



(12) 发明专利申请

(10) 申请公布号 CN 115550687 A

(43) 申请公布日 2022. 12. 30

(21) 申请号 202211167495.8

(22) 申请日 2022.09.23

(71) 申请人 中国电信股份有限公司
地址 100033 北京市西城区金融大街31号

(72) 发明人 郑秋宏 张志超 魏莱 丁鹏
沈云

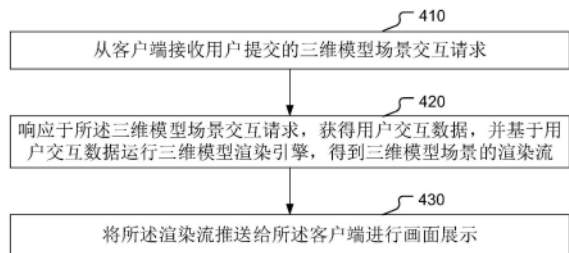
(74) 专利代理机构 北京律智知识产权代理有限公司 11438
专利代理师 孙宝海

(51) Int. Cl.
H04N 21/234 (2011.01)
H04N 21/431 (2011.01)
H04L 65/1108 (2022.01)
H04L 65/80 (2022.01)
H04N 13/106 (2018.01)

权利要求书2页 说明书10页 附图4页

(54) 发明名称
三维模型场景交互方法、系统、设备、装置及存储介质

(57) 摘要
本发明提供了三维模型场景交互方法、系统、设备、装置及存储介质,通过响应于三维模型场景交互请求,获得用户交互数据,并基于用户交互数据运行三维模型渲染引擎,得到三维模型场景的渲染流,将渲染流推送给客户端进行画面展示。在本实施例中,将基于数字孪生应用的三维模型场景渲染进程放到服务器,相比于用户本地设备,服务器能够提供更强大的GPU计算能力,具有更强大的三维模型场景渲染能力,显著提升客户端侧的三维模型场景的画质。同时有效降低对用户端设备的硬件要求,减小其运行压力。



1. 一种三维模型场景交互方法,其特征在于,应用于服务器,所述三维模型场景交互方法包括:

从客户端接收用户提交的三维模型场景交互请求;

响应于所述三维模型场景交互请求,获得用户交互数据,并基于用户交互数据运行三维模型渲染引擎,得到三维模型场景的渲染流;

将所述渲染流推送给所述客户端进行画面展示。

2. 根据权利要求1所述的三维模型场景交互方法,其特征在于,所述将所述渲染流推送给所述客户端进行画面展示,包括:

在与所述客户端之间建立基于WebRTC技术的数据交互通道的情况下,通过所述数据交互通道,将所述渲染流推送给所述客户端进行画面展示。

3. 根据权利要求1所述的三维模型场景交互方法,其特征在于,所述响应于所述三维模型场景交互请求,获得用户交互数据,并基于用户交互数据运行三维模型渲染引擎,得到三维模型场景的渲染流,包括:

响应于所述三维模型场景交互请求,获取用户交互数据并解析得到交互数据类型、格式及所述三维模型渲染引擎的接入路径;

根据所述交互数据类型、格式及所述三维模型渲染引擎的接入路径匹配所述三维模型渲染引擎,并基于所述三维模型渲染引擎的数据格式对所述用户交互数据进行重构;

调用所述三维模型渲染引擎对重构的用户交互数据进行解析,基于解析的用户交互数据调用封装的扩展功能组件对所述三维模型场景执行对应功能,得到渲染流。

4. 根据权利要求1所述的三维模型场景交互方法,其特征在于,在从客户端接收用户提交的三维模型场景交互请求之前,所述三维模型场景交互方法还包括:

接收所述客户端提交的三维模型渲染引擎接入请求;

响应于所述三维模型渲染引擎接入请求获得用户配置信息,并基于所述用户配置信息建立数据交互通道,其中,所述渲染流是通过所述数据交互通道推送给所述客户端的。

5. 根据权利要求4所述的三维模型场景交互方法,其特征在于,在从客户端接收用户提交的三维模型场景交互请求之前,所述三维模型场景交互方法还包括:

响应于所述三维模型渲染引擎接入请求,获得接入配置信息,并基于所述接入配置信息进行三维模型场景的初始渲染,得到所述三维模型场景的初始渲染流,并通过所述数据交互通道推送给所述客户端。

6. 根据权利要求4所述的三维模型场景交互方法,其特征在于,响应于所述三维模型渲染引擎接入请求获得用户配置信息,并基于所述用户配置信息建立数据交互通道,包括:

在接收到多个客户端提交的三维模型渲染引擎接入请求的情况下,响应于每个三维模型渲染引擎接入请求获得用户配置信息,并基于每个用户配置信息建立对应的数据交互通道,对应于多个客户端建立得到多个数据交互通道。

7. 一种三维模型场景交互系统,其特征在于,包括:

客户端,接收用户的输入,响应于用户的输入,向服务器发送三维模型场景交互请求,并在从所述服务器接收到渲染流的情况下,对所述渲染流进行解析,得到并显示三维模型场景的画面;

所述服务器,响应于所述三维模型场景交互请求,获得用户交互数据,并基于所述用户

交互数据运行三维模型渲染引擎,得到所述三维模型场景的渲染流,将所述渲染流推送给所述客户端。

8. 一种三维模型场景交互装置,其特征在于,应用于服务器,所述三维模型场景交互装置包括:

接收模块,从客户端接收用户提交的三维模型场景交互请求;

渲染模块,响应于所述三维模型场景交互请求,获得用户交互数据,并基于用户交互数据运行三维模型渲染引擎,得到三维模型场景的渲染流;

推送模块,将所述渲染流推送给所述客户端进行画面展示。

9. 一种电子设备,其特征在于,包括:

处理器;

存储器,其中存储有所述处理器的可执行指令;

其中,所述处理器配置为经由执行所述可执行指令来执行权利要求1至6任一项所述的三维模型场景交互方法的步骤。

10. 一种计算机可读存储介质,用于存储程序,其特征在于,所述程序被处理器执行时实现权利要求1至6任一项所述的三维模型场景交互方法的步骤。

三维模型场景交互方法、系统、设备、装置及存储介质

技术领域

[0001] 本发明涉及计算机技术领域,具体地说,涉及三维模型场景交互方法、系统、设备、装置及存储介质。

背景技术

[0002] 数字孪生(Digital Twin),也称为数字映射、数字镜像,指的是在信息化平台内模拟物理实体、流程或者系统,类似实体系统在信息化平台中的双胞胎。借助数字孪生,可以在信息化平台上了解物理实体的状态,甚至可以对物理实体里面预定义的接口组件进行控制,从而帮助组织监控运营、执行预测性维护和改进流程。

[0003] 数字孪生的本质是信息建模,旨在为现实世界中的实体对象在数字虚拟世界中构建完全一致的数字模型。但数字孪生涉及的信息建模已不再是基于传统的底层信息传输格式的建模,而是对实体对象外部形态、内部机理和运行关系等方面的整体抽象描述,其难度和应用效果相较于传统建模呈指数级增长。主要表现在数字孪生可以有多个变身,即根据不同用途和场景构建形态各异的数字模型。

[0004] 数字孪生应用依托具备高可视化能力的三维模型场景实现,因此如何获得高质量画质的三维模型场景,是业界普遍考虑的课题。

[0005] 需要说明的是,上述背景技术部分公开的信息仅用于加强对本发明的背景的理解,因此可以包括不构成对本领域普通技术人员已知的现有技术的信息。

发明内容

[0006] 针对现有技术中的问题,本发明的目的在于提供三维模型场景交互方法、系统、设备、装置及存储介质,克服了现有技术的困难,能够在降低对用户设备的硬件要求下获得高质量画质的三维模型场景。

[0007] 本发明的实施例提供一种三维模型场景交互方法,其应用于服务器,三维模型场景交互方法包括:

[0008] 从客户端接收用户提交的三维模型场景交互请求;

[0009] 响应于三维模型场景交互请求,获得用户交互数据,并基于用户交互数据运行三维模型渲染引擎,得到三维模型场景的渲染流;

[0010] 将渲染流推送给客户端进行画面展示。

[0011] 在一些实施例中,从客户端接收用户提交的三维模型场景交互请求,包括:

[0012] 在与客户端之间建立基于WebRTC技术的数据交互通道的情况下,通过数据交互通道,将渲染流推送给客户端进行画面展示。

[0013] 在一些实施例中,响应于三维模型场景交互请求,获得用户交互数据,并基于用户交互数据运行三维模型渲染引擎,得到三维模型场景的渲染流,包括:

[0014] 响应于三维模型场景交互请求,获取用户交互数据并解析得到交互数据类型、格式及三维模型渲染引擎的接入路径;

[0015] 根据交互数据类型、格式及三维模型渲染引擎的接入路径匹配三维模型渲染引擎,并基于三维模型渲染引擎的数据格式对用户交互数据进行重构;

[0016] 调用三维模型渲染引擎对重构的用户交互数据进行解析,基于解析的用户交互数据调用封装的扩展功能组件对三维模型场景执行对应功能,得到渲染流。

[0017] 在一些实施例中,在从客户端接收用户提交的三维模型场景交互请求之前,三维模型场景交互方法还包括:

[0018] 接收客户端提交的三维模型渲染引擎接入请求;

[0019] 响应于三维模型渲染引擎接入请求获得用户配置信息,并基于用户配置信息建立数据交互通道,其中,渲染流是通过数据交互通道推送给客户端的。

[0020] 在一些实施例中,在从客户端接收用户提交的三维模型场景交互请求之前,三维模型场景交互方法还包括:

[0021] 响应于三维模型渲染引擎接入请求,获得接入配置信息,并基于接入配置信息进行三维模型场景的初始渲染,得到三维模型场景的初始渲染流,并通过数据交互通道推送给客户端。

[0022] 在一些实施例中,响应于三维模型渲染引擎接入请求获得用户配置信息,并基于用户配置信息建立数据交互通道,包括:

[0023] 在接收到多个客户端提交的三维模型渲染引擎接入请求的情况下,响应于每个三维模型渲染引擎接入请求获得用户配置信息,并基于每个用户配置信息建立对应的数据交互通道,对应于多个客户端建立得到多个数据交互通道。

[0024] 本发明实施例还提供一种三维模型场景交互系统,其包括:

[0025] 客户端,接收用户的输入,响应于用户的输入,向服务器发送三维模型场景交互请求,并在从服务器接收到渲染流的情况下,对渲染流进行解析,得到并显示三维模型场景的画面;

[0026] 服务器,响应于三维模型场景交互请求,获得用户交互数据,并基于用户交互数据运行三维模型渲染引擎,得到三维模型场景的渲染流,将渲染流推送给客户端。

[0027] 本发明实施例还提供一种三维模型场景交互装置,其应用于服务器,三维模型场景交互装置包括:

[0028] 接收模块,从客户端接收用户提交的三维模型场景交互请求;

[0029] 渲染模块,响应于三维模型场景交互请求,获得用户交互数据,并基于用户交互数据运行三维模型渲染引擎,得到三维模型场景的渲染流;

[0030] 推送模块,将渲染流推送给客户端进行画面展示。

[0031] 本发明的实施例还提供一种电子设备,包括:

[0032] 处理器;

[0033] 存储器,其中存储有处理器的可执行指令;

[0034] 其中,处理器配置为经由执行可执行指令来执行上述三维模型场景交互方法的步骤。

[0035] 本发明的实施例还提供一种计算机可读存储介质,用于存储程序,程序被执行时实现上述三维模型场景交互方法的步骤。

[0036] 本发明的目的在于提供三维模型场景交互方法、系统、设备及存储介质,通过响应

于三维模型场景交互请求,获得用户交互数据,并基于用户交互数据运行三维模型渲染引擎,得到三维模型场景的渲染流,将渲染流推送给客户端进行画面展示。在本实施例中,将基于数字孪生应用的三维模型场景渲染进程放到服务器,相比于用户本地设备,服务器能够提供更强大的GPU计算能力,具有更强大的三维模型场景渲染能力,显著提升客户端侧的三维模型场景的画质。同时有效降低对用户端设备的硬件要求,减小其运行压力。

附图说明

[0037] 通过阅读参照以下附图对非限制性实施例所作的详细描述,本发明的其它特征、目的和优点将会变得更加明显。

[0038] 图1是本公开实施例提供的三维模型场景交互系统的架构图;

[0039] 图2是本公开实施例提供的三维模型场景交互方法的时序图;

[0040] 图3是本公开实施例提供的三维模型场景交互系统的原理示意图;

[0041] 图4是本公开实施例提供的三维模型场景交互方法的流程图;

[0042] 图5是本公开实施例提供的三维模型场景交互装置的模块结构示意图;

[0043] 图6是本发明的电子设备运行的示意图;

[0044] 图7示出了根据本公开实施方式的存储介质的示意图。

具体实施方式

[0045] 现在将参考附图更全面地描述示例实施方式。然而,示例实施方式能够以多种形式实施,且不应被理解为限于在此阐述的实施方式。相反,提供这些实施方式使本发明全面和完整,并将示例实施方式的构思全面地传达给本领域的技术人员。

[0046] 附图仅为本发明的示意性图解,并非一定是按比例绘制。图中相同的附图标记表示相同或类似的部分,因而将省略对它们的重复描述。附图中所示的一些方框图是功能实体,不一定必须与物理或逻辑上独立的实体相对应。可以采用软件形式来实现这些功能实体,或在一个或多个硬件转发模块或集成电路中实现这些功能实体,或在不同网络和/或处理器装置和/或微控制器装置中实现这些功能实体。

[0047] 此外,附图中所示的流程仅是示例性说明,不是必须包括所有的步骤。例如,有的步骤可以分解,有的步骤可以合并或部分合并,且实际执行的顺序有可能根据实际情况改变。具体描述时使用的“第一”、“第二”以及类似的词语并不表示任何顺序、数量或者重要性,而只是用来区分不同的组成部分。需要说明的是,在不冲突的情况下,本发明的实施例及不同实施例中的特征可以相互组合。

[0048] 在相关技术中,复杂三维模型场景的高质量画质渲染依赖高性能GPU主机的计算能力,但是普通用户的设备通常不具备本地渲染条件。

[0049] 本公开实施例提出一种基于数字孪生应用的三维模型场景构建方法,其发明思想是,将三维模型渲染引擎部署在服务器,通过数字孪生应用与服务器交互,利用三维模型渲染引擎渲染三维模型场景。相比于相关技术中使用用户设备部署三维模型渲染引擎的方案,服务器能够提供更强大的GPU计算能力,具有更强大的三维模型场景渲染能力,显著提升三维模型场景的画质。

[0050] 图1展示本公开实施例提供的三维模型场景渲染系统的架构图,如图1所示,该三

维模型场景渲染系统包括：客户端11及服务器12。其中，客户端11可以是数字孪生应用、浏览器或运行数字孪生应用的用户设备。

[0051] 基于图1所示系统，图2展示基于一种三维模型场景渲染方法的时序图，如图2所示，该三维模型场景渲染方法包括如下步骤：

[0052] 步骤210：客户端11接收用户的输入；

[0053] 步骤220：客户端11响应于用户的输入，向服务器12发送三维模型场景交互请求；

[0054] 步骤230：服务器12响应于三维模型场景交互请求，获得用户交互数据，并基于用户交互数据运行三维模型渲染引擎，得到三维模型场景的渲染流；

[0055] 步骤240：服务器12将渲染流推送给客户端11；

[0056] 步骤250：客户端11对渲染流进行解析，得到并展示三维模型场景的画面。

[0057] 在本实施例中，三维模型场景在服务器12一侧渲染完成，渲染流可以是以HTML文件的形式发送给客户端11，这个HTML文本被客户端11解析之后，不需要经过JavaScript脚本的执行，即可直接构建出希望的DOM树，即三维模型场景的画面，并展示到页面中。

[0058] 在本实施例中，将基于数字孪生应用的三维模型场景渲染进程放到服务器，相比于用户本地设备，服务器能够提供更强大的GPU计算能力，具有更强大的三维模型场景渲染能力，显著提升客户端侧的三维模型场景的画质。同时有效降低对用户端设备的硬件要求，减小其运行压力。

[0059] 在本公开实施例中，服务器12可以是云端服务器或本地物理服务器，如远端高性能服务器。

[0060] 在本公开实施例中，客户端11与服务器12之间可以基于WebRTC技术建立数据交互通道，服务器12通过该数据交互通道向客户端11推送渲染流。

[0061] 其中，WebRTC (Web Real-Time Communications) 是一项实时通讯技术，它允许网络应用或者站点，在不借助中间媒介的情况下，建立浏览器之间点对点 (Peer-to-Peer) 的连接，实现视频流和 (或) 音频流或者其他任意数据的传输。

[0062] 在这种情况下，使用WebRTC，客户端11与服务器12之间能够实现渲染流的即时性。

[0063] 在本公开实施例中，如图3所示，服务器3还部署有交互模块30，交互模块30包括交互能力封装子模块31和交互数据隔离子模块32；

[0064] 其中，交互能力封装子模块31用于将数字孪生应用中对三维模型场景的一个或多个用户交互能力进行封装，得到扩展功能组件并实现在三维模型渲染引擎中，例如对多个用户交互能力进行一一封装。

[0065] 其中，用户交互能力可以包括对三维模型场景进行三维定位、三维轨迹绘制、三维空间镜头行为操控、三维模型拆解等，在此不作限定。

[0066] 其中，对用户交互能力封装得到扩展功能组件，该扩展功能组件是相对于三维模型渲染引擎内部的基础功能组件来说的，三维模型渲染引擎可通过基础功能组件构建得到，在此基础上，可在三维模型渲染引擎之外进行功能组件的扩展。

[0067] 因此，本实施例通过封装，得到具有通用性的扩展功能组件，任何三维模型渲染引擎或用户均能够调用，提升本实施例的应用范围。

[0068] 在本实施例中，通过交互能力封装子模块31的用户交互能力封装方法包括如下步骤：

[0069] 根据三维模型场景的用户交互能力的需求定义面向用户的交互数据的类型、格式、交互请求路径规范等交互数据；

[0070] 匹配到三维模型渲染引擎的数据格式，以面向三维模型渲染引擎对交互数据进行解析与重构后发送给对应的三维模型渲染引擎；

[0071] 通过三维模型渲染引擎对重新构造后的用户交互数据进行提取和解析，根据解析的用户交互数据进行扩展功能组件的封装。

[0072] 其中，交互请求路径包括：三维渲染引擎地址+三维模型场景id/端口，因此，交互请求路径可以是三维渲染引擎的接入路径。在一种示例中，交互数据包括如下字段：

	参数	类型	必填	取值范围	备注
[0073]	apiType	string	是	“SetCameraRoaming”	标识该接口为镜头漫游接口
	points	object	是		结构数组
	points.camera_location	string	是		路径其中一个镜头的位置
[0074]	points.pitch	int	是	[-90,89]	镜头俯仰角
	points.yaw	int	是	[0,359]	镜头偏航角
	points.fly_time	int	是	任意正整数	镜头移动持续时间(单位:秒)

[0075] 通过将数字孪生应用的通用性用户交互能力进行封装，可支持用户以跨平台的方式对三维模型渲染引擎进行调用，大大简化数字孪生应用的构建，提高数字孪生应用的构建效率，能够向各类数字孪生应用的垂直行业提供一套灵活便捷的即插即用的能力服务，解除用户与三维模型渲染引擎间交互的耦合性，增强交互的便捷性与灵活性，有利于交互能力的泛化。

[0076] 而且，使用WebRTC技术，能够实现用户与服务器三维模型场景的实时互动操作，充分利用云端服务器或远端高性能主机的GPU计算能力对三维模型场景进行渲染，使得用户端与三维模型渲染引擎端远程交互的反馈和本地直接交互一样，有效降低对用户端设备的硬件要求，减小其运行压力。

[0077] 在这种情况下，用户调用交互模块3发起交互请求，交互能力封装子模块31根据定义的交互数据类型、格式要求以及交互请求路径规范对交互数据进行重构，并向三维模型渲染引擎传递携带重构交互数据的交互请求。

[0078] 三维模型渲染引擎处理接收到交互请求后,对经过交互封装子模块31重新构造后的交互数据进行提取和解析,根据场景配置数据的格式、类型以及配置逻辑调用封装的扩展功能组件进行交互功能实现。处理完成后,三维模型渲染引擎进行渲染流的推送,完成用户侧三维模型场景画面的显示更新。

[0079] 因此,通过对用户交互能力进行封装,本公开实施例提供一种接口调用的方式对三维模型渲染引擎渲染的三维模型场景进行交互操作,易于支持丰富的交互功能实现。

[0080] 在本公开实施例中,交互模块3还包括交互数据隔离子模块32,交互数据隔离子模块32用于:

[0081] 实现不同用户对同一三维模型场景的交互数据间的相互隔离,即用户1对三维模型场景的交互操作不会改变用户2所获取的渲染流,能够消除各自操作之间的影响。

[0082] 具体地,用户在建立和三维模型渲染引擎的连接前,进行用户配置与接入配置,用户配置指定用户的身份、权限类型。接入配置指定三维模型渲染引擎的接入路径、三维模型场景的编号(同一三维模型渲染引擎可能同时搭载多个三维模型场景)、三维模型场景的画质参数等。

[0083] 三维模型渲染引擎在接收到接入请求后,自动检测用户配置信息与接入配置信息。若用户配置信息正确且为新增接入用户,则交互数据隔离子模块32为其建立专享的数据交互通道,实现交互渲染流的隔离。根据接入配置信息调取相应的三维模型数据进行三维模型场景渲染,通过用户专享的数据交互通道进行渲染流的推送。此时,用户侧可成功加载三维模型场景初始画面。

[0084] 之后,用户调用交互能力封装子模块31发起三维模型场景交互请求,完成用户侧三维模型场景画面的更新。

[0085] 因此,通过对不同用户设置专享的数据交互通道,能够有效对交互数据进行隔离,避免不同用户对同一三维模型场景交互操作间的相互干扰。

[0086] 图4为本公开实施例提供的一种具体的三维模型场景交互方法的时序图,本方法的执行主体是对应数字孪生应用的服务器,如图4所示,该三维模型场景交互方法包括如下步骤:

[0087] 步骤410:从客户端接收用户提交的三维模型场景交互请求;

[0088] 步骤420:响应于三维模型场景交互请求,获得用户交互数据,并基于用户交互数据运行三维模型渲染引擎,得到三维模型场景的渲染流;

[0089] 步骤430:将渲染流推送给客户端进行画面展示。

[0090] 在本实施例中,将基于数字孪生应用的三维模型场景渲染进程放到服务器,相比于用户本地设备,服务器能够提供更强大的GPU计算能力,具有更强大的三维模型场景渲染能力,显著提升客户端侧的三维模型场景的画质。同时有效降低对用户端设备的硬件要求,减小其运行压力。

[0091] 在本公开实施例中,客户端与服务器之间可以基于WebRTC技术建立数据交互通道,服务器通过该数据交互通道向客户端推送渲染流。

[0092] 这可以提升三维模型场景的交互即时性。

[0093] 在本公开其他实施例中,服务器与客户端之间又可以通过其他通信方式建立通信连接,在此并不做限定。

[0094] 在本公开实施例中,响应于三维模型场景交互请求,获得用户交互数据,并基于用户交互数据运行三维模型渲染引擎,得到三维模型场景的渲染流,包括:

[0095] 响应于三维模型场景交互请求,获取用户交互数据并解析得到交互数据类型、格式及三维模型渲染引擎的接入路径;

[0096] 根据交互数据类型、格式及三维模型渲染引擎的接入路径匹配三维模型渲染引擎,并基于三维模型渲染引擎的数据格式对用户交互数据进行重构;

[0097] 调用三维模型渲染引擎对重构的用户交互数据进行解析,基于解析的用户交互数据调用封装的扩展功能组件对三维模型场景执行对应功能,得到渲染流。

[0098] 在本实施例,服务器侧事先定义面向用户的交互数据类型、格式及三维模型渲染引擎的接入路径等交互模块信息,使得用户能够根据交互需求调用交互模块来发起三维模型场景交互请求。

[0099] 响应于该三维模型场景交互请求匹配到具体的三维模型渲染引擎,通过该三维模型渲染引擎的数据格式对用户交互数据进行重构,重构的用户交互数据能够被三维模型渲染引擎进行识别及解析,以根据解析数据调用对应的扩展功能组件,该扩展功能组件是事先对数字孪生应用中的用户交互能力进行封装得到的。对于封装方法可参考上文内容,在此不作限定。

[0100] 借助于封装的扩展功能组件,用户能够以跨平台的方式对三维模型渲染引擎进行调用,解除用户与三维模型渲染引擎间交互的耦合性,增强交互的便捷性与灵活性,有利于交互能力的泛化。

[0101] 在本公开实施例中,在从客户端接收用户提交的三维模型场景交互请求之前,三维模型场景交互方法还包括:

[0102] 接收客户端提交的三维模型渲染引擎接入请求;

[0103] 响应于三维模型渲染引擎接入请求获得用户配置信息,并基于用户配置信息建立数据交互通道,其中,渲染流是通过数据交互通道推送给客户端的。

[0104] 在本实施例中,数据交互通道是专用于客户端与服务器之间的数据交互通道,能够提升数据安全性和防干扰性。

[0105] 在本公开实施例中,在接收到多个客户端提交的三维模型渲染引擎接入请求的情况下,响应于每个三维模型渲染引擎接入请求获得用户配置信息,并基于每个用户配置信息建立对应的数据交互通道,对应于多个客户端建立得到多个数据交互通道。

[0106] 在本实施例中,该多个数据交互通道之间相互隔离,每个用户对三维模型场景的交互操作不会改变其他用户所获取的渲染流,能够消除各自交互操作之间的影响。

[0107] 在本公开实施例中,在从客户端接收用户提交的三维模型场景交互请求之前,三维模型场景交互方法还包括:

[0108] 响应于三维模型渲染引擎接入请求,获得接入配置信息,并基于接入配置信息进行三维模型场景的初始渲染,得到三维模型场景的初始渲染流,并通过数据交互通道推送给客户端。

[0109] 在该实施例中,在客户端显示初始化的三维模型场景的情况下,可使用图1所示方法对三维模型场景进行进一步渲染更新。

[0110] 本公开实施例图4所示三维模型场景交互方法可以基于图1或图3所示三维模型场

景交互系统实现,也可以由其他系统实现,在此不作限定。

[0111] 图5是本公开提供的三维模型场景交互装置的一种实施例的模块示意图,如图5所示,三维模型场景交互装置500包括但不限于如下模块:

[0112] 接收模块510,从客户端接收用户提交的三维模型场景交互请求;

[0113] 渲染模块520,响应于三维模型场景交互请求,获得用户交互数据,并基于用户交互数据运行三维模型渲染引擎,得到三维模型场景的渲染流;

[0114] 推送模块530,将渲染流推送给客户端进行画面展示。

[0115] 上述模块的实现原理参见图4所示三维模型场景交互方法中的相关介绍,此处不再赘述。

[0116] 在可选实施例中,接收模块510具体用于:

[0117] 与客户端通过WebRTC技术建立通信连接,通过通信连接,从客户端接收用户提交的三维模型场景交互请求。

[0118] 在可选实施例中,渲染模块520具体用于:

[0119] 响应于三维模型场景交互请求,获取用户交互数据并解析得到交互数据类型、格式及三维模型渲染引擎的接入路径;

[0120] 根据交互数据类型、格式及三维模型渲染引擎的接入路径匹配三维模型渲染引擎,并基于三维模型渲染引擎的数据格式对用户交互数据进行重构;

[0121] 调用三维模型渲染引擎对重构的用户交互数据进行解析,基于解析的用户交互数据调用封装的扩展功能组件对三维模型场景执行对应功能,得到渲染流。

[0122] 在可选实施例中,接收模块510具体用于:

[0123] 在从客户端接收用户提交的三维模型场景交互请求之前,接收客户端提交的三维模型渲染引擎接入请求;

[0124] 响应于三维模型渲染引擎接入请求获得用户配置信息,并基于用户配置信息建立数据交互通道,其中,三维模型场景交互请求和渲染流均是通过数据交互通道传送。

[0125] 在可选实施例中,渲染模块520具体用于:

[0126] 在从客户端接收用户提交的三维模型场景交互请求之前,响应于三维模型渲染引擎接入请求,获得接入配置信息,并基于接入配置信息进行三维模型场景的初始渲染,得到三维模型场景的初始渲染流,并通过数据交互通道推送给客户端。

[0127] 在可选实施例中,接收模块510具体用于:

[0128] 在接收到多个客户端提交的三维模型渲染引擎接入请求的情况下,响应于每个三维模型渲染引擎接入请求获得用户配置信息,并基于每个用户配置信息建立对应的数据交互通道,对应于多个客户端建立得到多个数据交互通道。

[0129] 在本实施例中,将基于数字孪生应用的三维模型场景渲染进程放到服务器,相比于用户本地设备,服务器能够提供更强大的GPU计算能力,具有更强大的三维模型场景渲染能力,显著提升客户端侧的三维模型场景的画质。同时有效降低对用户端设备的硬件要求,减小其运行压力。

[0130] 本发明实施例还提供一种电子设备,包括处理器、存储器,其中存储有处理器的可执行指令。其中,处理器配置为经由执行可执行指令来执行的三维模型场景交互方法的步骤。

[0131] 如上所示,本公开实施例的电子设备将基于数字孪生应用的三维模型场景渲染进程放到服务器,相比于用户本地设备,服务器能够提供更强大的GPU计算能力,具有更强大的三维模型场景渲染能力,显著提升客户端侧的三维模型场景的画质。同时有效降低对用户端设备的硬件要求,减小其运行压力。

[0132] 所属技术领域的技术人员能够理解,本发明的各个方面可以实现为系统、方法或程序产品。因此,本发明的各个方面可以具体实现为以下形式,即:完全的硬件实施方式、完全的软件实施方式(包括固件、微代码等),或硬件和软件方面结合的实施方式,这里可以统称为“电路”、“模块”或“平台”。

[0133] 图6本发明的电子设备的结构示意图。下面参照图6来描述根据本发明的这种实施方式的电子设备600。图6显示的电子设备600仅仅是一个示例,不应对本发明实施例的功能和使用范围带来任何限制。

[0134] 如图6所示,电子设备600以通用计算设备的形式表现。电子设备600的组件可以包括但不限于:至少一个处理单元610、至少一个存储单元620、连接不同平台组件(包括存储单元620和处理单元610)的总线630、显示单元640等。

[0135] 其中,存储单元存储有程序代码,程序代码可以被处理单元610执行,使得处理单元610执行本说明书三维模型场景交互方法部分中描述的根据本发明各种示例性实施方式的步骤。例如,处理单元610可以执行图4所示的步骤。

[0136] 存储单元620可以包括易失性存储单元形式的可读介质,例如随机存取存储单元(RAM) 621和/或高速缓存存储单元622,还可以进一步包括只读存储单元(ROM) 623。

[0137] 存储单元620还可以包括具有一组(至少一个)程序模块625的程序/实用工具624,这样的程序模块625包括但不限于:处理系统、一个或者多个应用程序、其它程序模块以及程序数据,这些示例中的每一个或某种组合中可能包括网络环境的实现。

[0138] 总线630可以为表示几类总线结构中的一种或多种,包括存储单元总线或者存储单元控制器、外围总线、图形加速端口、处理单元或者使用多种总线结构中的任意总线结构的局域总线。

[0139] 电子设备600也可以与一个或多个外部设备60(例如键盘、指向设备、蓝牙设备等)通信,还可与一个或者多个使得用户能与该电子设备600交互的设备通信,和/或与使得该电子设备600能与一个或多个其它计算设备进行通信的任何设备(例如路由器、调制解调器等等)通信。这种通信可以通过输入/输出(I/O)接口650进行。

[0140] 并且,电子设备600还可以通过网络适配器660与一个或者多个网络(例如局域网(LAN),广域网(WAN)和/或公共网络,例如因特网)通信。网络适配器660可以通过总线630与电子设备600的其它模块通信。应当明白,尽管图中未示出,可以结合电子设备600使用其它硬件和/或软件模块,包括但不限于:微代码、设备驱动器、冗余处理单元、外部磁盘驱动阵列、RAID系统、磁带驱动器以及数据备份存储平台等。

[0141] 本发明实施例还提供一种计算机可读存储介质,用于存储程序,程序被执行时实现的三维模型场景交互方法的步骤。在一些可能的实施方式中,本发明的各个方面还可以实现为一种程序产品的形式,其包括程序代码,当程序产品在终端设备上运行时,程序代码用于使终端设备执行本说明书上述三维模型场景交互方法部分中描述的根据本发明各种示例性实施方式的步骤。

[0142] 参考图7所示,描述了根据本公开实施方式的用于实现上述方法的程序产品700。根据本发明的实施方式的用于实现上述方法的程序产品,其可以采用便携式紧凑盘只读存储器(CD-ROM)并包括程序代码,并可以在终端设备,例如个人电脑上运行。然而,本发明的程序产品不限于此,在本文件中,可读存储介质可以是任何包含或存储程序的有形介质,该程序可以被指令执行系统、装置或者器件使用或者与其结合使用。

[0143] 程序产品可以采用一个或多个可读介质的任意组合。可读介质可以是可读信号介质或者可读存储介质。可读存储介质例如可以为但不限于电、磁、光、电磁、红外线、或半导体的系统、装置或器件,或者任意以上的组合。可读存储介质的更具体的例子(非穷举的列表)包括:具有一个或多个导线的电连接、便携式盘、硬盘、随机存取存储器(RAM)、只读存储器(ROM)、可擦式可编程只读存储器(EPROM或闪存)、光纤、便携式紧凑盘只读存储器(CD-ROM)、光存储器件、磁存储器件、或者上述的任意合适的组合。

[0144] 计算机可读存储介质可以包括在基带中或者作为载波一部分传播的数据信号,其中承载了可读程序代码。这种传播的数据信号可以采用多种形式,包括但不限于电磁信号、光信号或上述的任意合适的组合。可读存储介质还可以是可读存储介质以外的任何可读介质,该可读介质可以发送、传播或者传输用于由指令执行系统、装置或者器件使用或者与其结合使用的程序。可读存储介质上包含的程序代码可以用任何适当的介质传输,包括但不限于无线、有线、光缆、RF等等,或者上述的任意合适的组合。

[0145] 可以以一种或多种程序设计语言的任意组合来编写用于执行本发明处理的程序代码,程序设计语言包括面向对象的程序设计语言—诸如Java、C++等,还包括常规的过程式程序设计语言—诸如“C”语言或类似的设计语言。程序代码可以完全地在用户计算设备上执行、部分地在用户设备上执行、作为一个独立的软件包执行、部分在用户计算设备上部分在远程计算设备上执行、或者完全在远程计算设备或服务器上执行。在涉及远程计算设备的情形中,远程计算设备可以通过任意种类的网络,包括局域网(LAN)或广域网(WAN),连接到用户计算设备,或者,可以连接到外部计算设备(例如利用因特网服务提供商来通过因特网连接)。

[0146] 综上,本发明的目的在于提供三维模型场景交互方法、系统、设备、装置及存储介质,通过响应于三维模型场景交互请求,获得用户交互数据,并基于用户交互数据运行三维模型渲染引擎,得到三维模型场景的渲染流,将渲染流推送给客户端进行画面展示。在本实施例中,将基于数字孪生应用的三维模型场景渲染进程放到服务器,相比于用户本地设备,服务器能够提供更强大的GPU计算能力,具有更强大的三维模型场景渲染能力,显著提升客户端侧的三维模型场景的画质。同时有效降低对用户端设备的硬件要求,减小其运行压力。

[0147] 以上内容是结合具体的优选实施方式对本发明所作的进一步详细说明,不能认定本发明的具体实施只局限于这些说明。对于本发明所属技术领域的普通技术人员来说,在不脱离本发明构思的前提下,还可以做出若干简单推演或替换,都应当视为属于本发明的保护范围。

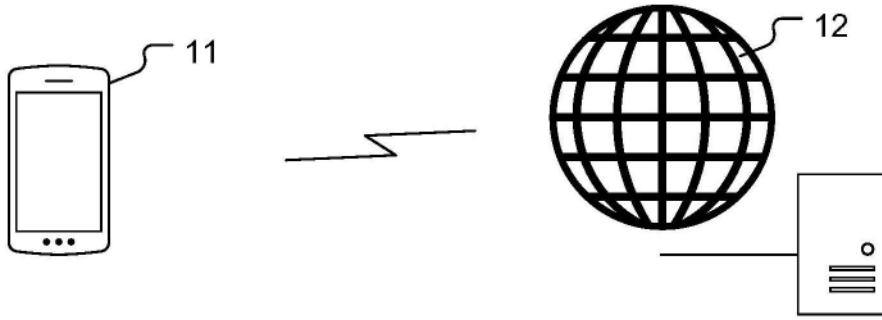


图1

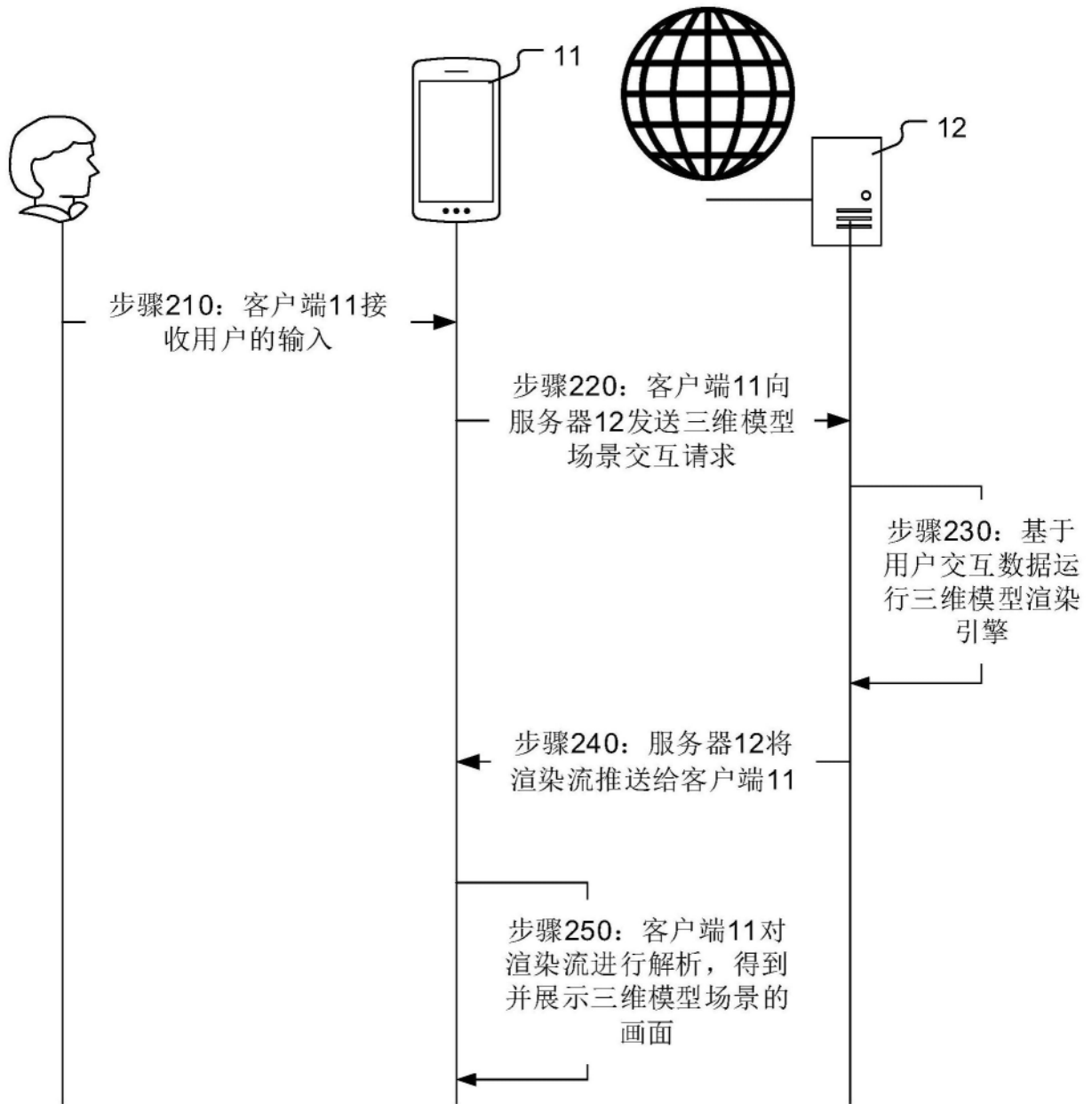


图2

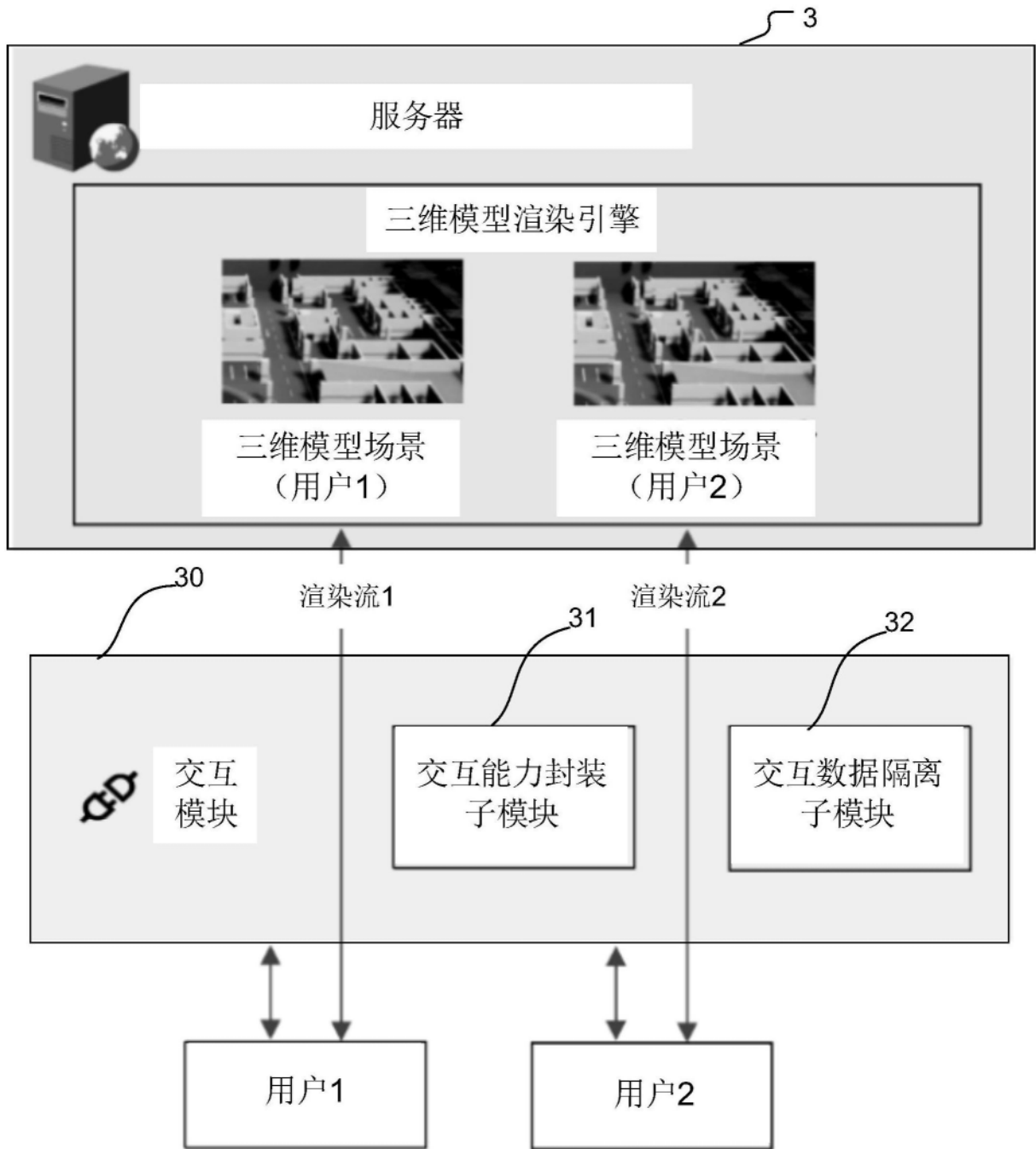


图3

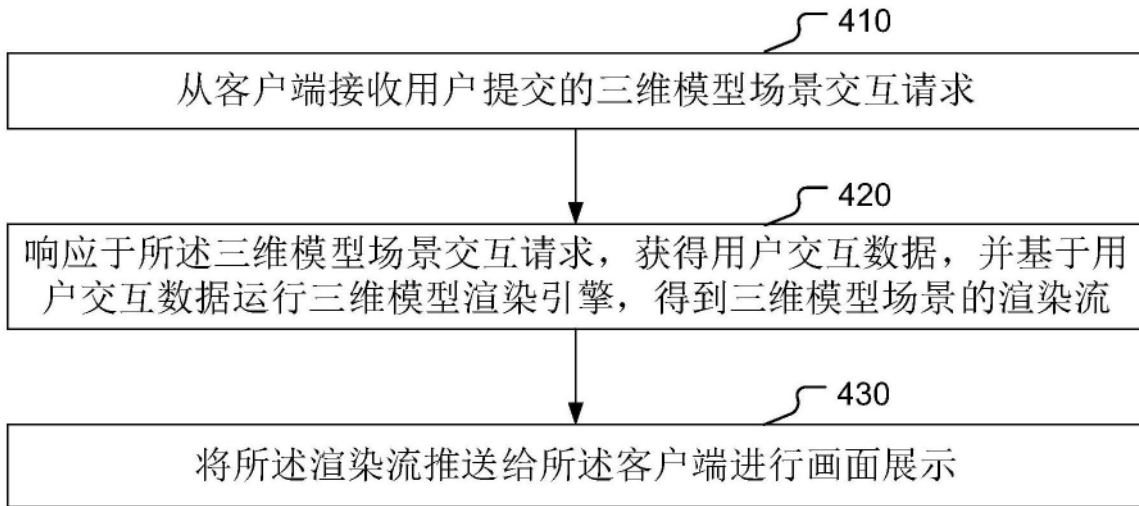


图4

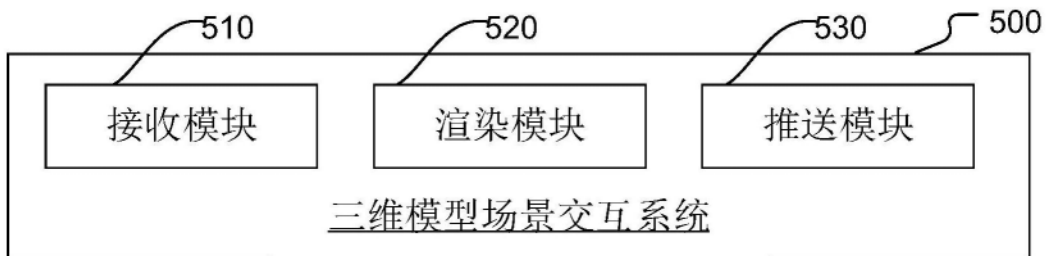


图5

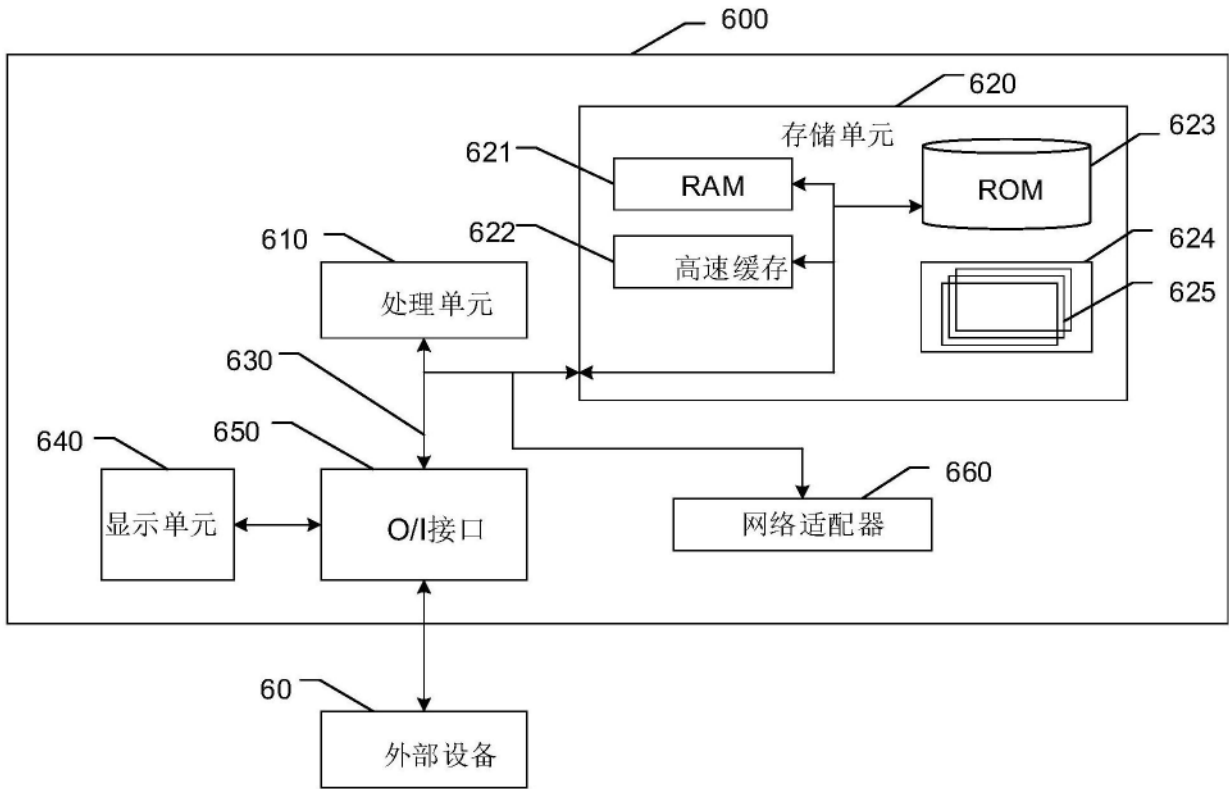


图6

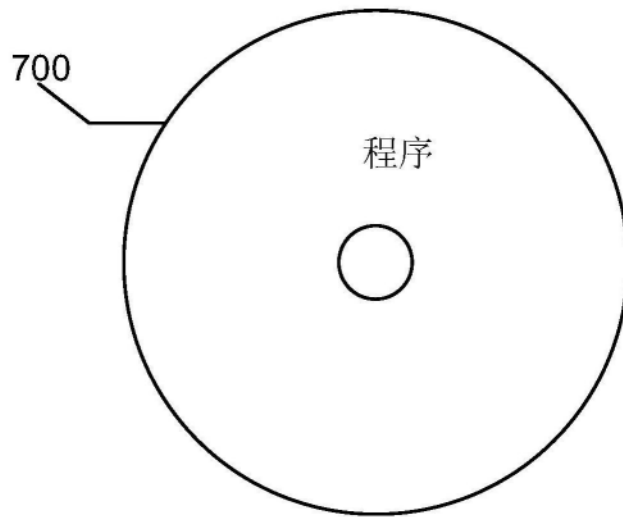


图7