



(12) 发明专利

(10) 授权公告号 CN 102736912 B

(45) 授权公告日 2016. 01. 27

(21) 申请号 201110414237. 0

CN 100481013 C, 2009. 04. 22,

(22) 申请日 2011. 12. 13

审查员 章媛

(30) 优先权数据

12/966787 2010. 12. 13 US

(73) 专利权人 微软技术许可有限责任公司

地址 美国华盛顿州

(72) 发明人 B. 刘 邹松 张炜 B. 贝克

J. 克利斯曼 陈百宏

(74) 专利代理机构 中国专利代理(香港)有限公

司 72001

代理人 孙之刚 刘鹏

(51) Int. Cl.

G06F 9/44(2006. 01)

(56) 对比文件

US 2009/0315897 A1, 2009. 12. 24,

US 2009/0144402 A1, 2009. 06. 04,

US 2008/0165207 A1, 2008. 07. 10,

CN 100428182 C, 2008. 10. 22,

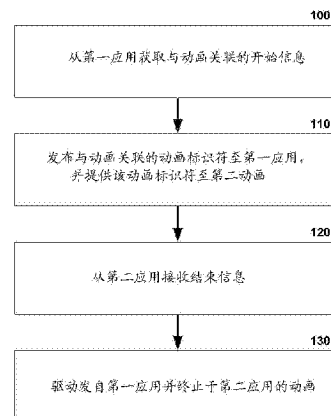
权利要求书2页 说明书14页 附图12页

(54) 发明名称

提供应用之间的动画图形转换的方法和计算设备

(57) 摘要

提供了动画协调系统和方法,它们管理在多个应用之间和 / 或之中的动画场景转换。全局协调器能够从启动应用获取初始信息,诸如初始图形表示和对象类型、初始位置等,以及从目的应用获取最终信息,诸如最终图形表示和对象类型、最终位置等。全局协调基于初始信息和最终信息创建在启动应用和目的应用之间的动画场景转换。



1. 一种提供在计算设备上运行的应用之间的动画图形转换的方法,该方法包括:

由所述计算设备接收将所述计算设备的场景从关联于启动应用的第一场景转换到关联于不同于启动应用的目的应用的第二场景的请求;

由运行在所述计算设备上的动画协调模块至少部分基于来自启动应用的信息和来自目的应用的信息来协调从启动应用到目的应用的动画图形转换,其中协调的动画图形转换提供源自启动应用显示的对象的表现并在目的应用显示的该对象的另一个表示中结束的表现;和

在与所述计算设备关联的显示设备上显示从所述启动应用到所述目的应用的协调的动画图形转换。

2. 权利要求 1 的方法,其中:

协调所述动画图形转换包括:

由所述动画协调模块接收来自启动应用的信息;

由所述动画协调模块接收来自目的应用的信息;和

至少部分基于来自所述启动应用的信息和来自所述目的应用的信息呈现从所述启动应用到所述目的应用的协调的动画图形转换。

3. 权利要求 1 的方法,其中所述方法进一步包括:

将所述计算设备的所述场景从所述第一场景切换到所述第二场景,其中所述协调的动画图形转换被显示为所述场景的所述切换的一部分。

4. 权利要求 1 的方法,其中所述方法进一步包括:

呈现所述协调的动画图形转换,包括:

为一组图形表示中的图形表示来创建帧的顺序;

呈现所述组图形表示的多个图形表示;

确定相对于彼此所述多个图形表示的单独图形表示何时将被显示。

5. 权利要求 1 的方法,进一步包括:

生成对应于所述动画图形转换的动画标识符;

向所述启动应用和所述目的应用发布所述动画标识符;和

与所述动画标识符相结合地存储来自所述启动应用的信息和来自所述目的应用的信息。

6. 权利要求 1 的方法,其中显示所述协调的动画图形转换包括:

在所述显示设备上最顶部的 z 位置处显示所述协调的动画图形转换。

7. 权利要求 1 的方法,显示所述协调的动画图形转换包括:

在开始显示所述协调的动画图形转换之前等待来自所述目的应用的准备就绪指示符。

8. 一种用来提供在其上运行的应用之间的动画图形转换的计算设备,所述计算设备包括:

存储器和处理器,其中所述存储器和所述处理器分别被配置来存储和运行计算机可运行指令,包括执行操作的指令,所述操作包括:

接收将所述计算设备的场景从关联于启动应用的第一场景转换到关联于不同于启动应用的目的应用的第二场景的请求;

至少部分基于来自启动应用的信息和来自目的应用的信息来协调从启动应用到目的

应用的动画图形转换,其中协调的动画图形转换提供源自启动应用显示的对象的表示并在目的应用显示的该对象的另一个表示中结束的表象;和

在与所述计算设备关联的显示设备上显示从所述启动应用到所述目的应用的协调的动画图形转换。

9. 权利要求 8 的计算设备,其中

转换所述计算设备的场景的请求表示用所述目的应用来打开在所述启动应用中描述的对象请求;和

协调所述动画图形转换包括:

由所述动画协调模块接收来自启动应用的信息;

由所述动画协调模块接收来自目的应用的信息;和

至少部分基于来自所述启动应用的信息和来自所述目的应用的信息来协调从所述启动应用到所述目的应用的所述协调的动画图形转换。

10. 权利要求 8 的计算设备,其中协调所述协调的动画图形转换包括:

为一组图形表示中的图形表示来创建帧的顺序;

呈现所述组图形表示的多个图形表示;

确定相对于彼此所述多个图形表示中的单独图形表示何时将被显示。

11. 权利要求 8 的计算设备,其中所述操作进一步包括:

生成对应于所述动画图形转换的动画标识符;

向所述启动应用和所述目的应用发布所述动画标识符;和

与所述动画标识符相结合地存储来自所述启动应用的信息和来自所述目的应用的信息。

12. 权利要求 8 的计算设备,其中显示所述协调的动画图形转换包括:

在所述显示设备上强制的最顶端 z 位置处显示所述协调的动画图形转换。

13. 权利要求 8 的计算设备,其中协调所述协调的动画图形转换包括:

创建多个过渡所述启动应用的初始图形表示和所述目的应用的最后图形表示之间差别的中间图形表示;

其中所述多个中间图形表示的数目根据所述动画图形转换的预定的帧速率而定。

14. 权利要求 13 的计算设备,其中所述操作进一步包括:

确定一组关联于所述多个中间图形表示的位置信息,包括:

基于来自所述启动应用的初始位置信息或者来自所述目的应用的最后位置信息中的至少一个来确定所述位置信息。

提供应用之间的动画图形转换的方法和计算设备

技术领域

[0001] 公开的主题涉及跨计算系统的多个应用或者进程(例如从计算系统的一个场景(context)到另一个场景)的动画协调。

背景技术

[0002] 当可用计算资源与计算技术的发展相一致地持续增加时,在提供丰富的用户体验方面,额外的资源变得可消耗。提供丰富体验的一个机制是动画。动画能够被用来在不同的场景间创建流畅和无缝地转换。举例来说,例如从应用的一个模式到应用的另一个模式的转换之类的场景转换能够是动画式的,从而为用户提供在模式之间无缝的、可见的切换。

[0003] 然而,在一些情形中,场景转换能够涉及或扩展跨过多个进程或应用。在这种转换的一个例子中,从文件浏览应用中的缩略表示打开一个在图片观看应用中的图片,涉及从文件浏览应用到图片观看应用的场景转换。进一步地对于这个例子,图片观看应用能够以全屏表示显示该图片。

[0004] 典型地,能够以单向或双向的形式尝试动画化例如在文件浏览应用和图片观看应用之间的这样一种场景转换。以单向的方式,文件浏览应用或者图片观看应用能够驱动可视的转换动画。然而,如果文件浏览应用绘制(draw)动画,则它不知道在图片观看应用中例如显示的终点是在何处。类似地,当图片观看应用绘制动画时,图片观看应用不能处理关于文件浏览应用中动画起始点的知识。

[0005] 为了双向地驱动动画,在转换中涉及的应用将联合地绘制该动画。根据前面的例子,文件浏览应用和图片观看应用都能够绘制动画。例如,文件浏览能够启动动画,例如开始呈现(render)(例如绘制)该动画,图片观看应用能够终止该动画。因此,在文件浏览应用和图片观看应用之间发生绘制责任的切换。该切换引起动画中的缝隙或者不协调,其对于用户是可见的。

[0006] 当前可见的转换系统的上述不足仅仅是意图提供传统系统中的一些问题的概况,并不是全面的。在查看了下面的描述时,传统系统的其它问题以及此处描述的多个非限制性实施方式的相应益处将进一步变得明显。

背景技术

[0007] 此处提供了简要的概述以帮助使得对于下面更详细描述和附图中的例子、非限制性实施方式的多个方面能够有基本或一般性的理解。然而,该概述并不意图作为广泛的或者全面的概述。替代地,该概述的唯一目的是以简要形式提出与一些示例性非限制性实施方式相关的一些概念,作为下面多个具体实施方式的更详细描述的前序。

[0008] 在一个或多个实施方式中,能够使用通用的构架创建在任意应用之间的场景转换的无缝动画。该构架包括动画协调器,其与应用交互以从启动应用(initiator application)(例如在动画起始点的应用)获得初始信息和从目的应用(例如在动画结束点的应用)获得最终信息。动画协调器基于初始信息和最终信息产生动画并从启动应用和 /

或目的应用之外驱动动画以使得在不同场景之间能够实现无缝且平滑的动画化的转换。

[0009] 在一些实施方式中,动画协调器同步启动应用和目的应用,这样目的应用能够在动画完成时立即捕获焦点和 / 或控制。例如,在应用装入 (launch) 转换中,动画协调器能够暂停以等待来自目的应用处于准备就绪状态的指示。在接收到该指示时,动画协调器能够驱动动画。

[0010] 在下面将更详细地描述这些和其它实施方式。

附图说明

[0011] 参考附图进一步描述了多个非限制性实施方式,其中:

[0012] 图 1 是图示了用于跨越多个应用协调动画的示例性非限制实施方式的流程图;

[0013] 图 2 是图示了另一个用于跨多个应用协调动画的示例性非限制性实施方式的流程图;

[0014] 图 3 是图示用于生成动画的示例性非限制性实施方式的流程图;

[0015] 图 4 是图示用于绘制动画的示例性非限制性实施方式的流程图;

[0016] 图 5 是跨接 (bridge) 多个应用的转换动画的示例性非限制性图示;

[0017] 图 6 是图示示例性非限制动画协调系统的框图;

[0018] 图 7 是跨接多个应用和包括完全不同的初始和最终图形表示的转换动画的示例性非限制图示;

[0019] 图 8 是图示用于协调由一个用户启动并在多个应用中终止的一个或多个动画的示例性非限制实施方式的流程图;

[0020] 图 9 是在一个应用中发起和在多个应用中终止的多个转换动画的示例性非限制图示;

[0021] 图 10 是根据一个或多个实施方式的示例性非限制动画协调系统的框图;

[0022] 图 11 是根据一个或多个实施方式的示例性非限制协调器的框图;

[0023] 图 12 是图示根据一个或多个实施方式的示例性非限制动画协调系统的框图;

[0024] 图 13 是表示此处描述的多个实施方式能够在其中实现的示例性非限制联网环境的框图;和

[0025] 图 14 是表示示例性非限制计算系统、设备或操作环境的框图,此处描述的多个实施方式的一个或多个方面能够在其中实现。

具体实施方式

[0026] 概述

[0027] 如背景技术中讨论的,在不同应用之间可视地动画化的场景转换实际上不能被例如计算设备之类的计算系统上的任何应用广泛地使用,而同时对用户保持无缝。例如,两个不相关的应用(例如由不同开发者开发和 / 或控制的应用)不能交换信息以使得能够在其之间无缝的动画化转换。然而,由单个开发者开发和 / 或控制的两个应用能够通过一种机制被硬编码以动画化场景转换,这种机制仅仅对于那两个应用可用并且只能被那两个应用用于在该两个应用之间的转换动画。

[0028] 进一步地,两个任意的应用或进程(例如不相关的应用)可以使用其它传统机制以

实现转换动画。例如,这种应用能够使用进程间通信技术、共享存储空间、进程间同步技术等以执行由一个应用开始并由另一个应用结束的转换动画的切换。然而传统的机制不能创建无缝和平滑的动画,因为切换点(例如绘制动画的责任改变的点)导致用户可见的动画伪像。因此,创建场景转换的可视动画的传统技术不能以一般化和通用的方式生成在多个任意应用和进程之间的无缝动画。

[0029] 在多个、非限制性实施方式中,能够使用通用的构架来创建在任意应用之间的场景转换的无缝动画。该通用构架提供统一应用编程接口(API),可由应用使用以提供信息来创建转换动画。该通用构架保持来自启动应用(例如动画从其发起的应用)和目的应用(例如动画在该处终止的应用)的信息,并关联这种信息以使得动画能够从特定启动应用绘制到特定目的应用。

[0030] 根据另一个实施方式,该构架包括全局协调器,其与应用交互以获取初始信息和终止信息。另外,全局协调器基于该初始和终止信息控制动画定时和绘制以创建无缝转换。进一步地,全局协调器能够与协调器、排序器(compositor)或其它虚拟或物理机的其它组件通信以动画化在完全不同的虚拟或物理机上的应用和/或进程之间的场景切换。

[0031] 在一个实施方式中,此处描述了一种方法,其包括从启动应用接收初始信息,其中该初始信息涉及可视化用户环境从启动应用到目的应用的转移的转换动画;从目的应用接收与转换动画相关联的最终信息;至少部分地基于初始信息和最终信息生成转换动画;以及在计算系统的显示设备上绘制该转换动画。该方法还能够包括生成相应于转换动画的动画标识符,发布(issue)该动画标识符至启动应用或者目的应用中的至少一个;以及与该动画标识符相关联地保持初始信息和最终信息。进一步地,在一个例子中,接收最终信息能够包括与动画标识符一起接收来自目的应用的最终信息以使得能够将最终信息与初始信息相关联。在另一个例子中,生成转换动画能够包括创建多个中间图形表示,该多个中间图形表示过渡在初始信息中包括的初始图形表示和最终信息中详述的最终图形表示之间的差异,其中创建多个中间图形表示是根据预定的转换动画的帧速率,并进一步包括确定分别与多个中间图形表示相关联的位置信息的集合,其中位置信息是基于初始信息中的初始位置信息、包括在最终信息中的最终位置信息或者包括在位置信息集合中的其它位置信息中的至少一个确定的。

[0032] 根据再另一个例子,绘制转换动画包括排序初始图形表示、多个中间图形表示和最终图形表示以创建帧次序,并根据该帧次序,顺序地并一个一个地绘制初始图形表示、多个中间图形表示和最终图形表示,其中在绘制之间的时间是基于预定的帧速率而确定的。此外,绘制能够涉及强制(enforce)转换动画的z位置(z-position),从而使得转换动画保持是在显示设备上的最顶部对象。在又一个例子中,该方法能够包括在开始绘制转换动画之前等待来自目的应用的准备就绪指示符,其中接收来自目的应用的结束信息是该准备就绪指示符。

[0033] 在另一个实施方式中,如此处描述的系统包括动画协调组件,其包括输入组件和动画组件,该输入组件被配置以从一个或多个启动应用获取初始信息和从一个或多个目的应用获取最终信息,该动画组件被配置以在例如显示屏的显示设备上创建并绘制动画,其中该动画提供从与一个或多个启动应用关联的环境至与一个或多个目的应用关联的环境的无缝转换。动画协调器组件能够进一步地包括标识符生成组件,该标识符生成组件被配

置以将动画标识符发布至由动画协调组件所协调的个别的动画,其中动画标识符包含全局标识符(GID),例如但不限于全局唯一标识符(GUID)。还提供了一种输出组件,其被配置为传输动画标识符至一个或多个启动应用作为初始信息的收据(receipt)。在另外的例子中,动画协调组件包括帧生成组件,该帧生成组件被配置以创建将由动画组件绘制的动画的动画帧。

[0034] 在一个例子中,初始信息包括一个或多个启动应用中的对象的初始图形表示、一个或多个启动应用中的对象的类型、初始图形表示的各自的位置或者初始图形表示的各自的大小中的至少一个。类似地,最终信息包括一个或多个目的应用中的对象的最终图形表示、一个或多个目的应用中的对象的类型、最终图形表示的各自的位置或者最终图形表示的各自的大小中的至少一个。

[0035] 在附加的实施方式中,动画协调子系统能够被集成至操作系统中,并且包括被配置为使得能够触发动画的第一应用编程接口(API)和被配置为接收表示动画结束点的最终信息的第二 API,该动画无缝地可视化在完全不同场景间的转换,其中第一 API 适于从启动应用接收表示动画起始点的初始信息,并返回相应于该动画的动画标识符,其中最终信息与动画标识符相关联地被接收且来自目的应用。动画协调模块基于初始信息和最终信息生成动画并输出该动画至显示屏。

[0036] 此处,提供跨多个应用的无缝转换动画的一些实施方式的概念已经在上面提出。作为下面内容的路标,全局动画协调的各种示例性、非限制性实施方式和特征被更详细地描述。然后,给出了一些非限制性实现方式和例子用于附加地阐述,随后是能够在其中实现这种实施方式和 / 或特征的有代表性的网络和计算环境。

[0037] 在多个应用 / 进程间的转换动画

[0038] 如上面提到的,在多个实施方式中,提供了一种构架,其被配置以生成在多个应用之间延伸的动画。在一个例子中,动画包括在第一应用中的对象的图形表示至第二应用中的对象的另一个图形表示的可视变形。在一个方面,初始图形表示和终点图形表示能够是常见基本对象的不同表示。例如,初始图形表示和终点图形表示能够在一个或多个特征上改变,例如解析度、大小、保真度、颗粒度、位置等,动画能够包含从初始图形表示至终点图形表示的无缝变换。在另一个方面,初始图形表示和终点图形表示能够表示完全不同的对象,例如但不限于文本对象、图片对象、图像对象、视频对象、地图对象、列表对象、web 对象等。在完全不同的对象的情况下,动画能够包含从一个对象的一个表示至另一个对象的另一个表示的形态变换。例如,动画能够可视地描述文本的一部分至图像的变换。在另一个实施方式中,能够生成将多个对象变换为单个对象和 / 或将单个对象变换为多个对象的动画。

[0039] 而且,多个应用能够驻留在单个虚拟或物理机上或者驻留在不同的虚拟和 / 或物理机上。进一步地,多个应用能够是单个应用的多个实例或进程。在又一个实施方式中,单个应用能够使用此处提供的构架以用于应用内转换。例如,当在不同的域之间浏览,例如从第一 web 页面浏览到第二 web 页面时,浏览器应用能够利用动画化的转换。因此,构架除了创建应用间的动画,还被按照一般化的方式用于应用内转换。

[0040] 关于一个或多个非限制性方式以如上所述地协调场景转换的动画,图 1 示出了一个流程图,图示了用于跨越多个应用协调的动画的示例性、非限制性实施方式。在 100 处,

从第一应用获取与动画关联的开始信息。在一个具体到非限制性的例子中,如此处描述的,由全局协调器来获取该开始信息,该全局协调器可以与第一应用一起驻留在物理或虚拟机上。然而,应当认识到全局协调器能够驻留在与第一应用不同的物理或虚拟机上。

[0041] 开始信息能够指定参数以开始动画。例如,开始信息能够包括在第一应用中的对象的初始图形表示,其是动画的开始帧。开始信息还能够包括绘制初始图形表示的显示器上的位置,例如 x 坐标、y 坐标等。显示器上的位置与实际的物理显示器上的位置或者虚拟显示器上的位置相关联。进一步地,初始图形表示的大小能够被包括在开始信息中。在一个例子中,位置和大小信息能够作为初始图形表示的边界长方形而传送。该边界长方形可以通过高度、宽度、预先确定的角(例如左上、右上、左下或右下)的坐标来指定。

[0042] 在 110 处,与动画关联的动画标识符被发布。在一个方面,动画标识符是全局标识符;然而,能够使用其它类型的标识符作为动画标识符。根据一个实施方式,全局协调器能够生成动画标识符并将该动画标识符发布至第一应用。在另一个实施方式中,第一应用能够生成动画标识符。另外,根据哪个生成标识符,全局协调器和/或第一应用能够将动画标识符提供给第二应用,动画在第二应用中结束。

[0043] 在 120 处,从第二应用接收结束信息。类似于开始信息,结束信息能够包括第二应用的最终图形表示,其是最终的动画帧。另外,结束信息能够包括最终图形表示的位置、最终图形表示的大小和/或最终图形表示的边界长方形。在 130 处,在第一应用中发起并在第二应用中终止的动画是基于开始信息和结束信息而驱动的。例如,全局协调器能够绘制该动画,该动画开始于开始信息中指示的位置处的初始图形表示,结束于结束信息中指示的位置处的最终图形表示。全局协调器能够影响在显示器上的图形对象(例如窗口等)的 z 轴次序(z-order),以确保动画保持至少相对于第一应用和第二应用是最上的对象。

[0044] 在另一个实施方式中,动画能够根据由全局协调器强制的定时或同步被驱动。例如,全局协调器在接收到第二应用已准备好的通知之后开始呈现动画。该通知可以是显示的,例如第二应用发送特定消息至全局协调器,或者该通知可以是隐式的,例如全局协调器在接收到结束信息时假定准备就绪。在发送准备就绪的通知之前,第二应用能够呈现相关联的窗口和用户界面,但是保持用户界面隐藏。一旦动画完成,使得所呈现的窗口和用户界面可见以创建无缝场景转换。通过等待来自第二应用的准备就绪信号,由于第二应用的用户界面能够被快速地显示,所以能够发生无缝动画。

[0045] 图 2 示出了流程图,图示了跨越多个应用协调动画的另一个示例性和非限制性实施方式。在 200 处,从启动应用接收初始信息。初始信息与转换动画关联并触发该转换动画,该转换动画可视化用户场景从启动应用至目的应用的切换。在 210 处,生成相应于转换动画的动画标识符。在 220 处,动画标识符被发布至启动应用和/或目的应用。动画标识符提供初始信息的收据,以及指示转换动画正在被构建。而且,动画标识符提供参考,该参考能够被用于将多个数据和/或消息与转换动画相关联。

[0046] 在 230 处,从目的应用接收与转换动画相关联的最终信息。在一个例子中,最终信息能够与动画标识符相关地被接收。在 240 处,初始信息和最终信息能够可选地被与动画标识符相关联地存储。在另一个实施方式中,在开始显示转换动画之前,进程等待直到从目的应用接收到准备就绪指示符。在特定的、非限制性例子中,接收最终信息能够操作为隐式的准备就绪指示。

[0047] 在 250 处,生成转换动画。简单地转向图 3,其图示了生成动画的示例性、非限制性实施方式的流程图。在 300 处,能够创建多个中间图形表示。中间图形表示能够过渡在初始图形表示和最终图形表示之间的间隔。在一个实施方式中,创建的多个中间图形表示能够基于转换动画的预定帧速率。在 310 处,确定位置集合,其中位置集合分别与多个中间图形表示相关联。根据一个例子,能够基于初始图形表示的初始位置、最终图形表示的最终位置和 / 或位置集合中的其它位置确定该位置。在 320 处,通过将图形表示与位置匹配来生成动画帧。

[0048] 向回转至图 2,在 260 处绘制转换动画,其在图 4 描述的示例性、非限制性实施方式中被进一步图示。如图 4 中所示出的,在 400 处,图形表示集合被排序以创建帧次序。在 410 处,根据帧次序,来自图形表示集合的图形表示被顺序地并一个一个地绘制。此外,在 420 处,在绘制期间,动画的 z 位置被强制以保持动画在最顶部位置。

[0049] 图 5 是过渡多个应用的转换动画的示例性、非限制性图示。如图 5 中所图示的,动画能够可视地执行从第一应用(例如电子邮件应用 500)至第二应用(例如图形浏览器应用 510)的场景转换。电子邮件应用 500 能够是任何合适的,能够接收、浏览、撰写、发送邮件等的电子邮件应用。如图 5 中所图示的,电子邮件应用 500 能够浏览包含嵌入的诸如图像 502 之类图像的电子邮件。在一个实施方式中,电子邮件 500 将图像 502 显示为嵌入在所显示的电子邮件消息内的缩略表示。

[0050] 如果可得到,用户能够选择在图形浏览器应用 510 中观看较大的表示。为了实现此,用户能够双击图像 502、右键点击图像 502 并选择命令以在图片浏览器应用 510 中打开图像 502,或者使用一些其它可在电子邮件应用 500 的用户界面中可用的其它手段。在非限制性例子中,电子邮件应用 500 能够利用主操作系统的工具以查找由操作系统保持的文件关联,以识别哪个应用与图像 502 的文件类型相关联。一旦电子邮件应用 500 识别图像浏览器应用 510 作为与图像 502 的文件类型相关联的应用,电子邮件应用 500 能够使用附加的操作系统的机制以使用相应于图形 502 的文件装入图形浏览器应用 510。图形浏览器应用 510 能够执行,打开相应于图像 502 的文件,并以图像 512 的形式显示较大的表示。在另一个例子中,电子邮件应用 500 能够发信号通知操作系统期望在不同的应用中打开 / 观看的文件类型和 / 或文件名。操作系统能够识别合适的程序,例如图像浏览器应用 510,执行该程序并为该程序提供文件。

[0051] 如在上面的例子中所描述的,应用装入场景包含从电子邮件应用 500 的环境到图像浏览器应用 510 的环境的突然场景转换。典型地,这种场景转换是突然的。例如,用户在电子邮件应用 500 中双击图像 502 并指令操作系统利用与图像 502 关联的文件装入图形浏览器应用 510。一旦图片浏览器应用 510 开始执行,多种后台任务被执行。图片浏览器应用 510 能够访问文件并访问所存储的该文件的信息。进一步地,图片浏览器应用 510 能够在后台处理期间显示闪屏 (splash screen)。最终,图像浏览器应用 510 呈现窗口和 / 或用户界面,其被显示在屏幕上或者其它输出显示器上。惯常地,从双击图像 502 至显示图片浏览器应用 510 的窗口或用户界面的转换可以是急剧的。例如,从视觉观点,例如用户在输出显示器上看到的,用户观看电子邮件应用 500,随后在经过一些时间段之后的一个时刻图片浏览器应用 510 变得可见。可选地,图片浏览器应用 510 的显示之前能够是突然显示闪屏。在另一个例子中,根据系统资源和 / 或应用被编码的方式,图像浏览器应用 510 的显示能够

是不连续的,用户界面的多个部分以步进式的方式变得可见。

[0052] 如上面描述的,动画能够被生成以提供从电子邮件应用 500 至图片浏览器应用 510 的无缝和平滑转换。如图 5 中所示出的,动画能够是图像 502 (例如初始图形表示) 至图像 512 (例如最终图形表示) 的可视放大或转换。动画能够包括从开始帧(图像 502) 至结束帧(512) 排序的多个帧。诸如帧 504 和帧 514 之类的中间帧能够被生成。能够从在前帧和 / 或开始帧生成中间帧。在另一个例子中,能够以相反顺序生成中间帧,例如基于后续帧和 / 或结束帧。应当认识到两种生成技术都能够被使用以提供与相邻帧紧密相关的高质量中间帧。例如,如果与从开始帧生成中间帧相反,从结束帧生成中间帧,接近结束帧的中间帧能够具有更高质量。多个中间帧能够根据多种因素,诸如但不限于,动画的持续时间和期望的帧速率。根据一种非限制性的例子,假设一个两秒的动画具有每秒 24 帧的帧速率。该动画将总共包括 48 个帧。帧能够被排序并显示,从而使得占据开始位置的图像 502 变形为较大表示,例如转换为图像 512 占据的最终位置时的图像 512。

[0053] 图 6 是图示示例性、非限制性动画协调系统的框图。动画协调系统包括应用 600 (例如第一应用) 和应用 610 (例如第二应用),在它们之间发生动画化的场景转换。系统还能够包括协调器 620,被配置以生成动画 624,同步应用 600 和应用 610,以及绘制动画 624。在一个特定的、非限制性实施方式中,协调器 620 能够是作为主机管理应用 600 和应用 610 的操作系统(未示出)的全局组件。然而,应当认识到协调器 620 能够是与操作系统分离的进程。

[0054] 应用 600 能够通过发送开始信息 602 至协调器 620 启动动画 624。开始信息 602 在一个实施方式中能够指定一个或多个初始图形表示、初始位置信息、和 / 或初始大小信息。如上面描述的,协调器 620 能够生成动画,所述动画将许多对象转换为单个对象,将单个对象转换为多个对象,和 / 或将单个对象转换为另一个单个对象。因此,在多到一转换的情况下,开始信息 602 能够包括多个初始图形表示,具有各自的位置和大小信息。在一到多或者一到一转换的情况下,开始信息 602 能够包括单个初始图形表示,具有各自的位置和大小信息。

[0055] 协调器 620 在接收到开始信息 602 时生成全局标识符(GID)622 作为动画标识符。GID 622 与动画 624 相关联并被使用来标引开始信息 602 以与动画 624 相关。协调器 620 响应于开始信息 602 提供 GID 622 给应用 600。另外,协调器 620 还能够提供 GID 622 给应用 610。可替换地,应用 600 能够在应用装入信息中包括 GID 622,该应用装入信息从应用 600 经由用以开始应用 610 的操作系统装入机制传送至应用 610。在任一种考虑中,当被应用 610 接收时,GID 622 操作以通知应用 610 动画 624 将发生以便可视化地从应用 600 的场景转换至应用 610 的场景。

[0056] 在获取 GID 622 时,应用 610 能够传输结束信息 612 至协调器 620。类似于开始信息 602,结束信息 612 能够包括一个或多个最终图形表示、最终位置信息和 / 或最终大小信息。应用 610 能够与 GID 622 相关地传输结束信息 612,以使得协调器 620 能够将结束信息 612 与开始信息 602 以及已经与 GID 622 相关联的动画 624 相关联。

[0057] 应用 610 能够传输结束信息 612 作为准备就绪的隐式指示。例如,通过发送结束信息 612,应用 610 通知协调器 620 应用 610 已准备在绘制时捕获动画 624。因此,在传输结束信息 612 之前,应用 610 能够执行任何启动处理以准备显示用户界面或窗口,从而防止

在应用 610 能够绘制它的窗口或用户界面之前动画 624 完成。

[0058] 在接收结束信息 612 时,协调器 620 能够至少部分地基于开始信息 602 和结束信息 612 生成动画 624。例如,协调器 620 能够利用在开始信息 602 中的初始图形表示作为开始帧和在结束信息中的最终图形表示作为结束帧。协调器 620 从初始图像表示和 / 或最终图形表示中生成多个中间帧,如关于图 5 所描述的。协调器 620 能够排序并绘制动画 624 的帧,以提供从应用 600 的场景至应用 610 的场景的无缝和平滑转换。

[0059] 如上面描述的,作为动画的结束点的图形表示能够描述多种对象,诸如但不限于,信息、图片、相片、视频素材、徽标、文本、视频、web 页面、图形、地图、前述的组合等的列表。为了图示的目的,随后的非穷举和非限制性的例子展现了一些动画类型,它们能够被用于应用之间、应用之中、虚拟机之间、物理机之间等的场景转换。在一个例子中,条形码能够是动画的初始图形表示,该动画终止于一些由该条形码指定的产品的表示。例如,条形码能够转换至关于产品的 web 页面、产品的图像、产品说明书页等。类似地,例如在 web 页面上、在电子杂志中或类似媒介中与公司关联的广告或图像能够转换为公司提供的产品列表、购买产品的 web 页面等。在另一个例子中,例如地址簿应用中、在电子名片上等的联系人姓名能够被动画化至该联系人的另一个表示,诸如相片、公共社交媒体简档 (public social media profile)、描述位置的地图等。在再另一个例子中,对象的图表或图示能够从折叠视图动画转换至展开视图。例如,工程设计应用能够使用此处描述的实施方式来动画化设备设计的展开。

[0060] 在另一个例子中,图 7 是一个过渡多个应用并包括与完全不同的对象关联的初始和最终图形表示的转换动画的示例性、非限制性图示。如图 7 中所示出的,动画能够可视地执行从例如电子邮件应用 700 的第一应用至例如地图浏览器应用 710 的第二应用的场景转换。特别地,动画能够包括从电子邮件消息中的地址 702 至地图 712 上相应位置的场景转换。从电子邮件应用 700 装入地图浏览器应用 710 能够以类似于上面参照图 5 所描述的那种方式而发生。另外,在电子邮件应用 700 和地图浏览器应用 710 之间的动画的协调能够使用在前描述的方法和系统来实现。

[0061] 在图 7 所图示的示例性动画的初始图形表示是地址 702,其由不可见矩形作为界限的一部分文本组成。最终图形表示是地图 712,其包括道路、路口、建筑物、卫星图等图形图像。诸如协调器 620 之类协调组件被配置来生成并控制过渡多个应用的动画,能够生成一个或多个中间帧 704。中间帧 704 能够基于初始图形表示和最终图形表示的各种合成画面以提供从包括在地址 702 中的文本至包括在地图 712 中的图形的动画化变形。

[0062] 在另一个实施方式中,图 8 图示了用于协调由一个应用启动并在多个应用中终止的一个或多个动画的示例性、非限制性方法的流程图。在 800 处,从第一应用获取与一个或多个动画关联的开始信息。在 810 处,生成动画标识符并将其发布至第一应用,其中该动画标识符与一个或多个动画相关联。进一步地,该动画标识符被提供给目的应用集合。在一个实施方式中,能够为一个或多个动画共同地发布单个动画标识符,动画个别地开始于开始信息中指定的初始图形表示并分别终止于目的应用集合中的一个或多个应用的终点。在另一个例子中,能够生成不同的动画标识符并为第一应用至集合中的目的应用的每个配对发布。

[0063] 在 820 处,从目的应用集合中的一个或多个目的应用接收各自的结束信息。结束

信息能够共同地包括用于一个或多个目的应用中的每一个的最终图形表示以及用于每个最终图形表示的各自的位置和大小信息。在 830 处,绘制了一个或多个动画,从第一应用开始并在目的应用集合的一个或多个目的应用中终止。

[0064] 图 9 是在一个应用中发起并在多个应用中终止的多个转换动画的示例性、非限制性图示。如图 9 中所示出的,动画能够可视地执行从例如电子邮件应用 900 的第一应用至例如地址簿应用 910 和浏览器应用 920 的多个目的应用的场景转换。特别地,图 9 图示了多个动画,分别描述了从电子邮件应用 900 至地址簿应用 910 的场景转换以及从电子邮件应用 900 至浏览器应用 920 的场景转换。在电子邮件应用 900 和地址簿应用 910 之间的场景切换能够包括观看与由电子邮件应用 900 所显示的电子邮件消息中包括的联系人 902 (例如联系人的姓名)关联的地址簿中的条目。另外,在电子邮件应用 900 和浏览器应用 920 之间的场景切换能够包含从在电子邮件消息中选择联系人 902 至在浏览器应用 920 中观看 / 显示公共社交媒体简档的转换。

[0065] 在图 9 中图示的示例动画的初始图形表示是联系人 902,其由不可见矩形作为界限的一部分文本组成。动画的最终图形表示包括地址簿应用中的地址簿条目 912 和浏览器应用 920 的简档 922。如图 9 中公所描述的,地址簿条目 912 能够包含图像和文本的组合。类似地,简档 922 能够是同样地包括诸如文本、图像、视频等多种媒体的 web 页面。为了生成动画,能够为每个构成的动画分别地生成中间帧。例如,能够为在联系人 902 和地址簿条目 912 之间的构成动画生成中间帧 914,为在联系人 902 和简档 922 之间的构成动画创建中间帧 924。

[0066] 在一个实施方式中,构成动画能够被共同地或分别地绘制。例如,一旦地址簿应用 910 已经准备好,在联系人 902 和地址簿条目 912 之间的动画就能够被绘制,即使浏览器应用 920 还没有准备好。在另一个例子中,没有动画开始,直到地址簿应用 910 和浏览器应用 920 都报告准备就绪。

[0067] 图 10 是根据一个或多个实施方式的示例性、非限制性动画协调系统的框图。动画协调系统包括启动应用 1000 和目的应用集合 1010,在它们之间发生动画化的场景转换。系统还包括协调器 1020,被配置以生成动画 1024,该动画描述在启动应用 1000 和目的应用集合 1010 之间的场景转换。

[0068] 启动应用 1000 能够利用动画 1024 来可视化与目的应用集合 1010 的场景转换。为了启动动画,启动应用 1000 发送开始信息 1002 至协调器 1020。开始信息 1002 包括初始图形表示、初始位置信息和 / 或初始大小信息。协调器 1020 在接收到开始信息 1002 时生成一个或多个全局标识符(GID)1022。GID 1022 与动画 1024 相关联,并被使用以标引开始信息 1002 与动画 1024 相关。协调器 1020 响应于开始信息 1002 提供 GID 1022 给启动应用 1000。另外,协调器 1020 还能够提供 GID 1022 给目的应用集合 1010 中的应用。

[0069] 在获得 GID 1022 时,目的应用集合 1010 能够传输结束信息集合 1016 给协调器 1020。结束信息集合 1016 包括分别与目的应用集合 1010 中每个目的应用相关联的结束信息。目的应用集合 1010 包括一个或多个应用,例如目的应用 1012 和目的应用 1014,其包括作为转换动画的终点的图形表示。虽然图 10 描述了目的应用集合 1010 中的两个目的应用,但是应当理解目的应用集合 1010 能够包括 N 个应用, N 是大于或等于 1 的整数。结束信息集合 1016 能够包括最终图形表示集合、最终位置信息集合和 / 或最终大小信息集合。

由集合中最终图形表示与所关联的最终位置信息和最终大小信息一起组成的组能够相应于目的应用集合 1010 中的一个目的应用。

[0070] 在接收到结束信息集合 1016 时,协调器 1020 能够基于开始信息 1002 和结束信息集合 1016 生成动画 1024。协调器 1020 能够分别地绘制子动画,例如在启动应用 1000 和目的应用中的一个之间的动画,或者能够共同地绘制在启动应用 1000 和目的应用集合 1010 之间的所有动画。

[0071] 参考图 11,提供了图示根据一个或多个实施方式的示例性、非限制性动画协调器 1100 的框图。如图 11 中所示出的,协调器 1100 能够包括输入组件 1102,被配置为从动画化场景转换中所涉及的一个或多个应用获取开始信息、结束信息、准备就绪指示符等。如上面所述的,开始信息能够包括初始图形表示、与初始图形表示相关联的对象的类型、图形表示的初始位置、图形表示的大小、动画化转换终止的一个或多个目的应用的标识、地址等,或者任何其它合适的信息以便于全局呈现在应用之间的动画。类似地,结束信息指定最终图形表示、与最终图形表示相关联的对象的类型和图形表示的位置和 / 或大小。另外,结束信息还能够包括与特定动画相关联的动画标识符,其将结束信息与相应的开始信息相联系。输入组件 1102 能够在存储器 1104 中与动画标识符相关联地保存开始信息和结束信息。

[0072] 提供了标识符生成组件 1106,其被配置为生成新动画的动画标识符。例如,在从应用接收到新的开始信息时,协调器 1100 开始进程以生成并绘制转换动画。响应于接收开始信息,协调器 1100 能够使用标识符生成组件 1106 以生成相应的动画标识符。在一个实施方式中,标识符生成组件 1106 生成全局标识符(GID),以便被用作动画标识符。然而,应当认识到其他类型的标识符也能被利用作为动画标识符代替 GID。协调器 1100 经由输出组件 1108 能够提供生成的动画标识符给发送新开始信息的应用。另外,输出组件 1108 能够发送动画标识符至在开始信息中指定的任何目的应用。

[0073] 一旦从启动应用接收到开始信息,并从目的应用接收到结束信息,协调器 1100 能够基于接收到的信息创建合适的动画。在一个实施方式中,协调器 1100 能够把初始图形表示和相应的位置 / 大小信息作为动画的开始帧和把最终图形表示(和位置 / 大小细节)作为动画的最终帧。协调器 1100 能够使用帧生成组件 1110 以创建在开始帧和最终帧之间的动画的中间帧。帧生成组件 1110 能够对最终图形表示和 / 或初始图形表示执行多种变换以创建图示在初始图形表示和最终图形表示之间的转换点的中间图形表示。另外,帧生成组件 1110 能够基于初始位置信息和最终位置信息为中间图形表示确定合适的位置信息。协调器 1100 能够在存储器 1104 中与动画标识符相关联地存储生成的中间帧。协调器 1100 还包括动画组件 1112,被配置以在显示器上按照合适的次序和定时绘制开始帧、中间帧和最终帧以创建无缝动画。

[0074] 图 12 是图示了根据一个或多个实施方式的示例性、非限制性动画协调系统的框图。如图 12 中所示出的,动画协调模块 1202 能够被集成在操作系统 1200 中。操作系统 1200 能够被提供在计算系统中以管理计算系统的硬件组件(例如输入设备、输出设备、存储器设备、存储设备、处理设备)并提供通用服务(例如运行时程序库(runtime library)等)给在计算系统上执行的程序,诸如上面所描述的启动和 / 或目的应用。例如,操作系统 1200 能够包括诸如动画协调模块 1202 之类的多个模块,经由多个应用编程接口(API)扩展通用功能。操作系统 1200 能够接收对 API 的调用,执行与其相关的功能或模块,并将结果

返回启动该调用的实体。

[0075] 动画协调模块 1202 能够包括第一 API 1204, 可用于触发在任意应用之间的转换动画。第一 API 1204 适于从启动应用接收初始信息, 并返回动画标识符。动画协调模块 1202 进一步包括第二 API 1206, 被配置以从目的应用接收最终信息。动画协调模块 1202 在经由第二 API 1206 接收最终信息时, 被配置以生成转换动画并输出该动画至显示屏。例如, 动画协调模块 1202 能够利用操作系统 1200 的显示驱动器(未示出) 访问视频呈现设备(例如作为主机管理操作系统 1200 的计算系统的图形卡) 以便在耦合到视频呈现设备的显示屏上可视地显示动画。

[0076] 示例性联网和分布式环境

[0077] 本领域技术人员能够认识到此处描述的转换动画系统和方法的各种实施方式能够关于任何计算机或其它客户机或服务器设备而实现, 其能够被配置作为计算机网络的一部分或者配置在分布式计算环境中, 并且能够连接至任何类型的数据存储器。在这点上, 此处描述的各种实施方式能够被实现在任何计算机系统或者环境中, 该计算机系统或者环境具有任意数量的存储器或存储单元, 以及跨任意数量的存储单元的任意数量的应用和进程。这包括但不限于具有在网络环境中布置的服务器计算机和客户机计算机的环境、或者具有远程或本地存储器的分布式计算环境。

[0078] 通过在计算设备和系统中通过通信交换, 分布式计算提供了计算机资源和服务的共享。这些资源和服务包括信息交换、诸如文件的对象的高速缓存存储和盘存储。这些资源和服务还包括跨多个处理单元的处理能力的共享, 用于负载均衡、资源扩展、处理特殊化等。分布式计算利用网络连接, 允许客户机使用它们的共同能力以有益于整个事业。在这点上, 多种设备能够具有应用、对象或资源, 可以参与所描述的用于公开主题的多个实施方式的资源管理机制。

[0079] 图 13 提供了示例性联网或分布式计算环境的示意图。分布式计算环境包括计算对象 1310, 1312 等和计算对象或设备 1320, 1322, 1324, 1326, 1328 等, 其可以包括程序、方法、数据存储器、可编程逻辑等, 正如通过应用 1330, 1332, 1334, 1336, 1338 所表示的。能够认识到计算对象 1310, 1312 等和计算对象或设备 1320, 1322, 1324, 1326, 1328 等可以包括不同的设备, 诸如个人数字助理(PDA)、音频 / 视频设备、移动电话、MP3 播放器、个人计算机、膝上型电脑等。

[0080] 通过通信网络 1340, 每个计算对象 1310, 1312 等和计算对象或设备 1320, 1322, 1324, 1326, 1328 等能够直接地或间接地与一个或多个其它计算对象 1310, 1312 等和计算对象或设备 1320, 1322, 1324, 1326, 1328 等通信。即使在图 13 中图示为单个元件, 通信网络 1340 可以包括给图 13 的系统提供服务的其它计算对象和计算设备, 和 / 或可以表示多个互相连接的网络, 其未示出。每个计算对象 1310, 1312 等或者计算对象或设备 1320, 1322, 1324, 1326, 1328 等还能够包含诸如应用 1330, 1332, 1334, 1336, 1338 之类的应用, 其可以使用 API 或其它对象、软件、固件和 / 或硬件, 适于与根据所公开的主题的多个实施方式提供的应用测试技术通信或者实现该应用测试技术。

[0081] 有多种系统、组件和网络配置支持分布式计算环境。例如, 计算系统能够通过有线或无线系统, 通过本地网或者广泛分布式网络被连接在一起。当前, 许多网络被耦合至互联网, 其提供了一种用于广泛分布式计算的基础架构, 并包含许多不同的网络, 然而任意网络

基础架构都能够被用于对于在多个实施方式中描述的系统而言易于发生的示例性通信。

[0082] 因此,能够利用许多网络拓扑结构和网络基础架构,诸如客户机/服务器、对等或混合架构。“客户机”是使用它未关联的另一个类或组的服务的类或组的成员。客户机能够是进程,即大致是指令或任务的集合,其请求由另一个程序或进程提供的服务。客户机进程利用所请求的服务,而不需要“知道”关于其它程序或服务自身的工作细节。

[0083] 在客户机/服务器架构中,特别地是联网系统中,客户机通常是访问由另一个计算机(例如服务器)提供的共享网络资源的计算机。在图 13 的图示中,作为非限制性的例子,计算对象或设备 1320, 1322, 1324, 1326, 1328 等能够作为客户机而考虑,计算对象 1310, 1312 等能够作为服务器而考虑,其中充当服务器的计算对象 1310, 1312 等提供数据服务,例如从客户机计算对象或设备 1320, 1322, 1324, 1326, 1328 等接收数据,存储数据,处理数据,传输数据至客户机计算对象或设备 1320, 1322, 1324, 1326, 1328 等,然而根据情况,任何计算机能够被认为是客户机、服务器或二者都是。

[0084] 服务器典型地是可通过远程或本地网络(诸如互联网或者无线网络基础架构)访问的远程计算机系统。客户机进程可以在第一计算机系统中是活动的,服务器进程可以在第二计算机系统中是活动的,通过通信媒介彼此通信,因此提供分布式功能性并允许多个客户机利用服务器的信息聚合能力。

[0085] 在通信网络 1340 或总线是互联网的网络环境中,例如,计算对象 1310, 1312 等可以是 Web 服务器,其它计算对象或设备 1320, 1322, 1324, 1326, 1328 等经由诸如超文本传输协议(HTTP)之类的多个已知协议中的任意一个与之通信。充当服务器的计算对象 1310, 1312 等还可以充当客户机,例如计算对象或设备 1320, 1322, 1324, 1326, 1328 等,如分布式计算环境所特有的。

[0086] 示例性计算设备

[0087] 如所提及的,有利地,此处描述的技术能够被应用于有需要创建动画的任何设备,该动画可视化地描述在计算系统中多个任意应用或进程之间的场景转换。因此能够理解,所有种类的手持、便携和其它计算设备和计算对象都预期在各种实施方式中使用,即动画能够便于无缝场景转换的任何地方。因此,下面在图 14 中描述的通用远程计算机仅是计算系统的一个例子。

[0088] 实施方式部分地能够经由操作系统实现以便供设备或对象的服务开发者使用,和/或包括在操作以执行此处描述的各种实施方式的一个或多个功能方面的应用软件中。以计算机可执行指令的一般上下文描述软件,例如由一个或多个计算机执行的程序模块,该一个或多个计算机例如是客户机工作站、服务器或其它设备。本领域技术人员将认识到计算机系统具有多种配置和协议,能够被用于传送数据,因此没有特定配置或协议被认为是限制。

[0089] 图 14 因此图示了其中能够实现此处描述的实施方式的一个或多个方面的,合适的计算系统环境 1400 的一个例子,虽然如上面所清楚表明的,该计算系统环境 1400 仅是一个合适的计算环境的例子,并且不意图暗示对使用或功能性的范围的任何限制。另外,计算系统环境 1400 不意图被解释为关于在示例性计算系统环境 1400 中图示的组件的任何一个或其组合有依赖性。

[0090] 参考图 14,用于实现一个或多个实施方式的示例性远程设备包括以计算机 1410

形式的通用计算设备。计算机 1410 的组件可以包括但不限于处理单元 1420、系统存储器 1430、耦合包括系统存储器的各种系统组件至处理单元 1420 的系统总线 1422。

[0091] 计算机 1410 典型地包括多种计算机可读介质,并能够是可由计算机 1410 访问的任何可用介质。系统存储器 1430 可以包括易失和 / 或非易失存储器形式的计算机存储介质,诸如只读存储器(ROM)和 / 或随机存取存储器(RAM)。作为例子而非限制,系统存储器 1430 还可以包括操作系统、应用程序、其它程序模块和程序数据。根据另一个例子,计算机 1410 还能够包括多种其它媒介(未示出),其能够包括而限于 RAM, ROM, EEPROM, 闪存或其它存储器技术, CD-ROM, 数字多功能盘(DVD)或其它光盘存储器, 盒式磁带, 磁带, 磁盘存储器或其它磁存储设备, 或者其它能够被用于存储期望信息的有形和 / 或非暂态媒介。

[0092] 用户能够通过输入设备 1440 输入命令和信息至计算机 1410。监视器或者其它类型的显示设备也经由诸如输出接口 1450 的接口被连接至系统总线 1422。除了监视器, 计算机还能够包括诸如扬声器和打印机之类的其它外设输出设备, 其可以通过输出接口 1450 被连接。

[0093] 计算机 1410 可以运行在联网或分布式环境中, 该环境使用连接至诸如远程计算机 1470 的一个或多个其它远程计算机的逻辑连接。远程计算机 1470 可以是个人计算机、服务器、路由器、网络 PC、对等设备或其它通用网络节点, 或者任何其它远程媒体消费或传输设备, 并且可以包括上面关于计算机 1410 描述的任意或全部元件。在图 14 中描述的逻辑连接包括网络 1472, 例如局域网(LAN) 或者广域网(WAN), 但是还可以包括其它网络 / 总线。这种联网环境常见于家庭、办公室、企业计算机网、内联网和互联网中。

[0094] 如上面描述的, 虽然结合各种计算设备和网络架构描述了示例性实施方式, 但是根本的原理可以被应用于期望改善资源使用效率的任何网络系统和任何计算设备或系统。

[0095] 而且, 有多种方式实现相同或类似的功能性, 例如合适的 API、工具箱、驱动器代码、操作系统、控制、独立或可下载软件对象等, 它们使得应用和服务能够利用此处提供的技术。因此, 从 API (或者其它软件对象) 的观点, 以及从实现此处描述的一个或多个实施方式的软件或硬件对象的观点, 此处的实施方式是可预期的。因此, 此处描述的各种实施方式可以具有全部硬件、部分硬件和部分软件、以及全部软件的多个方面。

[0096] 词语“示例性”此处被用于意味着作为例子、实例或例证。为了避免疑问, 此处公开的主题不受这些例子的限制。另外, 此处作为“示例性”描述的任何方面或设计不是必须被解释为对于其它方面或设计是优选或有优势的, 也不意味着排除等同示例结构和本领域普通技术人员已知的技术。此外, 在使用术语“包括”、“具有”、“包含”和其它类似词语的地方, 为了避免疑问, 这些术语意图以类似于作为开放式转换词的术语“包含”的方式是包含性的, 当在权利要求中使用不排除任何附加地或其它元件。

[0097] 如所提及的, 此处描述的多种技术可以结合硬件、软件或合适时结合二者的组合实现。如此处使用的, 术语“组件”、“模块”、“系统”等同样是意图指计算机相关实体、硬件、硬件和软件的组合、软件、或执行中的软件。例如, 组件可以是但不限于在处理器上运行的进程、处理器、对象、可执行、执行的线程、程序和 / 或计算机。通过图示, 在计算机上运行的应用和计算机可以是一个组件。一个或多个组件可以驻留在进程和 / 或执行线程上, 组件可以位于一个计算机和 / 或分布在两个或多个计算机之间。

[0098] 前述的系统已经关于在多个组件之间的交互而描述。能够认识到这种系统和组件

能够包括那些组件或指定的子组件、指定组件或子组件中的一些,和 / 或附加组件,和根据前面的各种置换和组合。子组件还能够被实现为可通信地耦合至其它组件的组件,而非包括在父组件中(分级的)。附加地,能够注意到一个或多个组件可以合并成提供聚合功能性的单个组件,或者分开成多个分离的子组件,诸如协调层之类的任何一个或多个中间层可以被提供以可通信地耦合至这种子组件从而提供集成的功能性。此处描述的任何组件还可以与在此处未特别描述但是通常本领域技术人员已知的一个或多个其它组件交互。

[0099] 鉴于上文描述的示例性系统,参考多个图的流程图,根据所描述的主题可以实现的方法也能够被认识到。然而为了解释简洁的目的,该方法被示出并被描述为一系列框,将理解并认识到多个实施方式不限于框的次序,因为根据这里所描绘和描述的一些框可以按照不同的次序发生和 / 或其它框同时地发生。在经由流程图图示了非顺序的或分支的流程地方,能够认识到各种其它分支、流程路径、框的次序可以被实现,其获得相同或类似的结果。而且,在实现下文描述的方法中一些图示的框是可选的。

[0100] 除了此处描述的各种实施方式,可以理解能够使用其它类似的实施方式,或者能够对所描述的(多个)实施方式做出修改和增加以执行与这些相应的(多个)实施方式同样或等同的功能,而不偏离所述实施方式。再进一步地,多个处理芯片或者多个设备能够共享此处描述的一个或多个功能的性能,类似地,存储能够跨多个设备实现。因此,发明不限于任何单个实施方式,而是要按照根据所附权利要求的广度、精神和范围来解释。

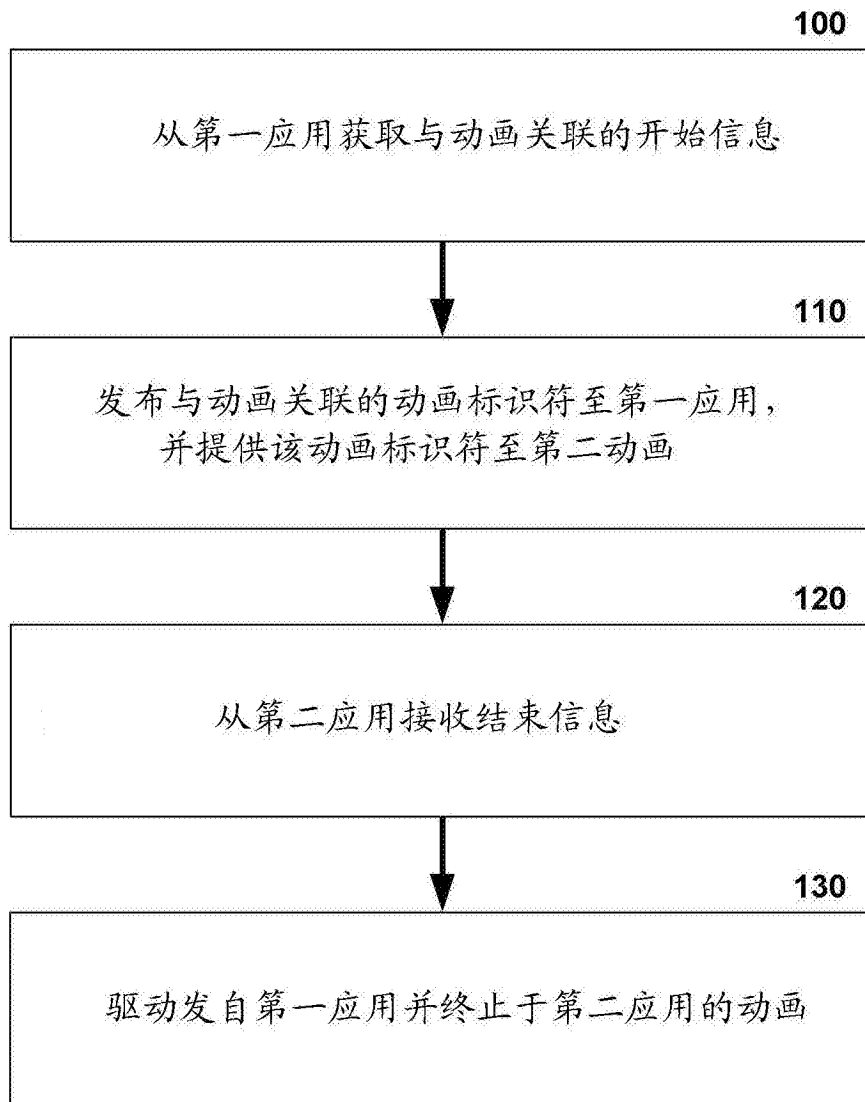


图 1

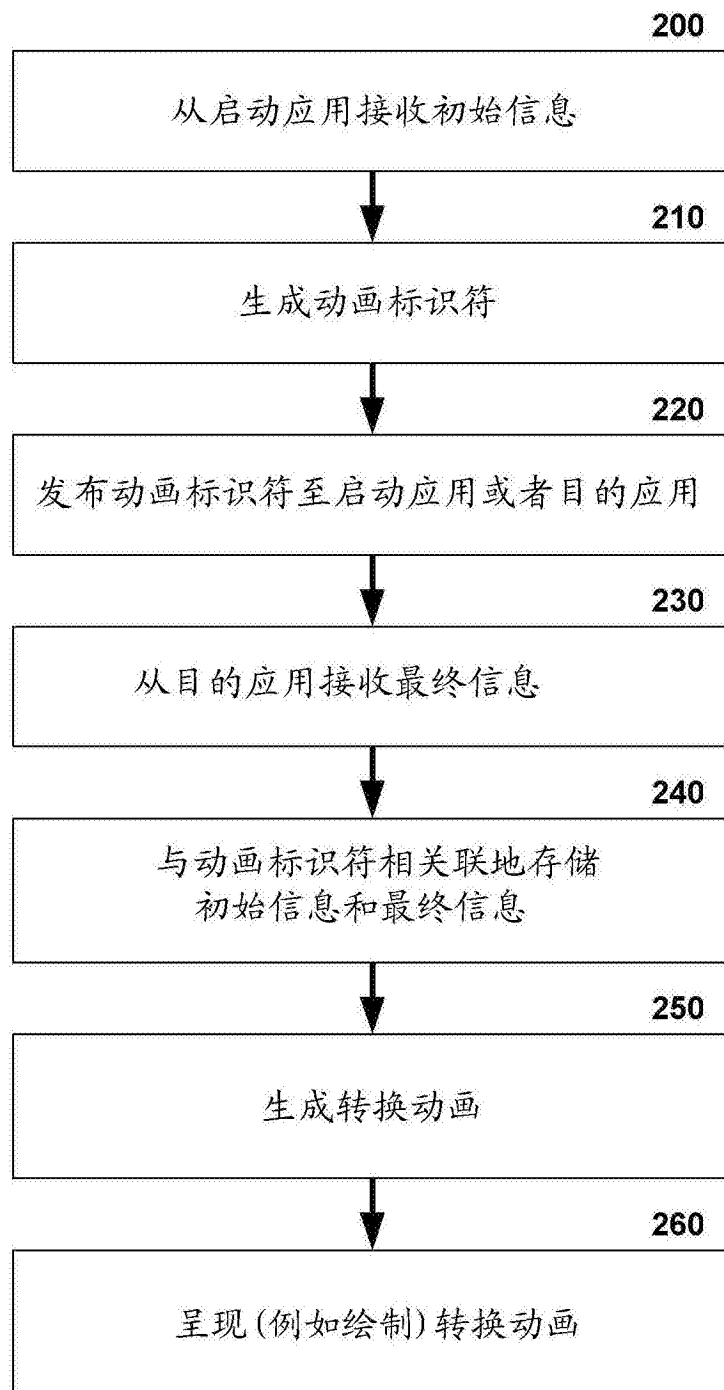


图 2

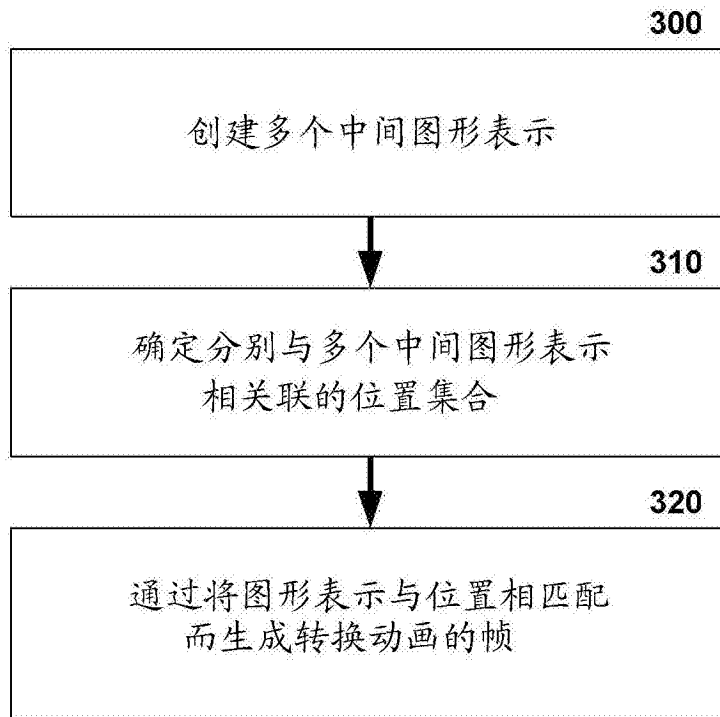


图 3

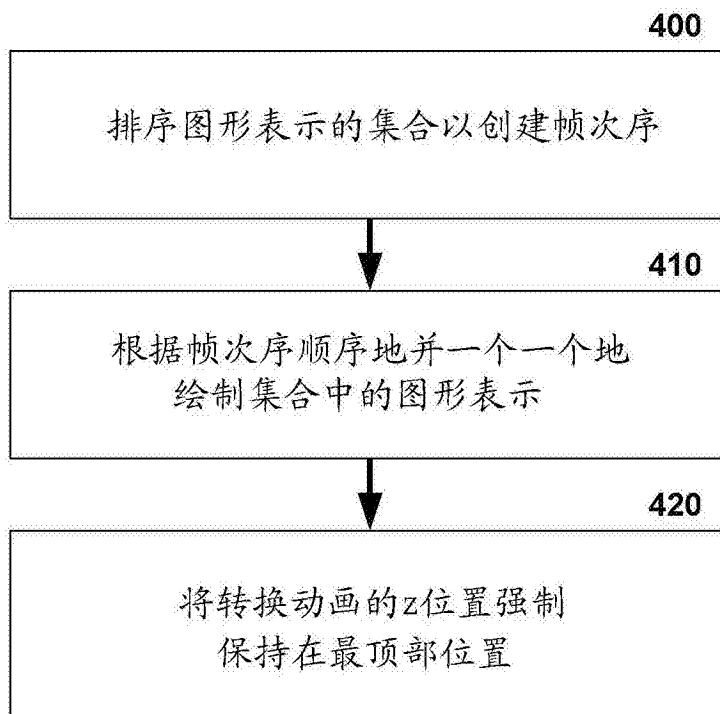


图 4

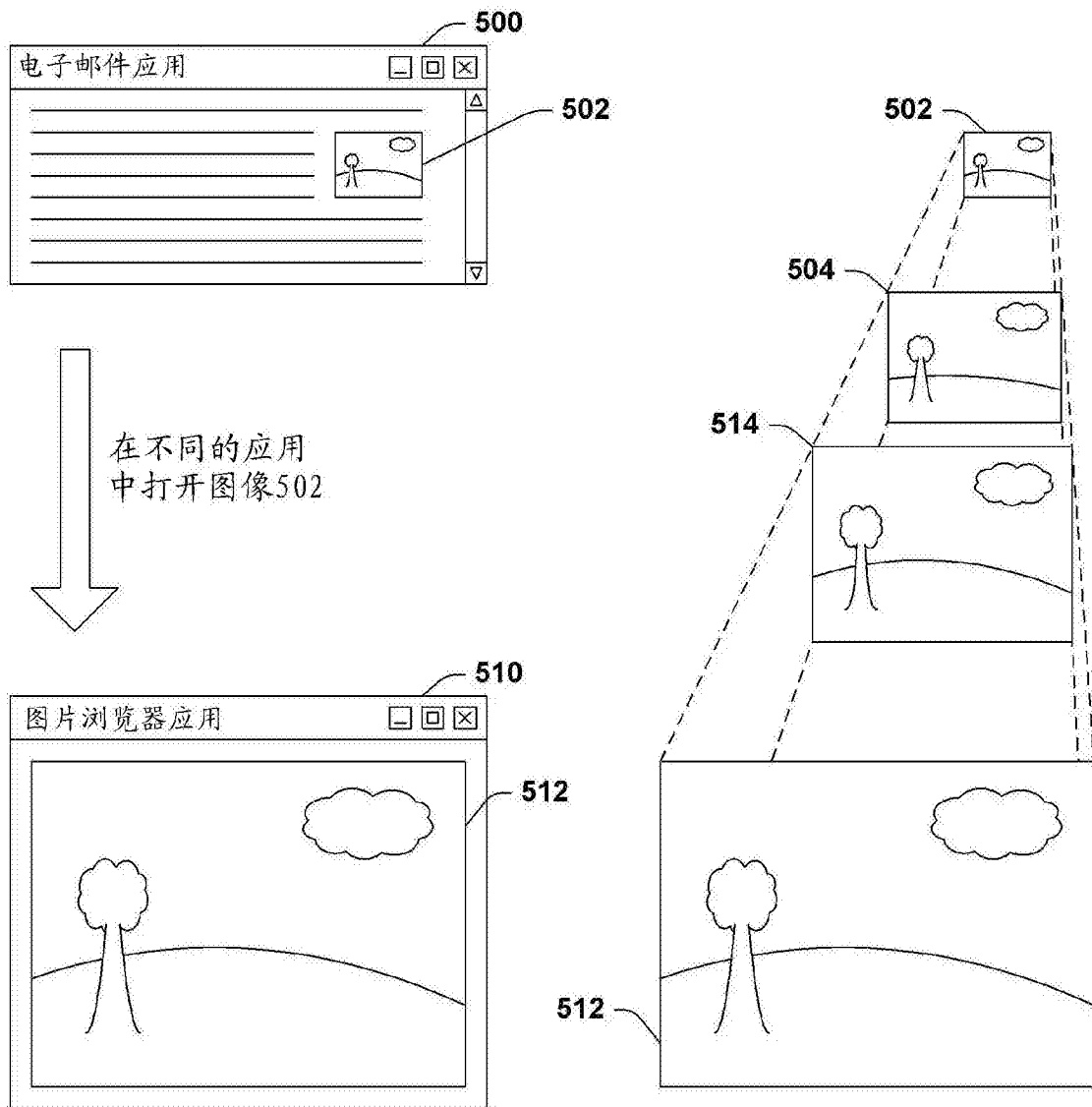


图 5

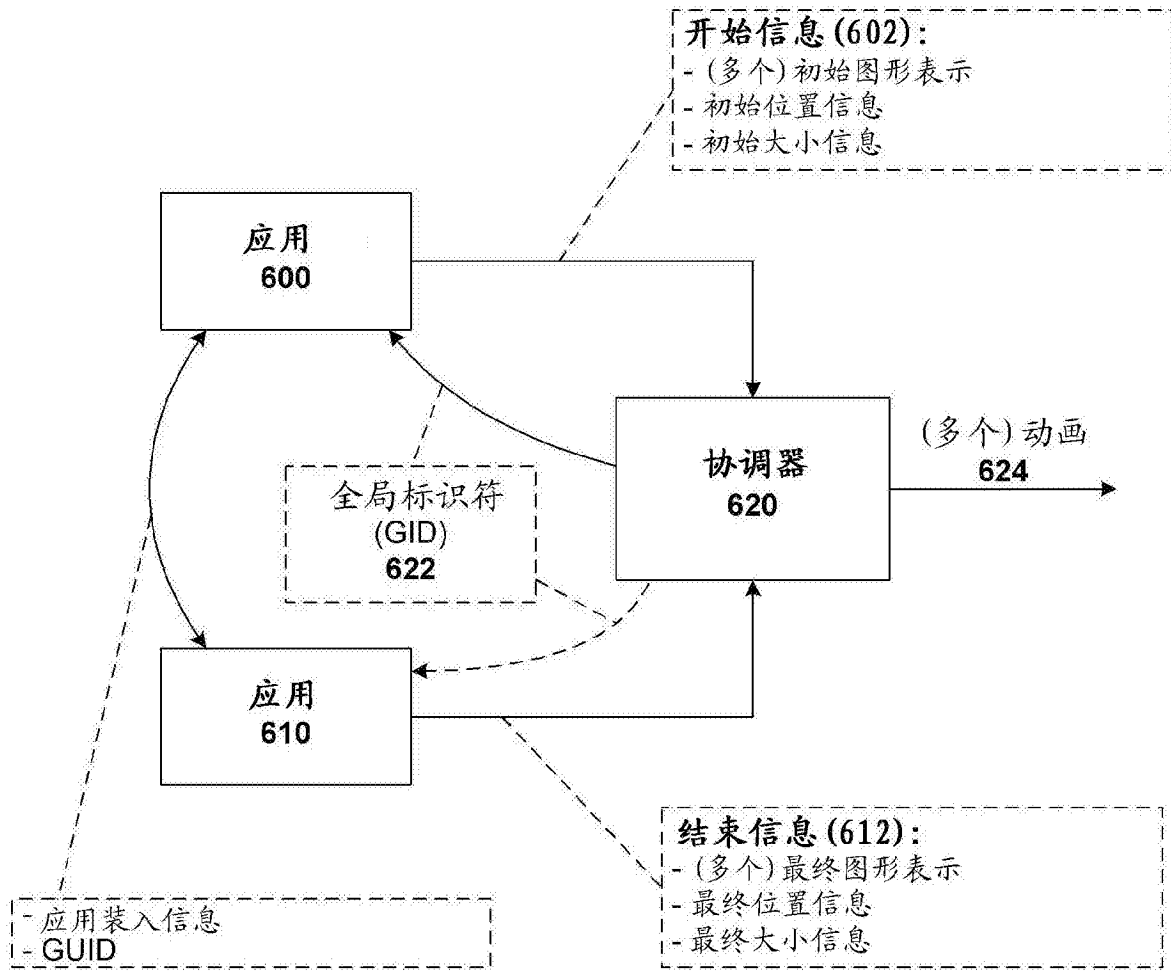


图 6

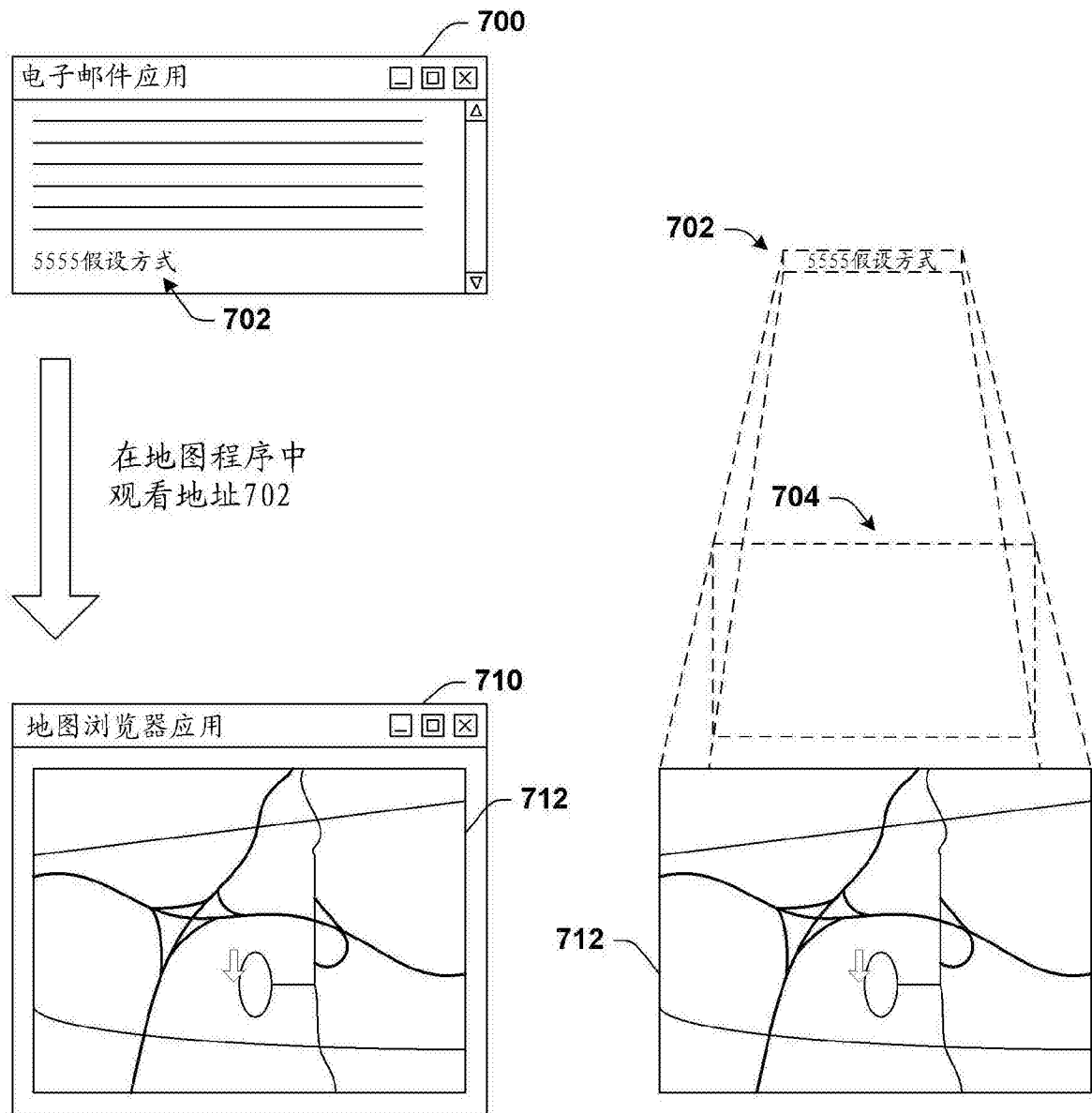


图 7

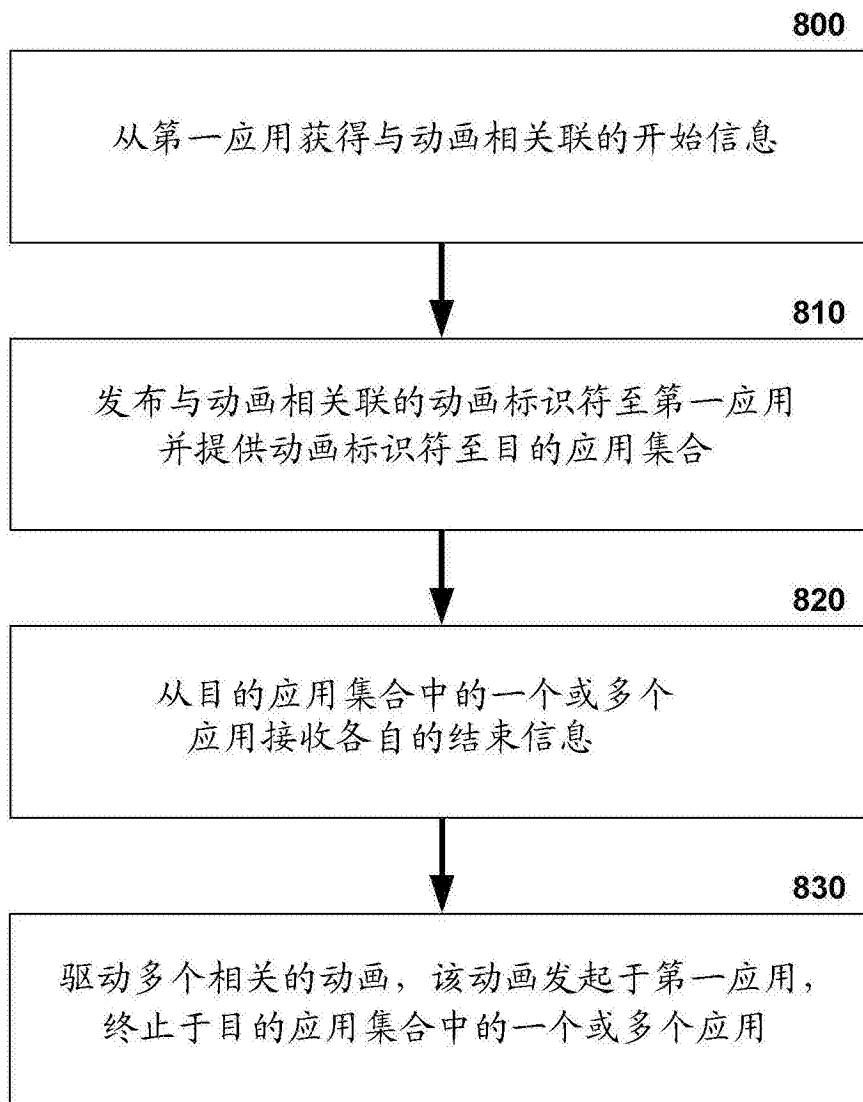


图 8

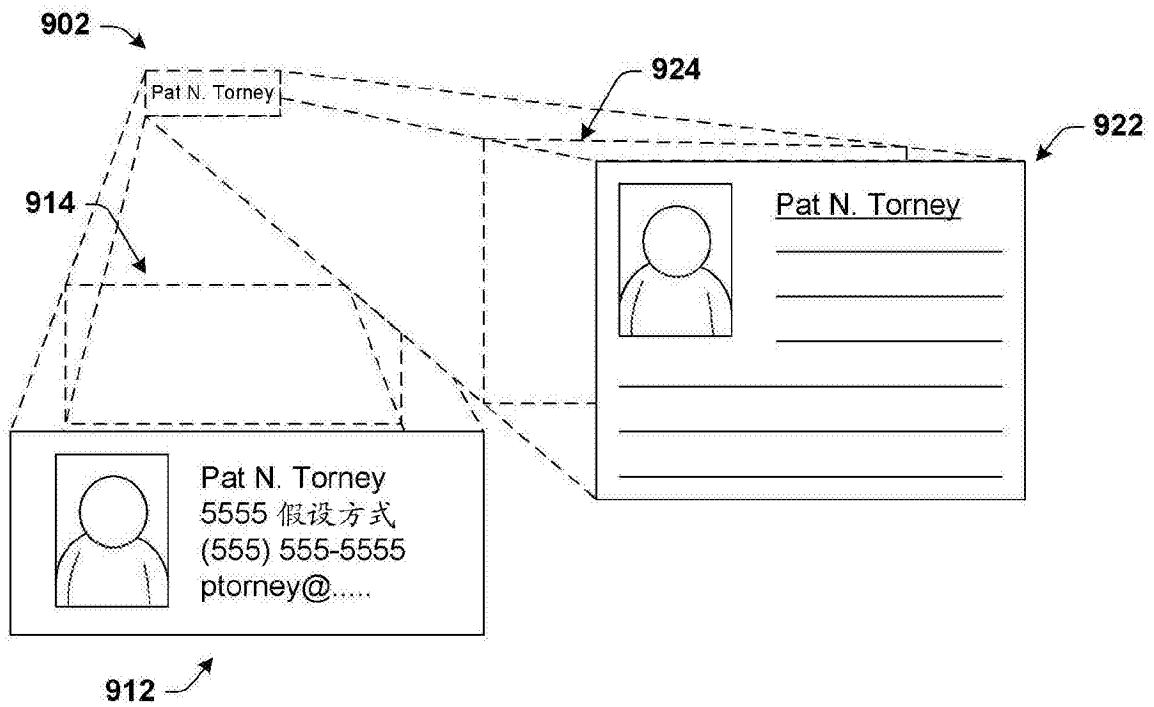
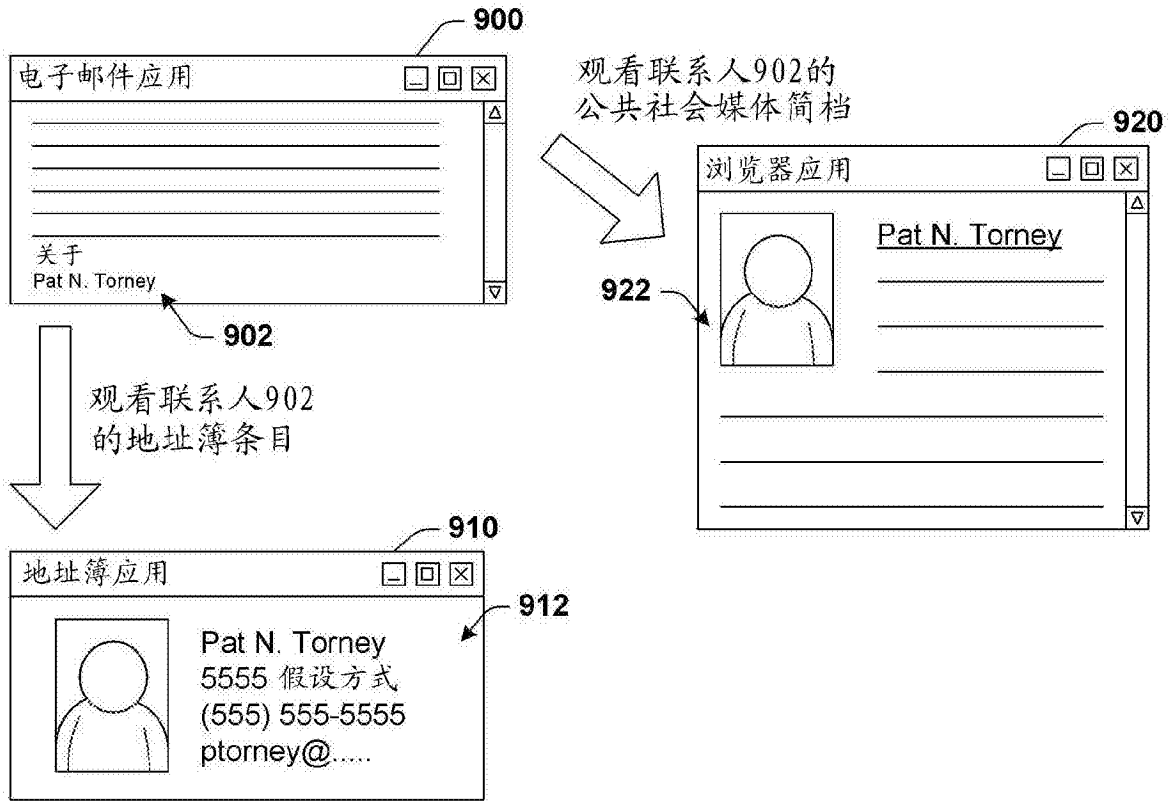


图 9

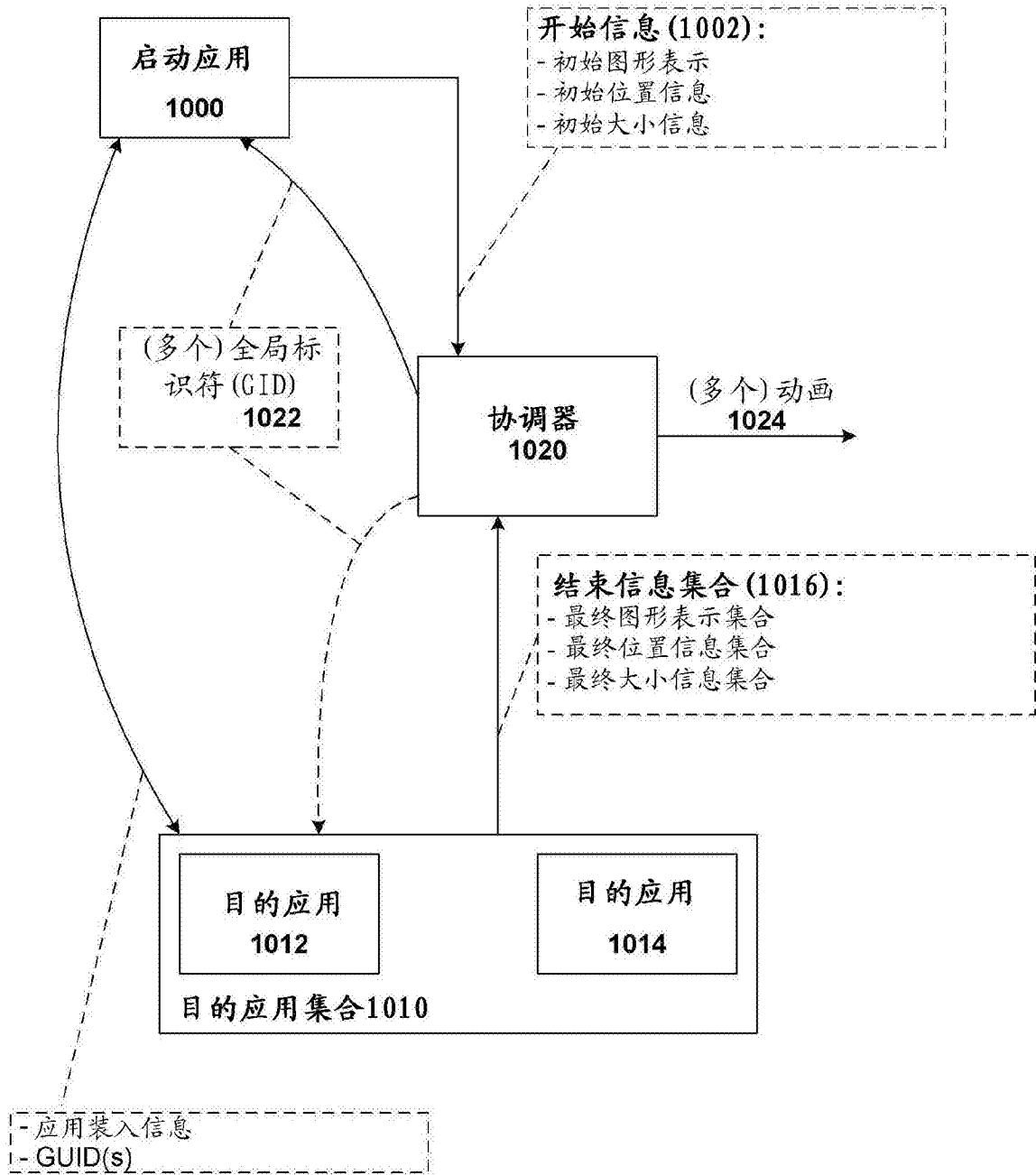


图 10

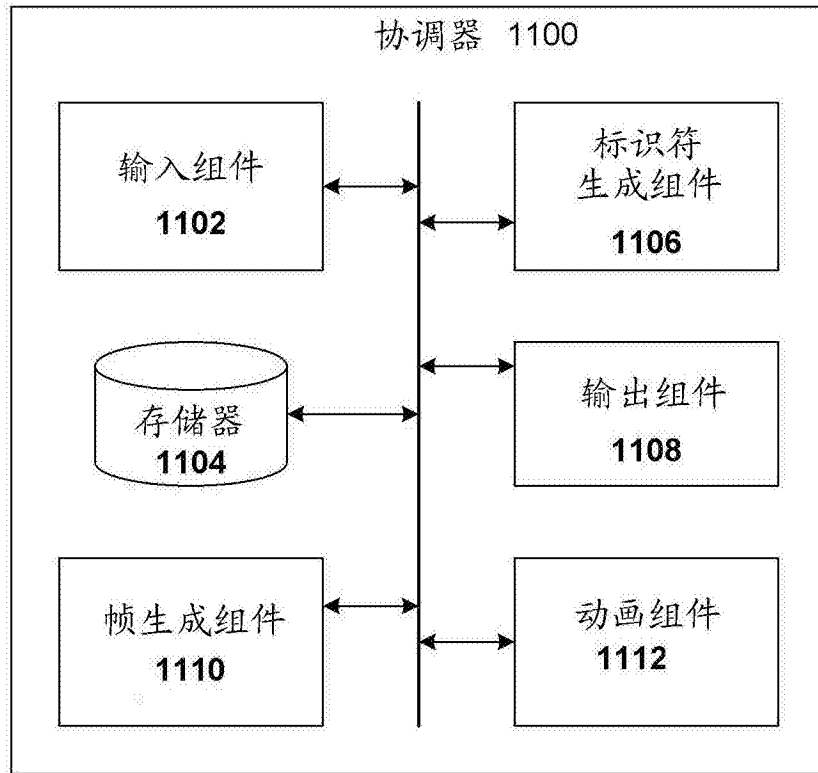


图 11

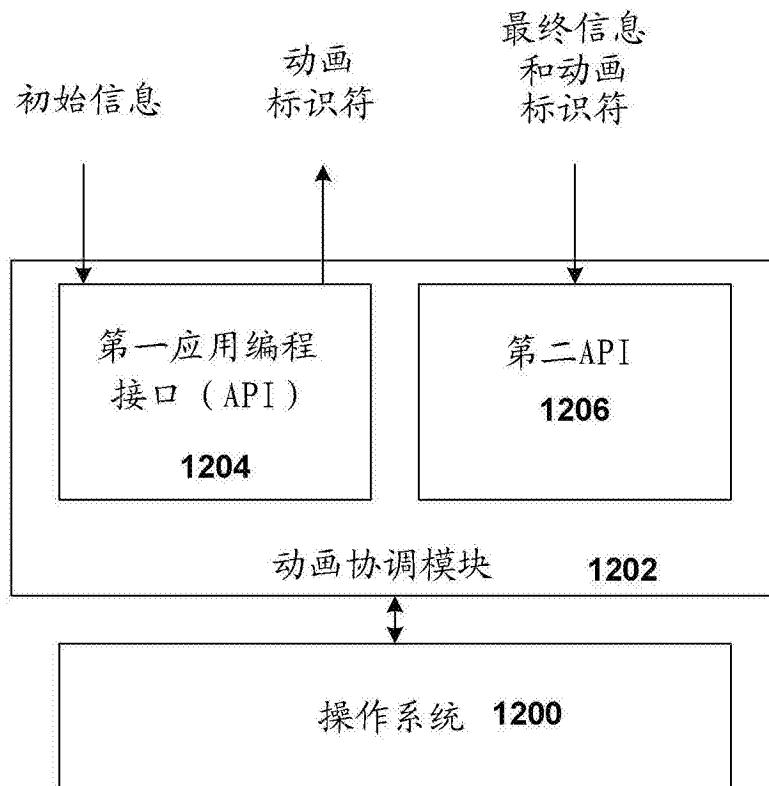


图 12

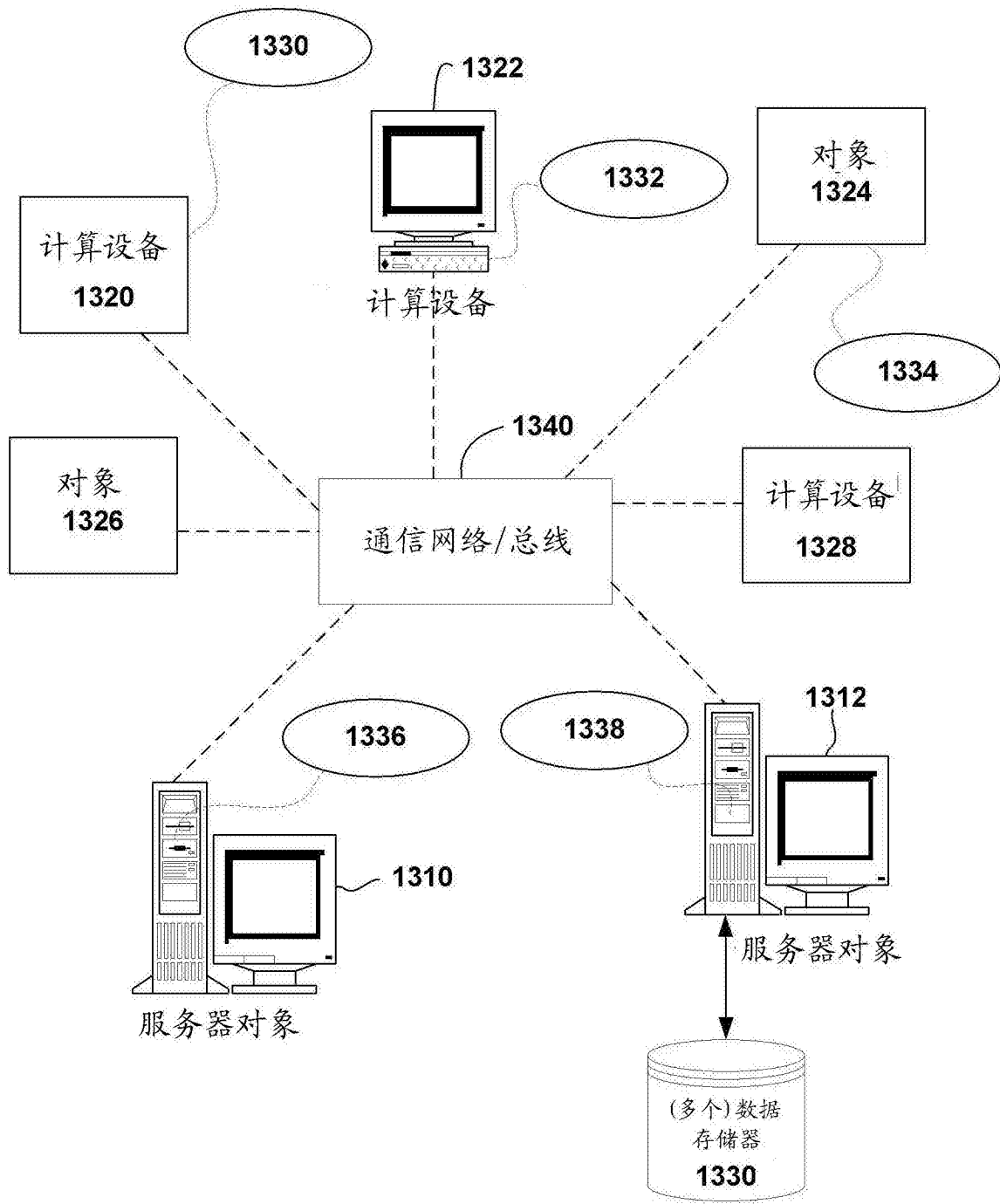


图 13

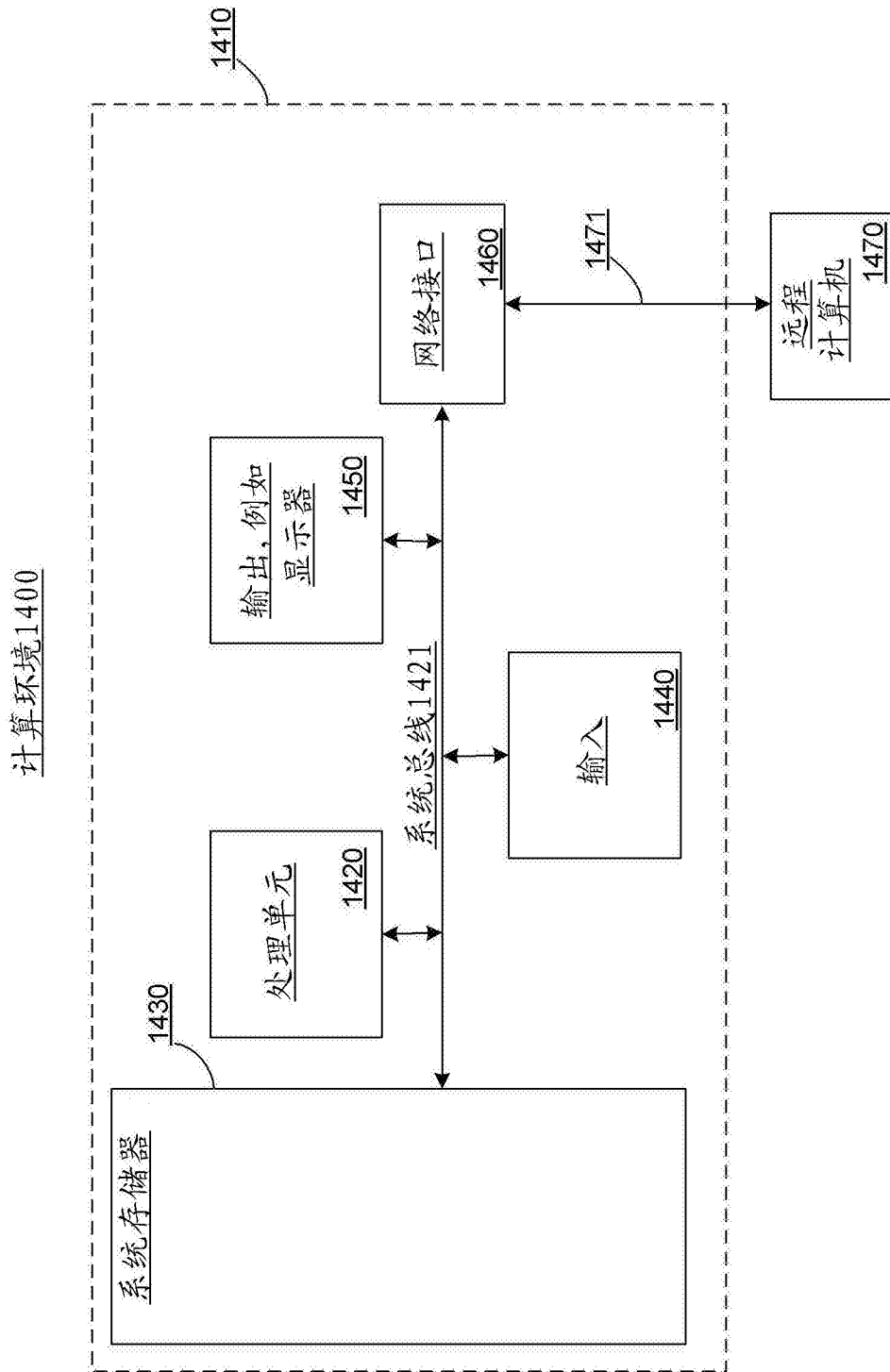


图 14