



(12) 发明专利

(10) 授权公告号 CN 116368802 B

(45) 授权公告日 2024.06.21

(21) 申请号 202280007232.2

(74) 专利代理机构 北京派特恩知识产权代理有

(22) 申请日 2022.04.29

限公司 11270

(65) 同一申请的已公布的文献号

专利代理人 刘晖铭 胡春光

申请公布号 CN 116368802 A

(51) Int.CI.

H04N 19/124 (2006.01)

(43) 申请公布日 2023.06.30

H04N 19/13 (2006.01)

(30) 优先权数据

H04N 19/122 (2006.01)

63/250,166 2021.09.29 US

H03M 7/42 (2006.01)

17/710,778 2022.03.31 US

H04N 19/61 (2006.01)

(85) PCT国际申请进入国家阶段日

(56) 对比文件

2023.04.27

Byeongdoo Choi等 .AHG8: General

(86) PCT国际申请的申请数据

Constraints Information (GCI) flags for
operation range extensions, JVET-X0095-
v2. Joint Video Experts Team (JVET) of
ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29
24th Meeting, by teleconference, 6-15
October 2021.2021,全文.

PCT/US2022/072015 2022.04.29

审查员 赵慧敏

(87) PCT国际申请的公布数据

W02023/056105 EN 2023.04.06

(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

权利要求书1页 说明书37页 附图23页

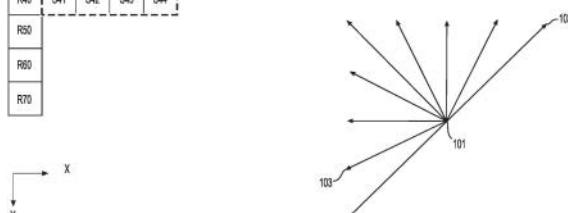
(54) 发明名称

视频解码方法、装置和存储介质

(57) 摘要

本公开的各方面提供了用于视频数据处理的方法和装置。在一些示例中，用于视频数据处理的装置包括处理电路。例如，处理电路确定第一语法元素，第一语法元素用于比特流中的编码视频数据的第一范围中的编码控制，第一语法元素与第二编码工具关联，第二编码工具替代第一编码工具用于残差编码中的Rice参数推导。响应于第一语法元素是指示在第一范围内禁用第二编码工具的第一值，处理器在不调用第二编码工具的情况下解码编码视频数据的第一范围，编码视频数据的第一范围包括编码视频数据的一个或多个第二范围。

	R01	R02	R03	R04	R05	R06	R07	R08	R09
R10	S11	S12	S13	S14					
R20	S21	S22	S23	S24					
R30	S31	S32	S33	S34					
R40	S41	S42	S43	S44					
R50									
R60									
R70									



1.一种视频解码的方法,其特征在于,用于对比特流解码,所述方法包括:

确定第一语法元素,所述第一语法元素用于比特流中的编码视频数据的第一范围中的编码控制,所述第一语法元素与第二编码工具关联,所述第二编码工具替代第一编码工具以用于残差编码中的Rice参数推导;以及

响应于所述第一语法元素是指示在所述第一范围内禁用所述第二编码工具的第一值,在不调用所述第二编码工具的情况下解码所述编码视频数据的第一范围,所述编码视频数据的第一范围包括所述编码视频数据的一个或多个第二范围;

所述第一语法元素在通用约束信息中,所述通用约束信息用于对解码器处的输出层集中的图片的编码控制;

响应于所述第一语法元素是第二值,从所述编码层视频序列CLVS的序列参数集SPS中确定第二语法元素的值,所述第二语法元素用于对在比特流中的编码层视频序列CLVS的编码控制,所述第二语法元素指示在所述编码层视频序列CLVS中启用/禁用所述第二编码工具;

响应于所述第二语法元素不存在于用于所述编码层视频序列CLVS的序列参数集SPS中,推断所述第二语法元素的值用于指示在所述编码层视频序列CLVS中禁用所述第二编码工具。

2.根据权利要求1所述的方法,其特征在于,所述第一语法元素的所述第一值指示在所述输出层集中的每个编码层视频序列CLVS中禁用所述第二编码工具。

3.根据权利要求1所述的方法,其特征在于,所述方法还包括:

将用于对所述比特流中的编码层视频序列CLVS的编码控制的第二语法元素约束为具有指示不调用所述第二编码工具用于解码所述编码层视频序列CLVS的值。

4.根据权利要求1至3中任一项所述的方法,其特征在于,所述确定第一语法元素,还包括:

响应于针对通用约束信息的语法结构中的语法元素在语法结构中指示针对通用约束信息的附加比特,从所述语法结构中解码所述第一语法元素。

5.根据权利要求1至3中任一项所述的方法,其特征在于,所述第二编码工具在所述解码器不支持的范围扩展中定义。

6.根据权利要求4所述的方法,其特征在于,所述第二编码工具是用于残差编码期间的绝对值的二值化的Rice参数推导。

7.一种用于视频解码的装置,其特征在于,用于对比特流解码,包括

处理器;和

存储指令的存储器,

所述处理器被配置为执行所述指令以实现权利要求1至6中任一项所述的方法。

8.一种存储指令的非暂态计算机可读存储介质,其特征在于,所述指令在由至少一个处理器执行时使所述至少一个处理器执行权利要求1至6中任一项所述的视频解码方法,以对比特流进行解码。

视频解码方法、装置和存储介质

[0001] 相关申请的交叉引用

[0002] 本公开要求于2022年3月31日提交的题为“TECHNIQUES FOR CONSTRAINT FLAG SIGNALING FOR RANGE EXTENSION WITH RESIDUAL RICE CODING EXTENSION”的美国专利申请第17/710778号的权益，前述美国专利申请要求于2021年9月29日提交的题为“TECHNIQUES FOR CONSTRAINT FLAG SIGNALING FOR RANGE EXTENSION WITH RESIDUAL RICE CODING EXTENSION”的第63/250166号美国临时申请的优先权的权益，在先申请的公开内容全部通过引用并入本文。

技术领域

[0003] 本公开描述了总体上与视频编码相关的实施例。

背景技术

[0004] 本文所提供的背景描述是出于总体上呈现本公开的内容的目的。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，并不表明其在本申请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] 视频编码和解码可使用带有运动补偿的帧间图片预测来执行。未压缩的数字视频可包括一系列图片，每个图片具有例如为 1920×1080 的亮度样本及相关联的色度样本的空间大小。该一系列图片可具有例如每秒60幅图片或60Hz的固定或可变的图片速率(还非正式地称为帧率)。未压缩的视频具有特定比特率要求。例如，每个样本8位的1080p60 4:2:0视频(在60Hz帧率下，具有 1920×1080 亮度样本分辨率)需要接近1.5Gbit/s的带宽。一小时的此类视频需要600GB以上的存储空间。

[0006] 视频编码和解码的一个目的可以是通过压缩来减少输入视频信号中的冗余。压缩可有助于减小上述带宽和/或存储空间需求，在一些情况下可减小两个数量级或大于两个数量级。可采用无损压缩和有损压缩以及它们的组合。无损压缩指的是可以从已压缩的原始信号中重建原始信号的精确副本的技术。当使用有损压缩时，已重建的信号可能与原始信号不同，但是原始信号和已重建的信号之间的失真足够小，以使已重建的信号可用于预期的应用。在视频的情况下，广泛采用有损压缩。可容忍的失真量取决于应用；例如，某些消费流式应用的用户相比电视分布应用的用户来说可以容忍更高的失真。可实现的压缩率可反映：更高的可允许/可容忍的失真可产生更高的压缩率。

[0007] 视频编码器和解码器可利用来自多个宽泛类别的技术，这些技术包括例如运动补偿、变换、量化和熵编码。

[0008] 视频编解码器技术可包括称为帧内编码的技术。在帧内编码中，在不参考来自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，图片在空间上细分成样本块。当所有样本块都以帧内模式编码时，该图片可以是帧内图片。帧内图片及其推导物(例如，独立解码器刷新图片)可用于重置解码器状态，因此可用作已编码视频码流和视频会话中的第一张图片，或者用作静止图像。帧内块的样本可进行变换，且可

以在熵编码之前对变换系数进行量化。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。在一些情况下,变换之后的DC值越小,且AC系数越小,则在给定的量化步长下表示熵编码之后的块所需的位越少。

[0009] 例如从诸如MPEG-2代编码技术已知的传统帧内编码不使用帧内预测。然而,一些更新的视频压缩技术包括基于例如在空间相邻的编码/解码期间获得的、按解码次序在数据块之前的周围样本数据和/或元数据来进行尝试的技术。在下文中这样的技术称为“帧内预测”技术。应注意,至少在一些情况下,帧内预测仅使用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] 帧内预测可以有许多不同形式。当在给定的视频编码技术中可使用不止一种这样的技术时,使用中的技术可以以帧内预测模式编码。在一些情况下,模式可具有子模式和/或参数,且可单独编码或包含在模式码字中。给定的模式/子模式/参数组合使用哪个码字,可能会影响通过帧内预测的编码效率增益,因此可能会影响用于将码字转换成码流的熵编码技术。

[0011] H.264引入了某种帧内预测模式,该帧内预测模式在H.265中得到完善,且在诸如联合探索模型(JEM)、下一代视频编码(VVC)和基准集(BMS)的更新的编码技术中进一步得到完善。可使用属于已经可用的样本的相邻样本值来形成预测块。相邻样本的样本值根据一定方向复制到预测块中。对使用中的方向的参考可以在码流中进行编码,或者可以对其本身进行预测。

[0012] 参考图1A,在右下方描绘了从H.265的33种可能的预测方向(对应于35个帧内模式的33个角模式)已知的9个预测方向的子集。箭头汇聚的点(101)表示正在被预测的样本。箭头表示样本的预测所沿的方向。例如,箭头(102)指示从在右上方、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)指示从在样本(101)的左下方、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,在左上方描绘了 4×4 个样本的正方形块(104)(由粗体虚线指示)。正方形块(104)包括16个样本,每个样本用“S”、其在Y维度上的位置(例如,行索引)和其在X维度上的位置(例如,列索引)来标记。例如,样本S21是在Y维度上(从顶部开始)的第二个样本和在X维度上(从左侧开始)的第一个样本。类似地,样本S44是在Y和X维度上、块(104)中的第四个样本。由于块的大小为 4×4 个样本,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、其相对于块(104)的Y位置(例如,行索引)和X位置(列索引)来标记。在H.264和H.265中,使用与正在重建的块相邻的预测样本;无需使用负值。

[0014] 帧内图片预测可通过根据用信号表示的预测方向从相邻样本复制参考样本值来开始。例如,假设已编码视频码流包括信令,该信令针对该块指示与箭头(102)一致的预测方向,即从在右上方、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,从同一个参考样本R05预测样本S41、S32、S23和S14。然后从参考样本R08预测样本S44。

[0015] 在一些情况下,可例如通过插值来组合多个参考样本的值,以计算参考样本;尤其是当方向无法以45度均匀分开时。

[0016] 随着视频编码技术继续发展,可能的方向的数量增加。在H.264(2003年)中,可表示九个不同的方向。在H.265(2013年)中,增加到33个方向,以及在本公开时,JEM/VVC/BMS可支持多达65个方向。已进行实验来识别最可能的方向,且熵编码中的一些技术用于以少

量比特表示那些可能的方向,对于不太可能的方向,接受一定的代价。此外,有时可以从已经解码的相邻块中使用的相邻方向来预测方向本身。

[0017] 图1B示出了示意图(180),其描绘了根据JEM的65个帧内预测方向,以说明随着时间的推移,预测方向的数量增加。

[0018] 已编码视频码流中表示方向的帧内预测方向位的映射,可能因视频编码技术的不同而不同;例如,其范围可以从预测方向简单直接映射到帧内预测模式,映射到码字,映射到涉及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,可存在某些方向,这些方向与某些其它方向相比,在统计上在视频内容中出现的可能性较小。由于视频压缩的目标是减少冗余,因此在一种设计良好的视频编码技术中,那些不太可能的方向相比更可能的方向来说,将由更多的位数表示。

[0019] 运动补偿可以是有损压缩技术,且可涉及以下技术:来自先前重建的图片或其部分(参考图片)的样本数据块在沿着由运动矢量(此后称为MV)指示的方向在空间上偏移之后,用于预测新重建的图片或图片部分。在一些情况下,参考图片可与当前正在重建的图片相同。MV可具有X和Y两个维度,或具有三个维度,第三个维度指示正在使用的参考图片(间接地,第三个维度可以是时间维度)。

[0020] 在一些视频压缩技术中,可根据其它MV,例如根据在空间上与正在重建的区域相邻的样本数据的另一区域相关的、且按解码次序在MV之前的其它MV来预测适用于样本数据的某个区域的该MV。这样做可大大减少对MV进行编码所需的数据量,从而消除冗余并增加压缩。MV预测可有效地工作,例如,由于在对从相机获得的输入视频信号(称为自然视频)进行编码时,存在以下统计可能性:比适用单个MV的区域更大的区域沿着相似的方向移动,因此在一些情况下,可使用从相邻区域的MV导出的相似运动矢量来预测该更大的区域。这使得被认为是用于给定区域的MV与根据周围MV所预测的MV相似或相同,进而在熵编码之后,该MV可以用比直接对MV进行编码时使用的位数更少的位数来表示。在一些情况下,MV预测可以是无损压缩从原始信号(即:样本流)中导出的信号(即:MV)的示例。在其它情况下,例如由于根据多个周围MV计算预测值时出现舍入误差,使得MV预测本身可以是有损的。

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

[0022] 参考图2,当前块(201)包括在运动搜索过程期间已由编码器找到的样本,可根据已产生空间偏移的相同大小的先前块来预测该样本。可以从与一个或多个参考图片相关联的元数据中导出MV,而非直接对该MV进行编码,例如使用与被标记为A0、A1和B0、B1、B2(分别对应202到206)的五个周围样本中的任一样本相关联的MV,(按解码次序)从最近的参考图片中导出该MV。在H.265中,MV预测可使用来自相邻块正在使用的相同参考图片的预测值。

发明内容

[0023] 本公开的各方面提供了用于视频数据处理的方法和装置。在一些示例中,用于视频数据处理的装置包括处理电路。例如,处理电路确定第一语法元素,第一语法元素用于对比特流中的编码视频数据的第一范围中的编码控制。第一语法元素与第二编码工具关联,

第二编码工具替代第一编码工具用于残差编码中的Rice参数推导。响应于所述第一语法元素是指示在所述第一范围内禁用所述第二编码工具的第一值,处理电路在不调用所述第二编码工具的情况下解码编码视频数据的第一范围,编码视频数据的第一范围包括编码视频数据的一个或多个第二范围。

[0024] 在一些实施例中,第一语法元素在通用约束信息中,该通用约束信息用于对输出层集中的图片的编码控制。在一些示例中,第一语法元素的第一值指示在输出层集中的每个编码层视频序列(CLVS)中禁用第二编码工具。

[0025] 在一些示例中,处理电路可以将用于对比特流中的编码层视频序列(CLVS)的编码控制的第二语法元素约束为具有指示不调用第二编码工具用于解码编码层视频序列(CLVS)的值。

[0026] 在一些实施例中,响应于第一语法元素是第二值,处理电路确定第二语法元素的值,所述第二语法元素用于对比特流中的编码层视频序列(CLVS)的编码控制,第二语法元素指示在所述编码层视频序列(CLVS)中启用/禁用第二编码工具。第二编码工具用于残差编码期间的绝对值的二值化的Rice参数推导。

[0027] 在一些示例中,处理电路响应于第二语法元素不存在于用于编码层视频序列(CLVS)的序列参数集(SPS)中,推断第二语法元素的值用于指示在编码层视频序列(CLVS)中禁用第二编码工具。

[0028] 在一些示例中,响应于针对通用约束信息的语法结构中的语法元素在语法结构中指示针对通用约束信息的附加比特,处理电路从语法结构中解码第一语法元素。

[0029] 在一些示例中,第二编码工具在标准的范围扩展中定义,且所述装置不支持该范围扩展。

[0030] 本公开的各方面还提供了一种存储指令的非暂态计算机可读存储介质,所述指令在由计算机用于视频解码执行时使计算机执行用于视频解码的方法。

附图说明

[0031] 根据下面的详细描述和附图,所公开的主体的其他特征、性质以及各种优点将更加明显,在附图中:

[0032] 图1A是帧内预测模式的示例性子集的示意性图示。

[0033] 图1B是示例性帧内预测方向的图示。

[0034] 图2是一个示例中的当前块和其周围的空间合并的候选者的示意图。

[0035] 图3是根据一个实施例的通信系统(300)的简化框图的示意图。

[0036] 图4是根据一个实施例的通信系统(400)的简化框图的示意图。

[0037] 图5是根据一个实施例的解码器的简化框图的示意图。

[0038] 图6是根据一个实施例的编码器的简化框图的示意图。

[0039] 图7示出了根据另一个实施例的编码器的框图。

[0040] 图8示出了根据另一个实施例的解码器的框图。

[0041] 图9示出了根据本公开一些实施例的用于发信令通知自适应分辨率改变(adaptive resolution change, ARC)参数的一些示例。

[0042] 图10示出了用于映射上采样或下采样因子、码字和Ext-Golomb码的表(1000)的示

例。

- [0043] 图11示出了根据本公开一些实施例的ARC参数信令的一些示例。
- [0044] 图12示出了一些示例中的一组PTL语法元素的语法结构示例。
- [0045] 图13示出了一些示例中的通用约束信息的语法结构示例。
- [0046] 图14A至图14B示出了根据本公开一些实施例的包括PTL语法结构和通用约束信息语法结构的PTL信息的示例。
- [0047] 图15A至图15B示出了根据本公开实施例的通用约束信息语法结构的示例。
- [0048] 图16示出了根据本公开一些实施例的通用约束信息的语法结构。
- [0049] 图17示出了根据本公开一些实施例的序列参数集 (SPS) 范围扩展的语法结构示例。
- [0050] 图18示出了概述根据本公开实施例的处理的流程图。
- [0051] 图19示出了概述根据本公开实施例的处理的流程图。
- [0052] 图20是根据实施例的计算机系统的示意图。

具体实施方式

[0053] 图3示出了根据本公开的一个实施例的通信系统(300)的简化框图。通信系统(300)包括多个终端设备，该多个终端设备可通过例如网络(350)彼此通信。例如，通信系统(300)包括通过网络(350)互连的第一终端设备对(310)和(320)。在图3的示例中，第一终端设备对(310)和(320)执行单向数据传输。例如，终端设备(310)可以对视频数据(例如，由终端设备(310)采集的视频图片流)进行编码，以通过网络(350)传输到另一终端设备(320)。已编码视频数据可以以一个或多个已编码视频码流的形式传输。终端设备(320)可以从网络(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图片，以及根据所恢复的视频数据显示视频图片。单向数据传输可以在媒体服务等应用中是常见的。

[0054] 在另一示例中，通信系统(300)包括执行已编码视频数据的双向传输的第二终端设备对(330)和(340)，该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，终端设备(330)和(340)中的每个终端设备可以对视频数据(例如，由终端设备采集的视频图片流)进行编码，以通过网络(350)传输到终端设备(330)和(340)中的另一终端设备。终端设备(330)和(340)中的每个终端设备还可接收由终端设备(330)和(340)中的另一终端设备传输的已编码视频数据，且可以对已编码视频数据进行解码以恢复视频图片，以及可根据所恢复的视频数据在可访问的显示设备上显示视频图片。

[0055] 在图3的示例中，终端设备(310)、终端设备(320)、终端设备(330)和终端设备(340)可示出为服务器、个人计算机和智能电话，但是本公开的原理可不限于此。本公开的实施例在膝上型计算机、平板电脑、媒体播放器和/或专用视频会议设备上找到应用。网络(350)表示在终端设备(310)、终端设备(320)、终端设备(330)和终端设备(340)之间传送已编码视频数据的任何数量的网络，包括例如有线(连线的)和/或无线通信网络。通信网络(350)可以在电路交换信道和/或分组交换信道中交换数据。代表性的网络包括电信网络、局域网、广域网和/或互联网。出于本讨论的目的，除非在下文中予以说明，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。

[0056] 作为用于所公开的主题的应用的示例，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。所公开的主题可同等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。

[0057] 流式传输系统可包括采集子系统(413),采集子系统(413)可包括诸如数码相机的视频源(401),视频源(401)创建例如未压缩的视频图片流(402)。在一个示例中,视频图片流(402)包括由数码相机拍摄的样本。相较于已编码视频数据(404)(或已编码视频码流),被描绘为粗线以强调高数据量的视频图片流(402)可由电子设备(420)处理,电子设备(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的主题的各方面。相较于视频图片流(402),被描绘为细线以强调较低数据量的已编码视频数据(404)(或已编码视频码流)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)可包括例如电子设备(430)中的视频解码器(410)。视频解码器(410)对已编码视频数据的传入副本(407)进行解码,且产生可以在显示器(412)(例如,显示屏)或其它呈现设备(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/压缩标准对已编码视频数据(404)、(407)和(409)(例如,视频码流)进行编码。这些标准的示例包括ITU-T H.265建议书。在一个示例中,正在开发的视频编码标准非正式地称为下一代视频编码(VVC)。所公开的主题可用于VVC的上下文中。

[0058] 应注意,电子设备(420)和电子设备(430)可包括其它组件(未示出)。例如,电子设备(420)可包括视频解码器(未示出),且电子设备(430)还可包括视频编码器(未示出)。

[0059] 图5示出了根据本公开的一个实施例的视频解码器(510)的框图。视频解码器(510)可包括在电子设备(530)中。电子设备(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的示例中的视频解码器(410)。

[0060] 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,一次可以对一个已编码视频序列进行解码,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。可以从信道(501)接收已编码视频序列,信道(501)可以是硬件/软件链路,其通向存储已编码视频数据的存储设备。接收器(531)可接收可转发到它们各自的使用实体(未描绘)的已编码视频数据和其它数据,例如已编码音频数据和/或辅助数据流。接收器(531)可以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)和熵解码器/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。在其它应用中,缓冲存储器(515)可位于视频解码器(510)的外部(未描绘)。而在另一些其它应用中,在视频解码器(510)的外部可设置缓冲存储器(未描绘)以例如防止网络抖动,且在视频解码器(510)的内部可设置另一附加的缓冲存储器(515)以例如处理播出定时。当接收器(531)从具有足够带宽和可控性的存储/转发设备或从等时同步网络接收数据时,可能不需要缓冲存储器(515),或可以将缓冲存储器做得较小。为了在诸如互联网等业务分组网络上使用,可能需要缓冲存储器(515),缓冲存储器(515)可相对较大,有利地可具有自适应大小,且可至少部分地在操作系统或视频解码器(510)外部的类似元件(未描绘)中实现。

[0061] 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,以及用于控制诸如呈现设备(512)(例如,显示屏)之类的呈现设备的潜在信息,该呈现设备不是电子设备(530)的整体部分,但是可耦接到电子设备(530),如图5所示。用于呈现设备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(VUI)参数集片段(未描绘)的形式。解析器(520)可以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(Huffman)编码、具有或不具有上下文敏感度的算术编码等。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(GOP)、图片、图块、切片、宏块、编码单元(CU)、块、变换单元(TU)、预测单元(PU)等。解析器(520)还可以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。

[0062] 解析器(520)可以对从缓冲存储器(515)接收的视频序列执行熵解码/解析操作,从而创建符号(521)。

[0063] 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)的类型以及其他因素,符号(521)的重建可涉及多个不同的单元。涉及哪些单元以及涉及方式可由解析器(520)通过从已编码视频序列解析的子群控制信息来控制。为了清楚起见,未描绘解析器(520)与下文的多个单元之间的此类子群控制信息流。

[0064] 除了已提及的功能块之外,视频解码器(510)可以在概念上细分成如下文所描述的多个功能单元。在商业约束下运行的实际实现方式中,这些单元中的许多单元彼此紧密交互且可至少部分地彼此集成。然而,出于描述所公开的主题的目的,在概念上细分成下文的多个功能单元是合适的。

[0065] 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,包括要使用哪种变换、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,样本值可输入到聚合器(555)中。

[0066] 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片的预测信息,但是可使用来自当前图片的先前重建部分的预测信息的块。此类预测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)使用从当前图片缓冲器(558)提取的周围已重建信息来生成大小和形状与正在重建的块相同的块。例如,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。

[0067] 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的块。在这种情况下,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。在根据属于块的符号(521)对所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,称为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)内的地址提取预测样本可受到运动矢量控制,该运动矢量可以以符号(521)的形式提供给运动补偿预测单元(553)使用,符号(521)可具有例如X分量、Y分量和参考图片分量。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。

[0068] 聚合器(555)的输出样本可经受环路滤波器单元(556)中的各种环路滤波技术。视频压缩技术可包括环路内滤波器技术,该环路内滤波器技术受控于包括在已编码视频序列(还称为已编码视频码流)中且可作为来自解析器(520)的符号(521)而用于环路滤波器单元(556)的参数,然而,视频压缩技术还可响应于在对已编码图片或已编码视频序列的先前(按解码次序)部分进行解码期间获得的元信息,以及响应于先前重建且经过环路滤波的样本值。

[0069] 环路滤波器单元(556)的输出可以是样本流,该样本流可输出到呈现设备(512)以及存储在参考图片存储器(557)中以用于将来的帧间图片预测。

[0070] 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的预测。例如,一旦对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)可变成参考图片存储器(557)的一部分,且可以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。

[0071] 视频解码器(510)可根据诸如ITU-T H.265建议书的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。具体而言,配置文件可以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该配置文件下可供使用的仅有工具。为了合规性,还可能需要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、最大重建采样率(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(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。

[0072] 在一个实施例中,接收器(531)可以在接收已编码视频时接收附加(冗余)数据。附加数据可以被包括作为已编码视频序列的一部分。附加数据可由视频解码器(510)使用来对数据进行适当解码和/或更准确地重建原始视频数据。附加数据可采用例如时间、空间或信噪比(SNR)增强层、冗余切片、冗余图片、前向纠错码等形式。

[0073] 图6示出了根据本公开的一个实施例的视频编码器(603)的框图。视频编码器(603)包括在电子设备(620)中。电子设备(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的示例中的视频编码器(403)。

[0074] 视频编码器(603)可以从视频源(601)(并非图6的示例中的电子设备(620)的一部分)接收视频样本,视频源(601)可采集将由视频编码器(603)编码的视频图像。在另一示例中,视频源(601)是电子设备(620)的一部分。

[0075] 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,该数字视频样本流可具有任何合适的位深(例如:8位、10位、12位.....)、任何色彩空间(例如BT.601YCrCB,RGB.....)和任何合适的采样结构(例如YCrCb 4:2:0,YCrCb 4:4:4)。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设备。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。视频数据可作为多个单独的图片来提供,当按顺序观看时,这些图片被赋予运动。图片本身可构建为空间

像素阵列，其中取决于所使用的采样结构、色彩空间等，每个像素可包括一个或多个样本。本领域的技术人员可容易地理解像素和样本之间的关系。下文侧重于描述样本。

[0076] 根据一个实施例，视频编码器(603)可实时地或在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并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所述的其它功能单元。为了清楚起见，图中未描绘耦接。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……)、图片大小、图片群组(GOP)布局、最大运动矢量搜索范围等。控制器(650)可配置成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)。

[0077] 在一些实施例中，视频编码器(603)配置成在编码环路中进行操作。作为过于简化的描述，在一个示例中，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来创建符号，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)重建符号以用类似于(远程)解码器还可创建样本数据的方式创建样本数据(因为在所公开的主题中考虑的视频压缩技术中，符号与已编码视频码流之间的任何压缩是无损的)。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和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)还用于一些相关技术。

[0078] “本地”解码器(633)的操作可与诸如已在上文结合图5详细描述的视频解码器(510)的“远程”解码器相同。然而，另外简要参考图5，由于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/解码成已编码视频序列，因此包括缓冲存储器(515)和解析器(520)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)中实现。

[0079] 在这一点上可以进行的观察是，除了存在于解码器中的解析/熵解码之外的解码器技术，以相同或基本上相同的功能形式存在于对应的编码器中。因此，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。在某些区域中，在下文提供更详细的描述。

[0080] 在操作期间，在一些示例中，源编码器(630)可执行运动补偿预测编码，通过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，该运动补偿预测编码对输入图片进行预测性编码。以这种方式，编码引擎(632)对输入图片的像素块和参考图片的像素块之间的差异进行编码，该参考图片可被选作该输入图片的预测参考。

[0081] 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图片的已编码视频数据进行解码。编码引擎(632)的操作有利地可以是有损过程。当已编码视频数据可以在视频解码器(图6未示出)中被解码时，已重建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，该解码过程可由视频解码器对参考图片执行，且可使已重建参考图片存储在参考图片存储器(634)中。以这种方式，视频编码器(603)可以在本地存储已重建参考图片的副本，该副本与将由远端视频解码器获得的已重建参考图片具有共同内容(不存在传输误差)。

[0082] 预测器(635)可针对编码引擎(632)执行预测搜索。即,对于待编码的新图片,预测器(635)可以在参考图片存储器(634)中搜索可用作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,如由预测器(635)获得的搜索结果所确定的,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。

[0083] 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。

[0084] 可以在熵编码器(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、算术编码等的技术来对各种功能单元生成的符号进行无损压缩,从而将该符号转换成已编码视频序列。

[0085] 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,通信信道(660)可以是通向可存储已编码视频数据的存储设备的硬件/软件链路。传输器(640)可以将来自视频编码器(603)的已编码视频数据与待传输的其它数据合并,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)。

[0086] 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)可以向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,通常可以将图片分配为以下任一种图片类型:

[0087] 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(“IDR”)图片。本领域的技术人员了解I图片的这些变体及其相应的应用和特征。

[0088] 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,该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。

[0089] 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,该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的元数据来用于重建单个块。

[0090] 源图片通常可以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)块进行预测性编码,其它(已编码)块由应用于块的相应图片的编码分配来确定。例如,I图片的块可进行非预测性编码,或者I图片的块可参考同一图片的已编码块来进行预测性编码(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间预测来进行预测性编码。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间预测来进行预测性编码。

[0091] 视频编码器(603)可根据诸如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性编码操作。因此,已编码视频数据可符合所使用的视频编码技术或标准指定的语法。

[0092] 在一个实施例中,传输器(640)可以在传输已编码视频时传输附加数据。源编码器

(630) 可包括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、诸如冗余图片和切片的其它形式的冗余数据、SEI消息、VUI参数集片段等。

[0093] 所采集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(通常简化为帧内预测)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。在一个示例中,将正在编码/解码的特定图片分成块,正在编码/解码的特定图片称为当前图片。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,可通过称为运动矢量的矢量对当前图片中的块进行编码。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,运动矢量可具有识别参考图片的第三个维度。

[0094] 在一些实施例中,双向预测技术可用于帧间图片预测。根据该双向预测技术,使用两个参考图片,例如按解码次序在视频中的当前图片之前(但是按显示次序可能分别是过去和将来)的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来对当前图片中的块进行编码。可通过第一参考块和第二参考块的组合来预测该块。

[0095] 此外,合并模式技术可用于帧间图片预测以改善编码效率。

[0096] 根据本公开的一些实施例,诸如帧间图片预测和帧内图片预测的预测以块为单位来执行。例如,根据HEVC标准,将视频图片序列中的图片分成编码树单元(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。通常,CTU包括三个编码树块(CTB),这三个编码树块是一个亮度CTB和两个色度CTB。可以将每个CTU递归地以四叉树划分成一个或多个编码单元(CU)。例如,可以将 64×64 像素的CTU划分成一个 64×64 像素的CU,或4个 32×32 像素的CU,或16个 16×16 像素的CU。在一个示例中,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。根据时间和/或空间可预测性,将CU划分成一个或多个预测单元(PU)。通常,每个PU包括亮度预测块(PB)和两个色度PB。在一个实施例中,编码(编码/解码)中的预测操作以预测块为单位来执行。使用亮度预测块作为预测块的示例,预测块包括针对像素的值(例如,亮度值)的矩阵,所述的像素例如是 8×8 像素、 16×16 像素、 8×16 像素、 16×8 像素等。

[0097] 图7示出了根据本公开的另一实施例的视频编码器(703)的图。视频编码器(703)配置成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。在一个示例中,视频编码器(703)用于代替图4的示例中的视频编码器(403)。

[0098] 在HEVC示例中,视频编码器(703)接收用于处理块的样本值的矩阵,该处理块例如是 8×8 样本的预测块等。视频编码器(703)使用例如率失真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最佳地对处理块进行编码。当在帧内模式中对处理块进行编码时,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;以及当在帧间模式或双向预测模式中对处理块进行编码时,视频编码器(703)可分别使用帧间预测或双向预测技术以将处理块编码到已编码图片中。在一些视频编码技术中,合并模式可以是帧间图片预测子模式,其中,在不借助预测器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器导出运动矢量。在一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。在一个示例中,视频编码器(703)包括其它组件,例如用于确定处理块的模

式的模式决策模块(未示出)。

[0099] 在图7的示例中,视频编码器(703)包括如图7所示的耦接在一起的帧间编码器(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)。

[0100] 帧间编码器(730)配置成接收当前块(例如,处理块)的样本,将该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)进行比较,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来计算帧间预测结果(例如,已预测块)。在一些示例中,参考图片是基于已编码视频信息解码的已解码参考图片。

[0101] 帧内编码器(722)配置成接收当前块(例如,处理块)的样本,在一些情况下将该块与同一图片中已编码的块进行比较,在变换之后生成量化系数,以及在一些情况下还生成帧内预测信息(例如,根据一个或多个帧内编码技术的帧内预测方向信息)。在一个示例中,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)。

[0102] 通用控制器(721)配置成确定通用控制数据,且基于该通用控制数据控制视频编码器(703)的其它组件。在一个示例中,通用控制器(721)确定块的模式,且基于该模式将控制信号提供给开关(726)。例如,当该模式是帧内模式时,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并将帧内预测信息包括在码流中;以及当该模式是帧间模式时,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并将帧间预测信息包括在码流中。

[0103] 残差计算器(723)配置成计算所接收的块与从帧内编码器(722)或帧间编码器(730)选择的预测结果之间的差(残差数据)。残差编码器(724)配置成基于残差数据进行操作,以对残差数据进行编码来生成变换系数。在一个示例中,残差编码器(724)配置成将残差数据从空间域转换到频域,并生成变换系数。变换系数随后经受量化处理以获得量化变换系数。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)配置成执行逆变换,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。例如,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。在一些示例中,适当地处理已解码块以生成已解码图片,且已解码图片可以在存储器电路(未示出)中缓冲并用作参考图片。

[0104] 熵编码器(725)配置成将码流格式化以包括已编码块。根据诸如HEVC标准的合适标准,熵编码器(725)配置成包括各种信息。在一个示例中,熵编码器(725)配置成将通用控制数据、所选预测信息(例如,帧内预测信息或帧间预测信息)、残差信息和其它合适的信息包括在码流中。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。

[0105] 图8示出了根据本公开的另一实施例的视频解码器(810)的图。视频解码器(810)配置成接收作为已编码视频序列的一部分的已编码图片,且对已编码图片进行解码以生成已重建图片。在一个示例中,视频解码器(810)用于代替图4的示例中的视频解码器(410)。

[0106] 在图8的示例中,视频解码器(810)包括如图8所示的耦接在一起的熵解码器(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)。

[0107] 熵解码器(871)可配置成根据已编码图片来重建某些符号,这些符号表示构成已编码图片的语法元素。此类符号可包括例如对块进行编码的模式(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)、可识别分别供帧内解码器(872)或帧间解码器(880)使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)、呈例如量化变换系数形式的残差信息等。在一个示例中,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给帧间解码器(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给帧内解码器(872)。残差信息可经受逆量化并提供给残差解码器(873)。

[0108] 帧间解码器(880)配置成接收帧间预测信息,且基于该帧间预测信息生成帧间预测结果。

[0109] 帧内解码器(872)配置成接收帧内预测信息,且基于该帧内预测信息生成预测结果。

[0110] 残差解码器(873)配置成执行逆量化以提取解量化的变换系数,且处理该解量化的变换系数,以将残差从频域转换到空间域。残差解码器(873)还可能需要某些控制信息(用以包括量化器参数(QP)),该信息可由熵解码器(871)提供(未描绘数据路径,因为这仅仅是低数据量控制信息)。

[0111] 重建模块(874)配置成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,视情况而定)以形成已重建块,已重建块可以是已重建图片的一部分,已重建图片进而可以是已重建视频的一部分。应注意,可执行诸如去块操作等其它合适的操作来改善视觉质量。

[0112] 应注意,可使用任何合适的技术来实现视频编码器(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)。在一个实施例中,可使用一个或多个集成电路来实现视频编码器(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现视频编码器(403)、视频编码器(603)和视频编码器(603)以及视频解码器(410)、视频解码器(510)和视频解码器(810)。

[0113] 本公开的各方面提供了编码工具的控制技术和在编码视频流中具有约束标志的功能。

[0114] 根据本公开的一方面,比特流中的图片尺寸可以保持不变或者可以改变。在一些相关示例中,视频编码器和解码器可以在给定的图片尺寸上操作,该给定的图片尺寸针对编码视频序列(CVS)、画面组(group of pictures, GOP)或类似的多图片时间帧定义并保持恒定。在一个示例中,例如在MPEG-2中,已知系统设计仅在I图片处依赖于诸如场景活动性的因素来改变水平分辨率(并因此改变图片尺寸),因此定义图片尺寸并且图片尺寸通常针对GOP保持恒定。在CVS内对参考图片进行重采样以使用不同分辨率是已知的,例如,可以从ITU-T Rec.H.263Annex P中获知。然而,CVS中的图片尺寸不改变,仅参考图片被重采样,导致可能使用图片画布的仅一部分(例如,在下采样情况下),或者被捕获的场景的仅一部分(例如,在上采样情况下)。在一些示例中,例如在H.263Annex Q中,允许在每个维度(例如,

向上或向下)中以系数2对单个宏块重采样,但是,图片尺寸保持不变。当宏块的尺寸可以固定时,例如在H.263中,因此,宏块的尺寸不需要用信号通知。

[0115] 在一些相关示例中,预测图片中的图片尺寸可以改变。在一个示例例如VP 9中,允许参考图片重采样和针对整幅图片改变分辨率。在一些示例中(例如,Hendry等人的“On adaptive resolution change (ARC) for VVC”,Joint Video Team document JVET-M0135-v1,Jan 9-19,2019,其全部内容并入本文),允许将整个参考图片重采样为不同的分辨率(例如,更高的分辨率或更低的分辨率)。不同的候选分辨率可以编码在序列参数集(SPS)中,并且可以由图片参数集(picture parameter set,PPS)中的按照图片语法元素来引用。

[0116] 根据本公开的一方面,源视频可以通过分层编码来压缩,分层编码可以将图片编码成包括具有不同质量(例如不同分辨率)的一个或多个层的比特流。该比特流可以具有明确指出在解码器侧可以输出哪个层或哪些层(或层集)的语法元素。可以将要输出的层集定义为输出层集。例如,在支持多层和可扩展性的视频编解码器中,可以在视频参数集(video parameter set,VPS)中用信号通知一个或多个输出层集。可以在VPS、解码参数集(DPS)(在一些示例中DPS可能被称为解码能力信息(DCI))、SPS、PPS、SEI消息等中发信令通知语法元素,该语法元素明确指出用于整个比特流或一个或多个输出层集的配置文件层级水平(profile tier level,PTL)。在PTL信息中,可以存在通用约束信息,该通用约束信息可以明确指出对编码工具或功能的约束。有效地表示并发信令通知针对各种编码工具和功能的约束信息是期望的。

[0117] 在一些示例中,术语“子图片”可用于指代例如样本、块、宏块、编码单元或类似实体的矩形排列,其在语义上被分组并且可以以改变的分辨率被独立编码。一个或多个子图片可以形成图片。一个或多个编码子图片可以形成编码图片。一个或多个子图片可以组合成图片,并且可以从图片中提取一个或多个子图片。在一些示例中,一个或多个编码子图片可以在压缩域中组合,而无需将样本水平转码为编码图片。在一些示例中,可以从压缩域中的编码图片中提取一个或多个编码子图片。

[0118] 在一些示例中,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可以称为自适应分辨率改变(ARC)。用于执行自适应分辨率改变的控制信息可以称为ARC参数。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、各种控制标志和/或类似的参数。

[0119] 在一些示例中,ARC的编码/解码以图片为单位,因此控制信息集(ARC参数)用于编码/解码单张且语义独立的编码视频图片。在一些示例中,ARC的编码/解码以子图片为单位,因此可以使用独立的ARC参数对图片中的多个子图片进行编码/解码。应当注意,可以使用各种技术来发信令通知ARC参数。

[0120] 图9示出了根据本公开一些实施例的用于发信令通知ARC参数的技术的示例(例如,选项)。编码效率、复杂度和架构可能因示例而异。视频编码标准或技术可以选择一个或多个示例或其他变体,用于发信令通知ARC参数。这些示例可以不是互斥的,并且可以基于应用需求、标准技术、编码器的选择和/或类似的因素来互换。

[0121] 根据本公开的一方面,ARC参数可以以各种方式提供为不同类别的ARC参数。在一些示例中,一类ARC参数包括在X维度和Y维度中分离或组合的上采样和/或下采样因子。在一个示例中,可以对一个或多个短语法元素进行编码,该一个或多个短语法元素可以指向

包括上采样和/或下采样因子的表。

[0122] 在一些示例中,一类ARC参数包括上采样和/或下采样因子,加上指示针对给定数量的图片以恒定速度放大和/或缩小的时间维度。在一个示例中,可以对一个或多个短语法元素进行编码,该一个或多个短语法元素可以指向包括上采样和/或下采样因子并增加了时间维度的表。

[0123] 在一些示例中,一类ARC参数包括组合或单独的输入图片、输出图片、参考图片、编码图片的以样本、块、宏块、CU或任何其他合适的粒度为单位的X维或Y维分辨率。在一些示例中,视频编码中使用多于一种分辨率(例如,用于输入图片的一种分辨率,用于参考图片的另一种分辨率),一组值(对应于分辨率中的一种)可以从另一组值(对应于分辨率中的另一种)中推断。例如,可以基于标志的使用来门控对值的确定。针对门控的标志的使用将在进一步的描述中详细描述。

[0124] 在一些示例中,一类ARC参数包括类似于H.263Annex P中使用的翘曲坐标的,以如上所述的合适粒度的翘曲坐标。H.263Annex P定义了编码翘曲坐标的有效方法。也可以设计其他有效的方式。例如,Annex P的翘曲坐标的可变长度可逆霍夫曼式编码可以由合适长度的二进制编码代替,其中二进制码字的长度可以从乘以一因子并偏移一个值的最大图片尺寸推导,以允许在最大图片尺寸的边界之外翘曲。

[0125] 在一些示例中,一类ARC参数包括上采样和/或下采样滤波器参数。在一个示例中,仅存在用于上采样和/或下采样的单个滤波器。在另一示例中,可以使用多个滤波器。在一些示例中,可以发信令通知滤波器参数以允许更大灵活性的滤波器设计。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,使用合适的熵编码技术明确指出滤波器系数的列表),可以通过根据上述任何机制和/或类似的机制用信令通知的上采样或下采样比率隐式地选择滤波器。

[0126] 在下面的描述中,上采样或下采样因子的有限集(在X维和Y维中使用的相同因子)用于示出通过码字用信令通知的ARC参数。在一些示例中,码字可以是可变长度编码的,例如,对视频编码规范(例如,H.264和H.265)中的某些语法元素使用Ext-Golomb码。

[0127] 图10示出了用于映射上采样或下采样因子、码字和Ext-Golomb码的表(1000)的示例。

[0128] 需要注意的是,可以根据视频压缩技术或标准中可用的上缩放(upscale)或下缩放(downscale)机制的应用和能力来设计其他类似的映射。在一些示例中,表1可以适当地扩展到附加值。需要注意的是,值可以由不同于Ext-Golomb码的熵编码机制来表示,例如,通过使用二进制编码来表示。在一个示例中,当重采样因子例如通过媒体感知网络元件(media-aware network element, MANE)对视频处理引擎(例如,编码器和解码器)外部感兴趣时,不同于Ext-Golomb码的熵编码机制可能具有特定的优点。在一些示例中,当需要分辨率不变时(例如初始/目标分辨率为表1中的1),可以选择短Ext-Golomb码(例如,仅在表1中示出单个比特),其相比于例如使用对于最常见的情况的二进制码可以具有编码效率优势。

[0129] 根据本公开的一方面,诸如表1的映射表可以是可配置的。例如,表1中的条目数量和相应的语义可以是全部或部分可配置的。在一些示例中,映射表的基本大纲在例如SPS或DPS的高水平参数集中传送。可替换地或附加地,在一些示例中,可以在视频编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来选择多个表中的一个。

[0130] 诸如如上所述编码的上采样或下采样因子的ARC信息可以包括在视频编码技术或标准语法中。需要注意的是,可以使用一个或多个码字来控制其他类别的ARC信息,例如上采样或下采样滤波器。在一些示例中,滤波器或其他数据结构需要相对大量的数据。

[0131] 参考图9,在示例(910)中,例如在H.263Annex P中,ARC信息(912)可以以四个翘曲坐标的形式,并且包括在图片头部(911)中,例如在H.263PLUSPTYPE(913)头部扩展中。示例(910)可以在情况i)图片头部可用以及情况ii)预期ARC信息的频繁变化时应用。然而,当使用例如在示例(910)中示出的H.263式的信令时,开销可能很高,并且由于图片头部可能具有瞬态性质,缩放因子可能在图片边界中不适用。

[0132] 参考图9,在示例(920)中,例如JVCET-M135-v1中,ARC参考信息(925)(例如,索引)可以放置在PPS(924)中,并且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)(926)。在一个示例中,表(926)位于SPS(927)内。通过在能力交换期间使用SPS作为互操作性协商点,可以证明将目标分辨率放置在SPS(927)中的表(926)中是合理的。通过适当的PPS(924)中的参考(例如,ARC参考信息(925)),分辨率可以在表(926)中的值(例如,分辨率1-3)的有限集内因图片而改变。

[0133] 图9还示出了其他技术,例如示例(930),(940)和(950),这些技术可用于在视频比特流中传送ARC信息。在相同的视频编码技术或标准中,这些技术可以单独使用,也可以适当组合使用。

[0134] 参考图9,在示例(930)中,诸如重采样因子(或缩放因子)的ARC信息(939)可能存在与诸如切片头部、GOB头部、瓦片头部、瓦片组头部等头部中。例如,图9示出了瓦片组头部(938)。当ARC信息(939)可以用少量比特(例如单个可变长度ue(v)或几比特的固定长度码字)编码时,可以使用示例(930)所示的技术。

[0135] 根据本公开的一方面,头部(例如,图9中的瓦片组头部(938),切片头部或瓦片头部)中直接具有ARC信息(939)可以具有额外的优点,因为ARC信息(939)可以适用于例如由对应的切片组(或切片、瓦片)表示的子图片,而不是整张图片。此外,在一个示例中,即使视频压缩技术或标准仅设想全图片自适应分辨率改变(与例如基于瓦片组的自适应分辨率改变相反),从错误恢复力的视点来看,示例(930)可以具有优于示例(910)的某些优点。

[0136] 参考图9,在示例(940)中,ARC信息(942)可以存在于例如PPS、头部参数集、瓦片参数集、自适应参数集(adaptation parameter set,APS)等的参数集(941)中。例如图9示出了APS(941)。在一些示例中,参数集(941)的范围可以不大于图片,例如,可以是图片、瓦片组等。ARC信息(例如ARC信息(942))的使用可以通过激活相关参数集(例如APS(941))来隐含。例如,当视频编码技术或标准仅考虑基于图片的ARC时,PPS或其等效物可以是合适的。

[0137] 参考图9,在示例(950)中,ARC参考信息(953)可以存在于如上所述的瓦片组头部(954)或类似的数据结构(例如,图片头部、切片头部、瓦片头部或GOP头部)中。图9中示出了瓦片组头部(954)作为示例。ARC参考信息(953)可以指在参数集(956)中可用的ARC信息(955)的子集,该子集例如为SPS、DPS等,该参数集具有超出单张图片的范围。SPS(956)在图9中示出作为示例。

[0138] 图11示出了根据本公开一些实施例的ARC参数信令的一些示例。图11示出了视频编码标准中使用的语法图示例。在一个示例中,语法图的符号大致遵循C型(C-style)编程。粗体的行可以指示比特流中存在的语法元素,而非粗体的行可以指示控制流或变量的设

置。

[0139] 参考图11,瓦片组头部(1101)包括适用于图片的一部分(例如矩形部分)的头部的语法结构。在一个示例中,瓦片组头部(1101)可以有条件地包括可变长度的Exp-Golomb编码语法元素**dec_pic_size_idx**(1102)(以粗体示出)。可以基于例如由标志(1103)(例如,adaptive_pic_resolution_change_flag)表示的自适应分辨率来门控瓦片组头部(1101)中存在的语法元素(例如,dec_pic_size_idx(1102))。标志(1103)(例如,adaptive_pic_resolution_change_flag)的值没有以粗体示出,由此该标志在语法图中出现该标记的点处呈现在比特流中。可以用比特流内部或外部的高水平语法结构(例如,图11中的SPS(1110))发信令通知自适应分辨率是用于整张图片还是用于图片的一部分。

[0140] 参考图11,示出了SPS(1110)的摘录。SPS(1110)包括第一语法元素(1111),所述第一语法元素是标志(1111)(例如,adaptive_pic_resolution_change_flag)。当标志(1111)为真时,标记(1111)可以指示使用可能需要特定控制信息的自适应分辨率。在一个示例中,该特定控制信息基于标志(1111)的值有条件地存在,如瓦片组头部(1101)和SPS(1110)中的if()语句(1112)所示。

[0141] 当使用自适应分辨率时,例如图11中所示出的,可以编码以样本为单位的输出分辨率(或输出图片的分辨率)(1113)。在一个示例中,基于宽度分辨率(例如,output_pic_width_in_luma_samples)和高度分辨率(例如,output_pic_height_in_luma_samples)对输出分辨率(1113)编码。在视频编码技术或标准中,可以定义对输出分辨率(1113)的值的特定限制。例如,水平定义可以限制总输出样本的数量(例如,output_pic_width_in_luma_samples和output_pic_height_in_luma_samples的乘积)。在一些示例中,视频编码技术或标准或外部技术或标准(例如,系统标准)可以限制宽度分辨率和/或高度分辨率(例如,宽度分辨率和/或高度分辨率可被2的幂整除)、宽度分辨率相对于高度分辨率的纵横比(例如,宽度分辨率相对于高度分辨率的比率是4:3或16:9)等的编号范围。在一个示例中,可以引入上述限制以便于硬件实现。

[0142] 在某些应用中,编码器可以指示解码器使用特定的参考图片尺寸,而不是隐式地假定某个尺寸是输出图片尺寸。例如,语法元素(1114)(例如,reference_pic_size_present_flag)对参考图片尺寸(1115)的条件性存在进行门控。在一个示例中,参考图片尺寸(1115)可以包括宽度(例如,reference_pic_width_in_luma_samples)和高度(例如,reference_pic_height_in_luma_samples)两者。

[0143] 图11中还示出了适用的解码图片宽度和高度的表。在一个示例中,表中条目的数量可以通过表指示(1116)(例如,语法元素**num_dec_pic_size_in_luma_samples_minus1**)来表示。“minus1”可以指对语法元素(1116)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5,则存在6个表条目。对于表中的每个条目,解码的图片宽度和高度被包含作为语法元素(1117)。

[0144] 由语法元素(1117)表示的表条目可以使用瓦片组头部(1101)中的语法元素**dec_pic_size_idx**(1102)来索引,并由此允许每个瓦片组使用不同的解码尺寸和缩放因子。

[0145] 根据本公开的一方面,某些视频编码技术或标准(例如,VP 9)可以通过结合时间的可缩放性实现特定形式的参考图片重采样来保证空间的可缩放性。在一个实施例中,使用ARC-style技术将参考图片上采样到更高的分辨率,以形成空间增强层的基础。例如,可

以使用高分辨率的正常预测机制(例如,用于从参考图片的帧间预测的运动补偿预测)来细化上采样的图片以增加细节。

[0146] 在一些示例中,网络抽象层(network abstraction layer,NAL)单元头部中的值,例如时间ID字段,用于指示时间层信息以及空间层信息。使用NAL单元头部中的值来指示时间层信息和空间层信息两者可以使得能够在不修改的情况下将现存的选择转发单元(selected forwarding unit,SFU)用于可缩放的环境。例如,基于NAL单元头部时间ID值,可以为时间层选择转发创建和优化现存的SFU。然后,在一些示例中,现存的SFU可以用于空间可缩放性(例如,空间层的选择),而无需修改。在一些示例中,可以在编码图片尺寸和由NAL单元头部中的时间ID字段指示的时间层之间提供映射。

[0147] 根据本公开的一方面,可以使用包括配置文件、等级、水平以及通用约束信息的配置文件、层级和水平组合(profile, tier and level combination, PTL)信息来指定编码比特流的一些特征。在一些示例中,配置文件定义比特流的特征的子集,例如颜色再现、分辨率、附加视频压缩等。视频编解码器可以定义各种配置文件,例如基准配置文件(例如,具有低压缩比率的简单配置文件)、高配置文件(具有高压缩比率的复杂配置文件)、主(Main)配置文件(例如,在基准配置文件和高配置文件之间具有中等压缩比率的配置文件,可以是默认的配置文件设置)等。

[0148] 此外,可以使用等级和水平来指定用于根据最大比特率、最大亮度采样速率、最大亮度图片尺寸、最小压缩比率、允许的最大切片数、允许的最大瓦片数等来定义比特流的某些约束。较低的等级比较高的等级受到更多的约束,且较低的水平比较高的水平受到更多的约束。在一个示例中,标准可以定义两种等级:主(Main)和高(High)。主等级比高等级低。这些等级用于处理最大比特率不同的应用。在一个示例中,主等级是为大多数应用设计的,而高等级是为要求非常苛刻的应用设计的。一种标准可以定义多个水平。水平是用于比特流的约束集。在一个示例中,对于水平4以下的水平,只允许主等级。在一些示例中,符合给定等级/水平的解码器需要有能力解码针对该给定等级/水平编码的以及针对所有较低等级/水平编码的所有比特流。

[0149] 通用约束信息可以包括关于视频源类型、编码工具和功能的约束信息。例如,约束标志可以指示在编码视频比特流中是否存在或使用帧间编码工具、帧内编码工具、DBF、熵编码、变换、分区(例如,切片、瓦片)、缓冲管理、随机存取(例如, IDR)、参数集(例如, SPS、PPS)和/或类似物。可以在参数集(例如, SPS、VPS或DCI)中发信令通知约束信息。可以在高水平语法结构(例如, SPS、VPS、DCI)中发信令通知约束标志。

[0150] 根据本公开的一些方面,PTL信息可以与范围(例如,比特流中的编码视频数据的一部分)关联。在一些示例中,可以为例如整个比特流、比特流的CVS、比特流的每个输出层集(output layer set,OLS)和/或类似物指定PTL信息,并可以在例如VPS、DPS、DCI、SPS、PPS、APS、GOP、序列、头部、SEI消息等高水平语法(high-level syntax,HLS)结构中发信令通知PTL信息。

[0151] 在一些示例中,高水平语法(HLS)是关于块水平来定义的。块水平编码工具可用于解码图片内的像素或样本,以重构图片。块水平编码工具可以包括在编码块的重构中使用的任何合适的编码工具,例如用于帧间预测的编码工具(或帧间编码工具)、用于帧内预测的编码工具(或帧内编码工具)、自适应环路滤波器(adaptive loop filter, ALF)、去块滤

波器(DBF)、熵编码、变换等。

[0152] 高水平语法(HLS)可以指定关于功能、系统接口、工具的图片水平的控制和缓冲控制等的信息。例如,HLS可以指定分区(例如,瓦片、切片、子图片)、缓冲管理、随机访问(例如, IDR、净随机访问(clean random access,CRA))、参数集(例如,VPS、SPS、PPS、APS)、参考图片重采样(reference picture resampling,RPR)、可缩放性和/或类似物。高水平语法可以高于块水平。

[0153] 控制信息可以具有合适的水平,例如SPS水平工具控制信息、PPS水平工具控制信息、序列水平控制信息、比特流水平控制信息和/或类似的。在一些示例中,PTL信息是控制信息的一部分,并可以作为HLS结构中的约束标志发信令通知,可以指示对应于HLS结构的范围内的工具的控制或约束。例如,用于PTL信息的约束标志可以在序列水平控制信息和比特流水平控制信息之一中提供。在一个示例中,如果某些工具被HLS结构中的约束标志禁用,则这些工具不被用于例如编码对应于HLS的范围内的块。

[0154] 图12和图13示出了根据本公开一些实施例的示例PTL信息。图12示出了PTL语法元素集的语法结构示例(1200)。图13示出了通用约束信息的语法结构示例(1300)。

[0155] 在图12中,PTL语法元素集可以包括general_profile_idc、general_tier_flag、general_level_idc、num_sub_profiles、general_sub_profile_idc、sublayer_level_present_flag、ptl_alignment_zero_bit和sublayer_level_idc。

[0156] 在图13中,通用约束信息可以包括多个约束标志。在一个示例中,约束标志(1305)(例如,intra_only_constraint_flag)等于1可以指示参数sh_slice_type应为I(即,切片是帧内切片)。参数sh_slice_type是切片头部中的参数,该参数在类型I、P和B中指定切片的编码类型。等于0的约束标志(1305)(例如,intra_only_constraint_flag)不对PTL信息范围内的所有编码图片施加约束(例如,sh_slice_type应为I),其他信息(例如,profile_idc)可以允许非帧内切片。在另一示例中,约束标志(1306)(例如,no_alf_constraint_flag)等于1可指示对于PTL信息范围内的所有CVS,sps_alf_enabled_flag等于0,并由此即使基于例如profile_idc允许使用自适应环路滤波器,也不使用自适应环路滤波器。约束标志(1306)(例如,no_alf_constraint_flag)等于0不施加上述约束。

[0157] 在另一示例中,如图13所示,可以在通用约束信息中发信令通知约束标志(1301)(例如,no_lossless_coding_tool_constraint_flag)。约束标志(1301)(例如,no_lossless_coding_tool_constraint_flag)等于1可指示在包括约束标志(1301)的PTL信息的范围内不能使用与无损编码相关的编码工具。约束标志(1301)(例如,no_lossless_coding_tool_constraint_flag)等于0不施加上述约束。

[0158] 在另一示例中,如图13所示,约束标志(1302)(例如,no_lossy_coding_tool_constraint_flag)可以在通用约束信息中发信令通知。约束标志(1302)(例如,no_lossy_coding_tool_constraint_flag)等于1可以指示在包括约束标志(1302)的PTL信息的范围内不能使用与无损编码相关的编码工具。约束标志(1302)(例如,no_lossy_coding_tool_constraint_flag)等于0不施加上述约束。

[0159] 在一个实施例中,当约束标志(1302)(例如,no_lossy_coding_tool_constraint_flag)等于1时,约束标志(1301)(例如,no_lossless_coding_tool_constraint_flag)可以不等于1。可替选地,当约束标志(1301)(例如,no_lossless_coding_tool_constraint_flag)

flag) 等于1时, 约束标志(1302)(例如,no_lossy_coding_tool_constraint_flag)可以不等于1。

[0160] 通用约束信息中的多个约束标志可以按特定顺序分类。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顺序。该顺序可以称为优先级顺序。可以在通用约束信息语法结构中呈现从高优先级到低优先级的顺序, 其中高优先级指示不使用工具(或机制)是高可能性的, 而低优先级指示不使用工具(或机制)是低可能性的。影响该顺序的附加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/或支持隔行扫描的工具)、该工具对编码器/解码器/实现方式的复杂度的影响等。

[0161] 图14A至图14B示出了根据本公开的一些实施例的PTL信息的示例, 该PTL信息包括PTL语法结构(也称为PTL括号)的语法结构示例(1410)和用于通用约束信息语法结构(也称为通用约束信息括号)的语法示例(1420)。在一些示例中, 可以用信令通知指示约束标志的数量(例如,num_available_constraint_flags)的语法元素。在一个示例中, 指示约束标志的数量的语法元素可以在PTL语法结构中用信令通知, 例如在图14A中所示的语法例子(1410)中由(1401)所示, 其可以在用于通用约束信息括号的语法示例(1420)之外。可选地, 指示约束标志的数量的语法元素可以在通用约束信息括号的起始位置(例如语法示例(1420)的起始位置)用信令通知。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个约束标志可能在通用约束信息语法结构中存在。此外, 其他的约束标志可能不存在, 并可以推断为等于特定值。N可以是非负整数。

[0162] 在一个实施例中, N值(例如,num_available_constraint_flags)的范围在0到约束标志的最大数量(例如, 参数MaxNumConstraintFlags的值)之间。约束标志的最大数量可以是任何正整数。约束标志(例如, MaxNumConstraintFlags)的最大数量的值可以预定义为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。可以选择对值N(例如,num_available_constraint_flags)的编码, 使得用于值N和约束标志的对应的熵编码表示加起来的数可以被8整除, 从而确保字节对齐。

[0163] 在一些示例中, 约束标志可以被分类为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, 并可以有相应的门控标志(gate flag)。对应约束信息组的门控标志可以指示对应的约束信息组中的约束标志是否可以存在。在一个示例中, 门控标志可以被称为约束组存在标志(constraint group present flag)。通常, 门控标志与相应的约束信息组关联, 并与相应的约束信息组中的约束标志关联。在一个实施例中, 门控标志门控相应约束信息组中的约束标志是否存在于约束信息中(或用信令通知)。例如, 如果对应的约束信息组的门控标志等于1, 则对应约束信息组的约束标志可以存在于例如通用约束信息中。如果对应的约束信息组的门控标志等于0, 则对应约束信息组的约束标志可以不存在于例如通用约束信息中。在一个示例中, 如果所有门控标志都等于0, 则不存在约束标志。

[0164] 约束标志可以具有不同的应对(copes)。例如, DCI中的约束标志的范围可以是编码视频比特流。VPS中约束标志的范围可以是具有多个层的CLVS。SPS中的约束标志的范围可以是单个CLVS。

[0165] 图15A至图15B示出了根据本公开实施例的通用约束信息语法结构(1500)的示例。通用约束信息语法结构(1500)包括代表通用约束信息的标志。具体地,通用约束信息语法结构(1500)包括一个或多个门控标志,例如图15A中的门控标志(1501)(例如,general_frame_structure_constraint_group_flag)、门控标志(1502)(例如,High_Level_Functionality_Constraint_Group_Flag)、门控标志(1503)(例如,scalability_constraint_group_flag)、门控标志(1504)(例如,partitioning_constraint_group_flag)、门控标志(1505)(例如,intra_coding_tool_constraint_group_flag)、门控标志(1506)(例如,inter_coding_tool_constraint_group_flag)、门控标志(1507)(例如,transfom_constraint_group_flag),门控标志(1508)(例如,inloop_filtering_constraint_group_flag)。如图15A所示,该一个或多个门控标志(例如,门控标志(1501)-(1508))可以存在于通用约束信息语法结构(1500)的起始位置处。

[0166] 门控标志(1501)(例如,general_frame_structure_constraint_group_flag)与约束信息组(1510)关联,并与约束信息组(1510)中的约束标志(1511)-(1514)关联。等于1的门控标志(1501)(例如,general_frame_structure_constraint_group_flag)可以指定可以存在约束信息组(1510)中的约束标志(1511)-(1514)。

[0167] 约束信息组(1510)(或约束标志(1511)-(1514))可以与输入源和帧封装(例如,封装的帧或投影的帧)相关。参考图15A,约束标志(1511)-(1514)对应general_non_packed_constraint_flag(1511)、general_frame_only_constraint_flag(1512)、general_non_projected_constraint_flag(1513)和general_one_picture_only_constraint_flag(1514)。否则,等于0的门控标志(例如,general_frame_structure_constraint_group_flag)(1501)可以指定约束信息组(1510)中的约束标志(1511)-(1514)可以不存在于通用约束信息语法结构(1500)中。

[0168] 此外,在一些示例中,如图15B所示,等于1的门控标志(1502)(例如,high_level_functional_constraint_group_flag)可以指定与约束信息组(1520)中的高水平功能(例如,参考图片重采样)相关的约束标志可以存在。否则,等于0的门控标志(1502)(例如,high_level_functional_constraint_group_flag)可以指定约束信息组(1520)中的约束标志可以不存在于通用约束信息语法结构(1500)中。

[0169] 回到图15A,等于1的门控标志(1503)(例如,scalability_constraint_group_flag)可以指定与可缩放性(例如层间预测)相关的约束标志可以存在。否则,与可伸缩性相关的约束标志可以不存在于通用约束信息语法结构(1500)中。

[0170] 等于1的门控标志(1504)(例如,partitioning_constraint_group_flag)可以指定与高水平分区(例如子图片或瓦片)相关的约束标志可以存在。否则,与高水平分区相关的约束标志可以不存在于通用约束信息语法结构(1500)中。

[0171] 等于1的门控标志(1505)(例如,intra_coding_tool_constraint_group_flag)可以指定与帧内编码(例如帧内预测)相关的约束标志可以存在。否则,与帧内编码相关的约束标志可以不存在于通用约束信息语法结构(1500)中。

[0172] 等于1的门控标志(1506)(例如,inter_coding_tool_constraint_group_flag)可以指定与帧间编码(例如,用于帧间预测的运动补偿)相关的约束标志可以存在。否则,与帧间编码相关的约束标志可以不存在于通用约束信息语法结构(1500)中。

[0173] 等于1的门控标志(1507)(例如,transfom_constraint_group_flag)可以指定与变换编码(例如,多个变换矩阵)相关的约束标志可以存在。否则,与变换编码相关的约束标志可以不存在于通用约束信息语法结构(1500)中。

[0174] 在一个实施例中,当所有门控标志(例如,图15A中的门控标志(1501)-(1508))等于0,则在通用约束信息语法结构(例如,通用约束信息语法结构(1500))中不存在约束标志。

[0175] 根据本公开的各方面,语法可以被设计成使得包括门控标志(例如,门控标志(1501)-(1508))、相关联的约束标志(例如,约束标志(1511)-(1512)和约束信息组(1520)中的约束标志)、附加控制信息和/或类似物的控制信息可以字节对齐,例如,标志的数量可以被8整除以保持字节对齐。在一个示例中,约束信息(例如,通用约束信息语法结构(1500))中的门控标志和约束标志的数量可被8整除。字节对齐机制可用于实现控制信息的字节对齐。参考图15B,语法(1530)(例如,while循环)可用于字节对齐。

[0176] 在一些实施例中,约束信息中(例如,在通用约束信息语法结构的起始位置处)存在例如偏移量(例如,使用语法元素constraint_info_offset[])的偏移量信息和例如长度(例如,使用语法元素constraint_info_length[])的长度信息,以帮助在与约束信息中的门控标志关联的相应的约束信息组中呈现约束标志。在一个实施例中,编码视频比特流中存在至少一个约束信息组中的一个或多个。对于约束信息组,偏移量和长度可以存在于约束信息组的约束信息中。偏移量可以指示对约束信息组中的第一约束标志的偏移量,且长度可以指示约束信息组中的约束标志的数量。在一些示例中,约束信息组的数量可以例如通过语法元素num_constraint_info_set明确地指示。num_constraint_info_set的值可以是大于或等于0的整数。当num_constraint_info_set的值为0时,通用约束信息语法结构中不会存在constraint_info_offset[]、constraint_info_length[]和约束标志。

[0177] 在一个实施例中,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])可以帮助在约束信息(例如,通用约束信息语法结构)中呈现约束信息组i(i是正整数)的约束标志。在一个示例中,当约束信息偏移量(例如,语法元素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,第五个、第六个和第七个约束标志与约束信息组i关联并且存在于约束信息中(例如,通用约束信息语法结构)。

[0178] 在一个示例中,游程长度编码可用于编码以预定顺序(或给定顺序)指定的约束标志。

[0179] 在一个实施例中,可以使用游程编码,其中,约束标志以预定顺序(或给定顺序)指定。替代直接对约束标志编码,适当编码的“skip”值列表可以指示等于零的约束标志,而接下来的约束标志被暗示为等于1。如果(i)约束标志的数量大且(ii)小比例的约束标志等于1,则上述游程编码会特别有效。

[0180] 在一个实施例中,编码视频比特流中存在至少一个约束信息组中的一个或多个。根据预定顺序发指令通知至少一个约束信息组中的一个或多个中的多个约束标志。由此,可以游程编解码(例如,游程编码或游程解码)该多个约束标志。此外,可以基于多个约束标志来确定用于编码块的子集的预测信息。

[0181] 在一个实施例中,门控标志的约束信息组中的至少一个约束标志包括根据预定顺

序发信令通知的多个约束标志。由此,可以游程编解码(例如,游程编码或游程解码)该多个约束标志。

[0182] 在一个实施例中,可以在视频编码标准(例如,VVC标准)、外部表等中指定约束标志的完整列表。在一个示例中,仅多个约束标志中的可用的一个或多个约束标志例如由以下中一个或多个来指示:编码视频流中存在的可用的约束标志的数量(例如,num_available_constraint_flags)、一个或多个门控标志(或约束组存在标志)、约束信息偏移量信息和约束信息长度信息等。

[0183] 在一个示例中,指定了约束标志的完整列表且该完整列表可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。该100个约束标志中的10个存在于用于CLVS的约束信息中,并由此可用于CLVS中的编码块的子集。该100个约束标志中的10个被称为10个可用的约束标志。在一个示例中,发信令通知可用的约束标志的数量(例如,10)。在一个示例中,该10个可用的约束标志位于两个约束信息组中,且由第一门控标志和第二门控标志来门控。因此,可以发信令通知第一门控标志和第二门控标志以指示该10个可用的约束标志。

[0184] 在一个示例中,发信令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])。发信令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]是15且语法元素constraint_info_length[0]是3,以及语法元素constraint_info_offset[1]是82且语法元素constraint_info_length[1]是7,由此指示完整列表(例如,100个约束标志)中的第15至第17个约束标志和第82至第88个约束标志在约束信息中可用或存在。

[0185] 在一个实施例中,可以使用适当的控制信息来组合用于约束标志的有效编码的各种技术(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样的技术的合适组合。可选地,也可以独立使用各种技术(或方法、实施例、示例)中的一种。约束标志可以分组。在特定的一个或多个组中,可以使用游程编码,而其他的一个或多个组可以采用直接的二进制编码。

[0186] 约束标志的最大数量(例如,MaxNumConstraintFlags)的值可以预定义为16、32、64、128等。

[0187] 约束标志的最大数量(例如,MaxNumConstraintFlags)的值可以由配置文件信息(例如,general_profile_idc或general_sub_profile_idc)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(1401))的范围可以由配置文件信息或版本信息来限制。例如,主配置文件(例如,其中MaxNumConstraintFlags=64)中的约束标志的数量(例如,num_available_constraint_flags(1401))的值可以在0到64的范围内,而高级配置文件(例如,其中MaxNumConstraintFlags=128)中的约束标志的数量(例如,num_available_constraint_flags(1401))的值可以在0到128的范围内。

[0188] 在一个实施例中,约束标志的数量(例如,num_available_constraint_flags)的值可以推断为等于由配置文件信息(例如general_profile_idc或general_sub_profile_idc)或编解码器版本信息预定义的值,从而在没有显式信令的情况下可以确定num_

available_constraint_flags的值。

[0189] 在一些实施例中,保留字节信息可以存在于通用约束信息语法结构中。例如,如图13所示,标志gci_num_reseved_bytes(1303)和gci_reseved_bytes[](1304)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可以指定保留约束字节数。在一个示例中,保留约束字节用于发信令通知附加标志(例如,附加约束标志)。标志gci_reserved_byte[]可以具有任何合适的值。

[0190] 在一个实施例中,gci_num_reserved_bytes的值可以由诸如general_profile_idc或general_sub_profile_idc之类的配置文件信息或编解码器版本信息来限制或确定。对于基本配置文件(或主配置文件),标志gci_num_reserved_bytes的值可以是0。对于扩展配置文件(或高级配置文件),gci_num_reserved_bytes的值可以大于0。

[0191] 在一些实施例中,可以在编码视频比特流中用信令通知字段序列标志。字段序列标志可以指示输出层中的图片是否用字段编码进行编码。在一些示例中,可以在SPS中使用句法元素sps_field_seq_flag发信令通知字段序列标志。在一个实施例中,标志sps_field_seq_flag可以存在于SPS中。等于1的标志sps_field_seq_flag可以指示CLVS传送表示字段的图片。等于0的标志sps_field_seq_flag可以指示CLVS传送表示帧的图片。

[0192] 图13中的通用约束信息语法结构中,可以存在标志general_frame_only_constraint_flag。等于1的标志general_frame_only_constraint_flag可指定输出层集的范围(例如01sInScope)传送表示帧的图片。等于0的标志general_frame_only_constraint_flag指定输出层集(例如01sInScope)的范围传送可以表示或可以不表示帧的图片。在一个实施例中,标志general_frame_only_constraint_flag指示输出层集中的图片是否用字段编码进行编码。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,为1)确定为假,标志general_frame_only_constraint_flag指示图片的子集不用字段编码进行编码。图片的子集可以在输出层集的一层中。

[0193] 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。

[0194] 在一个实施例中,标志pps_mixed_nalu_types_in_pic_flag可以存在于PPS中。等于1的标志pps_mixed_nalu_types_in_pic_flag可以指定涉及PPS的每个图片具有多于一个VCL NAL单元,且VCL NAL单元不具有相同的nal_unit_type值。等于0的标志pps_mixed_nalu_types_in_pic_flag可以指定涉及PPS的每个图片具有一个或多个VCL NAL单元,且涉及PPS的每个图片的VCL NAL单元具有相同的nal_unit_type值。图13中的通用约束信息语法结构中,可以存在标志no_mixed_nalu_types_in_pic_constraint_flag。等于1的标志no_mixed_nalu_types_in_pic_constraint_flag可以指定pps_mixed_nalu_types_in_pic_flag的值应等于0。等于0的标志no_mixed_nalu_types_in_pic_constraint_flag不施加这样的约束。

[0195] 在一个实施例中,例如如图13所示,标志general_one_picture_only_constraint_flag可以存在于通用约束信息语法结构中。等于1的general_one_picture_only_constraint_flag可以指定比特流中仅存在一张编码的图片。等于0的标志general_one_picture_only_constraint_flag不施加这样的约束。

[0196] 在一个实施例中,例如如图13所示,标志single_layer_constraint_flag可以存在于通用约束信息语法结构中。等于1的标志single_layer_constraint_flag可以指定sps_video_parameter_set_id应等于0。等于0的标志single_layer_constraint_flag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的值可等于1。

[0197] 在一个实施例中,例如如图13所示,标志all_layers_independent_constraint_flag可以存在于通用约束信息语法结构中。等于1的标志all_layers_independent_constraint_flag可以指定标志vps_all_independent_layers_flag可等于1。等于0的标志all_layers_independent_constraint_flag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的值可等于1。

[0198] 在一个实施例中,例如如图13所示,标志no_res_change_in_clvs_constraint_flag可以存在于通用约束信息语法结构中。等于1的标志no_res_change_in_clvs_constraint_flag可以指定标志sps_res_change_in_clvs_allowed_flag可等于0。等于0标志no_res_change_in_clvs_constraint_flag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的值可等于1。

[0199] 在一个实施例中,标志no_mixed_nalu_types_in_pic_constraint_flag可以存在于图13中的通用约束信息语法结构中。等于1的标志no_mixed_nalu_types_in_pic_constraint_flag指定标志pps_mixed_nalu_types_in_pic_flag的值可等于0。等于0的标志no_mixed_nalu_types_in_pic_constraint_flag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_types_in_pic_constraint_flag的值可等于1。

[0200] 在一个实施例中,标志no_trail_constraint_flag可以存在于图13中的通用约束信息语法结构中。等于1的标志no_trail_constraint_flag可以指定在01sInScope(01sInScope是输出层集,其包括涉及DPS的整个比特流中的所有层)中可以不存在nuh_unit_type等于TRAIL_NUT的NAL单元。等于0的标志no_trail_constraint_flag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可等于1。

[0201] 在一个实施例中,标志no_stsa_constraint_flag可以存在于图13中的通用约束信息语法结构中。等于1的标志no_stsa_constraint_flag可以指定01sInScope中可以不存在nuh_unit_type等于STSA_NUT的NAL单元。等于0的标志no_stsa_constraint_flag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可等于1。

[0202] 在一个实施例中,标志no_trail_constraint_flag可以存在于图13中的通用约束信息语法结构中。等于1的标志no_trail_constraint_flag可以指定01sInScope中可以不存在nuh_unit_type等于TRAIL_NUT的NAL单元。等于0的标志no_trail_constraint_flag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可等于1。

[0203] 在一个实施例中,标志no_stsa_constraint_flag可以存在于图13中的通用约束

信息语法结构中。等于1的标志no_stsa_constraint_flag可以指定0lsInScope中可以不存在nuh_unit_type等于STSA_NUT的NAL单元。等于0的标志no_stsa_constraint_flag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可等于1。

[0204] 在一个实施例中，例如如图13所示，标志no_idr_constraint_flag可以存在于通用约束信息语法结构中。等于1的no_idr_constraint_flag可以指定0lsInScope中可以不存在nuh_unit_type等于IDR_W_RADL或IDR_N_LP的NAL单元。等于0的标志no_idr_constraint_flag不施加这样的约束。

[0205] 在一个实施例中，例如如图13所示，标志no_cra_constraint_flag可以存在于通用约束信息语法结构中。等于1的标志no_cra_constraint_flag可以指定0lsInScope中可以不存在nuh_unit_type等于CRA_NUT的NAL单元。等于0的标志no_cra_constraint_flag不施加这样的约束。

[0206] 在一个实施例中，标志no_rasl_constraint_flag可以存在于图13中的通用约束信息语法结构中（标志no_rasl_constraint_flag未示出）。等于1的标志no_rasl_constraint_flag可以指定0lsInScope中可以不存在nuh_unit_type等于RASL_NUT的NAL单元。等于0的标志no_rasl_constraint_flag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的值可等于1。

[0207] 在一个实施例中，例如如图13所示，标志no_radl_constraint_flag可以存在于通用约束信息语法结构中。等于1的标志no_radl_constraint_flag可以指定0lsInScope中可以不存在nuh_unit_type等于RADL_NUT的NAL单元。等于0的标志no_radl_constraint_flag不施加这样的约束。当标志no_idr_constraint_flag等于1且标志no_cra_constraint_flag等于1时，标志no_rasl_constraint_flag的值可等于1。

[0208] 本公开的一些方面提供了用于范围扩展的约束标志信令技术，该范围扩展例如为具有残差rice编码扩展的范围扩展。

[0209] 根据本公开的一方面，可以基于Rice编码来执行残差编码。Rice编码使用可调参数将输入值分成两部分，然后使用不同的编码技术来编码这两部分。在一些示例中，可调参数被称为Rice参数。Rice参数可以影响Rice编码的编码效率。可以使用一些技术来确定特定应用的Rice参数，以提高针对特定应用的编码效率。确定Rice参数的一些技术可以包括在视频标准的范围扩展中。

[0210] 根据本公开的一方面，针对特定应用最初开发了一些标准。为了使标准适用于其他应用，使用工具开发了范围扩展用以支持其他应用。例如，最初HEVC的目标是应用每个样本8-10比特的4:2:0色度格式。为了使标准适用于除特定色度格式和特定位深以外的其他格式和位深，可以开发范围扩展以支持使用其他色度格式和/或更高位深的应用。

[0211] 为了将特征集限制为特定应用组所需的特征集，视频编码标准定义了配置文件，配置文件可以包括针对与使用这些特征的编码器的互操作性而要支持的定义的解码器特征集。例如，配置文件可以定义一组编码工具或算法，这些工具或算法可以用于生成一致的比特流。除了配置文件以外，一些标准（例如，VVC、HEVC等）还定义了水平和等级。水平对比特流施加与可以对应于解码器处理负载和存储器能力的空间分辨率、像素比、比特率值和变体有关的限制。可以根据最大采样率、最大图片尺寸、最大比特率、最小压缩比、编码图片

缓冲器的能力等来表示水平限制。更高的水平值可以对应于更高的复杂性限制。等级修改针对每一水平的比特率值和变化限制。例如,主(Main)等级用于大多数应用,而高(High)等级设计用于处理要求更高(例如具有显著比视频分发应用(video distribution application)高的比特率值)的视频贡献应用(video contribution application)。配置文件、等级和水平中的每一个都会影响实现方式和解码的复杂性,三者的组合为比特流和解码器指定了一个互操作点。

[0212] 在一些示例中,要求符合某一等级和水平的解码器,能够解码符合相同等级或相同水平中的较低等级或该水平之下的任何水平的所有比特流,并且符合特定配置文件的解码器可以支持该配置文件中的所有特征。在一些示例中,编码器不需要利用配置文件中支持的任何特定特征集,但是需要产生一致的比特流,即,服从使它们能够被一致的解码器解码的指定约束的比特流。

[0213] 除了PTL信息以外,PTL语法结构还可以包括通用约束信息(general constraints information,GCI)语法结构,其包括指示比特流的特定约束属性的约束标志和非标志语法元素的列表。

[0214] 在一个示例中,HEVC最初包括被称为“主(Main)”配置文件、“主10”配置文件(Main 10)和“主静止图片(Main Still Picture)”配置文件的三个配置文件。这三个配置文件具有一些限制,例如只支持4:2:0色度采样。在主配置文件和主静止图片配置文件中,仅支持每个样本8比特的视频精度,而主10配置文件最多支持每个样本10比特的视频精度。在主静止图片配置文件中,整个比特流仅包括一个编码图片。

[0215] 在一些示例中,具有范围扩展的HEVC可以支持附加的配置文件。在一个示例中,以下配置文件被统称为范围扩展配置文件:单色(Monochrome)配置文件、单色10(Monochrome 10)配置文件、单色12(Monochrome 12)配置文件、单色16(Monochrome 16)配置文件、主12(Main 12)配置文件、主4:2:2 10(Main 4:2:2 10)配置文件、主4:2:2 12(Main 4:2:2 12)配置文件、主4:4:4(Main 4:4:4)配置文件、主4:4:4 10(Main 4:4:4 10)配置文件、主4:4:4 12(Main 4:4:4 12)配置文件、主帧内(Main Intra)配置文件、主10帧内(Main10Intra)配置文件、主12帧内(Main 12Intra)配置文件、主4:2:2 10帧内(Main4:2:2 10Intra)配置文件、主4:2:2 12帧内(Main 4:2:2 12Intra)配置文件、主4:4:4帧内(Main 4:4:4Intra)配置文件、主4:4:4 10帧内(Main 4:4:4 10Intra)配置文件、主4:4:4 12帧内(Main 4:4:4 12Intra)配置文件、主4:4:4 16帧内(Main4:4:4 16Intra)配置文件、主4:4:4静止图片配置文件(Main 4:4:4Still Picture profile)和主4:4:4 16静止图片配置文件(Main 4:4:4 16Still Picture profile)。

[0216] 一些范围扩展配置文件可以支持更高的位深,并可以被称为用于具有高位深的操作范围扩展的配置文件。在一些示例中,用于具有高位深的操作范围扩展的配置文件包括支持每个样本大于10比特的配置文件,例如主12(Main 12)配置文件、主12 4:4:4(Main 12 4:4:4)配置文件、主16 4:4:4(Main 16 4:4:4)配置文件、主12帧内(Main 12Intra)配置文件、主12 4:4:4帧内(Main 12 4:4:4Intra)配置文件、主16 4:4:4帧内(Main 16 4:4:4Intra)配置文件、主12静止图片(Main 12Still Picture)配置文件、主12 4:4:4静止图片(Main 12 4:4:4Still Picture)配置文件、主16 4:4:4静止图片(Main 16 4:4:4Still Picture)配置文件等。

[0217] 具体地,主12配置文件允许每个样本具有8比特到12比特的位深,支持4:0:0和4:2:0色度采样,支持帧内预测模式和帧间预测模式两者。在一些示例中,符合主12配置文件的解码器有能力解码由以下配置文件构成的比特流:单色配置文件、单色12配置文件、主配置文件、主10配置文件和主12配置文件。

[0218] 主12 4:4:4配置文件允许每个样本具有8比特到12比特的位深,支持4:0:0、4:2:0、4:2:2和4:4:4色度采样,并支持帧内预测模式和帧间预测模式两者。在一些示例中,符合主12 4:4:4配置文件的解码器有能力解码由以下配置文件构成的比特流:单色配置文件、主配置文件、主10配置文件、主12配置文件、主10 4:2:2配置文件、主12 4:2:2配置文件、主4:4:4配置文件、主10 4:4:4配置文件、主12 4:4:4配置文件和单色12配置文件。

[0219] 主16 4:4:4配置文件允许每个样本具有8比特到16比特的位深,支持4:0:0、4:2:0、4:2:2和4:4:4色度采样,并支持帧内预测模式和帧间预测模式两者。

[0220] 主12帧内配置文件允许每个样本具有8比特到12比特的位深,支持4:0:0和4:2:0色度采样,以及帧内预测模式。

[0221] 主12 4:4:4帧内配置文件允许每个样本具有8比特到12比特的位深,支持4:0:0、4:2:0、4:2:2和4:4:4色度采样,以及帧内预测模式。

[0222] 主16 4:4:4帧内配置文件允许每个样本具有8比特到16比特的位深,支持4:0:0、4:2:0、4:2:2和4:4:4色度采样,以及帧内预测模式。

[0223] 主12静止图片配置文件允许每个样本具有8比特到12比特的位深,支持4:0:0和4:2:0色度采样。在主12静止图片配置文件中,整个比特流仅包括一个编码图片。

[0224] 主12 4:4:4静止图片配置文件允许每个样本具有8比特到12比特的位深,支持4:0:0、4:2:0、4:2:2和4:4:4色度采样。在主12 4:4:4静止图片配置文件中,整个比特流仅包括一个编码图片。

[0225] 主16 4:4:4静止图片配置文件允许每个样本具有8比特到16比特的位深,支持4:0:0、4:2:0、4:2:2和4:4:4色度采样。在主16 4:4:4静止图片配置文件中,整个比特流仅包括一个编码图片。

[0226] 根据本公开的一些方面,各种范围(例如,编码视频数据中的使用针对编码工具控制的持续存在的语法元素的实例而编码的一部分)处可以执行编码工具控制,该范围例如为比特流的范围、编码层视频序列(CLVS)的范围、图片、图片的切片等。在一些示例中,可以在通用约束信息(GCI)语法结构中提供编码工具控制,该通用约束信息语法结构通常包括比特流的约束信息,该比特流携带到解码器的输出层集。在一些示例中,可以在与CLVS相关联的序列参数集(SPS)中提供编码工具控制,SPS通常包括用于CLVS的信息。在一些示例中,可以在图片的图片头部中提供编码工具控制,且图片头部通常包括用于图片的信息。在一些示例中,可以在切片的切片头部中提供编码工具控制,切片头部通常包括用于切片的信息。

[0227] 根据本公开的一个方面,可以在各种范围处提供针对范围扩展中的编码工具的控制信息。在一些示例中,使用更大范围的语法元素可以提高编码效率。例如,大于0的GCI语法元素值指示比特流以特定方式被约束,通常以指示比特流中没有使用特定的编码工具。此外,等于0的GCI语法元素值发信令通知可以不应用关联的约束,使得允许(但不要求)在比特流中使用关联的编码工具(如果在指示的配置文件中支持使用关联的编码工具)。

[0228] 根据本公开的另一方面,当在比特流中的视频数据的编码中不使用编码工具时,指示例如在PTL信息和/或通用约束信息中不使用编码工具,不支持编码工具的视频解码器可以基于PTL信息和/或通用约束信息中的信令确定视频解码器能够解码比特流,并且可以扩展视频解码器的功能。

[0229] 在一些实施例中,编码器可以产生符合具有范围扩展的视频标准的比特流,但是不使用范围扩展中支持的一个或多个特征。在一些示例中,在知道不使用范围扩展中的一个或多个特征的情况下,符合视频标准但不支持范围扩展中的一个或多个特征的解码器可以确定解码器能够解码比特流,并且可以接受用于解码的比特流而不是拒绝比特流。

[0230] 图16示出了根据本公开的一些实施例的通用约束信息的语法结构(1600)。在一些示例中,语法结构(1600)包括将要应用于比特流的约束,例如包括到解码器的输出层集的比特流。在图16的示例中,在语法结构(1600)中由gci_num_adternaid_bits表示的语法元素用于在通用约束信息语法结构(1600)中指定不同于对齐零比特(alignment zero bits)语法元素(当存在时)的附加的通用约束信息(GCI)比特数。在一些标准中,要求gci_num_adteration_bits的值等于0或1。在一些标准中,解码器可以允许大于1的gci_num_additional_bits的值出现在语法结构中。

[0231] 在图16的示例中,语法结构(1600)包括5个附加的GCI比特(语法元素)(1601)-(1605),由general_no_extended_precision_constraint_flag、general_no_ts_residual_coding_rice_present_in_sh_constraint_flag、general_no_rice_extension_constraint_flag、general_no_persistent_rice_adaption_constraint_flag和general_no_reverse_last_sig_coeff_constraint_flag表示。在一些示例中,5个附加GCI比特(1601)-(1605)分别在输出层集的比特流的范围内提供编码工具的编码控制信息。

[0232] 图17示出了根据本公开一些实施例的序列参数集(SPS)范围扩展的语法结构(1700)示例。语法结构(1700)可以添加到用于CLVS的SPS中,以提供对用于CLVS的范围扩展的编码工具的控制。语法结构(1700)包括5个语法元素(1701)-(1705),由sps_extended_precision_flag、sps_ts_residual_coding_rice_present_in_sh_flag、sps_rrc_rice_extension_flag、sps_persistent_rice_adaptation_enabled_flag和sps_reverse_last_sig_coeff_enabled_flag表示。在一些示例中,5个语法元素(1701)-(1705)提供CLVS范围中的编码工具的编码控制信息。

[0233] 具体地,在一个实施例中,GCI比特(1601)和语法元素(1701)用于在不同的范围内提供对使用的扩展精度的控制,例如对扩展动态范围的编码工具的控制,该扩展动态范围用于缩放和变换过程中的变换系数,以及用于诸如abs_resement[]和dec_abs_level[]等的一些语法元素的二值化。

[0234] 语法元素(1701)等于1指定扩展的动态范围用于缩放和变换过程中的变换系数以及用于诸如abs_remainder[]和dec_abs_level[]等的一些语法元素的二值化。语法元素abs_remainst[scanning position n]是在扫描位置n处用Golomb-Rice码编码的变换系数水平的残差绝对值。当abs_remainder[]不存在时,推断它等于0。语法元素dec_abs_level[scanning position n]可以对应中间值,该中间值在扫描位置n处用Golomb-Rice码编码并用于确定扫描位置n处的变换系数的水平。语法元素(1701)等于0指定扩展的动态范围不用于缩放和变换过程中,并且不用于例如语法元素abs_resement[]和dec_abs_level[]等

的二值化。当语法元素(1701)不存在时,推断语法元素(1701)的值等于0。

[0235] 在一个示例中,由Log2TransformRange表示的变量用于确定缩放和变换过程中的变换系数的动态范围以及用于确定某些语法元素的二值化的动态范围。例如,变量Log2TransformRange可以是用于表示缩放和变换过程中的变换系数以及用于表示某些语法元素的二值化的比特数。动态范围可以是使用比特数表示的最大数和最小数的差。在一个示例中,变量Log2TransformRange是根据语法元素(1701)sps_extended_precision_flag推导的,例如使用公式(1): $\text{Log2TransformRange} = \text{sps_extended_precision_flag} ? \text{Max}(15, \text{Min}(20, \text{BitDepth}+6)) : 15$ (公式1)。

[0236] 用于缩放和变换过程中的变换系数的动态范围以及用于某些语法元素的二值化的动态范围可以基于变量Log2TransformRange来确定。在一些示例中,当标志sps_extended_precision_flag的值为0时,不使用扩展动态范围特征(例如,扩展动态范围的编码工具),并且变换系数的动态范围基于固定比特数例如15比特。当标志sps_extended_precision_flag的值为1时,使能扩展动态范围特征,并且基于公式1中示例的bitdepth BitDepth,表示缩放和变换过程中的变换系数的比特数可以是15比特、16比特、17比特、18比特、19比特和20比特之一。变换系数的动态范围可以基于比特数来确定。

[0237] 根据本公开的一方面,语法元素(例如,由SPS_BitDepth_minus_8表示)可用于发信号通知亮度和色度阵列的样本的位深(例如,由BitDepth表示)以及亮度和色度量化参数范围偏移的值(例如,由QpBdOffset表示)。在一个示例中,位深BitDepth可以根据公式(2)计算,并且QP(量化参数)范围偏移QpBdOffset可以根据公式(3)计算:

[0238] $\text{BitDepth} = 8 + \text{sps_bitdepth_minus8}$ 公式(2)

[0239] $\text{QpBdOffset} = 6 \times \text{sps_bitdepth_minus8}$ 公式(3)

[0240] 在一些示例中,GCI比特(1601)等于1指定用于输出层集(01sInScope)的范围中的所有图片的语法元素(1701)可以等于0。GCI比特(1601)等于0不施加这样的约束。因此,GCI比特(1601)等于1可以指定在比特流的编码中不使用扩展动态范围编码工具。

[0241] 在一些实施例中,GCI比特(1602)和语法元素(1702)用于在不同范围内提供对用于以变换跳过模式的残差编码的基于切片的Rice码的编码工具的控制,例如用于以变换跳过模式的残差编码的基于切片的Rice参数选择。

[0242] 根据本公开的一方面,用于变换跳过残差编码(transform skip residual coding,TSRC)的基于切片的Rice参数选择可以包括在视频标准的范围扩展中。在一些示例中,例如如图17所示,当启用变换跳过模式(例如,语法元素sps_transform_skip_enabled_flag为真)时,在序列参数集(SPS)中发信令通知一个控制标志(例如,由sps_ts_residual_coding_rice_present_in_sh_flag,语法元素(1702)表示),以指示启用或禁用用于变换跳过切片的Rice参数的信令。

[0243] 当发信令通知控制标志为启用(例如,等于“1”)时,针对每个变换跳过切片,例如在切片头部中进一步发信令通知一个语法元素(例如,由sh_ts_residual_coding_rice_idx_minus_1表示),以指示对该变换跳过切片的Rice参数的选择。当发信令通知控制标志为禁用(例如等于“0”)时,在切片水平下(例如,切片头部)没有发信令通知另一语法元素以指示用于变换跳过切片的Rice参数选择,且在一个示例中,默认的Rice参数可用于涉及SPS的编码视频数据中的所有变换跳过切片。

[0244] 例如,SPS中的语法元素(1702)等于1指定由sh_ts_residual_coding_rice_idx_minus 1表示的切片头部标志可以存在于涉及SPS的切片的切片头部(例如,slice_header())语法结构中。SPS中的语法元素(1702)等于0指定在涉及SPS的切片的slice_header()语法结构中不存在切片头部标志sh_ts_residual_coding_rice_idx_minus 1。在一些示例中,当不存在语法元素(1702)时,推断sps_ts_residual_codinge_rice_present_in_sh_flag的值等于0。

[0245] 在一些示例中,语法元素可以包括在通用约束信息中,以在输出层集的范围中控制基于切片的Rice码的编码工具的使用,该基于Rice码的切片用于以变换跳过模式的残差编码。例如,语法元素(1602)等于1指定用于输出层集(01sInScope)的范围中的所有图片的语法元素(1702)可以等于0。语法元素(1602)等于0不施加这样的约束。因此,在一些示例中,比特流中GCI比特(1602)等于1可以指定不使用用于变换跳过残差编码的基于切片的Rice参数选择来编码该比特流。

[0246] 在一些实施例中,GCI比特(1603)和语法元素(1703)用于在不同范围内提供对一个或多个编码工具的控制,该一个或多个编码工具用于针对一些语法元素的二值化的Rice参数推导,这些语法元素例如为常规残差编码(RRC)中的abs_resement[]和dec_abs_level[]等。在一些示例中,常规残差编码(RRC)指的是用于对通过变换和量化获得的块进行编码的一些技术。在一些示例中,可以针对仅通过量化获得的块修改RRC。在一些示例中,变换跳过残差编码(TSRC)指的是专用于对绕过变换(也称为变换跳过)获得的块进行编码的一些技术。

[0247] 在一些示例中,视频编码标准可以包括用于一些语法元素(例如abs_resement[]和dec_abs_level[])的二值化的Rice参数推导的一个或多个编码工具,且视频编码标准的范围扩展可以包括用于一些语法元素(例如abs_resement[]和dec_abs_level[])的二值化的Rice参数推导的一个或多个替代编码工具。

[0248] 在一些示例中,视频标准将基于本地模板技术用于Rice参数推导。例如,包括一个或多个(例如,在一个示例中为5个)相邻系数水平的模板用于rice参数推导。例如,可以计算模板内绝对系数值的和,然后基于该和确定Rice参数。在一个示例中,可以基于该和使用查找表来确定Rice参数。

[0249] 需要注意的是,Rice参数可以由其他合适的编码工具来确定。在一个示例中,一个公式可用于基于该和确定Rice参数。在另一示例中,上下文建模可用于基于对相邻系数水平的统计来确定Rice参数。在一些示例中,视频标准的范围扩展可以指定用于Rice参数推导的一个或多个替代编码工具。

[0250] 在一些示例中,视频标准的范围扩展可以包括对用于在其他场景中使用的RRC的修改。在一个示例中,范围扩展可以包括不同的上下文建模工具和用于变换跳过模式中的残差编码的残差信号旋转工具。

[0251] 在一些示例中,SPS中的语法元素(1703)等于1指定用于abs_remainder[]和dec_abs_level[]的二值化的替代Rice参数推导(例如,在范围扩展中用于Rice参数推导的替代编码工具)用于编码涉及SPS的CLVS。语法元素(1703)等于0指定用于abs_remainder[]和dec_abs_level[]的二值化的替代Rice参数推导不用于对涉及SPS的CLVS进行编码。当不存在语法元素(1703)时,推断语法元素(1703)的值等于0。

[0252] 在一些示例中,语法元素(1603)等于1指定用于输出层集(01sInScope)范围中的所有图片的语法元素(1703)可以等于0。语法元素(1603)等于0不施加这样的约束。因此,在一些示例中,GCI比特(1603)等于1可以指定不使用用于abs_remainder[]和dec_abs_level[]的二值化的替代Rice参数推导(例如,在指定的范围扩展中指定的用于Rice参数推导的替代编码工具)编码比特流。

[0253] 在一些实施例中,GCI比特(1604)和语法元素(1704)用于在不同范围内提供对用于abs_remainder[]和dec_abs_level[]的二值化的基于统计的Rice参数推导的控制。

[0254] 根据本公开的一方面,可以在每个变换单元(TU)的开始处使用从先前的TU累积的统计来初始化用于abs_remainder[]和dec_abs_level[]的二值化的Rice参数推导。在一些示例中,基于统计的Rice参数推导可以包括在视频标准的范围扩展中。

[0255] 在一些示例中,SPS中的控制标志(例如由sps_persistent_rice_adaptation_enabled_flag表示的语法元素(1704))用于控制基于统计的Rice参数推导。例如,SPS中的语法元素(1704)等于1指定在每个TU开始处使用从先前的TU累积的统计来初始化用于abs_remainder[]和dec_abs_level[]的二值化的Rice参数推导。语法元素(1704)等于0指定在当前TU的Rice参数推导中不使用先前的TU状态。当不存在语法元素(1704)时,推断语法元素(1704)的值等于0。

[0256] 此外,在一个实施例中,语法元素(1604)等于1指定用于输出层集(01sInScope)的范围中的所有图片的语法元素(1704)可以等于0。语法元素(1604)等于0不施加这样的约束。因此,在一些示例中,GCI比特(1604)等于1可以指定不使用基于统计的Rice参数推导编码比特流。

[0257] 在一些实施例中,GCI比特(1605)和语法元素(1705)用于在不同范围内提供对编码工具的控制,该编码工具用于编码在变换系数的熵编码期间最后的显著性系数(last significant coefficient)的位置。在一个示例中,最后的显著性系数的位置可以被不同的编码工具编码。例如,视频标准可以指定第一编码工具,该第一编码工具可以通过对由LastSignificantCoeffX和LastSignificantCoeffY变量表示的位置的两个坐标进行编码来确定最后的显著性系数的位置;并且视频标准的范围扩展可以指定例如第二编码工具的替代编码工具,在一个示例中,该第二编码工具可以通过参考归零变换块的右下角对最后的显著性系数的相对坐标进行编码来确定最后的显著性系数的位置。

[0258] 在一些示例中,SPS中的语法元素(1705)等于1指定由sh_reverse_last_sig_coeff_flag表示的切片头部标志(切片范围)存在于涉及SPS的切片头部语法结构(例如,在一些示例中为slice_header())中。SPS中的语法元素(1705)等于0指定在涉及SPS的切片头部语法结构中不存在切片头部标志sh_reverse_last_sig_coeff_flag,并且可以推断切片头部标志sh_reverse_last_sig_coeff_flag为零。当不存在语法元素(1705)时,推断语法元素(1705)的值等于0。

[0259] 在一些示例中,切片的切片头部标志sh_reverse_last_sig_coeff_flag的值用于确定在编码切片时的缩放和变换过程中的变换系数中的最后的显著性系数的位置推导。在一个示例中,当sh_reverse_last_sig_coeff_flag等于1时,最后的显著性系数位置由视频标准的范围扩展中的替代编码工具(例如第二编码工具)编码;否则,针对最后的显著性系数位置的当前坐标由第一编码工具编码。

[0260] 在一些示例中,等于1的GCI比特(1605)指定用于输出层集(01sInScope)的范围中的所有图片的语法元素(1705)可以等于0。GCI比特(1605)等于0不施加这样的约束。因此,GCI比特(1605)等于1可以指定在最后的显著性系数的位置推导中不将第二编码工具用于比特流的范围。

[0261] 图18示出了概述根据本公开实施例的处理(1800)的流程图。处理(1800)可以用在视频解码器中。在各种实施例中,处理(1800)由处理电路执行,处理电路为例如终端设备(310)、(320)、(330)和(340)中的处理电路、执行视频解码器(410)的功能的处理电路、执行视频解码器(510)的功能的处理电路等。在一些实施例中,处理(1800)用软件指令实现,由此当处理电路执行该软件指令时,该处理电路执行处理(1800)。该处理从(S1801)开始,并进行到(S1810)。

[0262] 在(S1810),确定第一语法元素(例如general_no_rrc_rice_extension_constraint_flag)的值,该第一语法元素用于比特流中的编码视频数据(例如,输出层集)的第一范围中的编码控制。第一语法元素与第二编码工具关联,该第二编码工具替代第一编码工具以用于残差编码中的Rice参数推导。

[0263] 在一个示例中,响应于针对通用约束信息的语法结构中的语法元素(例如,gci_num_adternaid_bits)在语法结构中指示针对通用约束信息的附加比特,从语法结构中解码第一语法元素。

[0264] 在一些示例中,在标准中定义第一编码工具,并且在标准的范围扩展中定义第二编码工具作为第一编码工具的替代编码工具。在一个示例中,视频解码器可以支持该标准,但可以不支持该标准的范围扩展中的第二编码工具。

[0265] 在(1820),当第一语法元素的值是第一值时,处理进行到(S1830);否则,该处理进行到(S1840)。第一值指示在比特流中的编码视频数据的第一范围的编码中不使用第二编码工具,该比特流中的编码视频数据的第一范围包括编码视频数据(例如,输出层集中的一或多个CLVS)的一个或多个第二范围。

[0266] 在一些示例中,第一语法元素在通用约束信息中,该通用约束信息用于对解码器输出的输出层集中的图片的编码控制。在一个示例中,第一语法元素的第一值指示在对输出层集中的每个编码层视频序列(CLVS)的编码中不使用第二编码工具。

[0267] 在(1830),响应于第一语法元素是第一值,在不调用第二编码工具的情况下解码比特流中的编码视频数据的第一范围。

[0268] 在一些示例中,将用于对比特流中的编码层视频序列(CLVS)的编码控制的第二语法元素(例如,sps_extended_precision_flag)约束为具有指示不调用第二编码工具用于解码CLVS的的值。

[0269] 在(1840),响应于第一语法元素是第二值,确定第二语法元素(例如,sps_extended_precision_flag)的值,以解码第二范围中的编码视频数据,第二语法元素用于对比特流中的编码视频数据(例如编码层视频序列(CLVS))的第二范围的编码控制。第二语法元素指示在CLVS中使用/不使用第二编码工具。在一个示例中,用于CLVS的序列参数集(SPS)中不存在第二语法元素,则推断第二语法元素的值指示CLVS中不使用第二编码工具。

[0270] 然后根据第二语法元素的值解码编码视频数据的第二范围(例如,调用或不调用第二编码工具)。

[0271] 可以适当地修改处理(1800)。可以修改和/或省略处理(1800)中的步骤。可以添加额外的步骤。可以使用任何合适的实现顺序。

[0272] 图19示出了概述根据本公开实施例的处理(1900)的流程图。处理(1900)可以用在视频编码器中。在各种实施例中,处理(1900)由处理电路执行,处理电路为例如终端设备(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、执行视频编码器(603)的功能的处理电路、执行视频编码器(703)的功能的处理电路等。在一些实施例中,处理(1900)用软件指令实现,由此当处理电路执行该软件指令时,该处理电路执行处理(1900)。该处理从(S1901)开始,并进行到(S1910)。

[0273] 在(S1910),处理电路确定在编码比特流中的编码视频数据的第一范围(例如,输出层集)期间是否使用第二编码工具替代第一编码工具。第二编码工具是用于残差编码中的Rice参数推导的第一编码工具的替代编码工具。编码视频数据的第一范围包括编码视频数据的一个或多个第二范围(例如,CLVS)。

[0274] 在一些示例中,处理电路可以基于用于对比特流中的编码层视频序列(CLVS)的编码控制的第二语法元素(例如,sps_extended_precision_flag)来确定是否使用第二编码工具。

[0275] 在(S1920),当在对编码视频数据的第一范围的编码中不使用第二编码工具时,处理进行到(S1930);否则,处理进行到(S1940)。

[0276] 在(S1930),在比特流中编码具有第一值的第一语法元素(例如,general_no_rrc_rice_extension_constraint_flag)。第一语法元素用于在比特流中的编码视频数据的第一范围(例如,输出层集合)中的编码控制。第一语法元素与用于残差编码中的Rice参数推导的第二编码工具关联。第一值指示在对编码视频数据的第一范围的编码中不使用第二编码工具。

[0277] 在一个示例中,在针对通用约束信息的语法结构中编码第一语法元素,并调整该语法结构中的语法元素(例如,gci_num_additional_bits)以在语法结构中指示针对通用约束信息的附加比特。

[0278] 在(S1940),在比特流中编码具有第二值的第一语法元素。在一些示例中,在比特流中不编码第一语法元素,例如,在第二值是第一语法元素的默认值的情况下,由此如果比特流中不存在第一语法元素,则可以推断出第一语法元素的值为第二值,然后可以跳过(S1940)。

[0279] 可以适当地修改处理(1900)。可以修改和/或省略处理(1900)中的步骤。可以添加额外的步骤。可以使用任何合适的实现顺序。

[0280] 上述描述的技术(例如,发信令通知约束标志的技术、自适应分辨率参数的技术和/或类似的技术)可以使用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20示出了适用于实现所公开主题的某些实施例的计算机系统(2000)。

[0281] 可使用任何合适的机器代码或计算机语言对计算机软件进行编码,任何合适的机器代码或计算机语言可经受汇编、编译、链接或类似的机制以创建包括指令的代码,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,或者通过解释、微代码执行等执行。

[0282] 指令可以在各种类型的计算机或其组件上执行,计算机或其组件包括例如个人计算机、平板计算机、服务器、智能电话、游戏设备、物联网设备等。

[0283] 图20所示的计算机系统(2000)的组件本质上是示例性的,并不旨在对实施本公开的实施例的计算机软件的用途或功能的范围提出任何限制。组件的配置也不应解释为具有与计算机系统(2000)的示例性实施例中所示的组件中的任何一个组件或组件的组合相关的任何依赖或要求。

[0284] 计算机系统(2000)可包括某些人机接口输入设备。此类人机接口输入设备可响应于一个或多个用户通过例如下述的输入:触觉输入(例如:击键、划动,数据手套移动)、音频输入(例如:语音、拍手)、视觉输入(例如:手势)、嗅觉输入(未描绘)。人机接口设备还可用于捕获不一定与人的意识输入直接相关的某些媒介,例如音频(例如:语音、音乐、环境声音)、图像(例如:扫描图像、从静止图像相机获取的拍摄图像)、视频(例如,二维视频、包括立体视频的三维视频)。

[0285] 人机接口输入设备可包括下述的一项或多项(每种中仅示出一个):键盘(2001)、鼠标(2002)、触控板(2003)、触摸屏(2010)、数据手套(未示出)、操纵杆(2005)、麦克风(2006)、扫描仪(2007)、相机(2008)。

[0286] 计算机系统(2000)还可包括某些人机接口输出设备。此类人机接口输出设备可通过例如触觉输出、声音、光和气味/味道来刺激一个或多个用户的感官。此类人机接口输出设备可包括触觉输出设备(例如,触摸屏(2010)的触觉反馈、数据手套(未示出)或操纵杆(2005),但还可以是不作为输入设备的触觉反馈设备)、音频输出设备(例如:扬声器(2009)、耳机(未描绘))、视觉输出设备(例如,包括CRT屏幕、LCD屏幕、等离子屏幕、OLED屏幕的屏幕(2010),每种屏幕具有或没有触摸屏输入功能,每种屏幕具有或没有触觉反馈功能,其中的一些屏幕能够通过诸如立体图像输出之类的装置、虚拟现实眼镜(未描绘)、全息显示器和烟箱(未描绘)以及打印机(未描绘)来输出二维视觉输出或超过三维的输出)。

[0287] 计算机系统(2000)还可包括人类可访问存储设备及其关联介质,例如包括具有CD/DVD等介质(2021)的CD/DVD ROM/RW(2020)的光学介质、指状驱动器(2022)、可拆卸硬盘驱动器或固态驱动器(2023)、诸如磁带和软盘之类的传统磁性介质(未描绘)、诸如安全软件狗之类的基于专用ROM/ASIC/PLD的设备(未描绘)等。

[0288] 本领域技术人员还应该理解,结合当前公开的主题所使用的术语“计算机可读介质”不涵盖传输介质、载波或其它暂时性信号。

[0289] 计算机系统(2000)还可包括通向一个或多个通信网络(2055)的接口(2054)。网络可例如是无线网络、有线网络、光网络。网络可进一步是本地网络、广域网络、城域网络、车辆和工业网络、实时网络、延迟容忍网络等。网络的示例包括诸如以太网之类的局域网、无线LAN、包括GSM、3G、4G、5G、LTE等的蜂窝网络、包括有线电视、卫星电视和地面广播电视的电视有线或无线广域数字网络、包括CANBus的车辆和工业网络等。某些网络通常需要附接到某些通用数据端口或外围总线(2049)的外部网络接口适配器(例如,计算机系统(2000)的USB端口);如下所述,其它网络接口通常通过附接到系统总线而集成到计算机系统(2000)的内核中(例如,连接到PC计算机系统中的以太网接口或连接到智能手机计算机系统中的蜂窝网络接口)。计算机系统(2000)可使用这些网络中的任何网络与其它实体通信。此类通信可以是仅单向接收的(例如,广播电视)、仅单向发送的(例如,连接到某些CANBus

设备的CANBus)或双向的,例如,使用局域网或广域网数字网络连接到其它计算机系统。如上所述,可以在那些网络和网络接口中的每一个上使用某些协议和协议栈。

[0290] 上述人机接口设备、人机可访问的存储设备和网络接口可附接到计算机系统(2000)的内核(2040)。

[0291] 内核(2040)可包括一个或多个中央处理单元(CPU)(2041)、图形处理单元(GPU)(2042)、现场可编程门区域(FPGA)(2043)形式的专用可编程处理单元、用于某些任务的硬件加速器(2044)、图形适配器(2050)等。这些设备以及只读存储器(ROM)(2045)、随机存取存储器(2046)、诸如内部非用户可访问的硬盘驱动器、SSD等之类的内部大容量存储器(2047)可通过系统总线(2048)连接。在一些计算机系统中,可以以一个或多个物理插头的形式访问系统总线(2048),以能够通过附加的CPU、GPU等进行扩展。外围设备可直接附接到内核的系统总线(2048)或通过外围总线(2049)附接到内核的系统总线(2048)。在一个示例中,屏幕(2010)可连接到图形适配器(2050)。外围总线的架构包括PCI、USB等。

[0292] CPU(2041)、GPU(2042)、FPGA(2043)和加速器(2044)可执行某些指令,这些指令可组合来构成上述计算机代码。该计算机代码可存储在ROM(2045)或RAM(2046)中。过渡数据可存储在RAM(2046)中,而永久数据可例如存储在内部大容量存储器(2047)中。可通过使用高速缓存来进行通向任何存储设备的快速存储及检索,该高速缓存可与下述紧密关联:一个或多个CPU(2041)、GPU(2042)、大容量存储(2047)、ROM(2045)、RAM(2046)等。

[0293] 计算机可读介质可以在其上具有执行各种由计算机实现的操作的计算机代码。介质和计算机代码可以是出于本公开的目的而专门设计和构造的介质和计算机代码,或者介质和计算机代码可以是计算机软件领域的技术人员公知且可用的类型。

[0294] 作为示例,而非限制,可由于一个或多个处理器(包括CPU、GPU、FPGA、加速器等)执行包含在一种或多种有形的计算机可读介质中的软件而使得具有架构(2000),特别是内核(2040)的计算机系统提供功能。此类计算机可读介质可以是与如上所介绍的用户可访问的大容量存储相关联的介质,以及某些非暂时性内核(2040)的存储器,例如内核内部大容量存储器(2047)或ROM(2045)。实施本公开的各种实施例的软件可存储在此类设备中并由内核(2040)执行。根据特定需要,计算机可读介质可包括一个或多个存储设备或芯片。软件可使得内核(2040),特别是其中的处理器(包括CPU、GPU、FPGA等)执行本文所描述的特定过程或特定过程的特定部分,包括定义存储在RAM(2046)中的数据结构以及根据由软件定义的过程来修改此类数据结构。附加地或作为替代,可由于硬连线或以其它方式体现在电路(例如:加速器(2044))中的逻辑而使得计算机系统提供功能,该电路可替代软件或与软件一起运行以执行本文描述的特定过程或特定过程的特定部分。在适当的情况下,提及软件的部分可包含逻辑,反之亦然。在适当的情况下,提及计算机可读介质的部分可包括存储用于执行的软件的电路(例如,集成电路(IC))、体现用于执行的逻辑的电路或包括两者。本公开包括硬件和软件的任何合适的组合。

[0295] 附录A:首字母缩写词

[0296] JEM:联合探索模型

[0297] VVC:下一代视频编码

[0298] BMS:基准集

[0299] MV:运动矢量

- [0300] HEVC:高效视频编码
- [0301] SEI:辅助增强信息
- [0302] VUI:视频可用性信息
- [0303] GOP:图片群组
- [0304] TU:变换单元
- [0305] PU:预测单元
- [0306] CTU:编码树单元
- [0307] CTB:编码树块
- [0308] PB:预测块
- [0309] HRD:假想参考解码器
- [0310] SNR:信噪比
- [0311] CPU:中央处理单元
- [0312] GPU:图形处理单元
- [0313] CRT:阴极射线管
- [0314] LCD:液晶显示器
- [0315] OLED:有机发光二极管
- [0316] CD:光盘
- [0317] DVD:数字视频光盘
- [0318] ROM:只读存储器
- [0319] RAM:随机存取存储器
- [0320] ASIC:专用集成电路
- [0321] PLD:可编程逻辑器件
- [0322] LAN:局域网
- [0323] GSM:全球移动通信系统
- [0324] LTE:长期演进
- [0325] CANBus:控制器局域网络总线
- [0326] USB:通用串行总线
- [0327] PCI:互连外围组件
- [0328] FPGA:现场可编程门区域
- [0329] SSD:固态驱动器
- [0330] IC:集成电路
- [0331] CU:编码单元
- [0332] 虽然本公开已描述多个示例性实施例,但是存在落入本公开的范围内的修改、置换和各种替换等效物。因此,应理解,本领域技术人员将能够设计出许多虽然未在本文中明确示出或描述,但是体现了本公开的原理,因此落入本公开的精神和范围内的系统和方法。

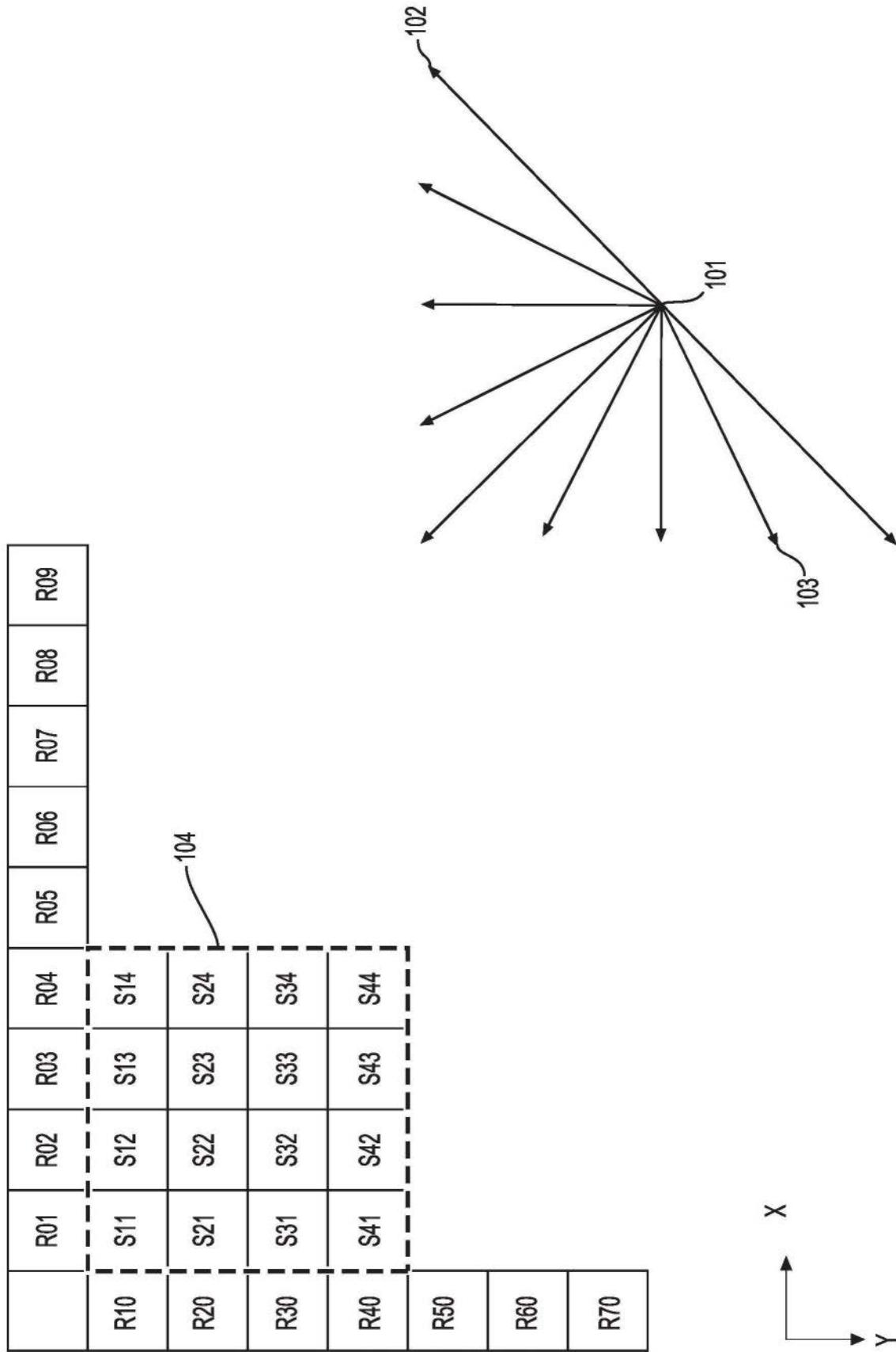


图1A(相关技术)

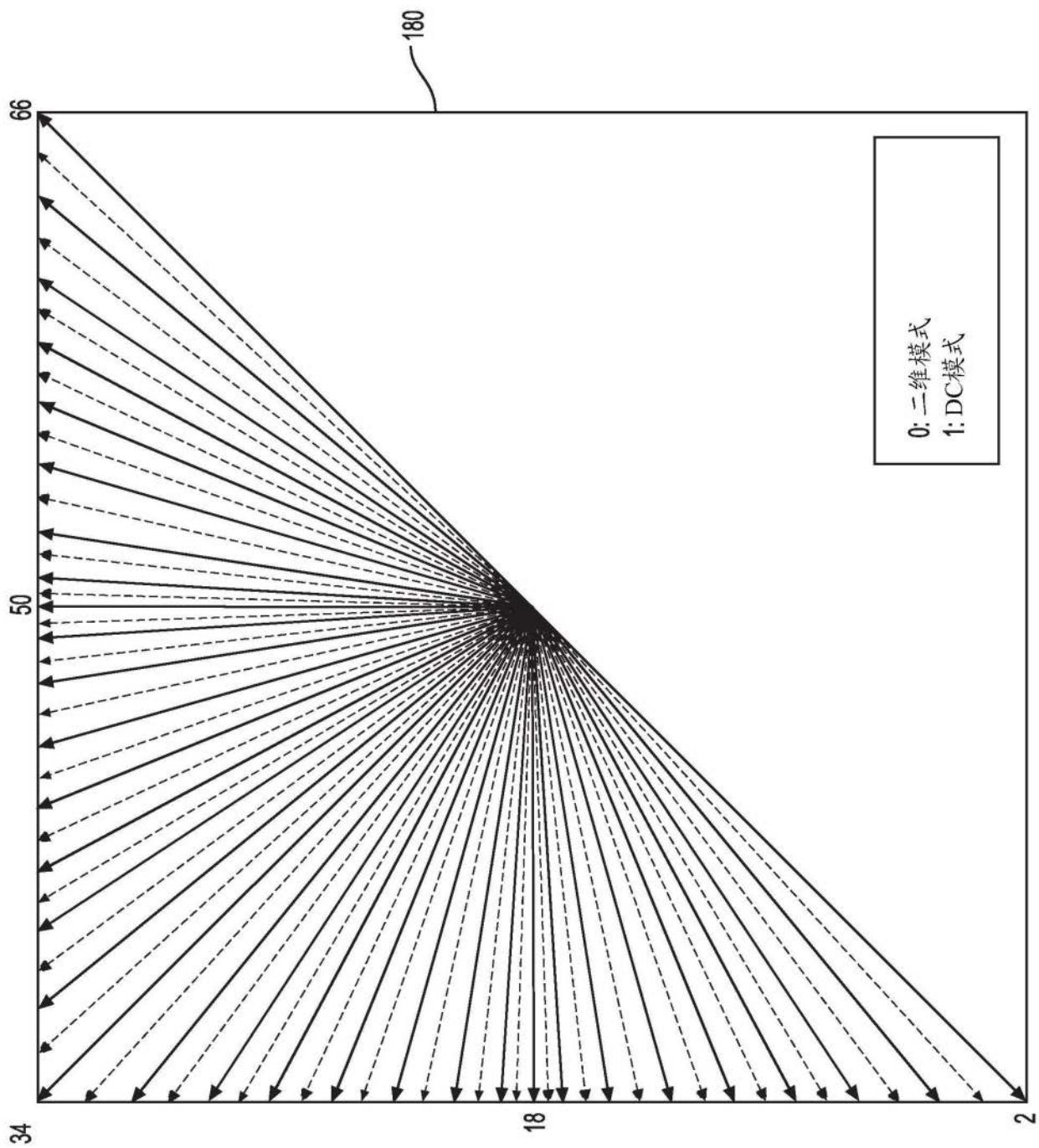


图1B(相关技术)

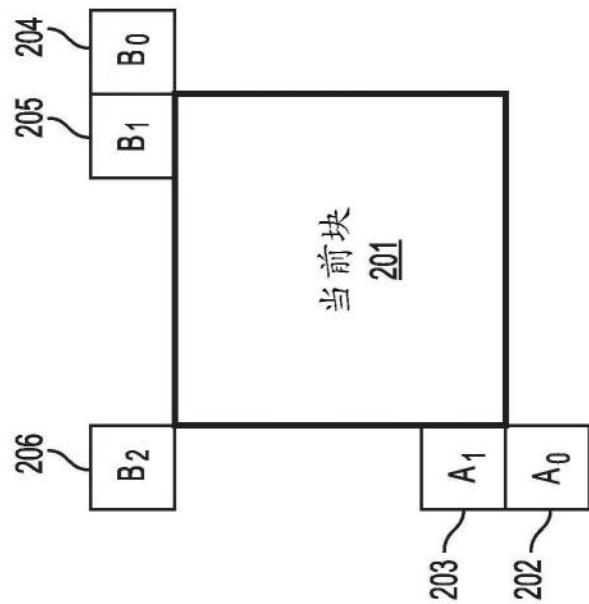


图2(相关技术)

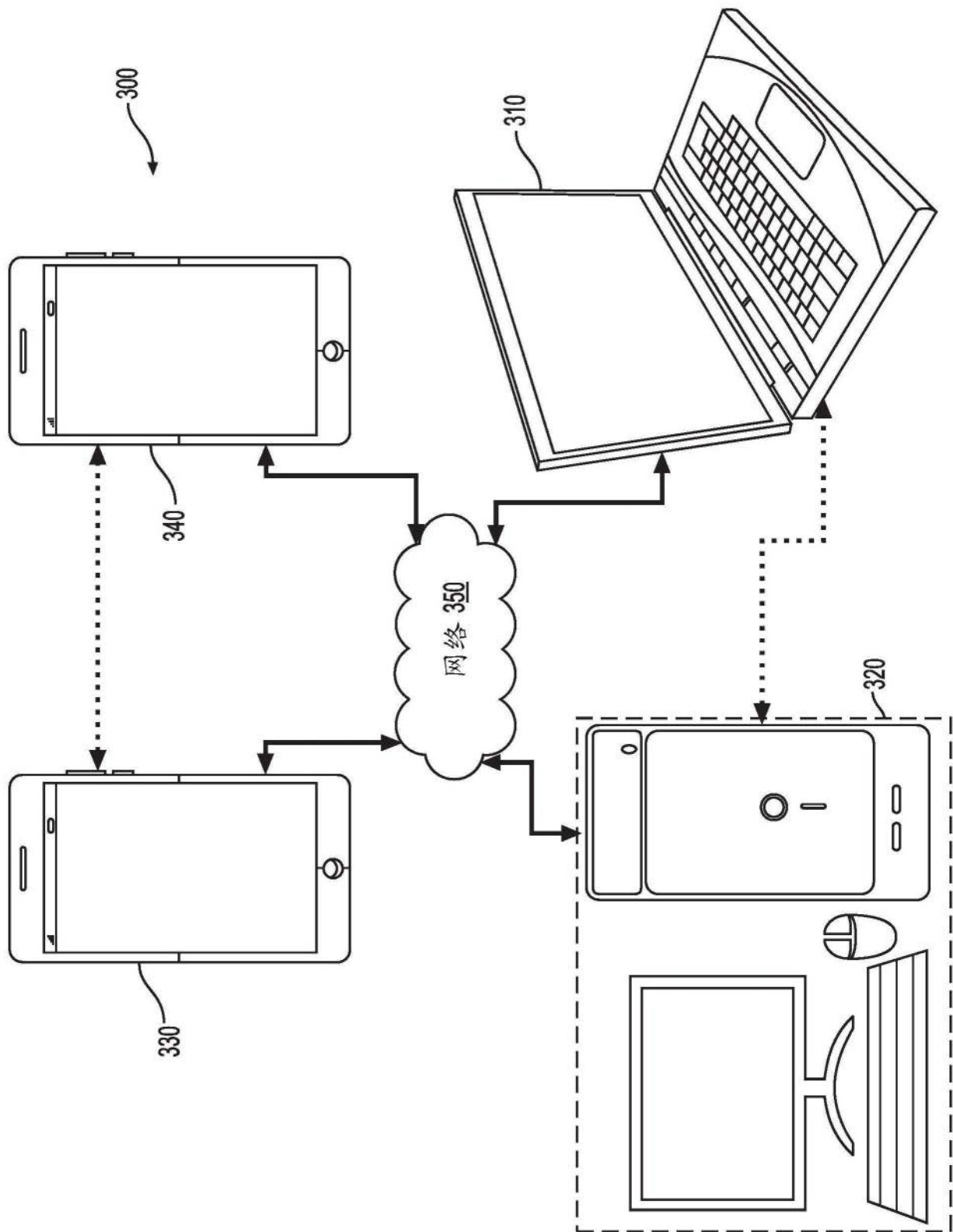


图3

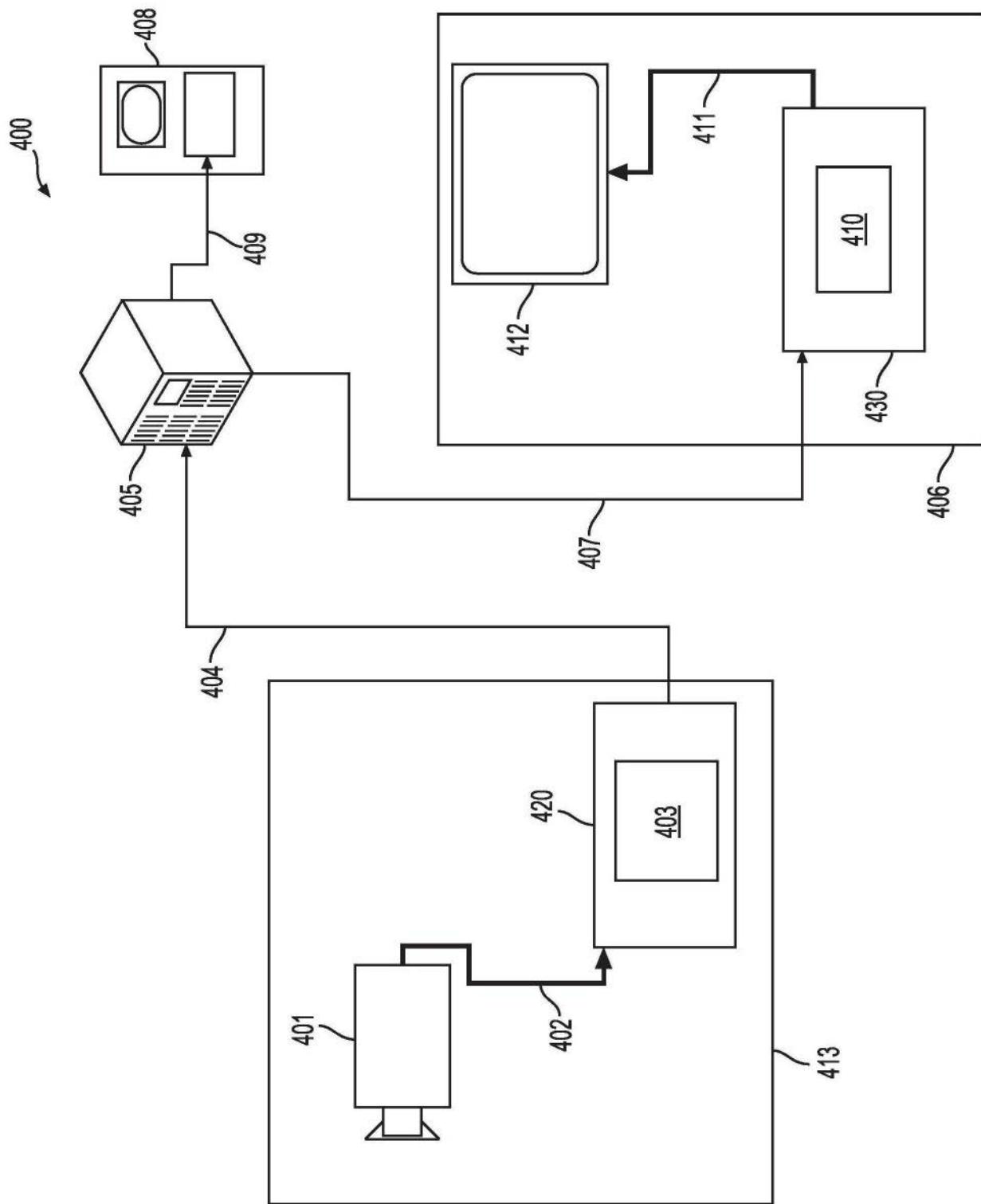


图4

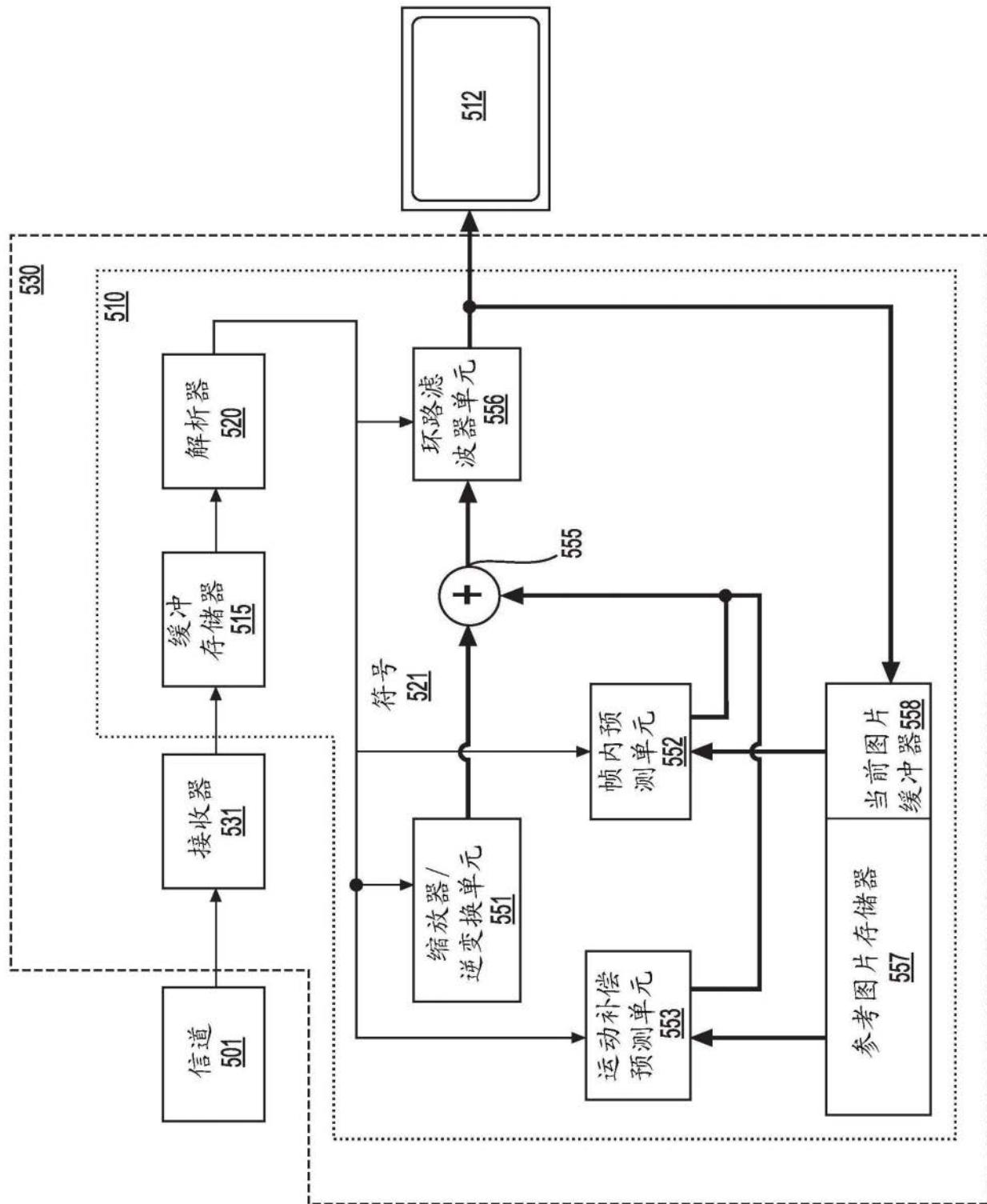


图5

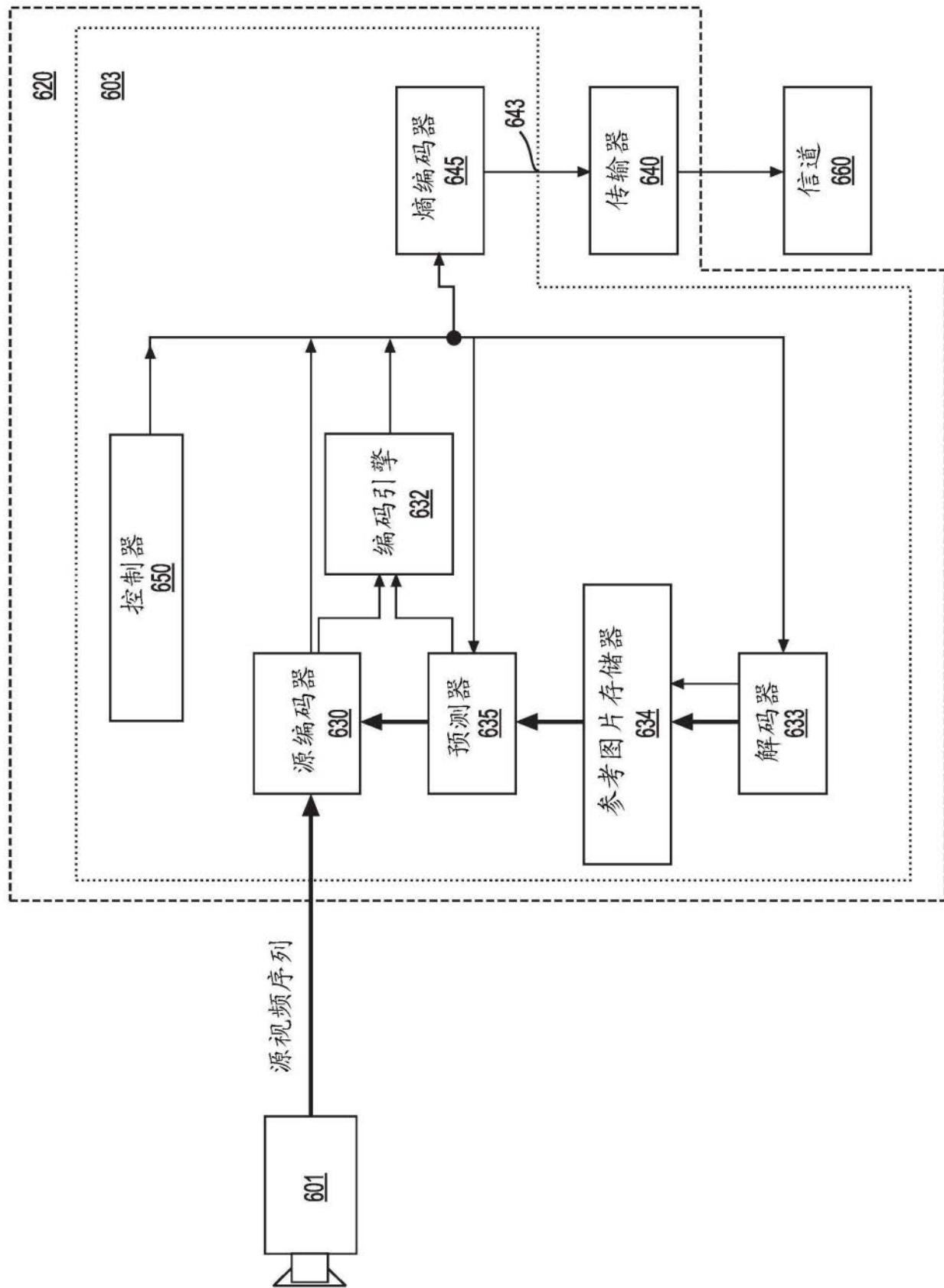


图6

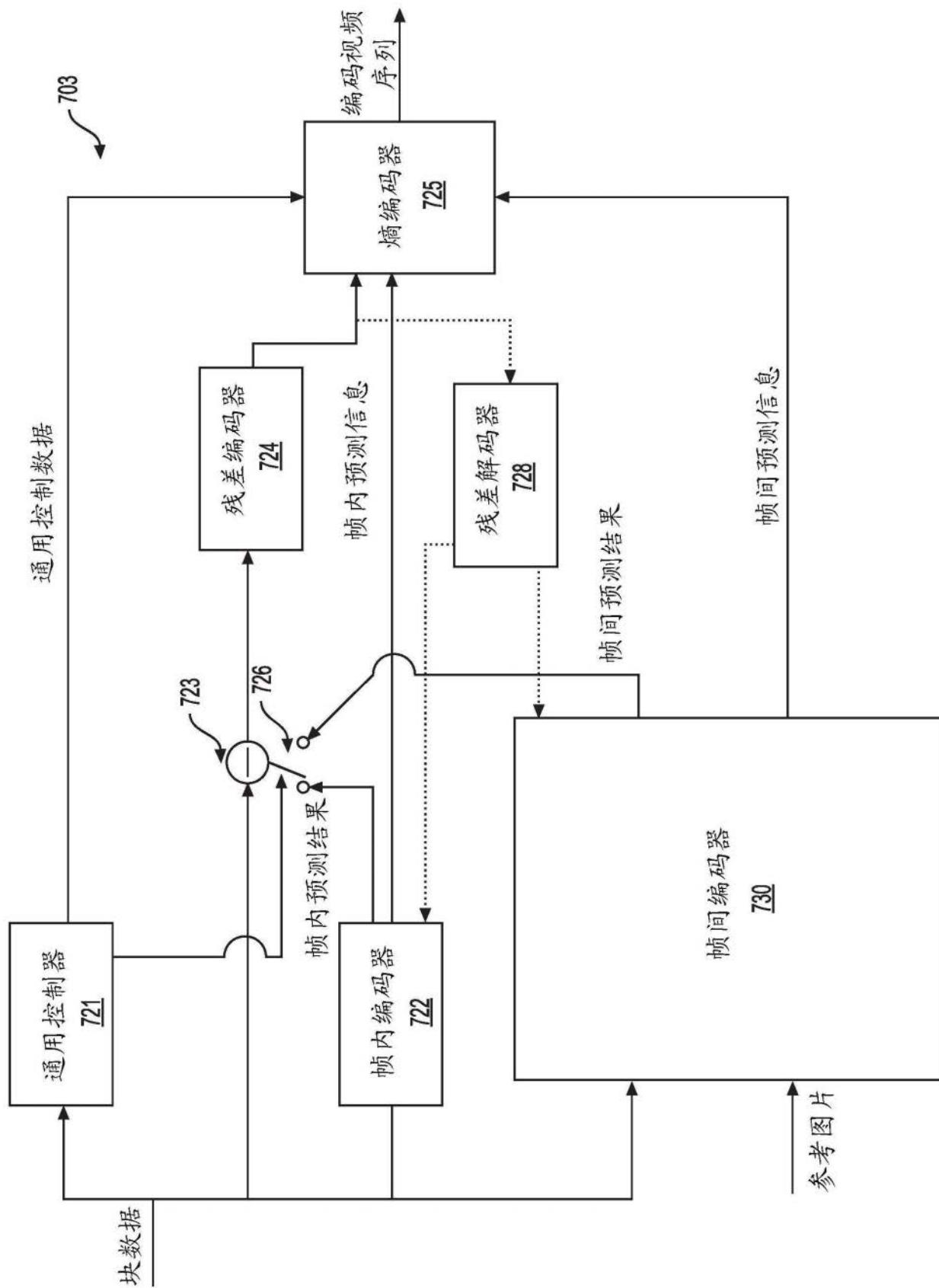


图7

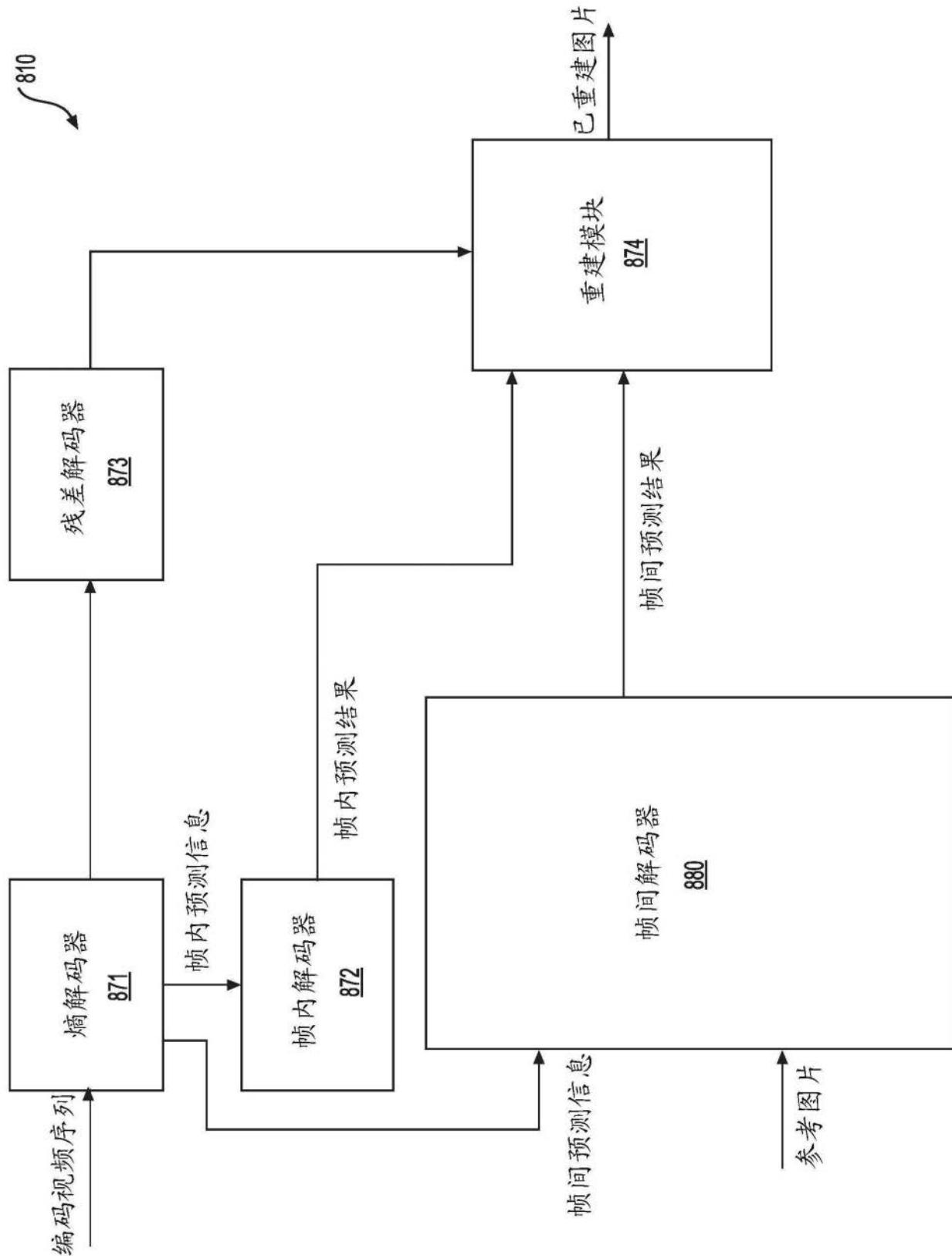


图8

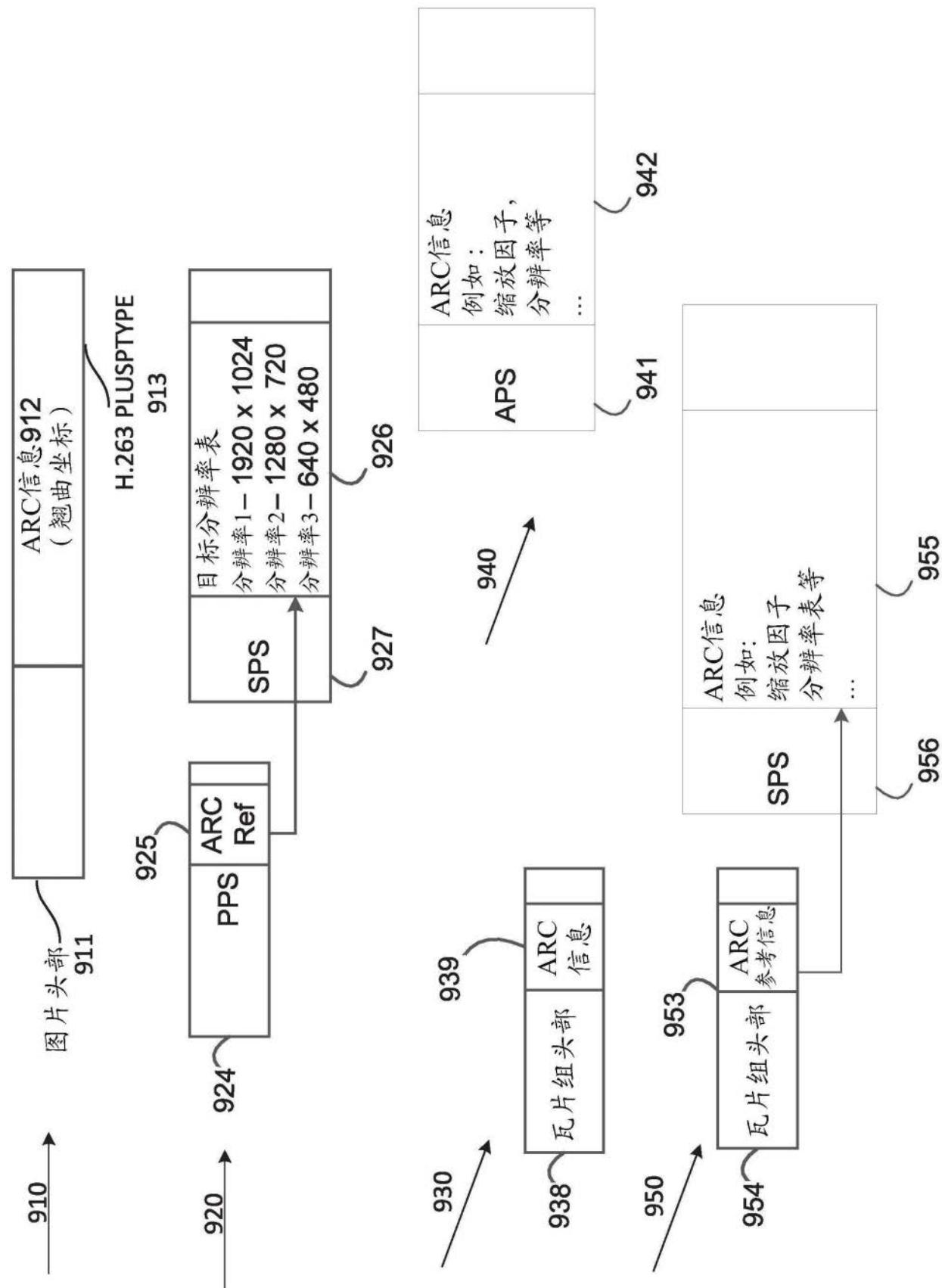


图9

1000
↙

表1:映射上采样或下采样因子、码字和Ext-Golomb码

码字	Ext-Golomb 码	初始/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%)

图10

1101	tile_group_header() {	
	...	
1103	if(adaptive_pic_resolution_change_flag) {	
1102	dec_pic_size_idx u(1)	
	}	
	...	
	}	
1110	seq_parameter_set_rbsp() {	描述
	...	
1111	adaptive_pic_resolution_change_flag u(1)	
1112	if(adaptive_pic_resolution_change_flag) {	
1113	output_pic_width_in_luma_samples ue(v)	
	output_pic_height_in_luma_samples ue(v)	
1114	reference_pic_size_present_flag u(1)	
	if(reference_pic_size_present_flag)	
	{	
1115	reference_pic_width_in_luma_samples ue(v)	
	reference_pic_height_in_luma_samples ue(v)	
	}	
1116	num_dec_pic_size_in_luma_samples_minus1 ue(v)	
	for(i = 0; i <=	
	num_dec_pic_size_in_luma_samples_minus1 ;	
	i++) {	
1117	dec_pic_width_in_luma_samples[i] ue(v)	
	dec_pic_height_in_luma_samples[i] ue(v)	
	}	
	}	
	...	
	}	

图11

1200
~~~~~

|                                                                       | 描述    |
|-----------------------------------------------------------------------|-------|
| profile tier level( profileTierPresentFlag, maxNumSubLayersMinus1 ) { |       |
| if( profileTierPresentFlag ) {                                        |       |
| <b>general profile idc</b>                                            | u(7)  |
| <b>general tier flag</b>                                              | u(1)  |
| general constraint info()                                             |       |
| }                                                                     |       |
| <b>general level idc</b>                                              | u(8)  |
| if( profileTierPresentFlag ) {                                        |       |
| <b>num sub profiles</b>                                               | u(8)  |
| for( i = 0; i < num sub profiles; i++ )                               |       |
| <b>general sub profile idc[ i ]</b>                                   | u(32) |
| }                                                                     |       |
| for( i = 0; i < maxNumSubLayersMinus1; i++ )                          |       |
| <b>sublayer level present flag[ i ]</b>                               |       |
| while( !byte aligned() )                                              |       |
| <b>ptl alignment zero bit</b>                                         | f(1)  |
| for( i = 0; i < maxNumSubLayersMinus1; i++ )                          |       |
| if( sublayer level present flag[ i ] )                                |       |
| <b>sublayer level idc[ i ]</b>                                        | u(8)  |
| }                                                                     |       |

图12

|                                                   | 描述   |
|---------------------------------------------------|------|
| 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) |
| no lossless_coding_tool_constraint_flag           | u(1) |
| no lossy_coding_tool_constraint_flag              | u(1) |
| 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) |
| no_sao_constraint_flag                            | u(1) |
| no_alf_constraint_flag                            | u(1) |
| no_rndl_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) |
| gci_num_reserved_bytes                            | u(8) |
| for( i = 0; i < gci_num_reserved_bytes; i++ )     |      |
| gci_reserved_byte[ i ]                            |      |
| }                                                 | u(8) |

图13

1410 ↗

|                                                                       | 描述    |
|-----------------------------------------------------------------------|-------|
| profile tier level( profileTierPresentFlag, maxNumSubLayersMinus1 ) { |       |
| if( profileTierPresentFlag ) {                                        |       |
| <b>general profile idc</b>                                            | u(7)  |
| <b>general tier flag</b>                                              | u(1)  |
| <b>num available constraint flags</b>                                 | u(8)  |
| general constraint info( num available constraint flags )             |       |
| }                                                                     |       |
| <b>general level idc</b>                                              | u(8)  |
| if( profileTierPresentFlag ) {                                        |       |
| <b>num sub profiles</b>                                               | u(8)  |
| for( i = 0; i < num sub profiles; i++ )                               |       |
| <b>general sub profile idc[ i ]</b>                                   | u(32) |
| }                                                                     |       |
| for( i = 0; i < maxNumSubLayersMinus1; i++ )                          |       |
| <b>sublayer level present flag[ i ]</b>                               |       |
| while( !byte aligned() )                                              |       |
| <b>ptl alignment zero bit</b>                                         | f(1)  |
| for( i = 0; i < maxNumSubLayersMinus1; i++ )                          |       |
| if( sublayer level present flag[ i ] )                                |       |
| <b>sublayer level idc[ i ]</b>                                        | u(8)  |
| }                                                                     |       |

1401 ↘

图14A

1402 ↗ 1420 ↘

```

general constraint info( num available constraint flags ) {
    count = 0
    if( count < num available constraint flags )
        general non packed constraint flag
        count++
    if( count < num available constraint flags )
        general frame only constraint flag
        count++
    if( count < num available constraint flags )
        general non projected constraint flag
        count++
    ...
    while(!byte aligned())
        gci alignment zero bit
        gci num reserved bytes
        for(i = 0; i < gci num reserved bytes; i++)
            gci reserved byte[i]
}

```

图14B

|      |                                                              | 描述   |
|------|--------------------------------------------------------------|------|
| 1501 | general_constraint_info0 {                                   |      |
| 1502 | <b>general_frame_structure_constraint_group_flag</b>         | u(1) |
| 1503 | <b>high_level_functionality_constraint_group_flag</b>        | u(1) |
| 1504 | <b>scalability_constraint_group_flag</b>                     | u(1) |
| 1505 | <b>partitioning_constraint_group_flag</b>                    | u(1) |
| 1506 | <b>intra_coding_tool_constraint_group_flag</b>               | u(1) |
| 1507 | <b>inter_coding_tool_constraint_group_flag</b>               | u(1) |
| 1508 | <b>transform_constraint_group_flag</b>                       | u(1) |
|      | <b>inloop_filtering_constraint_group_flag</b>                | u(1) |
|      | ...                                                          |      |
|      | <b>if( general_frame_structure_constraint_group_flag ) {</b> |      |
| 1511 | <b>general_non_packed_constraint_flag</b>                    | u(1) |
| 1512 | <b>general_frame_only_constraint_flag</b>                    | u(1) |
| 1513 | <b>general_non_projected_constraint_flag</b>                 | u(1) |
| 1514 | <b>general_one_picture_only_constraint_flag</b>              | u(1) |
|      | <b>}</b>                                                     |      |
|      |                                                              | 1510 |

图15A

1500

续图15A

1520

```

if( high_level_functionality & constraint_group_flag ) {
    pic_header_in_slice_header_constraint_flag|-----|
    no ral1 constraint flag|-----|
    no idr constraint flag|-----|
    no cra constraint flag|-----|
    no gdr constraint flag|-----|
    no aps constraint flag|-----|
    ...|-----|
} if(intra_coding_tool_constraint_group_flag ) {
    ...
}
} if(inter_coding_tool_constraint_group_flag ) {
    ...
}
} 1430
while( !byte_aligned() )|-----|
|-----|-----|
|-----|-----|
|-----|-----|
|-----|-----|
|-----|-----|
|-----|-----|
for( i = 0; i < gci_num_reserved_bytes; i++ )
    gci_reserved_byte[ i ]|-----|
    u(8)|-----|
}

```

图15B

1600  
~~~~~

		描述
1601	gci num additional bits	u(1)
1602	if(gci num additional bits>0) {	u(8)
1603	general no extended precision constraint flag	u(1)
1604	general no ts residual coding rice present in sh constraint flag	u(1)
1605	general no rrc rice extension constraint flag	u(1)
	general no persistent rice adaptation constraint flag	u(1)
	general no reserve last sig coeff constraint flag	u(1)
	numAdditionalBitsUsed=5	
	}	
	else	
	numAdditionalBitsUsed=0	
	for(i=0; i<gci num additional bits-numAdditionalBitsUsed;i++)	
	gci reserved zero bit[i]	u(1)
	}	

图16

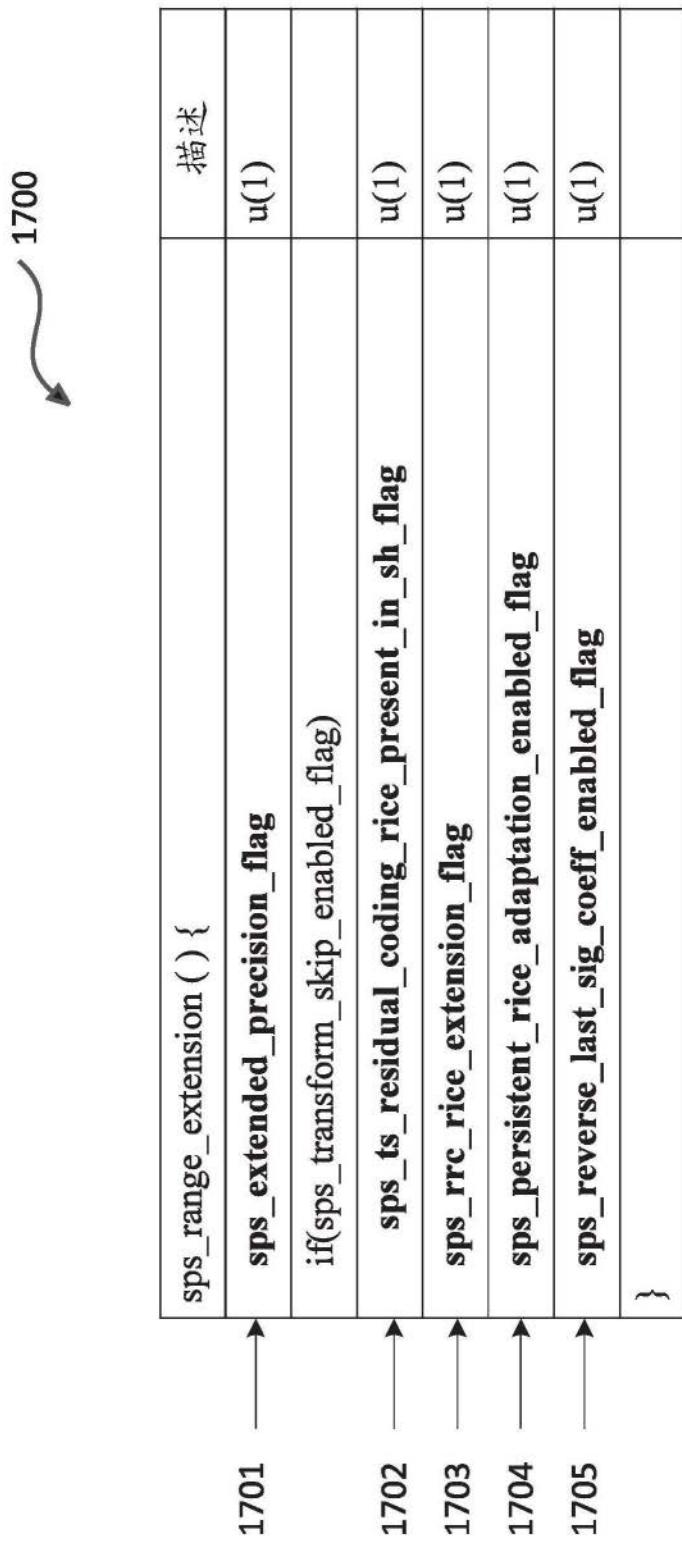


图17

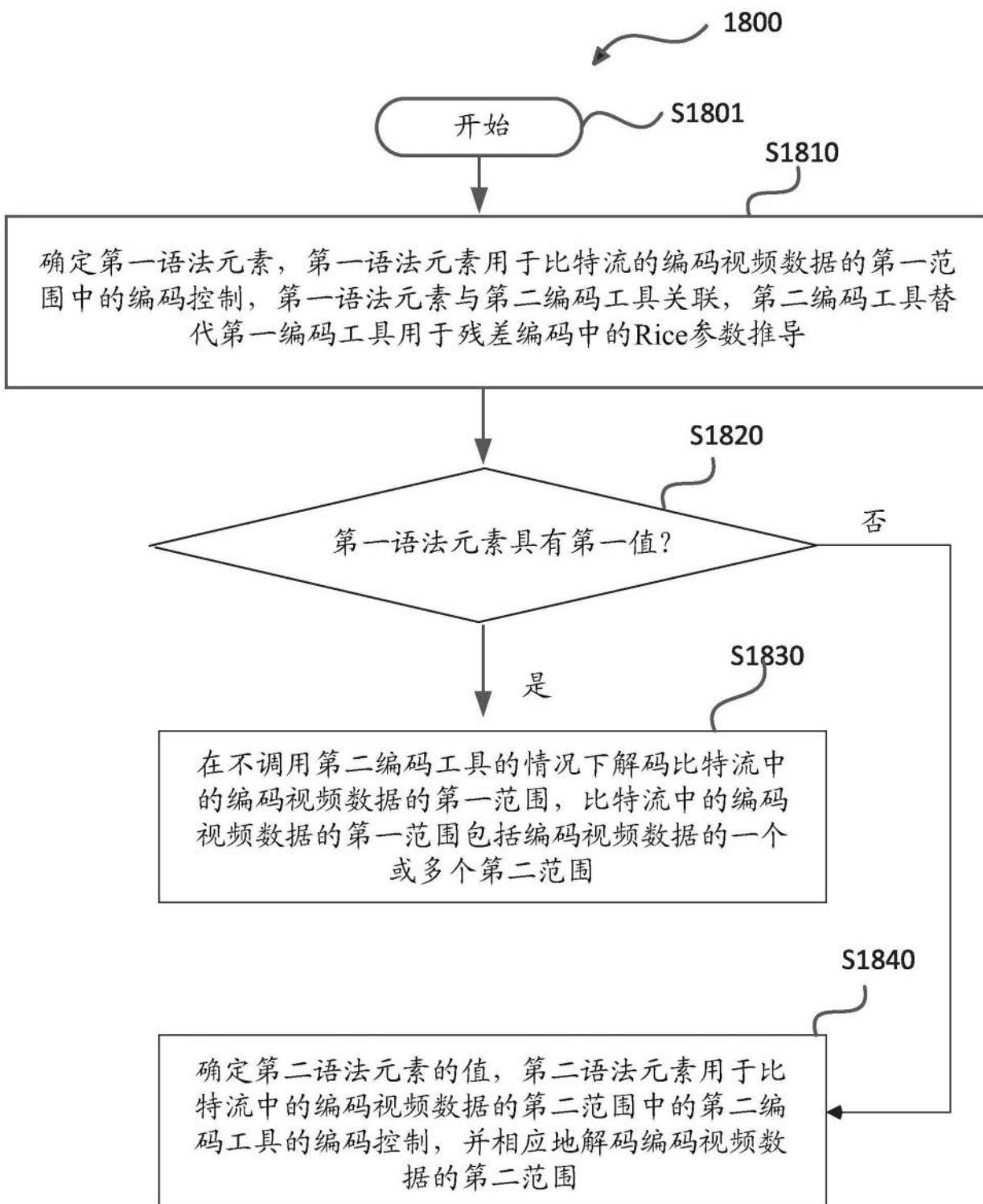


图18

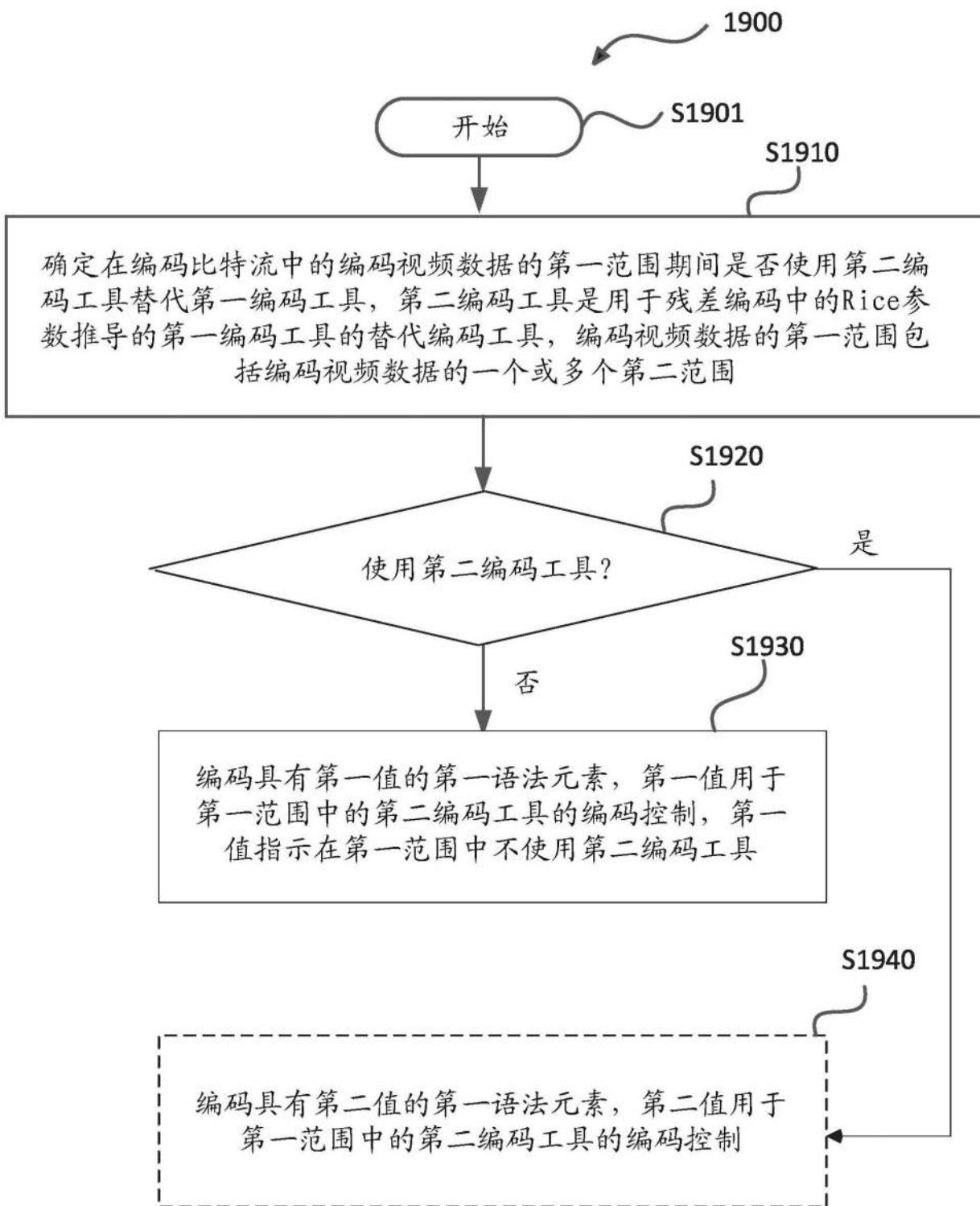


图19

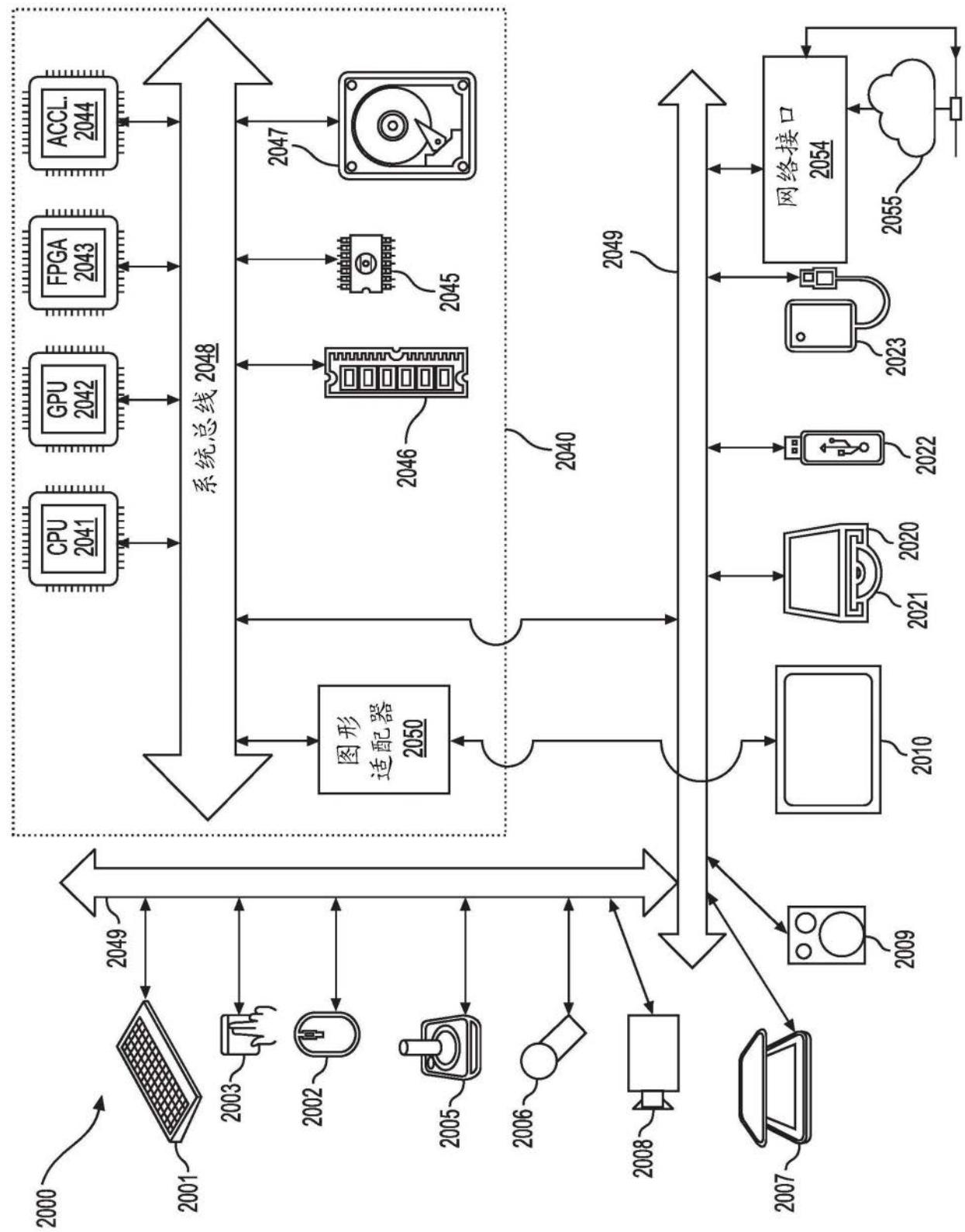


图20