



(12) 发明专利

(10) 授权公告号 CN 112270756 B

(45) 授权公告日 2022.03.25

(21) 申请号 202011327455.6

G06T 1/20 (2006.01)

(22) 申请日 2020.11.24

审查员 穆滢

(65) 同一申请的已公布的文献号
申请公布号 CN 112270756 A

(43) 申请公布日 2021.01.26

(73) 专利权人 山东汇颐信息技术有限公司
地址 250000 山东省济南市历下区草山岭
南路801号万科中心B座7层

(72) 发明人 郭翔铭 刘宁

(74) 专利代理机构 青岛致嘉知识产权代理事务
所(普通合伙) 37236
代理人 高维波

(51) Int. Cl.

G06T 17/10 (2006.01)

G06T 15/00 (2011.01)

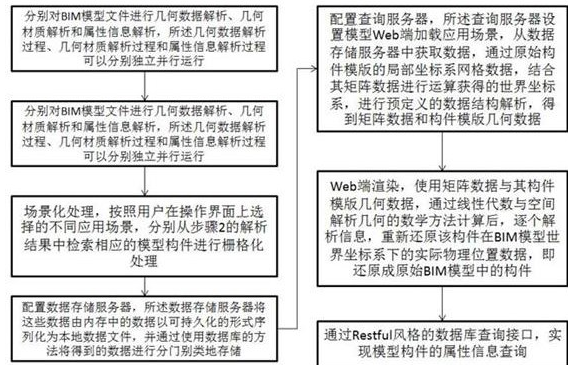
权利要求书3页 说明书7页 附图1页

(54) 发明名称

一种应用于BIM模型文件的数据渲染方法

(57) 摘要

本发明通过对BIM模型文件进行轻量化处理的基础上,为进一步减少模型数据量,提高数据传输、加载和渲染的速度,对模型按工作场景进一步拆解,按细分场景组织模型构件文件,满足工程建设管理的高性能、高场景化的业务需求;采用多重LOD用不同级别的几何体来表示物体,距离越远加载的模型越粗糙,距离越近加载的模型越精细,从而在不影响视觉效果的前提下提高显示效率并降低存储;批量绘制,提升渲染流畅度;绘制调用非常耗费CPU、并且通常会造成GPU时间闲置。本发明方法按应用场景加载和渲染模型构件,具备降低服务器硬件配置要求,减少IO操作频率,缩短前端用户的等待时间,提高用户体验等优点。



1. 一种应用于BIM模型文件的数据渲染方法,其特征在于,包括步骤:

步骤1: 分别对BIM模型文件进行几何数据解析、几何材质解析和属性信息解析,所述几何数据解析过程、几何材质解析过程和属性信息解析过程可以分别独立并行运行;

步骤2: 解析数据,使用线性代数与空间解析几何的数学方法,对步骤一中同类型的构件的几何数据和材质数据进行分析,分析出使用可复用技术成型的构件模版数据和在BIM模型中的世界坐标系矩阵数据;

步骤3: 场景化处理,按照用户在操作界面上选择的不同应用场景,分别从步骤2的解析结果中检索相应的模型构件进行栅格化处理;

步骤4: 配置数据存储服务器,所述数据存储服务器将这些数据由内存中的数据以可持久化的形式序列化为本地数据文件,并通过使用数据库的方法将得到的数据进行分门别类地存储;

步骤5: 配置查询服务器,所述查询服务器设置模型Web端加载应用场景,从数据存储服务器中获取数据,通过原始构件模版的局部坐标系网格数据,结合其矩阵数据进行运算获得的世界坐标系,进行预定义的数据结构解析,得到矩阵数据和构件模版几何数据;

步骤6: Web端渲染,使用矩阵数据与其构件模版几何数据,通过线性代数与空间解析几何的数学方法计算后,逐个解析信息,重新还原该构件在BIM模型世界坐标系下的实际物理位置数据,即还原成原始BIM模型中的构件;

步骤7: 通过Restful风格的数据库查询接口,实现模型构件的属性信息查询;

所述步骤1: 分别对BIM模型文件进行几何数据解析、几何材质解析和属性信息解析,所述几何数据解析过程、几何材质解析过程和属性信息解析过程可以分别独立并行运行,还包括:

步骤101: 输入三维几何结构和二维纹理构成的三维建筑模型标识符,进行I/O操作,主要是指访问地图标识符库,从地图标识符库中调用、读取所需标识符的图形、纹理和材质描述数据;

步骤102: 对输入模型进行计算和分析,提取出基于点和线的三维模型表示,其中,点图元的提取通过递归化的多重抖动采样和蓝噪声采样完成,线模型则通过对输入的几何结构进行分析提取,并根据图像空间的颜色聚类关系进行分割标识符图形构造: 将地图标识符的描述信息中moveto、lineto、arcto、curveto、closepath绘制命令和坐标序列数据,构造成能够转换为光栅化技术可以处理的图形结构以及图形的样式配置;

步骤103: 标识符图形变换: 包括缩放、旋转、平移坐标变换,笔画、填充图形变换以及线宽、线型、线端头、线拐角样式处理,将图形构造操作产生的图形结构经图形变换操作生成与之对应的封闭多边形结构。

2. 如权利要求1所述的一种应用于BIM模型文件的数据渲染方法,其特征在于

所述步骤2: 解析数据,使用线性代数与空间解析几何的数学方法,对步骤一中同类型的构件的几何数据和材质数据进行分析,分析出使用可复用技术成型的构件模版数据和在BIM模型中的世界坐标系矩阵数据,具体包括:

步骤201: 通过标识符配置进行点状数据与地图标识符的关联,所有的点状数据均采用相同的点状标识符;根据标识符配置的地图标识符标识sid,从标识符库中读取该点状标识符,并构建相应的标识符缓存;

步骤202:将地图标识符标识sid作为键索引,标识符缓存SymbolCache作为键值传入到标识符缓存列表中;这样通过循环遍历点状数据,获取点状数据的坐标位置;依据坐标位置参数构建绘制矩阵,并从标识符缓存列表中获取相应的标识符缓存;通过矩阵乘法实现标识符缓存数据的坐标变换,将标识符缓存的顶点数据变换到对应点状数据的坐标位置,向GPU中提交渲染数据以及绘制命令便可进行该点状数据的标识符化渲染。

3.如权利要求1所述的一种应用于BIM模型文件的数据渲染方法,其特征在于所述步骤3:场景化处理,按照用户在操作界面上选择的不同应用场景,分别从步骤2的解析结果中检索相应的模型构件进行栅格化处理,具体包括:

步骤301:计算每个建筑的表面积;然后,通过应用透视投影变换和视口变换计算该表面积对应的屏幕空间投影面积,确定无缝地表示该建筑对象所需要的点和线场景数量;

步骤302:根据所需的点、线场景数量从点、线场景队列中选择对应的渲染场景进行渲染,场景离散:将封闭多边形进行离散的过程;针对二维数据,采用扫描线算法、将封闭多边形离散为水平方向分布的线段,针对三维数据,采用剖分算法将封闭多边形分解为多个三角形场景,其中,扫描线段和三角形场景都是着色的基本单元;

步骤303:由中央处理器CPU向显卡GPU发送BIM模型数据,利用顶点着色器对三角形顶点、法线进行几何变换,同时执行第一阶段的光照计算,得到光线方向Light、视线方向View和视线与光线夹角的半角方向HalfieVector;

步骤304:将三角形栅格化为片元且不剔除背面对离散的场景进行着色、反走样处理,完成图形的栅格化。

4.如权利要求1所述的一种应用于BIM模型文件的数据渲染方法,其特征在于,

所述步骤4:配置数据存储服务器,所述数据存储服务器将这些数据由内存中的数据以可持久化的形式序列化为本地数据文件,并通过使用数据库的方法将得到的数据进行分门别类地存储,具体包括:

步骤401:用数据中心的计算资源来处理客户端的图形渲染任务,即将任务分配到“云端”;

步骤402:用户生成图形资源并提交至渲染云;而渲染云由资源存储服务器、渲染脚本解析服务器、渲染服务器、图形压缩服务器共同组成,用户提交的图形资源由资源存储服务器存储;

步骤403:渲染请求经过渲染脚本服务器解析后转换为渲染信息,由渲染服务器获取对应图形资源进行渲染,产生图像信息提交至图形压缩服务器压缩后,形成最终图片、视频资源返回给请求者,终端为渲染请求的提交者和渲染结果的接收者;

步骤404:在需要渲染服务时,终端将提交请求至渲染云,在得到渲染结果后,终端将负责把结果呈现给用户。

5.如权利要求1所述的一种应用于BIM模型文件的数据渲染方法,其特征在于,

所述步骤5:配置查询服务器,所述查询服务器设置模型Web端加载应用场景,从数据存储服务器中获取数据,通过原始构件模板的局部坐标系网格数据,结合其矩阵数据进行运算获得的世界坐标系,进行预定义的数据结构解析,得到矩阵数据和构件模版几何数据,具体包括:

步骤501:遍历空间索引,视锥剔除,通过摄像机与场景中物体的包围盒进行检测,如

果包围盒在视锥体外则不需要进行渲染,反之是可见的并把它们送入绘制管线中进行渲染;

步骤502:根据国家标准工业基础类平台规范,应用面向对象的程序语言对规范中定义的各类实体定义出对应的实体类及其属性,并对各类实体分别建立一个容器存放BIM模型中的各类实体的实例。

6.如权利要求1所述的一种应用于BIM模型文件的数据渲染方法,其特征在于:

所述步骤6:Web端渲染,使用矩阵数据与其构件模版几何数据,通过线性代数与空间解析几何的数学方法计算后,逐个解析信息,重新还原该构件在BIM模型世界坐标系下的实际物理位置数据,即还原成原始BIM模型中的构件,具体包括:

步骤601:建立一个基于多核PC机集群的并行绘制系统来对重建的BIM模型进行实时渲染,所述多核PC机集群包括交换控制节点和数个渲染节点,每个集群节点使用的是多核的PC机,渲染节点内部进行了多线程并行优化,把多核平台与分布式渲染充分结合起来;

步骤602:遮挡剔除,减少渲染图元数量:对图元做八叉树空间索引,然后根据视点计算场景中要剔除掉的图元,只绘制可见的图元;

步骤603:采用多重LOD(Levels of Detail),加速单图元渲染速度:多重LOD用不同级别的几何体来表示物体,距离越远加载的模型越粗糙,距离越近加载的模型越精细,从而在不影响视觉效果的前提下提高显示效率并降低存储;

单次渲染体量 = 图元数量 * 图元精度。

7.如权利要求1所述的一种应用于BIM模型文件的数据渲染方法,其特征在于:

所述步骤7:通过Restful风格的数据库查询接口,实现模型构件的属性信息查询,具体包括:

查询场景每一个 BIM 模型的属性信息,至少包括门的高宽信息;

设置标记提示能够帮助用户在发现设计错误时进行标识提示,提醒设计人员纠正错误;

构件定位让用户快速地移动到构件的位置,视点保存让用户保存下当前视点的位置。

一种应用于BIM模型文件的数据渲染方法

技术领域

[0001] 本发明属于工程建筑领域的计算机图形学,是BIM(Building Information Modeling)技术的核心部分——三维建筑图形渲染技术,具体涉及一种应用于BIM模型文件的数据渲染方法。

背景技术

[0002] BIM(建筑信息模型)为近年来建筑信息化领域核心理念之一,其数据的基础即为建筑的三维信息模型。相比于传统的二维设计、绘图,BIM技术全面利用三维图形技术,以建筑物(建筑构件、建筑整体)的三维图形为载体进一步挂接各种建筑信息参数,形成建筑信息模型,然后进行建筑物、甚至构件的全生命周期管理。可以说,三维图形是BIM技术的肌肉与皮肤,直观的三维图形表达与处理能有效帮助BIM关键应用的落地,实现其价值,譬如建筑模型的可视化、碰撞检测、5D虚拟施工等。这些应用很难想象在非图形、或者二维图形的世界中完成。

[0003] 目前困扰BIM行业的模型渲染技术的一个难点在于,模型规模体量庞大,尤其对于施工阶段的模型渲染来说,由于承接了各个设计源端的模型数据,包括全楼层、全专业的模型,数量相较一般的三维图形渲染系统来说,其体量早已远远超过了其渲染引擎所能承受的极限。此外,对于BIM行业的应用来说,其模型的精度也非常重要,即又要保证模型显示的正确性,也要在细部观察时不能丢失模型显示的相关细节。

[0004] 针对建筑类的大规模场景渲染,BIM行业相关的设计软件、模型浏览软件以及渲染引擎,与一般大场景显示技术类似,目前做的比较好的也是基于空间索引,进行视点动态排序(预处理),其致力于场景的空间索引在八叉树、BVH、BSP树(步骤1.1)上,根据视点变化进行视锥剔除、排序,对整个场景进行基于视点和空间索引的动态预处理之后,进行渐进显示,从而获得比较好的渲染效率和体验。一般而言,可以通过让距离视点较近、较大的物体先于其他场景图元显示,从而增强模型的整体显示视觉效果。但是这种处理方式仍然面临着如下两个问题:

[0005] 1、建筑模型的完整性,并非仅能靠远近大小来保证,比如如果切换视点到轴测观察位置,那么一般而言大部分特征为方正的建筑,会将角落图元显示出来而非整体,同时由于楼板等大图元的存在,导致用户最先看到的是楼板等图元,在模型交互显示的时候,会出现大量残缺空挡。

[0006] 2、无论是视距排序,还是大小排序,都需要按照视点的变化针对全场景来进行,这同样无疑会增加动态预处理时间,而缩短渲染的时间预算。

[0007] 在限定的带宽和用户电脑终端、移动终端条件下,数据量与耗时成正比关系。也正因为如此,为了反应建筑工程的实际原貌,越是精确的建筑工程BIM模型,在数据量上也越多,从而带来了BIM模型在存储、传输和终端渲染方面的障碍和瓶颈问题。因此,BIM模型超大数据量在行业内应用的难题是业内普遍共识,在很大程度上阻碍了BIM技术在工程建设行业中的应用。目前的渲染引擎系统、架构和算法,在面对更大规模模型的时候,都面临着

无法很好的适应和扩展的情况,针对一些更大体量的建筑工程三维模型的显示,都存在着诸多瑕疵和不足。

发明内容

[0008] 本发明旨在针对现有渲染系统的缺点,提出一种更加具有可扩展性、更加高效的大规模建筑场景渲染加速方法。

[0009] 本发明请求保护一种应用于BIM模型文件的数据渲染方法,其特征在于,包括步骤:

[0010] 步骤1:分别对BIM模型文件进行几何数据解析、几何材质解析和属性信息解析,所述几何数据解析过程、几何材质解析过程和属性信息解析过程可以分别独立并行运行;

[0011] 步骤2:解析数据,使用线性代数与空间解析几何的数学方法,对步骤一中同类型的构件的几何数据和材质数据进行分析,分析出使用可复用技术成型的构件模版数据和在BIM模型中的世界坐标系矩阵数据;

[0012] 步骤3:场景化处理,按照用户在操作界面上选择的不同应用场景,分别从步骤2的解析结果中检索相应的模型构件进行栅格化处理;

[0013] 步骤4:配置数据存储服务器,所述数据存储服务器将这些数据由内存中的数据以可持久化的形式序列化为本地数据文件,并通过使用数据库的方法将得到的数据进行分门别类地存储;

[0014] 步骤5:配置查询服务器,所述查询服务器设置模型Web端加载应用场景,从数据存储服务器中获取数据,通过原始构件模版的局部坐标系网格数据,结合其矩阵数据进行运算获得的世界坐标系,进行预定义的数据结构解析,得到矩阵数据和构件模版几何数据;

[0015] 步骤6:Web端渲染,使用矩阵数据与其构件模版几何数据,通过线性代数与空间解析几何的数学方法计算后,逐个解析信息,重新还原该构件在BIM模型世界坐标系下的实际物理位置数据,即还原成原始BIM模型中的构件;

[0016] 步骤7:通过Restful风格的数据库查询接口,实现模型构件的属性信息查询。

[0017] 本发明通过对BIM模型文件进行轻量化处理的基础上,为进一步减少模型数据量,提高数据传输、加载和渲染的速度,对模型按工作场景进一步拆解,按细分场景组织模型构件文件,满足工程建设管理的高性能、高场景化的业务需求;采用多重LOD用不同级别的几何体来表示物体,距离越远加载的模型越粗糙,距离越近加载的模型越精细,从而在不影响视觉效果的前提下提高显示效率并降低存储;批量绘制,提升渲染流畅度:绘制调用非常耗费CPU、并且通常会造成GPU时间闲置。本发明方法按应用场景加载和渲染模型构件,具备降低服务器硬件配置要求,减少IO操作频率,缩短前端用户的等待时间,提高用户体验等优点。

附图说明

[0018] 为了更清楚地说明本发明实施例或现有技术中的技术方案,下面将对实施例或现有技术描述中所需要使用的附图作简单地介绍,显而易见地,下面描述中的附图仅仅是本发明的一些实施例,对于本领域普通技术人员来讲,在不付出创造性劳动性的前提下,还可以根据这些附图获得其他的附图。

[0019] 附图1为本发明所涉及的一种应用于BIM模型文件的数据渲染方法的工作流程图。

具体实施方式

[0020] 下面将结合本发明实施例中的附图,对本发明实施例中的技术方案进行清楚、完整地描述,显然,所描述的实施例仅仅是本发明一部分实施例,而不是全部的实施例。基于本发明中的实施例,本领域普通技术人员在没有作出创造性劳动前提下所获得的所有其他实施例,都属于本发明保护的范围。

[0021] 本发明请求保护一种应用于BIM模型文件的数据渲染方法,其特征在于,包括步骤:

[0022] 步骤1:分别对BIM模型文件进行几何数据解析、几何材质解析和属性信息解析,所述几何数据解析过程、几何材质解析过程和属性信息解析过程可以分别独立并行运行;

[0023] 步骤2:解析数据,使用线性代数与空间解析几何的数学方法,对步骤一中同类型的构件的几何数据和材质数据进行分析,分析出使用可复用技术成型的构件模版数据和在BIM模型中的世界坐标系矩阵数据;

[0024] 步骤3:场景化处理,按照用户在操作界面上选择的不同应用场景,分别从步骤2的解析结果中检索相应的模型构件进行栅格化处理;

[0025] 步骤4:配置数据存储服务器,所述数据存储服务器将这些数据由内存中的数据以可持久化的形式序列化为本地数据文件,并通过使用数据库的方法将得到的数据进行分门别类地存储;

[0026] 步骤5:配置查询服务器,所述查询服务器设置模型Web端加载应用场景,从数据存储服务器中获取数据,通过原始构件模版的局部坐标系网格数据,结合其矩阵数据进行运算获得的世界坐标系,进行预定义的数据结构解析,得到矩阵数据和构件模版几何数据;

[0027] 步骤6:Web端渲染,使用矩阵数据与其构件模版几何数据,通过线性代数与空间解析几何的数学方法计算后,逐个解析信息,重新还原该构件在BIM模型世界坐标系下的实际物理位置数据,即还原成原始BIM模型中的构件;

[0028] 步骤7:通过Restful风格的数据库查询接口,实现模型构件的属性信息查询。

[0029] 进一步地,所述步骤1:分别对BIM模型文件进行几何数据解析、几何材质解析和属性信息解析,所述几何数据解析过程、几何材质解析过程和属性信息解析过程可以分别独立并行运行,还包括:

[0030] 步骤101:输入三维几何结构和二维纹理构成的三维建筑模型标识符,进行I/O操作,主要是指访问地图标识符库,从地图标识符库中调用、读取所需标识符的图形、纹理和材质描述数据;

[0031] 步骤102:对输入模型进行计算和分析,提取出基于点和线的三维模型表示,其中,点图元的提取通过递归化的多重抖动采样和蓝噪声采样完成,线模型则通过对输入的几何结构进行分析提取,并根据图像空间的颜色聚类关系进行分割标识符图形构造:将地图标识符的描述信息中moveto、lineto、arcto、curveto、closepath等绘制命令和坐标序列数据,构造成能够转换为光栅化技术可以处理的图形结构以及图形的样式配置等;

[0032] 步骤103:标识符图形变换:包括缩放、旋转、平移等坐标变换,笔画、填充等图形变换以及线宽、线型、线端头、线拐角等样式处理,将图形构造操作产生的图形结构经图形变

换操作生成与之对应的封闭多边形结构。

[0033] 材质信息为BIM图元信息的一种,其包括了BIM模型使用的贴图和材质参数,不同的贴图和材质参数最后得到的渲染效果不同,因此需要根据贴图和材质参数创建仿真的3D场景。具体的,贴图分为凹凸贴图、自发光贴图,法线贴图,透明贴图等,每种贴图都具有不同的效果反应,如:凹凸贴图可通过机电管线渲染的处理后,使得物体的表面可呈现凹凸不平的效果;粗糙度贴图是由材质表面的金属性的纹理,使得物体粗糙度更加明显。如:当通过贴图通道,将这两种贴图以jpg格式图片的形式,导入并作为贴图材质用在构件上,则粗糙度贴图会让构件有粗糙效果,而凹凸贴图会让凹凸的位置呈现的更加明显,这两种贴图会让粗糙的物体(石头、砖、皮革等)在细节上会显得更加真实细腻。材质的设置可以达到模型构件能具有更加真实的效果,而形成的辅助功能,在设置不同贴图的属性参数、UV参数、角度等来实现目的,如:当导入贴图之后,设置凹凸比例、折射率、环境强度等属性参数,则会相应对材质的粗糙反应、对光源而影响的折射效果、以及环境对材质产生的强弱等产生变化,材质会根据设置属性参数而变化最终呈现的效果,随之设置展UV参数,可以将贴图平铺展开,使得贴图更加舒展自然,或者设置角度参数,可将贴图以360度去旋转角度,改变贴图显示的方式。

[0034] 进一步地,所述步骤2:解析数据,使用线性代数与空间解析几何的数学方法,对步骤一中同类型的构件的几何数据和材质数据进行分析,分析出使用可复用技术成型的构件模版数据和在BIM模型中的世界坐标系矩阵数据,具体包括:

[0035] 步骤201:通过标识符配置进行点状数据与地图标识符的关联,所有的点状数据均采用相同的点状标识符;根据标识符配置的地图标识符标识sid,从标识符库中读取该点状标识符,并构建相应的标识符缓存;

[0036] 步骤202:将地图标识符标识sid作为键索引,标识符缓存SymbolCache作为键值传入到标识符缓存列表中。这样通过循环遍历点状数据,获取点状数据的坐标位置。依据坐标位置等参数构建绘制矩阵,并从标识符缓存列表中获取相应的标识符缓存。通过矩阵乘法实现标识符缓存数据的坐标变换,将标识符缓存的顶点数据变换到对应点状数据的坐标位置,向GPU中提交渲染数据以及绘制命令便可进行该点状数据的标识符化渲染。当 $i=POimNum$,表示处理完加载的所有数据,结束绘制。

[0037] 进一步地,所述步骤3:场景化处理,按照用户在操作界面上选择的不同应用场景,分别从步骤2的解析结果中检索相应的模型构件进行栅格化处理,具体包括:

[0038] 步骤301:计算每个建筑的表面积;然后,通过应用透视投影变换和视口变换计算该表面积对应的屏幕空间投影面积,确定无缝地表示该建筑对象所需要的点和线场景数量;

[0039] 步骤302:根据所需的点、线场景数量从点、线场景队列中选择对应的渲染场景进行渲染,场景离散:将封闭多边形进行离散的过程;针对二维数据,采用扫描线算法、将封闭多边形离散为水平方向分布的线段,针对三维数据,采用剖分算法将封闭多边形分解为多个三角形场景,其中,扫描线段和三角形场景都是着色的基本单元;

[0040] 步骤303:由中央处理器CPU向显卡GPU发送BIM模型数据,利用顶点着色器对三角形顶点、法线进行几何变换,同时执行第一阶段的光照计算,得到光线方向Light、视线方向View和视线与光线夹角的半角方向HalfieVector;

[0041] 步骤304:将三角形栅格化为片元且不剔除背面对离散的场景进行着色、反走样处理,完成图形的栅格化。

[0042] 通过将 BIM 模型根据场景内容按照个体划分为单独的子场景,再根据每个子场景包含的面片数量进行平均分配,既可以使每个渲染节点上分配到相近的任务量,又不影响最终场景的合成,使其能够达到平衡的结果。在渲染过程中,将相机空间中的深度定义为近裁剪面(near)到远裁剪面(far)之间的 z 值。 z 坐标和 X 、 Y 坐标一样,在经过变换、裁剪和透视除法后得到最终在屏幕上的位置。如公式(4.1)所示,为在透视变换之后得到新的 z' 值。在计算机渲染物体时所生成像素的深度信息保存在 z 缓冲区或者深度缓冲区中。深度缓冲区的作用是将一个距离观察平面的深度值和窗口中的每个像素关联起来,通过深度测试确定像素的遮挡关系,而最终保留的是离屏幕近的像素还是远的取决于选择的深度函数。

[0043] 根据图像的深度信息来完成最终渲染结果的合成,当渲染节点获取三维场景深度图后,连同渲染中间结果一起经过压缩传输至主端。主端根据所获得的深度信息将三维场景进行合成,每更新一次深度信息,将会对合成的渲染结果图像进行刷新。

[0044] 每更新一帧渲染结果,主端将对每一个渲染节点传输来的渲染中间结果根据对应深度信息变化的区域进行深度值的比较,保留下离屏幕最近的像素点进行对应颜色区域信息的更新,这样就得到新一帧的渲染结果。即在进行合成绘制图像中的每个像素之前,利用`glEnable(GL_DEPTH_TEST)`函数启用深度缓冲区,`OpenGL`会把它的深度值和已经存储在这个像素的深度值进行比较,在进行深度信息的刷新过程中将离屏幕更近的像素点留下。接着开启剔除操作,可以根据需求隐藏多边形的正面或者背面,消除不必要的渲染计算。

[0045] 进一步地,所述步骤4:配置数据存储服务器,所述数据存储服务器将这些数据由内存中的数据以可持久化的形式序列化为本地数据文件,并通过使用数据库的方法将得到的数据进行分门别类地存储,具体包括:

[0046] 步骤401:用数据中心的计算资源来处理客户端的图形渲染任务,即将任务分配到“云端”;

[0047] 步骤402:用户通过该工具生成图形资源并提交至渲染云;而渲染云由资源存储服务器、渲染脚本解析服务器、渲染服务器、图形压缩服务器共同组成,用户提交的图形资源由资源存储服务器存储;

[0048] 步骤403:渲染请求经过渲染脚本服务器解析后转换为渲染信息,由渲染服务器获取对应图形资源进行渲染,产生图像信息提交至图形压缩服务器压缩后,形成最终图片、视频资源返回给请求者,终端为渲染请求的提交者和渲染结果的接收者;

[0049] 步骤404:在需要渲染服务时,终端将提交请求至渲染云,在得到渲染结果后,终端将负责把结果呈现给用户。

[0050] 进一步地,所述步骤5:配置查询服务器,所述查询服务器设置模型Web端加载应用场景,从数据存储服务器中获取数据,通过原始构件模版的局部坐标系网格数据,结合其矩阵数据进行运算获得的世界坐标系,进行预定义的数据结构解析,得到矩阵数据和构件模版几何数据,具体包括:

[0051] 步骤501:遍历空间索引,视锥剔除,通过摄像机与场景中物体的包围盒进行检测,

如果包围盒在视锥体外则不需要进行渲染,反之是可见的并把它们送入绘制管线中进行渲染;

[0052] 步骤502:根据国家标准工业基础类平台规范,应用面向对象的程序语言对规范中定义的各类实体定义出对应的实体类及其属性,并对各类实体分别建立一个容器存放BIM模型中的各类实体的实例。

[0053] 进一步地,所述步骤6:Web端渲染,使用矩阵数据与其构件模版几何数据,通过线性代数与空间解析几何的数学方法计算后,逐个解析信息,重新还原该构件在BIM模型世界坐标系下的实际物理位置数据,即还原成原始BIM模型中的构件,具体包括:

[0054] 步骤601:建立一个基于多核PC机集群的并行绘制系统来对重建的BIM模型进行实时渲染,所述多核PC机集群包括交换控制节点和数个渲染节点,每个集群节点使用的是多核的PC机,渲染节点内部进行了多线程并行优化,把多核平台与分布式渲染充分结合起来;

[0055] 步骤602:遮挡剔除,减少渲染图元数量:对图元做八叉树空间索引,然后根据视点计算场景中要剔除掉的图元,只绘制可见的图元;

[0056] 步骤603:采用多重LOD(Levels of Detail),加速单图元渲染速度:多重LOD用不同级别的几何体来表示物体,距离越远加载的模型越粗糙,距离越近加载的模型越精细,从而在不影响视觉效果的前提下提高显示效率并降低存储;

[0057] 单次渲染体量 = 图元数量 * 图元精度。

[0058] 其中,首先启动服务器端的渲染节点等待 交互信息的发送,然后运行客户端的交互控制节点。控制节点首先根据采集到 的交互数据进行场景的更新,然后判断图元组归属,判断后把图元组的编号以 及视点位置信息发送给渲染节点,渲染节点接收到控制节点发送的信息后渲染 场景,渲染场景时进行一个流水线的并行优化,然后进行边缘融合处理,最后 是投影显示,把显示的图像投影到大屏幕上。控制节点与渲染节点之间并行工作,当渲染节点正在渲染第n帧和进行第n帧的边缘融合处理的时候,控制节 点执行的是第n+1帧的图元组归属判断操作。

[0059] 数个渲染节点之间采用分布式的渲染方法,在渲染节点的内部采用多线程 并行的优化方法。节点间的分布渲染实现是在渲染之前首先进行场景的分割, 在场景图对场景进行分割实质上是对视景物进行分割,但对视景物进行分割十 分麻烦,而且不灵活,本发明是通过对视景物的缩放和投影中心的移动来实现 灵活的分屏的,通过对每个渲染节点编号,根据编号的不同渲染节点渲染整个 场景的不同部分。

[0060] 进一步地,所述步骤7:通过Restful风格的数据库查询接口,实现模型构件的属性信息查询,具体包括:

[0061] 查询场景每一个 BIM 模型的属性信息,至少包括门的高宽信息;

[0062] 设置标记提示能够帮助用户在发现设计错误时进行标识提示,提醒设计人员纠正错误;

[0063] 构件定位让用户快速地移动到构件的位置,视点保存让用户保存下当前视点的位置。

[0064] 对于建筑物类的场景,在外部观察就是整个建筑的最外层,这个 层次嵌套网格索引(OOSI)不同于目前被广泛采用的BSP、BVH、树八叉树等空 间索引,而是完全按照空间的层次嵌套关系来建立的,体现的是空间层次的内外 关系,而这种关系是通过遮挡关系来决

定的。如果从某建筑物外部看,就是类似于用一把刀,先把建筑物外墙切下来,作为静态显示的图元,这部分图元不需要任何跟随视点的动态预处理,直接通过渲染器进入显卡渲染管线,这样不仅可以节省中间流程,快速绘制最重要的图元,同时也保证了渲染视觉效果“保真”度。同时这个过程要足够快,当场景在发生变化时,例如可见性、位置、显示状态发生变化时,能够得到快速更新,不影响用户体验。同时,其处理过程将在后台子线程进行,从而不会阻塞用户的交互操作。

[0065] 上仅为本发明的较佳实施例而已,并不用以限制本发明,凡在本发明的精神和原则之内,所做的任何修改、等同替换、改进等,均应包含在本发明保护的范围之内。

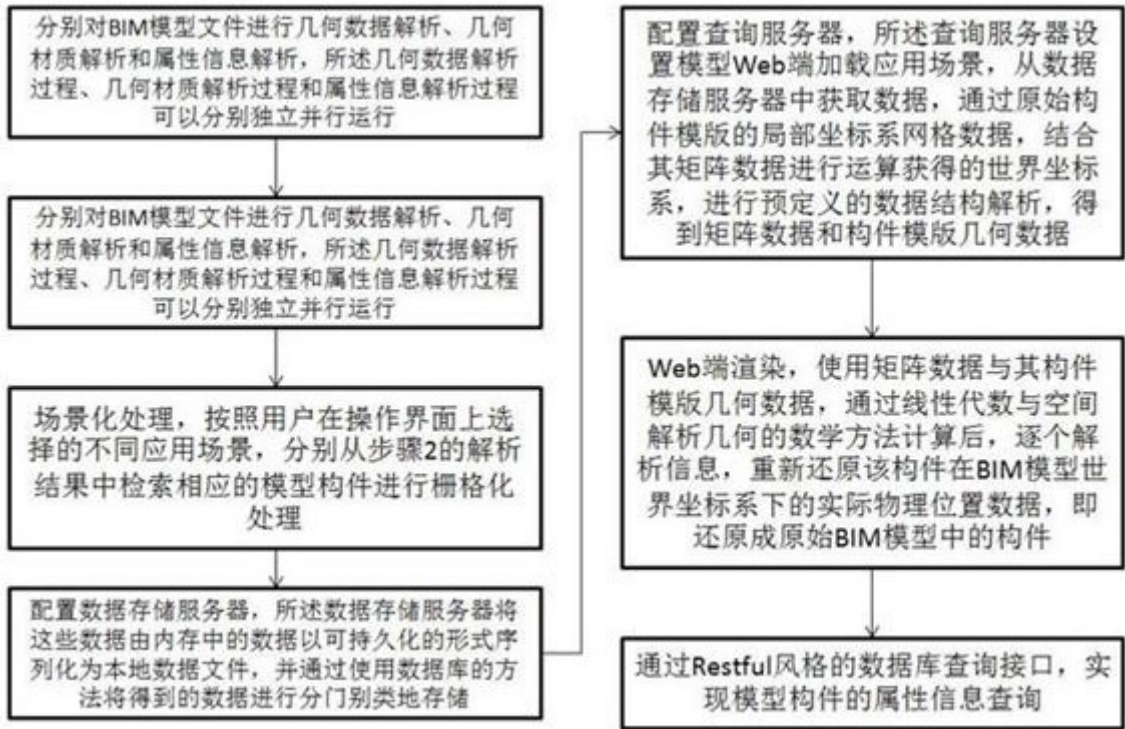


图1