

(12) 按照专利合作条约所公布的国际申请

(19) 世界知识产权组织
国际局

(43) 国际公布日
2024年3月21日 (21.03.2024)



(10) 国际公布号
WO 2024/055904 A1

- (51) 国际专利分类号:
G06F 9/451 (2018.01)
- (21) 国际申请号: PCT/CN2023/117580
- (22) 国际申请日: 2023年9月7日 (07.09.2023)
- (25) 申请语言: 中文
- (26) 公布语言: 中文
- (30) 优先权:
202211116851.3 2022年9月14日 (14.09.2022) CN
- (71) 申请人: 荣耀终端有限公司 (HONOR DEVICE CO., LTD.) [CN/CN]; 中国广东省深圳市福田区香蜜湖街道红荔西路8089号深业中城6号楼A单元3401, Guangdong 518040 (CN)。
- (72) 发明人: 忻振文 (XIN, Zhenwen); 中国广东省深圳市福田区香蜜湖街道红荔西路8089号深业中城6号楼A单元3401, Guangdong 518040 (CN)。孙文涌 (SUN, Wenyong); 中国广东省深圳市福

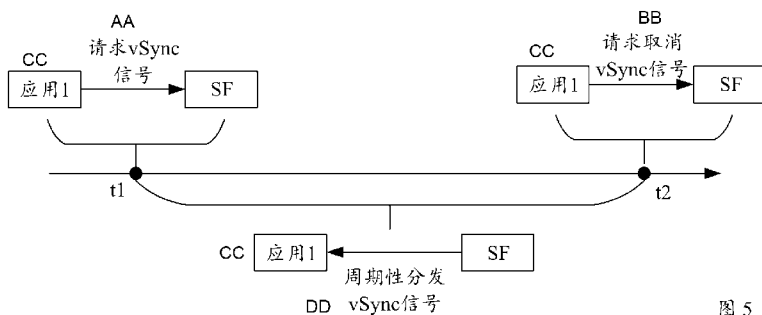
田区香蜜湖街道红荔西路8089号深业中城6号楼A单元3401, Guangdong 518040 (CN)。陈川福 (CHEN, Chuanfu); 中国广东省深圳市福田区香蜜湖街道红荔西路8089号深业中城6号楼A单元3401, Guangdong 518040 (CN)。李美君 (LI, Meijun); 中国广东省深圳市福田区香蜜湖街道红荔西路8089号深业中城6号楼A单元3401, Guangdong 518040 (CN)。

(74) 代理人: 北京中博世达专利商标代理有限公司 (BEIJING ZBSD PATENT & TRADEMARK AGENT LTD.); 中国北京市海淀区交大东路31号11号楼8层, Beijing 100044 (CN)。

(81) 指定国(除另有指明, 要求每一种可提供的国家保护): AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CV, CZ, DE, DJ, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IQ, IR, IS, IT, JM, JO, JP, KE, KG, KH, KN, KP, KR, KW, KZ,

(54) Title: METHOD FOR REQUESTING VSYNC SIGNAL, AND ELECTRONIC DEVICE

(54) 发明名称: 一种请求vSync信号的方法及电子设备



(57) Abstract: The present application relates to the technical field of image display, provides a method for requesting a vSync signal, and an electronic device, and can reduce the possibility of stuttering and unsmoothness when dynamically changing continuous multi-frame images need to be displayed. At a first moment, a first application in the electronic device calls a setVsyncRate interface of a window system SF in the electronic device, and sends a first request to the SF, wherein the first request is used for requesting the vSync signal. In response to the first request, the SF periodically distributes the vSync signal to the first application. At a second moment, the first application calls the setVsyncRate interface, and sends a second request to the SF, wherein the second request is used for requesting to cancel the vSync signal, and after receiving the second request, the SF ends periodically distributing the vSync signal to the first application.

(57) 摘要: 本申请提供一种请求vSync信号的方法及电子设备, 涉及图像显示技术领域, 在需要显示动态变化的连续多帧图像时, 可以降低卡顿、不流畅的可能性。在第一时刻, 电子设备中的第一应用调用电子设备中窗口系统SF的setVsyncRate接口, 向SF发送第一请求, 第一请求用于请求vSync信号。响应于第一请求, SF周期性向第一应用分发vSync信号。在第二时刻, 第一应用调用setVsyncRate接口, 向SF发送第二请求, 第二请求用于请求取消vSync信号, SF在接收到第二请求之后, 结束周期性向第一应用分发vSync信号。

LA, LC, LK, LR, LS, LU, LY, MA, MD, MG, MK, MN,
MU, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA,
PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD,
SE, SG, SK, SL, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ,
UA, UG, US, UZ, VC, VN, WS, ZA, ZM, ZW。

- (84) 指定国(除另有指明, 要求每一种可提供的地区
保护): ARIPO (BW, CV, GH, GM, KE, LR, LS, MW, MZ,
NA, RW, SC, SD, SL, ST, SZ, TZ, UG, ZM, ZW), 欧亚
(AM, AZ, BY, KG, KZ, RU, TJ, TM), 欧洲 (AL, AT, BE,
BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR,
HU, IE, IS, IT, LT, LU, LV, MC, ME, MK, MT, NL, NO,
PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF,
CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN,
TD, TG)。

本国际公布:

- 包括国际检索报告(条约第21条(3))。

一种请求 vSync 信号的方法及电子设备

5 本申请要求于 2022 年 9 月 14 日提交国家知识产权局、申请号为 202211116851.3、
发明名称为“一种请求 vSync 信号的方法及电子设备”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

本申请涉及图像显示技术领域，尤其涉及一种请求 vSync 信号的方法及电子设备。

10 背景技术

手机、平板等电子设备的显示屏亮屏的情况下，应用在需要显示新一帧图像时，
可以请求垂直同步(Vertical Synchronization, vSync)信号，并且电子设备的显示屏在一
次刷新完成后，可以将 vSync 信号分发给请求 vSync 信号的应用，即分发给有渲染需
求的应用。vSync 信号可以触发应用完成新一帧图像的渲染，然后经过合成，最终送
15 显。如此，在显示屏刷新一次后，才触发有渲染需求的应用开始渲染新一帧图像，然
后合成并送显。从而可以保证画面每秒传输帧数(Frame Per Second, FPS)不会超过
显示屏的刷新率。例如，刷新率为 60 赫兹(HZ)，则可以保证 FPS 不超过每秒 60 帧。

然而，发明人在实施本申请实施例的过程中发现，在现有技术中，在电子设备的
负载较高时，如中央处理器(Central Processing Unit, CPU)的占用超过 97%以上时，
20 应用请求 vSync 信号的时延会很长。若未及时收到应用对 vSync 信号的请求，自然就
不会将 vSync 信号分发给该应用，从而无法及时触发应用完成新一帧图像的渲染，导
致显示屏无法获得该新一帧图像，最终只能显示旧的图像。尤其在需要显示动态变化
的连续多帧图像时，若出现上述问题，则会造成动态变化的过程卡顿、不流畅。

发明内容

25 有鉴于此，本申请提供了一种请求 vSync 信号的方法及电子设备，在需要显示动
态变化的连续多帧图像时，可以降低卡顿、不流畅的可能性。

第一方面，本申请实施例提供一种请求 vSync 信号的方法，用于具有显示功能的
电子设备，如应用于手机、平板中，电子设备中安装有第一应用。其中，在第一时刻，
电子设备中的第一应用调用电子设备中窗口系统 SF 的 setVsyncRate 接口，向 SF 发送
30 第一请求，第一请求用于请求 vSync 信号。响应于第一请求，SF 周期性向第一应用分
发 vSync 信号。

应理解，SF 周期性向第一应用分发 vSync 信号，则第一应用可以周期性的接收到
vSync 信号。vSync 信号可以触发渲染图像。第一应用周期性的接收到 vSync 信号，则
可以周期性的渲染图像。这样，第一应用仅请求一次 vSync 信号，则可以完成多帧图
35 像的渲染，而无需针对多帧图像请求多次 vSync 信号。从而可以减少应用进程和 SF
进程之间的跨进程调用，降低了对 Binder 线程的依赖。如此，在需要显示动态变化的
连续多帧图像的场景中，可以降低出现显示卡顿、不流畅的现象的可能。

并且，在第二时刻，第一应用调用 setVsyncRate 接口，向 SF 发送第二请求，第

二请求用于请求取消 vSync 信号, SF 在接收到第二请求之后, 结束周期性向第一应用分发 vSync 信号。如此, 可以在不需要 SF 周期性分发 vSync 信号后, 及时触发 SF 结束向第一应用周期性分发 vSync 信号, 避免长期周期性分发造成资源浪费。

5 在一种可能的设计方式中, 上述方法还包括: 第一应用确定需要渲染连续多帧图像, 第一时刻包括请求渲染第一图像的时刻, 第一图像为多帧图像中的第一帧图像。如此, 在需要渲染多帧图像中的第一帧图像时, 则调用 setVsyncRate 接口向 SF 请求 vSync 信号, 从而可以在确定需要渲染连续多帧图像后, 及时使 SF 开始周期性分发 vSync 信号, 触发依次完成多帧图像的渲染。

10 在一种可能的设计方式中, 第一应用记录第一开关状态, 电子设备的框架层的渲染模块中记录第二开关状态, 第一开关状态和第二开关状态的初始值均为关闭状态。在第一应用确定需要渲染连续多帧图像之后, 上述方法还包括: 第一应用更新第一开关状态为开启状态, 使第一开关状态指示需要渲染连续多帧图像。

15 上述在第一时刻, 电子设备中的第一应用调用电子设备中窗口系统 SF 的 setVsyncRate 接口, 向 SF 发送第一请求, 包括: 在第一时刻, 第一应用向渲染模块发送第一消息, 第一消息用于请求 vSync 信号。响应于第一消息, 且第一开关状态为开启状态, 第二开关状态为关闭状态, 渲染模块调用 setVsyncRate 接口, 向 SF 发送第一请求。

20 其中, 第一开关状态为开启状态, 第二开关状态为关闭状态, 表明第一应用刚从不需要渲染连续多帧图像转变为需要渲染连续多帧图像, 因此, 渲染模块需要调用 setVsyncRate 接口来请求 vSync 信号, 及时触发 SF 周期性分发 vSync 信号。

在一种可能的设计方式中, 第一开关状态为开启状态, 第二开关状态为关闭状态, 上述方法还包括: 渲染模块更新第二开关状态为开启状态, 从而使渲染模块中的第二开关状态也可以同步指示第一应用需要渲染连续多帧图像。

25 在一种可能的设计方式中, 上述方法还包括: 在第三时刻, 第一应用不向 SF 请求 vSync 信号, 第三时刻为请求渲染第二图像的时刻, 第二图像为多帧图像中除第一帧图像之外的任一图像。如此, 在需要渲染多帧图像中的第二帧图像以及之后的图像时, 第一应用不会重复调用 setVsyncRate 接口向 SF 请求 vSync 信号。从而可以在需要渲染连续多帧图像的情况下, 仅请求一次 vSync 信号, 减少跨进程调用。

30 具体地, 在第三时刻, 第一应用向渲染模块发送第一消息, 第一消息用于请求 vSync 信号。响应于第一消息, 且第一开关状态为开启状态, 第二开关状态为开启状态, 渲染模块不向 SF 请求 vSync 信号。其中, 第一开关状态为开启状态, 表示第一应用需要渲染连续多帧图像的需求。但是, 第二开关状态也为开启状态, 表明第一应用并不是刚从不需要渲染连续多帧图像转变为需要渲染连续多帧图像, 因此, 渲染模块不需要重复调用 setVsyncRate 接口来请求 vSync 信号。那么, 在第一开关状态为开启状态, 第二开关状态为开启状态的情况下, 无需重复调用 requestNextVsync 接口来请求 vSync 信号。从而可以避免重复调用 requestNextVsync 接口来请求 vSync 信号, 保证在需要渲染连续多帧图像的场景中, 仅通过调用 requestNextVsync 接口来请求一次 vSync 信号。

在一种可能的设计方式中, 上述方法还包括: 第一应用确定结束渲染连续多帧图

像，第二时刻包括请求渲染第三图像的时刻，第三图像为完成多帧图像的渲染后，第一应用需要渲染的第一帧图像。如此，在多帧图像渲染完成后，需要渲染新一帧图像时，则调用 setVsyncRate 接口向 SF 请求取消 vSync 信号，从而可以在多帧图像渲染完成后，及时使 SF 结束周期性分发 vSync 信号，节省资源。

5 在一种可能的设计方式中，在第一应用确定结束渲染连续多帧图像之后，上述方法还包括：第一应用更新第一开关状态为关闭状态。上述在第二时刻，第一应用调用 setVsyncRate 接口，向 SF 发送第二请求，包括：在第二时刻，第一应用向渲染模块发送第一消息，第一消息用于请求 vSync 信号。响应于第一消息，且第一开关状态为关闭状态，第二开关状态为开启状态，渲染模块调用 setVsyncRate 接口，向 SF 发送第
10 二请求。

其中，第一开关状态为关闭状态，第二开关状态为开启状态，表明第一应用刚从需要渲染连续多帧图像转变为不需要渲染连续多帧图像，因此，渲染模块需要先调用 setVsyncRate 接口来请求取消 vSync 信号，避免在结束渲染连续多帧图像的需求后，SF 仍然周期性的分发 vSync 信号。

15 在一种可能的设计方式中，第一开关状态为关闭状态，第二开关状态为开启状态，上述方法还包括：渲染模块更新第二开关状态为关闭状态，从而使渲染模块中的第二开关状态也可以同步指示第一应用不需要渲染连续多帧图像。

在一种可能的设计方式中，在向 SF 发送第二请求之后，上述方法还包括：第一应用调用 SF 的 requestNextVsync 接口，向 SF 发送第三请求，第三请求用于请求 vSync 信号。响应于第三请求，SF 向第一应用分发一次 vSync 信号。
20

也就是说，采用本实施例的方法，可以在触发 SF 结束周期性分发 vSync 信号后，进一步向 SF 请求 vSync 信号，使 SF 可以分发 vSync 信号来触发第一应用渲染多帧图像之后的第一帧图像。从而可以在结束渲染连续多帧图像后，请求一次 vSync 信号，则分发一次 vSync 信号。

25 在一种可能的设计方式中，SF 周期性向第一应用分发 vSync 信号，包括：SF 周期性向渲染模块分发 vSync 信号。应理解，第一应用需要渲染模块来实现渲染，SF 向渲染模块分发 vSync 信号，则可以触发渲染模块来完成第一应用的图像渲染。因此，SF 向渲染模块分发 vSync 信号，相当于实现了向第一应用分发 vSync 信号。

在一种可能的设计方式中，上述方法还包括：在第一应用需要渲染第四图像时，
30 向渲染模块发送第一消息，第四图像是第一应用需要显示的任一帧图像，第四图像包括第一图像或者第二图像。响应于第一消息，渲染模块记录第一标记，并在接收到来自 SF 的 vSync 信号后，删除第一标记。渲染模块在每次接收到来自 SF 的 vSync 信号后，可以判断处理状态是否为未处理状态。若处理状态为未处理状态，表明第一应用有需要渲染的图像，则可以触发渲染模块渲染该图像，并更新处理状态为已处理状态。
35 若处理状态不是未处理状态，则表明第一应用并没有需要渲染的图像。也就是说，当前接收到的 vSync 信号是多余的。应理解，若通过调用 requestNextVsync 接口向 SF 请求 vSync 信号，SF 是请求一次，分发一次 vSync 信号，不会分发多余的 vSync 信号。那么，该多余的 vSync 信号极有可能是通过调用 setVsyncRate 接口向 SF 请求 vSync 信号后，SF 周期性分发的。第一应用在第二时刻向 SF 请求取消 vSync 信号，则该第

二时刻可以为渲染模块在不包括第一标记的情况下,接收到来自 SF 的 vSync 信号的时
刻。如此,渲染模块可以在发现多余的 vSync 信号后,及时触发 SF 结束周期性分发
vSync 信号。

5 在一种可能的设计方式中,上述方法还包括:在第二时刻,渲染模块通知第一应
用将第一开关状态更新为关闭状态,渲染模块更新第二开关状态为关闭状态,使第一
开关状态和第二开关状态均指示结束渲染连续多帧图像。

10 在一种可能的设计方式中,上述方法还包括:在第四时刻,第一应用向渲染模块
发送第一消息,第一消息用于请求 vSync 信号。响应于第一消息,且第一开关状态和
第二开关状态均为关闭状态,渲染模块调用 requestNextVsync 接口,向 SF 发送第四请
求,第四请求用于请求 vSync 信号。

15 其中,第一开关状态为关闭状态,则表示第一应用不需要渲染连续多帧图像。并
且,渲染模块本地的开关状态为关闭状态,表示此时并不是刚结束渲染连续多帧图像。
因此,渲染模块无需调用 requestNextVsync 接口来请求取消 vSync 信号。那么,在第
一开关状态和第二开关状态均为关闭状态的情况下,则可以调用 requestNextVsync 接
口来请求 vSync 信号。

20 在一种可能的设计方式中,SF 位于电子设备的本地层,第一应用位于电子设备的
应用程序层,在向 SF 发送第一请求之前,上述方法还包括:在电子设备的框架层中新
增第一接口,在电子设备的 Java 本地接口层中新增第二接口,第一接口和第二接口是
setVsyncRate 接口对 Java 开放的接口。第一应用调用 setVsyncRate 接口,包括:第
一应用依次经过第一接口和第二接口调用 setVsyncRate 接口。

也就是说,采用本实施例,通过第一接口、第二接口和 setVsyncRate 接口,打通
第一应用到本地层中的 SF 之间的通路,使得应用可以调用 setVsyncRate 接口,实现
向 SF 请求 vSync 信号。

25 在一种可能的设计方式中,第一请求中包括周期 n , $n \geq 1$, n 为整数。响应于第一
请求,SF 周期性向第一应用分发 vSync 信号,包括:响应于第一请求,SF 在每接收
到 n 次 vSync 信号后,向第一应用分发一次 vSync 信号。

也就是说,采用本实施例,第一应用可以根据显示需求,灵活指定 SF 分发 vSync
信号的周期。

30 第二方面,本申请实施例提供一种电子设备,电子设备包括存储器和处理器,所
述存储器和所述处理器耦合;其中,存储器中存储有计算机程序代码,计算机程序代
码包括计算机指令,当计算机指令被处理器执行时,使得电子设备执行如第一方面及
其任一种可能的设计方式的方法。

35 第三方面,本申请实施例提供一种芯片系统,该芯片系统应用于包括显示屏和存
储器的电子设备;所述芯片系统包括一个或多个接口电路和一个或多个处理器;所述
接口电路和所述处理器通过线路互联;所述接口电路用于从所述电子设备的存储器接
收信号,并向所述处理器发送所述信号,所述信号包括所述存储器中存储的计算机指
令;当所述处理器执行所述计算机指令时,所述电子设备执行如第一方面及其任一种
可能的设计方式所述的方法。

第四方面,本申请提供一种计算机存储介质,该计算机存储介质包括计算机指令,

当所述计算机指令在电子设备上运行时，使得电子设备执行如第一方面及其任一种可能的设计方式所述的方法。

第五方面，本申请提供一种计算机程序产品，当所述计算机程序产品在计算机上运行时，使得所述计算机执行如第一方面及其任一种可能的设计方式所述的方法。

5 可以理解地，上述提供的第二方面所述的电子设备，第三方面所述的芯片系统，第四方面所述的计算机存储介质，第五方面所述的计算机程序产品所能达到的有益效果，可参考第一方面及其任一种可能的设计方式中的有益效果，此处不再赘述。

附图说明

- 图 1 为本申请实施例提供的一种桌面动效的界面示意图；
10 图 2 为本申请实施例提供的一种图像处理过程的示意图；
图 3 为本申请实施例提供的一种图像处理方法的时序图；
图 4 为本申请实施例提供的另一种桌面动效的界面示意图；
图 5 为本申请实施例提供的一种请求 vSync 信号的方法的原理示意图；
图 6 为本申请实施例提供的电子设备的硬件结构图；
15 图 7 为本申请实施例提供的电子设备的软件架构图；
图 8 为本申请实施例提供的一种请求 vSync 信号的方法的时序图；
图 9 为本申请实施例提供的一种请求状态的状态转移示意图；
图 10 为本申请实施例提供的另一种请求 vSync 信号的方法的时序图；
图 11 为本申请实施例提供的又一种请求 vSync 信号的方法的时序图；
20 图 12 为本申请实施例提供的再一种请求 vSync 信号的方法的时序图；
图 13 为本申请实施例提供的一种芯片系统的结构示意图。

具体实施方式

下面结合本申请实施例中的附图，对本申请实施例中的技术方案进行描述。其中，在本申请实施例的描述中，以下实施例中所使用的术语只是为了描述特定实施例的目的，而并非旨在作为对本申请的限制。如在本申请的说明书和所附权利要求书中所使用的那样，单数表达形式“一种”、“所述”、“上述”、“该”和“这一”旨在也
25 包括例如“一个或多个”这种表达形式，除非其上下文中明确地有相反指示。还应当理解，在本申请以下各实施例中，“至少一个”、“一个或多个”是指一个或两个以上（包含两个）。术语“和/或”，用于描述关联对象的关联关系，表示可以存在三种
30 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。

在本说明书中描述的参考“一个实施例”或“一些实施例”等意味着在本申请的一个或多个实施例中包括结合该实施例描述的特定特征、结构或特点。由此，在本说明书中的不同之处出现的语句“在一个实施例中”、“在一些实施例中”、“在其他
35 一些实施例中”、“在另外一些实施例中”等不是必然都参考相同的实施例，而是意味着“一个或多个但不是所有的实施例”，除非是以其他方式另外特别强调。术语“包括”、“包含”、“具有”及它们的变形都意味着“包括但不限于”，除非是以其他方式另外特别强调。术语“连接”包括直接连接和间接连接，除非另外说明。“第一”、

“第二”仅用于描述目的，而不能理解为指示或暗示相对重要性或者隐含指明所指示的技术特征的数量。

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

本申请实施例提供一种请求 vSync 信号的方法，可用于需要显示动态变化的连续多帧图像的场景。其中，需要显示动态变化的连续多帧图像的场景具体是指在用户的触摸操作结束后、应用需要继续显示滑动、缩放等动效的场景。

10 示例性的，参见图 1，手机在检测到用户对桌面 101 上视频应用的应用图标 102 的点击操作后（即点击操作结束后），会依次显示如桌面 103、桌面 104、桌面 105、桌面 106 以及桌面 107 所示的动态变化的动效来过渡，并最终显示视频应用的应用界面（图 1 中未示出）。上述图 1 所示的场景即为需要显示动态变化的连续多帧图像的场景。

15 通常情况下，一帧图像的显示包括渲染、合成以及送显三个步骤。其中，由产生待显示图像的应用执行渲染的步骤，完成待显示图像的各个图层中内容的渲染。然后，应用将渲染得到的图层数据发送给窗口系统（SurfaceFlinger，SF），由 SF 基于图层数据执行合成的步骤，完成图层合成，得到待处理图像。再后，SF 执行送显的步骤，将待显示图像送往显示屏显示。

20 与此同时，参见图 2，数字 1-4 表示第 1-4 帧图像，实线无填充矩形框表示渲染的过程，虚线无填充矩形框表示合成的过程，实线有填充矩形框表示显示的过程。很显然，除第 1 帧图像之外，其它图像的渲染都是由 vSync 信号触发的。具体地，显示屏会按照其刷新率来刷新。例如，显示屏的刷新率为 60HZ，即每秒刷新 60 次，则显示屏会以 $1/60s \approx 16.67ms$ 的时间间隔刷新。当然，显示屏的刷新率也可以是 90Hz、120Hz
25 等。显示屏在完成一次刷新后，会产生 vSync 信号，并通过 SF 分发给应用，以触发应用开始渲染。应用渲染完成后，SF 再进行合成，最后送显。如此，显示屏每刷新一次后，才触发渲染、合成一帧新图像。从而可以保证 FPS 不会超过显示屏的刷新率。

在电子设备的使用过程中，不同应用对渲染的需求是不同的。通常情况下，应用是否有渲染的需求，需要应用基于自身的显示逻辑来确定。以图 1 为例，只有 Launcher
30 知道在检测到用户对桌面上应用图标的点击操作后，需要显示桌面 103-桌面 107 对应的动效，相应的，则需要连续渲染该动效包括的多帧图像。例如，桌面 103-桌面 107 依次对应第 1 帧-第 5 帧动效，则需要连续渲染 5 个动效帧。因此，需要应用在产生渲染需求后，向 SF 请求 vSync 信号，以指示 SF 在接收到来自显示屏的 vSync 信号后进一步分发，从而才能触发有渲染需求的应用（如 Launcher）开始渲染，满足渲染的需求。
35

在一些实施例中，参见图 3，从应用产生渲染需求，到最终显示一帧图像的具体实现包括：

S301、应用 1 在需要渲染图像 1 时，调用 SF 的 requestNextVsync 接口向 SF 请求 vSync 信号。

其中，应用 1（也可以称为第一应用）可以是电子设备中的任一应用，图像 1 是应用 1 需要显示的任一帧图像。SF 可用于将 vSync 信号分发给有渲染需求的应用（如应用 1），从而触发应用（如应用 1）开始渲染。SF 中的 requestNextVsync 接口供应用（如应用 1）用来请求 vSync 信号，即供应用（如应用 1）请求 SF 将 vSync 信号分发给应用（如应用 1），以触发应用（如应用 1）开始渲染。

以应用 1 是 Launcher 为例，在 Launcher 确定需要渲染图 1 所示桌面 103 对应的第 1 帧动效时，则可以调用 SF 的 requestNextVsync 接口请求 vSync 信号，即请求 SF 将 vSync 信号分发给 Launcher，以触发 Launcher 开始渲染第 1 帧动效。

在本实施例中，应用 1 通过调用 requestNextVsync 接口来请求 vSync 信号具有如下特性：应用 1 请求一次 vSync 信号后，SF 仅向应用 1 分发一次 vSync 信号，从而只能触发应用 1 完成图像 1 的渲染。后续，若应用 1 需要继续渲染，如渲染图像 2、图像 3……，则需要再次调用 requestNextVsync 接口来请求 vSync 信号。

示例性的，SF 中记录有应用 1 的请求状态，请求状态包括未请求和已请求两种。请求状态的初始值为未请求。SF 在接收到应用 1 对 vSync 信号的请求后，则更新请求状态为已请求，如下 S302 所示，以指示应用 1 有渲染需求。而在 SF 向应用 1 分发一次 vSync 信号后，则更新请求状态为未请求，如下 S305 所示，以指示应用 1 没有渲染需求。

S302、SF 更新请求状态为已请求。

S303、在一次刷新完成后，显示屏向 SF 发送 vSync 信号。

显示屏在每一次刷新完成后，则会向 SF 发送 vSync 信号，以指示开始下一帧图像的显示流程，如渲染、合成以及送显。

应理解，显示屏会按照其刷新率来刷新显示。例如，显示屏的刷新率为 60 赫兹(HZ)，即每秒刷新 60 次，则显示屏会以 $1/60s \approx 16.67ms$ 的时间间隔刷新显示。

S304、在请求状态为已请求的情况下，SF 向应用 1 分发 vSync 信号。

SF 在接收到 vSync 信号后，可查询应用 1 的请求状态。若查询到请求状态为已请求，则表示应用 1 有渲染需求。该情况下，SF 则将 vSync 信号分发给应用 1，以触发应用 1 开始渲染。

应理解，若有多个应用请求 vSync 信号，则表明多个应用具有渲染需求，SF 需要将 vSync 信号分发给多个应用。

S305、SF 更新请求状态为未请求。

S306、应用 1 响应于 vSync 信号，渲染图像 1。

应用 1 在每次接收到 vSync 信号后，才会开始渲染，以使应用的帧率与显示屏的数显率保持一致。

S307、应用 1 向 SF 发送渲染后的图层数据。

S308、SF 对图像 1 完成合成处理。

SF 还可以用于合成处理，合成处理主要是指将一个或多个图层进行合成。

S309、SF 向显示屏发送合成后的图像 1。

S310、显示屏显示图像 1。

采用本实施例，应用 1 可以在需要渲染任一帧图像时，通过调用 SF 的

requestNextVsync 接口来请求 vSync 信号, 后续 SF 则会向应用 1 分发一次 vSync 信号, 从而触发应用 1 完成一帧图像的渲染。相应的, 在需要显示动态变化的连续多帧图像的场景中, 应用 1 则需要连续渲染多帧图像, 相应的, 应用 1 需要连续多次调用 SF 的 requestNextVsync 接口来请求 vSync 信号。以图 1 为例, 应用 1 为 Launcher, Launcher 5 在需要渲染桌面 103 对应的第 1 帧动效时, 需要调用 requestNextVsync 接口第一次请求 vSync 信号; Launcher 在需要渲染桌面 104 对应的第 2 帧动效时, 需要调用 requestNextVsync 接口第二次请求 vSync 信号……Launcher 在需要渲染桌面 107 对应的第 5 帧动效时, 需要调用 requestNextVsync 接口第五次请求 vSync 信号。

然而, 上述应用 1 调用 SF 中的 requestNextVsync 接口, 属于不同进程之间的跨进 10 程调用, 需要 SF 中的 Binder 线程来实现该调用。Binder 线程的优先级较低, 在电子设备的负载较高时, 如 CPU 占用高达 97% 及以上时, 电子设备通常无法优先为 Binder 线程分配 CPU 资源用于处理 requestNextVsync 接口的调用, 则 Binder 线程只能一直等待 CPU 资源。从而导致 Binder 线程会长时间处于等待资源 (即 Runnable) 的状态, 无法及时处理 requestNextVsync 接口的调用。那么, SF 也不会及时接收到应用 1 对 15 vSync 信号的请求。后续, 当 SF 接收到 vSync 信号后, 则不会分发给应用 1, 无法触发应用 1 渲染图像。最终, 显示屏中则只能显示历史的图像, 而无法显示新一帧图像。尤其在需要显示动态变化的连续多帧图像的场景中, 应用 1 需要渲染连续多帧图像, 则需要频繁使用 Binder 线程来处理 requestNextVsync 接口的调用, 因此更有可能存在上述问题。

并且, 在需要显示动态变化的连续多帧图像的场景中, 若无法及时显示新一帧图 20 像, 最终会导致动态变化的过程出现显示卡顿、不流畅的现象。如图 4 所示, 在手机的负载正常时, 可以按照前述图 3 所示的流程来显示图像, 显示屏中可以依次显示第 1 帧动效 (如桌面 401) 和第 2 帧动效 (如桌面 402a)。此后手机的负载增大, 无法按照前述图 3 所示的流程来显示图像, 显示屏则只能一直显示第 2 帧动效, 如桌面 402b 25 和桌面 402c。再后, 当手机的负载降下来后, 手机可以继续按照前述图 3 所示的流程来显示图像, 但是此时已经需要渲染第 5 帧动效了, Launcher 则会渲染第 5 帧动效, 最终会显示桌面 403 对应的第 5 帧动效。很显然, 显示屏连续显示了 3 帧第 2 帧动效, 出现了卡顿, 并且直接从第 2 帧动效过渡到第 5 帧动效, 动效过渡的不流畅。

基于上述问题, 本申请实施例提供了一种请求 vSync 信号的方法, 该方法可应用 30 于类似图 1 需要显示动态变化的连续多帧图像的场景, 在这些场景中, 相应的应用需要渲染连续多帧图像。参见图 5, 应用 1 在确定需要连续渲染多帧图像后, 可以调用 SF 的 setVsyncRate 接口向 SF 请求 vSync 信号 (如在图 5 中的 t1 时刻请求)。SF 在接收到应用 1 通过 setVsyncRate 接口对 vSync 信号的请求后, 周期性将 vSync 信号分发给应用 1。从而可以周期性触发应用 1 渲染图像。如此, 应用 1 可以实现渲染连续 35 多帧图像。然后, 应用 1 在确定结束渲染连续多帧图像后, 可以调用 SF 的 setVsyncRate 接口向 SF 请求取消 vSync 信号 (如在图 5 中的 t2 时刻请求)。SF 在接收到应用 1 通过 setVsyncRate 接口取消对 vSync 信号的请求后, 则结束周期性将 vSync 信号分发给应用 1。为了方便说明, 可以将 t1 时刻称为第一时刻, 将 t2 时刻称为第二时刻, 将调用 setVsyncRate 接口请求 vSync 信号的请求称为第一请求, 将调用 setVsyncRate 接口

请求取消 vSync 信号的请求称为第二请求。

综上所述，采用本申请实施例，应用 1 仅需要向 SF 请求一次 vSync 信号，则可以接收到 SF 周期性分发的 vSync 信号，从而周期性触发应用 1 开始渲染。尤其在需要显示动态变化的连续多帧图像的场景中，SF 周期性分发 vSync 信号，从而可以周期性触发应用 1 完成该连续多帧图像的渲染。无需应用 1 连续请求多次 vSync 信号，极大地减少了跨进程调用的次数，降低了对 Binder 线程的依赖。如此，可以降低动态变化的过程出现显示卡顿、不流畅的现象的可能。并且，应用 1 可以请求取消 vSync 信号，使 SF 结束向应用 1 周期性分发 vSync 信号，避免长期周期性分发造成资源浪费。

仍以图 1 所示显示桌面动效的场景为例，Launcher 在检测到用户对桌面 101 中视频应用的应用图标 102 的点击操作后，Launcher 可以确定需要渲染桌面 103-桌面 107 所示的多帧动效，即需要渲染连续多帧图像。该情况下，Launcher 可以调用 SF 的 setVsyncRate 接口向 SF 请求 vSync 信号，触发 SF 周期性的将 vSync 信号分发给 Launcher。周期性的 vSync 信号可以触发 Launcher 依次渲染桌面 103-桌面 107 所示的多帧动效。在完成多帧动效的渲染后，Launcher 可以确定结束渲染连续多帧图像。该情况下，Launcher 可以调用 SF 的 setVsyncRate 接口向 SF 请求取消 vSync 信号，触发 SF 结束周期性的将 vSync 信号分发给 Launcher，减少资源浪费。

示例性的，本申请实施例中的电子设备可以是手机、平板电脑、桌面型、膝上型、手持计算机、笔记本电脑、超级移动个人计算机 (ultra-mobile personal computer, UMPC)、上网本，以及蜂窝电话、个人数字助理 (personal digital assistant, PDA)、增强现实 (augmented reality, AR)、虚拟现实 (virtual reality, VR) 设备等支持显示功能的设备。本申请实施例对该电子设备的具体形态不作特殊限制。下文中，将主要以电子设备是手机为例，来说明本申请方案。

参见图 6，为本申请实施例提供的一种电子设备的硬件结构图。如图 6 所示，以电子设备是手机 600 为例，电子设备可以包括处理器 610，外部存储器接口 620，内部存储器 621，通用串行总线 (universal serial bus, USB) 接口 630，充电管理模块 640，电源管理模块 641，电池 642，天线 1，天线 2，移动通信模块 650，无线通信模块 660，音频模块 670，扬声器 670A，受话器 670B，麦克风 670C，耳机接口 670D，传感器模块 680，按键 690，马达 691，指示器 692，摄像头 693，显示屏 694，以及用户标识模块 (subscriber identification module, SIM) 卡接口 695 等。

可以理解的是，本实施例示意的结构并不构成对手机 600 的具体限定。在另一些实施例中，手机 600 可以包括比图示更多或更少的部件，或者组合某些部件，或者拆分某些部件，或者不同的部件布置。图示的部件可以以硬件，软件或软件和硬件的组合实现。

处理器 610 可以包括一个或多个处理单元，例如：处理器 610 可以包括应用处理器 (application processor, AP)，调制解调处理器，图形处理器 (graphics processing unit, GPU)，图像信号处理器 (image signal processor, ISP)，控制器，存储器，视频编解码器，数字信号处理器 (digital signal processor, DSP)，基带处理器，和/或神经网络处理器 (neural-network processing unit, NPU) 等。其中，不同的处理单元可以是独立的器件，也可以集成在一个或多个处理器中。

5 充电管理模块 640 用于从充电器接收充电输入。其中，充电器可以是无线充电器，也可以是有线充电器。在一些有线充电的实施例中，充电管理模块 640 可以通过 USB 接口 630 接收有线充电器的充电输入。在一些无线充电的实施例中，充电管理模块 640 可以通过手机 600 的无线充电线圈接收无线充电输入。充电管理模块 640 为电池 642 充电的同时，还可以通过电源管理模块 641 为手机 600 供电。

10 电源管理模块 641 用于连接电池 642，充电管理模块 640 与处理器 610。电源管理模块 641 接收电池 642 和/或充电管理模块 640 的输入，为处理器 610，内部存储器 621，外部存储器，显示屏 694，摄像头 693，和无线通信模块 660 等供电。电源管理模块 641 还可以用于监测电池容量，电池循环次数，电池健康状态（漏电，阻抗）等参数。在其他一些实施例中，电源管理模块 641 也可以设置于处理器 610 中。在另一些实施例中，电源管理模块 641 和充电管理模块 640 也可以设置于同一个器件中。

手机 600 的无线通信功能可以通过天线 1，天线 2，移动通信模块 650，无线通信模块 660，调制解调处理器以及基带处理器等实现。

15 显示屏 694 用于显示图像，视频等。在一些实施例中，显示屏 694 可用于显示动态变化的界面内容，如图 1 所示的桌面动效。显示屏 694 包括显示面板。显示面板可以采用液晶显示屏 (liquid crystal display, LCD)，有机发光二极管 (organic light-emitting diode, OLED)，有源矩阵有机发光二极体或主动矩阵有机发光二极体 (active-matrix organic light emitting diode, AMOLED)，柔性发光二极管 (flex light-emitting diode, FLED)，Miniled, MicroLed, Micro-oLed, 量子点发光二极管 (quantum dot light emitting diodes, QLED) 等。

20 手机 600 通过 GPU，显示屏 694，以及应用处理器等实现显示功能。GPU 为图像处理的微处理器，连接显示屏 694 和应用处理器。GPU 用于执行数学和几何计算，用于图形渲染。处理器 610 可包括一个或多个 GPU，其执行程序指令以生成或改变显示信息。

25 手机 600 可以通过 ISP，摄像头 693，视频编解码器，GPU，显示屏 694 以及应用处理器等实现拍摄功能。ISP 用于处理摄像头 693 反馈的数据。摄像头 693 用于捕获静态图像或视频。物体通过镜头生成光学图像投射到感光元件。在一些实施例中，手机 600 可以包括 1 个或 N 个摄像头 693，N 为大于 1 的正整数。

30 外部存储器接口 620 可以用于连接外部存储卡，例如 Micro SD 卡，实现扩展手机 600 的存储能力。外部存储卡通过外部存储器接口 620 与处理器 610 通信，实现数据存储功能。例如将音乐，视频等文件保存在外部存储卡中。

内部存储器 621 可以用于存储计算机可执行程序代码，所述可执行程序代码包括指令。处理器 610 通过运行存储在内部存储器 621 的指令，从而执行手机 600 的各种功能应用以及数据处理。

35 手机 600 可以通过音频模块 670，扬声器 670A，受话器 670B，麦克风 670C，耳机接口 670D，以及应用处理器等实现音频功能。例如音乐播放，录音等。

按键 690 包括电源键（也可称开机键），音量键等。按键 690 可以是机械按键。也可以是触摸式按键。手机 600 可以接收按键输入，产生与手机 600 的用户设置以及功能控制有关的键信号输入。马达 691 可以产生振动提示。马达 691 可以用于来电振

动提示，也可以用于触摸振动反馈。指示器 692 可以是指示灯，可以用于指示充电状态，电量变化，也可以用于指示消息，未接来电，通知等。SIM 卡接口 695 用于连接 SIM 卡。

5 本申请实施例中，手机 600 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。下文实施例，将主要以分层架构的 Android 系统为例，示例性说明手机 600 的软件架构。

10 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，参见图 7，可以将 Android 系统分为六层，从上至下分别为应用程序层 (applications)、框架层 (framework)、Java 本地接口层 (Java Native Interface, JNI)、本地层 (Native)、硬件抽象层 (HAL) 以及内核层 (kernel)。

15 其中，应用程序层中可以安装通话、备忘录、桌面、浏览器、联系人、相机、图库、日历等应用。图 7 中仅示出了桌面应用 (Launcher)。这些应用基本都需要通过手机 600 的显示屏来显示其应用界面。例如，Launcher 需要通过显示屏来显示图 1 所示桌面 101，桌面 103-桌面 107 上的应用图标，以及显示图 1 所示桌面 103-桌面 107 对应的动效。

20 框架层为应用程序层提供应用编程接口 (application programming interface, API) 和编程框架。框架层中包括显示屏渲染模块 (Choreographer) 以及标记模块

(AnimationSmooth)。AnimationSmooth 可用于应用保存渲染连续多帧图像的开关状态，开关状态包括开启状态和关闭状态。开启状态指示需要渲染连续多帧图像，关闭状态指示结束渲染连续多帧图像。

25 Choreographer 可用于应用向 SF 请求 vSync 信号。也就是说，应用 1 需要通过 Choreographer 实现向 SF 请求 vSync 信号。那么，本文中应用 1 调用 requestNextVsync 接口或者 setVsyncRate 接口向 SF 请求 vSync 信号，实质是 Choreographer 调用 requestNextVsync 接口或者 setVsyncRate 接口向 SF 请求 vSync 信号；以及，应用 1 调用 setVsyncRate 接口向 SF 请求取消 vSync 信号，实质是 Choreographer 调用 setVsyncRate 接口向 SF 请求取消 vSync 信号。

30 并且，Choreographer 可用于应用实现渲染。也就是说，应用 1 需要通过 Choreographer 实现渲染。那么，本文中 SF 向应用 1 分发 vSync 信号，触发应用 1 渲染，实质是 SF 向 Choreographer 分发 vSync 信号，触发 Choreographer 渲染。

本地层为上层 (如框架层) 提供各种服务。本地层中包括 SF，SF 可用于合成和送显，以及刷新率控制。其中，刷新率控制即：通过向应用分发 vSync 信号，控制应用在显示屏刷新一次后开始渲染。本地层中包括 SF 的 requestNextVsync 接口和 setVsyncRate 接口，这两个接口都可以实现请求 vSync 信号。其中，requestNextVsync 接口可以实现每请求一次 vSync 信号，SF 则分发一次 vSync 信号。setVsyncRate 接口可以实现每请求一次 vSync 信号，SF 则周期性分发 vSync 信号。

需要在此说明的是，框架层通常是采用 Java 语言编写的，因此，框架层也可以称为 Java 层。本地层是采用 C++/C 语言编写的，因此，本地层也可以称为 C++/C 层。Java 本地接口层则是连接框架层和本地层之间的桥梁，用于打通框架层和本地层，使框架层和本地层之间可以相互访问。

Java 本地接口层中包括对 Java 开放的接口，以便框架层通过该接口调用本地层提供的服务。本申请实施例中，需要调用 SF 的 setVsyncRate 接口向 SF 请求 vSync 信号，而 SF 中的 setVsyncRate 接口只是本地层中的接口，其并没有对 Java 开放接口。基于此，在一些实施例中，需要在框架层和 Java 本地接口层中分别新增一个接口，如图 7 中的第一接口和第二接口。通过第一接口、第二接口和 setVsyncRate 接口，打通框架层中的 Choreographer 到本地层中的 SF 之间的通路，使得应用可以调用 setVsyncRate 接口，实现向 SF 请求 vSync 信号。

硬件抽象层对底层硬件驱动进行了一层封装，并向上层提供调用驱动的通用接口，以供上层调用驱动从而驱动相应的硬件工作。硬件抽象层中包括硬件混合渲染器 (Hwcomposer, HWC)。HWC 可以和 SF 结合，完成合成处理。其中，SF 可以使用 OpenGL ES 合成图层，但是该合成需要占用并消耗 GPU 资源。而 HWC 通过硬件设备进行图层合成，可以减轻 GPU 的合成压力。

内核层中包括驱动硬件工作的驱动，如显示驱动 (Display drivers)。显示驱动用于驱动显示屏工作，并且将来上层的 (如硬件抽象层) 的图像数据流 (如 RGB 图像数据流) 传输给显示屏显示 (即送显)。

本申请实施例提供的请求 vSync 信号的方法，可以在具有上述硬件结构和软件结构的手机 600 中实现。下面先结合上述手机 600 的软件架构，对本申请实施例提供的请求 vSync 信号的方法做简单介绍：

应用 1 在确定需要渲染连续多帧图像后，可以通过 Choreographer，依次调用框架层中的第一接口、Java 本地接口层中的第二接口以及本地层中的 setVsyncRate 接口，向 SF 请求 vSync 信号。SF 在接收到 Choreographer 调用 setVsyncRate 接口对 vSync 信号的请求后，周期性将 vSync 信号分发给 Choreographer，从而可以周期性触发 Choreographer 渲染应用 1 的图像。

然后，应用 1 在确定结束渲染连续多帧图像后，可以通过 Choreographer，依次调用第一接口、第二接口以及 setVsyncRate 接口，向 SF 请求取消 vSync 信号。SF 在接收到 Choreographer 调用 setVsyncRate 接口取消对 vSync 信号的请求后，则结束周期性将 vSync 信号分发给 Choreographer。从而不会继续周期性触发 Choreographer 渲染应用 1 的图像。

其中，应用 1 可基于自身的显示逻辑确定是否需要渲染连续多帧图像。

示例一，以前文图 1 所示的场景为例，Launcher 检测到用户对桌面上某应用图标 (如视频应用的应用图标 102) 的点击操作后，其显示逻辑为：显示从应用图标放大到应用界面的桌面动效。该桌面动效需要 Launcher 显示动态变化的连续多个动效帧，即多帧图像来实现。那么，Launcher 检测到用户对桌面上某应用图标的点击操作后，则可以确定需要渲染连续多帧图像。

示例二，仍以 Launcher 为例，Launcher 检测到用户在桌面上的滑动 (如向左或向右滑动) 操作后，随着用户的滑动操作，桌面也会随之滑动。并且，在用户的滑动操作结束后，Launcher 的显示逻辑为：继续滑动桌面直至完全显示桌面的某一个屏 (如返回滑动操作前显示的屏，或者切换到上一屏，或者切换到下一屏)。上述继续滑动桌面直至完全显示桌面的某一个屏的过程需要 Launcher 显示动态变化的连续多帧图像

来实现。那么，Launcher 检测到用户在桌面上的滑动操作结束后，则可以确定需要渲染连续多帧图像。

5 示例三，以日历应用为例，日历应用在检测到用户对月份的视图中某天的点击操作后，其显示逻辑为：显示从月份的视图变化到某天的视图的过渡动效，最终显示某天的视图。该过渡动效需要日历应用显示动态变化的连续多帧图像来实现。那么，日历应用检测到用户对月份的视图中某天的点击操作后，则可以确定需要渲染连续多帧图像。

也就是说，若在用户的触摸操作结束后，应用 1 需要在短时间内显示动态变化的连续多帧图像，如应用 1 需要以小于 20ms 的间隔显示滑动、缩放等动效，则应用 1
10 可以确定需要渲染连续多帧图像。而在完成该多帧图像的渲染后，或者，在结束该多帧图像的显示前、检测的到退出在前台运行应用 1 后，或者，在结束该多帧图像的显示前、检测到显示屏息屏，则应用 1 可以确定结束渲染连续多帧图像。

通常情况下，应用 1 在每次需要渲染新一帧图像之前，都需要通过 Choreographer 来请求 vSync 信号，以请求 SF 分发 vSync 信号。并且，Choreographer 在接收到 SF 分发的 vSync 信号后，才会渲染应用 1 的图像。而在本申请实施例中，应用 1 仅需在确定需要渲染连续多帧图像后，通过 Choreographer 来请求一次 vSync 信号；以及，在确定结束渲染连续多帧图像后，通过 Choreographer 来请求取消 vSync 信号。也就是说，尽管需要渲染连续多帧图像，也仅需通过 Choreographer 来请求一次 vSync 信号，而无需在每次渲染一帧图像之前，均通过 Choreographer 来请求 vSync 信号。
15

20 参见图 8，在一些实施例中，在需要渲染连续多帧图像的情况下，仅通过 Choreographer 来请求一次 vSync 信号的具体实现包括：

S801、应用 1 确定需要渲染连续多帧图像。

S802、在需要渲染多帧图像中的第一帧图像（也可以称为第一图像）时，应用 1 向 Choreographer 请求 vSync 信号。

25 应用 1 在需要渲染任一帧图像时，首先向 Choreographer 请求 vSync 信号，以请求渲染该任一帧图像（如第一图像）。从而触发 Choreographer 进一步向 SF 请求 vSync 信号。示例性的，应用 1 可以调用 Choreographer 的 scheduleVsyncLocked 接口，触发 Choreographer 向 SF 请求 vSync 信号。

30 为了方便说明，可以将应用 1 向 Choreographer 发送的、用于请求 vSync 信号的消息称为第一消息。

S803、Choreographer 调用 setVsyncRate 接口向 SF 请求 vSync 信号。

也就是说，Choreographer 在应用 1 确定渲染多帧图像后，首次接收到应用 1 对 vSync 信号的请求时，会调用 setVsyncRate 接口向 SF 请求 vSync 信号。即，在请求渲染多帧图像中的第一帧图像时，调用 setVsyncRate 接口向 SF 请求 vSync 信号，则图 5
35 中的 t1 时刻可以是请求渲染多帧图像中的第一帧图像的时刻。

示例性的，Choreographer 调用第一接口、第二接口以及 setVsyncRate 接口时携带的参数为第一参数，用于指示请求 vSync 信号。例如，第一参数为 1，则调用 setVsyncRate 接口具体为 setVsyncRate (1)，表示调用 setVsyncRate 接口请求 vSync 信号。

S804、SF 更新请求状态为周期性请求，SF 中请求状态的初始值为未请求。

在本实施例中，请求状态进一步包括周期性请求。请求状态为周期性请求，可以指示 SF 将 vSync 信号周期性的分发给 Choreographer。

示例性的，请求状态及其状态转移过程如图 9 所示，请求状态包括未请求（如 none）、已请求（如 single）以及周期性请求（如 period）。请求状态为未请求，则指示无需分发 vSync 信号给应用 1。请求状态为已请求，指示需要分发一次 vSync 信号给应用 1。请求状态为周期性请求，指示需要周期性分发 vSync 信号给应用 1。请求状态的初始值为未请求，在请求状态为未请求的情况下，Choreographer 调用 requestNextVsync 接口向 SF 请求 vSync 信号，则请求状态会更新为已请求（可参见前文图 5 中的相关说明）。在请求状态为已请求的情况下，SF 分发一次 vSync 信号给应用 1 后，则请求状态会更新为未请求（可参见前文图 5 中的相关说明）。在请求状态为未请求的情况下，Choreographer 调用 setVsyncRate 接口向 SF 请求 vSync 信号（如 setVsyncRate(1)），则请求状态会更新为周期性请求，如 S803-S804 所述。

需要在此说明的是，本文中的请求状态都是针对应用 1 而言的，实际上，SF 中维护有各个应用的请求状态。每个应用的请求状态可以指示相应的应用对 vSync 信号的需求情况。

本实施例中，在需要渲染多帧图像中的第一帧图像时，则调用 setVsyncRate 接口向 SF 请求 vSync 信号，从而可以在确定需要渲染连续多帧图像后，及时使 SF 将请求状态更新为周期性请求。后续，SF 则可以周期性分发 vSync 信号，触发依次完成多帧图像的渲染。

S805、显示屏向 SF 发送 vSync 信号。

显示屏完成一次刷新，则会向 SF 发送一次 vSync 信号。

S806、在请求状态为周期性请求的情况下，SF 向 Choreographer 分发 vSync 信号。

需要在此说明的是，继续参见图 9，在请求状态为周期性请求的情况下，SF 向 Choreographer 分发 vSync 信号后，并不会将请求状态更新为未请求，而是会保持周期性请求不变。如此，SF 后续接收到 vSync 信号后，才会继续分发给 Choreographer，从而实现周期性分发。

Choreographer 在接收到 vSync 信号后，则可以渲染多帧图像中的第一帧图像，然后发送给 SF 进行合成，最后送显，此处不再赘述。

S807、在需要渲染多帧图像中的第二帧图像时，应用 1 向 Choreographer 请求 vSync 信号。

经过前述 S802-S804，SF 已经将请求状态更新为周期性请求了，后续，SF 则可以周期性分发 vSync 信号，触发依次完成多帧图像的渲染。因此，在请求渲染多帧图像中除第一帧图像之外的图像（也可以称为第二图像）时，即使应用 1 会向 Choreographer 请求 vSync 信号，Choreographer 也无需再重复向 SF 请求 vSync 信号，如调用 setVsyncRate 接口向 SF 请求 vSync 信号。如此，可以实现在需要渲染连续多帧图像的情况下，仅通过 Choreographer 来请求一次 vSync 信号。

为了方便说明，可以将请求渲染多帧图像中除第一帧图像之外的任一帧图像的时刻称为第三时刻。

S808、显示屏向 SF 发送 vSync 信号。

显示屏再完成一次刷新，则可以显示 S806 之后 Choreographer 渲染，然后发送给 SF 进行合成，最后送显的图像，并且向 SF 发送 vSync 信号。

S809、在请求状态为周期性请求的情况下，向 Choreographer 分发 vSync 信号。

5 Choreographer 在接收到 vSync 信号后，则可以渲染多帧图像中的第二帧图像，然后发送给 SF 进行合成，最后送显，此处不再赘述。

之后，只要请求状态保持为周期性请求，显示屏向 SF 发送 vSync 信号后，SF 还会继续向 Choreographer 分发 vSync 信号，Choreographer 则可以继续渲染多帧图像中的第三帧图像、第四帧图像……。

S810、应用 1 确定结束渲染连续多帧图像。

10 S811、在需要渲染多帧图像后的第一帧图像时，应用 1 向 Choreographer 请求 vSync 信号。

多帧图像渲染完成后，当需要渲染新一帧图像（即多帧图像后的第一帧图像，也可以称为第三图像）时，应用 1 同样会向 Choreographer 请求 vSync 信号，以请求渲染该新一帧图像。

15 S812、Choreographer 调用 setVsyncRate 接口向 SF 请求取消 vSync 信号。

本实施例中，在完成多帧图像的渲染后，Choreographer 首次接收到应用 1 对 vSync 信号的请求时，首先会调用 setVsyncRate 接口，向 SF 请求取消 vSync 信号。即，在请求渲染多帧图像后的第一帧图像时，调用 setVsyncRate 接口向 SF 请求取消 vSync 信号，则图 5 中的 t2 时刻可以是请求渲染多帧图像后的第一帧图像的时刻。

20 示例性的，Choreographer 调用第一接口、第二接口以及 setVsyncRate 接口时携带的参数均为第二参数，用于指示请求取消 vSync 信号。例如，第二参数为 0，则调用 setVsyncRate 接口具体为 setVsyncRate(0)，表示调用 setVsyncRate 接口请求取消 vSync 信号。

S813、SF 更新请求状态为未请求。

25 继续参见图 9，在请求状态为周期性请求的情况下，Choreographer 调用 setVsyncRate 接口向 SF 请求取消 vSync 信号（如 setVsyncRate (0) ），则请求状态会更新为未请求。

在请求状态更新为未请求之后，后续 SF 接收到 vSync 信号后，则不会继续周期性分发 vSync 信号。

30 本实施例中，在多帧图像渲染完成后，需要渲染新一帧图像时，则调用 setVsyncRate 接口向 SF 请求取消 vSync 信号，从而可以在多帧图像渲染完成后，及时使 SF 将请求状态更新为未请求。后续，SF 则不会周期性分发 vSync 信号，可以节省资源。

上述实施例中，主要以周期为 1 来说明。即：Choreographer 在调用 setVsyncRate 35 接口向 SF 请求 vSync 信号时，携带指示周期为 1 的参数，如 setVsyncRate (1) ，括号中的 1 指示周期为 1；然后，SF 每次接收 vSync 信号后，都会向 Choreographer 分发 vSync 信号。当然，周期也可以为 2,3,4……。Choreographer 在调用 setVsyncRate 接口向 SF 请求 vSync 信号时，携带指示周期 n 的参数，如 setVsyncRate (n) ，括号中的 n 指示周期为 n。周期为 2 时，SF 每收到 2 次 vSync 信号后，SF 则向 Choreographer

分发一次 vSync。周期为 3 时，SF 每收到 3 次 vSync 信号后，SF 则向 Choreographer 分发一次 vSync……。

在手机 600 的使用过程中，存在大量无需渲染连续多帧图像的场景。以时钟应用中的计时功能为例，其通常以秒 (s) 为单位更新倒计时，例如，倒计时 15 分钟，当前显示的是 15:00，则下一次是过 1s 后显示 14:59，而且 1s 完全足够用户执行退出在前台运行时钟应用的操作，或者足够用户执行息屏的操作。也就是说，时钟应用难以预估接下来是否需要继续显示倒计时的图像，如接着显示 14:59，14:58，14:57……对应的图像。很显然，时钟应用的计时功能不属于需要渲染连续多帧图像的场景。在这些场景中，应用通常仅需在满足一定的条件时，渲染一帧图像即可。示例性的，上述时钟应用的计时功能为例，时钟应用仅需每隔 1s 渲染一帧新的倒计时图像即可。又示例性的，以计算器应用为例，计算器应用在检测到用户对计算器的键盘中数字或者符号的一次点击操作，才需渲染一帧新的图像。

也就是说，在这些场景中，若采用前述周期性分发的方式，应用在每次渲染新一帧图像之前，需要调用 setVsyncRate 接口向 SF 请求 vSync 信号，以触发 SF 将来自显示屏的 vSync 信号分发给应用；并且，在该新一帧图像渲染完成后，应用则需要调用 setVsyncRate 接口向 SF 请求取消 vSync 信号，以触发 SF 结束将来自显示屏的 vSync 信号分发给应用。即，一帧图像的渲染，需要调用两次 setVsyncRate 接口。这显然会导致资源浪费。

基于此，在一些实施例中，在需要渲染连续多帧图像的场景中，调用 setVsyncRate 接口向 SF 请求 vSync 信号，在不需要渲染连续多帧图像的场景中，调用 requestNextVsync 接口向 SF 请求 vSync 信号。

在本实施例中，应用 1 可以记录是否需要渲染连续多帧图像的开关状态（也可以称为第一开关状态）。开关状态包括开启状态（如为 true）和关闭状态（如为 false）。其中，开关状态的初始值为关闭状态。开关状态为关闭状态，则指示不需要渲染连续多帧图像。开关状态为开启状态，则指示需要渲染连续多帧图像。应用 1 在需要渲染任一帧图像时，向 Choreographer 请求 vSync 信号后，Choreographer 可先查询开关状态，并依据开关状态调用 requestNextVsync 接口或者 setVsyncRate 接口向 SF 请求 vSync 信号。

参见图 10，以应用 1 将开关状态存储在前述图 7 所示框架层中的 AnimationSmooth 为例，本实施例的请求 vSync 信号的方法在图 8 的 S801 之前，进一步包括 S1001-S1007：

S1001、在确定渲染连续多帧图像之前、需要渲染图像时，应用 1 向 Choreographer 请求 vSync 信号。

S1002、Choreographer 从 AnimationSmooth 中查询开关状态，AnimationSmooth 中开关状态的初始值为关闭状态。

S1003、Choreographer 查询到的开关状态为关闭状态，调用 requestNextVsync 接口向 SF 请求 vSync 信号。

本实施例中，Choreographer 在请求 vSync 信号之前，先查询 AnimationSmooth 中的开关状态。查询到为关闭状态，则表示当前没有渲染连续多帧图像的需求，因此，Choreographer 可以调用 requestNextVsync 接口来请求 vSync 信号。从而实现请求一次

vSync 信号，分发一次 vSync 信号。

S1004、SF 更新请求状态为已请求，SF 中应用 1 的请求状态的初始值为未请求。

S1005、显示屏向 SF 发送 vSync 信号。

S1006、在请求状态为已请求的情况下，SF 向 Choreographer 分发 vSync 信号。

5 Choreographer 在接收到 vSync 信号之后，则可以开始渲染，然后发送给 SF 进行合成，最终送显。

S1007、SF 更新请求状态为未请求。

10 经过上述 S1001-S1007，可以在需要渲染连续多帧图像之前，针对一帧图像，仅需调用 requestNextVsync 接口向 SF 请求一次 vSync 信号，SF 则可分发一次 vSync 信号以触发 Choreographer 完成渲染，而无需请求一次 vSync 信号和请求取消 vSync 信号，从而可以减少跨进程通信，节省资源。

需要说明的是，前述 S1001-S1007 仅说明了在确定渲染连续多帧图像之前，渲染一帧图像的处理过程，实际中，在确定需要渲染连续多帧图像之前，可能有至少两帧图像需要渲染，每帧图像都可以采用上述 S1001-S1007 的过程来处理。

15 继续参见图 10，在应用 1 确定需要渲染连续多帧图像之后，与前述图 8 所示的实施例不同的是：在 S801 之后，还包括 S1008-S1009；在应用 1 向 Choreographer 请求 vSync 信号之后，还包括查询 AnimationSmooth 中的开关状态的步骤，如 S1010、S1011；以及 S803 具体为 S803a：

S801、应用 1 确定需要渲染连续多帧图像。

20 S1008、应用 1 触发 AnimationSmooth 更新开关状态。

S1009、AnimationSmooth 更新开关状态为开启状态。

在本实施例中，应用 1 在确定需要渲染连续多帧图像时，首先需要将 AnimationSmooth 中的开关状态更新为开启状态，以指示需要渲染连续多帧图像。

25 S802、在需要渲染多帧图像中的第一帧图像时，应用 1 向 Choreographer 请求 vSync 信号。

S1010、Choreographer 从 AnimationSmooth 中查询开关状态。

S803a、Choreographer 查询到开关状态为开启状态，调用 setVsyncRate 接口请求 vSync 信号。

30 本实施例中，Choreographer 在请求 vSync 信号之前，先查询 AnimationSmooth 中的开关状态。查询到为开启状态，则表示当前有渲染连续多帧图像的需求，因此，Choreographer 可以调用 setVsyncRate 接口来请求 vSync 信号。

S804、SF 更新请求状态为周期性请求。

S805、显示屏向 SF 发送 vSync 信号。

S806、在请求状态为周期性请求的情况下，SF 向 Choreographer 分发 vSync 信号。

35 S807、在需要渲染多帧图像中的第二帧图像时，应用 1 向 Choreographer 请求 vSync 信号。

S1011、Choreographer 从 AnimationSmooth 中查询开关状态。

同样的，Choreographer 在请求 vSync 信号之前，先查询 AnimationSmooth 中的开关状态。查询到为开启状态，则表示当前有渲染连续多帧图像的需求。但是，经过前

述 S802、S1010、S803a 以及 S804，SF 已经将请求状态更新为周期性请求了，后续，SF 则可以周期性分发 vSync 信号，触发依次完成多帧图像的渲染。因此，在需要渲染多帧图像中的第二帧图像以及之后的图像时，即使应用 1 请求 vSync 信号，Choreographer 也查询到开关状态为开启状态，Choreographer 无需再重复向 SF 请求 vSync 信号，如调用 setVsyncRate 接口向 SF 请求 vSync 信号。

S808、显示屏向 SF 发送 vSync 信号。

S809、在请求状态为周期性请求的情况下，向 Choreographer 分发 vSync 信号。

继续参见图 10，在应用 1 确定结束渲染连续多帧图像之后，与前述图 8 所示的实施例不同的是：在 S810 之后，还包括 S1012-S1013；在应用 1 向 Choreographer 请求 vSync 信号之后，还包括查询 AnimationSmooth 中的开关状态的步骤，如 S1014；以及 S812 具体为 S812a：

S810、应用 1 确定结束渲染连续多帧图像。

S1012、应用 1 触发 AnimationSmooth 更新开关状态。

S1013、AnimationSmooth 更新开关状态为关闭状态。

在本实施例中，应用 1 在确定结束渲染连续多帧图像时，首先需要将 AnimationSmooth 中的开关状态更新为关闭状态，以指示结束渲染连续多帧图像。

S811、在需要渲染多帧图像后的第一帧图像时，应用 1 向 Choreographer 请求 vSync 信号。

S1014、Choreographer 从 AnimationSmooth 中查询开关状态。

S812a、Choreographer 查询到开关状态为关闭状态，并且前一次查询到为开启状态，调用 setVsyncRate 接口请求取消 vSync 信号。

Choreographer 查询到开关状态为关闭状态，则表示当前无需连续渲染多帧需求；在此基础上，若前一次查询到的为开启状态，则表示刚结束渲染连续多帧图像的需求。此时，Choreographer 则调用 setVsyncRate 接口请求取消 vSync 信号。

S813、SF 更新请求状态为未请求。

进一步的，在一些实施例中，可以在 AnimationSmooth 和 Choreographer 中分别维护开关状态，从而可以在需要渲染连续多帧图像的场景中，调用 setVsyncRate 接口向 SF 请求 vSync 信号，在不需要渲染连续多帧图像的场景中，调用 requestNextVsync 接口向 SF 请求 vSync 信号；还可以准确确定调用 setVsyncRate 接口向 SF 请求 vSync 信号和取消 vSync 信号的时机。其中，AnimationSmooth 和 Choreographer 中的开关状态的初始值均为关闭状态。为了区分第一开关状态，可以将 Choreographer 中的开关状态称为第二开关状态。

在本实施例中，Choreographer 在请求 vSync 信号之前，需要先查询 AnimationSmooth 中的开关状态，并基于查询到的开关状态和 Choreographer 本地的开关状态，确定是否向 SF 请求 vSync 信号，以及确定调用 setVsyncRate 接口或者 requestNextVsync 接口向 SF 请求 vSync 信号。

参见图 11，本实施例包括如下步骤：

S1001、在确定需要渲染连续多帧图像之前、需要渲染图像时，应用 1 向 Choreographer 请求 vSync 信号。

为了方便说明，可以将请求渲染多帧图像之前、需要渲染的任一帧图像的时刻称为第四时刻。

S1002、Choreographer 从 AnimationSmooth 中查询开关状态，AnimationSmooth 中开关状态的初始值为关闭状态。

5 继续参见图 11，前述图 10 中的 S1003 具体可以为如下 S1003a：

S1003a、Choreographer 查询到的开关状态与 Choreographer 本地的开关状态均为关闭状态，调用 requestNextVsync 接口向 SF 请求 vSync 信号。

为了方便说明，可以将调用 requestNextVsync 接口请求 vSync 信号、以渲染多帧图像前的任一帧图像的请求称为第四请求。

10 本实施例中，Choreographer 在请求 vSync 信号之前，先查询 AnimationSmooth 中的开关状态。查询到为关闭状态，则表示当前没有渲染连续多帧图像的需求，因此，Choreographer 需要调用 requestNextVsync 接口来请求 vSync 信号，从而实现请求一次 vSync 信号，分发一次 vSync 信号。并且，Choreographer 本地的开关状态为关闭状态，表示此时并不是刚结束渲染连续多帧图像的需求。因此，Choreographer 无需调用
15 requestNextVsync 接口来请求取消 vSync 信号。那么，在查询到的开关状态和本地的开关状态均为关闭状态时，则可以调用 requestNextVsync 接口来请求 vSync 信号。

S1004、SF 更新请求状态为已请求，SF 中应用 1 的请求状态的初始值为未请求。

S1005、显示屏向 SF 发送 vSync 信号。

S1006、在请求状态为已请求的情况下，SF 向 Choreographer 分发 vSync 信号。

20 S1007、SF 更新请求状态为未请求。

S801、应用 1 确定需要渲染连续多帧图像。

S1008、应用 1 触发 AnimationSmooth 更新开关状态。

S1009、AnimationSmooth 更新开关状态为开启状态。

25 S802、在需要渲染多帧图像中的第一帧图像时，应用 1 向 Choreographer 请求 vSync 信号。

S1010、Choreographer 从 AnimationSmooth 中查询开关状态。

继续参见图 11，前述图 10 中的 S803a 进一步包括如下 S1101-S1102：

S1101、Choreographer 查询到的开关状态为开启状态，Choreographer 本地的开关状态为关闭状态，更新 Choreographer 本地的开关状态为开启状态。

30 更新 Choreographer 本地的开关状态为开启状态，从而使 Choreographer 本地的开关状态与应用 1 是否需要渲染连续多帧图像的需求保持一致。例如，应用 1 需要渲染连续多帧图像，则 Choreographer 本地的开关状态更新为开启状态。

S1102、Choreographer 调用 setVsyncRate 接口向 SF 请求 vSync 信号。

35 本实施例中，Choreographer 在请求 vSync 信号之前，先查询 AnimationSmooth 中的开关状态。查询到为开启状态，并且更新前 Choreographer 本地的开关状态为关闭状态，表明应用 1 刚从不需要渲染连续多帧图像转变为需要渲染连续多帧图像，因此，Choreographer 需要调用 setVsyncRate 接口来请求 vSync 信号，及时触发 SF 周期性分发 vSync 信号。从而实现请求一次 vSync 信号，SF 周期性分发 vSync 信号。

S804、SF 更新请求状态为周期性请求。

S805、显示屏向 SF 发送 vSync 信号。

S806、在请求状态为周期性请求的情况下，SF 向 Choreographer 分发 vSync 信号。

S807、在需要渲染多帧图像中的第二帧图像时，应用 1 向 Choreographer 请求 vSync 信号。

5 S1011、Choreographer 从 AnimationSmooth 中查询开关状态。

继续参见图 11，在前述图 10 中的 S1011 之后，还包括如下 S1103：

S1103、Choreographer 查询的开关状态与本地的开关状态均为开启状态，则不继续请求 vSync 信号。

Choreographer 查询的开关状态为开启状态，表示当前有渲染连续多帧图像的需求，
10 因此，Choreographer 需要调用 setVsyncRate 接口来请求 vSync 信号。但是，查询到的开关状态与更新前 Choreographer 本地的开关状态一致，表明应用 1 并不是刚从不需要渲染连续多帧图像转变为需要渲染连续多帧图像，此前 Choreographer 应该已经调用 setVsyncRate 接口来请求过 vSync 信号（如 S1102），因此，Choreographer 不需要重复调用 setVsyncRate 接口来请求 vSync 信号。那么，在查询到的开关状态和本地的开关状态均为开启状态时，
15 则无需重复调用 requestNextVsync 接口来请求 vSync 信号。从而可以避免重复调用 requestNextVsync 接口来请求 vSync 信号，保证在需要渲染连续多帧图像的场景中，仅通过调用 requestNextVsync 接口来请求一次 vSync 信号。

S808、显示屏向 SF 发送 vSync 信号。

S809、在请求状态为周期性请求的情况下，向 Choreographer 分发 vSync 信号。

20 S810、应用 1 确定结束渲染连续多帧图像。

S1012、应用 1 触发 AnimationSmooth 更新开关状态。

S1013、AnimationSmooth 更新开关状态为关闭状态。

S811、在需要渲染多帧图像后的第一帧图像时，应用 1 向 Choreographer 请求 vSync 信号。

25 S1014、Choreographer 从 AnimationSmooth 中查询开关状态。

继续参见图 11，前述图 10 中的 S812a 进一步包括如下 S1104-S1105：

S1104、Choreographer 查询到的开关状态为关闭状态，本地的开关状态为开启状态，更新本地的开启状态为关闭状态。

更新 Choreographer 本地的开关状态为关闭状态，从而使 Choreographer 本地的开关状态与应用 1 是否需要渲染连续多帧图像的需求保持一致。例如，应用 1 不需要渲染连续多帧图像，
30 则 Choreographer 本地的开关状态更新为关闭状态。

S1105、Choreographer 调用 setVsyncRate 接口向 SF 请求取消 vSync 信号。

Choreographer 查询的开关状态为关闭状态，并且，更新前 Choreographer 本地的开关状态为开启状态，表明应用 1 刚从需要渲染连续多帧图像转变为不需要渲染连续多帧图像，
35 因此，Choreographer 需要先调用 setVsyncRate 接口来请求取消 vSync 信号，避免在结束渲染连续多帧图像的需求后，SF 仍然周期性的分发 vSync 信号。从而可以在结束渲染连续多帧图像的需求后，及时触发 SF 结束周期性发送 vSync 信号。

S813、SF 更新请求状态为未请求。

截止前述 S813，将请求状态由周期性请求更新为未请求，那么，SF 则不会继续

周期性分发 vSync 信号。但是，应用 1 当前需要渲染多帧图像之后的第一帧图像，因此，继续参见图 11，在 S813 之后，还需要执行下述 S1106-S1110，以完成多帧图像后的新一帧图像的渲染：

S1106、Choreographer 调用 requestNextVsync 接口向 SF 请求 vSync 信号。

5 由于当前已经结束渲染连续多帧图像，因此，Choreographer 需要调用 requestNextVsync 接口向 SF 请求 vSync 信号，以实现请求一次 vSync 信号，SF 则分发一次 vSync 信号。

也就是说，本实施例中，Choreographer 在调用 setVsyncRate 接口向 SF 请求取消 vSync 信号后，还需要调用 requestNextVsync 接口向 SF 请求 vSync 信号。为了方便说明，可以将调用 requestNextVsync 接口请求 vSync 信号、以渲染多帧图像后的第一帧图像的请求称为第三请求。

S1107、SF 更新请求状态为已请求。

S1108、显示屏向 SF 发送 vSync 信号。

S1109、在请求状态为已请求的情况下，SF 向 Choreographer 分发 vSync 信号。

15 SF 向 Choreographer 分发 vSync 信号，则可以触发 Choreographer 完成多帧图像后的新一帧图像的渲染，然后发送给 SF 进行合成，最终送显。

S1110、SF 更新请求状态为未请求。

至此，需要说明的是，前述图 11 的实施例中仅说明到多帧图像之后的一帧图像的处理过程，此后，若还有需要渲染的图像，则可以重复按照前述图 11 所示的流程来处理。本实施例这种不再一一赘述。

另外，前述图 11 的实施例中，针对 AnimationSmooth 中的开关状态和 Choreographer 中的开关状态的四种不同组合，Choreographer 需要分别采用对应的方式来处理，包括前述 S1003a 所示的第一种方式，S1101-S1102 所示的第二种方式，S1103 所示的第三种方式，以及 S1104-S1105、S1106 所示的第四种方式。从而可以保证准确的调用 requestNextVsync 接口或者 setVsyncRate 接口请求 vSync 信号，或者调用 setVsyncRate 接口请求取消 vSync 信号。

在另一些实施例中，与前述图 11 中 S1104-S1105、S1106 所示的第四种方式不同的是：Choreographer 在查询到的开关状态为关闭状态（即 AnimationSmooth 中开关状态为关闭状态），本地的开关状态为开启状态（即 Choreographer 中的开关状态为开启状态）的情况下，首先更新本地的开启状态为关闭状态，然后直接调用 requestNextVsync 接口向 SF 请求 vSync 信号。在本实施例中，请求状态可以在周期性请求和已请求之间转移，如图 9 中的虚线箭头所示，在请求状态为周期性请求的情况下，Choreographer 调用 requestNextVsync 接口向 SF 请求 vSync 信号，则请求状态会更新为已请求。那么，后续则可以继续执行 S1107-S1110，完成对多帧图像之后的新一帧图像的渲染。也就是说，与图 11 所示的实施例不同的是：本实施例中可以省略 S1105 和 S813，而可以通过直接调用 requestNextVsync 接口向 SF 请求 vSync 信号，使请求状态由周期性请求转变为已请求，从而可以在结束周期性分发的同时，实现请求一次 vSync 信号，分发一次 vSync 信号。

前述实施例中，都是在应用 1 确定结束渲染连续多帧图像后，SF 则结束周期性分

发 vSync 信号。而在另一些实施例中，也可以由 Choreographer 确定是否结束渲染连续多帧图像。本实施例中，应用 1 在每次需要渲染任一帧图像（也可以称为第四图像）时，可以向 Choreographer 请求 vSync 信号，此时 Choreographer 可以记录处理状态为未处理状态（也可以称为第一标记）。未处理状态指示有需要渲染的图像。然后，当
5 Choreographer 接收到来自 SF 的 vSync 信号后，则可以更新处理状态为已处理状态。已处理状态指示没有需要渲染的图像。

在本实施例中，Choreographer 在每次接收到来自 SF 的 vSync 信号后，可以判断处理状态是否为未处理状态。若处理状态为未处理状态，表明应用 1 有需要渲染的图像，则可以触发 Choreographer 渲染该图像，并更新处理状态为已处理状态。若处理状
10 态不是未处理状态，则表明应用 1 并没有需要渲染的图像。也就是说，当前接收到的 vSync 信号是多余的。应理解，若通过调用 requestNextVsync 接口向 SF 请求 vSync 信号，SF 是请求一次，分发一次 vSync 信号，不会分发多余的 vSync 信号。那么，该多余的 vSync 信号极有可能是通过调用 setVsyncRate 接口向 SFSF 请求 vSync 信号后，SF 周期性分发的。针对这种情况，Choreographer 可以确定当前已经结束渲染连续多
15 帧图像的需求，Choreographer 可以调用 setVsyncRate 接口向 SF 请求取消 vSync 信号。后续，SF 则可以结束周期性分发 vSync 信号。如此，Choreographer 可以在发现多余的 vSync 信号后，及时触发 SF 结束周期性分发 vSync 信号。

参见图 12，本实施例中，Choreographer 发现多余的 vSync 信号后，上述图 11 所示的实施例中的 S810 及其之后的步骤可以替换为图 12 中 S1201-S1213 的处理过程：

20 S1201、Choreographer 接收到 vSync 信号，但处理状态不是未处理状态，确定结束渲染连续多帧图像。

S1202、Choreographer 调用 setVsyncRate 接口向 SF 请求取消 vSync 信号。

即，在 Choreographer 不包括未处理状态、但接收到来自 SF 的 vSync 信号时，调用 setVsyncRate 接口向 SF 请求取消 vSync 信号。为了方便说明，可以将 Choreographer
25 不包括未处理状态、但接收到来自 SF 的 vSync 信号的时刻也称为第二时刻。

S1203、SF 更新请求状态为未请求。

S1204、Choreographer 触发 AnimationSmooth 更新开关状态。

S1205、AnimationSmooth 更新开关状态为关闭状态。

S1206、Choreographer 更新开关状态为关闭状态。

30 也就是说，Choreographer 在确定结束渲染连续多帧图像后：第一方面如 S1202 所示，Choreographer 需要向 SF 请求取消 vSync 信号，使 SF 结束周期性分发 vSync 信号。第二方面如 S1204 所示，Choreographer 需要触发 AnimationSmooth 更新开关状态为关闭状态，使 AnimationSmooth 中的开关状态指示应用 1 结束渲染连续多帧图像。第三方面如 S1206 所示，Choreographer 更新本地的开关状态为关闭状态，使本地的开关状态指示应用 1 结束渲染连续多帧图像。需要说明的是，上述三个方面的执行顺序并不
35 以图 12 所示为限，实际实施时，上述三个方面的先后顺序没有严格限制。

S1207、应用 1 在需要渲染多帧图像后的第一帧图像时，向 Choreographer 请求 vSync 信号。

S1208、Choreographer 查询开关状态。

S1209、Choreographer 查询到的开关状态和本地的开关状态均为关闭状态，调用 requestNextVsync 接口向 SF 请求 vsync 信号。

也就是说，本实施例中，针对多帧图像之后的第一帧图像，Choreographer 可以直接调用 requestNextVsync 接口向 SF 请求 vsync 信号，而无需先调用 setVsyncRate 接口向 SF 请求取消 vsync 信号。

S1210、SF 更新请求状态为已请求。

S1211、显示屏向 SF 发送 vsync 信号。

S1212、在请求状态为已请求的情况下，SF 向分发 ChoreographerVsync 信号。

S1213、SF 更新请求状态为未请求。

10 本申请实施例还提供一种电子设备，该电子设备可以包括：存储器和一个或多个处理器。存储器和处理器耦合。该存储器用于存储计算机程序代码，该计算机程序代码包括计算机指令。当处理器执行计算机指令时，电子设备可执行上述方法实施例中手机执行的各个功能或者步骤。

15 本申请实施例还提供一种芯片系统，如图 13 所示，该芯片系统 1300 包括至少一个处理器 1301 和至少一个接口电路 1302。处理器 1301 和接口电路 1302 可通过线路互联。例如，接口电路 1302 可用于从其它装置（例如电子设备的存储器）接收信号。又例如，接口电路 1302 可用于向其它装置（例如处理器 1301）发送信号。示例性的，接口电路 1302 可读取存储器中存储的指令，并将该指令发送给处理器 1301。当所述指令被处理器 1301 执行时，可使得电子设备执行上述实施例中的各个步骤。当然，该
20 芯片系统还可以包含其他分立器件，本申请实施例对此不作具体限定。

本实施例还提供一种计算机存储介质，该计算机存储介质中存储有计算机指令，当该计算机指令在电子设备上运行时，使得电子设备执行上述相关方法步骤实现上述实施例中的图像处理方法。

25 本实施例还提供了一种计算机程序产品，当该计算机程序产品在计算机上运行时，使得计算机执行上述相关步骤，以实现上述实施例中的图像处理方法。

另外，本申请的实施例还提供一种装置，这个装置具体可以是芯片，组件或模块，该装置可包括相连的处理器和存储器；其中，存储器用于存储计算机执行指令，当装置运行时，处理器可执行存储器存储的计算机执行指令，以使芯片执行上述各方法实施例中
30 的图像处理方法。

其中，本实施例提供的电子设备、计算机存储介质、计算机程序产品或芯片均用于执行上文所提供的对应的方法，因此，其所能达到的有益效果可参考上文所提供的对应的方法中的有益效果，此处不再赘述。

35 通过以上的实施方式的描述，所属领域的技术人员可以清楚地了解到，为描述的方便和简洁，仅以上述各功能模块的划分进行举例说明，实际应用中，可以根据需要而将上述功能分配由不同的功能模块完成，即将装置的内部结构划分成不同的功能模块，以完成以上描述的全部或者部分功能。

在本申请所提供的几个实施例中，应该理解到，所揭露的装置和方法，可以通过其它的方式实现。例如，以上所描述的装置实施例仅仅是示意性的，例如，该模块或单元的划分，仅仅为一种逻辑功能划分，实际实现时可以有另外的划分方式，例如多

个单元或组件可以结合或者可以集成到另一个装置,或一些特征可以忽略,或不执行。另一点,所显示或讨论的相互之间的耦合或直接耦合或通信连接可以是通过一些接口,装置或单元的间接耦合或通信连接,可以是电性,机械或其它的形式。

5 该作为分离部件说明的单元可以是或者也可以不是物理上分开的,作为单元显示的部件可以是一个物理单元或多个物理单元,即可以位于一个地方,或者也可以分布到多个不同地方。可以根据实际的需要选择其中的部分或者全部单元来实现本实施例方案的目的。

10 另外,在本申请各个实施例中的各功能单元可以集成在一个处理单元中,也可以是各个单元单独物理存在,也可以两个或两个以上单元集成在一个单元中。上述集成的单元既可以采用硬件的形式实现,也可以采用软件功能单元的形式实现。

15 该集成的单元如果以软件功能单元的形式实现并作为独立的产品销售或使用,可以存储在一个可读取存储介质中。基于这样的理解,本申请实施例的技术方案本质上或者说对现有技术做出贡献的部分或者该技术方案的全部或部分可以以软件产品的形式体现出来,该软件产品存储在一个存储介质中,包括若干指令用以使得一个设备(可以是单片机,芯片等)或处理器(processor)执行本申请各个实施例方法的全部或部分步骤。而前述的存储介质包括:U盘、移动硬盘、只读存储器(Read-Only Memory, ROM)、随机存取存储器(Random Access Memory, RAM)、磁碟或者光盘等各种可以存储程序代码的介质。

20 最后应说明的是,以上实施例仅用以说明本申请的技术方案而非限制,尽管参照较佳实施例对本申请进行了详细说明,本领域的普通技术人员应当理解,可以对本申请的技术方案进行修改或等同替换,而不脱离本申请技术方案的精神和范围。

权 利 要 求 书

1.一种请求 vSync 信号的方法，其特征在于，用于具有显示功能的电子设备，所述电子设备中安装有第一应用，所述方法包括：

5 在第一时刻，所述电子设备中的第一应用调用所述电子设备中窗口系统 SF 的 setVsyncRate 接口，向所述 SF 发送第一请求，所述第一请求用于请求 vSync 信号；
响应于所述第一请求，所述 SF 周期性向所述第一应用分发所述 vSync 信号；

在第二时刻，所述第一应用调用所述 setVsyncRate 接口，向所述 SF 发送第二请求，所述第二请求用于请求取消 vSync 信号，所述 SF 在接收到所述第二请求之后，结束周期性向所述第一应用分发所述 vSync 信号。

10 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：

所述第一应用确定需要渲染连续多帧图像，所述第一时刻包括请求渲染第一图像的时刻，所述第一图像为所述多帧图像中的第一帧图像。

3.根据权利要求 2 所述的方法，其特征在于，所述第一应用记录第一开关状态，所述电子设备的框架层的渲染模块中记录第二开关状态，所述第一开关状态和所述第
15 二开关状态的初始值均为关闭状态；

在所述第一应用确定需要渲染连续多帧图像之后，所述方法还包括：

所述第一应用更新所述第一开关状态为开启状态；

所述在第一时刻，所述电子设备中的第一应用调用所述电子设备中窗口系统 SF 的 setVsyncRate 接口，向所述 SF 发送第一请求，包括：

20 在所述第一时刻，所述第一应用向所述渲染模块发送第一消息，所述第一消息用于请求 vSync 信号；

响应于所述第一消息，且所述第一开关状态为所述开启状态，所述第二开关状态为所述关闭状态，所述渲染模块调用所述 setVsyncRate 接口，向所述 SF 发送第一请求。

25 4.根据权利要求 3 所述的方法，其特征在于，所述第一开关状态为所述开启状态，所述第二开关状态为所述关闭状态，所述方法还包括：

所述渲染模块更新所述第二开关状态为所述开启状态。

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：

30 在第三时刻，所述第一应用向所述渲染模块发送第一消息，所述第一消息用于请求 vSync 信号，所述第三时刻为请求渲染第二图像的时刻，所述第二图像为所述多帧图像中除所述第一帧图像之外的任一图像；

响应于所述第一消息，且所述第一开关状态为所述开启状态，所述第二开关状态为所述开启状态，所述渲染模块不向所述 SF 请求所述 vSync 信号。

6.根据权利要求 4 或 5 所述的方法，其特征在于，在所述向所述 SF 发送第一请求
35 之后，所述方法还包括：

所述第一应用确定结束渲染连续多帧图像，所述第二时刻包括请求渲染第三图像的时刻，所述第三图像为完成所述多帧图像的渲染后、所述第一应用需要渲染的第一帧图像。

7.根据权利要求 6 所述的方法，其特征在于，在所述第一应用确定结束渲染连续

多帧图像之后，所述方法还包括：

所述第一应用更新所述第一开关状态为所述关闭状态；

所述在第二时刻，所述第一应用调用所述 setVsyncRate 接口，向所述 SF 发送第二请求，包括：

5 在所述第二时刻，所述第一应用向所述渲染模块发送第一消息，所述第一消息用于请求 vSync 信号；

响应于所述第一消息，且所述第一开关状态为所述关闭状态，所述第二开关状态为所述开启状态，所述渲染模块调用所述 setVsyncRate 接口，向所述 SF 发送第二请求。

10 8.根据权利要求 7 所述的方法，其特征在于，所述第一开关状态为所述关闭状态，所述第二开关状态为所述开启状态，所述方法还包括：

所述渲染模块更新所述第二开关状态为所述关闭状态。

9.根据权利要求 7 或 8 所述的方法，其特征在于，在所述向所述 SF 发送第二请求之后，所述方法还包括：

15 所述第一应用调用所述 SF 的 requestNextVsync 接口，向所述 SF 发送第三请求，所述第三请求用于请求 vSync 信号；

响应于所述第三请求，所述 SF 向所述第一应用分发一次所述 vSync 信号。

10.根据权利要求 4 或 5 所述的方法，其特征在于，所述 SF 周期性向所述第一应用分发所述 vSync 信号，包括：

20 所述 SF 周期性向所述渲染模块分发所述 vSync 信号。

11.根据权利要求 10 所述的方法，其特征在于，所述方法还包括：

在第一应用需要渲染第四图像时，向渲染模块发送所述第一消息，第四图像是第一应用需要显示的任一帧图像，所述第四图像包括所述第一图像或者所述第二图像；

25 响应于所述第一消息，渲染模块记录第一标记，并在接收到来自所述 SF 的所述 vSync 信号后，删除所述第一标记；

其中，第二时刻包括所述渲染模块在不包括所述第一标记的情况下，接收到来自所述 SF 的所述 vSync 信号的时刻。

12.根据权利要求 11 所述的方法，其特征在于，所述方法还包括：

30 在所述第二时刻，所述渲染模块通知所述第一应用将所述第一开关状态更新为所述关闭状态，所述渲染模块更新所述第二开关状态为所述关闭状态。

13.根据权利要求 3-12 中任一项所述的方法，其特征在于，所述方法还包括：

在第四时刻，所述第一应用向所述渲染模块发送第一消息，所述第一消息用于请求 vSync 信号；

35 响应于所述第一消息，且所述第一开关状态和所述第二开关状态均为所述关闭状态，所述渲染模块调用所述 requestNextVsync 接口，向所述 SF 发送第四请求，所述第四请求用于请求 vSync 信号。

14.根据权利要求 1-13 中任一项所述的方法，其特征在于，所述 SF 位于所述电子设备的本地层，所述第一应用位于所述电子设备的应用程序层，在所述向所述 SF 发送第一请求之前，所述方法还包括：

在所述电子设备的框架层中新增第一接口，在所述电子设备的 Java 本地接口层中新增第二接口，所述第一接口和所述第二接口是所述 setVsyncRate 接口对 Java 开放的接口；

所述第一应用调用所述 setVsyncRate 接口，包括：

5 所述第一应用依次经过所述第一接口和所述第二接口调用所述 setVsyncRate 接口。

15. 根据权利要求 1-14 中任一项所述的方法，其特征在于，所述第一请求中包括周期 n ， $n \geq 1$ ， n 为整数；

所述响应于所述第一请求，所述 SF 周期性向所述第一应用分发所述 vSync 信号，包括：

10 响应于所述第一请求，所述 SF 在每接收到 n 次所述 vSync 信号后，向所述第一应用分发一次所述 vSync 信号。

15 16. 一种电子设备，其特征在于，所述电子设备包括存储器和处理器，所述存储器和所述处理器耦合；其中，所述存储器中存储有计算机程序代码，所述计算机程序代码包括计算机指令，当所述计算机指令被所述处理器执行时，使得所述电子设备执行如权利要求 1-15 中任一项所述的方法。

17. 一种计算机可读存储介质，其特征在于，包括计算机指令，当所述计算机指令在电子设备上运行时，使得所述电子设备执行如权利要求 1-15 中任一项所述的方法。

20 18. 一种芯片系统，其特征在于，所述芯片系统应用于包括处理器和存储器的电子设备，所述芯片系统包括一个或多个接口电路和一个或多个处理器，所述接口电路和所述处理器通过线路互联，所述接口电路用于从所述电子设备的存储器接收信号，并向所述处理器发送所述信号，所述信号包括所述存储器中存储的计算机指令，当所述处理器执行所述计算机指令时，使得所述电子设备执行如权利要求 1-15 中任一项所述的方法。

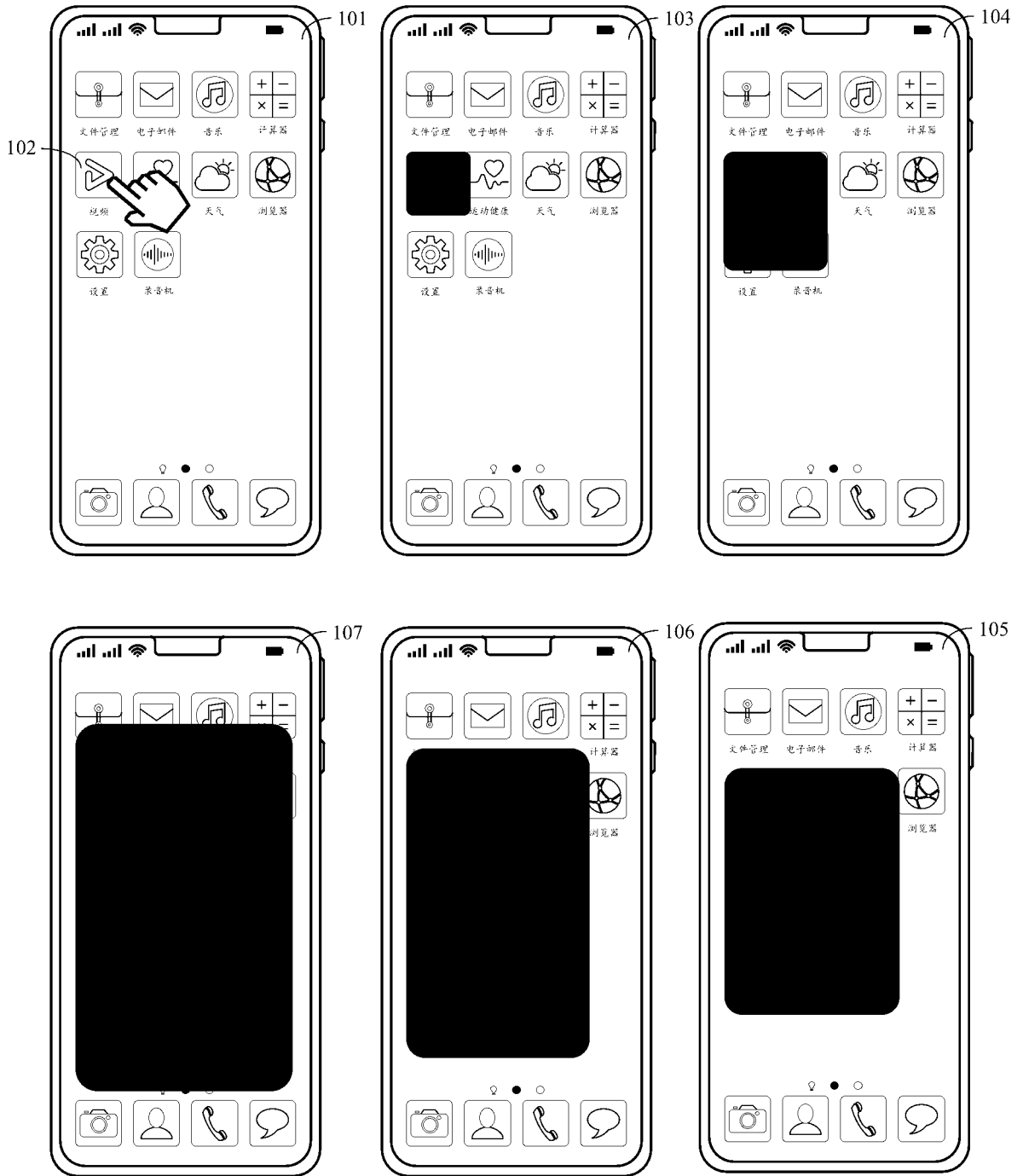


图 1

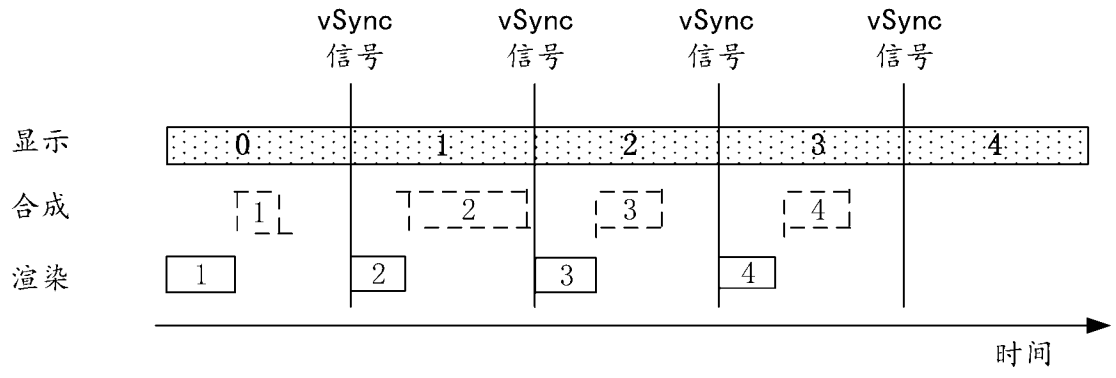


图 2

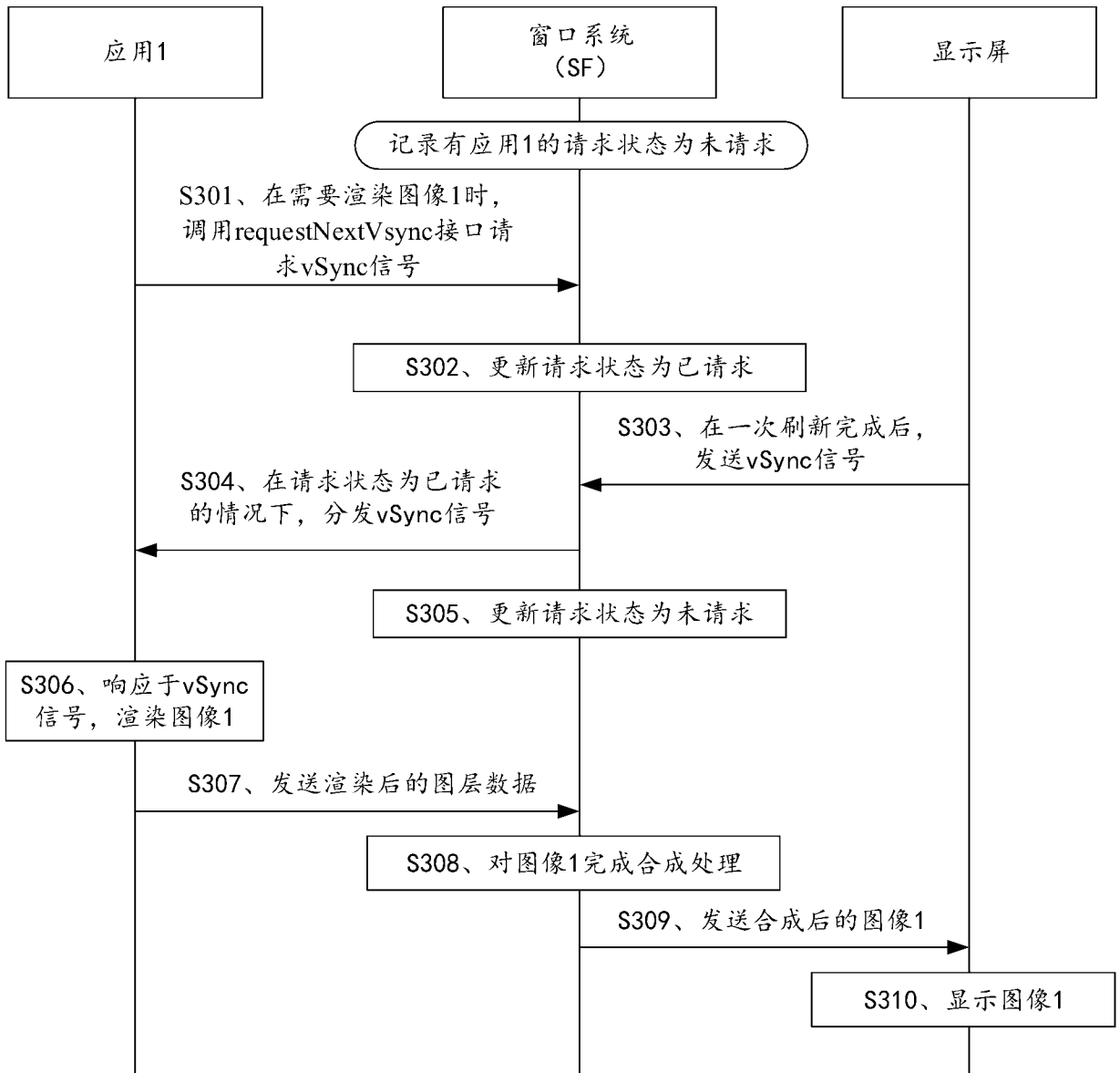


图 3

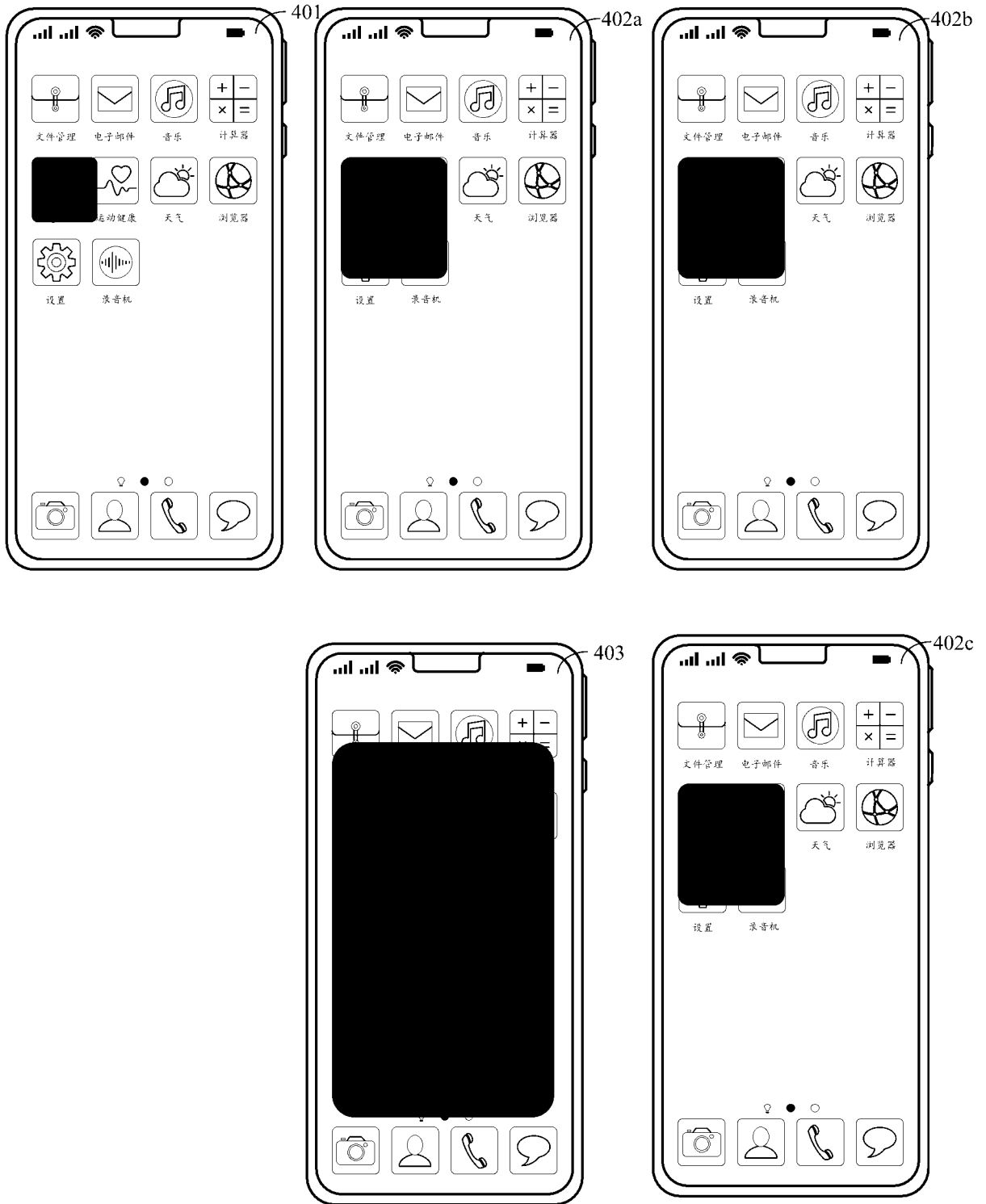
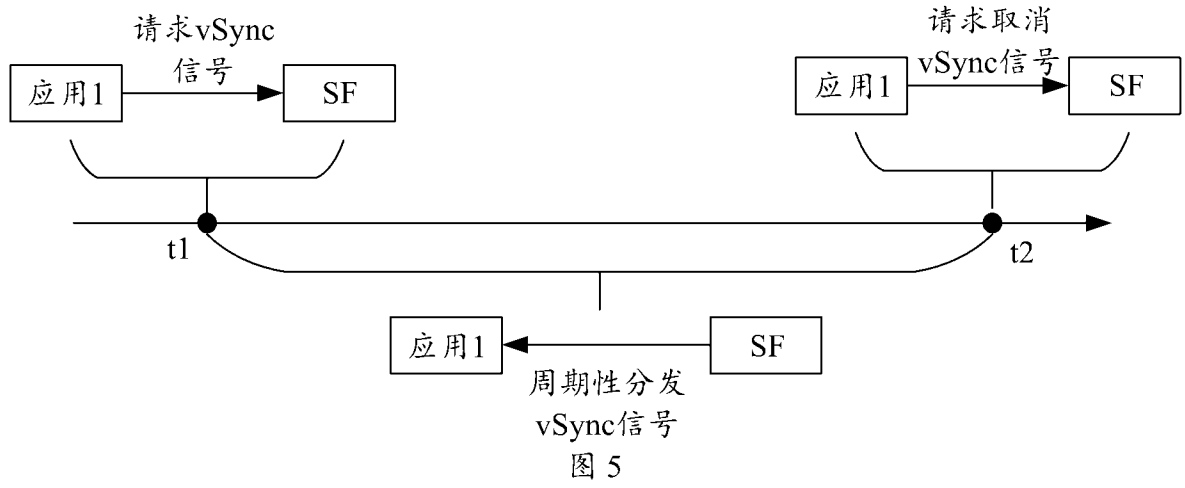


图 4



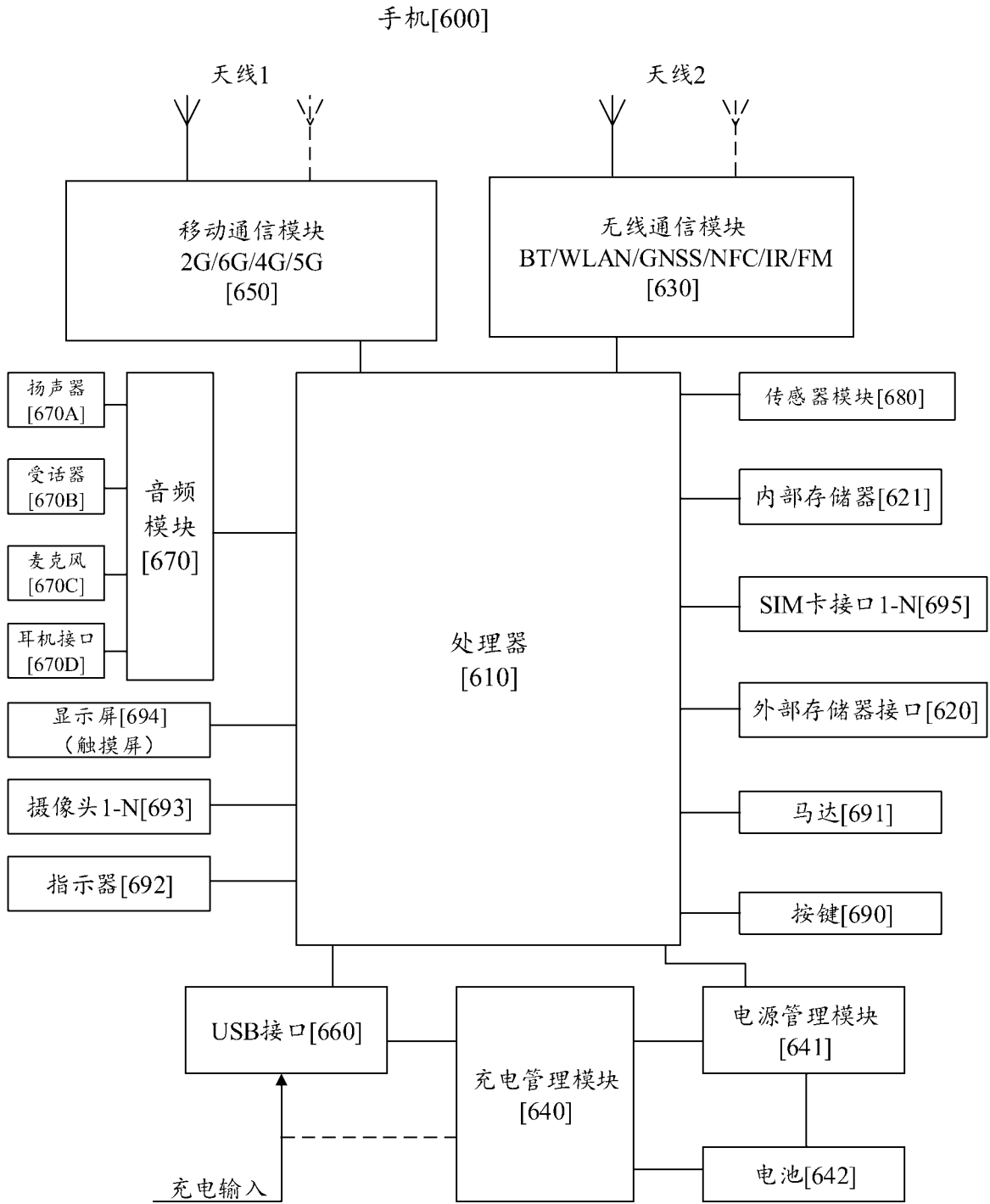


图 6

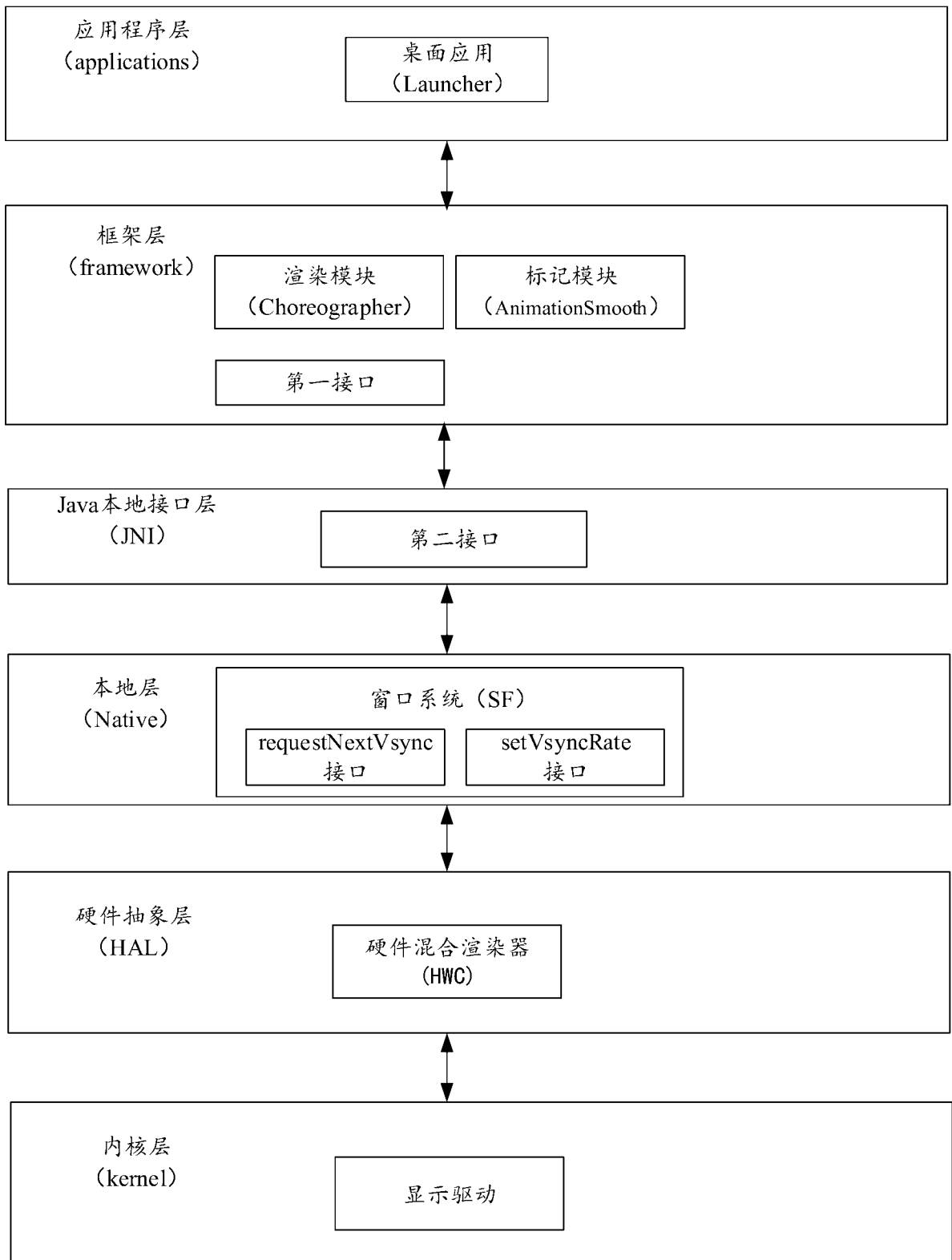


图 7

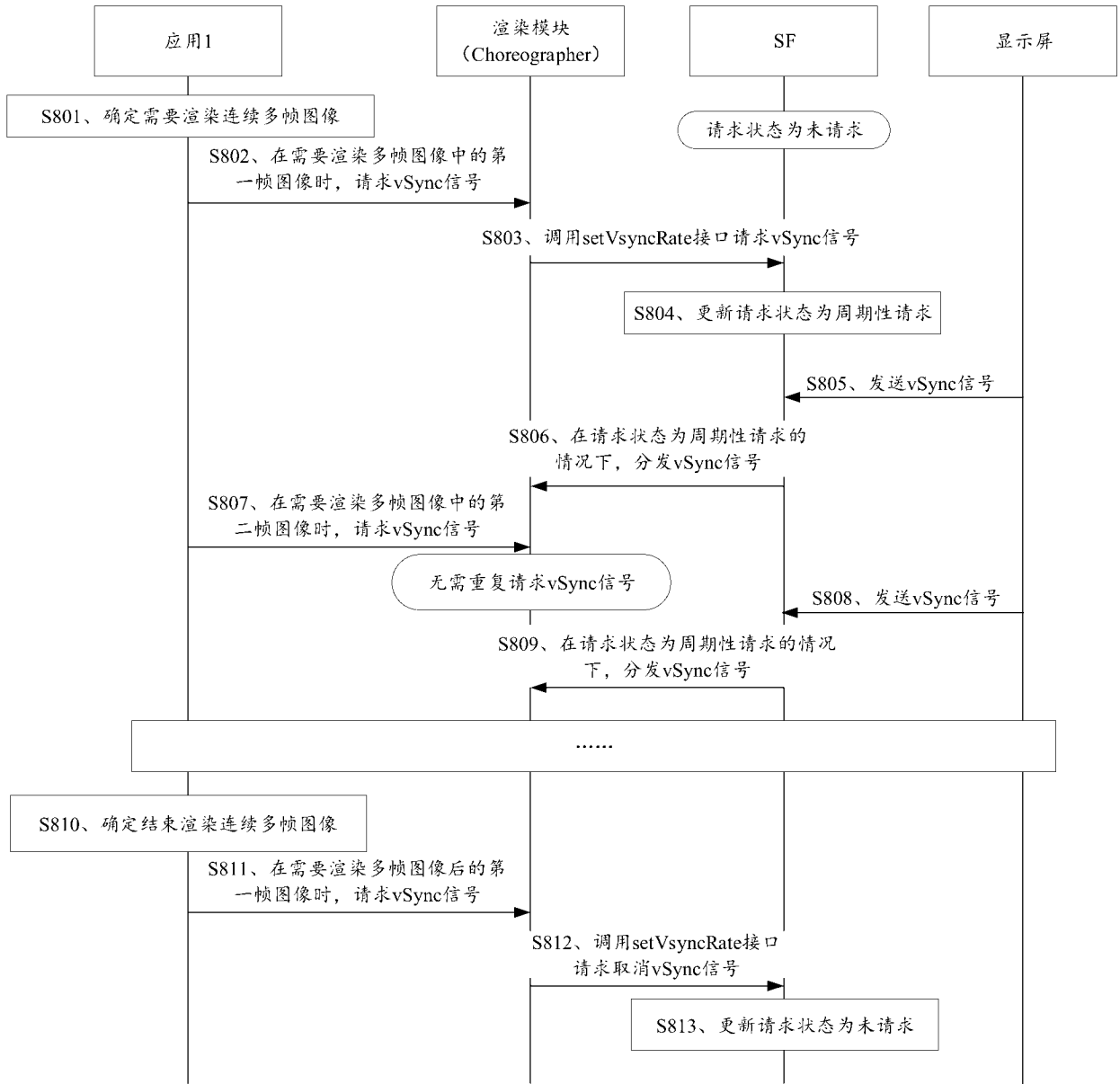


图 8

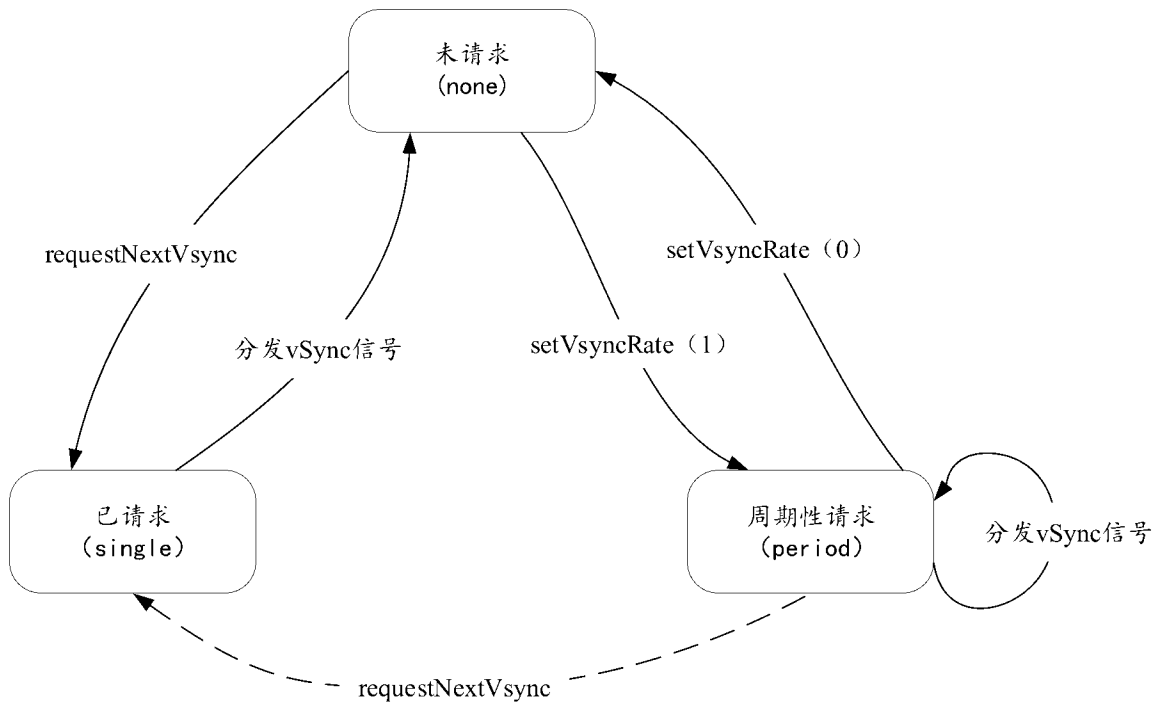


图 9

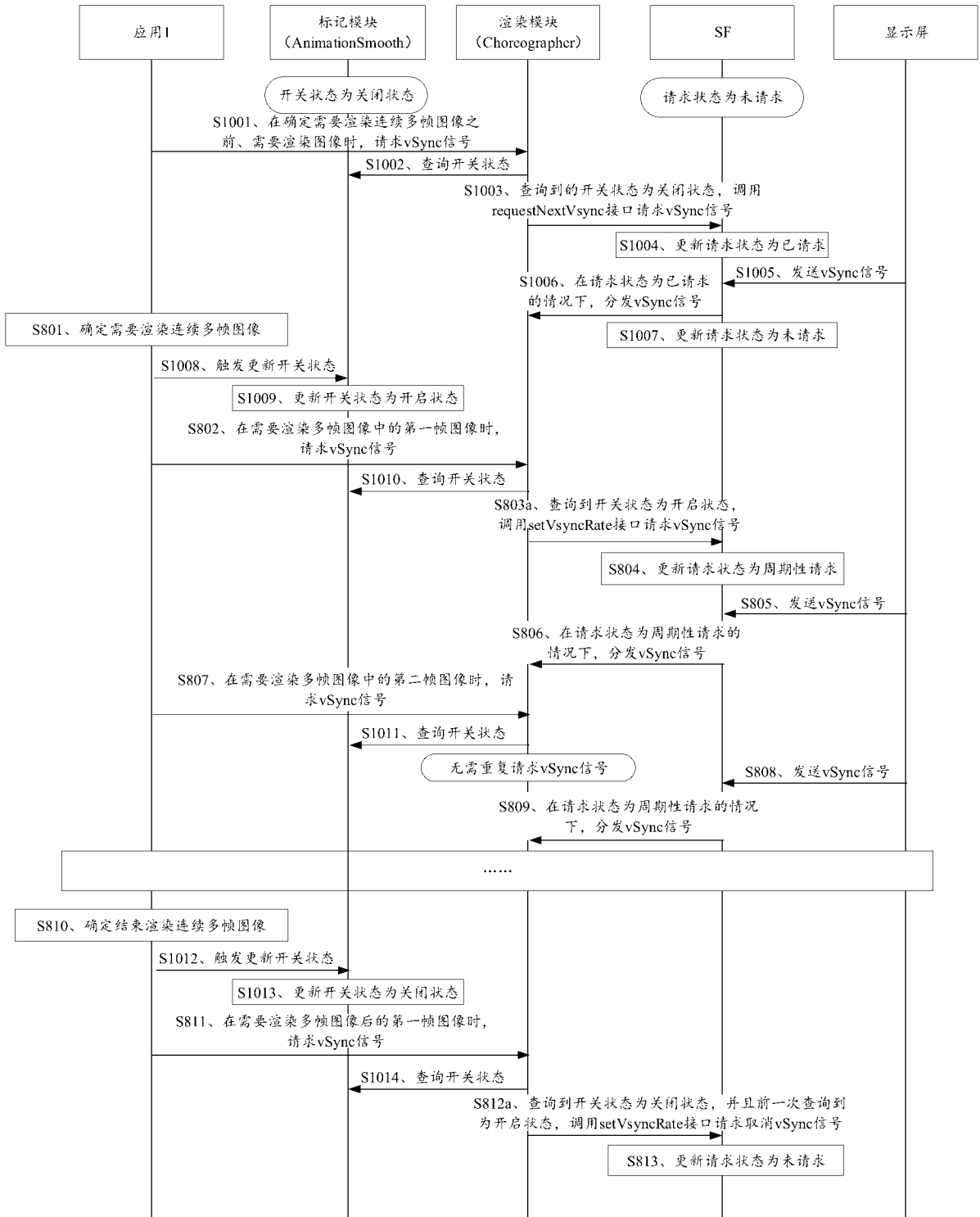


图 10

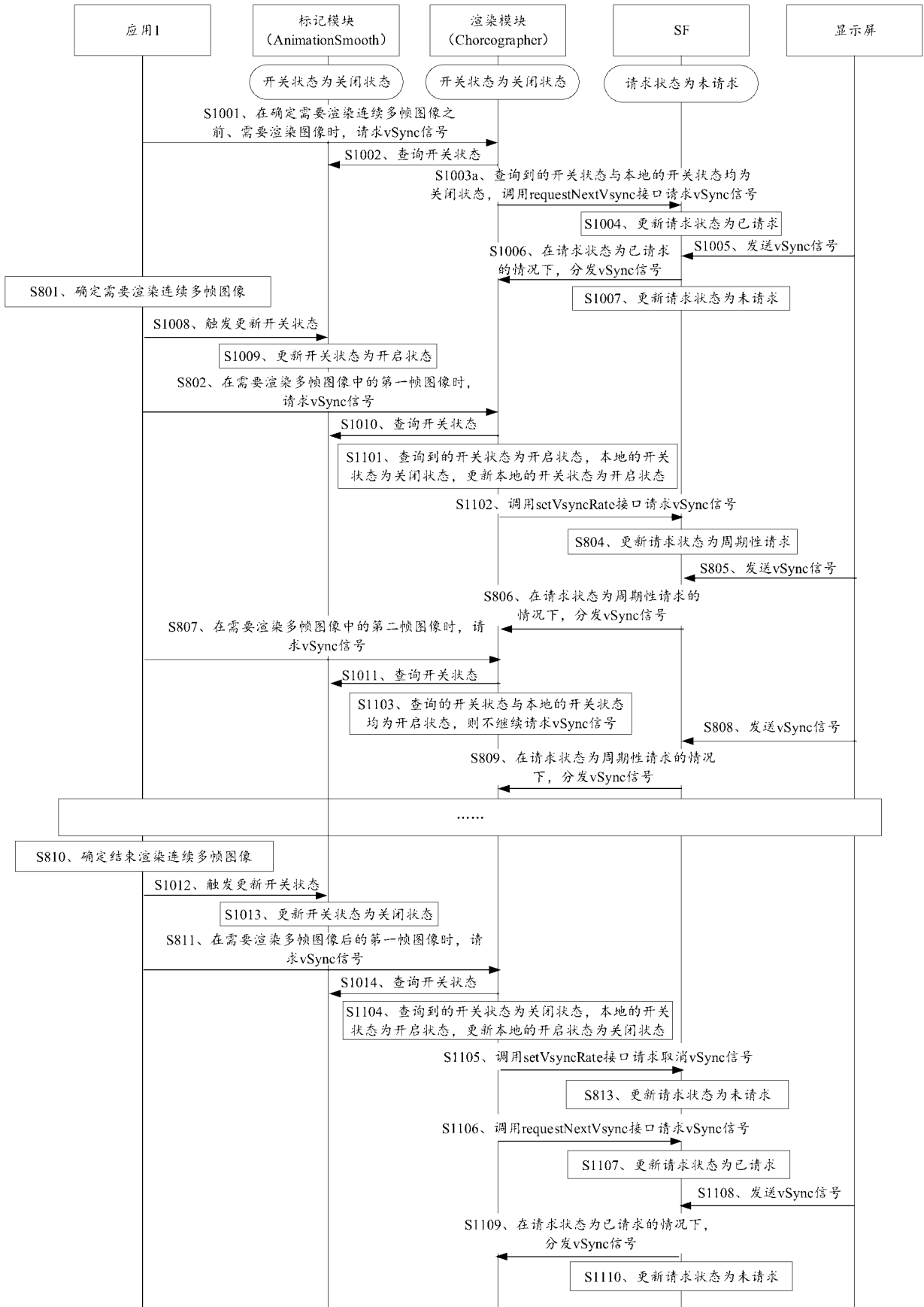


图 11

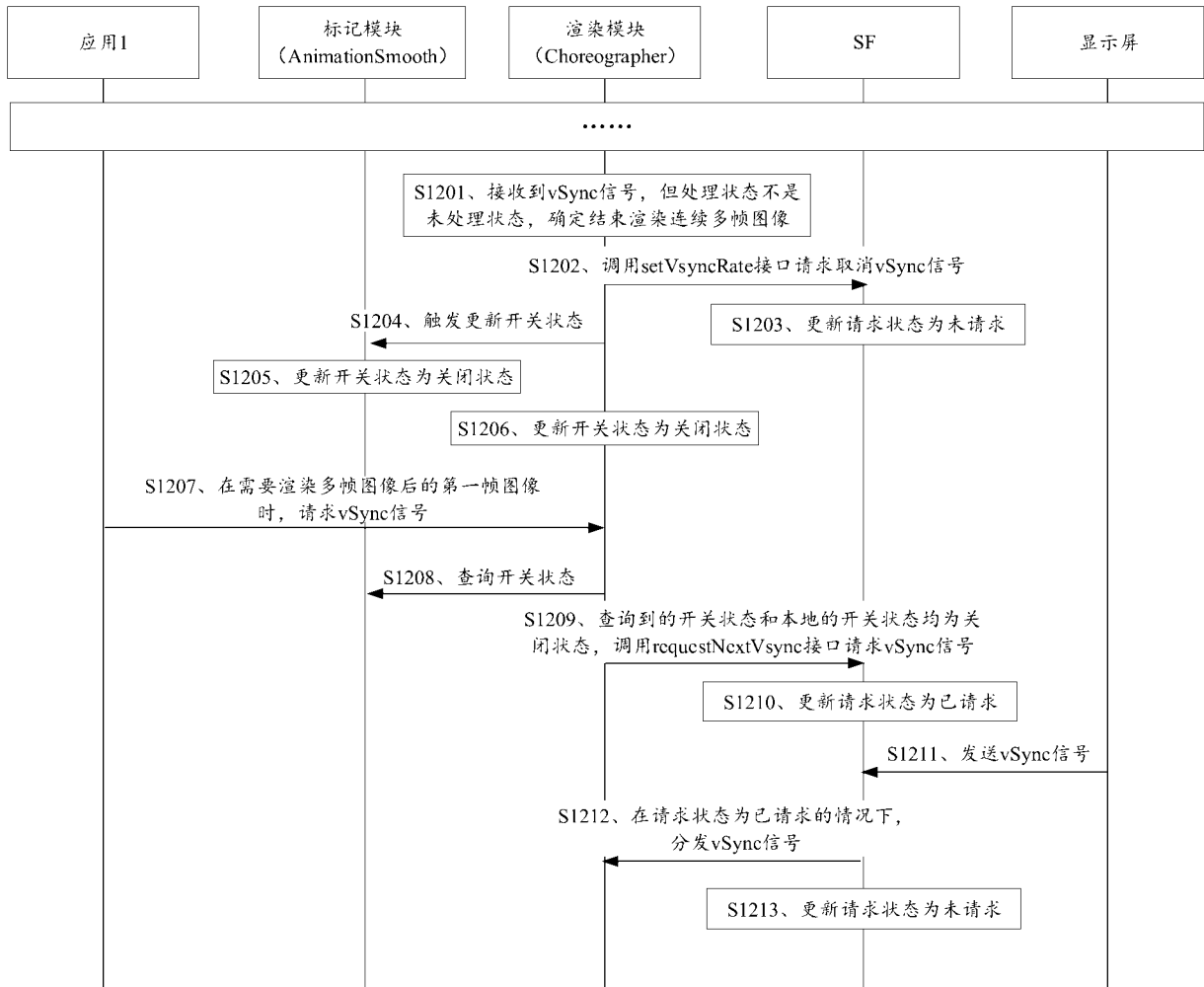


图 12

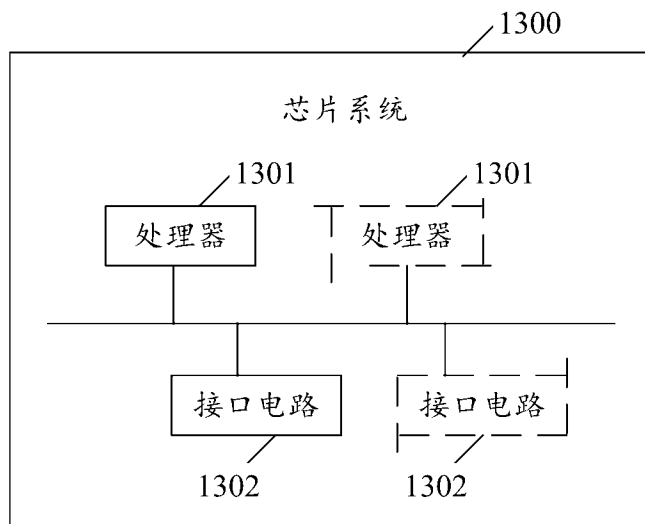


图 13

INTERNATIONAL SEARCH REPORT

International application No.

PCT/CN2023/117580

A. CLASSIFICATION OF SUBJECT MATTER G06F9/451(2018.01)i According to International Patent Classification (IPC) or to both national classification and IPC		
B. FIELDS SEARCHED Minimum documentation searched (classification system followed by classification symbols) IPC: G06F Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched Electronic data base consulted during the international search (name of data base and, where practicable, search terms used) VEN, CNABS, CNTXT, WOTXT, EPTXT, USTXT, CNKI, IEEE: 应用, 显示, 渲染, 窗口, 帧率, 信号, 周期, 请求, 结束, 取消, 接口, application, display, render, screen, frame, signal, period, request, end, cancel, interface		
C. DOCUMENTS CONSIDERED TO BE RELEVANT		
Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	CN 114518817 A (HONOR TERMINAL CO., LTD.) 20 May 2022 (2022-05-20) description, paragraphs [0120]-[0211], [0230], and [0234]	1-18
A	CN 108228358 A (GUANGDONG OPPO MOBILE TELECOMMUNICATIONS CO., LTD.) 29 June 2018 (2018-06-29) entire document	1-18
A	CN 114443269 A (HONOR TERMINAL CO., LTD.) 06 May 2022 (2022-05-06) entire document	1-18
A	US 2019206367 A1 (SAMSUNG ELECTRONICS CO., LTD.) 04 July 2019 (2019-07-04) entire document	1-18
<input type="checkbox"/> Further documents are listed in the continuation of Box C. <input checked="" type="checkbox"/> See patent family annex.		
* Special categories of cited documents: "A" document defining the general state of the art which is not considered to be of particular relevance "D" document cited by the applicant in the international application "E" earlier application or patent but published on or after the international filing date "L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified) "O" document referring to an oral disclosure, use, exhibition or other means "P" document published prior to the international filing date but later than the priority date claimed "T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention "X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone "Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art "&" document member of the same patent family		
Date of the actual completion of the international search 10 November 2023		Date of mailing of the international search report 21 November 2023
Name and mailing address of the ISA/CN China National Intellectual Property Administration (ISA/ CN) China No. 6, Xitucheng Road, Jimenqiao, Haidian District, Beijing 100088		Authorized officer Telephone No.

INTERNATIONAL SEARCH REPORT
Information on patent family members

International application No. PCT/CN2023/117580

Patent document cited in search report			Publication date (day/month/year)	Patent family member(s)	Publication date (day/month/year)
CN	114518817	A	20 May 2022	None	

CN	108228358	A	29 June 2018	None	

CN	114443269	A	06 May 2022	None	

US	2019206367	A1	04 July 2019	KR	20180022211 A 06 March 2018
				EP	3489906 A1 29 May 2019
				WO	2018038483 A1 01 March 2018

<p>A. 主题的分类</p> <p>G06F9/451(2018.01)i</p> <p>按照国际专利分类(IPC)或者同时按照国家分类和IPC两种分类</p>																	
<p>B. 检索领域</p> <p>检索的最低限度文献(标明分类系统和分类号)</p> <p>IPC: G06F</p> <p>包含在检索领域中的除最低限度文献以外的检索文献</p> <p>在国际检索时查阅的电子数据库(数据库的名称, 和使用的检索词(如使用))</p> <p>VEN, CNABS, CNTXT, WOTXT, EPTXT, USTXT, CNKI, IEEE: 应用, 显示, 渲染, 窗口, 帧率, 信号, 周期, 请求, 结束, 取消, 接口, application, display, render, screen, frame, signal, period, request, end, cancel, interface</p>																	
<p>C. 相关文件</p> <table border="1"> <thead> <tr> <th>类型*</th> <th>引用文件, 必要时, 指明相关段落</th> <th>相关的权利要求</th> </tr> </thead> <tbody> <tr> <td>X</td> <td>CN 114518817 A (荣耀终端有限公司) 2022年5月20日 (2022 - 05 - 20) 说明书[0120]-[0211]、[0230]-[0234]段</td> <td>1-18</td> </tr> <tr> <td>A</td> <td>CN 108228358 A (广东欧珀移动通信有限公司) 2018年6月29日 (2018 - 06 - 29) 全文</td> <td>1-18</td> </tr> <tr> <td>A</td> <td>CN 114443269 A (荣耀终端有限公司) 2022年5月6日 (2022 - 05 - 06) 全文</td> <td>1-18</td> </tr> <tr> <td>A</td> <td>US 2019206367 A1 (SAMSUNG ELECTRONICS CO., LTD.) 2019年7月4日 (2019 - 07 - 04) 全文</td> <td>1-18</td> </tr> </tbody> </table> <p><input type="checkbox"/> 其余文件在C栏的续页中列出。 <input checked="" type="checkbox"/> 见同族专利附件。</p> <p>* 引用文件的具体类型: “A” 认为不特别相关的表示了现有技术一般状态的文件 “D” 申请人在国际申请中引证的文件 “E” 在国际申请日的当天或之后公布的在先申请或专利 “L” 可能对优先权要求构成怀疑的文件, 或为确定另一篇引用文件的公布日而引用的或者因其他特殊理由而引用的文件(如具体说明的) “O” 涉及口头公开、使用、展览或其他方式公开的文件 “P” 公布日先于国际申请日但迟于所要求的优先权日的文件 “T” 在申请日或优先权日之后公布, 与申请不相抵触, 但为了理解发明之理论或原理的在后文件 “X” 特别相关的文件, 单独考虑该文件, 认定要求保护的发明不是新颖的或不具有创造性 “Y” 特别相关的文件, 当该文件与另一篇或者多篇该类文件结合并且这种结合对于本领域技术人员为显而易见时, 要求保护的发明不具有创造性 “&” 同族专利的文件</p>			类型*	引用文件, 必要时, 指明相关段落	相关的权利要求	X	CN 114518817 A (荣耀终端有限公司) 2022年5月20日 (2022 - 05 - 20) 说明书[0120]-[0211]、[0230]-[0234]段	1-18	A	CN 108228358 A (广东欧珀移动通信有限公司) 2018年6月29日 (2018 - 06 - 29) 全文	1-18	A	CN 114443269 A (荣耀终端有限公司) 2022年5月6日 (2022 - 05 - 06) 全文	1-18	A	US 2019206367 A1 (SAMSUNG ELECTRONICS CO., LTD.) 2019年7月4日 (2019 - 07 - 04) 全文	1-18
类型*	引用文件, 必要时, 指明相关段落	相关的权利要求															
X	CN 114518817 A (荣耀终端有限公司) 2022年5月20日 (2022 - 05 - 20) 说明书[0120]-[0211]、[0230]-[0234]段	1-18															
A	CN 108228358 A (广东欧珀移动通信有限公司) 2018年6月29日 (2018 - 06 - 29) 全文	1-18															
A	CN 114443269 A (荣耀终端有限公司) 2022年5月6日 (2022 - 05 - 06) 全文	1-18															
A	US 2019206367 A1 (SAMSUNG ELECTRONICS CO., LTD.) 2019年7月4日 (2019 - 07 - 04) 全文	1-18															
<p>国际检索实际完成的日期</p> <p>2023年11月10日</p>	<p>国际检索报告邮寄日期</p> <p>2023年11月21日</p>																
<p>ISA/CN的名称和邮寄地址</p> <p>中国国家知识产权局 中国北京市海淀区蓟门桥西土城路6号 100088</p>	<p>授权官员</p> <p>李迪</p> <p>电话号码 (+86) 010-53961300</p>																

国际检索报告
关于同族专利的信息

国际申请号

PCT/CN2023/117580

检索报告引用的专利文件			公布日 (年/月/日)	同族专利			公布日 (年/月/日)
CN	114518817	A	2022年5月20日	无			
CN	108228358	A	2018年6月29日	无			
CN	114443269	A	2022年5月6日	无			
US	2019206367	A1	2019年7月4日	KR	20180022211	A	2018年3月6日
				EP	3489906	A1	2019年5月29日
				WO	2018038483	A1	2018年3月1日