



(12)发明专利申请

(10)申请公布号 CN 111133763 A

(43)申请公布日 2020.05.08

(21)申请号 201880061896.0

(22)申请日 2018.09.21

(30)优先权数据

10-2018-0040350 2018.04.06 KR

10-2018-0077375 2018.07.04 KR

62/563,093 2017.09.26 US

(85)PCT国际申请进入国家阶段日

2020.03.24

(86)PCT国际申请的申请数据

PCT/KR2018/011323 2018.09.21

(87)PCT国际申请的公布数据

W02019/066436 KO 2019.04.04

(71)申请人 LG 电子株式会社

地址 韩国首尔

(72)发明人 许惠桢 吴世珍 黄修振 李长远

(74)专利代理机构 中原信达知识产权代理有限
责任公司 11219

代理人 夏凯 张伟峰

(51)Int.Cl.

H04N 21/236(2006.01)

H04N 21/434(2006.01)

H04N 21/435(2006.01)

H04N 21/81(2006.01)

H04N 13/366(2006.01)

H04N 13/363(2006.01)

H04N 13/194(2006.01)

H04N 21/218(2006.01)

H04N 21/2368(2006.01)

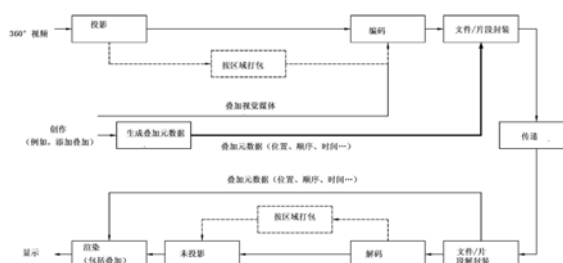
权利要求书2页 说明书45页 附图45页

(54)发明名称

360视频系统中的叠加处理方法及其设备

(57)摘要

一种根据本发明的由360视频接收设备执行的360图像数据处理方法,包括下述步骤:接收360图像数据;从360图像数据获取关于编码的图片的信息和元数据;基于关于编码的图片的信息对图片进行解码;以及基于元数据来渲染解码后的图片和叠加,其中元数据包括与叠加有关的元数据,其中可以基于与叠加有关的元数据来渲染叠加,并且与叠加有关的元数据包括关于叠加的区域的信息。



1. 一种由360度视频接收设备执行的360度图像数据处理方法,所述方法包括:
接收360度图像数据;
从所述360度图像数据获取关于编码的图片的信息和元数据;
基于所述关于编码的图片的信息来解码图片;以及
基于所述元数据来渲染所述解码的图片和叠加,
其中,所述元数据包括与叠加有关的元数据,
其中,基于所述与叠加有关的元数据来渲染所述叠加,并且
其中,所述与叠加有关的元数据包括关于所述叠加的区域的信息。
2. 根据权利要求1所述的方法,其中,关于所述叠加的区域的信息基于方位角和仰角来指示所述叠加的区域。
3. 根据权利要求1所述的方法,其中,关于所述叠加的区域的信息包括所述叠加的位置信息和大小信息中的至少一个。
4. 根据权利要求3所述的方法,其中,所述叠加的位置信息指示在所述解码的图片被渲染到的3D坐标空间内的所述叠加的中心点的x坐标、y坐标和z坐标中的至少一个,并且
其中,所述叠加的大小信息指示所述叠加的宽度和高度中的至少一个。
5. 根据权利要求1所述的方法,其中,所述与叠加有关的元数据包括关于所述叠加的类型的信息,并且
其中,关于所述叠加的类型的所述信息指示所述叠加的类型是在3D空间中被渲染的叠加类型、在视口中被渲染的叠加类型或者在球体表面中被渲染的叠加类型。
6. 根据权利要求5所述的方法,其中,当关于所述叠加的类型的信息指示所述叠加的类型是在3D空间中被渲染的叠加类型时,关于所述叠加的区域的信息基于方位角和仰角来指示所述叠加的区域。
7. 根据权利要求5所述的方法,其中,当关于所述叠加的类型的信息指示所述叠加的类型是在3D空间中被渲染的叠加类型时,所述与叠加有关的元数据包括关于所述叠加的旋转的信息。
8. 根据权利要求7所述的方法,其中,关于所述叠加的旋转的信息指示关于所述叠加的旋转的偏航值、俯仰值和横滚值中的至少一个。
9. 根据权利要求5所述的方法,其中,当关于所述叠加的类型的信息指示所述叠加的类型是在3D空间中被渲染的叠加类型时,所述与叠加有关的元数据进一步包括指示所述叠加的层或平面基于观看方向是否旋转的标志信息。
10. 根据权利要求5所述的方法,其中,当关于所述叠加的类型的信息指示所述叠加的类型是在3D空间中被渲染的叠加类型时,所述与叠加有关的元数据进一步包括指示距单位球体的中心的距离的距离信息。
11. 根据权利要求5所述的方法,其中,当关于所述叠加的类型的信息指示所述叠加的类型是在球体表面中被渲染的叠加类型时,所述与叠加有关的元数据包括关于所述叠加的区域的深度的信息。
12. 根据权利要求11所述的方法,其中,指示所述叠加的区域的深度的信息指示所述叠加的渲染优先级。
13. 根据权利要求12所述的方法,其中,在多个叠加被重叠的情况下,对于所述多个叠

加当中的重叠区域,渲染与较小深度值相关联的叠加。

14.根据权利要求1所述的方法,其中,所述与叠加有关的元数据进一步包括指示所述叠加是否与参考点相关联的目标标志信息。

15.根据权利要求14所述的方法,其中,通过所述叠加提供由所述参考点指示的图像、文本、声音或补充信息,或者其中,渲染由所述参考点指示的位置的图像。

16.根据权利要求1所述的方法,其中,所述与叠加有关的元数据包括关于叠加的数量的信息,并且

其中,与所述叠加中的每一个相对应的叠加有关的信息被包括在所述与叠加有关的元数据中。

17.根据权利要求1所述的方法,其中,所述与叠加有关的元数据被包括在叠加定时元数据轨道中。

18.根据权利要求1所述的方法,其中,所述与叠加有关的元数据包括与每个均具有在视口中被渲染的叠加类型的叠加的顺序有关的信息。

19.一种由360度视频发送设备执行的360度图像数据处理方法,所述方法包括:

获取360度图像;

通过处理所述360度图像来导出图片;

生成与所述360度图像有关的元数据;

对所述图片进行编码;以及

执行用于所述编码的图片和所述元数据的存储或者传输的处理,

其中,所述元数据包括与叠加有关的元数据,并且

其中,所述与叠加有关的元数据包括关于所述叠加的区域的信息。

20.一种360度视频接收设备,所述设备包括:

接收处理器,所述接收处理器接收360度图像数据并且从所述360度图像数据中获取关于编码的图片的信息和元数据;

数据解码器,所述数据解码器基于关于所述编码的图片的信息来解码图片;以及

渲染器,所述渲染器基于所述元数据来渲染所述解码的图片和叠加,

其中,所述元数据包括与叠加有关的元数据,

其中,所述渲染器基于所述与叠加有关的元数据来渲染所述叠加,并且

其中,所述与叠加有关的元数据包括关于所述叠加的区域的信息。

360视频系统中的叠加处理方法及其设备

技术领域

[0001] 本发明涉及360视频(或360度视频)并且,更加具体地,涉及360视频系统中的叠加处理方法及其设备。

背景技术

[0002] 虚拟现实(VR)系统使用户感觉仿佛在电子投影环境中一样。可以改进用于提供VR的系统,以便于提供具有更高图片质量和空间声音的图像。VR系统允许用户以交互方式消费VR内容。

发明内容

[0003] 技术问题

[0004] 本发明的技术目的是提供一种360视频数据处理方法及其设备。

[0005] 本发明的另一个技术目的是提供一种用于发送360视频数据的元数据的方法和设备。

[0006] 本发明的又一个技术目的是提供一种用于360视频的叠加处理的方法和设备。

[0007] 本发明的又一个技术目的是提供一种用于发送用于360视频的叠加的元数据的方法和设备。

[0008] 技术解决方案

[0009] 根据本发明的示例性实施例,本文提供一种由360度视频接收设备执行的360度图像数据处理方法。该方法可以包括下述步骤:接收360度图像数据;从360度图像数据获取关于编码的图片的信息和元数据;基于关于编码的图片的信息来解码图片;以及基于元数据来渲染解码的图片和叠加,其中所述元数据可以包括与叠加有关的元数据,其中可以基于与叠加有关的元数据来渲染叠加,并且其中与叠加有关的元数据可以包括关于叠加的区域的信息。

[0010] 根据本发明的另一示例性实施例,提供一种360度视频接收设备。该设备可以包括:接收处理器,该接收处理器接收360度图像数据并且从360度图像数据中获取关于编码的图片的信息和元数据;数据解码器,该数据解码器基于关于编码的图片的信息来解码图片;以及渲染器,该渲染器基于元数据来渲染解码的图片和叠加,其中所述元数据可以包括与叠加有关的元数据,其中渲染器可以基于与叠加有关的元数据来渲染叠加,并且其中与叠加有关的元数据可以包括关于叠加的区域的信息。

[0011] 根据本发明的另一示例性实施例,提供一种由360度视频发送设备执行的360度图像数据处理方法。该方法可以包括以下步骤:获取360度图像;通过处理360度图像来导出图片;生成与360度图像有关的元数据;对图片进行编码;以及执行用于编码的图片和元数据的存储或者传输的处理,其中,所述元数据可以包括与叠加有关的元数据,并且其中与叠加有关的元数据可以包括关于叠加的区域的信息。

[0012] 根据本发明的又一示例性实施例,提供一种由360度视频发送设备执行的360度图

像数据处理方法。该设备可以包括：数据输入单元，该数据输入单元获取360度图像；投影处理器，该投影处理器通过处理360度图像来导出图片；元数据处理器，该元数据处理器生成与360度图像有关的元数据；数据编码器，该数据编码器对图片进行编码；以及传输处理器，该传输处理器执行用于编码的图片和元数据的存储或传输的处理，其中所述元数据可以包括与叠加有关的元数据，并且其中与叠加有关的元数据可以包括关于叠加的区域的信息。

[0013] 有益效果

[0014] 根据本发明，可以在支持使用陆地网络和互联网网络两者的下一代混合广播的环境中有效地发送VR内容(360内容)。

[0015] 根据本发明，当用户消费360内容时，可以提出用于提供交互体验的解决方案。

[0016] 根据本发明，当用户消费360内容时，可以提出一种用于执行信令使得可以准确地反映360内容提供商的意图的解决方案。

[0017] 根据本发明，当传递360内容时，可以提出一种用于有效地扩展传输容量并且允许传送(或传递)必要信息的解决方案。

[0018] 根据本发明，可以将叠加有效地提供给360视频，并且可以有效地显示基于用户视点的辅助信息。

[0019] 根据本发明，可以通过用于360视频的叠加来提供到特定目标的链接。

[0020] 根据本发明，可以通过叠加来提供用于有效的屏幕移位或辅助信息提供的链接。

[0021] 根据本发明，可以经由诸如ISO基本媒体文件格式(ISOBMFF)的基于国际标准化组织(ISO)的媒体文件格式等等有效地存储和发送与360度视频数据相对应的信令信息。

[0022] 根据本发明，可以经由诸如基于HTTP的动态自适应流(DASH)的基于超文本传输协议(HTTP)的自适应流等，来发送与360度视频数据相对应的信令信息。

[0023] 根据本发明，可以经由补充增强信息(SEI)消息或视频可用性信息(VUI)来存储和发送与360度视频数据相对应的信令信息，并且因此，总的传输效率可以被增强。

附图说明

[0024] 图1是图示根据本发明的用于提供360度视频的总体架构的视图。

[0025] 图2和图3是图示根据本发明的实施例的媒体文件的结构的视图。

[0026] 图4图示基于DASH的自适应流模型的整体操作的示例。

[0027] 图5是示意性地图示可应用本发明的360度视频传输装置的配置的视图。

[0028] 图6是示意性地图示可应用本发明的360度视频接收装置的配置的视图。

[0029] 图7是图示用于描述本发明的3D空间的飞行器主轴的概念的视图。

[0030] 图8图示应用360度视频的处理过程以及应用根据投影格式的按区域打包过程的2D图像。

[0031] 图9a和图9b图示根据本发明的投影格式。

[0032] 图10a和图10b图示根据本发明实施例的图块。

[0033] 图11示出根据本发明的示例性实施例的与360视频有关的元数据的示例。

[0034] 图12示出视点、观看位置和观看定向的概念的总体视图。

[0035] 图13示出根据本发明的用于提供3DoF+视频的示例性架构的总体视图。

[0036] 图14a和图14b是3DoF+端对端系统架构的示例。

- [0037] 图15示出示例性的直播上行链路流框架 (FLUS) 架构的总体视图。
- [0038] 图16示出3DoF+发送端的一般配置。
- [0039] 图17示出3DoF+接收端的一般配置。
- [0040] 图18至图20分别示出根据叠加类型的示例性叠加位置。
- [0041] 图21示出被存储在样本中的叠加轨道和显示时间信息的示例。
- [0042] 图22示出基于与叠加有关的元数据的360视频传输/接收方法的总体视图。
- [0043] 图23和图24是叠加被定位在球体区域上的情况的示例。
- [0044] 图25图示叠加被定位在当前视口上的情况。
- [0045] 图26是叠加被定位在球体或立方体内的3D空间中的情况的示例。
- [0046] 图27示出包括与叠加有关的元数据的示例性叠加元数据轨道。
- [0047] 图28是叠加被定位在视口上的情况的示例。
- [0048] 图29是叠加被定位在球体上的情况的示例。
- [0049] 图30是叠加被定位在3维 (3D) 空间上的情况的示例。
- [0050] 图31示出当叠加存在于球体内的3维 (3D) 空间中时的叠加的位置/大小/旋转。
- [0051] 图32示出叠加渲染属性的示例。
- [0052] 图33示出动态叠加元数据的示例性配置。
- [0053] 图34示出动态叠加元数据轨道和叠加媒体轨道链路信令的示例。
- [0054] 图35示出在叠加媒体轨道上的叠加元数据信令的示例。
- [0055] 图36和图37示出叠加媒体打包、投影和默认渲染信令的示例。
- [0056] 图38是凝视点计算的示例。
- [0057] 图39示出叠加渲染过程的示例。
- [0058] 图40示出根据叠加类型的示例性叠加几何图形生成。
- [0059] 图41示出根据叠加内容类型的示例性叠加纹理生成。
- [0060] 图42示出叠加渲染示例。
- [0061] 图43示出由根据本发明的360视频发送设备执行的360视频/图像数据处理方法的总体视图。
- [0062] 图44示出由根据本发明的360视频接收设备执行的360视频/图像数据处理方法的总体视图。

具体实施方式

[0063] 可以以各种形式修改本发明,并且将在附图中描述和图示其具体实施例。然而,实施例并不旨在限制本发明。在以下描述中使用的术语仅用于描述特定实施例,而不旨在限制本发明。只要清楚地以不同的方式理解,单数的表达包括复数的表达。诸如“包括”和“具有”的术语旨在指示存在在下面的描述中使用的特征、数量、步骤、操作、元件、组件或其组合,并且因此应理解一个或多个不同特征、数量、步骤、操作、元件、组件或其组合的存在或者添加的可能性没有被排除。

[0064] 另一方面,在本发明中描述的附图中的元件是为了方便地解释不同的特定功能而被独立地绘制,并不意味着该元件由独立的硬件或独立的软件来体现。例如,可以将元件中的两个或更多个元件组合以形成单个元件,或者可以将一个元件划分为多个元件。在不脱

离本发明的概念的情况下其中元件被组合和/或划分的实施例属于本发明。

[0065] 在下文中,将参考附图更详细地描述本公开的优选实施例。在下文中,在整个说明书中将使用相同的附图标记指代相同的组件,并且将省略相同组件的冗余描述。

[0066] 图1是图示根据本发明的用于提供360度视频的总体架构的视图。

[0067] 本发明提出一种提供360度内容以便向用户提供虚拟现实 (VR) 的方法。VR可以是指用于复制实际或虚拟环境的技术或那些环境。VR人为地为用户提供感官体验,并且因此用户可以体验电子投影的环境。

[0068] 360度内容是指用于实现和提供VR的内容,并且可以包括360度视频和/或360度音频。360度视频可以指的是提供VR所需的并且在所有方向上 (360度) 捕获或再现的视频或图像内容。在下文中,360视频可以指的是360度视频。360度视频可以指的是根据3D模型以各种形式在3D空间上表现的视频或图像。例如,可以在球面上表现360度视频。360度音频是用于提供VR的音频内容,并且可以指的是其音频生成源可以被识别为位于特定3D空间中的空间音频内容。可以生成、处理并且将360度内容发送给用户,并且用户可以使用360度内容来消费VR体验。360度视频可以被称为全向视频,而360度图像可以被称为全向图像。

[0069] 特别地,本发明提出一种用于有效地提供360度视频的方法。为了提供360度视频,可以通过一个或多个相机捕获360度视频。可以通过一系列处理来发送捕获的360度视频,并且接收侧可以将所发送的360度视频处理为原始360度视频并渲染360度视频。以这种方式,可以将360度视频提供给用户。

[0070] 具体地,用于提供360度视频的过程可以包括捕获过程、准备过程、传输过程、处理过程、渲染过程和/或反馈过程。

[0071] 捕获过程可以指的是通过一个或多个相机捕获针对多个视点的图像或视频的过程。图1中所示的图像/视频数据 (110) 可以通过捕获过程被生成。图1中 (110) 的每个平面可以表示每个视点的图像/视频。多个捕获的图像/视频可以被称为原始数据。可以在捕获过程中生成与捕获有关的元数据。

[0072] 为了捕获,可以使用用于VR的专用相机。当根据实施例提供关于由计算机生成的虚拟空间的360度视频时,可以不执行通过实际相机的捕获。在这种情况下,简单地生成有关的数据的过程可以代替捕获过程。

[0073] 准备过程可以是处理捕获的图像/视频和在捕获过程中生成的元数据的过程。在准备过程期间,捕获的图像/视频可以经历拼接过程、投影过程、按区域打包过程和/或编码过程。

[0074] 首先,每个图像/视频可以经历拼接过程。拼接过程可以是连接捕获的图像/视频以生成一个全景图像/视频或球形图像/视频的过程。

[0075] 随后,拼接的图像/视频可以经历投影过程。在投影过程中,可以将拼接的图像/视频投影到2D图像上。根据上下文,可以将2D图像称为2D图像帧。在2D图像上的投影可以被映射到2D图像。投影的图像/视频数据可以具有图1中的2D图像 (120) 的形式。

[0076] 投影在2D图像上的视频数据可以经历按区域打包过程,以便于提高视频编码效率。按区域打包可以指的是将投影在2D图像上的视频数据划分成区域进行处理的过程。在此,区域可以指在其上投影360度视频数据的2D图像的被划分的区域。根据实施例,可以通过均等地或任意地划分2D图像来获得区域。此外,在实施例中,可以根据投影方案来划分区

域。按区域打包过程是可选过程,并且在准备过程中可以被省略。

[0077] 根据实施例,处理过程可以包括旋转区域或在2D图像上重新排列区域以便于提高视频编码效率的过程。例如,可以旋转区域,使得区域的特定的边彼此接近地定位,以提高编码效率。

[0078] 根据实施例,处理过程可以包括增加或降低特定区域的分辨率以便区分化360度视频的区域的分辨率的过程。例如,可以将与360度视频中与相对更重要的区域相对应的区域的分辨率增加到高于其他区域的分辨率。可以通过视频编解码器对投影在2D图像上的视频数据或按区域打包的视频数据进行编码处理。

[0079] 根据一个实施例,准备过程可以进一步包括附加的编辑过程。在该编辑过程中,可以对投影之前和之后的图像/视频数据执行编辑。在准备过程中,也可以生成关于拼接/投影/编码/编辑的元数据。此外,可以生成关于投影在2D图像上的视频数据的初始视点或感兴趣区域(ROI)的元数据。

[0080] 传输过程可以是处理和发送已经经过准备过程的图像/视频数据和元数据的过程。可以执行根据任意传输协议的处理用于传输。可以通过广播网络和/或宽带来传递已经被处理用于传输的数据。这样的数据可以按需方式传递到接收侧。接收侧可以通过各种路径接收数据。

[0081] 处理过程可以指的是对接收的数据进行解码并且将投影的图像/视频数据重新投影到3D模型上的过程。在此过程中,可以将投影到2D图像上的图像/视频数据重新投影到3D空间上。根据上下文,此过程可以被称为映射或投影。在此,根据3D模型,图像/视频数据被映射到的3D模型可以具有不同的形式。例如,3D模型可以包括球体、立方体、圆柱体和角锥体。

[0082] 根据实施例,处理过程可以附加地包括编辑过程和放大(up-scaling)过程等。在编辑过程中,可以进一步执行对重新投影之前和之后的图像/视频数据的编辑。当图像/视频数据已经被减少时,可以通过在放大过程中放大样本来增大图像/视频数据的大小。可以根据需要执行通过缩小(down-scaling)来减小大小的操作。

[0083] 渲染过程可以指渲染和显示重新投影在3D空间上的图像/视频数据的过程。重新投影和渲染可以被组合并表示为在3D模型上的渲染。重新投影到3D模型上(或渲染到3D模型上)的图像/视频可以具有图1中所示的形式(130)。图1中所示的形式(130)对应于其中图像/视频被重新投影在3D球形模型上的情况。用户可以通过VR显示器查看渲染的图像/视频的区域。在此,用户观看的区域可以具有图1中所示的形式(140)。

[0084] 反馈过程可以指的是将可以在显示过程中获取的各种类型的反馈信息传递到传输侧的过程。可以通过反馈过程提供360度视频消费中的交互性。根据实施例,可以在反馈过程中将头部定向信息、表示由用户当前观看的区域的视口信息等传递到传输侧。根据实施例,用户可以与在VR环境中实现的对象进行交互。在这种情况下,可以在反馈过程中将关于交互的信息传递到传输侧或服务提供商。根据实施例,可以不执行反馈过程。

[0085] 头部定向信息可以指关于用户的头部的位置、角度、运动等的信息。基于此信息,可以计算关于由用户当前正在观看的360度视频中的区域的信息,即,视口信息。

[0086] 视口信息可以是关于由用户当前正在观看的360度视频中的区域的信息。可以通过视点信息执行凝视分析,以检查用户如何消费360度视频、用户凝视360度视频的哪个区

域、凝视该区域多长时间等。凝视分析可以在接收侧执行,并且其结果可以通过反馈信道被传递到传输侧。诸如VR显示器的设备可以基于用户头部的位置/方向、该设备所支持的关于垂直或水平视场(FOV)的信息等来提取视口区域。

[0087] 根据实施例,前述反馈信息不仅可以被发送到传输侧还可以在接收侧被消费。即,可以使用上述反馈信息执行在接收侧的解码、重新投影和渲染。例如,可以使用头部定向信息和/或视口信息仅对相对于由用户当前观看的区域的360度视频进行优先解码和渲染。

[0088] 这里,视口或视口区域可以指由用户正在观看的360度视频中的区域。视点是由用户正在观看的360度视频中的点,并且可以指视口区域的中心点。即,视口是在其中心具有视点的区域,并且该区域的尺寸和形状可以通过将在后面描述的FOV来确定。

[0089] 在上述用于提供360度视频的总体体系结构中,经历捕获/投影/编码/传输/解码/重新投影/渲染过程的图像/视频数据可以被称为360度视频数据。术语“360度视频数据”可以被用作包括与这样的图像/视频数据有关的元数据和信令信息的概念。

[0090] 为了存储和发送诸如上述音频和视频数据的媒体数据,可以定义标准化的媒体文件格式。根据实施例,媒体文件可以具有基于ISO BMFF (ISO基本媒体文件格式)的文件格式。

[0091] 图2和图3是图示根据本发明的实施例的媒体文件的结构的视图。

[0092] 根据本发明的媒体文件可以包括至少一个盒子(box)。这里,盒子可以是包括媒体数据或与媒体数据有关的元数据的数据块或对象。盒子可以具有层级结构,并且因此可以对数据进行分类,并且媒体文件可以具有适合于存储和/或传输大容量媒体数据的格式。此外,媒体文件可以具有允许用户容易地访问诸如移动到媒体内容的特定点的媒体信息的结构。

[0093] 根据本发明的媒体文件可以包括ftyp盒子、moov盒子和/或mdat盒子。

[0094] ftyp盒子(文件类型盒子)可以提供关于相应媒体文件的文件类型或与兼容性有关的信息。该ftyp盒子可以包括关于相应媒体文件的媒体数据的配置版本信息。解码器可以参考ftyp盒子识别相应的媒体文件。

[0095] moov盒子(电影盒子)可以是包括关于相应媒体文件的媒体数据的元数据的盒子。moov盒子可以用作所有元数据的容器。moov盒子可能是与元数据相关的盒子中的最高层。根据一个实施例,媒体文件中仅可以存在一个moov盒子。

[0096] mdat盒子(媒体数据盒子)可以是包含相应媒体文件的实际媒体数据的盒子。媒体数据可以包括音频样本和/或视频样本。mdat盒子可用作包含此类媒体样本的容器。

[0097] 根据一个实施例,前述的moov盒子可以进一步包括mvhd盒子、trak盒子和/或mvex盒子作为下位盒子。

[0098] mvhd盒子(电影报头盒子)可以包含与相应媒体文件中包含的媒体数据的媒体呈现有关的信息。即,mvhd盒子可以包括诸如相应媒体呈现的媒体生成时间、改变时间、时间标准和时段的信息。

[0099] trak盒子(轨道盒子)可以提供有关相应媒体数据的轨道的信息。trak盒子可以包括诸如关于音频轨道或视频轨道的与流有关的信息、与呈现有关的信息和与访问有关的信息的信息。根据轨道的数量,可以存在多个trak盒子。

[0100] trak盒子还可以包括作为下位盒子的tkhd盒子(轨道报头盒子)。tkhd盒子可以包

含有关trak盒子指示的轨道的信息。该tkhd盒子可以包括诸如相应轨道的生成时间、改变时间和轨道标识符的信息。

[0101] mvex盒子(电影扩展盒子)可以指示相应的媒体文件可能具有moof盒子,稍后将对此进行描述。为了识别特定轨道的所有媒体样本,可能需要扫描moof盒子。

[0102] 根据一个实施例,根据本发明的媒体文件可以被划分为多个分片(fragment)(200)。因此,媒体文件可以被分段来存储或发送。媒体文件的媒体数据(mdat盒子)可以被划分成多个分片,并且每个分片可以包括moof盒子和被划分的mdat盒子。根据实施例,可能需要ftyp盒子和/或moov盒子的信息以使用分片。

[0103] moof box(电影分片盒子)可以提供有关相应分片的媒体数据的元数据。moof盒子可以是与相应分片的元数据有关的盒子之中的最高层的盒子。

[0104] mdat盒子(媒体数据盒子)可以包含如上所述的实际媒体数据。mdat盒子可以包括与对应于其的每个分片相对应的媒体数据的媒体样本。

[0105] 根据实施例,前述的moof盒子可以进一步包括mfhd盒子和/或traf盒子作为下位盒子。

[0106] mfhd盒子(电影分片报头盒子)可以包含有关划分的分片之间的相关性的信息。mfhd盒子可以通过包含序列号来指示相应分片的划分媒体数据的顺序。此外,可以使用mfhd盒子检查在划分的数据中是否有丢失的数据。

[0107] traf盒子(轨道分片盒子)可以包含有关相应轨道分片的信息。traf盒子可以提供有关包含在相应分片中的划分的轨道分片的元数据。traf盒子可以提供元数据,使得相应的轨道分片中的媒体样本可以被解码/再现。根据轨道分片的数量,可以存在多个traf盒子。

[0108] 根据实施例,前述的traf盒子可以进一步包括作为下位盒子的tfhd盒子和/或trun盒子。

[0109] tfhd盒子(轨道分片报头盒子)可以包含相应轨道分片的报头信息。tfhd盒子可以提供诸如由上述的traf盒子指示的轨道分片的媒体样本的基本样本大小、时段、偏移和标识符的信息。

[0110] trun盒子(轨道分片运行盒子)可以包含与相应轨道分片有关的信息。trun盒子可以包括诸如每个媒体样本的时段、大小和再现时间的信息。

[0111] 前述媒体文件及其分片可以被处理成片段(segment)并被发送。片段可以包括初始化片段和/或媒体片段。

[0112] 所图示的实施例(210)的文件可以包括除了媒体数据之外的与媒体解码器初始化有关的信息。例如,该文件可以对应于上述初始化片段。初始化片段可以包括上述的ftyp盒子和/或moov盒子。

[0113] 所图示的实施例(220)的文件可以包括上述分片。例如,该文件可以对应于上述媒体片段。媒体片段可以进一步包括styp盒子和/或sidx盒子。

[0114] styp盒子(片段类型盒子)可以提供用于识别划分的分片的媒体数据的信息。styp盒子可以用作上述划分的分片的ftyp盒子。根据实施例,styp盒子可以具有与ftyp盒子相同的格式。

[0115] sidx盒子(片段索引盒子)可以提供指示划分的分片的索引的信息。因此,可以指

示划分的分片的顺序。

[0116] 根据实施例(230),可以进一步包括ssix盒子。ssix盒子(子片段索引盒子)可以提供指示在将片段划分为子片段时子片段索引的信息。

[0117] 媒体文件中的盒子可以包括基于盒子或FullBox的更多扩展信息,如所图示的的实施例(250)中所示。在本实施例中,size字段和largesize字段可以以字节为单位表示相应盒子的长度。version字段可以指示相应盒子格式的的版本。type字段可以指示相应盒子的类型或标识符。flag字段可以指示与相应盒子相关联的标志。

[0118] 图4图示基于DASH的自适应流模型的整体操作的示例。根据所图示的实施例(400)的基于DASH的自适应流模型图示HTTP服务器和DASH客户端之间的操作。在此,基于HTTP的动态自适应流(DASH)是用于支持基于HTTP的自适应流的协议,并且可以根据网络状态动态地支持流。因此,可以无缝地再现AV内容。

[0119] 首先,DASH客户端可以获取MPD。MPD可以从诸如HTTP服务器的服务提供商传递。DASH客户端可以使用MPD中描述的片段访问信息从服务器请求片段。在此,可以鉴于网络状况来执行该请求。

[0120] 在获取片段之后,DASH客户端可以在媒体引擎中处理该片段,并且可以在屏幕上显示该片段。鉴于再现时间和/或实时的网络状态(自适应流),DASH客户端可以请求并获取必要的片段。因此,可以无缝地再现内容。

[0121] 媒体呈现描述(MPD)是包括用于允许DASH客户端动态获取片段的详细信息的文件,并且可以以XML格式表达。

[0122] DASH客户端控制器可以鉴于网络状态而生成命令以请求MPD和/或片段。另外,控制器可以控制所获取的信息以在诸如媒体引擎的内部块中使用。

[0123] MPD解析器可以实时解析所获取的MPD。因此,DASH客户端控制器可以生成命令以获取所需的片段。

[0124] 片段解析器可以实时解析所获取的片段。根据片段中包括的多条信息,包括媒体引擎在内的内部块可以执行某些操作。

[0125] HTTP客户端可以从HTTP服务器请求所需的MPD和/或片段。HTTP客户端还可以将从服务器获取的MPD和/或片段传递到MPD解析器或片段解析器。

[0126] 媒体引擎可以使用片段中包括的媒体数据在屏幕上显示内容。在此,可以使用多条MPD的信息。

[0127] DASH数据模型可以具有层级结构(410)。MPD可以描述媒体呈现。MPD可以描述形成媒体呈现的多个时段的时间序列。时段可以代表媒体内容的一个区段。

[0128] 在一个区段中,多条数据可以被包括在自适应集中。自适应集可以是可以彼此交换的多个媒体内容组件的集合。自适应集可以包括表示(representation)的集合。表示可以对应于媒体内容组件。在一个表示中,内容可以在时间上划分为多个片段,这可以用于适当的可访问性和传递。可以提供每个片段的URL以使得能够访问每个片段。

[0129] MPD可以提供与媒体呈现有关的信息,并且时段元素、自适应集元素和呈现元素可以分别描述时段、自适应集和呈现。表示可以被划分成子表示,并且子表示元素可以描述子表示。

[0130] 可以定义公共属性/元素,可以将其应用于(包括在)自适应集、表示、子表示等等

中。在公共属性/元素中,可能存在必要属性和/或补充属性。

[0131] 必要属性可以是包括在处理与媒体呈现有关的数据时被认为必不可少的元素的信息。补充属性可以是包括可被用于处理与媒体呈现有关的数据的元数据的信息。当经由MPD传递时,可以以必要属性和/或补充属性来定义和传递在以下实施例中描述的描述符。

[0132] 图5是示意性地图示可应用本发明的360度视频传输装置的配置的视图。

[0133] 根据本发明的360度视频传输装置可以执行与上述准备过程和传输过程有关的操作。360度视频传输装置可以包括数据输入单元、拼接器、投影处理器、按区域打包处理器(未示出)、元数据处理器、(传输侧)反馈处理器、数据编码器、封装处理器、传输处理器和/或发射器作为内部/外部元件。

[0134] 数据输入单元可以接收针对各个视点的捕获图像/视频。各个视点的图像/视频可以由一个或多个相机捕获的图像/视频。此外,数据输入单元可以接收在捕获过程中生成的元数据。数据输入单元可以将所接收的针对视点的图像/视频转发给拼接器,并且将在捕获过程中生成的元数据转发给信令处理器。

[0135] 拼接器可以对捕捉到的视点的图像/视频执行拼接操作。拼接器可以将拼接的360度视频数据转发到投影处理器。拼接器可以从元数据处理器接收必要的元数据,并且根据需要将元数据用于拼接操作。拼接器可以将将在拼接过程中生成的元数据转发到元数据处理器。拼接过程中的元数据可以包括诸如表示是否已经执行拼接的信息以及拼接类型的信息。

[0136] 投影处理器可以将拼接的360度视频数据投影到2D图像上。投影处理器可以根据稍后将描述的各种方案来执行投影。投影处理器可以考虑每个视点的360度视频数据的深度来执行映射。投影处理器可以从元数据处理器接收投影所需的元数据,并根据需要将元数据用于投影操作。投影处理器可以将将在投影过程中生成的元数据转发到元数据处理器。在投影处理器中生成的元数据可以包括投影方案类型等。

[0137] 按区域打包处理器(未示出)可以执行上述按区域打包过程。即,按区域打包处理器可以执行将投影的360度视频数据划分为多个区域并旋转和重新排列区域或改变每个区域的分辨率的过程。如上所述,按区域打包过程是可选的,并且因此当不执行按区域打包时,可以省略按区域打包处理器。按区域打包处理器可以从元数据处理器接收按区域打包所需的元数据,并根据需要将元数据用于按区域打包操作。按区域打包处理器可以将将在按区域打包过程中生成的元数据转发到元数据处理器。在按区域打包的处理器中生成的元数据可以包括每个区域的旋转度、大小等。

[0138] 根据实施例,前述的拼接器、投影处理器和/或按区域打包处理器可以被集成到单个硬件组件中。

[0139] 元数据处理器可以处理可以在捕获过程、拼接过程、投影过程、按区域打包过程、编码过程、封装过程和/或传输过程中生成的元数据。元数据处理器可以使用此类元数据生成与360度视频有关的元数据。根据实施例,元数据处理器可以以信令表的形式生成与360度视频有关的元数据。根据信令上下文,与360度视频有关的元数据也可以被称为元数据或与360度视频有关的信令信息。此外,元数据处理器可以根据需要将获取或生成的元数据转发到360度视频传输装置的内部元件。元数据处理器可以将360度视频有关的元数据转发到数据编码器、封装处理器和/或传输处理器,使得可以将360度视频有关的元数据发送到接

收侧。

[0140] 数据编码器可以对投影在2D图像上的360度视频数据和/或按区域打包的360度视频数据进行编码。360度视频数据可以以各种格式被编码。

[0141] 封装处理器可以以文件格式封装编码的360度视频数据和/或与360度视频有关的元数据。在此,可以从元数据处理器接收与360度视频有关的元数据。封装处理器可以以诸如ISOBMFF、CFF等的文件格式封装数据,或者将数据处理为DASH片段等。封装处理器可以将与360度视频有关的元数据包括在文件格式中。例如,与360度视频有关的元数据可以被包括在具有ISOBMFF中的各种级别的盒子中,或者可以被包括为文件中的单独轨道的数据。根据实施例,封装处理器可以将与360度视频有关的元数据封装到文件中。传输处理器可以根据文件格式对封装的360度视频数据进行传输处理。传输处理器可以根据任意传输协议来处理360度视频数据。用于传输的处理可以包括用于在广播网络上进行传递的处理和用于在宽带上进行传递的处理。根据实施例,传输处理器可以从元数据处理器接收360度视频有关的元数据以及360度视频数据,并且对360度视频有关元数据执行传输处理。

[0142] 发射器可以通过广播网络和/或宽带来发送为了传输而处理的360度视频数据和/或与360度视频有关的元数据。发射器可以包括用于通过广播网络传输的元件和/或用于通过宽带传输的元件。

[0143] 根据本发明的360度视频传输装置的实施例,360度视频传输装置还可以包括数据存储单元(未示出)作为内部/外部元件。在将编码的360度视频数据和/或与360度视频有关的元数据传递到传输处理器之前,数据存储单元可以存储编码的360度视频数据和/或与360度视频有关的元数据。这样的数据可以以诸如ISOBMFF的文件格式存储。尽管在实时发送360度视频时可能不需要数据存储单元,但是在通过按需、NRT(非实时)、宽带等传递封装的360度数据时,封装的360度数据在特定的时间段内可以被存储在数据存储单元,并且然后被发送。

[0144] 根据本发明的360度视频传输装置的另一实施例,360度视频传输装置可以进一步包括(传输侧)反馈处理器和/或网络接口(未示出)作为内部/外部元件。网络接口可以根据本发明的360度视频接收装置接收反馈信息,并将该反馈信息转发到传输侧反馈处理器。传输侧反馈处理器可以将反馈信息转发给拼接器、投影处理器、按区域打包处理器、数据编码器、封装处理器、元数据处理器和/或传输处理器。根据实施例,反馈信息可以被传递到元数据处理器,并且然后被传递到每个内部元件。已经接收到反馈信息的内部元件可以在随后的360度视频数据处理中反映反馈信息。

[0145] 根据本发明的360度视频传输装置的另一实施例,按区域打包处理器可以旋转区域并将旋转的区域映射在2D图像上。在此,区域可以沿不同的方向以不同的角度旋转并映射到2D图像上。可以考虑投影之前球面上的360度视频数据的邻近部分和拼接部分来执行区域旋转。可以通过与360度视频有关的元数据来用信号发送关于区域旋转的信息,即,旋转方向、角度等。根据本发明的360度视频传输装置的另一实施例,数据编码器可以针对各个区域不同地执行编码。数据编码器可以以高质量编码特定区域并且可以以低质量编码其他区域。传输侧反馈处理器可以将360度视频接收装置接收到的反馈信息转发到数据编码器,使得数据编码器可以使用针对各个区域而区分的编码方法。例如,传输侧反馈处理器可以将接收侧接收到的视口信息转发到数据编码器。数据编码器可以以比其他区域更高

的质量(UHD等)对包括由视口信息指示的区域的区域进行编码。

[0146] 根据本发明的360度视频传输装置的另一实施例,传输处理器可以针对各个区域不同地执行用于传输的处理。传输处理器可以将不同的传输参数(调制阶数、编码率等)应用于各个区域,使得传递至各个区域的数据具有不同的鲁棒性。

[0147] 这里,传输侧反馈处理器可以将360度视频接收装置接收到的反馈信息转发到传输处理器,使得传输处理器可以执行针对各个区域而区分的传输处理。例如,传输侧反馈处理器可以将接收侧接收到的视口信息转发给传输处理器。传输处理器可以对包括由视口信息指示的区域的区域执行传输处理,使得这些区域具有比其他区域更高的鲁棒性。

[0148] 根据本发明的360度视频传输装置的上述内部/外部元件可以是硬件元件。根据实施例,内部/外部元件可以被改变、省略、由其他元件代替或被集成。

[0149] 图6是示意性地图示可应用本发明的360度视频接收装置的配置的视图。

[0150] 根据本发明的360度视频接收装置可以执行与上述处理过程和/或渲染过程有关的操作。360度视频接收装置可以包括作为内部/外部元件的接收器、接收处理器、解封装处理器、数据解码器、元数据解析器、(接收侧)反馈处理器、重新投影处理器和/或渲染器。信令解析器可以被称为元数据解析器。

[0151] 接收器可以接收从根据本发明的360度视频传输装置发送的360度视频数据。接收器可以根据通过其发送360度视频数据的信道,通过广播网络或宽带来接收360度视频数据。

[0152] 接收处理器可以根据传输协议对接收到的360度视频数据执行处理。接收处理器可以执行上述传输处理器的处理的逆过程,使得逆过程对应于在传输侧执行的传输处理。接收处理器可以将获取的360度视频数据转发到解封装处理器,并且将获取的360度视频有关元数据转发到元数据解析器。接收处理器获取的与360度视频有关的元数据可以具有信令表的形式。

[0153] 解封装处理器可以以从接收处理器接收到的文件格式对360度视频数据进行解封装。解封装处理器可以通过将解封装以ISOBMFF等的文件来获取360度视频数据和与360度视频有关的元数据。解封装处理器可以将获取的360度视频数据转发到数据解码器,并将获取的360度视频有关元数据转发到元数据解析器。由解封装处理器获取的与360度视频有关的元数据可以具有文件格式中的盒子或轨道的形式。解封装处理器可以根据需要从元数据解析器接收解封装所需的元数据。

[0154] 数据解码器可以解码360度视频数据。数据解码器可以从元数据解析器接收解码所需的元数据。可以将在数据解码过程中获取的与360度视频相关的元数据转发到元数据解析器。

[0155] 元数据解析器可以解析/解码与360度视频有关的元数据。元数据解析器可以将获取的元数据转发到数据解封装处理器、数据解码器、重新投影处理器和/或渲染器。

[0156] 重新投影处理器可以对解码的360度视频数据执行重新投影。重新投影处理器可以在3D空间上重新投影360度视频数据。取决于3D模型,3D空间可能具有不同的形式。重新投影处理器可以从元数据解析器接收重新投影所需的元数据。例如,重新投影处理器可以从元数据解析器接收关于所使用的3D模型的类型的信息及其详细信息。根据实施例,重新投影处理器可以使用重新投影所需的元数据仅将与3D空间的特定区域相对应的360度视频

数据重新投影到3D空间上。

[0157] 渲染器可以渲染重新投影的360度视频数据。如上所述,在3D空间上的360度视频数据的重新投影可以表示为在3D空间上的360度视频数据的渲染。当以这种方式同时发生两个过程时,重新投影处理器和渲染器可以被集成,并且渲染器可以执行这些过程。根据实施例,渲染器可以根据用户的视点信息仅渲染由用户观看的部分。

[0158] 用户可以通过VR显示器等观看渲染的360度视频的一部分。VR显示器是再现360度视频的设备,并且可以被包括在360度视频接收装置(系统)中或作为单独的设备(未系统)连接到360度视频接收装置。

[0159] 根据本发明的360度视频接收装置的实施例,360度视频接收装置可以进一步包括(接收侧)反馈处理器和/或网络接口(未示出)作为内部/外部元件。接收侧反馈处理器可以从渲染器、重新投影处理器、数据解码器、解封装处理器和/或VR显示器获取反馈信息,并处理该反馈信息。反馈信息可以包括视口信息、头部定向信息、凝视信息等。网络接口可以从接收侧反馈处理器接收反馈信息,并将该反馈信息发送到360度视频传输装置。

[0160] 如上所述,反馈信息不仅可以被发送到传输侧,还可以在接收侧被消费。接收侧反馈处理器可以将获取的反馈信息转发到360度视频接收装置的内部元件,使得反馈信息反映在诸如渲染的过程中。接收侧反馈处理器可以将反馈信息转发到渲染器、重新投影处理器、数据解码器和/或解封装处理器。例如,渲染器可以使用反馈信息来优先渲染用户观看的区域。另外,解封装处理器和数据解码器可以优先地解封装和解码正被用户观看或将被用户观看的区域。

[0161] 根据本发明的360度视频接收装置的上述内部/外部元件可以是硬件元件。根据实施例,内部/外部元件可以被改变、省略、由其他元件代替或被集成。根据实施例,附加元件可以被添加到360度视频接收装置。

[0162] 本发明的另一方面可以涉及一种用于发送360度视频的方法和用于接收360度视频的方法。根据本发明的用于发送/接收360度视频的方法可以由上述的360度视频传输/接收装置或其实施例执行。

[0163] 上述360度视频传输/接收装置和传输/接收方法的实施例和装置的内部/外部元件的实施例可以被组合。例如,投影处理器的实施例和数据编码器的实施例可以被组合以生成与情况数量一样多的360度视频传输装置的实施例。以这种方式组合的实施例也包括在本发明的范围内。

[0164] 图7是图示用于描述本发明的3D空间的飞行器主轴的概念的视图。在本发明中,飞行器主轴的概念可以用于表示3D空间中的特定点、位置、方向、间隔、区域等。也就是说,在本发明中,飞行器主轴的内容可以被用于描述投影之前或重新投影之后的3D空间并为此执行信令。根据一个实施例,可以使用使用X、Y和Z轴或球面坐标的概念的方法。

[0165] 飞行器可以自由地三维旋转。构成三维的轴称为俯仰轴、偏航轴和横滚轴。在说明书中,这些可以被称为俯仰、偏航和横滚或者俯仰方向、偏航方向和横滚方向。

[0166] 俯仰轴可以指的是飞行器的前端向上和向下旋转的方向的基础的轴。在所图示的飞行器主轴的概念中,俯仰轴可以指连接飞行器机翼的轴。

[0167] 偏航轴可以指的是飞行器的前端左右旋转的方向的基础的轴。在所图示的飞行器主轴的概念中,偏航轴可以指的是将飞行器的顶部连接至底部的轴。在图示的飞行器主轴

中,横滚轴可以指的是将前端连接到飞行器的尾部的轴,并且沿横滚方向的旋转可以指的是基于横滚轴的旋转。如上所述,可以使用俯仰、偏航和横滚的概念来描述本发明中的3D空间。

[0168] 如上所述,为了增强视频编码效率,可以对投影在2D图像上的视频数据进行按区域打包。按区域打包可以指的是按区域处理投影在2D图像上的视频数据的过程。这里,区域可以指的是在其上投影360度视频数据的2D图像的划分区域。可以根据投影方案来划分2D图像的划分区域。2D图像可以被称为视频帧或帧。

[0169] 本发明提出关于根据投影方案的按区域打包过程的元数据以及用信号发送该元数据的方法。可以基于元数据有效地执行按区域打包过程。

[0170] 图8图示应用360度视频的处理过程以及应用根据投影格式的按区域打包过程的2D图像。在图8中,(a)图示处理输入的360度视频数据的过程。参考图8的(a),可以根据各种投影方案将来自视点的输入的360度视频数据拼接并投影在3D投影结构上,并且可以将投影在3D投影结构上的360度视频数据表示为2D图像。即,可以拼接360度视频数据,并且可以将其投影到2D图像中。将360度视频数据投影到其中的2D图像可以被称为投影帧。投影帧可以经受上述的按区域打包过程。具体地,可以处理投影帧,使得可以将包括投影帧上的投影360度视频数据的区域划分为多个区域,并且可以旋转或重新排列每个区域,或者可以改变每个区域的分辨率。即,按区域打包过程可以指示将投影帧映射到一个或多个打包帧的过程。可以可选地执行按区域打包过程。当不应用按区域打包过程时,打包帧和投影帧可能相同。当应用按区域的打包过程时,投影帧的每个区域可以被映射到打包帧的区域,并且可以导出指示映射到投影帧的每个区域的打包帧的区域的区域的位置、形状和大小的元数据。

[0171] 在图8中,图8(b)和8(c)图示将投影帧的每个区域映射到打包帧的区域的示例。参考图8的(b),可以根据全景投影方案将360度视频数据投影到2D图像(或帧)上。投影帧的顶部、中间和底部区域可以经由按区域打包被重新排列为如右图中所示。这里,顶部区域可以表示2D图像上全景的顶部区域,中间区域可以表示2D图像上全景的中间区域,并且底部区域可以表示2D图像上全景的底部区域。参考图8的(c),可以根据立方体投影方案将360度视频数据投影到2D图像(或帧)上。投影帧的前部、后部、顶部、底部、右侧和左侧区域可以经由按区域打包重新排列为如右图所示。在此,前部区域可以表示2D图像上的立方体的前部区域,而后部区域可以表示2D图像上的立方体的后部区域。顶部区域可以表示2D图像上的立方体的顶部区域,而底部区域可以表示2D图像上的立方体的底部区域。右侧区域可以表示2D图像上的立方体的右侧区域,并且左侧区域可以表示2D图像上的立方体的左侧区域。

[0172] 在图8中,(d)图示用于投影360度视频数据的各种3D投影格式。参考图8的(d),3D投影格式可以包括四面体、立方体、八面体、十二面体和二十面体。图8的(d)中所示的2D投影可以表示与从根据3D投影格式的360度视频数据的投影而产生的2D图像相对应的投影帧。

[0173] 提供前述投影格式是出于说明性目的,并且根据本发明,可以使用以下各种投影格式(或投影方案)中的一些或全部。例如,可以通过元数据的投影格式字段来指示用于360度视频的投影格式。

[0174] 图9a和图9b图示根据本发明的投影格式。

[0175] 在图9a中,(a)图示等矩形投影格式。当使用等矩形投影格式时,可以将球面上的

点 $(r, \theta_0, 0)$, 即, $\theta = \theta_0$ 和 $\varphi = 0$ 映射到2D图像的中心像素。此外, 可以假定前置相机的主要点是球面上的点 $(r, 0, 0)$, 并且 $\varphi_0 = 0$ 。因此, 可以将XY坐标系上的转换值 (x, y) 通过以下等式将其转换为2D图像上的像素 (X, Y) 。

[0176] [等式1]

$$[0177] \quad X = K_x * x + X_O = K_x * (\theta - \theta_0) * r + X_O$$

$$[0178] \quad Y = -K_y * y - Y_O$$

[0179] 当2D图像的左上像素被定位在XY坐标系上的 $(0, 0)$ 时, x 轴的偏移和 y 轴的偏移可以由以下等式表示。

[0180] [等式2]

$$[0181] \quad X_O = K_x * \pi * r$$

$$[0182] \quad Y_O = -K_y * \pi / 2 * r$$

[0183] 使用这些偏移量, 可以如下修改用于转换到XY坐标系的等式。

[0184] [等式3]

$$[0185] \quad X = K_x x + X_O = K_x * (\pi + \theta - \theta_0) * r$$

$$[0186] \quad Y = -K_y y - Y_O = K_y * (\pi / 2 - \varphi) * r$$

[0187] 例如, 当 $\theta_0 = 0$ 时, 即, 当2D图像的中心像素指示球面上的与 $\theta = 0$ 相对应的数据时, 球面可以映射到相对于2D图像上的 $(0, 0)$ 由宽度 $= 2K_x \pi r$ 和高度 $= K_x \pi r$ 定义的区域。与球面上的 $\varphi = \pi / 2$ 相对应的数据可以被映射到2D图像的整个顶侧。此外, 可以将与球面上的 $(r, \pi / 2, 0)$ 相对应的数据映射到2D图像上的点 $(3\pi K_x r / 2, \pi K_x r / 2)$ 。

[0188] 接收侧可以将2D图像上的360度视频数据重新投影到球面上, 这可以由以下等式表示以进行转换。

[0189] [等式4]

$$[0190] \quad \theta = \theta_0 + X / K_x * r - \pi$$

$$[0191] \quad \varphi = \pi / 2 - Y / K_y * r$$

[0192] 例如, 可以将2D图像上由XY坐标 $(K_x \pi r, 0)$ 定义的像素重新投影到球面上的由 $\theta = \theta_0$ 和 $\varphi = \pi / 2$ 定义的点中。

[0193] 在图9a中, (b) 图示立方体投影格式。例如, 可以在球面上表示拼接的360度视频数据。投影处理器可以将360度视频数据划分为立方体形状, 并且可以将360度视频数据投影到2D图像上。球面上的360度视频数据可以被投影在与立方体的每个面相对应的2D图像上, 如图9a的 (b) 中的左图或右图所示。

[0194] 在图9a中, (c) 图示圆柱投影格式。假设可以在球面上表示拼接的360度视频数据, 则投影处理器可以将360度视频数据划分为圆柱形状, 并将360度视频数据投影到2D图像上。球面上的360度视频数据可以被投影在与圆柱体的侧面、顶面和底面对应的2D图像上, 如图9a的 (c) 中的左图或右图所示。

[0195] 在图9a中, (d) 图示基于图块的投影格式。当使用基于图块的投影方案时, 投影处理器可以将球面上的360度视频数据划分为一个或多个子区域, 以被投影到2D图像上, 如图9a的 (d) 中所示。子区域可以被称为图块。

[0196] 在图9b中, (e) 图示金字塔投影格式。假设可以在球面上表示拼接的360度视频数据, 则投影处理器可以将360度视频数据视为金字塔形状, 并且可以将360度视频数据划分为要投影到2D图像上的面。球面上的360度视频数据可以被投影在与金字塔的正面和金字塔的四个侧面相对应的2D图像上, 所述四个侧面包括左上面、左下面、右上面和右下面, 如图9b的(e)中的左图或右图所示。在此, 底表面可以是包括由面对正面的相机获取的数据的区域。在此, 正面可以是包括由前置相机获取的数据的区域。

[0197] 在图9b中, (f) 图示全景投影格式。当使用全景投影格式时, 投影处理器可以仅将球面上的360度视频数据的侧面投影到2D图像上, 如图9b的(f)所示。除了没有顶面和底面外, 此方案可能与圆柱投影方案相同。

[0198] 根据本发明的实施例, 可以在没有拼接的情况下执行投影。在图9b中, (g) 图示不进行拼接而执行投影的情况。当在没有拼接的情况下执行投影时, 投影处理器可以将360度视频数据按原样投影到2D图像上, 如图9b的(g)中所示。在这种情况下, 在没有拼接的情况下, 可以将各个相机获取的图像如原样投影到2D图像上。

[0199] 参考图9b的(g), 在没有拼接的情况下两个图像可以被投影到2D图像上。每个图像可以通过球形相机(或鱼眼相机)的每个传感器获取的鱼眼图像。如上所述, 接收侧可以拼接由相机传感器获取的图像数据, 并且可以将拼接的图像数据映射到球面上, 从而呈现球形视频, 即, 360度视频。

[0200] 图10a和图10b图示根据本发明的实施例的图块。

[0201] 投影到2D图像上的360度视频数据或经历按区域打包的360度视频数据可以被划分成一个或多个图块。图10a示出一个2D图像被划分为16个图块。这里, 如上所述, 2D图像可以是投影帧或打包帧。在根据本发明的360度视频传输装置的另一个实施例中, 数据编码器可以独立地编码每个图块。

[0202] 可以区分上述的按区域打包和分块(tiling)。上述的按区域打包可以指的是将投影在2D图像上的360度视频数据划分为多个区域并处理划分的区域以提高编译效率或调整分辨率的过程。分块可以指的是其中数据编码器将投影的或打包的帧划分为图块并且独立地编码每个图块的过程。当提供360度视频时, 用户不会同时消费360度视频的所有部分。分块可以允许用户仅将与诸如由用户当前观看的视口的重要部分或特定部分相对应的图块发送到接收侧, 或者以有限的带宽消费图块。与一次处理整个360度视频数据的情况相比, 分块能够有效利用有限的带宽, 并且使接收侧能够减轻操作负荷。

[0203] 因为区分区域和图块, 所以这两个区域不必相同。然而, 在一个实施例中, 区域和图块可以指代相同的区域。在实施例中, 当根据图块执行按区域的打包时, 区域和图块可以是相同的。此外, 在根据投影方案的每个面和每个区域相同的实施例中, 根据投影方案, 每个面、每个区域和每个图块可以指代相同的区域。根据上下文, 区域也可以被称为VR区域, 并且图块也可以被称为图块区域。

[0204] 感兴趣区域(ROI)可以指360度内容提供商提议的来自用户的感兴趣区域。当产生360度视频时, 360度内容提供商可以考虑用户感兴趣的特定区域来产生360度视频。在实施例中, ROI可以对应于其中再现360度视频的内容的重要部分的区域。

[0205] 在根据本发明的360度视频传输/接收装置的另一实施例中, 接收侧的反馈处理器可以提取并收集视口信息, 并且可以将视口信息发送到传输侧的反馈处理器。在该过程中,

可以使用双方的网络接口来发送视口信息。图10a示出2D图像中的视口(1000)。在此,视口可以在2D图像中的九个图块上延伸。

[0206] 在这种情况下,360度视频传输装置可以进一步包括分块系统。在实施例中,分块系统可以位于数据编码器之后(在图10b中),可以被包括在上述数据编码器或传输处理器中,或者可以被包括作为360度视频传输装置中的单独的内部/外部元件。

[0207] 分块系统可以从传输侧的反馈处理器接收视口信息。分块系统可以选择性地仅发送包括视口区域的图块。仅可以发送图10a中所示的2D图像中的总共16个图块当中的包括视口区域(1000)的9个图块。在此,因为视口区域根据用户而变化,所以分块系统可以经由宽带以单播方式发送图块。

[0208] 在这种情况下,传输侧的反馈处理器可以将视口信息发送到数据编码器。数据编码器可以以比其他图块更高的质量对包括视口区域的图块进行编码。

[0209] 此外,传输侧的反馈处理器可以将视口信息发送到元数据处理器。元数据处理器可以将与视口区域有关的元数据发送到360度视频传输装置的每个内部元件,或者可以将元数据包括在360度视频有关的元数据中。

[0210] 通过使用这种分块方法,能够节省传输带宽并针对每个图块不同地执行处理,从而实现有效的数据处理/传输。

[0211] 与视口区域有关的前述实施例可以类似地应用于除视口区域之外的特定区域。例如,可以对被确定为其中用户通过前述的凝视分析而感兴趣的区域、ROI以及当通过VR显示器用户观看360度视频时首先再现的区域(初始视点)的区域同等地执行在视口区域上执行的处理。

[0212] 在根据本发明的360度视频传输装置的另一实施例中,传输处理器可以针对每个图块不同地执行传输处理。传输处理器可以将不同的传输参数(调制阶数或编译率)应用于每个图块,使得改变经由每个图块传递的数据的鲁棒性。

[0213] 这里,传输侧的反馈处理器可以将从360度视频接收装置接收到的反馈信息传递到传输处理器,并且传输处理器可以执行针对图块而区分的传输处理。例如,传输侧的反馈处理器可以将接收侧接收的视口信息传递给传输处理器。传输处理器可以对包括视口区域的图块执行传输处理,以具有比其他图块更高的鲁棒性。

[0214] 图11示出根据本发明的示例性实施例的与360视频有关的元数据的示例。如上所述,与360度视频有关的元数据可以包括与360度视频有关的各种元数据。根据上下文,与360度视频有关的元数据也可以被称为与360度视频有关的信令信息。360度视频相关的元数据可以被包括在单独的信令表中,并且然后可以被发送,或者360度视频相关的元数据可以被包括在DASH MPD中并且然后可以被发送,或者360度视频相关的元数据可以以盒子的形式被包括在诸如ISO BMFF等的文件格式中,并且然后可以被传递。如果以盒子形式包括与360度视频有关的元数据,则元数据被包括在多个级别中,诸如文件、分片(fragment)、轨道、样本条目、样本等,并且因此,可以包括针对相应的级别的数据的元数据。

[0215] 根据示例性实施例,稍后将描述的元数据的一部分可以被配置成信令表,并且然后被传递,并且元数据的其余部分可以以盒子或轨道格式被包括在文件格式内。

[0216] 根据示例性实施例,根据本发明的与360视频有关的元数据可以包括与投影格式有关的默认元数据、与立体有关的元数据、与初始视图/初始视点有关的元数据、与ROI有关

的元数据、与视场 (FOV) 有关的元数据和/或与裁剪区域有关的元数据。根据示例性实施例,除了上述元数据之外,与360视频有关的元数据还可以包括附加元数据。

[0217] 根据本发明的与360有关的元数据的示例性实施例可以对应于一种格式,该格式包括上述默认元数据、与立体有关的元数据、与初始视点有关的元数据、与ROI有关的元数据、与FOV有关的元数据、与裁剪区域有关的元数据和/或稍后可能添加的元数据中的至少一个或多个。根据本发明的与360有关的元数据的示例性实施例可以根据每个示例性实施例中包括的详细元数据的情况的数量来不同地配置。根据示例性实施例,除了上述元数据之外,与360视频有关的元数据还可以包括附加信息。

[0218] `stereo_mode`字段可以指示由相应的360度视频支持的3D布局。也可以仅使用此字段来指示相应的360度视频是否支持3D。在这种情况下,可以省略`is_stereoscopic`字段。当该字段的值等于0时,360度视频可以对应于单声道模式。更具体地,投影的2D图像可以仅包括一个单色视图。在这种情况下,相应的360度视频可能不支持3D。

[0219] 当该字段的值等于1或2时,相应的360度视频可以遵循左右布局或上下布局。左右布局和上下布局可以分别被称为并排格式和上下格式。在左右布局中,在其上投影左图像/右图像的2D图像可以位于图像帧的左侧/右侧。在上下布局中,在其上投影左图像/右图像的2D图像可以位于图像帧的顶/底侧。当此字段等于其他值时,其余值可保留以供将来使用。

[0220] 与初始视图有关的元数据可以包括关于用户在首次再现(或回放)360度视频时所观看的视图(初始视点)的信息。与初始视图相关的元数据可以包括`initial_view_yaw_degree`字段、`initial_view_pitch_degree`字段和/或`initial_view_roll_degree`字段。在一些实施例中,与初始视图有关的元数据可以进一步包括附加信息。

[0221] `initial_view_yaw_degree`字段、`initial_view_pitch_degree`字段和`initial_view_roll_degree`字段可以指示在360度视频再现(或回放)时的初始视点。换句话说,可以通过这三个字段来指示在执行回放(或再现)时第一次被观看的视口的中心点。更具体地,`initial_view_yaw_degree`字段可以指示初始视图的偏航值。更具体地,`initial_view_yaw_degree`字段可以指示中心点的位置作为与偏航轴相关的旋转方向(符号)和旋转程度(角度)。另外,`initial_view_pitch_degree`字段可以指示初始视图的俯仰值。更具体地,`initial_view_pitch_degree`字段可以指示中心点的位置作为与俯仰轴相关的旋转方向(符号)和旋转程度(角度)。并且,另外,`initial_view_roll_degree`字段可以指示初始视图的横滚值。更具体地,`initial_view_roll_degree`字段可以指示中心点的位置作为与横滚轴相关的旋转方向(符号)和旋转程度(角度)。可以基于`initial_view_yaw_degree`字段、`initial_view_pitch_degree`字段和`initial_view_roll_degree`字段指示在再现(或回放)相应的360度视频时的初始视点,即,在再现(或回放)视频时首先显示的视点的中心点。因此,可以显示360度视频的特定区域,并且然后将其提供给用户的初始视点。另外,可以通过视场 (FOV) 来确定基于指示的初始视点的初始视口的宽度和高度。更具体地,360度视频接收设备(或接收装置)可以使用这三个字段和FOV信息向用户提供360度视频的预定区域作为初始视口。

[0222] 在一些实施例中,由与初始视图有关的元数据指示的初始视点可以针对每个场景被改变。更具体地,可以根据360度内容随着时间流逝来改变360度视频的场景。可以针对

360度视频的每个场景改变用户首先观看的初始视点或初始视口。在这种情况下,与初始视图有关的元数据可以指示每个场景的初始视点。为此,与初始视图有关的元数据可以进一步包括场景标识符,该场景标识符标识应用初始视点的场景。另外,因为可以针对360度视频的每个场景改变FOV,所以与初始视图有关的元数据可以进一步包括指示与该场景相对应的FOV的按场景FOV信息。

[0223] 与ROI有关的元数据可以包括与上述ROI有关的信息。与ROI有关的元数据可以包括2d_roi_range_flag字段和/或3d_roi_range_flag字段。2d_roi_range_flag字段可以指示与ROI有关的元数据是否包括基于2D图像表达ROI的字段,并且3d_roi_range_flag字段可以指示与ROI有关的元数据是否包括基于3D空间表达ROI的字段。在一些实施例中,与ROI有关的元数据可以进一步包括附加信息,诸如基于ROI的差分编码信息和基于ROI的差分传输处理信息。

[0224] 当与ROI有关的元数据包括基于2D图像表达ROI的字段时,与ROI有关的元数据可以包括min_top_left_x字段、max_top_left_x字段、min_top_left_y字段、max_top_left_y字段、min_width字段、max_width字段、min_height字段、max_height字段、min_x字段、max_x字段、min_y字段和/或max_y字段。

[0225] min_top_left_x字段、max_top_left_x字段、min_top_left_y字段和max_top_left_y字段可以指示ROI的左上端的坐标的最小/最大值。这些字段可以分别指示左上端的最小x坐标、最大x坐标、最小y坐标和最大y坐标。

[0226] min_width字段、max_width字段、min_height字段和max_height字段可以分别指示ROI的宽度和高度的最小/最大值。更具体地,这些字段可以分别指示宽度的最小值、宽度的最大值、高度的最小值和高度的最大值。

[0227] min_x字段、max_x字段、min_y字段和max_y字段可以指示ROI中的坐标的最小/最大值。更具体地,这些字段可以分别指示ROI中的坐标的最小x坐标、最大x坐标、最小y坐标和最大y坐标。这些字段可以省略。

[0228] 当与ROI有关的元数据包括基于3D渲染空间中的坐标表达ROI的字段时,与ROI有关的元数据可以包括min_yaw字段、max_yaw字段、min_pitch字段、max_pitch字段、min_roll字段、max_roll字段、min_field_of_view字段和/或max_field_of_view字段。

[0229] min_yaw字段、max_yaw字段、min_pitch字段、max_pitch字段、min_roll字段和max_roll字段可以分别指示ROI在3D空间中占据的区域,作为偏航、俯仰以及横滚的最小/最大值。更具体地,这些字段可以分别指示相对于偏航轴的参考旋转量的最小值、相对于偏航轴的参考旋转量的最大值、相对于俯仰轴的参考旋转量的最小值、相对于俯仰轴的参考旋转量的最大值、相对于横滚轴的参考旋转量的最小值、以及相对于横滚轴的参考旋转量的最大值。

[0230] min_field_of_view字段和max_field_of_view字段可以指示360度视频数据的FOV的最小值/最大值。FOV可以指的是当视频被再现(或回放)时一次显示360度视频的视觉范围。min_field_of_view字段和max_field_of_view字段可以分别指示FOV的最小值和最大值。这些字段可以省略。这些字段可以被包括在稍后将描述的与FOV有关的元数据中。

[0231] 与FOV有关的元数据可以包括与上述FOV有关的信息。与FOV有关的元数据可以包括content_fov_flag字段和/或content_fov字段。在一些实施例中,与FOV有关的元数据可

以进一步包括附加信息,诸如关于FOV的最小/最大值的信息。

[0232] `content_fov_flag`字段可以指示是否存在关于在制作时想要的360度视频的FOV的信息。如果该字段的值等于1,则`content_fov`字段可能存在。

[0233] `content_fov`字段可以指示在制作时想要的有关360度视频的FOV的信息。在一些实施例中,可以基于360度视频接收设备的垂直或水平FOV来确定一次显示给用户的360度视频的一部分。可替代地,在一些实施例中,可以在反映该字段的FOV信息的同时确定一次显示给用户的360度视频的一部分。

[0234] 与裁剪区域有关的元数据可以包括关于区域的信息,该区域包括图像帧中的实际360度视频数据。图像帧可以包括活动视频区域和不活动视频区域,在活动视频区域中实际的360度视频数据被投影。在此,活动视频区域可以被称为裁剪区域或默认显示区域。活动视频区域是在实际VR显示器中被看作360度视频的区域。360度视频接收设备或VR显示器可能仅处理/显示活动视频区域。例如,当图像帧的纵横比为4:3时,仅排除图像帧的上部的一部分和下部的一部分的图像帧的区域可以包括360度视频数据。该区域可以被称为活动视频区域。

[0235] 与裁剪区域有关的元数据可以包括`is_cropped_region`字段、`cr_region_left_top_x`字段、`cr_region_left_top_y`字段、`cr_region_width`字段和/或`cr_region_height`字段。在一些实施例中,与裁剪区域有关的元数据可以进一步包括附加信息。

[0236] `is_cropped_region`字段可以是指指示图像帧的整个区域是否被360度视频接收设备或VR显示器使用的标志。在此,该字段可以被称为活动视频区域,其在映射到360度视频数据的区域中或在VR显示器中示出。`is_cropped_region`字段可以指示是否整个图像帧对应于活动视频区域。当图像帧的仅一部分是活动视频区域时,可以进一步包括以下四个字段。

[0237] `cr_region_left_top_x`字段、`cr_region_left_top_y`字段、`cr_region_width`字段和`cr_region_height`字段可以指示图像帧中的活动视频区域。这些字段可以分别指示活动视频区域的左上方的x坐标、活动视频区域的左上方的y坐标、活动视频区域的宽度和活动视频区域的高度。宽度和高度可以以像素为单位来表达。

[0238] 基于360度视频的VR系统可以基于上述的360度视频处理过程基于用户相对于360度视频的位置为不同的观看定向提供视觉/听觉体验。基于与360度视频有关的用户的固定位置为不同观看定向提供视觉/听觉体验的VR系统可以被称为基于三自由度(3DoF)的VR系统。同时,在不同视点和/或不同观看位置处为不同观看定向提供扩展的视觉/听觉体验的VR系统可以被称为基于3DoF plus或3DoF+的VR系统。

[0239] 图12示出视点、观看位置和观看定向的概念的总体视图。

[0240] 参考图12,当假定(a)中所示的空间(例如,表演台)时,每个标记的圆圈可以指示不同的视点。从被定位在相同空间内的每个视点提供的视频/音频可以在同一时隙处彼此相关。在这种情况下,取决于特定视点处用户观看方向的变化(例如,头部运动),可以向用户提供不同的视觉/听觉体验。更具体地,如(b)中所示,可以假设关于特定视点的不同观看位置的球体,并且可以提供反映每个观看位置的相对位置的视觉/音频/文本信息。

[0241] 同时,如(c)中所示,在特定视点的特定观看位置中,可以传递诸如常规3DoF的不同方向(或定向)的视觉/听觉信息。此时,可以集成并且然后提供主要源(例如,视频/音频/

文本)以及各种其他源,并且在这种情况下,可以将相应的源链接到用户的观看定向,并且然后传递,或者可以独立地传递相应的信息。

[0242] 图13示出根据本发明的用于提供3DoF+视频的示例性架构的总体视图。图13可以图示3DoF+端对端系统流程图,其包括3DoF+的图像获取、预处理、传输、(后)处理、渲染和反馈过程。

[0243] 参考图13,获取过程可以指通过执行360视频的捕获、合成或生成过程来获取360视频(或360度视频)的过程。通过执行获取过程,可以获取根据相对于多个位置的观看定向(例如,头部运动)的变化的各种视频/音频信息。此时,视频信息不仅可以包括视觉信息(例如,纹理),还可以包括深度信息。此时,如(1310)的示例性视觉信息所示,可以分别获取根据不同视点的不同观看位置的各种信息。

[0244] 合成过程可以包括用于执行合成以便于在用户的体验中不仅包括通过视频/音频输入设备获取的信息而且包括视频(视频/图像等)、音频(音频/声音效果)和文本(字幕等)的过程和方法。

[0245] 预处理过程对应于用于发送/传递所获取的360视频的准备(预处理)过程,其可以包括上述的拼接、投影、按区域打包和/或编码过程等等。更具体地,该过程可以包括预处理过程和编码过程,其用于根据制作人的意图来修改/补充视频/音频/文本信息的数据。例如,在音频的预处理过程中,可以包括用于将获取的视觉信息映射到360球体上的拼接过程、用于去除区域边界或减少色度/亮度差异或添加图像的视觉效果 of 的编辑过程、用于根据视点对图像进行分割的视图分割过程、用于将360球面的图像映射到2D图像的投影过程、用于根据区域重新定位图像的按区域打包过程以及用于压缩图像信息的编码过程。如(1320)的视频方面的示例所示,可以生成根据不同视点的不同观看位置的多个投影图像。

[0246] 传输过程可以指的是处理和发送在准备过程(预处理过程)中处理的音频/视频数据和元数据的过程。如上所述,作为根据不同视点来传递(或传送)不同观看位置的各种音频/视频数据和元数据的方法,可以使用广播网络和/或通信网络,或者可以使用单向传送方法。

[0247] (后)处理和合成过程可以指用于对接收/存储的视频/音频/文本数据进行解码并用于执行最终回放(或再现)的后处理过程。例如,后处理过程可以包括用于对上述打包的图像进行解包的解包过程、用于将2D投影的图像恢复为3D球形图像的再投影过程等等。

[0248] 渲染过程可以指的是用于渲染和显示被重新投影到3D空间的图像/视频数据的过程。在此过程中,视频/音频信号可以重新配置成最终输出的格式。可以跟踪观看定向、观看位置/头部位置以及用户感兴趣的区域存在的视点,并且根据该信息,可以选择性地仅使用必要的图像/音频/文本信息。此时,在图像信号的情况下,取决于用户的感兴趣的区域,可以选择不同的观看位置,如(1330)中所示,并且最后,可以输出在特定视点处的特定观看位置的特定观看定向的图像,如(1340)中所示。

[0249] 图14a和图14b是3DoF+端对端系统架构的示例。可以通过图14a和图14b的架构来提供上述3DoF+360内容。

[0250] 参考图14a,360视频发送装置(发送端)可以广泛地由获取360视频(图像)/音频数据的部分(获取单元)、处理获取的数据的部分(视频/音频预处理器)、用于合成附加信息的部分(合成生成单元)、用于对文本、音频和投影的360度视频进行编码的部分(编码单元)以

及用于对编码数据进行封装的部分(封装单元)构成。如上所述,编码数据可以以比特流格式输出,并且编码数据可以以诸如ISOBMFF、CFF等的文件格式被封装,或者可以以其他DASH片段的形式被处理。编码的数据可以通过数字存储介质传递(或传送)到360视频接收设备。可替代地,尽管在附图中未明确示出,但是编码数据可以被传输处理器处理以便进行传输,并且然后可以经由广播网络或宽带来发送。

[0251] 在数据获取部分(或单元)中,可以根据传感器定向(如果是图像,则为观看定向)、传感器位置(如果是图像,则为观看位置)、以及传感器的信息获取点(如果是图像,则为视点)同时或连续地获取不同的信息。并且,此时,可以获得视频、图像、音频、位置信息等。

[0252] 在图像数据的情况下,可以获得纹理和深度信息中的每一个,并且可以根据每个组件的特性来执行不同的视频预处理。例如,在纹理信息的情况下,可以通过使用通过图像传感器位置信息从相同视点获取的具有相同观看位置的不同观看方位的图像来配置360度全向图像。并且为此,可以执行视频拼接过程。另外,可以执行投影和/或按区域打包过程,以便于将图像改变为适合于编码的格式。在深度图像的情况下,通常可以通过深度相机获取图像,并且在这种情况下,可以将深度图像创建为具有纹理格式。可替代地,还可以基于单独测量的数据来生成深度数据。在为每个组件生成图像(或按组件的图像)之后,通过对适用于有效压缩的视频格式执行额外打包或将图像分割为实际需要的必要部分来重新配置图像(子图像生成)的过程可以被执行。视频预处理端使用的有关图像配置的信息作为视频元数据被传递(或传送)。

[0253] 除了所获取的数据(或将主要服务的数据)之外,在服务额外给出的视频/音频/文本信息的情况下,用于在执行此类信息的最终回放(或再现)时执行合成的信息可能需要被提供。合成生成单元生成基于制作人的意图用于使从外部来源生成的媒体数据在最终回放端执行合成的信息(在图像的情况下为视频/图像、在声音(音频)的情况下为音频/声音效果、以及在文本的情况下为字幕等),并且此类信息作为合成元数据被传递(或传送)。

[0254] 通过每个过程处理的视频/音频/文本信息均可以通过使用各自的编码器进行压缩,并且取决于应用,该信息可以以文件或片段为单位被封装。此时,可以根据视频、文件或片段配置方法仅(由文件提取器)提取必要信息。

[0255] 另外,使接收器能够重新配置每个数据的信息以文件格式/系统级别被传递(或传送)。这样的信息可以包括用于视频/音频重新配置的信息(视频/音频元数据)、用于叠加的合成信息(合成元数据)、根据每个视点的视频/音频可播放的视点信息和视点位置信息(观看位置和视点元数据)等等。

[0256] 参考图14b,360视频接收设备(接收端)可以广泛地由用于解封装接收到的文件或片段的部分(文件/片段解封装单元)、用于从比特流生成视频/音频/文本信息的部分(解码单元)、用于将视频/音频/文本重新配置成适合回放(或再现)的格式的部分(后处理器)、用于跟踪用户感兴趣区域的部分(跟踪单元)以及对应于回放(或再现)设备的显示器构成。

[0257] 取决于数据的类型,可以将通过解封装生成的比特流分割(或划分)为视频/音频/文本等,并可以分别解码为可播放格式。

[0258] 跟踪部分(或单元)可以生成关于用户感兴趣的区域的视点、在相应视点处的观看位置以及在相应视点位置处的观看定向的信息,并且该信息可以用于在360视频接收设备中通过每个模块选择或者提取感兴趣的区域,或者可以在后处理过程中使用来强调有关感

兴趣区域的信息。此外,在将信息传递到360视频发送设备的情况下,该信息可用于现有效的带宽使用的文件选择(文件提取器)或子图片选择、基于感兴趣的区域的多种图像重新配置方法(视口/观看位置/视点相关处理)。

[0259] 可以根据图像配置方法通过各种处理方法来处理解码图像信号。在360视频发送设备已经执行图像打包的情况下,需要基于经由元数据传递的信息来重新配置相应图像的处理。在这种情况下,可以使用由360视频发送设备生成的视频元数据。另外,在解码图像中包括多个视点或多个观看位置或各种观看定向的图像的情况下,可以选择和处理通过执行跟踪而生成的与用户的感兴趣的区域的视点、观看位置、观看定向相匹配的信息。此时,可以使用由发送端生成的与观看位置和视点有关的元数据。另外,在与特定视点、观看位置和观看定向有关的多个组件被传递的情况下,或者在分别传递用于叠加的视频信息的情况下,可以包括针对每种情况的渲染处理。用单独的渲染过程处理的视频数据(纹理、深度、叠加)可以用合成过程来处理。并且,此时,可以使用由发送端生成的合成元数据。最后,可以生成用于根据用户感兴趣的区域在视口处执行回放的信息。

[0260] 可以通过渲染器和/或通过(后)处理过程将解码的音频信号生成为可播放的音频信号。并且,此时,可以基于关于用户感兴趣的区域的信息和由360视频接收设备传递的元数据来生成最适合用户需求的信息。

[0261] 可以将解码后的文本信号传递到渲染器,并且然后可以基于诸如字幕的文本将其作为叠加信息进行处理。当需要时,可以包括单独的测试后处理过程。

[0262] 图15示出示例性的直播上行链路流框架(FLUS)架构的总体视图。

[0263] 在图14(图14a和图14b)中,上述发送端和接收端的详细块可以分别通过直播上行链路流框架(FLUS)中的源和宿(sink)功能来分类。

[0264] 在通过源和宿功能对发送端和接收端的详细块进行分类的情况下,如图14所示,源功能可以在360度视频获取设备中实现,并且宿功能可以在网络中实现,或者每个源/宿功能可以在网络节点中实现。

[0265] 基于上述架构的传输/接收处理过程的一般示例可以如图15和图16中所图示。基于图像信号处理过程来描述图15和图16的传输/接收处理过程,并且在处理诸如音频或文本的其他信号的情况下,某些部分(例如,拼接器、投影处理器、打包处理器、子图片处理器、解包/选择、渲染、合成、视口生成等)可以被省略,或者可以被改变和处理以适合音频或文本处理过程。

[0266] 图16示出3DoF+发送端的一般配置。

[0267] 参考图16,在输入的数据对应于相机输出图像的情况下,发送端(360视频发送设备)可以针对视点/观看位置/组件中的每一个执行用于球体图像配置的拼接。当配置按视点/观看位置/组件的球体图像时,可以将所配置的球体图像投影到2D图像以便执行编译。取决于应用,可以执行打包以将多个图像创建为整合图像,或者可以将图像创建为将图像分割成详细的分割区域的子图片。如上所述,作为可选处理,可以不执行按区域打包过程,并且在这种情况下,可以省略打包处理器。在输入的数据对应于附加视频/音频/文本信息的情况下,可以通知将附加信息添加并显示到中心图像的方法,并且附加数据也可以与该通知一起被发送。生成的图像和添加的数据可以通过编码过程进行处理,其中生成的图像和添加的数据被压缩并生成为比特流,并且然后可以通过封装过程进行处理,其中处理后

的数据被转换为适合传输或存储的文件格式。此时,取决于应用或系统要求,可以执行提取由接收单元所需的文件的过程。所生成的比特流可以被传输处理器转换成传送格式,并且然后可以被发送。此时,发送端的反馈处理器可以基于从接收端传递的信息来处理视点/观看位置/观看定向信息和必要的元数据。此后,可以将处理的信息和元数据传递到相关的发送单元使得进行处理。

[0268] 图17示出3DoF+接收端的一般配置。

[0269] 参考图17,在接收到从发送端传递的比特流之后,接收端(360视频接收设备)可以提取必要的文件。接收端通过使用从反馈处理器传递的视点/观看位置/观看定向信息和视频元数据来选择并处理所生成的文件格式内的图像流,并且然后,接收端可以通过解码器将所选择的比特流重新配置成图像信息。在打包图像的情况下,可以基于通过元数据传递的打包信息来执行解打包。在发送端省略打包过程的情况下,接收端的解包也可以被省略。此外,当需要时,可以执行选择从反馈处理器传递的、适合于视点/观看位置/观看定向的图像以及选择必要的组件的过程。可以执行用于将图像的纹理、深度、叠加信息等重新配置成适合于回放(或再现)的格式的渲染过程。在生成最终图像之前,可以执行用于整合不同层的合成过程,并且可以生成适合于显示视口的图像,并且然后回放(或再现)。

[0270] 本说明书提供一种用于360视频(VR或增强现实(AR))的图像叠加方法。在360视频的情况下,因为用户可以通过将视点/观看定向改变360度来观看图像/视频/现实(在透明的情况下),所以用户可以被暴露于更广泛的内容。尽管可以向用户的视口立即显示关于360内容的特定位置的辅助信息,但是在这种情况下,辅助信息和连接位置之间的连接可能会断开。在这种情况下,用户应定位在他(或她)的视点中显示的信息的位置。为了同时直观地显示特定位置和各个辅助信息,可以将与特定视点匹配的辅助信息叠加显示,或者可以在与辅助信息匹配的特定视点附近显示辅助信息并且绘制并显示线以指示相应的信息是关于特定视点的信息。对于此类用法,360视频叠加应支持一种用于指定叠加位置使得叠加可以存在于用户视口以及三维(3D)空间中的方法。本发明提供一种用于支持关于360视频的图像叠加的方法。本发明可以应用于基于3DoF的VR系统以及基于3DoF+的VR系统,并且本发明也可以应用于增强现实(AR)系统。

[0271] 本发明提供一种用于存储和用信号发送用于支持叠加的信息的方法,这将在后面详细描述。用于支持叠加的信息可以以与叠加有关的元数据的格式来传递(或传送)。可以通过发送端的数据输入单元和/或元数据处理器将用于支持叠加的信息传递给数据编码器和/或封装处理器,并且然后可以进行处理以进行传输,并且然后,可以将该信息发送给接收端。在这种情况下,可以以比特流格式来传递相关信息。比特流也可以被称为视频流或视频比特流。接收端可以从接收到的比特流中提取必要的文件,并且可以经由封装处理器和元数据解析器提取与叠加有关的元数据。此后,所提取的元数据被传递到渲染单元,并且渲染单元渲染叠加,其然后通过合成过程被处理,从而被显示在屏幕上。

[0272] 因为用于二维(2D)图像/视频的图像叠加尚未考虑360 3D空间,所以其特性与360视频的叠加的特性不同,并且因此无法立即应用于360视频。根据本发明,如下所述,可以定义叠加的位置和顺序(叠加内容、叠加纹理、叠加图像等)。例如,根据本发明的叠加可以根据叠加的(渲染)类型广泛地分类到类别。这些类别对应于:1)当前视口上的叠加,2)360表面上的叠加,和3)360表面内的3D空间的叠加。这里,360表面可以对应于球体表面或立方体

表面。当前视口上的叠加也可以被称为视口相关的叠加,并且360表面上的叠加也可以被称为球体相关投影的全向叠加,并且360表面内的3D空间的叠加也可以被称为球体相关的2D叠加。在360表面内部的3D空间中叠加的图像可以对应于2D平面图像。根据本发明,可以根据每种叠加类型来用信号发送各种相关的元数据,并且元数据可以包括与叠加有关的位置和顺序信息。在下文中,叠加主体可以可互换地被称为叠加内容、叠加视觉媒体、叠加纹理、(叠加)图形、叠加区域、叠加图像、叠加纹理、叠加视频、叠加文本、叠加等。

[0273] 为了支持根据本发明的上述叠加,例如,如下所述,可以生成并发送与叠加有关的元数据。下表显示在OverlayLocationConfigBox中发送的相关元数据的示例。然而,这仅是示例性的,并且因此,可以不同地设置正在发送元数据的盒子的标题,并且元数据可以被包括作为文件内单独轨道的数据。而且,也可以省略与叠加有关的元数据内的部分信息。

[0274] [表1]

[0275]

```

Class OverlayLocationConfigBox extends FullBox('olcb', version=0, flags=0) {
    unsigned int(1) stereoscopic_flag;
    unsigned int(8) overlay_group_id;
    unsigned int(1) sphere_region_flag;
    unsigned int(1) range_included_flag;

    if(sphere_region_flag == 1) {
        signed int(32) centre_azimuth;
        signed int(32) centre_elevation;
        signed int(32) centre_tilt;
        if (range_included_flag) {
            unsigned int(32) azimuth_range;
            unsigned int(32) elevation_range;
        }
    }

    if(overlay_group_id == 5) {          // 在球体或立方体上
        unsigned int(16) depth;          // 在组中合成的顺序
    } else if(overlay_group_id == 4) { // 在3D空间中
        if(sphere_region_flag == 1) {
            unsigned int(16) distance; // 从球体中心的距离
        } else {
            signed int(32) centre_x;
            signed int(32) centre_y;
            signed int(32) centre_z;
            if (range_included_flag) {
                unsigned int(32) width;
                unsigned int(32) height;
            }
            signed int(32) rotation_yaw;
            signed int(32) rotation_pitch;
            signed int(32) rotation_roll;
        }
        unsigned int(1) billboards_flag; // 始终面对相机
    } else {          // 在当前视口上
        unsigned int(16) reference_width;
        unsigned int(16) reference_height;
        unsigned int(16) top_left_x;
        unsigned int(16) top_left_y;
        unsigned int(16) 2d_width;
        unsigned int(16) 2d_height;
        unsigned int(16) depth;          //在组中合成的顺序
        unsigned int(32) tilt;
        if(stereoscopic_flag== 1) {
            unsigned int(1) relative_disparity_flag;
            if(relative_disparity_flag ==1) {
                signed int(16) disparity_in_percent;
            } else {
                signed int(16) disparity_in_pixels;
            }
        }
    }
    unsigned int(1) target_flag;          // 指示目标位置
    if(target_flag == 1) {
        signed int(32) target_azimuth;
        signed int(32) target_elevation;
    }
}

```

[0276] 参考表1, `stereoscopic_flag` 字段指示是否应立体地渲染叠加层。在 `stereoscopic_flag` 字段的值等于0的情况下, 这可以指示叠加层将被单视场渲染。并且, 在其他情况下, 即, 在 `stereoscopic_flag` 字段的值等于1的情况下, 这可以指示应当通过使用视差来立体地渲染叠加层。

[0277] `overlay_group_id` 字段指示视觉媒体 (即, 叠加图像) 将被定位的位置。更具体地, `overlay_group_id` 对应于指示上述叠加类型的信息 (或用于叠加类型的信息)。 `overlay_group_id` 也可以被称为 `overlay_rendering_type`、`overlay_type` 或 `display_mode` 等。用于叠加类型的信息可以指示叠加类型中的一种。例如, 叠加类型可以包括1) 当前视口上的叠加, 2) 360表面上的叠加, 以及3) 360表面内部的3D空间的叠加。在这种情况下, 例如, `overlay_group_id` 字段的值0至2可以指示上述的1) 至3) 的叠加类型。可替代地, 通过进一步分割当前视口上的叠加类型, 可以进一步定义顺序。例如, 在 `overlay_group_id` 字段的值等于1的情况下, 视觉媒体可以被定位在当前视口上, 并且这可以指示相应的视觉媒体具有最高优先级, 诸如, 标志 (logo) 等等。并且, 在 `overlay_group_id` 字段的值等于2的情况下, 视觉媒体可以被定位在当前视口上, 并且这可以指示相应的视觉媒体具有比该字段的值等于1的情况低的优先级。例如, 字幕和/或闭合字幕等可能对应于 `overlay_group_id` 字段等于2的情况。在 `overlay_group_id` 字段的值等于3的情况下, 视觉媒体可能被定位在当前视口上, 并且这可能指示相应的视觉媒体具有比此字段的值等于2的情况低的优先级。取决于内容的上下文, 可以预先定义部分顺序。例如, 可以预先定义标志 (logo) 和/或字幕的优先级。如果 `overlay_group_id` 字段的值等于4, 这可能指示视觉媒体可能被定位在360表面内的3D空间上。360表面可以对应于球体表面或立方体表面。并且, 在 `overlay_group_id` 字段的值等于5的情况下, 这可以指示视觉媒体可以被定位在360表面上。 `overlay_group_id` 字段可以暗示 (或建议) 将要配置的叠加的顺序。可替代地, 可以显式地使用 (或用信号发送) 与叠加顺序有关的信息。因此, 可以特定地指示相同叠加组内的视觉媒体的叠加顺序, 即, 可以在叠加情况下给哪个视觉媒体更高的优先级。如下表所示, 可以表示根据 `overlay_group_id` 字段的值的示例性叠加位置和顺序。

[0278] [表2]

[0279]	overlay_group_id 值	意义	示例
	1	位于当前视口上方, 并具有最高优先级	标志 (logo)
	2	位于当前视口上方, 并且优先级低于 overlay_group_id 1	字幕, 闭合字幕
	3	位于当前视口上方, 并且优先级低于 overlay_group_id 2	
	4	位于 3d 空间中	
	5	位于作为球体或立方体的 360°表面上方	

[0280] 例如, 可以如以下附图所示, 指示根据 `overlay_group_id` 字段值的叠加位置。图18至图20分别示出根据叠加类型的示例性叠加位置。更具体地, 例如, 图18对应于叠加图像与当前视口重叠的情况。这可以表示 `overlay_group_id` 字段值等于1至3的情况。图19对应于叠加图像在360表面内与3D空间重叠的情况。这可以表示 `overlay_group_id` 字段值等于4的情况。图20对应于叠加图像与360表面重叠的情况。这可以表示 `overlay_group_id` 字段等于

5的情况。

[0281] sphere_region_flag可以指示叠加位置是否基于与3D坐标相对应的球体坐标的方位角和仰角。在sphere_region_flag的值等于1的情况下,这可以指示基于方位角和仰角来指示叠加位置。在其他情况下,可以通过其他方法指示叠加位置。

[0282] centre_azimuth字段和centre_elevation字段指示球体区域的方位角和仰角。这些值可以 2^{-16} 度为单位指示。centre_azimuth字段值可以在 -180×2^{16} 至 $180 \times 2^{16}-1$ 的范围内。centre_elevation字段值可以在 -90×2^{16} 到 $90 \times 2^{16}-1$ 的范围内。

[0283] centre_tilt可以指示球体区域的倾斜角。倾斜值可以 2^{-16} 度为单位指示。centre_tilt值可以在 -180×2^{16} 到 $180 \times 2^{16}-1$ 的范围内。

[0284] azimuth_range字段和elevation_range字段指示通过球体区域的中心点的方位角和仰角范围。字段的值可以 2^{-16} 度为单位指示。azimuth_range字段的值在0到 360×2^{16} 的范围内。并且,elevation_range字段的值在0至 180×2^{16} 的范围内。

[0285] depth字段指示合成顺序。较小的值可以指示相应的叠加被定位为靠近叠加组内的叠加层的顶部。该字段的值在0到 2^{16} 的范围内。

[0286] distance字段指示单位球体距中心的距离。换句话说,distance字段可以对应于单位球体的半径。distance字段的值是相对于单位球体,并且可以在0到 2^{-16} 的范围内。

[0287] centre_x字段、centre_y字段和centre_z字段指示3D坐标空间内的位置。3D坐标空间内的原点可能对应于单位球体的中心。该字段的值可以相对于单位球体。该字段的值可以 2^{-16} 为单位指示。

[0288] width字段和height字段指示叠加大小。这些字段的值可以相对于单位球体,并且这些字段的值可以 2^{-16} 为单位指示。

[0289] rotation_yaw字段、rotation_pitch字段和rotation_roll字段可以分别指示应用于单位球体的旋转的偏航角、俯仰角和横滚角。该字段的值可以相对于单位球体。该字段的值可以 2^{-16} 为单位指示。rotation_yaw字段的值可以在 -180×2^{16} 至 $180 \times 2^{16}-1$ 的范围内。rotation_pitch字段的值可以在 -90×2^{16} 至 $90 \times 2^{16}-1$ 的范围内。rotation_roll字段的值可以在 -180×2^{16} 至 $180 \times 2^{16}-1$ 的范围内。

[0290] billboards_flag字段指示旋转是否根据相机观看定向而改变。在billboards_flag字段的值等于1的情况下,视觉媒体的叠加层(平面)的法线值可以等于相机的当前面部矢量。

[0291] reference_width字段和reference_height字段分别指示在其中计算所有ROI坐标(top_left_x、top_left_y、2d_width和2d_height)的参考矩形空间的宽度和高度。

[0292] top_left_x字段和top_left_y字段分别指示矩形区域的水平和垂直坐标。

[0293] 2d_width字段和2d_height字段分别指示矩形区域的宽度和高度。

[0294] Tilt字段指示球体区域的倾斜角。Tilt字段可以以 2^{-16} 度为单位指示倾斜角。Tilt字段的值可以在 -180×2^{-16} 至 $180 \times 2^{-16}-1$ 的范围内。

[0295] relative_disparity_flag字段可以指示是否以单个视图的显示窗口的宽度的百分比值给出视差(字段值等于1的情况),或者是否以像素的数量给出视差(字段值等于0的情况)。

[0296] disparity_in_percent字段以 2^{-16} 为单位指示视差,作为针对单个视图的显示窗

口宽度的分数。在位移方向相反的情况下,该字段的值可以等于负数。disparity_in_percent字段的值可用于针对左眼视野将区域定位在左侧,并且针对右眼视野将区域定位在右侧。

[0297] Disparity_in_pixels字段指示以像素为单位的视差。在位移方向相反的情况下,该字段的值可以等于负数。Disparity_in_pixels字段的值可用于针对左眼视野将区域定位在左侧,并且针对右眼视野将区域定位在右侧。

[0298] target_flag字段可以指示视觉媒体是否具有参考点。叠加可以用于多种目的。例如,叠加可以用于提供与特定位置有关的补充信息。在这种情况下,叠加视觉媒体可以被放置在精确的位置处,或者可以放置在相应位置附近,以便于避免阻挡(或覆盖)主要信息。在后者的情况下,可以使用target_flag字段。如果该字段的值等于1,则视觉媒体可能具有与特定点有关的位置信息。

[0299] target_azimuth字段和target_elevation字段可以分别指示球体区域的方位角和仰角值。target_azimuth字段和target_elevation字段的值可以以 2^{-16} 度为单位指示方位角和仰角。target_azimuth字段的值可以在 -180×2^{16} 至 $180 \times 2^{16} - 1$ 的范围内。并且,target_elevation字段的值可以在 -90×2^{16} 至 $90 \times 2^{16} - 1$ 的范围内。

[0300] 可以通过使用各种方法将上述与叠加有关的元数据从发送设备发送到接收设备。

[0301] 例如,可以以叠加定时的元数据轨道格式来发送与叠加有关的元数据。为此,可以定义叠加定时元数据轨道,并且可以将叠加定时元数据轨道链接到一个或多个媒体轨道。在这种情况下,叠加定时元数据轨道可以与“sdsc”轨道参考一起被链接到一个或多个媒体。在这种情况下,OverlayLocationConfigBox可能存在于元数据样本条目中,并且OverlayLocationConfigBox可以指示与叠加有关的位置和顺序。例如,可以包括OverlayLocationConfigBox,如下表所示。

[0302] [表3]

[0303]

```
class OverlayObjectSampleEntry(type) extends MetaDataSampleEntry (type) {
    OverlayLocationConfigBox();
}
```

[0304] 作为另一示例,可以在叠加轨道和样本内以时间信息格式发送与叠加有关的元数据。为此,可以定义叠加轨道。在这种情况下,在样本条目中可能存在OverlayLocationConfigBox。OverlayLocationConfigBox可以指示与叠加有关的位置和顺序。媒体数据中的单个样本可以包括用于执行附加显示的开始时间和结束时间信息。例如,可以包括OverlayLocationConfigBox,如下表所示。

[0305] [表4]

[0306]

```
class OverlayObjectSampleEntry(type) extends SampleEntry ('ovob') {
    OverlayLocationConfigBox();
}
```

[0307] 图21示出被存储在样本中的叠加轨道和显示时间信息的示例。

[0308] 图22示出基于与叠加有关的元数据的360视频传输/接收方法的总体视图。

[0309] 参考图22,如上所述,360视频发送设备可以对360视频执行投影过程,并且在需要时执行按区域打包的过程。因此,可以导出2D图片。可以通过执行编码过程以比特流格式对

2D图片进行编码。

[0310] 同时,可以通过执行编码过程来附加地编码用于叠加的叠加图像(叠加视觉媒体)。此后,当生成与叠加有关的元数据时,将生成的元数据与编码过程的输出一起封装为媒体文件格式。然后,可以将处理后的媒体文件发送到360视频接收设备。

[0311] 360视频接收设备可以接收媒体文件,并且然后可以通过执行解封装过程来获取与360视频有关的比特流(编码的360视频信息)、与叠加图像有关的比特流(编码的叠加图像信息)和元数据。360视频接收设备可以基于与360视频有关的比特流(编码的360视频信息)来执行解码过程,并且可以对与360图像/视频有关的2D图片进行解码。2D图片可以对应于投影图片或打包图片。另外,360视频接收设备可以基于与360视频有关的比特流(编码的360视频信息)来解码叠加图像。当基于与叠加有关的元数据来渲染360图像/视频时,360视频接收设备可以根据特定位置和特定顺序来叠加和渲染叠加图像。

[0312] 同时,可以基于叠加被定位(或放置)的位置来不同地或单独地定义元数据样本条目。

[0313] 例如,在将叠加放置(或定位)在球体区域的情况下,OverlayOnSphereSample可能存在于样本条目中。例如,这可以被包括,如下表所示。

[0314] [表5]

[0315]

```
Class OverlayOnSphereSample Entry(type)
    extends SphereRegionSampleEntry('ovsp', version=0, flags=0) {
        OverlayOnSphereConfigBox();
    }

Class OverlayOnSphereConfigBox extends FullBox('oscb', version=0, flags=0) {
    unsigned int(16) depth;    // 在组中合成的顺序
    unsigned int(1) billboards_flag; // 始终面向相机
    unsigned int(1) target_flag;    // 指示目标位置
    if(target_flag == 1) {
        signed int(32) target_azimuth;
        signed int(32) target_elevation;
    }
}
```

[0316] 每个字段的语义可以如以上呈现的表1中所定义。更具体地,与上面列出的字段具有相同名称的字段可以承载相同的信息。在以下描述中也相同。

[0317] 图23和图24是叠加被定位在球体区域上的情况的示例。在叠加被定位在球体区域上的情况下,可以显示叠加图像,如图23和图24中所示。

[0318] 作为另一示例,在将叠加被定位在当前视口上的情况下,OverlayOverViewportSample可以存在于样本条目中。可以如下表所示表示其示例。

[0319] [表6]

[0320]

```

Class OverlayOverViewportSample Entry(type)
    extends MetaDataSampleEntry('ovvi', version=0, flags=0) {

        OverlayOverViewportConfigBox ();

    }

Class OverlayOverViewportConfigBox extends FullBox('ovcb', version=0, flags=0) {
    unsigned int(16) reference_width;
    unsigned int(16) reference_height;
    unsigned int(16) top_left_x;
    unsigned int(16) top_left_y;
    unsigned int(16) 2d_width;
    unsigned int(16) 2d_height;
    unsigned int(16) depth;        //在组中合成的顺序
    unsigned int(32) tilt;
    if(stereoscopic_flag== 1) {
        unsigned int(1) relative_disparity_flag;
        if(relative_disparity_flag ==1) {
            signed int(16) disparity_in_percent;
        } else {
            signed int(16) disparity_in_pixels;
        }
    }

    unsigned int(1) target_flag;    // 指示目标位置
    if(target_flag == 1) {
        signed int(32) target_azimuth;
        signed int(32) target_elevation;
    }
}

```

[0321] 图25图示叠加被定位在当前视口上的情况。在叠加被定位在当前视口上的情况下,可以如图25中所示指示叠加图像。

[0322] 作为另一示例,在叠加被定位在球体或立方体内的3D空间中的情况下,OverlayIn3DSpaceSample可以存在于样本条目中。例如,这可以被表示,如下表所示。

[0323] [表7]

[0324]

```

Class OverlayIn3DSpaceSample Entry(type)
    extends MetaDataSampleEntry('ov3d', version=0, flags=0) {

        OverlayIn3DSpaceConfigBox ();

    }

Class OverlayIn3DSpaceConfigBox extends FullBox('o3cb', version=0, flags=0) {
    signed int(32) centre_x;
    signed int(32) centre_y;
    signed int(32) centre_z;
    if (range_included_flag) {
        unsigned int(32) width;
        unsigned int(32) height;
    }
    signed int(32) rotation_yaw;
    signed int(32) rotation_pitch;
    signed int(32) rotation_roll;

    unsigned int(1) billboards_flag; // 始终面向相机

    unsigned int(1) target_flag;    // 指示目标位置
    if(target_flag == 1) {
        signed int(32) target_azimuth;
        signed int(32) target_elevation;
    }
}

```

[0325] 图26是叠加被定位在球体或立方体内的3D空间中的情况的示例。在叠加被定位在球体或立方体内的3D空间中的情况下,叠加图像可以如图26中所示被指示。

[0326] 同时,叠加元数据轨道可以包括关于用于叠加渲染的叠加位置、大小、属性(不透明度、交互作用等)的信息。叠加的渲染元数据(与叠加有关的元数据)可以随时间改变,并且在这种情况下,如上所述,可以将叠加的渲染元数据存储为定时元数据。

[0327] 图27示出包括与叠加有关的元数据的示例性叠加元数据轨道。

[0328] 下表中示出与叠加图像(媒体)叠加的位置相关信息的示例。

[0329] [表8]

[0330]

```
aligned(8) class OverlayPosStruct() {
    unsigned int(8) region_type;

    if(region_type == 0) {
        ViewportOverlayRegion();
    } else if (region_type == 1) {
        SphereOverlayRegion();
    } else if (region_type == 2) {
        3DOverlayRegion();
    }
}
```

[0331] 这里,region_type字段指示层位置类型。例如,当region_type字段的值等于0时,这可以指示在用户视口上显示叠加。并且,当region_type字段的值等于1时,这可以指示叠加被定位在球体上,并且当region_type字段的值等于2时,这可以指示叠加被定位在3D空间中。

[0332] 例如,在叠加被定位在用户视口上的情况下,与叠加有关的元数据可以包括下述。

[0333] [表9]

[0334]

```
aligned(8) class ViewportOverlayRegion() {
    unsigned int(16) rect_left_percent;
    unsigned int(16) rect_top_percent;
    unsigned int(16) rect_width_percent;
    unsigned int(16) rect_height_percent;
    unsigned int(16) order;

    unsigned int(1) stereoscopic_flag;
    bit(7) reserved=0;

    if (stereoscopic_flag == 1) {
        unsigned int(1) relative_disparity_flag;

        if (relative_disparity_flag == 1) {
            signed int(16) disparity_in_percent;
        } else {
            signed int(16) disparity_in_pixels;
        }
    }
}
```

[0335] ViewportOverlayRegion()指示其中叠加了叠加媒体的与位置有关的信息。

[0336] 在此,因为矩形平面的左上点的位置信息(rect_left_percent和rect_top_percent字段)和矩形平面的大小信息(rect_width_percent字段、rect_height_percent字段)可以根据显示器的大小而变化,所以值可以用百分比来指定。

[0337] 顺序字段指定叠加的顺序,以准备叠加(或重叠)的情况。并且,当执行渲染时,接收器可以控制顺序或可以控制位移(或定位)值。

[0338] stereoscopic_flag字段对应于指示是否支持立体声的标志信息。

[0339] relative_disparity_flag字段对应于指示在支持立体声时是否指配相对视差值的标志信息。

[0340] disparity_in_percent字段指示相对视差值。

[0341] disparity_in_pixels字段指示像素单位视差值。

[0342] 图28是叠加被定位在视口上的情况的示例。通过使用与叠加有关的元数据,可以将叠加定位在视口上,如附图中所示。

[0343] 作为另一示例,在叠加被定位在球体上的情况下,与叠加有关的元数据可以包括以下内容。

[0344] [表10]

[0345]

```
aligned(8) class SphereOverlayRegion() {
    unsigned int(1) proj_shape=0;

    if (proj_shape == 1) {
        unsigned int(32) proj_reg_width_percent;
        unsigned int(32) proj_reg_height_percent;
        unsigned int(32) proj_reg_top_percent;
        unsigned int(32) proj_reg_left_percent;

    } else if (proj_shape == 2) {
        unsigned int(8) num_rings;
        unsigned int(8) num_sectors;
        for (i = 0; i < num_rings; i++) {
            for (j = 0; j < num_sectors; j++) {
                unsigned int(16) proj_points_x[i][j];
                unsigned int(16) proj_points_y[i][j];
            }
        }
        unsigned int(3) transform_type[i];
        bit(5) reserved = 0;
        for (i = 0; i < num_rings; i++) {
            for (j = 0; j < num_sectors; j++) {
                unsigned int(16) proj_points_x[i][j];
                unsigned int(16) proj_points_y[i][j];
            }
        }
    } else if (proj_shape == 0) {
        unsigned int(8) shape_type;
        signed int(32) centre_azimuth;
        signed int(32) centre_elevation;
        unsigned int(32) azimuth_range;
        unsigned int(32) elevation_range;
        signed int(32) centre_tilt;

    }

    unsigned int(1) interpolate;
    unsigned int(16) depth;
}
```

[0346] SphereOverlayRegion() 指示与叠加媒体进行叠加的与位置有关的信息。

[0347] proj_shape字段指示投影的形状。如果proj_shape字段的值等于0,则指示无,如果proj_shape字段的值等于1,则指示矩形,并且proj_shape字段的值等于2,这指示多边形。

[0348] 如果投影形状是矩形,并且如果其位置由投影图片内的位置值来指定,则可以以投影图片的大小参考百分比来指定位置(proj_reg_top_percent字段,proj_reg_left_percent字段)和大小(proj_reg_width_percent字段,proj_reg_height_percent字段)。

[0349] 在投影形状是多边形的情况下,投影图片内的位置值可以由投影图片内的水平分割区域的数量(num_rings字段)和垂直分割区域的数量(num_sectors字段)来指定。

[0350] proj_points_x[i][j]字段和proj_points_y[i][j]字段分别指示每个分割点的投影图像内的位置值。

[0351] packed_points_x[i][j]字段和packed_points_y[i][j]字段分别指示每个分割点的打包图片内的位置值。

[0352] 在proj_shape字段的值等于0的情况下,指定球体上位置表达类型(基于shape_type字段)。如果shape_type字段的值等于0,则形状可以由4个大圆构成,并且如果shape_type字段的值等于1,则形状可以由2个方位角圆和2个仰角圆构成。

[0353] 为了指定叠加的中心位置处的方位角和仰角的值和大小(centre_azimuth字段, center_elevation字段),可以指定方位角/仰角范围(azimuth_range字段,elevation_range字段)和从叠加的中心点到球体原点的向量作为轴的周围的旋转值(centre_tilt字段)。

[0354] interpolate字段指示用于通过填充改变的(或变化的)值之间的值来执行平滑改变的标志信息。

[0355] depth字段指示从原点到叠加中心点的距离(深度)值,以便指示叠加的顺序,以准备叠加重叠的情况。

[0356] 图29是叠加被定位在球体上的情况的示例。如附图中所示,可以通过使用与叠加有关的元数据将叠加定位在球体上。

[0357] 作为又一示例,在叠加被定位在3维(3D)空间上的情况下,与叠加有关的元数据可包括以下内容。

[0358] [表11]

[0359]

```
aligned(8) class Overlay3DPositionStruct() {
    signed int(32) overlay_pos_x;
    signed int(32) overlay_pos_y;
    signed int(32) overlay_pos_z;
}

aligned(8) class OverlayRotationStruct() {
    signed int(32) overlay_rot_yaw;
    signed int(32) overlay_rot_pitch;
    signed int(32) overlay_rot_roll;
}

aligned(8) class 3DOverlayRegion() {
    unsigned int(32) width;
    unsigned int(32) height;
    bit(7) reserved=0;
    unsigned int(1) interpolate;
    Overlay3DPositionStruct();
    OverlayRotationStruct();
}
```

[0360] 3DOverlayRegion() 指示与叠加媒体进行叠加的与位置有关的信息。

[0361] 可以假设叠加平面具有矩形形状,并且矩形平面的尺寸可以基于y轴和z轴平面通过“width”字段和“height”字段来确定。另外,可以基于球坐标系内的位置(Overlay3DPositionStruct())的x、y和z(overlay_pos_x字段,overlay_pos_y字段,overlay_pos_z字段)中的每一个来重定位矩形平面。

[0362] 可以将叠加中心设置为中心,并且可以通过overlay_rot_yaw字段、overlay_rot_pitch字段和overlay_rot_roll字段来指示基于分别平行于球体的每个轴的叠加坐标系的每个轴的旋转。

[0363] interpolate字段指示用于通过填充改变的(或变化的)值之间的值来执行平滑改变的标志信息。

[0364] 图30是叠加被定位在3维(3D)空间上的情况的示例。如附图中所示,可以通过使用与叠加有关的元数据将叠加定位在3D空间上。

[0365] 图31示出当叠加存在于球体内的3维(3D)空间中时的叠加的位置/大小/旋转。如附图中所示,可以基于被包括在上述与叠加有关的元数据中的overlay_pos_x字段、overlay_pos_y字段、overlay_pos_z字段、width字段、height字段、overlay_rot_yaw字段、overlay_rot_pitch字段和/或overlay_rot_roll字段等来推导叠加的位置、大小和旋转。

[0366] 此外,与叠加有关的元数据可以包括叠加渲染属性信息。叠加渲染属性信息可以指定当渲染叠加时叠加平面的不透明度、当在VR媒体上混合叠加时执行的渲染选项以及焦点效果。

[0367] 图32示出叠加渲染属性的示例。

[0368] 例如,上述叠加渲染属性可以包括下述。

[0369] [表12]

[0370]

```
aligned(8) class OverlayRenderStruct() {
    unsigned int(1) opacity_info_flag;
    unsigned int(1) blending_flag;
    unsigned int(1) focus_flag;
    unsigned int(5) reserve =0;

    if(opacity_info_flag == 1) {
        unsigned int(8) opacity;
    }
    if (blending_flag == 1) {
        unsigned int(8) blending_type=1;
    }
    if (focus_flag == 1) {
        unsigned int(8) focus;
    }
}
```

[0371] OverlayRenderStruct() 指示在显示/渲染叠加显示时可以应用的渲染属性信息。

[0372] opacity_info_flag字段对应于指示是否指定叠加平面的整个不透明度的标志,并且opacity字段指定不透明度值。

[0373] blending_flag字段对应于指示在执行叠加合成时是否指定将要应用的混合的标志,并且blending_type字段指示混合类型。blending_type字段的值指示下述:1=source_over,2=source_atop,3=source_in,4=source_out,5=dest_atop,6=dest_over,7=dest_in,8=dest_out,9=clear,10=xor并且默认设置可以将source_over字段的值设置为1。

[0374] focus_flag字段对应于指示是否指定叠加焦点的标志,并且focus字段指示焦点值,并且focus字段的值可以在0至1.0的范围内。当在叠加上指定焦点时,模糊(blur)可能会被应用于由接收器渲染的另一个叠加以及VR媒体。

[0375] 如上所述,与叠加有关的元数据可以承载与叠加位置、大小、渲染特性和交互有关的信息。在这种情况下,与叠加有关的元数据可以包括下面例如在下表中所示的信息。

[0376] [表13]

[0377]

```
aligned(8) class OverlayInfoStruct() {
    unsigned int(32) overlay_id;
    unsigned int(16) overlay_source_id;
    unsigned int(1) overlay_essential_flag;
    unsigned int(1) overlay_priority;

    OverlayPosStruct();
    OverlayRenderStruct();
    OverlayMiscStruct();
    OverlayInteractionStruct();
}
```

[0378] OverlayInfoStruct() 指示叠加元数据。叠加元数据也可以被称为与叠加有关的元数据。

[0379] overlay_id 字段指示叠加元数据标识符。

[0380] overlay_source_id 字段指示叠加媒体源数据标识符。

[0381] overlay_essential_flag 字段对应于标志信息, 该标志信息指示叠加是否应必要地被叠加。

[0382] overlay_priority 字段指示当叠加媒体被叠加时的优先级。

[0383] OverlayPosStruct 可以指示与叠加位置有关的信息。

[0384] OverlayRenderStruct 可以指示与叠加渲染属性有关的信息。

[0385] OverlayMiscStruct 可以指示叠加渲染其它信息。

[0386] OverlayInteractionStruct 可以指示叠加交互信息。

[0387] 同时, 动态叠加元数据 (与动态叠加有关的信息) 可以由定时元数据构成。

[0388] 图33示出动态叠加元数据的示例性配置。

[0389] 例如, 在动态叠加元数据由定时元数据构成的情况下, 如图33中所示, 定义 OverlaySampleEntry, 并且 OverlaySampleEntry 继承 MetadataSampleEntry 并且可以调用 OverlayConfigBox。静态叠加渲染元数据可以在 OverlayConfigBox 中被定义。实际的动态叠加元数据可以被存储在样本中。OverlaySample 可以由对应于叠加的数量的 OverlayInfoStruct 构成。可以在下面的表14和15中指示。

[0390] [表14]

[0391]

```
aligned(8) class OverlayInfoStruct() {
    unsigned int(32) overlay_id;
    unsigned int(16) overlay_source_id;
    unsigned int(1) overlay_essential_flag;
    unsigned int(1) overlay_priority;

    OverlayPosStruct();
    OverlayRenderStruct();
    OverlayMiscStruct();
    OverlayInteractionStruct();
}
```

[0392] [表15]

[0393]

```
class OverlaySample {
    unsigned int(16) num_overlays;
    for (i == 0; i < num_overlays; i++) {
        OverlayInfoStruct();
    }
}
```

[0394] 为了支持叠加媒体的叠加位置或渲染属性等随时间改变的情况, 可以将叠加元数据存储于单独的轨道中, 并且然后被传递。相应的叠加媒体元数据轨道可以包括一个或多个

个样本,并且每个样本可以包括一个或多个叠加元数据。每个样本可以包括一个或多个OverlayInfoStruct。

[0395] 可以通过使用叠加元数据轨道的TrackReferenceTypeBox来指示叠加媒体轨道。更具体地,通过分配“cdsc”作为参考类型值,并且通过将一个或多个叠加媒体轨道标识符或轨道组标识符(经由一个或多个轨道传递叠加媒体的情况)指示给track_ID,可以指示与叠加元数据相关联的叠加媒体轨道。

[0396] 图34示出动态叠加元数据轨道和叠加媒体轨道链路信令的示例。

[0397] 同时,还可以通过使用下面的图35中所示的方法在叠加媒体轨道上用信号发送叠加元数据。

[0398] 图35示出在叠加媒体轨道上的叠加元数据信令的示例。

[0399] 参考图35,OverlayConfigBox可以被包括在叠加媒体轨道的样本条目中。因此,相应的媒体轨道可以包括叠加媒体,并且可以用信号发送轨道中包括的与叠加媒体有关的元数据。

[0400] 在这种情况下,与叠加有关的元数据可以包括下表中所示的信息。

[0401] [表16]

[0402]

```
class OverlayConfigBox extends FullBox('ovcf', 0, 0) {  
    unsigned int(8) num_overlays;  
    OverlayMediaPackingStruct(num_overlays)  
    for (i=0;i< num_overlays;i++) {  
        OverlayInfoStruct();  
    }  
}
```

[0403] 在此,num_overlay可以指示在叠加媒体轨道的每个样本中包括的叠加媒体的数量或者在样本中包括的叠加媒体的最大数量。

[0404] OverlayMediaPackingStruct() 指定叠加媒体的投影和打包信息。

[0405] OverlayInfoStruct() 可以包括上面提出的叠加元数据。这可以应用于被包括在轨道的样本中的叠加媒体。

[0406] 叠加媒体轨道可以包括具有grouping_type值为“ovgr”的SampleToGroupBox。

[0407] SampleToGroupBox可以指示在包括在相应轨道中的样本当中的要对其应用相同的叠加元数据的样本。

[0408] 如果在相应的轨道中存在具有grouping_type值为“ovgr”的SampleToGroupBox,则可能存在具有grouping_type值为“ovgr”的SampleGroupDescriptionBox,并且可以包括通常应用于相应样本的下述信息。可以如下所示定义具有grouping_type值为“ovgr”的样本组条目(以下称为OverlayGroupEntry)。

[0409] [表17]

[0410]

```
class OverlayGroupEntry() extends SampleGroupDescriptionEntry('ovmm') {  
    OverlayInfoStruct();  
}
```

[0411] 这里,OverlayInfoStruct() 可以包括将被应用于组中包括的样本的叠加元数据。

[0412] 叠加媒体轨道可以在样本条目中包括OverlayConfigBox,并且同时,叠加媒体轨

道还可以包括均具有“ovgr”的grouping_type值的SampleToGroupBox和OverlayGroupEntry()。在这种情况下,可以应用包括与OverlayGroupEntry()相关联的叠加媒体样本的叠加元数据。

[0413] 可替代地,为了一起指定叠加默认渲染信息以及叠加媒体轨道内的投影和打包信息,可以将传递方法更改为在OverlayConfigProperty(例如,图36)或OverlayConfigBox(例如,图37)内部定义与轨道中存在的叠加的数量相对应的num_overlay,并且然后将其作为参数传递的方法,并且然后可以添加OverlayInfoStruct()。图36和图37示出叠加媒体打包、投影和默认渲染信令的示例。图36示出叠加媒体轨道对应于图像的情况,并且图37示出叠加媒体轨道对应于视频的情况。

[0414] 例如,图36和图37的OverlayMediaPackingStruct可以包括下表中所示的信息。

[0415] [表18]

[0416]

```
aligned(8) class OverlayMediaPackingStruct(num_overlays) {
    unsigned int(5) num_regions;
    for(i = 0; i < num_regions; i++) {
        unsigned int(8) overlay_region_id[i];
        unsigned int(16) overlay_region_width[i];
        unsigned int(16) overlay_region_height[i];
        unsigned int(16) overlay_region_top[i];
        unsigned int(16) overlay_region_left[i];
    }
    for (i = 0; i < num_overlays; i++) {
        unsigned int(16) overlay_source_id[i];
        unsigned int(5) projection_type[i];
        unsigned int(3) packing_type;
        unsigned int(8) overlay_region_id[i];
        if(packing_type != 0) {
            unsigned int(1) guard_band_flag[i];
            if (packing_type == 1)
                TextureAtlasPacking(i);
            else if (packing_type == 2)
                RectRegionPacking(i);
            else if (packing_type == 3)
                PolygonRegionPacking(i);
            if (guard_band_flag[i])
                GaurdBand(i);
        }
    }
}
```

[0417] 另外,与叠加有关的元数据可以包括凝视信息。例如,可以如下表所示包括凝视信息。

[0418] [表19]

[0419]

```

class GraphicsSampleEntry() extends SampleEntry ('graf') {
    string    namespace;
    string    schema_location;           // 可选的
    string    auxiliary_mime_types;      // 可选的, 如果存在辅助资源则是必要的
    BitRateBox ();                       // 可选的
    GraphicConfigBox();                  // 可选的
}
class GraphicConfigBox extends FullBox('grcb', version = 0, flags) {
    unsigned int(8) display_mode;
    unsigned int(1) stereoscopic_flag;
    bit(7) reserved = 0;
    if(display_mode==1){ // 球体上固定的位置
        ...
    }
    else if(display_mode==0 || display_mode==2 || display_mode==3){ // HMD、视口或凝视
        unsigned int(1) window_flag;
        bit(7) reserved = 0;
        if(window_flag==1){
            ...
        }
        if(display_mode==3){ // 凝视
            signed int(32) x_offset;
            signed int(32) y_offset;
            signed int(32) z_offset;
            ...
        }
    }
}

```

[0420] 这里,例如,在使用能够执行眼睛凝视跟踪的设备的情况下,display_mode=3可以指示叠加被定位为在眼睛的瞳孔凝视的区域附近的偏移。凝视信息可以包括用于基于例如凝视位置来指示相对叠加渲染位置的x_offset字段、y_offset字段和z_offset字段中的至少一个。x_offset字段、y_offset字段和z_offset字段的值可以指示角度并且也可以指示百分比。例如,在x_offset字段、y_offset字段和z_offset字段的值指示角度的情况下,可以将叠加定位在球体上。作为另一个示例,在x_offset字段、y_offset字段和z_offset字段的值指示百分比的情况下,可以将叠加定位在视口上。

[0421] 可以基于凝视信息来执行基于凝视的(叠加)图像渲染。基于凝视的图像渲染对应于提取实际用户的凝视点(x,y,z)并基于所提取的凝视点来渲染与凝视相对应的图像。类似地,在本发明中提到的基于凝视的图形(叠加)渲染对应于基于用户的视口来确定用户的凝视点,并且然后渲染该图形(叠加图像)。为了获取有关实际用户的视口和凝视的信息,将会接收HMD的与位置有关的传感器值,并且然后应基于接收到的传感器值执行图形(叠加图像)的叠加。更具体地,可以通过使用用户的实际视口内的凝视点将图形(叠加图像)渲染到相应点。因此,将其起点(或在某些情况下为图形区域的中点)设置为由传感器计算出的基于凝视点(x,y,z)的(x+x_offset,y+y_offset,z+z_offset)的图形渲染区域可以被产生。如果打算将图形定位在凝视点所位于的点,则x、y和z中的每一个的偏移值都可以作为0来传递。但是,在这种情况下,因为凝视信息不断变化(为了读取文本等),所以可以设置凝视位置值的阈值,而不是连续使用传感器值,并且可以仅当凝视的运动超出阈值范围时将图形渲染到不同的位置。可替换地,可以仅在过了预定时间段的情况下,将图形区域设置为通过凝视点的移动而改变。

[0422] 作为参考,可以通过使用以下信息来计算凝视点(x,y,z)。

[0423] [表20]

[0424] 头盔位置的x坐标

头盔位置的y坐标
头盔位置的z坐标
观看者的左眼局部坐标的x坐标
观看者的左眼局部坐标的y坐标
观看者的右眼局部坐标的x坐标
观看者的右眼局部坐标的y坐标
沿着观看者的局部z轴的焦点距离

[0425] 图38是凝视点计算的示例。基于表19中公开的信息，可以如下图38中所示计算凝视点。

[0426] 为了基于HMD、视口和凝视来执行图形叠加，应获知实际显示于相应HMD的画面、实际视口、实际凝视等的信息。然而，难以预先确定用户将实际凝视(或看着)的位置。另外，也难于确定关于是否基于整个360视频创建相应的图形，或者是否仅基于360视频的一部分创建相应的图形的信息。因此，需要关于图形的尺寸/位置的信息以及指示基于其创建图形的窗口的信息。如果实际视口正在观看小于参考窗口的大小的图像，则可以根据成比例的比率适当地减小现有图形或位置信息的大小，并且在某些情况下，位置可以被固定，并且大小可能会减小。可替代地，不管创建的窗口如何，图形大小或位置都可以被固定。更具体地，因为可能发生各种情况，所以可以包括关于窗口的信息或指示是否要改变位置或大小的信令信息。更具体地，取决于包括在PlaneRegionConfigBox()中的与窗口有关的信息以及在实际视口/凝视/HMD上显示的实际窗口的大小，图形区域的大小或位置可以是固定的，也可以按照成比例的比率被放大或缩小。在此，如上所述，图形可以对应于叠加媒体或叠加图像。

[0427] 还可以基于DASH用信号发送根据本发明的上述与叠加有关的元数据。当提供基于DASH的服务时，如上定义的GraphicConfigBox()中包括的字段可以作为其属性被包括在MPD中的Period元素、AdaptationSet元素、Representation元素、SubRepresentation元素、ContentComponent元素、SegmentBase元素、EssentialProperty元素、SupplementaryProperty元素、Accessibility元素和Subset元素中的至少一个中。在以下示例中，基本属性和/或补充属性描述符仅包括display_mode和stereoscopic_flag。

[0428] [表21]

[0429]	具有等于“urn:mpeg:dash:graphic:2017”的@schemeIdURI和@value属性的基本属性和/或补充属性描述符将会被用于用信号发送被关联到图形数据的元数据。@value语法将会如下面的ABNF中所描述。
--------	---

[0430]

@value = "display-mode" ":" display-mode "," "stereoscopic-flag" ":" s display-mode = (%d1-%d64) stereoscopic-flag = BIT; default value 0
--

[0431] 在自适应集中,可以包括图形叠加信息作为辅助或可访问性信息,并且表示 (Representation) 可以通知图形数据被包括。示例性实施例如下所示。此时,可以包括@mimeType (例如:应用程序/比特图) 和角色 (Role) 描述符 (例如,@schemeIdUri = "urn:mpeg:dash:role:2011" 和@value = "graphic") 等等。在定义了用于图形的编解码器参数的情况下,则可以被另外地添加。

[0432] [表22]

[0433]

<AdaptationSet contentType="graphic" mimeType="application/bitmap"> <EssentialProperty schemeIdUri="urn:mpeg:dash:graphic:2017" value="2, 0"/> <Role schemeIdUri="urn:mpeg:dash:role:2011" value="graphic"/> <Representation id="graphic" bandwidth="1000"> </Representation> </AdaptationSet>

[0434] 可以基于上述叠加元数据对360视频/图像执行叠加的叠加渲染。

[0435] 图39示出叠加渲染过程的示例。

[0436] 360视频接收设备解析叠加元数据并获取与叠加有关的信息 (S3900)。更具体地说,例如,360视频接收设备可以获取OverlayPosStruct (区域类型,区域位置/大小/旋转) 和与叠加内容类型有关的信息 (文本、图像、视频、推荐视口等)。

[0437] 360视频接收设备根据region_type生成叠加几何图形 (S3910)。region_type对应于指示如上所述的叠加类型 (或渲染类型) 的信息。将参考图40详细描述步骤S3910的详细示例。

[0438] 360视频接收设备根据叠加内容类型生成叠加纹理 (S3920)。叠加纹理可以包括文本、图像/视频和推荐的视口。将参考图41详细描述步骤S3920的详细示例。

[0439] 360视频接收设备可以存储和渲染/显示所生成的叠加几何图形/纹理。为此,360视频接收设备可以存储在叠加结构 (Struct) 中解析的元数据和所生成的叠加几何图形/纹理,并且可以将存储的几何图形/纹理注册在叠加列表中 (S3930)。

[0440] 图40示出根据叠加类型的示例性叠加几何图形生成。

[0441] 参考图40,在叠加类型指示当前视口上的叠加的情况下 (例如,在叠加region_type等于0的情况下),360视频接收设备生成用于存储在矩形的各个顶点处的x、y值的几何缓冲区 (例如,geo[4][2])。360视频接收设备可以将以百分比指定的叠加区域 (位置/大小) 信息更改为x范围 (-1到1) 和y范围 (-1到1) 值,并且然后将更改的值存储在缓冲区。

[0442] 在叠加类型指示360表面上的叠加的情况下 (例如,在叠加region_type等于1的情况下),360视频接收设备可以确定投影类型是否为ERP,并且在投影类型为ERP的情况下,360视频接收设备可以根据球体表面的区域信息 (中心方位角/仰角、方位角/仰角范围) 值和复杂度 (环数、切片数) 来确定要生成多少个点。此后,360视频接收设备生成用于存储每

个点的x、y、z值的几何缓冲区(例如,geo[#ring][#slice][3])。

[0443] 在叠加类型指示360表面内的3D空间中的叠加的情况下(例如,在叠加region_type等于2的情况下),或者在叠加类型指示360表面上的叠加和投影类型不是ERP的情况下,360视频接收设备会生成几何缓冲区,用于存储在矩形的每个顶点处的x、y值(例如,geo[4][2])。360视频接收设备将区域信息(中心方位角/仰角、方位角/仰角范围)值改变为x、y、z坐标值,并且然后可以将改变后的值存储在缓冲区中。

[0444] 图41示出根据叠加内容类型的示例性叠加纹理生成。

[0445] 参考图41,在叠加内容类型指示文本的情况下,360视频接收设备在考虑文本长度和字体以及字体大小的情况下计算纹理缓冲区大小,并生成缓冲区。360视频接收设备可以将与文本相对应的区域的图像从字体图像复制到缓冲区。

[0446] 在叠加内容类型指示图像/视频的情况下,360视频接收设备计算必要的缓冲区大小并生成缓冲区。360视频接收设备可以将与图像/视频有关的像素复制到所生成的缓冲区。

[0447] 在叠加内容类型指示推荐的视口的情况下,360视频接收设备计算必要的缓冲区大小并生成可以存储渲染结果的帧缓冲区。

[0448] 图42示出叠加渲染示例。

[0449] 在存在帧渲染请求的情况下,360视频接收设备渲染用户视点360主内容。

[0450] 360视频接收设备确定推荐的视口是否被激活并且存在于叠加当中。如果推荐视口被激活并且存在,则360视频接收设备可以将相机和近/远平面位置更改为推荐视口位置,并且然后可以再次对预生成的帧缓冲区执行渲染,从而能够生成/注册叠加纹理。

[0451] 如果叠加类型指示360表面内部3D空间中的叠加(例如,如果叠加region_type等于2),则在叠加中存在旋转信息的情况下,360视频接收设备应用旋转,并可以将叠加渲染为预先生成的几何图形和纹理。

[0452] 如果叠加物类型指示360表面上的叠加(例如,如果叠加region_type等于1),则360视频接收设备可以将叠加渲染为预先生成的几何图形和纹理。

[0453] 如果叠加类型指示当前视口上的叠加(例如,如果叠加region_type等于0),则360视频接收设备可以将叠加渲染为预先生成的几何图形和纹理。

[0454] 根据上述本发明,可以基于叠加类型将叠加纹理(叠加媒体)有效地渲染到360内容至所需位置和大小,并且可以将必要信息有效地提供给用户。

[0455] 图43示出由根据本发明的360视频发送设备执行的360视频/图像数据处理方法的总体视图。图43所示的方法可以由图5或者图16中公开的360视频发送设备执行。

[0456] 参考图43,360视频发送设备获取360视频/图像(S4300)。360视频/图像可以对应于由至少一个相机捕获的视频/图像。可替代地,360视频/图像的部分或全部可以对应于由计算机程序生成的虚拟视频/图像等等。360图像可以对应于独立的静止图像或360视频的部分。

[0457] 360视频发送设备处理360视频/图像并导出图片(S4310)。360视频发送设备可以基于上述多样的投影格式、按区域打包过程等来导出基于2D的图片。导出的图片可以对应于投影的图片,或者可以对应于打包的图片(在应用按区域打包的情况下)。

[0458] 360视频发送设备生成与360视频/图像有关的元数据(S4320)。在此,元数据可以

包括在本说明书中以上描述的字段。这些字段可以被包含在具有各种级别的盒子中,或者可以作为数据被包括在文件内的单独轨道中。例如,元数据可以包括以上在表1至表19中描述的字段/信息的一部分或全部。例如,元数据可以包括上述与叠加有关的元数据(包括信息/字段)。与叠加有关的元数据可以包括关于上述叠加的区域的信息、关于叠加的(渲染)类型的信息、关于是否执行立体应用的信息、关于叠加的旋转的信息、指示距单位球体中心的距离的距离信息、指示叠加(叠加区域)深度的信息、关于叠加(分层)顺序的信息、目标插件信息、关于叠加数量的信息、叠加渲染属性信息和/或凝视信息等等中的至少一个。叠加的区域也可以被称为叠加区域。

[0459] 例如,与叠加有关的元数据可以包括关于叠加区域的信息。叠加区域可以指示叠加被定位的层或平面的区域。

[0460] 例如,可以基于方位角和仰角来指示关于叠加区域的信息。例如,关于叠加区域的信息可以包括关于叠加的位置的信息和关于叠加的大小的信息中的至少一个。并且,叠加的位置的信息可以指示在解码的图片被渲染到的3D坐标空间内的叠加的中心点的x坐标、y坐标和z坐标中的至少一个。并且,关于叠加的大小的信息可以指示叠加的宽度和高度中的至少一个。关于叠加区域的信息可以包括上述的sphere_region_flag字段、centre_azimuth字段、centre_elevation字段和/或centre_tilt字段。可替选地,关于叠加区域的信息可以包括上述centre_x字段、centre_y字段、centre_z字段、width字段和/或height字段中的至少一个。

[0461] 例如,与叠加有关的元数据可以包括关于叠加类型的信息。关于叠加类型的信息可以指示具有被渲染到3D空间的叠加类型的叠加类型、具有在视口上渲染的叠加类型的叠加类型或具有在球面上渲染的叠加类型的叠加类型。

[0462] 例如,关于叠加类型的信息可以指示具有被渲染到3D空间的叠加类型的叠加类型。在这种情况下,关于叠加类型的信息可以基于方位角和仰角指示叠加区域。在这种情况下,关于叠加区域的信息可以包括上述sphere_region_flag字段、centre_azimuth字段、centre_elevation字段和/或centre_tilt字段中的至少一个。另外,在关于叠加类型的信息指示被渲染到3D空间的叠加类型的情况下,与叠加有关的元数据可以进一步包括关于叠加旋转的信息。关于叠加旋转的信息可以指示与叠加旋转有关的偏航值、俯仰值和横滚值中的至少一个。另外,在关于叠加类型的信息指示被渲染到3D空间的叠加类型的情况下,与叠加有关的元数据可以进一步包括指示叠加区域或平面是否基于观看方位旋转的标志信息。通过使用此标志信息,可以根据用户视野的角度来旋转3D空间内的叠加/平面,从而允许用户面对叠加层/平面。标志信息可以包括上述的billboards_flag字段。另外,在关于叠加类型的信息指示被渲染到3D空间的叠加类型的情况下,与叠加有关的元数据还包括距离信息,该距离信息指示距单位球体的中心的距离。

[0463] 作为另一个示例,在关于叠加类型的信息指示在视口上渲染的叠加类型的情况下,或者在关于叠加类型的信息指示在球体表面上渲染的叠加类型的情况下,与叠加有关的元数据还可以包括指示叠加的区域的深度的信息。指示叠加的区域的深度的信息可以指示叠加的渲染优先级。例如,在多个叠加进行叠加的情况下,对于进行叠加的区域,基于指示叠加的区域的深度的信息,在多个叠加当中,可以渲染与较低深度值相关联的叠加。指示叠加的区域的深度的信息可以包括上述深度字段。另外,与叠加有关的元数据可以进一步

包括关于叠加的(分层)顺序的信息。例如,如果关于叠加类型的信息指示在视口上渲染的叠加类型,或者如果关于叠加类型的信息指示在球体表面上渲染的叠加类型,则在多个叠加进行叠加的情况下,关于叠加的(分层)顺序的信息可以指示多个进行叠加的叠加之间的优先级。换句话说,与叠加有关的元数据可以包括指示在均具有在视口上渲染的叠加类型的叠加之间的顺序的信息。

[0464] 例如,与叠加有关的元数据可以进一步包括指示叠加是否与参考点相关联的目标标志信息。可以通过叠加提供由参考点指示的图像、文本、声音或补充信息,或者可以渲染与由参考点指示的位置相对应的图像。目标标志信息可以包括上述target_flag字段。

[0465] 例如,与叠加有关的元数据可以包括关于叠加的数量的信息。在这种情况下,与叠加中的每个相对应的与叠加有关的信息可以被包括在与叠加有关的元数据中。

[0466] 另外,例如,与叠加有关的元数据可以包括叠加渲染属性信息。叠加渲染属性信息可以包括叠加的平面的不透明度、焦点效果和/或混合选项中的至少一项。

[0467] 与叠加有关的元数据可以被包括在叠加定时元数据轨道中。可替代地,可以基于叠加轨道和样本内的时间信息来配置与叠加有关的元数据。

[0468] 360视频发送设备对导出的图片进行编码(S4330)。360视频发送设备可以对2D图片进行编码并且以比特流格式输出编码的2D图片。

[0469] 360视频发送设备可以根据要叠加的纹理(媒体)类型对叠加纹理(媒体)进行编码和输出。在这种情况下,编码的叠加纹理(媒体)可以被包括在360图像/视频数据中,这将在后面更详细地描述。

[0470] 可替代地,要叠加的纹理(媒体)类型可以预先存储在360视频接收设备中,或者可以通过网络单独发送。

[0471] 360视频发送设备执行用于存储或发送编码的图片和元数据的处理(S4340)。360视频发送设备可以基于与编码的图片和/或元数据有关的数据来生成360图像/视频数据。在对与一系列图像相对应的一系列图片进行编码的情况下,可以生成包括编码的圖片的360视频数据。

[0472] 360视频发送设备可以以文件格式封装与编码的图片和/或元数据有关的数据,并且360视频发送设备可以以诸如ISOBMFF, CFF等的文件格式封装编码的360视频数据和/或元数据,以便于存储或发送编码的360视频数据和/或元数据,或者360视频发送设备可以以其他DASH片段格式来处理编码的360视频数据和/或元数据。360视频发送设备可以包括以文件格式的元数据。例如,元数据可以被包括在ISOBMFF文件格式内具有各种级别的盒子中,或者可以作为数据被包括在文件内的单独轨道中。

[0473] 如上所述,与叠加有关的元数据可以被包括在叠加定时元数据轨道中。可替代地,可以基于叠加轨道和样本内的时间信息来配置与叠加有关的元数据。

[0474] 此外,360视频发送设备可以将元数据本身封装到文件中。360视频发送设备可以根据文件格式被封装的360视频数据执行用于传输的处理。360视频发送设备可以根据随机(或任何)传输协议来处理360视频数据。用于传输的处理可以包括用于通过广播网络进行传递的处理、或者用于通过诸如宽带等通信网络进行传递的处理。另外,还可以对元数据执行用于传输的处理。360视频发送设备可以通过广播网络和/或宽带发送经过传输处理的360图像/视频数据(包括元数据)。

[0475] 图44示出由根据本发明的360视频接收设备执行的360视频/图像数据处理方法的总体视图。图44中所示的方法可以由图6或者图17中公开的360视频接收设备执行。

[0476] 参考图44,360视频接收设备接收360图像/视频数据(信号)(S4400)。360视频接收设备可以通过广播网络接收从360视频发送设备用信号发送的360图像/视频数据。360图像/视频数据可以包括关于360图像/视频的编码图片和元数据的信息。另外,360视频接收设备可以通过诸如宽带等的通信网络或者通过存储介质来接收360图像/视频数据。

[0477] 360视频接收设备获取关于编码的图片和元数据的信息(S4410)。360视频接收设备可以通过文件/片段解封装过程从360图像/视频数据获取关于编码的图片和元数据的信息。

[0478] 元数据可以包括以上在本说明书中描述的字段。这些字段可以被包括在具有各种级别的盒子中,也可以作为数据包括在文件内的单独轨道中。例如,元数据可以包括以上在表1至表19中描述的字段/信息的一部分或全部。例如,元数据可以包括上述与叠加有关的元数据(包括信息/字段)。与叠加有关的元数据可以包括关于上述叠加的区域的信息、关于叠加的(渲染)类型的信息、关于是否执行立体应用的信息、关于叠加的旋转的信息、指示距单位球体中心的距离的距离信息、指示叠加(叠加区域)深度的信息、关于叠加(分层)顺序的信息、目标插件信息、关于叠加数量的信息、叠加渲染属性信息和/或凝视信息等等中的至少一个。

[0479] 例如,与叠加有关的元数据可以包括关于叠加区域的信息。叠加区域可以指示叠加被定位的层或平面的区域。

[0480] 例如,可以基于方位角和仰角来指示关于叠加区域的信息。例如,关于叠加区域的信息可以包括关于叠加的位置的信息和关于叠加的大小的信息中的至少一个。并且,叠加的位置的信息可以指示在解码的图片被渲染到的3D坐标空间内的叠加的中心点的x坐标、y坐标和z坐标中的至少一个。并且,关于叠加的大小的信息可以指示叠加的宽度和高度中的至少一个。关于叠加区域的信息可以包括上述的sphere_region_flag字段、centre_azimuth字段、centre_elevation字段和/或centre_tilt字段。可替选地,关于叠加区域的信息可以包括上述centre_x字段、centre_y字段、centre_z字段、width字段和/或height字段中的至少一个。

[0481] 例如,与叠加有关的元数据可以包括关于叠加类型的信息。关于叠加类型的信息可以指示具有被渲染到3D空间的叠加类型的叠加类型、具有在视口上渲染的叠加类型的叠加类型或具有在球面上渲染的叠加类型的叠加类型。

[0482] 例如,关于叠加类型的信息可以指示具有被渲染到3D空间的叠加类型的叠加类型。在这种情况下,关于叠加类型的信息可以基于方位角和仰角指示叠加区域。在这种情况下,关于叠加区域的信息可以包括上述sphere_region_flag字段、centre_azimuth字段、centre_elevation字段和/或centre_tilt字段中的至少一个。另外,在关于叠加类型的信息指示被渲染到3D空间的叠加类型的情况下,与叠加有关的元数据可以进一步包括关于叠加旋转的信息。关于叠加旋转的信息可以指示与叠加旋转有关的偏航值、俯仰值和横滚值中的至少一个。另外,在关于叠加类型的信息指示被渲染到3D空间的叠加类型的情况下,与叠加有关的元数据可以进一步包括指示叠加区域或平面是否基于观看定向是否旋转的标志信息。通过使用此标志信息,可以根据用户视野的角度来旋转3D空间内的叠加/平面,从

而允许用户面对叠加/平面。标志信息可以包括上述的billboards_flag字段。另外,在关于叠加类型的信息指示被渲染到3D空间的叠加类型的情况下,与叠加有关的元数据还可以包括距离信息,该距离信息指示距单位球体的中心的距离。

[0483] 作为另一个示例,在关于叠加类型的信息指示在视口上渲染的叠加类型的情况下,或者在关于叠加类型的信息指示在球体表上渲染的叠加类型的情况下,与叠加有关的元数据还可以包括指示叠加的区域的深度的信息。指示叠加的区域的深度的信息可以指示叠加的渲染优先级。例如,在多个叠加进行叠加的情况下,对于进行叠加的区域,基于指示叠加的区域的深度的信息,在多个叠加当中,可以渲染与较低深度值相关联的叠加。指示叠加的区域的深度的信息可以包括上述深度字段。另外,与叠加有关的元数据可以进一步包括关于叠加的(分层)顺序的信息。例如,如果关于叠加类型的信息指示在视口上渲染的叠加类型,或者如果关于叠加类型的信息指示在球体表面上渲染的叠加类型,则在多个叠加进行叠加的情况下,关于叠加的(分层)顺序的信息可以指示多个进行叠加的叠加之间的优先级。换句话说,与叠加有关的元数据可以包括指示在均具有在视口上渲染的叠加类型的叠加之间的顺序的信息。

[0484] 例如,与叠加有关的元数据可以进一步包括指示叠加是否与参考点相关联的目标标志信息。可以通过叠加提供由参考点指示的图像、文本、声音或补充信息,或者可以渲染与由参考点指示的位置相对应的图像。目标标志信息可以包括上述target_flag字段。

[0485] 例如,与叠加有关的元数据可以包括关于叠加的数量的信息。在这种情况下,与叠加中的每个相对应的与叠加有关的信息可以被包括在与叠加有关元数据中。

[0486] 另外,例如,与叠加相关的元数据可以包括叠加渲染属性信息。叠加渲染属性信息可以包括叠加的平面的不透明度、焦点效果和/或混合选项中的至少一项。

[0487] 与叠加有关的元数据可以被包括在叠加定时元数据轨道中。可替选地,可以基于叠加轨道和样本内的时间信息来配置与叠加有关的元数据。

[0488] 360视频接收设备基于关于编码的图片的信息对编码的图片进行解码(S4420)。解码的图片可以对应于投影的图片,或者可以对应于打包的图片(在应用按区域打包的情况下)。

[0489] 360视频接收设备可以根据要叠加的纹理(媒体)类型解码叠加纹理(媒体)。在这种情况下,编码的叠加纹理(媒体)可以被包括在360图像/视频数据中。

[0490] 可替选地,要被叠加的纹理(媒体)类型可以被预先存储在360视频接收设备中,或者可以通过网络被单独接收。

[0491] 在某些情况下,360视频接收设备可能会基于元数据对图片进行解码。例如,这可以包括对在图片内视点被定位的区域的一部分执行解码的情况,或者需要对与具有改变的视点或被链接到叠加的位置相对应的另一特定图片进行解码的情况。

[0492] 360视频接收设备基于元数据来渲染解码的图片和叠加(S4430)。360视频接收设备可以基于元数据来处理和渲染解码的图片和叠加。在这种情况下,可以使用诸如重新投影等过程来处理解码后的图片,并且然后可以将其渲染到3D表面。在叠加的情况下,可以根据基于元数据的上述叠加类型,将叠加渲染到视口、3D表面、3D空间等上的位置。

[0493] 根据示例性实施例,可以省略上述处理步骤,或者可以通过执行类似/相同操作的处理步骤来代替上述处理步骤。

[0494] 以上图示的装置的内部组件可以是执行存储在存储器中的连续处理的处理器,或者可以是配置有其他硬件的硬件组件。这些部件可以被布置在装置内部或外部。

[0495] 根据实施例,前述模块可以被省略,或者可以由用于执行类似/等效操作的其他模块代替。

[0496] 前述部分、模块或单元中的每一个可以是执行存储在存储器(或存储单元)中的连续过程的处理器或硬件部分。前述实施例中描述的每个步骤可以由处理器或硬件部分执行。前述实施例中描述的每个模块/块/单元可以作为硬件/处理器操作。此外,本发明提出的方法可以被执行为代码。这些代码可以被写入处理器可读存储介质中,并且因此可以由装置提供的处理器读取。

[0497] 尽管前述实施例基于具有一系列步骤或块的流程图阐明方法,但是本发明不限于步骤或块的顺序。如上所述,一些步骤或块可以同时发生或以与上述其他步骤或块不同的顺序发生。此外,本领域的技术人员将理解,在不影响本公开的范围的情况下以上流程图所示的步骤不是排他的,可以包括其他步骤,或者删除流程图中的一个或多个步骤。

[0498] 当本发明的实施例以软件实现时,前述方法可以由执行上述功能的模块(过程、功能等)来实现。这样的模块可以存储在存储器中并且可以由处理器执行。存储器可以在处理器内部或外部,并且可以使用各种众所周知的手段连接到处理器。处理器可以包括专用集成电路(ASIC)、其他芯片组、逻辑电路和/或数据处理设备。存储器可以包括只读存储器(ROM)、随机存取存储器(RAM)、闪存、存储卡、存储介质和/或其他存储设备。

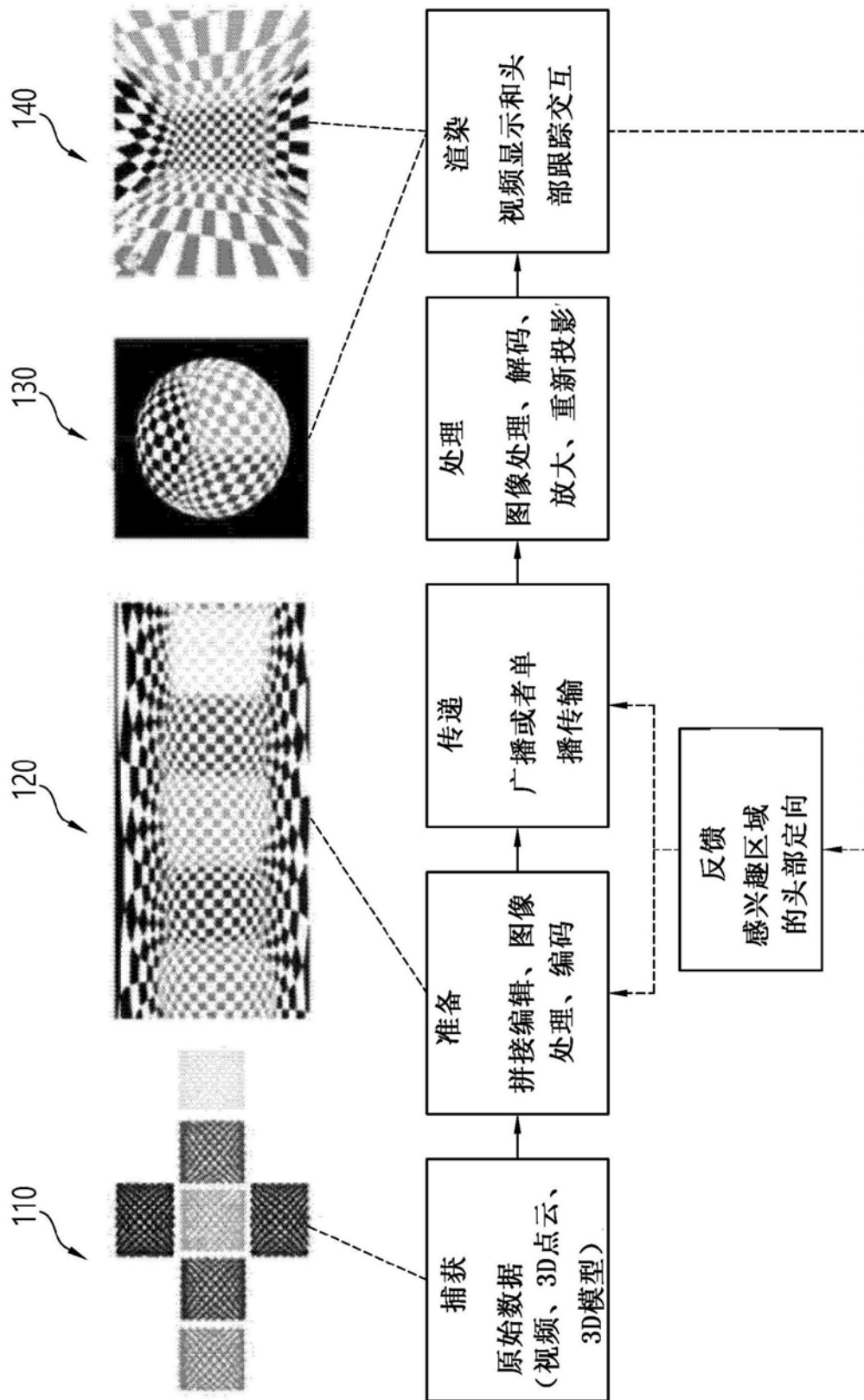
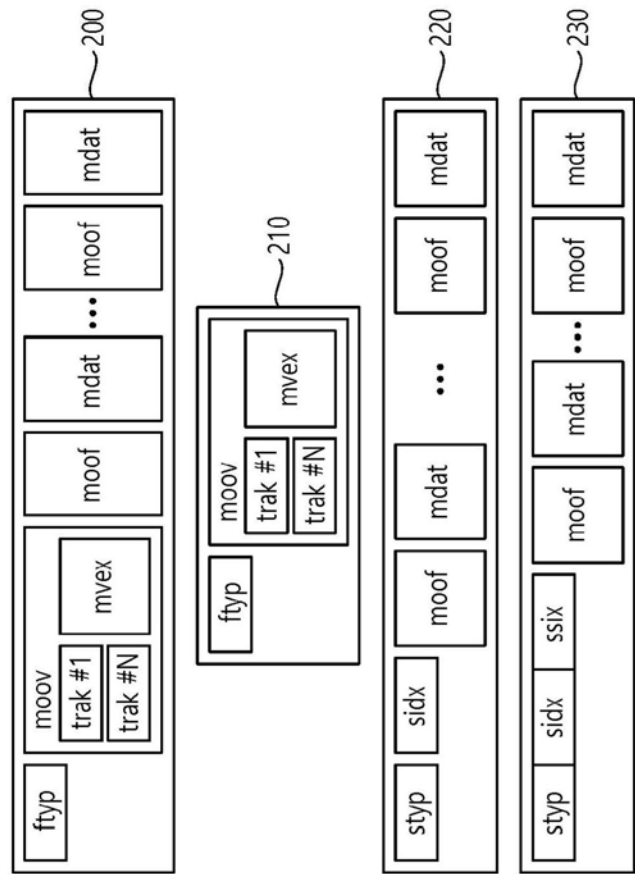


图1



```
250 {
    aligned(8) class Box (unsigned int(32) boxtype, optional
    unsigned int(8)[16] extended_type)
    {
        unsigned int(32) size;
        unsigned int(32) type = boxtype;
        if (size==1) {
            unsigned int(64) largesize;
        } else if (size==0) {
            // 盒子扩展到文件末端
        }
        if (boxtype=='uuid') {
            unsigned int(8)[16] usertype = extended_type;
        }
    }
}

aligned(8) class FullBox(unsigned int(32) boxtype, unsigned int(8)
v, bit(24) f) extends Box(boxtype) {
    unsigned int(8)    version = v;
    bit(24)            flags = f;
}
```

图2

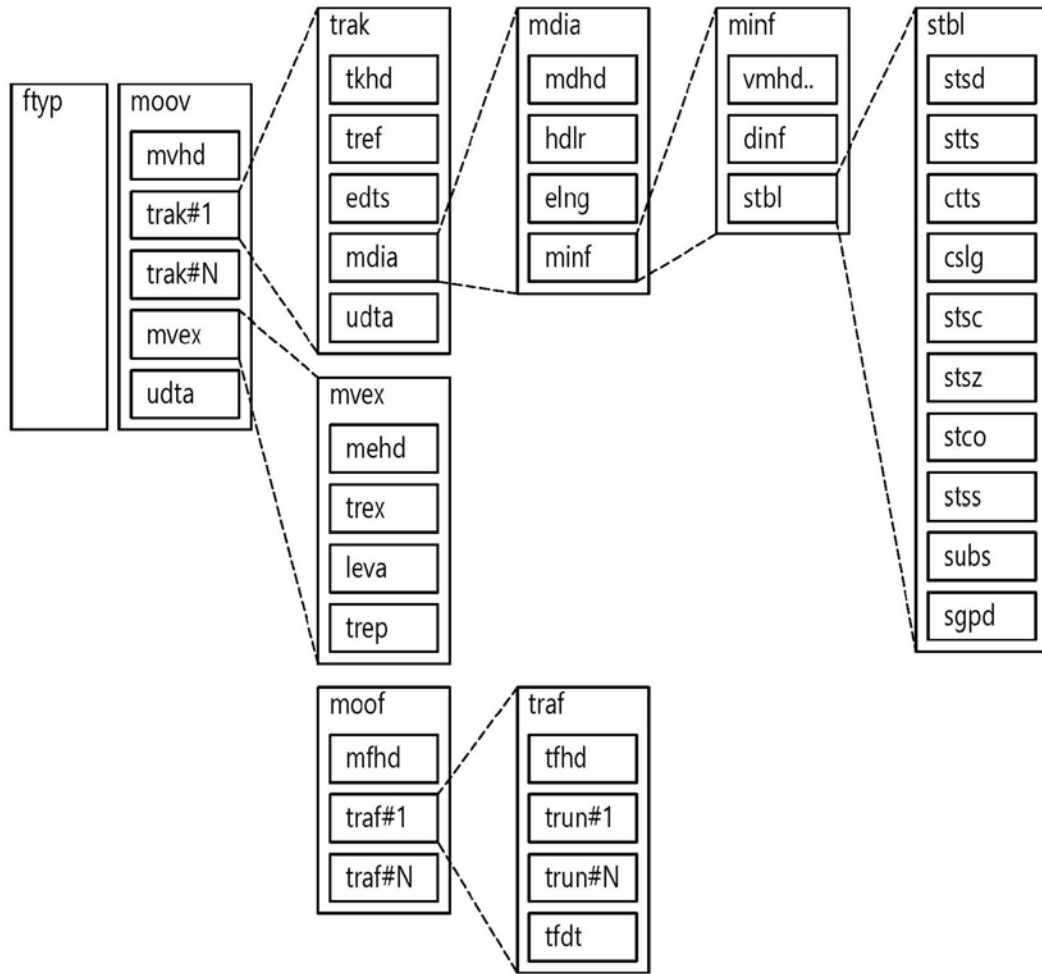


图3

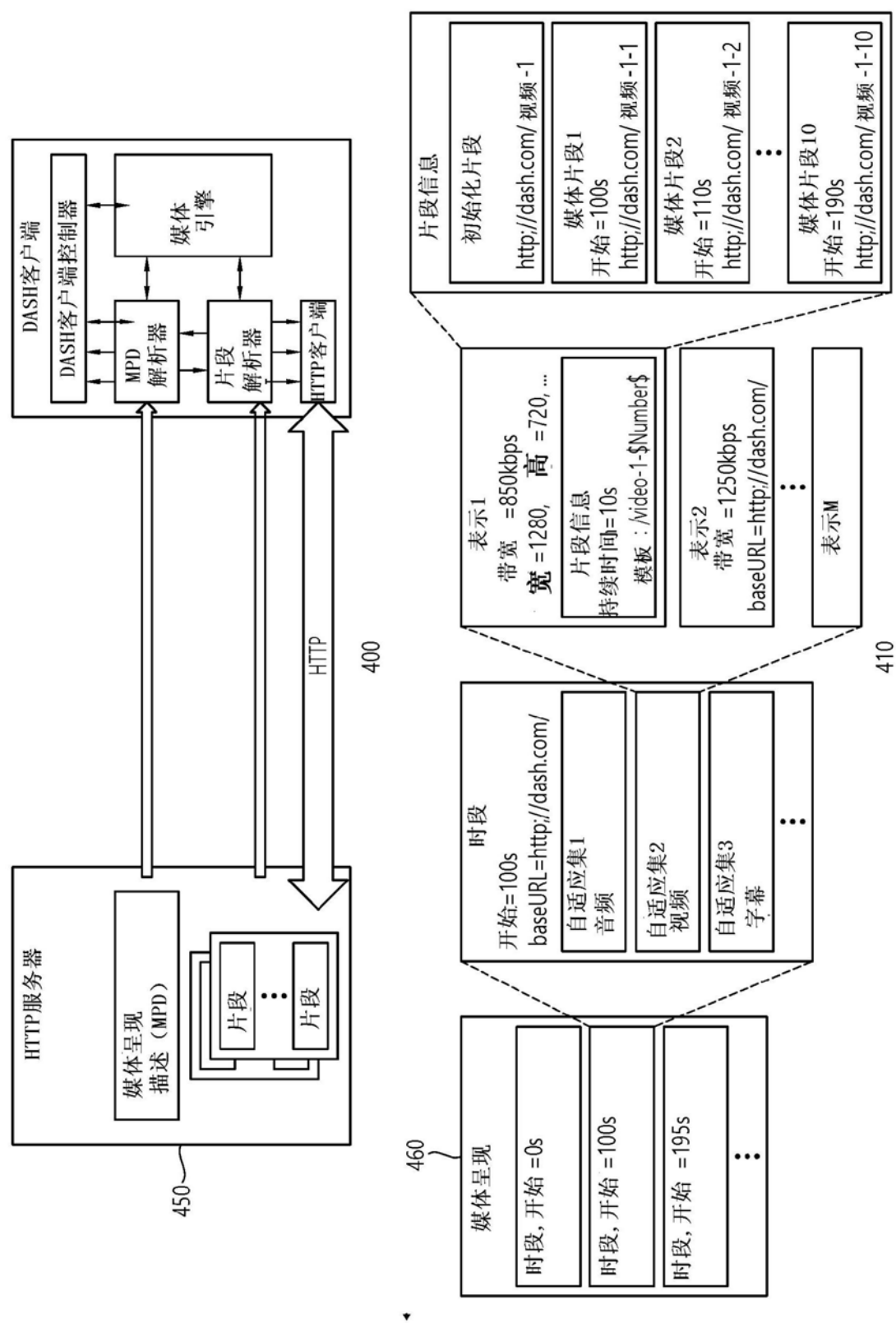


图4

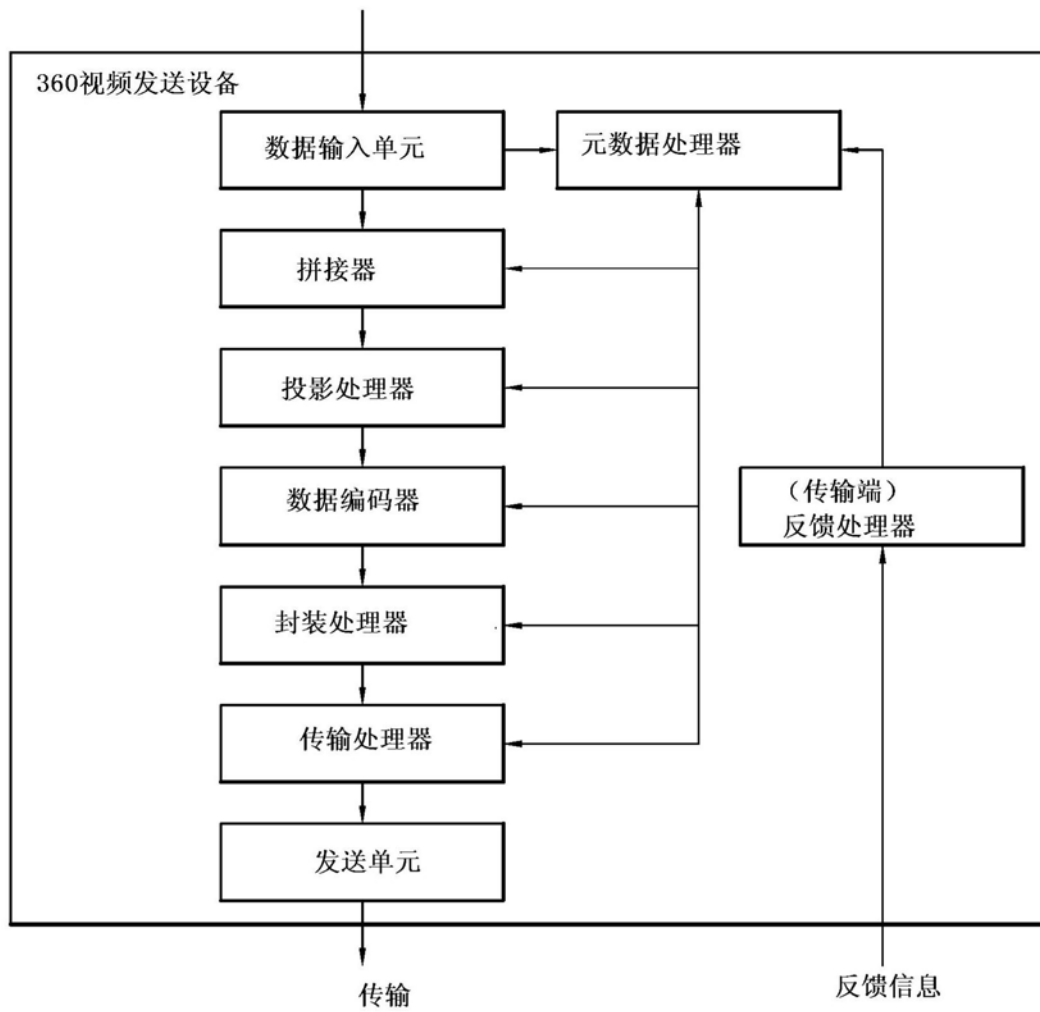


图5

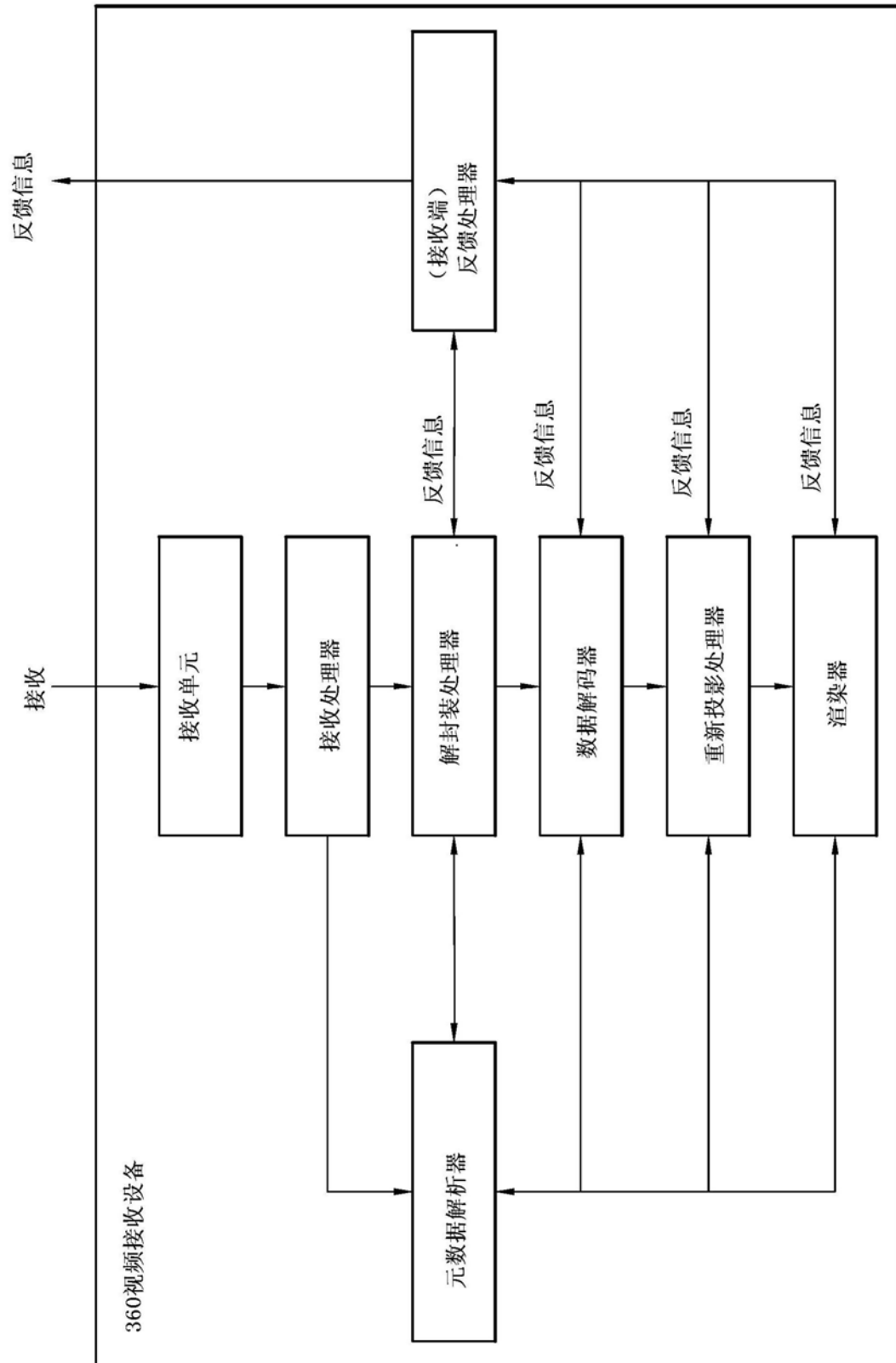


图6

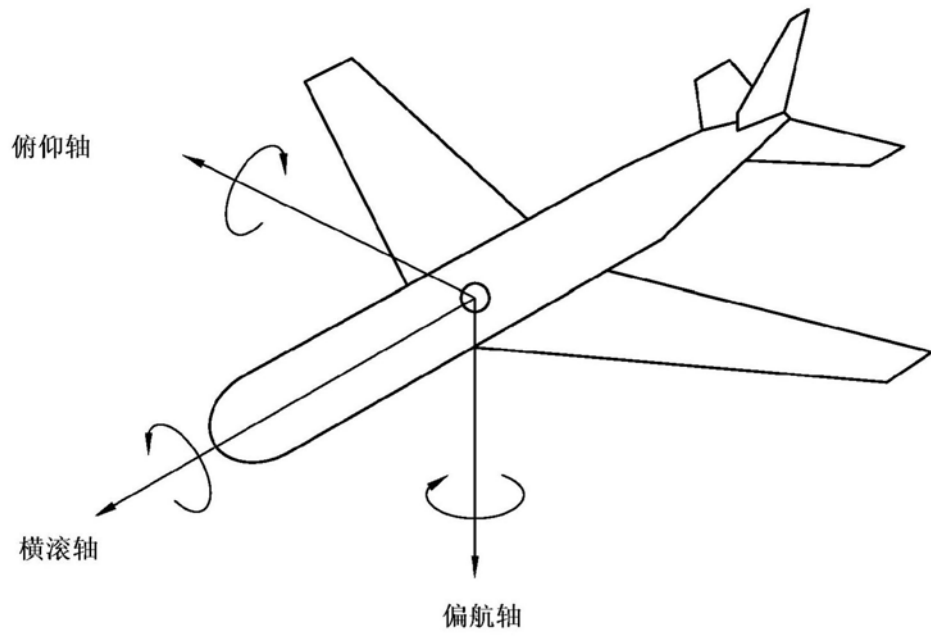


图7

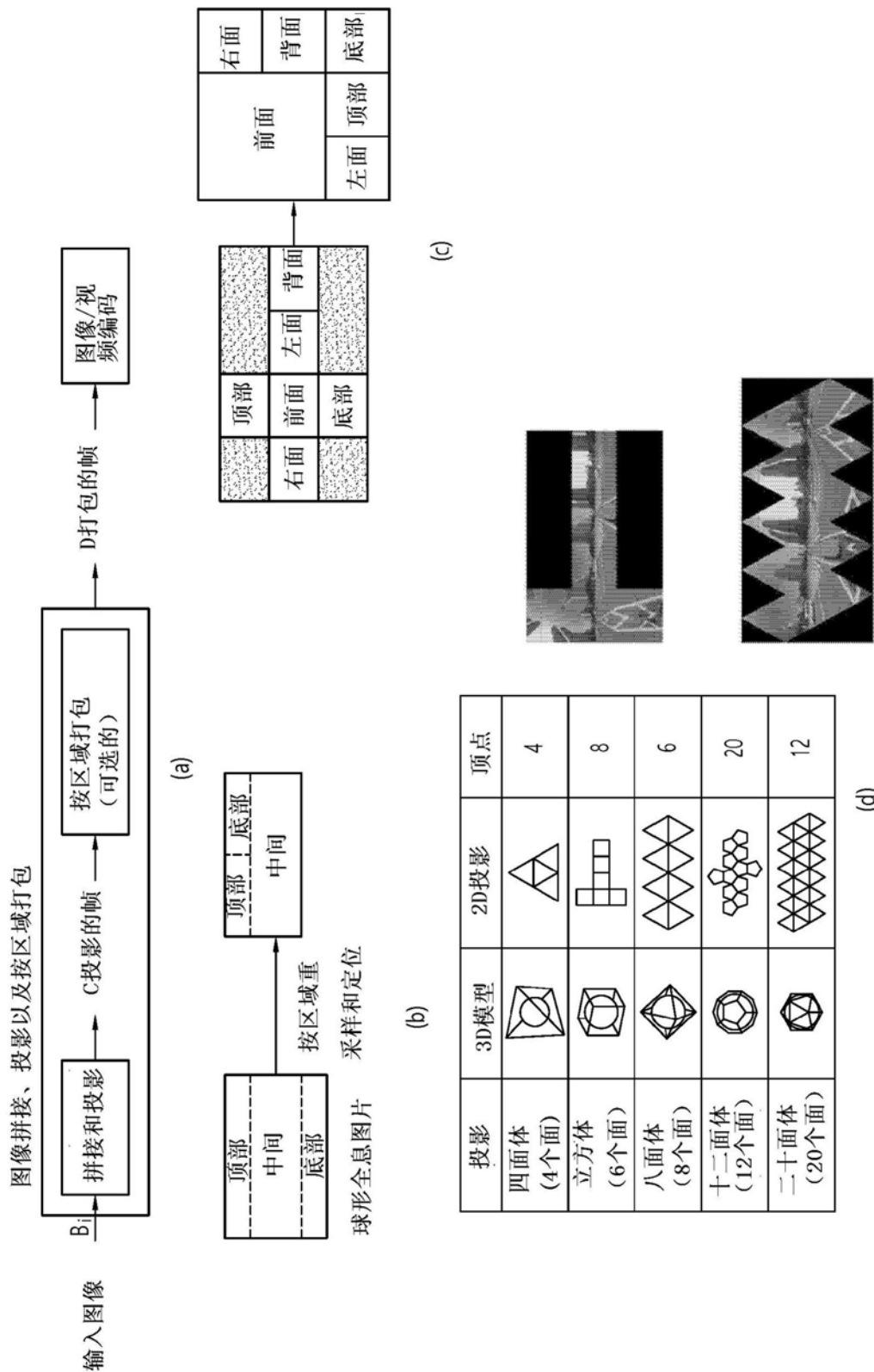


图8

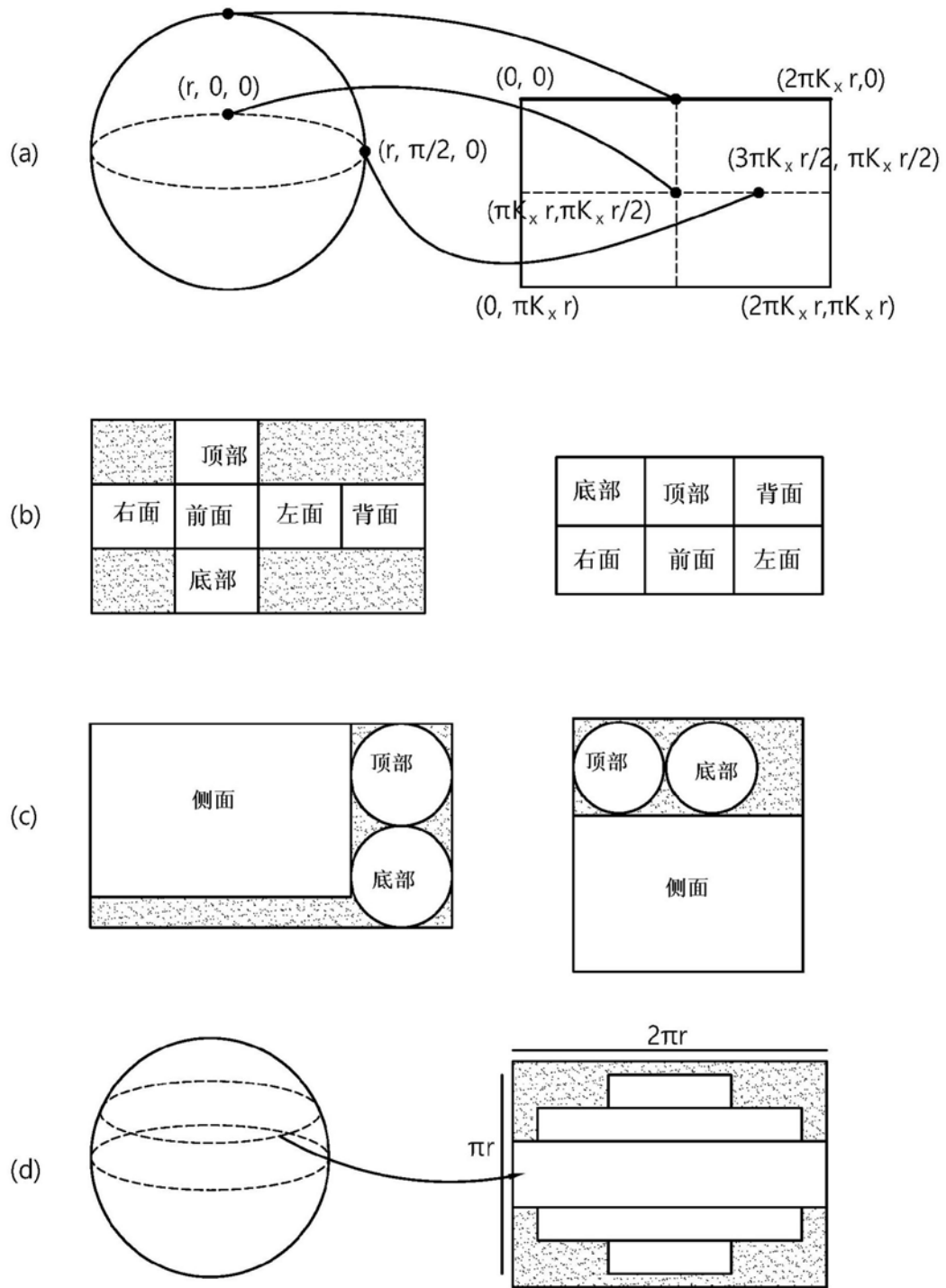


图9a

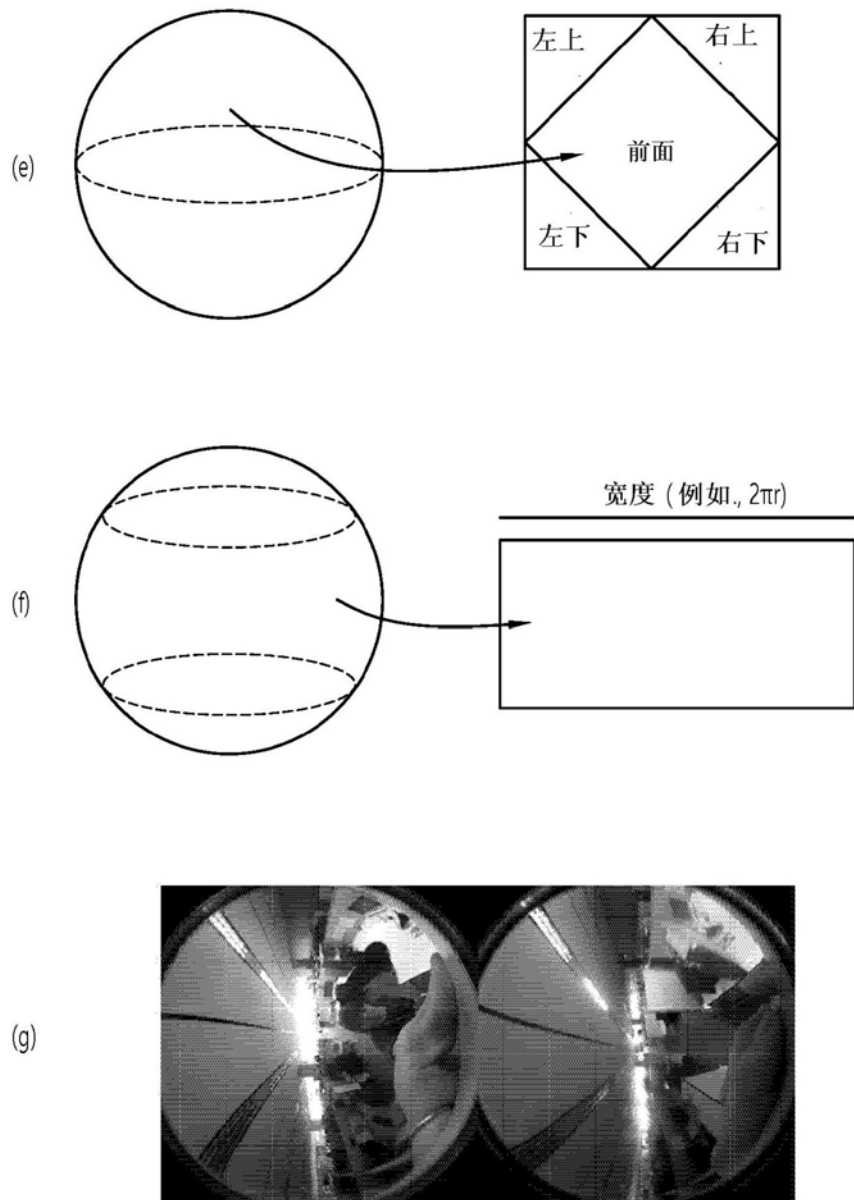


图9b

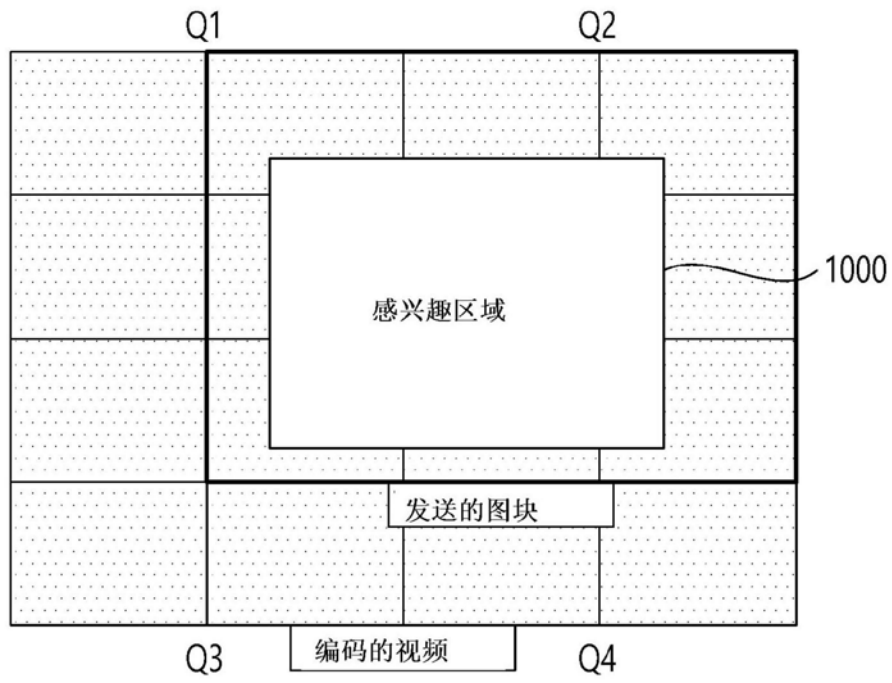


图10a

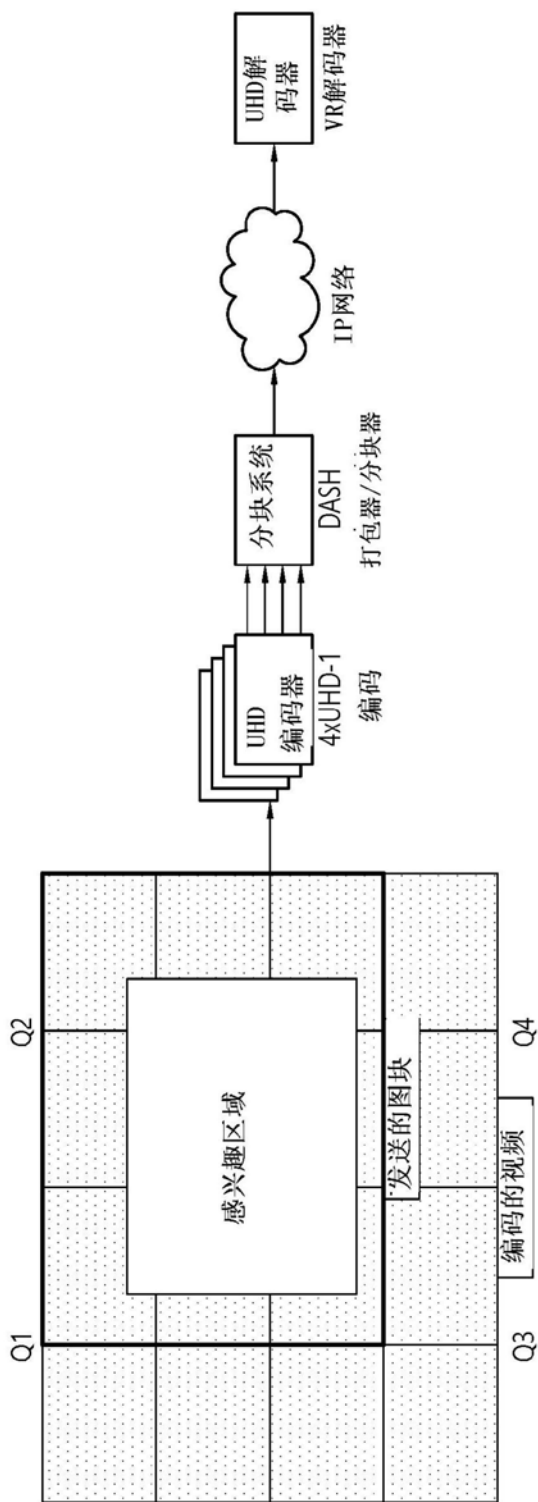


图10b

```

...
    <默认元数据>
    unsigned int(8)    vr_geometry;
    unsigned int(8)    projection_schme;

    <与立体有关的元数据>
    unsigned int(1)    is_stereoscopic;
    unsigned int(3)    stereo_mode;

    <与初始视野有关的元数据>
    signed int(8)      initial_view_yaw_degree;
    signed int(8)      initial_view_pitch_degree;
    signed int(8)      initial_view_roll_degree;

    <与ROI有关的元数据>
    unsigned int(1)    2d_roi_range_flag;
    unsigned int(1)    3d_roi_range_flag;
    if (2d_roi_range_flag==1) {
        unsigned int(16) min_top_left_x;
        unsigned int(16) max_top_left_x;
        unsigned int(16) min_top_left_y;
        unsigned int(16) max_top_left_y;
        unsigned int(16) min_width;
        unsigned int(16) max_width;
        unsigned int(16) min_height;
        unsigned int(16) max_height;
        unsigned int(16) min_x;
        unsigned int(16) max_x;
        unsigned int(16) min_y;
        unsigned int(16) max_y;
    }
    if (3d_roi_range_flag==1) {
        unsigned int(16) min_yaw;
        unsigned int(16) max_yaw;
        unsigned int(16) min_pitch;
        unsigned int(16) max_pitch;
        unsigned int(16) min_roll;
        unsigned int(16) max_roll;
        unsigned int(16) min_field_of_view;
        unsigned int(16) max_field_of_view;
    }

    <与视场有关的元数据>
    unsigned int(1)    content_fov_flag;
    if (content_fov_flag==1) {
        unsigned int(16)    content_fov;
    }

    <与裁剪区域有关的元数据>
    unsigned int(1)    is_copped_region;
    if (content_fov_flag==1) {
        unsigned int(16)    cr_region_left_top_x;
        unsigned int(16)    cr_region_left_top_y;
        unsigned int(16)    cr_region_width;
        unsigned int(16)    cr_region_height;
    }
    ...

```

图11

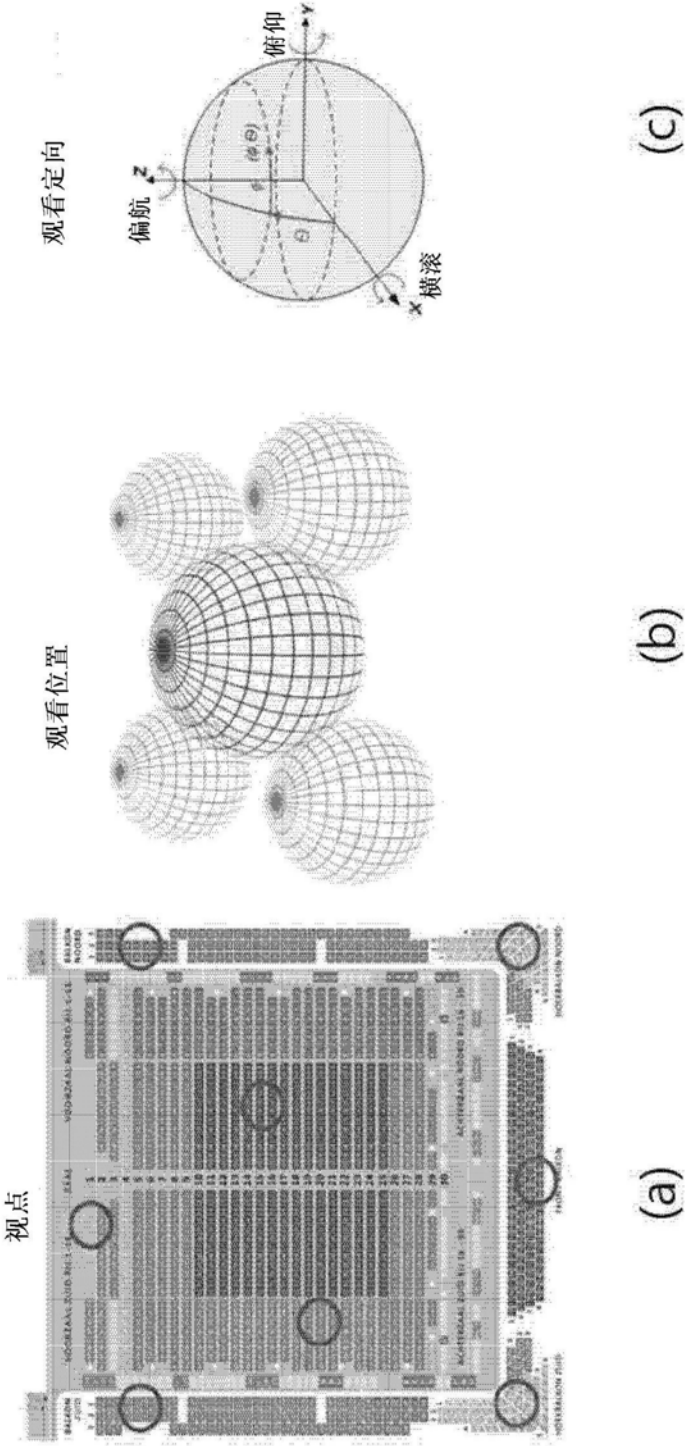


图12

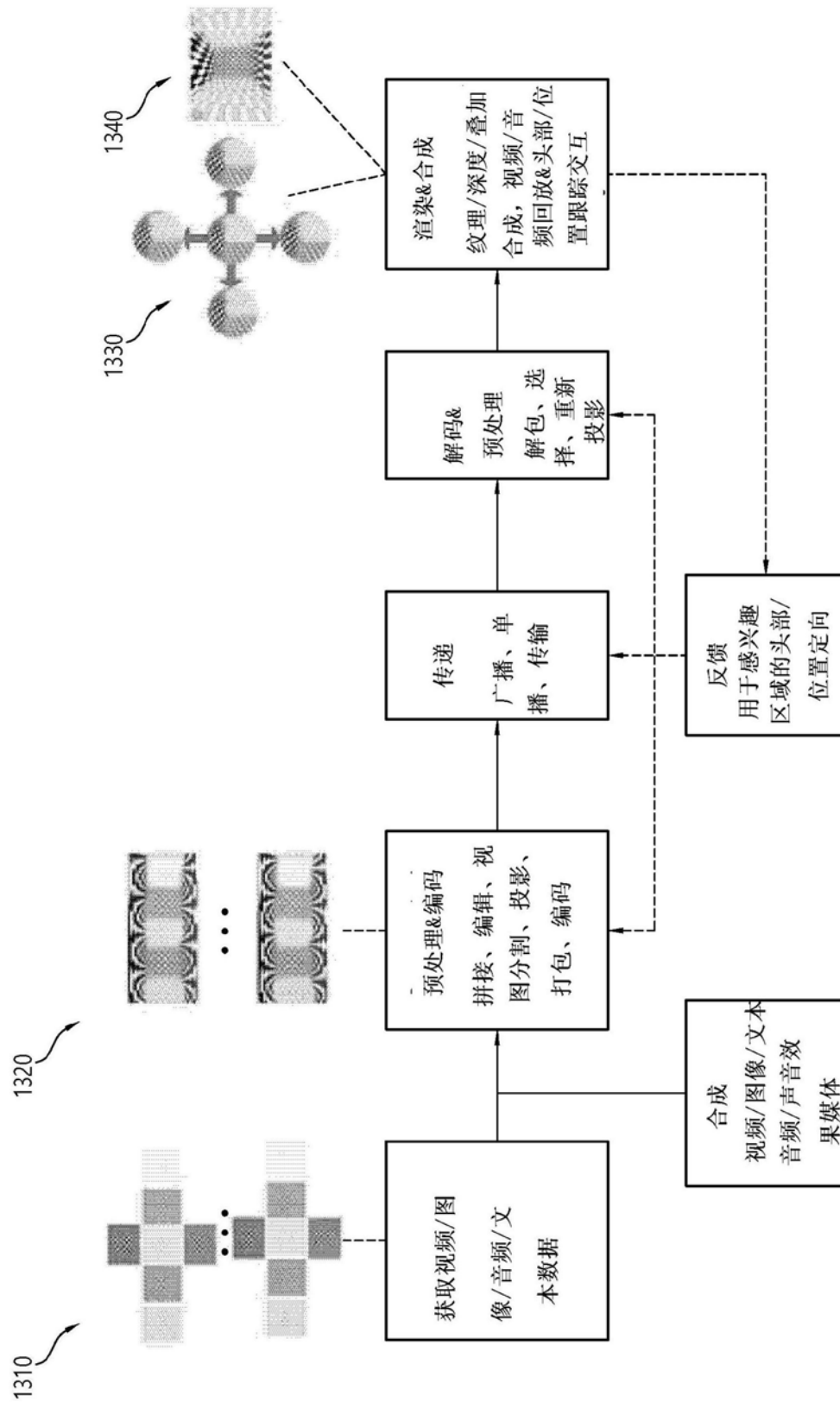


图13

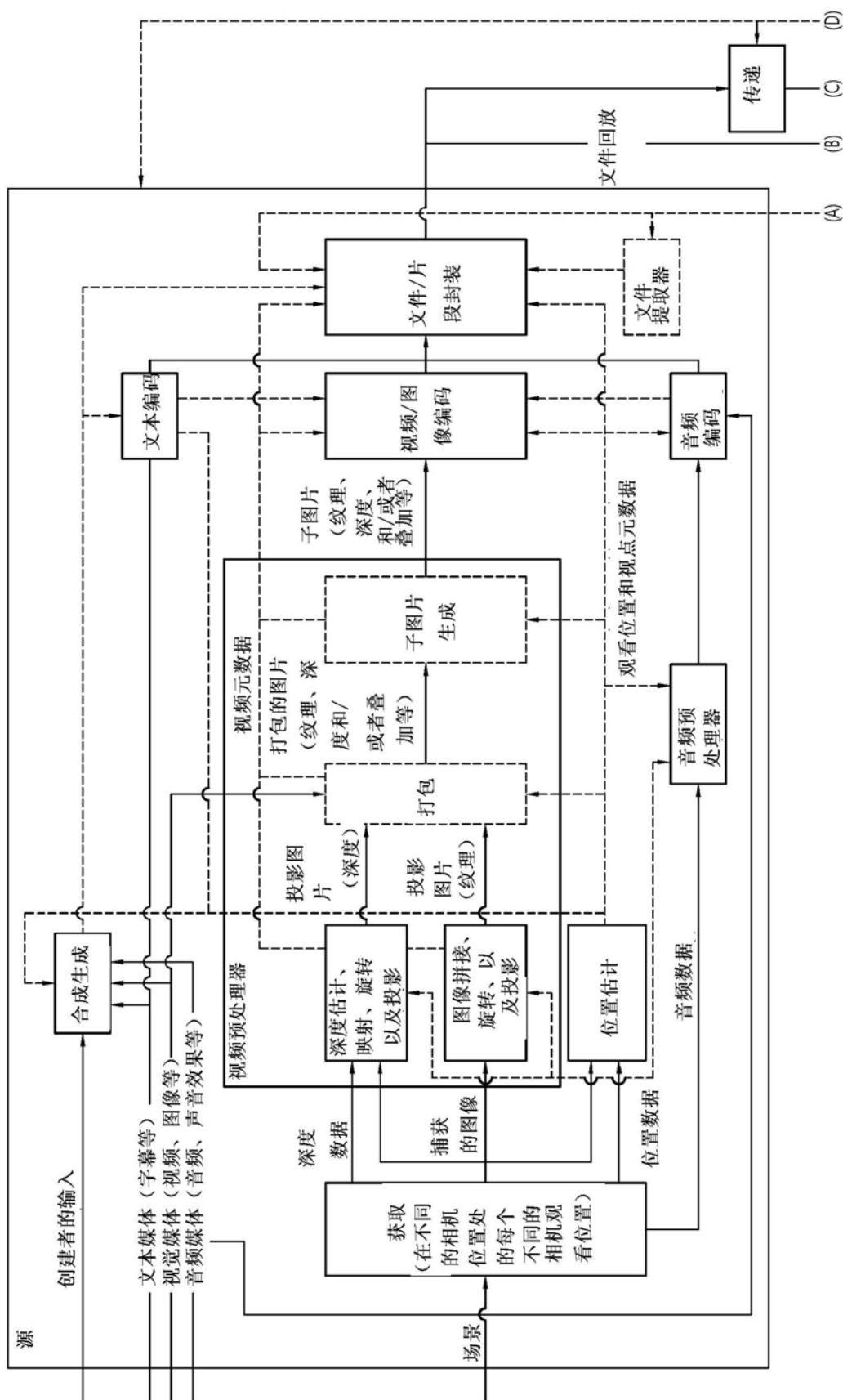


图14a

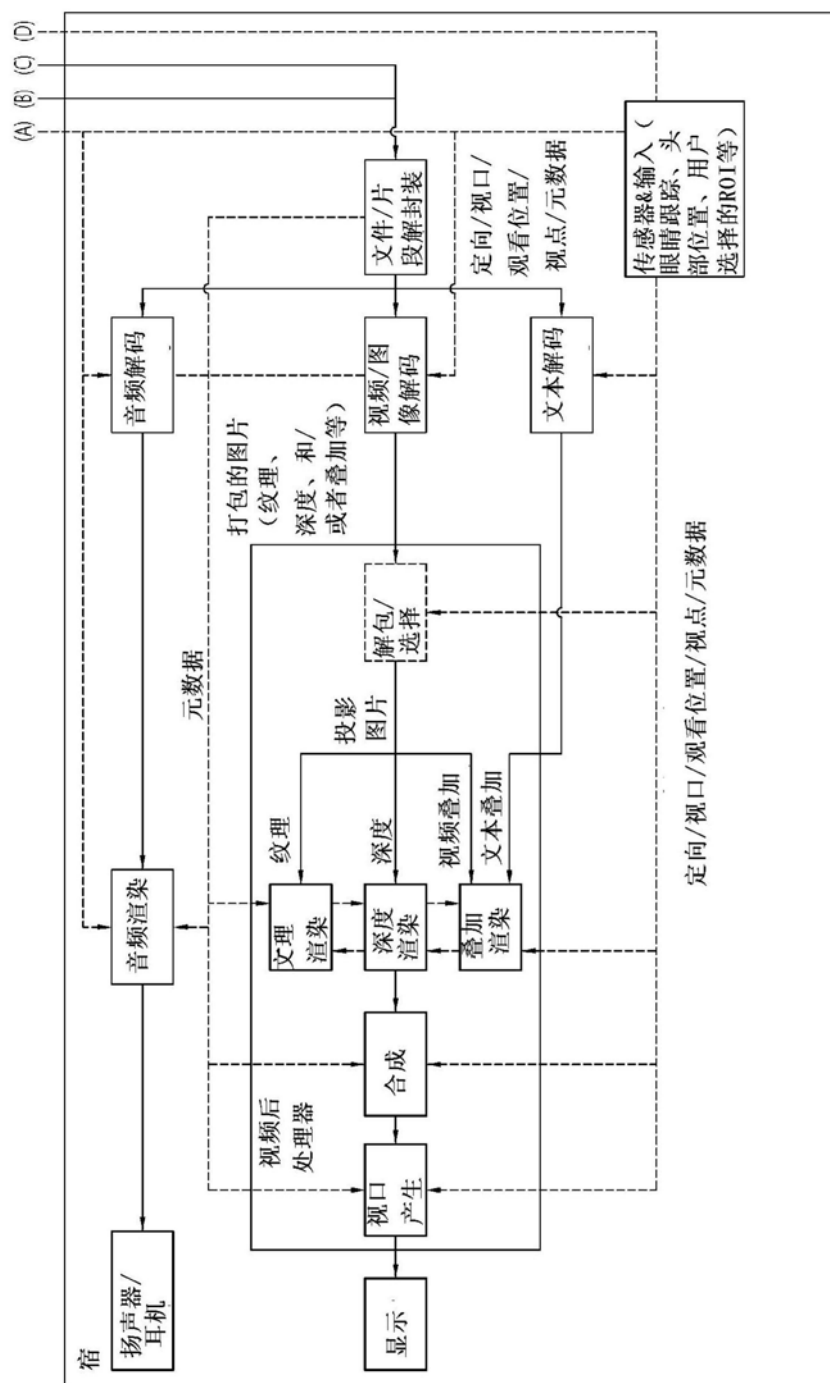


图14b

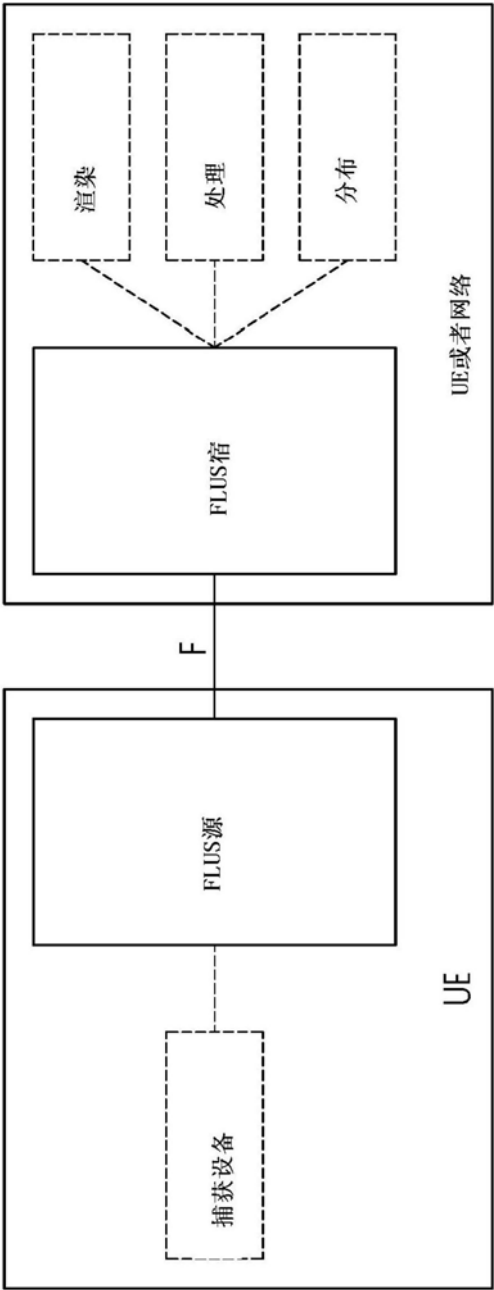


图15

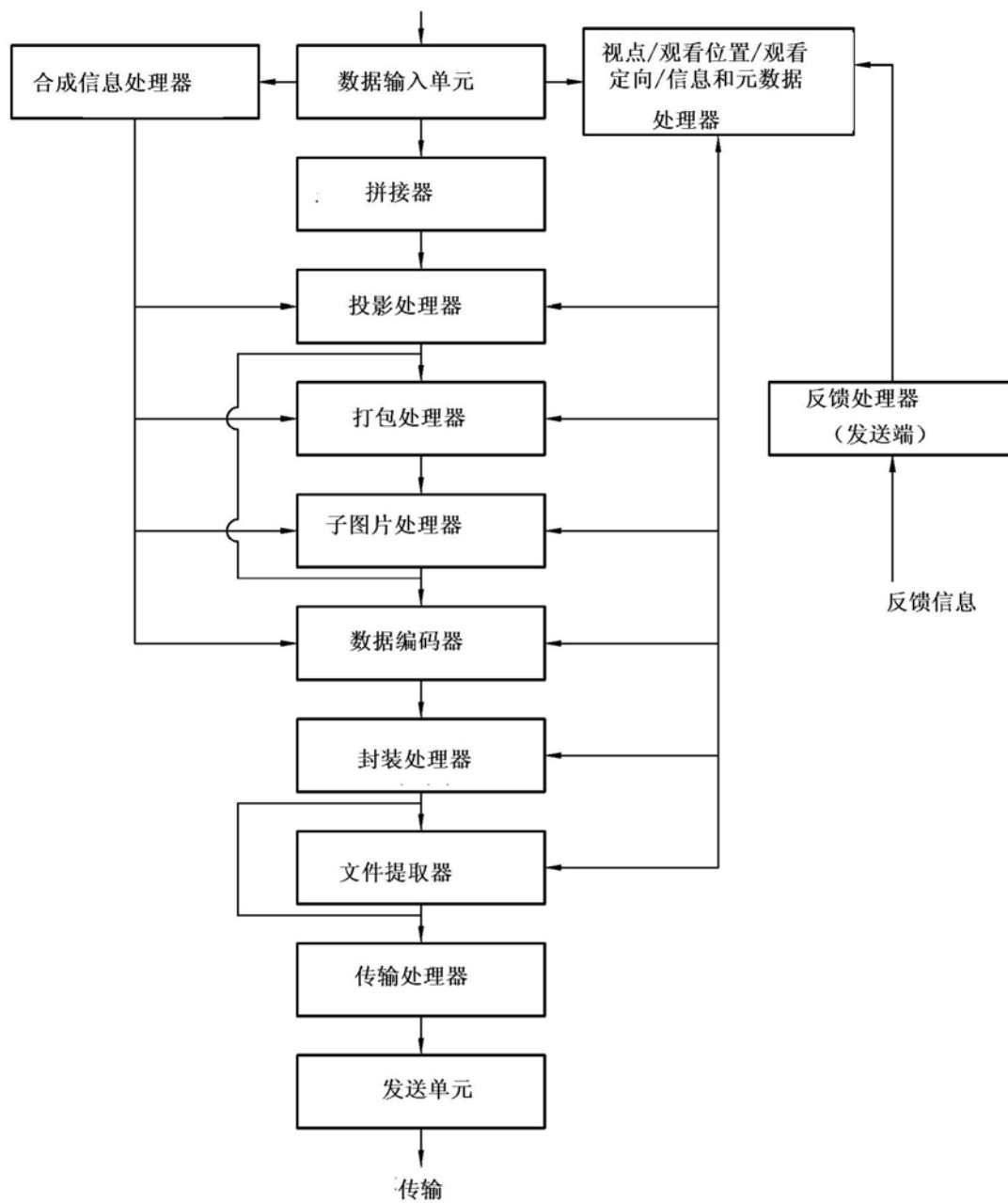


图16

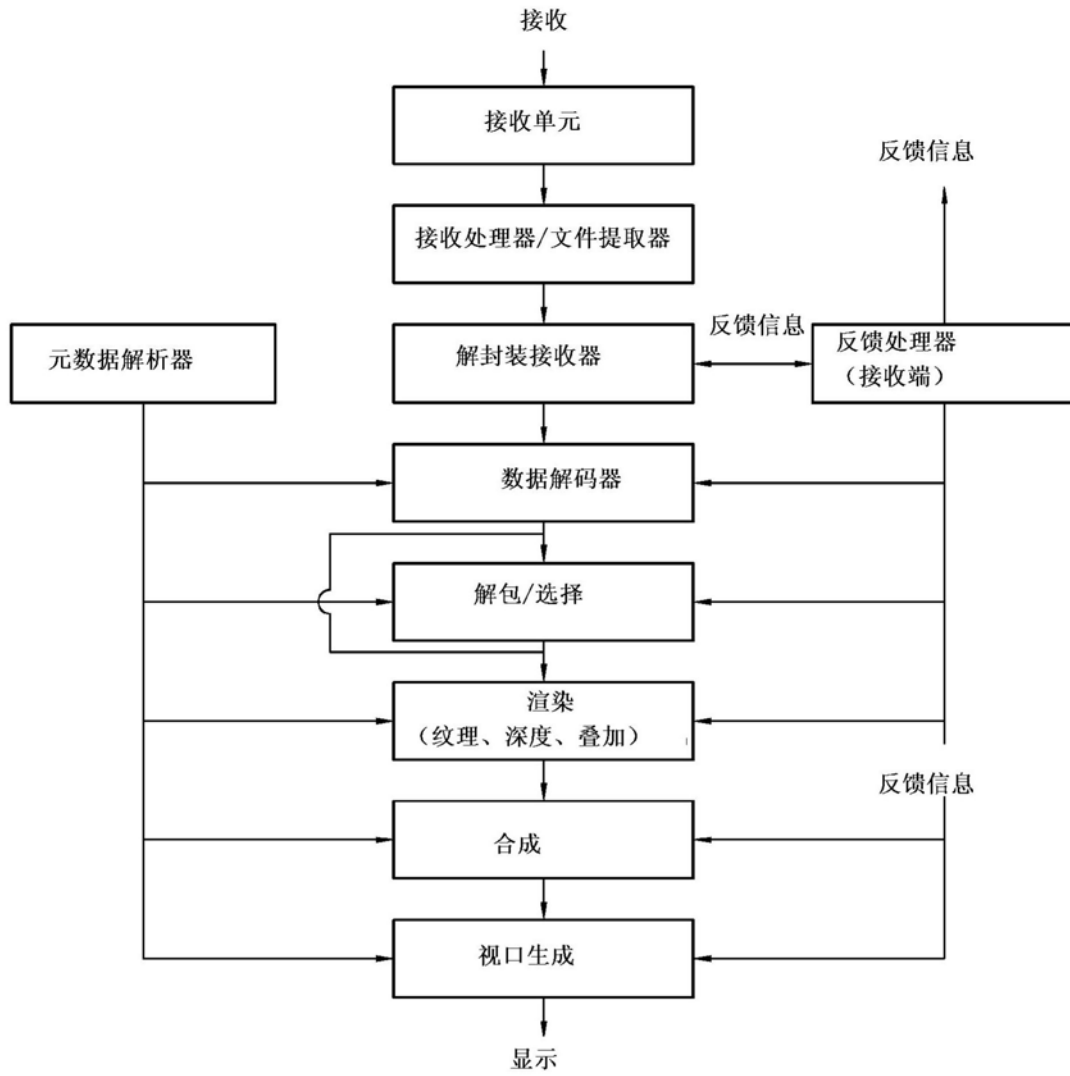


图17

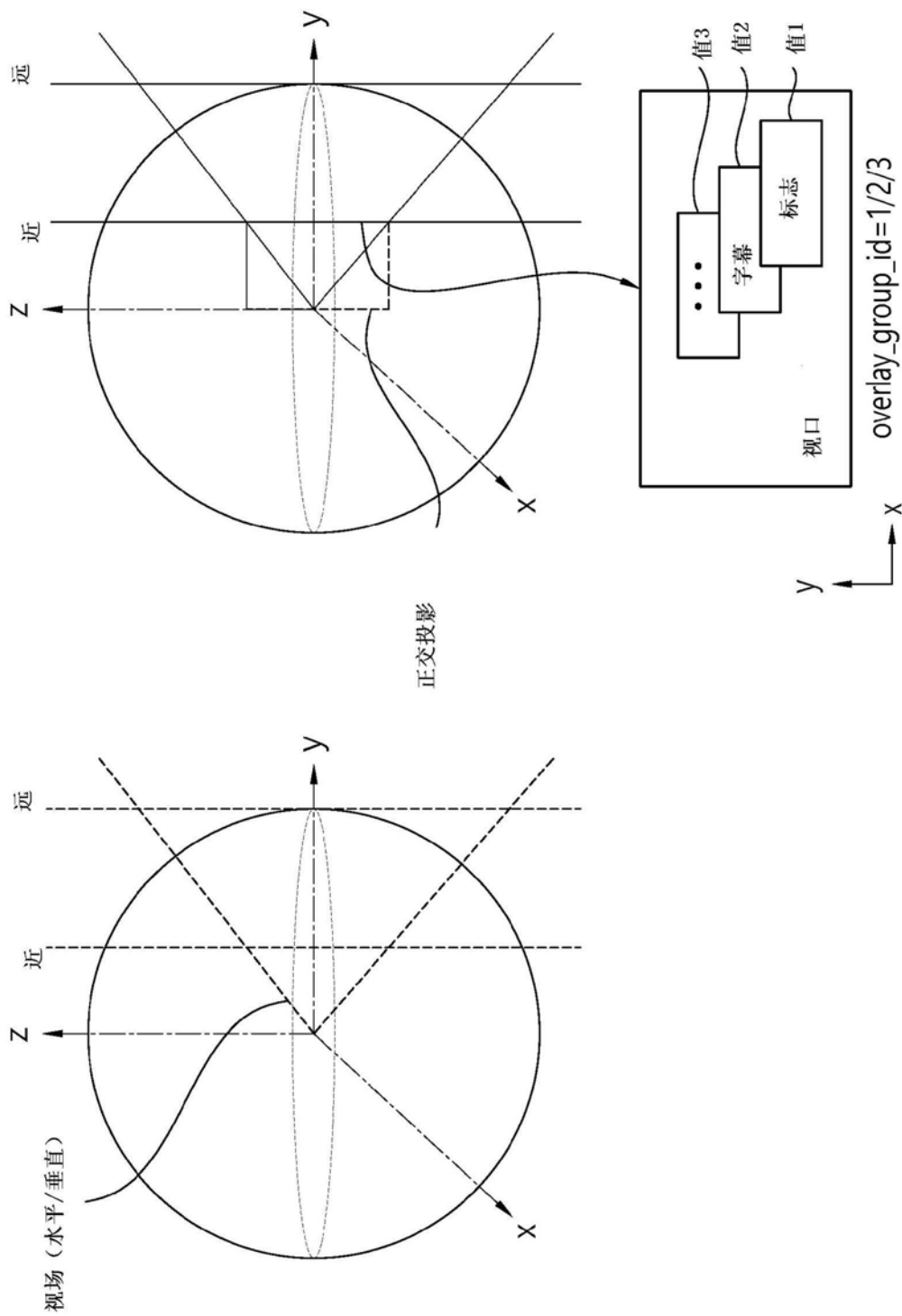
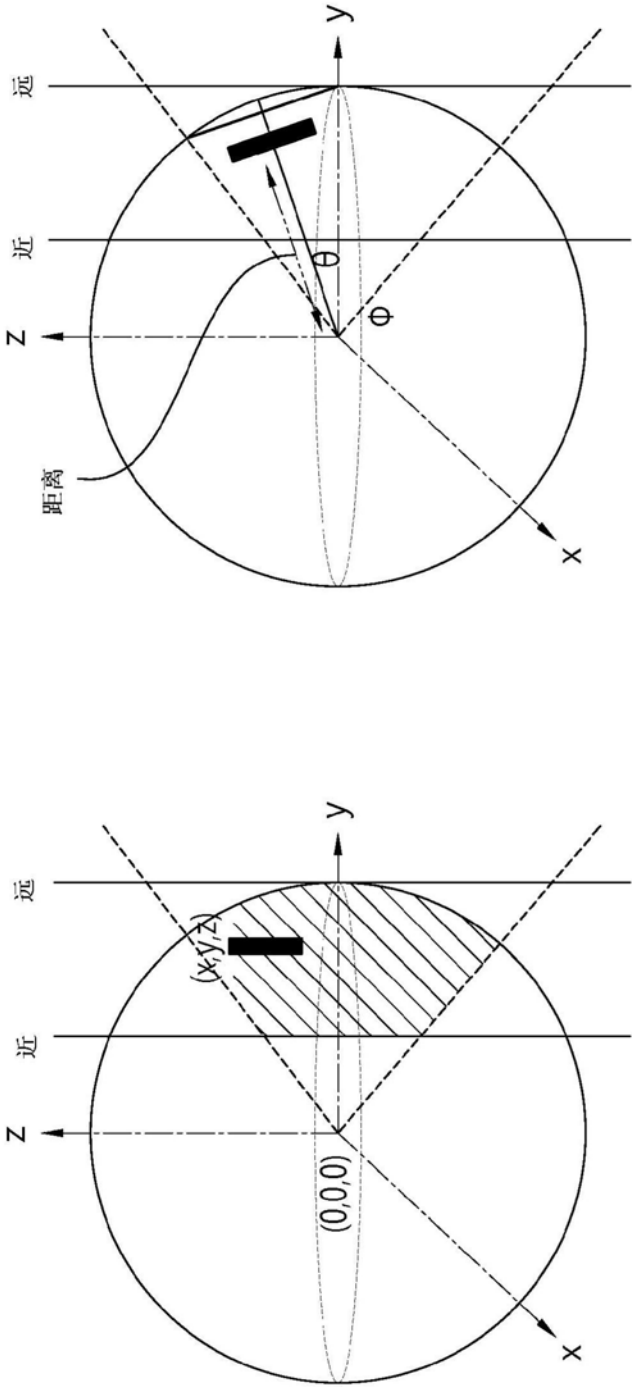
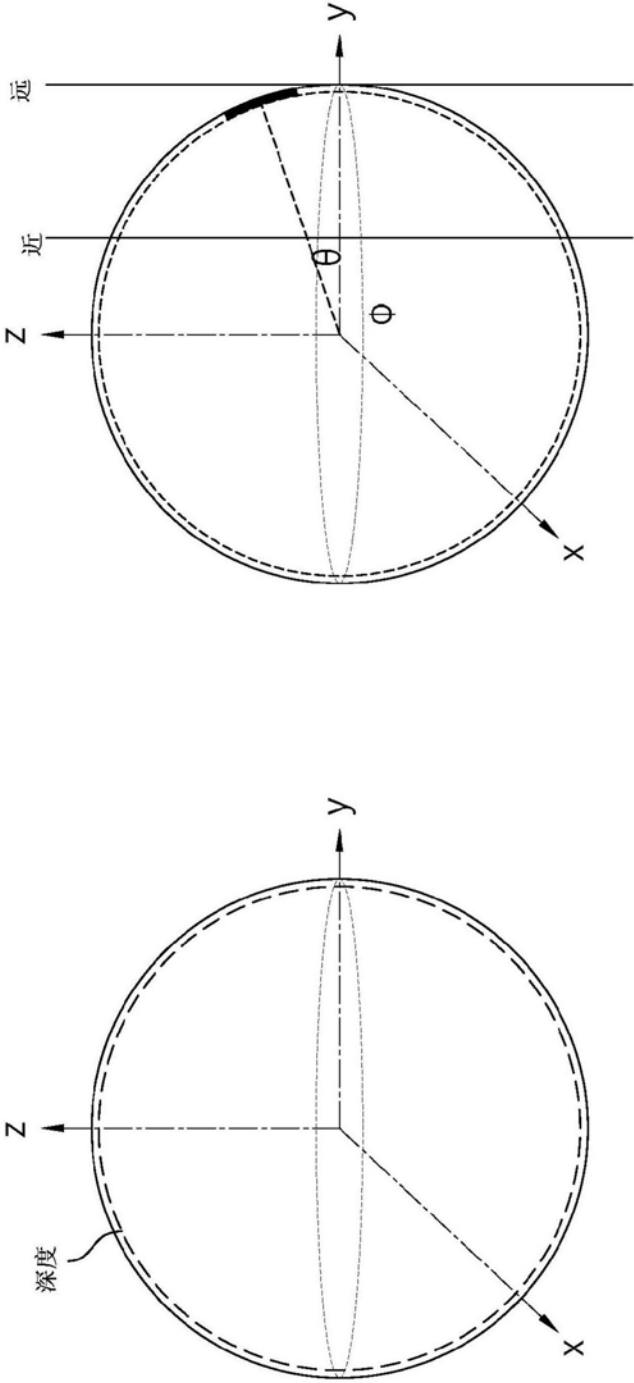


图18



overlay_group_id=4

图19



overlay_group_id=5

图20

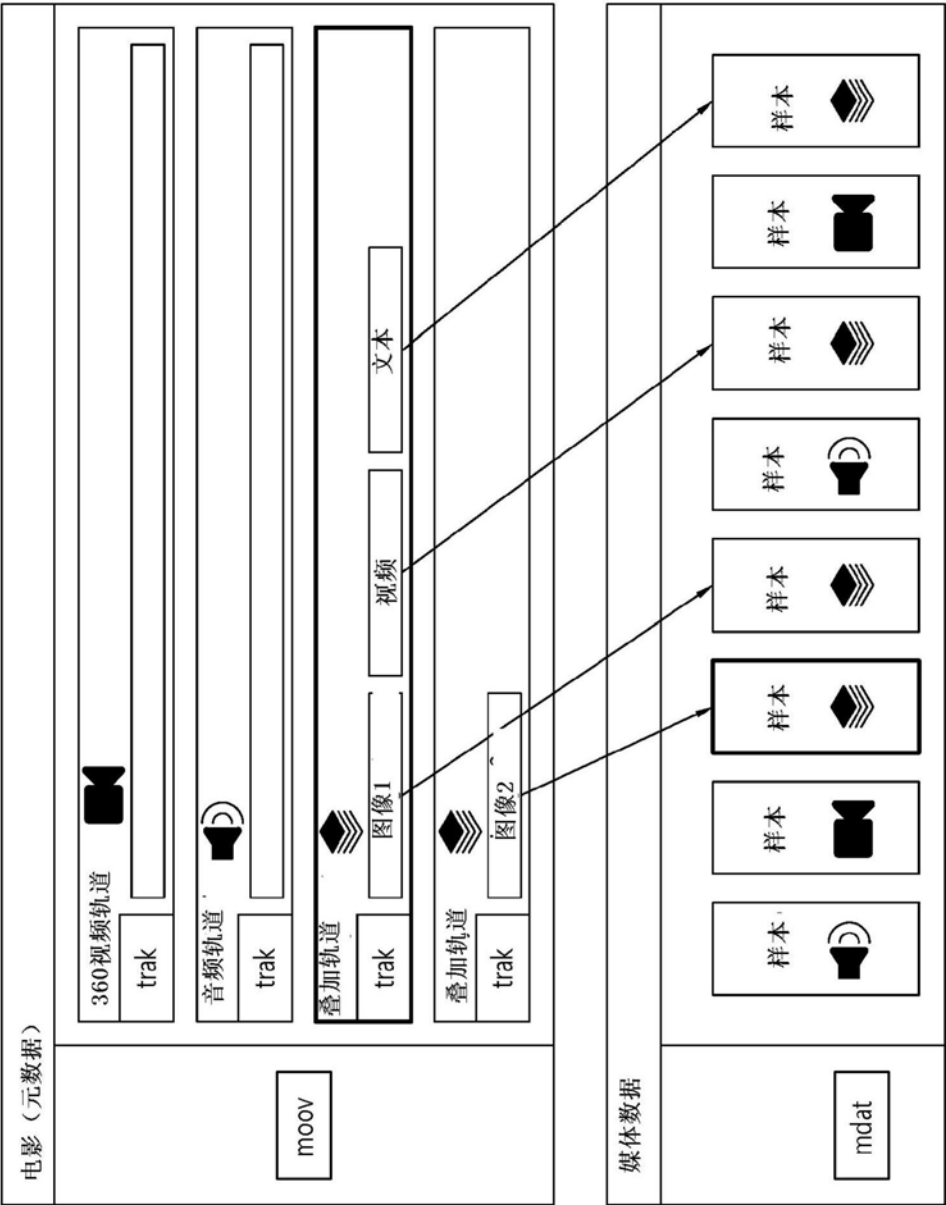


图21

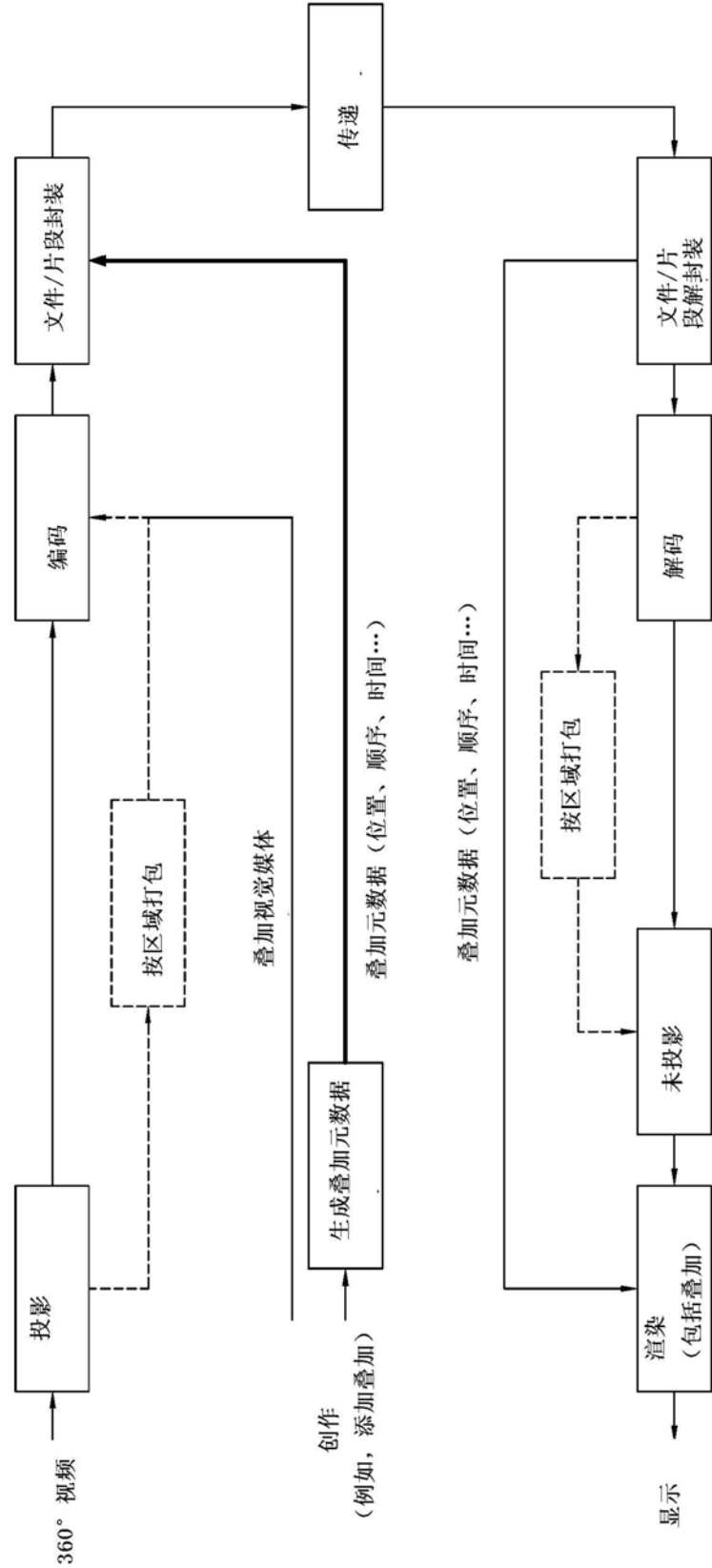


图22

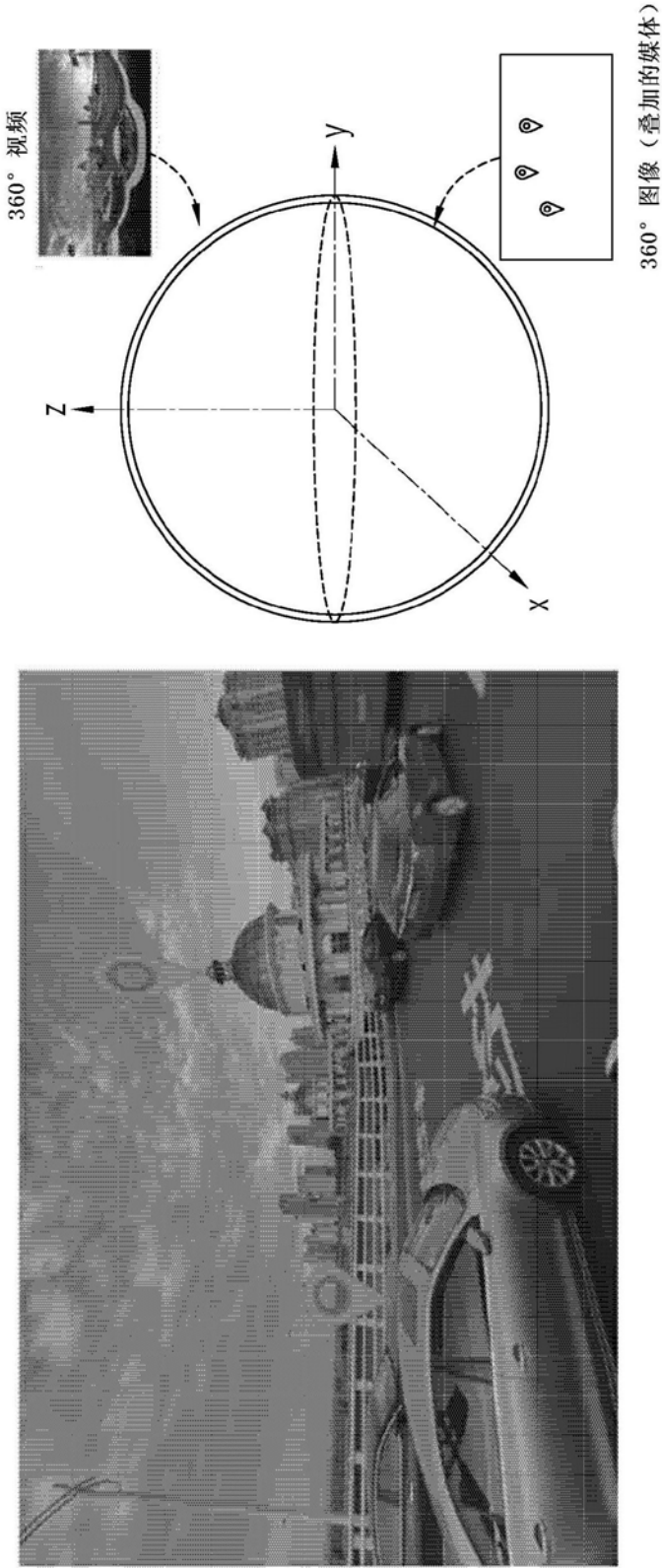


图23

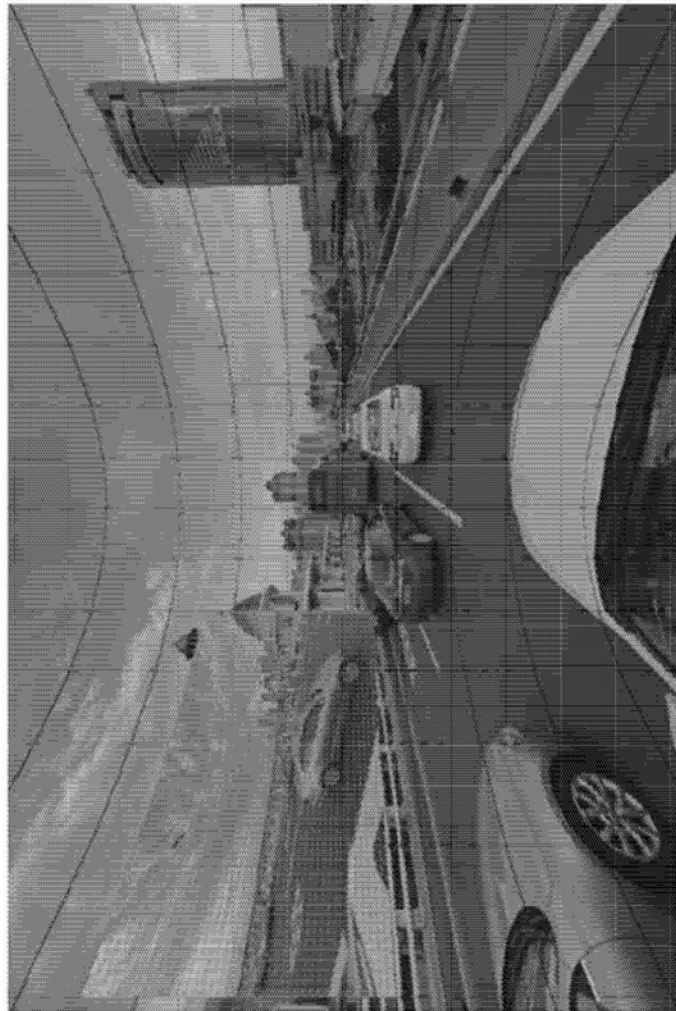
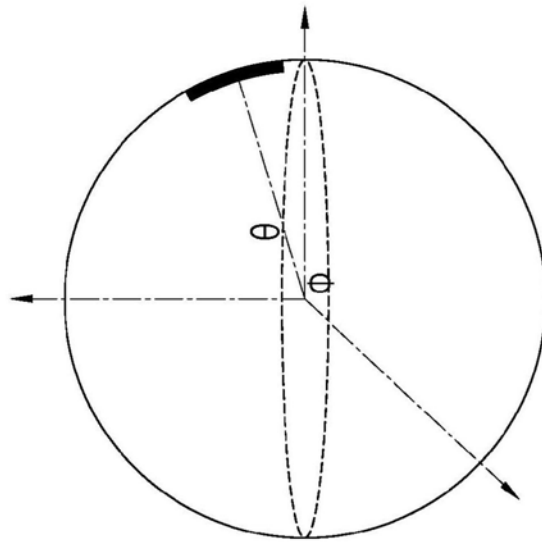


图24

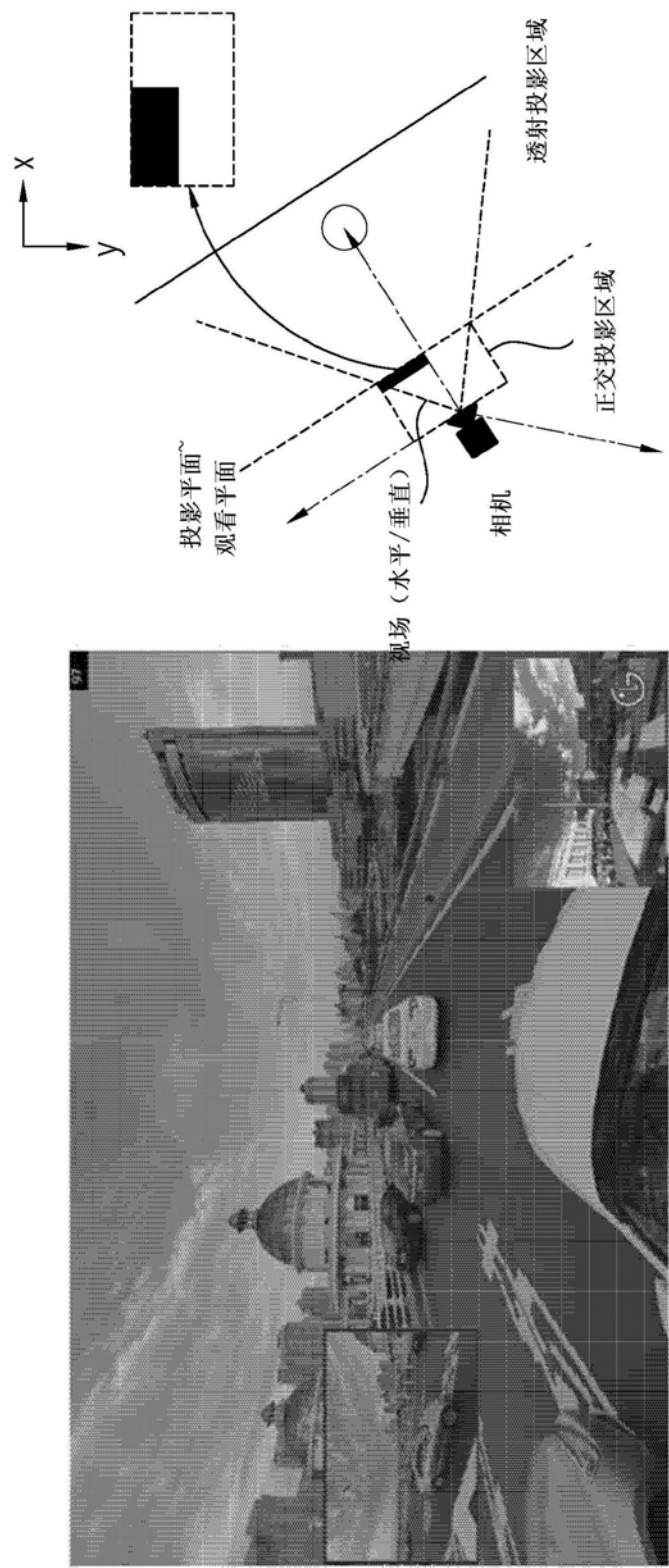


图25

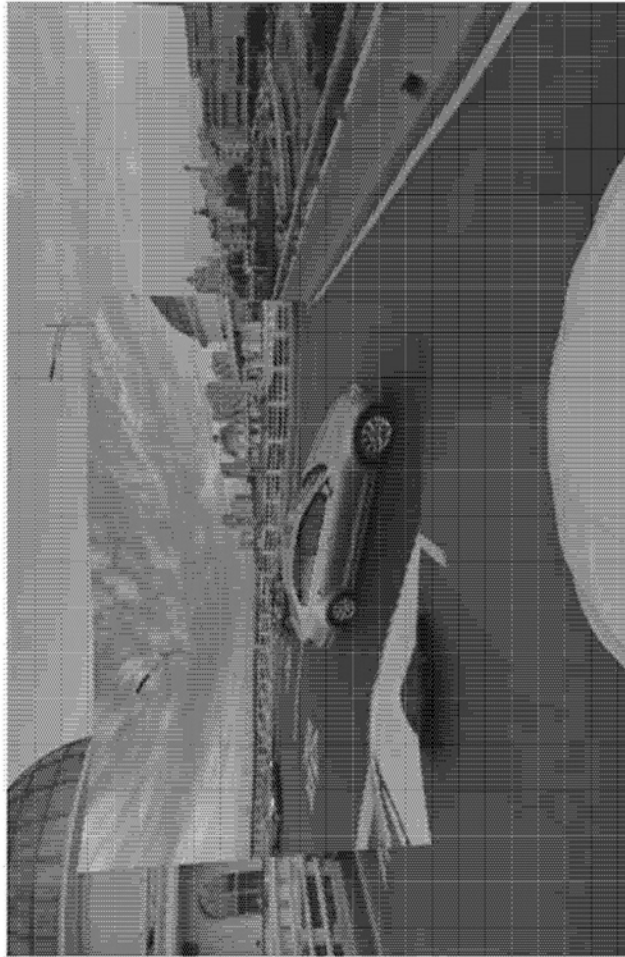
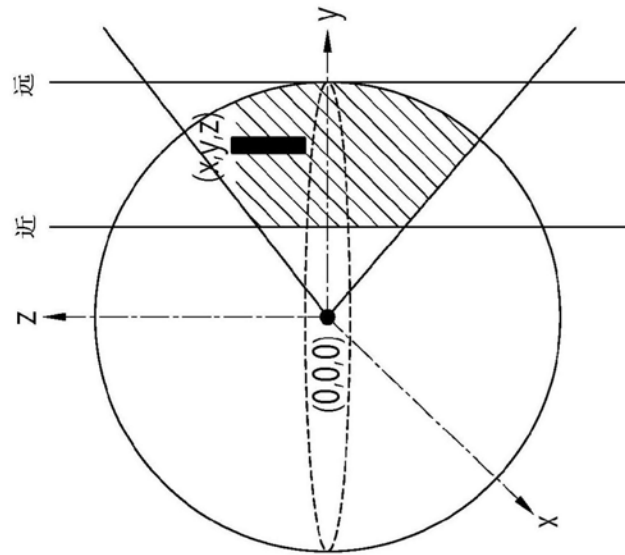


图26

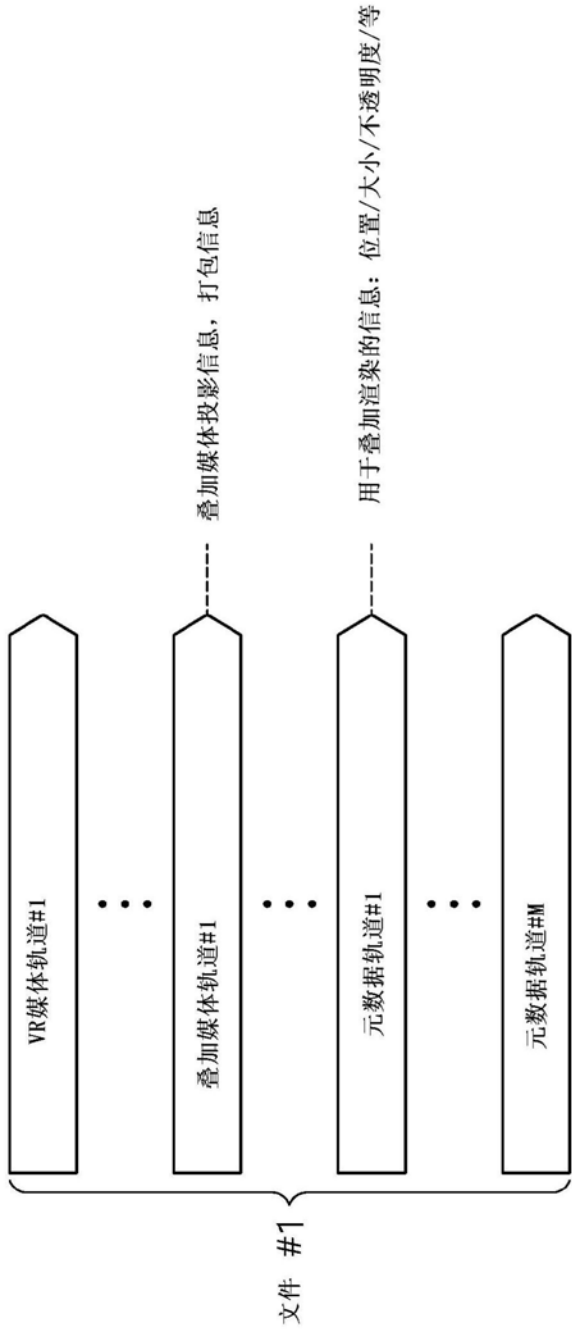


图27



图28

当定位在球体上时

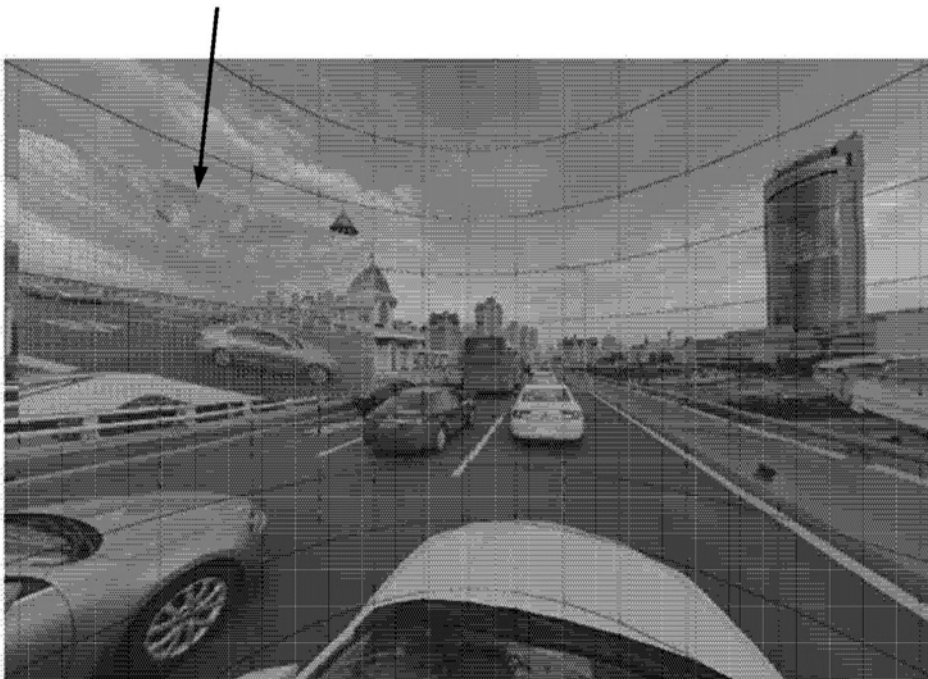


图29



当定位在3D空间上时

图30

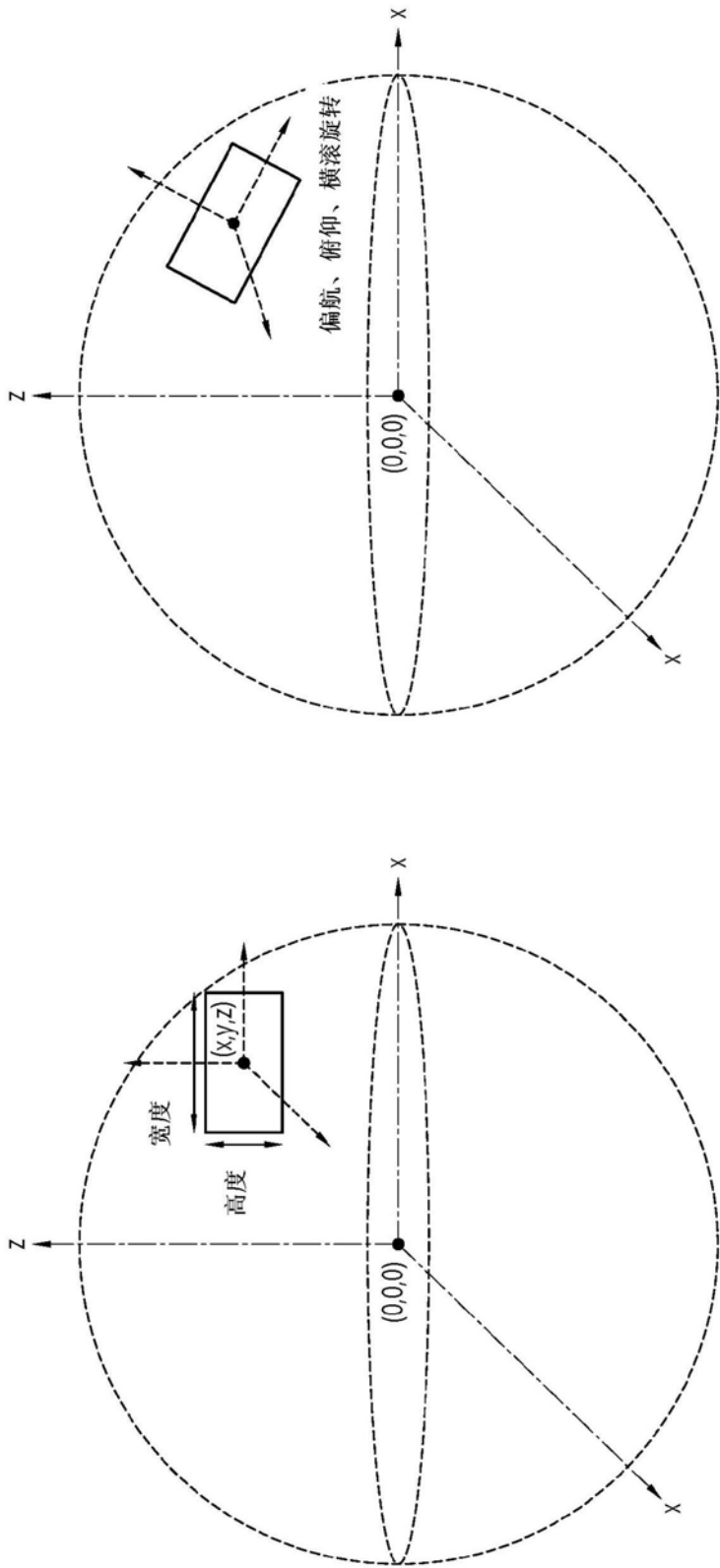


图31

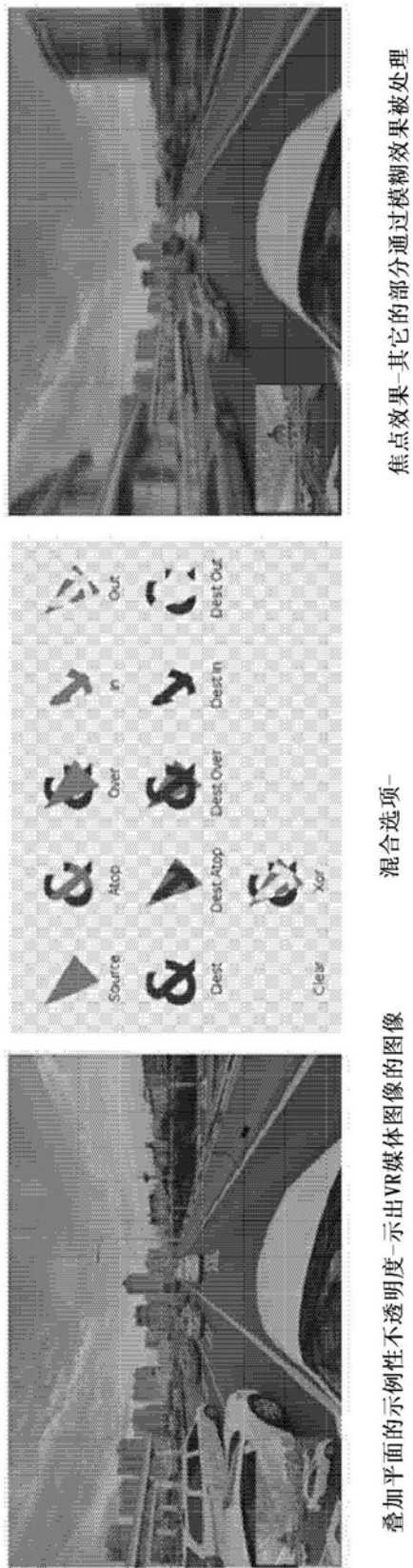


图32

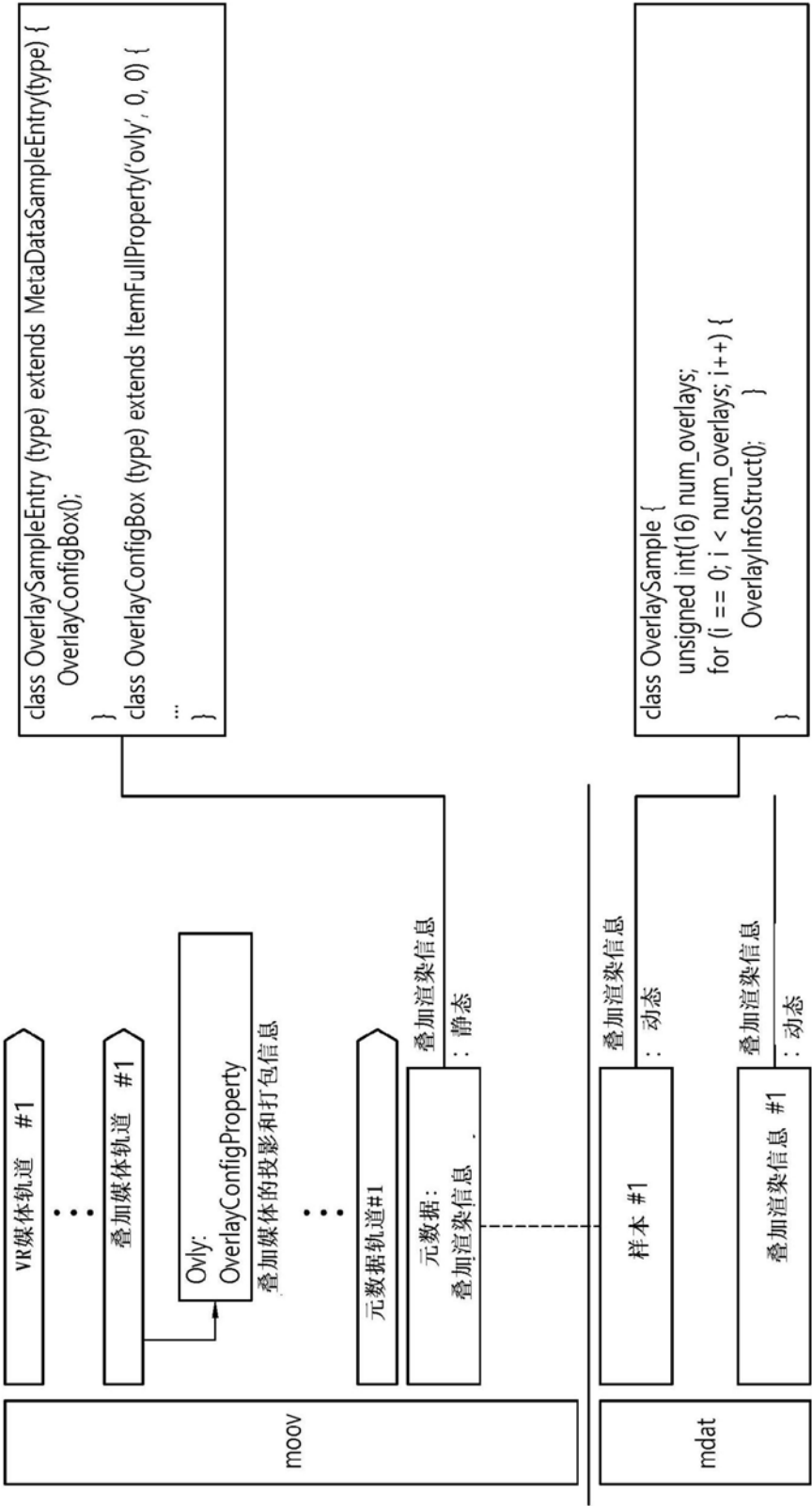


图33

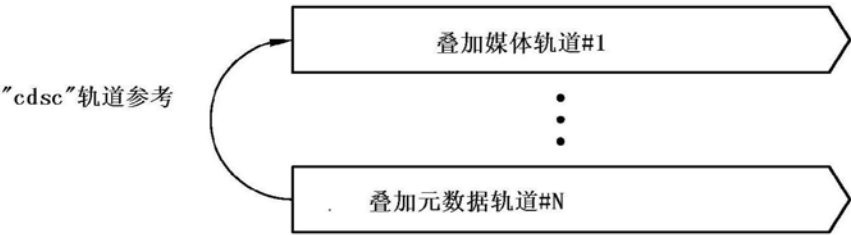


图34

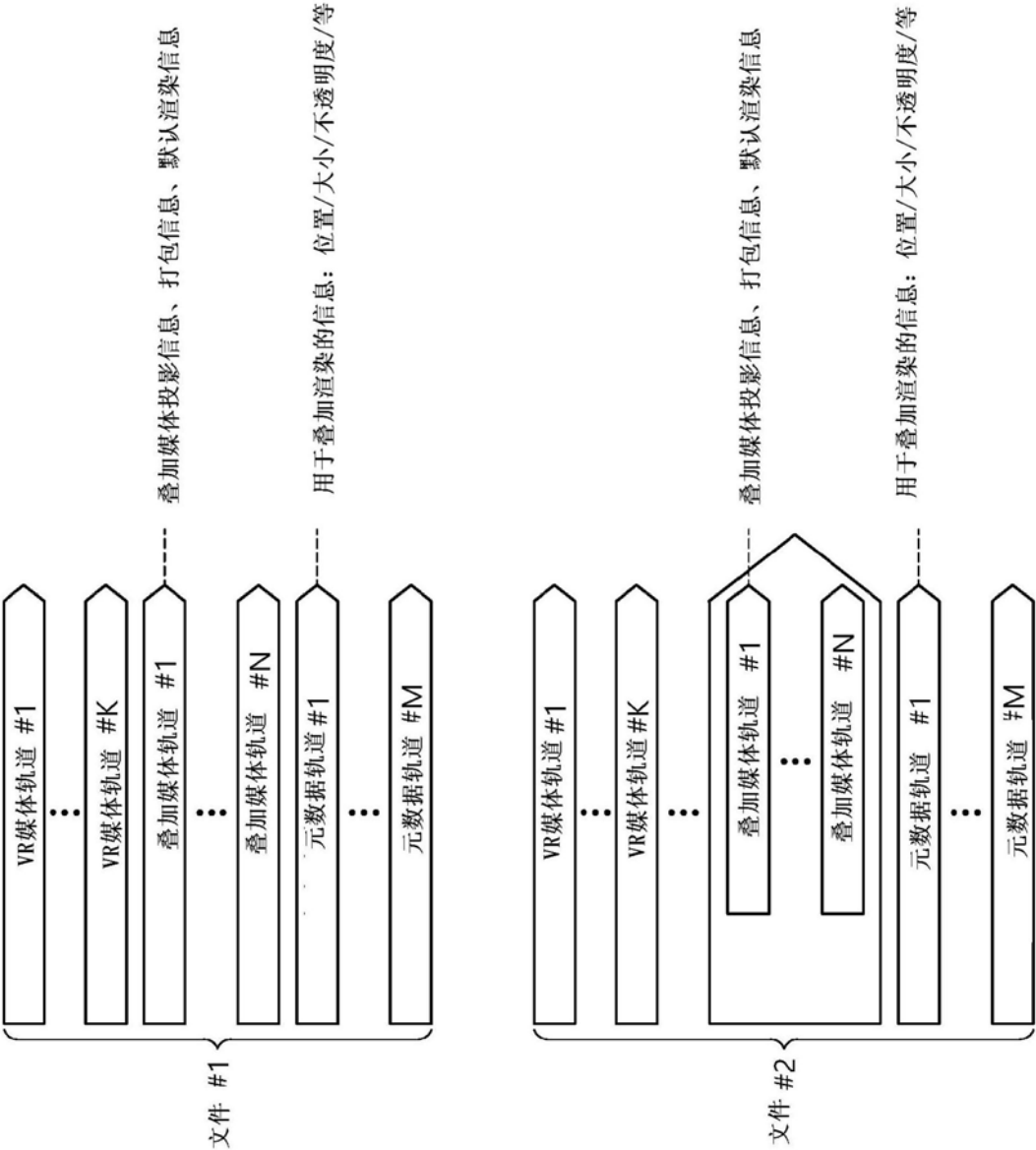


图35

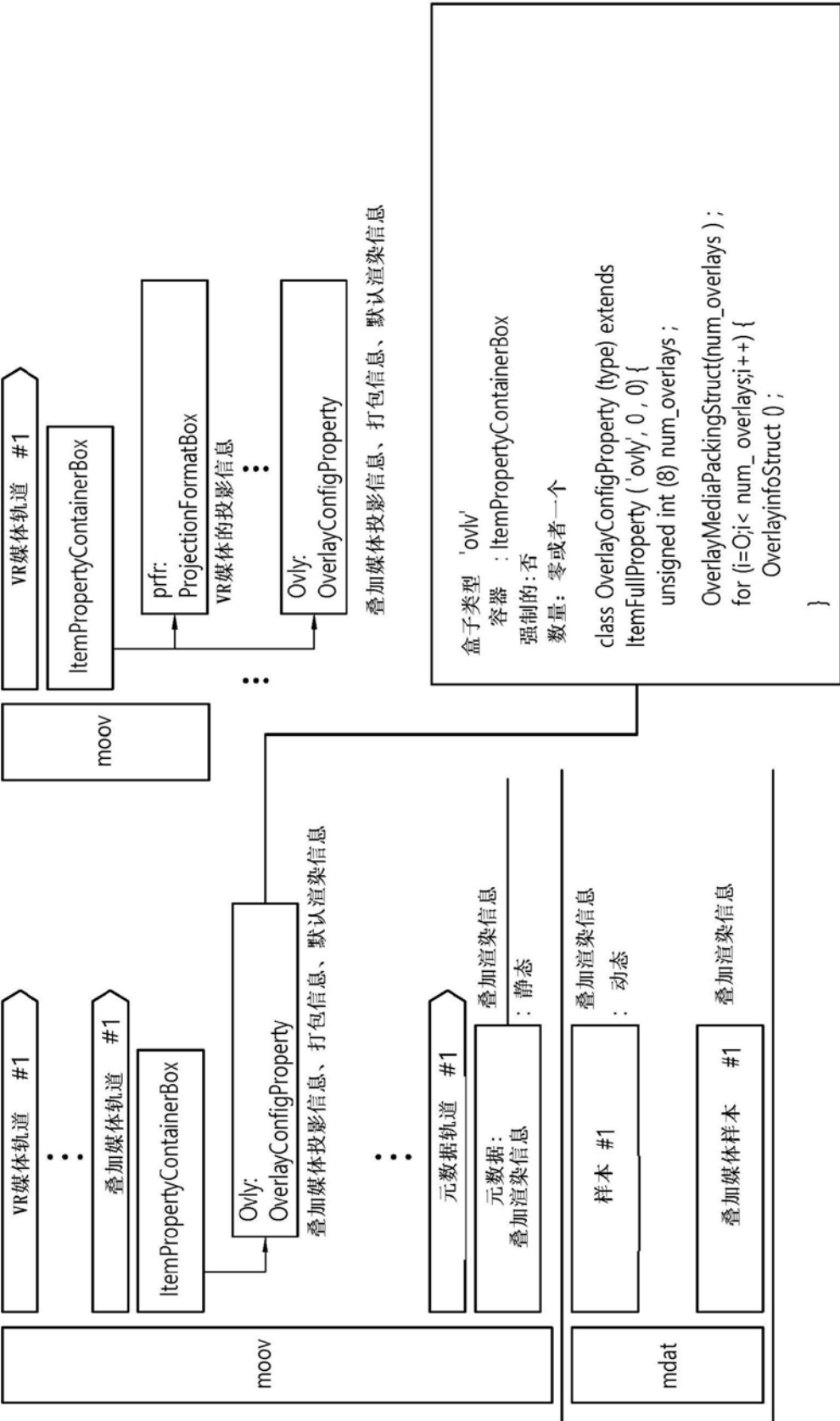


图36

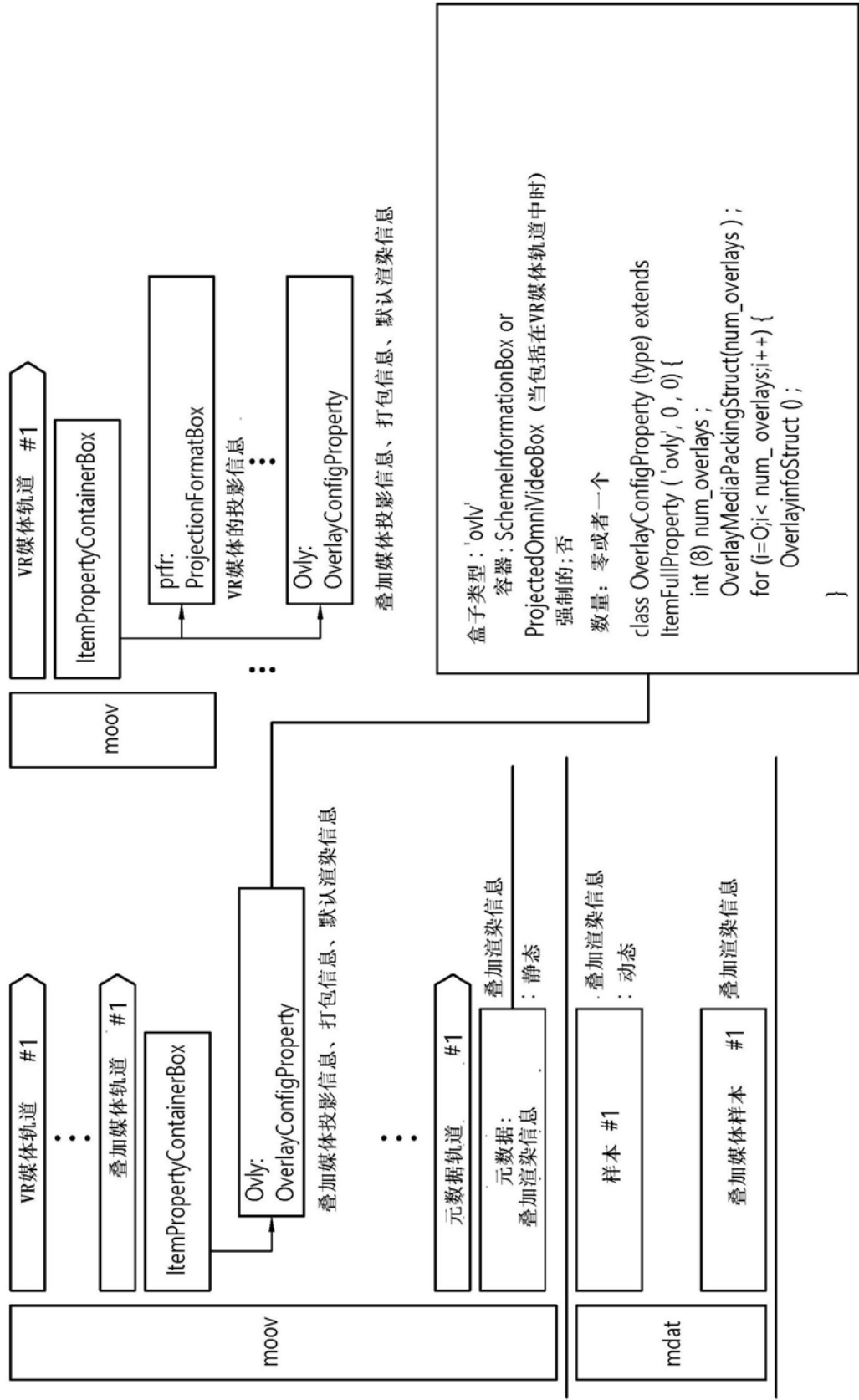


图37

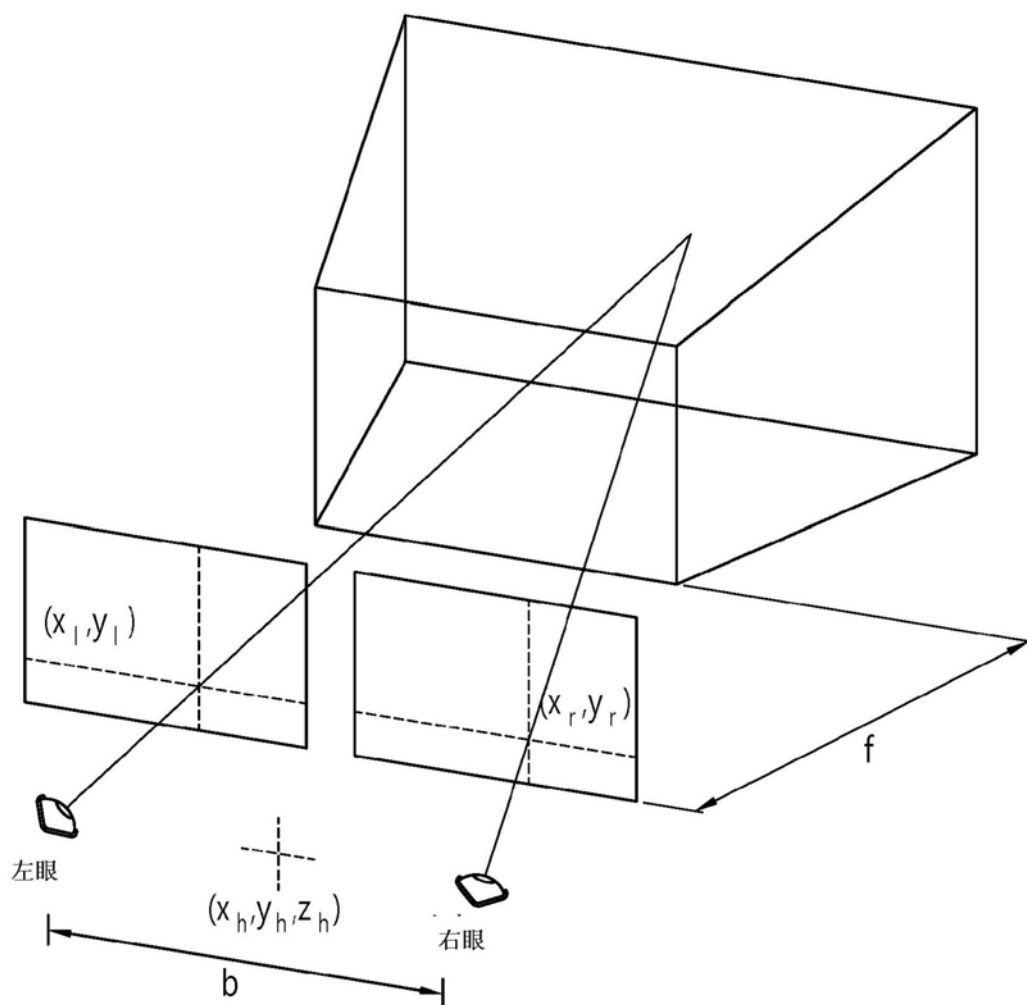


图38

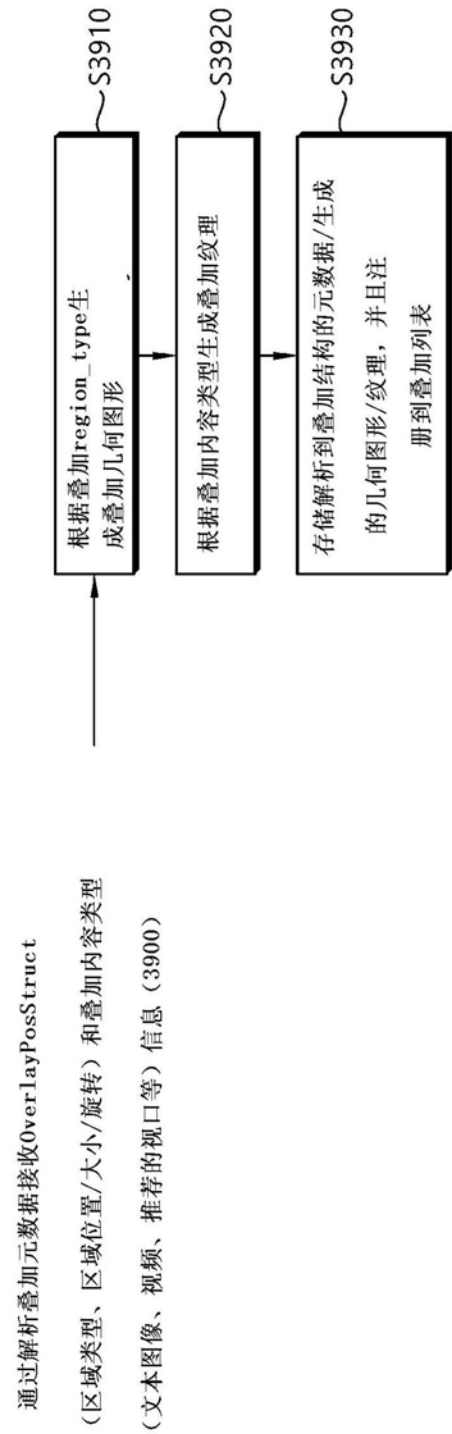


图39

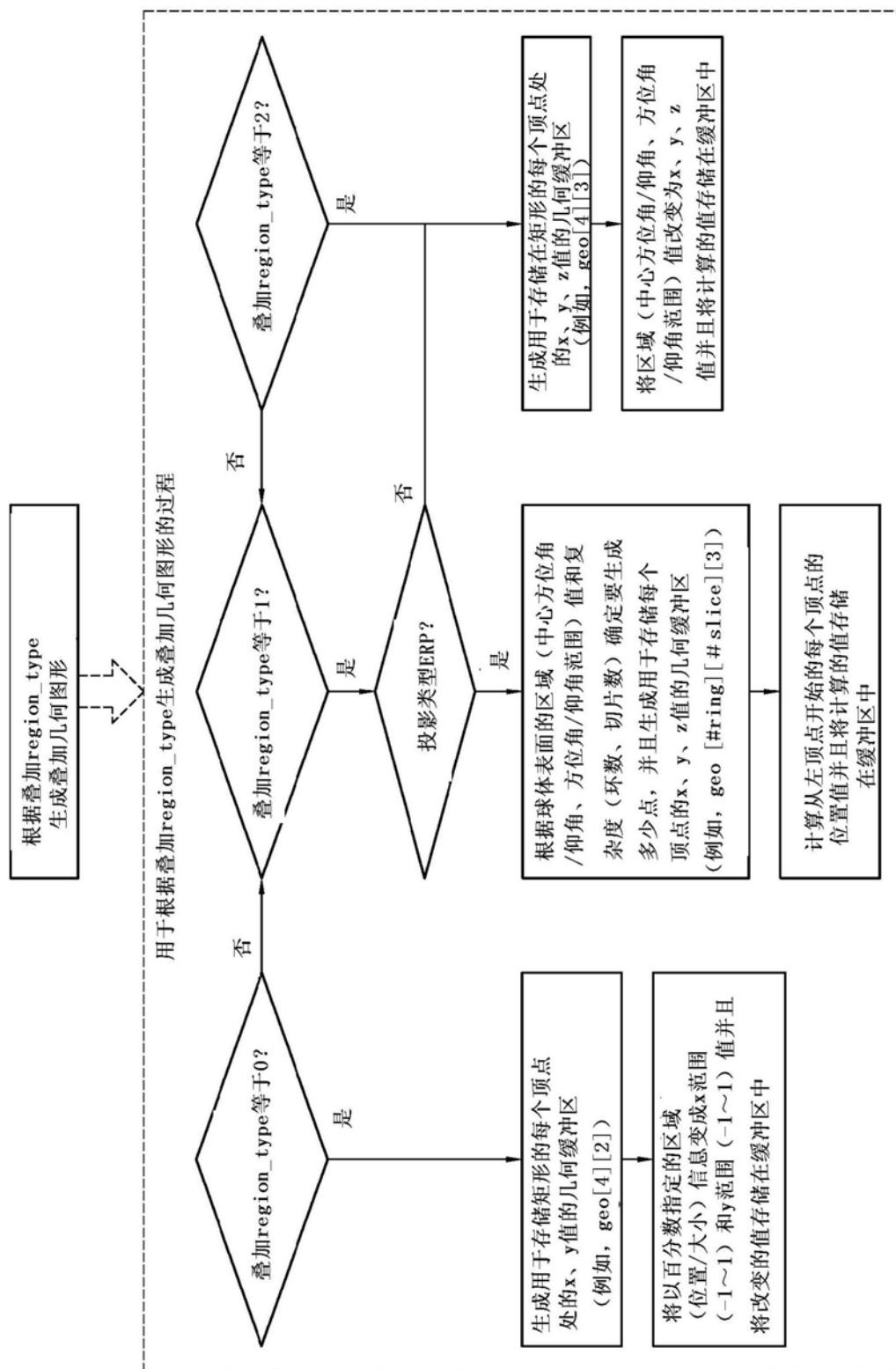


图40

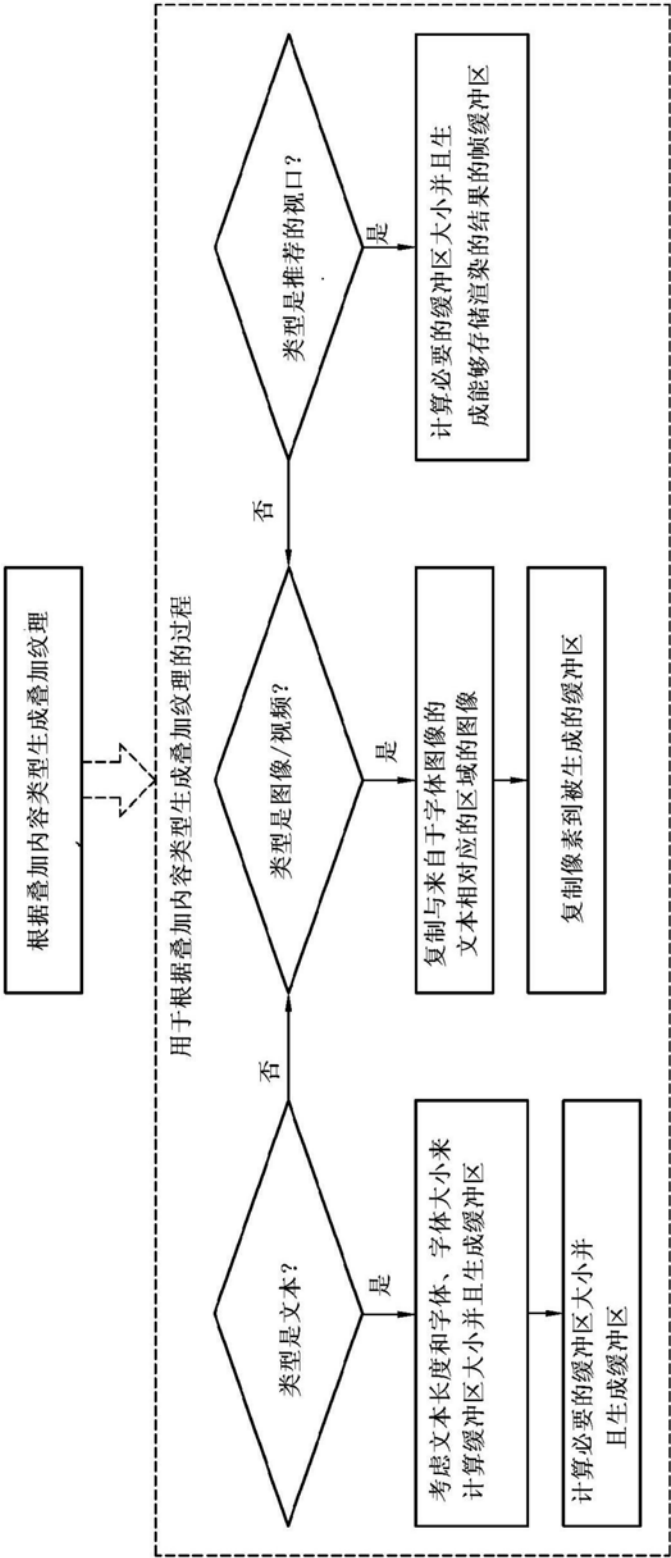


图41

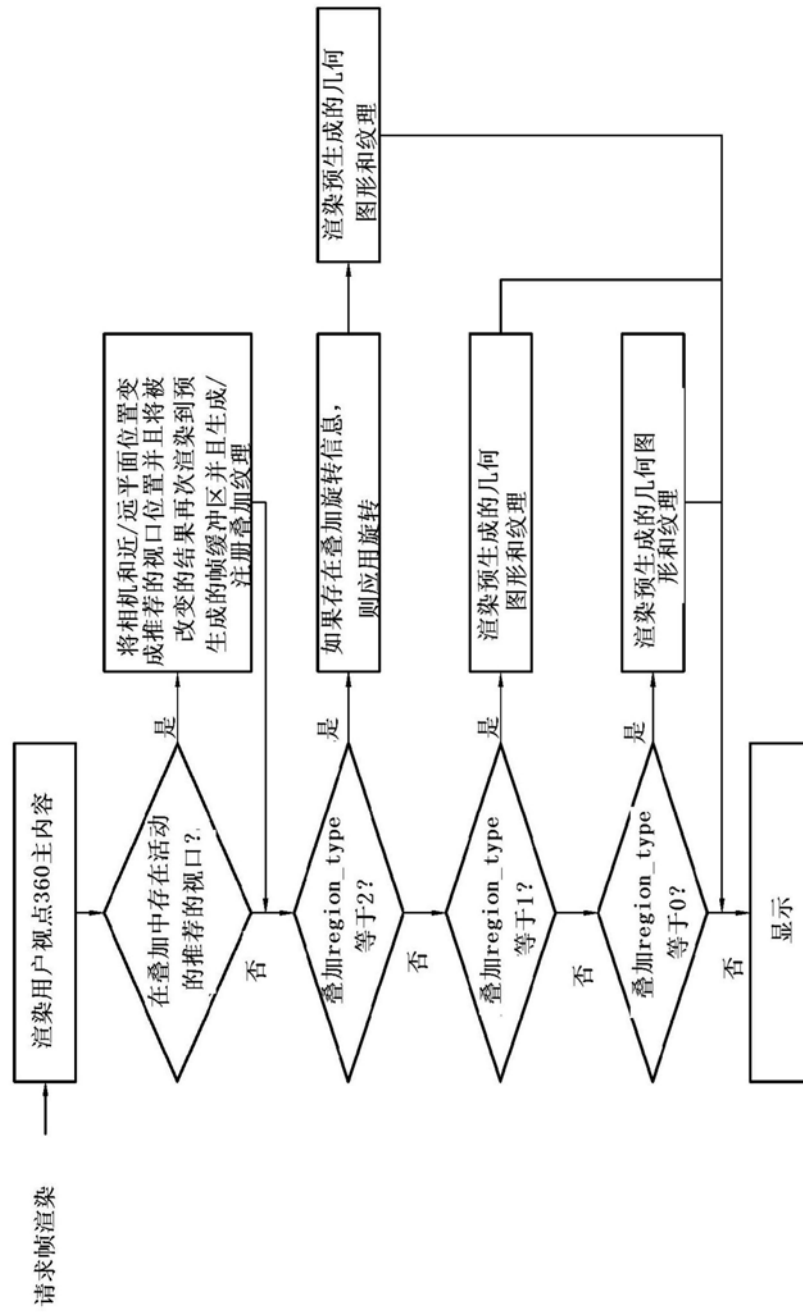


图42

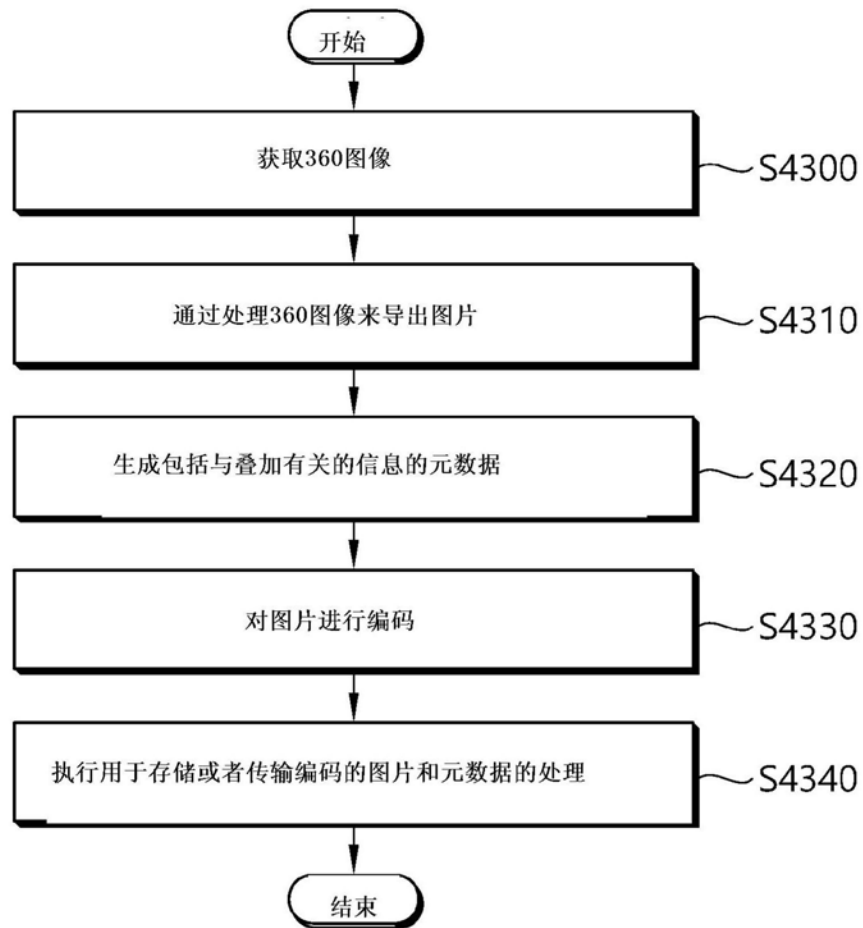


图43

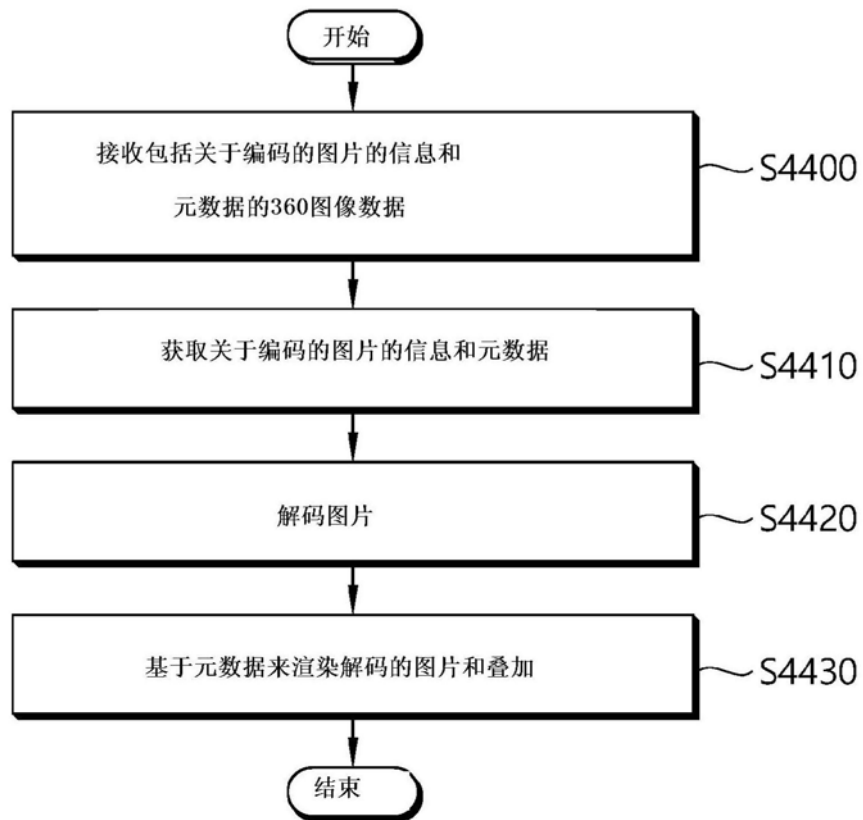


图44