



(12) 发明专利

(10) 授权公告号 CN 115097994 B

(45) 授权公告日 2023.05.09

(21) 申请号 202111676017.5

(22) 申请日 2021.12.31

(65) 同一申请的已公布的文献号
申请公布号 CN 115097994 A

(43) 申请公布日 2022.09.23

(66) 本国优先权数据
202111278454.1 2021.10.30 CN

(73) 专利权人 荣耀终端有限公司
地址 518040 广东省深圳市福田区香蜜湖
街道东海社区红荔西路8089号深业中
城6号楼A单元3401

(72) 发明人 庄秉彦

(74) 专利代理机构 北京同立钧成知识产权代理
有限公司 11205
专利代理师 余娜 黄健

(51) Int.Cl.

G06F 3/0484 (2022.01)

G06F 9/50 (2006.01)

(56) 对比文件

CN 109448089 A, 2019.03.08

CN 111124230 A, 2020.05.08

CN 112717370 A, 2021.04.30

审查员 王露露

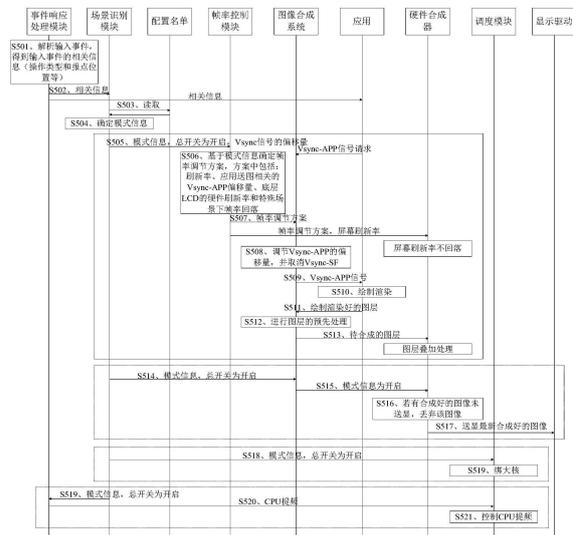
权利要求书2页 说明书19页 附图9页

(54) 发明名称

数据处理方法和相关装置

(57) 摘要

本申请实施例提供一种数据处理方法和相关装置,应用于终端技术领域。该方法应用于终端设备,终端设备包括事件响应处理模块、场景识别模块和硬件合成器,方法包括:事件响应处理模块获取到手写笔的手写输入事件时,确定手写输入事件对应的应用;事件响应处理模块向场景识别模块发送应用的信息;场景识别模块调用预先配置的配置名单;当应用在配置名单中对应为快速送显特性开启时,场景识别模块向硬件合成器发送第一信息;第一信息用于指示快速送显特性开启;硬件合成器基于第一信息,将缓存队列中的部分待送显图像帧丢弃。这样,通过丢弃部分待送显图像帧,缩短输入操作后的响应时延,进而缩短整个图像显示的耗时,提高输入操作的跟手性。



1. 一种数据处理方法,其特征在于,应用于终端设备,所述终端设备包括事件响应处理模块、场景识别模块和硬件合成器,所述方法包括:

所述事件响应处理模块获取到手写笔的手写输入事件时,确定所述手写输入事件对应的应用;

所述事件响应处理模块向所述场景识别模块发送所述应用的信息;

所述场景识别模块调用预先配置的配置名单;

当所述应用在所述配置名单中对应为快速送显特性开启时,所述场景识别模块向所述硬件合成器发送第一信息;所述第一信息用于指示所述快速送显特性开启;

所述硬件合成器基于所述第一信息,将缓存队列中的部分待送显图像帧丢弃;

所述方法还包括:

所述场景识别模块从所述配置名单中获取所述应用对应的第二信息,所述第二信息用于指示偏移量;所述第二信息为合成垂直同步Vsync-SF信号的周期起始时间与Vsync-APP信号的周期起始时间的差值;

所述场景识别模块向帧率控制模块发送所述第二信息;

所述帧率控制模块向图像合成系统发送所述第二信息;

所述图像合成系统基于所述第二信息调整应用垂直同步Vsync-APP信号的偏移量,使得图像的绘制渲染,以及图像的合成在一个所述Vsync-APP信号对应周期时长内完成;所述Vsync-APP信号用于触发所述应用绘制渲染图像。

2. 根据权利要求1所述的方法,其特征在于,所述将缓存队列中的部分待送显图像帧丢弃,包括:

当所述缓存队列包括第N帧和第N帧之前的图像帧时,将所述缓存队列中第N帧之前的图像帧丢弃;

当垂直同步Vsync信号到来时,若所述缓存队列接收到第N+1帧,则将丢弃所述第N帧。

3. 根据权利要求1或2所述的方法,其特征在于,所述应用的信息包括所述应用的包名信息。

4. 根据权利要求1所述的方法,其特征在于,所述方法还包括:

所述图像合成系统在接收到绘制渲染后的图像执行立即合成。

5. 根据权利要求4所述的方法,其特征在于,所述方法还包括:

当所述应用在所述配置名单中对应为帧率不回落特性开启时,所述场景识别模块向帧率控制模块发送第三信息,所述第三信息用于指示帧率不回落;

所述帧率控制模块基于所述第三信息调高屏幕刷新率,得到目标帧率;

所述帧率控制模块经图像合成系统将所述目标帧率发送至所述硬件合成器。

6. 根据权利要求5所述的方法,其特征在于,所述方法还包括:

所述场景识别模块向调度模块发送所述第一信息;

所述调度模块基于所述第一信息,将与显示相关的进程从第一核心调度到第二核心,所述第二核心的计算能力高于所述第一核心。

7. 根据权利要求6所述的方法,其特征在于,所述方法还包括:

所述场景识别模块向所述事件响应处理模块发送所述第一信息;

所述事件响应处理模块基于所述第一信息向调度模块发送用于指示提高处理器CPU运

行频率的消息；

所述调度模块基于所述用于指示提高处理器CPU运行频率的消息提高所述CPU运行频率。

8. 根据权利要求7所述的方法,其特征在于,所述配置名单包括下述一种或多种:快速送显特性的总开关、与所述快速送显特性的总开关对应的应用的包名、所述应用对应的偏移量,与帧率不回落特性对应的应用的包名。

9. 一种终端设备,其特征在于,所述终端设备包括处理器,所述处理器用于调用存储器中的计算机程序,以执行如权利要求1-8任一项所述的方法。

10. 一种计算机可读存储介质,其特征在于,所述计算机可读存储介质存储有计算机指令,当所述计算机指令在终端设备上运行时,使得所述终端设备执行如权利要求1-8任一项所述的方法。

11. 一种计算机程序产品,其特征在于,包括计算机程序,当所述计算机程序被运行时,使得终端设备执行如权利要求1-8任一项所述的方法。

12. 一种芯片,其特征在于,所述芯片包括处理器,所述处理器用于调用存储器中的计算机程序,以执行如权利要求1-8任一项所述的方法。

数据处理方法和相关装置

[0001] 本申请申请要求于2021年10月30日提交中国国家知识产权局、申请号为202111278454.1、申请名称为“数据处理方法和相关装置”的中国专利申请的优先权,其全部内容通过引用结合在本申请中。

技术领域

[0002] 本申请涉及终端技术领域,尤其涉及数据处理方法和相关装置。

背景技术

[0003] 目前,用户可以通过终端设备的显示屏查阅各类内容。终端设备的显示屏的界面显示通常需要经过绘制、渲染和合成等过程。示例性的,终端设备的界面绘制过程可以包括背景绘制、子视图的绘制、滚动条的绘制等过程。终端设备的界面合成过程可以包括顶点处理和像素处理等处理过程。

[0004] 但是,终端设备的显示屏在显示界面时可能会出现卡顿现象,用户体验差。

发明内容

[0005] 本申请实施例提供数据处理方法和相关装置,应用于终端设备。在送显过程中减少的送显任务堆积等待,使得应用绘制的图形能够及时送至屏幕显示给用户,减少卡顿感,减少送显卡顿的堆积,增强设备高负载下的送显稳定性;该方法用于减少终端设备显示的时延,减少卡顿现象,提升跟手性。

[0006] 第一方面,本申请实施例提出一种数据处理方法,应用于终端设备,终端设备包括事件响应处理模块、场景识别模块和硬件合成器,方法包括:事件响应处理模块获取到手写笔的手写输入事件时,确定手写输入事件对应的应用;事件响应处理模块向场景识别模块发送应用的信息;场景识别模块调用预先配置的配置名单;当应用在配置名单中对应为快速送显特性开启时,场景识别模块向硬件合成器发送第一信息;第一信息用于指示快速送显特性开启;硬件合成器基于第一信息,将缓存队列中的部分待送显图像帧丢弃。

[0007] 第一消息可以为模式信息中的一部分。图像帧可以称为图像或者帧。这样,通过丢弃部分待送显图像帧,缩短输入操作后的响应时延,进而缩短整个图像显示的耗时,提高输入操作的跟手性。

[0008] 可选的,将缓存队列中的部分待送显图像帧丢弃,包括:当缓存队列包括第N帧和第N帧之前的图像帧时,将缓存队列中第N帧之前的图像帧丢弃;当垂直同步Vsync信号到来时,若缓存队列接收到第N+1帧,则将丢弃第N帧。

[0009] 可选的,应用的信息包括应用的包名信息。

[0010] 可选的,方法还包括:场景识别模块从配置名单中获取应用对应的第二信息,第二信息用于指示偏移量;场景识别模块向帧率控制模块发送第二信息;帧率控制模块向图像合成系统发送第二信息;图像合成系统基于第二信息调整应用垂直同步Vsync-APP信号的偏移量,使得图像的绘制渲染,以及图像的合成在一个Vsync-APP信号对应周期时长内完

成;Vsync-APP信号用于触发应用绘制渲染图像。

[0011] 第二消息可以为模式信息中的一部分。这样,调整Vsync-APP信号的偏移量,使得图像可以在一个Vsync信号对应周期时长内完成绘制渲染,以及合成,缩短绘制渲染至合成的时间,进而缩短图像从绘制渲染到显示的时间,加速显示,提高跟手性。

[0012] 可选的,第二信息为合成垂直同步Vsync-SF信号的周期起始时间与Vsync-APP信号的周期起始时间的差值。

[0013] 这样,可以将Vsync-APP和Vsync-SF的差值调节在一个最优的状态,使得图形绘制流程各模块等待耗时少,使得图像在一个硬件Vsync信号内完成,提升快速送显的成功率。

[0014] 可选的,方法还包括:图像合成系统在接收到绘制渲染后的图像执行立即合成。

[0015] 这样,图像在绘制渲染后的立即合成,缩短绘制渲染至合成的时间,进而缩短图像从绘制渲染到显示的时间,加速显示,提高跟手性。

[0016] 可选的,方法还包括:当应用在配置名单中对应为帧率不回落特性开启时,场景识别模块向帧率控制模块发送第三信息,第三信息用于指示帧率不回落;帧率控制模块基于第三信息调高屏幕刷新率,得到目标帧率;帧率控制模块经图像合成系统将目标帧率发送至硬件合成器。

[0017] 屏幕刷新率也可以称为硬件刷新率。这样,提高屏幕刷新率,缩短Vsync周期时间,加快终端设备的绘制渲染、合成及送显节奏,进而缩短各个流程的等待时间,缩短输入操作后的响应时间,提高跟手性

[0018] 可选的,方法还包括:场景识别模块向调度模块发送第一信息;调度模块基于第一信息,将与显示相关的进程从第一核心调度到第二核心,第二核心的计算能力高于第一核心。

[0019] 这样,提高与显示相关的进程(例如,渲染进程、合成进程等)的运算能力,缩短各个进程的响应时间,缩短图像的绘制渲染、合成等流程的耗时,提高跟手性。

[0020] 可选的,方法还包括:场景识别模块向事件响应处理模块发送第一信息;事件响应处理模块基于第一信息向调度模块发送用于指示提高CPU运行频率的消息;调度模块基于用于指示提高CPU运行频率的消息提高CPU运行频率。

[0021] 这样,提高CPU的运行频率,提高运算能力,缩短终端设备的响应时间,缩短图像的绘制渲染、合成等流程的耗时,提高跟手性。

[0022] 可选的,配置名单包括下述一种或多种:快速送显特性的总开关、与快速送显特性的总开关对应的应用的包名、应用对应的偏移量,与帧率不回落特性对应的应用的包名。

[0023] 第二方面,本申请实施例提供了一种终端设备,终端设备也可以称为终端(terminal)、用户设备(user equipment,UE)、移动台(mobile station,MS)、移动终端(mobile terminal,MT)等。终端设备可以是手机(mobile phone)、智能电视、穿戴式设备、平板电脑(Pad)、带无线收发功能的电脑、虚拟现实(virtual reality,VR)终端设备、增强现实(augmented reality,AR)终端设备、工业控制(industrial control)中的无线终端、无人驾驶(self-driving)中的无线终端、远程手术(remote medical surgery)中的无线终端、智能电网(smart grid)中的无线终端、运输安全(transportation safety)中的无线终端、智慧城市(smart city)中的无线终端、智慧家庭(smart home)中的无线终端等等。

[0024] 该终端设备包括处理器,处理器用于调用存储器中的计算机程序以执行如第一方

面的方法。

[0025] 第三方面,本申请实施例提供了一种计算机可读存储介质,计算机可读存储介质存储有计算机指令,当计算机指令在终端设备上运行时,使得终端设备执行如第一方面的方法。

[0026] 第四方面,本申请实施例提供了一种计算机程序产品,当计算机程序被运行时,使得终端设备执行如第一方面的方法。

[0027] 第五方面,本申请实施例提供了一种芯片,芯片包括处理器,处理器用于调用存储器中的计算机程序,以执行如第一方面的方法。

[0028] 应当理解的是,本申请的第二方面至第五方面与本申请的第一方面的技术方案相对应,各方面及对应的可行实施方式所取得的有益效果相似,不再赘述。

附图说明

[0029] 图1为本申请实施例提供的终端设备硬件系统结构示意图;

[0030] 图2为本申请实施例提供的终端设备软件系统结构示意图;

[0031] 图3为Vsync信号触发三个显示流程的一种显示示意图;

[0032] 图4为本申请实施例提供的应用场景示意图;

[0033] 图5为本申请实施例提供的一种模块之间的交互示意图示意图;

[0034] 图6为本申请实施例提供的一种显示流程的示意图;

[0035] 图7为本申请实施例提供的一种显示流程的示意图;

[0036] 图8为本申请实施例提供的一种数据处理方法的流程示意图;

[0037] 图9为本申请实施例提供的一种数据处理方法的流程示意图;

[0038] 图10为本申请实施例提供的一种数据处理方法的流程示意图;

[0039] 图11为本申请实施例提供的一种数据处理装置的结构示意图;

[0040] 图12为本申请实施例提供的一种数据处理装置的硬件结构示意图。

具体实施方式

[0041] 为了便于清楚描述本申请实施例的技术方案,在本申请的实施例中,采用了“第一”、“第二”等字样对功能和作用基本相同的相同项或相似项进行区分。例如,第一芯片和第二芯片仅仅是为了区分不同的芯片,并不对其先后顺序进行限定。本领域技术人员可以理解“第一”、“第二”等字样并不对数量和执行次序进行限定,并且“第一”、“第二”等字样也并不限定一定不同。

[0042] 需要说明的是,本申请实施例中,“示例性的”或者“例如”等词用于表示作例子、例证或说明。本申请中被描述为“示例性的”或者“例如”的任何实施例或设计方案不应被解释为比其他实施例或设计方案更优选或更具优势。确切而言,使用“示例性的”或者“例如”等词旨在以具体方式呈现相关概念。

[0043] 本申请实施例中,“至少一个”是指一个或者多个,“多个”是指两个或两个以上。“和/或”,描述关联对象的关联关系,表示可以存在三种关系,例如,A和/或B,可以表示:单独存在A,同时存在A和B,单独存在B的情况,其中A,B可以是单数或者复数。字符“/”一般表示前后关联对象是一种“或”的关系。“以下至少一项(个)”或其类似表达,是指的这些项中

的任意组合,包括单项(个)或复数项(个)的任意组合。例如,a,b,或c中的至少一项(个),可以表示:a,b,c,a-b,a-c,b-c,或a-b-c,其中a,b,c可以是单个,也可以是多个。

[0044] 本申请实施例提供的数据处理方法,可以应用在具备显示功能的电子设备中。

[0045] 电子设备包括终端设备,终端设备也可以称为终端(terminal)、用户设备(user equipment,UE)、移动台(mobile station,MS)、移动终端(mobile terminal,MT)等。终端设备可以是手机(mobile phone)、智能电视、穿戴式设备、平板电脑(Pad)、带无线收发功能的电脑、虚拟现实(virtual reality,VR)终端设备、增强现实(augmented reality,AR)终端设备、工业控制(industrial control)中的无线终端、无人驾驶(self-driving)中的无线终端、远程手术(remote medical surgery)中的无线终端、智能电网(smart grid)中的无线终端、运输安全(transportation safety)中的无线终端、智慧城市(smart city)中的无线终端、智慧家庭(smart home)中的无线终端等等。本申请的实施例对终端设备所采用的具体技术和具体设备形态不做限定。

[0046] 为了更好地理解本申请实施例,下面对本申请实施例的终端设备的结构进行介绍:

[0047] 图1示出了终端设备100的结构示意图。终端设备可以包括:射频(radio frequency,RF)电路110、存储器120、输入单元130、显示单元140、传感器150、音频电路160、无线保真(wireless fidelity,WiFi)模块170、处理器180、电源190以及蓝牙模块1100等部件。本领域技术人员可以理解,图1中示出的终端设备结构并不构成对终端设备的限定,可以包括比图示更多或更少的部件,或者组合某些部件,或者不同的部件布置。

[0048] 下面结合图1对终端设备的各个构成部件进行具体的介绍:

[0049] RF电路110可用于收发信息或通话过程中,信号的接收和发送,特别地,将基站的下行信息接收后,给处理器180处理;另外,将设计上行的数据发送给基站。通常,RF电路包括但不限于天线、至少一个放大器、收发信机、耦合器、低噪声放大器(low noise amplifier,LNA)、双工器等。此外,RF电路110还可以通过无线通信与网络和其他设备通信。上述无线通信可以使用任一通信标准或协议,包括但不限于全球移动通讯系统(global system of mobile communication,GSM)、通用分组无线服务(general packet radio service,GPRS)、码分多址(code division multiple access,CDMA)、宽带码分多址(wideband code division multiple access,WCDMA)、长期演进(long term evolution,LTE)、电子邮件、以及短消息服务(short messaging service,SMS)等。

[0050] 存储器120可用于存储软件程序以及模块,处理器180通过运行存储在存储器120的软件程序以及模块,从而执行终端设备的各种功能应用以及数据处理。存储器120可主要包括存储程序区和存储数据区,其中,存储程序区可存储操作系统、至少一个功能所需的应用程序(比如声音播放功能、图像播放功能等)、引导装载程序(boot loader)等;存储数据区可存储根据终端设备的使用所创建的数据(比如音频数据、电话本等)等。此外,存储器120可以包括高速随机存取存储器,还可以包括非易失性存储器,例如至少一个磁盘存储器件、闪存器件、或其他易失性固态存储器件。可以理解的是,本申请实施例中,存储器120中存储有蓝牙设备回连的程序。

[0051] 输入单元130可用于接收输入的数字或字符信息,以及产生与终端设备的用户设置以及功能控制有关的键信号输入。具体地,输入单元130可包括触控面板131以及其他输

入设备132。触控面板131,也称为触摸屏,可收集用户在其上或附近的触摸操作(比如用户使用手指、触笔等任何适合的物体或附件在触控面板131上或在触控面板131附近的操作),并根据预先设定的程式驱动相应的连接装置。可选的,触控面板131可包括触摸检测装置和触摸控制器两个部分。其中,触摸检测装置检测用户的触摸方位,并检测触摸操作带来的信号,将信号传送给触摸控制器;触摸控制器从触摸检测装置上接收触摸信息,并将它转换成触点坐标,再送给处理器180,并能接收处理器180发来的命令并加以执行。此外,可以采用电阻式、电容式、红外线以及表面声波等多种类型实现触控面板131。除了触控面板131,输入单元130还可以包括其他输入设备132。具体地,其他输入设备132可以包括但不限于物理键盘、功能键(比如音量控制按键、开关按键等)、轨迹球、鼠标、操作杆等中的一种或多种。

[0052] 显示单元140可用于显示由用户输入的信息或提供给用户的信息以及终端设备的各种菜单。显示单元140可包括显示面板141,可选的,可以采用液晶显示器(liquid crystal display,LCD)、有机发光二极管(organic light-emitting diode,OLED)等形式来配置显示面板141。进一步的,触控面板131可覆盖显示面板141,当触控面板131检测到在其上或附近的触摸操作后,传送给处理器180以确定触摸事件的类型,随后处理器180根据触摸事件的类型在显示面板141上提供相应的视觉输出。虽然在图1中,触控面板131与显示面板141是作为两个独立的部件来实现终端设备的输入和输入功能,但是在某些实施例中,可以将触控面板131与显示面板141集成而实现终端设备的输入和输出功能。

[0053] 终端设备还可包括至少一种传感器150,比如光传感器、运动传感器以及其他传感器。具体地,光传感器可包括环境光传感器及接近传感器,其中,环境光传感器可根据环境光线的明暗来调节显示面板141的亮度,接近传感器可在终端设备移动到耳边时,关闭显示面板141或背光。作为运动传感器的一种,加速度计传感器可检测各个方向上(一般为三轴)加速度的大小,静止时可检测出重力的大小及方向,可用于识别终端设备姿态的应用(比如横竖屏切换、相关游戏、磁力计姿态校准)、振动识别相关功能(比如计步器、敲击)等;至于终端设备还可配置的陀螺仪、气压计、湿度计、温度计、红外线传感器等其他传感器,在此不再赘述。

[0054] 音频电路160、扬声器161,传声器162可提供用户与终端设备之间的音频接口。音频电路160可将接收到的音频数据转换后的电信号,传输到扬声器161,由扬声器161转换为声音信号输出;另一方面,传声器162将收集的声音信号转换为电信号,由音频电路160接收后转换为音频数据,再将音频数据输出处理器180处理后,经RF电路110以发送给比如另一终端设备,或者将音频数据输出至存储器120以便进一步处理。

[0055] WiFi属于短距离无线传输技术,终端设备通过WiFi模块170可以帮助用户收发电子邮件、浏览网页和访问流式媒体等,它为用户提供了无线的宽带互联网访问。虽然图1示出了WiFi模块170,但是可以理解的是,其并不属于终端设备的必须构成,完全可以根据需要在不改变发明的本质的范围内而省略。

[0056] 处理器180是终端设备的控制中心,利用各种接口和线路连接整个终端设备的各个部分,通过运行或执行存储在存储器120内的软件程序或模块,以及调用存储在存储器120内的数据,执行终端设备的各种功能和处理数据,从而对终端设备进行整体监控。可选的,处理器180可包括一个或多个处理单元;优选的,处理器180可集成应用处理器和调制解调处理器,其中,应用处理器主要处理操作系统、用户界面和应用程序等,调制解调处理器

主要处理无线通信。可以理解的是,上述调制解调处理器也可以不集成到处理器180中。可以理解的是,本申请实施例中,存储器120中存储有蓝牙设备回连的程序,而处理器180可以用于调用存储器120中存储的蓝牙设备回连的程序并执行,以实现本申请实施例的蓝牙设备回连的方法。

[0057] 终端设备还包括给各个部件供电的电源190(比如电池),优选的,电源可以通过电源管理系统与处理器180逻辑相连,从而通过电源管理系统实现管理充电、放电、以及功耗管理等功能。

[0058] 蓝牙技术属于短距离无线传输技术,终端设备通过蓝牙模块1100可以与其他具备蓝牙模块的终端设备建立蓝牙连接,从而基于蓝牙通信链路进行数据传输。蓝牙模块1100根据实际需要,可以为低功耗蓝牙(blueetooth low energy,BLE),或模块。可以理解的是,本申请实施例中终端设备为用户终端和业务机具的情形下,终端设备包括蓝牙模块。但是可以理解的是,蓝牙模块并不属于终端设备的必须构成,完全可以根据需要在不改变发明的本质的范围内而省略,比如服务器中可以不包括蓝牙模块。

[0059] 尽管未示出,终端设备还可以包括摄像头。可选地,摄像头在终端设备上的位置可以为前置的,也可以为后置的,还可以为内置的(在使用时可伸出机身),本申请实施例对此不作限定。

[0060] 可选地,终端设备可以包括单摄像头、双摄像头或三摄像头等,本申请实施例对此不作限定。摄像头包括但不限于广角摄像头、长焦摄像头或深度摄像头等。例如,终端设备可以包括三摄像头,其中,一个为主摄像头、一个为广角摄像头、一个为长焦摄像头。

[0061] 可选地,当终端设备包括多个摄像头时,这多个摄像头可以全部前置,或者全部后置,或者全部内置,或者至少部分前置,或者至少部分后置,或者至少部分内置等,本申请实施例对此不作限定。

[0062] 终端设备100的软件系统可以采用分层架构,事件驱动架构,微核架构,微服务架构,或云架构,等。本申请实施例以分层架构的Android系统为例,示例性说明终端设备100的软件结构。

[0063] 图2是本申请实施例的终端设备的软件结构框图。分层架构将软件分成若干个层,每一层都有清晰的角色和分工。层与层之间通过软件接口通信。在一些实施例中,将Android系统分为五层,从上至下分别为应用层,应用程序框架层,安卓运行时(Android runtime)和系统库,硬件抽象层以及内核层。

[0064] 应用层可以包括一系列应用程序包。如图2所示,应用程序包可以包括电话、邮箱、日历、相机等应用程序。

[0065] 应用程序框架层为应用层的应用程序提供应用编程接口(application programming interface,API)和编程框架。应用程序框架层包括一些预先定义的函数。如图2所示,应用程序框架层可以包括窗口管理器、帧率控制模块、图像合成系统(surface flinger)、视图系统、场景识别模块、事件响应处理模块、配置名单、和安全模块等。

[0066] 窗口管理器用于管理窗口程序。窗口管理器可以获取显示屏大小,判断是否有状态栏,锁定屏幕,截取屏幕等。

[0067] 帧率控制模块用于进行帧率调节,进而控制图像的显示过程中各个流程(绘制、渲染、合成、送显)的节奏。本申请实施例中,帧率控制模块可以基于场景识别模块传输的模式

信息,调节显示屏的硬件刷新率、图像的显示过程中各个流程中相关的垂直同步(vetical synchronization,Vsync)信号的偏移量(offset)和确定特殊场景下帧率回落的特性。帧率控制模块也可以称为帧率控制模块。

[0068] 图像合成系统用于控制图像合成,即将应用绘制渲染好的图像进行整合叠加预处理。图像合成系统还可产生Vsync-APP信号以控制应用绘制渲染节奏(Vsync),以及产生Vsync-SF控制自身的合成节奏。图像合成系统中产生的各个Vsync信号的偏移量和时间间隔受到帧率控制模块的控制。

[0069] 视图系统包括可视控件,例如显示文字的控件,显示图片的控件等。视图系统可用于构建应用程序。显示界面可以由一个或多个视图组成的。例如,包括短信通知图标的显示界面,可以包括显示文字的视图以及显示图片的视图。

[0070] 场景识别模块用于基于用户操作信息和配置信息识别确认特殊场景的模式信息,并将模式信息传输至多个相关模块。相关模块可以为帧率控制模块、图像合成系统和事件响应处理模块等。

[0071] 事件响应处理模块用于处理用户和设备的交互的事件信息。事件响应处理模块可以在接收到场景识别模块传递的信息时,通过调度模块对CPU进行提频处理以提升用户体验。

[0072] 配置名单用于存储应用相关的配置信息,以为场景识别模块提供数据支撑。配置名单是xml文件,需要提前配置,且具有可扩展性和可更改性。配置名单中包括应用对应的包名信息以及用于指示快速送显的特性总开关。

[0073] 安全模块需要进行规则添加适配,实现安全适配。安全模块可以使得硬件合成器在读取属性值时正常的读(read)和写(write)权限不会被拦截。

[0074] Android runtime包括核心库和虚拟机。Android runtime负责安卓系统的调度和管理。

[0075] 核心库包含两部分:一部分是java语言需要调用的功能函数,另一部分是安卓的核心库。

[0076] 应用层和应用程序框架层运行在虚拟机中。虚拟机将应用层和应用程序框架层的java文件执行为二进制文件。虚拟机用于执行对象生命周期的管理,堆栈管理,线程管理,安全和异常的管理,以及垃圾回收等功能。

[0077] 系统库可以包括多个功能模块。例如:图像渲染库、图像合成库、函数库、媒体库和输入处理库等。

[0078] 图像渲染库用于二维或三维图像的渲染。图像合成库用于二维或三维图像的合成。

[0079] 可能的实现方式中,应用通过图像渲染库对图像进行绘制渲染,然后应用将绘制渲染后的图像发送至图像合成系统的缓存队列中。每当Vsync信号到来时,图像合成系统(例如,surface flinger)从缓存队列中按顺序获取待合成的一帧图像,然后通过图像合成库进行图像合成。

[0080] 函数库提供C语言中所使用的宏、类型定义、字符串操作函数、数学计算函数以及输入输出函数等。

[0081] 媒体库支持多种常用的音频,视频格式回放和录制,以及静态图像文件等。媒体库

可以支持多种音视频编码格式,例如:MPEG4、H.264、MP3、AAC、AMR、JPG和PNG等。

[0082] 输入处理库用于处理输入设备的库,可以实现鼠标、键盘和触摸输入处理等。

[0083] 硬件抽象层,可以包含多个库模块,库模块如可以为硬件合成器(hwcomposer, HWC)、调度模块等。Android系统可以为设备硬件加载相应的库模块,进而实现应用程序框架层访问设备硬件的目的。设备硬件可以包括如电子设备中的LCD显示屏、摄像头等。

[0084] 硬件合成器用于将图像合成系统合成好的图像控制硬件(显示屏)的刷新率,并且能够将图层叠加预处理模块预处理好的所有图层进行整合叠加所需要的信息。

[0085] 本申请实施例中,硬件合成器可以接收并存储场景识别模块下发的模式信息,以及基于该模式信息进行将图像合成系统合成好的图像通过显示驱动送至显示屏(屏幕)显示。

[0086] 可能的实现方式中,通过全局属性值“vendor.oneframe.hwc.no_wait_pf”存储模式信息。在将合成的图像送显给显示屏时,确认当前的模式。示例性的,在特殊场景下,丢弃待送显队列中的缓存数据(缓存的图像),并送显最新的一帧图像缓存(buffer)数据。

[0087] 调度模块用于负责系统各进程处理器CPU和图像处理器GPU资源的调度。本申请实施例中,调度模块可以基于模式信息进行相应的模块(例如,图像合成系统、硬件合成器和触控面板,等)提频和绑大核操作。

[0088] 内核层是硬件和软件之间的层。内核层至少包含触控(touch panel, TP)驱动、显示驱动、蓝牙驱动、WIFI驱动、键盘驱动、共用存储器驱动和相机驱动等。

[0089] 显示驱动可以基于硬件Vsync信号(Vsync-HW),通过前台缓存(FrontBuffer)和后台缓存(BackBuffer)的切换将图像显示到显示屏上,以供用户观看。

[0090] Android系统可能还包括硬件层。硬件层包括多种硬件。硬件可以是触控面板和显示屏等。

[0091] 下面结合应用程序示例性说明终端设备100软件以及硬件的工作流程。

[0092] 当触控面板中传感器接收到手写笔的触控操作时,内核层将触控操作加工成原始输入事件(包括触摸坐标,触摸力度,触摸操作的时间戳,触控主体的类型等信息)。原始输入事件被存储在内核层。内核层通过输入处理库将原始输入事件上报至应用程序框架层的事件响应处理模块。应用程序框架层的事件响应处理模块解析该原始输入事件的信息(包括:操作类型和报点位置等)和根据当前焦点确定焦点应用,并将解析后的信息发送至焦点应用和场景识别模块。焦点可以是触摸操作中触碰点或者鼠标点击操作中点击位置。焦点应用为终端设备前台运行的应用或者触摸操作中触碰位置对应的应用。焦点应用根据解析后的原始输入事件的信息(例如,报点位置)确定该原始输入事件所对应的控件。

[0093] 场景识别模块根据解析后的原始输入事件的信息(例如,操作类型)通过帧率控制模块控制图像显示的相关流程的节奏。

[0094] 以该触摸操作是触摸滑动操作,该触摸滑动操作所对应的控件为微信应用的列表控件为例,微信应用通过应用程序框架层的视图系统,调用系统库中图像渲染库对图像进行绘制渲染。微信应用将绘制渲染后的图像发送至图像合成系统(surface flinger)的缓存队列中。通过系统库中图像合成库将图像合成系统中绘制渲染后的图层进行预先处理,并发送至硬件抽象层的硬件合成器(hwc)。硬件合成器将预先处理好的图层(待合成的图层)进行叠加合成为图像,并将图像通过内核层的显示驱动,使得屏幕(显示屏)显示微信应

用的相应界面。

[0095] 为了便于理解,示例的给出部分与本申请实施例相关概念的说明以供参考。

[0096] 1、帧:是指界面显示中最小单位的单幅画面。一帧可以理解为一副静止的画面,快速连续地显示多个相连的帧可以形成物体运动的假象。帧率是指在1秒钟时间里刷新图片的帧数,也可以理解为终端设备中图形处理器每秒钟刷新画面的次数。高的帧率可以得到更流畅和更逼真的动画。每秒钟帧数越多,所显示的动作就会越流畅。

[0097] 需要说明的是,界面显示帧前通常需要经过绘制、渲染、合成等过程。

[0098] 2、帧绘制:是指显示界面的图片绘制。显示界面可以由一个或多个视图组成,各个视图可以由视图系统的可视控件绘制,各个视图由子视图组成,一个子视图对应视图中的一个部件,例如,其中的一个子视图对应图片视图中的一个符号。

[0099] 3、帧渲染:是将绘制后的视图进行着色操作或增加3D效果等。例如:3D效果可以是灯光效果、阴影效果和纹理效果等。

[0100] 4、帧合成:是将多个上述一个或多个渲染后的视图合成为显示界面的过程。

[0101] 下面对终端设备的界面的显示过程进行说明。

[0102] 需要说明的是,为了提高显示的流畅性,减少出现显示卡顿等现象,终端设备一般基于Vsync信号进行显示,以对图像的绘制、渲染、合成和屏幕刷新显示等流程进行同步。

[0103] 可以理解的是,Vsync信号为周期性信号,Vsync信号周期可以根据屏幕刷新率进行设置,例如,屏幕刷新率为60Hz时,Vsync信号周期可以为16.6ms,即终端设备每间隔16.6ms生成一个控制信号使Vsync信号周期触发。

[0104] 需要说明的是,Vsync信号可以分为软件Vsync信号和硬件Vsync信号。软件Vsync信号包括Vsync-APP和Vsync-SF。Vsync-APP用于触发绘制渲染流程。Vsync-SF用于触发合成流程。硬件Vsync信号(Vsync-HW)用于触发屏幕显示刷新流程。

[0105] 通常情况下,软件Vsync信号和硬件Vsync信号保持周期同步。以60Hz和120Hz变化为例,若Vsync-HW从60Hz切换到120Hz,Vsync-APP、Vsync-SF同步变化,从60Hz切换到120Hz。

[0106] 示例性的,图3为Vsync信号触发三个显示流程的一种显示示意图。如图3所示,若用户通过应用向图像合成系统(surface flinger)发送的Vsync信号请求,以请求显示第4帧图像。surface flinger在第N个Vsync周期的起始时刻之后,在第N+1个Vsync周期的起始时刻或之前的任一时刻接收到该Vsync信号请求(也可以称为视图更新请求),即surface flinger在图3中的时间段A(不包含端点)中的任一时刻或时刻a接收到该Vsync信号请求。则,在第N+1个Vsync周期的起始时刻,应用启动应用绘制渲染流程,开始对第4帧图像进行应用绘制渲染。同时,surface flinger启动图像合成流程,开始对第3帧图像进行图像合成,且硬件启动硬件送显流程,开始对第2帧图像进行硬件送显。硬件送显流程完成后,显示屏对第2帧图像进行显示。

[0107] 同理,若surface flinger在第N+1个Vsync周期的起始时刻之后,在第N+2个Vsync周期的起始时刻或之前的任一时刻,即图3中的时间段B(不包含端点)中的任一时刻或时刻b,接收到第5帧图像的Vsync信号请求(也可以称为视图更新请求)。则在第N+2个Vsync周期的起始时刻,应用启动应用绘制渲染流程,开始对第5帧图像进行应用绘制渲染,同时,surface flinger启动图像合成流程,开始对第4帧图像进行图像合成,且硬件启动硬件送

显流程,开始对第3帧图像进行硬件送显。硬件送显流程完成后,显示屏对第3帧图像进行显示。以此类推。

[0108] 从图3可以看出,显示系统基于Vsync信号进行显示能够实现显示流水线中各个流程延迟时间的一致,可以减少应用绘制渲染流程和surface flinger合成流程中的错误,减少显示卡顿等现象。然而,由图3可见,对于每一帧图像,从开始应用绘制渲染到硬件送显至少需要经过2个Vsync周期。例如,第4帧图像在第N+1个Vsync周期开始应用绘制渲染,至少需要在第N+3个Vsync周期才能进行显示,图形显示效率有待提升。

[0109] 进一步的,结合视图更新请求时间和硬件送显时长来分析:

[0110] 硬件送显时长较短,一般为2ms左右。若surface flinger接收到第4帧图像的视图更新请求的时刻刚好为第N+1个Vsync周期起始时刻,即图3中的a时刻。那么,从surface flinger接收到视图更新请求到硬件送显的时间,与从开始应用绘制渲染到硬件送显的时间相同,需要2个Vsync周期,如图3中的时间段C。进一步的,从surface flinger接收到视图更新请求到显示屏中显示第4帧图像需要2个周期+2ms,如图3中的时间段D。因此,可以看出,显示系统对于第4帧图形的显示,总的延迟时间(即从接收到视图更新请求到显示屏中显示该帧图像的时间)大于2个Vsync周期。

[0111] 若surface flinger接收到第4帧图像的视图更新请求的时刻在第N+1个Vsync周期起始时刻之前,且在N个Vsync周期起始时刻之的任一时刻,即图3中的时间段A(不包含端点)中的任一时刻,则第4帧图像的应用绘制渲染流程需要等到第N+1个Vsync周期的起始时刻开始执行。以第4帧图像的视图更新请求时刻为图3中的c时刻为例,从surface flinger接收到视图更新请求到硬件送显的时间,比从开始应用绘制渲染到硬件送显的时间更长,需要2个Vsync周期+第N+1个Vsync周期的起始时刻a与视图更新请求时刻c之差,如图3中的时间段E。进一步的,从surface flinger接收到视图更新请求到显示屏中显示第4帧图像需要2个Vsync周期+2ms+第N+1个Vsync周期的起始时刻a与视图更新请求时刻c之差,如图3中的时间段F。因此,可以看出,显示系统对于第4帧图像的显示延迟大于2个Vsync周期。可以理解的是,图3中Vsync信号同时触发显示流水线中的三个流程,可以理解为Vsync-APP信号、Vsync-SF信号和Vsync-HW信号一致。

[0112] 显示系统中显示流水线中的三个流程可以同时被Vsync信号触发,每一帧图像从surface flinger接收到Vsync信号请求到硬件送显至少需要2个Vsync周期,且每一帧图像的显示,总的延迟时间大于2个Vsync周期,图形显示效率较差,触控响应较慢,导致手写笔场景中手写笔的跟手性差。

[0113] 有鉴于此,本申请实施例提供一种数据处理方法,通过读取配置名单,确认是否为特殊场景,当处于特殊场景时(手写笔场景)时,丢弃待送显的帧,以缩短图像送显的时间。进而缩短整个图像显示的耗时,提高输入操作的跟手性。此外,还可以在处于特殊场景时(手写笔场景)时,调节Vsync信号的偏移量,将应用的绘制渲染时间提前,缩短输入操作的响应至应用绘制渲染时间,

[0114] 本申请实施例提供的方法可以应用于包括触控面板的终端设备中。示例性的,图4为本申请实施例提供的一种应用场景示意图。

[0115] 如图4所示,该场景中包括触控笔(stylus) 101、电子设备201和无线键盘301。触控笔101可以包括但不限于为:电感笔和电容笔。本申请实施例对此不作限定。电子设备201可

以为具有触控屏202的终端设备。本申请实施例对于电子设备200的种类不做限定。

[0116] 以电子设备201为平板电脑(tablet)为例进行说明。触控笔101和无线键盘301可以向电子设备201提供输入,电子设备201基于触控笔101或无线键盘301的输入,执行响应于该输入的操作。

[0117] 可以理解的是,电子设备201的触控屏202可以基于触控笔101的触控操作进行相应的界面变化。可以理解的是,还可以应用于终端设备的触控输入操作,例如,通过手指触控输入等场景。此处不做限定。

[0118] 下面结合图5对本申请实施例提供的方法中涉及的模块之间的交互进行说明。涉及的模块包括:事件响应处理模块、场景识别模块、配置名单、帧率控制模块、图像合成系统、应用、硬件合成器、调度模块和显示驱动。

[0119] S501、当终端设备接收到手写笔(触控笔)的输入事件时,事件响应处理模块解析输入事件,得到输入事件的相关信息。可能实现的方式中,相关信息包括:触控主体的类型、操作类型和报点位置等。

[0120] S502、事件响应处理模块将解析后的相关信息发送至场景识别模块。可能的实现中,事件响应处理模块还将解析后的相关信息发送至应用。

[0121] S503、场景识别模块在接收到相关信息后,调用读取配置名单中相应的配置文件。

[0122] 可能的实现方式中,场景识别模块基于前台运行的应用的包名从配置名单中获取相应的配置文件。

[0123] 可以理解的是,解析后的相关信息中可以包括前台运行的应用的包名;场景识别模块还可以从窗口管理器(ams)中获取前台运行的应用的应用包名。或者当应用启动时,场景识别模块可以获取得到前台运行的应用的包名。

[0124] S504、场景识别模块确定模式信息。

[0125] 本申请实施例中,场景识别模块基于配置文件中存储的数据,可以确定模式信息。可能实现的方式中,配置文件中包括:特性的总开关,以及与特性相对应的配置参数。

[0126] 示例性的,特性可以为快速送显,帧率不回落等。示例性的,当前台运行的应用对应的快速送显特性的总开关为开启时,终端设备根据相应的配置参数(例如,Vsync信号的偏移量)控制显示,以加快显示节奏。当前台运行的应用对应的快速送显特性的总开关为关闭时,不进行快速送显。

[0127] 本申请实施例中,场景识别模块确定的模式信息包括:快速送显特性的总开关开启,以及Vsync信号的偏移量。快速送显特性的总开关用于控制是否进行快速送显。

[0128] 当模式信息中快速送显特性的总开关开启时,场景识别模块将模式信息发送至多个模块。

[0129] 可能的实现方式一中,场景识别模块将模式信息发送至帧率控制模块。

[0130] S505、场景识别模块将模式信息发送至帧率控制模块。

[0131] 示例性的,模式信息包括Vsync信号的偏移量。

[0132] S506、帧率控制模块基于模式信息确定帧率调节方案。

[0133] 帧率调节方案包括但不限于:应用帧率、相关的Vsync信号的偏移量、底层LCD的硬件刷新率和特殊场景下帧率不回落等。

[0134] 可能的实现方式中,模式信息中Vsync信号的偏移量用于指示Vsync-APP信号的偏

移量;帧率调节方案包括:Vsync-APP信号的偏移量。

[0135] 可能的实现方式中,模式信息中Vsync信号的偏移量用于指示Vsync-APP信号的偏移量与Vsync-SF信号的偏移量之间的差值。帧率调节方案中,Vsync-APP信号的偏移量为Vsync-SF信号的偏移量减去该差值。这样,终端设备可以在一个Vsync-APP周期内完成绘制渲染流程和合成流程,进而缩短输入操作后的响应时间。

[0136] S507、帧率控制模块将帧率调节方案发送至图像合成系统。

[0137] 可能的实现方式中,帧率调节方案还包括硬件刷新率(屏幕刷新率)时,帧率控制模块可以将硬件刷新率发送至硬件合成器,以调节硬件刷新率。

[0138] 这样,可以通过调高硬件刷新率,缩短Vsync周期时间,加快终端设备的绘制渲染、合成及送显节奏,进而缩短各个流程的等待时间,缩短输入操作后的响应时间,提高跟手性。

[0139] S508、图像合成系统根据帧率调节方案调节Vsync-APP信号的偏移量,并取消Vsync-sf信号。

[0140] 可以理解的是,后续图像合成系统在接收到绘制渲染后的图像执行立即合成,因此,可以取消Vsync-sf信号。

[0141] 可以理解的是,图像合成系统调节Vsync信号的偏移量,以提前Vsync信号的生成时间,进而提前触发显示流程,缩短显示流程的等待时间。

[0142] 可能的实现方式中,图像合成系统调节Vsync-APP信号的偏移量,以提前Vsync-APP信号的生成时间,并取消Vsync-SF信号。这样,输入操作后应用绘制渲染图像的时间可以提前,进而缩短输入响应至应用绘制渲染时间,进而缩短整个图像显示的耗时,提高输入操作的跟手性。图像合成系统在取消Vsync-sf信号后,合成应用绘制渲染好的图像,并将合成后的图像发送至硬件合成器进行处理并送显。这样,图像合成系统可以立即合成绘制渲染好的图像以进行显示,缩短整个图像显示的耗时,提高输入操作的跟手性。

[0143] 示例性的,如图6所示,为了减少帧图像显示的延迟,通过图像合成系统调节Vsync-APP信号的偏移量实现偏移设置。图6中,Vsync-APP信号和Vsync-HW信号分别触发应用绘制渲染流程和硬件送显流程。图像合成系统合成流程在接收到绘制渲染好的图像后触发。

[0144] 如图6所示,若在c点时,终端设备接收到手写笔(触控笔)的输入事件时,终端设备的应用向图像合成系统发送的Vsync信号请求,以请求显示第4帧图像。同时,由于该输入事件对应的场景为快速送显,帧率控制模块调节Vsync信号的偏移量,以提前Vsync信号的生成时间。

[0145] 相比较于图3所示的显示流程,帧率控制模块通过图像合成系统调节Vsync-APP信号的偏移量,Vsync-APP信号的生成时间提前,缩短输入事件至绘制渲染图像的时间。这样,缩短输入操作后绘制渲染的第一帧(帧4)的显示时间,提高输入操作的跟手性。此外,surface flinger合成流程不受Vsync-SF信号的控制,在接收到绘制渲染好的图像后触发,进而可以缩短绘制渲染至合成的时间,进而缩短整个图像显示的耗时,提高输入操作的跟手性。

[0146] 图6中,从surface flinger接收到视图更新请求到显示屏中显示第4帧图像需要1个Vsync周期+Vsync-APP信号的偏移量+2ms+第N+1个Vsync周期的起始时刻a与视图更新请

求时刻c之差,如图6中的时间段F。因此,可以看出,显示系统对于第4帧图像的显示延迟小于2个Vsync周期。另一种可能的实现方式中,图像合成系统调节Vsync-APP信号的偏移量使得Vsync-APP信号和Vsync-SF信号之间满足一定的差值,进而图像的绘制渲染和合成可以在一个Vsync周期内完成,缩短输入操作响应至图像送显的时间。

[0147] 示例性的,如图7所示,为了减少帧图像显示的延迟,通过图像合成系统调节Vsync-APP信号的偏移量实现偏移设置。图7中,Vsync-APP信号和Vsync-SF信号和Vsync-HW信号分别触发应用绘制渲染流程、图像合成系统合成流程和硬件送显流程。

[0148] 如图7所示,若在c点时,终端设备接收到手写笔(触控笔)的输入事件时,终端设备的应用向图像合成系统发送的Vsync信号请求,以请求显示第4帧图像。同时,由于该输入事件对应的场景为快速送显,帧率控制模块根据Vsync-APP信号的偏移量与Vsync-SF信号的偏移量的差值通过图像合成系统调节Vsync-APP信号的偏移量,使得图像的绘制渲染和合成可以在一个Vsync-APP周期内完成。

[0149] 相比较于图3所示的显示流程,Vsync-APP信号的生成时间提前,缩短输入事件至绘制渲染图像的时间。Vsync-APP信号与Vsync-SF信号满足一定的差值,使得图像从绘制渲染结束至合成流程中的等待时间缩短。示例性的,在第N+1个Vsync-APP周期中,帧4完成绘制渲染,以及合成流程。这样,图像的绘制渲染和合成可以在一个Vsync周期内完成,进而提高跟手性。

[0150] 图7中,从surface flinger接收到视图更新请求到显示屏中显示第4帧图像需要1个Vsync周期+Vsync-APP信号的偏移量+2ms+第N+1个Vsync周期的起始时刻a与视图更新请求时刻c之差,如图7中的时间段F。因此,可以看出,显示系统对于第4帧图像的显示延迟小于2个Vsync周期。

[0151] S509、图像合成系统在接收到应用发送的Vsync-APP请求后,向应用发送Vsync-APP信号以控制应用的绘制渲染。

[0152] S510、应用在接收到Vsync-APP后,开始绘制渲染图层。

[0153] S511、应用将绘制渲染好的图层发送至图像合成系统。

[0154] S512、图像合成系统在接收到绘制渲染好的图层后,进行图层的预先处理,得到待合成图层。

[0155] S513、图像合成系统将待合成图层发送至硬件合成器以进行图层叠加处理。硬件合成器将图层叠加处理后的图像送至显示驱动以进行显示。

[0156] 可能的实现方式二中,场景识别模块将模式信息发送至图像合成系统。

[0157] S514、场景识别模块将模式信息发送至图像合成系统。

[0158] 模式信息中用于指示快速送显的总开关为开启状态。

[0159] S515、图像合成系统将模式信息发送至硬件合成器,以丢弃全部或部分堆积的待送显的图像缓存。

[0160] S516、硬件合成器在接收到模式信息后,若硬件合成器中存在合成好的图像未送显时,丢弃全部或部分堆积的待送显的图像。

[0161] S517、硬件合成器在接收到模式信息后,向显示驱动送显最新的合成后的图像。这样,丢弃堆积的缓存,缩短输入操作后的图像显示的耗时,提高输入操作的跟手性。

[0162] 示例性的,当硬件合成器中存在2帧图像(帧3和帧4)未送显时,丢弃帧3。当Vsync-

HW信号到来,且图像合成系统没有发送帧5至硬件合成器时,硬件合成器向显示驱动送显帧4。当Vsync-HW信号到来,且图像合成系统发送帧5至硬件合成器时,硬件合成器向显示驱动送显帧5。这样,可以及时送显新合成的图像,缩短输入操作至显示的时间,提高跟手性。

[0163] 可能的实现方式三中,场景识别模块将模式信息发送至调度模块。

[0164] S518、场景识别模块将模式信息发送至调度模块。

[0165] S519、调度模块控制图像显示相关的模块绑大核。

[0166] 相关模块为显示流程中涉及的模块。相关模块包括但不限于:应用、图像合成系统、硬件合成器。可以理解的是,终端设备中有多个核心。小核心的计算能力低,大核心的计算能力高。调度模块控制图像显示相关的模块运行在大核心上。示例性的,终端设备中有7个核心。0-3与小核心相对应,计算能力低;4-7与大核心相对应,计算能力高。调度模块控制图像显示相关的模块运行4-7任一个核心上。

[0167] 这样,可以提升相关模块的运行速率,进而缩短相应流程的耗时,缩短整个图像显示的耗时,提高输入操作的跟手性。

[0168] 可能的实现方式四中,场景识别模块将模式信息发送至事件响应处理模块。

[0169] S520、场景识别模块将模式信息发送至事件响应处理模块。

[0170] S521、事件响应处理模块在接收到模式信息后,通过调度模块控制CPU提频。

[0171] 这样,CPU提频后可以提升相关模块的运行速率,进而缩短相应流程的耗时,缩短整个图像显示的耗时,提高输入操作的跟手性。

[0172] 可以理解的是,上述多种方式终端设备可以执行一种或多种,本申请实施例对于上述多种方式执行的先后顺序不做限定。

[0173] 下面结合图8-图10对本申请实施例的方法中相应模块具体的执行流程进行说明。

[0174] 图8为本申请实施例中数据处理方法的流程示意图。如图8所示,在接收到输入操作后,

[0175] S601、场景识别模块获取配置文件。可能的实现方式中,场景识别模块基于前台应用的名称从配置名单中获取相应的配置文件。

[0176] 配置文件可以包括特性名称、应用的包名(名称)、偏移量和特性总开关的配置信息;总开关的配置信息可以用于指示硬件合成器丢弃缓存。

[0177] 场景识别模块基于前台应用的名称确认快速送显特性的总开关的配置信息,当快速送显特性的总开关的配置信息为开启时,获取Vsync信号的偏移量。当快速送显特性的总开关的配置信息为关闭时,按默认方式处理。

[0178] 示例性的,配置文件的内容可以参照表1。

[0179] 表1配置文件内容

	Key(Param name)	属性名称 (特性名称)	Value	值
[0180]	itouch_one_frame_app_vsync_offset_config	快速送显的特性		
	app_vsync_offset_hwc_switch	总开关	1:打开	开
	com.hihonor.notepad	包名 (记事本)	"3500000" (3.5ms)	Offset偏移量
其他应用名称
[0181]	app_vsync_offset_hwc_switch	总开关	0:关闭	关
	"com.hihonor.notepad	(日历)		

	itouch_one_frame_list_high_freq	帧率不回落的特性		
	com.hihonor.notepad	包名 (记事本)	1	

[0182] 配置文件中配置有多个特性。xml解析的时候可以通过配置文件的特性名称进行特性的区分标识。

[0183] 具体的,场景识别模块逐行解析配置文件,基于特性名称查询相应的配置,确认总开关的配置。

[0184] 示例性的,表1中配置文件的特性包括:快速送显的特性和帧率不回落的特性。快速送显的特性名称可以为itouch_one_frame_app_vsync_offset_config。帧率不回落的特性名称可以为"itouch_one_frame_list_high_freq"。

[0185] xml解析的时候可以通过配置文件的特性名称进行特性的区分标识;其中,"app_vsync_offset_hwc_switch"为该特性的总开关,value为0时关闭,为1时打开,可配置的特性总开关能够方便后期调试与问题定位;

[0186] 特性名称下有多个应用的的包名信息。"com.hihonor.notepad"用于指示记事本的应用包名配置,其包名对应的value值"3500000"为Vsync-APP的偏移量和/或Vsync-SF的偏移量。可以理解的是,未在该特性名称后的包名默认不开启该特性。

[0187] 可能的实现方式中,包名对应的value值表示Vsync-APP与Vsync-SF之间的差值。这样,图像合成系统可以调节应用侧的绘制节奏(Vsync-APP),将Vsync-APP和Vsync-SF的差值调节在较优的状态,缩短图像绘制渲染至合成的等待时间,使得终端设备可以在一个Vsync周期内完成绘制渲染和合成。

[0188] 示例性的,表1中,"itouch_one_frame_list_high_freq"为帧率不回落方案的特性名称,xml解析的时候可以通过该名称进行区分标识;若没有该配置项,则默认关闭该特性;"com.hihonor.notepad"配置为应用包名,未在名单内的包名默认不开启该特性,其包名对应的value值为"1"时,默认该应用开启该特性,为"0"时关闭该特性。

[0189] S602、场景识别模块基于配置文件,确认场景的模式信息。

[0190] 模式信息用于指示快速送显的特性总开关是否开启,以及快速送显特性的总开关开启时,相应的Vsync信号的偏移量。Vsync信号的偏移量可以为Vsync-APP的偏移量。

[0191] 示例性的,若前台应用为记事本,且记事本对应的配置文件中快速送显特性总开关为开启和Vsync-APP的偏移量。当前台运行的应用为记事本时,场景识别模块基于记事本对应的配置文件,确认场景的模式信息为快速送显特性总开关为开启以及Vsync信号的偏移量。

[0192] 示例性的,若前台应用为日历,且日历对应的配置文件中的数据为总开关为关闭。当前台运行的应用为记事本时,场景识别模块基于记事本对应的配置文件,确认场景的模式信息为总开关为关闭。

[0193] 当应用对应的快速送显特性的总开关开启时,执行S603-S605。

[0194] S603、当应用对应的快速送显特性的总开关开启时,场景识别模块将Vsync信号的偏移量下发至帧率控制模块。

[0195] S604、帧率控制模块下发帧率控制策略。

[0196] 帧率控制策略也可以称为帧率调节方案。帧率控制策略包括但不限于:应用帧率、相关的Vsync信号(例如,Vsync-APP)的偏移量、底层LCD的硬件刷新率和特殊场景下帧率不回落等。可能的实现方式中,帧率控制策略中包括Vsync-APP的偏移量。帧率控制模块在接收到Vsync-APP的偏移量后,将Vsync-APP的偏移量下发至图像合成系统。

[0197] S605、图像合成系统基于帧率控制策略修改Vsync-APP的偏移量。

[0198] 图像合成系统基于Vsync-APP的偏移量,生成Vsync-APP信号,使得应用提前绘制渲染图像。这样,可以将应用绘制渲染的时间提前,以缩短整个图像的显示流程的耗时,缩短响应时延,提升输入操作的跟手性。

[0199] 图像合成系统将合成后的图像通过硬件合成器送至显示驱动以进行显示。

[0200] 当应用对应的总开关关闭时,不进行处理。

[0201] 综上,通过修改Vsync-APP及相应的offset来控制应用绘图的节奏,让应用能提前绘制图像的时间提前,减少图像合成系统的等待时间。

[0202] 在上述实施例的基础上,在总开关开启时,还可以执行S606。

[0203] S606、图像合成系统还取消Vsync-SF信号,将绘制渲染好的图像立即进行合成。

[0204] 可以理解的是,当关闭Vsync-SF时,绘制渲染好的图像可以立即合成,及时将合成好的图像送至硬件合成器进行处理和送显。此外,在没有新图层时进行额外预处理的资源消耗。

[0205] 这样,可以将合成的时间提前,以缩短整个图像的显示流程的耗时,缩短响应时延,提升输入操作的跟手性。

[0206] 可以理解的是,图8所示的流程中,通过修改软件APP依赖的Vsync及相应的offset来控制应用绘图的节奏,让应用能提前开始绘制图像,减少图像合成系统不需等待。同时会关闭系统侧的Vsync-SF,让应用在绘制完毕后立即调用叠加合成预处理,加快将需要送显的各图层送至图像合成系统处理,同时也减少在没有新图层时进行额外预处理的资源消耗。

[0207] 另一种实现的方式中,通过加快图像的送显流程,实现跟手性的提高。

[0208] 图9为本申请实施例中数据处理方法的流程示意图。如图9所示,在接收到输入操作后,

[0209] S701、终端设备获取配置文件。

- [0210] 配置文件可以参照上述相关描述,此处不再赘述。
- [0211] 可能的实现方式中,场景识别模块获取配置文件。
- [0212] 可能的实现方式中,配置文件可以包括应用的包名信息,以及快速送显特性的总开关的配置信息。
- [0213] S702、基于配置文件确认场景的模式信息,模式信息用于指示快速送显特性的总开关是否开启。
- [0214] 可能还包括:Vsync-APP和Vsync-SF的偏移量。
- [0215] 示例性的,基于“vendor.oneframe.hwc.no_wait_pf”判断场景的模式信息。
- [0216] S703、当硬件合成器在接收到合成好的图像,且确认快速送显特性的总开关开启时,
- [0217] 确认送显队列中是否有待处理的缓存。
- [0218] 可以理解的是,当送显队列中有待处理的缓存时,丢弃送显队列中待处理的缓存后送显当前帧。当送显队列中没有待处理的缓存时,送显当前帧。
- [0219] 当快速送显特性的总开关未开启时,按照顺序送显送显队列中待处理的缓存和当前帧的缓存。
- [0220] 可以理解的是,图9所示的流程中,在每一帧处理图形buffer送显示时判断应用对应的特性,以判断是否为特殊场景。如果该属性为打开状态,为特殊场景,则丢弃之前堆积的图形缓存buffer将当前帧的缓存立即送显,使得最新时间绘制的图像能够及时送到LCD显示。可以减少该模块对于每一帧的送显处理中前帧耗时则会导致任务堆积,减少影响后续的送显时延。
- [0221] 另一种实现的方式中,通过调度系统资源,实现跟手性的提高。示例性的,如图10所示,
- [0222] S801、终端设备获取配置文件。
- [0223] 可能的实现方式中,场景识别模块获取配置文件。
- [0224] 可能的实现方式中,配置文件可以包括应用的包名信息,以及总开关的配置信息等。
- [0225] S802、基于配置文件确认是否为帧率不回落的特殊场景。
- [0226] 基于配置文件确认场景的模式信息,场景的模式信息用于指示帧率不回落方案的特性开关是否开启。
- [0227] 模式信息还用于指示快速送显特性的总开关是否开启和Vsync-APP和Vsync-SF的偏移量等。当在帧率不回落方案的特性名称中包括相应的应用时,为帧率不回落的特殊场景。
- [0228] S803、当终端设备处于帧率不回落的特殊场景时,帧率控制模块将屏幕刷新率调节至最高。
- [0229] S804、终端设备中的相应模块按照调整后的屏幕刷新率进行绘制和叠加送显。
- [0230] 可能的实现方式中,通过判断有无前台应用运行,可以判断终端设备是否在使用。
- [0231] 当终端设备在使用中,屏幕刷新率保持默认配置。当终端设备未在使用时,调至帧率较低的配置。这样,通过帧率控制模块通过不同的模式信息控制屏幕刷新率的切换,可以有效的减少帧率过高导致的卡顿、帧率过低导致的体验较差和高帧率对智能设备功耗的消

耗。

[0232] 可以理解的是,本申请实施例的方法可以有效的节省功耗,基于配置名单中预先的配置,能够针对特定的应用APP使用场景进行优化。用户在特殊场景的使用过程中,为了给用户极致的显示延迟体验,帧率通常需要维持在最高的状态,功耗的损失大。帧率控制模块在收到场景模式识别模块传递来的开启信息时,确认帧率不回落,使得用户在特殊场景使用下的体验不受影响;在收到场景模式识别模传递到的关闭信息时,及时将帧率回落至正常值,在保证用户体验的基础上,进一步减少功耗消耗,平衡了性能和功耗之间的关系,达到节省功耗的目的。

[0233] 可以理解的是,提升CPU频率、提升GPU频率和目标任务绑定CPU大核不仅对功耗的影响较大不能长时间的运行,同时也会因为依赖于android原有的渲染机制存在硬件能力上限;在一些硬件性能较弱的设备上,通过使用该方案可以进一步的降低送显流程上的耗时,减少送显的卡顿感,提升高负载显示流程的稳定性。

[0234] 综上,本申请实施例中,针对现有技术存在的硬件能力上限,通过动态的切换设备帧率来适配用户场景的特殊需求;通过优化android原生的显示流程,在送显过程中不再等待高负载场景下的送显任务堆积,确保了上层应用绘制的图形能够及时送至LCD显示给用户,减少了卡顿感,避免了送显卡顿的堆积,增强了设备高负载下的送显稳定性;通过在android显示流程的图层叠加预处理阶段(合成阶段)调节应用侧的绘制节奏(Vsync-APP),并将Vsync-APP和Vsync-APP的差值调节在一个最优的状态,从而缩短图形绘制流程各模块的等待耗时,使得每一帧在一个硬件Vsync信号周期内完成,提升快速送显的成功率。

[0235] 用户可感知到系统卡顿的交互场景或某些场景下对任务有较强的性能要求时,本发明可广泛应用在此类场景,如打开应用内某个界面,观看视频,播放音频,滑动列表,多屏协同,手机投屏等。此外,本方案可广泛适用于手机、平板电脑等设备。

[0236] 上面已对本申请实施例的数据处理方法进行了说明,下面对本申请实施例提供的执行上述数据处理方法的终端设备进行描述。本领域技术人员可以理解,方法和装置可以相互结合和引用,本申请实施例提供的终端设备可以执行上述数据处理方法中的步骤。

[0237] 如图11所示,图11示出了本申请实施例提供的一种数据处理装置的结构示意图。该数据处理装置可以是本申请实施例中的终端设备。该数据处理装置包括:显示屏1801,显示屏用于显示图像;一个或多个处理器1802;存储器1803;多个应用程序;以及一个或多个计算机程序,其中一个或多个计算机程序被存储在存储器1803中,一个或多个计算机程序包括指令,当指令被数据处理装置执行时,使得数据处理装置执行上述数据处理方法中的步骤。

[0238] 图12为本申请实施例提供的一种数据处理装置的硬件结构示意图。请参见图12,该装置包括:存储器1901、处理器1902和接口电路1903。该装置还可以包括显示屏1904,其中,存储器1901、处理器1902、接口电路1903和显示屏1904可以通信;示例性的,存储器1901、处理器1902、接口电路1903和显示屏1904可以通过通信总线通信,存储器1901用于存储计算机执行指令,由处理器1902来控制执行,并由接口电路1903来执行通信,从而实现本申请实施例提供的数据处理方法。

[0239] 可选的,接口电路1903还可以包括发送器和/或接收器。可选的,上述处理器1902可以包括一个或多个CPU,还可以是其他通用处理器、数字信号处理器(digital signal

processor,DSP)、专用集成电路(application specific integrated circuit,ASIC)等。通用处理器可以是微处理器或者该处理器也可以是任何常规的处理器等。结合本申请所公开的方法的步骤可以直接体现为硬件处理器执行完成,或者用处理器中的硬件及软件模块组合执行完成。

[0240] 可能的实现方式中,本申请实施例中的计算机执行指令也可以称之为应用程序代码,本申请实施例对此不作具体限定。

[0241] 本申请实施例提供的数据处理装置,用于执行上述实施例的数据处理方法,技术原理和技术效果相似,此处不再赘述。

[0242] 本申请实施例提供一种终端设备,结构参见图1。终端设备的存储器可用于存储至少一个程序指令,处理器用于执行至少一个程序指令,以实现上述方法实施例的技术方案。其实现原理和技术效果与上述方法相关实施例类似,此处不再赘述。

[0243] 本申请实施例提供一种芯片。芯片包括处理器,处理器用于调用存储器中的计算机程序,以执行上述实施例中的技术方案。其实现原理和技术效果与上述相关实施例类似,此处不再赘述。

[0244] 本申请实施例提供一种计算机程序产品,当计算机程序产品在电子设备运行时,使得终端设备执行上述实施例中的技术方案。其实现原理和技术效果与上述相关实施例类似,此处不再赘述。

[0245] 本申请实施例提供一种计算机可读存储介质,其上存储有程序指令,程序指令被终端设备执行时,使得终端设备执行上述实施例的技术方案。其实现原理和技术效果与上述相关实施例类似,此处不再赘述。

[0246] 本申请实施例是参照根据本申请实施例的方法、设备(系统)、和计算机程序产品的流程图和/或方框图来描述的。应理解可由计算机程序指令实现流程图和/或方框图中的每一流程和/或方框、以及流程图和/或方框图中的流程和/或方框的结合。可提供这些计算机程序指令到通用计算机、专用计算机、嵌入式处理机或其他可编程数据处理设备的处理单元以产生一个机器,使得通过计算机或其他可编程数据处理设备的处理单元执行的指令产生用于实现在流程图一个流程或多个流程和/或方框图一个方框或多个方框中指定的功能的装置。

[0247] 以上的具体实施方式,对本发明的目的、技术方案和有益效果进行了进一步详细说明,所应理解的是,以上仅为本发明的具体实施方式而已,并不用于限定本发明的保护范围,凡在本发明的技术方案的基础之上,所做的任何修改、等同替换、改进等,均应包括在本发明的保护范围之内。

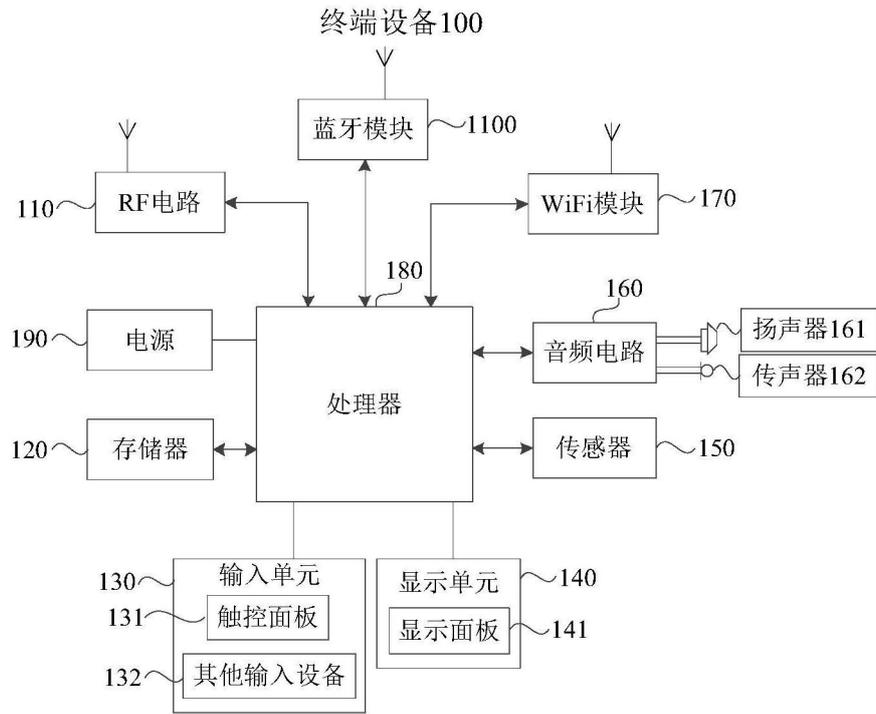


图1

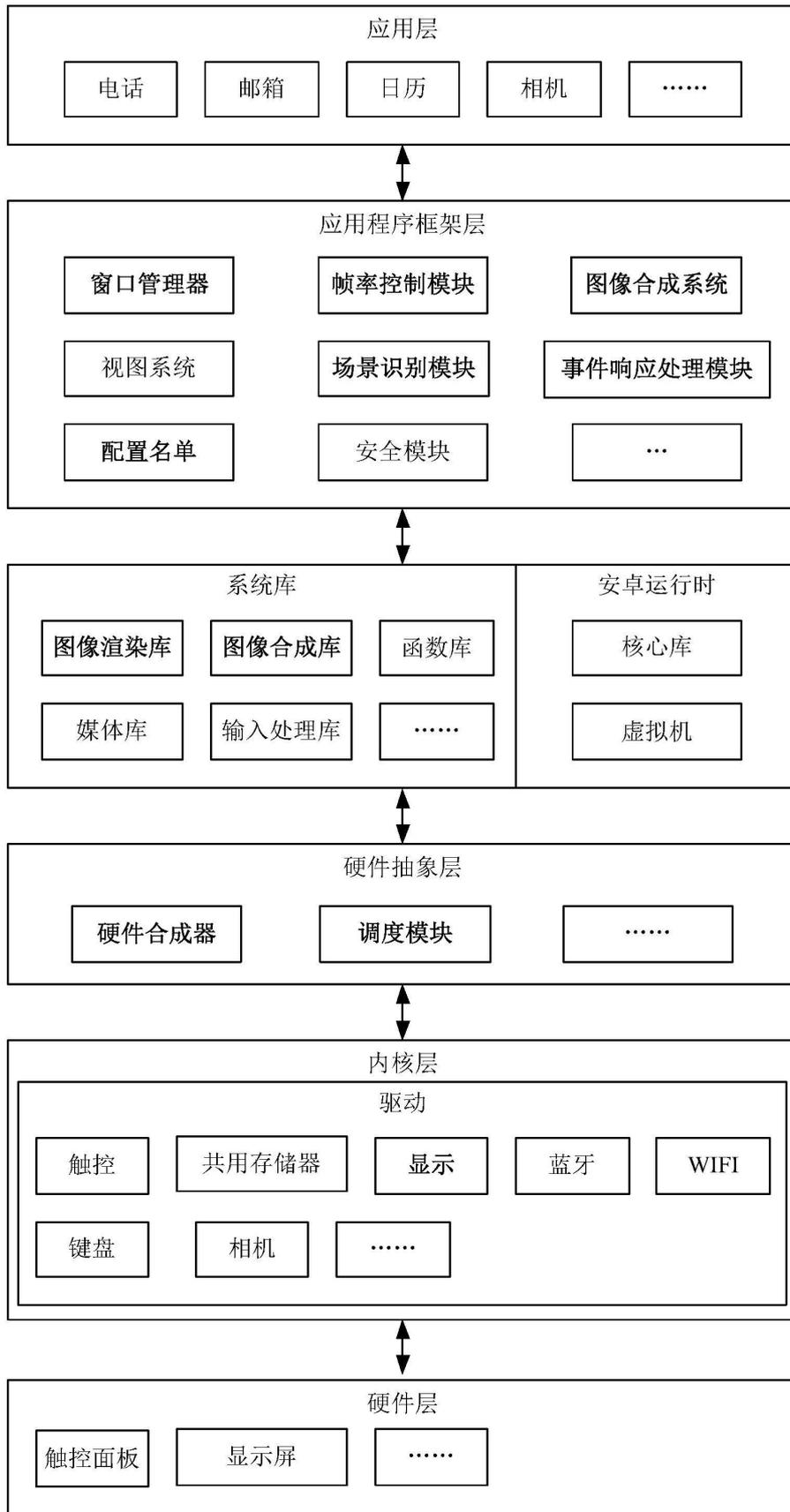


图2

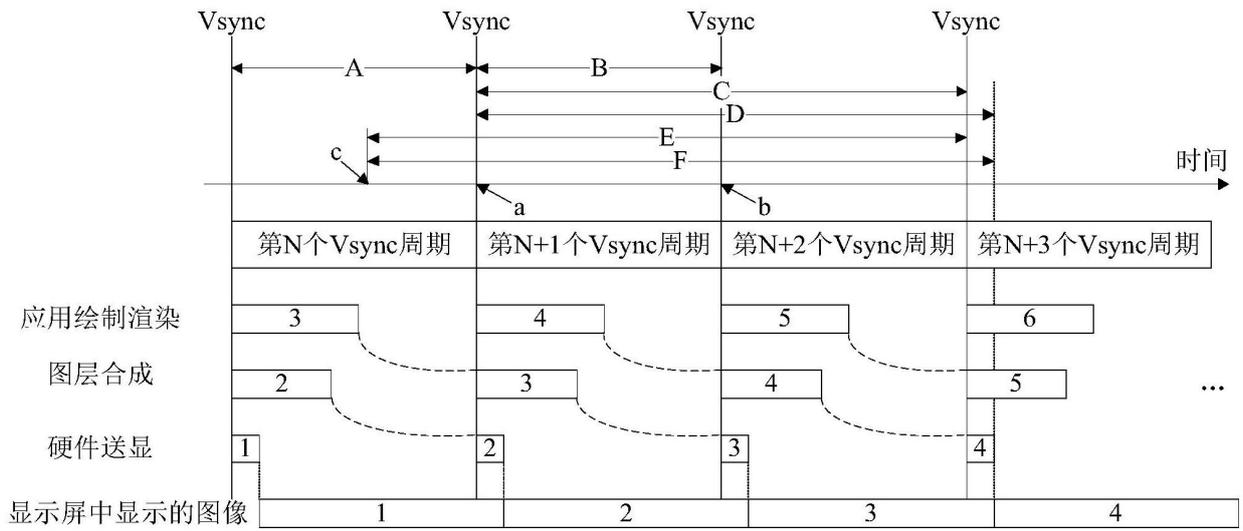


图3

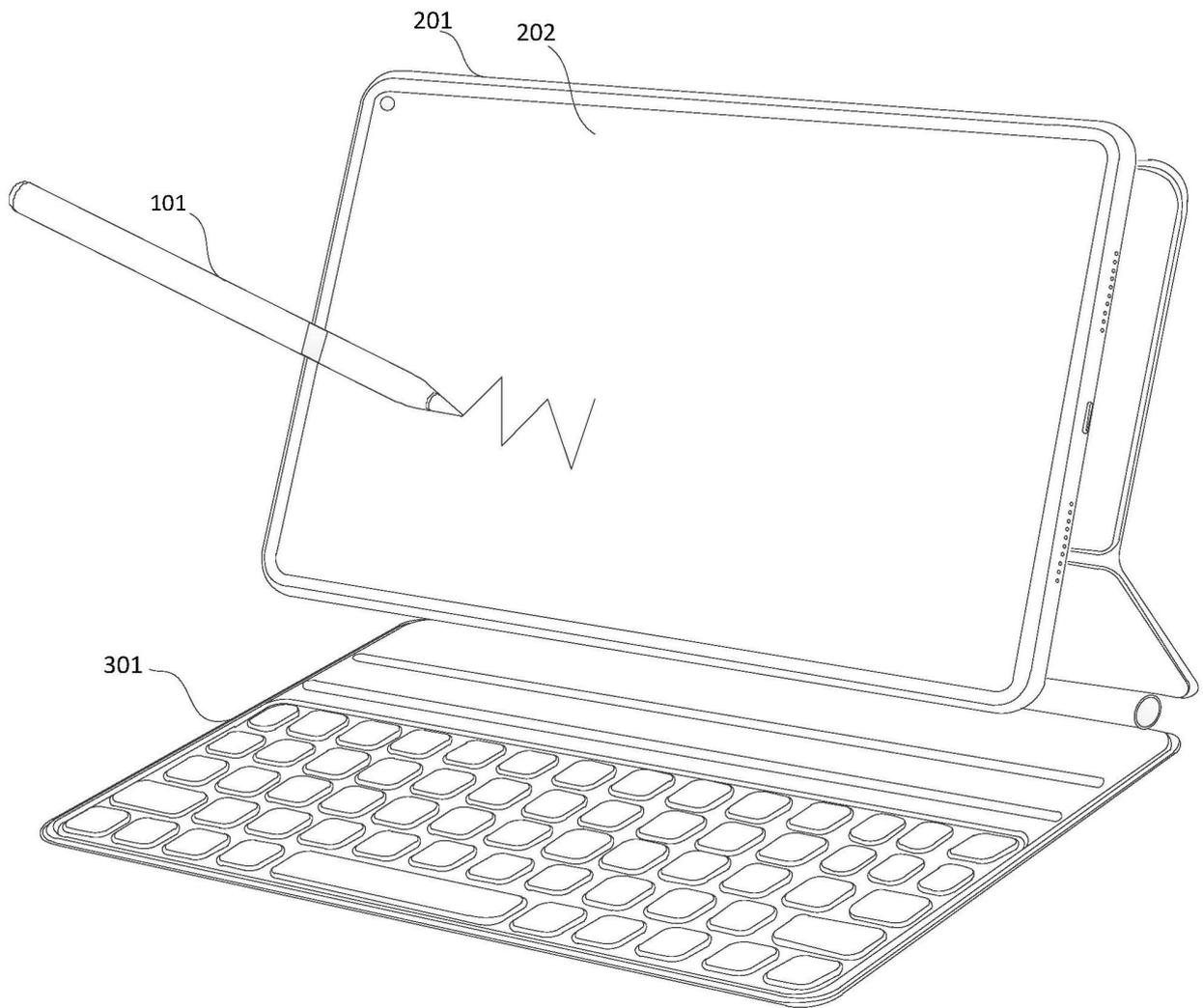


图4

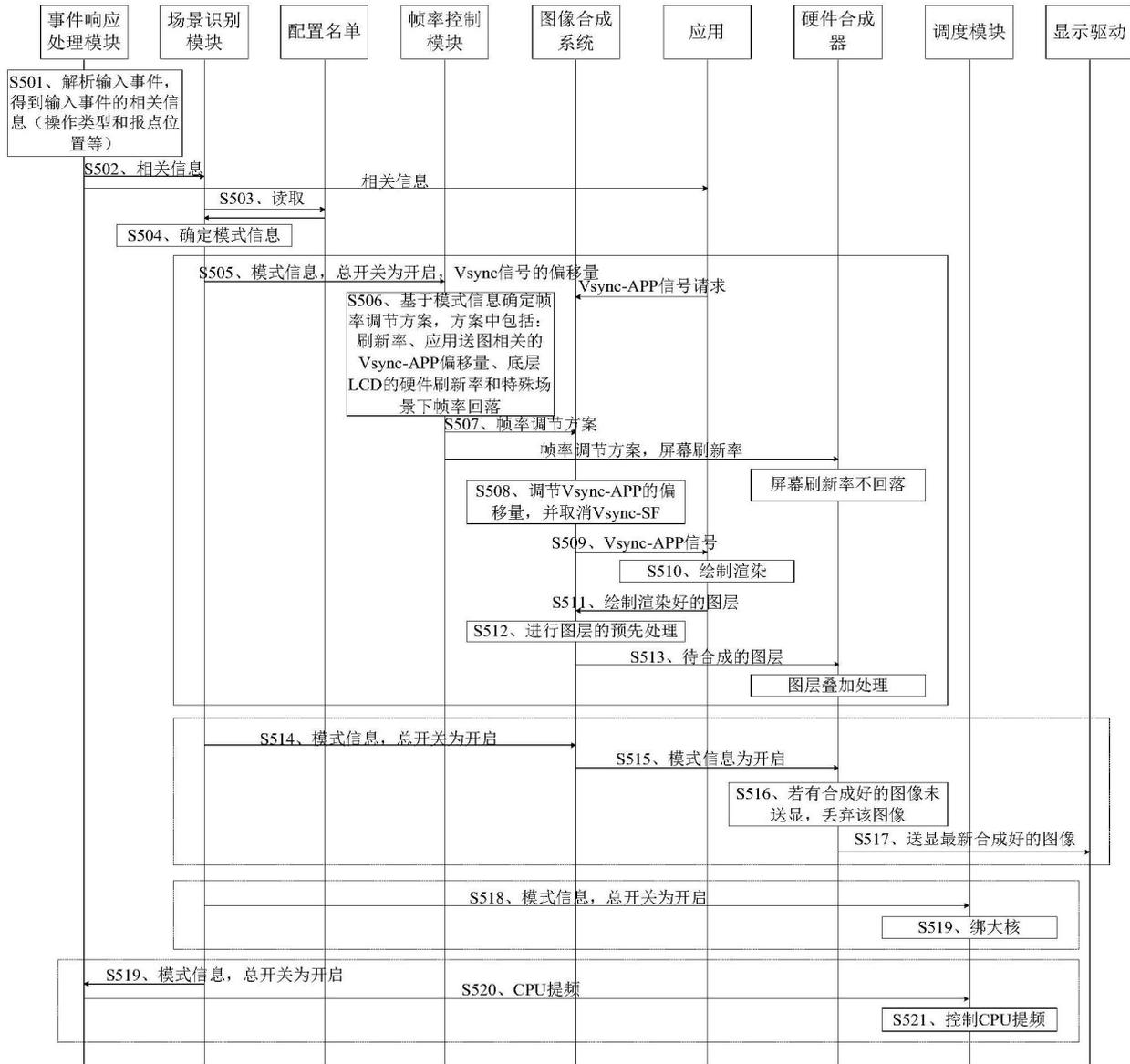


图5

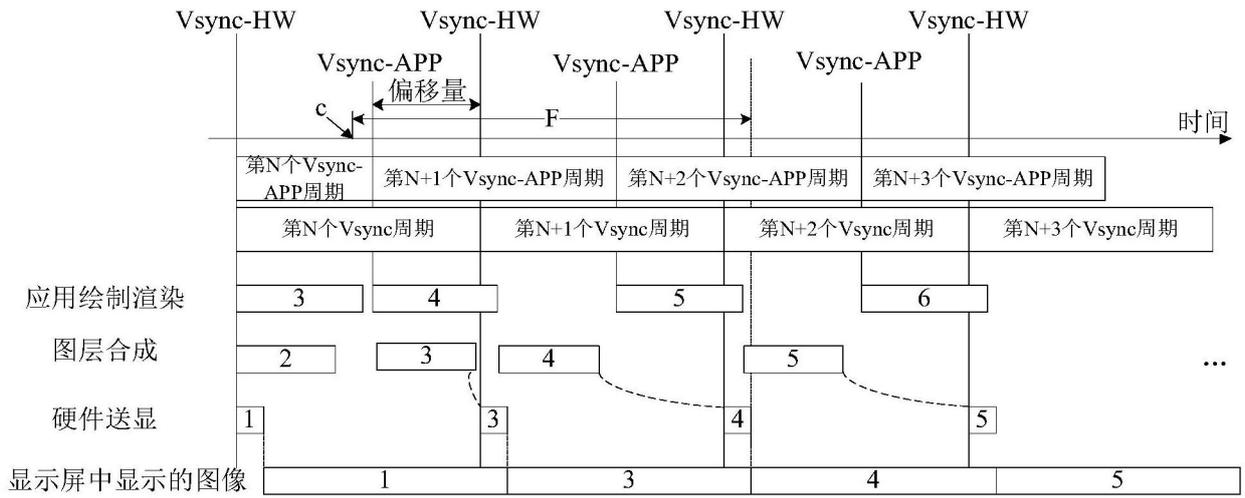


图6

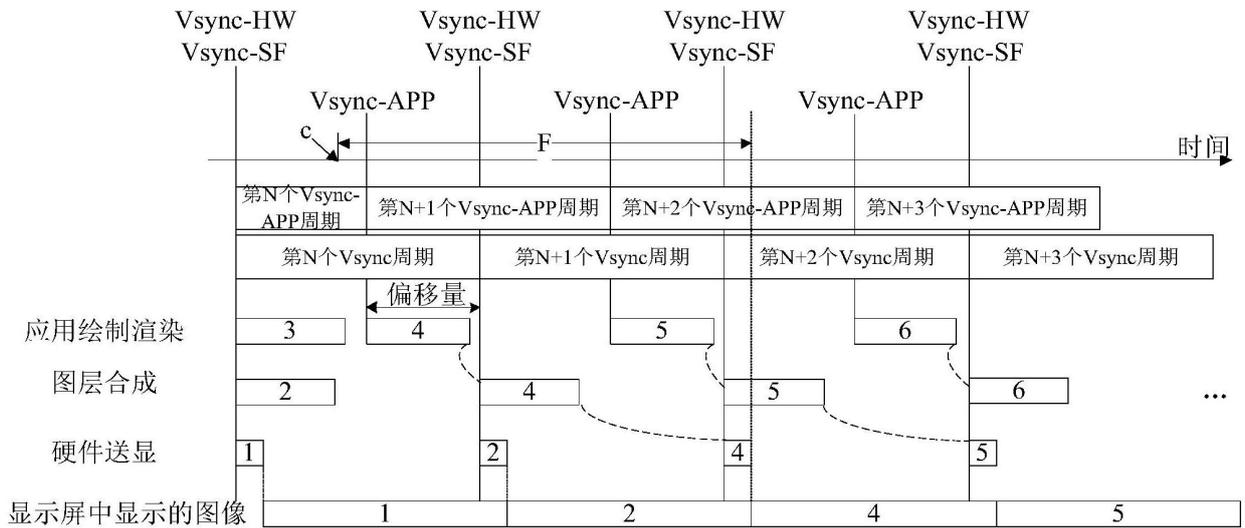


图7

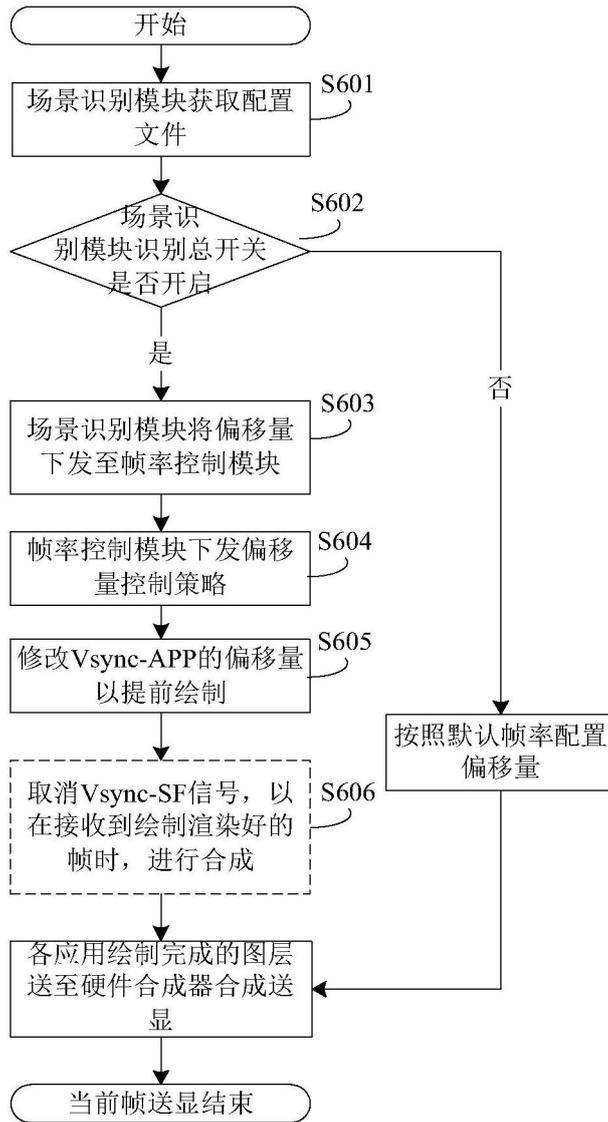


图8

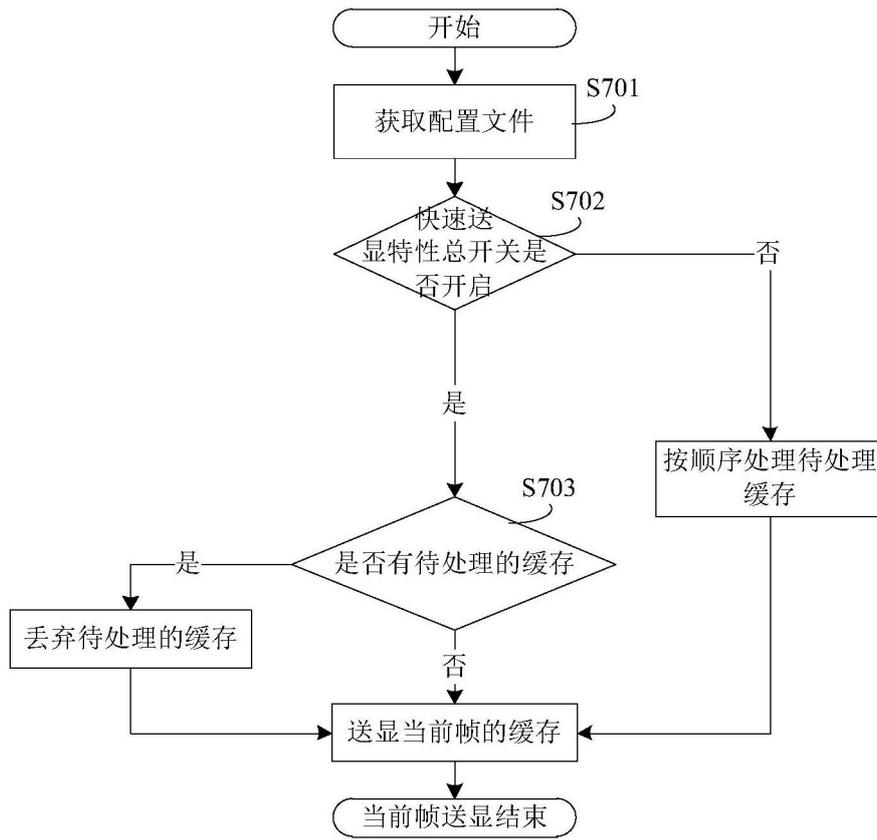


图9

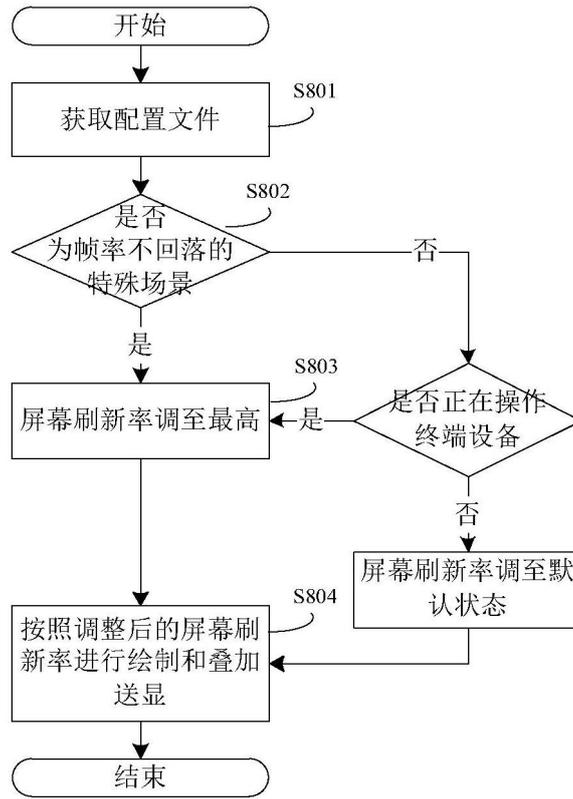


图10

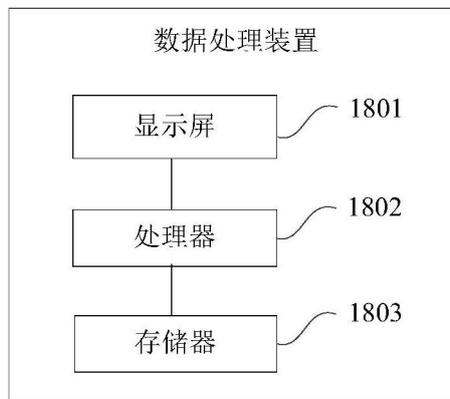


图11

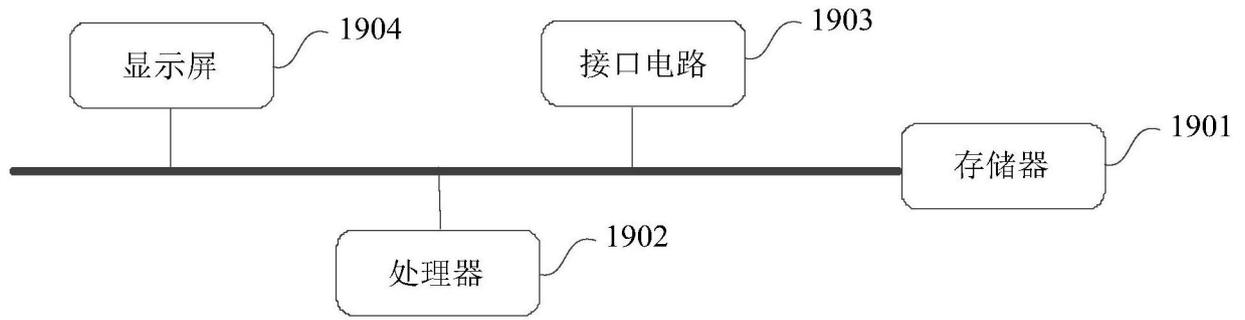


图12