



(12) 发明专利申请

(10) 申请公布号 CN 103473011 A

(43) 申请公布日 2013. 12. 25

(21) 申请号 201310395728. 4

(22) 申请日 2013. 09. 03

(71) 申请人 小米科技有限责任公司
地址 100085 北京市海淀区清河中街 68 号
华润五彩城购物中心二期 13 层

(72) 发明人 梁金祥 孙鹏 闫昊

(51) Int. Cl.
G06F 3/0488 (2013. 01)

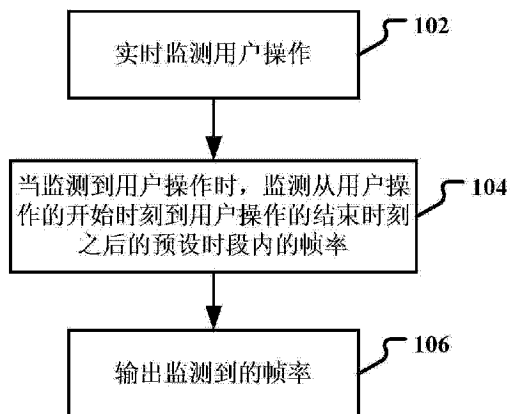
权利要求书2页 说明书9页 附图4页

(54) 发明名称

一种移动终端性能检测方法、装置及移动终端

(57) 摘要

本发明公开了一种移动终端性能检测方法、装置及移动终端。所述方法包括：实时监测用户操作；当监测到用户操作时，监测从所述用户操作的开始时刻到所述用户操作的结束时刻之后的预设时段内的帧率；输出监测到的帧率。本发明可以直观准确地获得操作系统或应用程序的性能及移动终端对用户操作的响应速度进而可以分析出移动终端与用户交互过程的流畅性，使得对移动终端的性能测试更加精确。



1. 一种移动终端性能检测方法,其特征在于,包括:
实时监测用户操作;
当监测到用户操作时,监测从所述用户操作的开始时刻到所述用户操作的结束时刻之后的预设时段内的帧率;
输出监测到的帧率。
2. 根据权利要求1所述的方法,其特征在于,所述监测帧率,采用如下方式:
记录每帧对应的实际绘画开始时间和实际绘画结束时间、绘画同步信号接收时间和渲染开始时间;
根据当前帧的实际绘画结束时间查找当前帧对应的绘画同步信号接收时间,并根据查找到的所述绘画同步信号接收时间查找当前帧对应的渲染开始时间;
计算所述当前帧对应的渲染开始时间与前一帧的渲染开始时间的时间间隔;
根据所述当前帧对应的渲染开始时间与前一帧的渲染开始时间的的时间间隔,计算当前帧对应的帧率。
3. 根据权利要求2所述的方法,其特征在于,监测帧率,还包括:
对于第一个帧,计算所述第一个帧的渲染开始时间与所述第一个帧的实际绘画开始时间之间的时间间隔;
根据所述第一个帧的渲染开始时间与所述第一个帧的实际绘画开始时间之间的时间间隔,计算帧率。
4. 根据权利要求2所述的方法,其特征在于,所述方法还包括:
输出每帧的绘画同步信号接收时间。
5. 根据权利要求1所述的方法,其特征在于,输出监测到的帧率包括:
生成所述帧率和对应帧数的坐标图;
显示所述坐标图。
6. 根据权利要求1所述的方法,其特征在于,所述用户操作包括:用户在移动终端上的触摸操作和/或按键操作。
7. 一种移动终端性能检测装置,其特征在于,包括:
用户操作监测模块,用于实时监测用户操作;
帧率监测模块,用于当监测到用户操作时,监测从所述用户操作的开始时刻到所述用户操作的结束时刻之后的预设时段内的帧率;
输出模块,用于输出监测到的帧率。
8. 根据权利要求7所述的装置,其特征在于,所述帧率监测模块包括:
记录子模块,用于记录每帧对应的实际绘画开始时间和实际绘画结束时间、绘画同步信号接收时间和渲染开始时间;
查找子模块,用于根据当前帧的实际绘画结束时间查找当前帧对应的绘画同步信号接收时间,并根据查找到的所述绘画同步信号接收时间查找当前帧对应的渲染开始时间;
第一计算子模块,用于计算所述当前帧对应的渲染开始时间与前一帧的渲染开始时间的的时间间隔;
第二计算子模块,用于根据所述当前帧对应的渲染开始时间与前一帧的渲染开始时间的的时间间隔计算当前帧对应的帧率。

9. 根据权利要求 8 所述的装置,其特征在于,
所述第一计算子模块,用于对于第一个帧,计算所述第一个帧的渲染开始时间与所述第一个帧的实际绘画开始时间之间的时间间隔;
所述第二计算子模块,用于根据所述第一个帧的渲染开始时间与所述第一个帧的实际绘画开始时间之间的时间间隔计算帧率。
10. 根据权利要求 8 所述的装置,其特征在于,所述输出模块,还用于输出每帧的绘画同步信号接收时间。
11. 根据权利要求 8 所述的装置,其特征在于,所述输出模块,还用于生成所述帧率和对应帧数的坐标图,显示所述坐标图。
12. 一种移动终端,其特征在于,包括有存储器,以及一个或者一个以上的程序,其中一个或者一个以上程序存储于存储器中,且经配置以由一个或者一个以上处理器执行所述一个或者一个以上程序包含用于进行以下操作的指令:
实时监测用户操作;
当监测到用户操作时,监测从所述用户操作的开始时刻到所述用户操作的结束时刻之后的预设时段内的帧率;
输出监测到的帧率。

一种移动终端性能检测方法、装置及移动终端

技术领域

[0001] 本发明涉及移动终端技术领域,尤其涉及一种移动终端性能检测方法、装置及移动终端。

背景技术

[0002] 在目前情况下,移动终端技术发展的越来越快,其携带便捷,具有人性化的操作方式,并能够通过安装应用程序给用户多种便捷的服务。

[0003] 同时,移动终端的操作系统以及应用程序性能也越来越受到开发商和用户的重视。例如:某一个应用程序安装后,运行缓慢,点击一下需要很久才有响应,显然影响用户体验。

[0004] 但是,随着移动终端的硬件配置的增强,操作系统或应用程序性能差异已经越来越小,很难通过量化的数据去对比、说明一个操作系统或应用程序的性能,以及对用户操作的响应速度。

发明内容

[0005] 本发明实施例提供一种移动终端性能检测方法、装置及移动终端,用于实现直观准确地获得操作系统或应用程序的性能及移动终端对用户操作的响应速度。

[0006] 一种移动终端性能检测方法,包括:

[0007] 实时监测用户操作;

[0008] 当监测到用户操作时,监测从所述用户操作的开始时刻到所述用户操作的结束时刻之后的预设时段内的帧率;

[0009] 输出监测到的帧率。

[0010] 本实施例中,通过监测用户操作过程中以及用户操作结束后一段时间内的移动终端的帧率,即用户手指按下、抬起这一动作过程中移动终端的显示帧率,可以直观准确地获得操作系统或应用程序的性能及移动终端对用户操作的响应速度进而可以分析出移动终端与用户交互过程的流畅性,使得对移动终端的性能测试更加精确。

[0011] 优选地,监测帧率,采用如下方式:

[0012] 记录每帧对应的实际绘画开始时间和实际绘画结束时间、绘画同步信号接收时间和渲染开始时间;

[0013] 根据当前帧的实际绘画结束时间查找当前帧对应的绘画同步信号接收时间,并根据查找到的所述绘画同步信号接收时间查找当前帧对应的渲染开始时间;

[0014] 计算所述当前帧对应的渲染开始时间与前一帧的渲染开始时间的时间间隔;

[0015] 根据所述当前帧对应的渲染开始时间与前一帧的渲染开始时间的的时间间隔,计算当前帧对应的帧率。

[0016] 优选地,监测帧率还采用如下方式:

[0017] 对于第一个帧,计算所述第一个帧的渲染开始时间与所述第一个帧的实际绘画开

始时间之间的时间间隔；

[0018] 根据所述第一个帧的渲染开始时间与所述第一个帧的实际绘画开始时间之间的时间间隔，计算帧率。

[0019] 本实施例中，通过对用户操作过程中移动终端的帧率进行实时监测，可以直观准确地获得操作系统或应用程序的性能及移动终端对用户操作的响应速度进而可以分析出移动终端与用户交互过程的流畅性，使得对移动终端的性能测试更加精确。

[0020] 优选地，所述方法还包括：

[0021] 输出每帧的绘画同步信号接收时间。

[0022] 本实施例中，通过对绘画同步接收信号接收时间的输出，可以分析出帧率与绘画同步信号接收时间的关系，跟准确地分析出移动终端与用户交互过程的流畅性。

[0023] 优选地，输出监测到的帧率包括：

[0024] 生成所述帧率和对应帧数的坐标图；

[0025] 显示所述坐标图。

[0026] 本实施例中，通过坐标图显示帧率，可以更加直观地获得移动终端对用户操作的显示流畅性。

[0027] 优选地，所述用户操作包括：用户在移动终端上的触摸操作和 / 或按键操作。

[0028] 一种移动终端性能检测装置，包括：

[0029] 用户操作监测模块，用于实时监测用户操作；

[0030] 帧率监测模块，用于当监测到用户操作时，监测从所述用户操作的开始时刻到所述用户操作的结束时刻之后的预设时段内的帧率；

[0031] 输出模块，用于输出监测到的帧率。

[0032] 优选地，所述帧率监测模块包括：

[0033] 记录子模块，用于记录每帧对应的实际绘画开始时间和实际绘画结束时间、绘画同步信号接收时间和渲染开始时间；

[0034] 查找子模块，用于根据当前帧的实际绘画结束时间查找当前帧对应的绘画同步信号接收时间，并根据查找到的所述绘画同步信号接收时间查找当前帧对应的渲染开始时间；

[0035] 第一计算子模块，用于计算所述当前帧对应的渲染开始时间与前一帧的渲染开始时间的时间间隔；

[0036] 第二计算子模块，用于根据所述当前帧对应的渲染开始时间与前一帧的渲染开始时间的时间间隔计算当前帧对应的帧率。

[0037] 优选地，所述第一计算子模块，用于对于第一个帧，计算所述第一个帧的渲染开始时间与所述第一个帧的实际绘画开始时间之间的时间间隔；

[0038] 所述第二计算子模块，用于根据所述第一个帧的渲染开始时间与所述第一个帧的实际绘画开始时间之间的时间间隔计算帧率。

[0039] 优选地，所述输出模块，还输出显示每帧的绘画同步信号接收时间。

[0040] 优选地，所述输出模块，用于生成所述帧率和对应帧数的坐标图，显示所述坐标图。

[0041] 一种移动终端，包括有存储器，以及一个或者一个以上的程序，其中一个或者一个

以上程序存储于存储器中,且经配置以由一个或者一个以上处理器执行所述一个或者一个以上程序包含用于进行以下操作的指令:

[0042] 实时监测用户操作;

[0043] 当监测到用户操作时,监测从所述用户操作的开始时刻到所述用户操作的结束时刻之后的预设时段内的帧率;

[0044] 输出监测到的帧率。

[0045] 本发明的其它特征和优点将在随后的说明书中阐述,并且,部分地从说明书中变得显而易见,或者通过实施本发明而了解。本发明的目的和其他优点可通过在所写的说明书、权利要求书、以及附图中所特别指出的结构来实现和获得。

[0046] 下面通过附图和实施例,对本发明的技术方案做进一步的详细描述。

附图说明

[0047] 附图用来提供对本发明的进一步理解,并且构成说明书的一部分,与本发明的实施例一起用于解释本发明,并不构成对本发明的限制。在附图中:

[0048] 图 1 为本发明实施例中移动终端性能检测方法的流程示意图;

[0049] 图 2 为本发明实施例中监测帧率的流程示意图;

[0050] 图 3 为本发明实施例中打开和关闭帧率的选项的示意图;

[0051] 图 4 为本发明实施例中帧率在显示界面上的示意图;

[0052] 图 5 为本发明实施例中移动终端性能检测装置的结构示意图;

[0053] 图 6 为本发明实施例中帧率检测模块的结构示意图;

[0054] 图 7 为本发明实施例中移动终端的结构示意图。

具体实施方式

[0055] 以下结合附图对本发明的优选实施例进行说明,应当理解,此处所描述的优选实施例仅用于说明和解释本发明,并不用于限定本发明。

[0056] 本发明实施例中,为了直观准确地获得操作系统或应用程序的性能及移动终端对用户操作的响应速度进而可以分析出移动终端与用户交互过程的流畅性,使得对移动终端的性能测试更加精确,监测移动终端对于用户操作做出响应时的实时帧率。

[0057] 如图 1 所示,本发明实施例提供一种移动终端性能检测方法,包括以下步骤:

[0058] 步骤 102,实时监测用户操作;

[0059] 优选地,用户操作包括:用户在移动终端上的触摸操作和/或按键操作,即触摸操作和按键操作至少一个操作。

[0060] 步骤 104,当监测到用户操作时,监测从用户操作的开始时刻到用户操作的结束时刻之后的预设时段内的帧率。例如在用户操作结束后的 600ms 内,持续监测帧率。

[0061] 步骤 106,输出监测到的帧率。

[0062] 本实施例中,通过监测用户操作过程中以及用户操作结束后一段时间内的移动终端的帧率,即用户手指按下、抬起这一动作过程中移动终端的显示帧率,可以直观准确地获得操作系统或应用程序的性能及移动终端对用户操作的响应速度进而可以分析出移动终端与用户交互过程的流畅性,使得对移动终端的性能测试更加精确。

[0063] 优选地,如图 2 所示,步骤 104 中,监测帧率,采用如下方式:

[0064] 步骤 202,记录每帧对应的实际绘画开始时间和实际绘画结束时间、绘画同步信号接收时间和渲染开始时间。

[0065] 步骤 204,根据当前帧的实际绘画结束时间查找当前帧对应的绘画同步信号接收时间,并根据查找到的绘画同步信号接收时间查找当前帧对应的渲染开始时间。

[0066] 由于当前帧对应的绘画同步信号的接收应发生在实际绘画结束之后,而对于当前帧的渲染开始则发生在实际绘画结束之后。因此,当前帧对应的绘画同步信号接收时间大于实际绘画结束时间,渲染开始时间大于绘画同步信号接收时间。

[0067] 查找大于当前帧的实际绘画结束时间的所有绘画同步信号接收时间,其中最小的一个为当前帧对应的绘画同步信号接收时间。查找大于该绘画同步信号接收时间的所有渲染开始时间,其中最小的一个为当前帧对应的渲染开始时间。

[0068] 步骤 206,计算当前帧对应的渲染开始时间与前一帧的渲染开始时间的时间间隔。

[0069] 步骤 208,根据当前帧对应的渲染开始时间与前一帧的渲染开始时间的的时间间隔计算当前帧对应的帧率。

[0070] 例如,计算当前帧对应的帧率为该时间间隔的倒数。

[0071] 优选地,监测帧率,还采用如下方式:

[0072] 对于第一个帧,计算第一个帧的渲染开始时间与第一个帧的实际绘画开始时间之间的时间间隔;

[0073] 根据第一个帧的渲染开始时间与第一个帧的实际绘画开始时间之间的时间间隔计算帧率。

[0074] 本实施例中,根据上述方法计算每帧所对应的帧率,得到用户操作过程中的实时的帧率。

[0075] 本实施例中,通过对用户操作过程中移动终端的帧率进行实时监测,可以直观准确地获得操作系统或应用程序的性能及移动终端对用户操作的响应速度进而可以分析出移动终端与用户交互过程的流畅性,使得对移动终端的性能测试更加精确。

[0076] 优选地,该方法还包括:输出每帧的绘画同步信号接收时间。

[0077] 本实施例中,通过对绘画同步接收信号接收时间的输出,可以分析出帧率与绘画同步信号接收时间的关系,跟准确地分析出移动终端与用户交互过程的流畅性。

[0078] 优选地,输出监测到的帧率包括:生成帧率和对应帧数的坐标图;显示坐标图。

[0079] 本实施例中,通过坐标图显示帧率,可以更加直观地获得移动终端对用户操作的显示流畅性。

[0080] 下面以实际实现为例,对帧率的监测过程进行具体说明。

[0081] 如图 3 所示,可以在系统设置的开发者选项中提供一个打开和关闭帧率的选项。

[0082] 在 PhoneWindowManager 类中实现监控该选项打开或者关闭的代码。PhoneWindowManager 类实现了 Android 系统上窗口用户界面 (User Interface, UI) 相关行为的接口,提供了与 Android 系统窗口管理的底层交互。

[0083] 当打开该选项时,创建一个用于显示帧率信息的 View, View 为 Android 系统的显示视图类,其他视图如文本框、按钮都是该类的子类,包括 InputFpsView。将该 View 设置为显示系统最上层、不可获取焦点、不可触摸等,然后创建一个输入事件接收器,将

InputFpsView 注册到上述输入事件接收器中。

[0084] 当有输入事件到来,即监测到用户操作,如触摸操作,包括在触摸屏上按下、滑动等,以及按键操作,都会通知 InputFpsView。InputFpsView 判断输入事件是用户交互的开始或者结束。当判断为开始时,InputFpsView 通知 SurfaceFlinger 开始记录渲染每帧的渲染开始时间,并记录绘画同步信号接收时间。其中, SurfaceFlinger 为 Android 系统中负责合成各个显示 Surface 并将合成结果显示到屏幕上。当判断为结束时, InputFpsView 延迟一段时间(用户停止跟 Android 手机交互后系统动画的时间,比如 600ms) 通知 SurfaceFlinger 停止记录。并将该段时间内记录的结果返回给 InputFpsView。

[0085] 当显示帧率打开且有输入事件到来时,ViewRootImpl 中开始记录当前界面每帧的实际绘画开始时间和实际绘画结束时间,并将记录结果返回给 InputFpsView。

[0086] InputFpsView 将从 SurfaceFlinger 返回的记录结果和从 ViewRootImpl 的返回的记录结果进行处理,结果存放在数组中。例如:将渲染开始时间按从小到大的顺序存放在 frameTimeStamps[] 数组中,绘画同步信号接收时间按从小到大的顺序存放在 vsyncTimeStamps[] 数组中,实际绘画开始时间和实际绘画结束时间按从小到大的顺序存放在 drawTimeStampsStart[] 数组和 drawTimeStampsEnd[] 数组中,对于上述三个数组中的数据做如下处理:

[0087] (1) 对于每个 drawTimeStampsEnd[] 数组中的实际绘画结束时间 T_{ei} (i 的取值范围为 0 到数组元素大小),在 vsyncTimeStamps[] 数组中寻找第一个大于 T_{ei} 的 V_i ,然后在 frameTimeStamps[] 数组中寻找第一个大于 V_i 的 F_i 。直到 drawTimeStampsEnd[] 数组中的所有元素均被处理。

[0088] (2) 根据第(1)步的处理结果,计算当前帧对应的渲染开始时间 F_i 与前一帧的渲染开始时间 $F_{(i-1)}$ 的时间间隔为 $F_i - F_{(i-1)}$,将得到的时间间隔转化为以秒为单位的 D_i ,计算当前帧 i 对应的帧率 FPS_i 为 1 除以 D_i 。

[0089] (3) 对于监测过程中的第一个帧,从 frameTimeStamps[] 数组中获取下标为 0 的渲染开始时间 F_0 ,从 drawTimeStampsStart[] 数组获取实际绘画开始时间 T_{s0} ,计算两者的时间间隔为 $F_0 - T_{s0}$,转化为以秒为单位的 D_0 ,计算当前帧 i 对应的帧率 FPS_0 为 1 除以 D_0 。

[0090] 如图 4 所示,将得到的帧率画在图中,并将 drawTimeStampsStart[] 数组, drawTimeStampsEnd[] 数组,和 vsyncTimeStamps[] 数组相对关系以二维坐标图的方式绘画到移动终端的显示界面上。其中,纵坐标为帧率,横坐标为帧数,横坐标下方为绘画同步信号接收时间。

[0091] 对于帧率的输出方式并不限于坐标图,也可采用文件或表格等其他形式。

[0092] 本发明实施例还提供一种移动终端性能检测装置,如图 5 所示,该装置包括:

[0093] 用户操作监测模块 51,用于实时监测用户操作;

[0094] 帧率监测模块 52,用于当监测到用户操作时,监测从所述用户操作的开始时刻到所述用户操作的结束时刻之后的预设时段内的帧率;

[0095] 输出模块 53,用于输出监测到的帧率。

[0096] 优选地,如图 6 所示,帧率监测模块 52 包括:

[0097] 记录子模块 521,用于记录每帧对应的实际绘画开始时间和实际绘画结束时间、绘画同步信号接收时间和渲染开始时间;

[0098] 查找子模块 522,用于根据当前帧的实际绘画结束时间查找当前帧对应的绘画同步信号接收时间,并根据查找到的所述绘画同步信号接收时间查找当前帧对应的渲染开始时间;

[0099] 第一计算子模块 523,用于计算所述当前帧对应的渲染开始时间与前一帧的渲染开始时间的时间间隔;

[0100] 第二计算子模块 524,用于根据所述当前帧对应的渲染开始时间与前一帧的渲染开始时间的时间间隔计算当前帧对应的帧率。

[0101] 优选地,第一计算子模块 523,用于对于第一个帧,计算所述第一个帧的渲染开始时间与所述第一个帧的实际绘画开始时间之间的时间间隔;

[0102] 第二计算子模块 524,用于根据所述第一个帧的渲染开始时间与所述第一个帧的实际绘画开始时间之间的时间间隔计算帧率。

[0103] 优选地,输出模块 53,还用于输出每帧的绘画同步信号接收时间。

[0104] 优选地,输出模块 53,用于生成所述帧率和对应帧数的坐标图,显示所述坐标图。

[0105] 图 7 是本发明实施例提供的一种移动终端结构示意图。如图 7 所示,该移动终端可以用于实施上述实施例中提供的移动终端性能检测方法。其中,该移动终端可以为手机、平板电脑 pad、穿戴式移动设备(如智能手表)等。优先的:

[0106] 移动终端 300 可以包括通信单元 110、包括有一个或一个以上计算机可读存储介质的存储器 120、输入单元 130、显示单元 140、传感器 150、音频电路 160、WiFi(wireless fidelity,无线保真)模块 170、包括有一个或者一个以上处理核心的处理器 180、以及电源 190 等部件。本领域技术人员可以理解,图 7 中示出的移动终端结构并不构成对移动终端的限定,可以包括比图示更多或更少的部件,或者组合某些部件,或者不同的部件布置。其中:

[0107] 通信单元 110 可用于收发信息或通话过程中,信号的接收和发送,该通信单元 110 可以为 RF(Radio Frequency,射频)电路、路由器、调制解调器、等网络通信设备。特别地,当通信单元 110 为 RF 电路时,将基站的下行信息接收后,交由一个或者一个以上处理器 180 处理;另外,将涉及上行的数据发送给基站。通常,作为通信单元的 RF 电路包括但不限于天线、至少一个放大器、调谐器、一个或多个振荡器、用户身份模块(SIM)卡、收发信机、耦合器、LNA(Low Noise Amplifier,低噪声放大器)、双工器等。此外,通信单元 110 还可以通过无线通信与网络和其他设备通信。无线通信可以使用任一通信标准或协议,包括但不限于 GSM(Global System of Mobile communication,全球移动通讯系统)、GPRS(General Packet Radio Service,通用分组无线服务)、CDMA(Code Division Multiple Access,码分多址)、WCDMA(Wideband Code Division Multiple Access,宽带码分多址)、LTE(Long Term Evolution,长期演进)、电子邮件、SMS(Short Messaging Service,短消息服务)等。存储器 120 可用于存储软件程序以及模块,处理器 180 通过运行存储在存储器 120 的软件程序以及模块,从而执行各种功能应用以及数据处理。存储器 120 可主要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程序(比如声音播放功能、图像播放功能等)等;存储数据区可存储根据移动终端 300 的使用所创建的数据(比如音频数据、电话本等)等。此外,存储器 120 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、或其他易失性固态存储器

件。相应地,存储器 120 还可以包括存储器控制器,以提供处理器 180 和输入单元 130 对存储器 120 的访问。

[0108] 输入单元 130 可用于接收输入的数字或字符信息,以及产生与用户设置以及功能控制有关的键盘、鼠标、操作杆、光学或者轨迹球信号输入。优选地,输入单元 130 可包括触敏表面 131 以及其他输入设备 132。触敏表面 131,也称为触摸显示屏或者触控板,可收集用户在其上或附近的触摸操作(比如用户使用手指、触笔等任何适合的物体或附件在触敏表面 131 上或在触敏表面 131 附近的操作),并根据预先设定的程式驱动相应的连接装置。可选的,触敏表面 131 可包括触摸检测装置和触摸控制器两个部分。其中,触摸检测装置检测用户的触摸方位,并检测触摸操作带来的信号,将信号传送给触摸控制器;触摸控制器从触摸检测装置上接收触摸信息,并将它转换成触点坐标,再送给处理器 180,并能接收处理器 180 发来的命令并加以执行。此外,可以采用电阻式、电容式、红外线以及表面声波等多种类型实现触敏表面 131。除了触敏表面 131,输入单元 130 还可以包括其他输入设备 132。优选地,其他输入设备 132 可以包括但不限于物理键盘、功能键(比如音量控制按键、开关按键等)、轨迹球、鼠标、操作杆等中的一种或多种。

[0109] 显示单元 140 可用于显示由用户输入的信息或提供给用户的信息以及移动终端 300 的各种图形用户接口,这些图形用户接口可以由图形、文本、图标、视频和其任意组合来构成。显示单元 140 可包括显示面板 141,可选的,可以采用 LCD(Liquid Crystal Display, 液晶显示器)、OLED(Organic Light-Emitting Diode, 有机发光二极管)等形式来配置显示面板 141。进一步的,触敏表面 131 可覆盖显示面板 141,当触敏表面 131 检测到在其上或附近的触摸操作后,传送给处理器 180 以确定触摸事件的类型,随后处理器 180 根据触摸事件的类型在显示面板 141 上提供相应的视觉输出。虽然在图 7 中,触敏表面 131 与显示面板 141 是作为两个独立的部件来实现输入和输出功能,但是在某些实施例中,可以将触敏表面 131 与显示面板 141 集成而实现输入和输出功能。

[0110] 移动终端 300 还可包括至少一种传感器 150,比如光传感器、运动传感器以及其他传感器。优选地,光传感器可包括环境光传感器及接近传感器,其中,环境光传感器可根据环境光线的明暗来调节显示面板 141 的亮度,接近传感器可在移动终端 300 移动到耳边时,关闭显示面板 141 和/或背光。作为运动传感器的一种,重力加速度传感器可检测各个方向上(一般为三轴)加速度的大小,静止时可检测出重力的大小及方向,可用于识别手机姿态的应用(比如横竖屏切换、相关游戏、磁力计姿态校准)、振动识别相关功能(比如计步器、敲击)等;至于移动终端 300 还可配置的陀螺仪、气压计、湿度计、温度计、红外线传感器等其他传感器,在此不再赘述。

[0111] 音频电路 160、扬声器 161,传声器 162 可提供用户与移动终端 300 之间的音频接口。音频电路 160 可将接收到的音频数据转换后的电信号,传输到扬声器 161,由扬声器 161 转换为声音信号输出;另一方面,传声器 162 将收集的声音信号转换为电信号,由音频电路 160 接收后转换为音频数据,再将音频数据输出处理器 180 处理后,经 RF 电路 110 以发送给比如另一移动终端,或者将音频数据输出至存储器 120 以便进一步处理。音频电路 160 还可能包括耳塞插孔,以提供外设耳机与移动终端 300 的通信。

[0112] 为了实现无线通信,该移动终端上可以配置有无线通信单元 170,该无线通信单元 170 可以为 WiFi 模块。WiFi 属于短距离无线传输技术,移动终端 300 通过无线通信单元

170 可以帮助用户收发电子邮件、浏览网页和访问流式媒体等,它为用户提供了无线的宽带互联网访问。虽然图 7 示出了无线通信单元 170,但是可以理解的是,其并不属于移动终端 300 的必须构成,完全可以根据需要在不改变发明的本质的范围内而省略。

[0113] 处理器 180 是移动终端 300 的控制中心,利用各种接口和线路连接整个手机的各个部分,通过运行或执行存储在存储器 120 内的软件程序和 / 或模块,以及调用存储在存储器 120 内的数据,执行移动终端 300 的各种功能和处理数据,从而对手机进行整体监控。可选的,处理器 180 可包括一个或多个处理核心;优选的,处理器 180 可集成应用处理器和调制解调处理器,其中,应用处理器主要处理操作系统、用户界面和应用程序等,调制解调处理器主要处理无线通信。可以理解的是,上述调制解调处理器也可以不集成到处理器 180 中。

[0114] 移动终端 300 还包括给各个部件供电的电源 190 (比如电池),优选的,电源可以通过电源管理系统与处理器 180 逻辑相连,从而通过电源管理系统实现管理充电、放电、以及功耗管理等功能。电源 190 还可以包括一个或一个以上的直流或交流电源、再充电系统、电源故障检测电路、电源转换器或者逆变器、电源状态指示器等任意组件。

[0115] 尽管未示出,移动终端 300 还可以包括摄像头、蓝牙模块等,在此不再赘述。

[0116] 具体在本实施例中,移动终端的显示单元是触摸屏显示器,移动终端还包括有存储器,以及一个或者一个以上的程序,其中一个或者一个以上程序存储于存储器中,且经配置以由一个或者一个以上处理器执行一个或者一个以上程序包含用于进行以下操作的指令:

[0117] 实时监测用户操作;

[0118] 当监测到用户操作时,监测从所述用户操作的开始时刻到所述用户操作的结束时刻之后的预设时段内的帧率;

[0119] 输出监测到的帧率。

[0120] 较佳的,还包含用于进行以下操作的指令:记录每帧对应的实际绘画开始时间和实际绘画结束时间、绘画同步信号接收时间和渲染开始时间;

[0121] 根据当前帧的实际绘画结束时间查找当前帧对应的绘画同步信号接收时间,并根据查找到的绘画同步信号接收时间查找当前帧对应的渲染开始时间;

[0122] 计算当前帧对应的渲染开始时间与前一帧的渲染开始时间的时间间隔;

[0123] 根据当前帧对应的渲染开始时间与前一帧的渲染开始时间的的时间间隔,计算当前帧对应的帧率。

[0124] 较佳的,还包含用于进行以下操作的指令:对于第一个帧,计算第一个帧的渲染开始时间与第一个帧的实际绘画开始时间之间的时间间隔;

[0125] 根据第一个帧的渲染开始时间与第一个帧的实际绘画开始时间之间的时间间隔,计算帧率。

[0126] 较佳的,还包含用于进行以下操作的指令:输出每帧的绘画同步信号接收时间。

[0127] 较佳的,还包含用于进行以下操作的指令:生成帧率和对应帧数的坐标图;显示坐标图。

[0128] 本发明的移动终端性能检测方法、装置及移动终端,通过监测用户操作过程中以及用户操作结束后一段时间内的移动终端的帧率,即用户手指按下、抬起这一动作过程中

移动终端的帧率,可以直观准确地获得操作系统或应用程序的性能及移动终端对用户操作的响应速度进而可以分析出移动终端与用户交互过程的流畅性,使得对移动终端的性能测试更加精确。

[0129] 本领域内的技术人员应明白,本发明的实施例可提供为方法、系统、或计算机程序产品。因此,本发明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,本发明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。

[0130] 本发明是参照根据本发明实施例的方法、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。

[0131] 这些计算机程序指令也可存储在能引导计算机或其他可编程数据处理设备以特定方式工作的计算机可读存储器中,使得存储在该计算机可读存储器中的指令产生包括指令装置的制品,该指令装置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。

[0132] 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。

[0133] 显然,本领域的技术人员可以对本发明进行各种改动和变型而不脱离本发明的精神和范围。这样,倘若本发明的这些修改和变型属于本发明权利要求及其等同技术的范围之内,则本发明也意图包含这些改动和变型在内。

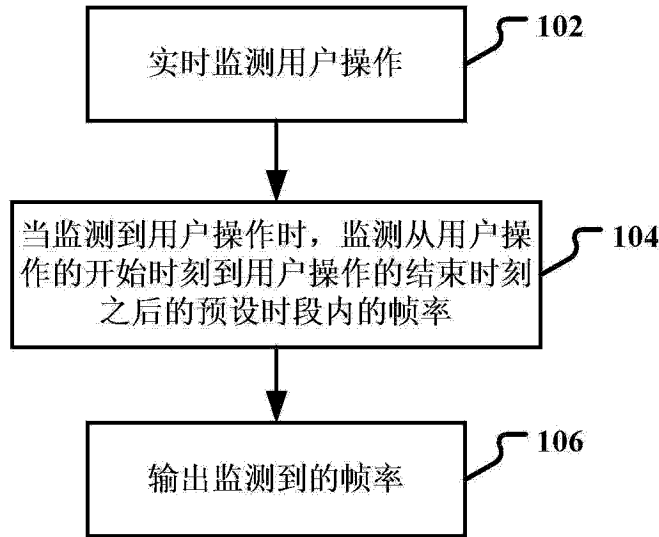


图 1

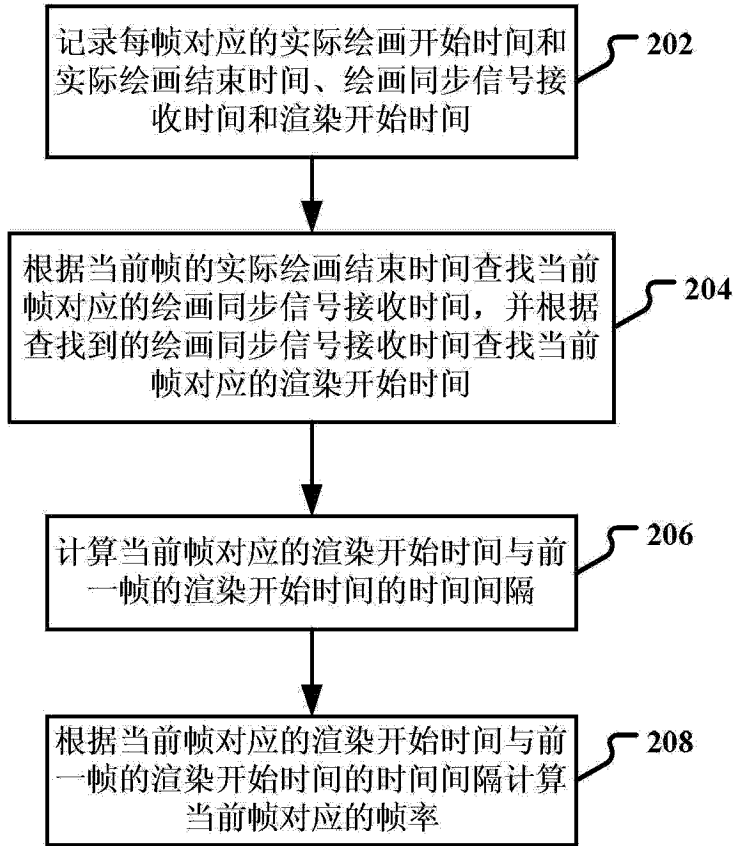


图 2



图 3

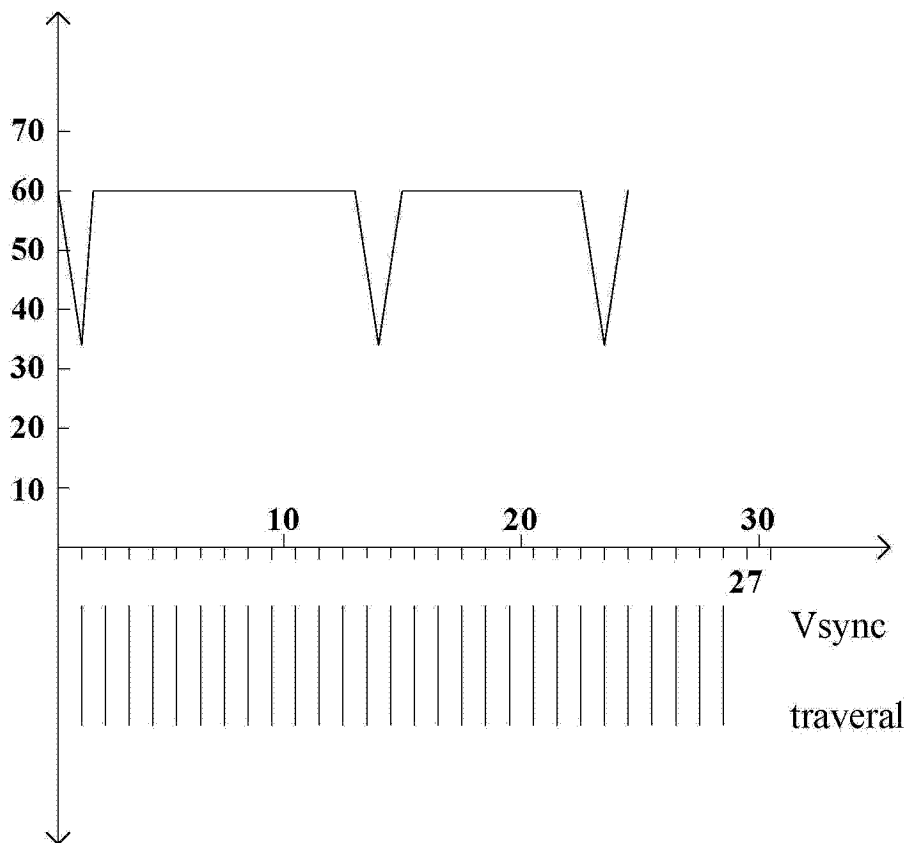


图 4

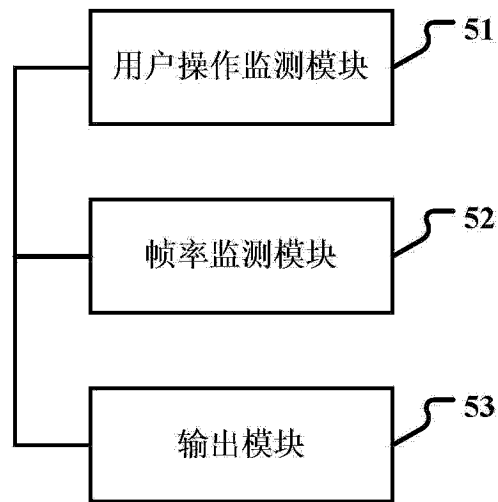


图 5

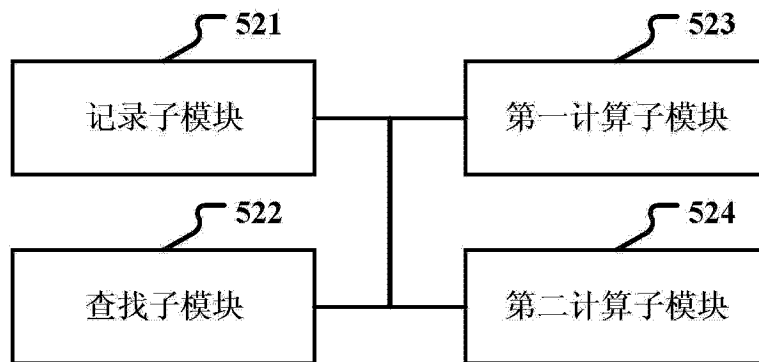


图 6

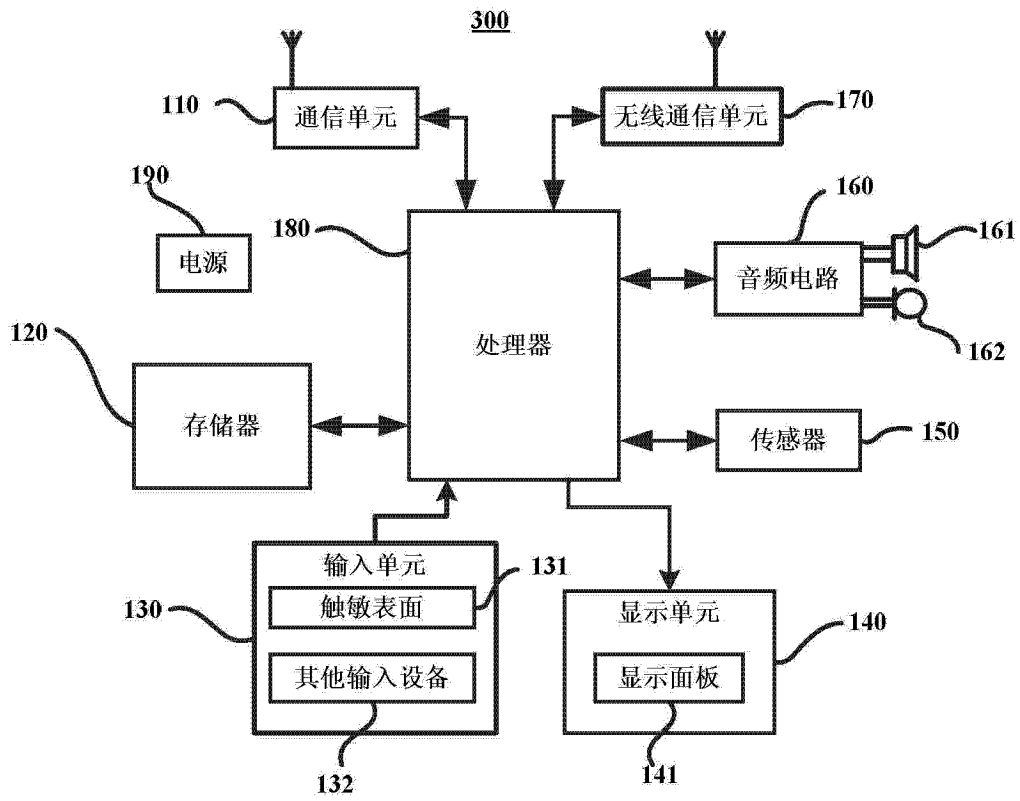


图 7