



(12) 发明专利

(10) 授权公告号 CN 110825467 B

(45) 授权公告日 2023. 10. 24

(21) 申请号 201810900517.4

(22) 申请日 2018.08.09

(65) 同一申请的已公布的文献号
申请公布号 CN 110825467 A

(43) 申请公布日 2020.02.21

(73) 专利权人 北京微播视界科技有限公司
地址 100080 北京市海淀区知春路51号4层
408

(72) 发明人 刘昂 游东 冯宇飞

(74) 专利代理机构 北京风雅颂专利代理有限公司 11403
专利代理师 李莎

(51) Int. Cl.
G06F 9/451 (2018.01)

(56) 对比文件

- CN 102323882 A, 2012.01.18
- CN 102867284 A, 2013.01.09
- CN 105096373 A, 2015.11.25
- CN 105678680 A, 2016.06.15
- CN 105741227 A, 2016.07.06
- CN 106504185 A, 2017.03.15
- CN 106575228 A, 2017.04.19
- CN 107735997 A, 2018.02.23
- JP 2016212874 A, 2016.12.15
- US 2014184623 A1, 2014.07.03
- US 2016328816 A1, 2016.11.10
- US 8913068 B1, 2014.12.16
- CN 105867911 A, 2016.08.17

审查员 邓欣

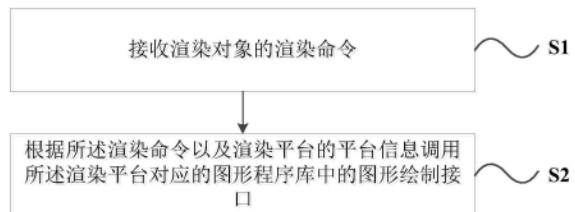
权利要求书1页 说明书12页 附图3页

(54) 发明名称

渲染方法、装置、硬件装置和计算机可读存储介质

(57) 摘要

本公开公开一种渲染方法、装置、硬件装置和计算机可读存储介质。其中,该方法包括接收渲染对象的渲染命令;根据所述渲染命令以及渲染平台的平台信息调用所述渲染平台对应的图形程序库中的图形绘制接口。本公开实施例一方面简化了调用流程,另一方面使得开发者无需关注具体的渲染平台,而专注于如何提升图像显示效果的实现即可,从而节省了开发者的时间和精力,提高了用户体验效果,从而提高了用户体验效果。



1. 一种渲染方法,其特征在于,包括:

接收渲染对象的渲染命令;其中,所述渲染命令为平台无关的渲染命令,所述接收渲染对象的渲染命令进一步包括:接收按照预定的格式生成的所述渲染命令;

根据所述渲染命令以及渲染平台的平台信息调用所述渲染平台对应的图形程序库中的图形绘制接口。

2. 根据权利要求1所述的方法,其特征在于,所述渲染方法还包括:

获取所述渲染平台的平台信息。

3. 根据权利要求1所述的方法,其特征在于,所述预定的格式至少包括:渲染对象几何信息、渲染状态信息以及渲染输出信息;其中,所述渲染状态信息包括渲染所述渲染对象时传递给所述图形绘制接口的参数。

4. 根据权利要求3所述的方法,其特征在于,所述渲染状态信息以下至少一项:纹理信息、深度信息、着色方式和颜色信息。

5. 根据权利要求1所述的方法,其特征在于,根据所述渲染命令以及渲染平台的平台信息调用所述渲染平台对应的图形程序库中的图形绘制接口,包括:

将所述渲染命令放入对应的渲染队列中;

从所述渲染队列中取出所述渲染命令,并从所述渲染平台对应的所述图形程序库中调用所述图形绘制接口执行所述渲染命令。

6. 根据权利要求5所述的方法,其特征在于,将所述渲染命令放入对应的渲染队列中,包括:

将所述渲染命令放入所述渲染命令中的预设渲染状态对应的渲染队列中。

7. 根据权利要求5或6所述的方法,其特征在于,所述渲染队列包括一个或多个渲染命令。

8. 根据权利要求1所述的方法,其特征在于,所述渲染方法还包括:

针对多个渲染平台中的每一个,预先将多个图形绘制接口封装成对应的图形程序库;

其中,所述图形程序库中的图形绘制接口能够在所述图形程序库对应的所述渲染平台上执行。

9. 一种渲染装置,其特征在于,包括:

接收模块,被配置为接收渲染对象的渲染命令;其中,所述渲染命令为平台无关的渲染命令,所述接收渲染对象的渲染命令进一步包括:接收按照预定的格式生成的所述渲染命令;

调用模块,被配置为根据所述渲染命令以及渲染平台的平台信息调用所述渲染平台对应的图形程序库中的图形绘制接口。

10. 一种渲染硬件装置,包括:

存储器,用于存储非暂时性计算机可读指令;以及

处理器,用于运行所述计算机可读指令,使得所述处理器执行时实现根据权利要求1-8中任意一项所述的方法。

11. 一种计算机可读存储介质,用于存储非暂时性计算机可读指令,当所述非暂时性计算机可读指令由计算机执行时,使得所述计算机执行权利要求1-8中任意一项所述的方法。

渲染方法、装置、硬件装置和计算机可读存储介质

技术领域

[0001] 本公开涉及显示技术领域,特别是涉及一种渲染方法、装置、硬件装置和计算机可读存储介质。

背景技术

[0002] 图片渲染是程序可视图片生成的一个重要过程,在游戏等具有图片渲染需求的应用程序中应用广泛;图片渲染主要是对图片渲染对象(如渲染对象)执行图片渲染逻辑实现,图片渲染对象是图片渲染所针对的渲染对象,如按钮控件、美术字控件、背景图等需要渲染的图片对象。

[0003] 随着智能终端的广泛使用,人们在拍照或者录制视频时越来越多的使用特效功能。目前对于特效的渲染,采用具有平台依赖性的图形程序接口,例如PC上使用OpenGL ES来实现渲染对象渲染过程,而如果要在其他平台如IOS上实现时,需要将OpenGL ES进行硬件平台的移植。

[0004] 由于当前流行的智能终端的硬件平台不尽相同,不同的硬件平台所处理的图形应用接口不同,在渲染对象应用开发时较为复杂,资源耗费较多。因此,如何在渲染对象处理方面减轻开发者的负担、节省开发者时间和精力,使得开发者将重点放在应用层面是目前亟待解决的难题。

发明内容

[0005] 根据本公开的一个方面,提供以下技术方案:

[0006] 一种渲染方法,包括:接收渲染对象的渲染命令;根据所述渲染命令以及渲染平台的平台信息调用所述渲染平台对应的图形程序库中的图形绘制接口。

[0007] 进一步地,所述渲染方法还包括:获取所述渲染平台的平台信息。

[0008] 进一步地,接收渲染对象的渲染命令,包括:接收按照预定的格式生成的所述渲染命令。

[0009] 进一步地,所述预定的格式包括:渲染对象几何信息、渲染状态信息以及渲染输出信息;其中,所述渲染状态信息包括渲染所述渲染对象时传递给所述图形绘制接口的参数。

[0010] 进一步地,所述渲染状态信息以下至少一项:纹理信息、深度信息、着色方式和颜色信息。

[0011] 进一步地,根据所述渲染命令以及渲染平台的平台信息调用所述渲染平台对应的图形程序库中的图形绘制接口,包括:将所述渲染命令放入对应的渲染队列中;从所述渲染队列中取出所述渲染命令,并从所述渲染平台对应的所述图形程序库中调用所述图形绘制接口执行所述渲染命令。

[0012] 进一步地,将所述渲染命令放入对应的渲染队列中,包括:将所述渲染命令放入所述渲染命令中的预设渲染状态对应的渲染队列中。

[0013] 进一步地,所述渲染队列包括一个或多个渲染命令。

[0014] 进一步地,所述渲染方法还包括:针对多个渲染平台中的每一个,预先将多个图形绘制接口封装成对应的图形程序库;其中,所述图形程序库中的图形绘制接口能够在所述图形程序库对应的所述渲染平台上执行。

[0015] 进一步地,所述渲染命令为平台无关的渲染命令。

[0016] 根据本公开的另一个方面,还提供以下技术方案:

[0017] 一种渲染装置,包括:接收模块,被配置为接收渲染对象的渲染命令;调用模块,被配置为根据所述渲染命令以及渲染平台的平台信息调用所述渲染平台对应的图形程序库中的图形绘制接口。

[0018] 进一步地,所述渲染装置还包括:获取模块,被配置为获取所述渲染平台的平台信息。

[0019] 进一步地,所述接收模块,包括:接收子模块,被配置为接收按照预定的格式生成的所述渲染命令。

[0020] 进一步地,所述预定的格式包括:渲染对象几何信息、渲染状态信息以及渲染输出信息;其中,所述渲染状态信息包括渲染所述渲染对象时传递给所述图形绘制接口的参数。

[0021] 进一步地,所述渲染状态信息以下至少一项:纹理信息、深度信息、着色方式和颜色信息。

[0022] 进一步地,所述调用模块,包括:第一队列子模块,被配置为将所述渲染命令放入对应的渲染队列中;第二队列子模块,被配置为从所述渲染队列中取出所述渲染命令,并从所述渲染平台对应的所述图形程序库中调用所述图形绘制接口执行所述渲染命令。

[0023] 进一步地,所述第一队列子模块,包括:第三队列子模块,被配置为将所述渲染命令放入所述渲染命令中的预设渲染状态对应的渲染队列中。

[0024] 进一步地,所述渲染队列包括一个或多个渲染命令。

[0025] 进一步地,所述渲染装置还包括:封装模块,被配置为针对多个渲染平台中的每一个,预先将多个图形绘制接口封装成对应的图形程序库;其中,所述图形程序库中的图形绘制接口能够在所述图形程序库对应的所述渲染平台上执行。

[0026] 进一步地,所述渲染命令为平台无关的渲染命令。

[0027] 根据本公开的又一个方面,还提供以下技术方案:

[0028] 一种电子设备,包括:存储器,用于存储非暂时性计算机可读指令;以及处理器,用于运行所述计算机可读指令,使得所述处理器执行时实现上述任一方法所述的步骤。

[0029] 根据本公开的又一个方面,还提供以下技术方案:

[0030] 一种计算机可读存储介质,用于存储非暂时性计算机可读指令,当所述非暂时性计算机可读指令由计算机执行时,使得所述计算机执行上述任一方法中所述的步骤。

[0031] 本公开实施例提供一种渲染方法、装置、硬件装置和计算机可读存储介质。其中,该渲染方法包括:接收渲染对象的渲染命令;根据所述渲染命令以及渲染平台的平台信息调用所述渲染平台对应的图形程序库中的图形绘制接口。本公开实施例在接收到渲染对象的渲染命令后,从渲染平台对应的图形程序库中调用图形绘制接口执行所述渲染命令,一方面简化了调用流程,另一方面使得开发者无需关注具体的渲染平台,而专注于如何提升图像显示效果的实现即可,从而节省了开发者的时间和精力,提高了用户体验效果。

[0032] 上述说明仅是本公开技术方案的概述,为了能更清楚了解本公开的技术手段,而

可依照说明书的内容予以实施,并且为了让本公开的上述和其他目的、特征和优点能够更明显易懂,以下特举较佳实施例,并配合附图,详细说明如下。

附图说明

- [0033] 图1为根据本公开一个实施例的渲染方法的流程示意图;
- [0034] 图2为根据图1所示实施例中步骤S2的流程示意图;
- [0035] 图3为根据本公开一个实施例的渲染装置的结构示意图;
- [0036] 图4为根据图3所示实施例中调用模块32的结构示意图;
- [0037] 图5为根据本公开一个实施例的渲染硬件装置的结构示意图;
- [0038] 图6为根据本公开一个实施例的计算机可读存储介质的结构示意图;
- [0039] 图7为根据本公开一个实施例的渲染终端的结构示意图。

具体实施方式

[0040] 以下通过特定的具体实例说明本公开的实施方式,本领域技术人员可由本说明书所揭露的内容轻易地了解本公开的其他优点与功效。显然,所描述的实施例仅仅是本公开一部分实施例,而不是全部的实施例。本公开还可以通过另外不同的具体实施方式加以实施或应用,本说明书中的各项细节也可以基于不同观点与应用,在没有背离本公开的精神下进行各种修饰或改变。需说明的是,在不冲突的情况下,以下实施例及实施例中的特征可以相互组合。基于本公开中的实施例,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,都属于本公开保护的范围。

[0041] 需要说明的是,下文描述在所附权利要求书的范围内的实施例的各种方面。应显而易见,本文中所描述的方面可体现于广泛多种形式中,且本文中所描述的任何特定结构及/或功能仅为说明性的。基于本公开,所属领域的技术人员应了解,本文中所描述的一个方面可与任何其它方面独立地实施,且可以各种方式组合这些方面中的两者或两者以上。举例来说,可使用本文中所阐述的任何数目个方面来实施设备及/或实践方法。另外,可使用除了本文中所阐述的方面中的一或多者之外的其它结构及/或功能性实施此设备及/或实践此方法。

[0042] 还需要说明的是,以下实施例中所提供的图示仅以示意方式说明本公开的基本构想,图式中仅显示与本公开中有关的组件而非按照实际实施时的组件数目、形状及尺寸绘制,其实际实施时各组件的型态、数量及比例可为一种随意的改变,且其组件布局型态也可能更为复杂。

[0043] 另外,在以下描述中,提供具体细节是为了便于透彻理解实例。然而,所属领域的技术人员将理解,可在没有这些特定细节的情况下实践所述方面。

[0044] 为了解决如何提高用户体验效果的技术问题,本公开实施例提供一种渲染方法。本实施例提供的该渲染方法可以由一渲染装置来执行,该渲染装置可以实现为软件,或者实现为软件和硬件的组合,该渲染装置可以集成设置在图像处理系统中的某设备中,比如图像处理服务器或者图像处理终端设备中。如图1所示,该渲染方法主要包括如下步骤S1至步骤S2。其中:

[0045] 步骤S1:接收渲染对象的渲染命令。

[0046] 其中,渲染对象可以包括需要由图像处理器如GPU进行渲染绘制的显示对象,例如可以是三维场景模型中的人物图像、物品图像、背景图像等等;还可以是网页中的各种显示控件,如,按钮、边框等。渲染命令可以是用户通过系统提供的接口配置而成的,也可以是其他应用程序在绘制三维场景模型或网页时自动生成的。本公开实施例可以被实现为一种渲染引擎,接收进行图像渲染时的渲染命令,渲染命令用于指示对当前渲染对象进行何种渲染操作,进而如何显示该渲染对象;渲染命令可以是平台无关的,也即与底层实现的渲染平台无关。渲染平台包括显示设备上运行的操作系统,平台信息可以是操作系统信息,例如PC系统、IOS系统平台等。渲染的过程实际上是将渲染对象按照渲染命令中指示的显示效果输出,输出的目的地可以是显示屏幕,也可以是缓存(将渲染好的渲染对象暂时缓存起来,待后续显示时机到了之后再显示在屏幕上)。

[0047] 步骤S2:根据所述渲染命令以及渲染平台的平台信息调用所述渲染平台对应的图形程序库中的图形绘制接口。

[0048] 其中,渲染平台是用于对渲染对象进行渲染并进行显示的操作系统平台,主要依赖设备所安装的操作系统、CPU和/或GPU。在渲染的过程中,有些图形绘制接口需要通过操作系统提供的驱动程序直接操作CPU,在配置有GPU的硬件设备中,CPU可以通过调用GPU实现图像渲染的加速。因此,通过执行渲染命令而实现渲染对象的渲染过程中,必不可少的存在图形绘制接口会调用操作系统的驱动程序而直接操作CPU和/或GPU的情形,而对于同一图像效果的实现,如果操作系统不同,则需要编写不同的图形绘制接口。因此,本公开中在接收到渲染命令后,针对不同的渲染平台,调用不同的图形程序库中的图形绘制接口来完成渲染命令。

[0049] 为实现这一功能,本公开实施例中的渲染命令首先可以是平台无关的,也即在配置渲染命令时无需考虑操作系统平台,而只需要专注如何实现图像的渲染效果即可;其次,本公开可以预先针对不同的渲染平台封装成不同的图形程序库,以便在执行渲染命令时,根据当前要进行图像渲染(也即绘制并显示图像)的渲染平台找到对应的图形程序库,进而根据渲染命令中的信息调用该图形程序库中的图形绘制接口完成图像的绘制以及效果的显示。图形程序库中的图形绘制接口是平台相关的,也即可以直接操作硬件平台的程序软件。

[0050] 为了便于理解,下面以具体实施例对上述图像的渲染过程进行详细说明。

[0051] 在一实施例中,绘制三维场景模型的过程中,需要对三维场景模型中的人脸叠加上特效显示,因此三维场景模型的开发者配置了相关的渲染命令,期中的渲染对象为人脸图像,特效为脸蛋叠加红晕效果。而当前进行渲染的系统平台为IOS平台。那么在接收到该渲染命令以后,从IOS平台对应的图形程序库中调用红晕效果对应的图形绘制接口,并完成渲染命令,实现人脸图像的红晕特效显示。需要说明的是,在此仅仅是示例性的说明本公开实现的步骤,而实际应用中渲染命令中还会包括人脸的其他绘制信息,例如纹理、颜色、几何形状等等也都需要调用相应的图形绘制接口来完成。

[0052] 本公开实施例提供一种渲染方法、装置、硬件装置和计算机可读存储介质。其中,该渲染方法包括:接收渲染对象的渲染命令;根据所述渲染命令以及渲染平台的平台信息调用所述渲染平台对应的图形程序库中的图形绘制接口。本公开实施例在接收到渲染对象的渲染命令后,从渲染平台对应的图形程序库中调用图形绘制接口执行所述渲染命令,一

方面简化了调用流程,另一方面使得开发者无需关注具体的渲染平台,而专注于如何提升图像显示效果的实现即可,从而节省了开发者的时间和精力,提高了用户体验效果。

[0053] 在一个可选的实施例中,所述渲染方法,还可以包括:

[0054] 获取所述渲染平台的平台信息。

[0055] 该可选的实施例中,在三维场景模型的开发或者网页的开发中,所形成的渲染命令可以是平台无关的,也即其中并没有依据不同的操作系统平台配置特定的信息,而是可以在任何系统平台上执行的渲染命令。但是实际应用中,如生成三维场景模型时,三维场景模型的软件包需要在具体的操作系统平台上运行,而由于三维场景模型的渲染需要调用依赖系统平台的图形绘制接口来实现,因此可以预先获得当前生成该三维场景模型系统平台信息,也即执行渲染命令的渲染平台的平台信息。渲染平台的平台信息至少包括图像设备的操作系统信息。例如在苹果手机上生成三维场景模型时,所获得的平台信息为IOS系统,而在安卓手机上生成三维场景模型时,所获得的平台信息为安卓系统,在PC机上生成三维场景模型时,所获得的平台信息为Windows操作系统等。

[0056] 在一实施例中,渲染平台的平台信息可以在本公开的实现在设备上安装注册时获得。在其他实施例中,如果实现本公开方法的系统为分布式分发系统,也可以在执行步骤S2即根据所述渲染命令以及渲染平台的平台信息调用所述渲染平台对应的图形程序库中的图形绘制接口之前获得;具体可根据实际情况设定,在此不做限制。

[0057] 在一个可选的实施例中,步骤S1即接收渲染对象的渲染命令,可以包括:

[0058] 接收按照预定的格式生成的所述渲染命令。

[0059] 该可选的实施例中,所接收到的渲染命令是具有预定的格式的。也即生成渲染命令的用户或者程序,都可以按照本公开提供的预定格式生成渲染命令。这是因为本公开实施例在执行渲染命令时,需要根据渲染命令中的信息调用相应的图形绘制接口完成渲染对象的渲染,而由于渲染需求各种各样,且随着技术的发展在不断的更新换代,如果渲染命令不是按照统一格式定义的,很难通过渲染命令的解析确定渲染的各种属性。因此本公开实施例预先定义好渲染命令的格式,生成渲染命令的一方可以按照该预定的格式生成渲染命令,本公开实施例可以从具有预定格式的渲染命令中提取相应的信息,进而执行渲染命令。例如,该预定的格式可以包括渲染命令相关的各个字段,每个字段都有固定的位置,在生成渲染命令时将相应的字段内容填充到其对应的位置处;本公开实施例在接收到渲染命令以后,从固定位置处找到相应字段的内容并调用相应的图形绘制接口;如渲染命令最开始处填充渲染对象标识,之后是渲染对象的纹理等信息。通过这种方式,既可以实现渲染命令的平台无关性,也可以方便地实现渲染命令种类的扩展更新。

[0060] 在一个可选的实施例中,所述预定的格式至少包括:渲染对象几何信息、渲染状态信息以及渲染输出信息;其中,所述渲染状态信息包括渲染所述渲染对象时传递给所述图形绘制接口的参数。

[0061] 该可选的实施例中,将渲染命令的预定格式定义为包括:渲染对象几何信息、渲染状态信息和渲染输出信息。几何信息为渲染状态的几何形状等特征信息,而状态信息为进行渲染时传递给图形绘制接口的具体绘制参数,如渲染对象的属性(颜色、特效)等;渲染输出信息包括将渲染后的渲染对象输出至的目的地,包括当前显示屏幕以及缓存等。如果当前是在实时渲染,则将渲染后的对象实时显示在屏幕上,而如果是非实时渲染,则将渲染后

的对象存储在缓存中,在显示时间到了后,再进行显示。

[0062] 在一个可选的实施例中,所述渲染状态信息以下至少一项:纹理信息、深度信息、着色方式和颜色信息。

[0063] 渲染状态信息定义了对渲染对象进行渲染所必须的一些参数,例如纹理信息、深度信息、着色方式和颜色信息。在渲染过程中,可以通过调用相应的图形绘制接口绘制渲染对象。深度信息用于表示当前渲染对象在三维场景模型中所处的深度位置,跟渲染顺序相关(深度信息较高的渲染对象先进行渲染,而深度信息较低的对象后进行渲染,如果位置重合的话,先渲染的对象会被后渲染的对象所覆盖),着色方式包括几何着色、顶点着色、片段着色等;颜色信息包括环境光颜色、漫反射颜色、高光颜色。这些是图像渲染所需要的信息,具体可参见已有技术中的图像渲染过程,在此不再详述。

[0064] 在一个可选的实施例中,如图2所示,步骤S2即根据所述渲染命令以及渲染平台的平台信息调用所述渲染平台对应的图形程序库中的图形绘制接口,可以包括:

[0065] S21.将所述渲染命令放入对应的渲染队列中;

[0066] S22.从所述渲染队列中取出所述渲染命令,并从所述渲染平台对应的所述图形程序库中调用所述图形绘制接口完成所述渲染命令。

[0067] 该可选的实现方式中,预先可以形成多个渲染队列,不同类型的渲染命令放入不同的渲染队列,例如渲染对象是透明的和不透明放入不同的渲染队列中。在一实施例中,渲染队列中包括一个或多个渲染命令。在有多个渲染对象等待被渲染时,渲染队列中可以包括多个渲染命令,多个渲染命令按照渲染的顺序放在渲染队列中,先进入渲染队列的渲染命令先执行,后进入渲染队列的渲染命令后执行。在执行到当前渲染命令时,从渲染队列取出该当前渲染命令,并从渲染平台对应的图形程序库中调用图形绘制接口,完成该当前的渲染命令。在一些实施例中,由于同类型的渲染命令所使用的基础信息相同,为了加速渲染过程,还可以将多个在同一渲染队列中的命令进行合并,以批量执行渲染命令。

[0068] 在一个可选的实施例中,步骤S21即将所述渲染命令放入对应的渲染队列中,可以包括:

[0069] 将所述渲染命令放入所述渲染命令中的预设渲染状态对应的渲染队列中。

[0070] 该可选的实施例中,预设渲染状态可以是预先设定的多种不同的渲染状态,并且每种不同的渲染状态对应设置有一个渲染队列。当渲染命令中的渲染状态与预设渲染状态一致时,将该渲染命令放入该预设渲染状态对应的渲染队列中,这样可以将渲染状态一致的多个渲染命令放入同一个渲染队列中。由于渲染状态一致的多个渲染命令,所使用的基础信息是一致的,因此可以进行批量操作,或者可以分配给同一个GPU进行处理,这样能够节省资源,加快渲染速度,提高实时性。

[0071] 在一个可选的实施例中,所述渲染方法还可以包括:

[0072] 针对多个渲染平台中的每一个,预先将多个图形绘制接口封装成对应的图形程序库;其中,所述图形程序库中的图形绘制接口能够在所述图形程序库对应的所述渲染平台上执行。

[0073] 该可选的实施例中,为了能够在不同的渲染平台上执行接收到的渲染命令,使得渲染命令与渲染平台无关,可以预先针对多个渲染平台中的每一个,将直接操作CPU的图形绘制接口(由于直接操作CPU的图形绘制接口需要调用操作系统中的驱动程序,因此图形绘

制接口是系统相关的)封装成图形程序库的形式(如SDK库),在执行渲染命令时,只需要调用该渲染平台对应的图形程序库中的图形绘制接口即可。例如,预先封装形成针对IOS平台的图形程序库和针对PC平台的图形程序库,三维场景模型A的生成代码在IOS平台上执行时,调用针对IOS平台对应的图形程序库中的图形绘制接口执行渲染命令,而同样的三维场景模型A的生成代码在PC平台上执行时,调用针对PC平台对应的图形程序库中的图形绘制接口执行渲染命令。

[0074] 在渲染平台上安装执行本公开方法的系统时,可以依据渲染平台的平台信息安装对应的图形程序库,并在执行渲染命令时也调用该图形程序库中的图形绘制接口。通过这种方式,可以使得三维场景模型的开发者们专注于特效等显示效果的开发,而无需考虑渲染平台相关性,给予了开发者们更大的发挥余地,提高了用户体验。

[0075] 在上文中,虽然按照上述的顺序描述了上述方法实施例中的各个步骤,本领域技术人员应清楚,本公开实施例中的步骤并不必然按照上述顺序执行,其也可以倒序、并行、交叉等其他顺序执行,而且,在上述步骤的基础上,本领域技术人员也可以再加入其他步骤,这些明显变型或等同替换的方式也应包含在本公开的保护范围之内,在此不再赘述。

[0076] 下面为本公开装置实施例,本公开装置实施例可用于执行本公开方法实施例实现的步骤,为了便于说明,仅示出了与本公开实施例相关的部分,具体技术细节未揭示的,请参照本公开方法实施例。

[0077] 为了解决如何提高用户体验效果的技术问题,本公开实施例提供一种渲染装置。该装置可以执行上述渲染方法实施例中所述的步骤。如图3所示,该装置主要包括:接收模块31和调用模块32。其中,接收模块31被配置为接收渲染对象的渲染命令;调用模块32被配置为根据所述渲染命令以及渲染平台的平台信息调用所述渲染平台对应的图形程序库中的图形绘制接口。

[0078] 本公开实施例中,渲染对象可以包括需要由图像处理器如GPU进行渲染绘制的显示对象,例如可以是三维场景模型中的人物图像、物品图像、背景图像等等;还可以是网页中的各种显示控件,按钮、边框之类的。渲染命令可以是用户通过系统提供的接口配置而成的,也可以是其他应用程序在绘制三维场景模型或网页时自动生成的。本公开实施例可以被实现为一种渲染引擎,接收模块31接收进行图像渲染时的渲染命令,渲染命令用于指示对当前渲染对象进行何种渲染操作,进而如何显示该渲染对象;渲染命令可以是平台无关的,也即与底层实现的渲染平台无关。

[0079] 渲染平台是用于对渲染对象进行渲染并进行显示的操作系统平台,主要依赖设备所安装的操作系统、CPU和/或GPU。在渲染的过程中,有些图形绘制接口需要通过操作系统提供的驱动程序直接操作CPU,在配置有GPU的硬件设备中,CPU可以通过调用GPU实现图像渲染的加速。因此,通过执行渲染命令而实现渲染对象的渲染过程中,必不可少的存在图形绘制接口会调用操作系统的驱动程序而直接操作CPU和/或GPU的情形,而对于同一图像效果的实现,如果操作系统不同,则需要编写不同的图形绘制接口。因此,接收模块31接收到渲染命令后,调用模块32针对不同的渲染平台,调用不同的图形程序库中的图形绘制接口来完成渲染命令。

[0080] 上述显示装置与上述图1所示实施例中的渲染方法对应一致,具体细节可参考上述对显示方法的描述,在此不再赘述。

[0081] 在一个可选的实施例中,所述渲染装置还可以包括:获取模块,被配置为获取所述渲染平台的平台信息。

[0082] 该可选的实施例中,在三维场景模型的开发或者网页的开发中,所形成的渲染命令可以是平台无关的,也即其中并没有依据不同的操作系统平台配置特定的信息,而是可以在任何系统平台上执行的渲染命令。但是实际应用中,如生成三维场景模型时,三维场景模型的软件包需要在具体的操作系统平台上运行,而由于三维场景模型的渲染需要调用依赖系统平台的图形绘制接口来实现,因此:获取模块可以预先获得当前生成该三维场景模型系统平台信息,也即执行渲染命令的渲染平台的平台信息。渲染平台的平台信息至少包括图像设备的操作系统信息。例如在苹果手机上生成三维场景模型时,所获得的平台信息为IOS系统,而在安卓手机上生成三维场景模型时,所获得的平台信息为安卓系统,在PC机上生成三维场景模型时,所获得的平台信息为Windows操作系统等。

[0083] 在一实施例中,获取模块可以在本公开的实现在设备上安装注册时获得渲染平台的平台信息。在其他实施例中,如果实现本公开方法的系统为分布式分发系统,也可以在执行步骤S2即根据所述渲染命令以及渲染平台的平台信息调用所述渲染平台对应的图形程序库中的图形绘制接口之前获得;具体可根据实际情况设定,在此不做限制。

[0084] 在一个可选的实施例中,接收子模块,被配置为接收按照预定的格式生成的所述渲染命令。

[0085] 该可选的实施例中,接收子模块所接收到的渲染命令是具有预定的格式的。也即生成渲染命令的用户或者程序,都可以按照本公开提供的预定格式生成渲染命令。这是因为本公开实施例在执行渲染命令时,需要根据渲染命令中的信息调用相应的图形绘制接口完成渲染对象的渲染,而由于渲染需求各种各样,且随着技术的发展在不断的更新换代,如果渲染命令不是按照统一格式定义的,很难通过渲染命令的解析确定渲染的各种属性。因此本公开实施例预先定义好渲染命令的格式,生成渲染命令的一方可以按照该预定的格式生成渲染命令,本公开实施例可以从具有预定格式的渲染命令中提取相应的信息,进而执行渲染命令。例如,该预定的格式可以包括渲染命令相关的各个字段,每个字段都有固定的位置,在生成渲染命令时将相应的字段内容填充到其对应的位置处;本公开实施例在接收到渲染命令以后,从固定位置处找到相应字段的内容并调用相应的图形绘制接口;如渲染命令最开始处填充渲染对象标识,之后是渲染对象的纹理等信息。通过这种方式,既可以实现渲染命令的平台无关性,也可以方便地实现渲染命令种类的扩展更新。

[0086] 在一个可选的实施例中,所述预定的格式至少包括:渲染对象几何信息、渲染状态信息以及渲染输出信息;其中,所述渲染状态信息包括渲染所述渲染对象时传递给所述图形绘制接口的参数。

[0087] 该可选的实施例中,将渲染命令的预定格式定义为包括:渲染对象几何信息、渲染状态信息和渲染输出信息。几何信息为渲染状态的几何形状等特征信息,而状态信息为进行渲染时传递给图形绘制接口的具体绘制参数,如渲染对象的属性(颜色、特效)等;渲染输出信息包括将渲染后的渲染对象输出至的目的地,包括当前显示屏幕以及缓存等。如果当前是在实时渲染,则将渲染后的对象实时显示在屏幕上,而如果是非实时渲染,则将渲染后的对象存储在缓存中,在显示时间到了后,再进行显示。

[0088] 在一个可选的实施例中,所述渲染状态信息以下至少一项:纹理信息、深度信息、

着色方式和颜色信息。

[0089] 渲染状态信息定义了对渲染对象进行渲染所必须的一些参数,例如纹理信息、深度信息、着色方式和颜色信息。在渲染过程中,可以通过调用相应的图形绘制接口绘制渲染对象。深度信息用于表示当前渲染对象在三维场景模型中所处的深度位置,跟渲染顺序相关(深度信息较高的渲染对象先进行渲染,而深度信息较低的对象后进行渲染,如果位置重合的话,先渲染的对象会被后渲染的对象所覆盖),着色方式包括几何着色、顶点着色、片段着色等;颜色信息包括环境光颜色、漫反射颜色、高光颜色。这些是图像渲染所需要的信息,具体可参见已有技术中的图像渲染过程,在此不再详述。

[0090] 在一个可选的实施例中,如图4所示,所述调用模块32,包括:

[0091] 第一队列子模块41,被配置为将所述渲染命令放入对应的渲染队列中;

[0092] 第二队列子模块42,被配置为从所述渲染队列中取出所述渲染命令,并从所述渲染平台对应的所述图形程序库中调用所述图形绘制接口执行所述渲染命令。

[0093] 该可选的实现方式中,预先可以形成多个渲染队列,第一队列子模块41将不同类型的渲染命令放入不同的渲染队列,例如渲染对象是透明的和不透明放入不同的渲染队列中。在一实施例中,渲染队列中包括一个或多个渲染命令。在有多个渲染对象等待被渲染时,渲染队列中可以包括多个渲染命令,多个渲染命令按照渲染的顺序放在渲染队列中,先进入渲染队列的渲染命令先执行,后进入渲染队列的渲染命令后执行。在执行到当前渲染命令时,第二队列子模块42从渲染队列取出该当前渲染命令,并从渲染平台对应的图形程序库中调用图形绘制接口,完成该当前的渲染命令。在一些实施例中,由于同类型的渲染命令所使用的基础信息相同,为了加速渲染过程,还可以将多个在同一渲染队列中的命令进行合并,以批量执行渲染命令。

[0094] 在一个可选的实施例中,所述第一队列子模块41,包括:第三队列子模块,被配置为将所述渲染命令放入所述渲染命令中的预设渲染状态对应的渲染队列中。

[0095] 该可选的实施例中,预设渲染状态可以是预先设定的多种不同的渲染状态,并且每种不同的渲染状态对应设置有一个渲染队列。当渲染命令中的渲染状态与预设渲染状态一致时,将该渲染命令放入该预设渲染状态对应的渲染队列中,这样可以将渲染状态一致的多个渲染命令放入同一个渲染队列中。由于渲染状态一致的多个渲染命令,所使用的基础信息是一致的,因此可以进行批量操作,或者可以分配给同一个GPU进行处理,这样能够节省资源,加快渲染速度,提高实时性。

[0096] 在一个可选的实施例中,所述渲染装置还可以包括:封装模块,被配置为针对多个渲染平台中的每一个,预先将多个图形绘制接口封装成对应的图形程序库;其中,所述图形程序库中的图形绘制接口能够在所述图形程序库对应的所述渲染平台上执行。

[0097] 该可选的实施例中,为了能够在不同的渲染平台上执行接收到的渲染命令,使得渲染命令与渲染平台无关,封装模块可以预先针对多个渲染平台中的每一个,将直接操作CPU的图形绘制接口(由于直接操作CPU的图形绘制接口需要调用操作系统中的驱动程序,因此图形绘制接口是系统相关的)封装成图形程序库的形式(如SDK库),在执行渲染命令时,只需要调用该渲染平台对应的图形程序库中的图形绘制接口即可。例如,预先封装形成针对IOS平台的图形程序库和针对PC平台的图形程序库,三维场景模型A的生成代码在IOS平台上执行时,调用针对IOS平台对应的图形程序库中的图形绘制接口执行渲染命令,而同

样的三维场景模型A的生成代码在PC平台上执行时,调用针对PC平台对应的图形程序库中的图形绘制接口执行渲染命令。

[0098] 在渲染平台上安装执行本公开方法的系统时,可以依据渲染平台的平台信息安装对应的图形程序库,并在执行渲染命令时也调用该图形程序库中的图形绘制接口。通过这种方式,可以使得三维场景模型的开发者们专注于特效等显示效果的开发,而无需考虑渲染平台相关性,给予了开发者们更大的发挥余地,提高了用户体验。

[0099] 有关显示装置实施例的工作原理、实现的技术效果等详细说明可以参考前述显示方法实施例中的相关说明,在此不再赘述。

[0100] 图5是图示根据本公开的实施例的渲染硬件装置的硬件框图。如图5所示,根据本公开实施例的渲染硬件装置50包括存储器51和处理器52。

[0101] 该存储器51用于存储非暂时性计算机可读指令。具体地,存储器51可以包括一个或多个计算机程序产品,该计算机程序产品可以包括各种形式的计算机可读存储介质,例如易失性存储器和/或非易失性存储器。该易失性存储器例如可以包括随机存取存储器(RAM)和/或高速缓冲存储器(cache)等。该非易失性存储器例如可以包括只读存储器(ROM)、硬盘、闪存等。

[0102] 该处理器52可以是中央处理单元(CPU)或者具有数据处理能力和/或指令执行能力的其它形式的处理单元,并且可以控制渲染硬件装置50中的其它组件以执行期望的功能。在本公开的一个实施例中,该处理器52用于运行该存储器51中存储的该计算机可读指令,使得该渲染硬件装置50执行前述的本公开各实施例的显示方法的全部或部分步骤。

[0103] 本领域技术人员应能理解,为了解决如何获得良好用户体验效果的技术问题,本实施例中也可以包括诸如通信总线、接口等公知的结构,这些公知的结构也应包含在本公开的保护范围之内。

[0104] 有关本实施例的详细说明可以参考前述各实施例中的相应说明,在此不再赘述。

[0105] 图6是图示根据本公开的实施例的计算机可读存储介质的示意图。如图6所示,根据本公开实施例的计算机可读存储介质60,其上存储有非暂时性计算机可读指令61。当该非暂时性计算机可读指令61由处理器运行时,执行前述的本公开各实施例的渲染方法的全部或部分步骤。

[0106] 上述计算机可读存储介质60包括但不限于:光存储介质(例如:CD-ROM和DVD)、磁光存储介质(例如:MO)、磁存储介质(例如:磁带或移动硬盘)、具有内置的可重写非易失性存储器的媒体(例如:存储卡)和具有内置ROM的媒体(例如:ROM盒)。

[0107] 有关本实施例的详细说明可以参考前述各实施例中的相应说明,在此不再赘述。

[0108] 图7是图示根据本公开实施例的渲染终端的硬件结构示意图。如图7所示,该渲染终端70包括上述显示装置实施例。

[0109] 该终端设备可以以各种形式来实施,本公开中的终端设备可以包括但不限于诸如移动电话、智能电话、笔记本电脑、数字广播接收器、PDA(个人数字助理)、PAD(平板电脑)、PMP(便携式多媒体播放器)、导航装置、车载终端设备、车载显示终端、车载电子后视镜等等的移动终端设备以及诸如数字TV、台式计算机等等的固定终端设备。

[0110] 作为等同替换的实施方式,该终端还可以包括其他组件。如图7所示,该渲染终端70可以包括电源单元71、无线通信单元72、A/V(音频/视频)输入单元73、用户输入单元74、

感测单元75、接口单元76、控制器77、输出单元78和存储单元79等等。图7示出了具有各种组件的终端,但是应理解的是,并不要求实施所有示出的组件,也可以替代地实施更多或更少的组件。

[0111] 其中,无线通信单元72允许终端70与无线通信系统或网络之间的无线电通信。A/V输入单元73用于接收音频或视频信号。用户输入单元74可以根据用户输入的命令生成键输入数据以控制终端设备的各种操作。感测单元75检测终端70的当前状态、终端70的位置、用户对于终端70的触摸输入的有无、终端70的取向、终端70的加速或减速移动和方向等等,并且生成用于控制终端160的操作的命令或信号。接口单元76用作至少一个外部装置与终端70连接可以通过的接口。输出单元78被构造为以视觉、音频和/或触觉方式提供输出信号。存储单元79可以存储由控制器77执行的处理和控制的软件程序等等,或者可以暂时地存储已经输出或将要输出的数据。存储单元79可以包括至少一种类型的存储介质。而且,终端70可以与通过网络连接执行存储单元79的存储功能的网络存储装置协作。控制器77通常控制终端设备的总体操作。另外,控制器77可以包括用于再现或回放多媒体数据的多媒体模块。控制器77可以执行模式识别处理,以将在触摸屏上执行的手写输入或者图片绘制输入识别为字符或渲染对象。电源单元71在控制器77的控制下接收外部电力或内部电力并且提供操作各元件和组件所需的适当的电力。

[0112] 本公开提出的显示方法的各种实施方式可以以使用例如计算机软件、硬件或其任何组合的计算机可读介质来实施。对于硬件实施,本公开提出的显示方法的各种实施方式可以通过使用特定用途集成电路(ASIC)、数字信号处理器(DSP)、数字信号处理装置(DSPD)、可编程逻辑装置(PLD)、现场可编程门阵列(FPGA)、处理器、控制器、微控制器、微处理器、被设计为执行这里描述的功能的电子单元中的至少一种来实施,在一些情况下,本公开提出的显示方法的各种实施方式可以在控制器77中实施。对于软件实施,本公开提出的显示方法的各种实施方式可以与允许执行至少一种功能或操作的单独的软件模块来实施。软件代码可以由以任何适当的编程语言编写的软件应用程序(或程序)来实施,软件代码可以存储在存储单元79中并且由控制器77执行。

[0113] 有关本实施例的详细说明可以参考前述各实施例中的相应说明,在此不再赘述。

[0114] 以上结合具体实施例描述了本公开的基本原理,但是,需要指出的是,在本公开中提及的优点、优势、效果等仅是示例而非限制,不能认为这些优点、优势、效果等是本公开的各个实施例必须具备的。另外,上述公开的具体细节仅是为了示例的作用和便于理解的作用,而非限制,上述细节并不限制本公开为必须采用上述具体的细节来实现。

[0115] 本公开中涉及的器件、装置、设备、系统的方框图仅作为例示性的例子并且不意图要求或暗示必须按照方框图示出的方式进行连接、布置、配置。如本领域技术人员将认识到的,可以按任意方式连接、布置、配置这些器件、装置、设备、系统。诸如“包括”、“包含”、“具有”等等的词语是开放性词汇,指“包括但不限于”,且可与其互换使用。这里所使用的词汇“或”和“和”指词汇“和/或”,且可与其互换使用,除非上下文明确指示不是如此。这里所使用的词汇“诸如”指词组“诸如但不限于”,且可与其互换使用。

[0116] 另外,如在此使用的,在以“至少一个”开始的项的列举中使用的“或”指示分离的列举,以便例如“A、B或C的至少一个”的列举意味着A或B或C,或AB或AC或BC,或ABC(即A和B和C)。此外,措辞“示例的”不意味着描述的例子是优选的或者比其他例子更好。

[0117] 还需要指出的是,在本公开的系统和方法中,各部件或各步骤是可以分解和/或重新组合的。这些分解和/或重新组合应视为本公开的等效方案。

[0118] 可以不脱离由所附权利要求定义的教导的技术而进行对在此所述的技术的各种改变、替换和更改。此外,本公开的权利要求的范围不限于以上所述的处理、机器、制造、事件的组成、手段、方法和动作的具体方面。可以利用与在此所述的相应方面进行基本相同的功能或者实现基本相同的结果的当前存在的或者稍后要开发的处理、机器、制造、事件的组成、手段、方法或动作。因而,所附权利要求包括在其范围内的这样的处理、机器、制造、事件的组成、手段、方法或动作。

[0119] 提供所公开的方面的以上描述以使本领域的任何技术人员能够做出或者使用本公开。对这些方面的各种修改对于本领域技术人员而言是非常显而易见的,并且在此定义的一般原理可以应用于其他方面而不脱离本公开的范围。因此,本公开不意图被限制到在此示出的方面,而是按照与在此公开的原理和新颖的特征一致的最宽范围。

[0120] 为了例示和描述的目的已经给出了以上描述。此外,此描述不意图将本公开的实施例限制到在此公开的形式。尽管以上已经讨论了多个示例方面和实施例,但是本领域技术人员将认识到其某些变型、修改、改变、添加和子组合。

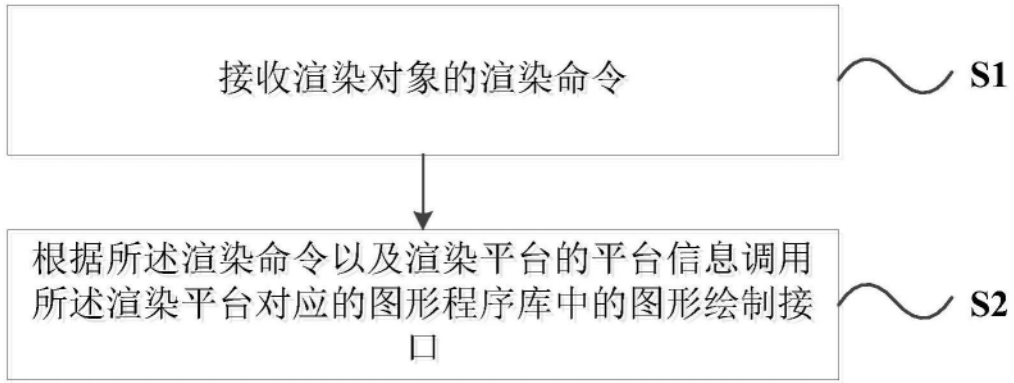


图1

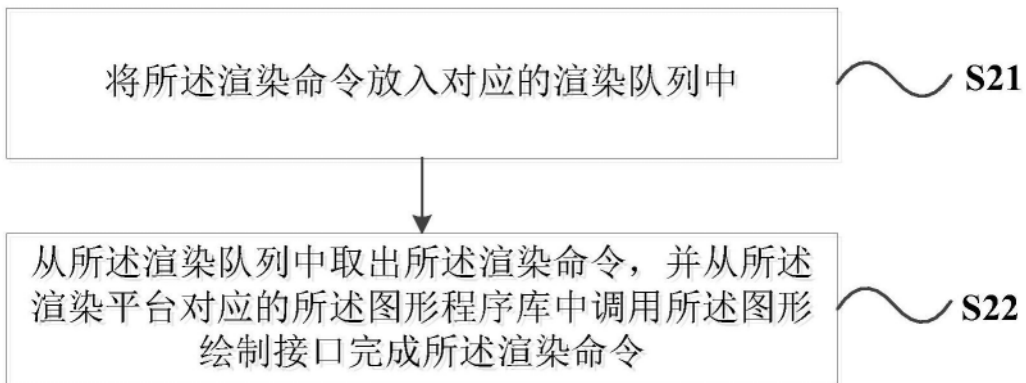


图2

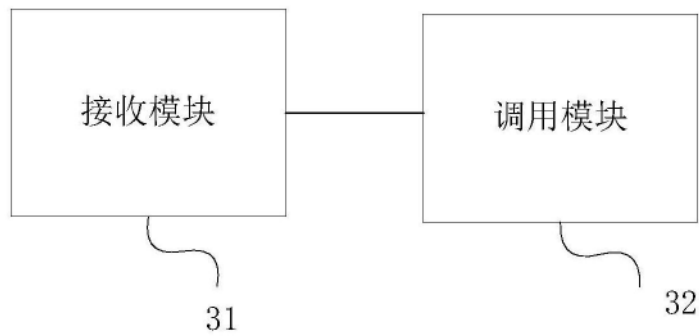


图3

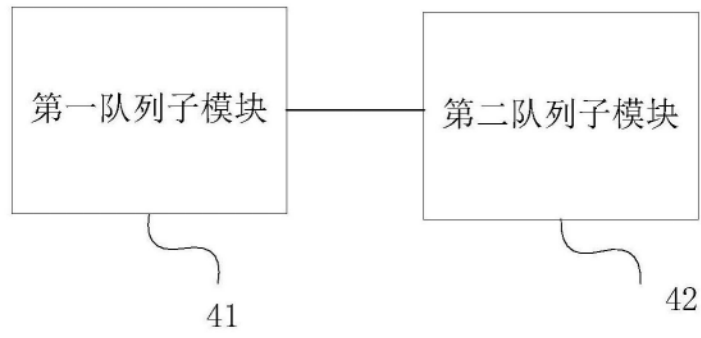


图4

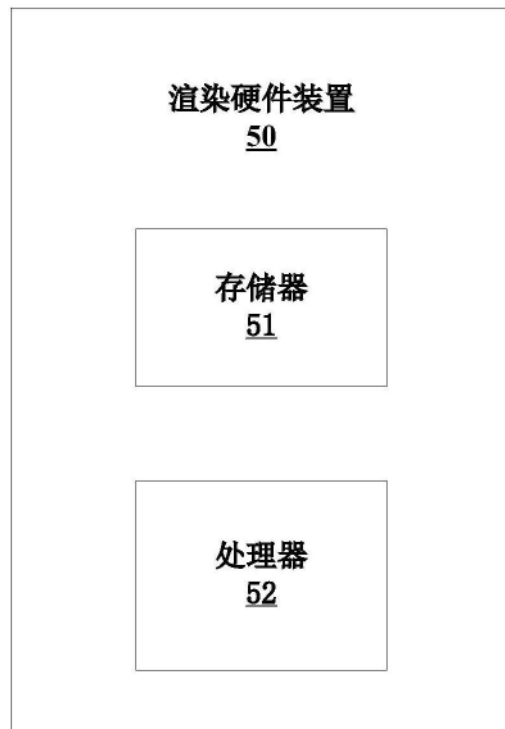


图5

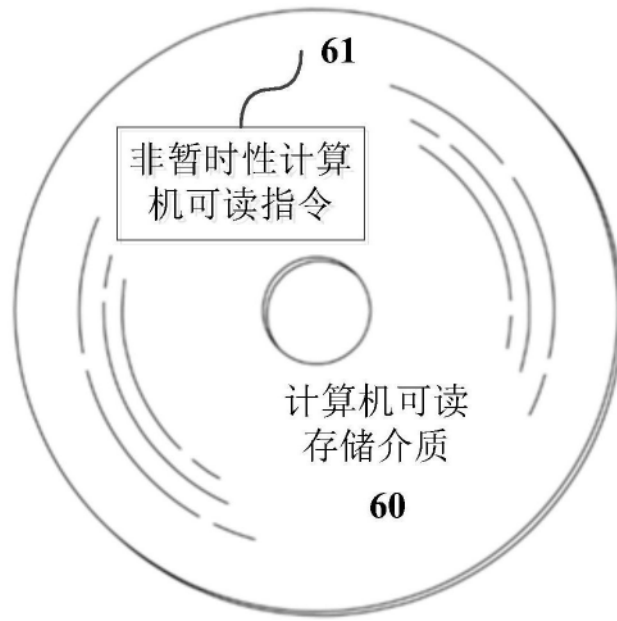


图6

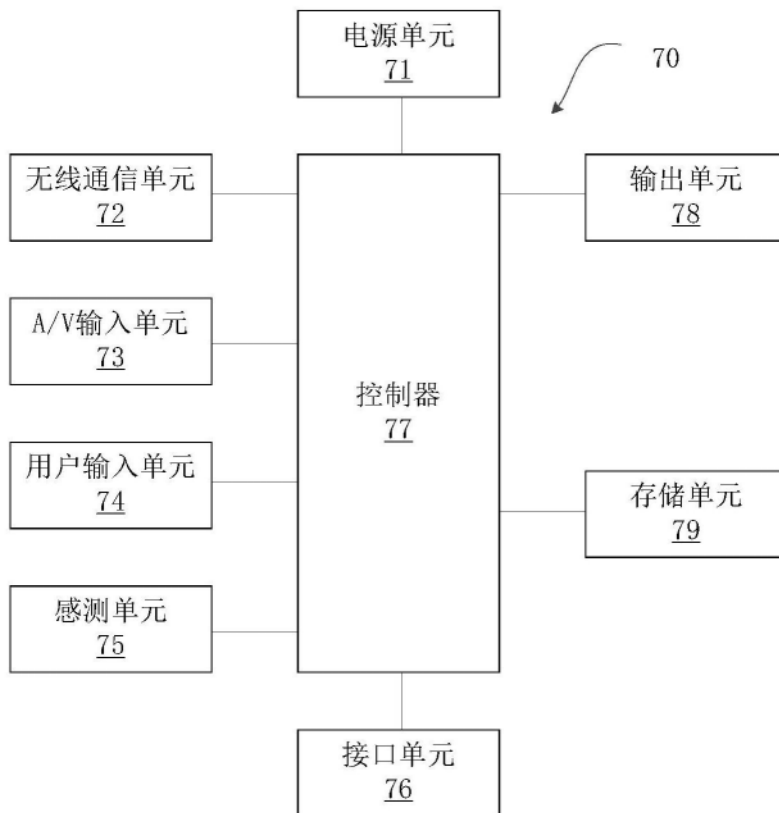


图7