(19)中华人民共和国国家知识产权局

(12)发明专利

(10)授权公告号 CN 104391707 B
(45)授权公告日 2018.01.30

(21)申请号 201410738275.5

(22)申请日 2014.12.05

(65)同一申请的已公布的文献号
申请公布号 CN 104391707 A

(43)申请公布日 2015.03.04

(73)专利权人 上海斐讯数据通信技术有限公司
地址 201616 上海市松江区思贤路3666号

(72)发明人 梁小飞

(74)专利代理机构 杭州千克知识产权代理有限公司 33246
代理人 周希良

(51)Int.Cl.
G06F 9/451(2018.01)
G06F 9/445(2018.01)
G06F 3/0488(2013.01)

(56)对比文件
CN 102880414 A,2013.01.16,
CN 103927028 A,2014.07.16,
US 2012/0084663 A1,2012.04.05,
CN 102880414 A,2013.01.16,
CN 103905653 A,2014.07.02,
US 8577803 B2,2013.11.05,
CN 103793243 A,2014.05.14,

审查员 陈安安

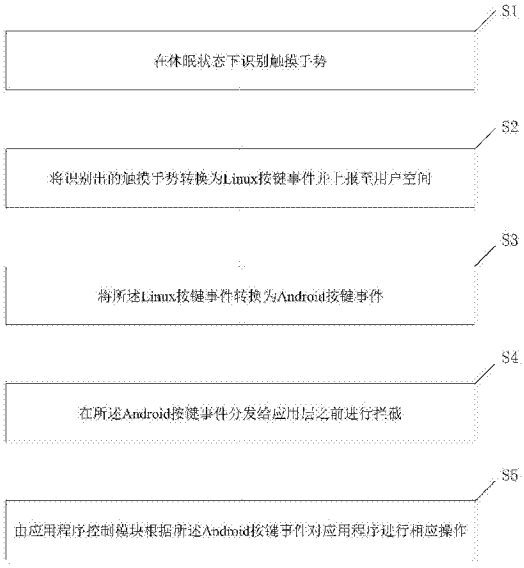
权利要求书2页 说明书6页 附图4页

(54)发明名称

一种应用程序的启动方法及启动装置

(57)摘要

本发明提供一种应用程序的启动方法及应用程序的启动装置。所述应用程序的启动方法包括:在休眠状态下,识别触摸手势;将识别出的触摸手势转换为Linux按键事件并上报至用户空间;将所述Linux按键事件转换为Android按键事件;在所述Android按键事件分发给应用层之前进行拦截;由应用程序控制模块根据所述Android按键事件对应用程序进行相应操作。本发明的应用程序的启动方法及装置在不唤醒触摸屏的情况下可以快速启动应用程序,从而简化了启动流程,提高了启动速率;另外,由于不需要唤醒触摸屏,还可以有效地降低功耗。



CN 104391707 B

1.一种应用程序的启动方法,其特征在于,所述应用程序的启动方法包括:

在休眠状态下,识别触摸手势;

将识别出的触摸手势转换为Linux按键事件并上报至用户空间;

将所述Linux按键事件转换为Android按键事件;

在所述Android按键事件分发给应用层之前进行拦截;

由应用程序控制模块根据所述Android按键事件对应用程序进行相应操作。

2.根据权利要求1所述的应用程序的启动方法,其特征在于:所述在休眠状态下,识别触摸手势的步骤包括:

在触摸屏驱动中修改宏定义;

在休眠状态下,检测触摸屏上的触摸手势;

判断当前触摸屏是否为休眠状态;

当处于休眠状态时,判断所述触摸手势是否为预设手势;

当所述触摸手势为预设手势时,将识别出的触摸手势转换为Linux按键事件并上报至用户空间。

3.根据权利要求1所述的应用程序的启动方法,其特征在于:所述将识别出的触摸手势转换为Linux按键事件并上报至用户空间的步骤包括:

将预设手势和应用程序的按键事件进行定义以实现匹配;

当识别到触摸手势为预设手势时,通过Linux input subsystem将所述预设手势对应的应用程序的按键事件上报至用户空间。

4.根据权利要求1所述的应用程序的启动方法,其特征在于:所述在所述Android按键事件分发给应用层之前进行拦截的步骤包括:在将Android按键事件分发给应用层之前,通过PhoneWindowManager的interceptKeyBeforeQueueing()方法实现拦截。

5.根据权利要求1所述的应用程序的启动方法,其特征在于:在实现拦截之后,使用Androidbrodcast通知应用程序控制模块。

6.一种应用程序的启动装置,其特征在于,所述应用程序的启动装置包括:

手势识别模块,用于在休眠状态下识别触摸手势;

手势转换模块,用于将识别出的触摸手势转换为Linux按键事件并上报至用户空间;

事件转换模块,用于将所述Linux按键事件转换为Android按键事件;

事件拦截模块,用于在所述Android按键事件分发给应用层之前进行拦截;

应用程序控制模块,用于根据所述Android按键事件对应用程序进行相应操作。

7.根据权利要求6所述的应用程序的启动装置,其特征在于:所述手势识别模块包括:

修改单元,用于在触摸屏驱动中修改宏定义;

检测单元,用于在休眠状态下,检测触摸屏上的触摸手势;

状态判断单元,用于判断当前触摸屏是否为休眠状态;

手势判断单元,用于当处于休眠状态时,判断所述触摸手势是否为预设手势;

所述手势转换模块连接所述手势判断单元,用于当所述触摸手势为预设手势时,将识别出的触摸手势转换为Linux按键事件并上报至用户空间。

8.根据权利要求6所述的应用程序的启动装置,其特征在于:所述手势转换模块包括:

定义单元,用于将预设手势和应用程序的按键事件进行定义以实现匹配;

调用单元，用于当识别到触摸手势为预设手势时，调用Linux input subsystem将所述预设手势对应的应用程序的按键事件上报至用户空间。

9.根据权利要求6所述的应用程序的启动装置，其特征在于：所述事件拦截模块用于在将Android按键事件分发给应用层之前，通过PhoneWindowManager的interceptKeyBeforeQueueing()方法实现拦截。

10.根据权利要求6所述的应用程序的启动装置，其特征在于：所述事件拦截模块在实现拦截之后，使用Androidbrodcast通知应用程序控制模块。

一种应用程序的启动方法及启动装置

技术领域

[0001] 本发明涉及通信技术领域，特别是涉及一种应用程序的启动方法及应用程序的启动装置。

背景技术

[0002] 随着技术的发展，移动终端特别是手机被广泛应用。目前大多数手机采用Android系统，并且现有的Android系统手机，基本都是全触屏操作，屏幕是一个基本的触摸输入和图像显示设备。在手机进入休眠状态时，由于屏幕也会一同进入休眠状态，也就是休眠状态。进入休眠状态的屏幕会关闭触摸和显示功能，因此用户无法在休眠状态下通过触摸屏操作手机，也无法看到图像。

[0003] 在现有Android系统中，若想在休眠状态下启动某项应用程序，如打开手电筒时，需要先按下电源键等按键点亮屏幕，解锁后进入Launcher应用，然后找到手电筒应用打开手电筒。也就是说，现有的Android系统，要想在休眠状态下打开手电筒，就需要点亮屏幕、解锁、启动手电筒应用、打开手电筒等操作。但是这种操作比较繁琐，而且点亮屏幕的过程消耗了部分电池电量。

[0004] 因此，如何在休眠状态下简便快捷的启动应用程序就成为本领域技术人员亟待解决的问题之一。

发明内容

[0005] 鉴于以上所述现有技术的缺点，本发明的目的在于提供一种应用程序的启动方法及应用程序的启动装置，用于解决现有技术中启动应用程序繁琐的问题。

[0006] 为实现上述目的及其他相关目的，本发明提供一种应用程序的启动方法，包括：在休眠状态下，识别触摸手势；将识别出的触摸手势转换为Linux按键事件并上报至用户空间；将所述Linux按键事件转换为Android按键事件；在所述Android按键事件分发给应用层之前进行拦截；由应用程序控制模块根据所述Android按键事件对应用程序进行相应操作。

[0007] 优选地，所述在休眠状态下，识别触摸手势的步骤包括：在触摸屏驱动中修改宏定义；在休眠状态下，检测触摸屏上的触摸手势；判断当前触摸屏是否为休眠状态；当处于休眠状态时，判断所述触摸手势是否为预设手势；当所述触摸手势为预设手势时，将识别出的触摸手势转换为Linux按键事件并上报至用户空间。

[0008] 优选的，所述将识别出的触摸手势转换为Linux按键事件并上报至用户空间的步骤包括：

[0009] 将预设手势和应用程序的按键事件进行定义以实现匹配；当识别到触摸手势为预设手势时，通过Linux input subsystem将所述预设手势对应的应用程序的按键事件上报至用户空间。

[0010] 优选的，所述在所述Android按键事件分发给应用层之前进行拦截的步骤包括：在将Android按键事件分发给应用层之前，通过PhoneWindowManager的

interceptKeyBeforeQueueing()方法实现拦截。

[0011] 优选的，在实现拦截之后，使用Androidbrodcast通知应用程序控制模块。

[0012] 相应地，本发明还提供一种应用程序的启动装置，所述应用程序的启动装置包括：手势识别模块，用于在休眠状态下识别触摸手势；手势转换模块，用于将识别出的触摸手势转换为Linux按键事件并上报至用户空间；事件转换模块，用于将所述Linux按键事件转换为Android按键事件；事件拦截模块，用于在所述Android按键事件分发给应用层之前进行拦截；应用程序控制模块，用于根据所述Android按键事件对应用程序进行相应操作。

[0013] 优选的，所述手势识别模块包括：修改单元，用于在触摸屏驱动中修改宏定义；检测单元，用于在休眠状态下，检测触摸屏上的触摸手势；状态判断单元，用于判断当前触摸屏是否为休眠状态；手势判断单元，用于当处于休眠状态时，判断所述触摸手势是否为预设手势；所述手势转换模块连接所述手势判断单元，用于当所述触摸手势为预设手势时，将识别出的触摸手势转换为Linux按键事件并上报至用户空间。

[0014] 优选的所述手势转换模块包括：定义单元，用于将预设手势和应用程序的按键事件进行定义以实现匹配；调用单元，用于当识别到触摸手势为预设手势时，调用Linux input subsystem将所述预设手势对应的应用程序的按键事件上报至用户空间。

[0015] 优选的，所述事件拦截模块用于在将Android按键事件分发给应用层之前，通过PhoneWindowManager的interceptKeyBeforeQueueing()方法实现拦截。

[0016] 优选的，所述事件拦截模块在实现拦截之后，使用Androidbrodcast通知应用程序控制模块。

[0017] 如上所述，本发明的应用程序的启动方法及启动装置，具有以下有益效果：

[0018] 本发明应用程序的启动方法中，在休眠状态下对触摸手势进行识别和转换，并在按键事件分发给应用层之前进行拦截，由应用程序控制模块对应用程序进行相应操作。通过这样的方式，可以在休眠状态下实现应用程序的启动，由于按键事件不会被分发至应用层，因此简化了按键事件的处理流程，从而提高了应用程序的启动速率；

[0019] 进一步地，本发明的应用程序方法，在按键事件分发给应用层之前进行拦截，从而无需唤醒屏幕，从而降低了功耗；

[0020] 另外，在实现按键事件的拦截之后，使用Androidbrodcast通知应用程序控制模块；应用程序控制模块与事件拦截模块相互独立，减少对两模块之间的相互影响，从而方便了功能的维护和扩展。

附图说明

[0021] 图1显示为本发明应用程序的启动方法的流程示意图。

[0022] 图2显示为图1中所示步骤S1的具体实现方式的流程示意图。

[0023] 图3显示为图1中步骤S2的具体实现方式的流程示意图。

[0024] 图4显示为本发明应用程序的启动装置的结构示意图。

[0025] 图5显示为本发明应用程序的启动装置的具体实现方式的结构示意图。

[0026] 元件标号说明

[0027] 10 手势识别模块

[0028] 20 手势转换模块

[0029] 30 事件转换模块

[0030] 40 事件拦截模块

[0031] 50 应用程序控制模块

[0032] 110 修改单元

[0033] 120 检测单元

[0034] 130 状态判断单元

[0035] 140 手势判断单元

[0036] 210 定义单元

[0037] 220 调用单元

[0038] S1～S5 步骤

[0039] S11～S14 步骤

[0040] S21～S22 步骤

具体实施方式

[0041] 正如背景技术中所述的，现有技术中在休眠状态下打开某项应用程序时，需要很繁琐的流程，并且，每次启动应用程序都需要唤醒触摸屏，从而增加功耗。

[0042] 本发明提供一种可以在休眠状态下直接打开应用程序的实现方法，简化了点亮屏幕、解锁、启动应用等操作，不仅简化了应用程序的启动流程，而且本发明的方法不需要唤醒屏幕，从而还可以降低功耗。

[0043] 以下通过特定的具体实例说明本发明的实施方式，本领域技术人员可由本说明书所揭露的内容轻易地了解本发明的其他优点与功效。本发明还可以通过另外不同的具体实施方式加以实施或应用，本说明书中的各项细节也可以基于不同观点与应用，在没有背离本发明的精神下进行各种修饰或改变。需说明的是，在不冲突的情况下，以下实施例及实施例中的特征可以相互组合。

[0044] 需要说明的是，以下实施例中所提供的图示仅以示意方式说明本发明的基本构想，遂图式中仅显示与本发明中有关的组件而非按照实际实施时的组件数目、形状及尺寸绘制，其实际实施时各组件的型态、数量及比例可为一种随意的改变，且其组件布局型态也可能更为复杂。

[0045] 请参阅图1，本发明提供一种应用程序的启动方法，所述应用程序的启动方法包括：

[0046] 步骤S1，在休眠状态下，识别触摸手势；

[0047] 步骤S2，将识别出的触摸手势转换为Linux按键事件并上报至用户空间；

[0048] 步骤S3，将所述Linux按键事件转换为Android按键事件；

[0049] 步骤S4，在所述Android按键事件分发给应用层之前进行拦截；

[0050] 步骤S5，由应用程序控制模块根据所述Android按键事件对应用程序进行相应操作。

[0051] 具体地，为实现触摸屏在休眠状态下的手势检查功能，需要先修改触摸屏驱动中的宏定义，修改好相关宏定义后，在休眠状态下，触摸屏的处理芯片即可实现手势识别功能。此时，当移动终端，如手机，进入休眠状态后，用手指在触摸屏上滑出手势，触摸屏即可

根据硬件支持的手势，匹配识别出所滑的手势，从而完成手势识别功能。

[0052] 对应地，参考图2，所述在休眠状态下，识别触摸手势的步骤可以包括：

[0053] 步骤S11，在触摸屏驱动中修改宏定义；

[0054] 步骤S12，在休眠状态下，检测触摸屏上的触摸手势；

[0055] 步骤S13，判断当前触摸屏是否为休眠状态；

[0056] 当触摸屏不处于休眠状态时，则按照原有的处理方式进行操作。此与现有技术相类似，本发明对此不做赘述。

[0057] 当处于休眠状态时，执行步骤S14，判断所述触摸手势是否为预设手势；

[0058] 当所述触摸手势为预设手势时，执行步骤S2，将识别出的触摸手势转换为Linux按键事件并上报至用户空间。

[0059] 当所述触摸手势不为预设手势时，返回执行步骤S12，继续检测触摸屏上的触摸手势。

[0060] 参考图3，所述将识别出的触摸手势转换为Linux按键事件并上报至用户空间的步骤可以包括：

[0061] 步骤S21，将预设手势和应用程序的按键事件进行定义以实现匹配；

[0062] 步骤S22，当识别到触摸手势为预设手势时，通过Linux input subsystem将所述预设手势对应的应用程序的按键事件上报至用户空间。

[0063] 下面以应用程序为打开手电筒为例对图3所示的工作流程做进一步详细说明。

[0064] 首先，根据图2所示的步骤将触摸屏中的相关宏定义修改好，从而使得触摸屏在休眠状态下仍然可以识别触摸手势。

[0065] 然后，为了实现触摸手势与Linux按键事件的转换，需要定义打开手电筒手势和打开手电筒按键事件。打开手电筒手势从触摸屏支持的手势中选择一个，例如选择一个识别率高且手指容易滑的，如“L”。打开手电筒按键事件是一个自定义的按键事件，因此需要修改input.h，添加一个Linux按键事件，命名为TP_FLASHLIGHT，为避免与已有键码重复，按键码定义可以为现有最后一个键码加1。这样就完成了打开手电筒手势和按键事件的定义。

[0066] 当触摸屏识别到“L”手势时，通过Linux input subsystem，将TP_FLASHLIGHT按键上报到用户空间，实现将手势转换为Linux按键事件。

[0067] 需要说明的是，为了避免其他不需要的手势唤醒系统，触摸屏可以在休眠状态下只处理“L”手势，其他手势不作处理。当然，也可以设置其他的手势与应用程序的匹配，本发明对此不做限制。

[0068] 继续参考图1，所述在所述Android按键事件分发给应用层之前进行拦截的步骤包括：在将Android按键事件分发给应用层之前，通过PhoneWindowManager的interceptKeyBeforeQueueing()方法实现拦截。

[0069] 另外，在实现拦截之后，使用Androidbrodcast通知应用程序控制模块。

[0070] 下面仍以打开手电筒为例对上述步骤S3、步骤S4和步骤S5的工作过程做进一步说明。

[0071] 在具体实施例中，可以由Android framework按键事件处理模块读取上报的Linux按键事件，将Linux按键事件转换为Android按键事件，并在按键事件分发前拦截并通知手电筒控制模块。

[0072] 虽然通过步骤S2上报了TP_FLASHLIGHT按键事件(Linux按键事件),但是Android并不能直接识别处理这个按键事件,需要转换为Android按键事件。

[0073] 具体地,Android input reader读取Linux上报的按键事件时,会根据keyboard layout配置,将Linux按键事件转换为Android按键事件。因此需要在Android input system中添加与Linux按键TP_FLASHLIGHT的对应的Android按键定义,并实现从Linux按键到Android按键的转换。

[0074] 要在Android中添加一个新的按键事件,首先需要在当前设备的keyboard layout文件中添加一个按键定义。keyboard layout定义格式为“key keycode name”,其中“key”关键字代表这是一个按键事件,keycode为linux按键码,name为按键名称,例如“key 500TP_FLASHLIGHT”,代表一个按键,键码为500,键码名称为“TP_FLASHLIGHT”。添加完keyboard layout后,Android就可以识别TP_FLASHLIGHT这个Linux按键事件了。然后要添加Linux按键和Android对应,将Linux按键转换为Android按键事件。

[0075] 定义一个按键对应关系需要添加以下三个部分:

[0076] 在KeycodeLabels.h中添加Linux按键名称对应Android的键码,定义格式为“name keycode”。其中name为Linux按键名称,keycode为Android键码。例如“TP_FLASHLIGHT,300”,代表TP_FLASHLIGHT这个Linux按键名对应的Android按键码为300。

[0077] 在keycodes.h的Android按键事件枚举中添加Android按键事件的定义,例如“AKEYCODE_TP_FLASHLIGHT＝300”,代表Android按键AKEYCODE_TP_FLASHLIGHT的Android按键码为300。

[0078] 在KeyEvent.java中添加Android按键定义、按键码对应的按键名称,这个按键将作为按键事件的标准API,供Android java framework和application使用。例如按键码定义为public static final int KEYCODE_TP_FLASHLIGHT＝300,对应的按键名为“KEYCODE_TP_FLASHLIGHT”。

[0079] 添加完成后,Android input system根据现有的按键处理流程,将Linux按键事件TP_FLASHLIGHT转换为Android按键事件KEYCODE_TP_FLASHLIGHT,并实现按键事件分发。当在休眠状态下,在屏幕上滑出“L”手势时,Android framework和application都可以识别到KEYCODE_TP_FLASHLIGHT这个按键事件。

[0080] 为了减少按键事件的处理,将KEYCODE_TP_FLASHLIGHT这个按键事件(Android按键事件)在分发给application(应用层)前,在framework中通过PhoneWindowManager的interceptKeyBeforeQueueing()方法实现按键事件拦截,并使用Android broadcast通知手电筒控制模块来完成手电筒的操作。

[0081] 手电筒控制模块接收Android framework的手电筒控制事件,完成对手电筒的控制。

[0082] 该模块与Android framework部分使用Android broadcast通信方式,framework发出一个broadcast,手电筒控制模块接收该broadcast。手电筒可以使用照相机的LED闪光灯,通过Android Camera的API可以实现手电筒的打开和关闭功能。当接收到broadcast时,若当前手电筒处于关闭状态,则通过Android Camera API打开手电筒;若当前手电筒处于打开状态,则通过Android Camera API关闭手电筒。

[0083] 本发明应用程序的启动方法,通过修改相应的宏定义来实现休眠状态下的手势识

别功能，经过对手势的处理后在Android按键事件分发给应用层之前实现拦截，这样就不需要唤醒触摸屏，从而可以有效地降低功耗；并且，本发明的方法可以由手势直接在休眠状态下启动相应的应用程序，简化了应用程序的启动流程，提高了启动速率。

[0084] 进一步地，应用程序控制模块与Android framework相互独立，减少了对Android framework的信赖，从而提高了功能的扩展性及便利性。

[0085] 相应地，本发明还提供一种应用程序的启动装置，参考图4，所述应用程序的启动装置可以包括：

[0086] 手势识别模块10，用于在休眠状态下识别触摸手势；

[0087] 手势转换模块20，用于将识别出的触摸手势转换为Linux按键事件并上报至用户空间；

[0088] 事件转换模块30，用于将所述Linux按键事件转换为Android按键事件；

[0089] 事件拦截模块40，用于在所述Android按键事件分发给应用层之前进行拦截；

[0090] 应用程序控制模块50，用于根据所述Android按键事件对应用程序进行相应操作。

[0091] 具体的，参考图5，所述手势识别模块10包括：

[0092] 修改单元110，用于在触摸屏驱动中修改宏定义；

[0093] 检测单元120，用于在休眠状态下，检测触摸屏上的触摸手势；

[0094] 状态判断单元130，用于判断当前触摸屏是否为休眠状态；

[0095] 手势判断单元140，用于当处于休眠状态时，判断所述触摸手势是否为预设手势；

[0096] 所述手势转换模块20连接所述手势判断单元140，用于当所述触摸手势为预设手势时，将识别出的触摸手势转换为Linux按键事件并上报至用户空间。

[0097] 具体的，所述手势转换模块20包括：

[0098] 定义单元210，用于将预设手势和应用程序的按键事件进行定义以实现匹配；

[0099] 调用单元220，连接所述手势判断单元140，用于当识别到触摸手势为预设手势时，调用Linux input subsystem将所述预设手势对应的应用程序的按键事件上报至用户空间。

[0100] 在具体实施例中，所述事件拦截模块40用于在将Android按键事件分发给应用层之前，通过PhoneWindowManager的interceptKeyBeforeQueueing()方法实现拦截。

[0101] 所述事件拦截模块40在实现拦截之后，使用Androidbrodcast通知应用程序控制模块50。

[0102] 本发明应用程序的启动装置的工作过程可参考前述应用程序的启动方法的描述，在此不再赘述。

[0103] 本发明的应用程序的启动装置可以在不唤醒触摸屏的情况下实现应用程序的快速启动，从而提高应用程序的启动速率；另外，由于应用程序启动时不需要唤醒触摸屏，从而还可以降低功耗。

[0104] 上述实施例仅例示性说明本发明的原理及其功效，而非用于限制本发明。任何熟悉此技术的人士皆可在不违背本发明的精神及范畴下，对上述实施例进行修饰或改变。因此，举凡所属技术领域中具有通常知识者在未脱离本发明所揭示的精神与技术思想下所完成的一切等效修饰或改变，仍应由本发明的权利要求所涵盖。

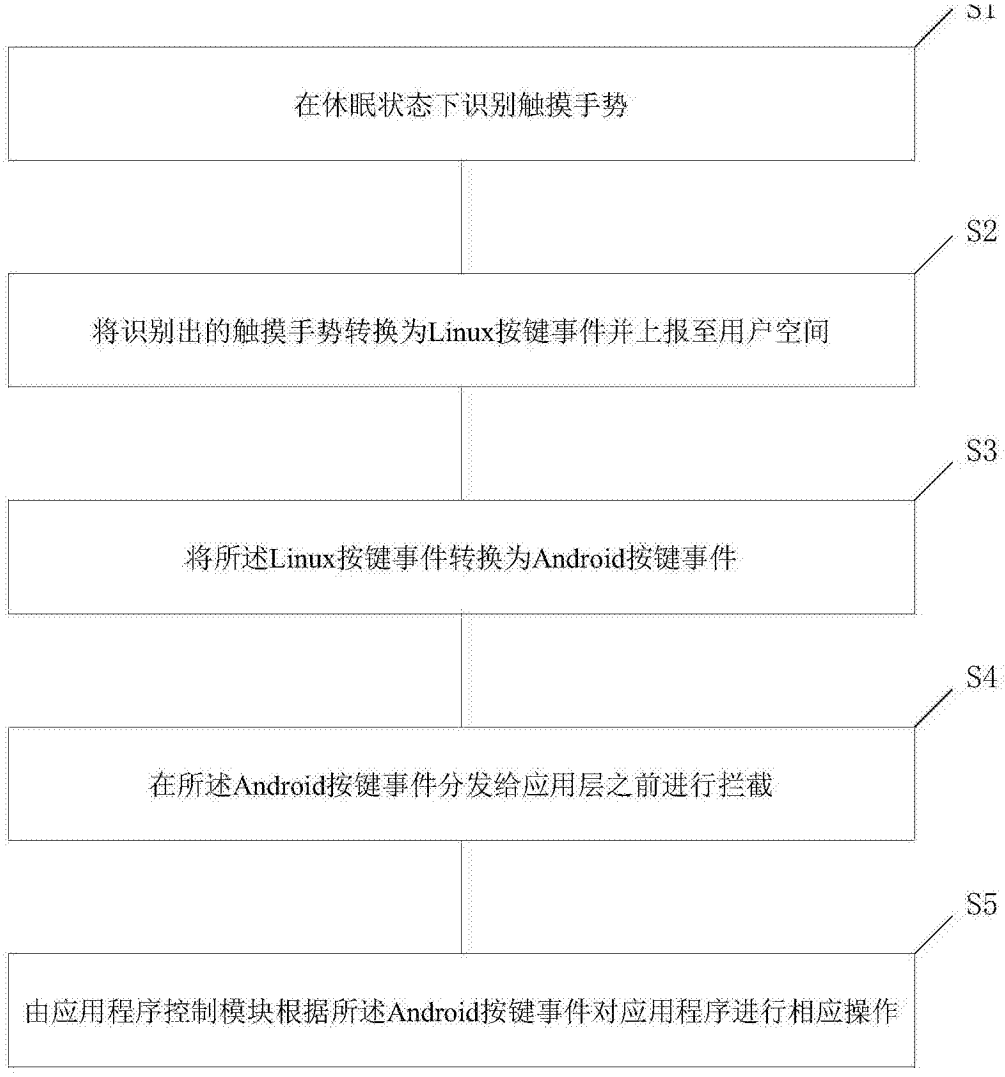

S1
在休眠状态下识别触摸手势
S2
将识别出的触摸手势转换为Linux按键事件并上报至用户空间
S3
将所述Linux按键事件转换为Android按键事件
S4
在所述Android按键事件分发给应用层之前进行拦截
S5
由应用程序控制模块根据所述Android按键事件对应用程序进行相应操作

图1

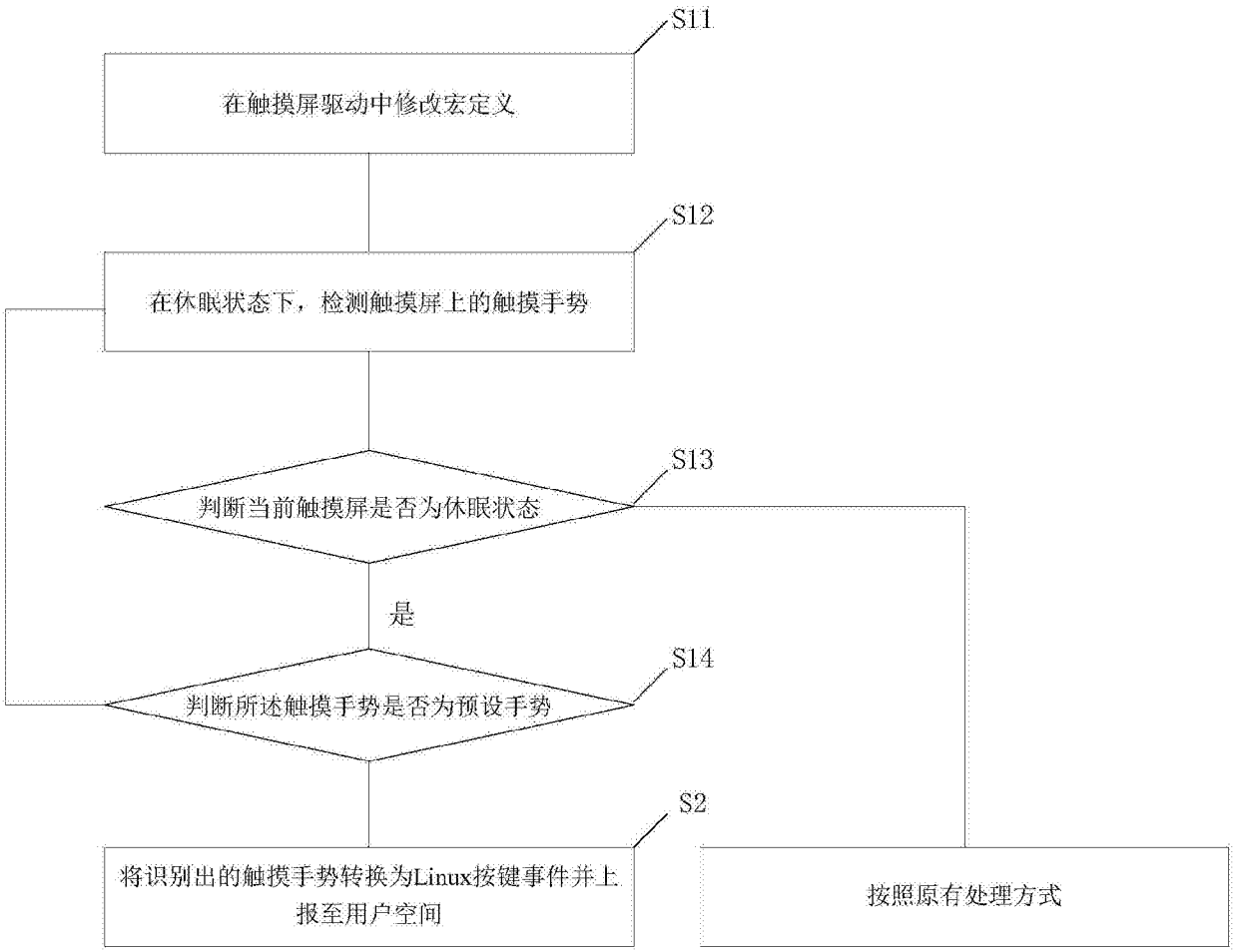

S11
在触摸屏驱动中修改宏定义
S12
在休眠状态下，检测触摸屏上的触摸手势
S13
判断当前触摸屏是否为休眠状态
是
S14
判断所述触摸手势是否为预设手势
S2
将识别出的触摸手势转换为Linux按键事件并上报至用户空间
按照原有处理方式

图2

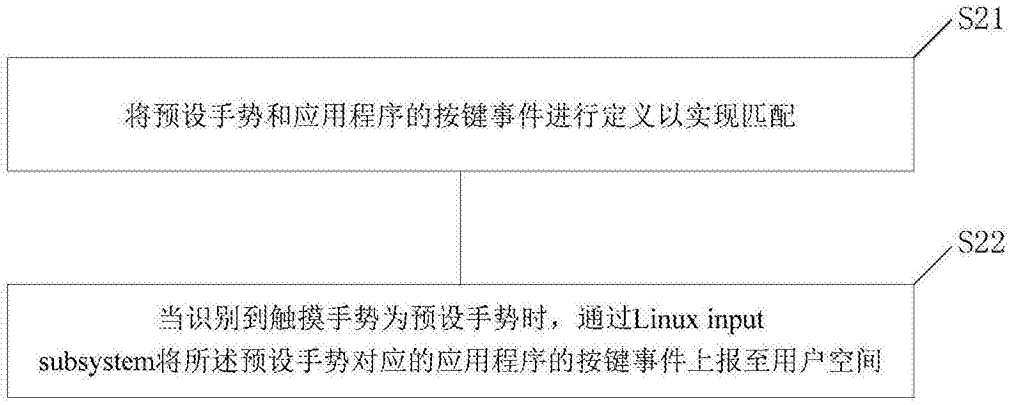

S21
将预设手势和应用程序的按键事件进行定义以实现匹配
S22
当识别到触摸手势为预设手势时，通过Linux input subsystem将所述预设手势对应的应用程序的按键事件上报至用户空间

图3

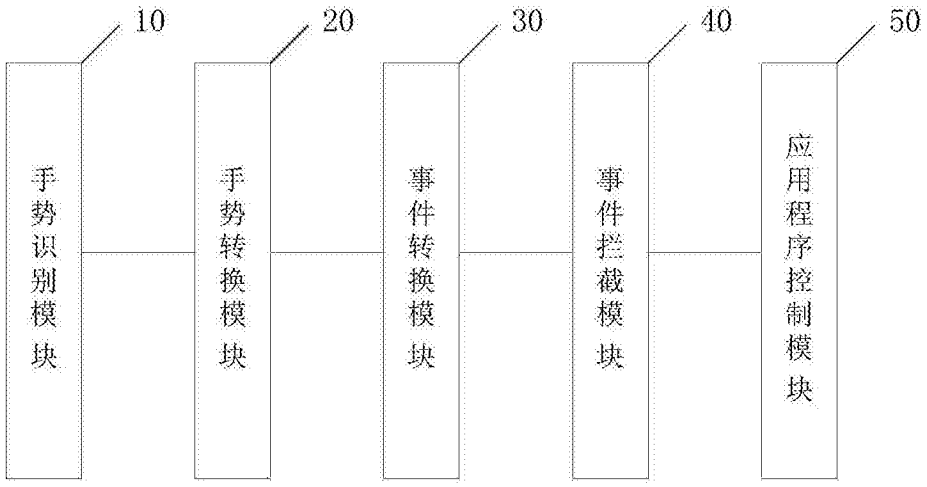


图4

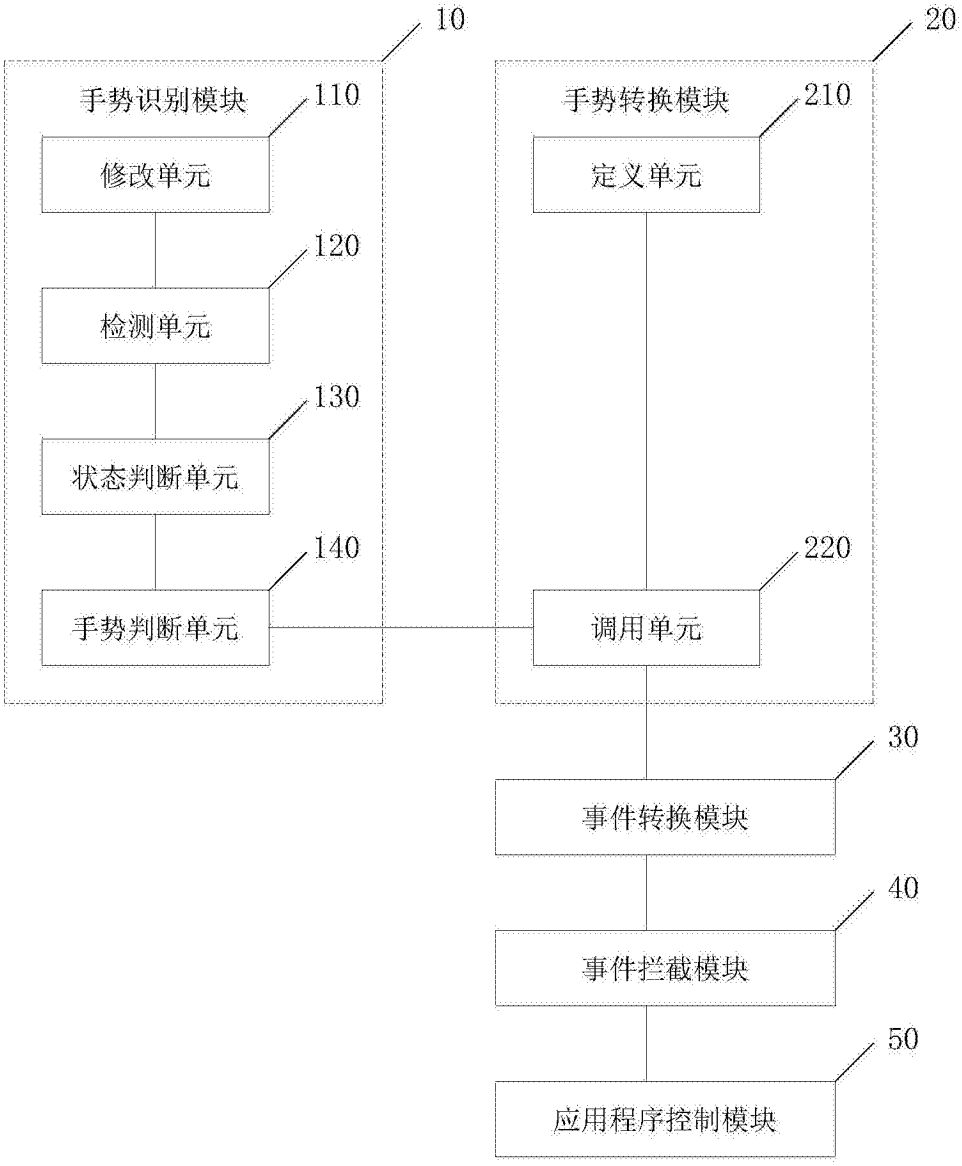

10
手势识别模块
110
修改单元
120
检测单元
130
状态判断单元
140
手势判断单元
20
手势转换模块
210
定义单元
220
调用单元
30
事件转换模块
40
事件拦截模块
50
应用程序控制模块

图5