



(12) 发明专利

(10) 授权公告号 CN 112806016 B

(45) 授权公告日 2024.12.03

(21) 申请号 201980065591.1

J·拉奇

(22) 申请日 2019.10.04

(74) 专利代理机构 北京润平知识产权代理有限公司 11283

(65) 同一申请的已公布的文献号
申请公布号 CN 112806016 A

专利代理人 肖冰滨 王晓晓

(43) 申请公布日 2021.05.14

(51) Int.CI.

H04N 19/597 (2006.01)

(30) 优先权数据

(56) 对比文件

18306318.9 2018.10.05 EP
19305307.1 2019.03.14 EP

Rufael Mekuria等.Design, Implementation, and Evaluation of a Point Cloud Codec for Tele-Immersive Video.IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY, VOL. 27, NO. 4.2017, 第831页右栏和图4.

(85) PCT国际申请进入国家阶段日

Sebastian Schwarz等.2D VIDEO CODING

2021.04.02

OF VOLUMETRIC VIDEO DATA.2018 Picture Coding Symposium (PCS), IEEE.2018, 第62页第3节, 图2、5-6.

(86) PCT国际申请的申请数据

审查员 吴倩倩

PCT/US2019/054637 2019.10.04

权利要求书2页 说明书16页 附图7页

(87) PCT国际申请的公布数据

W02020/072865 EN 2020.04.09

(73) 专利权人 交互数字VC控股公司

地址 美国特拉华州

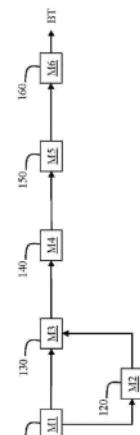
(72) 发明人 Y·奥利维尔 J·C·切维特

(54) 发明名称

一种用于编码/重构点云的点的属性的方法和设备

(57) 摘要

提供了一种编码/解码点云的点的属性的方法和装置。该编码方法对界定了待编码点的边界框进行细分，通过参考点所属的子部分来编码表示该点的属性的数据；并对表示每个子部分中包括的点的数量的数据进行编码。



1. 一种方法,包括:

-将界定了点云的点的边界框细分为多个子部分,其中所述子部分取决于用以编码所述点的编码器的比特深度;以及

-通过参考所述点云的至少一个点所属的子部分,对表示该至少一个点的属性的数据进行编码,其中所述子部分的所述至少一个点的3D坐标被编码为从所述子部分在几何图像内编码的2D面片中的参考点的坐标获得的差分坐标,所述2D面片专用于所述子部分的未由通过将所述点云投影到投影平面上获得的几何和属性图像所表示的点。

2. 根据权利要求1所述的方法,其中所述方法进一步包括:对表示每一子部分中所包括的点的数量的数据进行编码。

3. 根据权利要求1所述的方法,其中点的属性对应于所述点的几何形状或纹理/颜色信息。

4. 一种方法,包括:

-通过参考点云的点所属的子部分,对表示该点的属性的数据进行解码,所述子部分是通过分割界定了所述点云的所述点的边界框而获得的多个子部分之一,其中所述子部分取决于用以解码所述点的解码器的比特深度;以及

-从所解码的数据中重构所述点云的至少一个点的属性,这其中包括:从解码的几何图像,获得用于所述子部分的2D面片,以及从获取自所述2D面片的像素值的差分坐标以及该2D面片的参考点的坐标,获得所述至少一个点的3D坐标,所述2D面片专用于所述子部分的未由通过将所述点云投影到投影平面上获得的几何和属性图像所表示的点。

5. 根据权利要求4所述的方法,其中所述方法进一步包括:解码表示所述边界框的每一子部分中所包括的点的数量的数据,以及进一步从所述点的数量重构点的属性。

6. 根据权利要求4所述的方法,其中点的属性通过解码至少一几何和属性图像而从所解码的数据被重构,所解码的数据重构所述点云的点。

7. 根据权利要求4所述的方法,其中所述方法进一步包括:当子部分包括所述点中的至少一者时,获得所述子部分的2D面片。

8. 根据权利要求4所述的方法,其中点的属性对应于所述点的几何形状或纹理/颜色信息。

9. 一种包括处理器的装置,其中所述处理器被配置为执行以下步骤:

-将界定了点云的点的边界框细分为多个子部分,其中所述子部分取决于用以编码所述点的编码器的比特深度;以及

-通过参考所述点云的至少一个点的所属的子部分,对表示所述至少一个点的属性的数据进行编码,其中所述子部分的所述至少一个点的3D坐标被编码为从所述子部分在几何图像内编码的2D面片中的参考点的坐标获得的差分坐标,所述2D面片专用于所述子部分的未由通过将所述点云投影到投影平面上获得的几何和属性图像所表示的点。

10. 一种包括处理器的装置,其中所述处理器被配置为执行以下步骤:

-通过参考点云的点所属的子部分,对表示该点的属性的数据进行解码,所述子部分是通过分割界定了所述点云的所述点的边界框而获得的多个子部分之一,其中所述子部分取决于用以解码所述点的解码器的比特深度;以及

-从所解码的数据中重构所述点云的至少一个点的属性,这其中包括:从解码的几何图

像,获得用于所述子部分的2D面片,以及从获取自所述2D面片的像素值的差分坐标以及该2D面片的参考点的坐标,获得所述至少一个点的3D坐标,所述2D面片专用于所述子部分的未由通过将所述点云投影到投影平面上获得的几何和属性图像所表示的点。

11. 根据权利要求10所述的装置,其中所述处理器进一步被配置为:

解码表示所述边界框的每一子部分中所包括的点的数量的数据,以及进一步从所述点的数量重构点的属性。

12. 根据权利要求10所述的装置,其中点的属性通过解码至少一几何和属性图像而从所解码的数据被重构,所解码的数据重构所述点云的点。

13. 一种非暂时性计算机可读介质,其包含用于致使一个或多个处理器执行以下操作的指令:

-通过参考点云的点所属的子部分,对表示该点的属性的数据进行解码,所述子部分是通过分割界定了所述点云的所述点的边界框而获得的多个子部分之一,其中所述子部分取决于用以解码所述点的解码器的比特深度;以及

-从所解码的数据中重构所述点云的至少一个点的属性,这其中包括:从解码的几何图像,获得用于所述子部分的2D面片,以及从获取自所述2D面片的像素值的差分坐标以及该2D面片的参考点的坐标,获得所述至少一个点的3D坐标,所述2D面片专用于所述子部分的未由通过将所述点云投影到投影平面上获得的几何和属性图像所表示的点。

一种用于编码/重构点云的点的属性的方法和设备

技术领域

[0001] 本实施例主要涉及点云的点的编码和重构。具体地,但不排除其他,本实施例的技术领域涉及表示3D对象的外表面的点云的编码/重构。

背景技术

[0002] 本部分旨在向读者介绍可能与下面描述和/或要求保护的本发明实施例的各方面相关的技术的各方面。相信该讨论有助于向读者提供背景信息,以便于更好地理解本实施例的各个方面。因此,应当理解,这些陈述应就此而论来阅读,而不是作为对现有技术的承认。

[0003] 点云是某一坐标系中的一组点(数据点)。在三维坐标系(3D空间)中,这些点通常旨在表示3D对象的外表面。点云的每个点通常由其位置(3D空间中的X、Y和Z坐标)定义,并且可能由其他相关属性定义,例如颜色(在例如RGB或YUV颜色空间中表示的颜色)、透明度、反射率、两分量法矢量等。

[0004] 通常将点云表示为一组6分量点(X,Y,Z,R,G,B)或等效地(X,Y,Z,Y,U,V),其中(X,Y,Z)定义3D空间中的色点的坐标,并且(R,G,B)或(Y,U,V)定义该色点的颜色。

[0005] 点云可以是静态的或动态的,这取决于该云是否相对于时间演变。应当注意,在动态点云的情况下,点的数量不是恒定的,而是相反,通常随着时间演变。因此,动态点云是一组点的时间顺序列表。

[0006] 实际上,点云可以用于各种目的,诸如文化遗产/建筑,其中像雕像或建筑的对象以3D扫描,以便共享该对象的空间配置而不发送或访问它。而且,这是一种确保在对象可能被破坏(例如被地震破坏的寺庙)的情况下保留关于该对象的知识的方式。这种点云通常是静态的、有颜色的和巨大的。

[0007] 另一使用情况是在使用3D表示的地形图和制图中,地图不限于平面且可包含起伏。谷歌地图现在是3D地图的良好示例,但是使用的是网格而不是点云。然而,点云可以是用于3D地图的合适的数据格式,并且这样的点云通常是静态的、有颜色的和巨大的。

[0008] 汽车工业和自动驾驶汽车也是其中可以使用点云的领域。自动驾驶汽车应当能够“探测”它们的环境,以基于它们紧邻的现实来做出良好的驾驶决定。像LIDAR的典型传感器产生由决策引擎所使用的动态点云。这些点云不旨在被人类查看,并且它们通常是小的,不一定是有色的,并且是动态的,具有高捕获频率。它们可以具有其他属性,例如由LIDAR提供的反射率,因为该属性是关于被感测物体的材料的良好信息并且可以帮助进行所述决策。

[0009] 虚拟现实和沉浸式世界近来已经成为热点主题,并且被很多人预见为2D平面视频的未来。基本思想是使观看者沉浸在他周围的环境中,这与标准TV相反,在标准TV中,观看者仅能够看到他前方的虚拟世界。根据观看者在环境中的自由度,在所述沉浸度中存在若干层次。有色点云是分布虚拟现实(或VR)世界的良好格式候选。它们可以是静态的或动态的,并且通常具有平均的大小的,比如一次不超过数百万个点。

[0010] 仅当比特流的大小足够低以允许对终端用户的实际存储/传输时,点云压缩才将

成功地存储/传输用于沉浸式世界的3D对象。

[0011] 能够以比特率的合理消耗而将动态点云分发给终端用户同时保持可接受的(或者优选地非常好的)体验质量是至关重要的。为了使沉浸式世界的分发链切实可行,这些动态点云的有效压缩是关键点。

[0012] 基于图像的点云压缩技术由于其压缩效率和低复杂度的组合而变得越来越流行。它们以两个主要步骤进行:首先,它们将点云(即,点)投影(正交投影)到2D图像上,该2D图像表示与所述点云的点相关联的属性。例如,至少一个几何图像表示所述点云的几何形状,即,所述点在3D空间中的空间坐标,并且至少一个属性图像表示与所述点云的所述点相关联的属性,例如与那些点相关联的纹理/颜色信息。接着,这些技术用传统视频编码器对这种深度和属性图像进行编码。

[0013] 基于图像的点云压缩技术通过利用2D视频编码器的性能来实现良好的压缩性能,同时它们通过使用简单的投影方案来保持低复杂度,所述编码器例如HEVC (“ITU的ITU-T.265电信标准化部门(10/2014),系列H:视听和多媒体系统,视听服务的基础设施-运动视频的编码、高效视频编码,ITU-T.265建议 (ITU-T H.265 Telecommunication standardization sector of ITU(10/2014), series H: audiovisual and multimedia systems, infrastructure of audiovisual services-coding of moving video, High efficiency video coding, Recommendation ITU-T H.265)”)。

[0014] 基于图像的点云压缩技术的挑战之一在于:点云可能不适合投影到图像上,尤其是在点分布遵循具有许多褶皱的表面(凹/凸区域,如在衣服中)时或者在点分布根本不遵循表面(如在皮毛或毛发中)时。在这些情况下,基于图像的点云压缩技术遭受低压缩效率(需要许多小投影,从而降低2D视频压缩的效率)或质量差(由于将点云投影到表面上的困难)。

[0015] 在现有技术中使用的减轻这个问题的方法之一在于将多个几何形状和纹理信息投影到图像的相同空间位置(像素)上。这意味着,可以为点云的每个点生成若干几何和/或属性图像。

[0016] 例如,在ISO/IEC JTC1/SC29/WG11 MPEG2018/N17767,Ljubljana,2018年7月(附录A)中定义的所谓的测试模型类型2点云编码器(TMC2)中便是这种情况。

[0017] 在TMC2中,点云被正交地投影到投影平面上。针对所述投影平面的每个坐标,两个深度值因此被关联:一个代表与最近点相关联的深度值(最小深度值),另一个代表最远点的深度值(最大深度值)。然后,从最小深度值(D0)生成第一几何图像,从最大深度值的绝对值(D1)生成第二几何图像,其中D1-D0小于或等于最大表面厚度。还产生与第一(D0)和第二(D1)几何图像相关联的第一和第二属性图像。然后,使用诸如HEVC的任何传统视频编解码器对所述属性图像和几何图像进行编码和解码。因此,通过对所解码的第一和第二几何图像中进行去投影来重构所述点云的几何形状,并且与所重构的点云的点相关联的其他属性通过所解码的属性图像而被获得。

发明内容

[0018] 以下给出了本发明实施例的简化概述,以便提供对本发明实施例的一些方面的基本理解。该概述不是对本实施例的广泛综述。其不旨在标识本实施例的关键或重要元素。以

下概述仅以简化形式呈现本实施例的一些方面,作为以下提供的更详细描述的序言。

[0019] 本发明的实施例被提出以利用一种方法来弥补现有技术的至少一个缺失点,该方法包括:

[0020] -将界定了点云的点的边界框(bounding box)细分(subdividing)为多个子部分(subdivisions);

[0021] -通过参考所述点云的点所属的子部分,对表示该点的属性的数据进行编码;以及

[0022] -对表示每一子部分中所包括的点的数量的数据进行编码。

[0023] 根据实施例,该方法还包括:获得图像的单个2D面片(patch),所述面片包括表示所述点的所述属性的数据。

[0024] 根据实施例,该方法还包括:当子部分包括所述点中的至少一个时,获得用于该子部分的2D面片。

[0025] 根据实施例,所述边界框的所述子部分取决于用以编码所述点的编码器的比特深度。

[0026] 根据实施例,其中点的属性对应于所述点的几何形状或纹理/颜色信息。

[0027] 根据实施例,所述点通过比较输入点云的点和重构点云的点来获得,所述重构点云通过以下步骤而被获得:

[0028] -通过将所述输入点云的一些点投影到一投影平面上而获得至少一个几何和属性图像;

[0029] -对所述至少一个几何和属性图像进行编码;以及

[0030] -通过解码所述至少一个几何和属性图像,重构所述重构点云的点。

[0031] 根据本发明的另一个方面,至少一个本发明实施例涉及一种方法,该方法包括:

[0032] -通过参考点云的点所属的子部分,对表示该点的属性的数据进行解码,所述子部分是通过分割界定了所述点云的所述点的边界框而获得的多个子部分之一;以及

[0033] -解码表示所述边界框的每一子部分中所包括的点的数量的数据;以及

[0034] -从所述解码数据中重构点的属性。

[0035] 根据实施例,点的属性通过解码至少一几何和属性图像而从所述解码数据被重构,所述解码数据重构所述重构点云的点。

[0036] 根据实施例,重构点的属性进一步包括:从解码的几何图像,获得用于子部分的2D面片,以及从获取自所述2D面片的像素值的坐标及该2D面片的参考点的坐标,获得点的3D坐标。

[0037] 根据实施例,该方法还包括:当子部分包括所述点中的至少一者时,获得所述子部分的2D面片。

[0038] 根据实施例,所述边界框的所述子部分取决于用于解码所述点的解码器的比特深度。

[0039] 根据一个实施例,点的属性对应于所述点的几何形状或纹理/颜色信息。

[0040] 根据本发明的其它方面,本发明的实施例涉及装置、比特流、计算机程序产品和非暂时性可读介质。

[0041] 从结合附图进行的以下示例描述中,本发明实施例的特定性质以及本发明实施例的其它目的、优点、特征和用途将变得显而易见。

附图说明

- [0042] 在附图中,示出了本实施例的示例。其显示了:
- [0043] -图1示意性地示出了根据本实施例的示例的编码点云的点的属性的方法100的步骤的图;
- [0044] -图2示意性地示出了根据本实施例的示例的从比特流解码表示点云的数据的方法200的步骤的图;
- [0045] -图3示出了根据本实施例的示例的通过通信网络进行通信的两个远程设备;
- [0046] -图4示出了根据本实施例的示例的信号的语法;
- [0047] -图5-10示出了根据实施例的用于携带解码器重构点云的缺失点所需的信息的语法元素的示例;以及
- [0048] -图11示出了根据本发明实施例的示例实现了各个方面和实施例的系统的示例的框图。
- [0049] 相似或相同的元件用相同的参考数字表示。
- [0050] 本发明实施例的描述

[0051] 以下将参考附图更全面地描述本发明,在附图中示出了本发明的示例。然而,本发明的实施例可以以许多替代形式实施,且不应解释为限于本文所陈述的示例。因此,尽管本发明的实施例易于进行各种修改和替换形式,但是其具体示例在附图中以示例的方式示出并且将在此详细描述。然而,应理解,不希望将本发明实施例限于所揭示的特定形式,而是相反,本说明书将涵盖属于如权利要求书所界定的本发明实施例的精神和范围内的所有修改、等效物和替代方案。

[0052] 本文所用的术语仅是为了描述特定示例的目的,而不是要限制本发明的实施例。如本文所用,单数形式“一(a)”、“一(an)”和“该”旨在也包括复数形式,除非上下文另有明确指示。还将理解,术语“包括”和/或“包含”在本说明书中使用时,指定所陈述的特征、整数、步骤、操作、元件和/或组件的存在,但不排除一个或多个其它特征、整数、步骤、操作、元件、组件和/或其群组的存在或添加。此外,当元件被称为“响应”或“连接”到另一元件时,它可以直接响应或连接到另一元件,或者可以存在中间元件。相反,当元件被称为“直接响应”或“直接连接”到其它元件时,不存在中间元件。如本文所用,术语“和/或”包括一个或多个相关的所列项目的任意者和所有组合,并且可以缩写为“/”。

[0053] 应当理解,尽管术语第一、第二等可以在这里用来描述各种元件,但是这些元件不应当受这些术语的限制。这些术语仅用于将一个元件与另一个元件区分。例如,在不脱离本实施例的教导的情况下,第一元件可以被称为第二元件,并且类似地,第二元件可以被称为第一元件。

[0054] 尽管一些图包括通信路径上的箭头以示出通信的主要方向,但是应当理解,通信可以在与所描绘的箭头相反的方向上发生。

[0055] 关于框图和操作流程图描述了一些示例,其中每个框表示电路元件、模块或代码的一部分,其包括用于实现指定的逻辑功能(多个)的一个或多个可执行指令。还应当注意,在其它实现中,在所述框中所提到的(一个或多个)功能可以不按所提到的顺序发生。例如,取决于所涉及的功能,连续示出的两个框实际上可以基本上同时执行,或者这些框有时可以以相反的顺序执行。

[0056] 本文中提到“根据示例”或“在示例中”意味着结合该示例描述的特定特征、结构或特性可包含在本实施例的至少一个实施方式中。在说明书中的各个地方出现的根据示例“或”在示例中“的短语不一定全部指代相同的示例，也不是必须与其他示例互斥的单独的或替代的示例。

[0057] 权利要求中出现的附图标记仅作为说明，并且不应对权利要求的范围具有限制作用。

[0058] 虽然没有明确描述，但是本示例和变型可以以任何组合或子组合来采用。

[0059] 在下文中，术语“图像区域”指定至少一个图像的一组像素。这些像素可以是相邻像素，也可以不是相邻像素，和/或可以属于同一图像，也可以不属于同一图像，但它们都具有至少一个共同的属性。

[0060] 例如，整个图像本身可以被认为是图像区域。图像也可以被分割成多个块，并且一个块或一组块于是成为图像区域。

[0061] 图像区域也可具有非矩形形状。例如，当图像的具有相同(或相似)提取特征的像素被关联以形成一图像区域时，就是这种情况。图像区域可以是TMC2中定义的面片。

[0062] 从图像提取的特征的示例可以是颜色、纹理、法向矢量等。

[0063] 一些图表示了TMC2中广泛使用的语法表，用于定义符合TMC2的比特流的结构。在这些语法表中，术语“,,,”表示相对于TMC2中给出的原始定义的语法的未改变部分，并且在图中被去除以便于阅读。图中的粗体项指示该项的值是通过解析比特流获得的。语法表的右列指示用于编码语法元素的数据的比特数。例如，u(4)指示4比特用于编码数据，u(8)指示8比特，ae(v)指示上下文自适应算术熵编码的语法元素。

[0064] 如本文所使用的语法元素是描述性术语。因此，它们不排除使用其它语法元素名称。

[0065] 本实施例被描述用于编码/解码表示点云的一个或多个几何图像和一个或多个属性图像，但是可扩展到编码/解码表示点云序列(时间动态点云)的两个序列(视频)的几何图像和两个序列(视频)的属性图像，因为所述点云序列的点云的几何形状(两个几何图像)和属性(两个属性图像)然后独立于所述点云序列的另一点云的几何形状(两个几何图像)和属性(两个属性图像)而被编码/解码。

[0066] 当编码和/或重构点云时，数据可能缺失，这导致输入点云的部分重构。这种信息缺失在渲染所述输入点云时可能产生问题。

[0067] TMC2为有损压缩情形提供了良好的速率失真性能。然而，当需要无损压缩时，压缩性能会显著降低，因为需要创建大量面片以捕获孤立(缺失)点。

[0068] 其它替代方案将孤立点编码为特殊的面片，而不考虑这些孤立点与在“规则”面片中捕获的那些点(点在投影平面上的投影的结果)之间的潜在相关性。基本思想是通过使用TMC2而将点云投影到投影平面上，并且捕获所述点云的缺失(孤立)点，即，不是由所述几何/属性图像表示的点。每个缺失/孤立点的X坐标、随后的Y坐标以及最后的Z坐标被堆叠到单个矢量中，并且生成被表示为“缺失点”面片的“特殊”面片。然后将该面片存储在10比特4:2:0几何图像的单通道(分量)中(更详细地参见ISO/IEC JTC1/SC29/WG11 MPEG2018/m42680, 2018年4月, San Diego, CA, USA)。

[0069] 这个替换方案把缺失点的坐标范围限制到一最大范围(当使用通常的10比特4:2:

0视频格式时为1024)。这是重点,因为缺失/孤立点的坐标被存储为绝对值,即,所存储的坐标值是从界定所述点云的边界框的原点(0,0,0)定义的。

[0070] 一般而言,至少一个实施例涉及一种编码/重构缺失/孤立点的属性的方法,但是该方法也可以应用于编码/重构点云的任何其他点的属性。因此,该方法可被认为是基于图像的点云压缩技术(例如TMC2)的替代,用于编码(和重构)点云的点的属性。该方法也可以被认为是通过提供点云的无损压缩(编码缺失点)来改进诸如TMC2的基于有损图像的点云压缩技术的手段。

[0071] 所述编码方法细分用于界定待编码点的边界框,通过参考点所属的子部分来编码表示该点的属性的数据;并对表示每个子部分中包括的点的数量的数据进行编码。

[0072] 所述解码方法通过参考点所属的子部分来解码表示该点的所述属性的编码数据。所述子部分是通过分割界定所述点的边界框而获得的多个子部分中的子部分。所述方法还解码表示所述边界框的每个子部分中所包括的点的数量的编码数据;以及从所述解码数据对点进行重构。

[0073] 对界定待编码点的边界框进行细分以及通过参考点所属的子部分而对表示该点的属性的数据进行编码,减少了表示所述点的所述属性所需的比特数。例如,那些点的坐标值的范围与现有技术相比可以被减小,因为那些坐标可以从那些点所属的子部分的参考点而不是从所述边界框的参考点来定义。

[0074] 本发明允许对任何大小的点云图像/视频进行无损编码,而与利用视频TMC2点云编码器编码的视频的比特深度无关。

[0075] 图1示意性地示出了根据本实施例的示例的用于对输入点云的至少一些点的属性进行编码的方法100的操作的图。

[0076] 在操作110中,所述点的属性由编码器M1编码。所述属性可对应于几何形状(或深度或到视点的距离)信息和/或纹理(或颜色)信息。例如,点被“群集”为TMC2(参见TMC2)中描述的连接分量并被编码。例如,每个连接分量的点可以被投影在投影平面上,投影点的属性被存储在至少一对几何和属性图像的所谓2D面片(像素集合)中。表示用于获得所述2D面片的投影参数的元数据也可以由编码器M1编码。所述编码器M1例如与传统编码器兼容,诸如:

[0077] ●JPEG,规范ISO/CEI 10918-1UIT-T建议T.81,<https://www.itu.int/rec/T-REC-T.81/en>,

[0078] ●AVC,也称为MPEG-4AVC或h264。UIT-T H.264和ISO/CEI MPEG-4第10部分(ISO/CEI 14496-10)中有所规定,<http://www.itu.int/rec/T-REC-H.264/en>,

[0079] ●HEVC(其规范可在ITU网站,T推荐,H系列,H265,<http://www.itu.int/rec/T-REC-H.265-201612-I/en>找到),或

[0080] ●3D-HEVC(HEVC的扩展,其规范在ITU网站,T推荐,H系列,H265,<http://www.itu.int/rec/T-REC-H.265-201612-I/en>附录G和I中找到)。

[0081] 在操作120中,由解码器M2对利用编码器M1编码的数据进行解码,并且根据解码的数据(即,根据从解码的至少一个几何形状/属性图像获得的解码属性,以及根据表示所述投影参数的元数据),重构点云,例如如TMC2中所解释的。所述解码器M2与所述编码器M1兼容,例如与传统解码器兼容,诸如:

- [0082] ●JPEG，
[0083] ●AVC,也称为MPEG-4AVC或h264，
[0084] ●HEVC,或
[0085] ●3D-HEVC(HEVC的扩展)。
- [0086] 在由模块M3实现的操作130中,将重构的点云与所述输入点云进行比较。可以检测所述重构点云中的缺失且存在于所述输入点云中的缺失点。
- [0087] 在操作140中,将界定在操作130处检测到的缺失点的边界框细分成多个子部分。
- [0088] 根据实施例,界定所述缺失点的边界框可对应于界定所述输入点云的边界框。
- [0089] 根据实施例,所述边界框可以对应于被细分成多个子立方体的立方体(或更一般地,平行四边形)。
- [0090] 根据实施例,所述对边界框的所述子部分可以取决于特定的比特深度,例如,用于对所述输入点云的点和/或缺失点进行编码的编码器的比特深度(通常为8比特/10比特)。因此,子部分(例如子立方体的边)的大小和/或尺寸不超过所述特定的比特深度(可以通过使用至多所述最大比特数来表示)。
- [0091] 因此,该方法可以适用于在TMC2中编码几何和属性图像的编码器的约束(比特深度)。
- [0092] 根据一个实施例,使用基于八叉树的方法,将所述边界框分成多个子立方体。
- [0093] 基本上,一立方体递归地被分成八个较小的子立方体。根据在叶子立方体中是否存在至少一个缺失点,将该叶子立方体标记为已占用或未占用。
- [0094] 在实施例中,所述八叉树深度是恒定值。
- [0095] 在该实施例的变型中,立方体索引按照每个子立方体而被发送。
- [0096] 在该实施例的变型中,遍历顺序(例如,深度优先或宽度优先)被发送或假定。
- [0097] 在该实施例的一个变形中,所述八叉树深度被发送。
- [0098] 在实施例中,所述八叉树深度是可变的。
- [0099] 在该实施例的一个变形中,分割八叉树的被占用叶子被发送。
- [0100] 在该实施例的变型中,在缺失点面片中没有索引;面片元数据具有与遍历顺序相同的顺序。
- [0101] 在一种变型中,所述八叉树的深度取决于用于编码所述输入点云和/或所述缺失点的编码器的比特深度。
- [0102] 根据实施例,聚类算法可以用于对子立方体内部的缺失点进行重组,从而最小化“缺失点”面片的数量和/或最小化编码的缺失点的能量,同时考虑子立方体的最大尺寸的限制,这取决于视频编解码器的输入比特深度(2^n ,其中n等于所述比特深度)。利用这种算法,所述子立方体的原点完全不受限制。最小化“缺失点”面片的数量将降低编码的比特率。
- [0103] 在操作150中,通过模块M5获得一个或多个2D面片。所述一个或多个面片包括表示所述每个缺失点的属性的数据。
- [0104] 根据一个示例,单个2D面片被获得。
- [0105] 根据另一示例,当子部分包括所述缺失点中的至少一个时,一个2D面片被获得(或生成)以用于所述边界框的该子部分。
- [0106] 根据实施例,表示属于相对于子部分的2D面片的点的属性的数据被编码如下:

[0107] 对于每个缺失点,存储表示该缺失点的属性的数据,并将其与“缺失点”面片(单个2D面片或相对于子部分的2D面片)的像素相关联。然后,可以通过参考该“缺失点”面片的参考点来表示该数据。例如,缺失点的坐标由从包括所考虑的缺失点的子部分的参考点计算的dX、dY和dZ差分坐标来表示。将每个缺失点的dX坐标、随后的dY以及最后的dZ坐标堆叠到单个矢量中。然后,使用为属于所述子部分的所有缺失点生成的所有矢量来生成“缺失点”面片。

[0108] 接下来,将所有“缺失点”面片(每个子部分一个面片)打包在几何图像中,因此该几何图像包括表示所述输入点云的至少一些点的属性的“规则”面片(步骤110)和表示缺失点的属性的“缺失点”面片。

[0109] 在操作160中,通过参考缺失点所属的子部分来对表示该缺失点的属性的数据进行编码。表示每个子部分中所包括的缺失点的数量的数据也可以由编码器M6编码。

[0110] 为了说明的目的,给出了用于编码表示所述缺失点的属性的数据的以下算法:

[0111] -如果要编码的输入点云被包含在大小为 2048^3 的一个边界框中,并且视频编解码器具有10比特深度(例如HEVC主10)

[0112] -所述边界框可以被分成8个大小为 1024^3 的子立方体

[0113] -对每个子立方体

[0114] ○如果内部有至少一个缺失点

[0115] ■创建存储所述边界框中的所述子立方体的位置的MissingPointsPatch对象

[0116] ●更新所述子立方体的原点(u_1, v_1, d_1)

[0117] ●更新缺失点的数量

[0118] ■创建几何图像中的2D面片MissingPointsPatch(打包算法)

[0119] ●更新所述几何图像中对解码过程有用的位置(u_0, v_0)

[0120] ●顺序地添加所述几何图像中每个缺失点(X, Y, Z)的差分坐标(dX, dY, dZ):

[0121] o $dX = X - u_1; dY = Y - v_1; dZ = Z - d_1;$

[0122] ●顺序地添加属性图像中的相应的属性(颜色分量

[0123] (R, G, B))

[0124] 比特流BT被获得。它包括代表所述缺失点的所述属性的编码数据。所述编码数据参考了所述缺失点所属的子部分。还可能对表示每个子部分中包括的缺失点的数量的数据进行编码。

[0125] 所述比特流BT可以例如为了解码目的和/或重构点云的渲染而被传输。

[0126] 图1的编码方法例如可以结合到根据TMC2的编码过程中。

[0127] 图2示意性地示出了根据本实施例的示例的解码点云的点的属性的方法200的操作的图。

[0128] 在操作210中,解码器M7可解码包括在比特流BT中的编码数据。所述比特流BT可以是图1的方法的输出。所述编码数据表示点的属性。

[0129] 所述比特流BT还可以包括表示被包括在通过细分界定所述点的边界框而获得的多个子部分中的每个子部分中的点的数量的编码数据。

[0130] 解码器M7可通过参考点所属的子部分来解码每个子部分中包括的点的数量和表示点的属性的编码数据。

[0131] 所述解码器M7可以对应于图1的解码器M2，并且例如与传统解码器兼容，诸如：

[0132] ●JPEG，

[0133] ●AVC，也称为MPEG-4 AVC或H264，

[0134] ●HEVC，或

[0135] ●3D-HEVC (HEVC的扩展)。

[0136] 在由模块M8实现的操作220中，所述输入点云的至少一部分被从在先前操作中获得的解码数据重构。因此，所重构的点云 \widehat{PC} 至少包括重构版本中的缺失点。

[0137] 图2的解码方法例如可以结合到根据TMC2的解码过程中。

[0138] 根据实施例，可以从表示缺失点的属性的解码数据并且通过参考子部分来重构缺失点如下：

[0139] 从解码的几何图像中解码“缺失点”片。然后，从“缺失点”面片中获得相对于缺失点的坐标矢量，并从该矢量中检索相对于该缺失点的dX、dY、dZ坐标。

[0140] 然后，通过将所述dX、dY、dZ坐标加到所述“缺失点”面片 (PatchList[i]的元素) 的参考点的坐标 (u1, v1, z1) 来给出所述缺失点的3D坐标 (Xrec, Yrec, Zrec)：

[0141] $X_{rec} = dX + PatchList[u].u1$

[0142] $Y_{rec} = dY + PatchList[i].v1$

[0143] $Z_{rec} = dZ + PatchList[i].d1$

[0144] 所述重构点的相关属性从属性图像中获得，例如作为位于 (Xrec, Yrec) 的像素值。

[0145] 根据一个实施例，当一子部分包括至少一个缺失点时，获得用于该子部分的“缺失点”面片。

[0146] 根据实施例，边界框的子部分取决于用于对所述缺失点进行解码的解码器的比特深度。

[0147] 根据一个实施例，缺失点的属性对应于所述缺失点的几何形状或纹理/颜色信息。

[0148] 根据图3中所示的本实施例的示例，在两个远程设备A和B之间通过通信网络NET的传输上下文中，设备A包括与存储器RAM和ROM相关的处理器，它们被配置为实现如关于图1所描述的方法，并且设备B包括与存储器RAM和ROM相关的处理器，它们被配置为实现如关于图2所描述的方法。

[0149] 根据一个示例，所述网络是广播网络，其适于将静止图像或视频图像从设备A广播到包括所述设备B的解码设备。

[0150] 打算由设备A发送的信号承载了所述比特流BT。所述比特流BT包括编码数据，该编码数据表示点云的点的深度值、所述点云的至少一个重构点、以及至少一个内插编码模式，该模式指示所述至少一个重构点的邻域中的至少一个点如何被至少一个其他内插点近似的。

[0151] 例如，如关于图1所解释的，获得所述编码数据。

[0152] 图4示出了当通过基于分组的传输协议传输数据时这种信号的语法的示例。每个传输的分组P包括报头H和有效载荷PAYLOAD。TMC2中给出了包括在这种信号中的语法元素的示例。

[0153] 为了重构缺失点的几何形状 (2D位置)，解码器需要知道子部分的参考点 (通常是其原点) (x1, y1, z1) 以及每个“缺失点”面片的缺失点的数量 (num_of_mps)，以重构所有这

些缺失点。

[0154] TMC2的多个语法元素用于承载重构缺失点所需的编码数据。

[0155] 图5示出了根据实施例的用于携带解码器重构点云的缺失点所需的信息的语法元素的示例。

[0156] 此语法元素是基于TMC2中定义的“group_of_frames_auxiliary_information”语法元素的。

[0157] 语法元素patch_count指示包括“缺失点”面片的面片的总数。语法元素mps_patch_count指示几何图像中的“缺失点”面片的数量，并且语法元素num_of_mps[i]也被添加以指示每个“缺失点”面片i中的缺失点的数量，即，几何图像中的最后面片（来自“patch_count-mps_patch_count”）是“缺失点”面片。

[0158] 图6示出了根据实施例的用于携带解码器重构点云的所有缺失点所需的信息的语法元素的示例。

[0159] 此语法元素是基于TMC2中定义的“group_of_frames_auxiliary_information”语法元素的。

[0160] 在该实施例中，最后的面片是几何图像中的“缺失点”面片。

[0161] 语法元素patch_count指示除了“缺失点”面片之外的面片的总数。语法元素mps_patch_count指示几何图像中的“缺失点”面片的数量，并且语法元素num_of_mps[i]也被添加以指示每个“缺失点”面片i中的缺失点的数量，即，几何图像中的最后面片（来自“patch_count-mps_patch_count”）是“缺失点”面片。

[0162] 图7示出了根据实施例的用于携带解码器重构点云的缺失点所需的信息的语法元素的示例。

[0163] 此语法元素是基于如TMC2中所定义的“序列参数集”语法元素的。

[0164] 语法元素sps_pcm_multiple_patch_present_flag指示是否使用多个PCM面片。

[0165] 语法元素sps_pcm_relative_3D_offset指示3D偏移是在PCM面片语法中以绝对还是相对方式传输。值0指示绝对模式，值1指示相对模式（用一个索引对偏移值进行编码）。

[0166] 语法元素sps_pcm_relative_3D_offset指示PCM面片的最大数量。当模式sps_pcm_relative_3D_offset_flag被设置为1时，该值用于编码PCM面片索引。

[0167] 图8示出了根据实施例的用于携带解码器重构点云的缺失点所需的信息的语法元素的示例。

[0168] 该语法元素基于TMC2中定义的“patch_frame_data_unit”语法元素。

[0169] 添加的语法（灰色阴影区域）添加了关于附加类型的面片数据单元的解析，用于发信号通知EOM纹理面片的存在。

[0170] 图9示出了根据实施例的用于携带解码器重构点云的缺失点所需的信息的语法元素的示例。

[0171] 该语法元素基于TMC2中定义的“pcm_patch_data_unit”语法元素的。

[0172] 图10示出了根据实施例的用于携带解码器重构点云的缺失点所需的信息的语法元素的示例。

[0173] 该语法元素基于TMC2中定义的“pcm_patch_data_unit”语法元素。

[0174] 附加数据被引入：

[0175] pdu_3d_pcm_shift_tangent_axis[frmIdx][patchIndex]指定了要沿着切向轴应用于面片帧frmIdx的索引p的面片中的重构的PCM面片点的移位。pdu_3d_shift_tangent_axis[frmIdx][p]的值应当在 0 到 $\text{Min}(2^{\text{pfh}_3d_shift_tangent_axis_bit_count_minus1[frmIdx]+1}, 2^{\text{gps_geometry_3d_coordinates_bitdepth_minus1+1}}-1)$ 的范围内,包括端值。

[0176] pdu_3d_pcm_shift_bitangent_axis[frmIdx][patchIndex]指定了要沿着双切向(bitangent)轴应用于面片帧frmIdx的索引p的面片中的PCM重构面片点的移位。pdu_3d_shift_bitangent_axis[frmIdx][patchIndex]的值应当在 0 到 $\text{Min}(2^{\text{pfh}_3d_shift_bitangent_axis_bit_count_minus1[frmIdx]+1}, 2^{\text{gps_geometry_3d_coordinates_bitdepth_minus1+1}}-1)$ 的范围内,包括端值。

[0177] pdu_3d_pcm_shift_normal_axis[frmIdx][patchIndex]指定了要沿着法线轴应用于面片帧frmIdx的索引p的面片中的PCM重构面片点的移位。pdu_3d_shift_normal_axis[frmIdx][patchIndex]的值应当在 0 到 $\text{Min}(2^{\text{pfh}_3d_shift_normal_axis_bit_count_minus1[frmIdx]+1}, 2^{\text{gps_geometry_3d_coordinates_bitdepth_minus1+1}}-1)$ 的范围内,包括端值。

[0178] dpdu_3d_relative_shift_index[frmIdx][patchIndex]指定了可允许计算三个3D移位值(切向、双切向和正态)的索引。在xxxxx中描述的这个索引的解码过程计算3个以下参数:

[0179] ● pdu_3d_pcm_shift_tangent_axis[frmIdx][p]

[0180] ● pdu_3d_pcm_shift_bitangent_axis[frmIdx][p]

[0181] ● pdu_3d_pcm_shift_normal_axis[frmIdx][p]

[0182] 根据实施例,所述编码方法可以生成一个附加的面片:表示缺失点的属性的OEM纹理面片。该面片可以被打包在属性图像中。所述附加面片的面片信息也可被编码。

[0183] 根据实施例,所述解码方法可以获得遵循上述语法之一的OEM纹理面片。根据EOM纹理面片中的参数(以及最终固定的参数,如排序和光栅化模式),可以从所述EOM纹理面片获得缺失点的属性。

[0184] 根据实施例,dpdu_3d_relative_shift_index被解码如下:

[0185] 该过程的输入是:

[0186] dpdu_3d_relative_shift_index

[0187] 来自geometry_parameter_set()的gps_geometry_nominal_2d_bitdepth_minus1

[0188] 来自geometry_parameter_set()的gps_geometry_3d_coordinates_bitdepth_minus1

[0189] 该过程的输出:

[0190] pdu_3d_pcm_shift_tangent_axis

[0191] pdu_3d_pcm_shift_bitangent_axis

[0192] pdu_3d_pcm_shift_normal_axis

[0193] 所述dpdu_3d_relative_shift_index指定了要解码的索引。gps_geometry_nominal_2d_bitdepth_minus1指示了所有几何视频将被转换成的标称2D比特深度。gps_geometry_nominal_2d_bitdepth_minus1应该在 0 到 31 的范围内,包括端值。gps_geometry_3d_coordinates_bitdepth_minus1指示重构点云的几何坐标的比特深度。gps_geometry_3d_coordinates_bitdepth_minus1应该在 0 到 31 的范围内,包括端值。

[0194] 所述重构点云的边界框可以被分成一组子立方体。每个子立方体由一个索引来索引。该索引的解码结果给出了每个子立方体(3D PCM框)的3D位置。

[0195] 所述重构点云的边界框的大小可以被给定为所述重构点云参数的3D坐标的比特深度(gps_geometry_3d_coordinates_bitdepth_minus1)。

[0196] 所述3D PCM框的大小可由几何视频参数的比特深度(gps_geometry_nominal_2d_bitdepth_minus1)给出。所述3D PCM框的边等于 $2^{\text{gps_geometry_nominal_bitdepth_minus1}+1}$ 。

[0197] 因此,gps_geometry_3d_coordinates_bitdepth_minus1和gps_geometry_nominal_2d_bitdepth_minus1之间的差指示一个3D轴中的3D位置数量索引。

[0198] $n = \text{gps_geometry_3d_coordinates_bitdepth_minus1} - \text{gps_geometry_nominal_2d_bitdepth_minus1};$

[0199] $\text{pdu_3d_pcm_shift_normal_axis} = (\text{dpdu_3d_relative_shift_index} \& 2n - 1)$

[0200] $* 2^{\text{gps_geometry_nominal_bitdepth_minus1} + 1};$

[0201] $\text{pdu_3d_pcm_shift_tangent_axis} = (\text{dpdu_3d_relative_shift_index} \& (2n - 1)) \ll (2n - 1))$

[0202] $* 2^{\text{gps_geometry_nominal_bitdepth_minus1} + 1}$

[0203] $\text{pdu_3d_pcm_shift_bitangent_axis} = (\text{dpdu_3d_relative_shift_index} \& 2 * (2n - 1)) \ll (2n - 1))$

[0204] $* 2^{\text{gps_geometry_nominal_bitdepth_minus1} + 1}$

[0205] 示例：

[0206] $\text{gps_geometry_3d_coordinates_bitdepth_minus1} = 14;$

[0207] $2^{\text{gps_geometry_nominal_bitdepth_minus1}} = 10;$

[0208] $\text{pdu_3d_pcm_shift_normal_axis} = (\text{index} \& 0x000f) * 1024;$

[0209] $\text{pdu_3d_pcm_shift_tangent_axis} = (\text{index} \& 0x00f0) * 1024;$

[0210] $\text{pdu_3d_pcm_shift_bitangent_axis} = (\text{index} \& 0x0f00) * 1024;$

[0211] 在该示例中,每个3D坐标用4比特来编码。

[0212] 等于0x0123的索引值给出以下结果：

[0213] $\text{pdu_3d_pcm_shift_normal_axis} = 3 * 1024 = 3072;$

[0214] $\text{pdu_3d_pcm_shift_tangent_axis} = 2 * 1024 = 2048;$

[0215] $\text{pdu_3d_pcm_shift_bitangent_axis} = 1 * 1024 = 1024;$

[0216] 根据实施例,点云的点从PCM编码的面片而被重构如下。

[0217] 该过程的输入是：

[0218] gFrame,所解码的几何帧结构

[0219] aFrame,所解码的属性帧

[0220] recPc,用于保存所述重构点云中的点的列表的容器。

[0221] pointsAdded,一种变量,其指示已添加到所述重构点云中的点的列表的点的数量。

[0222] patch,面片,从其检索点并将该点添加到重构点云中的点的列表(该过程假设输入面片是PCM编码的面片)。

[0223] 该过程的输出是在recPC中无损重构的点云。

- [0224] 首先,如下检索面片信息:
- [0225] pdu_2d_shift_u=patch.pdu_2d_shift_u
- [0226] pdu_2d_shift_v=patch.pdu_2d_shift_v
- [0227] Patch2dSizeU=patch.pdu_3d_shift_tangent_axis-patch.pdu_2d_shift_u
- [0228] Patch2dSizeV=patch.pdu_3d_shift_bitangent_axis-patch.pdu_2d_shift_v
- [0229] occupancyResolution=patch.occupancy_resolution
- [0230] 如果lossless_geometry_444标记等于0,则使用如TMC2的条款3.3.1.1中所指定的4:4:4格式从存储在几何帧中的PCM编码的面片中检索点。
- [0231] 从存储在4:4:4几何帧中的PCM编码的面片中检索点
- [0232] 从PCM编码的面片中检索点,并将该点如下添加到所述重构点云中:

3d_pcm_shift [0] = pdu_3d_pcm_shift_tangent_axis

3d_pcm_shift [1] = pdu_3d_pcm_shift_bitangent_axis

3d_pcm_shift [2] = pdu_3d_pcm_shift_normal_axis

```
for( x = Patch2dShiftU ; x < Patch2dShiftU + Patch2dSizeU; x++ )
```

- [0233]
- ```
for(y = Patch2dShiftV ; y < Patch2dShiftV + Patch2dSizeV; y++)
 if(oFrame[y][x] != 0) {
 for(i = 0; i < 3; i++)
 point[i] = gFrame[0][i][y][x] + 3d_pcm_shift [i]
 }
}
```

[0234] 图11示出了其中实现了各个方面和实施例的系统的示例的框图。系统11000可以被实现为包括以下描述的各种组件的设备,并且被配置为执行本申请中描述的一个或多个方面。此类设备的示例包括但不限于各种电子设备,诸如个人计算机、膝上型计算机、智能电话、平板计算机、数字多媒体机顶盒、数字电视接收机、个人视频记录系统、连接的家用电器和服务器。系统11000的元件可以单独地或组合地被实现在单个集成电路、多个IC和/或分立组件中。例如,在至少一个实施例中,系统11000的处理和编码器/解码器元件分布在多个IC和/或分立组件上。在各种实施例中,所述系统11000经由例如通信总线或通过专用输入和/或输出端口而被通信地耦合到其他类似系统或其他电子设备。在各种实施例中,所述系统11000被配置为实现本文中描述的一个或多个方面。

[0235] 所述系统11000包括至少一个处理器11010,其被配置为执行加载在其中的指令,以用于实现例如本文中描述的各个方面。处理器11010可以包括嵌入式存储器、输入输出接口和本领域已知的各种其它电路。所述系统11000包括至少一个存储器11020(例如,易失性存储器设备和/或非易失性存储器设备)。系统11000包括存储设备11040,其可以包括非易失性存储器和/或易失性存储器,这其中包括但不限于EEPROM、ROM、PROM、RAM、DRAM、SRAM、闪存、磁盘驱动器和/或光盘驱动器。作为非限制性示例,所述存储设备11040可以包括内部存储设备、附接的存储设备和/或网络可访问的存储设备。

[0236] 系统11000包括编码器/解码器模块11030,其被配置为例如处理数据以提供所编码的视频或所解码的视频,并且所述编码器/解码器模块11030可以包括其自己的处理器和存储器。所述编码器/解码器模块11030表示可包括在设备中以执行编码和/或解码功能的

模块(一个或多个)。如已知的,设备可以包括所述编码模块和解码模块中的一个或两个。另外,编码器/解码器模块11030可实施为系统11000的单独元件或可并入处理器11010内作为如所属领域的技术人员已知的硬件与软件的组合。

[0237] 要加载到处理器11010或编码器/解码器11030上以执行本文档中描述的各个方面的程序代码可以存储在存储设备11040中,并且随后加载到存储器11020上以供处理器11010执行。根据各种实施例,处理器11010、存储器11020、存储设备11040和编码器/解码器模块11030中的一者或者可以在执行本文中描述的过程期间存储各种项中的一者或者。这些存储的项可以包括但不限于输入视频、所解码的视频或该解码的视频的部分、比特流、矩阵、变量以及来自方程式、公式、运算和运算逻辑的处理的中间或最终结果。

[0238] 在若干实施例中,所述处理器11010和/或所述编码器/解码器模块11030内的存储器用于存储指令,并且提供用于在编码或解码期间需要的处理的工作存储器。

[0239] 然而,在其它实施例中,所述处理设备(例如,所述处理设备可为所述处理器11010或所述编码器/解码器模块11030)外部的存储器用于这些功能中的一者或者。外部存储器可以是存储器11020和/或存储设备11040,例如,动态易失性存储器和/或非易失性闪存。在若干实施例中,外部非易失性闪存用于存储电视的操作系统。在至少一个实施例中,诸如RAM的快速外部动态易失性存储器被用作视频编码和解码操作的工作存储器,诸如用于MPEG-2、HEVC、VVC(通用视频编码)或TMC2的工作存储器。

[0240] 如框11130中所示,可以通过各种输入设备来提供对系统11000的元件的输入。这样的输入设备包括但不限于:(i)接收例如由广播者通过空中传输的RF信号的RF部分,(ii)复合输入端子,(iii)USB输入端子,和/或(iv)HDMI输入端子。

[0241] 在各种实施例中,框11130的输入设备具有本领域已知的相关联的相应输入处理元件。例如,所述RF部分可以与用于以下所必须的元件相关联:(i)选择期望频率(也称为选择信号,或将信号频带限制到一频带),(ii)将所选择的信号下变频,(iii)再次将频带限制到较窄频带,以选择(例如,)在某些实施例中可以称为信道的信号频带,(iv)解调所述下变频且频带限制的信号,(v)执行纠错,和(vi)解复用以选择期望的数据分组流。各种实施例的RF部分包括一个或多个元件以执行这些功能,例如,频率选择器、信号选择器、限带器、信道选择器、滤波器、下变频器、解调器、纠错器和解复用器。所述RF部分可以包括执行各种这些功能的调谐器,这些功能包括例如将所接收的信号下变频到较低频率(例如,中频或近基带频率)或基带。

[0242] 在一个机顶盒实施例中,所述RF部分及其相关的输入处理元件接收通过有线(例如,电缆)介质发送的RF信号,并通过滤波、下变频和再次滤波来执行到期望频带的频率选择。

[0243] 各种实施例重新安排上述(和其它)元件的顺序,移除这些元件中的一些,和/或添加执行类似或不同功能的其它元件。

[0244] 添加元件可以包括在现有元件之间插入元件,例如插入放大器和模数转换器。在各种实施例中,所述RF部分包括天线。

[0245] 另外,USB和/或HDMI终端可以包括用于通过USB和/或HDMI连接将系统11000连接到其它电子设备的相应接口处理器。应当理解,输入处理的各个方面(例如,所罗门纠错)可以根据需要在例如单独的输入处理IC或处理器11010内实现。类似地,USB或HDMI接口处理

的各方面可以根据需要在单独的接口IC内或在处理器11010内实现。解调、纠错和解复用的流被提供给各种处理元件,这其中包括例如处理器11010和编码器/解码器11030,其与存储器和存储元件结合操作以根据需要来处理所述数据流以便在输出设备上呈现。

[0246] 系统11000的各种元件可以设置在集成壳体内。在该集成壳体内,各种元件可以使用合适的连接布置(例如,本领域已知的内部总线,包括I2C总线、布线和印刷电路板)互连并在其间传输数据。

[0247] 所述系统11000包括通信接口11050,其使得能够经由通信信道11060与其他设备通信。所述通信接口11050可以包括但不限于被配置为通过通信信道11060发送和接收数据的收发器。所述通信接口11050可以包括但不限于调制解调器或网卡,并且所述通信信道11060可以例如在有线和/或无线介质内实现。

[0248] 在各种实施例中,使用Wi-Fi网络(例如IEEE 802.11),将数据流式传输给所述系统11000。这些实施例的Wi-Fi信号通过适用于Wi-Fi通信的通信信道11060和通信接口11050来接收。这些实施例的通信信道11060通常连接到接入点或路由器,所述接入点或路由器提供对包括因特网的外部网络的接入以允许流式传输应用和其它云上通信。

[0249] 其它实施例使用机顶盒向系统11000提供流式传输数据,该机顶盒通过输入框11130的HDMI连接来传递所述数据。

[0250] 还有一些实施例使用输入框11130的RF连接来向系统11000提供流式传输数据。

[0251] 所述系统11000可以向各种输出设备(包括显示器11100、扬声器11110和其他外围设备11120)提供输出信号。在各实施例的各示例中,所述其它外围设备11120包括以下中的一者或者者:独立DVR、盘播放器、立体声系统、照明系统和基于系统11000的输出来提供功能的其他设备。

[0252] 在各种实施例中,使用信令(诸如,AV.Link(AV.链路)、CEC、或在有或没有用户干预的情况下实现设备到设备控制的其他通信协议)在系统11000和显示器11100、扬声器11110或其它外围设备11120之间传送控制信号。

[0253] 所述输出设备可以经由通过相应接口11070、11080和11090的专用连接而通信地耦合到系统11000。

[0254] 作为替代,所述输出设备可以使用通信信道11060经由通信接口11050连接到系统11000。所述显示器11100和扬声器11110可以与系统11000的其它组件一起集成在电子设备(例如,电视机)中的单个单元中。

[0255] 在各种实施例中,显示接口11070包括显示驱动器,例如定时控制器((T Con)芯片。

[0256] 例如,如果输入11130的RF部分是单独机顶盒的一部分,则所述显示器11100和扬声器11110可以备选地与其它组件中的一个或多个分离。在所述显示器11100和扬声器11110是外部组件的各种实施例中,所述输出信号可以经由专用输出连接来提供,所述专用输出连接例如包括HDMI端口、USB端口或COMP输出。

[0257] 本文所述的各种过程和特征的实施方式可以体现在各种不同的设备或应用中。这种设备的示例包括编码器、解码器、处理来自解码器的输出的后处理器、向编码器提供输入的预处理器、视频编码器、视频解码器、视频编解码器、网络服务器、机顶盒、膝上型计算机、个人计算机、蜂窝电话、PDA、以及用于处理图片或视频的任何其他设备或其他通信设备。应

当清楚,该设备可以是移动的,甚至可以安装在移动车辆中。

[0258] 另外,所述方法可由正被处理器执行的指令实施,且此类指令(和/或由实施方案产生的数据值)可存储在计算机可读存储介质上。计算机可读存储介质可以采取计算机可读程序产品形式,该计算机可读程序产品被嵌入在一个或多个计算机可读介质中,并且其上嵌入有计算机可执行的计算机可读程序代码。如本文所使用的计算机可读存储介质被认为是非暂时性存储介质,其被赋予在其中存储信息的固有能力以及提供从其检索信息的固有能力。计算机可读存储介质可以是例如但不限于:电、磁、光、电磁、红外或半导体系统、装置或设备,或前述的任何合适的组合。应当理解,尽管提供了可以应用本实施例的计算机可读存储介质的更具体的示例,但是如本领域普通技术人员容易理解的,以下仅是说明性的而非穷举性的列举:便携式计算机磁盘;硬盘;只读存储器(ROM);可擦除可编程只读存储器(EPROM或闪存);便携式光盘只读存储器(CD-ROM);光学存储设备;磁存储装置;或前述的任何合适的组合。

[0259] 所述指令可以形成有形地体现在处理器可读介质上的应用程序。

[0260] 指令可以是例如硬件、固件、软件或其组合。指令可以在例如操作系统、单独的应用程序或两者的组合中找到。因此,处理器可以被表征为例如被配置为执行处理的设备和包括具有用于执行处理的指令的处理器可读介质(诸如存储设备)的设备这两者。此外,处理器可读介质可以存储由实现方式产生的数据值,作为指令的补充或替代。

[0261] 如对于本领域技术人员将显而易见的,实现方式可以产生被格式化以携带例如可以被存储或发送的信息的各种信号。该信息可以包括例如用于执行方法的指令,或者由所描述的实现方式之一产生的数据。例如,信号可以被格式化为将用于写入或读取本实施例的所述示例的语法的规则作为数据来携带,或者将由本实施例的所述示例写入的实际语法值作为数据来携带。这种信号可以被格式化为例如电磁波(例如,使用频谱的射频部分)或基带信号。所述格式化可以包括例如对数据流进行编码并且利用所编码的数据流对载波进行调制。所述信号携带的信息可以是例如模拟或数字信息。如已知的,所述信号可以通过各种不同的有线或无线链路来传输。所述信号可以存储在处理器可读介质上。

[0262] 已经描述了许多实现方式。然而,应当理解,可以进行各种修改。例如,不同实现方式的元素可以被组合、补充、修改或移除以产生其他实现方式。另外,本领域技术人员将理解,其他结构和过程可以替代所公开的那些,并且所得到的实现方式将以至少基本上相同的方式(一个或多个)执行至少基本上相同的功能(一个或多个),以实现与所公开的实现方式至少基本上相同的结果(一个或多个)。因此,本申请考虑了这些和其他实现方式。

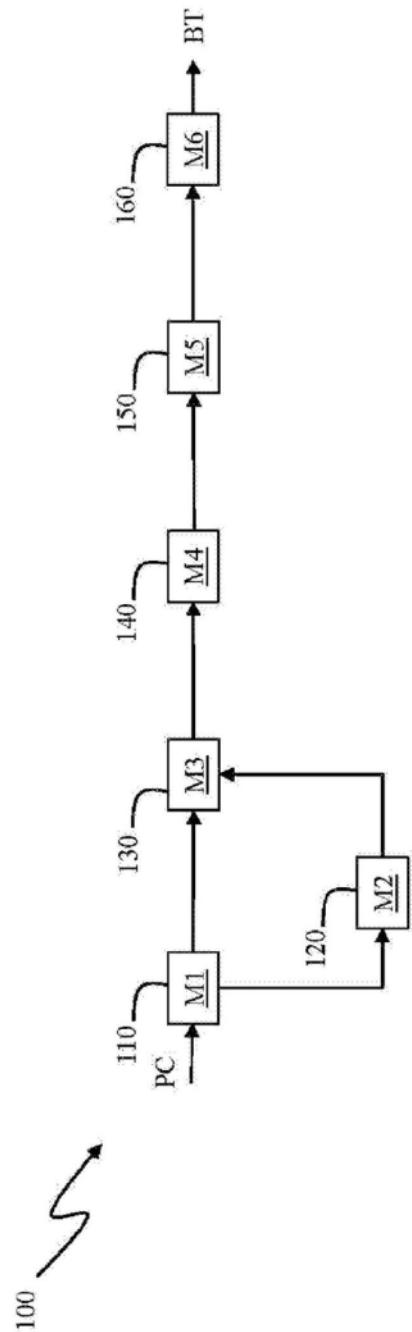


图1

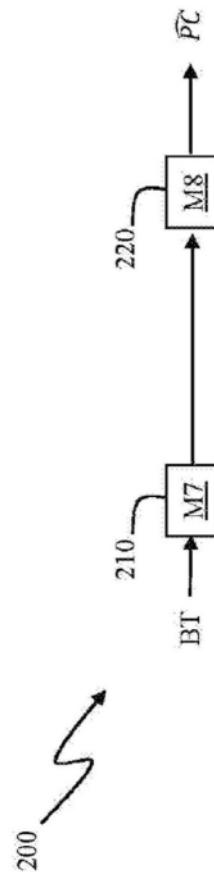


图2



图3



图4

| group of frames auxiliary information( ) { | 描述符   |
|--------------------------------------------|-------|
| patch count                                | u(32) |
| mps patch count                            | u(32) |
| occupancy precision                        | u(8)  |
| max candidate Count                        | u(8)  |
| bit count u0                               | u(8)  |
| bit count v0                               | u(8)  |
| bit count u1                               | u(8)  |
| bit count v1                               | u(8)  |
| bit count d1                               | u(8)  |
| occupancy aux stream size                  | u(32) |
| ByteCount += 15                            |       |
| for( i = 0; i < mps_patch_count; i++ ) {   |       |
| num_of_mps[i]                              | u(32) |
| }                                          |       |
| for( i = 0; i < patch_count; i++ ) {       |       |
| patchList[i].patch_u0                      | ae(v) |
| patchList[i].patch_v0                      | ae(v) |
| patchList[i].patch_u1                      | ae(v) |
| patchList[i].patch_v1                      | ae(v) |
| patchList[i].patch_d1                      | ae(v) |
| patchList[i].delta_size_u0                 | se(v) |
| patchList[i].delta_size_v0                 | se(v) |
| }                                          |       |
| }                                          |       |
| ...                                        |       |

图5

|                                          | 描述符   |
|------------------------------------------|-------|
| patch_count                              | u(32) |
| mps_patch_count                          | u(32) |
| occupancy_precision                      | u(8)  |
| max_candidate_Count                      | u(8)  |
| bit_count_u0                             | u(8)  |
| bit_count_v0                             | u(8)  |
| bit_count_u1                             | u(8)  |
| bit_count_v1                             | u(8)  |
| bit_count_d1                             | u(8)  |
| bit_count_mps_u1                         | u(8)  |
| bit_count_mps_v1                         | u(8)  |
| bit_count_mps_d1                         | u(8)  |
| mps_resolution                           | u(8)  |
| occupancy_aux_stream_size                | u(32) |
| ByteCount += 15                          |       |
| for( i = 0; i < patch_count; i++ ) {     |       |
| patchList[i].patch_u0                    | ae(v) |
| patchList[i].patch_v0                    | ae(v) |
| patchList[i].patch_u1                    | ae(v) |
| patchList[i].patch_v1                    | ae(v) |
| patchList[i].patch_d1                    | ae(v) |
| patchList[i].delta_size_u0               | se(v) |
| patchList[i].delta_size_v0               | se(v) |
| }                                        |       |
| }                                        |       |
| for( i = 0; i < mps_patch_count; i++ ) { |       |
| mpsPatchList[i].patch_u0                 | ae(v) |
| mpsPatchList[i].patch_v0                 | ae(v) |
| mpsPatchList[i].patch_u1                 | ae(v) |
| mpsPatchList[i].patch_v1                 | ae(v) |
| mpsPatchList[i].patch_d1                 | ae(v) |
| mpsPatchList[i].delta_size_u0            | se(v) |
| mpsPatchList[i].delta_size_v0            | se(v) |
| mpsPatchList[i].num_of_mps               | se(v) |
| }                                        |       |
| ...                                      |       |

图6

| sequence parameter set( ) {                      | 描述符  |
|--------------------------------------------------|------|
| ...                                              |      |
| <b>sps_pcm_patch_enabled_flag</b>                | u(1) |
| if( sps_pcm_patch_enabled_flag )                 |      |
| <b>sps_pcm_patch_separate_video_present_flag</b> | u(1) |
| <b>sps_pcm_relative_3d_offset_flag</b>           | u(1) |
| occupancy parameter set( )                       |      |
| ...                                              |      |
| }                                                |      |

图7

| patch frame data unit( frmIdx ) {                                            | 描述符   |
|------------------------------------------------------------------------------|-------|
| <b>pfdu_patch_count_minus1</b>                                               | u(32) |
| if( pfh_type[ frmIdx ] != 1 )                                                |       |
| <b>pfdu_inter_patch_count</b>                                                | ae(v) |
| else                                                                         |       |
| pfdu_inter_patch_count = 0                                                   |       |
| for( p = 0; p < pfdu_inter_patch_count; p++ ) {                              |       |
| <b>pfdu_patch_mode[ frmIdx ][ p ]</b> = P_INTER                              |       |
| patch information data( frmIdx, p, pfdu_patch_mode[ frmIdx ][ p ] )          |       |
| }                                                                            |       |
| for( p = pfdu_inter_patch_count; p <= pfdu_patch_count_minus1; p++ ) {       |       |
| pfdu_patch_mode[ frmIdx ][ p ] = pfh_type[ frmIdx ] == 1 ? I_INTRA : P_INTRA |       |
| patch information data( frmIdx, p, pfdu_patch_mode[ frmIdx ][ p ] )          |       |
| }                                                                            |       |
| PfduTotalNumberOfPatches[ frmIdx ] = p                                       |       |
| if( sps_pcm_patch_enabled_flag ) {                                           |       |
| <b>pfdu_pcm_patch_count_minus1</b>                                           | u(32) |
| for( p = pfdu_patch_count_minus1+1;                                          |       |
| p <= pfdu_patch_count_minus1 + pfdu_pcm_patch_count_minus1;                  |       |
| p++ ) {                                                                      |       |
| pfdu_patch_mode[ frmIdx ][ p ] = pfh_type[ frmIdx ] == 1 ? I_PCM : P_PCM     |       |
| patch information data( frmIdx, p, pfdu_patch_mode[ frmIdx ][ p ] )          |       |
| PfduTotalNumberOfPatches[ frmIdx ]++                                         |       |
| }                                                                            |       |
| if( sps_point_local_reconstruction_enabled_flag )                            |       |
| point_local_reconstruction()                                                 |       |
| byte_alignment()                                                             |       |
| }                                                                            |       |

图8

| pcm patch data unit( frmIdx, patchIndex ) {        | 描述符   |
|----------------------------------------------------|-------|
| if(sps_pcm_separate_video_flag )                   |       |
| <b>ppdu_patch_in_pcm_video_flag[ frmIdx ][ p ]</b> | ae(v) |
| <b>ppdu_2d_shift_u[ frmIdx ][ p ]</b>              | ae(v) |
| <b>ppdu_2d_shift_v[ frmIdx ][ p ]</b>              | ae(v) |
| <b>ppdu_2d_delta_size_u[ frmIdx ][ p ]</b>         | ae(v) |
| <b>ppdu_2d_delta_size_v[ frmIdx ][ p ]</b>         | ae(v) |
| <b>ppdu_pcm_points[ frmIdx ][ p ]</b>              | ae(v) |
| }                                                  |       |

图9

| 描述符                                                             |
|-----------------------------------------------------------------|
| if(sps pcm separate video flag )                                |
| <b>ppdu patch in pcm video flag[ frmIdx ][ patchIndex ]</b>     |
| ae(v)                                                           |
| if(!ppdu patch in pcm video flag[ frmIdx ][ patchIndex ]) {     |
| <b>ppdu 2d shift u[ frmIdx ][ patchIndex ]</b>                  |
| ae(v)                                                           |
| <b>ppdu 2d shift v[ frmIdx ][ patchIndex ]</b>                  |
| ae(v)                                                           |
| <b>ppdu 2d delta size u[ frmIdx ][ p ]</b>                      |
| bpd                                                             |
| <b>ppdu 2d delta size v[ frmIdx ][ p ]</b>                      |
| }                                                               |
| if(sps pcm relative 3D offset flag){                            |
| <b>dpdu 3d relative shift index[ frmIdx ][ patchIndex ]</b>     |
| ae(v)                                                           |
| }                                                               |
| else {                                                          |
| <b>dpdu 3d pem shift tangent axis[ frmIdx ][ patchIndex ]</b>   |
| ae(v)                                                           |
| <b>dpdu 3d pem shift bitangent axis[ frmIdx ][ patchIndex ]</b> |
| ae(v)                                                           |
| <b>dpdu 3d pem shift normal axis[ frmIdx ][ patchIndex ]</b>    |
| ae(v)                                                           |
| }                                                               |
| <b>ppdu pem points[ frmIdx ][ patchIndex ]</b>                  |
| ae(v)                                                           |
| }                                                               |

图10

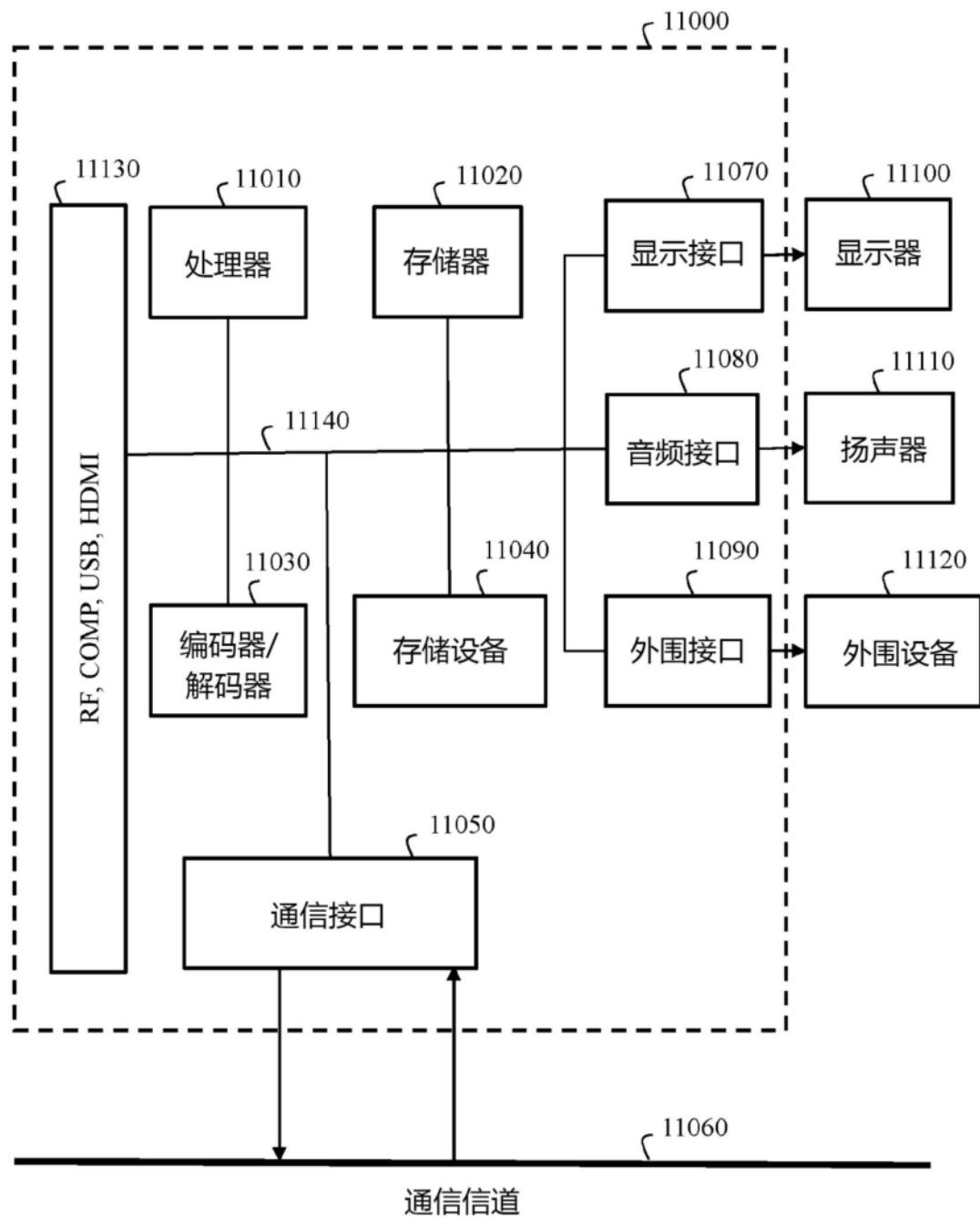


图11