



(12) 发明专利

(10) 授权公告号 CN 111971967 B

(45) 授权公告日 2024. 09. 17

(21) 申请号 201980025241.2

(22) 申请日 2019.04.03

(65) 同一申请的已公布的文献号  
申请公布号 CN 111971967 A

(43) 申请公布日 2020.11.20

(30) 优先权数据  
18305431.1 2018.04.11 EP  
18306132.4 2018.08.23 EP

(85) PCT国际申请进入国家阶段日  
2020.10.12

(86) PCT国际申请的申请数据  
PCT/US2019/025485 2019.04.03

(87) PCT国际申请的公布数据  
WO2019/199531 EN 2019.10.17

(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-C.谢维特 Y.奥利维尔 蔡康颖

(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.  
H04N 19/597 (2006.01)

(56) 对比文件  
US 2016275690 A1, 2016.09.22  
Khaled Mammou. PCC Test Model Category 2 v0.《120. MPEG Meeting; 2017.10.23-2017.10.27; Macau; MOTION PICTURE EXPERT GROUP OR ISO/IEC JTC1/SC29/WG11, no. N17248, 2017.12.15, XP030023909》. 2017, 1-11.

INDRANIL SINHAROY等. Lossless Coding in TMC2.《121. MPEG MEETING; 22-1-2018 - 26-1-2018; GWANGJU; MOTION PICTURE EXPERT GROUP OR ISO/IEC JTC1/SC29/WG11, no. m42170, 2018-1-17, XP030070512》. 2018, 1-10.

审查员 章子衡

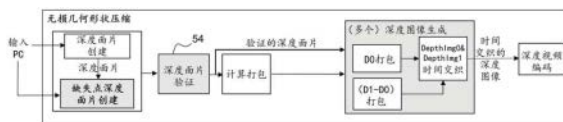
权利要求书2页 说明书16页 附图17页

(54) 发明名称

一种用于编码/解码表示3D对象的点云的方法和装置

(57) 摘要

至少一个实施例涉及一种方法, 该方法包括获得占用信息, 该占用信息指示在投影平面上定义的2D规则网格的块是否与点云的至少一个点在该投影平面上的正交投影的深度信息相关联; 将占用信息嵌入到存储该深度信息的图像中; 以及编码所述图像。



1. 一种用于编码表示3D对象的点云的方法,包括:
  - 获得占用信息,所述占用信息指示表示点云的第一点沿投影线在投影平面上的投影的像素是否被占用;
  - 将所述占用信息存储到图像中,所述占用信息具有将比特级联的码字,所述比特针对沿所述投影线在所述投影平面上投影的所述点云的所述第一点的第一深度值和第二点的第二深度值之间沿所述投影线的位置,指示所述位置是否被占用;以及
  - 编码所述图像。
2. 根据权利要求1所述的方法,其中将所述占用信息存储到图像中包括将要存储的所述码字增加1。
3. 根据权利要求2所述的方法,其中,所述方法还包括传输用于指示所述码字已经被偏移的信息。
4. 一种用于编码表示3D对象的点云的设备,包括:
  - 用于获得占用信息的部件,所述占用信息指示表示点云的第一点沿投影线在投影平面上的投影的像素是否被占用;
  - 用于将所述占用信息存储到图像中的部件,所述占用信息具有将比特级联的码字,所述比特针对沿所述投影线在所述投影平面上投影的所述点云的所述第一点的第一深度值和第二点的第二深度值之间沿所述投影线的位置,指示所述位置是否被占用;以及
  - 用于编码所述图像的部件。
5. 根据权利要求4所述的设备,其中将所述占用信息存储到图像中包括将要存储的所述码字增加1。
6. 根据权利要求5所述的设备,还包括用于传输用于指示所述码字已经被偏移的信息的部件。
7. 一种用于解码表示3D对象的点云的方法,包括:
  - 解码存储占用信息和将比特级联的码字的图像,所述比特针对沿投影线在投影平面上投影的点云的第一点的第一深度值和沿所述投影线在所述投影平面上投影的所述点云的第二点的第二深度值之间沿所述投影线的位置,指示所述位置是否被占用;
  - 从解码的图像中推导出所述占用信息,所述占用信息指示表示所述第一点在所述投影平面上的投影的像素是否被占用。
8. 根据权利要求7所述的方法,其中,所述方法还包括接收指示所述码字是否必须被偏移的信息,并且根据所接收的信息将解码的码字减少1。
9. 一种用于解码表示3D对象的点云的设备,包括:
  - 用于解码存储占用信息和将比特级联的码字的图像的部件,所述比特针对沿投影线在投影平面上投影的点云的第一点的第一深度值和沿所述投影线在所述投影平面上投影的所述点云的第二点的第二深度值之间沿所述投影线的位置,指示所述位置是否被占用;
  - 用于从解码的图像中推导出所述占用信息的部件,所述占用信息指示表示所述第一点在所述投影平面上的投影的像素是否被占用。
10. 根据权利要求9所述的设备,还包括用于接收指示所述码字是否必须被偏移的信息的部件,以及用于根据所接收的信息将解码的码字减少1的部件。
11. 一种非暂时性处理器可读介质,包括用于使一个或多个处理器执行以下操作的指

令：

-获得占用信息,所述占用信息指示表示点云的第一点沿投影线在投影平面上的投影的像素是否被占用;

-将所述占用信息存储到图像中,所述占用信息具有将比特级联的码字,所述比特针对沿所述投影线在所述投影平面上投影的所述点云的所述第一点的第一深度值和第二点的第二深度值之间沿所述投影线的位置,指示所述位置是否被占用;以及

-编码所述图像。

12.一种非暂时性处理器可读介质,包括用于使一个或多个处理器执行以下操作的指

令：

-解码存储占用信息和将比特级联的码字的图像,所述比特针对沿投影线在投影平面上投影的点云的第一点的第一深度值和沿所述投影线在所述投影平面上投影的所述点云的第二点的第二深度值之间沿所述投影线的位置,指示所述位置是否被占用;

-从解码的图像中推导出所述占用信息,所述占用信息指示表示所述第一点在所述投影平面上的投影的像素是否被占用。

## 一种用于编码/解码表示3D对象的点云的方法和装置

### 技术领域

[0001] 本原理通常涉及对表示3D对象的点云的编码和解码。特别地,但不排他地,本原理的技术领域涉及点云的基于投影的编码/解码。

### 背景技术

[0002] 本章节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/或要求保护的本原理的各个方面相关。该讨论被认为有助于向读者提供背景信息,以便更好地理解本原理的各个方面。因此,应该理解,这些陈述应该从这个角度来理解,而不是作为对现有技术承认。

[0003] 点云是一组点,通常旨在表示3D对象的外表面,但也包括更复杂的几何形状,如头发或毛发,它们可能无法通过如网格的其它数据格式来有效地表示。点云的每个点通常由3D空间位置(3D空间中的X、Y和Z坐标)定义,也可能由其它相关联的属性定义,诸如例如在RGB或YUV颜色空间中表示的颜色、透明度、反射率、法线量等。

[0004] 着色点云(colored point)可以是6个分量(X,Y,Z,R,G,B)或等同的(X,Y,Z,Y,U,V)的点的集合,其中(X,Y,Z)定义了3D空间中的点的空间位置,(R,G,B)或(Y,U,V)定义了该点的颜色。

[0005] 在下文中,术语“点云”是指包括着色点云在内的任何点云。

[0006] 着色点云可能是静态的,也可能是动态的,这取决于点云是否随时间演变。应该注意,在动态点云的情况下,点数不一定是常数,而是相反地,通常随着时间而演变。因此,动态点云是按时间顺序排列的点的集合列表。

[0007] 实际上,着色点云可以用于各种目的,诸如文化遗产/建筑物,其中,像雕像或建筑物这样的对象被3D扫描,以便共享对象的空间配置,而无需发送或访问它。同时,这也是确保保存对象的知识以防其被破坏的方法;例如,地震中的寺庙。这种着色点云通常是静态的、巨大的。

[0008] 另一用例是在地形学和制图学中,通过使用3D表示,地图不局限于平面,并且可能包括地形(relief)。

[0009] 汽车工业和自动驾驶汽车也是其中可以使用点云的领域。自动驾驶汽车应该能够“探测”它们的环境,以根据它们紧邻的现实做出安全驾驶的决策。典型的传感器产生由决策引擎使用的动态点云。这些点云不旨在被人看到。它们通常较小,不一定是着色的,并且是动态的具有较高的捕获频率。它们可以具有其它属性,如反射率,反射率是与被感测对象的物理表面的材料相关的有价值的信息,并且可能有助于决策。

[0010] 虚拟现实(Virtual Reality,VR)和沉浸式世界最近成为热门话题,并被许多人预见为2D平板视频的未来。基本思想是让观众沉浸在其周围的环境中,而不是只能看着面前的虚拟世界的标准TV。沉浸感有几个层次,取决于观众在环境中的自由度。着色点云是用于分布式VR世界的很好的格式候选。它们可以是静态的,也可以是动态的,通常是平均大小的,比如一次不超过几百万个点。

[0011] 只有当比特流的大小足够低以允许向终端用户进行实际的存储/传输时,点云压缩才能成功地成为沉浸式世界存储/传输3D对象。

[0012] 能够以合理的带宽消耗向终端用户分发动态点云,同时保持可接受(或最好非常好)的体验质量,这一点也很重要。与视频压缩类似,时间相关性的良好使用被认为是将导致动态点云的有效压缩的关键因素。

[0013] 众所周知的方法将表示3D对象的几何形状和颜色的着色点云投影到包围3D对象的立方体的面上,以获得包含表示点云的颜色信息的纹理和表示点云的几何形状信息的深度信息的视频,并使用诸如3D-HEVC的传统编码器对纹理和深度视频进行编码(HEVC的扩展,其规范可在ITU网站上找到,T建议,H系列,h265,https://www.itu.int/rec/T-REC-H.265-201802-I annex I)。

[0014] 对于每个投影点,压缩的性能接近于视频压缩,但是当考虑动态点云时,由于遮挡、冗余和时间稳定性,一些内容可能更复杂。因此,就比特率而言,点云压缩比视频压缩要求更高。

[0015] 关于遮挡,如果不使用许多投影,实际上不可能获得复杂拓扑的完整几何形状。因此,对所有这些投影进行编码/解码所需的资源(计算能力、储存存储器)通常过高。

[0016] 最近在第120次MPEG会议上提出了一种使用纹理和深度投影方案的点云编解码器。该方案在文件S0/IEC JTC1/SC29 WG11 Doc.N17248,澳门,CN,2017年10月中有所描述,其中提出的方案称为“点云压缩测试模型类别2.0.0版(TMC2v0)”。

[0017] 通过本质上将动态点云数据转换成一组不同的视频序列,该编解码器利用现有的视频编解码器来压缩动态点云的几何形状和纹理信息。

[0018] 具体地,使用现有的视频编解码器,例如使用HEVC主轮廓编码器,生成并压缩两个视频序列,一个用于捕获点云数据的几何形状信息,另一个用于捕获纹理信息,即点的颜色。

[0019] 还分别生成和压缩解析这两个视频序列所需的附加元数据,即占用图和辅助面片(patch)/块信息。然后,将生成的视频比特流和元数据复用在一起,以生成最终的点云比特流。元数据进一步由诸如算术编解码器的熵编解码器编码。

[0020] 该点云编解码器的示例性编码器2的结构如图1所示。

[0021] 编码器2包括3个主要的块:几何形状压缩块4、元数据压缩块6和纹理压缩块8。

[0022] 几何形状压缩块4包括深度面片创建块10,其将输入点云分解成深度面片。

[0023] 深度面片创建过程10旨在将点云分解成具有平滑边界的最小数量的深度面片,同时也最小化重构误差。深度面片通过与深度值相关联的一组像素来表示点云的一部分。

[0024] 作为示例,实施了图2中详述的以下方法。

[0025] 首先,在102处估计每个点的法线,如Hugues Hoppe, Tony DeRose, Tom Duchamp, John McDonald, Werner Stuetzle, “从无组织点进行表面重构(Surface reconstruction from unorganized point)” ACM SIGGRAPH 1992会议录, 71-78中所述。

[0026] 然后,在104处,通过将每个点与由它们的法线所定义的以下六个定向平面之一相关联来获得点云的初始聚类:

[0027] - (1.0, 0.0, 0.0),

[0028] - (0.0, 1.0, 0.0),

[0029] - (0.0,0.0,1.0) ,

[0030] - (-1.0,0.0,0.0) ,

[0031] - (0.0,-1.0,0.0) ,和

[0032] - (0.0,0.0,-1.0) 。

[0033] 更准确地说,每个点与具有最接近法线的平面相关联(即,最大化点法线与平面法线的点积)。

[0034] 根据一个实施例,上述六个定向平面被索引如下:

[0035] - (1.0,0.0,0.0) 由聚类索引0索引;

[0036] - (0.0,1.0,0.0) 由聚类索引1索引;

[0037] - (0.0,0.0,1.0) 由聚类索引2索引;

[0038] - (-1.0,0.0,0.0) 由聚类索引0索引;

[0039] - (0.0,-1.0,0.0) 由聚类索引1索引;

[0040] - (0.0,0.0,-1.0) 由聚类索引2索引。

[0041] 然后,每个点都与相应聚类的索引相关联。

[0042] 然后,通过基于每个点的法线和其最近邻近点的聚类索引来迭代更新与每个点相关联的聚类索引,从而细化初始聚类。根据一个实施例,对于所考虑的点,如果它的所有邻近点(例如在3D空间中到当前点的距离小于2的所有点)被分配给一个聚类,记为聚类ClusterA,而所考虑的点被分配给另一个聚类,则所考虑的点的聚类索引被更新为聚类ClusterA。

[0043] 然后,在106处,通过应用连通分量提取过程(即提取具有相同聚类索引的邻近点以形成连通分量)来提取深度面片。根据一个实施例,通过对同时满足以下两个条件的所有点进行分组来构建一个连通分量:

[0044] -所有点都具有相同的聚类索引。

[0045] -对于所关注的连通分量中的每个点,同一连通分量中至少存在一个点到当前点的距离小于预定义阈值(例如,等于1)。

[0046] 每个连通分量根据其聚类索引(即属于它的点的聚类索引)与投影平面相关联。一个实施例如下:

[0047] -对于聚类索引为0的连通分量,YOZ平面;

[0048] -对于聚类索引为1的连通分量,XOZ平面;

[0049] -对于聚类索引为2的连通分量,XOY平面。

[0050] 为了更好地处理沿着同一投影线的多个点的情况,(其中投影线包括具有三个坐标(X,Y,Z)中两个相同坐标的点);例如,投影线可以由具有相同(X,Y)坐标的所有点组成,为每个连通分量生成两个深度面片,记录深度值D0和D1,对应于相应像素的最小和最大深度值,并满足 $D1 - D0 \leq \text{SurfaceThickness}$ ,其中SurfaceThickness是最大表面厚度,例如等于4。

[0051] 更准确地说,让 $\{H(u,v)\}$ 是当前连通分量中被投影到同一个像素(u,v)的点的集合。第一深度面片,也称为近层,存储 $\{H(u,v)\}$ 中具有最低深度D0的点的深度。第二深度面片,称为远层,捕获 $\{H(u,v)\}$ 中具有在区间 $[D0, D0 + \text{SurfaceThickness}]$ 内的最高深度的点。

[0052] 深度面片创建块10之后是打包块(packaging block)12,打包块12将所有投影的连

通分量映射到2D网格上,同时尝试最小化未使用的空间,并保证网格的每个TxT(例如,16×16)块与唯一的投影的连通分量相关联,其中T是被编码在比特流中并被发送到解码器的用户定义参数。2D网格上的每个点对应于图像上的像素。

[0053] 打包使用简单的打包策略,该策略迭代地尝试将面片插入WxH网格,其中W和H是用户定义参数,例如1028x1028,其对应于将被编码的几何形状/纹理图像的分辨率。面片位置通过以光栅扫描顺序执行的穷举搜索来确定。选择能够保证面片的无重叠插入的第一个位置,并且被面片覆盖的网格单元被标记为已使用。如果当前图像分辨率中没有空白空间可以容纳面片,则网格的高度H暂时加倍,并再次应用搜索。在该过程结束时,修剪H,以便适合所使用的网格单元。2D网格中的打包位置(其是每个面片的投影区域的边界框的左上角的位置)和每个面片的投影区域的大小被信令通知到经压缩的比特流中。

[0054] 占用图从打包块12输出到元数据压缩块6。占用图由二进制图组成,该二进制图为网格的每个单元指示它是属于空白空间,即是未被占用还是属于点云(即被占用),其中2D网格的一个单元将在图像生成过程中产生携带深度或颜色信息的像素。

[0055] 几何形状压缩块4还包括图像生成块14,图像生成块14利用在打包过程12期间计算的3D到2D映射(更具体地,每个连通分量的投影区域的打包位置和大小)来将点云的几何形状存储为图像。

[0056] 对应于为每个连通分量生成两个深度面片的事实,生成两个图像来存储点云的几何形状,这两个图像被称为DepthImg0和DepthImg1。

[0057] 参考图3进一步详细描述编码器2的图像生成块14。

[0058] 图像生成包括打包过程(深度D0打包),其中,根据每个面片的投影区域的打包位置,每个面片的D0值被复制到第二网格的相应部分。

[0059] 图像生成还包括增量\_深度(delta\_depth)确定,其中,对于depthImage0的每个被占用的像素,增量\_深度=D1-D0(D1和D0在同一投影线上),并且增量\_深度被存储在存储相应D0值的相同位置。

[0060] 图像生成还包括如下时间交织(time interleaving):(depthImage0\_0, depthImage1\_0, ..., depthImage0\_i, depthImage1\_i, ..., depthImage0\_n, depthImage1\_n),其中depthImage0\_i和depthImage1\_i是从点云帧i生成的。

[0061] 图像生成还包括填充过程20,其目的在于填充面片之间的空白空间,以便生成适于视频压缩的分段平滑图像。它使用简单的填充策略,如下所示:

[0062] • TxT(例如,16×16)像素的每个块被独立处理。

[0063] • 如果该块为空白(即其所有像素都属于空白空间),则通过以光栅顺序复制前一个TxT块的最后一行或最后一列来填充该块的像素。

[0064] • 如果该块已满(即没有空白像素),则不执行任何操作。

[0065] • 如果该块既有空白像素又有填充像素,则空白像素用其非空白邻近像素的平均值迭代填充。

[0066] 图像生成块14之后是深度视频编码块21,其中所生成的深度图像被存储为视频帧,并根据作为参数提供的HM配置,使用任何传统的视频编解码器(诸如HM16.16视频编解码器)进行压缩。

[0067] 在编码器2的元数据压缩块6中,以下元数据(称为每面片/连通分量元数据)被收

集,然后对每个连通分量编码如下:

[0068] • 投影平面的索引

[0069] ○对于平面 (1.0,0.0,0.0) 和 (-1.0,0.0,0.0),索引为0;

[0070] ○对于平面 (0.0,1.0,0.0) 和 (0.0,-1.0,0.0),索引为1;

[0071] ○对于平面 (0.0,0.0,1.0) 和 (0.0,0.0,-1.0),索引为2。

[0072] • 打包的几何形状/纹理图像上的2D边界框 (u0,v0,u1,v1);

[0073] • 以深度偏移 $\delta_0$ 、切向偏移 $s_0$ 和双切向偏移 $r_0$ 表示的连通分量的3D位置 (x0,y0,z0)。根据所选择的投影平面, ( $\delta_0,s_0,r_0$ ) 计算如下:

[0074] ○对于 $\delta_0=x_0,s_0=z_0$  and  $r_0=y_0$ ,索引为0

[0075] ○对于 $\delta_0=y_0,s_0=z_0$  and  $r_0=x_0$ ,索引为1

[0076] ○对于 $\delta_0=z_0,s_0=x_0$  and  $r_0=y_0$ ,索引为2。

[0077] 此外,为每个TxT块提供其相关联的面片/连通分量索引的映射信息(称为块到面片映射)按如下方式来生成并编码:

[0078] • 对于每个TxT块,令L是面片索引的有序列表,使得它们的2D边界框包含该块。列表中的顺序与用于编码2D边界框的顺序相同。L被称为候选面片的列表。

[0079] • 几何形状/纹理图像上的面片之间的空白空间被视为面片,并被分配特殊索引0,该索引被添加到所有块的候选面片列表中。

[0080] • 令I为占用当前TxT块的面片的索引,令J为I在L中的位置,代替对索引I进行显式编码,取而代之的是对其位置J进行算术编码,从而导致更好的压缩效率。

[0081] 此外,元数据压缩块6实施对从打包块12生成的占用图的编码22。

[0082] 占用图压缩利用上述块到面片映射信息,以便检测空白TxT块(即,面片索引为0的块)。在23处,剩余的块被编码如下。

[0083] 占用图可以以 $B_0 \times B_0$ 块的精度在22处编码,其中 $B_0$ 是用户定义参数。为了实现无损编码, $B_0$ 应该设置为1。在实践中, $B_0=2$ 或 $B_0=4$ 产生视觉上可接受的结果,同时显著减少编码占用图所需的比特数。

[0084] 块到面片映射编码23如下进行:

[0085] • 二进制值与属于同一TxT块的 $B_0 \times B_0$ 子块相关联。如果其至少包含未填充的像素,则值1与子块相关联,否则值0与子块相关联。如果子块具有值为1,则称其已满,否则为空白子块。

[0086] • 如果TxT块的所有子块都已满(即值为1),则该块被称为已满。否则,该块被称为未满。

[0087] • 为每个TxT块编码二进制信息,以指示它是否已满。

[0088] • 如果块未满,则指示满/空白子块的位置的额外信息被编码如下:

[0089] ○为子块定义不同的遍历顺序。图4显示四种被考虑的遍历顺序。

[0090] ○编码器选择遍历顺序之一并在比特流中显示地信令通知其索引。

[0091] ○通过使用游程编码策略来编码与子块相关联的二进制值。

[0092] ■初始子块的二进制值被编码。

[0093] ■检测到0和1的连续运行,同时遵循编码器所选择的遍历顺序。

[0094] ■对检测到运行的次数进行编码。

[0095] ■除最后一次运行外,每次运行的长度也被编码。

[0096] 从占用图编码22输出占用图流,从占用图(附图标记23)输出块到面片映射流。块到面片的索引和占用图被称为每个块的元数据。

[0097] 纹理压缩块8包括由诸如3D-HEVC (HEVC的扩展,其规范可以在ITU网站上找到,T建议,H系列,h265,https://www.itu.int/rec/T-REC-H.265-201802-I annex I)的传统解码器进行的深度视频解码,随后是几何形状重构过程24,其利用占用图信息以便检测重构的几何形状图像中的占用像素。通过平衡元数据和几何形状图像来计算与那些占用像素相关联的点的3D位置。更准确地说,令P为从占用像素(u,v)重构的点,并且令( $\delta_0, s_0, r_0$ )为它所属的连通分量的3D位置以及( $u_0, v_0, u_1, v_1$ )为相应深度面片的2D边界框。P可以用深度

[0098]  $\delta(u, v)$ 、切向偏移 $s(u, v)$ 和双切向偏移 $r(u, v)$ 表示如下:

[0099]  $\delta(u, v) = \delta_0 + g(u, v)$

[0100]  $s(u, v) = s_0 - u_0 + u$

[0101]  $r(u, v) = r_0 - v_0 + v,$

[0102] 其中 $g(u, v)$ 是重构的几何形状图像的亮度分量。

[0103] 点云几何形状重构过程24还为每个重构点输出像素的位置,该位置存储重构当前点的深度值,即( $i, u, v$ ),其中i表示深度图像,即DepthImg0或DepthImg1,并且( $u, v$ )是第i深度图像中像素的坐标。

[0104] 纹理压缩块8还包括3D颜色转移(transfer)块26,其中以最小化颜色和/或其它属性信息编码误差的方式来确定要为每个重构点编码的颜色和/或其它属性值。根据一个实施例,对于每个重构点,输入/原始点云中其最近点的颜色被指定为其要编码的颜色。

[0105] 3D颜色转移块26之后是一个或多个纹理图像的生成块(a texture image(s) generation block)28,一个或多个纹理图像的生成块28通过将每个重构点的要编码的颜色信息存储在与几何形状图像相同的位置(即, ( $i, u, v$ ))来生成要编码的纹理图像。

[0106] 一个或多个纹理图像的生成块28之后是使用诸如3D-HEVC的传统编码器的纹理视频编码(HEVC的扩展,其规范可以在ITU网站上找到,T建议,H系列,h265,https://www.itu.int/rec/T-REC-H.265-201802-I annex I)。

[0107] 从几何形状压缩块4生成的深度视频流、从元数据压缩流6生成的每块和每面片元数据流以及从纹理压缩块8生成的颜色视频流被复用在从编码器2输出的压缩流中。

[0108] 图5表示对应于编码器2的解码器30,用于重构点云。

[0109] 在解码器30处,从编码器2输出的压缩流被解复用成深度视频流、每块元数据流、每面片元数据流和颜色视频流。

[0110] 解码器30包括几何形状重构块31,几何形状重构块31包括由诸如3D-HEVC (HEVC的扩展,其规范可以在ITU网站上找到,T建议,H系列,h265,https://www.itu.int/rec/T-REC-H.265-201802-I annex I)的传统解码器进行的深度视频解码。

[0111] 图像重构块31还包括在图6中详细描述图像重构块32。该块32包括:

[0112] 从重构的深度视频流中时间去交织(time-deinterleaving)重构的depthImage0和重构的depthImage1;

[0113] 检索存储在重构的depthImage0中的重构的D0值;和

[0114] 使用重构的\_D0+重构的\_增量\_深度来重构D1值,其中重构的\_增量\_深度值被存储

在depthImage1中。

[0115] 图像重构块31还包括以与编码器2的点云几何形状重构块24相同的方式操作的点云几何形状重构块33。

[0116] 解码器30还包括元数据重构块34,元数据重构块34包括:

[0117] -占用图解码块35,用于解码占用图流。从该块35输出的占用图被传送到深度图像重构块32。

[0118] -块到面片映射解码块36,用于解码块到面片映射流。从该块36输出的块到面片映射被传送到点云几何形状重构块33。

[0119] -每面片元数据解码块37,用于解码每面片元数据流,从该块37输出的每面片元数据被传送到点云几何形状重构块33。

[0120] 解码器30还包括纹理重构过程,该过程包括由诸如3D-HEVC (HEVC的扩展,其规范可以在ITU网站上找到,T建议,H系列,h265,https://www.itu.int/rec/T-REC-H.265-201802-I annex I)的传统解码器进行的纹理视频解码阶段。

[0121] 纹理重构过程还包括纹理面片重构阶段,其中使用重构的每面片和每块元数据来重构纹理面片。

[0122] 此外,纹理重构过程包括点云纹理重构阶段35,其中使用重构每个点的像素的位置从重构的纹理图像中获取每个重构点的颜色。

[0123] 图1的编码器2的无损变型在文件“ISO/IEC JTC1/SC29/WG11MPEG2018/m42170, 2018年1月,韩国,广光州”中提出。

[0124] 如图7所示,这种无损编码器50的基本思想是构建特殊的连通分量,该分量由编码器2所构建的连通分量缺失的所有点组成。然后,在深度和纹理图像中打包对应于缺失点面片的相应的额外深度面片和额外纹理面片。

[0125] 输入点云中未投影到2D视频帧上的点称为缺失点。这些缺失点包括位于沿一条投影线具有最小和最大深度值的两个点之间的点、被认为是噪声点的点、属于被认为对于投影来说太小的分量的点。所有缺失点组成特殊的连通分量,称为missedPointsPatch。所有缺失点的(x,y,z)坐标被存储在一个2D面片中,该2D面片将单独或与其它深度值一起存储在一个3D视频帧中。

[0126] 对于图1和图7的有损和无损编码模式,除了打包的深度图像和纹理图像之外,压缩的比特流的重要部分包括辅助信息,该辅助信息有助于从打包的图像重构点云并且其包括:

[0127] • 每面片元数据,其包括法线轴、法线方向的位移、打包图像上的位置等,

[0128] • 块到面片映射,其指示打包图像中的每个块被哪个面片占用,以及

[0129] • 占用图,其指示像素是否被占用。

[0130] 在图7的无损点云编码中,占用图对于重构点云仍然是必要的,因为:

[0131] • 需要将具有等于0的要编码的值的占用像素与也具有等于0的值的未占用像素区分开,

[0132] • 填充打包图像中的未占用部分,以提高压缩性能。此操作称为图像填充。

[0133] 然而,占用图本身通常压缩起来非常昂贵,尤其是在无损模式下。在图7的无损编码器50中,编码占用图的成本比图像填充带来的增益大得多。

[0134] 2016年12月21日提交的欧洲专利申请第16306764.8号(公开为EP3340629)提出在要编码的深度值(即,码字的低2位)中保留脚部空间(foot-room)来存储占用图。这是通过左移一定数量的比特(例如左移2比特)来扩大要编码的原始深度数据来实现的。然而,随着扩大操作大幅度增加数据范围,深度图像的编码变得更加昂贵。

[0135] 对于无损编码模式,这种扩大操作是去除占用图编码的过度解决方案。此外,扩大不能解决区分值为“0”的占用像素和未占用像素的问题。

## 发明内容

[0136] 以下呈现本发明原理的简化概述,以提供对本发明原理的一些方面的基本理解。本概述不是对本原则的广泛概述。它并不旨在确定本原则的关键或重要要素。以下概述仅仅以简化的形式呈现了本原理的一些方面,作为下面提供的更详细描述的前言。

[0137] 总的来说,本公开提出了用于改善这种情况的解决方案。

[0138] 本公开提出通过在保持要编码的数据范围不变的情况下移除占用图编码来降低解码复杂度并节省无损点云编码的比特率。

[0139] 有利的是,在编码器和解码器侧都去除了占用图编码。

[0140] 有利的是,占用信息被嵌入到要编码的深度图像之一中,例如以下列方式:

[0141] -如果占用信息被嵌入到存储沿投影线的最小深度值的DepthImg0中,则修改深度面片生成步骤以确保要编码的所有深度值(更具体地是深度D0)大于0;

[0142] -如果占用信息被嵌入到DepthImg1中,当DepthImg1存储沿投影线的最小深度值和最大深度值之间的差增量\_深度时,要编码的所有增量\_深度值增加1;

[0143] -如果占用信息被嵌入到DepthImg1中,当DepthImg1存储沿投影线的最大深度值时,修改深度面片生成步骤以确保要编码的所有深度值(更具体地是深度D1)大于0。

[0144] 因此,本公开提供了一种方法,其包括:

[0145] -获得占用信息,该占用信息指示在投影平面上定义的2D规则网格的块是否与点云的至少一个点在所述投影平面上的正交投影的深度信息相关联;

[0146] -将所述占用信息嵌入到存储所述深度信息的图像中;以及

[0147] -编码所述图像。

[0148] 本公开还提供了一种设备,其包括:

[0149] -用于获得占用信息的部件,该占用信息指示在投影平面上定义的2D规则网格的块是否与深度信息相关联,该深度信息是点云的至少一个点在所述投影平面上的正交投影的结果;

[0150] -用于将所述占用信息嵌入到存储所述深度信息的图像中的部件;以及

[0151] -用于编码所述图像的部件。

[0152] 有利地,在存储所述深度信息的图像中嵌入所述占用信息包括偏移要存储的深度信息。

[0153] 根据实施例,该方法还包括发送信息以指示深度信息已经被偏移。

[0154] 根据实施例,所述图像存储距投影平面具有最小距离的点云的点的深度信息。

[0155] 根据实施例,所述图像存储距投影平面具有最大距离的点云的点的深度信息。

[0156] 根据实施例,所述图像存储深度信息,该深度信息表示距投影平面具有最大距离

的点云的点的深度信息和距离投影平面具有最小距离的点云的点的深度信息之间的差。

[0157] 本公开还提供了一种方法,其包括:

[0158] -解码存储点云的至少一个点在投影平面上的正交投影的深度信息的图像;

[0159] -从解码的深度图像推导出占用信息,所述占用信息指示在所述投影平面上定义的2D规则网格的块是否与所述解码的深度信息相关联。

[0160] 本公开还提供了一种设备,其包括:

[0161] -用于解码存储点云的至少一个点在投影平面上的正交投影的深度信息的图像的部件;以及

[0162] -从解码的深度图像推导出占用信息,所述占用信息指示在所述投影平面上定义的2D规则网格的块是否与所述解码的深度信息相关联。

[0163] 编码器/解码器模块有利地由编码器/解码器内的一个或多个处理器实施。

[0164] 根据本公开的方法可以在可编程装置上的软件中实施。它们可以单独用硬件、软件或它们的组合来实施。

[0165] 由于本发明可以以软件实施,所以本发明可以体现为计算机可读代码,用于提供给任何合适的载体介质上的可编程装置。载体介质可以包括存储介质,诸如软盘、CD-ROM、硬盘驱动器、磁带设备或固态存储设备等。

[0166] 因此,本公开提供了一种包括计算机可执行指令的计算机可读程序,以使计算机能够执行本发明的编码/解码方法。

## 附图说明

[0167] 在附图中,示出了本原理的示例。它显示:

[0168] -已经描述的图1示出了编码器;

[0169] -已经描述的图2示出了图1的编码器的深度面片创建过程;

[0170] -已经描述的图3示出了图1的编码器的图像生成过程;

[0171] -已经描述的图4示出了在图1的编码器中使用的子块遍历的示例;

[0172] -已经描述的图5示了解码器;

[0173] -已经描述的图6示出了图5的解码器的图像重构过程;

[0174] -图7示出了无损编码器,其中可以有利地实施本原理;

[0175] -图8示出了根据本公开实施例的图7的无损编码器的修改;

[0176] -图9示出了根据本公开实施例的图5的解码器的修改;

[0177] -图10示出了根据本公开第一实施例修改的几何形状压缩块;

[0178] -图11示出了根据本公开第二实施例修改的几何形状压缩块;

[0179] -图12示出了根据本公开第三实施例修改的几何形状压缩块;

[0180] -图13示出了根据本公开实施例的增强的增量深度 (enhanced-delta-depth, EDD) 码的示例;

[0181] -图14示出了根据本公开第四实施例修改的几何形状压缩块;

[0182] -图15示出了根据本公开第五实施例修改的几何形状压缩块;

[0183] -图16示出了根据本原理的示例的设备的架构的示例;

[0184] -图17示出了根据本原理的示例的通过通信网络通信的两个远程设备;和

[0185] -图18示出了根据本原理的示例的信号的语法。

[0186] 相似或相同的元件用相同的附图标记表示。

### 具体实施方式

[0187] 下文将参考附图更全面地描述本原理,在附图中示出了本原理的示例。然而,本原理可以以多种替代形式来体现,并且不应被解释为限于本文阐述的示例。因此,尽管本原理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,并将在此详细描述。然而,应该理解的是,并不旨在将本原理限制于所公开的特定形式,而是相反,本公开将覆盖落入由权利要求限定的本原理的精神和范围内的所有修改、等同物和替代物。

[0188] 本文使用的术语仅仅是为了描述特定的示例,而不是为了限制本发明的原理。如本文使用的,单数形式“一”、“一个”和“该”也旨在包括复数形式,除非上下文清楚地指示出其它形式。将进一步理解,当在本说明书中使用时,术语“包括”和/或“包含”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,则不存在中间元件。如本文使用的,术语“和/或”包括相关联的列出项目中的一个或多个的任意和所有组合,并且可以缩写为“/”。

[0189] 应当理解,尽管术语第一、第二等可以在本文中用于描述各种元件,但是这些元件不应该被这些术语限制。这些术语仅用于区分一个元件与另一元件。例如,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,而不脱离本原理的教导。

[0190] 虽然图中的一些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,通信可以在与所示箭头相反的方向上发生。

[0191] 关于框图 and 操作流程图描述了一些示例,其中每个框表示电路元件、模块或代码部分,其包括用于实施一个或多个指定逻辑功能的一个或多个可执行指令。还应该注意的,在其它实施方式中,在框中标注的一个或多个功能可以不按所标注的顺序发生。例如,根据所涉及的功能,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。

[0192] 本文提到的“根据示例”或“在示例中”意味着结合该示例描述的特定特征、结构或特性可以被包括在本原理的至少一个实施方式中。说明书中不同地方出现的短语“根据示例”或“在示例中”不一定都指同一示例,也不一定是与其它示例互斥的单独的或替代的示例。

[0193] 权利要求中出现的附图标记仅仅是为了说明,对权利要求的范围没有限制作用。

[0194] 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使用。

[0195] 本原理被描述用于编码/解码着色点云,但是扩展到编码/解码着色点云序列,因为该序列的每个着色点云如下所述被顺序编码/解码。

[0196] 根据实施例,已经描述的无损编码器50和解码器30用于实施本原理。

[0197] 如在表示根据本公开实施例修改的无损编码器51的图8中清楚地显示的,已经去除了占用图编码。相反,占用信息被嵌入到要编码的深度图像中,并且有利地在图像生成块14和元数据压缩块6之间创建链接52。

[0198] 对应地,如在表示解码器30的图9中显示的,在解码器30接收的压缩流中没有占用图流,并且占用图(附图标记35)的解码块被抑制。

[0199] 如参考示出图像生成块14的图3所述,在压缩的比特流中记录了两个点,这两个点生成沿每条投影线的最小和最大深度值。相应地,编码器51为每个点云帧编码两个深度图像DepthImg0和DepthImg1。相对最小深度值D0在DepthImg0中被信令通知,同时 $\Delta D = (D1 - D0)$ 在DepthImg1中被信令通知,其中相对最小深度值D0是沿每条投影线的绝对最小深度值和该投影线所属的连通分量的最小深度值之间的差,其中D1是相对最大深度值,即沿每条投影线的绝对最大深度值和该投影线所属的连通分量的最小深度值之间的差。

[0200] 值得注意的是,每个面片的最小深度值在每面片元数据中被信令通知。

[0201] 根据图10所示的第一实施例,占用信息被嵌入到DepthImg0中。

[0202] 如图3所示,通过使所有D0值都大于0并去除DepthImg0的填充步骤20,无损解码的D0值也可以揭示占用信息。

[0203] 为了使所有D0值都大于0,添加了深度面片验证块54。这个新块包括两个阶段。

[0204] 在第一阶段,如果需要,连通分量将被拆分。如果D0深度面片的数据范围大于最大可能范围 $[0, N-1]$ ,则相应连通分量被拆分成两个。 $N$ 的值取决于编码比特。如果使用8比特HEVC,则 $N=255$ 。如果使用10比特HEVC,则 $N=1023$ 。

[0205] 然后,在第二阶段,对于每个D0面片,根据以下过程来转换要编码的D0值或D0面片的最小深度值:

[0206] • 如果D0面片或相应连通分量的最小深度大于0,则当前最小面片深度将减少1。因此,要编码的所有相关D0值增加1。在这种情况下,在使用重构的D0值来重构点云几何形状之前,不需要改变重构的D1值。

[0207] • 如果D0面片或相应连通分量的最小深度已经为0,则当前最小面片深度保持不变。然后,当前D0面片的所有D0值增加1。在这种情况下,在重构点云几何形状之前,需要将重构的D0值减少1。

[0208] 如有必要,每面片(即每连通分量)增加1个比特,以指示重构的D0值在被使用前是否需要被校正。

[0209] 根据图11所示的第二实施例,当存储 $\Delta D$ 值时,占用信息被嵌入到DepthImg1中。

[0210] 根据该实施例,在56处,在打包DepthImg1之前,要编码的所有 $\Delta D$ 值增加1。由于占用像素的大量 $\Delta D$ 值为0,在编码之前添加了 $\Delta D$ 的转换阶段58。相应地,在解码器侧,在被用于重构点云之前,重构的 $\Delta D$ 值减少1。

[0211] 根据图12所示的第三实施例,当存储最大深度值D1时,占用信息被嵌入到DepthImg1中。

[0212] 通过使所有D1值大于0并去除DepthImg1的填充步骤,无损解码的D1值也可以揭示占用信息。

[0213] 为了使所有D1值都大于0,添加了深度面片验证块60。这个新块包括两个阶段。

[0214] 在第一阶段期间,如果需要,连通分量将被拆分。如果D1深度面片的数据范围大于最大可能范围 $[0, N-1]$ ,则相应连通分量被拆分成两个。 $N$ 的值取决于编码比特。如果使用8比特HEVC,则 $N=255$ 。如果使用10比特HEVC,则 $N=1023$ 。

[0215] 然后,在第二阶段期间,对于每个D1面片,要编码的D1值根据以下过程进行转换:

[0216] • 如果D1面片中要编码的最小D1大于0,则在使用重构的D1值重构点云几何形状之前,无需改变重构的D1值。

[0217] • 如果D1面片中要编码的最小D1为0,则当前D1面片的所有D1值增加1。在这种情况下,在重构点云几何形状之前,需要将重构的D1值减少1。

[0218] 每面片(即每连通分量)增加一个比特,以指示重构的D1值在被使用之前是否需要被校正。

[0219] 另一种可能性是,不存储DeltaD或D1,而在DepthImg1中存储增强的增量深度(EDD)码,增强的增量深度(EDD)码是将比特级联的码字,其中该比特针对沿投影线的深度D0和深度D1之间的每个位置,指示该位置是否被占用,如图13所示。

[0220] 根据图14所示的第四实施例,以与第一实施例中所描述的相同的方式将占用信息嵌入到DepthImg0中,并且DepthImg1存储这些EDD码。

[0221] 根据图15所示的第五实施例,当DepthImg1存储EDD码时,占用信息被嵌入到DepthImg1中。根据该实施例,在62处,打包DepthImg1之前,要编码的所有EDD码增加1。由于占用像素的大部分EDD码是0,所以在编码之前增加了EDD码的转换阶段64。对应地,在解码器侧,在被用于重构点云之前重构的EDD码减少1。

[0222] 有利地,与图7的编码器50相比,以下元数据被添加到由编码器51输出的比特流中:

[0223] • 指示占用图是否在比特流中被信令通知的一个比特被添加。例如,1意味着是,0意味着不是。

[0224] • 如果第一比特等于0,则指示哪个深度图像包含占用信息的一个比特被添加。例如,0意味着DepthImg0,1意味着DepthImg1。

[0225] • 指示DepthImg1中存储了哪个数据的几个比特被添加。例如,两个比特,其中0意味着DepthImg1存储D1,1意味着DepthImg1存储 $\Delta D = D1 - D0$ ,以及2意味着DepthImg1存储EDD码。

[0226] • 如果DepthImg0携带占用信息,则用于指示每面片一比特元数据是否与每面片元数据一起被信令通知的一个比特被添加,该每面片一比特元数据指示当前面片的重构的D0值在被用于重构点云几何形状之前是否需要被校正。

[0227] • 如果DepthImg1携带占用信息并且DepthImg1存储D1值,则用于指示当前面片的重构的D1值在被用于重构点云几何形状之前是否需要被校正的每面片一比特元数据被添加。

[0228] 对于整个比特流,上述新的元数据可以被信令通知一次,或者针对被压缩在一起的每个帧或每组帧被信令通知多次,这允许为每个帧或每组帧选择不同的编码方案。上述元数据的(多个)数据字段(多个)记录值可以直接输出到(多个)压缩的比特流,或者在输出到(多个)压缩的比特流之前被进一步熵编码。

[0229] 根据参考图10至图15描述的编码器51的实施例,修改图9的解码器30。

[0230] 对于第一实施例,其中占用信息被嵌入到DepthImg0中,该占用图根据下式生成:

$$[0231] \quad Occupied(u, v) = \begin{cases} \text{如果 } DepthImg0(u, v) == 0, \text{ 则为假} \\ \text{如果 } DepthImg0(u, v) > 0, \text{ 则为真} \end{cases}$$

[0232] 此外,必要时通过将D0值减少1来重构深度面片。

[0233] 对于第二实施例,其中占用信息被嵌入到DepthImg1中,该占用图根据下式生成:

$$[0234] \quad Occupied(u, v) = \begin{cases} \text{如果 } DepthImg1(u, v) == 0, \text{ 则为假} \\ \text{如果 } DepthImg1(u, v) > 0, \text{ 则为真} \end{cases}$$

[0235] 此外,通过将所有DeltaD值减少1来重构深度面片。

[0236] 对于第三实施例,其中占用率信息被嵌入到DepthImg1中,并且DepthImg1存储D1值,占用图根据下式生成:

$$[0237] \quad Occupied(u, v) = \begin{cases} \text{如果 } DepthImg1(u, v) == 0, \text{ 则为假} \\ \text{如果 } DepthImg1(u, v) > 0, \text{ 则为真} \end{cases}$$

[0238] 如果需要,通过将相关的D1值减少1来重构D1深度面片。

[0239] 对于第四实施例,其中占用信息被嵌入到DepthImg0中,占用图根据下式生成:

$$[0240] \quad Occupied(u, v) = \begin{cases} \text{如果 } DepthImg0(u, v) == 0, \text{ 则为假} \\ \text{如果 } DepthImg0(u, v) > 0, \text{ 则为真} \end{cases}$$

[0241] 此外,在必要时,通过将D0值减少1并且从EDD码来重构深度值进而重构深度面片。

[0242] 对于第五实施例,其中占用信息被嵌入到DepthImg1中,占用图根据下式生成:

$$[0243] \quad Occupied(u, v) = \begin{cases} \text{如果 } DepthImg1(u, v) == 0, \text{ 则为假} \\ \text{如果 } DepthImg1(u, v) > 0, \text{ 则为真} \end{cases}$$

[0244] 此外,通过将所有EDD码减少1来重构深度面片。

[0245] 在图1-图15中,模块是功能单元,它们可能与可区分的物理单元相关,也可能不相关。例如,这些模块或其中的一些模块可以在一个独特的组件或电路中集合在一起,或者构成软件的功能。相反,一些模块可能由独立的物理实体组成。与本原理兼容的装置可以使用纯硬件来实施,例如使用诸如ASIC、FPGA、VLSI(分别是专用集成电路、现场可编程门阵列、超大规模集成)的专用硬件来实施,或者由被嵌入到设备中的几个集成电子组件或者由硬件和软件组件的混合来实施。

[0246] 图16表示设备1100的示例性架构,设备1100可以被配置为实施关于图1-图15来描述的方法。

[0247] 设备1100包括由数据和地址总线1101链接在一起的以下元件:

[0248] -微处理器1102(或CPU),其例如是DSP(或数字信号处理器);

[0249] -ROM(只读存储器)1103;

[0250] -RAM(随机存取存储器)1104;

[0251] -I/O接口1105,用于从应用接收要传输的数据;以及

[0252] -电池1106。

[0253] 根据示例,电池1106在设备外部。在每个提及的存储器中,说明书中使用的词“寄存器”可以对应于小容量区域(一些比特)或非常大的区域(例如,整个程序或大量接收或解

码的数据)。ROM 1103至少包括程序和参数。ROM 1103可以存储用于执行根据本原理的技术的算法和指令。当被打开时,CPU 1102将程序上传到RAM中,并执行对应的指令。

[0254] RAM 1104包括在寄存器中由CPU 1102执行并在设备1100打开后上传的程序、寄存器中的输入数据、寄存器中的方法的不同状态的中间数据以及寄存器中用于执行该方法的其它变量。

[0255] 本文描述的实施方式可以在例如方法或过程、装置、软件程序、数据流或信号中实施。即使仅在单一实现形式的上下文中讨论(例如,仅作为方法或设备讨论),所讨论的特征的实施方式也可以以其它形式(例如,程序)实施。装置可以在例如适当的硬件、软件和固件中实施。这些方法可以在例如诸如处理器的装置中实施,处理器通常指处理设备,包括例如计算机、微处理器、集成电路或可编程逻辑设备。处理器还包括通信设备,例如计算机、手机、便携式/个人数字助理(“PDA”)和便于终端用户之间信息通信的其它设备。

[0256] 根据编码或编码器的示例,从源获得输入点云。例如,源属于包括以下各项的集合:

[0257] -本地存储器(1103或1104),例如视频存储器或RAM(或随机存取存储器)、闪存、ROM(或只读存储器)、硬盘;

[0258] -存储接口(1105),例如具有大容量存储装置、RAM、闪存、ROM、光盘或磁性支持的接口;

[0259] -通信接口(1105),例如有线接口(例如总线接口、广域网接口、局域网接口)或无线接口(例如IEEE 802.11接口或蓝牙接口);以及

[0260] -图像捕获电路(例如传感器,例如CCD(电荷耦合器件)或CMOS(互补金属氧化物半导体))。

[0261] 根据解码或解码器的示例,解码的点云被传送到目的地;具体地,目的地属于包括以下各项的集合:

[0262] -本地存储器(1103或1104),例如视频存储器或RAM、闪存、硬盘;

[0263] -存储装置接口(1105),例如具有大容量存储装置、RAM、闪存、ROM、光盘或磁性支持的接口;

[0264] -通信接口(1105),例如有线接口(例如总线接口(例如USB(或通用串行总线))、广域网接口、局域网接口、HDMI(高清晰度多媒体接口)接口)或无线接口(例如IEEE 802.11接口、WiFi或蓝牙接口);

[0265] -渲染设备;以及

[0266] -显示器。

[0267] 根据编码或编码器的示例,比特流(压缩流)被传送到目的地。作为示例,比特流被存储在本地或远程存储器中,例如视频存储器(1104)或RAM(1104)、硬盘(1103)。在变型中,比特流被传送到存储接口(1105),例如具有大容量存储器、闪存、ROM、光盘或磁支持的接口,和/或通过通信接口(1105)传输,例如点到点链路、通信总线、点到多点链路或广播网络的接口。

[0268] 根据解码或解码器的示例,从源获得比特流(压缩流)。示例性地,比特流从本地存储器读取,例如视频存储器(1104)、RAM(1104)、ROM(1103)、闪存(1103)或硬盘(1103)。在一个变型中,比特流从存储装置接口(1105)接收,例如具有大容量存储装置、RAM、ROM、闪存、

光盘或磁支持的接口,和/或通过通信接口(1105)接收,例如点到点链路、总线、点到多点链路或广播网络的接口。

[0269] 根据示例,被配置为实施关于图1和图9描述的编码方法的设备1100属于包括以下各项的集合:

[0270] -移动设备;

[0271] -具有3D捕获功能的智能手机或电视机;

[0272] -通信设备;

[0273] -游戏设备;

[0274] -平板(或平板电脑);

[0275] -膝上型计算机;

[0276] -静止图像照相机;

[0277] -摄像机;

[0278] -编码芯片;

[0279] -静止图像服务器;以及

[0280] -视频服务器(例如广播服务器、视频点播服务器或网络服务器)。

[0281] 根据示例,被配置为实施关于图3和图10描述的解码方法的设备1100属于包括以下各项的集合:

[0282] -移动设备;

[0283] -头戴式显示器(Head Mounted Display,HMD);

[0284] -(混合现实)智能眼镜;

[0285] -全息设备;

[0286] -通信设备;

[0287] -游戏设备;

[0288] -机顶盒;

[0289] -电视机;

[0290] -平板(或平板电脑);

[0291] -膝上型计算机;

[0292] -显示器;

[0293] -立体显示器;以及

[0294] -解码芯片。

[0295] 根据图17所示的本原理的示例,在通信网络NET上的两个远程设备A和B之间的传输环境中,设备A包括与存储器RAM和ROM相关的处理器,其被配置为实施关于图8描述的用于编码着色点云的方法,并且设备B包括与存储器RAM和ROM相关的处理器,其被配置为实施关于图9描述的用于解码的方法。

[0296] 根据示例,该网络是广播网络,适于将编码的着色点云从设备A广播到包括设备B的解码设备。

[0297] 旨在由设备A发送的信号携带比特流。

[0298] 图18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。每个被传输的分组P包括报头H和有效载荷PAYLOAD。

[0299] 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、从解码器输出的后处理器处理、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、HMD、智能眼镜以及用于处理图像或视频的任何其它设备或其它通信设备。应该清楚的是,该装备可以是移动的,甚至可以安装在移动车辆中。

[0300] 此外,这些方法可以通过由处理器执行的指令来实施,并且这样的指令(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的计算机可读程序产品的形式,并且其上包含可由计算机执行的计算机可读程序代码。本文使用的计算机可读存储介质被认为是非暂时性存储介质,其具有在其中存储信息的固有能力以及提供从中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,电子、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,虽然下面提供了可以应用本发明原理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那样,仅仅是说明性的而不是穷举性的列表: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(EPR0M或闪存);便携式紧凑光盘只读存储器(CD-ROM);光学存储设备;磁性存储设备;或前述的任何合适的组合。

[0301] 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。

[0302] 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、单独的应用或这两者的组合中找到。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(例如存储设备)的设备。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。

[0303] 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以被存储或传输的信息。该信息可以包括例如用于执行方法的指令,或者由所描述的实施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原理的所述示例的语法的规则作为数据携带,或者将由本原理的所述示例写入的实际语法值作为数据携带。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。格式化可以包括,例如,编码数据流和用编码的数据流调制载波。信号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传输。信号可以存储在处理器可读介质上。

[0304] 已经描述了许多方式。然而,应当理解,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、修改或去除以产生其它实施方式。此外,本领域普通技术人员将理解,可以用其它结构和过程来代替所公开的那些结构或过程,并且得到的实施方式将以至少基本相同的方式来执行至少基本相同的功能,以实现与所公开的实施方式至少基本相同的结果。因此,本申请预期了这些和其它实施方式。



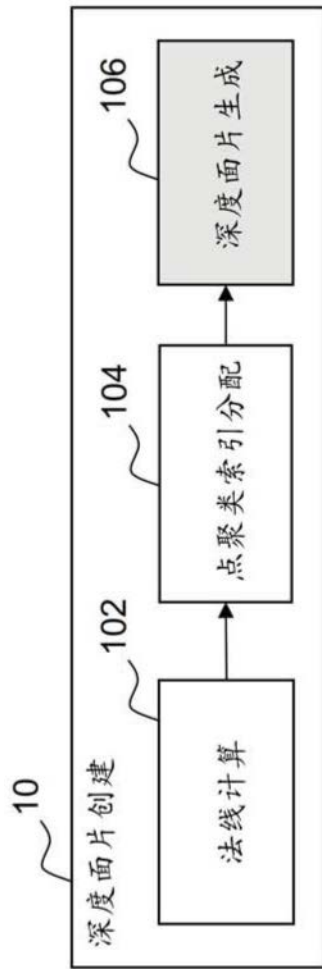


图2

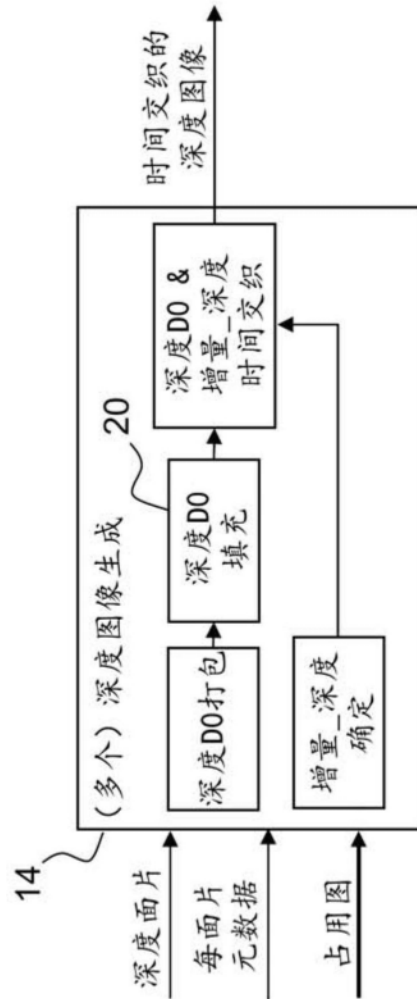


图3

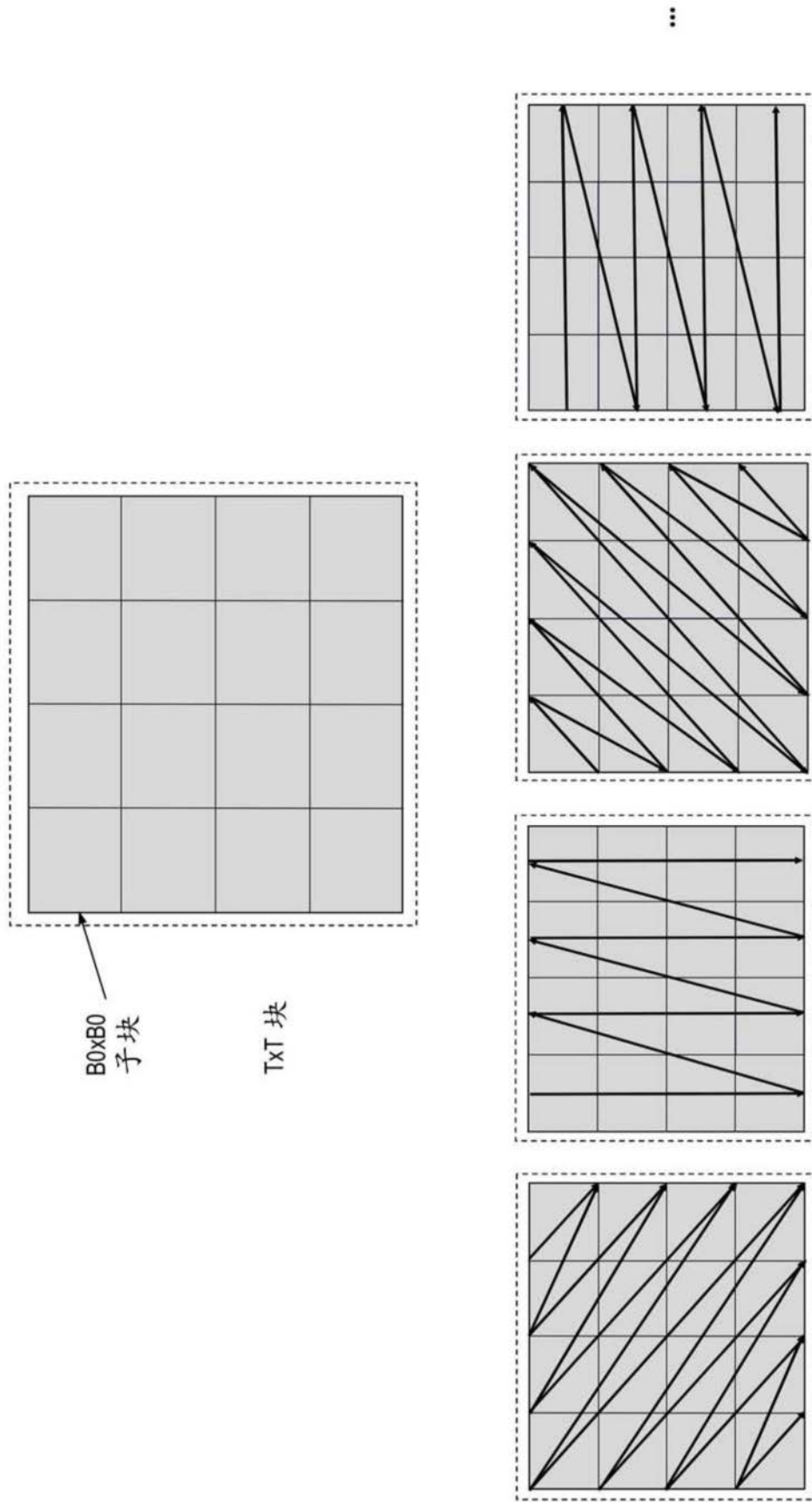
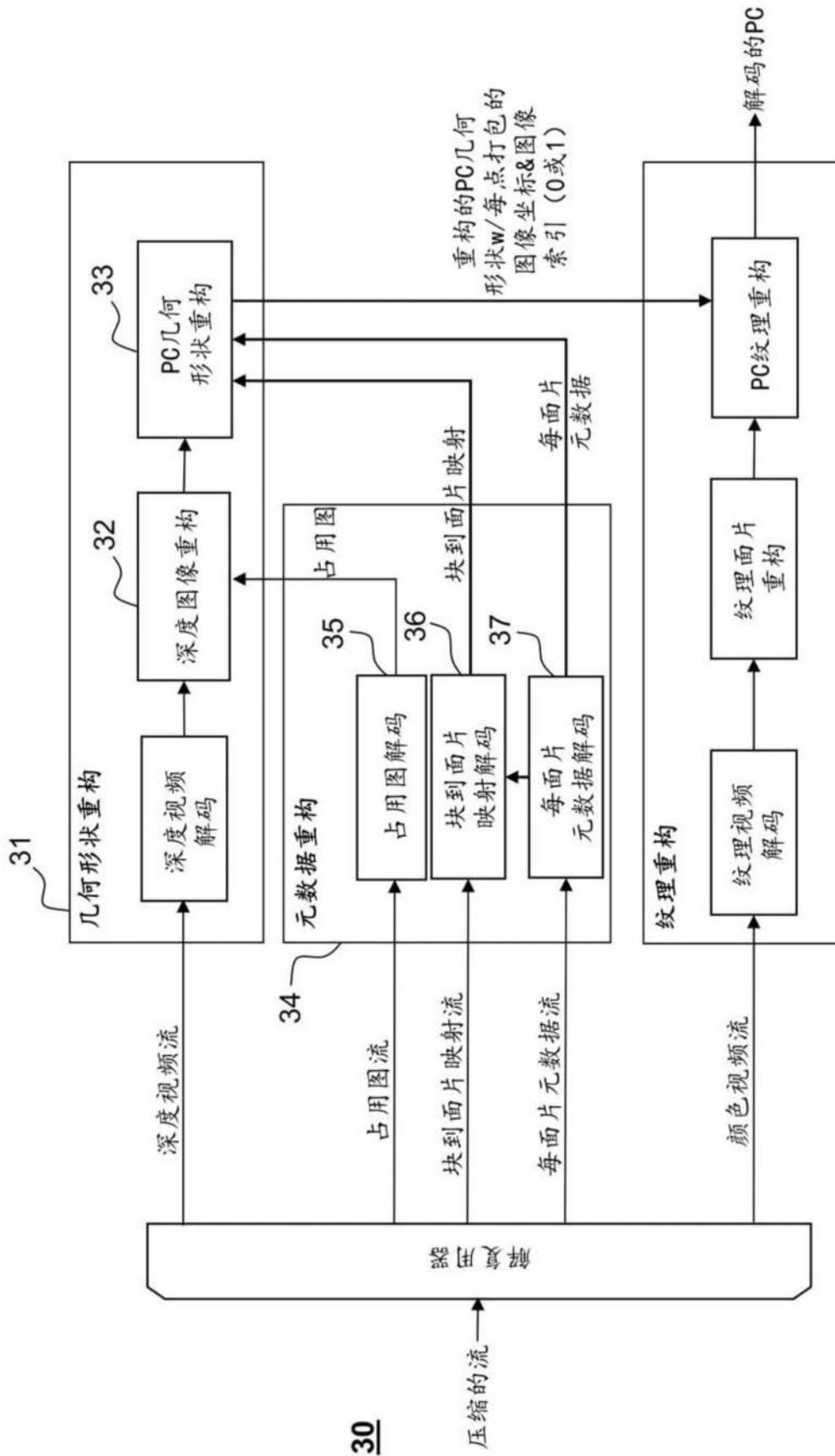


图4



30

图5

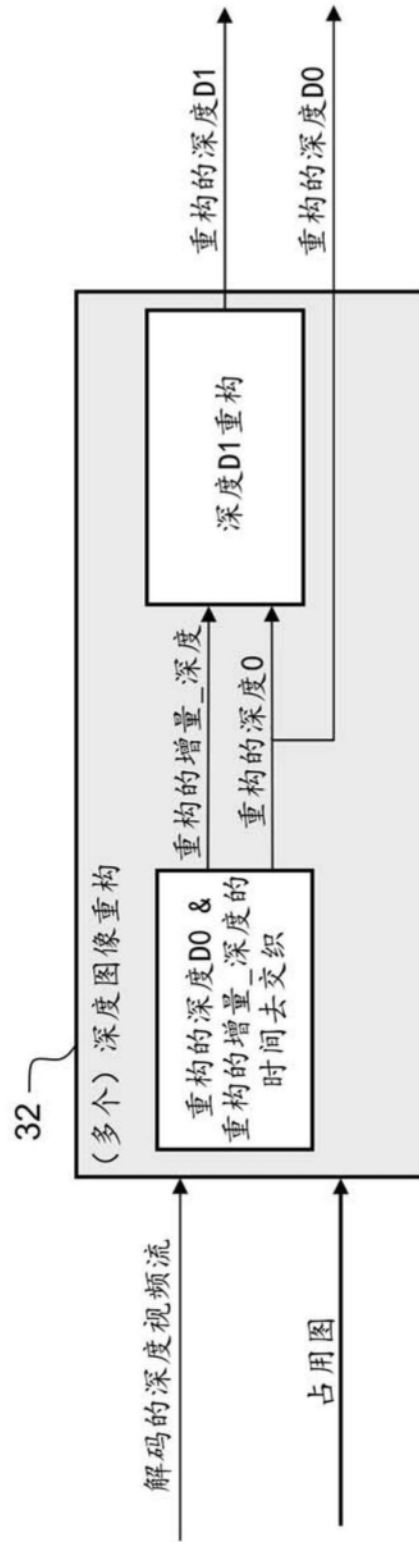
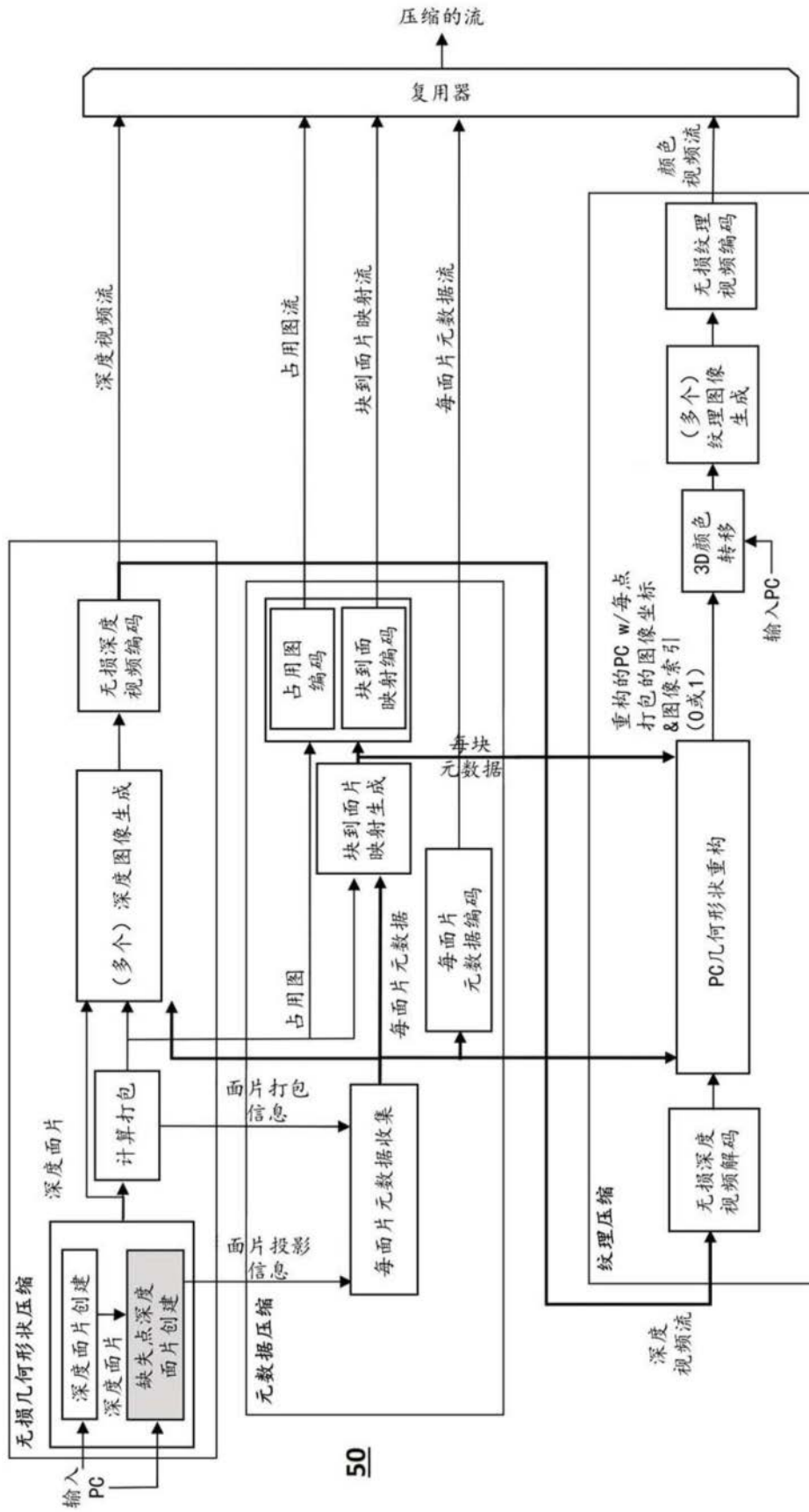


图6



50

图7

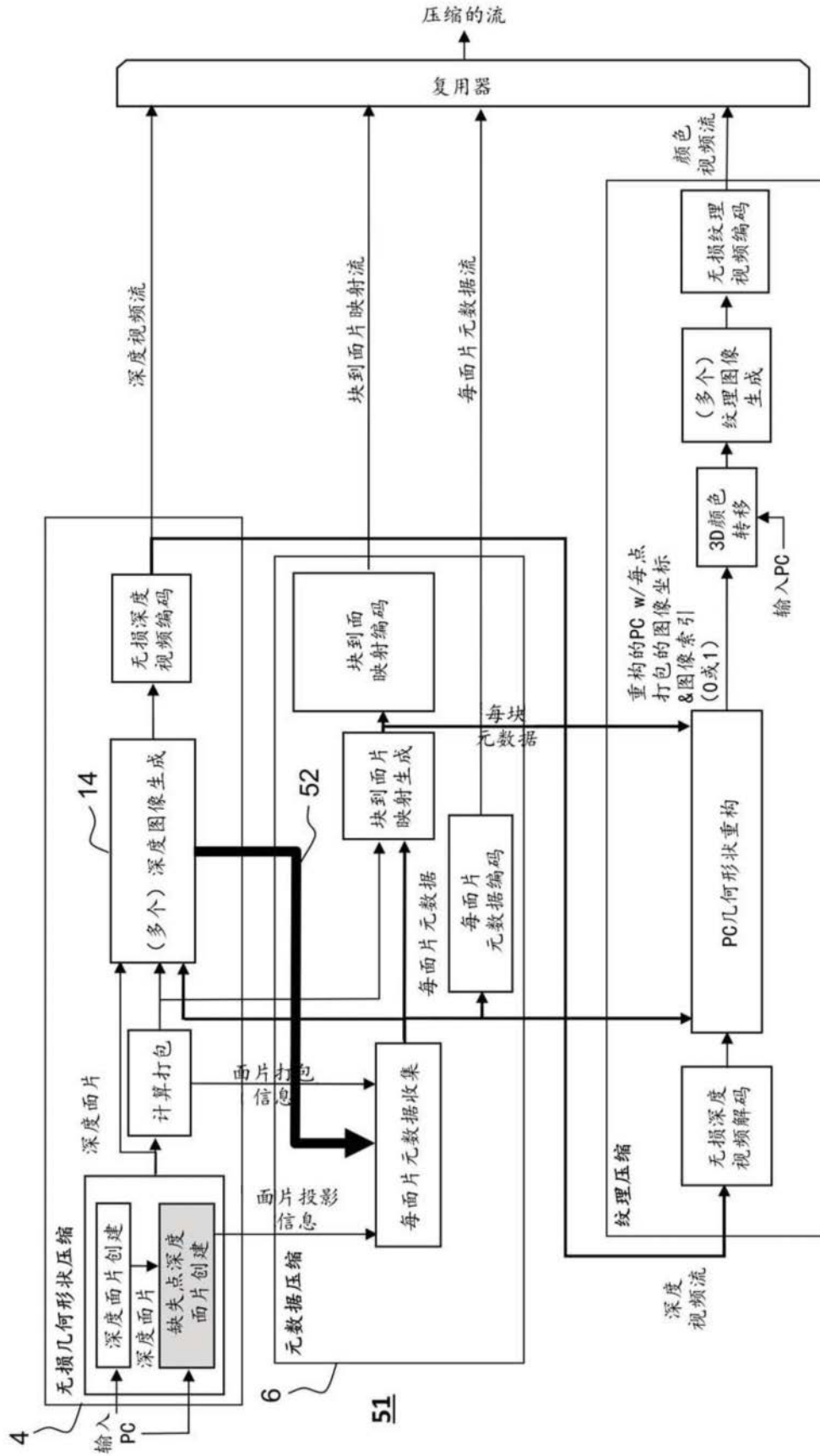


图8

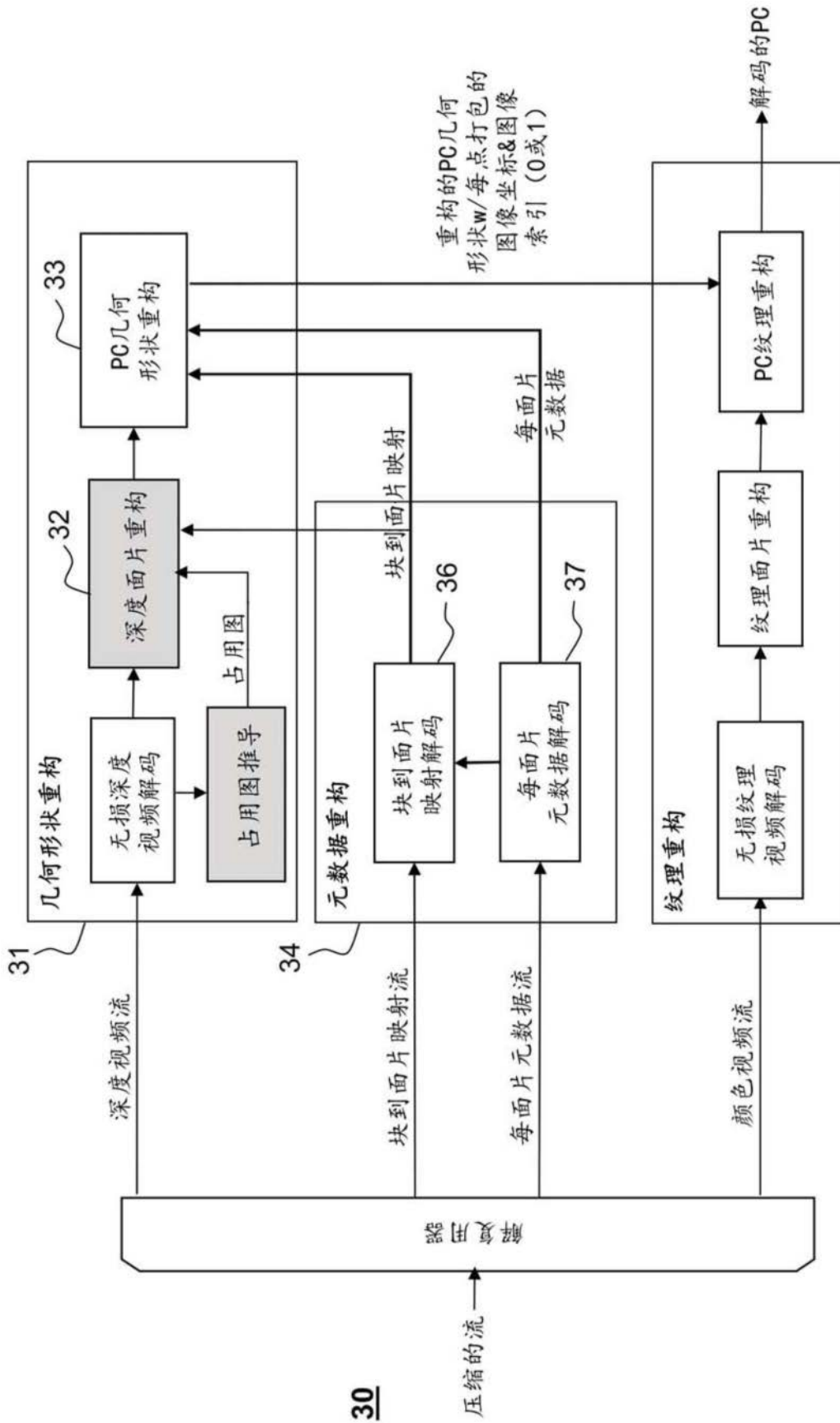


图9

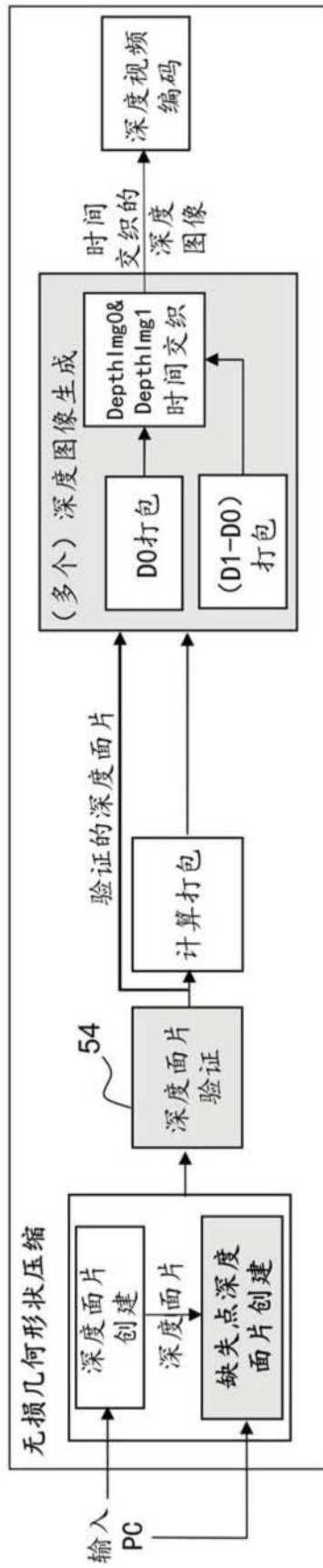


图10



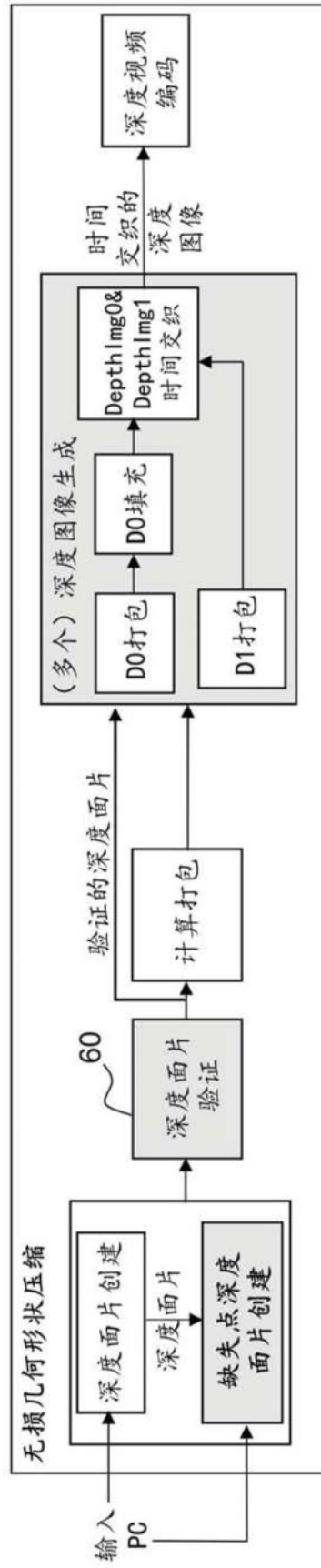


图12

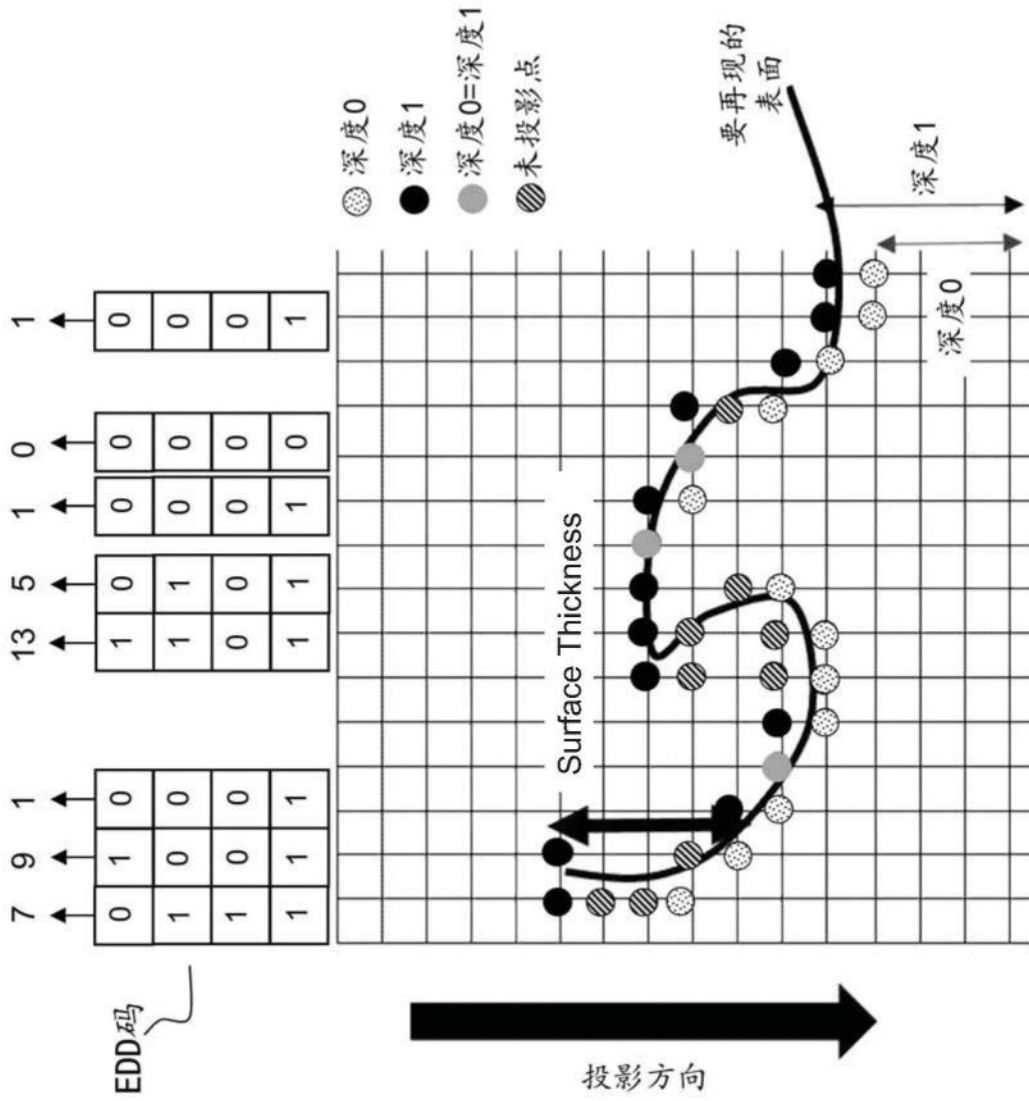


图13

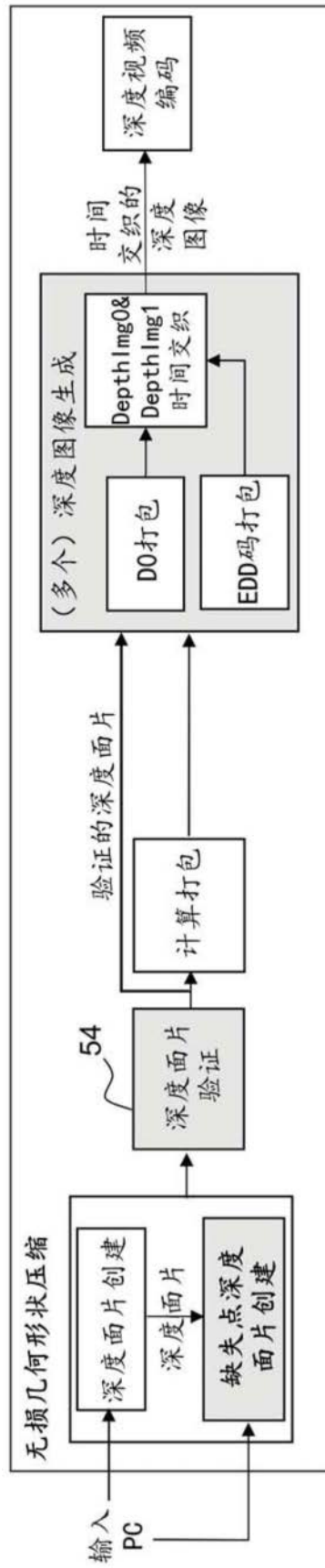


图14

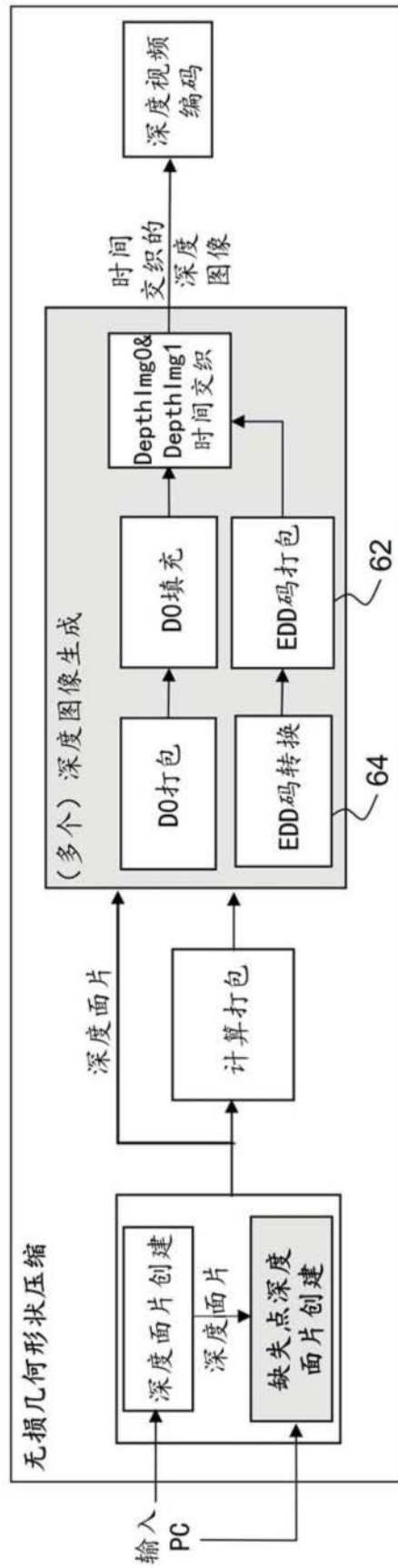


图15

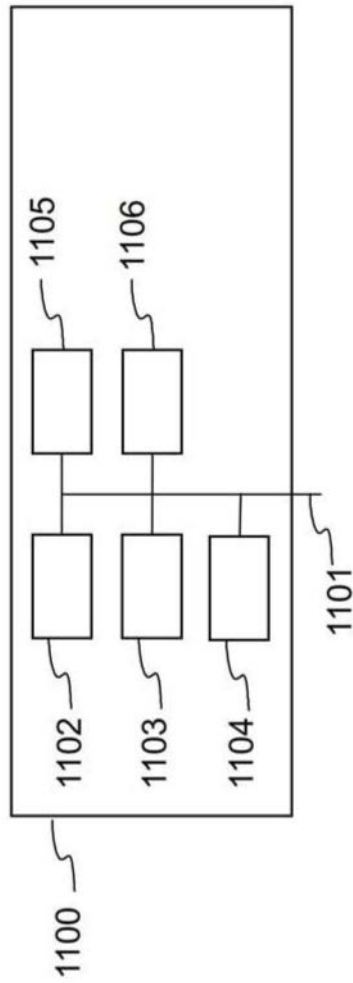


图16



图17



图18