

Web3D 家居素材库的轻量化技术研究

赵双燕, 贾金原, 周文

(同济大学 软件学院, 上海 201804)

摘要: 随着 Web3D 技术的不断发展, 使得人们对基于 Web3D 家居模型的互联网在线即时可视化服务需求激增. 利用轻量化关键技术, 方便快捷地解决了大规模 WebVR 家居模型的处理、传输、重建、实时渲染等问题. 提出了基于 WebVR 的轻量级三维家居素材库在线可视化服务, 详细介绍了该系统对家居模型轻量化的实现过程中一些关键技术, 包括轻量级渐进式网格处理、细粒化处理、基于视点传输等. 结果表明, 新的轻量化框架能够较好地处理模型, 提升了传输速度.

关键词: Web3D; WebVR 家居; 轻量级家居建模; 增量式传输; 轻量级 Web3D 渲染

中图分类号: TP391.9 文献标志码: A doi:10.13705/j.issn.1671-6833.2019.01.005

0 引言

互联网+时代已经到来, 而 2017 年正是 VR+元年, Web 在线服务越来越火热. 以往互联网一直以 2D 的形态存在, 但是随着 3D 技术的不断成熟, 在未来的 5 年时间里, 互联网应用将会广泛地以 3D 的方式呈现给用户, 家居模型库作为应用之一将在互联网上以 3D 的形式来让用户体验, 这种体验的真实震撼程度要远超以往的 2D 环境. Web3D 应用需求的急速激增, 以及人们对 3D 场景质量要求的不断提高, 促使 2D 的展示方式在不远的将来被 3D 立体方式所代替, 这将成为一个必然趋势, 但这必须以一种操作简便、成本低、真实感强的 3D 图像建模技术为基础. 市场需求的变化趋势将引领技术的发展方向, 因此, 基于 3D 家居模型的互联网服务也发生了日新月异的变化.

随着虚拟现实和 Web3D 技术的发展, 基于网页的虚拟现实(WebVR)^[1-2]进入了我们的生活, 通过 WebVR 在网页上在线展示大规模虚拟家居场景已越来越受到人们的青睐. 相关方面的研究也相继出现, 但受 WebVR 传输、下载和渲染大规模虚拟家居场景发展缓慢的阻碍, 我们仍面临如下问题:

(1) 对于现存的可用网络带宽, 要实现实时

传输, 需要从服务器端下载的 WebVR 数据总量太大.

(2) 基于计算机图形学的 WebVR 渐进式传输仍然不能保证流畅地浏览 WebVR 家居场景, 尤其当场景较大或多个场景时, 延迟问题将会比较突出.

(3) 尽管大规模的 WebVR 家居场景已经被成功地下载了, 由于受浏览器的弱计算和有限的缓存能力限制, 在浏览器端实时地渲染它们, 这仍然是一个难题.

因此, 如何能在较少影响用户体验质量的前提下, 采用合适的方法, 降低浏览器和服务器端所处理的数据量就显得尤为重要. 为克服上述问题, 笔者拟通过场景中三维模型轻量化处理与流式化传输技术创新、细粒度处理、结合基于谷歌 Draco 压缩传输方法一起来解决所面临的挑战. 笔者在三维模型轻量化处理与流式化传输方面提出了一些新的想法, 鉴于单一的体素特征描述符对于局部特征的提取存在缺失, 加入了新的特征描述符及机器学习作为辅助, 同时对于渐进式网格(progressive meshes, PM)存在的顶点塌陷问题, 提出保留部件边缘约束的方法来保证对曲率的保持, 这样既保证了流式化处理的时间和效率, 又保证了模型的处理效果.

收稿日期:2018-08-11; 修订日期:2018-09-23

基金项目: 中央高校基本科研业务费——学科交叉类项目重点项目(20153797)

通信作者: 贾金原(1963—), 男, 山东乐陵人, 同济大学教授, 博士, 主要从事轻量级 3D 建模、虚拟现实及 Web3D 等方面的研究, E-mail: jyjia@tongji.edu.cn.

1 三维模型的轻量化与流式化

渐进式流式化是一种基于渐进式网格的模型编码方法. 渐进式网格最早由 Hoppe^[3] 提出, 在该方法中将原始的场景模型表示成一个“基网格 (base mesh)”和一系列的“网格增量”. 在场景加载之初, 可直接加载数据量相对较少的、精细度比较低的“基网格”来代替整个场景模型. 随着时间的推移, 将后续接收到的“网格增量”依次合并到当前的“基网格”中, 达到场景模型逐渐精细化为原始场景模型的目的.

在进行渐进式编码时, 算法首先采用“边折叠”的方式对场景模型进行简化处理, 并将整个模型的几何、拓扑等信息的变化进行编码形成编码流. 简化过程在满足终止条件时结束, 此时简化后的模型就表示“基网络”, 简化过程中的每一步形成的编码即为“网格增量”. 在进行解码时, 算法通过“点分裂”的方式将这些“网格增量”逆序合并到“基网格”中, 以恢复原始模型.

然而, 由于渐进式编码压缩本身的特点, 在将其应用于 WebVR 系统中时还需要综合考虑诸如压缩模型本身的几何特征和传输时的丢包处理以及频繁的“网格增量”请求导致服务器的负载加重等问题, 在后续的发展过程中出现了很多改进的新方法. 例如在大规模 3D 模型中寻找重复性组件的想法, 该想法最早是在文献[4]中提出的. Cai 等^[5] 采用姿态单位化的方法来查找更多的重复性三维组件, 以此来获得更高的压缩率. Wen 等^[6] 提出的基于体素化的查重方法, 相比较其他采用渐进式网格的非重复三维组件方法, 能获得更高的查重率. Englert 等^[7] 提出了一种轻量化渐进式网格加轻量化渐进式材质的框架. Liu 等^[8] 改进文献[6]的方法, 通过语义轻量化将大规模 WebVR 分成一些细粒度块的方法来适应网络传输.

笔者在借鉴前人研究经验的基础上, 对三维模型的轻量化与流式化作出了一些改进, 以期更好地适应当前 WebVR 家居素材库发展的需要.

2 关键技术和成果

2.1 轻量级渐进式网格 (lightweight progressive meshes, LPM) 处理

2.1.1 LM (lightweight meshes) 处理

(1) 预处理. 对输入的三维模型首先要进行预处理, 基于对文献[3]中提及的边折叠、点分裂

的 LOD (level of detail) 进行优化, 同时依据三角形网格的连接性对模型进行预处理, 经过预处理后生成模型的基网格, 对所有部件从模型中分离作预处理.

(2) 姿态对齐. 温来祥^[9] 提出的改进的主成分分析 (principal component analysis, PCA) 方法来寻找模型的 3 个主轴, 由于 PCA 3 个主轴方向问题, 提高了姿态单位化带来的姿态对齐效果, 有效区别顶点相同而拓扑不同的模型, 使得姿态对齐的精确度有所提高. 在对模型进行姿态对齐时, 利用启发式的遗传算法搜索最优解, 解决了两模型间的姿态对齐问题. 并考虑了旋转、平移、缩放的调整, 提高了对三维模型进行姿态对齐的时间效率. 通过姿态对齐, 使得三维模型处于同一参考坐标系下, 所提取的模型特征具有几何不变性.

(3) 特征提取. 温来祥^[9] 给出了一种体素特征描述符 (voxel shape descriptor, VSD) 来免疫 PCA 的退化问题和模型的三角化问题. VSD 使用与三维模型表面轮廓相交的体素作为模型特征, 这些体素称为表面体素, 它可以用位序列保存, 当利用位操作时可以提高匹配效率, 但仅采用 VSD 特征进行模型特征提取难免不够全面, 因此笔者提出在 VSD 基础上采用综合特征来提高模型匹配质量和效率, 并加入多个特征如草图、骨架特征、二维光场描述符等, 在此基础上加入机器学习来提高模型部件的匹配精确度. 首先采用 SIFT (scale-invariant feature transform)^[10] 算法提取出局部区域多种特征, 借鉴文献[11]使用的 BoF (bag of features) 方法将所有提取出的特征放在一起, 对这些特征向量做聚类分析, 将质心作为类别的代表. 将每个模型的特征在它对应的质心上做直方图, 依据欧氏距离进行类别划分, 采用关键点特征向量的欧氏距离来作为两个草图中关键点的相似性判别度量, 在此基础上使用分类器就可以很容易进行分类. 在聚类过程中, 采用高斯混合模型方法, 对训练的特征库进行多次迭代来对参数进行估计, 并使用支撑向量机分类. 借鉴文献[11]中的基于视点的匹配方法来进行模型的匹配, 引入机器学习的方法后, 模型的匹配度得到了提升.

(4) 去重、构造轻量级场景图 (lightweight scene graph, LSG)^[9] 结构. 依据(3)中的模型部件的特征描述符, 对各部件进行分类, 形成相应的集合, 对每个集合中的部件进行去重, 形成基于部件的表示方式的集合. 该集合的组织结构形式为轻量级场景图 LSG, 它作为一种有向无环图, 具有层

次和部件重用的双重特性,去掉冗余数据后,原三维模型的数据量很大程度上减少了,并且没有对精度产生影响.

2.1.2 PM 处理

选择 PM 流式化处理,对轻量化预处理后的非重复性部件或单元构件,判断外形是不是曲面,只有曲面才进行 PM 流式化处理,简单形体无需 PM 流式化处理.

对经过去重处理及曲面判断的部件库进行 PM 处理,目前的 PM 是边折叠和点分裂的方法,温来祥^[9]提出的是基于二次误差矩阵 (quadric error metric, QEM) 的简化方法来生成 PM,该方法可以生成高质量的粗糙模型,将每个唯一部件生成 PM,并用 LSG 来组织,生成了模型的 LPM. 但该方法不能使唯一部件的边缘得到很好的保留,内部三角面的连接效果也不好. 因此笔者提出了改进的 PM 方法:在 PM 方法基础上对部件边缘保留约束,实现无错位,最终使得计算顶点的时间不受影响,边的曲率尽可能小. 该改进的 PM 方法借鉴文献[12]的方法提出了一种新的三角网格简化算法,这种基于全局的简化降低了三角面片的数量而又尽可能地保留了网格表面的整体外观效果,使部件的边缘得到很好的保留. 该简化算法的实现不仅考虑到表面网格、预期停止点预测,还将大量的额外参数也考虑在内. 这些参数控制和监控简化过程与文献[9]使用的边重叠、点分裂的方法不同,这里提出了半边重叠的概念,半边重叠的操作是通过移动实现的,包含一个顶点和两个边,它将保留的顶点移动到一个新的位置. 具体三角网格的简化实现过程如图 1 所示.

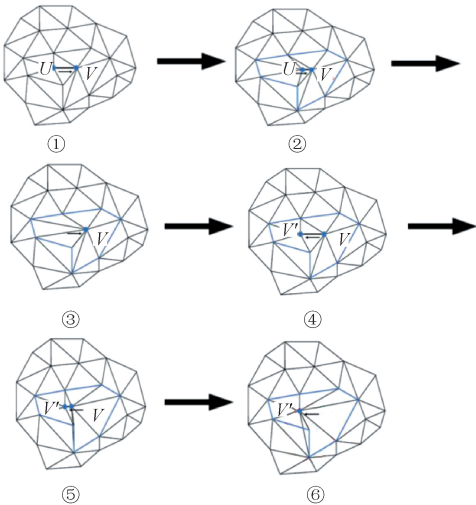


图 1 三角网格简化过程

Fig.1 Process of triangular mesh simplification

经过改进的 PM 处理后,在基网格的基础上生成原始模型的渐进式重建信息. 经过 LM 和改进的 PM 处理后,三维模型输出自定义的流式化传输格式. 具体 LPM 处理流程如图 2 所示.

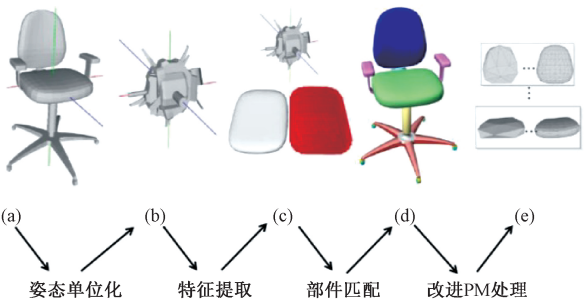


图 2 LPM 处理流程

Fig.2 LPM processing flow

LPM 处理流程:(a→b)LM 预处理,姿态单位化,将部件变换到同一坐标系中;(b→c)对部件进行多特征提取;(c→d)依据特征描述符进行分类、去重,构造 LSG 组织结构;(d→e)改进的 PM 处理.

2.2 流式化传输与实时渲染

经过 LPM 处理的模型存储于服务器端,当客户发出请求后,客户端接收来自于服务器端的基网格,进而接收网格增量,最终展现原模型. 客户端对接收到的数据进行解析,并存储在 cache 中,依据最近 (least recently used) 算法. 具体实现过程采用 WebGL 和 JS.

2.2.1 细粒度预处理

当用户浏览大规模的 3D 场景时,需要在 Web 端实时渲染这些场景,细粒度预处理的流程是将大规模 3D 场景分解成一些子空间,当漫游进入这个子空间时,作为一个下载和渲染的独立单元,每个子空间是个相对自我封闭的空间. Web 浏览器根据当前视点的位置和方位仅下载对应的子空间. 在通过室内场景细粒度化后,整个构筑物的室内数据又分为多个子空间,而对于每个子空间,则以子空间内部的构件为单位,将该子空间作为场景进行索引构建,并将构建完成的索引信息作为该子空间的组织方式. 对经过改进的 PM 处理后的模型,原始模型被处理成含有较少数据量的基网格和增量网格两部分. 在进行场景漫游时,首先将子空间中的构件处理成基网格数据,并在后续的漫游过程中,根据视点位置,增加相应的增量网格来实时调整该模型的精度,使得在整个漫游过程中,不仅不影响用户体验质量,而且能做到增量数据

的按需处理.例如,对于多层多间居室包含许多个房间,这些房间不能被浏览器一下子渲染出来,而是被分割成细粒化的一些子空间.每个子空间仅包含一定数量的组件,因其相对的空间封闭,它们可以被实时渲染.

2.2.2 LPM 数据的压缩传输

笔者采用谷歌开发的开源压缩工具 Draco 来进行 LPM 数据的压缩、传输.谷歌开发的 Draco 是关于点云的编码与压缩的源码,Draco 由谷歌 Chrome 媒体团队设计,旨在大幅加速 3D 数据的编码、传输和解码. Draco 可以被用来压缩 mesh 和点云数据.它还支持压缩点 (compressing points)、连接信息、纹理协调、颜色信息、法线 (normals) 以及其他与几何相关的通用属性.谷歌官方发布的 Draco Mesh 文件压缩率大幅优于 ZIP. Draco 的算法,它既支持有损模式,也支持无损模式.笔者采用 Draco 的无损模式,保证压缩与传输过程中不丢失数据.根据谷歌官方公布数据,Draco 有较好的压缩率,对于较小的模型,压缩效果提高 15%,法线的压缩效果提高 40%.

2.2.3 Web 前端的数据解析

对于收到的基网格数据,采用 direct mapping 的方式直接创建 vertex buffer (VB)、index buffer (IB),并将数据映射到 WebGL 端,提升渲染数据的准备效率,减少准备时间.增量网格数据采用最小邻近插入的方法,直接更新到 VB、IB 中,并更新局部缓冲区,实现增量渲染.

2.2.4 依据视点部件的优先级进行传输

对于客户端请求的数据,依据当前的观察者视角,对数据处理的优先级进行排序,优先进入客户视野的模型数据优先传输.根据视线方向与视距来确定初始传输哪些部件以及传输优先级与分辨率,然后根据当前网络带宽决定适应性打包处理.不同的物体对人眼的视觉贡献度不同,由于与视点的距离以及物体自身的体积等原因,离视点较近、体积较大且不易被遮挡的物体总是更容易被观察到.因此,从构件与视点的距离和构件本身的特征出发,根据视觉贡献度得到构件实体的优先级,优先级较高的构件实体先于其他构件实体被装载和渲染.一个包被看作是一个网络传输单元.

2.2.5 高效网页级渲染结构

随着每个子空间模型的组件数量的增多,使所有的加载时间变长,从而用户的等待时间

也会相对变长.笔者实行了一种高效网页级渲染结构,这种结构利用边加载边渲染的高效率以及轻量级结构带来的低数据提高传输率,这种多线程模式降低了用户的延迟,改善了传统的下载-渲染策略会占用用户很长的视觉等待时间的情况.一旦用户下载数据,它会分析数据并传递至渲染线程在网页端显示,这样会降低用户的视觉等待时间.

3 实验部分

开发语言:C++\JS\WebGL;

测试用例:普林斯顿 PSB 模型库中部分种类模型及典型模型如 CAD 汽车、服装商品、建筑物房间、家居平台中常规家具模型;

测试环境:笔记本电脑两台,服务器端为 Windows 7, Mem 为 8 G, CPU 为 Intel Xeon 2.39 G;客户端为 Windows 7, Mem 为 8 G, CPU 为 Intel i7-4700MQ,带宽为 5 MB/s.

3.1 三维模型轻量化与流式化实验

实验前需对测试模型进行格式统一,转换成 LPM 可以接受的 FBX、OBJ 格式,进而再对测试模型进行处理.

将本文的轻量化方法与文献[9]的方法进行比较,通过考察识别重复模型部件的能力和数据的压缩率两个指标,可以发现笔者的方法明显优于文献[9]的方法,具体比较数据如图 3 和表 1 所示.

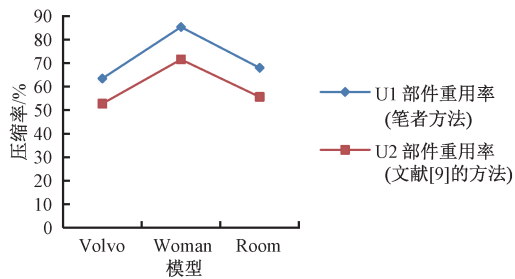


图 3 部件重用率比较

Fig. 3 Comparison of component reuse rate

表 1 对 PSB 模型库中模型进行 LPM 处理后的关键数据

Tab.1 Critical data of PSB models after LPM processing

模型	模型三角面数	处理时间/s	处理压缩率/%	部件重用率/%
Volvo	754 234	29.0	69.83	63.47
Woman	62 342	7.6	89.40	85.30
Room	292 743	13.0	78.35	68.00
PSB 部分模型结果拟合	64 803	7.9	86.73	75.47

从数据中可以看出,LPM 算法对于模型的可扩展性支持较高,从小规模的模型到大规模的模型均可以进行较好的处理.算法对模型的处理时间跟模型的规模有相关性,处理结果中的压缩率与部件重用率也有相关性,如果模型中的部件重用率较高,那么对应的压缩效果也越好,反之对应的压缩效果下降.这主要是由于在 PM 处理过程中需要存储重建时的信息,

因而会含有部分冗余的信息,若部件的重用率较高则可在重用的部件上减少重建信息的数量,因而会有较好的压缩率.从模型相关性上可以看出,对于机械、建筑之类对称性或模块化较明显的模型或资源上,部件的重用率会更高.

3.2 数据压缩与网络传输性能测试

Draco 压缩测试数据处理流程如图 4 所示.

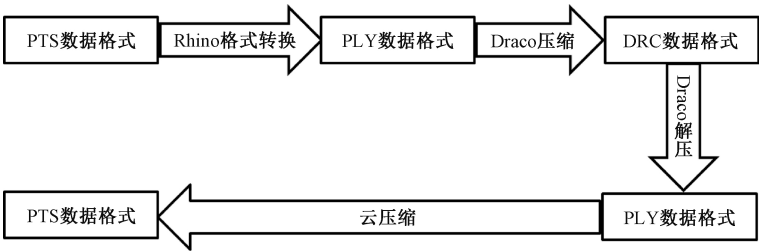


图 4 Draco 压缩测试数据处理流程

Fig.4 Compressed test data processing flow

- (1) Draco 支持的数据源为 .ply 和 .obj 格式.对于 PTS 数据格式需要先使用 Rhino 转换为 .ply 格式,然后在 Draco 中进行压缩处理;
- (2) Draco 压缩之后的格式为 .drc,在使用时需要对其 .drc 文件进行解压;
- (3) Draco 解压缩出来的格式为 .ply,不能直接当成点云处理的对象来使用,需要使用 .ply 再转化为 PTS 格式.

小的模型,由于传输并不会成为瓶颈,因而对应的传输比提升不是太明显.使用压缩进行数据传输需要在压缩比与传输比之间达成一个平衡,避免过于依赖传输或过于依赖解压.

3.3 前端解析与渲染效果

笔者对室内家居场景进行了 Web 端解析与渲染,具体效果如图 5 所示.



图 5 家居场景渲染效果图

Fig.5 Rendering of house scene

利用 Draco 对 PSB 模型进行测试后发现,该方法在解压缩和传输时间上效率很高,节省时间效率约为 11% ~ 22%.具体的传输率和传输时间等数据如表 2 和表 3 所示.

表 2 利用 Draco 对模型处理的压缩率

Tab.2 Compressed rate of models processing with Draco			
PTS 源文件 大小/MB	源 PLY 文件 大小/MB	由 PLY 转换得 到的 PTS 文件 大小/MB	源 PTS 到转 换后的 PTS 压缩率/%
241	192	215	89.21

表 3 利用 Draco 对模型处理结果

Tab.3 Result of models processing with Draco					
模型	三角 面数/个	压缩传 输时间/s	解压缩 时间/s	直接传输 时间/s	处理时 间/s
Volvo	754 234	1.12	1.30	2.73	0.31
Woman	62 342	0.30	0.95	1.40	0.15
Room	292 743	0.73	0.70	1.90	0.47
Dragon	871 414	1.20	1.53	3.50	0.77

从数据中可以看出,对于较大规模的模型,压缩之后传输可以得到较好的性能提升,而对于较

4 结论

笔者基于 WebVR 和成熟的轻量化技术对大规模的三维家居模型的在线可视化服务进行了探索.笔者提出的新的轻量化框架能够较好地处理模型,在保证最终渲染质量无损失的情况下,提升了预处理与传输的速度,以便在 Web 客户端更好地实现可视化.但是,在笔者提出的方法中仍旧存在一些短板,以下几个方面是我们接下来努力的方向:(1)增加算法的模型适配性,现在对汽车等构建重复度较高的模型类型效果会比较好一些,需要提升 LM 的效果;(2)建立基于视点的预测加载机制,提升加载效率;(3)基于视点的操作在云端来进行.

参考文献:

[1] BARSOUM E, KUESTER F. WebVR: an interactive Web browser for virtual environments [C]//Stereo-scopic Displays and Virtual Reality Systems XII. Bellingham: Proc of SPIE, 2005: 540 – 547.

[2] YIN T F, HAN Y, CHEN Y, et al. WebVR-web virtual reality engine based on P2P network[J]. Journal of networks, 2011, 6(7): 990 – 998.

[3] HOPPE H. Progressive meshes[C]//Proceedings of the 25th Conference on Computer Graphics and Interactive Techniques. New York: ACM, 1996:99 – 108.

[4] SHIKHARE D, BHAKAR S, MUDUR S P. Compression of large 3D engineering models using automatic discovery of repeating geometric features [C]//Proceedings of the Vision Modeling and Visualization Conference. India: Juhu, Mumbai, 2001:233 – 240.

[5] CAI K, WANG W, CHEN Z, et al. Exploiting repeated patterns for efficient compression of massive models [C]//International Conference on Virtual Reality Continuum and ITS Applications in Industry. New York: ACM, 2009:145 – 150.

[6] WEN L X, XIE N, JIA J Y. Client-Driven Strategy of Large-Scale Scene Streaming[C]//Proceedings of International Conference MultiMedia Modeling. Miami: Springer, 2016:93 – 103.

[7] ENGLERT M, JUNG Y, KLOMANN M, et al. A streaming framework for instant 3D rendering and interaction[C]//Proceedings of the 21st ACM Symposium on Virtual Reality Software and Technology. Beijing: ACM, 2015:192 – 192.

[8] LIU X J, XIE N, TANG K, et al. Lightweight for Web3D visualization of large-scale BIM scenes in real-time[J]. Graphical models, 2016, 88:40 – 56.

[9] 温来祥. 大规模三维场景的轻量化与流式化传输 [D]. 上海: 同济大学软件学院, 2015.

[10] LOWE D G. Distinctive image features from scale-Invariant keypoints[J]. International journal of computer vision, 2004, 60(2): 91 – 110.

[11] EITZ M, RICHTER R, BOUBEKEUR T, et al. Sketch-based shape retrieval[J]. Acm transactions on graphics, 2012, 31(4): 1 – 10.

[12] CACCIOLA F. CGAL 4.10-triangulated surface mesh-simplification[EB/OL]. [2018-04-03]. https://doc.cgal.org/latest/Surface_mesh_simplification/index.html#title0

The Research of Web3D Household Repository Lightweight Technology

ZHAO Shuangyan, JIA Jinyuan, ZHOU Wen

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

Abstract: With the development of Web3D technology, Web service demand for 3D household models on line application surged. Taking advantage of lightweight technology and so on, the processing, transmission, re-building, and real-time rendering problems were solved for large scale WebVR household models. Based on WebVR, this paper put forward the online visual service of Web3D lightweight household models repository was introduced. The detailed process of the lightweight household models, and some related key technologies, including lightweight progressive meshes processing, fine-grained preprocessing, viewpoint based transforming and so on were also introduced. The results showed that lightweight technology could process the household models rapidly.

Key words: Web3D; WebVR household; lightweight household modeling; incremental transmission; lightweight Web3D rendering