web 的大规模点云可视化方法, 将点云按照可修改的嵌套八叉树结构进行数据组织, 并引入了泊松盘采样算法, 改进了层级索引方式, 最后基于细节层次按可见性判断方法按需加载, 为实现面向 Web 的大规模点云可视化提供了一个有效解决方案。

10

15

关键词: 计算机应用; WebAR; 三维点云; 空间索引; 可视化;
中图分类号: TP399

A Web-Oriented Approach to Large-Scale Point Cloud Visualization

20

Ren Yican, Qiao Xiuquan

(State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing 100876)

25

Abstract: In the current navigation field, most of the navigation is based on two-dimensional flat map, but with the development of China's society, the enlargement of the city scale, the rise of a variety of clustered buildings lead to people in the actual navigation will encounter a lot of problems, in such a background, the exploration of AR technology navigation system can play its value. WebAR as a lightweight, cross-platform emerging technology for the development of AR navigation WebAR as a lightweight and cross-platform emerging technology provides a new direction for the development of AR navigation. However, in current WebAR navigation applications, rendering large-scale point clouds encounters challenges such as memory constraints, real-time rendering performance requirements, and the lack of a Web-oriented large-scale point cloud visualization method. This study addresses these challenges and proposes a Web-oriented large-scale point cloud visualization method, which organizes the point cloud according to a modifiable nested octree structure, introduces a Poisson disk sampling algorithm, improves the hierarchical indexing method, and finally loads on-demand based on the level of detail according to the visibility judgement method, which provides an effective solution for realizing the Web-oriented large-scale point cloud visualization.

30

35

Key words: Computer application technology; WebAR; 3D point cloud; data index; visualization

40

0 引言

随着我国社会发展, 城市规模的扩大化, 各种集群式建筑的兴起导致人们在实际导航中会遇到诸多问题。传统的导航大多是基于二维平面地图的导航, 比如百度地图、高德地图等, 其基于卫星定位的导航服务在室外场景中具有广泛的实用应用, 但是在室内大型建

作者简介: 任以燊(1998-), 男, 硕士研究生, 主要研究方向: WebAR

通信联系人: 乔秀全(1978-), 男, 教授, 主要研究方向: ARVR, 计算机视觉, 边缘计算. E-mail: qiaoxq@bupt.edu.cn

45 筑中，传统的导航软件受制于环境因素和技术因素，导致起不到显著效果，在这样的背景
下，探索 AR 技术的导航系统能够发挥出其价值，通过将虚拟信息叠加在真实世界中，能
够让用户更快地定位环境，找到目标，获得更直观丰富的导航体验。WebAR 是一种在
Web 浏览器上实现增强现实体验的技术，其具有轻量级和跨平台的优势，用户只需要通过
浏览器访问便可便捷地体验 AR 应用。因此，面向 Web 的 AR 导航应用既有 AR 导航技术
50 的优点，又包含 WebAR 的轻量级和跨平台优势，是一个热门且具有实用意义的研究领
域。

在实际的 WebAR 导航应用中，有一种广泛应用的场景是针对采集到的三维点云地图
进行基于 Web 的三维可视化，使用户在浏览器上就可以直观的对三维点云地图进行查看和
管理。随着传感器技术的进步，我们已经可以快速准确地捕获物体表面的三维信息，从而
产生大规模的三维点云数据，这种点云数据的量级少则几 MB，多则上 GB，这对面向
55 Web 端的可视化渲染带来了极大的挑战。主要包含内存限制、实时渲染需求、网络传输速
度等难题。

针对以上所述的问题，需要对于点云渲染的各个阶段采取一定的措施，包括：数据预
处理、数据加载、渲染计算等，但归根结底，本质就在于给出高效的点云数据结构和索引
构建方式。因此，本文提出一种改进的嵌套八叉树的数据结构为点云构建索引，并引入了
60 细节层次结构与层级渲染等方法，按可见性判断按需渲染，为适配低性能设备提供了可
能。

1 相关工作

1.1 三维点云与数据索引

点云是一种表示三维空间中物体表面的数据结构，由在三维坐标系中分布的一系列点
65 组成。这些点通常通过各种三维扫描技术获取，如激光扫描（LIDAR）、立体视觉、结构
光扫描等。点云能够提供关于物体形状、大小和空间位置的详细信息，因此在许多领域如
自动驾驶、机器人导航、地形测绘、文化遗产保护、建筑信息模型（BIM）和虚拟现实等
都有广泛的应用^[1]。描的大规模的点云数据集如图 2-1 所示，其包含了 RGB 以及强度信
息。由于点云数据通常包含大量的点（可能达到数百万甚至数十亿），高效的数据组织结
70 构变得至关重要。有效的数据组织不仅能够提高数据处理和查询的效率，还能支持复杂的
三维空间分析和操作。以下是几种常见的点云数据组织结构及其相关技术：

（1）体素栅格（Voxel Grids）

将三维空间划分为规则的网格^[2]，每个体素存储在该区域内的点云信息。这种方法可
以大大减少数据量，通过降采样简化点云，但可能会丢失一些细节信息。

75 （2）八叉树（Octree）

八叉树^[3]是一种树状数据结构，将三维空间递归划分为八个八分体。每个节点代表一
个空间区域，叶节点存储实际的点云数据。八叉树支持高效的空間查詢和分区操作，适用
于点云的快速访问和处理。

（3）k-d 树（k-dimensional tree）

80 k-d 树是一种用于管理多维空间数据的二叉树结构^[4]。通过在每个节点上按照某一维度的值将数据集分割，k-d 树能够在点云数据中进行高效的搜索和邻近查询。

随着计算能力的提升和算法的发展，点云及其数据组织结构的相关技术正不断进步，使得点云数据的处理更加高效和精确。这些技术的改进创新，为点云在各个应用领域的利用提供了强大的支持，推动了数据处理和分析技术的发展。

85 1.2 Web 渲染技术

Web 端渲染引擎是一种在浏览器中渲染图形内容的技术，它允许开发者使用 HTML、CSS 和 JavaScript 等 Web 技术来创建复杂的 2D 和 3D 图形应用。随着 Web 技术的发展，特别是 WebGL（Web Graphics Library）的出现，Web 端的图形渲染能力得到了极大的增强，使得在浏览器中运行高性能的图形应用成为可能。下面介绍一下相关的渲染引擎和技术。

90

1.2.1 WebGL

WebGL^[5]（Web Graphics Library）是一种在不需要使用插件的情况下在网页浏览器中进行 3D 绘图的技术。它是一种低级的、基于 OpenGL ES 的 API，允许开发者利用客户端的图形处理器（GPU）直接在网页上渲染复杂的 3D 图形和动画。通过 JavaScript 编程接口，WebGL 提供了创建实时图形、游戏、可视化数据和交互式 3D 应用的能力，从而极大地丰富了 Web 应用的表现力和互动性。WebGL 的出现标志着 Web 技术的一个重要进步，使得开发者能够构建出以往只能在桌面应用中见到的高性能、高质量的 3D 体验，而用户

95

则可以直接在最新版的网页浏览器中享受到这些内容，无需安装额外的软件或插件。Three.js 是目前最流行的 Web 端 3D 渲染引擎之一，它是一个基于原生 WebGL 封装的高级 JavaScript 库。通过提供一套简洁的 API，Three.js 让开发者能够轻松地在 Web 浏览器中创建和渲染 3D 场景，而无需深入了解复杂的 WebGL 概念和细节。

100

1.2.2 Three.js

Three.js^[6]是一个基于 WebGL 的 JavaScript 库，它提供了一套简洁的 API 来创建和显示三维图形内容在网页浏览器中。通过抽象 WebGL 的复杂性，Three.js 使得开发者能够轻松地在网页上构建和渲染三维场景、模型、动画、粒子系统等，无需深入了解 WebGL 的复杂底层实现。

105

2 框架设计

面向 Web 的大规模点云可视化系统的框架图如图 1 所示：

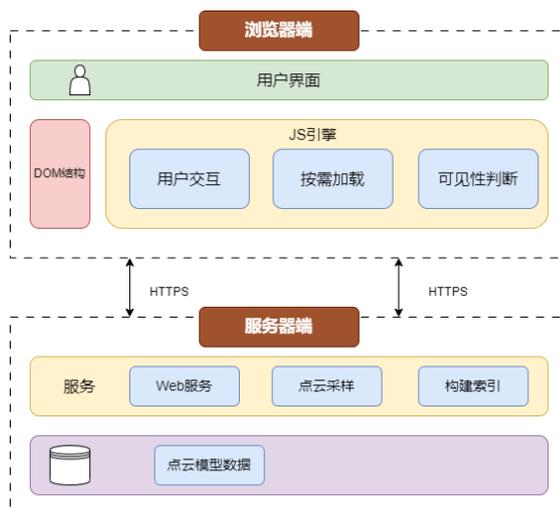


图 1 系统框架

Fig.1 System Framework

110

115

主要分为服务端以及浏览器端，首先在浏览器端，用户上传点云文件，并将点云文件存储在数据库中；在服务端，调用点云模型进行处理，主要包括构建可嵌套的八叉树索引，并进行基于泊松盘采样的采样优化，然后建立分层索引文件，按照可见性判断设计细节层次模型，最终交给浏览器端的 Web 渲染引擎进行按需加载以及实现实时交互操作的功能。

3 方法实现

3.1 构建点云空间索引

120

125

首先，对点云进行数据组织索引的构建，采用一种可修改的嵌套八叉树(MNO)结构为点云构建空间索引^[7]。递归地将三维点云空间划分为更小的立方体区域（称为节点），从而实现对空间数据的高效组织。每个节点在八叉树中分裂成八个子节点，这一分裂过程基于三维空间中的每个维度进行，直到满足特定的终止条件，如节点内数据点数量少于预设阈值或达到最大深度限制。针对 MNO 构建时相邻单元间的间隔距离可能很短，影响采样样本的质量的问题，引入了空间加速的泊松盘采样算法，保证了每个点与其他点的最小间距，并且显著提升了采样的效率。

3.2 渲染优化

组织好点云的空间结构后，还需要针对点云的结构在渲染时采用一定的技术进行渲染优化，本文采用的方法主要包含可见性判断中的视锥体剔除以及细节层次模型^[8]。

3.2.1 视锥体剔除

130

135

可见性判断及剔除技术是计算机图形学中的一种关键优化方法，旨在提高三维图形渲染的效率和性能。通过判断场景中的对象是否在摄像机的视野内，这些技术能够有效地排除那些不可见（被遮挡或在视野之外）的对象，从而避免对它们进行资源密集型的渲染计算。剔除操作减少了渲染过程中需要处理的几何数据量，显著提高了渲染速度和响应性，尤其是在复杂场景或实时渲染应用中尤为重要。在本文的研究内容中，需要实时渲染点云，对帧率和渲染效率有着严格需求。

因此，为了减轻 Web 端渲染的压力，本文通过视锥体剔除的方法维护和渲染可见的节点，应渲染的节点在八叉树遍历步骤中确定。遍历是按照屏幕投影大小的顺序完成的。首先访问屏幕上最大的节点，然后访问第二大的节点，依此类推。投影尺寸是作为视场、到节点中心的距离、节点的边界球半径和屏幕高度的函数而获得的，如图 2 所示。

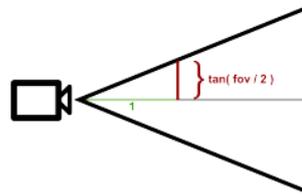


图 2 视锥体的斜率

Fig.2 Slope of the optic vertebrae

140

3.2.2 细节层次模型

细节层次模型（Level of Detail, LOD）是一种在计算机图形学中广泛应用的技术，旨在根据观察者的视角和视距动态调整对象的复杂度，以优化渲染性能和提升用户体验。该模型通过为同一对象创建不同级别的细节表示，使得当对象距离观察者较远时，使用较低细节的模型进行渲染；而当对象靠近观察者时，使用较高细节的模型。

八叉树的每一层代表不同的细节层次（LOD）。根节点包含整个点云的粗略表示，而越往树的下层，表示的点云数据越精细。在上述的空间索引的构建中和索引的分割中，已经将点云分割成了不同细节层次的结构，本文依据物体在屏幕的的投影尺寸进行计算评估，设计一个阈值来判断是否可以渲染节点。

150

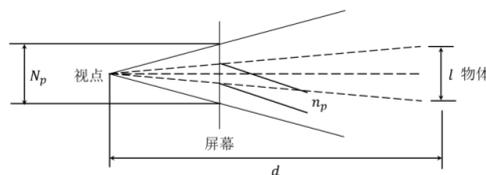


图 3 屏幕空间尺寸图

Fig.3 Schematic diagram of screen space dimensions

155

如图 3 所示，其中物体的长度为 l ，与视点的距离为 d ， θ 是垂直视野角度， N_p 是屏幕分辨率的高度， n_p 是屏幕投影的空间尺寸。求出空间尺寸的公式如下：

$$n_p = \frac{l}{2d \tan \frac{\theta}{2}} N_p$$

依据 n_p 的大小，可以设定不同 LOD 级别对应的屏幕占比阈值。当 n_p 只占据一个像素时，可以将 n_p 视为 1，反求出物体的长度 l ：

160

$$l = \frac{2d \tan \frac{\theta}{2}}{N_p}$$

如果节点的内点云的间距不大于 l ，即屏幕占比小于该阈值，则选择该节点的父节点进行渲染，即渲染更为粗糙的层次。在前文的泊松圆盘采样中，我们已经根据不同层级的索引进行了采样，保证了节点的平均间距，节省了大量的计算时间。

3.3 Web 端渲染

对点云数据进行数据构建和索引分割后，针对整个渲染的场景以及维护好可见节点列

165

170 表，采用 WebGL 与 ThreeJS 等前端图形渲染引擎技术进行渲染。场景、视图以及可见节点的渲染由 ThreeJS 处理。可见节点用一个标志 (Marked) 进行标记，然后 ThreeJS 渲染器将进行自己的遍历，以调用所有标记为可见的节点来进行绘制，同时采用 Potree 进一步实现前 endpoint 云的交互，其针对 CPU 和 GPU 渲染计算进行了优化，使得对渲染点云的实时交互更加迅速。

4 系统效果

4.1 点云可视化效果

175 浏览器调用存储在服务器数据库中处理好的点云数据，本测试使用的是真实环境采集的科研楼六楼楼道点云，在前端进行渲染，PC端可视化结果如图 4 所示。

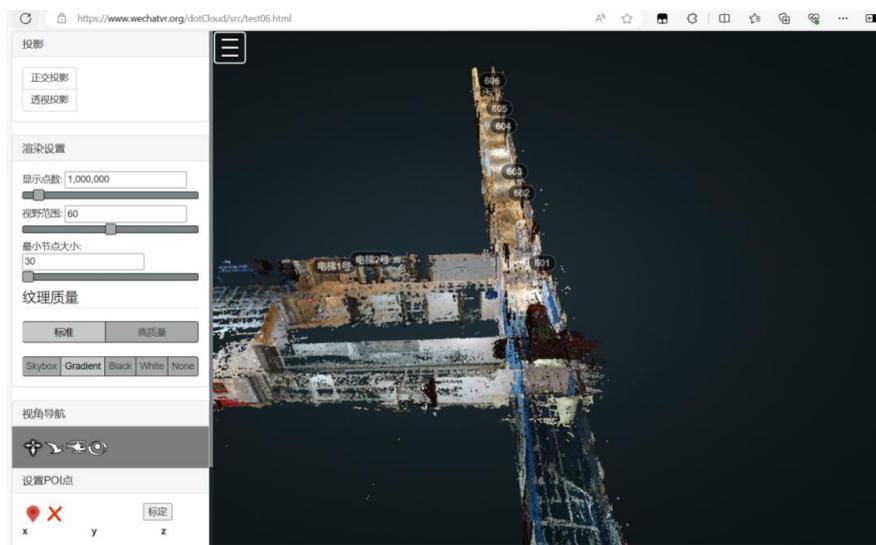


图 4 PC 端可视化结果

Fig.4 Visualize the results on the PC

180 移动端的可视化效果如图 5 所示：

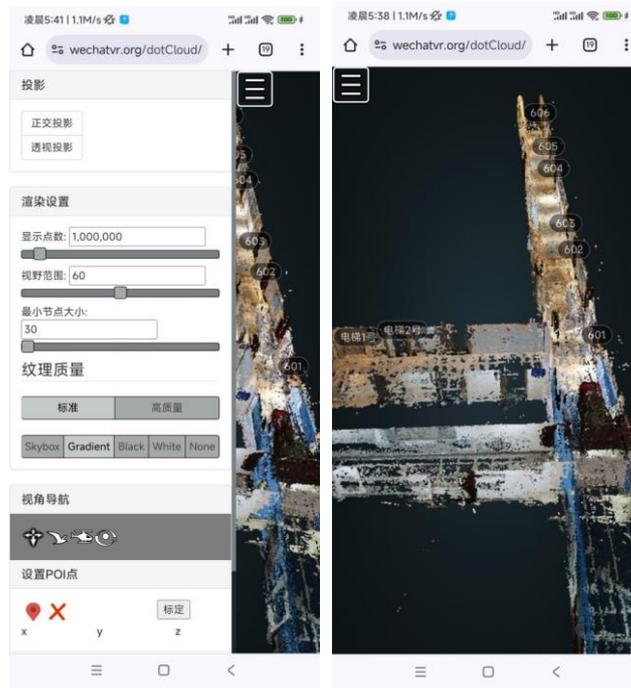
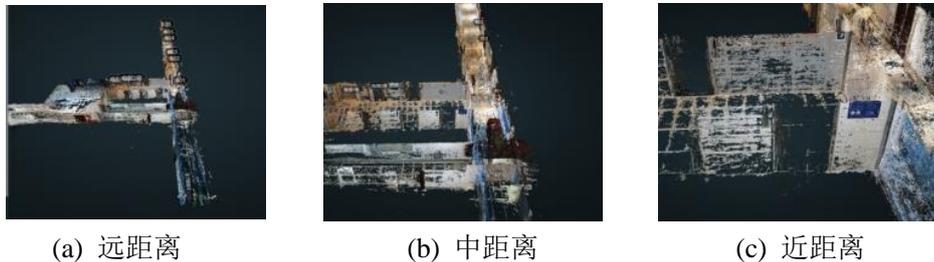


图 5 移动端可视化结果

Fig.5 Visualize the results on the mobile phone

4.2 细节层次模型

185 通过设置视野范围可以展示不同视距下点云的可视化结果，不同视距展现的细节层次不同，如图 6 所示：



(a) 远距离

(b) 中距离

(c) 近距离

图 6 不同距离的细节层次加载

Fig.6 Levels of detail at different distances

190

5 结论

195 本文提出并实现了一个面向 Web 的大规模点云渲染方法，通过在服务端处理点云，将点云按照可修改的嵌套八叉树结构建立索引，引入了空间加速的泊松盘采样算法使节点内点云间隔均匀，并按照层次结构生成不同层级的索引文件，通过视锥体剔除的可见性判断，按需加载细节层次模型的节点，结合 Web 端渲染引擎技术实现了在 Web 端快速实时的大规模点云可视化系统。经过测试和验证，该方法具有重要的实际应用价值。

[参考文献] (References)

200 [1] Gadelha M, Wang R, Maji S. Multiresolution tree networks for 3d point cloud processing[C]//Proceedings of the European Conference on Computer Vision (ECCV). 2018: 103-118.
 [2] Xiong B, Jiang W, Li D, et al. Voxel grid-based fast registration of terrestrial point cloud[J]. Remote Sensing, 2021, 13(10): 1905.

- 205 [3] 熊心一,姚宇.基于混合八叉树的动态对象碰撞检测算法[J].计算机应用,2019,39(S1):96-99.
- [4] Klovov R, Lempitsky V. Escape from cells: Deep kd-networks for the recognition of 3d point cloud models[C]//Proceedings of the IEEE international conference on computer vision. 2017: 863-872.
- [5] WebGL [EB/OL].<https://caniuse.com/webgl2>
- [6] Three.js [EB/OL].<https://www.threejs.org>
- 210 [7] Kang L, Jiang J, Wei Y, et al. Efficient randomized hierarchy construction for interactive visualization of large scale point clouds[C]//2019 IEEE Fourth International Conference on Data Science in Cyberspace (DSC). IEEE, 2019: 593-597
- [8] Takikawa T, Litalien J, Yin K, et al. Neural geometric level of detail: Real-time rendering with implicit 3d shapes[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2021: 11358-11367.