控制虚拟对象的方法、装置、电子设备及计算机可读介质

技术领域

本申请涉及计算机视觉技术领域，尤其涉及一种控制虚拟对象的方法、装置，电子设备及计算机可读介质。

背景技术

随着计算机技术的发展，互联网和计算机可以为用户提供虚拟场景，以使得用户可以获取更为丰富、多元的交互体验。例如，在虚拟场景中，可以为用户提供物理场景中并不存在、不易被观察的对象的虚拟对象，利用这样的虚拟对象与用户交互，可以丰富用户的视觉交互体验。另外，用户也可以通过在虚拟场景中控制虚拟对象的方式，获取“身临其境”的交互体验。

通常，虚拟对象可以基于控制指令的指示，来做出相应的动作。例如，控制指令可以指示虚拟对象中的控制点、关键点的位置移动，以使得虚拟对象相应地动作。由此，如何更好地对虚拟对象进行控制，使得虚拟对象有质量地做出动作，是值得关注和迫切需求的。

发明内容

本申请的多个方面提供一种控制虚拟对象的方法、装置，电子设备及计算机可读存储介质，可以通过将其他坐标系下生成的控制指令统一至场景坐标系下坐标的方式，使得控制指令可以实际、准确地被执行，能够避免因坐标系差异导致的控制错误、虚拟对象扭曲，能够提升对于虚拟对象的控制质量。

本申请的一方面，提供一种控制虚拟对象的方法，包括：接收针对虚拟场景中虚拟对象的控制点的控制指令；如果虚拟对象控制指令通过指示虚拟对象控制点在模型坐标系下的模型坐标的变化来控制虚拟对象控制点的动作，获取虚拟对象控制点在虚拟场景的场景坐标系下的场景坐标；基于虚拟对象场景坐标更新虚拟对象控制指令，生成更新后的控制指令，其中，虚拟对象更新后的控制指令通过指示虚拟对象控制点的虚拟对象场景坐标的变化来执行虚拟对象动作；以及利用更新后的控制指令控制虚拟对象控制点，以执行虚拟对象动作。

本申请的另一方面，提供一种用于控制虚拟对象的装置，包括：接收模块，被配置为接收针对虚拟场景中虚拟对象的控制点的控制指令；获取模块，被配置为如果虚拟对象控制指令通过指示虚拟对象控制点在模型坐标系下的模型坐标的变化来控制虚拟对象控制点的动作，获取虚拟对象控制点在虚拟场景的场景坐标系下的场景坐标；更新模块，被配置为基于虚拟对象场景坐标更新虚拟对象控制指令，生成更新后的控制指令，其中，虚拟对象更新后的控制指令通过指示虚拟对象控制点的虚拟对象场景坐标的变化来执行虚拟对象动作；以及第一控制模块，被配置为利用更新后的控制指令控制虚拟对象控制点，以执行虚拟对象动作。

本申请的另一方面，提供一种电子设备，该电子设备包括：至少一个处理器；以及与至少一个处理器通信连接的存储器；其中，存储器存储有可被至少一个处理器执行的指令，指令被至少一个处理器执行，以使至少一个处理器能够执行如上提供的控制虚拟对象的方法。

本申请的另一方面，提供一种计算机可读存储介质，其上存储有计算机程序指令，计算机程序指令可被处理器执行以实现如上提供的控制虚拟对象的方法。

本申请实施例提供的方案中，能够接收针对虚拟场景中虚拟对象的控制点的控制指令；如果控制指令通过指示控制点在模型坐标系下的模型坐标的变化来控制控制点的动作，获取控制点在虚拟场景的场景坐标系下的场景坐标；基于场景坐标更新控制指令，生成更新后的控制指令，其中，更新后的控制指令通过指示控制点的场景坐标的变化来执行动作；以及利用更新后的控制指令控制控制点，以执行动作。由此，通过将其他坐标系下生成的控制指令统一至场景坐标系下坐标的方式，使得控制指令可以实际、准确地被执行，能够避免因坐标系差异导致的控制错误、虚拟对象扭曲，能够提升对于虚拟对象的控制质量。

附图说明

为了更清楚地说明本申请实施例中的技术方案，下面将对实施例或现有技术描述中所需要使用的附图作一简单地介绍，显而易见地，下面描述中的附图是本申请的一些实施例，对于本领域普通技术人员来讲，在不付出创造性劳动性的前提下，还可以根据这些附图获得其它的附图。

通过阅读参照以下附图所作的对非限制性实施例所作的详细描述，本申请的其它特征、目的和优点将会变得更明显：

图1为本申请一实施例提供的控制虚拟对象的过程的示意图；

图2为本申请另一实施例提供的控制虚拟对象的过程的示意图；

图3为本申请一实施例提供的用于控制虚拟对象的装置的结构示意图；

图4为适用于实现本申请实施例中的方案的一种电子设备的结构示意图。

附图中相同或相似的附图标记代表相同或相似的部件。

具体实施方式

为使本申请实施例的目的、技术方案和优点更加清楚，下面将结合本申请实施例中的附图，对本申请实施例中的技术方案进行清楚、完整地描述，显然，所描述的实施例是本申请一部分实施例，而不是全部的实施例。基于本申请中的实施例，本领域普通技术人员在没有作出创造性劳动前提下所获得的所有其他实施例，都属于本申请保护的范围。

在本申请一个典型的配置中，终端、服务网络的设备均包括一个或多个处理器 (CPU)、输入/输出接口、网络接口和内存。

内存可能包括计算机可读介质中的非永久性存储器，随机存取存储器 (RAM) 和/或非易失性内存等形式，如只读存储器 (ROM) 或闪存(flash RAM)。内存是计算机可读介质的示例。

计算机可读介质包括永久性和非永久性、可移动和非可移动媒体，可以由任何方法或技术来实现信息存储。信息可以是计算机程序指令、数据结构、程序的模块或其他数据。计算机的存储介质的例子包括，但不限于相变内存 (PRAM)、静态随机存取存储器 (SRAM)、动态随机存取存储器 (DRAM)、其他类型的随机存取存储器 (RAM)、只读存储器 (ROM)、电可擦除可编程只读存储器 (EEPROM)、快闪记忆体或其他内存技术、只读光盘(CD-ROM)、数字多功能光盘 (DVD) 或其他光学存储、磁盒式磁带，磁带磁盘存储或其他磁性存储设备或任何其他非传输介质，可用于存储可以被计算设备访问的信息。

如上文说明的，如何更好地对虚拟对象进行控制，使得虚拟对象有质量地做出动作，是值得关注和迫切需求的。

在一些方案中，为了方便对虚拟对象进行控制，可以将虚拟对象模型化，或者利用模型来生成、控制虚拟对象。例如，可以基于所期望的动漫人物模型生成虚拟对象后，获取针对虚拟对象的控制点（例如，关键点、骨骼点等等）。然后，建立控制指令和控制点对应关系，以通过控制控制点的动作的方式，来控制虚拟对象的运动。通常，可以通过指示控制点的位置（坐标）移动的方式，例如，从A位置移动到B位置，来指示控制点的移动。

这样的方式中，在构建模型、构建虚拟对象的过程中，为了表征各个内容（例如，控制点）的位置，会针对模型来构建坐标系，来表征模型中各个部分的相对位置。这样的方式虽然有助于构建虚拟对象，但因其坐标系仅针对“虚拟对象”，在将虚拟对象添加、呈现在虚拟场景后，针对模型构建的坐标系中的控制点可能不再适宜控制虚拟对象。例如，在虚拟场景中添加虚拟对象时，控制点可能被虚拟场景赋予了新的场景坐标，如果仍基于模型坐标系下的坐标来控制虚拟对象的动作，可能会因为例如两个坐标系未对齐，而导致控制指令无法被正确识别，或者导致虚拟对象在虚拟场景中发生错误、异常的扭曲，影响控制质量。这一问题，尤其在不同用户针对同一虚拟对象进行控制时，更为明显。例如，不同的用户通过不同的模型来控制同一个虚拟对象，因各个用户的模型的构建坐标系可能不同，导致虚拟对象在被控制时，极容易出现左右摇头、角度怪异的问题。

对此，本申请实施例提供了一种控制虚拟对象的方法，该方法接收针对虚拟场景中虚拟对象的控制点的控制指令；如果控制指令通过指示控制点在模型坐标系下的模型坐标的变化来控制控制点的动作，获取控制点在虚拟场景的场景坐标系下的场景坐标；基于场景坐标更新控制指令，生成更新后的控制指令，其中，更新后的控制指令通过指示控制点的场景坐标的变化来执行动作；以及利用更新后的控制指令控制控制点，以执行动作。由此，通过将其他坐标系下生成的控制指令统一至场景坐标系下坐标的方式，使得控制指令可以实际、准确地被执行，能够避免因坐标系差异导致的控制错误、虚拟对象扭曲，能够提升对于虚拟对象的控制质量。

在实际场景中，该方法的执行主体可以是用户设备、或者用户设备与网络设备通过网络相集成所构成的设备，或者也可以是运行于上述设备的应用程序，用户设备包括但不限于计算机、手机、平板电脑、智能手表、手环等各类终端设备，网络设备包括但不限于如网络主机、单个网络服务器、多个网络服务器集或基于云计算的计算机集合等实现，可以用于实现设置闹钟时的部分处理功能。在此，云由基于云计算（Cloud Computing）的大量主机或网络服务器构成，其中，云计算是分布式计算的一种，由一群松散耦合的计算机集组成的一个虚拟计算机。

图1示出了本申请实施例提供的一种控制虚拟对象的过程100，过程100至少包括以下的处理步骤：

步骤S101，接收针对虚拟场景中虚拟对象的控制点的控制指令。

在本申请的实施例中，控制虚拟对象的主体（例如，用户）可以向执行主体通信控制指令，以对虚拟场景中的虚拟对象进行控制。例如，用户可以通过发送针对虚拟对象的关键点、骨骼点的坐标（例如，期望达到的、变化后的坐标），来指示虚拟对象的控制。以虚拟人体对象为示例，可以基于虚拟人体对象的部位划分，确定与各个部位划分结果相对应的控制点，控制点可以与对应的部位划分结果中的例如像素等相关联。以“头部”作为部分划分结果进行示例，如果在期望对虚拟人体对象的“头部”进行控制时（例如，移动）执行主体可以通过改变与“头部”对应控制点的位置，使得与“头部”相关联的各像素整体地被移动，以达到例如移动头部的动作效果。类似地，在一些场景中，针对某一个部分，控制点可以成组地被配置，以实现更细粒地运动、控制。例如，针对“手部”，可以例如针对一个手指，可以根据手指的各个骨骼点、关节点位置配置对应的控制点，以通过控制点之间的协同运动，来控制虚拟对象执行“手指动作”。

在一些实施例中，为方便用户使用，这样的控制指令也可以基于例如针对用户的动作捕捉数据、面部捕捉数据等提供。例如，通过对用户面部的一组采集点的动作数据采集，以生成与各个采集点相应的控制点的控制指令，以还原用户的面部动作。在一些实施例中，虚拟对象可以基于Unity引擎（或者说，平台）被渲染、添加至虚拟场景中，进行呈现。Unity是实时3D互动内容创作和运营平台，其可以用于游戏开发、美术、建筑、汽车设计、影视在内的所有创作者，以借助Unity将创意变成现实。 Unity平台提供一整套完善的软件解决方案，可用于创作、运营和变现任何实时互动的2D和3D内容，支持平台包括手机、平板电脑、PC、游戏主机、增强现实和虚拟现实设备。

在一些实施例中，为了方便用户添加、创作虚拟对象，如上述说明地，执行主体允许用户通过构建模型、提供模型封装文件的方式来提供虚拟对象。例如，虚拟对象可以是VRM（Virtual Reality Model）虚拟模型形象，或者说，基于VRM格式来提供虚拟对象。VRM，是一种用于表示和交换虚拟现实和增强现实应用中的人物模型的文件格式和标准。由此，利用VRM可以提供跨不同虚拟现实和增强现实平台的互操作性，以扩展虚拟对象的应用场景，丰富用户的使用、观看体验。

步骤S102，识别控制指令是否通过指示虚拟对象控制点在模型坐标系下的模型坐标的变化来控制虚拟对象控制点的动作。

在本申请的实施例中，本步骤旨在步骤S101的基础上，识别控制指令所指示的动作方式。执行主体可以识别控制指令是否通过指示虚拟对象控制点在模型坐标系下的模型坐标的变化，来控制虚拟对象中的控制点的动作。

模型坐标（Model Coordinate System）系中，模型坐标是相对于虚拟对象自身的局部坐标系，在模型坐标系下的坐标，可以被用于描述虚拟对象内部的位置和方向（例如，虚拟对象中各个控制点的位置和方向）。模型坐标系可以以虚拟对象的局部原点为中心，相对于虚拟对象自身进行定义。模型坐标系可以随着物体的变换（例如平移、旋转、缩放）而改变。相应地，执行主体可以基于控制指令中所指示需要被移动的控制点的坐标是否为模型坐标系下的坐标的方式，来识别其是否通过指示虚拟对象控制点在模型坐标系下的模型坐标的变化来控制虚拟对象控制点的动作。在一些实施例中，为了更便利、效率地识别控制指令的控制方式，或者说，识别所指示的动作实现方式，还可以在控制指令中添加例如方式标签、方式标识位，以利用方式标签、方式标识位来通信控制指令的识别方式。例如，执行主体可以基于控制指令中方式标识为“0”，获知控制指令是通过指示虚拟对象控制点在模型坐标系下的模型坐标的变化来控制虚拟对象控制点的动作。类似地，执行主体可以基于控制指令中方式标识为“1”，获知控制指令是通过指示控制点在其他坐标系下的坐标变化。

在一些实施例中，模型坐标可以被利用四元数形式表示。四元数（Quaternion）是一种数学概念，用于表示三维空间中的旋转，其是一种复数扩展，由一个实部和三个虚部组成，通常表示为(*w, x, y, z)*。四元数是由实部和虚部组成的扩展复数。示例性地，一个四元数可以表示为 *q = w + xi + yj + zk*，其中 *w* 是实部，(*x, y, z*) 是虚部对应的旋转轴上的分量。例如，一个常见的四元数表示为 *q = 1 + 2i + 3j + 4k*。这个四元数表示了绕以 (2, 3, 4) 为轴，角度为 2cos⁻¹(1/) 的旋转。由此，可以利用四元数形式来表征模型坐标，以方便指示其位置、角度，以及通过改变其坐标的方式直接来指示对应的控制点例如旋转的运动。

进一步地，如果执行主体确定控制指令是通过指示虚拟对象控制点在模型坐标系下的模型坐标的变化来控制虚拟对象控制点的动作的，则其可以继续执行步骤S103。

步骤S103，获取虚拟对象控制点在虚拟场景的场景坐标系下的场景坐标。

在本申请的实施例中，执行主体可以获取在虚拟场景的场景坐标系下，控制点的场景坐标。如上述说明地，场景坐标系可以是与虚拟场景对应的规定坐标系，以用于固定、确定地指示虚拟场景中各个位置。例如，场景坐标系可以是世界坐标系，世界坐标系是一个三维空间中的全局坐标系，用于描述虚拟对象在整个虚拟场景中的位置和方向。它是一个固定的坐标系，不会随着虚拟对象的变换而改变。在世界坐标系中，原点通常表示虚拟场景的原点或参考点，而坐标轴表示虚拟场景中的方向。在一些实施例中，也可以单独或同时地利用四元数形式来表示场景坐标，以方便指示其位置、角度，以及通过改变其坐标的方式直接来指示对应的控制点例如旋转的运动。

在一些实施例中，执行主体可以基于与虚拟对象相关联的参考点的参考模型坐标和参考场景坐标，来确定虚拟对象场景坐标。

具体地，执行主体可以确定至少一个与虚拟对象相关联的参考点。例如，虚拟对象中的一个或者多个控制点。在一些实施例中，执行主体，可以将用于定位虚拟对象在虚拟场景中添加位置的首点作为参考点。例如，执行主体可以根据用户的指示，从虚拟场景中确定一个用于添加、呈现虚拟对象的位置。例如，用户可以通过拖动、释放首点的方式，以指示首点在虚拟场景中的位置。又例如，用户可以通过设置首点场景坐标系下的坐标的方式，以指示首点在虚拟场景中的位置。进一步地，执行主体可以基于首点的放置位置以及虚拟对象中各个控制点与首点的位置关系，来放置各个控制点，以达到将虚拟对象添加至虚拟场景的目的。由此，以方便便利地指示虚拟对象的呈现位置。在一些实施例中，执行主体中也可以基于预先被配置的参数、描述信息等（例如，虚拟场景中用于呈现虚拟对象的默认位置）的方式，来确定首点的位置。由此，可与利用首点作为参照，来便利、完整地添加虚拟对象。

相应地，以将首点作为参考点为例，执行主体可以基于首点在模型坐标系下的模型坐标作为参考模型坐标、在场景坐标系下的场景坐标作为参考场景坐标。由此，以利用参考模型坐标和参考场景坐标来分析不同坐标系之间的对应关系，以实现坐标系间的转化、对齐。

在一些实施例中，执行主体可以获取虚拟对象参考模型坐标和虚拟对象参考场景坐标在各个坐标方向上的偏移值。例如，以包括(*x, y, z*)三个方向的模型坐标系和场景坐标系为示例，例如，参考点在模型坐标系下的模型坐标为(*x*1*, y*1*, z*1)的情况下，执行主体可以基于参考点在场景坐标系下的坐标(*x*2*, y*2*, z*2)，来获取参考场景坐标在各个坐标方向上的偏移值。例如，对于*x*方向，偏移值可以例如（*x*2*- x*1）。进一步地，执行主体可以利用虚拟对象模型坐标各个方向上的坐标值与相应的偏移值的组合结果，确定虚拟对象场景坐标。例如，对于控制点A，如果其模型坐标系下*x*方向上的坐标值为*x*3，则执行可以基于（*x*2*- x*1）与*x*3的组合，例如，*x*3*+x*2*-*1的数值结果，对其进行修正，以确定场景坐标系下对应控制点A的*x*方向上的场景坐标的值。由此，可以参考点数值偏移情况，来确定偏移值，以通过例如补偿值的方式，来确定控制点的场景坐标。

为方便理解，还可以同时参考图2。图2示出了本申请另一个实施例提供的控制虚拟对象的过程200。

在过程200中，执行主体可以接收到控制指令210，以被指示对虚拟对象220进行控制、执行例如旋转虚拟对象220的动作。例如，控制指令210可以通过指示控制点240在模型坐标系下的模型坐标的变化，来控制控制点240的动作、达到旋转虚拟对象220的目的。

在此情况下，因控制指令210是基于指示控制点240在模型坐标系下的模型坐标的变化来实现控制的。执行主体可以获取作与虚拟对象220相关联的参考点230的参考点场景坐标231、参考点模型坐标232，以及控制点240的控制点模型坐标241。进一步地，执行主体可以基于参考点场景坐标231、参考点模型坐标232和控制点模型坐标241来确定控制点240的控制点场景坐标242。

执行主体也可以选择利用参考模型坐标作为基准，基于模型坐标指向虚拟对象参考模型坐标的方向向量，以向量的方式来确定控制点的场景坐标。在一些实施例中，获取由虚拟对象模型坐标指向虚拟对象参考模型坐标的方向向量。

具体地，执行主体可以在模型坐标系下，基于控制点的模型坐标和参考模型坐标确定方向向量。应当理解的是，利用方向向量来确定场景坐标的方式下，需要模型坐标系与虚拟对象场景坐标系的单位距离一致，即，模型坐标系下的单位长度和场景坐标系下的单位长度一致。如果虚拟对象模型坐标系与虚拟对象场景坐标系的单位距离相同，则执行主体可以在场景坐标系下，以参考场景坐标为原点“还原”该方向向量，并基于“还原”的方向向量所指向的位置的场景坐标来确定控制点的场景坐标。由此，可以利用具有方向、长度的方向向量，以利用例如接近欧式距离的方式，通过相对位置关系来确定场景坐标，提升准确性、确定效率。

进一步地，在一些实施例中，如果虚拟对象模型坐标系与虚拟对象场景坐标系的单位距离不同相同。在此情况下，执行主体基于虚拟对象模型坐标系的单位距离与虚拟对象场景坐标系的单位距离的差异，修正虚拟对象方向向量，并基于虚拟对象参考场景坐标和修正后的方向向量，确定虚拟对象场景坐标。例如，执行主体可以基于模型坐标系的单位距离与虚拟对象场景坐标系的单位距离之间的比例，来确定缩放比例。然后，执行主体可以基于该缩放比例来缩放、修正方向向量，并基于虚拟对象参考场景坐标和修正后的方向向量，确定虚拟对象场景坐标。由此，能够避免因单位距离不同所导致的例如长度确定失准，保障坐标确定质量。

应当理解的是，为避免错误偏移，执行主体还可以通过调整原点位置、旋转坐标轴等方式对模型坐标系、模型坐标系进行调整，以使得调整后的模型坐标系中各个方向轴的朝向与场景坐标系对齐，能够避免因方向差异影响修正结果的准确性。

应当理解的是，执行主体可以在将虚拟对象添加至虚拟场景的过程中，或者，在完成将虚拟对象添加至虚拟场景的执行进程后，基于虚拟场景对应的场景坐标系，来获取各个控制点对应的场景坐标。

步骤S104，基于虚拟对象场景坐标更新虚拟对象控制指令，生成更新后的控制指令。

在本申请的实施例中，执行主体在基于上述步骤S103获取到控制点的场景坐标后，可以基于该场景坐标来对控制指令进行更新。例如，在以四元数形式来表示模型坐标和场景坐标的情况下，执行主体可以对控制点的坐标（例如，x、y、z方向上的坐标）进行更新后，再指示其按照指示的角度进行旋转。更新后的控制指令通过指示虚拟对象控制点的虚拟对象场景坐标的变化来执行虚拟对象动作。例如，更新后的控制指令可以通过指示控制点的场景坐标由场景坐标A变化为场景坐标B，以指示控制点在虚拟场景下进行移动。由此，更新后的控制指令可以通过直接作用于场景坐标的方式，使得控制点可以基于场景坐标系下的动作指令（例如，坐标变化）来被控制。由此，可能够避免例如因模型坐标系与场景坐标系中未对齐、对齐失准所导致的虚拟对象扭曲。

在一些实施例中，在基于参考形式进行动作的情况下，例如，执行主体通过指示一组控制点的相对运动，以使得虚拟对象做出某个动作的情况下。示例性地，例如一组控制节点可以包括与虚拟对象头部相关联的控制点A、控制点B和控制点C。控制指令可以通过指示控制点A、控制点B和控制点C对应的场景坐标（例如，方向向量）的变化指示各控制点相对于“首点”的移动状况来实现对虚拟对象的控制。

示例性地，可以继续参考图2。在过程200中，执行主体可以利用获取到的控制点场景坐标242对控制指令210进行更新，以得到更新后的控制指令210’。控制指令210’中，可以通过指示控制点场景坐标242的方式，来控制控制点240。

步骤S105，利用更新后的控制指令控制虚拟对象控制点，以执行虚拟对象动作。

在本申请的实施例中，在基于上述步骤S104完成对控制指令的更新后，执行主体可以利用更新后的控制指令控制虚拟对象控制点，以利用与场景坐标系相关联的控制指令来指示控制点的位置变化，以控制虚拟对象执行期望被执行的动作。例如，执行主体通过改变控制点的位置，以移动、旋转与控制点相关联的各像素元素，以达到控制虚拟对象、使虚拟对象做出相应动作的目的。

示例性地，可以继续参考图2。在过程200中，执行主体得到更新后的控制指令210’后，利用控制指令210’来控制控制点240的场景坐标的变化，以使得控制点240移动至控制点240’所示例的位置。进而，虚拟对象220可以基于控制点的位置变化，相应地动作，以达到旋转虚拟对象220的目的。

在一些实施例中，在上述步骤S102中，如果控制指令不是通过指示虚拟对象控制点在模型坐标系下的模型坐标的变化来控制虚拟对象控制点的动作，则在此情况下，可以选择直接执行步骤S106。

步骤S106，识别控制指令是否通过指示虚拟对象控制点在虚拟对象场景坐标系下的场景坐标的变化来控制虚拟对象控制点的动作。

具体地，执行主体可以基于识别控制指令是否通过指示虚拟对象控制点在虚拟对象模型坐标系下的模型坐标的变化来控制虚拟对象控制点的动作的方式，进一步地识别控制指令的实现方式，以确定控制指令是否通过指示虚拟对象控制点在虚拟对象场景坐标系下的场景坐标的变化来控制虚拟对象控制点的动作。

如果是，则可以继续执行步骤S107。

步骤S107，利用虚拟对象控制指令控制虚拟对象控制点，以执行虚拟对象动作。

具体地，如果执行主体识别出控制指令通过指示虚拟对象控制点在虚拟对象场景坐标系下的场景坐标的变化来控制虚拟对象控制点的动作，即，控制指令中直接通过指示控制点场景坐标的变化情况来实现对虚拟对象的控制的。在此情况下，执行主体则可以直接基于控制指令的指示来控制控制点，以实现动作。由此，对于不会导致扭曲的基于场景坐标系搭建的控制指令，其可以被执行主体直接执行，使得用户可以基于实际需求来个性化地提供控制指令，并保障控制指令的执行效果。

继而，本申请提供的控制虚拟对象的方法，接收针对虚拟场景中虚拟对象的控制点的控制指令；如果控制指令通过指示控制点在模型坐标系下的模型坐标的变化来控制控制点的动作，获取控制点在虚拟场景的场景坐标系下的场景坐标；基于场景坐标更新控制指令，生成更新后的控制指令，其中，更新后的控制指令通过指示控制点的场景坐标的变化来执行动作；以及利用更新后的控制指令控制控制点，以执行动作。由此，通过将其他坐标系下生成的控制指令统一至场景坐标系下坐标的方式，使得控制指令可以实际、准确地被执行，能够避免因坐标系差异导致的控制错误、虚拟对象扭曲，能够提升对于虚拟对象的控制质量。

本申请实施例还提供了一种用于控制虚拟对象的装置，该装置的结构如图3所示出的装置300。装置300包括：接收模块310，被配置为接收针对虚拟场景中虚拟对象的控制点的控制指令；获取模块320，被配置为如果虚拟对象控制指令通过指示虚拟对象控制点在模型坐标系下的模型坐标的变化来控制虚拟对象控制点的动作，获取虚拟对象控制点在虚拟场景的场景坐标系下的场景坐标；更新模块330，被配置为基于虚拟对象场景坐标更新虚拟对象控制指令，生成更新后的控制指令，其中，虚拟对象更新后的控制指令通过指示虚拟对象控制点的虚拟对象场景坐标的变化来执行虚拟对象动作；以及第一控制模块340，被配置为利用更新后的控制指令控制虚拟对象控制点，以执行虚拟对象动作。

在一些实施例中，装置300还包括：确定模块，被配置为基于与虚拟对象相关联的参考点的参考模型坐标和参考场景坐标，确定虚拟对象场景坐标。

在一些实施例中，确定虚拟对象场景坐标，包括：获取虚拟对象参考模型坐标和虚拟对象参考场景坐标在各个坐标方向上的偏移值；利用虚拟对象模型坐标各个方向上的坐标值与相应的偏移值的组合结果，确定虚拟对象场景坐标。

在一些实施例中，确定虚拟对象场景坐标，包括：获取由虚拟对象模型坐标指向虚拟对象参考模型坐标的方向向量；如果虚拟对象模型坐标系与虚拟对象场景坐标系的单位距离相同，基于虚拟对象参考场景坐标和虚拟对象方向向量，确定虚拟对象场景坐标。

在一些实施例中，装置300还包括：修正模块，被配置为如果虚拟对象模型坐标系与虚拟对象场景坐标系的单位距离不同相同，基于虚拟对象模型坐标系的单位距离与虚拟对象场景坐标系的单位距离的差异，修正虚拟对象方向向量，并基于虚拟对象参考场景坐标和修正后的方向向量，确定虚拟对象场景坐标。

在一些实施例中，参考点包括用于定位虚拟对象在虚拟场景中添加位置的首点。

在一些实施例中，装置300还包括：第二控制模块，被配置为如果虚拟对象控制指令通过指示虚拟对象控制点在虚拟对象场景坐标系下的场景坐标的变化来控制虚拟对象控制点的动作，利用虚拟对象控制指令控制虚拟对象控制点，以执行虚拟对象动作。

在一些实施例中，虚拟对象包括VRM虚拟模型形象。

在一些实施例中，虚拟对象模型坐标和虚拟对象场景坐标被利用四元数形式表示。

基于同一发明构思，本申请实施例中还提供了一种电子设备，电子设备对应的方法可以是前述实施例中的控制虚拟对象的方法，并且其解决问题的原理与该方法相似。本申请实施例提供的电子设备包括：至少一个处理器；以及与至少一个处理器通信连接的存储器；其中，存储器存储有可被至少一个处理器执行的指令，指令被至少一个处理器执行，以使至少一个处理器能够执行前述本申请的多个实施例的方法和/或技术方案。

电子设备可以是用户设备、或者用户设备与网络设备通过网络相集成所构成的设备，或者也可以是运行于上述设备的应用程序，用户设备包括但不限于计算机、手机、平板电脑、智能手表、手环等各类终端设备，网络设备包括但不限于如网络主机、单个网络服务器、多个网络服务器集或基于云计算的计算机集合等实现，可以用于实现设置闹钟时的部分处理功能。在此，云由基于云计算（Cloud Computing）的大量主机或网络服务器构成，其中，云计算是分布式计算的一种，由一群松散耦合的计算机集组成的一个虚拟计算机。

图4示出了适用于实现本申请实施例中的方法和/或技术方案的一种电子设备的结构，该电子设备400包括中央处理单元（CPU，Central Processing Unit）401，其可以根据存储在只读存储器（ROM，Read Only Memory）402中的程序或者从存储部分408加载到随机访问存储器（RAM，Random Access Memory）403中的程序而执行各种适当的动作和处理。在RAM 403中，还存储有系统操作所需的各种程序和数据。CPU 401、ROM 402以及RAM 403通过总线404彼此相连。输入/输出（I/O，Input / Output）接口405也连接至总线404。

以下部件连接至I/O接口405：包括键盘、鼠标、触摸屏、麦克风、红外传感器等的输入部分406；包括诸如阴极射线管（CRT，Cathode Ray Tube）、液晶显示器（LCD，Liquid Crystal Display）、LED显示器、OLED显示器等以及扬声器等的输出部分407；包括硬盘、光盘、磁盘、半导体存储器等一个或多个计算机可读介质的存储部分408；以及包括诸如LAN（局域网，Local Area Network）卡、调制解调器等的网络接口卡的通信部分409。通信部分409经由诸如因特网的网络执行通信处理。

特别地，本申请实施例中的方法和/或实施例可以被实现为计算机软件程序。例如，本申请公开的实施例包括一种计算机程序产品，其包括承载在计算机可读介质上的计算机程序，该计算机程序包含用于执行流程图所示的方法的程序代码。在该计算机程序被中央处理单元（CPU）401执行时，执行本申请的方法中限定的上述功能。

本申请另一实施例还提供了一种计算机可读存储介质，其上存储有计算机程序指令，计算机程序指令可被处理器执行以实现前述本申请的任意一个或多个实施例的方法和/或技术方案。

具体来说，本实施例可以采用一个或多个计算机可读介质的任意组合。计算机可读介质可以是计算机可读信号介质或者计算机可读存储介质。计算机可读存储介质例如可以是包括但不限于电、磁、光、电磁、红外线、或半导体的系统、装置或器件，或者任意以上的组合。计算机可读存储介质的更具体的例子（非穷举的列表）包括：具有一个或多个导线的电连接、便携式计算机磁盘、硬盘、随机存取存储器（RAM）、只读存储器(ROM)、可擦式可编程只读存储器（EPROM或闪存）、光纤、便携式紧凑磁盘只读存储器（CD-ROM）、光存储器件、磁存储器件、或者上述的任意合适的组合。在本文件中，计算机可读存储介质可以是任何包含或存储程序的有形介质，该程序可以被指令执行系统、装置或者器件使用或者与其结合使用。

计算机可读信号介质可以包括在基带中或者作为载波一部分传播的数据信号，其中承载了计算机可读的程序代码。这种传播的数据信号可以采用多种形式，包括但不限于电磁信号、光信号或上述的任意合适的组合。计算机可读的信号介质还可以是计算机可读存储介质以外的任何计算机可读介质，该计算机可读介质可以发送、传播或者传输用于由指令执行系统、装置或者器件使用或者与其结合使用的程序。

计算机可读介质上包含的程序代码可以用任何适当的介质传输，包括但不限于无线、电线、光缆、RF等等，或者上述的任意合适的组合。

可以以一种或多种程序设计语言或其组合来编写用于执行本申请操作的计算机程序代码，程序设计语言包括面向对象的程序设计语言—诸如Java、Smalltalk、C++，还包括常规的过程式程序设计语言—诸如“C”语言或类似的程序设计语言。程序代码可以完全地在用户计算机上执行、部分地在用户计算机上执行、作为一个独立的软件包执行、部分在用户计算机上部分在远程计算机上执行、或者完全在远程计算机或服务器上执行。在涉及远程计算机的情形中，远程计算机可以通过任意种类的网络，包括局域网（LAN）或广域网（WAN）连接到用户计算机，或者，可以连接到外部计算机（例如利用因特网服务提供商来通过因特网连接）。

附图中的流程图或框图示出了按照本申请各种实施例的设备、方法和计算机程序产品的可能实现的体系架构、功能和操作。在这点上，流程图或框图中的每个方框可以代表一个模块、程序段、或代码的一部分，该模块、程序段、或代码的一部分包含一个或多个用于实现规定的逻辑功能的可执行指令。也应当注意，在有些作为替换的实现中，方框中所标注的功能也可以以不同于附图中所标注的顺序发生。例如，两个接连地表示的方框实际上可以基本并行地执行，它们有时也可以按相反的顺序执行，这依所涉及的功能而定。也要注意的是，框图和/或流程图中的每个方框、以及框图和/或流程图中的方框的组合，可以用执行规定的功能或操作的专用的针对硬件的系统来实现，或者可以用专用硬件与计算机指令的组合来实现。

所属领域的技术人员可以清楚地了解到，为描述的方便和简洁，上述描述的系统，装置和单元的具体工作过程，可以参考前述方法实施例中的对应过程，在此不再赘述。

在本申请所提供的几个实施例中，应该理解到，所揭露的系统，装置和方法，可以通过其它的方式实现。例如，以上所描述的装置实施例仅仅是示意性的，例如，单元的划分，仅仅为一种逻辑功能划分，实际实现时可以有另外的划分方式，例如多个单元或页面组件可以结合或者可以集成到另一个系统，或一些特征可以忽略，或不执行。另一点，所显示或讨论的相互之间的耦合或直接耦合或通信连接可以是通过一些接口，装置或单元的间接耦合或通信连接，可以是电性，机械或其它的形式。

作为分离部件说明的单元可以是或者也可以不是物理上分开的，作为单元显示的部件可以是或者也可以不是物理单元，即可以位于一个地方，或者也可以分布到多个网络单元上。可以根据实际的需要选择其中的部分或者全部单元来实现本实施例方案的目的。

另外，在本申请各个实施例中的各功能单元可以集成在一个处理单元中，也可以是各个单元单独物理存在，也可以两个或两个以上单元集成在一个单元中。上述集成的单元既可以采用硬件的形式实现，也可以采用硬件加软件功能单元的形式实现。

上述以软件功能单元的形式实现的集成的单元，可以存储在一个计算机可读取存储介质中。上述软件功能单元存储在一个存储介质中，包括若干指令用以使得一个计算机设备（可以是个人计算机，服务器，或者网络设备等）或处理器（processor）执行本申请各个实施例方法的部分步骤。而前述的存储介质包括：U盘、移动硬盘、只读存储器（Read-Only Memory，ROM）、随机存取存储器（Random Access Memory，RAM）、磁碟或者光盘等各种可以存储程序代码的介质。

最后应说明的是：以上实施例仅用以说明本申请的技术方案，而非对其限制；尽管参照前述实施例对本申请进行了详细的说明，本领域的普通技术人员应当理解：其依然可以对前述各实施例所记载的技术方案进行修改，或者对其中部分技术特征进行等同替换；而这些修改或者替换，并不使相应技术方案的本质脱离本申请各实施例技术方案的精神和范围。

此外，显然“包括”一词不排除其他单元或步骤，单数不排除复数。装置权利要求中陈述的多个单元或装置也可以由一个单元或装置通过软件或者硬件来实现。第一，第二等词语用来表示名称，而并不表示任何特定的顺序。

1. 一种控制虚拟对象的方法，包括：

接收针对虚拟场景中虚拟对象的控制点的控制指令；

如果所述控制指令通过指示所述控制点在模型坐标系下的模型坐标的变化来控制所述控制点的动作，获取所述控制点在虚拟场景的场景坐标系下的场景坐标；

基于所述场景坐标更新所述控制指令，生成更新后的控制指令，其中，所述更新后的控制指令通过指示所述控制点的所述场景坐标的变化来执行所述动作；以及

利用更新后的控制指令控制所述控制点，以执行所述动作。

2. 根据权利要求1所述的方法，还包括：

基于与所述虚拟对象相关联的参考点的参考模型坐标和参考场景坐标，确定所述场景坐标。

3. 根据权利要求2所述的方法，其中确定所述场景坐标，包括：

获取所述参考模型坐标和所述参考场景坐标在各个坐标方向上的偏移值；

利用所述模型坐标各个方向上的坐标值与相应的偏移值的组合结果，确定所述场景坐标。

4. 根据权利要求2所述的方法，其中确定所述场景坐标，包括：

获取由所述模型坐标指向所述参考模型坐标的方向向量；

如果所述模型坐标系与所述场景坐标系的单位距离相同，基于所述参考场景坐标和所述方向向量，确定所述场景坐标。

5. 根据权利要求4所述的方法，还包括：

如果所述模型坐标系与所述场景坐标系的单位距离不同相同，基于所述模型坐标系的单位距离与所述场景坐标系的单位距离的差异，修正所述方向向量，并基于所述参考场景坐标和修正后的方向向量，确定所述场景坐标。

6. 根据权利要求2所述的方法，其中所述参考点包括用于定位所述虚拟对象在所述虚拟场景中添加位置的首点。

7. 根据权利要求1所述的方法，还包括：

如果所述控制指令通过指示所述控制点在所述场景坐标系下的场景坐标的变化来控制所述控制点的动作，利用所述控制指令控制所述控制点，以执行所述动作。

8. 根据权利要求1所述的方法，其中所述虚拟对象包括VRM虚拟模型形象。

9. 根据权利要求1-8中任一项所述的方法，其中所述模型坐标和所述场景坐标被利用四元数形式表示。

10. 一种用于控制虚拟对象的装置，包括：

接收模块，被配置为接收针对虚拟场景中虚拟对象的控制点的控制指令；

获取模块，被配置为如果所述控制指令通过指示所述控制点在模型坐标系下的模型坐标的变化来控制所述控制点的动作，获取所述控制点在虚拟场景的场景坐标系下的场景坐标；

更新模块，被配置为基于所述场景坐标更新所述控制指令，生成更新后的控制指令，其中，所述更新后的控制指令通过指示所述控制点的所述场景坐标的变化来执行所述动作；以及

第一控制模块，被配置为利用更新后的控制指令控制所述控制点，以执行所述动作。

11. 一种电子设备，所述电子设备包括：

至少一个处理器；以及

与所述至少一个处理器通信连接的存储器；其中，

所述存储器存储有可被所述至少一个处理器执行的指令，所述指令被所述至少一个处理器执行，以使所述至少一个处理器能够执行权利要求1至9中任一项所述的方法。

12. 一种计算机可读介质，其上存储有计算机程序指令，所述计算机程序指令可被处理器执行以实现如权利要求1至9中任一项所述的方法。

本申请提供了一种控制虚拟对象的方法、装置、电子设备及计算机可读介质，该申请接收针对虚拟场景中虚拟对象的控制点的控制指令；如果控制指令通过指示控制点在模型坐标系下的模型坐标的变化来控制控制点的动作，获取控制点在虚拟场景的场景坐标系下的场景坐标；基于场景坐标更新控制指令，生成更新后的控制指令，其中，更新后的控制指令通过指示控制点的场景坐标的变化来执行动作；以及利用更新后的控制指令控制控制点，以执行动作。由此，通过将其他坐标系下生成的控制指令统一至场景坐标系下坐标的方式，使得控制指令可以实际、准确地被执行，能够避免因坐标系差异导致的控制错误、虚拟对象扭曲，能够提升对于虚拟对象的控制质量。



图1



图2



图3



图4

