

(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
2022年3月3日 (03.03.2022)

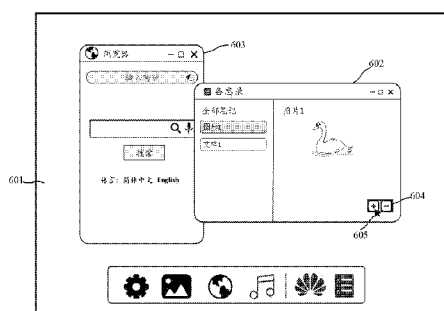


(10) 国际公布号
WO 2022/042285 A1

- (51) 国际专利分类号:
G06F 9/44 (2018.01)
- (21) 国际申请号: PCT/CN2021/111624
- (22) 国际申请日: 2021年8月9日 (09.08.2021)
- (25) 申请语言: 中文
- (26) 公布语言: 中文
- (30) 优先权:
202010899520.6 2020年8月31日 (31.08.2020) CN
- (71) 申请人: 华为技术有限公司 (HUAWEI TECHNOLOGIES CO., LTD.) [CN/CN]; 中国广东省深圳市龙岗区坂田华为总部办公楼, Guangdong 518129 (CN).
- (72) 发明人: 何书杰 (HE, Shujie); 中国广东省深圳市龙岗区坂田华为总部办公楼, Guangdong 518129 (CN)。 罗朴良 (LUO, Puliang); 中国广东省深圳市龙岗区坂田华为总部办公楼, Guangdong 518129 (CN)。 胡迁乔 (HU, Qianqiao); 中国广东省深圳市龙岗区坂田华为总部办公楼, Guangdong 518129 (CN)。
- (74) 代理人: 广州三环专利商标代理有限公司 (SCIHEAD IP LAW FIRM); 中国广东省广州市越秀区先烈中路80号汇华商贸大厦1508室, Guangdong 510070 (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, CZ, DE, DJ, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB,

(54) Title: METHOD FOR DISPLAYING INTERFACE OF APPLICATION PROGRAM AND ELECTRONIC DEVICE

(54) 发明名称: 一种应用程序界面显示的方法及电子设备



放大显示 AA

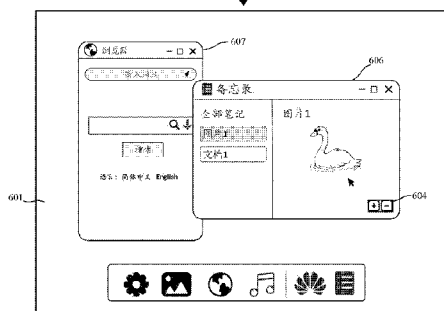


图 6a

AA Zoom in display

(57) Abstract: A method for displaying an interface of an application program and an electronic device. When the electronic device detects a first user operation of scaling an application interface, the first user operation correspondingly generates a system display pixel density, and the system display pixel density is not a system display pixel density actually set of the current system. A change in system configuration triggers an update of the settings of the application program, and after the system display pixel density correspondingly assigned by the first user operation is acquired, a first application program loads a corresponding application resource and displays the zoomed-in or zoomed-out application interface. By implementing said method, a user interface of an application program can be quickly scaled and displayed, satisfying the requirements of a user for zooming in or zooming out the displayed content in the application interface, providing a user with a friendly operating environment, improving the user's visual experience, facilitating developers, and reducing the development resources of the application program.

(57) 摘要: 一种应用程序界面显示的方法及电子设备, 在电子设备检测到缩放应用界面的第一用户操作时, 第一用户操作对应生成一个系统显示像素密度, 该系统显示像素密度并不是当前系统真实设置的系统显示像素密度。系统配置的改变会触发应用程序设置的更新, 在获取到第一用户操作对应派发的系统显示像素密度之后, 第一应用程序会加载对应的应用资源并显示放大或缩小的应用界面。实施本方法, 可以使得应用程序的用户界面实现快捷的缩放显示, 满足用户对于应用界面中的显示内容放大或缩小的需求, 给用户提供了一个友好的操作环境, 提升了用户视觉体验, 并便利了开发人员, 节省了应用程序的开发资源。

WO 2022/042285 A1

GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, IT, JO, JP, KE, KG, KH, KN, KP, KR, KW, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, ME, MG, MK, MN, 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, GH, GM, KE, LR, LS, MW, MZ, NA, RW, 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, 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))。

一种应用程序界面显示的方法及电子设备

本申请要求于2020年8月31日提交中国专利局、申请号为202010899520.6、申请名称为“一种应用程序界面显示的方法及电子设备”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

本申请涉及终端领域，尤其涉及一种应用程序界面显示的方法及电子设备。

背景技术

在搭载安卓(Android®)系统的个人计算机(personal computer, PC)中，应用程序界面会根据系统设置的显示像素密度的大小进行适配显示。通常在系统显示像素密度确定的情况下，应用程序只支持一种适配显示情况，这可能造成应用程序界面中显示内容过大或过小，不符合用户阅读需求的情况。如果强行放大应用程序的显示界面，应用程序中的各个元素可能会进行尺寸的缩放以及位置的调整，同时增加或减少应用程序显示界面内容，还可能造成应用程序显示界面中图片失真，控件变形，字体显示模糊等问题，用户视觉体验不佳。

发明内容

本申请提供一种界面显示的方法、电子设备及计算机存储介质，用来解决应用程序界面中缩放显示的问题。

上述目标和其他目标将通过独立权利要求中的特征来达成。进一步的实现方式在从属权利要求、说明书和附图中体现。

第一方面，本申请实施例提供了一种界面显示的方法，该方法可包括：

电子设备可以显示第一应用程序的第一窗口、第二应用程序的第二窗口，第一窗口和第二窗口中显示的界面元素采用第一像素密度。电子设备检测到第一输入操作，第一输入操作用于确定第二像素密度，第二像素密度不同于所述第一像素密度。电子设备显示第一应用程序的第三窗口和第二应用程序的第二窗口，第三窗口中显示的界面元素采用第二像素密度，第二窗口中显示的界面元素采用第一像素密度。

实施第一方面的方法，可以使得应用程序的用户界面实现快捷的缩放显示，满足用户对于应用界面中的显示内容放大或缩小的需求，给用户提供了一个友好的操作环境，提升了用户视觉体验，并便利了开发人员，节省了应用程序的开发资源。

结合第一方面，在一些实施例中，该方法还包括：在显示第三窗口之前，电子设备向第一应用程序通知第二像素密度。

结合第一方面，在一些实施例中，电子设备可以包括缩放控件，缩放控件可以包括放大控件和缩小控件。

结合第一方面，在一些实施例中，如果第一输入操作为作用于上述放大控件的操作，则第二像素密度小于第一像素密度。

结合第一方面，在一些实施例中，如果第一输入操作为作用于上述放大控件的操作，则第三窗口中显示的界面元素的尺寸大于第一窗口中显示的界面元素的尺寸。

结合第一方面，在一些实施例中，如果第一输入操作为作用于上述缩小控件的操作，则第二像素密度大于第一像素密度。

结合第一方面，在一些实施例中，如果第一输入操作为作用于上述缩小控件的操作，则第三窗口中显示的界面元素的尺寸小于第一窗口中显示的界面元素的尺寸。

结合第一方面，在一些实施例中，该方法还包括：在获取到第二像素密度之后，第一应用程序可以根据第二像素密度刷新应用配置并加载第二应用资源，该第二应用资源为针对第二像素密度而配置的资源文件，第一应用资源为针对第一像素密度而配置的资源文件，第二应用资源不同于第一应用资源。

结合第一方面，在一些实施例中，第一应用程序可以采用的像素密度 dpi 为以下之一：120dpi、160dpi、240dpi、320dpi、480dpi、640dpi。

结合第一方面，在一些实施例中，如果第一应用程序包含多套应用资源，则第一应用程序在第一名单中；或者如果第一应用程序只有单套应用资源，则第一应用程序在第二名单中。

结合第一方面，在一些实施例中，该方法还包括：如果第一应用程序位于第一名单中，或者第一应用程序不在第二名单中，则电子设备响应第一输入操作。

结合第一方面，在一些实施例中，第一输入操作可以为以下任意一项或多项放大操作：同时按住键盘的第一按键和上滑鼠标的滚轮、同时按住键盘的第二按键和第三按键、点击缩放控件的放大控件、双指在触摸屏或触控板上向相反方向张开滑动并达到第一距离、第一手势操作。

结合第一方面，在一些实施例中，前述缩放控件为悬浮控件。

结合第一方面，在一些实施例中，前述缩放控件位于第一窗口内。

结合第一方面，在一些实施例中，第一输入操作可以为以下任意一项或多项缩小操作：同时按住键盘的第一按键和下滑鼠标的滚轮、同时按住键盘的第二按键和第四按键、点击缩放控件的缩小控件、双指在触摸屏或触控板上向相反方向聚拢滑动并达到第二距离、第二手势操作。

第二方面，本申请实施例提供了另外一种界面显示的方法，该方法可包括：

电子设备显示第一用户界面，第一用户界面包括第一区域和第二区域，其中，第一区域中显示的界面元素来自电子设备的操作系统，第二区域中显示的界面元素来自第一应用程序，第一区域、第二区域的显示像素密度为第一系统显示像素密度。电子设备检测到第一输入操作，第一输入操作用于确定第二系统显示像素密度。电子设备显示第二用户界面，第二用户界面包括第一区域和第二区域，其中，第一区域的显示像素密度为第一系统显示像素密度，第二区域中的显示像素密度为第二系统显示像素密度。

实施第二方面的方法，可以使得应用程序的用户界面实现快捷的缩放显示，满足用户对于应用界面中的显示内容放大或缩小的需求，给用户提供了一个友好的操作环境，提升了用户视觉体验，并便利了开发人员，节省了应用程序的开发资源。

结合第二方面，在一些实施例中，该方法还包括：在显示第二用户界面之前，电子设备向第一应用程序通知第二显示像素密度。

结合第二方面，在一些实施例中，电子设备可以包括缩放控件，缩放控件可以包括放大控件和缩小控件。

结合第二方面，在一些实施例中，第一输入操作可以为放大操作或缩小操作。

结合第二方面，在一些实施例中，如果第一输入操作为放大操作，则第二显示像素密度小于第一显示像素密度。

结合第二方面，在一些实施例中，如果第一输入操作为缩小操作，则第二显示像素密度大于第一显示像素密度。

结合第二方面，在一些实施例中，该方法还包括：在获取到第二显示像素密度之后，第一应用程序可以根据第二显示像素密度刷新应用配置并加载第二应用资源，该第二应用资源为针对第二显示像素密度而配置的资源文件，第一应用资源为针对第一显示像素密度而配置的资源文件，第二应用资源不同于第一应用资源。

结合第二方面，在一些实施例中，第一应用程序可以采用的像素密度 dpi 为以下之一：120dpi、160dpi、240dpi、320dpi、480dpi、640dpi。

结合第二方面，在一些实施例中，如果第一应用程序包含多套应用资源，则第一应用程序在第一名单中；或者如果第一应用程序只有单套应用资源，则第一应用程序在第二名单中。

结合第二方面，在一些实施例中，该方法还包括：如果第一应用程序位于第一名单中，或者第一应用程序不在第二名单中，则电子设备响应第一输入操作。

结合第二方面，在一些实施例中，放大操作可以为以下任意一项或多项：同时按住键盘的第一按键和上滑鼠标的滚轮、同时按住键盘的第二按键和第三按键、点击缩放控件的放大控件、双指在触摸屏或触控板上向相反方向张开滑动并达到第一距离、第一手势操作。

结合第二方面，在一些实施例中，缩小操作可以为以下任意一项或多项：同时按住键盘的第一按键和下滑鼠标的滚轮、同时按住键盘的第二按键和第四按键、点击缩放控件的缩小控件、双指在触摸屏或触控板上向相反方向聚拢滑动并达到第二距离、第二手势操作。

第三方面，本发明实施例提供了一种电子设备，该电子设备包括：存储器以及耦合于所述存储器的处理器，存储器中存储有计算机可执行指令，处理器用于调用指令以使得电子设备实现如第一方面、第二方面中电子设备具有的任一功能，这里不再赘述。

第四方面，本发明实施例提供了一种计算机存储介质，该存储介质中存储有计算机程序，该计算机程序包括可执行指令，该可执行指令当被处理器执行时使该处理器执行如第一方面、第二方面所提供的方法对应的操作。

实施本申请的技术方案，可以使得应用程序的用户界面实现快捷的缩放显示，满足用户对于应用界面中的显示内容放大或缩小的需求，给用户提供了一个友好的操作环境，提升了用户视觉体验，并便利了开发人员，节省了应用程序的开发资源。

附图说明

为了更清楚地说明本申请实施例或背景技术中的技术方案，下面将对本申请实施例或背景技术中所需要使用的附图进行说明。

图 1 是本申请实施例提供的一种电子设备的硬件结构示意图；

图 2 是本申请实施例提供的一种电子设备的软件结构框图；

图 3a 是本申请实施例提供的一种用户界面示意图；

图 3b 是本申请实施例提供的一种用户界面示意图；

图 3c 是本申请实施例提供的一种用户界面示意图；

图 4a 是本申请实施例提供的一种用户界面示意图；

图 4b 是本申请实施例提供的一种用户界面示意图；

图 4c 是本申请实施例提供的一种用户界面示意图；

图 5a 是本申请实施例提供的一种用户界面示意图；

图 5b 是本申请实施例提供的一种用户界面示意图；
图 5c 是本申请实施例提供的一种用户界面示意图；
图 6a 是本申请实施例提供的一种用户界面示意图；
图 6b 是本申请实施例提供的一种用户界面示意图；
图 7 是本申请一个实施例提供的一种界面显示的方法流程图；
图 8 是本申请一个实施例提供的一种界面显示的方法流程图。

具体实施方式

下面结合附图对本申请实施例进行具体说明。本申请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“一个”、“一种”、“所述”、“上述”、“该”和“这一”旨在也包括复数表达形式，除非其上下文中明确地有相反指示。还应当理解，本申请中使用的术语“和/或”是指并包含一个或多个所列出项目的任何或所有可能组合。

本申请提供了一种电子设备，该电子设备可以是笔记本电脑、台式电脑、平板电脑、智能手机以及其他类型的电子设备。本申请还提供了一种应用程序界面显示的方法，该方法可应用于本申请提供的电子设备，在电子设备检测到缩放应用界面的第一用户操作时，第一用户操作对应生成一个系统显示像素密度，该系统显示像素密度并不是当前系统真实设置的系统显示像素密度。系统配置的改变会触发应用程序设置的更新，在获取到第一用户操作对应派发的虚拟的系统显示像素密度之后，第一应用程序会加载对应的应用资源并显示放大或缩小的应用界面。其中，第一用户操作可以是在第一应用程序的用户界面内点击缩放控件，比如放大（+）、缩小（-）的控件；也可以是在鼠标指针位于第一应用程序的用户界面中时，同时按住“ctrl”键和上滑/下滑鼠标滚轮，或者同时按住“ctrl”键和“▲”/“▼”键等等；还可以是触摸事件，例如在触摸屏的第一应用程序的用户界面中，用户两只手指向相反方向聚拢/张开触摸屏；还可以是手势操作事件，例如隔空识别特殊手势等等。实施本申请的技术方案，可以使得应用程序的用户界面实现快捷的缩放显示，满足用户对于应用界面中的显示内容放大或缩小的需求，给用户提供了一个友好的操作环境，提升了用户视觉体验，并便利了开发人员，节省了应用程序的开发资源。

首先介绍本申请实施例中提供的示例性电子设备 100。应该理解的是，电子设备 100 可以具有比图中所示的更多的或者更少的部件，可以组合两个或多个的部件，或者可以具有不同的部件配置。图中所示出的各种部件可以在包括一个或多个信号处理和/或专用集成电路在内的硬件、软件、或硬件和软件的组合中实现。

图 1 为电子设备 100 的结构示意图。

电子设备 100 可以包括：处理器 110，外部存储器接口 120，内部存储器 121，通用串行总线(universal serial bus, USB)接口 130，充电管理模块 140，电源管理模块 141，电池 142，天线 1，天线 2，移动通信模块 150，无线通信模块 160，音频模块 170，扬声器 170A，受话器 170B，麦克风 170C，耳机接口 170D，传感器模块 180，按键 190，马达 191，指示器 192，摄像头 193，显示屏 194 以及用户标识模块(subscriber identification module, SIM)卡接口 195 等。其中传感器模块 180 可以包括压力传感器 180A，陀螺仪传感器 180B，气压传感器 180C，磁传感器 180D，加速度传感器 180E，距离传感器 180F，接近光传感器 180G，指纹传感器

180H, 温度传感器 180J, 触摸传感器 180K, 环境光传感器 180L, 骨传导传感器 180M 等。

可以理解的是, 本申请实施例示意的结构并不构成对电子设备 100 的具体限定。在本申请另一些实施例中, 电子设备 100 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 110 可以包括一个或多个处理单元, 例如: 处理器 110 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。

其中, 控制器可以是电子设备 100 的神经中枢和指挥中心。控制器可以根据指令操作码和时序信号, 产生操作控制信号, 完成取指令和执行指令的控制。

处理器 110 中还可以设置存储器, 用于存储指令和数据。在一些实施例中, 处理器 110 中的存储器为高速缓冲存储器。该存储器可以保存处理器 110 刚用过或循环使用的指令或数据。如果处理器 110 需要再次使用该指令或数据, 可从所述存储器中直接调用。避免了重复存取, 减少了处理器 110 的等待时间, 因而提高了系统的效率。

在一些实施例中, 处理器 110 可以包括一个或多个接口。接口可以包括集成电路(inter-integrated circuit, I2C)接口, 集成电路内置音频(inter-integrated circuit sound, I2S)接口, 脉冲编码调制(pulse code modulation, PCM)接口, 通用异步收发传输器(universal asynchronous receiver/transmitter, UART)接口, 移动产业处理器接口(mobile industry processor interface, MIPI), 通用输入输出(general-purpose input/output, GPIO)接口, 用户标识模块(subscriber identity module, SIM)接口, 和/或通用串行总线(universal serial bus, USB)接口等。

I2C 接口是一种双向同步串行总线, 包括一根串行数据线(serial data line, SDA)和一根串行时钟线(derail clock line, SCL)。在一些实施例中, 处理器 110 可以包含多组 I2C 总线。处理器 110 可以通过不同的 I2C 总线接口分别耦合触摸传感器 180K, 充电器, 闪光灯, 摄像头 193 等。例如: 处理器 110 可以通过 I2C 接口耦合触摸传感器 180K, 使处理器 110 与触摸传感器 180K 通过 I2C 总线接口通信, 实现电子设备 100 的触摸功能。

I2S 接口可以用于音频通信。在一些实施例中, 处理器 110 可以包含多组 I2S 总线。处理器 110 可以通过 I2S 总线与音频模块 170 耦合, 实现处理器 110 与音频模块 170 之间的通信。在一些实施例中, 音频模块 170 可以通过 I2S 接口向无线通信模块 160 传递音频信号, 实现通过蓝牙耳机接听电话的功能。

PCM 接口也可以用于音频通信, 将模拟信号抽样, 量化和编码。在一些实施例中, 音频模块 170 与无线通信模块 160 可以通过 PCM 总线接口耦合。在一些实施例中, 音频模块 170 也可以通过 PCM 接口向无线通信模块 160 传递音频信号, 实现通过蓝牙耳机接听电话的功能。所述 I2S 接口和所述 PCM 接口都可以用于音频通信。

UART 接口是一种通用串行数据总线, 用于异步通信。该总线可以为双向通信总线。它将要传输的数据在串行通信与并行通信之间转换。在一些实施例中, UART 接口通常被用于连接处理器 110 与无线通信模块 160。例如: 处理器 110 通过 UART 接口与无线通信模块 160 中的蓝牙模块通信, 实现蓝牙功能。在一些实施例中, 音频模块 170 可以通过 UART 接口向无线通信模块 160 传递音频信号, 实现通过蓝牙耳机播放音乐的功能。

MIPI 接口可以被用于连接处理器 110 与显示屏 194, 摄像头 193 等外围器件。MIPI 接口包括摄像头串行接口 (camera serial interface, CSI), 显示屏串行接口 (display serial interface, DSI) 等。在一些实施例中, 处理器 110 和摄像头 193 通过 CSI 接口通信, 实现电子设备 100 的拍摄功能。处理器 110 和显示屏 194 通过 DSI 接口通信, 实现电子设备 100 的显示功能。

GPIO 接口可以通过软件配置。GPIO 接口可以被配置为控制信号, 也可被配置为数据信号。在一些实施例中, GPIO 接口可以用于连接处理器 110 与摄像头 193, 显示屏 194, 无线通信模块 160, 音频模块 170, 传感器模块 180 等。GPIO 接口还可以被配置为 I2C 接口, I2S 接口, UART 接口, MIPI 接口等。

SIM 接口可以被用于与 SIM 卡接口 195 通信, 实现传送数据到 SIM 卡或读取 SIM 卡中数据的功能。

USB 接口 130 是符合 USB 标准规范的接口, 具体可以是 Mini USB 接口, Micro USB 接口, USB Type C 接口等。USB 接口 130 可以用于连接充电器为电子设备 100 充电, 也可以用于电子设备 100 与外围设备之间传输数据。也可以用于连接耳机, 通过耳机播放音频。该接口还可以用于连接其他电子设备, 例如 AR 设备等。

可以理解的是, 本申请实施例示意的各模块间的接口连接关系, 只是示意性说明, 并不构成对电子设备 100 的结构限定。在本申请另一些实施例中, 电子设备 100 也可以采用上述实施例中不同的接口连接方式, 或多种接口连接方式的组合。

充电管理模块 140 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。

电源管理模块 141 用于连接电池 142, 充电管理模块 140 与处理器 110。电源管理模块 141 接收电池 142 和/或充电管理模块 140 的输入, 为处理器 110, 内部存储器 121, 外部存储器, 显示屏 194, 摄像头 193, 和无线通信模块 160 等供电。

电子设备 100 的无线通信功能可以通过天线 1, 天线 2, 移动通信模块 150, 无线通信模块 160, 调制解调处理器以及基带处理器等实现。

天线 1 和天线 2 用于发射和接收电磁波信号。电子设备 100 中的每个天线可用于覆盖单个或多个通信频带。不同的天线还可以复用, 以提高天线的利用率。例如: 可以将天线 1 复用为无线局域网的分集天线。在另外一些实施例中, 天线可以和调谐开关结合使用。

移动通信模块 150 可以提供应用在电子设备 100 上的包括 2G/3G/4G/5G 等无线通信的解决方案。移动通信模块 150 可以包括至少一个滤波器, 开关, 功率放大器, 低噪声放大器 (low noise amplifier, LNA) 等。移动通信模块 150 可以由天线 1 接收电磁波, 并对接收的电磁波进行滤波, 放大等处理, 传送至调制解调处理器进行解调。移动通信模块 150 还可以对经调制解调处理器调制后的信号放大, 经天线 1 转为电磁波辐射出去。在一些实施例中, 移动通信模块 150 的至少部分功能模块可以被设置于处理器 110 中。在一些实施例中, 移动通信模块 150 的至少部分功能模块可以与处理器 110 的至少部分模块被设置在同一个器件中。

调制解调处理器可以包括调制器和解调器。其中, 调制器用于将待发送的低频基带信号调制为中高频信号。解调器用于将接收的电磁波信号解调为低频基带信号。随后解调器将解调得到的低频基带信号传送至基带处理器处理。低频基带信号经基带处理器处理后, 被传递给应用处理器。应用处理器通过音频设备 (不限于扬声器 170A, 受话器 170B 等) 输出声音信号, 或通过显示屏 194 显示图像或视频。在一些实施例中, 调制解调处理器可以是独立的器件。在另一些实施例中, 调制解调处理器可以独立于处理器 110, 与移动通信模块 150 或其他功能模块设置在同一个器件中。

无线通信模块 160 可以提供应用在电子设备 100 上的包括无线局域网 (wireless local area networks, WLAN) (如无线保真 (wireless fidelity, Wi-Fi) 网络), 蓝牙 (bluetooth, BT), 全球导航卫星系统 (global navigation satellite system, GNSS), 调频 (frequency modulation, FM), 近距离无线通信技术 (near field communication, NFC), 红外技术 (infrared, IR) 等无线通信的解决方案。无线通信模块 160 可以是集成至少一个通信处理模块的一个或多个器件。无线通信模块 160 经由天线 2 接收电磁波, 将电磁波信号调频以及滤波处理, 将处理后的信号发送到处理器 110。无线通信模块 160 还可以从处理器 110 接收待发送的信号, 对其进行调频, 放大, 经天线 2 转为电磁波辐射出去。

在一些实施例中, 电子设备 100 的天线 1 和移动通信模块 150 耦合, 天线 2 和无线通信模块 160 耦合, 使得电子设备 100 可以通过无线通信技术与网络以及其他设备通信。所述无线通信技术可以包括全球移动通讯系统 (global system for mobile communications, GSM), 通用分组无线服务 (general packet radio service, GPRS), 码分多址接入 (code division multiple access, CDMA), 宽带码分多址 (wideband code division multiple access, WCDMA), 时分码分多址 (time-division code division multiple access, TD-SCDMA), 长期演进 (long term evolution, LTE), BT, GNSS, WLAN, NFC, FM, 和/或 IR 技术等。所述 GNSS 可以包括全球卫星定位系统 (global positioning system, GPS), 全球导航卫星系统 (global navigation satellite system, GLONASS), 北斗卫星导航系统 (beidou navigation satellite system, BDS), 准天顶卫星系统 (quasi-zenith satellite system, QZSS) 和/或星基增强系统 (satellite based augmentation systems, SBAS)。

电子设备 100 通过 GPU, 显示屏 194, 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 194 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 110 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。处理器 110 可以在应用程序进行界面刷新时, 根据预设的布局规则和显示屏 194 当前的显示状态, 对应用程序界面上的元素进行动态布局。

显示屏 194 用于显示图像, 视频等。显示屏 194 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。在一些实施例中, 电子设备 100 可以包括 1 个或 N 个显示屏 194, N 为大于 1 的正整数。显示屏 194 的分辨率可以表示为显示屏所能显示的物理像素点的总数, 例如 1920×1080。通常用像素密度 (pixelsperinch, PPI) 表示显示屏显示的细腻程度, PPI 越高, 显示屏显示的界面就会越细腻, 显示细节就越丰富。

电子设备 100 可以通过 ISP, 摄像头 193, 视频编解码器, GPU, 显示屏 194 以及应用处理器等实现拍摄功能。

ISP 用于处理摄像头 193 反馈的数据。例如, 拍照时, 打开快门, 光线通过镜头被传递到摄像头感光元件上, 光信号转换为电信号, 摄像头感光元件将所述电信号传递给 ISP 处理, 转化为肉眼可见的图像。ISP 还可以对图像的噪点, 亮度, 肤色进行算法优化。ISP 还可以对拍摄场景的曝光, 色温等参数优化。在一些实施例中, ISP 可以设置在摄像头 193 中。

摄像头 193 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。感光元件可以是电荷耦合器件 (charge coupled device, CCD) 或互补金属氧化物半导体

(complementary metal-oxide-semiconductor, CMOS)光电晶体管。感光元件把光信号转换成电信号,之后将电信号传递给ISP转换成数字图像信号。ISP将数字图像信号输出到DSP加工处理。DSP将数字图像信号转换成标准的RGB, YUV等格式的图像信号。在一些实施例中,电子设备100可以包括1个或N个摄像头193, N为大于1的正整数。

数字信号处理器用于处理数字信号,除了可以处理数字图像信号,还可以处理其他数字信号。例如,当电子设备100在频点选择时,数字信号处理器用于对频点能量进行傅里叶变换等。

视频编解码器用于对数字视频压缩或解压缩。电子设备100可以支持一种或多种视频编解码器。这样,电子设备100可以播放或录制多种编码格式的视频,例如:动态图像专家组(moving picture experts group, MPEG)1, MPEG2, MPEG3, MPEG4等。

NPU为神经网络(neural-network, NN)计算处理器,通过借鉴生物神经网络结构,例如借鉴人脑神经元之间传递模式,对输入信息快速处理,还可以不断的自学习。通过NPU可以实现电子设备100的智能认知等应用,例如:图像识别,人脸识别,语音识别,文本理解等。

外部存储器接口120可以用于连接外部存储卡,例如Micro SD卡,实现扩展电子设备100的存储能力。外部存储卡通过外部存储器接口120与处理器110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121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110通过运行存储在内部存储器121的指令,从而执行电子设备100的各种功能应用以及数据处理。内部存储器121可以包括存储程序区和存储数据区。其中,存储程序区可存储操作系统,至少一个功能所需的应用(比如人脸识别功能,指纹识别功能、移动支付功能等)等。存储数据区可存储电子设备100使用过程中所创建的数据(比如人脸信息模板数据,指纹信息模板等)等。此外,内部存储器121可以包括高速随机存取存储器,还可以包括非易失性存储器,例如至少一个磁盘存储器件,闪存器件,通用闪存存储器(universal flash storage, UFS)等。

电子设备100可以通过音频模块170,扬声器170A,受话器170B,麦克风170C,耳机接口170D,以及应用处理器等实现音频功能。例如音乐播放,录音等。

音频模块170用于将数字音频信息转换成模拟音频信号输出,也用于将模拟音频输入转换为数字音频信号。音频模块170还可以用于对音频信号编码和解码。在一些实施例中,音频模块170可以设置于处理器110中,或将音频模块170的部分功能模块设置于处理器110中。

扬声器170A,也称“喇叭”,用于将音频电信号转换为声音信号。电子设备100可以通过扬声器170A收听音乐,或收听免提通话。

受话器170B,也称“听筒”,用于将音频电信号转换成声音信号。当电子设备100接听电话或语音信息时,可以通过将受话器170B靠近人耳接听语音。

麦克风170C,也称“话筒”,“传声器”,用于将声音信号转换为电信号。当拨打电话或发送语音信息时,用户可以通过人嘴靠近麦克风170C发声,将声音信号输入到麦克风170C。电子设备100可以设置至少一个麦克风170C。在另一些实施例中,电子设备100可以设置两个麦克风170C,除了采集声音信号,还可以实现降噪功能。在另一些实施例中,电子设备100还可以设置三个,四个或更多麦克风170C,实现采集声音信号,降噪,还可以识别声音来源,实现定向录音功能等。

耳机接口170D用于连接有线耳机。耳机接口170D可以是USB接口130,也可以是3.5mm

的开放移动电子设备平台(open mobile terminal platform, OMTp)标准接口, 美国蜂窝电信工业协会(cellular telecommunications industry association of the USA, CTIA)标准接口。

压力传感器 180A 用于感受压力信号, 可以将压力信号转换成电信号。在一些实施例中, 压力传感器 180A 可以设置于显示屏 194。压力传感器 180A 的种类很多, 如电阻式压力传感器, 电感式压力传感器, 电容式压力传感器等。电容式压力传感器可以是包括至少两个具有导电材料的平行板。当有力作用于压力传感器 180A, 电极之间的电容改变。电子设备 100 根据电容的变化确定压力的强度。当有触摸操作作用于显示屏 194, 电子设备 100 根据压力传感器 180A 检测所述触摸操作强度。电子设备 100 也可以根据压力传感器 180A 的检测信号计算触摸的位置。在一些实施例中, 作用于相同触摸位置, 但不同触摸操作强度的触摸操作, 可以对应不同的操作指令。例如: 当有触摸操作强度小于第一压力阈值的触摸操作作用于短消息应用图标时, 执行查看短消息的指令。当有触摸操作强度大于或等于第一压力阈值的触摸操作作用于短消息应用图标时, 执行新建短消息的指令。

陀螺仪传感器 180B 可以用于确定电子设备 100 的运动姿态。在一些实施例中, 可以通过陀螺仪传感器 180B 确定电子设备 100 围绕三个轴(即, x, y 和 z 轴)的角速度。陀螺仪传感器 180B 可以用于拍摄防抖。示例性的, 当按下快门, 陀螺仪传感器 180B 检测电子设备 100 抖动的角度, 根据角度计算出镜头模组需要补偿的距离, 让镜头通过反向运动抵消电子设备 100 的抖动, 实现防抖。陀螺仪传感器 180B 还可以用于导航, 体感游戏场景。

气压传感器 180C 用于测量气压。在一些实施例中, 电子设备 100 通过气压传感器 180C 测得的气压值计算海拔高度, 辅助定位和导航。

磁传感器 180D 包括霍尔传感器。电子设备 100 可以利用磁传感器 180D 检测翻盖皮套的开合。在一些实施例中, 当电子设备 100 是翻盖机时, 电子设备 100 可以根据磁传感器 180D 检测翻盖的开合。进而根据检测到的皮套的开合状态或翻盖的开合状态, 设置翻盖自动解锁等特性。

加速度传感器 180E 可检测电子设备 100 在各个方向上(一般为三轴)加速度的大小。当电子设备 100 静止时可检测出重力的大小及方向。还可以用于识别电子设备姿态, 应用于横竖屏切换, 计步器等应用。

距离传感器 180F, 用于测量距离。电子设备 100 可以通过红外或激光测量距离。在一些实施例中, 拍摄场景, 电子设备 100 可以利用距离传感器 180F 测距以实现快速对焦。

接近光传感器 180G 可以包括例如发光二极管(LED)和光检测器, 例如光电二极管。发光二极管可以是红外发光二极管。电子设备 100 通过发光二极管向外发射红外光。电子设备 100 使用光电二极管检测来自附近物体的红外反射光。当检测到充分的反射光时, 可以确定电子设备 100 附近有物体。当检测到不充分的反射光时, 电子设备 100 可以确定电子设备 100 附近没有物体。电子设备 100 可以利用接近光传感器 180G 检测用户手持电子设备 100 贴近耳朵通话, 以便自动熄灭屏幕达到省电的目的。接近光传感器 180G 也可用于皮套模式, 口袋模式自动解锁与锁屏。

环境光传感器 180L 用于感知环境光亮度。电子设备 100 可以根据感知的环境光亮度自适应调节显示屏 194 亮度。环境光传感器 180L 也可用于拍照时自动调节白平衡。环境光传感器 180L 还可以与接近光传感器 180G 配合, 检测电子设备 100 是否在口袋里, 以防误触。

指纹传感器 180H 用于采集指纹。电子设备 100 可以利用采集的指纹特性实现指纹解锁, 访问应用锁, 指纹拍照, 指纹接听来电等。

温度传感器 180J 用于检测温度。在一些实施例中, 电子设备 100 利用温度传感器 180J

检测的温度，执行温度处理策略。例如，当温度传感器 180J 上报的温度超过阈值，电子设备 100 执行降低位于温度传感器 180J 附近的处理器的性能，以便降低功耗实施热保护。在另一些实施例，当温度低于另一阈值时，电子设备 100 对电池 142 加热，以避免低温导致电子设备 100 异常关机。在其他一些实施例，当温度低于又一阈值时，电子设备 100 对电池 142 的输出电压执行升压，以避免低温导致的异常关机。

触摸传感器 180K，也称“触控面板”。触摸传感器 180K 可以设置于显示屏 194，由触摸传感器 180K 与显示屏 194 组成触摸屏，也称“触控屏”。触摸传感器 180K 用于检测作用于其上或附近的触摸操作。触摸传感器可以将检测到的触摸操作传递给应用处理器，以确定触摸事件类型。可以通过显示屏 194 提供与触摸操作相关的视觉输出。在另一些实施例，触摸传感器 180K 也可以设置于电子设备 100 的表面，与显示屏 194 所处的位置不同。

键盘 190 是电子设备 100 的输入设备，用于向电子设备 100 输入指令和数据，通过键盘 190 可以将英文字母、标点符号、数字等输入到电子设备 100 中，键盘 190 包括若干按键，电子设备 100 可以接收按键输入，产生与电子设备 100 的用户设置以及功能控制有关的键信号输入。

鼠标 191 是电子设备 100 的输入设备，用于向电子设备 100 输入指令，也是电子设备 100 显示系统纵横坐标定位的指示器。鼠标 191 可以对当前显示屏上的游标进行定位，并通过鼠标按键或滚轮装置对游标所在的位置的显示元素进行操作。

指示器 192 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 195 用于连接 SIM 卡。SIM 卡可以通过插入 SIM 卡接口 195，或从 SIM 卡接口 195 拔出，实现和电子设备 100 的接触和分离。电子设备 100 可以支持 1 个或 N 个 SIM 卡接口，N 为大于 1 的正整数。SIM 卡接口 195 可以支持 Nano SIM 卡，Micro SIM 卡，SIM 卡等。同一个 SIM 卡接口 195 可以同时插入多张卡。所述多张卡的类型可以相同，也可以不同。SIM 卡接口 195 也可以兼容不同类型的 SIM 卡。SIM 卡接口 195 也可以兼容外部存储卡。电子设备 100 通过 SIM 卡和网络交互，实现通话以及数据通信等功能。

图 2 是本申请实施例的电子设备 100 的软件结构框图。

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。如图 2 所示，在一些实施例中，可以将 Android[®] 系统分为五层，从上至下分别为应用程序层 (applications)，应用程序框架层 (applications framework)，系统库 (nativelibraries) 和安卓运行时 (Android[®] runtime)，硬件抽象层 (hardwareabstractlayer, HAL)，以及内核层 (kernel)。

应用程序层 (applications) 可以包括一系列应用程序包。

如图 2 所示，应用程序包可以包括相机，地图，备忘录，桌面 (launcher) 等应用程序 (也可以称为应用)。在本申请的一个实施例中，可以在应用程序层增加一个多像素密度入口 (multi-DPI entry) 控件，比如该多像素密度入口控件可以由放大 (+) 控件和缩小 (-) 控件组成，鼠标点击放大 (+) 控件或缩小 (-) 控件，应用程序的就会对应缩放用户界面，具体可以参考后文实施例，这里暂不赘述。

应用程序框架层 (applications framework) 为应用程序层的应用程序提供应用编程接口 (application programming interface, API) 和编程框架。应用程序框架层包括一些预先定义的函数。

如图 2 所示, 应用程序框架层可以包括窗口管理器 (windowmanager), 活动管理器 (activitymanager), 显示管理器 (displaymanage), 资源管理器 (resourcemanager), 输入管理器 (inputmanager), 通知管理器 (notificationmanager), 视图系统 (views) 等。

窗口管理器 (windowmanager) 用于管理窗口程序。窗口管理器可以用于绘制窗口的大小、位置区域等, 控制显示或隐藏窗口, 管理多个窗口的显示顺序, 还可以获取显示屏大小, 判断是否有状态栏, 锁定屏幕, 截取屏幕等。

活动管理器 (activitymanager) 用于管理应用程序的活动的生命周期, 比如对创建、后台、销毁等活动进程进行管理。

显示管理器 (displaymanage) 用于管理应用程序的显示的生命周期, 它可以决定如何根据当前连接的物理显示设备控制其逻辑显示, 并且在状态更改时, 向系统和应用程序发送通知等等。

输入管理器 (inputmanager) 用于监听并统一管理输入事件, 例如当用户使用键盘或鼠标进行输入时, 输入管理器可以监听到系统的调用, 并将监听到的键盘或鼠标事件进一步转发或者处理。

资源管理器 (resourcemanager) 为应用程序提供各种非代码资源的访问, 比如本地化字符串, 图标, 图片, 布局文件, 视频文件等等。资源管理接口类 ResourceImpl 可以作为对外的资源管理接口, 通过该接口类, 可以实现对应用资源的更新。

通知管理器 (notificationmanager) 使应用程序可以在状态栏中显示通知信息, 可以用于传达告知类型的消息, 可以短暂停留后自动消失, 无需用户交互。比如通知管理器被用于告知下载完成, 消息提醒等。通知管理器还可以是以图表或者滚动条文本形式出现在系统顶部状态栏的通知, 例如后台运行的应用程序的通知, 还可以是以对话框形式出现在屏幕上的通知。例如在状态栏提示文本信息, 发出提示音, 电子设备振动, 指示灯闪烁等。

视图系统 (views) 包括可视控件, 例如显示文字的控件, 显示图片的控件等。视图系统可用于构建应用程序。显示界面可以由一个或多个视图组成的。例如, 包括短信通知图标的显示界面, 可以包括显示文字的视图以及显示图片的视图。

安卓运行时 (Android[®] Runtime) 包括核心库和虚拟机。Android[®] Runtime 负责安卓系统的调度和管理。

核心库包含两部分: 一部分是 java 语言需要调用的功能函数, 另一部分是安卓的核心库。

应用程序层和应用程序框架层运行在虚拟机中。虚拟机将应用程序层和应用程序框架层的 java 文件执行为二进制文件。虚拟机用于执行对象生命周期的管理, 堆栈管理, 线程管理, 安全和异常的管理, 以及垃圾回收等功能。

系统库 (nativelibraries) 可以包括多个功能模块。例如: 界面管理器 (surface manager), 资源加载管理器 (assetmanager), 媒体库 (media libraries), 三维图形处理库 (例如: OpenGL ES), 二维图形引擎 (例如: SGL) 等。

界面管理器 (surface manager) 用于对显示子系统进行管理, 并且为多个应用程序提供了二维 (2-Dimensional, 2D) 和三维 (3-Dimensional, 3D) 图层的融合。界面管理器可以获取 SurfaceFlinger 服务, SurfaceFlinger 服务是图形用户界面 (graphicaluserinterface, GUI) 的核心, 负责将所有应用的图形数据按照顺序混合并输出到缓冲流。

资源加载管理器 (assetmanager) 用于对资源类 Resources 进行加载。

媒体库支持多种常用的音频, 视频格式回放和录制, 以及静态图像文件等。媒体库可以支持多种音视频编码格式, 例如: MPEG4, H.264, MP3, AAC, AMR, JPG, PNG 等。

三维图形处理库用于实现 3D 图形绘图，图像渲染，合成和图层处理等。

2D 图形引擎是 2D 绘图的绘图引擎。

硬件抽象层 (hardwareabstractlayer, HAL) 是位于操作系统内核与硬件电路之间的接口层，其目的在于将硬件抽象化，它可以对 Linux 内核驱动程序进行封装，向上提供标准的接口，隐藏较低级别的驱动程序的实现细节。HAL 可以包含硬件合成器 (hardwarecomposer, HWC)，帧缓冲区 (framebuffer)，鼠标接口，键盘接口，蓝牙接口，GPS 接口，WiFi 接口，音视频接口，通话接口等等。

硬件合成器 (hardwarecomposer, HWC) 用于对 SurfaceFlinger 合成的图形数据缓冲流，进行最终的合成显示。

帧缓冲区 (framebuffer) 是 SurfaceFlinger 服务合成的图形缓冲流，SurfaceFlinger 服务通过向帧缓冲区写入内容来绘制应用程序的用户界面。

鼠标接口、键盘接口可以用于从鼠标驱动、键盘驱动中获取相应的数据。

内核层 (kernel) 是硬件和软件之间的层，用于提供核心系统服务，比如安全、内存管理、进程管理、网络协议栈、驱动模型等。内核层中可以包含显示控制驱动 (displaycontrollerdrive)，传感器驱动 (sensordriver)，中央处理器驱动 (CPU driver)，图形处理器驱动 (GPU driver)，鼠标驱动 (mousedriver)，键盘驱动 (keyboarddriver)，数字版权管理 (digitalrightsmanagement, DRM) 等。驱动是通过总线与硬件设备进行通信，控制硬件进入各种工作状态，获取器件相关寄存器的值，从而得到设备的状态。比如通过驱动可以获取用户操作事件，比如鼠标输入、键盘输入、旋转电子设备等，并将事件转化成数据。

数字版权管理 (digitalrightsmanagement, DRM) 用于对安卓框架底层的媒体类数据进行管理，为安卓平台提供有效的音视频加密、文档加密等版权保护功能，以便应用根据与受版权保护的内容关联的许可限制条件来管理这些内容。

这里结合应用程序缩放显示的场景，示例性说明电子设备 100 软件以及硬件的工作流程。

在一个实施例中，输入事件可以是点击应用界面中的缩放控件，在一个示例中，如图 3a 中所示，缩放控件 306 可以由放大 (+) 控件 307 和缩小 (-) 控件 308 组成。当鼠标 191 接收到点击按键的操作，相应的硬件中断被发给内核层。内核层将点击操作加工成原始输入事件 (包括鼠标点击的坐标，点击操作的时间戳等信息)。原始输入事件被存储在内核层。应用程序框架层的输入管理器从内核层获取原始输入事件，识别该输入事件所对应的控件。比如在应用程序层增加一个 Multi-DPI Entry 控件，即缩放控件，比如点击放大 (+) 控件 307 的事件对应应用程序获取到系统显示像素密度减小一个梯度，点击缩小 (-) 控件 308 的事件对应应用程序获取到系统显示像素密度增大一个梯度，需要注意的是，这里系统显示像素密度的改变并不是真正的发生事件，而是给到应用程序的一个欺骗事件。然后资源管理器会将系统显示像素密度值改变后的处理事件，下发到资源加载管理器，通知应用程序当前的应用资源加载配置项改变，需要对应用资源进行刷新。然后资源加载管理器进一步将事件下发到 Resources 资源模块，应用程序会通过 Resources 资源模块读取相应的应用资源。在应用资源加载的同时，窗口管理器可能需要更新应用窗口的大小或位置，更新显示活动的生命周期，然后在窗口管理器将窗口的大小和位置、活动显示界面调整好之后，显示管理器会进行窗口区域的刷新。在应用资源加载以及窗口活动更新之后，界面管理器会通过 SurfaceFlinger 服务，将应用程序的图形数据按照顺序混合，生成图形缓冲流，并输出到帧缓冲区。帧缓冲区再将图形缓冲流发送至硬件抽象层的硬件合成器，硬件合成器对图形缓冲流进行最终的合成。硬

件合成器将最终的图形数据送至内核层的显示驱动，由显示屏 194 进行显示。

在另一个实施例中，输入事件可以为键盘 190 和鼠标 191 的组合输入事件，例如同时按住键盘 190 中的“ctrl”键和滑动鼠标滚轮。当键盘 190 接收到按压操作和鼠标 191 滑动同时接收到滑动操作，相应的硬件中断被发给内核层。内核层中的鼠标驱动、键盘驱动将该操作加工成原始输入事件，并发送到硬件抽象层的鼠标接口、键盘接口，应用程序框架层的输入管理器从鼠标接口、键盘接口获取原始输入事件，识别该输入事件所对应的指令。例如，同时按住“ctrl”键和上滑鼠标滚轮的事件对应应用程序获取到系统显示像素密度减小一个梯度，同时按住“ctrl”键和下滑鼠标滚轮的事件对应应用程序获取到系统显示像素密度增大一个梯度。需要注意的是，这里系统显示像素密度的改变并不是真正的事件，而是给到应用程序的一个欺骗事件。输入管理器会将该输入事件转发到资源管理接口，然后资源管理器会将系统显示像素密度值改变后的处理事件，下发到资源加载管理器，通知应用程序当前的应用资源加载配置项改变，需要对应用资源进行刷新。然后资源加载管理器进一步将事件下发到 Resources 资源模块，应用程序会通过 Resources 资源模块读取相应的应用资源。在应用资源加载的同时，窗口管理器可能需要更新应用窗口的大小或位置，更新显示活动的生命周期，然后在窗口管理器将窗口的大小和位置、活动显示界面调整好之后，显示管理器会进行窗口区域的刷新。在应用资源加载以及窗口活动更新之后，界面管理器会通过 SurfaceFlinger 服务，将应用程序的图形数据按照顺序混合，生成图形缓冲流，并输出到帧缓冲区。帧缓冲区再将图形缓冲流发送至硬件抽象层的硬件合成器，硬件合成器对图形缓冲流进行最终的合成。硬件合成器将最终的图形数据送至内核层的显示驱动，由显示屏 194 进行显示。

在另一个实施例中，输入事件还可以为触摸事件，当触摸传感器 180K 接收到在应用界面中的触摸操作，相应的硬件中断被发给内核层。内核层将触摸操作加工成原始输入事件（包括触摸坐标，触摸操作的时间戳，触摸拖动的距离等信息）。原始输入事件被存储在内核层。应用程序框架层的输入管理器从内核层获取原始输入事件，识别该输入事件所对应的指令。以该触摸操作是双指捏合操作，该双指捏合操作发生在应用程序的用户界面内为例，输入管理器会将触摸输入事件转发到资源管理接口，比如双指触摸聚拢的事件对应应用程序获取到系统显示像素密度增大一个梯度，双指触摸张开的事件对应应用程序获取到系统显示像素密度减小一个梯度，需要注意的是，这里系统显示像素密度的改变并不是真正的事件，而是给到应用程序的一个欺骗事件。资源管理器会将系统显示像素密度值改变后的处理事件，下发到资源加载管理器，通知应用程序当前的应用资源加载配置项改变，需要对应用资源进行刷新。然后资源加载管理器进一步将事件下发到 Resources 资源模块，应用程序会通过 Resources 资源模块读取相应的应用资源。在应用资源加载的同时，窗口管理器可能会更新应用窗口的大小或位置，更新显示活动的生命周期，然后在窗口管理器将窗口的大小和位置、活动显示界面调整好之后，显示管理器会进行窗口区域的刷新。在应用资源加载以及窗口活动更新之后，界面管理器会通过 SurfaceFlinger 服务，将应用程序的图形数据按照顺序混合，生成图形缓冲流，并输出到帧缓冲区。帧缓冲区再将图形缓冲流发送至硬件抽象层的硬件合成器，硬件合成器对图形缓冲流进行最终的合成。硬件合成器将最终的图形数据送至内核层的显示驱动，由显示屏 194 进行显示。

输入事件并不局限于点击缩放控件的操作、按住键盘按键加滑动鼠标滚轮操作、双指捏合操作等，上述实施例中提及的输入事件仅为示例，并不对本申请构成任何限制，可以实现本申请所阐述的目的即可。

下面介绍一些与本申请相关的术语、概念。

像素 (pixel, PX) 为图像显示的基本单位。每个像素可有各自的颜色值, 可采用三原色显示, 比如可以分成红、绿、蓝三种子像素 (RGB 色域), 或者青、品红、黄和黑 (CMYK 色域)。图像是一个个像素点的集合, 通常情况下, 单位面积内的像素越多代表分辨率越高, 所显示的图像就会接近于真实物体。在电子设备上, 像素数可分为水平像素数和垂直像素数。水平像素数表示水平方向含有的像素点数, 垂直像素数表示垂直方向上含有的像素点数。

屏幕尺寸表示电子设备屏幕的物理尺寸, 可以用屏幕的对角线的长度表示, 单位是英寸, 1 英寸=2.54 厘米, 比如常见的手机屏幕尺寸有 3.5 英寸、4.2 英寸、5.0 英寸、5.5 英寸、6.0 英寸等, 常见的 PC 的显示器屏幕尺寸有 15.6 英寸, 20.0 英寸, 27.0 英寸等。

分辨率是指横向、纵向上的像素点数, 单位是 px, 1px=1 个像素点。分辨率可以确定显示多少信息, 以水平像素数和垂直像素数进行衡量, 即分辨率=水平像素数*垂直像素数, 如 1960*1080。就相同物理尺寸大小的图像而言, 当分辨率相对较低时 (例如 640*480), 显示的像素少, 单个像素尺寸比较大, 显示效果较为粗糙; 当分辨率相对较高时 (例如 1600*1200), 显示的像素多, 单个像素尺寸比较小, 显示效果比较精细。

屏幕像素密度 (pixels per inch, PPI), 可以用来表示屏幕每英寸所拥有的物理像素数。屏幕分辨率描述的是显示屏自身的像素点的数量, 是显示器固有的硬件物理特性, 是不可改变的。PPI 的一种计算方法是: $ppi = \frac{\sqrt{height^2 + width^2}}{size}$, height 和 width 分别是显示屏高和屏宽对应的像素点的数量, size 表示显示屏的对角线长度, 单位为英寸, 即该计算方法表示的是根据显示屏的高和宽的像素点数, 通过勾股定理计算出斜对角线的像素点数, 再除以显示屏对角线尺寸, 得到 PPI。PPI 值越高, 即表示显示屏能够以越高的密度显示图像, 拟真度就越高, 越接近真实的图像。

显示像素密度 (dots per inch, DPI), 是指每英寸上可取样、显示、输出的像素点数。它描述的是软件显示时的像素密度, 是一种软件属性, 可以进行配置。DPI 与图像尺寸和图像分辨率有关。在图像尺寸相同的情况下, 图像分辨率越高, DPI 越大; 在图像分辨率相同的情况下, 图像尺寸越小, DPI 越大。DPI 值越高, 图像就越清晰。一种 DPI 的计算方式可以由图像的像素点数比物理尺寸得到, 另外还可以通过代码获取到 DPI 值:

```
float xdpi=getResources().getDisplayMetrics().xdpi;
float ydpi= getResources().getDisplayMetrics().ydpi;
```

其中 xdpi 表示横向 dpi, ydpi 表示纵向 dpi, 一般情况下所获取的 xdpi 和 ydpi 基本一样, 因为一个电子设备显示屏的屏幕像素点是追求平均分布的, 所以无论是宽、高、还是斜边, 其 dpi 应该都是一样的, 当然由于技术限制, 可能会有些误差。

密度独立像素 (density independent pixels, dp), 又称密度无关像素, 指的是在 160dpi 的屏幕上, 1 个像素所占的长度。dpi 是屏幕像素密度, 假如一英寸里面有 160 个像素, 这个屏幕的像素密度就是 160dpi, 在 Android® 中, 规定以 160dpi 为基准, 1dp=1px = 1/160 inch, 如果密度是 320dpi, 则 1dp=2px, 以此类推。假如同样都是画一条 320px 的线, 在 480*800 分辨率手机上显示为 2/3 屏幕宽度, 在 320*480 的手机上则占满了全屏, 如果使用 dp 为单位, 在这两种分辨率下, 160dp 都显示为屏幕一半的长度。

缩放独立像素 (scale-independent pixels, sp), 与 dp 类似, 但是 sp 用于文字单位, 可以

根据系统设定缩放文字大小，是设置字体大小的常用单位。

目前主流使用的像素密度有 mdpi、hdpi、xdpi、xxdpi、xxxdpi，业界区分标准如下表 1 所示。

名称	像素密度范围
ldpi	(0dpi, 120dpi)
mdpi	[120dpi, 160dpi)
hdpi	[160dpi, 240dpi)
xhdpi	[240dpi, 320dpi)
xxhdpi	[320dpi, 480dpi)
xxxhdpi	[480dpi, 640dpi)

表 1 DPI 区分标准

系统显示分辨率在本实施例中可以指的是电子设备的实际显示界面所采用的分辨率。系统显示分辨率可以是电子设备根据检测到的显示器的大小默认适配的系统显示分辨率，也可以为用户自定义设置的系统显示分辨率。通常情况下，用户可以在系统设置选项中选择不同的系统显示分辨率。一般来说，电子设备的系统显示分辨率等于或小于显示屏的屏幕分辨率，屏幕分辨率描述的是显示屏自身的像素点的数量，是显示器固有的，不可改变的。系统显示像素密度，又称系统显示 dpi，可以根据屏幕尺寸和系统显示分辨率计算得出。

关于资源是如何根据系统显示像素密度加载相应资源的，这里拿图片加载举个例子，相关资源文件可以位于的 drawable 文件夹及 values 文件夹中，用来区分不同像素密度下的图片和 dimen 值。首先获取到系统显示 dpi，然后根据表 1，找出对应的像素密度。比如某个显示设备的屏幕分辨率为 1080*1920，屏幕像素密度为 400dpi，有一张图片其大小为 270*480 像素，在这里系统显示 dpi 为 400dpi，根据表 1，则对应 xxhdpi，系统会自动优先在 drawable-xxhdpi 文件夹中找对应的图片，如果找到了就加载，此时上述图片在显示设备上显示的就是它本身的大小，也就是 270*480 像素；如果未找到，系统就会去更高 dpi 的文件夹 xxxhdpi 中找，一直找到最高也没有的话，就会查找 drawable-nodpi 文件夹，还是没有就开始依次查询低分辨率的文件夹，由高到低一直查到 ldpi。

如果将 xxhdpi 下的图片剪切到 drawable-mdpi 中，可以发现显示的图片明显变大。因为系统在加载图片时，首先去对应的文件夹查找，没找到，又依次按照顺序一直在 drawable-mdpi 找到，但是系统会认为这张图是专门为低密度的设备所设计的，如果直接将这张图在当前的高密度设备上使用就有可能出现像素过低的情况，于是系统自动做了这样一个放大操作。以 mdpi 为基准，缩放倍数可以表示为：drawable-mdpi：drawable-hdpi：drawable-xhdpi：drawable-xxhdpi：drawable-xxxhdpi=1：1.5：2：3：4，即 dpi 范围的最大值之比。

关于应用程序是如何加载应用资源的，在一些实施例中，通常会通过 getResources()去获取 Resources 对象，Resource 对象是应用进程内的一个全局对象，它用来访问应用的资源。除了 Resources 对象我们还可以通过 getAsset()获取 AssetManger 来读取指定文件路径下的文件。Resource 与 AssetManger 就构造了资源访问框架的基础。

本申请实施例中的术语“用户界面 (user interface, UI)”，是应用程序或操作系统与用户之间进行交互和信息交换的介质接口，它实现信息的内部形式与用户可以接受形式之间的转换。用户界面常用的表现形式是图形用户界面 (graphic user interface, GUI)，是指采用图形方式显示的与计算机操作相关的用户界面。它可以是在电子设备的显示屏中显示的一个图标、窗口、控件等界面元素，其中控件可以包括图标、按钮、菜单、选项卡、文本框、对话框、

状态栏、导航栏、Widget 等可视的界面元素。

下面结合附图介绍本申请涉及的一些人机交互界面的实施例。

本申请是在电子设备 100 检测到在第一应用程序界面内发生第一用户操作时，第一用户操作对应生成一个“虚拟的”系统显示像素密度改变的事件。第一用户操作又可称为第一输入操作。在本申请的一些实施例中，“虚拟的系统显示像素密度”是指这个系统显示像素密度并不是当前系统界面显示所采用的真实像素密度，而是为了让第一应用程序用户界面发生缩放而给到第一应用程序的一个欺骗值。第一应用程序根据获取到的“虚拟的”系统显示像素密度加载对应的应用资源，第一应用程序的用户界面也随之发生改变，比如显示的字体、控件、图片等放大或缩小。其中，第一用户操作可以是键盘输入事件或鼠标输入事件，键盘或鼠标输入事件可以生成对应的递变值，例如鼠标点击表示放大（+）、缩小（-）的缩放控件，或者同时按住“ctrl”键和上滑/下滑鼠标滚轮，或者同时按住“ctrl”键和“▲”/“▼”键等等；也可以是触摸事件，例如在触摸屏上，用户两只手指向相反方向聚拢/张开触摸屏幕；还可以是手势操作事件，例如隔空识别特殊手势；或者用户可以自定义第一用户操作等等。上述情况仅为示例，本申请实施例对第一用户操作不作任何限制，能够实现本申请阐述的目的即可。其中，第一应用程序可以对一个或多个不同类型的用户操作事件响应，上述举例的用户操作之间并不一定构成冲突，即多种用户操作方式可以同时作为触发应用界面的缩放功能的事件，比如用户可以点击缩放控件来实现应用界面的缩放，以及也可以通过按住“ctrl”键和滑动鼠标滚轮来实现应用界面的缩放，同时还可以在触控屏上通过手指捏合的操作，或者特定的手势操作等等实现应用界面的缩放，本申请对此不作任何限制。

在一些实施例中，用户可以通过点击缩放控件来实现界面的缩放显示。图 3a 示出了一个点击缩放控件 306 作为用户输入事件的场景。如图 3a 所示，在电子设备的用户界面 301 中，可以显示有 dock 栏 305，以及第一应用程序的用户界面 302，又称为应用界面 302，在这里，以第一应用程序为备忘录作为示例说明本实施例。dock 栏 305 中可以显示有锁定的应用程序的图标和最近启动的应用程序的图标，方便用户快速启动应用程序。本申请实施例并不对电子设备所使用的操作系统有任何限制，应用的操作系统可以是 Android[®]，也可以是 iOS、Linux 等其他操作系统，以可以实现本申请阐述的目的即可。

应用界面 302 又可以包括标题栏 303 和应用内容显示区 304。不同应用的用户界面的组成也是可能不尽相同的，有些应用可能划分为更多的功能区，比如还可能包括工具栏、菜单栏等等。

在应用界面 302 内，可以显示有缩放控件 306，缩放控件 306 可以由放大（+）控件 307 和缩小（-）控件 308 组成。其中缩放控件可以是一个悬浮控件，悬浮控件是指在系统中开发的一种控件，显示效果是悬浮于整个用户界面上，不必局限于某一个应用窗口之内，点击该悬浮缩放控件，当前的活动窗口就可以响应对应的指令。该缩放控件还可以是内嵌于应用窗口中的控件，控件的显示周期绑定于当前应用程序窗口的生命周期，例如第一应用程序包含多套应用资源，并且应用资源的跳转加载的适配效果较好，在该第一应用程序的应用窗口内就可以显示有缩放控件，点击该缩放控件就可以控制缩放显示第一应用程序的界面；如果第一应用程序并没有多套应用资源，或者对应用资源的跳转加载的适配效果不好，那么在该第一应用程序的应用窗口内就可以不显示缩放控件，以免给用户造成困扰，这样用户界面的设计更加人性化。关于缩放控件 306 的样式，本申请不作任何限制，图 3a 仅为一个示例，具体以能实现本实施例所阐述的功能即可。

如图 3a 所示, 点击放大控件 307, 第一应用程序会获取到系统显示像素密度递变到下一个低于当前系统显示像素密度的值, 然后第一应用程序会加载对应的应用资源, 应用界面 302 相应的改变为放大的应用界面 309。应用界面 309 相比于应用界面 302, 其中显示的文字、图片、控件等等都放大了, 包括放大的标题栏 310 和放大的应用内容显示区 311, 与应用程序自身的缩放功能有所区别。有一些应用程序自身有缩放的功能, 例如应用程序 Word、Excel, 通常缩放只是缩放应用内容显示区, 标题栏、菜单栏、工具栏等部分并不会同比例缩放。

缩放控件 306 并不一定在每一个应用程序的应用界面中都显示, 比如, 当检测到该应用程序有多套应用资源可以实现在不同应用资源之间切换加载时, 该应用程序的用户界面中可以显示缩放控件 306; 如果某些应用程序不支持多套应用资源, 或者某些应用程序的应用资源切换加载时容易使应用崩溃, 这些应用程序的用户界面中可以不显示缩放控件 306, 以免给用户造成不必要的困扰。关于哪些应用程序的应用界面中可以显示有缩放控件 306, 可以对应用程序做一个分类, 例如将对本申请所述的缩放功能适配好的应用程序分类到“白名单”中, 将没有多套应用资源或者对本申请所述的缩放功能适配不好的应用程序分类到“黑名单”中, “白名单”中的应用程序的应用界面就可以显示有缩放控件 306, “黑名单”中的应用程序的应用界面就没必要显示缩放控件 306, 本申请对此不作任何限制。

缩小应用界面与放大应用界面同理, 如图 3b 所示, 点击缩小控件 308, 第一应用程序会获取到系统显示像素密度递变到下一个高于当前系统显示像素密度的值, 然后第一应用程序会加载对应的应用资源, 应用界面 312 相应的改变为缩小的应用界面 313。相应的, 应用界面 313 相比于应用界面 312, 其中显示的文字、图片、控件等等都缩小了。

如图 3c 所示, 在用户第一次打开第一应用程序时, 应用界面 314 中可以显示有引导语 315, 比如引导语 315 可以是“新增应用界面缩放功能, 点击试试吧!”并指向缩放控件 306, 引导用户了解缩放功能。图 3c 所示的引导界面仅为一个示例, 不限于引导语, 还可以通过图片、动画演示等引导用户了解缩放功能, 本申请对此不作任何限制, 能实现引导用户的目的即可。

不限于点击缩放控件 306, 第一用户操作还可以是同时按住“ctrl”键和上滑鼠标滚轮, 或者同时按住“ctrl”键和“▲”键等等。图 4a 示出了一个同时按住“ctrl”键 407 和上滑鼠标滚轮 406 作为用户输入事件的场景。

如图 4a 所示, 在电子设备的用户界面 401 中, 可以显示有第一应用程序的用户界面 402, 又称为应用界面 402, 电子设备外接有键盘和鼠标 405。当鼠标指针 403 显示在应用界面 402 中时, 用户同时按住“ctrl”键 407 和上滑鼠标滚轮 406, 这时第一应用程序会获取到系统显示像素密度递变到下一个低于当前系统显示像素密度的值, 然后第一应用程序会加载对应的应用资源, 应用界面 402 相应的改变为放大的应用界面 408。应用界面 408 相比于应用界面 402, 其中显示的文字、图片、控件等等都放大了。另外, 在应用界面 402 中, 还可以显示有缩放控件 404, 即可以通过点击缩放控件 404 来实现应用界面的缩放, 同时也可以通过按住“ctrl”键 407 和滑动鼠标滚轮来实现应用界面的缩放, 二者并不冲突, 两种缩放方式可以同时存在。

缩小应用界面与放大应用界面同理, 如图 4b 所示, 当鼠标指针 403 显示在应用界面 409 中时, 用户同时按住“ctrl”键 407 和下滑鼠标滚轮 410, 这时第一应用程序会获取到系统显示像素密度递变到下一个高于当前系统显示像素密度的值, 然后第一应用程序会加载对应的应用资源, 应用界面 409 相应的改变为缩小的应用界面 411。应用界面 411 相比于应用界面 409, 其中显示的文字、图片、控件等等都缩小了。

同样的，不一定每一个应用程序都会响应“ctrl”键和滑动鼠标滚轮的事件，如前述实施例，可以对应用程序做一个分类，比如设置“黑名单”、“白名单”，将对本申请所述的缩放功能适配好的应用程序分类到“白名单”中，将没有多套应用资源或者对本申请所述的缩放功能适配不好的应用程序分类到“黑名单”中，“白名单”中的应用程序的应用界面就可以响应“ctrl”键和滑动鼠标滚轮的事件，“黑名单”中的应用程序的应用界面就不会响应“ctrl”键和滑动鼠标滚轮的事件。这里仅为举例说明，本申请对此不作任何限制。

如图 4c 所示，在用户第一次打开第一应用程序时，应用界面 412 中可以显示有引导语 413，比如引导语 413 可以是“新增应用界面缩放功能，同时按住 ctrl 键和滑动鼠标滚轮可以快捷应用界面”，引导用户了解缩放功能。应用界面 412 中还可以同时显示有对话框 414 “我知道了”，当用户点击对话框 414 “我知道了”，表示用户已知晓该缩放功能，此时引导语 413 消失，第一应用程序进入正常的用户界面。图 4c 所示的引导界面仅为一个示例，不限于引导语，还可以通过图片、动画演示等引导用户了解缩放功能，本申请对此不作任何限制，能实现引导用户的目的即可。

图 5a 示出了一个在触控屏上用户手指捏合操作作为用户输入事件的场景。如图 5a 所示，在电子设备的用户界面 501 中，可以显示有第一应用程序的用户界面 502，又称为应用界面 502，其中电子设备的显示屏为可触控屏幕，可以响应触控操作。当用户两个手指同时触摸屏幕的应用界面 502 区域内，并向相反方向张开 503 一定距离，这时第一应用程序会获取到系统显示像素密度递变到下一个低于当前系统显示像素密度的值，然后第一应用程序会加载对应的应用资源，应用界面 502 相应的改变为放大的应用界面 504。应用界面 504 相比于应用界面 502，其中显示的文字、图片、控件等等都放大了。另外，在应用界面 502 中，还可以显示有缩放控件 505，即可以通过多种方式实现应用界面的缩放，比如前述点击缩放控件 505 来实现应用界面的缩放，以及也可以通过按住“ctrl”键和滑动鼠标滚轮来实现应用界面的缩放，以及手势操作，触控操作等等，这几种方式之间并不冲突。

缩小应用界面与放大应用界面同理，如图 5b 所示，当用户两个手指同时触摸屏幕的应用界面 506 区域内，并向相反方向聚拢 507 一定距离，这时第一应用程序会获取到系统显示像素密度递变到下一个高于当前系统显示像素密度的值，然后第一应用程序会加载对应的应用资源，应用界面 506 相应的改变为缩小的应用界面 508。应用界面 508 相比于应用界面 506，其中显示的文字、图片、控件等等都缩小了。

同样的，不一定每一个应用程序都会响应用户手指捏合操作，如前述实施例，可以对应用程序做一个分类，比如设置“黑名单”、“白名单”，将对本申请所述的缩放功能适配好的应用程序分类到“白名单”中，将没有多套应用资源或者对本申请所述的缩放功能适配不好的应用程序分类到“黑名单”中，“白名单”中的应用程序的应用界面就可以响应用户手指捏合操作，“黑名单”中的应用程序的应用界面就不会响应用户手指捏合操作。这里仅为举例说明，本申请对此不作任何限制。

如图 5c 所示，在用户第一次打开第一应用程序时，应用界面 509 中可以显示有引导语 510，比如引导语 510 可以是“新增应用界面缩放功能”并给出放大、缩小应用界面对应的用户操作引导图 511，引导用户了解缩放功能。应用界面 509 中还可以同时显示有对话框 512 “我知道了”，当用户点击对话框 512 “我知道了”，表示用户已知晓该缩放功能，此时引导语 510 和引导图 511 消失，第一应用程序进入正常的用户界面。图 5c 所示的引导界面仅为一个示例，图 5c 所示的引导界面并不对本申请构成任何限制，能实现引导用户的功能即在本申请保护范围之内。

图 6a 示出了一个多窗口下响应缩放操作的场景。在一些实施例中，如图 6a 所示，在电子设备的用户界面 601 中，可以显示有第一应用程序的用户界面 602 和第二应用程序的用户界面 603，又可称为第一应用界面 602 和第二应用界面 603。其中，鼠标指针位于第一应用界面 602 内，第一应用程序的窗口为当前的活动窗口，活动窗口是指应用程序的顶层窗口，也就是表示当前正在使用的窗口，一般来说，系统中只有一个顶层窗口可以是活动窗口，用户可以通过单击窗口，或者也可以使用特殊设置的按键，比如 ALT+TAB 或 ALT+ESC 组合键来激活一个顶层窗口，或者还可以通过调用函数 SetActiveWindow 来激活一个顶层窗口。

在一些实施例中，在用户界面 601 中，还可以显示有缩放控件 604，用户可以通过点击缩放控件 604 来实现应用界面的缩放。在图 6a 中，用户可以控制鼠标指针 605 点击缩放控件 604 中的放大（+）控件，然后当前的活动窗口，即第一应用程序会获取到系统显示像素密度递变到下一个低于当前系统显示像素密度的值，然后第一应用程序会加载对应的应用资源，第一应用界面 602 中的界面元素就会相应的放大，比如显示为第一应用界面 606。放大显示后的第一应用界面 606 相比于第一应用界面 602，其中显示的文字、图片、控件等等都放大了。而第二应用界面 603 并不是当前的活动窗口，所以不会响应放大操作，因此第二应用界面 602 的界面元素的显示大小不变，即放大操作后显示的第二应用界面 607 与第二应用界面 603 保持一致。

缩小应用界面与放大应用界面同理，如图 6b 所示，在用户界面 601 中，可以显示有第一应用界面 608 和第二应用界面 609 以及缩放控件 604。当用户控制鼠标指针 608 点击缩放控件 604 中的缩小（-）控件后，当前的活动窗口，即第一应用程序会获取到系统显示像素密度递变到下一个高于当前系统显示像素密度的值，然后第一应用程序会加载对应的应用资源，第一应用界面 608 中的界面元素就会相应的缩小，比如显示为第一应用界面 610。缩小显示后的第一应用界面 610 相比于第一应用界面 608，其中显示的文字、图片、控件等等都缩小了。而第二应用界面 609 并不是当前的活动窗口，所以不会响应缩小操作，因此第二应用界面 609 的界面元素的显示大小不变，即缩小操作后显示的第二应用界面 611 与第二应用界面 610 保持一致。

可以理解，图 3a、图 3b、图 3c、图 4a、图 4b、图 4c、图 5a、图 5b、图 5c、图 6a、图 6b 示出的应用界面并不对本申请其他实施例构成限制，在本申请的其他实施例中，可以增加、减少、改变不同样式的部件或采用其他的实现方式，并不限于前述实施例，具体实现可以根据具体情况进行调整和设置，本申请对此不作任何限制，可以实现本申请阐述的缩放应用界面的功能即可。

基于前述的一些实施例，下面介绍本申请提供的一种应用程序界面显示的方法。

实施例一

参见图 7，图 7 是本申请一个实施例提供的应用程序界面显示的方法的流程图。如图 7 所示，该方法可包括：

S101，电子设备可以显示第一应用程序的第一窗口和第二应用程序的第二窗口，其中，第一窗口和第二窗口中显示的界面元素采用第一显示像素密度。

第一显示像素密度又可称为第一像素密度。本实施例可参考图 6a、图 6b 相关实施例的描述。在电子设备显示多窗口的场景中，比如电子设备可以显示有第一应用程序的第一窗口和第二应用程序的第二窗口，其中，第一窗口为当前的活动窗口，第一窗口和第二窗口中显示的界面元素可以采用第一显示像素密度，第一显示像素密度可以为当前系统所采用的显示

像素密度。

第一应用程序可以根据获取到的第一显示像素密度加载第一应用资源，第一应用程序的显示界面中的界面元素引用于第一应用资源，第一应用资源为针对第一显示像素密度而配置的资源文件。

应用资源可以包括图片、字符串、控件等等。对于大多数的应用程序，开发人员可以提供不同的资源，用来根据不同的系统显示像素密度来进行适配显示。比如对于同一张图片 image.png，可以提供 hdpi, mdpi 和 ldpi 三个版本，它们都必须以相同的名字存储在各个 drawable 目录下。当应用程序运行时，系统会根据当前电子设备的系统显示像素密度来加载对应的资源。

第一应用程序可以拥有一套或多套应用资源，第一应用程序可以根据获取到的系统显示像素密度来加载对应的应用资源。通常来说，对应的应用资源是第一应用程序根据系统显示像素密度，适配最佳显示效果的应用配置文件。

S102，电子设备检测到第一输入操作，第一输入操作用于确定第二显示像素密度，第二显示像素密度不同于第一显示像素密度。

第一输入操作又称第一用户操作，第一输入操作可以是键盘输入事件或鼠标输入事件，可以生成对应的递变值，第一输入操作可以是放大操作或者缩小操作，例如鼠标点击第一应用程序的用户界面内表示放大（+）、缩小（-）的缩放控件，点击放大（+）控件为放大操作，点击缩小（-）控件为缩小操作。其中缩放控件可以是一个悬浮控件，该悬浮控件浮于整个用户界面之上，不必局限于某一个应用窗口之内，点击该悬浮缩放控件，当前的活动窗口就可以响应对应的指令。该缩放控件还可以是内嵌于应用窗口中的控件，控件的显示周期绑定于当前应用程序窗口的生命周期，例如第一应用程序包含多套应用资源，并且应用资源的跳转加载的适配效果较好，在该第一应用程序的应用窗口内就可以显示有缩放控件，点击该缩放控件就可以控制缩放显示第一应用程序的界面；如果第一应用程序并没有多套应用资源，或者对应用资源的跳转加载的适配效果不好，那么在该第一应用程序的应用窗口内就可以不显示缩放控件，以免给用户造成困扰，这样用户界面的设计更加人性化。

除了点击缩放控件，放大操作/缩小操作也可以是在鼠标指针位于第一应用程序的用户界面中时，同时按住“ctrl”键和上滑/下滑鼠标滚轮，或者同时按住“ctrl”键和“▲”/“▼”键等等；放大操作/缩小操作还可以是触摸事件，例如在触摸屏或触控板的第一应用程序的用户界面对应位置中，用户两只手指向相反方向张开/聚拢触摸屏幕并达到第一距离值，第一距离值例如可以为1厘米；放大操作/缩小操作还可以是手势操作事件，例如隔空识别特殊手势等等，具体可以参考前述图3a、图3b、图4a、图4b、图5a、图5b、图6a、图6b所示的实施例。第一输入操作用于确定第二显示像素密度，第二显示像素密度又可称为第二像素密度。上述情况仅为示例，本申请实施例对第一输入操作不作任何限制，能够实现本申请阐述的目的即可。

需要说明的是，在本实施例中，第一输入操作对应生成的第二显示像素密度为一个“欺骗值”，并不是当前系统真实设置的系统显示像素密度，当前系统的显示界面所采用的仍为第一显示像素密度。

S103，第一应用程序获取第二显示像素密度。

第一应用程序的第一窗口可以为当前的活动窗口，因此第一应用程序响应于第一输入操作，获取到第二显示像素密度。

需要说明的是，在本实施例中，第一应用程序获取到的第二显示像素密度并不是当前系

统界面显示所真实采用的第一显示像素密度，而是为了让第一应用程序用户界面发生界面缩放而给到第一应用程序的一个“欺骗”值。如果第一输入操作为放大操作，则第二显示像素密度小于第一显示像素密度。如果第一输入操作为缩小操作，则第二显示像素密度大于第一显示像素密度。

通常情况下，应用程序可以采用的像素密度为以下之一：120dpi、160dpi、240dpi、320dpi、480dpi、640dpi。

在一些实施例中，应用程序可以通过调用 updateConfiguration 方法重新加载应用资源，具体地，可以通过 getResources()函数和 getAssets()函数获取到 Resource 对象和 AssetManager 对象，然后调用 setConfiguration()重新加载第二应用资源。

具体地，在 ResourcesImpl 中的 updateConfiguration 处主要实现应用程序 DPI 的改变，系统配置变化时就会触发 Configuration 模块刷新，示例程序代码如下：

```
String currentPackageName= android. app. ActivityThread. currentPackageNwe();
if (currentPackageName != null) {
    if (!mNoSupportMultiDPIWindow. containsKey(currentPackageName)) {
        mConfiguration. densityDpi = MIDDIE_DPI;
        mMetrics. densityDpi = MIDDIE_DPI;
    }
}
```

然后将改变的应用程序 DPI，即 densityDpi，更新到资源加载管理器（AssetManager）的 mAsset 函数中，示例程序代码如下：

```
mAssets.setConfiguration(mConfiguration.mcc, mConfiguration.mnc,
    adjustLanguageTag(mConfiguration.getLocales().get(0).toLanguageTag()),
    mConfiguration.orientation,
    mConfiguration.touchscreen,
    mConfiguration.densityDpi, mConfiguration.keyboard,
    keyboardHidden, mConfiguration.navigation, width, height,
    mConfiguration.smallestScreenWidthDp,
    mConfiguration.screenWidthDp, mConfiguration.screenHeightDp,
    mConfiguration.screenLayout, mConfiguration.uiMode,
    mConfiguration.colorMode, Build.VERSION.RESOURCES_SDK_INT);
```

资源加载管理器可以进一步将该更新应用资源事件下发到 Resources 资源模块，应用程序最终会通过 Resources 资源模块读取相应的应用资源。

S104，电子设备显示第一应用程序的第三窗口和第二应用程序的第二窗口，其中，第三窗口中显示的界面元素采用第二显示像素密度，第二窗口中显示的界面元素采用第一显示像素密度。

响应于第一输入操作，第一应用程序根据获取到的第二显示像素密度更新应用配置并加载第二应用资源，显示第一应用程序的第三窗口，第三窗口的显示界面中的界面元素引用于第二应用资源，第二应用资源为针对第二显示像素密度而配置的资源文件。

如果第一输入操作为放大操作，则第三窗口中显示的界面元素的尺寸大于第一窗口中显示的界面元素的尺寸。如果第一输入操作为缩小操作，则第三窗口中显示的界面元素的尺寸小于第一窗口中显示的界面元素的尺寸。

第二应用程序的第二窗口并不是当前的活动窗口，因此第二窗口中显示的界面元素尺寸的大小不变，仍然采用第一显示像素密度。

实施例二

参见图 8，图 8 是本申请另一个实施例提供的应用程序界面显示的方法的流程图。如图 8 所示，该方法可包括：

S201，电子设备显示第一用户界面，第一用户界面包括第一区域和第二区域，其中，第一区域中显示的界面元素来自电子设备的操作系统，第二区域中显示的界面元素来自第一应用程序，第一区域、第二区域的显示像素密度为第一系统显示像素密度，第二区域中显示的界面元素所采用的第一应用资源为针对第一系统显示像素密度所配置的资源文件。

第一用户界面可以是电子设备的显示器所显示的用户界面。第一用户界面可以包括第一区域和第二区域，其中，第一区域可以为电子设备桌面的全部或部分显示界面，第二区域为第一应用程序的显示界面。例如，第一区域中显示的界面元素包括以下任意一项或多项：任务栏、通知栏、应用程序图标、快速启动工具栏、开始按钮、桌面背景图片、时间栏、网络信号状态指示图标、电量指示图标、声音指示图标。

第一系统显示像素密度在本实施例中可以指的是电子设备的操作系统所采用的显示像素密度，即第一区域的界面实际显示所采用的显示像素密度。

第一应用程序根据获取到的第一系统显示像素密度加载第一应用资源，第一应用程序的显示界面中的界面元素引用用于第一应用资源，第一应用资源为针对第一系统显示像素密度而配置的资源文件。

应用资源可以包括图片、字符串、控件等等。对于大多数的应用程序，开发人员可以提供不同的资源，用来根据不同的系统显示像素密度来进行适配显示。比如对于同一张图片 image.png，可以提供 hdpi，mdpi 和 ldpi 三个版本，它们都必须以相同的名字存储在各个 drawable 目录下。当应用程序运行时，系统会根据当前电子设备的系统显示像素密度来加载对应的资源。

第一应用程序可以拥有一套或多套应用资源，第一应用程序可以根据获取到的当前系统显示像素密度来加载对应的应用资源。通常来说，对应的应用资源是第一应用程序根据系统显示像素密度，适配最佳显示效果的应用配置文件。

S202，电子设备检测到第一输入操作，第一输入操作对应生成第二系统显示像素密度。

第一输入操作又称第一用户操作，第一输入操作可以是键盘输入事件或鼠标输入事件，可以生成对应的递变值，第一输入操作可以是放大操作或者缩小操作，例如鼠标点击第一应用程序的用户界面内表示放大（+）、缩小（-）的控件，点击放大（+）控件为放大操作，点击缩小（-）控件为缩小操作；放大操作/缩小操作也可以是在鼠标指针位于第一应用程序的用户界面中时，同时按住“ctrl”键和上滑/下滑鼠标滚轮，或者同时按住“ctrl”键和“▲”/“▼”键等等；放大操作/缩小操作还可以是触摸事件，例如在触摸屏或触控板的第一应用程序的用户界面对应位置中，用户两只手指向相反方向张开/聚拢触摸屏幕并达到第一距离值，第一距离值例如可以为 1 厘米；放大操作/缩小操作还可以是手势操作事件，例如隔空识别特殊手势等等，具体可以参考前述图 3a、图 3b、图 4a、图 4b、图 5a、图 5b 所示的实施例。上述情况仅为示例，本申请实施例对第一输入操作不作任何限制，能够实现本申请阐述的目的即可。

需要说明的是，在本实施例中，第一输入操作对应生成的第二系统显示像素密度为一个

“欺骗值”，并不是当前系统真实设置的系统显示像素密度，当前系统的显示界面所采用的仍为第一系统显示像素密度。

S203，第一应用程序根据获取到的第二系统显示像素密度加载第二应用资源，第二应用资源为针对第二系统显示像素密度所配置的资源文件。

需要说明的是，在本实施例中，第一应用程序获取到的第二系统显示像素密度并不是当前系统界面显示所真实采用的第一系统显示像素密度，而是为了让第一应用程序用户界面发生界面缩放而给到第一应用程序的一个“欺骗”值。如果第一输入操作为放大操作，则第二系统显示像素密度小于第一系统显示像素密度。如果第一输入操作为缩小操作，则第二系统显示像素密度大于第一系统显示像素密度。

关于应用程序是如何根据系统显示像素密度更新应用配置，重新加载资源的代码实现示例，可以参考前述步骤 S103，这里不再赘述。

S204，电子设备显示第二用户界面，第二用户界面包括第一区域和第二区域，其中，第一区域的显示像素密度为第一系统显示像素密度，第二区域的显示像素密度为第二系统显示像素密度。

第二用户界面可以是电子设备响应第一输入操作后所显示的用户界面。第二用户界面可以包括第一区域和第二区域，其中，第一区域可以为电子设备桌面的全部或部分显示界面，第二区域为第一应用程序的显示界面。

第一区域的界面实际显示所采用的显示像素密度仍为第一系统显示像素密度。

响应于第一输入操作，第一应用程序根据获取到的第二系统显示像素密度加载第二应用资源，第一应用程序的显示界面中的界面元素引用于第二应用资源，第二应用资源为针对第二系统显示像素密度而配置的资源文件。如果第一输入操作为放大操作，则第一输入操作后第二区域中显示的界面元素的尺寸大于第一输入操作之前显示的界面元素的尺寸。如果第一输入操作为缩小操作，则第一输入操作后第二区域中显示的界面元素的尺寸小于第一输入操作之前显示的界面元素的尺寸。

在一些实施例中，有些应用程序没有配置多套应用资源，比如安卓系统桌面启动器 launcher，或者有些应用程序对本申请所述的缩放功能适配不好，例如应用程序在加载应用资源时卡顿严重或容易崩溃，这种情况下，可以对应用程序做一个分类，比如设置应用程序缩放显示“黑名单”、“白名单”，将对本申请所述的缩放功能适配好的应用程序分类到“白名单”中，“白名单”中的应用程序就可以呈现本申请前述实施例中所描述的应用程序界面的缩放功能；将没有多套应用资源或者对本申请所述的缩放功能适配不好的应用程序分类到“黑名单”中，“黑名单”中的应用程序的应用界面就不会响应前述实施例所示的缩放应用界面的用户操作，以及可以不在用户界面中显示缩放控件，以免对用户产生误导，进而这些应用程序界面也不会发生缩放显示。例如，当检测到应用程序 A 在所设置的“黑名单”中，应用程序 A 的用户界面中就取消显示缩放控件，或者在检测到用户的鼠标或键盘或触摸等对应的缩放第一用户操作时，系统不会将该输入事件上报给应用程序 A，应用程序 A 即不会实现缩放显示的效果。例如，安卓系统桌面启动器 launcher 以及系统预置应用程序不支持多套 DPI 资源，将其分类到“黑名单”，示例程序代码如下：

```
static HashMap<String, Integer>mNoSupportMultiDPIWindow = new HashMap <>();
static final int DEFAULT_MULTI_DPI = 2;
private static final int SMALLEST_DPI = 240;
private static final int MIDDLE_DPI = 320;
```

```
private static final int LARGE_DPIPI = 400;
static {
    mNoSupportMultiDPIWindow. clear();
        mNoSupportMultiDPIWindow. put ("com. huawei. android. launcher",
DEFAULT_MULTI_DPI);
        mNoSupportMultiDPIWindow. put ("com. Android. systemui", DEFAULT_MULTI_DPI);
    }
}
```

具体的分类情况，开发人员可以根据具体需求进行适应调整，这里的分类方法仅为举例说明，本申请对此不作任何限制。

实施本申请方法的实施例，可以使得应用程序的用户界面实现快捷的缩放显示，满足用户对于应用界面中的显示内容放大或缩小的需求，给用户提供了一个友好的操作环境，提升了用户视觉体验，并且可以便利开发人员的开发工作量，节省了应用程序的开发资源。

以上所述的具体实施方式，对本申请的目的、技术方案和有益效果进行了进一步详细说明，所应理解的是，以上所述仅为本申请的具体实施方式而已，并不用于限定本申请的保护范围，凡在本申请的技术方案的基础之上，所做的任何修改、等同替换、改进等，均应包括在本申请的保护范围之内。

本领域内的技术人员应明白，本申请的实施例可提供为方法、系统、或计算机程序产品。因此，本申请可采用完全硬件实施例、完全软件实施例、或结合软件和硬件方面的实施例的形式。而且，本申请可采用在一个或多个其中包含有计算机可用程序代码的计算机可用存储介质（包括但不限于磁盘存储器和光学存储器等）上实施的计算机程序产品的形式。

本申请是参照根据本申请实施例的方法、设备（系统）、和计算机程序产品的流程图和/或方框图来描述的。应理解可由计算机程序指令实现流程图和/或方框图中的每一流程和/或方框、以及流程图和/或方框图中的流程和/或方框的结合。可提供这些计算机程序指令到通用计算机、专用计算机、嵌入式处理机或其他可编程数据处理设备的处理器以产生一个机器，使得通过计算机或其他可编程数据处理设备的处理器执行的指令产生用于实现在流程图一个流程或多个流程和/或方框图一个方框或多个方框中指定的功能的装置。

这些计算机程序指令也可装载到计算机或其他可编程数据处理设备上，使得在计算机或其他可编程设备上执行一系列操作步骤以产生计算机实现的处理，从而在计算机或其他可编程设备上执行的指令提供用于实现在流程图一个流程或多个流程和/或方框图一个方框或多个方框中指定的功能的步骤。

显然，本领域的技术人员可以对本申请进行各种改动和变型而不脱离本申请的精神和范围。这样，倘若本申请的这些修改和变型属于本申请权利要求及其等同技术的范围之内，则本申请也意图包含这些改动和变型在内。

权 利 要 求 书

1.一种界面显示的方法，其特征在于，所述方法包括：

电子设备显示第一应用程序的第一窗口、第二应用程序的第二窗口和缩放控件，所述第一窗口和所述第二窗口中显示的界面元素采用第一像素密度；

所述电子设备检测到作用于所述缩放控件的第一输入操作，所述第一输入操作用于确定第二像素密度，所述第二像素密度不同于所述第一像素密度；

所述电子设备显示所述第一应用程序的第三窗口和所述第二应用程序的所述第二窗口，所述第三窗口中显示的界面元素采用所述第二像素密度，所述第二窗口中显示的界面元素采用所述第一像素密度。

2.如权利要求1所述的方法，其特征在于，还包括：

在显示所述第三窗口之前，所述电子设备向所述第一应用程序通知所述第二像素密度。

3.如权利要求1或2所述的方法，其特征在于，所述缩放控件包括放大控件和缩小控件。

4.如权利要求3所述的方法，其特征在于，如果所述第一输入操作为作用于所述放大控件的操作，则所述第二像素密度小于所述第一像素密度。

5.如权利要求3或4所述的方法，其特征在于，如果所述第一输入操作为作用于所述放大控件的操作，则所述第三窗口中显示的界面元素的尺寸大于所述第一窗口中显示的界面元素的尺寸。

6.如权利要求3所述的方法，其特征在于，如果所述第一输入操作为作用于所述缩小控件的操作，则所述第二像素密度大于所述第一像素密度。

7.如权利要求3或6所述的方法，其特征在于，如果所述第一输入操作为作用于所述缩小控件的操作，则所述第三窗口中显示的界面元素的尺寸小于所述第一窗口中显示的界面元素的尺寸。

8.如权利要求1-7任一项所述的方法，其特征在于，还包括：

在获取到所述第二像素密度之后，所述第一应用程序根据所述第二像素密度刷新应用配置并加载第二应用资源，所述第二应用资源为针对所述第二像素密度而配置的资源文件，第一应用资源为针对所述第一像素密度而配置的资源文件，所述第二应用资源不同于所述第一应用资源。

9.如权利要求1-8任一项所述的方法，其特征在于，所述第一应用程序所采用的像素密度dpi为以下之一：120dpi、160dpi、240dpi、320dpi、480dpi、640dpi。

10.如权利要求1-9任一项所述的方法，其特征在于，还包括：

如果所述第一应用程序包含多套应用资源，则所述电子设备响应所述第一输入操作。

11.如权利要求 1-10 任一项所述的方法，其特征在于，所述缩放控件为悬浮控件。

12.如权利要求 1-11 任一项所述的方法，其特征在于，所述缩放控件位于所述第一窗口内。

13.一种电子设备，其特征在于，所述电子设备包括：存储器以及耦合于所述存储器的处理器，所述存储器中存储有计算机可执行指令，所述处理器用于调用所述指令，使得所述电子设备执行以下步骤：

显示第一应用程序的第一窗口、第二应用程序的第二窗口和缩放控件，所述第一窗口和所述第二窗口中显示的界面元素采用第一像素密度；

检测到作用于所述缩放控件的第一输入操作，所述第一输入操作用于确定第二像素密度，所述第二像素密度不同于所述第一像素密度；

显示所述第一应用程序的第三窗口和所述第二应用程序的所述第二窗口，所述第三窗口中显示的界面元素采用所述第二像素密度，所述第二窗口中显示的界面元素采用所述第一像素密度。

14.如权利要求 13 所述的电子设备，其特征在于，所述处理器调用所述指令，使得所述电子设备还执行以下：

在显示所述第三窗口之前，向所述第一应用程序通知所述第二像素密度。

15.如权利要求 13 或 14 所述的电子设备，其特征在于，所述缩放控件包括放大控件和缩小控件。

16.如权利要求 15 所述的电子设备，其特征在于，如果所述第一输入操作为作用于所述放大控件的操作，则所述第二像素密度小于所述第一像素密度。

17.如权利要求 15 或 16 所述的电子设备，其特征在于，如果所述第一输入操作为作用于所述放大控件的操作，则所述第三窗口中显示的界面元素的尺寸大于所述第一窗口中显示的界面元素的尺寸。

18.如权利要求 15 所述的电子设备，其特征在于，如果所述第一输入操作为作用于所述缩小控件的操作，则所述第二像素密度大于所述第一像素密度。

19.如权利要求 15 或 18 所述的电子设备，其特征在于，如果所述第一输入操作为作用于所述缩小控件的操作，则所述第三窗口中显示的界面元素的尺寸小于所述第一窗口中显示的界面元素的尺寸。

20.如权利要求 13-19 任一项所述的电子设备，其特征在于，所述处理器调用所述指令，使得所述电子设备还执行以下：

在获取到所述第二像素密度之后，所述第一应用程序根据所述第二像素密度刷新应用配置并加载第二应用资源，所述第二应用资源为针对所述第二像素密度而配置的资源文件，第一应用资源为针对所述第一像素密度而配置的资源文件，所述第二应用资源不同于所述第一

应用资源。

21.如权利要求 13-20 任一项所述的电子设备，其特征在于，所述第一应用程序所采用的像素密度 dpi 为以下之一：120dpi、160dpi、240dpi、320dpi、480dpi、640dpi。

22.如权利要求 13-21 任一项所述的电子设备，其特征在于，所述处理器调用所述指令，使得所述电子设备还执行以下：

如果所述第一应用程序包含多套应用资源，则响应所述第一输入操作。

23.如权利要求 13-22 任一项所述的电子设备，其特征在于，所述缩放控件为悬浮控件。

24.如权利要求 13-23 任一项所述的电子设备，其特征在于，所述缩放控件位于所述第一窗口内。

25.一种计算机存储介质，其特征在于，所述存储介质中存储有计算机程序，所述计算机程序包括可执行指令，所述可执行指令当被处理器执行时使所述处理器执行如权利要求 1-12 中任一项所述的方法对应的操作。

电子设备100

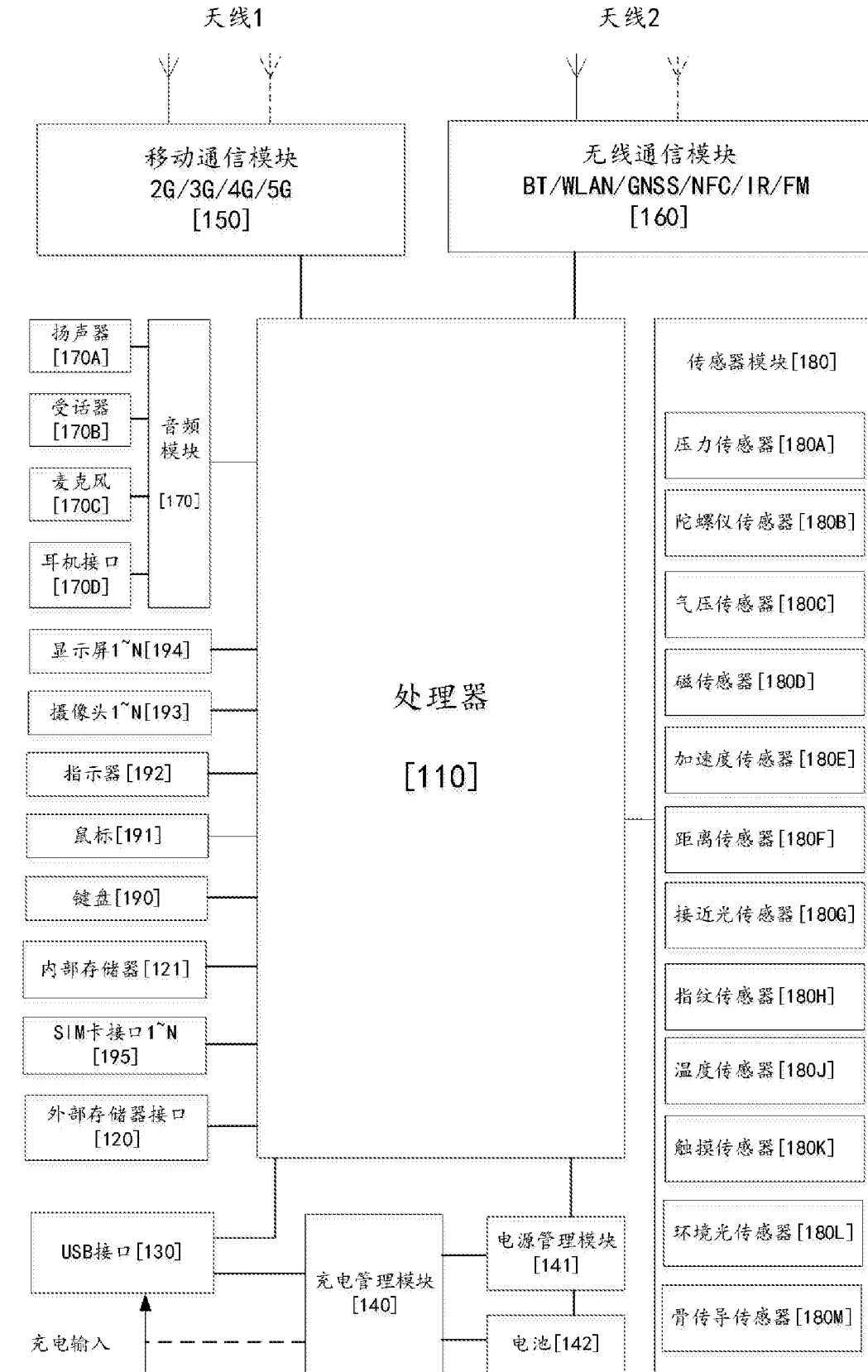
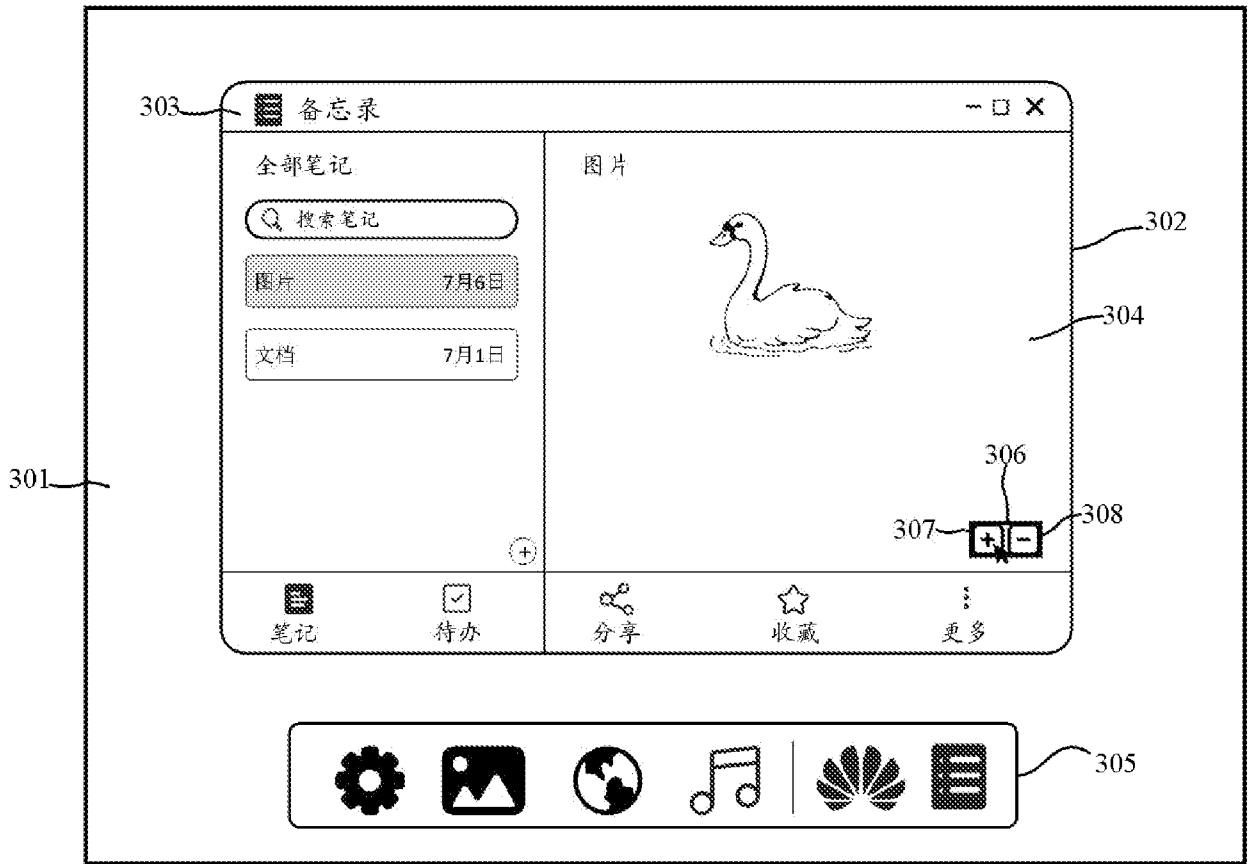


图 1



图 2



↓ 放大显示

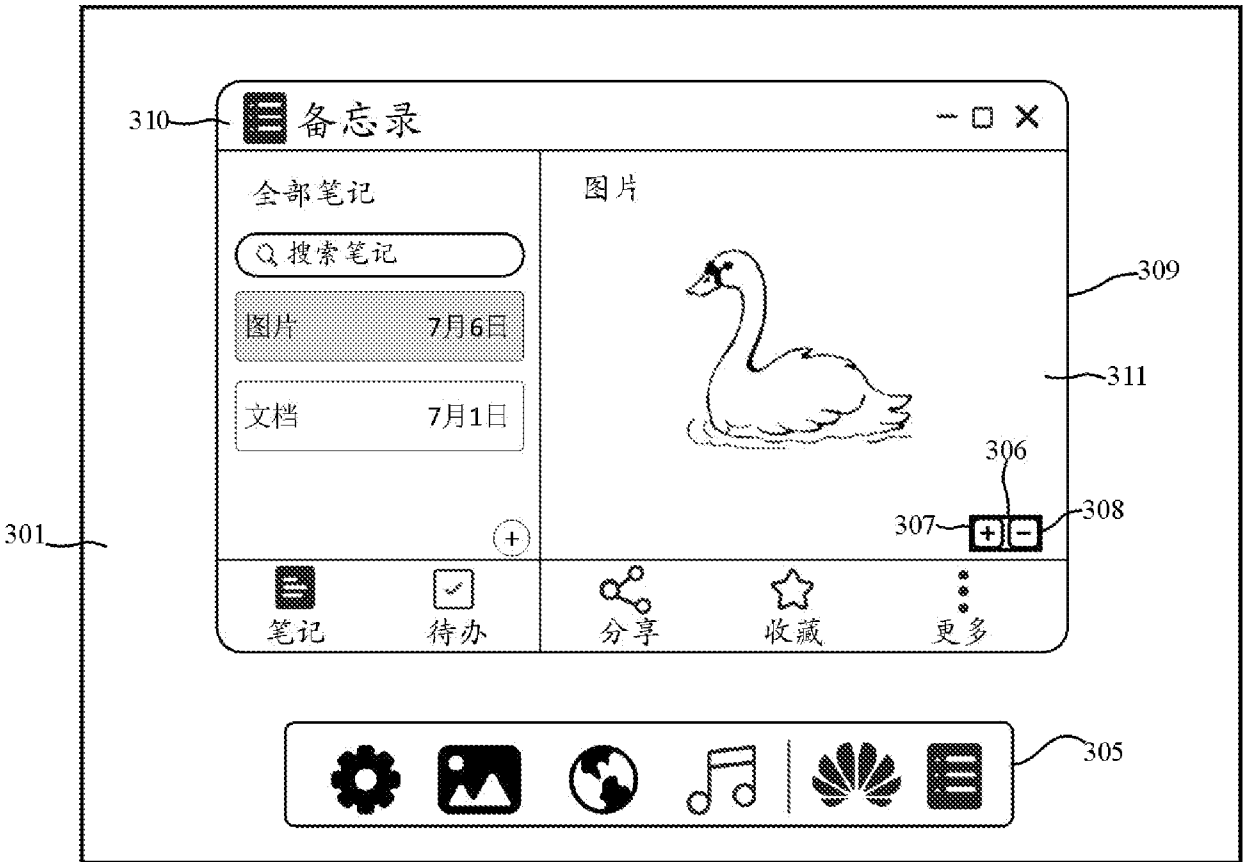
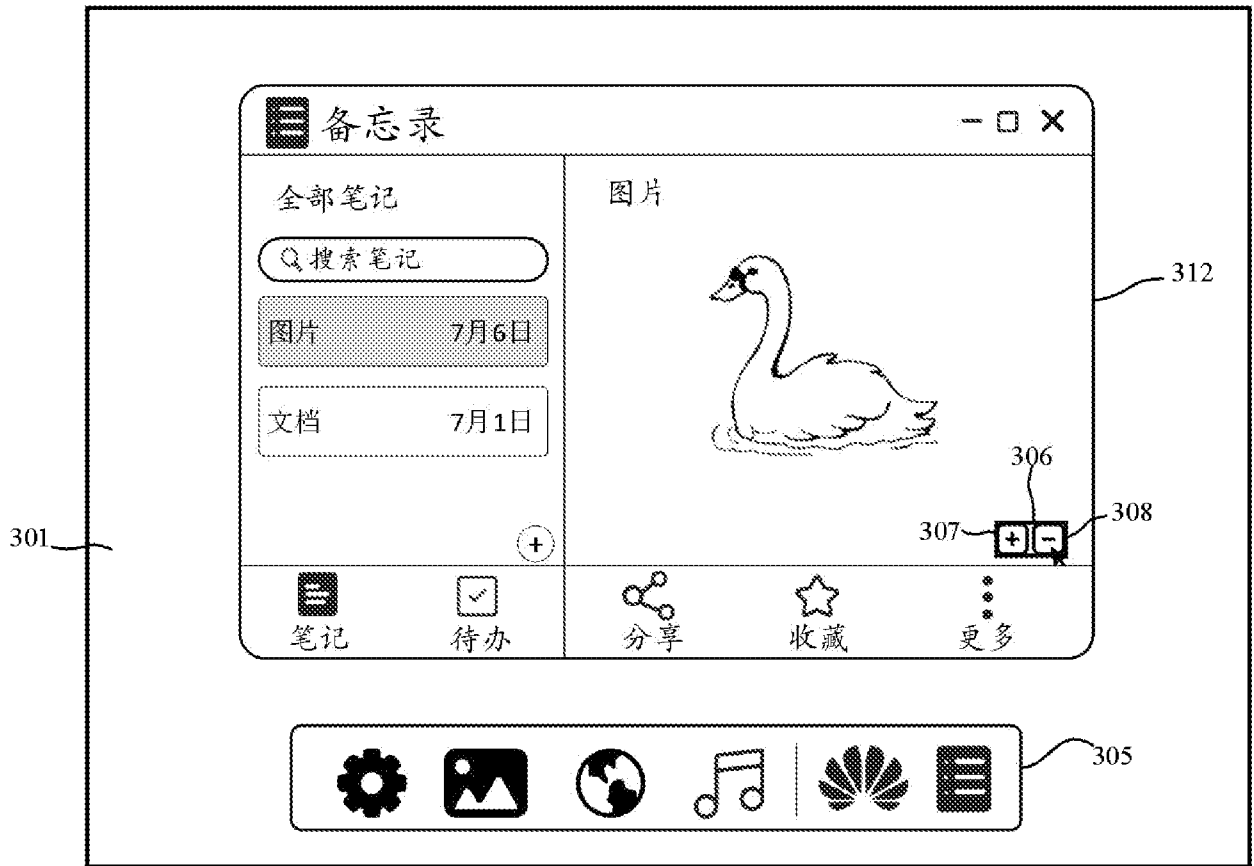


图 3a



↓ 缩小显示

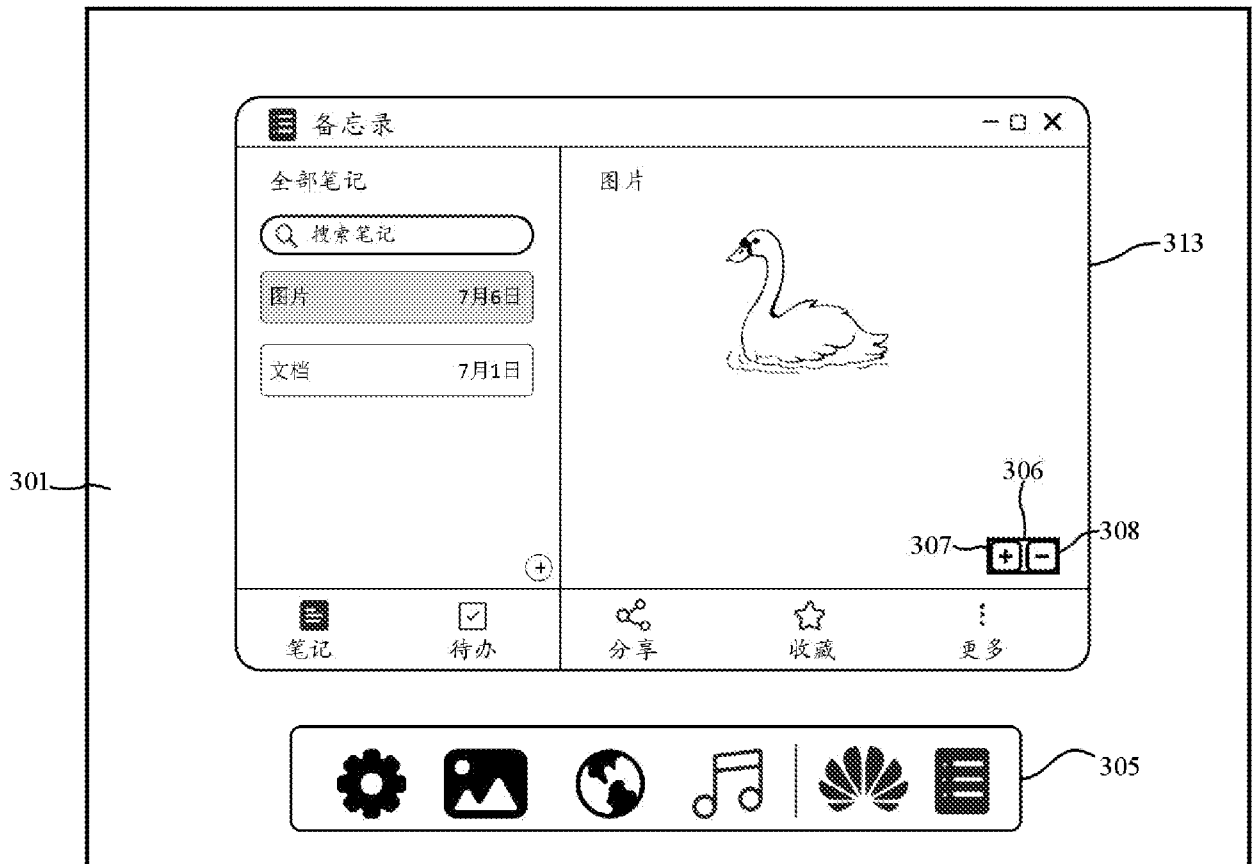


图 3b

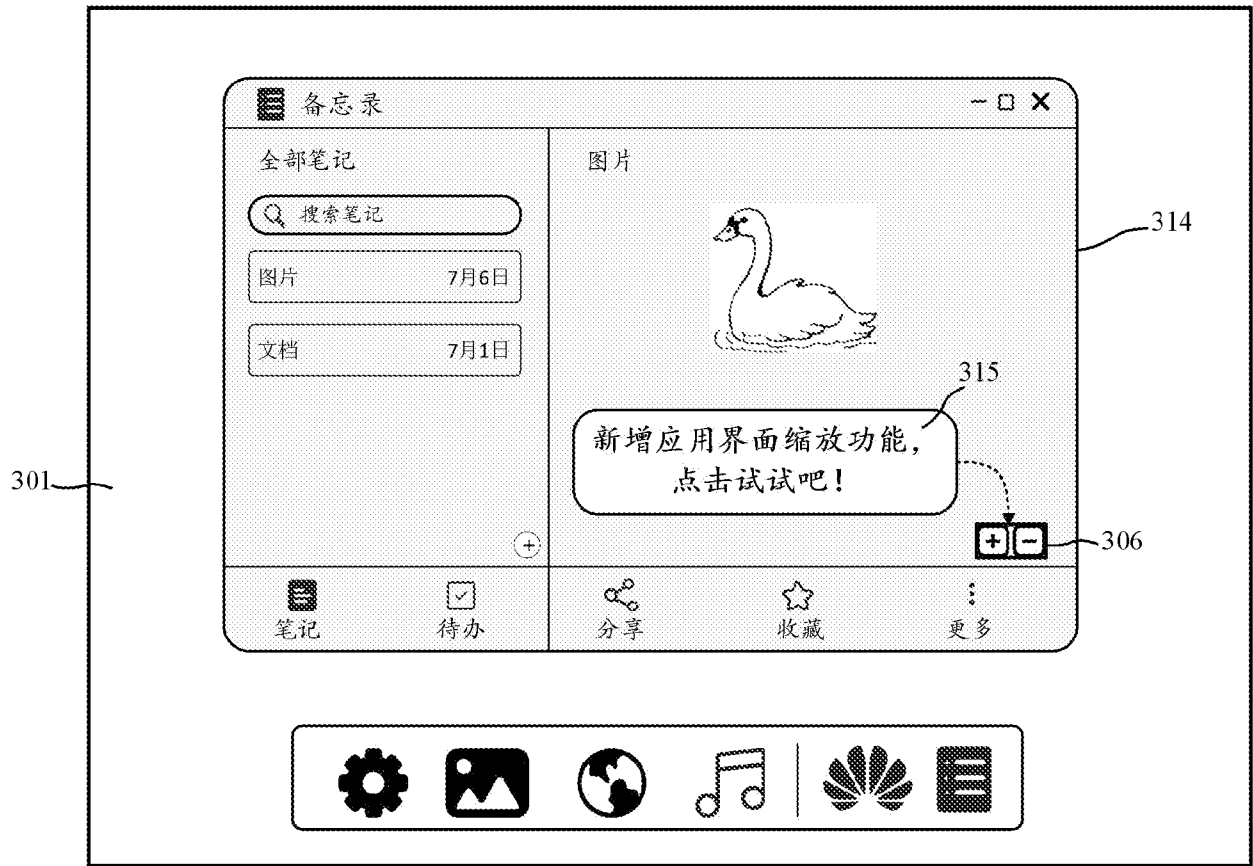


图 3c

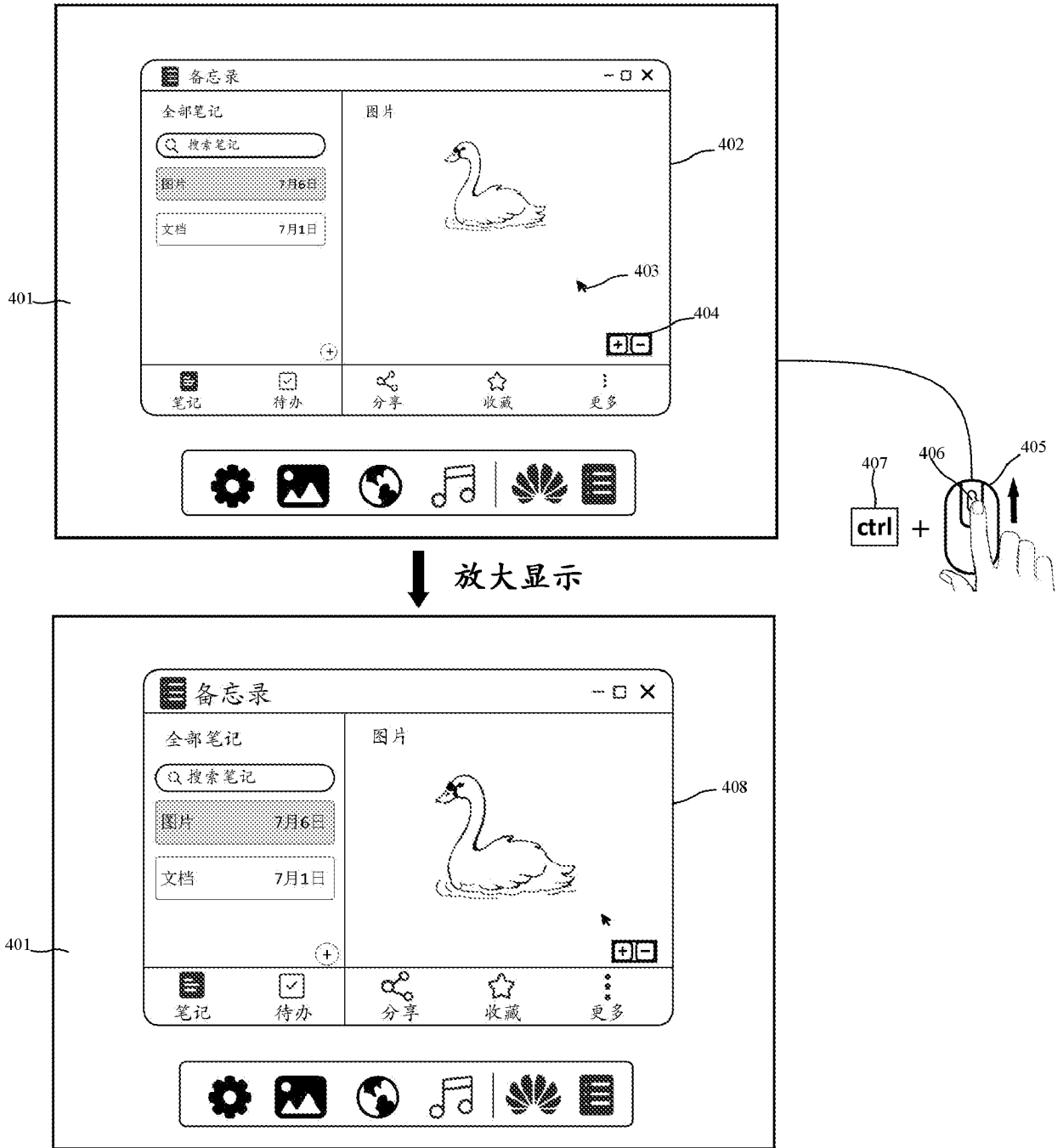


图 4a

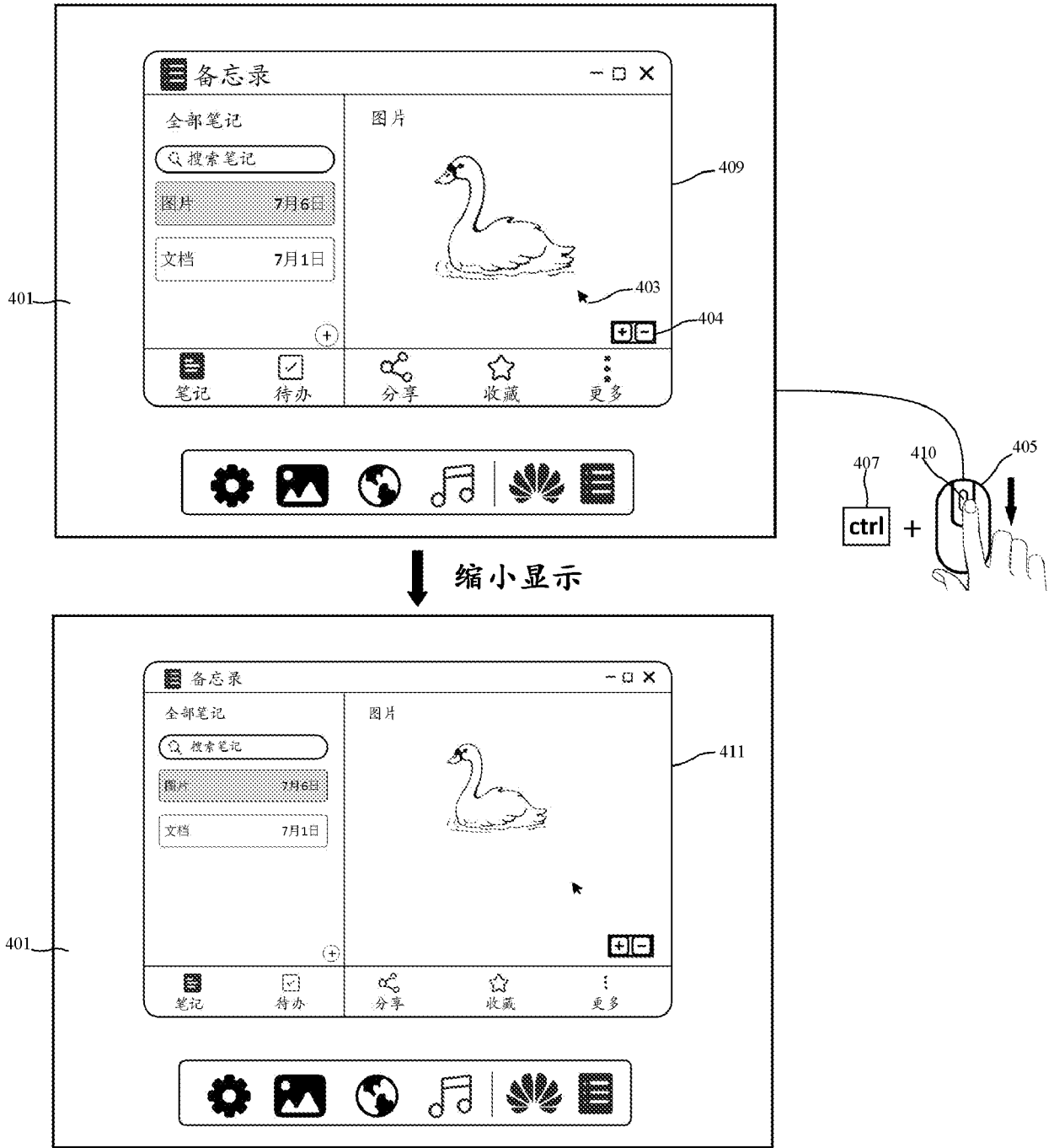
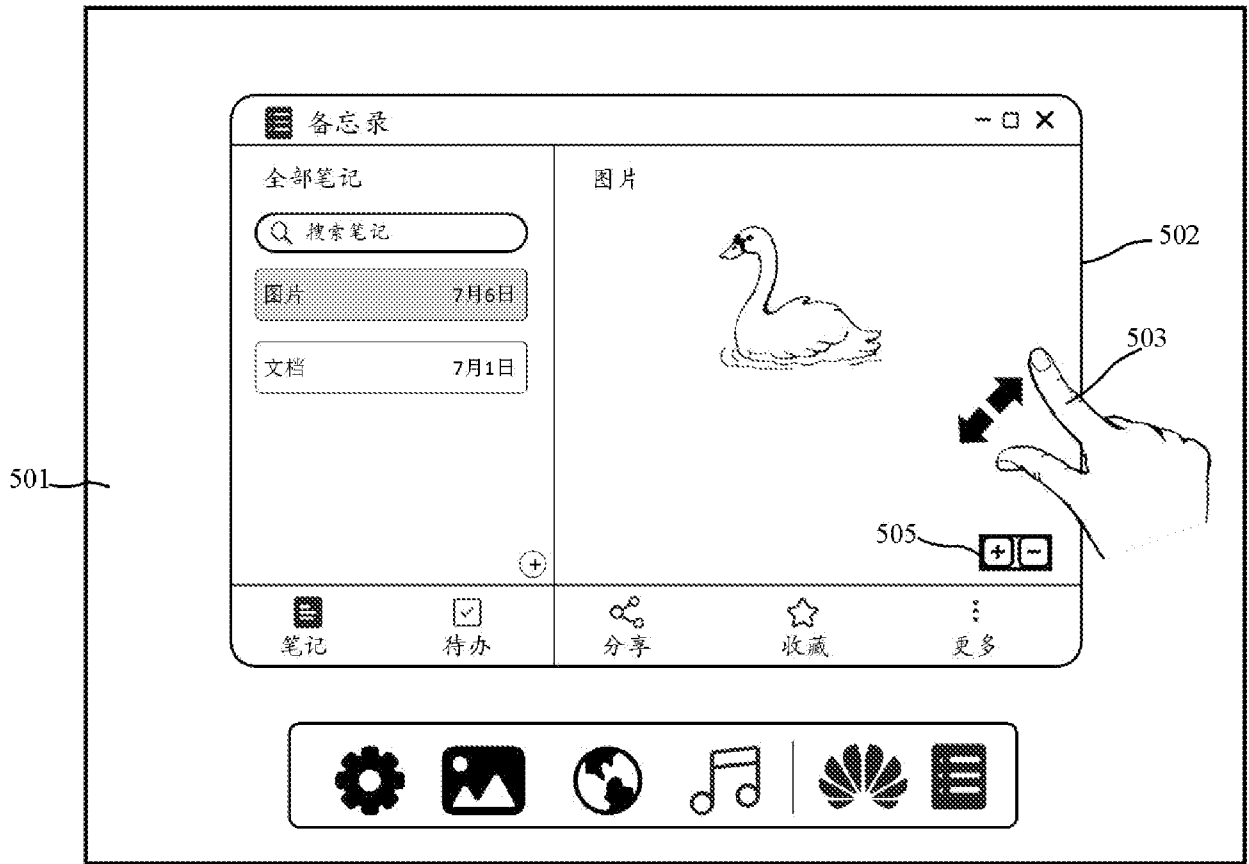


图 4b



图 4c



↓ 放大显示

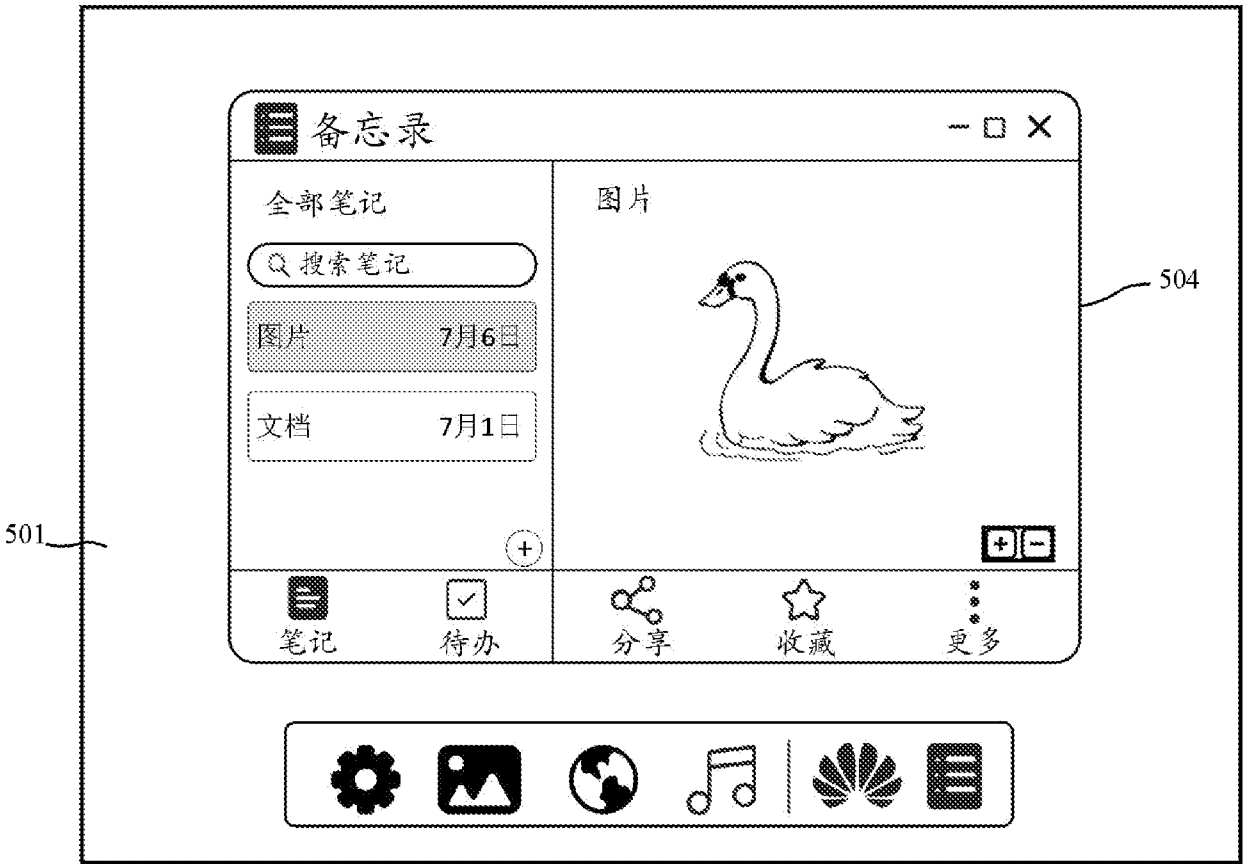
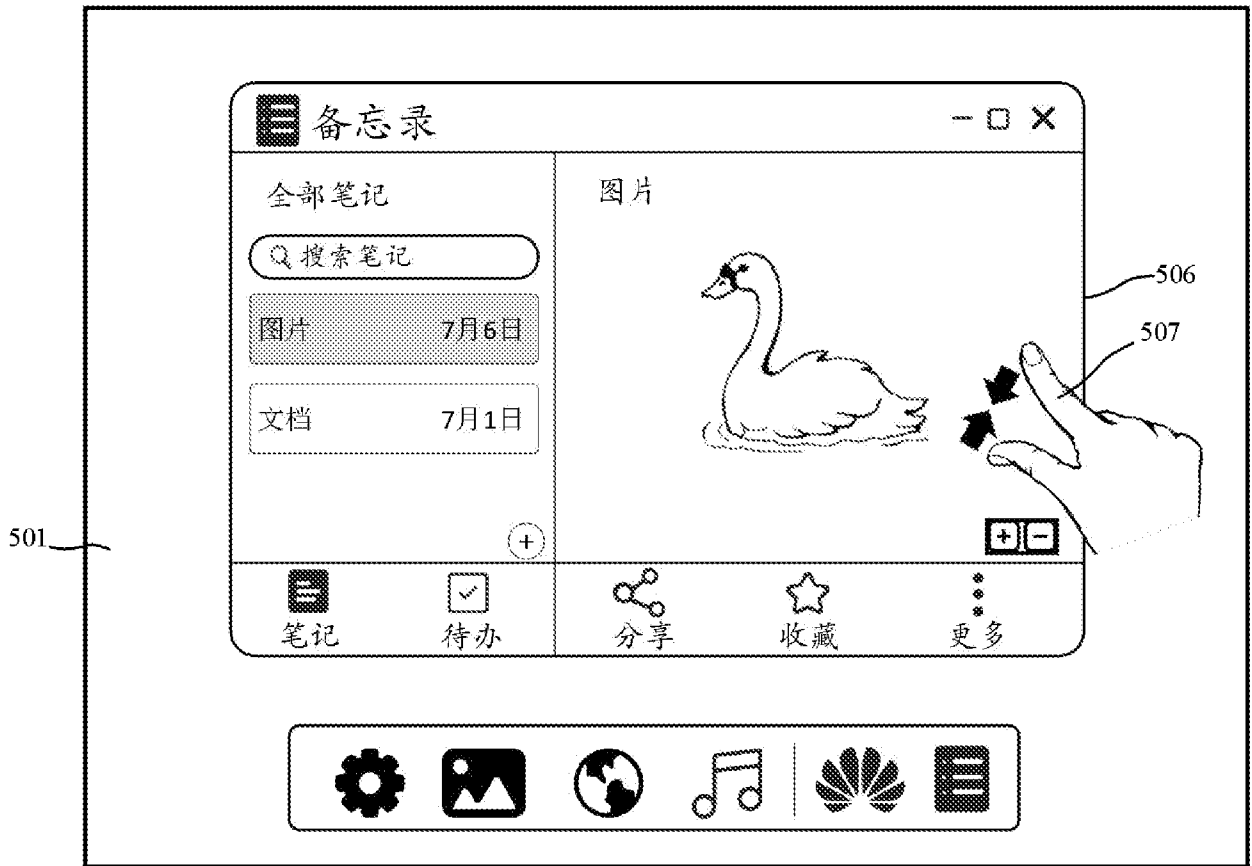


图 5a



↓ 缩小显示

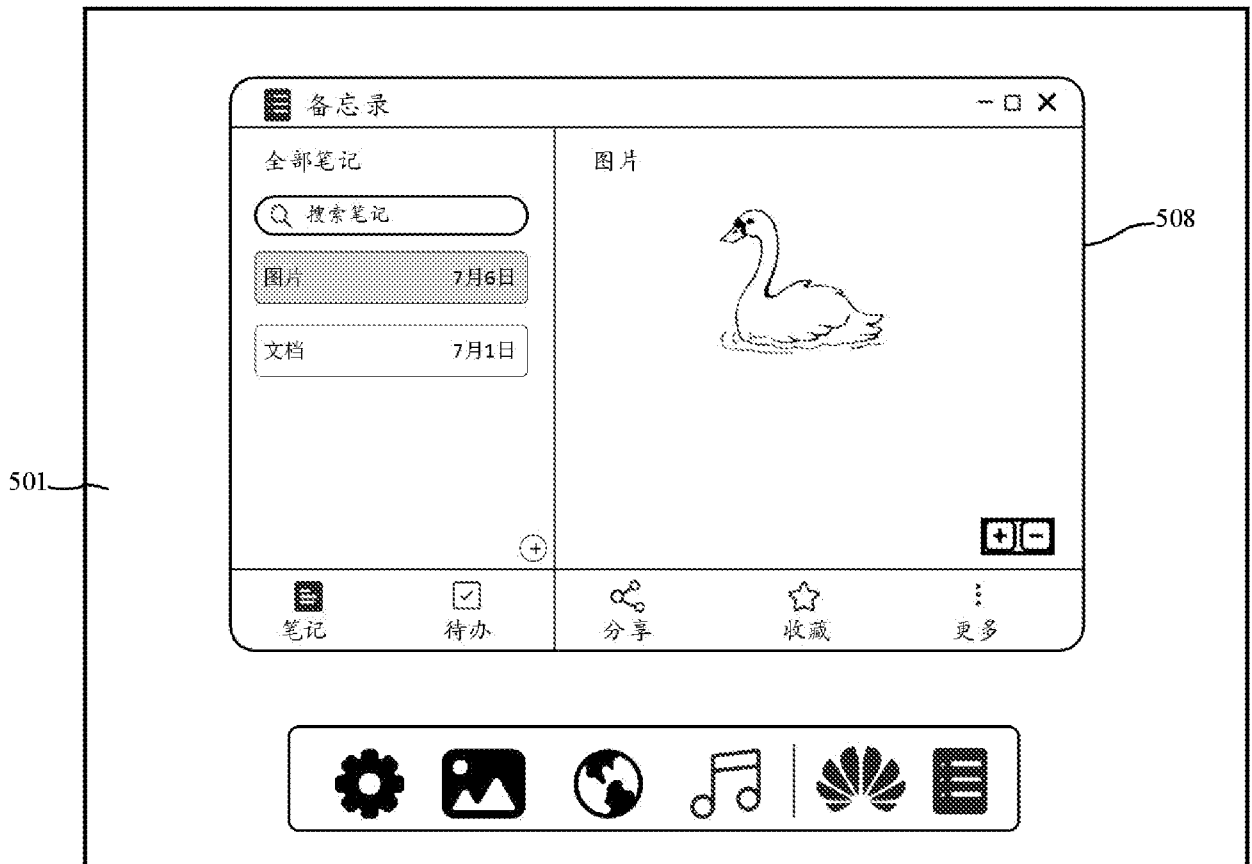


图 5b

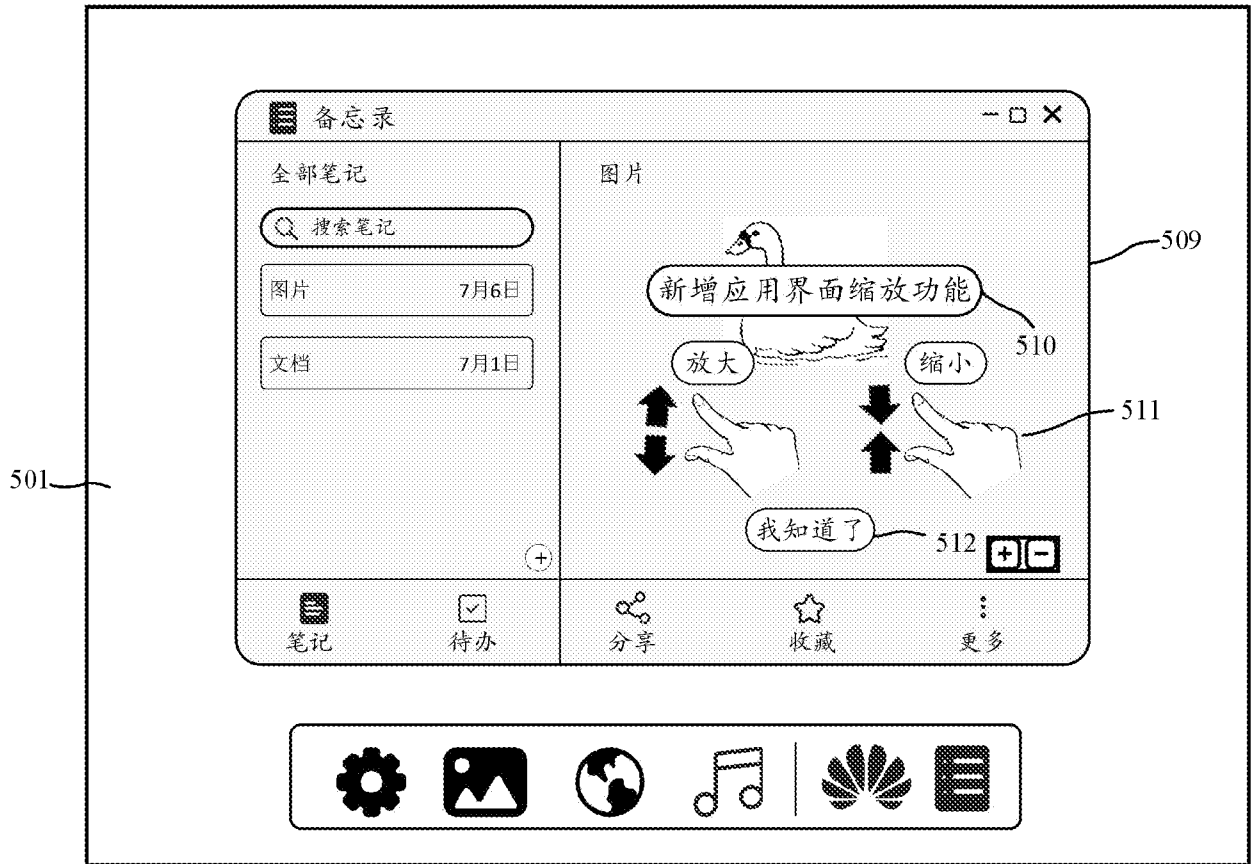
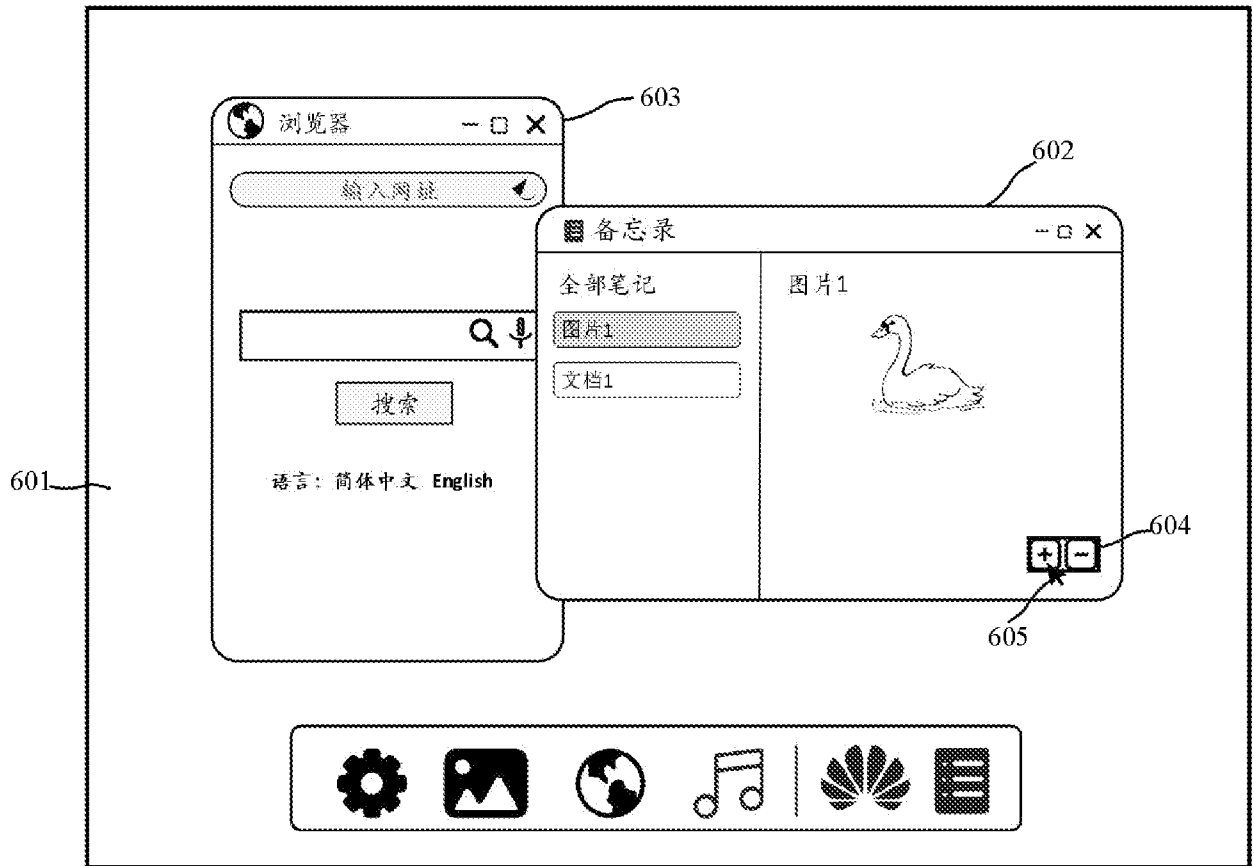


图 5c



放大显示

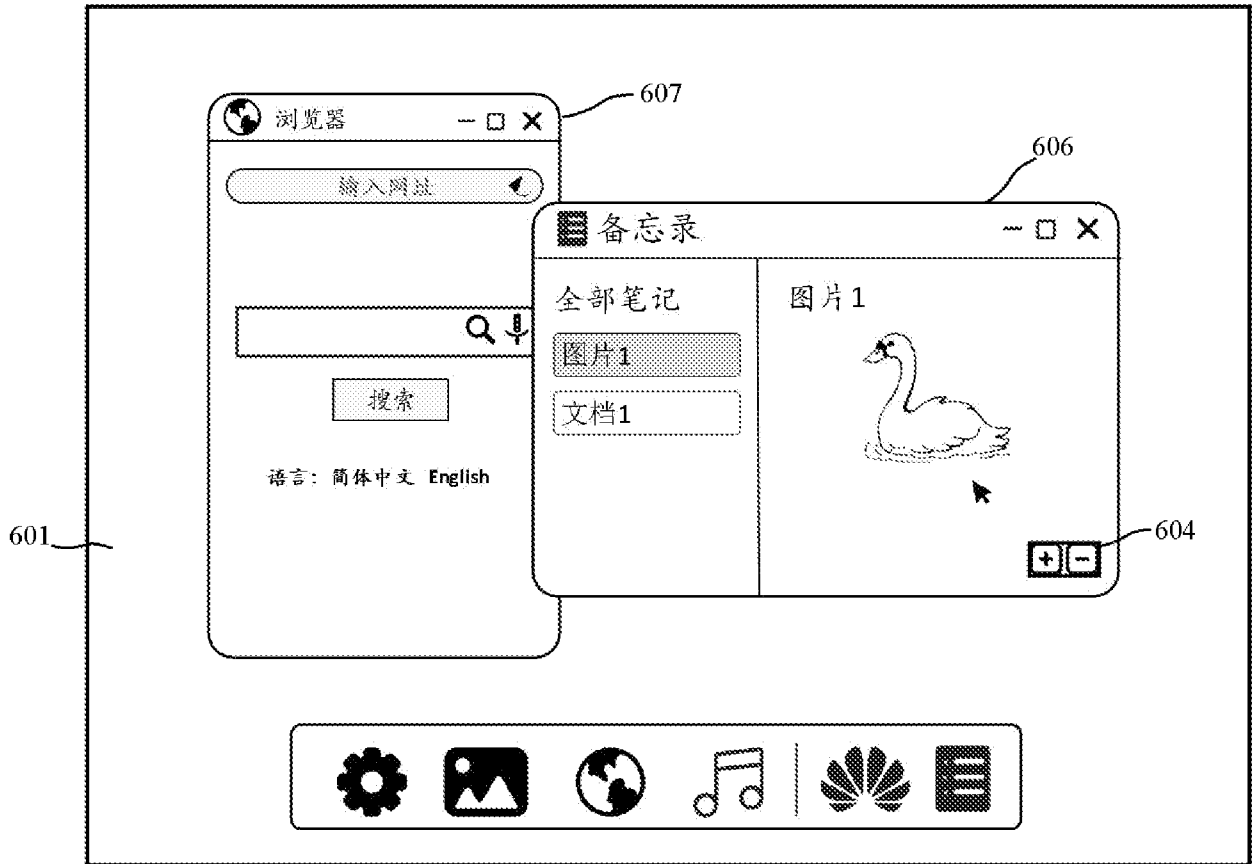


图 6a

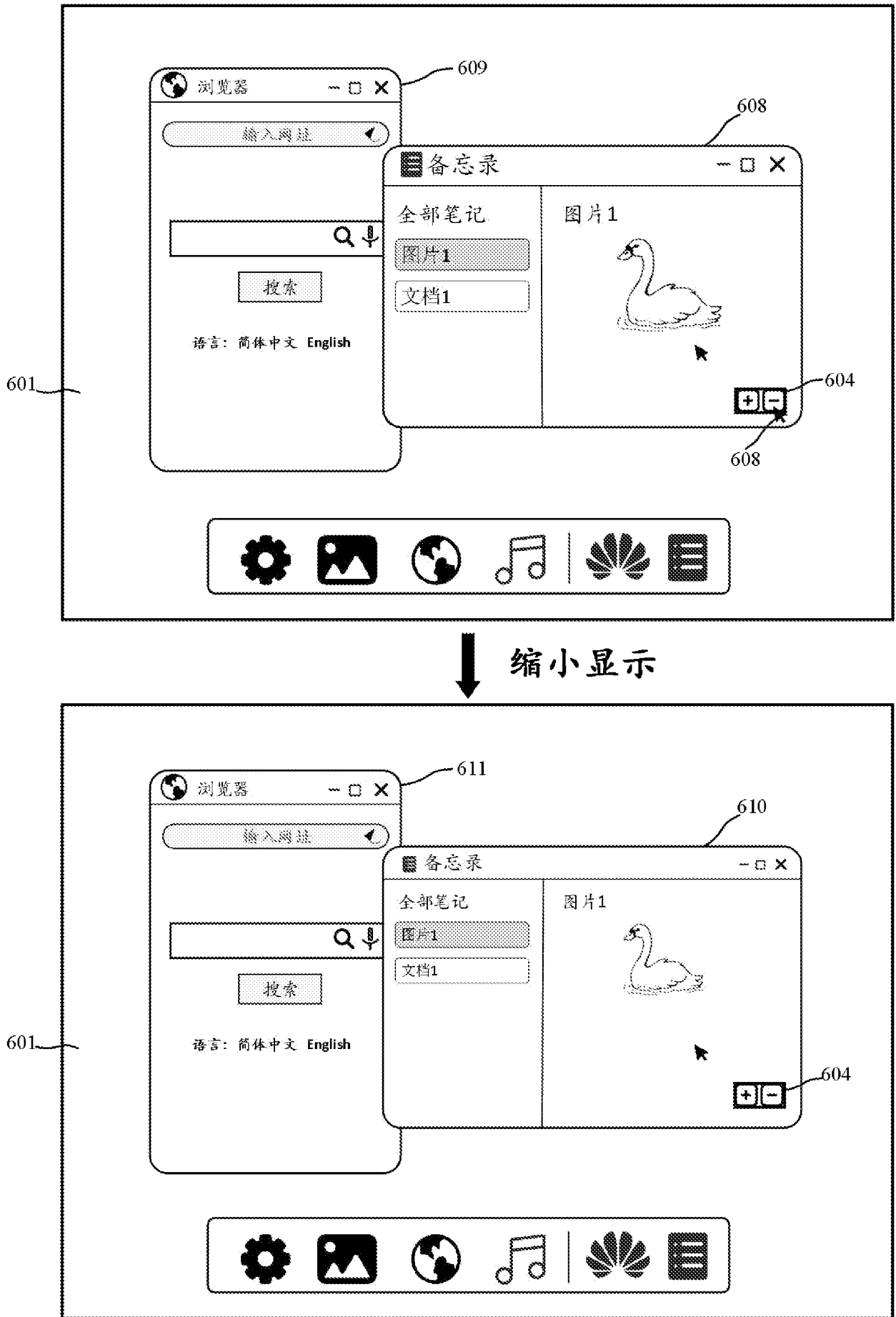


图 6b

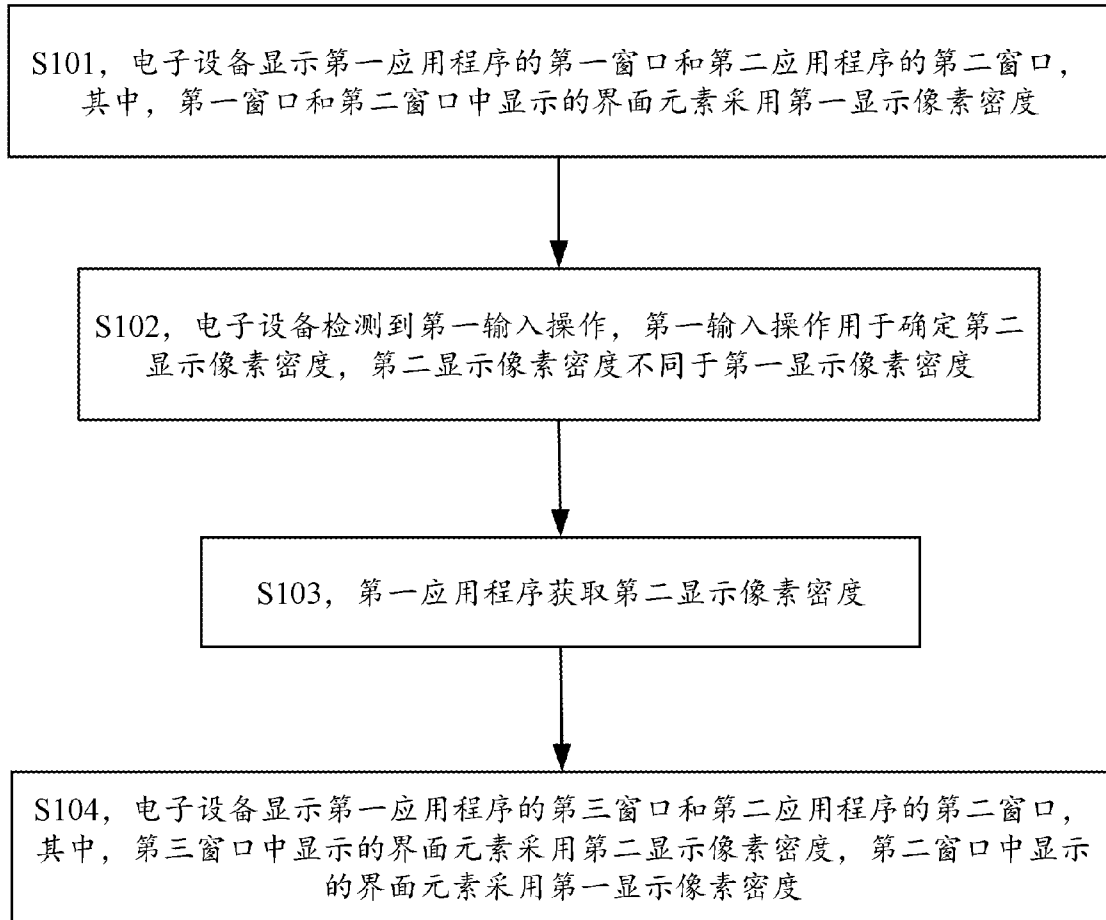


图 7

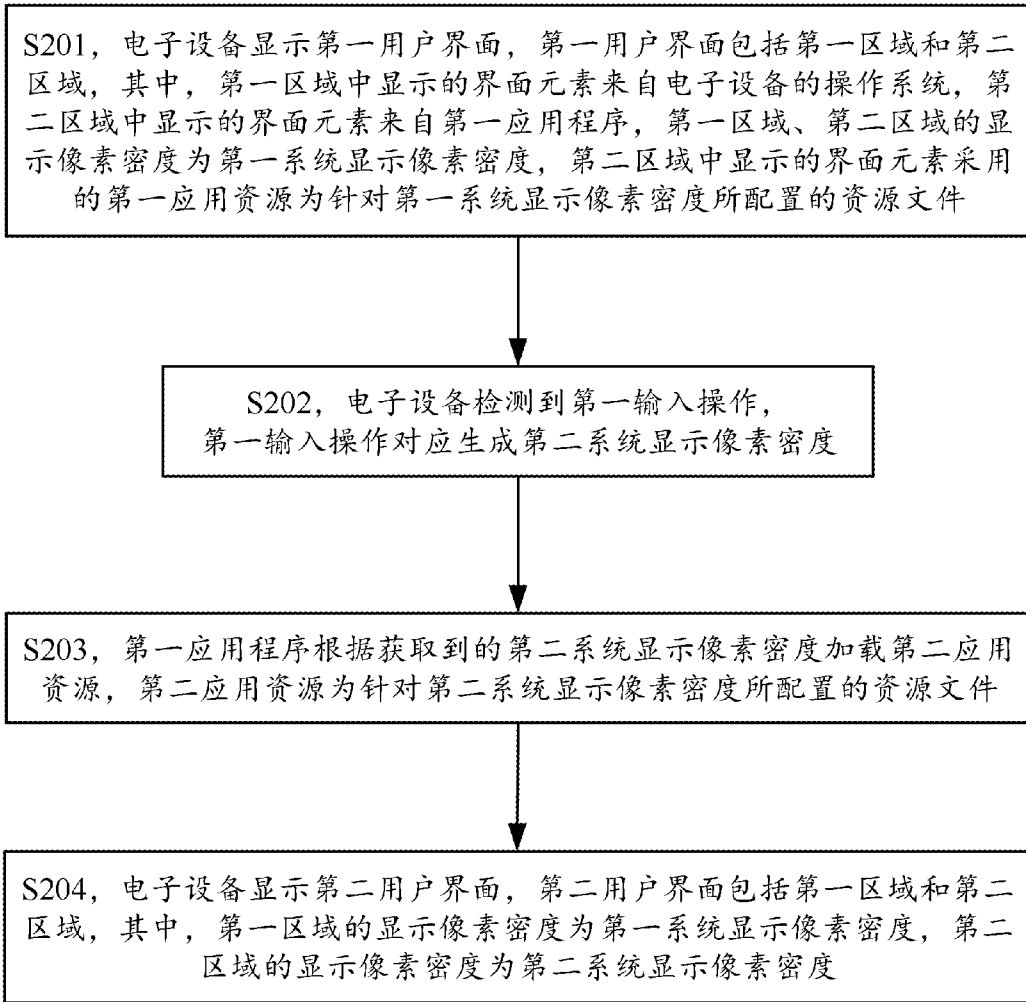


图 8

INTERNATIONAL SEARCH REPORT

International application No.

PCT/CN2021/111624

A. CLASSIFICATION OF SUBJECT MATTER

G06F 9/44(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)

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)

CNPAT, WPI, EPODOC, CNKI, IEEE: 界面, 窗口, 应用, app, 程序, 多, 同时, 前台, 缩放, 像素, interface, window, multi, scale, zoom, pixel

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	CN 104572058 A (LENOVO (BEIJING) LIMITED) 29 April 2015 (2015-04-29) description, paragraphs [0221]-[0263], paragraphs [0350]-[0353]	1-25
X	CN 105892823 A (YULONG COMPUTER TELECOMMUNICATION SCIENTIFIC (SHENZHEN) CO., LTD.) 24 August 2016 (2016-08-24) description, paragraphs [0033]-[0052]	1-25
A	CN 103870194 A (LENOVO (BEIJING) LIMITED) 18 June 2014 (2014-06-18) entire document	1-25
A	US 2010281400 A1 (QUALCOMM INCORPORATED) 04 November 2010 (2010-11-04) entire document	1-25

 Further documents are listed in the continuation of Box C. 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

"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

27 October 2021

Date of mailing of the international search report

15 November 2021

Name and mailing address of the ISA/CN

China National Intellectual Property Administration (ISA/
CN)
No. 6, Xitucheng Road, Jimenqiao, Haidian District, Beijing
100088
China

Authorized officer

Facsimile No. (86-10)62019451

Telephone No.

INTERNATIONAL SEARCH REPORT
Information on patent family members

International application No.

PCT/CN2021/111624

Patent document cited in search report			Publication date (day/month/year)	Patent family member(s)			Publication date (day/month/year)
CN	104572058	A	29 April 2015	None			
CN	105892823	A	24 August 2016	None			
CN	103870194	A	18 June 2014	None			
US	2010281400	A1	04 November 2010	JP	2012525652	A	22 October 2012
				BR	PI1011968	A2	09 April 2019
				CN	102414682	A	11 April 2012
				KR	20120024700	A	14 March 2012
				EP	2425359	A1	07 March 2012
				WO	2010129394	A1	11 November 2010

国际检索报告

国际申请号

PCT/CN2021/111624

<p>A. 主题的分类</p> <p>G06F 9/44(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>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>CNPAT, WPI, EPODOC, CNKI, IEEE: 界面, 窗口, 应用, app, 程序, 多, 同时, 前台, 缩放, 像素, interface, window, multi, scale, zoom, pixel</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 104572058 A (联想北京有限公司) 2015年 4月 29日 (2015 - 04 - 29) 说明书第[0221]-[0263]段、第[0350]-[0353]段</td> <td>1-25</td> </tr> <tr> <td>X</td> <td>CN 105892823 A (字龙计算机通信科技深圳有限公司) 2016年 8月 24日 (2016 - 08 - 24) 说明书第[0033]-[0052]段</td> <td>1-25</td> </tr> <tr> <td>A</td> <td>CN 103870194 A (联想北京有限公司) 2014年 6月 18日 (2014 - 06 - 18) 全文</td> <td>1-25</td> </tr> <tr> <td>A</td> <td>US 2010281400 A1 (QUALCOMM INC.) 2010年 11月 4日 (2010 - 11 - 04) 全文</td> <td>1-25</td> </tr> </tbody> </table>			类型*	引用文件, 必要时, 指明相关段落	相关的权利要求	X	CN 104572058 A (联想北京有限公司) 2015年 4月 29日 (2015 - 04 - 29) 说明书第[0221]-[0263]段、第[0350]-[0353]段	1-25	X	CN 105892823 A (字龙计算机通信科技深圳有限公司) 2016年 8月 24日 (2016 - 08 - 24) 说明书第[0033]-[0052]段	1-25	A	CN 103870194 A (联想北京有限公司) 2014年 6月 18日 (2014 - 06 - 18) 全文	1-25	A	US 2010281400 A1 (QUALCOMM INC.) 2010年 11月 4日 (2010 - 11 - 04) 全文	1-25
类型*	引用文件, 必要时, 指明相关段落	相关的权利要求															
X	CN 104572058 A (联想北京有限公司) 2015年 4月 29日 (2015 - 04 - 29) 说明书第[0221]-[0263]段、第[0350]-[0353]段	1-25															
X	CN 105892823 A (字龙计算机通信科技深圳有限公司) 2016年 8月 24日 (2016 - 08 - 24) 说明书第[0033]-[0052]段	1-25															
A	CN 103870194 A (联想北京有限公司) 2014年 6月 18日 (2014 - 06 - 18) 全文	1-25															
A	US 2010281400 A1 (QUALCOMM INC.) 2010年 11月 4日 (2010 - 11 - 04) 全文	1-25															
<p><input type="checkbox"/> 其余文件在C栏的续页中列出。</p> <p><input checked="" type="checkbox"/> 见同族专利附件。</p>																	
<p>* 引用文件的具体类型:</p> <p>“A” 认为不特别相关的表示了现有技术一般状态的文件</p> <p>“E” 在国际申请日的当天或之后公布的在先申请或专利</p> <p>“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)</p> <p>“O” 涉及口头公开、使用、展览或其他方式公开的文件</p> <p>“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</p> <p>“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</p> <p>“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</p> <p>“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</p> <p>“&” 同族专利的文件</p>																	
<p>国际检索实际完成的日期</p> <p>2021年 10月 27日</p>	<p>国际检索报告邮寄日期</p> <p>2021年 11月 15日</p>																
<p>ISA/CN的名称和邮寄地址</p> <p>中国国家知识产权局(ISA/CN) 中国 北京市海淀区蓟门桥西土城路6号 100088</p> <p>传真号 (86-10)62019451</p>	<p>授权官员</p> <p>李文浩</p> <p>电话号码 86-(10)-53961341</p>																

国际检索报告
关于同族专利的信息

国际申请号

PCT/CN2021/111624

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	104572058	A	2015年 4月 29日	无			
CN	105892823	A	2016年 8月 24日	无			
CN	103870194	A	2014年 6月 18日	无			
US	2010281400	A1	2010年 11月 4日	JP	2012525652	A	2012年 10月 22日
				BR	PI1011968	A2	2019年 4月 9日
				CN	102414682	A	2012年 4月 11日
				KR	20120024700	A	2012年 3月 14日
				EP	2425359	A1	2012年 3月 7日
				WO	2010129394	A1	2010年 11月 11日