(19)国家知识产权局

(12)发明专利

(10)授权公告号 CN 113924770 B
(45)授权公告日 2024.08.16

(21)申请号 202180003565.3
(22)申请日 2021.02.22
(65)同一申请的已公布的文献号
申请公布号 CN 113924770 A
(43)申请公布日 2022.01.11
(30)优先权数据
63/003,123 2020.03.31 US
17/081,392 2020.10.27 US
(85)PCT国际申请进入国家阶段日
2021.11.26
(86)PCT国际申请的申请数据
PCT/US2021/019020 2021.02.22
(87)PCT国际申请的公布数据
WO2021/202012 EN 2021.10.07

(73)专利权人 腾讯美国有限责任公司
地址 美国加利福尼亚州帕洛阿尔托公园大道2747号
(72)发明人 崔秉斗 刘杉 史蒂芬·文格尔
(74)专利代理机构 北京德琦知识产权代理有限公司 11018
专利代理师 陈美娥 王琦

(51)Int.Cl.
H04N 19/00(2006.01)

(56)对比文件
Benjamin Bross等.Versatile Video Coding (Draft 8).《Joint Video Experts Team (JVET) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11》.2020,7.3节和7.5节.

审查员 李若云

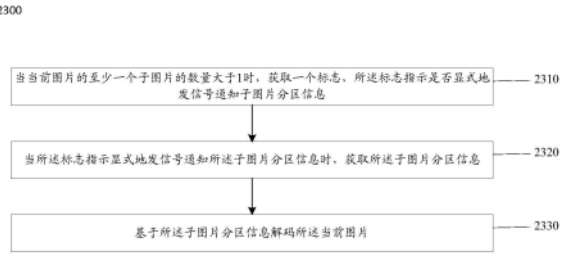
权利要求书3页 说明书30页 附图19页

(54)发明名称

视频编解码的方法和设备

(57)摘要

本公开实施例提供了一种对已编码视频比特流进行解码的方法和设备。该方法包括:当当前图片的至少一个子图片的数量大于1时,获取一个标志,所述标志指示是否显式地发信号通知子图片分区信息;当所述标志指示显式地发信号通知所述子图片分区信息时,获取所述子图片分区信息;以及基于所述子图片分区信息解码所述当前图片。



CN 113924770 B

1.一种对已编码视频比特流进行解码的方法,其特征在于,所述方法包括:

确定当前图片的至少一个子图片的数量,所述确定包括:

从编码比特流中获取所述当前图片的图片宽度pic_width_max_in_luma_samples和图片高度pic_height_max_in_luma_sample;以及

如果所述图片宽度不大于所述当前图片的亮度分量的编码树单元的大小CtbSizeY,并且所述图片高度不大于所述亮度分量的编码树单元的大小,则确定所述至少一个子图片的数量等于1;

当所述当前图片的至少一个子图片的数量大于1时,确定一个标志subpic_info_present_flag,所述标志指示是否显式地发信号通知子图片分区信息;

当所述标志指示显式地发信号通知所述子图片分区信息时,获取所述子图片分区信息;以及

基于所述子图片分区信息解码所述当前图片;

所述方法进一步包括:

当所述当前图片不是编码层视频序列CLVS的第一个图片时,对于所述当前图片的每个子图片,如果所述当前图片的子图片的SubpicIdVal[i]的值不等于同一层中按解码顺序的前一个图片中的子图片的SubpicIdVal[i]的值,则确定所述当前图片的子图片对应的序列参数集SPS独立子图片标志sps_independent_subpics_flag等于1,或者,子图片视为图片标志subpic_treated_as_pic_flag[i]和跨子图片环路滤波器启用标志loop_filter_across_subpic_enabled_flag[i]等于1。

2.根据权利要求1所述的方法,其特征在于,所述标志和所述子图片分区信息在序列参数集SPS中发信号通知。

3.根据权利要求2所述的方法,其特征在于,所述至少一个子图片的数量在所述SPS中发信号通知。

4.根据权利要求1所述的方法,其特征在于,当所述至少一个子图片的数量等于1时,所述标志指示不显式地发信号通知所述子图片分区信息。

5.根据权利要求1-4任一项所述的方法,其特征在于,所述子图片分区信息指示所述至少一个子图片中的子图片的大小。

6.根据权利要求5所述的方法,其特征在于,所述子图片的大小包括所述子图片的高度和宽度。

7.根据权利要求5所述的方法,其特征在于,所述子图片的大小在所述子图片分区信息中以所述当前图片的亮度分量的编码树单元的大小为单位来指示。

8.根据权利要求5所述的方法,其特征在于,基于所述子图片分区信息指示的所述子图片的大小,确定另一子图片的大小。

9.一种产生视频比特流的编码方法,其特征在于,所述方法包括:

确定当前图片的图片宽度pic_width_max_in_luma_samples和图片高度pic_height_max_in_luma_sample,如果所述图片宽度不大于所述当前图片的亮度分量的编码树单元的大小CtbSizeY,并且所述图片高度不大于所述亮度分量的编码树单元的大小,则确定至少一个子图片的数量等于1;

当所述当前图片的至少一个子图片的数量大于1时,确定一个标志subpic_info_

present_flag,所述标志指示是否显式地发信号通知子图片分区信息;

当所述标志指示显式地发信号通知所述子图片分区信息时,获取所述子图片分区信息;以及

基于所述子图片分区信息编码所述当前图片,以产生所述视频比特流;

所述方法进一步包括:

当所述当前图片不是编码层视频序列CLVS的第一个图片时,对于所述当前图片的每个子图片,如果所述当前图片的子图片的SubpicIdVal[i]的值不等于同一层中按解码顺序的前一个图片中的子图片的SubpicIdVal[i]的值,则设置所述当前图片的子图片对应的序列参数集SPS独立子图片标志sps_independent_subpics_flag等于1,或者,子图片视为图片标志subpic_treated_as_pic_flag[i]和跨子图片环路滤波器启用标志loop_filter_across_subpic_enabled_flag[i]等于1。

10.一种对已编码视频比特流进行解码的设备,其特征在于,所述设备包括:

至少一个存储器,被配置为存储程序代码;以及

至少一个处理器,被配置为读取所述程序代码,并按照所述程序代码的指示进行操作,以执行权利要求1-8任一项所述的方法。

11.一种产生视频比特流的编码设备,其特征在于,所述设备包括:

至少一个存储器,被配置为存储程序代码;以及

至少一个处理器,被配置为读取所述程序代码,并按照所述程序代码的指示进行操作,以执行权利要求9所述的编码方法。

12.一种对已编码视频比特流进行解码的装置,其特征在于,所述装置包括:

确定模块,被配置为确定当前图片的至少一个子图片的数量,所述确定包括:

从编码比特流中获取所述当前图片的图片宽度pic_width_max_in_luma_samples和图片高度pic_height_max_in_luma_sample;以及

如果所述图片宽度不大于所述当前图片的亮度分量的编码树单元的大小CtbSizeY,并且所述图片高度不大于所述亮度分量的编码树单元的大小,则确定所述至少一个子图片的数量等于1;

第一获取模块,被配置为在所述当前图片的至少一个子图片的数量大于1时,确定一个标志subpic_info_present_flag,所述标志指示是否显式地发信号通知子图片分区信息;

第二获取模块,被配置为在所述标志指示显式地发信号通知所述子图片分区信息时,获取所述子图片分区信息;以及

解码模块,被配置为基于所述子图片分区信息解码所述当前图片;

所述装置进一步包括:

确定模块,被配置为当所述当前图片不是编码层视频序列CLVS的第一个图片时,对于所述当前图片的每个子图片,如果所述当前图片的子图片的SubpicIdVal[i]的值不等于同一层中按解码顺序的前一个图片中的子图片的SubpicIdVal[i]的值,则确定所述当前图片的子图片对应的序列参数集SPS独立子图片标志sps_independent_subpics_flag等于1,或者,子图片视为图片标志subpic_treated_as_pic_flag[i]和跨子图片环路滤波器启用标志loop_filter_across_subpic_enabled_flag[i]等于1。

13.一种非易失性计算机可读介质,其特征在于,用于存储指令,所述指令包括一个或

多个指令,当所述一个或多个指令由一个或多个处理器执行时,使所述一个或多个处理器执行权利要求1-9任一项所述的方法。

视频编解码的方法和设备

[0001] 相关申请的交叉引用

[0002] 本申请要求于2020年3月31日提交的、申请号为63/003,123的美国临时专利申请、以及于2020年10月27日提交的、申请号为17/081,392的美国专利申请的优先权,其全部内容并入本文。

技术领域

[0003] 所公开的主题涉及视频编码和解码,更具体地,涉及在已编码视频比特流中发信号通知子图片划分。

背景技术

[0004] 使用具有运动补偿的图片间预测来进行视频编码和解码已经众所周知。未压缩的数字视频可以包括一系列图片,每个图片具有例如1920×1080亮度样本及相关色度样本的空间维度。所述系列图片可以具有固定的或可变的图片速率(也非正式地称为帧率),例如每秒60个图片或60Hz。未压缩的视频具有很高的比特率要求。例如,每个样本8比特的1080p60 4:2:0视频(60Hz帧率下1920x1080亮度样本分辨率)要求接近1.5Gbit/s带宽。一小时这样的视频就需要超过600GB的存储空间。

[0005] 视频编码和解码的一个目的是通过压缩减少输入视频信号的冗余。压缩可以帮助降低对上述带宽或存储空间的要求,在某些情况下可降低两个或更多数量级。无损压缩和有损压缩、以及两者的组合均可采用。无损压缩是指从压缩的原始信号中重建原始信号精确副本的技术。当使用有损压缩时,重建信号可能与原始信号不完全相同,但是原始信号和重建信号之间的失真足够小,使得重建信号可用于预期应用。有损压缩广泛应用于视频。容许的失真量取决于应用。例如,相比于电视应用的用户,某些消费流媒体应用的用户可以容忍更高的失真。可实现的压缩比反映出:较高的允许/容许失真可产生较高的压缩比。

[0006] 视频编码器和解码器可以利用几大类技术,例如包括运动补偿、变换、量化和熵编码,其中的一些技术将会在下文中介绍。

[0007] 历史上,视频编码器和解码器倾向于在给定图片大小上进行操作,在大多数情况下,该给定图片大小是针对已编码视频序列(CVS)、图片群组(GOP)或类似的多图片时间帧定义的,并保持不变。例如,在MPEG-2中,已知系统设计根据诸如场景的活动等因素改变水平分辨率(从而改变图片大小),但是仅在I图片上改变水平分辨率,因此通常用于GOP。例如,根据ITU-T Rec.H.263附录P,在CVS内使用不同分辨率对参考图片进行重采样是已知的。然而,这里的图片大小没有改变,只有参考图片被重采样,导致可能只有部分图片画布被使用(在下采样的情况下),或者只有部分场景被捕获(在上采样的情况下)。此外,H.263附录Q允许以因子2(在每个维度中)对单个宏块向上或向下重采样。同样,图片大小保持不变。宏块的大小在H.263中是固定的,因此不需要发信号通知。

[0008] 在现代视频编解码中,已预测图片中的图片大小的变化越来越成为主流。例如,VP9允许参考图片重采样并改变整个图片的分辨率。类似地,针对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)”,联合视频组文档JVET-M0135-v1,2019年1月9日至19日,其整体并入本文),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,建议在序列参数集中对不同的候选分辨率进行编码,并由图片参数集中的每图片语法元素引用。

发明内容

[0009] 在实施例中,提供了一种使用至少一个处理器对已编码视频比特流进行解码的方法,包括:当当前图片的至少一个子图片的数量大于1时,获取一个标志,所述标志指示是否显式地发信号通知子图片分区信息;当所述标志指示显式地发信号通知所述子图片分区信息时,获取所述子图片分区信息;以及基于所述子图片分区信息解码所述当前图片。

[0010] 在实施例中,提供了一种对已编码视频比特流进行解码的设备,包括:至少一个存储器,被配置为存储程序代码;以及至少一个处理器,被配置为读取所述程序代码,并按照所述程序代码的指令进行操作,所述程序代码包括:第一获取代码,被配置为使所述至少一个处理器在当前图片的至少一个子图片的数量大于1时,获取一个标志,所述标志指示是否显式地发信号通知子图片分区信息;第二获取代码,被配置为使所述至少一个处理器在所述标志指示显式地发信号通知所述子图片分区信息时,获取所述子图片分区信息;以及解码代码,被配置为使所述至少一个处理器基于所述子图片分区信息解码所述当前图片。

[0011] 在实施例中,提供了一种非易失性计算机可读介质,用于存储指令,所述指令包括一个或多个指令,当所述一个或多个指令由用于对已编码视频比特流进行解码的设备的一个或多个处理器执行时,使所述一个或多个处理器:当当前图片的至少一个子图片的数量大于1时,获取一个标志,所述标志指示是否显式地发信号通知子图片分区信息;当所述标志指示显式地发信号通知所述子图片分区信息时,获取所述子图片分区信息;以及基于所述子图片分区信息解码所述当前图片。

附图说明

[0012] 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

[0013] 图1是根据实施例的通信系统的简化框图的示意图。

[0014] 图2是根据实施例的通信系统的简化框图的示意图。

[0015] 图3是根据实施例的解码器的简化框图的示意图。

[0016] 图4是根据实施例的编码器的简化框图的示意图。

[0017] 图5A-5E是根据实施例的用于发信号通知ARC参数的选项的示意图。

[0018] 图6A-6B是根据实施例的语法表的示例的示意图。

[0019] 图7是根据实施例的具有自适应分辨率变化的可缩放性的预测结构的示例。

[0020] 图8是根据实施例的语法表的示例。

[0021] 图9是根据实施例的解析和解码每个访问单元的POC周期和访问单元计数值的简化框图的示意图。

[0022] 图10是根据实施例的包括多层子图片的视频比特流结构的示意图。

[0023] 图11是根据实施例的具有增强分辨率的所选子图片的显示的示意图。

[0024] 图12是根据实施例的包括多层子图片的视频比特流的解码和显示过程的框图。

[0025] 图13是根据实施例的具有子图片的增强层的360视频显示的示意图。

[0026] 图14是根据实施例的子图片及其对应层的布局信息和图片预测结构的示例。

[0027] 图15是根据实施例的具有局部区域的空间可缩放性模态的子图片及其对应层的布局信息和图片预测结构的示例。

[0028] 图16A-16B是根据实施例的用于子图片布局信息的语法表的示例。

[0029] 图17是根据实施例的用于子图片布局信息的SEI消息的语法表的示例。

[0030] 图18是根据实施例的语法表的示例,用于指示每个输出层集的输出层和配置文件/层/级别(profile/tier/level)信息。

[0031] 图19是根据实施例的语法表的示例,用于指示每个输出层集的输出层模式开启。

[0032] 图20是根据实施例的语法表的示例,用于指示每个输出层集的每个层的当前子图片。

[0033] 图21是根据实施例的语法表的示例,用于指示子图片标识符。

[0034] 图22是根据实施例的语法表的示例,用于指示子图片分区信息。

[0035] 图23是根据实施例的对已编码视频比特流进行解码的示例过程的流程图。

[0036] 图24是根据实施例的计算机系统的示意图。

具体实施方式

[0037] 图1示出了根据本公开实施例的通信系统(100)的简化框图。通信系统(100)可以包括通过网络(150)互联的至少两个终端装置(110,120)。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。

[0038] 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。

[0039] 在图1中,终端装置(110-140)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(110-140)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。

[0040] 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。

[0041] 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。

[0042] 图3是根据本公开实施例的视频解码器(210)的功能框图。

[0043] 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。

[0044] 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和/或熵解码。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、子图片、图块(tile)、切片(slice)、砖块(brick)、宏块、编码树单元(Coding Tree Unit,CTU)、编码单元(Coding Unit,CU)、块、变换单元(Transform Unit,TU)、预测单元(Prediction Unit,PU)等等。图块可以指示图片中的特定图块列和行内的CU/CTU的矩形区域。砖块可以指示特定图块内的CU/CTU行的矩形区域。切片可以指示图片的一个或多个砖块,其包含在NAL单元中。子图片可以指示图片中的一个或

多个切片的矩形区域。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。

[0045] 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和/或解析操作,从而创建符号(321)。

[0046] 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。

[0047] 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。

[0048] 第一单元是缩放器和/或逆变换单元(351)。缩放器和/或逆变换单元(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。

[0049] 在一些情况下,缩放器和/或逆变换单元(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片的预测性信息,但可使用来自当前图片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和/或逆变换单元(351)提供的输出样本信息中。

[0050] 在其它情况下,缩放器和/或逆变换单元(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)添加到缩放器和/或逆变换单元的输出(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,从参考图片存储器提取的样本值的内插、运动矢量预测机制等等。

[0051] 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。

[0052] 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,以用于后续的帧间图片预测。

[0053] 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。

[0054] 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。

[0055] 在实施例中,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。

[0056] 图4是根据本公开实施例的视频编码器(203)的功能框图。

[0057] 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。

[0058] 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.601 Y 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。

[0059] 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)。

[0060] 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。

[0061] “本地”解码器(433)的操作可与已在上文结合图3详细描述的“远程”解码器(210)相同。然而，另外简要参考图3，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和/或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。

[0062] 此时可以观察到，除存在于解码器中的解析和/或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的解码器技术互逆。仅在某些区域中需要更详细的描述，并且在下文提供。

[0063] 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。

[0064] 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)。

[0065] 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。

[0066] 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。

[0067] 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。

[0068] 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)。

[0069] 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。

[0070] 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。

[0071] 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。

[0072] 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。

[0073] 源图片通常可在空间上细分成多个样本块(例如,4×4、8×8、4×8或16×16个样本的块),且逐块进行编码。这些块可参考其它(已编码)块进行预测编码,根据应用于块的相应图片的编码分配来确定所述其它块。举例来说,I图片的块可进行非预测编码,或所述块可参考同一图片的已经编码的块来进行预测编码(空间预测或帧内预测)。P图片的像素块可参考一个先前编码的参考图片通过空间预测或通过时域预测进行预测编码。B图片的块可参考一个或两个先前编码的参考图片通过空间预测或通过时域预测进行预测编码。

[0074] 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。

[0075] 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。

[0076] 最近,将多个语义独立的图片部分进行压缩域聚合或提取到单个视频图片中已经引起了一些关注。具体地,在例如360编码或某些监视应用的上下文中,多个语义独立的源图片(例如立方体投影的360场景的六个立方体表面、或在多摄像头监视设置情况下的各个摄像头输入)可能需要单独的自适应分辨率设置,以应对给定时间点的不同场景活动。换句话说,编码器在给定时间点可以对构成整个360场景或监视场景的不同语义独立的图片选择使用不同的重采样因子。当组合成单个图片时,这又要求对已编码图片的部分执行参考图片重采样,并且自适应分辨率编码信令可用。

[0077] 下面将介绍一些术语,这些术语将在本说明书的其余部分中提及。

[0078] 在某些情况下,子图片可以指样本、块、宏块、编码单元或类似实体的矩形排列,这些实体按语义分组,并且可以以变化的分辨率独立编码。一个或多个子图片可以形成一个图片。一个或多个已编码子图片可以形成一个已编码图片。可以将一个或多个子图片组合成图片,并且可以从图片中提取一个或多个子图片。在某些环境中,一个或多个已编码子图片可以在压缩域中组合,而不需要转码到样本级别成为已编码图片,并且在相同或其它情况下,可以从压缩域中的已编码图片中提取一个或多个已编码子图片。

[0079] 自适应分辨率改变(ARC)可以指允许通过例如参考图片重采样来改变已编码视频序列内的图片或子图片的分辨率的机制。此后,ARC参数是指执行自适应分辨率改变所需的控制信息,该控制信息可以包括,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。

[0080] 在实施例中,可对单个语义独立的已编码视频图片执行编码和解码。在描述具有独立ARC参数的多个子图片的编码/解码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ARC参数的选项。

[0081] 参考图5A-5E,示出了发信号通知ARC参数的几个实施例。如每个实施例所述,从编解码效率、复杂度和架构的角度来看,它们可能具有某些优点和缺点。视频编解码标准或技术可以选择这些实施例中的一个或多个,或相关技术中已知的选项,以发信号通知ARC参数。这些实施例可以不相互排斥,并且可以基于应用需要、所涉及的标准技术或编码器的选择进行互换。

[0082] ARC参数的类别可以包括:

[0083] -在X维度和Y维度上分开或组合的上采样和/或下采样因子

[0084] -加上时间维度的上采样和/或下采样因子,表示对给定数量的图片进行恒速放大/缩小

[0085] -上述两种类别中的任何一种都可能涉及一个或多个可能较短的语法元素的编码,该语法元素可以指向包含一个或多个因子的表。

[0086] -以样本、块、宏块、编码单元(CU)或任何其它合适的粒度为单位,输入图片、输出图片、参考图片、已编码图片在X或Y维度上的组合或单独的分辨率。如果存在多于一个的分辨率(例如,一个用于输入图片,一个用于参考图片),则在某些情况下,一组值可以从另一组值推断出来。这可以例如使用标志来选通。更详细的示例请参见下文。

[0087] -“翘曲”坐标类似于H.263附录P中使用的那些坐标,同样采用上述合适的粒度。H.263附录P定义了一种对这种翘曲坐标(warping coordinate)进行编码的有效方式,但也可以设想其它可能更有效的方式。例如,附录P的翘曲坐标的可变长度可逆的“霍夫曼(Huffman)”式编码可以由适当长度的二进制编码代替,其中二进制码字的长度例如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,以允许在最大图片大小的边界之外“翘曲”。

[0088] -上采样和/或下采样滤波器参数。在实施例中,可能只有单个滤波器用于上采样和/或下采样。然而,在实施例中,可能希望在滤波器设计中允许有更大的灵活性,并且这可能需要滤波器参数的信令。这些参数可以通过可能的滤波器设计列表中的索引来选择,滤波器可以(例如,通过滤波器系数的列表、使用适当的熵编码技术)完全指定,滤波器可以通过上采样和/或下采样率隐式地选择,该上采样/下采样率又根据上述机制中的任何一种发信号通知,等等。

[0089] 此后,本说明书假设对一组有限的上采样和/或下采样因子进行编码(在X维度和Y维度上使用相同的因子),通过码字表示。可以例如使用对诸如H.264和H.265的视频编解码规范中的某些语法元素通用的指数哥伦布(Ext-Golomb)代码,对该码字进行可变长度编码。例如,一种合适的从值到上采样和/或下采样因子的映射可以根据表1:

[0090] 表1

[0091]

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

[0092] 可以根据应用的需要以及视频压缩技术或标准中可用的放大和缩小机制的能力来设计许多类似的映射。该表可扩展到更多的值。值也可以由除Ext-Golomb代码之外的熵编码机制来表示,例如使用二进制编码来表示。当例如MANE对视频处理引擎(最重要的是编码器和解码器)本身之外的重采样因子感兴趣时,这可能具有某些优势。应当注意,对于不需要改变分辨率的情况,可以选择较短的Ext-Golomb代码;在上表中,只有一个比特。在最常见的情况下,这比使用二进制码具有编解码效率的优势。

[0093] 该表中的条目的数量以及它们的语义可以完全或部分可配置。例如,该表的基本轮廓(outline)可以在诸如序列或解码器参数集之类的“高”参数集中传送。在实施例中,一个或多个这样的表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或序列参数集来选择。

[0094] 下面描述如何在视频编解码技术或标准语法中包括如上所述编码的上采样和/或下采样因子(ARC信息)。类似的考虑可以应用于一个或几个控制上采样和/或下采样滤波器的码字。参见以下关于何时滤波器或其它数据结构需要相对大量的数据的讨论。

[0095] 如图5A所示,H.263附录P以四个翘曲坐标形式将ARC信息(502)包括在图片头(501)中,具体包括在H.263 PLUSPTYPE(503)头扩展中。当a)有可用的图片头,并且b)预计ARC信息频繁改变时,这可能是一个明智的设计选择。然而,当使用H.263样式的信令时,开销可能相当高,并且因为图片头可能具有瞬态性质,缩放因子可能不适用于图片边界。

[0096] 如图5B所示,JVCET-M135-v1包括位于图片参数集(504)中的ARC参考信息(505)(索引),该ARC参考信息(505)对包括目标分辨率的表(506)进行索引,该表(506)位于序列参数集(507)内。根据作者的口头陈述,通过在能力交换期间使用SPS作为互操作性协商点(interoperability negotiation point),可以证明在序列参数集(507)的表(506)中,可能的分辨率的放置。通过参考适当的图片参数集(504),分辨率可以在表(506)中的值所设置的限制内随图片的变化而变化。

[0097] 参考图5C-5E,可以存在以下实施例,以在视频比特流中传送ARC信息。这些选项中的每一个都具有优于上述实施例的某些优点。实施例可以同时存在于同一视频编解码技术或标准中。

[0098] 在实施例中,例如图5C所示的实施例,诸如重采样(缩放)因子之类的ARC信息(509)可以存在于切片头、GOP头、图块头或图块组头。图5C示出了使用图块组头(508)的实施例。例如如上所述的,如果ARC信息很小,例如单个可变长度ue(v)或几个比特的固定长度码字,这就足够了。将ARC信息直接放在图块组头中具有额外的优点,即ARC信息可以应用于例如由该图块组表示的子图片,而不是整个图片,另请参见下文。另外,即使视频压缩技术或标准仅设想了整个图片的自适应分辨率的改变(例如与基于图块组的自适应分辨率的改变相比),从差错修复的角度来看,将ARC信息放入图块组头与将其放入H.263格式的图片头

相比具有一定的优势。

[0099] 在实施例中,例如图5D所示的实施例,ARC信息(512)本身可以存在于适当的参数集中,例如图片参数集、头参数集、图块参数集、自适应参数集等。图5D示出了使用自适应参数集(511)的实施例。更有利地,该参数集的范围可以不大于图片,例如图块组。通过激活相关参数集,ARC信息的使用是隐式的。例如,当视频编解码技术或标准仅考虑基于图片的ARC时,则图片参数集或等效参数可能适用。

[0100] 在实施例中,例如图5E所示的实施例,ARC参考信息(513)可以存在于图块组头(514)或类似的数据结构中。该参考信息(513)可以指参数集(516)中可用的ARC信息(515)的子集,该参数集的范围超出单个图片,例如序列参数集或解码器参数集。

[0101] 如图6A所示,作为可应用于图片的一部分(可能是矩形的)的头的示例性语法结构,图块组头(601)可以有条件地包含可变长度的、Exp-Golomb编码的语法元素dec_pic_size_idx(602)(以粗体字表示)。图块组头中该语法元素的存在可以基于自适应分辨率(603)的使用来选通—在这里,标志的值未以加粗字体表示,这意味着该标志在它在语法图中出现的点处存在于比特流中。是否对该图片或其部分使用自适应分辨率可以在比特流内部或外部的任何高级语法结构中发信号通知。在所示的示例中,它在序列参数集中发信号通知,如下所述。

[0102] 参考图6B,还示出了序列参数集(610)的摘录。所示的第一语法元素是adaptive_pic_resolution_change_flag(611)。当为真时,该标志可以指示使用自适应分辨率,这反过来可能需要某些控制信息。在该示例中,这种控制信息基于该标志的值有条件地出现,该标志的值基于参数集(612)和图块组头(601)中的if()语句。

[0103] 当使用自适应分辨率时,在该示例中,被编码的是以样本为单位的输出分辨率(613)。数字613是指output_pic_width_in_luma_samples和output_pic_height_in_luma_samples,它们可以共同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,该数量可以是这两个语法元素的值的乘积。而且,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)可能会限制编号范围(例如,一个或两个维度必须能够被2的幂整除)或宽高比(例如,宽度和高度必须是例如4:3或16:9的关系)。可以引入此类限制以促进硬件实现,或出于其它原因引入此类限制,并且在本领域中是公知的。

[0104] 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大小,而不是隐式地假设该大小为输出图片大小。在该示例中,语法元素reference_pic_size_present_flag(614)选通参考图片维度(615)的有条件存在(同样,该数字指的是宽度和高度)。

[0105] 最后,示出了可能的解码图片宽度和高度的表。可以例如通过表指示(num_dec_pic_size_in_luma_samples_minus1)(616)来表示这样的表。“minus1”可以指对该语法元素的值的解释。例如,如果已编码值为零,则存在一个表条目。如果该值为5,则存在六个表条目。对于表中的每一“行”,已解码图片宽度和高度则包括在语法(617)中。

[0106] 可以使用图块组头中的语法元素dec_pic_size_idx(602)对所呈现的表条目(617)进行索引,从而允许每个图块群组具有不同的已解码大小——实际上是缩放因子。

[0107] 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施某些形式的参考图片重采样(以与所公开的主题完全不同的方式发信号通知)来支持空间可缩放性,从而

实现空间可缩放性。具体地,某些参考图片可以使用ARC样式的技术上采样到更高的分辨率,以形成空间增强层的基础。这些上采样的图片可以使用高分辨率的正常预测机制来修正,以增加细节。

[0108] 本文讨论的实施例可以在这样的环境中使用。在某些情况下,在相同或另一实施例中,NAL单元头中的值(例如时间ID字段)不仅可用于指示时间层,还可用于指示空间层。这样做对于某些系统设计可能有一定的好处。例如,基于NAL单元头时间ID值为时间层选择转发创建和优化的现有的选择转发单元(SFU,Selected Forwarding Unit)可以在不进行修改的情况下用于可缩放环境。为了实现这一点,可能需要已编码图片大小与时间层之间的映射由NAL单元头中的时间ID字段来指示。

[0109] 在一些视频编解码技术中,访问单元(AU)可以指在给定的时间实例处捕获的并组合成相应的图片、切片、图块和/或NAL单元比特流的一个或多个已编码图片、一个或多个切片、一个或多个图块、一个或多个NAL单元等。该给定的时间实例可以是例如合成时间。

[0110] 在HEVC和某些其他视频编解码技术中,图片顺序计数(POC)值可用于指示从解码图片缓冲器(DPB)存储的多个参考图片中选择的参考图片。当访问单元(AU)包括一个或多个图片、切片或图块时,属于同一AU的每个图片、切片或图块可以携带相同的POC值,从该POC值可以导出,它们是根据同一合成时间的内容创建的。换言之,在两个图片/切片/图块携带相同的给定POC值的情况下,该POC值可以指示属于同一AU并且具有相同合成时间的两个图片/切片/图块。相反地,具有不同POC值的两个图片/切片/图块可以指示属于不同AU并且具有不同合成时间的那些图片/切片/图块。

[0111] 在实施例中,因为访问单元可以包括具有不同POC值的图片、切片或图块,因此可以放宽这种刚性关系。通过允许在一个AU内使用不同的POC值,可以使用POC值来识别具有相同呈现时间的可能独立解码的图片/切片/图块。这进而可以支持多个可缩放层而无需改变参考图片选择信令,例如,参考图片集信令或参考图片列表信令,如下面更详细的描述。

[0112] 然而,对于具有不同POC值的其他图片/切片/图块,仍然希望能够仅根据POC值来识别图片/切片/图块所属的AU。这是可以实现的,如下所述。

[0113] 在实施例中,可以在高级语法结构(例如NAL单元头、切片头、图块组头、SEI消息、参数集或AU定界符)中发信号通知访问单元计数(AUC)。AUC值可用于识别哪些NAL单元、图片、切片或图块属于给定的AU。AUC值可以对应于不同的合成时间实例。AUC值可以等于POC值的倍数。通过将POC值除以一个整数值,可以计算AUC值。在某些情况下,除法运算会给解码器实现带来一定的负担。在这种情况下,在AUC值的编号空间中做一些小的限制可以允许用移位运算代替除法运算。例如,AUC值可以等于POC值范围内的最高有效位(MSB)值。

[0114] 在实施例中,可以在高级语法结构(例如NAL单元头、切片头、图块组头、SEI消息、参数集或AU定界符)中发信号通知每个AU的POC周期的值(poc_cycle_au)。poc_cycle_au可以指示多少个不同且连续的POC值可与同一AU相关联。例如,如果poc_cycle_au的值等于4,则POC值等于0到3(包括0和3)的图片、切片或图块可以与AUC值等于0的AU相关联,并且POC值等于4到7(包括4和7)的图片、切片或图块可以与AUC值等于1的AU相关联。因此,可以通过将POC值除以poc_cycle_au的值来推断AUC的值。

[0115] 在实施例中,poc_cycle_au的值可以从例如位于视频参数集(VPS)中的信息导出,该信息标识已编码视频序列中的空间或SNR层的数量。下面简要描述这种可能的关系的示

例。尽管如上所述的推导可以在VPS中节省几个比特,并因此可以提高编解码效率,但在一些实施例中,以层级低于视频参数集的、适当的高级语法结构来显式地对poc_cycle_au进行编码,以便能够针对给定的一小部分比特流(例如图片)来最小化poc_cycle_au。由于POC值和/或间接引用POC的语法元素的值可以用低级语法结构来编码,所以与通过上述推导过程所能节省的比特相比,这种优化可以节省更多的比特。

[0116] 在实施例中,图8示出了语法表的示例。该语法表用于在VPS(或SPS)中发信号通知vps_poc_cycle_au的语法元素,其中vps_poc_cycle_au指示用于已编码视频序列中的所有图片/切片的poc_cycle_au。该语法表还用于发信号通知slice_poc_cycle_au的语法元素,其中slice_poc_cycle_au在切片头中指示当前切片的poc_cycle_au。如果每个AU的POC值均匀增加,则VPS中的vps_contant_poc_cycle_per_au可以设置为1,并且可以在VPS中发信号通知vps_poc_cycle_au。在这种情况下,可以不显式地发信号通知slice_poc_cycle_au,并且可以通过将POC值除以vps_poc_cycle_au来计算每个AU的AUC值。如果每个AU的POC值不是均匀增加,则VPS中的vps_contant_poc_cycle_per_au可以设置为0。在这种情况下,可以不发信号通知vps_access_unit_cnt,而是可以针对每个切片或图片在切片头中发信号通知slice_access_unit_cnt。每个切片或图片可以具有不同的slice_access_unit_cnt值。可以通过将POC值除以slice_poc_cycle_au来计算每个AU的AUC值。

[0117] 图9示出了图示上述过程的示例的框图。例如,在操作S910中,可以解析VPS(或SPS),并且可以在操作S920处确定每个AU的POC周期在已编码视频序列内是否恒定。如果每个AU的POC周期是恒定的(在操作S920处为“是”),则在操作S930处,可以根据为已编码视频序列发信号通知的poc_cycle_au和特定访问单元的POC值计算特定访问单元的访问单元计数的值。如果每个AU的POC周期不是恒定的(在操作S920处为“否”),则在操作S940处,可以根据在图片级别发信号通知的poc_cycle_au和特定访问单元的POC值计算特定访问单元的访问单元计数的值。在操作S950处,可以解析新的VPS(或SPS)。

[0118] 在实施例中,即使图片、切片或图块的POC值可能不同,与具有相同AUC值的AU对应的图片、切片或图块也可以与同一解码或输出时间实例相关联。因此,在同一AU中的图片、切片或图块之间没有任何帧间解析/解码依赖性(dependency)的情况下,与该同一AU相关联的图片、切片或图块的全部或子集可以并行解码,并且可以在同一时间实例处输出。

[0119] 在实施例中,即使图片、切片或图块的POC值可能不同,与具有相同AUC值的AU对应的图片、切片或图块也可以与同一合成/显示时间实例相关联。当合成时间包含在容器格式中时,即使图片对应于不同的AU,但如果图片具有相同的合成时间,也可以在同一时间实例处显示。

[0120] 在实施例中,每个图片、切片或图块在同一AU中可以具有相同的时间标识符(temporal_id)。与时间实例对应的图片、切片或图块的全部或子集可以与同一时间子层相关联。在实施例中,每个图片、切片或图块在同一AU中可以具有相同或不同的空间层id(layer_id)。与时间实例对应的图片、切片或图块的全部或子集可以与相同或不同的空间层相关联。

[0121] 图7示出了具有自适应分辨率改变的temporal_id、layer_id、POC值和AUC值的组合的视频序列结构的示例。在该示例中,AUC=0的第一AU中的图片、切片或图块可以具有temporal_id=0和layer_id=0或1,而AUC=1的第二AU中的图片、切片或图块可以分别具

有temporal_id＝1和layer_id＝0或1。不管temporal_id和layer_id的值如何,每个图片的POC值都增加1。在该示例中,poc_cycle_au的值可以等于2。在实施例中,可以将poc_cycle_au的值设置为等于(空间可缩放性)层的数量。因此,在该示例中,POC的值增加2,而AUC的值增加1。

[0122] 在以上实施例中,可以通过使用HEVC中的现有参考图片集(RPS)信令或参考图片列表(RPL)信令来支持图片间或层间预测结构和参考图片指示的全部或子集。在RPS或RPL中,该所选参考图片可以通过发信号通知当前图片和所选参考图片之间的POC值或POC的增量值来指示。在实施例中,RPS和RPL可用于指示图片间或层间预测结构而无需改变信令,但是具有以下限制。如果参考图片的temporal_id的值大于当前图片的temporal_i的值,则当前图片可以不使用参考图片进行运动补偿或其他预测。如果参考图片的layer_id的值大于当前图片的layer_id的值,则当前图片可以不使用参考图片进行运动补偿或其他预测。

[0123] 在实施例中,可以在访问单元内的多个图片之间禁止将基于POC差的运动矢量缩放用于时间运动矢量预测。因此,尽管每个图片在访问单元内可以具有不同的POC值,但是运动矢量没有被缩放并用于访问单元内的时间运动矢量预测。这是因为在同一AU中具有不同POC的参考图片被认为具有相同的时间实例。因此,在该实施例中,当参考图片属于与当前图片相关联的AU时,运动矢量缩放功能可以返回1。

[0124] 在实施例中,当参考图片的空间分辨率不同于当前图片的空间分辨率时,可选地可以在多个图片之间禁止将基于POC差的运动矢量缩放用于时间运动矢量预测。当允许运动矢量缩放时,基于POC差和当前图片与参考图片之间的空间分辨率之比来缩放运动矢量。

[0125] 在实施例中,对于时间运动矢量预测,尤其是在poc_cycle_au具有非均匀值时(例如当vps_contant_poc_cycle_per_au＝＝0时),可以基于AUC差而不是POC差来缩放运动矢量。否则(例如当vps_contant_poc_cycle_per_au＝＝1时),基于AUC差的运动矢量缩放可能与基于POC差的运动矢量缩放相同。

[0126] 在实施例中,当基于AUC差对运动矢量进行缩放时,与当前图片相同的AU(具有相同的AUC值)中的参考运动矢量不基于AUC差进行缩放,而是无需缩放就用于运动矢量预测或者基于当前图片和参考图片之间的空间分辨率之比进行缩放后用于运动矢量预测。

[0127] 在实施例中,AUC值可用于识别AU的边界,并用于假想参考解码器(HRD)操作,这需要输入和输出具有AU粒度的定时。在实施例中,可以输出具有AU中最高层的已解码图片以进行显示。AUC值和layer_id值可用于识别输出图片。

[0128] 在实施例中,图片可以包括一个或多个子图片。每个子图片可以覆盖该图片的局部区域或整个区域。一个子图片支持的区域可以与另一个子图片支持的区域重叠或不重叠。一个或多个子图片覆盖的区域可以覆盖或不覆盖图片的整个区域。如果一个图片包括一个子图片,则该子图片支持的区域可以与该图片支持的区域相同。

[0129] 在实施例中,可以通过与用于已编码图片的类似的编码方法对子图片进行编码。子图片可以独立地进行编码,或者可以根据另一个子图片或已编码图片进行编码。子图片可以对另一个子图片或已编码图片具有或不具有任何解析依赖性。

[0130] 在实施例中,已编码子图片可以包含在一个或多个层中。层中的已编码子图片可以具有不同的空间分辨率。原始子图片可以在空间上重新采样(例如上采样或下采样),用不同的空间分辨率参数进行编码,并且包含在与层对应的比特流中。

[0131] 在实施例中,具有(W,H)的子图片可以被编码并包含在与层0对应的已编码比特流中,其中W指示子图片的宽度,H指示子图片的高度。从具有原始空间分辨率的子图片上采样(或下采样)的子图片(具有($W*S_{w,k}$,$H*S_{h,k}$))可以被编码并包含在与层k对应的已编码比特流中,其中$S_{w,k}$、$S_{h,k}$指示水平和垂直方向的重采样率。如果$S_{w,k}$、$S_{h,k}$的值大于1,则重采样可以是上采样。然而,如果$S_{w,k}$、$S_{h,k}$的值小于1,则重采样可以是下采样。

[0132] 在实施例中,一个层中的已编码子图片的视觉质量可能与同一子图片或不同子图片中的另一层中的已编码子图片的视觉质量不同。例如,层n中的子图片i可以用量化参数$Q_{i,n}$进行编码,而层m中的子图片j可以用量化参数$Q_{j,m}$进行编码。

[0133] 在实施例中,一个层中的已编码子图片可以是可独立解码的(independently decodable),其对相同局部区域的另一层中的已编码子图片没有任何解析或解码依赖性。可以在不参考相同局部区域的另一子图片层的情况下独立解码的子图片层可以是独立子图片层。独立子图片层中的已编码子图片对同一子图片层中的先前已编码子图片可以具有或不具有解码或解析依赖性,但是该已编码子图片对另一子图片层中的已编码图片可以不具有任何依赖性。

[0134] 在实施例中,一个层中的已编码子图片可以是从属解码的(dependently decodable),其对相同局部区域的另一层中的已编码子图片具有任何解析或解码依赖性。可以在参考相同局部区域的另一子图片层的情况下从属解码的子图片层可以是从属(dependent)子图片层。从属子图片中的已编码子图片可以参考属于同一子图片的已编码子图片、同一子图片层中的先前已编码子图片、或者两种参考子图片。

[0135] 在实施例中,已编码子图片可以包括一个或多个独立子图片层和一个或多个从属子图片层。然而,对于已编码子图片,可以存在至少一个独立子图片层。独立子图片层的层标识符(layer_id)的值可以等于0,该层标识符可以存在于NAL单元头或另一高级语法结构中,其值等于0。layer_id等于0的子图片层可以是基础子图片层。

[0136] 在实施例中,图片可以包括一个或多个前景子图片和一个背景子图片。背景子图片支持的区域可以等于图片的区域。前景子图片支持的区域可以与背景子图片支持的区域重叠。背景子图片可以是基础子图片层,而前景子图片可以是非基础(增强)子图片层。一个或多个非基础子图片层可以参考同一基础层进行解码。每个layer_id等于a的非基础子图片层可以参考layer_id等于b的非基础子图片层,其中a大于b。

[0137] 在实施例中,图片可以包括具有或不具有背景子图片的一个或多个前景子图片。每个子图片可以具有其自己的基础子图片层和一个或多个非基础(增强)层。每个基础子图片层都可以被一个或多个非基础子图片层参考。每个layer_id等于a的非基础子图片层可以参考layer_id等于b的非基础子图片层,其中a大于b。

[0138] 在实施例中,图片可以包括具有或不具有背景子图片的一个或多个前景子图片。(基础或非基础)子图片层中的每个已编码子图片可以被属于同一子图片的一个或多个非基础层子图片和不属于同一子图片的一个或多个非基础层子图片参考。

[0139] 在实施例中,图片可以包括具有或不具有背景子图片的一个或多个前景子图片。层a中的子图片可以被进一步划分为同一层中的多个子图片。层b中的一个或多个已编码子图片可以参考层a中划分后的子图片。

[0140] 在实施例中,已编码视频序列(CVS)可以是一组已编码图片。CVS可以包括一个或

多个已编码子图片序列(CSPS),其中CSPS可以是覆盖图片的相同局部区域的一组已编码子图片。CSPS可以具有与已编码视频序列相同或不同的时间分辨率。

[0141] 在实施例中,CSPS可以被编码并包含在一个或多个层中。CSPS可以包括一个或多个CSPS层。对与CSPS对应的一个或多个CSPS层进行解码可以重建对应于相同局部区域的子图片序列。

[0142] 在实施例中,与CSPS对应的CSPS层的数量可以与与另一CSPS对应的CSPS层的数量相同或不同。

[0143] 在实施例中,CSPS层可以具有与另一CSPS层不同的时间分辨率(例如帧速率)。原始(未压缩的)子图片序列可以在时间上重新采样(例如上采样或下采样),用不同的时间分辨率参数进行编码,并且包含在与层对应的比特流中。

[0144] 在实施例中,具有帧速率F的子图片序列可以被编码并包含在与层0对应的已编码比特流中,而原始子图片序列中的具有$F*S_{t,k}$的时间上采样(或下采样)子图片序列可以被编码并包含在与层k对应的已编码比特流中,其中$S_{t,k}$指示层k的时间采样率。如果$S_{t,k}$的值大于1,则时间重采样过程可以是帧速率上变换(frame rate up conversion)。然而,如果$S_{t,k}$的值小于1,则时间重采样过程可以是帧速率下变换(frame rate down conversion)。

[0145] 在实施例中,当具有CSPS层a的子图片被具有CSPS层b的子图片参考以进行运动补偿或任何层间预测时,如果CSPS层a的空间分辨率不同于CSPS层b的空间分辨率,则CSPS层a中的已解码像素被重新采样并用作参考。重采样过程可以使用上采样滤波或下采样滤波。

[0146] 图10示出了示例视频流,该示例视频流包括layer_id等于0的背景视频CSPS和多个前景CSPS层。虽然已编码子图片可以包括一个或多个CSPS层,但是不属于任何前景CSPS层的背景区域可以包括基础层。基础层可以包含背景区域和前景区域,而增强CSPS层可以包含前景区域。在同一区域,增强CSPS层可能比基础层具有更好的视觉质量。增强CSPS层可以参考对应同一区域的重建像素和基础层的运动矢量。

[0147] 在实施例中,在视频文件中,与基础层对应的视频比特流包含在轨道中,而与每个子图片对应的CSPS层包含在单独的轨道中。

[0148] 在实施例中,与基础层对应的视频比特流包含在轨道中,而具有相同layer_id的CSPS层包含在单独的轨道中。在该示例中,与层k对应的轨道仅包括与层k对应的CSPS层。

[0149] 在实施例中,每个子图片的每个CSPS层存储在单独的轨道中。每个轨道对一个或多个其他轨道可以具有或不具有任何解析或解码依赖性。

[0150] 在实施例中,每个轨道可以包含与子图片的全部或子集的CSPS层的层i至层j对应的比特流,其中0<i=<j=<k,k是CSPS的最高层。

[0151] 在实施例中,图片包括一个或多个相关联的媒体数据,这些相关联的媒体数据包括深度图、阿尔法(α)图、3D几何数据、占用图等。这种相关联的定时媒体数据可以被划分成一个或多个数据子流,每个数据子流对应一个子图片。

[0152] 图11示出了基于多层子图片方法的视频会议的示例。在视频流中,包含与背景图片对应的一个基础层视频比特流和与前景子图片对应的一个或多个增强层视频比特流。每个增强层视频比特流可以对应一个CSPS层。在显示器中,默认显示与基础层对应的图片。它包含一个或多个用户的画中画(PIP)。当通过客户端的控件选择了特定用户时,与所选择的用户对应的增强CSPS层可以被解码并以增强的质量或空间分辨率进行显示。图13示出了该

操作的示意图。

[0153] 图12示出了图示上述过程的示例的框图。例如,在操作S1210中,可以解码具有多个层的视频比特流。在操作S1220中,可以识别背景区域和一个或多个前景子图片。在操作S1230处,可以确定是否选择了特定子图片区域,例如前景子图片之一。如果选择了特定子图片区域(操作S1240处为“是”),则可以解码并显示增强子图片。如果未选择特定子图片区域(在操作S1240处为“否”),则可以解码并显示背景区域。

[0154] 在实施例中,网络中间盒(例如路由器)可以根据其带宽选择要发送给用户的层的子集。图片/子图片组织可用于带宽适配。例如,如果用户没有带宽,则路由器会由于其重要性或基于所使用的设置而剥离层或选择一些子图片,并且可以动态地做到这一点以适应带宽。

[0155] 图13示出了涉及360视频的使用情况的实施例。当球形360图片(例如图片1310)被投影到平面图片上时,投影360图片可以被划分成多个子图片作为基础层。例如,该多个子图片可以包括后子图片、顶部子图片、右子图片、左子图片、前子图片和底部子图片。特定子图片的增强层(例如前子图片)可以被编码并发送到客户端。解码器能够对包括所有子图片的基础层和所选择的子图片的增强层进行解码。当当前视口(viewport)与所选择的子图片相同时,所显示的图片与具有增强层的已解码子图片相比可能具有更高的质量。否则,可以以较低质量显示具有基础层的已解码图片。

[0156] 在实施例中,用于显示的任何布局信息可以作为补充信息(例如SEI消息或元数据)存在于文件中。一个或多个已解码子图片可以根据用信号通知的布局信息重新定位和显示。布局信息可以由流式传输服务器或广播设备发信号通知,或者可以由网络实体或云服务器重新生成,或者可以由用户的定制设置来确定。

[0157] 在实施例中,当输入图片被划分成一个或多个(矩形)子区域时,每个子区域可以被编码为独立层。与局部区域对应的每个独立层可以具有唯一的layer_id值。对于每个独立层,可以发信号通知子图片大小和位置信息,例如,图片大小(宽度,高度)、左上角的偏移信息(x_offset,y_offset)。图14示出了划分的子图片的布局、其子图片大小和位置信息以及其对应的图片预测结构的示例。包括一个或多个子图片大小和一个或多个子图片位置的布局信息可以在高级语法结构(例如一个或多个参数集、切片头或图块组头、或SEI消息)中发信号通知。

[0158] 在实施例中,与独立层对应的每个子图片可以在AU内具有唯一的POC值。当通过使用RPS或RPL结构中的语法元素来指示存储在DPB中的图片中的参考图片时,可以使用与层对应的每个子图片的POC值。

[0159] 在实施例中,为了指示(层间)预测结构,可以不使用layer_id并且可以使用POC(增量)值。

[0160] 在实施例中,与层(或局部区域)对应的、POC值等于N的子图片可以用作或可以不用作与同一层(或同一局部区域)对应的、POC值等于N+K的子图片的参考图片,以进行运动补偿预测。在大多数情况下,数量K的值可以等于(独立)层的最大数量,该数量可以与子区域的数量相同。

[0161] 在实施例中,图15示出了图14的扩展情况。当输入图片被划分成多个(例如四个)子区域时,每个局部区域可以以一个或多个层来编码。在这种情况下,独立层的数量可以等

于子区域的数量,并且一个或多个层可以对应于子区域。因此,每个子区域可以以一个或多个独立层以及零个或多个从属层来编码。

[0162] 在实施例中,在图15中,输入图片可以被划分成四个子区域。作为示例,右上子区域可以被编码为两个层,即层1和层4,而右下子区域可以被编码为两个层,即层3和层5。在这种情况下,层4可以参考层1进行运动补偿预测,而层5可以参考层3进行运动补偿。

[0163] 在实施例中,可以(可选地)禁用跨层边界的环路内(in-loop)滤波(例如解块滤波、自适应环路内滤波、整形器、双边滤波或任何基于深度学习的滤波)。

[0164] 在实施例中,可以(可选地)禁用跨层边界的运动补偿预测或块内复制。

[0165] 在实施例中,可以可选地处理用于运动补偿预测的边界填充或子图片边界处的环路内滤波。可以在高级语法结构(例如一个或多个参数集(VPS、SPS、PPS或APS)、切片头或图块组头、或SEI消息)中发信号通知一个标志,以指示边界填充是否被处理。

[0166] 在实施例中,可以在VPS或SPS中发信号通知一个或多个子区域(或者一个或多个子图片)的布局信息。图16A示出了VPS中的语法元素的示例,图16B示出了SPS中的语法元素的示例。在该示例中,在VPS中发信号通知vps_sub_picture_dividing_flag。该标志可以指示一个或多个输入图片是否被划分成多个子区域。当vps_sub_picture_dividing_flag的值等于0时,与当前VPS对应的一个或多个已编码视频序列中的一个或多个输入图片可以不被划分成多个子区域。在这种情况下,输入图片大小可以等于已编码图片大小(pic_width_in_luma_samples,pic_height_in_luma_samples),该已编码图片大小在SPS中发信号通知。当vps_sub_picture_dividing_flag的值等于1时,该一个或多个输入图片可以被划分成多个子区域。在这种情况下,在VPS中发信号通知语法元素vps_full_pic_width_in_luma_samples和vps_full_pic_height_in_luma_samples。vps_full_pic_width_in_luma_samples和vps_full_pic_height_in_luma_samples的值可以分别等于该一个或多个输入图片的宽度和高度。

[0167] 在实施例中,vps_full_pic_width_in_luma_samples和vps_full_pic_height_in_luma_samples的值可以不用于解码,而是用于合成和显示。

[0168] 在实施例中,当vps_sub_picture_dividing_flag的值等于1时,可以在SPS中发信号通知语法元素pic_offset_x和pic_offset_y,该语法元素对应于一个或多个特定层。在这种情况下,在SPS中发信号通知的已编码图片大小(pic_width_in_luma_samples,pic_height_in_luma_samples)可以等于与特定层对应的子区域的宽度和高度。而且,可以在SPS中发信号通知子区域的左上角的位置(pic_offset_x,pic_offset_y)。

[0169] 在实施例中,子区域的左上角的位置信息(pic_offset_x,pic_offset_y)可以不用于解码,而是用于合成和显示。

[0170] 在实施例中,可以在参数集或SEI消息中发信号通知一个或多个输入图片的全部或子集子区域的布局信息(大小和位置)、以及层之间的从属性信息。图17示出了语法元素的示例,用于指示子区域的布局、层之间的从属性以及子区域与一个或多个层之间的关系的信息。在该示例中,语法元素num_sub_region指示当前已编码视频序列中(矩形)子区域的数量。语法元素num_layers指示当前已编码视频序列中层的数量。num_layers的值可以等于或大于num_sub_region的值。当任何子区域以单个层进行编码时,num_layers的值可以等于num_sub_region的值。当一个或多个子区域以多个层进行编码时,num_layers的值

可以大于num_sub_region的值。语法元素direct_dependency_flag[i][j]指示从第j层到第i层的从属性。num_layers_for_region[i]指示与第i个子区域相关联的层的数量。sub_region_layer_id[i][j]指示与第i个子区域相关联的第j层的layer_id。sub_region_offset_x[i]和sub_region_offset_y[i]分别指示第i个子区域的左上角的水平和垂直位置。sub_region_width[i]和sub_region_height[i]分别指示第i个子区域的宽度和高度。

[0171] 在实施例中,可以在高级语法结构(例如VPS、DPS、SPS、PPS、APS或SEI消息)中发信号通知一个或多个语法元素,该一个或多个语法元素指定输出层集,以指示要输出的具有或不具有配置文件层级信息的多个层中的一个。参考图18,可以在VPS中发信号通知语法元素num_output_layer_sets,该语法元素num_output_layer_sets指示参考VPS的已编码视频序列中的输出层集(OLS)的数量。对于每个输出层集,可以发信号通知与输出层的数量一样多的output_layer_flag。

[0172] 在实施例中,output_layer_flag[i]等于1指定输出第i层。vps_output_layer_flag[i]等于0指定不输出第i层。

[0173] 在实施例中,可以在高级语法结构(例如VPS、DPS、SPS、PPS、APS或SEI消息)中发信号通知一个或多个语法元素,该一个或多个语法元素指定每个输出层集的配置文件层级信息。仍然参考图18,可以在VPS中发信号通知语法元素num_profile_tile_level,该语法元素num_profile_tile_level指示参考VPS的已编码视频序列中的每个OLS的配置文件层级信息的数量。对于每个输出层集,可以发信号通知与输出层的数量一样多的用于配置文件层级信息的语法元素集或指示配置文件层级信息中的条目的特定配置文件层级信息的索引。

[0174] 在实施例中,profile_tier_level_idx[i][j]指定应用于第i个OLS的第j层的profile_tier_level()语法结构在VPS中的profile_tier_level()语法结构的列表中的索引。

[0175] 在实施例中,参考图19,当最大层数大于1(vps_max_layers_minus1>0)时,可以发信号通知语法元素num_profile_tile_level和/或num_output_layer_sets。

[0176] 在实施例中,参考图19,指示第i个输出层集的输出层信令的模式的语法元素vps_output_layers_mode[i]可以存在于VPS中。

[0177] 在实施例中,vps_output_layers_mode[i]等于0指定使用第i个输出层集仅输出最高层。vps_output_layer_mode[i]等于1指定使用第i个输出层集输出所有层。vps_output_layer_mode[i]等于2指定输出的层是vps_output_layer_flag[i][j]等于1且使用第i个输出层集的层。可以保留更多的值。

[0178] 在实施例中,根据用于第i个输出层集的vps_output_layers_mode[i]的值,可以发信号通知或不发信号通知output_layer_flag[i][j]。

[0179] 在实施例中,参考图19,对于第i个输出层集,可以存在标志vps_ptl_signal_flag[i]。根据vps_ptl_signal_flag[i]的值,可以发信号通知或不发信号通知第i个输出层集的配置文件层级信息。

[0180] 在实施例中,参考图20,可以在高级语法结构(例如VPS、DPS、SPS、PPS、APS或SEI消息)中发信号通知当前CVS中的子图片的数量max_subpics_minus1。

[0181] 在实施例中,参考图20,当子图片的数量大于1(max_subpics_minus1>0)时,可以

发信号通知第i个子图片的子图片标识符sub_pic_id[i]。

[0182] 在实施例中,可以在VPS中发信号通知一个或多个语法元素,该一个或多个语法元素指示属于每个输出层集的每个层的子图片标识符。参考图20,sub_pic_id_layer[i][j][k]指示存在于第i个输出层集的第j层中的第k个子图片。利用这些信息,解码器可以识别出对于特定输出层集的每个层哪个子图片被解码和输出。

[0183] 在实施例中,图片头(PH)是包含语法元素的语法结构,这些语法元素应用于已编码图片的全部切片。图片单元(PU)是一组NAL单元,它们根据指定的分类规则相互关联,在解码顺序上连续,并且正好包含一个已编码图片。PU可包含图片头(PH)和组成已编码图片的一个或多个VCL NAL单元。

[0184] 在实施例中,SPS(RBSP)在被参考之前可用于解码过程,包括在TemporalId等于0的至少一个AU中,或者通过外部方式提供。

[0185] 在实施例中,SPS(RBSP)在被参考之前可用于解码过程,包括在CVS中TemporalId等于0的至少一个AU中,或者通过外部方式提供,CVS包含一个或多个参考SPS的PPS。

[0186] 在实施例中,SPS(RBSP)在被一个或多个PPS参考之前可用于解码过程,包括在至少一个PU中,或者通过外部方式提供,该至少一个PU的nuh_layer_id等于在CVS中参考SPS NAL单元的PPS NAL单元的最低nuh_layer_id值,CVS包含一个或多个参考SPS的PPS。

[0187] 在实施例中,SPS(RBSP)在被一个或多个PPS参考之前可用于解码过程,包括在至少一个PU中,或者通过外部方式提供,该至少一个PU的TemporalId等于0并且nuh_layer_id等于参考SPS NAL单元的PPS NAL单元的最低nuh_layer_id值。

[0188] 在实施例中,SPS(RBSP)在被一个或多个PPS参考之前可用于解码过程,包括在至少一个PU中,或者通过外部方式提供,该至少一个PU的TemporalId等于0并且nuh_layer_id等于在CVS中参考SPS NAL单元的PPS NAL单元的最低nuh_layer_id值,CVS包含一个或多个参考SPS的PPS。

[0189] 在同一或另一实施例中,pps_seq_parameter_set_id可以为所参考的SPS指定sps_seq_parameter_set_id的值。pps_seq_parameter_set_id的值在CLVS中的已编码图片所参考的所有PPS中可以相同。

[0190] 在同一或另一实施例中,CVS中具有特定sps_seq_parameter_set_id值的所有SPS NAL单元可以具有相同的内容。

[0191] 在同一或另一实施例中,不管nuh_layer_id值如何,SPS NAL单元都可以共享sps_seq_parameter_set_id的相同值空间。

[0192] 在同一或另一实施例中,SPS NAL单元的nuh_layer_id值可以等于参考SPS NAL单元的PPS NAL单元的最低nuh_layer_id值。

[0193] 在实施例中,当nuh_layer_id等于m的SPS被nuh_layer_id等于n的一个或多个PPS参考时,nuh_layer_id等于m的层可以与nuh_layer_id等于n的层或nuh_layer_id等于m的层的(直接或间接)参考层相同。

[0194] 在实施例中,PPS(RBSP)在被参考之前可用于解码过程,包括在至少一个AU中,或者通过外部方式提供,该至少一个AU的TemporalId等于PPS NAL单元的TemporalId。

[0195] 在实施例中,PPS(RBSP)在被参考之前可用于解码过程,包括在至少一个AU中,或者通过外部方式提供,该至少一个AU的TemporalId等于CVS中的PPS NAL单元的

TemporalId,CVS包含参考PPS的一个或多个PH(或已编码切片NAL单元)。

[0196] 在实施例中,PPS(RBSP)在被一个或多个PH(或已编码切片NAL单元)参考之前可用于解码过程,包括在至少一个PU中,或者通过外部方式提供,该至少一个PU的nuh_layer_id等于CVS中参考PPS NAL单元的已编码切片NAL单元的最低nuh_layer_id值,CVS包含参考PPS的一个或多个PH(或已编码切片NAL单元)。

[0197] 在实施例中,PPS(RBSP)在被一个或多个PH(或已编码切片NAL单元)参考之前可用于解码过程,包括在至少一个PU中,或者通过外部方式提供,该至少一个PU的TemporalId等于PPS NAL单元的TemporalId并且nuh_layer_id等于CVS中参考PPS NAL单元的已编码切片NAL单元的最低nuh_layer_id值,CVS包含参考PPS的一个或多个PH(或已编码切片NAL单元)。

[0198] 在同一或另一实施例中,PH中的ph_pic_parameter_set_id可以为使用中所参考的PPS指定pps_pic_parameter_set_id的值。pps_seq_parameter_set_id的值在CLVS中的已编码图片所参考的所有PPS中可以相同。

[0199] 在同一或另一实施例中,PU中具有特定pps_pic_parameter_set_id值的所有PPS NAL单元可以具有相同的内容。

[0200] 在同一或另一实施例中,不管nuh_layer_id值如何,PPS NAL单元都可以共享pps_pic_parameter_set_id的相同值空间。

[0201] 在同一或另一实施例中,PPS NAL单元的nuh_layer_id值可以等于参考NAL单元(其参考PPS NAL单元)的已编码切片NAL单元的最低nuh_layer_id值。

[0202] 在实施例中,当nuh_layer_id等于m的PPS被nuh_layer_id等于n的一个或多个已编码切片NAL单元参考时,nuh_layer_id等于m的层可以与nuh_layer_id等于n的层或nuh_layer_id等于m的层的(直接或间接)参考层相同。

[0203] 在实施例中,PPS(RBSP)在被参考之前可用于解码过程,包括在至少一个AU中,或者通过外部方式提供,该至少一个AU的TemporalId等于PPS NAL单元的TemporalId。

[0204] 在实施例中,PPS(RBSP)在被参考之前可用于解码过程,包括在至少一个AU中,或者通过外部方式提供,该至少一个AU的TemporalId等于CVS中的PPS NAL单元的TemporalId,CVS包含参考PPS的一个或多个PH(或已编码切片NAL单元)。

[0205] 在实施例中,PPS(RBSP)在被一个或多个PH(或已编码切片NAL单元)参考之前可用于解码过程,包括在至少一个PU中,或者通过外部方式提供,该至少一个PU的nuh_layer_id等于CVS中参考PPS NAL单元的已编码切片NAL单元的最低nuh_layer_id值,CVS包含参考PPS的一个或多个PH(或已编码切片NAL单元)。

[0206] 在实施例中,PPS(RBSP)在被一个或多个PH(或已编码切片NAL单元)参考之前可用于解码过程,包括在至少一个PU中,或者通过外部方式提供,该至少一个PU的TemporalId等于PPS NAL单元的TemporalId并且nuh_layer_id等于CVS中参考PPS NAL单元的已编码切片NAL单元的最低nuh_layer_id值,CVS包含参考PPS的一个或多个PH(或已编码切片NAL单元)。

[0207] 在同一或另一实施例中,PH中的ph_pic_parameter_set_id可以为使用中所参考的PPS指定pps_pic_parameter_set_id的值。pps_seq_parameter_set_id的值在CLVS中的已编码图片所参考的所有PPS中可以相同。

[0208] 在同一或另一实施例中,PU中具有特定pps_pic_parameter_set_id值的所有PPS NAL单元可以具有相同的内容。

[0209] 在同一或另一实施例中,不管nuh_layer_id值如何,PPS NAL单元都可以共享pps_pic_parameter_set_id的相同值空间。

[0210] 在同一或另一实施例中,PPS NAL单元的nuh_layer_id值可以等于参考NAL单元(其参考PPS NAL单元)的已编码切片NAL单元的最低nuh_layer_id值。

[0211] 在实施例中,当nuh_layer_id等于m的PPS被nuh_layer_id等于n的一个或多个已编码切片NAL单元参考时,nuh_layer_id等于m的层可以与nuh_layer_id等于n的层或nuh_layer_id等于m的层的(直接或间接)参考层相同。

[0212] 在实施例中,如图21所示,图片参数集中的pps_subpic_id[i]可以指定第i个子图片的子图片ID。pps_subpic_id[i]语法元素的长度为pps_subpic_id_len_minus1+1比特。

[0213] 对于0到sps_num_subpics_minus1(包括端值)范围内的每个i值,变量SubpicIdVal[i]可以导出如下:

[0214]
```
for( i = 0; i <= sps_num_subpics_minus1; i++ )
    if( subpic_id_mapping_explicitly_signalled_flag )
        SubpicIdVal[ i ] = subpic_id_mapping_in_pps_flag ? pps_subpic_id[ i ] :
sps_subpic_id[ i ]
    else
        SubpicIdVal[ i ] = i
```

[0215] 在同一或另一实施例中,对于0到sps_num_subpics_minus1(包括端值)范围内的i和j的任何两个不同值,SubpicIdVal[i]可以不等于SubpicIdVal[j]。

[0216] 在同一或另一实施例中,当当前图片不是CLVS的第一个图片时,对于0到sps_num_subpics_minus1(包括端值)范围内的每个i值,如果SubpicIdVal[i]的值不等于同一层中按解码顺序的前一个图片的SubpicIdVal[i]的值,则当前图片中的子图片(具有子图片索引i)的所有已编码切片NAL单元的nal_unit_type可以等于IDR_W_RADL到CRA_NUT(包括端值)范围内的特定值。

[0217] 在同一或另一实施例中,当当前图片不是CLVS的第一个图片时,对于0到sps_num_subpics_minus1(包括端值)范围内的每个i值,如果SubpicIdVal[i]的值不等于同一层中按解码顺序的前一个图片的SubpicIdVal[i]的值,则sps_independent_subpics_flag可以等于1。

[0218] 在同一或另一实施例中,当当前图片不是CLVS的第一个图片时,对于0到sps_num_subpics_minus1(包括端值)范围内的每个i值,如果SubpicIdVal[i]的值不等于同一层中按解码顺序的前一个图片的SubpicIdVal[i]的值,则subpic_treated_as_pic_flag[i]和loop_filter_across_subpic_enabled_flag[i]可以等于1。

[0219] 在同一或另一实施例中,当当前图片不是CLVS的第一个图片时,对于0到sps_num_subpics_minus1(包括端值)范围内的每个i值,如果SubpicIdVal[i]的值不等于同一层中按解码顺序的前一个图片的SubpicIdVal[i]的值,则sps_independent_subpics_flag可以等于1或者subpic_treated_as_pic_flag[i]和loop_filter_across_subpic_enabled_flag[i]可以等于1。

[0220] 在同一或另一实施例中,当一个子图片被独立编码而不参考另一个子图片时,区域的子图片标识符的值可以在已编码视频序列中改变。

[0221] 在实施例中,图片可以划分成一个或多个子图片。每个子图片可以进一步划分成一个或多个切片。需要一种有效的信令方法来发信号通知子图片分区信息。

[0222] 样本可以以CTB为单位进行处理。每个亮度CTB的阵列大小在宽度和高度上都是CtbSizeY(以样本为单位)。每个色度CTB的阵列的宽度和高度分别是CtbWidthC和CtbHeightC(以样本为单位)。每个CTB被分配一个分区信令,以识别用于帧内或帧间预测和用于变换编码的块大小。分区是递归四叉树分区。四叉树的根与CTB相关联。四叉树被分割,直到到达叶,该叶称为四叉树叶。当分量宽度不是CTB大小的整数时,右部分量边界处的CTB是不完整的。当分量高度不是CTB大小的整数倍时,底部分量边界处的CTB是不完整的。

[0223] 每个子图片的宽度和高度信息可以以CtbSizeY为单位在SPS中发信号通知。例如,在图22中,subpic_width_minus1[i]加1指定第i个子图片的宽度,单位为CtbSizeY。该语法元素的长度为Ceil(Log2((pic_width_max_in_luma_samples+CtbSizeY-1)>>CtbLog2SizeY))比特。当不存在时,subpic_width_minus1[i]的值被推断为等于((pic_width_max_in_luma_samples+CtbSizeY-1)>>CtbLog2SizeY)-subpic_ctu_top_left_x[i]-1。subpic_height_minus1[i]加1指定第i发子图片的高度,单位为CtbSizeY。该语法元素的长度为Ceil(Log2((pic_height_max_in_luma_samples+CtbSizeY-1)>>CtbLog2SizeY))比特。当不存在时,subpic_height_minus1[i]的值被推断为等于((pic_height_max_in_luma_samples+CtbSizeY-1)>>CtbLog2SizeY)-subpic_ctu_top_left_y[i]-1。

[0224] 当图片宽度大于或等于CtbSizeY时,每个子图片的宽度可以大于或等于CtbSizeY。当图片高度大于或等于CtbSizeY时,每个子图片的高度可以大于或等于CtbSizeY。

[0225] 如果图片宽度不大于CtbSizeY并且图片高度不大于CtbSizeY,则图片可以不划分成多于一个子图片。在这种情况下,子图片的数量可以等于1。

[0226] 当pic_width_max_in_luma_samples不大于CtbSizeY并且pic_height_max_in_luma_samples不大于CtbSizeY时,subpic_info_present_flag的值可以等于0。当subpic_info_present_flag等于0时,不存在用于子图片分区信息的显式信令,并且图片中的子图片的数量等于1。

[0227] 在同一或另一实施例中,sps_subpic_id_len_minus1加1指定用于表示语法元素sps_subpic_id[i]、语法元素pps_subpic_id[i](当存在时)和语法元素slice_subpic_id(当存在时)的比特数。sps_subpic_id_len_minus1的值可以在0到15(包括端值)的范围内。1<<(sps_subpic_id_len_minus1)的值可以大于或等于sps_num_subpics_minus1+1。

[0228] 在同一或另一实施例中,当子图片的数量等于1时,subpic_info_present_flag可以等于0,并且子图片分区信息可以不显式地发信号通知,因为在这种情况下,子图片宽度和高度信息等于图片的宽度和高度信息并且子图片的左上位置等于图片的左上位置。

[0229] 例如,subpic_ctu_top_left_x[i]指定第i个子图片的左上CTU的水平位置,单位为CtbSizeY。该语法元素的长度为Ceil(Log2((pic_width_max_in_luma_samples+CtbSizeY-1)>>CtbLog2SizeY))比特。当不存在时,subpic_ctu_top_left_x[i]的值被推断

为等于0。subpic_ctu_top_left_y[i]指定第i个子图片的左上CTU的垂直位置,单位为CtbSizeY。该语法元素的长度为Ceil(Log2((pic_height_max_in_luma_samples+CtbSizeY-1)>>CtbLog2SizeY))比特。当不存在时,subpic_ctu_top_left_y[i]的值被推断为等于0。subpic_width_minus1[i]加1指定第i个子图片的宽度,单位为CtbSizeY。该语法元素的长度为Ceil(Log2((pic_width_max_in_luma_samples+CtbSizeY-1)>>CtbLog2SizeY))比特。当不存在时,subpic_width_minus1[i]的值被推断为等于((pic_width_max_in_luma_samples+CtbSizeY-1)>>CtbLog2SizeY)-subpic_ctu_top_left_x[i]-1。subpic_height_minus1[i]加1指定第i个子图片的高度,单位为CtbSizeY。该语法元素的长度为Ceil(Log2((pic_height_max_in_luma_samples+CtbSizeY-1)>>CtbLog2SizeY))比特。当不存在时,subpic_height_minus1[i]的值被推断为等于((pic_height_max_in_luma_samples+CtbSizeY-1)>>CtbLog2SizeY)-subpic_ctu_top_left_y[i]-1。

[0230] 在同一或另一实施例中,当子图片的数量大于1时,subpic_info_present_flag可以等于1,并且子图片分区信息可以在参数集中显式地发信号通知,如图22所示。

[0231] 例如,在图22中,sps_num_subpics_minus2加2指定CLVS中的每个图片中的子图片的数量。sps_num_subpics_minus2的值可以在0到Ceil(pic_width_max_in_luma_samples÷CtbSizeY)*Ceil(pic_height_max_in_luma_samples÷CtbSizeY)-2(包括端值)的范围内。当不存在时,sps_num_subpics_minus2的值被推断为等于0。

[0232] 在同一实施例中,对于0到sps_num_subpics_minus1(包括端值)范围内的i,列表SubpicWidthInTiles[i]和SubpicHeightInTiles[i]分别指定图块列和行中的第i个子图片的宽度和高度;对于0到sps_num_subpics_minus1(包括端值)范围内的i,列表subpicHeightLessThanOneTileFlag[i]指定第i个子图片的高度是否小于一个图块行,这些列表可如下导出:

[0233]
```
for( i = 0; i <= sps_num_subpics_minus2 + 1; i++ ) {
  leftX = subpic_ctu_top_left_x[ i ]
  rightX = leftX + subpic_width_minus1[ i ]
  SubpicWidthInTiles[ i ] = ctbToTileColIdx[ rightX ] + 1 − ctbToTileColIdx[ leftX ]
  topY = subpic_ctu_top_left_y[ i ]
  bottomY = topY + subpic_height_minus1[ i ]
  SubpicHeightInTiles[ i ] = ctbToTileRowIdx[ bottomY ] + 1 − ctbToTileRowIdx[ topY ]
```

[0234]
```
  if( SubpicHeightInTiles[ i ] == 1 &&
      subpic_height_minus1[ i ] + 1 < RowHeight[ ctbToTileRowIdx[ topY ] ] )
    subpicHeightLessThanOneTileFlag[ i ] = 1
else
    subpicHeightLessThanOneTileFlag[ i ] = 0
}
```

[0235] 当rect_slice_flag等于1时,对于0到num_slices_in_pic_minus1(包括端值)范围内的i,列表NumCtusInSlice[i]指定第i个切片中CTU的数量;对于0到num_slices_in_pic_minus1(包括端值)范围内的i,列表SliceTopLeftTileIdx[i]指定包含切片中的第一

个CTU的图块的图块索引;对于0到num_slices_in_pic_minus1(包括端值)范围内的i,以及0到NumCtusInSlice[i]-1(包括端值)范围内的j,矩阵CtbAddrInSlice[i][j]指定第i个切片中的第j个CTB的图片光栅扫描地址,变量NumSlicesInTile[i]指定包含第i个切片的图块中的切片的数量,这些列表、矩阵和变量可以导出如下:

```
if( single_slice_per_subpic_flag ) {
  for( i = 0; i   <=     sps_num_subpics_minus2 + 1; i++ ) {
    NumCtusInSlice[ i ] = 0
    if( subpicHeightLessThanOneTileFlag[ i ] ) /* The slice consists of a number of CTU
rows in a tile. */
      AddCtbsToSlice( i, subpic_ctu_top_left_x[ i ],
        subpic_ctu_top_left_x[ i ] + subpic_width_minus1[ i ] + 1,
subpic_ctu_top_left_y[ i ],
        subpic_ctu_top_left_y[ i ] + subpic_height_minus1[ i ] + 1 )
```

[0236]

```
    else { /* The slice consists of a number of complete tiles covering a rectangular
region. */
      tileX = CtbToTileColBd[ subpic_ctu_top_left_x[ i ] ]
      tileY = CtbToTileRowBd[ subpic_ctu_top_left_y[ i ] ]
      for( j = 0; j < SubpicHeightInTiles[ i ]; j++ )
        for( k = 0; k < SubpicWidthInTiles[ i ]; k++ )
          AddCtbsToSlice( i, tileColBd[ tileX + k ], tileColBd[ tileX + k + 1 ],
tileRowBd[ tileY + j ],
            tileRowBd[ tileY + j + 1 ] )
    }
```

```
    }
  } else {
    tileIdx = 0
    for( i = 0; i <= num_slices_in_pic_minus1; i++ )
      NumCtusInSlice[ i ] = 0
    for( i = 0; i <= num_slices_in_pic_minus1; i++ ) {
      SliceTopLeftTileIdx[ i ] = tileIdx
      tileX = tileIdx % NumTileColumns
      tileY = tileIdx / NumTileColumns
      if( i < num_slices_in_pic_minus1 ) {
        sliceWidthInTiles[ i ] = slice_width_in_tiles_minus1[ i ] + 1
        sliceHeightInTiles[ i ] = slice_height_in_tiles_minus1[ i ] + 1
      } else {
        sliceWidthInTiles[ i ] = NumTileColumns − tileX
        sliceHeightInTiles[ i ] = NumTileRows − tileY
        NumSlicesInTile[ i ] = 1
      }
      if( slicWidthInTiles[ i ] == 1 && sliceHeightInTiles[ i ] == 1 ) {
        if( num_exp_slices_in_tile[ i ] == 0 ) {
```

[0237]

```
          NumSlicesInTile[ i ] = 1
          sliceHeightInCtus[ i ] =
RowHeight[ SliceTopLeftTileIdx[ i ] / NumTileColumns ]
        } else {
          remainingHeightInCtbsY =
RowHeight[ SliceTopLeftTileIdx[ i ] / NumTileColumns ]
          for( j = 0; j < num_exp_slices_in_tile[ i ] − 1; j++ ) {
            sliceHeightInCtus[ i + j ] = exp_slice_height_in_ctus_minus1[ i ][ j ] + 1
            remainingHeightInCtbsY −= sliceHeightInCtus[ i + j ]
          }
          uniformSliceHeight = exp_slice_height_in_ctus_minus1[ i ][ j ] + 1
          while( remainingHeightInCtbsY >= uniformSliceHeight ) {
            sliceHeightInCtus[ i + j ] = uniformSliceHeight
            remainingHeightInCtbsY −= uniformSliceHeight
            j++
          }
          if( remainingHeightInCtbsY > 0 ) {
            sliceHeightInCtus[ i + j ] = remainingHeightInCtbsY
            j++
          }
```

```
                NumSlicesInTile[ i ] = j
            }
            ctbY = tileRowBd[ tileY ]
            for( j = 0; j < NumSlicesInTile[ i ]; j++ ) {
                AddCtbsToSlice( i + j, tileColBd[ tileX ], tileColBd[ tileX + 1 ],
                    ctbY, ctbY + sliceHeightInCtus[ i + j ] )
                ctbY += sliceHeightInCtus[ i + j ]
            }
            i += NumSlicesInTile[ i ] - 1
        } else
            for( j = 0; j < sliceHeightInTiles[ i ]; j++ )
                for( k = 0; k < sliceWidthInTiles[ i ]; k++ )
[0238]              AddCtbsToSlice( i, tileColBd[ tileX + k ], tileColBd[ tileX + k + 1 ],
                        tileRowBd[ tileY + j ], tileRowBd[ tileY + j + 1 ] )
        if( i < num_slices_in_pic_minus1 ) {
            if( tile_idx_delta_present_flag )
                tileIdx += tile_idx_delta[ i ]
            else {
                tileIdx += sliceWidthInTiles[ i ]
                if( tileIdx % NumTileColumns == 0 )
                    tileIdx += ( sliceHeightInTiles[ i ] - 1 ) * NumTileColumns
            }
        }
    }
}
```

[0239] 图23是用于对已编码视频比特流进行解码的示例过程2300的流程图。在一些实施方式中,图23的一个或多个过程框可以由解码器210执行。在一些实施方式中,图23的一个或多个过程框可以由与解码器210分离或包括解码器210的另一设备或一组设备执行,例如编码器203。

[0240] 如图23所示,过程2300可以包括当当前图片的至少一个子图片的数量大于1时,获取一个标志,所述标志指示是否显式地发信号通知子图片分区信息(框2310)。在实施例中,所述第一标志可以对应于subpic_info_present_flag。

[0241] 如图23进一步所示,过程2300可以包括当所述标志指示显式地发信号通知所述子图片分区信息时,获取所述子图片分区信息(框2320)。

[0242] 如图23进一步所示,过程2300可以包括基于所述子图片分区信息解码所述当前图片(框2330)。

[0243] 在实施例中,所述标志和所述子图片分区信息可以在序列参数集(SPS)中发信号通知。

[0244] 在实施例中,所述至少一个子图片的数量可以在所述SPS中发信号通知。

[0245] 在实施例中,当所述当前图片的图片大小小于所述当前图片的亮度分量的编码树单元的大小时,所述至少一个子图片的数量等于1。所述当前图片的亮度分量的编码树单元的大小可以对应于CtbSizeY。

[0246] 在实施例中,当所述至少一个子图片的数量等于1时,所述标志可以指示不显式地发信号通知所述子图片分区信息。

[0247] 在实施例中,述子图片分区信息可以指示所述至少一个子图片中的子图片的大小。

[0248] 在实施例中,所述子图片的大小可以包括所述子图片的高度和宽度。

[0249] 在实施例中,所述子图片的大小可以在所述子图片分区信息中以所述当前图片的亮度分量的编码树单元的大小为单位来指示。所述当前图片的亮度分量的编码树单元的大小可以对应于CtbSizeY。

[0250] 在实施例中,基于所述子图片分区信息指示的所述子图片的大小,确定另一子图片的大小。

[0251] 虽然图23示出了过程2300的示例框,但是在一些实施方式中,过程2300可以包括比图23中所描绘的更多的框、更少的框、不同的框或不同排列的框。另外或可替代地,过程2300的两个或多于两个的框可以并行执行。

[0252] 此外,所提出的方法可以由处理电路(例如,一个或多个处理器或一个或多个集成电路)来实现。在一示例中,该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,以执行所提出的方法中的一个或多个。

[0253] 上述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24示出了计算机系统2400,其适于实现所公开主题的一些实施例。

[0254] 所述计算机软件可通过任何合适的机器代码或计算机语言进行编码,通过汇编、编译、链接等机制创建包括指令的代码,所述指令可由一个或多个计算机中央处理单元(CPU),图形处理单元(GPU)等直接执行或通过译码、微代码等方式执行。

[0255] 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。

[0256] 图24所示的用于计算机系统2400的组件本质上是示例性的,并不用于对实现本公开实施例的计算机软件的使用范围或功能进行任何限制。也不应将组件的配置解释为与计算机系统2400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或要求。

[0257] 计算机系统2400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)。

[0258] 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2401、鼠标2402、触控板2403、触摸屏2410和相关图形适配器2450、数据手套、操纵杆2405、麦克风2406、扫描仪2407、照相机2408。

[0259] 计算机系统2400还可以包括某些人机界面输出设备。这种人机界面输出设备可以通过例如触觉输出、声音、光和嗅觉/味觉来刺激一个或多个人类用户的感觉。这样的人机界面输出设备可包括触觉输出设备(例如通过触摸屏2410、数据手套或操纵杆2405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2409、耳机(未示出))、视觉输出设备(例如,包括阴极射线管(CRT)屏幕、液晶显示(LCD)屏幕、等离子屏幕、有机发光二极管(OLED)屏幕的屏幕2410,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)。

[0260] 计算机系统2400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)2420或类似介质2421的光学介质、拇指驱动器2422、可移动硬盘驱动器或固体状态驱动器2423,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。

[0261] 本领域技术人员还应当理解,结合当前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。

[0262] 计算机系统2400还可以包括通往一个或多个通信网络(2455)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。网络还包括诸如以太网、无线局域网、蜂窝网络(包括全球移动通信系统(GSM)、第三代(3G)、第四代(4G)、第五代(5G)、长期演进(LTE)等)等的局域网、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(2454),用于连接到某些通用数据端口或外围总线(2449)(例如,计算机系统2400的通用串行总线(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2400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)。作为示例,网络2455可以使用网络接口2454连接到外围总线2449。通过使用这些网络中的任何一个,计算机系统2400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(2454)可使用某些协议和协议栈。

[0263] 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2400的核心2440。

[0264] 核心2440可包括一个或多个中央处理单元(CPU)2441、图形处理单元(GPU)2442、以现场可编程门阵列(FPGA)2443形式的专用可编程处理单元、用于特定任务的硬件加速器2444等。这些设备以及只读存储器(ROM)2445、随机存取存储器(RAM)2446、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(SSD)等)2447等可通过系统总线2448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2448,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2448,或通过外围总线2449进行连接。外围总线的体系结构包括外围组件互联(PCI)、通用串行总线USB等。

[0265] CPU 2441、GPU 2442、FPGA2443和加速器2444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 2445或RAM 2446中。过渡数据也可以存储在RAM 2446中,而永久数据可以存储在例如内部大容量存储器2447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 2441、GPU 2442、大容量存储器2447、ROM 2445、RAM 2446等紧密关联。

[0266] 所述计算机可读介质上可具有计算机代码,用于执行各种计算机实现的操作。介质和计算机代码可以是为本公开的目的而特别设计和构造,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。

[0267] 作为实施例而非限制,具有体系结构2400的计算机系统,特别是核心2440,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2440的特定存储器,例如核心内部大容量存储器2447或ROM 2445。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2440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2440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 2446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2444)中的功能,该电路可以代替软件或与软件一起运行以执行本文所述的特定过程或特定过程的特定部分。在适当的情况下,对软件的引用可以包括逻辑,反之亦然。在适当的情况下,对计算机可读介质的引用可包括存储执行软件的电路(如集成电路(IC)),包含执行逻辑的电路,或两者兼备。本公开包括任何合适的硬件和软件组合。

[0268] 虽然本公开已对多个示例性实施例进行了描述,但实施例的各种变更、排列和各种等同替换均属于本公开的范围内。因此应理解,本领域技术人员能够设计多种系统和方法,所述系统和方法虽然未在本文中明确示出或描述,但其体现了本公开的原则,因此属于本公开的精神和范围之内。

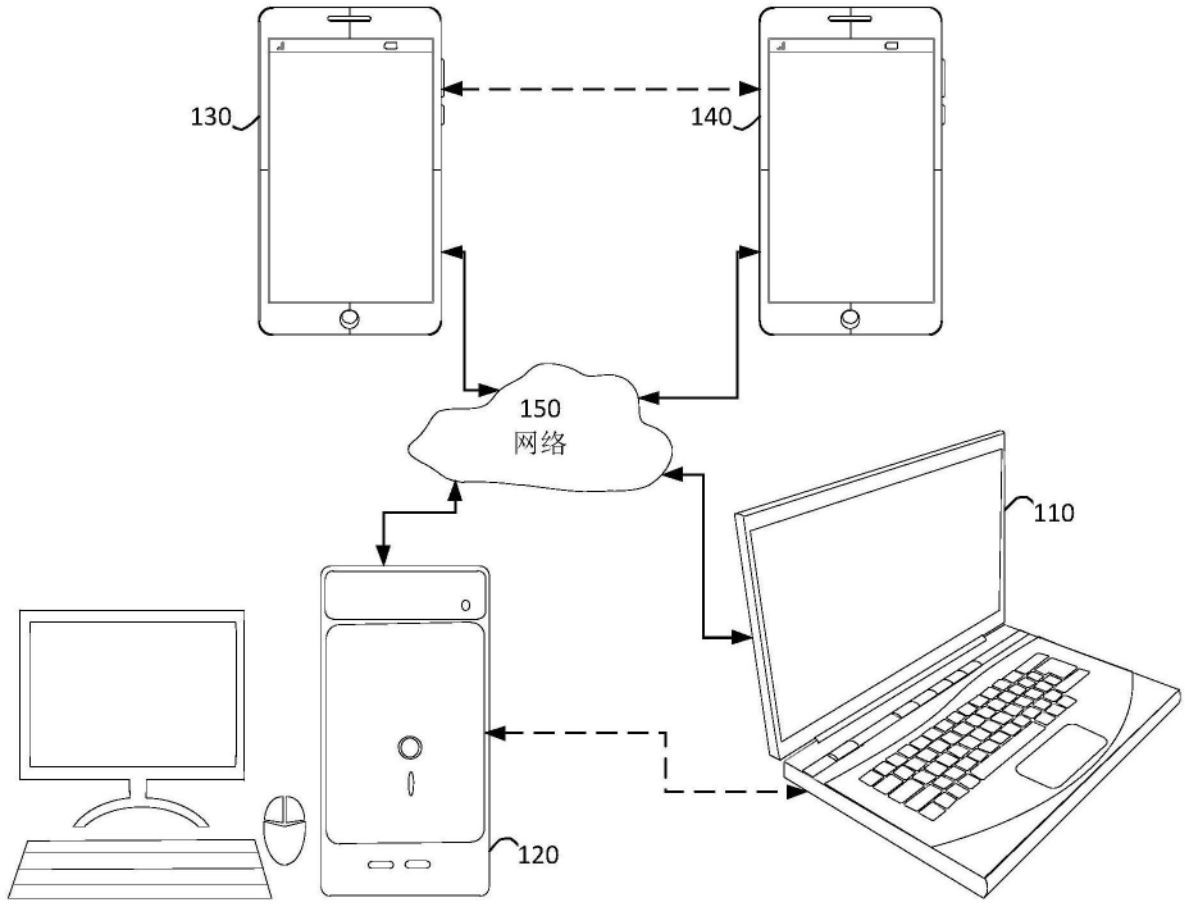


图1

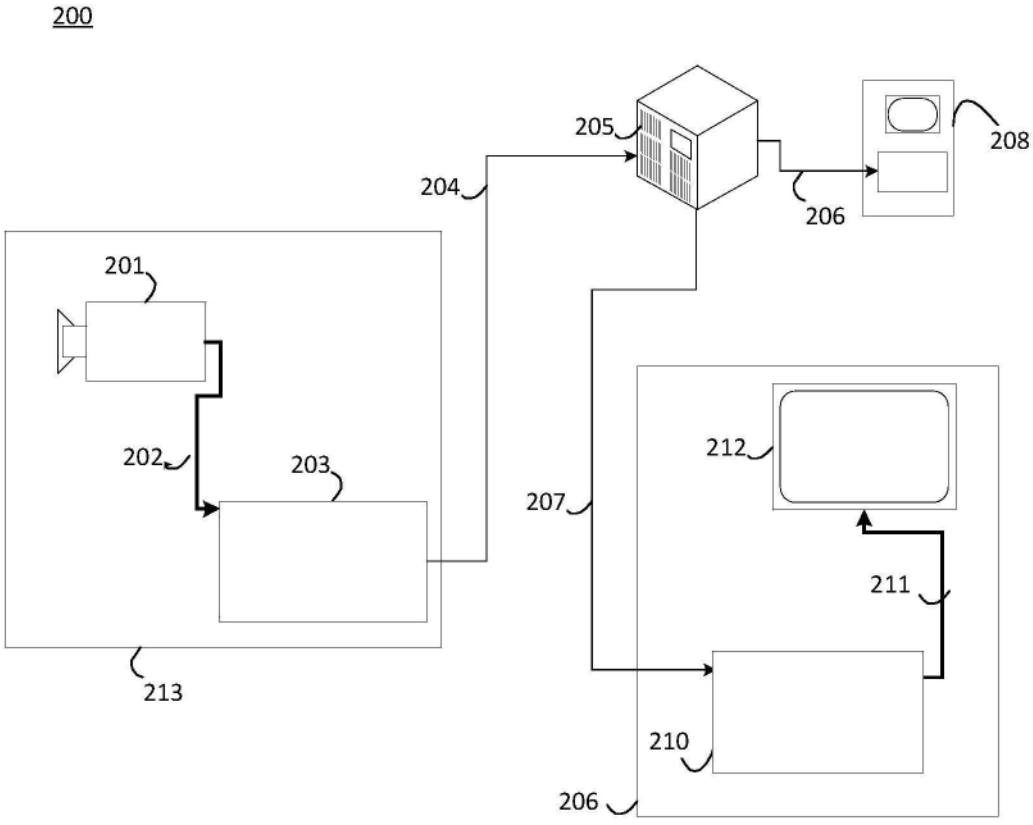


图2

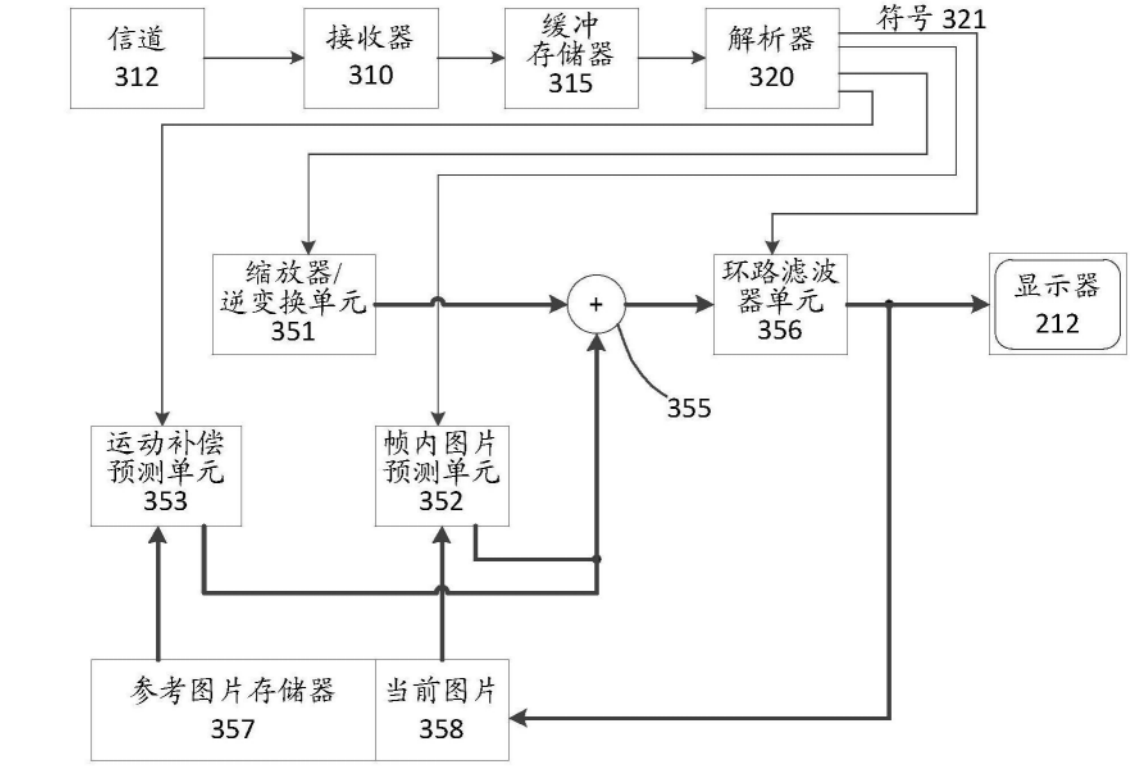

解码器 210
信道
312
接收器
310
缓冲
存储器
315
解析器
320
符号 321
缩放器/
逆变换单元
351
+
环路滤波
器单元
356
显示器
212
355
运动补偿
预测单元
353
帧内图片
预测单元
352
参考图片存储器
357
当前图片
358

图3

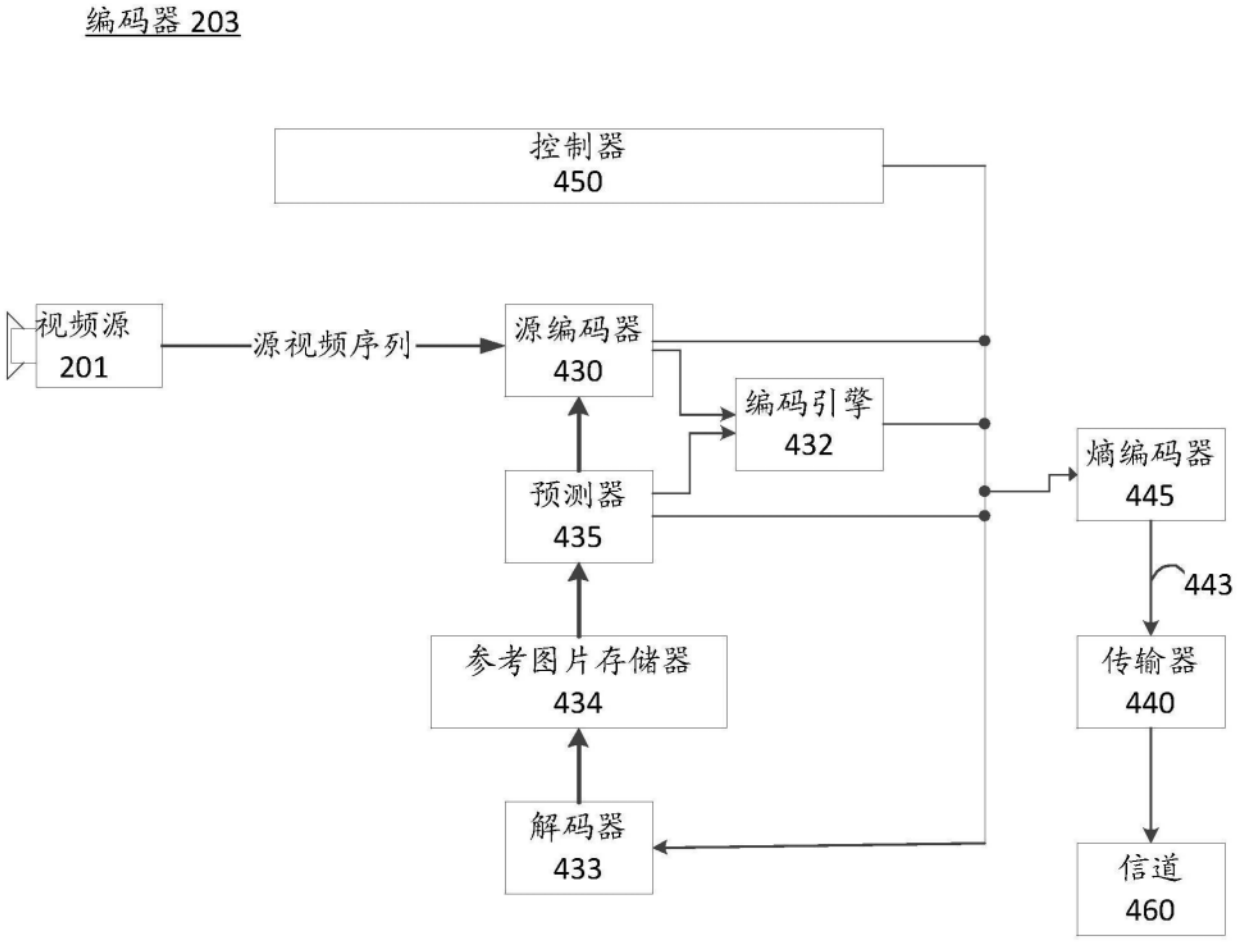

编码器 203
控制器
450
视频源
201
源视频序列
源编码器
430
编码引擎
432
熵编码器
445
预测器
435
443
参考图片存储器
434
传输器
440
解码器
433
信道
460

图4

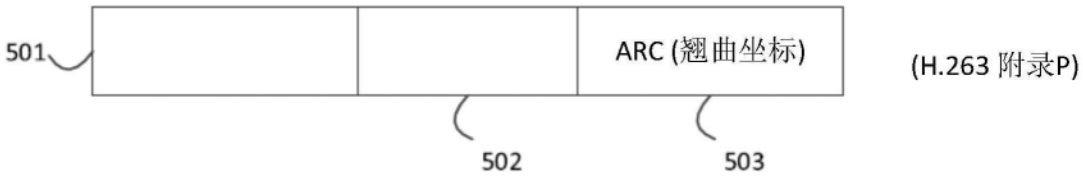

501
ARC (翘曲坐标)
(H.263 附录P)
502
503

图5A

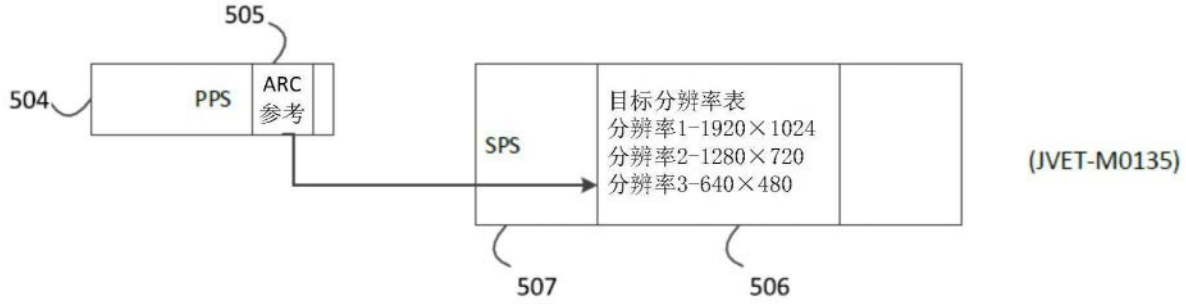

505
504
PPS
ARC
参考
SPS
目标分辨率表
分辨率1-1920×1024
分辨率2-1280×720
分辨率3-640×480
(JVET-M0135)
507
506

图5B

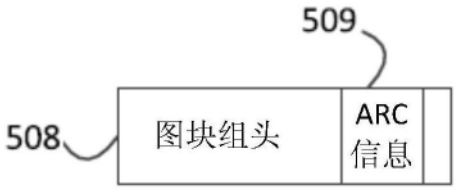


图5C



图5D

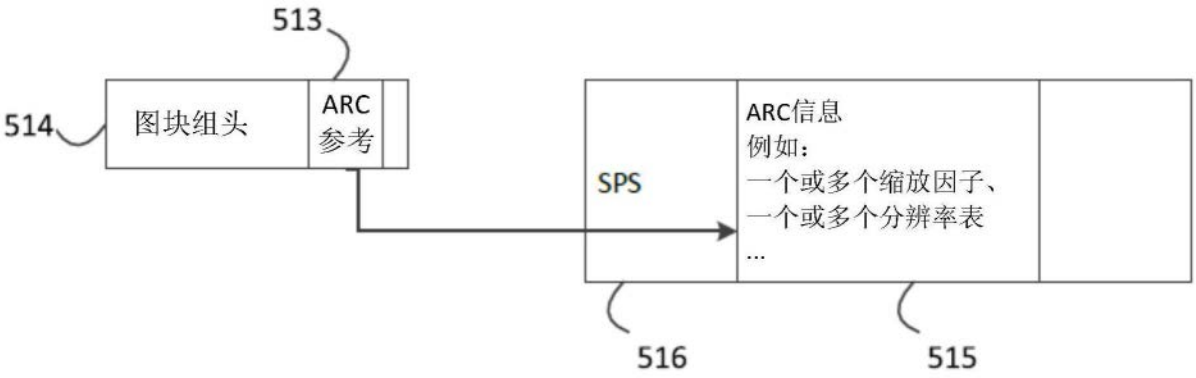


图5E

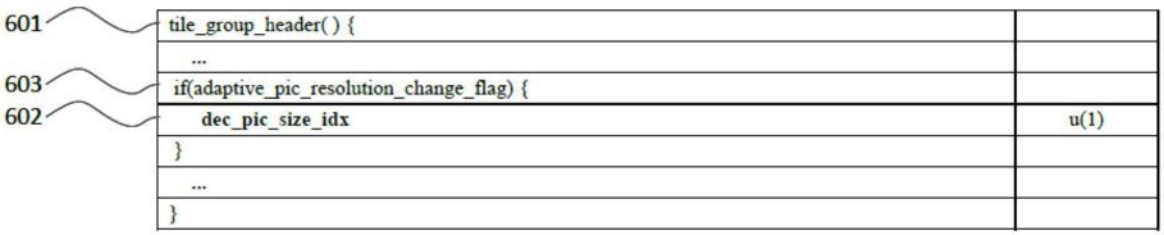
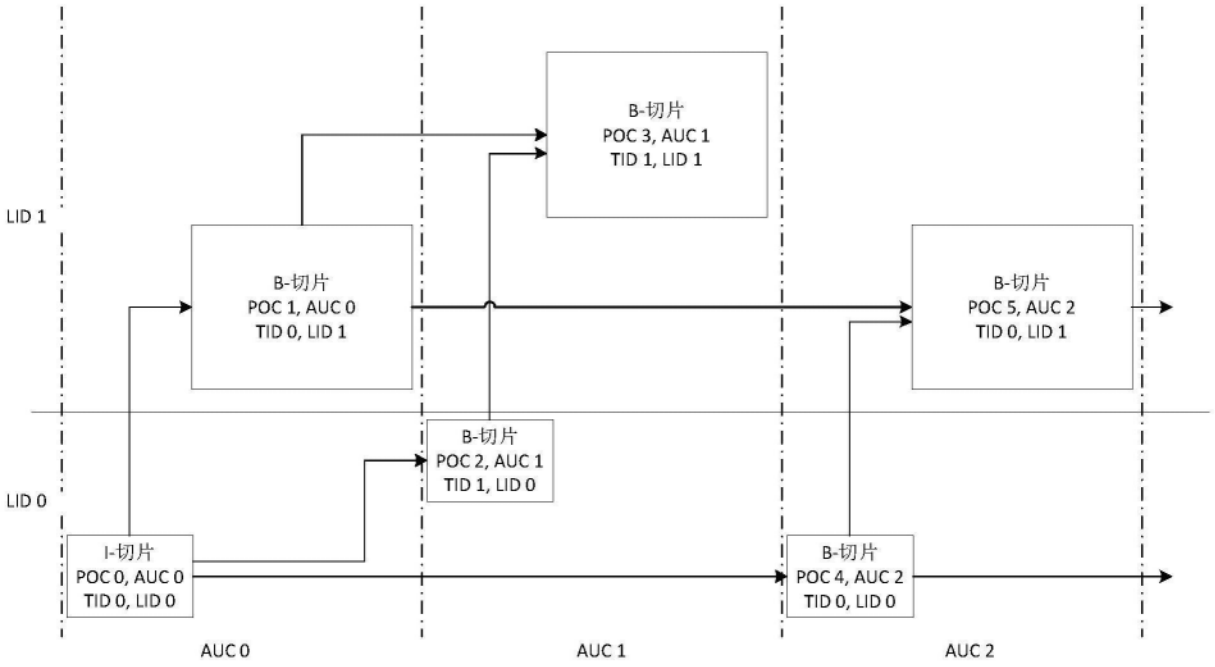


图6A

610	seq_parameter_set_rbsp( ) {	描述符
	...	
611	adaptive_pic_resolution_change_flag	u(1)
612	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)
	}	
616	num_dec_pic_size_in_luma_samples_minus1	ue(v)
	for(i = 0; i <= num_dec_pic_size_in_luma_samples_minus1; i++) {	
617	dec_pic_width_in_luma_samples[i]	ue(v)
	dec_pic_height_in_luma_samples[i]	ue(v)
	}	
	}	
	...	
	}	

图6B



POC : 图片顺序计数
AUC : 访问单元技术
TID : 时间ID
LID : 层ID

图7

video_parameter_set_rbsp( ) {	描述符
vps_video_parameter_set_id	u(4)
vps_max_layers_minus1	u(8)
for(i = 0; i <= vps_max_layers_minus1; i++) {	
vps_included_layer_id[i]	u(7)
vps_reserved_zero_bit	u(1)
}	
vps_constraint_info_present_flag	u(1)
vps_constant_poc_cycle_per_au	u(1)
if(vps_constant_poc_cycle_per_au)	
vps_poc_cycle_au	u(8)
…	
}	

slice_header( ) {	描述符
slice_pic_parameter_set_id	ue(v)
if(rect_slice_flag || NumBricksInPic > 1)	
slice_address	u(v)
if(!rect_slice_flag && !single_brick_per_slice_flag)	
num_bricks_in_slice_minus1	ue(v)
slice_type	ue(v)
if(NalUnitType == GRA_NUT)	
recovery_poc_cnt	se(v)
slice_pic_order_cnt_lsb	u(v)
…	
if(!vps_constant_poc_cycle_per_au)	
slice_poc_cycle_au	u(8)
}	

图8

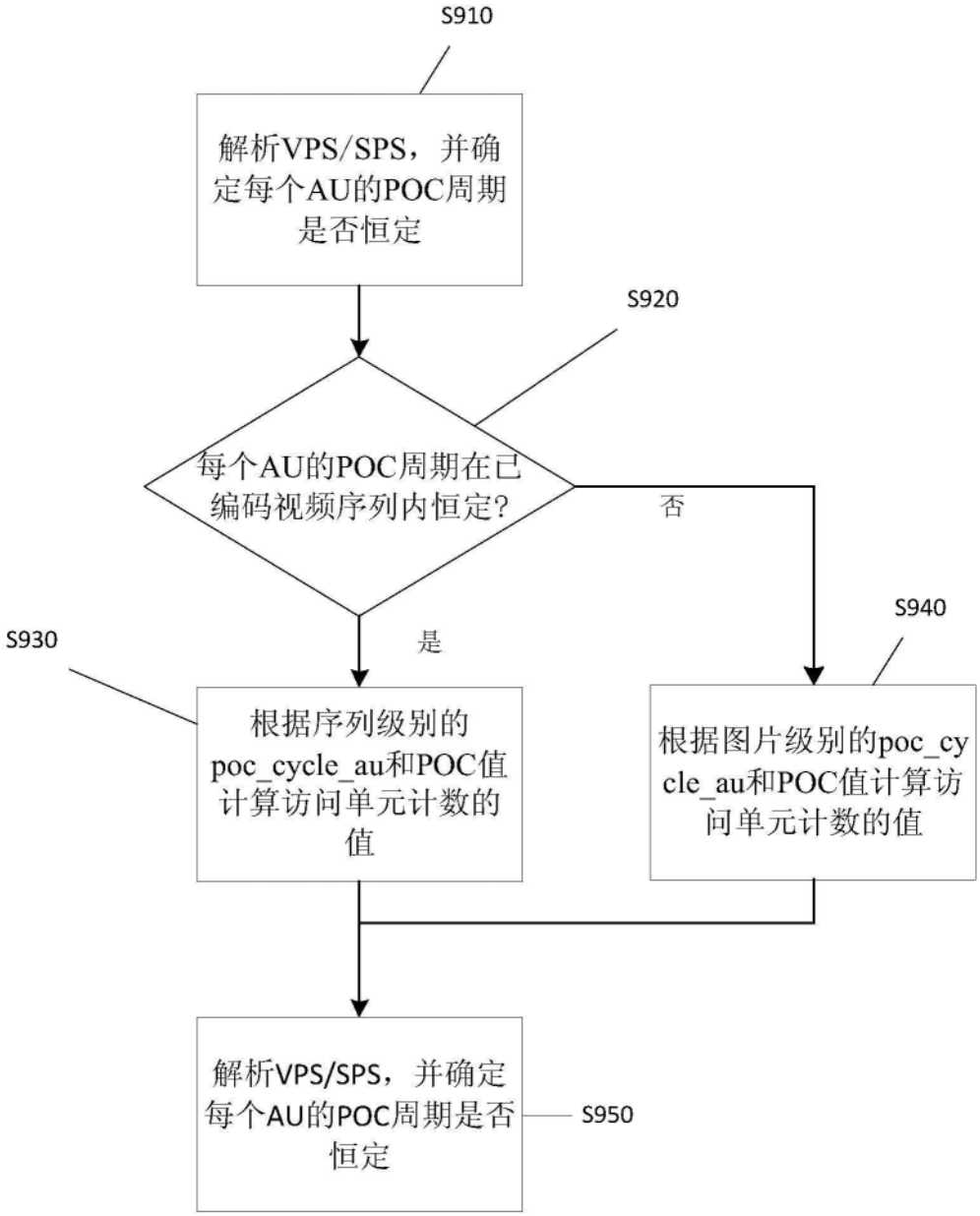

S910
解析VPS/SPS，并确定每个AU的POC周期是否恒定
S920
每个AU的POC周期在已编码视频序列内恒定?
否
是
S930
根据序列级别的poc_cycle_au和POC值计算访问单元计数的值
S940
根据图片级别的poc_cycle_au和POC值计算访问单元计数的值
解析VPS/SPS，并确定每个AU的POC周期是否恒定
S950

图9

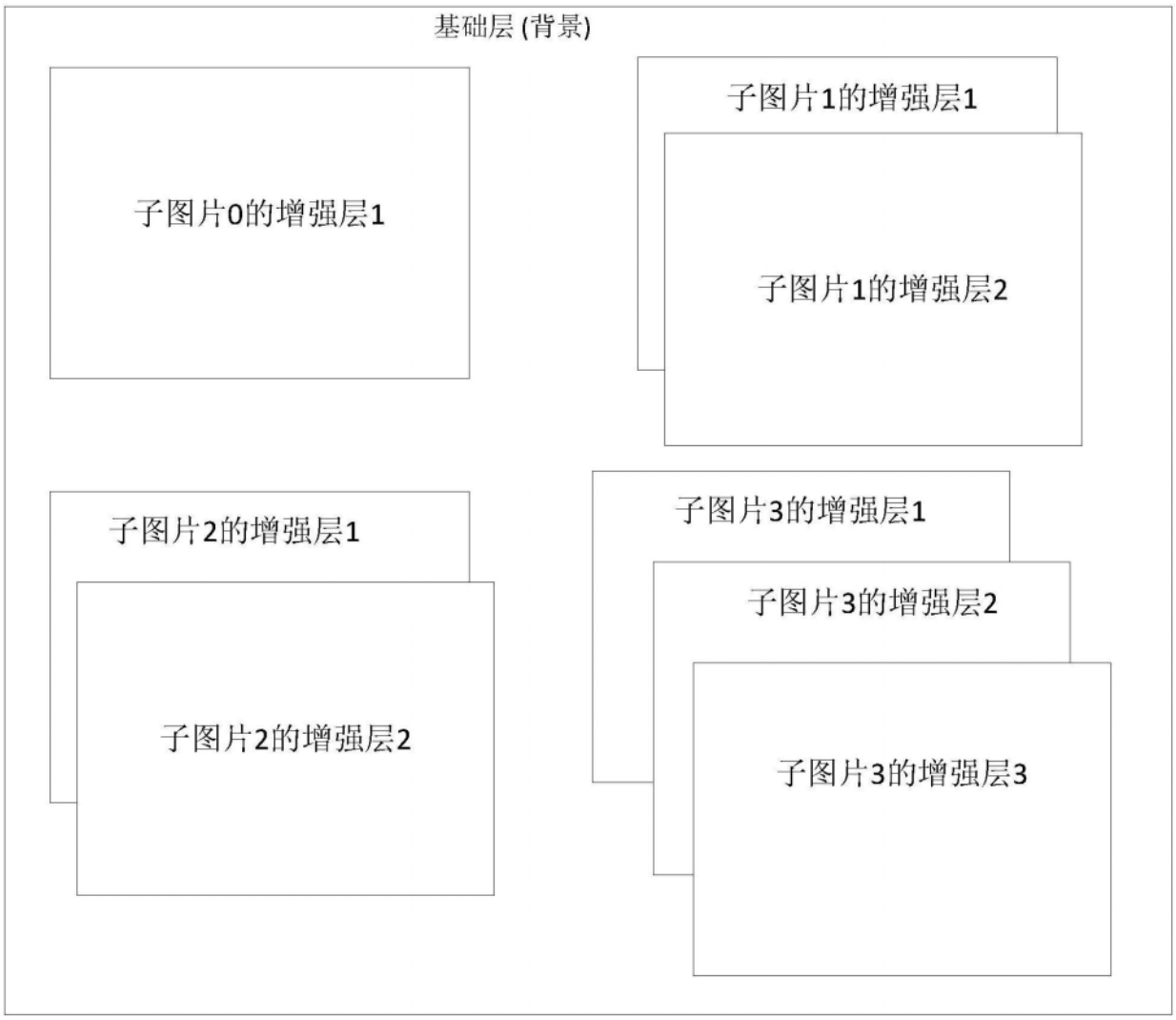

基础层(背景)
子图片0的增强层1
子图片1的增强层1
子图片1的增强层2
子图片2的增强层1
子图片2的增强层2
子图片3的增强层1
子图片3的增强层2
子图片3的增强层3

图10

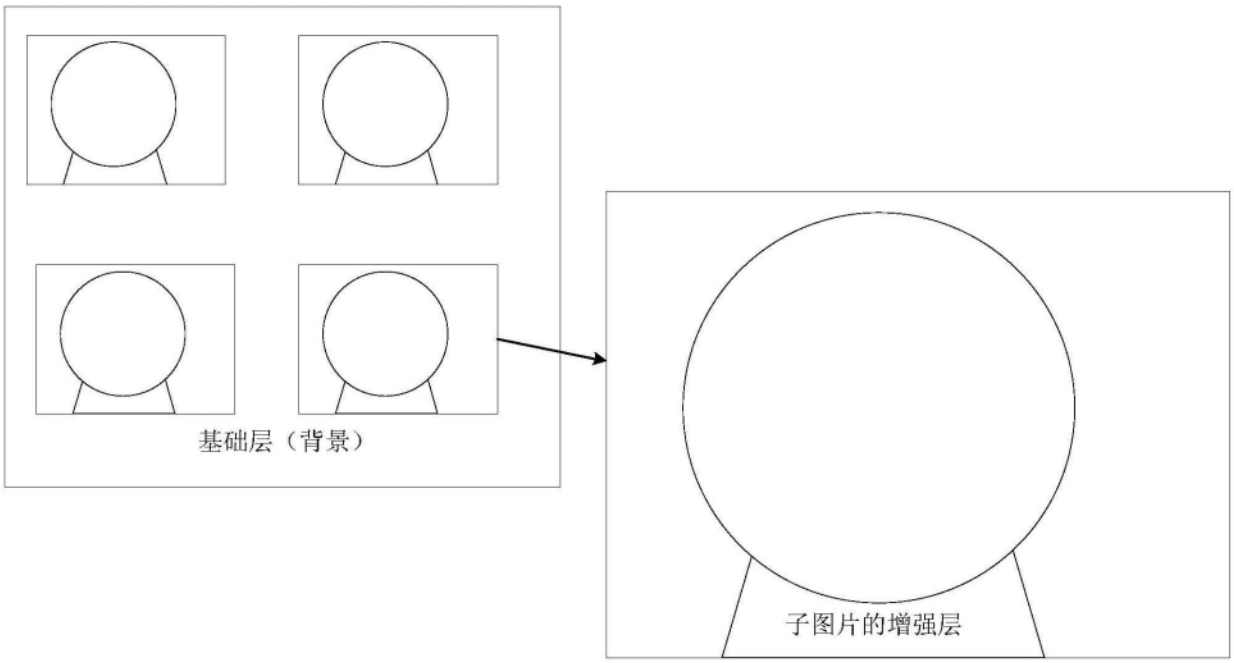

基础层（背景）
子图片的增强层

图11

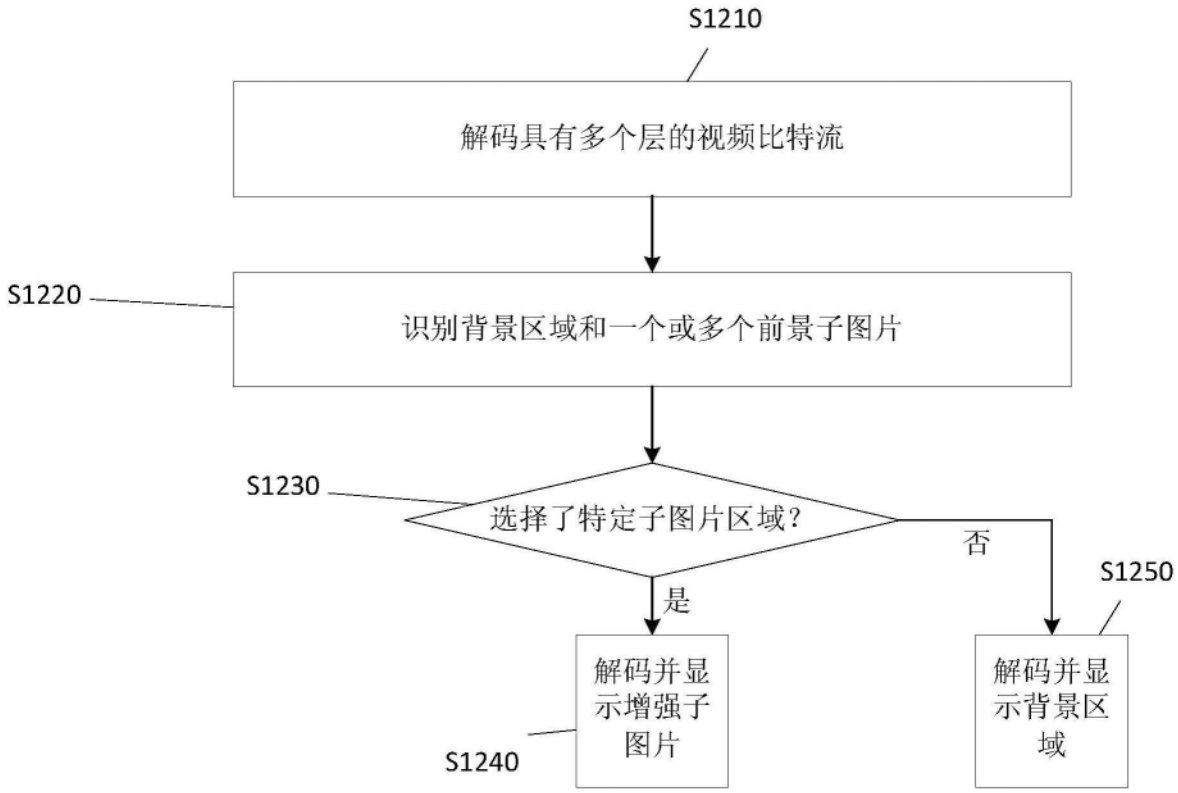

S1210
解码具有多个层的视频比特流
S1220
识别背景区域和一个或多个前景子图片
S1230
选择了特定子图片区域？
否
是
S1250
解码并显示增强子图片
S1240
解码并显示背景区域

图12

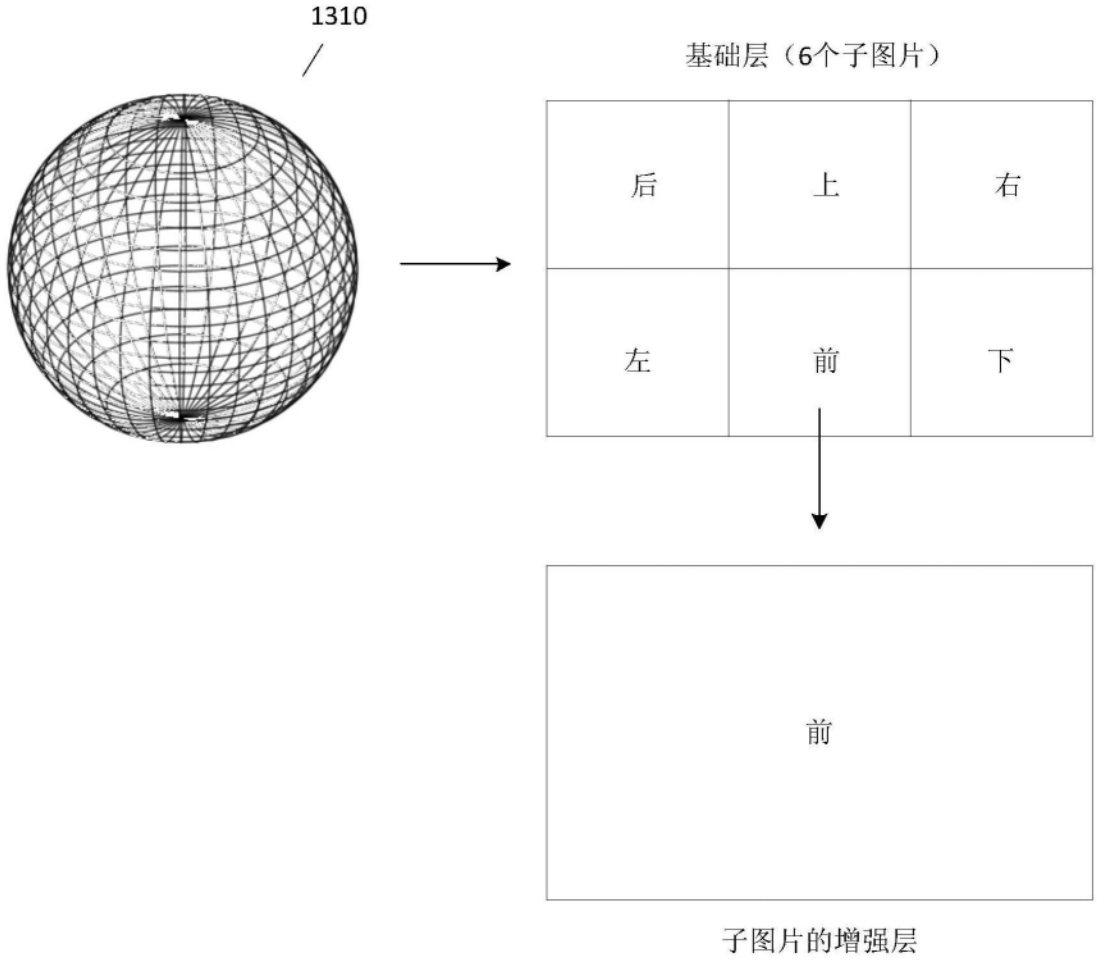


图13

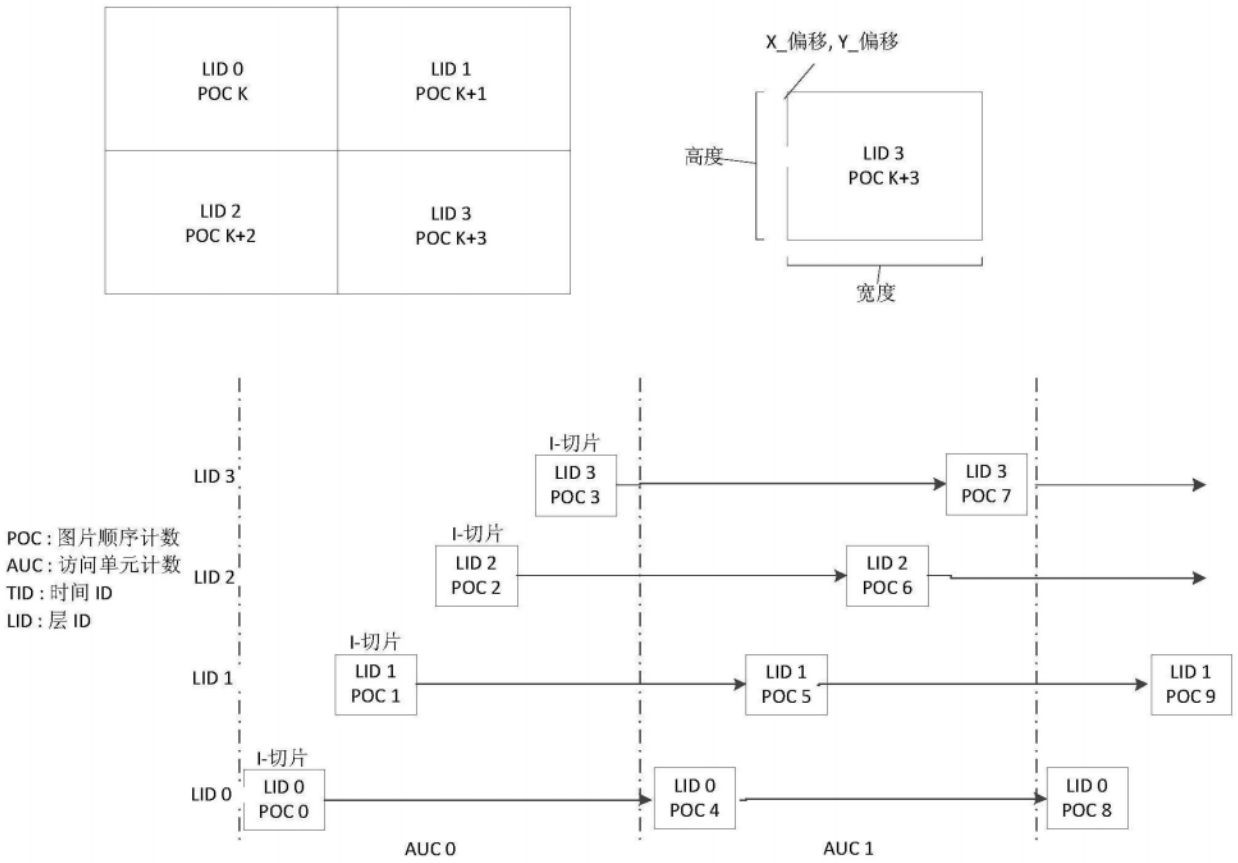

LID 0
POC K
LID 1
POC K+1
LID 2
POC K+2
LID 3
POC K+3
X_偏移, Y_偏移
高度
宽度
POC : 图片顺序计数
AUC : 访问单元计数
TID : 时间 ID
LID : 层 ID
LID 3
LID 2
LID 1
LID 0
I-切片
LID 3 POC 3
LID 3 POC 7
I-切片
LID 2 POC 2
LID 2 POC 6
I-切片
LID 1 POC 1
LID 1 POC 5
LID 1 POC 9
I-切片
LID 0 POC 0
LID 0 POC 4
LID 0 POC 8
AUC 0
AUC 1

图14

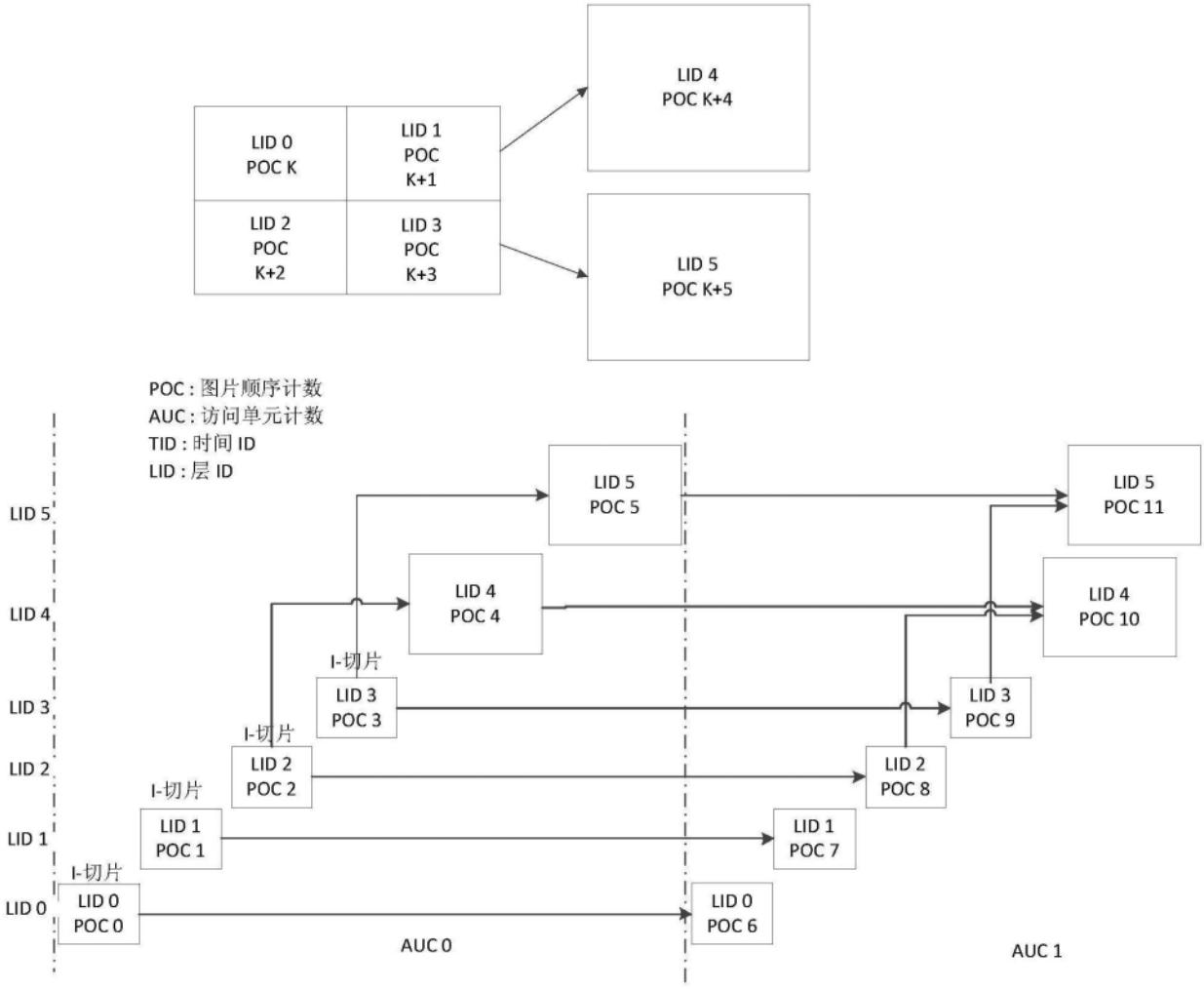


图15

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

图16A

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

图16B

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

图17

video_parameter_set_rbsp() {	描述符
…	
vps_max_layers_minus1	u(6)
num_output_layer_sets	ue(v)
num_profile_tier_level	ue(v)
for(i = 0; i < num_profile_tier_level; i++)	
profile_tier_level(vps_max_sub_layers_minus1)	
for(i = 0; i < num_output_layer_sets; i++)	
for(j = 0; j < NumLayersInIdList[i]; j++) {	
output_layer_flag[i][j]	u(1)
profile_tier_level_idx[i][j]	u(v)
}	
…	
}	

图18

video_parameter_set_rbsp() {	描述符
…	
vps_max_layers_minus1	u(6)
if(vps_max_layers_minus1 > 0) {	
num_output_layer_sets	ue(v)
num_profile_tier_level	ue(v)
}	
for(i = 0; i < num_profile_tier_level; i++)	
profile_tier_level(vps_max_sub_layers_minus1)	
for(i = 0; i < num_output_layer_sets; i++) {	
vps_output_layers_mode[i]	u(2)
vps_ptl_signal_flag[i]	u(1)
for(j = 0; j < NumLayersInIdList[i]; j++) {	
if(vps_output_layers_mode[i] == 2)	
output_layer_flag[i][j]	u(1)
if(vps_ptl_signal_flag[i])	
profile_tier_level_idx[i][j]	u(v)
}	
}	
…	
}	

图19

video_parameter_set_rbsp() {	描述符
...	
vps_max_layers_minus1	u(6)
if(vps_max_layers_minus1 > 0) {	
num_output_layer_sets	ue(v)
num_profile_tier_level	ue(v)
}	
max_subpics_minus1	u(8)
for(i = 0; i <= max_subpics_minus1; i++) {	
sub_pic_id[i]	u(8)
}	
for(i = 0; i < num_profile_tier_level; i++)	
profile_tier_level(vps_max_sub_layers_minus1)	
for(i = 0; i < num_output_layer_sets; i++) {	
vps_output_layers_mode[i]	u(2)
vps_ptl_signal_flag[i]	u(1)
for(j = 0; j < NumLayersInIdList[i]; j++) {	
num_output_subpic_layers[i][j]	ue(v)
for(k = 0; k < num_output_subpic_layer[i][j]; k++)	
sub_pic_id_layer[i][j] [k]	u(8)
if(vps_output_layers_mode[i] == 2)	
output_layer_flag[i][j]	u(1)
if(vps_ptl_signal_flag[i])	
profile_tier_level_idx[i][j]	u(v)
}	
}	
...	
}	

图20

pic_parameter_set_rbsp( ) {	描述符
pps_pic_parameter_set_id	ue(v)
pps_seq_parameter_set_id	u(4)
mixed_nalu_types_in_pic_flag	u(1)
pic_width_in_luma_samples	ue(v)
pic_height_in_luma_samples	ue(v)
pps_conformance_window_flag	u(1)
if(pps_conformance_window_flag) {	
pps_conf_win_left_offset	ue(v)
pps_conf_win_right_offset	ue(v)
pps_conf_win_top_offset	ue(v)
pps_conf_win_bottom_offset	ue(v)
}	
scaling_window_explicit_signalling_flag	u(1)
if(scaling_window_explicit_signalling_flag) {	
scaling_win_left_offset	ue(v)
scaling_win_right_offset	ue(v)
scaling_win_top_offset	ue(v)
scaling_win_bottom_offset	ue(v)
}	
output_flag_present_flag	u(1)
subpic_id_mapping_in_pps_flag	u(1)
if(subpic_id_mapping_in_pps_flag) {	
pps_num_subpics_minus1	ue(v)
pps_subpic_id_len_minus1	ue(v)
for(i = 0; i <= pps_num_subpic_minus1; i++)	
pps_subpic_id[i]	u(v)
}	
…	
}	

图21

seq_parameter_set_rbsp( ) {	描述符
…	
subpic_info_present_flag	u(1)
if(subpic_info_present_flag) {	
sps_num_subpics_minus2	ue(v)
sps_independent_subpics_flag	u(1)
for(i = 0; i <= sps_num_subpics_minus2 + 1; i++) {	
if(i > 0 && pic_width_max_in_luma_samples > CtbSizeY)	
subpic_ctu_top_left_x[i]	u(v)
if(i > 0 && pic_height_max_in_luma_samples > CtbSizeY) {	
subpic_ctu_top_left_y[i]	u(v)
if(i < sps_num_subpics_minus2 + 1 && pic_width_max_in_luma_samples > CtbSizeY)	
subpic_width_minus1[i]	u(v)
if(i < sps_num_subpics_minus2 + 1 && pic_height_max_in_luma_samples > CtbSizeY)	
subpic_height_minus1[i]	u(v)
if(!sps_independent_subpics_flag) {	
subpic_treated_as_pic_flag[i]	u(1)
loop_filter_across_subpic_enabled_flag[i]	u(1)
}	
}	
sps_subpic_id_len_minus1	ue(v)
subpic_id_mapping_explicitly_signalled_flag	u(1)
if(subpic_id_mapping_explicitly_signalled_flag) {	
subpic_id_mapping_in_sps_flag	u(1)
if(subpic_id_mapping_in_sps_flag)	
for(i = 0; i <= sps_num_subpics_minus2 + 1; i++)	
sps_subpic_id[i]	u(v)
}	
}	
…	
}	

图22

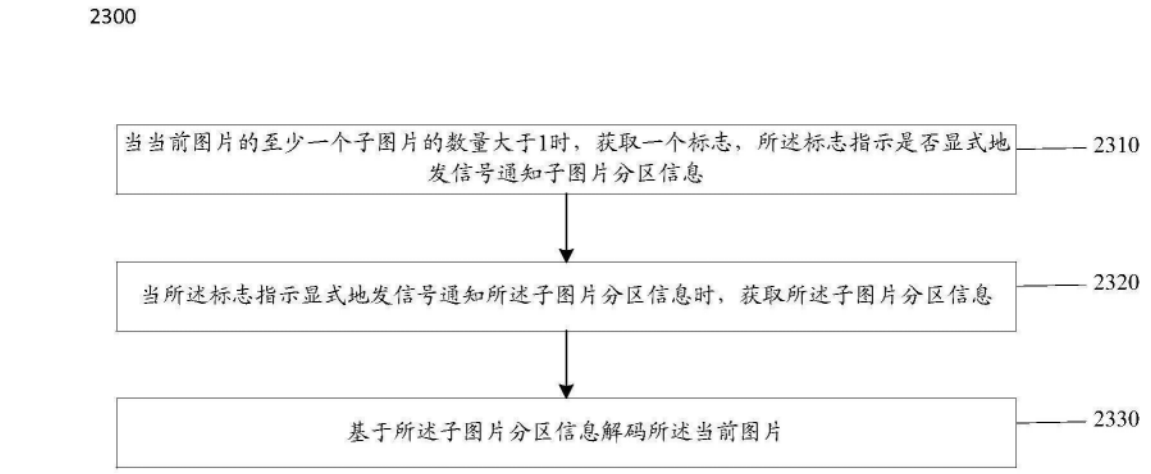

2300
当当前图片的至少一个子图片的数量大于1时，获取一个标志，所述标志指示是否显式地发信号通知子图片分区信息
2310
当所述标志指示显式地发信号通知所述子图片分区信息时，获取所述子图片分区信息
2320
基于所述子图片分区信息解码所述当前图片
2330

图23

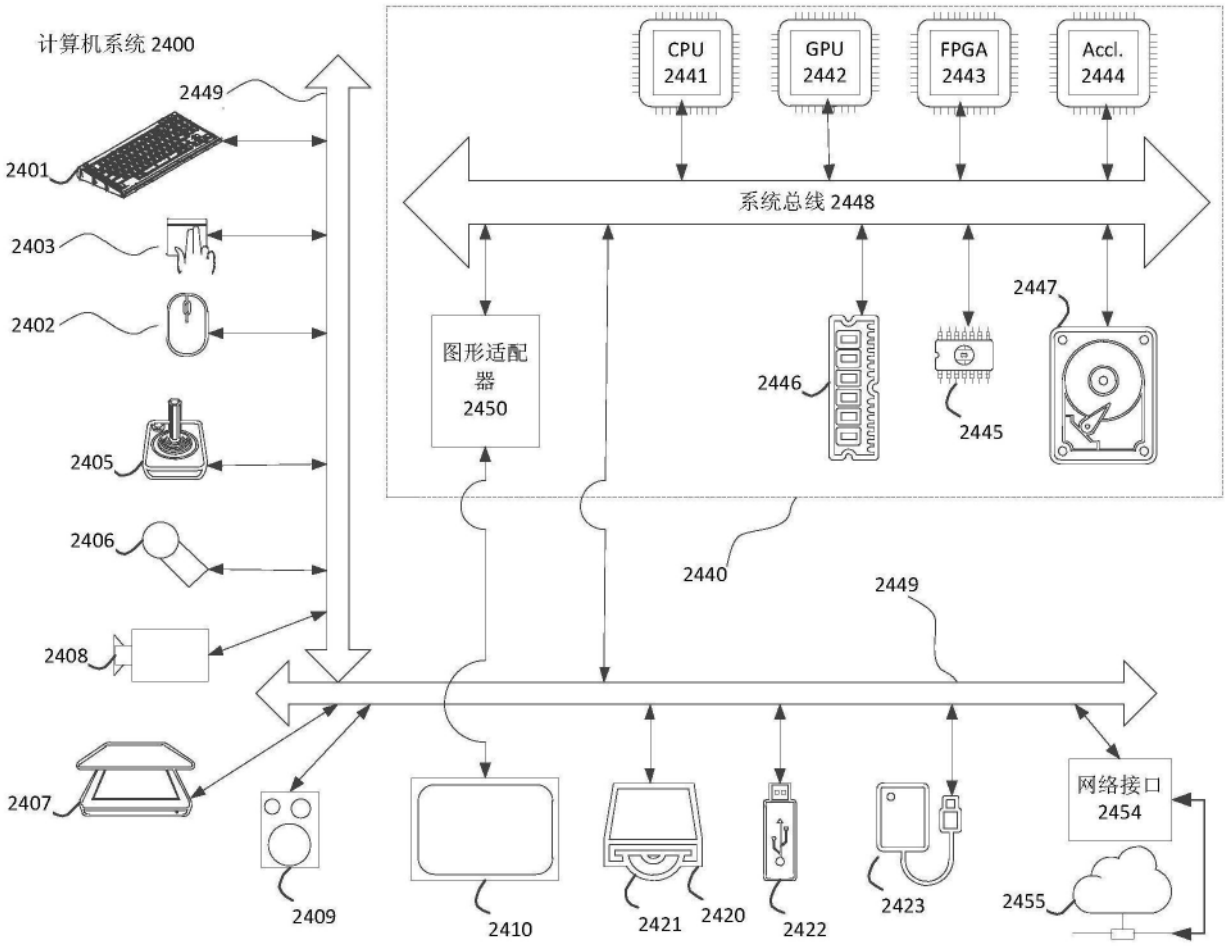

计算机系统 2400
2449
2401
2403
2402
2405
2406
2408
2407
2409
2410
2421
2420
2422
2423
2455
网络接口
2454
CPU
2441
GPU
2442
FPGA
2443
Accl.
2444
系统总线 2448
图形适配器
2450
2446
2445
2447
2440
2449

图24