



(12)发明专利

(10)授权公告号 CN 105786479 B

(45)授权公告日 2019.05.28

(21)申请号 201410836471.6

(22)申请日 2014.12.23

(65)同一申请的已公布的文献号
申请公布号 CN 105786479 A

(43)申请公布日 2016.07.20

(73)专利权人 联芯科技有限公司
地址 200233 上海市徐汇区钦江路333号41
幢4楼

(72)发明人 季卫华

(74)专利代理机构 上海专利商标事务所有限公
司 31100

代理人 骆希聪

(51)Int.Cl.
G06F 9/448(2018.01)

(56)对比文件

CN 103646638 A,2014.03.19,

US 8069450 B2,2011.11.29,

CN 103106058 A,2013.05.15,

杨倩 等.“Android显示服务器——
SurfaceFlinger研究”.《计算机应用与软件》
.2014,

审查员 韩典伯

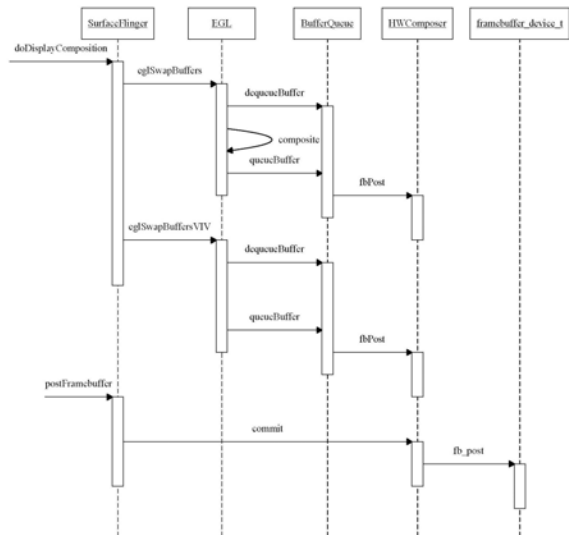
权利要求书2页 说明书5页 附图2页

(54)发明名称

一种合成内存管理方法及系统

(57)摘要

本发明提供一种合成内存管理方法及系统，应用于Android平台，包括：在EGL中添加eglSwapBuffersVIV接口；在SurfaceFlinger中进行2D合成时，调用所述eglSwapBuffersVIV接口；在SurfaceFlinger中进行3D合成时，调用eglSwapBuffers接口。本发明在进行不同的合成时调用了不同的接口，避免了来回切换时候的图像错乱问题。



1. 一种合成内存管理方法,应用于Android平台,其特征在于,包括:
 - 在EGL中添加一第一接口;
 - 在SurfaceFlinger中进行2D合成时,调用所述第一接口;在SurfaceFlinger中进行3D合成时,调用一第二接口;
 - 所述在SurfaceFlinger中进行2D合成时,调用所述第一接口的步骤包括:
 - 调用BufferQueue的dequeueBuffer函数获得一片帧缓冲区;
 - 调用BufferQueue的queueBuffer函数,使BufferQueue调用HWComposer的fbPost函数,把该片帧缓冲区交给HWComposer,作为2D加速器的目标缓冲区;
 - SurfaceFlinger调用HWComposer的commit函数,HWComposer启动2D加速器进行合成;
 - 调用framebuffer_device_t的接口fb_post,把合成好的目标缓冲区交给framebuffer_device_t送入显示器显示。
2. 如权利要求1所述的合成内存管理方法,其特征在于,所述在SurfaceFlinger中进行3D合成时,调用所述第二接口的步骤包括:
 - 调用BufferQueue的dequeueBuffer函数获得一片帧缓冲区;
 - 启动图像处理器把各个窗口的内容合成到该片帧缓冲区上,获得合成好的帧缓冲区;
 - 调用BufferQueue的queueBuffer函数,使BufferQueue调用HWComposer的fbPost函数,把所述合成好的帧缓冲区交给HWComposer,作为2D加速器的目标缓冲区;
 - SurfaceFlinger调用HWComposer的commit函数,使HWComposer调用framebuffer_device_t的接口fb_post,把所述目标缓冲区交给framebuffer_device_t送入液晶显示器显示。
3. 如权利要求1所述的合成内存管理方法,其特征在于,所述第一接口为eglSwapBuffersVIV接口,所述第二接口为eglSwapBuffers接口。
4. 一种合成内存管理系统,应用于Android平台,其特征在于,包括:
 - 添加模块,用于在EGL中添加一第一接口;
 - 调用模块,用于在SurfaceFlinger中进行2D合成时,调用所述第一接口;在SurfaceFlinger中进行3D合成时,调用一第二接口;
 - 在所述调用模块中,在SurfaceFlinger中进行2D合成调用所述第一接口时,还包括:
 - 第一调用模块,用于调用BufferQueue的dequeueBuffer函数获得一片帧缓冲区;
 - 第二调用模块,用于调用BufferQueue的queueBuffer函数,使BufferQueue调用HWComposer的fbPost函数,把该片帧缓冲区交给HWComposer,作为2D加速器的目标缓冲区;
 - 第三调用模块,用于SurfaceFlinge调用HWCompose的commit函数,HWComposer启动2D加速器进行合成;
 - 第四调用模块,用于调用framebuffer_device_t的接口fb_post,把合成好的目标缓冲区交给framebuffer_device_t送入液晶显示器显示。
5. 如权利要求4所述的合成内存管理系统,其特征在于,在调用模块中,在SurfaceFlinger中进行3D合成调用一第二接口时,还包括:
 - 第五调用模块,用于调用BufferQueue的dequeueBuffer函数获得一片帧缓冲区;
 - 合成模块,用于启动图像处理器把各个窗口的内容合成到该片帧缓冲区上,获得合成好的帧缓冲区;

第六调用模块,用于调用BufferQueue的queueBuffer函数,使BufferQueue调用HWComposer的fbPost函数,把所述合成好的帧缓冲区交给HWComposer,作为2D加速器的目标缓冲区;

第七调用模块,用于SurfaceFlinger调用HWComposer的commit函数,使HWComposer调用framebuffer_device_t的接口fb_post,把所述目标缓冲区交给framebuffer_device_t送入液晶显示器显示。

6.如权利要求4所述的合成内存管理系统,其特征在于,所述第一接口为eglSwapBuffersVIV接口,所述第二接口为eglSwapBuffers接口。

一种合成内存管理方法及系统

技术领域

[0001] 本发明涉及通讯技术领域,特别涉及一种合成内存管理方法及系统。

背景技术

[0002] 智能手机终端领域,需要根据窗口的位置、大小、层叠顺序,把多个窗口的内容合成为一幅图像,保存到帧缓冲区,送给LCD显示。这个合成工作通常是靠GPU(图像处理器)完成的,但是为了降低功耗,提高续航时间,会加入2D芯片。大多数时候不需要3D芯片合成,就可让GPU休息,改让2D芯片加速器完成合成工作。这样就涉及到2D和3D合成切换时对帧缓冲区的管理问题。

[0003] 请参考图1,其是Android原生平台的合成序列图。如图1所示,SurfaceFlinger是整个Android显示系统的管理核心,控制各个窗口的叠加,调用GPU或2D加速器进行合成。EGL是OpenGL ES中与窗口操作相关的一套标准API。BufferQueue是缓存队列,为了画面流畅,通常含有三帧缓存。HWComposer是硬件合成器,其实就是对2D加速器的封装。framebuffer_device_t是帧缓冲区设备,负责把帧缓冲区送入LCD显示。

[0004] 从图1的时序图可以看出,SurfaceFlinger在准备好要合成的各个窗口后,就调用EGL标准接口eglSwapBuffers。在这个接口中,首先调用BufferQueue的dequeueBuffer函数获得一片帧缓冲区,然后启动GPU把各个窗口的内容合成到这片帧缓冲区上,接着调用BufferQueue的queueBuffer函数。BufferQueue就会调用HWComposer的fbPost函数,把这片合成好的帧缓冲区交给HWComposer,作为2D加速器的目标缓冲区。最后,SurfaceFlinger调用HWComposer的commit函数,HWComposer就会调用framebuffer_device_t的接口fb_post,把目标缓冲区交给framebuffer_device_t送入LCD显示。

[0005] 以上是针对3D合成的,而对于2D合成,SurfaceFlinger不会调用EGL标准接口eglSwapBuffers,而是直接调用HWComposer的commit函数。HWComposer会启动2D加速器进行合成,最后也会调用framebuffer_device_t的接口fb_post,把合成好的目标缓冲区交给framebuffer_device_t送入LCD显示。

[0006] 但是,通过上述方法进行合成,存在以下缺陷:

[0007] 1) 当3D合成切换到2D合成后,帧缓冲区不再轮转。也就是说,2D加速器正在合成的目标缓冲区就是LCD正在显示的帧缓冲区,这会导致图像错乱。

[0008] 2) 如果让2D加速器独自管理帧缓冲区,实现帧缓冲区的轮转,是可以保证2D合成时目标缓冲区不是LCD正在显示的帧缓冲区。但是,当2D合成切换回3D合成时,GPU并不知道帧缓冲区发生了轮转,所以3D合成的目标缓冲区可能正是LCD正在显示的帧缓冲区。不过,这只会发生在切换的过程中,当EGL轮转完一遍帧缓冲区后,就进入了正常状态。但是,短暂的花屏也会影响用户体验。

发明内容

[0009] 本发明的目的在于提供一种合成内存管理方法及系统,以解决现有合成内存方法

管理不善而带来的图像错乱的问题。

[0010] 为解决上述技术问题,本发明提供一种合成内存管理方法,应用于Android平台,包括:

[0011] 在EGL中添加一第一接口;

[0012] 在SurfaceFlinger中进行2D合成时,调用所述第一接口;在SurfaceFlinger中进行3D合成时,调用一第二接口。

[0013] 进一步的,在所述的合成内存管理方法中,所述在SurfaceFlinger中进行2D合成时,调用所述第一接口的步骤包括:

[0014] 调用BufferQueue的dequeueBuffer函数获得一片帧缓冲区;

[0015] 调用BufferQueue的queueBuffer函数,使BufferQueue调用HWComposer的fbPost函数,把该片帧缓冲区交给HWComposer,作为2D加速器的目标缓冲区;

[0016] SurfaceFlinger调用HWComposer的commit函数,HWComposer启动2D加速器进行合成;

[0017] 调用framebuffer_device_t的接口fb_post,把合成好的目标缓冲区交给framebuffer_device_t送入液晶显示器显示。

[0018] 进一步的,在所述的合成内存管理方法中,所述在SurfaceFlinger中进行3D合成时,调用一第二接口的步骤包括:

[0019] 调用BufferQueue的dequeueBuffer函数获得一片帧缓冲区;

[0020] 启动图像处理器把各个窗口的内容合成到该片帧缓冲区上;

[0021] 调用BufferQueue的queueBuffer函数,使BufferQueue调用HWComposer的fbPost函数,把所述合成好的帧缓冲区交给HWComposer,作为2D加速器的目标缓冲区;

[0022] SurfaceFlinger调用HWComposer的commit函数,使HWComposer调用framebuffer_device_t的接口fb_post,把所述目标缓冲区交给framebuffer_device_t送入液晶显示器显示。

[0023] 进一步的,在所述的合成内存管理方法中,所述第一接口为eglSwapBuffersVIV接口,所述第二接口为eglSwapBuffers接口。

[0024] 相应的,本发明还提供一种合成内存管理系统,应用于Android平台,包括:

[0025] 添加模块,用于在EGL中添加一第一接口;

[0026] 调用模块,用于在SurfaceFlinger中进行2D合成时,调用所述第一接口;在SurfaceFlinger中进行3D合成时,调用一第二接口。

[0027] 进一步的,在所述的合成内存管理系统中,在调用模块中,在SurfaceFlinger中进行2D合成调用所述第一接口时,还包括:

[0028] 第一调用模块,用于调用BufferQueue的dequeueBuffer函数获得一片帧缓冲区;

[0029] 第二调用模块,用于调用BufferQueue的queueBuffer函数,使BufferQueue调用HWComposer的fbPost函数,把该片帧缓冲区交给HWComposer,作为2D加速器的目标缓冲区;

[0030] 第三调用模块,用于SurfaceFlinger调用HWComposer的commit函数,HWComposer启动2D加速器进行合成;

[0031] 第四调用模块,用于调用framebuffer_device_t的接口fb_post,把合成好的目标缓冲区交给framebuffer_device_t送入液晶显示器显示。

[0032] 进一步的,在所述的合成内存管理系统中,在调用模块中,在SurfaceFlinger中进行3D合成调用一第二接口时,还包括:

[0033] 第五调用模块,用于调用BufferQueue的dequeueBuffer函数获得一片帧缓冲区;

[0034] 合成模块,用于启动图像处理器把各个窗口的内容合成到该片帧缓冲区上;

[0035] 第六调用模块,用于调用BufferQueue的queueBuffer函数,使BufferQueue调用HWComposer的fbPost函数,把所述合成好的帧缓冲区交给HWComposer,作为2D加速器的目标缓冲区;

[0036] 第七调用模块,用于SurfaceFlinger调用HWComposer的commit函数,使HWComposer调用framebuffer_device_t的接口fb_post,把所述目标缓冲区交给framebuffer_device_t送入液晶显示器显示。

[0037] 进一步的,在所述的合成内存管理系统中,所述第一接口为eglSwapBuffersVIV接口,所述第二接口为eglSwapBuffers接口。

[0038] 本发明在进行不同的合成时调用了不同的接口,避免了来回切换时候的图像错乱问题。此外,在2D合成时不启动GPU,省去了进行3D合成的动作,只保留轮转帧缓冲区的功能,维持了低功耗;进一步的,简化了2D加速器设计,直接使用传过来的帧缓冲区作为目标缓冲区,而不需要独自管理帧缓冲区;更进一步的,统一由SurfaceFlinger触发帧缓冲区轮转,不破坏原生架构,代码结构清晰,利于维护。

附图说明

[0039] 图1是现有Android原生平台的合成时序图;

[0040] 图2是本发明实施例的合成内存管理方法及系统的时序图。

具体实施方式

[0041] 以下结合附图和具体实施例对本发明提出的合成内存管理方法及系统作进一步详细说明。根据下面说明和权利要求书,本发明的优点和特征将更清楚。需说明的是,附图均采用非常简化的形式且均使用非精准的比例,仅用以方便、明晰地辅助说明本发明实施例的目的。

[0042] 请参考图2,其是本发明实施例的合成内存管理方法及系统的时序图。本发明提供一种合成内存管理方法,应用于Android平台,具体包括以下步骤:

[0043] S100:在EGL中添加一第一接口,在本实施例中,命名第一接口为eglSwapBuffersVIV接口,它的功能与eglSwapBuffersVIV接口相似,是为了在后续2D合成时调用的。

[0044] S200:在SurfaceFlinger中进行2D合成时,调用所述eglSwapBuffersVIV接口;在SurfaceFlinger中进行3D合成时,调用一第二接口,在本实施例中,即调用eglSwapBuffers接口。

[0045] 在SurfaceFlinger中进行2D合成,调用所述eglSwapBuffersVIV接口时,首先调用BufferQueue的dequeueBuffer函数获得一片帧缓冲区;其次,调用BufferQueue的queueBuffer函数,使BufferQueue调用HWComposer的fbPost函数,把该片帧缓冲区交给HWComposer,作为2D加速器的目标缓冲区;然后SurfaceFlinger调用HWComposer的commit

函数,HWComposer启动2D加速器进行合成;最后,调用framebuffer_device_t的接口fb_post,把合成好的目标缓冲区交给framebuffer_device_t送入液晶显示器(LCD)显示。

[0046] 在SurfaceFlinger中进行3D合成,调用所述eglSwapBuffers接口时,首先调用BufferQueue的dequeueBuffer函数获得一片帧缓冲区;其次启动图像处理器(GPU)把各个窗口的内容合成到该片帧缓冲区上;随后调用BufferQueue的queueBuffer函数,使BufferQueue调用HWComposer的fbPost函数,把所述合成好的帧缓冲区交给HWComposer,作为2D加速器的目标缓冲区;最后,SurfaceFlinger调用HWComposer的commit函数,使HWComposer调用framebuffer_device_t的接口fb_post,把所述目标缓冲区交给framebuffer_device_t送入液晶显示器(LCD)显示。

[0047] 相应的,本发明还提供一种合成内存管理系统,应用于Android平台,包括:

[0048] 添加模块,用于在EGL中添加一第一接口,在本实施例中,命名第一接口为eglSwapBuffersVIV接口;及

[0049] 调用模块,用于在SurfaceFlinger中进行2D合成时,调用所述eglSwapBuffersVIV接口;在SurfaceFlinger中进行3D合成时,调用一第二接口,在本实施例中,即调用eglSwapBuffers接口。

[0050] 具体来说,在调用模块中,在SurfaceFlinger中进行2D合成调用所述eglSwapBuffersVIV接口时,还包括:

[0051] 第一调用模块,用于调用BufferQueue的dequeueBuffer函数获得一片帧缓冲区;

[0052] 第二调用模块,用于调用BufferQueue的queueBuffer函数,使BufferQueue调用HWComposer的fbPost函数,把该片帧缓冲区交给HWComposer,作为2D加速器的目标缓冲区;

[0053] 第三调用模块,用于SurfaceFlinger调用HWComposer的commit函数,HWComposer启动2D加速器进行合成;

[0054] 第四调用模块,用于调用framebuffer_device_t的接口fb_post,把合成好的目标缓冲区交给framebuffer_device_t送入液晶显示器显示。

[0055] 进一步的,在调用模块中,在SurfaceFlinger中进行3D合成调用所述eglSwapBuffers接口时,还包括:

[0056] 第五调用模块,用于调用BufferQueue的dequeueBuffer函数获得一片帧缓冲区;

[0057] 合成模块,用于启动图像处理器把各个窗口的内容合成到该片帧缓冲区上;

[0058] 第六调用模块,用于调用BufferQueue的queueBuffer函数,使BufferQueue调用HWComposer的fbPost函数,把所述合成好的帧缓冲区交给HWComposer,作为2D加速器的目标缓冲区;

[0059] 第七调用模块,用于SurfaceFlinger调用HWComposer的commit函数,使HWComposer调用framebuffer_device_t的接口fb_post,把所述目标缓冲区交给framebuffer_device_t送入液晶显示器显示。

[0060] 基此,本发明在进行不同的合成时调用了不同的接口,避免了来回切换时候的图像错乱问题。此外,在2D合成时不启动GPU,省去了进行3D合成的动作,只保留轮转帧缓冲区的功能,维持了低功耗;进一步的,简化了2D加速器设计,直接使用传过来的帧缓冲区作为目标缓冲区,而不需要独自管理帧缓冲区;更进一步的,统一由SurfaceFlinger触发帧缓冲区轮转,不破坏原生架构,代码结构清晰,利于维护。

[0061] 上述描述仅是对本发明较佳实施例的描述,并非对本发明范围的任何限定,本发明领域的普通技术人员根据上述揭示内容做的任何变更、修饰,均属于权利要求书的保护范围。

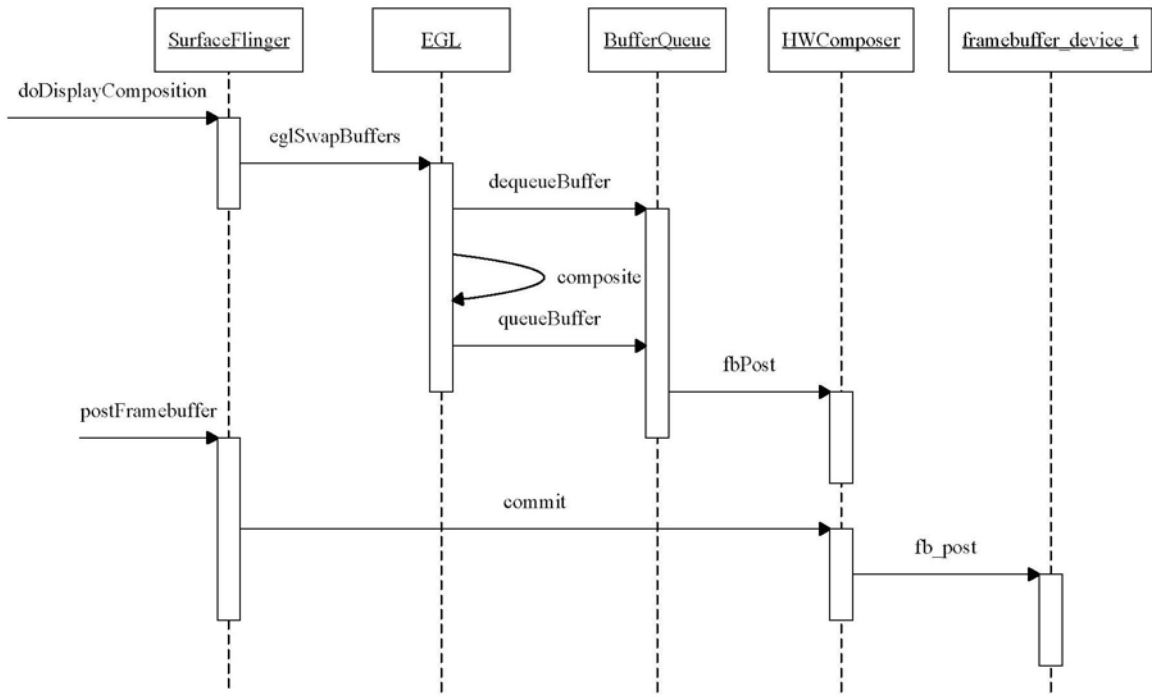


图1

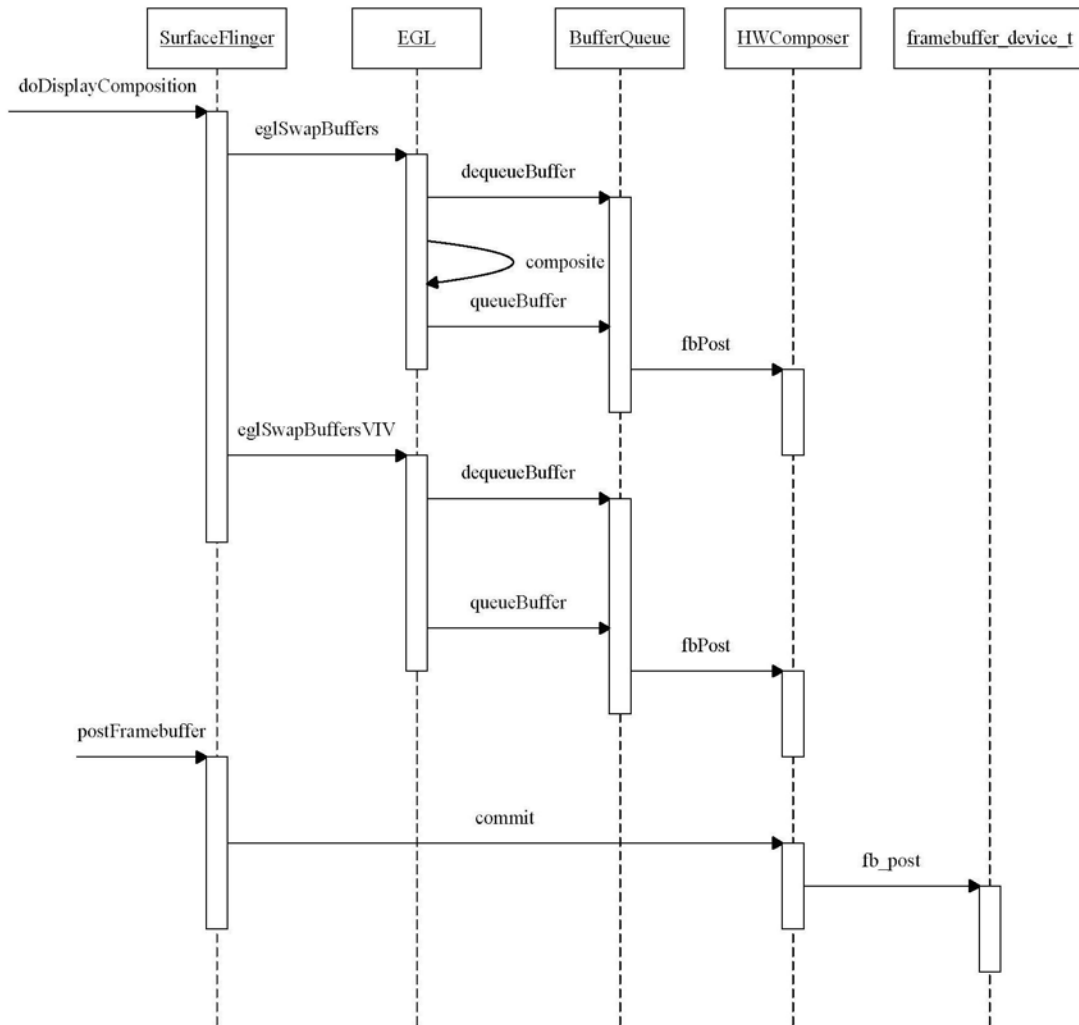


图2