

(12) 按照专利合作条约所公布的国际申请

(19) 世界知识产权组织
国际局

(43) 国际公布日
2020年5月22日 (22.05.2020)



(10) 国际公布号
WO 2020/098531 A1

- (51) 国际专利分类号:
G06T 15/20 (2011.01)
- (21) 国际申请号: PCT/CN2019/115611
- (22) 国际申请日: 2019年11月5日 (05.11.2019)
- (25) 申请语言: 中文
- (26) 公布语言: 中文
- (30) 优先权:
201811361060.0 2018年11月15日 (15.11.2018) CN
- (71) 申请人: 腾讯科技(深圳)有限公司 (TENCENT TECHNOLOGY (SHENZHEN) COMPANY LIMITED) [CN/CN]; 中国广东省深圳市南山区高新区科技中一路腾讯大厦35层, Guangdong 518057 (CN)。
- (72) 发明人: 魏知晓 (WEI, Zhixiao); 中国广东省深圳市南山区高新区科技中一路腾讯大厦35层, Guangdong 518057 (CN)。
- (74) 代理人: 深圳市深佳知识产权代理事务所(普通合伙) (SHENPAT INTELLECTUAL PROPERTY AGENCY); 中国广东省深圳市罗湖区南湖街道春风路庐山大厦B座18C2、18D、18E、18E2, Guangdong 518001 (CN)。
- (81) 指定国(除另有指明, 要求每一种可提供的国家保护): AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DJ, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, JO, JP, KE, KG, KH, KN, KP, KR, KW, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX,

(54) Title: OBJECT LOADING METHOD, DEVICE, STORAGE MEDIUM, AND ELECTRONIC DEVICE

(54) 发明名称: 对象的加载方法和装置、存储介质、电子装置

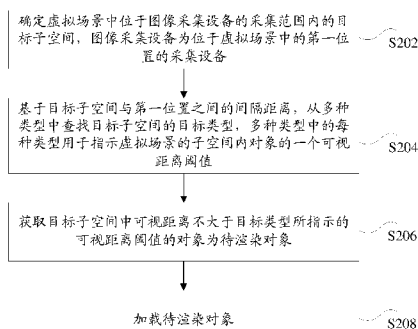


图 2

- S202 Determine, in a virtual scene, a target sub-space within an acquisition range of an image acquisition apparatus, the image acquisition apparatus being an acquisition apparatus located at a first position in the virtual scene
- S204 Search, on the basis of a spacing distance between the target sub-space and the first position, multiple types for a target type of the target sub-space, each type in the multiple types indicating one visible distance threshold of an object in the sub-space of the virtual scene
- S206 Acquire an object in the target sub-space that has a visible distance not greater than the visible distance threshold indicated by the target type as an object to be rendered
- S208 Load the object to be rendered

(57) Abstract: The present application discloses an object loading method, a device, a storage medium, and an electronic device. The method comprises: determining, in a virtual scene, a visible space within an acquisition range of an image acquisition apparatus, the image acquisition apparatus being an acquisition apparatus located at a first position in the virtual scene; determining, on the basis of the first position and in the visible space, a target sub-space located within a visible distance threshold indicated by a target type among multiple types, each type in the multiple types indicating one visible distance threshold for an object in the sub-space of the virtual scene; acquiring an object in the target sub-space that has a visible distance not greater than the visible distance threshold indicated by the target type as an object to be rendered; and loading the object to be rendered into a memory resource of a user terminal having a target application installed, the user terminal being used to render an image of the virtual scene. The present application solves the technical problem in the related art in which rendering objects in a virtual scene occupies too many hardware resources.

(57) 摘要: 本申请公开了一种对象的加载方法和装置、存储介质、电子装置。其中, 该方法包括: 确定虚拟场景中位于图像采集设备的采集范围内的可视空间, 图像采集设备为位于虚拟场景中的第一位置的采集设备; 基于第一位置确定可视空间内位于多种类型中目标类型所指示的可视距离阈值内的目标子空间, 多种类型中的每种类型用于指示虚拟场景的子空间内对象的一个可视距离阈值; 获取目标子空间中可视距离不大于目标类型所指示的可视距离阈值的对象为待渲染对象; 在安装目标应用的用户终端的存储资源中加载待渲染对象, 用户终端用于渲染出虚拟场景的图像。本申请解决了相关技术中渲染虚拟场景中的对象所占用的硬件资源较多的技术问题。

MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL,
PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL,
SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG,
US, UZ, VC, VN, ZA, ZM, ZW。

- (84) 指定国(除另有指明, 要求每一种可提供的地区
保护): ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ,
NA, RW, SD, SL, ST, SZ, TZ, UG, ZM, ZW), 欧亚 (AM,
AZ, BY, KG, KZ, RU, TJ, TM), 欧洲 (AL, AT, BE, BG,
CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU,
IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT,
RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI,
CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG)。

本国际公布:

- 包括国际检索报告(条约第21条(3))。

对象的加载方法和装置、存储介质、电子装置

本申请要求于 2018 年 11 月 15 日提交中国专利局、申请号为 201811361060.0、申请名称为“对象的加载方法和装置、存储介质、电子装置”的中国专利申请的优先权，其全部内容通过引用结合在本申请中。

5

技术领域

本申请涉及互联网技术领域，具体涉及一种对象加载技术。

背景技术

10 三维立体（3 Dimensions，3D）是空间的概念，由 X、Y、Z 三个轴组成的空间，是相对于只有长和宽的二维（2Dimensions，2D）平面而言。

3D 虚拟场景的搭建是 3D 的重要应用之一，例如，在大型游戏中，往往配以大场景，场景内诸如人、动物、房屋、树木、石头等单位数目众多，游戏过程中，游戏画面每更新一帧，就要对场景内的对象进行一次逐一处理，包括
15 渲染和动画等。

发明内容

本申请实施例提供了一种对象的加载方法和装置、存储介质、电子装置，以至少解决相关技术中渲染虚拟场景中的对象所占用的硬件资源较多的技术问题。

20 根据本申请实施例的一个方面，提供了一种对象的加载方法，包括：确定虚拟场景中位于图像采集设备的采集范围内的可视空间，其中，图像采集设备为位于虚拟场景中的第一位置的采集设备；基于第一位置确定可视空间内位于多种类型中目标类型所指示的可视距离阈值内的目标子空间，其中，多种类型中的每种类型用于指示虚拟场景的子空间内对象的一个可视距离阈值；获取目
25 标子空间中可视距离不大于目标类型所指示的可视距离阈值的对象为待渲染对象；加载待渲染对象。

-2-

根据本申请实施例的另一方面，还提供了一种对象的加载装置，包括：第一确定单元，用于确定虚拟场景中位于图像采集设备的采集范围内的可视空间，其中，图像采集设备为位于虚拟场景中的第一位置的采集设备；第二确定单元，用于基于第一位置确定可视空间内位于多种类型中目标类型所指示的可视距离阈值内的目标子空间，其中，多种类型中的每种类型用于指示虚拟场景的子空间内对象的一个可视距离阈值；获取单元，用于获取目标子空间中可视距离不大于目标类型所指示的可视距离阈值的对象为待渲染对象；加载单元，用于加载待渲染对象。

根据本申请实施例的另一方面，还提供了一种存储介质，该存储介质包括存储的程序，程序运行时执行上述的方法。

根据本申请实施例的另一方面，还提供了一种电子装置，包括存储器、处理器及存储在存储器上并可在处理器上运行的计算机程序，处理器通过计算机程序执行上述的方法。

根据本申请实施例的另一方面，还提供了一种计算机程序产品，包括指令，当其在计算机上运行时，使得所述计算机执行上述的方法。

在本申请实施例中，确定虚拟场景中位于图像采集设备的采集范围内的可视空间，获取目标子空间中可视距离不大于目标类型所指示的可视距离阈值的对象为待渲染对象；加载待渲染对象，对于位于不同位置的可视空间，仅仅加载对于用户而言实际可见的对象，可以解决相关技术中渲染虚拟场景中的对象所占用的硬件资源较多的技术问题，可以显著降低对内存的资源占用，还能够降低对如中央处理器（Central Processing Unit, CPU）和图形处理器（Graphics Processing Unit, GPU）等硬件资源的消耗。

附图说明

为了更清楚地说明本申请实施例或现有技术中的技术方案，下面将对实施例或现有技术描述中所需要使用的附图作简单地介绍，显而易见地，下面描述中的附图仅仅是本申请的实施例，对于本领域普通技术人员来讲，在不付出创造性劳动的前提下，还可以根据提供的附图获得其他的附图。

- 图 1 是根据本申请实施例的对象的加载方法的硬件环境的示意图；
图 2 是根据本申请实施例的一种可选的对象的加载方法的流程图；
图 3 是根据本申请实施例的一种可选的场景中网格区块的示意图；
图 4 是根据本申请实施例的一种可选的场景中网格区块的示意图；
5 图 5 是根据本申请实施例的一种可选的场景中网格区块的示意图；
图 6 是根据本申请实施例的一种可选的场景中网格区块的示意图；
图 7 是根据本申请实施例的一种可选的场景中坐标系的示意图；
图 8 是根据本申请实施例的一种可选的场景的预处理的流程图；
图 9 是根据本申请实施例的一种可选的场景中包围盒的示意图；
10 图 10 是根据本申请实施例的一种可选的场景中网格的示意图；
图 11 是根据本申请实施例的一种可选的场景的示意图；
图 12 是根据本申请实施例的一种可选的场景的示意图；
图 13 是根据本申请实施例的一种可选的场景的示意图；
图 14 是根据本申请实施例的一种可选的场景的对象加载的流程图；
15 图 15 是根据本申请实施例的一种可选的对象的加载装置的示意图； 以及
图 16 是根据本申请实施例的一种终端的结构框图。

具体实施方式

为了使本技术领域的人员更好地理解本申请方案，下面将结合本申请实施
20 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
描述的实施例仅仅是本申请一部分的实施例，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做出创造性劳动前提下所获得的所
有其他实施例，都应当属于本申请保护的范围。

需要说明的是，本申请的说明书和权利要求书及上述附图中的术语“第
25 一”、“第二”等是用于区别类似的对象，而不必用于描述特定的顺序或先后次
序。应该理解这样使用的数据在适当情况下可以互换，以便这里描述的本申请
的实施例能够以除了在这里图示或描述的那些以外的顺序实施。此外，术语“包
括”和“具有”以及他们的任何变形，意图在于覆盖不排他的包含，例如，包含

了一系列步骤或单元的过程、方法、系统、产品或设备不必限于清楚地列出的那些步骤或单元，而是可包括没有清楚地列出的或对于这些过程、方法、产品或设备固有的其它步骤或单元。

5 本申请主要用于解决大型 3D 场景在性能有限的设备上的动态加载和连续展现的问题，例如在计算机游戏中为用户展现虚拟的超大型 3D 世界，通常一个 3D 场景里有大量点、面、图片数据，而大型 3D 场景模拟至少几平方公里以上的虚拟世界，更是无法全部载入到性能有限的设备上被处理。

10 为了克服上述问题，根据本申请实施例的一方面，提供了一种对象的加载方法的方法实施例，利用本申请的技术方案，处理过程中在可视范围内的场景内容会被加载到，且保证了最小化性能开销，不应该被看见的场景内容被卸载出内存等硬件资源，场景内容的连贯出现，即不出现因可视物件闪现引起的视觉卡顿。

15 本申请实施例提供的对象的加载方法可以应用于图像处理设备，如终端设备、服务器。其中，该方法可以应用到终端设备中，终端设备例如可以是智能终端、计算机、个人数字助理（Personal Digital Assistant，简称 PDA）、平板电脑等设备。该方法还可以应用到服务器中，该服务器可以是独立的服务器，也可以是集群中的服务器。

20 举例来说，在本实施例中，上述对象的加载方法可以应用于如图 1 所示的由服务器 101 和/或用户终端 103 所构成的硬件环境中。如图 1 所示，服务器 101 通过网络与用户终端 103 进行连接，可用于为用户终端或用户终端上安装的客户端提供服务（如游戏服务、应用服务等），可在服务器上或独立于服务器设置数据库 105，用于为服务器 101 提供数据存储和数据读取服务，上述网络包括但不限于：广域网、城域网或局域网，用户终端 103 并不限定于 PC、手机、平板电脑等。

25 本申请实施例的对象的加载方法可以由用户终端 103 来执行，用户终端 103 执行本申请实施例的对象的加载方法也可以是由安装在其上的客户端来执行。本申请的应用场景可以在各种计算机设备（包括但不限于 PC 电脑、移

动手机、平板电脑、嵌入式设备、服务器、游戏主机等设备)中使用目标应用这一软件渲染出的 3D 大场景(如大型 3D 游戏、3D 动画、VR 展示中的场景渲染),此处的渲染可以理解为计算机处理数字化的美术内容并将其绘制在显示设备上的过程。本申请的技术方案适用的场景包括但不限于如下场景:

5 1) 在采用增强现实(Augmented Reality, AR)、虚拟现实(Virtual Reality, VR)等的社交应用中,在用户登陆后,若在其视野范围内出现大量对象(如场景中的山水、建筑,其它用户,用户宠物等对象等),对大量对象进行图像渲染需要长时间的占用大量性能资源(包括对象信息在内存中的加载、中央处理器 CPU 和图像处理器 GPU 的渲染等操作),此时则可以采用本申请的技术方
10 案来对大量对象进行筛选,筛选出对于当前用户而言可见的对象,并加载和渲染这些可见的对象,从而达到降低图像渲染占用的硬件资源(如 CPU、内存、GPU)的目的。

 2) 在游戏应用中,当玩家处于游戏的过程中,可能在其可视范围内突然出现大量其它玩家角色、非玩家控制(Non Player Character, NPC)角色、山
15 水建筑等对象,受限于终端的性能资源,此时终端的性能资源会大量地被渲染操作所占用,从而造成终端的可用性能资源大量下降,在用户侧的感观为终端卡顿、响应缓慢、掉帧等,此时则可以采用本申请的技术方案,对大量对象进行筛选,筛选出对于当前用户而言可见的对象,并加载和渲染这些可见的对象,从而达到降低图像渲染占用的硬件资源的目的。

20 3) 在军事战争类、医学等仿真应用中,以军事战争仿真应用为例,当可视范围内存在大量军事人员、军事设备、军事场景等对象时,终端可对大量对象进行筛选,筛选出对于当前用户而言可见的对象,并加载和渲染这些可见的对象,从而降低了图像渲染所占用的硬件资源。

 上述实施例的场景通常有这样的特性,需要渲染大规模的 3D 场景(通常
25 超过几平方公里),场景包含的物件对象众多(数量通常在几万以上),细节复杂,甚至有些产品中需要模拟真实的看上去无边际的超大世界,事实上设备的内存和处理器性能有限,尤其是在移动设备上内存和处理器性能更加有限,无

法将整个场景全部载入并处理,需要动态的载入可视的场景区域和卸载不可视的场景区域,来达到渲染目的。本申请的方案正是在此类产品中解决动态加卸载场景区域的问题,并能够使用一种优化的方案保证动态加卸载的过程中,尽量最小化的内存占用,并能够平滑的展现新进入视野的物体。

5 接下来以用户终端作为图像处理设备为例,对本申请实施例提供的对象的加载方法进行说明。图2是根据本申请实施例的一种可选的对象的加载方法的流程图,如图2所示,该方法可以包括以下步骤:

S202,用户终端确定虚拟场景中位于图像采集设备的采集范围内的可视空间,图像采集设备为位于虚拟场景中的第一位置的采集设备。

10 上述虚拟场景可以是由目标应用提供的虚拟场景,其中,上述的目标应用可为采用AR、VR的社交应用、游戏应用、军事战争类、医学类仿真应用等应用;该目标应用安装在用户终端上后可形成客户端,虚拟场景可以为社交场景、游戏场景、仿真场景等,这些场景可以通过用户终端上的客户端呈现。

15 上述的图像采集设备可为虚拟的图像采集设备,用于模拟第一人称视角、第三人称视角(视角也可称为视场角,是用来量化当前观察场景的位置的用户这一观察者的视野的角度)等进行图像采集,相当于图像采集设备的可视范围即用户的可视范围,该图像采集设备具体可以为在渲染引擎中调用的功能组件,如游戏引擎Unity中的相机Camera。

20 上述虚拟场景为3D场景,即依靠计算机设备(如前述服务器或者用户终端)模拟计算并绘制到显示设备上的虚拟场景内容,可以按照某个间隔单位(如间隔1米、10米等)将虚拟场景划分为若干个子空间(或称为网格),上述的可视空间即为位于图像采集设备的采集范围内的子空间的集合(可以包括至少一个子空间),例如,给定图像采集设备可采集距离,即可按确定距离在该可采集距离内的子空间为可视空间。

25 S204,用户终端基于第一位置确定可视空间内位于多种类型中目标类型所指示的可视距离阈值内的目标子空间,多种类型中的每种类型用于指示虚拟场

景的子空间内对象的一个可视距离阈值。

S206, 用户终端获取目标子空间中可视距离不大于目标类型所指示的可视距离阈值的对象为待渲染对象。

5 可选地, 对于虚拟场景中的对象(如场景对象山水建筑、其他用户、宠物等), 有一定的可视距离(即能够看见该对象的最远距离), 换言之, 超过了该可视距离, 那么就对于用户不可见, 上述的多种类型中的每种类型对应于一个可视距离阈值, 即将虚拟场景的子空间按照可视距离进行分类, 对于相同类型的子空间, 可以采用相同的方式过滤掉可视距离大于对应的可视距离阈值的对象, 而获取其中可视距离小于等于可视距离阈值的对象, 即目标子空间内的对象, 也即仅仅获取对于用户而言实际可见的对象。

10 S208, 加载待渲染对象, 如用户终端可以在安装有目标应用的用户终端的存储资源中加载待渲染对象。其中, 用户终端用于渲染出虚拟场景的图像, 此处的渲染是指计算机设备处理数字化的美术内容并将其绘制在显示设备上的过程。

15 相较于相关技术中, 加载整个场景中的对象, 或者加载用户的可视范围内的所有对象, 本申请的技术方案是仅仅加载对于用户而言可见的对象, 可以显著降低对内存的需求(即加载对象所需的内存会更小), 与此同时, CPU 和 GPU 所处理的对象也更少, 还能够降低对 CPU 和 GPU 硬件资源的消耗。

20 上述实施例以本申请实施例的对象的加载方法由用户终端 103 来执行为例进行说明, 本申请实施例的对象的加载方法也可以由服务器 101 来执行, 该实施例与前述实施例的区别在于执行主体由用户终端替换为了服务器, 本申请实施例的对象的加载方法还可以是由服务器 101 和终端 103 共同执行。

25 通过上述步骤, 确定目标应用提供的虚拟场景中位于图像采集设备的采集范围内的可视空间, 基于第一位置确定可视空间内位于多种类型中目标类型所指示的可视距离阈值内的目标子空间, 获取目标子空间中可视距离不大于目标类型所指示的可视距离阈值的对象为待渲染对象; 在安装有目标应用的用户终

端的存储资源中加载待渲染对象，由于所加载的是对用户而言实际可见的对象，可以解决相关技术中渲染虚拟场景中的对象所占用的硬件资源较多的技术问题，可以显著降低对内存的资源占用，还能够降低对 CPU 和 GPU 硬件资源的消耗。

5 本申请通过自动化地把大型 3D 场景（即前述虚拟场景）处理成一种多级（每一级可以认为是一种类型，多级即前述的多种类型）的细节网格结构，该网格的特点是每一层次的网格密度和大小不一，物件根据其可视距离情况被放置在合适的一级，加载中在每一个细节层对观察位置周围网格确定可视性后动态地加载卸载网格中的物件。下面结合图 2 所示的步骤，以将本申请的技术方案应用于游戏场景为例详述本申请的技术方案：

10

在 S202 提供的技术方案中，在用户进入游戏应用后，用户终端确定目标应用提供的虚拟场景中位于图像采集设备的采集范围内的（或称待呈现给用户的）可视空间，该图像采集设备为位于虚拟场景中的第一位置用于模拟用户视野的采集设备。

15 为了在整个虚拟场景中分离出位于用户的可视范围内的可视空间，本申请的技术方案可对场景物件的可视情况预先进行量化并组织在一种空间数据结构中，该结构保证了越细节的场景物体被加载的范围越小，大场景中同时被加载卸载处理的空间区域始终在一定数量下，大部分物件都不会被纳入到可视化判断和加载处理，在保证可视范围内的物体一定被准确加载的同时，尽可能地最小化载入到设备的资源。下面结合步骤 1-步骤 2 进行详述：

20

步骤 1，用户终端在确定目标应用提供的虚拟场景中位于图像采集设备的采集范围内的可视空间之前，确定多种类型中的每种类型所对应的网格密度，换言之，就是多种类型中的每种类型用于表示一个网格密度（即一种虚拟场景的空间划分粒度），那么有多少种网格密度，那么就存在多少种类型。

25 可选地，用户终端在确定多种类型中的每种类型所对应的网格密度时，可以通过如下所示的步骤 11-步骤 12 实现：

步骤 11, 用户终端获取虚拟场景中所有对象的可视距离中的最大可视距离和最小可视距离。

用户终端在获取虚拟场景中所有对象的可视距离中的最大可视距离和最小可视距离时, 可以按照如下方式获取虚拟场景中每个对象的可视距离: 预先
5 确定对象的最小可视屏占比 S_{min} (即预设参数), 运算屏占比阈值 (或称可视屏占比 S_{min}) 与图像采集设备的目标视角的正切值的 2 倍之间的乘积, 记为第二乘积, 并计算每个对象在三维坐标系中三个维度上的坐标的最大值 (即在 X 轴、Y 轴和 Z 轴上的坐标的最大值) 与预设参数 (预设参数可以认为是一个预先设定的为正数的参数, 如 1.1) 之间的乘积, 记为第一乘积, 然后计算每
10 个对象的第一乘积与第二乘积之间的比值, 该比值即每个对象的可视距离; 在按照上述方式计算完每个对象的可视距离后, 即可从虚拟场景中所有对象的可视距离查找最大可视距离和最小可视距离。

在上述实施例中, 在物体的屏占比时, 可假设物体最终渲染到屏幕上所占有的区域的长和宽的像素是 h 和 w , 设当前整个屏幕的长和宽的像素分别为
15 Sh , Sw , 取 h/Sh 和 w/Sw 的最小值为物体当前的屏占比, 最小可视屏占比即为用户可视化的最小屏占比。

步骤 12, 用户终端按照最大可视距离和最小可视距离确定多种类型中的每种类型所对应的网格密度。

用户终端求取所有对象的可视距离中的最大可视距离 D_{max} 和最小可视
20 距离 D_{min} 的目的在于, 将最大可视距离和最小可视距离之间的距离进行等分处理, 可选地, 此处的等分数量可以是预先确定分成 m 份, 即份数的数值 m 是预先确定好的, 第 i 种 (i 的取值大于等于 1 且小于等于 $m+1$) 类型对应的可视距离即 $D_{min+i} * (D_{max}-D_{min}) / m$ 。

可选地, 此处的等分数量也可根据最大可视距离 D_{max} 和最小可视距离
25 D_{min} 确定, 例如, 在按照最大可视距离和最小可视距离确定多种类型中的每种类型所对应的网格密度时, 可对最大可视距离与最小可视距离之间的比值向

下取整，得到第一阈值（即等分数量，此处可以用 n 表示），该第一阈值也可用于表示多种类型的类型数量；然后将（最大可视距离 D_{max} -最小可视距离 D_{min} ）/最小可视距离 D_{min} 作为目标距离；将多种类型中第 i 个类型的所对应的网格密度所表示的网格间距设置为目标间距的 i 倍， i 为大于 0 且小于等于 5 第一阈值的整数。

本申请中，用户终端可采用不同的方式对多层网格进行划分，例如采用多层网格中每一层的加载距离定义为一个 $D_{sequence} = \{D_1, D_1 * 2, D_1 * 3, \dots, D_1 * n\}$ 序列，当然，也可定义其他类型的距离序列，虽然可能加载卸载效果有差别，应该认为不管多层网格距离间的关系序列是怎样，所有设计成多种层次加载距 10 离网格，且这些距离间存在某种关系或者无关系的方案都同本方案类似。

可选地，用户终端可采用其他方式来衡量物件的加载距离，而不一定是 $D = 1.1 * \max(B_x, B_y, B_z) / (2 * S_{min} * \tan(fov / 2))$ ，应该认为所有基于为物件衡量加载距离，然后基于加载距离在多层细节网格上加载的方案都同本方案类似。

步骤 2，用户终端按照多种类型中每种类型所对应的网格密度为多种类型 15 中每种类型创建配置信息，并将多种类型中每种类型的配置信息保存至配置文件，配置信息中记录有按照相应的网格密度进行划分的子空间内的对象和子空间所在的第二位置。

可选地，用户终端在按照前述内容确定每种类型对应的网格密度之后，在按照多种类型中每种类型所对应的网格密度为多种类型中每种类型创建配置 20 信息时，可以通过如下方式实现：

步骤 21，用户终端确定虚拟场景中按照多种类型中每种类型所对应的网格密度进行划分而得到的子空间（每个网格相当于一个子空间），也即对于每种类型，可以按照该类型所对应的网格密度来对整个虚拟场景进行划分，并对虚拟场景中的网格进行顺序编码（每个格子相当于三维空间中的一个单位）； 25 用户终端查找虚拟场景中每个子空间内可视距离与每种类型所指示的可视距离阈值匹配的对象，对第 i 种类型，可以将可视距离在 $[i * D_{min}, (i-1) * D_{min})$

内的对象作为与第*i*种类型所指示的可视距离阈值匹配的对象，同时可定位好该对象在虚拟场景中所在的网格，并在一条配置信息中保存每个子空间与查找到的对象之间的关联关系，对于多种类型中的每种类型，可以在一条配置信息中保存，每条配置信息中保存有相应网格密度下所有网格的内的对象。

- 5 在一个可选的实施例中，若某种类型下，所对应的网格密度下，不存在可视的对象，那么可删除该类型下的配置信息，以节省存储空间。

在 S204 提供的技术方案中，用户终端基于第一位置确定可视空间内位于多种类型中目标类型所指示的可视距离阈值内的目标子空间，多种类型中的每种类型用于指示虚拟场景的子空间内对象的一个可视距离阈值。

- 10 上述的目标类型为多种类型中的任意类型，换言之，对于每种类型，可以存在一个对应的目标子空间，即对第*i*种类型，可以将与第一位置之间的间距在 $i \cdot D_{min}$ 以内的子空间作为第*i*种类型对应的目标子空间。

可选地，用户终端在基于第一位置确定可视空间内位于多种类型中目标类型所指示的可视距离阈值内的目标子空间时，可以按照如下方式实现：

- 15 可选地，用户终端可获取可视空间的子空间与第一位置之间的间隔距离；将可视空间内与第一位置之间的间隔距离不大于目标类型所指示的可视距离阈值的子空间作为目标子空间。

在 S206 提供的技术方案中，用户终端获取目标子空间中可视距离不大于目标类型所指示的可视距离阈值的对象为待渲染对象。

- 20 在该实施例中，用户终端获取目标子空间中可视距离不大于目标类型所指示的可视距离阈值的对象为待渲染对象可以通过如下方式实现：用户终端获取配置文件，配置文件中记录虚拟场景按照以上所述的网格密度进行划分时的子空间（或称网格）内与多种类型中每种类型关联的对象，虚拟场景的子空间内的每个对象被设置为与多种类型中不大于所指示的可视距离阈值的类型关联；
25 获取配置文件中记录的目标子空间内与目标类型关联的对象为待渲染对象。

用户终端在根据配置文件中记录的可视空间内与目标类型关联的对象确定待渲染对象时,可以将可视空间内所有与目标类型关联的对象作为待渲染对象。

5 在 S208 提供的技术方案中,用户终端在安装有目标应用的用户终端的存储资源中加载待渲染对象,用户终端用于渲染出虚拟场景的图像,此处的渲染是指计算机设备处理数字化的美术内容并将其绘制在显示设备上的过程。

10 可选地,用户终端在安装有目标应用的用户终端的存储资源中加载待渲染对象时,在用户终端的存储资源当前保存的对象中包括待渲染对象的情况下,继续在用户终端的存储资源中保存待渲染对象,并删除用户终端的存储资源中除待渲染对象以外的对象,也即本次渲染与前一次渲染存在部分或者全部相同的对象的情况下,仅需要增加前一次没有的待渲染对象并删除前一次需要渲染而本次不需要渲染的对象;在用户终端的存储资源当前保存的对象中不包括待渲染对象的情况下,删除保存在用户终端的存储资源中的对象,并将待渲染对象
15 对象加载至用户终端的存储资源中,也即本次渲染与前一次渲染全部不相同的情况下,直接删除前一次的所有对象,然后增加本次待渲染对象。

20 可选地,用户终端在安装有目标应用的用户终端的存储资源中加载待渲染对象之后,将目标子空间的网格密度设置为与目标类型对应的网格密度匹配,并在设置网格密度后的目标子空间中渲染出待渲染对象,多种类型中的任一类型用于指示按照与任一类型对应的网格密度来渲染虚拟场景的子空间内可视距离不大于任一类型所指示的可视距离阈值的对象,多种类型中第一类型对应的网格密度大于多种类型中第二类型对应的网格密度,第一类型所指示的可视距离阈值小于第二类型所指示的可视距离阈值。

25 采用本申请的技术方案:用户终端量化场景里所有物件的可视距离并把它们组织到一种多级场景网格的数据结构上,以物件的可视距离为基本单元,程序运行时,对于每个层次的观察位置周围的固定数量网格列为可见性的考察范

围,判断物件的可见性并加载,由于细小的物件在尺度更小的场景网格中,因此越细节的物体判断可见性的范围也越小,加载进入的内存的量也越小,从而使得对系统存储、CPU 以及 GPU 的资源占用减小。

5 作为一种可选的实施例,下面以将本申请的技术方案应用于游戏的虚拟场景为例进一步详述本申请的技术方案。

在相关的进行 3D 大场景加载的技术方案中,如 Unity3D 引擎中,把大场景人工地划分成几个分隔开的单独的小场景来制作,简单按照空间位置划分,程序运行中动态地载入卸载这些小场景; UnrealEngine4 引擎中,人工地把大场景划分成几个子关卡,每个子关卡可以单独设置加载距离和场景中物件,子
10 关卡之间没有必然的空间关系,程序运行中遍历每个子关卡的加载距离确定其加载卸载。

在上述的相关技术方案中,采取全场景加载的方式,对于超大型场景载入的设备内存占用是瓶颈;在以 Unity3D 和 UnrealEngine4 为代表的主流商业游戏引擎中,这些方案虽然能够实现大场景中物件加载的准确性,但是方案难以
15 最小化场景对设备内存的占用,即都会出现过度加载的情况。

申请人认识到,理想的情况下,可以只用加载可视物件,这样既能保证加载的准确性,还能是的对于硬件资源的消耗最低,相关技术中方案都是以空间位置为基本单元进行加载或编辑,而在相同空间区域上也存在不同体积和细节程度的物体,在给定观察距离上,不同体积和细节的物体的可见性也是不同的,
20 把整个相关区域都加载进来会浪费不必要的内存。

若采用人工的对场景进行子场景划分,当场景达到一定规模之后,这种人工划分耗费的时间是不可承受的,并且也很难完全准确地将物件放置在最合适的网格内,本方案以物件的可视距离为依据,用算法自动地对场景做多级网格划分,更科学且节省人力。

25 本申请技术方案的核心是场景物件的加载距离是和物件的可视距离相关的,不以空间位置为基本单元加载,而是以物件的可视距离为基本单元,首先

量化场景里所有物件的可视距离并把它们组织到一种多级场景网格的数据结构上,程序运行时,对于每个层次的观察位置周围的固定数量网格列为可见性的考察范围,判断物件的可见性并加载,由于细小的物件在尺度更小的场景网格中,导致越细节的物体判断可见性的范围也越小,加载进入的内存的量也越

5 小,这也符合越细小的物体可视距离越近的规律。

本方案主要包括两个阶段的实现,第一个阶段是对场景的预处理,第二个阶段是在处理后的场景上进行动态的加载卸载部分场景。下面仍以用户终端作为图像处理设备为例进行说明。

用户终端预处理的过程可把场景自动化地处理成本申请所指的多级细节

10 网格结构,将场景处理成这种多级细节网格的方案是本申请的重要组成部分,它有利于场景在动态加载时根据物件的细节程度进行加载,最小化加载时设备占用的资源,场景被预处理的结果是场景中的每一个对象被划分到某个层次 L 的某个空间区块上,如图 3 至图 6 示出了不同层次网格区块内包含的场景物件,从图 3 至图 6 示出的网格尺度逐渐变小,可以看到网格尺度越小的网格容纳的

15 场景物件尺寸也越小,场景物件的加载距离也越近,尺度较大的网格层次内包含的都是更粗粒度的场景物件,尺度较小的网格层次内包含的都是更细节的场景物件。

在下面计算步骤的表达中要做如下几个公式定义:

坐标系的定义,本申请的方案所在的坐标系可为 3D 笛卡尔坐标系,坐标

20 系的三个轴 X 轴、Y 轴以及 Z 轴,如图 7 所示。

$\max(n_0, n_1, n_2, \dots)$, 表示求这个序列 n_0, n_1, n_2 等中的最大值。

$\tan(x)$, 表示求 x (单位为弧度)的正切值。

$\text{floor}(x)$, 表示所有不大于 x 的整数中最大的那个整数。

$\text{ceil}(x)$, 表示所有不小于 x 的整数中最小的那个整数。

25 $\text{abs}(x)$, 表示 x 的绝对值。

用户终端对场景的预处理阶段的处理流程如流程图 8 所示，左侧是用户终端对场景中的每个物件（或称对象）进行处理，右侧是用户终端对场景空间网格进行处理。

详细步骤描述如下：

- 5 S801，用户终端基于包围盒和最小屏幕显示比例（即最下屏幕占比）计算加载距离。

10 预处理的输入可以是美术设计人员使用各种场景编辑器编辑好的原始大场景中的场景资源对象，场景资源由很多场景物件组成，一个场景物件是场景中一个相对独立的渲染对象实体（例如石头、树木、建筑等），场景物件通常包含一个它所处于的场景的世界坐标位置 P ， P 可以使用一个三维坐标 (P_x, P_y, P_z) 表示，此外每个场景物件还包含一个它的包围盒 $Bounds$ $(x_{min}, y_{min}, z_{min}, length, height, width)$ ，它的最小可视屏占比 S_{min} ，通常在渲染中，可认为物体当前的屏占比小于 S_{min} ，该物件就不可见（例如离观察位置很远的一个物件，它的屏占比很小）。场景的规模很大，即场景中物件分布的区域很大，可能高于几公里，且场景的物件数量很多，通常高于几万个。
15 如图 9 是一个典型的输入场景，图 9 中的立方体表示的是这个建筑的包围盒。

20 物体的包围盒，即包括一个物体的体积最小的立方体称为它的包围盒。物体的包围盒可以表示成一个 6 元的数值 $Bounds$ $(x_{min}, y_{min}, z_{min}, length, height, width)$ ，其中 $x_{min}, y_{min}, z_{min}$ 分分别代表这个包围盒包含区域中在世界坐标轴 x 、 y 、 z 上的最小值， $length, height, width$ 分别代表这个包围盒包含区域在世界坐标轴 x 、 y 、 z 三个方向的跨度长度。

S802，生成物件的加载代理对象 Proxy。

25 用户终端遍历计算场景的所有物件，对于每个遍历到的物件，计算它的加载距离 D 。设当前观察位置的视场角为 fov （单位为弧度）， D 的计算方式定义为 $D = 1.1 * \max(B_x, B_y, B_z) / (2 * S_{min} * \tan(fov / 2))$ 。它的含义是当物件与观察位置的距离大于 D 的时候就可认为该物件不可视了， D 的计算和当前的观察位置，

物件的最小屏占比相关。

用户终端遍历计算场景的所有物件，对于每个遍历到的物件，生成一个加载代理对象 Proxy，Proxy 可包含如下括号中的几个元素(Path,P,R,S,D,B,id),其中 Path 是这个物件的资源路径，D 是加载距离，P 是场景中的位置，R 是旋转，S 是缩放，B 是包围盒，id 是这个场景物件的唯一标识。

S803，根据物件加载距离分布计算多层网格中最小和最大的网格大小 G_{min} ， G_{max} ，并在其间均匀产生一系列网格大小 $\{G_{min}, G_{min} * 2, G_{min} * 3, \dots, G_{max}\}$ 。

用户终端统计所有场景物件的加载距离，找出其中最小的加载距离 D_{min} 和最大的加载距离 D_{max} ，根据 D_{min} 和 D_{max} 生成一组序列 $D_{sequence} = \{D_1, D_1 * 2, D_1 * 3, \dots, D_1 * n\}$ ，其中， $n = \text{ceil}(D_{max} / D_{min})$ ， D_1 可以为 D_{min} ，计算整个场景的包围盒大小 B_{scene} ， B_{scene} 的大小是能包含场景所有物体的最小的立方体。 B_{scene} 的 6 元数值表示记为 $(B_{scene_x}, B_{scene_y}, B_{scene_z}, B_{scene_length}, B_{scene_height}, B_{scene_width})$ 。

S804，对每个层次，按照该层次的网格大小来均匀的划分整个场景空间，生成一系列场景网格。

用户终端基于 $D_{sequence}$ 生成一组网格层次，对于 $D_{sequence}$ 中的每一个元素 D_i (其中 $1 \leq i \leq n$)，代表第 i 层网格，遍历每一个层此作如下操作：在第 l 层上，将场景均匀分成如图 10 的网格结构，它由很多区块组成，其中在 x 轴方向第 i 列， z 轴方向第 k 行的区块表示为 $Block(l, i, k)$ ，每个 $Block$ 至少包含两个元素，它包含的场景物件代理对象 id 列表，和它的包围盒 $Bounds(l, i, k)$ ， $Bounds(l, i, k)$ 的 6 元数值表示是 $[(i+0.5) * D_l + B_{scene_x}, B_{scene_y}, (k+0.5) * D_l + B_{scene_z}, D_l, B_{scene_height}, D_l]$ 。

S805，根据加载距离计算适合容纳它的网格大小 G ，并把它归属在大小为 G 的那一层网格中。

用户终端遍历场景中的所有代理对象 Proxy，首先根据它的加载距离 D 选

择它所在的网格层次 L ， L 是 D_{sequece} 中大于或者等于 D 的所有元素中的最小的那一个。

S806，在大小为 G 的网格层上，找到所有与其有空间重叠的网格集合 S ，对集合的每个网格记录下 Proxy 的 Id。

5 然后，用户终端再遍历第 L 层上的所有区块，计算每个区块的包围盒 $\text{Bounds}(l,i,k)$ 和代理对象 Proxy 的包围盒 B 的重合度 S ，两个包围盒的重合度计算是计算他们重叠的空间区域。如果 $S > 0$ 说明这个代理对象相关的场景物件在这个区块 $\text{Block}(l,i,k)$ 的空间内，如果 $\text{Block}(l,i,k)$ 同 Proxy 的重合度 S 大于 0，则将 Proxy 的 id 记录在这个 $\text{Block}(l,i,k)$ 的 id 列表内。

10 S807，移除没有容纳任何场景物件的空间网格和层次。

用户终端遍历每一个层次网格的每一个空间区块 $\text{Bounds}(l,i,k)$ ，找到 id 列表内容为空的区块，将其删除。遍历每一个层次的网格，对其中空间区块数量为 0 的层次删除。

15 S808，将每个层次的网格及其包含的物件的 Proxy 信息序列化保存到当前场景中。

用户终端保存当前剩下的所有空间网格区块 $\text{Bounds}(l,i,k)$ 到场景文件中。

S809，清空当前场景的所有显示物体，保存为新的场景（即处理后的场景）。

用户终端保存所有代理对象 Proxy 信息到场景文件中，清空场景中的所有场景物件，保存成新的场景文件。

20 本申请的方案在运行时部分是指场景在被预先处理后可以在运行时动态的加载卸载，即根据当前观察者的位置（或摄像机位置）准确找到当前可视的物件进行加载。场景随观察观察位置的加载卸载过程可以参考图 11 至图 13 所示，显示了场景的一个固定区域，随着观察位置（摄像机）在场景中的移动，这个场景区域内加载的物件会发生变化，如图 11 至图 13 的变化趋势是观察位置越接近这个区域，这个区域的细节程度越丰富，越远离这个区域，这个区域

25

内加载的就越只剩下高大的物体。这符合真实的情况，即远离一个区域，只能看到该区域明显的物体，而细节物体会逐渐消失。同不考虑空间细节层次的场景加载方法相比，本方案可以最大程度节省当前场景中加载的物件的数量。

5 场景加载阶段的运行时加卸载的流程如流程图 14 所示，运行时在每一帧都做流程图的处理，每一帧的处理过程又分为两部分，第一部分是在每一个层级的网格上基于视点判断当前可视的场景网格，并将其加入待考察列表；第二部分则是对待考察列表范围内的每个物件根据其加载距离去判定其是否应该被加载。

一种可选的实现步骤如下：

10 S1401，用户终端对于含有多层细节网格结构信息的场景，加载初始化网格信息。

用户终端在程序启动（或场景初始化）时，加载场景文件，将当期场景的所有空间网格区块 $Bounds(l,i,k)$ 和场景物件代理对象 Proxy 保存到内存中。

在程序运行的每一帧（或者定时地）进行如下第 3 步骤及以下的流程。

15 首先，对于每一层级 L 判定网格是否加入待考察列表 C。

S1402，用户终端计算当前视点所在的当前网格序号 id。

S1403，用户终端判断当前 id 与 id_old 相比，是否相同，若是则执行 S1408，否则执行 S1404。

S1404，用户终端设置 $id_old = id$ 。

20 S1405，用户终端找到当前网格 id 附近的 9 块不为空的网格集合 S。

S1406，用户终端将 S 与 S_old 相比，把在 S 不在 S-old 中的网格加入待考察网格列表 C，把不在 S 中在 S_old 中的网格移除 C，并卸载其中的物件。

S1407，用户终端设置 $S_old = S$ 。

在上述方案中可获取当前所处观察位置的 3D 坐标为 Location，遍历当前

的每一层网格进行如下计算：在第 1 层网格上，计算 Location 所处的场景网格坐标 P_location, P_location 在 x 轴上的网格序号 Li 为 $\text{floor}((\text{Location.x} - \text{Bscene_x})/\text{Dl})$ ，在 z 轴上的网格序号 Lk 为 $\text{floor}((\text{Location.z} - \text{Bscene_z})/\text{Dl})$ ，找到所有满足 $\text{abs}(\text{Li}-i) \leq 1$ ，并且 $\text{abs}(\text{Lk}-k) \leq 1$ 的网格区块 Block(l,i,k)，将这些

5 Block 中的 id 列表中记录的所有 Proxy 加入到当前场景的待加载列表 LoadList 中。

对待考察网格中每个物件的可见性判定。

S1408，用户终端计算物件到当前视角的距离的平方 D1。

S1409，用户终端 D1 同该物件的加载距离平方 D2 相比。

10 S1410，若比较结果为小于等于，如果物件没有加载则加载。

S1411，若比较结果为小于大于，如果物件有加载且再所有包含它的网格中不可见则卸载。

在上述方案，对于当前场景中所有已经加载了的 proxy，用户终端检查它是否处于 LoadList 中，如果不处于，则将其卸载或隐藏。遍历当前的 LoadList

15 中的每一个场景物件代理对象 Proxy，计算该物件距离当前观察位置的距离 D，如果 $D < \text{Proxy.D}$ ，则将该物件加载，即读取 Proxy 的物件资源路径和位置信息，将其加载并显示在该位置上。清空当前的 LoadList。

采用本申请的技术方案，至少具备如下技术效果：本方案通过一种动态加载卸载场景区域内容的方案实现了超大型 3D 场景在性能有限的硬件设备（如

20 移动手机）上的加载和连续展现；本方案同现有业界其他方案相比最大的性能提升在于：通过将场景组织成一种多层细节网格结构，相当于表达了场景中的不同物件的细节程度信息，并且对物件基于可视距离加载，而非现有主流方案那样对场景区域不考虑细节程度统一按照同一个空间距离加载，可以最大程度的减少实际上看不到的细节物体的加载，减少场景加载中产生的内存占用，降

25 低图形渲染的复杂度，在同一个场景观察位置，随着摄像机的远离，高大的建筑仍然保留加载，而更细节的物体会逐渐消失，相关技术方案中不考虑细节层

次的分块加载方案难以保证细节的物体先于高大物体消失,通常为了照顾到要
远距离观察到高大的建筑物而扩大加载区块的大小,大量增加了看不见的细节
物件的加载;本方案的场景细节层次结构的划分和处理完全自动化处理,在实
际的项目应用中,场景的艺术设计人员不需要关心考虑场景物件的区块划分或
5 者分块设计,制作流程简单。

需要说明的是,对于前述的各方法实施例,为了简单描述,故将其都表述
为一系列的动作组合,但是本领域技术人员应该知悉,本申请并不受所描述的
动作顺序的限制,因为依据本申请,某些步骤可以采用其他顺序或者同时进行。
其次,本领域技术人员也应该知悉,说明书中所描述的实施例均属于优选实施
10 例,所涉及的动作和模块并不一定是本申请所必须的。

通过以上的实施方式的描述,本领域的技术人员可以清楚地了解到根据上
述实施例的方法可借助软件加必需的通用硬件平台的方式来实现,当然也可以
通过硬件,但很多情况下前者是更佳的实施方式。基于这样的理解,本申请的
技术方案本质上或者说对现有技术做出贡献的部分可以以软件产品的形式体
15 现出来,该计算机软件产品存储在一个存储介质(如 ROM/RAM、磁碟、光盘)
中,包括若干指令用以使得一台终端设备(可以是手机,计算机,服务器,或
者网络设备等)执行本申请各个实施例所述的方法。

根据本申请实施例的另一个方面,还提供了一种用于实施上述对象的加载
20 方法的对象的加载装置。图 15 是根据本申请实施例的一种可选的对象的加载
装置的示意图,如图 15 所示,该装置可以包括:第一确定单元 1501、第二确
定单元 1503、获取单元 1505 以及加载单元 1507。

第一确定单元 1501,用于确定虚拟场景中位于图像采集设备的采集范围
内的可视空间,其中,图像采集设备为位于虚拟场景中的第一位置的采集设备;

25 第二确定单元 1503,用于基于第一位置确定可视空间内位于多种类型中
目标类型所指示的可视距离阈值内的目标子空间,其中,多种类型中的每种类

型用于指示虚拟场景的子空间内对象的一个可视距离阈值；

获取单元 1505，用于获取目标子空间中可视距离不大于目标类型所指示的可视距离阈值的对象为待渲染对象；

加载单元 1507，用于加载待渲染对象。

- 5 需要说明的是，该实施例中的第一确定单元 1501 可以用于执行本申请实施例中的 S202，该实施例中的第二确定单元 1503 可以用于执行本申请实施例中的 S204，该实施例中的获取单元 1505 可以用于执行本申请实施例中的 S206，该实施例中的加载单元 1507 可以用于执行本申请实施例中的 S208。

- 10 此处需要说明的是，上述模块与对应的步骤所实现的示例和应用场景相同，但不限于上述实施例所公开的内容。需要说明的是，上述模块作为装置的一部分可以运行在如图 1 所示的硬件环境中，可以通过软件实现，也可以通过硬件实现。

- 15 通过上述模块，确定目标应用提供的虚拟场景中位于图像采集设备的采集范围内的可视空间，基于第一位置确定可视空间内位于多种类型中目标类型所指示的可视距离阈值内的目标子空间，获取目标子空间中可视距离不大于目标类型所指示的可视距离阈值的对象为待渲染对象；在安装目标应用的用户终端的存储资源中加载待渲染对象，由于所加载的是对用户而言实际可见的对象，可以解决相关技术中渲染虚拟场景中的对象所占用的硬件资源较多的技术问题，可以显著降低对内存的资源占用，还能够降低对 CPU 和 GPU 硬件资源的消耗。
- 20

可选地，第二确定单元包括：第一获取模块，用于获取可视空间的子空间与第一位置之间的间隔距离；查找模块，用于将可视空间内与第一位置之间的间隔距离不大于目标类型所指示的可视距离阈值的子空间作为目标子空间。

- 25 可选地，获取单元包括：第二获取模块，用于获取配置文件，其中，配置文件中记录虚拟场景的子空间内与多种类型中每种类型关联的对象，虚拟场景的子空间内的每个对象根据可视距离被设置为与多种类型中的一种类型关联；

确定模块,用于获取配置文件中记录的目标子空间内与目标类型关联的对象为待渲染对象。

5 可选地,加载单元包括:第一加载模块,用于在保存的对象中包括待渲染对象的情况下,继续保存待渲染对象,并删除除待渲染对象以外的对象;第二加载模块,用于在保存的对象中不包括待渲染对象的情况下,删除保存的对象,并加载待渲染对象。

10 可选地,本申请的装置还可包括,渲染单元,用于在安装有目标应用的用户终端的存储资源中加载待渲染对象之后,将目标子空间的网格密度设置为与目标类型对应的网格密度匹配,并在设置网格密度后的目标子空间中渲染出待渲染对象,其中,多种类型中的任一类型用于指示按照与任一类型对应的网格密度来渲染虚拟场景的子空间内可视距离不大于任一类型所指示的可视距离阈值的对象,多种类型中第一类型对应的网格密度大于多种类型中第二类型对应的网格密度,第一类型所指示的可视距离阈值小于第二类型所指示的可视距离阈值。

15 可选地,该装置还可包括:第三确定单元,用于在确定目标应用提供的虚拟场景中位于图像采集设备的采集范围内的可视空间之前,确定多种类型中的每种类型所对应的网格密度;创建单元,用于按照多种类型中每种类型所对应的网格密度为多种类型中每种类型创建配置信息,并将多种类型中每种类型的配置信息保存至配置文件,其中,配置信息中记录有按照相应的网格密度进行划分的子空间内的对象和子空间所在的第二位置。

20 可选地,上述第三确定单元可包括:第三获取模块,用于获取虚拟场景中所有对象的可视距离中的最大可视距离和最小可视距离;密度确定模块,用于按照最大可视距离和最小可视距离确定多种类型中的每种类型所对应的网格密度。

25 上述的第三获取模块还可用于:获取虚拟场景中每个对象的可视距离,其中,每个对象的可视距离为每个对象的第一乘积与第二乘积之间的比值,第一

乘积为每个对象在三维坐标系中三个维度上的坐标的最大值与预设参数之间的乘积,第二乘积为屏占比阈值与图像采集设备的目标视角的正切值的2倍之间的乘积;从虚拟场景中所有对象的可视距离查找最大可视距离和最小可视距离。

- 5 上述密度确定模块还可用于:对最大可视距离与最小可视距离之间的比值向下取整,得到第一阈值,其中,第一阈值用于表示多种类型的类型数量;将多种类型中第*i*个类型的所对应的网格密度所表示的网格间距设置为目标间距的*i*倍,其中,*i*为大于0且小于等于第一阈值的整数。

- 10 上述创建单元还可用于:确定虚拟场景中按照多种类型中每种类型所对应的网格密度进行划分而得到的子空间;查找虚拟场景中每个子空间内可视距离与每种类型所指示的可视距离阈值匹配的对象,并在一条配置信息中保存每个子空间与查找到的对象之间的关联关系。

- 15 此处需要说明的是,上述模块与对应的步骤所实现的示例和应用场景相同,但不限于上述实施例所公开的内容。需要说明的是,上述模块作为装置的一部分可以运行在如图1所示的硬件环境中,可以通过软件实现,也可以通过硬件实现,其中,硬件环境包括网络环境。

根据本申请实施例的另一个方面,还提供了一种用于实施上述对象的加载方法的服务器或终端。

- 20 图16是根据本申请实施例的一种终端的结构框图,如图16所示,该终端可以包括:一个或多个(图16中仅示出一个)处理器1601、存储器1603、以及传输装置1605,如图16所示,该终端还可以包括输入输出设备1607。

- 25 其中,存储器1603可用于存储软件程序以及模块,如本申请实施例中的对象的加载方法和装置对应的程序指令/模块,处理器1601通过运行存储在存储器1603内的软件程序以及模块,从而执行各种功能应用以及数据处理,即实现上述的对象的加载方法。存储器1603可包括高速随机存储器,还可以包

括非易失性存储器，如一个或者多个磁性存储装置、闪存、或者其他非易失性固态存储器。在一些实例中，存储器 1603 可进一步包括相对于处理器 1601 远程设置的存储器，这些远程存储器可以通过网络连接至终端。上述网络的实例包括但不限于互联网、企业内部网、局域网、移动通信网及其组合。

5 上述的传输装置 1605 用于经由一个网络接收或者发送数据，还可以用于处理器与存储器之间的数据传输。上述的网络具体实例可包括有线网络及无线网络。在一个实例中，传输装置 1605 包括一个网络适配器 (Network Interface Controller, NIC)，其可通过网线与其他网络设备与路由器相连从而可与互联网或局域网进行通讯。在一个实例中，传输装置 1605 为射频 (Radio Frequency, 10 RF) 模块，其用于通过无线方式与互联网进行通讯。

其中，具体地，存储器 1603 用于存储应用程序。

处理器 1601 可以通过传输装置 1605 调用存储器 1603 存储的应用程序，以执行下述步骤：

15 确定虚拟场景中位于图像采集设备的采集范围内的可视空间，其中，图像采集设备为位于虚拟场景中的第一位置的采集设备；

基于第一位置确定可视空间内位于多种类型中目标类型所指示的可视距离阈值内的目标子空间，其中，多种类型中的每种类型用于指示虚拟场景的子空间内对象的一个可视距离阈值；

20 获取目标子空间中可视距离不大于目标类型所指示的可视距离阈值的对象为待渲染对象；

加载待渲染对象。

处理器 1601 还用于执行下述步骤：

获取可视空间的子空间与第一位置之间的间隔距离；

25 将可视空间内与第一位置之间的间隔距离不大于目标类型所指示的可视距离阈值的子空间作为目标子空间。

可选地，本实施例中的具体示例可以参考上述实施例中所描述的示例，本实施例在此不再赘述。

本领域普通技术人员可以理解，图 16 所示的结构仅为示意，终端可以是智能手机（如 Android 手机、iOS 手机等）、平板电脑、掌上电脑以及移动互
5 联网设备（Mobile Internet Devices, MID）、PAD 等终端设备。图 16 其并不对上述电子装置的结构造成限定。例如，终端还可包括比图 16 中所示更多或者更少的组件（如网络接口、显示装置等），或者具有与图 16 所示不同的配置。

本领域普通技术人员可以理解上述实施例的各种方法中的全部或部分步骤是可以通过程序来指令终端设备相关的硬件来完成，该程序可以存储于一计
10 算机可读存储介质中，存储介质可以包括：闪存盘、只读存储器（Read-Only Memory, ROM）、随机存取器（Random Access Memory, RAM）、磁盘或光盘等。

本申请的实施例还提供了一种存储介质。可选地，在本实施例中，上述存
15 储介质可以用于执行对象的加载方法的程序代码。

可选地，在本实施例中，上述存储介质可以位于上述实施例所示的网络中的多个网络设备中的至少一个网络设备上。

可选地，在本实施例中，存储介质被设置为存储用于执行以下步骤的程序代码：

20 确定虚拟场景中位于图像采集设备的采集范围内的可视空间，其中，图像采集设备为位于虚拟场景中的第一位置的采集设备；

基于第一位置确定可视空间内位于多种类型中目标类型所指示的可视距离阈值内的目标子空间，其中，多种类型中的每种类型用于指示虚拟场景的子空间内对象的一个可视距离阈值；

25 获取目标子空间中可视距离不大于目标类型所指示的可视距离阈值的对

象为待渲染对象;

加载待渲染对象。

可选地, 存储介质还被设置为存储用于执行以下步骤的程序代码:

获取可视空间的子空间与第一位置之间的间隔距离;

- 5 将可视空间内与第一位置之间的间隔距离不大于目标类型所指示的可视距离阈值的子空间作为目标子空间。

可选地, 本实施例中的具体示例可以参考上述实施例中所描述的示例, 本实施例在此不再赘述。

- 10 可选地, 在本实施例中, 上述存储介质可以包括但不限于: U 盘、只读存储器 (Read-Only Memory, ROM)、随机存取存储器 (Random Access Memory, RAM)、移动硬盘、磁碟或者光盘等各种可以存储程序代码的介质。

本申请的实施例还提供了一种计算机程序产品, 包括指令, 当其在计算机上运行时, 使得所述计算机执行上述的对象的加载方法。

上述本申请实施例序号仅仅为了描述, 不代表实施例的优劣。

- 15 上述实施例中的集成的单元如果以软件功能单元的形式实现并作为独立的产品销售或使用, 可以存储在上述计算机可读的存储介质中。基于这样的理解, 本申请的技术方案本质上或者说对现有技术做出贡献的部分或者该技术方案的全部或部分可以以软件产品的形式体现出来, 该计算机软件产品存储在存储介质中, 包括若干指令用以使得一台或多台计算机设备 (可为个人计算机、服务器或者网络设备) 执行本申请各个实施例所述方法的全部或部分步
- 20 骤。

在本申请的上述实施例中, 对各个实施例的描述都各有侧重, 某个实施例中未详述的部分, 可以参见其他实施例的相关描述。

在本申请所提供的几个实施例中, 应该理解到, 所揭露的客户端, 可通过

其它的方式实现。其中，以上所描述的装置实施例仅仅是示意性的，例如所述单元的划分，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，单元或模块的间接耦合或通信连接，可以是电性或其它的形式。

所述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，也可以采用软件功能单元的形式实现。

以上所述仅是本申请的优选实施方式，应当指出，对于本技术领域的普通技术人员来说，在不脱离本申请原理的前提下，还可以做出若干改进和润饰，这些改进和润饰也应视为本申请的保护范围。

权 利 要 求

1. 一种对象的加载方法，应用于图像处理设备，包括：

确定虚拟场景中位于图像采集设备的采集范围内的可视空间，其中，所述图像采集设备为位于所述虚拟场景中的第一位置的采集设备；

5 基于所述第一位置确定所述可视空间内位于多种类型中目标类型所指示的可视距离阈值内的目标子空间，其中，所述多种类型中的每种类型用于指示所述虚拟场景的子空间内对象的一个可视距离阈值；

获取所述目标子空间中可视距离不大于所述目标类型所指示的可视距离阈值的对象为待渲染对象；

10 加载所述待渲染对象。

2. 根据权利要求 1 所述的方法，所述基于所述第一位置确定所述可视空间内位于多种类型中目标类型所指示的可视距离阈值内的目标子空间，包括：

获取所述可视空间的子空间与所述第一位置之间的间隔距离；

15 将所述可视空间内与所述第一位置之间的间隔距离不大于所述目标类型所指示的可视距离阈值的子空间作为所述目标子空间。

3. 根据权利要求 1 所述的方法，所述获取所述目标子空间中可视距离不大于所述目标类型所指示的可视距离阈值的对象为待渲染对象，包括：

20 获取配置文件，其中，所述配置文件中记录所述虚拟场景的子空间内与所述多种类型中每种类型关联的对象，所述虚拟场景的子空间内的每个对象根据可视距离被设置为与所述多种类型中的一种类型关联；

获取所述配置文件中记录的所述目标子空间内与所述目标类型关联的对象为所述待渲染对象。

4. 根据权利要求 1 所述的方法，所述加载所述待渲染对象，包括：

25 在保存的对象中包括所述待渲染对象的情况下，继续保存所述待渲染对象，并删除除所述待渲染对象以外的对象；

在所述保存的对象中不包括所述待渲染对象的情况下，删除所述保存的对象，并加载所述待渲染对象。

5. 根据权利要求 1 所述的方法，在所述加载所述待渲染对象之后，所述方法还包括：

将所述目标子空间的网格密度设置为与所述目标类型对应的网格密度匹配，并在设置网格密度后的所述目标子空间中渲染出所述待渲染目标对象，其中，所述多种类型中的任一类型用于指示按照与所述任一类型对应的网格密度来渲染所述虚拟场景的子空间内可视距离不大于所述任一类型所指示的可视距离阈值的对象，所述多种类型中第一类型对应的网格密度大于所述多种类型中第二类型对应的网格密度，所述第一类型所指示的可视距离阈值小于所述第二类型所指示的可视距离阈值。

6. 根据权利要求 1 至 5 中任意一项所述的方法，在所述确定虚拟场景中位于图像采集设备的采集范围内的可视空间之前，所述方法还包括：

10 确定所述多种类型中的每种类型所对应的网格密度；

按照所述多种类型中每种类型所对应的网格密度为所述多种类型中每种类型创建配置信息，并将所述多种类型中每种类型的配置信息保存至配置文件，其中，所述配置信息中记录有按照相应的网格密度进行划分的子空间内的对象和子空间所在的第二位置。

15 7. 根据权利要求 6 所述的方法，所述确定所述多种类型中的每种类型所对应的网格密度，包括：

获取所述虚拟场景中所有对象的可视距离中的最大可视距离和最小可视距离；

20 按照所述最大可视距离和所述最小可视距离确定所述多种类型中的每种类型所对应的网格密度。

8. 根据权利要求 7 所述的方法，所述获取所述虚拟场景中所有对象的可视距离中的最大可视距离和最小可视距离，包括：

25 获取所述虚拟场景中每个对象的可视距离，其中，所述每个对象的可视距离为所述每个对象的第一乘积与第二乘积之间的比值，所述第一乘积为所述每个对象在三维坐标系中三个维度上的坐标的最大值与预设参数之间的乘积，所述第二乘积为屏占比阈值与所述图像采集设备的目标视角的正切值的 2 倍之间的乘积；

从所述虚拟场景中所有对象的可视距离查找所述最大可视距离和所述最小可视距离。

9. 根据权利要求 7 所述的方法, 所述按照所述最大可视距离和所述最小可视距离确定所述多种类型中的每种类型所对应的网格密度, 包括:

对所述最大可视距离与所述最小可视距离之间的比值向下取整, 得到第一阈值, 其中, 所述第一阈值用于表示所述多种类型的类型数量;

5 将所述多种类型中第 i 个类型的所对应的网格密度所表示的网格间距设置为目标间距的 i 倍, 其中, i 为大于 0 且小于等于第一阈值的整数。

10. 根据权利要求 6 所述的方法, 所述按照所述多种类型中每种类型所对应的网格密度为所述多种类型中每种类型创建配置信息, 包括:

10 确定所述虚拟场景中按照所述多种类型中每种类型所对应的网格密度进行划分而得到的子空间;

查找所述虚拟场景中每个子空间内可视距离与所述每种类型所指示的可视距离阈值匹配的对象, 并在一条配置信息中保存每个子空间与查找到的对象之间的关联关系。

11. 一种对象的加载装置, 包括:

15 第一确定单元, 用于确定虚拟场景中位于图像采集设备的采集范围内的可视空间, 其中, 所述图像采集设备为位于所述虚拟场景中的第一位置的采集设备;

20 第二确定单元, 用于基于所述第一位置确定所述可视空间内位于多种类型中目标类型所指示的可视距离阈值内的目标子空间, 其中, 所述多种类型中的每种类型用于指示所述虚拟场景的子空间内对象的一个可视距离阈值;

获取单元, 用于获取所述目标子空间中可视距离不大于所述目标类型所指示的可视距离阈值的对象为待渲染对象;

加载单元, 用于加载所述待渲染对象。

12. 根据权利要求 11 所述的装置, 所述第二确定单元, 包括:

25 第一获取模块, 用于获取所述可视空间的子空间与所述第一位置之间的间隔距离;

查找模块, 用于将所述可视空间内与所述第一位置之间的间隔距离不大于所述目标类型所指示的可视距离阈值的子空间作为所述目标子空间。

13. 根据权利要求 11 所述的装置, 所述获取单元, 包括:

第二获取模块，用于获取配置文件，其中，所述配置文件中记录所述虚拟场景的子空间内与所述多种类型中每种类型关联的对象，所述虚拟场景的子空间内的每个对象根据可视距离被设置为与所述多种类型中的一种类型关联；

确定模块，用于获取所述配置文件中记录的所述目标子空间内与所述目标
5 类型关联的对象为所述待渲染对象。

14.一种存储介质，所述存储介质包括存储的程序，其中，所述程序运行时执行上述权利要求 1 至 10 任一项中所述的方法。

15.一种电子装置，包括存储器、处理器及存储在所述存储器上并可在所述处理器上运行的计算机程序，所述处理器通过所述计算机程序执行上述权利
10 要求 1 至 10 任一项中所述的方法。

16.一种计算机程序产品，包括指令，当其在计算机上运行时，使得所述计算机执行权利要求 1-10 任意一项所述的对象的加载方法。

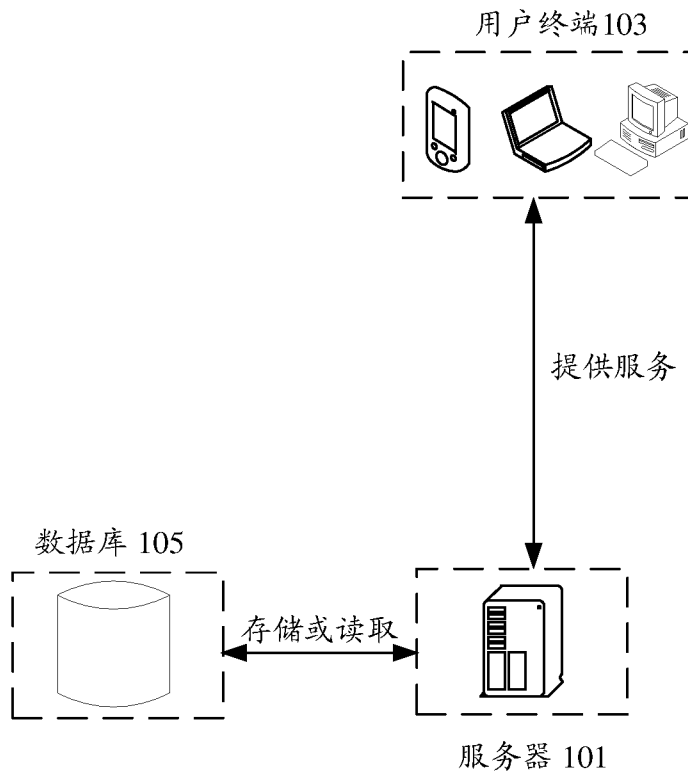


图 1

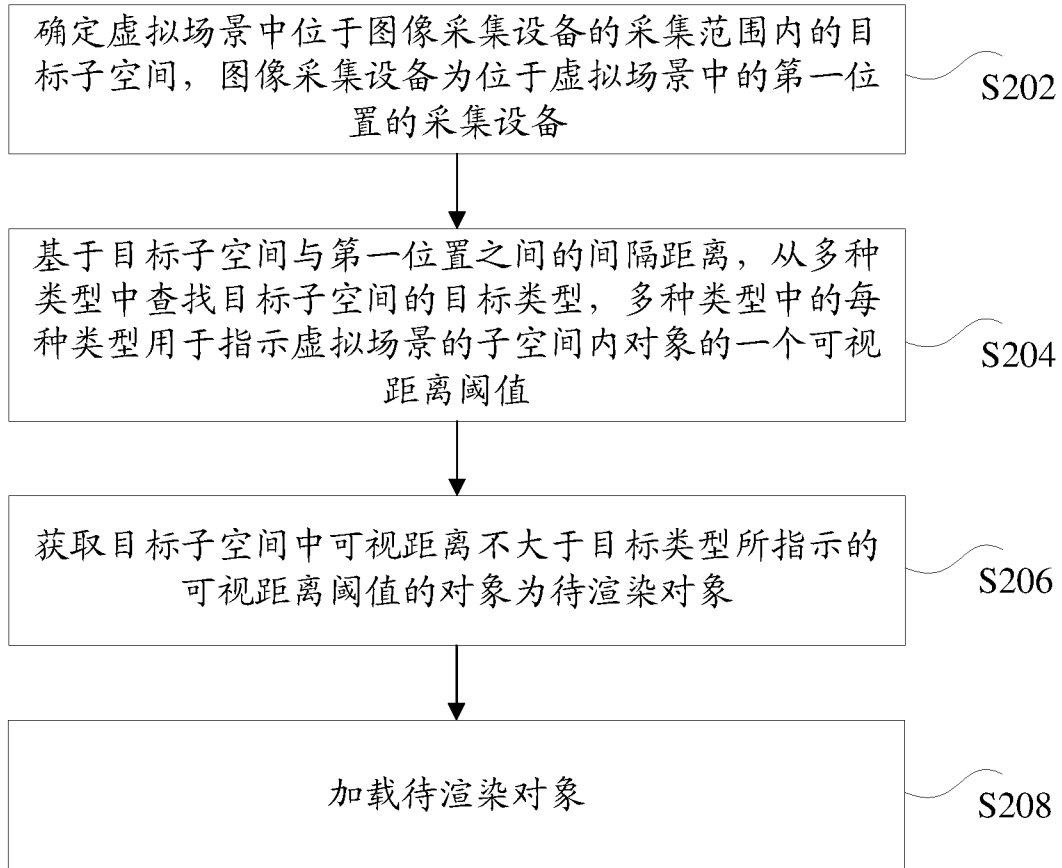


图 2

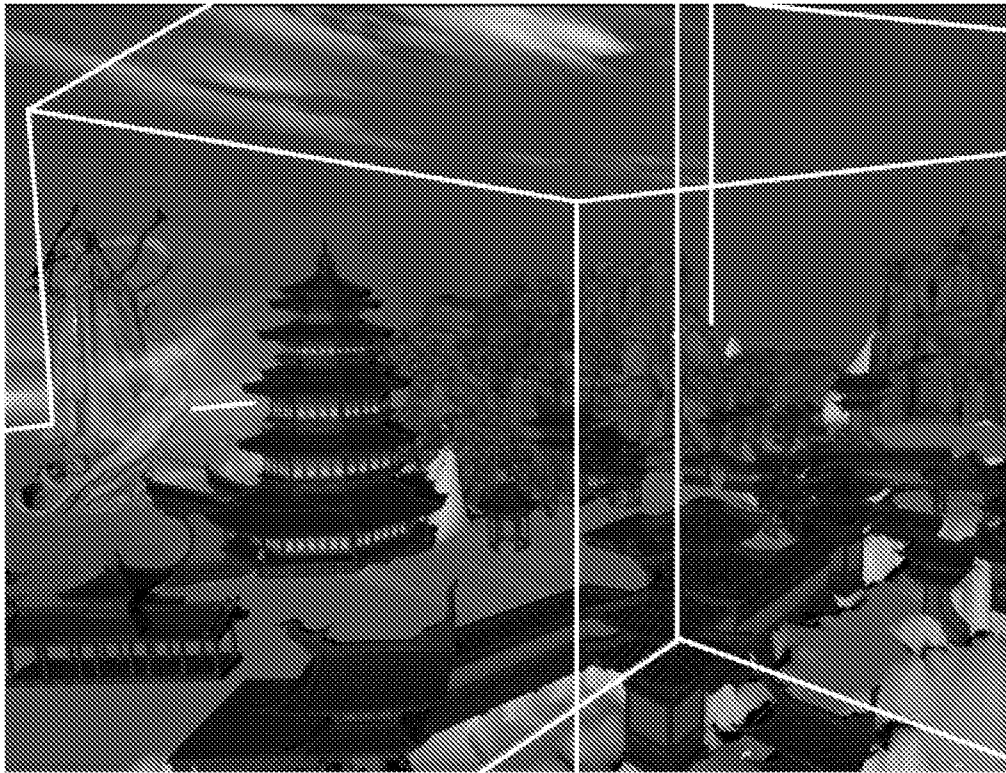


图 3

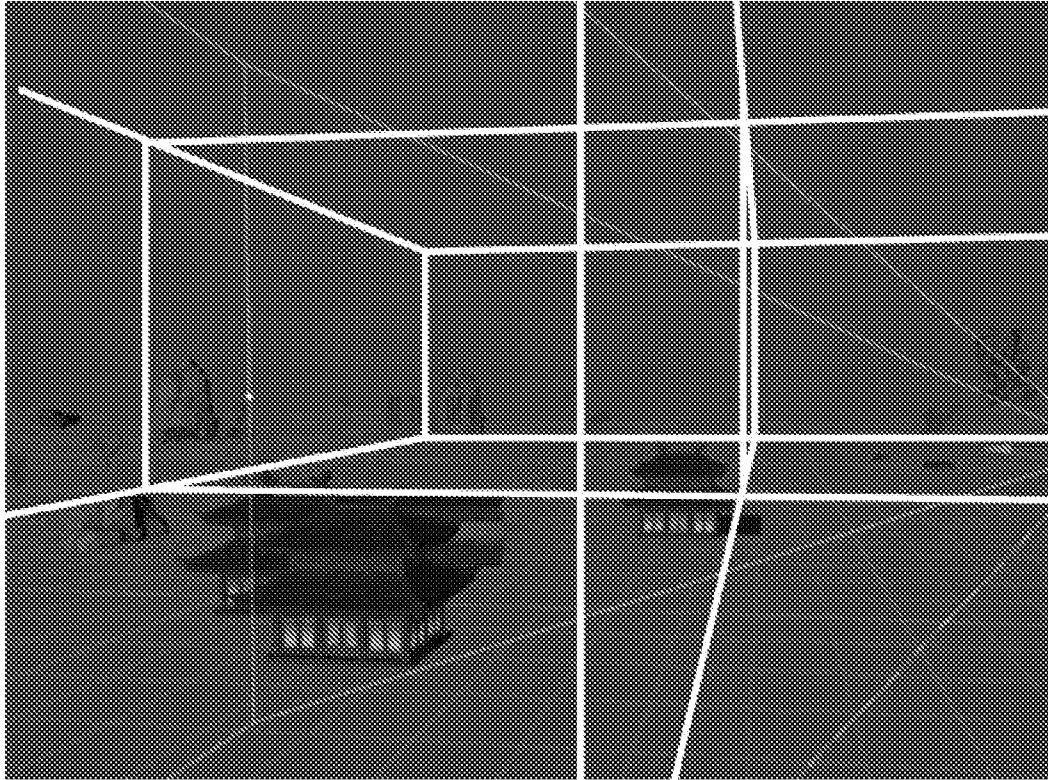


图 4

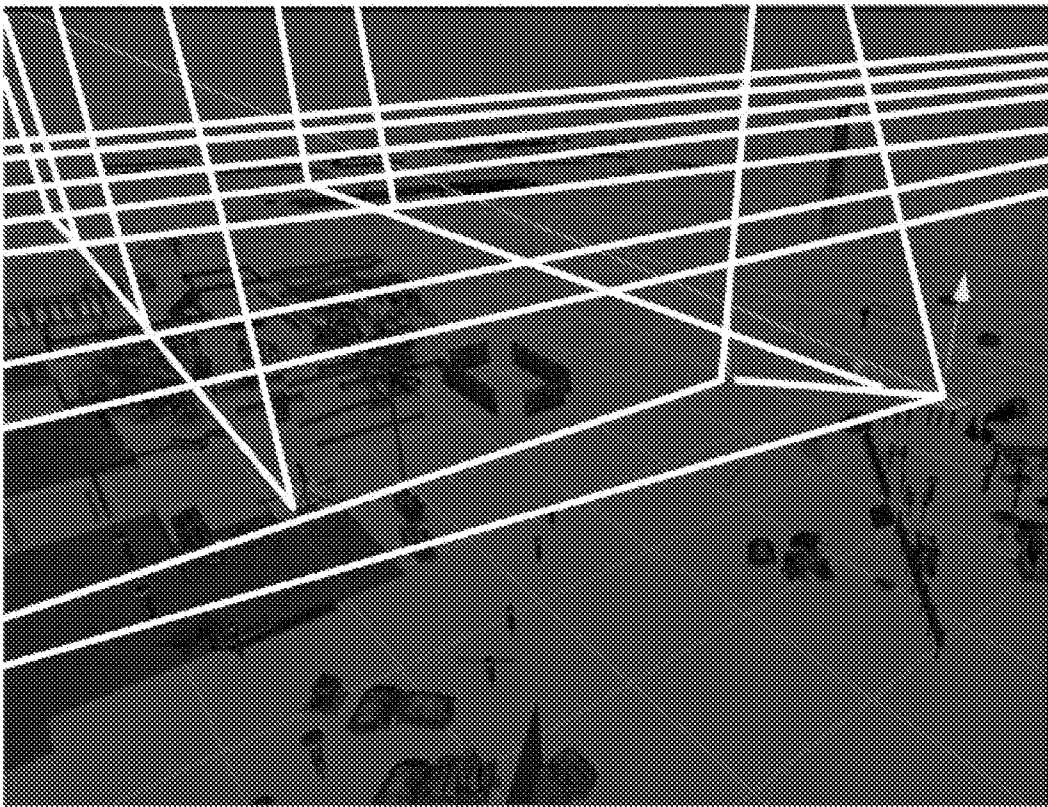


图 5

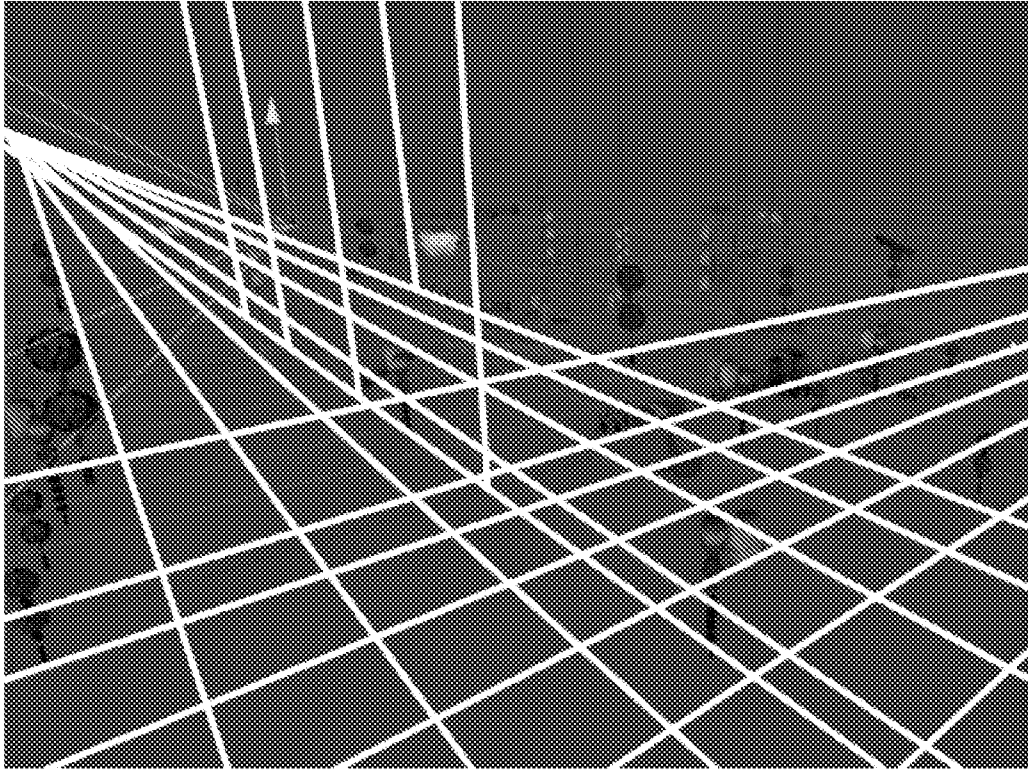


图 6

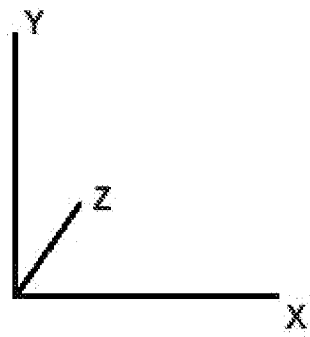


图 7

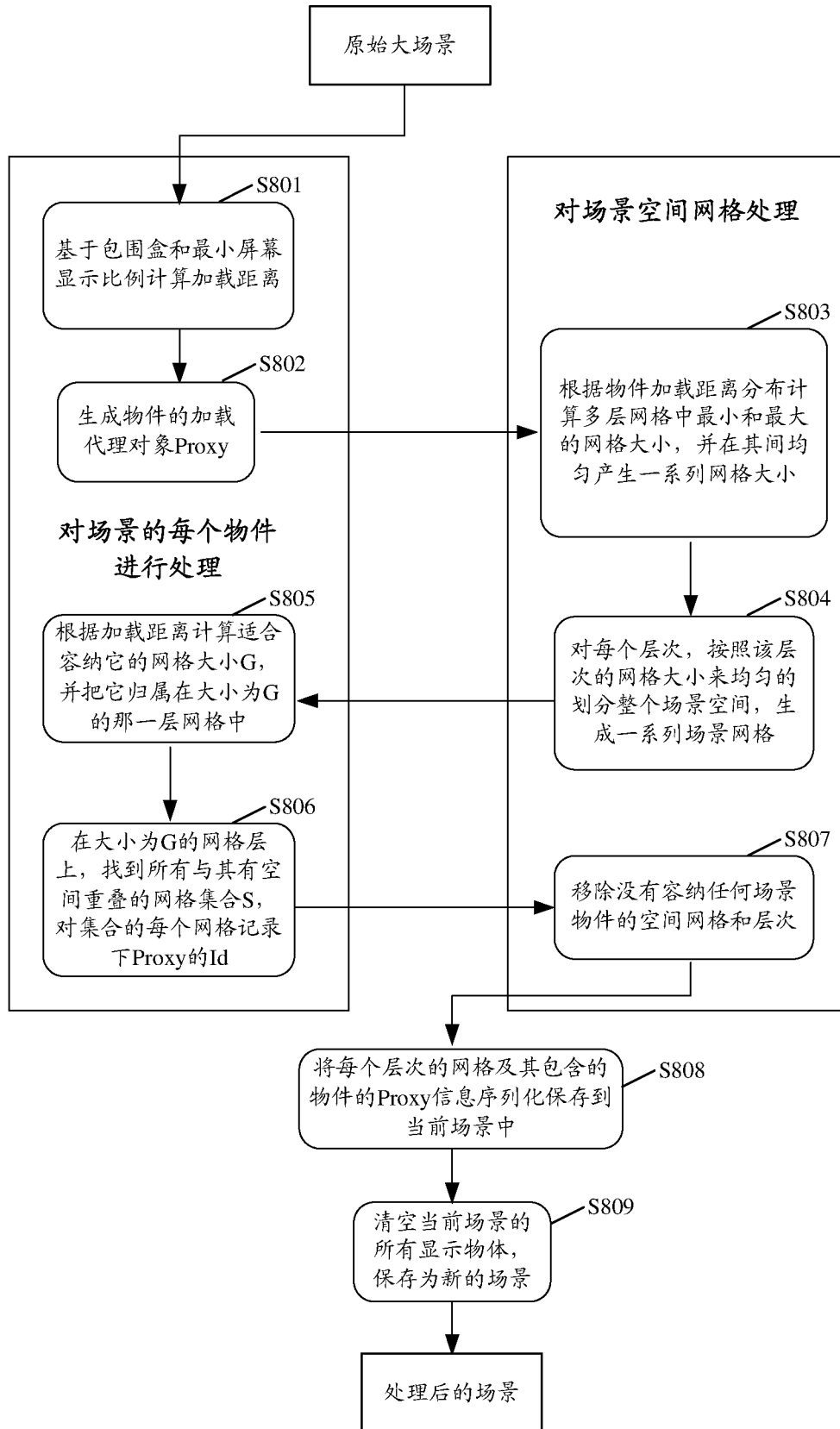


图 8

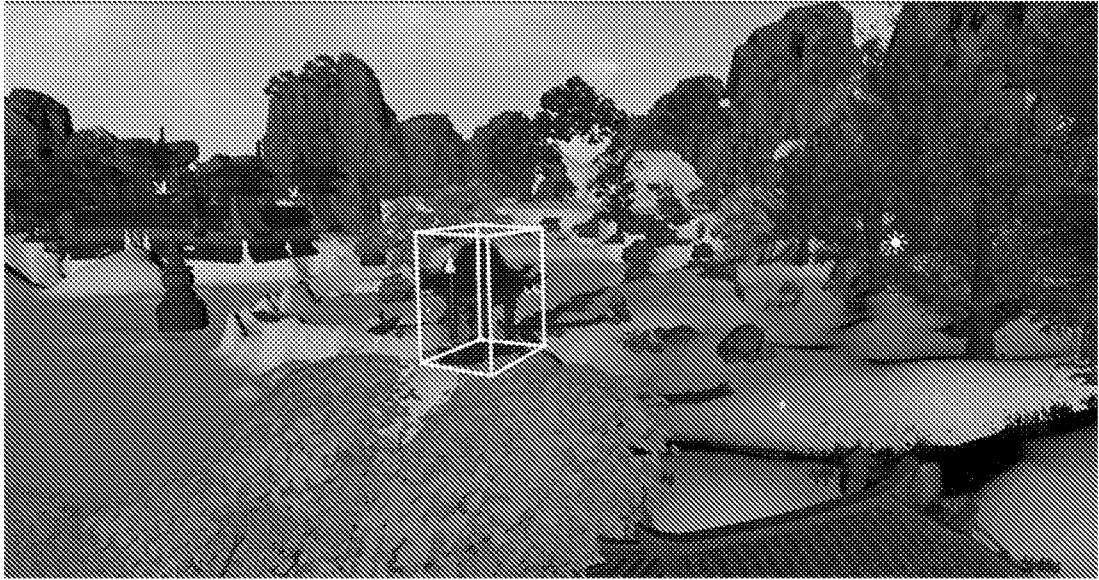


图 9

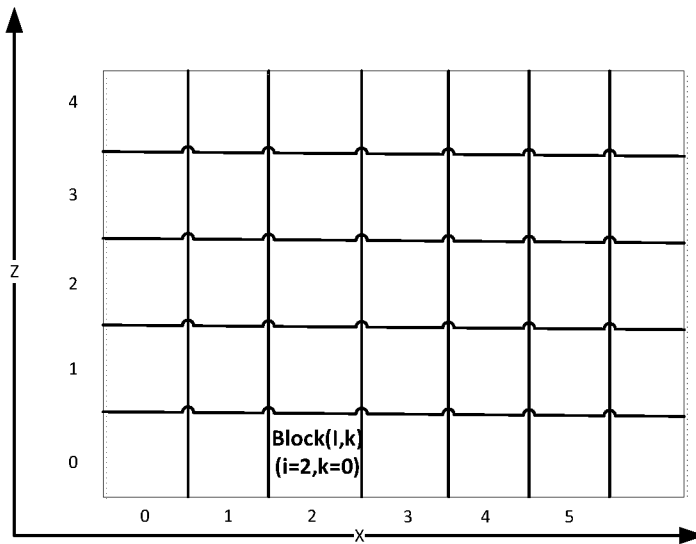


图 10

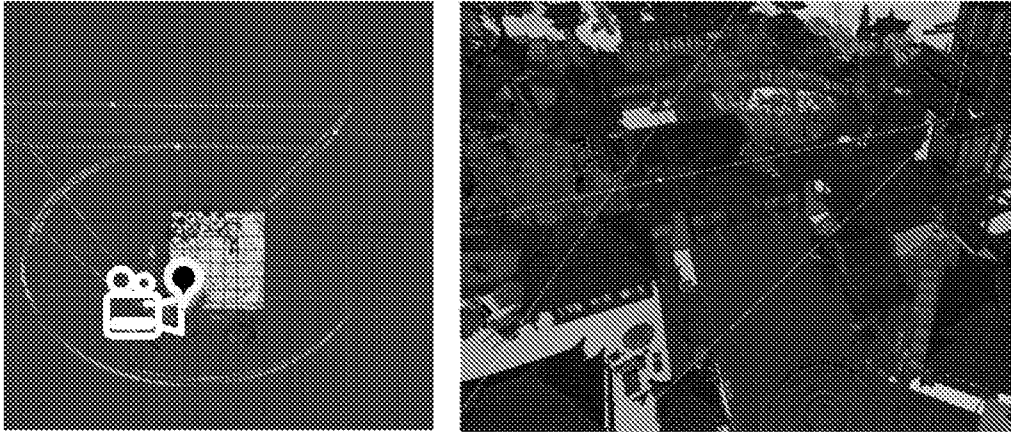


图 11

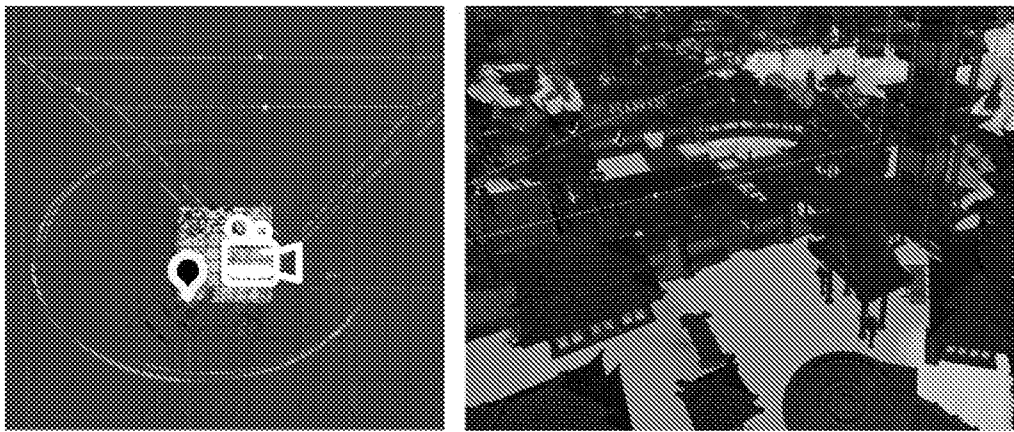


图 12

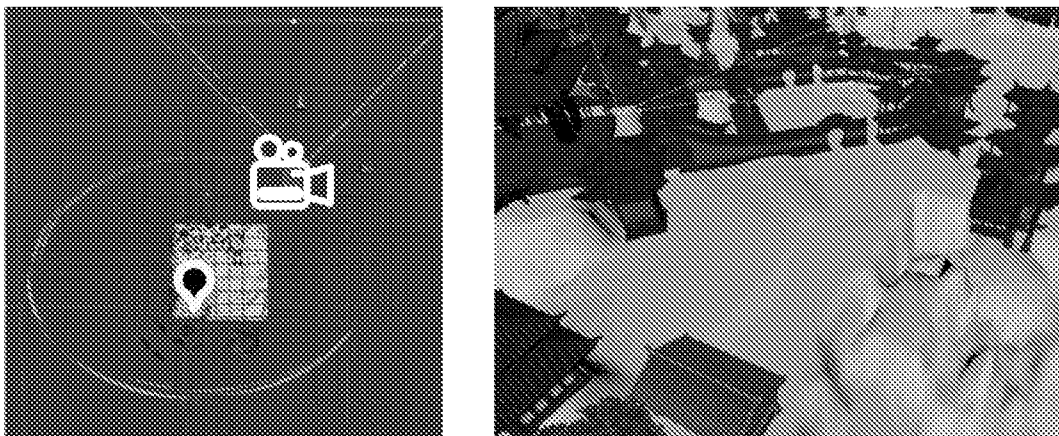


图 13

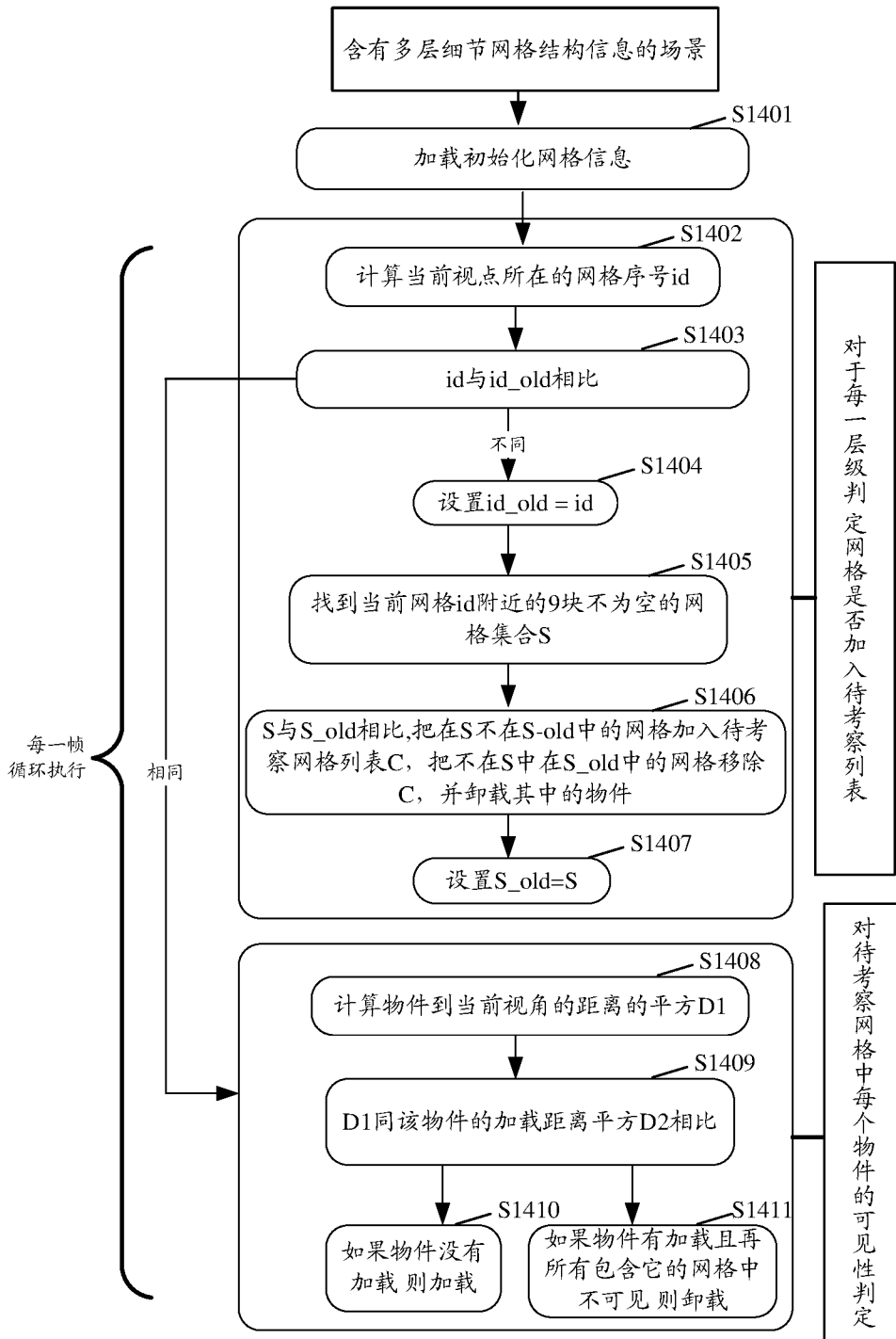


图 14

— 10/11 —

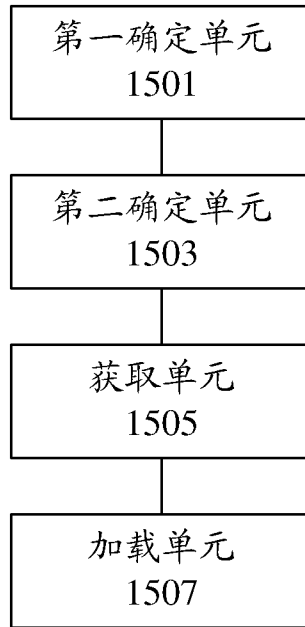


图 15

输入输出设备 1607

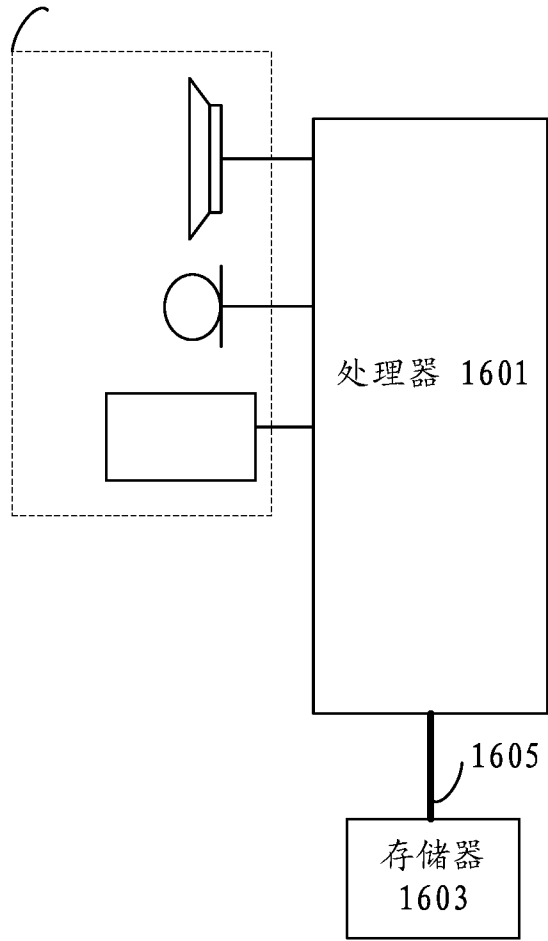


图 16

INTERNATIONAL SEARCH REPORT

International application No.

PCT/CN2019/115611

A. CLASSIFICATION OF SUBJECT MATTER G06T 15/20(2011.01)i According to International Patent Classification (IPC) or to both national classification and IPC		
B. FIELDS SEARCHED Minimum documentation searched (classification system followed by classification symbols) G06T Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched Electronic data base consulted during the international search (name of data base and, where practicable, search terms used) CNABS, CNTXT, VEN, USTXT, WOTXT, EPTXT, CNKI: 对象, 图像, 虚拟, 可视, 距离, 阈值, object, image, virtual, visible, distance, threshold		
C. DOCUMENTS CONSIDERED TO BE RELEVANT		
Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
PX	CN 109523621 A (TENCENT TECHNOLOGY SHENZHEN CO., LTD.) 26 March 2019 (2019-03-26) entire document	1-16
X	CN 106991713 A (NETEASE (HANGZHOU) NETWORK CO., LTD.) 28 July 2017 (2017-07-28) description, paragraphs 9-22, 28-31 and 75-78, and figures 1-2	1-16
A	CN 105701851 A (STARGIS (TIANJIN) TECHNOLOGY DEVELOPMENT CO., LTD.) 22 June 2016 (2016-06-22) entire document	1-16
A	US 2006008121 A1 (MICROSOFT CORPORATION) 12 January 2006 (2006-01-12) entire document	1-16
<input type="checkbox"/> Further documents are listed in the continuation of Box C. <input checked="" type="checkbox"/> See patent family annex.		
<p>* Special categories of cited documents:</p> <p>“A” document defining the general state of the art which is not considered to be of particular relevance</p> <p>“E” earlier application or patent but published on or after the international filing date</p> <p>“L” document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)</p> <p>“O” document referring to an oral disclosure, use, exhibition or other means</p> <p>“P” document published prior to the international filing date but later than the priority date claimed</p> <p>“T” later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention</p> <p>“X” document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone</p> <p>“Y” document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art</p> <p>“&” document member of the same patent family</p>		
Date of the actual completion of the international search 09 January 2020		Date of mailing of the international search report 22 January 2020
Name and mailing address of the ISA/CN China National Intellectual Property Administration No. 6, Xitucheng Road, Jimenqiao Haidian District, Beijing 100088 China Facsimile No. (86-10)62019451		Authorized officer Telephone No.

INTERNATIONAL SEARCH REPORT
Information on patent family members

International application No.

PCT/CN2019/115611

Patent document cited in search report			Publication date (day/month/year)	Patent family member(s)			Publication date (day/month/year)
CN	109523621	A	26 March 2019	None			
CN	106991713	A	28 July 2017	None			
CN	105701851	A	22 June 2016	CN	105701851	B	19 February 2019
US	2006008121	A1	12 January 2006	US	7308115	B2	11 December 2007

国际检索报告

国际申请号

PCT/CN2019/115611

<p>A. 主题的分类</p> <p>G06T 15/20 (2011.01) i</p> <p>按照国际专利分类(IPC)或者同时按照国家分类和IPC两种分类</p>																	
<p>B. 检索领域</p> <p>检索的最低限度文献(标明分类系统和分类号)</p> <p>G06T</p> <p>包含在检索领域中的除最低限度文献以外的检索文献</p> <p>在国际检索时查阅的电子数据库(数据库的名称, 和使用的检索词(如使用))</p> <p>CNABS, CNTXT, VEN, USTXT, WOTXT, EPTXT, CNKI:对象, 图像, 虚拟, 可视, 距离, 阈值, object, image, virtual, visible, distance, threshold</p>																	
<p>C. 相关文件</p> <table border="1"> <thead> <tr> <th>类型*</th> <th>引用文件, 必要时, 指明相关段落</th> <th>相关的权利要求</th> </tr> </thead> <tbody> <tr> <td>PX</td> <td>CN 109523621 A (腾讯科技深圳有限公司) 2019年 3月 26日 (2019 - 03 - 26) 全文</td> <td>1-16</td> </tr> <tr> <td>X</td> <td>CN 106991713 A (网易杭州网络有限公司) 2017年 7月 28日 (2017 - 07 - 28) 说明书第9-22、28-31、75-78段, 图1-2</td> <td>1-16</td> </tr> <tr> <td>A</td> <td>CN 105701851 A (星际空间天津科技发展有限公司) 2016年 6月 22日 (2016 - 06 - 22) 全文</td> <td>1-16</td> </tr> <tr> <td>A</td> <td>US 2006008121 A1 (MICROSOFT CORPORATION) 2006年 1月 12日 (2006 - 01 - 12) 全文</td> <td>1-16</td> </tr> </tbody> </table>			类型*	引用文件, 必要时, 指明相关段落	相关的权利要求	PX	CN 109523621 A (腾讯科技深圳有限公司) 2019年 3月 26日 (2019 - 03 - 26) 全文	1-16	X	CN 106991713 A (网易杭州网络有限公司) 2017年 7月 28日 (2017 - 07 - 28) 说明书第9-22、28-31、75-78段, 图1-2	1-16	A	CN 105701851 A (星际空间天津科技发展有限公司) 2016年 6月 22日 (2016 - 06 - 22) 全文	1-16	A	US 2006008121 A1 (MICROSOFT CORPORATION) 2006年 1月 12日 (2006 - 01 - 12) 全文	1-16
类型*	引用文件, 必要时, 指明相关段落	相关的权利要求															
PX	CN 109523621 A (腾讯科技深圳有限公司) 2019年 3月 26日 (2019 - 03 - 26) 全文	1-16															
X	CN 106991713 A (网易杭州网络有限公司) 2017年 7月 28日 (2017 - 07 - 28) 说明书第9-22、28-31、75-78段, 图1-2	1-16															
A	CN 105701851 A (星际空间天津科技发展有限公司) 2016年 6月 22日 (2016 - 06 - 22) 全文	1-16															
A	US 2006008121 A1 (MICROSOFT CORPORATION) 2006年 1月 12日 (2006 - 01 - 12) 全文	1-16															
<p><input type="checkbox"/> 其余文件在C栏的续页中列出。</p> <p><input checked="" type="checkbox"/> 见同族专利附件。</p>																	
<p>* 引用文件的具体类型:</p> <p>“A” 认为不特别相关的表示了现有技术一般状态的文件</p> <p>“E” 在国际申请日的当天或之后公布的在先申请或专利</p> <p>“L” 可能对优先权要求构成怀疑的文件, 或为确定另一篇引用文件的公布日而引用的或者因其他特殊理由而引用的文件(如具体说明的)</p> <p>“O” 涉及口头公开、使用、展览或其他方式公开的文件</p> <p>“P” 公布日先于国际申请日但迟于所要求的优先权日的文件</p> <p>“T” 在申请日或优先权日之后公布, 与申请不相抵触, 但为了理解发明之理论或原理的在后文件</p> <p>“X” 特别相关的文件, 单独考虑该文件, 认定要求保护的发明不是新颖的或不具有创造性</p> <p>“Y” 特别相关的文件, 当该文件与另一篇或者多篇该类文件结合并且这种结合对于本领域技术人员为显而易见时, 要求保护的发明不具有创造性</p> <p>“&” 同族专利的文件</p>																	
<p>国际检索实际完成的日期</p> <p>2020年 1月 9日</p>		<p>国际检索报告邮寄日期</p> <p>2020年 1月 22日</p>															
<p>ISA/CN的名称和邮寄地址</p> <p>中国国家知识产权局(ISA/CN) 中国北京市海淀区蓟门桥西土城路6号 100088</p> <p>传真号 (86-10)62019451</p>		<p>授权官员</p> <p>古志春</p> <p>电话号码 86-(20)-28950386</p>															

国际检索报告
关于同族专利的信息

国际申请号
PCT/CN2019/115611

检索报告引用的专利文件			公布日 (年/月/日)	同族专利			公布日 (年/月/日)
CN	109523621	A	2019年 3月 26日	无			
CN	106991713	A	2017年 7月 28日	无			
CN	105701851	A	2016年 6月 22日	CN	105701851	B	2019年 2月 19日
US	2006008121	A1	2006年 1月 12日	US	7308115	B2	2007年 12月 11日