



(12) 发明专利申请

(10) 申请公布号 CN 103077239 A

(43) 申请公布日 2013. 05. 01

(21) 申请号 201310008499. 6

(22) 申请日 2013. 01. 10

(71) 申请人 上海电力学院

地址 200090 上海市杨浦区平凉路 2103 号

(72) 发明人 郑莉

(74) 专利代理机构 上海申汇专利代理有限公司

31001

代理人 吴宝根

(51) Int. Cl.

G06F 17/30(2006. 01)

G06T 19/00(2011. 01)

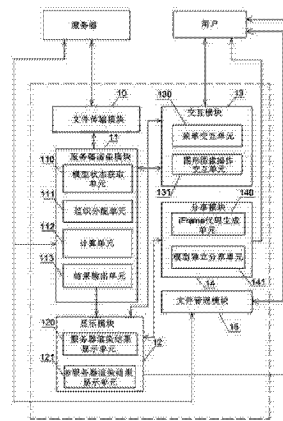
权利要求书2页 说明书8页 附图3页

(54) 发明名称

基于云渲染的 iFrame 嵌入式 Web3D 系统

(57) 摘要

本发明涉及一种基于云渲染的 iFrame 嵌入式 Web3D 系统,包括文件传输模块、服务器渲染模块、显示模块、交互模块、分享模块、文件管理模块,所述服务器渲染模块包括模型状态获取单元、组织分配单元、计算单元和结果输出单元。此系统不用安装任何插件即可通过浏览器直接流畅地在网页上浏览操控各种大型或者超大型的三维模型;对显示终端几乎没有要求,能够支持各种移动智能终端流畅地查看模型;能够支持生成 iFrame 代码,将模型嵌入各种不同的平台或者网站;能够支持多用户并发访问;由于客户端只能获取渲染后的图片,无法得到模型数据,用户的数据安全得到了很好的保障。



1. 一种基于云渲染的 iFrame 嵌入式 Web3D 系统,其特征在于,包括文件传输模块、服务器渲染模块、显示模块、交互模块、分享模块、文件管理模块,其中

所述文件传输模块,供用户将模型或者图纸上传至服务器;

所述服务器渲染模块,在服务器端根据模型状态对模型进行图形图像渲染,然后将渲染的结果以图片的形式发送给显示模块;

所述显示模块,用于前端显示服务器渲染模块的渲染结果和用户自定义显示内容;

所述交互模块,获取用户的交互操作请求和命令,并发送给服务器;

所述分享模块,将上传至服务器的模型分享给不同的用户,也能生成 iFrame 代码,供用户嵌入不同的平台;

所述文件管理模块,供用户对模型文件进行集中管理;

所述服务器渲染模块,包括模型状态获取单元、组织分配单元、计算单元和结果输出单元,其中:

所述模型状态获取单元,连续的从各终端获取模型状态,并将模型状态的时间点对比服务器时间,按照时间先后顺序进行排队,时间早的模型状态置于模型状态池的上端,时间晚的模型状态置于模型状态池的下端;

所述组织分配单元,组织将模型状态池中的模型状态按照时间先后的顺序作为渲染任务合理分配给 GPU/CPU,排在模型状态池上端的任务先分配计算,排在模型状态池下端的任务后分配计算,同时根据服务器端计算单元的计算能力和工作状态进行合理分配任务;

所述计算单元完成渲染任务;

所述结果输出单元输出渲染结果。

2. 根据权利要求 1 所述基于云渲染的 iFrame 嵌入式 Web3D 系统,其特征在于,所述服务器渲染模块采用硬件 GPU/CPU、虚拟 GPU/CPU 或 GPU/CPU 集群进行渲染。

3. 根据权利要求 1 所述基于云渲染的 iFrame 嵌入式 Web3D 系统,其特征在于,所述服务器渲染模块采用单树结构或者双树结构的渲染机制,在双树结构中:一棵树表示传统场景图中的节点树,用于选择、隐藏、显示、添加和删除几何节点,另一棵树是将传统节点树中的节点,按材质和坐标变换分类合并成数量更少但几何数据更多的大节点,专门用于渲染,两棵树的几何节点被设计为共享同一份几何数据的形式。

4. 根据权利要求 1 所述基于云渲染的 iFrame 嵌入式 Web3D 系统,其特征在于,

所述服务器渲染模块进行渲染的三维模型可以包括通过计算机辅助绘制的三维模型、通过扫描设备获取的三维模型、通过摄像设备摄像或摄像加后期编辑获取的三维模型、通过照相设备照相或照相加后期编辑合成获取的三维模型。

5. 根据权利要求 1 所述基于云渲染的 iFrame 嵌入式 Web3D 系统,其特征在于,

所述显示模块包括服务器渲染结果显示单元和非服务器渲染结果显示单元,服务器渲染结果显示单元显示来自于服务器渲染模块的渲染后的图片,非服务器渲染结果显示单元显示本地 GPU/CPU 上的信息。

6. 根据权利要求 5 所述基于云渲染的 iFrame 嵌入式 Web3D 系统,其特征在于,

所述服务器渲染结果显示单元解压并显示来自服务器渲染结果池中的内容,显示是按照发出的模型状态顺序逐帧显示,如果中间有模型状态渲染失败或者获取模型状态失败,则直接跳过,自动获取下一个模型状态的渲染结果。

7. 根据权利要求 5 所述的基于云渲染的 iFrame 嵌入式 Web3D 系统,其特征在于,所述非服务器渲染结果显示单元显示本地需求渲染结果,所有非服务器渲染结果的显示都在非服务器渲染结果显示单元中完成。

8. 根据权利要求 5 所述的基于云渲染的 iFrame 嵌入式 Web3D 系统,其特征在于,所述显示模块显示内容包括三维模型、二维图纸、二维图片、文字、动画、视频、图形图像。

9. 根据权利要求 1 所述的基于云渲染的 iFrame 嵌入式 Web3D 系统,其特征在于,所述交互模块包括菜单交互单元和图形图像操作交互单元,菜单交互单元是指通过菜单或者按钮的方式发出指令,通过指令来决定交互内容;图形图像操作交互单元是指在图形区域对模型的操作。

10. 根据权利要求 9 所述的基于云渲染的 iFrame 嵌入式 Web3D 系统,其特征在于,所述交互模块中交互方式包括选择、隐藏、切割、移动、旋转、缩放、播放、文本输入、文本编辑、文本删除、符号输入、符号编辑、符号删除、标记输入、标记编辑、标记删除、图形截取、图形绘制、图形编辑、图形删除、亮度调节、透明度调节、光影效果调节、投影方式调节、清晰度调节、渲染方式调节、模型颜色更换、布局调整、视图切换、干涉检查交互操作控制。

11. 根据权利要求 1 所述的基于云渲染的 iFrame 嵌入式 Web3D 系统,其特征在于,所述分享模块包括 iFrame 代码生成单元和模型独立分享单元,其中

iFrame 代码生成单元生成用户可嵌入其他网页的一段代码,用户嵌入其他网页后,任何用户可以通过被嵌入的网页浏览操控该三维模型,而实现的结果,依然是服务器渲染的结果,相当于在其他网页嵌入一个窗口,而每个用户之间相互独立,即 iFrame 代码嵌入是模型独立分享的一种方式;

模型独立分享单元负责实现模型独立分享机制的实现,每个用户都从渲染结果池中获取自己的模型状态渲染结果,每个模型状态的渲染结果不仅附带来源信息,而且附带模型状态来源时间,按照时间先后的顺序获取渲染结果。

12. 根据权利要求 11 所述的基于云渲染的 iFrame 嵌入式 Web3D 系统,其特征在于,所述分享模块可以嵌入的平台包括:网页、移动智能终端客户端、电脑客户端、平板电脑客户端、数控设备。

基于云渲染的 iFrame 嵌入式 Web3D 系统

技术领域

[0001] 本发明涉及一种网页浏览并操控三维模型技术,特别涉及一种基于云渲染的 iFrame 嵌入式 Web3D 系统。

背景技术

[0002] 现有的 web3D 系统有多种实现方式,例如申请号为 CN2010188278.8 的中国专利申请公开了一种可编辑的网页三维几何造型渲染方法,属于计算机图形处理技术领域。此专利在使用基于场景树的渲染引擎,解析数据交换协议实现 web3D 的基础上,通过解析图形节点中的几何造型,使用渲染引擎几何造型工场生成几何造型对象,通过插件扩展 JavaScript 底层 API 接口实现可编辑的三维几何造型渲染。该申请采用的是本地渲染的方式,需要使用插件,同时并未阐述云渲染技术,很难实现多用户并发同步访问。

[0003] 申请号为 CN201010293205.5 中国专利申请公开了一种 3D 模型的浏览实现方法、系统和服务器。该方法包括:服务器接收来自终端的页面访问请求,从请求访问的页面中解析 3D 模型文件;服务器根据所述终端的性能确定 3D 模型的精度;服务器获得与精度对应的简化 3D 模型;服务器将简化 3D 模型的参数发送给终端以便终端绘制 3D 模型。该申请只是利用服务器进行模型简化,而并未利用服务器进行渲染,仍然是采用客户端进行简化模型的渲染,将导致模型的信息大量丢失,并且简化的过程并不是客户指定的过程,并不能满足客户的需求,终端绘制 3D 模型必然影响本地浏览性能,同时如果是多个用户并发访问,则由于各个终端的性能不同而导致不同的用户看到不同的结果。

[0004] 申请号为 CN201110256005.7 中国专利申请公开了一种基于 Web3D 的超大场景实施渲染装置和方法。包括数据传输部分和数据渲染部分,数据传输部分包括结构转换装置,序列化装置,服务端线程管理装置,数据渲染部分包括客户端数据加载装置,动态显示管理装置。该发明并未改变客户端渲染的本质,只是利用服务器进行数据管理和分配。对客户端的要求也比较高,无法在移动设备上和配置较低的终端上实现大型模型和超大型模型的流畅浏览和操控。

[0005] Sketchfab(<http://sketchfab.com>)/sunglass(<https://www.sunglass.io>) 等公司采用了 WebGL 的方式进行 web3D 渲染,利用支持 html5 的浏览器安装插件打开三维模型进行浏览。他们的技术依赖本地渲染,而非本专利所述的服务器端渲染;他们要求浏览器必须支持 HTML5 的最新浏览器,在使用浏览器打开模型之前要下载模型至本地,因为其支持模型大小取决于客户端硬件性能和浏览器对 WebGL 的实现质量,由于本机硬件的限制,极难在 web 上打开大型及超大型模型,不可能在移动客户端上浏览大型及超大型模型,即便是在性能较好的个人电脑上,每次下载都需要等待很长时间,尤其是针对大于 100M 的大型模型,将几乎不可用。同时很难实现多用户并发访问。由于用户能够直接获取模型,所以用户模型数据安全性很低,很难实现多用户同步并发访问。

[0006] 如上所述,现有的 Web3D 技术日趋成熟,在电子商务、游戏、工程、教育、医学、地理信息等领域应用前景广阔。但在实际应用中,基于 B/S 模式(即浏览器和服务端架构)的

Web3D 应用发展存在两个致命性的因素,即网络带宽和用户客户端的实时渲染速度。用户要访问、使用一个 web3D 应用程序,客户端程序需要调用本地操作系统的底层图形接口完成三维图形渲染,如果每次使用都需要等待几分钟甚至十几分钟以上的时间来通过网络将三维数据从服务器端下载到本地,并在验证完数据完整性之后开始渲染,那么这个应用程序是无法满足实际需求的。随着电缆和光纤通信技术的快速发展,网络带宽虽有着大幅提升,但对于文件体积 100MB 以上超大场景的三维数据而言,传输带宽仍是制约发展的主要瓶颈。

[0007] 在电子商务及模型展示中,由于 web3D 技术采用本地渲染的限制,也很难快速的将模型在各种的网站上自由转述展示,很难类似优酷视频一般,能够自由嵌入各个网站内,使得 Web3D 技术很难得到广泛的应用。

发明内容

[0008] 本发明是针对浏览器无插件无法直接浏览大型及超大型模型的问题,提出了一种基于云渲染的 iFrame 嵌入式 Web3D 系统,实现浏览器直接浏览并操控三维模型、图纸,以及嵌入式图形图像分享浏览操控,解决浏览器无插件直接浏览大型及超大型模型的难题。

[0009] 本发明的技术方案为:一种基于云渲染的 iFrame 嵌入式 Web3D 系统,包括文件传输模块、服务器渲染模块、显示模块、交互模块、分享模块、文件管理模块,其中

所述文件传输模块,供用户将模型或者图纸上传至服务器;

所述服务器渲染模块,在服务器端根据模型状态对模型进行图形图像渲染,然后将渲染的结果以图片的形式发送给显示模块;

所述显示模块,用于前端显示服务器渲染模块的渲染结果和用户自定义显示内容;

所述交互模块,获取用户的交互操作请求和命令,并发送给服务器;

所述分享模块,将上传至服务器的模型分享给不同的用户,也能生成 iFrame 代码,供用户嵌入不同的平台;

所述文件管理模块,供用户对模型文件进行集中管理;

所述服务器渲染模块,包括模型状态获取单元、组织分配单元、计算单元和结果输出单元,其中:

所述模型状态获取单元,连续的从各终端获取模型状态,并将模型状态的时间点对比服务器时间,按照时间先后顺序进行排队,时间早的模型状态置于模型状态池的上端,时间晚的模型状态置于模型状态池的下端;

所述组织分配单元,组织将模型状态池中的模型状态按照时间先后的顺序作为渲染任务合理分配给 GPU/CPU,排在模型状态池上端的任务先分配计算,排在模型状态池下端的任务后分配计算,同时根据服务器端计算单元的计算能力和工作状态进行合理分配任务;

所述计算单元完成渲染任务;

所述结果输出单元输出渲染结果。

[0010] 所述服务器渲染模块采用硬件 GPU/CPU、虚拟 GPU/CPU 或 GPU/CPU 集群进行渲染。

[0011] 所述服务器渲染模块采用单树结构或者双树结构的渲染机制,在双树结构中:一棵树表示传统场景图中的节点树,用于选择、隐藏、显示、添加和删除几何节点,另一棵树是将传统节点树中的节点,按材质和坐标变换分类合并成数量更少但几何数据更多的大节

点,专门用于渲染,两棵树的几何节点被设计为共享同一份几何数据的形式。

[0012] 所述服务器渲染模块进行渲染的三维模型可以包括通过计算机辅助绘制的三维模型、通过扫描设备获取的三维模型、通过摄像设备摄像或摄像加后期编辑获取的三维模型、通过照相设备照相或照相加后期编辑合成获取的三维模型。

[0013] 所述显示模块包括服务器渲染结果显示单元和非服务器渲染结果显示单元,服务器渲染结果显示单元显示来自于服务器渲染模块的渲染后的图片,非服务器渲染结果显示单元显示本地 GPU/CPU 上的信息。所述服务器渲染结果显示单元解压并显示来自服务器渲染结果池中的内容,显示是按照发出的模型状态顺序逐帧显示,如果中间有模型状态渲染失败或者获取模型状态失败,则直接跳过,自动获取下一个模型状态的渲染结果。所述非服务器渲染结果显示单元显示本地需求渲染结果,所有非服务器渲染结果的显示都在非服务器渲染结果显示单元中完成。所述显示模块显示内容包括三维模型、二维图纸、二维图片、文字、动画、视频、图形图像。

[0014] 所述交互模块包括菜单交互单元和图形图像操作交互单元,菜单交互单元是指通过菜单或者按钮的方式发出指令,通过指令来决定交互内容;图形图像操作交互单元是指在图形区域对模型的操作。

[0015] 所述交互模块中交互方式包括选择、隐藏、切割、移动、旋转、缩放、播放、文本输入、文本编辑、文本删除、符号输入、符号编辑、符号删除、标记输入、标记编辑、标记删除、图形截取、图形绘制、图形编辑、图形删除、亮度调节、透明度调节、光影效果调节、投影方式调节、清晰度调节、渲染方式调节、模型颜色更换、布局调整、视图切换、干涉检查交互操作控制。

[0016] 所述分享模块包括 iFrame 代码生成单元和模型独立分享单元,其中

iFrame 代码生成单元生成用户可嵌入其他网页的一段代码,用户嵌入其他网页后,任何用户可以通过被嵌入的网页浏览操控该三维模型,而实现的结果,依然是服务器渲染的结果,相当于在其他网页嵌入一个窗口,而每个用户之间相互独立,即 iFrame 代码嵌入是模型独立分享的一种方式;

模型独立分享单元负责实现模型独立分享机制的实现,每个用户都从渲染结果池中获取自己的模型状态渲染结果,每个模型状态的渲染结果不仅附带来源信息,而且附带模型状态来源时间,按照时间先后的顺序获取渲染结果。

[0017] 所述分享模块可以嵌入的平台包括:网页、移动智能终端客户端、电脑客户端、平板电脑客户端、数控设备。

[0018] 本发明的有益效果在于:本发明基于云渲染的 iFrame 嵌入式 Web3D 系统,采用云渲染技术,通过浏览器直接流畅的浏览操控各种大型或者超大型的三维模型(>100MB),而不用安装任何插件;对显示终端几乎没有要求,能够支持各种移动智能终端查看模型;能够支持多用户并发访问;能够支持异地移动跨平台(如手机端、PC端)同步浏览操控模型,实现协同工作;由于客户端只能获取渲染后的图片,无法得到模型数据,用户的数据安全得到了很好的保障。

附图说明

[0019] 图 1 为本发明基于云渲染的 iFrame 嵌入式 Web3D 系统的结构框图;

图 2 为本发明基于云渲染的 iFrame 嵌入式 Web3D 系统实现服务器渲染的流程图；
图 3 为本发明基于云渲染的 iFrame 嵌入式 Web3D 系统流程图。

具体实施方式

[0020] 基于云渲染的 iFrame 嵌入式 Web3D 系统利用云渲染技术,利用云端强大的计算能力和渲染能力对模型状态进行渲染,通过浏览器或客户端的交互显示,将对模型的渲染分化成一帧帧图片的渲染,将服务器计算渲染的结果实时的和用户进行交互,实现 web3D 及各种移动客户端均可以无缝浏览操控模型的结果,同时,利用 iFrame 技术,实现 Web3D 模型的自由分享。

[0021] 如图 1 所示基于云渲染的 iFrame 嵌入式 Web3D 系统的结构框图。系统包括:文件传输模块 10、服务器渲染模块 11、显示模块 12、交互模块 13、分享模块 14 以及文件管理模块 15。

[0022] 文件传输模块 10 供用户上传模型至服务器。用户将模型上传至服务器后,获得一个服务器地址,后期的分享可以是该地址的分享。文件的内容,包括但不限于三维模型、二维图纸、二维图片、文字、动画、和 / 或视频、及其他图形图像。三维模型的来源包括但不限于通过计算机辅助绘制的三维模型、通过扫描设备获取的三维模型、通过摄像设备摄像或摄像加后期编辑获取的三维模型、通过照相设备照相或照相加后期编辑合成获取的三维模型。

[0023] 在极简单的系统中,可以利用系统(如 Linux 或 windows)本身的文件传输功能实现文件传输模块 10 的功能。

[0024] 服务器渲染模块 11 负责在服务器端根据模型状态对模型进行图像渲染,然后将渲染的结果以图片的形式发送给前端显示模块。服务器渲染是对某个时间点的模型状态进行渲染,将渲染的结果以图片的呈现,连续的时间点渲染的结果连续的以图片的形式呈现,实现对模型的实时渲染和动态交互。服务器渲染模块 11 可以采用多种组织方式。当计算量需求呈曲线变化,在某些特定时间段突然出现较大峰值时,可以采用虚拟 GPU/CPU 技术进行合理组织分配计算任务,当长期持续有较高增长的计算量需求时,可以采用 GPU/CPU 集群的方式进行合理组织分配计算任务。

[0025] 服务器渲染模块 11 包括模型状态获取单元 110、组织分配单元 111、计算单元 112 和结果输出单元 113。

[0026] 模型状态获取单元 110 连续的从不同的终端获取模型状态,并将模型状态的时间点对比服务器时间,按照时间先后顺序进行排队,时间早的模型状态置于模型状态池的上端,时间晚的模型状态置于模型状态池的下端。

[0027] 组织分配单元 111 组织将模型状态池中的模型状态按照时间先后的顺序作为渲染任务合理分配给多个 GPU/CPU,排在模型状态池上端的任务先分配计算,排在模型状态池下端的任务后分配计算,同时根据服务器端计算单元 112 采用的组织方式进行合理分配。例如,如果采用虚拟 GPU/CPU 的方式,当计算量小的时候,将按照小的计算量进行 GPU 虚拟,让多余的计算能力进行保留储备,当突然出现计算量峰值时,迅速利用闲置的计算能力进行计算以满足计算需求,而不是无论多少计算量需求就使用全部的 GPU 计算能力进行计算,以致于当新的计算需求出现的时候无法应对,避免了计算能力的浪费,达到有效管理和

利用有限计算资源和能力的效果；如果采用 GPU/CPU 集群的方式，当计算需求量持续增长时，现有的所有计算能力全部利用上都无法应对，则需要硬件上扩展 GPU 的个数或者计算机的台数，已达到扩展服务器整体计算能力，利用 GPU/CPU 集群管理，对多个 GPU/CPU 的计算能力进行集中管理，将不同的计算任务进行合理分配，计算量大的任务分配给计算能力较强的 GPU/CPU 或者分配给几个计算能力较弱的 GPU/CPU，以达到有效的管理利用计算能力并合理扩展计算资源和能力的效果。

[0028] 计算单元 112 完成渲染任务。渲染机制为 GPU 根据模型状态进行实施渲染，渲染结果为 Bitmap，CPU 生成相应的图片，进行图片压缩后，将相应的结果输送至渲染结果池。接到模型状态后，对比上一状态，相同内容部分予以继承，不同的部分重新渲染，以减少计算量。服务器渲染利用现代显卡 (GPU) 的特点，针对大数据量的场景图 (Scene Graph)，采用相应的渲染机制 (例如传统的单树结构或者一种可被高效解析和渲染的双树结构，在双树结构中：一棵树表示传统场景图中的节点树，用于选择、隐藏、显示、添加和删除几何节点。另一棵树是将传统节点树中的节点，按材质和坐标变换分类合并成数量更少但几何数据更多的大节点，专门用于渲染，对于材质相对较少的大场景，可大大减少渲染过程中 CPU 和 GPU 之间的通信次数，从而极大的提高渲染速度。同时，为了节约空间，两棵树的几何节点被设计为共享同一份几何数据的形式，并将其持久化到文件中，大大减少了每次打开文件时用于解析、构建场景图的时间，也不会因为增加一棵树而明显增大文件大小)，对模型进行渲染。

[0029] 结果输出单元 113 输出渲染结果。将计算的结果发送到渲染结果池，每个模型状态的渲染结果都附带来源信息，渲染结果按照获取模型状态时间的先后顺序进行排列，获取模型时间状态早的渲染结果置于渲染结果池的上端，获取模型时间晚的渲染结果置于渲染结果池的下端。由于是实时渲染状态，假设由于网速或者其他因素的影响，假设最大延迟为 5s，则某个模型状态的渲染结果输出后每经过 5s 被自动删除，一旦在该时间段内未获取该状态，则自动获取下一个模型状态的渲染结果。如图 2 为基于云渲染的 iFrame 嵌入式 Web3D 系统实现服务器渲染的流程图。

[0030] 显示模块 12 用于前端显示渲染结果和用户自定义显示内容。既可以显示服务器对模型的渲染结果，也可以显示本地其他需求渲染结果。

[0031] 显示模块 12 包括服务器渲染结果显示单元 120 和非服务器渲染结果显示单元 121。

[0032] 在服务器渲染结果显示单元 120 中，解压并显示来自服务器渲染结果池中的内容，显示是按照发出的模型状态顺序逐帧显示，如果中间有模型状态渲染失败或者获取模型状态失败，则直接跳过，自动获取下一个模型状态的渲染结果。服务器渲染结果显示单元 120 显示的内容主要体现在对模型 (或其他文件) 的操作以及控制上。服务器渲染结果显示单元 120 中显示的内容为图片，但是连续显示的结果对用户来说，即用户直观看到的内容，包括但不限于三维模型、二维图纸、二维图片、文字、动画、和 / 或视频、及其他图形图像。所渲染显示的三维模型，包括但不限于通过计算机辅助绘制的三维模型、通过扫描设备获取的三维模型、通过摄像设备摄像或摄像加后期编辑获取的三维模型、通过照相设备照相或照相加后期编辑合成获取的三维模型。无论通过何种途径获得的模型，可以直接导入或者通过软件转换成相应的格式导入即可。

[0033] 在非服务器渲染结果显示单元 121 中,显示本地需求渲染结果,所有非服务器渲染结果的显示都应该在非服务器渲染结果显示单元 121 中完成。该单元的显示结果应该来自本地 GPU/CPU,例如独立加入的水印、广告、本地标记(图形标记或者文本标记,本地标记是指未附着在模型而是添加在屏幕上的标记,当模型静止时添加标记,当对模型进行旋转或者其他操作时,该标记被删除或者被储存在特定的场景中)、本地注释、截图等等。

[0034] 交互模块 13 完成用户与系统的交互。按照交互对象分类,分为与模型的交互和与其他交互;按照交互的区域分类,分为菜单交互和图形区域交互。交互的方式包括:鼠标、手势、摄像、语音等等。交互的内容包括但不限于:选择、取消选择、隐藏、取消隐藏、切割、取消切割、移动、旋转、缩放、播放、附着在模型上的文本输入、文本编辑、文本删除、附着在模型上的符号输入、符号编辑、符号删除、附着在模型上的标记输入、标记编辑、标记删除、图形绘制、图形编辑、图形删除、亮度调节、透明度调节、光影效果调节、投影方式调节(如:平行投影、透视投影)、清晰度调节、渲染方式调节(如:线框模型、实色模型等等)、模型颜色更换、布局调整、视图切换、干涉检查等等交互操作控制。

[0035] 交互模块 13 包括菜单交互单元 130 和图形图像操作交互单元 131。菜单交互单元 130 是指通过菜单或者按钮的方式发出指令,通过指令来决定交互内容。菜单交互包括选择菜单栏的命令(例如点击、输入等等),还包括点击菜单栏后的下拉菜单或者新弹出的对话框,以及在对话框中的交互(例如点击、勾选、输入、选择等等);图形图像操作交互单元 131 是指在图形区域对模型的操作,所有的操作必然引起图形区域内容发生改变,例如:指定用户坐标系并旋转模型,则体现在图形区域的变化是:用户坐标系位置随着指定位置的变化而发生改变,模型随着旋转操作的变化而进行相应的旋转响应。

[0036] 分享模块 14 用于将已经上传至服务器的模型分享给其他用户,实际的分享分为模型分享和非模型分享,模型分享是指接受分享的用户同样具有模型操控权,非模型分享是指用户可以看到或者听到模型展示的结果,但是并不具有真正的模型操控权,例如在自己当面展示给其他人观看或体验都属于非模型分享。本例中讨论的分享指的是模型分享,更具体的说是指模型独立分享,即当我将模型分享给其他用户时,其他用户同样具有模型操控权,而且每个用户之间都是独立的,自己操控的结果只能自己看见,自己在操控模型的过程中其他人也不能通过网络进行干预。

[0037] 分享模块 14 包括 iFrame 代码生成单元 140、模型独立分享单元 141。iFrame 代码生成单元 140 生成用户可嵌入其他网页的一段代码,用户嵌入其他网页后,任何用户可以通过被嵌入的网页浏览操控该三维模型,而实现的结果,依然是服务器渲染的结果,相当于在其他网页嵌入一个窗口,而每个用户之间相互独立,及 iFrame 代码嵌入是模型独立分享的一种方式。下面的是 iFrame 代码示例:

```
<iframe id="frame-viewer" src="http://www.xxx.com/viewer/v1/?path=5090859d5716755726000014.sat&view=5" width="1024" height="768" frameborder="no" border="0px"></iframe>
```

本代码中使用基本的 html 代码格式 `<iframe></iframe>`;其中指定了 viewer(模型浏览器)的 ID(标识符),模型的来源及位置 src(source):

```
http://www.xxx.com/viewer/v1/?path=5090859d5716755726000014.sat&view=5 同
```

时指定 viewer 窗口的大小 :width="1024" height="768";以及是否有边框及边框的大小 frameborder="no" border="0px"。

[0038] 模型独立分享单元 141,负责实现模型独立分享机制的实现。每个用户都从渲染结果池中获取自己的模型状态渲染结果,不获取其他用户的模型状态渲染结果,即得到的是相互独立的分享。每个模型状态的渲染结果不仅仅附带来源信息,而附带模型状态来源时间,按照时间先后的顺序获取渲染结果。由于是实时渲染状态,假设由于网速或者其他因素的影响,假设最大延迟为 5s,如果假定某个模型状态的渲染结果输出后每经过 5s 被自动删除,一旦在该时间段内获取该状态失败,则自动获取下一个模型状态的渲染结果。模型独立分享单元在提供图形图像分享的同时,根据用户的交互或者 API 的调用获取相关的模型数据,将模型数据发送给显示单元或者 API 调用者,将模型相关数据进行表达。模型的相关数据可以包括所有的前期设计输入或者硬件输入的数据,例如模型的尺寸、精度、材质、数量、价格等等。

[0039] 文件管理模块 15 供用户管理服务器端的模型。模型管理可以根据用户需要设定多种功能,例如删除、重命名、替代等功能。在极简系统中,可以利用系统(如 Linux 或 windows)本身的文件管理功能实现文件管理模块 15 的功能。

[0040] 本发明的系统流程图如图 3 所示,用户上传模型至服务器后,自己可以浏览并操控模型,如果希望分享给其他用户,如果不嵌入其他网站,则可以直接将模型地址发给其他用户,其他用户可以直接在该网站上浏览并操控模型;如果需要嵌入其他网站,则生成 iFrame 代码,将 iFrame 代码插入需要嵌入 viewer(视口)的网站源代码中,调整到需要的位置,用户便可以在已经嵌入 iFrame 代码的网站中浏览、操控模型,允许多个用户并发访问;具有服务器管理员权限的用户可以在服务器对模型进行管理,例如删除、替代等等。

[0041] 本发明的发明点主要在于:1. 实现技术为云渲染(或服务器渲染);2. 浏览器浏览三维模型时不需要安装插件;3. 使用浏览器浏览三维模型时不需要下载模型;4. 支持模型大小与客户端和浏览器无关,主要取决于服务器性能;5. 渲染帧率主要取决于网速及服务器性能,不取决于客户端的硬件性能;6. 客户端只能获取渲染后的图片,而不会缓存在客户端;7. 支持多用户并发访问,并发访问数量与渲染服务器数量及计算能力相关,而与客户端无关;8. 支持 iFrame 代码嵌入其他网站来实现内容分享。

[0042] 本发明所使用的技术手段包括云计算、服务器端渲染、web 图形图像的浏览、基于图形图像的操作、iFrame 技术。

[0043] 服务器渲染采用的是对当前瞬间状态的静止图形图像进行渲染,将连续的静止图形图像的渲染结果按照先后顺序发送至前台进行显示,以产生动态浏览操控的结果。

[0044] 能够同时支持多用户同时并发访问,即多个用户可以同时对相同的模型进行操作也可以同时对不同的模型进行操作。最大并发渲染请求数与渲染服务器数量成正比。用户在浏览模型的过程中,一直需要网络连接。

[0045] 当 iFrame 被嵌入至各种不同的平台时,不同的用户访问相同的模型时,互不影响,每个用户都能看到自己想要的操控结果。可以嵌入的平台包括:网页、移动智能终端客户端、电脑客户端、平板电脑客户端、数控设备等等。该系统可以与 PDM、PLM、ERP 等等各种管理系统结合。该系统与其他系统结合后,可以根据需要的流程传递相关模型数据。

[0046] 对本地显卡无特殊要求,云端完成渲染。即利用较低配置的客户端一样可以高速

流畅的浏览服务器上的大型和超大型模型。支持模型大小取决于服务器硬件性能,与客户端和浏览器无关。模型浏览操控的速度和流畅度与网速和服务器端硬件及软件相关。渲染帧率主要取决于网速,而不是客户端的硬件性能。

[0047] 数据安全性很高,客户端只能获取渲染后的图片,无法得到模型数据。

[0048] 本发明的应用范围包括:各种电子商务网站;各种企业网站;各种门户网站;各种中专业网站;各种交流网站;各种论坛;各种微博;各种博客;分散式异地协作;3D模型档案集中管理;可与ISO文控中心整合;可与PDM/PLM/ERP整合;3D档案互联网异地备份;企业沟通协作;各种移动审图系统;各种客户端软件;各种网络沟通插件;基于网页的沟通插件;各种专业技术交流相关产品;各种移动终端上的应用服务;基于项目的沟通;基于作品的沟通;基于设计产品的沟通;基于三维模型的沟通;基于二维图纸的沟通;基于图片的沟通;其他交流沟通应用领域。

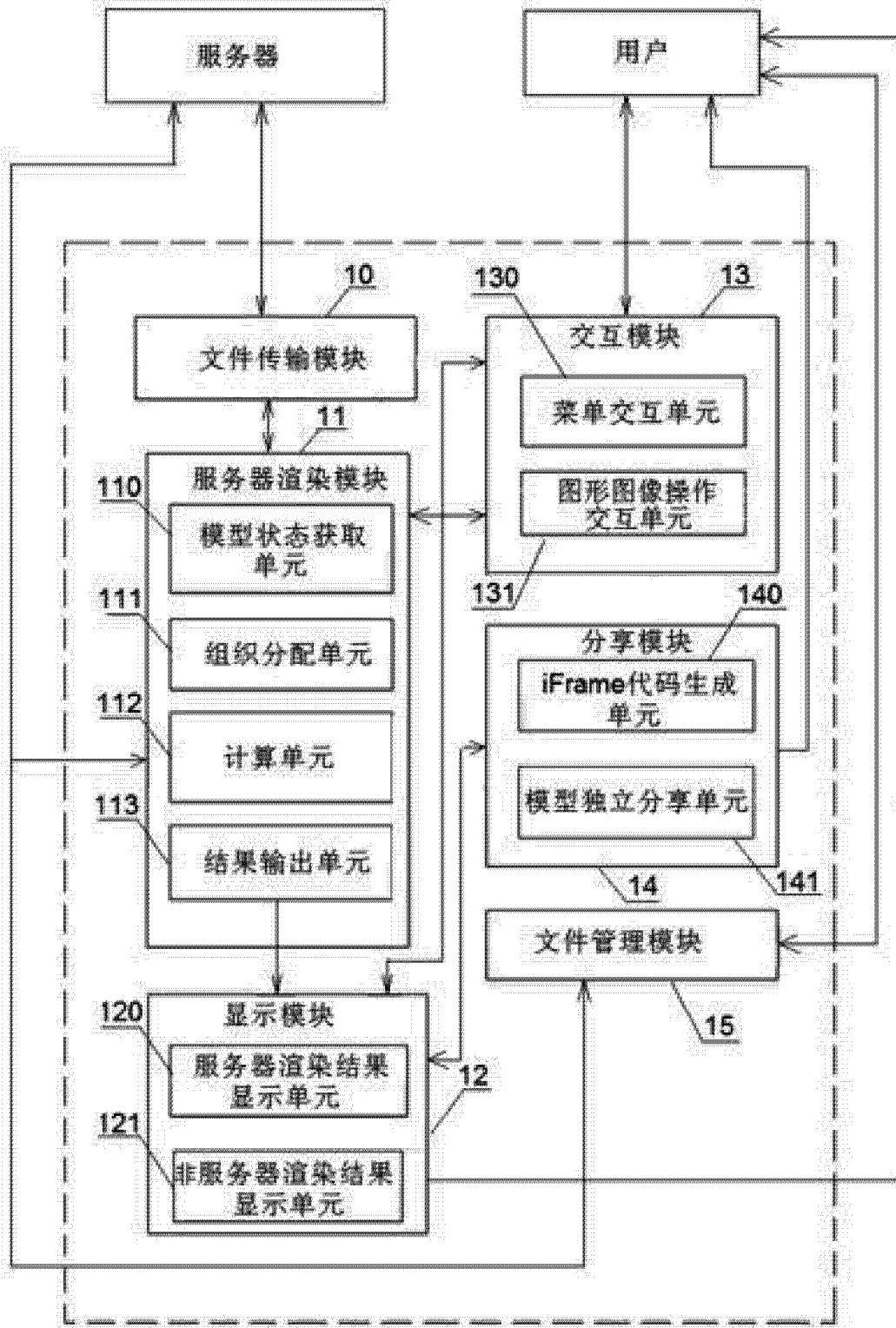


图 1

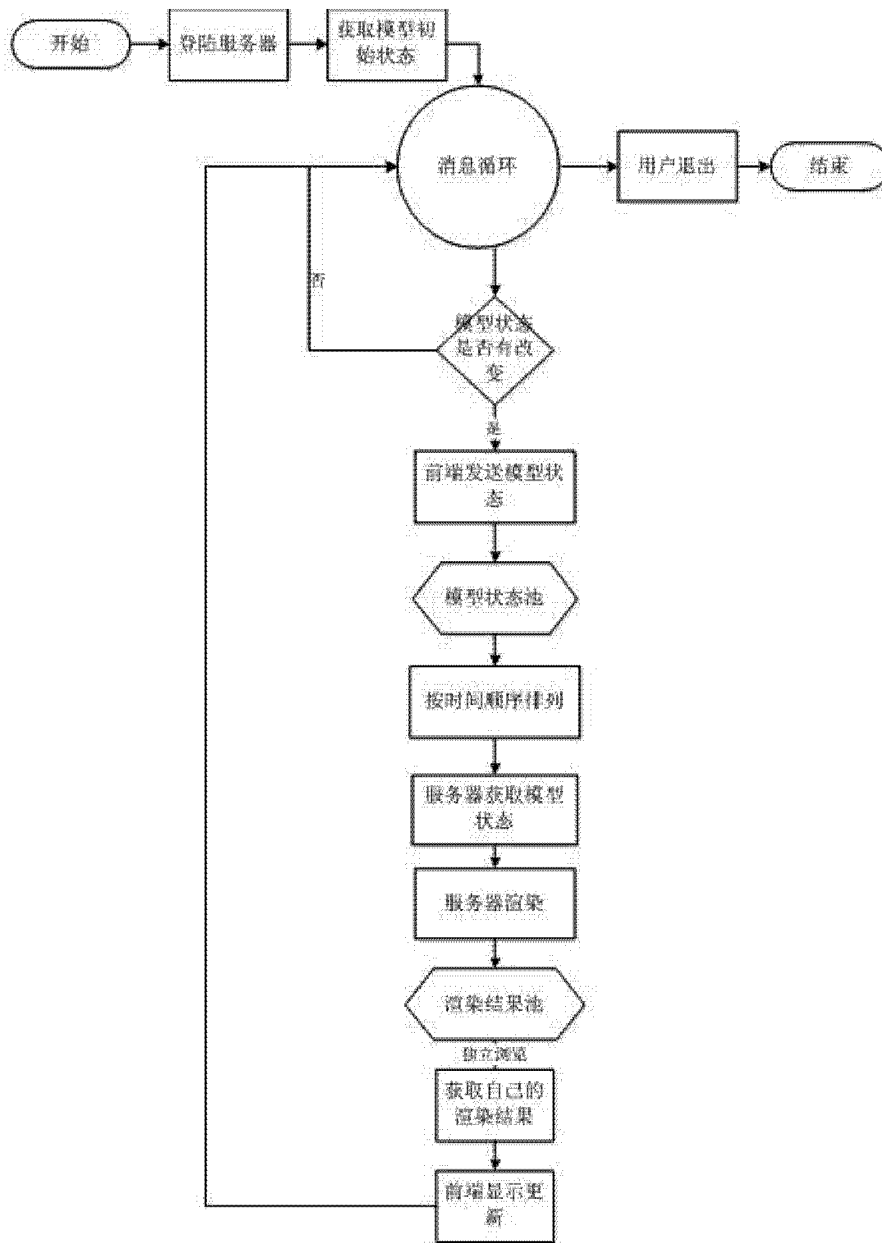


图 2

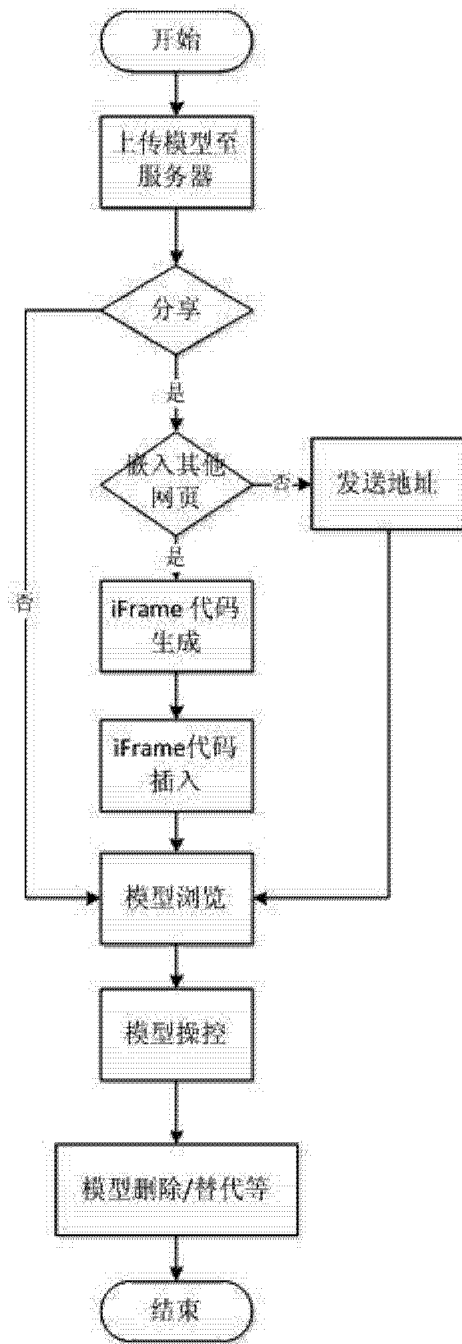


图 3