



(12) 发明专利

(10) 授权公告号 CN 108366301 B

(45) 授权公告日 2021.03.09

(21) 申请号 201810373530.9

H04N 21/472 (2011.01)

(22) 申请日 2018.04.24

H04N 21/4782 (2011.01)

(65) 同一申请的已公布的文献号
申请公布号 CN 108366301 A

(56) 对比文件

CN 107197393 A, 2017.09.22

US 2006246985 A1, 2006.11.02

(43) 申请公布日 2018.08.03

审查员 叶会

(73) 专利权人 中国广播电视网络有限公司
地址 100045 北京市西城区白云路10号

(72) 发明人 李爽 傅力军 杨旭 白鹤 谌颖
贾兴华 刘江

(74) 专利代理机构 北京天达知识产权代理事务
所(普通合伙) 11386

代理人 庞许倩 张焱

(51) Int. Cl.

H04N 21/443 (2011.01)

H04N 21/431 (2011.01)

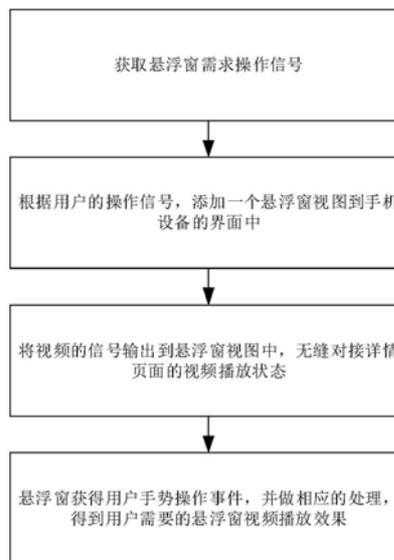
权利要求书2页 说明书5页 附图1页

(54) 发明名称

一种基于Android的视频悬浮播放方法

(57) 摘要

本发明涉及一种基于Android的视频悬浮播放方法,包括步骤S1、产生悬浮窗需求的信号;步骤S2、通过系统的窗口管理器,创建悬浮窗视图并添加到系统视图的最顶层;步骤S3、将视频的信号输出到悬浮窗视图中,无缝对接详情页面的视频播放状态步骤S4、悬浮窗获得用户手势操作事件,并做相应的处理,得到用户需要的悬浮窗视频播放效果。本发明提升用户使用视频播放类App的用户体验,完成在用户观看视频的同时,能够自由操作手机的目的,不再受限于离开视频播放页面视频关闭的情况。



1. 一种基于Android的视频悬浮播放方法,其特征在于,包括以下步骤:

步骤S1、获取悬浮窗需求操作信号;

步骤S2、Android终端根据获取到的所述需求操作信号,通过系统的窗口管理器创建悬浮窗视图,并将所创建的悬浮窗视图添加到系统视图的最顶层;

步骤S3、输出视频信号到悬浮窗视图中,无缝对接当前页面的视频播放状态;由播放器负责视频图像的处理和输出,由Surface来实现悬浮窗呈现视频图像;

视频播放状态的无缝对接是指当视频在详情页面播放的时候,如果要切换到悬浮窗呈现视频,其切换到悬浮窗过程中视频不会出现包括暂停和回退在内的断播现象;

所述无缝对接的实现方法包括:

1) 通过将播放器做成单例模式,实现播放器的状态保持,播放器不能被暂停、释放播放资源;所述单例模式的播放器只产生一个实例,并且在有装置获取播放器实例时,始终返回该实例;

2) 采用Surface切换,将视频播放器的原始视频呈现对象切换为悬浮窗的视频呈现对象;所述切换通过调用播放器设置的输出对象,并将要输出的对象进行传递实现的;

采用TextureView视图实现Surface底层图像渲染;所述TextureView运行在Android的UI主线程中,视图更新操作对于TextureView会直接进行;

3) 将悬浮窗的视频呈现对象传递给播放器,播放器接收到后主动将视频输出对象切换为目标;

所述无缝对接对播放器单例后,不再对播放器进行暂停操作和释放资源的操作;

步骤S4、根据用户手势操作指令,处理视频图像,得到用户需要的悬浮窗视频播放效果。

2. 根据权利要求1所述的视频悬浮播放方法,其特征在于,所述系统的窗口管理器是利用Android的WindowManager窗口管理器。

3. 根据权利要求2所述的视频悬浮播放方法,其特征在于,对WindowManager窗口管理器悬浮窗视图参数mWindowParams.flags配置为:mWindowParams.flags =

WindowManager.LayoutParams.FLAG_NOT_TOUCH_MODAL,

WindowManager.LayoutParams.FLAG_NOT_FOCUSABLE。

4. 根据权利要求1所述的视频悬浮播放方法,其特征在于,悬浮窗获得用户手势操作事件包括:缩小悬浮窗视图、放大悬浮窗视图和移动悬浮窗视图。

5. 根据权利要求4所述的视频悬浮播放方法,其特征在于,所述缩小悬浮窗视图的手势操作包括以下操作中的至少一个:点击缩小按钮;点击悬浮窗的固定按钮,当按下按钮不松开且向上移动;触摸事件中的双指缩小操作;

通过windowManager窗口管理器将缩小悬浮窗视图的手势操作传递给悬浮窗视图,更新悬浮窗视图的大小,每次缩小的动作都将悬浮窗的宽高按比例减小10像素,缩小的最低值为屏幕宽的1/4。

6. 根据权利要求4所述的视频悬浮播放方法,其特征在于,所述放大悬浮窗视图的手势操作包括以下操作中的至少一个:点击放大按钮;点击悬浮窗的固定按钮,当按下按钮不松开且向下移动;触摸事件中的双指放大操作;

通过windowManager窗口管理器将放大悬浮窗视图的手势操作传递给悬浮窗视图,更

新悬浮窗视图的大小,每次放大的动作都将悬浮窗的宽高按比例放大10像素,放大的最大值为屏幕宽。

7.根据权利要求4所述的视频悬浮播放方法,其特征在于,所述移动悬浮窗视图手势操作,包括手指按住悬浮窗视图不松开,且移动的手势操作;

根据手势操作,得到触摸事件屏幕坐标的变化,计算手指的移动坐标,并通过windowManager窗口管理器传递给悬浮窗视图,更新悬浮窗视图的位置。

8.根据权利要求5-7任一所述的视频悬浮播放方法,其特征在于,实时传递所述用户手势操作事件,当检测到手势操作引起变化后,即通过窗口管理器更新视图。

一种基于Android的视频悬浮播放方法

技术领域

[0001] 本发明涉及通信技术领域,尤其涉及一种基于Android的视频悬浮播放方法。

背景技术

[0002] 目前Android市场的视频播放类应用中,支持悬浮窗播放的应用比较少,部分手机版的网页支持浏览器端的悬浮窗播放效果。一般的视频播放View层都放在Activity/Fragment容器中,要实现悬浮窗效果,可以把Activity/Fragment的布局文件模拟一下小窗的效果,并把Activity的theme属性设置为透明,这样,Activity/Fragment在播放视频的时候可以达到一种悬浮窗播放的效果,但实际视频的播放还存在Activity/Fragment中播放,用户可以看到后面的Activity的内容,但是不可以操作后面的Activity问题;通过事件分发的手段来解决这个问题,实现视频播放页和小窗播放页面为同一个Activity,视频切换到小窗播放的时候是无缝切换的,即,画面和声音都不会出现暂停或中断,但仍然存在如下缺点:

[0003] 无法实现针对于系统层次的悬浮窗效果,即悬浮窗播放效果的作用域为当前的应用,一旦应用后台或者关闭,悬浮窗则会关闭;

[0004] 通过事件分发来达到底层Activity响应的效果,会造成Activity的事件响应的不同程度的延迟,延迟程度与手机的硬件配置有关。

发明内容

[0005] 鉴于上述的分析,本发明旨在提供一种基于Android的视频悬浮播放方法,实现作用域为系统层的悬浮窗播放窗口,在不影响系统正常使用的同时,在悬浮窗播放视频,且实现播放的无缝切换。

[0006] 本发明的目的主要是通过以下技术方案实现的:

[0007] 一种基于Android的视频悬浮播放方法,包括以下步骤:

[0008] 步骤S1、获取悬浮窗需求操作信号;

[0009] 步骤S2、Android终端根据获取到的所述需求操作信号,通过系统的窗口管理器创建悬浮窗视图,并将所创建的悬浮窗视图添加到系统视图的最顶层;

[0010] 步骤S3、输出视频信号到悬浮窗视图中,无缝对接当前页面的视频播放状态;由播放器负责视频图像的处理和输出,由Surface来实现悬浮窗呈现视频图像;

[0011] 步骤S4、根据用户手势操作指令,处理视频图像,得到用户需要的悬浮窗视频播放效果。

[0012] 进一步地,所述系统的窗口管理器是利用Android的WindowManager窗口管理器。

[0013] 使用WindowManager添加的视图是系统全局的,与各应用独立,不会互相影响,在页面或者应用退到后台后,悬浮窗仍然工作,不受影响。

[0014] 进一步地,对WindowManager窗口管理器悬浮窗视图View的参数mWindowParams.flags配置为:

[0015] `mWindowParams.flags=`

[0016] `WindowManager.LayoutParams.FLAG_NOT_TOUCH_MODAL,`

[0017] `WindowManager.LayoutParams.FLAG_NOT_FOCUSABLE;`

[0018] 所述`FLAG_NOT_TOUCH_MODAL`:即使在该窗口在可获得焦点情况下,仍然把该窗口之外的任何事件发送到该窗口之后的其他窗口;

[0019] 所述`FLAG_NOT_FOCUSABLE`:让窗口不能获得焦点,这样用户就不能向该窗口发送按键事件及按钮事件。

[0020] 进一步地,采用`TextureView`视图实现`Surface`底层图像渲染。所述`TextureView`运行在Android的UI主线程中,视图更新操作对于`TextureView`会直接进行,因此,不会产生闪烁、黑屏现象。

[0021] 进一步地,所述无缝对接的实现方法包括:

[0022] 将播放器做成单例模式,实现播放器的状态保持;

[0023] 采用`Surface`切换,将视频播放器的原始视频呈现对象切换为悬浮窗的视频呈现对象;

[0024] 将悬浮窗的视频呈现对象传递给播放器,播放器主动将视频输出对象切换为目标。

[0025] 播放器单例后,不再对播放器进行暂停操作和释放资源的操作,切换到悬浮窗呈现视频,不出现暂停、回退等现象。

[0026] 进一步地,悬浮窗获得用户手势操作事件包括:缩小悬浮窗视图、放大悬浮窗视图和移动悬浮窗视图。

[0027] 进一步地,所述缩小悬浮窗视图的手势操作包括以下操作中的至少一个:点击缩小“-”按钮;点击悬浮窗的固定按钮,当按下按钮不松开且向上移动;触摸事件中的双指缩小操作;

[0028] 通过`windowManager`窗口管理器将缩小悬浮窗视图的手势操作传递给悬浮窗视图,更新悬浮窗视图的大小,每次缩小的动作都将悬浮窗的宽高按比例减小10像素,缩小的最低值为屏幕宽的1/4。

[0029] 进一步地,所述放大悬浮窗视图的手势操作包括以下操作中的至少一个:点击放大“+”按钮;点击悬浮窗的固定按钮,当按下按钮不松开且向下移动;触摸事件中的双指放大操作;

[0030] 通过`windowManager`窗口管理器将放大悬浮窗视图的手势操作传递给悬浮窗视图,更新悬浮窗视图的大小,每次放大的动作都将悬浮窗的宽高按比例放大10像素,放大的最大值为屏幕宽。

[0031] 进一步地,所述移动悬浮窗视图手势操作,包括手指按住悬浮窗视图不松开,且移动的手势操作;

[0032] 根据手势操作,得到触摸事件屏幕坐标的变化,计算手指的移动坐标,并通过`windowManager`窗口管理器传递给悬浮窗视图,更新悬浮窗视图的位置。

[0033] 进一步地,所述用户手势操作事件的传递是实时的,当检测到手势操作引起变化后,立刻通过窗口管理器更新视图。这样可达到连续的效果,避免悬浮窗闪烁。

[0034] 本发明有益效果如下:

[0035] 通过使用悬浮窗使在用户观看视频的同时,能够自由操作手机的目的,不再受限于离开视频播放页面视频关闭的情况,提升用户使用视频播放类App的用户体验;使用系统全局视图,与各应用独立,不会互相影响,在页面或者应用退到后台后,悬浮窗仍然工作,不受影响;采用TextureView视图实现Surface底层图像渲染,不产生闪烁、黑屏现象;实现播放器与悬浮窗的无缝对接,不出现暂停、回退等现象。

附图说明

[0036] 附图仅用于示出具体实施例的目的,而并不认为是对本发明的限制,在整个附图中,相同的参考符号表示相同的部件。

[0037] 图1为基于Android的视频悬浮播放方法流程图。

具体实施方式

[0038] 下面结合附图来具体描述本发明的优选实施例,其中,附图构成本申请一部分,并与本发明的实施例一起用于阐释本发明的原理。

[0039] 本发明的一个具体实施例,公开了一种基于Android的视频悬浮播放方法,如图1所示,包括以下步骤:

[0040] 步骤S1、用户通过对Android终端界面进行悬浮窗需求操作,产生悬浮窗需求的信号;

[0041] 所述悬浮窗需求是针对全局的悬浮窗,而不是只能在某一个页面或者某一个应用中才能使用的悬浮窗,离开页面或者应用后悬浮窗不能使用;

[0042] 步骤S2、Android终端接收到用户的操作信号后,通过系统的窗口管理器,创建悬浮窗视图并添加到系统视图的最顶层;

[0043] 在页面或者应用退到后台后,悬浮窗仍然工作,不受影响;

[0044] 所述系统的窗口管理器是利用Android的WindowManager窗口管理器添加一个悬浮窗视图到手机设备的界面中;所述WindowManager类是Android中Android API中提供的一个重要服务;WindowManager Service是全局的,是唯一的;它将用户的操作,翻译成为指令,发送给呈现在界面上的各个Window;Activity会将顶级的控件注册到Window Manager中;因此WindowManager中添加的视图是系统全局的,与各应用独立,不会互相影响。

[0045] 用WindowManager的好处是事件的处理是由系统分发并处理的,应用层没有拦截,其次是作用域为全局的,悬浮窗的操作与系统的操作没有冲突。

[0046] 实现悬浮窗的具体方法包括:

[0047] 1) 获取窗口管理器服务WindowManager;

[0048] 通过:

[0049] (WindowManager) context.getSystemService(Context.WINDOW_SERVICE) 获取系统的窗口管理器;

[0050] 2) 添加悬浮窗视图View:

[0051] 通过:

[0052] mWindowManager.addView(View view,ViewGroup.LayoutParams params) 添加悬浮窗视图到系统视图中;

[0053] 其中LayoutParams为修改所述悬浮窗视图View的关键配置参数,包括有获取焦点、不获取焦点、能否触摸、能否响应按键在内的关键配置参数,

[0054] 通过其中一个重要参数mWindowParams.flags,指定View的接触模式touch;

[0055] 本实施例对该参数的配置如下:

[0056] mWindowParams.flags=

[0057] WindowManager.LayoutParams.FLAG_NOT_TOUCH_MODAL,

[0058] WindowManager.LayoutParams.FLAG_NOT_FOCUSABLE;

[0059] 所述FLAG_NOT_TOUCH_MODAL:即使在该窗口在可获得焦点情况下,仍然把该窗口之外的任何事件发送到该窗口之后的其他窗口;

[0060] FLAG_NOT_FOCUSABLE:让窗口不能获得焦点,这样用户就不能向该窗口发送按键事件及按钮事件。

[0061] 步骤S3、将视频的信号输出到悬浮窗视图中,无缝对接当前页面的视频播放状态;由播放器负责视频图像的处理和输出;由Surface来实现悬浮窗呈现视频图像;

[0062] 所述当前页面的视频播放状态为包含视频的播放窗口和相关信息展示的详情页面视频播放状态,并且是设有悬浮窗按钮的视频播放状态;在此状态下,点击悬浮窗按钮即产生悬浮窗需求的信号,实现视频的悬浮窗播放;所述播放的无缝对接是指当视频在详情页面播放的时候,如果要切换到悬浮窗呈现视频,即切换到悬浮窗过程中视频不会出现包括暂停和回退在内的断播现象。

[0063] 所述Surface是Android底层图像的渲染区,而SurfaceView和TextureView两种视图均对这个渲染区做了具体实现;

[0064] 本方案采用TextureView视图,因为TextureView运行在Android的UI主线程中,我们的视图更新操作对于TextureView会直接进行,不会产生闪烁、黑屏现象;而采用SurfaceView视图,因为SurfaceView由Android底层直接处理,非运行在UI主线程中,则会产生黑屏、闪烁等现象。

[0065] 所述无缝对接的实现方法包括:

[0066] 1) 通过将播放器做成单例模式,实现播放器的状态保持,即播放器不能被暂停、释放播放资源;所述单例模式播放器只产生一个实例,并且有装置在获取播放器实例时,始终返回该实例。

[0067] 2) 采用Surface切换,将视频播放器的原始视频呈现对象切换为悬浮窗的视频呈现对象;所述切换通过调用播放器设置的输出对象,并将要输出的对象进行传递实现的。

[0068] 3) 将悬浮窗的视频呈现对象传递给播放器,播放器接收到后主动将视频输出对象切换为目标。

[0069] 所述无缝对接对播放器单例后,不再对播放器进行暂停操作和释放资源的操作。

[0070] 步骤S4、悬浮窗获得用户手势操作事件,并做相应的处理;

[0071] 具体包括:

[0072] 1) 缩小悬浮窗视图手势操作,该操作包括但不限于点击“-”按钮;包括以下操作:点击缩小按钮;点击悬浮窗的固定按钮,当按下按钮不松开且向上移动;触摸事件中的双指缩小操作;通过windowManager窗口管理器将缩小操作传递给悬浮窗视图,更新悬浮窗视图的大小,达到缩小效果;每次缩小的动作通过窗口管理器,按比例减小10像素改变悬浮窗视

图的长、宽相对于屏幕的坐标参数后,重新绘制视图进行显示;所述悬浮窗视图缩小的最低值为屏幕宽的1/4;当检测到发生缩小变化后,更新视图大小,每次等比例减小的耗时约50毫秒,超出人体的感官能力范围,达到连续的、实时的效果,无闪烁。

[0073] 2) 放大悬浮窗视图手势操作,该操作包括但不限于点击“+”按钮;包括以下操作:点击放大按钮;点击悬浮窗的固定按钮,当按下按钮不松开且向下移动;触摸事件中的双指放大操作;通过windowManager窗口管理器将放大操作传递给悬浮窗视图,更新悬浮窗视图的大小,达到放大效果;每次放大的动作通过窗口管理器,按比例放大10像素改变悬浮窗视图的长、宽相对于屏幕的坐标参数后,重新绘制视图进行显示;所述悬浮窗视图放大的最大值为屏幕宽;当检测到发生放大变化后,更新视图大小,每次等比例放大的耗时约50毫秒,超出人体的感官能力范围,达到连续的、实时的效果,无闪烁。

[0074] 3) 移动悬浮窗视图手势操作,该操作包括但不限于移动手势,手指按住悬浮窗视图不松开,且移动,此时根据触摸事件,拿到触摸事件的屏幕坐标,分局坐标的变化来计算手指的移动坐标,并通过窗口管理器传递给悬浮窗视图,改变悬浮窗视图相对于屏幕的x、y坐标参数,更新悬浮窗视图的位置,并重新绘制视图,达到移动效果;所述坐标的传递,视图绘制耗时很短,人眼无法分辨,当检测到坐标发生变化后,立刻通过窗口管理器更新视图,达到连续的效果,避免悬浮窗闪烁。

[0075] 综上所述,本实施例所公开的基于Android的视频悬浮播放方法,通过使用悬浮窗使在用户观看视频的同时,能够自由操作手机的目的,不再受限于离开视频播放页面视频关闭的情况,提升用户使用视频播放类App的用户体验;使用系统全局视图,与各应用独立,不会互相影响,在页面或者应用退到后台后,悬浮窗仍然工作,不受影响;采用TextureView视图实现Surface底层图像渲染,不产生闪烁、黑屏现象;实现播放器与悬浮窗的无缝对接,不出现暂停、回退等现象。

[0076] 本领域技术人员可以理解,实现上述实施例方法的全部或部分流程,可以通过计算机程序来指令相关的硬件来完成,所述的程序可存储于计算机可读存储介质中。其中,所述计算机可读存储介质为磁盘、光盘、只读存储记忆体或随机存储记忆体等。

[0077] 以上所述,仅为本发明较佳的具体实施方式,但本发明的保护范围并不局限于此,任何熟悉本技术领域的技术人员在本发明揭露的技术范围内,可轻易想到的变化或替换,都应涵盖在本发明的保护范围之内。

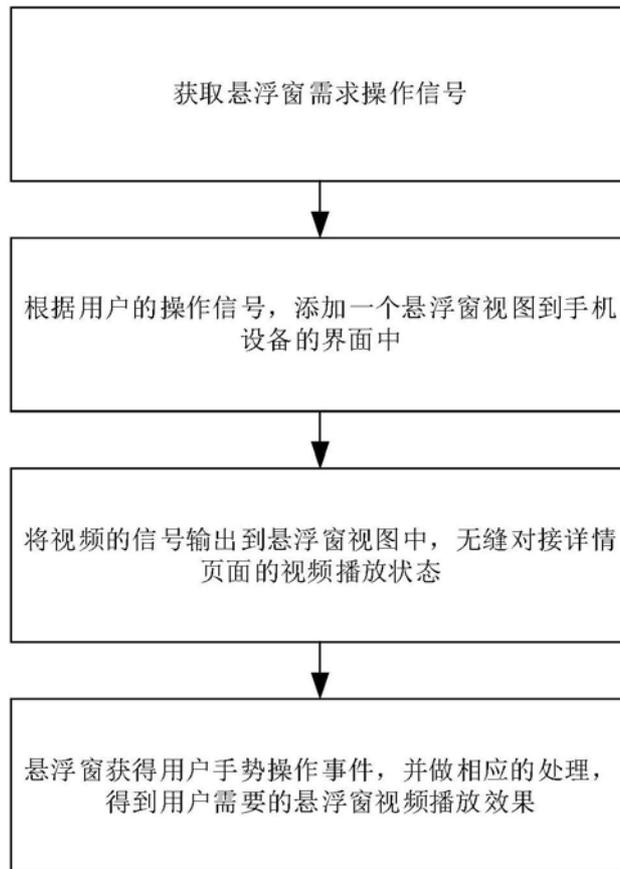


图1