



(21) 申请号 201980046431.2

(22) 申请日 2019.06.28

(65) 同一申请的已公布的文献号
申请公布号 CN 112673398 A

(43) 申请公布日 2021.04.16

(30) 优先权数据
18305935.1 2018.07.11 EP
19305057.2 2019.01.17 EP

(85) PCT国际申请进入国家阶段日
2021.01.11

(86) PCT国际申请的申请数据
PCT/US2019/039784 2019.06.28

(87) PCT国际申请的公布数据
W02020/014011 EN 2020.01.16

(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 J.里卡德 C.盖德 J-C.切维特
D.金德隆 Y.奥利维尔 J.拉奇

(74) 专利代理机构 北京市柳沈律师事务所
11105
专利代理师 张贵东

(51) Int.Cl.
G06T 9/00 (2006.01)

(56) 对比文件
Celine Guede et al..PCC CE2.5 Report
on optimizations for DepthImage1 geometry
information.MPEG MEETING.2018,图18-19,第
1-3节,结果,附录C.

Lanyi He et al..Best-Effort
Projection Based Attribute Compression
for 3D Point Cloud.IEEE.2017,第1-6页.

审查员 王婷

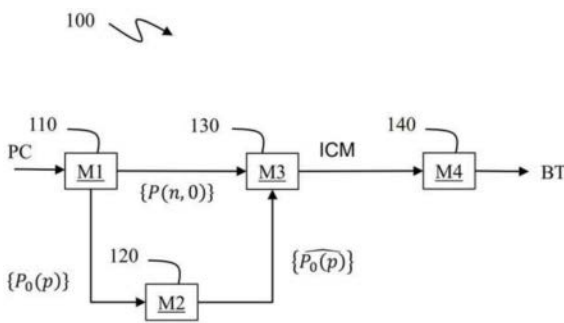
权利要求书2页 说明书17页 附图9页

(54) 发明名称

一种用于编码/解码点云的几何结构的方法和
设备

(57) 摘要

本实施例涉及一种方法和设备。该方法包括:通过将点云的至少一个点投影到投影平面上来从点云的所述点获得至少一个第一点,并且获得根据所述至少一个第一点确定的点云的至少一个其他点;基于从所述至少一个第一点获得的至少一个重构点和至少一个内插点为所述至少一个第一点确定并编码至少一个内插编译码模式,所述至少一个内插点由所述至少一个内插编译码模式所定义以逼近点云的所述至少一个其他点;以及信令通知所述至少一个内插编译码模式作为图像数据的值。



1. 一种方法, 包括:

通过将点云的3D点投影到投影平面上来获得第一2D点;

根据第一2D点获得至少一个其他2D点, 其中所述至少一个其他2D点属于第一2D点周围的邻域;

基于从第一2D点获得的重构2D点以及根据至少一个2D内插点为第一2D点确定内插编译码模式, 所述至少一个2D内插点由所述内插编译码模式所定义以便逼近所述至少一个其他2D点, 所述内插编译码模式定义至少一个重构2D点周围的邻域和最小深度值, 深度值与根据所述最小深度值和在该所述重构2D点的深度和所述邻域中至少一个点的深度之间确定的最大深度差的所述至少一个2D内插点相关联, 以及

对所述内插编译码模式进行编码, 其中所述内插编译码模式针对所述点云的每个块或分片被信令通知。

2. 根据权利要求1所述的方法, 其中, 所述方法还包括将第一2D点的深度值分配给第一深度图像的像素值, 通过对第一深度图像进行编码来获得编码后的第一深度图像, 以及通过对编码后的第一深度图像进行解码来获得解码后的第一深度图像, 所述重构2D点是解码后的第一深度图像的像素。

3. 根据权利要求1所述的方法, 其中, 确定所述内插编译码模式基于与至少一个候选内插编译码模式相关联的率失真成本, 所述候选内插编译码模式指示通过根据候选内插编译码模式定义的至少一个2D内插点对所述至少一个其他2D点的特定编码。

4. 根据权利要求1所述的方法, 其中, 所述邻域是第一2D点周围的图像区域。

5. 一种方法, 包括:

获得被编码在比特流中的点云的至少一个重构点;

从所述比特流获得至少一个内插编译码模式, 所述至少一个内插编译码模式定义所述至少一个重构点周围的邻域和最小深度值; 以及

根据所述内插编译码模式为所述至少一个重构点创建至少一个内插点, 深度值与根据所述最小深度值和在该所述至少一个重构点的解码后的深度和所述邻域中至少一个点的深度之间确定的最大深度差的所述至少一个内插点中的每一个内插点相关联, 其中所述至少一个内插编译码模式在比特流中按所述点云的每个块或分片被信令通知。

6. 根据权利要求5所述的方法, 其中, 与所述至少一个内插点相关联的深度值在从所述重构点的深度值到最大深度值的范围内。

7. 根据权利要求5所述的方法, 其中, 获得所述至少一个重构点包括:

从所述比特流解码编码后的第一深度图像, 所述至少一个重构点对应于解码后的深度图像的至少一个像素, 所述编码后的第一深度图像是通过将点云的3D点投影到投影平面上而获得的第一2D点的深度值分配给第一深度图像的像素值并且对第一深度图像进行编码而获得的。

8. 根据权利要求1或5所述的方法, 其中, 所述内插编译码模式包括表示所述邻域的元素的数量数据。

9. 一种包括至少一个处理器的设备, 所述至少一个处理器被配置为:

通过将点云的3D点投影到投影平面上来获得第一2D点;

根据第一2D点获得至少一个其他2D点, 其中所述至少一个其他2D点属于第一2D点周围

的邻域；

基于从第一2D点获得的重构2D点以及根据至少一个2D内插点为第一2D点确定内插编译码模式,所述至少一个2D内插点由所述内插编译码模式所定义以便逼近所述至少一个其他2D点,所述内插编译码模式定义所述重构2D点周围的邻域和最小深度值,深度值与根据所述最小深度值和所述重构2D点的深度和所述邻域中至少一个点的深度之间确定的最大深度差的所述至少一个2D内插点相关联,以及

对所述内插编译码模式进行编码,其中所述内插编译码模式针对所述点云的每个块或分片被信令通知。

10. 一种包括至少一个处理器的设备,所述至少一个处理器被配置为:

获得被编码在比特流中的点云的至少一个重构点;

从所述比特流获得至少一个内插编译码模式,所述至少一个内插编译码模式定义所述至少一个重构点周围的邻域和最小深度值;以及

根据所述内插编译码模式为所述至少一个重构点创建至少一个内插点,深度值与根据所述最小深度值和所述至少一个重构点的解码后的深度和所述邻域中至少一个点的深度之间确定的最大深度差的所述至少一个内插点中的每一个内插点相关联,其中所述至少一个内插编译码模式在比特流中按所述点云的每个块或分片被信令通知。

11. 根据权利要求9或10所述的设备,其中,所述内插编译码模式包括表示所述邻域的元素的数量数据。

12. 根据权利要求9或10所述的设备,其中,至少两个内插编译码模式由最小深度值的不同值确定。

13. 根据权利要求10所述的设备,其中,所述至少一个处理器还被配置成从所述比特流解码编码后的第一深度图像,所述至少一个重构点对应于所述解码后的深度图像的至少一个像素,所述编码后的第一深度图像是通过将点云的3D点投影到投影平面上而获得的第一2D点的深度值分配给第一深度图像的像素值并且对第一深度图像进行编码而获得的,所述邻域是至少一个重构点的邻域。

14. 根据权利要求10所述的设备,其中,与所述至少一个内插点相关联的深度值在从所述重构点的深度值到最大深度值的范围内。

15. 一种非暂时性计算机可读介质,包括用于使得一个或多个处理器执行权利要求1的方法的指令。

16. 一种非暂时性计算机可读介质,包括用于使得一个或多个处理器执行权利要求5的方法的指令。

17. 根据权利要求15或16所述的非暂时性计算机可读介质,其中,所述内插编译码模式包括表示所述邻域的元素的数量数据。

一种用于编码/解码点云的几何结构的方法和设备

技术领域

[0001] 本实施例总体上涉及对表示3D对象外表面的点云的编码和解码。特别地,但不排他地,本实施例的技术领域涉及对这种点云的几何结构的编码/解码。

背景技术

[0002] 本部分旨在向读者介绍本领域的各个方面,这些方面可能与下面描述和/或要求保护的本实施例的各个方面相关。该讨论被认为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。相应地,应该理解的是,这些陈述应该从这个角度来理解,而不是作为对现有技术的承认。

[0003] 点云是某个坐标系中的数据点集合。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定位(3D空间中的X、Y和Z坐标)定义,也可以由其他相关属性(诸如颜色(例如,在RGB或YUV颜色空间中表示的颜色)、透明度、反射率、双分量法向量等)定义。

[0004] 通常将点云表示为6个分量点的集合(X,Y,Z,R,G,B)或等效地(X,Y,Z,Y,U,V),其中,(X,Y,Z)定义3D空间中彩色点的坐标,并且(R,G,B)或(Y,U,V)定义该彩色点的颜色。

[0005] 点云可以是静态的,也可以是动态的,这取决于云是否随时间而演变。应该注意的是,在动态点云的情况下,点的数量不是恒定的,而相反通常是随时间而演变的。因此,动态点云是按时间顺序排列的点集合的列表。

[0006] 实际上,点云可以用于各种目的(诸如文化遗产/建筑,其中,像雕像或建筑这样的对象被3D扫描)以便共享对象的空间配置,而无需发送或访问它。而且,这也是一种确保保存对象的知识以防其被破坏的方式;例如,地震中的寺庙。这种点云通常是静态的、彩色的和巨大的。

[0007] 另一用例是在地形学和地图学方面,其中使用3D表示,地图不限于平面并且可以包括地形。谷歌地图现在是3D地图的一个很好的示例,但是其使用网格而不是点云。然而,点云可能是3D地图的合适的数据格式,并且这种点云通常是静态的、着色的和巨大的。

[0008] 汽车工业和自主汽车也是可以使用点云的领域。自主汽车应该能够“探测”它们的环境,以基于它们临近地区的现实做出很好的驾驶决策。像LIDAR这样的典型传感器产生决策引擎所使用的动态点云。这些点云不旨在被人看到,并且它们通常很小,不一定是着色的,并且是动态的、捕获频率很高。它们可能具有其他属性,如LIDAR所提供的反射率,因为该属性是关于感测对象的材料的良好信息并且可能有助于决策。

[0009] 虚拟现实和沉浸式世界最近已经成为热门话题,并且被许多人预见为2D平面视频的未来。基本思想是让观众沉浸在他周围的环境中,而不是只能看着他面前的虚拟世界的标准TV。沉浸有几个层级,这取决于观众在环境中的自由度。着色点云是分布式虚拟现实世界(或VR)的一种很好的格式候选。它们可能是静态的或动态的,并且通常是平均尺寸,比如一次不超过数百万个点。

[0010] 仅当比特流的尺寸足够低以允许向末端用户进行实际的存储/传输时,点云压缩

才将成功地存储/传输沉浸式世界的3D对象。

[0011] 关键在于,能够以合理的比特率消耗向末端用户分发动态点云,同时保持可接受的(或最好是非常好的)体验质量。为了使沉浸式世界的分发链切实可行,对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性的结合而正在变得越来越流行。它们以两个主要步骤进行:首先,它们将点云(即,3D点)投影(正交投影)到2D图像上。例如,至少一个深度图像表示点云的几何结构,即,3D空间中的3D点的空间坐标,并且至少一个纹理图像表示与点云的3D点相关联的属性,例如,与那些3D点相关联的纹理/颜色信息。接下来,这些技术用传统视频编码器来编码这种深度和纹理图像。

[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很好的压缩性能,像例如HEVC("ITU-T H.265ITU的电信标准化部门(10/2014),系列H:audiovisual and multimedia systems,infrastructure of audiovisual services-coding of moving video,High efficiency video coding,Recommendation ITU-T H.265"),而在同时,它们通过使用简单的投影方案来保持低复杂性。

[0014] 基于图像的点云压缩技术的挑战之一在于,点云可能不适合于投影到图像上,尤其是当点分布遵循具有许多褶皱的表面(凹/凸区域,如在服装中)时,或者当点分布根本不遵循表面(如在毛发或头发中)时。在这些情况下,基于图像的点云压缩技术遭受很低的压缩效率(需要许多小的投影,这降低了2D视频压缩的效率)或很差的质量(由于难以将点云投影到表面上)。

[0015] 在现有技术中用于缓解该方法之一在于将多个几何结构和纹理信息投影到图像的同一空间定位(像素)上。这意味着可以为点云的每个3D点生成几个深度和/或纹理图像。

[0016] 例如,如在2017年10月中国澳门的ISO/IEC JTC1/SC29/WG11/N17248中定义的所谓的测试模型类别2点云编码器(Test Model Category 2point cloud encoder,TMC2)就是这种情况,其中点云被正交投影到投影平面上。然后,两个深度值与所述投影平面的每个坐标相关联:一个深度值表示与最近点关联的深度值(最小深度值),另一深度值表示最远点的深度值(最大深度值)。然后从最小深度值生成第一深度图像(D0),从最大深度值的绝对值生成第二深度图像(D1),其中D1-D0小于或等于最大表面厚度。

[0017] 然后深度图像和相关元数据被编码和解码。然后,点云的几何结构从两个解码后的深度图像来重构。但是,为了降低点云的全局编译码(code)成本,所述两个深度图像的编译码效率应该被提高。

发明内容

[0018] 以下内容呈现了本实施例的简化概述,以便提供对本实施例的某些方面的基本理解。该概述不是本实施例的广泛综述。它并不旨在标识本实施例的重要或关键元素。以下概述仅仅以简化的形式呈现了本实施例的某些方面,作为下面提供的更详细描述的前言。

[0019] 本实施例着手用一种方法来补救现有技术的至少一个缺点,该方法包括:

[0020] -通过将点云的至少一个点投影到投影平面上来从点云的所述点获得至少一个第一点,并且获得根据所述至少一个第一点确定的点云的至少一个其他点;

[0021] -基于从所述至少一个第一点获得的至少一个重构点和至少一个内插点为所述至少一个第一点确定并编码至少一个内插编译码模式,所述至少一个内插点由所述至少一个内插编译码模式所定义以逼近点云的所述至少一个其他点;以及

[0022] -信令通知所述至少一个内插编译码模式作为图像数据的值。

[0023] 从以下结合附图对示例的描述中,本实施例的具体性质以及本实施例的其他目的、优点、特征和用途将变得显而易见。

附图说明

[0024] 在附图中,示出了本实施例的示例。示出了:

[0025] 图1示意性示出了根据本实施例的示例的用于对点云的点的深度值进行编码的方法100的步骤图;

[0026] 图2示意性示出了根据本实施例中的实施例的步骤130的实施例的图;

[0027] 图3、图3a示出了根据本实施例的示例的候选内插编译码模式的实施例;

[0028] 图4示意性地示出了根据本实施例的示例的用于从比特流解码第一深度图像和第二深度图像的方法200的步骤图。

[0029] 图5示意性地示出了如现有技术(TMC2)中定义的用于对点云的几何结构和纹理进行编码的方法;

[0030] 图6示意性地示出了在图5的编码方法中使用方法100和200的示例;

[0031] 图6a示出了图6的变型;

[0032] 图7示意性地示出了如现有技术(TMC2)中定义的用于对点云的几何结构和纹理进行解码的方法;

[0033] 图8示意性地示出了在图7的解码方法中使用方法200的示例;

[0034] 图8a示出了图8的变型;

[0035] 图9示出了根据本实施例的示例来实施各个方面和实施例的系统的示例框图;

[0036] 图10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;

[0037] 图11示出了根据本实施例的示例的信号语法;

[0038] 图12示出了在TMC2中重构模式的使用的分布的示例;以及

[0039] 图13-图13b示出了占用图的示例。

[0040] 相似或相同的元素用相同的附图标记来引用。

具体实施方式

[0041] 下文将参照附图更全面地描述本实施例,在附图中示出了本实施例的示例。然而,本实施例可以以许多替代形式来体现,并且不应被解释为限于本文阐述的示例。相应地,尽管本实施例易于进行各种修改和替代形式,但是其具体示例在附图中以示例的方式示出,并将在本文详细描述。然而,应该理解的是,并不旨在将本实施例限制于所公开的特定形式,而相反的是,本说明书将覆盖落入如由权利要求限定的本实施例的精神和范围内的所有修改、等同物和替代。

[0042] 本文使用的术语仅用于描述具体示例的目的,并不旨在限制本实施例。如本文所使用的,单数形式“一”、“一个”和“该”也旨在包括复数形式,除非上下文另有明确指示。还

应该理解的是,当在本说明书中使用时,术语“包括”、“包含”、“含有”和/或“涵盖”指定所陈述的特征、整数、步骤、操作、元素和/或组件的存在,但不排除一个或多个其他特征、整数、步骤、操作、元素、组件和/或其组合的存在或添加。此外,当一个元素被称为“响应于”或“连接到”另一元素时,它可以直接响应于或连接到另一元素,或者可以存在中间元素。相反,当一个元素被称为“直接响应于”或“直接连接到”其他元素时,则不存在中间元素。如本文所使用的,术语“和/或”包括一个或多个相关列出项目的任何一个和所有组合,并且可以被缩写为“/”。

[0043] 应该理解的是,尽管术语第一、第二等可以在本文用于描述各种元素,但是这些元素不应该被这些术语所限制。这些术语仅用于区分一个元素与另一元素。例如,第一元素可以被称为第二元素,并且类似地,第二元素可以被称为第一元素,而不脱离本实施例的教导。

[0044] 尽管某些图包括通信路径上的箭头以示出通信的主要方向,但是应该理解的是,通信可以发生在与所示箭头相反的方向上。

[0045] 关于框图 and 操作流程图描述了某些示例,其中每个框表示电路元件、模块或代码部分,代码部分包括用于实施(多个)指定逻辑功能的一个或多个可执行指令。还应该注意,在其他实施方式中,在框中标注的(多个)功能可以不按所标注的顺序发生。例如,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,这取决于所涉及的功能。

[0046] 本文对“根据示例”或“在示例中”的引用意味着结合该示例来描述的特定特征、结构或特性可以被包括在本实施例的至少一个实施方式中。说明书中各个地方出现的短语“根据示例”或“在示例中”不一定都指同一示例,也不一定是与其他示例互斥的单独或替代示例。

[0047] 权利要求中出现的附图标记仅仅是为了说明,而对权利要求的范围没有限制作用。

[0048] 尽管没有明确描述,但是本示例和变型可以以任何组合或子组合来采用。

[0049] 本实施例被描述为用于对表示点云的两个深度图像和两个纹理图像进行编码/解码,但是扩展到对表示点云序列(时域动态点云)的两个深度图像序列(视频)和两个纹理图像序列(视频)进行编码/解码,因为点云序列的点云的几何结构(两个深度图像)和纹理(颜色)然后独立于点云序列的另一点云的几何结构(两个深度图像)和纹理(颜色)被编码/解码。

[0050] 如上所述,点云被正交投影到投影平面上,并且从与所述投影的3D点相关联的深度值获得两个深度图像D0和D1。D0是表示点云的最近点的深度值的第一深度图像,D1是表示点云的最远点的深度值的第二深度图像。第一深度图像D0使用例如传统图像/视频编码器来编码。

[0051] 在以下内容中,术语“图像区域”表示至少一个图像的像素集合。这些像素可以是或不是相邻的像素和/或属于或不属于同一图像,但是它们全部都共享至少一个共同属性。

[0052] 例如,整个图像本身可以被认为是图像区域。图像也可以被划分成多个块(block),然后块或一组块是图像区域。

[0053] 图像区域也可以具有非矩形形状。例如,当具有相同(或相似)提取特征的图像的

像素被关联以形成图像区域时,就是这种情况。图像区域可以是如TMC2中定义的分片(patch)。

[0054] 从图像中提取的特征的示例可以是颜色、纹理、法向量等……

[0055] 图1示意性地示出了根据本实施例的示例的用于对点云的点的深度值进行编码的方法100的步骤图。

[0056] 在步骤110中,模块M1通过将点云PC的至少一个点投影到投影平面上来从点云PC的所述至少一个点获得至少一个第一点 $P_0(p)$ 。

[0057] 例如,可以使用正交投影。

[0058] 对于每个第一点 $P_0(p)$,在步骤100中,模块M1还获得根据所述第一点 $P_0(p)$ 确定的点云PC的至少一个其他点 $P(n,0)$ 。点 $P(n,0)$ 表示N个点的集合中的点n,并且索引0是指第一点 $P_0(p)$ 。

[0059] 根据步骤110的实施例,点云PC的相对于第一点 $P_0(p)$ 的至少一个其他点 $P(n,0)$ 属于所述第一点 $P_0(p)$ 周围的邻域。

[0060] 根据实施例,相对于第一点 $P_0(p)$ 的邻域包括点云PC的靠近所述第一点 $P_0(p)$ 的点,例如,当这些点和第一点 $P_0(p)$ 之间的空间距离低于给定值时。

[0061] 根据实施例,邻域的点都与第一点 $P_0(p)$ 对齐。

[0062] 根据实施例,与第一点 $P_0(p)$ 相关联的邻域根据为所述第一点 $P_0(p)$ 确定的内插编译码模式ICM来定义。

[0063] 定义邻域意味着定义其元素的数量、空间距离等……

[0064] 根据实施例,与第一点 $P_0(p)$ 相关联的邻域是该第一点 $P_0(p)$ 周围的图像区域。

[0065] 在步骤120中,模块M2获得从所述至少一个第一点 $P_0(p)$ 获得的至少一个重构点 $\widehat{P_0(p)}$ 。

[0066] 根据步骤120的实施例,至少一个重构点 $\widehat{P_0(p)}$ 通过对第一点 $P_0(p)$ 进行编码和解码来获得。

[0067] 在步骤130中,模块M3基于从所述至少一个第一点 $P_0(p)$ 获得的至少一个重构点 $\widehat{P_0(p)}$ (步骤120)和至少一个内插点 $PI(i,0)$ 为所述至少一个第一点 $P_0(p)$ 确定至少一个内插编译码模式ICM,至少一个内插点 $PI(i,0)$ 由所述至少一个内插编译码模式ICM所定义以逼近(approximate)点云PC的所述至少一个其他点 $P(n,0)$ 。

[0068] 内插点 $PI(i,0)$ 表示I个内插点的集合中相对于第一点 $P_0(p)$ (索引0)的内插点i。

[0069] 在步骤140中,模块M4将所述至少一个第一点 $P_0(p)$ 和所述至少一个内插编译码模式ICM编码在比特流BT中。

[0070] 根据步骤120或140的实施例,所述至少一个第一点 $P_0(p)$ 的深度值被分配给第一深度图像D0,该第一深度图像D0由传统图像编码器编码在比特流BT中,诸如HEVC编码器/解码器或由TMC2指定的编码器/解码器。

[0071] 根据该实施例,在步骤120中,解码后的第一深度图像 $\widehat{D_0}$ 通过解码第一深度图像D0来获得,并且所述至少一个重构点 $\widehat{P_0(p)}$ 是解码后的第一深度图像 $\widehat{D_0}$ 的像素。

[0072] 根据步骤130的实施例,为第一点 $P_0(p)$ 确定内插编译码模式ICM是基于与至少一

个候选内插编译码模式CICM_k相关联的率失真成本RDC_k,候选内插编译码模式CICM_k指示通过根据候选内插编译码模式CICM_k定义的至少一个内插点PI_k(i,0) (i=1,⋯,I_k)对与第一点P0(p)相关联的至少一个其他点P(n,0) (n=1,⋯,N)的特定编码。

[0073] 内插点PI_k(i,0)表示I_k个内插点的集合中相对于第一点P0(p) (索引0)的内插点i, I_k是根据候选内插编译码模式CICM_k定义的整数值。

[0074] 根据步骤130的实施例,率失真成本RDC_k由拉格朗日方程给出:

$$[0075] \quad RDC_k = D_k + \lambda_k \cdot R_k$$

[0076] 其中,D_k是相对于候选内插编译码模式CICM_k的度量,R_k是用于对候选内插编译码模式CICM_k进行编码的比特率, λ_k 是相对于所述候选内插编译码模式CICM_k的拉格朗日乘数。

[0077] 关于拉格朗日乘数 λ_k ,该乘数的高值严重惩罚比特率R_k并且导致编译码的低质量,而该乘数的低值轻易地允许比特率R_k的高值并且导致编译码的质量高。该乘数可以与用于图像编译码的量化参数(quantization parameter,QP)相关。该乘数的典型值在从几百(对于非常差的编译码)到十分之一(tenth of unity) (对于很好的编译码)的范围内。这些值是指示性的,并且也可以取决于视频内容。

[0078] 根据步骤130的实施例,内插编译码模式ICM可以是与最低率失真成本RDC_k相关联的候选内插编译码模式CICM_k。

[0079] 根据步骤130的实施例,所述率失真成本RDC_k基于指示I_k个内插点PI_k(i,0)和N个其他点P(n,0)之间的空间距离的度量。

[0080] 根据实施例,度量D_k由下式给出:

$$[0081] \quad D_k = \text{Max}(\text{dist}_{\text{point}}(A,B); \text{dist}_{\text{point}}(B,A))$$

[0082] 其中

$$[0083] \quad \text{dist}_{\text{point}}(A,B) = \sum_{p \in A} \|P(n,0) - \text{closest}(P(n,0),B)\|_2^2$$

[0084] 其中, $\| \cdot \|_2^2$ 是指点集合A的点P(n,0)和点集合B的最近内插点PI_k(i,0)之间的空间距离的度量。例如,欧几里得距离用作度量,A中B的最近点P(n,0)是被如下定义的内插点PI_k(i,0):

$$[0085] \quad \text{closest}(P(n,0),B) = \underset{i=1,\dots,I_k}{\text{argmin}} \|P(n,0) - \text{PI}_k(i,0)\|_2^2$$

[0086] 距离dist_{point}(A,B)确保重构点离第一点不会太远,避免编译码无关的点。距离dist_{point}(B,A)确保第一点离重构点不会太远。

[0087] 根据变型,方法包括在计算率失真成本之前去除集合A中的重复。

[0088] 根据步骤130的实施例,至少一个候选内插编译码模式CICM_k通过创建具有在从重构点 $\widehat{P_0(p)}$ 的深度值到最大深度值的范围内的相关深度值的至少一个内插点PI_k(i,0)来定义。

[0089] 根据图2所示的步骤130的所述实施例的变型,至少一个候选内插编译码模式CICM_k定义如下。

[0090] 在步骤1300中,计算与重构点 $\widehat{P_0(p)}$ 相关联的深度值D($\widehat{P_0(p)}$)和至少一个点P(n,

0) 的深度 $D(n, 0)$ 之间的最大深度差 (距离) $\text{deltaMax}(\widehat{P_0(p)})$:

[0091] $\text{deltaMax}(\widehat{P_0(p)}) = \max(D(\widehat{P_0(p)}) - D(n, 0)), \quad n=0, \dots, N$

[0092] 根据变型, 如果 $D(\widehat{P_0(p)})$ 和 $D(n, 0)$ 之间的差属于由下限 LB 和上限 UB 定义的给定值范围, 则该差被忽略。例如, $LB=0, UB=4$ 。

[0093] 当没有差属于给定值范围时, 最大深度差 $\text{deltaMax}(\widehat{P_0(p)})$ 被设置为 1。

[0094] 然后, 不止一个候选内插编译码模式 CICM_k 可以通过例如选择不同的 LB 值和 UB 值来确定。在该情况下, 边界 LB 和 UB 不必被传输, 因为解码器根据内插编译码模式 ICM_k 的具体值预先知道那些值是什么。

[0095] 典型地, 一种候选内插模式可以用 $LB=0$ 和 $UB=1$ 来定义, 另一种可以用 $LB=0$ 和 $UB=2$ 来定义, 另一种可以用 $LB=0$ 和 $UB=4$ 来定义

[0096] 在步骤 1310 中, 至少一个内插点根据最大深度差 (距离) $\text{deltaMax}(\widehat{P_0(p)})$ 来创建。

[0097] 根据如图 3 所示的步骤 1310 的变型, 当 UB (阈值) = 4 时, 所有创建的内插点都与重构点 $\widehat{P_0(p)}$ 对齐, 并且步长 1 用于连续地将深度值分配给新的内插点。

[0098] 更详细地, 至少一个内插点被创建如下:

[0099] • 如果 $\text{deltaMax}(\widehat{P_0(p)})$ 等于为 0 或 1, 则在作为 $\widehat{P_0(p)}$ 的同一定位 p 处并且用相同深度值创建单个内插点。在图 3 中, 叉 20 与点 $\widehat{P_0(p)}$ 重叠。

[0100] • 如果 $\text{deltaMax}(\widehat{P_0(p)})$ 大于 1, 则在作为 $\widehat{P_0(p)}$ 的同一定位 p 处创建至少一个内插点, 每个创建的内插点具有相关联的深度值, 该相关联的深度值在从重构点 $\widehat{P_0(p)}$ 的深度值到最大深度值的范围内, 步长 1 用于连续地将深度值分配给新创建的内插点。

[0101] 在图 3 中, 内插点由叉 21 表示。

[0102] 注意在图 3 上, 为了清楚起见, 仅示出了单个点 $P(n, 0)$ 。

[0103] 根据图 3a 所示的变型, 上半部分的内插点与设置 deltaMax 的值的相邻点对齐, 而不是沿同一轴 (即在当前点的上方) 对齐所有内插点。

[0104] 根据变型, 最大深度差至少等于最小值 minDepth :

[0105] $\text{deltaMax}(\text{current}) = \max(\widehat{P_0(p)} - D(n, 0) - \text{minDepth})$

[0106] 其中, minDepth 是取决于要重构的表面厚度的给定值。

[0107] 然后, 不止一个候选内插编译码模式 CICM_k 可以通过选择不同的 minDepth 值 (通常为 0、1 或 2) 来确定。

[0108] 根据步骤 130 的实施例, 率失真成本 RDC_k 还基于指示 I_k 个内插点 $\text{PI}_k(i, 0)$ 的颜色和 N 个其他点 $P(n, 0)$ 的颜色之间的颜色距离的度量。

[0109] 根据实施例, 度量 D_k 由下式给出:

[0110] $D_k = \text{Max}(\text{dist}_{\text{geo+text}}(A, B); \text{dist}_{\text{geo+text}}(B, A))$

[0111] 其中

$$[0112] \quad dist_{geo+text}(A,B) = \frac{dist_{point}(A,B) + \mu * dist_{yuv}(A,B)}{1 + \mu}$$

[0113] 其中, μ 是例如等于 0.25 的权重系数, 并且

$$[0114] \quad dist_{yuv}(A,B) = \sum_{p \in A} \frac{(6D_y + D_u + D_v)}{8}$$

[0115] 其中:

$$[0116] \quad D_y = Y_p - Y_{closest}(P(n,0), B)$$

$$[0117] \quad D_u = U_p - U_{closest}(P(n,0), B)$$

$$[0118] \quad D_v = V_p - V_{closest}(P(n,0), B)$$

[0119] 并且, 三元组 (Y_p, U_p, V_p) 表示 YUV 颜色空间中点 $P(n,0)$ 的颜色, 三元组 $(Y_{closest}(P(n,0), B), U_{closest}(P(n,0), B), V_{closest}(P(n,0), B))$ 表示 YUV 颜色空间中所述点 $P(n,0)$ 的最近内插点 $P1_k(i,0)$ 的颜色。

[0120] 根据实施例, 至少一个候选内插编译码模式 $CICM_k$ 还定义颜色如何被分配给至少一个内插点。

[0121] 根据实施例, 分配给内插点的颜色是所述内插点的最近重构点的颜色。

[0122] 在变型中, 分配给内插点的颜色是所述内插点的相邻重构点的颜色的函数。函数意味着例如加权平均、最大值或最小值。

[0123] 根据步骤 110 的实施例, 邻域是图像区域, 即, 与第一点 $P0(p)$ 相关联的点云 PC 的所述至少一个其他点 $P(n,0)$ 属于图像区域。

[0124] 根据实施例, 率失真成本通过将为所述图像区域的图像子区域确定的率失真成本相加来获得。

[0125] 根据实施例, 为不同形状和/或不同尺寸的多个图像区域确定率失真成本, 并且选择图像区域的形状和/或尺寸以及与所述最低率失真成本相关联的内插编译码模式。

[0126] 根据步骤 140 的实施例, 内插编译码模式 ICM 被编码为与图像区域相关联的元数据。

[0127] 在变型中, 分配给两个不同图像区域的两个内插编译码模式之间的差被编码为元数据。

[0128] 例如, 对于每个分片, 每个分片 k 的每个模式 RDC_k (或两个 RDC_k 的差) 在比特流 BT 中被信令通知, 如下:

[0129]	对于所有分片	
	U0	DecodeUInt32(bitCountU0)
	V0	DecodeUInt32(bitCountV0)
	U1	DecodeUInt32(bitCountU1)
	V1	DecodeUInt32(bitCountV1)
	D1	DecodeUInt32(bitCountD1)
	deltaSizeU0	DecodeExpGolomb()

deltaSizeV0	DecodeExpGolomb()
NormaAxis	Decode
M_k	Decode

[0130] 其中, U_0 、 V_0 、 V_1 、 U_1 、 deltaSizeU0 和 deltaSizeV0 以及NormaAxis是由TMC2定义的语法元素。

[0131] 例如,对于每个块,每个块k的每个模式 RDC_k (或两个 RDC_k 的差) 在比特流BT中被信令通知:

[0132]	对于所有块	
	If 候选分片的数量 > 1	
	候选索引	Decode
	If 候选索引 == maxCandidateCount	
	块到分片索引	DecodeUInt32(bitCountPatch)
	Else	
	块到分片索引 = 候选索引	
	If (块到分片索引 != 0)	
	RDC_k	Decode

[0133] 其中,“候选分片的数量”、“候选索引”、“maxCandidateCount”、“块到分片索引”和块到分片索引是由TMC2定义的语法元素。

[0134] 图4示意性地示出了根据本实施例的示例的用于从比特流解码点云的点的深度值的方法200的步骤图。

[0135] 在步骤210中,模块M5从比特流BT获得至少一个重构点 $\widehat{P_0(p)}$ 。

[0136] 根据步骤210的实施例,模块M5通过解码比特流BT获得解码后的第一深度图像 $\widehat{D_0}$,并且考虑相对于所述解码后的第一深度图像 $\widehat{D_0}$ 的至少一个像素的至少一个重构点 $\widehat{P_0(p)}$ 。

[0137] 在步骤220中,模块M6从比特流BT获得至少一个内插编译码模式ICM,内插编译码模式ICM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逼近。

[0138] 根据步骤220的实施例,至少一个内插编译码模式ICM是从自比特流BT提取的元数据中获得的。

[0139] 在步骤230中,模块M7根据所述至少一个内插编译码模式ICM为至少一个重构点 $\widehat{P_0(p)}$ 创建至少一个内插点PI(i, 0),深度值与所述至少一个内插点中的每一个内插点相关联。

[0140] 然后,重构点云是从重构点 $\widehat{P_0(p)}$ 和内插点PI(i, 0)获得的。

[0141] 根据步骤230的实施例,与所述至少一个内插点相关联的深度值在从重构点 $\widehat{P_0(p)}$ 的深度值到最大深度值的范围内。

[0142] 根据实施例,相对于重构点 $\widehat{P_0(p)}$ 的所有创建的内插点都与重构点 $\widehat{P_0(p)}$ 对齐,并且步长1用于连续地将深度值分配给新的内插点。

[0143] 根据步骤230的所述实施例的变型,至少一个内插点根据如图2所示的步骤130的实施例的变型来创建。

[0144] 根据该方法的实施例,在步骤210中,该方法还包括获得包含至少一个重构点 $\widehat{P_0(p)}$ 的至少一个图像区域。

[0145] 在步骤220中,然后,内插编译码模式ICM按每图像区域从比特流BT中获得,所述内插编译码模式ICM指示图像区域中的点如何被至少一个其他内插点逼近。

[0146] 在步骤230中,然后,按每图像区域的至少一个内插点根据与所述图像区域相关联的内插编译码模式ICM来创建,并且深度值与所述至少一个内插点中的每一个内插点相关联。

[0147] 图5示意性地示出了如TMC2中定义的对点云的几何结构和纹理进行编码的方法。

[0148] 基本上,编码器在第一深度图像(D0)和深度图像第二(D1)中捕获点云PC的几何结构信息。

[0149] 作为示例,第一深度图像和第二深度图像在TMC2中被如下获得。

[0150] 深度分片(点云PC的3D点集合)是通过根据在点云PC的点处的法向量对这些点进行聚类来获得的。然后,所有提取的深度分片被投影到2D网格上并被打包,同时试图最小化未使用的空间,并保证网格的每个 $T \times T$ (例如, 16×16)块与唯一的分片相关联,其中T是被信令通知到比特流中的用户定义参数。

[0151] 然后,深度图像通过利用在打包过程期间计算的3D到2D映射来生成,更具体地,每个分片的投影区域的打包位置和尺寸。更准确地,让 $H(u, v)$ 作为当前分片中被投影到同一像素 (u, v) 的点的集合。第一层(也称为最近层或第一深度图像D0)存储具有最小深度值的点 $H(u, v)$ 。第二层(被称为最远层或第二深度图像D1)捕获在间隔 $[D, D + \Delta]$ 内具有最高深度值的点 $H(u, v)$,其中D是第一深度图像D0中像素的深度值, Δ 是描述表面厚度的用户定义参数。

[0152] 然后,第一深度图像D0输出打包过程。填补(pad)过程也用于填充(fill)分片之间的空白空间,以便生成适于视频压缩的分段平滑的第一深度图像。

[0153] 然后,生成的深度图像/层D0和D1被存储为视频帧,并使用任何传统视频编解码器(诸如HEVC)来压缩。

[0154] 编码器还通过对第一深度图像和第二深度图像进行编码/解码以及通过对所述解码后的第一深度图像 $\widehat{D_0}$ 和所述解码后的第二深度图像 $\widehat{D_1}$ 去投影(deproject)以重构点云的几何形状,来捕获两个纹理图像中的原始点云PC的纹理信息。一旦被重构,颜色(颜色转移(transferring))就以最小化颜色信息编译码误差的方式被分配给从原始点云PC的颜色信息重构的点云的每个点。

[0155] 根据一个实施例,对于每个重构点,原始点云中其最近点的颜色被分配为其要编译码的颜色。

[0156] 然后,第一纹理图像T0和第二纹理图像T1通过在与深度图像相同的位置中存储每个重构点的要编译码的颜色信息来生成,即 (i, u, v) 。

[0157] 图6示意性地示出了在图5的编码方法中使用方法100和200的示例。

[0158] 根据该示例,图5的第二深度图像D1的编码被图1的编码方法100代替,并且图5的

第二深度图像 $\widehat{D}_1$ 的解码被图4的解码方法200代替。

[0159] 在图6a所示的变型中,图5的第一深度图像和第二深度图像的编码与图1的编码方法100相关联,图5的第一深度图像和第二深度图像的解码与图4的解码方法200相关联。

[0160] 图7示意性地示出了如现有技术(TMC2)中定义的对点云的几何结构和纹理进行解码的方法。

[0161] 解码后的第一深度图像 $\widehat{D}_0$ 和解码后的第二深度图像 $\widehat{D}_1$ 通过解码比特流BT来获得。可能地,元数据也被解码以重构点云 $\widehat{PC}$ 的几何结构。

[0162] 因此,点云的几何结构通过对所述解码后的第一深度图像和所述解码后的第二深度图像以及可能还有所述元数据进行去投影来重构。

[0163] 图8示意性地示出了在图7的解码方法中使用方法200的示例。

[0164] 根据该示例,图7的第一深度图像和第二深度图像的解码被图4的解码方法代替。

[0165] 在图8a所示的变型中,图7的第一深度图像和第二深度图像的解码与图4的解码方法200相关联。

[0166] 在图1-图8a上,模块是功能单元,其可能与可区分的物理单元相关,也可能不相关。例如,这些模块或其中的某些模块可以在唯一的组件或电路中集合在一起,或者构成软件的功能。相反,某些模块可能由独立的物理实体组成。与本实施例兼容的装置或者使用纯硬件(诸如ASIC或FPGA或VLSI,分别是专用集成电路、现场可编程门阵列、超大规模集成电路)来实施,或者从嵌入在设备中的几个集成电子组件来实施,或者从硬件和软件组件的混合来实施。

[0167] 图9示出了实施了各个方面和实施例的系统的示例的框图。系统9000可以被体现为包括以下描述的各种组件并被配置为执行本申请中描述的一个或多个方面的设备。这种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、智能手机、平板电脑、数字多媒体机顶盒、数字电视接收机、个人视频记录系统、联网家用电器和服务。系统9000的元件可以单独地或组合地体现在单个集成电路、多个IC和/或分立组件中。例如,在至少一个实施例中,系统9000的处理元件和编码器/解码器元件分布在多个IC和/或分立组件上。在各个实施例中,系统9000经由例如通信总线或通过专用输入和/或输出端口而通信地耦合到其他类似系统或其他电子设备。在各个实施例中,系统9000被配置为实施本文档中描述的一个或多个方面。

[0168] 系统9000包括至少一个处理器9010,处理器9010被配置为执行加载在其中的指令,以用于实施例如本文档中描述的各个方面。处理器9010可以包括嵌入式内存、输入输出接口和本领域已知的各种其他电路。系统9000包括至少一个内存9020(例如,易失性存储设备和/或非易失性存储设备)。系统9000包括存储设备9040,存储设备9040可以包括非易失性存储器和/或易失性存储器,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,存储设备9040可以包括内部存储设备、附加存储设备和/或网络可访问存储设备。

[0169] 系统9000包括编码器/解码器模块9030,编码器/解码器模块9030被配置为例如处理数据以提供编码后的视频或解码后的视频,并且编码器/解码器模块9030可以包括其自己的处理器和内存。编码器/解码器模块9030表示可被包括在设备中以执行编码和/或解码

功能的(多个)模块。众所周知,设备可以包括编码和解码模块中的一个或两个。另外,编码器/解码器模块9030可以被实施为系统9000的独立元件,或者可以作为本领域技术人员已知的硬件和软件的组合被结合在处理器9010内。

[0170] 要加载到处理器9010或编码器/解码器9030上以执行本文档中描述的各个方面的程序代码可以被存储在存储设备9040中,并且随后被加载到内存9020上以供处理器9010执行。根据各个实施例,处理器9010、内存9020、存储设备9040和编码器/解码器模块9030中的一个或多个可以在执行本文档中描述的过程期间存储各种项目中的一个或多个。这种存储的项目可以包括但不限于输入视频、解码后的视频、点云、重构点云或解码后的视频的部分、比特流、矩阵、变量,以及来自等式、公式、操作和操作逻辑的处理的中间或最终结果。

[0171] 在几个实施例中,处理器9010和/或编码器/解码器模块9030内部的存储器用于存储指令,以及为编码或解码期间所需的处理提供工作内存。

[0172] 然而,在其他实施例中,处理设备外部的存储器(例如,处理设备可以是处理器9010或编码器/解码器模块9030)用于这些功能中的一个或多个。外部存储器可以是内存9020和/或存储设备9040,例如,动态易失性存储器和/或非易失性闪存。在几个实施例中,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用于视频编译码和解码操作的工作内存,诸如用于MPEG-2、HEVC、VVC(通用视频编译码)或TMC2。

[0173] 对系统9000的元件的输入可以通过如框9130所示的各种输入设备来提供。这种输入设备包括但不限于:(i)接收例如由广播公司通过空中传输的RF信号的RF部分,(ii)复合输入终端,(ii)USB输入终端,和/或(iv)HDMI输入终端。

[0174] 在各个实施例中,框9130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,RF部分可以与以下所需的元件相关联:(i)选择期望频率(也称为选择信号,或者将信号限带(band-limit)到一频带),(ii)对选择的信号进行下变频,(iii)再次限带到更窄的频带,以选择(例如)在某些实施例中可称为信道的信号频带,(iv)解调经下变频和限带的信号,(v)执行纠错,以及(vi)解复用以选择期望的数据包流。各个实施例的RF部分包括一个或多个执行这些功能的元件,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。RF部分可以包括执行各种这些功能的调谐器,包括例如将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。

[0175] 在一个机顶盒实施例中,RF部分及其相关联的输入处理元件接收通过有线(例如,电缆)介质传输的RF信号,并且通过滤波、下变频和再次滤波到期望频带来执行频率选择。

[0176] 各个实施例重新排列上述(和其他)元件的顺序,移除这些元件中的某些,和/或添加执行相似或不同功能的其他元件。

[0177] 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。在各个实施例中,RF部分包括天线。

[0178] 此外,USB和/或HDMI终端可以包括各自的接口处理器,以用于通过USB和/或HDMI连接将系统9000连接到其他电子设备。应该理解的是,输入处理的各个方面(例如,里德-所罗门纠错)可以根据需要在例如单独的输入处理IC内或处理器9010内实施。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内或处理器9010内实施。经解调、纠错和解复用的流被提供给各种处理元件,包括例如处理器9010和编码器/解码器9030,其与

存储器和存储元件结合操作,从而根据需要处理数据流以在输出设备上呈现。

[0179] 系统9000的各个元件可以设置在集成外壳内。在集成外壳内,各个元件可以互连,并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)在其间传输数据。

[0180] 系统9000包括能够经由通信信道9060与其他设备通信的通信接口9050。通信接口9050可以包括但不限于被配置为通过通信信道9060发送和接收数据的收发器。通信接口9050可以包括但不限于调制解调器或网卡,并且通信信道9060可以例如在有线和/或无线介质中实施。

[0181] 在各个实施例中,数据使用诸如IEEE 802.11的Wi-Fi网络被流式传输到系统9000。这些实施例的Wi-Fi信号通过适于Wi-Fi通信的通信信道9060和通信接口9050来接收。这些实施例的通信信道9060通常连接到接入点或路由器,该接入点或路由器提供对包括互联网在内的外部网络的访问,以允许流式应用和其他过顶通信。

[0182] 其他实施例使用机顶盒向系统9000提供流式数据,机顶盒通过输入框9130的HDMI连接传送数据。

[0183] 还有其他实施例使用输入框9130的RF连接向系统9000提供流式数据。

[0184] 系统9000可以向各个输出设备(包括显示器9100、扬声器9110和其他外围设备9120)提供输出信号。在实施例的各个示例中,其他外围设备9120包括独立DVR、光盘播放器、立体声系统、照明系统以及基于系统9000的输出提供功能的其他设备中的一个或多个。

[0185] 在各个实施例中,控制信号使用诸如AV.Link、CEC或支持带有或无需用户干预的设备到设备通信的其他通信协议之类的信令在系统9000和显示器9100、扬声器9110或其他外围设备9120之间传送。

[0186] 输出设备可以通过各自的接口9070、9080和9090经由专用连接而通信地耦合到系统9000。

[0187] 替代地,输出设备可以经由通信接口9050使用通信信道9060连接到系统9000。显示器9100和扬声器9110可以与电子设备(例如,电视机)中的系统9000的其他组件集成在单个单元中。

[0188] 在各个实施例中,显示接口9070包括显示驱动器,例如时序控制器(T Con)芯片。

[0189] 替代地,显示器9100和扬声器9110可以与一个或多个其他组件分离,例如,如果输入9130的RF部分是独立机顶盒的一部分的话。在显示器9100和扬声器9110是外部组件的各个实施例中,输出信号可以经由专用输出连接(包括例如HDMI端口、USB端口或COMP输出)来提供。

[0190] 根据图10所示的本实施例的示例,在通信网络NET上的两个远程设备A和B之间的传输上下文中,设备A包括与存储器RAM和ROM相关的处理器,其被配置为实施与图1-图3a或图5-图6相关地描述的方法,设备B包括与存储器RAM和ROM相关的处理器,其被配置为实施与图4或图7-图8相关地描述的方法。

[0191] 根据示例,网络是广播网络,其适于将静止图片或视频图片从设备A广播到包括设备B的解码设备。

[0192] 旨在要由设备A传输的信号携带比特流BT。比特流BT包括表示点云的点的深度值的编译码数据、点云的至少一个重构点、以及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逼近的至少一个内插编译译码模式。

[0193] 例如,如与图1-图3a相关地解释的,获得所述编译码的数据。

[0194] 图11示出了当数据通过基于分组的传输协议传输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。例如,报头H的比特专用于表示内插编译码模式ICM。因此,报头H的至少一个比特专用于表示至少一个内插编译码模式ICM。

[0195] TMC2中的空间自适应重构模式的实施方式如下:

[0196] 1. 编码单个深度&纹理层的可能性——目前TMC2总是编码两层:错误!未找到参考源。这是通过添加编码器参数/比特流参数以在GOP级别信令通知深度&纹理层的数量(两者相等)来实施的。

[0197] 2. 在单层模式中,使用优化框架以将点重构模式(称为点局部重构模式(Point Local Reconstruction Mode, PLRM))局部适应于点云的特性。在编码器中使用类似RDO的方法来执行优化,从而在比特流中添加所需的信令,使得解码器获悉局部重构模式要按每点使用。这是以区域尺寸(以占用图(occupancy map)分辨率的块,在本文(contribution)中为 16×16 像素)完成的。该过程被命名为点局部重构框架(Point Local Reconstruction Framework, PLRF)。

[0198] 信令通知PLRM需要专用算术编码器,这增加了编码器和解码器的复杂性。

[0199] 至少一个实施例将在点局部重构模式(PLRM)(即至少一个内插编译码模式)上传送(信令通知)数据,以作为图像数据。

[0200] 每个预测模式被编译码到块到分片(block to patch)元数据信息,该元数据信息被编译码到辅助分片信息中(使用算术编码器)。当前的实施方式尝试了多达10个不同的模式。图12示出了在编码器端决定的不同模式,以及每个模式在TMC2 v3.0软件中关于速率和配置(全帧内模式、随机访问模式)的使用的分布。

[0201] 在实施方式中,关于PLRM的数据使用块到分片元数据被发送到解码器,并且由算术编码器编码。

[0202] 占用图(占用图精度为 $B_0 \times B_0$ 的块,即 4×4 像素)被编译码为视频(使用HEVC)。它由二进制图(binary map)组成,该二进制图为网格的每个单元指示它是属于空白空间还是属于点云。占用图的使用可以在PLRF的情况下扩展,以便传送(信令通知)这种信息。

[0203] 本实施例的至少一个实施例涉及一种将PLRM编译码到占用图中的方式,因为它是最有可能可用于在当前TMC2中编译码这种信息的模式。

[0204] 主要思想是展示一种如何使用占用图来传送关于点局部重构模式的信息的方式。到目前为止,占用图包含“0”以指示对应的3D块(点云)为空,或包含“1”以指示该3D块已被完全占用,如图12所示。至少一个实施例获得更多的值来指示PLRM。

[0205] 更详细地,本实施例中的至少一个涉及以下方法:

[0206] 对于占用图精度为 (4×4) 的每个块,PLRM设置如下:

[0207] -如果块为空,则该块被设置为0。

[0208] -如果块被占用,则分配给该块的值对应于范围 $[1 \cdots N]$ 内的点局部重构模式,N是要应用的点局部重构模式的最大数量。注意,“1”值仍然指示该块已被占用,并且没有补充点会被添加到利用该模式的重构点云。

[0209] 根据实施方式,由于点局部重构模式是在占用图分辨率(16×16)上,所以,除非块

未被占用并且在该情况下被设置为0,否则重复占用图精度(4*4)处的每个块上的信息。

[0210] 图13a利用图13的示例示出了如何将PLRM编译码到占用图中的示例。它示出了“1”、“2”、“5”和“7”四种模式的使用。“0”意味着重构块为空。

[0211] 注意,PLRM可以以占用精度(4×4)来完成。在该情况下,PLRM的编译码信息可能因块而异,如图13b所示。这将使重构过程更加精确。

[0212] PLRM编译码的变型是为不同类型的重构(PLRM和其他可应用的重构方式)分配值范围:

[0213] -如果块为空,则块被设置为0(如今天一样(as today))。

[0214] -如果块被占用,则第一范围[0;m]被保留为分配PLRM,第二范围[m;N]被保留为分配描述点云重构的另一方法的信息。

[0215] 实施方式的另一变型是使用比特掩码:

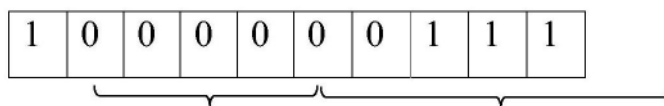
[0216] -如果块为空,则该块被设置为0(如今天一样)。

[0217] -如果块被占用,则我们通过以下方式使用分量的N个比特:

[0218] o将PLRM分配给前n个比特

[0219] o分配(N-n)个比特以指示其他重构模式

[0220] 例如,利用编译码在10个比特上的分量,4个比特用于编译码信息“1”,6个其他比特用于编译码PLRM信息。



[0221]

$I=1000b$

$PLRM=000111b$

[0222] 实施方式的另一变型是使用哈希图(hash map)来确定不同的PLRM。它可以为每一层指定PLRM。

[0223] 例如,如果 $PLRM_0$ 是第0层的重构编译码模式, $PLRM_1$ 是第1层的重构编译码模式, $PLRM_2$ 是第2层的重构编译码模式,则占用图可被填充如下:

[0224] -如果块为空,则该块被设置为0(如今天一样)。

[0225] -如果块被占用,则哈希表可以如下:

键	值
1	$PLRM_0$ 为 1 $PLRM_1$ 为 1 $PLRM_2$ 为 1

[0226]

[0227]	2	$PLRM_0$ 为 2 $PLRM$ 为 1 $PLRM_2$ 为 1
	...	...
	N	$PLRM_0$ 为 N $PLRM_1$ 为 N $PLRM_2$ 为 N

[0228] 另一变型可以是使用占用图的色度分量来编译码PLRM信息。

[0229] 主要优点在于避免将数据编译码为辅助分片信息,以及利用现有硬件对占用图进行视频压缩。

[0230] 它遵循这样的思想,即使用占用图信息作为如何重构点云的收集器(collector)。

[0231] 注意,这是点局部重构模式如何在TMC2内传送的最有可能的情况。

[0232] 本文描述的各种过程和特征的实施方式可以体现在各种不同的装备或应用中。这种装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前处理器、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、手机、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。应该清楚的是,装备可以是移动的,甚至可以安装在移动车辆中。

[0233] 此外,这些方法可以通过由处理器执行的指令来实施,并且这种指令(和/或由实施方式产生的数据值)可以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体现在一个或多个计算机可读介质中并且其上体现有可由计算机执行的计算机可读程序代码的计算机可读程序产品的形式。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对其中的信息的检索的固有能力。计算机可读存储介质可以是,例如但不限于电子、磁性、光学、电磁、红外或半导体的系统、装置或设备,或前述各项的任何合适的组合。应该理解的是,尽管提供了可应用本实施例的计算机可读存储介质的更具体示例,但是如本领域的普通技术人员容易理解的,以下各项仅仅是说明性的而非穷尽性的列举: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;磁性存储设备;或前述各项的任何合适的组合。

[0234] 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。

[0235] 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过程的设备和包括

具有用于执行过程的指令的处理器可读介质的设备(诸如存储设备)。此外,除了指令之外或代替指令,处理器可读介质可以存储由实施方式产生的数据值。

[0236] 对于本领域技术人员来说显而易见的是,实施方式可以产生各种信号,这些信号被格式化以携带例如可被存储或传输的信息。信息可以包括例如用于执行方法的指令,或者由所描述的实施方式之一产生的数据。例如,信号可以被格式化以携带用于写入或读取本实施例的所描述的示例的语法的规则作为数据,或者携带由本实施例的所描述的示例写入的实际语法值作为数据。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。格式化可以包括例如编码数据流和用编码后的数据流调制载波。信号所携带的信息可以是例如模拟或数字信息。众所周知,信号可以通过各种不同的有线或无线链路来传输。信号可以存储在处理器可读介质上。

[0237] 已经描述了许多实施方式。然而,应该理解的是,可以进行各种修改。例如,不同实施方式的元素可以被组合、补充、修改或移除以产生其他实施方式。此外,本领域普通技术人员将理解,其他结构和过程可以被用来代替所公开的那些结构和过程,并且所得到的实施方式将以至少基本相同的(多个)方式执行至少基本相同的(多个)功能,以实现与所公开的实施方式至少基本相同的(多个)结果。相应地,本申请考虑了这些实施方式和其他实施方式。

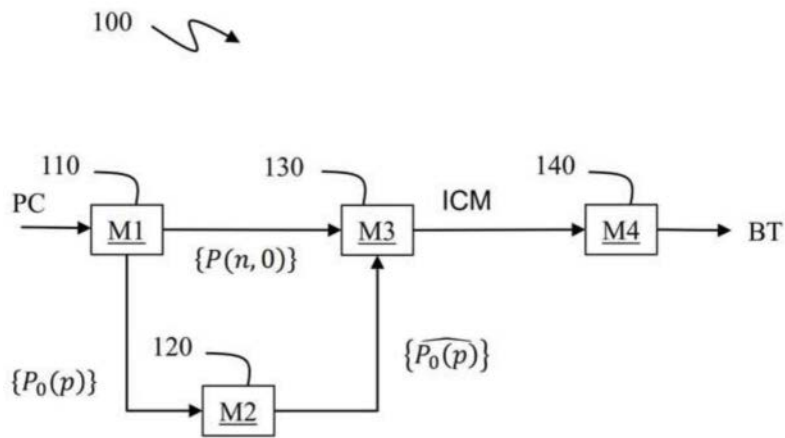


图1

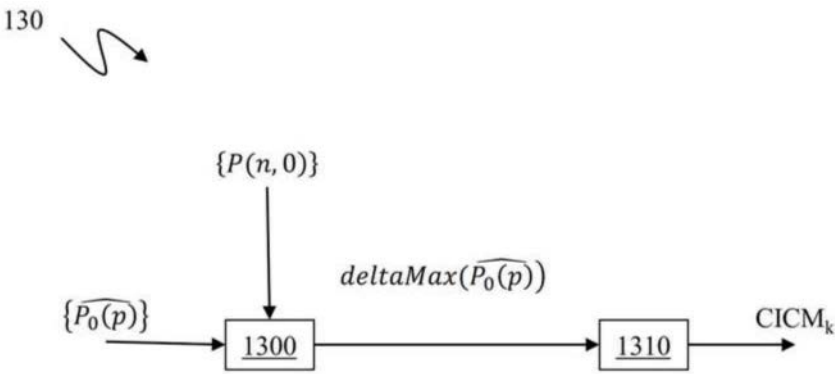


图2

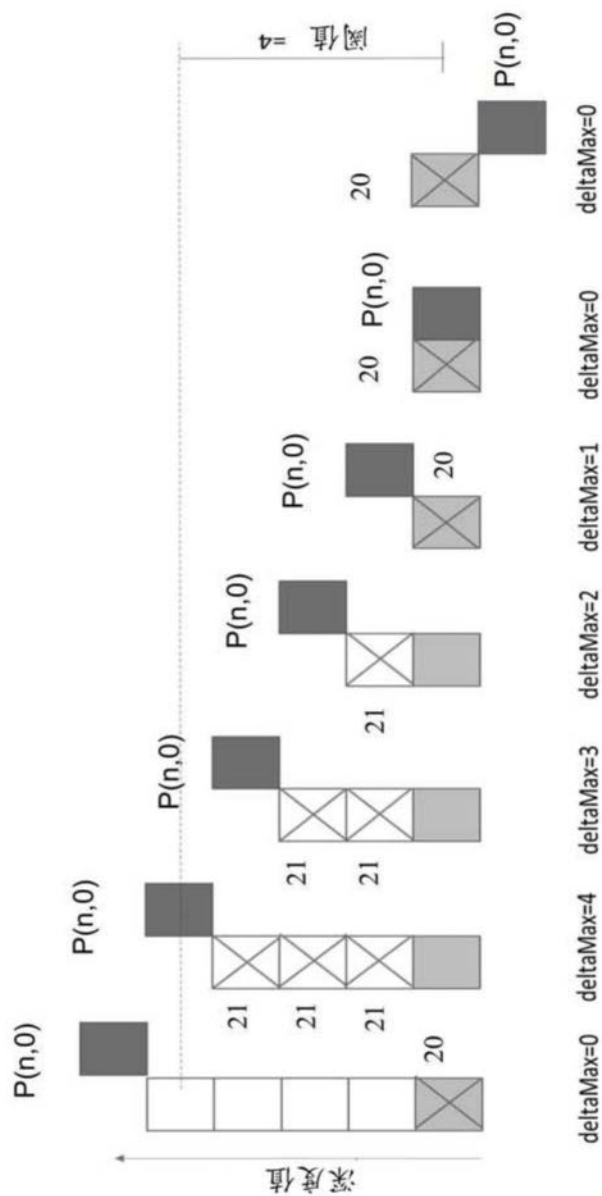


图3

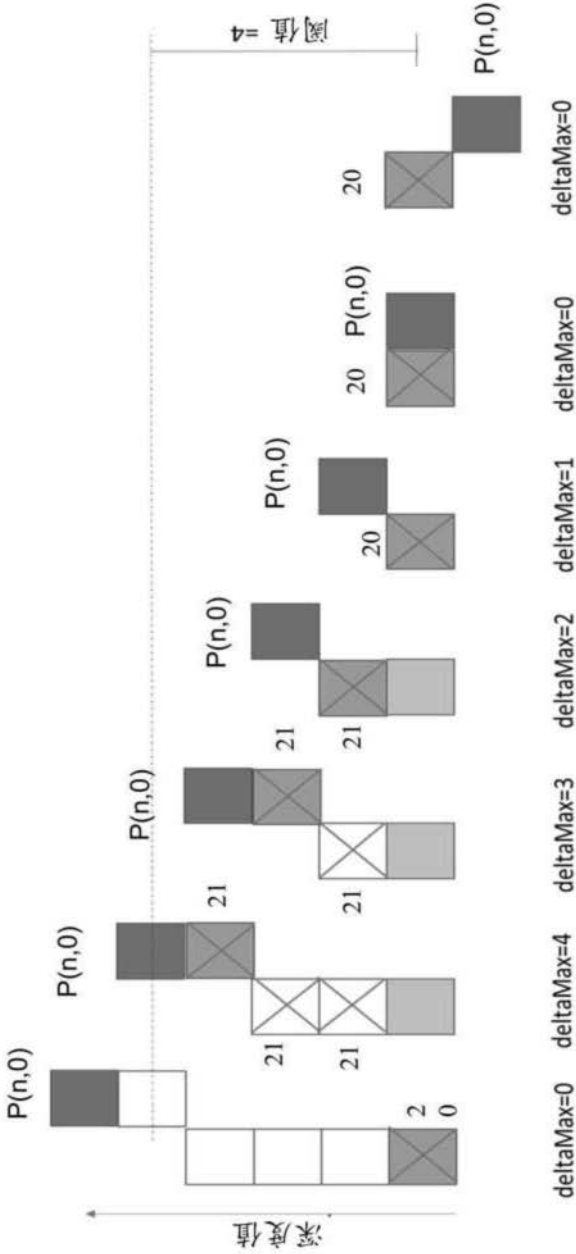


图3a

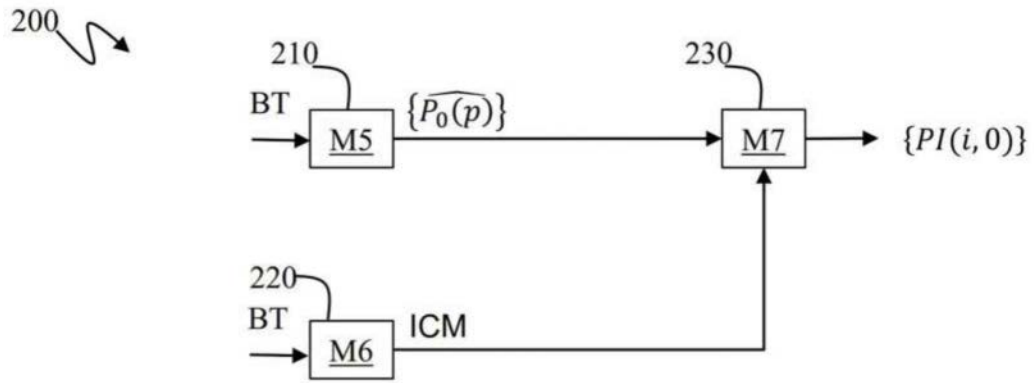


图4



图10



图11

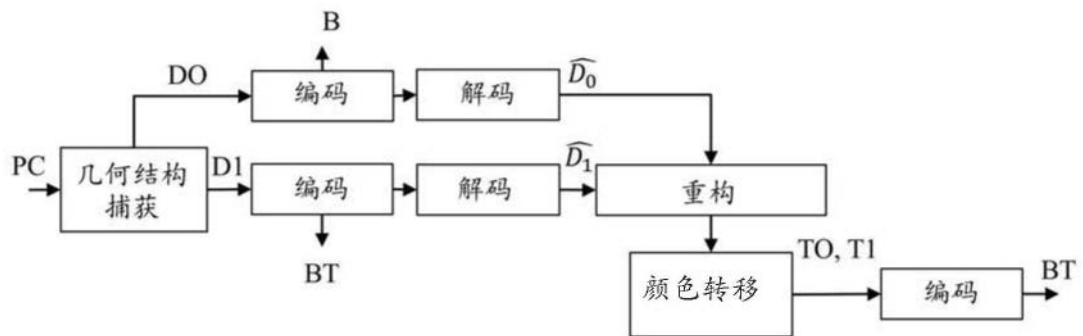


图5

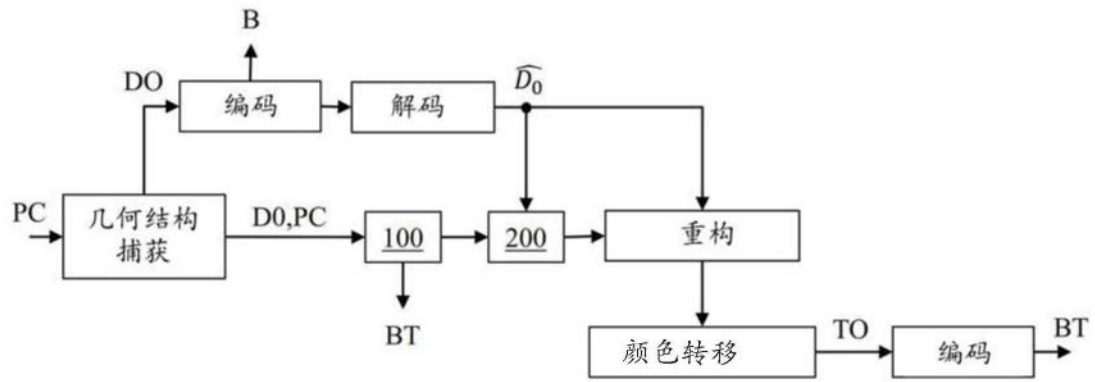


图6

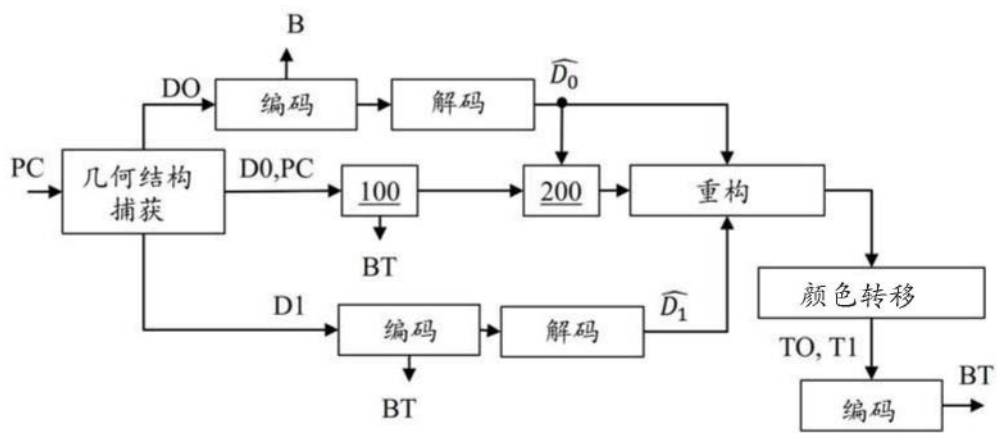


图6a

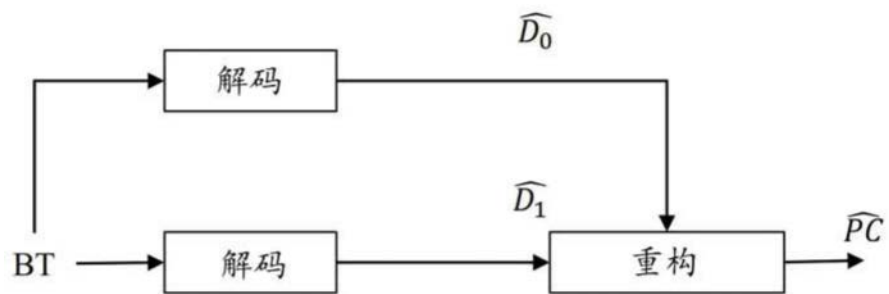


图7

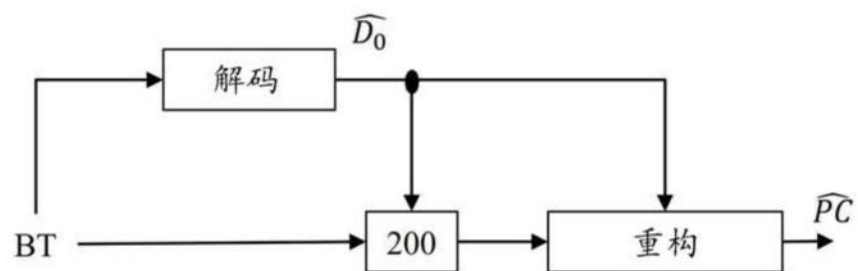


图8

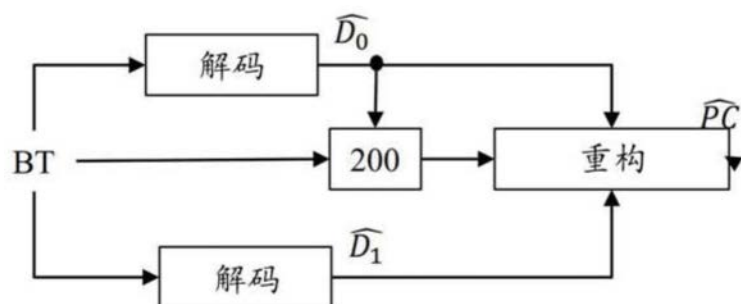


图8a

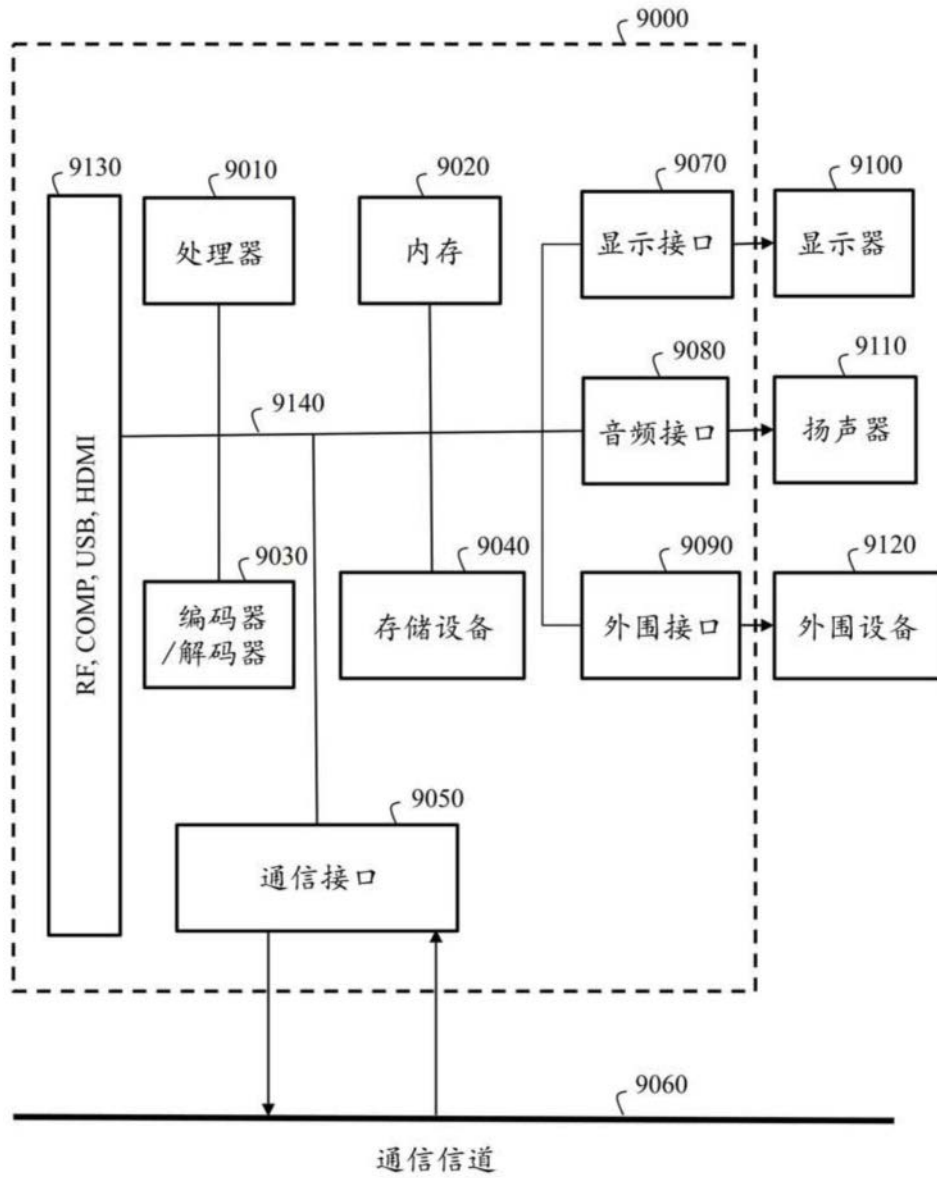


图9

重构编译码模式	1	2	3	4	5	6	7	8	9	10
MinD1	0	0	0	0	0	1	1	1	1	1
填充	0	0	1	0	1	0	0	1	0	1
内插	0	1	1	2	2	0	1	1	2	2
平均值 R01	24.71%	7.30%	2.77%	23.36%	19.02%	15.08%	0.86%	1.24%	2.66%	3.04%
平均值 R02	24.01%	8.12%	3.40%	26.98%	22.15%	10.86%	0.77%	1.41%	0.85%	1.46%
平均值 R03	22.10%	10.02%	4.36%	29.51%	24.20%	6.97%	0.36%	1.64%	0.11%	0.74%
平均值 R04	20.37%	11.67%	4.86%	30.35%	24.81%	5.48%	0.20%	1.67%	0.07%	0.51%
平均值 R05	18.95%	12.61%	5.55%	29.14%	26.34%	4.78%	0.09%	2.02%	0.05%	0.48%
平均值 AI	22.30%	9.89%	4.14%	27.83%	23.20%	8.61%	0.43%	1.59%	0.73%	1.28%
平均值 RA	21.77%	10.00%	4.21%	27.91%	23.41%	8.66%	0.48%	1.60%	0.76%	1.21%
平均值	22.03%	9.94%	4.18%	27.87%	23.30%	8.63%	0.46%	1.60%	0.75%	1.25%

TMC2中关于速率和配置的重构模式的使用的分布的示例

图12

0	0	0	1	1	0	0	1
0	0	1	1	1	1	1	1
0	0	0	1	1	1	1	1
0	0	0	0	0	1	1	0
0	0	0	1	1	1	1	1
0	0	0	1	1	1	1	1
0	0	0	1	1	1	1	1
0	0	0	0	1	1	1	1

当前TMC2占用图图像

图13

0	0	0	7	2	0	0	2
0	0	7	7	2	2	2	2
0	0	0	7	2	2	2	2
0	0	0	0	0	2	2	0
0	0	0	5	1	1	1	1
0	0	0	5	1	1	1	1
0	0	0	5	1	1	1	1
0	0	0	0	1	1	1	1

利用PLRM的占用图（16×16精度）

图13a

0	0	0	7	2	0	0	2
0	0	7	7	7	2	2	2
0	0	0	2	2	2	1	1
0	0	0	0	0	1	1	0
0	0	0	5	5	1	1	1
0	0	0	5	5	1	1	1
0	0	0	5	5	5	1	1
0	0	0	0	5	5	1	1

利用重构编译码模式的占用图（4×4精度）

图13b