



(12) 发明专利

(10) 授权公告号 CN 107220019 B

(45) 授权公告日 2021.01.08

(21) 申请号 201710338807.X

(22) 申请日 2017.05.15

(65) 同一申请的已公布的文献号  
申请公布号 CN 107220019 A

(43) 申请公布日 2017.09.29

(73) 专利权人 固安县朔程燃气有限公司  
地址 065000 河北省廊坊市固安县温泉园  
区南陈村254号

(72) 发明人 蔡创业 汪飞

(74) 专利代理机构 北京睿博行远知识产权代理  
有限公司 11297

代理人 龚家骅

(51) Int. Cl.

G06F 3/14 (2006.01)

G06F 9/50 (2006.01)

(56) 对比文件

CN 106652919 A, 2017.05.10

CN 104168487 A, 2014.11.26

CN 103137104 A, 2013.06.05

CN 106296566 A, 2017.01.04

US 2017006230 A1, 2017.01.05

CN 106447595 A, 2017.02.22

CN 106657681 A, 2017.05.10

CN 106484348 A, 2017.03.08

审查员 白硕

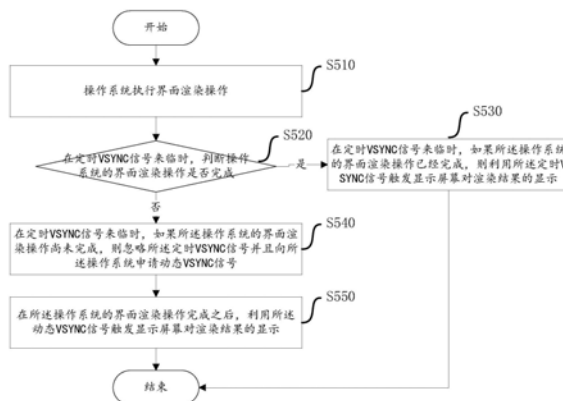
权利要求书2页 说明书12页 附图5页

(54) 发明名称

一种基于动态VSYNC信号的渲染方法、移动终端及存储介质

(57) 摘要

本发明提供一种基于动态VSYNC信号的渲染方法、移动终端及存储介质。在定时VSYNC信号来临时,判断操作系统的界面渲染操作是否完成;如果所述界面渲染操作尚未完成,则忽略所述定时VSYNC信号并且向所述操作系统申请动态VSYNC信号;在所述界面渲染操作完成之后,利用所述动态VSYNC信号触发显示屏幕对渲染结果的显示。本发明在不影响基于定时VSYNC信号的刷新机制的前提下,在操作系统负载较大无法及时完成界面渲染操作时,可以暂时忽略定时VSYNC信号,在操作系统的界面渲染操作执行完毕之后,触发动态VSYNC信号,及时将已经延误显示的渲染结果在显示屏幕中显示出来,提升了画面流畅度。



1. 一种基于动态垂直同步VSYNC信号的渲染方法,其特征在于,包括:

在定时VSYNC信号来临时,判断操作系统的界面渲染操作是否完成,所述定时VSYNC信号为固定频率的信号;

如果所述操作系统的界面渲染操作已完成,则利用所述定时VSYNC信号触发显示屏幕对渲染结果的显示;

如果所述界面渲染操作尚未完成,则忽略所述定时VSYNC信号并且向所述操作系统申请动态VSYNC信号,所述动态VSYNC信号是操作系统根据需要临时发出的VSYNC信号,该动态VSYNC信号使得显卡生成帧的速度和显示屏幕的刷新速度不一致,该动态VSYNC信号可以在相邻两个定时VSYNC信号之间的时间段内触发显示屏幕刷新;

在所述界面渲染操作完成之后,利用所述动态VSYNC信号触发显示屏幕对渲染结果的显示。

2. 如权利要求1所述的方法,其特征在于,所述向所述操作系统申请动态VSYNC信号,包括:

向所述操作系统注册用于触发所述动态VSYNC信号的回调函数。

3. 如权利要求2所述的方法,其特征在于,所述向所述操作系统注册用于触发动态VSYNC信号的回调函数,包括:

向所述操作系统中的硬件合成器HWC注册用于触发所述动态VSYNC信号的回调函数。

4. 如权利要求2或3所述的方法,其特征在于,所述在所述界面渲染操作完成之后,利用所述动态VSYNC信号触发显示屏幕对渲染结果的显示,包括:

在所述操作系统的界面渲染操作完成之后,通过调用所述回调函数,触发所述操作系统发出所述动态VSYNC信号,以便使所述动态VSYNC信号触发显示屏幕对渲染结果的显示。

5. 一种移动终端,其特征在于,所述移动终端包括处理器和存储器;

所述处理器用于执行所述存储器中存储的基于动态垂直同步VSYNC信号的渲染程序,以实现以下步骤:

在定时VSYNC信号来临时,判断操作系统的界面渲染操作是否完成,所述定时VSYNC信号为固定频率的信号;

如果所述操作系统的界面渲染操作已完成,则利用所述定时VSYNC信号触发显示屏幕对渲染结果的显示;

如果所述界面渲染操作尚未完成,则忽略所述定时VSYNC信号并向所述操作系统申请动态VSYNC信号,所述动态VSYNC信号是操作系统根据需要临时发出的VSYNC信号,该动态VSYNC信号使得显卡生成帧的速度和显示屏幕的刷新速度不一致,该动态VSYNC信号可以在相邻两个定时VSYNC信号之间的时间段内触发显示屏幕刷新;

在所述界面渲染操作完成之后,利用所述动态VSYNC信号触发显示屏幕对渲染结果的显示。

6. 如权利要求5所述的移动终端,其特征在于,所述处理器还用于执行所述存储器中存储的基于动态VSYNC信号的渲染程序,以实现以下步骤:

向所述操作系统注册用于触发所述动态VSYNC信号的回调函数;

在所述操作系统的界面渲染操作完成之后,通过调用所述回调函数,触发所述操作系统发出所述动态VSYNC信号,以便使所述动态VSYNC信号触发显示屏幕对渲染结果的显示。

7. 如权利要求6所述的移动终端,其特征在于,所述处理器还用于执行存储器中存储的基于动态VSYNC信号的渲染程序,以实现以下步骤:

向所述操作系统中的硬件合成器HWC注册用于触发所述动态VSYNC信号的回调函数。

8. 一种存储介质,其特征在于,所述存储介质存储有一个或者多个程序,所述一个或者多个程序可被一个或者多个处理器执行,以实现权利要求1~4任一项所述的基于动态VSYNC信号的渲染方法。

## 一种基于动态VSYNC信号的渲染方法、移动终端及存储介质

### 技术领域

[0001] 本发明涉及终端技术领域,尤其涉及一种基于动态VSYNC信号的渲染方法、移动终端及存储介质。

### 背景技术

[0002] 目前,软件设计人员希望其设计的软件能够采用丰富的动画、图片等元素来展现软件产品特性。然而,相比较IOS操作系统,Android操作系统经常出现卡顿,这一问题最主要的根源在于系统渲染性能不及时导致的掉帧/丢帧。

[0003] 如图1所示,为现有技术中正常的渲染过程示意图。

[0004] Display为显示屏幕,GPU(Graphics Processing Unit,图形处理器)和CPU(Central Processing Unit,中央处理器)负责渲染帧数据,每帧数据以方框表示并以数字进行编号,如0、1、2、3和4。Android操作系统每隔16.6ms发出VSYNC信号(也可以写作VSync信号),触发对UI(User Interface,用户界面)的渲染。Display显示第0帧图像时,CPU和GPU渲染第1帧图像,Display在第0帧图像显示完成后,第1个VSYNC信号到来,Display显示第1帧图像,CPU和GPU开始渲染第2帧图像,并且在第2个VSYNC信号到来时,Display正常显示第2帧图像,以此类推,CPU和GPU都可以在两个VSYNC信号之间的16.6ms内完成渲染。

[0005] 如果每次渲染都成功,这样就能够达到流畅的画面所需要的60fps(Frames Per Second,每秒传输帧数),为了能够实现60fps,这意味着界面渲染操作都必须在16.6ms内完成,但是CPU和GPU很有可能无法在16.6ms内及时地完成复杂的界面渲染操作,操作时间超出16.6ms越多,丢帧现象越严重。

[0006] 如图2所示,为现有技术中出现卡顿的渲染过程示意图。由于某些原因,例如:CPU资源被占用忙于执行其他计算密集型的任务,或者GPU渲染过于复杂的动画画面,没有及时地开始处理第2帧图像,直到第2个VSYNC信号即将来临时开始处理,那么CPU和GPU的渲染耗时将超出16.6ms,在第2个VSYNC信号到来时,由于第2帧图像还没有准备就绪,所以Display不能显示第2帧图像,进而出现丢帧现象,并会影响CPU和GPU对下一帧以及后续多帧图像的处理,对于这种现象用户会感觉画面卡顿,而且在性能较低的Android设备或者动画效果复杂的App(Application,应用程序)上,卡顿现象更为严重,画面流畅度差,影响用户的体验效果。

### 发明内容

[0007] 本发明的主要目的在于提出一种基于动态VSYNC信号的渲染方法、移动终端及存储介质,旨在解决现有技术中以固定频率(16.6ms)发出VSYNC信号,如果操作系统无法在两个VSYNC信号之间的时间内及时地完成界面渲染操作,则会导致画面流畅度差的问题。

[0008] 针对上述技术问题,本发明是通过以下技术方案来解决的:

[0009] 本发明提供了一种基于动态垂直同步VSYNC信号的渲染方法,包括:在定时VSYNC信号来临时,判断操作系统的界面渲染操作是否完成;如果所述界面渲染操作尚未完成,则

忽略所述定时VSYNC信号并且向所述操作系统申请动态VSYNC信号;在所述界面渲染操作完成之后,利用所述动态VSYNC信号触发显示屏幕对渲染结果的显示。

[0010] 可选地,所述方法还包括:在所述定时VSYNC信号来临时,如果所述操作系统的界面渲染操作已经完成,则利用所述定时VSYNC信号触发显示屏幕对渲染结果的显示。

[0011] 可选地,所述向所述操作系统申请动态VSYNC信号,包括:向所述操作系统注册用于触发所述动态VSYNC信号的回调函数。

[0012] 可选地,所述向所述操作系统注册用于触发动态VSYNC信号的回调函数,包括:向所述操作系统中的硬件合成器HWC注册用于触发所述动态VSYNC信号的回调函数。

[0013] 可选地,所述在所述界面渲染操作完成之后,利用所述动态VSYNC信号触发显示屏幕对渲染结果的显示,包括:在所述操作系统的界面渲染操作完成之后,通过调用所述回调函数,触发所述操作系统发出所述动态VSYNC信号,以便使所述动态VSYNC信号触发显示屏幕对渲染结果的显示。

[0014] 本发明还提供了一种移动终端,所述移动终端包括处理器和存储器;所述处理器用于执行所述存储器中存储的基于动态垂直同步VSYNC信号的渲染程序,以实现以下步骤:在定时VSYNC信号来临时,判断操作系统的界面渲染操作是否完成;如果所述界面渲染操作尚未完成,则忽略所述定时VSYNC信号并向所述操作系统申请动态VSYNC信号;在所述界面渲染操作完成之后,利用所述动态VSYNC信号触发显示屏幕对渲染结果的显示。

[0015] 可选地,所述在所述定时VSYNC信号来临时,所述处理器还用于执行所述存储器中存储的基于动态VSYNC信号的渲染程序,以实现以下步骤:在所述定时VSYNC信号来临时,如果所述操作系统的界面渲染操作已经完成,则利用所述定时VSYNC信号触发显示屏幕对渲染结果的显示。

[0016] 可选地,所述处理器还用于执行所述存储器中存储的基于动态VSYNC信号的渲染程序,以实现以下步骤:向所述操作系统注册用于触发所述动态VSYNC信号的回调函数;在所述操作系统的界面渲染操作完成之后,通过调用所述回调函数,触发所述操作系统发出所述动态VSYNC信号,以便使所述动态VSYNC信号触发显示屏幕对渲染结果的显示。

[0017] 可选地,所述处理器还用于执行存储器中存储的基于动态VSYNC信号的渲染程序,以实现以下步骤:向所述操作系统中的硬件合成器HWC注册用于触发所述动态VSYNC信号的回调函数。

[0018] 本发明还提供了一种存储介质,所述存储介质存储有一个或者多个程序,所述一个或者多个程序可被一个或者多个处理器执行,以实现上述基于动态VSYNC信号的渲染方法。

[0019] 本发明所提出的基于动态VSYNC的渲染方法、移动终端及存储介质具有以下有益效果:

[0020] 本发明在不影响基于定时VSYNC信号的刷新机制的前提下,在操作系统负载较大无法及时完成界面渲染操作时,可以暂时忽略定时VSYNC信号,在操作系统的界面渲染操作执行完毕之后,触发动态VSYNC信号,及时将已经延误显示的渲染结果在显示屏幕中显示出来,提升了画面流畅度。

## 附图说明

- [0021] 图1为现有技术中正常的渲染过程示意图；
- [0022] 图2为现有技术中出现卡顿的渲染过程示意图；
- [0023] 图3为实现本发明各个实施例一可选的移动终端的硬件结构示意图；
- [0024] 图4为如图3所示的移动终端的无线通信系统示意图；
- [0025] 图5为根据本发明第一实施例的基于动态VSYNC信号的渲染方法的流程图；
- [0026] 图6为根据本发明第二实施例的基于动态VSYNC信号的渲染方法的流程图；
- [0027] 图7为根据本发明第三实施例的基于动态VSYNC信号的渲染方法的示意图；
- [0028] 图8为根据本发明第四实施例的向操作系统申请动态VSYNC信号的示意图；
- [0029] 图9为根据本发明第五实施例的移动终端的结构图。
- [0030] 本发明目的的实现、功能特点及优点将结合实施例，参照附图做进一步说明。

## 具体实施方式

- [0031] 应当理解，此处所描述的具体实施例仅仅用以解释本发明，并不用于限定本发明。
- [0032] 在后续的描述中，使用用于表示元件的诸如“模块”、“部件”或“单元”的后缀仅为为了有利于本发明的说明，其本身没有特定的意义。因此，“模块”、“部件”或“单元”可以混合地使用。
- [0033] 终端可以以各种形式来实施。例如，本发明中描述的终端可以包括诸如手机、平板电脑、笔记本电脑、掌上电脑、个人数字助理(Personal Digital Assistant, PDA)、便携式媒体播放器(Portable Media Player, PMP)、导航装置、可穿戴设备、智能手环、计步器等移动终端，以及诸如数字TV、台式计算机等固定终端。
- [0034] 后续描述中将以移动终端为例进行说明，本领域技术人员将理解的是，除了特别用于移动目的的元素之外，根据本发明的实施方式的构造也能够应用于固定类型的终端。
- [0035] 请参阅图3，其为实现本发明各个实施例的一种移动终端的硬件结构示意图，该移动终端100可以包括：RF(Radio Frequency, 射频)单元101、WiFi模块102、音频输出单元103、A/V(音频/视频)输入单元104、传感器105、显示单元106、用户输入单元107、接口单元108、存储器109、处理器110、以及电源111等部件。本领域技术人员可以理解，图3中示出的移动终端结构并不构成对移动终端的限定，移动终端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。
- [0036] 下面结合图3对移动终端的各个部件进行具体的介绍：
- [0037] 射频单元101可用于收发信息或通话过程中，信号的接收和发送，具体的，将基站的下行信息接收后，给处理器110处理；另外，将上行的数据发送给基站。通常，射频单元101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、双工器等。此外，射频单元101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GSM(Global System of Mobile communication, 全球移动通讯系统)、GPRS(General Packet Radio Service, 通用分组无线服务)、CDMA2000(Code Division Multiple Access 2000, 码分多址2000)、WCDMA(Wideband Code Division Multiple Access, 宽带码分多址)、TD-SCDMA(Time Division-Synchronous Code Division Multiple Access, 时分同步码分多址)、FDD-LTE(Frequency Division

Duplexing-Long Term Evolution,频分双工长期演进)和TDD-LTE(Time Division Duplexing-Long Term Evolution,分时双工长期演进)等。

[0038] WiFi属于短距离无线传输技术,移动终端通过WiFi模块102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3示出了WiFi模块102,但是可以理解的是,其并不属于移动终端的必须构成,完全可以根据需要在不改变发明的本质的范围内而省略。

[0039] 音频输出单元103可以在移动终端100处于呼叫信号接收模式、通话模式、记录模式、语音识别模式、广播接收模式等等模式下时,将射频单元101或WiFi模块102接收的或者在存储器109中存储的音频数据转换成音频信号并且输出为声音。而且,音频输出单元103还可以提供与移动终端100执行的特定功能相关的音频输出(例如,呼叫信号接收声音、消息接收声音等等)。音频输出单元103可以包括扬声器、蜂鸣器等等。

[0040] A/V输入单元104用于接收音频或视频信号。A/V输入单元104可以包括图形处理器(Graphics Processing Unit,GPU)1041和麦克风1042,图形处理器1041对在视频捕获模式或图像捕获模式中由图像捕获装置(如摄像头)获得的静态图片或视频的图像数据进行处理。处理后的图像帧可以显示在显示单元106上。经图形处理器1041处理后的图像帧可以存储在存储器109(或其它存储介质)中或者经由射频单元101或WiFi模块102进行发送。麦克风1042可以在电话通话模式、记录模式、语音识别模式等等运行模式中经由麦克风1042接收声音(音频数据),并且能够将这样的声音处理为音频数据。处理后的音频(语音)数据可以在电话通话模式的情况下转换为可经由射频单元101发送到移动通信基站的格式输出。麦克风1042可以实施各种类型的噪声消除(或抑制)算法以消除(或抑制)在接收和发送音频信号的过程中产生的噪声或者干扰。

[0041] 移动终端100还包括至少一种传感器105,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1061的亮度,接近传感器可在移动终端100移动到耳边时,关闭显示面板1061和/或背光。作为运动传感器的一种,加速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,可用于识别手机姿态的应用(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;至于手机还可配置的指纹传感器、压力传感器、虹膜传感器、分子传感器、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。

[0042] 显示单元106用于显示由用户输入的信息或提供给用户的信息。显示单元106可包括显示面板1061,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1061。

[0043] 用户输入单元107可用于接收输入的数字或字符信息,以及产生与移动终端的用户设置以及功能控制有关的键信号输入。具体地,用户输入单元107可包括触控面板1071以及其他输入设备1072。触控面板1071,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1071上或在触控面板1071附近的操作),并根据预先设定的程式驱动相应的连接装置。触控面板1071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110,并能接收处理器110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1071。除了触控面板1071,用户输入单元107还可以包括其他输入设备1072。具体地,其他输入设备1072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,具体此处不做限定。

[0044] 进一步的,触控面板1071可覆盖显示面板1061,当触控面板1071检测到在其上或附近的触摸操作后,传送给处理器110以确定触摸事件的类型,随后处理器110根据触摸事件的类型在显示面板1061上提供相应的视觉输出。虽然在图3中,触控面板1071与显示面板1061是作为两个独立的部件来实现移动终端的输入和输出功能,但是在某些实施例中,可以将触控面板1071与显示面板1061集成而实现移动终端的输入和输出功能,具体此处不做限定。

[0045] 接口单元108用作至少一个外部装置与移动终端100连接可以通过的接口。例如,外部装置可以包括有线或无线头戴式耳机端口、外部电源(或电池充电器)端口、有线或无线数据端口、存储卡端口、用于连接具有识别模块的装置的端口、音频输入/输出(I/O)端口、视频I/O端口、耳机端口等等。接口单元108可以用于接收来自外部装置的输入(例如,数据信息、电力等等)并且将接收到的输入传输到移动终端100内的一个或多个元件或者可以用于在移动终端100和外部装置之间传输数据。

[0046] 存储器109可用于存储软件程序以及各种数据。存储器109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、图像播放功能等等);存储数据区可存储根据手机的使用所创建的数据(比如音频数据、电话本等等)等。此外,存储器109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。

[0047] 处理器110是移动终端的控制中心,利用各种接口和线路连接整个移动终端的各个部分,通过运行或执行存储在存储器109内的软件程序和/或模块,以及调用存储在存储器109内的数据,执行移动终端的各种功能和处理数据,从而对移动终端进行整体监控。处理器110可包括一个或多个处理单元;优选的,处理器110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110中。

[0048] 移动终端100还可以包括给各个部件供电的电源111(比如电池),优选的,电源111可以通过电源管理系统与处理器110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。

[0049] 尽管图3未示出,移动终端100还可以包括蓝牙模块等,在此不再赘述。

[0050] 为了便于理解本发明实施例,下面对本发明的移动终端所基于的通信网络系统进行描述。

[0051] 请参阅图4,图4为本发明实施例提供的一种通信网络系统架构图,该通信网络系统为通用移动通信技术的LTE系统,该LTE系统包括依次通讯连接的UE(User Equipment,用户设备)201,E-UTRAN(Evolved UMTS Terrestrial Radio Access Network,演进式UMTS陆地无线接入网)202,EPC(Evolved Packet Core,演进式分组核心网)203和运营商的IP业务204。



[0052] 具体地,UE201可以是上述终端100,此处不再赘述。

[0053] E-UTRAN202包括eNodeB2021和其它eNodeB2022等。其中,eNodeB2021可以通过回程(backhaul)(例如X2接口)与其它eNodeB2022连接,eNodeB2021连接到EPC203,eNodeB2021可以提供UE201到EPC203的接入。

[0054] EPC203可以包括MME(Mobility Management Entity,移动性管理实体)2031,HSS(Home Subscriber Server,归属用户服务器)2032,其它MME2033,SGW(Serving GateWay,服务网关)2034,PGW(PDN GateWay,分组数据网络网关)2035和PCRF(Policy and Charging Rules Function,政策和资费功能实体)2036等。其中,MME2031是处理UE201和EPC203之间信令的控制节点,提供承载和连接管理。HSS2032用于提供一些寄存器来管理诸如归属位置寄存器(图中未示)之类的功能,并且保存有一些有关服务特征、数据速率等用户专用的信息。所有用户数据都可以通过SGW2034进行发送,PGW2035可以提供UE 201的IP地址分配以及其它功能,PCRF2036是业务数据流和IP承载资源的策略与计费控制策略决策点,它为策略与计费执行功能单元(图中未示)选择及提供可用的策略和计费控制决策。

[0055] IP业务204可以包括因特网、内联网、IMS(IP Multimedia Subsystem,IP多媒体子系统)或其它IP业务等。

[0056] 虽然上述以LTE系统为例进行了介绍,但本领域技术人员应当知晓,本发明不仅仅适用于LTE系统,也可以适用于其他无线通信系统,例如GSM、CDMA2000、WCDMA、TD-SCDMA以及未来新的网络系统等,此处不做限定。

[0057] 基于上述移动终端硬件结构以及通信网络系统,提出本发明方法各个实施例。

[0058] 本发明提供一种基于动态VSYNC(Vertical Synchronization,垂直同步)信号的渲染方法、移动终端及存储介质。本发明在不影响基于定时VSYNC信号的刷新机制的前提下,在操作系统可以及时完成界面渲染操作的情况下,利用定时VSYNC信号来触发显示屏幕对渲染结果的显示;在操作系统负载较大无法及时完成界面渲染操作的情况下,可以暂时忽略定时VSYNC信号,在操作系统的界面渲染操作执行完毕之后,触发动态VSYNC信号,并利用该动态VSYNC信号及时将已经延误显示的渲染结果在显示屏幕中显示出来,提升了画面流畅度。

[0059] 在本发明中涉及定时VSYNC信号和动态VSYCN信号;其中,

[0060] 定时VSYNC信号是使显卡的输出帧数和显示屏幕的垂直刷新率相同,进而使得显卡生成帧的速度和显示屏幕的刷新速度保持一致。相邻两个定时VSYNC之间的时间间隔为16.6ms。

[0061] 动态VSYNC信号是操作系统根据需要临时发出的VSYNC信号,该动态VSYNC信号使得显卡生成帧的速度和显示屏幕的刷新速度不一致,该动态VSYNC信号可以在相邻两个定时VSYNC信号之间的时间段内触发显示屏幕刷新,及时将已经延误显示的渲染结果在显示屏幕中显示出来。

[0062] 以下结合附图以及实施例,对本发明进行进一步详细说明。本领域技术人员应当理解是,此处所描述的具体实施例仅仅用以解释本发明,而并不限定本发明。

[0063] 实施例一

[0064] 本发明实施例提供一种基于动态VSYNC信号的渲染方法。本实施例的执行主体为移动终端。本实施例的移动终端包括但不限于智能手机、平板电脑。

[0065] 如图5所示,图5为根据本发明第一实施例的基于动态VSYNC信号的渲染方法的流程图。

[0066] 步骤S510,操作系统执行界面渲染操作。

[0067] 在本实施例中,操作系统为Android操作系统。

[0068] 在本实施例中,渲染的过程由CPU与GPU协作完成,Android操作系统中的CPU和GPU,在VSYNC信号来临之前,开始执行界面渲染操作。该VSYNC信号为定时VSYNC信号或者动态VSYNC信号。

[0069] 操作系统执行的界面渲染操作,具体包括:CPU执行计算处理,CPU的计算处理包括但不限于:视图的创建、布局计算、图片解码、文本绘制等;CPU会将计算结果提交到GPU,由GPU执行变换、合成、渲染等处理,获得渲染结果,并将渲染结果提交帧缓冲区,等待VSync信号到来。

[0070] 步骤S520,在定时VSYNC信号来临时,判断操作系统的界面渲染操作是否完成;如果是,则执行步骤S530;如果不是,则执行步骤S540。

[0071] 在本实施例中,定时VSYNC信号为固定频率的信号,操作系统每隔16.6ms发出一次定时VSYNC信号。

[0072] 可以在定时VSYNC信号每次到来时,触发本实施例的判断逻辑,以便判断操作系统的界面渲染操作是否执行完毕。进一步地,在本实施例中,界面渲染操作是否执行完毕的判断标准是:GPU将渲染结果提交到帧缓冲区中。

[0073] 步骤S530,在定时VSYNC信号来临时,如果所述操作系统的界面渲染操作已经完成,则利用所述定时VSYNC信号触发显示屏幕对渲染结果的显示。

[0074] 步骤S540,在定时VSYNC信号来临时,如果所述操作系统的界面渲染操作尚未完成,则忽略所述定时VSYNC信号并且向所述操作系统申请动态VSYNC信号。

[0075] 步骤S550,在所述操作系统的界面渲染操作完成之后,利用所述动态VSYNC信号触发显示屏幕对渲染结果的显示。

[0076] 在本实施例中,提供以下两种申请动态VSYNC信号触发显示屏幕对渲染结果的显示方式:

[0077] 方式一,向所述操作系统申请动态VSYNC信号,包括:向所述操作系统注册用于触发所述动态VSYNC信号的回调函数。利用所述动态VSYNC信号触发显示屏幕对渲染结果的显示,包括:在所述操作系统的界面渲染操作完成之后,通过调用所述回调函数,触发所述操作系统发出所述动态VSYNC信号,以便使所述动态VSYNC信号触发显示屏幕对渲染结果的显示。

[0078] 方式二,向所述操作系统申请动态VSYNC信号,包括:计算界面渲染操作的剩余时间,向操作系统发送动态VSYNC信号请求,并在该动态VSYNC信号请求中携带界面渲染操作的剩余时间。利用所述动态VSYNC信号触发显示屏幕对渲染结果的显示,包括:操作系统在所述界面渲染操作的剩余时间结束时(这时所述界面渲染操作执行完毕),发出动态VSYNC信号,这样就可以利用所述动态VSYNC信号触发显示屏幕对渲染结果的显示。

[0079] 本领域技术人员应当知道的是,以上两种方式仅用于说明本实施例,而不用于限定本实施例。

[0080] 在本实施例中,无论是在正常情况下,还是在异常情况下,利用定时VSYNC信号或

者动态VSYNC信号,都可以及时将显示屏幕带显示的渲染结果显示出来,有效缓解了卡顿现象,提升了画面流畅度。

[0081] 在正常情况下,操作系统可以在相邻两个定时VSYNC信号之间的16.6ms内完成界面渲染操作,这样无需向操作系统申请动态VSYNC信号,就可以在定时VSYNC信号来临时触发显示屏幕显示操作系统的渲染结果。因此,本实施例对于基于定时VSYNC信号的刷新机制没有任何影响。

[0082] 在异常情况下,由于操作系统的负载较大,无法在相邻两个定时VSYNC信号之间的16.6ms内完成界面渲染操作,这时可以暂时忽略已经来临的VSYNC信号,待本次的界面渲染操作完成之后,无需等待下一次定时VSYNC信号的到来,触发动态VSYNC信号,及时显示在上一次VSYNC信号来临时应当显示的渲染结果,提升画面流畅度,降低画面卡顿带来的用户体验效果差的问题。

[0083] 实施例二

[0084] 为了使本发明更加清楚,下面对本发明的技术方案进行进一步地描述。

[0085] 如图6所示,为根据本发明第二实施例的基于动态VSYNC信号的渲染方法的流程图。本实施例的执行主体为移动终端。

[0086] 步骤S610,操作系统执行界面渲染操作。

[0087] 步骤S620,在定时VSYNC信号来临时,判断操作系统的界面渲染操作是否完成;如果是,则执行步骤S630,如果不是,则执行步骤S640。

[0088] 步骤S630,在定时VSYNC信号来临时,如果所述操作系统的界面渲染操作已经完成,则利用所述定时VSYNC信号触发显示屏幕对渲染结果的显示。

[0089] 步骤S640,在定时VSYNC信号来临时,如果所述操作系统的界面渲染操作尚未完成,则忽略所述定时VSYNC信号并且向所述操作系统注册用于触发所述动态VSYNC信号的回调函数。

[0090] 在Android操作系统中VSYNC信号分为硬件生成的VSYNC信号和软件模拟的VSYNC信号。

[0091] 在本实施例中,定时VSYNC信号和动态VSYNC信号可以是硬件生成的VSYNC信号或者软件模拟的VSYNC信号。

[0092] 硬件生成的VSYNC信号是由HWC (Hardware Composer,硬件合成器)提供的,HWC封装了相关的HAL层,如果硬件厂商提供的HAL (Hardware Abstraction Layer,硬件抽象层)层能够定时产生VSYNC中断,进而发出硬件生成的VSYNC信号,否则HWC会通过VSYNC Thread (VSYNC线程)来模拟产生VSYNC中断,进而发出软件模拟的VSYNC信号。

[0093] 在本实施例中,向所述操作系统中的硬件合成器HWC注册用于触发所述动态VSYNC信号的回调函数。

[0094] 步骤S650,在所述操作系统的界面渲染操作完成之后,通过调用所述回调函数,触发所述操作系统发出所述动态VSYNC信号,以便使所述动态VSYNC信号触发显示屏幕对渲染结果的显示。

[0095] 本发明在不影响基于定时VSYNC信号的刷新机制的前提下,在操作系统可以及时完成界面渲染操作的情况下,利用定时VSYNC信号来触发显示屏幕对渲染结果的显示;在操作系统负载较大无法及时完成界面渲染操作的情况下,可以暂时忽略定时VSYNC信号,在操

作系统的界面渲染操作执行完毕之后,触发动态VSYNC信号,并利用该动态VSYNC信号及时将已经延误显示的渲染结果在显示屏幕中显示出来,提升了画面流畅度。

#### [0096] 实施例三

[0097] 下面给出一个较为具体的实施例来对本发明进行更进一步地的介绍。

[0098] 如图7所示,图7是根据本发明第三实施例的基于动态VSYNC信号的渲染方法的示意图。

[0099] 在图7中,Display表示显示屏幕。每个具有编号的矩形框表示1帧图像。

[0100] 下面针对图7对本发明的基于动态VSYNC信号的渲染方法进行说明:

[0101] 步骤1,Display显示第0帧图像,此时CPU和GPU渲染第1帧图像(编号为1的矩形框),而且在第1个定时VSYNC信号到来之前,CPU和GPU就已经将第1帧图像的渲染结果准备好,Display在第0帧显示完成后,通过正常的定时VSYNC信号就可以触发Display刷新,正常显示第1帧图像。

[0102] 步骤2,Display显示第1帧图像,此时CPU和GPU渲染第2帧图像,CPU和GPU渲染异常,例如CPU资源被占用,CPU和GPU还未渲染完成,第2个定时VSYNC就已经到来,这时忽略该第2个VSYNC信号,不对Display刷新,向操作系统注册一个DisplaySyncSource回调函数,在CPU和GPU渲染完成之后,通过回调DisplaySyncSource,触发动态VSYNC信号,对Display进行刷新,从而以最快的速度显示渲染数据。

[0103] 步骤3,Display显示第2帧图像,此时CPU和GPU渲染第3帧图像,而且在第3个定时VSYNC信号到来之前,CPU和GPU就已经将第3帧图像的渲染结果准备好,Display在第2帧显示完成后,通过第3个定时VSYNC信号就可以触发Display刷新,正常显示第3帧图像。

[0104] 在图7中可以看到,第1个定时VSYNC信号和第1个动态VSYNC信号之间的时间长度大于16.6ms;第1个动态VSYNC信号和第3个定时VSYNC信号之间的时间长度小于16.6ms。定时VSYNC信号和动态VSYNC信号之间互不影响,定时VSYNC信号以固定频率发出,动态VSYNC根据需求发出,动态VSYNC信号可以提前触发Display刷新,以便及时将已经延误显示的渲染结果(第2帧图像)在显示屏幕中显示出来。

[0105] 因此,本实施例不影响基于定时VSYNC信号的刷新机制,在此基础上,允许动态申请动态VSYNC信号;在操作系统中的CPU和GPU无法在相邻两个定时VSYNC信号之间的时间间隔内完成界面渲染操作时,申请动态VSYNC信号,暂时忽略已经来临的定时VSYNC信号(图7中虚线表示的定时VSYNC信号),在CPU和GPU的界面渲染操作执行完毕之后,触发动态VSYNC信号,并利用该动态VSYNC信号及时将已经延误显示的渲染结果在Display中显示出来,提升了画面流畅度。

#### [0106] 实施例四

[0107] 下面对本发明向操作系统申请动态VSYNC信号的实现方式进行进一步地描述。如图8所示,为根据本发明第四实施例的向操作系统申请动态VSYNC信号的示意图。

[0108] 在本实施例中,向操作系统申请动态VSYNC信号即是执行VSYNC事件的注册与分发的操作。

[0109] VSYNC事件的注册与分发主要包括以下三个过程的操作:

[0110] 第一个过程:初始化硬件操作

[0111] 操作系统启动后,启动SurfaceFlinger进程;在SurfaceFlinger进程启动之后,

SurfaceFlinger进程首先通过init(初始化函数)方法打开并初始化HWC硬件,使HWC硬件可以正常工作。其中,

[0112] HWC硬件模块,用于产出VSYNC信号。

[0113] SurfaceFlinger(图像绘制)进程,用于通过onVsyncReceived(事件到来回调函数)函数接收VSYNC信号,SurfaceFlinger进程,还用于渲染图像。

[0114] 第二个过程:分发VSYNC信号操作

[0115] 操作系统中有多个模块都需要用到VSYNC信号,当VSYNC信号到来时,SurfaceFlinger进程在VSYNC信号分发之前已经注册到DisSync的观察者,其中,DisSync:DisSyncThread(显示同步线程)起到了注册VSYNC观察者和分发事件的作用。

[0116] 第三个过程:处理VSYNC信号操作

[0117] DisSync将VSYNC信号分发到每一个已经注册过的DisplaySyncSource(同步显示源)对象中。DisplaySyncSource用来表示一个关注VSYNC信号的对象。

[0118] VSYNC事件响应函数为统一的接口:onVsyncEvent(VSYNC到来事件),即所有对VSYNC信号感兴趣的对象都需要实现该接口,具体的实现可以取决于其自身的需求。onVsyncEvent接口是在系统的EventThread中运行的,即对VSYNC事件的处理是在EventThread(事件线程)线程中运行的。EventThread线程通过Connection(连接器)对象,从SurfaceFlinger中取出渲染的图像,从而将渲染的图像显示到显示屏幕上。

[0119] 实施例五

[0120] 本发明实施例提供一种移动终端。

[0121] 如图9所示,为根据本发明第五实施例的移动终端的结构图。

[0122] 所述移动终端900包括处理器910和存储器920;

[0123] 所述处理器910用于执行所述存储器920中存储的基于动态VSYNC信号的渲染程序,以实现以下步骤:

[0124] 在定时VSYNC信号来临时,判断操作系统的界面渲染操作是否完成;

[0125] 如果所述界面渲染操作尚未完成,则忽略所述定时VSYNC信号并向所述操作系统申请动态VSYNC信号;

[0126] 在所述界面渲染操作完成之后,利用所述动态VSYNC信号触发显示屏幕对渲染结果的显示。

[0127] 可选地,所述在所述定时VSYNC信号来临时,所述处理器910还用于执行所述存储器920中存储的基于动态VSYNC信号的渲染程序,以实现以下步骤:在所述定时VSYNC信号来临时,如果所述操作系统的界面渲染操作已经完成,则利用所述定时VSYNC信号触发显示屏幕对渲染结果的显示。

[0128] 可选地,所述处理器910还用于执行所述存储器920中存储的基于动态VSYNC信号的渲染程序,以实现以下步骤:向所述操作系统注册用于触发所述动态VSYNC信号的回调函数;在所述操作系统的界面渲染操作完成之后,通过调用所述回调函数,触发所述操作系统发出所述动态VSYNC信号,以便使所述动态VSYNC信号触发显示屏幕对渲染结果的显示。

[0129] 可选地,所述处理器910还用于执行存储器920中存储的基于动态VSYNC信号的渲染程序,以实现以下步骤:向所述操作系统中的硬件合成器HWC注册用于触发所述动态VSYNC信号的回调函数。

[0130] 可选地,所述处理器910还用于执行存储器920中存储的基于动态VSYNC信号的渲染程序,以实现以下步骤:计算界面渲染操作的剩余时间,向操作系统发送动态VSYNC信号请求,并在该动态VSYNC信号请求中携带界面渲染操作的剩余时间;操作系统在所述界面渲染操作的剩余时间结束时(这时所述界面渲染操作执行完毕),发出动态VSYNC信号,这样就可以利用所述动态VSYNC信号触发显示屏幕对渲染结果的显示。

[0131] 本发明在不影响基于定时VSYNC信号的刷新机制的前提下,在操作系统负载较大无法及时完成界面渲染操作时,可以暂时忽略定时VSYNC信号,在操作系统的界面渲染操作执行完毕之后,触发动态VSYNC信号,及时将已经延误显示的渲染结果在显示屏幕中显示出来,提升了画面流畅度。

[0132] 实施例六

[0133] 本发明实施例还提供了一种存储介质。这里的存储介质存储有一个或者多个程序。其中,存储介质可以包括易失性存储器,例如随机存取存储器;存储器也可以包括非易失性存储器,例如只读存储器、快闪存储器、硬盘或固态硬盘;存储器还可以包括上述种类的存储器的组合。当存储介质中一个或者多个程序可被一个或者多个处理器执行,以实现以下步骤:

[0134] 在定时VSYNC信号来临时,判断操作系统的界面渲染操作是否完成;

[0135] 如果所述界面渲染操作尚未完成,则忽略所述定时VSYNC信号并向所述操作系统申请动态VSYNC信号;

[0136] 在所述界面渲染操作完成之后,利用所述动态VSYNC信号触发显示屏幕对渲染结果的显示。

[0137] 可选地,所述在所述定时VSYNC信号来临时,所述处理器还用于执行所述存储器中存储的基于动态VSYNC信号的渲染程序,以实现以下步骤:在所述定时VSYNC信号来临时,如果所述操作系统的界面渲染操作已经完成,则利用所述定时VSYNC信号触发显示屏幕对渲染结果的显示。

[0138] 可选地,所述处理器还用于执行所述存储器中存储的基于动态VSYNC信号的渲染程序,以实现以下步骤:向所述操作系统注册用于触发所述动态VSYNC信号的回调函数;在所述操作系统的界面渲染操作完成之后,通过调用所述回调函数,触发所述操作系统发出所述动态VSYNC信号,以便使所述动态VSYNC信号触发显示屏幕对渲染结果的显示。

[0139] 可选地,所述处理器还用于执行存储器中存储的基于动态VSYNC信号的渲染程序,以实现以下步骤:向所述操作系统中的硬件合成器HWC注册用于触发所述动态VSYNC信号的回调函数。

[0140] 可选地,所述处理器还用于执行存储器中存储的基于动态VSYNC信号的渲染程序,以实现以下步骤:计算界面渲染操作的剩余时间,向操作系统发送动态VSYNC信号请求,并在该动态VSYNC信号请求中携带界面渲染操作的剩余时间;操作系统在所述界面渲染操作的剩余时间结束时(这时所述界面渲染操作执行完毕),发出动态VSYNC信号,这样就可以利用所述动态VSYNC信号触发显示屏幕对渲染结果的显示。

[0141] 本发明在不影响基于定时VSYNC信号的刷新机制的前提下,在操作系统负载较大无法及时完成界面渲染操作时,可以暂时忽略定时VSYNC信号,在操作系统的界面渲染操作执行完毕之后,触发动态VSYNC信号,及时将已经延误显示的渲染结果在显示屏幕中显示出

来,提升了画面流畅度。

[0142] 需要说明的是,在本文中,术语“包括”、“包含”或者其任何其他变体意在涵盖非排他性的包含,从而使得包括一系列要素的过程、方法、物品或者装置不仅包括那些要素,而且还包括没有明确列出的其他要素,或者是还包括为这种过程、方法、物品或者装置所固有的要素。在没有更多限制的情况下,由语句“包括一个……”限定的要素,并不排除在包括该要素的过程、方法、物品或者装置中还存在另外的相同要素。

[0143] 上述本发明实施例序号仅仅为了描述,不代表实施例的优劣。

[0144] 通过以上的实施方式的描述,本领域的技术人员可以清楚地了解到上述实施例方法可借助软件加必需的通用硬件平台的方式来实现,当然也可以通过硬件,但很多情况下前者是更佳的实施方式。基于这样的理解,本发明的技术方案本质上或者说对现有技术做出贡献的部分可以以软件产品的形式体现出来,该计算机软件产品存储在一个存储介质(如ROM/RAM、磁碟、光盘)中,包括若干指令用以使得一台终端设备(可以是手机,计算机,服务器,空调器,或者网络设备等)执行本发明各个实施例的方法。

[0145] 以上仅为本发明的优选实施例,并非因此限制本发明的专利范围,凡是利用本发明说明书及附图内容所作的等效结构或等效流程变换,或直接或间接运用在其他相关的技术领域,均同理包括在本发明的专利保护范围内。

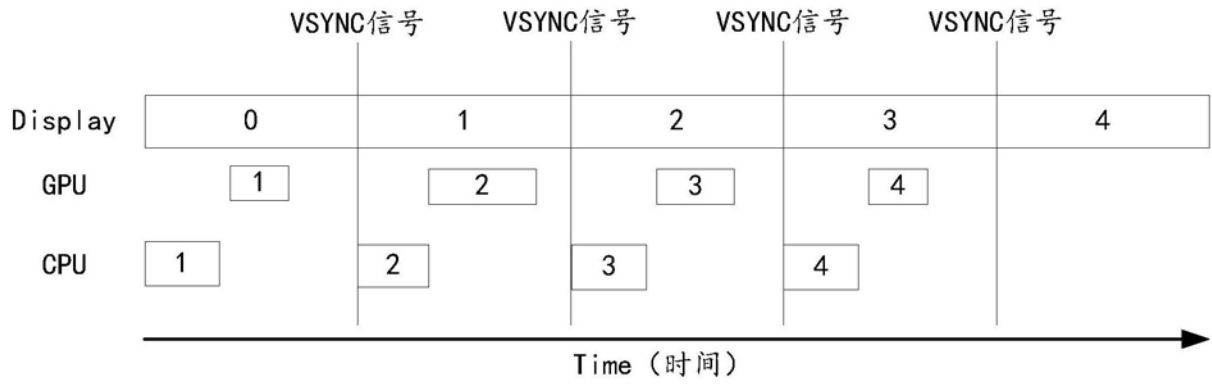


图1

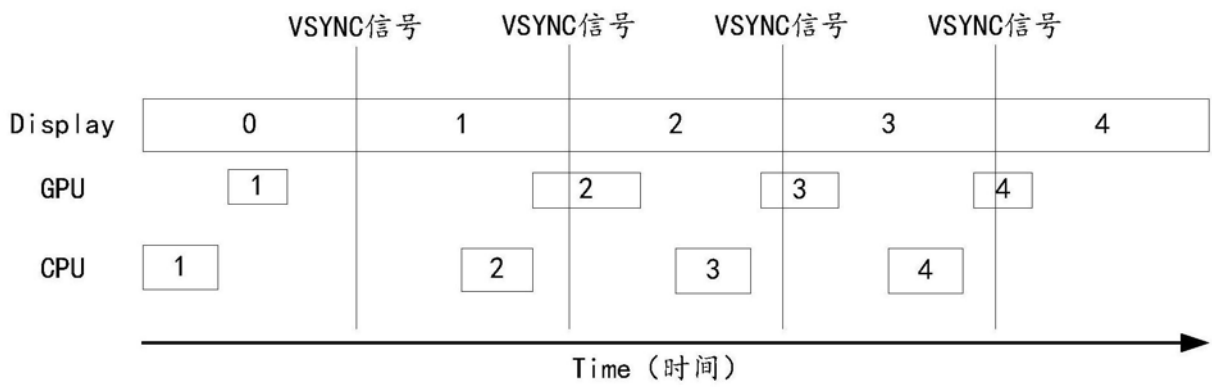


图2



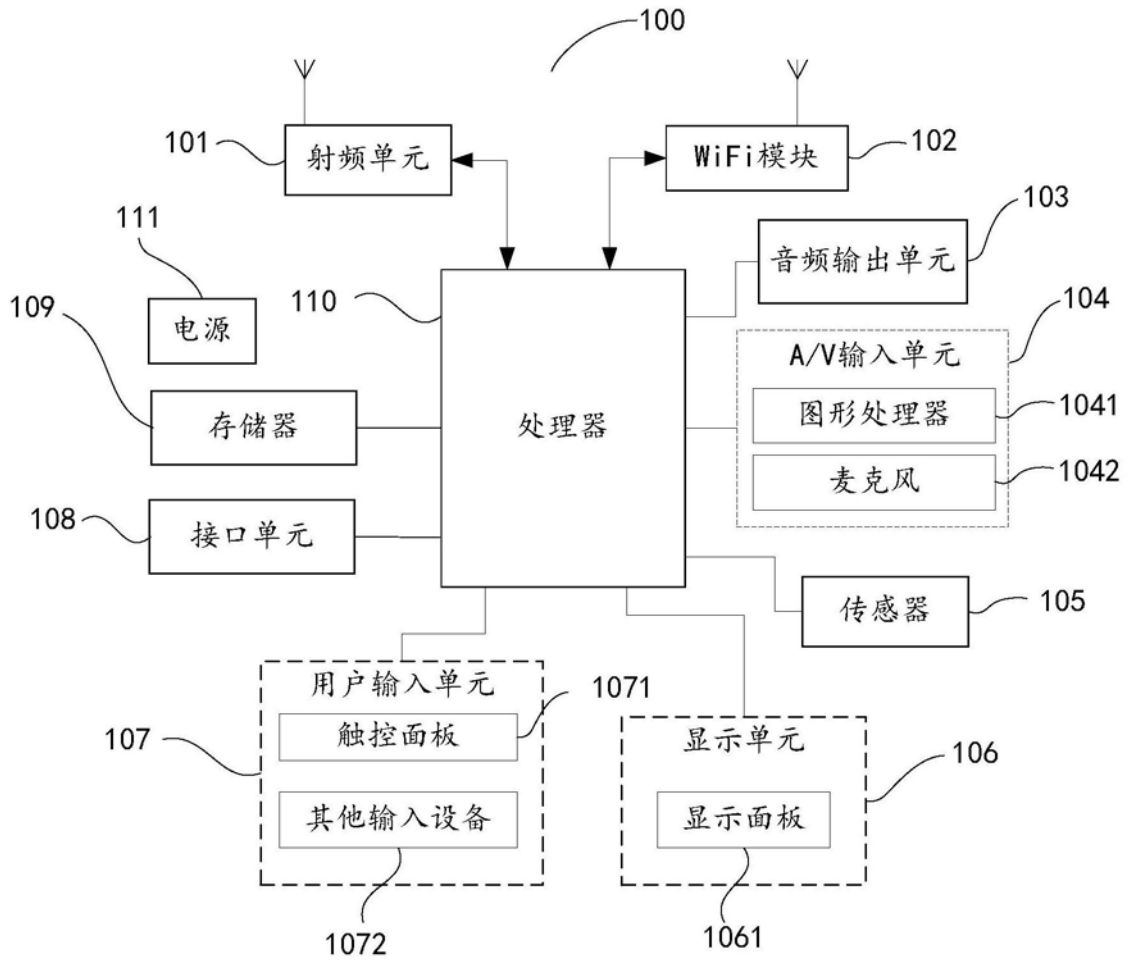


图3

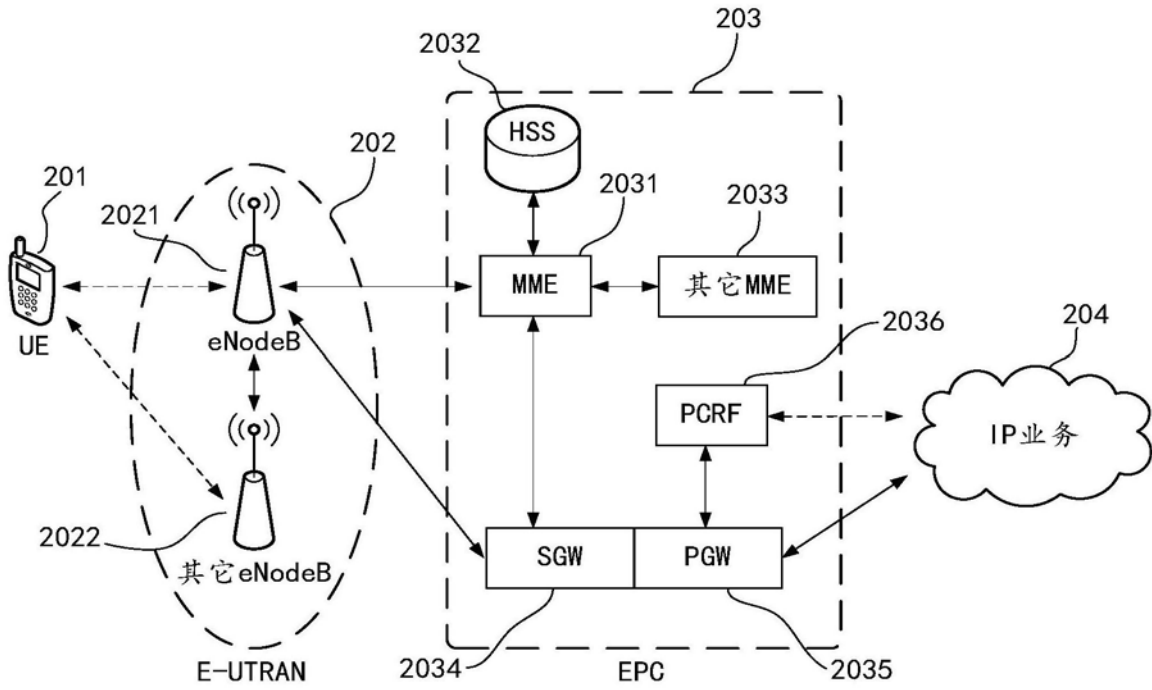


图4

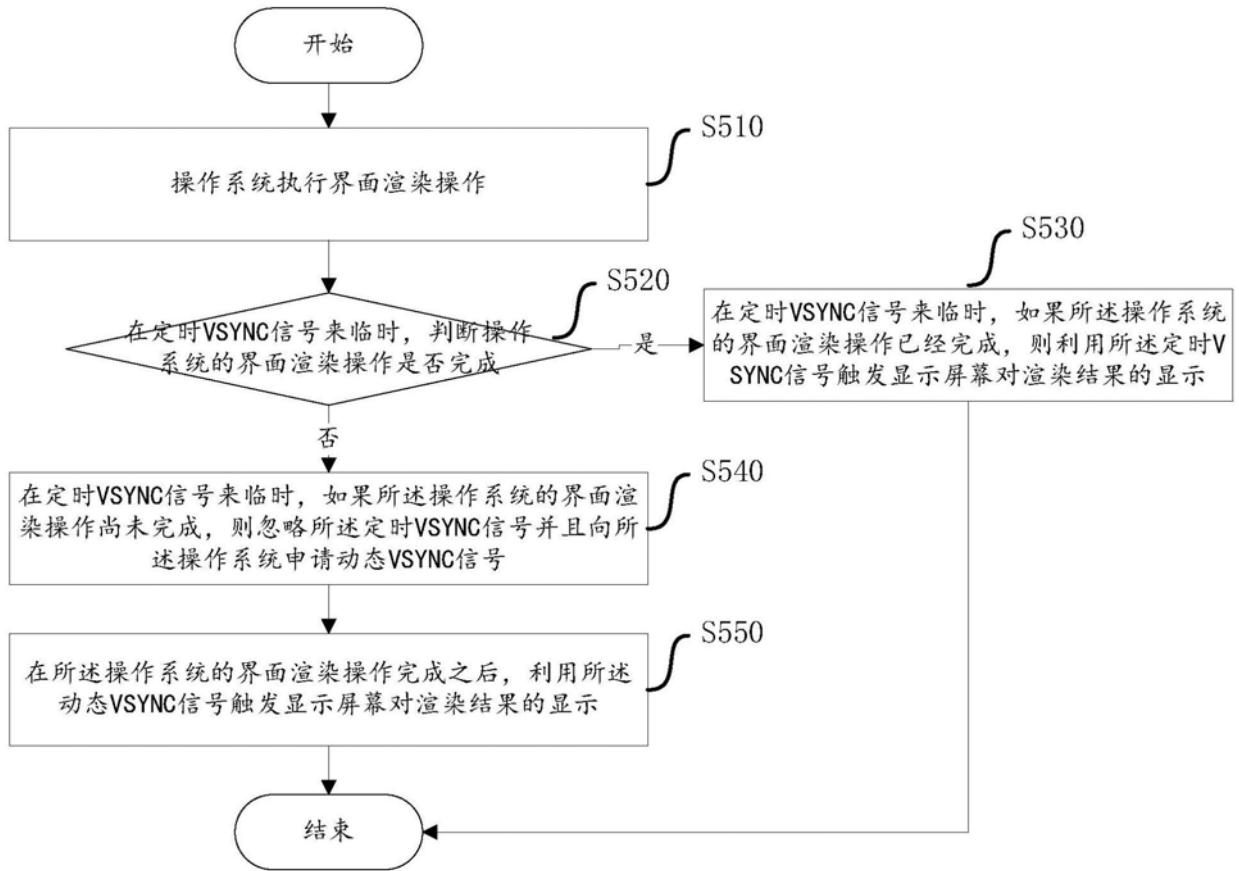


图5

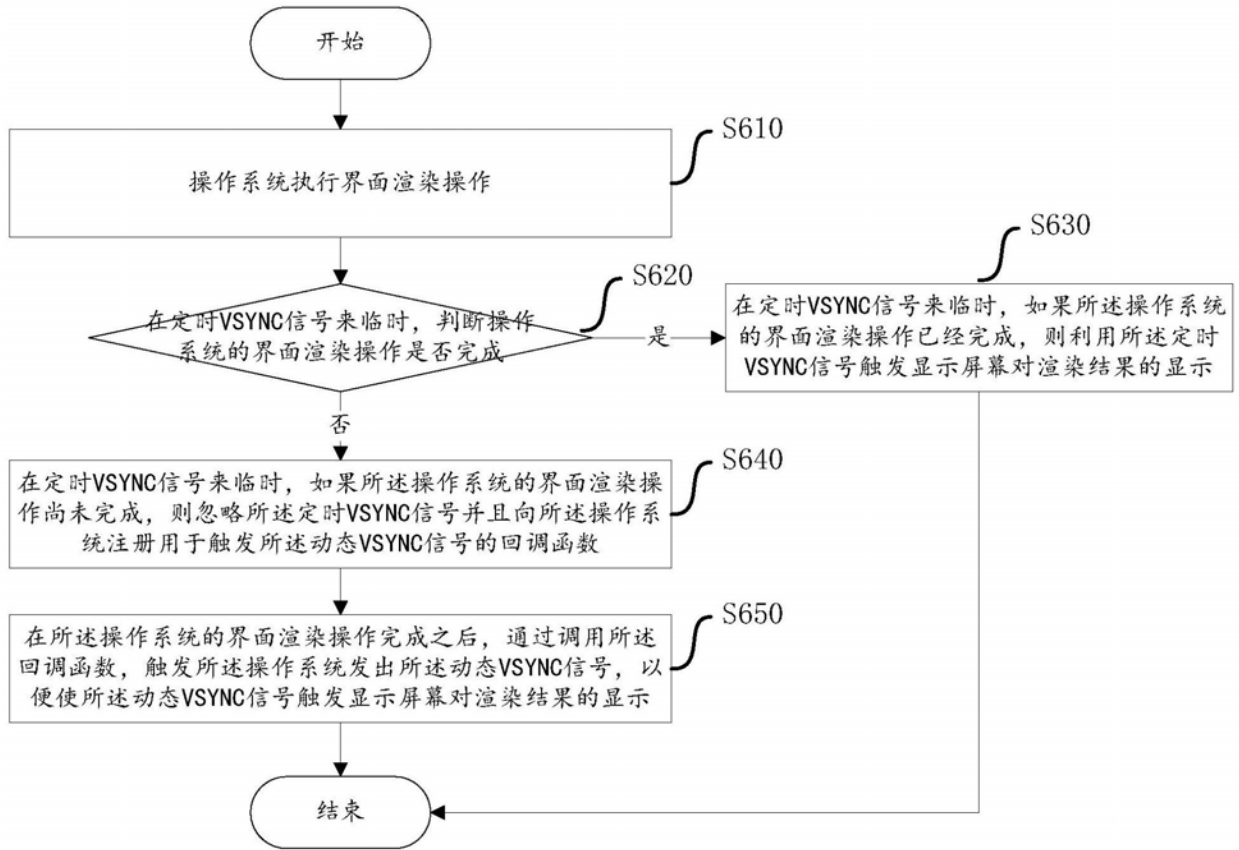


图6

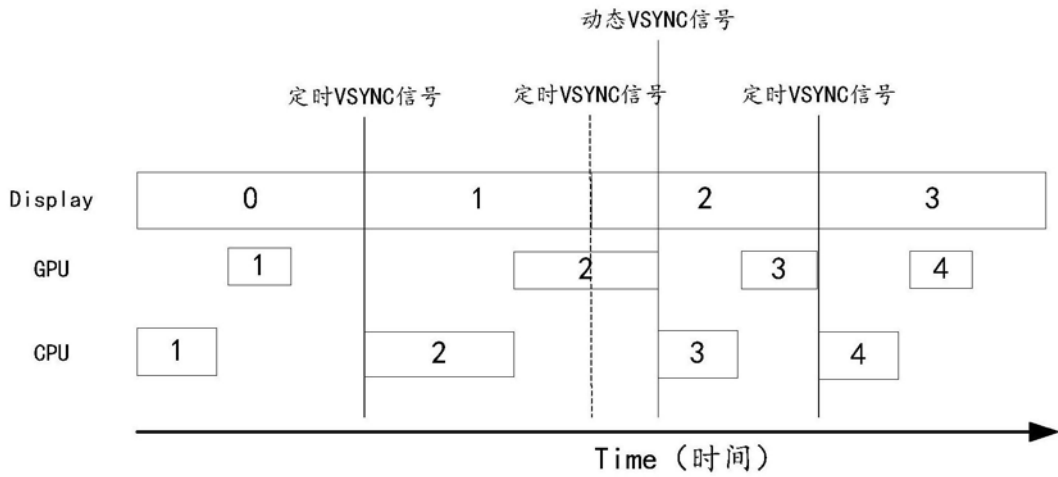


图7

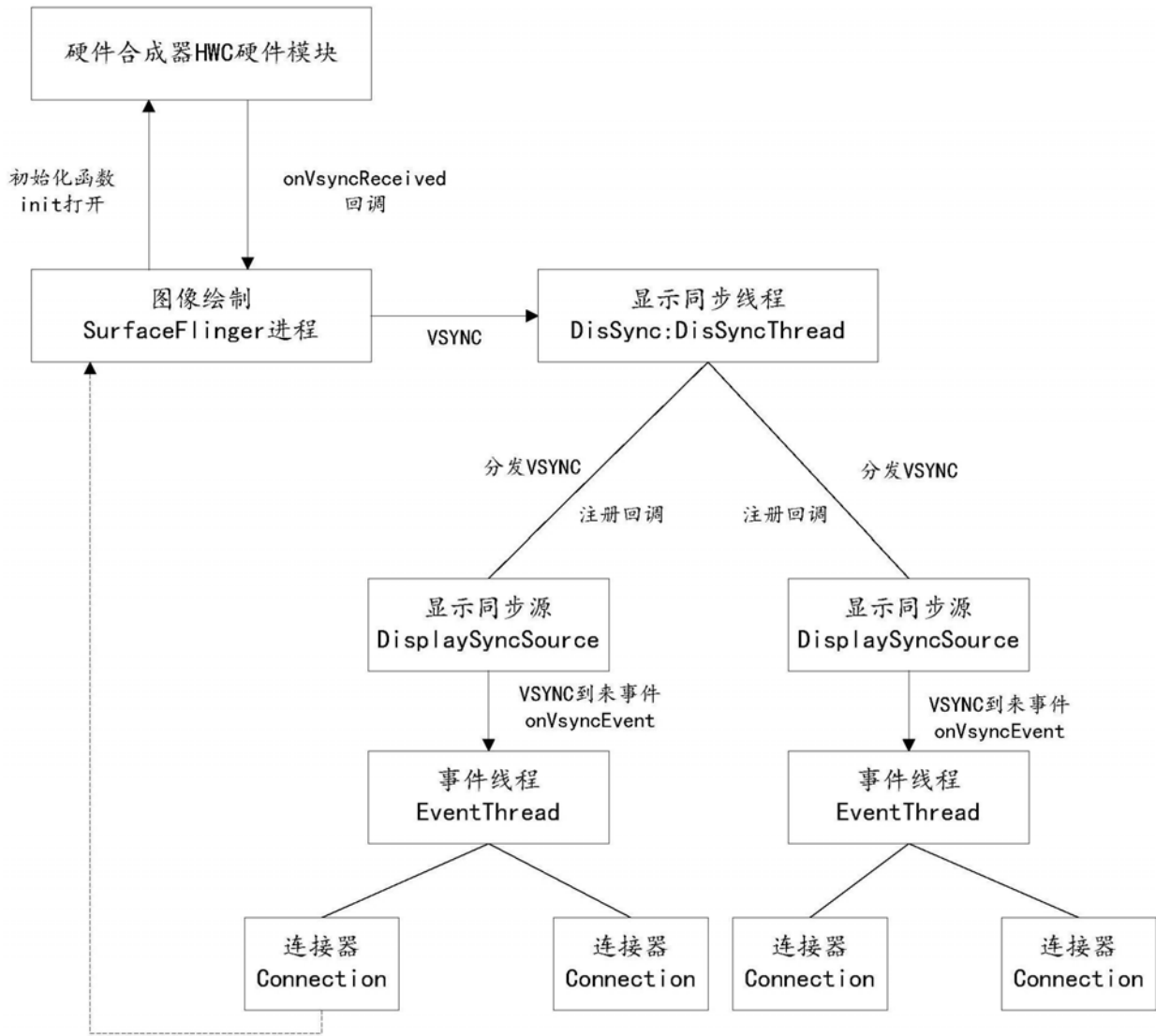


图8

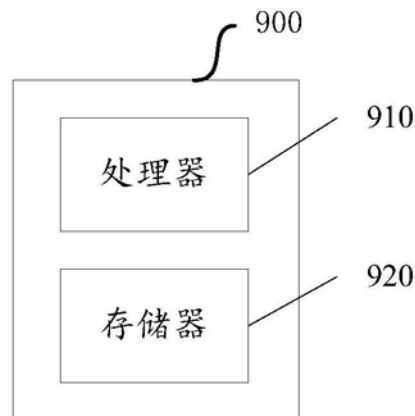


图9