

(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
2020 年 3 月 26 日 (26.03.2020)



(10) 国际公布号
WO 2020/056878 A1

(51) 国际专利分类号:
G06F 11/34 (2006.01)

(21) 国际申请号: PCT/CN2018/113732

(22) 国际申请日: 2018 年 11 月 2 日 (02.11.2018)

(25) 申请语言: 中文

(26) 公布语言: 中文

(30) 优先权:
201811109357.8 2018年9月21日 (21.09.2018) CN

(71) 申请人: 网宿科技股份有限公司
(WANGSU SCIENCE & TECHNOLOGY CO., LTD.)
[CN/CN]; 中国上海市嘉定区环城路 200
号, Shanghai 201800 (CN)。

(72) 发明人: 苏晓强(SU, Xiaoqiang); 中国上海市嘉定
区环城路200号, Shanghai 201800 (CN)。

(74) 代理人: 北京华智则铭知识产权代理有限公司
(BEIJING HUA-WISDOM INTELLECTUAL
PROPERTY INC.); 中国北京市大兴区宏业路9号
院7号楼1307室, Beijing 100162 (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,
GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS,
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,

(54) Title: METHOD FOR ACQUIRING LAUNCHING DURATION OF APPLICATION INTERFACE, APPARATUS, AND TERMINAL DEVICE

(54) 发明名称: 一种应用界面启动时长的获取方法、装置及终端设备

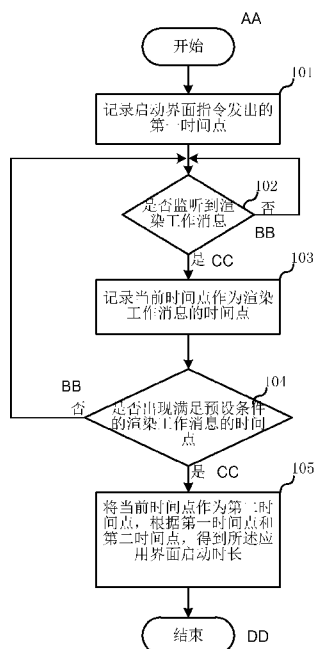


图 1

101 Record a first time point of sending out an interface
launching instruction
102 Has a rendering operation message been detected?
103 Record a current time point as a time point of the
rendering operation message
104 Has a time point of a rendering operation message
meeting a pre-determined condition been detected?
105 Use the current time point as a second time point, and
obtain, according to the first time point and the second time point,
a launching duration of an application interface
AA Start
BB No
CC Yes
DD End

(57) Abstract: A method for acquiring a launching duration of an application interface, an apparatus, and a terminal device, pertaining to the field of terminal apparatuses. The method comprises: recording a first time point of sending out an interface launching instruction; upon each detection of a rendering operation message, recording a current time point as a time point of the rendering operation message, and using the first time point of a rendering operation message that satisfies a pre-determined condition, among time points of multiple rendering operation messages recorded according to a temporal sequence, as a second time point, wherein the pre-determined condition

SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG,
US, UZ, VC, VN, 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))。

is: a time difference between the time point of said rendering operation message and a previous rendering operation message is a pre-determined duration; and using a time difference between the first time point and the second time point as a launching duration of an application interface. The acquired launching duration is accurate, easy to implement, and can be directly integrated in an application program, thereby reducing the workload of testers.

(57) 摘要: 一种应用界面启动时长的获取方法、装置及终端设备, 属于终端设备领域。通过记录启动界面指令发出的第一时间点; 在每一次监听到渲染工作消息时, 记录当前时间点作为渲染工作消息的时间点, 并在按时间顺序记录的多个渲染工作消息的时间点中, 将首个满足预设条件的渲染工作消息的时间点, 作为第二时间点; 其中, 预设条件为: 渲染工作消息的时间点与前一个的渲染工作消息的时间点的时差为预设时长; 将第一时间点和第二时间点的时差, 作为应用界面启动时长, 获得的启动时长准确且实现简单, 可以直接集成在应用程序中, 节省了测试人员的工作量。

一种应用界面启动时长的获取方法、装置及终端设备

技术领域

本发明实施例涉及终端设备领域，特别涉及应用界面启动时长的获取方法及其装置、设备。

背景技术

随着产品业务的不断迭代，业务场景越来越复杂、交互方式越来越多时，必然会造成应用的性能问题，尤其是在应用各界面的启动时长这个方面，随着应用场景的不断复杂化，所耗的工作必然增多，造成的后果之一就是用户从开始操作要打开某个界面到这个界面最终渲染并呈现给用户这段时长会有所增加。而同样的程序，在不同的终端设备中运行时，由于硬件不同、网络不同、设备系统不同等等因素，所反应出的性能问题也有所不同，或许在某些设备上性能良好，界面启动时长很短，但在某些设备上就会出现性能问题，出现界面启动耗时过长问题。因此，能够正确获取、并采集用户使用应用过程中，各界面的启动时长数据，对于开发人员的性能优化工作有着重要的意义。

由于单单只靠系统 API（Application Program Interface 应用程序接口）无法正确计算出界面的启动时长，最多只能获取到系统发出启动界面指令这个时刻的时间与将要启动的界面框架搭建完毕时刻的时间，这个时刻界面还未绘制渲染完成，而我们需要的是界面完全呈现到用户眼前的这个时刻的时间点。

现有技术中的普遍做法是通过测试人员人工的来测试、体验界面的启动时长，以这点作为依据交给开发人员进行性能优化。但这种方法仍然存在以下不足：

（1）测试人员工作量大，且用例覆盖面低，无法覆盖市场所有设备各种场景。

（2）只在应用测试阶段采集相关性能数据，一旦应用上线则无法继续收集相关信息。

因为界面最终绘制并渲染呈现到用户眼前的这个时刻，只能通过肉眼观察判断，系统没有给出这个时刻的回调通知。因此，基于这点，目前有一些公开的技术方案则是提出，借助图片采集、识别、匹配技术，结合设备的摄像头等硬件，通过脚本达到让机器代替人工来判断界面渲染完成的时刻，以此来采集相关数据，节省人力工作量。

这种技术方案虽然可以准确的获取界面的启动时长，但需要相关硬件配置支持，且实现难度较大，在上线的产品中大量应用不大现实。

发明内容

本发明实施方式的目的在于提供一应用界面启动时长的获取方法、装置及终端设备，能够准确获取到应用界面的启动时长，而且可以直接集成在应用程序中，实现简单，节省了测试人员的工作量。

5 为解决上述技术问题，本发明的实施方式提供了一种应用界面启动时长的获取方法，包括以下步骤：

记录启动界面指令发出的第一时间点；

10 监听渲染工作消息；其中，每一次监听到渲染工作消息时，记录当前时间点作为渲染工作消息的时间点，并在按时间顺序记录的多个渲染工作消息的时间点中，将首个满足预设条件的渲染工作消息的时间点，作为第二时间点；其中，预设条件为：渲染工作消息的时间点与后一个渲染工作消息的时间点的时差为预设时长；

将第一时间点和第二时间点的时差，作为应用界面启动时长。

本发明的实施方式还提供了一种应用界面启动时长的获取装置，包括：

记录模块，用于记录启动界面指令发出的第一时间点；

15 监听模块，用于监听渲染工作消息；

记录模块还用于在监听模块每一次监听到渲染工作消息时，记录当前时间点作为渲染工作消息的时间点；

20 应用界面启动时长确定模块，用于在按时间顺序记录的多个渲染工作消息的时间点中，将首个满足预设条件的渲染工作消息的时间点，作为第二时间点；其中，预设条件为：渲染工作消息的时间点与后一个的渲染工作消息的时间点的时差为预设时长，将第一时间点和第二时间点的时差，作为应用界面启动时长。

本发明的实施方式还提供了一种终端设备，包括：

至少一个处理器；以及，

与上述至少一个处理器通信连接的存储器；其中，

25 存储器存储有可被上述至少一个处理器执行的指令，该指令被上述至少一个处理器执行，以使上述至少一个处理器能够执行上述的应用界面启动时长的获取方法。

本发明的实施方式还提供了一种计算机可读存储介质，存储有计算机程序，计算机程序被处理器执行时实现上述的应用界面启动时长的获取方法。

30 本发明实施方式相对于现有技术而言，通过记录启动界面指令发出的第一时间点作为应用程序界面开始启动的时间；利用渲染机制中，当设备完成复杂且耗时的工作后将按照严格

的渲染频率来刷新界面的特性（即相邻两个渲染消息之间的间隔为一特定时间间隔的特性），获取到渲染机制首次按渲染频率刷新界面的时间点作为应用界面启动完成的结束时间，从而根据开始启动的时间和结束时间，即可得到准确的应用界面的启动时长。并且由于这些时间点的记录都是由终端设备操作的，所以可以直接集成在应用程序中，以此达到采集线上用户的大量数据，用例覆盖广的目的，同时不需要专门的测试人员进行大量测试，可以节省人工工作量。

另外，上述监听渲染工作消息，包括：持续监听渲染工作消息，直至监听到第 N 个渲染工作消息；其中，所述 N 为预设数值。在大部分场景下，应用的界面启动时所处理的工作大多数都是分布在一定帧数内，因此采集预设的 N 帧内每个渲染消息执行的时间点信息，基本就可以覆盖大部分场景，供开发人员分析性能问题。

另外，上述监听渲染工作消息，包括：在每一次监听到渲染工作消息后，判断是否出现满足预设条件的渲染工作消息的时间点；在出现满足预设条件的渲染工作消息的时间点时，停止渲染工作消息的监听。由于当应用处于正常可交互、流畅的情况下，渲染机制是会严格按照一定频率刷新屏幕的，即两个渲染工作消息之间的时差一定，因此若出现满足预设条件的

时间点，则表明应用已处于正常可交互情况，不需要再记录更多的时间点了。

另外，上述记录启动界面指令发出的第一时间点，包括：监听应用所有界面的生命周期；在接收到应用所有界面的生命周期的监听回调通知时，将接收到监听回调通知的时间点，记录为所述第一时间点。

另外，上述的应用界面启动时长的获取方法，还包括：将记录的多个渲染工作消息的时间点，上报至服务器。记录的一串时间点信息中包括了一些重要节点的信息，以及界面渲染的大致时间信息，将这些信息上传服务器可方便开发人员分析性能问题。

附图说明

一个或多个实施例通过与之对应的附图中的图片进行示例性说明，这些示例性说明并不构成对实施例的限定。

图 1 是根据本发明第一实施方式提供的应用界面启动时长的获取方法的流程图；

图 2 是根据本发明第二实施方式提供的应用界面启动时长的获取方法的流程图；

图 3 是根据本发明第二实施方式中监听到的渲染工作消息的时间点的示意图；

图 4 是根据本发明第三实施方式提供的应用界面启动时长的获取装置的结构示意图；

图 5 是根据本发明第四实施方式提供的终端设备的结构示意图。

具体实施方式

为使本发明实施例的目的、技术方案和优点更加清楚，下面将结合附图对本发明的各实施方式
5 进行详细的阐述。然而，本领域的普通技术人员可以理解，在本发明各实施方式中，为了使读者更好地理解本申请而提出了许多技术细节。但是，即使没有这些技术细节和基于以下各实施方式的种种变化和修改，也可以实现本申请所要求保护的技术方案。

本发明的第一实施方式涉及一种应用界面启动时长的获取方法。本实施方式中，记录启动界面指令发出的第一时间点；监听渲染工作消息；其中，每一次监听到渲染工作消息时，记录当前时间点作为渲染工作消息的时间点，并在按时间顺序记录的多个渲染工作消息的时间点中，将首个满足预设条件的渲染工作消息的时间点，作为第二时间点；其中，预设条件
10 为：渲染工作消息的时间点与后一个的渲染工作消息的时间点的时差为预设时长；将第一时间点和第二时间点的时差，作为应用界面启动时长。

通过记录启动界面指令发出的第一时间点作为应用程序界面开始启动的时间；利用渲染机制中，当设备完成复杂且耗时的工作后将按照严格的渲染频率来刷新界面的特性（即相邻两个渲染消息之间的间隔为一特定时间间隔的特性），获取到渲染机制首次按渲染频率刷新界
15 面的时间点作为应用界面启动完成的结束时间，从而根据开始启动的时间和结束时间，即可得到准确的应用界面的启动时长。并且由于这些时间点的记录都是由终端设备操作的，所以可以直接集成在应用程序中，以此达到采集线上用户的大量数据，用例覆盖广的目的，同时不需要专门的测试人员进行大量测试，可以节省人工工作量。

下面对本实施方式的应用界面启动时长的获取方法的实现细节进行具体的说明，以下内
20 容仅为方便理解提供的实现细节，并非实施本方案的必须。

本实施方式中的应用界面启动时长的获取方法如图 1 所示，具体包括以下步骤：

步骤 101：终端设备记录启动界面指令发出的第一时间点。本实施方式的终端设备可以是手机、电脑、平板等终端设备，在此不一一赘述。

具体地说，以 Android 系统的终端为例，终端设备记录启动界面指令发出的第一时间点
25 可以有两种方式，第一种方式是通过在源程序中埋点，在每条 `startActivity()` 指令的执行时刻主动记录当前时间点 T1，T1 即为发出启动界面指令的第一时间点。第二种方式是监听应用所用界面的生命周期，在接收到应用所有界面的生命周期的监听回调通知时，将接收到监听回调通知的时间点，记录为第一时间点。也就是说，在终端应用的入口中，通过“`ActivityLifecycleCallbacks`”指令统一注册监听应用所有界面的生命周期，在接收到
30 `onActivityCreated()` 回调时，将当前时间点作为 T1 时间点记录。值得注意的是，本实施例中

是以 Android 系统为例进行说明，但本发明所提供的技术方案同样能应用在其他操作系统中，而并不限制于应用在 Android 系统中，当应用在其他操作系统中时，所调用的接口可基于操作系统的设定而调整。

其中，第一种方式记录的时间点精准，但实现较复杂，需要在源程序中各处埋点、处理相关代码。而第二种方式记录的时间点会稍微有点延迟，但实现简单，只需要在源程序入口处处理即可，而且记录的时间点虽然是在界面启动指令执行之后，但却是在界面对应的各种业务场景工作、绘制工作、渲染工作之前，因此这点时间延迟并不会太大影响。这两种方式都可以实现本步骤，具体实施时可根据需要选择以第一种方式或第二种方式，记录启动界面指令发出的第一时间点。

在步骤 102 至步骤 105 中，监听渲染工作消息。由于终端设备系统，如 Android 系统，是基于消息驱动的，一个应用的运行过程中就是在处理一个个的消息，每个消息都会携带一些任务，比如打开界面指令是一个任务，绘制界面是一个任务，处理业务场景工作的是一个任务，渲染呈现给用户也是一个任务。在某一时刻，只会有一个消息处于执行中，其余消息都会在消息队列中等待执行。终端设备界面的刷新、呈现都是依赖于渲染机制，都是以一个固定的频率来刷新界面，比如 Android 设备，固定以 60 帧/秒的频率刷新，也就是说，正常情况下，每隔 16.6 ms 会有一个渲染界面的任务被消息携带并放于消息队列中等待执行。

而应用的界面启动之所以会耗时，是因为界面对应的业务场景工作、界面 UI(User Interface 用户接口)的计算和绘制、界面的渲染操作都是在界面框架由系统搭建完毕之后会去执行的操作。一旦由于产品的业务场景过于复杂，所需要处理的工作过多，界面的 UI 过于复杂，UI 控件嵌套层次过多，那么这些操作就都会过于耗时。用户最终看到界面渲染并呈现的时刻，是需要等这些工作都处理完毕后，那么就会造成不好的用户体验。因此，在本实施方式中，通过对渲染工作消息的监听，利用渲染机制中，当设备完成复杂且耗时的工作后将按照严格的渲染频率来刷新界面的特性（即相邻两个渲染消息之间的间隔为一特定时间间隔的特性），获取到应用界面启动完成的结束时间。

下面对本实施方式的步骤 102 至步骤 104 进行具体说明：

步骤 102：判断是否监听到渲染工作消息，若是，则执行步骤 103，否则继续执行步骤 102。

具体地说，终端设备可通过以下方式实现对渲染工作消息的监听：在源程序中，可通过系统的“Choreographer.postFrameCallback()”指令来注册监听渲染工作执行的回调通知。当系统的消息队列已经轮到当前界面的渲染工作消息执行时，系统的

Choreographer.FrameCallback.doFrame() 会被回调, 将接收到回调通知的时刻, 作为监听到渲染工作消息的时间点。

也就是说, 终端设备判断是否监听到渲染工作消息, 实际上就是在监听渲染工作是否被执行。若接收到回调通知时则判定渲染工作已执行, 即监听到渲染工作消息, 此时执行步骤 103, 记录当前时间点作为监听到的渲染工作消息的时间点; 否则继续判断是否监听到渲染工作消息。

步骤 104: 判断是否出现满足预设条件的渲染工作消息的时间点, 若是, 则进入步骤 105, 否则进入步骤 102, 继续监听下一个渲染工作消息。

本实施方式中的预设条件为渲染工作消息的时间点与后一个的渲染工作消息的时间点的时差为预设时长, 其中, 预设时长可以为应用程序在运行过程中的相邻两次刷新界面的间隔时长。若终端设备判定当前渲染工作消息的时间点满足预设条件, 则表明应用程序已正常运行, 无需继续监听渲染工作消息, 直接进入步骤 105, 否则进入步骤 102, 继续监听。

具体地, 可以通过以下方式判断渲染工作消息的时间点是否满足预设条件:

比对记录的最近的两个渲染工作消息的时间点, 计算它们的斜率, 当计算出最近记录到的时间点已经趋于 60 帧/秒, 也就是两个时间点之间大概间隔 16.6ms 时, 就可以认为当前页面已经处理完复杂的工作任务, 那么就可以认为界面已经呈现给用户了。

由于终端设备界面的刷新、呈现都是依赖于渲染机制, 都是以一个固定的频率来刷新界面, 比如安卓设备, 固定以 60 帧/秒的频率刷新, 也就是说正常情况下, 应用程序在运行过程中每隔 16.6 ms 会有一个渲染界面的任务被消息携带并放于消息队列中等待执行, 若相邻两个渲染工作消息的时间点之间的时差为应用程序在运行过程中的相邻两次刷新界面的间隔时长, 则表明该应用程序界面已经绘制完成。首个满足这个条件的时间点, 即为应用程序界面绘制完毕且渲染呈现在用户眼前的时间点。

如果相邻两个渲染工作消息的时间点之间的时差并不是应用程序在运行过程中的相邻两次刷新界面的间隔时长, 则表明该应用程序界面还未绘制结束, 需要继续监听。

步骤 105: 将满足预设条件的渲染工作消息的时间点作为第二时间点, 根据第一时间点和第二时间点, 得到应用界面启动时长。

具体地说, 由于在步骤 102 至步骤 104 中, 每一次监听到渲染工作消息时, 都会记录当前时间点作为渲染工作消息的时间点, 直至得到满足预设条件的渲染工作消息的时间点, 该时间点即为应用程序界面绘制完成的时间点, 记为第二时间点, 表明应用已处于正常可交互情况, 不需要再记录更多的时间点了, 终端设备不再继续监听渲染工作消息。

由于第一时间点为启动界面指令的发出时刻，第二时间点为程序界面绘制完成的时刻，因此，根据第一时间点和第二时间点，即可得到应用界面启动时长。即，将第一时间点和第二时间点的时差，作为应用界面启动时长。

在具体实现时，可以由终端设备自身实现第一时间点和第二时间点的时差计算，也可以
5 由服务器实现第一时间点和第二时间点的时差计算。例如，终端设备将记录的多个渲染工作消息的时间点，上报至服务器，通过服务器计算第一时间点和第二时间点的时差，从而得到应用界面启动时长。由于记录的渲染工作消息时间点包括了界面渲染的大致时间信息，因此，将记录的多个渲染工作消息的时间点，上报至服务器，还可以方便开发人员分析设备的性能问题。而且，如将记录的多个渲染工作消息的时间点，上报至服务器，则还可以由服务器根
10 据记录的渲染工作消息的时间点，检测是否出现满足预设条件的渲染工作消息的时间，并在检测到出现满足预设条件的渲染工作消息的时间时，通知给终端设备。也就是说，在实际应用中，终端设备还可以根据服务器的通知消息，判断是否出现满足预设条件的渲染工作消息的时间点。

在本实施方式中，通过记录启动界面指令发出的第一时间点作为应用程序界面开始启动
15 的时间；利用渲染机制中，当设备完成复杂且耗时的工作后将按照严格的渲染频率来刷新界面的特性（即相邻两个渲染消息之间的间隔为一特定时间间隔的特性），获取到渲染机制首次按渲染频率刷新界面的时间点作为应用界面启动完成的结束时间，从而根据开始启动的时间和结束时间，即可得到准确的应用界面的启动时长。并且由于这些时间点的记录都是由终端设备操作的，所以可以直接集成在应用程序中，以此达到采集线上用户的大量数据，用例覆
20 盖广的目的，同时不需要专门的测试人员进行大量测试，可以节省人工工作量。

本发明的第二实施方式涉及一种应用界面启动时长的获取方法。第二实施方式与第一实施方式大致相同，主要区别之处在于：在第一实施方式中，在每一次监听到渲染工作消息后，判断是否出现满足预设条件的渲染工作消息的时间点，在出现满足预设条件的渲染工作消息的时间点时，停止渲染工作消息的监听。而在本发明第二实施方式中，持续监听渲染工作消
25 息，直至监听到第 N 个渲染工作消息；其中，N 为预设数值。

本实施方式中的应用界面启动时长的获取方法如图 2 所示，具体包括以下步骤。

步骤 201：终端设备记录启动界面指令发出的第一时间点。

步骤 202：判断是否监听到渲染工作消息，若是，则执行步骤 203，否则继续执行步骤
202。

30 步骤 203：记录当前时间点作为监听到的渲染工作消息的时间点。

至步骤 203 与第一实施方式的步骤 101 至步骤 103 大致相同，为避免重复，此处不再一一赘述。

步骤 204：判断监听到的渲染工作消息的个数是否达到 N 个，若是，则执行步骤 205，否则执行步骤 202，监听下一个渲染工作消息。

5 具体地说，N 为预设值，具体可设为大于 5 的数值。在源程序中，终端设备可注册监听渲染工作执行的回调通知。当系统的消息队列已经轮到当前界面的渲染工作消息执行时，会被回调，在这里记录当前时间点，循环这个过程，直到记录了 N 条渲染工作消息，终端设备不再监听渲染工作消息。在大部分场景下，终端设备应用的界面启动时所处理的工作大多数都是分布在一定帧数内，因此采集预设的 N 帧内每个渲染消息执行的时间点信息，基本就可
10 以覆盖大部分场景，供开发人员分析性能问题，无需继续监听。

 在具体实现时，N 可取为 10。由于渲染机制当应用处于正常可交互、流畅的情况下，是会严格按照 60 帧/秒刷新屏幕，而在大部分场景下，应用的界面启动时所处理的工作大多数都是分布在 5 帧内，因此，N 大于 5 就已经能够覆盖到较多的场景，将 N 取为 10，可进一步保证得到满足条件的渲染工作消息的时间点。当然也可以修改这个数值，但建议是先用 10 作
15 为基准，然后再收到大量的数据之后，对数据进行分析确认好，再看是否需要进行相应的缩短或增加。

 步骤 205：从 N 个渲染工作消息的时间点中得到第二时间点，并根据第一时间点和第二时间，得到应用界面启动时长。

 具体地说，终端设备已经记录了 N 个渲染工作消息的时间点，对这 N 这时间点逐一判断，
20 若某一渲染工作消息的时间点为首个满足预设条件的时间点作为第二时间点，预设条件为：渲染工作消息的时间点与后一个渲染工作消息的时间点的时差为预设时长（如应用程序在运行过程中的相邻两次刷新界面的间隔时长），第二时间点即为程序界面启动完成的时间点。因此，第一时间点与第二时间点的差值即可作为应用界面启动时长。

 以图 4 所示的监听到的渲染工作消息的时间点为例，图 4 中的 T1-T10 为监听到的 10 个
25 渲染工作消息的时间点，可见，应用界面在启动之后，总共有三个较为耗时的操作，一个是 T1-T2 耗费了 2023ms，一个是 T2-T3 耗费了 1020ms，一个是 T3-T4 耗时了 300ms，之后趋向于正常的渲染频率：16.6ms。那么就可以认为 T1-T4 的时间：3043ms 为该界面的启动时长，开发人员便可以根据这些信息去查找上述耗时的三个操作都是什么任务，是否可以优化处理。

30 上面各种方法的步骤划分，只是为了描述清楚，实现时可以合并为一个步骤或者对某些

步骤进行拆分，分解为多个步骤，只要包括相同的逻辑关系，都在本专利的保护范围内；对算法中或者流程中添加无关紧要的修改或者引入无关紧要的设计，但不改变其算法和流程的核心设计都在该专利的保护范围内。

本发明第三实施方式涉及一种应用界面启动时长的获取装置，如图所示，包括：

5 记录模块 401，用于记录启动界面指令发出的第一时间点；

监听模块 402，用于监听渲染工作消息；

记录模块 401 还用于在监听模块 402 每一次监听到渲染工作消息时，记录当前时间点作为渲染工作消息的时间点；

10 应用界面启动时长确定模块 403，用于在按时间顺序记录的多个渲染工作消息的时间点中，将首个满足预设条件的渲染工作消息的时间点，作为第二时间点；其中，预设条件为：渲染工作消息的时间点与后一个的渲染工作消息的时间点的时差为预设时长，将第一时间点和第二时间点的时差，作为所述应用界面启动时长。

15 不难发现，本实施方式为与第一实施方式或第二实施方式相对应的系统实施例，本实施方式可与第一实施方式或第二实施方式互相配合实施。第一实施方式或第二实施方式中提到的相关技术细节在本实施方式中依然有效，为了减少重复，这里不再赘述。相应地，本实施方式中提到的相关技术细节也可应用在所述第一实施方式中或第二实施方式中。

20 值得一提的是，本实施方式中所涉及到的各模块均为逻辑模块，在实际应用中，一个逻辑单元可以是一个物理单元，也可以是一个物理单元的一部分，还可以以多个物理单元的组合实现。此外，为了突出本发明的创新部分，本实施方式中并没有将与解决本发明所提出的技术问题关系不太密切的单元引入，但这并不表明本实施方式中不存在其它的单元。

本发明第四实施方式涉及一种终端设备，如图 5 所示，包括：

至少一个处理器 501；以及，

与至少一个处理器 501 通信连接的存储器 502；其中，

25 存储器 502 存储有可被至少一个处理器 501 执行的指令，指令被至少一个处理器 501 执行，以使至少一个处理器 501 能够执行上述应用界面启动时长的获取方法。

其中，存储器 502 和处理器 501 采用总线方式连接，总线可以包括任意数量的互联的总线和桥，总线将一个或多个处理器 501 和存储器 502 的各种电路连接在一起。总线还可以将诸如外围设备、稳压器和功率管理电路等之类的各种其他电路连接在一起，这些都是本领域所公知的，因此，本文不再对其进行进一步描述。总线接口在总线和收发机之间提供接口。
30 收发机可以是一个元件，也可以是多个元件，比如多个接收器和发送器，提供用于在传输介

质上与各种其他装置通信的单元。经处理器 501 处理的数据通过天线在无线介质上进行传输，进一步，天线还接收数据并将数据传送给处理器 501。

5 处理器 501 负责管理总线和通常的处理，还可以提供各种功能，包括定时，外围接口，电压调节、电源管理以及其他控制功能。而存储器 502 可以被用于存储处理器 501 在执行操作时所使用的数据。

处理器负责管理总线和通常的处理，还可以提供各种功能，包括定时，外围接口，电压调节、电源管理以及其他控制功能。而存储器可以被用于存储处理器在执行操作时所使用的数据。

10 本发明第五实施方式涉及一种计算机可读存储介质，存储有计算机程序。计算机程序被处理器执行时实现上述方法实施例。

即，本领域技术人员可以理解，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，该程序存储在一个存储介质中，包括若干指令用以使得一个设备（可以是单片机，芯片等）或处理器（processor）执行本申请各个实施例所述方法的全部或部分步骤。而前述的存储介质包括：U 盘、移动硬盘、只读存储器（ROM，
15 Read-Only Memory）、随机存取存储器（RAM，Random Access Memory）、磁碟或者光盘等各种可以存储程序代码的介质。

本领域的普通技术人员可以理解，上述各实施方式是实现本发明的具体实施例，而在实际应用中，可以在形式上和细节上对其作各种改变，而不偏离本发明的精神和范围。

权 利 要 求 书

1. 一种应用界面启动时长的获取方法，其特征在于，包括：

记录启动界面指令发出的第一时间点；

5 监听渲染工作消息；其中，每一次监听到所述渲染工作消息时，记录当前时间点作为所述渲染工作消息的时间点，并在按时间顺序记录的多个所述渲染工作消息的时间点中，将首个满足预设条件的所述渲染工作消息的时间点，作为第二时间点；其中，所述预设条件为：所述渲染工作消息的时间点与后一个所述渲染工作消息的时间点的时差为预设时长；

将所述第一时间点和所述第二时间点的时差，作为所述应用界面启动时长。

10 2. 根据权利要求1所述的应用界面启动时长的获取方法，其特征在于，所述监听渲染工作消息，包括：

持续监听所述渲染工作消息，直至监听到第N个所述渲染工作消息；其中，所述N为预设数值。

15 3. 根据权利要求2所述的应用界面启动时长的获取方法，其特征在于，所述预设数值大于5。

4. 根据权利要求1所述的应用界面启动时长的获取方法，其特征在于，所述监听渲染工作消息，包括：

在每一次监听到所述渲染工作消息后，判断是否出现满足所述预设条件的所述渲染工作消息的时间点；

20 在出现满足所述预设条件的所述渲染工作消息的时间点时，停止所述渲染工作消息的监听。

5. 根据权利要求1至4中任一项所述的应用界面启动时长的获取方法，其特征在于，所述记录启动界面指令发出的第一时间点，包括：

监听应用所有界面的生命周期；

25 在接收到所述应用所有界面的生命周期的监听回调通知时，将接收到所述监听回调通知的时间点，记录为所述第一时间点。

6. 根据权利要求1至4中任一项所述的应用界面启动时长的获取方法，其特征在于，还包括：

将所述记录的多个所述渲染工作消息的时间点，上报至服务器。

30 7. 根据权利要求1至4中任一项所述的应用界面启动时长的获取方法，其特征在于，

所述预设时长为应用程序在运行过程中的相邻两次刷新界面的间隔时长。

8. 一种应用界面启动时长的获取装置，其特征在于，包括
记录模块，用于记录启动界面指令发出的第一时间点；
监听模块，用于监听渲染工作消息；

5 所述记录模块还用于在所述监听模块每一次监听到所述渲染工作消息时，记录当前时间点作为所述渲染工作消息的时间点；

应用界面启动时长确定模块，用于在按时间顺序记录的多个所述渲染工作消息的时间点中，将首个满足预设条件的所述渲染工作消息的时间点，作为第二时间点；其中，所述预设条件为：所述渲染工作消息的时间点与后一个所述渲染工作消息的时间点的时差为预设时长，
10 将所述第一时间点和所述第二时间点的时差，作为所述应用界面启动时长。

9. 一种终端设备，其特征在于，包括：

至少一个处理器；以及，

与所述至少一个处理器通信连接的存储器；其中，

所述存储器存储有可被所述至少一个处理器执行的指令，所述指令被所述至少一个处理
15 器执行，以使所述至少一个处理器能够执行如权利要求 1 至 7 中任一所述的应用界面启动时长的获取方法。

10. 一种计算机可读存储介质，存储有计算机程序，其特征在于，所述计算机程序被处理器执行时实现权利要求 1 至 7 中任一项所述的应用界面启动时长的获取方法。

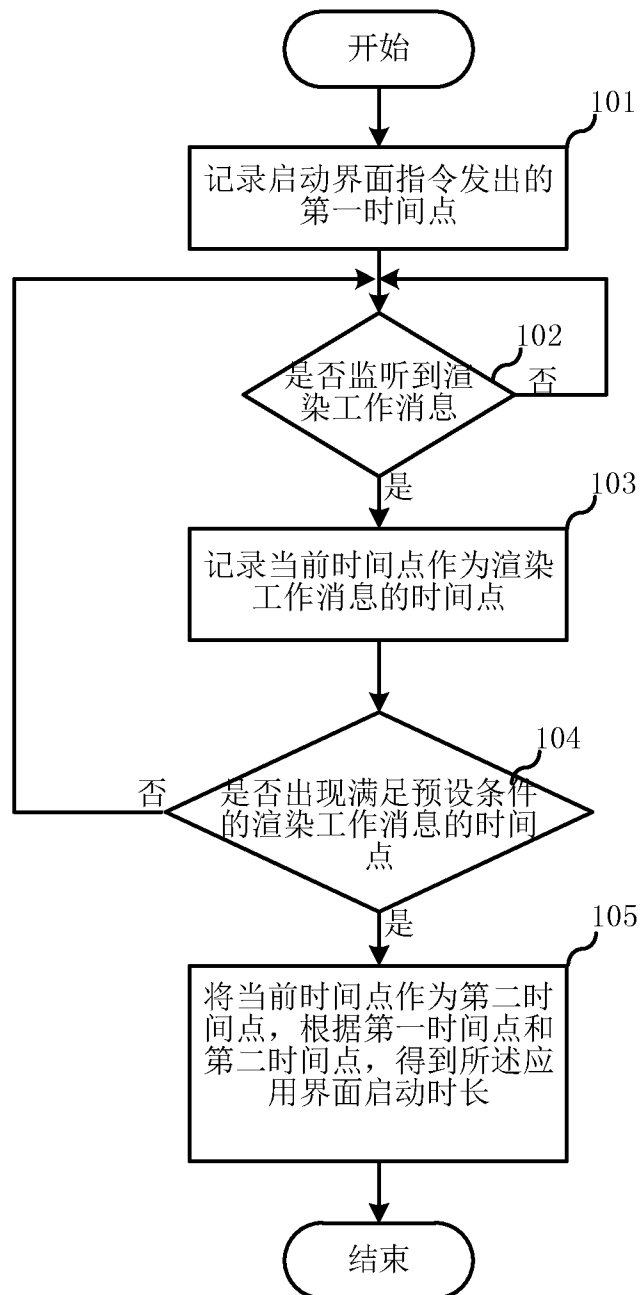


图 1

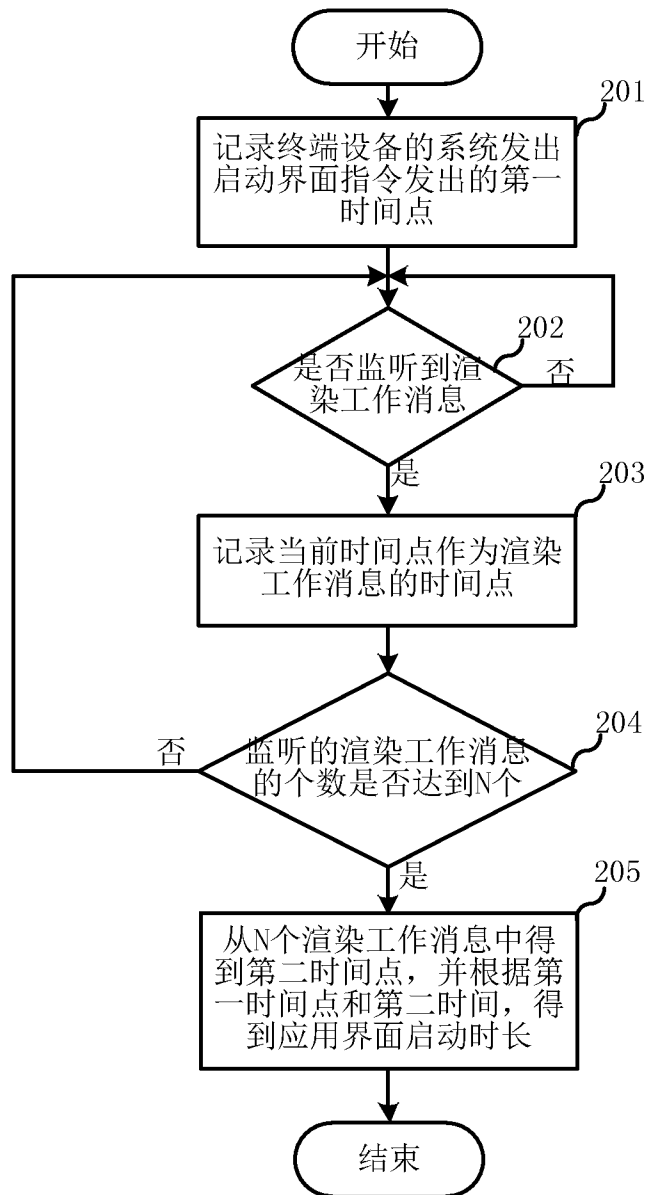


图 2

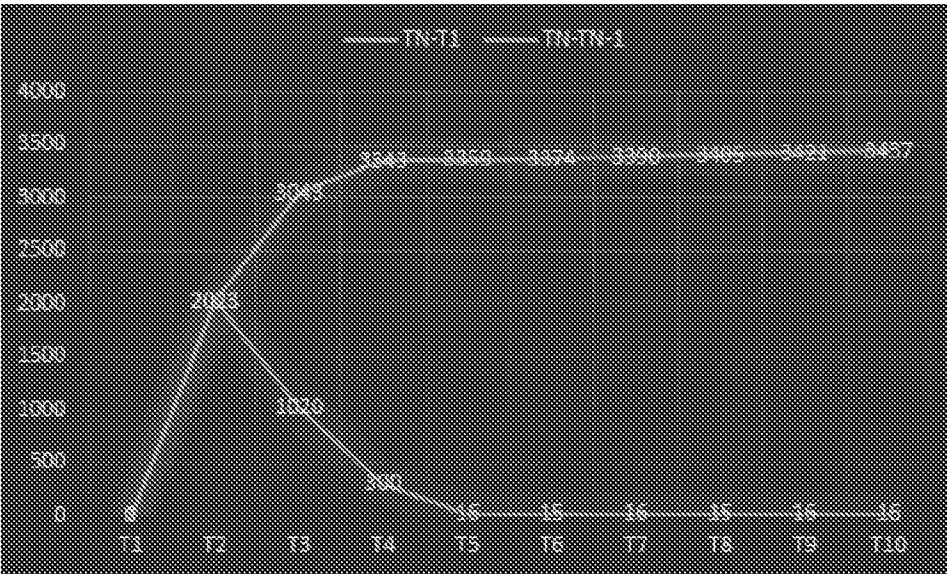


图 3

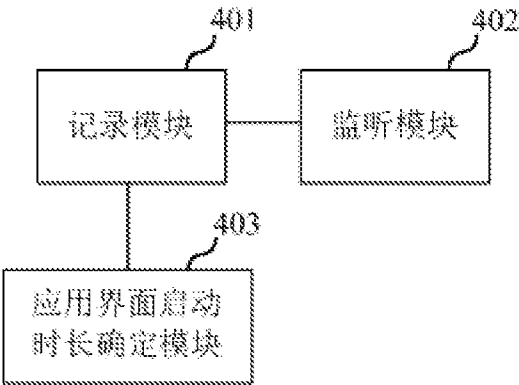


图 4

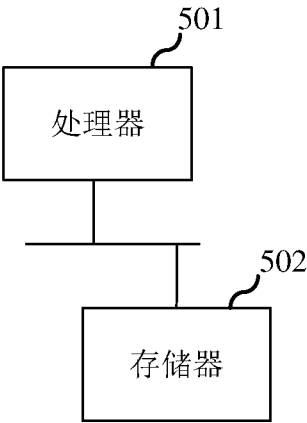


图 5

INTERNATIONAL SEARCH REPORT

International application No.

PCT/CN2018/113732

A. CLASSIFICATION OF SUBJECT MATTER

G06F 11/34(2006.01)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, CNKI, WPI, EPODOC: 启动, 加载, 渲染, 时长, 时间点, 监听, start, load, render, time, duration, point

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	CN 107797904 A (FUJIAN TQ DIGITAL CO., LTD.) 13 March 2018 (2018-03-13) description, paragraphs [0005]-[0015]	1-10
A	CN 108196904 A (GUANGZHOU VIP SOFTWARE CO., LTD.) 22 June 2018 (2018-06-22) entire document	1-10
A	CN 104424092 A (TENCENT TECHNOLOGY SHENZHEN CO., LTD.) 18 March 2015 (2015-03-18) entire document	1-10
A	CN 106897209 A (BEIJING QIHOO TECHNOLOGY CO., LTD. ET AL.) 27 June 2017 (2017-06-27) entire document	1-10
A	CN 105573755 A (BEIJING QIHOO TECHNOLOGY CO., LTD. ET AL.) 11 May 2016 (2016-05-11) entire document	1-10

☐ 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

21 May 2019

Date of mailing of the international search report

19 June 2019

Name and mailing address of the ISA/CN

State Intellectual Property Office of the P. R. China
No. 6, Xitucheng Road, Jimenqiao Haidian District, Beijing
100088
China

Facsimile No. (86-10)62019451

Authorized officer

Telephone No.

INTERNATIONAL SEARCH REPORT
Information on patent family members

International application No.

PCT/CN2018/113732

Patent document cited in search report			Publication date (day/month/year)	Patent family member(s)	Publication date (day/month/year)
CN	107797904	A	13 March 2018	None	
CN	108196904	A	22 June 2018	None	
CN	104424092	A	18 March 2015	None	
CN	106897209	A	27 June 2017	None	
CN	105573755	A	11 May 2016	None	

国际检索报告

国际申请号

PCT/CN2018/113732

A. 主题的分类

G06F 11/34 (2006.01) i

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

B. 检索领域

检索的最低限度文献 (标明分类系统和分类号)

G06F

包含在检索领域中的除最低限度文献以外的检索文献

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))

CNPAT、CNKI、WPI、EP0D0C: 启动、加载、渲染、时长、时间点、监听、start、load、render、time、duration、point

C. 相关文件

类 型*	引用文件, 必要时, 指明相关段落	相关的权利要求
X	CN 107797904 A (福建天晴数码有限公司) 2018年 3月 13日 (2018 - 03 - 13) 说明书第[0005]-[0015]段	1-10
A	CN 108196904 A (广州品唯软件有限公司) 2018年 6月 22日 (2018 - 06 - 22) 全文	1-10
A	CN 104424092 A (腾讯科技深圳有限公司) 2015年 3月 18日 (2015 - 03 - 18) 全文	1-10
A	CN 106897209 A (北京奇虎科技有限公司 等) 2017年 6月 27日 (2017 - 06 - 27) 全文	1-10
A	CN 105573755 A (北京奇虎科技有限公司 等) 2016年 5月 11日 (2016 - 05 - 11) 全文	1-10

☐ 其余文件在C栏的续页中列出。☒ 见同族专利附件。

* 引用文件的具体类型:

“A” 认为不特别相关的表示了现有技术一般状态的文件

“E” 在国际申请日的当天或之后公布的在先申请或专利

“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)

“O” 涉及口头公开、使用、展览或其他方式公开的文件

“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

“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

“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

“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

“&” 同族专利的文件

国际检索实际完成的日期

2019年 5月 21日

国际检索报告邮寄日期

2019年 6月 19日

ISA/CN的名称和邮寄地址

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

传真号 (86-10)62019451

受权官员

赵晓敏

电话号码 86-(10)-53961332

国际检索报告
关于同族专利的信息

国际申请号

PCT/CN2018/113732

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	107797904	A	2018年 3月 13日	无	
CN	108196904	A	2018年 6月 22日	无	
CN	104424092	A	2015年 3月 18日	无	
CN	106897209	A	2017年 6月 27日	无	
CN	105573755	A	2016年 5月 11日	无	