



(12) 发明专利

(10) 授权公告号 CN 108829475 B

(45) 授权公告日 2021.08.17

(21) 申请号 201810533938.8

(22) 申请日 2018.05.29

(65) 同一申请的已公布的文献号
申请公布号 CN 108829475 A

(43) 申请公布日 2018.11.16

(73) 专利权人 北京小米移动软件有限公司
地址 100085 北京市海淀区清河中街68号
华润五彩城购物中心二期9层01房间

(72) 发明人 顾云建

(74) 专利代理机构 北京英创嘉友知识产权代理
事务所(普通合伙) 11447

代理人 曹寒梅

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

(56) 对比文件

CN 107273130 A, 2017.10.20

CN 104461255 A, 2015.03.25

CN 104714675 A, 2015.06.17

CN 107220019 A, 2017.09.29

US 2010177176 A1, 2010.07.15

CN 105589883 A, 2016.05.18

US 2016247484 A1, 2016.08.25

审查员 吴阳

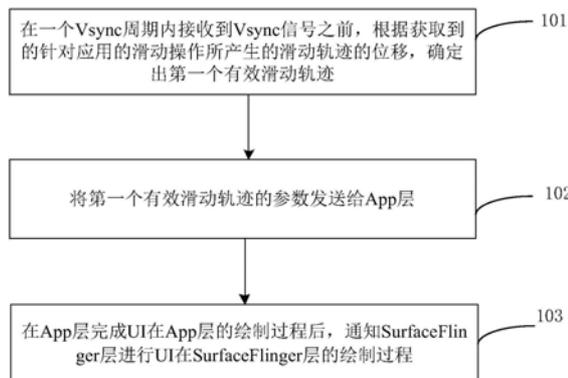
权利要求书3页 说明书8页 附图9页

(54) 发明名称

UI绘制方法、装置及存储介质

(57) 摘要

本公开是关于一种UI绘制方法、装置及存储介质,其中,UI绘制方法包括:在一个Vsync周期内接收到Vsync信号之前,根据获取到的基于应用的滑动操作所产生的滑动轨迹的位移确定出第一个有效滑动轨迹;将所述第一个有效滑动轨迹的参数发送给App层,以使所述App层基于所述参数进行UI在所述App层的绘制过程;在所述App层完成所述UI在所述App层的绘制过程后,通知SurfaceFlinger层进行所述UI在所述SurfaceFlinger层的绘制过程,从而提高了终端的跟手性。



1. 一种用户界面UI绘制方法,其特征在于,该方法应用于终端,包括:

在一个Vsync周期内接收到Vsync信号之前,根据获取到的针对应用的滑动操作所产生的滑动轨迹的位移确定出第一个有效滑动轨迹;

将所述第一个有效滑动轨迹的参数发送给App层,以使所述App层基于所述参数进行用户界面UI在所述App层的绘制过程;

在所述App层完成所述UI在所述App层的绘制过程后,通知SurfaceFlinger层进行所述UI在所述SurfaceFlinger层的绘制过程;

所述根据获取到的针对应用的滑动操作所产生的滑动轨迹的位移确定出第一个有效滑动轨迹,包括:

确定所述滑动轨迹的位移是否超过第一阈值;

在所述滑动轨迹的位移超过所述第一阈值时,确定该滑动轨迹为有效滑动轨迹;

在所述滑动轨迹的位移未超过所述第一阈值时,对所述滑动轨迹的位移进行补偿;

确定补偿后的位移是否超过第二阈值;

在补偿后的位移超过所述第二阈值时,确定所述滑动轨迹为有效滑动轨迹;

在补偿后的位移未超过所述第二阈值时,确定所述滑动轨迹为无效滑动轨迹;

其中,通过下述计算式对所述滑动轨迹的位移进行补偿,该计算式包括:

$$\text{syntheticSlop} = \frac{\text{mTouchSlop} * \text{RESAMPLE_MAX_PREDICTION}}{2 * \text{RESAMPLE_MAX_PREDICTION} - \text{RESAMPLE_LATENCY}} + 1$$

其中,syntheticSlop为补偿后的距离,mTouchSlop是终端系统内配置TouchSlop值,RESAMPLE_MAX_PREDICTION为最大重采样预测时间,RESAMPLE_LATENCY为重采样延迟。

2. 根据权利要求1所述的方法,其特征在于,所述方法还包括:

在所述App层创建所述应用的窗口时,保存通信接口,所述通信接口作为通知所述SurfaceFlinger层进行所述UI在SurfaceFlinger层的绘制过程的通信接口。

3. 根据权利要求1所述的方法,其特征在于,所述第一阈值和/或所述第二阈值为所述终端中配置的TouchSlop值。

4. 根据权利要求1至3任一项所述的方法,其特征在于,所述方法还包括:

在所述一个Vsync周期内确定出第一个有效滑动轨迹之后,获取其他针对所述应用的滑动操作所产生的其他有效滑动轨迹;

待接收到Vsync信号后,将所述其他有效滑动轨迹对应的参数发送给App层,以使所述App层基于所述其他有效滑动轨迹对应的参数进行UI在所述App层的绘制过程。

5. 一种用户界面UI绘制装置,其特征在于,该装置应用于终端,包括:

确定模块,用于在一个Vsync周期内接收到Vsync信号之前,根据获取到的针对应用的滑动操作所产生的滑动轨迹的位移确定出第一个有效滑动轨迹;

第一发送模块,用于将所述第一个有效滑动轨迹的参数发送给App层,以使所述App层基于所述参数进行用户界面UI在所述App层的绘制过程;

通知模块,用于在所述App层完成所述UI在所述App层的绘制过程后,通知SurfaceFlinger层进行所述UI在所述SurfaceFlinger层的绘制过程;

所述确定模块,包括:

第一确定子模块,用于确定获取到的滑动轨迹的位移是否超过第一阈值;

第二确定子模块,用于在所述滑动轨迹的位移超过所述第一阈值时,确定该滑动轨迹为有效滑动轨迹;

第三确定子模块,用于在所述滑动轨迹的位移未超过所述第一阈值时,对所述滑动轨迹的位移进行补偿;

第四确定子模块,用于确定补偿后的位移是否超过第二阈值;

第五确定子模块,用于在补偿后的位移超过所述第二阈值时,确定所述滑动轨迹为有效滑动轨迹;

第六确定子模块,用于在补偿后的位移未超过所述第二阈值时,确定所述滑动轨迹为无效滑动轨迹;

其中,所述第三确定子模块通过下述计算式对所述滑动轨迹的位移进行补偿,该计算式包括:

$$\text{syntheticSlop} = \frac{\text{mTouchSlop} * \text{RESAMPLE_MAX_PREDICTION}}{2 * \text{RESAMPLE_MAX_PREDICTION} - \text{RESAMPLE_LATENCY}} + 1$$

其中,syntheticSlop为补偿后的距离,mTouchSlop是终端系统内配置TouchSlop值,RESAMPLE_MAX_PREDICTION为最大重采样预测时间,RESAMPLE_LATENCY为重采样延迟。

6. 根据权利要求5所述的装置,其特征在于,所述装置还包括:

保存模块,用于在所述App层创建所述应用的窗口时,保存通信接口,所述通信接口作为通知所述SurfaceFlinger层进行所述UI在SurfaceFlinger层的绘制过程的通信接口。

7. 根据权利要求5所述的装置,其特征在于,所述第一阈值和/或所述第二阈值为所述终端中配置的TouchSlop值。

8. 根据权利要求5至7任一项所述的装置,其特征在于,所述装置还包括:

获取模块,用于在所述一个Vsync周期内确定出第一个有效滑动轨迹之后,获取其他针对所述应用的滑动操作所产生的其他有效滑动轨迹;

第二发送模块,用于待接收到Vsync信号后,将所述其他有效滑动轨迹对应的参数发送给App层,以使所述App层基于所述其他有效滑动轨迹对应的参数进行UI在所述App层的绘制过程。

9. 一种用户界面UI绘制装置,其特征在于,包括:

处理器;

用于存储处理器可执行指令的存储器;

其中,所述处理器被配置为:

在一个Vsync周期内接收到Vsync信号之前,根据获取到的针对应用的滑动操作所产生的滑动轨迹的位移确定出第一个有效滑动轨迹;

将所述第一个有效滑动轨迹的参数发送给App层,以使所述App层基于所述参数进行用户界面UI在所述App层的绘制过程;

在所述App层完成所述UI在所述App层的绘制过程后,通知SurfaceFlinger层进行所述UI在所述SurfaceFlinger层的绘制过程;

所述根据获取到的针对应用的滑动操作所产生的滑动轨迹的位移确定出第一个有效滑动轨迹,包括:

确定所述滑动轨迹的位移是否超过第一阈值;

在所述滑动轨迹的位移超过所述第一阈值时,确定该滑动轨迹为有效滑动轨迹;
在所述滑动轨迹的位移未超过所述第一阈值时,对所述滑动轨迹的位移进行补偿;
确定补偿后的位移是否超过第二阈值;
在补偿后的位移超过所述第二阈值时,确定所述滑动轨迹为有效滑动轨迹;
在补偿后的位移未超过所述第二阈值时,确定所述滑动轨迹为无效滑动轨迹;
其中,通过下述计算式对所述滑动轨迹的位移进行补偿,该计算式包括:

$$\text{syntheticSlop} = \frac{\text{mTouchSlop} * \text{RESAMPLE_MAX_PREDICTION}}{2 * \text{RESAMPLE_MAX_PREDICTION} - \text{RESAMPLE_LATENCY}} + 1$$

其中,syntheticSlop为补偿后的距离,mTouchSlop是终端系统内配置TouchSlop值,RESAMPLE_MAX_PREDICTION为最大重采样预测时间,RESAMPLE_LATENCY为重采样延迟。

10.一种非临时性计算机可读存储介质,当所述存储介质中的指令由处理器执行时,使得处理器能够执行根据权利要求1至4任一项所述的方法。

UI绘制方法、装置及存储介质

技术领域

[0001] 本公开涉及计算机应用技术领域,尤其涉及一种UI(User Interface,用户界面)绘制方法、装置及存储介质。

背景技术

[0002] 当前,移动终端的性能在不断的提升,用户对移动终端的流畅性要求也越来越高,跟手性是衡量移动终端流畅性的一个重要指标,跟手性是通过用户手指在移动终端触摸屏上按下进行滑动到移动终端的画面出现变化的时间来衡量的。

[0003] 在Android(安卓)系统中,Android系统的绘制过程分为App(应用)层和SurfaceFlinger层,SurfaceFlinger层负责管理应用端的surface(界面),将所有应用端的surface复合,它是介于图形库和应用之间的一层。每个应用在其surface完成各种图形操作后,请求SurfaceFlinger显示到屏幕,surfaceflinger就会将所有的surface叠加起来,并且反映到framebuffer(帧缓冲)。App层和SurfaceFlinger层的绘制是通过Vsync周期来控制的,Vsync周期决定了用户在屏幕上进行点击或滑动操作后,屏幕所呈现出的画面的变化速度,App和SurfaceFlinger都必须等待Vsync信号后才执行绘制操作。

发明内容

[0004] 为克服相关技术中存在的问题,本公开提供一种UI绘制方法、装置及存储介质。

[0005] 根据本公开的一个方面,提供了一种用户界面UI绘制方法,该方法应用于终端,包括:在一个Vsync周期内接收到Vsync信号之前,根据获取到的针对应用的滑动操作所产生的滑动轨迹的位移确定出第一个有效滑动轨迹;将所述第一个有效滑动轨迹的参数发送给App层,以使所述App层基于所述参数进行UI在所述App层的绘制过程;在所述App层完成所述UI在所述App层的绘制过程后,通知SurfaceFlinger层进行所述UI在所述SurfaceFlinger层的绘制过程。

[0006] 可选地,所述方法还包括:在所述App层创建所述应用的窗口时,保存通信接口,所述通信接口作为通知所述SurfaceFlinger层进行所述UI在SurfaceFlinger层的绘制过程的通信接口。

[0007] 可选地,所述根据获取到的针对应用的滑动操作所产生的滑动轨迹的位移确定出第一个有效滑动轨迹,包括:确定所述滑动轨迹的位移是否超过第一阈值;在所述滑动轨迹的位移超过所述第一阈值时,确定该滑动轨迹为有效滑动轨迹;在所述滑动轨迹的位移未超过所述第一阈值时,对所述滑动轨迹的位移进行补偿;确定补偿后的位移是否超过第二阈值;在补偿后的位移超过所述第二阈值时,确定所述滑动轨迹为有效滑动轨迹;在补偿后的位移未超过所述第二阈值时,确定所述滑动轨迹为无效滑动轨迹。

[0008] 可选地,所述第一阈值和/或第二阈值为所述终端中配置的TouchSlop值。

[0009] 可选地,所述方法还包括:在所述一个Vsync周期内确定出第一个有效滑动轨迹之后,获取其他针对所述应用的滑动操作所产生的其他有效滑动轨迹;待接收到Vsync信号

后,将所述其他有效滑动轨迹对应的参数发送给App层,以使所述App层基于所述其他有效滑动轨迹对应的参数进行UI在所述App层的绘制过程。

[0010] 根据本公开的第二个方面,提供了一种用户界面UI绘制装置,该装置应用于终端,包括:确定模块,用于在一个Vsync周期内接收到Vsync信号之前,根据获取到的针对应用的滑动操作所产生的滑动轨迹的位移确定出第一个有效滑动轨迹;第一发送模块,用于将所述第一个有效滑动轨迹的参数发送给App层,以使所述App层基于所述参数进行UI在所述App层的绘制过程;通知模块,用于在所述App层完成所述UI在所述App层的绘制过程后,通知SurfaceFlinger层进行所述UI在所述SurfaceFlinger层的绘制过程。

[0011] 可选地,所述装置还包括:保存模块,用于在所述App层创建所述应用的窗口时,保存通信接口,所述通信接口作为通知所述SurfaceFlinger层进行所述UI在SurfaceFlinger层的绘制过程的通信接口。

[0012] 可选地,所述确定模块,包括:第一确定子模块,用于确定获取到的滑动轨迹的位移是否超过第一阈值;第二确定子模块,用于在所述滑动轨迹的位移超过所述第一阈值时,确定该滑动轨迹为有效滑动轨迹;第三确定子模块,用于在所述滑动轨迹的位移未超过所述第一阈值时,对所述滑动轨迹的位移进行补偿;第四确定子模块,用于确定补偿后的位移是否超过第二阈值;第五确定子模块,用于在补偿后的位移超过所述第二阈值时,确定所述滑动轨迹为有效滑动轨迹;第六确定子模块,用于在补偿后的位移未超过所述第二阈值时,确定所述滑动轨迹为无效滑动轨迹。

[0013] 可选地,所述第一阈值和/或第二阈值为所述终端中配置的TouchSlop值。

[0014] 可选地,所述装置还包括:获取模块,用于在所述一个Vsync周期内确定出第一个有效滑动轨迹之后,获取其他针对所述应用的滑动操作所产生的其他有效滑动轨迹;第二发送模块,用于待接收到Vsync信号后,将所述其他有效滑动轨迹对应的参数发送给App层,以使所述App层基于所述其他有效滑动轨迹对应的参数进行UI在所述App层的绘制过程。

[0015] 根据本公开的第三个方面,提供了一种用户界面UI绘制装置,包括:处理器;用于存储处理器可执行指令的存储器;其中,所述处理器被配置为:在一个Vsync周期内接收到Vsync信号之前,根据获取到的针对应用的滑动操作所产生的滑动轨迹的位移确定出第一个有效滑动轨迹;将所述第一个有效滑动轨迹的参数发送给App层,以使所述App层基于所述参数进行UI在所述App层的绘制过程;在所述App层完成所述UI在所述App层的绘制过程后,通知SurfaceFlinger层进行所述UI在所述SurfaceFlinger层的绘制过程。

[0016] 根据本公开的第四个方面,提供了一种非临时性计算机可读存储介质,当所述存储介质中的指令由处理器执行时,使得处理器能够执行根据本公开第一个方面所述的任一项所述的方法。

[0017] 本公开的实施例提供的技术方案可以包括以下有益效果:

[0018] 本公开的实施例提供的终端界面绘制方法,在接收到Vsync之前,确定出针对应用的滑动操作所产生的第一个有效滑动轨迹后,即将该滑动轨迹的参数发送给App层,使得App层提前执行UI的绘制过程,在App层绘制结束后,通知SurfaceFlinger层开始绘制,又使得SurfaceFlinger层也提前了UI的绘制过程,缩短了用户开始滑动操作到终端画面出现变化的时间,提高了终端的跟手性。

[0019] 应当理解的是,以上的一般描述和后文的细节描述仅是示例性和解释性的,并不

能限制本公开。

附图说明

[0020] 此处的附图被并入说明书中并构成本说明书的一部分，示出了符合本公开的实施例，并与说明书一起用于解释本公开的原理。

[0021] 图1是根据一示例性实施例示出的一种UI绘制方法的流程图。

[0022] 图2是根据一示例性实施例示出的一种UI绘制方法的流程图。

[0023] 图3是根据一示例性实施例示出的一种UI绘制方法的流程图。

[0024] 图4是根据一示例性实施例示出的一种UI绘制方法的流程图。

[0025] 图5是App层以及SurfaceFlinger层均等待接收到Vsync信号后再执行UI绘制过程的示意图。

[0026] 图6是App层以及SurfaceFlinger层不等待接收到Vsync信号执行UI绘制过程的示意图。

[0027] 图7是根据一示例性实施例示出的一种UI绘制装置的框图。

[0028] 图8是根据一示例性实施例示出的一种UI绘制装置的框图。

[0029] 图9是根据一示例性实施例示出的一种UI绘制装置的框图。

[0030] 图10是根据一示例性实施例示出的一种UI绘制装置的框图。

[0031] 图11是根据一示例性实施例示出的一种UI绘制装置的框图。

具体实施方式

[0032] 这里将详细地对示例性实施例进行说明，其示例表示在附图中。下面的描述涉及附图时，除非另有表示，不同附图中的相同数字表示相同或相似的要素。以下示例性实施例中所描述的实施方式并不代表与本公开相一致的所有实施方式。相反，它们仅是与如所附权利要求书中所详述的、本公开的一些方面相一致的装置和方法的例子。

[0033] 图1是根据一示例性实施例示出的一种UI绘制方法的流程图，如图1所示，该方法包括：

[0034] 步骤101：在一个Vsync周期内接收到Vsync信号之前，根据获取到的针对应用的滑动操作所产生的滑动轨迹的位移，确定出第一个有效滑动轨迹；

[0035] 其中，针对应用的滑动操作可以是用户通过终端屏幕对应用进行的任意滑动操作，例如在应用的用户界面UI中进行的用于翻页、快进、快退等滑动操作。滑动操作所产生的滑动轨迹的位移可为滑动操作产生的滑动轨迹的起点至终点之间的距离。

[0036] 步骤102：将第一个有效滑动轨迹的参数发送给App层，以使App层基于该参数进行UI在App层的绘制过程；

[0037] 其中，滑动轨迹的参数可包括：滑动方向以及滑动轨迹的位移，或者该参数可包括滑动轨迹起点的坐标以及滑动轨迹终点的坐标。

[0038] 步骤103：在App层完成UI在App层的绘制过程后，通知SurfaceFlinger层进行UI在SurfaceFlinger层的绘制过程。

[0039] 其中，UI在App层的绘制过程可包括对应用在UI中显示的部分的绘制，UI在SurfaceFlinger层的绘制过程可包括SurfaceFlinger层将该应用在UI中显示的部分以及

其他应用在UI中显示的部分进行叠加。

[0040] 在步骤103中,例如可通过一个简单的触发信号来通知SurfaceFlinger层进行UI在该层的绘制过程,SurfaceFlinger层在接收到该触发信号后即开始UI的在该层的绘制过程,无需等待Vsync信号。

[0041] 本实施例的终端界面绘制方法,在接收到Vsync之前,确定出针对应用的滑动操作所产生的第一个有效滑动轨迹后,即将该滑动轨迹的参数发送给App层,使得App层提前执行UI的绘制过程,在App层绘制结束后,通知SurfaceFlinger层开始绘制,又使得SurfaceFlinger层也提前了UI的绘制过程,缩短了用户开始滑动操作到终端画面出现变化的时间,提高了终端的跟手性。图2是根据一示例性实施例示出的一种UI绘制方法的流程图,如图2所示,该方法在图1所示的方法的基础上,还可以包括步骤100,在该步骤100中,在创建应用的窗口时,保存通信接口,例如保存一个通信接口,所述通信接口作为通知所述SurfaceFlinger层进行所述UI在SurfaceFlinger层的绘制过程的通信接口。通过该接口可使得App层得以直接与SurfaceFlinger层进行通信,例如,App层通过该通信接口向SurfaceFlinger层发送一个触发信号,触发SurfaceFlinger层开始绘制,故App层绘制结束后,即可通知SurfaceFlinger层进行绘制,从而不仅提前执行了App层的绘制,同时也提前执行了SurfaceFlinger层的绘制。

[0042] 图3是根据一示例性实施例示出的一种UI绘制方法的流程图,如图3所示,在该方法中,图1中所示的步骤101可由步骤1011以及步骤1016执行,其中,在步骤1011中,确定滑动轨迹的位移是否超过第一阈值;在步骤1012中,在滑动轨迹的位移超过第一阈值时,确定滑动轨迹为有效滑动轨迹;在步骤1013中,在滑动轨迹的位移未超过第一阈值时,对该位移进行补偿;在步骤1014中,确定补偿后的位移是否超过第二阈值;在步骤1015中,在补偿后的位移超过第二阈值时,确定滑动轨迹为有效滑动轨迹;在步骤1016中,在补偿后的位移未超过第二阈值时,确定滑动轨迹为无效滑动轨迹。通过对滑动轨迹位移的补偿,能够避免由于采样的延迟导致获取到的滑动轨迹位移存在误差的问题。其中,第二阈值可与第一阈值相同,或者,第二阈值也可以与第一阈值不同,例如,第二阈值可以略小于第一阈值,或略大于第一阈值。

[0043] 在一种可能的实现方式中,该步骤1011至1016是在终端的触摸屏幕获取到滑动操作即需要执行的步骤,图3中所示出的仅是在获取到的第一个有效滑动轨迹之前执行该步骤1011-1016的一种情况,在获取到第一个有效滑动轨迹后获取到的其他滑动轨迹也可执行步骤1011至1016。基于步骤1011至1016,可以确保滑动轨迹的位移满足一定长度时即通知应用进行UI的绘制,当在Vsync信号还未到达但当前获取的滑动轨迹的位移已达到阈值或在进行补偿后达到阈值的情况下,使得UI的绘制过程提前。

[0044] 在一种可能的实现方式中,也可不进行步骤1013-1016,即可在步骤1012中,在滑动轨迹的位移未超过阈值时,确定滑动轨迹为无效滑动轨迹。

[0045] 在一种可实现方式中,上述第一阈值和/或第二阈值为终端中配置的TouchSlop的值,终端的操作系统(以下简称系统)利用该值区分滑动操作和点击操作,如果滑动轨迹的位移没有超过TouchSlop的值,系统不会认为用户在终端屏幕上进行了滑动操作,当用户在屏幕上的滑动操作产生的滑动轨迹的位移超过TouchSlop的值时,系统会认为用户执行了滑动操作并去响应该滑动操作,不同的设备中配置的TouchSlop的值可能是不同的。本实施

例的UI绘制方法无需改变终端系统中配置的TouchSlop的值,与系统内原有的滑动操作识别机制并不冲突。

[0046] 图4是根据一示例性实施例示出的一种UI绘制方法的流程图,如图4所示,该方法在图1所示的方法的基础上还可包括步骤104以及105,在步骤104中,在一个Vsync周期内确定出第一个有效滑动轨迹之后,获取其他针对应用的其他有效滑动轨迹;在步骤105中,待接收到Vsync信号后,将其他有效滑动轨迹对应的参数发送给App层,以使App层基于其他有效滑动轨迹对应的参数进行UI在App层的绘制过程。SurfaceFlinger层在接收到Vsync信号后,执行UI在SurfaceFlinger层的绘制过程。即,在本实施例的UI绘制方法中,对在当前Vsync获取的第一个有效滑动轨迹提前开启App层以及SurfaceFlinger层的UI绘制过程,在该周期内获取的其他有效滑动轨迹,则仍需等待接收到Vsync信号后,再开启App层以及SurfaceFlinger层的UI绘制过程。这样,则无需更改系统内原有的Vsync信号响应机制。

[0047] 需要说明的是,在图4中步骤104可以是上述一个Vsync周期内确定出第一个有效滑动轨迹之后执行的步骤,该步骤可能发生在步骤102或步骤103之前,也可能发生在步骤102以及步骤103之后,图4中示出了该步骤104发生在步骤103之后的一种情况,而步骤105则需在接收到Vsync信号后执行。

[0048] 本公开实施例的UI绘制方法可在终端中运行,终端屏幕上显示该应用的界面,用户在终端触摸屏幕上进行滑动操作后,在用户手指触碰到触摸屏幕后产生Action_Down事件之后的第一个导致帧绘制的Action_Move后即执行UI的绘制,该UI的绘制包括App层的绘制以及SurfaceFlinger层的绘制,无需等待Vsync信号到后再进行UI的绘制。

[0049] 在一个示例性实施例中,UI绘制方法可包括如下操作:

[0050] 在系统中添加一个dispatchMotionEvent函数,基于该函数在收到Action_Down事件(用户触摸终端触摸屏的事件)之后,记下该Action_Down对应的坐标。在接收到Action_Move(在终端触摸屏上滑动/移动的事件)之后,计算Action_Down和Action_Move之间的位移是否超过系统属性设置的TouchSlop值,如果超过,说明应用层会基于当前滑动操作进行UI绘制,则将该Action_Down以及Action_Move的参数(该参数为上述滑动轨迹对应的参数的一个示例)发送给应用层,该参数例如可包括:Action_Down以及Action_Move的坐标,或者,Action_Down以及Action_Move所产生的矢量信息,例如位移大小,位移方向。如果不超过,则需要调整判断条件为:

[0051]
$$\text{int syntheticSlop} = ((\text{mTouchSlop} * \text{RESAMPLE_MAX_PREDICTION}) / (2 * \text{RESAMPLE_MAX_PREDICTION} - \text{RESAMPLE_LATENCY})) + 1$$
,其中,syntheticSlop为补偿后的距离,mTouchSlop是终端系统内配置的上述阈值,RESAMPLE_MAX_PREDICTION为最大重采样预测时间,RESAMPLE_LATENCY为重采样延迟,即根据该公式在滑动操作产生的位移的基础上再补偿一定距离,如果补偿后的位移超过TouchSlop的值,则将Action_Down以及Action_Move的参数发送给应用层,应用层在接收到该参数后,根据该参数进行UI绘制。

[0052] 在应用层完成其UI绘制过程后,立即通知SurfaceFlinger层进行绘制过程。

[0053] 图5是App层以及SurfaceFlinger层均等待接收到Vsync信号后再执行UI绘制过程的示意图,在图5中,在接收到第一个Vsync信号之前,App层接收到Action_Down事件,在接收到第一个Vsync信号后,App层接收到Action_Move事件,在接收到第二个Vsync信号后,App层开始进行UI在App层的绘制过程(即,doFrame drawFrame的过程),在接收到第三个

Vsync信号后, SurfaceFlinger层开始进行UI在SurfaceFlinger层的绘制过程(即docompostion的过程)。

[0054] 图6是App层以及SurfaceFlinger层不等待Vsync信号执行UI绘制过程的示意图, 在图6中, 在接收到第一Vsync信号之前, App层接收到Action_Down事件, 在接收到第二个Vsync信号之前接收到Action_Move事件, 此处, 假该Action_Move事件以及Action_Down事件产生的滑动轨迹为有效滑动轨迹, 进程UI在App层的绘制过程, 在App层完成该绘制过程后, 通知SurfaceFlinger层开始进行UI在SurfaceFlinger层的绘制过程。

[0055] 对比图5以及图6可以看出, 在图6中, UI在App层以及SurfaceFlinger层的绘制过程均较图5中的相应绘制过程提前了一段时间。

[0056] 由于系统中应用层通常不存在可以与SurfaceFlinger层进行通信的接口, 因此可在应用窗口创建的过程中保存通信接口, 应用层在完成UI绘制过程后, 可基于该接口通知SurfaceFlinger层进行绘制。

[0057] 图7是根据一示例性实施例示出的一种UI绘制装置的框图, 如图7所示, 该装置50包括:

[0058] 确定模块51, 用于在一个Vsync周期内接收到Vsync信号之前, 根据获取到的针对应用的滑动操作所产生的位移确定出第一个有效滑动轨迹;

[0059] 第一发送模块52, 用于将所述第一个有效滑动轨迹的参数发送给App层, 以使App层基于该参数进行UI在App层的绘制过程;

[0060] 通知模块53, 用于在App层完成UI在App层的绘制过程后, 通知SurfaceFlinger层进行UI在SurfaceFlinger层的绘制过程。

[0061] 图8是根据一示例性实施例示出的一种UI绘制装置的框图, 如图8所示, 该装置60在上述装置50的基础上, 还可包括保存模块61, 该保存模块用于在创建应用的窗口时, 保存通信接口, 该通信接口作为通知SurfaceFlinger层进行UI在SurfaceFlinger层的绘制过程的通信接口, App层在完成UI在App层的绘制过程后, 通过该通信接口通知SurfaceFlinger层进行UI在SurfaceFlinger层的绘制。

[0062] 图9是根据一示例性实施例示出的一种UI绘制装置的框图, 如图9所示, 装置50中的确定模块51可包括: 第一确定子模块71, 用于确定获取到的滑动轨迹的位移是否超过第一阈值; 第二确定子模块72, 用于在滑动轨迹的位移超过第一阈值时, 确定该滑动轨迹为有效滑动轨迹; 第三确定子模块73, 用于在滑动轨迹的位移未超过第一阈值时, 对滑动轨迹的位移进行补偿; 第四确定子模块74, 用于确定补偿后的位移是否超过第二阈值; 第五确定子模块75, 用于在补偿后的位移超过第二阈值时, 确定滑动轨迹为有效滑动轨迹; 第六确定子模块76, 用于在补偿后的位移未超过第二阈值时, 确定滑动轨迹为无效滑动轨迹。

[0063] 在一种可实现方式中, 上述第一阈值和/或第二阈值为终端中配置的TouchSlop值。

[0064] 图10是根据一示例性实施例示出的一种UI绘制装置的框图, 如图10所示, 该装置80在上述装置50的基础上, 还可包括获取模块81以及第二发送模块82, 其中, 获取模块81用于在一个Vsync周期内确定出第一个有效滑动轨迹之后, 获取其他针对应用的滑动操作所产生的其他有效滑动轨迹; 第二发送模块82用于待接收到Vsync信号后, 将其他有效滑动轨迹对应的参数发送给App层, 以使App层基于其他有效滑动轨迹对应的参数进行UI在App层

的绘制过程。

[0065] 图11是根据一示例性实施例示出的一种UI绘制装置800的框图。例如,装置800可以是移动电话,计算机,数字广播终端,消息收发设备,游戏控制台,平板设备,医疗设备,健身设备,个人数字助理等。

[0066] 参照图11,装置800可以包括以下一个或多个组件:处理组件802,存储器804,电源组件806,多媒体组件808,音频组件810,输入/输出(I/O)的接口812,传感器组件814,以及通信组件816。

[0067] 处理组件802通常控制装置800的整体操作,诸如与显示,电话呼叫,数据通信,相机操作和记录操作相关联的操作。处理组件802可以包括一个或多个处理器820来执行指令,以完成上述的方法的全部或部分步骤。此外,处理组件802可以包括一个或多个模块,便于处理组件802和其他组件之间的交互。例如,处理组件802可以包括多媒体模块,以方便多媒体组件808和处理组件802之间的交互。

[0068] 存储器804被配置为存储各种类型的数据以支持在装置800的操作。这些数据的示例包括用于在装置800上操作的任何应用程序或方法的指令,联系人数据,电话簿数据,消息,图片,视频等。存储器804可以由任何类型的易失性或非易失性存储设备或者它们的组合实现,如静态随机存取存储器(SRAM),电可擦除可编程只读存储器(EEPROM),可擦除可编程只读存储器(EPROM),可编程只读存储器(PROM),只读存储器(ROM),磁存储器,快闪存储器,磁盘或光盘。

[0069] 电源组件806为装置800的各种组件提供电力。电源组件806可以包括电源管理系统,一个或多个电源,及其他与为装置800生成、管理和分配电力相关联的组件。

[0070] 多媒体组件808包括在所述装置800和用户之间的提供一个输出接口的屏幕。在一些实施例中,屏幕可以包括液晶显示器(LCD)和触摸面板(TP)。如果屏幕包括触摸面板,屏幕可以被实现为触摸屏,以接收来自用户的输入信号。触摸面板包括一个或多个触摸传感器以感测触摸、滑动和触摸面板上的手势。所述触摸传感器可以不仅感测触摸或滑动动作的边界,而且还检测与所述触摸或滑动操作相关的持续时间和压力。在一些实施例中,多媒体组件808包括一个前置摄像头和/或后置摄像头。当装置800处于操作模式,如拍摄模式或视频模式时,前置摄像头和/或后置摄像头可以接收外部的多媒体数据。每个前置摄像头和后置摄像头可以是一个固定的光学透镜系统或具有焦距和光学变焦能力。

[0071] 音频组件810被配置为输出和/或输入音频信号。例如,音频组件810包括一个麦克风(MIC),当装置800处于操作模式,如呼叫模式、记录模式和语音识别模式时,麦克风被配置为接收外部音频信号。所接收的音频信号可以被进一步存储在存储器804或经由通信组件816发送。在一些实施例中,音频组件810还包括一个扬声器,用于输出音频信号。

[0072] I/O接口812为处理组件802和外围接口模块之间提供接口,上述外围接口模块可以是键盘,点击轮,按钮等。这些按钮可包括但不限于:主页按钮、音量按钮、启动按钮和锁定按钮。

[0073] 传感器组件814包括一个或多个传感器,用于为装置800提供各个方面的状态评估。例如,传感器组件814可以检测到装置800的打开/关闭状态,组件的相对定位,例如所述组件为装置800的显示器和小键盘,传感器组件814还可以检测装置800或装置800一个组件的位置改变,用户与装置800接触的存在或不存在,装置800方位或加速/减速和装置800的

温度变化。传感器组件814可以包括接近传感器,被配置用来在没有任何的物理接触时检测附近物体的存在。传感器组件814还可以包括光传感器,如CMOS或CCD图像传感器,用于在成像应用中使用。在一些实施例中,该传感器组件814还可以包括加速度传感器,陀螺仪传感器,磁传感器,压力传感器或温度传感器。

[0074] 通信组件816被配置为便于装置800和其他设备之间有线或无线方式的通信。装置800可以接入基于通信标准的无线网络,如WiFi,2G或3G,或它们的组合。在一个示例性实施例中,通信组件816经由广播信道接收来自外部广播管理系统的广播信号或广播相关信息。在一个示例性实施例中,所述通信组件816还包括近场通信(NFC)模块,以促进短程通信。例如,在NFC模块可基于射频识别(RFID)技术,红外数据协会(IrDA)技术,超宽带(UWB)技术,蓝牙(BT)技术和其他技术来实现。

[0075] 在示例性实施例中,装置800可以被一个或多个应用专用集成电路(ASIC)、数字信号处理器(DSP)、数字信号处理设备(DSPD)、可编程逻辑器件(PLD)、现场可编程门阵列(FPGA)、控制器、微控制器、微处理器或其他电子元件实现,用于执行上述方法。

[0076] 在示例性实施例中,还提供了一种包括指令的非临时性计算机可读存储介质,例如包括指令的存储器804,上述指令可由装置800的处理器820执行以完成上述方法。例如,所述非临时性计算机可读存储介质可以是ROM、随机存取存储器(RAM)、CD-ROM、磁带、软盘和光数据存储设备等。

[0077] 在示例性实施例中,还提供了一种包括指令的非临时性计算机可读存储介质,例如包括指令的存储器1932,上述指令可由装置1900的处理组件1922执行以完成上述方法。例如,所述非临时性计算机可读存储介质可以是ROM、随机存取存储器(RAM)、CD-ROM、磁带、软盘和光数据存储设备等。

[0078] 本领域技术人员在考虑说明书及实践这里公开的发明后,将容易想到本公开的其他实施方案。本申请旨在涵盖本公开的任何变型、用途或者适应性变化,这些变型、用途或者适应性变化遵循本公开的一般性原理并包括本公开未公开的本技术领域中的公知常识或惯用技术手段。说明书和实施例仅被视为示例性的,本公开的真正范围和精神由下面的权利要求指出。

[0079] 应当理解的是,本公开并不局限于上面已经描述并在附图中示出的精确结构,并且可以在不脱离其范围进行各种修改和改变。本公开的范围仅由所附的权利要求来限制。

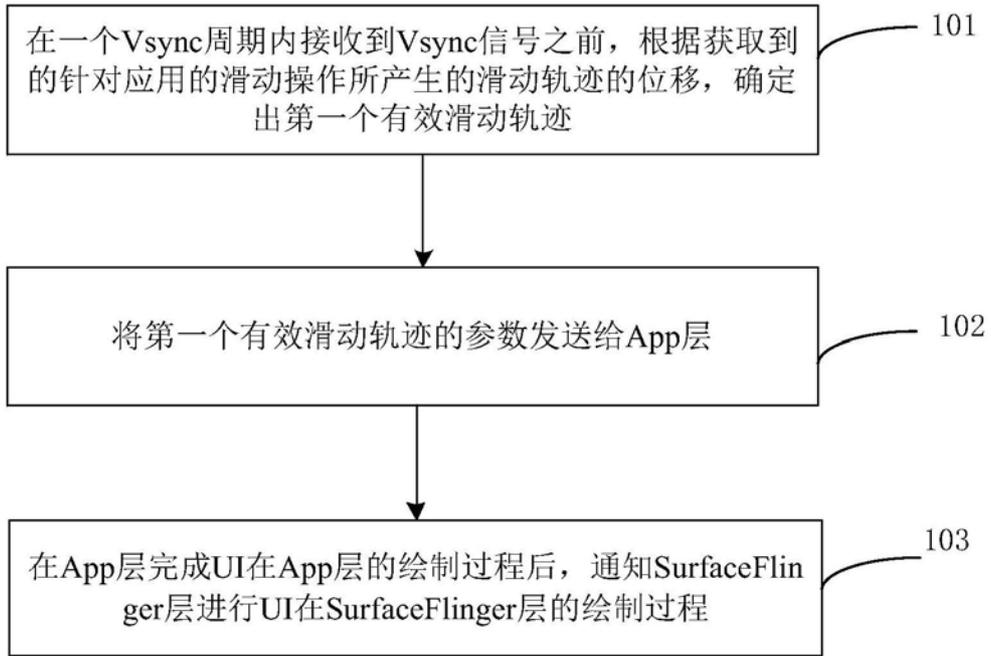


图1

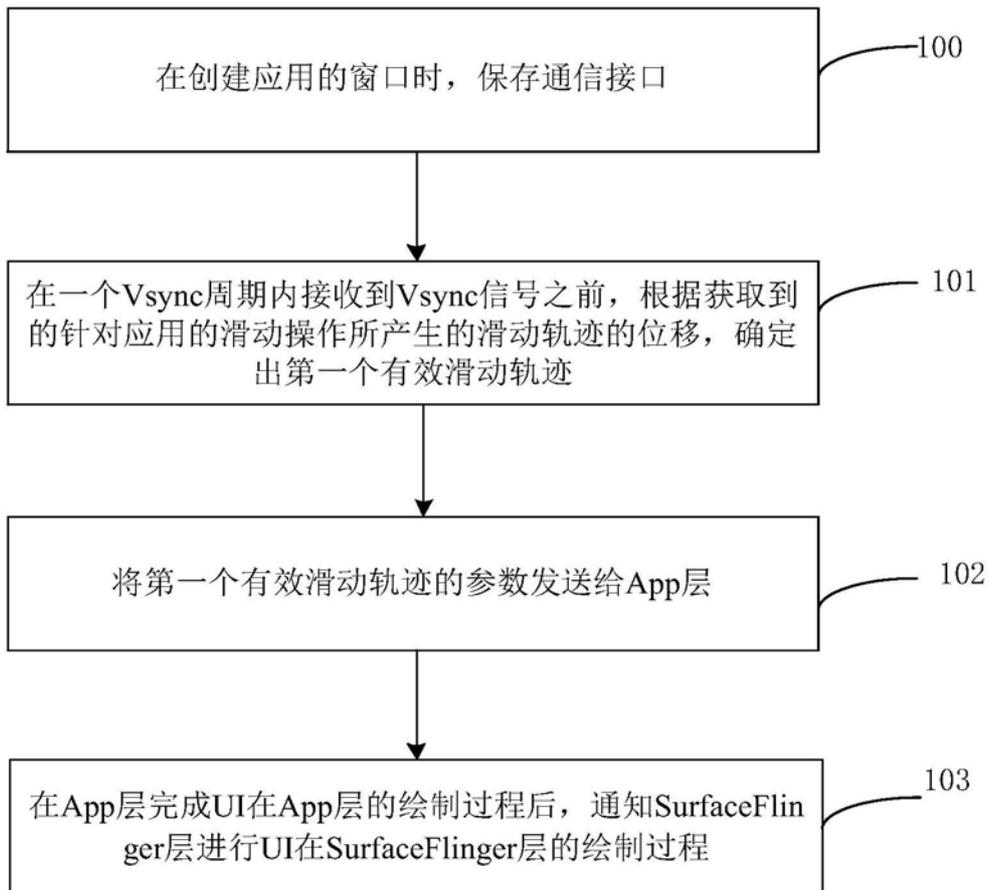


图2

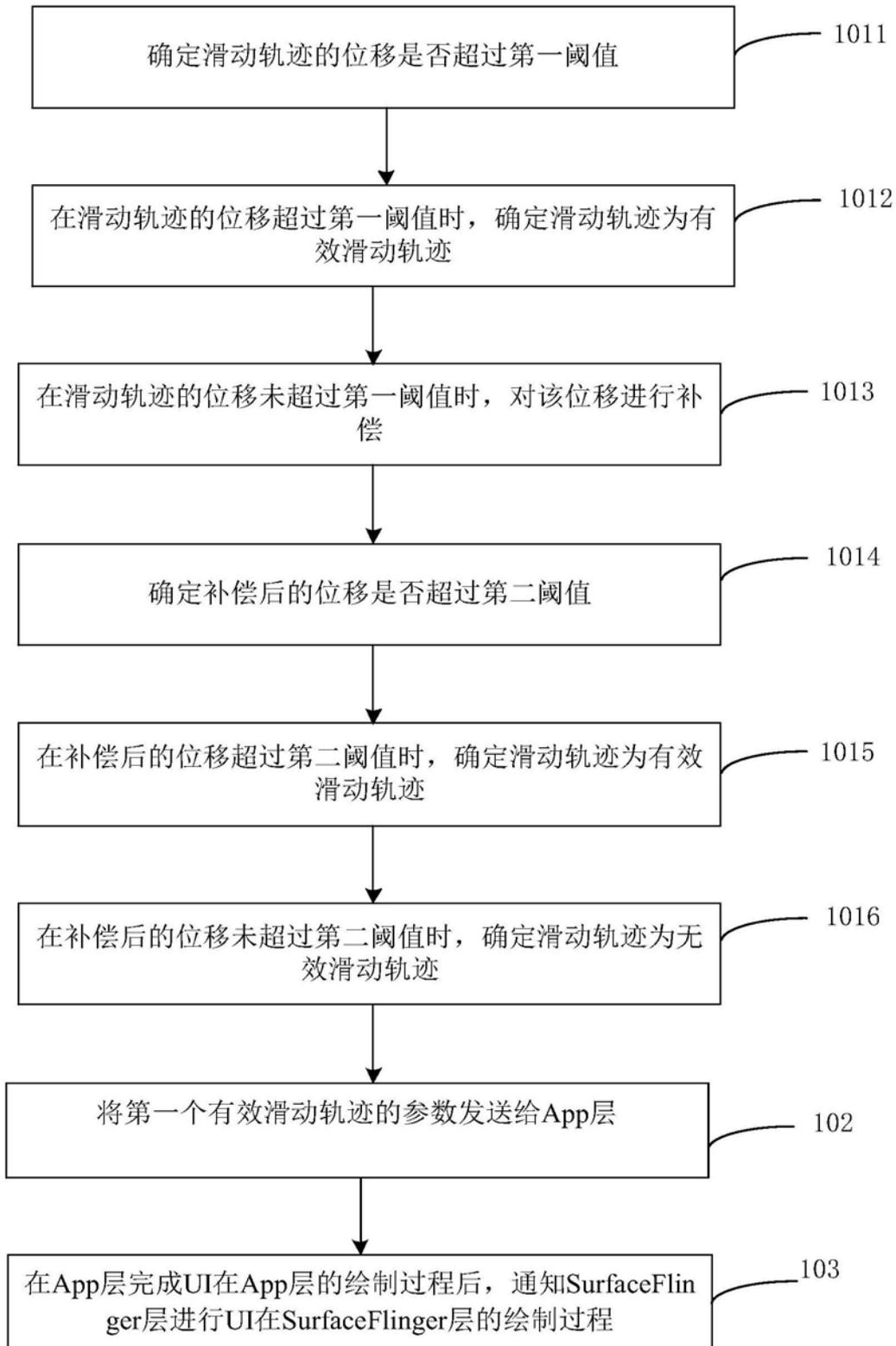


图3

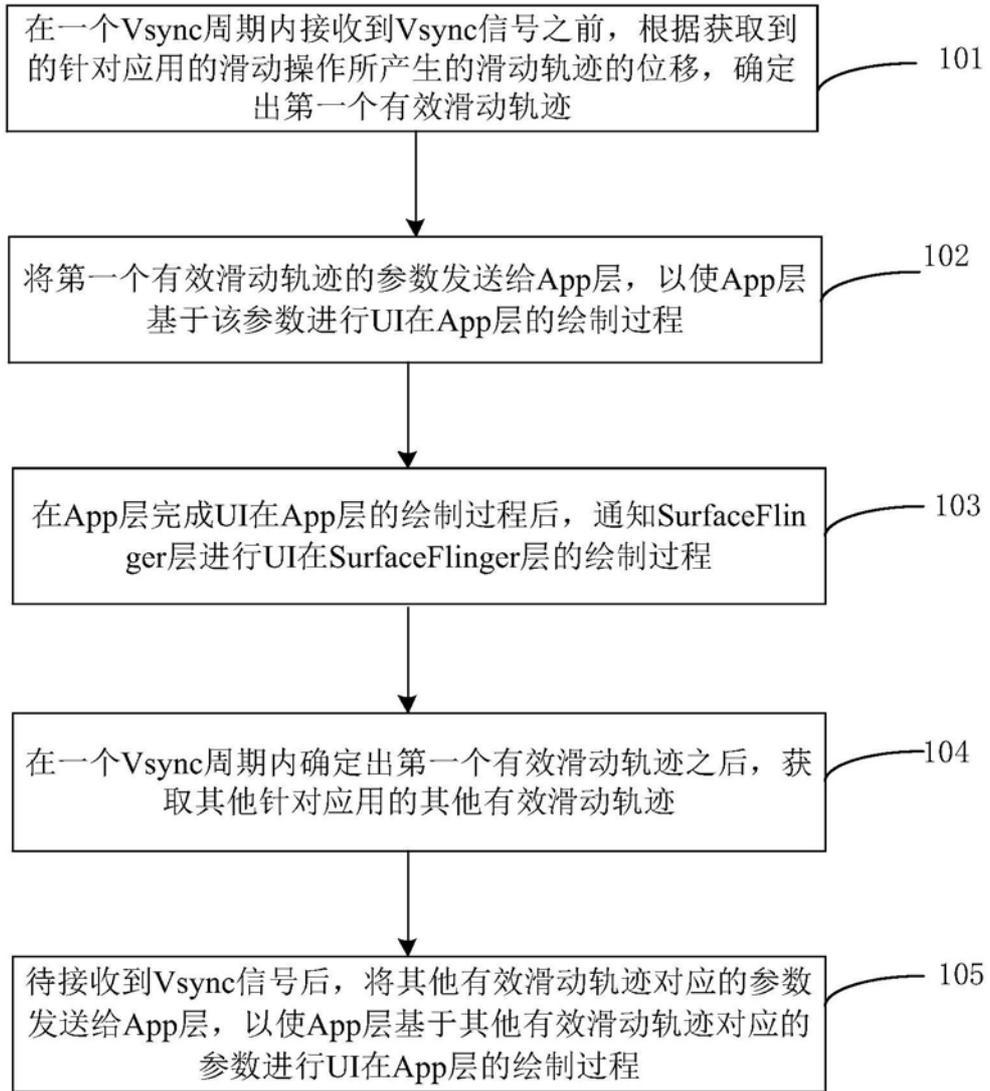


图4

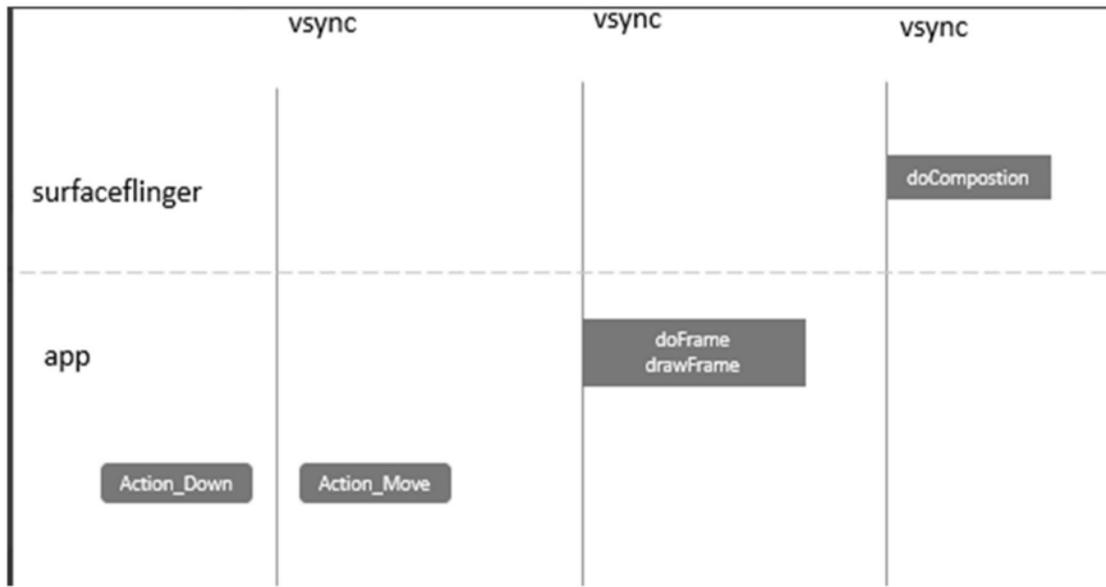


图5

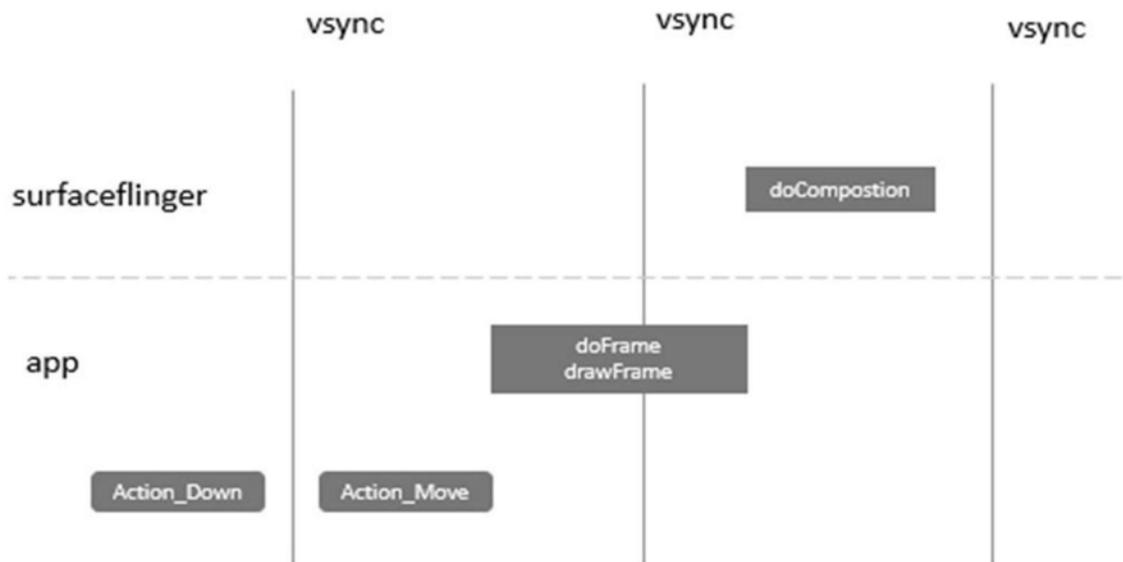


图6

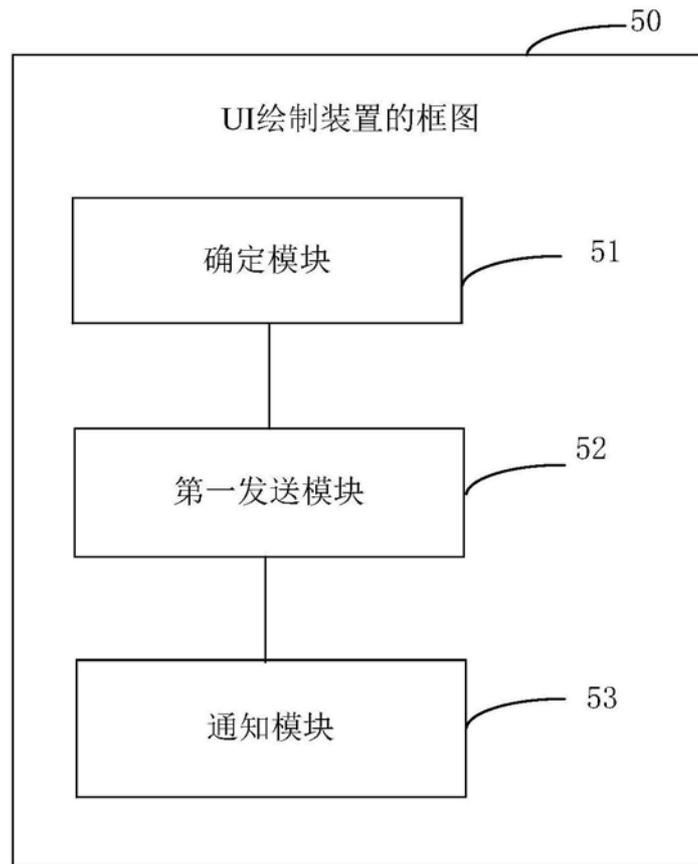


图7

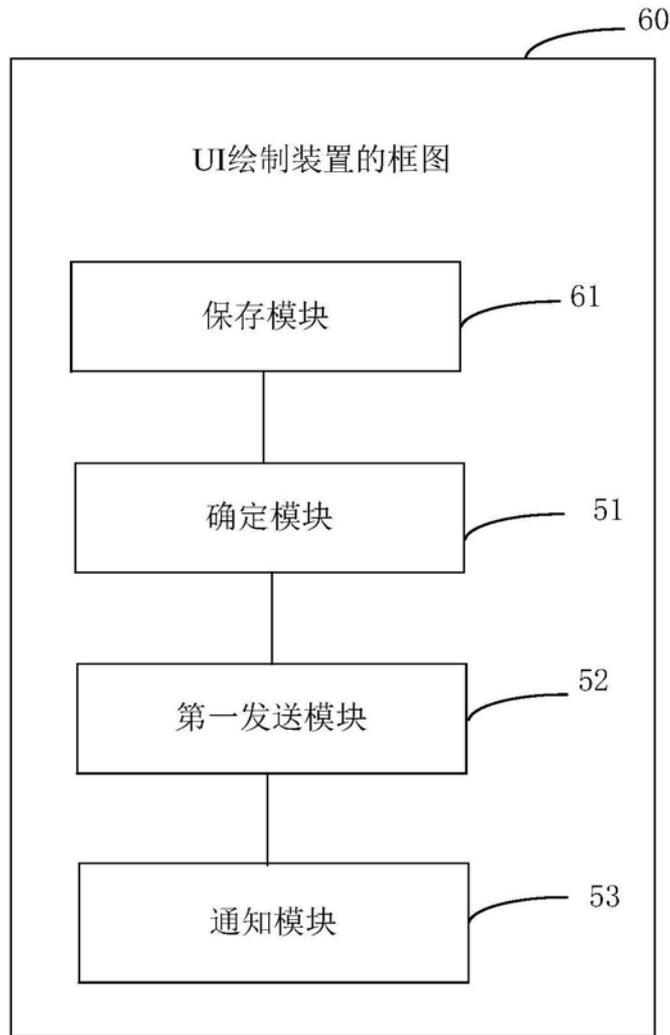


图8

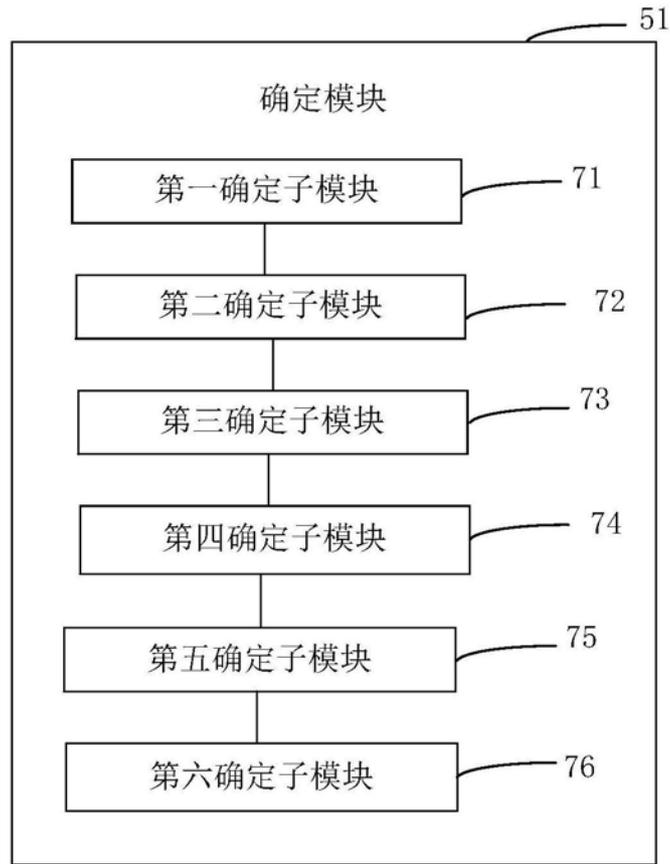


图9

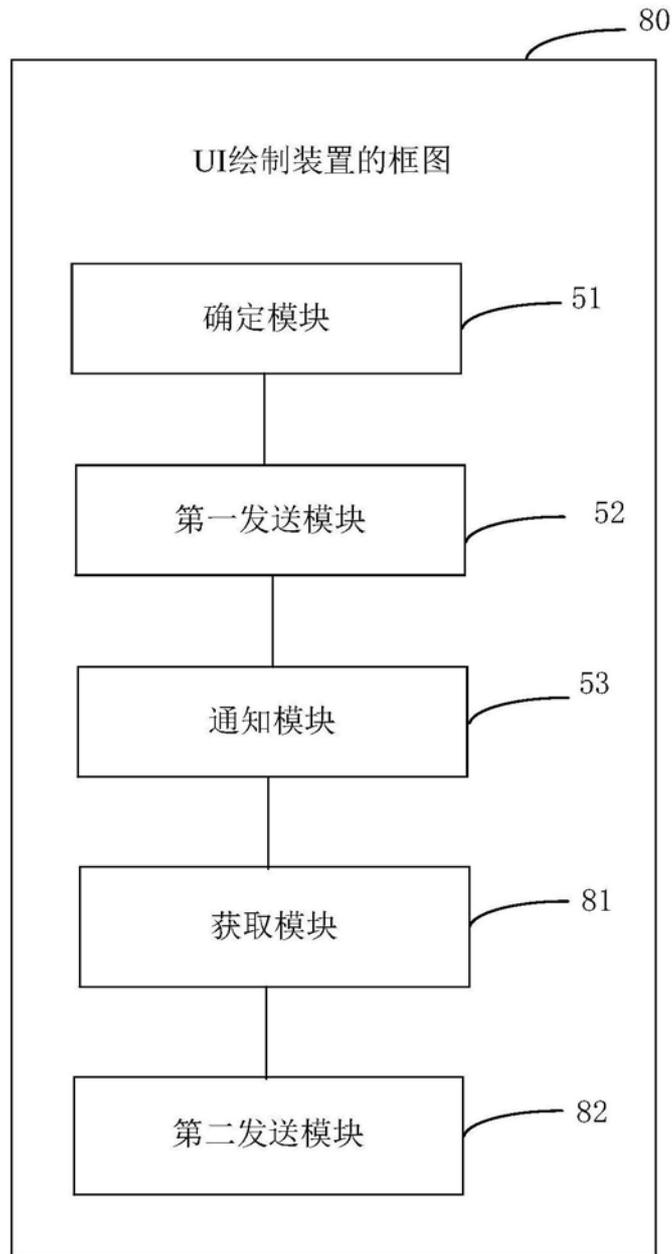


图10

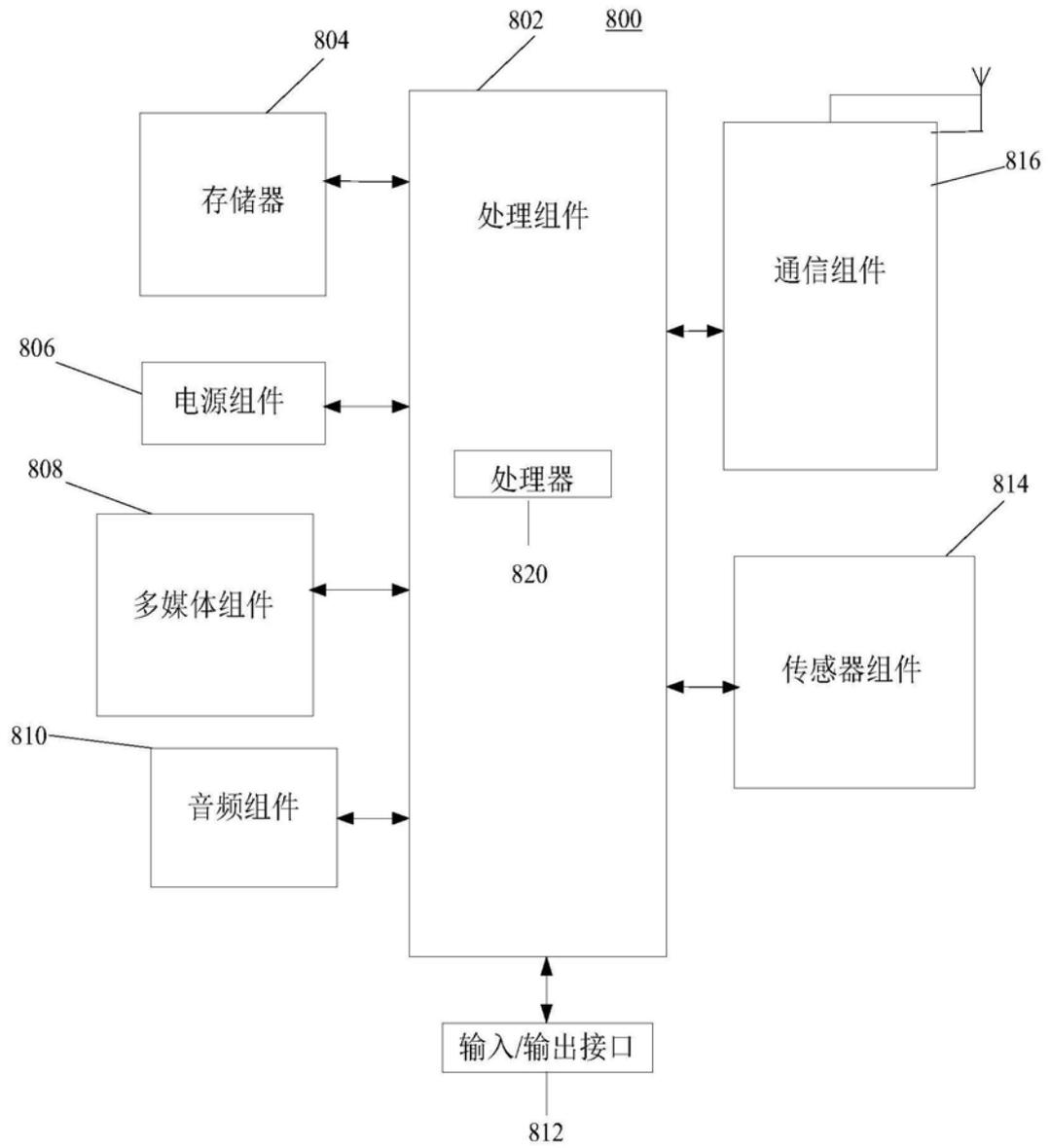


图11