

图像获取方法、装置、服务器及存储介质

本申请要求于 2019 年 5 月 29 日提交中国专利局、申请号为 201910468135.3、申请名称为“图像获取方法、装置、服务器及存储介质”的中国专利申请的优先权。

技术领域

5 本申请涉及互联网技术领域，尤其涉及一种图像获取方法、装置、服务器及存储介质。

发明背景

云游戏是一种在线游戏技术，在云游戏场景中，游戏运行在云端服务器，由云端服务器将游戏场景渲染为游戏画面后实时传输到用户终端，以供用户终端的用户
10 针对游戏画面输入相应操作。由此可见，云游戏可实现在图形处理与数据运算能力相对有限的用户终端上运行高品质游戏。

由于上述优点，云游戏成为近年来游戏领域中的研究热点。在对云游戏的研究中涉及的关键问题之一是云端服务器如何获取到正在运行游戏的游戏画面，目前解决该问题常用的方法是基于虚拟机获取画面，具体地：为每个用户终端布置一个虚拟机，虚拟机对游戏场景渲染得到游戏画面，云端服务器直接截取虚拟机的桌面即可获取到游戏画面。但是该方法需要大量部署虚拟机，这样会占用大量的服务器资源，从而导致服务器资源的浪费。

发明内容

本申请实施例提供了一种图像获取方法、装置、服务器及存储介质，服务器截
20 取目标应用程序运行过程中产生的窗口图像数据，并通过对所述窗口图像数据进行合成处理，便可得到与目标应用程序对应的待显示的用户界面图像，与现有技术中通过虚拟机获取用户界面图像的方式相比，节省了布置虚拟机的资源开销。

一方面，本申请实施例提供了一种图像获取方法，所述方法应用于服务器，包括：

25 从应用进程集合中获取用户标识对应的目标应用进程；

当检测到所述目标应用进程中的图像渲染函数被调用时，调用数据截取模块截取当前生成的多个窗口图像数据；

对所述多个窗口图像数据进行图像合成处理，得到待显示的用户界面图像；

将携带所述用户界面图像的通知消息发送给所述用户标识对应的用户终端，以

5 使所述用户终端在用户界面上显示所述用户界面图像。

另一方面，本申请实施例提供了一种图像获取装置，所述装置配置于服务器中，包括：

获取单元，用于从应用进程集合中获取用户标识对应的目标应用进程；

截取单元，用于当检测到所述目标应用进程中的图像渲染函数被调用时，调用
10 数据截取模块截取当前生成的多个窗口图像数据；

第一处理单元，用于对所述多个窗口图像数据进行图像合成处理，得到待显示的用户界面图像；

发送单元，用于将携带所述用户界面图像的通知消息发送给所述用户标识对应的用户终端，以使所述用户终端在用户界面上显示所述用户界面图像。

15 再一方面，本申请实施例提供了一种服务器，所述服务器包括：

处理器，适于实现一条或多条指令；以及，

计算机存储介质，所述计算机存储介质存储有一条或多条指令，所述一条或多条指令适于由所述处理器加载并执行如上所述的图像获取方法。

再一方面，本申请实施例提供了一种计算机存储介质，所述计算机存储介质存储有一条或多条指令，所述一条或多条指令适于由处理器加载并执行如上所述的图像获取方法。
20

附图简要说明

为了更清楚地说明本申请实施例或现有技术中的技术方案，下面将对实施例或现有技术描述中所需要使用的附图作简单地介绍，显而易见地，下面描述中的附图仅仅是本申请的一些实施例，对于本领域技术人员来讲，在不付出创造性劳动性的前提下，还可以根据这些附图获得其他的附图。
25

- 图 1 是一种图像获取的架构图示例；
- 图 2 是本申请实施例提供的一种图像获取的架构图；
- 图 3a 是本申请实施例提供的一种云端服务器的桌面视图；
- 图 3b 是本申请实施例提供的一种用户界面图像的示意图；
- 5 图 4 是本申请实施例提供的一种的图像获取方法的流程示意图；
- 图 5a 是本申请实施例提供的一种获取窗口图像数据的示意图；
- 图 5b 是本申请实施例提供的一种窗口图像数据的示意图；
- 图 5c 是本申请实施例提供的另一种窗口图像数据的示意图；
- 图 5d 是本申请实施例提供的又一种窗口图像数据的示意图；
- 10 图 6a 是本申请实施例提供的另一种图像获取方法的流程示意图；
- 图 6b 是本申请实施例提供的又一种图像获取方法的流程示意图；
- 图 6c 是本申请实施例提供的另一种图像获取方法的流程示意图；
- 图 7a 是本申请实施例提供的一种图像多叉树的示意图；
- 图 7b 是本申请实施例提供的一种图像二叉树的示意图；
- 15 图 8 是本申请实施例提供的一种数据截取模块的示意图；
- 图 9 是本申请实施例提供的一种图像合成处理的示意图；
- 图 10a 是本申请实施例提供的一种应用服务器的桌面视图；
- 图 10b 是本申请实施例提供的一种用户界面图像的示意图；
- 图 11 是本申请实施例提供的一种图像获取装置的结构示意图；
- 20 图 12 是本申请实施例提供的一种服务器的结构示意图。

实施方式

下面将结合本申请实施例中的附图，对本申请实施例中的技术方案进行清楚、完整地描述。

- 在用户终端从服务器获取图像的实现方式中，一种图像获取的架构图可以如图
- 25 1 所示。在图 1 中 101 表示应用服务器，101 中可运行多个应用进程，102 表示多个应用进程中与用户标识对应的目标应用进程，103 表示以所述用户标识登录了目标

应用程序的用户终端，104 表示与用户终端 103 对应的虚拟机。应当理解的，为了保证用户终端中的目标应用程序正常运行，应用服务器需要为目标应用程序的运行提供服务支持，或者说应用服务器用于执行为实现目标应用程序的应用功能提供服务支持的应用进程集合，其中，所述目标应用程序可以指任意的应用程序，比如游戏
5 应用程序、即时聊天应用程序以及购物应用程序等。

应用服务器为目标应用程序提供服务支持是基于与目标应用程序有关的服务程序实现的。具体地，应用服务器可通过执行服务程序来支持目标应用程序实现应用功能，在计算机领域中，应用服务器运行服务程序的活动称为应用进程。如果有多个用户终端中同时运行目标应用程序，为了保证各个用户终端中的目标应用程序的
10 运行互不干扰，应用服务器为每一个用户创建一个应用进程来支持各自的目标应用程序的运行。也即应用服务器中可以执行有为实现目标应用程序的应用功能提供服务支持的应用进程集合，应用进程集合中包括多个应用进程，每个应用进程对应一个用户标识。本申请实施例所述的目标应用进程可以指任意一个用户标识对应的应用进程。

15 例如，在云游戏应用场景中，假设目标应用程序为游戏应用程序，如果用户 A 和用户 B 通过不同的用户终端同时启动了游戏应用程序，应用服务器检测到用户 A 和用户 B 对游戏应用程序的操作指令后，为用户 A 和用户 B 分别创建应用进程 A 和应用进程 B 来支持两个用户终端的玩家体验游戏应用程序。

图 1 所示的图像获取的架构图中，如果应用服务器 101 接收到用户终端 103 发
20 送的针对目标应用程序的操作指令时，应用服务器 101 通知与用户终端 103 中用户标识对应的虚拟机 104 运行目标应用进程 102，其他的虚拟机分别运行应用进程 1 和应用进程 2。应用服务器 101 通过在虚拟机 104 上运行所述目标应用程序得到与用户标识对应的用户界面图像，可选的，通过虚拟机得到与用户标识对应的用户界面图像的方式可以为：虚拟机 104 调用目标应用进程 102 的图像渲染函数进行渲染得
25 到渲染后的窗口图像数据；然后虚拟机 104 对所述窗口图像数据和其他参数进行合成处理，得到待显示的图像；在虚拟机 104 的用户界面中显示所述待显示的图像；

应用服务器 101 截取虚拟机 104 当前的用户界面，将截取到的图像作为获取到与用户标识对应的用户界面图像，并将所述用户界面图像发送给用户终端 103。这样，用户标识所标识的用户便可通过用户终端 103 查看到与目标应用进程 102 对应的用户界面图像。

5 在一个实施例中，图像获取的架构图还可以如图 2 所示，在图 2 所示的图像获取的架构图中，201 表示应用服务器，所述应用服务器 201 中运行有为目标应用程序的应用功能提供服务支持的应用进程集合；202 表示应用进程集合中与用户标识对应的目标应用进程；203 表示以所述用户标识登录了目标应用程序的用户终端，也即所述用户标识对应的用户终端。本申请实施例中，应用进程集合中的各个应用进
10 程中还配置有数据截取模块，例如在目标应用进程中配置有数据截取模块 2021，在应用进程 1 中配置的数据截取模块，以及在应用进程 2 中配置的数据截取模块。

 在一个实施例中，在目标应用进程 202 的运行过程中，应用服务器 201 可以调用目标应用进程 202 中的数据截取模块 2021 截取窗口图像数据，所述窗口图像数据是在执行所述目标应用进程 202 中的图像渲染函数进行渲染时得到的；再对窗口图
15 像数据进行图像合成处理，得到待显示的用户界面图像。可选的，应用服务器 201 中可配置有画面合成器 204，应用服务器 201 可以将调用数据截取模块截取到窗口图像数据发送给画面合成器 204，以调用画面合成器 204 对窗口图像数据进行图像合成处理得到待显示用户界面图像。在得到用户界面图像后，应用服务器 201 将携带用户界面图像的通知消息发送给用户终端 203，以便于用户终端 203 在用户界面上显示
20 所述用户界面图像。

 通过比较图 1 和图 2 所示的图像获取的架构图发现：在图 1 中需要为每个用户标识部署对应的虚拟机，以使得每个用户标识对应的应用进程在相应的虚拟机中运行。当大量部署虚拟机时需要消耗较多的资源开销。此外，由于虚拟机中的显卡不能较好的支持多媒体，在部署虚拟机时需要在应用服务器 101 中单独配置一个高性能显卡给虚拟机使用，如图 1 中 1011 所示。但是，由于该高性能显卡 1011 只用来
25 支持虚拟机中的多媒体业务，在其他情况下显卡处于闲置状态，导致显卡使用率不

高而造成成本浪费。

另外，为了实现图 1 中应用服务器 101、虚拟机 104 以及目标应用进程 102 三者之间的通信，还需要在虚拟机 104 中部署相应的服务模块，如图 1 中 1041 所示。一般情况下，通过用户权限控制用户通过服务模块 1041 对虚拟机的操作权限，这样，
5 容易存在某些恶意用户突破权限控制并且通过服务模块 1041 进行信息窃取以及破坏，导致图像获取的安全性较差。

此外，在图 1 所示的架构图中各个应用进程分别运行在相应的虚拟机上而不是运行在应用服务器上，导致应用进程的运行效率低。

相比于图 1，图 2 所示的图像获取的架构图中，不进行虚拟机的部署可节省资源消耗；并且，应用服务器 201 只向用户终端 203 展示与目标应用进程相关的用户
10 界面图像，提高了应用的安全性；此外，图 2 所示的图像获取的架构图中应用进程直接运行在应用服务器 201 上，而不是运行在虚拟机上，提高应用进程的运行效率。综上所述，下面将具体介绍基于图 2 所示的图像获取架构的本申请实施例。

15 基于图 2 所述的图像获取的架构图，本申请实施例提供了一种图像获取方法，所述图像获取方法可以应用在教育服务器中，所述应用服务器用于执行为实现目标应用程度的应用功能提供服务支持的应用进程集合，例如图 1 中的应用服务器 101 或者图 2 中的应用服务器 201。采用本申请实施例所述的图像获取方法可以获取到某个用户标识对应的用户界面图像，并将得到的用户界面图像携带在通知消息中发
20 送给所述用户标识对应的用户终端，以指示用户终端在用户界面显示所述用户界面图像。这样，用户标识所标识的用户可通过用户界面图像对目标应用程序进行操作。与现有技术通过虚拟机获取待显示的用户界面图像的方式相比，本申请实施例节省了在教育服务器中布置虚拟机的资源开销。

在一个实施例中，本申请实施例所述的图像获取方法可以应用在云游戏的应用
25 场景中，所谓云游戏是指：多个用户标识对应的游戏运行在云端服务器中，由云端服务器将各个游戏中的游戏场景渲染为游戏动画后，实时传输到各个用户标识对应

的用户终端，以供各个用户标识所标识的用户针对游戏画面输入相应的操作。

本申请实施例应用于云游戏的应用场景时，所述应用服务器相当于云端服务器，所述用户界面图像相当于游戏画面。例如，图 3a 为本申请实施例提供的一种云端服务器的桌面视图，例如，该云端服务器为图 1 中的应用服务器 101 或者图 2 中的应用服务器 201，在图 3a 中 300 表示云端服务器的桌面。在图 3a 中，云端服务器中同时运行了多个支持游戏运行的应用进程，采用本申请实施例所述的图像获取方法，云端服务器可以获取到多个应用进程对应的游戏画面，在云端服务器的桌面视图中可同时展示每个应用进程对应的游戏画面例如 301、302 以及 303。由于一个应用进程对应一个用户标识，所述每个应用进行对应的游戏画面就是指每个用户标识对应的游戏画面。例如，301 表示与用户标识 A 对应的游戏画面，302 表示与用户标识 B 对应的游戏画面，303 表示与用户标识 C 对应的游戏画面。

采用本申请实施例所述的图像获取方法，可以实现将各个用户标识对应的游戏画面，分别传输给对应的用户终端以进行显示，对于某个用户标识所标识的用户来说，在用户终端的用户界面中看到的游戏画面是云端服务器中与该用户标识对应的游戏画面，不会显示其他用户标识对应的游戏画面。也即各个用户对应的游戏互相之间不会影响。例如，如图 3b 为多个用户标识中用户标识 A 对应的用户终端的用户界面图像，例如，该用户终端为图 1 中的用户终端 103 和图 2 中的用户终端 203，可见，用户标识 A 对应的用户终端的用户界面中只展示了与用户标识 A 对应的游戏画面 301。

20

基于上述描述，本申请实施例提供了一种图像获取方法的流程示意图，如图 4 所示。图 4 所述的图像获取方法可以由服务器执行，具体地可由服务器的处理器执行。例如，应用于图 1 中所示的应用服务器 101 或者图 2 所示的应用服务器 201 中，该应用服务器用于执行为实现目标应用程序的应用功能提供服务支持的应用进程集合。图 4 所述的图像获取方法可包括如下步骤：

25

S401、从应用进程集合中获取用户标识对应的目标应用进程。

其中，所述用户标识是用于表示用户的身份，所述用户标识可以包括用户的身份证、用户的手机号以及用户的登录账号中的任意一种或多种。

在一个实施例中，所述获取用户标识对应的目标应用进程可以是应用服务器在接收到用户终端发送的用户操作请求后执行的。其中，所述用户终端是指基于用户标识完成登录了的终端设备，所述终端设备可包括手机、平板、台式电脑等任何具备视频解压能力的设备；所述用户操作请求可以包括针对目标应用程序的登录操作、窗口切换操作以及登出操作等操作中的任意一种或多种。

例如，如果用户标识所指示的用户初次在用户终端中登录到目标应用程序时，此时所述用户终端接收的用户发送的用户操作请求可以指登录操作；如果用户终端中基于用户标识已登录了目标应用程序，此时用户操作消息可以指登出操作，或者对目标应用程序包括的各个窗口的窗口切换操作，或者对目标应用程序的某个应用功能的运行请求等。

S402，当检测到目标应用进程中的图像渲染函数被调用时，调用数据截取模块截取当前生成的多个窗口图像数据。

本步骤中，窗口图像数据是在运行所述目标应用进程的过程中，执行所述目标应用进程中的图像渲染函数时得到的。可选的，调用数据截取模块截取当前生成的窗口图像数据的实施方式，如图 5a 所示。图 5a 是本申请实施例提供的一种获取窗口图像数据的示意图，该获取方法可以由服务器执行，例如，图 1 中的应用服务器 101 或者图 2 中的应用服务器 201，包括如下步骤：

S4021，执行目标应用进程中的图像渲染函数；
S4022，当检测到目标应用进程中的图像渲染函数被调用时，拦截图像渲染函数；
S4023，调用数据截取模块，获取图像渲染函数针对各个窗口生成的窗口图像数据。

S4024，图像渲染函数执行结束。
其中，图像渲染指将三维的光能传递处理转换为一个二维图像的过程，图像渲染函数是指可以实现图像渲染功能的函数。

在图像渲染函数中常用的图像渲染方式可以包括：开放图像库（Open Graphics Library, OpenGL）、图形设备接口（Graphics Device Interface, GDI）和编程接口 DirectX 三种。其中，OpenGL 是一个底层 3D 图形库，只提供渲染功能，可以在不同平台之间进行移植；DirectX 主要为 GPU 设计需要硬件支持，并且 DirectX 更适合对游戏应用
5 用的图像渲染；GDI 为广谱设计的，也可以应用在对游戏应用的图像渲染中，但是效果不及 DirectX。

在一个实施例中，针对不同窗口进行渲染时，目标应用进程的图像渲染函数中对应各个窗口的渲染方式可不同，例如，对目标应用程序的第一窗口进行渲染时，图像渲染函数中包括的渲染方式可以为 GDI；在对目标应用程序的第二窗口进行
10 渲染时，图像渲染函数中包括的渲染方式可以为 OpenGL。

在一个实施例中，参考图 5b-图 5d 为本申请实施例提供的一种窗口图像数据的示意图，假设目标应用程序为游戏应用，所述游戏应用可包括三个窗口，分别为“主页”窗口、“商店”窗口和“视频”窗口，每个窗口对应一个窗口图像数据，具体为，“主页”窗口图像数据如图 5b 所示，“商店”窗口图像数据如图 5c 所示，“视频”
15 窗口图像数据如图 5d 所示。

S403、对多个窗口图像数据进行图像合成处理，得到待显示的用户界面图像。

在一个实施例中，所述图像合成处理可以指，按照各个窗口图像数据的合成顺序，将各个窗口图像数据依次叠加。所述合成顺序可以指各个窗口图像数据在图像合成处理过程中被处理的顺序，合成顺序越靠前的窗口图像数据，在图像合成处理
20 时越先被处理。所述合成顺序可以是依据各个窗口图像数据对应的窗口层级标识确定的。所述窗口层级标识用于表示窗口的层级高低，窗口层级标识所述指示的层级越高，表明窗口图像数据的合成顺序越靠后。

例如，假设在图 5b-图 5d 中，三个窗口图像数据的窗口层级标识由高到低的顺序依次为“视频”，“商店”，“主页”，根据上述窗口层级标识越高的窗口图像数据的合成顺序越靠后，则图 5b-图 5d 中三个窗口图像数据的合成顺序可以为“主页”、
25 “商店”和“视频”；如果按照合成顺序将三个窗口数据进行合成处理，得到的待显

示的用户界面图像可如图 3b 所示。可选的，应用服务器可以调用画面合成器将各个窗口图像数据进行合成处理。

S404、将携带用户界面图像的通知消息发送给用户标识对应的用户终端，以使用户终端在用户界面上显示用户界面图像。

5 其中，所述通知消息用于指示用户终端在用户界面显示用户界面图像，所述用户终端在用户界面中显示用户界面图像的示意图可图 3b 所示。

与现有技术中通过部署虚拟机获取用户界面图像的方式相比，本申请实施例中应用服务器拦截执行目标应用程序过程中，调用渲染函数得到的窗口图像数据，然后对窗口图像数据进行图像合成处理得到用户界面图像，可节省应用服务器的资源
10 开销。

请参见图 6a，是本申请实施例提供的另一种图像获取方法的流程示意图。图 6a 所述的图像获取方法可以由服务器执行，具体地可由服务器的处理器执行。例如，可以应用于图 1 所示的应用服务器 101 或者图 2 所示的应用服务器 201 中，所述应用服务器用于执行为实现目标应用程序的应用功能提供服务支持的应用进程集合。
15 图 6a 所述的图像获取方法可包括如下步骤：

S601，从应用进程集合中获取用户标识对应的目标应用进程。

此步骤与上述步骤 401 相同，在此不再赘述。

S602，当检测到目标应用进程中的图像渲染函数被调用时，确定图像渲染函数
20 中包括的渲染方式。

窗口图像数据的数量为多个，各个窗口图像数据是执行目标应用程序的一个或多个图像渲染函数得到的，各个图像渲染函数中包括的渲染方式可以为 OpenGL、GDI 和 DirectX 中的任意一种。

S603，根据渲染方式，确定数据截取模块中与渲染方式对应的数据截取子模块。

25 所述数据截取模块可包括三个截取子模块，所述三个截取子模块的分别于上述三种渲染方式相对应，具体地，数据截取模块包括的是三个截取子模块可以为

OpenGL 截取子模块、GDI 截取子模块以及 DirectX 截取子模块。可选的，各个截取子模块与渲染方式的对应关系可以为：OpenGL 截取子模块对应的渲染方式为 OpenGL、GDI 截取子模块对应的渲染方式为 GDI 以及 DirectX 截取子模块对应的渲染方式为 DirectX。应当理解的，调用与渲染方式相对应的截取子模块来获取窗口
5 图像数据，可以保证截取到的窗口图像数据的准确性。

在一个实施例中，本申请实施例提供的的数据截取模块的示意图可如图 8 所示，图 8 所示的数据截取模块可包括 GDI 截取子模块、DirectX 截取子模块和 OpenGL 截取子模块。

S604，调用数据截取子模块，截取在执行图像渲染函数时得到的多个窗口图像
10 数据。

在执行步骤 S602-S604 的同时，还可以执行如下步骤 S605：

S605，获取为目标应用进程设置的画面显示参数，并根据画面显示参数，确定各个窗口图像数据的窗口层级标识。

其中，所述画面显示参数可以包括各个窗口图像数据对应的窗口是否被最小化、
15 最大化、以及各个窗口之间的排布关系中的任意一种或多种，比如所述各个窗口之间的排布关系可以为，窗口 A 放在窗口 B 之前；窗口 C 放在窗口 B 之后等。

在一个实施例中，所述画面显示参数可以是存储于沙箱中的，在本申请实施例中所述的沙箱是一种机制，可以使得应用服务器运行的多个应用进程互相隔离。

在一个实施例中，沙箱中存储的为目标应用进程设置的画面显示参数，可以是
20 应用服务器为目标应用进程默认设置的参数，例如，假设目标应用程序为游戏应用，应用服务器为用于实现游戏应用的游戏功能提供支持的目标应用进程设置的参数可以包括：“主页”窗口数据对应的窗口放在最前面显示，“商店”窗口图像数据对应的窗口放在“主页”窗口图像数据对应窗口之后，“视频”窗口图像数据对应的窗口放在“商店”图像数据对应的窗口之后；此时画面显示参数可以为：“主页”窗口图像数据放在最前面显示，“商店”窗口图像数据放在“主页”窗口图像数据之后，“视
25 频”窗口图像数据放在“商店”窗口图像数据之后。

在其他实施例中，沙箱中存储的为目标应用进程设置的画面显示参数，可以是根据用户终端发送的用户操作请求和应用服务器为目标应用进程默认设置的参数确定的。例如，假设目标应用程序为游戏，为用于实现游戏应用的游戏功能提供服务支持的目标应用进程设置的参数可以包括：“主页”窗口图像数据对应的放在最前面显示，“商店”窗口图像数据对应的窗口放在“主页”窗口图像数据对应的窗口之后，“视频”窗口图像数据对应的窗口放在“商店”窗口图像数据对应的窗口之后；检测到用户操作请求中包括从“主页”窗口图像数据对应的窗口切换到“视频”窗口图像数据对应的窗口，此时根据用户操作请求和应用服务器为应用程序默认设置的参数确定的画面显示参数可以包括：“视频”窗口图像数据对应的窗口放在最前面显示，“商店”窗口图像数据对应的窗口放在“主页”窗口图像数据对应的窗口之后，“主页”窗口图像数据对应的窗口放在“视频”窗口图像数据对应的窗口之后。

在获取到为目标应用进程设置的画面显示参数之后，可以根据画面显示参数确定各个窗口图像数据的窗口层级标识，所述窗口层级标识用于唯一表示对应的窗口图像数据，并且所述窗口层级标识指示了窗口图像数据之间的层级关系。

所述窗口层级标识可以以一个或多个数字表示，例如 1, 2, 3, 或者 (1, 2), (2, 1)、(3,1) 等；或者所述窗口层级表示也可以以其他形式表示。在本申请实施例中，如果用数字表示窗口层级标识，如果用于代表各个窗口层级标识的数字的个数不相同，数字个数越多的窗口层级标识越高，例如 (1, 2) 代表的窗口层级标识高于 1 代表的窗口层级标识；如果用于代表各个窗口层级标识的数字个数都为 N 个，N 为大于 1 的任意正整数，且前 N-1 个数字相同，而第 N 个数字不相同，则各个数字的之和越小的窗口层级标识越高，例如 (1, 2) 代表的窗口层级标识高于 (1, 5) 代表的窗口层级标识。其中，窗口层级标识 (1, n) (n 为大于等于 1 的任意正整数) 对应的窗口是窗口层级标识 (1) 对应的窗口的子窗口；窗口层级标识 (1, n, m) 对应的窗口是窗口层级标识 (1, n) 对应的窗口的子窗口。基于上述描述可知，子窗口对应的窗口层级标识高于其父窗口对应的窗口层级标识。

S606，根据窗口层级标识，对多个窗口图像数据进行图像合成处理，得到待显

示的用户界面图像。

由于窗口层级标识表征了各个窗口之间的层级关系，那么按照层级关系从高到低的顺序，对多个窗口图像数据进行图像合成处理。

S607，将携带用户界面图像的通知消息发送给用户标识对应的用户终端，以使
5 用户终端在用户界面上显示用户界面图像。

此步骤与上述步骤 S404 相同，在此不再赘述。

请参见图 6b，是本申请实施例提供的又一种图像获取方法的流程示意图。图 6b
所述的图像获取方法可以由服务器执行，具体地可由服务器的处理器执行。例如，
10 可以应用于图 1 所示的应用服务器 101 或者图 2 所示的应用服务器 201 中，所述应用服务器用于执行为实现目标应用程序的应用功能提供服务支持的应用进程集合，
图 6b 所述的图像获取方法可包括如下步骤：

S701，从应用进程集合中获取用户标识对应的目标应用进程。

此步骤与上述步骤 401 相同，在此不再赘述。

15 S702，获取为目标应用进程设置的画面显示参数，并根据画面显示参数，确定各个窗口图像数据的窗口层级标识。

此步骤与上述步骤 605 相同，在此不再赘述。

S703，根据各个窗口图像数据的窗口层级标识，生成图像二叉树。

为了提高便利性，本申请实施例中根据各个窗口图像数据的窗口层级标识生成
20 一个图像二叉树。所述图像二叉树中可包括多个二叉树节点，每个二叉树节点存储了窗口层级标识。

在一个实施例中，所述根据所述各个窗口图像数据的窗层级标识生成图像二叉树，可包括：

S7031，根据各个窗口图像数据的窗口层级标识，构建图像多叉树。

25 其中，图像多叉树包括 N 个层级，每个层级中包括 M 个节点，其中，N 和 M 都是大于等于 1 的正整数，所述图像多叉树中相邻两个层级中包括为父子关系的父

节点和子节点，所述图像多叉树的同一层级中具有同一父节点的各个节点互为兄弟节点；

S7032，将图像多叉树转换为图像二叉树。

其中，是将所述图像多叉树的各个层级中的第一个节点转换为图像二叉树的左节点，将与所述第一个节点互为兄弟节点关系的其他节点转换为图像二叉树的右节点。

例如，假设根据各个窗口图像数据的窗口层级标识构建的图像多叉树如图 7a 所示，假设图像二叉树中包括 3 个层级，如 701-703 所示，各个层级的节点中的数字表示该节点对应的窗口图像数据的窗口层级标识，由图可见，在 701 层级中包一个节点，701 层级包括的节点可以称为根节点，本申请实施例中所称根节点为用户终端的桌面图像数据。所述用户终端的桌面图像数据可以是应用服务器接收到用户终端在接收到的用户输入的桌面图像数据的选择操作后发送的；在 702 层级中包括三个节点，在 703 中包括三个节点。

其中，702 层级中的窗口层级标识 (1, 1) 和 (1, 2) 对应的窗口是窗口层级标识 1 对应的窗口的子窗口，因此窗口层级标识 (1, 1) 和 (1, 2) 代表的子节点的父节点都是窗口层级标识 1 代表的节点，所以窗口层级标识 (1, 1) 和 (1, 2) 代表的节点互为兄弟节点。同理可知的，窗口层级标识 1, 2, 3，代表的节点互为兄弟节点。

将图 7a 所示的图像多叉树转化为图像二叉树是指：将图像多叉树中 701-703 各个层级中的第一节点转化为图像二叉树的左节点，例如，在 701 层级中第一个节点为窗口层级标识 0 代表的节点，将此节点作为图像二叉树的左节点，701 层级中不包括其他节点，则继续对 702 层级进行转换；在 702 层级中，第一个节点为窗口层级标识 1 代表的节点，将此节点继续作为图像二叉树的左节点，702 层级中还包括窗口层级标识 1 代表的节点的两个兄弟节点，分别为窗口层级标识 2 代表的节点和窗口层级标识 3 代表的节点，将这两个节点依次作为图像二叉树的右节点。按照上述过程遍历图像二叉树中所有节点，所有节点遍历结束后便可得到将如图 7b 所示的

图像二叉树。

应当理解的，将窗口层级标识 2 代表的节点和窗口层级标识 3 代表的节点依次作为图像二叉树的右节点可以理解为：窗口层级标识 2 代表的节点作为窗口层级标识 1 代表的节点的右节点，窗口层级标识 3 代表的节点作为窗口层级标识 2 代表的节点的右节点。

S704，按照预设的第一遍历顺序对图像二叉树进行遍历，得到各个窗口图像数据的合成顺序。

其中，所述预设的第一（即二叉树图像合成）遍历顺序可以指前序遍历，所述前序遍历是指：根节点-左节点-右节点。例如，在图 7b 所示的图像二叉树中前序遍历的结果为：0-1- (1, 1) - (1, 2) -2-3- (3, 1)。按照预设的二叉树显示遍历顺序对所述图像二叉树进行遍历，得到窗口图像数据中各个窗口图像数据的合成顺序 0-1- (1, 1) - (1, 2) -2-3- (3, 1)。

S705，按照预设的第二遍历顺序对图像二叉树进行遍历，得到各个窗口图像数据的图像捕获顺序，并按照图像捕获顺序，调用数据截取模块截取得到多个窗口图像数据。

在一实施例中，所述预设的第二（二叉图像获取）遍历顺序可以指后序遍历，所述后序遍历的遍历顺序为：左节点-右节点-根节点。例如，在图 7b 中所示的图像二叉树中后序遍历的结果为：(1, 1) - (1, 2) - (3, 1) -3-2-1-0。按照预设的二叉树图像获取的遍历顺序对所述图像二叉树进行遍历，得到窗口图像数据中各个窗口图像数据的图像捕获顺序即为 (1, 1) - (1, 2) - (3, 1) -3-2-1-0。

在一个实施例中，在获取到窗口图像数据后，可以将窗口图像数据进行图像合成处理，便可得到待显示的用户界面图像。

S706，按照合成顺序，对各个窗口图像数据进行图像合成处理，得到待显示的用户界面图像。

在一个实施例中，所述步骤 706 的实施方式可以为：为每个窗口图像数据分配一块子画布，将相应的窗口图像数据放在对应的子画布上，按照上述的合成顺序将

各个子画布进行组合，得到待显示的用户界面图像。

应当理解的，按照上述合成顺序进行组合，以实现窗口层级标识越高的窗口图像数据越后进行组合，这样一来，得到的用户界面图像中窗口层级标识高的窗口图像数据会覆盖掉窗口层级标识低的窗口图像数据。

5 S707，将携带用户界面图像的通知消息发送给用户标识对应的用户终端。

此步骤同上述步骤 404，在此不再赘述。

在本申请另一个实施例中，图 6c 是本申请实施例提供的另一种图像获取方法的流程图示意图，该图像获取方法可以由服务器执行，具体地可由服务器的处理器执行。

10 例如，可以应用于图 1 所示的应用服务器 101 或者图 2 所示的应用服务器 201 中。
如图 6c 所示，在图 6b 所述的实施例基础之上，在步骤 S702 之后，还可以执行：

S708，根据各个窗口图像数据的窗口层级标识，生成消息响应二叉树。

其中，消息响应二叉树中包括多个消息节点，每个消息节点中记录有相应的窗口图像数据对应的窗口位置信息和窗口尺寸信息。其中，为了简化操作，在根据各个窗口图像数据的窗口层级标识生成消息响应二叉树时，可以采用与步骤 S703 相同的方法，这样一来，消息响应二叉树和图像二叉树具有相同的结构。例如，根据各个窗口图像数据的窗口层级标识生成的消息二叉树也可以如图 7b 所示。

15

图像二叉树每个节点存储了窗口图像数据的窗口层级标识，与图像二叉树不同的是，消息响应二叉树中每个节点不仅存储了窗口图像数据的窗口层级标识，还存储有各个窗口图像数据对应的窗口位置信息和窗口尺寸信息。其中，所述窗口位置信息可以包括窗口的坐标，所述窗口的坐标可以是以像素坐标形式表示的例如（3 像素，6 像素），也可以是以世界坐标形式表示的例如（10，15）；所述窗口尺寸信息可以包括窗口的宽度和高度等信息。

20

S709，基于消息响应二叉树，确定各个窗口图像数据的显示位置信息。

25 其中，所述显示位置信息可以包括窗口图像数据的窗口位置信息和窗口尺寸信息中的任意一种或两种。

S710, 根据显示位置信息和合成顺序, 对各个窗口图像数据进行合成处理, 得到待显示的用户界面图像。

在对各个窗口图像数据进行合成时, 在知道了各个窗口图像数据之后, 还需要知道各个窗口图像数分别应用位于什么位置, 例如, 如图 5b-图 5d 所示, 在已知了
5 “主页”窗口图像数据以及“商店”窗口图像数据之后, 还需要两个窗口图像数据应该以多大尺寸, 在什么位置进行组合, 才能得到如图 3b 所示的用户界面图像。因此, 在对各个窗口图像数据进行合成时, 可以通过消息响应二叉树获取各个窗口图像数据的显示位置信息。

基于上述描述, 本申请实施例提供了如图 9 所示的图像合成处理的示意图, 在
10 图 9 中数据获取模块用于获取目标应用进程中的各个窗口图像数据, 应用服务器调用画面合成器根据图像二叉树和消息响应二叉树对各个窗口图像数据进行图像合成处理。

在一个实施例中, 步骤 S709 中, 基于消息响应二叉树, 确定各个窗口图像数据的显示位置信息的实施方式可包括:

15 针对各个窗口图像数据中的第一窗口图像数据(所述第一窗口图像数据为各个窗口图像数据中的任意一个窗口图像数据), 当检测到当前正在对第一窗口图像数据进行图像合成处理, 应用服务器可以根据图像二叉树获取第一窗口图像数据的窗口层级标识; 再从消息响应二叉树中查找与第一窗口数据的窗口层级标识对应的窗口位置信息和窗口尺寸信息, 将查找到的窗口位置信息和窗口尺寸信息确定为第一窗
20 口图像数据的显示位置信息。

在其他实施例中, 消息响应二叉树除了应用于图像合成处理过程中, 还可以用于查找与用户操作消息对应的目标窗口图像数据, 以便于应用服务器响应于用户操作消息对目标窗口图像数据进行处理。其中, 所述用户操作消息是用户通过用户终端输入的与用户界面图像相关的操作。具体地, 在图 6c 中还包括如下步骤:

25 S711, 接收用户终端发送的用户操作消息, 从用户操作消息中获取操作位置信息。

其中，所述用户操作消息中包括的操作位置信息是指所述用户输入操作的位置处的位置信息，比如用户点击、双击或长按用户界面图像中某个位置，将此位置的位置信息确定为操作位置信息。

S712，按照预设的第三遍历顺序遍历消息响应二叉树，将操作位置信息满足预设位置条件的消息节点确定为目标消息节点。

具体而言，目标消息节点为在消息响应二叉树中第一个查找到的满足预设位置条件的消息节点，其中，所述操作位置信息满足预设位置条件是指所述操作位置信息在所述目标消息节点记录的窗口位置信息和窗口尺寸信息所限定的范围内；

S713，将目标消息节点对应的窗口图像数据作为目标窗口图像数据，响应于用户操作消息，对目标窗口图像数据所对应的窗口进行处理。

由于应用服务器中运行多个用户标识对应的应用进程，应用服务器中显示多个用户标识对应的用户界面图像，不同用户标识对应的用户界面图像可能重叠在一起如图 10a 所示，在图 10a 中 A 和 B 表示不同用户标识对应的用户界面图像；同一用户标识对应的用户界面图像中各个窗口图像数据之间也可能有重叠如图 10b 所示，在图 10b 中 A 和 B 表示同一用户标识对应的用户界面图像中的两个窗口图像数据。

不同用户标识对应不同的应用进程，不同应用进程对应的画面显示参数不同，由于图像二叉树和消息二叉树是根据画面显示参数确定的窗口层级标识生成的，所以不同应用进程之间的图像二叉树和消息响应二叉树也不相同，因此可以理解为不同用户标识对应的消息响应二叉树不同。

针对图 10a 的情况，如果用户操作消息中包括的操作位置信息指示的位置位于 A 和 B 的重叠区域，应用服务器需要知道用户输入的用户操作消息是针对哪个用户界面图像的。可选的，可以根据不同用户标识对应的消息响应二叉树来确定当前接收到的用户操作消息是针对哪个用户界面图像的。

对于图 10b 的情况，应用服务器需要知道用户操作消息是针对用户界面图像中哪个窗口图像数据的。可选的，可以按照预设的第三（即二叉树窗口查找）遍历顺序，遍历消息响应二叉树，确定出目标消息节点，进而确定出目标节点对应的目标

窗口图像数据。

所述预设的第三遍历顺序可以指中序遍历，所述中序遍历是指按照：左节点-根节点-右节点的顺序对二叉树进行遍历。例如，假设消息响应二叉树如图 7b 所示，中序遍历结果为：(1, 1) - (1,2) -1-2- (3,1) - (3) -0。

5

基于上述图像获取方法的描述，本申请实施例还公开了一种图像获取装置，所述图像获取装置可以应用于服务器中，例如，图 1 中的应用服务器 101 或者图 2 中的应用服务器 201，所述应用服务器用于执行为实现目标应用程序的应用功能提供服务支持的应用进程集合，所述图像获取装置可以执行图 4 和图 6a-图 6c 所示的方法。请参见图 11，所述图像获取装置 1100 可运行如下单元：

10

获取单元 1101，用于从应用进程集合中获取用户标识对应的目标应用进程；

截取单元 1102，用于当检测到所述获取单元 1101 获取到的目标应用进程中的图像渲染函数被调用时，调用数据截取模块，截取当前生成的多个窗口图像数据；

第一处理单元 1103，用于对所述截取单元 1102 截取到的多个窗口图像数据进行图像合成处理，得到待显示的用户界面图像；

15

发送单元 1104，用于将携带所述第一处理单元 1103 得到的用户界面图像的通知消息发送给所述用户标识对应的用户终端，以使所述用户终端在用户界面上显示所述用户界面图像。

在一个实施例中，获取单元 1101，还用于获取为所述目标应用进程设置的画面显示参数；

20

装置 1100 还包括：

确定单元 1105，用于根据所述获取单元 1102 获取的画面显示参数，确定各个窗口图像数据的窗口层级标识；

第一处理单元 1103，还用于根据所述确定单元 1105 确定的窗口层级标识，对上述多个窗口图像数据进行图像合成处理。

25

在一个实施例中，装置 1100 还包括：

第一生成单元 1106, 用于根据所述确定单元 1105 确定的各个窗口图像数据的窗口层级标识, 生成图像二叉树;

第一处理单元 1103 还用于, 按照预设的第一遍历顺序, 对所述第一生成单元 1106 生成的图像二叉树进行遍历, 得到各个窗口图像数据的合成顺序; 按照所述合成顺序, 对所述多个窗口图像数据进行图像合成处理。

在一个实施例中, 第一生成单元 1106 用于, 根据所述确定单元 1105 确定的各个窗口图像数据的窗口层级标识, 构建图像多叉树; 将所述图像多叉树转换为所述图像二叉树。

在一个实施例中, 装置 1100 还包括:

10 第一生成单元 1106, 用于根据所述确定单元 1105 确定的各个窗口图像数据的窗口层级标识, 生成图像二叉树;

截取单元 1102 用于, 按照预设的第二遍历顺序, 对所述第一生成单元 1106 生成的图像二叉树进行遍历, 得到各个窗口图像数据的图像捕获顺序; 按照所述图像捕获顺序, 调用所述数据截取模块, 截取得到所述多个窗口图像数据。

15 在一个实施例中, 截取单元 1102 用于, 确定所述图像渲染函数中包括的渲染方式; 根据所述渲染方式, 确定所述数据截取模块中与所述渲染方式对应的数据截取子模块; 调用所述数据截取子模块, 截取在执行所述图像渲染函数时得到的多个窗口图像数据。

在一个实施例中, 装置 1100 还包括:

20 第二生成单元 1107, 用于根据所述确定单元 1105 确定的各个窗口图像数据的窗口层级标识, 生成消息响应二叉树;

第一处理单元 1103 用于, 基于所述第二生成单元 1107 生成的消息响应二叉树, 确定各个窗口图像数据的显示位置信息; 根据所述显示位置信息和所述合成顺序, 对所述多个窗口图像数据进行合成处理。

25 在一个实施例中, 所述消息响应二叉树中包括多个消息节点, 获取单元 1101 还用于, 接收所述用户终端发送的用户操作消息, 从所述用户操作消息中获取操作位

置信息；

装置 1100 还包括：

第二处理单元 1108 用于，按照预设的第三遍历顺序，遍历所述第二生成单元 1107 生成的消息响应二叉树，将所述获取单元 1101 获取的操作位置信息满足预设位置条件的消息节点，确定为目标消息节点；将所述目标消息节点对应的窗口图像数据作为目标窗口图像数据，响应于所述用户操作消息，对所述目标窗口图像数据所对应的窗口进行处理。

根据本申请的另一个实施例，图 11 所示的图像获取装置中的各个单元可以分别或全部合并为一个或若干个另外的单元来构成，或者其中的某个（些）单元还可以再拆分为功能上更小的多个单元来构成，这可以实现同样的操作，而不影响本申请的实施例的技术效果的实现。上述单元是基于逻辑功能划分的，在实际应用中，一个单元的功能也可以由多个单元来实现，或者多个单元的功能由一个单元实现。在本申请的其它实施例中，基于图像获取装置也可以包括其它单元，在实际应用中，这些功能也可以由其它单元协助实现，并且可以由多个单元协作实现。

根据本申请的另一个实施例，可以通过在包括中央处理单元（CPU）、随机存取存储介质（RAM）、只读存储介质（ROM）等处理元件和存储元件的例如计算机的通用计算设备上运行能够执行如图 4 或图 6a-6c 中所示的相应方法所涉及的各步骤的计算机程序（包括程序代码），来构造如图 11 中所示的图像获取装置，以及来实现本申请实施例的图像获取方法。所述计算机程序可以记载于例如计算机可读记录介质上，并通过计算机可读记录介质装载于上述计算设备中，并在其中运行。

基于上述方法实施例以及装置实施例的描述，本申请实施例还提供一种服务器，所述服务器可以为图 1 中所示的应用服务器 101 或者图 2 中所示的应用服务器 201，所述服务器用于执行为实现目标应用程序的应用功能提供服务支持的应用进程集合。请参见图 12，该服务器至少包括处理器 1201、网络接口 1202 以及计算机存储介质 1203。所述网络接口 1202 在连接网络时用于接收或发送数据。

计算机存储介质 1203 可以存储在服务器的存储器中, 所述计算机存储介质 1203 用于存储计算机程序, 所述计算机程序包括程序指令, 所述处理器 1201 用于执行所述计算机存储介质 1203 存储的程序指令。处理器 1201 或称 CPU(Central Processing Unit, 中央处理器)) 是终端的计算核心以及控制核心, 其适于实现一条或多条指令, 具体适于加载并执行一条或多条指令从而实现相应方法流程或相应功能; 在一个实施例中, 本申请实施例所述的处理器 1201 可以用于执行上述图 4、图 6a-6c 中所述的图像获取方法。

本申请实施例还提供了一种计算机存储介质 (Memory), 所述计算机存储介质是终端中的记忆设备, 用于存放程序和数据。可以理解的是, 此处的计算机存储介质既可以包括终端中的内置存储介质, 当然也可以包括终端所支持的扩展存储介质。计算机存储介质提供存储空间, 该存储空间存储了终端的操作系统。并且, 在该存储空间中还存放了适于被处理器 1201 加载并执行的一条或多条指令, 这些指令可以是一个或多个计算机程序 (包括程序代码)。需要说明的是, 此处的计算机存储介质可以是高速 RAM 存储器, 也可以是非不稳定的存储器 (non-volatile memory), 例如至少一个磁盘存储器; 可选的还可以是至少一个位于远离前述处理器的计算机存储介质。

在一个实施例中, 可由处理器 1201 加载并执行计算机存储介质中存放的一条或多条指令, 以实现上述有关图像获取方法实施例中的方法的相应步骤; 具体实现中, 计算机存储介质中的一条或多条指令由处理器 1201 加载并执行上述图 4、图 6a-6c 中所述的图像获取方法。

以上所揭露的仅为本申请较佳实施例而已, 当然不能以此来限定本申请之权利范围, 因此依本申请权利要求所作的等同变化, 仍属本申请所涵盖的范围。

权利要求书

- 1、一种图像获取方法，其特征在于，由服务器执行，包括：
从应用进程集合中获取用户标识对应的目标应用进程；
当检测到所述目标应用进程中的图像渲染函数被调用时，调用数据截取模块，
5 截取当前生成的多个窗口图像数据；
对所述多个窗口图像数据进行图像合成处理，得到待显示的用户界面图像；
将携带所述用户界面图像的通知消息发送给所述用户标识对应的用户终端，以使所述用户终端在用户界面上显示所述用户界面图像。
- 10 2、如权利要求1所述的方法，还包括：
获取为所述目标应用进程设置的画面显示参数；
根据所述画面显示参数，确定各个窗口图像数据的窗口层级标识；
其中，所述对所述多个窗口图像数据进行图像合成处理，得到待显示的用户界面图像，包括：
15 根据所述窗口层级标识，对所述多个窗口图像数据进行图像合成处理。
- 3、如权利要求2所述的方法，还包括：
根据各个窗口图像数据的窗口层级标识，生成图像二叉树；
所述根据所述窗口层级标识，对所述多个窗口图像数据进行图像合成处理，包
20 括：
按照预设的第一遍历顺序，对所述图像二叉树进行遍历，得到各个窗口图像数据的合成顺序；
按照所述合成顺序，对所述多个窗口图像数据进行图像合成处理。
- 25 4、如权利要求3所述的方法，其中，所述根据各个窗口图像数据的窗口层级标

识，生成图像二叉树，包括：

根据各个窗口图像数据的窗口层级标识，构建图像多叉树；

将所述图像多叉树转换为所述图像二叉树。

5 5、如权利要求2所述的方法，还包括：

根据各个窗口图像数据的窗口层级标识，生成图像二叉树；

所述当检测到所述目标应用进程中的图像渲染函数被调用时，调用数据截取模块，截取当前生成的多个窗口图像数据，包括：

按照预设的第二遍历顺序，对所述图像二叉树进行遍历，得到各个窗口图像数据
10 据的图像捕获顺序；

按照所述图像捕获顺序，调用所述数据截取模块，截取得到所述多个窗口图像数据。

6、如权利要求1所述的方法，其中，所述当检测到所述目标应用进程中的图像
15 渲染函数被调用时，调用数据截取模块，截取当前生成的多个窗口图像数据，包括：

确定所述图像渲染函数中包括的渲染方式；

根据所述渲染方式，确定所述数据截取模块中与所述渲染方式对应的数据截取子模块；

调用所述数据截取子模块，截取在执行所述图像渲染函数时得到的多个窗口图
20 像数据。

7、如权利要求3所述的方法，还包括：

根据各个窗口图像数据的窗口层级标识，生成消息响应二叉树；

其中，所述按照所述合成顺序，对所述多个窗口图像数据进行图像合成处理，
25 包括：

基于所述消息响应二叉树，确定各个窗口图像数据的显示位置信息；

根据所述显示位置信息和所述合成顺序，对所述多个窗口图像数据进行合成处理。

- 8、如权利要求7所述的方法，其中，所述消息响应二叉树中包括多个消息节点，
- 5 所述方法还包括：

接收所述用户终端发送的用户操作消息，从所述用户操作消息中获取操作位置信息；

按照预设的第三遍历顺序，遍历所述消息响应二叉树，将所述操作位置信息满足预设位置条件的消息节点，确定为目标消息节点；

- 10 将所述目标消息节点对应的窗口图像数据作为目标窗口图像数据，响应于所述用户操作消息，对所述目标窗口图像数据所对应的窗口进行处理。

9、如权利要求8所述的方法，其中，每个消息节点中记录有相应的窗口图像数据对应的窗口位置信息和窗口尺寸信息；

- 15 所述操作位置信息满足预设位置条件，是指所述操作位置信息在所述目标消息节点记录的窗口位置信息和窗口尺寸信息所限定的范围内。

10、一种图像获取装置，其特征在于，所述装置配置于服务器中，所述装置包括：

- 20 获取单元，用于从应用进程集合中获取用户标识对应的目标应用进程；

截取单元，用于当检测到所述获取单元获取到的目标应用进程中的图像渲染函数被调用时，调用数据截取模块，截取当前生成的多个窗口图像数据；

第一处理单元，用于对所述截取单元截取到的多个窗口图像数据进行图像合成处理，得到待显示的用户界面图像；

- 25 发送单元，用于将携带所述第一处理单元得到的用户界面图像的通知消息发送给所述用户标识对应的用户终端，以使所述用户终端在用户界面上显示所述用户界

面图像。

11、如权利要求 10 所述的装置，其中，所述获取单元还用于，获取为所述目标应用进程设置的画面显示参数；

5 所述装置还包括：

确定单元，用于根据所述获取单元获取的画面显示参数，确定各个窗口图像数据的窗口层级标识；

所述第一处理单元还用于，根据所述确定单元确定的窗口层级标识，对所述多个窗口图像数据进行图像合成处理。

10

12、如权利要求 11 所述的装置，还包括：

第一生成单元，用于根据所述确定单元确定的各个窗口图像数据的窗口层级标识，生成图像二叉树；

15 所述第一处理单元还用于，按照预设的第一遍历顺序，对所述第一生成单元生成的图像二叉树进行遍历，得到各个窗口图像数据的合成顺序；按照所述合成顺序，对所述多个窗口图像数据进行图像合成处理。

13、如权利要求 12 所述的装置，其中，所述第一生成单元用于，根据所述确定单元确定的各个窗口图像数据的窗口层级标识，构建图像多叉树；将所述图像多叉
20 树转换为所述图像二叉树。

14、如权利要求 11 所述的装置，还包括：

第一生成单元，用于根据所述确定单元确定的各个窗口图像数据的窗口层级标识，生成图像二叉树；

25 其中，所述截取单元用于，按照预设的第二遍历顺序，对所述第一生成单元生成的图像二叉树进行遍历，得到各个窗口图像数据的图像捕获顺序；按照所述图像

捕获顺序，调用所述数据截取模块，截取得到所述多个窗口图像数据。

15、如权利要求 10 所述的装置，其中，所述截取单元用于，确定所述图像渲染函数中包括的渲染方式；根据所述渲染方式，确定所述数据截取模块中与所述渲染方式对应的数据截取子模块；调用所述数据截取子模块，截取在执行所述图像渲染函数时得到的多个窗口图像数据。

16、如权利要求 12 所述的装置，还包括：

第二生成单元，用于根据所述确定单元确定的各个窗口图像数据的窗口层级标识，生成消息响应二叉树；

其中，所述第一处理单元用于，基于所述第二生成单元生成的消息响应二叉树，确定各个窗口图像数据的显示位置信息；根据所述显示位置信息和所述合成顺序，对所述多个窗口图像数据进行合成处理。

17、如权利要求 16 所述的装置，其中，所述消息响应二叉树中包括多个消息节点，所述获取单元还用于，接收所述用户终端发送的用户操作消息，从所述用户操作消息中获取操作位置信息；

所述装置还包括：

第二处理单元用于，按照预设的第三遍历顺序，遍历所述第二生成单元生成的消息响应二叉树，将所述获取单元获取的操作位置信息满足预设位置条件的消息节点，确定为目标消息节点；将所述目标消息节点对应的窗口图像数据作为目标窗口图像数据，响应于所述用户操作消息，对所述目标窗口图像数据所对应的窗口进行处理。

18、一种服务器，其特征在于，还包括：

处理器，适于实现一条或多条指令；以及，

计算机存储介质，所述计算机存储介质存储有一条或多条指令，所述一条或多条指令适于由所述处理器加载并执行如权利要求 1-9 任一项所述的图像获取方法。

- 19、一种计算机存储介质，其特征在于，所述计算机存储介质中存储有计算机程序指令，所述计算机程序指令被处理器执行时，用于执行如权利要求 1-9 任一项所述的图像获取方法。
- 5