(19)国家知识产权局

(12)发明专利

(10)授权公告号 CN 113966615 B
(45)授权公告日 2024.08.16

(21)申请号 202180003917.5
(22)申请日 2021.02.01
(65)同一申请的已公布的文献号
申请公布号 CN 113966615 A
(43)申请公布日 2022.01.21
(30)优先权数据
63/000,980 2020.03.27 US
17/097,636 2020.11.13 US
(85)PCT国际申请进入国家阶段日
2021.12.13
(86)PCT国际申请的申请数据
PCT/US2021/016019 2021.02.01
(87)PCT国际申请的公布数据
WO2021/194634 EN 2021.09.30

(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 徐文静 陈世华

(51)Int.Cl.
H04N 19/187(2006.01)

(56)对比文件
WO 2015194183 A1,2015.12.23
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》.2020,7.3-7.4节.

审查员 李若云

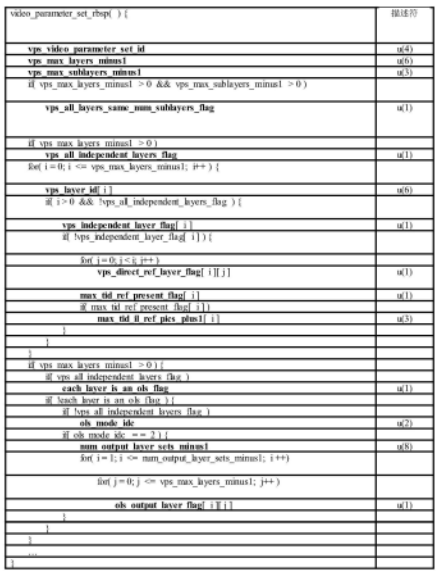
权利要求书2页 说明书34页 附图21页

(54)发明名称

对视频进行编解码的方法、设备以及存储介质

(57)摘要

一种对已编码视频序列进行解码的方法和系统包括:获得已编码视频序列,以及对该已编码视频序列进行解码。该已编码视频序列中的、视频编码层VCL网络抽象层NAL单元的时间子层标识符的值,被限制为小于或等于所述视频编码层VCL网络抽象层NAL单元所参考的视频参数集VPS中的vps_max_sublayers_minus1的值,所述vps_max_sublayers_minus1的值指示时间子层的最大数量,所述时间子层能够存在于参考所述视频参数集VPS的每个已编码视频序列中的层中。



video_parameter_set_rbsp( ) {	描述符
vps_video_parameter_set_id	u(4)
vps_max_layers_minus1	u(6)
vps_max_sublayers_minus1	u(3)
if(vps_max_layers_minus1 > 0 && vps_max_sublayers_minus1 > 0)	
vps_all_layers_same_num_sublayers_flag	u(1)
if(vps_max_layers_minus1 > 0)	
vps_all_independent_layers_flag	u(1)
for(i = 0; i <= vps_max_layers_minus1; i++) {	
vps_layer_id[i]	u(6)
if(i > 0 && !vps_all_independent_layers_flag) {	
vps_independent_layer_flag[i]	u(1)
if(!vps_independent_layer_flag[i]) {	
for(j = 0; j < i; j++)	
vps_direct_ref_layer_flag[i][j]	u(1)
max_tid_ref_present_flag[i]	u(1)
if(max_tid_ref_present_flag[i])	
max_tid_il_ref_pics_plus1[i]	u(3)
}	
}	
}	
if(vps_max_layers_minus1 > 0) {	
if(vps_all_independent_layers_flag)	
each_layer_is_an_ols_flag	u(1)
if(!each_layer_is_an_ols_flag) {	
if(!vps_all_independent_layers_flag)	
ols_mode_idc	u(2)
if(ols_mode_idc == 2) {	
num_output_layer_sets_minus1	u(8)
for(i = 1; i <= num_output_layer_sets_minus1; i++)	
for(j = 0; j <= vps_max_layers_minus1; j++)	
ols_output_layer_flag[i][j]	u(1)
}	
}	
}	
...	
}	

CN 113966615 B

1.一种对已编码视频序列进行解码的方法,其特征在于,所述方法包括:

获得所述已编码视频序列;

其中,所述已编码视频序列中的、视频编码层VCL网络抽象层NAL单元的时间子层标识符temporalId的值,被限制为小于或等于所述视频编码层VCL网络抽象层NAL单元所参考的视频参数集VPS中的vps_max_sublayers_minus1的值,所述vps_max_sublayers_minus1+1指示存在于参考所述视频参数集VPS的每个已编码视频序列中的层中的时间子层的最大数量;以及

使用所述时间子层标识符对所述已编码视频序列进行解码;

其中,max_tid_il_ref_pics_plus1[i]的值为零,指示第i层的非帧内随机存取点IRAP图片没有使用层间预测;

max_tid_il_ref_pics_plus1[i]的值大于零指示,为了对所述第i层的图片进行解码,不使用时间子层标识符大于max_tid_il_ref_pics_plus1[i]-1的图片作为层间参考图片ILRP;

当max_tid_il_ref_pics_plus1[i]不存在时,推断max_tid_il_ref_pics_plus1[i]的值等于vps_max_sublayers_minus1+1。

2.根据权利要求1所述的方法,其特征在于,max_tid_il_ref_pics_plus1[i]被限制为小于或等于vps_max_sublayers_minus1+1。

3.根据权利要求1所述的方法,其特征在于,逐子层输出层集合被限制为不为独立层导出。

4.根据权利要求3所述的方法,其特征在于,

第一变量NumSubLayersInLayerInOLS[i][j]指示第i个OLS中的第j层中的子层的数量,其中,第二变量OutputLayerIdInOls[i][j]指示所述第i个OLS中的第j个输出层的nuh_layer_id值,其中,第三变量LayerUsedAsOutputLayerFlag[k]指示第k层是否用作至少一个OLS中的输出层,并且其中,所述第一变量、所述第二变量和所述第三变量是从变量max_tid_il_ref_pics_plus1[i]的值推导出的。

5.根据权利要求3所述的方法,其特征在于,max_tid_il_ref_pics_plus1和layerIncludedInOlsFlag被限制为不为独立层导出。

6.一种产生已编码视频序列的方法,其特征在于,所述方法包括:

对视频数据进行编码,获得所述已编码视频序列;

其中,所述已编码视频序列中的、视频编码层VCL网络抽象层NAL单元的时间子层标识符temporalId的值,被限制为小于或等于所述视频编码层VCL网络抽象层NAL单元所参考的视频参数集VPS中的vps_max_sublayers_minus1的值,所述vps_max_sublayers_minus1+1指示存在于参考所述视频参数集VPS的每个已编码视频序列中的层中的时间子层的最大数量;

其中,max_tid_il_ref_pics_plus1[i]的值为零,指示第i层的非帧内随机存取点IRAP图片没有使用层间预测;

max_tid_il_ref_pics_plus1[i]的值大于零指示,为了对所述第i层的图片进行解码,不使用时间子层标识符大于max_tid_il_ref_pics_plus1[i]-1的图片作为层间参考图片ILRP;

当max_tid_il_ref_pics_plus1[i]不存在时,推断max_tid_il_ref_pics_plus1[i]的值等于vps_max_sublayers_minus1+1。

7.一种对已编码视频序列进行解码的设备,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且如所述程序代码所指示地操作,以实现:

获得所述已编码视频序列;

其中,所述已编码视频序列中的、视频编码层VCL网络抽象层NAL单元的时间子层标识符temporalId的值,被限制为小于或等于所述视频编码层VCL网络抽象层NAL单元所参考的视频参数集VPS中的vps_max_sublayers_minus1的值,所述vps_max_sublayers_minus1+1指示存在于参考所述视频参数集VPS的每个已编码视频序列中的层中的时间子层的最大数量;以及

使用所述时间子层标识符对所述已编码视频序列进行解码;

其中,max_tid_il_ref_pics_plus1[i]的值为零,指示第i层的非帧内随机存取点IRAP图片没有使用层间预测;

max_tid_il_ref_pics_plus1[i]的值大于零指示,为了对所述第i层的图片进行解码,不使用时间子层标识符大于max_tid_il_ref_pics_plus1[i]-1的图片作为层间参考图片ILRP;

当max_tid_il_ref_pics_plus1[i]不存在时,推断max_tid_il_ref_pics_plus1[i]的值等于vps_max_sublayers_minus1+1。

8.根据权利要求7所述的设备,其特征在于,max_tid_il_ref_pics_plus1[i]被限制为小于或等于vps_max_sublayers_minus1+1。

9.根据权利要求7所述的设备,其特征在于,逐子层输出层集合被限制为不为独立层导出。

10.根据权利要求9所述的设备,其特征在于,max_tid_il_ref_pics_plus1和layerIncludedInOlsFlag被限制为不为独立层导出。

11.一种产生已编码视频序列的编码设备,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且如所述程序代码所指示地操作,以实现如权利要求1-6任一项所述的方法。

12.一种计算机设备,其特征在于,其特征在于,包括:处理器和存储器;所述存储器存储有计算机代码,所述计算机代码被所述处理器执行时,使得所述处理器执行如权利要求1～6中任一项所述的方法。

13.一种存储指令的非易失性计算机可读介质,其特征在于,所述指令包括:一个或多个指令,所述一个或多个指令由一个或多个处理器执行,以实现如权利要求1～6中任一项所述的方法。

对视频进行编解码的方法、设备以及存储介质

[0001] 交叉引用

[0002] 本申请根据35U.S.C.§119要求于2020年3月27日向美国专利商标局提交的美国临时申请第63/000,980号以及于2020年11月13日向其提交的美国申请第17/097,636号的优先权,其公开内容通过引用结合在本申请中。

技术领域

[0003] 所公开的主题涉及视频编码和解码,并且更具体地,涉及具有多个层的已编码视频码流中的输出层推导。

背景技术

[0004] 使用具有运动补偿的图片间预测的视频编码和解码已经被熟知了几十年。未压缩数字视频可以由一系列图片组成,每个图片具有例如1920×1080个亮度样本和相关联的色度样本的空间维度。系列图片可以具有固定或可变的图片速率(也被非正式地称为帧速率),例如每秒60个图片或60Hz。未压缩视频具有显著的比特率要求。例如,每采样8比特的1080p60 4:2:0视频(60Hz帧速率的1920×1080亮度采样分辨率)需要接近1.5Gbit/s带宽。一个小时的这种视频需要超过600千兆字节的存储空间。

[0005] 视频编码和解码的一个目的可以是通过压缩来减少输入视频信号中的冗余。压缩可以帮助减少前述带宽或存储空间要求,在一些情况下减少两个数量级或更多。可以采用无损压缩和有损压缩,以及它们的组合。无损压缩是指可以从压缩的原始信号重建原始信号的精确副本的技术。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。在视频的情况下,广泛采用有损压缩。容许的失真量取决于应用;例如,某些消费流式应用的用户可以容忍比电视分发应用的用户更高的失真。可实现的压缩比可以反映出:较高的允许/容许失真可以产生较高的压缩比。

[0006] 视频编码器和解码器可以利用来自若干广泛类别的技术,包括例如运动补偿、变换、量化及熵编码,下文将介绍其中的一些技术。

[0007] 历史上,视频编码器和解码器倾向于对给定图片大小进行操作,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 coded video sequence,)、图片组(GOP,Group of Picture)或类似的多图片时间帧而定义并且保持恒定的。例如,在MPEG-2中,已知系统设计根据例如场景活动的因素来改变水平分辨率(并且因此改变图片大小),但是仅在I图片处改变水平分辨率,因此通常用于GOP。在CVS内使用不同分辨率的参考图片的重采样例如从ITU-T Rec.H.263附录PP已熟知。然而,这里的图片大小没有改变,仅参考图片被重采样,潜在地导致仅部分图片画布被使用(在下采样的情况下),或者仅部分场景被采集(在上采样的情况下)。进一步地,H.263附录Q允许以因子2(在每个维度中)向上或向下重采样单个宏块。同样,图片大小保持相同。宏块的大小在H.263中是固定的,并且因此不需要用信号通知。

[0008] 预测图片中图片大小的变化在现代视频编解码中变得更加主流。例如,VP9允许参考图片重采样和整个图像的分辨率的改变。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)”联合视频小组文件JVET-M0135-v1,2019年1月9-19日,其全文并入本文中)允许将整个参考图片重采样到不同的更高或更低分辨率。在该文献中,建议将不同的候选分辨率编码在序列参数集中,并且由图片参数集中的每图片语法元素来参考。

发明内容

[0009] 本公开为用于视频码流中的自适应图片大小的信令。

[0010] 一种由至少一个处理器执行的对已编码视频序列进行解码的方法,包括:获得所述已编码视频序列;以及对所述已编码视频序列进行解码,其中,所述已编码视频序列中的、视频编码层VCL网络抽象层NAL单元的时间子层标识符的值,被限制为小于或等于所述视频编码层VCL网络抽象层NAL单元所参考的视频参数集VPS中的vps_max_sublayers_minus1的值,所述vps_max_sublayers_minus1的值指示时间子层的最大数量,所述时间子层能够存在于参考所述视频参数集VPS的每个已编码视频序列中的层中。

[0011] 在一些实施例中,max_tid_il_ref_pics_plus1[i]的值为零,指示第i层的非帧内随机存取点IRAP图片没有使用层间预测。

[0012] 在一些实施例中,max_tid_il_ref_pics_plus1[i]的值大于零指示,为了对所述第i层的图片进行解码,不使用时间子层标识符大于max_tid_il_ref_pics_plus1[i]-1的图片作为层间参考图片ILRP。

[0013] 在一些实施例中,当不存在时,推断max_tid_il_ref_pics_plus1[i]的值等于vps_max_sublayers_minus1+1。

[0014] 在一些实施例中,max_tid_il_ref_pics_plus1[i]被限制为小于或等于vps_max_sublayers_minus1+1。

[0015] 在一些实施例中,逐子层输出层集合被限制为不为独立层导出。

[0016] 在一些实施例中,第一变量NumSubLayersInLayerInOLS[i][j]指示第i个OLS中的第j层中的子层的数量,其中,第二变量OutputLayerIdInOls[i][j]指示所述第i个OLS中的第j个输出层的nuh_layer_id值,其中,第三变量LayerUsedAsOutputLayerFlag[k]指示第k层是否用作至少一个OLS中的输出层,并且其中,所述第一变量、所述第二变量和所述第三变量是从变量max_tid_il_ref_pics_plus1[i]的值推导出的。

[0017] 在一些实施例中,max_tid_il_ref_pics_plus1和layerIncludedInOlsFlag被限制为不为独立层导出。

[0018] 一种对已编码视频序列进行解码的示例性设备,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且如所述程序代码所指示地操作,所述程序代码包括:获得代码,被配置为使所述至少一个处理器获得所述已编码视频序列;以及解码代码,被配置为使所述至少一个处理器对所述已编码视频序列进行解码,其中,所述已编码视频序列中的、视频编码层VCL网络抽象层NAL单元的时间子层标识符的值被限制为小于或等于所述视频编码层VCL网络抽象层NAL单元所参考的视频参数集VPS中的vps_max_sublayers_minus1的值,所述vps_max_sublayers_minus1的值指示时间

子层的最大数量,所述时间子层能够存在于参考所述视频参数集VPS的每个已编码视频序列中的层中。

[0019] 一种示例性非易失性计算机可读介质,存储有指令,所述指令包括:一个或多个指令,所述一个或多个指令当由用于对视频序列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:获得已编码视频序列;以及对所述已编码视频序列进行解码,其中,所述已编码视频序列中的、视频编码层VCL网络抽象层NAL单元的时间子层标识符的值被限制为小于或等于所述视频编码层VCL网络抽象层NAL单元所参考的视频参数集VPS中的vps_max_sublayers_minus1的值,所述vps_max_sublayers_minus1的值指示时间子层的最大数量,所述时间子层能够存在于参考所述视频参数集VPS的每个已编码视频序列中的层中。

附图说明

[0020] 通过以下详细描述和附图,所公开的主题的其它特征、性质及各种优点将更加明显,其中:

[0021] 图1是根据实施例的通信系统的简化框图的示意图;

[0022] 图2是根据实施例的通信系统的简化框图的示意图;

[0023] 图3是根据实施例的解码器的简化框图的示意图;

[0024] 图4是根据实施例的解码器的简化框图的示意图;

[0025] 图5是根据现有技术或实施例的用于发信号通知ARC参数的选项的示意图;

[0026] 图6是根据实施例的语法表的示例;

[0027] 图7是根据实施例的计算机设备的示意图;

[0028] 图8是具有自适应分辨率变化的可伸缩性的预测结构的示例;

[0029] 图9是根据实施例的与发表的示例;

[0030] 图10是解析和解码每访问单元的poc周期和访问单元计数值的简化框图的示意图;

[0031] 图11是包括多层子图片的视频码流结构的示意图;

[0032] 图12是具有增强分辨率的所选子图片的显示的示意图;

[0033] 图13是包括多层子图片的视频码流的解码和显示过程的框图;

[0034] 图14是具有子图片的增强层的360视频显示的示意图;

[0035] 图15是子图片的布局信息、其对应层和图片预测结构的示例;

[0036] 图16是具有局部区域的空间可缩放性模态的子图片的布局信息及其对应的层和画面预测结构的示例;

[0037] 图17是子图片布局信息的语法表的示例;

[0038] 图18是子图片的布局信息的SEI消息的语法表的示例;

[0039] 图19是用于指示输出层和每个输出层集的配置文件/层/级别的语法表的示例;

[0040] 图20是用于指示每个输出层集的输出层模式的语法表的示例;

[0041] 图21是针对每个输出层集指示每个层的当前子图片的语法表的示例;

[0042] 图22是使用子层级指示来指示输出层集的语法表的示例。

具体实施方式

[0043] 当图片被编码成由具有不同质量的多个层组成的码流时,该码流可以具有用于指示哪些层可以在解码器侧输出的语法元素。要输出的层的集合被定义为输出层集合。在支持多个层和可缩放性的最新视频编解码器中,在视频参数集中用信号通知一个或多个输出层集合。那些用于指示输出层集合以及它们的依赖性、配置文件(profile)/层(/tier)/级别(level)和假定解码器参考模型参数的语法元素需要在参数集中被有效地用信号通知。

[0044] 图1是根据本申请公开的实施例的通信系统(100)的简化框图。通信系统(100)包括至少两个终端装置(110-120),所述终端装置可通过例如网络(150)彼此通信。对于数据的单向传输,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)传输到第二端装置(120)。第二终端装置(120)可从网络(150)接收其他终端设备的已编码视频数据,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。

[0045] 图1示出了第二对终端设备(130-140)来用于支持已编码视频数据的双向传输,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,终端装置(130-140)中的每个终端装置可对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。终端装置(130-140)中的每个终端装置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。

[0046] 在图1中,第一终端装置(110)、第二终端装置(120)、第三终端装置(130)和第四终端装置(140)可为服务器、个人计算机和智能电话,但本申请公开的原理可不限于此。本申请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)表示在第一终端装置(110)、第二终端装置(120)、第三终端装置(130)和第四终端装置(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(连线的)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。该网络可包括电信网络、局域网、广域网和/或互联网。出于本申请的目的,除非在下文中有所解释,否则网络(150)的架构和拓扑对于本申请公开的操作来说可能是无关紧要的。

[0047] 作为实施例,图2示出视频编码器和解码器在流式传输环境中的放置方式。本申请所公开主题可同等地适用于其它支持视频的应用,包括例如视频会议、数字TV、在包括CD、DVD、存储棒等的数字介质上存储压缩视频等等。

[0048] 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数码相机等视频源(201),所述视频源创建未压缩的视频样本流(202)。相较于已编码的视频码流,被描绘为粗线以强调高数据量的样本流(302),可以由耦接到相机(201)的编码器(203)来处理。视频编码器(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,被描绘为细线以强调较低数据量的已编码的视频码流(204),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)和(208),可访问流式传输服务器(205)以检索已编码的视频码流(204)的副本(207)和副本(209)。客户端(206)可包括视频解码器(210),视频解码器(210)对已编码的视频数据的传入副本(207)进行解码,且产生可在显示器(212)或另一呈现装置(未描绘)上呈现的输出视频样本流(211)。在一些流式传输系统中,可根据某些视频编码/压缩标准对已编码的视频码流(204)、视频码流(207)和视频码流(209)进行编码。该些标准的示例包括ITU-T H.265。

在实施例中,正在开发的视频编码标准非正式地称为下一代视频编码(Versatile Video Coding,VVC),本申请可用于VVC标准的上下文中。

[0049] 图3是根据本公开的实施例的视频解码器(210)的功能框图。

[0050] 接收器(310)可接收将由解码器(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。而当接收器(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,为了在互联网等业务分组网络上使用,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具有自适应性大小。

[0051] 视频解码器(210)可包括解析器(320)以根据熵编码视频序列来重建符号(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,以及用以控制显示装置(例如,显示屏(212))等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2中所示。用于显示装置的控制信息可以是辅助增强信息(Supplemental Enhancement Information,SEI消息)或视频可用性信息(Video Usability Information,VUI)的参数集片段(未标示)。解析器(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码可根据视频编码技术或标准进行,且可遵循本领域技术人员熟知的原理,包括可变长度编码、霍夫曼编码(Huffman coding)、具有或不具有上下文灵敏度的算术编码等等。解析器(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、条带、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。

[0052] 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)。

[0053] 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。

[0054] 除已经提及的功能块以外,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例中,这些单元中的许多单元彼此紧密交互并且可以彼此集成。然而,出于描述所公开主题的目的,概念上细分成下文的功能单元是适当的。

[0055] 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。

[0056] 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(356)提取的已重建信息生成大小和形状与正在重建的块相同的周围块。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。

[0057] 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/逆变换单元3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。

[0058] 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码流中的参数,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,在其他实施例中,视频压缩技术还可响应于在解码已编码图片或已编码视频序列的先前(按解码次序)部分期间获得的元信息,以及响应于先前重建且经过环路滤波的样本值。

[0059] 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。

[0060] 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,则当前参考图片缓冲器(358)可变为参考图片缓冲器(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。

[0061] 视频解码器320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。编码的视频序列可以符合由正在使用的视频压缩技术或标准规定的语法,在所述编码的视频序列遵守视频压缩技术或标准的语法的意义上,如在视频压缩技术文档或标准中规定的,特别是在其中的配置文件所规定的。对于合规性,还要求已编码视频序列的复杂度处于视频压缩技术或标准的层级所限定的范围内。在一些情况下,层级限制最大图片大小、最大帧率、最大重建取样率(以例如每秒兆(mega)个样本为单位进行测量)、最大参考图片大小等。在一些情况下,由层级设定的限制可通过假想参考解码器(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。

[0062] 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320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。

[0063] 图4是根据本公开的实施例的视频编码器(203)的功能框图。

[0064] 编码器(203)可以从视频源(201)(不是编码器的一部分)接收视频样本,该视频源可以采集将要由编码器(203)编码的视频图像。

[0065] 视频源(201)可提供将由编码器(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 CrCb4:2:0、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(203)可以是采集本地图像信息作为视频序列的相机。可将视频数据提供为多个单独的图片,当按顺序观看时,这些图片被赋予运动。图片自身可构建为空间像素阵列,其中取决于所用的取样结构、色彩空间等,每个像素可包括一个或多个样本。所属领域的技术人员可以很容易理解像素与样本之间的关系。下文侧重于描述样本。

[0066] 根据实施例,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(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的识别控制器(450)的其他功能,因为这些功能涉及针对某一系统设计优化的视频编码器(203)。

[0067] 一些视频编码器(503)在本领域技术人员容易识别为“编码环路”中操作。作为简单的描述,在实施例中,编码环路可由编码器(430)(这里成为源编码器)的编码部分(负责基于待编码的输入图片和参考图片创建符号)和嵌入于编码器(203)中的(本地)解码器(433)组成。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(因为在本申请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,因此参考图片缓冲器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)对于本领域技术人员也是熟知的。

[0068] “本地”解码器(433)的操作可与例如已在上文结合图3详细的“远程”解码器相同。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/解码为已编码视频序列时,包括信道(312)、接收器(310)、缓冲器(315)和解析器(320)在内的视频解码器(210)的熵解码部分,可能无法完全在本地解码器(433)中实施。

[0069] 此时可以观察到,除存在于解码器中的解析/熵解码之外的任何解码器技术,也必定以基本上相同的功能形式存在于对应的编码器中。出于此原因,本申请侧重于解码器操作。可简化编码器技术的描述,因为编码器技术与全面地描述的解码器技术互逆。仅在某些区域中需要更详细的描述,并且在下文提供。

[0070] 作为操作的部分,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。

[0071] 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高速缓存(434)中。以此方式,视频编码器(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)。

[0072] 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。

[0073] 控制器(450)可管理视频编码器(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。

[0074] 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。熵编码器根据本领域技术人员熟知的,例如霍夫曼编码、可变长度编码、算术编码等技术对各种功能单元生成的符号进行无损压缩,从而将所述符号转换成已编码视频序列。

[0075] 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)可将来自视频编码器(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)。

[0076] 控制器(450)可管理编码器(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帧类型:

[0077] 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。所属领域的技术人员了解I图片的变体及其相应的应用和特征。

[0078] 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。

[0079] 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。

[0080] 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。

[0081] 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。

[0082] 在实施例中,传输器(440)可在传输已编码的视频时传输附加数据。视频编码器(430)可将此类数据作为已编码视频序列的一部分。附加数据可包括时间/空间/SNR增强层、冗余图片和切片等其它形式的冗余数据、SEI消息、VUI参数集片段等。

[0083] 在更详细地描述所公开的主题的某些方面之前,需要介绍将在本描述的剩余部分中提及的一些术语。

[0084] 此后,子图片在一些情况下是指样本、块、宏块、编码单元或相似实体的矩形布置,这些样本、块、宏块、编码单元或类似实体在语义上进行分组,并且可以以变化的分辨率进行独立地编码。一个或多个子图片可以用于图片。一个或多个已编码子图片可以形成已编码图片。可以将一个或多个子图片组合成图片,并且可以从图片中提取一个或多个子图片。在某些环境中,可以在压缩域中组合一个或多个已编码子图片,而不将样本级别代码转换成已编码图片,并且在相同或某些其它情况下,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。

[0085] 此后的自适应分辨率改变(ARC,Adaptive Resolution Change)是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。此后的ARC参数是指执行自适应分辨率改变所需的控制信息,其可以包括(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。

[0086] 以上描述集中于编码和解码单个语义上独立的已编码视频图片。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附加复杂度之前,可以描述用于发信号通知ARC参数的选项。

[0087] 参考图5,示出了用于发信号通知ARC参数的若干新颖选项。如选项中的每一个所指出的,从编解码效率、复杂度和体系结构的观点来看,它们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,或从现有技术中已知的选项,用于发信号通知ARC参数。这些选项可以不是互相排斥的,并且可以根据应用需要、所涉及的标准技术或编码器的选择而互换。

[0088] ARC参数的类别可以包括:

[0089] -在X维度和Y维度上分离或组合的上采样因子和/或下采样因子。

[0090] -上采样因子和/或下采样因子,其加上时间维度,指示对于给定数量的图片进行恒速放大/缩小。

[0091] -以上两者中的任何一者都可能涉及对可能指向包含因子的表的一个或多个可能

较短的语法元素进行编码。

[0092] -输入图片、输出图片、参考图片、已编码图片的在X维度或Y维度上的分辨率,其以样本、块、宏块、CU或任何其它合适的粒度为单位被组合或分开。如果存在多于一个分辨率(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,可以从一组值中推断出另一组值。这可以例如通过使用标志来门控。更详细的示例见下文。

[0093] -“翘曲(Warping)”坐标类似于H.263附录P中使用的那些坐标,同样是以上所描述的合适粒度。H.263附录P定义了一种对这种翘曲坐标进行编码的有效方式,但是也可以设想出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且偏移某个值,从而允许在最大图片大小的边界之外的“翘曲”。

[0094] -上采样滤波参数和/或下采样滤波参数。在最容易的情况下,可能仅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,并且这可能需要用信号通知滤波器参数。可以通过可能的滤波器设计列表中的索引来选择这样的参数,可以完全指定该滤波器(例如通过滤波器系数列表,使用适当的熵编码技术),可以通过上采样比率和/或下采样比率来隐式地选择该滤波器,进而,根据上述任一种机制,轮流发信号通知根据上采样率,还是下采样率来隐含地选择滤波器,等等。

[0095] 此后,该描述假设通过码字指示的上采样因子和/或下采样因子(在X维度和Y维度中使用的相同因子)的有限集合的编码。该码字,可以例如通过使用诸如H.264和H.265的视频编解码规范中的某些语法元素通用的Ext-Golomb代码,有利地进行可变长度编码。值到上采样因子和/或下采样因子的一种合适的映射可以例如根据下面的表1。

[0096] 表1

[0097]

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%)

[0098] 可以根据应用的需要以及视频压缩技术或标准中可用的向上缩放机制和向下缩放机制的能力来设计许多类似的映射。该表可以扩展到更多的值。值还可以由除Ext-Golomb代码之外的熵编码机制(例如,使用二进制编码)来表示。当重采样因子在视频处理引擎(最重要的是编码器和解码器)本身以外(例如由MANE)感兴趣时,这可能具有某些优点。应注意,对于不需要分辨率改变的(大概)最常见的情况,可以选择短的Ext-Golomb代码;在上表中,仅单个比特。对于最常见的情况,这可以比使用二进制代码具有编解码效率优势。

[0099] 表中条目的数量以及它们的语义可以是完全地或部分地可配置的。例如,表格的基本纲要可以在诸如序列参数集或解码器参数集的“高”参数集中传达。可替代地或另外地,可以在视频编解码技术或标准中定义一个或多个这种表,并且可以通过例如解码器参数集或序列参数集来选择一个或多个这种表。

[0100] 此后,我们描述了如何在视频编解码技术或标准语法中包括如以上所描述的编码的上采样因子和/或下采样因子(ARC信息)。类似的考虑可以应用于控制上采样滤波器和/或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,参见以下讨论。

[0101] H.263附录P将四个翘曲坐标形式的ARC信息502包括在图片头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)期望ARC信息的频繁改变时,这可能是明智的设计选择。然而,当使用H.263型信令时的开销可能相当高,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变性质。

[0102] 上面引用的JVCET-M135-v1包括位于图片参数集(504)中的ARC参考信息(505)(索引),其用于对包括目标分辨率的表(506)进行索引,所述目标分辨率依次位于序列参数集(507)中。根据作者的口头陈述,通过在能力交换期间将SPS用作互操作性协商点,可以证明将可能分辨率放置在序列参数集(507)中的表(506)中是合理的。通过参考适当的图片参数集(504),分辨率可以在由表格(506)中的值所设置的限制内逐张图片地改变。

[0103] 仍然参考图5,可以存在以下附加选项来在视频码流中传送ARC信息。这些选项中的每一个都具有优于以上描述的现有技术的某些优点。选项可以同时存在于同一视频编解码技术或标准中。

[0104] 在实施例中,诸如重采样(缩放)因子的ARC信息(509)可以存在于条带(slice)头、GOB头、图块(tile)头或图块组头(此后为图块组头)(508)中。这对于较小的ARC信息是足够的,例如单个可变长度ue(v)或几个比特的固定长度码字,例如如上所示。在图块组头中直接包含有ARC信息可以具有附加的优点:ARC信息可应用于由例如图块组表示的子图片,而不是整个图片。另见下文。另外,即使视频压缩技术或标准仅设想整个图片自适应分辨率改变(例如,与基于图块组的自适应分辨率改变形成对比),从容错的角度来看,将ARC信息放入图块组头中相对于将其放入H.263型图片头中具有一定优势。

[0105] 在相同或另一实施例中,ARC信息(512)本身可以存在于适当的参数集(511)中,例如图片参数集、头参数集、图块参数集、自适应参数集等(所描绘的自适应参数集)。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,则图片参数集或等同物可能是合适的。

[0106] 在相同或另一实施例中,ARC参考信息(513)可以存在于图块组头(514)或类似数据结构中。该参考信息(513)可以指参数集(516)中可用的ARC信息(515)的子集,所述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。

[0107] 如在JVET-MO135-v1中使用的,来自图块组头、PPS、SPS的间接隐含激活的附加级别看起来是不必要的,因为图片参数集(正如序列参数集)可以用于(并且在诸如RFC3984的某些标准中具有)能力协商或通告。然而,如果ARC信息应该适用于例如由图块组表示的子图片,则具有限于图块组的激活范围的参数集(例如自适应参数集或头参数集)也可以是更好的选择。此外,如果ARC信息的大小大于可忽略的大小(例如,包含诸如多个滤波器系数的滤波器控制信息),则从编解码效率的观点来看,参数可能是比直接使用头(508)更好的选择,因为这些设置可以通过参考相同的参数集,而由未来的图片或子图片重复使用。

[0108] 当使用具有跨越多个图片的范围的序列参数集或另一更高参数集时,可以应用某

些考虑:

[0109] 1、用于存储ARC信息表(516)的参数集在某些情况下可以是序列参数集,但在一些情况下有利地可以是解码器参数集。解码器参数集可以具有多个CVS的激活范围,即已编码视频码流,即从会话开始直到会话拆除的所有已编码视频比特。这样的范围可能更合适,因为可能的ARC因子可以是解码器特征,可能在硬件中实现,并且硬件特征倾向于不随任何CVS(其在至少一些娱乐系统中是图像组,长度为一秒或更短)而改变。也就是说,将表放入序列参数集中明确地包括在本文描述的放置选项中,特别是结合下面的点2。

[0110] 2、ARC参考信息(513)可以有利地直接放置到图片/条带图块/GOB/图块组头(下文中的图块组头)(514)中,而非放置到JVCET-M0135-v1中的图片参数集中。原因如下:当编码器想要改变图片参数集中的单个值时,例如ARC参考信息,则它必须创建新的PPS并且参考该新的PPS。假设仅ARC参考信息改变,但PPS中的其它信息,例如量化矩阵信息保持不变。这样的信息可以具有相当大的大小,并且需要重新传输以使新的PPS完整。由于ARC参考信息可以是单个码字,例如表(513)中的索引,并且这将是改变的唯一值,因此重新传输例如所有量化矩阵信息将是麻烦并且浪费的。到目前为止,从编解码效率的角度来看,避免通过PPS进行间接访问会更好,如JVET-M0135-v1中所提议的。类似地,将ARC参考信息放入PPS中具有另外的缺点,即由ARC参考信息(513)参考的ARC信息必须应用于整个图片而不是子图片,因为图片参数集激活的范围是图片。

[0111] 在相同或另一实施例中,ARC参数的信令可以遵循图6中所概述的详细示例。图6描绘了至少自1993年以来在视频编解码标准中使用的表示中的语法图。这种语法图的符号大致遵循C型编程。粗体行表示码流中存在的语法元素,非粗体行通常表示控制流或变量的设置。

[0112] 作为头的示例性语法结构的图块组头(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)(以黑体字示出),所述头可以应用于图片的(可能是矩形的)一部分。图块组头中的此语法元素的存在可以基于自适应分辨率的使用来进行门控(603)——这里,标志的值未以粗体字示出,其意味着标志存在于码流中该标志在语法图中出现的点处。自适应分辨率是否用于该图片或其中的部分,可以在码流内部或外部的任何高级语法结构中用信号通知。在所示的示例中,它在序列参数集中用信号通知,如下所概述。

[0113] 仍然参考图6,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,该标志可以指示自适应分辨率的使用,进而该自适应分辨率可能需要某些控制信息。在该示例中,基于参数集(612)中基于if()语句的标志的值和图块组头(601),这种控制信息有条件地存在。

[0114] 当使用自适应分辨率时,在该示例中,编码的是以样本(613)为单位的输出分辨率。附图标记613表示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,其可以是这两个语法元素的值的乘积。此外,某些视频编解码技术或标准,或外部技术或标准(诸如,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的幂数整除)或纵横比(例如,宽度和高度必须处于诸如4:3或16:9的关系)。可以引入这种限制以便于硬件实

现或出于其它原因,并且这种限制在本领域中是公知的。

[0115] 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,而不是隐含地假设该大小是输出图片大小。在该示例中,语法元素reference_pic_size_present_flag(614)门控参考图片尺寸(615)的条件式存在(同样,该附图标记是指宽度和高度两者)。

[0116] 最后,示出了可能的对图片宽度和高度进行解码的表。这样的表可以例如通过表指示(num_dec_pic_size_in_luma_samples_minus1)来表示(616)。“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。如果该值是五,则存在六个表条目。然后对于表中的每个“行”,将已解码图片宽度和高度包括在语法中(617)。

[0117] 可以使用图块组头中的语法元素dec_pic_size_idx(602)来索引所呈现(617)的表条目,从而允许每图块组中不同的已解码大小(实际上,缩放因子)。

[0118] 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(以与所公开的主题相当不同的方式用信号来表示)来支持空间可缩放性,以实现空间可缩放性。特别地,可以使用ARC式技术,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正常预测机制来细化那些上采样的图片,从而增加细节。

[0119] 所公开的主题可用于这样的环境中。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)可以不仅用于指示时间层,而且用于指示空间层。这样做对某些系统设计有一定的优势;例如,现有的选定转发单元(SFU,Selected Forwarding Units)是基于NAL单元头时间ID值针对时间层选择转发而创建和优化的,可以在不修改的情况下将该选定转发单元用于可扩展环境。为了实现这一点,可能需要已编码图片大小和时间层之间的映射由NAL单元头中的时间ID字段来指示。

[0120] 在一些视频编解码技术中,存取单元(AU,Access Unit)可以指在给定的时间实例(time instance)所采集并合成到相应图片/条带/图块/NAL单元码流中的一个或多个已编码图片、一个或多个条带、一个或多个图块、一个或多个NAL单元等。该时间实例可以是合成时间。

[0121] 在HEVC和某些其它视频编解码技术中,图片顺序计数(POC,picture order count)值可以用于指示存储在已解码图片缓冲器(DPB,decoded picture buffer)中的多个参考图片当中的选择的参考图片。当存取单元(AU,access unit)包括一个或多个图片、条带或图块时,属于相同AU的每个图片、条带或图块可以携带相同的POC值,从该POC值可以得出它们是从相同合成时间的内容创建的。换句话说,在两个图片/条带/图块携带相同的给定POC值的场景中,其可以指示属于相同AU并且具有相同合成时间的两个图片/条带/图块。相反,具有不同POC值的两个图片/条带/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/条带/图块。

[0122] 在所公开的主题的实施例中,由于访问单元可以包括具有不同POC值的图片、条带或图块,所以上述严格关系可以放宽。通过允许AU内的不同POC值,有可能使用POC值来识别具有相同呈现时间的潜在可独立解码的图片/条带/图块。这进而能够支持多个可伸缩层,而不改变参考图片选择信令(例如,参考图片集信令或参考图片列表信令),如下面更详细描述的。

[0123] 然而,相对于具有不同的POC值的其它图片/条带/图块,仅从POC值来识别图片/条

带/图块所属的AU仍然是合乎需要的。这可以如下所描述地实现。

[0124] 在相同或其它实施例中,可以在诸如NAL单元头、条带头、图块组头、SEI消息、参数集或AU定界符的高级语法结构中用信号通知存取单元计数(AUC,access unit count)。AUC的值可用于识别哪些NAL单元、图片、条带或图块属于给定AU。AUC的值可以与不同的合成时间实例相对应。AUC值可以等于POC值的倍数。通过将POC值除以整数值,可以计算出AUC值。在某些情况下,除法操作可能给解码器实施方案带来一定负担。在这种情况下,AUC值的编号空间中的小的限制可以允许通过移位运算代替除法运算。例如,AUC值可以等于POC值范围的最高有效位(MSB,Most Significant Bit)值。

[0125] 在相同实施例中,可以在诸如NAL单元头、条带头、图块组头、SEI消息、参数集或AU定界符的高级语法结构中用信号通知每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值可以与相同AU相关联。例如,如果poc_cycle_au的值等于4,则POC值等于0至3(包含端值)的图片、条带或图块与AUC值等于0的AU相关联,并且POC值等于4至7(包含端值)的图片、条带或图块与AUC值等于1的AU相关联。因此,可以通过将POC值除以poc_cycle_au的值来推断AUC的值。

[0126] 在相同或另一实施例中,poc_cyle_au的值可以根据位于(例如)视频参数集(VPS,video parameter set)中的信息得到,识别已编码视频序列中的空间或SNR层的数量的。下面简要描述这种可能的关系。虽然如以上所描述的推导可以在VPS中节省几个比特并且因此可以提高编解码效率,但是有利的是,以分层地适当的高级语法结构来显式地编码poc_cycle_au,以便能够针对码流(例如图片)的给定小部分来使poc_cycle_au最小化,所述适当的高级语法结构在视频参数集的下方。因为POC值(和/或间接参考POC的语法元素的值)可以用低级语法结构来编码,所以这种优化可以比通过以上推导过程节省更多的比特。在相同或另一实施例中,图9示出了语法表的示例,以用信号通知VPS(或SPS)中的vps_poc_cycle_au的语法元素(其指示用于已编码视频序列中的所有图片/条带的poc_cycle_au)以及条带头中条带_poc_cycle_au的语法元素(其指示当前条带的poc_cycle_au)。如果每AU的POC值均匀地增加,则将VPS中的vps_contant_poc_cycle_per_au设置为等于1,并且在VPS中用信号通知vps_poc_cycle_au。在这种情况下,没有显式地用信号通知slice_poc_cycle_au,并且通过将POC的值除以vps_poc_cycle_au来计算每个AU的AUC的值。如果每AU的POC值不均匀地增加,则将VPS中的vps_contant_poc_cycle_per_au设置为等于0。在这种情况下,不用信号通知vps_access_unit_cnt,同时在每个条带或图片的条带头中用信号通知slice_access_unit_cnt。每个条带或图片可以具有不同的slice_access_unit_cnt值。通过将POC值除以slice_poc_cycle_au来计算每个AU的AUC值。图10示出了图示相关工作流程的框图。

[0127] 在相同或其它实施例中,即使图片、条带或图块的POC值可能不同,对应于AU的图片、条带或图块可以与具有相同解码或输出时间实例相关联,所述AU具有相同的AUC值。因此,在相同AU中的图片、条带或图块之间没有任何内部解析和/或解码依赖性的情况下,可以并行地解码与相同AU相关联的图片、条带或图块的全部或子集,并且可以在相同时间实例输出。

[0128] 在相同或其它实施例中,即使图片、条带或图块的POC值可能不同,但与AU相对应的图片、条带或图块可以与相同组合物/显示时间实例相关联,所述AU具有相同的AUC值。当

合成时间被包含在容器格式中时,即使图片与不同的AU相对应,如果图片具有相同的合成时间,则可以在相同的时间实例显示图片。

[0129] 在相同或其它实施例中,每个图片、条带或图块在相同的AU中可以具有相同的时间标识符(temporal_id)。与时间实例相对应的图片、条带或图块的全部或子集可以与相同的时间子层相关联。在相同或其它实施例中,每个图片、条带或图块在相同的AU中可以具有相同或不同的空间层id(layer_id)。与时间实例相对应的图片、条带或图块的全部或子集可以与相同或不同的空间层相关联。

[0130] 图8示出了具有temporal_id、layer_id、POC和AUC值的组合的视频序列结构的示例,所述temporal_id、layer_id、POC和AUC值具有自适应分辨率改变。在该示例中,AUC=0的第一AU中的图片、条带或图块的temporal_id=0和layer_id=0或1,而AUC=1的第二AU中的图片、条带或图块的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。优选地,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值增加2,同时AUC值增加1。

[0131] 在以上实施例中,可以通过使用HEVC中的现有参考图片集合(RPS,reference picture set)信令或参考图片列表(RPL,reference picture list)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,通过用信号通知当前图片与选择的参考图片之间的POC值或POC增量值来指示所选择的参考图片。对于所公开的主题,RPS和RPL可用于指示图片间或层间预测结构而不改变信令,但具有以下限制。如果参考图片的temporal_id的值大于当前图片的temporal_id的值,则当前图片可以不将参考图片用于运动补偿或其它预测。如果参考图片的layer_id的值大于当前图片的layer_id的值,则当前图片可以不将参考图片用于运动补偿或其它预测。

[0132] 在相同和其它实施例中,可以在存取单元内的多个图片之间禁用基于POC差异的运动矢量缩放,所述POC差异用于时间运动矢量预测。因此,尽管每个图片在存取单元内可以具有不同的POC值,但运动矢量并未被缩放,并且用于存取单元内的时间运动矢量预测。这是因为在相同AU中具有不同POC的参考图片被认为是具有相同时间实例的参考图片。因此,在实施例中,当参考图片属于与当前图片相关联的AU时,运动矢量缩放函数可以返回1。

[0133] 在相同和其它实施例中,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用基于POC差异的运动矢量缩放,所述POC差异用于时间运动矢量预测。当允许运动矢量缩放时,基于当前图片与参考图片之间的POC差异以及空间分辨率比率来缩放运动矢量。

[0134] 在相同或另一实施例中,对于时间运动矢量预测,特别是当poc_cycle_au具有非均匀值时(例如,当vps_contant_poc_cycle_per_au==0时),可以基于AUC差异而不是POC差异来缩放运动矢量。否则(当vps_contant_poc_cycle_per_au==1时),基于AUC差异的运动矢量缩放可以与基于POC差异的运动矢量缩放相同。

[0135] 在相同或另一实施例中,当基于AUC差异对运动矢量进行缩放时,不基于AUC差异来对与当前图片在同一AU(具有相同的AUC值)中的参考运动矢量进行缩放,并且所述参考运动矢量用于运动矢量预测二不需要进行缩放或者需要基于基于当前图片与参考图片之间的空间分辨率比例进行缩放。

[0136] 在相同和其它实施例中,AUC值用于识别AU的边界,并且用于假设参考解码器(HRD,hypothetical reference decoder)操作,该操作需要以AU粒度的输入和输出定时。在大多数情况下,可以输出AU中最高层的已解码图片用于显示。AUC值和layer_id值可以被用于识别输出图片。

[0137] 在实施例中,图片可以由一个或多个子图片组成。每个子图片可以覆盖图片的局部区域或整个区域。由子图片支持的区域可以与由另一个子图片支持的区域重叠或不重叠。由一个或多个子图片组成的区域可以覆盖或不覆盖图片的整个区域。如果图片由子图片组成,则该子图片所支持的区域与该图片所支持的区域相同。

[0138] 在相同实施例中,可以通过编解码方法来对子图片进行编解码,所述编解码方法与用于已编码图片的编解码方法相似。可以独立地对子图片进行编码,或者可以取决于另一个子图片或已编码图片对子图片进行编码。子图片可以具有或可以不具有与另一个子图片或已编码图片的任何解析依赖性。

[0139] 在相同实施例中,已编码子图片可以被包含在一个或多个层中。层中的已编码子图片可以具有不同的空间分辨率。原始子图片可以在空间上进行重采样(上采样或下采样),用不同的空间分辨率参数进行编码,并且被包含在与层相对应的码流中。

[0140] 在相同或另一实施例中,可以对具有(W,H)的子图片(其中,分别地,W指示子图片的宽度,并且H指示子图片的高度)进行编码,并且具有(W,H)的子图片包含在与层0相对应的已编码码流中,同时可以对来自具有原始空间分辨率的子图片的上采样(或下采样)子图片(具有($W*S_{w,k}$,$H*S_{h,k}$))进行编码并且子图片(具有($W*S_{w,k}$,$H*S_{h,k}$))包含在与层k相对应的已编码码流中,其中,$S_{w,k}$和$S_{h,k}$指示水平重采样比率和垂直重采样比率。如果$S_{w,k}$、$S_{h,k}$的值大于1,则重采样等于上采样。然而,如果$S_{w,k}$、$S_{h,k}$的值小于1,则重采样等于下采样。

[0141] 在相同或另一实施例中,层中的已编码子图片的视觉质量与相同子图片或不同子图片中的另一个层中的已编码子图片的视觉质量不同。例如,层n中的子图片i用量化参数$Q_{i,n}$来编码,同时层m中的子图片j用量化参数$Q_{j,m}$来编码。

[0142] 在相同或另一实施例中,层中的已编码子图片可以是可独立解码的,而没有来自相同局部区域的另一个层中的已编码子图片的任何解析或解码依赖性。在不参考相同局部区域的另一个子图片层的情况下,可独立解码的子图片层是独立子图片层。独立子图片层中的已编码子图片可以具有或不具有来自相同子图片层中的先前已编码子图片的解码或解析依赖性,但是已编码子图片可以不具有与另一个子图片层中的已编码图片的任何依赖性。

[0143] 在相同或另一实施例中,层中的已编码子图片可以是依赖地可解码的,具有来自相同局部区域的另一个层中的已编码子图片的任何解析依赖性或解码依赖性。可通过参考相同局部区域的另一个子图片层而进行可依赖解码的子图片层是依赖子图片层。依赖子图片中的已编码子图片可以参考属于相同子图片的已编码子图片、相同子图片层中的先前已编码子图片,或这两种参考子图片。

[0144] 在相同或另一实施例中,已编码子图片由一个或多个独立子图片层和一个或多个依赖子图片层组成。然而,对于已编码子图片,可以存在至少一个独立子图片层。独立子图片层可以具有等于0的层标识符(layer_id)的值,其可以存在于NAL单元头或另一高级语法结构中。layer_id等于0的子图片层是基础子图片层。

[0145] 在相同或另一实施例中,图片可由一个或多个前景子图片和一个背景子图片组成。背景子图片所支持的区域可以等于图片的区域。前景子图片所支持的区域可以与背景子图片所支持的区域重叠。背景子图片可以是基础子图片层,同时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相同的基础层(base layer)进行解码。layer_id等于a的每个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。

[0146] 在相同或另一实施例中,图片可以由具有或不具有背景子图片的一个或多个前景子图片组成。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。每个基础子图片层可以由一个或多个非基础子图片层参考。layer_id等于a的每个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。

[0147] 在相同或另一实施例中,图片可以由具有或不具有背景子图片的一个或多个前景子图片组成。(基础或非基础)子图片层中的每个已编码子图片可以由属于相同子图片的一个或多个非基础层子图片和不属于相同子图片的一个或多个非基础层子图片来参考。

[0148] 在相同或另一实施例中,图片可以由具有或不具有背景子图片的一个或多个前景子图片组成。层a中的子图片可以被进一步分区成相同层中的多个子图片。层b中的一个或多个已编码子图片可以参考层a中的分区子图片。

[0149] 在相同或另一实施例中,已编码视频序列(CVS,coded video sequence)可以是一组已编码图片。CVS可以由一个或多个已编码子图片序列(CSPS,coded sub-picture sequence)组成,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。

[0150] 在相同或另一实施例中,可以对CSPS进行编码,并将其包含在一个或多个层中。CSPS可以由一个或多个CSPS层组成。解码与CSPS相对应的一个或多个CSPS层可以重建与相同局部区域相对应的子图片序列。

[0151] 在相同或另一实施例中,对应于CSPS的CSPS层的数量与对应于另一个CSPS的CSPS层的数量相同或不同。

[0152] 在相同或另一实施例中,CSPS层可以具有与另一个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进行重采样(上采样或下采样),可以用不同的时间分辨率参数进行编码,并且可以被包含在与层相对应的码流中。

[0153] 在相同或另一实施例中,具有帧速率F的子图片序列可以被编码并且包含在与层0相对应的已编码码流中,同时来自原始子图片序列的具有$F*S_{t,k}$的时间上采样(或下采样)的子图片序列可以被编码并且包含在与层k相对应的已编码码流中,其中,$S_{t,k}$指示层k的时间采样比。如果$S_{t,k}$的值大于1,则时间重采样过程等于帧速率上转换(up conversion)。然而,如果$S_{t,k}$的值小于1,则时间重采样过程等于帧速率下转换(down conversion)。

[0154] 在相同或另一实施例中,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,则将CSPS层a中的已解码像素进行重采样并且用于参考。重采样过程可能需要上采样滤波或下采样滤波。

[0155] 图11示出了包括layer_id等于0的的背景视频CSPS和多个前景CSPS层的示例视频码流。已编码子图片可以由一个或多个CSPS层组成,而不属于任何前景CSPS层的背景区域

可以由基础层(base layer)组成。基础层可以包含背景区域和前景区域,同时增强CSPS层包含前景区域。在相同区域处,增强CSPS层可以具有比基础层更好的视觉质量。增强CSPS层可以参考与相同区域相对应的基础层的重建像素和运动矢量。

[0156] 在相同或另一实施例中,在视频文件中,与基础层相对应的视频码流包含在轨道中,同时与每个子图片相对应的CSPS层包含在分离的轨道中。

[0157] 在相同或另一实施例中,与基础层相对应的视频码流包含在轨道中,同时具有相同layer_id的CSPS层包含在分离的轨道中。在该示例中,与层k相对应的轨道仅包括与层k相对应的CSPS层。

[0158] 在相同或另一实施例中,每个子图片的每个CSPS层存储在单独的轨道中。每个轨道可以具有或可以不具有来自一个或多个其它轨道的任何解析或解码依赖性。

[0159] 在相同或另一实施例中,每个轨道可以包含与子图片的全部或子集的CSPS层的层i到层j相对应的码流,其中,0<i=<j=<k,k是CSPS的最高层。

[0160] 在相同或另一实施例中,图片由一个或多个相关联的媒体数据组成,媒体数据包括深度图、阿尔法图、3D几何数据、占用图等。这样的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,每个数据子流与一个子图片相对应。

[0161] 在相同或另一实施例中,图12示出了基于多层子图片方法的视频会议的示例。在视频码流中,包含与背景图片相对应的一个基础层视频码流和与前景子图片相对应的一个或多个增强层视频码流。每个增强层视频码流与CSPS层相对应。在显示器中,默认地显示与基础层相对应的图片。它包含一个或多个用户画中画(PIP,picture in a picture)。当通过客户端的控制选择特定用户时,以增强的质量或空间分辨率来解码和显示与选择的用户相对应的增强CSPS层。图13示出了操作的图。

[0162] 在相同或另一实施例中,网络中间盒(诸如路由器)可以根据其带宽来选择要发送给用户的层的子集。图片/子图片组织可以用于带宽自适应。如果用户不具有带宽,则由于层或一些子图片的重要性或基于所使用的设置,路由器可以剥离层或选择一些子图,这可以动态地进行以适应带宽。

[0163] 图14示出了360视频的使用情况。当球形360图片被投影到平面图片上时,投影360图片可以被分区成多个子图片作为基础层。可以对特定子图片的增强层进行编码并且将其传输到客户端。解码器能够解码包括所有子图片的基础层和选择的子图片的增强层。当当前视口与选择的子图片相同时,显示的图片可以比具有增强层的已解码子图片具有更高的质量。否则,可以以低质量显示具有基础层的已解码图片。

[0164] 在相同或另一实施例中,用于显示的任何布局信息可以作为补充信息(诸如SEI消息或元数据)存在于文件中。取决于用信号通知的布局信息,可以重新定位和显示一个或多个已解码子图片。布局信息可以由流服务器(streaming server)或广播器用信号通知,或者可以由网络实体或云服务器重新生成,或者可以由用户的自定义设置来确定。

[0165] 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。与局部区域相对应的每个独立层可以具有唯一的layer_id值。对于每个独立层,可以用信号通知子图片大小和位置信息。例如,图片大小(宽度,高度)、左上角的偏移信息(x_offset,y_offset)。图15示出了划分的子图片的布局、其子图片大小和位置信息、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中用信号通知,例如如在一个或多个参数集、条带或图块组的头或SEI消息中通知。

[0166] 在相同实施例中,每个独立层对应的子图片在一个AU内可能有其唯一的POC值。当通过使用RPS或RPL结构中的一个或多个语法元素来指示存储在DPB中的图片当中的参考图片时,可以使用与层相对应的每个子图片的一个或多个POC值。

[0167] 在相同或另一实施例中,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。

[0168] 在相同实施例中,与的POC值等于N的子图片可以被用作或可以不被用作POC值等于N+K的子图片的参考图片,所述POC值等于N的子图片与层(或局部区域)相对应,所述POC值等于N+K的子图片与运动补偿预测的相同层(或相同局部区域)相对应。在大多数情况下,数量K的值可以等于(独立)层的最大数量,其可以与子区域的数量相同。

[0169] 在相同或另一实施例中,图16示出了图15的扩展情况。当输入图片被划分成多个(例如,四个)子区域时,每个局部区域可以用一个或多个层来编码。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与子区域相对应。因此,每个子区域可以用一个或多个独立层和零个或多个依赖层来编码。

[0170] 在相同实施例中,在图16中,输入图片可以被划分成四个子区域。右上子区域可以被编码为两层,即,层1和层4,同时右下子区域可以被编码为两层,即,层3和层5。在这种情况下,层4可以参考层1用于运动补偿预测,同时层5可以参考层3用于运动补偿。

[0171] 在相同或另一实施例中,可以(可选地)禁用跨层边界的环路滤波(例如去块滤波、自适应环路滤波、整形器、双边滤波或任何基于深度学习的滤波)。

[0172] 在相同或另一实施例中,可以(可选地)禁用跨层边界的运动补偿预测或块内复制。

[0173] 在相同或另一实施例中,可以可选地处理用于运动补偿预测的边界填充或者在子图片的边界处的环路滤波。可以在例如一个或多个参数集(VPS、SPS、PPS或APS)、条带或图块组头或SEI消息的高级语法结构中用信号通知用于指示边界填充是否被处理的标志。

[0174] 在相同或另一实施例中,可以在VPS或SPS中用信号通知一个或多个子区域(或一个或多个子图片)的布局信息。图17示出了VPS和SPS中的语法元素的示例。在该示例中,在VPS中用信号通知vps_sub_picture_dividing_flag。该标志可以指示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相对应的一个或多个已编码视频序列中的一个或多个输入图片可以不被划分成多个子区域。在这种情况下,输入图片大小可以等于在SPS中用信号通知的已编码图片大小(pic_width_in_luma_samples,pic_height_in_luma_samples)。当vps_sub_picture_dividing_flag的值等于1时,一个或多个输入图片可以被划分成多个子区域。在这种情况下,在VPS中用信号通知语法元素vps_full_pic_width_in_luma_samples和vps_full_pic_height_in_luma_samples。vps_full_pic_width_in_luma_samples和vps_full_pic_height_in_luma_samples的值可以分别等于一个或多个输入图片的宽度和高度。

[0175] 在相同实施例中,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。

[0176] 在相同实施例中,当vps_sub_picture_dividing_flag的值等于1时,可以在SPS中

用信号通知语法元素pic_offset_x和pic_offset_y,其与一个或多个特定层相对应。在这种情况下,在SPS中用信号通知的已编码图片大小(pic_width_in_luma_samples,pic_height_in_luma_samples)可以等于与特定层相对应的子区域的宽度和高度。此外,可以在SPS中用信号通知子区域的左上角的位置(pic_offset_x,pic_offset_y)。

[0177] 在相同实施例中,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,而是可以被用于合成和显示。

[0178] 在相同或另一实施例中,可以在参数集或SEI消息中用信号通知一个或多个输入图片的全部或一个或多个子集子区域的布局信息(大小和位置)以及一个或多个层之间的依赖性信息。图18示出了指示子区域的布局、层之间的依赖性以及子区域与一个或多个层之间的关系的信息的语法元素的示例。在该示例中,语法元素num_sub_region指示当前已编码视频序列中的(矩形)子区域的数量。语法元素num_layers指示当前已编码视频序列中的层数。num_layers的值可以等于或大于num_sub_region的值。当将任何子区域编码为单层时,num_layers的值可以等于num_sub_region的值。当将一个或多个子区域编码为多个层时,num_layers的值可以大于num_sub_region的值。语法元素direct_dependency_flag[i][j]指示从第j层到第i层的依赖性。num_layers_for_region[i]指示与第i子区域相关联的层的数量。sub_region_layer_id[i][j]指示与第i子区域相关联的第j层的layer_id。sub_region_offset_x[i]和sub_region_offset_y[i]分别指示第i子区域的左上角的水平位置和垂直位置。sub_region_width[i]和sub_region_height[i]分别指示第i子区域的宽度和高度。

[0179] 在一个实施例中,可以在例如VPS、DPS、SPS、PPS、APS或SEI消息的高级语法结构中用信号通知用于指示输出层集合的一个或多个语法元素,该输出层集合用于指示在有或没有配置文件层级(profile tier level)信息的情况下要输出的多个层中的一个。参考图19,可以在VPS中用信号通知语法元素num_output_layer_sets,所述语法元素num_output_layer_sets用于指示参考VPS的已编码视频序列中的输出层集合(OLS)的数量。对于每个输出层集合,可以用信号通知与输出层的数量一样多的output_layer_flag。

[0180] 在相同实施例中,output_layer_flag[i]等于1可以指示输出第i层。vps_output_layer_flag[i]等于0可以指示指定不输出第i层。

[0181] 在相同或另一实施例中,可以在高级语法结构(例如,VPS、DPS、SPS、PPS、APS或SEI消息)中用信号通知用于指示每个输出层集合的配置文件层级层级信息的一个或多个语法元素。仍然参考图19,可以在VPS中用信号通知语法元素num_profile_tile_level,所述语法元素num_profile_tile_level用于指示参考VPS的已编码视频序列中每个OLS的配置文件层级信息的数量。对于每个输出层集合,配置文件层级信息的语法元素的集合或用于指示特定配置文件层级信息的索引可以用信号通知的与输出层的数量一样多,特定配置文件层级信息在所述配置文件层级信息中的条目当中。

[0182] 在相同实施例中,profile_tier_level_idx[i][j]将应用于第i个OLS的第j层的profile_tier_level()语法结构的索引指定到VPS中的profile_tier_level()语法结构的列表中。

[0183] 在相同或另一实施例中,参考图20,当最大层的数量大于1(vps_max_layers_minus1>0)时,可以用信号通知语法元素num_profile_tile_level和/或num_output_

layer_sets。

[0184] 在相同或另一实施例中,参考图20,指示第i个输出层集合的输出层信令的模式的语法元素vps_output_layers_mode[i]存在于VPS中。

[0185] 在相同实施例中,vps_output_layers_mode[i]等于0可以指示只有最高层与第i个输出层集合一起输出。vps_output_layer_mode[i]等于1指定所有层与第i个输出层集合一起输出。vps_output_layer_mode[i]等于2指定被输出的层是vps_output_layer_flag[i][j]等于1的层和第i个输出层集合。可以保留更多的值。

[0186] 在相同实施例中,取决于第i个输出层集合的vps_output_layers_mode[i]的值,output_layer_flag[i][j]可以用信号通知或可以不用信号通知。

[0187] 在相同或另一实施例中,参考图20,标志vps_ptl_signal_flag[i]可以针对第i个输出层集合而存在。取决于vps_ptl_signal_flag[i]的值,第i个输出层集合的配置文件层级信息可以用信号通知或可以不用信号通知。

[0188] 在相同或另一实施例中,参考图21,可以在例如VPS、DPS、SPS、PPS、APS或SEI消息的高级语法结构中用信号通知当前CVS中的子图片的数量max_subpics_minus1。

[0189] 在相同实施例中,参考图21,当子图片的数量大于1(max_subpics_minus1>0)时,可以用信号通知第i子图片的子图片标识符sub_pic_id[i]。

[0190] 在相同或另一实施例中,可以在VPS中用信号通知用于指示属于每个输出层集合的每个层的子图片标识符的一个或多个语法元素。参考图22,sub_pic_id_layer[i][j][k]指示第i输出层集合的第j层中存在的第k个子图片。利用这些信息,解码器可以识别针对特定输出层集合的每一层可以解码和输出哪个子图片。

[0191] 在实施例中,图片头(PH,picture header)是包含语法元素的语法结构,所述语法元素应用于已编码图片的所有条带。图片单元(PU,picture unit)是NAL单元的集合,NAL单元根据指定的分类规则彼此相关联,是按照解码顺序连续的,并且包含正好一个已编码图片。PU可以包含图片头(PH,picture header)和构成已编码图片的一个或多个VCL NAL单元。

[0192] 在实施例中,SPS(RBSP)可以在其被参考之前用于解码过程,SPS(RBSP)可以包括在TemporalId等于0的至少一个AU中或通过外部方式提供。

[0193] 在实施例中,SPS(RBSP)可以在其被参考之前可用于解码过程,SPS(RBSP)包括在CVS中TemporalId等于0的至少一个AU中,或通过外部手段提供,所述CVS包含参考SPS的一个或多个PPS。

[0194] 在实施例中,SPS(RBSP)可以在其被一个或多个PPS参考之前用于解码过程,SPS(RBSP)包括在nuh_layer_id等于PPS NAL单元的最低nuh_layer_id值的至少一个PU中,或通过外部方式提供,所述PPS NAL单元参考CVS中的参考SPS NAL单元,所述CVS包含参考SPS的一个或多个PPS。

[0195] 在实施例中,SPS(RBSP)可以在其被一个或多个PPS参考之前用于解码过程,SPS(RBSP)包括在TemporalId等于0并且nuh_layer_id等于PPS NAL单元的最低nuh_layer_id值的至少一个PU中,或通过外部手段提供,所述PPS NAL单元参考SPS NAL单元。

[0196] 在实施例中,SPS(RBSP)可以在其被一或多个PPS参考之前用于解码过程,SPS(RBSP)包括在CVS中TemporalId等于0并且nuh_layer_id等于PPS NAL单元的最低nuh_

layer_id值的至少一个PU中,或通过外部方式提供,所述PPS NAL单元参考SPS NAL单元,所述CVS包含参考SPS的一个或多个PPS。

[0197] 在相同或另一实施例中,pps_seq_parameter_set_id可以指示所参考的SPS的sps_seq_parameter_set_id的值。pps_seq_parameter_set_id的值在所有PPS中可以是相同的,所述所有PPS由CLVS中的已编码图片参考。

[0198] 在相同或另一实施例中,CVS中具有sps_seq_parameter_set_id的特定值的所有SPS NAL单元可以具有相同内容。

[0199] 在相同或另一实施例中,不管nuh_layer_id值如何,SPS NAL单元可共享sps_seq_parameter_set_id的相同值空间。

[0200] 在相同或另一实施例中,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。

[0201] 在实施例中,当nuh_layer_id等于m的SPS由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的层相同或与nuh_layer_id等于m的层的(直接或间接)参考层相同。

[0202] 在实施例中,PPS(RBSP)可以在其被参考之前用于解码过程,PPS(RBSP)包括在TemporalId等于PPS NAL单元的TemporalId的至少一个AU中,或通过外部方式提供。

[0203] 在实施例中,PPS(RBSP)可以在其被参考之前用于解码过程,PPS(RBSP)包括在CVS中TemporalId等于PPS NAL单元的TemporalId的至少一个AU中,或通过外部方式提供,CVS包含参考PPS的一个或多个PH(或已编码条带NAL单元)。

[0204] 在实施例中,PPS(RBSP)可以在其被一个或多个PH(或已编码条带NAL单元)参考之前用于解码过程,PPS(RBSP)包括在CVS中nuh_layer_id等于参考PPS NAL单元的已编码条带NAL单元的最低nuh_layer_id值的至少一个PU中,或通过外部手段提供,所述CVS包含参考PPS的一个或多个PH(或已编码条带NAL单元))。

[0205] 在实施例中,PPS(RBSP)可以在其被一个或多个PH(或已编码条带NAL单元)参考之前用于解码过程,PPS(RBSP)包括在CVS中TemporalId等于PPS NAL单元的TemporalId和nuh_layer_id等于已编码条带NAL单元的最低nuh_layer_id值的至少一个PU中,或通过外部方式提供,CVS包含参考PPS的一个或多个PH(或已编码条带NAL单元),所述已编码条带NAL单元参考PPS NAL单元。

[0206] 在相同或另一实施例中,PH中的ph_pic_parameter_set_id可以指示使用中的所参考的PPS的pps_pic_parameter_set_id的值。pps_seq_parameter_set_id的值在所有PPS中可以是相同的,所述所有的PPS由CLVS中的已编码图片参考。

[0207] 在相同或另一实施例中,PU内具有pps_pic_parameter_set_id的特定值的所有PPS NAL单元可以可具有相同内容。

[0208] 在相同或另一实施例中,不管nuh_layer_id值如何,PPS NAL单元可以共享pps_pic_parameter_set_id的相同值空间。

[0209] 在相同或另一实施例中,PPS NAL单元的nuh_layer_id值可以等于已编码条带NAL单元的最低nuh_layer_id值,所述已编码条带NAL单元参考NAL单元,所述NAL单元参考PPS NAL单元。

[0210] 在实施例中,当nuh_layer_id等于m的PPS由nuh_layer_id等于n的一个或多个已

编码条带NAL单元参考时,nuh_layer_id等于m的层可以与nuh_layer_id等于n的层相同或与nuh_layer_id等于m的层的(直接或间接)参考层相同。

[0211] 在实施例中,PPS(RBSP)可以在其被参考之前用于解码过程,PPS(RBSP)包括在TemporalId等于PPS NAL单元的TemporalId的至少一个AU中,或通过外部方式提供。

[0212] 在实施例中,PPS(RBSP)可以在其被参考之前用于解码过程,PPS(RBSP)包括在CVS中TemporalId等于PPS NAL单元的TemporalId的至少一个AU中,或通过外部方式提供,CVS包含参考PPS的一个或多个PH(或已编码条带NAL单元)。

[0213] 在实施例中,PPS(RBSP)可以在其被一个或多个PH(或已编码条带NAL单元)参考之前用于解码过程,PPS(RBSP)包括在CVS中nuh_layer_id等于参考PPS NAL单元的已编码条带NAL单元的最低nuh_layer_id值的至少一个PU中,或通过外部手段提供(其包含参考PPS的一个或多个PH(或已编码条带NAL单元))。

[0214] 在实施例中,PPS(RBSP)可以在其被一个或多个PH(或已编码条带NAL单元)参考之前用于解码过程,PPS(RBSP)包括在CVS中TemporalId等于PPS NAL单元的TemporalId和nuh_layer_id等于已编码条带NAL单元的最低nuh_layer_id值的至少一个PU中,或通过外部方式提供,CVS包含参考PPS的一个或多个PH(或已编码条带NAL单元),所述已编码条带NAL单元参考PPS NAL单元。

[0215] 在相同或另一实施例中,PH中的ph_pic_parameter_set_id可以指示使用中的参考的PPS的pps_pic_parameter_set_id的值。pps_seq_parameter_set_id的值在由CLVS中的已编码图片参考的所有PPS中可以是相同的。

[0216] 在相同或另一实施例中,PU内具有pps_pic_parameter_set_id的特定值的所有PPS NAL单元可以可具有相同内容。

[0217] 在相同或另一实施例中,不管nuh_layer_id值如何,PPS NAL单元可以共享pps_pic_parameter_set_id的相同值空间。

[0218] 在相同或另一实施例中,PPS NAL单元的nuh_layer_id值可以等于参考PPS NAL单元的已编码条带NAL单元的最低nuh_layer_id值。

[0219] 在实施例中,当nuh_layer_id等于m的PPS由nuh_layer_id等于n的一个或多个已编码条带NAL单元参考时,nuh_layer_id等于m的层可以与nuh_layer_id等于n的层相同或与nuh_layer_id等于m的层的(直接或间接)参考层相同。

[0220] 输出层指示输出层集合中被输出的层。输出层集合(OLS,output layer set)指示由特定的层的集合组成的层集合,其中,该层集合中的一个或多个层被指定为输出层。输出层集合(OLS)层索引是OLS中的层到OLS中的层列表的索引。

[0221] 子层可以指示时间可伸缩码流的时间可伸缩层,由具有TemporalId变量的特定值的VCL NAL单元和相关联的非VCL NAL单元组成。子层表示可以指示由特定子层和较低子层的NAL单元组成的码流的子集。

[0222] 在实施例中,VPS RBSP可以在其被参考之前用于解码过程,VPS RBSP包括在TemporalId等于0的至少一个AU中,或通过外部方式提供。CVS中具有vps_video_parameter_set_id的特定值的所有VPS NAL单元可以具有相同内容。图22示出了VPS RBSP的示例。

[0223] vps_video_parameter_set_id提供VPS的标识符以供其它语法元素参考。vps_

video_parameter_set_id的值可以大于0。

[0224] vps_max_layers_minus1加1可以指示参考VPS的每个CVS中的最大允许层数。

[0225] vps_max_sublayers_minus1+1用于指示可以存在于参考VPS的每个CVS中的层中的时间子层的最大数量。vps_max_sublayer_minus1的值可以在0至6(包含端值)的范围内。

[0226] vps_all_layers_same_num_sublayers_flag等于1用于指示时间子层的数量对于参考VPS的每个CVS中的所有层是相同的。vps_all_layers_same_num_sublayers_flag等于0指示参考VPS的每个CVS中的层可以具有或可以不具有相同数量的时间子层。当不存在时,推断vps_all_layers_same_num_sublayers_flag的值等于1。

[0227] vps_all_independent_layers_flag等于1用于指示在不使用层间预测的情况下独立地编码CVS中的所有层。vps_all_independent_layers_flag等于0可以指示CVS中的一个或多个层可以使用层间预测。当不存在时,推断vps_all_independent_layers_flag的值等于1。

[0228] vps_layer_id[i]指示第i层的nuh_layer_id值。对于m和n的任意两个非负整数值,当m小于n时,vps_layer_id[m]的值可以小于vps_layer_id[n]。

[0229] vps_independent_layer_flag[i]等于1指示具有索引i的层不使用层间预测。vps_independent_layer_flag[i]等于0指定具有索引i的层可以使用层间预测,并且在0至i-1(包含端值)的范围内的j的语法元素vps_direct_ref_layer_flag[i][j]存在于VPS中。当不存在时,推断vps_independent_layer_flag[i]的值等于1。

[0230] vps_direct_ref_layer_flag[i][j]等于0指示具有索引j的层不是具有索引i的层的直接参考层。vps_direct_ref_layer_flag[i][j]等于1指示具有索引j的层是具有索引i的层的直接参考层。当对于在0至vps_max_layers_minus1(包含端值)的范围内的i和j不存在vps_direct_ref_layer_flag[i][j]时,推断其等于0。当vps_independent_layer_flag[i]等于0时,可以存在在0至i-1(包含端值)的范围内的j的至少一个值,使得vps_direct_ref_layer_flag[i][j]的值等于1。

[0231] 变量NumDirectRefLayers[i]、DirectRefLayerIdx[i][d]、NumRefLayers[i]、RefLayerIdx[i][r]和LayerUsedAsRefLayerFlag[j]如下导出:

```
for( i = 0; i <= vps_max_layers_minus1; i++ ) {
    for( j = 0; j <= vps_max_layers_minus1; j++ ) {
        dependencyFlag[ i ][ j ] = vps_direct_ref_layer_flag[ i ][ j ]
        for( k = 0; k < i; k++ )
            if( vps_direct_ref_layer_flag[ i ][ k ] && dependencyFlag[ k ][ j ] )
```

[0232]

```
                dependencyFlag[ i ][ j ] = 1
    }
    LayerUsedAsRefLayerFlag[ i ] = 0
}
for( i = 0; i <= vps_max_layers_minus1; i++ ) {
    for( j = 0, d = 0, r = 0; j <= vps_max_layers_minus1; j++ ) {
```

```
            if( vps_direct_ref_layer_flag[ i ][ j ] ) {
                DirectRefLayerIdx[ i ][ d++ ] = j
                LayerUsedAsRefLayerFlag[ j ] = 1
            }
            if( dependencyFlag[ i ][ j ] )
                RefLayerIdx[ i ][ r++ ] = j
        }
        NumDirectRefLayers[ i ] = d
        NumRefLayers[ i ] = r
    }
```

[0233]

[0234] 用于指示nuh_layer_id等于vps_layer_id[i]的层的层索引的变量GeneralLayerIdx[i]如下导出:

[0235] for(i＝0;i<＝vps_max_layers_minus1;i++)

[0236] GeneralLayerIdx[vps_layer_id[i]]＝i

[0237] 对于i和j的任何两个不同值(均在0至vps_max_layers_minus1(包含端值)的范围内),当dependencyFlag[i][j]等于1时,码流一致性的要求是应用于第i层的chroma_format_idc和bit_depth_minus8的值可以分别等于应用于第j层的chroma_format_idc和bit_depth_minus8的值。

[0238] max_tid_ref_present_flag[i]等于1指示语法元素max_tid_il_ref_pics_plus1[i]存在。max_tid_ref_present_flag[i]等于0指示语法元素max_tid_il_ref_pics_plus1[i]不存在。

[0239] max_tid_il_ref_pics_plus1[i]等于0指示第i层的非IRAP图片不使用层间预测。max_tid_il_ref_pics_plus1[i]大于0指示对于第i层的解码图片,不使用TemporalId大于max_tid_il_ref_pics_plus1[i]-1的图片作为ILRP。当不存在时,推断max_tid_il_ref_pics_plus1[i]的值等于7。

[0240] each_layer_is_an_ols_flag等于1指示每个OLS仅包含一个层,并且在参考VPS的CVS中的每个层本身是OLS,该OLS包括的单个层是唯一的输出层。each_layer_is_an_ols_flag等于0指示OLS可以包含多于一个层。如果vps_max_layers_minus1等于0,则推断each_layer_is_an_ols_flag的值等于1。否则,当vps_all_independent_layers_flag等于0时,推断each_layer_is_an_ols_flag的值等于0。

[0241] ols_mode_idc等于0指示由VPS指定的OLS的总数量等于vps_max_layers_minus1+1,第i个OLS包括层索引从0至i(包含端值)的层,并且对于每个OLS仅输出OLS中的最高层。

[0242] ols_mode_idc等于1指示由VPS指定的OLS的总数量等于vps_max_layers_minus1+1,第i个OLS包括层索引从0至i(包含端值)的层,并且对于每个OLS输出OLS中的所有层。

[0243] ols_mode_idc等于2指示由VPS指定的OLS的总数量被显式地用信号通知,并且对于每个OLS,输出层被显式地用信号通知,并且其它层是作为OLS的输出层的直接或间接参

考层的层。

[0244] ols_mode_idc的值可以在0至2(包含端值)的范围内。ols_mode_idc的值3由ITU-T|ISO/IEC保留以供将来使用。

[0245] 当vps_all_independent_layers_flag等于1并且each_layer_is_an_ols_flag等于0时,推断ols_mode_idc的值等于2。

[0246] num_output_layer_sets_minus1加1指示当ols_mode_idc等于2时由VPS指示的OLS的总数量。

[0247] 指定由VPS指定的OLS的总数量的变量TotalNumOlss如下导出:

```
if( vps_max_layers_minus1 == 0 )
    TotalNumOlss = 1
else if( each_layer_is_an_ols_flag || ols_mode_idc == 0 ||
```

[0248]

```
ols_mode_idc == 1 )
    TotalNumOlss = vps_max_layers_minus1 + 1
else if( ols_mode_idc == 2 )
    TotalNumOlss = num_output_layer_sets_minus1 + 1
```

[0249] ols_output_layer_flag[i][j]等于1指示当ols_mode_idc等于2时,nuh_layer_id等于vps_layer_id[j]的层是第i个OLS的输出层。ols_output_layer_flag[i][j]等于0指示当ols_mode_idc等于2时,nuh_layer_id等于vps_layer_id[j]的层不是第个i OLS的输出层。

[0250] 指示第个i OLS中的输出层的数量的变量NumOutputLayersInOls[i]、指示第i个OLS中的第j层中的子层的数量的变量NumSubLayersInLayerInOLS[i][j]、指示第i OLS中的第j输出层的nuh_layer_id值的变量OutputLayerIdInOls[i][j]以及指示第k个层是否用作至少一个OLS中的输出层的变量LayerUsedAsOutputLayerFlag[k]如下导出:

```
NumOutputLayersInOls[ 0 ] = 1
OutputLayerIdInOls[ 0 ][ 0 ] = vps_layer_id[ 0 ]
NumSubLayersInLayerInOLS[ 0 ][ 0 ] = vps_max_sub_layers_minus1 + 1
LayerUsedAsOutputLayerFlag[ 0 ] = 1
for( i = 1, i <= vps_max_layers_minus1; i++ ) {
    if( each_layer_is_an_ols_flag || ols_mode_idc < 2 )
        LayerUsedAsOutputLayerFlag[ i ] = 1
    else /*( !each_layer_is_an_ols_flag && ols_mode_idc == 2 ) */
        LayerUsedAsOutputLayerFlag[ i ] = 0
}
for( i = 1; i < TotalNumOlss; i++ )
    if( each_layer_is_an_ols_flag || ols_mode_idc == 0 ) {
        NumOutputLayersInOls[ i ] = 1
        OutputLayerIdInOls[ i ][ 0 ] = vps_layer_id[ i ]
[0251]  for( j = 0; j < i && ( ols_mode_idc == 0 ); j++ )
            NumSubLayersInLayerInOLS[ i ][ j ] = max_tid_il_ref_pics_plus1[ i ]
        NumSubLayersInLayerInOLS[ i ][ i ] = vps_max_sub_layers_minus1 + 1
    } else if( ols_mode_idc == 1 ) {
        NumOutputLayersInOls[ i ] = i + 1
        for( j = 0; j < NumOutputLayersInOls[ i ]; j++ ) {
            OutputLayerIdInOls[ i ][ j ] = vps_layer_id[ j ]
            NumSubLayersInLayerInOLS[ i ][ j ] = vps_max_sub_layers_minus1 + 1
        }
    } else if( ols_mode_idc == 2 ) {
        for( j = 0; j <= vps_max_layers_minus1; j++ ) {
            layerIncludedInOlsFlag[ i ][ j ] = 0
            NumSubLayersInLayerInOLS[ i ][ j ] = 0
        }
        for( k = 0, j = 0; k <= vps_max_layers_minus1; k++ )
```

```
                if( ols_output_layer_flag[ i ][ k ] ) {
                    layerIncludedInOlsFlag[ i ][ k ] = 1
                    LayerUsedAsOutputLayerFlag[ k ] = 1
                    OutputLayerIdx[ i ][ j ] = k
                    OutputLayerIdInOls[ i ][ j++ ] = vps_layer_id[ k ]
                    NumSubLayersInLayerInOLS[ i ][ j ] =
        vps_max_sub_layers_minus1 + 1
                }
            NumOutputLayersInOls[ i ] = j
            for( j = 0; j < NumOutputLayersInOls[ i ]; j++ ) {
[0252]          idx = OutputLayerIdx[ i ][ j ]
                for( k = 0; k < NumRefLayers[ idx ]; k++ ) {
                    layerIncludedInOlsFlag[ i ][ RefLayerIdx[ idx ][ k ] ] = 1
                    if( NumSubLayersInLayerInOLS[ i ][ RefLayerIdx[ idx ][ k ] ] <
                        max_tid_il_ref_pics_plus1[ OutputLayerIdInOls[ i ][ j ] ] )
                      NumSubLayersInLayerInOLS[ i ][ RefLayerIdx[ idx ][ k ] ] =
                        max_tid_il_ref_pics_plus1[ OutputLayerIdInOls[ i ][ j ] ]
                }
            }
        }
```

[0253] 对于在0至vps_max_layers_minus1(包含端值)范围内的i的每个值，LayerUsedAsRefLayerFlag[i]和LayerUsedAsOutputLayerFlag[i]的值可以不都等于0。换句话说，可能不存在既不是至少一个OLS的输出层也不是任何其它层的直接参考层的层。

[0254] 对于每个OLS，可以有至少一个层是输出层。换句话说，对于在0至TotalNumOlss-1(包含端值)范围内的i的任何值，NumOutputLayersInOls[i]的值可以大于或等于1。

[0255] 指示第i个OLS中的层的数量的变量NumLayersInOls[i]和指定第i个OLS中的第j层的nuh_layer_id值的变量LayerIdInOls[i][j]如下导出：

```
NumLayersInOls[ 0 ] = 1
LayerIdInOls[ 0 ][ 0 ] = vps_layer_id[ 0 ]
for( i = 1; i < TotalNumOlss; i++ ) {
    if( each_layer_is_an_ols_flag ) {
        NumLayersInOls[ i ] = 1
        LayerIdInOls[ i ][ 0 ] = vps_layer_id[ i ]
    } else if( ols_mode_idc == 0 || ols_mode_idc == 1 ) {
        NumLayersInOls[ i ] = i + 1
        for( j = 0; j < NumLayersInOls[ i ]; j++ )
            LayerIdInOls[ i ][ j ] = vps_layer_id[ j ]
    } else if( ols_mode_idc == 2 ) {
        for( k = 0, j = 0; k <= vps_max_layers_minus1; k++ )
            if( layerIncludedInOlsFlag[ i ][ k ] )
                LayerIdInOls[ i ][ j++ ] = vps_layer_id[ k ]
        NumLayersInOls[ i ] = j
    }
}
```

[0256]

[0257] 指定nuh_layer_id等于LayerIdInOls[i][j]的层的OLS层索引的变量OlsLayerIdx[i][j]如下导出:

[0258] for(i=0;i<TotalNumOlss;i++)

[0259] for j=0;j<NumLayersInOls[i];j++)

[0260] OlsLayerIdx[i][LayerIdInOls[i][j]]=j

[0261] 每个OLS中的最低层可以是独立层。换句话说,对于在0至TotalNumOlss-1(包含端值)的范围内的每个i,vps_independent_layer_flag[GeneralLayerIdx[LayerIdInOls[i][0]]]的值可以等于1。

[0262] 每个层可以包括在由VPS指定的至少一个OLS中。换句话说,对于nuh_layer_id的特定值nuhLayerId等于vps_layer_id[k]中的一个的每个层(其中,k在0至vps_max_layers_minus1(包含端值)的范围内),可以存在至少一对值i和j,其中,i在0至TotalNumOlss-1(包含端值)的范围内,并且j在0至NumLayersInOls[i]-1(包含端值)的范围内,使得LayerIdInOls[i][j]的值等于nuhLayerId。

[0263] 在实施例中,已编码视频序列中的任何VCL NAL单元的temporalId值可以不大于VCL NAL单元所参考的VPS中的vps_max_sublayers_minus1。

[0264] 在相同或另一实施例中,max_tid_il_ref_pics_plus1[i]具有以下语义:max_tid_il_ref_pics_plus1[i]等于0指示第i层的非IRAP图片不使用层间预测。max_tid_il_ref_pics_plus1[i]大于0指定对于第i层的解码图片,不使用TemporalId大于max_tid_il_

ref_pics_plus1[i]-1的图片作为ILRP。当不存在时,推断max_tid_il_ref_pics_plus1[i]的值等于vps_max_sublayers_minus1+1。

[0265] 在相同或另一实施例中,max_tid_il_ref_pics_plus1[i]可以不大于vps_max_sublayers_minus1+1。

[0266] 在实施例中,可以不为独立层导出逐子层(sublayer-wise)的输出层集合。因此,可以不为独立层导出max_tid_il_ref_pics_plus1和layerIncludedInOlsFlag。

[0267] 在相同或另一实施例中,指示第i个OLS中的输出层的数量的变量NumOutputLayersInOls[i]、指示第i个OLS中的第j层中的子层的数量的变量NumSubLayersInLayerInOLS[i][j]、指示第i个OLS中的第j输出层的nuh_layer_id值的变量OutputLayerIdInOls[i][j]以及指示第k层是否用作至少一个OLS中的输出层的变量LayerUsedAsOutputLayerFlag[k]可以如下导出:

[0268]

```
NumOutputLayersInOls[ 0 ] = 1
OutputLayerIdInOls[ 0 ][ 0 ] = vps_layer_id[ 0 ]
NumSubLayersInLayerInOLS[ 0 ][ 0 ] = vps_max_sub_layers_minus1 + 1
LayerUsedAsOutputLayerFlag[ 0 ] = 1
for( i = 1, i <= vps_max_layers_minus1; i++ ) {
    if( each_layer_is_an_ols_flag || ols_mode_idc < 2 )
        LayerUsedAsOutputLayerFlag[ i ] = 1
    else /*( !each_layer_is_an_ols_flag && ols_mode_idc == 2 ) */
        LayerUsedAsOutputLayerFlag[ i ] = 0
}
for( i = 1; i < TotalNumOlss; i++ )
    if( each_layer_is_an_ols_flag || ols_mode_idc == 0 ) {
        NumOutputLayersInOls[ i ] = 1
        OutputLayerIdInOls[ i ][ 0 ] = vps_layer_id[ i ]
```

```
            for( j = 0; j < i   &&   ( ols_mode_idc   ==   0 ); j++ )
               NumSubLayersInLayerInOLS[ i ][ j ] = max_tid_il_ref_pics_plus1[ i ]
            NumSubLayersInLayerInOLS[ i ][ i ] = vps_max_sub_layers_minus1 + 1
         } else if( ols_mode_idc   ==   1 ) {
            NumOutputLayersInOls[ i ] = i + 1
            for( j = 0; j < NumOutputLayersInOls[ i ]; j++ ) {
               OutputLayerIdInOls[ i ][ j ] = vps_layer_id[ j ]
               NumSubLayersInLayerInOLS[ i ][ j ] = vps_max_sub_layers_minus1 + 1
            }
         } else if( ols_mode_idc   ==   2 ) {
            for( j = 0; j   <=   vps_max_layers_minus1; j++ ) {
               layerIncludedInOlsFlag[ i ][ j ] = 0
               NumSubLayersInLayerInOLS[ i ][ j ] = 0
            }
            for( k = 0, j = 0; k   <=   vps_max_layers_minus1; k++ )
               if( ols_output_layer_flag[ i ][ k ] ) {
```

[0269]

```
                  layerIncludedInOlsFlag[ i ][ k ] = 1
                  LayerUsedAsOutputLayerFlag[ k ] = 1
                  OutputLayerIdx[ i ][ j ] = k
                  OutputLayerIdInOls[ i ][ j] = vps_layer_id[ k ]
                  NumSubLayersInLayerInOLS[ i ][ j++ ] =
vps_max_sub_layers_minus1 + 1
               }

            if( !vps_all_independent_layers_flag ) {
               NumOutputLayersInOls[ i ] = j
               for( j = 0; j < NumOutputLayersInOls[ i ]; j++ ) {
                  idx = OutputLayerIdx[ i ][ j ]
                  if( !vps_independent_layer_flag[ idx ] )
                     for( k = 0; k < NumRefLayers[ idx ]; k++ ) {
                        layerIncludedInOlsFlag[ i ][ RefLayerIdx[ idx ][ k ] ] = 1
                        if( NumSubLayersInLayerInOLS[ i ][ RefLayerIdx[ idx ][ k ] ] <
```

```
        max_tid_il_ref_pics_plus1[ OutputLayerIdInOls[ i ][ j ] ] )
                                    NumSubLayersInLayerInOLS[ i ][ RefLayerIdx[ idx ][ k ] ]
        =

        max_tid_il_ref_pics_plus1[ OutputLayerIdInOls[ i ][ j ] ]
                    }
                }
            }
        }
    }
```

[0270]

[0271] 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7示出了计算机设备700,其适于实现所公开主题的某些实施例。

[0272] 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(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。

[0273] 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。

[0274] 图7所示的用于计算机设备700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设备700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。

[0275] 计算机设备700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)。

[0276] 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701、鼠标702、触控板703、触摸屏710、操纵杆705、麦克风706、扫描仪708、照相机707。

[0277] 计算机设备700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710、数据手套704或操纵杆705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709、耳机(未示出))、视觉输出设备(例如,包括阴极射线管(CRT,cathode ray tube)屏幕、液晶(liquid-crystal display)屏幕、等离子屏幕、有机发光二极管屏(organic light-emitting diode)的屏幕710,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)。

[0278] 计算机设备700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)720或类似介质721的光学介质、拇指驱动器722、可移动硬盘驱动器或固体状态驱动器723,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。

[0279] 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。

[0280] 计算机设备700还可以包括通往一个或多个通信网络的接口。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。网络的示例可以包括以太网、无线局域网、蜂窝网络(全球移动通信系统(GSM,global systems for mobile communications)、3G、4G、5G、长期演进(LTE,Long-Term Evolution)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(749)(例如,计算机设备700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设备700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,计算机设备700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。

[0281] 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设备700的核心740。

[0282] 核心740可包括一个或多个中央处理单元(CPU)741、图形处理单元(GPU,Graphics Processing Units)742、以现场可编程门阵列(FPGA,Field Programmable Gate Areas)743形式的专用可编程处理单元、用于特定任务744的硬件加速器等。这些设备以及只读存储器(ROM)745、随机存取存储器746、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)747等可通过系统总线748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748,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748,或通过外围总线749进行连接。外围总线的架构包括外围组件互连(PCI,peripheral component interconnect)、USB等。

[0283] CPU741、GPU742、FPGA743和加速器744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ROM745或RAM746中。过渡数据也可以存储在RAM746中,而永久数据可以存储在例如内部大容量存储器747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CPU741、GPU742、大容量存储器747、ROM745、RAM746等紧密关联。

[0284] 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。

[0285] 作为实施例而非限制,具有体系结构700的计算机系统,特别是核心740,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740的特定存储器,例如核心内部大容量存储器747或ROM745。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740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740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RAM746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744)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。

[0286] 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。

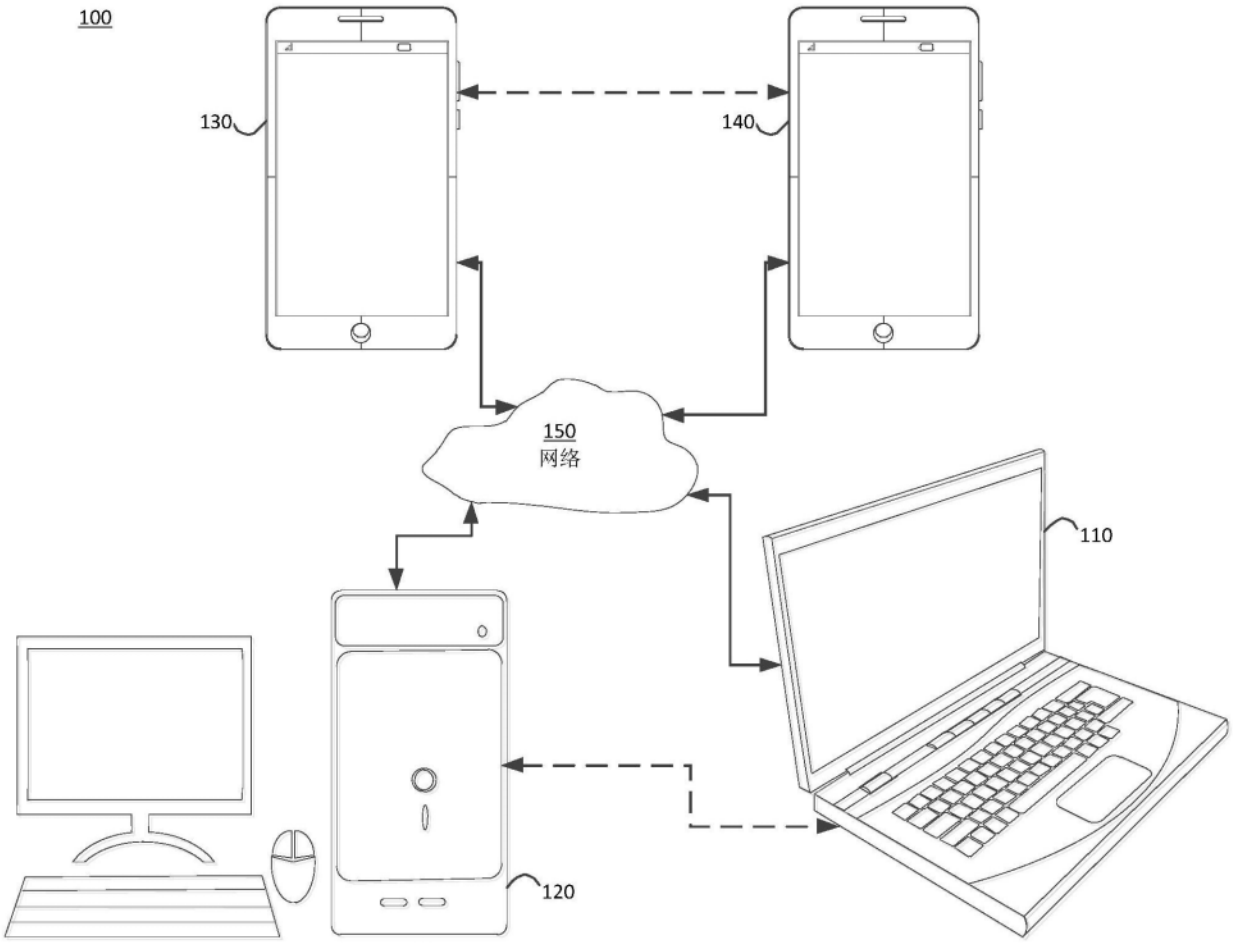


图1

流式传输系统200

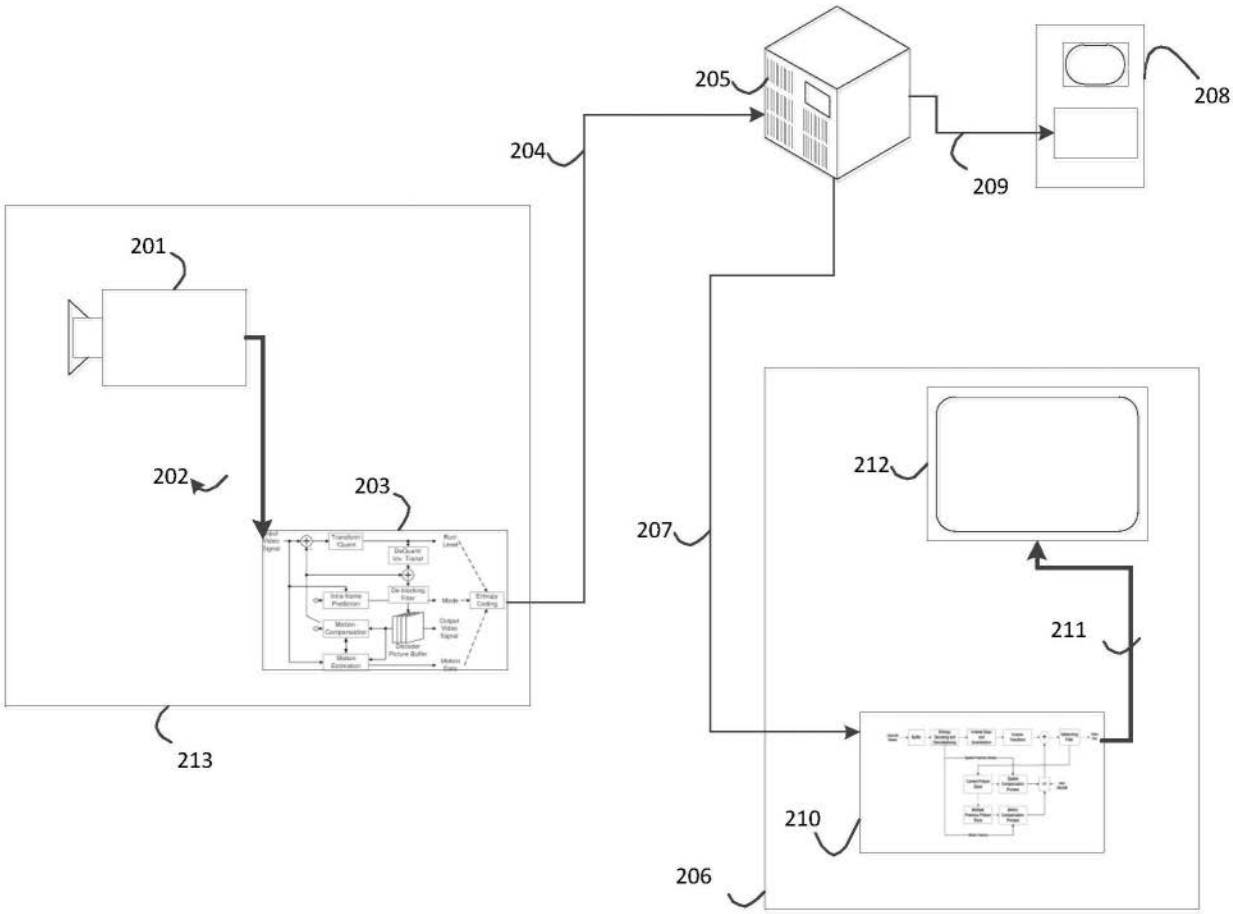


图2

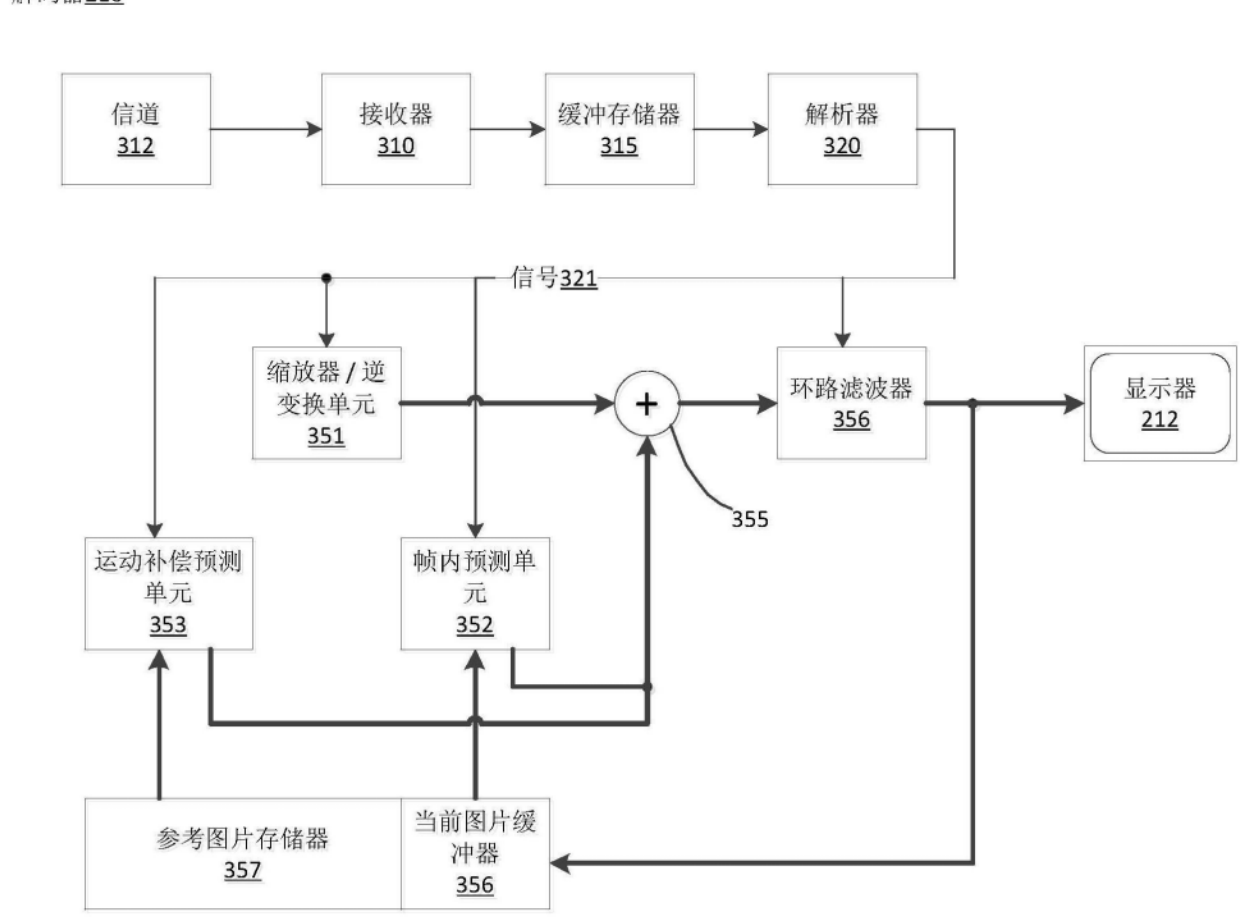

解码器210
信道
312
接收器
310
缓冲存储器
315
解析器
320
信号321
缩放器/逆变换单元
351
环路滤波器
356
显示器
212
355
运动补偿预测单元
353
帧内预测单元
352
参考图片存储器
357
当前图片缓冲器
356

图3

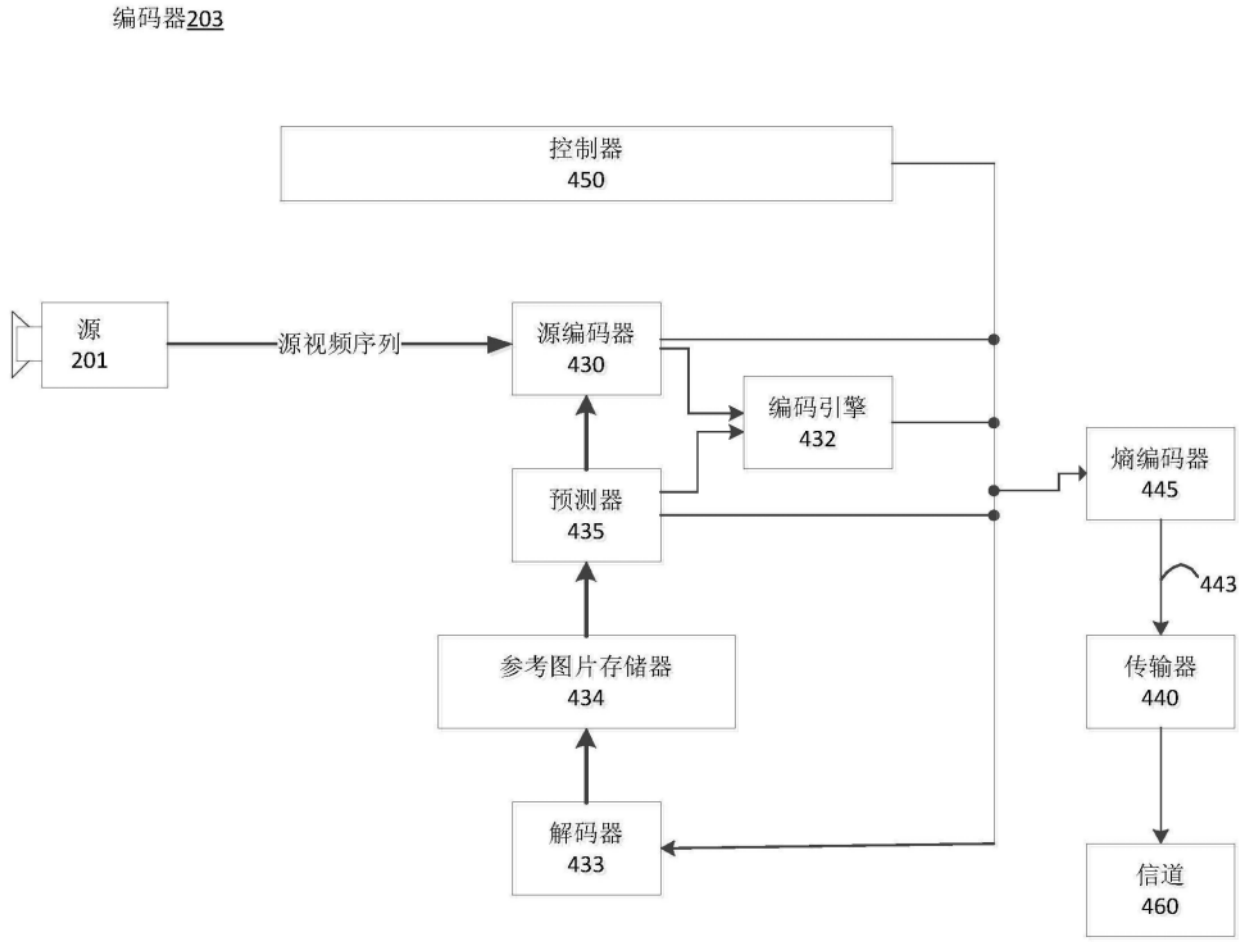

编码器203
控制器
450
源
201
源视频序列
源编码器
430
编码引擎
432
预测器
435
参考图片存储器
434
解码器
433
熵编码器
445
443
传输器
440
信道
460

图4

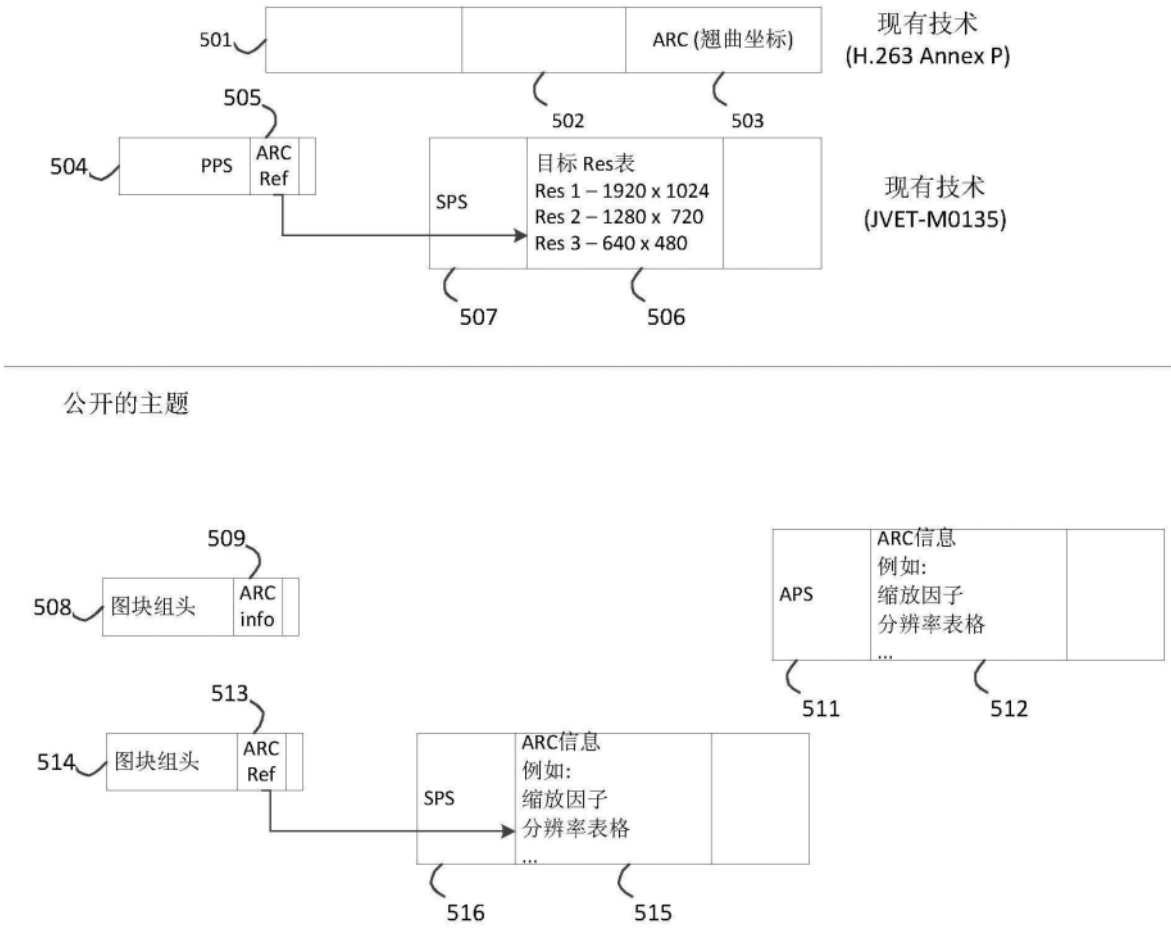

501
ARC (翘曲坐标)
现有技术
(H.263 Annex P)
502
503
505
504
PPS
ARC Ref
SPS
目标 Res表
Res 1 – 1920 x 1024
Res 2 – 1280 x 720
Res 3 – 640 x 480
现有技术
(JVET-M0135)
507
506
公开的主题
509
508
图块组头
ARC info
APS
ARC信息
例如:
缩放因子
分辨率表格
...
511
512
513
514
图块组头
ARC Ref
SPS
ARC信息
例如:
缩放因子
分辨率表格
...
516
515

图5

601
603
602

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

610
611
612
613
614
615
616
617

seq_parameter_set_rbsp( ) {	**描述符**
...	
adaptive_pic_resolution_change_flag	u(1)
if(adaptive_pic_resolution_change_flag) {	
output_pic_width_in_luma_samples	ue(v)
output_pic_height_in_luma_samples	ue(v)
reference_pic_size_present_flag	u(1)
if(reference_pic_size_present_flag)	
{	
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

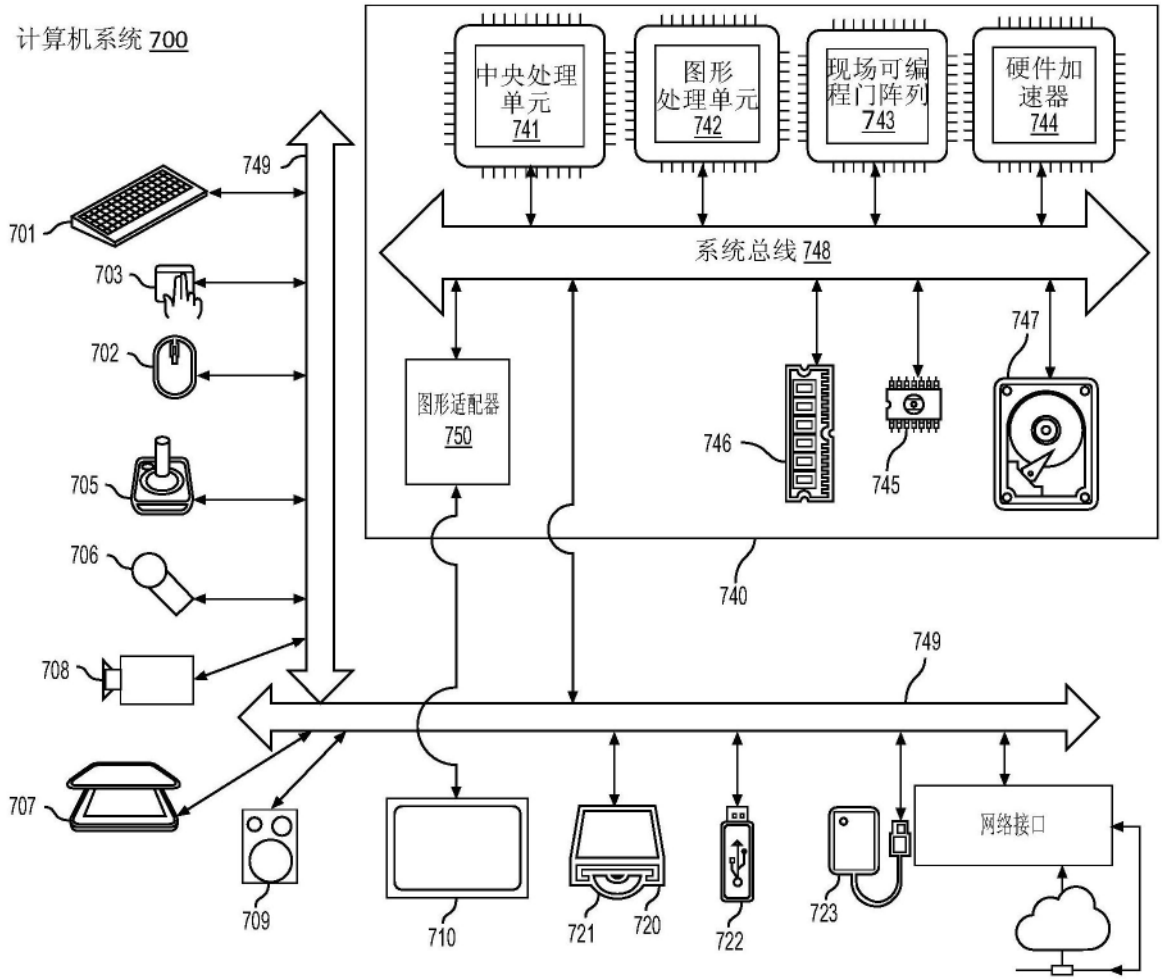

计算机系统 700
中央处理单元 741
图形处理单元 742
现场可编程门阵列 743
硬件加速器 744
系统总线 748
图形适配器 750
740
746
745
747
749
701
703
702
705
706
708
707
709
710
721
720
722
723
网络接口

图7

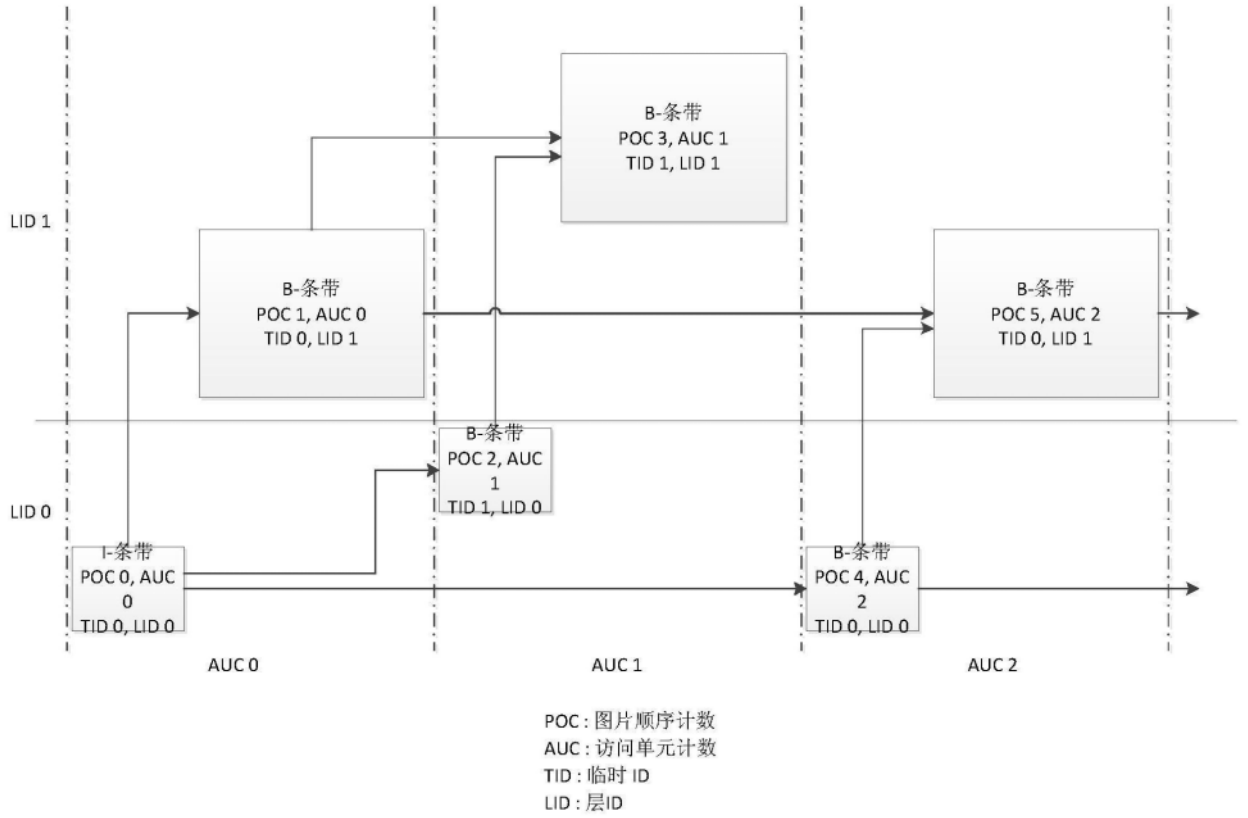

B-条带
POC 3, AUC 1
TID 1, LID 1
LID 1
B-条带
POC 1, AUC 0
TID 0, LID 1
B-条带
POC 5, AUC 2
TID 0, LID 1
B-条带
POC 2, AUC 1
TID 1, LID 0
LID 0
I-条带
POC 0, AUC 0
TID 0, LID 0
B-条带
POC 4, AUC 2
TID 0, LID 0
AUC 0
AUC 1
AUC 2
POC : 图片顺序计数
AUC : 访问单元计数
TID : 临时 ID
LID : 层ID

图8

video_parameter_set_rbsp() {	描述符
vps_video_paramter_set_id	u(4)
vps_max_layers_minus1	u(8)
for (i=0; i <= vps_max_layers_minus 1; i++){	
vps_included_layer_id[i]	u(7)
vps_reserved_zero_bit	u(1)
}	
vps_constraint_info_present_flag	u(1)
vps_constraint_poc_cycle_per_au	u(1)
if(vps_constant_poc_cycle_per_au)	
vps_pos_cycle_au	u(8)
. . .	
}	

slice header( ) {	Descriptor
slice_pic_parameter_set_id	ue(v)
if(rec_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9

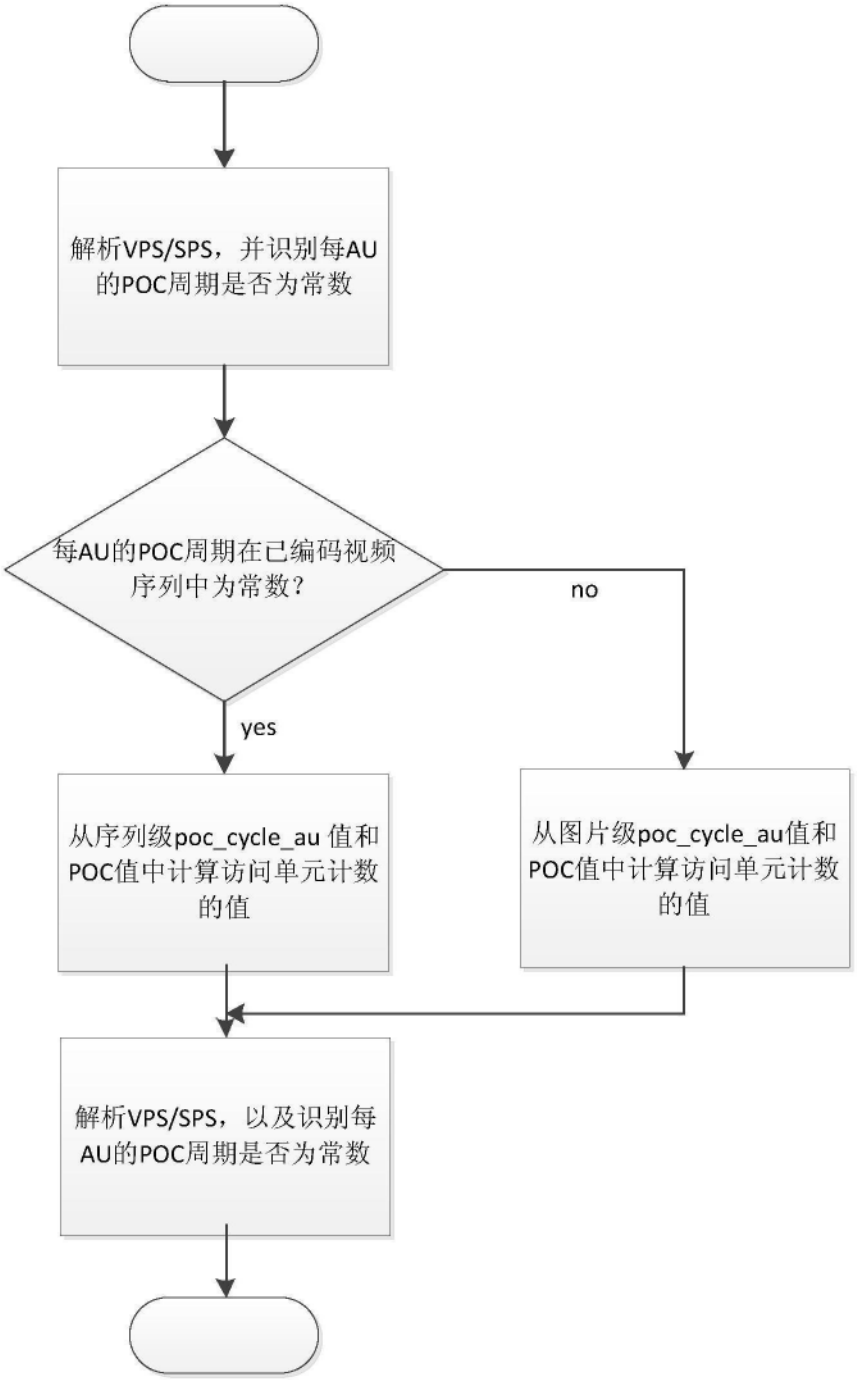

解析VPS/SPS，并识别每AU的POC周期是否为常数
每AU的POC周期在已编码视频序列中为常数？
no
yes
从序列级poc_cycle_au 值和POC值中计算访问单元计数的值
从图片级poc_cycle_au值和POC值中计算访问单元计数的值
解析VPS/SPS，以及识别每AU的POC周期是否为常数

图10


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11

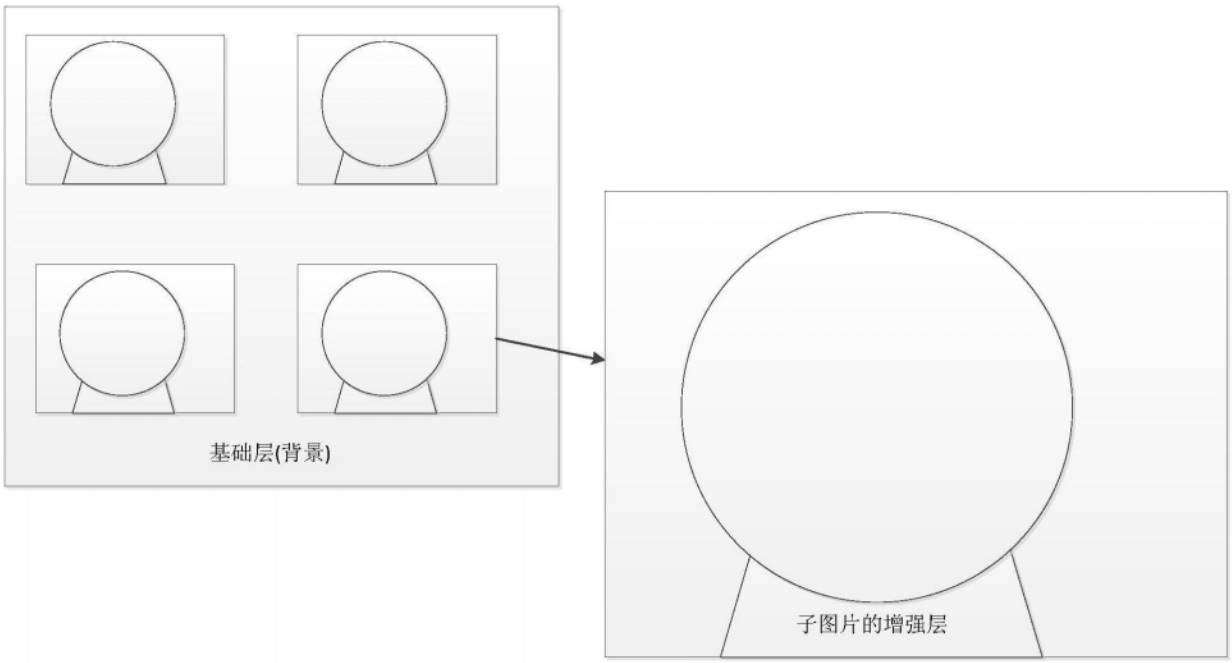


图12

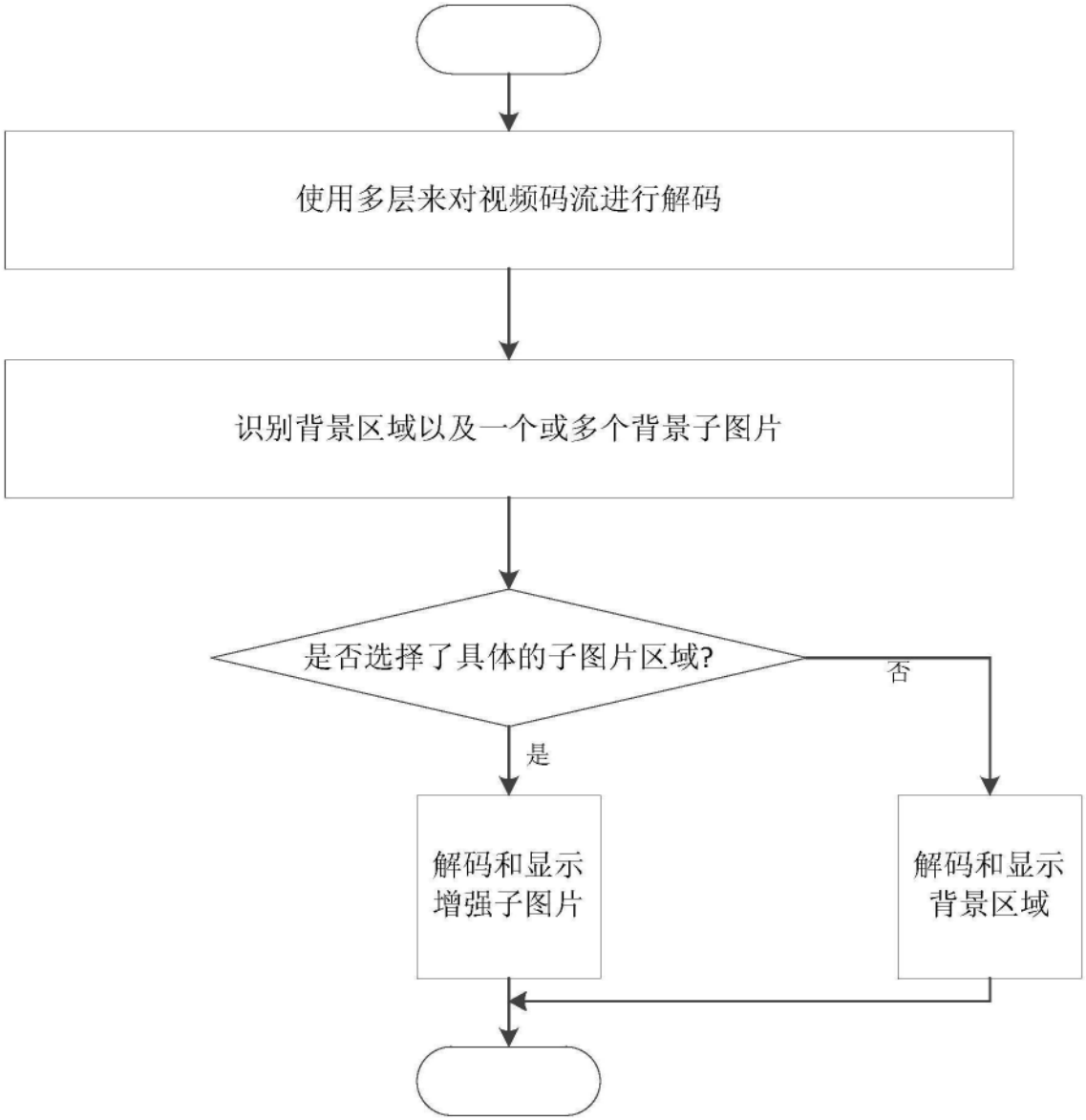

使用多层来对视频码流进行解码
识别背景区域以及一个或多个背景子图片
是否选择了具体的子图片区域?
否
是
解码和显示增强子图片
解码和显示背景区域

图13

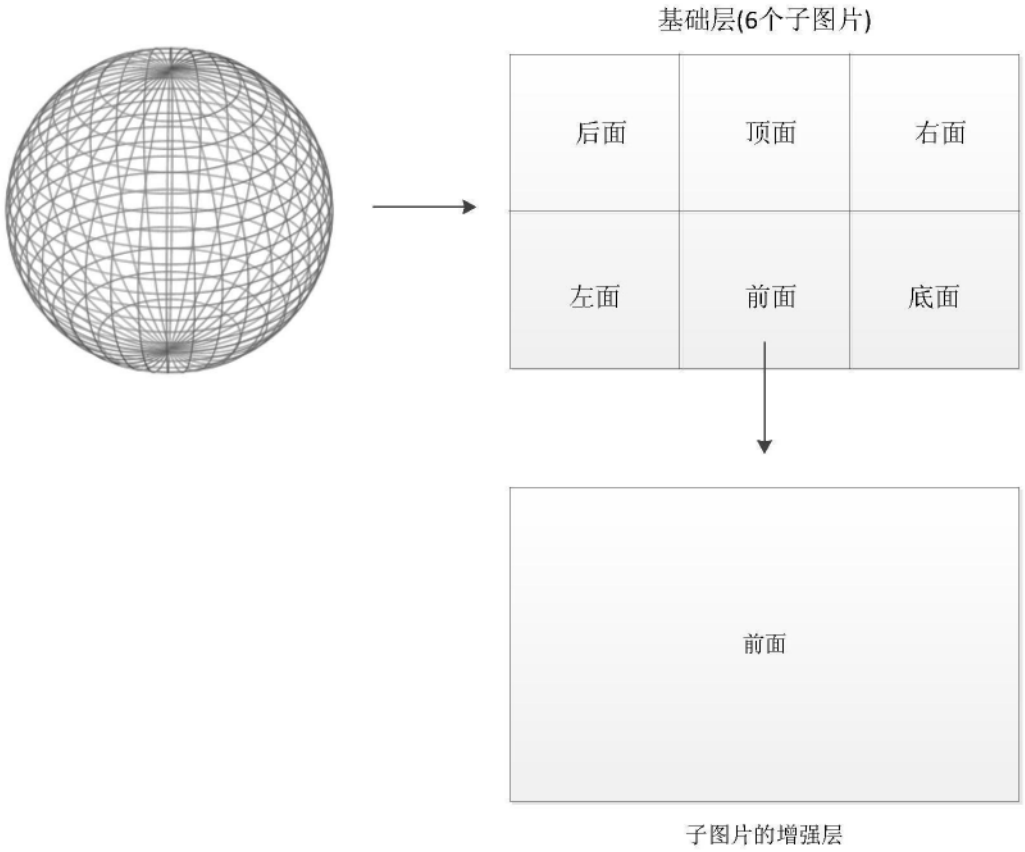


图14

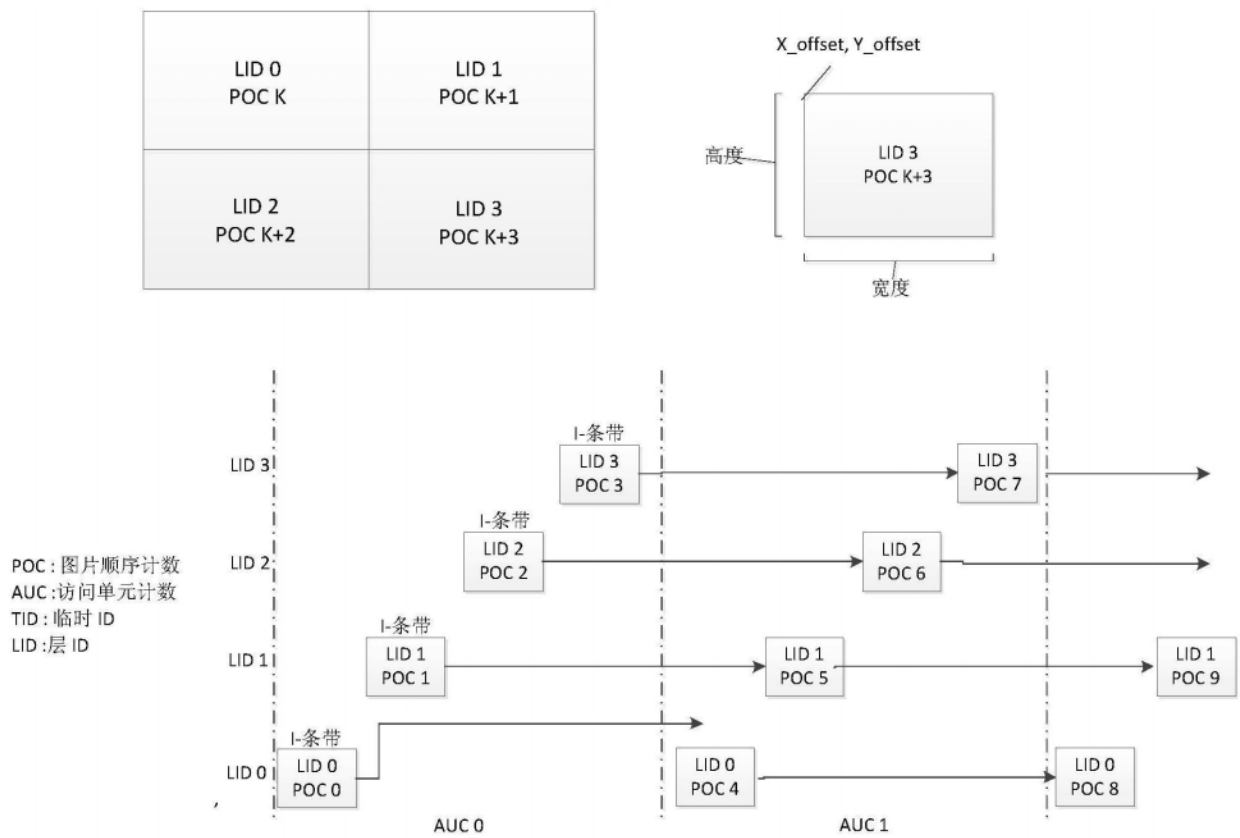

LID 0
POC K
LID 1
POC K+1
LID 2
POC K+2
LID 3
POC K+3
X_offset, Y_offset
高度
宽度
POC : 图片顺序计数
AUC :访问单元计数
TID : 临时 ID
LID :层 ID
LID 3
LID 2
LID 1
LID 0
I-条带
LID 3 POC 3
LID 3 POC 7
I-条带
LID 2 POC 2
LID 2 POC 6
I-条带
LID 1 POC 1
LID 1 POC 5
LID 1 POC 9
I-条带
LID 0 POC 0
LID 0 POC 4
LID 0 POC 8
AUC 0
AUC 1

图15

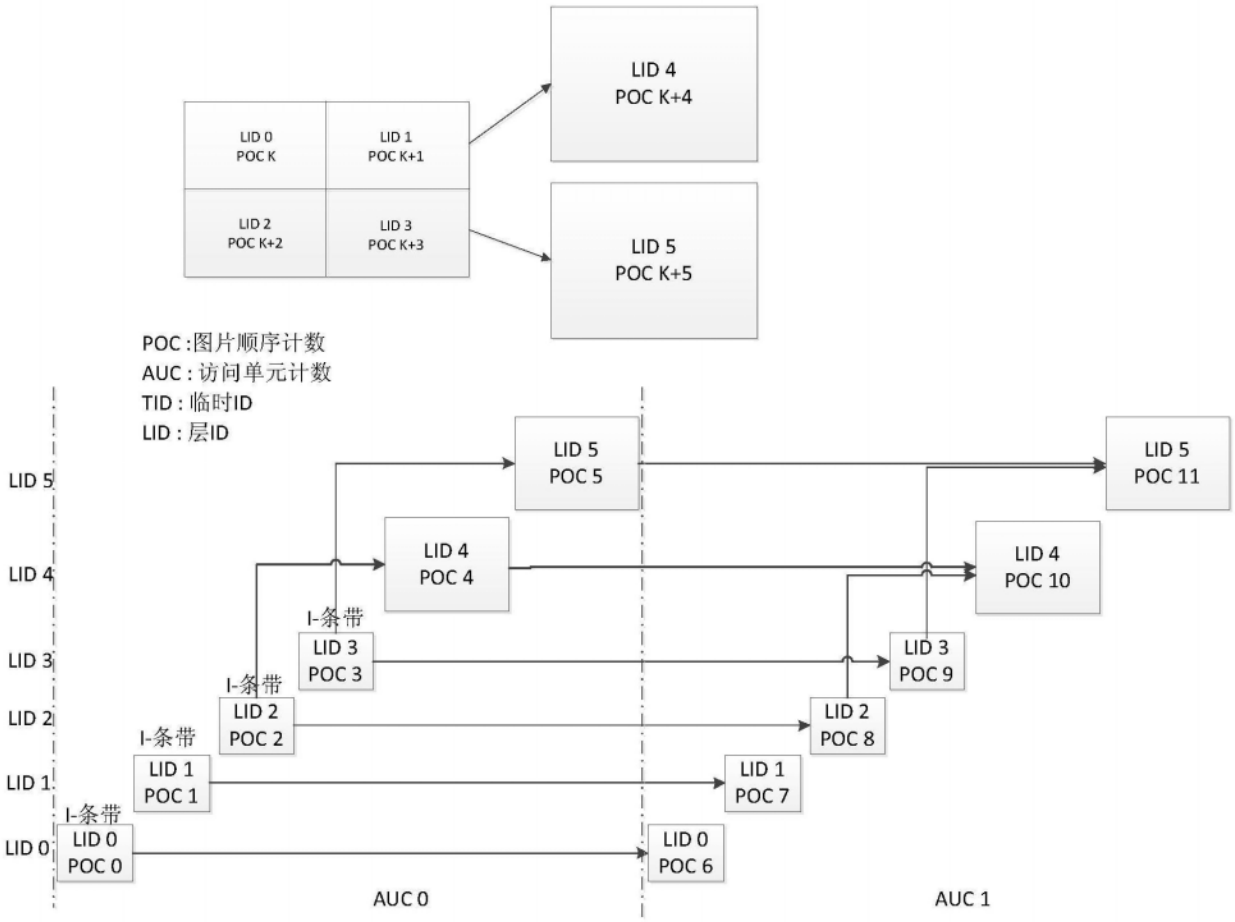

LID 0
POC K
LID 1
POC K+1
LID 2
POC K+2
LID 3
POC K+3
LID 4
POC K+4
LID 5
POC K+5
POC :图片顺序计数
AUC : 访问单元计数
TID : 临时ID
LID : 层ID
LID 5
LID 4
LID 3
LID 2
LID 1
LID 0
I-条带
LID 0
POC 0
LID 1
POC 1
LID 2
POC 2
LID 3
POC 3
LID 4
POC 4
LID 5
POC 5
LID 0
POC 6
LID 1
POC 7
LID 2
POC 8
LID 3
POC 9
LID 4
POC 10
LID 5
POC 11
AUC 0
AUC 1

图16

video_parameter_set_rbsp( ) {	描述符
vps_video_parameter_set_id	u(4)
vps_max_layer_minus1	u(8)
for(i=0; <= 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)
}	
. . .	
}	

seq_parameter_set_rbsp( ){	Descriptor
sps_decoding_parameter_set_id	u(4)
sps_video_parameter_set_id	u(4)
sps_max_sublayers_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17

sub_region_partitiong_info(payloadSize){	描述符
num_sub_region	ue(v)
num_layers	ue(v)
for(i=0; i<= num_layers; i++)	
layer_id[i]	ue(v)
for(i=0; i <= num_layer; i++)	
for(i=0; j < i; j++)	
direct_dependency_flag[i][j]	ue(1)
for(i=0; i< num_sub_region; i++){	
num_layers_for_region[i]	ue(v)
for(i=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18

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)
}	
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20

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_layer[i][j]	ue(v)
for(k = 0; k < num_output_subpic_layer[i][j]; k++)	
sub_pic_id_layer[i][j][k]	u(8)
if(vps_output_layers_mode[i] == 2)	
output_layer_flag[i][j]	u(1)
if(vps_ptl_signal_flag[i])	
profile_tier_level_idx[i][j]	u(v)
}	
}	
…	
}	

图21

video_parameter_set_rbsp( ) {	描述符
vps_video_parameter_set_id	u(4)
vps_max_layers_minus1	u(6)
vps_max_sublayers_minus1	u(3)
if(vps_max_layers_minus1 > 0 && vps_max_sublayers_minus1 > 0)	
vps_all_layers_same_num_sublayers_flag	u(1)
if(vps_max_layers_minus1 > 0)	
vps_all_independent_layers_flag	u(1)
for(i = 0; i <= vps_max_layers_minus1; i++) {	
vps_layer_id[i]	u(6)
if(i > 0 && !vps_all_independent_layers_flag) {	
vps_independent_layer_flag[i]	u(1)
if(!vps_independent_layer_flag[i]) {	
for(j = 0; j < i; j++)	
vps_direct_ref_layer_flag[i][j]	u(1)
max_tid_ref_present_flag[i]	u(1)
if(max_tid_ref_present_flag[i])	
max_tid_il_ref_pics_plus1[i]	u(3)
}	
}	
}	
if(vps_max_layers_minus1 > 0) {	
if(vps_all_independent_layers_flag)	
each_layer_is_an_ols_flag	u(1)
if(!each_layer_is_an_ols_flag) {	
if(!vps_all_independent_layers_flag)	
ols_mode_idc	u(2)
if(ols_mode_idc == 2) {	
num_output_layer_sets_minus1	u(8)
for(i = 1; i <= num_output_layer_sets_minus1; i++)	
for(j = 0; j <= vps_max_layers_minus1; j++)	
ols_output_layer_flag[i][j]	u(1)
}	
}	
}	
…	
}	

图22