

[19] 中华人民共和国国家知识产权局

[51] Int. Cl.



[12] 发明专利申请公布说明书

G06T 1/20 (2006.01)
G06F 15/76 (2006.01)
G06F 17/00 (2006.01)

[21] 申请号 200780043739.9

[43] 公开日 2009年10月7日

[11] 公开号 CN 101553843A

[22] 申请日 2007.11.27

[21] 申请号 200780043739.9

[30] 优先权

[32] 2006.11.28 [33] US [31] 11/605,183

[86] 国际申请 PCT/US2007/085666 2007.11.27

[87] 国际公布 WO2008/067330 英 2008.6.5

[85] 进入国家阶段日期 2009.5.26

[71] 申请人 微软公司

地址 美国华盛顿州

[72] 发明人 K·伯格伦德 D·R·里赫恩鲍尔

G·D·谢克特 D·R·尼德

A·M·史密斯

[74] 专利代理机构 上海专利商标事务所有限公司

代理人 顾嘉运

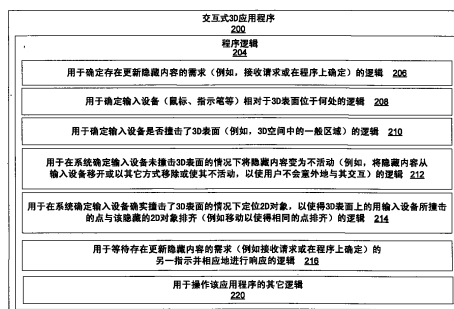
权利要求书 3 页 说明书 10 页 附图 9 页

[54] 发明名称

与 3D 表面上的 2D 内容进行交互

[57] 摘要

公开了启用与置于 3D 表面上的 2D 内容进行交互的各种技术和方法。系统确定输入设备相对于 3D 表面定位在何处。如果输入设备撞击 3D 表面，则定位 2D 中的隐藏内容以使表示在 3D 表面上撞击的区域的点与 2D 中的隐藏内容上的对应的点排齐。例如，在场景中的一位置处检测到输入设备时，在接收到对输入设备位置的请求时，3D 表面被投影到 2 维中。在所投影的 3D 表面上计算最接近输入设备的 2D 位置的点。响应于被用来定位隐藏内容，向该最接近的点提供 3D 表面的对应的点。



1. 一种具有用于使得计算机执行以下步骤的计算机可执行指令的计算机可读介质，所述步骤包括：

确定输入设备相对于 3D 表面定位在何处；以及

如果所述输入设备撞击所述 3D 表面，则定位 2D 中的隐藏内容以使表示在所述 3D 表面上撞击的区域的点与 2D 中的所述隐藏内容上的对应的点排齐。

2. 如权利要求 1 所述的计算机可读介质，其特征在于，还具有用于使得计算机执行以下步骤的计算机可执行指令，所述步骤包括：

如果所述输入设备未撞击所述 3D 表面，则使所述隐藏内容不活动。

3. 如权利要求 1 所述的计算机可读介质，其特征在于，还具有用于使得计算机执行以下步骤的计算机可执行指令，所述步骤包括：

在确定所述输入设备相对于所述 3D 表面位于何处之前，确定存在更新所述隐藏内容的需求。

4. 一种用于提供输入设备相对于 3D 表面的位置的方法，所述方法包括以下步骤：

在场景中的一位置处检测到输入设备时，接收对输入设备位置的请求；将 3D 表面投影到所述场景中的 2 维中；

计算所投影的 3D 表面上最接近所述输入设备的 2D 位置的点；以及响应于所述请求来返回所述最接近的点。

5. 如权利要求 4 所述的方法，其特征在于，所述输入设备是鼠标。

6. 如权利要求 4 所述的方法，其特征在于，所述输入设备是指示笔。

7. 如权利要求 4 所述的方法，其特征在于，所述请求是从一区域中接收的。

8. 如权利要求 7 所述的方法，其特征在于，所述区域是任意 3D 几何结构。

9. 一种具有用于使得计算机执行如权利要求 4 所述的步骤的计算机可执行指令的计算机可读介质。

10. 一种用于启用与置于 3D 表面上的 2D 内容的交互的方法，所述

方法包括以下步骤：

确定处于 3D 场景中的 2D 内的隐藏内容需要更新；

确定输入设备在所述 3D 场景中的位置；以及

如果所述 3D 场景中的 3D 表面不具有捕捉，则确定所述输入设备是否撞击所述 3D 场景中的 3D 表面，并且如果所述输入设备确实撞击了所述 3D 表面，则在 3D 三角形上使用纹理坐标来确定在 2D 中的隐藏内容上撞击了多个点中的什么点，并且将所述隐藏内容移到一位置以使得所述隐藏内容与所述 3D 表面上的对应的点排齐。

11. 如权利要求 10 所述的方法，其特征在于，还包括：

如果所述 3D 场景中的 3D 表面具有捕捉，则确定所述输入设备是否用捕捉内容撞击所述 3D 表面。

12. 如权利要求 11 所述的方法，其特征在于，还包括：

如果所述 3D 场景中的 3D 表面具有捕捉，并且如果确定所述输入设备用所述捕捉内容撞击了所述 3D 表面，则在所述 3D 三角形上使用纹理坐标来确定在 2D 中的所述隐藏内容上撞击了多个点中的什么点，并且将所述隐藏内容移到所述位置以使得所述隐藏内容与所述 3D 表面上的对应的点排齐。

13. 如权利要求 11 所述的方法，其特征在于，还包括：

如果所述 3D 场景中的 3D 表面具有捕捉，并且如果确定所述输入设备未用所述捕捉内容撞击所述 3D 表面，则计算所述捕捉内容的边界，寻找所述边界上最接近所述输入设备的位置的点，并将所述边界上最接近的点置于所述输入设备的位置的下方。

14. 如权利要求 10 所述的方法，其特征在于，在来自所述输入设备的正在移动事件发生时，确定所述隐藏内容需要更新。

15. 如权利要求 10 所述的方法，其特征在于，所述输入设备是鼠标。

16. 如权利要求 10 所述的方法，其特征在于，所述输入设备是指示笔。

17. 如权利要求 10 所述的方法，其特征在于，更新所述隐藏内容的需求是在接收到来自一区域的对所述输入设备的位置的请求时确定的。

18. 如权利要求 17 所述的方法，其特征在于，所述区域是任意 3D 几何结构。

19. 如权利要求 10 所述的方法，其特征在于，还包括：

如果所述 3D 场景中的 3D 表面不具有捕捉，并且如果确定所述输入设备未撞击所述 3D 场景中的 3D 表面，则将所述隐藏内容从所述输入设备的位置处移开。

20. 一种具有用于使得计算机执行如权利要求 10 所述的步骤的计算机可执行指令的计算机可读介质。

与 3D 表面上的 2D 内容进行交互

背景

在 2 维 (2D) 环境中, 系统通过简单地确定活动的 X 和 Y 坐标, 可以知道用户选择了什么区域或以其它方式与什么区域交互。然而, 在 3 维 (3D) 世界中, 寻找相对于 3D 表面上的交互式 2D 元素的 X/Y 坐标并不总是直截了当的。例如, 诸如用户界面等 2D 对象可被置于诸如球面等 3D 表面上。在这种 2D 对象被置于 3D 表面的情况下, 可能难以处理用户与该现在投影在 3D 中的 2D 对象的交互。

概述

公开了启用与置于 3D 表面上的 2D 内容进行交互的各种技术和方法。系统确定输入设备相对于 3D 表面定位在何处。如果输入设备撞击 3D 表面, 则定位 2D 中的隐藏内容以使表示在 3D 表面上撞击的区域的点与 2D 中的隐藏内容上的对应的点排齐。在一个实现中, 在场景中的一位置处检测到不在交互式 2D 内容的边界的上方的输入设备时, 在接收到对该输入设备位置的请求时, 3D 表面被投影到 2 维中。在所投影的 3D 表面上计算最接近输入设备的 2D 位置的点。响应于被用来定位隐藏内容, 向该最接近的点提供 3D 表面的对应的点。

在一个实现中, 取决于特定 3D 表面是否具有捕捉来遵循不同的过程。例如, 如果 3D 场景中的 3D 表面不具有捕捉, 并且如果输入设备撞击 3D 表面, 则在 3D 三角形上使用纹理坐标来确定在 2D 中的隐藏内容上撞击了什么点。隐藏内容随后被移动到一位置, 以使得隐藏内容与 3D 表面上的对应点排齐。类似地, 如果 3D 场景中的 3D 表面具有捕捉, 并且如果确定输入设备用该捕捉内容撞击 3D 表面, 则使用纹理坐标和先前所述的过程来排齐隐藏内容。

在另一实现中, 如果 3D 场景中的 3D 表面具有捕捉, 并且如果确定输入设备未用捕捉内容撞击 3D 表面, 则系统计算捕捉内容的边界, 寻找边界

上最接近输入设备的位置的点，并将边界上该最接近的点置于输入设备的位置的下方。

提供本概述以便以简化形式介绍将在以下详细描述中进一步描述的一些概念。该概述不旨在标识所要求保护的主题的关键特征或必要特征，也不旨在用于帮助确定所要求保护的主题的范围。

附图简述

图 1 是一个实现的计算机系统的图示。

图 2 是在图 1 的计算机系统上操作的一个实现的交互式 3D 应用程序的图示。

图 3 是图 1 的系统的实现的高级处理流程图。

图 4 是图 1 的系统的实现的处理流程图，其示出在用 3D 对象提供输入设备位置时所涉及各阶段。

图 5 是图 1 的系统的实现的处理流程图，其示出在启用与置于 3D 表面上的 2D 内容的交互时所涉及的更详细的阶段。

图 6 是图 1 的系统的实现的模拟图像，其示出在没有捕捉时隐藏内容的 2D 表示。

图 7 是图 1 的系统的实现的模拟图像，其示出在没有捕捉时与隐藏内容交互的 3D 表面。

图 8 是图 1 的系统的实现的模拟图像，其示出在没有捕捉时覆盖在 3D 表面上的 2D 表示。

图 9 是图 1 的系统的实现的模拟图像，其示出在存在捕捉时显示按钮和文本的 3D 表面。

图 10 是图 1 的系统的实现的模拟图像，其示出在存在捕捉时选择文本的一部分的图 9 所示的 3D 表面。

图 11 是图 1 的系统的实现的模拟图像，其示出其中预期输入设备与图 10 所示的 3D 表面的方位上的 2D 相关的最接近的边点。

图 12 是图 1 的系统的实现的模拟图像，其示出具有捕捉的 2D 文本框。

图 13 是图 1 的系统的—个实现的模拟图像，其示出获得图 12 的图像的边并将这些边投影回 2D 来以 2D 形式在 3D 上给出 2D 内容的轮廓。

详细描述

为促进对本发明的原理的理解，现将对图中所示的各实施例加以参考，同时也将用具体语言描述它们。不过，需要理解的是，并无意由此作范围上的限制。在所述实施例中的任何改变和进一步更改，以及在此所述的原理的进一步应用都可以预期将是本领域技术人员通常能想到的。

该系统可以在一般上下文中描述为提供与置于 3D 表面上的 2D 内容的交互的应用程序，但是该系统还用于除此之外的其它目的。在一个实现中，此处所描述的一种或多种技术可被实现为图形呈现程序内的特征，诸如包括在诸如 MICROSOFT® WINDOWS®等操作系统环境中或来自处理图形呈现的任何其它类型的程序或服务中的那些特征。在另一实现中，此处所描述的一种或多种技术被实现为处理允许 2D 内容与 3D 表面一起使用的其它应用程序的特征。

在一个实现中，系统通过使用隐藏的 2D 内容来提供与 3D 表面的交互。真实的交互式 2D 内容保持隐藏，但不隐藏该隐藏的 2D 内容的外观并将其置于 3D 上。该隐藏内容以截取用户与在 3D 表面上所呈现的内容的外观进行交互的尝试的方式来定位。此处所使用的术语“隐藏内容”旨在包括因其不可见、被定为不能被看到的大小、定位在另一对象之后等因而用户未注意到的 2D 内容。在另一实现中，在 2D 内容的任何部分请求输入设备的位置或请求捕捉时，该 2D 内容的 3D 表示被投影回 2D。该投影的内容的边界随后被用来确定如何响应来自所捕捉的 3D 表面的任何输入请求。此处所使用的术语“捕捉”意味着向 2D 内容请求通知输入设备状态改变的时候。

如图 1 所示，用于实现该系统的一个或多个部分的示例性计算机系统包括诸如计算设备 100 等计算设备。在其最基本的配置中，计算设备 100 通常包括至少一个处理单元 102 和存储器 104。取决于计算设备的确切配置和类型，存储器 104 可以是易失性的（如 RAM）、非易失性的（如 ROM、闪存等）或是两者的某种组合。该最基本配置在图 1 中由虚线 106 来例示。

另外，设备 100 还可具有附加的特征/功能。例如，设备 100 还可包含额外的存储（可移动和/或不可移动），其中包括但不限于磁盘、光盘或磁带。这样的附加存储在图 1 中由可移动存储 108 和不可移动存储 110 示出。计算机存储介质包括以用于存储诸如计算机可读指令、数据结构、程序模块或其它数据等信息的任何方法或技术来实现的易失性和非易失性、可移动和不可移动介质。存储器 104、可移动存储 108 和不可移动存储 110 都是计算机存储介质的示例。计算机存储介质包括但不限于，RAM、ROM、EEPROM、闪存或其它存储器技术、CD-ROM、数字多功能盘（DVD）或其它光存储、磁带盒、磁带、磁盘存储或其它磁存储设备、或者可用于存储所需信息并且可由设备 100 访问的任何其它介质。任何这样的计算机存储介质都可以是设备 100 的一部分。

计算设备 100 包括允许计算设备 100 与其它计算机/应用程序 115 进行通信的一个或多个通信连接 114。设备 100 还可以具有诸如键盘、鼠标、笔、语音输入设备、触摸输入设备等输入设备 112。还可以包括诸如显示器、扬声器、打印机等输出设备 111。这些设备在本领域中公知且无需在此处详细讨论。在一实现中，计算设备 100 包括交互式 3D 应用程序 200。交互式 3D 应用程序 200 将在图 2 中更详细地描述。

继续参照图 1，现转到图 2，其示出了运行于计算设备 100 上的交互式 3D 应用程序 200。交互式 3D 应用程序 200 是驻留在计算设备 100 上的应用程序之一。然而，可以理解，交互式 3D 应用程序 200 可另选地或另外地被具体化为一个或多个计算机上的计算机可执行指令和/或与图 1 所示的不同的变型。另选地或另外地，交互式 3D 应用程序 200 的一个或多个部分可以是系统存储器 104 的一部分、可以在其它计算机和/或应用程序 115 上、或可以是计算机软件领域的技术人员能想到的其它此类变型。

交互式 3D 应用程序 200 包括负责执行在此描述的一些或全部技术的程序逻辑 204。程序逻辑 204 包括用于确定存在更新隐藏内容的需求（例如，在接收请求时或在程序上确定）的逻辑 206；用于确定输入设备（例如鼠标、指示笔等）相对于 3D 表面位于何处的逻辑 208；用于确定输入设备是否撞击（hit）了 3D 表面的逻辑 210；用于在系统确定输入设备未撞击 3D 表面

的情况下将隐藏内容变为不活动（例如，将隐藏内容从输入设备移开或以其它方式移除或使其不活动，以使用户不会意外地与其交互）的逻辑 212；用于在系统确定输入设备确实撞击了 3D 表面的情况下定位 2D 对象，以使得 3D 表面上的用输入设备所撞击的点与该隐藏的 2D 对象排齐（例如移动以使得相同的点排齐）的逻辑 214；用于等待存在更新隐藏内容的需求（例如接收请求或在程序上确定）的另一指示并相应地进行响应的逻辑 216；以及用于操作该应用程序的其它逻辑 220。在一个实现中，程序逻辑 204 可用于通过编程，如使用对程序逻辑 204 中的一个过程的单一调用而从另一程序调用。

现在转向图 3-5，并继续参考图 1-2，更详细地描述了用于实现交互式 3D 应用程序 200 的一个或多个实现的各阶段。图 3 是交互式 3D 应用程序 200 的高级处理流程图。在一种形式中，图 3 的过程至少部分地在计算设备 100 的操作逻辑中实现。该过程在开始点 240 处开始，可任选地确定存在更新隐藏内容的需求（例如，在接收请求时或在程序上确定）（阶段 242）。系统确定输入设备（例如鼠标、指示笔等）相对于 3D 表面定位在何处（阶段 244）。如果输入设备未撞击（例如接触）3D 表面（例如 3D 空间中的一般区域）（判定点 246），则将隐藏内容变为不活动（例如，从输入设备移开或以其它方式移除或使其不活动，以使用户不会意外地与其交互）（阶段 248）。如果输入设备确实撞击了 3D 表面（判定点 246），则定位 2D 对象，以使得 3D 表面上的用输入设备所撞击的点与该隐藏的 2D 对象排齐（例如移动以使得相同的点排齐）（阶段 250）。系统可任选地等待存在更新隐藏内容的需求的另一指示并相应地作出响应（阶段 252）。该过程在结束点 256 处结束。

图 4 示出在提供输入设备相对于 3D 表面的位置时所涉及各阶段的一个实现。在一种形式中，图 4 的过程至少部分地在计算设备 100 的操作逻辑中实现。该过程在开始点 270 处开始，在场景的某处检测到输入设备时，接收（例如来自诸如任意 3D 几何结构、球等区域）对输入设备位置的请求或查询（阶段 272）。获得该 3D 表面并将其投影到 2 维中（阶段 274）。计算该投影上最接近于输入设备的 2D 位置的点（阶段 276）。投影对象上

最接近的点是响应于请求或查询所返回的位置(例如返回到 3D 空间中的请求对象)(阶段 278)。该过程在结束点 280 处结束。

图 5 示出在启用与置于 3D 表面上的 2D 内容的交互时所涉及的更详细的阶段的一个实现。在一种形式中,图 5 的过程至少部分地在计算设备 100 的操作逻辑中实现。该过程在开始点 310 处开始,可任选地确定存在更新隐藏内容的需求(“正在进行(on)”事件等)(阶段 312)。如果系统确定 3D 表面不具有捕捉(判定点 314),则执行撞击测试 3D 场景来确定输入设备相对于 3D 表面位于何处(阶段 316)。如果 3D 表面未被撞击(判定点 320),则将隐藏内容从输入设备移开(阶段 324)。如果 3D 表面被撞击(判定点 320),则在 3D 三角形上使用纹理坐标来寻找撞击了 2D 内容上的什么点(阶段 326)。2D 内容被置于隐藏层中,并且移动该隐藏层以使各点排齐(阶段 326)。

如果系统确定 3D 表面确实具有捕捉(判定点 314),则执行撞击测试 3D 场景来确定输入设备相对于 3D 表面位于何处(阶段 318)。系统确定 3D 表面是否被(例如由输入设备)用捕捉内容撞击(判定点 322)。如果是,则在 3D 三角形上使用纹理坐标来寻找撞击了 2D 内容上的什么点(阶段 326)。2D 内容被置于隐藏层中,并且移动该隐藏层以使各点排齐(阶段 326)。如果系统确定 3D 表面未被用捕捉内容撞击(判定点 322),则计算所捕捉的内容的边界(阶段 328)。定位边界上最接近于输入设备位置的点,并且将边界上该最接近的点置于输入设备位置下方(阶段 328)。该过程在结束点 330 处结束。

现转向图 6-13,使用模拟图像来更详细地示出图 3-5 的各阶段。在图 6-8 中,使用一些示例性模拟图像来示出 3D 表面不具有捕捉时的一些可能的场景。这些模拟图像和它们伴随的描述提供图 5 的和/或此处所描述的某些其它技术的阶段 314、316、320、324 和 326 的进一步图示。图 6 是图 1 的系统的实现的模拟图像 400,其示出没有捕捉时隐藏内容的 2D 表示。模拟图像 500 包含被映射到球体的内容。图 7 包含示出图 6 的图像 400 被映射到球体(例如 3D)的模拟图像 500。图 8 包含模拟图像 600,其示出如何对齐隐藏内容以使得 3D 表面上的、输入设备位于其上方的滑动条部分

与 2D 中的滑动条相同。点击输入设备将随后与拇指控件交互。因为维持了这一映射，所以 3D 表面被正确地通知输入设备何时进入并离开它们，以及其位于它们自身上方的什么部分。这创建能够与 3D 上的 2D 内容进行交互的结果。在一个实现中，输入设备移动作为隐藏内容需要更新的信号来跟踪。

现将使用一些非限制性示例来描述如何将 2D 内容映射到 3D 表面来达到图 6-8 所示的结果。在输入设备不在 3D 表面上方时，则隐藏层可被置于任何位置以使得输入设备不在其上方。在一个实现中，所需行为是 3D 表面上的 2D 内容不行动，如同输入设备在其上方一样，并且任何其它事件应当不影响它。将隐藏层远离输入设备放置使其不被告知移动或点击等。

为示例起见，假定所有 3D 表面都由三角形组成，并且所有三角形都具有与其相关联的纹理坐标。纹理坐标指定图像（纹理）的哪一部分应被显示在三角形上。例如，假定纹理坐标处于(0,0)到(1,1)的范围内，其中(0,0)是图像的左上角，而(1,1)是图像的右下角。则如果纹理坐标是(0,0)、(1,0)和(0,1)，则图像的左上半被显示在三角形上。此外，假定显示在 3D 表面上的 2D 内容可被表示为图像，并且该图像是应用于该图像的 3D 表面的纹理。例如，图 6 可被认为是纹理，并且纹理坐标是使其环绕球体的坐标，如图 7 所示。

现在，在输入设备在 3D 表面上方时，光线被射入到 3D 场景来查看其与该 3D 表面的什么部分相交。这可以用许多标准技术来完成。一旦系统知道相交的是什么，则可以确定三角形上的所撞击的点以及其纹理坐标。一旦确定了纹理坐标，由于还知道纹理，则系统可以从该纹理坐标映射到 2D 内容上的位置。该位置是 3D 表面上方的精确点。为正确定位，系统移动隐藏内容，以使得在先前部分中所计算的位置直接在输入设备位置的下方。3D 表面上方的点直接在隐藏内容上的同一位置的下方，其两者都直接在输入设备的下方。因此，如果用户点击或以其它方式从该位置输入，则他们将点击/输入隐藏内容上的和 3D 上的 2D 内容上的精确相同的点。同样，在输入设备移动时，由于这种定位，隐藏内容和其在 3D 上的 2D 表示都将被告知精确相同的点上方的输入设备移动。

现转向图 9-13，示出了一些示例性模拟图像来说明其中 3D 表面具有捕捉的一些可能的场景。这些模拟图像和它们伴随的描述提供图 5 的和/或此处所描述的某些其它技术的阶段 314、318、322、326 和 328 的进一步图示。在一个实现中，在 3D 上的 2D 元素得到捕捉时，正确的隐藏内容定位可能变得更复杂。作为一个示例，在 3D 中，由于 3D 到 2D 平面上的投影，输入设备的位置实际上对应于 3D 空间中的一条线。另外，具有捕捉的 3D 表面还可被映射到任何任意几何结构。因此，在输入设备位于 3D 表面上方时，撞击测试指示输入设备相对于 2D 视件位于何处。在其离开 3D 表面时，由于以上问题，不再有对该问题的直接回答：2D 点对应于 3D 线并且 2D 内容可以在任意几何结构上。同样，因为 3D 表面具有捕捉，所以其想要接收所有事件。之前，在不涉及捕捉时，系统只需要确保输入设备一直在正确的对象上方。现在，具有捕捉，则系统需要定位隐藏内容以使其处于相对于具有捕捉的对象的适当的位置。图 9-11 所示的模拟图像更详细地示出这一点。

在一个实现中，对该问题的一个可能的解决方案是将该 3D 问题还原回 2D。在正常的 2D 情况下，应用于该内容的变换可被用来将输入设备位置转换为该内容的局部坐标系。该经变换的位置随后使该内容知道输入设备相对于它处于什么位置。在 3D 中，由于几何结构和纹理坐标布局的多种方位，有时难以判断 3D 点位于 3D 上的 2D 内容的相关坐标系中的什么位置。在一个实现中，为近似这一点，在 3D 上的 2D 内容在被投影到屏幕空间后，计算其轮廓并随后基于该投影来定位输入设备。图 9-11 更详细地示出这一点。

图 9 的模拟图像 700 示出 3D 上的 2D 内容。图 10 上的模拟图像 750 示出已经选择了文本，并且输入设备被移动到离开该对象的点。图 11 示出具有文本框（即具有捕捉的对象）的轮廓的模拟图像 800。该轮廓随后被用来定位隐藏内容。

在轮廓可用之后，计算该轮廓上最接近输入设备位置的点，并随后轮廓上的该点被认为是被“撞击”的点，且其被置于输入设备位置的下方。在所示示例中，执行加亮直到图像 750 的中央的“T”。由于输入设备是按

最接近的边点放置的，所以交互趋向于像其在 2D 中一样行动，因为隐藏内容是基于输入设备最接近与 3D 上的 2D 内容上的位置来定位的。通过将隐藏内容放置在最接近的边点处，系统指示其预期输入设备相对于 3D 表面的方位上的 2D 位于何处。

为实际执行参考图 9-11 所述的过程，系统计算具有捕捉的对象的边界，该捕捉与该对象被包含在内的 2D 内容相关。作为示例，考虑图 12-13 所示的 2D 内容。假定文本框具有捕捉。在图 12 中，图像 900 中包含的文本框的边界被以粗体划出轮廓。这些边界可被转换为纹理坐标，因为具有捕捉的 3D 表面的边界是已知的并且 2D 内容作为整体的大小也是已知的。使用纹理坐标，系统随后可以检查 3D 表面位于其上的网格的每一个三角形，并寻找包含与边界坐标相交的纹理坐标的那些三角形。例如，假定存在一个三角形，并且具有纹理坐标的三角形如图 12 中的 2D 内容上所示的一样被绘出。系统检查来查看该三角形的边是否与具有捕捉的 3D 表面的边界相交，在这一情况下它们确实相交（它们与具有捕捉的文本框相交）。如果三角形面向查看者，并且边界边的任一个与其相交，则边界边和三角形相交的边被添加到最终列表。将被添加的边在图 13 的图像 950 中示出。通过执行这些步骤，确定与所捕捉的 3D 表面的边界相交的可视边。

在一个实现中，系统还跟踪哪些三角形面向查看者而哪些背向。如果两个三角形共享一条边，一个面向用户而一个背向用户，则系统还可以将该共享边的在所捕捉的 3D 表面的边界内的部分添加到最终列表。这可能是必要的，以便计算可视边界。作为这种情况的非限制性示例，考虑图 9 中的球体。左边和右边都是廓影边（即，边既具有可视三角形又具有不可视三角形）。系统添加这些来计算所捕捉的 3D 表面的整个可视轮廓（如图 11 所示）。否则，最左侧和最右侧图像将丢失，并且将不计算完全轮廓。一旦确定了边的列表，则随后将它们投影回 2D。这以 2D 形式给出 3D 上的 2D 内容的轮廓。随后，计算这些边上最接近输入设备的点。该点具有其自己的纹理坐标，并且该纹理坐标如上被用来定位隐藏内容。取决于所需行为，可以加粗所捕捉的 3D 表面的边界，如果有必要进一步从所捕捉的 3D 表面移开。

尽管用对结构特征和/或方法动作专用的语言描述了本主题，但可以理解，所附权利要求书中定义的主题不必限于上述具体特征或动作。相反，上述具体特征和动作是作为实现权利要求的示例形式公开的。落入在此所述和/或所附权利要求所描述的实现的精力的范围内的所有等效方案、更改和修正都期望受到保护。

例如，计算机软件领域普通技术人员会认识到客户机和/或服务器布置、用户界面屏幕内容、和/或在此讨论的示例中所述的数据布局可在一台或多台计算机上不同地组织，以包括比示例中所描绘的更少或更多的选项或特征。

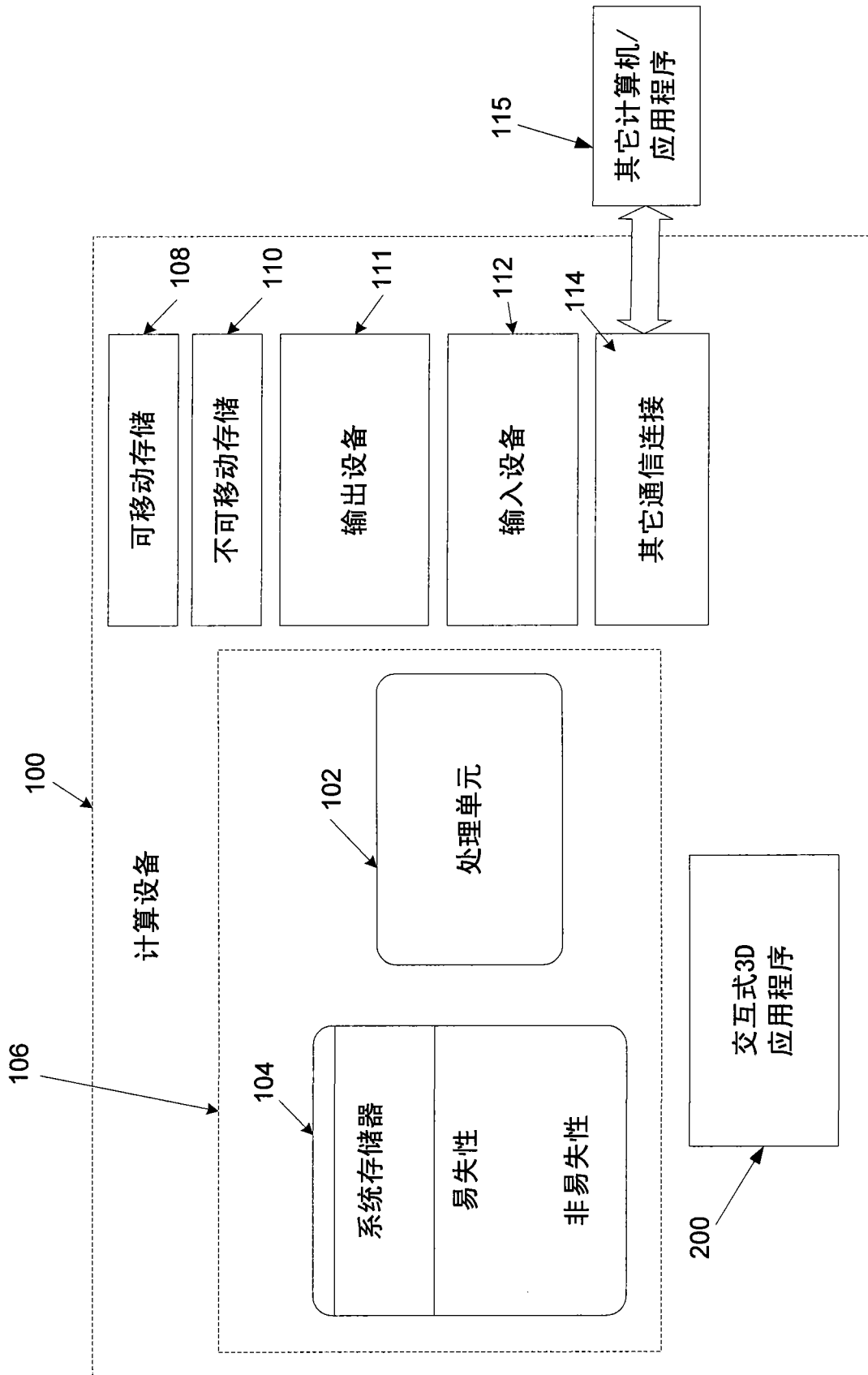


图 1

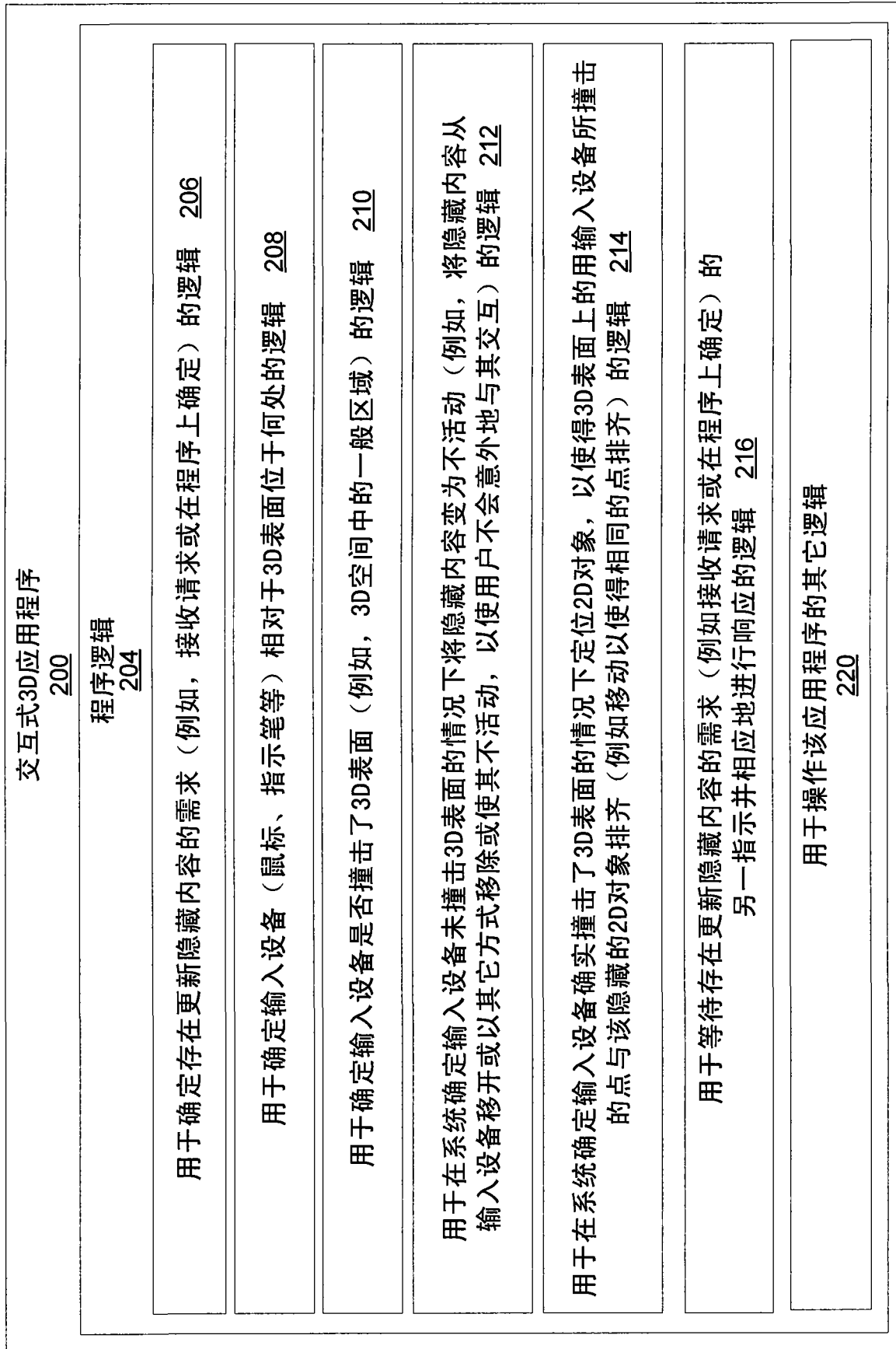


图 2

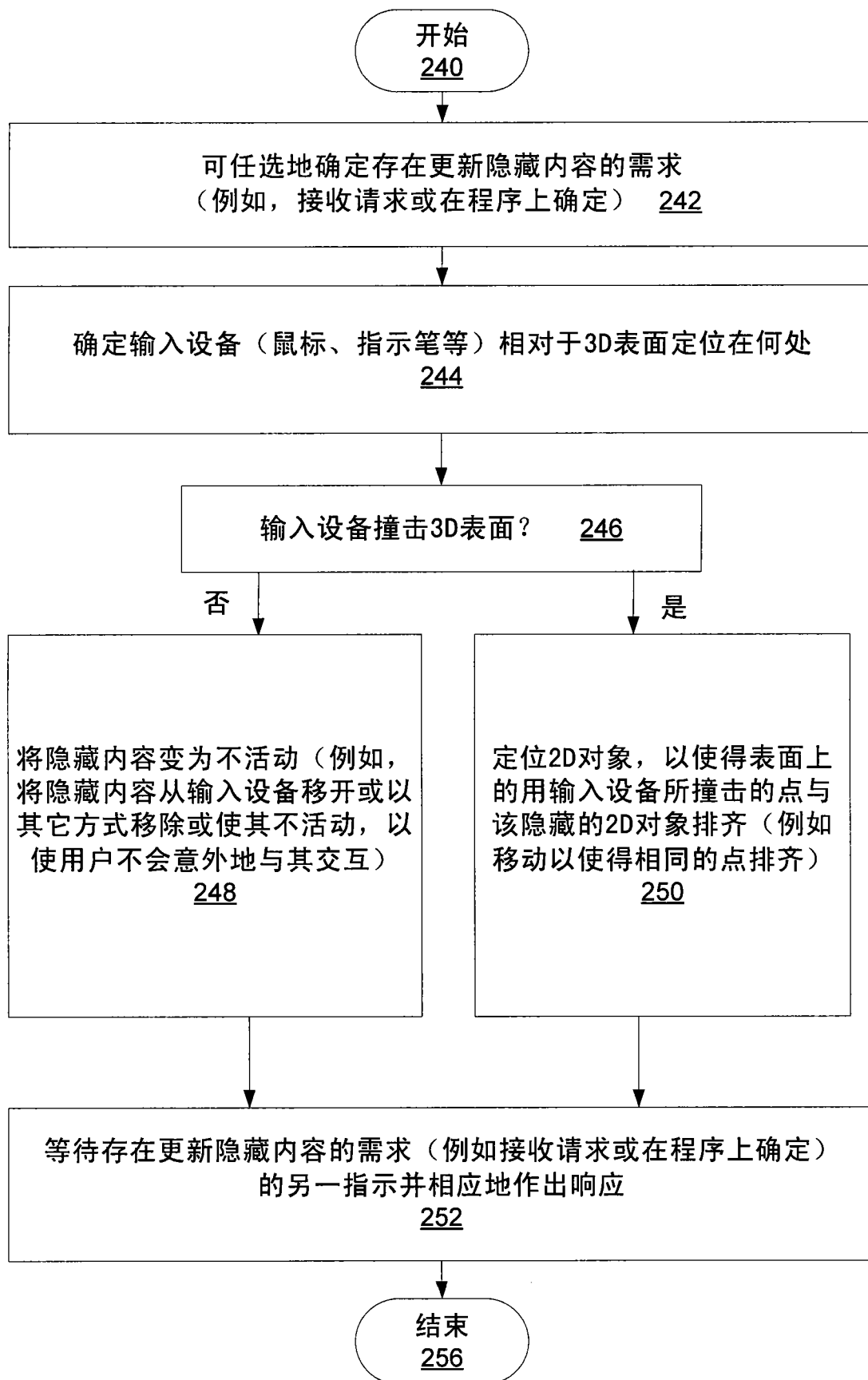


图 3

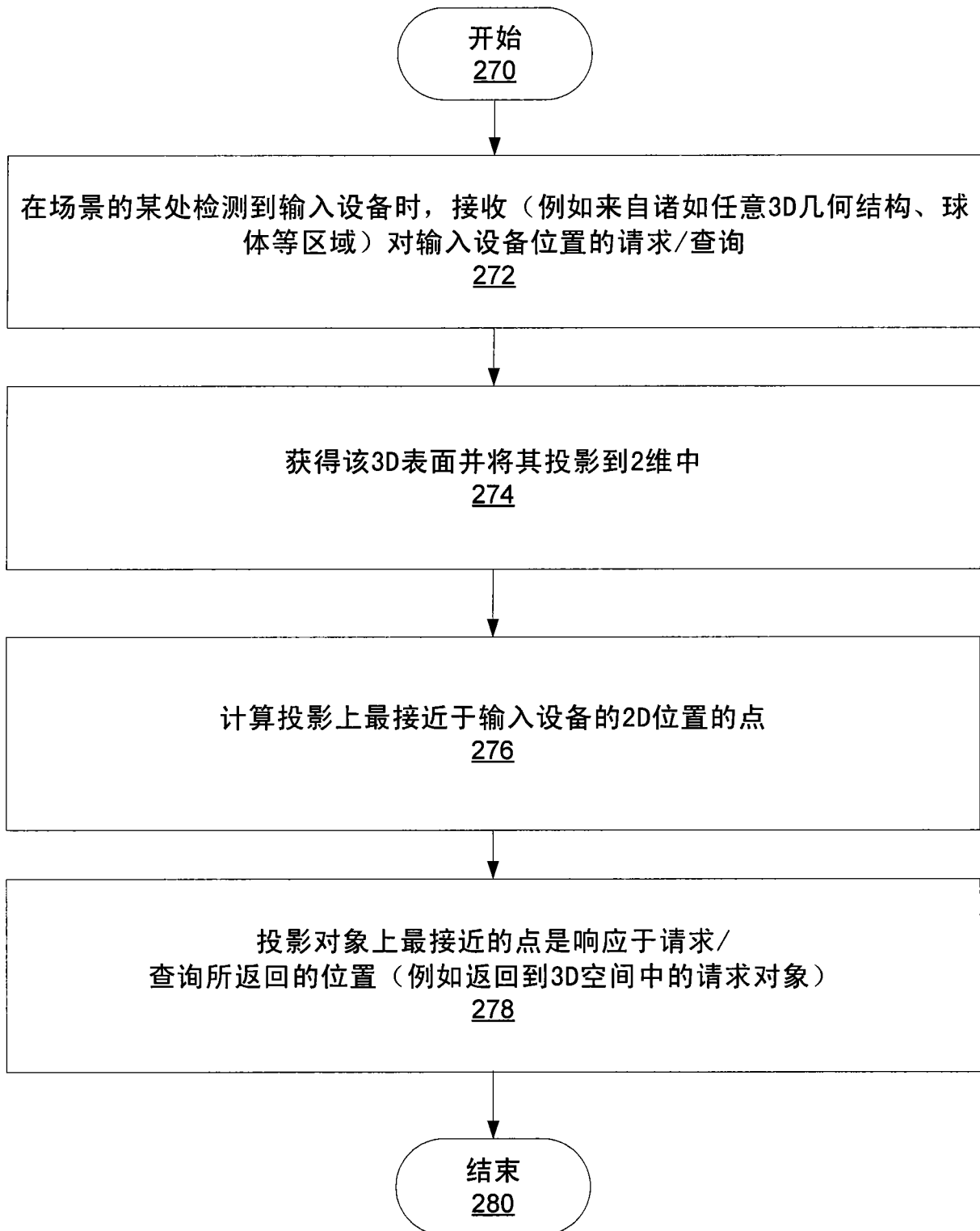


图 4

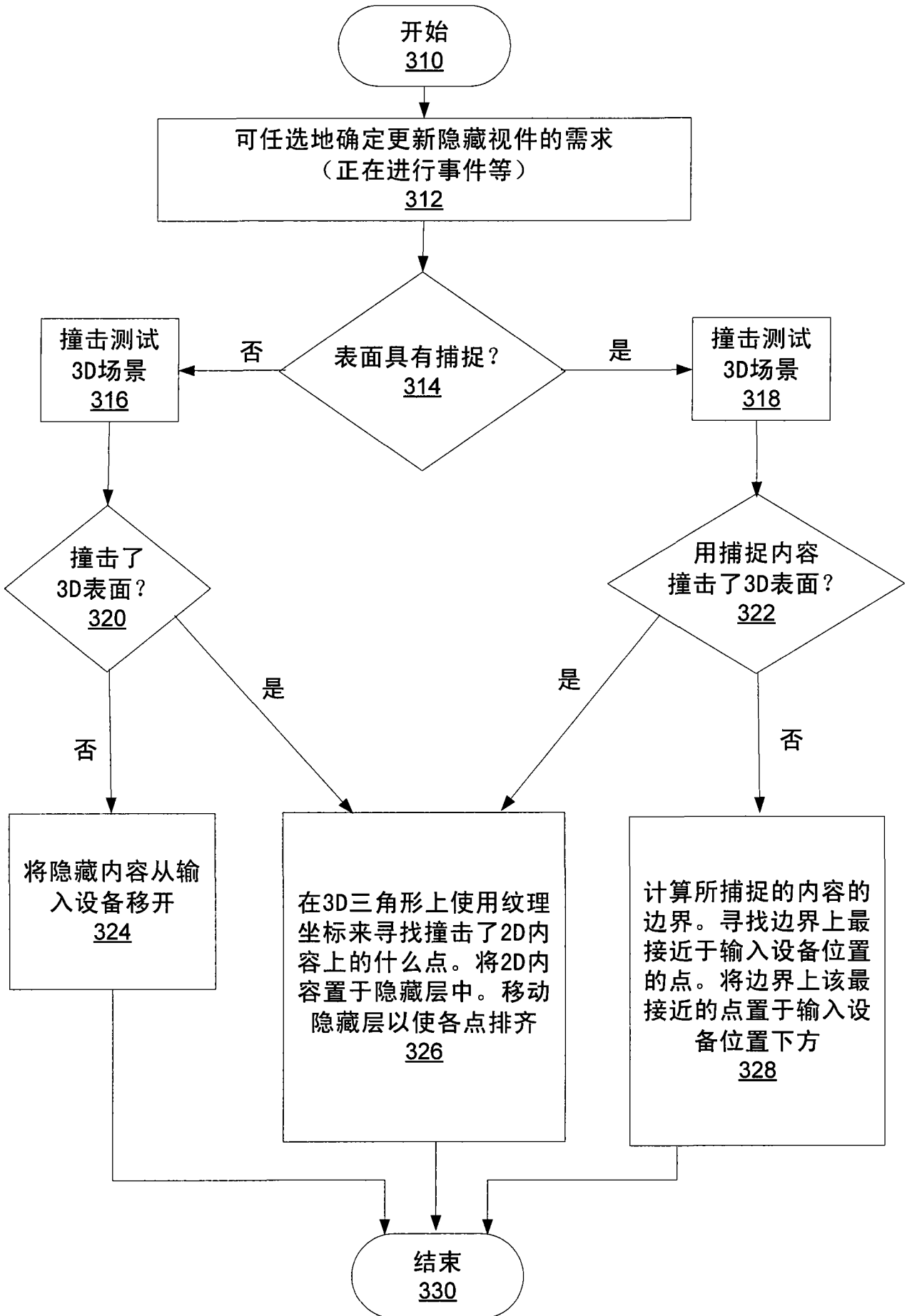


图 5

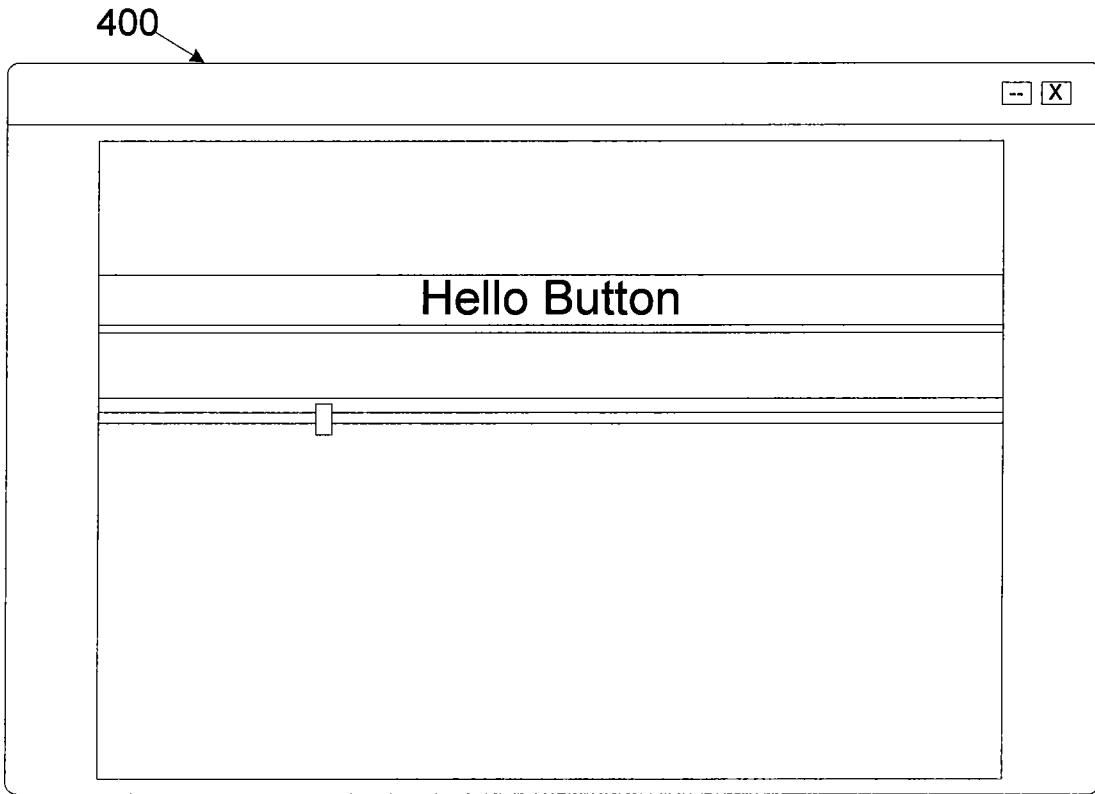


图 6

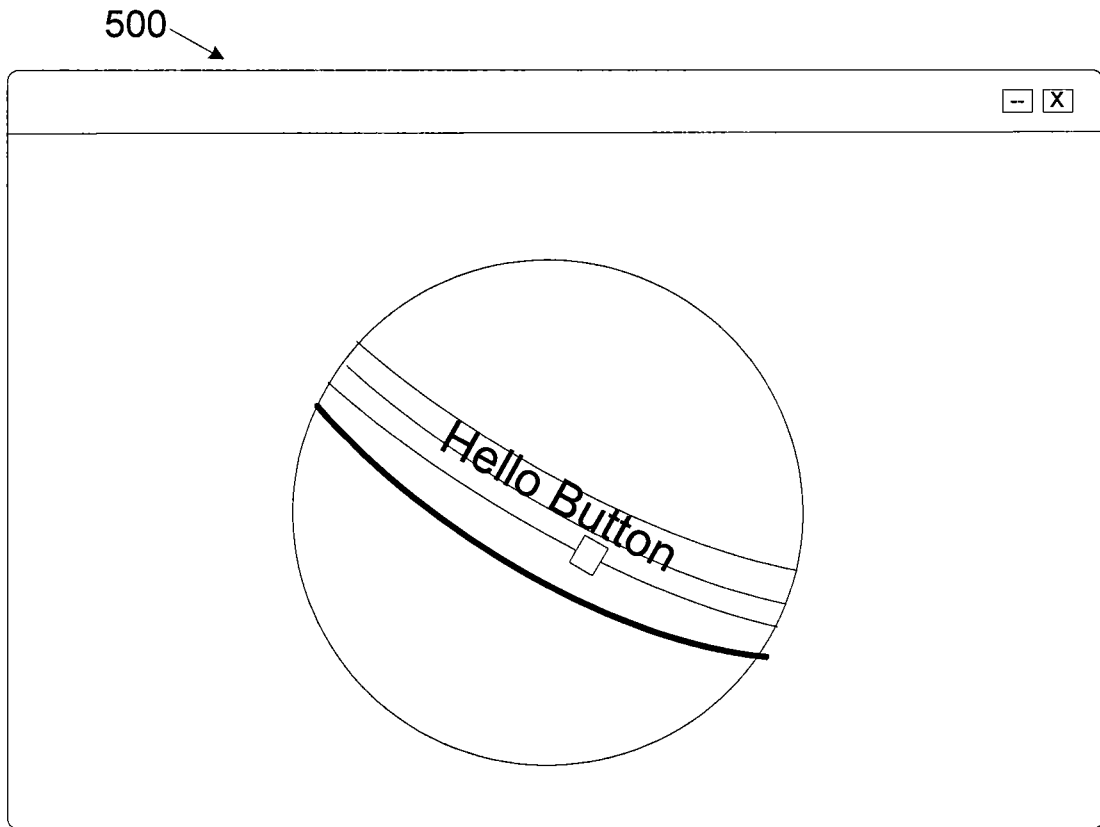


图 7

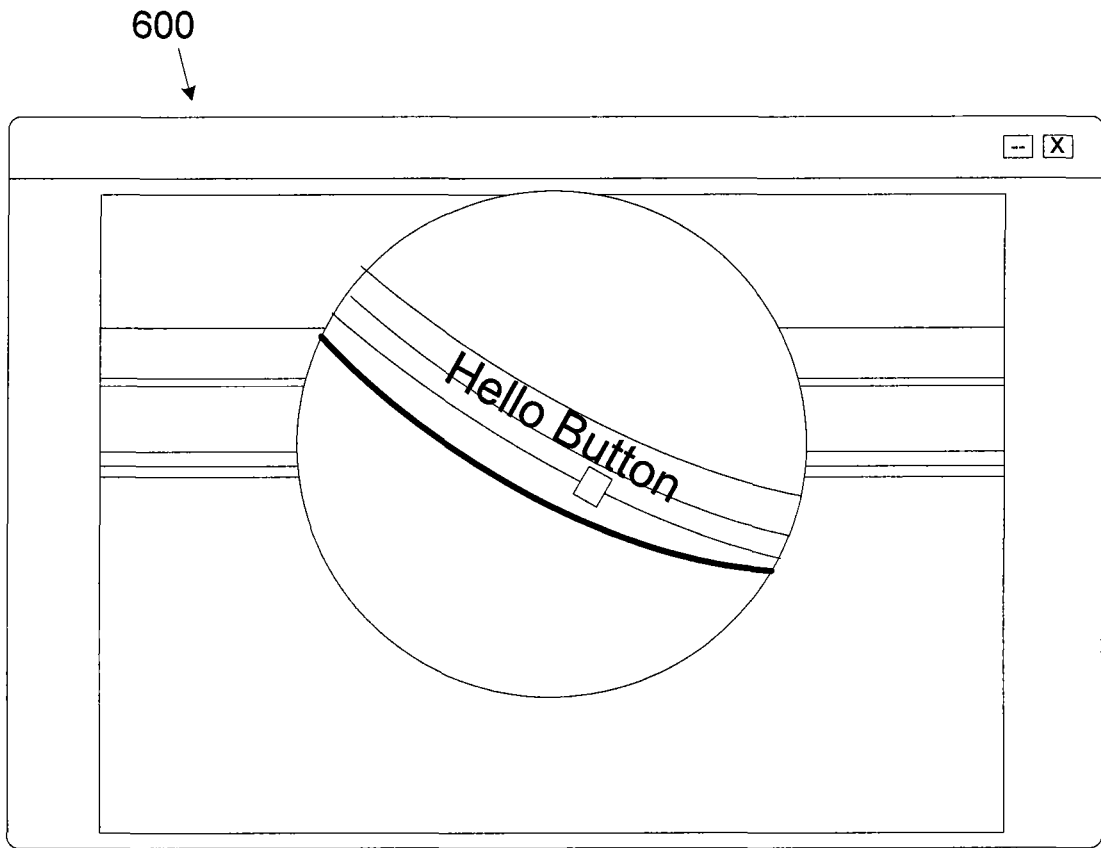


图 8

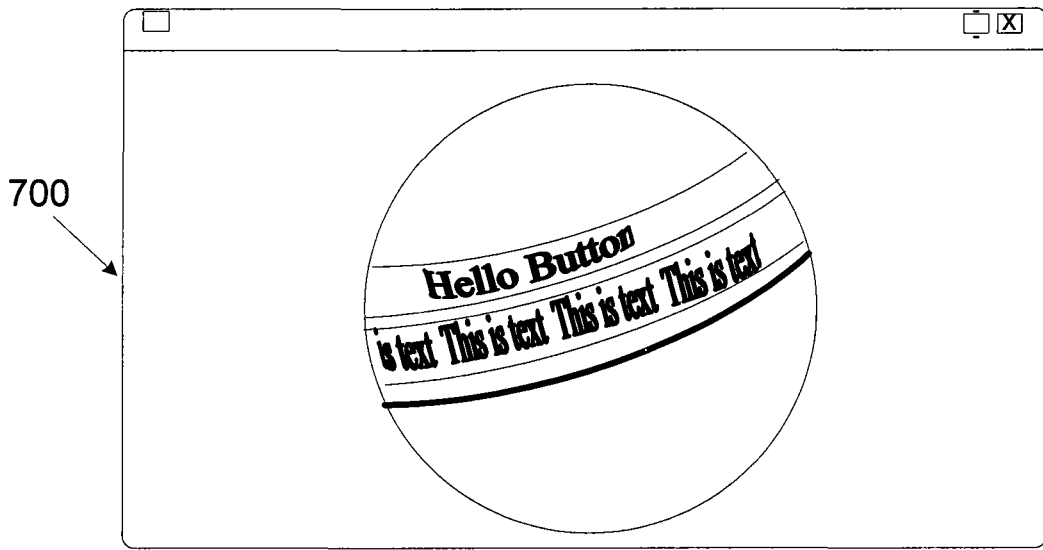


图 9

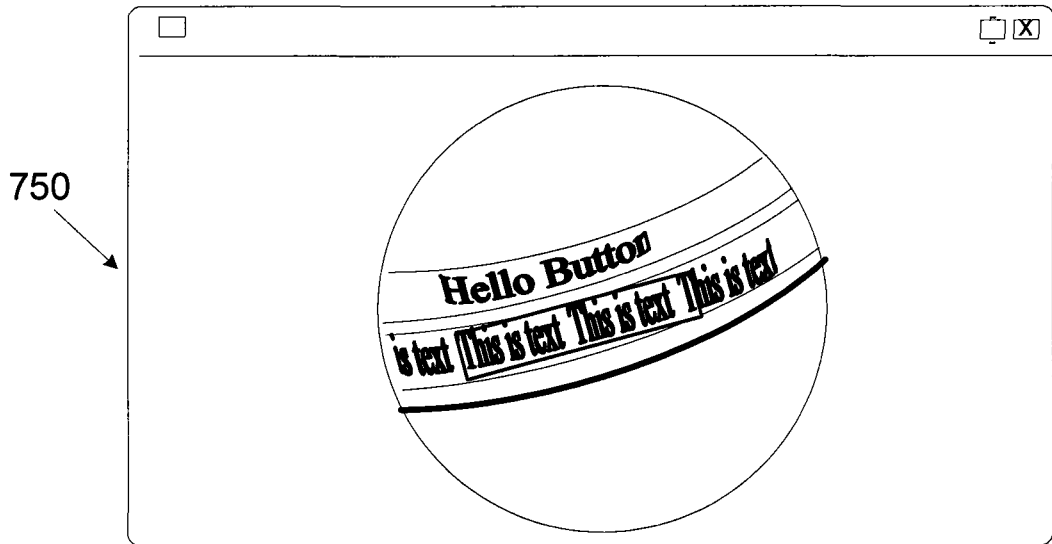


图 10

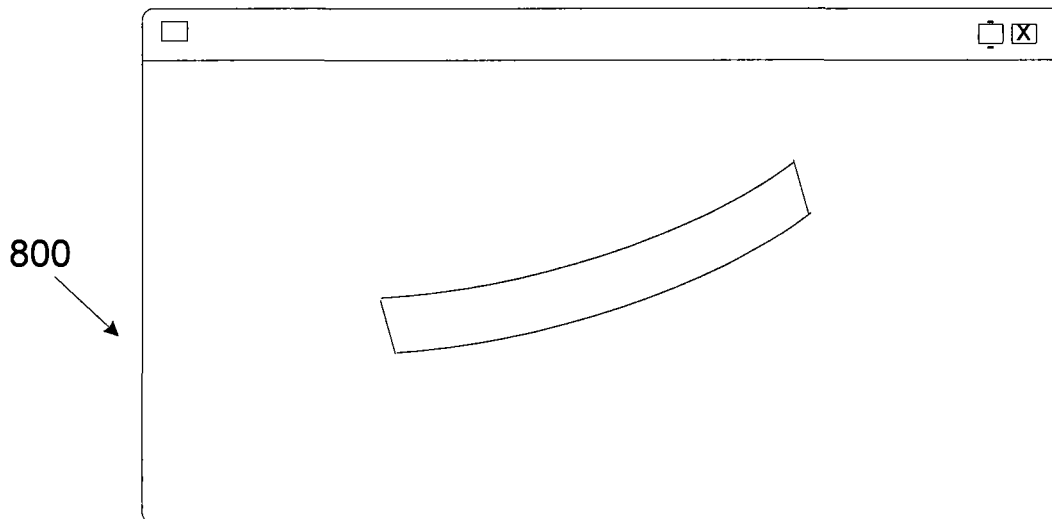


图 11

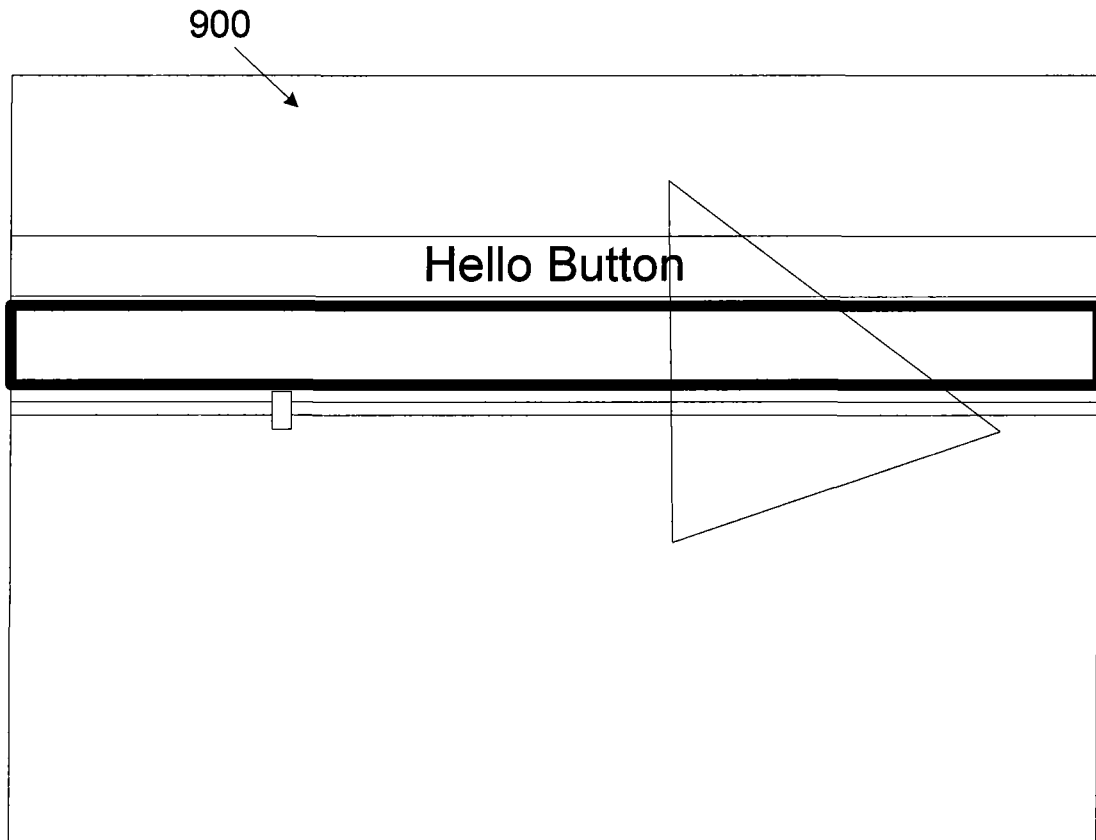


图 12

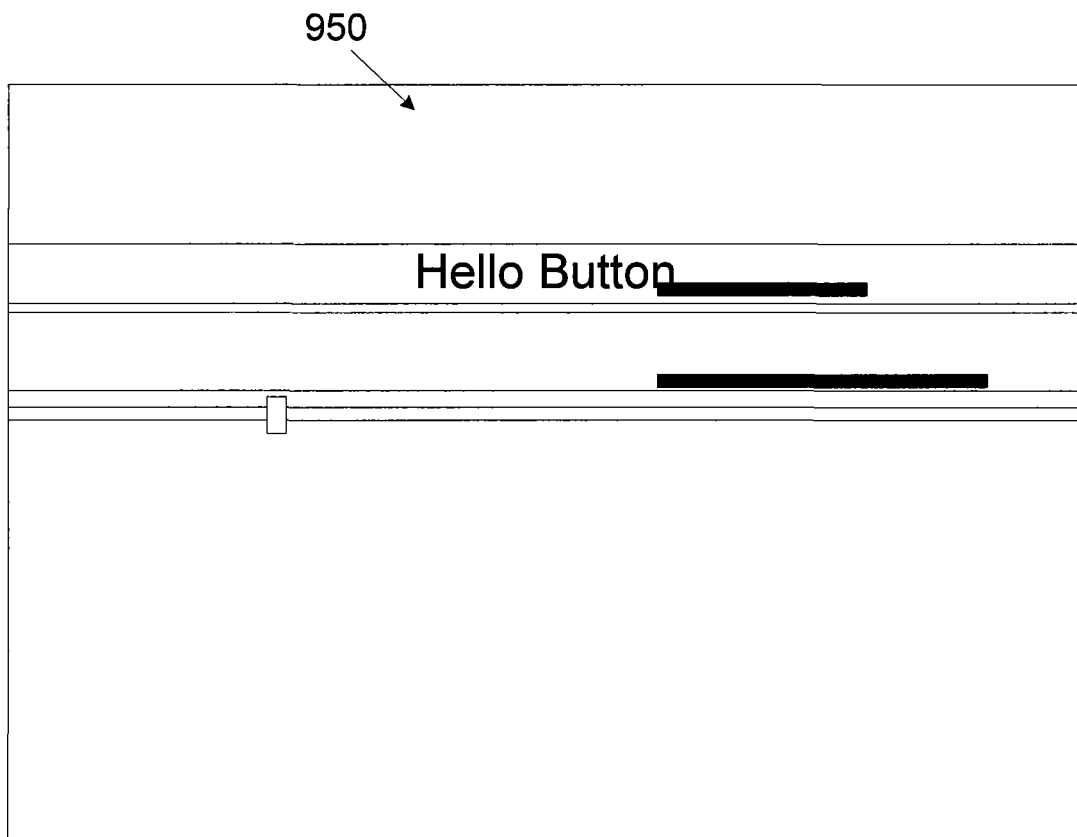


图 13