



(12) 发明专利申请

(10) 申请公布号 CN 116708799 A

(43) 申请公布日 2023.09.05

(21) 申请号 202310659608.4

H04N 19/70 (2014.01)

(22) 申请日 2019.09.10

H04N 19/593 (2014.01)

(30) 优先权数据

62/731,693 2018.09.14 US

H04N 19/513 (2014.01)

H04N 19/51 (2014.01)

H04N 19/567 (2014.01)

(62) 分案原申请数据

H04N 19/147 (2014.01)

201980060363.5 2019.09.10

H04N 19/117 (2014.01)

(71) 申请人 华为技术有限公司

H04N 19/82 (2014.01)

地址 518129 广东省深圳市龙岗区坂田华为总部办公楼

H04N 19/96 (2014.01)

H04N 19/124 (2014.01)

(72) 发明人 王业奎 弗努·亨德里

弗莱德斯拉夫·扎克哈成科

(51) Int.Cl.

H04N 19/176 (2014.01)

H04N 19/44 (2014.01)

H04N 19/42 (2014.01)

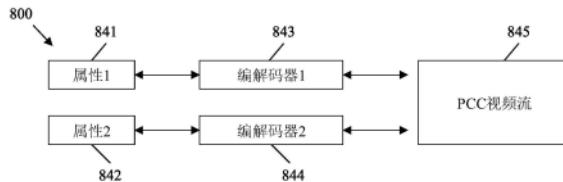
权利要求书2页 说明书29页 附图13页

(54) 发明名称

点云译码中属性层和指示的改进

(57) 摘要

本发明公开了一种视频译码机制。所述机制包括接收包括多个经译码的三维(three dimensional, 3D)内容帧序列的码流。所述多个经译码的3D内容帧序列表示多个3D内容属性。每个译码3D内容帧由一个或多个3D内容网络抽象层(network abstraction layer, NAL)单元表示。所述机制还包括对所述码流进行解析，以针对每个3D内容属性获取用于译码对应3D内容属性的多个视频编解码器中的一个视频编解码器的指示。所述机制还包括根据所述3D内容属性的所指示的视频编解码器对所述码流进行解码。



1.一种解码器实现的方法,其特征在于,所述方法包括:

所述解码器接收包括多个经译码的三维(three dimensional,3D)内容帧序列的码流,其中,所述多个经译码的3D内容帧序列表示多个3D内容属性,并且每个译码3D内容帧由一个或多个3D网络抽象层(network abstraction layer,NAL)单元表示;

所述解码器对所述码流进行解析,以针对每个3D内容属性获取用于译码对应3D内容属性的多个视频编解码器(codec)中的一个视频编解码器的指示;

所述解码器根据所述3D内容属性的所指示的视频编解码器对所述码流进行解码。

2.根据权利要求1所述的方法,其特征在于,3D内容帧的每个序列与包含序列级参数的序列级数据单元相关联,其中,所述序列级数据单元包括第一语法元素,所述第一语法元素表示第一属性由第一视频编解码器译码且表示第二属性由第二视频编解码器译码。

3.根据权利要求2所述的方法,其特征在于,所述第一语法元素是包含在所述码流中的帧组头中的identified_codec_for_attribute元素。

4.根据权利要求2所述的方法,其特征在于,所述第一属性组织成多个流,并且第二语法元素表示与所述第一属性相关联的所述码流的数据单元的流成员资格,所述第二语法元素是包含在所述码流中的帧组头中的num_streams_for_attribute元素。

5.根据权利要求2所述的方法,其特征在于,所述第一属性组织成多个层,并且第三语法元素表示与所述第一属性相关联的所述码流的数据单元的层成员资格。

6.根据权利要求5所述的方法,其特征在于,所述第三语法元素是包含在所述码流中的帧组头中的num_layers_for_attribute元素。

7.根据权利要求5所述的方法,其特征在于,第四语法元素表示所述多个层的第一层包含与不规则点云相关联的数据。

8.根据权利要求7所述的方法,其特征在于,所述第四语法元素是包含在所述码流中的帧组头中的regular_points_flag元素。

9.根据权利要求1所述的方法,其特征在于,所述码流被解码为经解码的3D内容帧序列,并且所述方法还包括向显示器转发所述经解码的3D内容帧序列以进行呈现。

10.一种在编码器中实现的方法,其特征在于,所述方法包括:

所述编码器利用多个视频编解码器(codec)将三维(three dimensional,3D)内容帧序列的多个3D内容属性编码到码流中,其中,每个译码3D内容帧由一个或多个3D内容网络抽象层(network abstraction layer,NAL)单元表示;

所述编码器针对每个3D内容属性对用于译码对应3D内容属性的所述视频编解码器中的一个视频编解码器的指示进行编码;

所述编码器向解码器发送所述码流。

11.根据权利要求10所述的方法,其特征在于,所述3D内容帧序列与包含序列级参数的序列级数据单元相关联,其中,所述序列级数据单元包括第一语法元素,所述第一语法元素表示第一属性由第一视频编解码器译码且表示第二属性由第二视频编解码器译码。

12.根据权利要求11所述的方法,其特征在于,所述第一语法元素是包含在所述码流中的帧组头中的identified_codec_for_attribute元素。

13.根据权利要求11所述的方法,其特征在于,所述第一属性组织成多个流,并且第二语法元素表示与所述第一属性相关联的所述码流的数据单元的流成员资格,所述第二语法

元素是包含在所述码流中的帧组头中的num_streams_for_attribute元素。

14. 根据权利要求11所述的方法,其特征在于,所述第一属性组织成多个层,并且第三语法元素表示与所述第一属性相关联的所述码流的数据单元的层成员资格。

15. 根据权利要求14所述的方法,其特征在于,所述第三语法元素是包含在所述码流中的帧组头中的num_layers_for_attribute元素。

16. 根据权利要求14所述的方法,其特征在于,第四语法元素表示所述多个层的第一层包含与不规则点云相关联的数据。

17. 根据权利要求16所述的方法,其特征在于,所述第四语法元素是包含在所述码流中的帧组头中的regular_points_flag元素。

18. 一种视频译码设备,其特征在于,包括:

处理器、与所述处理器耦合的接收器以及与所述处理器耦合的发送器,其中,所述处理器、接收器和发送器用于执行根据权利要求1至17中任一项所述的方法。

19. 一种非瞬时性计算机可读介质,其特征在于,包括供视频译码设备使用的计算机程序产品;所述计算机程序产品包括存储在所述非瞬时性计算机可读介质中的计算机可执行指令;在处理器执行所述计算机可执行指令时,使所述视频译码设备执行根据权利要求1至17中任一项所述的方法。

20. 一种编码器,其特征在于,包括:

第一属性编码模块和第二属性编码模块,用于利用多个视频编解码器(codec)将三维(three dimensional,3D)内容帧序列的多个3D内容属性编码到码流中,其中,每个译码3D内容帧由一个或多个3D内容网络抽象层(network abstraction layer,NAL)单元表示;

语法编码模块,用于针对每个3D内容属性对用于译码对应3D内容属性的所述视频编解码器中的一个视频编解码器的指示进行编码;

发送模块,用于向解码器发送所述码流。

21. 根据权利要求20所述的编码器,其特征在于,所述编码器还用于执行根据权利要求11至17中任一项所述的方法。

22. 一种解码器,其特征在于,包括:

接收模块,用于接收包括多个经译码的三维(three dimensional,3D)内容帧序列的码流,其中,所述多个经译码的3D内容帧序列表示多个3D内容属性,并且每个译码3D内容帧由一个或多个3D内容网络抽象层(network abstraction layer,NAL)单元表示;

解析模块,用于对所述码流进行解析,以针对每个3D内容属性获取用于译码对应3D内容属性的多个视频编解码器中的一个视频编解码器的指示;

解码模块,用于根据所述3D内容属性的所指示的视频编解码器对所述码流进行解码。

23. 根据权利要求22所述的解码器,其特征在于,所述解码器还用于执行根据权利要求2至9中任一项所述的方法。

点云译码中属性层和指示的改进

[0001] 相关申请的交叉引用

[0002] 本专利申请要求王业奎等人于2018年9月14日递交的发明名称为“用于点云译码的高级语法设计(High-Level Syntax Designs for Point Cloud Coding)”的第62/731,693号美国临时专利申请案的权益，该申请案通过引用的方式并入本文中。

技术领域

[0003] 本发明大体上涉及一种视频译码，具体地，涉及点云译码(point cloud coding, PCC)视频帧的视频属性的译码。

背景技术

[0004] 即使视频相对较短，也需要大量的视频数据来描述，当数据要在带宽容量有限的通信网络中进行流式传输或以其它方式传输时，这样可能会造成困难。因此，视频数据通常要先压缩，然后通过现代电信网络进行传输。由于内存资源可能有限，当在存储设备上存储视频时，该视频的大小也可能是一个问题。视频压缩设备通常在源侧使用软件和/或硬件对视频数据进行编码，然后进行传输或存储，从而减少表示数字视频图像所需的数据量。然后，由解码视频数据的视频解压缩设备在目的地侧接收压缩数据。在网络资源有限以及对更高视频质量的需求不断增长的情况下，需要改进压缩和解压缩技术，这些改进的技术在几乎不影响图像质量的情况下能够提高压缩比。

发明内容

[0005] 在一个实施例中，本发明包括一种视频解码器实现的方法。所述方法包括：接收器接收包括多个经译码的点云译码(point cloud coding, PCC)帧序列的码流，其中，所述多个经译码的PCC帧序列表示多个PCC属性，所述PCC属性包括几何形状、纹理以及反射率、透明度和法向中的一个或多个，并且每个译码PCC帧由一个或多个PCC网络抽象层(network abstraction layer, NAL)单元表示。所述方法还包括：处理器对所述码流进行解析，以针对每个PCC属性获取用于译码对应PCC属性的多个视频编解码器(codec)中的一个视频编解码器的指示。所述方法还包括：所述处理器根据所述PCC属性的所指示的视频编解码器对所述码流进行解码。在一些视频译码系统中，使用单个编解码器对整个PCC帧序列进行编码。PCC帧可以包括多个PCC属性。一些视频编解码器可以比其它视频编解码器更高效地编码一些PCC属性。本实施例允许不同的视频编解码器对相同的PCC帧序列编码不同的PCC属性。本实施例还提供各种语法元素，以便在序列中的PCC帧采用多个PCC属性(例如，三个或更多个)时支持译码灵活性。通过提供更多的属性，编码器可以编码更复杂的PCC帧。此外，解码器可以解码并因此显示更复杂的PCC帧。此外，通过允许对不同的属性使用不同的编解码器，可以根据编解码器选择优化译码过程。这可以减少编码器和解码器侧的处理器资源使用。此外，这可以提高压缩和译码效率，从而在编码器与解码器之间传输码流时减少内存使用和网络资源使用。

[0006] 可选地,在上述任一方面中,提供了所述方面的另一种实现方式:每个PCC帧序列与包含序列级参数的序列级数据单元相关联,其中,所述序列级数据单元包括第一语法元素,所述第一语法元素表示第一属性由第一视频编解码器译码且表示第二属性由第二视频编解码器译码。

[0007] 可选地,在上述任一方面中,提供了所述方面的另一种实现方式:所述第一语法元素是包含在所述码流中的帧组头中的identified_codec_for_attribute元素。

[0008] 可选地,在上述任一方面中,提供了所述方面的另一种实现方式:所述第一属性组织成多个流,并且第二语法元素表示与所述第一属性相关联的所述码流的数据单元的流成员资格。

[0009] 可选地,在上述任一方面中,提供了所述方面的另一种实现方式:所述第一属性组织成多个层,并且第三语法元素表示与所述第一属性相关联的所述码流的数据单元的层成员资格。

[0010] 可选地,在上述任一方面中,提供了所述方面的另一种实现方式:所述第二语法元素是包含在所述码流中的帧组头中的num_streams_for_attribute元素,并且所述第三语法元素是包含在所述码流中的所述帧组头中的num_layers_for_attribute元素。

[0011] 可选地,在上述任一方面中,提供了所述方面的另一种实现方式:第四语法元素表示所述多个层的第一层包含与不规则点云相关联的数据。

[0012] 可选地,在上述任一方面中,提供了所述方面的另一种实现方式:所述第四语法元素是包含在所述码流中的帧组头中的regular_points_flag元素。

[0013] 可选地,在上述任一方面中,提供了所述方面的另一种实现方式:所述码流被解码为经解码的PCC帧序列,并且所述方法还包括所述处理器向显示器转发所述经解码的PCC帧序列以进行呈现。

[0014] 在一个实施例中,本发明包括一种在视频编码器中实现的方法。所述方法包括:处理器利用多个编解码器将PCC帧序列的多个PCC属性编码到码流中,其中,所述多个PCC属性包括几何形状、纹理以及反射率、透明度和法向中的一个或多个,并且每个译码PCC帧由一个或多个PCC NAL单元表示。所述方法还包括:所述处理器针对每个PCC属性对用于译码对应PCC属性的所述视频编解码器中的一个视频编解码器的指示进行编码。所述方法还包括:发送器向解码器发送所述码流。在一些视频译码系统中,使用单个编解码器对整个PCC帧序列进行编码。PCC帧可以包括多个PCC属性。一些视频编解码器可以比其它视频编解码器更高效地编码一些PCC属性。本实施例允许不同的视频编解码器对相同的PCC帧序列编码不同的PCC属性。本实施例还提供各种语法元素,以便在序列中的PCC帧采用多个PCC属性(例如,三个或更多个)时支持译码灵活性。通过提供更多的属性,编码器可以编码更复杂的PCC帧。此外,解码器可以解码并因此显示更复杂的PCC帧。此外,通过允许对不同的属性使用不同的编解码器,可以根据编解码器选择优化译码过程。这可以减少编码器和解码器侧的处理器资源使用。此外,这可以提高压缩和译码效率,从而在编码器与解码器之间传输码流时减少内存使用和网络资源使用。

[0015] 可选地,在上述任一方面中,提供了所述方面的另一种实现方式:所述PCC帧序列与包含序列级参数的序列级数据单元相关联,其中,所述序列级数据单元包括第一语法元素,所述第一语法元素表示第一PCC属性由第一视频编解码器译码且表示第二PCC属性由第

二视频编解码器译码。

[0016] 可选地,在上述任一方面中,提供了所述方面的另一种实现方式:所述第一语法元素是包含在所述码流中的帧组头中的identified_codec_for_attribute元素。

[0017] 可选地,在上述任一方面中,提供了所述方面的另一种实现方式:所述第一属性组织成多个流,并且第二语法元素表示与所述第一属性相关联的所述码流的数据单元的流成员资格。

[0018] 可选地,在上述任一方面中,提供了所述方面的另一种实现方式:所述第一属性组织成多个层,并且第三语法元素表示与所述第一属性相关联的所述码流的数据单元的层成员资格。

[0019] 可选地,在上述任一方面中,提供了所述方面的另一种实现方式:所述第二语法元素是包含在所述码流中的帧组头中的num_streams_for_attribute元素,并且所述第三语法元素是包含在所述码流中的所述帧组头中的num_layers_for_attribute元素。

[0020] 可选地,在上述任一方面中,提供了所述方面的另一种实现方式:第四语法元素表示所述多个层的第一层包含与不规则点云相关联的数据。

[0021] 可选地,在上述任一方面中,提供了所述方面的另一种实现方式:所述第四语法元素是包含在所述码流中的帧组头中的regular_points_flag元素。

[0022] 在一个实施例中,本发明包括一种视频译码设备,所述视频译码设备包括:处理器、与所述处理器耦合的接收器以及与所述处理器耦合的发送器,其中,所述处理器、接收器和发送器用于执行根据上述任一方面所述的方法。

[0023] 在一个实施例中,本发明包括一种非瞬时性计算机可读介质,所述非瞬时性计算机可读介质包括供视频译码设备使用的计算机程序产品;所述计算机程序产品包括存储在所述非瞬时性计算机可读介质中的计算机可执行指令;在处理器执行所述计算机可执行指令时,使所述视频译码设备执行根据上述任一方面所述的方法。

[0024] 在一个实施例中,本发明包括一种编码器,所述编码器包括第一属性编码模块和第二属性编码模块,用于利用多个编解码器将PCC帧序列的多个PCC属性编码到码流中,其中,所述多个PCC属性包括几何形状、纹理以及反射率、透明度和法向中的一个或多个,并且每个译码PCC帧由一个或多个PCC NAL单元表示。所述编码器还包括语法编码模块,用于针对每个PCC属性对用于译码对应PCC属性的所述视频编解码器中的一个视频编解码器的指示进行编码。所述编码器还包括发送模块,用于向解码器发送所述码流。

[0025] 可选地,在上述任一方面中,提供了所述方面的另一种实现方式:所述编码器还用于执行根据上述任一方面所述的方法。

[0026] 在一个实施例中,本发明包括一种解码器,所述解码器包括接收模块,用于接收包括多个经译码的PCC帧序列的码流,其中,所述多个经译码的PCC帧序列表示多个PCC属性,所述PCC属性包括几何形状、纹理以及反射率、透明度和法向中的一个或多个,并且每个译码PCC帧由一个或多个PCC NAL单元表示。所述解码器还包括解析模块,用于对所述码流进行解析,以针对每个PCC属性获取用于译码对应PCC属性的多个视频编解码器中的一个视频编解码器的指示。所述解码器还包括解码模块,用于根据所述PCC属性的所指示的视频编解码器对所述码流进行解码。

[0027] 可选地,在上述任一方面中,提供了所述方面的另一种实现方式:所述解码器还用

于执行根据上述任一方面所述的方法。

[0028] 为了清楚起见,上述任一实施例可以与上述其它任一或多个实施例组合,在本发明范围内产生新实施例。

[0029] 根据以下具体实施方式结合附图和权利要求书能更清楚地理解这些和其它特征。

附图说明

[0030] 为了更全面地理解本发明,现在参考下文结合附图和详细说明进行的简要描述,其中,相同的附图标记表示相同的部件。

- [0031] 图1是对视频信号进行译码的示例性方法的流程图;
- [0032] 图2是用于视频译码的示例性编码和解码(编解码)系统的示意图;
- [0033] 图3是示例性视频编码器的示意图;
- [0034] 图4是示例性视频解码器的示意图;
- [0035] 图5是可以根据PCC机制进行译码的点云媒体的示例;
- [0036] 图6是点云媒体帧的数据分割和打包示例;
- [0037] 图7是具有扩展属性集的示例性PCC视频流的示意图;
- [0038] 图8是使用多个编解码器对PCC属性进行编码的示例性机制的示意图;
- [0039] 图9是属性层示例的示意图;
- [0040] 图10是属性流示例的示意图;
- [0041] 图11是使用多个编解码器对PCC视频序列进行编码的示例性方法的流程图;
- [0042] 图12是使用多个编解码器对PCC视频序列进行解码的示例性方法的流程图;
- [0043] 图13为示例性视频译码设备的示意图;
- [0044] 图14是用于使用多个编解码器对PCC视频序列进行译码的示例性系统的示意图;
- [0045] 图15是使用多个编解码器对PCC视频序列进行编码的另一示例性方法的流程图;
- [0046] 图16是使用多个编解码器对PCC视频序列进行解码的另一示例性方法的流程图;

具体实施方式

[0047] 首先应理解,尽管下文提供一个或多个实施例的说明性实现方式,但所公开的系统和/或方法可以使用任一数量的技术来实施,无论所述技术是当前已知还是现有的。本发明决不应限于下文所说明的说明性实施方案、附图和技术,包括本文所说明并描述的示例性设计和实施方案,而是可在所附权利要求书的范围以及其等效物的完整范围内修改。

[0048] 许多视频压缩技术可以用来减小视频文件,同时尽量减少数据丢失。例如,视频压缩技术可以包括执行空间(例如,帧内)预测和/或时间(例如,帧间)预测来减少或去除视频序列中的数据冗余。对于基于块的视频译码,视频条带(例如,视频图像或视频图像的一部分)可以分割成视频块,视频块也可以称为树块(treeblock)、编码树块(coding tree block,CTB)、编码树单元(coding tree unit,CTU)、编码单元(coding unit,CU)和/或编码节点。图像内的经帧内译码(I)条带中的视频块参照同一个图像内的相邻块中的参考样本使用空间预测进行译码。图像内的帧间译码(P或B)条带中的视频块可以使用针对同一图像内的相邻块中的参考样本的空间预测,或者使用针对其它参考图像内的参考样本的时间预测进行译码。图像可以称为帧,参考图像可以称为参考帧。空间预测或时间预测会产生表示

图像块的预测块。残差数据表示原始图像块与预测块之间的像素差。因此，帧间译码块根据运动矢量和残差数据进行编码，其中，运动矢量指向组成预测块的参考样本的块，残差数据表示译码块与预测块之间的差值。帧内编码块根据帧内译码模式和残差数据进行编码。为了进一步压缩，残差数据可以从像素域变换到变换域。从而产生可以被量化的残差变换系数。量化变换系数最初以二维阵列排列。可以扫描量化变换系数，以产生变换系数的一维矢量。可以应用熵编码以实现进一步压缩。下文更详细地论述了这类视频压缩技术。

[0049] 为了确保经编码视频能够被正确解码，视频根据对应的视频译码标准进行编码和解码。视频译码标准包括国际电联(international telecommunication union, ITU)标准化部门(international telecommunication union standardization sector, ITU-T) H.261、国际标准化组织/国际电工委员会(international organization for standardization/international electrotechnical commission, ISO/IEC)运动图像专家组(motion picture experts group, MPEG)-1第2部分、ITU-T H.262或ISO/IEC MPEG-2第2部分、ITU-T H.263、ISO/IEC MPEG-4第2部分、先进的视频译码(advanced video coding, AVC)(也称为ITU-TH.264或ISO/IEC MPEG-4第10部分)，以及高效视频译码(high efficiency video coding, HEVC)(也称为ITU-TH.265或MPEG-H第2部分)。AVC包括可分级的视频译码(scalable video coding, SVC)、多视图视频译码(multiview video coding, MVC)和多视图加深度视频译码(multiview video coding plus depth, MVC+D)以及三维(three dimensional, 3D) AVC(three dimensional AVC, 3D- AVC)等扩展。HEVC包括可分级HEVC(scalable HEVC, SHVC)、多视图HEVC(multiview HEVC, MV-HEVC)和3D HEVC(3D-HEVC)等扩展版。ITU-T和ISO/IEC的联合视频专家组(joint video experts team, JVET)已开始开发一种称为通用视频译码(versatile video coding, VVC)的视频译码标准。VVC包括在工作草案(working draft, WD)中，其中包括JVET-K1001-v4和JVET-K1002-v1。

[0050] PCC是用于对3D对象的视频进行编码的机制。点云是3D空间中的数据点集合。这些数据点包括确定在空间中的位置和颜色等的参数。点云可以用于各种应用中，例如实时3D沉浸式智真、具有交互式视差的内容虚拟现实(virtual reality, VR)观看、3D免费视点体育回放广播、地理信息系统、文化遗产、基于大规模3D动态地图的自主导航以及汽车应用。用于PCC的ISO/IEC MPEG编解码器可以对无损和/或有损压缩的点云数据进行操作，译码效率以及对网络环境具有鲁棒性较高。使用这种编解码器可以将点云作为一种计算机数据形式进行处理，存储在各种存储介质中，通过网络进行发送和接收，并在广播频道上分发。PCC译码环境分为PCC类别1、PCC类别2和PCC类别3。本发明针对PCC类别2，该PCC类别2与MPEG输出文件N17534和N17533相关。PCC类别2编解码器的设计旨在利用其它视频编解码器，通过将点云数据压缩为不同视频序列的集合来压缩动态点云的几何形状信息和纹理信息。例如，可以使用一个或多个视频编解码器生成并压缩两个视频序列，一个视频序列表示点云数据的几何形状信息，另一个视频序列表示纹理信息。还可以单独生成和压缩支持解释视频序列的其它元数据(例如，占用图和辅助片信息)。

[0051] PCC系统可以支持包含位置数据的几何形状PCC属性和包含颜色数据的纹理PCC属性。但是，一些视频应用可以包含其它类型的数据，例如反射率、透明度和法向量。使用某种编解码器可以比使用其它编解码器更有效地译码这些类型数据中的一些数据。但是，PCC系统可能要求整个PCC流由相同的编解码器编码，因此要求所有PCC属性由相同的编解码器编

码。此外, PCC属性可以划分成多个层。然后, 可以将这些层组合和/或译码到一个或多个PCC属性流中。例如, 可以根据时间交织译码方案对属性的层进行译码, 其中, 第一层在图像输出顺序为偶数值的PCC接入单元(access unit, AU)中译码, 第二层在图像输出顺序为奇数值的PCC AU中译码。由于每个属性可能有0到4个流并且此类流具有各种层组合, 因此正确识别流和层可能成为一个挑战。但是, PCC系统可能无法确定在给定PCC码流中译码或组合了多少个层。此外, PCC系统可能没有指示层组合的方式和/或指示这些层与PCC属性流之间的对应关系的机制。最后, 使用片对PCC视频数据进行译码。例如, 三维(three dimensional, 3D) PCC对象可以表示为二维(two dimensional, 2D) 片的集合。这允许PCC与视频编解码器结合使用, 用于编码2D视频帧。但是, 在一些情况下, 点云中的一些点可能无法被片捕获。例如, 3D空间中的孤立点可能难以作为片的一部分进行译码。在这种情况下, 唯一有意义的片是只包含单个点的一个像素×一个像素的片, 在具有许多此类点的情况下, 这会显著增加指示(signal)开销。相反, 可以使用不规则点云, 这是一种包含多个孤立点的特殊片。对于不规则点云片, 可以使用与其它片类型不同的方法来指示属性。但是, PCC系统可能无法指示PCC属性层携带不规则的点云点/片。

[0052] 本文公开了通过解决上述问题来改进PCC的机制。在一个实施例中, PCC系统可以使用不同的编解码器来译码不同的PCC属性。具体地, 可以使用单独的语法元素来识别每个属性的视频编解码器。在另一实施例中, PCC系统显式地指示经译码和/或组合以表示每个PCC属性流的层的数量。此外, PCC系统可以使用语法元素来指示用于译码和/或组合PCC属性流中的PCC属性层的方式。此外, PCC系统可以使用一个或多个语法元素来表示与对应PCC属性流的每个数据单元相关联的层的层索引。在又一实施例中, 可以对每个PCC属性层使用标志来表示PCC属性层是否携带任何不规则的点云点。此类实施例可以单独使用或结合使用。此外, 此类实施例允许PCC系统以解码器可识别并且因此解码器可解码的方式使用更复杂的译码机制。这些示例和其它示例在下面详细描述。

[0053] 图1为对视频信号进行译码的示例性操作方法100的流程图。具体地, 编码器对视频信号进行编码。在所述编码过程中, 采用各种机制来压缩所述视频信号, 以减小视频文件。文件较小, 压缩视频文件可以发送给用户, 同时降低相关带宽开销。然后, 解码器对压缩视频文件进行解码, 以重建原始视频信号, 从而向终端用户显示。所述解码过程通常是所述编码过程的逆过程, 使得所述解码器重建的视频信号与编码器侧的视频信号保持一致。

[0054] 在步骤101中, 将视频信号输入到编码器中。例如, 所述视频信号可以是存储在存储器中的未压缩视频文件。又例如, 视频文件可以由视频捕获设备(例如摄像机)捕获, 并且进行编码以支持视频的直播流传输。所述视频文件可以包括音频分量和视频分量。所述视频分量包括一系列图像帧。按顺序观看这些图像帧时, 给人以运动的视觉效果。这些帧包括以光线表示的像素, 在本文中称为亮度分量(或亮度样本), 还包括以颜色表示的像素, 称为色度分量(或颜色样本)。在一些示例中, 这些帧还可以包括深度值以支持三维观看。

[0055] 在步骤103中, 将视频分割成块。分割包括将每一帧中的像素细分成方块和/或矩形块, 以进行压缩。例如, 在高效视频译码(High EfficiencyVideo Coding, HEVC) (还称为H.265和MPEG-H第2部分)中, 可以先将帧分成编码树单元(coding tree unit, CTU), 这些CTU是预定义大小(例如, 64个像素×64个像素)的块。这些CTU包括亮度样本和色度样本。可以使用编码树将CTU分成块, 然后重复地对这些块进行细分, 直到获得支持进一步编码的配

置。例如,可以对帧的亮度分量进行细分,直到各个块包括相对均匀的亮度值。此外,可以对帧的色度分量进行细分,直到各个块包括相对均匀的色值。因此,分割机制根据视频帧的内容而不同。

[0056] 在步骤105中,采用各种压缩机制对在步骤103中分割的图像块进行压缩。例如,可以采用帧间预测和/或帧内预测。帧间预测利用了这样一个事实:常见场景下的对象往往出现在连续帧中。因此,表示参考帧中的对象的块不需要在相邻帧中进行重复表示。具体地,一个对象(例如一张桌子)可能在多个帧中保持固定位置。因此,该桌子表示一次,而且相邻帧可以重新参考该参考帧。可以采用模式匹配机制来匹配多个帧中的对象。此外,由于对象移动或摄像机移动等原因,可以在多个帧中表示移动对象。在一个特别的示例中,视频可以显示汽车跨屏幕在多个帧中移动。运动矢量可以用来表示这种移动。运动矢量是一个二维矢量,提供对象在一个帧中的坐标与该对象在参考帧中的坐标之间的偏移。因此,帧间预测可以将当前帧中的图像块编码为运动矢量集,这些运动矢量指示当前帧中的图像块与参考帧中的对应块之间的偏移。

[0057] 帧内预测用于对公共帧中的块进行编码。帧内预测利用了这样一个事实:亮度分量和色度分量往往聚集在一个帧中。例如,一棵树某个部分的一片绿色往往与类似的几片绿色相邻。帧内预测采用多种方向性预测模式(例如,HEVC中有33种)、平面模式和直流(direct current, DC)模式。这些方向性模式表示当前块的样本与对应方向上的相邻块的样本相似/相同。平面模式表示一行/列(例如平面)上的一系列块可以根据该行的边缘处的相邻块进行插值。平面模式实际上表示光线/颜色通过使用有数值变化的相对恒定的斜率在行/列之间进行平滑过渡。DC模式用于边界平滑,表示块与所有相邻块的样本的平均值相似/相同,这些相邻块与方向性预测模式的角度方向相关。因此,帧内预测块可以将图像块表示为各种关系预测模式值而不是实际值。此外,帧间预测块可以将图像块表示为运动矢量值而不是实际值。在任一种情况下,预测块在一些情况下可能无法准确表示图像块。所有差值都存储在残差块中。可以对残差块进行变换以进一步压缩文件。

[0058] 在步骤107中,可以使用各种滤波技术。在HEVC中,根据环内滤波方案使用滤波器。上文描述的基于块的预测可能会在解码器侧产生块状图像。此外,基于块的预测方案可以对块进行编码,然后重建经编码块,以后续用作参考块。环内滤波方案迭代地将噪声抑制滤波器、去块效应滤波器、自适应环路滤波器和样本自适应偏移(sample adaptive offset, SAO)滤波器应用于块/帧。这些滤波器减少了块伪影,这样可以准确地重建经编码文件。此外,这些滤波器减少了重建参考块中的伪影,使得伪影不太可能在后续根据重建参考块进行编码的块中产生其它伪影。

[0059] 一旦视频信号完成分割、压缩和滤波,则在步骤109中,将所得数据编码在码流中。所述码流包括上文描述的数据以及支持在解码器侧进行适当的视频信号重建所需要的任何指示数据。例如,这些数据可以包括分割部分数据、预测数据、残差块和提供译码指令给解码器的各种标志。所述码流可以存储在存储器中,以便在请求时发送给解码器。所述码流还可以广播和/或组播到多个解码器。所述码流的产生是一个迭代过程。因此,步骤101、步骤103、步骤105、步骤107和步骤109可以在多个帧和块中连续和/或同时执行。图1所示的顺序是为了清楚和便于描述而呈现的,并非旨在将视频译码过程限制于特定顺序。

[0060] 在步骤111中,所述解码器接收所述码流,开始解码过程。具体地,所述解码器采用

熵解码方案将所述码流转换为对应的语法数据和视频数据。在步骤111中，所述解码器使用所述码流中的语法数据来确定帧的分割部分。分割应该与步骤103中的块分割的结果匹配。下面描述在步骤111中采用的熵编码/解码。编码器在压缩过程中做出许多选择，例如根据一个或多个输入图像中的值的空间位置从若干个可能选择中选择块分割方案。指示确切的选择可能会使用大量的二元符号(bin)。本文所使用的“二元符号”是一个作为变量的二进制值(例如根据上下文可能变化的比特值)。熵编码使得编码器丢弃任何明显不适合特定情况的选项，从而留下一组可用选项。然后，为每个可用选项分配一个码字。码字的长度取决于可用选项的数量(例如，一个二元符号对应两个选项，两个二元符号对应三到四个选项，以此类推)。然后，编码器对所选选项的码字进行编码。该方案减小了码字，这是因为码字与预期的一样大，从而唯一地指示从可用选项的小子集中进行选择，而不是唯一地指示从所有可能选项的可能大集合中进行选择。然后，解码器通过以与编码器类似的方式确定该组可用选项对所述选项进行解码。通过确定该组可用选项，解码器可以读取码字，并确定编码器做出的选择。

[0061] 在步骤113中，所述解码器执行块解码。具体地，所述解码器进行逆变换，生成残差块。然后，所述解码器使用残差块和对应的预测块，根据分割来重建图像块。所述预测块可以包括编码器在步骤105中生成的帧内预测块和帧间预测块。然后，根据在步骤111中确定的分割数据将重建图像块放置在重建视频信号的帧中。还可以通过上文描述的熵编码在码流中指示用于步骤113的语法。

[0062] 在步骤115中，以类似于编码器侧的步骤107的方式对重建视频信号的帧执行滤波。例如，可以将噪声抑制滤波器、去块效应滤波器、自适应环路滤波器和SAO滤波器应用于帧，以去除块伪影。一旦对帧进行了滤波，则在步骤117中，可以将视频信号输出到显示器，以供终端用户观看。

[0063] 图2为用于视频译码的示例性编码和解码(编解码)系统200的示意图。具体地，编解码系统200的功能能够实现操作方法100。编解码系统200普遍适用于描述编码器和解码器中都使用的组件。编解码系统200接收视频信号并对视频信号进行分割，如结合操作方法100中的步骤101和步骤103所述，得到分割视频信号201。在充当编码器时，编解码系统200则将分割视频信号201压缩到经译码码流中，如结合方法100中的步骤105、步骤107和步骤109所述。在充当解码器时，编解码系统200从码流中生成输出视频信号，如结合操作方法100中的步骤111、步骤113、步骤115和步骤117所述。编解码系统200包括通用译码器控制组件211、变换缩放和量化组件213、帧内估计组件215、帧内预测组件217、运动补偿组件219、运动估计组件221、缩放和逆变换组件229、滤波器控制分析组件227、环内滤波器组件225、解码图像缓冲区组件223以及标头格式化和上下文自适应二进制算术译码(context adaptive binary arithmetic coding,CABAC)组件231。这些组件如图所示进行耦合。在图2中，黑线表示待编码/解码数据的移动，而虚线表示控制其它组件操作的控制数据的移动。编解码系统200中的组件都可以存在于编码器中。解码器可以包括编解码系统200中的组件的子集。例如，解码器可以包括帧内预测组件217、运动补偿组件219、缩放和逆变换组件229、环内滤波器组件225和解码图像缓冲区组件223。下面对这些组件进行描述。

[0064] 分割视频信号201为一个捕获到的视频序列，该视频序列已通过编码树分割成像素块。编码树采用各种划分模式将像素块细分成较小的像素块。然后，可以将这些块进一步

细分成较小的块。这些块可以称为编码树上的节点。较大的父节点被划分成较小的子节点。节点进行细分的次数称为节点/编码树的深度。在一些情况下，划分得到的块可以包括在编码单元(coding unit,CU)中。例如，CU可以是CTU的子部分，包括亮度块、一个或多个红色差色度(Cr)块和一个或多个蓝色差色度(Cb)块以及与CU对应的语法指令。划分模式可以包括用于将节点分割成不同形状(取决于所使用的划分模式)的两个、三个或四个子节点的二叉树(binary tree,BT)、三叉树(triple tree,TT)和四叉树(quad tree,QT)。将分割视频信号201转发给通用译码器控制组件211、变换缩放和量化组件213、帧内估计组件215、滤波器控制分析组件227和运动估计组件221，以进行压缩。

[0065] 通用译码器控制组件211用于根据应用约束条件决定将视频序列中的图像译码在码流中。例如，通用译码器控制组件211管理对码率/码流大小与重建质量的优化。可以根据存储空间/带宽可用性和图像分辨率请求做出这些决定。通用译码器控制组件211还根据传输速度来管理缓冲区利用率，以减轻缓存欠载和超载。为了解决这些问题，通用译码器控制组件211管理由其它组件进行的分割、预测和滤波。例如，通用译码器控制组件211可以动态增加压缩复杂度以提高分辨率和带宽利用率，或者降低压缩复杂度以降低分辨率和带宽利用率。因此，通用译码器控制组件211控制编解码系统200中的其它组件，平衡视频信号重建质量与码率。通用译码器控制组件211生成控制数据，这些控制数据用于控制其它组件的操作。还将控制数据转发给标头格式化和CABAC组件231，以编码在码流中，从而指示解码器进行解码使用的参数。

[0066] 还将分割视频信号201发送到运动估计组件221和运动补偿组件219，以进行帧间预测。可以将分割视频信号201中的帧或条带划分成多个视频块。运动估计组件221和运动补偿组件219相对于一个或多个参考帧中的一个或多个块对所接收到的视频块进行帧间预测译码，以提供时间预测。编解码系统200可以执行多个译码回合，以便为视频数据中的每个块选择合适的译码模式，等等。

[0067] 运动估计组件221和运动补偿组件219可以高度集成，但是出于概念目的，单独说明。由运动估计组件221执行的运动估计是生成运动矢量的过程，其中，这些运动矢量用于估计视频块的运动。例如，运动矢量可以表示经译码对象相对于预测块的位移。预测块是在像素差方面，与待译码块高度匹配的块。预测块还可以称为参考块。这种像素差可以通过绝对差异和(sum of absolute difference,SAD)、平方差异和(sum of square difference,SSD)或其它差异度量来确定。HEVC采用若干个经译码对象，包括CTU、编码树块(coding tree block,CTB)和CU。例如，可以将CTU分成CTB，然后将CTB分成CB，以包含在CU中。可以将CU编码为包括预测数据的预测单元(prediction unit,PU)和/或包括CU的变换残差数据的变换单元(transform unit,TU)。运动估计组件221进行作为率失真优化过程一部分的率失真分析，生成运动矢量、PU和TU。例如，运动估计组件221可以为当前块/帧确定多个参考块、多个运动矢量等，并且可以选择率失真特性最佳的参考块、运动矢量等。最佳率失真特性能够保持视频重建的质量(例如压缩造成的数据丢失量)和译码效率(例如最终编码的大小)之间的平衡。

[0068] 在一些示例中，编解码系统200可以计算存储在解码图像缓冲区组件223中的参考图像的子整数像素位置的值。例如，视频编解码系统200可以对参考图像的四分之一像素位置、八分之一像素位置或其它分数像素位置进行插值。因此，运动估计组件221可以相对于

整像素位置和分数像素位置执行运动搜索，并输出具有分数像素精度的运动矢量。运动估计组件221通过将PU的位置与参考图像的预测块的位置进行比较来计算帧间编码条带中视频块的PU的运动矢量。运动估计组件221将计算得到的运动矢量作为运动数据输出到标头格式化和CABAC组件231以进行编码，并作为运动数据输出到运动补偿组件219。

[0069] 由运动补偿组件219执行的运动补偿可以包括根据运动估计组件221所确定的运动矢量，获取或生成预测块。同样，在一些示例中，运动估计组件221和运动补偿组件219可以在功能上集成。在接收到当前视频块的PU的运动矢量之后，运动补偿组件219可以定位运动矢量指向的预测块。然后，将预测块的像素值从正在译码的当前视频块的像素值中减去，得到像素差，从而形成残差视频块。一般而言，运动估计组件221相对于亮度分量执行运动估计，运动补偿组件219将根据亮度分量计算得到的运动矢量用于色度分量和亮度分量。将所述预测块和所述残差块转发到所述变换缩放和量化组件213。

[0070] 还将分割视频信号201发送到帧内估计组件215和帧内预测组件217。与运动估计组件221和运动补偿组件219一样，帧内估计组件215和帧内预测组件217可以高度集成，但是出于概念目的，单独说明。帧内估计组件215和帧内预测组件217相对于当前帧中的各块对当前块进行帧内预测，以替代如上所述的由运动估计组件221和运动补偿组件219在各帧之间执行的帧间预测。具体地，帧内估计组件215确定帧内预测模式对当前块进行编码。在一些示例中，帧内估计组件215从多个测试的帧内预测模式中选择合适的帧内预测模式对当前块进行编码。然后，将选定的帧内预测模式转发给标头格式化和CABAC组件231，以进行编码。

[0071] 例如，帧内估计组件215对各种测试的帧内预测模式进行率失真分析，计算率失真值，并在测试的模式中选择率失真特性最佳的帧内预测模式。率失真分析一般用于确定经编码块与为了产生经编码块而进行编码的原始未编码块之间的失真(或误差)量，以及确定产生经编码块所用的码率(例如比特数)。帧内估计组件215根据各种经编码块的失真和速率计算比率，以确定得到块的最佳率失真值的帧内预测模式。另外，帧内估计组件215可以用于根据率失真优化(rate-distortion optimization, RDO)，使用深度建模模式(depth modeling mode, DMM)对深度图像中的深度块进行译码。

[0072] 在编码器上实现时，帧内预测组件217可以根据由帧内估计组件215确定的帧内预测模式，根据预测块生成残差块，或者在解码器上实现时，从码流中读取残差块。残差块包括预测块与原始块之间的差值，表示为矩阵。然后，将残差块转发给变换缩放和量化组件213。帧内估计组件215和帧内预测组件217可以对亮度分量和色度分量进行操作。

[0073] 变换缩放和量化组件213用于对残差块进一步压缩。变换缩放和量化组件213对残差块进行离散余弦变换(discrete cosine transform, DCT)、离散正弦变换(discrete sine transform, DST)等变换或者进行概念上类似的变换，从而产生包括残差变换系数值的视频块。还可以使用小波变换、整数变换、子频带变换或其它类型的变换。上述变换可以将残差信息从像素域转换到变换域(例如频域)。变换缩放和量化组件213还用于根据频率等对变换残差信息进行缩放。这种缩放涉及对残差信息应用缩放因子，以便在不同的粒度下对不同的频率信息进行量化，这可能会影响重建视频的最终视觉质量。变换缩放和量化组件213还用于对变换系数进行量化以进一步降低码率。该量化过程可以减小与部分或全部系数相关的位深度。量化的程度可以通过调整量化参数来修改。在一些示例中，变换缩放

和量化组件213随后可以对包括量化变换系数的矩阵执行扫描。将量化变换系数转发给标头格式化和CABAC组件231，以编码在码流中。

[0074] 缩放和逆变换组件229进行与变换缩放和量化组件213相反的操作以支持运动估计。缩放和逆变换组件229进行逆缩放、逆变换和/或反量化以重建像素域中的残差块，例如后续用作参考块。该参考块可会变为另一当前块的预测块。运动估计组件221和/或运动补偿组件219可以将残差块添加回对应的预测块来计算参考块，以用于对后续块/帧进行运动估计。将滤波器应用于重建参考块，以减少在缩放、量化和变换期间产生的伪影。当预测后续块时，这些伪影可能会使预测不准确（并产生额外的伪影）。

[0075] 滤波器控制分析组件227和环内滤波器组件225将滤波器应用于残差块和/或重建图像块。例如，可以将来自缩放和逆变换组件229的变换残差块与来自帧内预测组件217和/或运动补偿组件219的对应预测块组合以重建原始图像块。然后，可以将滤波器应用于重建图像块。在一些示例中，可以将滤波器应用于残差块。如同图2中的其它组件，滤波器控制分析组件227和环内滤波器组件225高度集成，可以一起实现，但出于概念目的进行单独描述。将应用于重建参考块的滤波器应用于特定空间区域，这些滤波器包括多个参数以调整使用这些滤波器的方式。滤波器控制分析组件227对重建参考块进行分析，以确定需要使用这些滤波器的位置并设置对应的参数。将这些数据转发给标头格式化和CABAC组件231，作为滤波器控制数据进行编码。环内滤波器组件225根据滤波器控制数据使用这些滤波器。这些滤波器可以包括去块效应滤波器、噪声抑制滤波器、SAO滤波器和自适应环路滤波器。这些滤波器可以根据示例应用于空域/像素域（例如重建像素块）或频域中。

[0076] 当作为编码器操作时，将经过滤波的重建图像块、残差块和/或预测块存储在解码图像缓冲区组件223中，后续用于如上所述的运动估计。当作为解码器操作时，解码图像缓冲区组件223存储经过滤波的重建块并将其转发给显示器，作为输出视频信号的一部分。解码图像缓冲区组件223可以是任何能够存储预测块、残差块和/或重建图像块的存储设备。

[0077] 标头格式化和CABAC组件231从编解码系统200中的各种组件接收数据，并将这些数据编码在经译码码流中，以发送给解码器。具体地，标头格式化和CABAC组件231生成各种标头以对控制数据（例如总体控制数据和滤波器控制数据）进行编码。此外，将预测数据（包括帧内预测和运动数据）以及以量化变换系数数据为形式的残差数据都编码在码流中。最终的码流包括解码器重建原始分割视频信号201所需要的所有信息。这些信息还可以包括帧内预测模式索引表（还称为码字映射表）、各种块的编码上下文的定义、最可能帧内预测模式的指示、分割信息的指示等。这些数据可以采用熵编码进行编码。例如，可以采用上下文自适应可变长度译码（context adaptive variable length coding, CAVLC）、CABAC、基于语法的上下文自适应二进制算术译码（syntax-based context-adaptive binary arithmetic coding, SBAC）、概率区间分割熵（probability interval partitioning entropy, PIPE）译码或其它熵编码技术对上述信息进行编码。在经过熵编码之后，可以将经译码码流发送到另一设备（例如视频解码器）或存档，以供后续发送或检索。

[0078] 图3为示例性视频编码器300的框图。视频编码器300可以用于实现编解码系统200的编码功能和/或执行操作方法100中的步骤101、步骤103、步骤105、步骤107和/或步骤109。编码器300对输入视频信号进行分割，得到基本上类似于分割视频信号201的分割视频信号301。然后，编码器300中的组件将分割视频信号301进行压缩并编码在码流中。

[0079] 具体地,将分割视频信号301转发给帧内预测组件317,以进行帧内预测。帧内预测组件317可以基本上类似于帧内估计组件215和帧内预测组件217。还将分割视频信号301转发给运动补偿组件321,以根据解码图像缓冲区组件323中的参考块进行帧间预测。运动补偿组件321可以基本上类似于运动估计组件221和运动补偿组件219。将来自帧内预测组件317和运动补偿组件321的预测块和残差块转发给变换和量化组件313,以对残差块进行变换和量化。变换和量化组件313可以基本上类似于变换缩放和量化组件213。将变换量化残差块和对应的预测块(连同相关的控制数据)转发给熵编码组件331,以译码在码流中。熵编码组件331可以基本上类似于标头格式化和CABAC组件231。

[0080] 还将变换量化残差块和/或对应的预测块从变换和量化组件313转发给逆变换和量化组件329,以重建为参考块供运动补偿组件321使用。逆变换和量化组件329可以基本上类似于缩放和逆变换组件229。根据示例,还将环内滤波器组件325中的环内滤波器应用于残差块和/或重建参考块。环内滤波器组件325可以基本上类似于滤波器控制分析组件227和环内滤波器组件225。环内滤波器组件325可以包括多个滤波器,如结合环内滤波器组件225所述。然后,将经过滤波的块存储在解码图像缓冲区组件323中,以作为参考块供运动补偿组件321使用。解码图像缓冲区组件323可以基本上类似于解码图像缓冲区组件223。

[0081] 图4为示例视频解码器400的框图。视频解码器400可以用于实现编解码系统200的解码功能和/或执行操作方法100中的步骤111、步骤113、步骤115和/或步骤117。解码器400从编码器300等接收码流,并根据所述码流生成重建输出视频信号,以向终端用户显示。

[0082] 所述码流由熵解码组件433接收。熵解码组件433用于执行熵解码方案,例如CAVLC、CABAC、SBAC、PIPE编码或其它熵编码技术。例如,熵解码组件433可以使用标头信息来提供上下文以解析编码在所述码流中作为码字的其它数据。上述解码信息包括对所述视频信号进行解码所需的任何信息,例如总体控制数据、滤波器控制数据、分割信息、运动数据、预测数据和残差块中的量化变换系数。将所述量化变换系数转发给逆变换和量化组件429,以重建为残差块。逆变换和量化组件429可以类似于逆变换和量化组件329。

[0083] 将所述重建残差块和/或预测块转发给帧内预测组件417,以根据帧内预测操作重建为图像块。帧内预测组件417可以类似于帧内估计组件215和帧内预测组件217。具体地,帧内预测组件417使用预测模式来定位帧中的参考块,并将残差块应用于上述结果以重建帧内预测图像块。上述重建帧内预测图像块和/或残差块以及对应的帧间预测数据通过环内滤波器组件425转发给解码图像缓冲区组件423。解码图像缓冲区组件423和环内滤波器组件425可以基本上分别类似于解码图像缓冲区组件223和环内滤波器组件225。环内滤波器组件425对上述重建图像块、残差块和/或预测块进行滤波。这些信息存储在解码图像缓冲区组件423中。将来自解码图像缓冲区组件423的重建图像块转发给运动补偿组件421,以进行帧间预测。运动补偿组件421可以基本上类似于运动估计组件221和/或运动补偿组件219。具体地,运动补偿组件421使用参考块的运动矢量来生成预测块,并将残差块应用于上述结果以重建图像块。所得到的重建块还可以通过环内滤波器组件425转发给解码图像缓冲区组件423。解码图像缓冲区组件423继续存储其它重建图像块。这些重建图像块可以通过分割信息重建为帧。这些帧还可以放置在一个序列中。所述序列作为重建输出视频信号输出到显示器。

[0084] 图5是可以根据PCC机制译码的点云媒体500的示例。点云是空间中的数据点集合。

点云可以由3D扫描仪产生,这些3D扫描仪测量3D扫描仪周围的对象外表面上的大量点。点云可以按照几何形状属性、纹理属性、反射率属性、透明度属性、法向属性等来描述。作为方法100的一部分,每个属性可以由视频编解码系统200、编码器300和/或解码器400等编解码器译码。具体地,PCC帧的每个属性可以在编码端单独编码,并在解码端解码和重新组合以重新创建PCC帧。

[0085] 点云媒体500包括三个边界框502、504和506。边界框502、504和506中的每个边界框表示来自当前帧的一部分或一段3D图像。虽然边界框502、504和506包含人的3D图像,但是在实际应用中,边界框中可以包括其它对象。每个边界框502、504和506包括x轴、y轴和z轴,分别表示3D图像在x、y和z方向上占用的像素数。例如,x轴和y轴描绘约400个像素(例如,约0-400个像素),而z轴描绘约1000个像素(例如,约0-1000个像素)。

[0086] 边界框502、504和506中的每个边界框包含一个或多个片508,在图5中由立方体或框表示。每个片508包含边界框502、504或506中的一个边界框内的整体对象的一部分,并且可以由片大小信息描述或表示。例如,片信息可以包括描述片508在边界框502、504或506内的位置的二维(two-dimensional, 2D)和/或三维(three-dimensional, 3D)坐标。片信息还可以包括其它参数。例如,片信息可以包括normalAxis等参数,所述参数从参考片信息继承得到,用于当前片信息。也就是说,可以从参考帧的片信息中继承一个或多个参数,用于当前帧的片信息。此外,当前帧可以继承参考帧的一个或多个元数据部分(例如,片旋转、尺度参数、材料标识符等)。片508在本文中可互换地称为3D片或片数据单元。可以在每个边界框502、504或506中生成片508列表,并且将该片列表按降序从最大片到最小片存储在片缓冲区中。然后,所述片可以由编码器编码和/或由解码器解码。

[0087] 片508可以描述点云媒体500的各种属性。具体地,每个像素在x轴、y轴和z轴上的位置是该像素的几何形状。可以对包含当前帧中所有像素的位置的片508进行译码以捕获点云媒体500的当前帧的几何形状属性。此外,每个像素可以包括红、蓝、绿(red, blue, green, RGB)和/或亮度和色度(YUV)谱中的颜色值。可以对包含当前帧中所有像素的颜色的片508进行译码以捕获点云媒体500的当前帧的纹理属性。

[0088] 此外,每个像素可以(或者可以不)包括一定反射率。反射率是从像素投射到相邻像素的光(例如,有色光)的量。有光泽对象具有高反射率,因此将它们对应像素的光/颜色扩散到附近的其它像素上。同时,哑光对象反射率很小或没有反射率,可能不会影响附近像素的颜色/光水平。可以对包含当前帧中所有像素的反射率的片508进行译码以捕获点云媒体500的当前帧的反射率属性。一些像素也可以部分透明至完全透明(例如,玻璃、透明塑料等)。透明度是邻近像素的可以通过当前像素的光/颜色的量。可以对包含当前帧中所有像素的透明度水平的片508进行译码以捕获点云媒体500的当前帧的透明度属性。此外,点云媒体的点可以生成表面。表面可以与法向量相关联,法向量是垂直于表面的向量。法向量可以用于描述对象运动和/或相互作用。因此,在一些情况下,用户可能希望对表面的法向量进行编码以支持附加功能。可以对包含当前帧中一个或多个表面的法向量的片508进行译码以捕获点云媒体500的当前帧的法向属性。

[0089] 根据示例,几何形状、纹理、反射率、透明度和法向属性可以包含描述点云媒体500中的一些或所有数据点的数据。例如,反射率、透明度和法向属性是可选的,因此,即使在相同码流中,对于一些点云媒体500示例,这些属性可以单独或组合出现,而对于其它点云媒

体500示例，则不会出现。因此，片508的数量以及属性的数量可以根据拍摄的主题、视频设置等而在帧与帧和视频与视频之间变化。

[0090] 图6是点云媒体帧600的数据分割和打包示例。具体地，图6的示例描绘了点云媒体500的片508的2D表示。点云媒体帧600包括与视频序列中的当前帧对应的边界框602。与图5的边界框502、504和506是3D不同，边界框602是2D的。如图所示，边界框602包含许多片604。片604在本文中可互换地称为2D片或片数据单元。总体而言，图6中的片604是图5的边界框504中的图像表示。因此，图5中的边界框504中的3D图像通过片604投射到边界框602上。边界框602中不包含片604的部分称为空的空间606。空的空间606也可以称为空隙空间、空样本等。

[0091] 考虑到以上内容，需要说明的是，基于视频的点云压缩 (point cloud compression, PCC) 编解码方案是基于3D点云数据(例如，图5的片508)分割成2D投影片(例如，图6的片604)。实际上，上述译码方法或过程可以有益地用于各种类型的技术，例如沉浸式六自由度 (six degrees of freedom, 6DoF)、动态增强现实/虚拟现实 (augmented reality/virtual reality, AR/VR) 对象、文化遗产、图形信息系统 (graphic information system, GIS)、计算机辅助设计 (computer aided design, CAD)、自主导航等。

[0092] 边界框(例如，边界框602)内的每个片(例如，图6的片604中的一个)的位置可以仅由片的大小来确定。例如，图6中最大的片604首先从左上角(0,0)开始投射到边界框602上。在将最大的片604投射到边界框602上之后，将下一个最大的片604投射到(也就是填充到)边界框602上，如此直到最小的片604投射到边界框602上。同样，此过程仅考虑每个片604的大小。在一些情况下，较小的片604可以占据各较大片之间的空间，并且位置可能最终比较大片604更靠近边界框602的左上角。在编码期间，可以对每个相关属性重复该过程，直到帧中每个属性的片被编码到一个或多个对应的属性流中。然后，用于重新创建单个帧的属性流中的数据单元组可以存储在PCC AU中的码流中。在解码器侧，从PCC AU获取这些属性流并对这些属性流进行解码以创建片604。然后，可以组合此类片604以重新创建PCC媒体。因此，点云媒体帧600可以由视频编解码系统200、编码器300和/或解码器400等编解码器译码，作为方法100的一部分来压缩点云媒体500以进行传输。

[0093] 图7是具有扩展属性集的示例性PCC视频流700的示意图；例如，当根据方法100通过使用视频编解码系统200、编码器300和/或解码器400等对来自点云媒体500的点云媒体帧600进行编码时，可以创建PCC视频流700。

[0094] PCC视频流700包括PCC AU 710的序列。PCC AU 710包括足以重建单个PCC帧的数据。数据位于NAL单元720中的PCCAU 710中。NAL单元720是分组大小的数据容器。例如，单个NAL单元720的尺寸通常设计为允许进行简单的网络传输。NAL单元720可以包含表示NAL单元720类型的标头和包含相关视频数据的有效载荷。PCC视频流700针对扩展属性集进行设计，因此包含若干属性特定的NAL单元720。

[0095] PCC视频流700可以包括帧组 (group offrame, GOF) 头721、辅助信息帧722、占用图帧723、几何形状NAL单元724、纹理NAL单元725、反射率NAL单元726、透明度NAL单元727和法向NAL单元728，其中的每个单元都是NAL单元720的一种类型。GOF头721包含描述对应PCC AU 710、与对应PCC AU 710相关的帧和/或PCC AU 710中的其它NAL单元720的各种语法元素。根据示例，PCC AU 710可以包含单个GOF头721，也可以不包含GOF头721。辅助信息帧722

可以包含与帧相关的元数据,例如与用于编码属性的片相关的信息。占用图帧723可以包含与帧相关的其它元数据,例如表示帧中被数据占用的区域与帧中空区域的占用图。其余NAL单元720包含PCCAU 710的属性数据。具体地,几何形状NAL单元724、纹理NAL单元725、反射率NAL单元726、透明度NAL单元727和法向NAL单元728分别包含几何形状属性、纹理属性、反射率属性、透明度属性和法向属性。

[0096] 如上所述,属性可以组织为流。例如,每个属性可能存在0到4个流。流可以包括PCC视频数据的逻辑上分离的部分。例如,不同对象的属性可以编码到相同类型的多个属性流(例如,第一3D边界框的第一几何形状流、第二3D边界框的第二属性流等)中。在另一示例中,与不同帧相关的属性可以编码到多个属性流(例如,偶数帧的透明度属性流和奇数帧的透明度属性流)中。在又一示例中,片可以放置在层中以表示3D对象。因此,单独的层可以包括在单独的流中(例如,顶层的第一纹理属性流、第二层的第二纹理属性流等)。不管示例如何,PCC AU 710可以包含对应属性的零个、一个、或多个NAL单元。

[0097] 本发明可以提高译码各种属性的灵活性(例如,如包括在几何形状NAL单元724、纹理NAL单元725、反射率NAL单元726、透明度NAL单元727中,和/或法向NAL单元728中)。在一示例中,可以使用不同的编解码器对不同的PCC属性进行译码。在一个具体示例中,可以使用第一编解码器将PCC视频的几何形状译码到几何形状NAL单元724中,而使用第二编解码器将PCC视频的反射率译码到反射NAL单元726中。在另一示例中,在对PCC视频进行译码时可以使用多达五个编解码器(例如,每个属性一个编解码器)。然后,用于一个或多个属性的一个或多个编解码器可以指示为PCC视频流700中(例如GOF头721中)的一个或多个语法元素。

[0098] 此外,如上所述,PCC属性可以使用层和/或流的多种组合。因此,一个或多个语法元素(例如,在GOF头721中)可用于在编码每个属性时指示编码器使用的层和/或流组合,以便允许解码器在解码时确定每个属性的层和/或流组合。此外,语法元素(例如,在GOF头721中)可以用于指示用于译码和/或组合PCC属性流中的PCC属性层的模式。此外,一个或多个语法元素(例如,在GOF头721中)可用于表示与PCC属性流对应的每个NAL单元720相关联的层的层索引。例如,GOF头721可用于指示与几何形状属性相关的层和流的数量、此类层和流的排列方式,以及每个几何形状NAL单元724的层索引,使得解码器可以在解码PCC帧时将每个几何形状NAL单元724分配给合适的层。

[0099] 最后,标志(例如,在GOF头721中)可以表示任何PCC属性层是否包含任何不规则的点云点。不规则点云是不与邻近数据点邻接的一个或多个数据点的集合,因此不能在片604等2D片中表示。代替地,这些点表示为不规则点云片的一部分,该不规则点云片包含与这些不规则点云点相关的坐标和/或平移参数。由于不规则点云使用不同于2D片的数据结构表示,因此该标志使解码器可以正确识别不规则点云的存在,并选择合适的机制来解码这些数据。

[0100] 下面是实现上述方面的一个示例性机制。定义:视频NAL单元是PccNalUnitType等于GMTRY_NALU、TEXTURE_NALU、REFLECT_NALU、TRANSP_NALU或NORMAL_NALU的PCC NAL单元。

[0101] 码流格式:该子句指定NAL单元流与字节流之间的关系,该NAL单元流和字节流中的任一个称为码流。码流可以具有两种格式:NAL单元流格式或字节流格式。NAL单元流格式在概念上是更基本的类型,包括称为PCC NAL单元的语法结构序列。该序列是按解码顺序排

序的。NAL单元流中的PCC NAL单元的解码顺序(和内容)受到约束。通过按解码顺序对NAL单元进行排序，并在每个NAL单元前添加起始码前缀和零个或多个零值字节以形成字节流，可以由NAL单元流格式构建字节流格式。通过在该字节流中搜索唯一的起始码前缀模式的位置，可以从字节流格式中提取NAL单元流格式。字节流格式与HEVC和AVC中使用的格式类似。

[0102] PCC NAL单元头语法可以如下表1所述实现。

[0103] 表1PCC NAL单元头语法

		描述符
[0104]	pcc_nal_unit_header()	
	forbidden_zero_bit	f(1)
	pcc_nal_unit_type_plus1	u(5)
	pcc_stream_id	u(2)
	}	

[0105] 帧组头原始字节序列有效载荷(raw byte sequence payload, RBSP)语法可以如下表2所述实现。

[0106] 表2帧组头RBSP语法

		描述符
	group_of_frames_header_rbsp()	
	num_attributes	u(5)
	if(num_attributes > 1)	
	attributes_first_ordering_flag	u(1)
	for(i = 0; i < num_attributes; i++) {	
	attribute_type[i]	u(5)
	identified_codec_for_attribute[i]	u(8)
	num_streams_for_attribute[i]	u(2)
	num_layers_for_attribute[i]	u(4)
	for(j = 0; j < num_streams_for_attribute[i]; j++) {	
[0107]	max_attribute_layer_idx[i][j]	u(4)
	if((j == 0 && max_attribute_layer_idx[i][j] > 0)	
	(j > 0 && max_attribute_layer_idx[i][j] >	
	(max_attribute_layer_id[i][j - 1] + 1)))	
	attribute_layers_combination_mode[i][j]	u(4)
	}	
	for(j = 0; j < num_layers_for_attribute[i]; j++)	
	regular_points_flag[i][j]	u(1)
	}	
	byte_alignment()	
	pcc_profile_level()	
	frame_width	u(16)
	frame_height	u(16)
	occupancy_resolution	u(8)
	radius_to_smoothing	u(8)
	neighbor_count_smoothing	u(8)
	radius2_boundary_detection	u(8)
	threshold_smoothing	u(8)
	lossless_geometry	u(8)

		描述符
[0108]	group_of_frames_header_rbsp()	
	lossless_texture	u(8)
	no_attributes	u(8)
	lossless_geometry_444	u(8)
	absolute_d1_coding	u(8)
	binary_arithmetic_coding	u(8)
	gof_header_extension_flag	u(1)
	if(gof_header_extension_flag)	
	while(more_rbsp_data())	
	gof_header_extension_data_flag	u(1)
	rbsp_trailing_bits()	
	}	

[0109] PCC档次和级别语法可以如下表3所述实现。

[0110] 表3PCC档次语法

		描述符
[0111]	pcc_profile_level()	
	pcc_profile_idc	u(5)
	pcc_pl_reserved_zero_19bits	u(19)
	pcc_level_idc	u(8)
	for(i = 0; i < num_attributes; i++)	
	if(identified_codec_for_attribute[i] == CODEC_HEVC)	
	hevc_ptl_12bytes_attribute[i]	u(96)
	else if(identified_codec_for_attribute[i] == CODEC_AVC)	
	avc_pl_3bytes_attribute[i]	u(24)
	}	

[0112] PCC NAL单元头语义可以如下实现: forbidden_zero_bit可以设置为零。pcc_nal_unit_type_plus1减1指定变量PccNalUnitType的值,该值指定PCC NAL单元中包含的RBSP数据结构类型,如表4所示。变量NalUnitType定义如下:

[0113] PccNalUnitType=pcc_nal_unit_type_plus1-1 (7-1)

[0114] nal_unit_type在UNSPEC25..UNSPEC30(包括端值)范围内的PCC NAL单元(语义未说明)不应影响此处指定的解码过程。需要说明的是,在UNSPEC25..UNSPEC30范围内的PCC NAL单元类型的使用可以由应用确定。本发明中未指定PccNalUnitType的这些值的解码过程。由于不同的应用可能出于不同目的使用这些PCC NAL单元类型,因此在设计使用这些PccNalUnitType值生成PCC NAL单元的编码器和设计使用这些PccNalUnitType值解释PCC NAL单元内容的解码器时应特别注意。本发明未定义对这些值的任何管理。这些PccNalUnitType值可能仅适用于以下上下文中:使用冲突(例如,对于同一PccNalUnitType值,PCC NAL单元内容的含义具有不同定义)不重要、不可能、受管理,例如,在控制应用或输

送规范中定义或管理,或者通过控制码流分布的环境。

[0115] 为了除确定码流的PCC AU中的数据量之外的目的,解码器可以忽略(从码流中移除并丢弃)使用PccNalUnitType保留值的所有PCC NAL单元的内容。该要求可以将来定义本发明的兼容扩展。

[0116] 表4PCC NAL单元类型代码

PccNalUnitType	PccNalUnitType 的名称	PCC NAL 单元和/或 RBSP 语法结构的内容
0	GOF_HEADER	帧组头 group_of_frames_header_rbsp()
1	AUX_INFO	辅助信息帧 auxiliary_info_frame_rbsp()
2	OCP_MAP	占用图帧 occupancy_map_frame_rbsp()
3	GMT_NALU	该 PCC NAL 单元的有效载荷包含每个识别的视频编解码器的几何形状属性的 NAL 单元。
4	TEXTURE_NALU	该 PCC NAL 单元的有效载荷包含根据识别的视频编解码器的纹理属性的 NAL 单元。
5	REFLECT_NALU	
6	TRANSP_NALU	
7	NORMAL_NALU	
8..24	RSV_8..RSV_24	保留
25..30	UNSPEC25.. UNSPEC30	未指定

[0117] [0118] 识别的视频编解码器(例如,HEVC或AVC)在帧组头NAL单元中指示,所述帧组头NAL单元存在于每个云点流(cloud point stream,CPS)的第一PCC AU中。pcc_stream_id表示PCC NAL单元的PCC流标识符(identifier, ID)。当PccNalUnitType等于GOF_HEADER、AUX_INFO或OCP_MAP时,pcc_stream_id的值设置为0。在定义一个或多个PCC档次和级别的集合时,pcc_stream_id的值可以限制为小于4。

[0119] PCC NAL单元的顺序及PCC NAL单元与PCC AU的关联描述如下。PCC AU包括零个或一个帧组头NAL单元、一个辅助信息帧NAL单元、一个占用图帧NAL单元和携带PCC属性(例如几何形状、纹理、反射率、透明度或法向属性)的数据单元的一个或多个视频AU。假设video_au(i, j)表示一个视频AU,其中,对于PCC属性ID等于attribute_type[i]的PCC属性,pcc_stream_id等于j。PCC AU中存在的视频AU可以按如下顺序排列。如果attributes_first_ordering_flag等于1,则对于PCC AU中存在的任何两个视频AU:video_au(i1, j1)和video_au(i2, j2),如下内容适用。如果i1小于i2,则无论j1和j2的值如何,video_au(i1, j1)都应该在video_au(i2, j2)之前。否则,如果i1等于i2,且j1大于j2,则video_au(i1, j1)应在video_au(i2, j2)之后。

[0120] 否则(例如,attributes_first_ordering_flag等于0),对于PCC AU中存在的任何两个视频AU:video_au(i1, j1)和video_au(i2, j2),以下内容适用。如果j1小于j2,则无论i1和i2的值如何,video_au(i1, j1)都应该在video_au(i2, j2)之前。否则,如果j1等于j2,

且 i_1 大于 i_2 ,则 $\text{video_au}(i_1, j_1)$ 应在 $\text{video_au}(i_2, j_2)$ 之后。视频AU的上述顺序导致以下结果。如果 $\text{attributes_first_ordering_flag}$ 等于1,则视频AU(当存在时)在PCC AU中的顺序如下(按所列顺序),其中,在PCCAU中,每个特定PCC属性的所有PCC NAL单元(当存在时)在解码顺序上是连续的,而不与其它PCC属性的PCC NAL单元交织:

[0121] $\text{video_au}(0, 0), \text{video_au}(0, 1), \dots, \text{video_au}(0, \text{num_streams_for_attribute}[0]),$

[0122] $\text{video_au}(1, 0), \text{video_au}(1, 1), \dots, \text{video_au}(1, \text{num_streams_for_attribute}[1]),$

[0123]

[0124] $\text{video_au}(\text{num_attributes}-1, 0), \text{video_au}(\text{num_attributes}-1, 1), \dots, \text{video_au}(\text{num_attributes}-1, \text{num_streams_for_attribute}[1]).$

[0125] 否则($\text{attributes_first_ordering_flag}$ 等于0),视频AU(当存在时)在PCC AU中的顺序如下(按所列顺序),其中,在PCC AU中,每个特定 pcc_stream_id 值的所有PCC NAL单元(当存在时)在解码顺序上是连续的,而不与其它 pcc_stream_id 值的PCC NAL单元交织:

[0126] $\text{video_au}(0, 0), \text{video_au}(1, 0), \dots, \text{video_au}(\text{num_attributes}-1, 0),$

[0127] $\text{video_au}(0, 1), \text{video_au}(1, 1), \dots, \text{video_au}(\text{num_attributes}-1, 1),$

[0128]

[0129] $\text{video_au}(0, \text{num_streams_for_attribute}[1]), \text{video_au}(1, \text{num_streams_for_attribute}[1]), \dots, \text{video_au}(\text{num_attributes}-1, \text{num_streams_for_attribute}[1]).$

[0130] NAL单元与视频AU的关联以及NAL单元在视频AU内的顺序在识别的视频编解码器的规范(例如HEVC或AVC)中指定。识别的视频编解码器在帧头NAL单元中指示,所述帧头NAL单元存在于每个CPS的第一个PCCAU中。

[0131] 每个CPS的第一个PCC AU以帧组头NAL单元开始,每个帧组头NAL单元表示新的PCCAU的开始。

[0132] 其它PCC AU以辅助信息帧NAL单元开始。换句话说,辅助信息帧NAL单元在前面没有帧组头NAL单元时开始新的PCC AU。

[0133] 帧组头RBSP语义如下: num_attributes 表示CPS中可以携带的PCC属性(例如几何形状、纹理等)的最大数量。需要说明的是,在定义一个或多个PCC档次和级别的集合时, num_attributes 的值可以限制为等于或小于5。 $\text{attributes_first_ordering_flag}$ 设置为0时表示:在PCC AU中,每个特定PCC属性的所有PCC NAL单元(当存在时)在解码顺序上是连续的,而不与其它PCC属性的PCC NAL单元交织。 $\text{attributes_first_ordering_flag}$ 设置为1时表示:在PCC AU中,每个特定 pcc_stream_id 值的所有PCC NAL单元(当存在时)在解码顺序上是连续的,而不与其它 pcc_stream_id 值的PCC NAL单元交织。 $\text{attribute_type}[i]$ 表示第*i*个PCC属性的PCC属性类型。下表5说明了不同PCC属性类型的解释。在定义一个或多个PCC档次和级别的集合时, $\text{attribute_type}[0]$ 和 $\text{attribute_type}[1]$ 的值可以分别限制为等于0和等于1。

[0134] 表5 $\text{attribute_type}[i]$ 的说明

值	PCC 属性名称	识别的 PCC 属性
0	ATTR_GEOMETRY	几何形状
1	ATTR_TEXTURE	纹理
2	ATTR_REFLECT	反射率
3	ATTR_TRANSP	透明度
4	ATTR_NORMAL	法向
5..31	CODEC_RSV_5.. CODEC_RSV_31	保留

[0136] identified_codec_for_attribute[i]表示用于译码第i个PCC属性的识别的视频编解码,如下表6所示。

[0137] 表6identified_codec_for_attribute[i]的说明

值	identified_codec 的名称	识别的视频编解码器
0	CODEC_HEVC	ISO/IEC IS 23008-2 (HEVC)
1	CODEC_AVC	ISO/IEC IS 14496-10 (AVC)
2..255	CODEC_RSV_2.. CODEC_RSV_255	保留

[0139] num_streams_for_attribute[i]表示第i个PCC属性的PCC流的最大数量。需要说明的是,在定义一个或多个PCC档次和级别的集合时,num_streams_for_attribute[i]的值可以限制为小于或等于4。num_layers_for_attribute[i]表示第i个PCC属性的属性层的数量。需要说明的是,在定义一个或多个PCC档次和级别的集合时,num_layer_for_attribute[i]的值可以限制为小于或等于4。max_attribute_layer_idx[i][j]表示PCC流的属性层索引的最大值,其中,对于第i个PCC属性,pcc_stream_id等于j。max_attribute_layer_idx[i][j]的值应小于num_layer_for_attribute[i]的值。attribute_layers_combination_mode[i][j]表示在PCC流中携带的属性层的属性层组合模式,其中,对于第i个PCC属性,pcc_stream_id等于j。attribute_layers_combination_mode[i][j]的不同值的解释在下表7中说明。

[0140] 表7attribute_layers_combination_mode[i][j]的说明

值	组合模式的名称	识别的组合模式
0	COMBIN_MODE_TEMP_INTLV	时间交织模式, 其中, 每个 PCC AU 包含 PCC 流的一个属性层, 其中, 对于第 i 个 PCC 属性, pcc_stream_id 等于 j, 并且属性层在不同的 PCC AU 中进行时间交织
1..15	COMBIN_MODE_RSV_1.. COMBIN_MODE_RSV_255	保留

[0143] 当attribute_layers_combination_mode[i][j]存在且等于0时,变量

attrLayerIdx[i][j]推导如下,其中,变量attrLayerIdx[i][j]表示PCC流的属性层的属性层索引(对于第i个PCC属性,pcc_stream_id等于j),属性层的PCC NAL单元携带在视频AU中,其中,图序计数值等于PicOrderCntVal,如所识别的视频编解码器的规范中所指定:

```

tmpVal = PicOrderCntVal % num_streams_for_attribute[i]
if(j == 0)
[0144]      attrLayerId[i][j] = tmpVal
else
            attrLayerId[i][j][k] = max_attribute_layer_id[i][j - 1] + 1 + tmpVal

```

[0145] regular) points) flag[i][j]等于1时表示对于第i个PCC属性,层索引等于j的属性层携带点云信号的规则点。regular_points_flag[i][j]在设置为0时,表示对于第i个PCC属性,层索引等于j的属性层携带点云信号的不规则点。需要说明的是,在定义一个或多个PCC档次和级别的集合时,regular_points_flag[i][j]的值可以限制为0。frame_width表示几何形状视频和纹理视频以像素计算的帧宽度。帧宽度应为occupancyResolution的倍数。frame_height表示几何形状视频和纹理视频以像素计算的帧高度。帧高度应为occupancyResolution的倍数。occupancy_resolution表示几何形状视频和纹理视频中打包片以像素计算的水平和垂直分辨率。occupancy_solution应为occupancyPrecision的偶数倍数值。radius_to_smoothing表示用于检测邻居以进行平滑的半径。radius_to_smoothing的取值范围应为0到255(包括端值)。

[0146] neighbor_count_smoothing表示用于平滑的邻居的最大数量。neighbor_count_smoothing的取值范围应为0到255(包括端值)。radius2_boundary_detection表示边界点检测半径。radius2_boundary_detection的取值范围应为0到255(包括端值)。threshold_smoothing表示平滑阈值。threshold_smoothing的取值范围应为0到255(包括端值)。lossless_geometry表示无损几何形状译码。lossless_geometry的值在等于1时,表示点云几何形状信息以无损方式译码。lossless_geometry的值在等于0时,表示点云几何形状信息以有损方式译码。lossless_texture表示无损纹理编码。lossless_texture的值在等于1时,表示点云纹理信息以无损方式译码。lossless_texture的值在等于0时,表示点云纹理信息以有损方式译码。lossless_geometry_444表示对几何形状帧使用4:2:0视频格式或4:4:4视频格式。lossless_geometry_444的值在等于1时,表示几何形状视频以4:4:4格式译码。lossless_geometry_444的值在等于0时,表示几何形状视频以4:2:0格式译码。

[0147] absolute_d1_coding表示除了最接近投影平面的层以外的几何层的译码方式。absolute_d1_coding在等于1时,表示对除了最接近投影平面的层以外的几何形状层的实际几何值进行译码。absolute_d1_coding在等于0时,表示除了最接近投影平面的层以外的几何形状层进行差分译码。bin_arithmetic_coding表示是否使用二进制算术译码。bin_arithmetic_coding的值在等于1时,表示对所有语法元素都使用二进制算术译码。bin_arithmetic_coding的值在等于0时,表示对一些语法元素使用非二进制算术译码。gof_header_extension_flag在等于0时,表示帧组头RBSP语法结构中不存在gof_header_extension_data_flag语法元素。gof_header_extension_flag在等于1时,表示帧组头RBSP语法结构中存在gof_header_extension_data_flag语法元素。解码器可以忽略帧组头NAL单元中gof_header_extension_flag的值1之后的所有数据。gof_header_extension_data_

flag可以具有任何值，并且该标志的存在和值不影响解码器一致性。解码器可以忽略所有gof_header_extension_data_flag语法元素。

[0148] PCC档次和级别语义如下：pcc_profile_idc表示CPS符合的档次。pcc_pl_reserved_zero_19bits在符合本发明版本的码流中等于0。保留pcc_pl_reserved_zero_19bits的其它值以供ISO/IEC使用。解码器可以忽略pcc_pl_reserved_zero_19bits的值。pcc_level_idc表示CPS符合的级别。当符合HEVC的解码器对等于attribute_type[i]（通过子码流提取过程指定提取）的PCC属性类型的HEVC码流进行解码时，在激活SPS中，hevc_ptl_12bytes_attribute[i]可以等于介于general_profile_idc与general_level_idc（包括端值）之间的12个字节的值。当符合AVC的解码器对等于attribute_type[i]（通过子码流提取过程指定提取）的PCC属性类型的AVC码流进行解码时，在激活SPS中，avc_pl_3bytes_attribute[i]可以等于介于profile_idc与level_idc（包括端值）之间的3个字节的值。

[0149] 子码流提取过程如下：该过程的输入为PCC码流inBitstream、目标PCC属性类型targetAttrType和目标PCC流ID值targetStreamId。该过程的输出是一个子码流。对于输入码流，码流一致性要求：根据属性类型targetAttrType的所识别的视频编解码规范，任何输出子码流应该是一致的视频码流，所述输出子码流是该节中规定的过程的输出，且具有一致的PCC码流inBitstream、表示存在于inBitstream中的PCC属性的任何类型的targetAttrType，以及对于属性类型targetAttrType，小于或等于存在于inBitstream中的PCC流的最大PCC流ID值的targetStreamId的。

[0150] 输出子码流通过以下顺序步骤推导。根据targetAttrType的值，以下内容适用。如果targetAttrType等于ATTR_GEOMETRY，则移除PccNalUnitType不等于GMTRY_NALU或pcc_stream_id不等于targetStreamId的所有PCC NAL单元。否则，如果targetAttrType等于ATTR_TEXTURE，则移除PccNalUnitType不等于TEXTURE_NALU或pcc_stream_id不等于targetStreamId的所有PCC NAL单元。否则，如果targetAttrType等于ATTR_REFLECT，则移除PccNalUnitType不等于REFLECT_NALU或pcc_stream_id不等于targetStreamId的所有PCC NAL单元。否则，如果targetAttrType等于ATTR_TRANSPI，则移除PccNalUnitType不等于TRANSPI_NALU或pcc_stream_id不等于targetStreamId的所有PCC NAL单元。否则，如果targetAttrType等于ATTR_NORMAL，则移除PccNalUnitType不等于NORMAL_NALU或pcc_stream_id不等于targetStreamId的所有PCC NAL单元。对于每个PCC NAL单元，也可以移除第一个字节。

[0151] 在如上所述的第一组方法的替代实施例中，PCC NAL单元头设计成对pcc_stream_id使用更多位，并允许每个属性有多于四个流。在这种情况下，在PCC NAL单元头中再添加一种类型。

[0152] 图8是使用多个编解码器843和844对PCC属性841和842进行编码的示例性机制800的示意图。例如，机制800可以用于对PCC视频流700的属性进行编码和/或解码。因此，机制800可以用于根据点云媒体500对点云媒体帧600进行编码和/或解码。因此，编码器300可以使用机制800从PCC序列创建码流，并且解码器400可以使用机制800从码流重建PCC序列。因此，机制800可以由编解码系统200使用，并且可以进一步用于支持方法100。

[0153] 机制800可以适用于多个PCC属性841和842。例如，PCC属性841和842可以是从几何形状属性、纹理属性、反射率属性、透明度属性和法向属性组成的组中选取的任意两个属

性。如图8所示,机制800描绘了从左到右的编码过程和从右到左的解码过程。编解码器843和844可以是任意两个编解码器,例如HEVC、AVC、VVC等,或其任何版本。当编码某些PCC属性841和842时,某些编解码器843和844或其版本可能比其它编解码器更高效。在该示例中,分别地,编解码器843用于编码属性841,而编解码器844用于编码属性842。组合这些编码的结果以创建包含PCC属性841和842的PCC视频流845。在解码器侧,分别地,编解码器843用于解码属性841,而编解码器844用于解码属性842。然后,可以重新组合解码属性841和842以生成经解码的PCC视频流845。

[0154] 采用机制800的益处是,可以为对应的属性841和842选择最高效的编解码器843和844。机制800不限于两个属性841和842,两个编解码器843和844。例如,每个属性(几何形状、纹理、反射率、透明度和法向)可以由单独的编解码器进行编码。为了确保可以选择正确的编解码器843和844来解码对应的属性841和842,编码器可以分别指示编解码器843和844及其与属性841和842的对应关系。例如,编码器可以在GOF头中添加一个或多个语法元素以表示编解码器与属性的对应关系。然后,解码器可以读取相关语法,为属性841和842选择正确的编解码器843和844,并解码PCC视频流845。在一个具体示例中,可以使用identified_codec_for_attribute语法元素来分别表示用于属性841和842的编解码器843和844。

[0155] 图9是属性层931、932、933和934的示例的示意图900。例如,可以使用属性层931、932、933和934来携带PCC视频流700的属性。因此,当根据点云媒体500对点云媒体帧600进行编码和/或解码时,可以使用层931、932、933和934。因此,编码器300可以使用层931、932、933和934从PCC序列创建码流,并且解码器400可以使用层931、932、933和934从码流重建PCC序列。因此,层931、932、933和934可以由编解码系统200使用,还可以用于支持方法100。此外,属性层931、932、933、934可用于携带属性841、842中的一个或多个。

[0156] 属性层931、932、933和934是与一种属性相关的数据分组,所述数据分组可以独立于与同一属性相关的其它数据分组来存储和/或修改。因此,可以在不影响其余属性层931、932、933和/或934的情况下改变和或表示每个属性层931、932、933和/或934。在一些示例中,属性层931、932、933和/或934可以在视觉上表示为在彼此顶部上,如图9所示。例如,覆盖整个对象的纹理(例如,总体)可以存储在属性层931中,而更详细的纹理(例如,具体)包括在属性层932、933和/或934中。在另一示例中,属性层931和/或932可应用于奇数编号的帧,属性层933和/或934可应用于偶数编号的帧。这可以响应于帧率的变化而省略一些层。每个属性可以具有0至4个属性层931、932、933和/或934。为了指示所使用的配置,编码器可以在序列级数据(例如GOF头)中使用语法元素,例如num_layers_for_attribute[i]。解码器可以读取语法元素,并确定用于每个属性的属性层931、932、933和/或934的数量。附加语法元素,例如attribute_layers_combination_mode[i][j]和attrLayerIdx[i][i],也可以用于分别表示PCC视频流中使用的属性层的组合以及对应属性所使用的每个层的索引。

[0157] 在又一示例中,一些属性层(例如,属性层931、932和933)可以携带与规则片相关的数据,而其它属性层(例如,属性层934)携带与不规则点云片相关的数据。这可能是有用的,因为不规则点云可以使用不同于规则点云片的数据来描述。为了指示特定层携带与不规则点云相关联的数据,编码器可以对序列级数据中的另一语法元素进行编码。在一个具体示例中,GOF头中的regular_points_flag可以用于表示属性层携带至少一个不规则点云点。然后,解码器可以读取语法元素并对应地解码对应的属性层。

[0158] 图10是属性流1031、1032、1033和1034的示例的示意图1000。例如,可以使用属性流1031、1032、1033和1034来携带PCC视频流700的属性。因此,当根据点云媒体500对点云媒体帧600进行编码和/或解码时,可以使用属性流1031、1032、1033和1034。因此,编码器300可以使用属性流1031、1032、1033和1034从PCC序列创建码流,并且解码器400可以使用属性流931、932、933和934从码流重建PCC序列。因此,属性流1031、1032、1033和1034可以由编解码系统200使用,还可以用于支持方法100。此外,属性流1031、1032、1033、1034可用于携带属性841、842中的一个或多个。此外,可以使用属性流1031、1032、1033和1034来携带属性层931、932、933和934。

[0159] 属性流1031、1032、1033和1034是随时间变化的属性数据序列。具体地,属性流1031、1032、1033和1034为PCC视频流的子流。每个属性流1031、1032、1033和1034携带属性特定NAL单元的序列,因此作为存储和/或传输数据结构。每个属性流1031、1032、1033和1034可以携带数据的一个或多个属性层931、932、933和934。例如,属性流1031可以携带属性层931和932,而属性流1032携带属性层931和932(省略属性流1033和1034)。在另一个示例中,每个属性流1031、1032、1033和1034携带单个对应的属性层931、932、933和934。在其它示例中,一些属性流1031、1032、1033和1034携带多个属性层931、932、933和934,而其它属性流1031、1032、1033和1034携带单个属性层931、932、933和934或被省略。可以看到,属性流1031、1032、1033和1034以及属性层931、932、933和934可以进行许多种组合和排列。因此,编码器可以在序列级数据(例如GOF头中)中使用语法元素(例如num_streams_for_attribute)来表示用于编码每个属性的属性流1031、1032、1033和1034的数量。然后,解码器可以使用此类信息(例如与属性层信息结合)来解码属性流1031、1032、1033和1034,以便重建PCC序列。

[0160] 图11是使用多个编解码器对PCC视频序列进行编码的示例性方法1100的流程图。例如,方法1100可以根据机制800使用属性层931、932、933和934和/或流1031、1032、1033和1034将数据组织到码流中。此外,方法1100可以指定用于编码GOF头中的属性的机制。此外,方法1100可以通过根据点云媒体500对点云媒体帧600进行编码来生成PCC视频流700。此外,编解码系统200和/或编码器300在执行方法100的编码步骤时可以使用方法1100。

[0161] 方法1100可以在编码器接收包含点云媒体的PCC帧序列时开始。例如,编码器可以响应于接收用户命令而确定编码此类帧。在方法1100中,编码器可以确定第一属性应该由第一编解码器编码,而第二属性应该由第二编解码器编码。例如,当第一编解码器对于第一属性更高效且第二编解码器对于第二属性更高效时,可以根据预定条件和/或根据用户输入来进行该决策。因此,在步骤1101中,编码器使用第一编解码器将PCC帧序列的第一属性编码到码流中。此外,在步骤1103中,编码器使用不同于第一编解码器的第二编解码器将PCC帧序列的第二属性编码到码流中。

[0162] 在步骤1105中,编码器将各种语法元素与经编码的视频数据一起编码到码流中。例如,可以将语法元素译码到包含序列级参数的序列级数据单元中,以便向解码器指示在编码期间作出的决策,从而可以正确地重建PCC帧。具体地,编码器可以对序列级数据单元进行编码以包括表示第一属性由第一编解码器译码且表示第二属性由第二编解码器译码的第一语法元素。在一个具体示例中,PCC帧可以包括多个属性,所述多个属性包括第一属性和第二属性。此外,PCC帧的多个属性可以包括几何形状、纹理、反射率,以及透明度和法

向中的一个或多个。此外，第一语法元素可以为码流中的GOF头中包含的identified_codec_for_attribute元素。

[0163] 在一些示例中，第一属性可以组织到多个流中。在这种情况下，可以使用第二语法元素来表示与第一属性相关联的码流的数据单元的流成员资格。在一些示例中，第一属性也可以组织到多个层中。在这种情况下，第三语法元素可以表示与第一属性相关联的码流的数据单元的层成员资格。在一个具体示例中，第二语法元素可以为num_streams_for_attribute元素，第三语法元素可以为num_layers_for_attribute元素，这些元素的每一个可以包含在码流中的帧组头中。在另一示例中，第四语法元素可以用于表示所述多个层中的第一层包含与不规则点云相关联的数据。在一个具体示例中，第四语法元素可以是码流中的帧组头中包含的regular_points_flag元素。

[0164] 通过在序列级数据中包括此类信息，解码器可以具有足以解码PCC视频序列的信息。因此，在步骤1107中，编码器可以发送码流，以便根据由第一编解码器译码的第一属性和第二编解码器译码的第二属性以及本文所述的其它属性和/或语法元素生成经解码的PCC帧序列。

[0165] 图12是使用多个编解码器对PCC视频序列进行解码的示例性方法1200的流程图。例如，方法1200可以根据机制800使用属性层931、932、933和934和/或流1031、1032、1033和1034从码流读取数据。此外，方法1200可以通过读取GOF头确定用于译码属性的机制。此外，方法1200可读取PCC视频流700，以便重建点云媒体帧600和点云媒体500。此外，编解码系统200和/或解码器400可以在执行方法100的解码步骤时使用方法1200。

[0166] 方法1200可以在步骤1201中解码器接收包括PCC帧序列的码流时开始。然后，在步骤1205中，解码器可以对码流或其部分进行解析。例如，解码器可以对码流进行解析，得到包含序列级参数的序列级数据单元。序列级数据单元可以包括描述编码过程的各种语法元素。因此，解码器可以从码流中解析视频数据，并使用语法元素来确定解码视频数据的正确过程。

[0167] 例如，序列级数据单元可以包括表示第一属性由第一编解码器译码并且表示第二属性由第二编解码器译码的第一语法元素。在一个具体示例中，PCC帧可以包括多个属性，所述多个属性包括第一属性和第二属性。此外，PCC帧的多个属性可以包括几何形状、纹理、反射率、透明度和法向中的一个或多个。此外，第一语法元素可以为码流中的GOF头中包含的identified_codec_for_attribute元素。

[0168] 在一些示例中，第一属性可以组织成多个流。在这种情况下，可以使用第二语法元素来表示与第一属性相关联的码流的数据单元的流成员资格。在一些示例中，第一属性也可以组织成多个层。在这种情况下，第三语法元素可以表示与第一属性相关联的码流的数据单元的层成员资格。在一个具体示例中，第二语法元素可以为num_streams_for_attribute元素，第三语法元素可以为num_layers_for_attribute元素，这些元素的每一个可以包含在码流中的帧组头中。在另一示例中，第四语法元素可以用于表示所述多个层中的第一层包含与不规则点云相关联的数据。在一个具体示例中，第四语法元素可以是码流中的帧组头中包含的regular_points_flag元素。

[0169] 因此，解码器可以在步骤1207处通过第一编解码器解码第一属性，并且通过第二编解码器解码第二属性，以便生成经解码的PCC帧序列。当确定基于编解码器对PCC视频序

列的各种属性进行解码时要采用的适当机制时,解码器还可以利用本文所述的其它属性和/或语法元素。

[0170] 图13为示例性视频译码设备1300的示意图。视频译码设备1300适用于实现本文描述的所公开示例/实施例。视频译码设备1300包括下行端口1320、上行端口1350,和/或收发单元(Tx/Rx)1310,其中,所述收发单元包括用于通过网络在上行和/或下行发送数据的发送器和/或接收器。视频译码设备1300还包括:处理器1330,包括用于处理数据的逻辑单元和/或中央处理单元(central processing unit,CPU);和用于存储数据的存储器1332。视频译码设备1300还可以包括与上行端口1350和/或下行端口1320耦合的电子组件、光电(optical-to-electrical,OE)组件、电光(electrical-to-optical,E0)组件,和/或无线通信组件,用于通过电、光或无线通信网络进行数据通信。视频译码设备1300还可以包括输入和/或输出(input/output,I/O)设备1360,用于向用户发送数据和从用户发送数据。I/O设备1360可以包括输出设备,例如用于显示视频数据的显示器、用于输出音频数据的扬声器等。I/O设备1360还可以包括输入设备,如键盘、鼠标、轨迹球等,和/或用于与此类输出设备交互的对应接口。

[0171] 处理器1330通过硬件和软件实现。处理器1330可以实现为一个或多个CPU芯片、核(例如像多核处理器)、现场可编程门阵列(field-programmable gate array,FPGA)、专用集成电路(application specific integrated circuit,ASIC)和数字信号处理器(digital signal processor,DSP)。处理器1330与下行端口1320、Tx/Rx 1310、上行端口1350和存储器1332通信。处理器1330包括译码模块1314。译码模块1314实现上述公开的实施例,例如方法100、1100、1200、1500和1600以及机制800,其可以采用点云媒体500、点云媒体帧600和/或以层931-934译码的PCC视频流700和/或流1031-1034。译码模块1314还可以实现本文描述的任何其它方法/机制。此外,所述译码模块1314可以实现编解码系统200、编码器300和/或解码器400。例如,译码模块1314可以针对具有多个流和层的PCC使用扩展属性集,并且可以在序列级数据中指示这种属性集的使用以支持解码。因此,译码模块1314使视频译码设备1300在译码PCC视频数据时提供其它功能和/或灵活性。因此,译码模块1314改进了视频译码设备1300的功能,并解决了视频译码领域特有的问题。此外,译码模块1314可以将视频译码设备1300变换到不同的状态。或者,译码模块1314可以实现为存储在存储器1332中并由处理器1330执行的指令(例如,存储在非瞬时性介质上的计算机程序产品)。

[0172] 存储器1332包括一个或多个类型的存储器,如磁盘、磁带机、固态驱动器、只读存储器(read only memory,ROM)、随机存取存储器(random access memory,RAM)、闪存、三态内容寻址存储器(ternary content-addressable memory,TCAM)、静态随机存取存储器(static random-access memory,SRAM)等。存储器1332可用作溢出数据存储设备,以在选择执行此类程序时存储该程序,以及存储在程序执行期间读取的指令和数据。

[0173] 图14是用于使用多个编解码器对PCC视频序列进行译码的示例性系统1400的示意图。系统1400包括视频编码器1402,所述视频编码器1402包括第一属性编码模块1401,用于使用第一编解码器将PCC帧序列的第一属性编码到码流中。视频编码器1402还包括第二属性编码模块1403,用于使用不同于第一编解码器的第二编解码器将PCC帧序列的第二属性编码到码流中。视频编码器1402还包括语法编码模块1405,用于将包含序列级参数的序列级数据单元编码到码流中,其中,所述序列级数据单元包括第一语法元素,所述第一语法元

素表示第一属性由第一编解码器译码，并且表示第二属性由第二编解码器译码。视频编码器1402还包括发送模块1407，用于发送码流以支持根据由第一编解码器译码的第一属性和由第二编解码器译码的第二属性生成经解码的PCC帧序列。视频编码器1402的模块也可以用于执行上文关于方法1100和/或1500所述的任何步骤/项目。

[0174] 系统1400还包括视频解码器1410，所述视频解码器1410包括接收模块1411，用于接收包括PCC帧序列的码流。视频解码器1410还包括解析模块1413，用于对码流进行解析，以获取包含序列级参数的序列级数据单元，其中，所述序列级数据单元包括第一语法元素，所述第一语法元素表示PCC帧的第一属性由第一编解码器译码，并且表示PCC帧的第二属性由第二编解码器译码。视频解码器1410还包括解码模块1415，用于通过第一编解码器对第一属性进行解码，并通过第二编解码器对第二属性进行解码，以生成经解码的PCC帧序列。视频解码器1410的模块也可以用于执行上文关于方法1200和/或1600所述的任何步骤/项目。

[0175] 图15是使用多个编解码器对PCC视频序列进行编码的另一示例性方法1500的流程图。例如，方法1500可以根据机制800同时使用属性层931、932、933和934和/或流1031、1032、1033和1034将数据组织到码流中。此外，方法1500可以指定用于编码GOF头中的属性的机制。此外，方法1500可以通过根据点云媒体500对点云媒体帧600进行编码来生成PCC视频流700。此外，在执行方法100的编码步骤时，编解码系统200和/或编码器300可以使用方法1500。

[0176] 在步骤1501处，多个PCC属性作为PCC帧序列的一部分编码到码流中。PCC属性使用多个编解码器进行编码。PCC属性包括几何形状和纹理。PCC属性还包括反射率、透明度和法向中的一个或多个。每个经译码的PCC帧由一个或多个PCC NAL单元表示。在步骤1503处，针对每个PCC属性编码一个指示。该指示表示用于译码对应PCC属性的视频编解码器。在步骤1505处，向解码器发送码流。

[0177] 图16是使用多个编解码器对PCC视频序列进行解码的另一示例性方法1600的流程图。例如，方法1600可以根据机制800同时使用属性层931、932、933和934和/或流1031、1032、1033和1034从码流读取数据。此外，方法1600可以通过读取GOF头确定用于译码属性的机制。此外，方法1600可读取PCC视频流700，以便重建点云媒体帧600和点云媒体500。此外，在执行方法100的解码步骤时，编解码系统200和/或解码器400可以使用方法1600。

[0178] 在步骤1601处，接收码流。码流包括多个经译码的PCC帧序列。经译码的PCC帧序列表示多个PCC属性。PCC属性包括几何形状和纹理。PCC属性还包括反射率、透明度和法向中的一个或多个。每个经译码的PCC帧由一个或多个PCC NAL单元表示。在步骤1603处，对码流进行解析，以针对每个PCC属性获取用于译码对应PCC属性的编解码器的指示。在步骤1605处，根据PCC属性的所指示的视频编解码器对码流进行解码。

[0179] 当第一组件与第二组件之间除了线、迹线或其它介质之外不存在中间组件时，第一组件与第二组件直接耦合。当第一组件与第二组件之间除了线、迹线或其它介质之外还存在中间组件时，第一组件与第二组件间接耦合。术语“耦合”及其变型包括直接耦合和间接耦合。除非另有说明，否则使用术语“约”意指包括后续数字±10%的范围。

[0180] 虽然本发明提供了若干个实施例，但应该理解，在不脱离本发明的精神或范围的情况下，所公开的系统和方法可以通过其它多种具体形式体现。本发明示例应被视为说明

性而非限制性的,且本发明并不限于本文所给出的细节。例如,可以在另一系统中结合或集成各种元件或组件,或者可以省略或不实现某些特征。

[0181] 此外,在不脱离本发明范围的情况下,各种实施例中描述和说明为离散或单独的技术、系统、子系统和方法可以与其它系统、组件、技术或方法进行组合或集成。变化、替换和变更的其它实例能够由本领域技术人员确定,并可在不脱离本文公开的精神和范围的情况下给出。

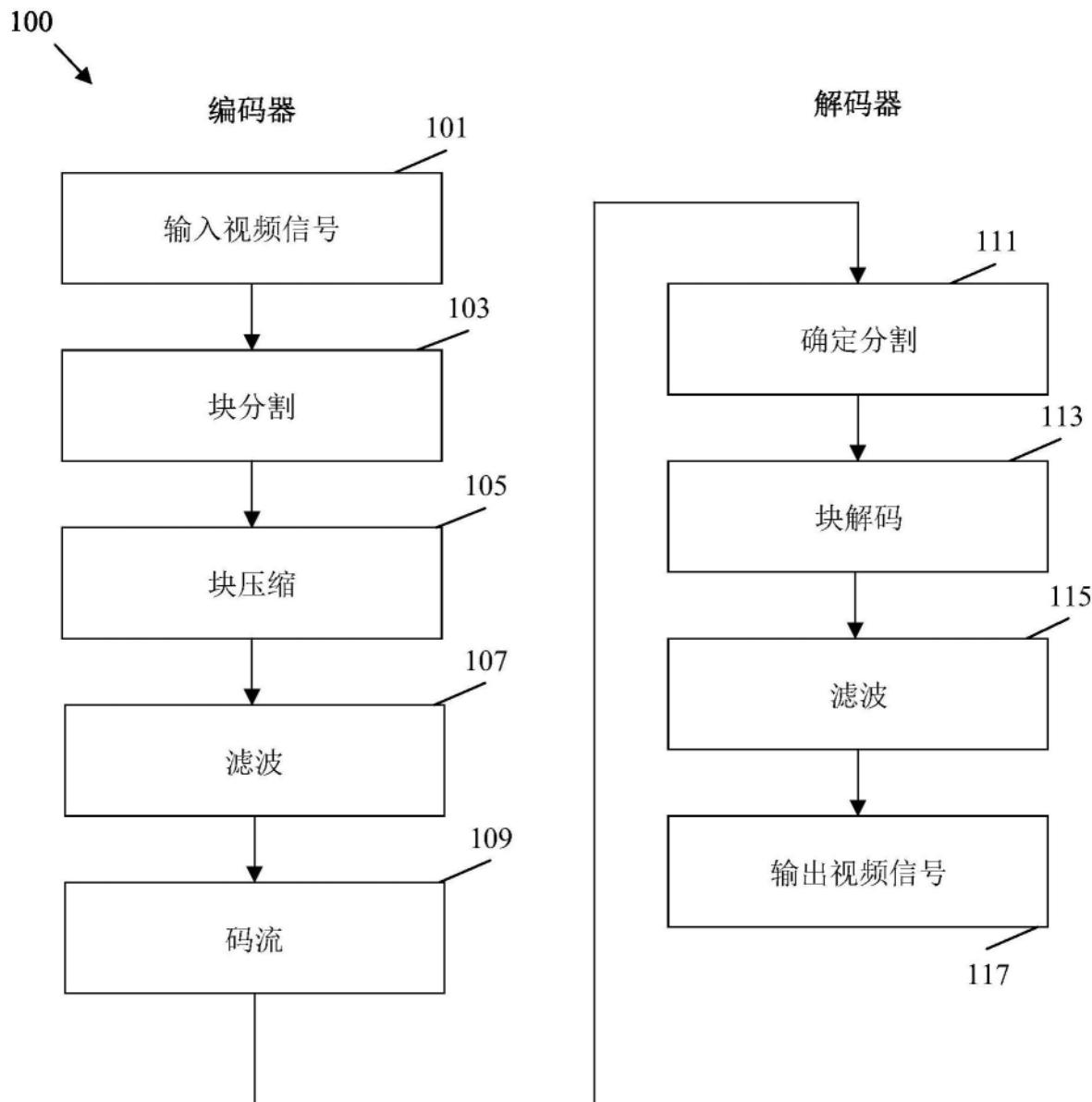


图1

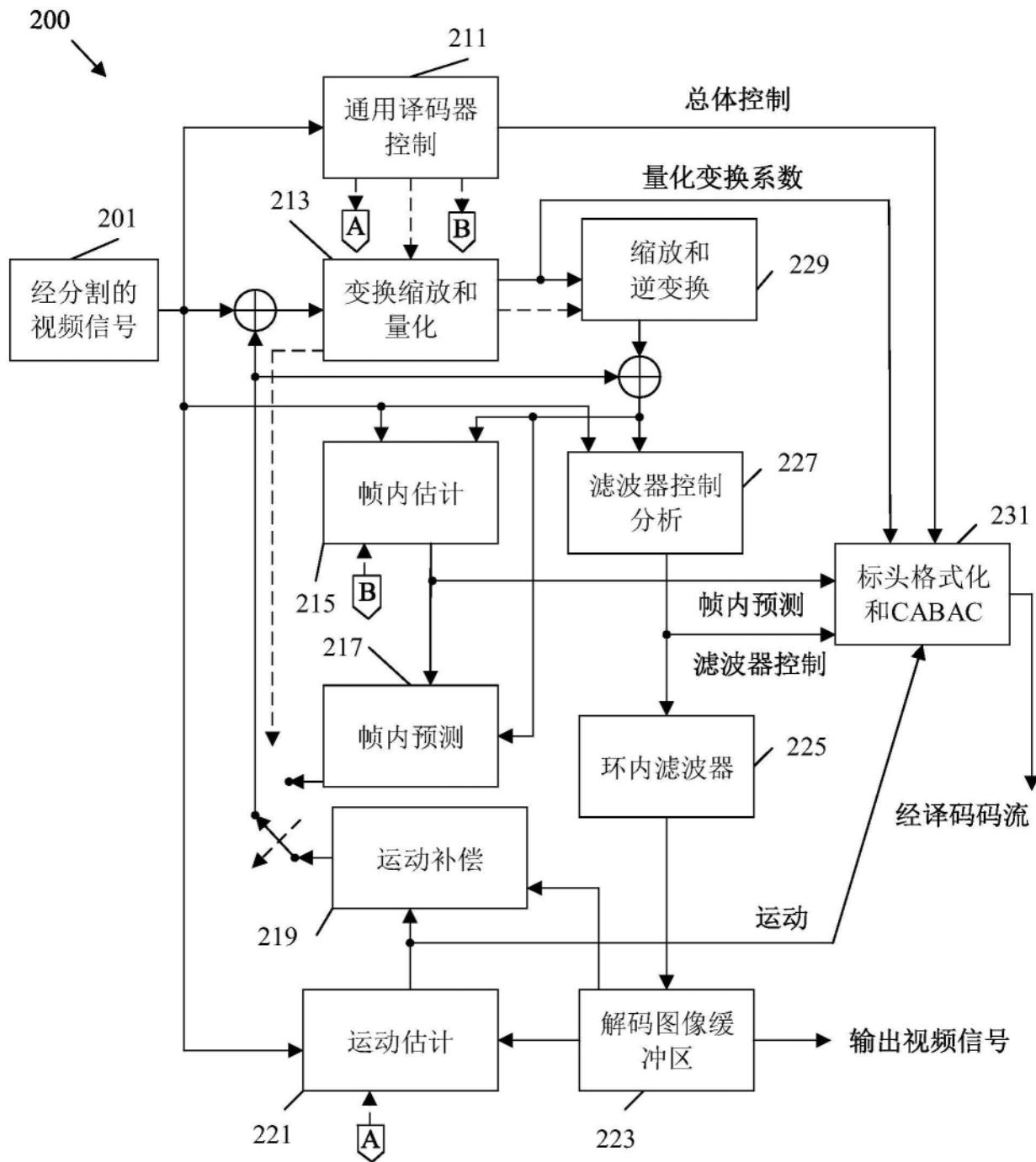


图2

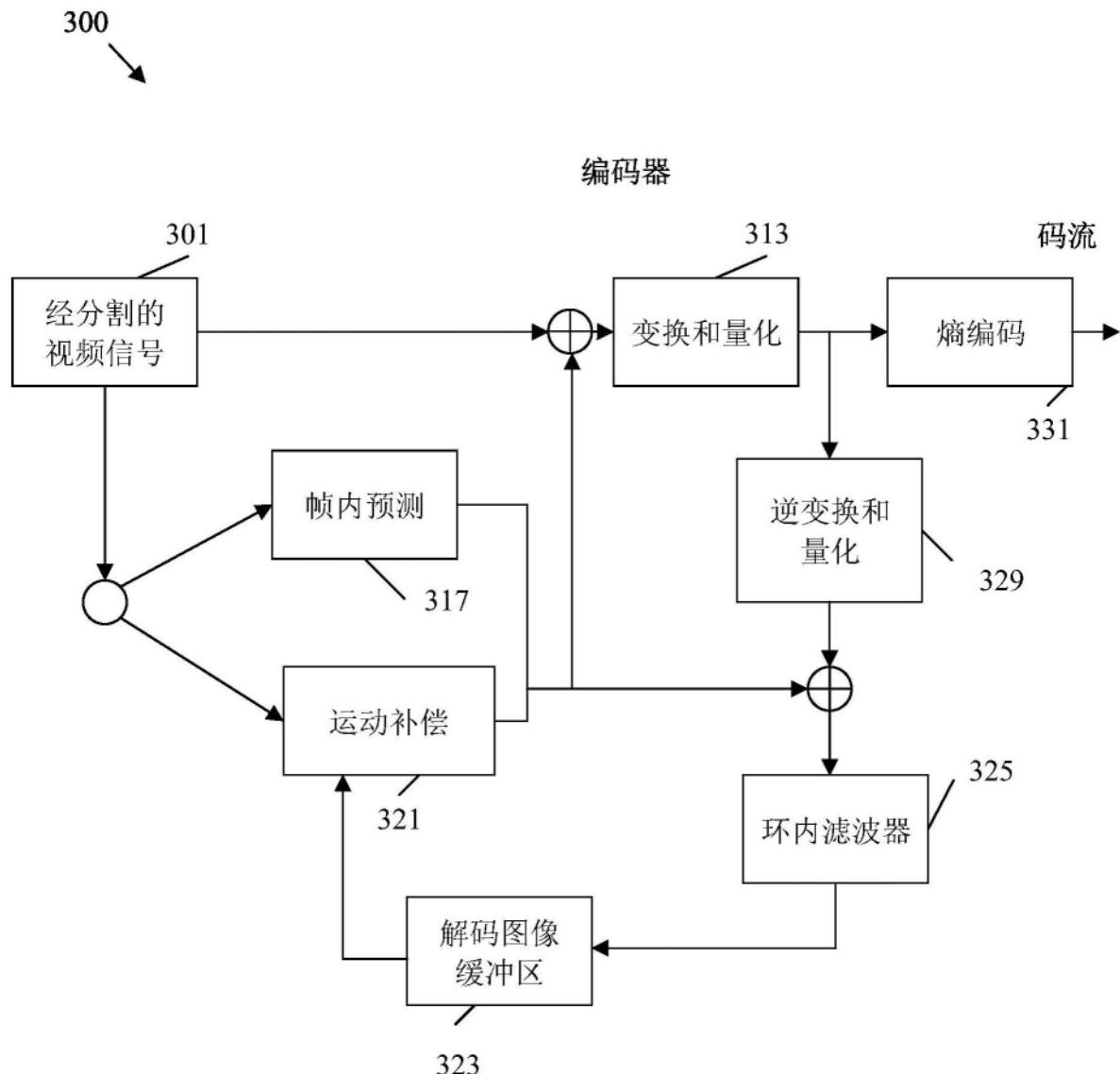


图3

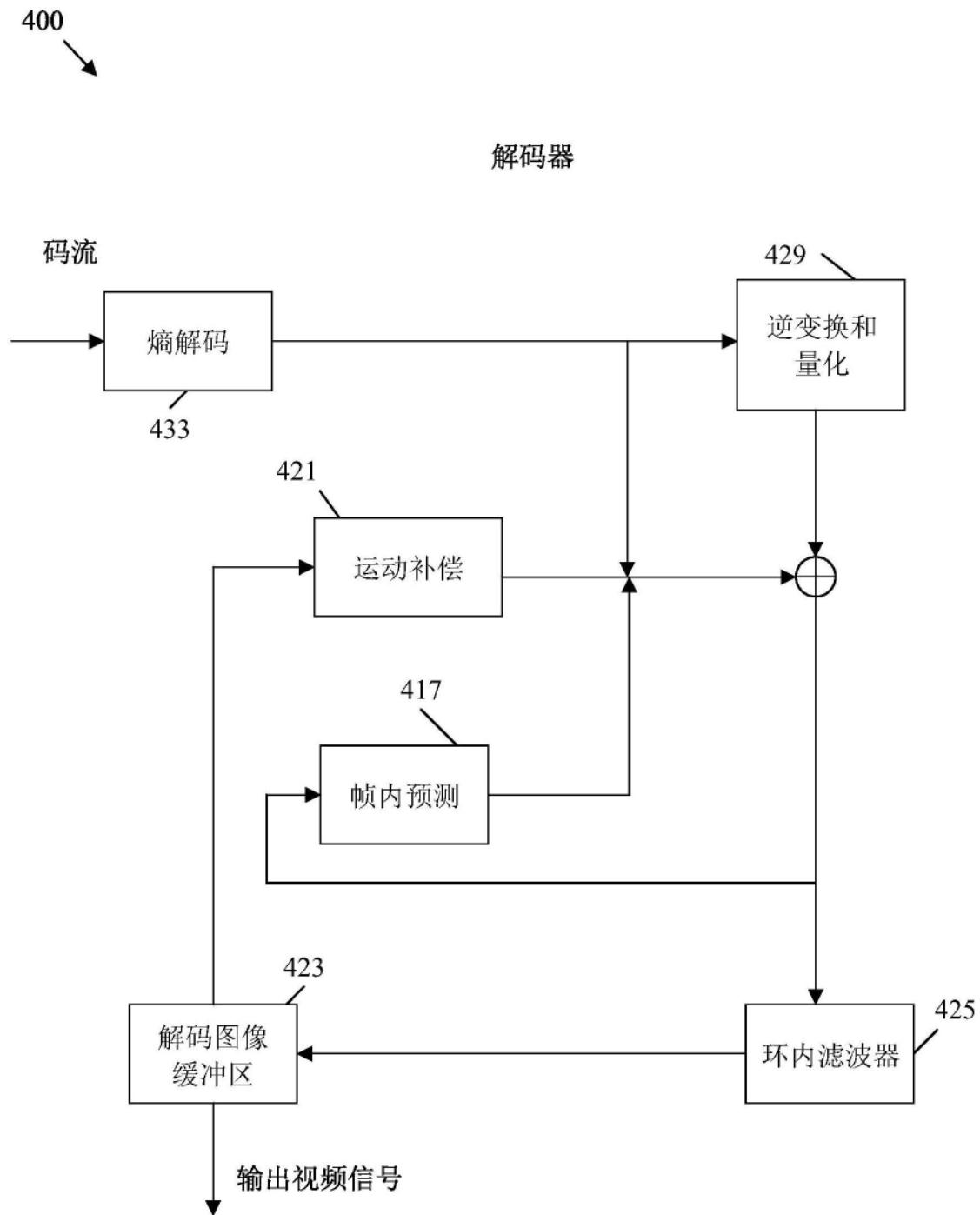


图4

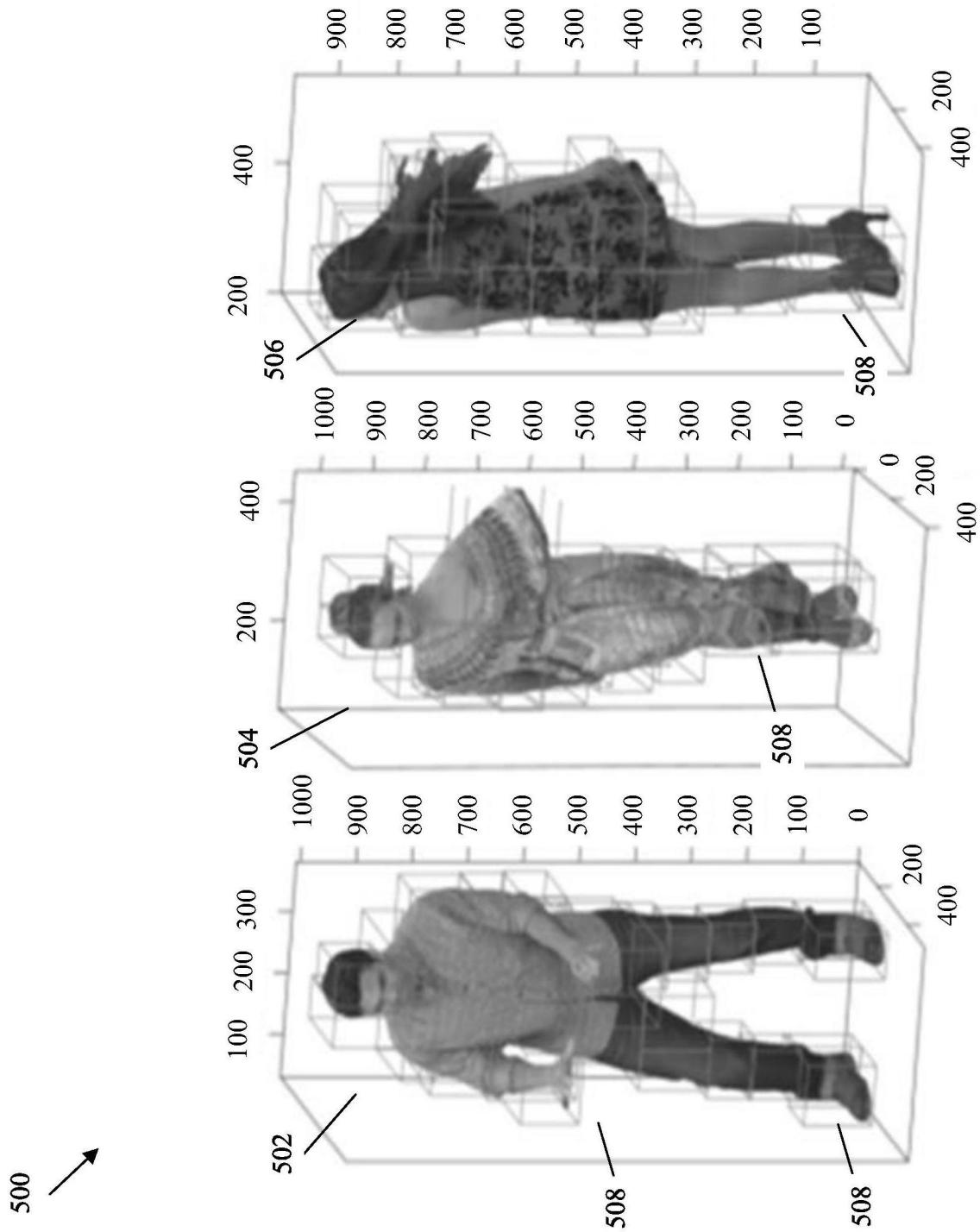


图5

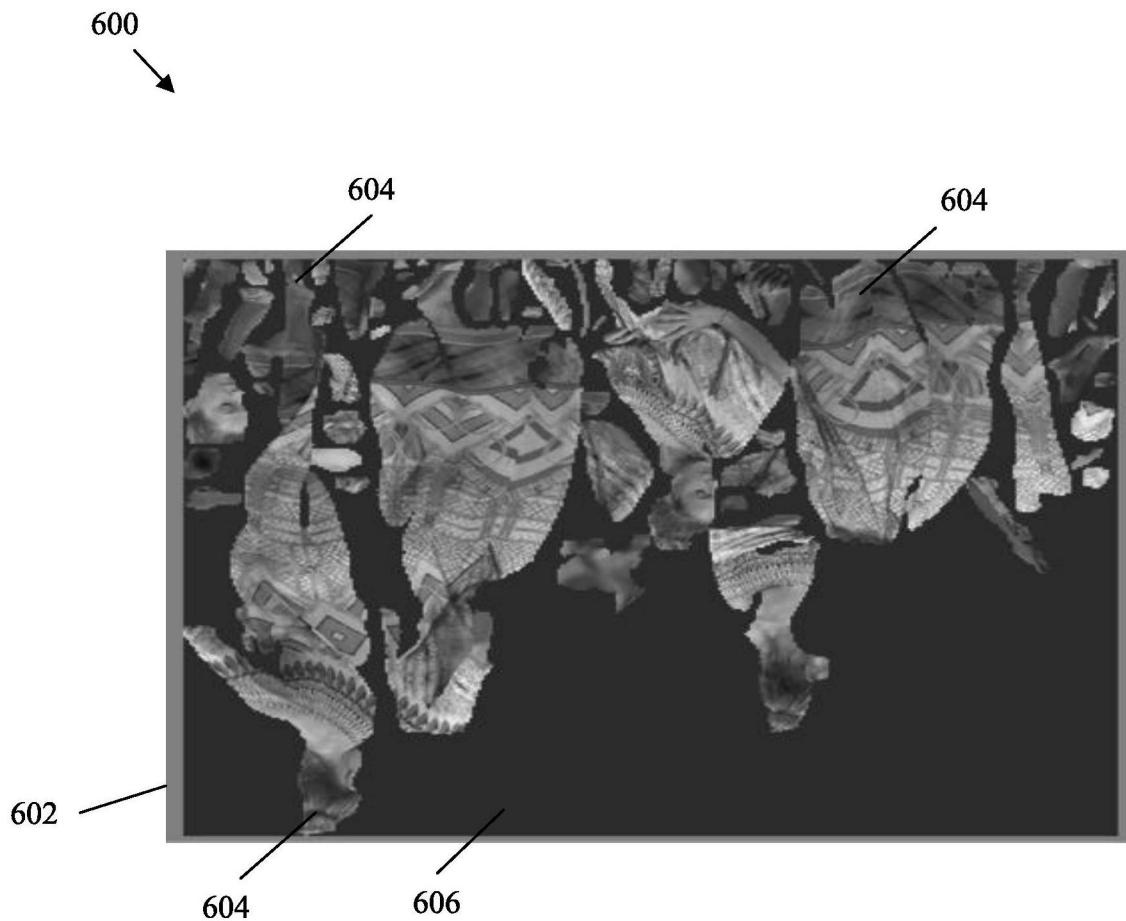


图6

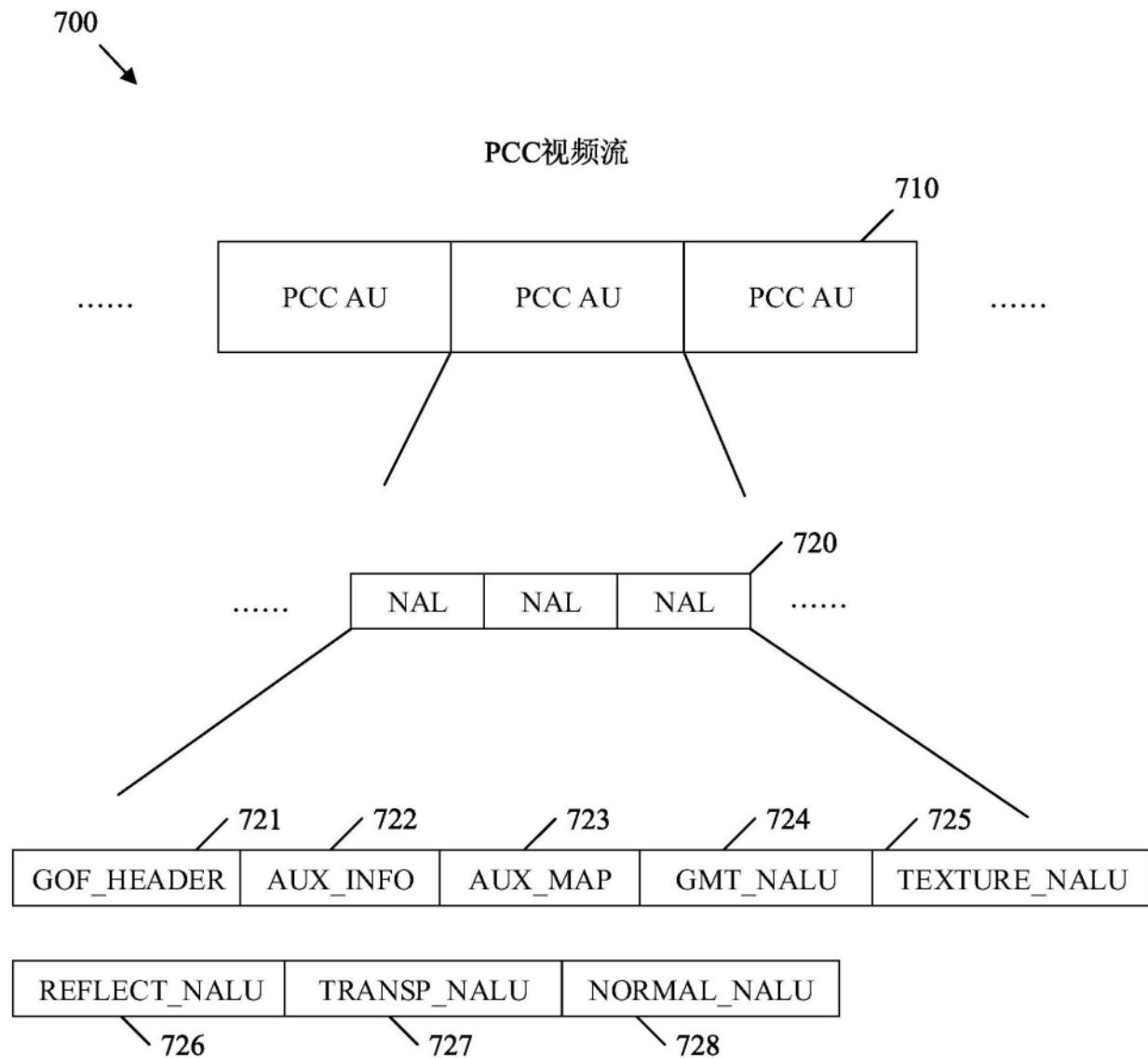


图7

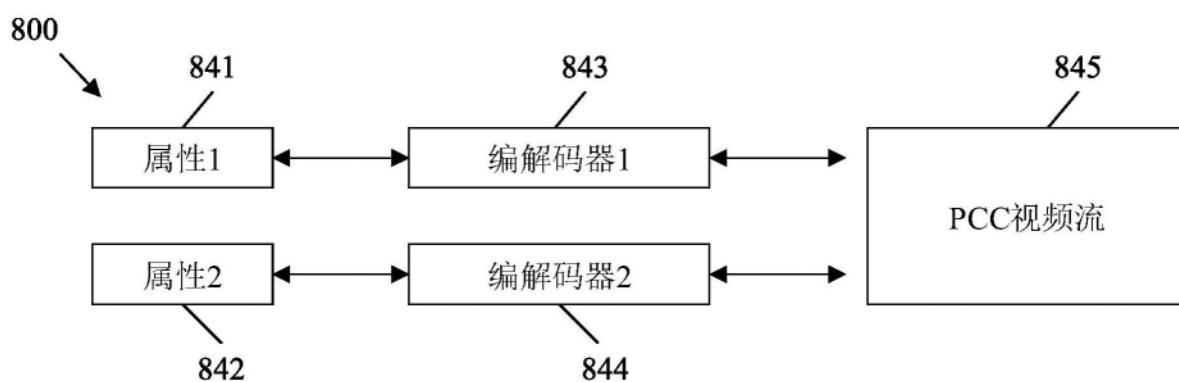


图8

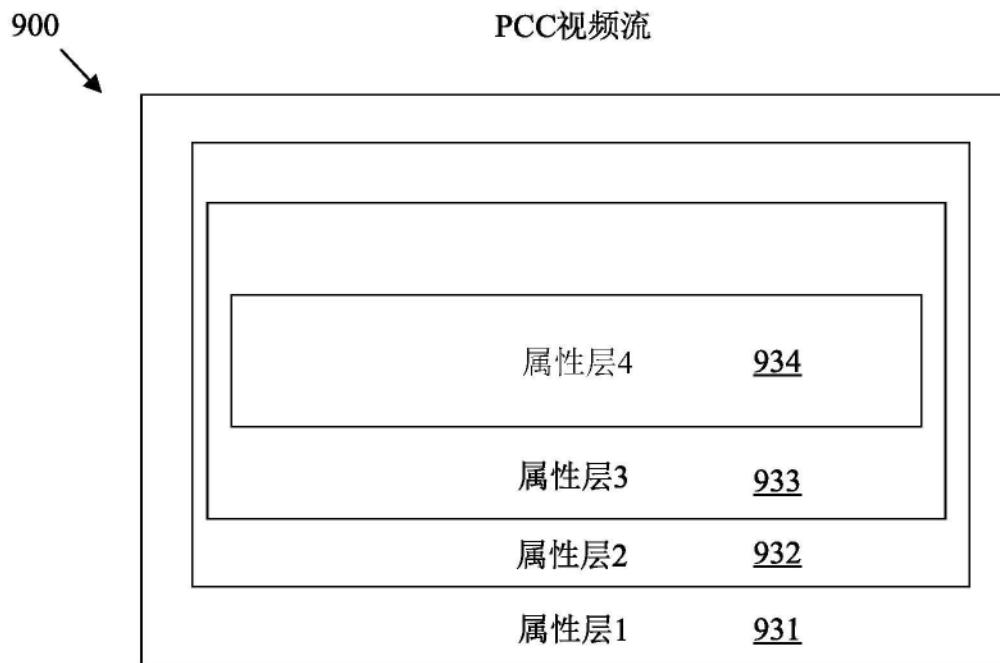


图9

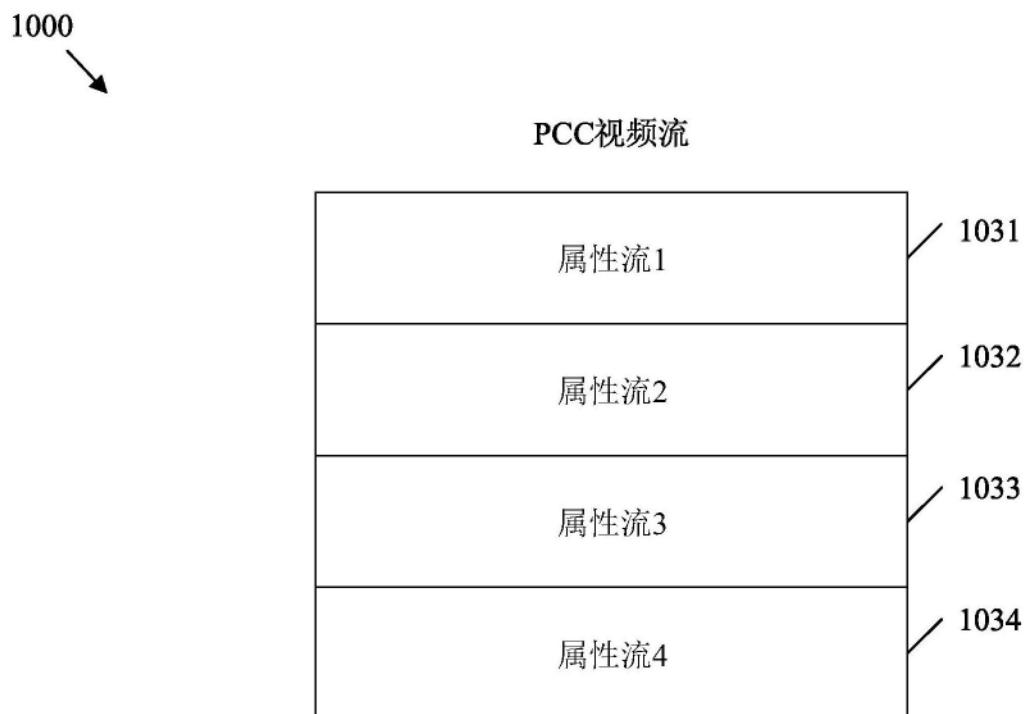


图10

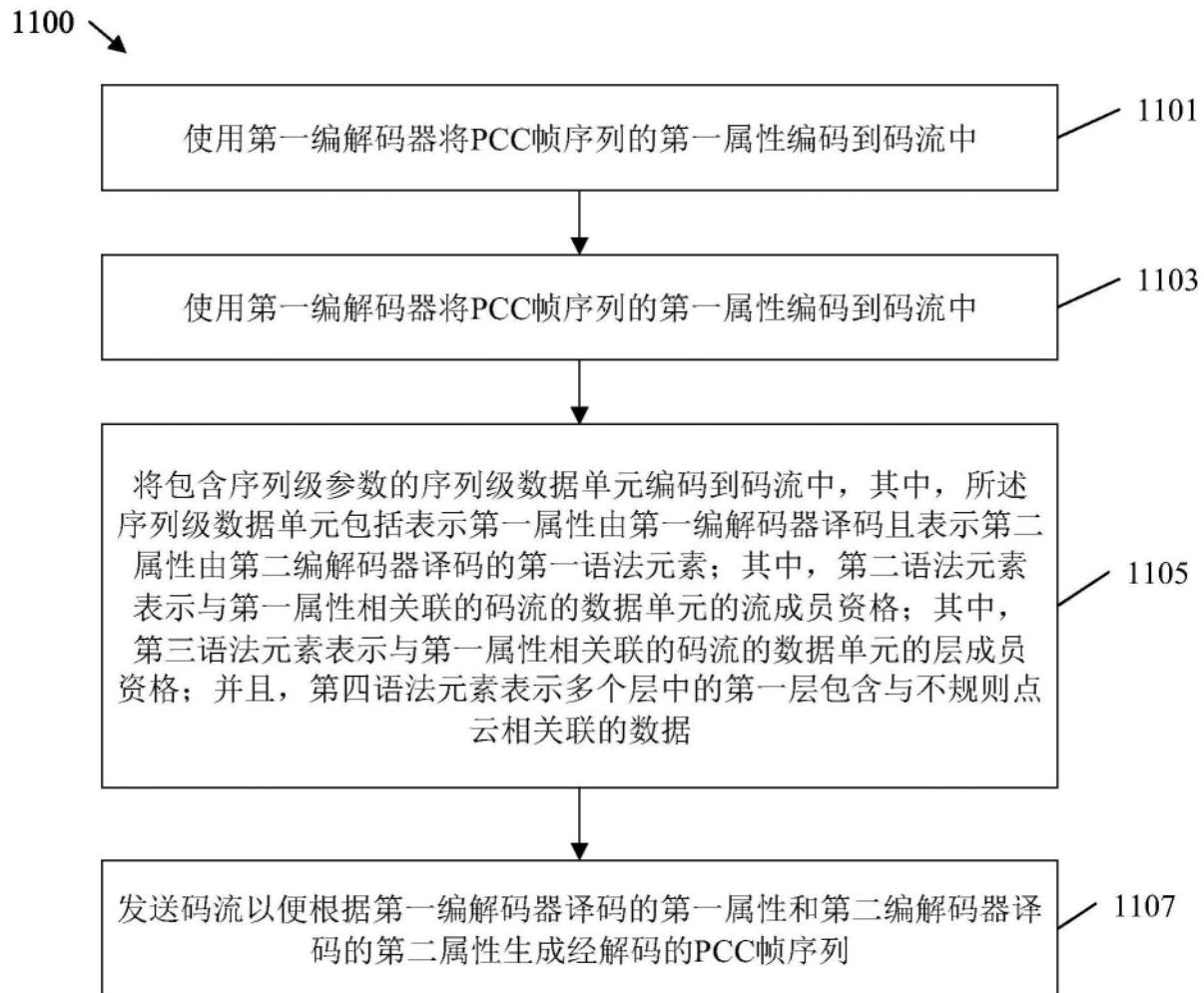


图11

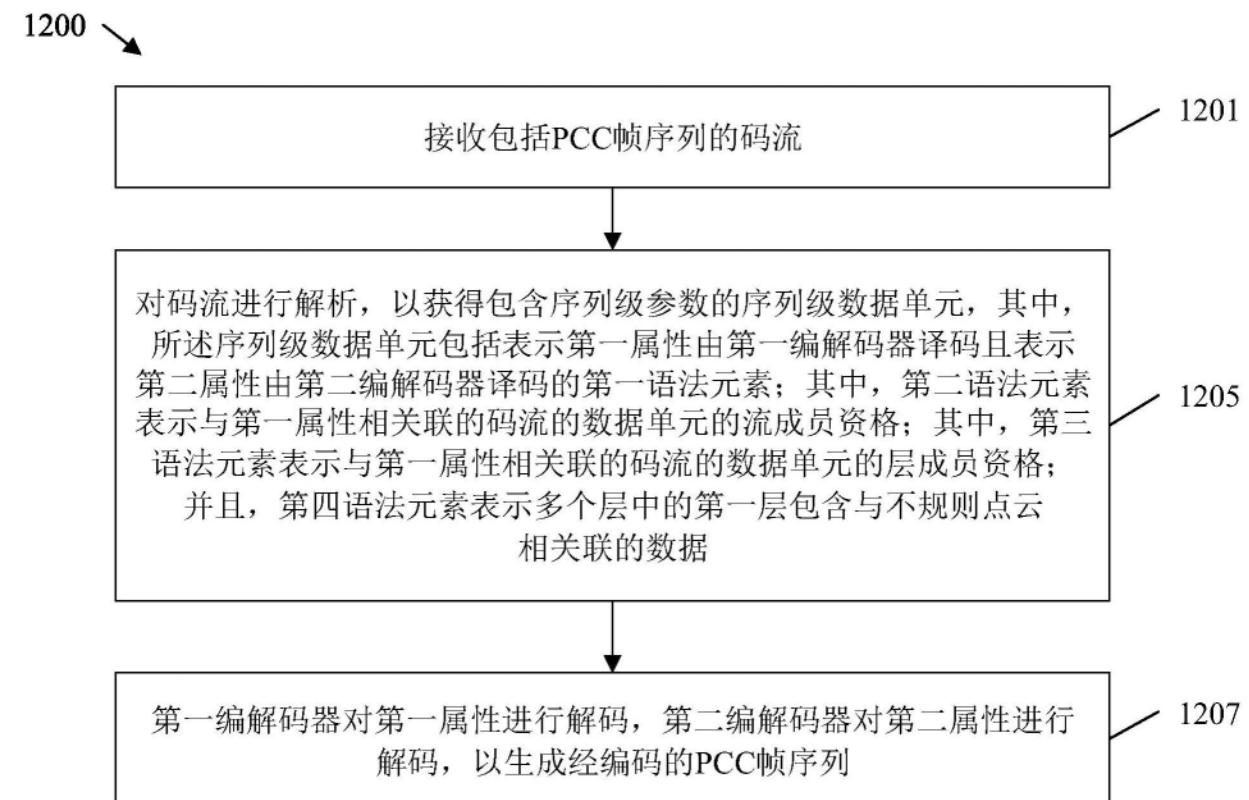


图12

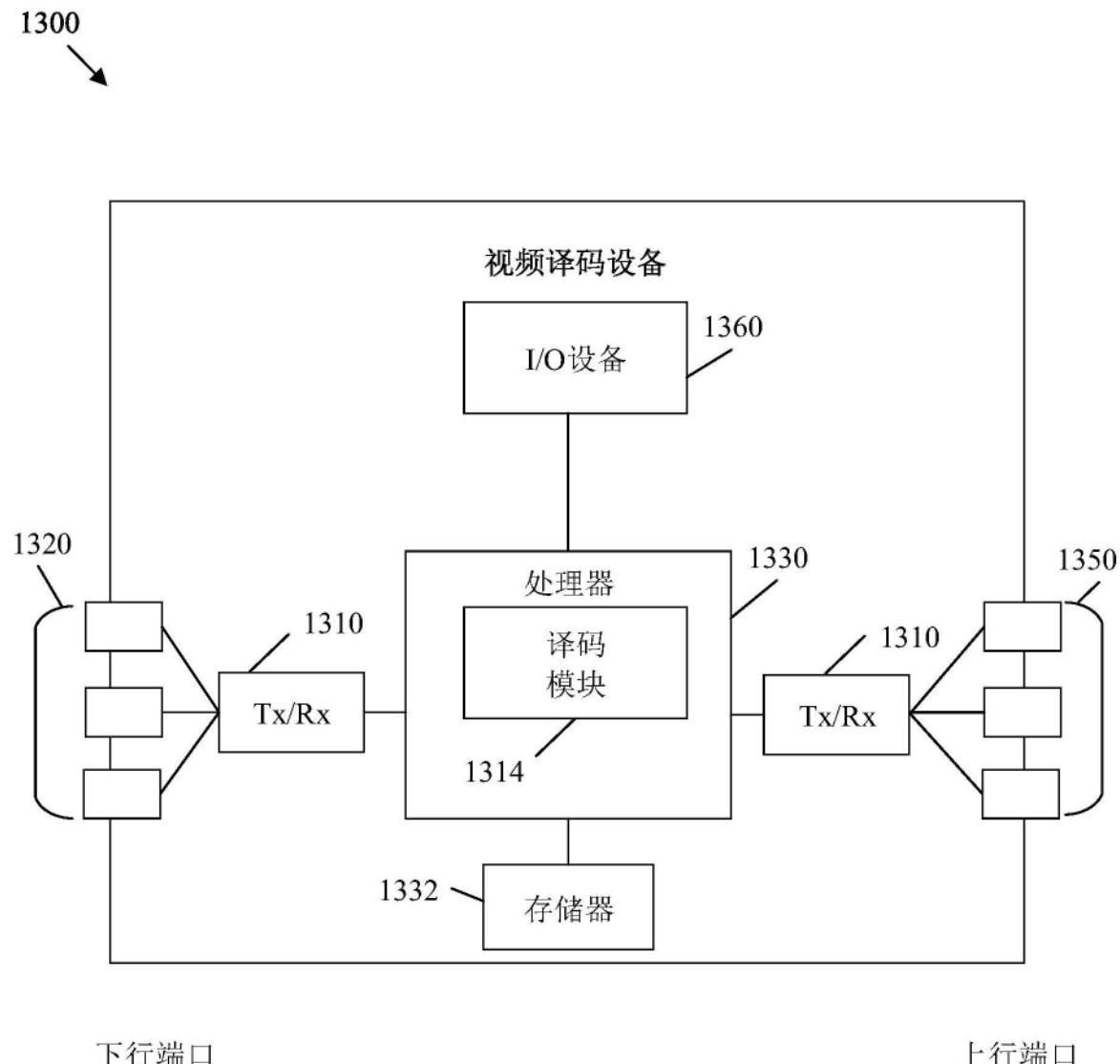


图13

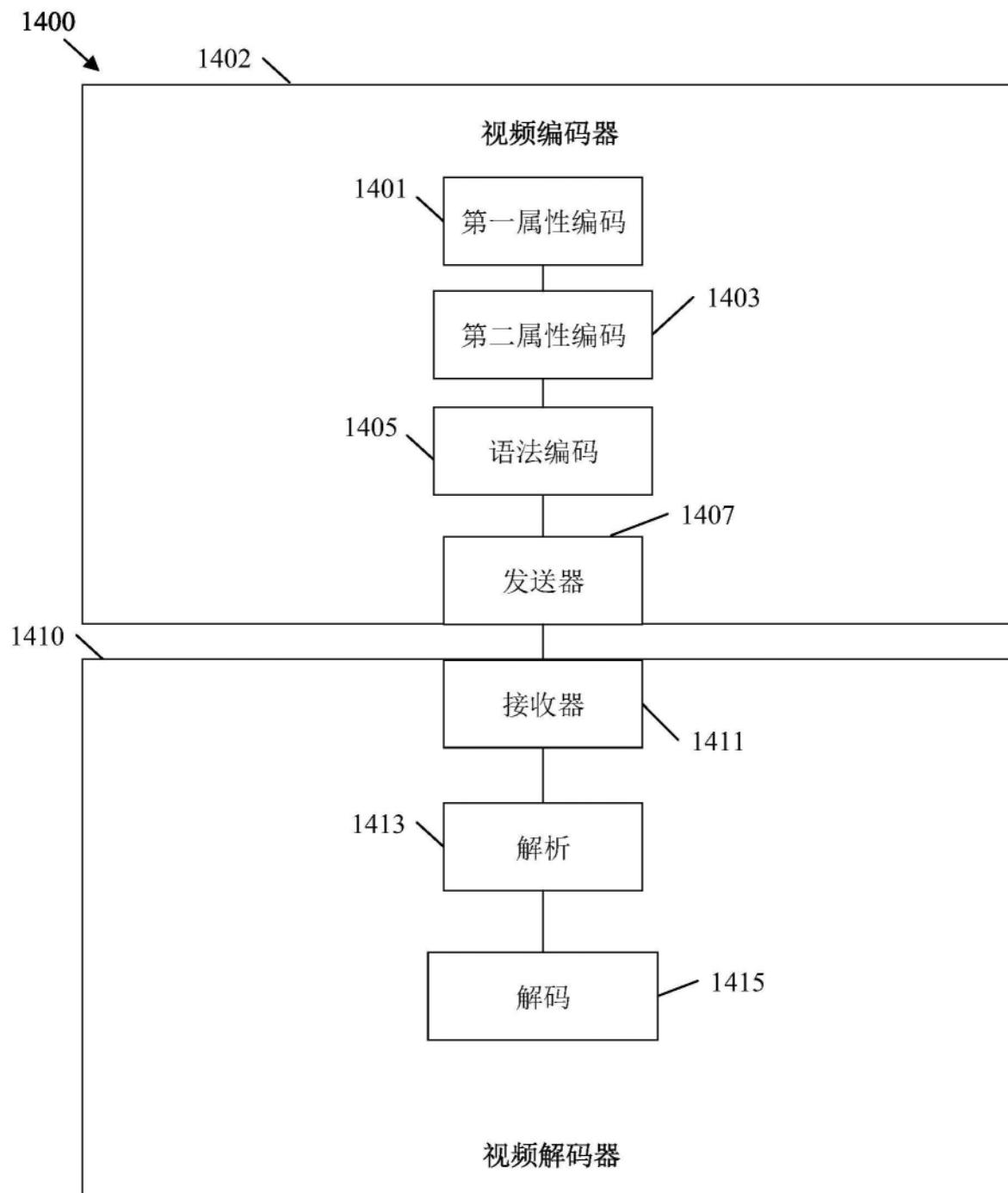


图14

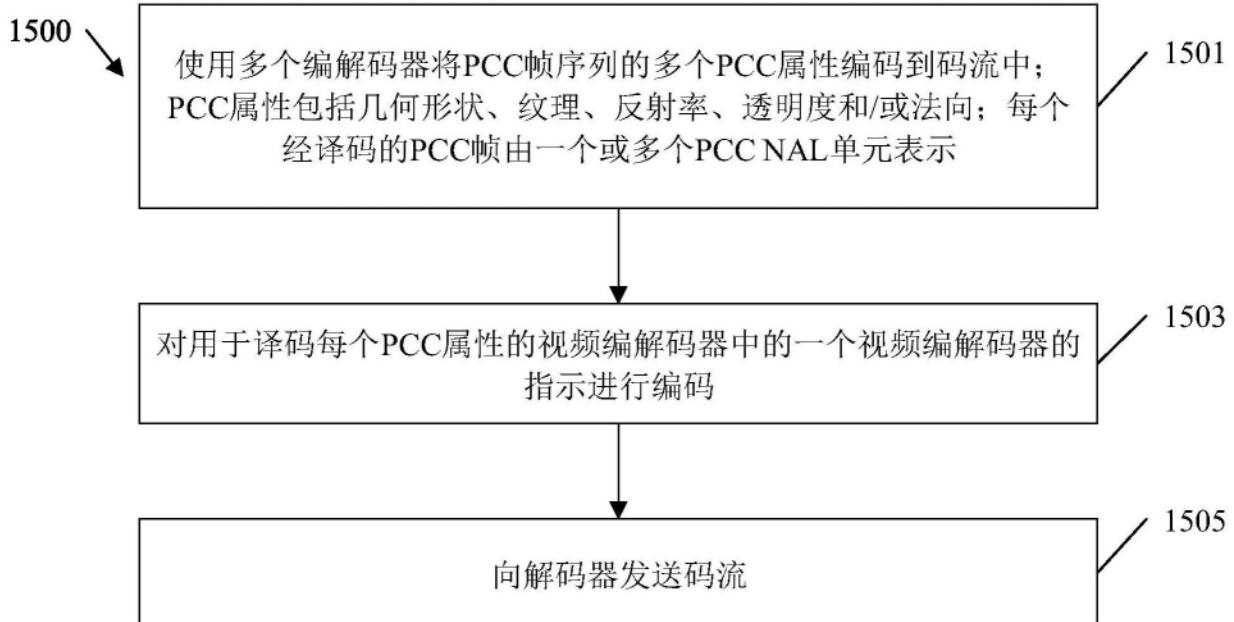


图15

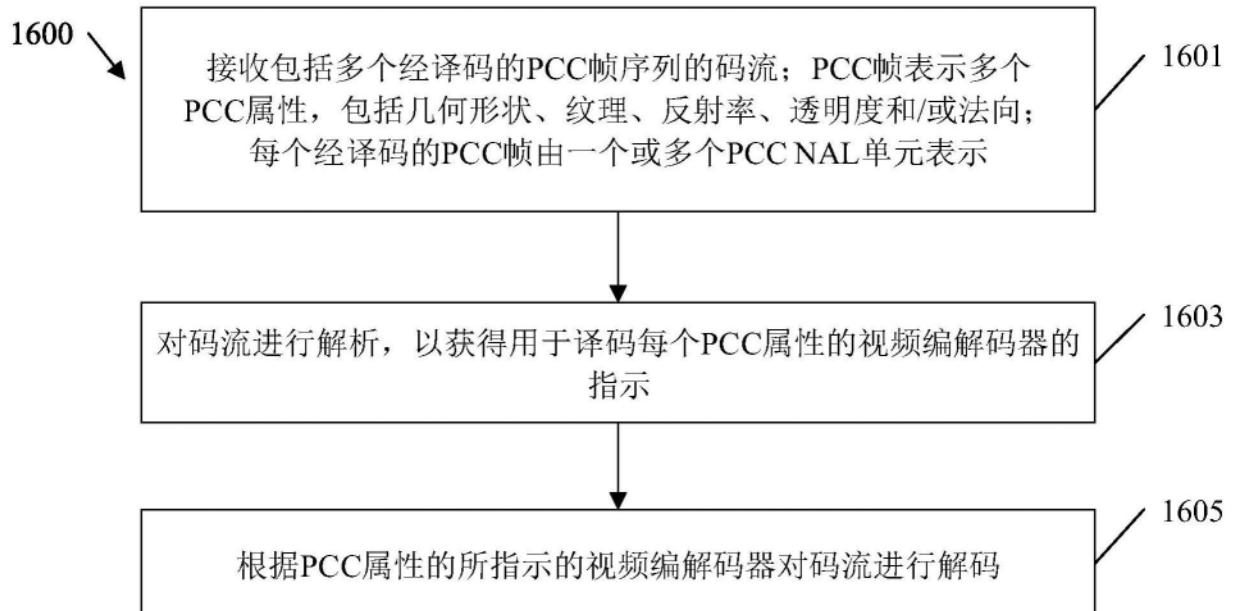


图16