(19)国家知识产权局

(12)发明专利申请

(10)申请公布号 CN 115330919 A
(43)申请公布日 2022.11.11

(21)申请号 202210438526.2

(22)申请日 2022.04.25

(30)优先权数据
63/180,035 2021.04.26 US

(71)申请人 波音公司
地址 美国伊利诺伊州

(72)发明人 S·A·皮塔里宁 B·P·法姆

(74)专利代理机构 北京三友知识产权代理有限公司 11127
专利代理师 王青芝 党晓林

(51)Int.Cl.
G06T 15/00(2011.01)
G06F 3/04845(2022.01)
G06T 13/20(2011.01)

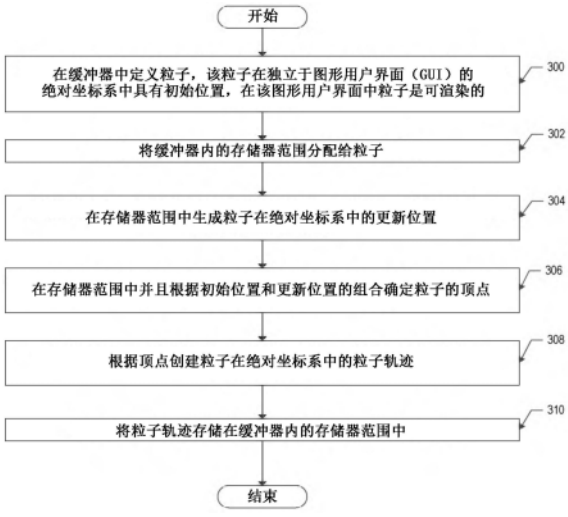
权利要求书1页 说明书20页 附图18页

(54)发明名称

用于动态显示的持久粒子轨迹的渲染

(57)摘要

本发明涉及用于动态显示的持久粒子轨迹的渲染。一种方法包括:在缓冲器中定义粒子,所述粒子在独立于图形用户界面的绝对坐标系中具有初始位置,在所述图形用户界面中所述粒子是可渲染的。该方法还包括将缓冲器内的存储器范围分配给相应的粒子。该方法还包括在存储器范围中生成粒子的更新位置。该方法还包括在存储器范围中根据初始位置和更新位置的组合确定每个粒子的相应顶点。该方法还包括根据相应的顶点创建粒子的相应粒子轨迹。该方法还包括将相应的粒子轨迹存储在与粒子相关联的存储器范围中。



CN 115330919 A

1.一种方法,所述方法包括以下步骤:

在缓冲器中定义多个粒子,所述多个粒子在独立于图形用户界面GUI的绝对坐标系中具有多个初始位置,在所述图形用户界面中所述多个粒子是可渲染的(300);

将所述缓冲器内的多个存储器范围分配给所述多个粒子中的相应粒子(302);

在所述多个存储器范围中生成所述多个粒子的多个更新位置(304);

在所述多个存储器范围中根据所述多个初始位置和所述多个更新位置的组合来确定所述多个粒子中的每一个的相应多个顶点(306);

根据所述相应多个顶点创建所述多个粒子的多个相应粒子轨迹(308);以及

将所述多个相应粒子轨迹存储在与所述多个粒子相关联的所述多个存储器范围中(310)。

2.根据权利要求1所述的方法,其中,所述多个存储范围中的每个存储范围被分配给所述多个粒子中的单个粒子并分配给所述多个相应粒子轨迹中的单个粒子轨迹。

3.根据权利要求1所述的方法,其中,创建所述多个相应粒子轨迹的步骤包括:在所述多个存储器范围上执行顶点着色器应用,使得对所述多个粒子中的每一个粒子执行一次所述顶点着色器应用(3F02)。

4.根据权利要求3所述的方法,所述方法还包括以下步骤:

将所述顶点着色器应用标记为无效着色器(3102)。

5.根据权利要求1所述的方法,其中,在第一渲染阶段(502)中所述相应多个顶点是劣化的且不可渲染的。

6.根据权利要求5所述的方法,其中,在所述第一渲染阶段之后的第二渲染阶段(504)中,所述相应多个顶点和多个附加顶点形成多个三角形作为所述多个相应粒子轨迹的可渲染表示。

7.根据权利要求5所述的方法,其中,在所述第一渲染阶段之后的第二渲染阶段(504)中,所述相应多个顶点和多个附加顶点形成多个三角形,所述多个三角形形成矩形四边形,作为所述多个相应粒子轨迹的可渲染表示。

8.根据权利要求5所述的方法,其中,在所述第一渲染阶段之后的第二渲染阶段(504)中,所述相应多个顶点和多个附加顶点形成多个多边形作为所述多个相应粒子轨迹的可渲染表示。

9.根据权利要求1所述的方法,所述方法还包括以下步骤:

在所述GUI上将所述多个粒子和所述多个相应粒子轨迹渲染为单个绘制调用(3B02)。

10.根据权利要求1所述的方法,其中,渲染包括:

将所述多个粒子的所述多个更新位置和所述相应多个粒子轨迹从所述绝对坐标系投影到所述GUI的屏幕坐标系,以形成第一投影(3C02);以及

在所述屏幕坐标系中在所述GUI上渲染所述多个粒子和所述相应多个粒子轨迹(3C04)。

用于动态显示的持久粒子轨迹的渲染

技术领域

[0001] 本发明涉及用于动态显示的持久粒子轨迹的渲染。

背景技术

[0002] 在图形用户界面(GUI)上渲染具有粒子轨迹的粒子在计算上是昂贵的。虽然可以使用屏幕空间缓冲器来存储粒子的位置及其轨迹,但是所存储的位置和轨迹仅对特定视点有效。当用户移动视点或移动缩放水平时,屏幕空间缓冲器被清除,并且为新视点或缩放水平重新计算粒子及其轨迹。结果是,在从新的视点或缩放水平渲染粒子和粒子轨迹之前,用户将察觉到GUI上的显示的清除,然后是没有显示任何内容的明显的暂停。

发明内容

[0003] 一个或更多个实施方式提供了一种方法。该方法包括在缓冲器中定义粒子,该粒子在独立于图形用户界面(GUI)的绝对坐标系中具有初始位置,在该图形用户界面中粒子是可渲染的。该方法还包括将缓冲器内的存储器范围分配给相应的粒子。该方法还包括在存储器范围中生成粒子的更新位置。该方法还包括在存储器范围中根据初始位置和更新位置的组合确定每个粒子的相应顶点。该方法还包括根据相应的顶点创建粒子的相应粒子轨迹。该方法还包括将相应粒子轨迹存储在与粒子相关联的存储器范围中。

[0004] 一个或更多个实施方式还提供了一种系统。该系统包括处理器和与该处理器通信的显示设备,该显示设备被配置为显示图形用户界面(GUI)。该系统还包括与处理器通信的存储器,该存储器具有缓冲器,该缓冲器具有存储范围。该系统还包括与处理器通信并存储计算机可读程序代码的非瞬态计算机可读存储介质,该计算机可读程序代码在由处理器执行时执行计算机实现的方法。该计算机实现的方法包括在缓冲器中定义具有独立于图形用户界面(GUI)的绝对坐标系中的初始位置的粒子,在该图形用户界面中粒子是可渲染的。计算机实现的方法还包括将缓冲器内的存储器范围分配给相应的粒子。计算机实现的方法还包括在存储器范围中生成粒子的更新位置。计算机实现的方法还包括在存储器范围中根据初始位置和更新位置的组合确定每个粒子的相应顶点。该计算机实现的方法还包括根据相应的顶点创建粒子的相应粒子轨迹。计算机实现的方法还包括将相应粒子轨迹存储在与粒子相关联的存储器范围中。

[0005] 该一个或更多个实施方式还包括存储计算机可读程序代码的非瞬态计算机可读存储介质,该计算机可读程序代码在由处理器执行时执行计算机实现的方法。该计算机实现的方法包括在缓冲器中定义粒子,该粒子在独立于图形用户界面(GUI)的绝对坐标系中具有初始位置,在该图形用户界面中粒子是可渲染的。计算机实现的方法还包括将缓冲器内的存储器范围分配给相应的粒子。计算机实现的方法还包括在存储器范围中生成粒子的更新位置。计算机实现的方法还包括在存储器范围中根据初始位置和更新位置的组合确定每个粒子的相应顶点。该计算机实现的方法还包括根据相应的顶点创建粒子的相应粒子轨迹。计算机实现的方法还包括将相应粒子轨迹存储在与粒子相关联的存储器范围中。

[0006] 根据以下描述和所附权利要求书，一个或一个以上实施方式的其它方面将显而易见。

附图说明

[0007] 图1示出了根据一个或更多个实施方式的显示设备，该显示设备显示围绕显示器移动的粒子，并且还示出了粒子的粒子轨迹。

[0008] 图2示出了根据一个或更多个实施方式的计算系统。

[0009] 图3A、图3B、图3C、图3D、图3E、图3F、图3G、图3H、图3I、图3J、图3K示出了根据一个或更多个实施方式的用于执行一个或更多个实施方式的算法的流程图。

[0010] 图4A、图4B、图4C、图4D、图4E、图4F、图4G、图4H、图4I和图4J示出了根据一个或更多个实施方式的用于执行一个或更多个实施方式的算法的附加流程图。

[0011] 图5示出了根据一个或更多个实施方式的用于渲染粒子和粒子轨迹的方法的流程图。

[0012] 图6、图7和图8示出了根据一个或更多个实施方式的与渲染粒子和粒子轨迹相关联的表。

[0013] 图9示出了根据一个或更多个实施方式的在位置更新之后关于粒子位置的顶点。

[0014] 图10和图11示出了根据一个或更多个实施方式的与渲染粒子和粒子轨迹相关联的附加表。

[0015] 图12示出根据一个或更多个实施方式的在附加更新之后关于粒子位置的可渲染的顶点。

[0016] 图13、图14和图15示出了根据一个或更多个实施方式的与渲染粒子和粒子轨迹相关联的附加表。

[0017] 图16示出了根据一个或更多个实施方式的用于在相对于图12的附加更新之后处理关于粒子位置的可渲染顶点的不同技术。

[0018] 图17A和图17B示出了根据一个或更多个实施方式的计算系统和网络环境。

具体实施方式

[0019] 现在将参考附图详细描述一个或更多个实施方式的特定实施方式。为了一致性，各图中相同的元件用相同的附图标记表示。

[0020] 在以下实施方式的详细描述中，阐述了多个具体细节以便提供对一个或更多个实施方式的更透彻理解。然而，对于本领域的普通技术人员显而易见的是，可以在没有这些具体细节的情况下实践一个或更多个实施方式。在其它情况下，没有详细描述公知的特征以避免不必要地使描述复杂化。

[0021] 在整个申请中，序数(例如，第一、第二、第三等)可用作元素(即，本申请中的任何名词)的形容词。除非明确公开，否则使用序数词并不暗示或产生元件的任何特定次序，也不将任何元件限制为仅单个元件，例如通过使用术语“之前”、“之后”、“单个”和其它此类术语。相反，序数的使用是为了区分元素。作为示例，第一元素不同于第二元素，并且第一元素可以包含多于一个元素，并且以元素的顺序在第二元素之后(或之前)。

[0022] 当关于可测量的物理性质使用时，术语“约”是指由本领域普通技术人员的工程师

或制造技术人员预期或确定的工程公差。工程公差的精确量化程度取决于所生产的产品和所测量的技术性质。对于非限制性示例，如果两个角度的值在彼此的百分之十内，则两个角度可以是“大约一致的”。然而，如果工程师确定特定产品的工程公差应该更严格，则“大约一致”可以是两个角度的值在彼此的百分之一内。同样，在其它实施方式中，工程公差可以放宽，使得“大约一致”的角度具有在彼此的百分之二十内的值。在任何情况下，普通技术人员能够评估什么是特定产品的可接受的工程公差，并且因此能够评估如何确定由术语“约”预期的测量的方差。

[0023] 通常，一个或更多个实施方式涉及提高在显示设备上渲染粒子及相关联的粒子轨迹的速度。粒子是小的渲染对象，并且粒子轨迹表示粒子的最近的先前位置。换句话说，粒子轨迹表示每个对象位置的简短历史，以便提供动态场的可视化。这种应用的一个例子是在地图上渲染风的显示。

[0024] 涉及粒子轨迹的粒子系统的渲染在动态视觉显示中是困难的。在显示器表示源空间(即，粒子位置的现实世界集合)中的移动粒子的情况下，即使当显示器视点相机移动时，粒子轨迹仍保持在其正确的源空间位置中，从而改变用于粒子轨迹的显示器的屏幕空间位置。

[0025] 关于渲染在显示屏幕上的粒子轨迹的信息被存储在表示固定视点的存储缓冲器中。因此，存在缓冲器中的值到屏幕上的渲染的直接映射。缓冲器包含投影到显示屏幕上的粒子位置历史，但是仅对当前观看位置有效。

[0026] 视点的任何改变(例如，缩放或平移显示器)导致所存储的粒子位置历史变得过时。当位置历史变得过时的时候，从屏幕上清除粒子和粒子轨迹。每次用户平移或缩放显示时，都执行清除显示屏幕。然后，渲染应用基于源数据生成一组新粒子，并在显示屏幕上渲染新粒子。每次视点被缩放或平移时，清除过时的粒子和渲染新粒子导致粒子显示的中断。

[0027] 一个或更多个实施方式通过提供用于产生和渲染粒子和粒子轨迹的技术来解决这些和其它技术问题。一个或更多个实施方式不使用屏幕空间缓冲器来存储投影粒子位置历史。相反，一个或更多个实施方式使用用于每个渲染粒子的小缓冲器来存储源坐标(例如，真实粒子的真实世界坐标)而不是屏幕空间坐标中的最近位置。然后将源数据坐标投影到屏幕空间中，并使用处理器(例如图形处理单元(GPU)，在所选视点改变时期望改变为使用投影，来将源数据渲染为粒子轨迹。因此，当用户缩放或平移具有粒子和粒子轨迹的图像时，所存储的粒子位置不会是无效的。

[0028] 换句话说，一个或更多个实施方式允许随着显示改变(即，随着用户平移或缩放显示)连续地渲染相同的粒子及其历史。如果平移或缩放导致在可见屏幕空间内单个粒子不再是可见的，则该粒子被清退(retire)，并且用于清退粒子的位置和历史的存储器然后被重新用于在当前可见区域内生成的新粒子。一个或更多个实施方式的过程允许在缩放和平移的同时更平滑地过渡和连续地渲染现有可见粒子，同时为任何新可见区域生成新粒子。

[0029] 因此，一个或更多个实施方式的过程在显示屏幕上呈现粒子和粒子轨迹的连续动态表示，同时避免显示屏幕上的图像的中断。因此，改善了用户体验。

[0030] 现在将注意力转向附图。图1示出了根据一个或更多个实施方式的显示设备，该显示设备显示围绕显示器移动的粒子，并且还示出了粒子的粒子轨迹。图1示出了根据关于图2至图16描述的原理和技术所渲染的粒子和粒子轨迹的性质的一个示例。

[0031] 图1示出了在显示设备(102)上生成的显示(100)。在显示设备(102)上生成的显示(100)示出了两个粒子及其相关联的粒子轨迹。特别地,将粒子(104)与相关联的粒子轨迹(106)一起显示。同样,粒子(108)与相关联的粒子轨迹(110)一起显示。

[0032] 诸如粒子(104)和粒子(108)的每个粒子表示对象。术语“对象”被定义为随时间被跟踪的物体。例如,“对象”可以是物理对象,例如但不限于灰尘粒子、地球上大气体积内的空气的部分(即,气团)、分子、原子、亚原子粒子等。在另一示例中,“对象”可以是正被跟踪的某些属性,例如但不限于正被跟踪的属性的多变量状态的当前值。一个或更多个实施方式考虑多个不同类型的对象。因此,术语“粒子”比通常与术语“粒子”相关的灰尘或其它小物理物体的粒子更宽泛地理解。

[0033] 如上所述,粒子可以表示被跟踪的某些属性的多变量状态的当前值。在更具体的示例中,考虑一种数值求解算法,该算法为表示模型的“N”个变量的一些选定集合找到最佳值。一个或更多个实施方式可以用于以图形方式描绘这种数值求解算法在潜在起始值范围内的行为。在该具体示例中,每个渲染的粒子的位置表示(首先)变量的初始开始值。这些值随时间的变化表示当试图找到解时调整变量值的解算法,例如变量关于条件集合的数学优化。在这种情况下,动画图形显示将表现为围绕显示器移动的随机间隔的一组粒子。粒子将倾向于会聚在求解空间中的一个或更多个点上。因此,在这种情况下,显示器将显示表示状态的变量的一个或更多个二维或三维图形。

[0034] 每个粒子轨迹表示粒子位置的定义的最近历史(即,具有预定时间段)。因此,粒子轨迹(106)示出了粒子(104)的最近位置的历史。同样,粒子轨迹(110)示出了粒子(108)的最近位置的历史。

[0035] 元数据还可以与粒子相关联或用粒子来识别。例如,元数据可以表示粒子运动的强度或速率或其它信息。可以通过改变粒子轨迹的大小、形状、颜色或散列来渲染元数据以供显示。在特定示例中,可以将气团(即,在该示例中为“粒子”)的风速和风向(即,速度)表示为粒子的元数据。风速可以与粒子和粒子轨迹一起渲染在显示器中。

[0036] 一个或更多个突出显示方案可用于粒子轨迹、或粒子或其组合。例如,可以使粒子轨迹是动画的,或者可以用不同的阴影来表示元数据等。粒子也可以用一个或更多个预定形状来表示,在本例中可以认为是突出显示。因此,如图1所示,粒子轨迹(106)具有与粒子轨迹(110)不同的散列样式。不同的散列样式表示不同的突出显示类型,但是如上所述,可以设想多个不同的突出显示样式和类型。

[0037] 在实际的例子中,显示器(100)可以显示成千上万个粒子及其粒子轨迹。因此,例如,对于大区域上(例如,在整个地球全球上)的多个气团可以显示风速和风速的最近历史。然而,一个或更多个实施方式具有多个其它应用,并且不必限于示出气团(作为粒子)和气团的历史位置(作为粒子轨迹)。

[0038] 如上所述,一个或更多个实施方式提供了一种用于改善粒子和粒子轨迹的渲染的技术。因此,即使在诸如以上给出的实际示例中,一个或更多个实施方式也可以使显示设备显示从一个视点或缩放水平到另一个视点或缩放水平的粒子和粒子轨迹的平滑过渡。

[0039] 图2示出了根据一个或更多个实施方式的计算系统。图2的计算系统可使用关于图17A和图17B所示的计算系统和网络环境的一个或更多个组件来实现。

[0040] 图2所示的系统包括计算机(200)。计算机(200)具有处理器(202)。处理器(202)是

由单独的可能是远程硬件处理单元支持的硬件处理单元或逻辑处理单元。

[0041] 处理器(202)与显示设备(204)通信。显示设备(204)可以是计算机(200)的一部分,但也可以是从计算机(200)接收图像或视频的远程显示设备。处理器(202)还与存储器(206)通信,如下所述。

[0042] 显示设备(204)被配置为生成显示(210)。显示(210)是由显示设备(204)生成或投影到显示设备(204)上的图像或视频。显示(210)还被配置为显示图形用户界面(GUI(212))。在一些情况下,GUI(212)可以构成显示(210)。GUI(212)可以包括一个或更多个窗口小控件。窗口小控件是按钮、滑杆、下拉菜单等,用户可以使用用户输入设备来操纵该窗口小控件以便与由计算机(200)执行的应用交互。

[0043] 存储器(206)是计算机(200)用作处理器(202)执行指令的一部分的瞬态(例如随机存取存储器)或非瞬态(例如持久存储介质)。在一个或更多个实施方式中,存储器(206)包括具有至少一个存储器范围(例如,但不限于存储器范围(216)和存储器范围(218))的缓冲器(214)。

[0044] 缓冲器(214)是物理存储器(在这种情况下是存储器(206),但也可能在下面进一步描述的图形处理单元(208)中)的、用于在计算机(200)中将数据从一个地方移动到另一个地方时临时存储数据的区域。缓冲器(214)可以被再划分为存储器范围。存储器范围是缓冲器(214)的预定部分。例如,如果存储器以比特数表示,则存储器范围可以是预定的比特数。在一个或更多个实施方式中,存储器范围(216)和存储器范围(218)是缓冲器(214)中的存储器范围。

[0045] 缓冲器(214)可以是各种不同的类型。例如,缓冲器(214)可以是环形缓冲器。环形缓冲器是被视为圆形的数据结构,尽管环形缓冲器的实现可以是线性的。当新数据被引入环形缓冲器时,数据被重写在缓冲器的头部。其它类型的缓冲器包括线性缓冲器、帧缓冲器、可变长度缓冲器等。

[0046] 图形处理单元(208)是一种处理器。然而,多个计算机具有一个以上的处理器,例如,中央处理单元(“CPU”)(在此示例中为处理器(202))和图形处理单元(208)。图形处理单元(208)是被设计并专用于在包括显示(210)和GUI(212)的显示设备(204)上处理和渲染图形的硬件。

[0047] 图形处理单元(208)和处理器(202)在多个应用中一起工作。因此,一个或更多个实施方式可在处理器(202)、图形处理单元(208)或其组合上执行。

[0048] 图1所示的系统还包括与处理器(202)通信的非瞬态计算机可读存储介质(220)。非瞬态计算机可读存储介质(220)是持久或非瞬态存储器,诸如硬盘驱动器、闪存驱动器等。非瞬态计算机可读存储介质(220)存储各种数据。

[0049] 例如,非瞬态计算机可读存储介质(220)存储计算机可读程序代码(222)。计算机可读程序代码(222)在由处理器执行时执行计算机实现的诸如关于图3A至图5描述的方法。

[0050] 非瞬态计算机可读存储介质(220)还存储附加信息。例如,非瞬态计算机可读存储介质(220)存储绝对坐标系(224)。绝对坐标系(224)是这样的坐标系:独立于粒子如何被示出在显示(210)或GUI(212)上,粒子的位置可以参考该坐标系。绝对坐标系(224)的示例可以是真实世界坐标系(例如,当使用纬度和经度将空气中的一部分参考为地球上的位置时)或制造场地的预定坐标系。绝对坐标系(224)的多个不同示例是可能的。

[0051] 相反，非瞬时计算机可读存储介质(220)还存储屏幕坐标系(226)。屏幕坐标系(226)是特定于显示(210)和/或显示在显示设备(204)上的GUI(212)的坐标系。因此，例如，屏幕坐标系(226)参考屏幕上的粒子的位置。当用户改变显示设备(204)的缩放水平或平移显示(210)或GUI(212)时，则每个粒子的屏幕坐标改变。因此，虽然屏幕坐标系(226)本身不一定改变，但是当用户缩放或平移显示(210)或GUI(212)时，屏幕坐标系(226)内的给定粒子的位置将改变。

[0052] 如参照图3A至图5所描述的，一个或更多个实施方式提供了生成投影(228)的技术。投影(228)是将绝对坐标系(224)变换为屏幕坐标系(226)的数学变换。计算机(200)跟踪粒子在绝对坐标系(224)中的位置，但是当用户缩放或平移显示(210)或GUI(212)时，计算机(200)改变投影(228)，使得粒子在屏幕坐标系(226)中的相对位置被适当地显示。再次参考图3A至图5描述用于产生投影(228)的过程的细节。

[0053] 非瞬态计算机可读存储介质(220)还存储顶点着色器应用(230)。顶点着色器应用(230)是图形处理单元(208)中的可编程功能或用于将顶点(例如三角形或其它多边形的点)的属性(例如颜色，纹理，位置和方向)从原始空间变换到显示装置(204)上的显示空间的独立应用。下面进一步描述关于绝对坐标系(224)中的粒子的顶点。

[0054] 顶点着色器应用(230)允许原始对象按期望的那样移动、扭曲或重新成形。换句话说，顶点着色器应用(230)可处理顶点并确定顶点的坐标在“剪辑空间”(也称为“视锥(view frustum)”)中是什么，所述“剪辑空间”是使计算机(200)易于理解哪些顶点对相机可见(用户的视点)而哪些不可见的空间。照相机(或视锥)不可见的顶点被切割或“剪除”。以此方式，图形处理单元(208)可在稍后的渲染阶段期间执行得更快，这是因为图形处理单元(208)具有较少的数据来工作。顶点着色器应用(230)通过接收来自顶点列表的顶点作为输入来操作，并返回确定顶点应存在于剪辑空间内何处的结果。

[0055] 顶点着色器应用(230)可以在在缓冲器(214)中的若干存储器范围(即，存储器范围(216)和存储器范围(218))上单独地执行。因此，如下文相对于图3A到图4J进一步解释的，通过将一个粒子和粒子轨迹指派给每个存储器范围，顶点着色器应用(230)可用于单独地处理每一个单个粒子(即，与每一个单个粒子相关联的顶点)。

[0056] 非瞬时计算机可读存储介质(220)还存储关于粒子的信息。然而，可以在处理期间将粒子的信息加载到存储器(206)中以便在显示设备(204)上显示。因此，对存储在非瞬态计算机可读存储介质(220)中的关于粒子或粒子轨迹的数据的任何参考也可以指存储在存储器(206)和/或图形处理单元(208)中的专用存储器中的关于粒子或粒子轨迹的数据。

[0057] 如上所述，粒子和粒子轨迹相对于绝对坐标系(224)被存储，并且因此在图2中被示出为在绝对坐标系(224)内。然而，期望保存定义粒子的数据的数据结构是关系数据库、树数据库、图形数据库或一些其它数据库的形式。在数据结构中，粒子由与关于粒子的信息(例如，粒子位置和可能的其它粒子特性，例如但不限于粒子速度、粒子类型等)相关联的标识符指定。

[0058] 因此，如图2所示，粒子可以包括粒子A(232)和粒子B(234)。可以存在更多或更少的粒子。粒子A(232)和粒子B(234)都是如上关于图1所定义的“粒子”。如上所述，“粒子”可以是地球上大气体积内的空气的一部分(即，气团)，可用于跟踪空气(即，风)的运动。然而，如上所述，“粒子”可以是人、动物或在物理世界中移动的对象、虚拟空间中的计算机生成的

对象、多变量状态、或表示要在绝对坐标系(224)中跟踪并最终要投影到屏幕坐标系(226)上的任何其它事物。

[0059] 粒子由标识符标识。在图2的例子中，标识符是标签“A”(对于粒子A(232))和标签“B”(对于粒子B(234))。可以使用其它标识符。

[0060] 粒子与绝对坐标系(224)中的位置相关联。因此，例如，在给定时间“T”，粒子A(232)位于绝对坐标系(224)中的位置A(236)。类似地，在给定时间“T”，粒子B(234)位于绝对坐标系(224)中的位置B(238)。

[0061] 在大多数情况下，粒子与粒子轨迹相关联。如上所述，粒子轨迹被定义为相应粒子的位置历史。然而，粒子轨迹还可以包括其他信息，例如粒子速度、粒子特性等。

[0062] 因此，例如，粒子A(232)具有相应的粒子轨迹A(240)。粒子轨迹A(240)是粒子A(232)在不同时间的过去位置的历史。同样，粒子B(234)具有相应的粒子轨迹B(242)。粒子轨迹B(242)是不同时间粒子B(234)的过去部分的历史。

[0063] 每个粒子与相应的顶点或一组顶点相关联。如本文中所使用的，顶点被定义为在绝对坐标系(224)或屏幕坐标系(226)中形成待渲染的粒子(例如，粒子A(232)或粒子B(234))的定义的一部分的一条或多条线的起点或终点。与粒子相关联的一组顶点可以定义粒子和/或粒子轨迹的形状。在一个示例中，视频游戏可以使用由多个顶点和表示几何形状的顶点之间的线定义的多个几何形状来渲染场景。(纹理可以用于填充线之间的细节，尽管这里没有描述纹理)。

[0064] 在一个或更多个实施方式中，可以使用各种顶点来定义关于粒子和粒子轨迹的各种形状和其他信息。顶点可以被分组以定义各种形状的多边形，例如但不限于三角形、矩形对象和更高阶的多边形(五边形，六边形，八边形等)。使用形成三角形的多个顶点的示例在图5至图12中示出，并且使用形成矩形对象的多个顶点的另一示例在图13至图16中示出。

[0065] 顶点着色器应用(230)可用于将一组顶点渲染为显示在显示装置(204)上的形状。因此，顶点着色器应用(230)可使用顶点A1(244)、顶点A2(246)和顶点A3(248)来渲染粒子A(232)，或可能渲染粒子A(232)的过去版本、粒子A(232)的任何历史位置(即，粒子轨迹A(240))。类似地，顶点着色器应用(230)可使用顶点B1(250)、顶点B2(252)和顶点B3(254)来渲染粒子B(234)，或者可能渲染粒子B(234)的任何历史位置(即粒子轨迹B(242))。

[0066] 虽然图2示出了组件的配置，但是在不脱离一个或更多个实施方式的范围的情况下可以使用其他配置。例如，可以组合各种组件以创建单个组件。作为另一示例，由单个组件执行的功能可由两个或更多个组件执行。

[0067] 图3A至图5是根据一个或更多个实施方式的流程图。图3A至图5的方法可由图2的系统执行和/或图17A和图17B的计算系统中所示的组件支持。如关于图3A至图4J所使用的，术语“GUI”涵盖图2中的显示(210)和GUI(212)两者。

[0068] 首先参考图3A。图3A的方法的特征在于创建粒子的粒子轨迹的方法。

[0069] 步骤300包括在缓冲器中定义粒子，该粒子在独立于图形用户界面(GUI)的绝对坐标系中具有初始位置，在该图形用户界面中粒子是可渲染的。通过定义粒子的特性来定义粒子。粒子的特性至少由粒子在绝对坐标系中的初始位置来定义。在适当的情况下，其他特性可以包括粒子的尺寸、粒子的质量、粒子特性等。通过将粒子特性加载到缓冲器中的存储器范围中来在缓冲器中定义粒子。

[0070] 如图2所示，绝对坐标系独立于屏幕坐标系。虽然在某些情况下这两个坐标系可以是相同的，但是在多种情况下这两个坐标系将是不同的，因为用户在使用显示设备期间将平移、缩放或以其他方式改变GUI的视图。

[0071] 步骤302包括将缓冲器内的存储器范围分配给粒子。通过将至少包括粒子的初始位置的粒子的至少一些特性加载到缓冲器内的指定存储器范围中来分配存储器范围。在一个实施方式中，粒子被约束到存储器范围，除非稍后的处理需求要求粒子在新的存储器范围中被移动然后被跟踪。一般而言，在缓冲器内的粒子和存储器范围之间存在一对一的对应关系。

[0072] 步骤304包括在存储器范围中生成粒子在绝对坐标系中的更新位置。如上所述，在存储范围内跟踪粒子在绝对坐标系中的变化位置。在图6到图16中示出了渲染粒子在存储缓冲器中跟踪粒子位置的示例。然而，通常，通过在缓冲器的存储器范围内添加粒子的新位置的新坐标来产生更新位置。

[0073] 步骤306包括在存储器范围中根据初始位置和更新位置的组合确定粒子的顶点。通过图6至图16中的示例来描述粒子的顶点的生成。简而言之，从粒子的当前和历史位置(即，将成为粒子轨迹的部分)生成顶点。在位置历史中直到预定数量的位置的每个位置代表一个顶点。

[0074] 步骤308包括根据顶点创建粒子在绝对坐标系中的粒子轨迹。一组顶点(即来自步骤306的预定数目的顶点)形成粒子轨迹。同样，粒子轨迹由粒子随时间的历史位置形成。然后，可以将粒子轨迹中的顶点布置成多边形，以便稍后由顶点着色器渲染。

[0075] 步骤310包括将粒子轨迹存储在缓冲器内的存储器范围中。因为粒子踪迹是粒子的过去位置的历史记录，所以存储粒子踪迹是通过将在步骤304确定的更新位置(和/或可能的粒子的任何其它过去位置)存储在缓冲器内的存储器范围内来完成的。

[0076] 存储粒子轨迹供以后使用。粒子轨迹的一个用途是通过渲染由顶点定义的多边形来渲染粒子轨迹，如图3B所示。粒子轨迹的另一用途是将粒子轨迹提供给以粒子的历史位置作为输入的另一应用。

[0077] 现在参考图3B。图3B的方法可以在图3A的方法之后执行。

[0078] 步骤3B02包括在GUI上将粒子和粒子踪迹渲染为单个绘制调用(draw call)。如上所述，顶点着色器可用于渲染由粒子轨迹中定义的顶点形成的多边形。在一个实施方式中，如图6至图16所示，可以使用单个绘制调用来渲染粒子缓冲器中的所有粒子。因此，渲染粒子的计算效率相对于试图单独渲染每个粒子而增加。

[0079] 现在将注意力转向图3C的方法。图3C的方法可以在图3B的方法之后执行。

[0080] 步骤3C02包括将粒子的更新位置和粒子轨迹从绝对坐标系投影到GUI的屏幕坐标系，以形成第一投影。当要在GUI上显示粒子时，在一些(如果不是大多数)情况下，在绝对坐标系到屏幕坐标系之间执行变换(即，投影)。以这种方式，在显示设备上观看粒子的比例和位置可以不同于从绝对(即不变的)视点观看相同粒子。

[0081] 步骤3C04包括使用第一投影在屏幕坐标系中在GUI上渲染粒子和粒子轨迹。在一个示例中，渲染粒子和粒子轨迹，并且渲染的粒子和粒子轨迹被投影变换并显示在显示设备上。在另一示例中，通过投影来变换粒子和粒子轨迹，然后在GUI上渲染变换后的粒子和粒子轨迹。

[0082] 现在转向图3D。图3D的方法在图3C的方法之后。

[0083] 步骤3D02包括接收改变第一投影的指令。该指令可以由用户经由用户输入设备(显示屏幕、键盘、鼠标等)提供摇摄、变焦、转动或以其他方式调整GUI上显示的图像的命令来接收。

[0084] 该指令可以是命令的形式,否则该命令影响用于将绝对坐标变换为屏幕坐标的计算。因此,例如,用户或另一自动化过程用户可以输入调整数学变换的命令,该数学变换实现从绝对坐标到屏幕坐标的投影。

[0085] 步骤3D04包括将粒子的更新位置和粒子轨迹从绝对坐标系投影到屏幕坐标系,以形成第二投影。可以如上关于图3C中的步骤3C04所述的那样执行投影。

[0086] 步骤3D06包括使用第二投影在屏幕坐标系中在GUI上重新渲染粒子和粒子轨迹。同样,可以如上关于图3C中的步骤3C04所述来执行渲染。

[0087] 在一个实施方式中,步骤3D02至3D06可以形成在步骤3D06之后终止的独立方法。然而,该方法可以继续进一步的步骤。

[0088] 因此,步骤3D08至3D12可以在图3B中的步骤3B02之后执行。然而,步骤3D08至3D12也可以在步骤3D02至3D06之后执行。

[0089] 步骤3D08包括确定在屏幕坐标系中粒子不再是可见的。在屏幕坐标系中粒子可能不再是可见的,这是因为用户已经通过平移或缩放选择了其中粒子在屏幕坐标系中不再可见的视图位置。

[0090] 步骤3D10包括从缓冲器中对粒子进行清退以形成清退粒子。通过清除或重写正在跟踪清退粒子的存储器范围来执行对粒子进行清退。当计算机仍然跟踪粒子在缓冲器外部的绝对坐标系中的位置时,通过从缓冲器对粒子进行清退来节省计算资源。

[0091] 步骤3D12包括通过产生新粒子代替清退粒子来维持缓冲器的大小。如本文所用,术语“产生”意指“生成”或“创建”。在一些情况下,从节省计算资源的观点来看,维持缓冲器的大小(即,维持预定数目的存储器范围)可能是有利的。因此,为了在清退粒子时保持缓冲器的尺寸,可以产生新粒子来代替清退粒子。然后在先前由清退粒子占用的存储器范围中跟踪新粒子。在这种情况下,新粒子数据重写缓冲器的存储器范围中的清退粒子数据。

[0092] 现在将注意力转向图3E的方法。图3E的方法可以在图3D的方法之后执行。

[0093] 步骤3E02包括在缓冲器的存储器范围中定义新粒子,新粒子在屏幕坐标系中是可见的并且在绝对坐标系中具有初始的第二位置。换句话说,在用户摇摄、缩放或以其他方式改变显示器的视点之后,新粒子在屏幕坐标系中是可见的。新粒子在绝对坐标系中还是移动的。

[0094] 步骤3E04包括在存储器范围中产生新粒子在绝对坐标系中的更新的第二位置。为了说明粒子的移动,粒子的更新的第二位置被记录在该粒子的存储器范围中。新粒子的先前位置(如果可用)被保留在存储器范围中,用于生成新粒子轨迹。

[0095] 步骤3E06包括在存储器范围中根据初始第二位置和更新的第二位置的组合确定新粒子的第二顶点。如上所述,从粒子的历史位置(即粒子轨迹)生成顶点。参见图3A的步骤306。

[0096] 步骤3E08包括根据第二顶点创建新粒子在绝对坐标系中的第二粒子轨迹。可以使用与关于图3A的步骤308所描述的技术类似的技术来执行第二粒子踪迹的创建。

[0097] 步骤3E10包括将第二粒子轨迹存储在缓冲器内的存储器范围中。存储第二粒子轨迹类似于图3A中的步骤310。

[0098] 现在参考图3F。图3F的方法可以作为图3A的方法的步骤304的一部分来执行。

[0099] 步骤3F02包括对存储器范围中的粒子执行顶点着色器应用。参考图2的顶点着色器应用(230)来描述顶点着色器应用。使用所述处理器或所述GPU执行所述顶点着色器应用。顶点着色器应用将图3A的方法和/或图3E的方法中产生的顶点作为输入,且产生可渲染的形状作为输出。参考图6到图16描述所使用的顶点着色器应用的示例。

[0100] 现在来看图3G。图3G的方法可以作为图3A的方法的步骤304的一部分来执行,并且还可以在图3F的步骤3F02之后执行。

[0101] 步骤3G02包括向粒子分配标识符。为粒子分配标识符(例如,数字或名称或其它字母数字串)。使用指针将标识符与其它粒子特性相关联,所述指针包括数据结构中的标识符,该数据结构还包括描述特性的数据,将标识符放置在这种数据结构的报头中,或通过一些其它技术将数据关联在一起。

[0102] 步骤3G04包括利用顶点着色器应用使用标识符跟踪粒子。顶点着色器应用通过从与粒子标识符相关联的对应存储器范围接收位置数据作为输入来使用标识符跟踪粒子。以这种方式,可以随时间跟踪粒子的正在进行的位置。

[0103] 现在参考图3H。图3H的方法可以作为图3A的方法的步骤304的一部分来执行,并且还可以在图3F的步骤3F02之后执行。

[0104] 步骤3H02包括在缓冲器的存储器范围内跟踪粒子的元数据。元数据可以包括时间戳、当前粒子年龄、最大粒子年龄、粒子特性等。在被跟踪的每个时间点,粒子的位置和元数据被记录在缓冲器的存储器范围内。

[0105] 步骤3H04包括将元数据作为输入应用于顶点着色器应用。在一些实施方式中,顶点着色器应用可使用元数据来细化或改变形成输出的形状的输出。在图6至图16中提供了示例。

[0106] 现在参考图3I。图3I的方法可以在图3A的方法的步骤306之后执行。步骤3I02包括将顶点着色器应用标记为无效着色器(void shader)。将顶点着色器应用标记为无效着色器意味着来自顶点着色器的所得几何结构将不会通过片段程序运行。结果是节省了处理资源,从而提高了速度性能。换句话说,无效着色器仅影响输入缓冲器且不具有如常规顶点着色器将具有的专用输出。

[0107] 现在转到图3J。图3J的方法可以在图3A的方法之后执行。图3J示出了可以针对第二粒子重复图3A的方法。然而,可使用(在一些实施方式中)单个顶点着色器应用同时执行图3J的方法和图3A的方法,从而允许多个粒子和粒子轨迹的平滑产生。

[0108] 步骤3J02包括在缓冲器的与存储范围不同的第二存储范围中定义第二粒子,第二粒子在绝对坐标系中具有初始第二位置。定义第二粒子类似于图3A中的步骤300。

[0109] 步骤3J04包括在第二存储器范围中生成第二粒子在绝对坐标系中的更新的第二位置。生成更新的第二位置类似于图3A中的步骤304。

[0110] 步骤3J06包括在第二存储器范围中根据初始第二位置和更新的第二位置的组合确定第二粒子的第二顶点。确定第二顶点类似于图3A中的步骤306。

[0111] 步骤3J08包括根据第二顶点创建第二粒子在绝对坐标系中的第二粒子轨迹。创建

第二粒子轨迹类似于图3A中的步骤308。

[0112] 步骤3J10包括将第二粒子轨迹存储在缓冲器内的第二存储器范围中。存储第二粒子轨迹类似于图3A中的步骤310。

[0113] 现在转向图3K。图3K的方法可以作为图3A的步骤308的一部分来执行。图3K示出了还可以在显示设备上渲染对粒子的正在进行的更新。因此,更新序列可以在显示设备上动画化,和/或粒子位置和粒子轨迹可以在显示设备上动画化。

[0114] 步骤3K02包括在缓冲器的存储器范围内更新粒子的更新位置。更新粒子的更新位置类似于在图3A中的步骤304生成更新位置,尽管正在执行连续的更新。

[0115] 步骤3K04包括针对多个更新位置中的每一个生成顶点。生成顶点类似于图3A中的步骤306。

[0116] 步骤3K06包括生成三角形作为粒子轨迹的可渲染的表示。步骤3K06是图3A中步骤308的变型。在图6至图12中示出了生成三角形的示例,从该三角形可以渲染粒子轨迹。

[0117] 现在参考图4A。图4A的方法是图3A的方法的变型。图3A的方法还可以使用图1的系统以及图17A和图17B的计算系统中所示的组件来实现。图4A至图4J示出了可以对多个粒子执行图3A至图3K的方法。在一个实施方式中,同时(即,并行地)执行多个粒子的处理,以便提高执行方法的处理时间。

[0118] 步骤400包括在缓冲器中定义粒子,该粒子在独立于图形用户界面(GUI)的绝对坐标系中具有初始位置,在该图形用户界面中粒子是可渲染的。步骤400可以以类似于图3A的步骤300所述的方式执行。

[0119] 步骤402包括将缓冲器内的存储器范围分配给相应的粒子。步骤402可以以类似于图3A的步骤302所描述的方式来执行。

[0120] 步骤404包括在存储器范围中生成粒子的更新位置。步骤404可以以类似于图3A的步骤304所述的方式执行。

[0121] 步骤406包括在存储器范围中根据初始位置和更新位置的组合确定每个粒子的相应顶点。步骤406可以以类似于图3A的步骤306所描述的方式来执行。

[0122] 步骤408包括根据相应的顶点创建粒子的相应粒子轨迹。步骤408可以以类似于图3A的步骤308所描述的方式来执行。

[0123] 步骤410包括将相应的粒子轨迹存储在与粒子相关联的存储器范围中。步骤410可以以类似于图3A的步骤310所描述的方式来执行。

[0124] 图4A所示的方法突出显示了一个或更多个实施方式的其他方面。例如,在一个实施方式中,存储器范围中的每个存储器范围被分配给粒子中的单个粒子和相应粒子轨迹中的单个粒子轨迹。换句话说,在该实施方式中,在给定的存储范围和给定的粒子以及相应的给定粒子轨迹之间存在一对一的对应关系。

[0125] 在另一示例中,创建相应粒子轨迹包含在存储器范围上执行顶点着色器应用,使得对粒子中的每一个粒子执行一次顶点着色器应用。换句话说,虽然多个顶点着色器应用可用于多个存储器范围中的多个粒子,但可使用对多个存储器范围中的一些或所有粒子和粒子轨迹进行操作的单个顶点着色器应用。

[0126] 现在参考图4B。图4B的方法可以在图4A的方法之后实现。

[0127] 步骤4B02包括将顶点着色器应用标记为无效着色器。步骤4B02类似于图3I中的步

骤3I02。

[0128] 现在转向图4C。图4C的方法可以在图4A的方法之后执行。

[0129] 步骤4C02包括在GUI上将粒子和相应的粒子轨迹渲染为单个绘制调用。步骤4C02类似于图3B中的步骤3B02。

[0130] 渲染可以在多个阶段中执行。在图5中示出了按阶段渲染的示例，并且关于图6至图12进一步详细描述。

[0131] 简而言之，在一个实施方式中，在第一渲染阶段中相应的顶点是劣化的且不可渲染的。术语“劣化”是指三角形的所有顶点都沿着单条线，使得多边形的特定区域为零，因此不能在屏幕上渲染为形状。图9中示出了劣化顶点的示例。

[0132] 继续该实施方式，在第一渲染阶段之后的第二渲染阶段中，相应顶点和附加顶点形成三角形作为相应粒子轨迹的可渲染表示。图12中示出了形成三角形的示例。

[0133] 第二渲染阶段可以使用其他多边形。例如，在第一渲染阶段之后的第二渲染阶段中，相应顶点和附加顶点形成三角形，三角形形成矩形四边形，作为相应粒子轨迹的可渲染表示。图16示出了形成矩形四边形的示例。

[0134] 因此，最一般地，在第一渲染阶段之后的第二渲染阶段中，相应顶点和附加顶点形成多边形作为相应粒子轨迹的可渲染表示。多边形可以由三角形的组合形成。多边形也可以形成独立的多边形形状(例如，矩形、五边形、六边形、八边形等)。

[0135] 现在参考图4D。图4D的方法可以在图4A的方法之后执行。

[0136] 步骤4D02包括将粒子的更新位置和相应的粒子轨迹从绝对坐标系投影到GUI的第一屏幕坐标系，以形成第一投影。步骤4D02类似于图3C中的步骤3C02。

[0137] 步骤4D04包括在第一屏幕坐标系中在GUI上渲染粒子和相应粒子轨迹。步骤4D04类似于图3C中的步骤3C04。

[0138] 现在参考图4E。图4E的方法可以在图4D的方法之后执行。

[0139] 步骤4E02包括接收改变第一投影的指令。步骤4E02类似于图3D中的步骤3D02。

[0140] 步骤4E04包括将粒子的更新位置和相应的粒子轨迹从绝对坐标系投影到屏幕坐标系以形成第二投影。步骤4E04类似于图3D中的步骤3D04。

[0141] 步骤4E06包括使用第二投影在第二屏幕坐标系中在GUI上重新渲染粒子和相应粒子轨迹。步骤4E06类似于图3D中的步骤3D06。

[0142] 现在参考图4F。图4F的方法可以在图4D的方法之后或者在图4E的方法之后执行。

[0143] 步骤4F02包括确定在第二屏幕坐标系中多个粒子中的一个粒子不再是可见的。步骤4F02类似于图3D中的步骤3D08。

[0144] 步骤4F04包括从缓冲器中对粒子进行清退以形成清退粒子。步骤4F04类似于图3D中的步骤3D10。

[0145] 步骤4F06包括通过在粒子中产生新粒子代替清退粒子来保持缓冲器的尺寸。步骤4F06类似于图3D中的步骤3D12。

[0146] 现在参考图4G。图4G的方法可以在图4F的方法之后执行。

[0147] 步骤4G02包括在缓冲器的多个存储器范围中的一个存储器范围中定义新粒子，新粒子在第二屏幕坐标系中是可见的并且在绝对坐标系中具有初始的新位置。步骤4G02类似于图3E中的步骤3E02。

[0148] 步骤4G04包括在多个存储范围中的一个存储范围中产生新粒子在绝对坐标系中的更新的新位置。步骤4G04类似于图3E中的步骤3E04。

[0149] 步骤4G06包括在多个存储范围中的一个存储范围中根据初始新位置和更新的新位置的组合确定新粒子的新顶点。步骤4G06类似于图3E中的步骤3E06。

[0150] 步骤4G08包括根据新顶点创建新粒子在绝对坐标系中的新粒子轨迹。步骤4G08类似于图3E中的步骤3E08。

[0151] 步骤4G10包括将新粒子踪迹存储在缓冲器内的多个存储器范围中的一个存储器范围中。步骤4G10类似于图3E中的步骤3E10。

[0152] 现在转到图4H。图4H的方法可以作为图4A的404的一部分来执行。

[0153] 步骤4H02包括对存储器范围中的每个粒子执行顶点着色器应用。步骤402类似于图3F中的步骤3F02。

[0154] 现在转到图4I。图4I的方法可以作为图4A的步骤404的一部分执行，并且还可以在图4H的步骤4H02之后执行。

[0155] 步骤4I02包括向粒子分配相应的标识符。步骤4I02类似于图3G中的步骤3G02。

[0156] 步骤4I04包括利用顶点着色器应用使用相应的标识符来跟踪粒子。步骤4I04类似于图3G中的步骤3G04。

[0157] 现在转到图4J。图4J的方法可以在图4A的方法之后执行，并且还可以作为图4A的步骤404的一部分。

[0158] 步骤4J02包括在缓冲器的存储器范围中跟踪粒子的元数据。步骤4J02类似于图3H中的步骤3H02。

[0159] 步骤4J04包括将元数据作为输入应用于顶点着色器应用。步骤4J04类似于图3H中的步骤3H04。同样，顶点着色器应用可被标记为无效着色器。

[0160] 现在参考图5。图5示出了根据一个或更多个实施方式的用于渲染粒子和粒子轨迹的方法的流程图。图5的方法可以由处理器执行的渲染应用来执行。因此，图5的方法可使用图1的系统或图17A和图17B所示的计算系统中的一个或更多个组件来实现。首先，给出图5的方法的概要。然后，给出关于图5的步骤的细节。

[0161] 在步骤500，执行调用以渲染帧。通过对渲染应用的请求来请求调用以渲染一系列帧中的要渲染的帧。

[0162] 在步骤502，执行第一渲染阶段。在第一阶段，使用顶点着色器计算更新粒子位置。

[0163] 在步骤504，执行第二渲染阶段。在第二阶段，使用缓冲器中的粒子位置信息生成可渲染的顶点。可渲染的顶点以期望的方式表示粒子轨迹。然后可渲染的顶点可用于渲染粒子轨迹。

[0164] 在步骤506，确定是否要处理新帧。如果是(在步骤506确定“是”)，则过程返回到步骤500并重复。如果不是(在步骤506确定为“否”)，则过程终止。

[0165] “帧”是测量粒子位置的时间帧。因此，如果每毫秒测量一次粒子的位置，则帧为1毫秒，并且每个帧将具有在给定时间点的粒子位置的测量。

[0166] 将注意力返回到一个或更多个实施方式的操作的概要。然后提供图5的方法的附加细节。

[0167] 一个或更多个实施方式针对每个渲染粒子使用小缓冲器来存储真实世界坐标(而

不是屏幕空间坐标)中的最近位置。因此,一个或更多个实施方式不使用屏幕空间缓冲器来存储投影粒子位置历史。

[0168] 然后,投影粒子历史被投影到屏幕空间中,并在GPU上被渲染为粒子轨迹,如果相机视点改变,则使用按需改变的投影,而不使存储的粒子位置无效。该过程允许随着显示改变(即,随着用户平移或缩放显示)连续地渲染相同的粒子及其历史。

[0169] 如果平移或缩放使得在可见屏幕空间内单个粒子不再是可见的,则该粒子被“清退”,并且其位置和历史的存储器然后被重新用于在当前可见区域内生成的新粒子。清退粒子允许在缩放和平移的同时更平滑地过渡和连续地渲染可见粒子,同时为任何新可见区域生成新粒子。

[0170] 粒子系统渲染任务分为两个阶段,如图5所示。渲染阶段1(步骤502)使用顶点着色器来计算更新粒子位置。顶点着色器运行的次数与粒子系统中存在的粒子一样多。渲染管线将所需数据传递到顶点着色器。传递到第一阶段顶点着色器的数据包含描述当前粒子位置的缓冲器以及其它相关元数据项,例如粒子当前年龄和最大年龄。

[0171] 假想三角测量对象的绘制命令使顶点着色器每粒子运行一次。将识别符整数传递到顶点着色器,所述顶点着色器可用于识别所述粒子。基于粒子系统的属性,粒子的位置被更新,并且新粒子位置被写回到位置缓冲器中,以便重写缓冲器中用于该粒子的最旧的位置。

[0172] 顶点着色器被标记为“无效”着色器。将顶点着色器标记为无效着色器意味着顶点着色器不具有相关联的片段着色器(fragment shader),因为在此阶段在屏幕上未产生任何内容。因此,将顶点着色器标记为无效进一步增加一个或一个以上实施方式的效率。

[0173] 渲染过程的渲染阶段2(步骤504)是其中使用粒子位置缓冲器来产生将以所需方式表示粒子轨迹的可渲染顶点的阶段。在图6至图12中提供的示例中,每个粒子的轨迹被渲染为三角形条带,其将表现为线或带。然而,所生成的顶点可以替代地表示虚线,每个粒子位置处的小球体,或表示粒子轨迹的任何数量的其它方式。

[0174] 对于以下给出的三角形条带示例,针对粒子轨迹中的每个位置(最新和最旧位置除外)生成一对可渲染顶点。在该阶段的顶点着色器中使每个粒子的第一顶点对和最后顶点对劣化。从每个粒子的当前和历史位置计算该阶段的顶点对,延伸出每个顶点到相反的方向,从而创建描绘粒子的历史轨迹的线。渲染阶段2(504)的最后部分是应用片段着色器,该片段着色器根据需要对粒子及其轨迹进行着色。

[0175] 虽然顺序地呈现和描述了流程图中的各个步骤,但是本领域普通技术人员将理解,可以以不同的顺序来执行这些步骤中的一些或全部,可以组合或省略这些步骤中的一些或全部,并且可以并行地执行这些步骤中的一些或全部。因此,一个或更多个实施方式不必受限于本文提供的示例。

[0176] 图6至图16示出了以上参照图1至图5描述的技术的具体示例。以下实施方式仅用于说明目的,而不旨在限制一个或更多个实施方案的范围。

[0177] 应当一起观察图6至图12。图6、图7和图8示出了根据一个或更多个实施方式的与渲染粒子和粒子轨迹相关联的表。图9示出了根据一个或更多个实施方式的关于在位置更新之后的粒子位置的顶点。图10和图11示出了根据一个或更多个实施方式的与渲染粒子和粒子轨迹相关联的附加表。图12示出根据一个或更多个实施方式的关于在附加更新之后的

粒子位置的可渲染的顶点。

[0178] 图6至图12的示例使用粒子位置的缓冲器，每个粒子位置被存储为二维“真实”世界位置。(然而，一个或更多个实施方式可以相对于任何数量的维度来使用。)使用当前相机(即，由用户选择的视点)在世界(即，地球)的当前可见部分内生成点。注意，二维足以描述物体在球体表面上的位置；因此，在一些实施方式中，二维可用于描述三维现实世界位置。

[0179] 如上所述，对于每个点，维护存储历史位置的缓冲器。然后，可以独立于相机角度、位置或参数使用历史位置来渲染粒子轨迹。

[0180] 历史位置阵列是环形缓冲器。因此，在存储器中保持指向阵列中可被最新位置重写的最老位置的索引。环形缓冲器数据结构允许在不复制任何值的情况下对缓冲器进行更新，从而提高效率。

[0181] 图6至图12的示例描述了包括两个粒子的粒子系统的处理。在这个例子中的粒子系统针对每个粒子存储五个位置，其中，这些位置表示在五个最近时间步长的每一个处粒子的位置。位置的数量可以根据应用而改变。

[0182] 参考图6，用初始粒子位置来准备粒子缓冲器，如表1(600)所示。因此，表1(600)示出了在渲染阶段1(502)之前在帧1处的粒子缓冲器。在该示例中，粒子在二维网格中从左向右移动。然而，类似的系统可以扩展到任何数量(“N”)的维度。在每个粒子的位置数据缓冲器之后，缓冲器还可以包含任何其它元数据项。

[0183] 在渲染的第一阶段中，在GPU上计算下一时间步长处粒子的位置，得到如表2所示的位置缓冲器(700)。因此，参考图7，表2(700)示出了在图5的渲染阶段1(502)之后在帧1处的粒子缓冲器。

[0184] 在渲染阶段1(502)处产生的输出被传递到渲染阶段2(504)。渲染阶段2(504)产生额外顶点以创建用于在显示装置上实际渲染的“三角形条带”。在该示例中，为每个粒子位置生成两个顶点。考虑到粒子宽度为1，在渲染阶段2(504)处顶点着色器输出如下。

[0185] 参考图8的表3(800)。通过散列突出显示的单元表示劣化的三角形的顶点。表3(800)示出了在帧1中在渲染阶段1(502)之后的粒子位置缓冲器。换言之，表3(800)示出了在一帧之后用于屏幕上渲染的计算出的顶点位置。

[0186] 图9是描绘在渲染阶段2(504)之后屏幕上的对应结果的图例(900)。轴(Y轴(902)和X轴(904))表示绝对坐标系。粒子位置和顶点位置如图例(906)所示被标记。因此，图9示出了在第一位置更新之后相对于粒子位置(正方形)的可渲染顶点(点)。

[0187] 可以看出，在第一帧之后(即在渲染阶段2(504)的输出处)，三角形条带仍然完全劣化。“劣化”是指三角形的所有顶点位于同一直线上。因为三角形条带是劣化的，所以不会在显示设备上渲染任何东西。

[0188] 然而，图5的方法以新的帧继续。换句话说，时间被提前一个单位，并且针对两个粒子测量新粒子位置。在新帧处重复图5中的两阶段过程。

[0189] 参考图10，表4(1000)示出了在五个帧之后的两个粒子的粒子位置缓冲器。因此，表4(1000)示出了在图5的方法的五次迭代之后在每个帧处跟踪粒子的粒子位置缓冲器。

[0190] 参考图11，表5(1100)示出了在五个帧之后用于在屏幕上渲染的顶点位置。因此，表5(1100)示出顶点条的相应第二阶段输出。

[0191] 图12是另一图例(1200)。图例(1200)类似于图9的图例(900)，但是它在使用图5的

方法执行了5个帧之后。因此，图例(1200)中的Y轴(902)、X轴(904)和图例(906)与图例(900)中的相同。

[0192] 与图9类似，图12的图例(1200)示出了在五帧中发生的附加位置更新之后相对于粒子位置(正方形)的可渲染顶点(点)。三角形现在至少是二维的，因此可以通过显示设备上的渲染应用来渲染。

[0193] 此后，可以执行数学变换以将渲染的三角形从绝对坐标系投影到屏幕坐标系。与如果必须清除缓冲器则必须重新组合然后重新计算位置和粒子轨迹相比，可以快速地计算投影。结果，当用户平移或缩放显示时，用户将感觉到所渲染的粒子和粒子轨迹的平滑过渡。

[0194] 虽然这里所示的示例描绘了三角形条带的创建和使用作为粒子轨迹的可渲染表示，但是渲染阶段2(504)的输出可以替代地生成粒子轨迹可能期望的任何替代表示。以下在图13至图16中呈现了替代表示的示例。

[0195] 应当一起考虑图13至图16。图13、图14和图15示出了根据一个或更多个实施方式的与渲染粒子和粒子轨迹相关联的附加表。图16示出了根据一个或更多个实施方式的用于在相对于图12的附加更新之后处理关于粒子位置的可渲染顶点的不同技术。因此，图13至图16是图6至图12所示示例的变型。

[0196] 图13到图16中所示的表类似于图7到图11中所示的表，但为简单起见将其截短。该过程也类似，因为仍然遵循图5的方法。

[0197] 与图6至图12中的示例一样，粒子系统具有两个粒子。在这个例子中的粒子系统针对每个粒子存储五个位置，其中，这些位置表示在五个最近时间步长的每一个处粒子的位置。位置的数量可以根据应用而改变。然而，为了简化该示例，在图13至图15中仅示出了一个位置。

[0198] 参考图13，表6(1300)是在渲染阶段1(502)之前在帧1处的粒子缓冲器。表6(1300)指示粒子缓冲器用初始粒子位置来准备，如图所示。

[0199] 在该示例中，粒子在二维网格中从左向右移动，但是相同的系统可以扩展到数量“N”维。在每个粒子的位置数据缓冲器之后，缓冲器还可以包含其它元数据项。例如，表6(1300)中的“年龄”和“最大年龄”项是与粒子相关联的元数据的示例。

[0200] 参考图14，表7(1400)示出了在渲染阶段1(502)之后在帧1处的粒子缓冲器。在GPU上计算在下一时间步长的粒子位置。粒子位置计算的结果在位置缓冲器中示出，如表7所示(1400)。

[0201] 渲染阶段1(502)的输出被传递到渲染阶段2(504)。同样，渲染阶段2(504)生成用于创建用于在显示器上实际渲染的所需几何形状的附加顶点。

[0202] 在该示例中，针对每个粒子位置生成四个顶点以形成由两个三角形组成的矩形四边形。考虑到四边长度为1，渲染阶段2(504)处的顶点着色器输出如下。

[0203] 参考图15，表8(1500)示出了在一帧之后用于在屏幕上渲染的计算出的顶点位置。换句话说，表8(1500)示出了在帧1中在渲染阶段1(502)之后的粒子位置缓冲器。

[0204] 图16示出了在5帧之后渲染阶段2(504)之后的屏幕上的相应结果。图例(1600)包括在绝对坐标系中具有单位“1”的Y轴(1602)和X轴(1604)。如图例(1606)所示，粒子位置显示为正方形，顶点位置显示为点。

[0205] 在五个帧之后的顶点位置每个粒子位置包括布置为矩形四边形的两个三角形(即,由两个三角形形成的矩形形状)。在如图16所示的渲染阶段2(506)之后,渲染程序可以在显示设备上渲染形状。

[0206] 图17A和图17B是根据一个或更多个实施方式的计算系统和网络的示例。一个或更多个实施方式可在专门设计成实现改进的技术结果的计算系统上实现。当在计算系统中实现时,本公开的特征和元素提供了优于不实现本公开的特征和元素的计算系统的显著技术进步。移动设备、台式机、服务器、路由器、交换机、嵌入式设备或其它类型的硬件的任何组合可以通过包括本公开中描述的特征和元件来改进。例如,如图17A所示,计算系统(1700)可包括一个或更多个计算机处理器(1702)、非持久存储设备(1704)(例如,易失性存储器,诸如随机存取存储器(RAM)、高速缓存存储器),持久存储设备(1706)(例如,硬盘、诸如光盘(CD)驱动器或数字多功能盘(DVD)驱动器的光驱动器、闪存等)、通信接口(1708)(例如,蓝牙接口、红外接口、网络接口、光接口等)。以及实现本公开的特征和元件的多个其他元件和功能。

[0207] 计算机处理器(1702)可以是用于处理指令的集成电路。例如,计算机处理器(1702)可以是处理器的一个或更多个核或微核。计算系统(1700)还可包括一个或更多个输入设备(1710),诸如触摸屏、键盘、鼠标、麦克风、触摸板、电子笔或任何其它类型的输入设备。

[0208] 通信接口(1708)可包括用于将计算系统(1700)连接到网络(未示出)(例如,局域网(LAN)、诸如因特网的广域网(WAN)、移动网络或任何其它类型的网络)和/或连接到诸如另一计算设备的另一设备的集成电路。

[0209] 此外,计算系统(1700)可包括一个或更多个输出设备(1712),诸如屏幕(例如,液晶显示器(LCD)、等离子显示器、触摸屏、阴极射线管(CRT)监视器、投影仪或其它显示设备)、打印机、外部存储器或任何其它输出设备。一个或更多个输出设备(1712)可以与输入设备(1710)相同或不同。输入和输出设备(1710和1712)可以本地或远程地连接到计算机处理器(1702)、非持久存储设备(1704)和持久存储设备(1706)。存在多个不同类型的计算系统,并且上述输入和输出设备(1710和1712)可以采取其它形式。

[0210] 可以将用于执行一个或更多个实施方式的计算机可读程序代码形式的软件指令全部或部分地临时或持久地存储在诸如CD、DVD、存储设备、磁盘、磁带、闪存、物理存储器或任何其它计算机可读存储介质等非瞬态计算机可读介质上。具体地,软件指令可以对应于计算机可读程序代码,其在由处理器执行时被配置为执行一个或更多个实施方式。

[0211] 图17A中的计算系统(1700)可以连接到网络或作为网络的一部分。例如,如图17B所示,网络(1720)可以包括多个节点(例如,节点X(1722)、节点Y(1724))。每个节点可以对应于计算系统(例如,图17A所示的计算系统(1700))或者组合的节点组可以对应于图17A所示的计算系统(1700)。作为示例,一个或更多个实施方式可在连接到其它节点的分布式系统的节点上实现。作为另一示例,一个或更多个实施方式可在具有多个节点的分布式计算系统上实现,其中一个或更多个实施方式的每个部分可位于分布式计算系统内的不同节点上。此外,上述计算系统(1700)的一个或更多个元件可位于远程位置并通过网络连接到其它元件。

[0212] 尽管在图17B中未示出,但是该节点可以对应于经由背板连接到其他节点的服务

器机箱中的刀片。作为另一个例子，节点可以对应于数据中心中的服务器。作为另一示例，节点可对应于具有共享存储器和/或资源的计算机处理器或计算机处理器的微核。

[0213] 网络(1720)中的节点(例如，节点X(1722)、节点Y(1724))可以被配置成为客户端设备(1726)提供服务。例如，节点可以是云计算系统的一部分。节点可以包括从客户端设备(1726)接收请求并向客户端设备(1726)发送响应的功能。客户机设备(1726)可以是计算系统，诸如图17A所示的计算系统(1700)。此外，客户机设备(1726)可包括和/或执行一个或更多个实施方式的全部或一部分。

[0214] 图17A和图17B中描述的计算系统(1700)或计算系统组可包括执行本文公开的各种操作的功能。例如，计算系统可以在相同或不同系统上的进程之间执行通信。采用某种形式的主动或被动通信的各种机制可以促进同一设备上的进程之间的数据交换。代表这些进程间通信的示例包括但不限于文件、信号、套接字、消息队列、管线、信号量、共享存储器、消息传递和存储器映射文件的实现。下面提供关于这些非限制性实施方式的进一步细节。

[0215] 基于客户机-服务器联网模型，套接字可用作允许在同一设备上的进程之间进行双向数据传输的接口或通信信道端点。首先，在客户机-服务器联网模型之后，服务器进程(例如，提供数据的进程)可创建第一套接字对象。接下来，服务器进程绑定第一套接字对象，从而将第一套接字对象与唯一的名称和/或地址相关联。在创建并绑定第一套接字对象之后，服务器进程然后等待并监听来自一个或更多个客户端进程(例如，寻找数据的进程)的传入连接请求。此时，当客户机进程希望从服务器进程获得数据时，客户机进程通过创建第二套接字对象开始。然后，客户端进程进行到生成至少包括第二套接字对象和与第一套接字对象相关联的唯一名称和/或地址的连接请求。然后，客户端进程将连接请求发送到服务器进程。根据可用性，服务器进程可接受连接请求，与客户端进程建立通信信道，或服务器进程忙于处理其它操作，可将连接请求排队在缓冲器中直到服务器进程就绪为止。已建立的连接通知客户端进程可以开始通信。作为响应，客户机进程可生成指定客户机进程希望获得的数据的数据请求。数据请求随后被发送到服务器进程。在接收到数据请求时，服务器进程分析该请求并收集所请求的数据。最后，服务器进程然后生成至少包括所请求的数据的回复，并将该回复发送到客户机进程。更通常地，数据可以作为数据报或字符流(例如字节)来传送。

[0216] 共享存储器是指虚拟存储器空间的分配，以便证实可以通过多个进程来传递和/或访问数据的机制。在实现共享存储器时，初始化过程首先在持久或非持久存储器中创建可共享段。创建之后，初始化进程随后安装可共享段，随后将可共享段映射到与初始化进程相关联的地址空间中。在安装之后，初始化过程继续识别并授予对一个或更多个授权过程的访问许可，所述授权过程还可以向可共享段写入数据和从可共享段读取数据。由一个进程对可共享段中的数据做出的改变可以立即影响也链接到可共享段的其它进程。此外，当授权进程之一访问可共享段时，可共享段映射到该授权进程的地址空间。通常，在任何给定时间，除了初始化进程之外，只有一个授权进程可以安装可共享段。

[0217] 在不脱离一个或更多个实施方式的范围的情况下，可以使用其它技术来在进程之间共享数据，诸如本申请中描述的各种数据。这些进程可以是相同或不同应用的一部分，并且可以在相同或不同计算系统上执行。

[0218] 除了在进程之间共享数据之外，执行一个或更多个实施方式的计算系统可以包括

从用户接收数据的功能。例如,在一个或更多个实施方式中,用户可以经由用户设备上的图形用户界面(GUI)提交数据。可以通过用户使用触摸板、键盘、鼠标、触摸屏或任何其他输入设备来选择一个或更多个图形用户界面窗口小控件或将文本和其他数据插入到图形用户界面窗口小控件中来经由图形用户界面提交数据。响应于选择特定项,关于该特定项的信息可由计算机处理器从持久或非持久存储器获得。在用户选择项时,可以响应于用户的选择在用户设备上显示所获得的关于特定项的数据的内容。

[0219] 作为另一示例,获得关于特定项的数据的请求可被发送到通过网络可操作地连接到用户设备的服务器。例如,用户可以选择用户设备的web客户端内的统一资源定位符(URL)链接,从而发起正被发送到与URL相关联的网络主机的超文本传输协议(HTTP)或其它协议请求。响应于该请求,服务器可以提取关于特定所选项的数据并将该数据发送到发起该请求的设备。一旦用户设备已经接收到关于特定项的数据,则可以响应于用户的选择在用户设备上显示所接收的关于特定项的数据的内容。进一步对于上述示例,在选择URL链接之后从服务器接收的数据可以提供超文本标记语言(HTML)的网页,该网页可以由web客户端渲染并显示在用户设备上。

[0220] 一旦例如通过使用上述技术或从存储器获得了数据,则在执行一个或更多个实施方式的一个或更多个实施方式中,计算系统可以从所获得的数据中提取一个或更多个数据项。例如,提取可以由图17A中的计算系统(1700)如下执行。首先,确定数据的组织样式(例如,语法、模式、布局),其可以基于以下中的一个或更多个:位置(例如,比特或列位置、数据流中的第N个令牌等)、属性(其中该属性与一个或更多个值相关联)、或分层/树结构(由不同细节层的节点层组成-例如在嵌套分组报头或嵌套文档部分中)。然后,在组织样式的上下文中,将原始的,未处理的数据符号流解析成令牌流(或分层结构)(其中每个令牌可以具有相关联的令牌“类型”)。

[0221] 接下来,使用提取标准从令牌流或结构中提取一个或更多个数据项,其中根据组织样式处理提取标准以提取一个或更多个令牌(或从分层结构中提取节点)。对于基于位置的数据,提取由提取标准标识的位置处的令牌。对于基于属性/值的数据,提取与满足提取标准的属性相关联的令牌和/或节点。对于分层/分层数据,提取与匹配提取标准的节点相关联的令牌。提取标准可以简单地作为标识符串,或者可以是呈现给结构化数据储存库的查询(其中数据储存库可以根据数据库模式或数据格式来组织,诸如可扩展标记语言(XML))。

[0222] 所提取的数据可由计算系统用于进一步处理。例如,图17A的计算系统(1700)在执行一个或更多个实施方式的同时可执行数据比较。数据比较可用于比较两个或更多个数据值(例如,A、B)。例如,一个或更多个实施方式可以确定是否A>B、A=B、A!=B、A<B。比较可通过将A、B和指定与比较相关的操作的操作码提交到算术逻辑单元(ALU)(即,对两个数据值执行算术和/或逐位逻辑运算的电路)中来执行。ALU输出操作的数值结果和/或与数值结果相关的一个或更多个状态标志。例如,状态标志可以指示数值结果是否是正数、负数、零等。通过选择适当的操作码,然后读取数值结果和/或状态标志,可以执行比较。例如,为了确定是否A>B,可以从A中减去B(即,A-B),并且可以读取状态标志以确定结果是否为正(即,如果A>B,则A-B>0)。在一个或一个以上实施方式中,可将B视为阈值,且如果A=B或如果A>B,则认为A满足阈值,如使用ALU所确定的。在一个或更多个实施方式中,A和B可以是向量,并且

将A与B进行比较需要将向量A的第一元素与向量B的第一元素进行比较，将向量A的第二元素与向量B的第二元素进行比较等。在一个或更多个实施方式中，如果A和B是串，则可以比较串的二进制值。

[0223] 图17A中的计算系统(1700)可以实现和/或连接到数据储存库。例如，一种类型的数据储存库是数据库。数据库是为便于数据检索、修改、重新组织和删除而配置的信息集合。数据库管理系统(DBMS)是为用户提供定义、创建、查询、更新或管理数据库的接口的软件应用。

[0224] 用户或软件应用可以向DBMS提交语句或查询。然后DBMS解释该语句。该语句可以是请求信息的选择语句、更新语句、创建语句、删除语句等。此外，该语句可以包括指定数据、数据容器(数据库、表、记录、列、视图等)、标识符、条件(比较操作符)、功能(例如加入、完全加入、计数、平均等)、排序(例如上升、下降)等的参数。DBMS可以执行该语句。例如，DBMS可以访问存储缓冲器、引用或索引文件，用于读、写、删除或其任何组合，以响应语句。DBMS可以从持久或非持久存储器加载数据，并执行计算以响应查询。DBMS可以将结果返回给用户或软件应用。

[0225] 图17A的计算系统(1700)可包括呈现原始和/或经处理的数据(诸如比较和其它处理的结果)的功能。例如，可以通过各种呈现方法来实现呈现数据。具体地，可以通过由计算设备提供的用户界面来呈现数据。用户界面可包括在诸如计算机监视器或手持计算机设备上的触摸屏等显示设备上显示信息的GUI。GUI可以包括组织示出什么数据以及如何向用户呈现数据的各种GUI窗口小控件。此外，GUI可以直接向用户呈现数据，例如通过文本呈现为实际数据值的数据，或者由计算设备渲染为数据的视觉表示，例如通过可视化数据模型。

[0226] 例如，GUI可以首先从软件应用获得请求在GUI内呈现特定数据对象的通知。接下来，GUI可确定与特定数据对象相关联的数据对象类型，例如，通过从数据对象内标识数据对象类型的数据属性获得数据。然后，GUI可以确定被指定用于显示该数据对象类型的任何规则，例如由软件框架为数据对象类指定的规则，或者根据由GUI定义的用于呈现该数据对象类型的任何本地参数。最后，GUI可从特定数据对象获得数据值，并根据该数据对象类型的指定规则在显示设备内渲染数据值的可视表示。

[0227] 还可以通过各种音频方法来呈现数据。具体地，数据可被渲染为音频格式并通过可操作地连接到计算设备的一个或更多个扬声器呈现为声音。

[0228] 数据也可以通过触觉方法呈现给用户。例如，触觉方法可以包括由计算系统生成的振动或其它物理信号。例如，可以使用由手持计算机设备产生的具有预定持续时间和振动强度的振动来向用户呈现数据，以传送数据。

[0229] 以上对功能的描述仅呈现由图17A的计算系统(1700)和图17B中的节点(例如，节点X(1722)、节点Y(1724))和/或客户端设备(1726)执行的功能的几个示例。可以使用一个或更多个实施方式来执行其他功能。

[0230] 虽然已经相对于有限数量的实施方式描述了一个或更多个实施方式，但是受益于本公开的本领域技术人员将理解，可以设计不脱离本文公开的一个或更多个实施方式的范围的其他实施方式。因此，一个或更多个实施方式的范围应仅由所附权利要求限制。

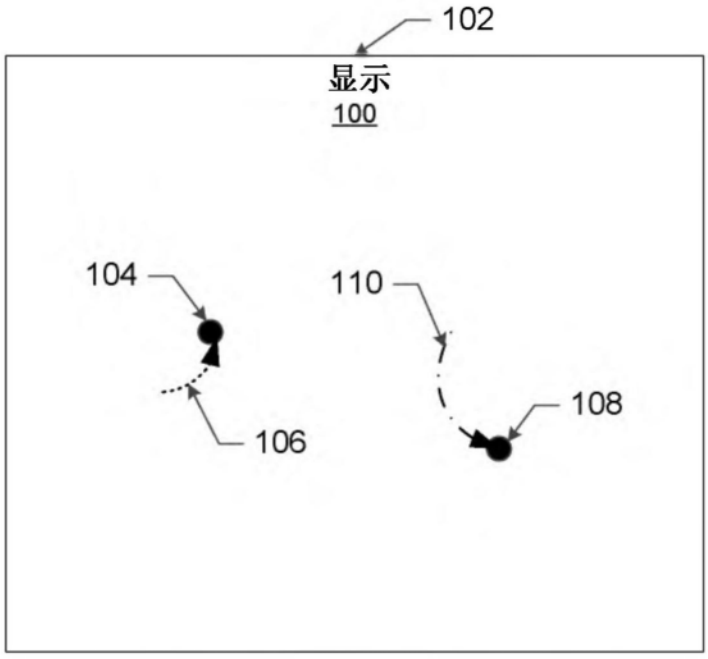


图1

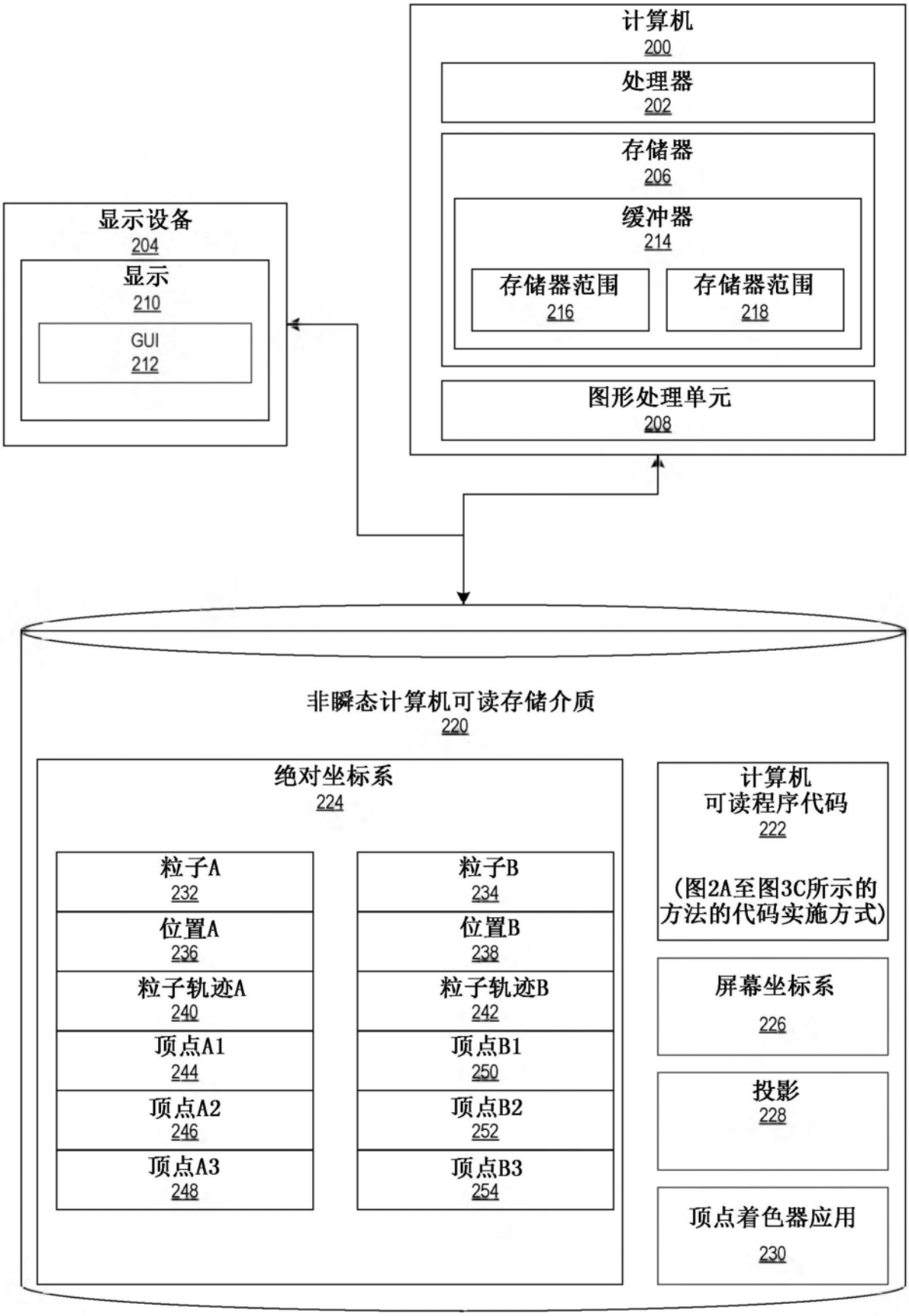

计算机
200
处理器
202
存储器
206
缓冲器
214
存储器范围
216
存储器范围
218
图形处理单元
208
显示设备
204
显示
210
GUI
212
非瞬态计算机可读存储介质
220
绝对坐标系
224
粒子A
232
位置A
236
粒子轨迹A
240
顶点A1
244
顶点A2
246
顶点A3
248
粒子B
234
位置B
238
粒子轨迹B
242
顶点B1
250
顶点B2
252
顶点B3
254
计算机
可读程序代码
222
(图2A至图3C所示的
方法的代码实施方式)
屏幕坐标系
226
投影
228
顶点着色器应用
230

图2

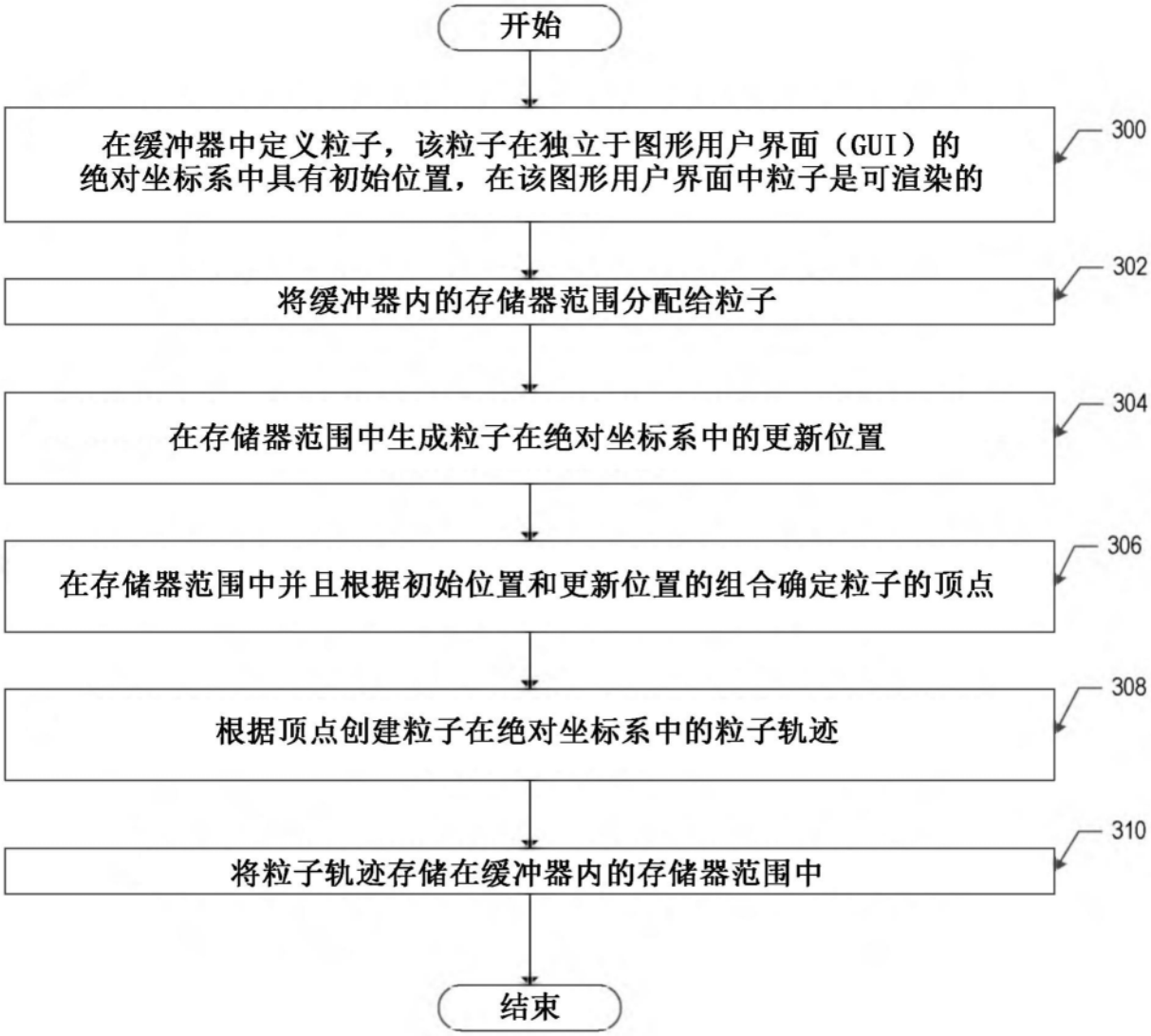

开始
在缓冲器中定义粒子，该粒子在独立于图形用户界面（GUI）的绝对坐标系中具有初始位置，在该图形用户界面中粒子是可渲染的
300
将缓冲器内的存储器范围分配给粒子
302
在存储器范围中生成粒子在绝对坐标系中的更新位置
304
在存储器范围中并且根据初始位置和更新位置的组合确定粒子的顶点
306
根据顶点创建粒子在绝对坐标系中的粒子轨迹
308
将粒子轨迹存储在缓冲器内的存储器范围中
310
结束

图3A

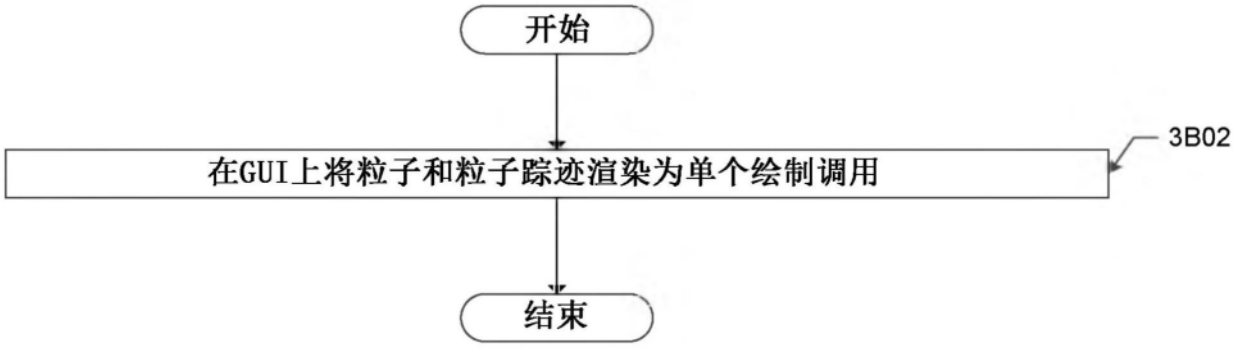

开始
在GUI上将粒子和粒子踪迹渲染为单个绘制调用
3B02
结束

图3B

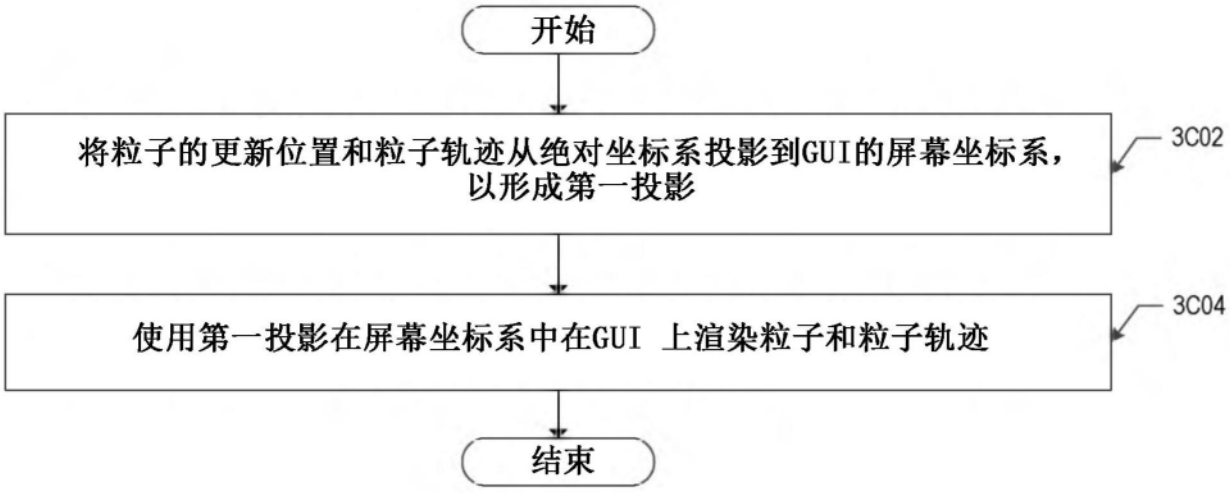

开始
将粒子的更新位置和粒子轨迹从绝对坐标系投影到GUI的屏幕坐标系，以形成第一投影
3C02
使用第一投影在屏幕坐标系中在GUI 上渲染粒子和粒子轨迹
3C04
结束

图3C

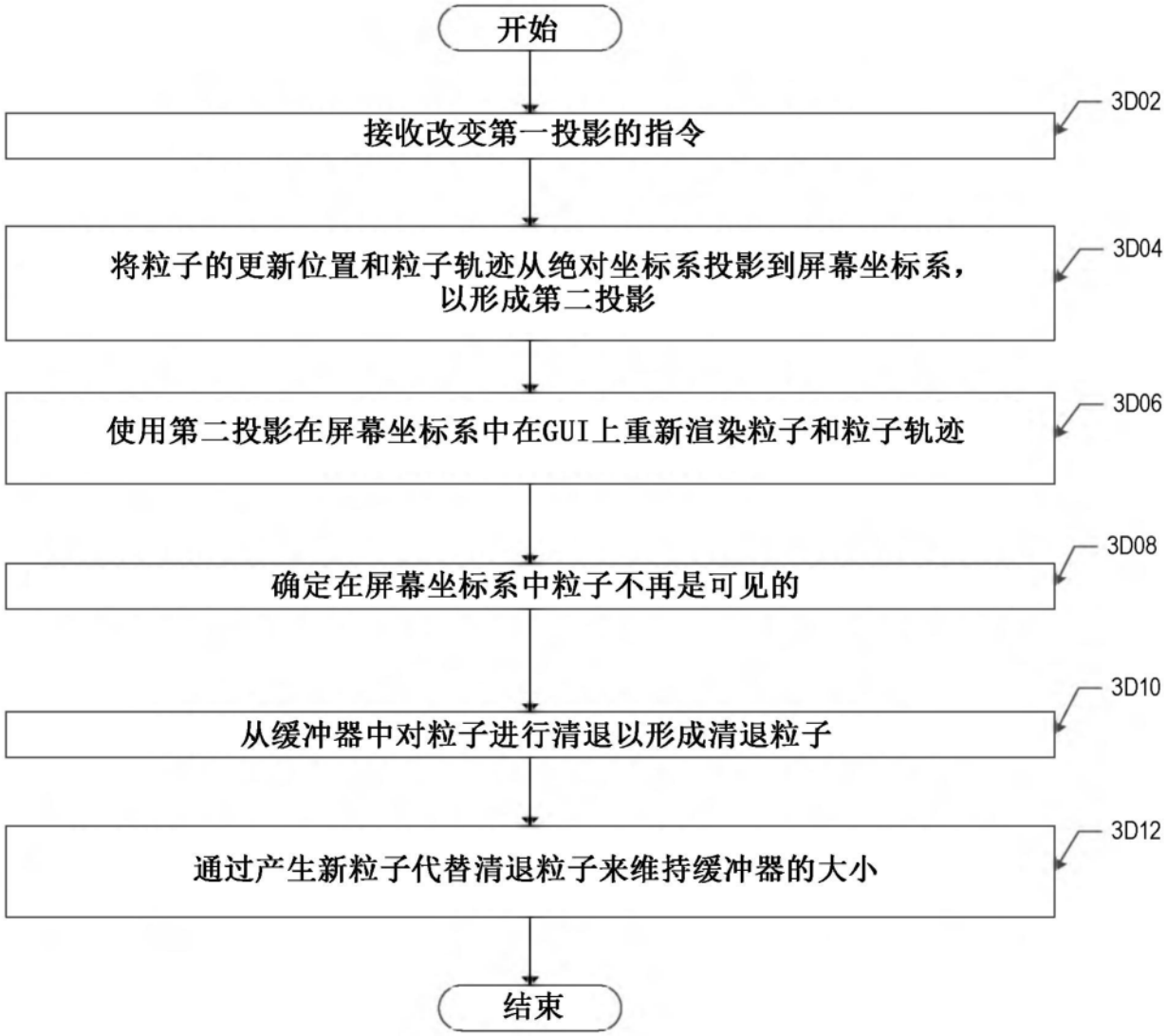

开始
接收改变第一投影的指令
3D02
将粒子的更新位置和粒子轨迹从绝对坐标系投影到屏幕坐标系，以形成第二投影
3D04
使用第二投影在屏幕坐标系中在GUI上重新渲染粒子和粒子轨迹
3D06
确定在屏幕坐标系中粒子不再是可见的
3D08
从缓冲器中对粒子进行清退以形成清退粒子
3D10
通过产生新粒子代替清退粒子来维持缓冲器的大小
3D12
结束

图3D

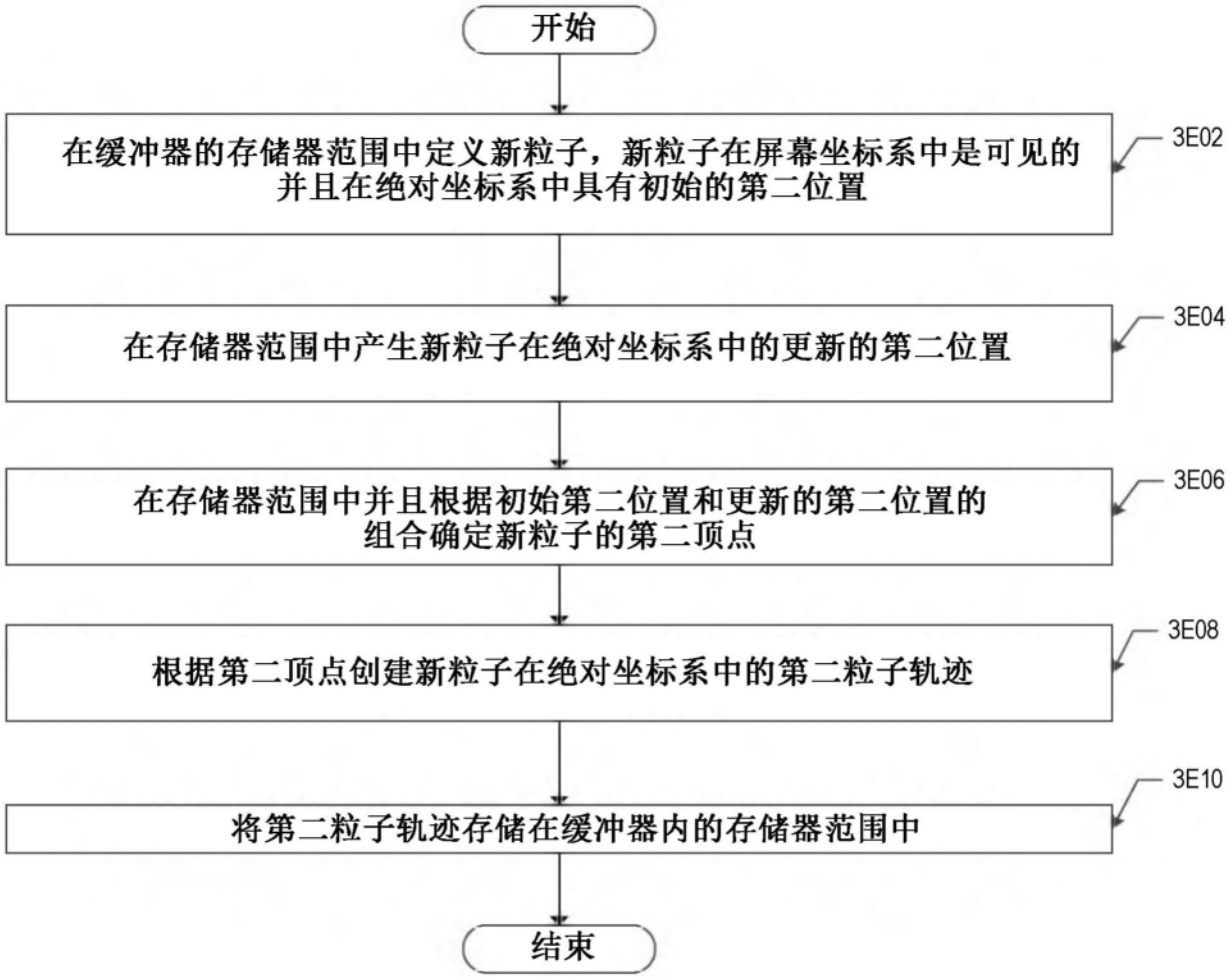

开始
在缓冲器的存储器范围中定义新粒子，新粒子在屏幕坐标系中是可见的并且在绝对坐标系中具有初始的第二位置
3E02
在存储器范围中产生新粒子在绝对坐标系中的更新的第二位置
3E04
在存储器范围中并且根据初始第二位置和更新的第二位置的组合确定新粒子的第二顶点
3E06
根据第二顶点创建新粒子在绝对坐标系中的第二粒子轨迹
3E08
将第二粒子轨迹存储在缓冲器内的存储器范围中
3E10
结束

图3E

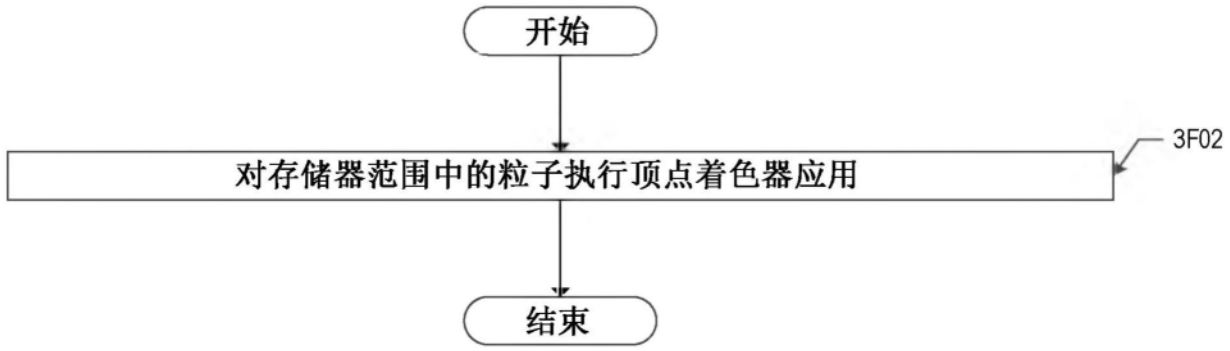

开始
对存储器范围中的粒子执行顶点着色器应用
3F02
结束

图3F

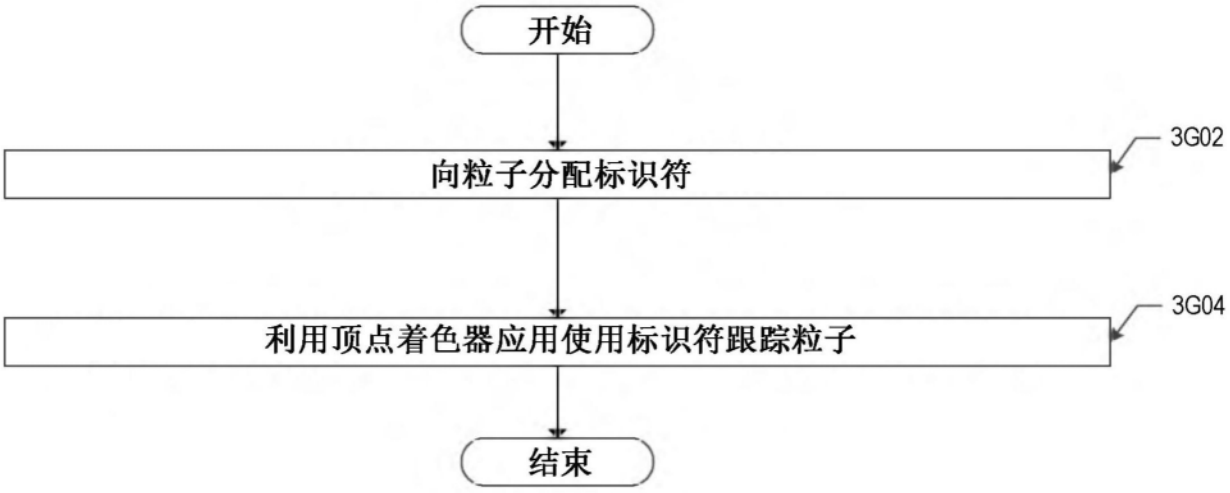

开始
向粒子分配标识符
3G02
利用顶点着色器应用使用标识符跟踪粒子
3G04
结束

图3G

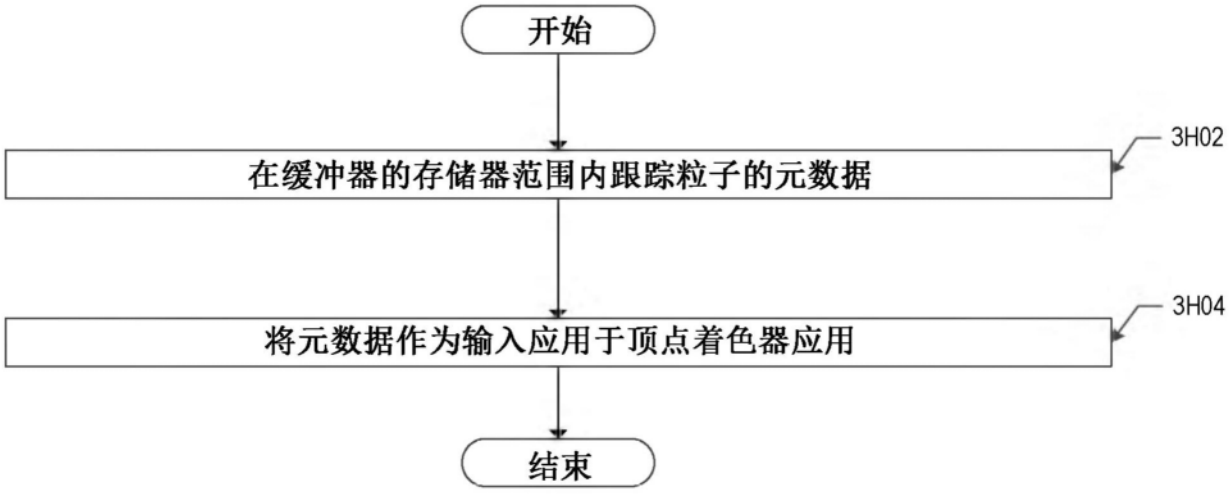

开始
在缓冲器的存储器范围内跟踪粒子的元数据
3H02
将元数据作为输入应用于顶点着色器应用
3H04
结束

图3H

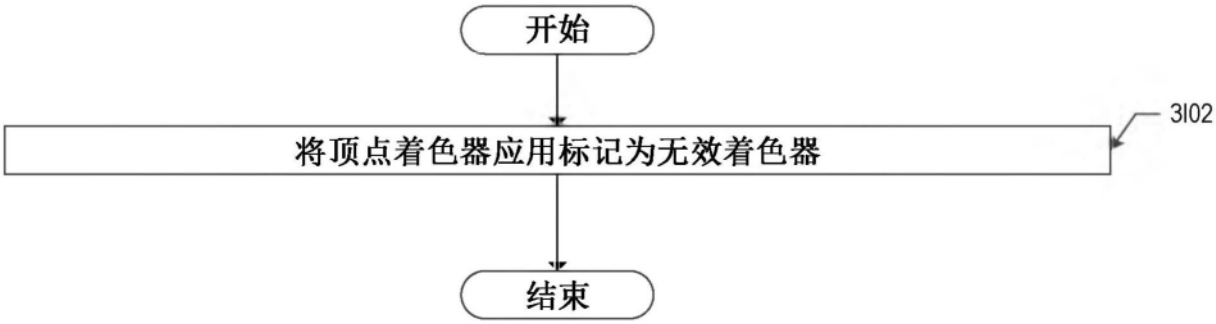

开始
将顶点着色器应用标记为无效着色器
3I02
结束

图3I

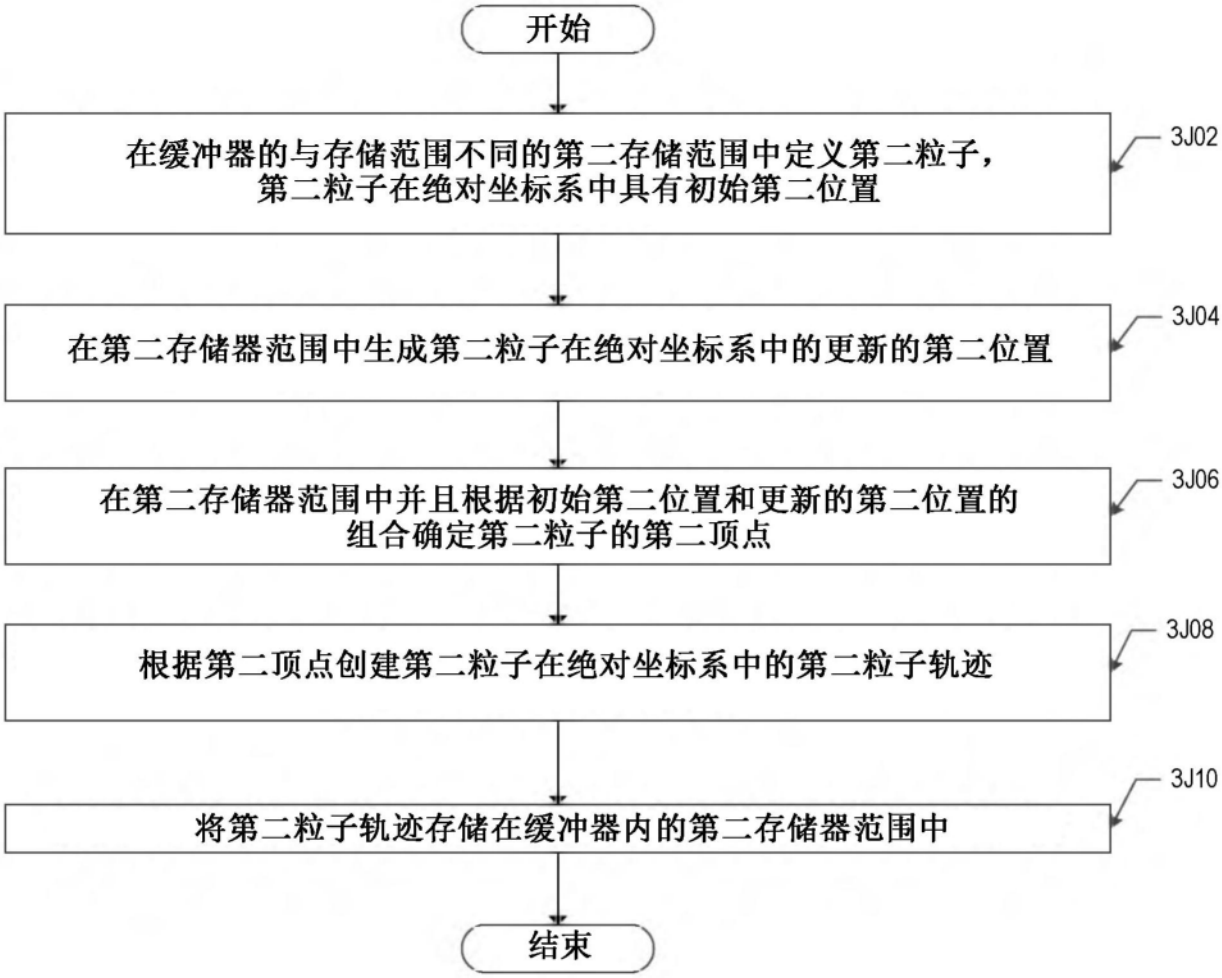

开始
在缓冲器的与存储范围不同的第二存储范围中定义第二粒子，第二粒子在绝对坐标系中具有初始第二位置
3J02
在第二存储器范围中生成第二粒子在绝对坐标系中的更新的第二位置
3J04
在第二存储器范围中并且根据初始第二位置和更新的第二位置的组合确定第二粒子的第二顶点
3J06
根据第二顶点创建第二粒子在绝对坐标系中的第二粒子轨迹
3J08
将第二粒子轨迹存储在缓冲器内的第二存储器范围中
3J10
结束

图3J

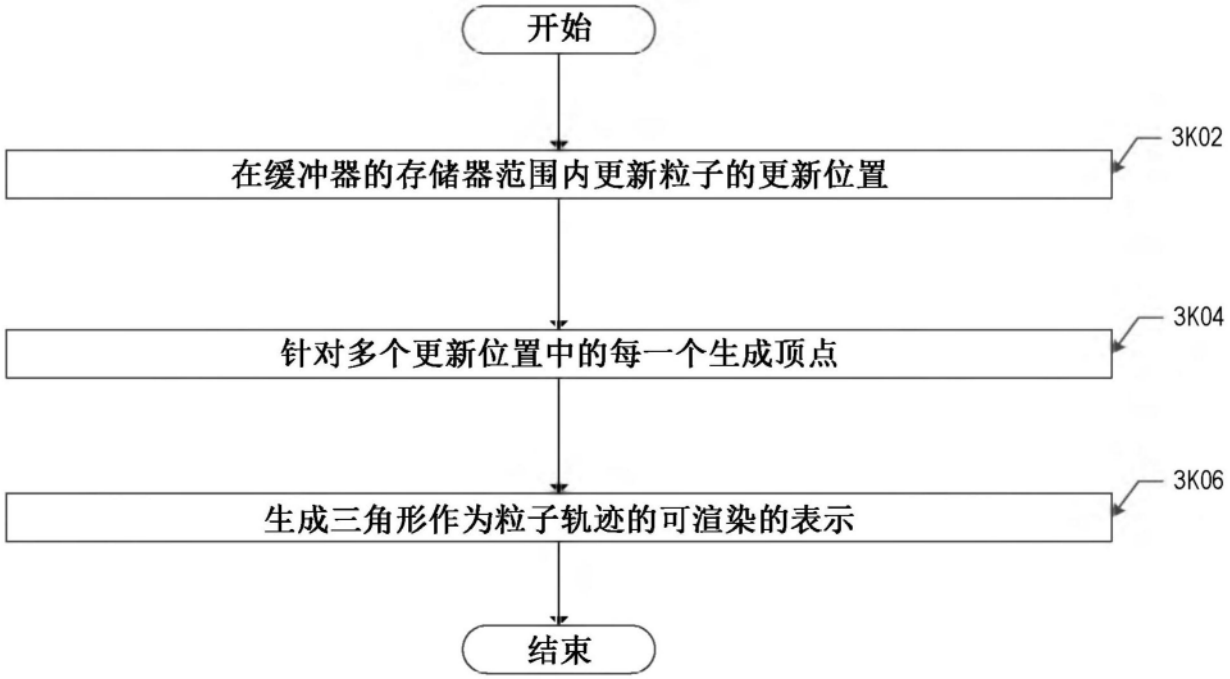

开始
在缓冲器的存储器范围内更新粒子的更新位置
3K02
针对多个更新位置中的每一个生成顶点
3K04
生成三角形作为粒子轨迹的可渲染的表示
3K06
结束

图3K

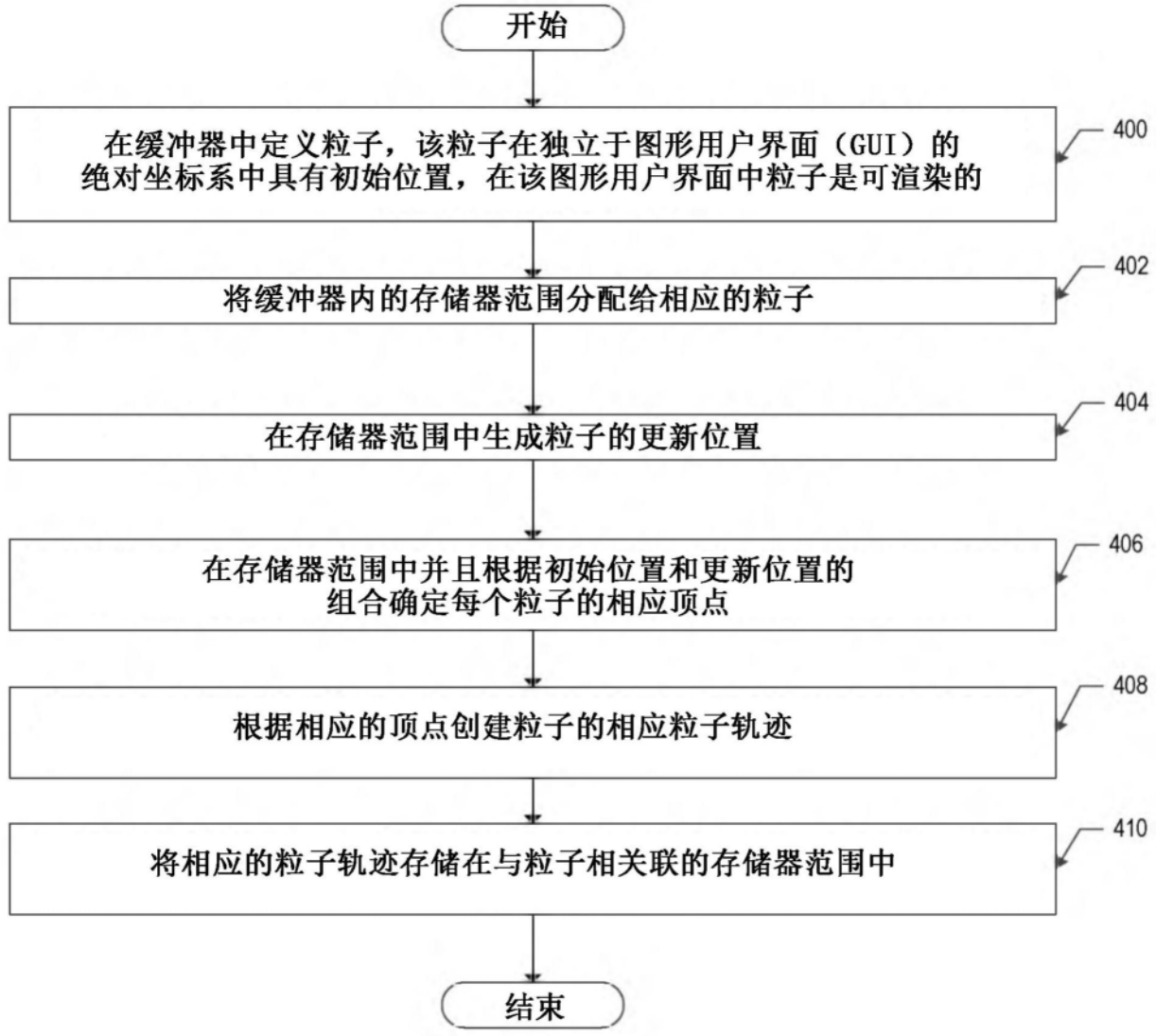

开始
在缓冲器中定义粒子，该粒子在独立于图形用户界面（GUI）的绝对坐标系中具有初始位置，在该图形用户界面中粒子是可渲染的
400
将缓冲器内的存储器范围分配给相应的粒子
402
在存储器范围中生成粒子的更新位置
404
在存储器范围中并且根据初始位置和更新位置的组合确定每个粒子的相应顶点
406
根据相应的顶点创建粒子的相应粒子轨迹
408
将相应的粒子轨迹存储在与粒子相关联的存储器范围中
410
结束

图4A

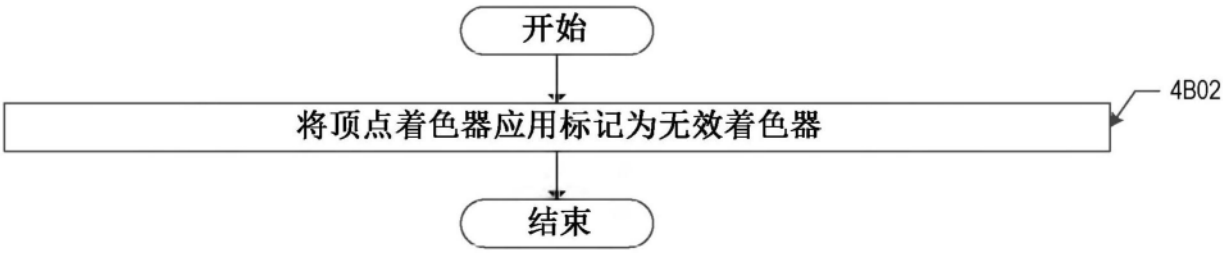

开始
将顶点着色器应用标记为无效着色器
4B02
结束

图4B

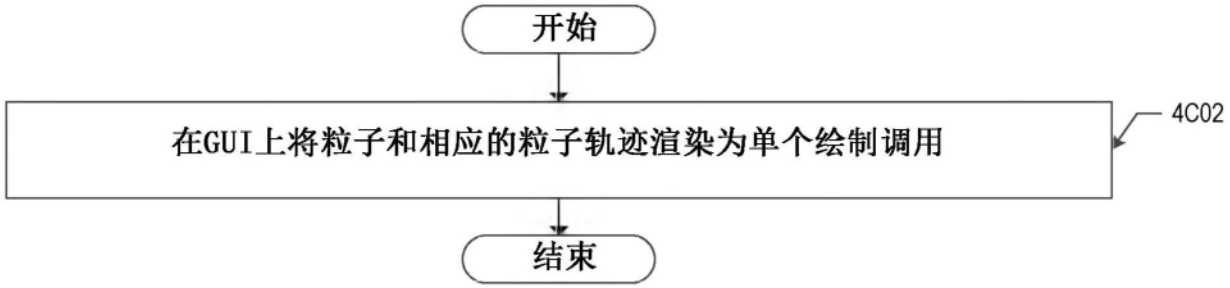

开始
在GUI上将粒子和相应的粒子轨迹渲染为单个绘制调用
4C02
结束

图4C

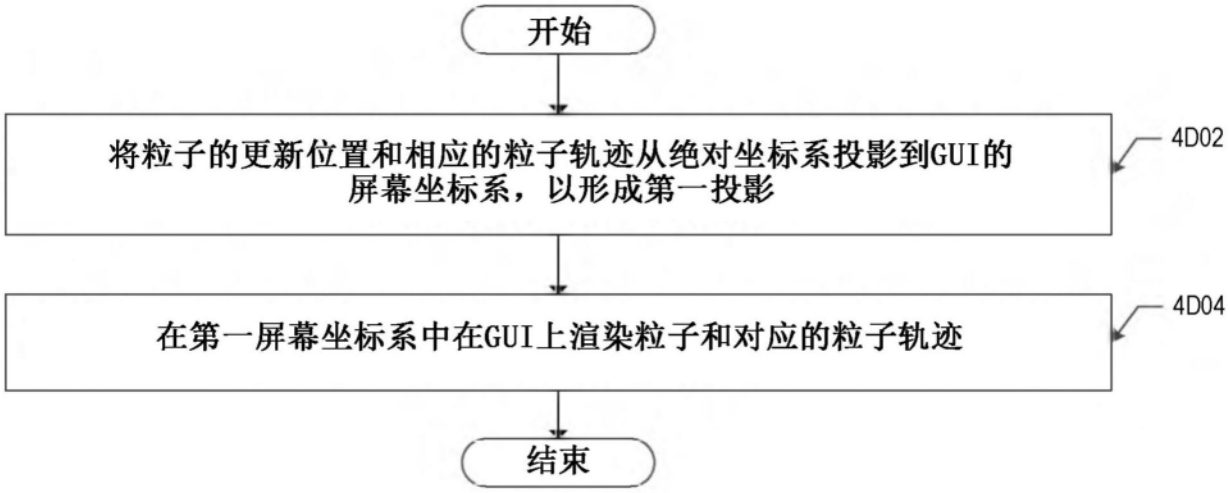

开始
将粒子的更新位置和相应的粒子轨迹从绝对坐标系投影到GUI的屏幕坐标系，以形成第一投影
4D02
在第一屏幕坐标系中在GUI上渲染粒子和对应的粒子轨迹
4D04
结束

图4D

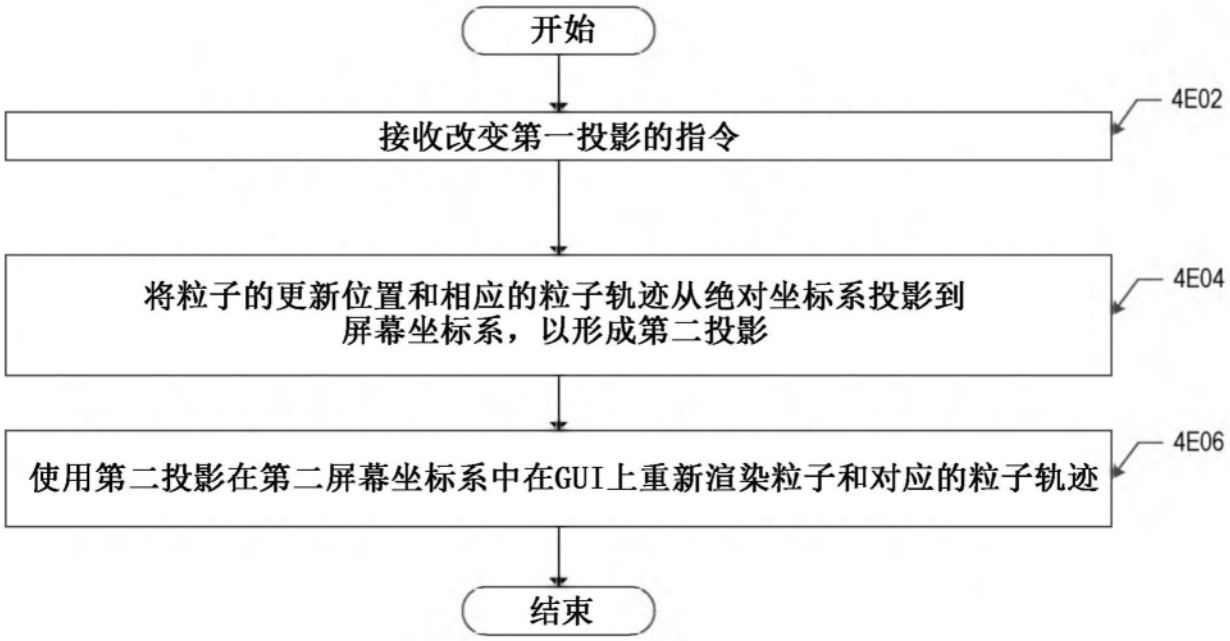

开始
接收改变第一投影的指令
4E02
将粒子的更新位置和相应的粒子轨迹从绝对坐标系投影到屏幕坐标系，以形成第二投影
4E04
使用第二投影在第二屏幕坐标系中在GUI上重新渲染粒子和对应的粒子轨迹
4E06
结束

图4E

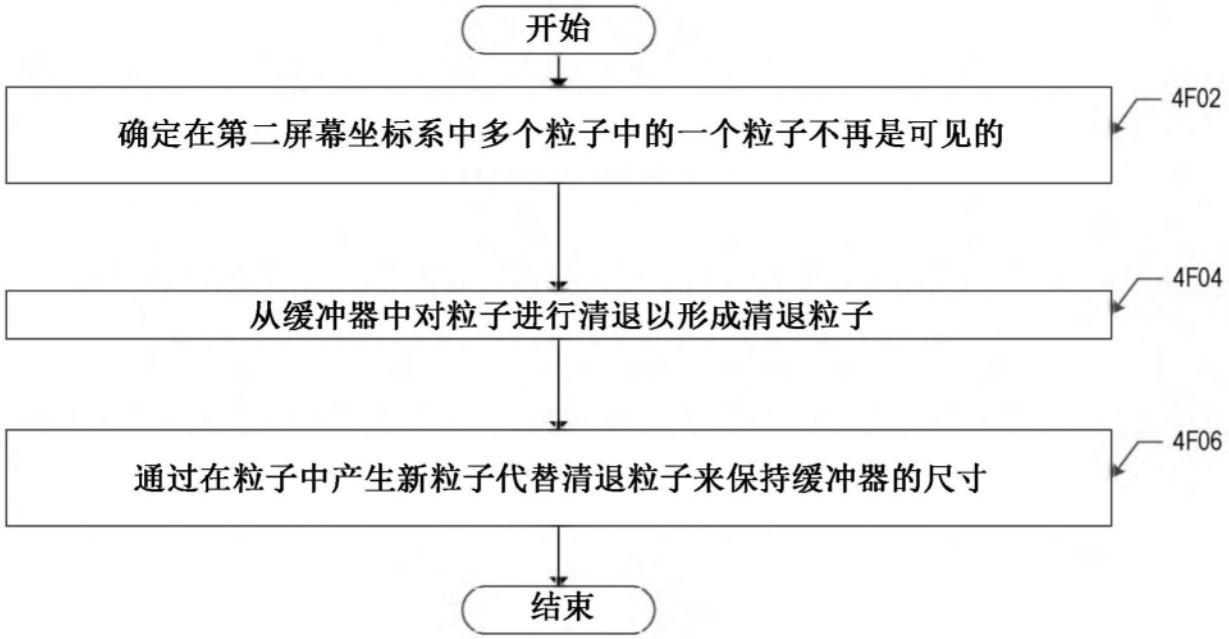

开始
确定在第二屏幕坐标系中多个粒子中的一个粒子不再是可见的
4F02
从缓冲器中对粒子进行清退以形成清退粒子
4F04
通过在粒子中产生新粒子代替清退粒子来保持缓冲器的尺寸
4F06
结束

图4F

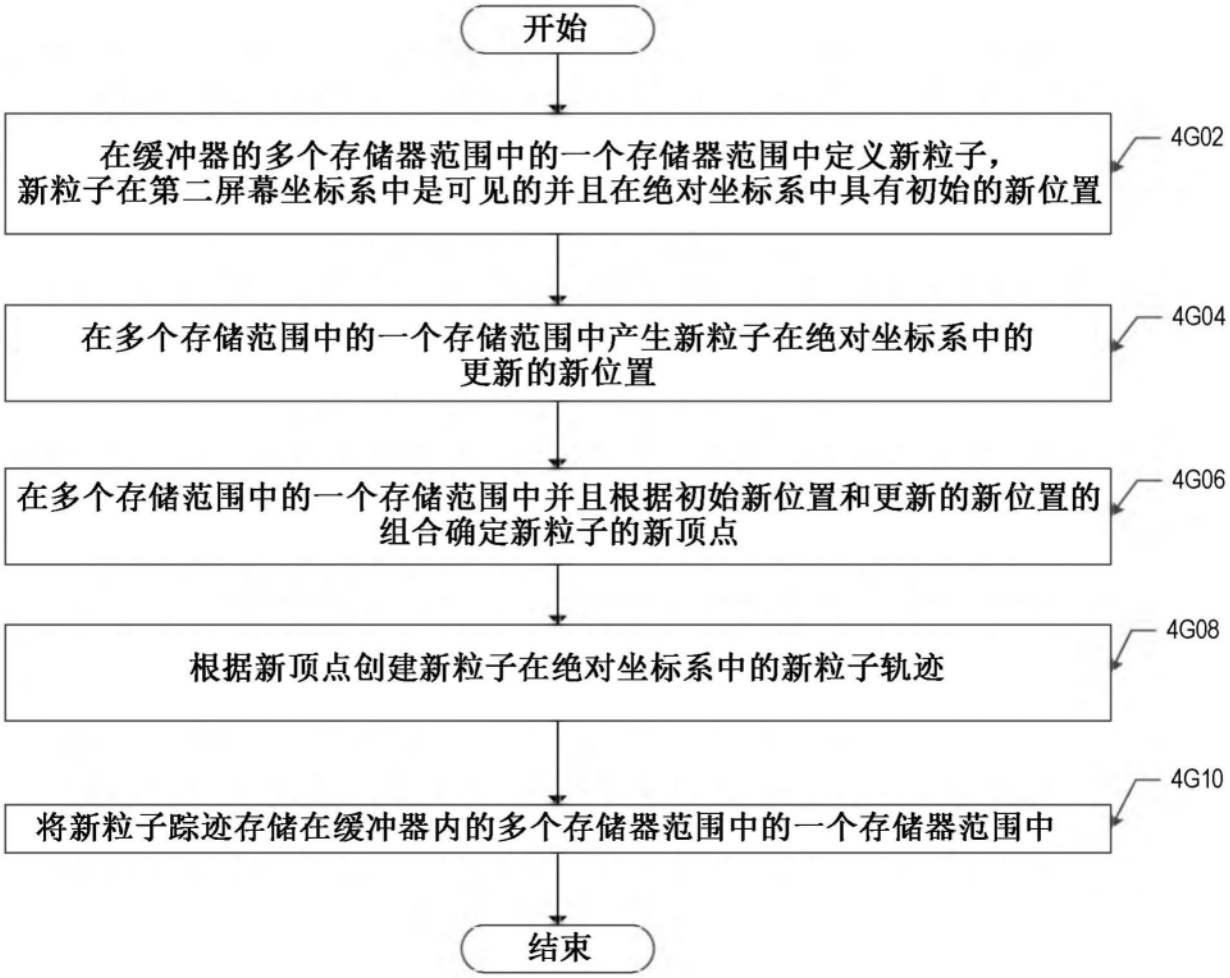

开始
在缓冲器的多个存储器范围中的一个存储器范围中定义新粒子，新粒子在第二屏幕坐标系中是可见的并且在绝对坐标系中具有初始的新位置
4G02
在多个存储范围中的一个存储范围中产生新粒子在绝对坐标系中的更新的新位置
4G04
在多个存储范围中的一个存储范围中并且根据初始新位置和更新的新位置的组合确定新粒子的新顶点
4G06
根据新顶点创建新粒子在绝对坐标系中的新粒子轨迹
4G08
将新粒子踪迹存储在缓冲器内的多个存储器范围中的一个存储器范围中
4G10
结束

图4G

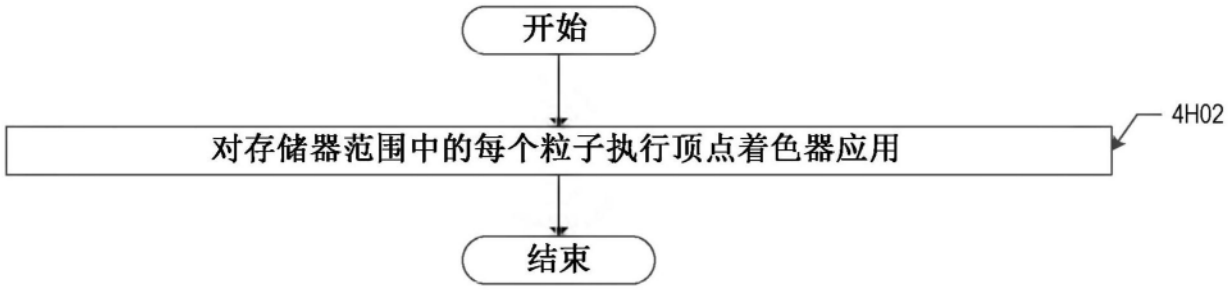

开始
对存储器范围中的每个粒子执行顶点着色器应用
4H02
结束

图4H

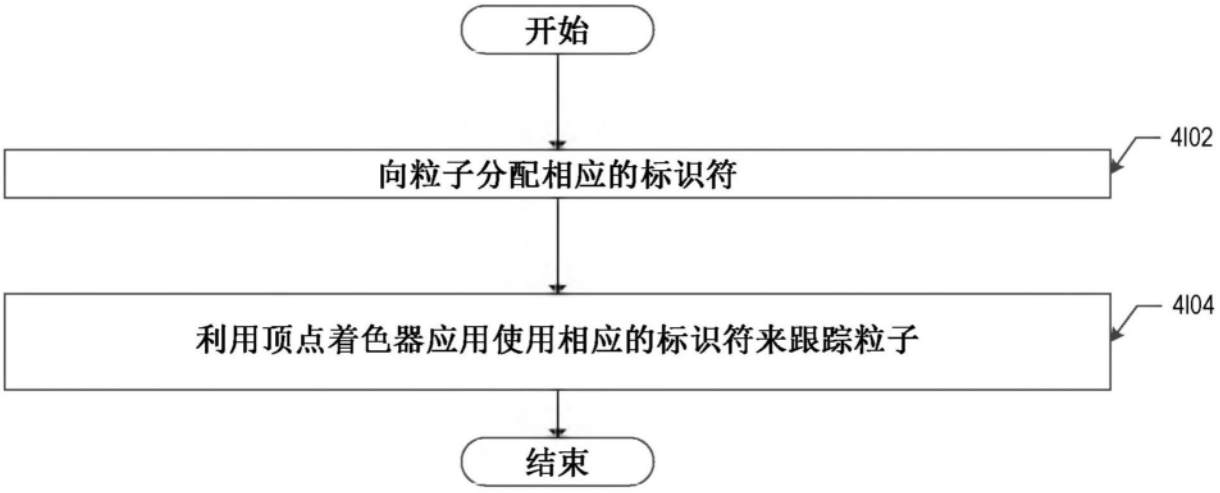

开始
向粒子分配相应的标识符
4I02
利用顶点着色器应用使用相应的标识符来跟踪粒子
4I04
结束

图4I

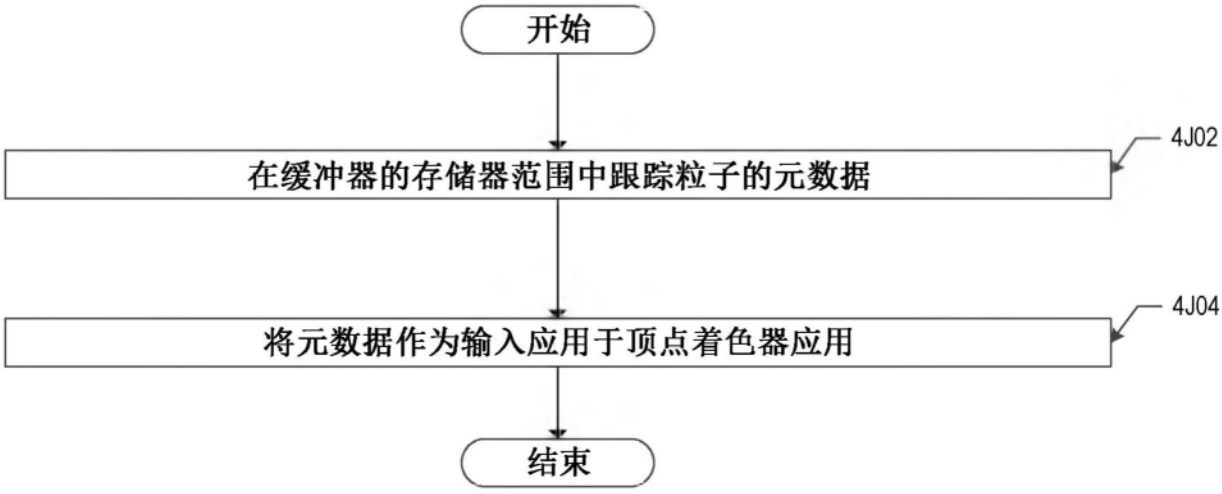

开始
在缓冲器的存储器范围中跟踪粒子的元数据
4J02
将元数据作为输入应用于顶点着色器应用
4J04
结束

图4J

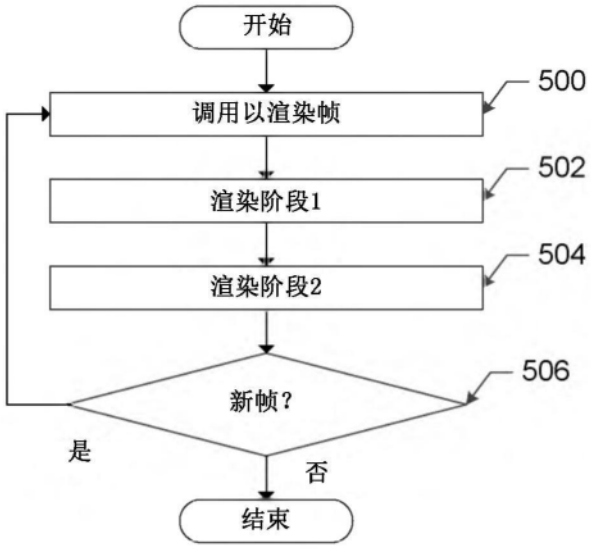


图5

600

表1. 在第一阶段渲染之前在帧1处的粒子缓冲器：

	位置	位置	位置	位置	位置	年龄	最大年龄
粒子1	(1,0)	nil	nil	nil	nil	0	10
粒子2	(6,0)	nil	nil	nil	nil	0	10

图6

700

表2. 在第一阶段渲染之后在帧1处的粒子缓冲器：

	位置	位置	位置	位置	位置	年龄	最大年龄
粒子1	(1,0)	(2,0)	nil	nil	nil	1	10
粒子2	(6,0)	(7,0)	nil	nil	nil	1	10

图7

800

表3. 在一帧之后用于屏幕上渲染的计算出的顶点位置

粒子1位置					粒子2位置				
顶点对1	顶点对2	顶点对3	顶点对4	顶点对5	顶点对6	顶点对7	顶点对8	顶点对9	顶点对10
(1,0)	(2,0)	(2,0)	(2,0)	(2,0)	(6,0)	(7,0)	(7,0)	(7,0)	(7,0)
(1,0)	(2,0)	(2,0)	(2,0)	(2,0)	(6,0)	(7,0)	(7,0)	(7,0)	(7,0)

图8

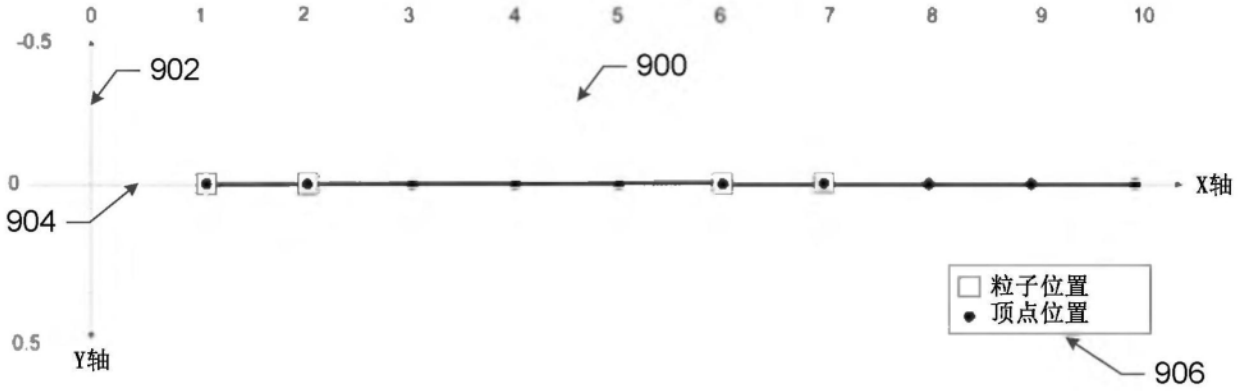


图9

1000

表4. 五个帧之后的粒子位置缓冲器

	位置	位置	位置	位置	位置	年龄	最大年龄
位置1	(1,0)	(2,0)	(3,0)	(4,0)	(5,0)	6	10
位置2	(6,0)	(7,0)	(8,0)	(9,0)	(10,0)	6	10

图10

1100

表5. 在五个帧之后用于在屏幕上渲染的计算出的顶点位置（三角形条带示例）

顶点对1	顶点对2	顶点对3	顶点对4	顶点对5	顶点对6	顶点对7	顶点对8	顶点对9	顶点对10
(1,0.0)	(2,-0.5)	(3,-0.5)	(4,-0.5)	(5,0.0)	(6,0.0)	(7,-0.5)	(8,-0.5)	(9,-0.5)	(10,0.0)
(1,0.0)	(2,0.5)	(3,0.5)	(4,0.5)	(5,0.0)	(6,0.0)	(7,0.5)	(8,0.5)	(9,0.5)	(10,0.0)

图11

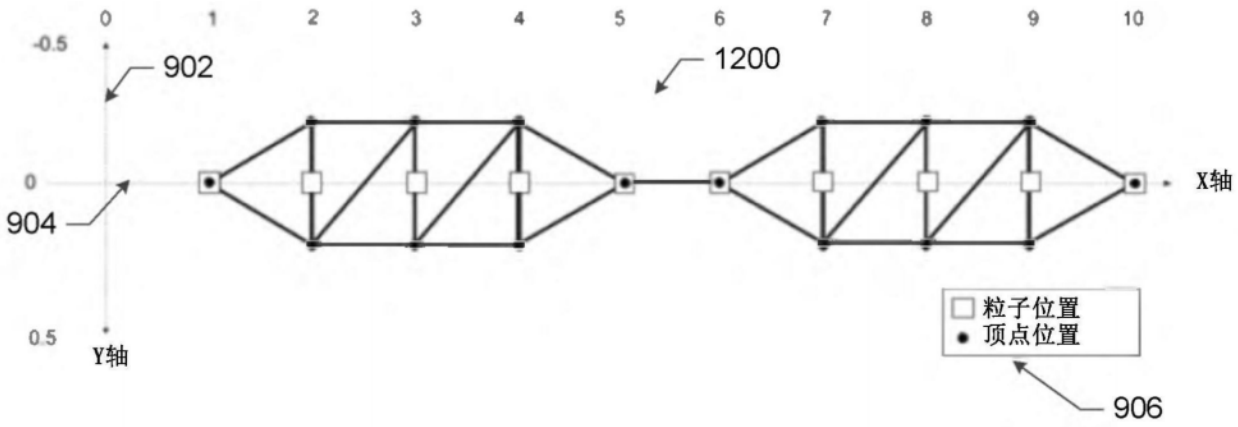


图12

1300

表6 在第一阶段渲染之前在帧1处的粒子缓冲器

	位置	年龄	最大年龄
粒子1	(1,0)	0	10
粒子2	(6,0)	0	10

图13

1400

表7 在第一阶段渲染之后在帧1处的粒子缓冲器

	位置	年龄	最大年龄
粒子1	(2,0)	1	10
粒子2	(7,0)	1	10

图14

1500

表8 在一个帧之后用于在屏幕上渲染的计算出的顶点位置

	粒子1顶点位置			
	顶点1	顶点2	顶点3	顶点4
粒子1	(0.5,-0.5)	(1.5,-0.5)	(1.5,0.5)	(0.5,0.5)
粒子2	(6.5,-0.5)	(7.5,-0.5)	(7.5,0.5)	(6.5,0.5)

图15

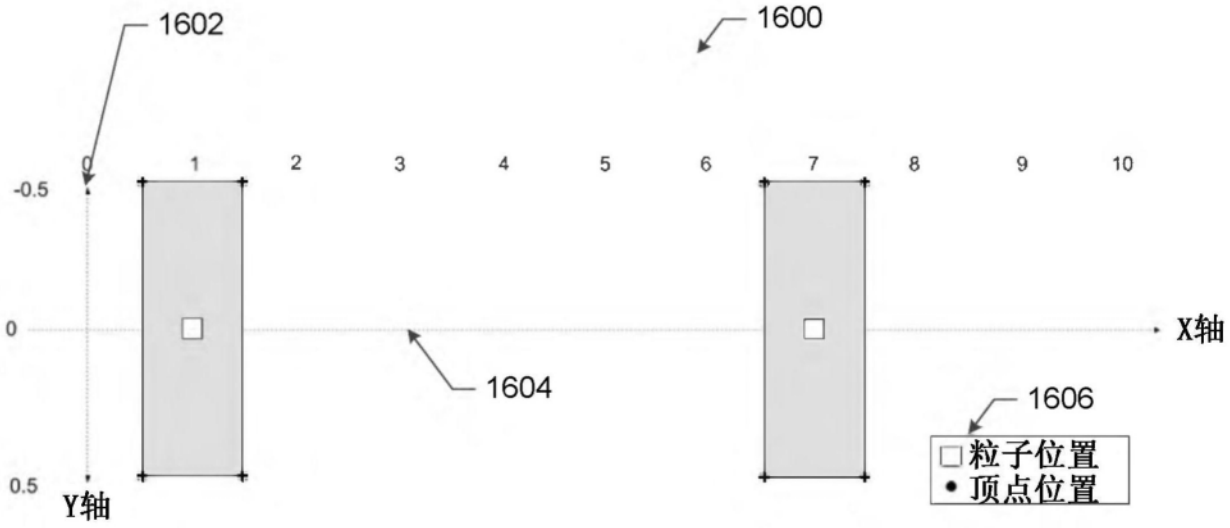


图16

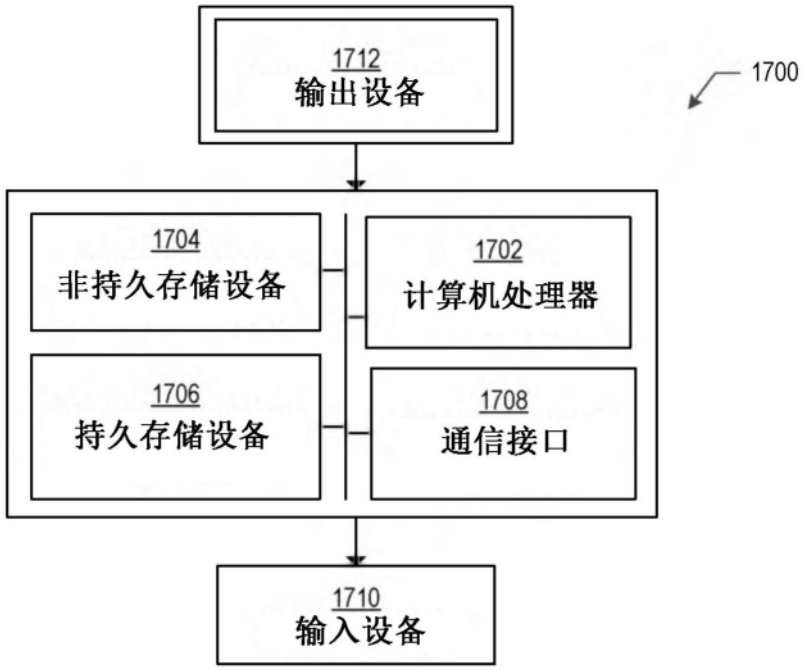

1712
输出设备
1700
1704
非持久存储设备
1702
计算机处理器
1706
持久存储设备
1708
通信接口
1710
输入设备

图17A

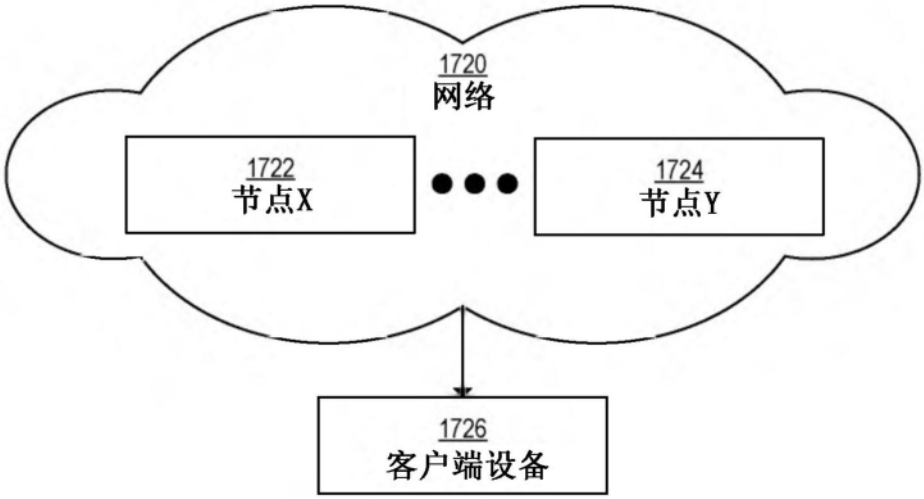

1720
网络
1722
节点X
1724
节点Y
1726
客户端设备

图17B