



(12) 发明专利

(10) 授权公告号 CN 114223199 B

(45) 授权公告日 2024. 07. 19

(21) 申请号 202180003503.2
(22) 申请日 2021.04.02
(65) 同一申请的已公布的文献号
申请公布号 CN 114223199 A
(43) 申请公布日 2022.03.22
(30) 优先权数据
63/005,345 2020.04.05 US
63/010,272 2020.04.15 US
63/024,140 2020.05.13 US
63/027,848 2020.05.20 US
63/028,222 2020.05.21 US
17/220,672 2021.04.01 US
(85) PCT国际申请进入国家阶段日
2021.11.22
(86) PCT国际申请的申请数据
PCT/US2021/025559 2021.04.02
(87) PCT国际申请的公布数据
W02021/207023 EN 2021.10.14

(73) 专利权人 腾讯美国有限责任公司
地址 美国加利福尼亚州帕洛阿尔托公园大道2747号
(72) 发明人 崔秉斗 刘杉 史蒂芬·文格尔
(74) 专利代理机构 北京德琦知识产权代理有限公司 11018
专利代理师 张路 王琦
(51) Int.Cl.
H04N 19/176 (2006.01)
H04N 19/105 (2006.01)
H04N 19/159 (2006.01)
H04N 19/70 (2006.01) (续)

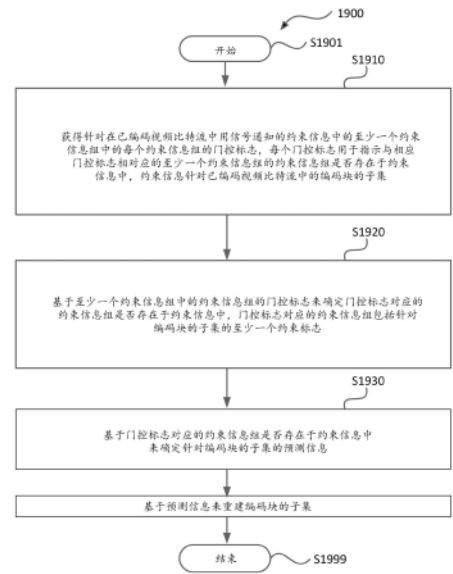
(56) 对比文件
Karam Naser.《AhG 9: Cleanup of Constraint Flags》.《Joint Video Experts Team (JVET) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11t: JVET-R0173-r2》.2020,正文第1-7页. (续)

审查员 师佳彬
权利要求书3页 说明书57页 附图43页

(54) 发明名称
用于视频编解码的方法和设备

(57) 摘要
本公开的各方面提供一种用于视频编码/解码的方法和设备。设备包括处理电路,该处理电路:获得针对在已编码视频比特流中,用信号通知的约束信息中的至少一个约束信息组中的每个约束信息组的门控标志。每个门控标志用于指示与相应门控标志相对应的至少一个约束信息组中的约束信息组是否存在于约束信息中。约束信息针对已编码视频比特流中的编码块的子集。基于门控标志来确定该门控标志对应的约束信息组是否存在于约束信息中。约束信息组包括针对编码块的子集的至少一个约束标志。基于门控标志对应的约束信息组是否存在于约束信息中来确定针对编码块的子集的预测信息。基于预测

信息来重建编码块的子集。



[接上页]

(51) Int.Cl.

H04N 19/44 (2006.01)

(56) 对比文件

Philippe Bordes.《AHG17: On general constraint information》.《Joint Video Experts Team (JVET) of ITU-T SG 16 WP 3

and ISO/IEC JTC 1/SC 29/WG 11: JVET-00417-v1》.2019,正文第1-2页.

Karam Naser.《AhG 9: Cleanup of Constraint Flags》.《Joint Video Experts Team (JVET) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11t: JVET-R0173-r2》.2020,正文第1-7页.

1. 一种用于在解码器中进行视频解码方法, 包括:

获得针对在已编码视频比特流中, 用信号通知的约束信息中的至少一个约束信息组中的每个约束信息组的门控标志, 每个门控标志用于指示与相应门控标志对应的所述至少一个约束信息组中的约束信息组是否存在于所述约束信息中, 所述约束信息针对所述已编码视频比特流中的编码块的子集;

基于所述至少一个约束信息组中的所述约束信息组的所述门控标志, 来确定所述门控标志对应的所述约束信息组是否存在于所述约束信息中, 所述门控标志对应的所述约束信息组包括针对所述编码块的子集的至少一个约束标志;

所述约束信息中包括偏移量和长度, 并且, 基于所述偏移量和所述长度指示至少一个约束标志存在于所述约束信息中;

基于所述约束信息中存在的约束标志, 来确定针对所述编码块的子集的预测信息; 以及

基于所述预测信息来重建所述编码块的子集。

2. 根据权利要求1所述的方法, 其中,

所述门控标志对应的所述约束信息组被确定为存在于所述约束信息中; 以及

所述确定所述预测信息包括基于所述门控标志对应的所述约束信息组中的所述多个约束标志, 来确定针对所述编码块的子集的所述预测信息。

3. 根据权利要求1所述的方法, 其中,

所述门控标志对应的所述约束信息组被确定为不存在于所述约束信息中。

4. 根据权利要求1所述的方法, 其中,

每个门控标志指示与所述相应门控标志对应的所述约束信息组不存在于所述约束信息中, 并且在所述约束信息中不存在与相应门控标志对应的所述约束信息组中的约束标志。

5. 根据权利要求1所述的方法, 其中,

所述约束信息存在于针对至少一个图片的高级别语法中, 并且所述编码块的子集包括所述至少一个图片中的编码块。

6. 根据权利要求1所述的方法, 其中,

所述至少一个约束信息组中的一个或多个约束信息组存在于所述已编码视频比特流中,

根据预定的顺序用信号通知所述至少一个约束信息组中的所述一个或多个约束信息组中的多个约束标志, 并且

所述确定所述预测信息进一步包括:

对所述约束信息中存在的约束标志进行游程解码; 以及

基于所述约束信息中存在的约束标志来确定针对所述编码块的子集的所述预测信息。

7. 根据权利要求2所述的方法, 其中,

所述门控标志对应的所述约束信息组中的所述多个约束标志是根据预定的顺序用信号通知的, 并且

确定所述预测信息进一步包括: 对所述约束信息中存在的约束标志进行游程解码。

8. 根据权利要求1所述的方法, 其中,

所述至少一个约束信息组中的一个或多个约束信息组存在于所述已编码视频比特流中,

所述约束信息中的一个或多个语法元素是字节对齐的,所述一个或多个语法元素包括针对所述至少一个约束信息组中的每个约束信息组的所述门控标志、以及所述至少一个约束信息组中的所述一个或多个约束信息组中的多个约束标志。

9. 一种视频编码方法,应用于包括本地解码器的编码器中,所述方法包括:

获得针对在已编码视频比特流中,用信号通知的约束信息中的至少一个约束信息组中的每个约束信息组的门控标志,每个门控标志用于指示与相应门控标志对应的所述至少一个约束信息组中的约束信息组是否存在于所述约束信息中,所述约束信息针对所述已编码视频比特流中的编码块的子集;

基于所述至少一个约束信息组中的所述约束信息组的所述门控标志,来确定所述门控标志对应的所述约束信息组是否存在于所述约束信息中,所述门控标志对应的所述约束信息组包括针对所述编码块的子集的至少一个约束标志;

所述约束信息中包括偏移量和长度,并且,基于所述偏移量和所述长度指示至少一个约束标志存在于所述约束信息中;

基于所述约束信息中存在的约束标志,来确定针对所述编码块的子集的预测信息;以及

基于所述预测信息来重建所述编码块的子集。

10. 一种用于视频解码的设备,包括处理电路,所述处理电路被配置为:获得针对在已编码视频比特流中,用信号通知的约束信息中的至少一个约束信息组中的每个约束信息组的门控标志,每个门控标志用于指示与相应门控标志对应的所述至少一个约束信息组中的约束信息组是否存在于所述约束信息中,所述约束信息针对所述已编码视频比特流中的编码块的子集;

基于所述至少一个约束信息组中的所述约束信息组的所述门控标志,来确定所述门控标志对应的所述约束信息组是否存在于所述约束信息中,所述门控标志对应的所述约束信息组包括针对所述编码块的子集的多个约束标志;

所述约束信息中包括偏移量和长度,并且,基于所述偏移量和所述长度指示至少一个约束标志存在于所述约束信息中;

基于所述约束信息中存在的约束标志,来确定针对所述编码块的子集的预测信息;以及

基于所述预测信息来重建所述编码块的子集。

11. 根据权利要求10所述的设备,其中,

所述门控标志对应的所述约束信息组被确定为存在于所述约束信息中;以及

所述处理电路被配置为:基于所述门控标志对应的所述约束信息组中的所述多个约束标志来确定针对所述编码块的子集的所述预测信息。

12. 根据权利要求10所述的设备,其中,

所述门控标志对应的所述约束信息组被确定为不存在于所述约束信息中。

13. 根据权利要求10所述的设备,其中,

每个门控标志用于指示与所述相应门控标志相对应的所述约束信息组不存在于所述

约束信息中,并且在所述约束信息中不存在与相应门控标志对应的所述约束信息组中的约束标志。

14.根据权利要求10所述的设备,其中,

所述约束信息存在于针对至少一个图片的高级别语法中,并且所述编码块的子集包括所述至少一个图片中的编码块。

15.根据权利要求10所述的设备,其中,

所述至少一个约束信息组中的一个或多个约束信息组存在于所述已编码视频比特流中,

根据预定的顺序用信号通知所述至少一个约束信息组中的所述一个或多个约束信息组中的多个约束标志,并且

所述处理电路被配置为:

对所述约束信息中存在的约束标志进行游程解码;以及

基于所述约束信息中存在的约束标志来确定针对所述编码块的子集的所述预测信息。

16.根据权利要求11所述的设备,其中,

所述门控标志对应的所述约束信息组中的所述多个约束标志是根据预定的顺序用信号通知的,并且

所述处理电路被配置为:对所述约束信息中存在的约束标志进行游程解码。

17.一种存储有指令的非暂时性计算机可读介质,该指令可由至少一个处理器执行,用以执行权利要求1-9中任一项所述的方法。

用于视频编解码的方法和设备

[0001] 相关申请的交叉引用

[0002] 本申请要求于2021年4月1日提交的美国专利申请第17/220,672号“用于视频编解码的方法和设备”的优先权,该申请要求于2020年4月5日提交的美国临时申请第63/005,345号“在已编码视频流中用信号通知子图片和切片分区的方法”、于2020年4月15日提交的美国临时申请第63/010,272号“在已编码视频流中用信号通知图片头和切片头的方法”、于2020年5月13日提交的美国临时申请第63/024,140号“在已编码视频流中使用约束标志来控制编码工具的技术”、2020年5月20日提交的美国临时申请第63/027,848号“用于在多层视频流中用信号通知场编解码信息的技术”、以及于2020年5月21日提交的美国临时申请第63/028,222号“用于在已编码视频流中条件性地用信号通知约束标志的技术”的优先权。在先申请的全部公开内容通过引用其整体被并入本文。

技术领域

[0003] 本公开描述了总体上涉及视频编解码的实施例。

背景技术

[0004] 本文所提供的背景描述旨在总体上呈现本公开的背景。在背景技术部分以及本说明书的各个方面中所描述的目前已署名的发明人的工作所进行的程度,并不表明其在本公开提交时作为现有技术,且从未明示或暗示其被承认为本公开的现有技术。

[0005] 可以使用具有运动补偿的图片间预测来执行视频编码和解码。未压缩的数字视频可以包括一系列图片,每个图片具有例如 1920×1080 亮度样本及相关色度样本的空间维度。所述系列图片可以具有固定的或可变的图片速率(也非正式地称为帧率),例如每秒60张图片或60Hz。未压缩的视频具有特定的比特率要求。例如,每个样本8比特的1080p60 4:2:0视频(60Hz帧率下 1920×1080 亮度样本分辨率)要求接近1.5Gbit/s带宽。一小时这样的视频就需要超过600GB的存储空间。

[0006] 视频编码和解码的一个目的是通过压缩减少输入视频信号的冗余。压缩可以帮助降低对上述带宽和/或存储空间的要求,在某些情况下可降低两个或更多数量级。无损压缩和有损压缩、以及两者的组合均可采用。无损压缩是指从压缩的原始信号中重建原始信号精确副本的技术。当使用有损压缩时,重建信号可能与原始信号不完全相同,但是原始信号和重建信号之间的失真足够小,使得重建信号可用于预期应用。有损压缩广泛应用于视频。容许的失真量取决于应用。例如,相比于电视应用的用户,某些消费流媒体应用的用户可以容忍更高的失真。可实现的压缩比反映出:较高的允许/容许失真可产生较高的压缩比。

[0007] 视频编码器和解码器可以利用几大类技术,例如包括:运动补偿、变换、量化和熵编码。

[0008] 视频编解码器技术可以包括已知的帧内编解码技术。在帧内编解码中,在不参考先前重建的参考图片的样本或其它数据的情况下表示样本值。在一些视频编解码器中,图片在空间上被细分为样本块。当所有的样本块都以帧内模式编解码时,该图片可以为帧内

图片。帧内图片及其衍生(例如独立解码器刷新图片)可用于复位解码器状态,并且因此可用作已编码视频比特流和视频会话中的第一图片,或用作静止图像。帧内块的样本可用于变换,并且可以在熵编码之前量化变换系数。帧内预测可以是一种使预变换域中的样本值最小化的技术。在某些情况下,变换后的DC值越小,且AC系数越小,则在给定的量化步长尺寸下表示熵编码后的块所需的比特越少。

[0009] 如同从诸如MPEG-2(运动图像专家组-2)编解码技术中所获知的,传统帧内编解码不使用帧内预测。然而,一些较新的视频压缩技术包括:试图从例如周围样本数据和/或元数据中得到数据块的技术,其中周围样本数据和/或元数据是在空间相邻块的编码/解码期间、且在解码顺序之前获得的。这种技术后来被称为“帧内预测”技术。需要注意的是,至少在某些情形下,帧内预测使用仅来自正在重建的当前图片的参考数据,而不使用来自参考图片的参考数据。

[0010] 可以存在许多不同形式的帧内预测。当在给定的视频编解码技术中可以使用超过一种这样的技术时,所使用的技术可以按帧内预测模式进行编解码。在某些情形下,模式可以具有子模式和/或参数,且这些模式可以单独编解码或包含在模式码字中。给定模式/子模式/参数组合使用哪个码字会影响通过帧内预测获得的编解码效率增益,因此用于将码字转换成比特流的熵编码技术也会出现这种情况。

[0011] H.264引入了一种帧内预测模式,其在H.265中进行了改进,且在诸如联合开发模型(JEM)、通用视频编解码(VVC)、基准集合(BMS)等更新的编解码技术中进一步被改进。使用属于已经可用的样本的相邻样本值可以形成预测块。将相邻样本的样本值按照某一方复制到预测块中。对所使用方向的引用可以被编码在比特流中,或者本身可以被预测。

[0012] 参考图1A,右下方描绘了来自H.265的33个可能的预测方向(对应于35个帧内模式的33个角模式)中已知的九个预测方向的子集。箭头会聚的点(101)表示正在被预测的样本。箭头表示样本正在被预测的方向。例如,箭头(102)表示根据右上方与水平方向成45度角的一个或多个样本,预测样本(101)。类似地,箭头(103)表示根据左下方与水平方向成22.5度角的一个或多个样本,预测样本(101)。

[0013] 仍然参考图1A,在左上方示出了一个包括 4×4 个样本的正方形块(104)(由粗虚线表示)。正方形块(104)由16个样本组成,每个样本用“S”、以及其在Y维度(例如行索引)上的位置和在X维度(例如列索引)上的位置来标记。例如,样本S21是Y维度上的第二个样本(从上方开始)和X维度上的第一个样本(从左侧开始)。类似地,样本S44在Y维度和X维度上都是块(104)中的第四个样本。由于该块为 4×4 大小的样本,因此S44位于右下角。还示出了遵循类似编号方案的参考样本。参考样本用“R”、以及其相对于块(104)的Y位置(例如,行索引)和X位置(例如,列索引)来标记。在H.264与H.265中,预测样本与正在重建的块相邻,因此不需要使用负值。

[0014] 通过从相邻样本复制参考样本值,可以进行帧内图片预测,其中相邻样本是由信号通知的预测方向所决定的。例如,假设已编码视频比特流包括信令,对于该块,该信令指示与箭头(102)一致的预测方向,即,根据右上方与水平方向成45度角的一个或多个预测样本来预测样本。在这种情况下,根据同一参考样本R05,预测样本S41、S32、S23和S14。然后,根据样本R08,预测样本S44。

[0015] 在某些情况下,例如通过内插,可以合并多个参考样本的值,以便计算参考样本,

尤其是当方向不能被45度整除时。

[0016] 随着视频编解码技术的发展,可能的方向的数量已经增加了。在H.264(2003年)中,可以表示九种不同的方向。在H.265(2013年)和JEM/VVC/BMS中增加到了33个,而在此公开时,可以支持多达65个方向。已经进行了实验来识别最可能的方向,并且熵编码中的某些技术被用来使用少量比特来表示那些可能的方向,对于较不可能的方向则接受某些代价。此外,有时可以根据在相邻的、已经解码的块中所使用的相邻方向来预测方向本身。

[0017] 图1B示出了一种示意图(180),其描述了根据JEM的65种帧内预测方向,以说明随着时间的推移预测方向的数量增加。

[0018] 从帧内预测方向到已编码视频比特流中的表示方向的比特的映射可以因视频编解码技术的不同而不同,例如,其范围可以从对帧内预测模式的预测方向到码字的简单直接映射,到包括最可能模式的复杂自适应方案以及类似技术。然而,在所有情况下,视频内容中可能存在某些方向,其在统计学上比其它方向更不可能出现。

[0019] 运动补偿可以是一种有损压缩技术,且可涉及如下技术:来自先前重建的图片或重建图片一部分(参考图片)的样本数据块在空间上按运动矢量(下文称为MV)指示的方向移位后,用于新重建的图片或图片部分的预测。在某些情况下,参考图片可与当前正在重建的图片相同。MV可具有两个维度X和Y,或者三个维度,其中第三个维度表示正在使用的参考图片(后者间接地可以是时间维度)。

[0020] 在一些视频压缩技术中,应用于样本数据的某个区域的MV可根据其它MV来预测,例如根据与正在重建的区域空间相邻的样本数据的另一个区域相关的、且按解码顺序在该MV前面的那些MV。这样做可以大大减少编解码MV所需的数据量,从而消除冗余并增加压缩量。MV预测可以有效地进行,例如,因为在对从相机导出的输入视频信号(称为自然视频)进行编解码时,存在一种统计上的可能性,即面积大于单个MV适用区域的区域,会朝着类似的方向移动,因此,在某些情况下,可以使用从相邻区域的MV导出的相似运动矢量进行预测。这导致针对给定区域发现的MV与根据周围MV预测的MV相似或相同,并且在熵编解码之后,又可以用比直接编解码MV时使用的比特数更少的比特数来表示。在某些情况下,MV预测可以是对从原始信号(即样本流)导出的信号(即MV)进行无损压缩的示例。在其它情况下,例如由于根据几个周围MV计算预测值时产生的取整误差,MV预测本身可能是有损的。

[0021] H.265/HEVC(ITU-T H.265建议书,“高效视频编解码”,2016年12月)描述了各种MV预测机制。在H.265提供的多种MV预测机制中,本公开描述的是下文称为“空间合并”的技术。

[0022] 参考图2,当前块(201)包括编码器在运动搜索过程中发现的样本,所述样本可以根据已在空间上移位的相同大小的先前块进行预测。不直接对MV进行编解码,而是使用与五个周围样本中的任何一个相关联的MV,从与一个或多个参考图片相关联的元数据中导出该MV,例如,从最近的(按解码顺序)参考图片中导出该MV。其中,五个周围样本分别用A0、A1和B0、B1、B2(分别为202到206)表示。在H.265中,MV预测可使用相邻块正在使用的同一参考图片的预测值。

发明内容

[0023] 本公开的各方面提供用于视频编码/解码的方法和设备。在一些示例中,用于视频

解码的设备包括处理电路。处理电路被配置为：获得针对在已编码视频比特流中用信号通知的约束信息中的至少一个约束信息组中的每个约束信息组的门控标志。每个门控标志可以指示与相应门控标志相对应的至少一个约束信息组中的约束信息组是否存在于约束信息中。约束信息可以针对已编码视频比特流中的编码块的子集。处理电路被配置为：基于至少一个约束信息组的约束信息组的门控标志来确定门控标志对应的约束信息组是否存在于约束信息中，门控标志对应的约束信息组包括针对编码块的子集的至少一个约束标志。所述约束信息中包括偏移量和长度，并且，处理电路被配置为基于所述偏移量和所述长度指示至少一个约束标志存在于所述约束信息中；处理电路被配置为基于所述约束信息中存在的约束标志来确定针对编码块的子集的预测信息。处理电路被配置为基于预测信息来重建编码块的子集。

[0024] 在实施例中，门控标志对应的约束信息组被确定为存在于约束信息中。处理电路被配置为基于门控标志对应的约束信息组中的至少一个约束标志来确定针对编码块的子集的预测信息。

[0025] 在实施例中，门控标志对应的约束信息组被确定为不存在于约束信息中。

[0026] 在实施例中，每个门控标志用于指示与相应门控标志相对应的约束信息组不存在于约束信息中，并且约束信息中不存在与相应门控标志对应的所述约束信息组中的约束标志。

[0027] 在实施例中，约束信息存在于针对至少一个图片的高级别语法中，并且编码块的子集包括至少一个图片中的编码块。

[0028] 在实施例中，至少一个约束信息组中的一个或多个约束信息组存在于已编码视频比特流中。根据预定的顺序用信号通知至少一个约束信息组中的一个或多个约束信息组中的多个约束标志。处理电路被配置为对约束信息中存在的约束标志进行游程解码，以及基于约束信息中存在的约束标志来确定针对编码块的子集的预测信息。

[0029] 在实施例中，门控标志对应的约束信息组中的至少一个约束标志包括根据预定的顺序用信号通知的多个约束标志。处理电路被配置为对约束信息中存在的约束标志进行游程解码。

[0030] 在实施例中，至少一个约束信息组中的一个或多个约束信息组存在于已编码视频比特流中。对于至少一个约束信息组中的一个或多个约束信息组中的每个约束信息组，偏移量和长度存在于约束信息中，指示至少一个约束信息组中的一个或多个约束信息组中的相应约束信息组中的至少一个约束标志存在于约束信息中。在已编码视频比特流中用信号通知偏移量的数量。

[0031] 在实施例中，至少一个约束信息组中的一个或多个约束信息组存在于已编码视频比特流中。约束信息中的一个或多个语法元素是字节对齐的。一个或多个语法元素可包括针对至少一个约束信息组中的每个约束信息组的门控标志、以及至少一个约束信息组中的一个或多个约束信息组中的多个约束标志。

[0032] 本公开的各方面还提供了一种视频编码方法，应用于包括本地解码器的编码器中，所述方法包括：

[0033] 获得针对在已编码视频比特流中，用信号通知的约束信息中的至少一个约束信息组中的每个约束信息组的门控标志，每个门控标志用于指示与相应门控标志对应的所述至

少一个约束信息组中的约束信息组是否存在于所述约束信息中,所述约束信息针对所述已编码视频比特流中的编码块的子集;

[0034] 基于所述至少一个约束信息组中的所述约束信息组的所述门控标志,来确定所述门控标志对应的所述约束信息组是否存在于所述约束信息中,所述门控标志对应的所述约束信息组包括针对所述编码块的子集的至少一个约束标志;

[0035] 所述约束信息中包括偏移量和长度,并且,基于所述偏移量和所述长度指示至少一个约束标志存在于所述约束信息中;

[0036] 基于所述约束信息中存在的约束标志,来确定针对所述编码块的子集的预测信息;以及

[0037] 基于所述预测信息来重建所述编码块的子集。

[0038] 本公开的各方面还提供了一种存储有指令的非暂时性计算机可读介质,当该指令由计算机执行以用于视频解码时,使计算机执行用于视频解码的方法。

附图说明

[0039] 根据以下详细描述和附图,所公开的主题的其他特征、性质和各种优点将进一步明确,其中:

[0040] 图1A是帧内预测模式的示例性子集的示意图。

[0041] 图1B是示例性帧内预测方向的图示。

[0042] 图2是一个示例中的当前块及其周围的空间合并候选的示意图。

[0043] 图3是根据实施例的通信系统(300)的简化框图的示意图。

[0044] 图4是根据另一实施例的通信系统(400)的简化框图的示意图。

[0045] 图5是根据实施例的解码器的简化框图的示意图。

[0046] 图6是根据实施例的编码器的简化框图的示意图。

[0047] 图7示出了根据另一实施例的编码器的框图。

[0048] 图8示出了根据另一实施例的解码器的框图。

[0049] 图9示出了根据本公开的实施例的用于发信号通知自适应分辨率改变(ARC)参数的示例。

[0050] 图10示出了根据本公开的实施例的ARC参数信令的示例。

[0051] 图11示出了根据本公开的实施例的档位、图块、等级(PTL)语法元素的集合的示例。

[0052] 图12示出了根据本公开的实施例的通用约束信息的示例。

[0053] 图13A至图13B示出了根据本公开的实施例的PTL语法元素的集合的示例。

[0054] 图14A至图14B示出了根据本公开的实施例的示例性通用约束信息语法结构。

[0055] 图15A至图15D示出了根据本公开的实施例的示例性通用约束信息语法结构。

[0056] 图16示出了根据本公开的实施例的PTL语法结构中的PTL语法元素的集合的示例。

[0057] 图17示出了根据本公开的实施例的用于指示序列参数集中的场编解码信息的语法表的示例。

[0058] 图18示出了根据本公开的实施例的用于指示视频参数集中的场编解码信息的语法表的示例。

- [0059] 图19示出了概述根据本公开的实施例的过程的流程图。
- [0060] 图20示出了根据本公开的实施例的语法表的示例。
- [0061] 图21示出了根据本公开的实施例的对每存取单元的图片顺序计数 (POC) 周期和存取单元计数值进行解析和解码的流程图。
- [0062] 图22示出了根据本公开的实施例的用于具有自适应分辨率改变的可缩放性的预测结构的示例。
- [0063] 图23示出了根据本公开的实施例的包括多层子图片的视频比特流结构的示例。
- [0064] 图24示出了根据本公开的实施例的具有增强分辨率的所选子图片的显示的示意图。
- [0065] 图25示出了根据本公开的实施例的用于包括多层子图片的视频比特流的解码和显示过程的框图。
- [0066] 图26示出了根据本公开的实施例的具有子图片的增强层的360度视频显示的示意图。
- [0067] 图27示出了根据本公开的实施例的子图片的布局信息以及对应的层和图片预测结构的示例。
- [0068] 图28示出了根据本公开的实施例的具有局部区域的空间可缩放性形态的子图片的布局信息以及对应的层和图片预测结构的示例。
- [0069] 图29A至图29B示出了根据本公开的实施例的子图片布局信息的语法表的示例。
- [0070] 图30示出了根据本公开的实施例的子图片布局信息的SEI消息的语法表的示例。
- [0071] 图31示出了根据本公开的实施例的用于指示输出层以及输出层集的PTL信息的语法表的示例。
- [0072] 图32示出了根据本公开的实施例的用于指示输出层集的输出层模式的语法表的示例。
- [0073] 图33示出了根据本公开的实施例的用于指示每个输出层集的每个层的目前子图片的语法表的示例。
- [0074] 图34示出了根据本公开的实施例的视频参数集的语法表的示例。
- [0075] 图35示出了根据本公开的实施例的用于指示具有输出层集模式的输出层集的语法表的示例。
- [0076] 图36是根据实施例的计算机系统的示意图。

具体实施方式

[0077] 图3示出了根据本公开实施例的通信系统 (300) 的简化框图。通信系统 (300) 包括多个终端装置, 所述终端装置可通过例如网络 (350) 彼此通信。举例来说, 通信系统 (300) 包括通过网络 (350) 互连的第一对终端装置 (310) 和 (320)。在图3的实施例中, 第一对终端装置 (310) 和 (320) 执行单向数据传输。举例来说, 终端装置 (310) 可对视频数据 (例如由终端装置 (310) 采集的视频图片流) 进行编码以通过网络 (350) 传输到另一终端装置 (320)。已编码的视频数据以一个或多个已编码视频比特流形式传输。终端装置 (320) 可从网络 (350) 接收已编码视频数据, 对已编码视频数据进行解码以恢复视频数据, 并根据恢复的视频数据显示视频图片。单向数据传输在媒体服务等应用中是较常见的。

[0078] 在另一实施例中,通信系统(300)包括执行已编码视频数据的双向传输的第二对终端装置(330)和(340),所述双向传输可例如在视频会议期间发生。对于双向数据传输,在一个示例中,终端装置(330)和(340)中的每个终端装置可对视频数据(例如由终端装置采集的视频图片流)进行编码,以通过网络(350)传输到终端装置(330)和(340)中的另一终端装置。终端装置(330)和(340)中的每个终端装置还可接收由终端装置(330)和(340)中的另一终端装置传输的已编码视频数据,且可对所述已编码视频数据进行解码以恢复视频数据,且可根据恢复的视频数据在可访问的显示装置上显示视频图片。

[0079] 在图3的实施例中,终端装置(310)、(320)、(330)和(340)可为服务器、个人计算机和智能电话,但本公开的原理可不限于此。本公开的实施例适用于膝上型计算机、平板电脑、媒体播放器和/或专用视频会议设备。网络(350)表示在终端装置(310)、(320)、(330)和(340)之间传送已编码视频数据的任何数目的网络,包括例如有线(连线的)和/或无线通信网络。通信网络(350)可在电路交换和/或分组交换信道中交换数据。该网络可包括电信网络、局域网、广域网和/或互联网。出于本公开的目的,除非在下文中有所解释,否则网络(350)的架构和拓扑对于本公开的操作来说可能是无关紧要的。

[0080] 作为所公开的主题的实施例,图4示出了视频编码器和视频解码器在流式传输环境中的放置方式。所公开主题可同等地适用于其它支持视频的应用,包括例如视频会议、数字TV、在包括CD、DVD、存储棒等的数字介质上存储压缩视频等等。

[0081] 流式传输系统可包括采集子系统(413),所述采集子系统可包括数码相机等视频源(401),所述视频源创建未压缩的视频图片流(402)。在实施例中,视频图片流(402)包括由数码相机拍摄的样本。相较于已编码的视频数据(404)(或已编码的视频比特流),视频图片流(402)被描绘为粗线以强调高数据量的视频图片流,视频图片流(402)可由电子装置(420)处理,所述电子装置(420)包括耦接到视频源(401)的视频编码器(403)。视频编码器(403)可包括硬件、软件或软硬件组合以实现或实施如下文更详细地描述的所公开主题的各方面。相较于视频图片流(402),已编码的视频数据(404)(或已编码的视频比特流(404))被描绘为细线以强调较低数据量的已编码的视频数据(404)(或已编码的视频比特流(404)),其可存储在流式传输服务器(405)上以供将来使用。一个或多个流式传输客户端子系统,例如图4中的客户端子系统(406)和客户端子系统(408),可访问流式传输服务器(405)以检索已编码的视频数据(404)的副本(407)和副本(409)。客户端子系统(406)可包括例如电子装置(430)中的视频解码器(410)。视频解码器(410)对已编码的视频数据的传入副本(407)进行解码,且产生可在显示器(412)(例如显示屏)或另一呈现装置(未描绘)上呈现的输出视频图片流(411)。在一些流式传输系统中,可根据某些视频编码/压缩标准对已编码的视频数据(404)、视频数据(407)和视频数据(409)(例如视频比特流)进行编码。这些标准的实施例包括ITU-T H.265。在实施例中,正在开发的视频编解码标准非正式地称为通用视频编解码(Versatile Video Coding, VVC),所公开的主题可用于VVC标准的上下文中。

[0082] 应注意,电子装置(420)和电子装置(430)可包括其它组件(未示出)。举例来说,电子装置(420)可包括视频解码器(未示出),且电子装置(430)还可包括视频编码器(未示出)。

[0083] 图5示出了根据本公开实施例的视频解码器(510)的框图。视频解码器(510)可设置在电子装置(530)中。电子装置(530)可包括接收器(531)(例如接收电路)。视频解码器

(510) 可用于代替图4实施例中的视频解码器 (410)。

[0084] 接收器 (531) 可接收将由视频解码器 (510) 解码的一个或多个已编码视频序列; 在同一实施例或另一实施例中, 一次接收一个已编码视频序列, 其中每个已编码视频序列的解码独立于其它已编码视频序列。可从信道 (501) 接收已编码视频序列, 所述信道可以是通向存储已编码的视频数据的存储装置的硬件/软件链路。接收器 (531) 可接收已编码的视频数据以及其它数据, 例如, 可转发到它们各自的使用实体 (未标示) 的已编码音频数据和/或辅助数据流。接收器 (531) 可将已编码视频序列与其它数据分开。为了防止网络抖动, 缓冲存储器 (515) 可耦接在接收器 (531) 与熵解码器/解析器 (520) (此后称为“解析器 (520)”) 之间。在某些应用中, 缓冲存储器 (515) 是视频解码器 (510) 的一部分。在其它情况下, 所述缓冲存储器 (515) 可设置在视频解码器 (510) 外部 (未标示)。而在其它情况下, 视频解码器 (510) 的外部设置缓冲存储器 (未标示) 以例如防止网络抖动, 且在视频解码器 (510) 的内部可配置另一缓冲存储器 (515) 以例如处理播出定时。而当接收器 (531) 从具有足够带宽和可控性的存储/转发装置或从等时同步网络接收数据时, 也可能不需要配置缓冲存储器 (515), 或可以将所述缓冲存储器做得较小。当然, 为了在互联网等业务分组网络上使用, 也可能需要缓冲存储器 (515), 所述缓冲存储器可相对较大且可具有自适应性大小, 且可至少部分地实施于操作系统或视频解码器 (510) 外部的类似元件 (未标示) 中。

[0085] 视频解码器 (510) 可包括解析器 (520) 以根据已编码视频序列重建符号 (521)。这些符号的类别包括用于管理视频解码器 (510) 的操作的信息, 以及用以控制显示装置 (512) (例如, 显示屏) 等显示装置的潜在信息, 所述显示装置不是电子装置 (530) 的组成部分, 但可耦接到电子装置 (530), 如图5中所示。用于显示装置的控制信息可以是辅助增强信息 (Supplemental Enhancement Information, SEI 消息) 或视频可用性信息 (Video Usability Information, VUI) 的参数集片段 (未标示)。解析器 (520) 可对接收到的已编码视频序列进行解析/熵解码。已编码视频序列的编码可根据视频编码技术或标准进行, 且可遵循各种原理, 包括可变长度编码、霍夫曼编码 (Huffman coding)、具有或不具有上下文灵敏度的算术编码等等。解析器 (520) 可基于对应于群组的至少一个参数, 从已编码视频序列提取用于视频解码器中的像素的子群中的至少一个子群的子群参数集。子群可包括图片群组 (Group of Pictures, GOP)、图片、图块、切片、宏块、编码单元 (Coding Unit, CU)、块、变换单元 (Transform Unit, TU)、预测单元 (Prediction Unit, PU) 等等。解析器 (520) 还可从已编码视频序列提取信息, 例如变换系数、量化器参数值、运动矢量等等。

[0086] 解析器 (520) 可对从缓冲存储器 (515) 接收的视频序列执行熵解码/解析操作, 从而创建符号 (521)。

[0087] 取决于已编码视频图片或一部分已编码视频图片 (例如: 帧间图片和帧内图片、帧间块和帧内块) 的类型以及其它因素, 符号 (521) 的重建可涉及多个不同单元。涉及哪些单元以及涉及方式可由解析器 (520) 从已编码视频序列解析的子群控制信息控制。为了简洁起见, 未描述解析器 (520) 与下文的多个单元之间的此类子群控制信息流。

[0088] 除已经提及的功能块以外, 视频解码器 (510) 可在概念上细分成如下文所描述的数个功能单元。在商业约束下运行的实际实施例中, 这些单元中的许多单元彼此紧密交互并且可以彼此集成。然而, 出于描述所公开主题的目的, 概念上细分成下文的功能单元是适当的。

[0089] 第一单元是缩放器/逆变换单元(551)。缩放器/逆变换单元(551)从解析器(520)接收作为符号(521)的量化变换系数以及控制信息,包括使用哪种变换方式、块大小、量化因子、量化缩放矩阵等。缩放器/逆变换单元(551)可输出包括样本值的块,所述样本值可输入到聚合器(555)中。

[0090] 在一些情况下,缩放器/逆变换单元(551)的输出样本可属于帧内编码块;即:不使用来自先前重建的图像的预测性信息,但可使用来自当前图像的先前重建部分的预测性信息的块。此类预测性信息可由帧内图片预测单元(552)提供。在一些情况下,帧内图片预测单元(552)采用从当前图片缓冲器(558)提取的周围已重建的信息生成大小和形状与正在重建的块相同的块。举例来说,当前图片缓冲器(558)缓冲部分重建的当前图片和/或完全重建的当前图片。在一些情况下,聚合器(555)基于每个样本,将帧内预测单元(552)生成的预测信息添加到由缩放器/逆变换单元(551)提供的输出样本信息中。

[0091] 在其它情况下,缩放器/逆变换单元(551)的输出样本可属于帧间编码和潜在运动补偿块。在此情况下,运动补偿预测单元(553)可访问参考图片存储器(557)以提取用于预测的样本。在根据符号(521)对提取的样本进行运动补偿之后,这些样本可由聚合器(555)添加到缩放器/逆变换单元(551)的输出(在这种情况下被称作残差样本或残差信号),从而生成输出样本信息。运动补偿预测单元(553)从参考图片存储器(557)内的地址获取预测样本可受到运动矢量控制,且所述运动矢量以所述符号(521)的形式而供运动补偿预测单元(553)使用,所述符号(521)例如是包括X、Y和参考图片分量。运动补偿还可包括在使用子样本精确运动矢量时,从参考图片存储器(557)提取的样本值的内插、运动矢量预测机制等等。

[0092] 聚合器(555)的输出样本可在环路滤波器单元(556)中被各种环路滤波技术采用。视频压缩技术可包括环路内滤波器技术,所述环路内滤波器技术受控于包括在已编码视频序列(也称作已编码视频比特流)中的参数,且所述参数作为来自解析器(520)的符号(521)可用于环路滤波器单元(556)。然而,在其他实施例中,视频压缩技术还可响应于在解码已编码图片或已编码视频序列的先前(按解码次序)部分期间获得的元信息,以及响应于先前重建且经过环路滤波的样本值。

[0093] 环路滤波器单元(556)的输出可以是样本流,所述样本流可输出到显示装置(512)以及存储在参考图片存储器(557),以用于后续的帧间图片预测。

[0094] 一旦完全重建,某些已编码图片就可用作参考图片以用于将来预测。举例来说,一旦对应于当前图片的已编码图片被完全重建,且已编码图片(通过例如解析器(520))被识别为参考图片,则当前图片缓冲器(558)可变为参考图片存储器(557)的一部分,且可在开始重建后续已编码图片之前重新分配新的当前图片缓冲器。

[0095] 视频解码器(510)可根据例如ITU-T H.265标准中的预定视频压缩技术执行解码操作。在已编码视频序列遵循视频压缩技术或标准的语法以及视频压缩技术或标准中记录的配置文件的意义上,已编码视频序列可符合所使用的视频压缩技术或标准指定的语法。具体地说,配置文件可从视频压缩技术或标准中可用的所有工具中选择某些工具作为在所述配置文件下可供使用的仅有工具。对于合规性,还要求已编码视频序列的复杂度处于视频压缩技术或标准的层级所限定的范围内。在一些情况下,层级限制最大图片大小、最大帧率、最大重建取样率(以例如每秒兆(mega)个样本为单位进行测量)、最大参考图片大小等。

在一些情况下,由层级设定的限制可通过假想参考解码器

(HypotheticalReferenceDecoder,HRD)规范和在已编码视频序列中用信号表示的HRD缓冲器管理的元数据来进一步限定。

[0096] 在实施例中,接收器(531)可连同已编码视频一起接收附加(冗余)数据。所述附加数据可以是已编码视频序列的一部分。所述附加数据可由视频解码器(510)用以对数据进行适当解码和/或较准确地重建原始视频数据。附加数据可呈例如时间、空间或信噪比(signalnoiseratio,SNR)增强层、冗余切片、冗余图片、前向纠错码等形式。

[0097] 图6示出了根据本公开实施例的视频编码器(603)的框图。视频编码器(603)设置于电子装置(620)中。电子装置(620)包括传输器(640)(例如传输电路)。视频编码器(603)可用于代替图4实施例中的视频编码器(403)。

[0098] 视频编码器(603)可从视频源(601)(并非图6实施例中的电子装置(620)的一部分)接收视频样本,所述视频源可采集将由视频编码器(603)编码的视频图像。在另一实施例中,视频源(601)是电子装置(620)的一部分。

[0099] 视频源(601)可提供将由视频编码器(603)编码的呈数字视频样本流形式的源视频序列,所述数字视频样本流可具有任何合适位深度(例如:8位、10位、12位……)、任何色彩空间(例如BT.601YCrCb、RGB……)和任何合适取样结构(例如YCrCb4:2:0、YCrCb4:4:4)。在媒体服务系统中,视频源(601)可以是存储先前已准备的视频的存储装置。在视频会议系统中,视频源(601)可以是采集本地图像信息作为视频序列的相机。可将视频数据提供为多个单独的图片,当按顺序观看时,这些图片被赋予运动。图片自身可构建为空间像素阵列,其中取决于所用的取样结构、色彩空间等,每个像素可包括一个或多个样本。所属领域的技术人员可以很容易理解像素与样本之间的关系。下文侧重于描述样本。

[0100] 根据实施例,视频编码器(603)可实时或在由应用所要求的任何其它时间约束下,将源视频序列的图片编码且压缩成已编码视频序列(643)。施行适当的编码速度是控制器(650)的一个功能。在一些实施例中,控制器(650)控制如下文所描述的其它功能单元且在功能上耦接到这些单元。为了简洁起见,图中未标示耦接。由控制器(650)设置的参数可包括速率控制相关参数(图片跳过、量化器、率失真优化技术的 λ 值等)、图片大小、图片群组(groupofpictures,GOP)布局,最大运动矢量搜索范围等。控制器(650)可用于具有其它合适的功能,这些功能涉及针对某一系统设计优化的视频编码器(603)。

[0101] 在一些实施例中,视频编码器(603)在编码环路中进行操作。作为简单的描述,在实施例中,编码环路可包括源编码器(630)(例如,负责基于待编码的输入图片和参考图片创建符号,例如符号流)和嵌入于视频编码器(603)中的(本地)解码器(633)。解码器(633)以类似于(远程)解码器创建样本数据的方式重建符号以创建样本数据(因为在公开的主题中所考虑的视频压缩技术中,符号与已编码视频比特流之间的任何压缩是无损的)。将重建的样本流(样本数据)输入到参考图片存储器(634)。由于符号流的解码产生与解码器位置(本地或远程)无关的位精确结果,因此参考图片存储器(634)中的内容在本地编码器与远程编码器之间也是按比特位精确对应的。换句话说,编码器的预测部分“看到”的参考图片样本与解码器将在解码期间使用预测时所“看到”的样本值完全相同。这种参考图片同步性基本原理(以及在例如因信道误差而无法维持同步性的情况下产生的漂移)也用于一些相关技术。

[0102] “本地”解码器 (633) 的操作可与例如已在上文结合图5详细描述视频解码器 (510) 的“远程”解码器相同。然而, 另外简要参考图5, 当符号可用且熵编码器 (645) 和解析器 (520) 能够无损地将符号编码/解码为已编码视频序列时, 包括缓冲存储器 (515) 和解析器 (520) 在内的视频解码器 (510) 的熵解码部分, 可能无法完全在本地解码器 (633) 中实施。

[0103] 此时可以观察到, 除存在于解码器中的解析/熵解码之外的任何解码器技术, 也必定以基本上相同的功能形式存在于对应的编码器中。出于此原因, 所公开的主题侧重于解码器操作。可简化编码器技术的描述, 因为编码器技术与全面地描述的解码器技术互逆。仅在某些区域中需要更详细的描述, 并且在下文提供。

[0104] 在操作期间, 在一些实施例, 源编码器 (630) 可执行运动补偿预测编码。参考来自视频序列中被指定为“参考图片”的一个或多个先前已编码图片, 所述运动补偿预测编码对输入图片进行预测性编码。以此方式, 编码引擎 (632) 对输入图片的像素块与参考图片的像素块之间的差异进行编码, 所述参考图片可被选作所述输入图片的预测参考。

[0105] 本地视频解码器 (633) 可基于源编码器 (630) 创建的符号, 对可指定为参考图片的已编码视频数据进行解码。编码引擎 (632) 的操作可为有损过程。当已编码视频数据可在视频解码器 (图6中未示) 处被解码时, 重建的视频序列通常可以是带有一些误差的源视频序列的副本。本地视频解码器 (633) 复制解码过程, 所述解码过程可由视频解码器对参考图片执行, 且可使重建的参考图片存储在参考图片高速缓存 (634) 中。以此方式, 视频编码器 (603) 可在本地存储重建的参考图片的副本, 所述副本与将由远端视频解码器获得的重建参考图片具有共同内容 (不存在传输误差)。

[0106] 预测器 (635) 可针对编码引擎 (632) 执行预测搜索。即, 对于将要编码的新图片, 预测器 (635) 可在参考图片存储器 (634) 中搜索可作为所述新图片的适当预测参考的样本数据 (作为候选参考像素块) 或某些元数据, 例如参考图片运动矢量、块形状等。预测器 (635) 可基于样本块逐像素块操作, 以找到合适的预测参考。在一些情况下, 根据预测器 (635) 获得的搜索结果, 可确定输入图片可具有从参考图片存储器 (634) 中存储的多个参考图片取得的预测参考。

[0107] 控制器 (650) 可管理源编码器 (630) 的编码操作, 包括例如设置用于对视频数据进行编码的参数和子群参数。

[0108] 可在熵编码器 (645) 中对所有上述功能单元的输出进行熵编码。熵编码器 (645) 根据诸如霍夫曼编码、可变长度编码、算术编码等技术对各种功能单元生成的符号进行无损压缩, 从而将所述符号转换成已编码视频序列。

[0109] 传输器 (640) 可缓冲由熵编码器 (645) 创建的已编码视频序列, 从而为通过通信信道 (660) 进行传输做准备, 所述通信信道可以是通向将存储已编码的视频数据的存储装置的硬件/软件链路。传输器 (640) 可将来自视频编码器 (603) 的已编码视频数据与要传输的其它数据合并, 所述其它数据例如是已编码音频数据和/或辅助数据流 (未示出来源)。

[0110] 控制器 (650) 可管理视频编码器 (603) 的操作。在编码期间, 控制器 (650) 可以为每个已编码图片分配某一已编码图片类型, 但这可能影响可应用于相应的图片的编码技术。例如, 通常可将图片分配为以下任一种图片类型:

[0111] 帧内图片 (I图片), 其可以是不将序列中的任何其它图片用作预测源就可被编码和解码的图片。一些视频编解码器容许不同类型的帧内图片, 包括例如独立解码器刷新

(IndependentDecoderRefresh, “IDR”) 图片。所属领域的技术人员了解I图片的变体及其相应的应用和特征。

[0112] 预测性图片(P图片),其可以是可使用帧内预测或帧间预测进行编码和解码的图片,所述帧内预测或帧间预测使用至多一个运动矢量和参考索引来预测每个块的样本值。

[0113] 双向预测性图片(B图片),其可以是可使用帧内预测或帧间预测进行编码和解码的图片,所述帧内预测或帧间预测使用至多两个运动矢量和参考索引来预测每个块的样本值。类似地,多个预测性图片可使用多于两个参考图片和相关联元数据以用于重建单个块。

[0114] 源图片通常可在空间上细分成多个样本块(例如, 4×4 、 8×8 、 4×8 或 16×16 个样本的块),且逐块进行编码。这些块可参考其它(已编码)块进行预测编码,根据应用于块的相应图片的编码分配来确定所述其它块。举例来说,I图片的块可进行非预测编码,或所述块可参考同一图片的已经编码的块来进行预测编码(空间预测或帧内预测)。P图片的像素块可参考一个先前编码的参考图片通过空间预测或通过时域预测进行预测编码。B图片的块可参考一个或两个先前编码的参考图片通过空间预测或通过时域预测进行预测编码。

[0115] 视频编码器(603)可根据例如ITU-T H.265建议书的预定视频编码技术或标准执行编码操作。在操作中,视频编码器(603)可执行各种压缩操作,包括利用输入视频序列中的时间和空间冗余的预测编码操作。因此,已编码视频数据可符合所用视频编码技术或标准指定的语法。

[0116] 在实施例中,传输器(640)可在传输已编码的视频时传输附加数据。源编码器(630)可将此类数据作为已编码视频序列的一部分。附加数据可包括时间/空间/SNR增强层、冗余图片和切片等其它形式的冗余数据、补充增强信息(SEI)消息、视觉可用性信息(VUI)参数集片段等。

[0117] 采集到的视频可作为呈时间序列的多个源图片(视频图片)。帧内图片预测(常常简化为帧内预测)利用给定图片中的空间相关性,而帧间图片预测则利用图片之间的(时间或其它)相关性。在实施例中,将正在编码/解码的特定图片分割成块,正在编码/解码的特定图片被称作当前图片。在当前图片中的块类似于视频中先前已编码且仍被缓冲的参考图片中的参考块时,可通过称作运动矢量的矢量对当前图片中的块进行编码。所述运动矢量指向参考图片中的参考块,且在使用多个参考图片的情况下,所述运动矢量可具有识别参考图片的第三维度。

[0118] 在一些实施例中,双向预测技术可用于帧间图片预测中。根据双向预测技术,使用两个参考图片,例如按解码次序都在视频中的当前图片之前(但按显示次序可能分别是过去和将来)第一参考图片和第二参考图片。可通过指向第一参考图片中的第一参考块的第一运动矢量和指向第二参考图片中的第二参考块的第二运动矢量对当前图片中的块进行编码。具体来说,可通过第一参考块和第二参考块的组合来预测所述块。

[0119] 此外,合并模式技术可用于帧间图片预测中以改善编码效率。

[0120] 根据本公开的一些实施例,帧间图片预测和帧内图片预测等预测的执行以块为单位。举例来说,根据HEVC标准,将视频图片序列中的图片分割成编码树单元(codingtreeunit,CTU)以用于压缩,图片中的CTU具有相同大小,例如 64×64 像素、 32×32 像素或 16×16 像素。一般来说,CTU包括三个编码树块(codingtreeblock,CTB),所述三个编码树块是一个亮度CTB和两个色度CTB。更进一步的,还可将每个CTU以四叉树拆分为一个或

多个编码单元(codingunit, CU)。举例来说,可将 64×64 像素的CTU拆分为一个 64×64 像素的CU,或4个 32×32 像素的CU,或16个 16×16 像素的CU。在实施例中,分析每个CU以确定用于CU的预测类型,例如帧间预测类型或帧内预测类型。此外,取决于时间和/或空间可预测性,将CU拆分为一个或多个预测单元(predictionunit, PU)。通常,每个PU包括亮度预测块(predictionblock, PB)和两个色度PB。在实施例中,编码(编码/解码)中的预测操作以预测块为单位来执行。以亮度预测块作为预测块为例,预测块包括像素值(例如,亮度值)的矩阵,例如 8×8 像素、 16×16 像素、 8×16 像素、 16×8 像素等等。

[0121] 图7示出了根据本公开另一实施例的视频编码器(703)的示意图。视频编码器(703)用于接收视频图片序列中的当前视频图片内的样本值的处理块(例如预测块),且将所述处理块编码到作为已编码视频序列的一部分的已编码图片中。在本实施例中,视频编码器(703)用于代替图4实施例中的视频编码器(403)。

[0122] 在HEVC实施例中,视频编码器(703)接收用于处理块的样本值的矩阵,所述处理块为例如 8×8 样本的预测块等。视频编码器(703)使用例如率失真(rate-distortion)优化来确定是否使用帧内模式、帧间模式或双向预测模式来编码所述处理块。当在帧内模式中编码处理块时,视频编码器(703)可使用帧内预测技术以将处理块编码到已编码图片中;且当在帧间模式或双向预测模式中编码处理块时,视频编码器(703)可分别使用帧间预测或双向预测技术将处理块编码到已编码图片中。在某些视频编码技术中,合并模式可以是帧间图片预测子模式,其中,在不借助预测值外部的已编码运动矢量分量的情况下,从一个或多个运动矢量预测值导出运动矢量。在某些其它视频编码技术中,可存在适用于主题块的运动矢量分量。在实施例中,视频编码器(703)包括其它组件,例如用于确定处理块模式的模式决策模块(未示出)。

[0123] 在图7的实施例中,视频编码器(703)包括如图7所示的耦接到一起的帧间编码器(730)、帧内编码器(722)、残差计算器(723)、开关(726)、残差编码器(724)、通用控制器(721)和熵编码器(725)。

[0124] 帧间编码器(730)用于接收当前块(例如处理块)的样本、比较所述块与参考图片中的一个或多个参考块(例如先前图片和后来图片中的块)、生成帧间预测信息(例如根据帧间编码技术的冗余信息描述、运动矢量、合并模式信息)、以及基于帧间预测信息使用任何合适的技术计算帧间预测结果(例如已预测块)。在一些实施例中,参考图片是基于已编码的视频信息解码的已解码参考图片。

[0125] 帧内编码器(722)用于接收当前块(例如处理块)的样本、在一些情况下比较所述块与同一图片中已编码的块、在变换之后生成量化系数、以及在一些情况下还(例如根据一个或多个帧内编码技术的帧内预测方向信息)生成帧内预测信息。在实施例中,帧内编码器(722)还基于帧内预测信息和同一图片中的参考块计算帧内预测结果(例如已预测块)。

[0126] 通用控制器(721)用于确定通用控制数据,且基于所述通用控制数据控制视频编码器(703)的其它组件。在实施例中,通用控制器(721)确定块的模式,且基于所述模式将控制信号提供到开关(726)。举例来说,当所述模式是帧内模式时,通用控制器(721)控制开关(726)以选择供残差计算器(723)使用的帧内模式结果,且控制熵编码器(725)以选择帧内预测信息且将所述帧内预测信息添加在比特流中;以及当所述模式是帧间模式时,通用控制器(721)控制开关(726)以选择供残差计算器(723)使用的帧间预测结果,且控制熵编码

器(725)以选择帧间预测信息且将所述帧间预测信息添加在比特流中。

[0127] 残差计算器(723)用于计算所接收的块与选自帧内编码器(722)或帧间编码器(730)的预测结果之间的差(残差数据)。残差编码器(724)用于基于残差数据操作,以对残差数据进行编码以生成变换系数。在实施例,残差编码器(724)用于将残差数据从空间域转换到频域,且生成变换系数。变换系数接着经由量化处理以获得量化的变换系数。在各种实施例中,视频编码器(703)还包括残差解码器(728)。残差解码器(728)用于执行逆变换,且生成已解码残差数据。已解码残差数据可适当地由帧内编码器(722)和帧间编码器(730)使用。举例来说,帧间编码器(730)可基于已解码残差数据和帧间预测信息生成已解码块,且帧内编码器(722)可基于已解码残差数据和帧内预测信息生成已解码块。适当处理已解码块以生成已解码图片,且在一些实施例中,所述已解码图片可在存储器电路(未示出)中缓冲并用作参考图片。

[0128] 熵编码器(725)用于将比特流格式化以产生已编码的块。熵编码器(725)根据HEVC标准等合适标准产生各种信息。在实施例,熵编码器(725)用于获得通用控制数据、所选预测信息(例如帧内预测信息或帧间预测信息)、残差信息和比特流中的其它合适的信息。应注意,根据所公开的主题,当在帧间模式或双向预测模式的合并子模式中对块进行编码时,不存在残差信息。

[0129] 图8示出了根据本公开另一实施例的视频解码器(810)的示意图。视频解码器(810)用于接收作为已编码视频序列的一部分的已编码图片,且对所述已编码图片进行解码以生成重建的图片。在实施例,视频解码器(810)用于代替图4实施例中的视频解码器(410)。

[0130] 在图8实施例中,视频解码器(810)包括如图8中所示耦接到一起的熵解码器(871)、帧间解码器(880)、残差解码器(873)、重建模块(874)和帧内解码器(872)。

[0131] 熵解码器(871)可用于根据已编码图片来重建某些符号,这些符号表示构成所述已编码图片的语法元素。此类符号可包括例如用于对所述块进行编码的模式(例如帧内模式、帧间模式、双向预测模式、后两者的合并子模式或另一子模式)、可分别识别供帧内解码器(872)或帧间解码器(880)用以进行预测的某些样本或元数据的预测信息(例如帧内预测信息或帧间预测信息)、呈例如量化的变换系数形式的残差信息等等。在实施例,当预测模式是帧间或双向预测模式时,将帧间预测信息提供到帧间解码器(880);以及当预测类型是帧内预测类型时,将帧内预测信息提供到帧内解码器(872)。残差信息可经由逆量化并提供到残差解码器(873)。

[0132] 帧间解码器(880)用于接收帧间预测信息,且基于所述帧间预测信息生成帧间预测结果。

[0133] 帧内解码器(872)用于接收帧内预测信息,且基于所述帧内预测信息生成预测结果。

[0134] 残差解码器(873)用于执行逆量化以提取解量化的变换系数,且处理所述解量化的变换系数,以将残差从频域转换到空间域。残差解码器(873)还可能某些控制信息(用以获得量化器参数QP),且所述信息可由熵解码器(871)提供(未标示数据路径,因为这仅仅是低量控制信息)。

[0135] 重建模块(874)用于在空间域中组合由残差解码器(873)输出的残差与预测结果

(可由帧间预测模块或帧内预测模块输出)以形成重建的块,所述重建的块可以是重建的图片的一部分,所述重建的图片继而可以是重建的视频的一部分。应注意,可执行解块操作等其它合适的操作来改善视觉质量。

[0136] 应注意,可使用任何合适的技术来实施视频编码器(403)、视频编码器(603)和视频编码器(703)以及视频解码器(410)、视频解码器(510)和视频解码器(810)。在实施例中,可使用一个或多个集成电路来实施视频编码器(403)、视频编码器(603)和视频编码器(703)以及视频解码器(410)、视频解码器(510)和视频解码器(810)。在另一实施例中,可使用执行软件指令的一个或多个处理器来实施视频编码器(403)、视频编码器(603)和视频编码器(703)以及视频解码器(410)、视频解码器(510)和视频解码器(810)。

[0137] 本公开的各方面包括在已编码视频流中使用约束标志来控制编解码工具(多个)以及功能。

[0138] 视频编码器和解码器可以对给定的图片大小进行操作,该给定的图片大小针对已编码视频序列(CVS,coded video sequence)、图片组(GOP,Group of Pictures)或相似的多图片时间帧而定义且保持不变。在示例中,例如在MPEG-2中,已知系统设计会根据诸如场景的热闹状况等因素来改变水平分辨率(并因此改变图片大小),但是仅在I图片处改变水平分辨率,因此通常用于GOP。例如,根据ITU-T Rec.H.263附录P,已知在CVS内使用不同分辨率对参考图片进行重采样。然而,图片大小并未改变,仅参考图片被重采样,这可能会导致仅使用图片画布的一部分(例如,在下采样中),或仅捕获场景的一部分(例如,在上采样中)。在一些示例中,诸如在H.263附录Q中,允许在每个维度(例如,向上或向下)对单个宏块进行二倍重采样。图片大小可以保持相同。宏块的大小可以是固定的,例如在H.263中,因此不需要用信号通知。

[0139] 可以改变预测图片的图片大小。在示例中,例如在VP9中,允许参考图片重采样以及改变整个图片的分辨率。在示例中,(包括,例如,Hendry等人,“On Adaptive Resolution Change (ARC) for VVC(关于VVC的自适应分辨率改变(ARC))”,联合视频小组文件JVET-M0135-v1,2019年1月9日-19日,其全部内容被并入本文)允许将整个参考图片重采样至不同的分辨率(例如,更高的分辨率或更低的分辨率)。可以将不同的候选分辨率编码在序列参数集(SPS,sequence parameter set)中,并且可以由图片参数集(PPS,picture parameter set)中的每个图片语法元素来引用。

[0140] 在本公开中公开了用于在视频比特流中用信号通知自适应图片大小的技术。

[0141] 当图片被编码成包括具有不同质量的一个或多个层的比特流时,该比特流可以具有用于指定可以在解码器侧输出哪个(些)层(或层的集合)的语法元素。要输出的层的集合可以被定义为输出层集。在支持多层及可缩放性的示例性视频编解码器中,可以在视频参数集(VPS,video parameter set)中用信号通知一个或多个输出层集。可以在VPS、SPS、PPS、SEI消息、与解码器能力信息(DCI,Decoder Capability Information)相同的DPS等中用信号通知语法元素,该语法元素用于为整个比特流或者一个或多个输出层集指定档位层级等级(PTL,profile tier level)。在PTL信息中,可以存在通用约束信息,该通用约束信息可以指定对编解码工具或功能的约束。期望有效地表示并用信号通知针对各种编解码工具和功能的约束信息。

[0142] 子图片可以指,例如,在语义上被分组并且可以以改变后的分辨率被独立编解码

的样本、块、宏块、编码单元或类似实体的矩形布置。一个或多个子图片可以形成图片。一个或多个已编码子图片可以形成已编码图片。可以将一个或多个子图片组合成图片,并且可以从图片中提取一个或多个子图片。在一些示例中,可以在压缩域中组合一个或多个已编码子图片,而无需将样本级别转码为已编码图片。在一些示例中,可以从压缩域中的已编码图片中提取一个或多个已编码子图片。

[0143] 自适应分辨率改变(ARC)可以指,允许通过例如参考图片重采样来改变CVS中的图片或子图片的分辨率的机制。ARC参数可以指,用于执行自适应分辨率改变的控制信息。ARC参数可以包括滤波器参数、缩放因子、输出和/或参考图片的分辨率、各种控制标志等。

[0144] 以上描述可应用于对单个且在语义上独立的已编码视频图片进行编码和解码。在描述具有独立ARC参数的多个子图片的编码/解码及其隐含的额外复杂性之前,对用信号通知ARC参数进行描述。

[0145] 图9示出了根据本公开的实施例的用信号通知ARC参数的示例(例如,选项)。编解码效率、复杂性以及架构可能因示例而不同。视频编解码标准或技术可以选择示例中的一个或多个或者其他变体,以用信号通知ARC参数。这些示例可以不是互相排斥的,并且可以基于应用需求、标准技术、编码器的选择等而互换。

[0146] ARC参数的类别可以包括:

[0147] 上采样因子和/或下采样因子,在X维度和Y维度中分离或组合,其中可以对可指向包括上采样因子和/或下采样因子的表的一个或多个短语法元素进行编解码;

[0148] 上采样因子和/或下采样因子,增加了时间维度,指示针对给定数量的图片的恒定速度放大和/或缩小,其中可以对可指向包括增加了时间维度的上采样因子和/或下采样因子的表的一个或多个短语法元素进行编解码;

[0149] 分辨率,在X维度或Y维度中以输入图片、输出图片、参考图片、已编码图片的样本、块、宏块、CU、或任何其他适当的粒度为单位组合或分离。如果存在多于一个的分辨率(例如,一个分辨率用于输入图片,另一个分辨率用于参考图片),则可以从一组值中推断出另一组值,这可以例如通过使用标志来进行门控,如下所述;

[0150] 翘曲坐标,与H.263附录P中使用的坐标相似,具有如上所述的适当粒度。H.263附录P定义了对翘曲坐标进行编解码的有效方式。可以设计出其他的有效方式。例如,附录P的翘曲坐标的可变长度可逆霍夫曼式编解码可以用适当长度的二进制编解码来代替,其中二进制码字的长度可以从最大图片大小导出,该最大图片大小乘以一个因子并偏移一个值,以允许在最大图片大小的边界之外翘曲;

[0151] 上采样滤波器参数和/或下采样滤波器参数。在示例中,仅单个滤波器用于上采样和/或下采样。在示例中,允许在滤波器设计中具有更大的灵活性是有利的。在示例中,用信号通知滤波器参数。可以通过使用可能的滤波器设计列表中的索引来选择滤波器参数。可以完全指定滤波器(例如,通过指定滤波器系数的列表,使用适当的熵编解码技术),可以通过上采样比率或下采样比率来隐式地选择滤波器,该上采样比率或下采样比率是根据以上描述的机制中的任一种和/或类似物来用信号通知的。

[0152] 该描述可被应用于通过码字指示的上采样因子或下采样因子的有限集合的编解码(在X维度和Y维度中使用相同的因子)。有利地,码字可以是可变长度编解码的,例如,针对视频编解码规范(例如,H.264和H.265)中的某些语法元素使用Ext-Golomb码。例如,可以

基于表1将值适当映射到上采样因子或下采样因子。

[0153] 表1:对上采样因子或下采样因子、码字和Ext-Golomb码进行映射

码字	Ext-Golomb码	原始/目标分辨率
0	1	1/1
1	010	1/1.5 (放大50%)
2	011	1.5/1 (缩小50%)
3	00100	1/2 (放大100%)
4	00101	2/1 (缩小100%)

[0155] 可以根据视频压缩技术或标准中可用的放大和缩小机制的应用及能力来设计类似的映射。表1可以适当地扩展到其他值。值可以由不同于Ext-Golomb码的熵编解码机制来表示,例如,通过使用二进制编解码。在示例中,当重采样因子被视频处理引擎(例如,编码器和解码器)之外的例如媒体感知网络元素(MANE, media-aware network elements)关注时,不同于Ext-Golomb码的熵编解码机制可能会具有某些优点。当不需要改变分辨率时(例如,表1中的原始/目标分辨率是1),可以选择短Ext-Golomb码(例如,表1中仅示出单个比特),这可以具有编解码效率优势,例如,优于针对最常见情况使用二进制码。

[0156] 表1中的多个条目以及对应的语义可以是完全或部分可配置的。例如,表1的基本概要在诸如SPS或DPS等高级参数集中传达。可替代地或另外地,可以在视频编解码技术或标准中定义类似于表1的一个或多个表,并且可以通过例如SPS或DPS来进行选择。

[0157] 以上描述了可以在视频编解码技术或标准语法中包括上采样因子或下采样因子(ARC信息)如何如上述那样被编解码。类似的考虑可以被应用于对上采样滤波器或下采样滤波器进行控制的一个或几个码字。当过滤器或其它数据结构需要相对大量的数据时,描述如下。

[0158] 参考图9,在示例中,例如在H.263附录P中,四个翘曲坐标的形式下的ARC信息(902)被包括在图片头(901)中,具体而言,被包括在H.263PLUSPTYPE(903)头扩展中。包括ARC信息(902)和H.263PLUSPTYPE(903)头扩展的图片头(901)可以被应用在,i)图片头可用时,以及ii)预期到ARC信息的频繁改变时。然而,使用H.263样式的信令时的开销可能会很高,并且缩放因子可能不适用于图片边界之间,因为图片头可以是瞬态的。

[0159] 参考图9,在诸如JVCET-M135-v1等示例中,位于PPS(904)中的ARC参考信息(905)(例如,索引)可以指向包括目标分辨率(例如,分辨率1-3)的表(或目标分辨率表)(906),其中表(906)位于SPS(907)内。通过在能力交换期间使用SPS作为互操作性协商点,从而可以证明将目标分辨率放置在SPS(907)中的表(906)中是合理的。通过参考适当的PPS(904),分辨率可以在由表(906)中的值(例如,分辨率1-3)设置的限制内从一个图片改变至另一个图片。

[0160] 返回参考图9,附加选项或示例可被用于在视频比特流中传达ARC信息。多个选项可以同时存在于同一视频编解码技术或标准中。

[0161] 在实施例中,诸如重采样因子(或缩放因子)等ARC信息(909)可以存在于诸如切片头、GOB头、图块头或图块组头等头中。例如,图9中示出了图块组头(908)。当用于ARC信息(909)的存储空间较小时,该实施例可以是合乎需要的,例如单个可变长度ue(v)或几个比特的固定长度码字,如上所示。

[0162] 直接在头(例如,图9中的图块组头(908)、切片头或图块头)中具备ARC信息(909)可以具有额外的优点,因为ARC信息(909)可以被应用于由例如对应的图块组(或切片、图块)表示的子图片,而不是整个图片。另外,在示例中,即使视频压缩技术或标准仅设想整个图片自适应分辨率改变(与例如基于图块组的自适应分辨率改变形成对比),从容错性的观点来看,将ARC信息(909)放入图块组头(908)中相对于将ARC信息(909)放入H.263样式图片头中,可以具有一定的优势。

[0163] 参考图9,ARC信息(912)可以存在于参数集(911)中,诸如PPS、头参数集、图块参数集、自适应参数集(APS)等。例如,图9中示出了APS(911)。有利地,参数集(911)的范围可以不大于图片,例如图块组。通过激活相关参数集(例如,APS(911)),ARC信息(例如,ARC信息(912))的使用可以是隐式的。例如,当视频编解码技术或标准仅考虑基于图片的ARC时,PPS或等效物可能是适当的。

[0164] 在一些实施例中,ARC参考信息(913)可以存在于图块组头(914)或如上所述的类似的数据结构(例如,图片头、切片头、图块头或GOP头)中。例如,图9中示出了图块组头(914)。ARC参考信息(913)可以指在具有超出单个图片的范围的参数集(916)中可用的ARC信息(915)的子集,例如SPS或DPS等。例如,图9中示出了SPS(916)。

[0165] 图10示出了根据本公开的实施例的ARC参数信令的示例。图10示出了在视频编解码标准中使用的语法图示例。在示例中,语法图的符号大致遵循C样式编程。黑体字的行可以表示比特流中存在的语法元素,没有黑体字的行可以表示控制流(多个)或者对变量的设置。

[0166] 参考图10,图块组头(1001),作为可应用于图片的一部分(例如,矩形部分)的头的示例性语法结构,可以有条件地包含,可变长度的、被Exp-Golomb编码的、语法元素`dec_pic_size_idx`(1002)(以黑体字描绘)。可以基于例如由标志(例如,`adaptive_pic_resolution_change_flag`)(1003)表示的自适应分辨率来对图块组头(1001)中语法元素(例如,`dec_pic_size_idx`(1002))的存在进行门控。标志(例如,`adaptive_pic_resolution_change_flag`)(1003)的值未以黑体字描绘,因此在语法图中出现该标志的位置处,该标志存在于比特流中。可以在比特流内部或外部在高级别语法结构(例如,如下所描述的SPS(1010))中用信号通知自适应分辨率是用于图片还是用于图片的一部分。

[0167] 参考图10,示出了SPS(1010)的摘录。所示的第一语法元素是标志(例如,`adaptive_pic_resolution_change_flag`)(1011)。当标志为真时,该标志可以指示对自适应分辨率的使用,这可能需要某些控制信息。在示例中,特定控制信息基于如SPS(1010)中的`if()`语句(1012)所示的标志(1011)的值以及图块组头(1001)而有条件地存在。

[0168] 当使用自适应分辨率时,如图10中的示例所示,可以对以样本为单位的输出分辨率(或输出图片的分辨率)(1013)进行编解码。输出分辨率(1013)可以指宽度分辨率(例如,`output_pic_width_in_luma_samples`)和高度分辨率(例如,`output_pic_height_in_luma_samples`)。宽度分辨率和高度分辨率可以定义输出分辨率(1013)。在视频编解码技术或标准中,可以定义对输出分辨率(1013)的一个或多个值的某些限制。例如,级别定义可以限制总输出样本的数量(例如,`output_pic_width_in_luma_samples`与`output_pic_height_in_luma_samples`的乘积)。视频编解码技术或标准,或者外部技术或标准(例如,系统标准)可以限制范围(例如,宽度分辨率和/或高度分辨率可以被2的幂整除)、纵横比(例如,宽度分

分辨率与高度分辨率的比率为4:3或16:9)等。在示例中,可以引入以上限制以便于硬件实现。

[0169] 在某些应用中,编码器可以指示解码器使用某个参考图片大小,而不是隐含地假定一个尺寸是输出图片大小。例如,语法元素(例如,reference_pic_size_present_flag)(1014)对参考图片尺寸(1015)的条件性存在进行门控。参考图片尺寸(1015)可以指宽度(例如,reference_pic_width_in_luma_samples)和高度(例如,reference_pic_height_in_luma_samples)。

[0170] 图10示出了可应用的解码图片宽度和高度的表。该表可以通过表指示(例如,语法元素num_dec_pic_size_in_luma_samples_minus1)(1016)来表达。“minus1”可以指语法元素(1016)的值的解释。例如,如果编码值为零,则存在一个表条目。如果编码值为五,则存在六个表条目。对于该表中的每一行,已解码图片的宽度和高度被包括在语法元素(1017)中。

[0171] 可以使用图块组头(1001)中的语法元素dec_pic_size_idx(1002)来对由语法元素(1017)表示的表条目进行索引,因此允许每一个图块组具有不同的解码大小和缩放因子。

[0172] 某些视频编解码技术或标准(例如,VP9)可以通过结合时间可缩放性以实施某些形式的参考图片重采样来支持空间可缩放性,因此可以启用空间可缩放性。在实施例中,使用ARC样式的技术将参考图片上采样至更高的分辨率,以形成空间增强层的基底。可以在高分辨率下使用正常预测机制(例如,用于来自参考图片的帧间预测的运动补偿预测)来细化被上采样的图片,用以例如添加细节。

[0173] 本公开可以在这类环境中使用。在一些示例中,网络抽象层(NAL,Network Abstraction Layer)单元头中的值(例如,Temporal ID字段)被用于指示时间层和空间层,优点是无需修改即可将现有的选定转发单元(SFU,Selected Forwarding Units)用于可缩放环境。基于NAL单元头的Temporal ID的值,可以针对所选转发的时间层来创建和优化SFU。为了实现这一点,可以在编码图片大小与如NAL单元头中的Temporal ID字段所指示的时间层之间执行映射。

[0174] 当图片被编码成包括具有不同质量的一个或多个层的比特流时,该比特流可以具有用于指定可以在解码器侧输出哪个(些)层的语法元素。要输出的层的集合可以被定义为输出层集。在支持多层及可缩放性的视频编解码器中,可以在一个或多个VPS中用信号通知一个或多个输出层集。可以在诸如VPS、DPS、DCI、SPS、PPS、APS、GOP、序列、头或SEI消息等高级别语法结构中用信号通知一个或多个语法元素,该一个或多个语法元素用于为整个比特流、CVS、每个输出层集等指定PTL信息。

[0175] 块级别编解码工具可以被用于对图片中的像素或样本进行解码以重建图片。块级别编解码工具可以包括在编码块的重建中使用的任何适当的编解码工具,诸如用于帧间预测的编解码工具(或帧间编解码工具)、用于帧内预测的编解码工具(多个)(或帧内编解码工具)、自适应环路滤波器(ALF)、去块滤波器(DBF)、熵编解码、变换等。

[0176] 高级别语法(HLS,High-level syntax)可以指定关于功能、系统接口、工具的图片级控制以及缓冲器控制等的信息。例如,HLS可以指定分区(例如,图块、切片、子图片)、缓冲器管理、随机存取(例如,IDR、干净随机存取(CRA,clean random access))、一个或多个参数集(例如,VPS、SPS、PPS、APS)、参考图片重采样(RPR,reference picture resampling)、可缩放性等。高级别语法可以高于块级别的。

[0177] 控制信息可以包括SPS级别工具控制信息、PPS级别工具控制信息、序列级别控制信息、比特流级别控制信息等。

[0178] 约束标志可以是HLS结构的一部分。

[0179] 在示例中,约束标志指示对工具的控制。在序列级别控制信息和比特流级别控制信息中的一个中提供约束标志。在示例中,如果某些工具被约束标志禁用,则在HLS中不存在控制信息,并且工具不被用于例如对与HLS相对应的范围中的块进行编解码。

[0180] 可以在档位语法结构中用信号通知约束标志。

[0181] 约束信息可以指示在对应的视频流或已编码视频比特流中是否存在或使用特定的工具、功能或语法元素。例如,约束标志可以指示在已编码视频比特流中是否存在或使用帧间编解码工具、帧内编解码工具、DBF、熵编解码、变换、分区(例如,图块、切片)、缓冲器管理、随机存取(例如,IDR)、参数集(例如,SPS、PPS)等。可以在参数集(例如,SPS、VPS或DCI)中用信号通知约束信息。可以在高级别语法结构(例如,SPS、VPS、DCI)中用信号通知约束标志。

[0182] 图11示出了根据本公开的实施例的PTL语法元素的集合的示例。PTL语法元素的集合可以包括general_profile_idc、general_tier_flag、general_level_idc、num_sub_profiles、general_sub_profile_idc、sublayer_level_present_flag、ptl_alignment_zero_bit、sublayer_level_idc和通用约束信息。

[0183] 通用约束信息可以包括关于视频源类型、编解码工具、功能等的约束信息。图12示出了根据本公开的实施例的通用约束信息的示例。如图12中所示,通用约束信息可以包括多个约束标志。参考图12,约束标志(例如,intra_only_constraint_flag) (1205) 等于1指示,sh_slice_type为I(即,切片是帧内切片)。约束标志(例如,intra_only_constraint_flag) (1205) 等于0,则不将约束(例如,sh_slice_type为I)施加于PTL信息的范围内的所有已编码图片,其中,其他信息(例如,profile_idc)可以允许非帧内切片。在示例中,约束标志(例如,no_alf_constraint_flag) (1206) 等于1可以指示,sps_alf_enabled_flag对于PTL信息的范围内的所有CVS都等于0,因此即使基于例如profile_idc而允许自适应环路滤波,也不使用自适应环路滤波。约束标志(例如,no_alf_constraint_flag) (1206) 等于0,则不会施加以上约束。

[0184] 可以在通用约束信息中用信号通知约束标志(例如,no_lossless_coding_tool_constraint_flag) (1201),如图12中所示。约束标志(例如,no_lossless_coding_tool_constraint_flag) (1201) 等于1可以指示,在包括约束标志(1201)的PTL信息的范围内不能使用与无损编解码相关的一个或多个编解码工具。约束标志(例如,no_lossless_coding_tool_constraint_flag) (1201) 等于0,则不会施加以上约束。

[0185] 可以在通用约束信息中用信号通知约束标志(例如,no_lossy_coding_tool_constraint_flag) (1202),如图12中所示。约束标志(例如,no_lossy_coding_tool_constraint_flag) (1202) 等于1可以指示,在包括约束标志(1202)的PTL信息的范围内不能使用与有损编解码相关的一个或多个编解码工具。约束标志(例如,no_lossy_coding_tool_constraint_flag) (1202) 等于0,则不会施加以上约束。

[0186] 在实施例,当约束标志(例如,no_lossy_coding_tool_constraint_flag) (1202) 等于1时,约束标志(例如,no_lossy_coding_tool_constraint_flag) (1201) 可以不

等于1。可替代地,当约束标志(例如,no_lossy_coding_tool_constraint_flag)(1201)等于1时,约束标志(例如,no_lossy_coding_tool_constraint_flag)(1202)可以不等于1。

[0187] 通用约束信息中的多个约束标志可以按特定顺序排序。该顺序可以基于例如在PTL的范围内不使用各个机制和/或工具的可能性来设置。该顺序可以被称为优先级顺序。可以在通用约束信息语法结构中从高优先级到低优先级呈现该顺序,其中,高优先级指示不使用工具(或机制)的可能性高,低优先级指示不使用工具(或机制)的可能性低。影响该顺序的额外因素可以包括,可能仅用于特定使用情况的工具(例如,用于子图片、可缩放性和/或隔行扫描支持的工具)、工具对编码器/解码器/实现复杂性的影响等。

[0188] 图13A至图13B示出了根据本公开的实施例的PTL信息(包括例如PTL语法元素的集合)的示例。可以在PTL信息中用信号通知用于指示约束标志(例如,num_available_constraint_flags)(1301)的数量的语法元素。例如,可以在通用约束信息括号外部用信号通知约束标志的数量,如图13A中所示。可替代地,可以在通用约束信息语法结构的开始处用信号通知用于指示约束标志的数量的语法元素。当语法元素(例如,num_available_constraint_flags)存在且语法元素(例如,num_available_constraint_flags)的值等于N时,前N个约束标志可以存在于通用约束信息语法结构中。此外,其他的约束标志可以不存在,并且可以被推断为等于特定值。N可以是非负整数。

[0189] 在实施例中,值N(例如,num_available_constraint_flags)在0至约束标志的最大数量(例如,MaxNumConstraintFlags)的范围内。约束标志的最大数量可以是任何正整数。约束标志的最大数量的值(例如,MaxNumConstraintFlags)可以被预定义为16、32、64、128等。当值N(例如,num_available_constraint_flags)等于0时,在通用约束信息语法结构中不存在约束标志。值N(例如,num_available_constraint_flags)的编码可以被选择为使得值N和约束标志的相应熵编码表示可以加起来为一个可被8整除的数字,用以确保字节对齐。

[0190] 根据本公开的各方面,可以将约束标志分类成一个或多个约束信息组。每个约束信息组可以包括一个或多个约束标志,并且可以具有对应的门控标志。对应的约束信息组的门控标志可以指示对应的约束信息组中是否可以存在一个或多个约束标志。在示例中,门控标志可以被称为约束组存在标志。通常,门控标志与对应的约束信息组相关联,并且与对应的约束信息组中的一个或多个约束标志相关联。在实施例中,门控标志用于对对应的约束信息组中的一个或多个约束标志是否存在于约束信息中(或在其中被用信号通知)进行门控。例如,如果对应的约束信息组的门控标志等于1,则与约束信息组相对应的一个或多个约束标志可以存在于例如约束信息中。如果对应的约束信息组的门控标志等于0,则与约束信息组相对应的一个或多个约束标志可能不存在于例如约束信息中。在示例中,如果所有的门控标志都等于0,则不存在约束标志。

[0191] 约束标志可以具有不同的副本。例如,DCI中的约束标志的范围可以是已编码视频比特流。VPS中的约束标志的范围可以是具有多个层的一个或多个CLVS。SPS中的约束标志的范围可以是单个CLVS。

[0192] 标志(例如,gate_flag_all)可以指示是否存在一个或多个门控标志。在示例中,标志为假指示,不存在一个或多个门控标志。在示例中,当不存在一个或多个门控标志时,约束信息中不存在约束标志。在示例中,标志为真指示,存在门控标志(多个)中的一个或多

个。

[0193] 根据本公开的各方面,可以获得针对在已编码视频比特流中用信号通知的约束信息中的至少一个约束信息组中的每个约束信息组的门控标志。每个门控标志可以指示,与相应门控标志相对应的至少一个约束信息组中的约束信息组是否存在于约束信息中。约束信息可用于已编码视频比特流中的编码块的子集。可以基于至少一个约束信息组中的约束信息组的门控标志来确定门控标志对应的约束信息组是否存在于约束信息中。门控标志对应的约束信息组可以包括针对编码块的子集的至少一个约束标志。可以基于门控标志对应的约束信息组是否存在于约束信息中来确定针对编码块的子集的预测信息。可以基于预测信息来重建编码块的子集。

[0194] 在实施例中,门控标志对应的约束信息组被确定为存在于约束信息中。可以基于门控标志对应的约束信息组中的至少一个约束标志来确定针对编码块的子集的预测信息。在实施例中,门控标志对应的约束信息组被确定为不存在于约束信息中。

[0195] 在实施例中,每个门控标志用于指示,与相应门控标志相对应的约束信息组不存在于约束信息中,并且在约束信息中不存在约束标志。

[0196] 在实施例中,约束信息存在于针对至少一个图片的高级别语法中,并且编码块的子集包括至少一个图片中的编码块。高级别语法可以是VPS、DPS、DCI、SPS、PPS、APS、GOP、序列等。在示例中,高级别语法是DCI或DPS,约束信息的范围包括已编码视频比特流,并因此包括已编码视频比特流中的编码块。在示例中,高级别语法是VPS,约束信息的范围包括具有多个层的一个或多个CLVS,并因此包括一个或多个CLVS中的编码块。在示例中,高级别语法是SPS,约束信息的范围包括单个CLVS,并因此包括单个CLVS中的编码块。

[0197] 在实施例中,可以存在于约束信息中(或在其中被用信号通知)的每个约束标志被分类成一个或多个约束信息组,并因此可由各自的门控标志来进行门控。在示例中,一个或多个约束信息组包括第一约束信息组和第二约束信息组。约束标志包括至少一个第一约束标志和至少一个第二约束标志。至少一个第一约束标志被分类到第一约束信息组中,并且至少一个第二约束标志被分类到第二约束信息组中。第一约束信息组的第一门控标志可以指示至少一个第一约束标志(或第一约束信息组)是否存在于约束信息中(例如,在其中被用信号通知)。第二约束信息组的第二门控标志可以指示至少一个第二约束标志(或第二约束信息组)是否存在于约束信息中(例如,在其中被用信号通知)。因此,在该示例中,约束信息不包括针对至少一个第一约束标志以及至少一个第二约束标志的一个或多个附加约束标志。

[0198] 在实施例中,约束标志的第一子集被分类成一个或多个约束信息组,并因此可由各自的门控标志来进行门控。此外,约束标志的第二子集在一个或多个约束信息组之外,因此不会由一个或多个门控标志来进行过门控。在示例中,一个或多个约束信息组包括第三约束信息组和第四约束信息组。约束标志的第一子集包括至少一个第三约束标志和至少一个第四约束标志。至少一个第三约束标志被分类到第三约束信息组中,并且至少一个第四约束标志被分类到第四约束信息组中。第三约束信息组的第三门控标志可以指示至少一个第三约束标志(或第三约束信息组)是否存在于约束信息中(例如,在其中被用信号通知)。第四约束信息组的第四门控标志可以指示至少一个第四约束标志(或第四约束信息组)是否存在于约束信息中(例如,在其中被用信号通知)。另外,约束信息可以包括约束标志的第

二子集。在示例中,第三门控标志和第四门控标志指示,第三约束信息组和第四约束信息组没有在约束信息中被用信号通知,因此约束信息包括约束标志的第二子集并且不包括至少一个第三约束标志和至少一个第四约束标志。在示例中,第三门控标志和第四门控标志指示,第三约束信息组和第四约束信息组在约束信息中被用信号通知,因此约束信息包括约束标志的第二子集、至少一个第三约束标志和至少一个第四约束标志。

[0199] 图14A至图14B示出了根据本公开的实施例的示例性通用约束信息语法结构(1400)。通用约束信息语法结构(1400)可以表示约束信息(例如,被称为通用约束信息)。通用约束信息(或通用约束信息语法结构(1400))可以包括一个或多个门控标志,诸如图14A中的门控标志general_frame_structure_constraint_group_flag(1401)、门控标志(例如,high_level_functionality_constraint_group_flag(1402)、门控标志(例如,scalability_constraint_group_flag(1403)、门控标志(例如,partitioning_constraint_group_flag(1404)、门控标志(例如,intra_coding_tool_constraint_group_flag(1405)、门控标志(例如,inter_coding_tool_constraint_group_flag(1406)、门控标志(例如,transfom_constraint_group_flag(1407)、门控标志(例如,inloop_filtering_constraint_group_flag(1408)。一个或多个门控标志(例如,门控标志(1401)-(1408))可以存在于通用约束信息语法结构(1400)的开始处,如图14A中所示。

[0200] 门控标志(例如,general_frame_structure_constraint_group_flag(1401)与约束信息组(1410)相关联,并且与约束信息组(1410)中的约束标志(1411)-(1414)相关联。门控标志(例如,general_frame_structure_constraint_group_flag(1401)等于1可以指定,约束信息组(1410)中的约束标志(1411)-(1414)可以存在。

[0201] 约束信息组(1410)(或约束标志(1411)-(1414))可以与输入源以及帧打包(例如,被打包的帧或被投影的帧)相关。参考图14A,约束标志(1411)-(1414)与general_non_packed_constraint_flag(1411)、general_frame_only_constraint_flag(1412)、general_non_projected_constraint_flag(1413)和general_one_picture_only_constraint_flag(1414)相对应。否则,门控标志(例如,general_frame_structure_constraint_group_flag(1401)等于0可以指定,约束信息组(1410)中的约束标志(1411)-(1414)可能不存在。

[0202] 参考图14B,门控标志(例如,high_level_functionality_constraint_group_flag(1402)等于1可以指定,约束信息组(1420)中的与高级别功能(例如,参考图片重采样)相关的约束标志可以存在。否则,门控标志(例如,high_level_functionality_constraint_group_flag(1402)等于0可以指定,约束信息组(1420)中的约束标志可能不存在。

[0203] 返回参考图14A,门控标志(例如,scalability_constraint_group_flag(1403)等于1可以指定,与可缩放性(例如,层间预测)相关的一个或多个约束标志可以存在。否则,与可缩放性相关的一个或多个约束标志可能不存在。

[0204] 门控标志(例如,partitioning_constraint_group_flag(1404)等于1可以指定,与高级别分区(例如,子图片或图块)相关的一个或多个约束标志可以存在。否则,与高级别分区相关的约束标志可能不存在。

[0205] 门控标志(例如,intra_coding_tool_constraint_group_flag(1405)等于1可以

指定,与帧内编解码(例如,帧内预测)相关的一个或多个约束标志可以存在。否则,与帧内编解码相关的一个或多个约束标志可能不存在。

[0206] 门控标志(例如,inter_coding_tool_constraint_group_flag) (1406) 等于1可以指定,与帧间编解码(例如,用于图片间预测的运动补偿)相关的一个或多个约束标志可以存在。否则,在与帧间编解码相关的约束标志可能不存。

[0207] 门控标志(例如,transfom_constraint_group_flag) (1407) 等于1可以指定,与变换编解码(例如,多重变换矩阵)相关的一个或多个约束标志可以存在。否则,与变换编解码相关的约束标志可能不存在。

[0208] 在实施例中,当所有门控标志(例如,图14A中的门控标志(1401) - (1408))都等于0时,约束信息语法结构(例如,通用约束信息语法结构(1400))中不存在约束标志。

[0209] 根据本公开的各方面,语法可以被设计为使得包括门控标志(例如,门控标志(1401) - (1408))、相关联的约束标志(例如,约束标志(1411) - (1412)以及约束信息组(1420)中的约束标志)、附加控制信息等的控制信息可以是字节对齐的,例如,标志的数量可以被8整除,用以保持字节对齐。在示例中,约束信息(例如,通用约束信息语法结构(1400))中的门控标志和约束标志的数量可以被8整除。可以使用字节对齐机制来实现控制信息的字节对齐。参考图14B,语法(例如,while循环) (1430) 可被用于字节对齐。

[0210] 在实施例中,诸如偏移量(例如,语法元素constraint_info_offset[])等偏移信息和诸如长度(例如,语法元素constraint_info_length[])等长度信息存在于约束信息中(例如,在通用约束信息语法结构的开始处),指示门控标志对应的约束信息组中的至少一个约束标志存在于该约束信息中。在实施例中,至少一个约束信息组中的一个或多个约束信息组存在于已编码视频比特流中。偏移量和长度可以存在于约束信息中,指示至少一个约束信息组中的一个或多个约束信息组中的每个约束信息组中的至少一个约束标志存在于约束信息中。可以在已编码视频比特流中用信号通知偏移量的数量。

[0211] 在实施例中,零个或多个偏移量(或约束信息偏移量)(例如,由语法元素constraint_info_offset[]指示)以及对应的零个或多个长度(或约束信息长度)(例如,由语法元素constraint_info_length[]指示)可以存在于约束信息中,例如在通用约束信息语法结构的开始处。零个或多个约束信息偏移量的数量以及零个或多个约束信息长度的数量可以由一个或多个语法元素来显式地指示。

[0212] 在示例中,零个或多个约束信息偏移量的数量与零个或多个约束信息长度的数量是相同的,并且零个或多个约束信息偏移量的数量(由M表示)由语法元素(例如num_constraint_info_set)显式地指示(或用信号通知)。M可以是大于或等于0的整数。M为0可以指示,在约束信息中不存在约束信息偏移量和约束信息长度。

[0213] 在实施例中,约束信息偏移量(例如,语法元素constraint_info_offset[i])和约束信息长度(例如,语法元素constraint_info_length[i])可以指定存在于约束信息(例如,通用约束信息语法结构)中的可用约束标志。在示例中,当约束信息偏移量(例如,语法元素syntax element constraint_info_offset[i])的值等于5,并且约束信息长度(例如,语法元素constraint_info_length[i])的值等于3时,约束信息(例如,通用约束信息语法结构)中存在第五约束标志、第六约束标志和第七约束标志。

[0214] 在示例中,如果M等于0,则约束信息(例如,通用约束信息语法结构)中不存在约束

标志。

[0215] 在示例中,游程长度编解码(run-length coding)可被用于对以预定顺序(或给定顺序)指定的约束标志进行编解码。

[0216] 在实施例中,在以预定顺序(或给定顺序)指定约束标志的情况下,可以使用游程编解码(run-coding)。代替直接对约束标志进行编解码,“跳过”(“skip”)值的被适当编码的列表可以指示等于零的约束标志,随后的约束标志被暗示为等于1。如果(i)约束标志的数量很大并且(ii)一小部分约束标志等于1,则上述的游程编解码可能会特别有效。

[0217] 在实施例中,至少一个约束信息组中的一个或多个约束信息组存在于已编码视频比特流中。根据预定的顺序用信号通知至少一个约束信息组中的一个或多个约束信息组中的多个约束标志。因此,可以对多个约束标志进行游程编解码(例如,游程编码或游程解码)。此外,可以基于多个约束标志来确定针对编码块的子集的预测信息。

[0218] 在实施例中,门控标志对应的约束信息组中的至少一个约束标志包括根据预定顺序用信号通知的多个约束标志。因此,可以对多个约束标志进行游程编解码(例如,游程编码或游程解码)。

[0219] 在实施例中,可以在视频编解码标准(例如,VVC规范)或外部表等中指定约束标志的完整列表。在示例中,约束标志中的一个或多个可用约束标志例如仅通过以下中的一个或多个来指示:可用约束标志的数量(例如,num_available_constraint_flags)、一个或多个门控标志(或一个或多个约束组存在标志)、约束信息偏移量信息和约束信息长度信息等存在于已编码视频流中。

[0220] 在示例中,约束标志的完整列表被指定并且可被用于编码器和解码器。约束标志的完整列表可以存储在解码器中。约束标志的完整列表可以包括100个约束标志。100个约束标志中的10个约束标志存在于针对CLVS的约束信息中,因此可被用于CLVS中的编码块的子集。100个约束标志中的10个约束标志被称为10个可用约束标志。在示例中,用信号通知可用约束标志的数量(例如,10个)。在示例中,10个可用约束标志在两个约束信息组中,并且由第一门控标志和第二门控标志来进行门控。因此,可以用信号通知第一门控标志和第二门控标志,以指示10个可用约束标志。

[0221] 在示例中,用信号通知第一约束信息偏移量(例如,语法元素constraint_info_offset[0])和第一约束信息长度(例如,语法元素constraint_info_length[0])。用信号通知第二约束信息偏移量(例如,语法元素constraint_info_offset[1])和第二约束信息长度(例如,语法元素constraint_info_length[1])。例如,语法元素constraint_info_offset[0]为15,语法元素constraint_info_length[0]为3,语法元素constraint_info_offset[1]为82,语法元素constraint_info_length[1]为7,因此指示完整列表(例如,100个约束标志)中的第15个到第17个约束标志以及第82个到第88个约束标志在约束信息中可用或者存在。

[0222] 在实施例中,可以采用适当的控制信息来组合用于对约束标志进行有效编解码的各种技术(或方法、实施例、示例)中的任何一种。该组合可以是两种或更多种这类技术的适当组合。可替代地,可以独立地使用各种技术(或方法、实施例、示例)中的一种。可以将约束标志分组。在特定的一个或多个组中,可以使用游程编解码,同时其他的一个或多个组可以采用简单的二进制编解码。

[0223] 约束标志的最大数量的值(例如,MaxNumConstraintFlags)可以被预定义为16、32、64、128等。

[0224] 约束标志的最大数量(例如,MaxNumConstraintFlags)的值可以通过档位信息(诸如general_profile_idc或general_sub_profile_idc等)或编解码器版本信息来确定,使得约束标志的数量(例如,num_available_constraint_flags(1301))的范围可以通过档位信息或版本信息来加以限制。例如,主档位(main profile)(例如,其中,MaxNumConstraintFlags=64)中的约束标志的数量(例如,num_available_constraint_flags(1301))的值可以在0至64的范围内,同时高级档位(advanced profile)(例如,其中,MaxNumConstraintFlags=128)中的约束标志的数量(例如,num_available_constraint_flags(1301))的值可以在0至128的范围内。

[0225] 在实施例中,约束标志的数量(例如,num_available_constraint_flags)的值可以被推断为等于由档位信息(诸如general_profile_idc或general_sub_profile_idc等)或编解码器版本信息预定义的值,使得num_available_constraint_flags的值可以在没有显式信令的情况下得到确定。

[0226] 在实施例中,在图12中,保留的字节信息可以存在于通用约束信息语法结构中。例如,标志gci_num_reserved_bytes(1203)和gci_reserved_bytes[] (1204)可以存在于通用约束信息语法结构中,用于扩展通用约束信息语法结构。标志gci_num_reserved_bytes可以指定保留的约束字节的数量。在示例中,保留的约束字节用于用信号通知附加标志(例如,附加约束标志)。标志gci_reserved_byte[]可以具有任何适当的值。

[0227] 在实施例中,gci_num_reserved_bytes的值可以由档位信息(诸如general_profile_idc或general_sub_profile_idc等)或编解码器版本信息来限制或确定。对于基本档位(base profile)(或主档位),标志gci_num_reserved_bytes的值可以为0。对于扩展档位(extended profile)(或高级档位),gci_num_reserved_bytes的值可以大于0。

[0228] 可以在已编码视频比特流中用信号通知场序列标志。场序列标志可以指示,输出层中的图片是否使用场编解码来进行编解码。场序列标志可以在SPS中作为sps_field_seq_flag而被用信号通知。在实施例中,标志sps_field_seq_flag可以存在于SPS中。标志sps_field_seq_flag等于1可以指示,CLVS传送表示场的图片。标志sps_field_seq_flag等于0可以指示,CLVS传送表示帧的图片。

[0229] 在图12的通用约束信息语法结构中,可以存在标志general_frame_only_constraint_flag。标志general_frame_only_constraint_flag等于1可以指定,输出层集的范围(例如,01sInScope)传送表示帧的图片。标志general_frame_only_constraint_flag等于0指定,输出层集的范围(例如,01sInScope)传送可以表示或者可以不表示帧的图片。在实施例中,标志general_frame_only_constraint_flag指示,是否使用场编解码来对输出层集中的图片进行编解码。输出层集可以包括编码块的子集。标志sps_field_seq_flag可以基于标志general_frame_only_constraint_flag(例如,是1)而为假,该标志general_frame_only_constraint_flag指示,未使用场编解码来对图片的子集进行编解码。图片的子集可以在输出层集的一个层中。

[0230] 当标志general_frame_only_constraint_flag等于1时,标志sps_field_seq_flag的值可以等于0。

[0231] 在实施例中,标志pps_mixed_nalu_type_in_pic_flag可以存在于PPS中。标志pps_mixed_nalu_types_in_pic_flag等于1可以指定,参考PPS的每个图片具有多于一个的VCL NAL单元,并且VCL NAL单元不具有相同的nal_unit_type值。标志pps_mixed_nalu_types_in_pic_flag等于0可以指定,参考PPS的每个图片具有一个或多个VCL NAL单元,并且参考PPS的每个图片的VCL NAL单元具有相同的nal_unit_type值。在图12的通用约束信息语法结构中,可以存在标志no_mixed_nalu_type_in_pic_constraint_flag。标志no_mixed_nalu_types_in_pic_constraint_flag等于1可以指定,pps_mixed_nalu_types_in_pic_flag的值应等于0。标志no_mixed_nalu_types_in_pic_constraint_flag等于0,则不施加这样的约束。

[0232] 在实施例中,标志general_one_picture_only_constraint_flag可以存在于图12中的通用约束信息语法结构中。标志general_one_picture_only_constraint_flag等于1可以指定,在比特流中仅存在一个已编码图片。标志general_one_picture_only_constraint_flag等于0,则不施加这样的约束。

[0233] 在实施例中,标志single_layer_constraint_flag可以存在于图12中的通用约束信息语法结构中。标志single_layer_constraint_flag等于1可以指定,sps_video_parameter_set_id应等于0。标志single_layer_constraint_flag等于0,则不施加这样的约束。当标志general_one_picture_only_constraint_flag等于1时,标志single_layer_constraint_flag的值可以等于1。

[0234] 在实施例中,标志all_layers_independent_constraint_flag可以存在于图12中的通用约束信息语法结构中。标志all_layers_independent_constraint_flag等于1可以指定,标志vps_all_independent_layers_flag可以等于1。标志all_layers_independent_constraint_flag等于0,则不施加这样的约束。当标志single_layer_constraint_flag等于1时,标志all_layers_independent_constraint_flag的值可以等于1。

[0235] 在实施例中,标志no_res_change_in_clvs_constraint_flag可以存在于图12中的通用约束信息语法结构中。标志no_res_change_in_clvs_constraint_flag等于1可以指定,标志sps_res_change_in_clvs_allowed_flag可以等于0。标志no_res_change_in_clvs_constraint_flag等于0,则不施加这样的约束。当标志no_ref_pic_resampling_constraint_flag为1时,标志no_res_change_in_clvs_constraint_flag的值可以等于1。

[0236] 在实施例中,标志no_mixed_nalu_types_in_pic_constraint_flag可以存在于图12中的通用约束信息语法结构中。标志no_mixed_nalu_type_in_pic_constraint_flag等于1指定,标志pps_mixed_nalu_types_in_pic_flag的值可以等于0。标志no_mixed_nalu_types_in_pic_constraint_flag等于0,则不施加这样的约束。当标志one_subpic_per_pic_constraint_flag等于1时,标志no_mixed_nalu_type_in_pic_constraint_flag的值可以等于1。

[0237] 在实施例中,标志no_trail_constraint_flag可以存在于图12中的通用约束信息语法结构中。标志no_trail_constraint_flag等于1可以指定,在0lsInScope中可以不存在nuh_unit_type等于TRAIL_NUT的NAL单元。标志no_trail_constraint_flag等于0,则不施加这样的约束。当标志general_one_picture_only_constraint_flag等于1时,标志no_trail_constraint_flag可以等于1。

[0238] 在实施例中,标志no_stsa_constraint_flag可以存在于图12中的通用约束信息语法结构中。标志no_stsa_constraint_flag等于1可以指定,在01sInScope中可以不存在nuh_unit_type等于STSA_NUT的NAL单元。标志no_stsa_constraint_flag等于0,则不施加这样的约束。当标志general_one_picture_only_constraint_flag等于1时,标志no_stsa_constraint_flag可以等于1。

[0239] 在实施例中,标志no_idr_constraint_flag可以存在于图12中的通用约束信息语法结构中。no_idr_constraint_flag等于1可以指定,在01sInScope中可以不存在nuh_unit_type等于IDR_W_RADL或IDR_N_LP的NAL单元。标志no_idr_constraint_flag等于0,则不施加这样的约束。

[0240] 在实施例中,标志no_cra_constraint_flag可以存在于图12中的通用约束信息语法结构中。标志no_cra_constraint_flag等于1可以指定,在01sInScope中可以不存在nuh_unit_type等于CRA_NUT的NAL单元。标志no_cra_constraint_flag等于0,则不施加这样的约束。

[0241] 在实施例中,标志no_rasl_constraint_flag可以存在于图12中的通用约束信息语法结构中。标志no_rasl_constraint_flag等于1可以指定,在01sInScope中可以不存在nuh_unit_type等于RASL_NUT的NAL单元。标志no_rasl_constraint_flag等于0,则不施加这样的约束。当标志no_cra_constraint_flag等于1时,标志no_rasl_constraint_flag的值可以等于1。

[0242] 在实施例中,标志no_radl_constraint_flag可以存在于图12中的通用约束信息语法结构中。标志no_radl_constraint_flag等于1可以指定,在01sInScope中可以不存在nuh_unit_type等于RADL_NUT的NAL单元。标志no_radl_constraint_flag等于0,则不施加这样的约束。当标志no_idr_constraint_flag等于1并且标志no_cra_constraint_flag等于1时,标志no_rasl_constraint_flag的值可以等于1。

[0243] 图15A至图15D示出了根据本公开的实施例的示例性通用约束信息语法结构(1500)。通用约束信息语法结构(1500)可以表示约束信息(例如,被称为通用约束信息)。通用约束信息(或通用约束信息语法结构(1500))可以包括用于指示约束标志是否存在于通用约束信息语法结构(1500)中的标志(例如,gci_present_flag)。在示例中,如果标志(例如,gci_present_flag)为真,则在通用约束信息语法结构(1500)中可以存在约束标志中的一个或多个。如果标志(例如,gci_present_flag)为假,则在通用约束信息语法结构中不存在约束标志(1500)。

[0244] 如以上参考图14A至图14B所描述的,图15A至图15D中的约束标志可以被分类为组(例如,约束信息组)。约束信息组中的约束标志是否可以存在于通用约束信息语法结构(1500)中可以例如通过约束信息组的相应门控标志来加以控制。

[0245] 通用约束信息语法结构(1500)可以包括任何适当数量的约束信息组。在示例中,通用约束信息语法结构(1500)可以进一步包括未被包括在约束信息组中的一个或多个约束标志,因此该一个或多个约束标志在约束信息组之外。

[0246] 在图15A至图15D所示的示例中,通用约束信息语法结构(1500)包括九个约束信息组(1510) - (1518)。约束信息组(1510)与通用约束信息相关,并且包括三个约束标志(例如,gci_intra_only_constraint_flag,gci_all_layers_indepdent_constraint_flag,gci_

one_au_only_constraint_flag)。在示例中,门控标志可以对约束信息组(1510)是否存在于通用约束信息语法结构(1500)中进行门控或控制。

[0247] 约束信息组(1511)及对应的约束标志与图片格式约束信息相关。在示例中,门控标志可以对约束信息组(1511)是否存在于通用约束信息语法结构(1500)中进行门控或控制。

[0248] 约束信息组(1512)及对应的约束标志与NAL单元类型约束信息相关。在示例中,门控标志可以对约束信息组(1512)是否存在于通用约束信息语法结构(1500)中进行门控或控制。

[0249] 约束信息组(1513)及对应的约束标志与图块、切片或子图片分区约束信息相关。在示例中,门控标志可以对约束信息组(1513)是否存在于通用约束信息语法结构(1500)中进行门控或控制。

[0250] 约束信息组(1514)及对应的约束标志与CTU和块分区约束信息相关。在示例中,门控标志可以对约束信息组(1514)是否存在于通用约束信息语法结构(1500)中进行门控或控制。

[0251] 约束信息组(1515)及对应的约束标志与帧内编解码约束信息相关。在示例中,门控标志可以对约束信息组(1515)是否存在于通用约束信息语法结构(1500)中进行门控或控制。

[0252] 约束信息组(1516)及对应的约束标志与帧间编解码约束信息相关。在示例中,门控标志可以对约束信息组(1516)是否存在于通用约束信息语法结构(1500)中进行门控或控制。

[0253] 约束信息组(1517)及对应的约束标志与变换、量化和残差约束信息相关。在示例中,门控标志可以对约束信息组(1517)是否存在于通用约束信息语法结构(1500)中进行门控或控制。

[0254] 约束信息组(1518)及对应的约束标志与环路滤波器约束信息相关。在示例中,门控标志可以对约束信息组(1518)是否存在于通用约束信息语法结构(1500)中进行门控或控制。

[0255] 在实施例中,在图15D中,语法元素gci_num_reserved_bits和gci_reserved_zero_bit[]可以存在于通用约束信息语法结构(1500)中,用于扩展通用约束信息语法结构。语法元素gci_num_reserved_bits可以指定保留的约束位的数量。语法元素gci_reserved_zero_bit[]可以具有任何适当的值。在示例中,语法元素gci_reserved_zero_bit[]的存在以及gci_reserved_zero_bit[]的值不会对某些视频编解码标准(例如,VVC规范)中指定的解码过程产生影响。语法元素gci_reserved_zero_bit[]的存在以及gci_reserved_zero_bit[]的值可以被用于某些视频编解码标准。

[0256] 图16示出了根据本公开的实施例的PTL语法结构中的PTL语法元素的集合的示例。PTL语法元素的集合可以包括general_profile_idc、general_tier_flag、general_level_idc、ptl_frame_only_constraint_flag、ptl_multilayer_enabled_flag、ptl_sublayer_level_present_flag[]、ptl_reserved_zero_bit、sublayer_level_idc[]、ptl_num_sub_profiles、general_sub_profile_idc、和/或通用约束信息。PTL语法元素的集合中的一些(例如,general_profile_idc、general_tier_flag、通用约束信息、ptl_num_sub_

profiles) 是否可以存在可以基于标志 (例如, profileTierPresentFlag) 的值。

[0257] 具有一个或多个层的视频序列可以被编解码为场编解码结构, 其中两个或更多个连续的场图片可以包括全帧图片。可以在诸如参数集或SEI消息等高级别语法结构中用信号通知场编解码信息。

[0258] 在实施例中, 每个隔行扫描视频帧针对每一帧具有两个场。在一些示例中, 3-2下拉 (3:2下拉) 可被用于将每秒24帧转换为每秒29.97或30帧。3:2下拉可以将4个帧转换为5个帧, 而且速度会略有下降。

[0259] 在实施例中, 如图17中所示, 可以在SPS中用信号通知标志 (例如, sps_field_seq_flag), 以指示是否使用场编解码。标志 (例如, sps_field_seq_flag) 等于1可以指示, CLVS传送表示场的图片。标志 (例如, sps_field_seq_flag) 等于0可以指示, CLVS传送表示帧的图片。当标志 (例如, general_frame_only_constraint_flag) 等于1时, 标志 (例如, sps_field_seq_flag) 的值可以等于0。在示例中, 标志 (例如, general_frame_only_constraint_flag) 等于1指示, 标志 (例如, sps_field_seq_flag) 的值是0。

[0260] 在实施例中, 当标志 (例如, sps_field_seq_flag) 等于1时, 针对CLVS中的每个已编码图片, 可以存在帧-场信息SEI消息。视频编解码标准或技术可能会需要这种存在。

[0261] 在实施例中, 解码过程可以不将表示场或帧的图片与其他图片区别对待。表示场的图片的序列可以用单个场的图片尺寸来进行编解码。例如, 表示1080i场的图片可以具有1920×540的裁剪输出尺寸, 同时序列图片速率可以表示源场的速率 (例如, 通常在50Hz至60Hz之间), 而不是源帧速率 (例如, 通常在25Hz至30Hz之间)。

[0262] 在实施例中, 标志 (例如, sps_field_seq_flag) 的值可以跨层相同。在示例中, 标志 (例如, sps_field_seq_flag) 的值可以在CVS中由CLVS参考的所有SPS中相同, 例如, 因为所有层可以具有相同的场结构或相同的帧结构。

[0263] 在实施例中, 层A的标志 (例如, sps_field_seq_flag) 的值等于输出层集中的层A的依赖层的标志 (例如, sps_field_seq_flag) 的值。

[0264] 在实施例中, 当标志 (例如, sps_field_seq_flag) 等于1时, 包括相同帧的所有场可以参考相同的PPS, 使得相同帧的所有场可以具有相同的图片大小、一致性窗口以及图块分区。

[0265] 在实施例中, 帧的两个场可以具有相同的分区结构, 该分区结构具有相同的图块语法元素或相同的子图片语法元素。

[0266] 在实施例中, 帧的两个场可以具有相同的层结构, 该层结构具有相同数量的层和子层。

[0267] 在实施例中, 如图18中所示, 可以在VPS中用信号通知标志 (例如, vps_field_seq_flag), 以指示参考VPS的层是否被编解码为场或帧。标志 (例如, vps_field_seq_flag) 等于1可以指示, 参考VPS的CVS传送表示场的图片。标志 (例如, vps_field_seq_flag) 等于0可以指示, 参考VPS的CVS传送表示帧的图片。当标志 (例如, general_frame_only_constraint_flag) 等于1时, 标志 (例如, sps_field_seq_flag) 的值应等于0。

[0268] 当VPS中存在标志 (例如, vps_field_seq_flag) 时, 参考VPS的SPS中的标志 (例如, sps_field_seq_flag) 的值可以等于标志 (例如, vps_field_seq_flag) 的值。

[0269] SPS中的高级别语法清理

[0270] 根据本公开的实施例,可以在序列参数集 (SPS) 中执行以下高级别语法 (HLS) 的清理: (1) 用信号通知sps_picture_header_in_slice_header_flag; (2) 跨层对齐subpic_treated_as_pic_flag值; (3) 对小图片的子图片数量的约束; (4) sps_subpic_id_len_minus1的取值范围的校正。

[0271] 1、用信号通知picture_header_in_slice_header标志

[0272] 当图片头被包含在切片头中时,假设不使用子图片信息,并且在参考序列参数集 (SPS) 的CLVS中每个图片仅存在一个切片。该假设是假定性地达成一致,并未明确描述。为了进行HTTP实时流清理,可以在SPS中定义并使用新的标志sps_picture_header_in_slice_header_flag。sps_picture_header_in_slice_header_flag等于1指定,图片头 (PH) 语法结构存在于切片头中,并且在CLVS中每个图片仅存在一个切片。另外,当sps_picture_header_in_slice_header_flag等于1时,不存在子图片信息。

[0273] 以下是新标志sps_picture_header_in_slice_header_flag的示例语义和语法定义:

```
seq_parameter_set_rbsp() { Descriptor
    ...
    sps_picture_header_in_slice_header_flag u(1)
    if( !sps_picture_header_in_slice_header_flag )
        subpic_info_present_flag u(1)
[0274] if( subpic_info_present_flag ) {
        ...
    }
    ...
}
```

[0275] sps_picture_header_in_slice_header_flag等于1指定,在切片头中存在PH语法结构,并且在CLVS中每个图片仅存在一个切片。sps_picture_header_in_slice_header_flag等于0指定,在CLVS中的切片头中不存在PH语法结构。

[0276] subpic_info_present_flag等于1指定,对于CLVS存在子图片信息,并且在CLVS的每个图片中可以存在一个或多于一个的子图片。subpic_info_present_flag等于0指定,对于CLVS不存在子图片信息,并且在CLVS的每个图片中仅存在一个子图片。当不存在时,subpic_info_present_flag的值被推断为等于0。

[0277] 当res_change_in_clvs_allowed_flag等于1时,subpic_info_present_flag的值应等于0。

[0278] 在一个实施例中,在图片头中用信号通知picture_header_in_slice_header_flag。

[0279] picture_header_in_slice_header_flag等于1指定,在切片头中存在PH语法结构。picture_header_in_slice_header_flag等于0指定,在切片头中不存在PH语法结构。

[0280] picture_header_in_slice_header_flag的值应等于sps_picture_header_in

slice_header_flag。

[0281] 当picture_header_in_slice_header_flag对于已编码切片等于1时,比特流一致性的要求是,在CLVS中不应存在nal_unit_type等于PH_NUT的VCL NAL单元。

[0282] 当picture_header_in_slice_header_flag等于0时,当前图片中的所有已编码切片都应具有等于0的picture_header_in_slice_header_flag,并且当前PU应具有PH NAL单元。

[0283] 此外,可以将约束标志pic_header_in_slice_header_constraint_flag添加到general_constraint_info()中。当pic_header_in_slice_header_constraint_flag等于1时,PH中的picture_header_in_slice_header_flag的值应等于1。

[0284]	general_constraint_info() {	描述符
	...	
	one_tile_per_pic_constraint_flag	u(1)
	one_slice_per_pic_constraint_flag	u(1)
	one_subpic_per_pic_constraint_flag	u(1)
	pic_header_in_slice_header_constraint_flag	u(1)
	...	
	}	

[0285] pic_header_in_slice_header_constraint_flag等于1指定,PH语法结构存在于切片头中。pic_header_in_slice_header_constraint_flag等于0,则不施加这样的约束。当pic_header_in_slice_header_constraint_flag应等于1时,picture_header_in_slice_header_flag的值应等于1。

[0286] 当pic_header_in_slice_header_constraint_flag等于1时,rpl_info_in_ph_flag、dbf_info_in_ph_flag、sao_info_in_ph_flag、wp_info_in_ph_flag、qp_delta_info_in_ph_flag应等于0。

[0287] 当pic_header_in_slice_header_constraint_flag等于1时,rect_slice_flag应等于1。

[0288] 当pic_header_in_slice_header_constraint_flag等于1时,subpic_info_present_flag应等于0。

[0289] 当pic_header_in_slice_header_constraint_flag等于1时,separate_colour_plane_flag应等于0。

[0290] 以下是针对新的约束标志的示例性语法定义:

[0291]	general_constraint_info() {	描述符
		符

[0292]

general_progressive_source_flag	u(1)
general_interlaced_source_flag	u(1)
general_non_packed_constraint_flag	u(1)
general_frame_only_constraint_flag	u(1)
general_non_projected_constraint_flag	u(1)
intra_only_constraint_flag	u(1)
max_bitdepth_constraint_idc	u(4)
max_chroma_format_constraint_idc	u(2)
no_res_change_in_clvs_constraint_flag	u(1)
one_tile_per_pic_constraint_flag	u(1)
one_slice_per_pic_constraint_flag	u(1)
one_subpic_per_pic_constraint_flag	u(1)
pic_header_in_slice_header_constraint_flag	u(1)
no_qtbt_dual_tree_intra_constraint_flag	u(1)
no_partition_constraints_override_constraint_flag	u(1)
no_sao_constraint_flag	u(1)
no_alf_constraint_flag	u(1)
no_ccalf_constraint_flag	u(1)
no_joint_cbr_constraint_flag	u(1)
no_ref_wraparound_constraint_flag	u(1)
no_temporal_mvp_constraint_flag	u(1)
no_sbtmvp_constraint_flag	u(1)
no_amvr_constraint_flag	u(1)
no_bdof_constraint_flag	u(1)
no_dmvr_constraint_flag	u(1)
no_cclm_constraint_flag	u(1)
no_mts_constraint_flag	u(1)
no_sbt_constraint_flag	u(1)
no_affine_motion_constraint_flag	u(1)
no_bcw_constraint_flag	u(1)
no_ibc_constraint_flag	u(1)

[0293]

no_ciip_constraint_flag	u(1)
no_fpel_mmvd_constraint_flag	u(1)
no_gpm_constraint_flag	u(1)
no_ladf_constraint_flag	u(1)
no_transform_skip_constraint_flag	u(1)
no_bdpcm_constraint_flag	u(1)
no_qp_delta_constraint_flag	u(1)
no_dep_quant_constraint_flag	u(1)
no_sign_data_hiding_constraint_flag	u(1)
no_mixed_nalu_types_in_pic_constraint_flag	u(1)
no_trail_constraint_flag	u(1)
no_stsa_constraint_flag	u(1)
no_rasl_constraint_flag	u(1)
no_radl_constraint_flag	u(1)
no_idr_constraint_flag	u(1)
no_cra_constraint_flag	u(1)
no_gdr_constraint_flag	u(1)
no_aps_constraint_flag	u(1)
while(!byte_aligned())	
gci_alignment_zero_bit	f(1)
num_reserved_constraint_bytes	u(8)
for(i = 0; i < num_reserved_constraint_bytes; i++)	
gci_reserved_constraint_byte[i]	u(8)
}	

[0294] 以下是针对新约束标志的示例语义定义：

[0295] separate_colour_plane_flag等于1指定,4:4:4色度格式的第三个颜色分量被单独编码。separate_colour_plane_flag等于0指定,颜色分量不被单独编码。当separate_colour_plane_flag不存在时,其被推断为等于0。当separate_colour_plane_flag等于1时,已编码图片由三个单独的分量组成,每个分量由一个颜色平面(Y、Cb或Cr)的已编码样本组成,并且使用单色编解码语法。在这种情况下,每个颜色平面与特定的colour_plane_id值相关联。

[0296] 注意——在具有不同colour_plane_id值的颜色平面之间的解码过程中不存在依赖性。例如,具有一个colour_plane_id值的单色图片的解码过程不使用来自具有不同colour_plane_id值的单色图片的任何数据来用于帧间预测。

[0297] 取决于separate_colour_plane_flag的值,变量ChromaArrayType的值被分配如

下:

[0298] -如果separate_colour_plane_flag等于0,则将ChromaArrayType设置为等于chroma_format_idc。

[0299] -否则(separate_colour_plane_flag等于1),将ChromaArrayType设置为等于0。

[0300] 当pic_header_in_slice_header_constraint_flag等于1时,separate_colour_plane_flag应等于0。

[0301] subpic_info_present_flag等于1指定,对于CLVS存在子图片信息,并且在CLVS的每个图片中可以存在一个或多于一个的子图片。subpic_info_present_flag等于0指定,对于CLVS不存在子图片信息,并且在CLVS的每个图片中仅存在一个子图片。

[0302] 当res_change_in_clvs_allowed_flag等于1时,subpic_info_present_flag的值应等于0。

[0303] 注意——当比特流是子比特流提取过程的结果并且仅包含子比特流提取过程的输入比特流的子图片的子集时,可能需要在SPS的RBSP中将subpic_info_present_flag的值设置为等于1。

[0304] 当pic_header_in_slice_header_constraint_flag等于1时,subpic_info_present_flag应等于0。

[0305] rect_slice_flag等于0指定,每个切片内的图块按光栅扫描顺序排列,并且不在PPS中用信号通知切片信息。rect_slice_flag等于1指定,每个切片内的图块覆盖图片的矩形区域,并且在PPS中用信号通知切片信息。当不存在时,rect_slice_flag被推断为等于1。当subpic_info_present_flag等于1时,rect_slice_flag的值应等于1。

[0306] 当pic_header_in_slice_header_constraint_flag等于1时,rect_slice_flag应等于1。

[0307] qp_delta_info_in_ph_flag等于1指定,QP增量信息存在于PH语法结构中并且不存在于参考不包含PH语法结构的PPS的切片头中。qp_delta_info_in_ph_flag等于0指定,QP增量信息不存在于PH语法结构中并且可以存在于参考不包含PH语法结构的PPS的切片头中。

[0308] 当pic_header_in_slice_header_constraint_flag等于1时,rpl_info_in_ph_flag、dbf_info_in_ph_flag、sao_info_in_ph_flag、wp_info_in_ph_flag、qp_delta_info_in_ph_flag应等于0。

[0309] one_subpic_per_pic_constraint_flag等于1指定,每个图片应仅包含一个子图片。one_subpic_per_pic_constraint_flag等于0,则不施加这样的约束。当one_slice_per_pic_constraint_flag等于1时,one_subpic_per_pic_constraint_flag的值应等于1。

[0310] pic_header_in_slice_header_constraint_flag等于1指定,PH语法结构存在于切片头中。pic_header_in_slice_header_constraint_flag等于0,则不施加这样的约束。当pic_header_in_slice_header_constraint_flag应等于1时,picture_header_in_slice_header_flag的值应等于1。

[0311] 2、跨层对齐subpic_treated_as_pic_flag值

[0312] 对于从多层比特流提取子图片,当subpic_treated_as_pic_flag[i]等于1时,对于OLS中的每个输出层及其参考层,输出层及其参考层中的所有图片对于在0至sps_num_

subpics_minus1范围内的每个j值,都应具有等于1的subpic_treated_as_pic_flag[j]的值。因此,根据本公开的实施例,增加了以下约束。

[0313] subpic_treated_as_pic_flag[i]等于1指定,CLVS中的每个已编码图片的第i子图片在除环路滤波操作之外的解码过程中被视为图片。subpic_treated_as_pic_flag[i]等于0指定,CLVS中的每个已编码图片的第i子图片在除环路滤波操作之外的解码过程中不被视为图片。当不存在时,subpic_treated_as_pic_flag[i]的值被推断为等于sps_independent_subpics_flag。

[0314] 当subpic_treated_as_pic_flag[i]等于1时,比特流一致性的要求是,对于OLS中的每个输出层及其参考层,所有以下条件都为真,该OLS包括作为输出层的包含第i子图片的层:

[0315] -输出层及其参考层中的所有图片应具有相同的pic_width_in_luma_samples的值和相同的pic_height_in_luma_samples的值。

[0316] -对于在0至sps_num_subpics_minus1(包含端值)的范围内的每个j值,输出层及其参考层所参考的所有SPS应具有相同的sps_num_subpics_minus1的值并且应分别具有相同的subpic_ctu_top_left_x[j]、subpic_ctu_top_left_y[j]、subpic_width_minus1[j]、subpic_height_minus1[j]和loop_filter_across_subpic_enabled_flag[j]的值。

[0317] -对于在0至sps_num_subpics_minus1(包含端值)范围内的每个j值,输出层及其参考层中的每个存取单元中的所有图片应具有相同的SubpicIdVal[j]的值。

[0318] -对于0至sps_num_subpics_minus1(包含端值)范围内的每个j值,输出层及其参考层中的所有图片应具有等于1的subpic_treated_as_pic_flag[j]的值。

[0319] 3、对小图片的子图片数量的约束

[0320] 在一个实施例中,可以施加附加的约束,使得当图片大小不大于CtbSizeY时,子图片的数量应等于1。

[0321] 以下是针对附加约束的示例语法定义和/或语义定义:

[0322] 当pic_width_max_in_luma_samples不大于CtbSizeY并且pic_height_max_in_luma_samples不大于CtbSizeY时,sps_num_subpics_minus1sh的值应等于0。

[0323] 4、sps_subpic_id_len_minus1的取值范围的校正

[0324] 在一个实施例中,可以如下修改sps_subpic_id_len_minus1的语义,使得该语法元素可以具有更准确的取值范围。

[0325] sps_subpic_id_len_minus1加1指定,用于表示语法元素sps_subpic_id[i]、语法元素pps_subpic_id[i](当存在时)和语法元素slice_subpic_id(当存在时)的比特的数量。sps_subpic_id_len_minus1的值应在0至15(包含端值)的范围内。 $1 \ll (\text{sps_subpic_id_len_minus1})$ 的值应大于或等于sps_num_subpics_minus1+1。

[0326] 以上提及的实施例可以单独使用或以任何顺序组合使用。此外,每个实施例,编码器和/或解码器可以由处理电路(例如,一个或多个处理器或者一个或多个集成电路)来实现。在一个示例中,一个或多个处理器执行被存储在非易失性计算机可读介质中的程序。

[0327] 图19示出了概述根据本公开的实施例的过程(1900)的流程图。过程(1900)可用于块的重建,从而为重建中的块生成预测块。本公开中的术语“块”可以被解释为预测块、CB、CU等。在各种实施例中,过程(1900)由处理电路执行,例如,终端装置(310)、(320)、(330)和

(340) 中的处理电路、执行视频编码器 (403) 的功能的处理电路、执行视频解码器 (410) 的功能的处理电路、执行视频解码器 (510) 的功能的处理电路、执行视频编码器 (603) 的功能的处理电路等。在一些实施例中,过程 (1900) 在软件指令中实现,因此当处理电路执行软件指令时,该处理电路执行过程 (1900)。过程从 (S1901) 开始并前进到 (S1910)。

[0328] 在 (S1910) 处,可以获得针对在已编码视频比特流中用信号通知的约束信息中的至少一个约束信息组中的每个约束信息组的门控标志。每个门控标志可以指示与相应门控标志相对应的至少一个约束信息组的约束信息组是否存在于约束信息中。约束信息可以针对已编码视频比特流中的编码块的子集。

[0329] 约束信息可以存在于至少一个图片的高级别语法 (例如,SPS、VPS、DCI、DPS) 中,并且编码块的子集可以包括该至少一个图片中的编码块。

[0330] 在示例中,每个门控标志指示,与相应门控标志相对应的约束信息组不存在于约束信息中,并且在约束信息中不存在约束标志。

[0331] 在 (S1920) 处,可以基于至少一个约束信息组中的约束信息组的门控标志来确定门控标志对应的约束信息组是否存在于约束信息中。门控标志对应的约束信息组可以包括针对编码块的子集的至少一个约束标志。

[0332] 在 (S1930) 处,可以基于门控标志对应的约束信息组是否存在于约束信息中来确定针对编码块的子集的预测信息。

[0333] 在示例中,确定出门控标志对应的约束信息组存在于约束信息中。针对编码块的子集的预测信息可以基于门控标志对应的约束信息组中的至少一个约束标志来确定。

[0334] 在示例中,确定出门控标志对应的约束信息组不存在于约束信息中。针对编码块的子集的预测信息可以独立于门控标志对应的约束信息组来确定。

[0335] 在 (S1940) 处,可以基于预测信息来重建编码块的子集。

[0336] 过程 (1900) 可以被适当地调整。可以修改和/或省略过程 (1900) 中的一个或多个步骤。可以增加一个或多个附加步骤。可以采用任何适当的实施顺序。例如,至少一个约束信息组中的一个或多个约束信息组存在于已编码视频比特流中。可以根据预定的顺序用信号通知至少一个约束信息组中的一个或多个约束信息组中的多个约束标志。可以对多个约束标志进行游程解码,并因此可以基于多个约束标志来确定针对编码块的子集的预测信息。

[0337] 在示例中,门控标志对应的约束信息组中的至少一个约束标志包括根据预定顺序用信号通知的多个约束标志,并且门控标志对应的约束信息组中的多个约束标志可以被游程解码。

[0338] 在示例中,偏移量和长度存在于约束信息中,指示门控标志对应的约束信息组中的至少一个约束标志存在于约束信息中。

[0339] 在示例中,至少一个约束信息组中的一个或多个约束信息组存在于已编码视频比特流中。偏移量和长度存在于约束信息中,指示至少一个约束信息组中的一个或多个约束信息组的每个约束信息组中的至少一个约束标志存在于约束信息中。可以在已编码视频比特流中用信号通知偏移量的数量。

[0340] 在示例中,至少一个约束信息组中的一个或多个约束信息组存在于已编码视频比特流中。约束信息中的一个或多个语法元素可以是字节对齐的。一个或多个语法元素可以

包括针对至少一个约束信息组中的每个约束信息组的门控标志以及至少一个约束信息组中的一个或多个约束信息组中的多个约束标志。

[0341] 在示例中,至少一个门控标志包括仅帧约束标志(例如,general_frame_only_constraint_flag),该仅帧约束标志用于指示输出层集中的图片是否使用场编解码来进行编解码。输出层集可以包括编码块的子集。场序列标志(例如,sps_field_seq_flag)基于仅帧约束标志而为假,该仅帧约束标志用于指示图片的子集没有用场编解码来进行编解码,其中图片的子集在输出层集的一个层中。

[0342] 在示例中,在已编码视频比特流中用信号通知场序列标志,并且场序列标志用于指示输出层中的图片是否用场编解码来进行编解码。

[0343] 本公开中的实施例可以单独使用或以任何顺序组合使用。此外,方法(或实施例)、编码器和解码器中的每一个可以由处理电路(例如,一个或多个处理器或一个或多个集成电路)来实现。在一个示例中,一个或多个处理器执行存储在非易失性计算机可读介质中的程序。

[0344] 本公开的各方面涉及在例如具有多个层的已编码视频流中用信号通知场编解码信息。

[0345] 当图片被编码成包括具有不同质量的多个层的比特流时,该比特流可以具有用于指定可以在解码器侧输出哪些层的语法元素。在支持多层及可缩放性的某些视频编解码器中,可以在VPS中用信号通知一个或多个输出层集。在参数集中有效地用信号通知用于指定输出层集及其依赖性、档位/层级/等级、以及假定解码器参考模型参数的语法元素。当多层视频序列被编码为其中两个或更多个场图片包括一个帧图片的场结构时,可以在诸如SPS、VPS或SEI消息等高级别语法结构中用信号通知场编解码信息。

[0346] 在一些视频编解码技术中,存取单元(AU)可以指一个或多个已编码图片、一个或多个切片、一个或多个图块、一个或多个NAL单元等,它们在给定的时间实例中被捕获并且被合成到相应的图片/切片/图块/NAL单元比特流中。给定的时间实例可以是合成时间。

[0347] 在一些示例中,诸如在HEVC和某些其他视频编解码技术中,图片顺序计数(POC)值可以被用于指示存储在已解码图片缓冲器(DPB,decoded picture buffer)中的多个参考图片当中的选定参考图片。当AU包括一个或多个图片、切片或图块时,属于相同AU的每个图片、切片或图块可以具有相同的POC值,从该POC值可以推导出一个或多个图片、切片或图块是从相同合成时间的内容创建的。因此,具有相同给定POC值的两个图片/切片/图块可以指示,这两个图片/切片/图块属于相同的AU并且具有相同的合成时间。相反,具有不同POC值的两个图片/切片/图块可以指示,这两个图片/切片/图块属于不同的AU并且具有不同的合成时间。

[0348] 在实施例中,前述的严格关系可以被放宽,其中AU可以包括具有不同POC值的图片、切片或图块。通过在AU内允许不同的POC值,可以将POC值用于识别具有相同呈现时间的潜在可独立解码的图片/切片/图块。因此,可以在不改变参考图片选择信令(例如,参考图片集信令或参考图片列表信令)的情况下支持多个可伸缩层,如下面更详细描述。

[0349] 在一些示例中,相对于具有不同POC值的其他图片/切片/图块,仅从POC值来识别图片/切片/图块所属的AU是合乎需要的。这可以如下所描述地实现。在实施例中,可以在诸如NAL单元头、切片头、图块组头、SEI消息、参数集或AU定界符等高级别语法结构中用信号

通知存取单元计数(AUC, access unit count)。AUC的值(或AUC值)可被用于识别哪些NAL单元、图片、切片或图块属于给定的AU。AUC的值可以与不同的合成时间实例相对应。AUC值可以等于POC值的倍数。通过将POC值除以整数值,可以计算出AUC值。在一些示例中,除法操作可能会给解码器实施方案带来一定负担。因此,AUC值的编号空间中的小限制可以允许用移位运算来代替除法运算。例如,AUC值可以等于POC值范围的最高有效位(MSB, Most Significant Bit)值。

[0350] 在实施例中,可以在诸如NAL单元头、切片头、图块组头、SEI消息、参数集或AU定界符等高级别语法结构中用信号通知每个AU的POC周期的值(例如,poc_cycle_au)。每个AU的POC周期的值(例如,poc_cycle_au)可以指示,多少个不同且连续的POC值可以与相同的AU相关联。例如,如果每个AU的POC周期的值(例如,poc_cycle_au)等于4,则POC值等于0至3(包含端值)的图片、切片或图块与AUC值等于0的AU相关联,并且POC值等于4至7(包含端值)的图片、切片或图块与AUC值等于1的AU相关联。因此,可以通过将POC值除以每个AU的POC周期的值的值(例如,poc_cycle_au)来推断AUC的值。

[0351] 在实施例中,每个AU的POC周期的值(例如,poc_cyle_au)可以从位于例如VPS中的信息导出,该信息用于识别已编码视频序列中的空间层或SNR层的数量。虽然如上所述的推导可以在VPS中节省一些比特并因此可以提高编解码效率,但是在谱系低于VPS的适当的高级别语法结构中显式地对每个AU的POC周期的值(例如,poc_cycle_au)进行编解码以最小化针对比特流(例如,图片)的给定小部分的每个AU的POC周期的值(例如,poc_cycle_au)可能是有利的。因为可以在低级别语法结构中对POC值(和/或间接参考POC的语法元素的值)进行编解码,所以该优化可以节省比通过以上推导过程能够节省的比特更多的比特。

[0352] 在实施例中,图20示出了语法表的示例,用以用信号通知VPS(或SPS)中的vps_poc_cycle_au的语法元素(该语法元素指示用于已编码视频序列中的所有图片/切片的poc_cycle_au)以及切片头中的slice_poc_cycle_au的语法元素(该语法元素指示当前切片的poc_cycle_au)。

[0353] 如果每个AU的POC值均匀增加,则将VPS中的语法元素vps_contant_poc_cycle_per_au设置为等于1,并且在VPS中用信号通知语法元素vps_poc_cycle_au。因此,语法元素slice_poc_cycle_au未被显式地用信号通知,并且可以通过将POC的值除以语法元素vps_poc_cycle_au来计算每个AU的AUC的值。

[0354] 如果每个AU的POC值未均匀增加,则将VPS中的语法元素vps_contant_poc_cycle_per_au设置为等于0。因此,不用信号通知语法元素vps_access_unit_cnt,同时,在每个切片或图片的切片头中用信号通知语法元素slice_access_unit_cnt。每个切片或图片可以具有不同的slice_access_unit_cnt值。可以通过将POC的值除以语法元素slice_poc_cycle_au来计算每个AU的AUC的值。

[0355] 图21示出了用于对上述相关工作流进行说明的框图。工作流程开始于(S2101)并且进行到(S2110)。

[0356] 在(S2110)处,可以对VPS或SPS进行解析,并且可以对每个AU的POC周期是否恒定进行识别。

[0357] 在(S2120)处,可以对每个AU的POC周期在已编码视频序列内是否恒定进行确定。如果确定出每个AU的POC周期在已编码视频序列内是恒定的,则工作流进行到(S2130)。否

则, 工作流程进行到 (S2140)。

[0358] 在 (S2130) 处, 可以根据序列级别的 `poc_cycle_au` 值和 POC 值来计算存取单元计数的值。然后, 工作流程进行到 (S2150)。

[0359] 在 (S2140) 处, 可以根据图片级别的 `poc_cycle_au` 值和 POC 值来计算存取单元计数的值。然后, 工作流程进行到 (S2150)。

[0360] 在 (S2150) 处, 可以对 VPS 或 SPS 进行解析, 并且可以对每个 AU 的 POC 周期是否恒定进行识别。

[0361] 在实施例, 即使图片、切片或图块的 POC 值可能不同, 与具有相同的 AUC 值的 AU 相对应的图片、切片或图块也可以与相同的解码或输出时间实例相关联。因此, 在相同的 AU 中的图片、切片或图块之间没有任何解析/解码相依性的情况下, 与相同的 AU 相关联的图片、切片或图块的全部或子集可以被并行解码, 并且可以在相同的时间实例处输出。

[0362] 在实施例, 即使图片、切片或图块的 POC 值可能不同, 与具有相同的 AUC 值的 AU 相对应的图片、切片或图块也可以与相同的合成/显示时间实例相关联。当合成时间被包含在容器格式中时, 即使图片与不同的 AU 相对应, 如果图片具有相同的合成时间, 该图片也可以在相同的时间实例处被显示。

[0363] 在实施例, 每个图片、切片或图块可以在相同的 AU 中具有相同的时间标识符 (例如, `temporal_id`)。与时间实例相对应的图片、切片或图块的全部或子集可以与相同的时间子层相关联。在实施例, 每个图片、切片或图块可以在相同的 AU 中具有相同或不同的一个或多个空间层标识 (例如, `layer_id`)。与时间实例相对应的图片、切片或图块的全部或子集可以与相同或不同的空间层相关联。

[0364] 图 22 示出了视频序列结构的示例, 该视频序列结构具有带有自适应分辨率改变的 `temporal_id`、`layer_id`、POC 值和 AUC 值的组合。在图 22 中所示的示例中, 分别地, $AUC=0$ 的第一 AU 中的图片、切片或图块可以具有 `temporal_id=0` 以及 `layer_id=0` 或 1, 同时 $AUC=1$ 的第二 AU 中的图片、切片或图块可以具有 `temporal_id=1` 以及 `layer_id=0` 或 1。不管 `temporal_id` 和 `layer_id` 的值如何, POC 值是每个图片都增加 1。参考图 22, `poc_cycle_au` 的值可以等于 2。在示例中, 可以优选地将 `poc_cycle_au` 的值设置为等于 (空间可缩放性) 层的数量。在该示例中, POC 值增加 2, 同时 AUC 值增加 1。

[0365] 在以上实施例, 可以通过使用视频标准 (诸如 HEVC) 中的参考图片集 (RPS) 信令或参考图片列表 (RPL) 信令来支持图片间或层间预测结构和参考图片指示的全部或子集。在 RPS 或 RPL 中, 可以通过用信号通知当前图片与所选择的参考图片之间的 POC 的值或 POC 的增量值, 来指示所选择的参考图片。在各种示例中, RPS 和 RPL 可被用于指示图片间或层间预测结构而无需改变信令, 其中具有以下限制。如果参考图片的 `temporal_id` 的值大于当前图片的 `temporal_id` 的值, 则当前图片可以不使用参考图片来进行运动补偿或其他预测。如果参考图片的 `layer_id` 的值大于当前图片的 `layer_id` 的值, 则当前图片可以不使用参考图片来进行运动补偿或其他预测。

[0366] 在实施例, 可以在 AU 内跨多个图片禁用用于时间运动矢量预测的基于 POC 差异的运动矢量缩放。因此, 尽管每个图片在 AU 内可以具有不同的 POC 值, 但是由于在相同的 AU 中具有不同的 POC 的参考图片被认为是具有相同时间实例的参考图片, 所以运动矢量不被缩放, 并且被用于 AU 内的时间运动矢量预测。因此, 当参考图片属于与当前图片相关联的 AU

时,运动矢量缩放函数可以返回1。

[0367] 在实施例中,当参考图片的空间分辨率不同于当前图片的空间分辨率时,可以可选择性地跨多个图片禁用用于时间运动矢量预测的基于POC差异的运动矢量缩放。当允许运动矢量缩放时,基于POC差异以及当前图片与参考图片之间的空间分辨率比率来对运动矢量进行缩放。

[0368] 在实施例中,对于时间运动矢量预测,尤其当poc_cycle_au具有非均匀值时(例如,当vps_contant_poc_cycle_per_au==0时),可以基于AUC差异而非POC差异来对运动矢量进行缩放。否则(例如,当vps_contant_poc_cycle_per_au==1时),基于AUC差异的运动矢量缩放可以与基于POC差异的运动矢量缩放相同。

[0369] 在实施例中,当基于AUC差异对运动矢量进行缩放时,与当前图片相同的AU(具有相同的AUC值)中的参考运动矢量不基于AUC差异被缩放,并且被用于其中不进行缩放的或者基于当前图片与参考图片之间的空间分辨率比例来进行缩放的运动矢量预测。

[0370] 在实施例中,AUC值被用于对AU的边界进行标识并且被用于假设参考解码器(HRD)操作,该操作可能需要具有AU粒度的输入和输出定时。在各种示例中,可以输出AU中具有最高层的已解码图片以供显示。AUC值和layer_id值可以被用于识别输出图片。

[0371] 在实施例中,图片可以包括一个或多个子图片。每个子图片可以覆盖图片的局部区域或整个区域。由子图片支持的区域可以与由另一个子图片支持的区域重叠或不重叠。由一个或多个子图片构成的区域可以覆盖或者不覆盖图片的整个区域。如果图片包括子图片,则由子图片支持的区域可以与由图片支持的区域相同。

[0372] 在实施例中,可以通过与用于已编码图片的编解码方法相似的编解码方法来对子图片进行编解码。可以独立地对子图片进行编解码,或者可以根据另一个子图片或已编码图片来对子图片进行编解码。子图片可以具有或者可以不具有来自另一个子图片或已编码图片的任何解析依赖性。

[0373] 在实施例中,已编码子图片可以被包含在一个或多个层中。层中的已编码子图片可以具有不同的空间分辨率。原始的子图片可以被空间地重采样(例如,上采样或下采样),以不同的空间分辨率参数被编码,并且被包含在与层相对应的比特流中。

[0374] 在实施例中,具有(W,H)的子图片可以被编码并且包含在与层0相对应的已编码比特流中。分别地,W指示子图片的宽度,并且H指示子图片的高度。上采样(或下采样)后的子图片可以从具有原始空间分辨率的子图片中被上采样(或下采样)。上采样(或下采样)后的子图片可以具有空间分辨率($W \cdot S_{w,k}$, $H \cdot S_{h,k}$),并且可以被编码并且包含在与层k相对应的已编码比特流中。参数 $S_{w,k}$ 和 $S_{h,k}$ 可以分别指示水平重采样比率和垂直重采样比率。如果 $S_{w,k}$ 、 $S_{h,k}$ 的值大于1,则重采样是上采样。如果 $S_{w,k}$ 、 $S_{h,k}$ 的值小于1,则重采样是下采样。

[0375] 在实施例中,层中的已编码子图片可以具有与相同子图片或不同子图片中的另一个层中的已编码子图片的视觉质量不同的视觉质量。例如,用量化参数(例如, $Q_{i,n}$)对层n中的子图片i进行编码,同时用量化参数(例如, $Q_{j,m}$)对层m中的子图片j进行编码。

[0376] 在实施例中,层中的已编码子图片可以是可独立地解码的,而没有来自相同局部区域的另一个层中的已编码子图片的任何解析或解码依赖性。在不参考相同局部区域的另一个子图片层的情况下,可独立地解码的子图片层是独立子图片层。独立子图片层中的已编码子图片可以具有或者可以不具有来自相同子图片层中的先前已编码子图片的解码或

解析依赖性。已编码子图片可以不具有来自另一个子图片层中的已编码图片的任何依赖性。

[0377] 在实施例中,层中的已编码子图片可以是可依赖地解码的,具有来自相同局部区域的另一个层中的已编码子图片的任何解析或解码依赖性。可通过参考相同局部区域的另一个子图片层而被可依赖地解码的子图片层是从属子图片层。从属子图片中的已编码子图片可以参考属于相同子图片的已编码子图片、相同子图片层中的先前已编码子图片,或者参考这两种子图片。

[0378] 在实施例中,已编码子图片包括一个或多个独立子图片层以及一个或多个从属子图片层。然而,对于已编码子图片,可以存在至少一个独立子图片层。独立子图片层可以具有等于0的层标识符(layer_id)的值,该层标识符的值可以存在于NAL单元头或其他高级别语法结构中。layer_id等于0的子图片层是基底子图片层。

[0379] 在实施例中,图片可以包括一个或多个前景子图片以及一个背景子图片。背景子图片支持的区域可以等于图片的区域。前景子图片支持的区域可以与背景子图片支持的区域重叠。背景子图片可以是基底子图片层,同时前景子图片可以是非基底子图片层(例如,增强子图片层)。一个或多个非基底子图片层可以参考相同的基底层进行解码。layer_id等于a的每个非基底子图片层可以参考layer_id等于b的非基底子图片层,其中,a大于b。

[0380] 在实施例中,图片可以包括具有或不具有背景子图片的一个或多个前景子图片。每个子图片可以具有其自己的基底子图片层以及一个或多个非基底(增强)层。每个基底子图片层可以被一个或多个非基底子图片层参考。layer_id等于a的每个非基底子图片层可以参考layer_id等于b的非基底子图片层,其中,a大于b。

[0381] 在实施例中,图片可以包括具有或不具有背景子图片的一个或多个前景子图片。(基底或非基底)子图片层中的每个已编码子图片可以被属于相同子图片的一个或多个非基底层子图片以及不属于相同子图片的一个或多个非基底层子图片参考。

[0382] 在实施例中,图片可以包括具有或不具有背景子图片的一个或多个前景子图片。层a中的子图片可以被进一步分区为相同层中的多个子图片。层b中的一个或多个已编码子图片可以参考层a中的分区后的子图片。

[0383] 在实施例中,已编码视频序列(CVS)可以是一组已编码图片。CVS可以包括一个或多个已编码子图片序列(CSPS),其中CSPS可以是将图片的相同局部区域覆盖的一组已编码子图片。CSPS可以具有与已编码视频序列相同或不同的时间分辨率。

[0384] 在实施例中,CSPS可以被编码并且包含在一个或多个层中。CSPS可以包括一个或多个CSPS层。对与CSPS相对应的一个或多个CSPS层进行解码可以重建与相同局部区域相对应的子图片的序列。

[0385] 在实施例中,与CSPS相对应的CSPS层的数量可以与跟另一个CSPS相对应的CSPS层的数量相同或不同。

[0386] 在实施例中,CSPS层可以具有与另一个CSPS层的时间分辨率不同的时间分辨率(例如,帧速率)。原始的(例如,未压缩的)子图片序列可以被时间地重采样(例如,上采样或下采样),以不同的时间分辨率参数被编码,并且被包含在与层相对应的比特流中。

[0387] 在实施例中,具有帧速率F的子图片序列可以被编码并且被包含在与层0相对应的已编码比特流中,同时来自原始子图片序列的具有 $F \cdot St, k$ 的时间上采样(或下采样)后的子

图片序列可以被编码并且被包含在与层k相对应的已编码比特流中。 St,k 指示针对层k的时间采样比率。如果 St,k 的值大于1,则时间重采样过程与帧速率上转换相对应。如果 St,k 的值小于1,则时间重采样过程与帧速率下转换相对应。

[0388] 在实施例中,当具有CSPS层a的子图片被具有CSPS层b的子图片参考以进行运动补偿或任何层间预测时,如果CSPS层a的空间分辨率不同于CSPS层b的空间分辨率,则CSPS层a中的已解码像素被重采样并且被用于参考。重采样过程可能需要上采样滤波或下采样滤波。

[0389] 图23示出了包括layer_id等于0的背景视频CSPS以及多个前景CSPS层的视频流的示例。虽然已编码子图片可以包括一个或多个CSPS层,但是不属于任何前景CSPS层的背景区域可以包括基底层。基底层可以包含背景区域和前景区域。增强CSPS层可以包含前景区域。增强CSPS层可以具有比相同区域处的基底层更好的视觉质量。增强CSPS层可以参考与相同区域相对应的基底层的重建像素和运动矢量。

[0390] 在实施例中,在视频文件中,与基底层相对应的视频比特流被包含在轨道中,同时与每个子图片相对应的CSPS层被包含在分离的轨道中。

[0391] 在实施例中,与基底层相对应的视频比特流被包含在轨道中,同时具有相同layer_id的CSPS层被包含在分离的轨道中。在示例中,与层k相对应的轨道仅包括与层k相对应的CSPS层。

[0392] 在实施例中,每个子图片的每个CSPS层被存储在单独的轨道中。每个轨道可以具有或者可以不具有来自一个或多个其他轨道的任何解析或解码依赖性。

[0393] 在实施例中,每个轨道可以包含与子图片的全部或子集的CSPS层的层i至层j相对应的比特流,其中 $0 < i \leq j \leq k$,并且k是CSPS的最高层。

[0394] 在实施例中,图片包括一个或多个相关联的媒体数据,媒体数据包括深度图、阿尔法图、30几何数据、占用图等。这种相关联的和定时的媒体数据可以被划分成一个或多个数据子流,每个数据子流与一个子图片相对应。

[0395] 在实施例中,图24示出了基于多层子图片方法的视频会议的示例。在视频流中,包含与背景图片相对应的一个基底层视频比特流以及与前景子图片相对应的一个或多个增强层视频比特流。每个增强层视频比特流与CSPS层相对应。在显示中,默认地显示与基底层相对应的图片。t包含一个或多个用户画中画(PIP,picture in a picture)。当通过客户端控件选择出特定用户时,以增强的质量或空间分辨率来对与所选择的用户相对应的增强CSPS层进行解码及显示。

[0396] 图25示出了以上描述的操作的图。操作开始于(S2501)并且进行到(S2510)。

[0397] 在(S2510)处,可以对具有多层的视频比特流进行解码。

[0398] 在(S2520)处,可以对背景区域以及一个或多个前景子图片进行识别。

[0399] 在(S2530)处,可以对是否选择了特定的子图片区域进行确定。如果确定为选择了特定的子图片区域,则操作进行到(S2540)。否则,操作进行到(S2550)。

[0400] 在(S2540)处,可以对增强的子图片进行解码及显示。然后,操作进行到(S2599),并且结束。

[0401] 在(S2550)处,可以对背景区域进行解码及显示。然后,操作进行到(S2599),并且结束。

[0402] 在实施例中,网络中间盒(诸如路由器)可以根据其带宽来选择要发送给用户的层的子集。图片/子图片组织可以被用于带宽自适应。例如,如果用户没有带宽,则路由器根据层或子图片的重要性或基于所使用的设置来剥离层或选择一些子图片,并且可以动态地进行操作以适应带宽。

[0403] 图26示出了360度视频的使用情况。当球形360度图片被投影到平面图片上时,投影360度图片可以被分区为多个子图片以作为基底层。特定子图片的增强层可以被编解码并且被传输到客户端。解码器能够对包括所有子图片的基底层以及所选择的子图片的增强层进行解码。在当前视口(viewport)与所选择的子图片相同时,所显示的图片可以具有比具有增强层的已解码子图片更高的质量。否则,可以以低质量来显示具有基底层的已解码图片。

[0404] 在实施例中,用于显示的任何适当的布局信息可以作为补充信息(诸如SEI消息或元数据)存在于文件中。一个或多个已解码子图片可以根据用信号通知的布局信息被重新定位并显示。布局信息可以由流服务器或广播器用信号通知,或可以由网络实体或云服务器重新生成,或者可以通过用户的自定义设置来确定。

[0405] 在实施例中,当输入图片被划分成一个或多个子区域(例如,一个或多个矩形子区域)时,每个子区域可以被编码为独立层。与局部区域相对应的每个独立层可以具有唯一的layer_id值。对于每个独立层,可以用信号通知子图片大小以及位置信息。例如,图片大小(例如,图片的宽度和/或高度)、左上角的偏移量信息(例如,x_offset,y_offset)。图27示出了所划分的子图片的布局、子图片大小、位置信息以及对应的图片预测结构的示例。可以在诸如一个或多个参数集、切片或图块组的头、或者SEI消息等高级别语法结构中用信号通知包括一个或多个子图片大小以及一个或多个子图片位置的布局信息。

[0406] 在实施例中,与独立层相对应的每个子图片可以在AU内具有唯一的POC值。当通过使用RPS或RPL结构中的一个或多个语法元素来指示被存储在DPB中的图片当中的参考图片时,可以使用与层相对应的每个子图片的一个或多个POC值。

[0407] 在实施例中,为了指示预测结构(例如,层间预测结构),可以不使用layer_id,并且可以使用POC值(例如,POC增量值)。

[0408] 在实施例中,与层(或局部区域)相对应的、POC值等于N的子图片可以被用作或者可以不被用作与用于运动补偿预测的相同层(或相同局部区域)相对应的、POC值等于N+K的子图片的参考图片。在各种示例中,数字K的值可以等于层(例如,独立层)的最大数量,其可以与子区域的数量相同。

[0409] 在实施例中,图28示出了图27的扩展情况。当输入图片被划分成多个(例如,四个)子区域时,每个局部区域可以用一个或多个层来进行编解码。参考图28,独立层的数量可以等于子区域的数量,并且一个或多个层可以与子区域相对应。因此,每个子区域可以用一个或多个独立层以及零个或多个依赖层来进行编解码。

[0410] 在实施例中,在图28中,输入图片可以被划分成四个子区域。右上子区域可以被编解码为两层(例如,层1和层4),并且右下子区域可以被编解码为两层(例如,层3和层5)。在图28中,层4可以参考层1进行运动补偿预测,并且层5可以参考层3进行运动补偿。

[0411] 在实施例中,可以(可选地)禁用跨层边界的环路滤波(诸如去块滤波、自适应环路滤波、整形器、双边滤波或任何基于深度学习的滤波)。

[0412] 在实施例中,可以(可选地)禁用跨层边界的运动补偿预测或块内复制。

[0413] 在实施例中,用于运动补偿预测的边界填充或者子图片的边界处的环路滤波可以被选择性地处理。可以在诸如一个或多个参数集(VPS、SPS、PPS或APS)、切片或图块组头、或者SEI消息等高级别语法结构中,用信号通知用于指示是否处理边界填充的标志。

[0414] 在实施例中,可以在VPS或SPS中用信号通知一个或多个子区域(或一个或多个子图片)的布局信息。图29A至图29B示出了VPS和SPS中的语法元素的示例。参考图29A,在VPS中用信号通知标志(例如,vps_sub_picture_dividing_flag)。标志(例如,vps_sub_picture_dividing_flag)可以指示一个或多个输入图片是否被划分成多个子区域。当标志(例如,vps_sub_picture_dividing_flag)的值等于0时,与当前VPS相对应的一个或多个已编码视频序列中的一个或多个输入图片可以不被划分成多个子区域。因此,输入图片大小可以等于在SPS中用信号通知的已编码图片大小(例如,pic_width_in_luma_samples,pic_height_in_luma_samples)。当标志(例如,vps_sub_picture_dividing_flag)的值等于1时,一个或多个输入图片可以被划分成多个子区域。因此,在VPS中用信号通知语法元素vps_full_pic_width_in_luma_samples和vps_full_pic_height_in_luma_samples.vps_full_pic_width_in_luma_samples和vps_full_pic_height_in_luma_samples的值可以分别等于一个或多个输入图片的宽度和高度。

[0415] 在实施例中,vps_full_pic_width_in_luma_samples和vps_full_pic_height_in_luma_samples的值可以不被用于解码,而是可以被用于合成及显示。

[0416] 在实施例中,当标志(例如,vps_sub_picture_dividing_flag)的值等于1时,可以在与一个或多个特定层相对应的SPS中用信号通知语法元素pic_offset_x和pic_offset_y。因此,在SPS中用信号通知的已编码图片大小(例如,pic_width_in_luma_samples,pic_height_in_luma_samples)可以等于与特定层相对应的子区域的宽度和高度。可以在SPS中用信号通知子区域左上角的位置(例如,语法元素pic_offset_x和pic_offset_y)。

[0417] 在实施例中,子区域的左上角的位置信息(例如,语法元素pic_offset_x、pic_offset_y)可以不被用于解码,而是可以被用于合成及显示。

[0418] 在实施例中,可以在参数集或SEI消息中用信号通知一个或多个输入图片的全部或子集子区域(多个)的布局信息(例如,大小和位置)以及一个或多个层之间的依赖性信息。图30示出了用于指示子区域的布局、层之间的依赖性、以及子区域与一个或多个层之间的关系的信息的语法元素的示例。参考图30,语法元素num_sub_region可以指示当前已编码视频序列中的(矩形)子区域的数量。语法元素num_layers可以指示当前已编码视频序列中的层的数量。num_layers的值可以等于或大于num_sub_region的值。当任何子区域被编解码为单层时,num_layers的值可以等于num_sub_region的值。当一个或多个子区域被编解码为多个层时,num_layers的值可以大于num_sub_region的值。语法元素direct_dependency_flag[i][j]可以指示从第j层至第i层的依赖性。语法元素num_layers_for_region[i]可以指示与第i子区域相关联的层的数量。语法元素sub_region_layer_id[i][j]可以指示与第i子区域相关联的第j层的layer_id。sub_region_offset_x[i]和sub_region_offset_y[i]可以分别指示第i子区域的左上角的水平位置和垂直位置。sub_region_width[i]和sub_region_height[i]可以分别指示第i子区域的宽度和高度。

[0419] 在实施例中,可以在例如VPS、DPS、SPS、PPS、APS或SEI消息等高级别语法结构中用

信号通知一个或多个语法元素,该一个或多个语法元素用于指定输出层集以指示要输出的具有或不具有PTL信息的多个层中一个。

[0420] 参考图31,可以在VPS中用信号通知语法元素num_output_layer_sets,该语法元素num_output_layer_sets用于指示在参考VPS的已编码视频序列中的输出层集(OLS)的数量。对于每个输出层集,可以针对输出层中的每一个输出层来用信号通知标志(例如,output_layer_flag)。

[0421] 在实施例中,output_layer_flag[i]等于1可以指定第i层是输出。vps_output_layer_flag[i]等于0可以指定第i层不是输出。

[0422] 在实施例中,可以在例如VPS、DPS、SPS、PPS、APS或SEI消息等高级别语法结构中用信号通知一个或多个语法元素,该一个或多个语法元素用于指定针对每个输出层集的PTL信息。参考图31,可以在VPS中用信号通知语法元素num_profile_tile_level,该语法元素num_profile_tile_level用于指示在参考VPS的已编码视频序列中每个OLS的档位层级等级信息的数量。对于每个输出层集,可以针对每个输出层用信号通知用于PTL信息的语法元素的集合、或者用于指示PTL信息中的条目当中的特定PTL信息的索引。

[0423] 在实施例中,profile_tier_level_idx[i][j]指定在第i个OLS的第j层应用的profile_tier_level()语法结构在VPS中的profile_tier_level()语法结构的列表中的索引。

[0424] 在实施例中,参考图32,当最大层的数量大于1(vps_max_layers_minus1>0)时,可以用信号通知语法元素num_profile_tile_level和/或num_output_layer_sets。

[0425] 在实施例中,参考图32,用于指示针对第i个输出层集的输出层信令模式的语法元素vps_output_layers_mode[i]可以存在于VPS中。

[0426] 在实施例中,vps_output_layers_mode[i]等于0指定,仅最高层是具有第i个输出层集的输出。vps_output_layer_mode[i]等于1指定,所有层是具有第i个输出层集的输出。vps_output_layer_mode[i]等于2指定,作为输出的层是具有第i个输出层集的、vps_output_layer_flag[i][j]等于1的层。可以保留更多的值。

[0427] 在实施例中,取决于用于第i个输出层集的vps_output_layers_mode[i]的值,可以用信号通知或者可以不用信号通知output_layer_flag[i][j]。

[0428] 在实施例中,参考图32,标志vps_ptl_signal_flag[i]可以针对第i个输出层集而存在。取决于vps_ptl_signal_flag[i]的值,可以用信号通知或者可以不用信号通知用于第i个输出层集的PTL信息。

[0429] 在实施例中,参考图33,可以在例如VPS、DPS、SPS、PPS、APS或SEI消息等高级别语法结构中,用信号通知当前CVS中的子图片的数量(例如,max_subpics_minus1)。

[0430] 在实施例中,参考图33,当子图片的数量大于1(max_subpics_minus1>0)时,可以用信号通知针对第i子图片的子图片标识符(例如,sub_pic_id[i])。

[0431] 在实施例中,可以在VPS中用信号通知用于指示属于每个输出层集中的每个层的子图标识符的一个或多个语法元素。参考图34,sub_pic_id_layer[i][j][k]指示第i个输出层集的第j层中存在的第k子图片。通过以上信息,解码器可以识别出针对特定输出层集的每一层可以解码并输出哪个子图片。

[0432] 在实施例中,图片头(PH)是包含在已编码图片的所有切片中应用的语法元素的语

法结构。切片头可以是已编码切片的一部分,该已编码切片包含与在该切片中表示的图块内的所有图块或CTU行有关的数据元素。图片单元(PU)是NAL单元的集合,NAL单元根据指定的分类规则而相互关联,在解码顺序上是连续的,并且例如恰好包含一个已编码图片。PU可以包含图片头(PH)、以及构成已编码图片的一个或多个VCL NAL单元。

[0433] 在实施例中,SPS(RBSP)可以在该SPS被参考之前被用于解码过程,被包括在TemporalId等于0的至少一个AU中,或者通过外部手段来提供。

[0434] 在实施例中,SPS(RBSP)可以在该SPS被参考之前被用于解码过程,被包括在CVS(其包含参考SPS的一个或多个PPS)中TemporalId等于0的至少一个AU中,或者通过外部手段来提供。

[0435] 在实施例中,SPS(RBSP)可以在该SPS被一个或多个PPS参考之前被用于解码过程,被包括在CVS(其包含参考SPS的一个或多个PPS)中nuh_layer_id等于参考SPS NAL单元的PPS NAL单元的最低nuh_layer_id值的至少一个PU中,或者通过外部手段来提供。

[0436] 在实施例中,SPS(RBSP)可以在该SPS被一个或多个PPS参考之前被用于解码过程,被包括在TemporalId等于0并且nuh_layer_id等于参考SPS NAL单元的PPS NAL单元的最低nuh_layer_id值的至少一个PU中,或者通过外部手段来提供。

[0437] 在实施例中,SPS(RBSP)可以在该SPS被一个或多个PPS参考之前被用于解码过程,被包括在CVS(其包含参考SPS的一个或多个PPS)中TemporalId等于0并且nuh_layer_id等于参考SPS NAL单元的PPS NAL单元的最低nuh_layer_id值的至少一个PU中,或者通过外部手段来提供或通过外部手段提供。

[0438] 在实施例中,pps_seq_parameter_set_id指定针对所参考的SPS的sps_seq_parameter_set_id的值。在示例中,pps_seq_parameter_set_id的值在由CLVS中的已编码图片所参考的所有PPS中可以是相同的。

[0439] 在实施例中,CVS中具有sps_seq_parameter_set_id的特定值的所有SPS NAL单元可以具有相同的内容。

[0440] 在实施例中,无论nuh_layer_id值如何,SPS NAL单元都可以共享sps_seq_parameter_set_id的相同值空间。

[0441] 在实施例中,SPS NAL单元的nuh_layer_id值可以等于参考SPS NAL单元的PPS NAL单元的最低nuh_layer_id值。

[0442] 在实施例中,当nuh_layer_id等于m的SPS被nuh_layer_id等于n的一个或多个PPS参考时,nuh_layer_id等于m的层可以与nuh_layer_id等于n的层或nuh_layer_id等于m的层的(直接或间接)参考层相同。

[0443] 在实施例中,PPS(RBSP)可以在该PPS被参考之前被用于解码过程,被包括在TemporalId等于PPS NAL单元的TemporalId的至少一个AU中,或者通过外部手段来提供。

[0444] 在实施例中,PPS(RBSP)可以在该PPS被参考之前被用于解码过程,被包括在CVS(其包含参考PPS的一个或多个PH(或已编码切片NAL单元))中TemporalId等于PPS NAL单元的TemporalId的至少一个AU中,或者通过外部手段来提供。

[0445] 在实施例中,PPS(RBSP)可以在该PPS被一个或多个PH(或已编码切片NAL单元)参考之前被用于解码过程,被包括在CVS(其包含参考PPS的一个或多个PH(或已编码切片NAL单元))中nuh_layer_id等于参考PPS NAL单元的已编码切片NAL单元的最低nuh_layer_id

值的至少一个PU中,或者通过外部手段来提供。

[0446] 在实施例中,PPS(RBSP)可以在该PPS被一个或多个PH(或已编码切片NAL单元)参考之前被用于解码过程,被包括在CVS(其包含参考PPS的一个或多个PH(或已编码切片NAL单元))中TemporalId等于PPS NAL单元的TemporalId并且nuh_layer_id等于参考PPS NAL单元的已编码切片NAL单元的最低nuh_layer_id值的至少一个PU中,或者通过外部手段来提供。

[0447] 在实施例中,PH中的ph_pic_parameter_set_id指定,用于使用中所参考的PPS的pps_pic_parameter_set_id的值。pps_seq_parameter_set_id的值在由CLVS中的已编码图片所参考的所有PPS中可以是相同的。

[0448] 在实施例中,PU内具有pps_pic_parameter_set_id的特定值的所有PPS NAL单元可以具有相同的内容。

[0449] 在实施例中,无论nuh_layer_id值如何,PPS NAL单元都可以共享pps_pic_parameter_set_id的相同值空间。

[0450] 在实施例中,PPS NAL单元的nuh_layer_id值可以等于参考NAL单元的已编码切片NAL单元的最低nuh_layer_id值。NAL单元指的是PPS NAL单元。

[0451] 在实施例中,当nuh_layer_id等于m的PPS被nuh_layer_id等于n的一个或多个已编码切片NAL单元参考时,nuh_layer_id等于m的层可以与nuh_layer_id等于n的层或nuh_layer_id等于m的层的(直接或间接)参考层相同。

[0452] 在实施例中,PPS(RBSP)可以在该PPS被参考之前被用于解码过程,被包括在TemporalId等于PPS NAL单元的TemporalId的至少一个AU中,或者通过外部手段来提供。

[0453] 在实施例中,PPS(RBSP)可以在该PPS被参考之前被用于解码过程,被包括在CVS(其包含参考PPS的一个或多个PH(或已编码切片NAL单元))中TemporalId等于PPS NAL单元的TemporalId的至少一个AU中,或者通过外部手段来提供。

[0454] 在实施例中,PPS(RBSP)可以在该PPS被一个或多个PH(或已编码切片NAL单元)参考之前被用于解码过程,被包括在CVS(其包含参考PPS的一个或多个PH(或已编码切片NAL单元))中nuh_layer_id等于参考PPS NAL单元的已编码切片NAL单元的最低nuh_layer_id值的至少一个PU中,或者通过外部手段来提供。

[0455] 在实施例中,PPS(RBSP)可以在该PPS被一个或多个PH(或已编码切片NAL单元)参考之前被用于解码过程,被包括在CVS(其包含参考PPS的一个或多个PH(或已编码切片NAL单元))中TemporalId等于PPS NAL单元的TemporalId并且nuh_layer_id等于参考PPS NAL单元的已编码切片NAL单元的最低nuh_layer_id值的至少一个PU中,或者通过外部手段来提供。

[0456] 在实施例中,PH中的ph_pic_parameter_set_id指定,用于使用中所参考的PPS的pps_pic_parameter_set_id的值。pps_seq_parameter_set_id的值在由CLVS中的已编码图片参考的所有PPS中可以是相同的。

[0457] 在实施例中,PU内具有pps_pic_parameter_set_id的特定值的所有PPS NAL单元可以具有相同的内容。

[0458] 在实施例中,无论nuh_layer_id值如何,PPS NAL单元都可以共享pps_pic_parameter_set_id的相同值空间。

[0459] 在实施例中,PPS NAL单元的nuh_layer_id值可以等于参考参考PPS NAL单元的NAL单元的已编码切片NAL单元的最低nuh_layer_id值。

[0460] 在实施例中,当nuh_layer_id等于m的PPS被nuh_layer_id等于n的一个或多个已编码切片NAL单元参考时,nuh_layer_id等于m的层可以与nuh_layer_id等于n的层或nuh_layer_id等于m的层的(直接或间接)参考层相同。

[0461] 输出层指示作为输出的输出层集中的层。输出层集(OLS)指示层的集合,该层的集合包括层的指定集合,其中,层的集合中的一个或多个层被指定为输出层。OLS层索引是OLS中的层的索引。OLS层索引可以指向OLS中的层的列表。

[0462] 子层指示包括具有TemporalId变量的特定值的VCL NAL单元以及相关联的非VCL NAL单元的时间可伸缩比特流的时间可伸缩层。子层表达用于指示包括特定子层以及较低子层的NAL单元的比特流的子集。

[0463] VPS RBSP可以在该VPS RBSP被参考之前被用于解码过程,被包括在TemporalId等于0的至少一个AU中,或者通过外部手段来提供。CVS中具有vps_video_parameter_set_id的特定值的所有VPS NAL单元可以具有相同的内容。

[0464] vps_video_parameter_set_id提供针对VPS的标识符,以供其他的语法元素参考。vps_video_parameter_set_id的值可以大于0。

[0465] vps_max_layers_minus1加1指定,参考VPS的每个CVS中的最大允许层数。

[0466] vps_max_sublayer_minus1加1指定,在参考VPS的每个CVS中的层中可以存在的时间子层的最大数量。vps_max_sublayer_minus1的值可以在0至6(包含端值)的范围内。

[0467] 在示例中,标志(例如,vps_all_layers_same_num_sublayers_flag)等于1指定,时间子层的数量对于参考VPS的每个CVS中的所有层都是相同的。标志(例如,vps_all_layers_same_num_sublayers_flag)等于0指定,参考VPS的每个CVS中的层可以具有或者可以不具有相同数量的时间子层。当不存在时,标志(例如,vps_all_layers_same_num_sublayers_flag)的值被推断为等于1。

[0468] 标志(例如,vps_all_independent_layers_flag)等于1指定,在不使用层间预测的情况下独立地对CVS中的所有层进行编解码。标志(例如,vps_all_independent_layers_flag)等于0指定,CVS中的层中一个或多个层可以使用层间预测。当不存在时,标志(例如,vps_all_independent_layers_flag)的值被推断为等于1。

[0469] vps_layer_id[i]可以指定第i层的nuh_layer_id值。对于m和n任意两个非负整数值,当m小于n时,vps_layer_id[m]的值可以小于vps_layer_id[n]。

[0470] vps_independent_layer_flag[i]等于1可以指定,具有索引i的层不使用层间预测。vps_independent_layer_flag[i]等于0可以指定,具有索引i的层可以使用层间预测,并且在VPS中存在针对0至i-1(包含端值)范围内的j的语法元素vps_direct_ref_layer_flag[i][j]。当不存在时,vps_independent_layer_flag[i]的值被推断为等于1。

[0471] vps_direct_ref_layer_flag[i][j]等于0可以指定,具有索引j的层不是针对具有索引i的层的直接参考层。vps_direct_ref_layer_flag[i][j]等于1可以指定,具有索引j的层是针对具有索引i的层的直接参考层。当vps_direct_ref_layer_flag[i][j]对于i和j在0至vps_max_layers_minus1(包含端值)的范围内不存在时,其被推断为等于0。当vps_independent_layer_flag[i]等于0时,j的至少一个值可以在0至i-1(包含端值)的范围内,

使得vps_direct_ref_layer_flag[i][j]的值等于1。

[0472] 变量NumDirectRefLayers[i]、DirectRefLayerIdx[i][d]、NumRefLayers[i]、RefLayerIdx[i][r]和LayerUsedAsRefLayerFlag[j]可以被如下导出:

[0473] for(i=0;i<=vps_max_layers_minus1;i++){for(j=0;j<=vps_max_layers_minus1;j++){

[0474] dependencyFlag[i][j]=vps_direct_ref_layer_flag[i][j]for(k=0;k<i;k++)

[0475] if(vps_direct_ref_layer_flag[i][k]&&dependencyFlag[k][j])

[0476] dependencyFlag[i][j]=1

[0477] }

[0478] LayerUsedAsRefLayerFlag[i]=0

[0479] }

[0480] for(i=0;i<=vps_max_layers_minus1;i++){

[0481] for(j=0,d=0,r=0;j<=vps_max_layers_minus1;j++){if(vps_direct_ref_layer_flag[i][j]){

[0482] DirectRefLayerIdx[i][d++]=j LayerUsedAsRefLayerFlag[j]=1

[0483] }

[0484] if(dependencyFlag[i][j])RefLayerIdx[i][r++]=j

[0485] }

[0486] NumDirectRefLayers[i]=d NumRefLayers[i]=r

[0487] }

[0488] 用于对nuh_layer_id等于vps_layer_id[i]的层的层索引进行指定的变量GeneralLayerIdx[i]可以被如下导出:

[0489] for(i=0;i<=vps_max_layers_minus1;i++)

[0490] GeneralLayerIdx[vps_layer_id[i]]=i

[0491] 在示例中,对于i和j任意两个不同的值,当标志(例如,dependencyFlag[i][j])等于1时,两者均在0至vps_max_layers_minus1(包含端值)的范围内。比特流一致性的要求可以是,应用于第i层的chroma_format_idc和bit_depth_minus8的值可以分别等于应用于第j层的chroma_format_idc和bit_depth_minus8的值。

[0492] max_tid_ref_present_flag[i]等于1可以指定,存在语法元素max_tid_il_ref_pics_plus1[i]。max_tid_ref_present_flag[i]等于0可以指定,不存在语法元素max_tid_il_ref_pics_plus1[i]。

[0493] max_tid_il_ref_pics_plus1[i]等于0可以指定,第i层的非IRAP图片不使用层间预测。max_tid_il_ref_pics_plus1[i]大于0可以指定,对于第i层的解码图片,没有TemporalId大于max_tid_il_ref_pics_plus1[i]-1的图片被用作ILRP。当不存在时,max_tid_il_ref_pics_plus1[i]的值被推断为等于7。

[0494] each_layer_is_an_ols_flag等于1可以指定,每个OLS仅包含一个层,并且在参考VPS的CVS中的每个层本身是OLS,其中单个包括的层是唯一的输出层。each_layer_is_an_ols_flag等于0可以指定,OLS可以包含一个以上的层。如果vps_max_layers_minus1等于0,

则each_layer_is_an_ols_flag的值被推断为等于1。否则,当vps_all_independent_layers_flag等于0时,each_layer_is_an_ols_flag的值被推断为等于0。

[0495] OLS_mode_idc等于0可以指定,由VPS指定的OLS的总数量等于vps_max_layers_minus1+1,第i个OLS包括层索引从0至i(包含端值)的层,并且对于每个OLS,仅OLS中的最高层是输出。

[0496] ols_mode_idc等于1可以指定,由VPS指定的OLS的总数量等于vps_max_layers_minus1+1,第i个OLS包括层索引从0至i(包含端值)的层。对于每个OLS,OLS中的所有层都是输出。

[0497] OLS_mode_idc等于2可以指定,由VPS指定的OLS的总数量被显式地用信号通知,并且对于每个OLS,输出层被显式地用信号通知,并且其他层是作为OLS的输出层的直接或间接参考层的层。

[0498] ols_mode_idc的值可以在0至2(包含端值)的范围内。ols_mode_idc的值3可以例如由ITU-T或ISO/IEC为某些视频编解码技术或标准而保留。

[0499] 当标志(例如,vps_all_independent_layers_flag)等于1并且标志(例如,each_layer_is_an_ols_flag)等于0时,ols_mode_idc的值被推断为等于2。

[0500] num_output_layer_sets_minus1加1可以指定,ols_mode_idc等于2时由VPS指定的OLS的总数量。

[0501] 用于对由VPS指定的OLS的总数量进行指定的变量TotalNumOlss可以被如下导出:

[0502] if (vps_max_layers_minus1 == 0)

[0503] TotalNumOlss = 1

[0504] else if (each_layer_is_an_ols_flag & !ols_mode_idc == 0 & !ols_mode_idc == 1)

[0505] TotalNumOlss = vps_max_layers_minus1 + 1 else if (ols_mode_idc == 2)

[0506] TotalNumOlss = num_output_layer_sets_minus1 + 1

[0507] ols_output_layer_flag[i][j]等于1可以指定,当ols_mode_idc等于2时,nuh_layer_id等于vps_layer_id[j]的层是第i个OLS的输出层。ols_output_layer_flag[i][j]等于0指定,当ols_mode_idc等于2时,nuh_layer_id等于vps_layer_id[j]的层不是第i个OLS的输出层。

[0508] 用于对第i个OLS中的输出层的数量进行指定的变量NumOutputLayersInOls[i]、用于对第i个OLS中的第j层中的子层的数量进行指定的变量NumSubLayersInLayerInOLS[i][j]、用于对第i个OLS中的第j输出层的nuh_layer_id值进行指定的变量OutputLayerIdInOls[i][j]、以及用于对第k层是否被用作至少一个OLS中的输出层进行指定的变量LayerUsedAsOutputLayerFlag[k]可以被如下导出:

[0509] NumOutputLayersInOls[0] = 1

[0510] OutputLayerIdInOls[0][0] = vps_layer_id[0]

[0511] NumSubLayersInLayerInOLS[0][0] = vps_max_sub_layers_minus1 + 1

[0512] LayerUsedAsOutputLayerFlag[0] = 1

[0513] for (i = 1, i <= vps_max_layers_minus1; i++) {

[0514] if (each_layer_is_an_ols_flag & !ols_mode_idc < 2)


```

[0515] LayerUsedAsOutputLayerFlag[i]=1
[0516] else/*(!each_layer_is_an_ols_flag&&ols_mode_idc==2)*/
LayerUsedAsOutputLayerFlag[i]=0
[0517] }
[0518] for(i=1;i<TotalNumOls;i++)
[0519] if(each_layer_is_an_ols_flag 1 1ols_mode_idc==0)
{NumOutputLayersInOls[i]=1OutputLayerIdxInOls[i][0]=vps_layer_id[i]
[0520] for(j=0;j<i&&(ols_mode_idc==0);j++)NumSubLayersInLayerInOLS[i][j]
=max_tid_il_ref_pics_plus1[i]
[0521] NumSubLayersInLayerInOLS[i][i]=vps_max_sub_layers_minus1+1
[0522] }else if(ols_mode_idc==1){
[0523] NumOutputLayersInOls[i]=i+1
[0524] for(j=0;j<NumOutputLayersInOls[i];j++){OutputLayerIdxInOls[i][j]=
vps_layer_id[j]
[0525] NumSubLayersInLayerInOLS[i][j]=vps_max_sub_layers_minus1+1
[0526] }
[0527] }else if(ols_mode_idc==2){
[0528] for(j=0;j<=vps_max_layers_minus1;j++){layerIncludedInOlsFlag[i][j]
=0NumSubLayersInLayerInOLS[i][j]=0
[0529] }
[0530] for(k=0,j=0;k<=vps_max_layers_minus1;k++)if(ols_output_layer_flag
[i][k]){
[0531] layerIncludedInOlsFlag[i][k]=1LayerUsedAsOutputLayerFlag[k]=
1OutputLayerIdx[i][j]=k
[0532] OutputLayerIdxInOls[i][j++]=vps_layer_id[k]NumSubLayersInLayerInOLS
[i][j]=vps_max_sub_layers_minus1+1
[0533] }
[0534] NumOutputLayersInOls[i]=j
[0535] for(j=0;j<NumOutputLayersInOls[i];j++){idx=OutputLayerIdx[i][j]
[0536] for(k=0;k<NumRefLayers[idx];k++){layerIncludedInOlsFlag[i]
[RefLayerIdx[idx][k]]=1
[0537] if(NumSubLayersInLayerInOLS[i][RefLayerIdx[idx][k]]<max_tid_il_ref_
pics_plus1[OutputLayerIdxInOls[i][j]])NumSubLayersInLayerInOLS[i][RefLayerIdx
[idx][k]]=
[0538] max_tid_il_ref_pics_plus1[OutputLayerIdxInOls[i][j]]
[0539] }
[0540] }
[0541] }
[0542] 对于在0至vps_max_layers_minus1(包含端值)范围内的i的每个值,

```

LayerUsedAsRefLayerFlag[i]和LayerUsedAsOutputLayerFlag[i]的值可以不都等于0。因此,可能不会存在既不是至少一个OLS的输出层也不是任何其他层的直接参考层的层。

[0543] 对于每个OLS,可以有至少一个层是输出层。因此,对于在0至TotalNumOls-1(包含端值)范围内的i的任何值,NumOutputLayersInOls[i]的值可以大于或等于1。

[0544] 用于对第i个OLS中的层数进行指定的变量NumLayersInOls[i]和用于对第i个OLS中的第j层的nuh_layer_id值进行指定的变量LayerIdInOls[i][j]被如下导出:

```
[0545] NumLayersInOls[0]=1
[0546] LayerIdInOls[0][0]=vps_layer_id[0]for(i=1;i<TotalNumOls;i++){
[0547] if(each_layer_is_an_ols_flag){NumLayersInOls[i]=1
[0548] LayerIdInOls[i][0]=vps_layer_id[i]
[0549] }else if(ols_mode_idc==0 1 ols_mode_idc==1){
[0550] NumLayersInOls[i]=i+1
[0551] for(j=0;j<NumLayersInOls[i];j++)LayerIdInOls[i][j]=vps_layer_id[j]
[0552] }else if(ols_mode_idc==2){
[0553] for(k=0,j=0;k<=vps_max_layers_minus1;k++)if(layerIncludedInOlsFlag
[i][k])
[0554] LayerIdInOls[i][j++]=vps_layer_id[k]NumLayersInOls[i]=j
[0555] }
[0556] }
```

[0557] 用于对nuh_layer_id等于LayerIdInOls[i][j]的层的OLS层索引进行指定的变量OlsLayerIdx[i][j]可以被如下导出:

```
[0558] for(i=0;i<TotalNumOls;i++)
[0559] for j=0;j<NumLayersInOls[i];j++)
[0560] OlsLayerIdx[i][LayerIdInOls[i][j]]=j
```

[0561] 每个OLS中的最低层可以是独立层。因此,对于在0至TotalNumOls-1(包含端值)范围内的每个i,vps_independent_layer_flag[GeneralLayerIdx[LayerIdInOls[i][0]]]的值可以等于1。

[0562] 每个层可以被包括在由VPS指定的至少一个OLS中。因此,对于nuh_layer_id的特定值nuhLayerId等于vps_layer_id[k](其中,k在0至vps_max_layers_minus1(包含端值)的范围内)中的一个的每个层,可以存在至少一对i和j的值,其中,i在0至TotalNumOls-1(包含端值)范围内,并且j在NumLayersInOls[i]-1(包含端值)范围内,使得LayerIdInOls[i][j]的值等于nuhLayerId。

[0563] 在实施例中,对于当前图片CurrPic,解码过程如下操作:

[0564] PictureOutputFlag被如下设置:

[0565] 如果以下条件之一为真,则PictureOutputFlag被设置为等于0:

[0566] -当前图片是RASL图片,并且相关联的1RAP图片的NoOutputBeforeRecoveryFlag等于1。

[0567] gdr_enabled_flag等于1,并且当前图片是NoOutputBeforeRecoveryFlag等于1的GDR图片。

[0568] gdr_enabled_flag等于1,当前图片与NoOutputBeforeRecoveryFlag等于1的GDR图片相关联,并且当前图片的PicOrderCntVal小于相关联的GDR图片的RpPicOrderCntVal。

[0569] sps_video_parameter_set_id大于0,ols_mode_idc等于0并且当前AU包含满足所有以下条件的图片picA:

[0570] PicA具有等于1的PictureOutputFlag。

[0571] PicA具有大于当前图片的nuh_layer_id nuhLid的nuh_layer_id nuhLid。

[0572] PicA属于OLS的输出层(即,OutputLayerIdInOls[TargetOlsIdx][0]等于nuhLid)。

[0573] sps_video_parameter_set_id大于0,ols_mode_idc等于2,并且

[0574] ols_output_layer_flag[TargetOlsIdx][GeneralLayerIdx[nuh_layer_id]]等于0。

[0575] 否则,PictureOutputFlag被设置为等于pic_output_flag。

[0576] 在当前图片的所有切片被解码之后,当前已解码图片被标记为“用于短期参考”,并且RefPicList[0]或RefPicList[1]中的每个ILRP条目被标记为“用于短期参考”。

[0577] 在实施例中,当每层都是输出层集时,无论ols_mode_idc的值如何,PictureOutputFlag都被设置为等于pic_output_flag。

[0578] 在实施例中,当sps_video_parameter_set_id大于0,each_layer_is_an_ols_flag等于0,ols_mode_idc等于0并且当前AU包含满足所有以下条件的图片picA时,PictureOutputFlag被设置为等于0:PicA具有等于1的PictureOutputFlag,PicA具有大于当前图片的nuh_layer_id nuhLid的nuh_layer_id nuhLid,并且PicA属于OLS的输出层(即,OutputLayerIdInOls[TargetOlsIdx][0]等于nuhLid)。

[0579] 在实施例中,当sps_video_parameter_set_id大于0,each_layer_is_an_ols_flag等于0,ols_mode_idc等于2,并且ols_output_layer_flag[TargetOlsIdx][GeneralLayerIdx[nuh_layer_id]]等于0时,PictureOutputFlag被设置为等于0。

[0580] 图35示出了VPS RBSP的语法结构的示例。图35中所示的语法结构与图34中所示的语法结构相似,因此为了简洁起见,省略了详细描述。

[0581] 上述技术(例如,用于信令约束标志、自适应分辨率参数等的技术)可以通过计算机可读指令实现为计算机软件,并且物理地存储在一个或多个计算机可读介质中。例如,图36示出了计算机系统(3600),其适于实现所公开主题的某些实施例。

[0582] 所述计算机软件可通过任何合适的机器代码或计算机语言进行编码,通过汇编、编译、链接等机制创建包括指令的代码,所述指令可由一个或多个计算机中央处理单元(CPU),图形处理单元(GPU)等直接执行或通过译码、微代码等方式执行。

[0583] 所述指令可以在各种类型的计算机或其组件上执行,包括例如个人计算机、平板电脑、服务器、智能手机、游戏设备、物联网设备等。

[0584] 图36所示的用于计算机系统(3600)的组件本质上是示例性的,并不用于对实现本公开实施例的计算机软件的使用范围或功能进行任何限制。也不应将组件的配置解释为与计算机系统(3600)的示例性实施例中所示的任一组件或其组合具有任何依赖性要求。

[0585] 计算机系统(3600)可以包括某些人机界面输入设备。这种人机界面输入设备可以通过触觉输入(如:键盘输入、滑动、数据手套移动)、音频输入(如:声音、掌声)、视觉输入

(如:手势)、嗅觉输入(未示出),对一个或多个人类用户的输入做出响应。所述人机界面设备还可用于捕获某些媒体,气与人类有意识的输入不必直接相关,如音频(例如:语音、音乐、环境声音)、图像(例如:扫描图像、从静止影像相机获得的摄影图像)、视频(例如二维视频、包括立体视频的三维视频)。

[0586] 人机界面输入设备可包括以下中的一个或多个(仅绘出其中一个):键盘(3601)、鼠标(3602)、触控板(3603)、触摸屏(3610)、数据手套(未示出)、操纵杆(3605)、麦克风(3606)、扫描仪(3607)、照相机(3608)。

[0587] 计算机系统(3600)还可以包括某些人机界面输出设备。这种人机界面输出设备可以通过例如触觉输出、声音、光和嗅觉/味觉来刺激一个或多个人类用户的感觉。这样的人机界面输出设备可包括触觉输出设备(例如通过触摸屏(3610)、数据手套(未示出)或操纵杆(3605)的触觉反馈,但也可以有不用作输入设备的触觉反馈设备)、音频输出设备(例如,扬声器(3609)、耳机(未示出))、视觉输出设备(例如,包括阴极射线管屏幕、液晶屏幕、等离子屏幕、有机发光二极管屏的屏幕(3610),其中每一个都具有或没有触摸屏输入功能、每一个都具有或没有触觉反馈功能——其中一些可通过诸如立体画面输出的手段输出二维视觉输出或三维以上的输出;虚拟现实眼镜(未示出)、全息显示器和放烟箱(未示出))以及打印机(未示出)。

[0588] 计算机系统(3600)还可以包括人可访问的存储设备及其相关介质,如包括具有CD/DVD的高密度只读/可重写式光盘(CD/DVDROM/RW)(3620)或类似介质(3621)的光学介质、拇指驱动器(3622)、可移动硬盘驱动器或固体状态驱动器(3623),诸如磁带和软盘(未示出)的传统磁介质,诸如安全软件保护器(未示出)等的基于ROM/ASIC/PLD的专用设备,等等。

[0589] 本领域技术人员还应当理解,结合所公开的主题使用的术语“计算机可读介质”不包括传输介质、载波或其它瞬时信号。

[0590] 计算机系统(3600)还可以包括通往一个或多个通信网络(3655)的接口(3654)。例如,网络可以是无线的、有线的、光学的。网络还可为局域网、广域网、城域网、车载网络和工业网络、实时网络、延迟容忍网络等等。网络还包括以太网、无线局域网、蜂窝网络(GSM、3G、4G、5G、LTE等)等局域网、电视有线或无线广域数字网络(包括有线电视、卫星电视、和地面广播电视)、车载和工业网络(包括CANBus)等等。某些网络通常需要外部网络接口适配器,用于连接到某些通用数据端口或外围总线(3649)(例如,计算机系统(3600)的USB端口);其它系统通常通过连接到如下所述的系统总线集成到计算机系统(3600)的核心(例如,以太网接口集成到PC计算机系统或蜂窝网络接口集成到智能电话计算机系统)。通过使用这些网络中的任何一个,计算机系统(3600)可以与其它实体进行通信。所述通信可以是单向的,仅用于接收(例如,无线电视),单向的仅用于发送(例如CAN总线到某些CAN总线设备),或双向的,例如通过局域或广域数字网络到其它计算机系统。上述的每个网络和网络接口可使用某些协议和协议栈。

[0591] 上述的人机界面设备、人可访问的存储设备以及网络接口可以连接到计算机系统(3600)的核心(3640)。

[0592] 核心(3640)可包括一个或多个中央处理单元(CPU)(3641)、图形处理单元(GPU)(3642)、以现场可编程门阵列(FPGA)(3643)形式的专用可编程处理单元、用于特定任务的

硬件加速器 (3644)、图形适配器 (~ ~ 50) 等。这些设备以及只读存储器 (ROM) (3645)、随机存取存储器 (3646)、内部大容量存储器 (例如内部非用户可存取硬盘驱动器、固态硬盘等) (3647) 等可通过系统总线 (3648) 进行连接。在某些计算机系统中,可以以一个或多个物理插头的形式访问系统总线 (3648),以便可通过额外的中央处理单元、图形处理单元等进行扩展。外围装置可直接附接到核心的系统总线 (3648),或通过外围总线 (3649) 进行连接。在示例中,屏幕 (3610) 可以连接到图形适配器 (3650)。外围总线的体系结构包括外部外围组件互联PCI、通用串行总线USB等。

[0593] CPU (3641)、GPU (3642)、FPGA (3643) 和加速器 (3644) 可以执行某些指令,这些指令组合起来可以构成上述计算机代码。该计算机代码可以存储在ROM (3645) 或RAM (3646) 中。过渡数据也可以存储在RAM (3646) 中,而永久数据可以存储在例如内部大容量存储器 (3647) 中。通过使用高速缓冲存储器可实现对任何存储器设备的快速存储和检索,高速缓冲存储器可与一个或多个CPU (3641)、GPU (3642)、大容量存储器 (3647)、ROM (3645)、RAM (3646) 等紧密关联。

[0594] 所述计算机可读介质上可具有计算机代码,用于执行各种计算机实现的操作。介质和计算机代码可以是为本公开的目的而特别设计和构造的,也可以是计算机软件领域的技术人员所熟知和可用的介质和代码。

[0595] 作为实施例而非限制,具有体系结构 (3600) 的计算机系统,特别是核心 (3640),可以作为处理器 (包括CPU、GPU、FPGA、加速器等) 提供执行包含在一个或多个有形的计算机可读介质中的软件的功能。这种计算机可读介质可以是与上述的用户可访问的大容量存储器相关联的介质,以及具有非易失性的核心 (3640) 的特定存储器,例如核心内部大容量存储器 (3647) 或ROM (3645)。实现本公开的各种实施例的软件可以存储在这种设备中并且由核心 (3640) 执行。根据特定需要,计算机可读介质可包括一个或一个以上存储设备或芯片。该软件可以使得核心 (3640) 特别是其中的处理器 (包括CPU、GPU、FPGA等) 执行本文所述的特定过程或特定过程的特定部分,包括定义存储在RAM (3646) 中的数据结构以及根据软件定义的过程来修改这种数据结构。另外或作为替代,计算机系统可以提供逻辑硬连线或以其它方式包含在电路 (例如,加速器 (3644)) 中的功能,该电路可以代替软件或与软件一起运行以执行本文所述的特定过程或特定过程的特定部分。在适当的情况下,对软件的引用可以包括逻辑,反之亦然。在适当的情况下,对计算机可读介质的引用可包括存储执行软件的电路 (如集成电路 (IC)), 包含执行逻辑的电路,或两者兼备。本公开包括任何合适的硬件和软件组合。

[0596] 附录A:首字母缩略词

[0597] JEM:联合开发模型 (jointexplorationmodel)

[0598] VVC:通用视频编解码 (versatilevideocoding)

[0599] BMS:基准集合 (benchmarkset)

[0600] MV:运动矢量 (MotionVector)

[0601] HEVC:高效视频编解码 (HighEfficiencyVideoCoding)

[0602] SEI:辅助增强信息 (SupplementaryEnhancementInformation)

[0603] VUI:视频可用性信息 (VideoUsabilityInformation)

[0604] GOPs:图片群组 (GroupsofPictures)

- [0605] TUs:变换单元(TransformUnits)
- [0606] PUs:预测单元(PredictionUnits)
- [0607] CTUs:编码树单元(CodingTreeUnits)
- [0608] CTBs:编码树块(CodingTreeBlocks)
- [0609] PBs:预测块(PredictionBlocks)
- [0610] HRD:假想参考解码器(HypotheticalReferenceDecoder)
- [0611] SNR:信噪比(SignalNoiseRatio)
- [0612] CPUs:中央处理单元(CentralProcessingUnits)
- [0613] GPUs:图形处理单元(GraphicsProcessingUnits)
- [0614] CRT:阴极射线管(CathodeRayTube)
- [0615] LCD:液晶显示器(Liquid-CrystalDisplay)
- [0616] OLED:有机发光二极管(OrganicLight-EmittingDiode)
- [0617] CD:光盘(CompactDisc)
- [0618] DVD:数字化视频光盘(DigitalVideoDisc)
- [0619] ROM:只读存储器(Read-OnlyMemory)
- [0620] RAM:随机存取存储器(RandomAccessMemory)
- [0621] ASIC:专用集成电路(Application-SpecificIntegratedCircuit)
- [0622] PLD:可编程逻辑设备(ProgrammableLogicDevice)
- [0623] LAN:局域网(LocalAreaNetwork)
- [0624] GSM:全球移动通信系统(GlobalSystemforMobilecommunications)
- [0625] LTE:长期演进(Long-TermEvolution)
- [0626] CANBus:控制器局域网络总线(ControllerAreaNetworkBus)
- [0627] USB:通用串行总线(UniversalSerialBus)
- [0628] PCI:外围组件互连(PeripheralComponentInterconnect)
- [0629] FPGA:现场可编程门阵列(FieldProgrammableGateAreas)
- [0630] SSD:固态驱动器(Solid-stateDrive)
- [0631] IC:集成电路(IntegratedCircuit)
- [0632] CU:编码单元(CodingUnit)

[0633] 虽然本公开已对多个示例性实施例进行了描述,但实施例的各种变更、排列和各种等同替换均属于本公开的范围内。因此应理解,本领域技术人员能够设计多种系统和方法,所述系统和方法虽然未在本文中明确示出或描述,但其体现了本公开的原则,因此属于本公开的精神和范围之内。

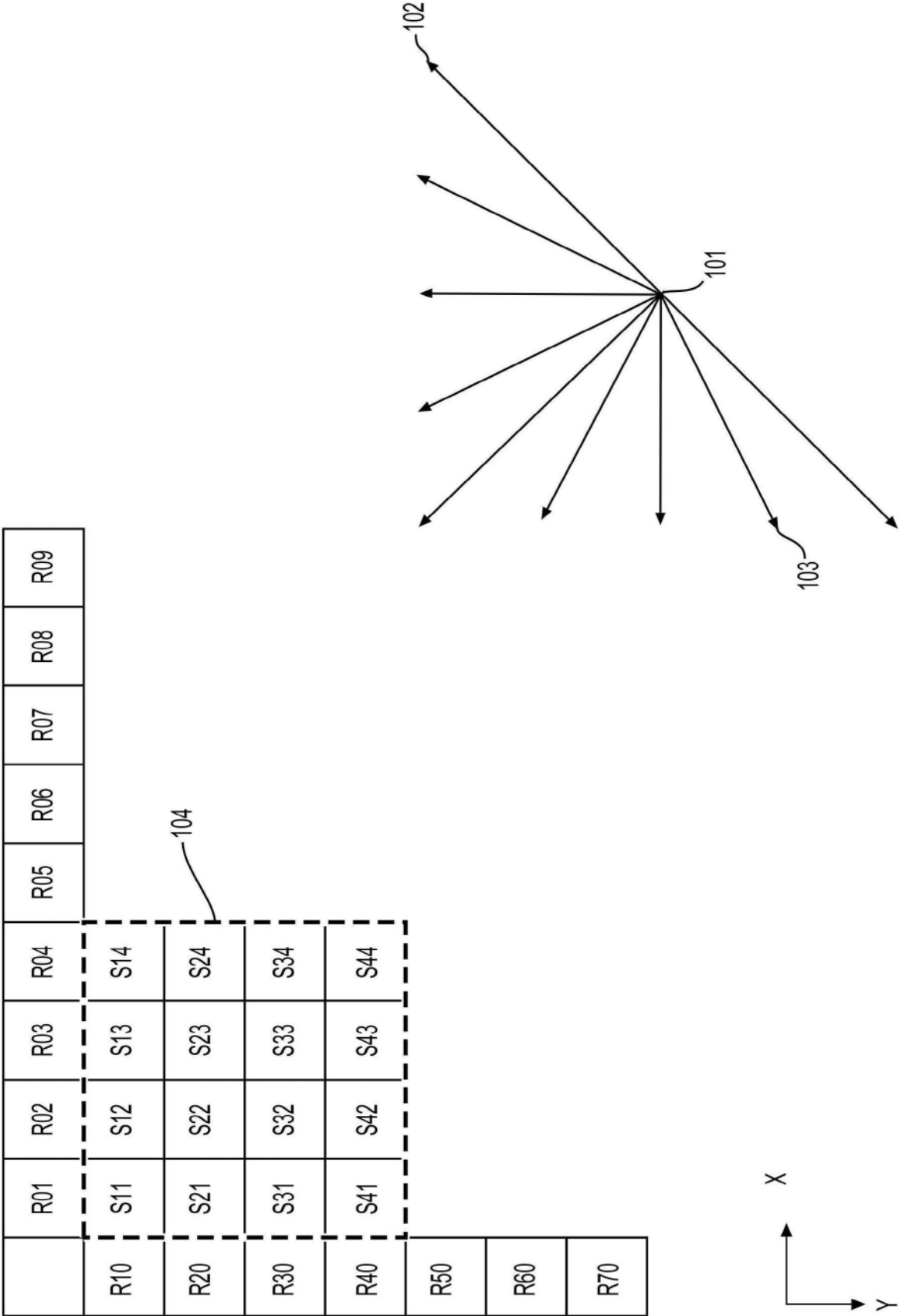


图1A(现有技术)

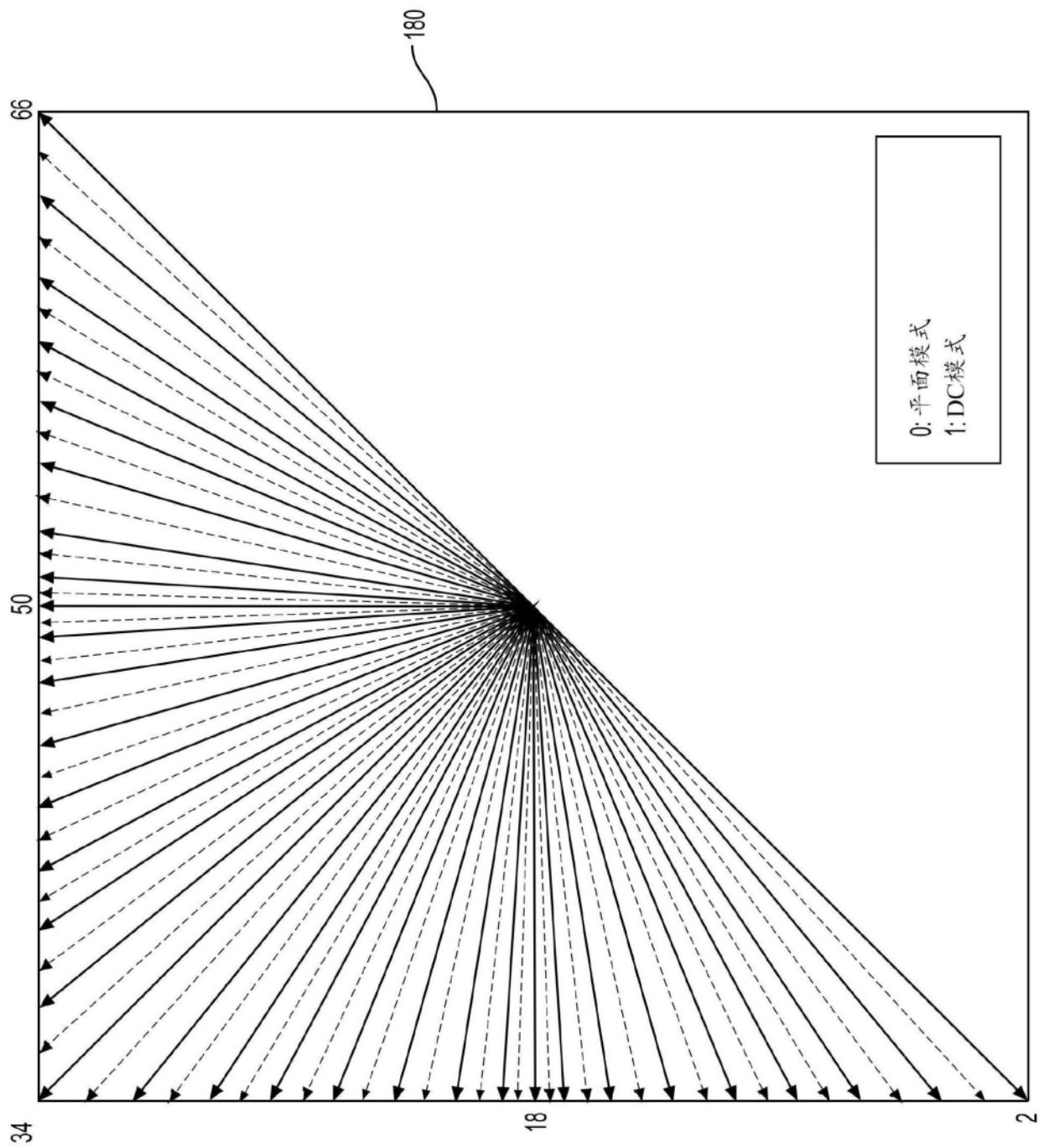


图1B(现有技术)

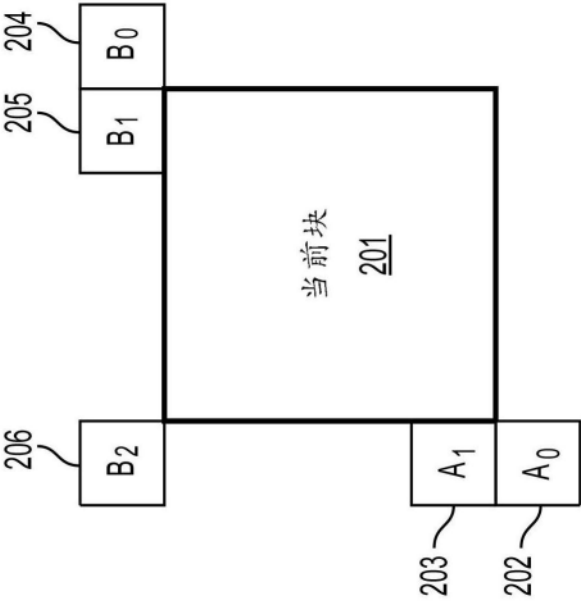


图2 (现有技术)

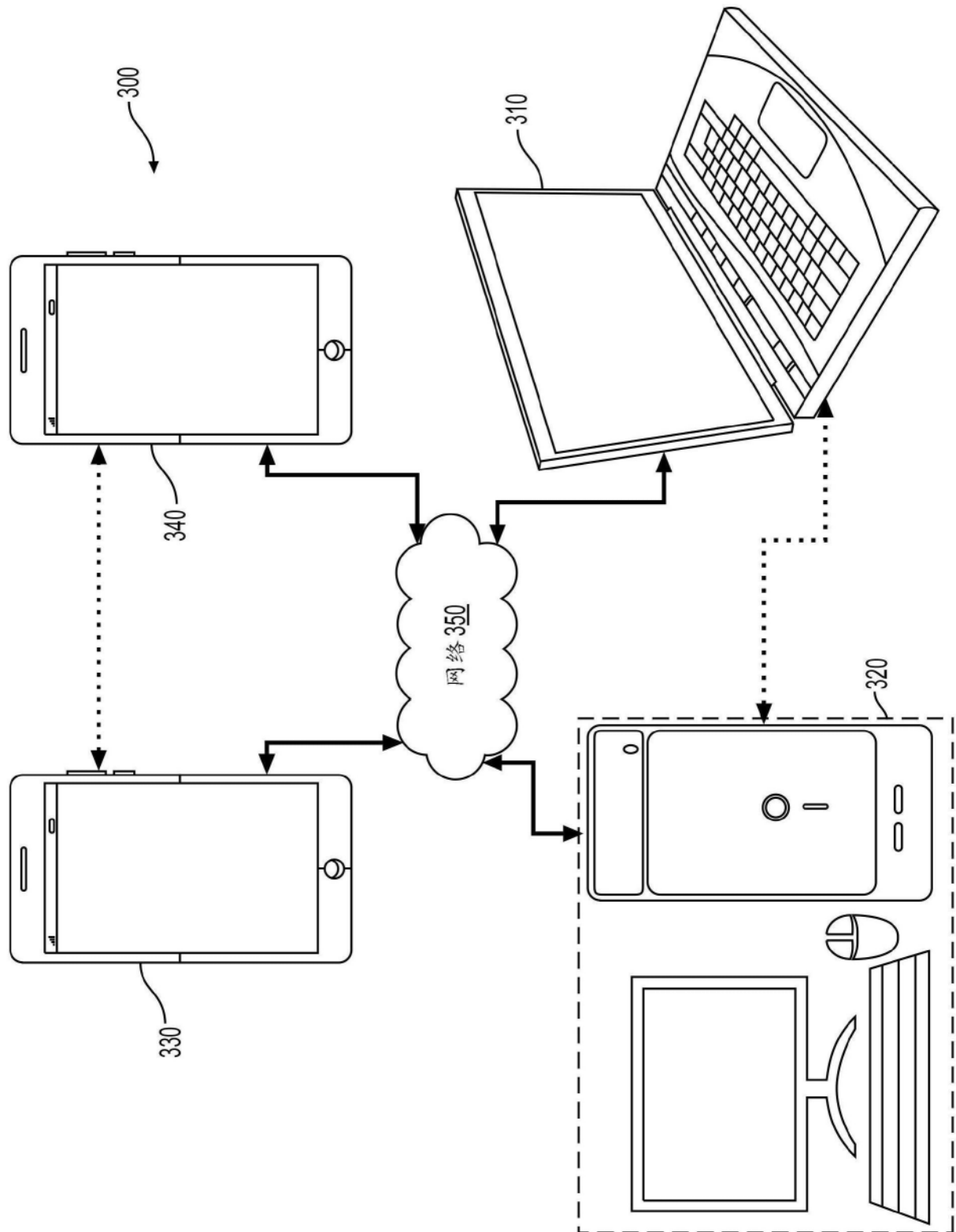


图3

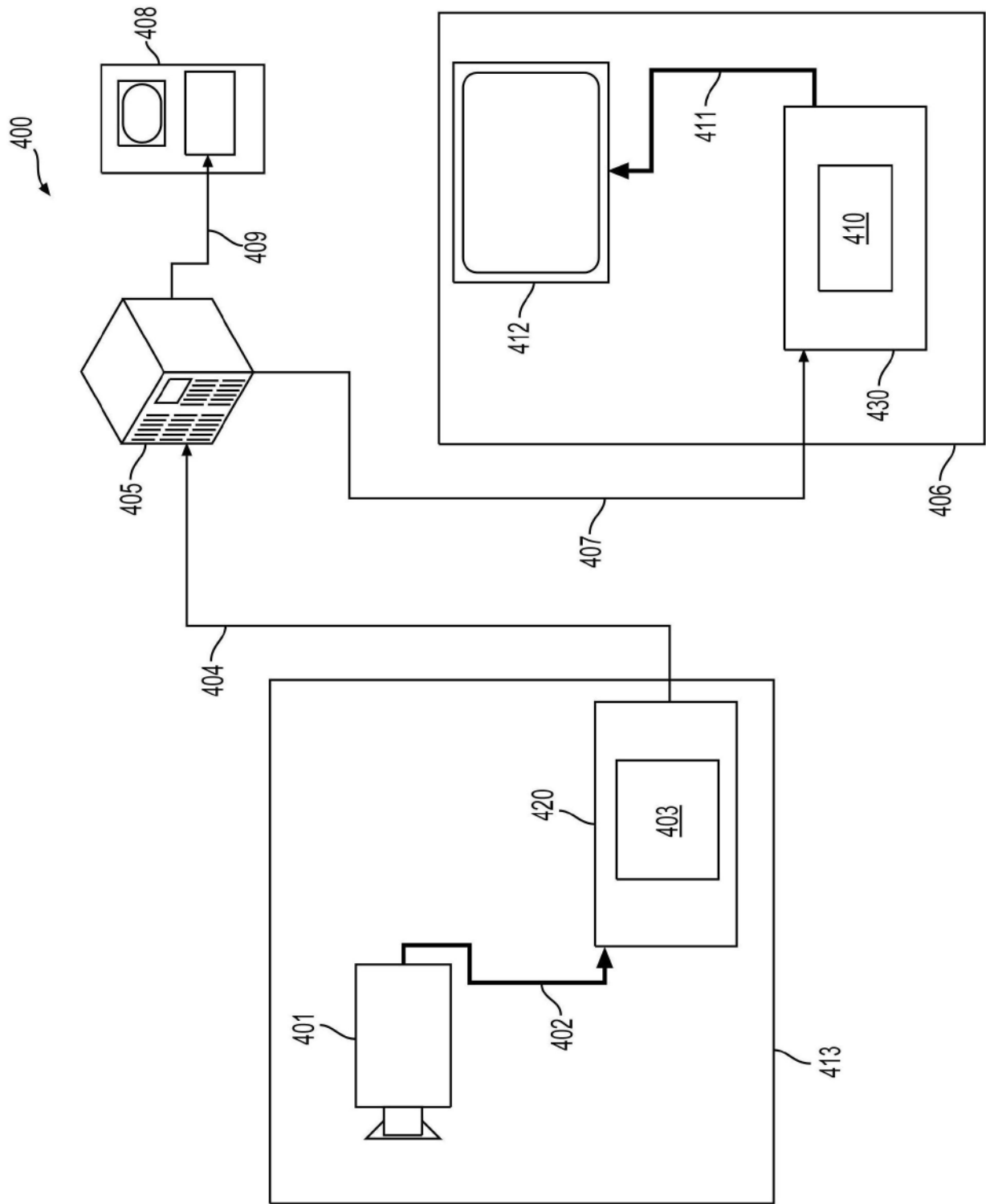


图4

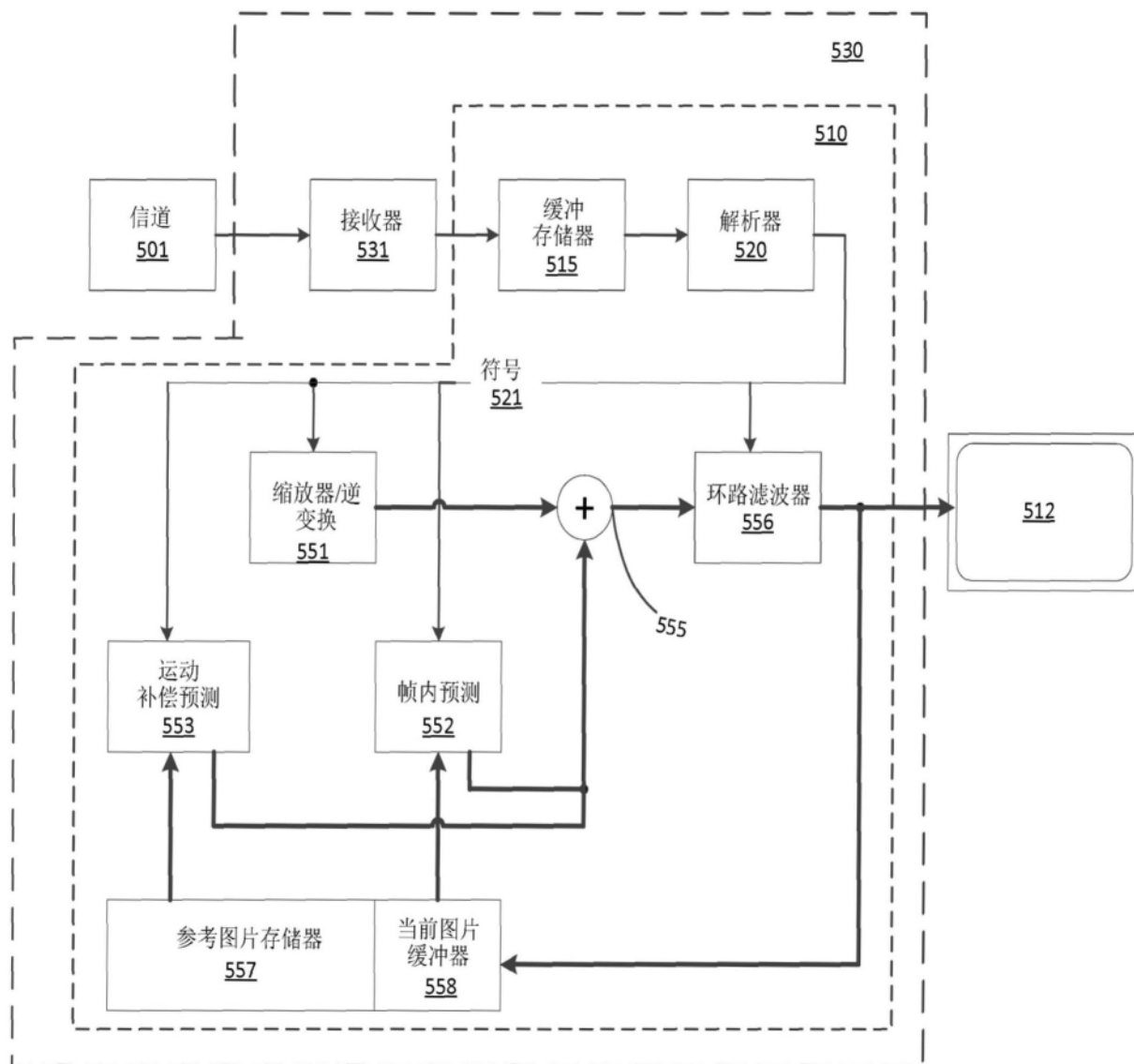


图5

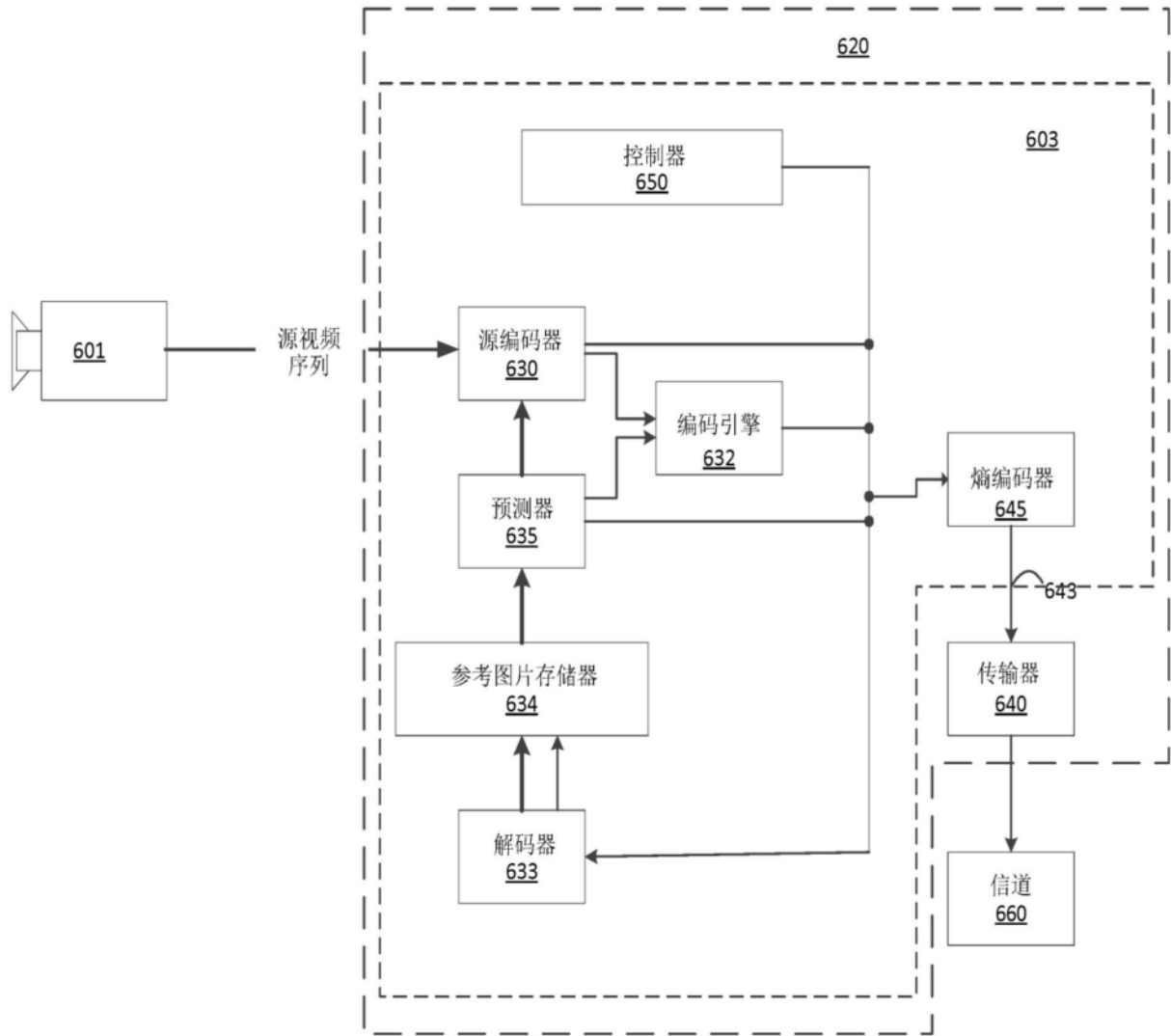


图6

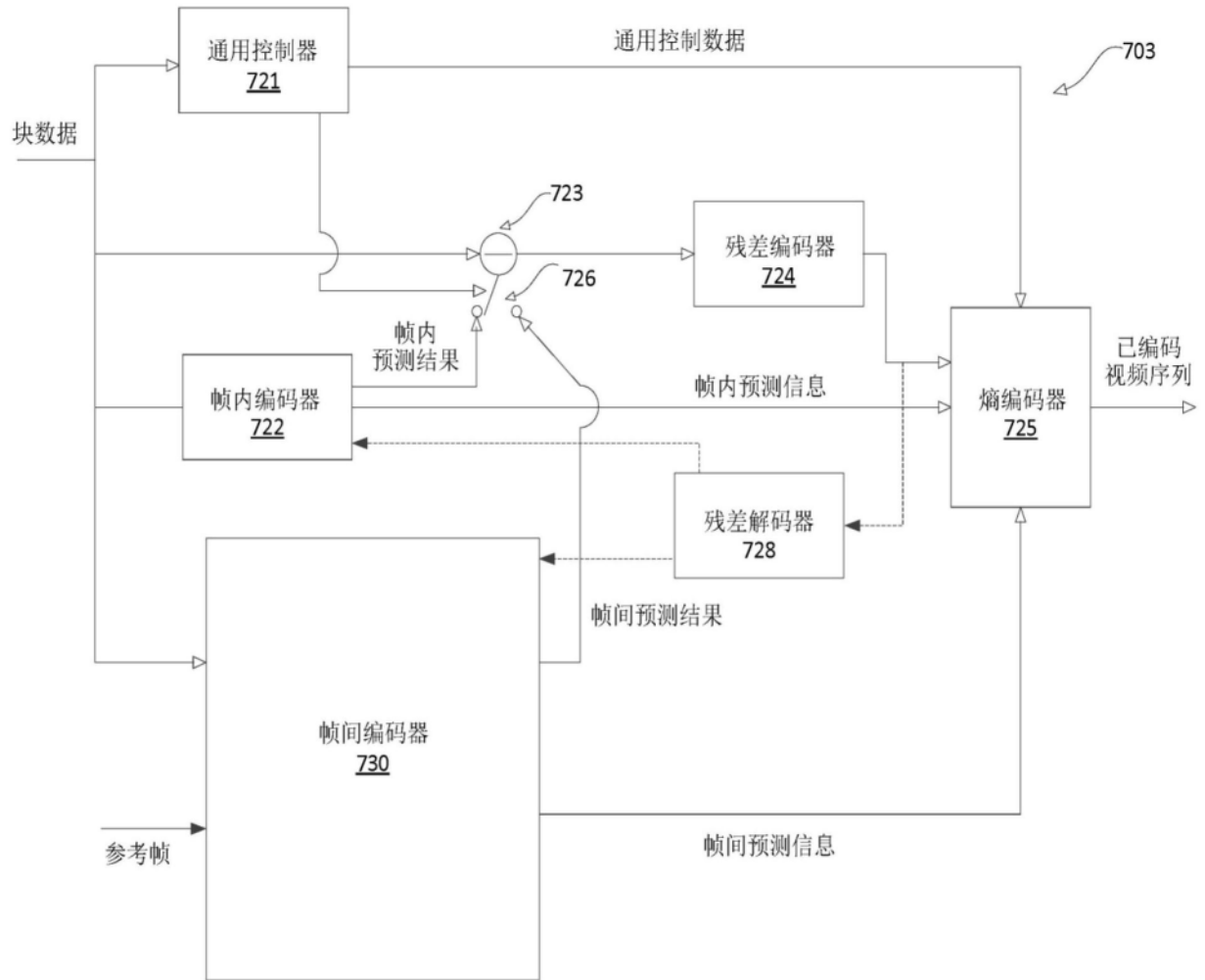


图7

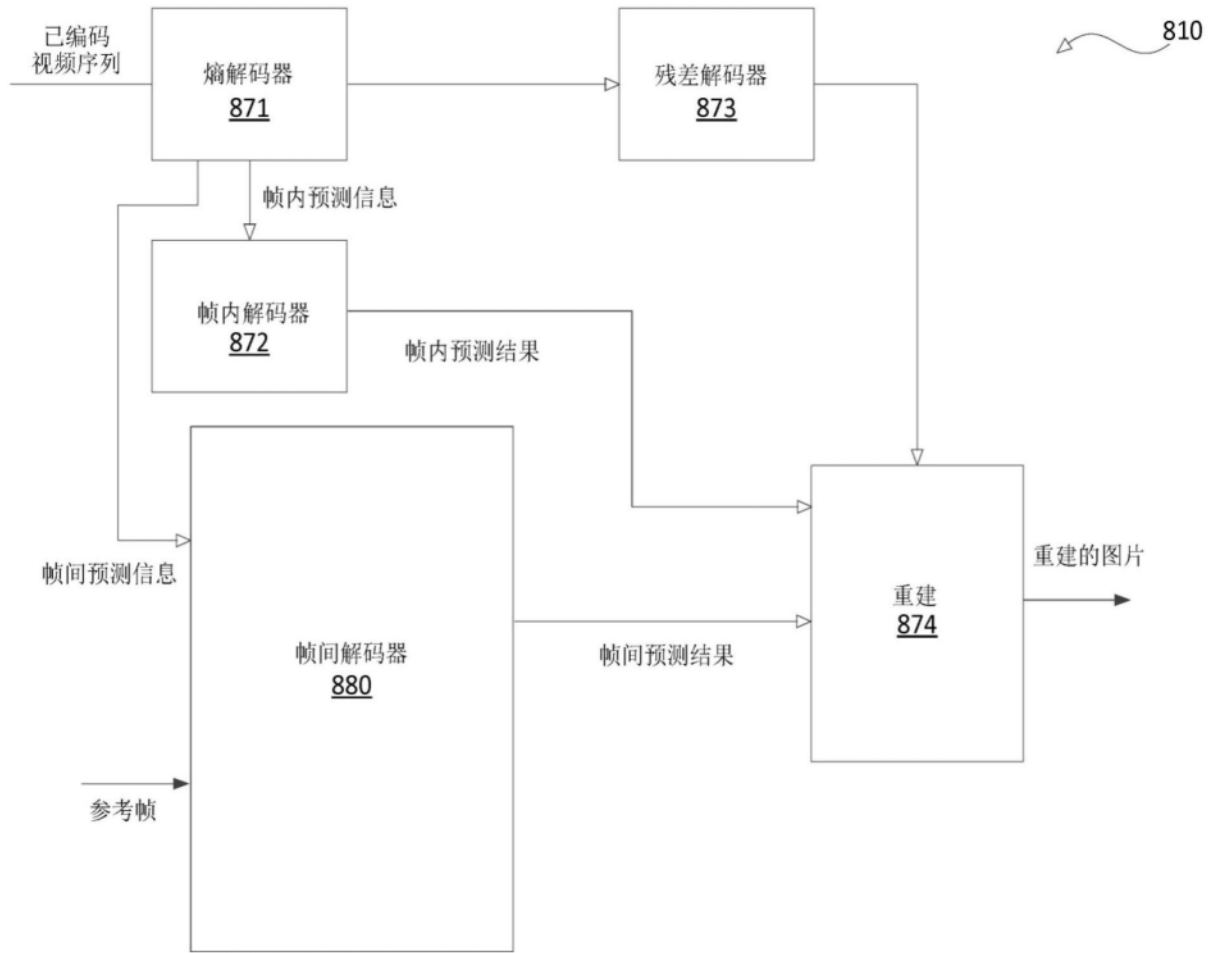


图8

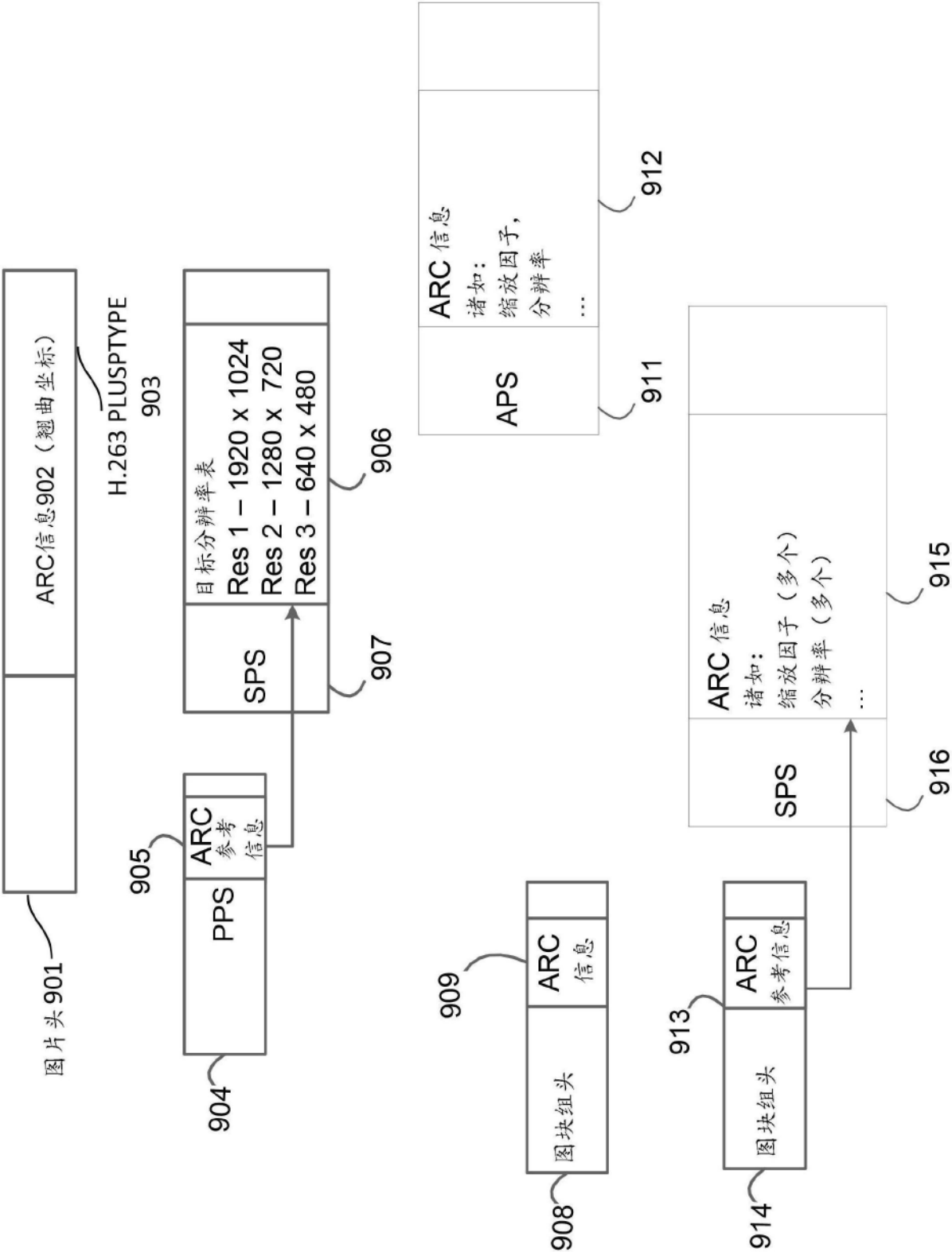


图9

1001	tile_group_header() {	
	...	
1003	if(adaptive_pic_resolution_change_flag) {	
1002	dec_pic_size_idx	u(1)
	}	
	...	
	}	
1010	seq_parameter_set_rbsp() {	描述符
	...	
1011	adaptive_pic_resolution_change_flag	u(1)
1012	if(adaptive_pic_resolution_change_flag) {	
1013 [output_pic_width_in_luma_samples	ue(v)
	output_pic_height_in_luma_samples	ue(v)
1014	reference_pic_size_present_flag	u(1)
	if(reference_pic_size_present_flag)	
	{	
1015 [reference_pic_width_in_luma_samples	ue(v)
	reference_pic_height_in_luma_samples	ue(v)
	}	
1016	num_dec_pic_size_in_luma_samples_minus1	ue(v)
	for(i = 0; i <=	
	num_dec_pic_size_in_luma_samples_minus1;	
	i++) {	
1017 [dec_pic_width_in_luma_samples[i]	ue(v)
	dec_pic_height_in_luma_samples[i]	ue(v)
	}	
	}	
	...	
	}	

图10

profile tier level(profileTierPresentFlag, maxNumSubLayersMinus1) {	描述符
if(profileTierPresentFlag) {	
general_profile_idc	u(7)
general_tier_flag	u(1)
general_constraint_info()	
}	
general_level_idc	u(8)
if(profileTierPresentFlag) {	
num_sub_profiles	u(8)
for(i = 0; i < num_sub_profiles; i++)	
general_sub_profile_idc[i]	u(32)
}	
for(i = 0; i < maxNumSubLayersMinus1; i++)	
sublayer_level_present_flag[i]	u(1)
while(!byte_aligned())	
ptl_alignment_zero_bit	f(1)
for(i = 0; i < maxNumSubLayersMinus1; i++)	
if(sublayer_level_present_flag[i])	
sublayer_level_idc[i]	u(8)
}	

图11

	描述符
general_constraint_info() {	
general_non_packed_constraint_flag	u(1)
general_frame_only_constraint_flag	u(1)
general_non_projected_constraint_flag	u(1)
general_one_picture_only_constraint_flag	u(1)
1201 no_lossless_coding_tool_constraint_flag	u(1)
1202 no_lossy_coding_tool_constraint_flag	u(1)
1205 intra_only_constraint_flag	u(1)
max_bitdepth_constraint_idc	u(4)
max_chroma_format_constraint_idc	u(2)
single_layer_constraint_flag	u(1)
all_layers_independent_constraint_flag	u(1)
no_ref_pic_resampling_constraint_flag	u(1)
no_res_change_in_clvs_constraint_flag	u(1)
one_tile_per_pic_constraint_flag	u(1)
pic_header_in_slice_header_constraint_flag	u(1)
one_slice_per_pic_constraint_flag	u(1)
one_subpic_per_pic_constraint_flag	u(1)
...	
no_partition_constraints_override_constraint_flag	u(1)
1206 no_sao_constraint_flag	u(1)
no_alf_constraint_flag	u(1)
no_radl_constraint_flag	u(1)
no_idr_constraint_flag	u(1)
no_cra_constraint_flag	u(1)
no_gdr_constraint_flag	u(1)
no_aps_constraint_flag	u(1)
while(lbyte_aligned())	
gci_alignment_zero_bit	f(1)
gci_num_reserved_bytes	u(8)
for(i = 0; i < gci_num_reserved_bytes; i++)	
1203 gci_reserved_byte[i]	
1204 }	u(8)
}	

图12

1301

profile tier_level(profileTierPresentFlag, maxNumSubLayersMinus1) {	描述符
if(profileTierPresentFlag) {	
general_profile_idc	u(7)
general_tier_flag	u(1)
num_available_constraint_flags	u(8)
general_constraint_info(num_available_constraint_flags)	
}	
general_level_idc	u(8)
if(profileTierPresentFlag) {	
num_sub_profiles	u(8)
for(i = 0; i < num_sub_profiles; i++)	
general_sub_profile_idc[i]	u(32)
}	
for(i = 0; i < maxNumSubLayersMinus1; i++)	
sublayer_level_present_flag[i]	u(1)
while(!byte_aligned())	
ptl_alignment_zero_bit	f(1)
for(i = 0; i < maxNumSubLayersMinus1; i++)	
if(sublayer_level_present_flag[i])	
sublayer_level_idc[i]	u(8)
}	

图13A

接续图13A

1302	general constraint info(num available constraint flags) {	描述符
	count = 0	
	if(count < num available constraint flags)	
	general_non_packed_constraint_flag	u(1)
	count++	
	if(count < num available constraint flags)	
	general_frame_only_constraint_flag	u(1)
	count++	
	if(count < num available constraint flags)	
	general_non_projected_constraint_flag	u(1)
	count++	
	...	
	while(!byte_aligned())	
	gci_alignment_zero_bit	f(1)
	gci_num_reserved_bytes	u(8)
	for(i = 0; i < gci_num_reserved_bytes; i++)	
	gci_reserved_byte[i]	u(8)
	}	

图13B

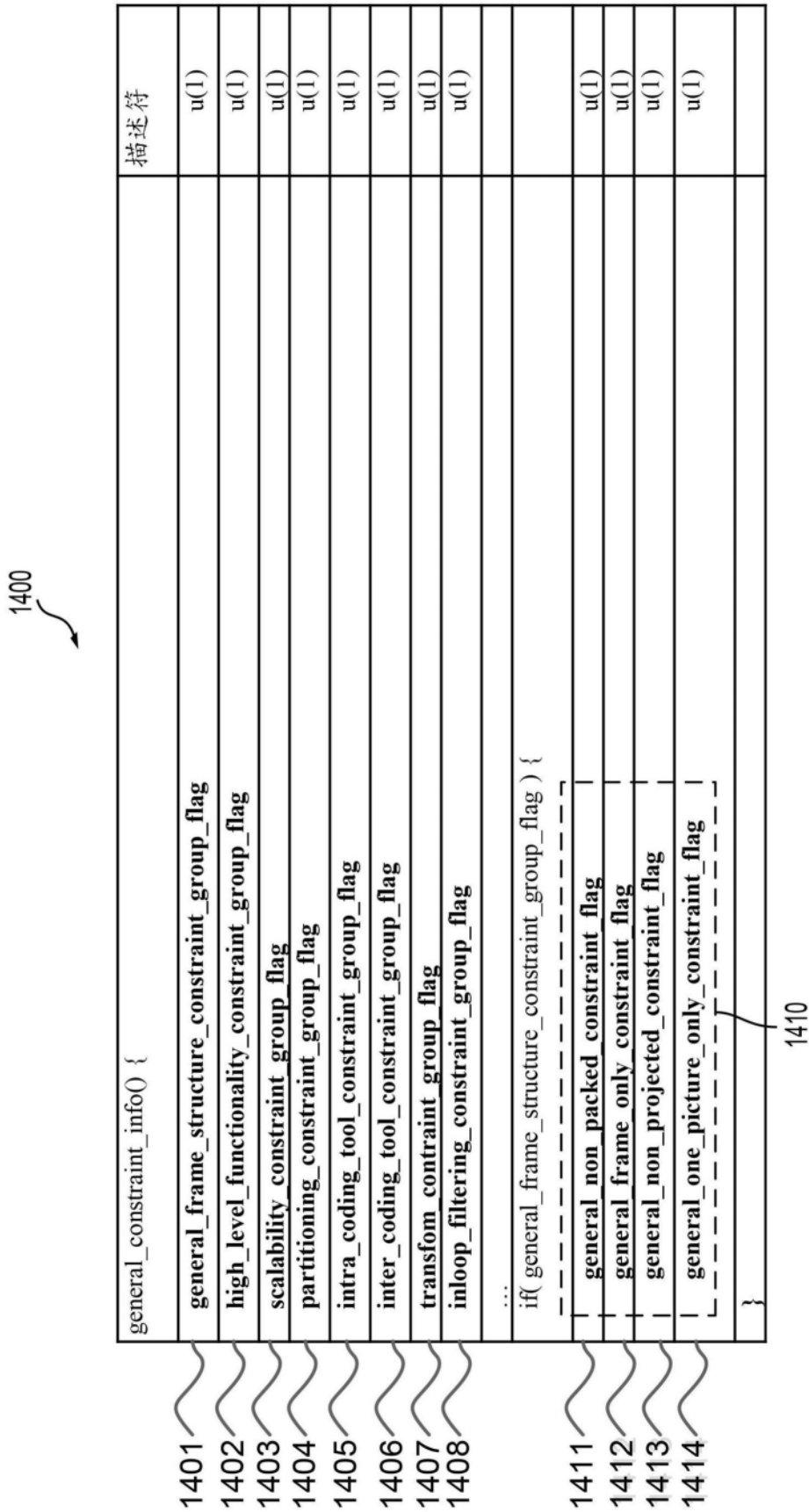


图14A

1400

接续图14A

1420

1430

if(high_level_functionality_constraint_group_flag) {		
{		
pic_header_in_slice_header_constraint_flag		u(1)
no radl constraint flag		u(1)
no idr constraint flag		u(1)
no cra constraint flag		u(1)
no gdr constraint flag		u(1)
no aps constraint flag		u(1)
...		
{		
if(intra_coding_tool_constraint_group_flag) {		
{		
...		
}		
if(inter_coding_tool_constraint_group_flag) {		
{		
...		
{		
while(!byte_aligned()) {		
gci_alignment_zero_bit		f(1)
gci_num_reserved_bytes		u(8)
for(i = 0; i < gci_num_reserved_bytes; i++)		
gci_reserved_byte[i]		u(8)
}		

图14B

1500

general_constraints_info() {		描述符
gci_present_flag		u(1)
if(gci_present_flag) {		
/* general */		
gci_intra_only_constraint_flag		u(1)
gci_all_layers_independent_constraint_flag		u(1)
gci_one_au_only_constraint_flag		u(1)
/* picture format */		
gci_sixteen_minus_max_bitdepth_constraint_idc		u(4)
gci_three_minus_max_chroma_format_constraint_idc		u(2)
/* NAL unit type related */		
gci_no_mixed_nalu_types_in_pic_constraint_flag		u(1)
gci_no_trail_constraint_flag		u(1)
gci_no_stsa_constraint_flag		u(1)
gci_no_rasl_constraint_flag		u(1)
gci_no_radl_constraint_flag		u(1)
gci_no_idr_constraint_flag		u(1)
gci_no_cra_constraint_flag		u(1)
gci_no_gdr_constraint_flag		u(1)
gci_no_aps_constraint_flag		u(1)
gci_no_idr_rpl_constraint_flag		u(1)

图15A

1500
图15A

/* tile, slice, subpicture partitioning */	
gci one tile per pic constraint flag	u(1)
gci pic_header_in_slice_header_constraint_flag	u(1)
gci one slice per pic constraint flag	u(1)
gci no rectangular slice constraint flag	u(1)
gci one slice per subpic constraint_flag	u(1)
gci no subpic info constraint flag	u(1)
/* CTU and block partitioning */	
gci three_minus_max_log2_ctu_size_constraint_idc	u(2)
gci no_partition_constraints_override_constraint_flag	u(1)
gci no mtt constraint flag	
gci no_qtbt dual tree intra constraint flag	u(1)
/* intra */	
gci no palette constraint flag	u(1)
gci no ibc constraint flag	u(1)
gci no isp constraint flag	u(1)
gci no mrl constraint flag	u(1)
gci no mip constraint flag	u(1)
gci no cclm constraint flag	u(1)

图15B

1500
 1516
 1500

1500
 1516
 1500

/* inter */	
gci_no_ref_pic_resampling_constraint_flag	u(1)
gci_no_res_change_in_clvs_constraint_flag	u(1)
gci_no_weighted_prediction_constraint_flag	u(1)
gci_no_ref_wraparound_constraint_flag	u(1)
gci_no_temporal_mvp_constraint_flag	u(1)
gci_no_sbtmvp_constraint_flag	u(1)
gci_no_amvr_constraint_flag	u(1)
gci_no_bdof_constraint_flag	u(1)
gci_no_smvd_constraint_flag	u(1)
gci_no_dmvr_constraint_flag	u(1)
gci_no_mmvd_constraint_flag	u(1)
gci_no_affine_motion_constraint_flag	u(1)
gci_no_prof_constraint_flag	u(1)
gci_no_bew_constraint_flag	u(1)
gci_no_ciip_constraint_flag	u(1)
gci_no_gpm_constraint_flag	u(1)

图15C

/* transform_quantization_residual */		
gci_no_luma_transform_size_64_constraint_flag		u(1)
gci_no_transform_skip_constraint_flag		u(1)
gci_no_bdpcm_constraint_flag		u(1)
gci_no_mts_constraint_flag		u(1)
gci_no_lfst_constraint_flag		u(1)
gci_no_joint_cbr_constraint_flag		u(1)
gci_no_sbt_constraint_flag		u(1)
gci_no_act_constraint_flag		u(1)
gci_no_explicit_scaling_list_constraint_flag		u(1)
gci_no_dep_quant_constraint_flag		u(1)
gci_no_sign_data_hiding_constraint_flag		u(1)
gci_no_cu_qp_delta_constraint_flag		u(1)
gci_no_chroma_qp_offset_constraint_flag		u(1)
/* loop filter */		
gci_no_sao_constraint_flag		u(1)
gci_no_alf_constraint_flag		u(1)
gci_no_ccalf_constraint_flag		u(1)
gci_no_lmcs_constraint_flag		u(1)
gci_no_ladf_constraint_flag		u(1)
gci_no_virtual_boundaries_constraint_flag		u(1)
gci_num_reserved_bits		u(8)
for(i = 0; i < gci_num_reserved_bits; i++)		
gci_reserved_zero_bit[i]		u(1)
}		
while(!byte_aligned())		
gci_alignment_zero_bit		f(1)
}		

1500

1517

1518

图15D

83

profile_tier_level(profileTierPresentFlag, MaxNumSubLayersMinus1) {	描述符
if(profileTierPresentFlag) {	
general_profile_idc	u(7)
general_tier_flag	u(1)
}	
general_level_idc	u(8)
ptl_frame_only_constraint_flag	u(1)
ptl_multilayer_enabled_flag	u(1)
if(profileTierPresentFlag)	
general_constraints_info()	
for(i = MaxNumSubLayersMinus1 - 1; i >= 0; i--)	
ptl_sublayer_level_present_flag[i]	u(1)
while(!byte_aligned())	
ptl_reserved_zero_bit	u(1)
for(i = MaxNumSubLayersMinus1 - 1; i >= 0; i--)	
if(ptl_sublayer_level_present_flag[i])	
sublayer_level_idc[i]	u(8)
if(profileTierPresentFlag) {	
ptl_num_sub_profiles	u(8)
for(i = 0; i < ptl_num_sub_profiles; i++)	
general_sub_profile_idc[i]	u(32)
}	
}	

图16

seq_parameter_set_rbsp() {	描述符
sps_seq_parameter_set_id	u(4)
...	
sps_field_seq_flag	u(1)
...	
sps_extension_flag	u(1)
if(sps_extension_flag)	
while(more_rbsp_data())	
sps_extension_data_flag	u(1)
rbsp_trailing_bits()	
}	

图17

描述符	
	video_parameter_set_rbsp() {
	...
u(1)	vps_field_seq_flag
	...
	}

图18

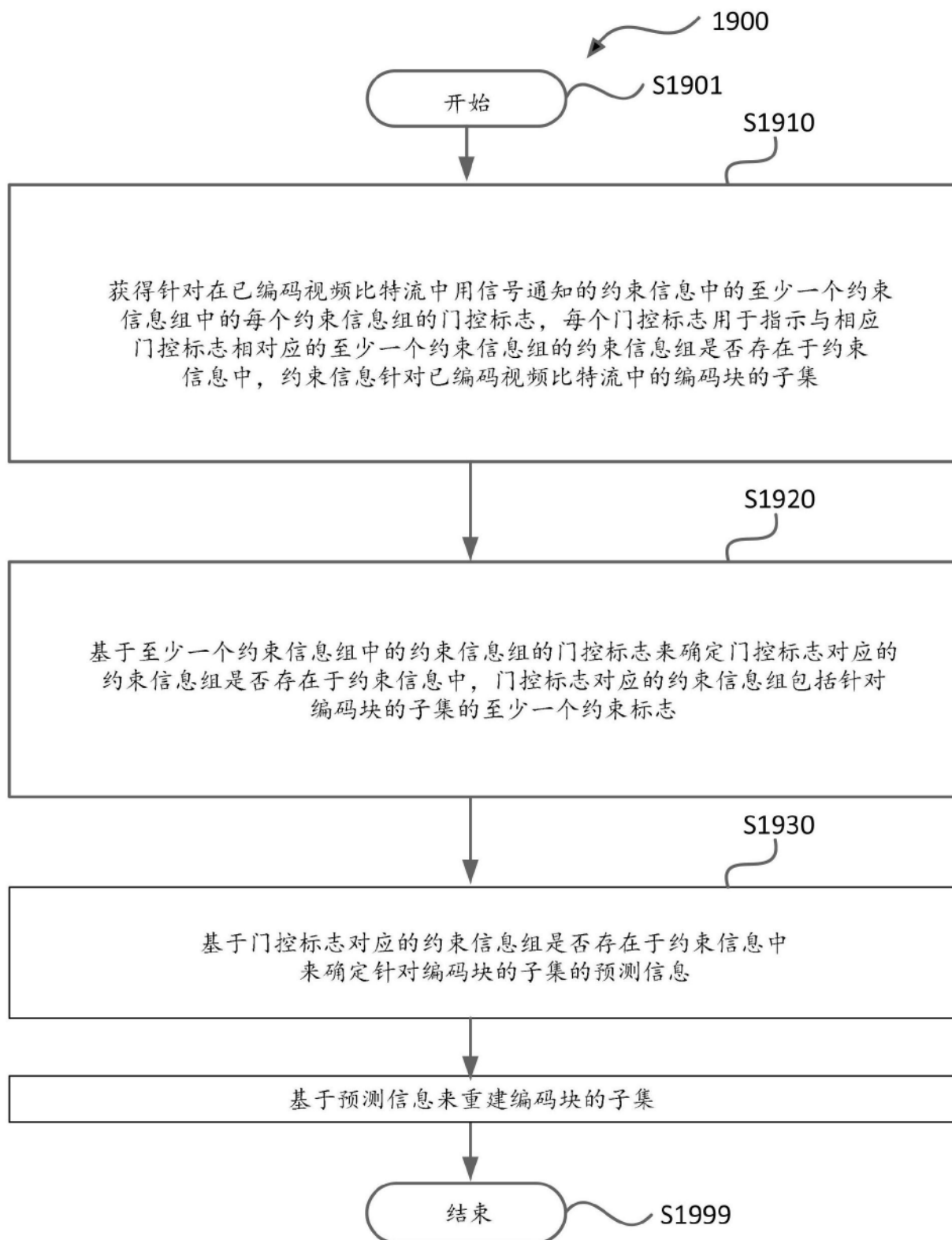


图19

video_parameter_set_rbsp() {	描述符
vps_video_parameter_set_id	u(4)
vps_max_layers_minus1	u(8)
for(i = 0; i <= vps_max_layers_minus1; i++) {	
vps_included_layer_id[i]	u(7)
vps_reserved_zero_bit	u(1)
}	
vps_constraint_info_present_flag	u(1)
vps_constant_poc_cycle_per_au	u(1)
if(vps_constant_poc_cycle_per_au)	
vps_poc_cycle_au	u(8)
...	
}	
slice_header() {	描述符
slice_pic_parameter_set_id	ue(v)
if(rect_slice_flag NumBricksInPic > 1)	
slice_address	u(v)
if(!rect_slice_flag && !single_brick_per_slice_flag)	
num_bricks_in_slice_minus1	ue(v)
slice_type	ue(v)
if(NalUnitType == GRA_NUT)	
recovery_poc_cnt	se(v)
slice_pic_order_cnt_lsb	u(v)
...	
if(!vps_constant_poc_cycle_per_au)	
slice_poc_cycle_au	u(8)
}	

图20

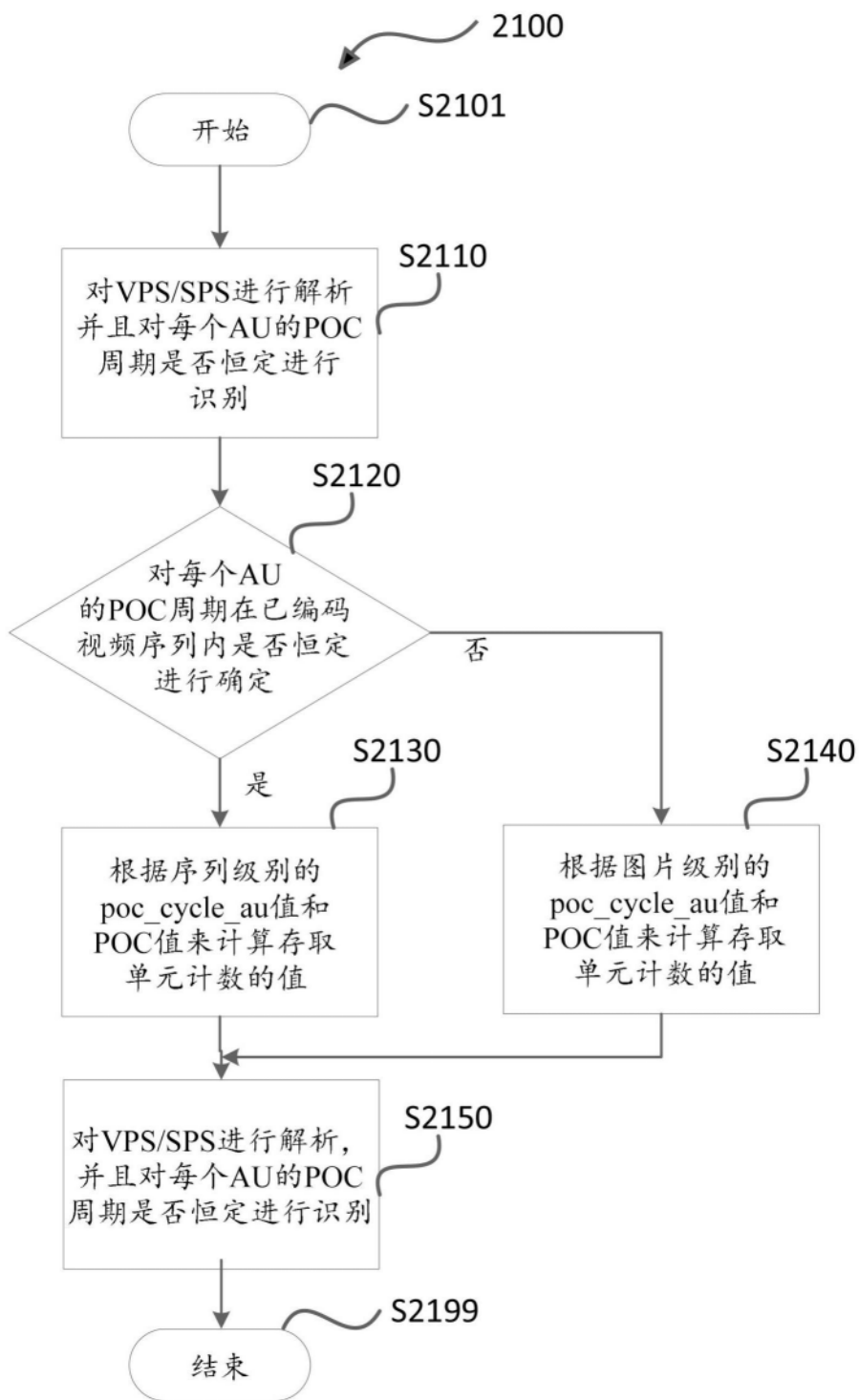


图21

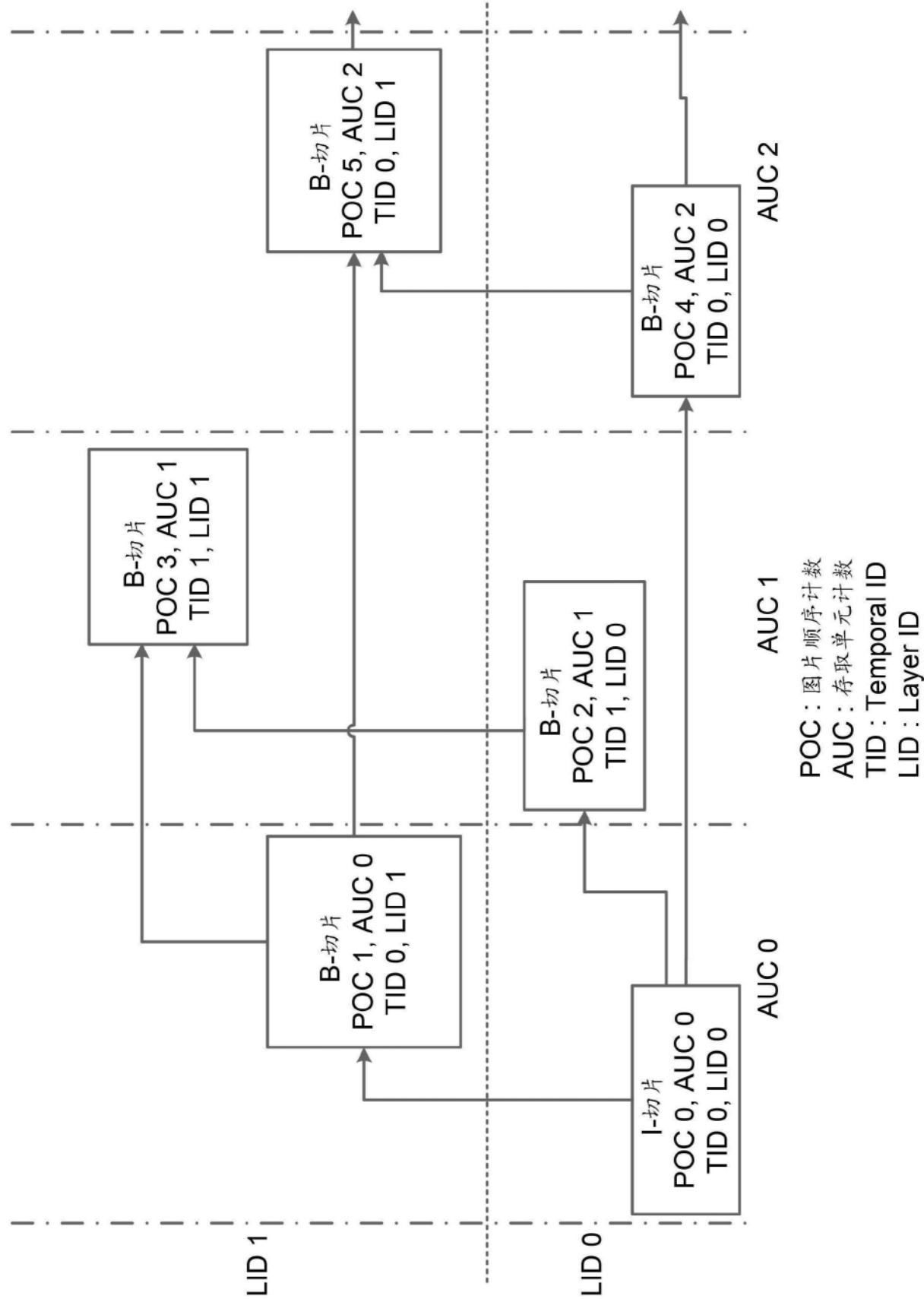


图22

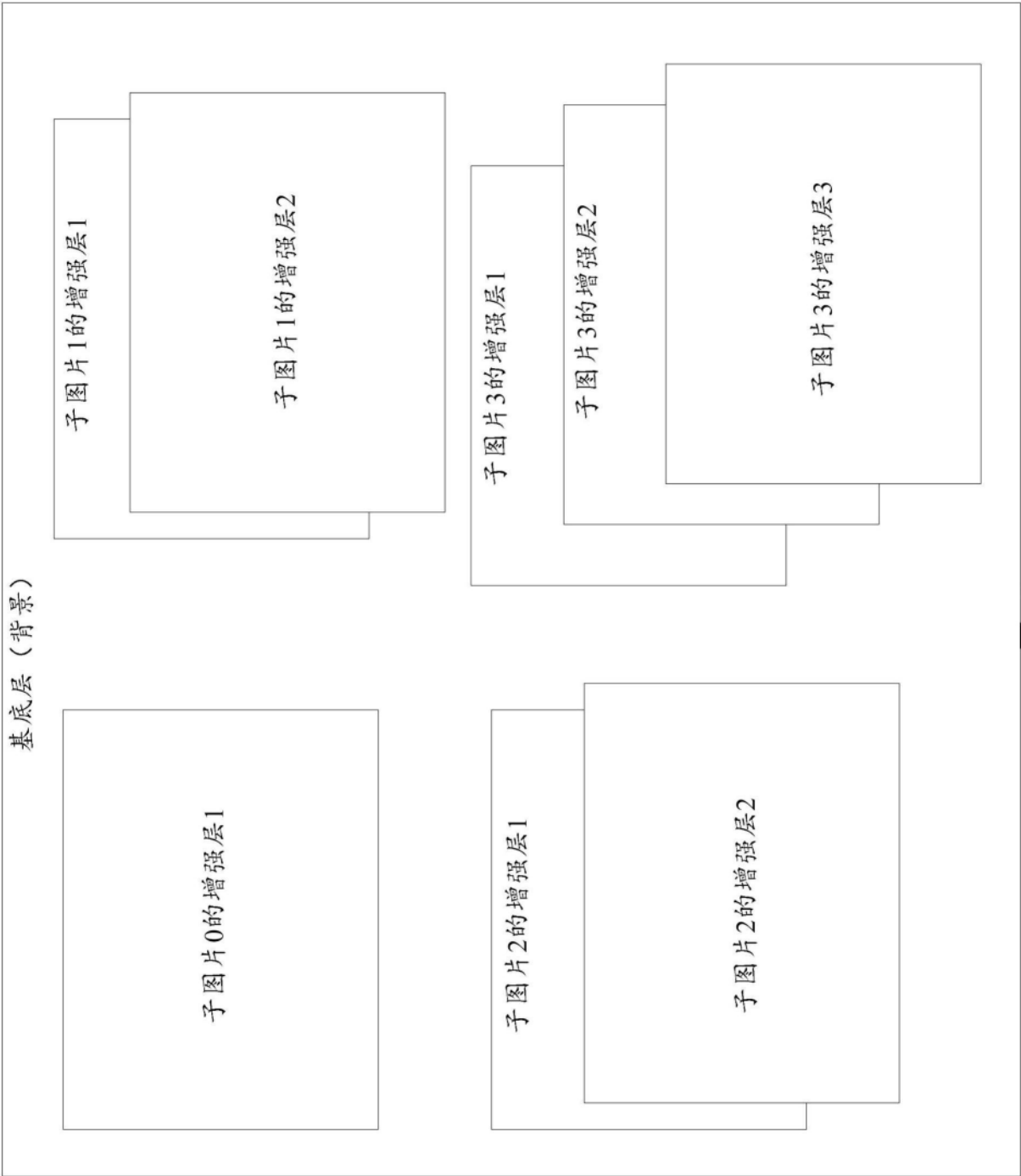


图23

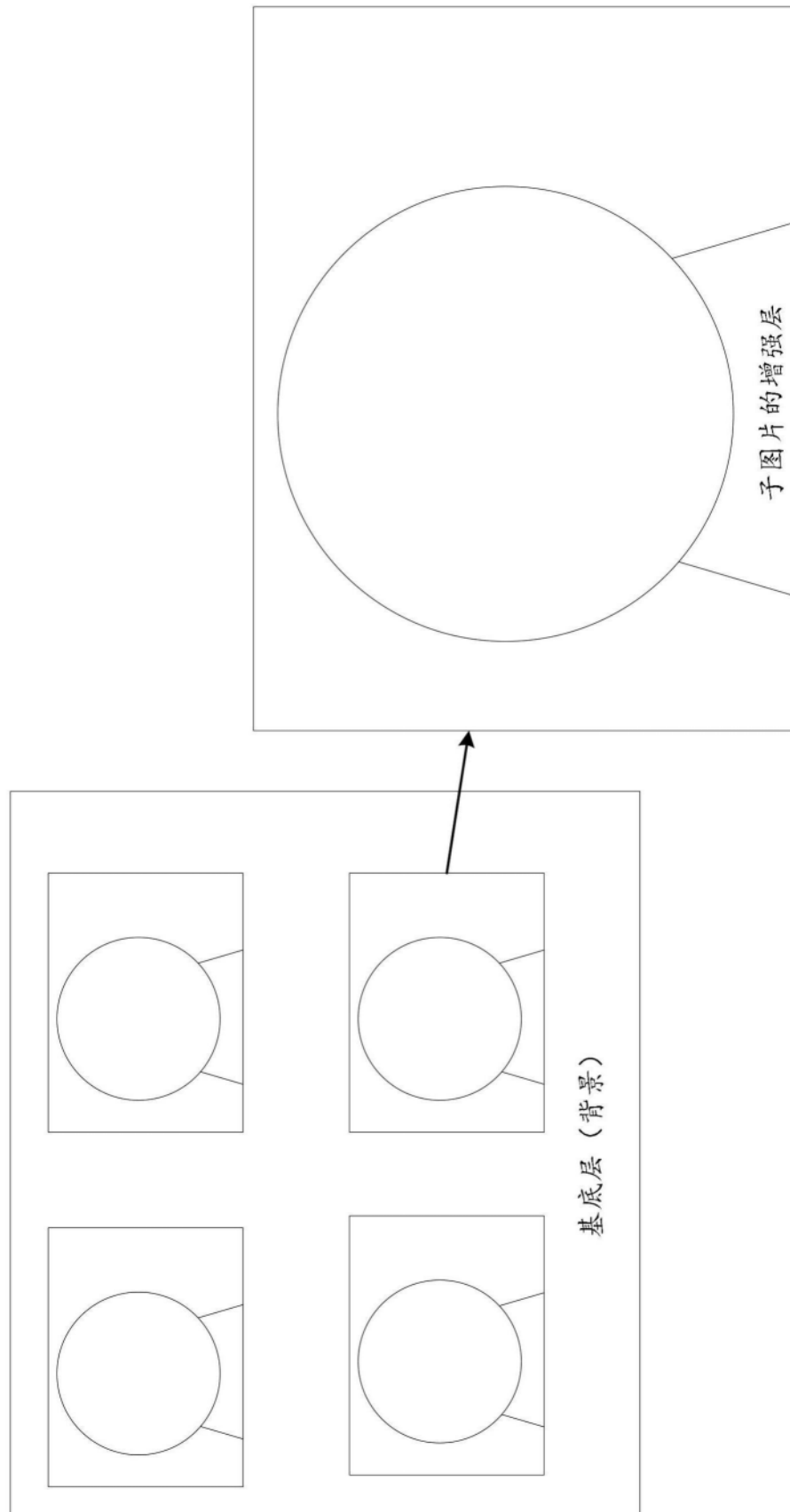


图24

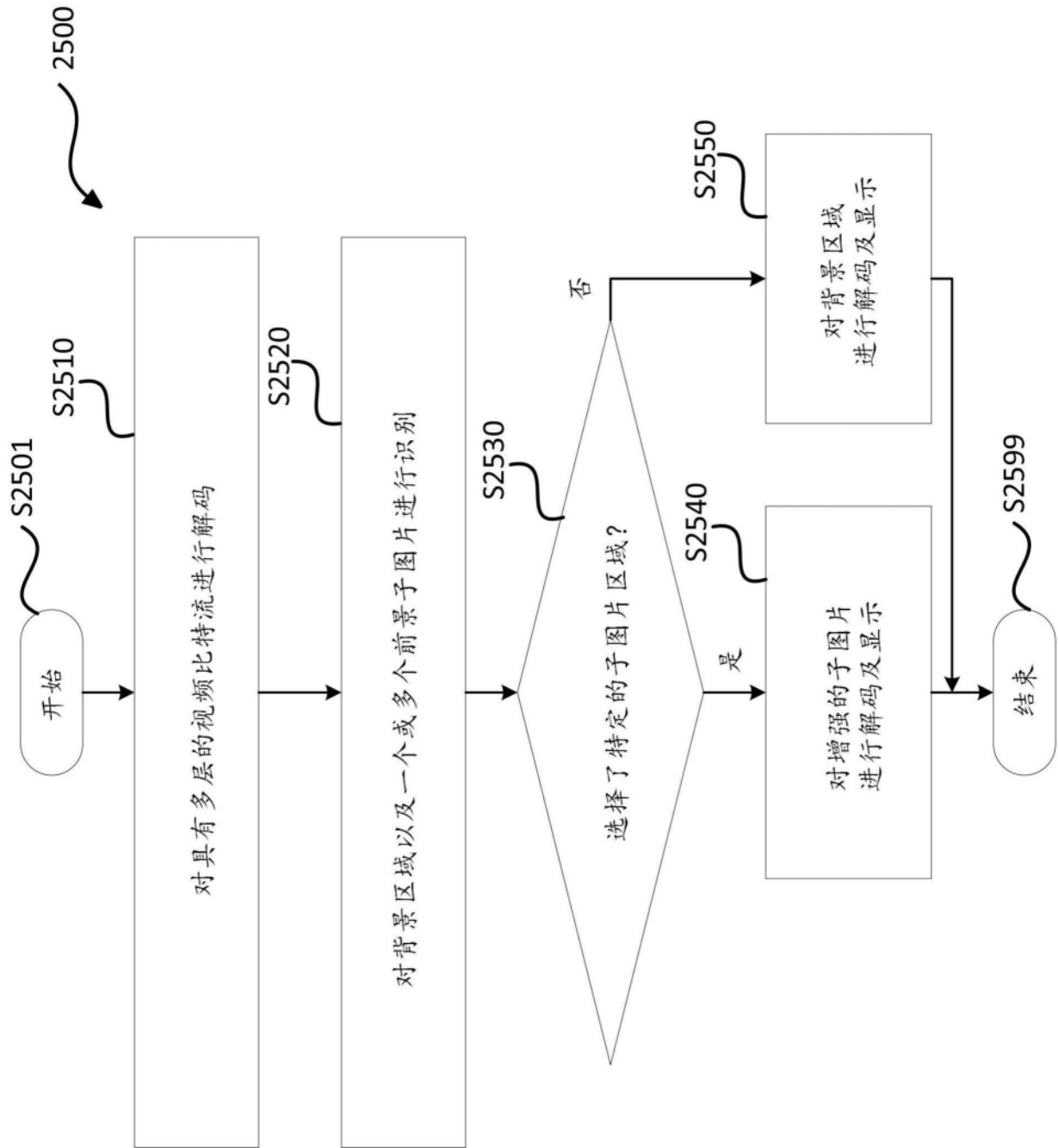


图25

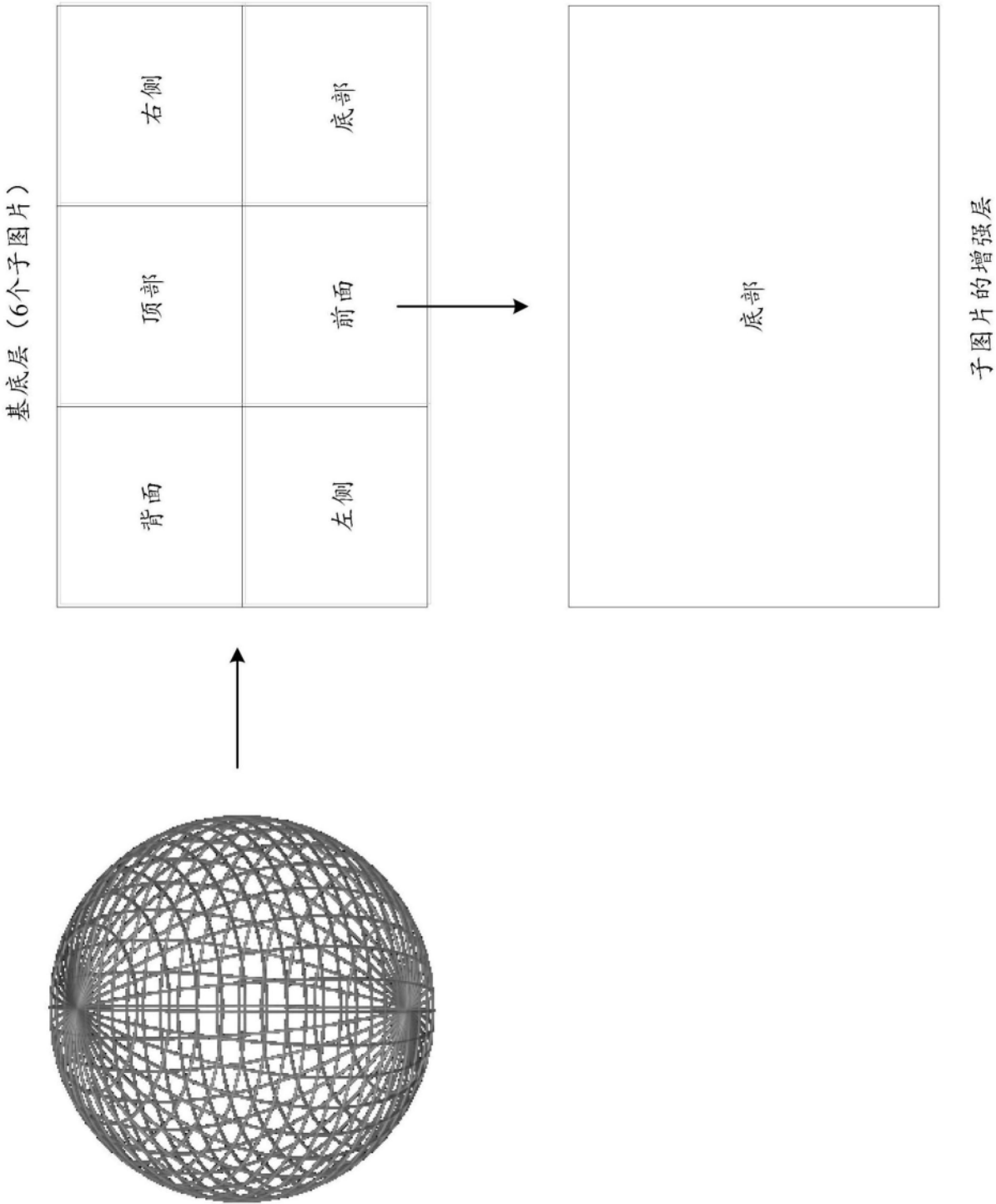


图26

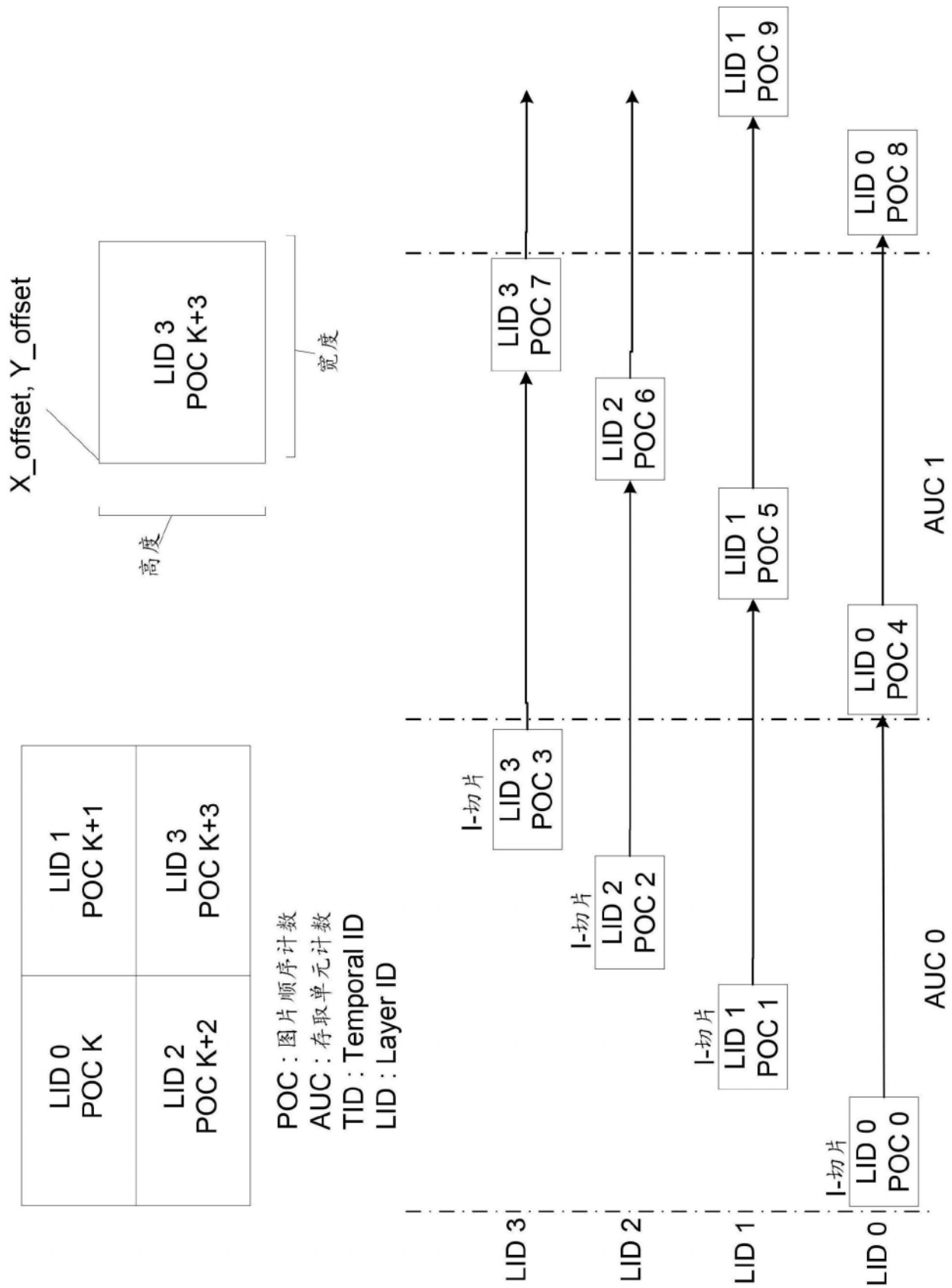


图27

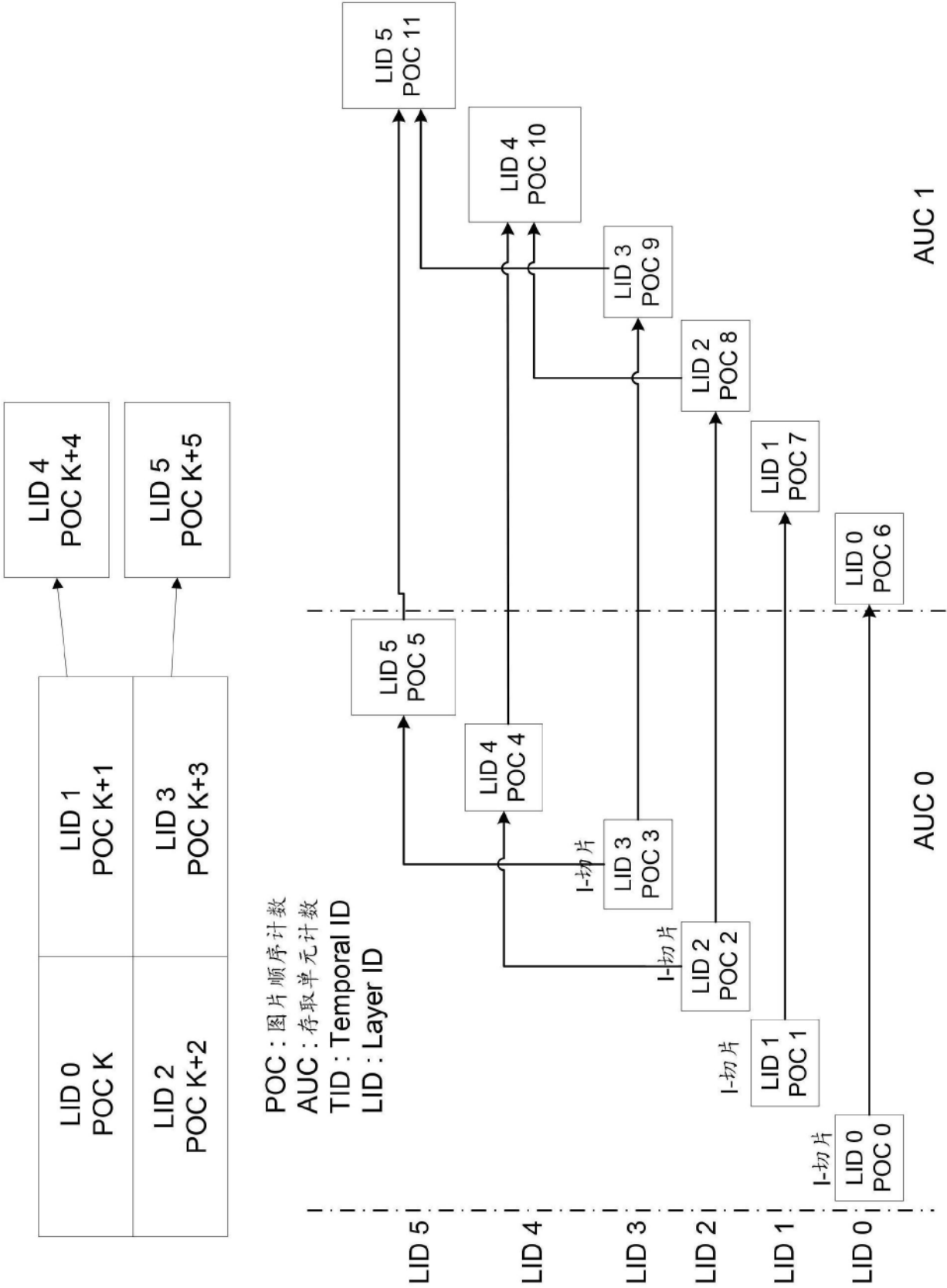


图28

video_parameter_set_rbsp() {	描述符
vps_video_parameter_set_id	u(4)
vps_max_layers_minus1	u(8)
for(i = 0; i <= vps_max_layers_minus1; i++) {	
vps_included_layer_id[i]	u(7)
vps_reserved_zero_bit	u(1)
}	
...	
vps_sub_picture_dividing_flag	u(1)
if(vps_sub_picture_dividing_flag) {	
vps_full_pic_width_in_luma_samples	ue(v)
vps_full_pic_height_in_luma_samples	ue(v)
}	
...	
}	

图29A

接续图29A

	描述符
▪ seq_parameter_set_rbsp() {	
sps_decoding_parameter_set_id	u(4)
sps_video_parameter_set_id	u(4)
sps_max_sub_layers_minus1	u(3)
...	↴
pic_width_in_luma_samples	ue(v)
pic_height_in_luma_samples	ue(v)
if(vps_sub_picture_dividing_flag) {	↴
pic_offset_x	ue(v)
pic_offset_y	ue(v)
}	↴
...	↴
}	↴

图29B

sub_region_partitioning_info(payloadSize) {	描述符
num_sub_region	ue(v)
num_layers	ue(v)
for(i = 0; i<= num_layers; i++)	
layer_id [i]	ue(v)
for(i = 0; i < num_layers; i++)	
for(j = 0; j<i; j++)	
direct_dependency_flag[i][j]	u(1)
for(i = 0; i<num_sub_region; i++) {	
num_layers_for_region[i]	ue(v)
for(j = 0; j<num_layers_for_region[i]; j++)	
sub_region_layer_id[i][j]	ue(v)
sub_region_offset_x[i]	ue(v)
sub_region_offset_y[i]	ue(v)
sub_region_width[i]	ue(v)
sub_region_height[i]	ue(v)
}	
...	
}	

图30

video parameter set rbsp() {	描述符
...	
vps max layers minus1	u(6)
num output layer sets	ue(v)
num profile tier level	ue(v)
for(i = 0; i < num profile tier level; i++)	
profile tier level(vps max sub layers minus1)	
for(i = 0; i < num output layer sets; i++)	
for(j = 0; j < NumLayersInIdList[i]; j++) {	
output layer flag[i][j]	u(1)
profile tier level idx[i][j]	u(v)
}	
...	
}	

图31

video parameter set rbsp() {	描述符
...	
vps max layers minus1	u(6)
if(vps max layers minus1 > 0) {	
num output layer sets	ue(v)
num profile tier level	ue(v)
}	
for(i = 0; i < num profile tier level; i++)	
profile tier level(vps max sub layers minus1)	
for(i = 0; i < num output layer sets; i++) {	
vps output layers mode[i]	u(2)
vps ptl signal flag[i]	u(1)
for(j = 0; j < NumLayersInList[i]; j++) {	
if(vps output layers mode[i] == 2)	
output layer flag[i][j]	u(1)
if(vps ptl signal flag[i])	
profile tier level idx[i][j]	u(v)
}	
}	
...	
}	

图32

video parameter set rbsp() {	描述符
...	
vps max layers minus1	u(6)
if(vps max layers minus1 > 0) {	
num output layer sets	ue(v)
num profile tier level	ue(v)
}	
max subpics minus1	u(8)
for(i = 0; i < max subpics minus1 ; i++) {	
sub pic id[i]	u(8)
}	
for(i = 0; i < num profile tier level; i++)	
profile tier level(vps max sub layers minus1)	
for(i = 0; i < num output layer sets; i++) {	
vps output layers mode[i]	u(2)
vps ptl signal flag[i]	u(1)
for(j = 0; j < NumLayersInList[i]; j++) {	
num output subpic layer[i][j]	ue(v)
for(k = 0; k < num output subpic layer[i][j]; k++)	
sub pic id layer[i][j][k]	u(8)
if(vps output layers mode[i] == 2)	
output layer flag[i][j]	u(1)
if(vps ptl signal flag[i])	
profile tier level idx[i][j]	u(v)
}	
}	
...	
}	

图33

video_parameter_set_rbsp() {	描述符
vps video parameter set id	u(4)
vps max layers minus1	u(6)
vps max sublayers minus1	u(3)
if(vps_max_layers_minus1 > 0 && vps_max_sublayers_minus1 > 0)	
vps_all_layers_same_num_sublayers_flag	u(1)
if(vps_max_layers_minus1 > 0)	
vps all independent layers flag	u(1)
for(i = 0; i <= vps_max_layers_minus1; i++) {	
vps layer id[i]	u(6)
if(i > 0 && !vps_all_independent_layers_flag) {	
vps independent layer flag[i]	u(1)
if(!vps_independent_layer_flag[i]) {	
for(j = 0; j < i; j++)	
vps_direct_ref_layer_flag[i][j]	u(1)
max tid ref present flag[i]	u(1)
if(max tid ref present flag[i])	
max tid il ref pics plus1[i]	u(3)
}	
}	
}	
if(vps_max_layers_minus1 > 0) {	
if(vps_all_independent_layers_flag)	
each layer is an ols flag	u(1)
if(!each layer is an ols flag) {	
if(!vps_all_independent_layers_flag)	
ols mode idc	u(2)
if(ols_mode_idc == 2) {	
num output layer sets minus1	u(8)
for(i = 1; i <= num_output_layer_sets_minus1; i++)	
for(j = 0; j <= vps_max_layers_minus1; j++)	
ols output layer flag[i][j]	u(1)
}	
}	
}	
...	
}	

图34

video parameter set rbsp() {	描述符
vps video parameter set id	u(4)
vps max layers minus1	u(6)
vps max sublayers minus1	u(3)
if(vps max layers minus1 > 0 && vps max sublayers minus1 > 0)	
vps all layers same num sublayers flag	u(1)
if(vps max layers minus1 > 0)	
vps all independent layers flag	u(1)
for(i = 0; i <= vps max layers minus1; i++) {	
vps layer id[i]	u(6)
if(i > 0 && !vps all independent layers flag) {	
vps independent layer flag[i]	u(1)
if(!vps independent layer flag[i]) {	
for(j = 0; j < i; j++)	
vps direct ref layer flag[i][j]	u(1)
max tid ref present flag[i]	u(1)
if(max tid ref present flag[i])	
max tid il ref pics plus1[i]	u(3)
}	
}	
}	
if(vps max layers minus1 > 0) {	
if(vps all independent layers flag)	
each layer is an ols flag	u(1)
if(!each layer is an ols flag) {	
if(!vps all independent layers flag)	
ols mode idc	u(2)
if(ols mode idc == 2 && vps max layers minus1 > 1) {	
num output layer sets minus1	u(8)
for(i = 1; i <= num output layer sets minus1; i++)	
for(j = 0; j <= vps max layers minus1; j++)	
ols output layer flag[i][j]	u(1)
}	
}	
}	
}	
}	
...	
}	

图35

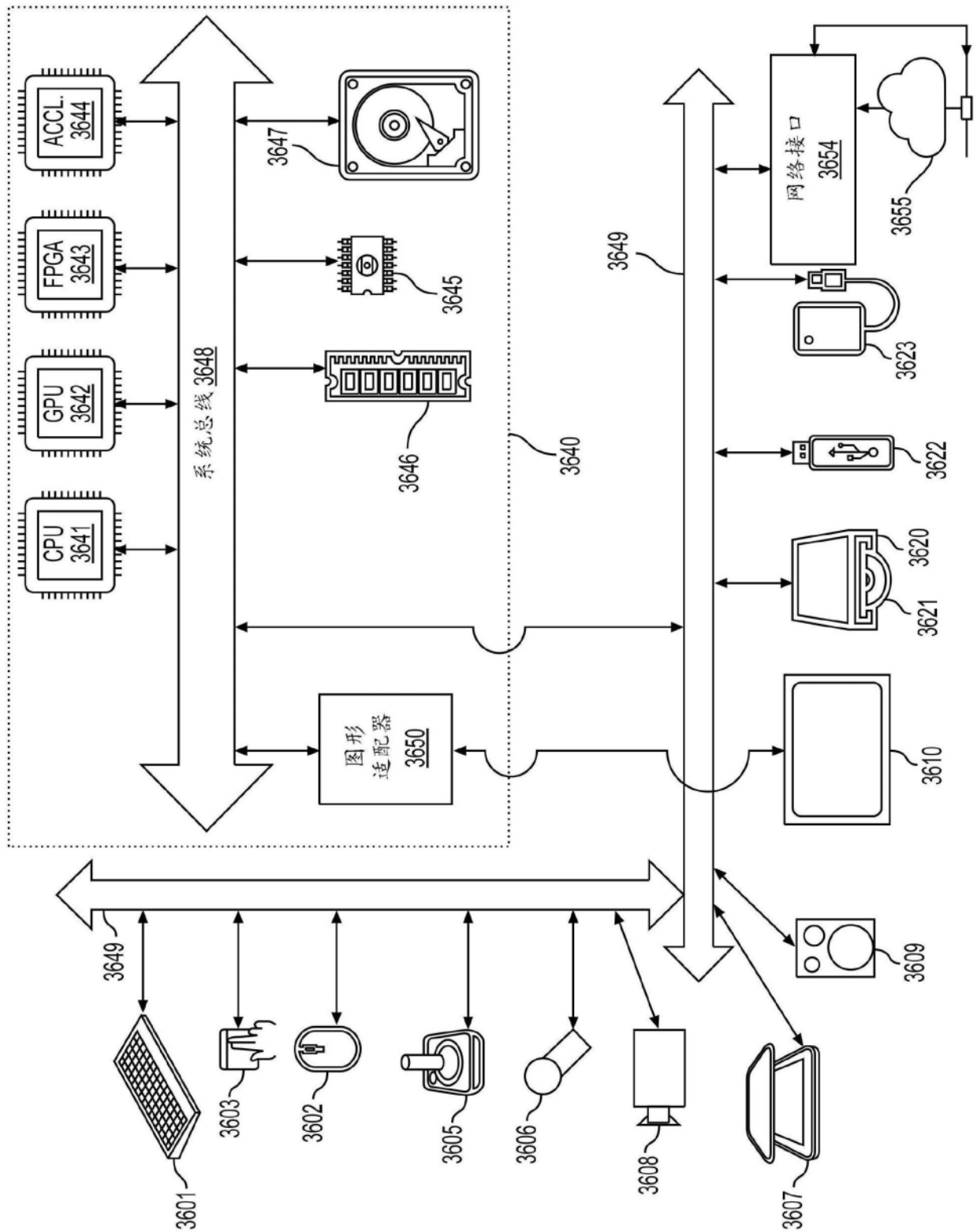


图36