

[19] 中华人民共和国国家知识产权局

[51] Int. Cl.

G06F 3/033 (2006.01)

A63H 33/04 (2006.01)



[12] 发明专利申请公开说明书

[21] 申请号 200480013672.0

[43] 公开日 2006年6月21日

[11] 公开号 CN 1791850A

[22] 申请日 2004.5.13

[21] 申请号 200480013672.0

[30] 优先权

[32] 2003.5.20 [33] DK [31] PA200300759

[86] 国际申请 PCT/DK2004/000341 2004.5.13

[87] 国际公布 WO2004/104811 英 2004.12.2

[85] 进入国家阶段日期 2005.11.18

[71] 申请人 英特莱格公司

地址 瑞士巴尔

[72] 发明人 龙尼·谢勒 奥尔加·提姆森科
内奥米·克拉克 彼得·阿克

[74] 专利代理机构 中国国际贸易促进委员会专利商
标事务所
代理人 董 莘

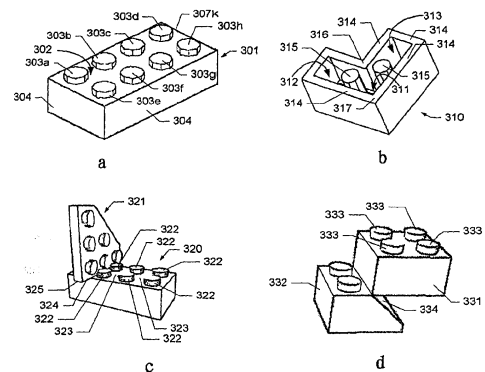
权利要求书 4 页 说明书 22 页 附图 10 页

[54] 发明名称

用于操纵三维对象的数字表示的方法和系统

[57] 摘要

依靠二维光标移动来操纵三维虚拟积木模型的方法，虚拟积木模型包括许多虚拟积木式组件，每块虚拟积木式组件包括许多用于根据一组连接规则连接该虚拟积木式组件与另一虚拟积木式组件的连接元件，该方法包括依靠光标在代表该模型二维投影的计算机显示区域中的移动来定位要被连接到结构的第一虚拟积木式组件的二维投影，得到该虚拟积木式组件的二维位置；由该二维位置确定第一虚拟积木式组件在三维坐标系中的许多候选位置；根据连接规则和一组预定的位置分级规则来选择候选位置其中之一；以及在选定候选位置上将第一积木式组件连接到结构。



1. 一种由计算机实现的、借助于二维光标移动来操纵三维虚拟积木模型的方法，所述虚拟积木模型包括多个虚拟积木式组件，其中
5 每个虚拟积木式组件包括许多用于根据一组连接规则连接所述虚拟积木式组件和另一虚拟积木式组件的连接元件，所述方法包括：
- 提供结构在三维坐标系中的数字表示，其中所述结构包括许多所述虚拟积木式组件；
 - 借助于光标在代表所述结构的投影的二维计算机显示区域中的
10 移动来定位要被连接到所述结构的第一虚拟积木式组件的二维投影，从而产生二维位置坐标；
 - 根据所述二维位置坐标确定所述第一虚拟积木式组件在所述三维坐标系中的许多三维候选位置；
 - 基于所述连接规则和一组预定的位置分级规则来选择所述候选
15 位置中的一个；以及
 - 在所述选定候选位置上将所述第一积木式组件连接到所述结构。
2. 根据权利要求1所述的方法，其中所述确定许多三维候选位置的步骤进一步包括确定所述第一虚拟积木式组件在所述三维坐标系
20 中的许多候选定向。
3. 根据权利要求1或2所述的方法，进一步包括：
- 确定所述结构的许多候选连接元件，其中每个所述候选连接元
25 件所述二维显示区域内在所述第一虚拟积木式组件的连接元件的投影周围的预定邻域内具有投影；
 - 基于所述连接规则和所述一组预定位置分级规则来选择所述候选连接元件中的一个；以及
 - 如果根据所述一组连接规则，所述第一虚拟积木式组件和所述

结构之间的连接是有效的，则至少通过所选定的候选连接元件连接所述第一虚拟积木式组件与所述结构。

4. 根据权利要求3所述的方法，其中所述选择候选连接元件中一个的步骤进一步包括对所述候选连接元件进行分级，其中所述分级对于它们与虚拟摄像头位置的距离来进行。

5. 根据权利要求3或4所述的方法，其中所述选择候选连接元件中一个的步骤进一步包括丢弃从当前虚拟摄像头位置不可见的候选连接元件。

6. 根据权利要求3到5中任一项所述的方法，其中所述选择候选连接元件中一个的步骤进一步包括对所述候选连接元件进行分级，其中所述分级是对于它们的二维投影与所述第一虚拟积木式组件的相应连接元件的二维投影的距离来进行的。

7. 根据权利要求3到6中任一项所述的方法，其中对于每个虚拟积木式组件，所述结构的数字表示包括许多对应于所述虚拟积木式组件至少一个表面的规则的栅格，其中每个规则的栅格包括许多栅格点，每个栅格点代表连接元件。

8. 根据权利要求7所述的方法，其中所述第一虚拟积木式组件的连接元件的投影周围的预定邻域的直径对应于所述相应规则栅格的相邻栅格点之间的间距。

9. 根据权利要求1到8中任一项所述的方法，其中所述确定许多候选位置的步骤进一步包括旋转所述第一积木式组件以获得旋转后的积木式组件的候选位置。

10. 根据权利要求 9 所述的方法，其中所述第一积木式组件的旋转被限制到用户选定的定向周围的预定空间角度。

5 11. 根据权利要求 1 到 10 中任一项所述的方法，进一步包括接收控制所述第一虚拟积木式组件的用户选定定向的用户命令。

12. 根据权利要求 11 所述的方法，其中所述用户选定定向被限制到一组离散定向中的一个。

10 13. 根据权利要求 1 到 12 中任一项所述的方法，其中所述二维投影是平行投影。

15 14. 一种包括存储介质的数据处理系统，其中在所述存储介质上存储有包括程序代码单元的计算机程序，当所述程序在所述数据处理系统上运行时，所述程序代码单元用于执行权利要求 1 到 13 中任一项的所有步骤。

20 15. 一种包含程序代码单元的计算机程序，当所述程序在计算机上运行时，所述程序代码单元用于执行权利要求 1 到 13 中任一项的所有步骤。

25 16. 一种包含程序代码单元的计算机程序产品，所述程序代码单元存储在计算机可读取介质上，当所述计算机程序产品被运行在计算机上时，所述程序代码单元用于执行权利要求 1 到 13 中任一项的方法。

17. 一种在其上存储有包含计算机程序单元的计算机程序的计算机可读取存储介质，当所述程序被运行在计算机上时，所述程序代码单元用于执行权利要求 1 到 13 中任一项的所有步骤。

18. 一种被实现为载波的数据信号，所述数据信号包括程序代码单元，其中当所述程序代码单元被运行在计算机上时，所述程序代码单元用于执行权利要求 1 到 13 中任一项的所有步骤。

用于操纵三维对象的数字表示的方法和系统

5 技术领域

本发明涉及操纵三维对象的数字表示。本发明尤其涉及由计算机实现的、依靠二维光标移动对三维虚拟积木模型所进行的操纵，虚拟积木模型包括多个虚拟积木式组件，每一块虚拟积木式组件包括许多连接元件，这些连接元件用于根据一组连接规则将虚拟积木组件与其它虚拟积木组件连接起来。

背景技术

虚拟现实的计算机辅助建模的任务是创建物理对象模型、操纵模型，此外在计算机系统中处理物理对象的模型。

15 从第一观点，虚拟现实的建模是令人感兴趣的课题，因为它使得可能当概念在现实世界中真正实现之前就将概念形象化。假如虚拟现实模型很容易修改，在与在现实世界中实现同样过程相比，能够在开发和改进物理对象的过程节约大量时间。在现实世界里给对象上色这样简单的任务可能就轻易花去几个小时，而计算机能够在几毫秒或几秒内就给模型涂上新颜色。

从第二观点，虚拟现实建模是令人感兴趣的，因为它使得可能创建现实世界中存在的对象模型，以及通过计算机来形象化和操纵模型。因此，可以存储对象模型以用于不同的目的，例如用于高级文档编制的目的。

25 尽管存在大量用于计算机辅助虚拟现实建模的可能应用，但一个特殊应用是将虚拟现实建模用于娱乐或教育。

物理建构玩具组的不同类型的建模思想同样是已知的。特别是，使用组合式或半组合式概念的思想是非常流行的。典型地，这些思想提供了一组预制元件，这些元件能够根据预制元件的模块以某种预定

方式彼此互联。这种建构玩具组的一个实例是名为 LEGO 的塑料玩具建构元件。

5 当在计算机上操纵三维(3D)对象或场景的表示时,通常的问题在于,对象在3D中的定位、选择或移动是借助于工作在二维中的输入设备来执行的,例如计算机鼠标在鼠标垫上的二维(2D)移动。此外,计算机的显示区域也是二维的。特别地,计算机屏幕的2D显示区域上的每一个选定的位置对应于3D空间内的多个位置,3D空间具有对应于所选定的2D位置的到2D显示区域的投影。问题在于确定可能符合用户意图的其中一个位置。

10 尽管已经进行了相当努力来开发更适于操纵3D计算机图像中的对象的计算机接口设备,但是这种工具没有被广泛使用,特别是没有与用于家庭娱乐、教育等的计算机系统相关。

已知的定位虚拟积木式组件的建构游戏典型地在对象的移动和旋转自由度或者将它们与其他对象连接的方式方面受到限制。

15 美国专利6,426,745描述了一种用鼠标操纵3D图形对象的方法,其中3D中的移动是通过在某个时刻把移动限制在一个平面或一个方向上来实现的,从而把3D移动用一系列受限的移动来实现,例如在3D坐标系中的X方向、随后在y方向、最后在Z方向上的一系列移动。

20 上述现有技术的系统的问题在于,用传统的用户接口来把积木式组件放置在3D场景中以及它们与3D结构的连接是个麻烦的过程,这个过程需要高水平的技巧和训练。尤其是在儿童及十几岁的青少年的娱乐和教育的环境中,这是个问题,因为儿童通常还没有发展出足够的肌肉运动技巧或者执行冗长操纵过程所需的耐性。因此,他们可能
25 很快就丧失对虚拟建构游戏的兴趣。

发明内容

上述问题及其他问题将通过由计算机实现的方法来解决,这种方法依靠二维的光标移动来操纵三维虚拟积木模型,虚拟积木模型包括

多个虚拟积木式组件，每块虚拟积木式组件包括许多连接元件，连接元件用于根据一组连接规则将虚拟积木式组件与其它虚拟积木式组件相连接，该方法包括：

5 -为结构的数字表示提供三维坐标系中的许多所述虚拟积木式组件；

-借助于光标在表示所述结构的投影的二维计算机显示区域中的运动来定位第一虚拟积木式组件的二维投影，以使其连接到结构上，从而产生二维位置坐标；

10 -根据二维位置坐标确定第一虚拟积木式组件在三维坐标系中的许多三维候选位置；以及

-基于连接规则和一组预定的位置分级（position rating）规则来选择所述候选位置中的一个；以及

-在选定的候选位置上将第一积木式组件连接到所述结构。

15 因此，提供了用于操纵三维对象的数字表示的有效方法，这种方法以对于用户直观的方式将 2D 显示区域中的位置转变（resolve）为相应的 3D 位置，因此使得用户很容易操纵 3D 虚拟结构。用户可以简单地在 2D 屏幕区域中定位新的积木式组件，因而只固定新积木式组件在两个方向上的坐标。随后根据许多连接规则以及一组位置分级规则，自动确定多个符合 2D 放置的可能 3D 位置中适当的一个。

20 特别地，通过根据一组预定的位置分级规则选择许多候选位置中的一个，处理导致的结果是把积木式组件定位在 3D 场景中可能符合用户意图的位置上。

进一步的优点在于，不符合用于连接元件的连接规则的位置被拒绝。例如，建构系统中可能有很多种不同的连接元件，例如不同形状和/或尺寸的突出物以及不同类型的、用于与相应一种突出物结合的孔洞。

假设新积木式组件被放置在符合 2D 放置的候选 3D 位置，通过选择靠近新积木式组件上连接元件的候选连接元件，并且通过检查与候选连接元件的有效连接是否可能，从而将计算机屏幕上新积木式组

件的 2D 放置正确地转变为 3D 模型中的 3D 位置。此外，避免了根据一组连接规则可能导致无效连接的 2D 到 3D 转变，例如突出物与不同类型孔洞的连接。

因此，优点在于，虚拟积木式组件与相应的实物建构组的物理限制一致，从而提高了虚拟建构游戏的教育价值。特别地，虚拟建构随后可以通过用相同类型的积木式组件建构物理模型而被变换成物理模型。

进一步的优点在于，本方法提供了大的移动和连接自由度。例如，积木式组件可以在任意方向上被定位和连接。特别地，在水平和垂直面上都可能连接到其他积木式组件。此外，即使积木式组件被旋转，例如围绕垂直轴或水平轴旋转，也能正确地定位它们。

优选地，通过经由诸如指针设备之类的适当输入设备从用户处接收到的控制命令来控制图形表示的定位。例如，定位可以作为标准光标操纵技术而被执行执行，诸如“拖放”操作，“单击-拖动-松开”操作及其他类似操作。

在一个优选实施例中，确定许多三维候选位置的步骤进一步包括确定第一虚拟积木式组件在三维坐标系中的许多候选位置。通过旋转第一积木式组件以获得旋转后的积木式组件的候选位置，将用户控制的 2D 位置转变为 3D 放置以及三个维度上的定向，从而只基于 2D 屏幕区域中的单个光标控制就确定 3D 场景中放置的全部六个自由度。当第一积木式组件的旋转被限制在用户选定定向周围的预定空间角度时，避免了诸如积木的完全翻转（complete flipping around）之类的令人混淆的旋转，从而避免了用户的迷惑（disorientation）。在该实施例中，可以通过允许用户选择虚拟积木式组件的定向来获得更多的定向。

在一个优选实施例中，方法进一步包括：

-确定所述结构的许多候选连接元件，每个候选连接元件具有第一虚拟积木式组件的连接元件的投影周围的预定邻域内二维显示区域中的投影；

-基于连接规则和该组预定的位置分级规则，选择所述候选连接元件中的一个；以及

5 -如果根据该组连接规则，第一虚拟积木式组件和所述结构之间的连接是有效的，则通过至少一个选定的候选连接元件将第一虚拟积木式组件与结构连接。

因此，提供了检测候选位置的计算有效方法，因为检测只依赖于单独连接元件的位置和属性。此外，提供了有效的选择机制，用于选择最可能符合用户意图的 3D 位置。

10 当分级包含一个或多个下述分级规则时，提供了特别有利的连接元件分级：

-关于它们到虚拟摄像头位置的距离地对候选连接元件进行分级；

-丢弃从当前虚拟摄像头位置看不到的候选连接元件；以及

15 -关于它们的二维投影到第一虚拟积木式组件的相应连接元件的二维投影的距离地对候选连接元件进行分级。

特别地，当分级规则是基于到虚拟摄像头位置的距离组合以及基于他们的二维投影与第一虚拟积木式组件的相应连接元件的二维投影的相应间距时，获得了可靠的 3D 定位。

20 在一个优选实施例中，对于每个虚拟积木式组件，所述结构的数字表示包括对应于虚拟积木式组件的至少一个表面的许多规则的栅格，每个规则的栅格包括许多栅格点，每个栅格点代表一个连接元件。因此，提供了系统化的框架，其允许 3D 位置的正确转变以及大量不同积木式组件的正确互联，即使积木式组件具有大量连接元件，甚至不同类型的连接元件。

25 优选地，所述第一虚拟积木式组件的连接元件的投影周围的所述预定邻域具有对应于相应规则栅格的相邻栅格点之间距离的线性标注（linear dimension），诸如直径。因此，提供了一种由用户控制的精确细致的放置，其产生 3D 分辨率和对于用户可预测的、透明的积木式组件连接。

进一步的优选实施例在从属权利要求中被公开。

本发明可以以不同的方式来实现，包括上述的以及下文中的方法、数据处理系统以及其他产品装置，其中每一种方式都产生一个或多个结合第一被提到的方法描述的好处和优点，并且每一种途径都具有对应于结合第一被提到的方法描述的优选实施例的一个或多个优选实施例。

注意，上述以及下文中的方法的特征可以在软件中实现，并且可以在数据处理系统或其他处理装置中通过执行计算机可执行指令而实现。指令可以是从存储介质或通过计算机网络从另一台计算机加载到诸如 RAM 之类的存储器中的程序代码单元。可选地，所述特征可以通过硬布线电路来实现，而不是通过软件或者结合软件来实现。

本发明还涉及包括程序代码单元的计算机程序，当所述程序在计算机中运行时，程序代码单元用于执行上述以及下文中的方法的所有步骤。计算机程序可以被实现为计算机可读存储介质，数据信号被实现为载波，等等。

本发明还涉及包含存储在计算机可读介质上的程序代码单元的计算机程序产品，当所述计算机程序运行在计算机上时，程序代码单元用于执行上述以及下文中的方法。

附图说明

以下将结合优选实施例，并参照附图来更为完整地阐明本发明，其中：

附图 1a-b 表示用于生成和操纵几何对象的计算机可读取模型的数据处理系统；

附图 2 表示虚拟积木系统的图形用户界面；

附图 3a-d 阐明了积木式组件及其连接元件的实例；

附图 4a-b 阐明了物理积木式组件的数字表示的实施例；

附图 4c 阐明了连接点周围的体积元素 (volume element) 的实例；

附图 5 表示在 3D 场景中定位积木式组件并且将它连接到已存在

于场景中的积木式组件的结构上的处理的流程图；

附图 6a-b 阐明了将新积木式组件定位在先前放置的积木式组件顶部上的实例；

附图 7a-b 阐明了由于屏幕上图形表示的略微不同的 2D 放置所导致的积木式组件的两个位置；

附图 8 表示在试验性的 3D 放置中，新积木式组件的连通性验证子程序的实施例的流程图；

附图 9a-b 阐明了包含了旋转的积木式组件的连接。

10 具体实施方式

附图 1a-b 表示用于生成和操纵几何对象的计算机可读取模型的数据处理系统；

附图 1a 表示计算机系统实例的示意图。计算机系统包括被适当编程的计算机 101，例如个人计算机，计算机 101 包括显示器 120、键盘 121 以及计算机鼠标 122 和/或另一指针设备，诸如触摸板、轨迹球、光笔、触摸屏等。

标号为 101 的计算机系统适用于促进设计、存储、操纵以及共享虚拟积木式组件模型。计算机系统 101 可以用作单机系统或者客户端/服务器系统中的客户端。

附图 1b 表示用于生成和操纵计算机可读取的虚拟积木模型的数据处理系统的方框图。计算机 101 包括存储器 102，存储器可以部分被实现为易失性存储装置，并且部分被实现为非易失性存储装置，例如随机存取存储器 (RAM) 和硬盘。存储器具有存储在其上的模型代码解释器 107、模型代码产生器 108、UI 事件处理程序 109 以及建模应用 110，其中每一个都可由中央处理单元 103 执行。而且，存储器具有存储在其中的模型数据 111，即代表物理对象的数字表示的一组数据结构，例如虚拟积木模型。用于存储虚拟积木模型的数据格式的一个实例在美国专利 6,389,375 中被公开了。

代码解释器 107 适用于读取和解释定义模型的代码，即代表模型

的积木式组件的数据结构的代码。在优选实施例中，代码解释器适用于读取模型并将这种模型转换为已知图片格式，用于在计算机显示器上显示。

5 UI 事件处理程序 109 适用于将用户与用户接口的交互转换为代码发生器 108 可识别的适当的用户命令。一组可能的并且可识别的命令可包括：从元件库获取积木式组件，放置积木式组件以连接到另一积木式组件，分离积木式组件，丢弃积木式组件，例如通过发起旋转等来操纵积木式组件或一组积木式组件等等。每一个命令可以与一组相应的参数相关联，例如相对于显示器坐标系的光标坐标、积木式组件
10 的类型等。

代码解释器 108 适用于响应于用户的命令修改模型的数据结构。作为同时的或后续的任务，可以执行代码解释器，用于表示代码产生器的结果。

建模应用 110 适用于控制存储器、文件、用户接口等。

15 用户 105 能够借助于用户接口 106 与计算机系统 101 进行交互，用户接口 106 优选地包括显示在计算机屏幕上的图形用户界面，以及一个或多个诸如键盘和/或指针设备的输入设备。

为了加载、存储或传递模型、几何描述或其他数据，计算机系统包括输入/输出单元 (I/O) 104。输入/输出单元可以用作与不同类型存储介质和不同类型计算机网络，例如因特网，的接口。此外，输入/
20 输出单元 (I/O) 104 可以被用于例如交互地与其他用户交换模型。

借助于数据总线 112 实现存储器 102、中央处理单元 (CPU) 103、用户接口 (UI) 106 以及输入/输出单元 104 之间的数据交换。

附图 2 表示了虚拟积木系统的图形用户界面。

25 用户界面包括显示区域 201，用于显示具有底盘 202 以及包含许多互联的虚拟积木式组件 204 的 3D 结构 203 的 3D 场景视图。从预定的观察点显示场景。以下，这个观察点也被称为 (虚拟) 摄像头位置，因为它对应于这样一个位置，从这个位置上摄像头记录对应于显示区域中所示图形图像的实际结构的图像。

每个积木式组件 204 对应于图形用户界面中的活性元件，例如通过用计算机鼠标点击可以将其激活，从而选择该积木式组件。在一个实施例中，被选中的虚拟积木式组件改变了外观。例如，被选中的积木式组件可以改变颜色、纹理等；可以通过在被选中的积木周围显示有边界框等来加亮这个积木。用户可以操纵被选中的积木式组件，例如改变其属性，例如其颜色、将其删除、执行复制和粘贴操作、将其拖放到不同位置等。

用户界面还包括选项面板 (palette panel) 205，选项面板 205 包括许多可以被用户选择的不同积木式组件 206。例如，用户可以用鼠标在其中一个积木式组件 206 上点击，从而选择积木式组件，以及将被选中的积木式组件拖进显示区域 201，以将它连接到结构 203 或者连接到底盘 202。

用户界面还包括菜单栏 207，菜单栏包含许多用于启动各种功能或工具的菜单按钮 208。例如，工具栏可以包括用于改变虚拟摄像头位置的旋转工具，从而允许用户从不同方向观察构造区域。菜单栏可以进一步包括用于放大和缩小 3D 场景的缩放工具。工具的其他实例包括用于选择不同选项面板 205 的选项面板工具，其中每个选项面板包括不同的积木式组件组合、用于为结构的部分上色的上色工具、用于擦除积木式组件的橡皮擦工具等。

菜单栏 207 可以进一步提供标准功能，诸如用于保存模型的功能、用于打开以前保存的模型的功能、用于打印模型图像的功能、帮助功能等。

附图 3a-d 阐明了积木式组件及其连接元件的实例。

附图 3a 表示积木式组件 301 的透视图。积木式组件 301 的上表面 302 带有 8 个突起 303a-h，这些突起能够与另一积木式组件的相应孔眼，例如另一积木式组件底面上的孔眼，结合。相应地，积木式组件 301 包括带有相应孔眼的底面 (未表示)。积木式组件 301 还包括不包含任何连接元件的侧面 304。

通常，连接元件可以被分组为不同种类的连接元件，例如连接件、

接受件以及混合元件。连接件是可以被另一积木式组件的接受件接纳以提供积木式组件之间连接的连接元件。例如，连接件可以容纳在另一组件的部分之间、插入孔眼内等等。接受件是能够接纳另一积木式组件的连接件的连接元件。混合元件是既能充当连接件又能充当接受件的部分，这通常取决于另一积木的合作连接元件的类型。

附图 3b 表示从底部观察积木式组件 310 的透视图。积木式组件 310 具有非矩形的顶面和底面。底面包括用于接纳一个或多个其他积木式组件，例如附图 3a 中的积木式组件 301，的相应突起的孔眼 311、312 以及 313。孔眼由边沿 314、由二级销柱 (secondary pin) 315 以及拐角 316 和 317 限定。因此，所有上述元件的属性决定了积木式组件 310 底面的连接属性。

附图 3c 阐明了积木式组件 320 与积木式组件 321 连接，从而形成组合的积木式组件。积木式组件 320 在其上表面上包含突起 322，突起 322 容纳到另一积木式组件的相应孔眼中。然而，如附图 3c 中所阐述的，也可以实现其他类型的连接。突起之间的间隙 323 充当其他连接件，诸如积木式组件 321 的侧面 324，的接受件。对于物理积木式组件，这种属性由间隙的尺寸以及积木式组件 321 的尺寸，即它的侧面宽度 325，决定。然而，在数字表示中，通过规则的栅格上的连接点的相应属性来表示这些属性，如在下文中更为详细地阐述。

附图 3d 表示两个积木式组件 331 和 332。积木式组件 331 是砖形的，在它的上表面上具有 4 个突起 333，在底面（未表示）上具有 4 个相应的孔眼。砖形物 332 是具有包含不相互正交的平面的表面的积木式组件实例。特别地，积木式组件 332 具有斜面 334。如附图 3d 所阐明的那样，在当前位置，积木式组件 331 和 332 不连接，因为在所示的位置上没有任何彼此结合的连接元件。

应该理解，上述积木式组件仅仅用作可能的积木式组件的实例。

附图 4a-c 阐明了物理积木式组件的数字表示的一个实施例。物理积木式组件的数字表示包括存储在计算机存储器中的数据结构，其中数字表示代表该积木式组件的属性。根据该实施例，积木式组件的

连接元件可以在种类 (nature)、外形、连接属性等方面各不相同。然而，他们在积木式组件上的位置遵循一组规则。

积木式组件具有许多与它相关的平面，例如由积木式组件的有边界框的表面所定义的平面。积木式组件的连接元件位于这些平面内，使得每个连接元件具有与其相关的轴。相同平面内所有连接元件的轴对应于规则的栅格，例如正交的栅格，的相应栅格点，所谓的节点，并且邻近栅格点之间的间距是固定的。优选地，与积木式组件关联的平面成对地彼此平行，例如通过定义一组对应于积木式组件的上表面和底面的水平平面以及许多对应于积木式组件侧面的垂直平面。优选地，在所有水平平面上，相邻栅格点之间的间距是相同。在一个实施例中，垂直平面中相邻栅格点之间的间距与水平平面中相邻栅格点之间的间距不相等。

附图 4a-b 示意性地阐明了数据结构的实施例如何描述附图 3a 的积木式组件 301 的属性。在附图 4a-b 中，显示了积木式组件 301 及其上表面和底面各自的相应正交栅格 405 和 406。附图 4a 表示积木式组件的透视图，而附图 4b 表示积木式组件的顶视图。在附图 4a-b 中，没有表示积木式组件 301 侧面上的栅格。积木式组件的某些实例，例如积木式组件 301，在某些面上，例如在侧面上，没有任何连接元件。积木式组件的其他实例可能在所有侧面上具有连接元件。

正交栅格也被称为连通性栅格，栅格点也被称为连接点。用如圆周 407a-k 所示例的圆周来阐明连接点。因此，连接点 407a-h 分别对应于突起 303a-h。由于侧面 304 没有任何连接元件，因此不需要为它们定义任何连接栅格。在可选实施例中，可以为侧面定义只包含虚接受件的连接栅格，即既不连接也不抵制其他连接元件的接受件。

相对于内部右手坐标系 408 描述积木式组件的数字表示。应该理解，坐标系的选择，特别是其原点的位置和坐标轴的方向，可以根据任意适当的惯例来选择。因此，在相应的数据结构中，积木式组件在 3D 场景中的位置和定向可以由坐标系 408 相对于外部坐标系，例如另一积木式组件的坐标系或全局“世界”坐标系，的原点坐标以及坐标轴

的方向来表示。

正如从附图 4a 中所能看到的那样，通过置于规则栅格中的连接点来表示积木式组件的连接元件就将某些约束强加在物理积木式组件上的连接元件的物理放置上。栅格 405 位于突起 303 从其延伸的积木式组件上表面的平面内。

在附图 4a-b 的实例中，栅格点被放置在正方形栅格内，其中每个正方形的尺寸为 $5*5$ 个的任意长度单位 (LU)。因此，在这个几何里，连接元件也被放置在相应的正方形栅格上，并且积木式组件的平面内连接元件之间的间距为 $10LU$ 的倍数。在附图 4 的实例中，积木式组件的上表面和下表面是矩形的，其尺寸为 $20LU*40LU$ ，并且相邻连接元件的空间距离为 $10LU$ 。另一方面，在垂直方向上，连接元件的空间距离为 $12LU$ 。

相对于积木式组件的内部坐标系 408 来定义连接点的位置。每个栅格点还具有与其相关联的方向，指示连接元件在哪个方向上可以与相应的连接元件结合。栅格点的方向垂直于平面，并且指向有边界框的外部，即在附图 4b 中指向图中平面的外部。

在一个实施例中，对于每个连接点，代表连接点的栅格的数据结构包括：连接点相对于坐标系 408 的坐标，连接元件的方向，以及连接类型。

在一个实施例中，代表积木式组件的数据结构包括积木式组件 ID (识别符)，描述积木式组件的有边界框的位置、定向以及尺寸的数据结构，以及描述各个栅格以及它们的相应连接点的许多数据结构。代表连接点的栅格的数据结构包括：

-用作局部栅格坐标系原点的栅格点相对于坐标系 408 的坐标。
在栅格 405 的例子中，栅格点 307i 被作为原点，坐标为 $P_0 = (-5, 12, -15)$ ，单位为 LU。

-连接元件的方向。在栅格 405 的例子中，这为 $(0, 1, 0)$ ，即坐标系 408 的 Y 轴方向。

-X 和 Z 方向上栅格点的数量。在栅格 405 的例子中，分别是 $n_x=9$

和 $n_z=5$ 。

$-n_x*n_z$ 数据结构的阵列，每个包括相应的连接点的连接属性。每个栅格点的数据结构包括连接点的连接类型，例如“孔眼”、“边”、“突起”等等。

5 在一个实施例中，每个连接点还具有与之相关的体积元素，定义了该连接点周围的邻域。

附图 4c 阐明了连接点周围的体积元素的实例。在一个实施例中，每个连接点具有与之相关联的球形，其中心位于连接点上。在附图 4c 中，这是通过上述积木式组件 301 的连接栅格 405 的连接点 407g 和 10 407j 来示例的。对于连接点 407g 和 407j，分别示出了相应的球形 410g 和 410j。球形的半径等于相邻连接点之间间距的一半，例如在附图 4a-b 的实例中为 2.5LU。应该了解，在可选实施例中，球形可以更小或更大，即它们可以交叉或者在相邻球形的球面之间可以有间隙，从而允许相对于小误差的定位方法的灵敏度控制。

15 还应该理解，也可以使用不同类型的体积元素。例如体积元素可以是包围连接元件的圆柱体，如包围连接点 407c 的圆柱体 410c 所示例。在这个实例中，圆柱体的半径为 2.5LU 并且高度为 5LU。体积元素的其他实例包括正方体、椭圆体等。

还应该理解，在某些实施例中不同类型的连接元件可以与不同体 20 积元素相关联，例如不同尺寸，例如不同半径和/或不同高度，的体积元素，或者甚至不同几何形状。在这种实施例中，代表连接点的数据结构可以还包括一个或多个用于识别相关联的体积元素的类型和/或尺寸的参数。

应该理解，当在显示器，例如计算机屏幕，上图形表示积木式组 25 件时，边界范围 (bounding volume)、坐标系、体积元素以及栅格不需要被显示。优选地，图形表示只包括积木式组件自身的图形描述。

以下将更加详细地描述把新积木式组件放入包含 3D 结构的场景的过程。将参照附图 5 和 6a-b。

附图 5 表示把积木式组件定位在 3D 场景并且将它连接到已经出

现在场景中的积木式组件结构上的过程的流程图。

在步骤 501 中，新积木式组件被放置在场景中，例如通过选择选项面板上的积木式组件并且根据鼠标移动将积木式组件的图形表示拖到 2D 显示区域中用户所选定的位置，其中 2D 显示区域表示 3D 场景的 2D 投影。在一个实施例中，当鼠标的移动至少停止了一个预定时间，例如 GUI 事件处理程序所检测到的，时，程序进入步骤 503。

附图 6a-b 阐明了将新积木式组件定位在先前放置的积木式组件的顶部的实例。新积木式组件 601 被图示为有边界框 620，有边界框 620 具有上表面 621 和底面 622，每一面包括许多如上所述的连接栅格的连接点，其一般地由 603 标示。类似地，先前放置的积木式组件 602 具有许多连接点，其一般地用 604 标示。在附图 6a 中没有表示积木式组件 601 和 602 的侧面的连接栅格和连接点。在这个实例中，3D 场景只包括一个先前放置的积木式组件 602。

附图 6a 阐明了在计算机屏幕的 2D 显示区域 600 中看到的积木式组件 601 和 602 的 2D 投影，其对应于 2D 显示坐标系 630。

附图 6b 阐明了显示区域 600 中的 2D 投影和积木式组件 601 和 602 在 3D 坐标系 640 中的 3D 位置之间的关系。当新积木式组件 601 被放置在屏幕 600 上时，只确定了它在 2D 坐标系 630 中的坐标，并且，如虚线投影线 631 所述，产生投影 601a 的任何 3D 位置符合这个投影。因此，由用户控制的将积木式组件放置于屏幕上并没有提供关于垂直于显示区域的第三维的信息。在附图 6b 中，投影是平行投影，即投影线 631 彼此平行，相当于虚拟摄像头位置 632 距离投影平面 600 无限远。

再次参照附图 5，在步骤 503 中，处理确定了，对于新积木式组件的每个连接点，是否存在属于已经存在于场景中的任何积木式组件（或，如果存在，属于底盘）的连接点，其具有在新积木式组件的该连接点投影的预定邻域内的 2D 投影。在一个实施例中，对于新积木式组件的每一个连接元件，确定显示区域内包围连接元件的圆周。类似地，确定已经存在于场景中的连接元件的相应圆周。如果这些圆周

中的任何圆周与新积木式组件的连接元件的圆周重叠，则选择相应的连接元件作为候选的连接元件。

在附图 6a 中，用新积木式组件 601 的连接点 605 来阐明这个步骤。标号 606 标示了包围连接元件 605 的圆周。类似地，标号 608、
5 610 和 612 分别标示了积木式组件 602 的连接元件 607、609 和 611 的圆周，这些圆周与圆周 605 交叠。因此，在这个实例中，连接元件 607、609 和 611 被确定为候选连接元件。应该理解，在这个实例中，可以选择进一步的连接元件。然而，它们的圆周在附图 6b 中没有被表示。

应该注意，圆周 606、608、610 和 612 分别对应于包围连接元件
10 605、607、609 和 611 的相应球形的投影。应该理解，在可选实施例中，为了确定哪些连接元件具有在新积木式组件的连接元件的邻域之内的投影，可以使用不同类型体积元素的投影。结合上述附图 4c 已经描述了体积元素的更多实例。当使用球形时，相应投影的确定尤为有效，因为球形的投影始终是圆，而与视点无关。

15 在一个优选实施例中，圆周的直径对应于每个栅格中相邻连接点之间的间距。

应该进一步理解，在某些实施例中，候选连接元件的搜索可以被限制在已经存在于场景中的积木式组件的连接元件的子集内，从而减少完成搜索所需要的时间以及候选连接元件的结果列表的大小。例如，
20 搜索可以被限制在还没有连接到另一连接元件的连接元件。在一个实施例中，搜索被限制在具有合适连接属性的连接元件，即能够被实际连接到新积木式组件的连接元件上的连接元件，如下文中将更详细地描述的。

步骤 530 产生已经存在于场景中的积木式组件的候选连接元件的
25 列表 520。因此，候选连接元件的 3D 坐标是已知的。在一个实施例中，列表 520 中的每个条目包括：已经存在于场景中的积木式组件的候选连接点的 3D 坐标，识别相应积木式组件的标识符，以及识别新积木式组件的相应连接元件的标识符和/或坐标。

在步骤 505 中，通过根据预定的一组分级规则对候选连接元件进

行分级来对候选连接元件的列表 520 排序。

在一个实施例中，候选连接元件是根据他们与虚拟摄像头的距离，即它们与对应于显示区域 600 的投影平面的相对距离，以及根据候选位置和用户选定放置之间的 2D 距离，即候选位置在显示区域中的投影与新积木式组件的图形的位置的位移，而被分级。例如，候选位置对于其更靠近摄像头位置的连接元件可以被分集为更高，即更可能符合用户所想要的位置。同样，在 3D 中与用户选定位置的位移对于其是高的候选连接点可以被分集为更低。

在某些实施例中，作为上述条件的附加或替代，根据其他标准执行排序。这种位置分级规则的实例可以包括以下一条或多条：

- 与摄像头位置的相对距离；
- 与用户选定的 2D 位置的位移；
- 积木式组件所需旋转的角度，以便允许与候选连接元件的连接；
- 可见性检测，例如候选连接元件从当前摄像头位置上是否可见。

在某些实施例中，从列表中丢弃从当前摄像头位置不可见的候选连接元件。

在某些实施例中，上述标准中的几个或者全部可被组合运用，例如通过定义上述条件的价值函数，可选地具有相应的加权因子。在其他实施例中，首先根据上述规则其中之一，例如所需的 2D 位移，执行分级。如果两个或多个候选连接元件具有相同的分级，可以用另一规则来区分它们，例如与摄像头的相对距离、所需的旋转等。

因此，完成了对所确定的候选连接元件的分级，并且有序列表 520 根据上述分级对应于有序列表。在此，有序列表 520 包括基于上述位置分级规则没有被丢弃的所有候选连接点。

在步骤 508 中，确定有序列表 520 是否为空。如果列表为空，即没有任何连接元件具有与新积木式组件的连接元件的足够小的投影距离，则 3D 放置被拒绝（步骤 509），即处理终止；否则，处理前进到步骤 510。

如果处理通过拒绝 3D 放置（步骤 509）而终止，则优选地将其

指示给用户，例如通过提供可视的指示。例如，可以以指示积木式组件还没有被放置的形式显示响应于鼠标移动而被移动的积木式组件的图形表示。仅当发现 3D 放置时，图形表示切换，从而向用户指示放置已经完成。例如，只要 3D 放置没有成功完成，积木式组件可以被显示为透明和/或具有可视的有边界框和/或闪烁和/或任何其他适当的形式。

在步骤 510 中，选择具有最高分级的候选连接元件，并且新积木式组件被放置在 3D 坐标系中，使得选中的候选连接点的坐标和新积木式组件的相应连接点的坐标相一致。因此，选择根据最高分级的候选位置。应该理解，该放置只是试验性的，即仍然没有被显示在计算机屏幕上。

在步骤 511 中，处理验证连接是否符合预定的一组连接规则，例如，在新积木式组件的这些和其他连接点上的连接元件是否允许与新积木式组件要连接到的积木式组件的连接元件的连接。这个其实例将在下文中更详细描述的检查也将被称为连通性验证。在优选实施例中，处理进一步验证新积木式组件在连接到候选连接元件时从摄像头位置上是否可见，或者其是否被存在的结构完全遮挡（obstruct）。如果连接被拒绝或者积木式组件将是不可见的，则试验性的 3D 放置被拒绝，并且处理前进到步骤 512；否则，如果处理成功地验证了连接，则处理前进到步骤 513。

在步骤 512 中，连接对于其被拒绝的候选连接点被从有序列表 520 中移除，并且处理返回步骤 508，即如果存在，则选择下一个候选连接点。

在步骤 513 中，即如果连接检查是成功的，则 3D 放置被接受，即通过在被接受的位置上“咬住”新积木式组件以及通过改变积木式组件的外观来更新结构的图形表示。在一个实施例中，用户可以接受位置，例如通过用鼠标在积木式组件上点击，从而完成放置处理。如果用户希望把积木式组件放置在另一位置，那么用户可以简单地移动鼠标，而不需要在积木式组件上点击，从而把新积木式组件的外观变回

最初的外观，并且把积木式组件移动到 2D 显示区域的另一位置。

处理保持一组数据结构，每个数据结构表示放置在 3D 场景中的积木式组件，例如如结合附图 4a-b 所描述。当接受新积木式组件的 3D 放置时，处理通过增加对应于新积木式组件的数据结构的新实例作为数据结构组的一部分来更新该数据结构组。

附图 7a-b 阐明了由屏幕上图形表示的略微不同的 2D 放置所导致的附图 6a-b 的新积木式组件的两个位置。在附图 7a 中，积木式组件 701 被放置在结构 702 的顶部。另一方面，在附图 7b 中，积木式组件 701 被定位在结构 702 后面在底盘 703 上。因此，新积木式组件的 2D 位置的小调整允许用户区分 3D 场景中的两种相当不同的放置。附图 7a-b 进一步阐明了新积木式组件 701 当其被用户操纵时与结构 702 的积木式组件 704 相比的不同外观。

附图 8 表示在试验性的 3D 放置中新积木式组件的连通性验证子处理 511 的实施例流程图。由此，处理验证新积木式组件是否能够被连接到现有结构的至少一个其他积木式组件上。该其他积木式组件被称为第二积木式组件。

在初始步骤 801 中，处理执行冲突检测和可视性检查，即检测新积木式组件是否与已经存在于场景中的任何其他积木式组件相交，以及检查新积木式组件是否将被已存在的结构完全遮挡并因此从当前摄像头位置是不可见的。冲突检测可以通过任何适当的冲突检测方法来执行，优选为基于积木式组件的边界范围的冲突检测方法。这种算法的实例被公开在例如 David H. Eberly 的“3D Game Engine Design”，Morgan Kaufmann, 2001 中。同样，可视性检查可以通过本技术领域已知的任何适当的方法来执行。如果检测到无效的交点，即如果积木式组件的边界范围相交，或者如果积木式组件从当前摄像头位置将是不可见的，则拒绝在这个位置和定向上放置积木式组件；否则，如果积木式组件将是至少部分可见的以及如果检测到有效的交点，即如果至少部分有边界框表面相交但它们的体积不相交，则处理进入步骤 802。

在步骤 802 中，确定新积木式组件和第二积木式组件的属于所检测到的有边界框表面交点的所有连接点。只有那些还没有被连接的连接点需要被考虑；这些连接点将被称为相关连接点。

5 在步骤 803 中，选择新积木式组件的第一相关连接点，例如任意选择的连接点。

在步骤 804 中，对于新积木式组件的选定连接点，处理检查是否存在任何坐标与所选中的连接点相同的第二积木式组件的相关连接点。在一个实施例中，其中积木式组件被放置在离散的体积参考栅格中，并且所有坐标都是任意长度单元的倍数，可能需要坐标的精确匹
10 配。在连续或准连续的参考坐标系中，可能需要栅格点在预定限制中一致。

如果没有发现这种匹配连接点，那么处理进入步骤 814。

在步骤 814 中，检测在被选中的连接点的预定邻域内是否存在任何其他相关连接点。例如，在其中相邻的两个连接元件之间的间距为
15 10LU 的实施例中，可以选择预定邻域为包围着被选中的、位于 (x, y, z) 的连接点的立方体 $(x \pm 5LU, y \pm 5LU, z \pm 5LU)$ 。如果在被选中的连接点的预定邻域之内存在任何其他相关连接点，那么两个积木式组件的连接被拒绝（步骤 811）并且算法终止。因此，当该实施例中的连接点被放置在规则的栅格上时，积木式组件的无效放置可以被
20 有效地检测：如果对于新积木式组件的相关连接点中的一个，发现与第二积木式组件的相关连接点不匹配，则不需要检测新积木式组件的剩余连接点，从而提高了检测处理的速度。

如果在步骤 814 中没有找到任何冲突的相关连接点，则处理进入步骤 809。

25 如果在步骤 804 中找到匹配的连接点，那么处理进入步骤 805，其中检测在被选中的连接点的预定邻域内是否存在任何其他相关连接点，例如如上所述，在包围着被选中的、位于 (x, y, z) 的连接点的立方体 $(x \pm 5LU, y \pm 5LU, z \pm 5LU)$ 之内。如果在预定邻域内找到另一连接点，那么拒绝位置（步骤 811）；否则，程序进入步骤 806。

在可选实施例中，上述约束可能是不希望的。而且，在再一实施例中，上述约束可能被限制到某些连接类型。

在步骤 806 中，处理检测被选中的连接点以及被检测到的匹配的连接点是否具有相反的方向，即它们的关联轴是否沿同一直线但在相反定向上。因此，只接受定位在适合他们结合的相关定向上的连接元件。

应该注意，在可选实施例中，这种限制可以被放松，例如通过在其中连接元件接受一个定向范围的实施例中接受一定定向范围。

如果连接点的相对方向被接受，那么处理进入步骤 807，否则，这个位置被拒绝（步骤 811）。

在步骤 807 中，比较被选中的连接点以及相应的被检测到的匹配连接点的连接类型。在一个实施例中，每个连接点都具有相关联的连接类型，例如“突起”、“孔眼”、“边”、“拐角”、“合叶”、“销”、“细销”等。处理存取存储在存储器中的连接表 813。连接表包括所有连接类型对的连通性类型，指示特定连接类型对如何影响两个积木式组件的连接。例如，每对连接类型可以被关联到连通性类型“真”、“假”或“空”。连通性类型“真”指示连接是有效的，并且相应连接元件结合以连接两个积木式组件，例如突起与相应的孔眼结合。连通性类型“假”指示连接不被允许。例如，突起与突起之间的连接是不可能的：它们不仅不结合以提供连接，而且它们甚至妨碍/抵触，使得连接不可能。最后，连通性类型“空”指示没有任何阻止连接的东西，也没有任何实际连接的东西。例如，孔眼和孔眼就是这样的例子。因此，对于给定的一对连接点，处理能够从所存储的连接表 813 中检索相应的连通性类型。

在随后的步骤 808 中，检查连通性结果是否为“假”，即在相应的连接类型之间不可能有任何有效的连接。如果连通性结果为“假”，那么新积木式组件的位置被拒绝（步骤 811），否则存储连通性结果，并且处理进入步骤 809。

在步骤 809 中，检查新积木式组件的所有相关连接点是否都被处理。如果没有，则选择仍然没有被处理的相关连接点（步骤 812），

并且通过执行上述步骤 804、805、806、807 以及 808 来处理当前选中的连接点。

如果新积木式组件的所有相关连接点都已经被处理，并且位置也没有被拒绝，则接受这个位置并且处理进入步骤 810。在步骤 810 中，
5 基于存储的连通性结果确定积木式组件如何连接，以及相应地更新它们各自的数据结构。

起初，检验是否所有连通性结果为“空”。如果是，即如果没有任何东西阻止积木式组件的位置，但是也没有任何连接元件实际结合以连接积木式组件，则允许新积木式组件处于其当前位置。在一个实施
10 例中，附加的算法可以决定，例如基于边界范围，被放置在这个位置上的物理积木式组件是否会掉落、倾斜等，并相应地允许或拒绝这个位置。

否则，即如果一个或多个连通性结果为真，则处理确定积木式组件如何被连接，即它们是否被刚性连接或者连接是否允许相对旋转、
15 转换和/或类似操作。

一旦数据结构被更新，则子处理终止并且返回到附图 5 的总处理。

附图 9a-b 阐明了包含旋转的积木式组件的连接。在某些实施例中，只对屏幕上可见的新积木式组件的定向，即用户选定的定向或缺省定向，检查连接点。当用户希望在积木式组件旋转之后连接新积木
20 式组件时，可以通过适当的用户接口控制来控制定向，例如借助于可以在图形用户界面的菜单中选择的旋转工具、借助于键盘命令等。例如，用户可以控制新积木式组件的定向，使逐步遍历（step through）离散定向序列，例如对应于围绕不同轴线的 90 度旋转。例如，在附图 9a 和 9b 中，积木式组件 901 根据积木式组件 901 的用户选定定向以不同的方式被连接到结构 902 中。在附图 9a 中，积木式组件 901 被连接到结构 902 的顶部。在附图 9b 中，积木式组件 901 被连接到结构
25 902 中位于垂直平面上的突起 903 上。

在一个可选实施例中，允许积木式组件的自动重定向。例如，当在附图 5 的步骤 510 和 511 中分析候选位置时，可以考虑新积木式组

件的不同旋转作为附加的候选位置，并且根据所实现的分级和随后的连通性验证来分析其。例如，当新积木式组件要在某任意角度上被连接到合叶上时，用户不必精确地将新积木式组件对准到合叶的轴上。处理将估计旋转后的位置并在旋转之后连接新积木式组件。在某些实施例中，自动重定向被限制到预定的空间角度，例如 45 度、30 度等，因而避免了大的旋转，大的旋转可能使用户混淆并导致意外的结果。

上述处理的另一个优点在于，用户接收到关于新积木式组件将被放置和连接到何处的清晰的指示。而且，考虑了由用户控制的、屏幕上新积木式组件的 2D 位置的细小修正，从而提供了流畅自然的工作流程印象。

应该理解，本领域熟练技术人员可以在本发明的范围之内实现上述方法的变型。例如，某些上述步骤的顺序可以被改变，步骤可能被结合等。

例如，当新积木式组件已经被放置到 3D 场景之后，用户可以启动搜索工具以搜索类似的位置。在一个实施例中，搜索工具可以允许用户逐步遍历附图 5 的有序列表 520 的所有位置，从而允许用户选择其中一个可能的位置。

术语“操纵”对象的数字表示是用来代表对于数字表示的任何受用户控制的操作，例如新积木式组件的连接，把现存积木式组件从一个位置移动到另一个位置，或者任何改变积木式组件在 3D 场景中的位置的其他操作。

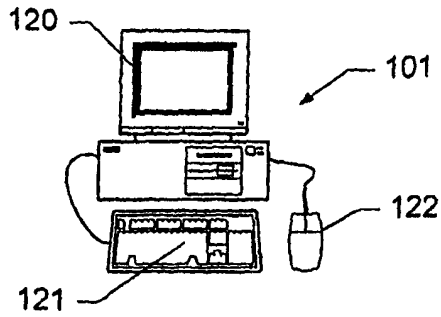


图 1a

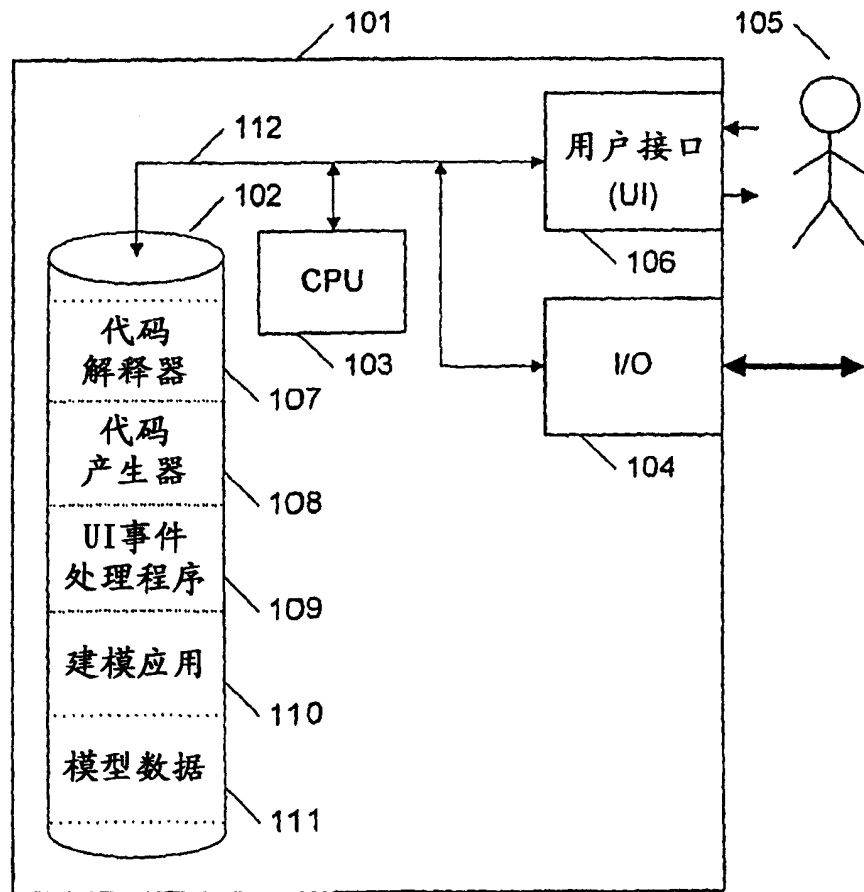


图 1b

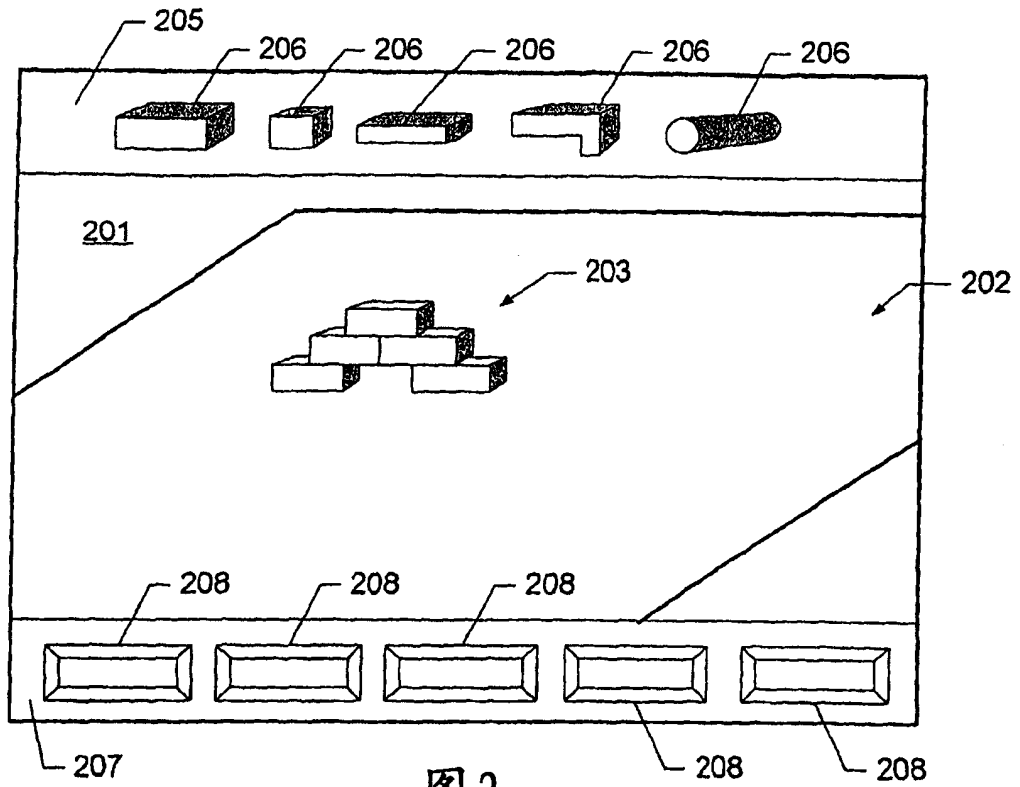


图 2

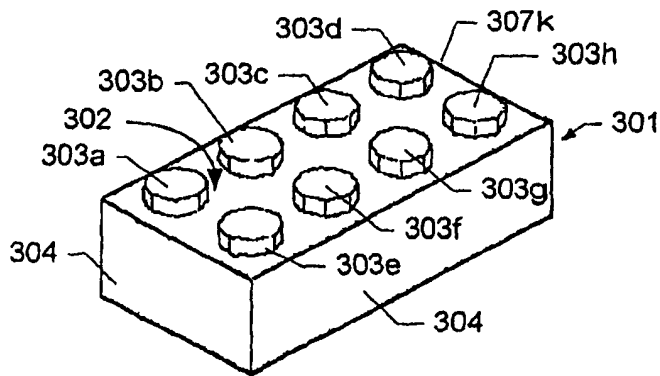


图 3a

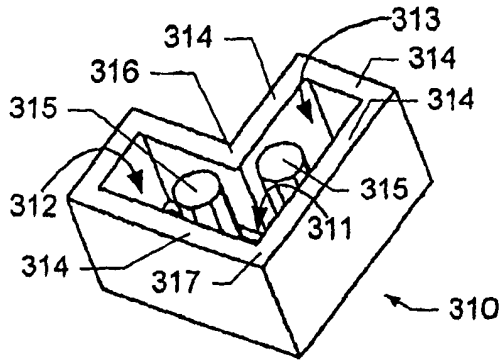


图 3b

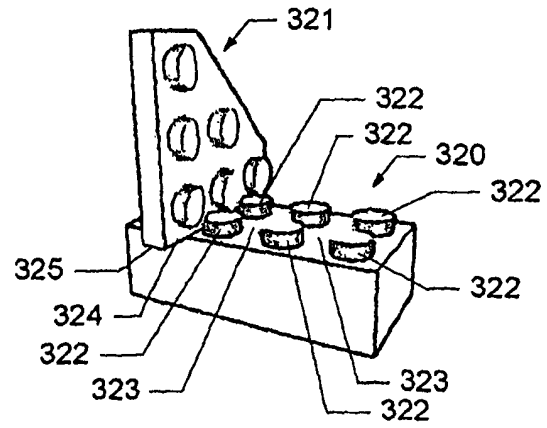


图 3c

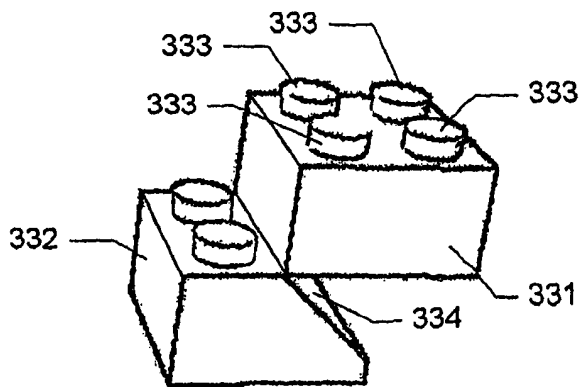


图 3d

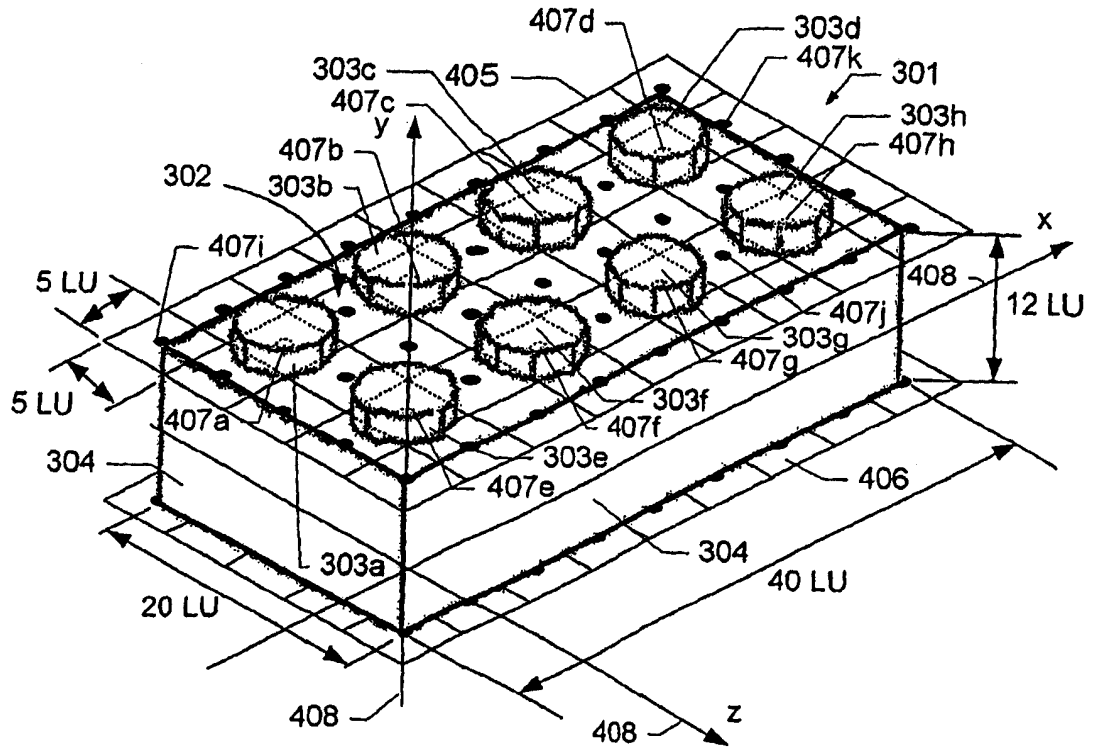


图 4a

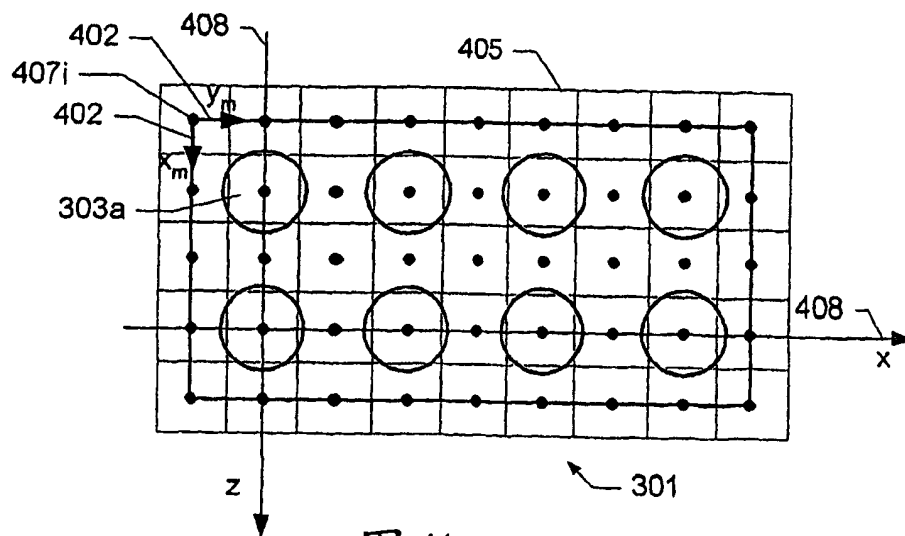


图 4b

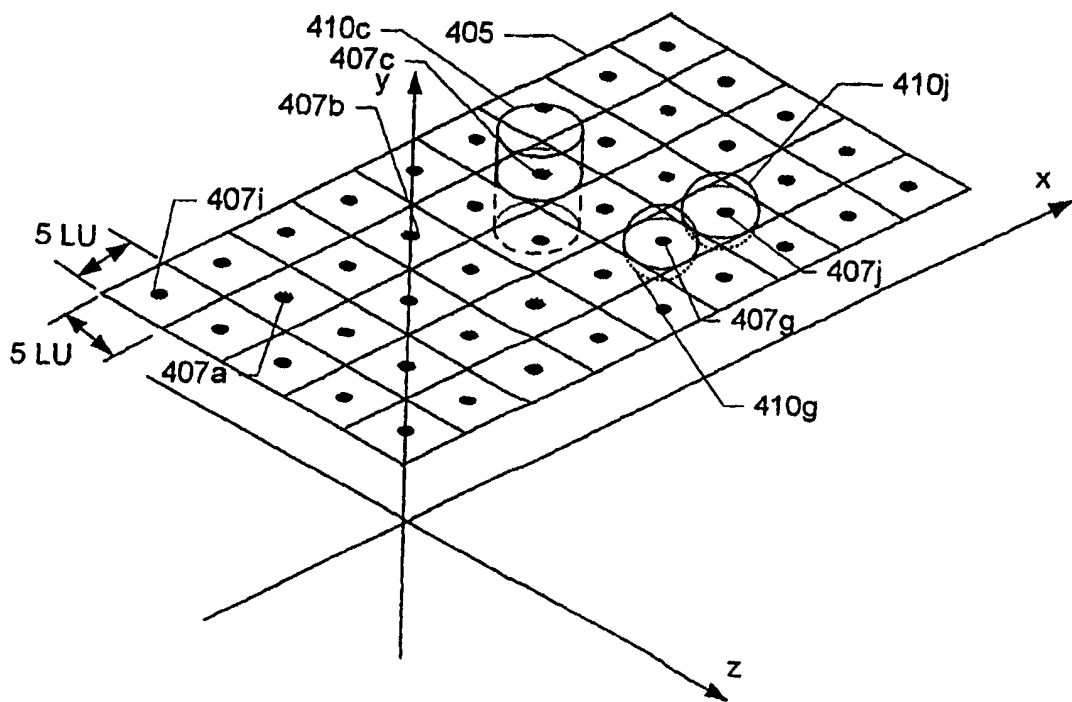


图 4c

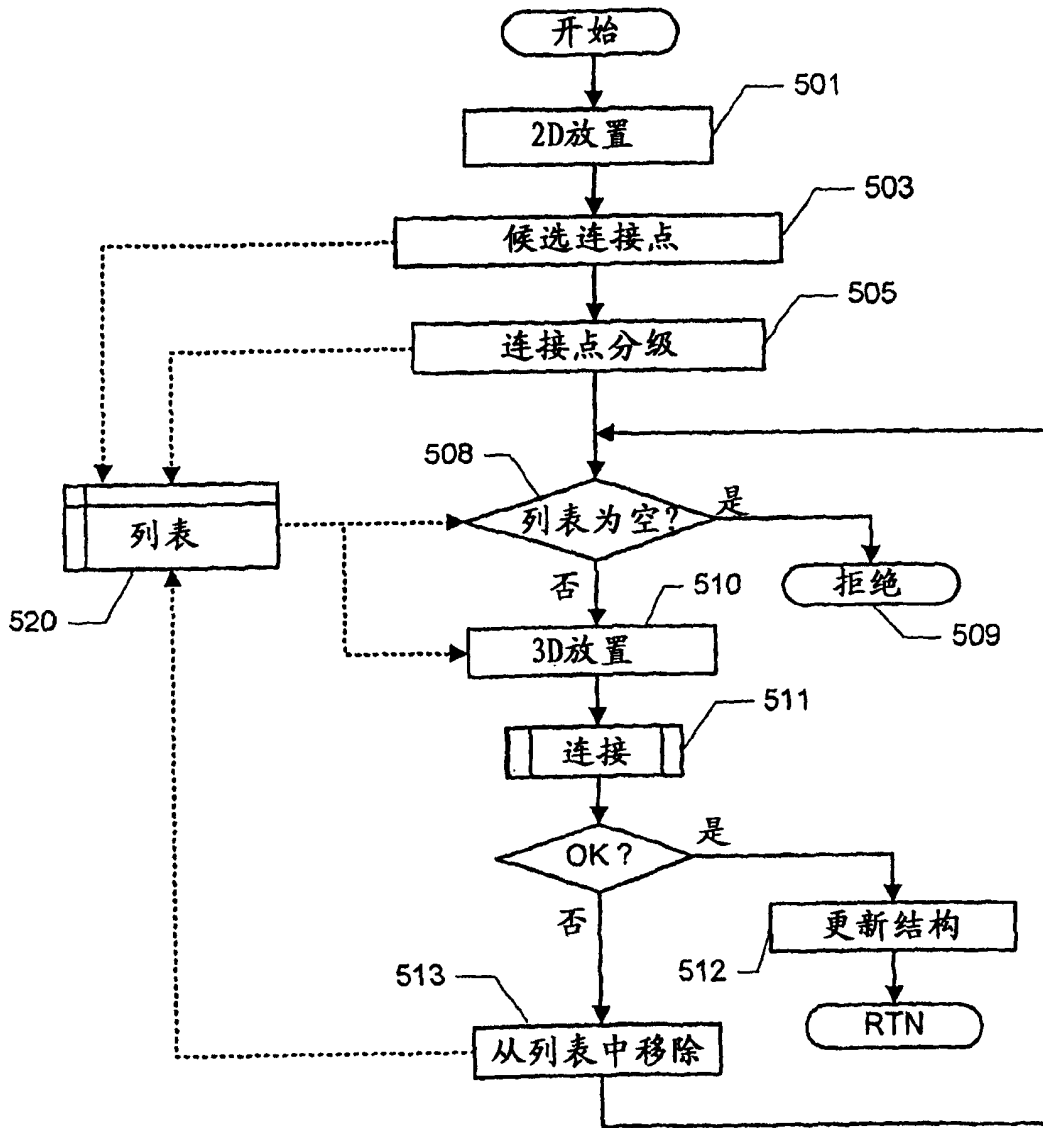


图5

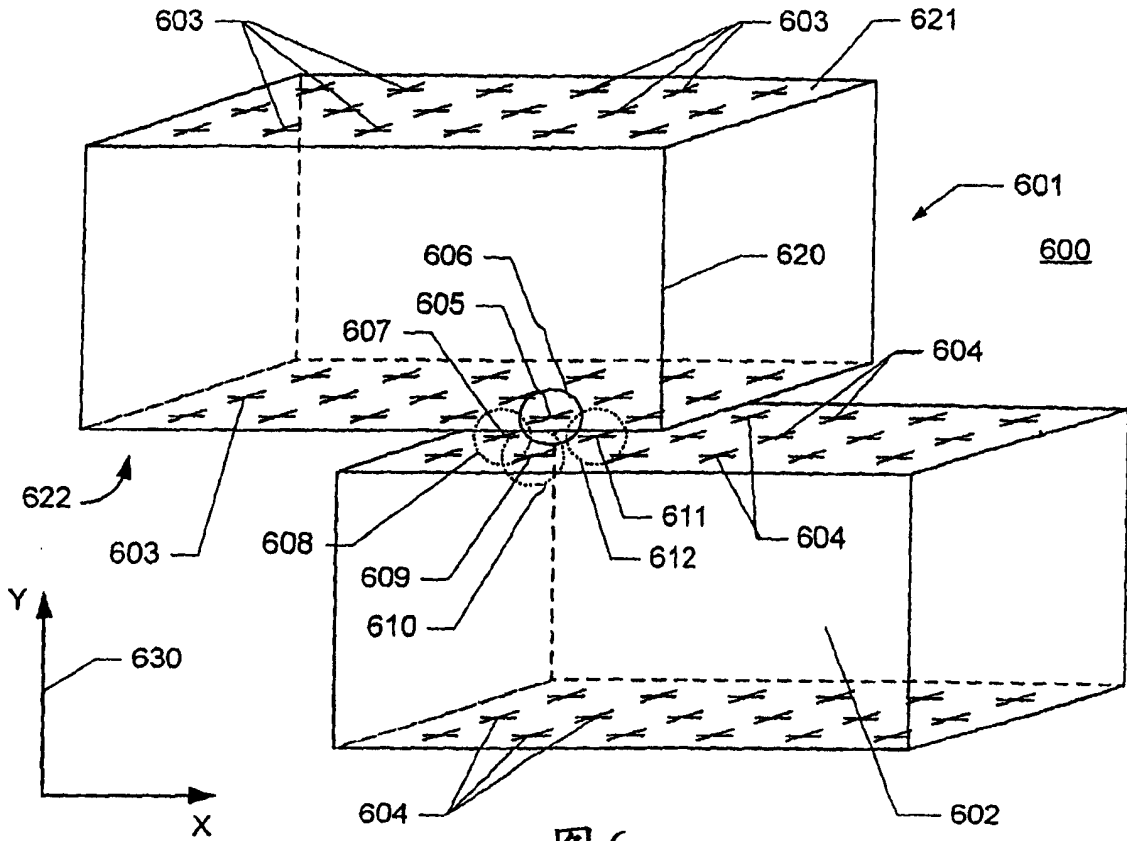


图 6a

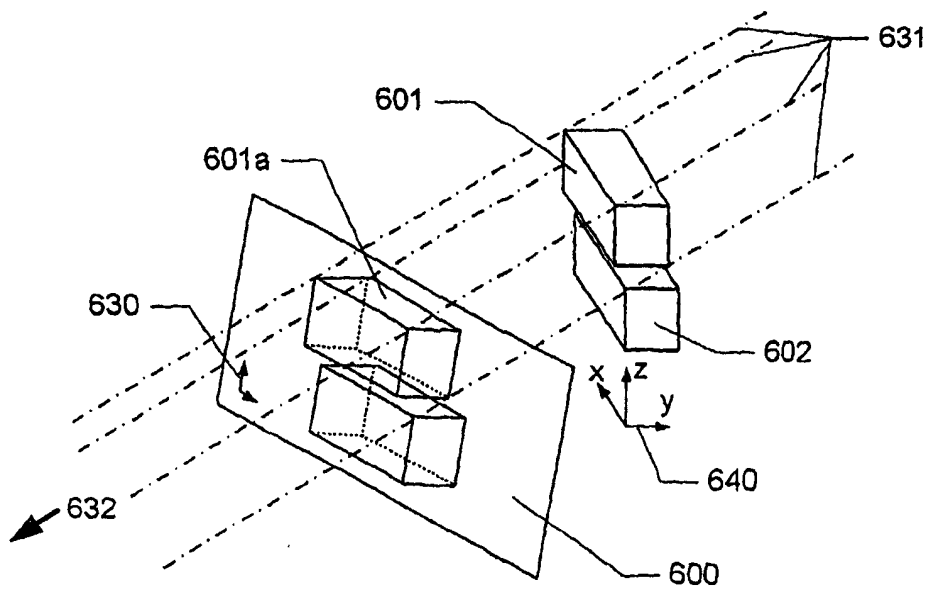


图 6b

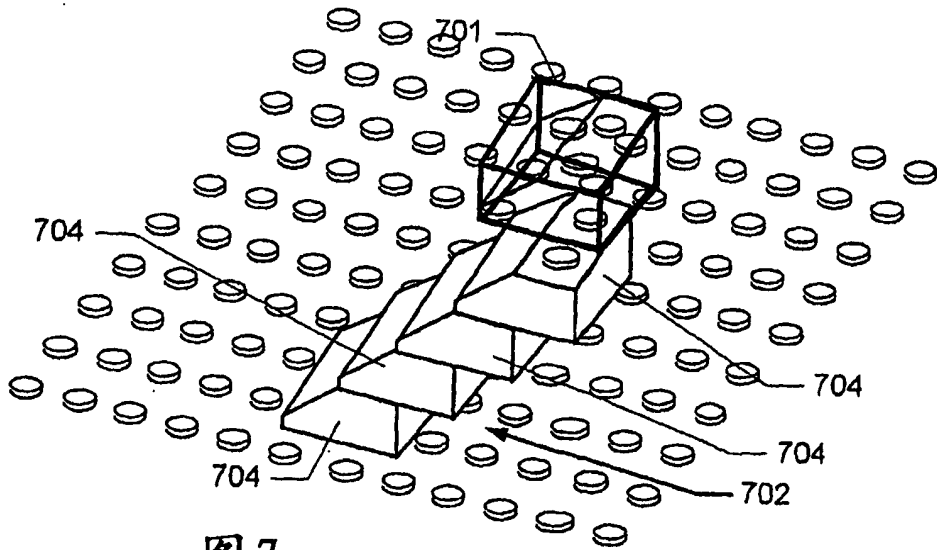


图 7a

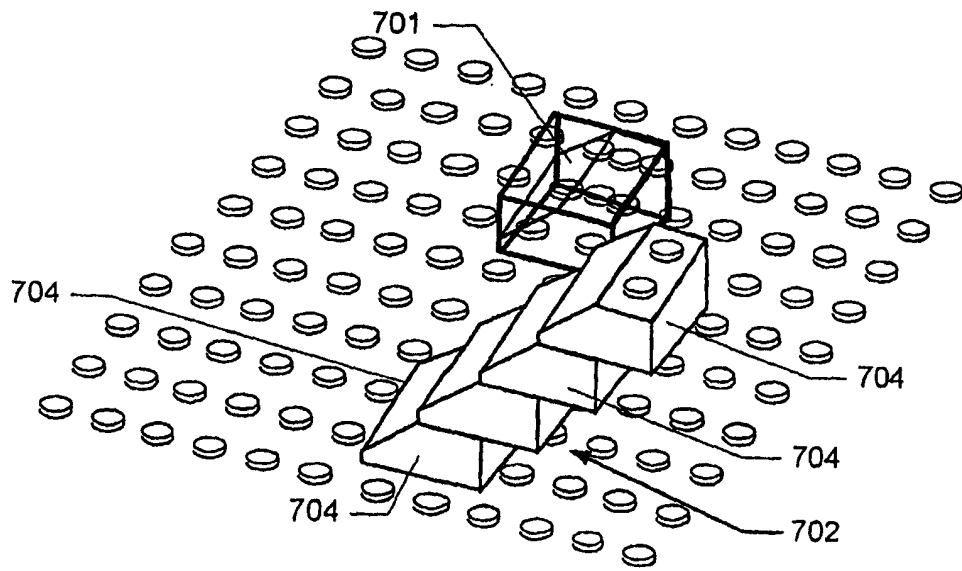


图 7b

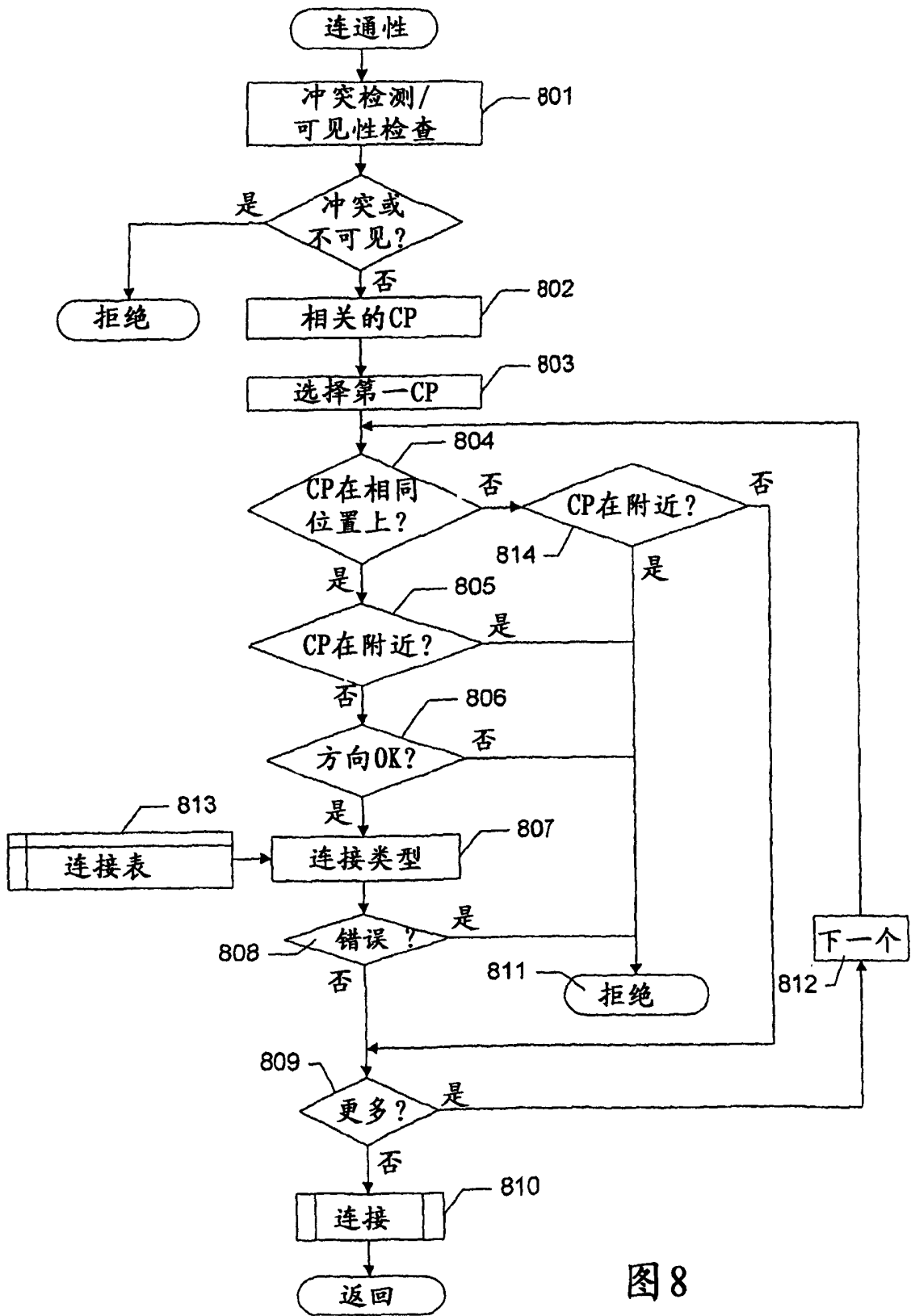


图8

