



(12) 发明专利

(10) 授权公告号 CN 102622776 B

(45) 授权公告日 2016. 01. 13

(21) 申请号 201210021228. X

(22) 申请日 2012. 01. 30

(30) 优先权数据

13/017, 690 2011. 01. 31 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 R·纽科姆 S·伊扎迪

D·莫利尼奥克斯 O·希利格斯

D·金 J·D·J·肖顿

S·E·豪杰斯 D·A·巴特勒

A·费茨吉本 P·科利

(74) 专利代理机构 上海专利商标事务所有限公

司 31100

代理人 顾嘉运

(51) Int. Cl.

G06T 17/00(2006. 01)

(56) 对比文件

CN 101605211 A, 2009. 12. 16, 全文.

US 2006/0006309 A1, 2006. 01. 12, 权利要求 7、9 及说明书第 1 页第 [0006] 段至第 5 页第 [0084] 段、第 9 页 [0123]-[0125] 段及说明书附图 1.

US 6084979 A, 2000. 07. 04, 说明书第 5 栏第 13 行至第 10 栏第 36 行, 说明书附图 8-11.

审查员 刘洋

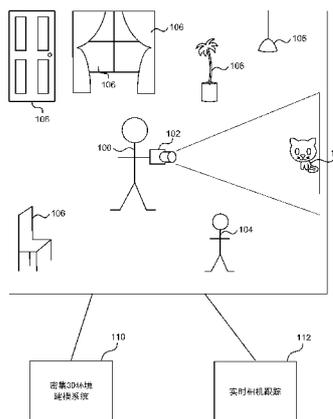
权利要求书2页 说明书12页 附图8页

(54) 发明名称

三维环境重构

(57) 摘要

描述了三维环境重构。在一示例中, 在由存储在存储器设备上的体素组成的 3D 容器中生成真实世界环境的 3D 模型。该模型从描述相机位置和定向的数据以及具有指示相机离环境中的一个点的距离的像素的深度图像中构建。单独的线程被分配给容体的平面中的每一个体素。每一个线程使用相机位置和定向来确定其相关联的体素的对应的深度图像位置, 确定与相关联的体素和环境中的对应位置处的点之间的距离相关的因子, 并且使用该因子来更新相关联的体素处的存储值。每一个线程迭代通过容体的其余平面中的等价体素, 从而重复该过程以更新存储值。



1. 一种生成真实世界环境的 3D 模型的计算机实现的方法,包括:

在存储器设备 (810、816) 中创建 (400) 用于存储所述模型的三维容体 (500),所述容体 (500) 包括多个体素 (512、610);

接收 (402) 描述捕捉设备 (102、204、302、306) 的位置和定向的数据以及从所述捕捉设备 (102、204、302、306) 输出的所述环境的至少一部分的深度图像 (314),所述深度图像 (314) 包括多个像素,每一个指示从所述捕捉设备 (102、204、302、306) 到所述环境中的一点 (614) 的距离;

将单独的执行线程分配 (404) 给所述容体 (500) 的平面中的每一个体素 (512、610);

每一个执行线程使用所述捕捉设备 (102、204、302、306) 位置和定向来确定其相关联的体素 (512、610) 在所述深度图像 (314) 中的对应位置,确定与所述相关联的体素 (512、610) 和所述环境中的所述对应位置处的所述点 (614) 之间的距离相关的因子,并且使用所述因子来更新所述相关联的体素 (512、610) 处存储的值;以及

每一个执行线程迭代通过所述容体 (500) 的每一个剩余平面中的等价体素 (512、610) 并且重复确定因子和更新所述相关联的体素 (512、610) 处存储的值的步骤,其中所述使用所述因子来更新所述相关联的体素处存储的值的步骤包括将所述因子与存储在所述相关联的体素处的先前值求和,或者其中所述确定所述因子的步骤包括计算所述相关联的体素和所述环境中的所述对应位置处的所述点之间的带符号的距离函数,以使得如果所述相关联的体素位于所述点的第一侧,则所述因子具有正值,并且如果所述相关联的体素位于所述点的相对侧,则所述因子具有负值。

2. 如权利要求 1 所述的方法,其特征在于,还包括以下步骤:在确定所述因子之前,确定所述对应位置是否在捕捉设备截锥体内。

3. 如权利要求 1 所述的方法,其特征在于,所述使用所述捕捉设备位置和定向来确定其相关联的体素在所述深度图像中的对应位置的步骤包括:

确定所述相关联的体素在真实世界坐标中的位置;

使用所述捕捉设备位置和定向通过所述捕捉设备的投影来投影所述真实世界坐标以确定所述相关联的体素在所述深度图像中的对应坐标。

4. 如权利要求 1 所述的方法,其特征在于,还包括生成所述模型的图像以便使用光线投射来显示在显示设备上的步骤,其中所述生成所述模型的图像的步骤包括:

接收虚拟相机的姿势数据;

将执行线程分配给每一个图像像素;

每一个图像像素执行线程确定从所述虚拟相机发出经过相关联的像素的光线的路径,将所述路径变换成所述容体中的坐标,沿所述路径移动通过所述容体直到检测到表面处的体素,并且使用该体素处存储的值来计算所述相关联的像素的阴影。

5. 如权利要求 1 所述的方法,其特征在于,还包括以下步骤:从所述模型中生成多边形网格并且将所述多边形网格存储在存储设备处,其中所述从所述模型中生成多边形网格的步骤包括使用行进立方体算法来选择表示所述模型的各部分的多边形。

6. 如权利要求 1 所述的方法,其特征在于,还包括以下步骤:使用所述存储的值和涉及所述相关联的体素离所述捕捉设备的距离的加权函数来确定加权均值。

7. 如权利要求 1 所述的方法,其特征在于,所述确定所述因子的步骤包括:

计算所述相关联的体素和所述环境中的所述对应位置处的所述点之间的所述带符号的距离函数,以使得如果所述相关联的体素位于所述点的第一侧,则所述因子具有所述正值,并且如果所述相关联的体素位于所述点的相对侧,则所述因子具有所述负值;以及

确定所述因子是否大于预定义的正阈值,并且如果是,则将所述因子舍位成所述正阈值;以及确定所述因子是否小于预定义的负阈值,并且如果是,则将所述因子舍位成所述负阈值。

8. 一种用于生成真实世界环境的 3D 模型的图形处理系统 (804), 包括:

通信接口 (806), 所述通信接口被安排成接收描述捕捉设备 (102、204、302、306) 的位置和定向的数据以及从所述捕捉设备 (102、204、302、306) 输出的所述环境的至少一部分的深度图像 (314), 所述深度图像 (314) 包括多个像素, 每一个指示从所述捕捉设备 (102、204、302、306) 到所述环境中的一点 (614) 的距离;

被安排成提供多个线程的并发执行的图形处理单元 (808); 以及
存储器设备 (810、816),

其中所述图形处理单元 (808) 被安排成: 在所述存储器设备 (810、816) 中创建 (400) 用于存储所述模型的三维容体 (500), 所述容体 (500) 包括多个体素 (512、610); 将单独的线程分配 (404) 给所述容体 (500) 的平面中的每一个体素 (512、610); 以及对于每一个执行线程, 使用所述捕捉设备 (102、204、302、306) 位置和定向来确定其相关联的体素 (512、610) 在所述深度图像 (314) 中的对应位置, 确定与所述相关联的体素 (512、610) 和所述环境中的所述对应位置处的所述点 (614) 之间的距离 (616) 相关的因子, 使用所述因子来更新所述相关联的体素 (512、610) 处存储的值, 并且迭代通过所述容体 (500) 的每一个剩余平面中的等价体素 (512、610) 并重复对所述因子的确定以及对所述相关联的体素 (512、610) 处存储的值的更新,

其中所述图形处理单元被安排成使用所述因子来更新所述相关联的体素处存储的值包括所述图形处理单元被安排成将所述因子与存储在所述相关联的体素处的先前值求和, 或者其中所述图形处理单元被安排成确定所述因子包括所述图形处理单元被安排成计算所述相关联的体素和所述环境中的所述对应位置处的所述点之间的带符号的距离函数, 以使得如果所述相关联的体素位于所述点的第一侧, 则所述因子具有正值, 并且如果所述相关联的体素位于所述点的相对侧, 则所述因子具有负值。

9. 如权利要求 8 所述的系统, 其特征在于, 所述捕捉设备是按一帧速率提供多个深度图像的视频相机, 并且所述图形处理单元还被安排成按所述帧速率重复更新所述容体中的所述相关联的体素处存储的值。

10. 如权利要求 8 所述的系统, 其特征在于, 所述捕捉设备包括深度相机, 所述深度相机包括以下各项中的至少一个: 结构化光发射器; 飞行时间相机; 以及立体相机。

三维环境重构

技术领域

[0001] 本发明涉及三维环境建模。

背景技术

[0002] 真实世界环境的三维计算机模型在各种各样的应用中是有用的。例如,这样的模型可以用在诸如沉浸式游戏、增强现实、建筑 / 规划、机器人技术、以及工程原型制作等应用中。

[0003] 深度相机 (也称为 z 相机) 可以生成真实世界环境的实时深度图。这些深度图中的每一像素与由相机从环境中的 3D 点捕捉到的离散的距离测量相对应。这意味着这些相机以实时速率提供包括未经排序的一组点 (称为点云) 的深度图。

[0004] 然而,使用深度相机提供的深度图来生成环境的有用 3D 模型是复杂的。在具有许多深度相机的情况下,每一个单独的深度图具有太低的分辨率并且具有太高的噪声以至于无法生成足够准确和详细的 3D 模型。此外,许多应用利用感测到的环境的表面而不是点云来重构。例如,重构的 3D 表面模型可以在物理模拟中用来对虚拟物体和深度相机在环境中感测到的真实物体之间的物理交互的效果建模。在这种情况下,点云不能起到很好的作用,因为模型中都是洞,这是由于在点云中的各个点之间缺少连接。

[0005] 通过将来自深度相机的点云的多个帧组合成连贯的可用模型以及确定这些点云中的点之间的连接来推断表面几何结构在计算上是非常密集的。表面重构是通过推断该连接来提取对相机可见的表面的过程。给定从深度相机实时接收的数据量以及支持交互式速率的需求,诸如泊松 (Poisson) 表面重构等已知算法在计算上太昂贵以致于无法支持实时处理。

[0006] 以下描述的各实施例不限于解决已知 3D 重构技术的缺点中的任一个或全部的实现。

发明内容

[0007] 下面提供本发明的简要概述以便向读者提供基本的理解。本概述不是本发明的详尽概观,并且既不标识本发明的关键 / 重要元素,也不描绘本发明的范围。其唯一目的是以简化形式提供在此公开一些概念作为稍后提供的更详细描述的前言。

[0008] 描述了三维环境重构。在一示例中,在由存储在存储器设备上的体素组成的 3D 容器中生成真实世界环境的 3D 模型。该模型从描述相机位置和定向的数据以及具有指示相机离环境中的一个点的距离的像素的深度图像中构建。单独的线程被分配给容体的平面中的每一个体素。每一个线程使用相机位置和定向来确定其相关联的体素的对应的深度图像位置,确定与相关联的体素和环境中的对应位置处的点之间的距离相关的因子,并且使用该因子来更新相关联的体素处的存储值。每一个线程迭代通过容体的其余平面中的等价体素,从而重复该过程以更新存储值。

[0009] 许多附带特征将随着参考下面的详细描述并结合附图进行理解而得到更好的认

识。

附图说明

[0010] 根据附图阅读以下详细描述,将更好地理解本发明,在附图中:

[0011] 图 1 示出房间中的持有移动深度相机的人的示意图,该移动深度相机被用于生成该房间的 3D 模型或图;

[0012] 图 2 示出由持有移动深度相机的人来探测的建筑物某层的平面视图;

[0013] 图 3 示出连接到实时相机跟踪系统、密集 3D 模型形成系统、以及游戏系统的移动深度相机的示意图;

[0014] 图 4 示出了用于生成 3D 环境模型的可并行化过程的流程图;

[0015] 图 5 示出了线性铺砌存储器 (linear pitched memory) 的 3D 容器构造;

[0016] 图 6 示出了示例可并行化的、带符号的距离函数计算;

[0017] 图 7 示出了用于从 3D 容器中进行光线投射的可并行化过程的流程图;

[0018] 图 8 示出了可在其中实现可并行化 3D 建模技术的各实施例的示例性的基于计算的设备。

[0019] 在各个附图中使用相同的附图标记来指代相同的部件。

具体实施例

[0020] 下面结合附图提供的具体实施例旨在作为本发明示例的描述,并不旨在表示可以构建或使用本发明示例的唯一形式。本描述阐述了本发明示例的功能,以及用于构建和操作本发明示例的步骤的序列。然而,可以通过不同的示例来实现相同或等效功能和序列。

[0021] 虽然在本文中本发明的示例描述并示出为在计算机游戏系统中实现,但所描述的系统只是作为示例而非限制来提供的。本领域的技术人员将理解,本示例适于在使用 3D 模型的各种不同类型的计算系统中应用。

[0022] 图 1 是站在房间中并持有移动深度相机 102 的人 100 的示意图。该房间包含各种对象 106,如椅子、门、窗、植物、灯、另一个人 104、以及一只猫 108。对象 106 中的许多是静态的,但这些对象中的一些可以移动,如人 104 和猫 108。当人在房间中四处移动时,移动深度相机捕捉该房间和各对象的深度图像。

[0023] 移动深度相机 102 与密集 3D 环境建模系统 110 进行通信(在该情况下,环境是该房间)。在该示例中,“密集”指的是从使用来自传入深度图的全部或许多点来隐式地描述表面中产生的模型的高准确度和高分辨率。这可以与只采用点的子集来加速计算和减少存储器占用的“稀疏”模型进行对比。例如,当人在房间中四处移动时,移动深度相机 102 捕捉到的图像被用来形成和构建环境的密集 3D 模型。

[0024] 实时相机跟踪系统 112 监视相机在房间中的位置和定向。实时相机跟踪系统 112 可以与移动深度相机 102 集成,或可以处于另一位置,只要它能够(直接或间接)接收来自移动深度相机 102 的通信。例如,实时相机跟踪系统 112 可以在房间中的个人计算机、专用计算机游戏装置、或其他计算设备处提供,并与移动深度相机 102 进行无线通信。在其他实施例中,实时相机跟踪系统 112 可以处于建筑物中的别处或处于使用任何合适类型的通信网络与移动深度相机 102 进行通信的另一远程位置。

[0025] 实时相机跟踪系统 112 向密集 3D 建模系统提供输入,以允许各单独的深度图像被构建到总体 3D 模型中。实时相机跟踪系统 112 还可跟踪相机相对于环境的 3D 模型的位置。相机跟踪和 3D 建模的组合被称为同时定位和绘图 (SLAM)。注意,在其他示例中,绘图和定位还能够例如通过使用独立于绘图的相机定位技术来单独执行 (即,不使用 SLAM 系统)。

[0026] 实时相机跟踪系统 112 的输出和密集 3D 建模系统 110 可由游戏系统或其他应用来使用,但这不是必要的。例如,经建模的真实世界对象可被包括在游戏环境中。在另一示例中,移动深度相机 102 处的投影仪可被安排成取决于实时相机跟踪系统 112 的输出和 3D 建模系统 110 来投影图像。

[0027] 作为另一示例,图 2 示出建筑物某层 200 的平面视图。持有移动深度相机 204 的人 202 正如虚线箭头 208 所示在该层四处移动。此人沿走廊 206 走过各个房间和家具 210。实时相机跟踪系统 112 能够在移动深度相机 204 移动时跟踪它的位置,并且 3D 建模系统生成该层的 3D 模型或图。人 202 不必携带移动深度相机 204。在其他示例中,移动深度相机 204 被安装在机器人或运载工具上。这也适用于图 1 的示例。

[0028] 图 3 是与实时相机跟踪器 316、密集模型形成系统 324 和可任选的游戏系统 332 一起使用的移动环境捕捉设备 300 的示意图。移动环境捕捉设备 300 包括被安排成捕捉场景的深度图像序列的深度相机 302。每一深度图像 314 (即,深度图帧) 包括二维图像,该二维图像中的每一图像元素 (即,像素) 包括诸如从相机到所捕捉的场景中的造成了该图像元素的对象的长度或距离等深度值。这一深度值可以是以指定测量单位 (如米或厘米) 来提供的绝对值,或可以是相对深度值。在每一捕捉到的深度图像 314 中,存在大约 300000 或更多个图像元素,每一图像元素具有深度值。深度相机 302 的帧速率足够高,使得深度图像能够被用于机器人技术、计算机游戏、或其他应用。例如,帧速率范围可以是每秒 20 到 100 帧。

[0029] 深度信息可以使用任何合适的技术来获得,包括但不限于飞行时间、结构化光、以及立体图像。移动环境捕捉设备 300 还可包括被安排成以可由深度相机 302 查明深度信息的方式来照亮该场景的发光器 304。

[0030] 例如,在深度相机 302 是红外 (IR) 飞行时间相机的情况下,发光器 304 将 IR 光发射到该场景上,并且深度相机 302 被安排成检测从该场景中的一个或多个对象的表面反向反射的光。在一些示例中,可以从发光器 304 发出脉冲红外光,使得外出光脉冲与对应的传入光脉冲之间的时间可由深度相机来检测和测量,并被用来确定从环境捕捉设备 300 到场景中的对象上的位置的物理距离。另外,在一些示例中,可将来自发光器 304 的出射光波的相位与入射光波的相位在深度相机 302 处进行比较来确定相移。随后经由包括例如快门式光脉冲成像的各种技术通过分析发射光束随时间的强度来将相移用于确定从移动环境捕捉设备 300 到各对象上的位置的物理距离。

[0031] 在另一示例中,移动环境捕捉设备 300 可使用结构化光来捕捉深度信息。在这种技术中,可使用发光器 304 将图案化光 (例如,显示为诸如斑点、网格、或条形图案等已知图案的光,它也可随时间变化) 投影到场景上。在到达场景中的物体的表面之后,该图案变形。深度相机 302 捕捉图案的这种变形并对其进行分析来确定从深度相机 302 到场景中的对象的绝对或相对距离。

[0032] 在另一示例中,深度相机 302 包括一对立体相机,以便获得并解析视觉立体数据

来生成相对深度信息。在这种情况下,发光器 304 可被用来照亮场景或可被省略。

[0033] 在一些示例中,除深度相机 302 之外,移动环境捕捉设备 300 包括被称为 RGB 相机 306 的彩色视频相机。RGB 相机 306 被安排成在可见光频率处捕捉场景的图像序列。

[0034] 移动环境捕捉设备 300 可包括定向传感器 308,诸如惯性测量单元(IMU)、加速计、陀螺仪、指南针、或其他定向或移动传感器。但是,并非必须使用定向或移动传感器。移动环境捕捉设备 300 可包括诸如 GPS 等位置跟踪设备,但这不是必须的。

[0035] 移动环境捕捉设备可包括如上所述的投影仪 312,但这不是必需的。移动环境捕捉设备还包括一个或多个处理器、存储器、以及通信基础结构,如下文更详细地描述的。移动环境捕捉设备可以在外壳中提供,该外壳的形状和大小制定成由用户手持或由用户佩戴。在其他示例中,移动环境捕捉设备的大小和形状被制定成被包括或安装在运载工具、玩具、或其他可移动装置上。

[0036] 移动环境捕捉设备 300 连接到实时跟踪器 316。这一连接可以是物理有线连接或可以使用无线通信。在一些示例中,移动环境捕捉设备 300 通过诸如因特网等一个或多个通信网络间接连接到实时跟踪器 316。

[0037] 实时跟踪器 316 使用控制一个或多个图形处理单元(GPU)的通用微处理器来用计算机实现。它包括帧对齐引擎 318 并且可任选地包括环闭合引擎 320 和重新定位引擎 322。实时跟踪器 316 从深度相机 302 获得深度图像帧,并还可任选地从移动环境捕捉设备 300 获得输入,并可任选地获得地图数据 334。实时跟踪器 316 可用于使深度图像帧在空间对齐,以产生深度相机 302 的六自由度(6DOF)姿势估计的实时系列 328。它还产生各对深度图像帧之间的变换的变换参数。在一些示例中,实时跟踪器操作来自深度相机的各对深度图像帧。在其他示例中,实时跟踪器 216 获取深度图像 314 并将它与该环境的密集 3D 模型 326 对齐而非与另一深度图像对齐。

[0038] 实时跟踪器 316 将相机姿势作为输出提供给密集 3D 模型生成系统 324,模型生成系统 324 使用该信息和深度图像帧来形成并存储移动环境捕捉设备 300 在其中移动的场景或环境的密集 3D 模型 326。这在以下参考图 4 到 6 更详细地描述。例如,在图 1 的情况下,3D 模型将是房间中的表面和物体的 3D 模型。在图 2 的情况下,3D 模型将是建筑物的该层的 3D 模型。密集 3D 模型可以被存储在 GPU 存储器中,如下所述。密集 3D 模型可以实时重构,或者密集 3D 模型生成系统 324 可收集关于相机姿势和深度图像的数据并在稍后重构 3D 模型。

[0039] 移动环境捕捉设备 300 可结合游戏系统 332 来使用,游戏系统 332 连接到显示设备 330。例如,游戏可以是 FPS(第一人称射击)游戏、高尔夫游戏、拳击游戏、赛车游戏、或其他类型的计算机游戏。密集 3D 模型可被提供给游戏系统 332,并且该模型的各方面被结合到该游戏中。例如,3D 模型可被用来确定房间中各对象的形状和位置,这可与基于相机的游戏一起使用来改进背景移除或合并到游戏本身中(例如,作为玩家可与之交互的游戏对象)。来自游戏系统 332 的数据(如与游戏相关的游戏状态或元数据)也可被提供给实时跟踪器 316。

[0040] 为了帮助用户构建密集 3D 模型,能够以密集 3D 模型的实时呈现的形式向用户提供反馈。这使用户能够在重构密集模型时查看该模型的状态,由此帮助用户捕捉已被遗漏的区域。这为用户提供更具交互性的体验。以下参考图 7 讨论用于从密集 3D 模型中呈现

视图的示例技术。还可向用户提供音频或视觉提示以将用户引导至密集 3D 模型中的已被遗漏或具有糟糕质量的区域。

[0041] 可任选地,图数据 334 对实时跟踪器 316 是可用的。例如,这可以是架构师对该环境(例如,房间或建筑物的层)的绘图、该环境中已知的地标的位置、可从另一源获得的该环境的图。

[0042] 实时跟踪器的帧对齐引擎 318 被安排成对齐各对深度图像帧,或深度图像帧和来自密集 3D 模型的深度图像帧的估计。它使用迭代过程,该迭代过程是用一个或多个图形处理单元来实现的,以使帧对齐引擎实时操作。环闭合引擎 230 被安排成检测移动环境捕捉设备何时环形移动,使得当前深度图像中描绘的场景至少部分与已经被集成到模型中的先前深度帧的场景相重叠。例如,这可在用户在图 2 中的建筑物的整层四处走动并再次到达起始点时发生。这也可以在用户在房间四处移动至某家具背后并再次出来到原始起始位置或接近该原始起始位置时发生。重新定位引擎 322 被安排成处理以下情况:实时跟踪器失去移动环境捕捉设备 300 的当前位置并且重新定位或再次找出当前位置。

[0043] 如上所述,在一个示例中,实时跟踪器 316 和 / 或密集 3D 模型生成系统 324 所执行的处理可以在移动环境捕捉设备 300 的远程位置处执行。例如,移动环境捕捉设备 300 可被连接到(或包括)具有相对低处理能力并且通过通信网络将深度图像流传送到服务器的计算设备。服务器具有相对高的处理能力,并且执行实时跟踪器 316 和 / 或密集 3D 模型生成系统 324 的在计算上复杂的任务。该服务器可以逐帧返回密集重构的已渲染图像以向用户提供交互体验,并且在模型完成时还返回最终密集 3D 重构以供后续本地使用(例如,在游戏中使用)。这样的安排避免了用户对拥有高能力本地计算设备的需要。

[0044] 在另一示例中,处理可以在本地和远程计算设备之间划分。例如,深度图像可通过通信网络传送到服务器以重构高分辨率密集 3D 模型,该模型在完成时返回,如上所述。与此同时,使用较少的处理能力和存储器来本地地构造模型的低分辨率版本,该版本用于呈现该模型的实时视图以供用户查看。这使用户能够从本地处理器获取来自模型构造的视觉反馈,以避免网络等待时间问题。

[0045] 现在参考图 4,图 4 示出了用于生成 3D 环境模型的可并行化过程的流程图。在一个示例中,图 4 的处理在 GPU 处理器、多核处理器或其他类型的处理器上执行以允许高度执行并行化。在另一示例中,还可使用单核处理器,如果它足够快的话。首先,在用于存储模型的存储器设备上创建 3D 容器 400。在一个示例中,3D 容器在允许来自 GPU 的快速、并行访问的图形处理系统的存储器上创建。

[0046] 该 3D 容器可被可视化为存储器的立方体,其中每一个存储器位置是表示正在建模的环境的空间中的点的体素。因此,3D 容器直接表示真实世界环境的空间部分。由于 3D 容器直接对应于真实世界容器,因此固定大小的存储器中所表示的真实世界容器的大小确定模型分辨率。例如,如果对较大的真实世界容器建模,则存储器的每一个体素表示真实世界空间中的较大区域,并因此分辨率低于对较小的真实世界容器建模的情况下的分辨率。然而,如果更多存储器可用,则能够以较高的分辨率对较大的真实世界容器建模。还可使用高效数据存储策略和 / 或压缩算法来减少存储器占用并提高空间分辨率。

[0047] 图 5 示出了如何能够在 GPU 存储器中构造 3D 容器的示例。图 5 示出了示例 4x4x43D 容器 500。这仅仅是说明性示例,并且实际 3D 容器可以更大或更小。3D 容器具有 x 轴 502、

y 轴 504 和 z 轴 506。沿 x 轴 502 的体素数量定义容体宽度 508(该示例中为四),而沿 y 轴 504 的体素数量定义容体高度 510(该示例中也是四)。然后可使用 x、y、z 坐标来标识各个体素。例如,体素 512 具有坐标 (0, 3, 2)。因为 3D 容体与其正在表示的真实世界容体的大小之间的关系是已知的,所以 3D 容体中的体素可使其位置在 3D 容体坐标和真实世界坐标之间转换。

[0048] 然而,用于存储模型的物理存储器设备通常不被安排为诸如图 5 所示的立方体。相反,存储器设备通常包括诸如存储器阵列 514 等存储器位置的线性阵列。为了在存储器阵列上创建 3D 容体,每一个体素都通过以下方式被映射到存储器阵列索引 : $x+(y \times \text{宽度})+(z \times \text{宽度} \times \text{高度})$ 。例如,对于具有坐标 (0, 3, 2) 的体素 512,它转换成索引 44,如图 5 所示。该安排被称为线性铺砌存储器,并且提供对存储在 GPU 存储器上的数据的快速、并行访问,如下所述。还能够对齐存储在该存储器中的数据以使得数据的存储地址按二次幂跨距间隔,例如通过填充数据以便与这些边界对齐。这使得能够对数据执行处理而不使用普通乘法,由此进一步提高计算速度。

[0049] 注意,在替换示例中,模型能够以与诸如上文中描述的 3D 容体不同的方式存储。例如,诸如八叉树等数据结构可用于进一步减少存储器消耗并加速计算。或者,可以应用 3D 空间和更低维表示之间的转换函数来提高存储效率。这种类型的转换函数的示例被称为“几何结构图像”。

[0050] 再次回到图 4,一旦在存储器中创建 4003D 容体,该建模过程就通过从移动环境捕捉设备 300 接收深度图像 314 以及对深度相机 302 在捕捉该深度图像时的 6DOF 姿势估计。该 6DOF 姿势估计指示深度相机 302 的位置和定向,并且可由实时跟踪器 316 来提供。在一个示例中,6DOF 姿势估计可具有 SE3 矩阵的形式,该矩阵描述深度相机 302 相对于真实世界坐标的旋转和平移。此变换矩阵可以被更加正式地表示为:

[0051]

$$\mathbf{T}_k = \begin{bmatrix} \mathbf{R}_k & \mathbf{t}_k \\ \mathbf{0}^\top & 1 \end{bmatrix} \in \mathbf{SE}_3$$

[0052] 其中 \mathbf{T}_k 是深度图像帧 k 的变换矩阵, \mathbf{R}_k 是帧 k 的相机旋转, \mathbf{t}_k 是帧 k 处的相机平移,而欧几里德群 $\mathbf{SE}_3 := \{\mathbf{R}, \mathbf{t} \mid \mathbf{R} \in \mathbf{SO}_3, \mathbf{t} \in \mathbb{R}^3\}$ 。通过乘以这个变换矩阵,可以将相机空间中的坐标(即,来自相机角度)映射到真实世界坐标,并且可通过逆变换来进行反向操作。

[0053] 单独的执行线程被分配给 3D 容体 500 的平面中的每一个体素 404。例如,如果容体的平面是 z 平面,则执行线程被分配给 3D 容体的 x 和 y 坐标的每一种组合。考虑图 5 的 3D 容体,这产生十六个线程,被分配给具有以下 x、y 坐标的体素 : (0, 0) ; (1, 0) ; (2, 0) ; ... (1, 3) ; (2, 3) ; (3, 3)。

[0054] 每一个单独的执行线程然后执行相同的操作,如图 4 中的框 406 所示。每一个线程执行的操作在图 6 中图解地示出。图 6 示出了 3D 容体 500 的自顶向下视图(即,示出 x 轴 502 和 z 轴 506,并且出于简明的目的忽略 y 轴。执行线程被分配给容体的 z 平面上的每一个坐标,被表示为 $T_1,602, T_2,604, \dots, T_{n-1},606, T_n,608$ 。框 406 中的每一个线程执行的操作首先参考 $T_1,602$ 示出。

[0055] 执行线程的 z 坐标被设为零 408,并且与该线程相关联的体素的 x, y, z 坐标被转

换 410 成真实世界坐标。例如,在线程 T_1 602 的情况下,与线程相关联的 x, y 坐标是 $(0, 0)$, 因此当 z 坐标被设为零时,这给出坐标 $(0, 0, 0)$ (即,图 6 中的体素 610),该坐标可被转换成真实世界坐标。一旦确定体素 610 的真实世界坐标,这些坐标就可使用深度相机的 6DOF 位置和定向来变换以确定深度图像 314 中与该体素相对应的位置。这意味着与线程相关联的当前体素的真实世界坐标是通过深度相机的投影来透视投影的,并且可以考虑相机的内在参数(被称为投影矩阵)。换言之,如果表示真实世界中的该点的体素正由具有其位置和定向的深度相机查看,则这确定深度图像 314 中对应于该点的坐标是什么。体素的深度相机角度坐标被表示为 pX, pY, pZ 。深度相机角度坐标可通过 $u = pX/pZ$ 和 $v = pY/pZ$ 被转换成深度图像像素坐标(被表示为 u, v)。

[0056] 深度相机 302 的角度在图 6 中用虚线示出,指示深度相机 302 的截锥体 612。图 6 还示出从深度相机输出的深度图像 314 中的表面 614。

[0057] 当前与线程相关联的体素的深度图像坐标 (u, v) 被用来确定 414 该体素是否位于深度相机的截锥体 612 内。如果否,则该体素可被跳过。参考图 6,注意,在线程 T_1 的情况下,当前体素 610 在相机的截锥体 612 内。这可以与例如线程 T_{n-1} 所考虑的在相机截锥体 612 之外并因此被跳过的第一个体素形成对比。

[0058] 如果确定当前体素在相机截锥体 612 内,则确定与该体素和环境中的与该体素相对应的位置处的点之间的距离相关的因子。环境中的对应位置处的点可由深度图像中的 (u, v) 处的深度值来给出。因此,参考图 6,该因子考虑体素 610 与深度图像 314 中的表面 614 上的映射到该体素 610 上的点之间的距离 616。

[0059] 在一个示例中,计算出的因子 420 是带符号的距离函数,但任何隐式表面表示可以是可适用的(例如,其中确切的表面位置可通过数字方法和规则网格上的有限数量的存储点之间的内插来确定的隐式表面表示)。带符号的距离函数计算给出当前体素和深度图像 314 中的对应点之间的距离的值,并且是带符号的以使得在深度图像中的对应点(从相机角度)之外(即外部)的体素被给予正距离,而在深度图像中的对应点(从相机角度)之内(即内部)的体素被给予负距离。值零指示相关联的体素恰好与对应点一致。带符号的距离函数可以根据深度图像中的 (u, v) 处的深度值减去 pZ 来容易地计算。

[0060] 例如,图 6 中的体素 610 在表面 614 上的对应点的前面,并因此具有带正号的距离函数值。相反,图 6 中的体素 618 在表面 614 上的对应点的后面,并因此具有带负号的距离函数值。

[0061] 带符号的距离函数值然后被规范化 422 为预定义距离值。在一个示例中,该预定义值可以是诸如 5cm 等较小距离,但可适用任何适合的值。例如,规范化可取决于正在重构的对象的噪声水平和厚度来自适应。这可通过用户手动定义,或通过分析数据中的噪声的自动导出。然后确定 424 规范化的距离是大于正阈值(如果带符号的距离为正)还是小于负阈值(如果带符号的距离为负)。如果是,则将带符号的距离函数值舍位 426 成最大或最小值。例如,如果规范化的距离大于正阈值,则该值可被舍位成 +1(规范化后的正阈值),并且如果规范化的距离小于负阈值,则该值可被舍位成 -1(规范化后的负阈值)。该计算的结果被称为舍位带符号距离函数(TSDF)。

[0062] 例如,参考图 6,体素 610 是表面 614 之外的相对较大的距离,并因此可被舍位成 +1。相反,体素 618 是表面 614 之内的相对较短的距离,并因此可以不舍位,并且保持其规

范化距离（例如，-0.6）。

[0063] 规范化的（并且在适当时候进行舍位）带符号距离函数值然后与存储在当前体素处的任何先前值进行组合。在这是被结合到 3D 容器中的第一个深度图像的情况下，不存在先前值。然而，在接收到并结合来自深度相机的其他帧时，这些值可能已经存在于体素处。

[0064] 在一个示例中，带符号的距离函数值通过求均值 428 来与先前值组合。这可以有助于构建具有移动对象的环境的模型，因为这使得已经移动的对象能够随时间在添加它的度量变旧时消失并且用更新近的度量来求均值。例如，可使用呈指数级衰减的正在移动的均值。在另一示例中，均值可以是使用涉及相关联的体素与深度相机的距离的加权函数的加权均值。平均的带符号距离函数值然后被存储 430 在当前体素处。

[0065] 在一替换示例中，可以在每一体素处存储两个值。可计算并存储带符号距离函数值的加权总和，并且还可计算和存储权重的总和。加权均值然后可被计算为（加权和）/（权重之和）。

[0066] 然后确定 416 该线程已经考虑 3D 容体的所有 z 平面。如果否，则递增 z 坐标 418，并且对 z 方向上的下一体素重复该过程。换言之，线程迭代通过沿容体的 z 方向的等价体素（即，具有相同 x, y 坐标的体素），确定舍位带符号距离函数值并且将它们与先前值求均值。

[0067] 重复这一操作直到线程迭代通过所有 z 平面。这由每一线程执行，每一个线程都被分配到 z 平面上的不同 x, y 坐标。对于图 5 中的存储器布局，这种一次一个容器平面的序列具有良好的存储器存取效率特性，因为一个 z 平面对应于线性铺砌阵列的较小的连续区域。因此，一旦所有线程都完成，就分析了整个 3D 容器，并且为相对于深度图像 314 的每一个容器计算舍位带符号距离函数。该过程然后等待从深度相机接收到下一深度图像，并且再次开始，从而在可能的情况下向 3D 容器添加更多值。

[0068] 该过程的结果是包括具有在 -1 和 +1 之间的平均值的体素的 3D 容器（但注意，该平均值可由诸如上文中的“加权和”和“权重之和”值等多个值来表示）。平均值零指示环境中的表面存在于该位置。因为深度图像的多个帧随着时间被添加到容器，逐渐越来越详细地构建存储在容器中的模型。对环境的相同区域的重复度量添加更多细节并且有效地提高深度相机提供的信息的分辨率。此外，噪声被从深度图像中有效地平均掉，从而留下平滑得多的表面。这使得能够在模型中看见无法从单独的深度图像中看见的细节。

[0069] 以此方式使用单独的线程使得能够对整个 3D 容器进行计算上高效的处理。例如，GPU 处理器被安排成高效地执行并行程序线程，其中每一线程执行相同的操作。这使得能够实时执行上文中参考图 4 描述的模型生成过程，以使得该模型以深度相机的帧速率构造。

[0070] 一旦如上所述的那样在 3D 容器中构造了模型，就能够以多种方式利用该模型。例如，可使用如图 7 所示的光线投射技术来呈现存储在容器中的模型视图。图 7 的流程图示出了用于从 3D 容器中进行光线投射的可并行化过程，该过程适于在 GPU 或多核 CPU 上以类似于上文中的模型生成过程的方式执行。

[0071] 为了呈现模型的视图，首先接收 700 定义要呈现的图像的视点的虚拟相机的姿势。该姿势可具有虚拟相机的 6DOF 位置和定向的形式。然后将单独的线程分配 702 给要呈现的图像中的每一个像素。

[0072] 然后由每一个执行线程执行框 704 中的操作以确定要应用于该线程的相关联的像素的值（例如，阴影、颜色等）。与线程相关联的像素的 x 和 y 坐标与预定义的初始 z 值组合以给出该像素的 3D 坐标，并且使用虚拟相机的姿势来将该像素的 3D x, y, z 坐标转换 706 成用 X, Y, Z 来表示的真实世界坐标。真实世界坐标 X, Y, Z 然后可被变换 708 成 3D 容器中的体素坐标。

[0073] 这些坐标定义具有从虚拟相机发出穿过 3D 容体的光线上的对应于像素的点。然后确定 710 3D 容器中的哪一个体素首先被该光线触碰到，并且该体素被设为光线投射的起始体素。光线投射操作（从起始体素）沿着光线移动 712，并且读取该光线触碰到的体素的平均带符号距离函数值，在每一体素处确定 714 是否出现体素值中的零交叉（即，存储在光线上的一个体素中的平均带符号距离函数值与沿光线的下一体素之间的符号变化）。任选地，这可以被安排成确定只从正通过零到负的符号变化的存在。这使得能够在从正面看到的表面和从对象“内部”看到的表面之间作出区分。

[0074] 当检测到零交叉时，这指示在模型中存在表面（如上所述）。因此，这指示表面与光线交叉处的体素。能够在光线上的体素的带符号距离函数值之间使用合适的线性内插技术，可使用零交叉的任一侧来更精确地定位零交叉，并因此更精确地定位与零交叉表面的交叉。在一个示例中，给出三线性采样的点在所检测的零交叉的任一侧上以找到零出现处的点，可以使用简单线性内插来计算沿光线的表面交叉点。在出现零交叉的点处，计算 716 表面法线。这可使用来自相邻体素的梯度来执行。线性内插还可用于更准确地确定来自相邻体素的梯度，并因此生成更准确的表面法线。在一个示例中，表面法线可使用后向差分数学导数来计算，如下：

[0075]

$$\hat{\mathbf{n}}(\mathbf{x}) = \frac{\nabla f(\mathbf{x})}{\|\nabla f(\mathbf{x})\|}, \nabla f = \left[\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}, \frac{\partial f}{\partial z} \right]^T$$

[0076] 其中 $\hat{\mathbf{n}}(\mathbf{x})$ 是体素 \mathbf{x} 的法线，并且 $f(\mathbf{x})$ 是体素 \mathbf{x} 的带符号的距离函数值。该导数能够在每一维度伸缩以确保在给定可能任意的体素分辨率和重构维度的情况下的正确各向同性。

[0077] 将出现零交叉处的体素的坐标转换 718 成真实世界坐标，从而给出模型中的表面的位置的真实世界坐标。根据表面的真实世界坐标加上其表面法线，可计算 720 阴影和 / 或颜色。计算出的阴影和 / 或颜色可基于任何合适的加阴影模型，并且考虑虚拟光源的位置。

[0078] 如上所述，框 704 中的操作由每一个执行线程并行地执行，这给出最终输出图像中的每一像素的阴影和 / 或颜色。然后可以组合为每一像素计算出的数据以给出输出图像 722，该输出图像是从虚拟相机的模型视图的呈现。

[0079] 上述光线投射操作的并行特质使得能够实时呈现 3D 环境的视图。当模型被存储在 GPU 存储器中的 3D 容器中时，光线投射操作在 GPU 上的执行是快速的且在计算上是高效的。

[0080] 该光线投射操作还可通过采用空白空间跳过算法来进一步增强。因为每一光线可以在其路径上触碰到数百个体素，所以通过分析空体素（即，不包含关于模型的任何信息的那些体素）来避免浪费计算资源是有益的。在一个示例中，为了避免分析沿光线的每一

单个体素,如果体素处的值大于预定义值,则跳过沿该光线的不止一个体素。跳过的体素的数量以及预定义值可被定义为相对于光线投射准确性来高效地折衷,只要这些参数是使用上述预定义距离值的知识来选择的(控制规范化)以确保不会由于空间跳过而遗漏零交叉。

[0081] 在另一示例中,空间跳过可通过利用诸如N元树(例如,八叉树)等多级数据结构来执行。此处,生成来自3D容体的具有比该3D容器更低的分辨率的至少一个附加容器。可以首先分析低分辨率容器以确定沿光线哪里有空白空间,并且可分析主3D容器以获悉光线中通过非空体素的各部分。

[0082] 在其他示例中,可使用存储在3D容器中的模型来提取环境的多边形网格表示。这可例如使用行进立方体算法(marching cubes algorithm)、行进四面体算法(marching tetrahedrons algorithm)、Bloomberg多边形化器(Polygonizer),或用于选择表示模型的各部分的多边形的任何其他合适的算法来实现。模型的多边形表示对于基于物理的应用或利用恒定的“水密(water-tight)”表面的其他应用而言可以是有用的。

[0083] 在一些示例中,多边形网格可以是表示模型的更存储高效的方式,但无法用来自深度相机的新度量来容易地更新。多边形网格因此可用于在其已经在3D容器中从深度相机中创建后高效地存储模型。例如,多边形网格可被存储在诸如CPU存储器等存储设备中或者存储在盘上,而不是存储在GPU存储器中。在一些示例中,模型能够在构造后以多边形网格的形式“高速缓存”,并且在需要时被转换回到带符号的距离函数形式。例如,可使用多边形网格来构造覆盖比能够适合在3D容器存储器中的区域更大的真实世界区域的大型模型。由于,模型在3D容器中针对特定真实世界区域完成,该模型能够以多边形网格形式写出,并且清除3D容器以便对新真实世界区域建模,该新真实世界区域可任选地随后被添加到多边形网格。在一些示例中,上述光线投射操作可以从多边形网格,而不是3D容器中执行。在其他示例中,模型视图可以从多边形网格中使用诸如标准图形呈现光栅化流水线等替换技术来生成。

[0084] 被存储为多边形网格的“高速缓存的”模型还可被转换成存储在容器上的3D模型。例如,所存储的、高速缓存的网格模型可作为“种子”被写回3D容器,可向该种子添加其他深度图像以进一步增强模型的准确性和细节。

[0085] 注意,在一些示例中,任何隐式表面表示都可能是可适用的(例如,其中确切的表面位置可通过数字方法和规则网格上的有限数量的存储点之间的内插来确定的隐式表面表示)。

[0086] 现在参考图8,它示出了可以实现为任何形式的计算和/或电子设备并且可由在其中实现上文所描述的3D环境建模的各实施例的示例性基于计算的设备800的各种组件。

[0087] 基于计算的设备800包括一个或多个处理器802,这些处理器可以是微处理器、控制器或用于处理计算可执行指令以控制设备操作以便执行3D重构的任何其他合适类型的处理器。在一些示例中,例如在使用片上系统架构的示例中,处理器802可以包括一个或多个固定功能块(亦称加速器),这些块以硬件(而非软件或固件)来实现建模方法的一部分。

[0088] 基于计算的设备800还包括经由通信接口806与处理器802进行通信的图形处理系统804,并包括被安排成以快速且高效的方式执行并行的、线程化的操作的一个或多个图

形处理单元 808。图形处理系统 804 还包括被安排成启用自图形处理单元 808 的快速并行访问的存储器设备 810。在各示例中,存储器设备 810 可存储 3D 容体,并且图形处理单元 808 可执行上述模型生成和光线投射操作。

[0089] 基于计算的设备 800 还包括被安排成接收来自诸如移动环境捕捉设备(包括深度相机)等一个或多个设备的输入以及可任选地接收来自一个或多个用户输入设备(例如,游戏控制器、鼠标、和/或键盘)的输入/输出接口 812。输入/输出接口 812 还可作为通信接口来操作,它被安排成与一个或多个通信网络(例如,因特网)进行通信。

[0090] 还提供了显示器接口 814 并且它被安排成向与该基于计算的设备集成在一起或与该基于计算的设备进行通信的显示系统提供输出。显示系统可提供图形用户界面,或任何合适类型的其他用户界面,但这不是必需的。

[0091] 可以使用可由基于计算的设备 800 访问的任何计算机可读介质来提供计算机可执行指令。计算机可读介质可以包括例如诸如存储器 816 等计算机存储介质和通信介质。诸如存储器 816 等计算机存储介质包括以用于存储如计算机可读指令、数据结构、程序模块或其他数据等信息的任何方法或技术实现的易失性和非易失性、可移动和不可移动介质。计算机存储介质包括但不限于, RAM、ROM、EPROM、EEPROM、闪存或其他存储器技术、CD-ROM、数字多功能盘(DVD)或其他光存储、磁带盒、磁带、磁盘存储或其他磁存储设备,或者可用于存储信息以供计算设备访问的任何其他非传输介质。相反,通信介质可以以诸如载波或其他传输机制等已调制数据信号来体现计算机可读指令、数据结构、程序模块或者其他数据。如本文所定义的,计算机存储介质不包括通信介质。虽然在基于计算的设备 800 中示出了计算机存储介质(存储器 816),然而应当理解,该存储可以是分布式的或位于远处并经由网络或其他通信链路(例如,使用输入/输出接口 812)来访问。

[0092] 可以在基于计算的设备处提供包括操作系统 818 或任何其他合适的平台软件的平台软件以使得能够在该设备上执行应用软件 820。存储器 816 可存储用于实现以下各项的功能的可执行指令:密集模型集成引擎 822(例如,被安排成使用参考图 4 描述的过程来在 3D 模型中构建模型)、密集模型可视化引擎 824(例如,被安排成使用图 7 的光线投射过程来输出模型的所呈现的图像)、以及密集模型查询引擎 826(被安排成从模型获取数据,例如用于构造多边形网格)。存储器 816 还可提供数据存储 830,数据存储 830 可被用来提供对由处理器 802 在执行诸如用于存储多边形网格等 3D 建模技术时使用的数据的存储。

[0093] 此处所使用的术语“计算机”是指带有处理能力使得它可以执行指令的任何设备。本领域的技术人员将认识到,这样的处理能力被集成到许多不同的设备中,因此,术语计算机包括 PC、服务器、移动电话、个人数字助理和许多其他设备。

[0094] 本文描述的方法可由有形存储介质上的机器可读形式的软件来执行,例如计算机程序的形式,该计算机程序包括在该程序在计算机上运行时适用于执行本文描述的任何方法的所有步骤的计算机程序代码装置并且其中该计算机程序可被包括在计算机可读介质上。有形(或非瞬态)存储介质的示例可包括盘、拇指型驱动器、存储器等,而不包括所传播的信号。软件可以适合于在并行处理器或串行处理器上执行,使得各方法步骤可以以任何适当的顺序实现,或同时实现。

[0095] 这确认了软件可以是有价值的、可单独交易的商品。它旨在包含运行于或者控制哑摄或标准硬件以实现所需功能的软件。它还旨在包含例如用于设计硅芯片,或者用于配

置通用可编程芯片的 HDL(硬件描述语言)软件等“描述”或者定义硬件配置以实现期望功能的软件。

[0096] 本领域的技术人员将认识到,用来存储程序指令的存储设备可以分布在网络上。例如,远程计算机可以存储被描述为软件的进程的示例。本地或终端计算机可以访问远程计算机并下载软件的一部分或全部以运行程序。可另选地,本地计算机可以根据需要下载软件的片段,或在本地终端上执行一些软件指令,并在远程计算机(或计算机网络)上执行另一些软件指令。本领域的技术人员还将认识到,通过利用本领域的技术人员已知的传统技术,软件指令的全部,或一部分可以通过诸如 DSP、可编程逻辑阵列等等之类的专用电路来实现。

[0097] 如本领域技术人员将清楚的,此处给出的任何范围或者设备值都可以被扩展或者改变而不失去所寻求的效果。

[0098] 可以理解,上文所描述的优点可以涉及一个实施例或可以涉及多个实施例。各实施例不限于解决所述问题中的任一个或全部的实施例或具有所述好处和优点中的任一个或全部的实施例。进一步可以理解,对“一个”项目的引用是指那些项目中的一个或多个。

[0099] 此处所描述的方法的步骤可以在适当的情况下以任何合适的顺序,或同时实现。另外,在不偏离此处所描述的主题的精神和范围的情况下,可以从任何一个方法中删除各单独的框。上文所描述的任何示例的各方面可以与所描述的其他示例中的任何示例的各方面相结合,以构成进一步的示例,而不会丢失寻求的效果。

[0100] 此处使用了术语“包括”旨在包括已标识的方法的框或元素,但是这样的框或元素不构成排它性的列表,方法或设备可以包含额外的框或元素。

[0101] 可以理解,上面对一较佳实施例的描述只是作为示例给出并且本领域的技术人员可以做出各种修改。以上说明、示例和数据提供了对本发明的各示例性实施例的结构和使用的全面描述。虽然上文以一定的详细度或参考一个或多个单个实施例描述了本发明的各实施例,但是,在不偏离本发明的精神或范围的情况下,本领域的技术人员可以对所公开的实施例作出很多更改。

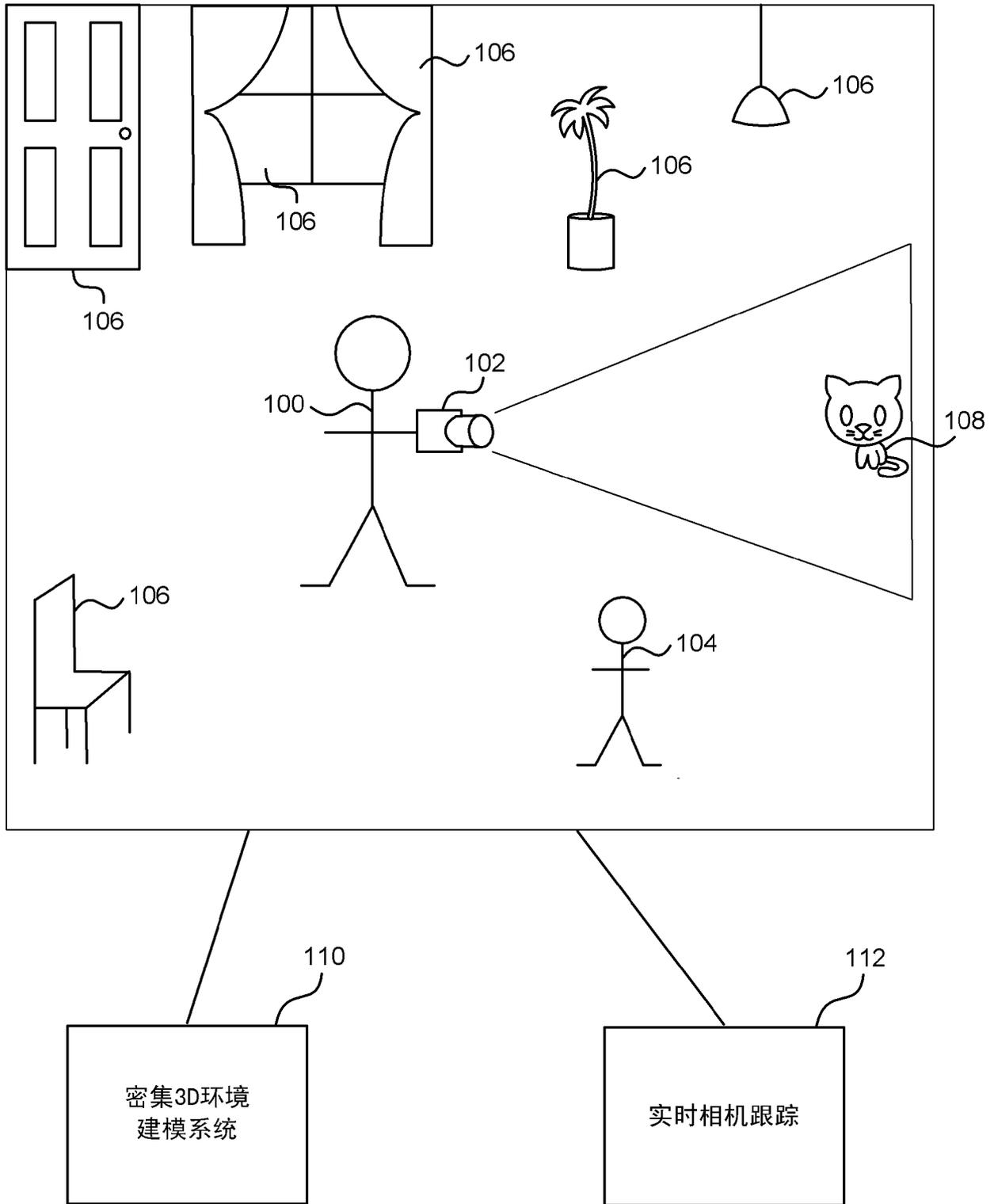


图 1

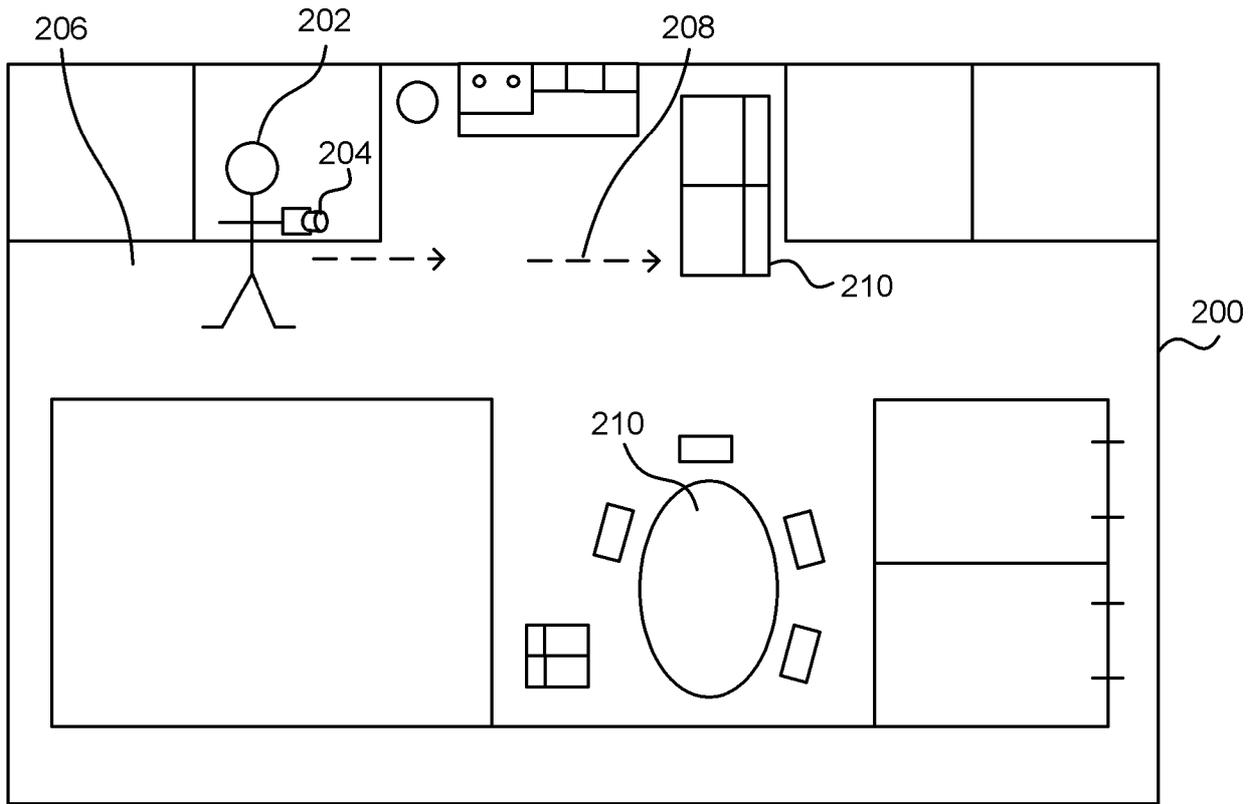


图 2

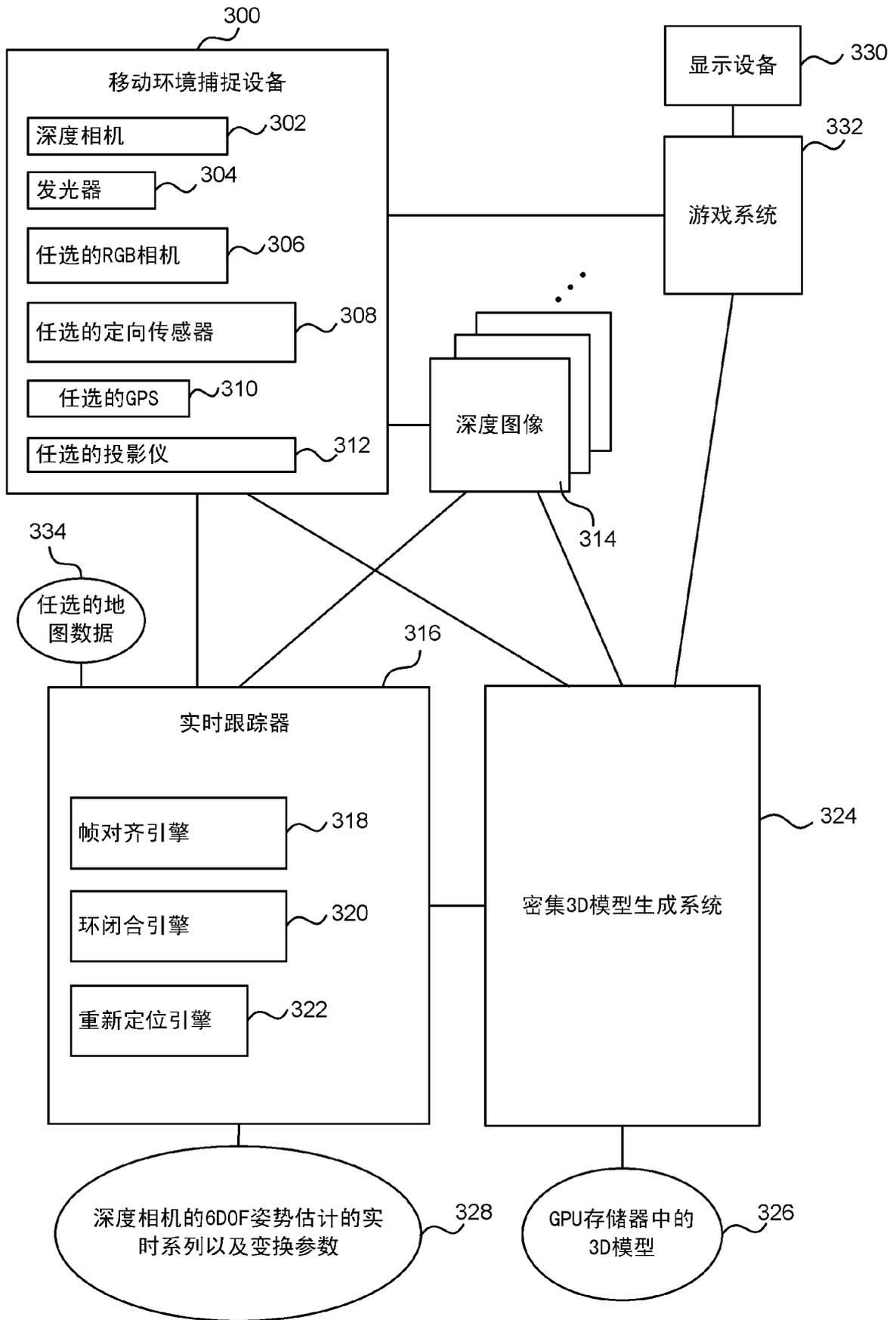


图 3

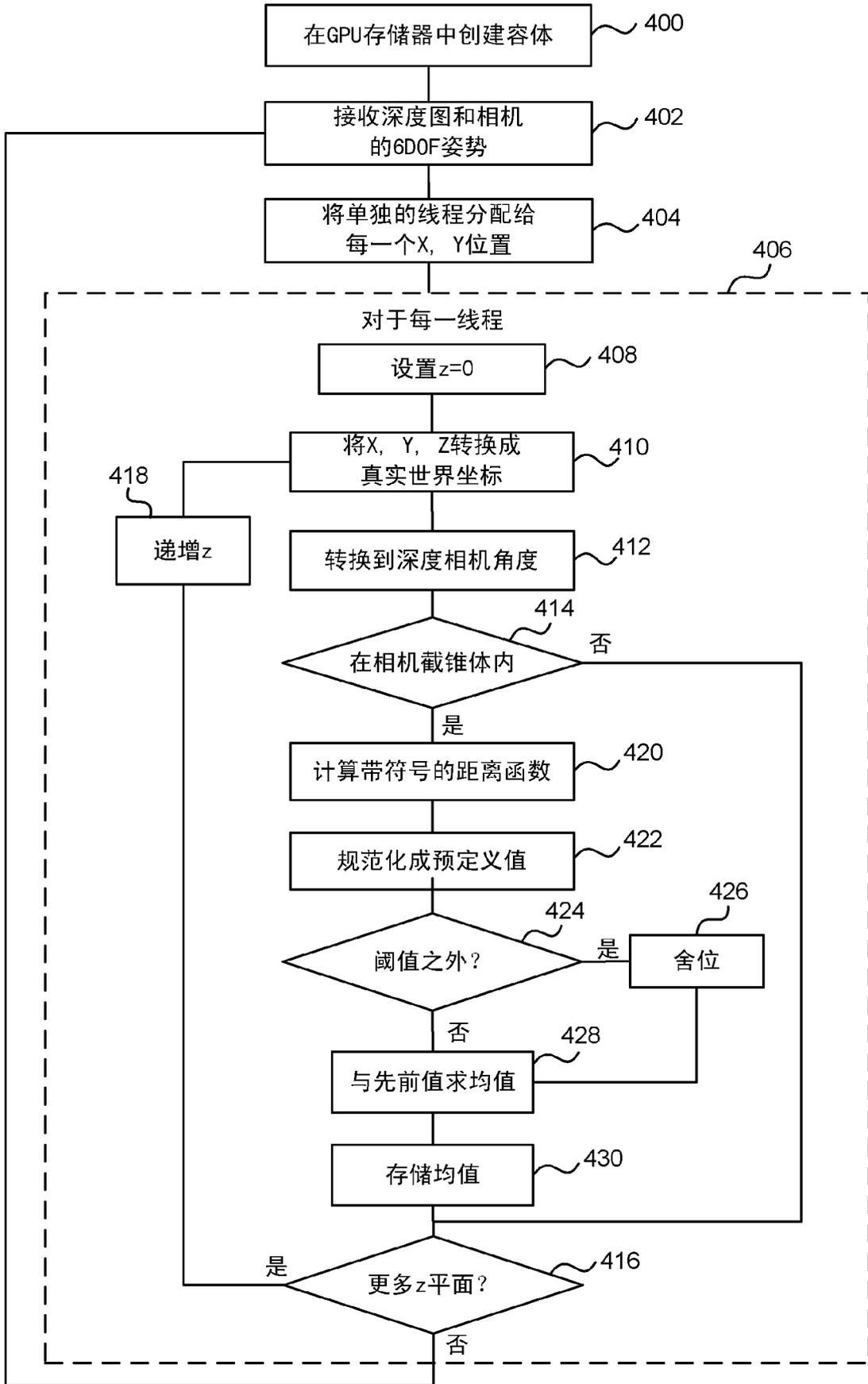


图 4

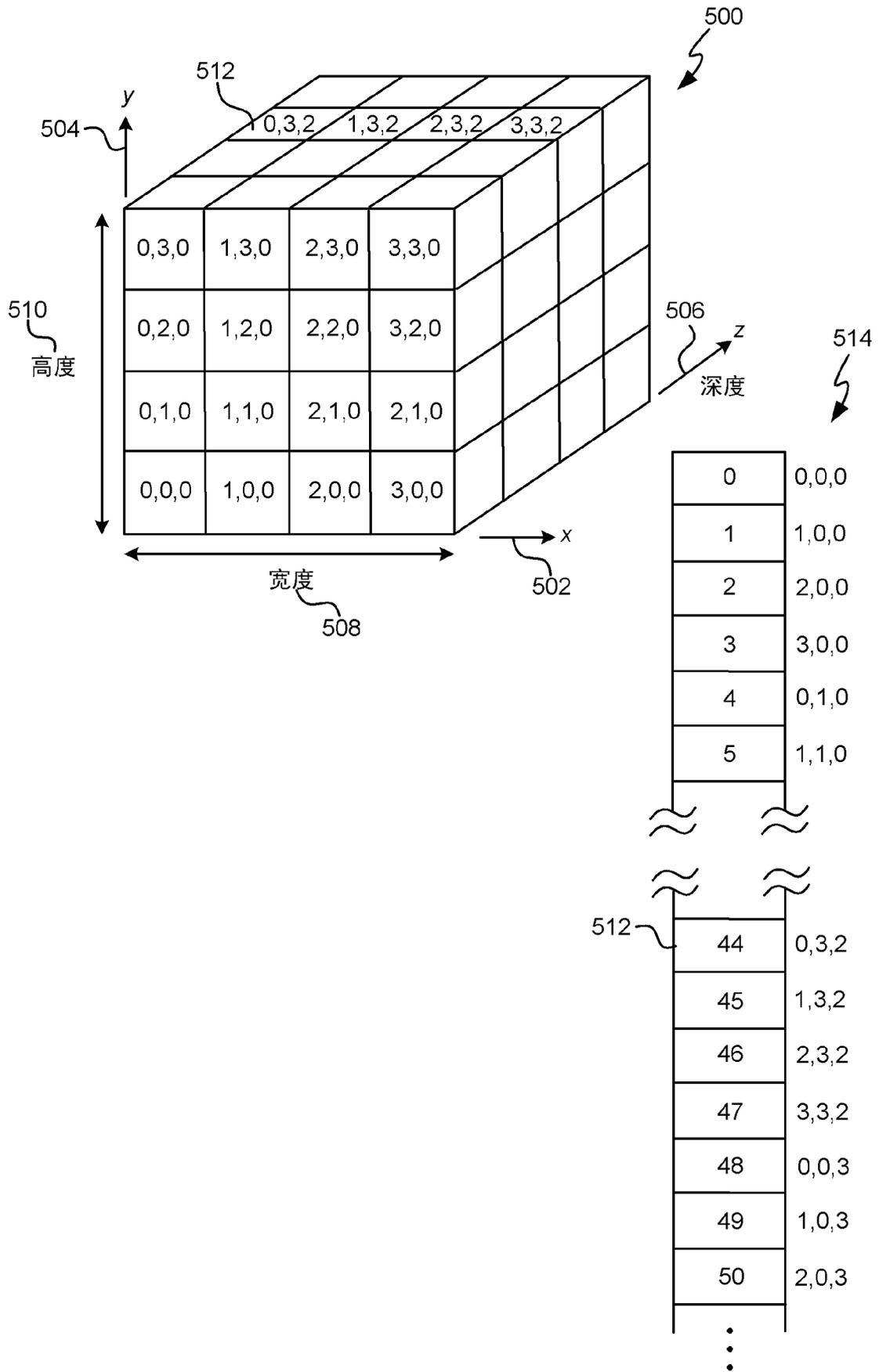


图 5

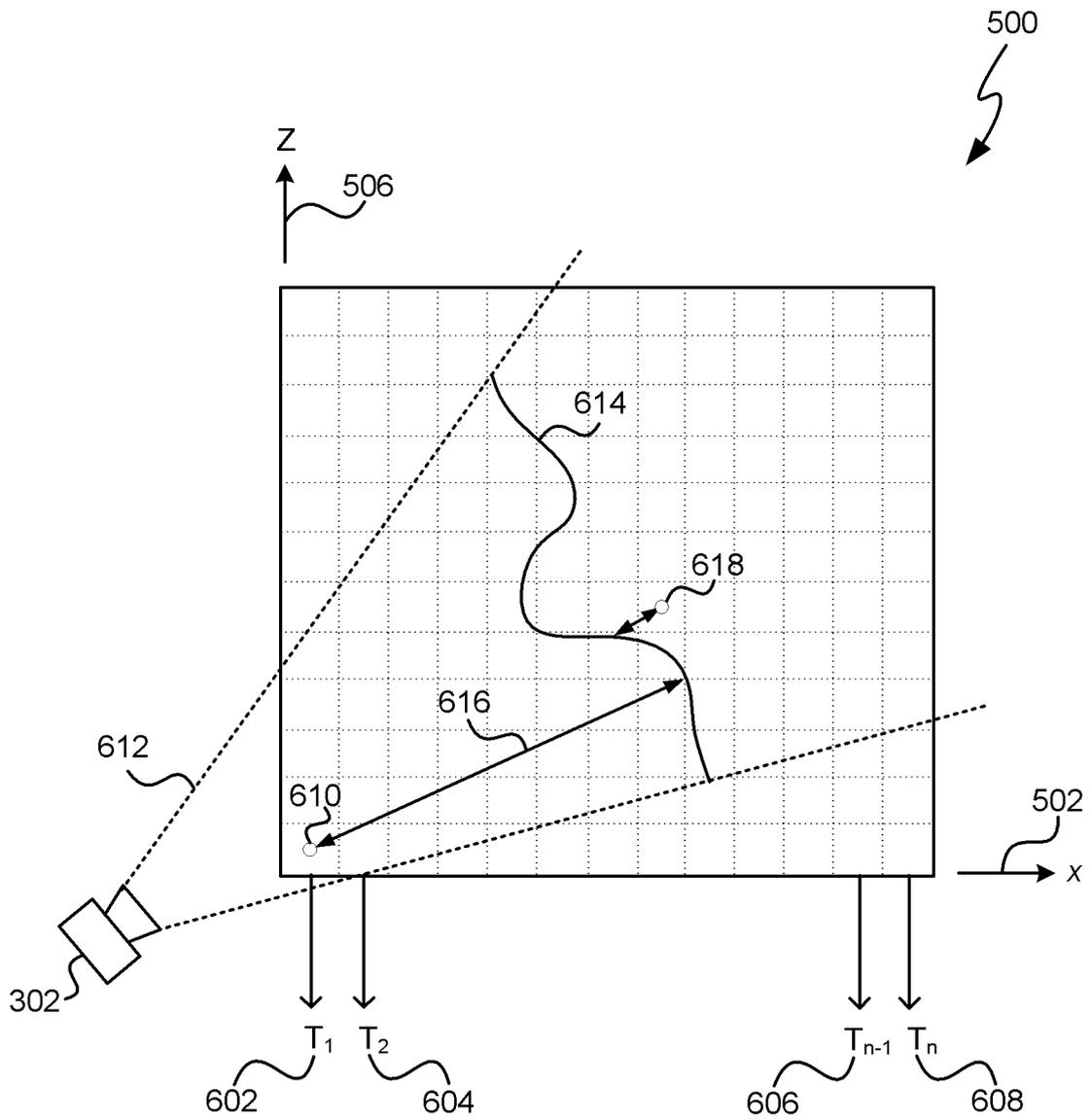


图 6

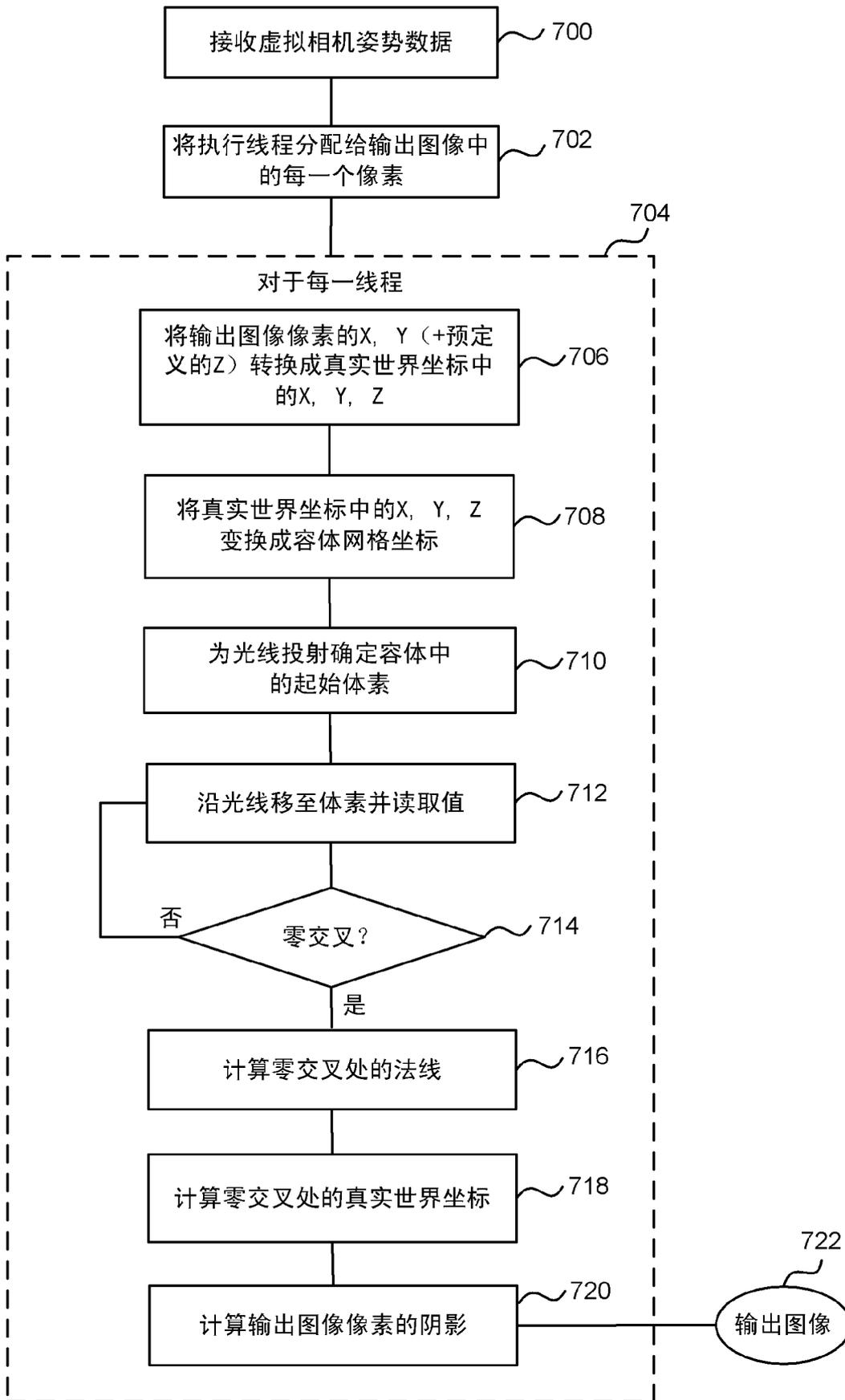


图 7

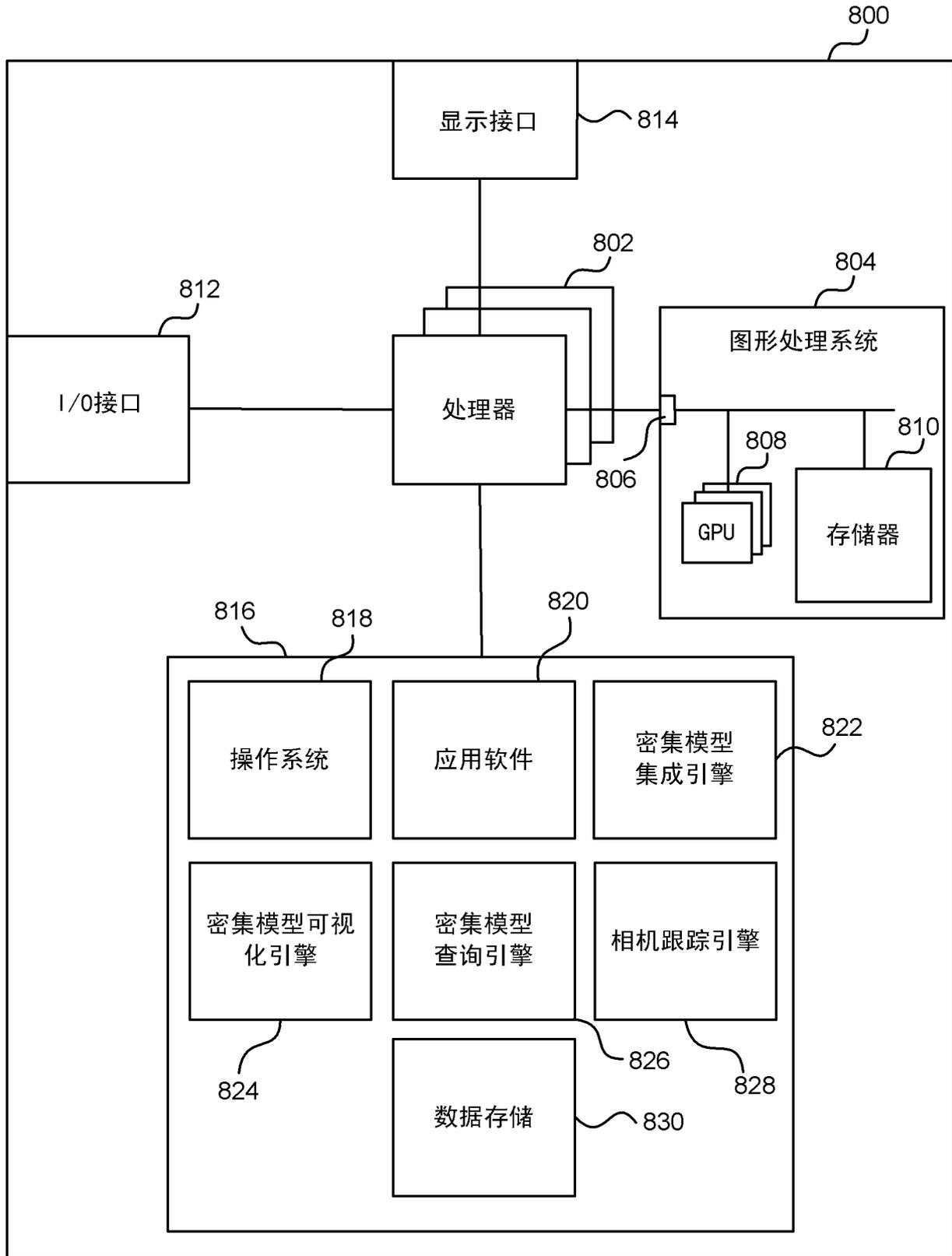


图 8