



(12) 发明专利申请

(10) 申请公布号 CN 119299715 A

(43) 申请公布日 2025. 01. 10

(21) 申请号 202411479916.X

(22) 申请日 2020.09.18

(30) 优先权数据

62/903,652 2019.09.20 US

17/019,713 2020.09.14 US

(62) 分案原申请数据

202080027117.2 2020.09.18

(71) 申请人 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 史蒂芬·文格尔 刘杉

(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 焦方佼 王琦

(51) Int.Cl.

H04N 19/39 (2014.01)

H04N 19/33 (2014.01)

H04N 19/70 (2014.01)

H04N 19/507 (2014.01)

H04N 19/105 (2014.01)

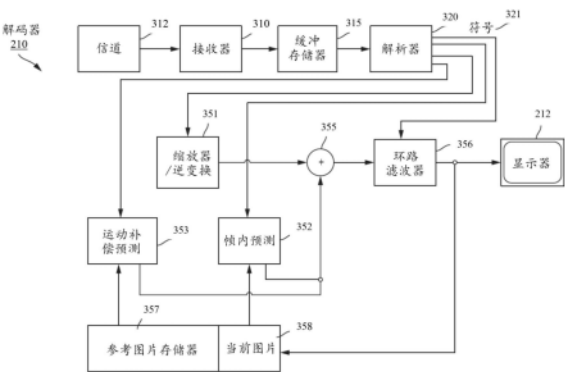
权利要求书2页 说明书19页 附图7页

(54) 发明名称

视频编码、解码方法、装置和介质

(57) 摘要

本公开提供了一种视频编码、解码方法、装置和介质。该装置包括计算机代码。该计算机代码用于使得一个或多个处理器解析至少一个视频参数集。该至少一个视频参数集包括至少一个语法元素，该至少一个语法元素指示可伸缩比特流中的至少一个层是否是以下二者之一：(i) 所述可伸缩比特流中的依赖层；(ii) 所述可伸缩比特流的独立层；基于所述视频参数集中包含的多个标志，确定所述可伸缩比特流中包含所述依赖层的依赖层数量；通过解析和解释层间参考图片列表，对所述依赖层中的图片进行解码；以及，在不解析和不解释所述层间参考图片列表的情况下，对所述独立层中的图片进行解码。



1. 一种视频解码方法,其特征在于,所述方法包括:
接收包括第一层的可伸缩比特流;
至少部分地基于所述可伸缩比特流的语法元素,确定所述第一层是否是所述可伸缩比特流的依赖层;
当所述第一层是依赖层时,基于层间参考图片ILRP列表,对所述依赖层中的图片进行解码,其中,所述层间参考图片ILRP列表的大小对应于所述可伸缩比特流中指示的直接参考层的数量;
当所述第一层不是依赖层时,在不使用所述ILRP列表的情况下,对所述第一层中的图片进行解码。
2. 根据权利要求1所述的方法,进一步包括:通过解析和解释不包括另一层的任何已解码图片的参考图片列表,对独立层中的第二图片进行解码。
3. 根据权利要求1所述的方法,其特征在于,所述层间参考图片ILRP列表包括不同层的已解码图片。
4. 根据权利要求1所述的方法,进一步包括:确定另一语法元素是否指示最大层数。
5. 根据权利要求1所述的方法,进一步包括:
确定所述可伸缩比特流的参数集是否包括一个标志,其中,所述标志指示所述可伸缩比特流中的不同层是否是至少一个层的参考层。
6. 根据权利要求5所述的方法,进一步包括:
通过指定所述不同层的索引和所述至少一个层的索引,确定所述标志是否指示所述不同层是所述至少一个层的参考层,以及,
确定所述参数集是否包括另一语法标志,其中,所述另一语法标志指示小于所确定的依赖层数量的值。
7. 根据权利要求5所述的方法,进一步包括:
通过指定所述不同层的索引和所述至少一个层的索引,确定所述标志是否指示所述不同层不是所述至少一个层的参考层,以及
确定所述参数集是否包括另一语法元素,其中,所述另一语法元素指示小于所确定的依赖层数量的值。
8. 根据权利要求1所述的方法,进一步包括:
确定所述可伸缩比特流的参数集是否包含一个标志,其中,所述标志指示是否通过解释所述层间参考图片ILRP列表对多个层进行解码。
9. 根据权利要求1所述的方法,进一步包括:
确定所述可伸缩比特流的参数集是否包含一个标志,其中,所述标志指示是否无需通过解释所述层间参考图片ILRP列表对多个层进行解码。
10. 根据权利要求1所述的方法,进一步包括:
在所述可伸缩比特流中,使用帧间预测参考图片IPRP列表发信号通知所述层间参考图片ILRP列表。
11. 一种视频编码方法,其特征在于,所述方法包括:
设置可伸缩比特流的语法元素,指示第一层是否是所述可伸缩比特流的依赖层;
当指示所述第一层是依赖层时,基于层间参考图片ILRP列表,对所述依赖层中的图片

进行编码,其中,所述层间参考图片ILRP列表的大小对应于所述可伸缩比特流中指示的直接参考层的数量;

当指示所述第一层不是依赖层时,在不使用所述ILRP列表的情况下,对所述第一层中的图片进行编码。

12.一种电子设备,包括:

控制电路;

存储器,以及,

存储在所述存储器中并配置为由所述控制电路执行的一组或多组指令,其中,所述一组或多组指令包括用于执行如权利要求1至11任一项所述的方法。

13.一种存储一组或多组指令的非易失性计算机可读存储介质,其特征在于,所述一组或多组指令被配置为由包含控制电路和存储器的计算设备执行,所述一组或多组指令包括用于执行如权利要求1至11任一项所述的方法。

14.一种处理视频比特流的方法,其特征在于,所述视频比特流根据权利要求11所述的编码方法产生,或者所述视频比特流基于权利要求1至10任一项所述的解码方法进行解码。

视频编码、解码方法、装置和介质

[0001] 引用并入

[0002] 本公开要求于2019年9月20日提交的美国临时申请第62/903,652号的优先权,以及2020年9月14日提交的美国申请第17/019,713号的优先权,在先申请的全部内容并入本文。

背景技术

[0003] 1. 领域

[0004] 所公开的主题涉及视频编码和解码,更具体地,涉及视频比特流中层间预测的信令。

[0005] 2. 相关技术的描述

[0006] 在几十年里,众所周知,可以通过具有运动补偿的帧间图片预测技术进行视频编码和解码。未压缩的数字视频可包括一系列图片,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个图片或60Hz。未压缩的视频具有非常大的比特率要求。例如,每个样本8比特的1080p60 4:2:0的视频(1920×1080 亮度样本分辨率,60Hz帧率)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。

[0007] 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信息。视频压缩可以帮助降低对上述带宽或存储空间的要求,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。

[0008] 视频编码器和解码器可利用几大类技术,例如包括:运动补偿、变换、量化和熵编码,下面将介绍其中的一些技术。

[0009] 历史上,视频编码器和解码器倾向于对给定图片尺寸进行操作,在大多数情况下,该给定图片尺寸是针对已编码视频序列(Coded Video Sequence, CVS)、图片群组(Group of Pictures, GOP)或类似的多图片时间帧而定义并保持不变。例如,在MPEG-2中,已知系统设计根据比如场景活动之类的因素改变水平分辨率(并且,从而改变图片尺寸),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,在CVS内使用不同分辨率的参考图片的重采样是已知的。然而,这里的图片尺寸没有改变,只有参考图片被重采样,可能导致只使用部分图片画布(在下采样的情况下),或者仅捕获部分场景(在上采样的情况下)。此外,H.263附录Q允许以因子2(在每一维度中)对单个宏块进行向上或向下重采样。同样,图片尺寸保持相同。在H.263中,宏块的尺寸是固定的,并且因此不需要发信号通知宏块的尺寸。

[0010] 在现代视频编解码中,预测图片中图片大小的变化变得更加主流。例如,VP9允许参考图片重采样(Reference Picture Resampling,RPR)和改变整个图片的分辨率。类似地,针对通用视频编解码(Versatile Video Coding,VVC),提出了某些建议(包括,例如,Hendry等人的“关于VVC的自适应分辨率改变(On adaptive resolution change (ARC) for VVC)”,联合视频小组文件JVET-M0135-v1,2019年1月9日至19日,整体并入本文),允许将整个参考图片重采样为不同的更高或更低的分辨率。在该文档中,建议将不同的候选分辨率在序列参数集(Sequence Parameter Set,SPS)中进行编码,并且由图片参数集(Picture Parameter Set,PPS)中的每个图片语法元素引用。

发明内容

[0011] 为了解决一个或多个不同的技术问题,本公开描述了设计用于缩放视频比特流中的信令的新语法及其使用。因此,可以实现改进的(解)编码效率。

[0012] 根据本文的实施例,使用参考图像重采样(Reference Picture Resampling,RPR)或自适应分辨率更改(Adaptive Resolution Change,ARC),可通过修改高级语法(High-Level Syntax,HLS)实现可扩展性支持的额外负担。在技术方面,在可扩展系统中采用层间预测提高增强层的编解码效率。除了在单层编解码器中可用的空间和时间运动补偿预测之外,层间预测还使用来自参考层的重建的参考图片的重采样视频数据来预测当前增强层。然后,通过修改现有的用于运动补偿的插值过程,在块级执行用于层间预测的重采样过程。这意味着不需要额外的重采样过程来支持可扩展性。在本公开中,公开了使用RPR支持空间/质量可伸缩性的高级语法元素。

[0013] 包括一种方法和装置,该装置包括存储器和一个或多个处理器,其中,所述存储器用于存储计算机程序代码,所述一个或多个处理器用于访问所述计算机程序代码,并按照计算机程序代码的指示进行操作。所述计算机程序代码包括:解析代码,用于使得所述至少一个处理器解析至少一个视频参数集VPS,所述至少一个VPS包括至少一个语法元素,其中,所述至少一个语法元素指示所述可伸缩比特流中的至少一个层是否是以下之一:(i)所述可伸缩比特流的依赖层;(ii)所述可伸缩比特流的独立层;确定代码,用于使得所述至少一个处理器基于VPS中包括的多个标志,确定所述可伸缩比特流中包含所述依赖层的依赖层的数量;第一解码代码,用于使得所述至少一个处理器通过解析和解释层间参考图片ILRP列表,对所述依赖层中的图片进行解码;以及,第二解码代码,用于使得所述至少一个处理器在不解析和不解释所述ILRP列表的情况下,对所述独立层中的图片进行解码。

[0014] 根据实施例,所述第二解码代码进一步用于使得所述至少一个处理器:通过解析和解释参考图片列表,对所述独立层中的图片进行解码,其中,所述参考图片列表不包括另一层的任何已解码图片。

[0015] 根据实施例,所述层间参考图片列表包括另一层的已解码图片。

[0016] 根据实施例,所述解析代码进一步使得所述至少一个处理器:通过确定另一语法元素是否指示最大层数,解析所述至少一个VPS。

[0017] 根据实施例,所述解析代码进一步使得所述至少一个处理器:通过确定所述VPS是否包括一个标志,其中,所述标志指示所述可伸缩比特流中的另一层是否是所述至少一个层的参考层,解析所述至少一个VPS。

[0018] 根据实施例,所述解析代码进一步使得所述至少一个处理器:通过指定另一层的索引和所述至少一个层的索引,确定所述标志是否指示所述另一层作为所述至少一个层的参考层,解析所述至少一个VPS,并且所述解析代码进一步使得所述至少一个处理器:通过确定所述VPS是否包含另一个语法元素,其中,所述另一个语法元素指示小于确定的依赖层数量的值,解析所述至少一个VPS。

[0019] 根据实施例,所述解析代码进一步使得所述至少一个处理器:通过指定另一层的索引和所述至少一个层的索引,确定所述标志是否指示所述另一层不是所述至少一个层的参考层,解析所述至少一个VPS,并且,所述解析代码进一步使得所述至少一个处理器:通过确定所述VPS是否包含另一个语法元素,其中,所述另一个语法元素指示小于确定的依赖层数量的值,解析所述至少一个VPS。

[0020] 根据实施例,所述解析代码进一步使得所述至少一个处理器:通过确定所述VPS是否包括一个标志,所述标志指示是否要通过解释所述ILRP列表对包括所述至少一个层的多个层进行解码,解析所述至少一个VPS。

[0021] 根据实施例,所述解析代码进一步使得所述至少一个处理器:通过确定所述VPS是否包括一个标志,所述标志指示是否要在不解释所述ILRP列表的情况下对包括所述至少一个层的多个层进行解码,解析所述至少一个VPS。

[0022] 根据实施例,所述解析代码进一步使得所述至少一个处理器解析所述至少一个VPS还包括:确定所述VPS是否包括一个标志,所述标志指示是否通过解释所述ILRP列表,对包括所述至少一个层的多个层进行解码。

附图说明

[0023] 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:

[0024] 图1是根据实施例的通信系统的简化框图的示意图。

[0025] 图2是根据实施例的通信系统的简化框图的示意图。

[0026] 图3是根据实施例的解码器的简化框图的示意图。

[0027] 图4是根据实施例的编码器的简化框图的示意图。

[0028] 图5A是根据相关技术的发信号通知ARC/RPR参数的选项的示意图。

[0029] 图5B是根据相关技术的发信号通知ARC/RPR参数的选项的示意图。

[0030] 图5C是根据实施例的发信号通知ARC/RPR参数的选项的示意图。

[0031] 图5D是根据实施例的发信号通知ARC/RPR参数的选项的示意图。

[0032] 图5E是根据实施例的发信号通知ARC/RPR参数的选项的示意图。

[0033] 图6是根据实施例的信令图片分辨率的示意图。

[0034] 图7是根据实施例的SPS中的信令图片大小和一致性窗口的示意图。

[0035] 图8是根据实施例的SPS中信令层间预测存在的示意图。

[0036] 图9是根据实施例的在切片头中信令层间预测索引的示意图。

[0037] 图10是根据实施例的计算机系统的示意图。

具体实施方式

[0038] 下面讨论的建议的特征可以单独使用或以任何顺序组合使用。此外,可以通过处理电路(例如,一个或多个处理器,或,一个或多个集成电路)实现实施例。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。

[0039] 图1示出了根据本公开实施例的通信系统(100)的简化框图。系统(100)可以包括通过网络(150)互联的至少两个终端装置(110和120)。对于单向数据传输,第一终端装置(110)可在本地位置对视频数据进行编码,以通过网络(150)传输到另一终端装置(120)。第二终端装置(120)可从网络(150)接收另一终端装置的已编码视频数据,对已编码视频数据进行解码并显示恢复的视频数据。单向数据传输在媒体服务等应用中是较常见的。

[0040] 图1示出了支持已编码视频的双向传输的第二对终端装置(130,140),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,每个终端装置(130,140)可对在本地位置采集的视频数据进行编码,以通过网络(150)传输到另一终端装置。每个终端装置(130,140)还可接收由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并在本地显示设备上显示恢复的视频数据。

[0041] 在图1中,终端装置(110,120,130,140)可为服务器、个人计算机和智能手机,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(150)表示在终端装置(110,120,130,140)之间传送已编码视频数据的任何数目的网络,包括例如有线和/或无线通信网络。通信网络(150)可在电路交换和/或分组交换信道中交换数据。代表性网络包括电信网络、局域网、广域网和/或互联网。出于本讨论的目的,除非在下文中有所解释,否则网络(150)的架构和拓扑对于本公开的操作来说可能是无关紧要的。

[0042] 作为所公开主题的应用实施例,图2示出了视频解码器和编码器在流式传输环境中的放置方式。所公开的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。

[0043] 流式传输系统可包括采集子系统(213),所述采集子系统可包括诸如数码相机等视频源(201),所述视频源创建例如未压缩的视频样本流(202)。较于已编码的视频比特流,样本流(202)被描绘为粗线,以强调其为高数据量的视频样本流,样本流(202)可由耦接到相机(201)的编码器(203)处理。编码器(203)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。相较于样本流(202),已编码的视频比特流(204)被描绘为细线,以强调较低数据量的已编码的视频比特流,其可存储在流式传输服务器(205)上以供将来使用。一个或多个流式传输客户端(206,208)可访问流式传输服务器(205)以检索已编码的视频比特流(204)的副本(207,209)。客户端(206)可包括视频解码器(210)。视频解码器(210)对已编码的视频比特流的传入副本(207)进行解码,且产生可在显示器(212)或另一呈现装置(未示出)上呈现的输出视频样本流(211)。在一些流式传输系统中,可以根据某些视频编解码/压缩标准对视频比特流(204,207,209)进行编码。这些标准的示例包括ITU-T H.265建议书。正在开发的视频编解码标准被非正式地称为多功能视频编解码(VVC)。所公开的主题可以在VVC的上下文中使用。

[0044] 图3是根据本公开实施例的视频解码器(210)的功能框图。

[0045] 接收器(310)可接收将由视频解码器(210)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(312)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(310)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(310)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(315)可耦接在接收器(310)与熵解码器/解析器(320)(此后称为“解析器”)之间。当接收器(310)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(315),或可以将所述缓冲存储器做得较小。当然,为了在互联网等分组网络上使用,也可能需要缓冲存储器(315),所述缓冲存储器可相对较大且可有利地具有自适应性大小。

[0046] 视频解码器(210)可包括解析器(320)以根据熵编码视频序列重建符号(321)。这些符号的类别包括用于管理视频解码器(210)的操作的信息,以及用以控制诸如显示器212的显示装置的潜在信息,所述显示装置不是解码器的组成部分,但可耦接到解码器,如图3中所示。用于显示装置的控制信息可以是辅助增强信息(Supplementary 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)的形式。解析器(320)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解码可根据视频编解码技术或标准进行,且可遵循本领域技术人员公知的原理,包括可变长度编解码、霍夫曼编解码(Huffman coding)、具有或不具有上下文灵敏度的算术编解码等等。解析器(320)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(tile)、切片(slice)、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。熵解码器/解析器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。

[0047] 解析器(320)可对从缓冲存储器(315)接收的视频序列执行熵解码/解析操作,从而创建符号(321)。

[0048] 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(321)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(320)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(320)与下文的多个单元之间的此类子群控制信息流。

[0049] 除已经提及的功能块以外,视频解码器210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中,这些单元中的许多单元彼此紧密交互并且可以至少部分地彼此集成。然而,出于描述所公开的主题的目的,概念上细分成下文的功能单元是适当的。

[0050] 第一单元是缩放器/逆变换单元(351)。缩放器/逆变换单元(351)从解析器(320)接收作为符号(321)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(351)可输出包括样本值的块,所述样本值可输入到聚合器(355)中。

[0051] 在一些情况下,缩放器/逆变换单元(351)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(352)提供。在一些情况下,帧内图片预测单元(352)采用从(部分重建的)当前图片(358)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。在一些情况下,聚合器(355)基于每个样本,将帧内预测单元(352)生成的预测信息添加到由缩放器/逆变换单元(351)提供的输出样本信息中。

[0052] 在其它情况下,缩放器/逆变换单元(351)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(353)可访问参考图片存储器(357)以提取用于预测的样本。在根据符号(321)对提取的样本进行运动补偿之后,这些样本可由聚合器(355)添加到缩放器/逆变换单元的输出(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。运动补偿预测单元从参考图片存储器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(321)的形式而供运动补偿预测单元(353)使用,所述符号(321)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(357)提取的样本值的内插、运动矢量预测机制等等。

[0053] 聚合器(355)的输出样本可在环路滤波器单元(356)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比特流中的参数,且所述参数作为来自解析器(320)的符号(321)可用于环路滤波器单元(356)。然而,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。

[0054] 环路滤波器单元(356)的输出可以是样本流,所述样本流可输出到显示装置(212)以及存储在参考图片存储器(357),以用于后续的帧间图片预测。

[0055] 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。一旦已编码图片被完全重建,且已编码图片(通过例如解析器(320))被识别为参考图片,则当前图片(358)可变为参考图片存储器(357)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存储器。

[0056] 视频解码器(210)可根据例如ITU-T H.265标准中记录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在视频压缩技术文献或标准、特别是配置文件中所规定的视频压缩技术或标准的语法的意义上,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。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(Hypothetical Reference Decoder,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。

[0057] 在实施例,接收器(310)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(210)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(signal noise ratio,SNR)增强层、冗余切片、冗余图片、前向纠错码等形式。

[0058] 图4是根据本公开实施例的视频编码器(203)的功能框图。

[0059] 视频编码器(203)可从视频源(201)(并非该编码器的一部分)接收视频样本,所述视频源可采集将由视频编码器(203)编码的视频图像。

[0060] 视频源 (201) 可提供将由视频编码器 (203) 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.601Y CrCb、RGB……) 和任何合适取样结构 (例如Y CrCb 4:2:0、Y CrCb 4:4:4)。在媒体服务系统中,视频源 (201) 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 (201) 可以为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。本领域技术人员可以容易地理解像素和样本之间的关系。下文侧重于描述样本。

[0061] 根据实施例,视频编码器 (203) 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 (443)。施行适当的编码速度是控制器 (450) 的一个功能。控制器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。由控制器设置的参数可包括速率控制相关参数 (例如,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 (group of pictures, GOP) 布局,最大运动矢量搜索范围等。本领域技术人员可以容易地识别控制器 (450) 的其他功能,因为它们可能属于针对特定系统设计而优化的视频编码器 (203)。

[0062] 一些视频编码器以本领域技术人员容易识别为“编码环路”方式进行操作。作为简单的描述,编码环路可包括编码器 (430) (下文称为“源编码器 (source coder)”,其负责基于待编码的输入图片和参考图片创建符号) 的编码部分、和嵌入于视频编码器 (203) 中的 (本地) 解码器 (433)。“本地”解码器 (433) 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 (因为在所公开的主题所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)。将重建的样本流输入到参考图片存储器 (434)。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) 是本领域技术人员公知的。

[0063] “本地”解码器 (433) 的操作可与已在上文结合图4详细描述“远程”解码器 (210) 相同。然而,另外简要参考图4,当符号可用且熵编码器 (445) 和解析器 (320) 能够无损地将符号编码/解码为已编码视频序列时,视频解码器 (210) 的熵解码部分 (包括信道 (312)、接收器 (310)、缓冲存储器 (315) 和解析器 (320)) 可能无法完全在本地解码器 (433) 中实施。

[0064] 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。由于这个原因,所公开的主题专注于解码器操作。对编码器技术的描述可以简写,因为它们与全面描述解码器技术互逆。仅在某些区域中需要更详细的描述,并且在下文提供。

[0065] 作为操作的一部分,源编码器 (430) 可执行运动补偿预测编码。参考来自视频序列中被指定为“参考帧”的一个或多个先前已编码帧,所述运动补偿预测编码对输入帧进行预测性编码。以此方式,编码引擎 (432) 对输入帧的像素块与参考帧的像素块之间的差异进行编码,所述参考帧可被选作所述输入帧的预测参考。

[0066] 本地视频解码器 (433) 可基于源编码器 (430) 创建的符号,对可指定为参考帧的帧

的已编码视频数据进行解码。编码引擎 (432) 的操作可以有利地为有损过程。当已编码视频数据可在视频解码器 (图4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 (433) 复制解码过程,所述解码过程可由视频解码器对参考帧执行,且可使重建的参考帧存储在参考图片存储器 (434) 中。以此方式,编码器 (203) 可在本地存储重建的参考帧的副本,所述副本与将由远端视频解码器获得的重建参考帧具有共同内容 (不存在传输误差)。

[0067] 预测器 (435) 可针对编码引擎 (432) 执行预测搜索。即,对于将要编码的新帧,预测器 (435) 可在参考图片存储器 (434) 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 (435) 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 (435) 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 (434) 中存储的多个参考图片取得的预测参考。

[0068] 控制器 (450) 可管理源编码器 (430) 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。

[0069] 可在熵编码器 (445) 中对所有上述功能单元的输出进行熵编码。熵编码器可根据例如霍夫曼编码、可变长度编码、算术编码等本领域技术人员公知的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。

[0070] 传输器 (440) 可缓冲由熵编码器 (445) 创建的已编码视频序列,从而为通过通信信道 (460) 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 (440) 可将来自源编码器 (430) 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)。

[0071] 控制器 (450) 可管理视频编码器 (203) 的操作。在编码期间,控制器 (450) 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。

[0072] 帧内图片 (I图片),其可以是不将序列中的任何其它帧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 (Independent Decoder Refresh, IDR) 图片。本领域技术人员知道I图片的那些变体以及它们各自的应用和特征。

[0073] 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。

[0074] 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。

[0075] 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非预测编码。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非预测编

码。

[0076] 视频编码器(203)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(203)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。

[0077] 在实施例中,传输器(440)可传输附加数据和已编码的视频。源编码器(430)可以包括诸如已编码视频序列的一部分的此类数据。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、SEI消息、VUI参数集片段等。

[0078] 在更详细地描述所公开主题的某些方面之前,需要介绍一些术语,这些术语将在本说明书的其余部分中提及。

[0079] 此后,在某些情况下,子图片指的是样本、块、宏块、编码单元或类似实体的矩形排列,对这些样本、块、宏块、编码单元或类似实体进行语义分组,并且可以使用改变的分辨率独立地编码。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装成图片,并且可以从图片中提取一个或多个子图片。在某些环境中,可以在压缩域中组装一个或多个已编码子图片,而无需转码到样本级别称为已编码图片。并且在相同或某些其它情况下,可以从该压缩域中的已编码图片中提取一个或多个已编码子图片。

[0080] 此后,参考图片重采样(Reference Picture Resampling,RPR)或自适应分辨率改变(Adaptive Resolution Change,ARC)指的是允许通过例如参考图片重采样,改变已编码视频序列内的图片或子图片的分辨率的机制。此后,RPR/ARC参数是指执行自适应分辨率改变所需的控制信息,可以包括例如滤波器参数、缩放因子、输出和/或参考图片的分辨率、各种控制标志等。

[0081] 以上描述集中于对单一的、语义上独立的已编码视频图片的编码和解码。在描述具有独立RPR/ARC参数的多个子图片的编码/解码的含义及其隐含的额外复杂度之前,将描述用于发信号通知RPR/ARC参数的选项。

[0082] 参考图5,示出了用于发信号通知RPR/ARC参数的几个新颖选项。如选项中的每一个所指出的,从编解码效率、复杂度和架构的角度来看,这些选项具有某些优点和某些缺点。视频编解码标准或技术可以选择这些选项中的一个或多个,或现有技术中已知的选项,用于发信号通知RPR/ARC参数。这些选项可以不是互斥的,可以基于应用需求、所涉及的标准技术或编码器的选择而进行互换。

[0083] RPR/ARC参数的类别可包括:

[0084] - 上/下采样因子,在X和Y维度上分开或组合,

[0085] - 上/下采样因子,加上时间维度,指示对于给定数量图片的恒定速度放大/缩小,

[0086] - 上述两个中的任何一个都可以涉及对一个或多个可能很短的语法元素的编解码,该语法元素可以指向包含一个或多个因子的表,

[0087] - X或Y维度中输入图片、输出图片、参考图片、已编码图片的分辨率,该分辨率以样本、块、宏块、编码单元(Coding Unit,CU)或任何其它合适的粒度为单位,并且是组合或单独的(如果存在多于一个分辨率(例如,一个分辨率用于输入图片,一个分辨率用于参考图片)),则在某些情况下,可从另一组值中推断出一组值。这可以例如通过使用标志来选通。

对于更多详细示例,如下可见:

[0088] - “翘曲”坐标,类似于在H.263附录P中使用的那些,并且以如上所述的合适的粒度(H.263附录P定义了对这种翘曲坐标进行编码的一种有效方式,但是也可以构想出其它可能更有效的方式。例如,根据实施例,附录P的翘曲坐标的可变长度可逆的“霍夫曼(Huffman)”型编解码由适当长度的二进制编解码代替,其中,例如,二进制码字的长度可以从最大图片大小导出、可能乘以某个因子并偏移某个值,从而允许在最大图片大小的边界之外“翘曲”),和/或

[0089] - 上采样或下采样滤波器参数(在最容易的情况下,可能只有单个滤波器用于上采样和/或下采样。然而,在某些情况下,在滤波器设计中允许更大的灵活性是有利的,并且这可能需要发信号通知滤波器参数。这样的参数可以通过可能的滤波器设计列表中的索引来选择,可以完全指定滤波器(例如,通过滤波器系数列表、使用适当的熵编解码技术),可以通过上/下采样比隐含地选择滤波器,根据该上/下采样比,继而又根据上述任何机制发信号通知该上/下采样比,等等)。

[0090] 此后,本说明书假设通过码字指示的上/下采样因子的有限集合的编解码(在X和Y维度中使用相同的因子)。可对该码字方便地进行可变长度编码,例如使用对于视频编解码规范(比如H.264和H.265)中的某些语法元素通用的哥伦布(Ext-Golomb)码。一个合适的从值到上/下采样因子的映射可以如下表1所示:

[0091] 表1

[0092]	码字(Codeword)	哥伦布代码 (Ext-Golomb Code)	原始/目标分辨率 (Original / Target resolution)
[0093]	0	1	1 / 1
	1	010	1 / 1.5 (提升 50%)
	2	011	1.5 / 1 (降低 50%)
	3	00100	1 / 2 (提升 100%)
	4	00101	2 / 1 (降低 100%)

[0094] 根据应用的需要和视频压缩技术或标准中可用的放大和缩小机制的能力,可以设计很多类似的映射。该表可以扩展到更多的值。值也可以由除哥伦布(Ext-Golomb)码之外的熵编解码机制表示,例如,使用二进制编解码。当重采样因子在视频处理引擎(首先是编码器和解码器)之外感兴趣时,例如MANE,这可能具有某些优势。应该注意的是,对于不需要更改分辨率的(大概)最常见的情况,可以选择较短的哥伦布代码。在上表中,只有一位。对

于最常见的情况,这比使用二进制代码具有编解码效率优势。

[0095] 表中条目的数量以及它们的语义可以是完全或部分可配置的。例如,可以在比如序列参数集或解码器参数集的“高”参数集中传达该表的基本要点。可选地或另外地,可以在视频编解码技术或标准中定义,并且可以通过例如解码器参数集或序列参数集选择一个或多个这样的表。

[0096] 此后,我们描述如何在视频编解码技术或标准语法中包含如上所述已编码的上采样/下采样因子 (ARC信息)。类似的考虑可以应用于控制上/下采样滤波器的一个或几个码字。当滤波器或其它数据结构需要相对大量的数据时,参见下面的讨论。

[0097] 如图5A的示例中所示,图示 (500A) 示出H. 263附录P将ARC信息 (502) 以四个翘曲坐标形式包含在图片头 (501) 中,具体地在H. 263PLUSPTYPE (503) 头扩展中。当a) 存在可用的图片头,以及b) 预期ARC信息频繁改变时,这可以是明智的设计选择。然而,当使用H. 263型信令时开销可能相当高,并且缩放因子可能不属于图片边界,因为图片头可能具有瞬态特征。此外,如图5B的示例所示,图示 (500B) 示出JVET-M0135包括PPS信息 (504)、ARC参考信息 (505)、SPS信息 (507) 和目标资源表信息 (506)。

[0098] 根据示例性实施例,图5C示出了示例 (500C),其中示出了图块组头信息 (508) 和ARC信息 (509)。图5D示出了示例 (500D),其中示出了图块组头信息 (514)、ARC参考信息 (513)、SPS信息 (516) 和ARC信息 (515),图5E示出了示例 (500E),其中示出了一个或多个自适应参数集 (APS) 信息 (511) 和ARC信息 (512)。

[0099] 图6示出了使用自适应分辨率的表的示例 (600),在该示例中,已编码的是以样本 (613) 为单位的输出分辨率。数字613指的是output_pic_width_in_luma_samples和output_pic_height_in_luma_samples,它们一起可以定义输出图片的分辨率。在视频编解码技术或标准的其它地方,可以定义对任一个值的某些限制。例如,级别定义可以限制总输出样本的数量,总输出样本的数量可以是那两个语法元素的值的乘积。并且,某些视频编解码技术或标准,或外部技术或标准 (例如,系统标准) 可以限制编号范围 (例如,一个或两个维度必须可被2的幂整除) 或纵横比 (例如,宽度和高度必须是4:3或16:9之类的关系)。为了促进硬件实现或鉴于本公开本领域的技术人员将了解的其它原因,可以引入此类限制。

[0100] 在某些应用中,可以建议编码器指示解码器使用某个参考图片尺寸,而不是隐含地假设该尺寸是输出图片尺寸。在该示例中,语法元素reference_pic_size_present_flag (614) 选通参考图片维度 (615) (同样,该数字指宽度和高度) 的条件存在。

[0101] 某些视频编解码技术或标准 (例如VP9) 通过结合时间可缩放性,实现某些形式的参考图片重采样 (采用与所公开的主题相当不同的方式发信号通知) 以支持空间可扩展性,从而实现了空间的可扩展性。具体地,可以使用ARC型技术,将某些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。可以使用高分辨率的常规预测机制修正那些上采样的图片,从而增加细节。

[0102] 根据实施例,所公开的主题可以在这样的环境中使用。在某些情况下,在同一或另一实施例中,NAL单元头中的值 (例如,时间ID字段) 可用于指示时间层和空间层。这样做对某些系统设计具有一些优势;例如,为时间层创建和优化的现有选定转发单元 (Selected Forwarding Units, SFU) 基于NAL单元头时间ID值选择转发,可以在不修改的情况下使用,用于可扩展环境。为了实现这一点,可能需要通过NAL单元头中的时间ID字段指示该已编码

图片尺寸和该时间层之间的映射。

[0103] 在实施例中,可以在VPS(或DPS、SPS或SEI消息)中发信号通知关于层间依赖性的信息。该层间依赖性信息可用于识别将哪一层用作参考层以对当前层进行解码。当n大于m并且两个图片picA和picB属于同一访问单元时,nuh_layer_id等于m的直接依赖层中的已解码图片picA可以用作nuh_layer_id等于n的图片picB的参考图片。

[0104] 在同一或其它实施例中,可以在切片头(或参数集)中,一起明确地发信号通知层间参考图片(inter-layer reference picture,ILRP)列表与帧间预测参考图片(inter-prediction reference picture,IPRP)列表。ILRP列表和IPRP列表都可用于构造前向和后向预测参考图片列表。

[0105] 在相同或其他实施例中,VPS(或其他参数集)中的语法元素可以指示每一层是相关的还是独立的。参考图7中的示例(700),语法元素vps_max_layers_minus1(703)加1可以指定一个或多个(可能是所有)CVS中允许的最大层数,参考VPS(701)。等于1的vps_all_independent_layers_flag(704)可以指定CVS中的所有层都是独立编码,即,不使用层间预测。等于0的vps_all_independent_layers_flag(704)可以指定CVS中的一个或多个层可以使用层间预测。当不存在时,可以推断vps_all_independent_layers_flag的值等于1。当vps_all_independent_layers_flag的值等于1时,推断vps_independent_layer_flag[i](706)的值等于1。当vps_all_independent_layers_flag等于0时,推断vps_independent_layer_flag[0]的值等于1。

[0106] 参考图7,vps_independent_layer_flag[i](706)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vps_layer_dependency_flag[i]存在于VPS中。vps_direct_dependency_flag[i][j](707)等于0可以指定索引为j的层不是索引为i的层的直接参考层。vps_direct_dependency_flag[i][j]等于1可以指定索引为j的层是索引为i的层的直接参考层。当i和j在0到vps_max_layers_minus1(包含vps_max_layers_minus1)的范围内,不存在vps_direct_dependency_flag[i][j]时,推断vps_direct_dependency_flag[i][j]的值等于0。

[0107] 指定第i层的第j个直接依赖层的变量DirectDependentLayerIdx[i][j],以及指定第i层的依赖层数的变量NumDependentLayers[i],推导如下:

```

for( i = 1; i < vps_max_layers_minus1; i++ )
    if( !vps_independent_layer_flag[ i ] ) {
        for( j = i, k = 0; j >= 0; j-- )
[0108]         if( vps_direct_dependency_flag[ i ][ j ] )
            DirectDependentLayerIdx[ i ][ k++ ] = j
        NumDependentLayers[ i ] = k
    }

```

[0109] 在相同或另一个实施例中,参考图7,当vps_max_layers_minus1的值大于0,并且vps_all_independent_layers_flag的值等于0时,可以发信号通知vps_output_layers_

mode和vps_output_layer_flags[i].vps_output_layers_mode(708)等于0可以指定只输出最高层。vps_output_layer_mode等于1指定可以输出所有层。vps_output_layer_mode等于2可以指定输出的层是vps_output_layer_flag[i](709)等于1的层。vps_output_layers_mode的值应在0到2(包括2)的范围内。vps_output_layer_mode的值3可能会保留以备将来使用。当不存在时,可以推断vps_output_layers_mode的值等于1。vps_output_layer_flag[i]等于1可以指定输出第i层。vps_output_layer_flag[i]等于0可以指定不输出第i层。

[0110] 列表OutputLayerFlag[i],其中值1可以指定输出第i层,值0指定不输出第i层,推导如下:

```

OutputLayerFlag[ vps_max_layers_minus1 ] = 1
for( i = 0; i < vps_max_layers_minus1; i++ )
    if( vps_output_layer_mode == 0 )
        OutputLayerFlag[ i ] = 0
[0111] else if( vps_output_layer_mode == 1 )
        OutputLayerFlag[ i ] = 1
    else if( vps_output_layer_mode == 2 )
        OutputLayerFlag[ i ] = vps_output_layer_flag[ i ]

```

[0112] 在相同或另一个实施例中,当前图片的输出可以指定如下:

[0113] -若PictureOutputFlag等于1,并且DpbOutputTime[n]等于CpbRemovalTime[n]时,输出当前图片。

[0114] -否则,若PictureOutputFlag等于0,不输出当前图片,但会按照子句中的规定将当前图片存储在DPB中。

[0115] -否则(PictureOutputFlag等于1,并且DpbOutputTime[n]大于CpbRemovalTime[n]),当前图片稍后输出,并将存储在DPB中(如在子句中指定的),并且在时间DpbOutputTime[n]输出,除非在DpbOutputTime[n]之前的时间通过no_output_of_prior_pics_flag等于1的解码或推断指示不输出。

[0116] 当输出时,使用PPS中为图片指定的一致性裁剪窗口裁剪图片。

[0117] 在相同或另一个实施例中,PictureOutputFlag可以设置如下:

[0118] -如果以下条件之一为真,将PictureOutputFlag设置为0:

[0119] -当前图片是RASL图片,并且关联的IRAP图片的

[0120] NoIncorrectPicOutputFlag等于1.

[0121] -gdr_enabled_flag等于1,并且当前图片是NoIncorrectPicOutputFlag等于1的GDR图片。

[0122] -gdr_enabled_flag等于1,当前图片与NoIncorrectPicOutputFlag等于1的GDR图片相关联,并且当前图片的PicOrderCntVal小于相关GDR图片的RpPicOrderCntVal。

[0123] -vps_output_layer_mode等于0或2,并且

[0124] OutputLayerFlag[GeneralLayerIdx[nuh_layer_id]]等于0。

[0125] -否则,将PictureOutputFlag设置为等于pic_output_flag。

- [0126] 在同一或其他实施例中,或者,PictureOutputFlag可以设置如下:
- [0127] -如果以下条件之一为真,可以将PictureOutputFlag设置为等于0:
- [0128] -当前图片是RASL图片,并且相关的IRAP图片的
- [0129] NoIncorrectPicOutputFlag等于1。
- [0130] -gdr_enabled_flag等于1,并且当前图片为NoIncorrectPicOutputFlag等于1的GDR图片。
- [0131] -gdr_enabled_flag等于1,当前图片与NoIncorrectPicOutputFlag等于1的GDR图片相关联,并且当前图片的PicOrderCntVal小于相关的GDR图片的RpPicOrderCntVal。
- [0132] -vps_output_layer_mode等于0,并且当前访问单元包含PictureOutputFlag
- [0133] 等于1的图片,其中,该图片的nuh_layer_id nuhLid大于当前图片的nuh_layer_id nuhLid,并且,该图片属于输出层(即,
- [0134] OutputLayerFlag[GeneralLayerIdx[nuhLid]]等于1)。
- [0135] -vps_output_layer_mode等于2,并且
- [0136] OutputLayerFlag[GeneralLayerIdx[nuh_layer_id]]等于0。
- [0137] -否则,将PictureOutputFlag设置为等于pic_output_flag。
- [0138] 在同一或其他实施例中,VPS(或另一个参数集)中的标志可以指示是否为当前切片(或图片)发信号通知ILRP列表。例如,参考图8中的示例(800),inter_layer_ref_pics_present_flag等于0可以指定没有ILRP用于CVS中任何已编码图片的帧间预测。inter_layer_ref_pics_flag等于1可以指定ILRPs可以用于CVS中一个或多个已编码图片的帧间预测。
- [0139] 在同一或其他实施例中,当第k层是从属层(dependent layer)时,可以发信号通知或不发信号通知第k层中图片的层间参考图片ILRP列表。但是,当第k层是独立层时,将不发信号通知第k层中图片的ILRP列表,并且,参考图片列表中不应包含任何ILRP。当sps_video_parameter_set_id等于0时,当nuh_layer_id等于0,或当vps_independent_layer_flag[GeneralLayerIdx[nuh_layer_id]]等于1时,可以将inter_layer_ref_pics_present_flag的值设为等于0。
- [0140] 在相同或其他实施例中,参考图9中的示例(900),可以在SPS、PPS、APS或切片头中发信号通知一组明确指示ILRP列表的语法元素。ILRP列表可用于构建当前图片的参考图片列表。
- [0141] 在相同或其他实施例中,ILRP列表可用于在解码图片缓冲区(Decoded Picture Buffer,DPB)中识别激活的(active)或非激活(inactive)的参考图片。激活的参考图片可用作解码当前图片的参考图片,而非激活的参考图片可能不会用于解码当前图片,但可用于按解码顺序对后续图片进行解码。
- [0142] 在相同或其他实施例中,ILRP列表可用于确定哪些参考图片可以存储在DPB中或可以从DPB中输出和删除。这些信息可用于基于假设参考解码器(Hypothetical Reference Decoder,HRD)模型和参数来操作解码器。
- [0143] 在相同或其他实施例中,可以在VPS、SPS、PPS、APS或切片头中发信号通知语法元素ilrp_idc[listIdx][rplsIdx][i]。语法元素ilrp_idc[listIdx][rplsIdx][i]指定直接依赖层列表的ref_pic_list_struct(listIdx,rplsIdx)语法结构中第i个条目的ILRP到直

接依赖层列表的索引。ilrp_idc[listIdx][rplsIdx][i]的值应在0到GeneralLayerIdx[nuh_layer_id]-1的范围内,包括GeneralLayerIdx[nuh_layer_id]-1。

[0144] 在相同的实施例中,语法元素ilrp_idc[listIdx][rplsIdx][i]可以是指直接依赖层之间的ILRP图片的索引,该索引由VPS中发信号通知的vps_direct_dependency_flag[i][j]标识。在这种情况下,ilrp_idc[listIdx][rplsIdx][i]的值应在0到NumDependentLayers[GeneralLayerIdx[nuh_layer_id]]-1的范围内,包括NumDependentLayers[GeneralLayerIdx[nuh_layer_id]]-1。

[0145] 在相同的实施例中,若当前层的nuh_layer_id等于k时,与在nuh_layer_id小于k的所有层中发信号通知指示ILRP的索引相比,发信号通知指示直接依赖层之间的ILRP的索引可能是比特有效的。

[0146] 在相同或另一个实施例中,仍然参考图9,参考图片列表RefPicList[0]和RefPicList[1]可以如下构造:

```

    for( i = 0; i < 2; i++ ) {
        for( j = 0, k = 0, pocBase = PicOrderCntVal; j < num_ref_entries[ i ][ RplsIdx[ i ] ]; j++ ) {
            if( !(inter_layer_ref_pic_flag[ i ][ RplsIdx[ i ] ][ j ] && GeneralLayerIdx[ nuh_layer_id ]) )
            {
                if( st_ref_pic_flag[ i ][ RplsIdx[ i ] ][ j ] ) {
                    RefPicPocList[ i ][ j ] = pocBase - DeltaPocValSt[ i ][ RplsIdx[ i ] ][ j ]
[0147]         if( there is a reference picture picA in the DPB with the same nuh_layer_id as the
                    current picture
                        and PicOrderCntVal equal to RefPicPocList[ i ][ j ] )
                        RefPicList[ i ][ j ] = picA
                else
                    RefPicList[ i ][ j ] = "no reference picture"
            }
        }
    }

```

```

        pocBase = RefPicPocList[ i ][ j ]
    } else {
        if( !delta_poc_msb_cycle_lt[ i ][ k ] ) {
            if( there is a reference picA in the DPB with the same nuh_layer_id as the
current picture and
                PicOrderCntVal & ( MaxPicOrderCntLsb - 1 ) equal to
PocLsbLt[ i ][ k ] )
                RefPicList[ i ][ j ] = picA
            else
                RefPicList[ i ][ j ] = "no reference picture"
                RefPicLtPocList[ i ][ j ] = PocLsbLt[ i ][ k ]
        } else {
            if( there is a reference picA in the DPB with the same nuh_layer_id as the
current picture and
                PicOrderCntVal equal to FullPocLt[ i ][ k ] )
                RefPicList[ i ][ j ] = picA
[0148] else
                RefPicList[ i ][ j ] = "no reference picture"
                RefPicLtPocList[ i ][ j ] = FullPocLt[ i ][ k ]
        }
        k++
    }
} else {
    layerIdx =
DirectDependentLayerIdx[ GeneralLayerIdx[ nuh_layer_id ] ][ ilrp_idc[ i ][ RplIdx[ i ] ][ j ]
]
    refPicLayerId = vps_layer_id[ layerIdx ]
    if( there is a reference picture picA in the DPB with nuh_layer_id equal to
refPicLayerId and
        the same PicOrderCntVal as the current picture )
        RefPicList[ i ][ j ] = picA
    else
        RefPicList[ i ][ j ] = "no reference picture"
    }
[0149] }
}

```

[0150] 上述发信号通知自适应分辨率参数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10示出了计算机系统(1000),其适于实现所公开主题的某些实施例。

[0151] 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,所述指令可由计算机中央处理单元(Central Processing Unit,CPU)、图形处理单元(Graphics Processing Units,GPU)等直接执行或通过译码、微代码等方式执行。

[0152] 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。

[0153] 图10所示的用于计算机系统(1000)的组件本质上是示例性的,并不用于对实现本申请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(1000)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 or 要求。

[0154] 计算机系统(1000)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)。

[0155] 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(1001)、鼠标(1002)、触控板(1003)、触摸屏(1010)、操纵杆(1005)、麦克风(1006)、扫描仪(1007)、照相机(1008)。

[0156] 计算机系统(1000)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受。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(1010)、或操纵杆(1005)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(1009)、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(1010),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)。

[0157] 计算机系统(1000)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/DVD ROM/RW)(1020)或类似介质(1021)的光学介质、拇指驱动器(1022)、可移动硬盘驱动器或固体状态驱动器(1023),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。

[0158] 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。

[0159] 计算机系统(1000)还可以包括通往一个或多个通信网络的接口。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(1049)(例如,计算机系统(1000)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(1000)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(1000)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。

[0160] 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(1000)的核心(1040)。

[0161] 核心(1040)可包括一个或多个中央处理单元(CPU)(1041)、图形处理单元(GPU)(1042)、以现场可编程门阵列(FPGA)(1043)形式的专用可编程处理单元、用于特定任务的硬件加速器(1044)等。这些设备以及只读存储器(ROM)(1045)、随机存取存储器(1046)、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)(1047)等可通过系统总线(1048)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(1048),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(1048),或通过外围总线(1049)进行连接。外围总线的体系结构包括外部控制器接口PCI、通用串行总线USB等。

[0162] CPU(1041)、GPU(1042)、FPGA(1043)和加速器(1044)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(1045)或RAM(1046)中。过渡数据也可以存储在RAM(1046)中,而永久数据可以存储在例如内部大容量存储器(1047)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(1041)、GPU(1042)、大容量存储器(1047)、ROM(1045)、RAM(1046)等紧密关联。

[0163] 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申请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。

[0164] 作为实施例而非限制,具有体系结构(1000)的计算机系统,特别是核心(1040),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(1040)的特定存储器,例如核心内部大容量存储器(1047)或ROM(1045)。实现本申请的各种实施例的软件可以存储在这种设备中并且由核心(1040)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(1040)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(1046)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(1044))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。本申请包括任何合适的硬件和软件组合。

[0165] 虽然本申请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申请的范围内。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,但其体现了本申请的原则,因此属于本申请的精神和范围之内。

100

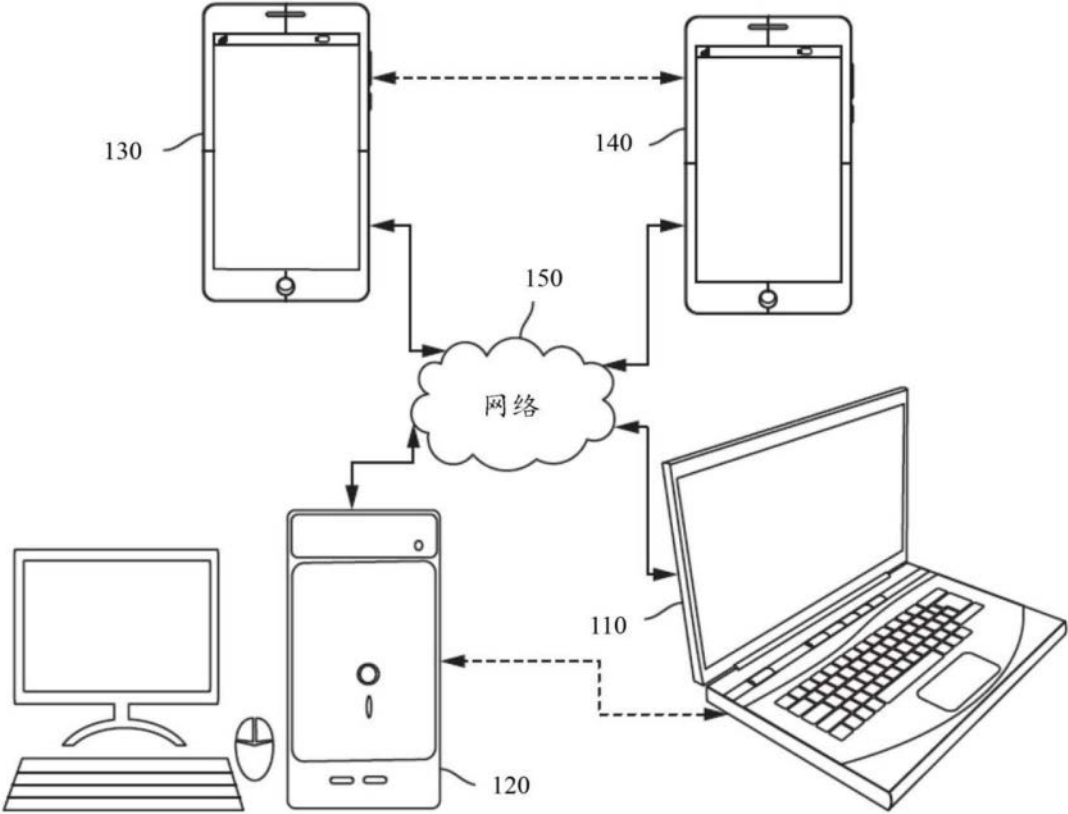


图1

200

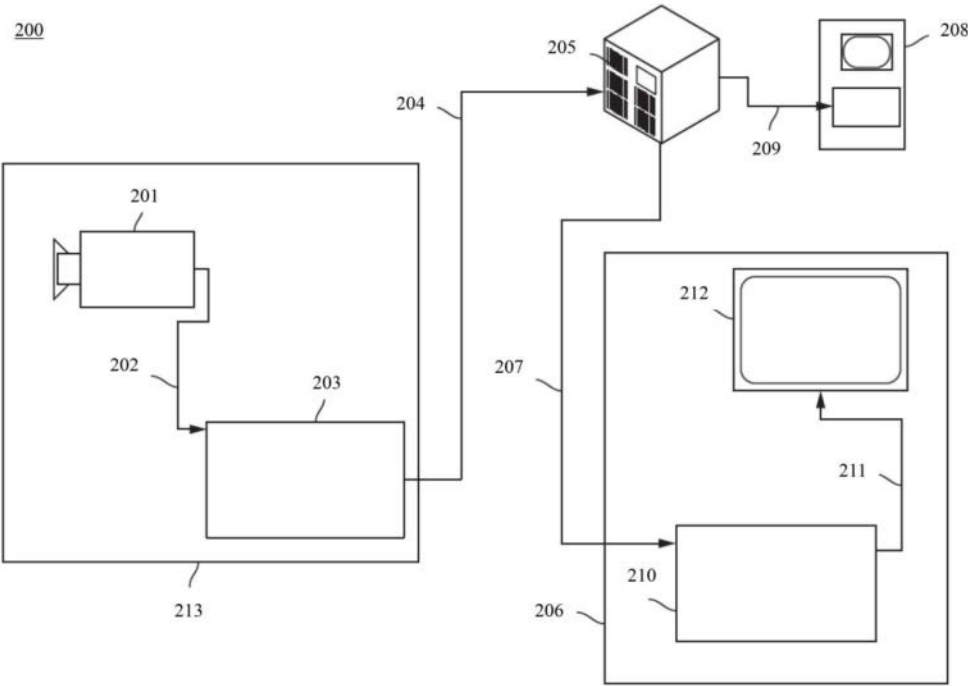


图2

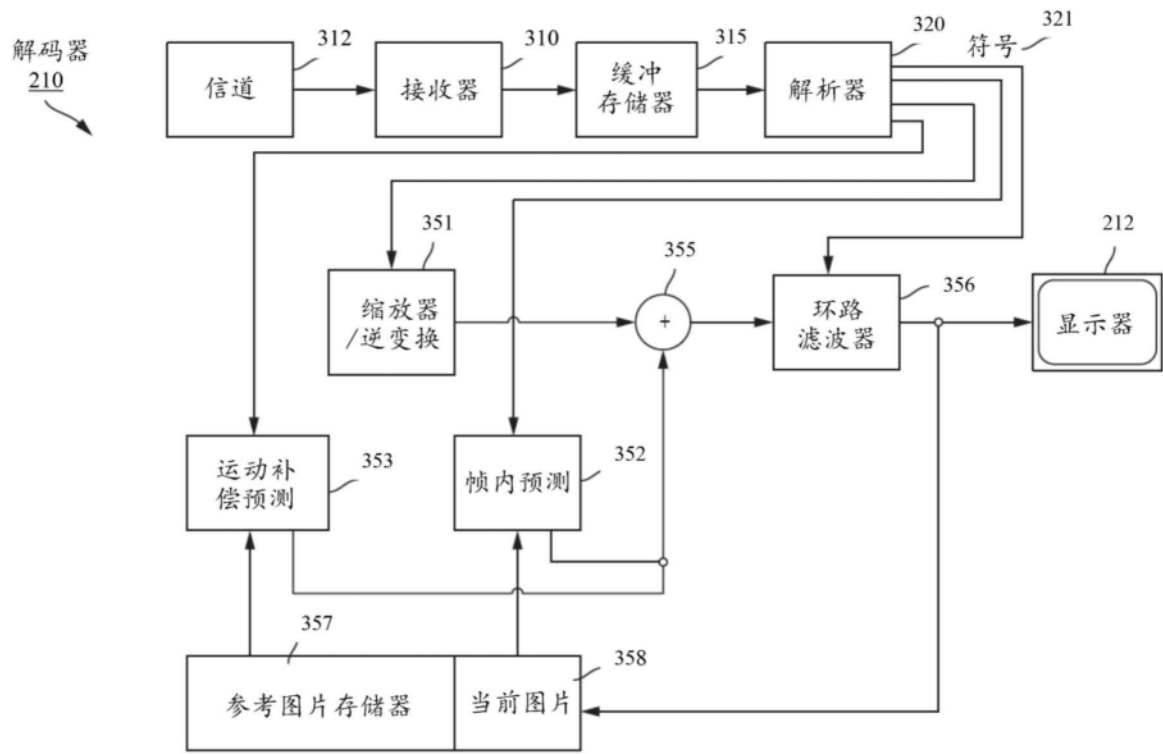


图3

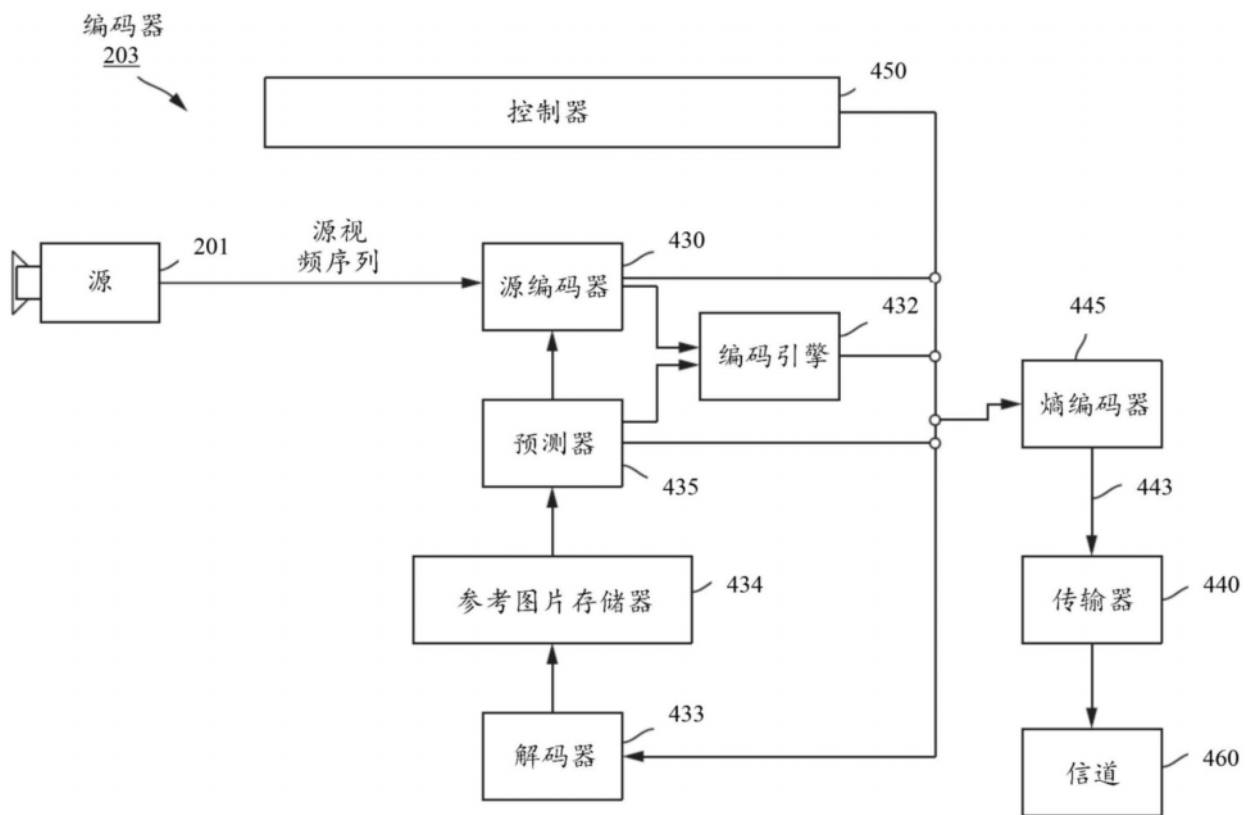


图4

500A

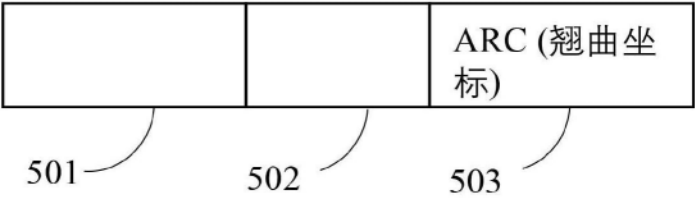


图5A (相关技术)

500B

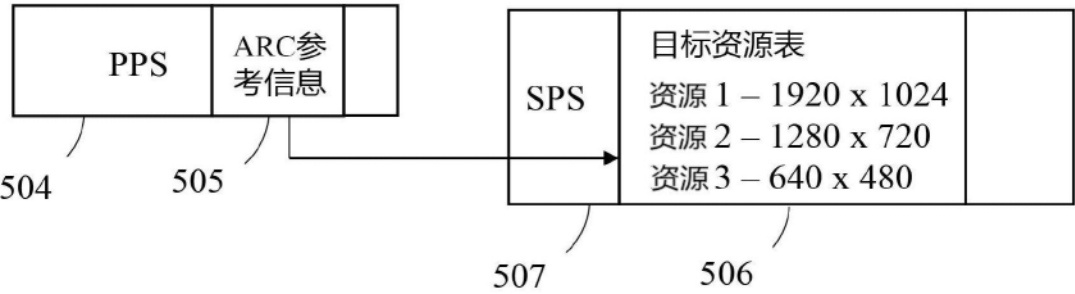


图5B (相关技术)

500C



图5C

500D

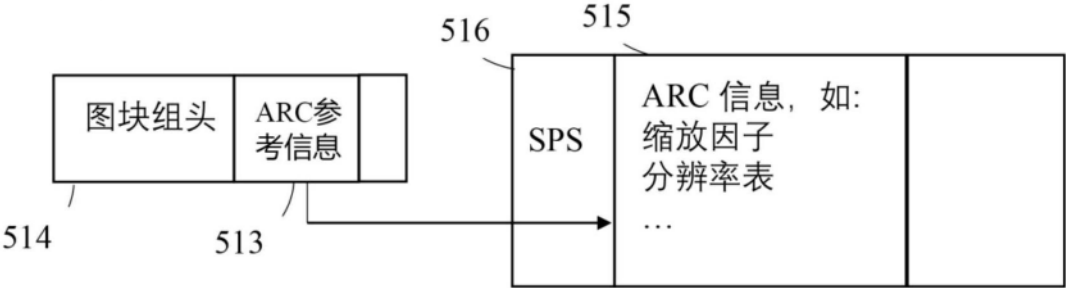


图5D

500E

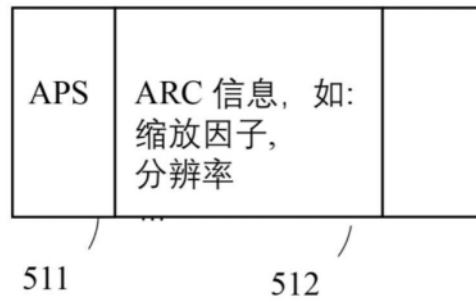


图5E

600

	tile_group_header(){	
	...	
	if(adaptive_pic_resolution_change_flag){	
	dec_pic_size_idx	u(1)
	}	
	...	
	}	

	seq_parameter_set_rbsp() {	描述符
	...	
	adaptive_pic_resolution_change_flag	u(1)
	if(adaptive_pic_resolution_change_flag) {	
613	output_pic_width_in_luma_samples	ue(v)
	output_pic_height_in_luma_samples	ue(v)
614	reference_pic_size_present_flag	u(1)
	if(reference_pic_size_present_flag)	
	{	
615	reference_pic_width_in_luma_samples	ue(v)
	reference_pic_height_in_luma_samples	ue(v)
	}	
	num_dec_pic_size_in_luma_samples_minus1	ue(v)
	for(i = 0; i <= num_dec_pic_size_in_luma_samples_minus1; i++){	
	dec_pic_width_in_luma_samples[i]	ue(v)
	dec_pic_height_in_luma_samples[i]	ue(v)
	}	
	}	
	...	
	}	

图6

700

701	video_parameter_set_rbsp() {	描述符
	vps_video_parameter_set_id	u(4)
703	vps_max_layers_minus1	u(6)
	if (vps_max_layers_minus1 > 0)	
704	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) {	
706	vps_independent_layer_flag [i]	u(1)
	if (!vps_independent_layer_flag [i])	
	for (i = 0; j < i; j++)	
707	vps_direct_dependency_flag [i][j]	u(1)
	}	
	}	
708	if (vps_max_layers_minus1 > 0 && !vps_all_independent_layers_flag) {	
	vps_output_layers_mode	u(2)
	if (vps_output_layers_mode == 2)	
	for (i = 0; i < vps_max_layers_minus1; i++)	
709	vps_output_layer_flag [i]	u(1)
	}	
	vps_constraint_info_present_flag	u(1)
	vps_reserved_zero_7bits	u(7)
	if (vps_constraint_info_present_flag)	
	general_constraint_info()	
	vps_extension_flag	u(1)
	if (vps_extension_flag)	
	while (more_rbsp_data ())	
	vps_extension_data_flag	u(1)
	rbsp_trailing_bits()	
	}	

图7

800

seq_parameter_set_rbsp() {	描述符
...	
inter_layer_ref_pics_present_flag	u(1)
...	
}	

图8

900

ref pic list struct (listIdx, rplsIdx) {	描述符
num_ref_entries [listIdx][rplsIdx]	ue(v)
if (long term ref pics flag)	
ltrp_in_slice_header_flag [listIdx][rplsIdx]	u(1)
for (i = 0, j = 0; i < num_ref_entries [listIdx][rplsIdx]; i++	
if (inter layer ref pics present flag)	
inter_layer_ref_pic_flag [listIdx][rplsIdx][i]	
if (!inter_layer_ref_pic_flag [listIdx][rplsIdx][i]) {	
if (long term ref pics flag)	
st_ref_pic_flag [listIdx][rplsIdx][i]	u(1)
if (st_ref_pic_flag [listIdx][rplsIdx][i]) {	
abs_delta_poc_st [listIdx][rplsIdx][i]	ue(v)
if (AbsDeltaPocSt [listIdx][rplsIdx][i] > 0)	
strp_entry_sign_flag [listIdx][rplsIdx][i]	u(i)
} else if (!ltrp_in_slice_header_flag [listIdx][rplsIdx])	
rpls_poc_lsb_lt [listIdx][rplsIdx][j++]	u(v)
} else	
ilrp_idc [listIdx][rplsIdx][i]	ue(v)
}	
}	

图9

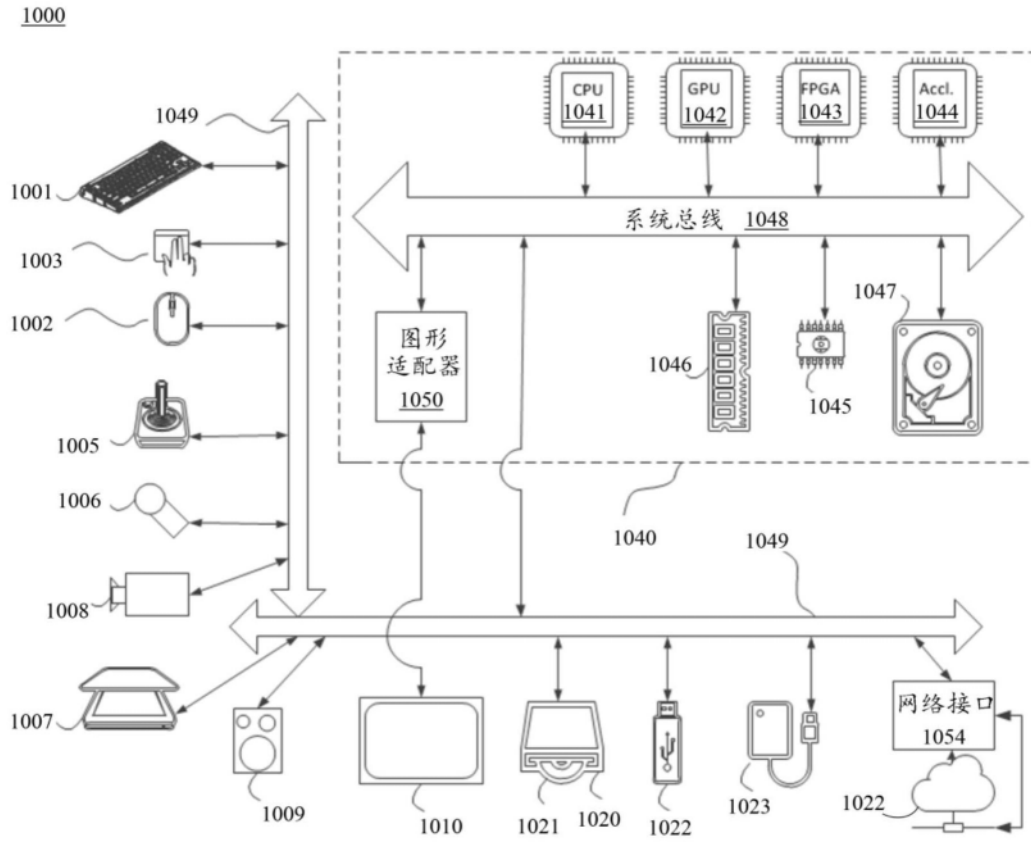


图10