

7-15-2020

Preloading Mechanism of Large-scale Web3D Scene Based on DR Prediction

Huijuan Zhang

Tongji University, School of Software Engineering, Shanghai 201804, China;

Xinqi Guo

Tongji University, School of Software Engineering, Shanghai 201804, China;

Dongqing Wang

Tongji University, School of Software Engineering, Shanghai 201804, China;

Jinyuan Jia

Tongji University, School of Software Engineering, Shanghai 201804, China;

Follow this and additional works at: <https://dc-china-simulation.researchcommons.org/journal>



Part of the Artificial Intelligence and Robotics Commons, Computer Engineering Commons, Numerical Analysis and Scientific Computing Commons, Operations Research, Systems Engineering and Industrial Engineering Commons, and the Systems Science Commons

This Paper is brought to you for free and open access by Journal of System Simulation. It has been accepted for inclusion in Journal of System Simulation by an authorized editor of Journal of System Simulation.

Preloading Mechanism of Large-scale Web3D Scene Based on DR Prediction

Abstract

Abstract: In order to improve the loading efficiency of large-scale Web3D scene, a mechanism based on DR prediction is proposed. It combines the DR track prediction algorithm and historical path-based interest path clustering algorithm to track the field. The medium path prediction and path clustering in the traffic domain are applied to the virtual scene loading field, and a preloading mechanism of the Web3D scene is proposed. Experiment results show that the preloading mechanism can significantly improve the data transmission efficiency, optimize the loading speed of large-scale Web3D scene, and effectively improve the roaming experience of users in Web3D scene. It provides a new way to the loading mechanism.

Keywords

preloading, Web3D, dead reckoning prediction, path clustering

Recommended Citation

Zhang Huijuan, Guo Xinqi, Wang Dongqing, Jia Jinyuan. Preloading Mechanism of Large-scale Web3D Scene Based on DR Prediction[J]. Journal of System Simulation, 2020, 32(7): 1341-1348.

基于 DR 预测的大规模 Web3D 场景预加载机制

张惠娟, 郭欣琪, 王冬青, 贾金原

(同济大学软件学院, 上海 201804)

摘要: 为改善大规模 Web3D 场景的加载效率, 提出一种基于 DR(Dead reckoning) 预测的大规模 Web3D 场景预加载机制, 结合 DR 航迹预测的算法和基于历史路径的兴趣路径聚类算法, 将航迹领域中路径预测和交通领域中路径聚类应用到虚拟场景加载领域中, 提出一种 Web3D 场景的预加载机制。实验与研究表明, 通过该算法实现的预加载机制可以显著地提高数据的传输效率, 优化大规模 Web3D 场景的加载速度, 有效地提高用户在 Web3D 场景中的漫游体验, 为优化大规模 Web3D 场景的加载机制提出了新思路。

关键词: 预加载; Web3D; DR 预测; 路径聚类

中图分类号: TP391.4

文献标识码: A

文章编号: 1004-731X (2020) 07-1341-08

DOI: 10.16182/j.issn1004731x.joss.19-VR0469

Preloading Mechanism of Large-scale Web3D Scene Based on DR Prediction

Zhang Huijuan, Guo Xinqi, Wang Dongqing, Jia Jinyuan

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

Abstract: In order to improve the loading efficiency of large-scale Web3D scene, a mechanism based on DR prediction is proposed. It combines the DR track prediction algorithm and historical path-based interest path clustering algorithm to track the field. The medium path prediction and path clustering in the traffic domain are applied to the virtual scene loading field, and a preloading mechanism of the Web3D scene is proposed. Experiment results show that the preloading mechanism can significantly improve the data transmission efficiency, optimize the loading speed of large-scale Web3D scene, and effectively improve the roaming experience of users in Web3D scene. It provides a new way to the loading mechanism.

Keywords: preloading; Web3D; dead reckoning prediction; path clustering

引言

随着计算机技术与通信技术的发展, 虚拟现实技术目前已渗透到军事、教育、工程、医疗等各个领域, 并在这些领域中起重要作用^[1]。刘小

军等^[2]提出了一种基于 Web3D 的大规模 BIM 场景实时可视化的轻量化解决方案, 考虑到了网络带宽和 Web 浏览器有限资源下 BIM 数据的冗余、语义和参数化。艾子豪等^[3]对轻量级 Web3D 地铁火灾逃生路径在线规划平台实时在线关键技术进行了研究。英国最近的数字化建设部门的战略计划采用了一个新颖的开源的基于 Web3D 版本控制系统, 通过适当管理数字信息和 3D 模型, 将建筑环境中的成本和碳排放减少 20%^[4]。



收稿日期: 2019-08-30 修回日期: 2020-01-03;
基金项目: 国家自然科学基金重点项目(U19A2063);
作者简介: 张惠娟(1970-), 女, 陕西扶风, 博士, 副教授, 研究方向为移动计算, 虚拟现实等; 郭欣琪(1995-), 女, 陕西西安, 硕士生, 研究方向为 Web3D 技术。

<http://www.china-simulation.com>

• 1341 •

Web3D 是一种虚拟交互场景,属于虚拟现实的一个方向。Web3D 场景中允许用户实时漫游,每个用户都可以控制化身在 3D 场景中移动。目前,展示大规模 Web3D 场景的需求日益增长,但是目前的 Web 应用中模型过小,不能满足越来越多的在 Web 中展示大规模场景的需求。然而,由于大规模场景中的建筑物模型的数据量过大,而数据传输需要一定时间,所以浏览器总会等待很久才能接收到模型数据进行渲染,这样的机制使用户在场景中漫游时等待场景加载的时间过长、用户感受到很明显的卡顿,限制了 Web3D 的发展。所以,如何减少频繁状态更新、降低带宽消耗和网络延迟的负面影响、优化加载模型中组件的加载顺序,提高用户体验成为目前的主要研究点。因此,本文引入了诸如兴趣管理^[5]和航位推算^[6]之类的机制来缓解原始加载机制带来的带宽消耗问题。

对于兴趣管理,通常是对漫游轨迹的处理。随着越来越多的用户参与到 Web3D 场景,他们的化身在场景中漫游经过的轨迹数据可以被服务器记录形成包含海量轨迹的数据库。通常情况下,这些大规模的轨迹数据包含大量有价值的潜在信息。通过适当的轨迹数据处理方法对这些轨迹数据进行分析是许多学者研究的热点问题。聚类分析是处理历史轨迹数据中最常用的方法之一。轨迹数据的聚类分析可以将具有高度相似性的轨迹聚合到同一个聚类中。进一步分析聚类结果可以得出场景中的热点和热点路径。本文也将轨迹数据的信息结合到预加载机制中,提取场景中的兴趣路径,提高路径预测的准确度。

航位推算(DR)是一个通过其历史状态预测化身未来状态的过程。通过使用 DR 模型,服务器可以预测其未来状态,每次预测后,服务器都会向 Web 端推送预测路径上的构件数据。当预测的化身模型在特定阈值内遵循实际化身模型时,Web 端就可以马上利用提前传输过来的数据进行

渲染。如果违反了阈值,则 Web 端会向服务器更新状态,该过程通常被称为“回滚”,以便服务器更新其预测模型、重新发送数据。通常,DR 预测通过牺牲预测与实际状态的一致性来减少带宽使用。DR 已经研究了 20 多年。然而,缺乏研究 DR 与加载机制的整合的工作。实现 DR 和兴趣管理的主要潜在缺点是由于频繁的回滚和由于兴趣度计算导致的高计算开销而导致传输效率低。但是,DR 和兴趣度的良好配置可以在较低带宽使用率方面带来更多好处,而不会显著缺乏一致性或增加计算开销。

本文专注于将基于 DR 模型的路径预测应用到 Web 场景中,介绍了一种结合了兴趣度和路径预测的混合预加载方法。这里使用的 DR 预测模型与传统使用纯运动学模型估计化身运动的传统航位推算方法有所不同,新方法还考虑了环境因素和化身历史行为影响来进行推断。它利用两种技术的优点,在带宽使用和计算开销之间进行可接受的折衷。基于加载的对比实验,本文还将其性能与现有的没有基于 DR 算法的预加载机制的传统加载方式进行了比较。结果表明,新方法提供了准确的预测,提高了传输数据的效率,且具有可接受的开销。

1 研究现状

1.1 历史轨迹聚类

从历史轨迹数据中提取隐藏的有价值信息的方法有很多,其中最常见就有聚类算法。聚类算法是指根据特定条件或标准将数据集聚集成多个不同的聚类。不同聚类中的数据具有不同的相似度,同一聚类中的数据相似度较高,反之,不同聚类中数据的相似度较低^[7]。聚类分析作为数据挖掘中最常用的方法,与分类有着很大的差异。聚类分析是一种无监督学习,即在进行聚类过程前不会预先设定该如何分类,而是根据一种信息相似度进行聚类;而分类则是一个有指导的学习

过程, 通过学习训练样本中标记的数据对象, 形成表达相应数据对象和类标识的知识^[8]。目前, 很多行业已经开始使用聚类分析方法^[9]。例如, 市场分析师可以通过聚类划分客户购买模式, 然后针对不同的客户群进行不同的商业营销; 研究人员通过对不同物种间蛋白质的聚类分析, 可以发现不同物种间蛋白质的同源性。

有 3 种常见的聚类算法: 基于划分的聚类、基于层次的聚类和基于密度的聚类。基于划分的聚类的原理是首先对原始数据集进行分类, 然后通过不断迭代过程更新聚类中心和聚类中的对象, 直到满足聚类指标的要求。基于划分的聚类效果的判断标准是: 同一簇中对象之间的相似度与不同簇对象之间的差异度。K-means 算法和 K-medoids 算法是两种常见的基于划分的聚类算法。K-medoids 算法选择当前聚类中与剩余点距离之和最小的点作为新的聚类中心点, 而 K-means 算法选择当前聚类中的平均点作为新的聚类中心点。基于层次的聚类分为合并的层次聚类和分割的层次聚类。他们之间的差异在于操作过程: 合并模式的分层聚类意味着将原始数据的每个样本作为一个单独的聚类使用, 每次选择最近的两个聚类合并成一个聚类, 当所有样本合并到一个聚类或是达到预设的聚类指标时, 聚类过程停止。分层聚类分割方法是指先将所有原始数据作为一个聚类, 每次选择样本之间距离最远的区域进行分层, 当每一个样本数据都形成了一个单独的聚类或达到聚类指标的时候, 聚类过程停止。基于密度的聚类基于的是数据集中样本的局部密度差异。基于密度的聚类的原理是计算每个样本点的局部密度值, 并在特定区域中的样本密度超过预定密度阈值时继续扩展聚类。密度聚类和其他聚类算法之间的主要区别在于密度聚类可以执行非“圆簇”聚类, 可以找到任何类型的聚类且对噪声数据不太敏感。

1.2 DR 算法

DR 技术用于通过对移动实体执行外推来降低

更新消息的频率。每个服务器存储所有用户的 DR 模型(或多个模型), 其移动和表现为 DR 算法如何预测。为了保持化身的状态一致性, 有必要检查预测模型与实际模型的偏差程度。如果偏差大于某个预定阈值, 则向用户更新消息, 以更新他们预测的该化身的模型。

传统的 DR 模型基本上是 2 种类型—1 阶和 2 阶, 它们表示用于运动预测的运动方程的顺序。设 s 表示实体的预测位置, s_0 表示其在上次更新中的位置, v 表示速度, a 表示加速度, Δt 表示自上次更新以来经过的时间, 运动方程定义为:

$$s = s_0 + v\Delta t \quad (1)$$

$$s = s_0 + vt + 0.5 \times a \times \Delta t \quad (2)$$

除了传统的基于 DIS(Distributed Interactive Simulation)的 DR 方法^[10-11]之外, 文献中已经提出了许多改进的 DR 算法^[12]。通常, 这些改进的方法关注于主要的设计要求—性能, 其通过减少网络带宽消耗来测量。此外, 在设计高效 DR 算法时, 计算效率也是一个需要考虑的重要因素。

基于位置历史的航位推算(PHBDR)^[11]是传统 DR 算法的早期变体之一。它基于典型的曲线拟合方法预测实体的未来移动。该方法使用远程主机发送的 3 个最新报告位置作为外推的基础。换句话说, 它的预测与实体的速度和加速度无关。结果, 其校正分组大小小于其他 DR 方法, 因为它仅包含位置信息。另一种方法, 即自适应 DR^[13], 基于用户与远程实体的交互来调整阈值。根据利益管理的概念^[14], 实体的感兴趣区域(AOI)用于动态确定阈值。该算法为不同范围的远程实体分配不同的阈值。因此, 较不感兴趣的实体更不频繁地更新, 这减少了分组的数量。为了避免由不可预测的运动产生的额外网络消耗, 预检算方法^[10]在检测到不可预测的输入时立即发出更新数据包。它引入了一个新的阈值—拥塞角度, 来指示一个运动是否是不可预测的急转弯。通过利用此阈值, 可以减少由不可预测的移动引起的回滚次数。为了利用各种 DR 算法的优势, 基于自适应分

类器系统的航位推算^[11]依赖于—组规则(例如传感器范围)来确定在运行时采用哪种 DR 算法。一些先前的工作集中于改善 DR 算法中使用的阈值。Seamus^[10]定义了一个称为“物理—一致性—成本”的度量,它考虑了回滚造成的后果。此外,Dave Roberts^[5]提出了一种时空阈值和一种混合方案来代替不考虑时间延迟的传统空间阈值。

2 技术路线

本文提出了基于 DR 预测的大规模 Web3D 场景预加载机制。主要思想是通过节点的历史行为特征进行预测,预测出用户可能的行为轨迹,并向服务器请求通过 DR 算法预测到的路径上的 Web3D 场景的相关数据,完成预加载。

如图 1 的技术路线所示,当一个全新节点化身加入网络时,服务器会通过 DR 算法根据它的漫游信息预测漫游路径,并向数据库请求查询预测路径上的构件列表和模型数据;同时,服务器也会通过 DBSCAN 密度聚类算法提取出路径上构件的兴趣度,然后依据兴趣度对预测的构件列表进行排序;最后,服务器向 Web 端发送构件列表、模型数据等信息以完成预加载。

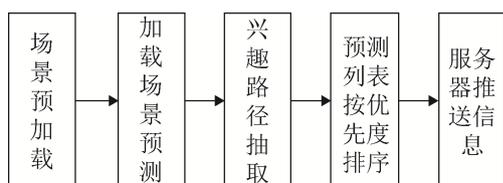


图 1 技术路线
Fig. 1 Technical route

这样的预加载机制在传输通道空闲时进行自发的路径预测,提前传输所需的构件数据,可以大大减少由于等待构件传输所带来的等待时间过长、显示卡顿的问题。

本文的预测算法主要由 3 个部分组成,即基于 DR 的路径预测、基于 DBSCAN 聚类的兴趣路径抽取、预加载机制。它们在以下的第 3 章中详细描述。

3 一种结合兴趣度与 DR 的路径预测算法

3.1 兴趣路径抽取

本文提出了一种基于行为预测的场景对等预加载机制。首先通过分析历史节点的轨迹、停留时间等行为特征来确定场景中的热门路径和热点区域。

如图 2 所示,在建立场景之前,首先为了减少服务器的计算负荷,将场景网格化,将同一网格中的历史节点聚合在一起抽象成一个有权重的聚合点,由网格中节点的数量代表网格节点的权重,这样可以为接下来的操作大大减少数据量。

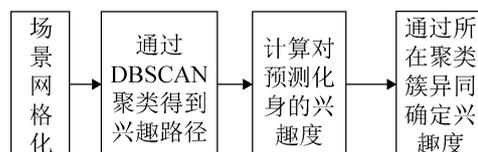


图 2 兴趣路径抽取的流程
Fig. 2 Process of interest path extraction

热点路径可以通过轨迹聚类的方法获得。基于轨迹聚类的热点路径分析方法可以直接使用原始历史轨迹数据,通过寻找相似的轨迹来反映漫游物体的兴趣模型。本文将一种典型的基于密度的聚类算法—DBSCAN(density-based space clustering of applications with noise)聚类算法的思想引入 Web 场景中。DBSCAN 算法将聚类定义为通过密度连接的最大数据集,将数据点密度高的区域划分为聚类,在有噪声的空间数据库中发现任意形状的聚类。与基于划分和层次的聚类方法不同,它不需要预先指定聚类结果个数,能够有效地处理噪声点,聚类形状的表达更为准确。所以,这里选择 DBSACN 密度聚类算法对上一步得到的网格节点进行聚类分析,即是网格节点通过历史行为轨迹、历史停留时间等行为特征信息以及上一步中定义的网格节点权重分为不同权重的新节点,再将这些节点进行 DBSCAN 密度聚类的分析,得出热点区域从而得到兴趣路径。

在得到场景兴趣路径之后, 可以分析场景中漫游化身与构件所在兴趣路径的异同, 得到场景中构件对在场景中漫游的化身的兴趣度; 然后结合得到的兴趣度, 再结合场景中构件的重用度和场景的填充度得到在场景中漫游的化身对场景中所有构件的关注度; 最后, 计算传输列表中每一个构件对预测化身的关注度并对其进行排序, 在服务器即将向 Web 端推送前优化预加载构件列表。

3.2 基于 DR 的路径预测

路径是虚拟场景中的公共元素, 它允许参与者从一个有意义的位置漫游到另一个有意义的位置, 而新加入场景的节点大概率会沿着一些已知的路径进行漫游。例如, 在高速公路, 狭窄区域和不熟悉的区域中经常可以观察到这种现象。

DR 算法的关键组成部分是运动预测模型。它的基本思想很简单: 参与者不会在虚拟空间中随机移动。如果他们在路径上移动, 他们的下一个位置很可能在同一条路径上。参与者不是在每个时间都接收实体更新, 而是使用运动学模型来预测远程实体基于其最后已知运动状态的移动。通过这种方式, 服务器可以在一段时间内不接收任何状态更新的情况下能够模拟化身的移动, 因此可以大大减少带宽消耗。此外, 服务器还为每个在场景中漫游的化身维护 DR 模型, 并不断监控 DR 模型与其相应的实际模型之间的偏差。当偏差超过预定阈值时, 服务器将更新相应实体的实际状态, 该过程通常被称为“回滚”, 以便纠正他们的 DR 模型。显然, DR 的效率很大程度上取决于 DR 模型的预测准确性。如前文所述, DR 的性能很大程度上取决于 DR 模型的预测准确性。如果经常广播校正、频繁发布回滚, 则会引起显著的带宽消耗, 并且参与者可能在客户端遇到严重的“滞后”效应。即使使用平滑算法, 存在的错觉也会受到严重影响。因此, 设计准确的预测模型是基于 DR 预测路径的预加载机制的重要要求。

所以在传输场景时需要格外注意阈值, 阈值水平会影响 DR 的性能。如果阈值设置得太高, 预测精度将比较低, 因此通过降低一致性会降低预测的准确度, 影响在场景中用户的体验。另一方面, 如果阈值设置得太低, 则会产生许多不必要的更新, 这会破坏 DR 的目的。所以为了确定合适的阈值, 后续本文在线上环境进行多次加载实验, 改变阈值水平并比较不同阈值水平下的加载效果, 最终确定最适合线上环境网络水平的阈值。

基于兴趣度的 DR 模型在传统 DR 模型上加入了场景兴趣度对路径的影响。设 P 表示实体的基于兴趣度的 DR 算法的预测位置, P_0 表示传统 DR 算法的预测位置, I 表示兴趣度影响, 运动方程定义为:

$$P = P_0 + I \quad (3)$$

图 3 展示了传统 DR 算法与基于兴趣度的 DR 算法预测的路径对比, 图中黑色为传统 DR 算法预测的路径、红色为基于兴趣度的 DR 算法预测的路径。相比传统 DR 算法, 基于兴趣度的 DR 算法预测的路径更加准确。

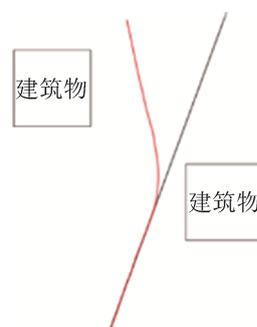


图 3 传统 DR 算法与基于兴趣度的 DR 算法预测的路径对比图

Fig. 3 Comparison diagram of predicted path between traditional DR algorithm and DR algorithm based on interest

3.3 预加载机制

如图 4 流程所示, 当用户化身进入 Web 场景后, 会向服务器发送其漫游位置; 服务器接到 Web 端发送来的坐标并记录到 Redis 数据库中; 然后, 服务器通过 DR 路径预测算法实时计算预测的漫游路径, 并向数据库查询构件列表的模型数据信

息；一旦传输通道空闲，服务器就向 web 端返回预测路径上的模型数据。

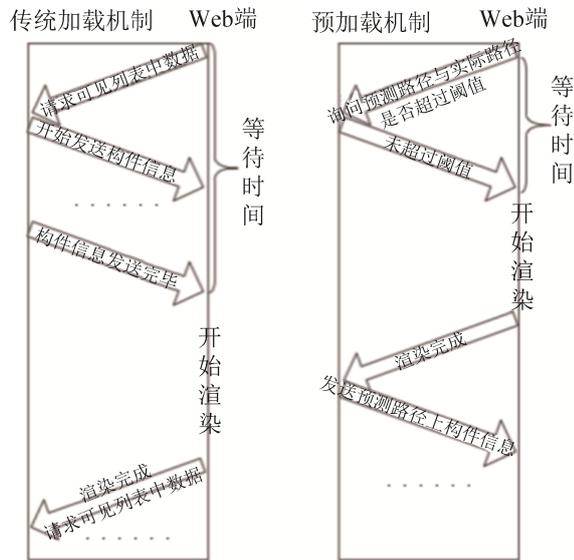


图 4 基于 DR 路径预测的预加载机制的流程图
Fig. 4 Flow chart of preloading mechanism based on DR path prediction

图 5 显示的是传统加载机制和基于 DR 路径预测的预加载机制的请求-传输-渲染时序流程对比。

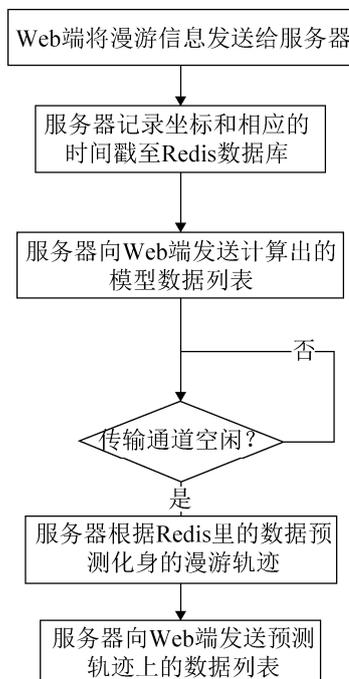


图 5 基于 DR 路径预测的预加载机制与传统加载方式对比
Fig. 5 Comparison of preloading mechanism based on DR path prediction and traditional loading method

从图 5 可以看出，在传统加载机制中，每当化身漫游到一个为渲染区域时，Web 端总要向服务器请求可视列表及其数据。每次等待的时间包括：向服务器发送请求的时间，服务器计算可视列表的时间、服务器向 Web 端传输构件数据的时间以及 Web 端的渲染时间，其中服务器向 Web 端传输构件数据的时间受网络带宽影响波动很大，若网速不理想，那么通过传统加载机制加载场景的等待时间很长，用户会感到明显的卡顿。而在基于 DR 路径预测的预加载机制中，每当化身漫游到一个未渲染区域时，化身只需向服务器发送实时漫游位置，若预测路径与实际路径之间的差距没有超过阈值，那么可以直接渲染之前空闲时服务器发送来的构件数据。每次渲染完成前的等待时间仅为向服务发送请求的时间和服务器应答的时间之和，远小于传统加载机制的等待时间。

4 实验结果与分析

为了验证预加载机制的有效性，本文使用阿里云平台，选择了共享计算型云服务器(ECS)，服务器操作系统为 Ubuntu 16.04 64 位，服务器还配置了 1 核 CPU、2 GB 内存、20 Mbps 带宽。

服务器部署了传统加载方式和基于 DR 预测的预加载机制显示的 Web 场景。服务器部分采用 nginx 和 pm2 的方式部署，同时 Web 部分代码采用 webpack 打包压缩，很大程度的压缩了数据量。

这里通过比较加载时间、预测准确度等指标来证明预加载机制的效果。加载时间是实验中最重要指标，这里比较了场景的完全加载时间和基于 DR 预测的预加载机制显示的 Web 场景的构件初始加载时间。

本实验选取了车公庙地铁模型在固定带宽下测试不同阈值模型的加载时间，车公庙地铁模型含有超过 1 000 万个三角面而且有利于模拟化身的。实验数据如图 6 所示，当阈值逐渐增加时，打包加载时间随着每次数据量变大逐渐增大，而

完全加载时间则先随着阈值增大而变大, 直到稳定在某一值。由实验结果可知完全加载时间在阈值在 800 时就稳定下来, 所以 800 是本实验的最佳阈值。后续实验将在阈值为最佳的情况下进行。

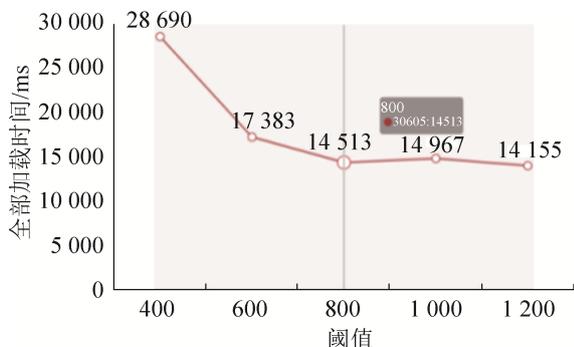


图 6 不同阈值下的加载情况
Fig. 6 Loading under different thresholds

图 7 是基于 DR 路径预测的预加载机制的传输过程, 上面的矩形块代表每次预加载传输的预测构件的数据, 矩形块的长度代表传输的时间, 其中绿色部分是 Waiting(TTFB), 指从网络请求开始直到从服务器收到第一个字节为止的时间, 蓝色部分是 Content Download, 代表 Web 端接受服务器传输数据的时间。每次一个矩形结束代表 Web 接收到了一次服务器预测的路径上构件数据。上半部分矩形块区域对应下半部分车公庙地铁站模型俯视图的漫游轨迹, 红色的垂线对应 Web 开始渲染模型的时间点, 蓝线对应渲染结束请求下一次预测路径上的数据的时间点, 从图 7 中可以看出在 Web 已经预先接收到了构件的数据, 每次漫游符合预测的情况下模型的渲染等待时间(两个矩形块之间的间隔长度)很小, 甚至可以忽略不计。

而在传统加载机制中, 每当化身漫游到一个为渲染区域时, Web 端总要向服务器请求可视列表及其数据。每次需要等待向服务发送请求的时间、服务器计算可视列表的时间、服务器向 Web 端传输构件数据的时间以及 Web 端的渲染时间之和, 传送时间基本由模型的大小确定, 从图 7 中可以看出服务器向 Web 端传输构件数据的时间(蓝绿色矩形块的长度)受网络带宽影响波动很大。所

以在网络条件优秀的情况下显示数据量很小的模型时, 速度很快; 显示效果好若网速不理想, 那么通过传统加载机制加载场景的等待时间很长, 用户会感到明显的卡顿, 比较两种不同的加载方式, 可以证明基于 DR 路径预测的预加载机制通过在传输通道空闲时自发下载预测路径上的构件信息的方式, 可以有效的减少频繁状态更新、降低带宽消耗和网络延迟的负面影响、减少加载过程中的卡顿现象、优化加载模型中组件的加载顺序, 提高用户体验。

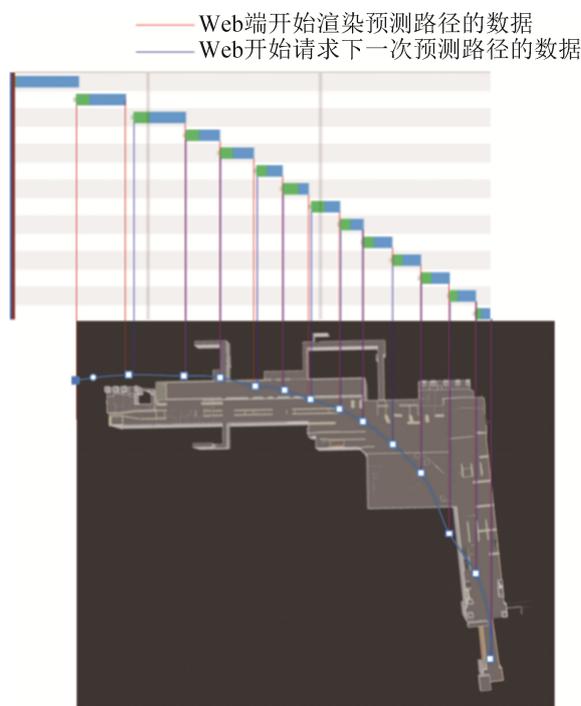


图 7 基于 DR 预加载的数据传输情况
Fig. 7 Data transmission based on DR pre-loading

5 结论

本文提出了一种在 Web 平台加载大规模场景改进方法, 它利用了两种技术: 基于 DBSCAN 密度聚类的兴趣度组网和基于 DR 的路径预测。本文进行了实验评估, 比较所提出的预加载的方法与现有的原始加载方法。总的来说, 评估结果表明预加载的方法在漫游场景中加载显示方面比原始加载方式表现更好, 同时不会产生更高的 GPU 计算开销。但是, 在预加载方法时, 必须针对服

务器带宽仔细配置打包的大小, 以使其不会导致性能不佳。

根据实验的观察, 开发者必须确定的位置准确性和带宽之间的阈值; 阈值设置太低会产生大量更新数据包, 这会破坏 DR 本身的目的, 而设置阈值太高会严重影响传输的效果, 例如未检测到的碰撞事件。在为混合方法配置阈值时, 与高阈值相比, 低阈值对带宽使用的影响更大。增加低阈值会迅速降低带宽使用率。尽管如此, 增加高阈值还会降低带宽使用率, 但不会像增加低阈值那样显着。

在未来的工作中, 将在更多的 Web 应用场景中评估所提出的混合方法, 例如多人 FPS, 角色扮演游戏等, 以适应很多不同的化身行为和行动选择。同时, 也可以继续改进预加载机制、使用内容分发网络(CDN)加载静态模型资源等方式更新。

参考文献:

- [1] 邹湘军, 孙健, 何汉武, 等. 虚拟现实技术的演变发展与展望[J]. 系统仿真学报, 2004, 16(9): 1905-1909.
Zou Xiangjun, Sun Jian, He Hanwu, et al. The Development and Prospects of Virtual Reality[J]. Journal of System Simulation, 2004, 16(9): 1905-1909.
- [2] Liu X, Xie N, Tang K, et al. Lightweighting for Web3D visualization of large-scale BIM scenes in real-time[J]. Graphical Models (S1524-0703), 2016, 88: 40-56.
- [3] Scully T, Doboš J, Sturm T, et al. 3drepo. io: building the next generation Web3D repository with AngularJS and X3DOM[C]// Proceedings of the 20th international conference on 3D web technology. New York, NY, USA: ACM, 2015: 235-243.
- [4] 艾子豪, 胡永豪, 闫丰亭, 等. 轻量级 Web3D 地铁火灾逃生在线规划关键技术[J]. 中国科学: 信息科学, 2019, 49(4): 405-421.
Ai Zihao, Hu Yonghao, Yan Fengting, et al. Key technology of lightweight Web3D online planning of metro escape[J]. Scientia Sinica (Informationis), 2019, 49(4): 405-421.
- [5] Liu E S, Theodoropoulos G K. Interest management for distributed virtual environments[J]. ACM Computing Surveys (S0360-0300), 2014, 46(4): 1-42.
- [6] Fujimoto R M. Parallel and Distributed Simulation Systems[C]// Proceeding of the 2001 Winter Simulation Conference. Arlington, VA, USA: IEEE, 2001: 147-157.
- [7] Calabrese F, Diao M, Di Lorenzo G, et al. Understanding individual mobility patterns from urban sensing data: A mobile phone trace example[J]. Transportation Research Part C: Emerging Technologies (S0968-090X), 2013, 26: 301-313.
- [8] Cheng T, Haworth J, Anbaroglu B, et al. Spatiotemporal Data Mining[M]. Berlin Heidelberg: Springer, 2014: 1173-1193.
- [9] 刘爽. 基于时空轨迹的交通数据分析与应用[D]. 成都: 电子科技大学, 2017.
Liu Shuang. Analysis and Application of Traffic Data Based on Time-space Trajectory[D]. Chengdu: University of Electronic Science and Technology of China, 2017.
- [10] Mcloone S C, Walsh P J, Ward T E. An Enhanced Dead Reckoning Model for Physics-Aware Multiplayer Computer Games[C]// IEEE/ACM International Symposium on Distributed Simulation & Real Time Applications. Dublin: IEEE, 2012: 111-117.
- [11] Singhal S K, Cheriton D R. Using a position history-based protocol for distributed object visualization[R]. Stanford, CA, USA: STANFORD UNIV CA DEPT OF COMPUTER SCIENCE, 1994: 1-25.
- [12] Cai W, Lee F B S, Chen L. An auto-adaptive dead reckoning algorithm for distributed interactive simulation[C]// Workshop on Parallel & Distributed Simulation. Atlanta, GA, USA: IEEE, 1999: 82-89.
- [13] Zhang X, Gracanin D, Duncan T P. Evaluation of a pre-reckoning algorithm for distributed virtual environments[C]// Parallel & Distributed Systems, Tenth International Conference. Newport Beach, CA, USA: IEEE Computer Society, 2004: 445-452.
- [14] Toriki S, Torguet P, Sanza C. Adaptive classifier system-based dead reckoning[C]// Proceedings of the 13th Eurographics conference on Virtual Environments. Goslar, DEU: Eurographics Association, 2007: 101-108.