



(12) 发明专利申请

(10) 申请公布号 CN 105812421 A

(43) 申请公布日 2016. 07. 27

(21) 申请号 201410848406. 5

(22) 申请日 2014. 12. 30

(71) 申请人 TCL 集团股份有限公司
地址 516006 广东省惠州市仲恺高新技术开
发区十九号小区

(72) 发明人 孙向作

(74) 专利代理机构 深圳中一专利商标事务所
44237

代理人 张全文

(51) Int. Cl.
H04L 29/08(2006. 01)

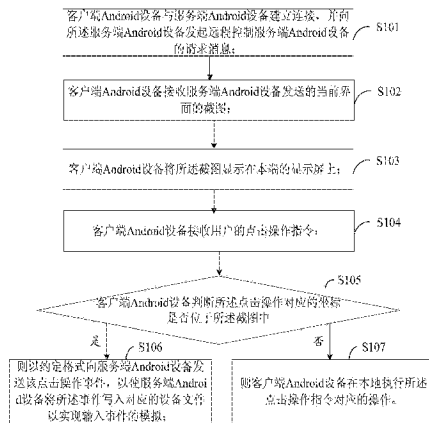
权利要求书3页 说明书9页 附图5页

(54) 发明名称

一种基于 Android 平台的远程控制方法及装置

(57) 摘要

本发明适用于远程控制技术领域, 提供了一种基于 Android 平台的远程控制方法包括: 客户端 Android 设备与服务端 Android 设备建立连接, 并向所述服务端 Android 设备发起远程控制请求消息; 接收服务端 Android 设备发送的当前界面的截图; 将所述截图显示在本端的显示屏上; 接收用户的点击操作指令; 判断所述点击操作对应的坐标是否位于所述截图中; 当判断出所述点击操作对应的坐标位于所述截图中, 则以约定格式向服务端 Android 设备发送该点击操作事件; 当判断出所述点击操作对应的坐标不在所述截图中, 则客户端 Android 设备在本地执行所述点击操作指令对应的操作。



1. 一种基于 Android 平台的远程控制方法,其特征在于,所述方法包括以下步骤:

客户端 Android 设备与服务端 Android 设备建立连接,并向所述服务端 Android 设备发起远程控制服务端 Android 设备的请求消息;

客户端 Android 设备接收服务端 Android 设备发送的当前界面的截图;

客户端 Android 设备将所述截图显示在本端的显示屏上;

客户端 Android 设备接收用户的点击操作指令;

客户端 Android 设备判断所述点击操作对应的坐标是否位于所述截图中;

当判断出所述点击操作对应的坐标位于所述截图中,则以约定格式向服务端 Android 设备发送该点击操作事件,以使服务端 Android 设备将所述事件写入对应的设备文件以实现输入事件的模拟;

当判断出所述点击操作对应的坐标不在所述截图中,则客户端 Android 设备在本地执行所述点击操作指令对应的操作。

2. 如权利要求 1 所述的基于 Android 平台的远程控制方法,其特征在于,在所述客户端 Android 设备与服务端 Android 设备建立连接的步骤之前还包括:

初始化 DisplayHardware 对象,并将所述 DisplayHardware 对象描述的远程桌面、Gralloc 模块与帧缓冲区模块进行绑定;

客户端 Android 设备向服务端 Android 设备发送连接请求消息;

当连接请求通过时,所述 DisplayHardware 对象生效,并向所述服务端 Android 设备发起远程控制服务端 Android 设备的请求消息。

3. 如权利要求 1 所述的基于 Android 平台的远程控制方法,其特征在于,在所述客户端 Android 设备接收用户的点击操作指令的步骤之后,还包括以下步骤:

获取点击屏幕当前坐标值,所述坐标值包括横坐标和纵坐标;

获取所述截图的宽与高;

获取所述截图坐标值,所述坐标值包括横坐标与纵坐标。

4. 如权利要求 3 所述的基于 Android 平台的远程控制方法,其特征在于,所述判断所述点击操作对应的坐标是否位于所述截图中的步骤,具体包括:

判断点击屏幕当前横坐标是否位于所述截图中,判断点击屏幕当前纵坐标是否位于所述截图中;如果判断出击屏幕当前横坐标和纵坐标均位于所述截图中,则认为当前点击屏幕坐标位于远程桌面的截图中;否则,认为当前点击屏幕坐标位于本地桌面。

5. 如权利要求 1 所述的基于 Android 平台的远程控制方法,其特征在于,在所述客户端 Android 设备接收服务端 Android 设备发送的当前界面的截图的步骤之后,还包括:

计算客户端 Android 设备与服务端 Android 设备的屏幕宽和高;

判断服务端 Android 设备屏幕宽度是否大于客户端 Android 设备屏幕宽度,服务端 Android 设备屏幕高度是否大于客户端 Android 设备屏幕高度;

如果服务端 Android 设备屏幕宽度是大于客户端 Android 设备屏幕宽度;和/或者是服务端 Android 设备屏幕高度是大于客户端 Android 设备屏幕高度,对所述截图进行等比例缩放。

6. 一种基于 Android 平台的远程控制装置,其特征在于,所述装置包括:

发起模块,用于客户端 Android 设备与服务端 Android 设备建立连接,并向所述服务端

Android 设备发起远程控制服务端 Android 设备的请求消息；

截图接收模块,用于接收服务端 Android 设备发送的当前界面的截图；

显示模块,用于将所述截图显示在本端的显示屏上；

指令接收模块,用于接收用户的点击操作指令；

判断模块,用于判断所述点击操作对应的坐标是否位于所述截图中；

事件发送模块,用于当判断出所述点击操作对应的坐标位于所述截图中,则以约定格式向服务端 Android 设备发送该点击操作事件,以使服务端 Android 设备将所述事件写入对应的设备文件以实现输入事件的模拟；

执行模块,用于当判断出所述点击操作对应的坐标不在所述截图中,则客户端 Android 设备在本地执行所述点击操作指令对应的操作。

7. 如权利要求6所述的基于 Android 平台的远程控制装置,其特征在于,所述装置还包括：

绑定模块,用于初始化 DisplayHardware 对象,并将所述 DisplayHardware 对象描述的远程桌面、Gralloc 模块与帧缓冲区模块进行绑定；

请求消息发送模块,用于向服务端 Android 设备发送连接请求消息；

所述发起模块,具体用于当连接请求通过时,所述 DisplayHardware 对象生效,并向所述服务端 Android 设备发起远程控制服务端 Android 设备的请求消息。

8. 如权利要求6所述的基于 Android 平台的远程控制装置,其特征在于,所述装置还包括：

计算模块,用于计算客户端 Android 设备与服务端 Android 设备的屏幕宽和高；

宽高判断模块,用于判断服务端 Android 设备屏幕宽度是否大于客户端 Android 设备屏幕宽度,以及判断服务端 Android 设备屏幕高度是否大于客户端 Android 设备屏幕高度；

缩放模块,用于如果服务端 Android 设备屏幕宽度是大于客户端 Android 设备屏幕宽度;和/或者如果服务端 Android 设备屏幕高度是大于客户端 Android 设备屏幕高度,对所述截图进行等比例缩放。

9. 一种基于 Android 平台的远程控制方法,其特征在于,所述方法包括以下步骤：

服务端 Android 设备接收客户端 Android 设备发起的远程控制服务端 Android 设备的请求消息；

服务端 Android 设备接受请求后,将其当前界面不断的截图后发送到客户端 Android 设备；

服务端 Android 设备接收客户端 Android 设备以约定格式发送的点击操作事件；

服务端 Android 设备将所述事件写入对应的设备文件以实现输入事件的模拟。

10. 如权利要求9所述的基于 Android 平台的远程控制方法,其特征在于,在所述服务端 Android 设备接收客户端 Android 设备发起的远程控制服务端 Android 设备的请求消息的步骤之前,还包括：

服务端 Android 设备接收客户端 Android 设备发送的连接请求消息；

服务端 Android 设备验证所述连接请求消息,若验证成功,则向客户端 Android 设备发送连接请求通过消息；

在所述服务端 Android 设备接受请求后,将其当前界面不断的截图后,还包括:
将图片格式的所述当前界面转换为视频流发送给客户端 Android 设备。

11. 一种基于 Android 平台的远程控制装置,其特征在于,所述装置包括:

控制请求接收模块,用于接收客户端 Android 设备发起的远程控制服务端 Android 设备的请求消息;

截图模块,用于服务端 Android 设备接受请求后,将其当前界面不断的截图后发送到客户端 Android 设备;

事件接收模块,用于接收客户端 Android 设备以约定格式发送的点击操作事件;

写入模块,用于将所述事件写入对应的设备文件以实现输入事件的模拟。

12. 如权利要求 11 所述的基于 Android 平台的远程控制装置,其特征在于,所述装置还包括:

连接请求接收模块,用于接收客户端 Android 设备发送的连接请求消息;

验证模块,用于验证所述连接请求消息,若验证成功,则向客户端 Android 设备发送连接请求通过消息;

转换模块,用于将图片格式的所述当前界面转换为视频流发送给客户端 Android 设备。

一种基于 Android 平台的远程控制方法及装置

技术领域

[0001] 本发明属于远程控制技术领域,尤其涉及一种基于 Android 平台的远程控制方法及装置。

背景技术

[0002] 众所周知,目前,Android 已经成为最具吸引力的开源操作系统之一,从手机到平板电脑,从智能电视到工业医疗领域,都能看见 Android 的身影。随着 Android 成为主流操作系统,人们对 Android 设备进行远程控制的需求也日益增多。Windows 等主流操作系统都有成熟的远程桌面软件,但 Android 系统至今没有成熟的远程桌面方案,从而给用户远程控制带来不便的问题。

[0003] 鉴于上述现状,有待于提出一种针对 Android 平台的远程控制方案。

发明内容

[0004] 本发明的目的在于提供一种基于 Android 平台的远程控制方法及装置,旨在解决的 Android 系统至今没有成熟的远程桌面方案,从而给用户远程控制带来不便的问题。

[0005] 第一方面,本发明提供了一种基于 Android 平台的远程控制方法,所述方法包括以下步骤:

[0006] 客户端 Android 设备与服务端 Android 设备建立连接,并向所述服务端 Android 设备发起远程控制服务端 Android 设备的请求消息;

[0007] 客户端 Android 设备接收服务端 Android 设备发送的当前界面的截图;

[0008] 客户端 Android 设备将所述截图显示在本端的显示屏上;

[0009] 客户端 Android 设备接收用户的点击操作指令;

[0010] 客户端 Android 设备判断所述点击操作对应的坐标是否位于所述截图中;

[0011] 当判断出所述点击操作对应的坐标位于所述截图中,则以约定格式向服务端 Android 设备发送该点击操作事件,以使服务端 Android 设备将所述事件写入对应的设备文件以实现输入事件的模拟;

[0012] 当判断出所述点击操作对应的坐标不在所述截图中,则客户端 Android 设备在本地执行所述点击操作指令对应的操作。

[0013] 第二方面,本发明提供了一种基于 Android 平台的远程控制装置,所述装置包括:

[0014] 发起模块,用于客户端 Android 设备与服务端 Android 设备建立连接,并向所述服务端 Android 设备发起远程控制服务端 Android 设备的请求消息;

[0015] 截图接收模块,用于接收服务端 Android 设备发送的当前界面的截图;

[0016] 显示模块,用于将所述截图显示在本端的显示屏上;

[0017] 指令接收模块,用于接收用户的点击操作指令;

[0018] 判断模块,用于判断所述点击操作对应的坐标是否位于所述截图中;

[0019] 事件发送模块,用于当判断出所述点击操作对应的坐标位于所述截图中,则以约

定格式向服务端 Android 设备发送该点击操作事件,以使服务端 Android 设备将所述事件写入对应的设备文件以实现输入事件的模拟;

[0020] 执行模块,用于当判断出所述点击操作对应的坐标不在所述截图中,则客户端 Android 设备在本地执行所述点击操作指令对应的操作。

[0021] 第三方面,本发明提供了一种基于 Android 平台的远程控制方法,所述方法包括以下步骤:

[0022] 服务端 Android 设备接收客户端 Android 设备发起的远程控制服务端 Android 设备的请求消息;

[0023] 服务端 Android 设备接受请求后,将其当前界面不断的截图后发送到客户端 Android 设备;

[0024] 服务端 Android 设备接收客户端 Android 设备以约定格式发送的点击操作事件;

[0025] 服务端 Android 设备将所述事件写入对应的设备文件以实现输入事件的模拟。

[0026] 第四方面,本发明提供了一种基于 Android 平台的远程控制装置,所述装置包括:

[0027] 控制请求接收模块,用于接收客户端 Android 设备发起的远程控制服务端 Android 设备的请求消息;

[0028] 截图模块,用于服务端 Android 设备接受请求后,将其当前界面不断的截图后发送到客户端 Android 设备;

[0029] 事件接收模块,用于接收客户端 Android 设备以约定格式发送的点击操作事件;

[0030] 写入模块,用于将所述事件写入对应的设备文件以实现输入事件的模拟。

[0031] 在本发明中,客户端 Android 设备与服务端 Android 设备建立连接,并向所述服务端 Android 设备发起远程控制请求消息;接收服务端 Android 设备发送的当前界面的截图;将所述截图显示在本端的显示屏上;接收用户的点击操作指令;判断所述点击操作对应的坐标是否位于所述截图中;当判断出所述点击操作对应的坐标位于所述截图中,则以约定格式向服务端 Android 设备发送该点击操作事件,以使服务端 Android 设备将所述事件写入对应的设备文件以实现输入事件的模拟;当判断出所述点击操作对应的坐标不在所述截图中,则客户端 Android 设备在本地执行所述点击操作指令对应的操作。本发明可实现任意两种基于 Android 平台的设备可互相对彼此进行远程桌面控制。

附图说明

[0032] 图 1 是本发明实施例提供的客户端侧基于 Android 平台的远程控制方法的实现流程示意图;

[0033] 图 2 是本发明另一实施例提供的客户端侧基于 Android 平台的远程控制方法的实现流程示意图;

[0034] 图 3 是本发明实施例提供的服务端侧基于 Android 平台的远程控制方法的实现流程示意图;

[0035] 图 4 是本发明实施例提供的客户端侧基于 Android 平台的远程控制装置的结构示意图;

[0036] 图 5 是本发明另一实施例提供的客户端侧基于 Android 平台的远程控制装置的结构示意图;

[0037] 图6是本发明实施例提供的服务端侧基于Android平台的远程控制装置的结构示意图。

具体实施方式

[0038] 为了使本发明的目的、技术方案及有益效果更加清楚明白,以下结合附图及实施例,对本发明进行进一步详细说明。应当理解,此处所描述的具体实施例仅仅用以解释本发明,并不用于限定本发明。

[0039] 在本发明中,通过在Android设备中建立DisplayHardware对象、Gralloc模块和帧缓冲区模块;并将所述DisplayHardware对象、Gralloc模块和帧缓冲区模块绑定,共同完成在客户端中显示远程桌面。本发明可实现任意两种基于Android平台的设备可互相对彼此进行远程桌面控制。

[0040] 为了说明本发明所述的技术方案,下面通过具体实施例来进行说明。

[0041] 请参阅图1,为本发明实施例提供的客户端侧基于Android平台的远程控制方法的实现流程,其包括以下步骤:

[0042] 在步骤S101中,客户端Android设备与服务端Android设备建立连接,并向所述服务端Android设备发起远程控制服务端Android设备的请求消息;

[0043] 在步骤S102中,客户端Android设备接收服务端Android设备发送的当前界面的截图;

[0044] 在步骤S103中,客户端Android设备将所述截图显示在本端的显示屏上;

[0045] 在步骤S104中,客户端Android设备接收用户的点击操作指令;

[0046] 在本发明实施例中,通过MotionEvent.getRawX()和MotionEvent.getRawY()获取点击屏幕当前坐标值,即横坐标EventX与纵坐标Eventy;通过Bitmap.getWidth()和Bitmap.getHeight()获取截图的宽width与高height;通过View.getTop()和View.getLeft()获取所述截图坐标值,即横坐标PicX与纵坐标PicY。

[0047] 在步骤S105中,客户端Android设备判断所述点击操作对应的坐标是否位于所述截图中;

[0048] 在本发明实施例中,判断点击屏幕当前横坐标是否位于所述截图中,即判断 $PicX < EventX < PicX + width$,判断点击屏幕当前纵坐标是否位于所述截图中,即 $PicY < EventY < PicY + height$;如果判断出击屏幕当前横坐标和纵坐标均位于所述截图中,则认为当前点击屏幕坐标位于远程桌面;否则,认为当前点击屏幕坐标位于本地桌面。

[0049] 在步骤S106中,当判断出所述点击操作对应的坐标位于所述截图中,则以约定格式向服务端Android设备发送该点击操作事件,以使服务端Android设备将所述事件写入对应的设备文件以实现输入事件的模拟;

[0050] 在步骤S107中,当判断出所述点击操作对应的坐标不在所述截图中,则客户端Android设备在本地执行所述点击操作指令对应的操作。

[0051] 在本发明实施例中,请参阅图2,在步骤S101之前,还包括以下步骤:

[0052] 在步骤S1011中,初始化DisplayHardware对象,并将所述DisplayHardware对象描述的远程桌面、Gralloc模块与帧缓冲区模块进行绑定;

[0053] 在本发明实施例中,在客户端Android设备中,建立DisplayHardware对象、

Gralloc 模块和帧缓冲区模块 ;其中, DisplayHardware 对象用于描述客户端 Android 设备显示屏的对象, Gralloc 模块是 Android 系统中一种硬件抽象层, 其封装了对显示屏的所有访问操作, 帧缓冲区模块用于在客户端 Android 设备的显示屏中显示远程桌面。

[0054] 在本发明实施例中, 在 Android 系统中, 把对硬件的支持分成两层, 一层是用户空间, 另一层是内核空间, 内核驱动层运行在内核空间, 硬件抽象层 (即 HAL) 运行在用户空间。硬件抽象层, 简单来说, 就是对 Linux 内核驱动层程序的封装, 向上层提供接口, 屏蔽底层的实现细节。

[0055] Android 设备中的显示屏被抽象成一个帧缓冲区 (FrameBuffer), 而 gralloc 模块则封装了对帧缓冲区的所有访问操作。用户空间的应用程序在使用帧缓冲区之前, 首先要加载 gralloc 模块, 其次获得一个 gralloc 设备和一个 fb 设备 (帧缓冲区设备)。gralloc 设备用来分配图形缓冲区, fb 设备用来将图形缓冲区渲染到帧缓冲区去。

[0056] 有了 gralloc 设备之后, 用户空间中的应用程序就可以申请分配一块图形缓冲区, 并且将这块图形缓冲区映射到应用程序的地址空间来, 以便可以向里面写入要绘制的画面的内容。最后, 用户空间中的应用程序就通过 fb 设备来将前面已经准备好了的图形缓冲区渲染到帧缓冲区中去, 即将图形缓冲区的内容绘制到显示屏中去, 实现显示功能。相应地, 当用户空间中的应用程序不再需要使用一块图形缓冲区的时候, 就可以通过 gralloc 设备来释放它, 并且将它从地址空间中解除映射, 从而不再实现显示功能。这便是 android 系统中控制显示屏显示的基本原理。

[0057] 在步骤 S1012 中, 客户端 Android 设备向服务端 Android 设备发送连接请求消息 ;

[0058] 在本发明实施例中, 客户端 Android 设备向服务端 Android 设备发送连接请求消息, 其中, 所述连接请求消息中携带有服务端 Android 设备的 IP、端口号和密码。

[0059] 在步骤 S1013 中, 当连接请求通过时, 所述 DisplayHardware 对象生效, 客户端 Android 设备向所述服务端 Android 设备发起远程控制服务端 Android 设备的请求消息。

[0060] 在本发明实施例中, 当密码错误时, 客户端 Android 设备接收服务端 Android 设备返回的密码错误消息, SurfaceFlinger 不使能 DisplayHardware 对象, 其中, SurfaceFlinger 用于管理帧缓冲区模块的服务。当密码正确时, 客户端 Android 设备接收服务端 Android 设备返回的密码正确消息, SurfaceFlinger 使能 DisplayHardware 对象。

[0061] 在本发明实施例中, 客户端 Android 设备接收服务端 Android 设备返回的当前界面的截图对应的视频信息 ;客户端 Android 设备对接收到的所述视频信息进行解码, 得到相应的截图 ;客户端 Android 设备将所述截图保存在所述帧缓冲区模块中 ;

[0062] 客户端 Android 设备从所述帧缓冲区模块中读取所述截图, 并在其显示屏上显示所述服务端 Android 设备的所述截图。

[0063] 在本发明实施例中, 利用 ffmpeg 命令对所述视频信息进行解码, 即将视频信息转换为截图。其中, ffmpeg 是一个非常强大的工具, 使用它可以转换视频格式, 包括视频容器格式, 视频编码格式。可以更改视频的其他的参数, 比如帧率 - 每秒播放的帧数, 视频的大小、尺寸等等。还可以将图片转换视频。本实施例中, 采用 ffmpeg 工具实现视频与图片的转化。

[0064] 作为本发明一优选实施例, 在得到相应的截图后, 对所述截图进行预处理, 具体为 : 计算客户端 Android 设备与服务端 Android 设备的屏幕宽和高, 判断服务端 Android 设

备屏幕宽度是否大于客户端 Android 设备屏幕宽度,服务端 Android 设备屏幕高度是否大于客户端 Android 设备屏幕高度,如果其中一个条件满足(即服务端 Android 设备屏幕宽度是大于客户端 Android 设备屏幕宽度;或者是服务端 Android 设备屏幕高度是大于客户端 Android 设备屏幕高度),对所述截图进行等比例缩放,使得远程桌面宽度和高度小于客户端 Android 设备屏幕的宽和高。

[0065] 在本发明实施例中,由于 Android 系统中 SurfaceFlinger 服务(用于管理系统帧缓冲区设备的服务)可以同时支持多个 DisplayHardware,即存在同时支持多个显示屏的可能。当客户端与服务端连接时,客户端除了显示本地桌面之外,还显示远程桌面。因此,在系统增加一个 DisplayHardware 对象,称为 DisplayHardware1,用来实现远程桌面的显示。而系统中原来的 DisplayHardware 对象称为 DisplayHardware0,用来实现本地桌面的显示。

[0066] 另外,将系统自带的 Gralloc 模块,称为 Gralloc0 模块,系统自带的帧缓冲区设备(即 fb 设备),称为 fb0 设备,并将 Gralloc0、fb0 与 DisplayHardware0 绑定,共同完成客户端本地桌面的显示。而且,在该系统中新增一个 Gralloc 模块,称为 Gralloc1 模块,新增一个帧缓冲区设备,称为 fb1 设备,并将 Gralloc1、fb1 与 DisplayHardware1 绑定,共同完成客户端中远程桌面的显示。

[0067] 客户端启动时,初始化了 DisplayHardware0 对象和 DisplayHardware1 对象,并将 DisplayHardware0 对象描述的本地桌面与 gralloc0 设备绑定,DisplayHardware1 对象描述的远程桌面与 gralloc1 设备绑定,且打开了 gralloc0 设备、fb0 设备、gralloc1 设备和 fb1 设备。

[0068] 请参阅图 3,为本发明实施例提供的服务端侧基于 Android 平台的远程控制方法的实现流程,其包括以下步骤:

[0069] 在步骤 S301 中,服务端 Android 设备接收客户端 Android 设备发起的远程控制服务端 Android 设备的请求消息;

[0070] 在本发明实施例中,在所述服务端 Android 设备接收客户端 Android 设备发起的远程控制服务端 Android 设备的请求消息的步骤之前,还包括:

[0071] 服务端 Android 设备接收客户端 Android 设备发送的连接请求消息;服务端 Android 设备验证所述连接请求消息,若验证成功,则向客户端 Android 设备发送连接请求通过消息;

[0072] 在步骤 S302 中,服务端 Android 设备接受请求后,将其当前界面不断的截图后发送到客户端 Android 设备;

[0073] 在本发明实施例中,在所述服务端 Android 设备接受请求后,将其当前界面不断的截图后,还包括:将图片格式的所述当前界面转换为视频流发送给客户端 Android 设备。

[0074] 在步骤 S303 中,服务端 Android 设备接收客户端 Android 设备以约定格式发送的点击操作事件;

[0075] 在步骤 S304 中,服务端 Android 设备将所述事件写入对应的设备文件以实现输入事件的模拟。

[0076] 在本发明实施例中,服务端 Android 设备截取当前界面,具体包括:读取 framebuffer,即读取服务端 Android 设备中的 /dev/graphics/fbx 设备文件;将读取的

数据转换为 bitmap 文件 ;再将 bitmap 文件转换为图像文件,将生成的图像文件保存在 /data/local/tmp/ 下。

[0077] 然而,可以理解的是,Android 是基于 linux 派生出来的,Android 的显示机制是与 Linux 一样的机制 :Framebuffer。Framebuffer 提供接口将显示屏抽象为帧缓冲区。用户可以将它看成是显示内存的一个映像,将其映射到进程地址空间之后,便可以直接进行读写操作,这种机制是把屏幕上的每个点映射成一段线性内存空间,程序可以简单的改变这段内存的值来改变屏幕上某一点的颜色。Framebuffer 属于标准的字符设备,就是说它以字符流的方式进行读写。Android 系统中 Framebuffer 的设备名称一般为 /dev/graphics/fbx。

[0078] 作为本发明一优选实施例,在整个系统实现中,截取当前界面是不断进行的。为了达到很好的显示效果,采用服务端 Android 设备主动更新策略。在 Android 系统 shell 终端 /proc/sys/fs/ 目录下,存在 inotify 目录,如下所示 :

[0079] root@android:/proc/sys/fs/inotify#ls-l

[0080] -rw-r--r--root root 0 2014-10-14 13:55max_queued_events

[0081] -rw-r--r--root root 0 2014-10-14 13:55max_user_instances

[0082] -rw-r--r--root root 0 2014-10-14 13:55max_user_watches

[0083] 为了对截取当前界面产生的图片文件进行监控,本文使用 Linux 内核的 inotify 机制。Inotify 是基于 inode 级别的文件系统监控技术,是一种强大的、细粒度的、异步的机制,它满足各种各样的文件监控需要,不仅限于安全和性能。Inotify 既可以监视文件,也可以监视目录。Inotify 使用系统调用来通知文件系统事件。Inotify 使用文件描述符作为接口,因而可以使用通常的文件 I/O 操作 select 和 poll 来监视文件系统的变化。本文采用 poll 来监视截取当前界面产生的图片,具体实现方案如下 :获取监视文件描述符 ;监视文件 IN_CLOSE_WRITE(可写文件被 close) 状态 ;poll 监视文件是否可读 ;如果可读,将监视事件从事件队列中移除,否则跳转到所述 poll 监视文件是否可读的步骤。

[0084] 作为本发明另一实施例,服务端 Android 设备接收客户端 Android 设备以约定格式发送过来的点击操作事件,服务端 Android 设备将所述事件写入对应的设备文件以实现输入事件的模拟。

[0085] 在本发明实施例中,所述约定格式主要包括键盘事件和触摸屏事件结构体的定义,具体如下 :

[0086] 键盘事件结构体定义

[0087] 在 Android 系统的 shell 终端 /system/usr/keylayout 目录下有一个 qwerty.kl 的文件,如下所示。文件中有字母和各个功能键的 Android 键值表,本系统实现 Android 原生系统中所有按键的模拟。

[0088] root@android:/system/usr/keylayout#ls-l

[0089] -rw-rw-r--1034103425772014-06-1711:05qwerty.kl

[0090] 键盘事件结构体定义主要包括 Android 键码字段、判断按下与松开值字段、事件类型字段和事件代码字段。

[0091] 触摸屏事件结构体定义

[0092] 触摸屏按下和松开在客户端 Android 设备对应的是屏幕的按下和松开。若按下屏

幕未松开,则一直发送屏幕当前的坐标给服务端 Android 设备,服务端 Android 设备未收到屏幕松开消息,则认为触摸屏上产生了拖动事件。

[0093] 触摸屏事件结构体定义主要包括横坐标 X 字段、纵坐标 Y 字段、判断按下与松开值字段、事件类型字段和事件代码字段。

[0094] 服务端 Android 设备模拟输入事件的实现,具体实现方式如下:

[0095] Android 系统中的输入子系统是所有 I/O 设备驱动的中间层,它为上层提供了一个统一的接口。设备驱动程序将检测到的事件都上报给输入子系统,这样在应用程序中,不需要去管键盘和鼠标的型号和数量,只要从输入子系统提供的设备节点文件中读取事件缓存则可。此外,事件处理层和应用程序层之间的关联是通过设备缓存区来实现的。设备缓存区以下是系统内核完成的任务,因此,如果要模拟输入事件,只要仿照事件处理层向设备缓存区写入事件的方法,直接将 `input_event` 结构体写入设备文件节点则可。

[0096] 按键事件的模拟如下:

[0097] 当服务端 Android 设备收到客户端 Android 设备发送的点击操作事件请求后,使用 `open` 函数打开键盘设备,在收到的 `key_event` 包中将 `code` 和 `value` 的值取出,填充到新的 `input_event` 里面,调用 `write(key_fd, &key_event, sizeof(key_event))` 函数,按照事件的发生顺序写入键盘设备文件 `/dev/input/eventx` 中。

[0098] 触摸屏事件的模拟具体如下:打开触摸屏设备;获取触摸屏参数;监测是否收到触摸屏事件;如果是,将 `mouse_event` 结构体写入设备节点文件 `/dev/input/mousex` 中;否则跳转到所述监测是否收到触摸屏事件的步骤中;监测是否收到松开指令;如果是,写入释放事件到设备节点文件 `/dev/input/mousex` 中;否则跳转到所述监测是否收到松开指令的步骤中。

[0099] 请参阅图 4,为本发明实施例提供的客户端侧基于 Android 平台的远程控制装置的结构。为了便于说明,仅示出了与本发明实施例相关的部分。所述基于 Android 平台的远程控制装置包括:发起模块 101、截图接收模块 102、显示模块 103、指令接收模块 104、判断模块 105、事件发送模块 106、以及执行模块 107。所述基于 Android 平台的远程控制装置可以是内置于客户端 Android 设备中的软件单元、硬件单元或者是软硬件结合的单元。

[0100] 发起模块 101,用于客户端 Android 设备与服务端 Android 设备建立连接,并向所述服务端 Android 设备发起远程控制服务端 Android 设备的请求消息;

[0101] 截图接收模块 102,用于接收服务端 Android 设备发送的当前界面的截图;

[0102] 显示模块 103,用于将所述截图显示在本端的显示屏上;

[0103] 指令接收模块 104,用于接收用户的点击操作指令;

[0104] 在本发明实施例中,通过 `MotionEvent.getRawX()` 和 `MotionEvent.getRawY()` 获取点击屏幕当前坐标值,即横坐标 `EventX` 与纵坐标 `Eventy`;通过 `Bitmap.getWidth()` 和 `Bitmap.getHeight()` 获取截图的宽 `width` 与高 `height`;通过 `View.getTop()` 和 `View.getLeft()` 获取所述截图坐标值,即横坐标 `PicX` 与纵坐标 `PicY`。

[0105] 判断模块 105,用于判断所述点击操作对应的坐标是否位于所述截图中;

[0106] 在本发明实施例中,所述判断模块 105,具体用于判断点击屏幕当前横坐标是否位于所述截图中,即判断 `PicX < EventX < PicX + width`,以及判断点击屏幕当前纵坐标是否位于所述截图中,即 `PicY < EventY < PicY + height`;如果判断出击屏幕当前横坐标和纵坐标均位

于所述截图中,则认为当前点击屏幕坐标位于远程桌面;否则,认为当前点击屏幕坐标位于本地桌面。

[0107] 事件发送模块 106,用于当判断出所述点击操作对应的坐标位于所述截图中,则以约定格式向服务端 Android 设备发送该点击操作事件,以使服务端 Android 设备将所述事件写入对应的设备文件以实现输入事件的模拟;

[0108] 执行模块 107,用于当判断出所述点击操作对应的坐标不在所述截图中,则客户端 Android 设备在本地执行所述点击操作指令对应的操作。

[0109] 在本发明实施例中,请参阅图 5,所述基于 Android 平台的远程控制装置还包括:绑定模块 201、以及请求消息发送模块 202。

[0110] 绑定模块 201,用于初始化 DisplayHardware 对象,并将所述 DisplayHardware 对象描述的远程桌面、Gralloc 模块与帧缓冲区模块进行绑定;

[0111] 请求消息发送模块 202,用于向服务端 Android 设备发送连接请求消息;

[0112] 所述发起模块 101,具体用于当连接请求通过时,所述 DisplayHardware 对象生效,并向所述服务端 Android 设备发起远程控制服务端 Android 设备的请求消息。

[0113] 作为本发明一优选实施例,所述基于 Android 平台的远程控制装置还包括:图片处理模块。

[0114] 所述图片处理模块,用于在得到相应的截图后,对所述截图进行预处理。其中,所述图片处理模块具体包括:计算模块、宽高判断模块、缩放模块。计算模块,用于计算客户端 Android 设备与服务端 Android 设备的屏幕宽和高。宽高判断模块,用于判断服务端 Android 设备屏幕宽度是否大于客户端 Android 设备屏幕宽度,服务端 Android 设备屏幕高度是否大于客户端 Android 设备屏幕高度;缩放模块,用于如果其中一个条件满足(即服务端 Android 设备屏幕宽度是大于客户端 Android 设备屏幕宽度;或者是服务端 Android 设备屏幕高度是大于客户端 Android 设备屏幕高度),对所述截图进行等比例缩放,使得远程桌面宽度和高度小于客户端 Android 设备屏幕的宽和高。

[0115] 请参阅图 6,为本发明实施例提供的服务端侧基于 Android 平台的远程控制装置的结构。为了便于说明,仅示出了与本发明实施例相关的部分。所述基于 Android 平台的远程控制装置包括:控制请求接收模块 301、截图模块 302、事件接收模块 303、以及写入模块 304。所述基于 Android 平台的远程控制装置可以是内置于服务端 Android 设备中的软件单元、硬件单元或者是软硬件结合的单元。

[0116] 控制请求接收模块 301,用于接收客户端 Android 设备发起的远程控制服务端 Android 设备的请求消息;

[0117] 截图模块 302,用于服务端 Android 设备接受请求后,将其当前界面不断的截图后发送到客户端 Android 设备;

[0118] 事件接收模块 303,用于接收客户端 Android 设备以约定格式发送的点击操作事件;

[0119] 写入模块 304,用于将所述事件写入对应的设备文件以实现输入事件的模拟。

[0120] 作为本发明一优选实施例,所述基于 Android 平台的远程控制装置还包括:连接请求接收模块、验证模块、以及转换模块。

[0121] 连接请求接收模块,用于接收客户端 Android 设备发送的连接请求消息;

[0122] 验证模块,用于验证所述连接请求消息,若验证成功,则向客户端 Android 设备发送连接请求通过消息;

[0123] 转换模块,用于将图片格式的所述当前界面转换为视频流发送给客户端 Android 设备。

[0124] 作为本发明一优选实施例,所述基于 Android 平台的远程控制装置还包括:获取模块、监视模块、可读判断模块、移除模块、以及跳转模块。

[0125] 获取模块,用于获取监视文件描述符;

[0126] 监视模块,用于监视文件 IN_CLOSE_WRITE 状态;

[0127] 可读判断模块,用于判断 poll 监视文件是否可读;

[0128] 移除模块,用于如果可读,将监视事件从事件队列中移除;

[0129] 跳转模块,用于跳转到所述可读判断模块中执行判断 poll 监视文件是否可读。

[0130] 综上所述,客户端 Android 设备与服务端 Android 设备建立连接,并向所述服务端 Android 设备发起远程控制请求消息;接收服务端 Android 设备发送的当前界面的截图;将所述截图显示在本端的显示屏上;接收用户的点击操作指令;判断所述点击操作对应的坐标是否位于所述截图中;当判断出所述点击操作对应的坐标位于所述截图中,则以约定格式向服务端 Android 设备发送该点击操作事件,以使服务端 Android 设备将所述事件写入对应的设备文件以实现输入事件的模拟;当判断出所述点击操作对应的坐标不在所述截图中,则客户端 Android 设备在本地执行所述点击操作指令对应的操作。本发明可实现任意两种基于 Android 平台的设备可互相对彼此进行远程桌面控制。

[0131] 本领域普通技术人员可以理解实现上述实施例方法中的全部或部分步骤是可以通程序来指令相关的硬件来完成,所述的程序可以存储于一计算机可读取存储介质中,所述的存储介质,如 ROM/RAM、磁盘、光盘等。

[0132] 以上所述仅为本发明的较佳实施例而已,并不用以限制本发明,凡在本发明的精神和原则之内所作的任何修改、等同替换和改进等,均应包含在本发明的保护范围之内。

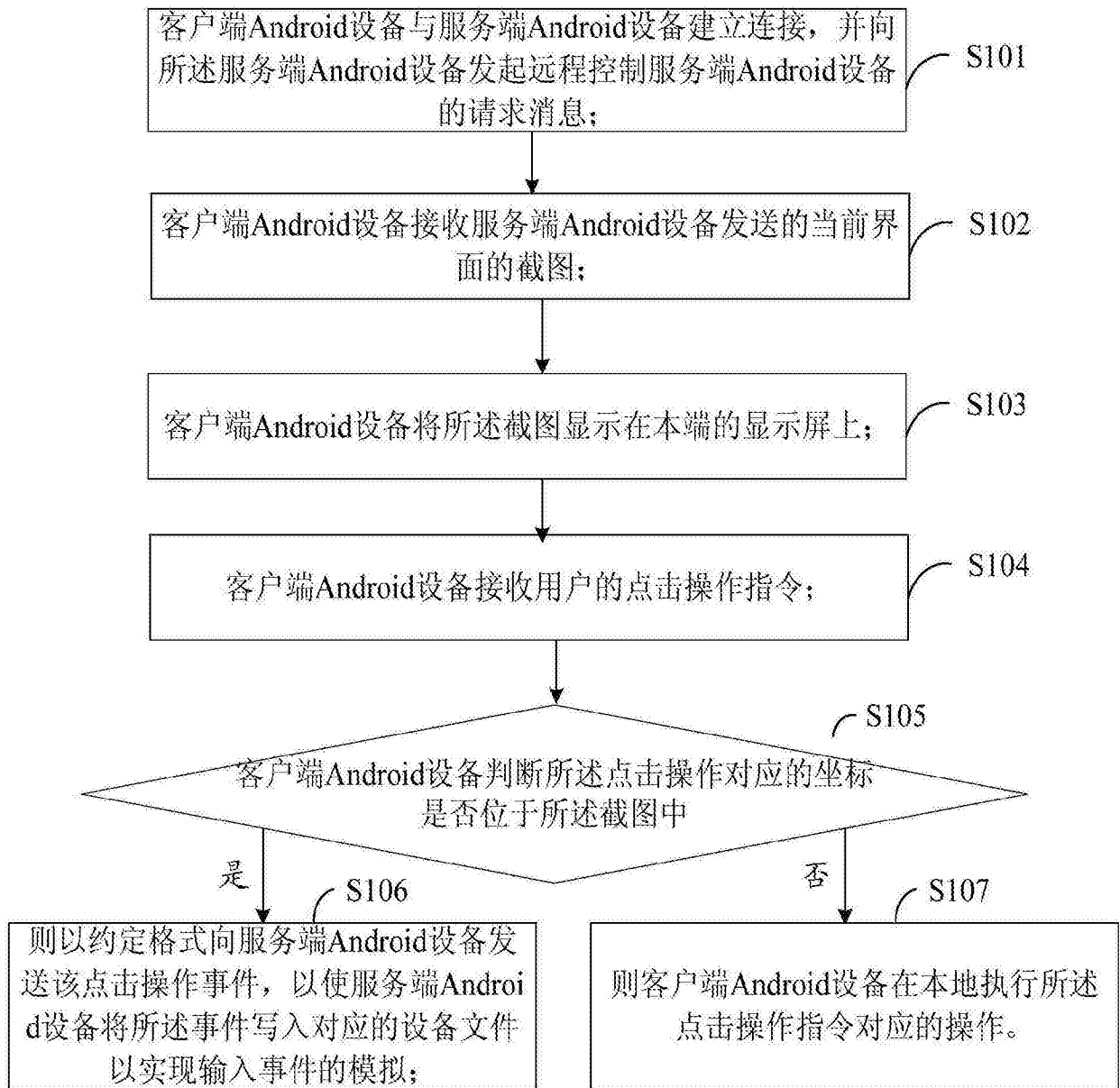


图 1

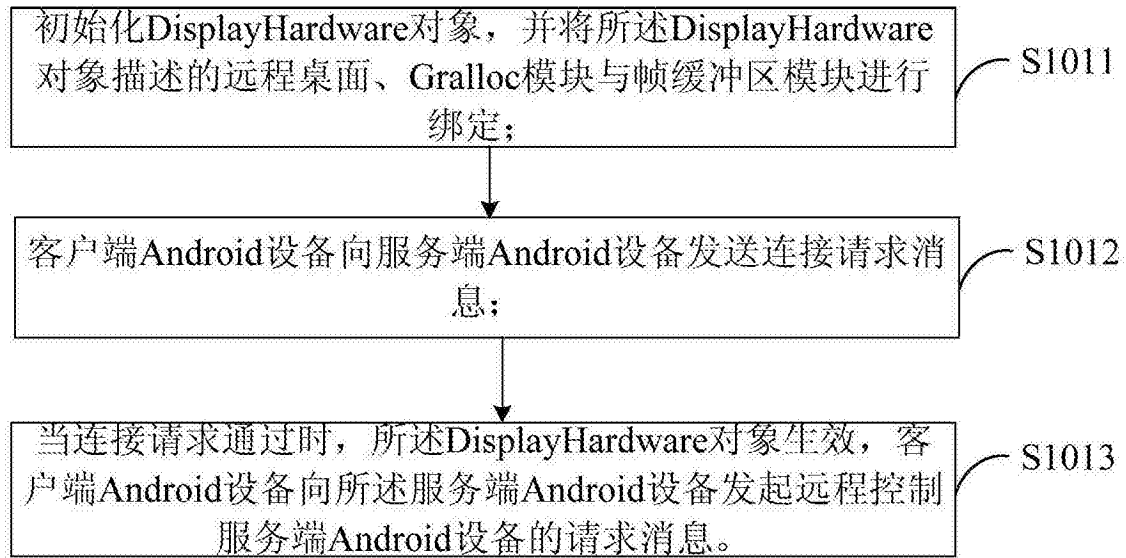


图 2

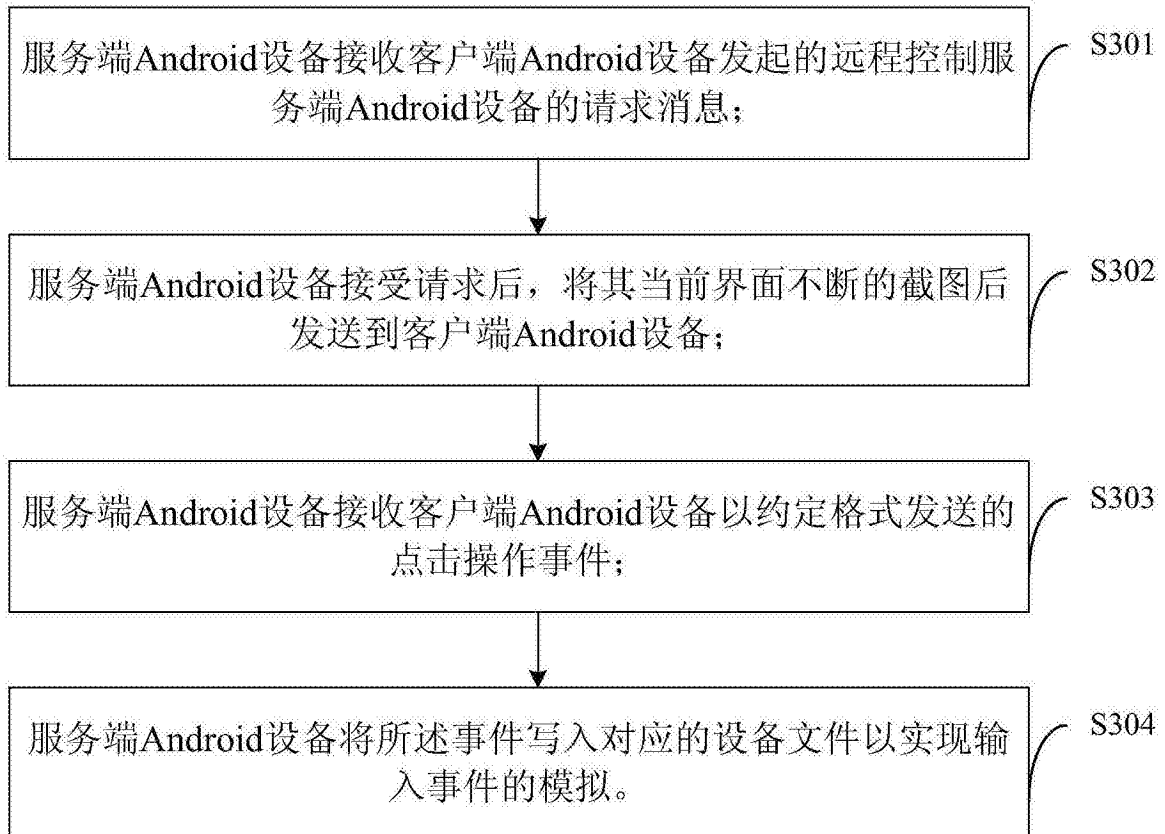


图 3

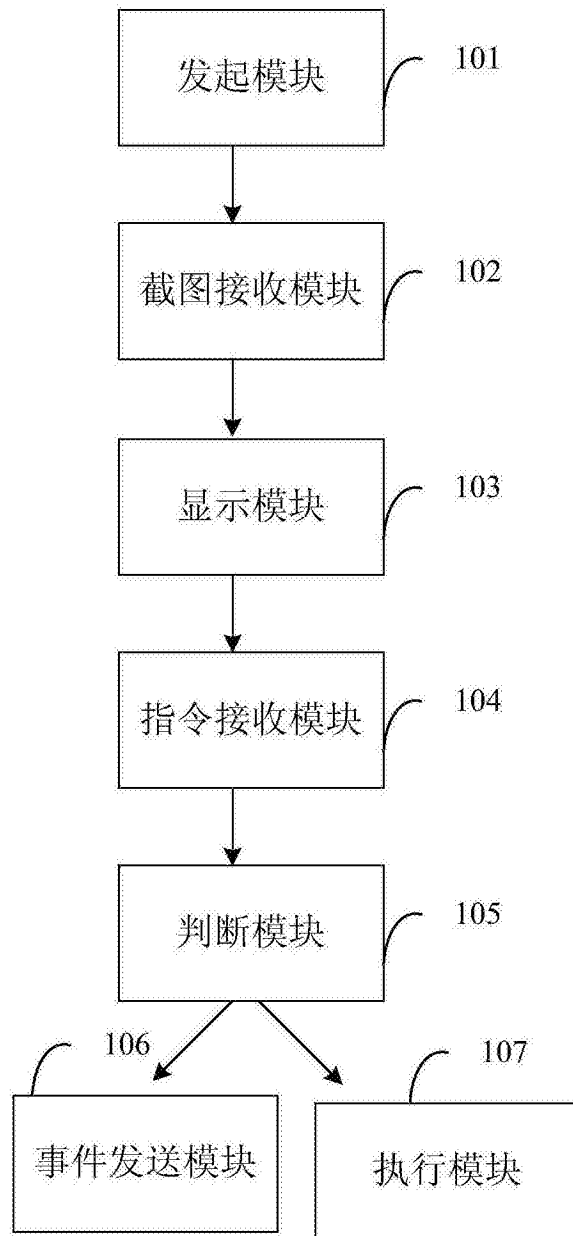


图 4

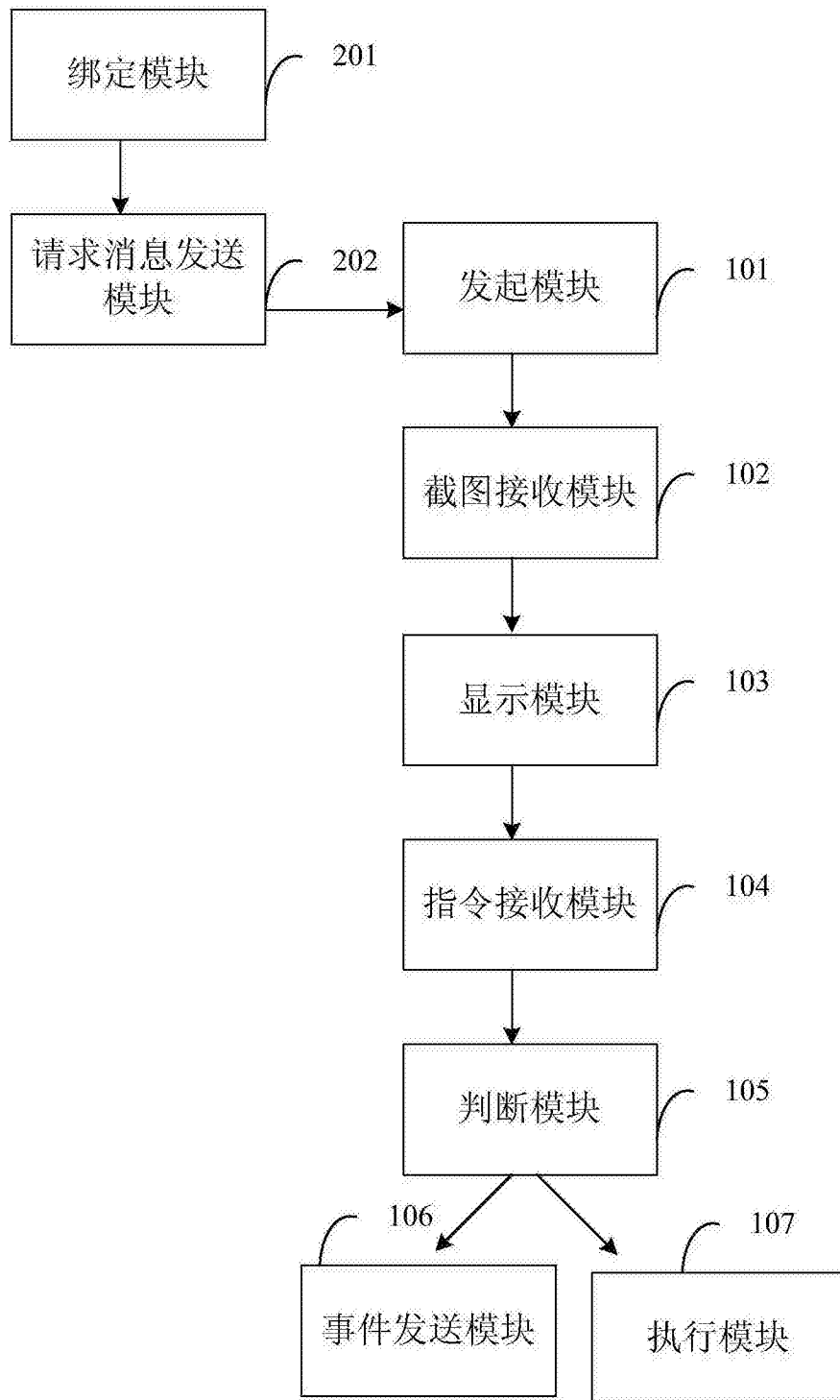


图 5

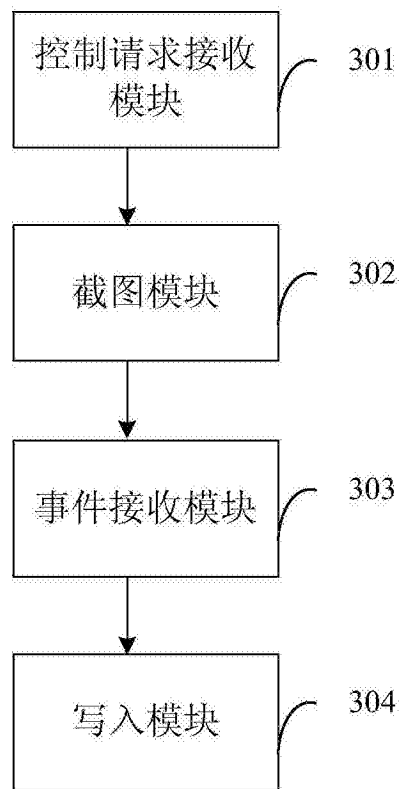


图 6