



(12) 发明专利申请

(10) 申请公布号 CN 115567707 A

(43) 申请公布日 2023. 01. 03

(21) 申请号 202211142583.2

(22) 申请日 2020.06.01

(66) 本国优先权数据

PCT/CN2019/089218 2019.05.30 CN

(62) 分案原申请数据

202080033241.X 2020.06.01

(71) 申请人 抖音视界有限公司

地址 100041 北京市石景山区实兴大街30
号院3号楼2层B-0035房间

申请人 字节跳动有限公司

(72) 发明人 张莉 张娜 张凯 刘鸿彬 王悦

(74) 专利代理机构 北京市柳沈律师事务所

11105

专利代理师 张亮

(51) Int.Cl.

H04N 19/117 (2014.01)

H04N 19/119 (2014.01)

H04N 19/122 (2014.01)

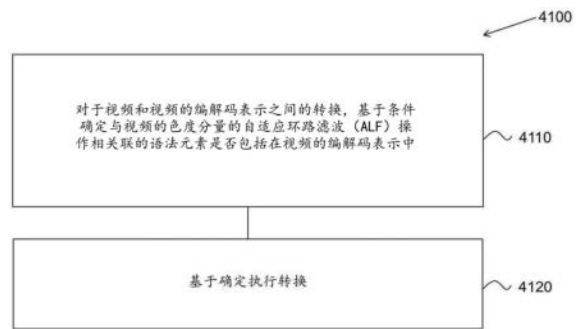
权利要求书4页 说明书91页 附图25页

(54) 发明名称

色度分量的自适应环路滤波

(57) 摘要

描述了与关于视频处理中的色度格式的语法标志的管理相关的装置、系统和方法。在一个代表性方面,一种视频解码方法包括:对于在视频和视频的编解码表示之间的转换,基于条件确定与视频的色度分量的自适应环路滤波(ALF)操作相关联的语法元素是否包括在编解码表示中。该方法还包括根据确定执行转换。



1. 一种处理视频数据的方法,包括:

对于视频的视频区域和所述视频的比特流之间的转换,基于所述视频的色度格式确定与所述视频区域的色度分量的自适应环路滤波操作相关联的至少一个语法元素是否包括在所述比特流中;以及

根据所述确定执行所述转换,

其中,与所述视频区域的色度分量的所述自适应环路滤波操作相对应的自适应参数集的第一网络抽象层单元的第一时域标识符小于或等于所述视频区域的第二网络抽象层单元的第二时域标识符,

其中,所述至少一个语法元素包括与所述视频区域的色度分量的所述自适应环路滤波操作的滤波器系数相关的多个语法元素,并且

其中,所述多个语法元素包括在与所述自适应环路滤波操作对应的所述自适应参数集中。

2. 根据权利要求1所述的方法,其中,所述至少一个语法元素还包括标志,所述标志指示所述视频区域的色度分量的所述自适应环路滤波操作的至少一个系数是否包括在所述比特流中,并且所述标志包括在与所述视频区域的色度分量的所述自适应环路滤波操作相对应的所述自适应参数集中。

3. 根据权利要求2所述的方法,其中,响应于所述标志不存在于所述比特流中,所述标志的值被推断为等于默认值,指示所述视频区域的色度分量的所述自适应环路滤波操作的所述至少一个系数中没有系数被包括在所述比特流中。

4. 根据权利要求1所述的方法,其中,所述至少一个语法元素还包括所述自适应参数集的索引,并且其中,所述索引包括在条带标头或图片标头中,并且所述自适应参数集包括一个或多个自适应环路滤波器的信息。

5. 根据权利要求4所述的方法,其中,包括在所述自适应参数集中的自适应参数集标识符与所述视频区域的色度分量参考的所述自适应参数集的索引相同。

6. 根据权利要求1所述的方法,其中,所述至少一个语法元素还包括第一指示,所述第一指示指示是否对任意色度分量启用所述自适应环路滤波操作。

7. 根据权利要求6所述的方法,其中,所述第一指示包括在条带标头中。

8. 根据权利要求1所述的方法,其中,所述至少一个语法元素还包括与所述色度分量的自适应环路滤波操作的滤波器时域预测相关的第二指示。

9. 根据权利要求1所述的方法,其中,在所述色度格式不等于特定值的情况下,所述至少一个语法元素被包括在所述比特流中。

10. 根据权利要求9所述的方法,其中,所述特定值为4:0:0。

11. 根据权利要求1所述的方法,其中,所述转换包括将所述视频区域编码成所述比特流。

12. 根据权利要求1所述的方法,其中,所述转换包括从所述比特流中解码所述视频区域。

13. 一种处理视频数据的装置,包括处理器和其上具有指令的非暂时性存储器,其中所述指令在由所述处理器执行时使处理器:

对于视频的视频区域和所述视频的比特流之间的转换,基于所述视频的色度格式确定

与所述视频区域的色度分量的自适应环路滤波操作相关联的至少一个语法元素是否包括在所述比特流中;以及

根据所述确定执行所述转换,

其中,与所述视频区域的色度分量的所述自适应环路滤波操作相对应的自适应参数集的第一网络抽象层单元的第一时域标识符小于或等于所述视频区域的第二网络抽象层单元的第二时域标识符,

其中,所述至少一个语法元素包括与所述视频区域的色度分量的所述自适应环路滤波操作的滤波器系数相关的多个语法元素,并且

其中,所述多个语法元素包括在与所述自适应环路滤波操作对应的所述自适应参数集中。

14. 根据权利要求13所述的装置,其中,在所述色度格式不等于4:0:0的情况下,在所述比特流中包括所述至少一个语法元素,

其中,所述至少一个语法元素还包括所述自适应参数集的索引,其中所述索引包括在条带标头或图片标头中和在所述自适应参数集中,所述自适应参数集包括一个或多个自适应环路滤波器的信息,并且

其中,所述自适应参数集中包括的自适应参数集标识符与所述视频区域的色度分量参考的所述自适应参数集的索引相同。

15. 根据权利要求13所述的装置,其中,所述至少一个语法元素还包括标志,所述标志指示所述视频区域的色度分量的所述自适应环路滤波操作的至少一个系数是否包括在所述比特流中,并且所述标志包括在与所述视频区域的色度分量的所述自适应环路滤波操作相对应的所述自适应参数集中,并且,

其中,响应于所述标志不存在于所述比特流中,所述标志的值被推断为等于默认值,指示所述视频区域的色度分量的所述自适应环路滤波操作的所述至少一个系数中没有有一个系数被包括在所述比特流中。

16. 一种存储指令的非暂时性计算机可读存储介质,所述指令使处理器:

对于视频的视频区域和所述视频的比特流之间的转换,基于所述视频的色度格式确定与所述视频区域的色度分量的自适应环路滤波操作相关联的至少一个语法元素是否包括在所述比特流中;以及

根据所述确定执行所述转换,

其中,与所述视频区域的色度分量的所述自适应环路滤波操作相对应的自适应参数集的第一网络抽象层单元的第一时域标识符小于或等于所述视频区域的第二网络抽象层单元的第二时域标识符,

其中,所述至少一个语法元素包括与所述视频区域的色度分量的所述自适应环路滤波操作的滤波器系数相关的多个语法元素,并且

其中,所述多个语法元素包括在与所述自适应环路滤波操作对应的所述自适应参数集中。

17. 根据权利要求16所述的非暂时性计算机可读存储介质,其中,在所述色度格式不等于4:0:0的情况下,在所述比特流中包括所述至少一个语法元素,

其中,所述至少一个语法元素还包括所述自适应参数集的索引,其中所述索引包括在

条带标头或图片标头中和在所述自适应参数集中,所述自适应参数集包括一个或多个自适应环路滤波器的信息,并且

其中,所述自适应参数集中包括的自适应参数集标识符与所述视频区域的色度分量参考的所述自适应参数集的索引相同。

18. 根据权利要求16所述的非暂时性计算机可读存储介质,其中,所述至少一个语法元素还包括标志,所述标志指示所述视频区域的色度分量的所述自适应环路滤波操作的至少一个系数是否包括在所述比特流中,并且所述标志包括在与所述视频区域的色度分量的所述自适应环路滤波操作相对应的所述自适应参数集中,并且,

其中,响应于所述标志不存在于所述比特流中,所述标志的值被推断为等于默认值,指示所述视频区域的色度分量的所述自适应环路滤波操作的所述至少一个系数中没有有一个系数被包括在所述比特流中。

19. 一种存储由视频处理装置执行的方法生成的视频的比特流的非暂时性计算机可读存储介质,其中,所述方法包括:

对于视频的视频区域,基于所述视频的色度格式确定与所述视频区域的色度分量的自适应环路滤波操作相关联的至少一个语法元素是否包括在所述比特流中;以及

根据所述确定生成所述比特流,

其中,与所述视频区域的色度分量的所述自适应环路滤波操作相对应的自适应参数集的第一网络抽象层单元的第一时域标识符小于或等于所述视频区域的第二网络抽象层单元的第二时域标识符,

其中,所述至少一个语法元素包括与所述视频区域的色度分量的所述自适应环路滤波操作的滤波器系数相关的多个语法元素,并且

其中,所述多个语法元素包括在与所述自适应环路滤波操作对应的所述自适应参数集中。

20. 根据权利要求19所述的非暂时性计算机可读存储介质,其中,在所述色度格式不等于4:0:0的情况下,在所述比特流中包括所述至少一个语法元素,

其中,所述至少一个语法元素还包括所述自适应参数集的索引,其中所述索引包括在条带标头或图片标头中和在所述自适应参数集中,所述自适应参数集包括一个或多个自适应环路滤波器的信息,并且

其中,所述自适应参数集中包括的自适应参数集标识符与所述视频区域的色度分量参考的所述自适应参数集的索引相同。

21. 一种存储视频的比特流的方法,包括:

对于视频的视频区域,基于所述视频的色度格式确定与所述视频区域的色度分量的自适应环路滤波操作相关联的至少一个语法元素是否包括在所述比特流中;

根据所述确定生成所述比特流;以及

将所述比特流存储到非暂时性计算机可读存储介质中,

其中,与所述视频区域的色度分量的所述自适应环路滤波操作相对应的自适应参数集的第一网络抽象层单元的第一时域标识符小于或等于所述视频区域的第二网络抽象层单元的第二时域标识符,

其中,所述至少一个语法元素包括与所述视频区域的色度分量的所述自适应环路滤波

操作的滤波器系数相关的多个语法元素,并且

其中,所述多个语法元素包括在与所述自适应环路滤波操作对应的所述自适应参数集中。

色度分量的自适应环路滤波

[0001] 相关申请的交叉引用

[0002] 本申请是申请日为2020年6月1日、申请号为202080033241.X、发明名称为“色度分量的自适应环路滤波”的发明专利申请的分案申请。

技术领域

[0003] 本专利文件涉及视频编解码技术、装置和系统。

背景技术

[0004] 目前,正在努力改善当前视频编解码技术的性能,以提供更好的压缩比,或提供允许较低复杂度的视频编解码和解码方案或并行的实现方式。业内专家最近提出了一些新的视频编解码工具,目前正在进行测试,以确定其有效性。

发明内容

[0005] 描述了与数字视频编解码相关的装置、系统和方法,特别是与关于色度格式的语法标志的管理相关的装置、系统和方法。所描述的方法可应用于现有的视频编解码标准(例如,高效视频编解码(HEVC)或多功能视频编解码)和未来的视频编解码标准或视频编解码。

[0006] 在一个代表性方面,所公开的技术可用于提供用于视频处理的方法。该方法包括:为视频和视频的编解码表示之间的转换,基于条件确定与视频的色度分量的自适应环路滤波(ALF)操作相关联的语法元素是否包括在编解码表示中。该方法还包括根据确定执行转换。

[0007] 在另一个代表性方面,所公开的技术可用于提供用于视频处理的方法。该方法包括,对于视频和视频的编解码表示之间的转换,基于条件确定与视频的色度分量的编解码技术相关联的语法元素是否包括在编解码表示中。该方法还包括根据确定执行转换。

[0008] 在另一个代表性方面,所公开的技术可用于提供用于视频处理的方法。该方法包括:对于视频和视频的编解码表示之间的转换,基于与视频的色度格式相关联的条件,确定表示视频的色度分量的特性的语法元素是否包括在编解码表示中。该方法还包括根据确定执行转换。

[0009] 在另一个代表性方面,所公开的技术可用于提供视频解码的方法。该方法包括基于规则确定适用于视频的区域的一个或多个语法标志是否包括在视频的编解码表示的语法结构中。该一个或多个语法标志指示用于在编解码表示中表示区域的联合色度残差编解码步骤。该方法还包括通过根据编解码表示中的一个或多个语法标志的存在或不存在而解析编解码表示,来生成该区域的一个或多个解码视频块。

[0010] 在一个代表性方面,所公开的技术可用于提供用于视频编码的方法。该方法包括基于规则对视频的编解码表示的语法结构中适用于视频的区域的一个或多个语法标志进行条件编码。该一个或多个语法标志指示用于在编解码表示中表示区域的联合色度残差编解码步骤。

[0011] 在另一个代表性方面,所公开的技术可用于提供用于视频解码的方法。该方法包括确定视频的区域的颜色格式,以及基于该颜色格式确定适用于视频的区域的一个或多个语法标志是否包括在视频的编解码表示的语法结构中。该一个或多个语法标志指示使用量化参数偏移来表示编解码表示中的区域。该方法还包括通过根据一个或多个语法标志的存在或不存在而解析编解码表示,来生成视频区域的一个或多个解码视频块。

[0012] 在另一个代表性方面,所公开的技术可用于提供用于视频解码的方法。该方法包括确定视频区域的颜色格式,以及基于该颜色格式确定在视频的编解码表示的语法结构中不存在一个或多个语法标志,指示适用于视频的区域的颜色二次变换的使用。在去量化步骤和逆一次变换之间应用二次变换。该方法还包括通过根据不存在一个或多个语法标志而解析编解码表示,来生成视频区域的一个或多个解码视频块。

[0013] 在另一个代表性方面,所公开的技术可用于提供用于视频编码的方法。该方法包括确定与视频的区域相关联的颜色格式,并基于该颜色格式对视频的编解码表示的语法结构中的一个或多个语法标志进行条件编码。该一个或多个语法标志指示启用用于在编解码表示中表示区域的量化参数偏移。

[0014] 在另一个代表性方面,所公开的技术可用于提供用于视频编码的方法。该方法包括确定视频区域的颜色格式,以及基于该颜色格式,通过在视频的编解码表示的语法结构中不包括指示二次变换的使用的一个或多个语法标志的情况下对该区域进行编码来生成视频的编解码表示。在正向一次变换和量化步骤之间应用二次变换。

[0015] 在另一个代表性方面,所公开的技术可用于提供用于视频解码的方法。该方法包括基于包括区域的视频的颜色格式来确定在视频的编解码表示的语法结构中是否缺少适用于视频的区域的一个或多个语法标志。该一个或多个语法标志指示用于在编解码表示中表示区域的依赖于亮度的颜色残差缩放编解码步骤的使用。依赖于亮度的颜色残差缩放编解码步骤包括缩放过程,其中基于重构的亮度样本推导出缩放系数,该缩放过程应用于颜色残差。该方法还包括通过根据确定而解析编解码表示来生成视频区域的一个或多个解码视频块。

[0016] 在另一个代表性方面,所公开的技术可用于提供用于视频解码的方法。该方法包括基于包括区域的视频的颜色格式来确定在视频的编解码表示的语法结构中不存在适用于视频的区域的一个或多个语法标志。该一个或多个语法标志指示用于表示颜色分量的每个脉冲码调制样本值的比特数。该方法还包括通过根据确定而解析编解码表示来生成视频区域的一个或多个解码视频块。

[0017] 在另一个代表性方面,所公开的技术可用于提供用于视频解码的方法。该方法包括基于包括区域的视频的颜色格式来确定适用于视频的区域的一个或多个语法标志是否包括在视频的编解码表示的语法结构中。该一个或多个语法标志与颜色分量的自适应环路滤波器(ALF)时域预测步骤相关。该方法还包括通过根据确定而解析编解码表示来生成视频区域的一个或多个解码视频块。

[0018] 在另一个代表性方面,所公开的技术可用于提供视频编码的方法。该方法包括确定包括区域的视频的颜色格式;以及基于该颜色格式,通过对区域进行编码而在视频的编解码表示的语法结构中不包括一个或多个语法标志的情况下来生成视频的编解码表示。该一个或多个语法标志指示用于在编解码表示中表示区域的依赖于亮度的颜色残差缩放编

解码步骤的使用。依赖于亮度的色度残差缩放编解码步骤包括缩放过程,其中基于重构的亮度样本推导出缩放系数,并且将缩放过程应用于色度残差。

[0019] 在另一个代表性方面,所公开的技术可用于提供视频编码的方法。该方法包括确定包括区域的视频的色度格式,以及基于该色度格式,通过对区域进行编码而在视频的编解码表示的语法结构中不包括一个或多个语法标志的情况下生成视频的编解码表示。该一个或多个语法标志指示用于表示色度分量的每个脉冲码调制样本值的比特数。

[0020] 在另一个代表性方面,所公开的技术可用于提供视频编码的方法。该方法包括确定包括区域的视频的色度格式,以及基于该色度格式对视频的编解码表示的语法结构中的一个或多个语法标志进行条件编码。该一个或多个语法标志与色度分量的自适应环路滤波器(ALF)时域预测步骤相关。

[0021] 在另一个代表性方面,所公开的技术可用于提供用于视频处理的方法。该方法包括:对于视频的块和视频的比特流表示之间的转换,确定对应于块的多个色度色彩分量的多个约化的二次变换(RST)矩阵。二次变换适用于正向一次变换与量化步骤之间或去量化步骤与逆一次变换之间。该方法还包括基于确定执行转换。

[0022] 在另一个代表性方面,所公开的技术可用于提供用于视频处理的方法。该方法包括:对于视频的块和视频的比特流表示之间的转换,在对块应用联合色度残差编解码步骤的情况下,确定在比特流表示中不存在用于二次变换的一个或多个矩阵索引。二次变换可适用于正向一次变换与量化步骤之间或去量化步骤与逆一次变换之间的块。联合色度残差编解码步骤包括确定作为与色度分量相关联的残差的平均值的联合残差。该方法还包括基于确定执行转换。

[0023] 在另一个代表性方面,所公开的技术可用于提供视频处理的方法。该方法包括:对于视频的块和视频的比特流表示之间的转换,在正向一次变换和量化步骤之间或去量化步骤和逆一次变换之间对块应用二次变换的情况下,确定在比特流表示中不存在指示联合色度残差编解码步骤的使用的语法标志。联合色度残差编解码步骤包括确定作为与色度分量相关联的残差的平均值的联合残差。该方法还包括基于确定执行转换。

[0024] 在另一个代表性方面,所公开的技术可用于提供视频处理的方法。该方法包括:对于视频的块和视频的比特流表示之间的转换,在对块应用联合色度残差编解码步骤的情况下,确定在比特流表示中不存在指示使用跨分量线性模型编解码步骤的语法标志。联合色度残差编解码步骤包括确定作为与色度分量相关联的残差的平均值的联合残差。该方法还包括基于确定执行转换。

[0025] 在另一个代表性方面,所公开的技术可用于提供视频处理的方法。该方法包括:对于视频的块和视频的比特流表示之间的转换,在对块应用跨分量线性模型的情况下,确定在比特流表示中不存在指示使用联合色度残差编解码步骤的语法标志。联合色度残差编解码步骤包括确定作为与色度分量相关联的残差的平均值的联合残差。该方法还包括基于确定执行转换。

[0026] 在另一个代表性方面,所公开的技术可用于提供视频处理的方法。该方法包括:对于视频的块和视频的比特流表示之间的转换,基于块的跨分量线性模型编解码步骤的系数,确定在联合色度残差编解码步骤中使用的两个色度残差之间的相关性。联合色度残差编解码步骤包括确定作为与色度分量相关联的残差的平均值的联合残差。该方法还包括基

于确定执行转换。

[0027] 在另一个代表性方面,所公开的技术可用于提供视频处理的方法。该方法包括:对于视频的块和视频的比特流表示之间的转换,确定在对块应用联合色度残差编解码步骤的情况下,禁用依赖于亮度的色度残差缩放编解码步骤。依赖于亮度的色度残差缩放编解码步骤包括缩放过程,其中基于重构的亮度样本推导出缩放系数,该缩放过程应用于色度残差。该方法还包括基于确定执行转换。

[0028] 在另一个代表性方面,所公开的技术可用于提供视频处理的方法。该方法包括:对于视频的块和视频的编解码表示之间的转换,基于与视频的编解码特性相关的规则,确定在编解码表示中的语法结构中是否包括与指示存在非零系数的块相关联的语法标志。该方法还包括根据确定执行转换。

[0029] 在另一个代表性方面,所公开的技术可用于提供视频处理的方法。该方法包括:对于视频的块和视频的编解码表示之间的转换,基于与视频的编解码特性相关联的规则,确定关于具有降低维度的二次变换的信息。二次变换可适用于正向一次变换和量化步骤之间,或者适用于去量化步骤和逆一次变换之间,并且基于块的维度来减小降低的维度。该方法还包括根据确定执行转换。

[0030] 在一个代表性方面,所公开的技术可用于提供视频处理的方法。该方法包括在当前视频块和当前视频块的比特流表示之间执行转换,其中该转换包括响应于检测到至少一个条件,选择性地启用或禁用与以下中的一个或多个的使用相关联的一个或多个语法标志的信令通知:联合色度残差编解码步骤、依赖于亮度的色度残差缩放(LDCRS)步骤、二次变换步骤、量化步骤或自适应环路滤波器步骤。

[0031] 在另一个代表性方面,所公开的技术可用于提供视频处理的方法。该方法包括在当前视频块和当前视频块的比特流表示之间执行转换,其中该转换包括响应于检测到至少一个条件,选择性地启用或禁用与以下中的一个或多个的使用相关联的一个或多个语法标志的信令通知:联合色度残差编解码步骤、依赖于亮度的色度残差缩放(LDCRS)步骤、二次变换步骤、量化步骤或自适应环路滤波器步骤,其中,当前视频块的第一色度分量与第一约化的二次变换(RST)矩阵相关联,并且当前视频块的第二色度分量与第二RST矩阵相关联。

[0032] 在又一个代表性方面,所公开的技术可用于提供视频处理的方法。该方法包括在当前视频块和当前视频块的比特流表示之间执行转换,其中该转换包括:响应于检测到至少一个条件,选择性地启用或禁用与以下中的一个或多个的使用相关联的一个或多个语法标志的信令通知:联合色度残差编解码步骤、依赖于亮度的色度剩余残差(LDCRS)步骤、二次变换步骤、量化步骤或自适应环路滤波器步骤,其中当前视频块的第一色度分量与第一约化的二次变换(RST)矩阵相关联,并且当前视频块的第二色度分量与第二RST矩阵相关联;以及响应于确定为当前视频块启用联合色度残差编解码步骤,禁用与在当前视频块的一个或多个色度分量上使用CCLM相关的标志的信令通知,其中一个或多个语法标志中的标志与跨分量线性模型(CCLM)的使用相关。

[0033] 在再一个代表性方面,所公开的技术可用于提供视频处理的方法。该方法包括:为当前视频块作出关于在当前视频块的比特流表示中选择性包括一个或多个编解码块标志的决定;以及基于该决定执行当前视频块和当前视频块的比特流表示之间的转换,其中,该选择性包括是基于色彩格式、当前视频块的分量编解码方法或当前视频块的编解码模式。

[0034] 在再一个代表性方面,所公开的技术可用于提供视频处理的方法。该方法包括:对于当前视频块,基于当前视频块的一个或多个色彩分量中的多个非零系数,作出关于将约化的二次变换(RST)选择性地应用于当前视频块的决定;以及根据该决定执行当前视频块和当前视频块的比特流表示之间的转换。

[0035] 在再一个代表性方面,公开了一种视频处理方法。该方法包括在当前视频块和当前视频块的比特流表示之间执行转换,其中,比特流表示是基于依据色度的特性选择性包括指示关于色度编解码的信息的语法元素。

[0036] 此外,在一代表性方面,公开了一种视频系统中的设备,该视频系统包括处理器和在其上具有指令的非暂时性存储器。由处理器执行的指令使处理器实现所公开的方法中的任何一个或多个。

[0037] 此外,公开了一种在非暂时性计算机可读介质上存储的计算机程序产品,该计算机程序产品包括用于执行所公开的方法中的任何一个或多个的程序代码。

[0038] 在附图、说明书和权利要求书中更详细地描述了上述和其他方面以及所公开技术的特征。

附图说明

[0039] 图1示出了VVC的示例编码器框图。

[0040] 图2示出了用于亮度分量的滤波器的示例。

[0041] 图3示出了基于几何变换的自适应环路滤波器(GALF)的编码器决策的示例流程图。

[0042] 图4A-4D示出了子采样拉普拉斯计算的示例。

[0043] 图5示出了示例解码器框图。

[0044] 图6示出了二次变换的示例。

[0045] 图7示出了约化的二次变换(RST)的示例。

[0046] 图8示出了4:2:0格式的图片中的亮度和色度样本的标称位置的示例。

[0047] 图9示出了4:2:2格式的图片中亮度和色度样本的标称位置的示例。

[0048] 图10示出了4:4:4格式的图片中亮度和色度样本的标称位置的示例。

[0049] 图11示出了亮度和色度子采样方法的示例。

[0050] 图12示出了左上色度样本的示例相对位置。

[0051] 图13示出了左上色度样本的另一示例相对位置。

[0052] 图14示出了子块变换(SBT)模式、SBT-V和SBT-H的示例。

[0053] 图15示出了SBT模式SBT-Q的示例。

[0054] 图16是用于实现本文档中描述的视觉媒体解码或视觉媒体编码技术的硬件平台的示例的框图。

[0055] 图17示出了用于视频编解码的示例方法的流程图。

[0056] 图18是帧内块复制模式的图示。

[0057] 图19示出了以调色板模式编解码的块的示例。

[0058] 图20示出了水平和垂直扫描的示例。

[0059] 图21示出了调色板索引的编解码的示例。

- [0060] 图22是其中可实现所公开的技术的示例视频处理系统的框图。
- [0061] 图23是根据本技术的用于视频解码的方法的流程图表示。
- [0062] 图24是根据本技术的用于视频编码的方法的流程图表示。
- [0063] 图25是根据本技术的用于视频解码的另一方法的流程图表示。
- [0064] 图26是根据本技术的用于视频解码的另一方法的流程图表示。
- [0065] 图27是根据本技术的用于视频编码的另一方法的流程图表示。
- [0066] 图28是根据本技术的用于视频编码的另一方法的流程图表示。
- [0067] 图29A是根据本技术的用于视频解码的另一方法的流程图表示。
- [0068] 图29B是根据本技术的用于视频编码的另一方法的流程图表示。
- [0069] 图30A是根据本技术的用于视频解码的另一方法的流程图表示。
- [0070] 图30B是根据本技术的用于视频编码的另一方法的流程图表示。
- [0071] 图31A是根据本技术的用于视频解码的另一方法的流程图表示。
- [0072] 图31B是根据本技术的用于视频编码的另一方法的流程图表示。
- [0073] 图32是根据本技术的用于视频处理的方法的流程图表示。
- [0074] 图33是根据本技术的用于视频处理的另一方法的流程图表示。
- [0075] 图34是根据本技术的用于视频处理的另一方法的流程图表示。
- [0076] 图35是根据本技术的用于视频处理的另一方法的流程图表示。
- [0077] 图36是根据本技术的用于视频处理的另一方法的流程图表示。
- [0078] 图37是根据本技术的用于视频处理的另一方法的流程图表示。
- [0079] 图38是根据本技术的用于视频处理的又一方法的流程图表示。
- [0080] 图39是根据本技术的用于视频处理的又一方法的流程图表示。
- [0081] 图40是根据本技术的用于视频处理的另一方法的流程图表示。
- [0082] 图41是根据本技术的用于视频处理的另一方法的流程图表示。
- [0083] 图42是根据本技术的用于视频处理的另一方法的流程图表示。
- [0084] 图43是根据本技术的用于视频处理的又一方法的流程图表示。

具体实施方式

[0085] 1. HEVC/H.265中的视频编解码

[0086] 视频编解码标准主要通过众所周知的ITU-T和ISO/IEC标准的发展而演化。ITU-T制作H.261和H.263, ISO/IEC制作MPEG-1和MPEG-4视觉,并且这两个组织联合制作H.262/MPEG-2视频和H.264/MPEG-4高级视频编解码(Advanced Video Coding, AVC)和H.265/HEVC标准。自H.262开始,视频编解码标准基于混合视频编解码结构,其中利用时域预测和变换编解码。为了探索超越HEVC的未来视频编解码技术,VCEG和MPEG于2015年联合地成立联合视频探索小组(JVET)。此后,JVET采纳了许多新方法且将其引入到命名为联合探索模型(JEM)的参考软件中。2018年4月,创建在VCEG(Q6/16)和ISO/IEC JTC1 SC29/WG11(MPEG)之间的联合视频专家小组(JVET),以致力于VVC标准,该VVC标准目标为与HEVC相比比特率降低50%。。

[0087] 2.1 色彩空间与色度子采样

[0088] 色彩空间,也称为色彩模型(或色彩系统),是一个抽象的数学模型,它将色彩范围

简单地描述为数字的元组,通常是3或4个值或色彩分量(例如RGB)。基本上说,色彩空间是坐标系和子空间的精化。

[0089] 对于视频压缩,最常用的色彩空间是YCbCr和RGB。

[0090] YCbCr、Y'CbCr或Y Pb/Cb Pr/Cr,也写作YCbCr或Y'CbCr,是色彩空间的家族,用作视频和数字摄影系统中彩色图片管线的一部分。Y'为亮度分量,Cb和Cr为蓝色差和红色差色度分量。Y'(带上标符号')与Y亮度不同,这意味着是基于gamma校正的RGB原色对光强进行非线性编码。

[0091] 色度子采样是通过对色度信息实现比对亮度信息更低的分辨率来编码图片的实践,利用了人类视觉系统对色差的敏锐度低于对亮度的敏锐度。

[0092] 2.1.1.4:4:4

[0093] 三个Y'CbCr分量中的每一个具有相同的采样率,因此没有色度子采样。该方案有时用于高端胶片扫描仪和电影后期制作中。

[0094] 2.1.2.4:2:2

[0095] 两个色度分量以亮度的一半采样率采样:水平色度分辨率减半。这使得未压缩视频信令信令的带宽减少了三分之一,几乎没有视觉差异

[0096] 2.1.3.4:2:0

[0097] 4:2:0中,水平采样比4:1:1增加了一倍,但由于在该方案中,Cb和Cr通道仅在每个交替线上被采样,因此垂直分辨率减半。因此,数据速率是相同的。Cb和Cr各自以2倍速在水平和垂直方向二者上进行子采样。有三种4:2:0方案的变型,它们具有不同的水平和垂直定址。

[0098] MPEG-2中,Cb和Cr水平同址。Cb和Cr在垂直方向上的像素之间定址(间隙地定址)。

[0099] JPEG/JFIF中,H.261和MPEG-1、Cb和Cr间隙地定址、在交替亮度样本之间的中间定址。

[0100] 4:2:0DV中,Cb和Cr在水平方向上同址。垂直方向上,它们在交替线上同址。

[0101] 2.2.典型视频编解码器的编解码流程

[0102] 图1显示了VVC的编码器框图的示例,其包括三个环路滤波块:解块滤波器(DF),样本自适应偏移(SAO)和自适应环路滤波器(ALF)。与使用预定义滤波器的DF不同,SAO和ALF利用当前图片的原始样本通过分别增加偏移和应用有限脉冲响应(FIR)滤波器来减少原始样本与重构样本之间的均方误差,其中编解码侧信息信令通知偏移和滤波器系数。ALF位于每个图片的最后一个处理阶段,可以看作是试图捕获和修复由先前阶段创建的伪像的工具。

[0103] 2.3 JEM中基于几何变换的自适应环路滤波器

[0104] JEM中,应用了具有基于块的滤波器自适应的基于几何变换的自适应环路滤波器(GALF)。对于亮度分量,基于局部梯度的方向和活动,为每个 2×2 块选择25个滤波器之中的一个。

[0105] 2.3.1滤波器形状

[0106] JEM中,可以为亮度分量选择最多三个菱形滤波器形状(如图2所示)。在图片级别处信令通知索引,以指示用于亮度分量的滤波器形状。

[0107] 对于图片中的色度分量,总是使用 5×5 菱形形状。

[0108] 2.3.1.1块分类

[0109] 每个 2×2 块被分类为25个类别中的一个。分类索引C是基于其方向性D和活动 \hat{A} 的量化值推导出的,如下所示:

$$[0110] \quad C = 5D + \hat{A} \quad (1)$$

[0111] 为了计算D和 \hat{A} ,首先使用以下的一维拉普拉斯公式计算水平、垂直和两个对角线方向的梯度:

$$[0112] \quad g_v = \sum_{k=i-2}^{i+3} \sum_{l=j-2}^{j+3} V_{k,l}, \quad V_{k,l} = |2R(k,l) - R(k,l-1) - R(k,l+1)|, \quad (2)$$

$$[0113] \quad g_h = \sum_{k=i-2}^{i+3} \sum_{l=j-2}^{j+3} H_{k,l}, \quad H_{k,l} = |2R(k,l) - R(k-1,l) - R(k+1,l)|, \quad (3)$$

$$[0114] \quad g_{d1} = \sum_{k=i-2}^{i+3} \sum_{l=j-3}^{j+3} D1_{k,l}, \quad D1_{k,l} = |2R(k,l) - R(k-1,l-1) - R(k+1,l+1)| \quad (4)$$

$$[0115] \quad g_{d2} = \sum_{k=i-2}^{i+3} \sum_{l=j-2}^{j+3} D2_{k,l}, \quad D2_{k,l} = |2R(k,l) - R(k-1,l+1) - R(k+1,l-1)| \quad (5)$$

[0116] 指数i和j是指 2×2 块中左上样本的坐标,并且R(i,j)表示坐标(i,j)处的重构样本。

[0117] 然后D水平和垂直方向的梯度的最大值和最小值被设置为:

$$[0118] \quad g_{h,v}^{max} = \max(g_h, g_v), \quad g_{h,v}^{min} = \min(g_h, g_v) \quad (6)$$

[0119] 并且两个对角方向的梯度的最大值和最小值被设置为:

$$[0120] \quad g_{d0,d1}^{max} = \max(g_{d0}, g_{d1}), \quad g_{d0,d1}^{min} = \min(g_{d0}, g_{d1}) \quad (7)$$

[0121] 为了推导出方向性D的值,将这些值相互比较,并使用两个阈值 t_1 和 t_2 :

[0122] 步骤1. 如果 $g_{h,v}^{max} \leq t_1 \cdot g_{h,v}^{min}$ 和 $g_{d0,d1}^{max} \leq t_1 \cdot g_{d0,d1}^{min}$ 同时为真,则D被设置为0。

[0123] 步骤2. 如果 $g_{h,v}^{max} / g_{h,v}^{min} > g_{d0,d1}^{max} / g_{d0,d1}^{min}$,则从步骤3继续;否则从步骤4继续。

[0124] 步骤3. 如果 $g_{h,v}^{max} > t_2 \cdot g_{h,v}^{min}$,则D被设置为2;否则D被设置为1。

[0125] 步骤4. 如果 $g_{d0,d1}^{max} > t_2 \cdot g_{d0,d1}^{min}$,则D被设置为4;否则D被设置为3。

[0126] 活动值A计算如下:

$$[0127] \quad A = \sum_{k=i-2}^{i+3} \sum_{l=j-2}^{j+3} (V_{k,l} + H_{k,l}). \quad (8)$$

[0128] A被进一步量化到0到4的范围(含端值),并且量化值表示为 \hat{A} 。

[0129] 对于图片中的两个色度分量,不应用分类方法,例如,对每个色度分量应用一组ALF系数。

[0130] 2.3.1.2.滤波器系数的几何变换

[0131] 在对每个 2×2 块进行滤波之前,根据为该块计算的梯度值,将几何变换(例如旋转或对角线和垂直翻转)应用于滤波器系数 $f(k,l)$ 。这等同于将这些变换应用于滤波器支持区域中的样本。想法是通过对准它们的方向性来使应用了ALF的不同块更加相似。

[0132] 引入了三种几何变换,包括对角线、垂直翻转和旋转:

对角线: $f_D(k, l) = f(l, k)$,

[0133] 垂直翻转: $f_V(k, l) = f(k, K - l - 1)$ (9)

旋转: $f_R(k, l) = f(K - l - 1, k)$

[0134] 其中K是滤波器的尺寸, $0 \leq k, 1 \leq K-1$ 是系数坐标, 因此位置(0,0)在左上角, 并且位置(K-1, K-1)在右下角。根据为该块计算出的梯度值, 将变换应用于滤波器系数 $f(k, l)$ 。表1总结了变换与四个方向的四个梯度之间的关系。

[0135] 表1: 为一个块计算出的梯度与变换的映射

梯度值	变换
$g_{d2} < g_{d1} \ \& \ g_o < g_{ave}$	没有变换
$g_{d2} < g_{d1} \ \& \ g_{ive} < g_h$	对角线
$g_{d1} < g_{d2} \ \& \ g_h < g_v$	垂直翻转
$g_{d1} < g_{d2} \ \& \ g_v < g_h$	旋转

[0137] 2.3.1.3. 滤波器参数信令通知

[0138] JEM中, 对于第一个CTU信令通知GALF滤波器参数, 例如, 在条带标头之后和第一个CTU的SA0参数之前。最多可以信令通知亮度滤波器系数的25个集合。为了降低比特开销, 可以merge不同分类的滤波器系数。此外, 参考图片的GALF系数被存储并且允许被重用为当前图片的GALF系数。当前图片可以选择使用为参考图片存储的GALF系数, 并且绕过GALF系数信令通知。在这种情况下, 仅信令通知参考图片中的一个的索引, 并且当前图片继承所指示的参考图片的存储的GALF系数。

[0139] 为了支持GALF时域预测, 维护GALF滤波器集的候选列表。在解码新序列的开始, 候选列表为空。解码一个图片后, 可将对应的滤波器集添加到候选列表中。一旦候选列表的尺寸达到最大允许值(例如, 当前JEM中的6), 则滤波器的新集合将盖写解码顺序中最旧的集合, 即, 应用先进先出(FIFO)规则来更新候选列表。为了避免重复, 只有当对应的图片不使用GALF时域预测时, 才能将一集合添加到列表中。为了支持时域可缩放, 有滤波器集的多个候选列表, 并且每个候选列表都与时域层相关联。更具体地说, 由时域层索引(TempIdx)分配的每个阵列可以构成具有等于较低的TempIdx的先前解码图片的滤波器集。例如, 第k个阵列被分配为与等于k的TempIdx相关联, 并且它只包括来自TempIdx小于或等于k的图片中的滤波器集。在对某一图片进行编解码后, 与该图片相关联的滤波器集将用于更新与相等或较高的TempIdx相关联的那些阵列。

[0140] GALF系数的时域预测用于帧间编解码帧以最小化信令通知开销。对于帧内帧, 时域预测不可用, 并且向每个类分配16个固定滤波器的集合。为了指示固定滤波器的使用, 信令通知每个类的标志, 并且如果需要的话, 信令通知所选择的固定滤波器的索引。即使在为给定类选择了固定滤波器时, 仍可以为该类发送自适应滤波器 $f(k, l)$ 的系数, 在这种情况下, 将应用于重构图片的滤波器的系数是系数的两个集合的总和。

[0141] 亮度分量的滤波过程可以控制在CU水平。标志被信令通知, 以表示GALF是否应用于CU的亮度分量。对于色度分量, 仅在图片级别指示是否应用GALF。

[0142] 2.3.1.4. 滤波过程

[0143] 在解码器侧, 当对块启用GALF时, 对块内的每个样本 $R(i, j)$ 进行滤波, 得到如下所

示的样本值 $R'(i, j)$, 其中 L 表示滤波器长度, $f_{m,n}$ 表示滤波器系数, $f(k, l)$ 表示解码的滤波器系数。

$$[0144] \quad R'(i, j) = \sum_{k=-L/2}^{L/2} \sum_{l=-L/2}^{L/2} f(k, l) \times R(i+k, j+l) \quad (10)$$

[0145] 2.3.1.5. 编码侧滤波器参数确定过程

[0146] GALF的整体编码器决策过程如图3所示。对于每个CU的亮度样本, 编码器决定是否应用GALF并且在条带标头中包括适当的信令通知标志。对于色度样本, 应用滤波器的决定是基于图片级别而不是基于CU级别来完成的。此外, 仅当为图片启用亮度GALF时, 才检查图片的色度GALF。

[0147] 2.4. VTM4.0中基于几何变换的自适应环路滤波器

[0148] VVC中GALF的当前设计与JEM中的GALF的当前设计相比有以下主要变化:

[0149] (1) 自适应滤波器形状被移除。亮度分量只允许 7×7 的滤波器形状, 色度分量允许 5×5 的滤波器形状。

[0150] (2) ALF参数的时域预测和来自固定滤波器的预测二者被移除。

[0151] (3) 对于每个CTU, 一个比特标志信令通知ALF是启用的还是禁用的。

[0152] (4) 类索引的计算是在 4×4 级别而不是 2×2 级别执行。此外, 还可以利用用于ALF分类的子采样拉普拉斯计算方法。更具体地说, 不需要计算一个块内每个样本的水平/垂直/45对角线/135度梯度。相反, 使用1:2的子采样。

[0153] 图4A-4D示出了一些子采样拉普拉斯计算的示例。

[0154] 2.4.1 滤波过程

[0155] VTM4.0中, 自适应环路滤波器的滤波过程被执行为如下:

$$[0156] \quad O(x, y) = \sum_{(i, j)} w(i, j) \times I(x+i, y+j) \quad (11)$$

[0157] 其中, 样本 $I(x+i, y+j)$ 是输入样本, $O(x, y)$ 是滤波后的输出样本(例如, 滤波器结果), $w(i, j)$ 表示滤波器系数。实际上, VTM4.0中, 使用整数算法实现定点精度计算:

$$[0158] \quad O(x, y) = \left(\sum_{i=-\frac{L}{2}}^{\frac{L}{2}} \sum_{j=-\frac{L}{2}}^{\frac{L}{2}} w(i, j) \times I(x+i, y+j) + 64 \right) \gg 7 \quad (12)$$

[0159] 式中 L 表示滤波器长度, 式中 $w(i, j)$ 为定点精度下的滤波器系数。

[0160] 2.5. 示例性非线性ALF

[0161] 2.5.1 滤波重形成

[0162] 公式(11)可以在不影响编解码效率的情况下重新形成如下表达式:

$$[0163] \quad O(x, y) = I(x, y) + \sum_{(i, j) \neq (0, 0)} w(i, j) \times (I(x+i, y+j) - I(x, y)) \quad (13)$$

[0164] 式中, $w(i, j)$ 与等式(11)中的滤波器系数相同[除了等式(13)中的 $w(0, 0)$ 等于1, 而等式(11)中等于 $1 - \sum_{(i, j) \neq (0, 0)} w(i, j)$ 以外]。

[0165] 2.5.2. 修改的滤波过程

[0166] 利用上述(13)的滤波方程, 当邻近样本值($I(x+i, y+j)$)与被滤波的当前样本值 $I(x, y)$ 相差过大时, 通过使用简单限幅函数来降低邻近样本值的影响, 可以引入非线性以使ALF更有效。

[0167] 在该提议中, ALF滤波器修改如下:

$$[0168] \quad O'(x, y) = I(x, y) + \sum_{(i, j) \neq (0, 0)} w(i, j) \times K(I(x+i, y+j) - I(x, y), k(i, j)) \quad (14)$$

[0169] 其中 $K(d, b) = \min(b, \max(-b, d))$ 是限幅函数, $k(i, j)$ 是限幅参数, 其取决于与 $I(x+i, y+j)$ 相关联的滤波器系数。编码器执行优化以找到最佳的 $k(i, j)$ 。

[0170] 在一些实施例中, 为每个ALF滤波器指定限幅参数 $k(i, j)$, 每个滤波器系数信令通知一个限幅值。这意味着在每个亮度滤波器的比特流中最多可以信令通知12个限幅值, 并且对于色度滤波器, 最多可以信令通知6个限幅值。

[0171] 为了限制信令通知成本和编码器复杂度, 可以将限幅值的评估限制为可能值的小集合。在一些实施例中, 可以使用4个固定值, 这些固定值对于帧间和帧内片组是相同的。

[0172] 由于亮度的局部差异的方差通常高于色度, 因此可以为亮度和色度滤波器使用两个不同的集合。最大样本值(这里是1024, 表示10比特的比特深度)可以包括在每个集合中, 使得如果没有必要的话, 可以禁用限幅。

[0173] 表2中提供了一些实施例中使用的限幅值的集合。在对数域中, 通过大致相等地划分亮度的样本值的全部范围(编解码为10比特)和色度的4到1024的范围, 来选择4个值。

[0174] 更精确地, 通过以下公式获得了限幅值的亮度表:

$$[0175] \quad \text{AlfClip}_L = \left\{ \text{取整} \left(\left(\left(M \right)^{\frac{1}{N}} \right)^{N-n+1} \right) \text{ 其中 } n \in 1..N \right\}, \text{ 其中 } M=2^{10} \text{ 且 } N=4.$$

[0176] 类似地, 根据以下公式获得限幅值的色度表:

$$[0177] \quad \text{AlfClip}_C = \left\{ \text{取整} \left(A \cdot \left(\left(\frac{M}{A} \right)^{\frac{1}{N-1}} \right)^{N-n} \right) \text{ 其中 } n \in 1..N \right\}, \text{ 其中 } M=2^{10}, N=4$$

且 $A=4$ 。

[0178] 表2: 授权的限幅值

	帧间/帧内片组
[0179] 亮度	{ 1024, 181, 32, 6 }
色度	{ 1024, 161, 25, 4 }

[0180] 通过使用与上表2中的限幅值的索引相对应的Golomb编解码方案, 在“alf_data”语法元素中对所选限幅值进行编码。此编码方案与滤波器索引的编码方案相同。

[0181] 2.6. 示例性基于CTB的ALF

[0182] 在该提议中引入了三个主要变化:

[0183] (1) 从CTB级别中的其他ALF APS (也称为时域预测) 继承滤波器系数。

[0184] (2) 来自固定滤波器的滤波器系数的预测编解码。

[0185] (3) CTB级别控制是使用来自固定滤波器集或时域预测的预测编解码, 或是从一个信令通知的滤波器集(条带标头中)中继承。

[0186] 2.6.1. 条带级别时域滤波器

[0187] VTM4中采用了自适应参数集(APS)。每个APS包括信令通知的ALF滤波器的一个集合, 最多支持32个APS。在该提议中, 测试了条带级别的时域滤波器。片组可以重新使用来自APS的ALF信息以降低开销。APS被更新为先进先出(FIFO)缓冲区。

[0188] 2.6.2. 附加的示例性基于CTB的ATF

[0189] 对于亮度分量,当将ALF应用于亮度CTB时,指示从16个固定滤波器集或5个时域滤波器集的预测中选择,还是从一个信令通知的滤波器集中(在条带标头中)继承。仅滤波器集索引被信令通知。对于一个条带,仅可以信令通知25个滤波器的一个新集合。如果为条带信令通知了新的集合,则同一条带中的所有亮度CTB都共享该集合。固定滤波器集可用于预测新的条带级别滤波器集,并且还用作亮度CTB的候选滤波器集。滤波器的总数为64。

[0190] 对于色度分量,当将ALF应用于色度CTB时,如果为条带信令通知了新滤波器,则CTB将使用该新滤波器,否则,将应用满足时域可缩放性约束条件的最新时域色度滤波器。

[0191] 作为条带级别的时域滤波器,将APS更新为先进先出(FIFO)缓冲区。

[0192] 2.6.3. 修改的规范

[0193] 以下文本用固定的滤波器、时域滤波器和基于CTB的滤波器索引进行了修改。

[0194] 2.6.3.1. 语法表

[0195] 7.3.3.2 自适应环路滤波器数据语法

[0196]

alf_data() {	描述符
alf_chroma_idc	tu(v)
alf_signal_new_filter_luma	u(1)
if(alf_signal_new_filter_luma > 0) {	
alf_luma_use_fixed_filter_flag	u(1)
if(alf_luma_use_fixed_filter_flag){	
alf_luma_fixed_filter_set_index	tb(v)
alf_luma_fixed_filter_usage_pattern	u(1)
if(alf_luma_fixed_filter_usage_pattern > 0)	
for(i = 0; i < NumAlFilters; i++)	
alf_luma_fixed_filter_usage[i]	u(1)
alf_num_available_temporal_filter_sets_luma	tb(1)
alf_luma_num_filters_signaled_minus1	tb(v)
alf_luma_type_flag	u(1)
if(alf_luma_num_filters_signaled_minus1 > 0) {	
for(filtIdx = 0; filtIdx < NumAlFilters; filtIdx++)	
alf_luma_coeff_delta_idx[filtIdx]	tu(v)
}	
alf_luma_coeff_delta_flag	u(1)
if(!alf_luma_coeff_delta_flag && alf_luma_num_filters_signaled_minus1 > 0)	
alf_luma_coeff_delta_prediction_flag	u(1)
alf_luma_min_eg_order_minus1	tu(v)
for(i = 0; i < (alf_luma_type_flag == 1) ? 2 : 3; i++)	
alf_luma_eg_order_increase_flag[i]	u(1)

	if (alf_luma_coeff_delta_flag) {	
	for(sigFiltIdx = 0; sigFiltIdx <= alf_luma_num_filters_signaled_minus1; sigFiltIdx++)	
	alf_luma_coeff_flag [sigFiltIdx]	u(1)
	}	
	for(sigFiltIdx = 0; sigFiltIdx <= alf_luma_num_filters_signaled_minus1; sigFiltIdx++){	
	if (alf_luma_coeff_flag[sigFiltIdx]) {	
	for (j = 0; j < (alf_luma_type_flag == 1) ? 6 : 12; j++) {	
	alf_luma_coeff_delta_abs [sigFiltIdx][j]	uek(v)
	if(alf_luma_coeff_delta_abs[sigFiltIdx][j])	
	alf_luma_coeff_delta_sign [sigFiltIdx][j]	u(1)
	}	
	}	
	}	
[0197]	alf_luma_coeff_delta_sign [sigFiltIdx][j]	u(1)
	}	
	if (alf_chroma_idc > 0) {	
	alf signal new filter chroma	u(1)
	if(alf signal new filter chroma){	
	alf_chroma_min_eg_order_minus1	tu(v)
	for(i = 0; i < 2; i++)	
	alf_chroma_eg_order_increase_flag [i]	u(1)
	for(j = 0; j < 6; j++) {	
	alf_chroma_coeff_abs [j]	uek(v)
	if(alf_chroma_coeff_abs[j] > 0)	
	alf_chroma_coeff_sign [j]	u(1)
	}	
	}	
	}	
	}	

[0198] 7.3.4.2. 编解码树单元语法

	描述符
coding_tree_unit() {	
xCtb = (CtbAddrInRs % PicWidthInCtbsY) << CtbLog2SizeY	
yCtb = (CtbAddrInRs / PicWidthInCtbsY) << CtbLog2SizeY	
if(slice_alf_enable_flag){	
alf_ctb_flag [0][xCtb >> Log2CtbSize][yCtb >> Log2CtbSize]	ae(v)
if(alf_ctb_flag[0][xCtb >> Log2CtbSize][yCtb >> Log2CtbSize] {	
<u>coding_alf_ctu_filter_set_index(CtbAddrInRs)</u>	
}	
if(alf_chroma_idc == 1 alf_chroma_idc == 3)	
alf_ctb_flag [1][xCtb >> Log2CtbSize][yCtb >> Log2CtbSize]	ae(v)
if(alf_chroma_idc == 2 alf_chroma_idc == 3)	
alf_ctb_flag [2][xCtb >> Log2CtbSize][yCtb >> Log2CtbSize]	ae(v)
}	
if(slice_type == I && qtbt_dual_tree_intra_flag) {	
dual_tree_implicit_qt_split (xCtb, yCtb, CtbLog2SizeY, 0)	
else	
coding_quadtree(xCtb, yCtb, CtbLog2SizeY, 0, SINGLE_TREE)	
}	
coding_alf_ctu_filter_set_index(CtbAddrInRs, slice_type){	Descriptor
xCtb = (CtbAddrInRs % PicWidthInCtbsY) << CtbLog2SizeY	
yCtb = (CtbAddrInRs / PicWidthInCtbsY) << CtbLog2SizeY	
alf_use_new_filter	ae(1)
if(alf_use_new_filter == 0){	
alf_use_fixed_filter	ae(1)
}	
if(alf_use_new_filter){	
alf_luma_ctb_filter_set_index[xCtb >> Log2CtbSize][yCtb >> Log2CtbSize] = 16	
}	
else if(alf_use_fixed_filter){	
alf_fixed_filter_index	tb(v)
alf_luma_ctb_filter_set_index[xCtb >> Log2CtbSize][yCtb >> Log2CtbSize] =	
alf_fixed_filter_index	
}	
else{	
alf_temporal_index	tb(v)
alf_luma_ctb_filter_set_index[xCtb >> Log2CtbSize][yCtb >> Log2CtbSize] = 16 +	
alf_temporal_index	
}	

[0199] 2.6.3.2. 语义

[0200] 7.4.4.2. 自适应环路滤波器数据语义

[0201] `alf_signal_new_filter_luma` 等于1指定信令通知新的亮度滤波器集。`alf_signal_new_filter_luma` 等于0指定不信令通知新的亮度滤波器集。`alf_signal_new`

filter_luma在不存在时为0。

[0203] alf_luma_use_fixed_filter_flag等于1指定固定滤波器集用于信令通知自适应环路滤波器。alf_luma_use_fixed_filter_flag等于0指定固定滤波器集不用于信令通知自适应环路滤波器。

[0204] alf_luma_fixed_filter_set_index指定固定滤波器集索引。它可以是0…15。

[0205] alf_luma_fixed_filter_usage_pattern等于0指定所有新滤波器使用固定滤波器。alf_luma_fixed_filter_usage_pattern等于1指定新滤波器中的一些滤波器使用固定滤波器,而其他滤波器不使用固定滤波器。

[0206] alf_luma_fixed_filter_usage[i]等于1指定第i个滤波器使用固定滤波器。alf_luma_fixed_filter_usage[i]等于0指定第i个滤波器不使用固定滤波器。alf_luma_fixed_filter_usage[i]在不存在时推断为1。

[0207] alf_signal_new_filter_chroma等于1指定信令通知新的色度滤波器。alf_signal_new_filter_chroma等于0指定不信令通知新的色度滤波器。

[0208] alf_num_available_temporal_filter_sets_luma指定可用于当前条带的可用时域滤波器集的数目,其可以是0…5。它在不存在时为0。

[0209] 将变量alf_num_available_filter_sets推导为 $16 + \text{alf_signal_new_filter_luma} + \text{alf_num_available_temporal_filter_sets_luma}$ 。

[0210] 如果alf_signal_new_filter_luma为1,则有以下过程:

[0211] 变量filterCoefficients[sigFiltIdx][j] (其中,sigFiltIdx=0..alf_luma_num_filters_signaled_minus1,j=0..11)初始化如下:

[0212] $\text{filterCoefficients}[\text{sigFiltIdx}][j] = \text{alf_luma_coeff_delta_abs}[\text{sigFiltIdx}][j] * \quad (7-50)$

[0213] $(1 - 2 * \text{alf_luma_coeff_delta_sign}[\text{sigFiltIdx}][j])$

[0214] 当alf_luma_coeff_delta_prediction_flag等于1时,filterCoefficients[sigFiltIdx][j] (其中,sigFiltIdx=1..alf_luma_num_filters_signaled_minus1且j=0..11)被修改为如下:

[0215] $\text{filterCoefficients}[\text{sigFiltIdx}][j] = \text{alf_luma_coeff_delta_abs}[\text{sigFiltIdx}][j] * \quad (7-50)$

[0216] $(1 - 2 * \text{alf_luma_coeff_delta_sign}[\text{sigFiltIdx}][j])$

[0217] 当alf_luma_coeff_delta_prediction_flag等于1时,filterCoefficients[sigFiltIdx][j] (其中,sigFiltIdx=1..alf_luma_num_filters_signaled_minus1且j=0..11)被修改为如下:

[0218] $\text{filterCoefficients}[\text{sigFiltIdx}][j] += \text{filterCoefficients}[\text{sigFiltIdx}-1][j] \quad (7-51)$

[0219] 亮度滤波器系数AlfCoeff_L具有元素AlfCoeff_L[filtIdx][j] (其中,filtIdx=0..NumAlfFilters-1且j=0..11)被推导为如下:

[0220] $\text{AlfCoeffL}[\text{filtIdx}][j] = \text{filterCoefficients}[\text{alf_luma_coeff_delta_idx}[\text{filtIdx}]] [j] \quad (7-52)$

[0221] 如果alf_luma_use_fixed_filter_flag为1且alf_luma_fixed_filter_usage

[filtIdx]为1,则以下适用:

[0222] $\text{AlfCoeff}_L[\text{filtIdx}][j] = \text{AlfCoeff}_L[\text{filtIdx}][j] + \text{AlfFixedFilterCoeff}[\text{AlfClassToFilterMapping}[\text{alf_luma_fixed_filter_index}][\text{filtIdx}]] [j]$

[0223] 针对 $\text{filtIdx} = 0 \dots \text{NumAlfFilters} - 1$,最后的滤波器系数 $\text{AlfCoeff}_L[\text{filtIdx}][12]$ 被推导为如下:

[0224] $\text{AlfCoeff}_L[\text{filtIdx}][12] = 128 - \sum_k (\text{AlfCoeff}_L[\text{filtIdx}][k] \ll 1)$,其中 $k = 0 \dots 11$ (7-53)

[0225] 比特流一致性要求具有 $\text{filtIdx} = 0 \dots \text{NumAlfFilters} - 1, j = 0 \dots 11$ 的 $\text{AlfCoeff}_L[\text{filtIdx}][j]$ 的值应在 -2^7 到 $2^7 - 1$ 的范围内(含端值),并且 $\text{AlfCoeff}_L[\text{filtIdx}][12]$ 的值应在 0 到 $2^8 - 1$ 的范围内(含端值)。

[0226] 亮度滤波器系数 $\text{AlfCoeff}_{\text{LumaAll}}$ 具有元素 $\text{AlfCoeff}_{\text{LumaAll}}[\text{filtSetIdx}][\text{filtIdx}][j]$ (其中, $\text{filtSetIdx} = 0 \dots 15, \text{filtSetIdx} = 0 \dots \text{NumAlfFilters} - 1$ 且 $j = 0 \dots 12$),被推导为如下:

[0227] $\text{AlfCoeff}_{\text{LumaAll}}[\text{filtSetIdx}][\text{filtIdx}][j] = \text{AlfFixedFilterCoeff}[\text{AlfClassToFilterMapping}[\text{filtSetIdx}][\text{filtIdx}]] [j]$

[0228] 亮度滤波器系数 $\text{AlfCoeff}_{\text{LumaAll}}$ 具有元素 $\text{AlfCoeff}_{\text{LumaAll}}[\text{filtSetIdx}][\text{filtIdx}][j]$ (其中, $\text{filtSetIdx} = 16, \text{filtSetIdx} = 0 \dots \text{NumAlfFilters} - 1$ 且 $j = 0 \dots 12$),被推导为如下:

[0229] 变量`closest_temporal_index`被初始化为-1。`Tid`为当前条带的时域层索引。

[0230] 如果`alf_signal_new_filter_luma`为1

[0231] $\text{AlfCoeff}_{\text{LumaAll}}[16][\text{filtIdx}][j] = \text{AlfCoeff}_L[\text{filtIdx}][j]$

[0232] 否则,调用以下过程

```
for (i = Tid; i >= 0; i--)
```

```
{
```

```
  for (k = 0; k < temp_size_L; k++)
```

```
  {
```

```
    if (tempTidL[k] == i)
```

[0233] {

```
      closest_temporal_index is set as k;
```

```
      break;
```

```
    }
```

```
  }
```

```
}
```

[0234] $\text{AlfCoeff}_{\text{LumaAll}}[16][\text{filtIdx}][j] = \text{Temp}_L[\text{closest_temporal_index}][\text{filtIdx}][j]$

[0235] 亮度滤波器系数 $\text{AlfCoeff}_{\text{LumaAll}}$ 具有元素 $\text{AlfCoeff}_{\text{LumaAll}}[\text{filtSetIdx}][\text{filtIdx}][j]$

[j] (其中, $\text{filtSetIdx}=17..\text{alf_num_available_filter_sets}-1$, $\text{filtSetIdx}=0..\text{NumAlfFilters}-1$ 且 $j=0..12$), 被推导为如下:

```

i = 17;
for (k = 0; k < temp_size_L and i < alf_num_available_filter_sets; j++)
{
    if (tempTid_L[ k ] <= Tid and k is not equal to closest_temporal_index)
    {
        AlfCoeffLumaAll[ i ][ filtIdx ][ j ] = TempL[ k ][ filtIdx ][ j ];
        i++;
    }
}

```

[0236]

AlfFixedFilterCoeff [64][13] =

```

{
    { 0, 0, 2, -3, 1, -4, 1, 7, -1, 1, -1, 5, 112 },
    ...
};

```

AlfClassToFilterMapping [16][25] =

```

{
    { 8, 2, 2, 2, 3, 4, 53, 9, 9, 52, 4, 4, 5, 9, 2,
    8, 10, 9, 1, 3, 39, 39, 10, 9, 52 },
    ... };

```

[0237]

[0238] -如果 $\text{alf_signal_new_filter_chroma}$ 为1,则有以下过程:

[0239] 色度滤波器系数 $\text{AlfCoeff}_c[j]$ (其中 $j=0..5$)被推导为如下:

[0240] $\text{AlfCoeff}_c[j] = \text{alf_chroma_coeff_abs}[j] * (1 - 2 * \text{alf_chroma_coeff_sign}[j])$
(7-57)

[0241] 对于 $j=6$,最后的滤波器系数被推导为如下:

[0242] $\text{AlfCoeff}_c[6] = 128 - \sum_k (\text{AlfCoeff}_c[k] \ll 1)$, 其中 $k=0..5$ (7-58)

[0243] 比特流一致性的要求具有 $j=0..5$ 的 $\text{AlfCoeff}_c[j]$ 的值应在 -2^7 到 2^7-1 的范围内(含端值),并且 $\text{AlfCoeff}_c[6]$ 的值应在0到 2^8-1 的范围内(含端值)。

[0244] -否则($\text{alf_signal_new_filter_chroma}$ 为0),调用以下:

```

- for (i = Tid; i >= 0; i--)
- {
-   for (k = 0; k < temp_size_C; k++)
-   {
-     if (tempTid_C[k] == i)
[0245] -   {
-       closest_temporal_index is set as k;
-       break;
-     }
-   }
- }
- }

```

[0246] 色度滤波器系数 $\text{AlfCoeff}_c[j]$ (其中 $j=0..6$)被推导为如下:

[0247] $\text{AlfCoeff}_c[j] = \text{Temp}_c[\text{closest_temporal_index}][j]$

[0248] 7.4.5.2. 编解码数单元语义

[0249] $\text{alf_luma_ctb_filter_set_index}[\text{xCtb} \gg \text{Log2CtbSize}][\text{yCtb} \gg \text{Log2CtbSize}]$

[0250] 指定位置(xCtb,yCtb)处的亮度CTB的滤波器集索引。

[0251] $\text{alf_use_new_filter}$ 等于1指定 $\text{alf_luma_ctb_filter_set_index}[\text{xCtb} \gg \text{Log2CtbSize}][\text{yCtb} \gg \text{Log2CtbSize}]$ 为16。 $\text{alf_use_new_filter}$ 等于0指定 $\text{alf_luma_ctb_filter_set_index}[\text{xCtb} \gg \text{Log2CtbSize}][\text{yCtb} \gg \text{Log2CtbSize}]$ 不为16。

[0252] $\text{alf_use_fixed_filter}$ 等于1指定固定滤波器集中的一个被使用。 $\text{alf_use_fixed_filter}$ 等于0指定当前亮度CTB不使用任何固定滤波器集。

[0253] $\text{alf_fixed_filter_index}$ 指定固定滤波器集索引,其可以为从0至15。

[0254] $\text{alf_temporal_index}$ 指定时域滤波器集索引,其可以为从0至 $\text{alf_num_available_temporal_filter_sets_luma}-1$ 。

[0255] 2.6.3.3编解码过程

[0256] 8.5.1总则

[0257] 1. 当 $\text{sps_alf_enabled_flag}$ 等于0,则以下适用:

[0258] -调用如8.5.4.5条中所指定的时域滤波器刷新过程。

[0259] -以重构图片样本阵列 S_L 、 S_{Cb} 和 S_{Cr} 为输入且以样本自适应偏移后的修改的重构图片样本阵列 S'_L 、 S'_{Cb} 和 S'_{Cr} 为输出,调用8.5.4.1条中指定的自适应环路滤波器过程。

[0260] -将阵列 S'_L 、 S'_{Cb} 和 S'_{Cr} 分别分配给阵列 S_L 、 S_{Cb} 和 S_{Cr} (它们代表解码的图片)。

[0261] -调用如8.5.4.6条中所指定的时域滤波器更新过程。

[0262] 8.5.4.2用于亮度样本的编解码树块滤波过程

[0263] -对应于由 $\text{filtIdx}[x][y]$ 指定的滤波器的亮度滤波器系数 $f[j]$ 的阵列被推导为如下,其中 $j=0..12$:

[0264] $f[j] = \text{AlfCoeff}_{\text{LumaAll}}[\text{alf_luma_ctb_filter_set_index}[\text{xCtb} \gg \text{Log2CtbSize}]$

[yCtb>>

[0265] $\text{Log2CtbSize}}]][\text{filtIdx}[x][y]][j]$ (8-732)

[0266] 8.5.4.5时域滤波器刷新

[0267] 如果下面的任何一个条件是真的，

[0268] -当前图片是IDR图片

[0269] -当前图片是BLA图片

[0270] -在解码顺序中，当前图片是POC大于最后解码的IRAP图片的POC的第一张图片，例如在前面图片之后和后续图片之前。

[0271] 那么，将temp_size_L和temp_size_C被设置为0。

[0272] 8.5.4.6时域滤波器更新

[0273] 如果slice_alf_enabled_flag为1且alf_signal_new_filter_luma为1，则以下适用：

[0274] 如果亮度时域滤波器缓存区尺寸temp_size_L<5，则temp_size_L=temp_size_L+1。

[0275] $\text{Temp}_L[i][j][k]$ (其中， $i=\text{temp_size_L}-1\cdots 1$ ， $j=0\cdots \text{NumAlfFilters}-1$ 且 $k=0\cdots 12$) 被更新为：

[0276] $\text{Temp}_L[i][j][k]=\text{Temp}_L[i-1][j][k]$

[0277] $\text{Temp}_L[i][j][k]$ (其中， $j=0\cdots \text{NumAlfFilters}-1$ 且 $k=0..12$) 被更新为：

[0278] $\text{Temp}_L[0][j][k]=\text{AlfCoeff}_L[j][k]$

[0279] $\text{Temp}_{\text{Tid}_L}[i]$ (其中， $i=\text{temp_size_L}-1\cdots 1$) 被更新为：

[0280] $\text{Temp}_{\text{Tid}_L}[i]=\text{Temp}_{\text{Tid}_L}[i-1]$

[0281] $\text{Temp}_{\text{Tid}_L}[0]$ 被设置为当前条带的时域层索引Tid。

[0282] 如果alf_chroma_idx不为0且alf_signal_new_filter_chroma为1，则以下适用：

[0283] $\text{Temp}_c[i][j]$ (其中， $i=\text{temp_size_c}-1\cdots 1$ 且 $j=0\cdots 6$) 被更新为：

[0284] $\text{Temp}_c[i][j]=\text{Temp}_c[i-1][j]$

[0285] $\text{Temp}_c[0][j]$ (其中， $j=0\cdots 6$) 被更新为：

[0286] $\text{Temp}_c[0][j]=\text{AlfCoeff}_c[j]$

[0287] $\text{Temp}_{\text{Tid}_C}[i]$ (其中， $i=\text{temp_size_C}-1\cdots 1$) 被更新为：

[0288] $\text{Temp}_{\text{Tid}_C}[i]=\text{Temp}_{\text{Tid}_C}[i-1]$

[0289] $\text{Temp}_{\text{Tid}_C}[0]$ 被设置为当前条带的Tid。

[0290] 表3: 语法元素和相关的二值化

语法结构	语法元素	二值化	
		过程	输入参数
[0291]	alf_ctb_flag[][]	FL	cMax = 1
	alf use new filter set	FL	cMax = 1
	alf used fixed filter	FL	cMax = 1
	alf fixed filter index	TB	cMax = 15

语法结构	语法元素	二值化	
		过程	输入参数
[0292] coding_tree_unit ()	<u>alf temporal index</u>	<u>TB</u>	<u>cMax =</u> <u>alf num available temporal filter sets luma - 1</u>

[0293] 表4:将ctxInc分配给具有上下文编解码bin的语法元素

语法元素	binIdx					
	0	1	2	3	4	>= 5
[0294] <u>alf use new filter set</u>	<u>0</u>	<u>na</u>	<u>na</u>	<u>na</u>	<u>na</u>	<u>na</u>
<u>alf used fixed filter</u>	<u>0</u>	<u>na</u>	<u>na</u>	<u>na</u>	<u>na</u>	<u>na</u>

[0295] 2.7. 示例性环路重整形 (ILR)

[0296] 环路重整形 (ILR) 的基本思想是将原始 (第一域中) 信令 (预测/重构信令) 转换到第二域 (重整形的域)。

[0297] 环路亮度重整形器实现为一对查找表 (LUT), 但是两个LUT中只有一个需要信令通知, 因为另一个可以从信令通知的LUT计算出。每个LUT是一维10位1024条目的映射表 (1D-LUT)。一个LUT是正向LUT, FwdLUT, 它将输入亮度代码值 Y_i 映射到更改的值 $Y_r: Y_r = FwdLUT[Y_i]$ 。另一个LUT是反向LUT, InvLUT, 它将更改的代码值 Y_r 映射到 $\hat{Y}_i: \hat{Y}_i = InvLUT[Y_r]$ 。(\hat{Y}_i 代表 Y_i 的重构值。)

[0298] 2.7.1. 示例性PWL模型

[0299] 概念上, 分段线性 (PWL) 以如下方式来实现:

[0300] 设 x_1 、 x_2 为两个输入枢轴点, 并且 y_1 、 y_2 一体地为它们对应的输出枢轴点。在 x_1 和 x_2 之间的任何输入值 x 的输出值 y 可以通过以下公式进行插值:

[0301] $y = ((y_2 - y_1) / (x_2 - x_1)) * (x - x_1) + y_1$

[0302] 在定点实现方式中, 公式可以重写为:

[0303] $y = ((m * x + 2^{FP_PREC - 1}) \gg FP_PREC) + c$

[0304] 其中 m 是标量, c 是偏移量, FP_PREC 是用于指定精度的恒定值。

[0305] 注意到, PWL模型可用于预计算1024个条目的FwdLUT和InvLUT映射表; 但PWL模型还允许实现实时 (on-the-fly) 计算相同的映射值, 而无需预计算LUT。

[0306] 2.7.2. 示例性测试模型

[0307] 2.7.2.1 亮度重整形

[0308] 在一些实施例中, 环路亮度重整形的测试模型提供了较低复杂度的管线, 该管线还消除了帧间条带重构中用于逐块的帧内预测的解码延迟。在帧间条带和帧内条带二者的重整形的域中执行帧内预测。

[0309] 无论条带类型如何, 总是在重整形的域中执行帧内预测。通过这样的布置, 帧内预测可以在先前的TU重构完成后立即开始。这种布置还可以为帧内模式提供统一的过程, 而不是依赖于条带。图5示出基于模式的示例性解码过程的框图。

[0310] 在一些实施例中, 可以测试16段分段线性 (PWL) 模型的亮度和色度残差缩放, 而不是32段PWL模型。

[0311] 在一些实施例中, 可以在测试模式中实现具有环路亮度再整形器的帧间条带重构

(浅绿色阴影块指示在重整形的域中的信令通知:亮度残差;预测的帧内亮度;以及重构的帧内亮度)。

[0312] 2.7.2.2. 依赖于亮度的色度残差缩放

[0313] 依赖于亮度的色度残差缩放是利用定点整数运算实现的乘法过程。它是一种缩放过程,其中基于重构的亮度样本推导出缩放系数。然后,将缩放系数应用于色度残差(即色度残差缩放)。色度残差缩放补偿与色度信令通知的相互作用的亮度信令。色度残差缩放适用于TU级别。更具体地,利用相应的亮度预测块的平均值。

[0314] 该平均值用于识别PWL模型中的索引。该索引标识缩放比例因子cScaleInv。色度残差乘以该数字。

[0315] 注意到,色度缩放比例因子是根据正向映射的预测亮度值而不是重构亮度值来计算的。

[0316] 2.7.3. ILR的使用

[0317] 在编码器侧,首先将每个图片(或片组)转换为重整形的域。并且,所有编解码过程都在重整形的域中执行。对于帧内预测,邻近块是在重整形的域中;对于帧间预测,首先将参考块(从来自解码图片缓冲区的原始域生成)转换为重整形的域。然后生成残差并将其编解码到比特流。

[0318] 在整个图片(或片组)完成编码/解码之后,将重整形的域中的样本转换到原始域,然后应用去块滤波器和其他滤波器。

[0319] 对于以下情况,预测信令的正向重整形是禁用的:

[0320] (1) 当前块是帧内编解码的。

[0321] (2) 当前块被编解码为CPR(当前图片参考,又名帧内块复制,IBC)。

[0322] (3) 当前块被编解码为组合的帧间帧内模式(CIIP),并且帧内预测块的正向重整形是禁用的。

[0323] 2.7.4. 示例性信令通知ILR侧信息

[0324] 2.7.4.1. 语法表

[0325] LMCS APS:具有等于LMCS_APS的aps_params_type的APS

[0326] 7.3.2.3序列参数集RBSP语法

	描述符
seq_parameter_set_rbsp() {	
sps_decoding_parameter_set_id	u(4)
sps_max_sub_layers_minus1	u(3)
sps_reserved_zero_5bits	u(5)
profile_tier_level(sps_max_sub_layers_minus1)	
gra_enabled_flag	u(1)
sps_seq_parameter_set_id	ue(v)
chroma_format_idc	ue(v)
if(chroma_format_idc == 3)	
separate_colour_plane_flag	u(1)
pic_width_in_luma_samples	ue(v)
pic_height_in_luma_samples	ue(v)
...	
sps_lmcs_enabled_flag	u(1)
sps_ladf_enabled_flag	u(1)
if(sps_ladf_enabled_flag) {	
sps_num_ladf_intervals_minus2	u(2)
sps_ladf_lowest_interval_qp_offset	se(v)
for(i = 0; i < sps_num_ladf_intervals_minus2 + 1; i++) {	
sps_ladf_qp_offset[i]	se(v)
sps_ladf_delta_threshold_minus1[i]	ue(v)
}	
}	
...	
sps_extension_flag	u(1)
if(sps_extension_flag)	
while(more_rbsp_data())	
sps_extension_data_flag	u(1)
rbsp_trailing_bits()	
}	

[0327]

[0328]

[0329] 7.3.2.5 自适应参数集语法

	adaptation_parameter_set_rbsp() {	描述符
	adaptation_parameter_set_id	u(5)
	aps_params_type	u(3)
	if(aps_params_type == ALF_APS)	
	alf_data()	
	else if(aps_params_type == LMCS_APS)	
[0330]	lmcs_data()	
	aps_extension_flag	u(1)
	if(aps_extension_flag)	
	while(more_rbsp_data())	
	aps_extension_data_flag	u(1)
	rbsp_trailing_bits()	
	}	

[0331] 7.3.5条带标头语法

[0332] 7.3.5.1常规条带标头语法

	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)	
[0333]	num_bricks_in_slice_minus1	ue(v)
	slice_type	ue(v)
	if(NalUnitType == GRA_NUT)	
	recovery_poc_cnt	se(v)
	...	
	if(pps_slice_chroma_qp_offsets_present_flag) {	

[0334]

slice_cb_qp_offset	se(v)
slice_cr_qp_offset	se(v)
}	
if(sps_sao_enabled_flag) {	
slice_sao_luma_flag	u(1)
if(ChromaArrayType != 0)	
slice_sao_chroma_flag	u(1)
}	
if(sps_alf_enabled_flag) {	
slice_alf_enabled_flag	u(1)
if(slice_alf_enabled_flag) {	
num_alf_aps_ids_minus1	ue(v)
for(i = 0; i <= num_alf_aps_ids_minus1; i++)	
slice_alf_aps_id[i]	u(5)
}	
}	
dep_quant_enabled_flag	u(1)
if(!dep_quant_enabled_flag)	
sign_data_hiding_enabled_flag	u(1)
if(deblocking_filter_override_enabled_flag)	
deblocking_filter_override_flag	u(1)
if(deblocking_filter_override_flag) {	
slice_deblocking_filter_disabled_flag	u(1)
if(!slice_deblocking_filter_disabled_flag) {	
slice_beta_offset_div2	se(v)
slice_tc_offset_div2	se(v)
}	
}	
if(sps_lmcs_enabled_flag) {	
slice_lmcs_enabled_flag	u(1)
if(slice_lmcs_enabled_flag) {	
slice_lmcs_aps_id	
if(!(qtbt_dual_tree_intra_flag && slice_type == I))	
slice_chroma_residual_scale_flag	u(1)

	}	
	if(entropy_coding_sync_enabled_flag)	
	num_entry_point_offsets	ue(v)
	if(NumEntryPoints > 0) {	
[0335]	offset_len_minus1	ue(v)
	for(i = 0; i < NumEntryPoints; i++)	
	entry_point_offset_minus1[i]	u(v)
	}	
	byte_alignment()	
	}	

[0336] 7.3.5.4具有色度缩放数据语法的亮度映射

	<u>lmcs_data () {</u>	描述符
	<u>lmcs_min_bin_idx</u>	<u>ue(v)</u>
	<u>lmcs_delta_max_bin_idx</u>	<u>ue(v)</u>
	<u>lmcs_delta_cw_prec_minus1</u>	<u>ue(v)</u>
[0337]	<u>for (i = lmcs_min_bin_idx; i <= LmcsMaxBinIdx; i++) {</u>	
	<u>lmcs_delta_abs_cw[i]</u>	<u>u(v)</u>
	<u>if (lmcs_delta_abs_cw[i] > 0)</u>	-
	<u>lmcs_delta_sign_cw_flag[i]</u>	<u>u(1)</u>
	<u>}</u>	
	<u>}</u>	

[0338] 2.7.4.2. 语义

[0339] sps_lmcs_enabled_flag等于1指定在CVS中使用具有色度缩放的亮度映射。sps_lmcs_enabled_flag等于0指定在CVS中不使用具有色度缩放的亮度映射。

[0340] 自适应参数集语义

[0341] adaptation_parameter_set_id提供APS的标识符,供其他语法元素参考。APS可以跨图片共享,并且在图片内的不同条带中可以是不同的。

[0342] aps_params_type指定APS中携带的APS参数的类型,如表5中所指定。

[0343] 表5:APS参数类型代码和APS参数的类型

aps_params_ty	aps_params_ty	APS 参数的类型
pe	pe 的名称	
0	ALF_APS	ALF 参数
<u>1</u>	<u>LMCS_APS</u>	<u>LMCS 参数</u>
2..7	保留	保留

[0345] slice_lmcs_enabled_flag等于1指定为当前条带启用具有色度缩放的亮度映射。slice_lmcs_enabled_flag等于0指定为当前条带不启用具有色度缩放的亮度映射。slice_lmcs_enabled_flag在不存在时,则推断为等于0。

[0346] slice_lmcs_aps_id指定条带指代的LMCS APS的adaptation_parameter_set_id。具有adaptation_parameter_set_id等于slice_lmcs_aps_id的LMCS APS NAL单元的TemporalId应小于或等于编解码的条带NAL单元的TemporalId。

[0347] 当同一图片的两个或多个条带指代具有相同值的adaptation_parameter_set_id的多个LMCS APS时,具有相同值的adaptation_parameter_set_id的多个LMCS APS应具有相同的内容。

[0348] 7.4.6.4具有色度缩放数据语义的亮度映射

[0349] lmcs_min_bin_idx指定在具有色度缩放构建过程的亮度映射中使用的最小bin索引。lmcs_min_bin_idx的值应在0到15的范围内(含端值)。

[0350] lmcs_delta_max_bin_idx指定介于15与最大bin索引LmcsMaxBinIdx之间的增量值,该最大bin索引LmcsMaxBinIdx用于具有色度缩放构建过程的亮度映射中。lmcs_delta_max_bin_idx的值应在0到15的范围内(含端值)。LmcsMaxBinIdx的值被设置为等于15-lmcs_delta_max_bin_idx。LmcsMaxBinIdx的值应大于或等于lmcs_min_bin_idx。

[0351] lmcs_delta_cw_prec_minus1加1指定用于表示语法lmcs_delta_abs_cw[i]的比特数。lmcs_delta_cw_prec_minus1的值应在0到BitDepthY-2的范围内(含端值)。

[0352] lmcs_delta_abs_cw[i]指定第i个bin的绝对增量代码字值。

[0353] lmcs_delta_sign_cw_flag[i]指定变量lmcsDeltaCW[i]的符号如下:

[0354] -如果lmcs_delta_sign_cw_flag[i]等于0,则lmcsDeltaCW[i]是正值。

[0355] -否则(lmcs_delta_sign_cw_flag[i]不等于0),则lmcsDeltaCW[i]是负值。

[0356] lmcs_delta_sign_cw_flag[i]在不存在时,推断等于0。

[0357] 变量OrgCW被推导为如下:

[0358]
$$\text{OrgCW} = (1 \ll \text{BitDepth}_Y) / 16 \quad (7-77)$$

[0359] 变量lmcsDeltaCW[i] (其中, $i = \text{lmcs_min_bin_idx} \dots \text{LmcsMaxBinIdx}$) 被推导为如下:

[0360]
$$\text{lmcsDeltaCW}[i] = (1 - 2 * \text{lmcs_delta_sign_cw_flag}[i]) * \text{lmcs_delta_abs_cw}[i] \quad (7-78)$$

[0361] 变量lmcsCW[i]被推导为如下:

[0362] -对于 $i = 0 \dots \text{lmcs_min_bin_idx} - 1$, lmcsCW[i]被设置为等于0。

[0363] -对于 $i = \text{lmcs_min_bin_idx} \dots \text{LmcsMaxBinIdx}$, 以下适用:

[0364]
$$\text{lmcsCW}[i] = \text{OrgCW} + \text{lmcsDeltaCW}[i] \quad (7-79)$$

[0365] lmcsCW[i]的值应在 $(\text{OrgCW} \gg 3)$ 至 $(\text{OrgCW} \ll 3 - 1)$ 的范围中(含端值)。

[0366] -对于 $i = \text{LmcsMaxBinIdx} + 1 \dots 15$, lmcsCW[i]被设置为等于0。

[0367] 满足以下条件是比特流一致性的要求:

[0368]
$$\sum_{i=0}^{15} \text{lmcsCW}[i] \leq (1 \ll \text{BitDepth}_Y) - 1 \quad (7-80)$$

[0369] 变量InputPivot[i] (其中, $i = 0 \dots 16$) 被推导为如下:

[0370] $InputPivot[i] = i * OrgCW$

[0371] 变量 $LmcsPivot[i]$ (其中, $i = 0..16$)、变量 $ScaleCoeff[i]$ 和 $InvScaleCoeff[i]$ (其中, $i = 0..15$) 被推导为如下:

```

     $LmcsPivot[0] = 0;$ 
    for( $i = 0; i \leq 15; i++$ ) {
         $LmcsPivot[i + 1] = LmcsPivot[i] + lmcscw[i]$ 
         $ScaleCoeff[i] = (lmcscw[i] * (1 \ll 14) + (1 \ll (Log2(OrgCW) - 1)))$ 
         $\gg (Log2(OrgCW))$  (7-82)
    [0372] if( $lmcscw[i] = 0$ )
         $InvScaleCoeff[i] = 0$ 
    else
         $InvScaleCoeff[i] = OrgCW * (1 \ll 14) / lmcscw[i]$ 
    }

```

[0373] 变量 $ChromaScaleCoeff[i]$ (其中, $i = 0..15$) 被推导为如下:

[0374] $chromaResidualScaleLut[] = \{16384, 16384, 16384, 16384, 16384, 16384, 16384, 8192, 8192, 8192, 8192, 5461, 5461, 5461, 5461, 4096, 4096, 4096, 4096, 3277, 3277, 3277, 3277, 2731, 2731, 2731, 2731, 2341, 2341, 2341, 2048, 2048, 2048, 1820, 1820, 1820, 1638, 1638, 1638, 1489, 1489, 1489, 1489, 1365, 1365, 1365, 1365, 1260, 1260, 1260, 1260, 1170, 1170, 1170, 1170, 1092, 1092, 1092, 1092, 1024, 1024, 1024, 1024\}$

```

    if( $lmcscw[i] = 0$ )
         $ChromaScaleCoeff[i] = (1 \ll 11)$ 
    else {
    [0375]      $binCW = BitDepth_Y > 10 ? (lmcscw[i] \gg (BitDepth_Y - 10)) : (7-83)$ 
         $BitDepth_Y < 10 ? (lmcscw[i] \ll (10 - BitDepth_Y)) : lmcscw[i]$ 
         $ChromaScaleCoeff[i] = chromaResidualScaleLut[Clip3(1, 64, binCW \gg 1) - 1]$ 
    }

```

[0376] 变量 $ClipRange$ 、 $LmcsMinVal$ 和 $LmcsMaxVal$ 被推导为如下:

[0377] $ClipRange = ((lmc_min_bin_idx > 0) \&\& (LmcsMaxBinIdx < 15))$ (7-84)

[0378] $LmcsMinVal = 16 \ll (BitDepth_Y - 8)$ (7-85)

[0379] $LmcsMaxVal = 235 \ll (BitDepth_Y - 8)$ (7-86)

[0380] 注意——仅当 $slice_lmc_model_present_flag$ 等于 1 时才更新阵列 $InputPivot[i]$ 和 $LmcsPivot[i]$ 、 $ScaleCoeff[i]$ 、以及 $InvScaleCoeff[i]$ 、 $ChromaScaleCoeff[i]$ 、 $ClipRange$ 、 $LmcsMinVal$ 和 $LmcsMaxVal$ 。因此, 例如, 可以与 IRAP 图片一起发送 lmc 模型, 但是对该 IRAP 图片禁用 lmc 。

[0381] 2.8. 示例性色度残差的联合编解码

[0382] 该测试测量仅对联合残差进行编解码的示例性联合Cb-Cr编解码模式的性能。

[0383] 如果色度重整器是有效的,则重整器应用于接收的残差,与在单独编解码模式下完成的相同(即,重整器联合残差信令)。在编码器侧,当测试此模式时,正的Cb残差和负的Cr残差的平均值用作联合残差:

[0384] $resJoint = (resCb - resCr) / 2$

[0385] 在比特流中信令通知一个bin指示符以启用该模式。在该模式被启用的情况下,联合残差信令被编解码在比特流中。在解码器侧,联合残差用于Cb分量,而负版本残差应用于Cr。

[0386] 预期该模式的信令通知以遵循以下给出的语法和语义:

	<code>residual_coding(x0, y0, log2TbWidth, log2TbHeight, cIdx) {</code>	描述符
	<code><u>if(cIdx == 2 && tu_cb_cr_joint_residual[x0][y0]) {</u></code>	
	<code><u>tu_cb_cr_joint_residual[x0][y0]</u></code>	<u>ae(v)</u>
[0387]	<code><u>if(tu_cb_cr_joint_residual [x0][y0]) {</u></code>	
	<code><u>return</u></code>	
	<code><u>}</u></code>	
	<code>...</code>	

[0388] `tu_cb_cr_joint_residual[x0][y0]` 指定是否使用指示的Cb残差来导出Cb和Cr残差的二者。阵列索引x0、y0指定所考虑的变换块的左上亮度样本相对于图片的左上亮度样本的位置(x0,y0)。

[0389] 7.3.2.2图片参数集RBSP语法

	<code>pic_parameter_set_rbsp() {</code>	描述符
	<code>...</code>	
	<code>pps_cb_qp_offset</code>	se(v)
[0390]	<code>pps_cr_qp_offset</code>	se(v)
	<code><u>pps_joint_cbr_qp_offset</u></code>	<u>se(v)</u>
	<code>...</code>	
	<code>}</code>	

[0391] 7.3.4.1常规片组标头语法

	tile_group_header() {	描述符
	...	
	if(pps_tile_group_chroma_qp_offsets_present_flag) {	
	tile_group_cb_qp_offset	se(v)
[0392]	tile_group_cr_qp_offset	se(v)
	<u>tile_group_joint_cbr_qp_offset</u>	<u>se(v)</u>
	}	
	...	
	}	

[0393] 7.3.6.12残差编解码语法

[0394]	residual_coding(x0, y0, log2TbWidth, log2TbHeight, cIdx) {	描述符
	<u>if(cIdx == 2 && tu_cbf_cb[x0 y0]) {</u>	
	<u>tu_joint_cbr_residual[x0 y0]</u>	<u>ae(v)</u>
	<u>if(tu_joint_cbr_residual [x0 y0]) {</u>	
[0395]	<u>return</u>	
	<u>}</u>	
	...	
	}	

[0396] 2.9约化的二次变换

[0397] JEM中,在正向一次变换和量化之间(编码器处)以及在去量化和逆一次变换之间(解码器侧)应用二次变换。如图6所示,执行 4×4 (或 8×8)二次变换,这取决于块尺寸。例如,每个 8×8 块,将 4×4 二次变换应用于较小的块(例如, $\min(\text{宽度}, \text{高度}) < 8$),将 8×8 二次变换应用于较大的块(例如, $\min(\text{宽度}, \text{高度}) > 4$)。

[0398] 对于二次变换,应用不可分变换,因此也被称为不可分二次变换(NSST)。共有35个变换集,并且每个变换集使用3个不可分变换矩阵(核,每个核有 16×16 矩阵)。

[0399] 在一些实施例中可以使用根据帧内预测方向的约化的二次变换(RST)和4个变换集(而不是35个变换集)映射。在某些情况下, 16×48 和 16×16 矩阵分别用于 8×8 和 4×4 块。为了便于注释, 16×48 变换表示为RST 8×8 ,并且 16×16 变换表示为RST 4×4 。这种方法最近被VVC采用。

[0400] 图7示出了约化的二次变换(RST)的示例。二次正变换和逆变换是与一次变换分离的过程步骤

[0401] 对于编码器,首先执行一次正变换,然后执行二次正变换和量化,以及CABAC比特编码。对于解码器,首先执行CABAC比特解码和逆量化,然后执行二次逆变换,然后一次逆变换。

[0402] RST仅应用于帧内编解码Tus。

[0403] 2.9.1.语法和语义

[0404] 7.3.2.1序列参数集RBSP语法

	seq_parameter_set_rbsp() {	描述符
	
[0405]	sps_mts_enabled_flag	u(1)
	if(sps_mts_enabled_flag) {	

	sps_explicit_mts_intra_enabled_flag	u(1)
	sps_explicit_mts_inter_enabled_flag	u(1)
	}	

[0406]

	
	sps_st_enabled_flag	u(1)
	
	}	

[0407] 7.3.6.11残差编解码语法

[0408]

residual_coding(x0, y0, log2TbWidth, log2TbHeight, cIdx) {	描述符
if((tu_mts_idx[x0][y0] > 0 (cu_sbt_flag && log2TbWidth < 6 && log2TbHeight < 6)) && cIdx == 0 && log2TbWidth > 4)	
log2TbWidth = 4	
else	
log2TbWidth = Min(log2TbWidth, 5)	
if(tu_mts_idx[x0][y0] > 0 (cu_sbt_flag && log2TbWidth < 6 && log2TbHeight < 6)) && cIdx == 0 && log2TbHeight > 4)	
log2TbHeight = 4	
else	
log2TbHeight = Min(log2TbHeight, 5)	
if(log2TbWidth > 0)	
last_sig_coeff_x_prefix	ac(v)
if(log2TbHeight > 0)	
last_sig_coeff_y_prefix	ac(v)
if(last_sig_coeff_x_prefix > 3)	
last_sig_coeff_x_suffix	ac(v)
if(last_sig_coeff_y_prefix > 3)	
last_sig_coeff_y_suffix	ac(v)
log2SbW = (Min(log2TbWidth, log2TbHeight) < 2 ? 1 : 2)	
log2SbH = log2SbW	
if(log2TbWidth < 2 && cIdx == 0) {	
log2SbW = log2TbWidth	

	$\log_2\text{SbH} = 4 - \log_2\text{SbW}$	
	} else if ($\log_2\text{TbHeight} < 2$ && $\text{cIdx} == 0$) {	
	$\log_2\text{SbH} = \log_2\text{TbHeight}$	
	$\log_2\text{SbW} = 4 - \log_2\text{SbH}$	
	}	
	$\text{numSbCoeff} = 1 \ll (\log_2\text{SbW} + \log_2\text{SbH})$	
	$\text{lastScanPos} = \text{numSbCoeff}$	
	$\text{lastSubBlock} = (1 \ll (\log_2\text{TbWidth} + \log_2\text{TbHeight} - (\log_2\text{SbW} + \log_2\text{SbH}))) - 1$	
	do {	
	if($\text{lastScanPos} == 0$) {	
	$\text{lastScanPos} = \text{numSbCoeff}$	
	$\text{lastSubBlock}--$	
	}	
	$\text{lastScanPos}--$	
[0409]	$\text{xS} = \text{DiagScanOrder}[\log_2\text{TbWidth} - \log_2\text{SbW}][\log_2\text{TbHeight} - \log_2\text{SbH}][\text{lastSubBlock}][0]$	
	$\text{yS} = \text{DiagScanOrder}[\log_2\text{TbWidth} - \log_2\text{SbW}][\log_2\text{TbHeight} - \log_2\text{SbH}][\text{lastSubBlock}][1]$	
	$\text{xC} = (\text{xS} \ll \log_2\text{SbW}) + \text{DiagScanOrder}[\log_2\text{SbW}][\log_2\text{SbH}][\text{lastScanPos}][0]$	
	$\text{yC} = (\text{yS} \ll \log_2\text{SbH}) + \text{DiagScanOrder}[\log_2\text{SbW}][\log_2\text{SbH}][\text{lastScanPos}][1]$	
	} while(($\text{xC} \neq \text{LastSignificantCoeffX}$) ($\text{yC} \neq \text{LastSignificantCoeffY}$))	
	$\text{QState} = 0$	
	for($i = \text{lastSubBlock}; i \geq 0; i--$) {	
	$\text{startQStateSb} = \text{QState}$	
	$\text{xS} = \text{DiagScanOrder}[\log_2\text{TbWidth} - \log_2\text{SbW}][\log_2\text{TbHeight} - \log_2\text{SbH}][\text{lastSubBlock}][0]$	

[0410]

yS	=	
DiagScanOrder[log2TbWidth - log2SbW][log2TbHeight - log2SbH]		
[lastSubBlock][1]		
inferSbDcSigCoeffFlag = 0		
if((i < lastSubBlock) && (i > 0)) {		
coded_sub_block_flag [xS][yS]		ae(v)
inferSbDcSigCoeffFlag = 1		
}		
firstSigScanPosSb = numSbCoeff		
lastSigScanPosSb = -1		
remBinsPass1 = ((log2SbW + log2SbH) < 4 ? 8 : 32)		
firstPosMode0 = (i == lastSubBlock ? lastScanPos : numSbCoeff - 1)		
firstPosMode1 = -1		
for(n = firstPosMode0; n >= 0 && remBinsPass1 >= 4; n--) {		
xC = (xS << log2SbW) + DiagScanOrder[log2SbW][log2SbH][n][0]		
yC = (yS << log2SbH) + DiagScanOrder[log2SbW][log2SbH][n][1]		
if(coded_sub_block_flag [xS][yS] && (n > 0		
!inferSbDcSigCoeffFlag) &&		
(xC != LastSignificantCoeffX yC != LastSignificantCoeffY) {		
sig_coeff_flag [xC][yC]		ae(v)
remBinsPass1--		
if(sig_coeff_flag [xC][yC])		
inferSbDcSigCoeffFlag = 0		
}		
if(sig_coeff_flag [xC][yC]) {		
if(!transform skip flag[x0][y0]) {		
numSigCoeff++		
if((((log2TbWidth == 2 && log2TbHeight == 2)		
 (log2TbWidth == 3 && log2TbHeight == 3)) && n >=		
8 && i == 0) ((log2TbWidth >= 3 && log2TbHeight		
>= 3 && (i == 1 i == 2)))) {		
numZeroOutSigCoeff++		
}		
}		

	abs_level_gt1_flag[n]	ae(v)
	remBinsPass1--	
	if(abs_level_gt1_flag[n]) {	
	par_level_flag[n]	ae(v)
	remBinsPass1--	
	abs_level_gt3_flag[n]	ae(v)
	remBinsPass1--	
	}	
	if(lastSigScanPosSb == -1)	
	lastSigScanPosSb = n	
	firstSigScanPosSb = n	
	}	
	AbsLevelPass1[xC][yC] = sig_coeff_flag[xC][yC] +	
	par_level_flag[n] +	
	abs_level_gt1_flag[n] + 2 * abs_level_gt3_flag[n]	
	if(dep_quant_enabled_flag)	
	QState = QStateTransTable[QState][AbsLevelPass1[xC][yC] & 1]	
[0411]	if(remBinsPass1 < 4)	
	firstPosMode1 = n - 1	
	}	
	for(n = numSbCoeff - 1; n >= firstPosMode1; n--) {	
	xC = (xS << log2SbW) + DiagScanOrder[log2SbW][log2SbH][n][0]	
	yC = (yS << log2SbH) + DiagScanOrder[log2SbW][log2SbH][n][1]	
	if(abs_level_gt3_flag[n])	
	abs_remainder[n]	ae(v)
	AbsLevel[xC][yC] = AbsLevelPass1[xC][yC] + 2 * abs_remainder[n]	
	}	
	... }	
	}	
	}	
	}	
	}	
	}	
	}	
	}	

[0412] 7.3.6.5编解码单元语法

[0413]

coding_unit(x0, y0, cbWidth, cbHeight, treeType) {	描述符
if(tile_group_type != I sps_IBC_enabled_flag) {	
if(treeType != DUAL_TREE_CHROMA)	
cu_skip_flag [x0][y0]	ac(v)
if(cu_skip_flag[x0][y0] == 0 && tile_group_type != I)	
pred_mode_flag	ac(v)
if(((tile_group_type == I && cu_skip_flag[x0][y0] == 0) (tile_group_type != I && CuPredMode[x0][y0] != MODE_INTRA)) && sps_IBC_enabled_flag)	
pred_mode_IBC_flag	ac(v)
}	
...	
if(!pcm_flag[x0][y0]) {	
if(CuPredMode[x0][y0] != MODE_INTRA && merge_flag[x0][y0] == 0)	
cu_cbf	ac(v)

[0414]

if(cu_cbf) {	
if(CuPredMode[x0][y0] == MODE_INTER && sps_sbt_enabled_flag && !ciip_flag[x0][y0]) {	
if(cbWidth <= MaxSbtSize && cbHeight <= MaxSbtSize) {	
allowSbtVerH = cbWidth >= 8	
allowSbtVerQ = cbWidth >= 16	
allowSbtHorH = cbHeight >= 8	
allowSbtHorQ = cbHeight >= 16	
if(allowSbtVerH allowSbtHorH allowSbtVerQ allowSbtHorQ)	
cu_sbt_flag	ae(v)
}	
if(cu_sbt_flag) {	
if((allowSbtVerH allowSbtHorH) && (allowSbtVerQ allowSbtHorQ))	
cu_sbt_quad_flag	ae(v)
if((cu_sbt_quad_flag && allowSbtVerQ && allowSbtHorQ) (!cu_sbt_quad_flag && allowSbtVerH && allowSbtHorH))	
cu_sbt_horizontal_flag	ae(v)
cu_sbt_pos_flag	ae(v)
}	
}	
<u>numZeroOutSigCoeff = 0</u>	
<u>transform tree(x0, y0, cbWidth, cbHeight, treeType)</u>	
<u>if(Min(cbWidth, cbHeight) >= 4 && sps st enabled flag</u> <u>== 1 && CuPredMode[x0][y0] == MODE_INTRA</u> <u>&& IntraSubPartitionsSplitType == ISP_NO_SPLIT) {</u>	
<u>if((numSigCoeff > ((treeType == SINGLE_TREE) ?</u> <u>2 : 1)) &&</u> <u>numZeroOutSigCoeff == 0) {</u>	
<u>st_idx[x0][y0]</u>	ae(v)
<u>}</u>	
<u>}</u>	

	}	
[0415]	}	
	}	

[0416] 7.4.3.1序列参数集RBSP语义

[0417] sps_st_enabled_flag等于1指定在帧内编解码单元的残差编解码语法中可能存在st_idx。sps_st_enabled_flag等于0指定在帧内编解码单元的残差编解码语法中不存在st_idx。

[0418] 7.4.7.5编解码单元语义

[0419] st_idx[x0][y0]指定在所选择的变换集中的两个候选核之间应用哪个二次变换核。st_idx[x0][y0]等于0指定不应用二次变换。阵列索引x0、y0指定所考虑的变换块的左上样本相对于图片的左上样本的位置(x0,y0)。

[0420] 当st_idx[x0][y0]不存在时,推断st_idx[x0][y0]等于0。

[0421] 2.10.VVC中各种色彩格式的支持

[0422] 2.10.1.语法、语义和解码过程

[0423] 7.3.2.3序列参数集RBSP语法

	seq_parameter_set_rbsp() {	描述符
	sps_decoding_parameter_set_id	u(4)
	sps_max_sub_layers_minus1	u(3)
	sps_reserved_zero_5bits	u(5)
	profile_tier_level(sps_max_sub_layers_minus1)	
	gra_enabled_flag	u(1)
	sps_seq_parameter_set_id	ue(v)
	<u>chroma format idc</u>	<u>ue(v)</u>
[0424]	<u>if(chroma format idc == 3)</u>	
	<u>separate colour plane flag</u>	<u>u(1)</u>
	pic_width_in_luma_samples	ue(v)
	pic_height_in_luma_samples	ue(v)
	conformance_window_flag	u(1)
	if(conformance_window_flag) {	
	conf_win_left_offset	ue(v)
	conf_win_right_offset	ue(v)
	conf_win_top_offset	ue(v)

[0425]

conf_win_bottom_offset	ue(v)
}	
bit_depth_luma_minus8	ue(v)
bit_depth_chroma_minus8	ue(v)
log2_max_pic_order_cnt_lsb_minus4	ue(v)
sps_sub_layer_ordering_info_present_flag	u(1)
for(i = (sps_sub_layer_ordering_info_present_flag ? 0 : sps_max_sub_layers_minus1); i <= sps_max_sub_layers_minus1; i++) {	
sps_max_dec_pic_buffering_minus1[i]	ue(v)
sps_max_num_reorder_pics[i]	ue(v)
sps_max_latency_increase_plus1[i]	ue(v)
}	
long_term_ref_pics_flag	u(1)
sps_idr_rpl_present_flag	u(1)
rpl1_same_as_rpl0_flag	u(1)
for(i = 0; i < !rpl1_same_as_rpl0_flag ? 2 : 1; i++) {	
num_ref_pic_lists_in_sps[i]	ue(v)
for(j = 0; j < num_ref_pic_lists_in_sps[i]; j++)	
ref_pic_list_struct(i, j)	
}	
qtbtt_dual_tree_intra_flag	u(1)
log2_ctu_size_minus2	ue(v)
log2_min_luma_coding_block_size_minus2	ue(v)
partition_constraints_override_enabled_flag	u(1)
sps_log2_diff_min_qt_min_cb_intra_slice_luma	ue(v)
sps_log2_diff_min_qt_min_cb_inter_slice	ue(v)
sps_max_mtt_hierarchy_depth_inter_slice	ue(v)
sps_max_mtt_hierarchy_depth_intra_slice_luma	ue(v)
if(sps_max_mtt_hierarchy_depth_intra_slice_luma != 0) {	
sps_log2_diff_max_bt_min_qt_intra_slice_luma	ue(v)
sps_log2_diff_max_tt_min_qt_intra_slice_luma	ue(v)
}	
if(sps_max_mtt_hierarchy_depth_inter_slices != 0) {	

	sps_log2_diff_max_bt_min_qt_inter_slice	ue(v)
	sps_log2_diff_max_tt_min_qt_inter_slice	ue(v)
	}	
	if(qtbtt_dual_tree_intra_flag) {	
	sps_log2_diff_min_qt_min_cb_intra_slice_chroma	ue(v)
	sps_max_mtt_hierarchy_depth_intra_slice_chroma	ue(v)
	if (sps_max_mtt_hierarchy_depth_intra_slice_chroma != 0) {	
	sps_log2_diff_max_bt_min_qt_intra_slice_chroma	ue(v)
	sps_log2_diff_max_tt_min_qt_intra_slice_chroma	ue(v)
[0426]	}	
	}	
	...	
	sps_extension_flag	u(1)
	if(sps_extension_flag)	
	while(more_rbsp_data())	
	sps_extension_data_flag	u(1)
	rbsp_trailing_bits()	
	}	

[0427] chroma_format_idc指定相对于第6.2条中指定的亮度采样的色度采样。chroma_format_idc的值应在0到3之间的范围中(含端值)。

[0428] separate_colour_plane_flag等于1指定4:4:4色度格式的三个色彩分量被单独编解码。separate_colour_plane_flag等于0指定色彩分量没有被单独编解码。separate_colour_plane_flag不存在时,则推断等于0。当separate_colour_plane_flag等于1时,编解码图片由三个独立的分量构成,每个分量由一个色彩平面(Y、Cb或Cr)的编解码样本构成,并使用单色编解码语法。在这种情况下,每个色彩平面都与特定的colour_plane_id值相关联。

[0429] 具有不同的colour_plane_id值的色彩平面之间的解码过程不存在相关性。例如,具有一个colour_plane_id值的单色图片的解码过程不使用来自具有不同colour_plane_id值的单色图片的任何数据来进行帧间预测。

[0430] 根据separate_colour_plane_flag的值,变量ChromaArrayType的值分配如下:

[0431] -如果separate_colour_plane_flag等于0,则ChromaArrayType被设置为等于chroma_format_idc。

[0432] -否则(separate_colour_plane_flag等于1),则ChromaArrayType被设置等于0。

[0433] 6.2源、解码和输出图片格式

[0434] 本节指定通过比特流给定的源图片与解码图片之间的关系。

[0435] 由比特流表示的视频源是按解码顺序排列的图片的序列。

[0436] 源图片和解码图片均由一个或多个样本阵列构成:

[0437] -只有亮度(Y) (单色)。

[0438] -亮度和两个色度(YCbCr或YCgCo)。

[0439] -绿色、蓝色和红色(GBR,也称为RGB)。

[0440] -表示其他未指定的单色或三刺激色采样的阵列(例如,YZX,也称为XYZ)。

[0441] 为了便于本说明书中的符号和术语,与这些阵列相关联的变量和术语被称为亮度(或者L或Y)和色度,其中两个色度阵列被称为Cb和Cr;与使用的实际色彩表示方法无关。使用的实际色彩表示方法可以用附录TBD中指定的语法指示。

[0442] 根据色度格式采样结构(通过chroma_format_idc和separate_colour_plane_flag指定),表6中指定变量SubWidthC和SubHeightC。未来,ITU-T|ISO/IEC可指定chroma_format_idc、SubWidthC和SubHeightC的其他值。

[0443] 表6—从chroma_format_idc和separate_colour_plane_flag推导的SubWidthC和SubHeightC值

<i>chroma_format_idc</i>	<i>separate_colour_plane_flag</i>	色度格式	<i>SubWidthC</i>	<i>SubHeightC</i>
<i>c</i>	<i>ag</i>		<i>C</i>	<i>C</i>
0	0	单色	1	1
1	0	4:2:0	2	2
2	0	4:2:2	2	1
3	0	4:4:4	1	1
3	1	4:4:4	1	1

[0444] 单色采样中,只有一个样本阵列,被标称地认为是亮度阵列。

[0445] 4:2:0采样中,两个色度阵列中的每一个具有亮度阵列的一半的高度和一半的宽度。

[0446] 4:2:2采样中,两个色度阵列中的每一个具有与亮度阵列相同的高度和一半的宽度。

[0447] 4:4:4采样中,根据separate_colour_plane_flag的值,以下适用:

[0448] -如果separate_colour_plane_flag等于0,则两个色度阵列中的每一个具有与亮度阵列相同的高度和宽度。

[0449] -否则(separate_colour_plane_flag等于1),三个色彩平面分别作为单色采样的图片进行处理。

[0450] 视频序列中亮度和色度阵列中每个样本的表示所需的比特数在8到16之间(含端值),并且亮度阵列中使用的比特数可以不同于色度阵列中使用的比特数。

[0451] 当chroma_format_idc的值等于1时,图8示出了图片中亮度和色度样本的标称垂直和水平相对位置。可在视频可用性信息中指示替代色度样本相对位置。

[0452] 当chroma_format_idc的值等于2时,色度样本与对应的亮度样本同址,并且图片中的标称位置如图9所示。

[0453] 当chroma_format_idc的值等于3时,所有情况下的图片的所有阵列样本同址,并且图片中的标称位置如图10所示。

[0454] 附件E:视频可用性信息

[0456] 即使对于相同的色度子采样格式,例如4:2:0,子采样方法也可以有多个不同的方式。图11中标记不同的类型。

[0457] 图12图示了当chroma_format_idc等于1(4:2:0色度格式)且chroma_sample_loc_type等于变量ChromaLocType的值时左上色度样本的指示相对位置。由左上4:2:0色度样本表示的区域(描绘为其中中心有一大红点的大红正方形)相对于由左上亮度样本表示的区域(描绘为其中中心有一小黑点的小黑正方形)显示。邻近的亮度样本所代表的区域被描绘成它们中心有小灰点的小灰正方形。

[0458] 如图13所图示,色度样本的相对空域定位可以通过将两个变量HorizontalOffsetC和VerticalOffsetC限定为如表7中给定的chroma_format_idc和变量ChromaLocType的函数来表示,其中HorizontalOffsetC是在亮度样本的单位中,左上色度样本的中心相对于左上亮度样本的中心的水平(x)位置;VerticalOffsetC是在亮度样本的单位中,左上色度样本的中心相对于左上亮度样本的中心的垂直(y)位置。

[0459] 在典型的FIR滤波器设计中,当chroma_format_idc等于1(4:2:0色度格式)或2(4:2:2色度格式)时,HorizontalOffsetC和VerticalOffsetC将分别作为水平和垂直滤波器操作的相位偏移,用于从4:4:4色度格式到由chroma_format_idc指示的色度格式的可分下采样。

[0460] 表7-作为chroma_format_idc和ChromaLocType的函数的HorizontalOffsetC和VerticalOffsetC的定义

chroma_format_idc	ChromaLocType	HorizontalOffsetC	VerticalOffsetC
1 (4:2:0)	0	0	0.5
1 (4:2:0)	1	0.5	0.5
1 (4:2:0)	2	0	0
1 (4:2:0)	3	0.5	0
1 (4:2:0)	4	0	1
1 (4:2:0)	5	0.5	1
2 (4:2:2)	—	0	0
3 (4:4:4)	—	0	0

[0462] 当chroma_format_idc等于1(4:2:0色度格式)且意图根据Rec.ITU-R BT.2020-2或Rec.ITU-R BT.2100-1解释解码的视频内容时,chroma_loc_info_present_flag应等于1,并且chroma_sample_loc_type应等于2。

[0463] 2.11. 示例性扩展的ALF

[0464] 在一些实施例中,可以扩展非线性ALF概念。更准确地说,可以添加一些语法元素,以启用替代ALF亮度滤波器集和替代色度滤波器的使用,并基于每个滤波器、每个CTU选择它们。对于每个CTU,如果在亮度分量上启用ALF滤波,则会为每个ALF亮度滤波索引信令通知附加的亮度滤波器集替代索引。类似地,对于每个CTU,对于启用ALF滤波的每个色度分量,信令通知替代色度滤波器索引,以为色度分量选择给定的滤波器。

[0465] 2.12 VVC中示例性子块变换(SBT)

[0466] 对于具有等于1的`cu_cbf` (也称为`cu_coded_flag`)的帧间预测的CU,可以信令通知`cu_sbt_flag`以指示是解码整个残差块还是解码残差块的子部分。在前一种情况下,进一步解析帧间MTS的信息以确定CU的变换类型。在后一种情况下,用推断出的自适应变换对残差块的一部分进行编解码,并将残差块的其他部分归零。SBT不应用于组合帧间帧内模式,因为几乎没有实现编解码增益。

[0467] 2.12.1子块TU平铺

[0468] 当SBT用于帧间CU时,从比特流中进一步解码SBT类型和SBT位置信息。有两种SBT类型和两个SBT位置,如图14所示。对于SBT-V(或SBT-H),TU宽度(或高度)可以等于CU宽度(或高度)的一半或CU宽度(或高度)的1/4,由另一个标志信令通知,导致2:2划分或1:3/3:1划分。2:2划分类似于二叉树(BT)划分,而1:3/3:1划分类似于非对称二叉树(ABT)划分。如果在亮度样本中,CU的一侧为8,则不允许沿该侧的1:3/3:1划分。因此,CU最多有8个SBT模式。

[0469] 四叉树(QT)划分进一步用于将一个CU平铺成4个子块,并且仍然有一个子块有残差,如图15所示。该SBT类型表示为SBT-Q。VVC未采用该部分。

[0470] 对于宽度和高度均不大于`maxSbtSize`的CU允许SBT-V、SBT-H和SBT-Q。在SPS中信令通知`maxSbtSize`。对于HD和4K序列,编码器将`maxSbtSize`设置为64;对于其他分辨率较小的序列,`maxSbtSize`被设置为32。

[0471] 2.12.2子块的变换类型

[0472] 位置相关变换应用于SBT-V和SBT-H中的亮度变换块(色度TB始终使用DCT-2)。SBT-H和SBT-V的两个位置与不同的核心变换相关联。更具体地说,图14中指定了每个SBT位置的水平和垂直变换。例如,SBT-V位置0的水平和垂直变换分别为DCT-8和DST-7。当残差TU的一侧大于32时,将对应的变换设置为DCT-2。因此,子块变换联合指定残差块的TU平铺、`cbf`/编解码标志以及水平和垂直变换,这对于块的主要残差在块的一侧的情况,可以被视为语法快捷方式。

[0473] 2.12.3 VVC中的子块变换

[0474] 对于具有等于1的`cu_cbf` (或`cu_coded_flag`) (例如具有至少一个非零系数)的帧间预测的CU,可以信令通知`cu_sbt_flag`,以指示是解码整个残差块还是解码残差块的子部分。在前一种情况下,进一步解析帧间MTS信息以确定CU的变换类型。在后一种情况下,用推断出的自适应变换对残差块的部分进行编解码,并将残差块的其他部分归零。SBT不应用于组合的帧间帧内模式,因为几乎没有实现编解码增益。

[0475] 当SBT用于帧间CU时,从比特流进一步解码SBT类型和SBT位置信息。对于SBT-V(或SBT-H),TU宽度(或高度)可以等于CU宽度(或高度)的一半或CU宽度(或高度)的1/4,由另一个标志信令通知,导致2:2划分或1:3/3:1划分。2:2划分类似于二叉树(BT)划分,而1:3/3:1划分类似于非对称二叉树(ABT)划分。如果在亮度样本中,CU的一侧为8,则不允许沿该侧的1:3/3:1划分。因此,CU最多有8个SBT模式。

[0476] 依赖于位置的变换应用于SBT-V和SBT-H中的亮度变换块(色度TB始终使用DCT-2)。SBT-H和SBT-V的两个位置与不同的核心变换相关联。更具体地说,图14中指定每个SBT位置的水平和垂直变换。例如,SBT-V位置0的水平和垂直变换分别为DCT-8和DST-7。当残差

TU的一侧大于32时,将对应的变换设置为DCT-2。因此,子块变换联合指定残差块的TU平铺、cbf/编解码标志以及水平和垂直变换,这对于块的主要残差在块的一侧的情况,可以被视为语法快捷方式。

[0477] 7.3.4.6编解码单元语法

	描述符
<code>coding_unit(x0, y0, cbWidth, cbHeight, treeType) {</code>	
<code> if(tile_group_type != 1) {</code>	
<code> if(treeType != DUAL_TREE_CHROMA)</code>	
<code> cu_skip_flag[x0][y0]</code>	ac(v)
<code> if(cu_skip_flag[x0][y0] == 0)</code>	
<code> pred_mode_flag</code>	ac(v)
<code> }</code>	
<code> if(CuPredMode[x0][y0] == MODE_INTRA) {</code>	
<code> ...</code>	
<code> } else if(treeType != DUAL_TREE_CHROMA) { /* MODE_INTER */</code>	
<code> ...</code>	
<code> }</code>	
<code> if(!pcm_flag[x0][y0]) {</code>	

[0478]

	if(CuPredMode[x0][y0] != MODE_INTRA && merge_flag[x0][y0] == 0)	
	cu_cbf	ae(v)
	if(cu_cbf) {	
	if(CuPredMode[x0][y0] == MODE_INTER && sps_sbt_enabled_flag && !ciip_flag[x0][y0]) {	
	if(cbWidth <= MaxSbtSize && cbHeight <= MaxSbtSize) {	
	allowSbtVerH = cbWidth >= 8	
	allowSbtVerQ = cbWidth >= 16	
	allowSbtHorH = cbHeight >= 8	
	allowSbtHorQ = cbHeight >= 16	
	if(allowSbtVerH allowSbtHorH allowSbtVerQ allowSbtHorQ)	
	cu_sbt_flag	ae(v)
[0479]	}	
	if(cu_sbt_flag) {	
	if((allowSbtVerH allowSbtHorH) && (allowSbtVerQ allowSbtHorQ))	
	cu_sbt_quad_flag	ae(v)
	if((cu_sbt_quad_flag && allowSbtVerQ && allowSbtHorQ) (!cu_sbt_quad_flag && allowSbtVerH && allowSbtHorH))	
	cu_sbt_horizontal_flag	ae(v)
	cu_sbt_pos_flag	ae(v)
	}	
	}	
	transform_tree(x0, y0, cbWidth, cbHeight, treeType)	
	}	
	}	
	}	

[0480] 7.3.4.10变换树语法

[0481]

transform_tree(x0, y0, tbWidth, tbHeight , treeType) {	描述符
InferTuCbfLuma = 1	
if(IntraSubPartSplitType == NO_ISP_SPLIT) {	
if(tbWidth > MaxTbSizeY tbHeight > MaxTbSizeY) {	
trafoWidth = (tbWidth > MaxTbSizeY) ? (tbWidth / 2) : tbWidth	
trafoHeight = (tbHeight > MaxTbSizeY) ? (tbHeight / 2) : tbHeight	
transform_tree(x0, y0, trafoWidth, trafoHeight)	
if(tbWidth > MaxTbSizeY)	
transform_tree(x0 + trafoWidth, y0, trafoWidth, trafoHeight, treeType)	
if(tbHeight > MaxTbSizeY)	
transform_tree(x0, y0 + trafoHeight, trafoWidth, trafoHeight, treeType)	
if(tbWidth > MaxTbSizeY && tbHeight > MaxTbSizeY)	
transform_tree(x0 + trafoWidth, y0 + trafoHeight, trafoWidth, trafoHeight, treeType) } else {	
transform_unit(x0, y0, tbWidth, tbHeight, treeType, 0)	
}	
} else if(cu_sbt_flag) {	
if(!cu_sbt_horizontal_flag) {	
trafoWidth = tbWidth * SbtNumFourthsTb0 / 4	
transform_unit(x0, y0, trafoWidth, tbHeight, treeType , 0)	
transform_unit(x0 + trafoWidth, y0, tbWidth - trafoWidth , tbHeight, treeType , 1)	
} else {	
trafoHeight = tbHeight * SbtNumFourthsTb0 / 4	
transform_unit(x0, y0, tbWidth, trafoHeight, treeType , 0)	
transform_unit(x0, y0 + trafoHeight, tbWidth, tbHeight - trafoHeight, treeType , 1)	
}	
} else if(IntraSubPartitionsSplitType == ISP_HOR_SPLIT) {	
trafoHeight = tbHeight / NumIntraSubPartitions	

	for(partIdx = 0; partIdx < NumIntraSubPartitions; partIdx++)	
	transform_unit(x0, y0 + trafoHeight * partIdx, tbWidth, trafoHeight, treeType, partIdx)	
	} else if(IntraSubPartitionsSplitType == ISP_VER_SPLIT) {	
	trafoWidth = tbWidth / NumIntraSubPartitions	
[0482]	for(partIdx = 0; partIdx < NumIntraSubPartitions; partIdx++)	
	transform_unit(x0 + trafoWidth * partIdx, y0, trafoWidth, tbHeight, treeType, partIdx)	
	}	
	}	

[0483] 7.3.4.11变换单元语法

	transform_unit(x0, y0, tbWidth, tbHeight, treeType, subTuIndex) {	描述符
	if(treeType == SINGLE_TREE treeType == DUAL_TREE_LUMA) {	
	if((IntraSubPartitionsSplitType == ISP_NO_SPLIT && !(cu_sbt_flag &&	
	((subTuIndex == 0 && cu_sbt_pos_flag) (subTuIndex == 1 && !cu_sbt_pos_flag)))	
	(IntraSubPartitionsSplitType != ISP_NO_SPLIT && (subTuIndex < NumIntraSubPartitions - 1 !InferTuCbfLuma)))	
	tu_cbf_luma [x0][y0]	ae(v)
	if(IntraSubPartitionsSplitType != ISP_NO_SPLIT)	
[0484]	InferTuCbfLuma = InferTuCbfLuma && !tu_cbf_luma[x0][y0]	
	}	
	if((treeType == SINGLE_TREE treeType == DUAL_TREE_CHROMA)	
	{	
	if((IntraSubPartitionsSplitType == ISP_NO_SPLIT && !(cu_sbt_flag &&	
	((subTuIndex == 0 && cu_sbt_pos_flag) (subTuIndex == 1 && !cu_sbt_pos_flag)))	
	(IntraSubPartitionsSplitType != ISP_NO_SPLIT && (subTuIndex == NumIntraSubPartitions - 1))) {	

[0485]

tu_cbf_cb [x0][y0]	ac(v)
tu_cbf_cr [x0][y0]	ac(v)
}	
}	
if(IntraSubPartitionsSplitType != ISP_NO_SPLIT && treeType == SINGLE_TREE && subTuIndex == NumIntraSubPartitions - 1))	
xC = CbPosX[x0][y0]	
yC = CbPosY[x0][y0]	
wC = CbWidth[x0][y0] / 2	
hC = CbHeight[x0][y0] / 2	
} else	
xC = x0	
yC = y0	
wC = tbWidth / SubWidthC	
hC = tbHeight / SubHeightC	
}	
if((tu_cbf_luma[x0][y0] tu_cbf_cb[x0][y0] tu_cbf_cr[x0][y0]) && treeType != DUAL_TREE_CHROMA) {	
if(cu_qp_delta_enabled_flag && !IsCuQpDeltaCoded) {	
cu_qp_delta_abs	ac(v)
if(cu_qp_delta_abs)	
cu_qp_delta_sign_flag	ac(v)
}	
}	
if(tu_cbf_luma[x0][y0] && treeType != DUAL_TREE_CHROMA && (tbWidth <= 32) && (tbHeight <= 32) && (IntraSubPartitionsSplit[x0][y0] == ISP_NO_SPLIT) && (!cu_sbt_flag)) {	
if(transform_skip_enabled_flag && tbWidth <= MaxTsSize && tbHeight <= MaxTsSize)	
transform_skip_flag [x0][y0]	ac(v)

	<pre> if(((CuPredMode[x0][y0] == MODE_INTER && sps_explicit_mts_inter_enabled_flag) (CuPredMode[x0][y0] == MODE_INTRA && sps_explicit_mts_intra_enabled_flag)) && (tbWidth <= 32) && (tbHeight <= 32) && (!transform_skip_flag[x0][y0])) tu_mts_idx[x0][y0] </pre>	ac(v)
[0486]	<pre> } if(tu_cbf_luma[x0][y0]) residual_coding(x0, y0, Log2(tbWidth), Log2(tbHeight), 0) if(tu_cbf_cb[x0][y0]) residual_coding(xC, yC, Log2(wC), Log2(hC), 1) if(tu_cbf_cr[x0][y0]) residual_coding(xC, yC, Log2(wC), Log2(hC), 2) } </pre>	

[0487] 2.13帧内块复制

[0488] 在HEVC屏幕内容编解码扩展 (HEVC-SCC) 和当前VVC测试模型 (VTM-4.0) 中已经采用帧内块复制 (IBC), 又名当前的图片参考。IBC将运动补偿的概念从帧间编解码扩展到帧内编解码。如图18所示 (图18是帧内块复制的图示), 当应用IBC时, 由同一图片中的参考块预测当前块。必须在对当前块进行编解码或解码之前, 已经重构参考块中的样本。尽管IBC对于大多数摄像机捕获的序列来说效率不高, 但它显示了屏幕内容的显著编解码增益。原因是屏幕内容图片中有很多重复图案, 如图标和文本字符。IBC可以有效地消除这些重复图案之间的冗余。HEVC-SCC中, 帧间编解码单元 (CU) 如果选择当前图片作为其参考图片, 则可以应用IBC。在这种情况下, MV被重命名为块矢量 (BV), 并且BV始终具有整数像素精度。为了与主配置文件HEVC兼容, 在解码图片缓冲区 (DPB) 中将当前图片标记为“长期”参考图片。应当注意, 类似地, 在多视图/3D视频编解码标准中, 视图间参考图片也被标记为“长期”参考图片。

[0489] 随BV找到其参考块之后, 可以通过复制参考块来生成预测。可以从原始信令中减去参考像素来得到残差。然后可以像在其他编解码模式中一样应用变换和量化。

[0490] 然而, 当参考块在图片的外部, 或与当前块重叠, 或在重构区域的外部, 或在受某些约束限制的有效区域的外部时, 不定义部分像素值或全部像素值。基本上, 有两种解决方案可以解决这样的问题。一种是禁止这种情况, 例如在比特流一致性方面。另一种是对那些未定义的像素值应用填充。下面的子节详细描述了解决方案。

[0491] 2.13.1 VVC测试模型中的IBC

[0492] 当前的VVC测试模型中, 例如VTM-4.0设计, 整个参考块应为当前编解码树单元 (CTU), 且不与当前块重叠。因此, 不需要填充参考块或预测块。IBC标志被编解码为当前CU的预测模式。因此, 每个CU共有三种预测模式, 即MODE_INTRA、MODE_INTER和MODE_IBC。

[0493] 2.13.1.1 IBC Merge模式

[0494] IBC merge模式下,从比特流中解析指向IBC merge候选列表中条目的索引。IBC merge列表的构建可以按照以下步骤的序列进行总结:

[0495] 第1步:推导空域候选

[0496] 第2步:插入HMVP候选

[0497] 第3步:插入成对平均候选

[0498] 空域merge候选的推导中,如图2所描绘的,在位于 A_1 、 B_1 、 B_0 、 A_0 和 B_2 所描绘的位置的候选之中选择最多四个merge候选。推导的顺序为 A_1 、 B_1 、 B_0 、 A_0 和 B_2 。只有当位置 A_1 、 B_1 、 B_0 、 A_0 的任何PU不可用(例如,因为它属于另一个条带或片)或未使用IBC模式编解码时,才考虑位置 B_2 。在添加位置 A_1 处的候选之后,对剩余候选的插入进行冗余检查,这确保将具有相同运动信息的候选从列表中排除,从而提高编解码效率。

[0499] 在插入空域候选后,如果IBC merge列表尺寸仍然小于最大IBC merge列表尺寸,则可以插入HMVP表中的IBC候选。在插入HMVP候选时执行冗余检查。

[0500] 最后,将成对平均的候选插入IBC merge列表中。

[0501] 当由merge候选标识的参考块在图片的外部、或与当前块重叠、或在重构区域的外部、或在受某些约束限制的有效区域的外部时,merge候选被称为无效merge候选。

[0502] 注意到,无效的merge候选可能会被插入IBC merge列表中。

[0503] 2.13.1.2 IBC AMVP模式

[0504] IBC AMVP模式中,从比特流解析指向IBC AMVP列表中的条目的AMVP索引点。IBC AMVP列表的构建可以按照以下步骤的序列进行总结:

[0505] 步骤1:空域候选的推导

[0506] -检查 A_0 、 A_1 ,直到找到可用的候选。

[0507] -检查 B_0 、 B_1 、 B_2 ,直到找到可用的候选。

[0508] 步骤2:插入HMVP候选

[0509] 步骤3:插入零候选。

[0510] 插入空域候选后,如果IBC AMVP列表尺寸仍然小于最大IBC AMVP列表尺寸,则可以插入HMVP表中的IBC候选。

[0511] 最后,在IBC AMVP列表中插入零候选。

[0512] 2.13.1.3色度IBC模式

[0513] 当前VVC中,在子块级别执行色度IBC模式中的运动补偿。色度块将被分割为几个子块。每个子块确定对应的亮度块是否具有块矢量和块矢量存在时的有效性。当前VTM中存在编码器约束,其中如果当前色度CU中的所有子块都具有有效的亮度块矢量,则将测试色度IBC模式。例如,YUV420视频上,色度块是 $N \times M$,然后并置亮度区域是 $2N \times 2M$ 。色度块的子块尺寸是 2×2 。有几个步骤来执行色度mv推导,然后执行块复制过程。

[0514] 1) 首先将色度块分割为 $(N \gg 1) * (M \gg 1)$ 个子块。

[0515] 2) 每个具有坐标在 (x, y) 处的左上样本的子块获取对应的亮度块,其覆盖坐标在 $(2x, 2y)$ 处的同一左上样本。

[0516] 3) 编码器检查获取的亮度块的块矢量(bv)。如果满足下列条件之一,则认为bv无效。

[0517] a. 对应的亮度块的bv不存在。

- [0518] b. 由bv标识的预测块尚未被重构。
- [0519] c. 由bv标识的预测块与当前块部分或完全重叠。
- [0520] 4) 将子块的色度运动矢量设置为对应亮度子块的运动矢量。
- [0521] 当所有子块找到有效的bv时,编码器允许IBC模式。
- [0522] 2.14 HEVC屏幕内容编解码扩展(HEVC-SCC)中的调色板模式
- [0523] 调色板模式背后的基本思想是, CU中的样本由代表性的色彩值的小集合表示。这个集合称为调色板。还可以通过信令通知逸出符号和随后的(可能量化的)分量值来指示在调色板外部的样本。这图示在图19中。
- [0524] HEVC-SCC的调色板模式中,使用预测方法对调色板和索引映射进行编解码。
- [0525] 2.14.1调色板索引的编解码
- [0526] 调色板索引使用水平和垂直遍历扫描进行编解码,如图20所示。扫描顺序在比特流中使用palette_transpose_flag来显式地信令通知。对于该子节的其余部分,假设扫描是水平的。
- [0527] 调色板索引使用两个主要调色板样本模式进行编解码:“INDEX”和“COPY_ABOVE”。如前所述,逸出符号也作为“INDEX”模式而被信令通知,并且被分配等于最大调色板尺寸的索引。除顶行或在前一个模式为“COPY_ABOVE”时,使用标志来信令通知该模式。在“COPY_ABOVE”模式下,将复制上面行中样本的调色板索引。在“INDEX”模式下,调色板索引被显式地信令通知。对于“INDEX”和“COPY_ABOVE”模式二者,都会信令通知运行值,该运行值指定也使用相同模式编解码的后续样本的数目。当逸出符号是“INDEX”或“COPY_ABOVE”模式中运行的一部分时,为每个逸出符号信令通知逸出分量值。调色板索引的编解码如图21所示。
- [0528] 按照如下方式完成该语法顺序。首先,信令通知CU的索引值的数目。这之后是使用截断的二进制编解码来信令通知整个CU的实际索引值。索引数和索引值二者以旁路模式进行编解码。这会将与索引相关的旁路bin组合在一起。然后以交织的方式来信令通知调色板采样模式(如果需要)和运行。最后,将与整个CU的逸出样本对应的分量逸出值组合在一起,并以旁路模式进行编解码。
- [0529] 信令通知索引值后,信令通知附加的语法元素last_run_type_flag。该语法元素与索引数结合,消除对信令通知与块中最后一次运行相对应的运行值的需求。
- [0530] HEVC-SCC中,还为4:2:2、4:2:0和单色色度格式启用调色板模式。调色板条目和调色板索引的信令对于所有的色度格式几乎是相同的。在非单色格式的情况下,每个调色板条目由3个分量构成。对于单色格式,每个调色板条目由单个分量构成。对于子采样色度方向,色度样本与可被2整除的亮度样本索引相关联。在重构CU的调色板索引后,如果样本只有一个与之相关联的分量,则仅使用调色板条目的第一分量。信令通知中的唯一区别是逸出分量值。对于每个逸出样本,根据与该样本相关联的分量数,信令通知的逸出分量值的数目可能不同。
- [0531] 2.15双树的调色板模式
- [0532] 当启用双树时,与亮度块的信令通知分开地信令通知是否为色度块使用调色板模式。
- [0533] 2. 现有实现方式的缺点
- [0534] 目前的VVC设计具有以下问题:

[0535] (1) 色度残差的联合编解码方法具有这样的问题:即使当色度分量不可用时,仍应用了启用/禁用该方法的信令通知和该方法中使用的色度增量量化参数(QP)的信令通知。

[0536] (2) 当启用双树时,即使序列是4:0:0,IBC和调色板模式的当前设计也需要信令通知色度编解码块的IBC/调色板模式。

[0537] (3) RST的方法具有这样的问题:即使在色度分量不可用的情况下,仍应用了启用/禁用该方法的信令通知和变换矩阵索引的信令通知。

[0538] (4) 无论ChromaArrayType的值是多少,都会信令通知slice_chroma_remain_scale_flag和用于表示色度分量的每个PCM样本值的比特数(例如,PCM_sample_bit_depth_chroma_minus1)。

[0539] (5) 对于每个CTU,对于在其上启用ALF滤波的每个色度分量,信令通知替代色度滤波器索引以选择色度分量的给定滤波器。这种信令方法不检查色度分量的可用性。

[0540] (6) 无论ChromaArrayType的值是多少,都会完成图片参数集(PPS)中pps_cb_qp_offset、pps_cr_qp_offset的信令通知。

[0541] (7) 两个色彩分量总是共享同一个RST矩阵。同时,可能一起应用RST和联合色度残差编解码,这是次优的。

[0542] (8) RST中的非零系数计数阈值遵循以下规则:

[0543] a. 如果是分开的树编解码,对于亮度分量,当存在至少1个非零系数时,可能信令通知亮度分量的RST索引;

[0544] b. 如果是分开的树编解码,对于两个色度分量,一旦当两个色度块(例如Cb和Cr)中存在至少1个非零系数时,就可能信令通知两个色度分量的RST索引;

[0545] c. 如果是单树编解码,一旦当在三个块(例如Y、Cb和Cr)中存在至少2个非零系数时,就可能信令通知整个编解码单元(包括三个色彩分量)的RST索引;

[0546] 可能需要与色彩格式以及是否启用分开的平面编解码保持一致。

[0547] (9) SBT标志的条件信令通知应与色彩格式和是否启用分开的平面编解码保持一致。

[0548] (10) 对于帧间编解码块,可能会发生cu_cbf(或cu_coded_flag)等于1,但是所有三个分量的编解码块标志(tu_cbf_luma/tu_y_coded_flag,tu_cbf_cb/tu_cb_coded_flag,tu_cbf_cr/tu_cr_coded/flag)全为零。语法元素cu_cbf/cu_coded_flag、tu_cbf_luma/tu_y_coded_flag、tu_cbf_cb/tu_cb_coded_flag和tu_cbf_cr/tu_cr_coded_flag之间可能存在冗余。

[0549] 4. 示例技术和实施例

[0550] 以下描述的详细实施例应被视为示例以解释一般概念。这些实施例不应以狭义的方式来解释。此外,这些实施例可以以任何方式组合。

[0551] 下文描述的方法还可以适用于其他编解码工具,其需要信令通知色度色彩分量(又名从属色彩分量,例如RGB色彩格式的B和R色彩分量)的某些语法元素。

[0552] 1. 可以有条件地信令通知与联合色度残差编解码相关的语法元素。

[0553] a. 可以有条件地信令通知与联合色度残差编解码中使用的色度增量QP/色度QP偏移(例如,pps_joint_cbr_qp_offset)的信令通知相关的语法元素。

[0554] b. 可以有条件地信令通知与联合色度残差编解码(例如,tu_cb_cr_joint_

residual) 的使用的指示相关的语法元素。

[0555] c. 可以信令通知语法元素, 该语法元素可以处于序列/视图/图片/条带/片组级别, 以指示是否启用联合色度残差编解码。

[0556] i. 在一个示例中, 可以在序列参数集 (SPS) / 视频参数集 (VPS) / 图片参数集 (PPS) / 条带标头/片组标头中信令通知一个标志。

[0557] d. 对于以上示例, 可以在色度格式不等于4:0:0和/或禁用分开的平面编解码的条件下信令通知与联合色度残差编解码相关的语法元素。

[0558] i. 替代地, 可以在ChromaArrayType不等于诸如0的特定值的条件下信令通知与联合色度残差编解码相关的语法元素。

[0559] ii. 替代地, 可以在ChromaArrayType和/或一个或多个语法元素 (例如, 序列级别标志) 的条件下信令通知与联合色度残差编解码相关的语法元素。

[0560] e. 替代地, 对于以上示例, 在色度格式等于4:0:0和/或启用分开的平面编解码的条件下, 可以跳过与联合色度残差编解码相关的语法元素的信令通知。

[0561] f. 替代地, 对于以上示例, 在ChromaArrayType等于诸如0的特定值的情况下, 可以跳过与联合色度残差编解码相关的语法元素的信令通知。

[0562] g. 以上方法还可以适用于联合色度残差编解码方法的其他变型。

[0563] 2. 根据色彩格式, 可以跳过二次变换的使用的指示的信令通知。

[0564] a. 在一个示例中, 当色度格式不等于4:0:0和/或启用分开的平面编解码时, 可以跳过二次变换的使用的指示 (例如, st_idx) 的信令通知。

[0565] b. 在一个示例中, 当ChromaArrayType等于诸如0的特定值时, 可以跳过二次变换的使用的指示 (例如, st_idx) 的信令通知。

[0566] c. 可以在ChromaArrayType不等于诸如0的特定值的条件下信令通知二次变换的使用的指示 (例如st_idx)。

[0567] d. 可以在色度格式不等于4:0:0和/或禁用分开的平面编解码的条件下信令通知二次变换的使用的指示 (例如, st_idx)。

[0568] 3. 可以有条件地信令通知与图片级别的色度增量QP/色度QP偏移 (例如, pps_cb_qp_offset和/或pps_cr_qp_offset) 的信令通知相关的语法元素。

[0569] a. 在一个示例中, 当色度格式不等于4:0:0和/或启用分开的平面编解码时, 可以跳过图片级别的色度增量QP (例如, pps_cb_qp_offset和/或pps_cr_qp_offset) 的信令通知。

[0570] b. 在一个示例中, 当ChromaArrayType等于诸如0的特定值时, 可以跳过图片级别的色度增量QP (例如pps_cb_qp_offset和/或pps_cr_qp_offset) 的信令通知。

[0571] c. 在ChromaArrayType不等于诸如0的特定值的条件下, 可以信令通知图片级别的色度增量QP (例如pps_cb_qp_offset和/或pps_cr_qp_offset)。

[0572] d. 在色度格式不等于4:0:0和/或禁用分开的平面编解码的条件下, 可以信令通知图片级别的色度增量QP (例如pps_cb_qp_offset和/或pps_cr_qp_offset) 的信令通知。

[0573] 4. 根据色彩格式, 可以跳过使用依赖于亮度的色度残差缩放 (LDCRS) 的指示的信令通知。

[0574] a. 在一个示例中, 当色度格式不等于4:0:0和/或启用分开的平面编解码时, 可以

跳过LDCRS的使用的指示(例如,slice_chroma_residual_scale_flag)的信令通知。

[0575] b. 在一个示例中,当ChromaArrayType等于诸如0的特定值时,可以跳过LDCRS的使用的指示(例如,slice_chroma_residual_scale_flag)的信令通知。

[0576] c. 在ChromaArrayType不等于诸如0的特定值的条件下,可以信令通知LDCRS的使用的指示(例如slice_chroma_residual_scale_flag)。

[0577] d. 在色度格式不等于4:0:0和/或禁用分开的平面编解码的条件下,可以信令通知LDCRS的使用的指示(例如,slice_chroma_residual_scale_flag)。

[0578] 5. 根据色彩格式,可以跳过色度阵列的样本的比特深度和/或色度量化参数范围偏移的值的指示的信令通知。

[0579] a. 在一个示例中,当色度格式等于4:0:0和/或启用分开的平面编解码时,可以跳过色度阵列的样本的比特深度和/或色度量化参数范围偏移的值的指示(例如,bit_depth_chroma_minus8)的信令通知。

[0580] b. 在一个示例中,当ChromaArrayType等于诸如0的特定值时,可以跳过色度阵列的样本的比特深度和/或色度量化参数范围偏移值的指示(例如bit_depth_chroma_minus8)的信令通知。

[0581] c. 在一个示例中,在ChromaArrayType不等于诸如0的特定值的条件下或在(色度格式不等于4:0:0和/或禁用分开的平面编解码的条件)条件下,可以信令通知色度阵列的样本的比特深度和/或色度量化参数范围偏移的值的指示(例如,bit_depth_chroma_minus8)。

[0582] 6. 根据色彩格式,可以跳过用于表示色度分量的每个PCM样本值的比特数的指示的信令通知。

[0583] a. 在一个示例中,当色度格式不等于4:0:0和/或启用分开的平面编解码时,可以跳过用于表示色度分量(例如,pcm_sample_bit_depth_chroma_minus1)的每个PCM样本值的比特数的指示的信令通知。

[0584] b. 在一个示例中,当ChromaArrayType等于诸如0的特定值时,可以跳过用于表示色度分量的每个PCM样本值的比特数的指示(例如pcm_sample_bit_depth_chroma_minus1)的信令通知。

[0585] c. 在ChromaArrayType不等于诸如0的特定值的条件下,可以信令通知用于表示色度分量的每个PCM样本值的比特数的指示(例如pcm_sample_bit_depth_chroma_minus1)。

[0586] d. 在色度格式不等于4:0:0和/或禁用分开的平面编解码的条件下,可以信令通知用于表示色度分量的每个PCM样本值的比特数的指示(例如pcm_sample_bit_depth_chroma_minus1)。

[0587] 7. 可以有条件地信令通知与色度分量的ALF滤波器相关的语法元素(例如,指示是否需要发送滤波器系数的标志、滤波器系数、APS索引)。

[0588] a. 在一个示例中,条件是ChromaArrayType是否不等于诸如0的特定值(例如,指示色彩格式和/或是否存在任何色度分量)。

[0589] b. 根据等于4:0:0的色彩格式和/或启用分开的平面编解码,可以跳过发送附加的ALF色度滤波器系数的指示的信令通知(例如alf_chroma_filter_signal_flag,slice_alf_chroma_idc,slice_alf_aps_id_chroma)。

- [0590] c. 根据色彩格式等于4:0:0和/或启用分开的平面编解码,可以跳过ALF色度滤波器系数(例如,限幅标记/限幅参数/滤波器系数)的信令通知。
- [0591] d. 根据某些条件,可以为色度编解码块禁用ALF色度滤波器的时域预测。
- [0592] e. 替代地,相关的语法元素可以仍然被信令通知,但是当条件为真时被推断为默认值,诸如alf_chroma_filter_signal_flag被推断为0。
- [0593] 8. 可以有条件地信令通知与色度分量的调色板模式/帧内块复制(IBC)模式相关的语法元素(例如,标志/模式索引,用于指示是否为色度块/条带/图片/片组/片/砖启用调色板/IBC)。
- [0594] a. 在一个示例中,条件是ChromaArrayType是否不等于诸如0的特定值。
- [0595] b. 在一个示例中,可以有条件地信令通知pred_mode_ibc_flag。
- [0596] c. 根据色彩格式等于4:0:0和/或启用分开的平面编解码,可以跳过用于色度分量的调色板模式指示的信令通知。
- [0597] d. 替代地,相关的语法元素可以仍然被信令通知,但是当条件为真时被推断为默认值,例如推断出色度分量的调色板模式被禁用。
- [0598] 9. 可以有条件地信令通知与色度分量的ALF滤波器时域预测相关的语法元素(例如,滤波器索引/ALF APS索引,开/关控制标志)。
- [0599] a. 在一个示例中,条件是ChromaArrayType是否不等于诸如0的特定值。
- [0600] b. 根据色彩格式等于4:0:0和/或启用分开的平面编解码,可以跳过二次变换的使用的指示的信令通知。
- [0601] c. 根据色彩格式等于4:0:0和/或启用分开的平面编解码,可以跳过与色度分量相关的语法元素的ALF滤波器时域预测的信令通知。
- [0602] d. 根据某些条件,可以针对色度编解码块禁用ALF色度滤波器的时域预测。
- [0603] 10. 根据ChromaArrayType有条件地信令通知的SPS中的那些语法元素可以与ChromaArrayType的仅一个条件检查一起被信令通知。
- [0604] a. 在一个示例中,当ChromaArrayType不等于诸如0的特定值时,还可以按一定的顺序信令通知以下部分或全部语法元素。
- [0605] i. qtbt_dual_tree_intra_flag
- [0606] ii. sps_cclm_enabled_flag
- [0607] iii. pcm_sample_bit_depth_chroma_minus1
- [0608] iv. bit_depth_chroma_minus8
- [0609] b. 在一个示例中,ChromaArrayType的条件检查可以替换为色彩格式等于4:0:0和/或启用分开的平面编解码的条件检查。
- [0610] 11. 对于上述语法元素,是否调用色度块的解码过程可取决于色彩格式和/或分开的平面编解码的使用。
- [0611] a. 对于上述语法元素,是否调用色度块的解码过程可以取决于ChromaArrayType。
- [0612] b. 在一个示例中,当ChromaArrayType等于诸如0的特定值、或色彩格式是4:0:0或启用分开的平面编解码时,ALF/二次变换/色度残差缩放过程/量化缩放矩阵不应用于色度块。
- [0613] c. 在一个示例中,当ChromaArrayType不等于诸如0的特定值时,ALF/二次变换/色

度残差缩放过程/量化缩放矩阵可以应用于色度块。

[0614] 12. 对于上述语法元素,即使在ChromaArrayType等于特定值(例如0)或色彩格式是4:0:0或启用分开的平面编解码时,它们仍然可以被信令通知。然而,解码过程和解码器应忽略解码值。

[0615] a. 在一个示例中,当上述条件之一为真时,在解码过程中不使用上文提及的语法元素,例如与联合残差色度编解码相关的那些语法元素(例如pps_joint_cbr_qp_offset),并且解码器应忽略它们的值。

[0616] b. 在一个示例中,当上述条件之一为真时,在解码过程中不使用上文提及的语法元素,例如与色度调色板模式/色度IBC模式/色度ALF相关的语法元素,并且解码器应忽略它们的值。

[0617] 13. 对于上述语法元素,可以应用编码器约束。

[0618] a. 一致性比特流应满足:当色度格式是4:0:0和/或启用分开的平面编解码时应禁用联合色度残差编解码。

[0619] b. 一致性比特流应满足:当色度格式是4:0:0和/或启用分开的平面编解码时应禁用色度ALF。

[0620] c. 一致性比特流应满足:当色度格式是4:0:0和/或启用分开的平面编解码时应禁用色度二次变换(又名,NSST/RST)(例如,sps_st_enabled_flag)。

[0621] d. 一致性比特流应满足:当色度格式是4:0:0和/或启用分开的平面编解码时,色度二次变换矩阵(又名,NSST/RST)的索引(例如st_idx)应等于诸如0的特定值。

[0622] e. 一致性比特流应满足:当色度格式是4:0:0和/或启用分开的平面编解码时,用于联合色度残差编解码的色度增量QP应等于诸如0的特定值。

[0623] f. 一致性比特流应满足:当色度格式是4:0:0和/或启用分开的平面编解码时,亮度相关色度残差缩放(例如,slice_chroma_residual_scale_flag)的使用应等于诸如0的特定值。

[0624] g. 一致性比特流应满足:当色度格式是4:0:0和/或启用分开的平面编解码时,图片级别中信令通知的色度增量QP应等于诸如0的特定值。

[0625] h. 当ChromaArrayType等于诸如0的特定值时,上述语法元素(例如,slice_chroma_residual_scale_flag)仍然可以被解码,但不用在解码过程中,并且解码器应忽略它们的值。

[0626] 14. 色度色彩分量可以使用不同的RST矩阵。

[0627] a. 在一个示例中,可以针对单个色度分量分别信令通知RST矩阵的单独指示。

[0628] i. 在一个示例中,第二色度分量的RST矩阵的指示的信令通知可以取决于第一色度分量的RST矩阵的指示。

[0629] b. 在一个示例中,可以信令通知用于第一色度分量的RST矩阵;

[0630] 而可以在解码器侧推导出第二色度分量的矩阵。

[0631] i. 在一个示例中,用于第二色度分量的矩阵可以从用于一个代表性亮度块的RST矩阵中推导出。

[0632] ii. 在一个示例中,用于第二色度分量的矩阵可以从用于第一色度分量的RST矩阵中推导出,两个矩阵可以是不相等的。

- [0633] iii. 在一个示例中,用于其他色度分量的矩阵可以从色度帧内模式推导出。
- [0634] c. 对于多于一个的色彩分量,需要信令通知多少索引可以取决于色彩子采样格式。
- [0635] d. 对于多于一个的色彩分量,需要信令通知多少索引可以取决于是否启用了双树。
- [0636] e. 对于多于一个的色彩分量,需要信令通知多少索引可以取决于是否使用联合色度残差编解码。
- [0637] i. 在一个示例中,可以信令通知一个索引以进行残差编解码。
- [0638] 15. 当将联合色度残差编解码应用于一个块时,可以跳过二次变换矩阵索引的信令通知。
- [0639] a. 在一个示例中,当将联合色度残差编解码应用于一个块时,为两个色度分量的二者禁用二次变换。
- [0640] b. 在一个示例中,当将联合色度残差编解码应用于一个块时,以预定义矩阵索引来启用二次变换。
- [0641] i. 在一个示例中,当将联合色度残差编解码应用于一个块时,以相同预定义矩阵索引将二次变换应用于两个色彩分量。
- [0642] ii. 在一个示例中,当将联合色度残差编解码应用于一个块时,以不同预定义矩阵索引将二次变换应用于两个色彩分量。
- [0643] iii. 在一个示例中,当将联合色度残差编解码应用于一个块时,以预定义矩阵索引将二次变换应用于一个色彩分量,并且为另一个色彩分量禁用二次变换。
- [0644] c. 替代地,当用二次变换对一个块进行编解码时,可以跳过联合色度残差编解码的使用的信令通知。
- [0645] i. 替代地,此外,对该块禁用联合色度残差编解码。
- [0646] 16. 当联合色度剩余编解码应用于一个块时,可以跳过跨分量线性模型 (CCLM) 方法的信令通知。
- [0647] a. 在一个示例中,当联合色度残差编解码应用于一个块时,为两个色度分量中的二者禁用CCLM。
- [0648] b. 替代地,当一个块用CCLM编解码时,可以跳过联合色度残差编解码的使用的信令通知。
- [0649] i. 替代地,此外,对该块禁用联合色度残差编解码。
- [0650] c. 在一个示例中,可以同时启用联合色度残差编解码和CCLM。
- [0651] 两个色度残差之间的相关性可以由CCLM系数推导出。
- [0652] i. 例如,假设 $Cb = a1 * luma + b1$ 且 $Cr = a2 * luma + b2$,Cb和Cr的残差(分别表示为 $resiCb$ 和 $resiCr$)之间的相关性可以假设为: $resiCb = a2 * ((resiCr - b1) / a1) + b2$ 。
- [0653] ii. 替代地,Cb和Cr的残差之间的相关性可以假设为: $resiCb = a2 * ((resiCr - b1) / a1)$ 。
- [0654] iii. 替代地,Cb和Cr的残差之间的相关性可以假设为: $resiCb = a2 * resiCr / a1$ 。
- [0655] iv. 替代地,可以允许两个色度残差之间的不同相关性,其中它们的K(K>=1)可以从CCLM系数推导出。

[0656] d. 在上述讨论中, CCLM模式可以指代在VVC中利用跨分量线性模型的任何类型的模式, 诸如LM模式、LM-T模式和LM-L模式。

[0657] 17. 当联合色度残差编解码应用于一个块时, 可以禁用LDCRS。

[0658] 18. 是否信令通知cu_cbf/cu_coded_flag可以取决于色彩格式和/或分量编解码方法(诸如是否启用分开的平面编解码)。

[0659] a. 在一个示例中, 如果色彩格式是4:0:0和/或应用分开的平面编解码, 则不会信令通知cu_cbf/cu_coded_flag。

[0660] i. 如果ChromaArrayType等于诸如0的特定值, 则不会信令通知cu_cbf/cu_coded_flag。

[0661] ii. 此外, 替代地, cu_cbf/cu_coded_flag如果在色彩格式是4:0:0和/或应用了分开的平面编解码时没有被信令通知, 则被推断为1。

[0662] 19. 是否信令通知亮度块的编解码块标志(例如, tu_cbf_luma/tu_y_coded_flag)可以取决于色彩格式和/或分量编解码方法。

[0663] a. 在一个示例中, 如果色彩格式是4:0:0和/或应用分开的平面编解码, 则不信令通知亮度块的编解码块标志(例如, tu_cbf_luma/tu_y_coded_flag)。

[0664] b. 在一个示例中, 如果当前块以非跳过帧间模式编解码并且色彩格式是4:0:0, 则不信令通知亮度块的编解码块标志(例如, tu_cbf_luma/tu_y_coded_flag)。

[0665] c. 在一个示例中, 如果当前块以非跳过帧间模式编解码并且应用分开的平面编解码, 则不信令通知亮度块的编解码块标志(例如, tu_cbf_luma/tu_y_coded_flag)。

[0666] d. 在一个示例中, 如果当前块以非跳过帧间模式编解码并且ChromaArrayType等于诸如0的特定值, 则不信令通知亮度块的编解码块标志(例如, tu_cbf_luma/tu_y_coded_flag)。

[0667] 20. 是否信令通知tu_cbf_cb/tu_cb_coded_flag和tu_cbf_cr/tu_cr_coded_flag取决于色彩格式和/或分量编解码方法(例如是否启用分开的平面编解码)。

[0668] a. 在一个示例中, 如果色彩格式是4:0:0和/或应用分开的平面编解码, 则不信令通知tu_cbf_cb/tu_cb_coded_flag和tu_cbf_cr/tu_cr_coded_flag。

[0669] b. 在一个示例中, 如果ChromaArrayType等于特定值(例如0), 则不信令通知tu_cbf_cb/tu_cb_coded_flag和tu_cbf_cr/tu_cr_coded_flag。

[0670] 21. 是否信令通知色彩分量的编解码块标记可以取决于其他色彩分量的编解码块标记。

[0671] a. 是否信令通知tu_cbf_cr/tu_cr_coded_flag可以取决于tu_cbf_luma/tu_y_coded_flag和/或tu_cbf_cb/tu_cb_coded_flag。

[0672] i. 在一个示例中, 如果tu_cbf_luma/tu_y_coded_flag和tu_cbf_cb/tu_cb_coded_flag都等于0, 则不信令通知tu_cbf_cr/tu_cr_coded_flag且将其推断为1。

[0673] b. 是否信令通知tu_cbf_cb/tu_cb_coded_flag可以取决于tu_cbf_luma/tu_y_coded_flag和/或tu_cbf_cr/tu_cr_coded_flag。

[0674] i. 在一个示例中, 如果tu_cbf_luma/tu_y_coded_flag和tu_cbf_cr/tu_cr_coded_flag都等于0, 则不信令通知tu_cbf_cb/tu_cb_coded_flag且将其推断为1。

[0675] c. 是否信令通知tu_cbf_luma/tu_y_coded_flag可以取决于tu_cbf_cb/tu_cb_

coded_flag和/或tu_cbf_cr/tu_cr_coded_flag。

[0676] i. 在一个示例中,如果tu_cbf_cb/tu_cb_coded_flag和tu_cbf_cr/tu_cr_coded_flag都等于0,则不信令通知tu_cbf_luma/tu_y_coded_flag且将其推断为1。

[0677] d. 用于三个色彩分量的编解码块标志的编解码顺序可以与用于亮度、Cb、Cr的编解码标志不同。

[0678] e. 仅当当前CU只有一个TU并且cu_cbf等于1时,才可以应用上述方法。

[0679] i. 替代地,此外,仅当当前CU具有多个TU但是仅一个TU具有非零系数(例如, SBT模式)并且cu_cbf等于1时,才可以应用上述方法。

[0680] 22. 根据相关联的色彩分量的非零系数(诸如,第2.9.1节中的numSigCoeff)的数目是否大于阈值并且该阈值可以取决于色彩格式和/或分量编解码方法(诸如是否启用分开的平面编解码),可以信令通知是否启用或/和如何将RST应用于块/编解码单元(例如, st_idx)。

[0681] a. 在一个示例中,单树编解码结构下的阈值可以取决于色彩格式是否是4:0:0和/或是否应用分开的平面编解码。

[0682] i. 例如,如果色彩格式是4:0:0和/或应用分开的平面编解码,则阈值为1;否则,阈值为2。

[0683] b. 在一个示例中,单树编解码结构下的阈值可以取决于ChromaArrayType是否等于诸如0的特定值。

[0684] i. 例如,如果ChromaArrayType等于0,则阈值为1;否则,阈值为2。

[0685] 23. 对于编解码单元中的RST侧信息(例如, st_idx)的信令通知,它可以仅取决于亮度分量的非零系数的数目,而不是所有三个色彩分量。

[0686] a. 在一个示例中,如果亮度块的非零系数的数目大于阈值(例如1),则可以信令通知RST侧信息。

[0687] b. 在一个示例中,当亮度块具有所有零系数(例如, tu_cbf_luma/tu_y_coded_flag等于0)时,不需要信令通知RST侧信息,并且推断RST被禁用。

[0688] c. 在一个示例中,如果三个色彩分量的编解码块标志(例如, tu_cbf_luma/tu_y_coded_flag、tu_cbf_cb/tu_cb_coded_flag、和tu_cbf_cr/tu_cr_coded_flag)中的一个等于0,则不需要信令通知RST侧信息,并且推断RST被禁用。

[0689] d. 当应用单树编解码结构时,可以应用上述方法。

[0690] i. 替代地,此外,当应用单树编解码结构且色彩格式不是4:0:0时,可以应用上述方法。

[0691] ii. 替代地,此外,当应用单树编解码结构,色彩格式不是4:0:0且禁用分开的平面编解码时,可以应用上述方法。

[0692] 24. 对于色度分量(诸如在单独的树编解码结构中)的RST侧信息(例如, st_idx)的信令通知,它可以取决于两个色度分量的编解码块标志。

[0693] a. 在一个示例中,如果编解码块标志(例如tu_cbf_cb/tu_cb_coded_flag或tu_cbf_cr/tu_cr_coded_flag)中的一个等于0,则可以跳过RST侧信息的信令通知。

[0694] 25. 是否和/或如何信令通知块的SBT的指示可以取决于色彩格式和/或分量编解码方法(诸如是否启用分开的平面编解码)和/或tu_cbf_luma/tu_y_coded_flag和/或tu_

cbf_cb/tu_cb_coded_flag和/或tu_cbf_cr/tu_cr_coded_flag。

[0695] a. 在一个示例中,如果tu_cbf_luma/tu_y_coded_flag、tu_cbf_cb/tu_cb_coded_flag和tu_cbf_cr/tu_cr_coded_flag都等于0,则不信令通知SBT的指示。

[0696] b. 在一个示例中,如果tu_cbf_luma/tu_y_coded_flag等于0,则不信令通知SBT的指示。

[0697] c. 在一个示例中,如果tu_cbf_cb/tu_cb_coded_flag和tu_cbf_cr/tu_cr_coded_flag都等于0,则不信令通知SBT的指示。

[0698] d. 在一个示例中,仅当色彩格式是4:0:0和/或应用分开的平面编解码时,才可以应用上述项目符号中的有条件的信令通知。

[0699] 26. 用于TU的编解码块标志的信令通知(例如,tu_cbf_luma/tu_y_coded_flag和/或tu_cbf_cb/tu_cb_coded_flag和/或tu_cbf_cr/tu_cr_coded_flag)可以取决于包括TU的CU的SBT的使用。

[0700] a. 在一个示例中,如果应用了SBT并且TU没有被SBT清零,则不信令通知该TU的tu_cbf_luma/tu_y_coded_flag且将其推断为1。

[0701] b. 在一个示例中,如果应用了SBT并且TU未被SBT清零,则不信令通知该TU的tu_cbf_cb/tu_cb_coded_flag和/或tu_cbf_cr/tu_cr_coded_flag且将其推断为1。

[0702] c. 一致性比特流应满足:当为编解码单元启用SBT时,亮度块应具有至少一个非零系数(例如,tu_cbf_luma/tu_y_coded_flag等于1)。

[0703] i. 替代地,一致性比特流应满足:当为编解码单元启用SBT时,三个色彩分量之一应具有至少一个非零系数(例如,tu_cbf_luma/tu_y_coded_flag,tu_cbf_cb/tu_cb_coded_flag,tu_cbf_cr/tu_cr_coded_flag中的至少一个等于1)。

[0704] 27. 一致性比特流应满足:当编解码单元的cu_cbf(例如,对于非跳过帧间编解码块)等于1时,亮度块应具有至少一个非零系数(例如,tu_cbf_luma/tu_y_coded_flag等于1)。

[0705] a. 替代地,一致性比特流应满足:当编解码单元的cu_cbf(例如,对于非跳过帧间编解码块)等于1时,三个色彩分量应具有至少一个非零系数(例如,tu_cbf_luma/tu_y_coded_flag,tu_cbf_cb/tu_cb_coded_flag,tu_cbf_cr/tu_cr_coded_flag中的至少一个等于1)。

[0706] 28. 无论ChromaArrayType的值或色彩格式还是分开的平面编解码,都可以仍然信令通知上述语法元素,但是,在解码过程中,可以忽略那些语法元素的解码值。即,在检查那些语法元素的解码值之前,解码过程可以首先检查ChromaArrayType的值或色彩格式或分开的平面编解码。

[0707] 5. 附加的实施例

[0708] 5.1 ALF解码过程的实施例1

[0709] 本实施例示出了调用自适应环路滤波器过程的示例,其中考虑了色度色彩格式/分开的平面编解码标志(由ChromaArrayType表示)的条件。

[0710] 如下建议的更改用下划线强调:

[0711] 8.8环路滤波器过程

[0712] 8.8.1概述

[0713] 如以下顺序步骤所指定的,应用三个环路滤波器,即去块滤波器、样本自适应偏移和自适应环路滤波器:

[0714] 对于去块滤波器,适用以下:

[0715] -调用第8.8.2.1节中指定的去块滤波过程,其中用重构图片样本阵列 S_L ,并且当ChromaArrayType不等于0时,阵列 S_{Cb} 和 S_{Cr} 作为输入,并且用修改的重构图片样本阵列 S'_L ,并且当ChromaArrayType不等于0时,去块后的阵列 S'_{Cb} 和 S'_{Cr} 作为输出。

[0716] -阵列 S'_L ,并且当ChromaArrayType不等于0时,将阵列 S'_{Cb} 和 S'_{Cr} 分配给阵列 S_L ,并且当ChromaArrayType不等于0时,分别为阵列 S_{Cb} 和 S_{Cr} (代表解码图片)。

[0717] 当sps_sao_enabled_flag等于1时,适用以下:

[0718] 调用第8.8.3.1节中指定的样本自适应偏移过程,其中用重构图片样本阵列 S_L 并且当ChromaArrayType不等于0时,将阵列 S_{Cb} 和 S_{Cr} 作为输入,以及使用修改的重构图片样本阵列 S'_L ,当ChromaArrayType不等于0时,样本自适应偏移后的阵列 S'_{Cb} 和 S'_{Cr} 作为输出。

[0719] 阵列 S'_L ,并且当ChromaArrayType不等于0时,将阵列 S'_{Cb} 和 S'_{Cr} 分配给阵列 S_L ,并且当ChromaArrayType不等于0时,分别为阵列 S_{Cb} 和 S_{Cr} (代表解码图片)。

[0720] 当sps_alf_enabled_flag等于1时,适用以下:

[0721] 调用第8.8.4.1节中指定的自适应环路滤波器过程,其中用重构图片样本阵列 S_L ,当ChromaArrayType不等于0,阵列 S_{Cb} 和 S_{Cr} 作为输入,以及用修改的重构图片样本阵列 S'_L ,并且当ChromaArrayType不等于0时,自适应环路滤波后的阵列 S'_{Cb} 和 S'_{Cr} 作为输出,。

[0722] 阵列 S'_L ,并且当ChromaArrayType不等于0时,阵列 S'_{Cb} 和 S'_{Cr} 被分配给阵列 S_L ,并且当ChromaArrayType不等于0时,分别为阵列 S_{Cb} 和 S_{Cr} (代表解码图片)。

[0723] 8.8.4自适应环路滤波器过程

[0724] 8.8.4.1概述

[0725] 该过程的输入是在自适应环路滤波器recPicture $_L$ 之前的重构的图片样本阵列,并且当ChromaArrayType不等于0时,是阵列recPicture $_{Cb}$ 和recPicture $_{Cr}$ 。

[0726] 该过程的输出是自适应环路滤波器alfPicture $_L$ 之后的修改后的重构图片样本阵列,并且当ChromaArrayType不等于0时,是阵列alfPictureCb和alfPictureCr。

[0727] 分别将自适应环路滤波器之后的修改的重构图片样本阵列alfPicture $_L$ 、alfPicture $_{Cb}$ 和alfPicture $_{Cr}$ 中的样本值初始地设置为等于自适应环路滤波器之前的重构图片样本阵列recPicture $_L$ 、recPicture $_{Cb}$ 和recPicture $_{Cr}$ 中的样本值。

[0728] 当slice_alf_enabled_flag等于1时,对于具有亮度编解码树块位置(rx,ry)的每个编解码树单元,其中 $rx=0..PicWidthInCtbs-1$ 和 $ry=0..PicHeightInCtbs-1$,以下适用:

[0729] 当alf_ctb_flag[0][rx][ry]等于1时,调用第8.8.4.2节中指定的亮度样本的编解码树块滤波过程,其中将recPicture $_L$ 、alfPicture $_L$ 和亮度编解码树块位置(xCtb,yCtb)设置为等于 $(rx \ll CtbLog2SizeY, ry \ll CtbLog2SizeY)$ 作为输入,并且输出是修改的已滤波图片alfPicture $_L$ 。

[0730] 当ChromaArrayType不等于0且alf_ctb_flag[1][rx][ry]等于1时,调用第8.8.4.4节中指定的色度样本的编解码树块过滤过程,其中将recPicture设置为等于recPicture $_{Cb}$,alfPicture设置为等于alfPicture $_{Cb}$,并将色度编解码树块位置(xCtbC,

yCtbC) 设置为等于 $(rx \ll (\text{CtbLog2SizeY}-1), ry \ll (\text{CtbLog2SizeY}-1))$ 作为输入, 并且输出是修改的滤波图片 $\text{alfPicture}_{\text{Cb}}$ 。

[0731] 当 ChromaArrayType 不等于 0 且 $\text{alf_ctb_flag}[2][rx][ry]$ 等于 1 时, 调用第 8.8.4.4 节中指定的色度样本的编解码树块滤波过程, 其中将 recPicture 设置为等于 $\text{recPicture}_{\text{Cr}}$, alfPicture 设置为等于 $\text{alfPicture}_{\text{Cr}}$, 并将色度编解码树块位置 $(x\text{CtbC}, y\text{CtbC})$ 设置为等于 $(rx \ll (\text{CtbLog2SizeY}-1), ry \ll (\text{CtbLog2SizeY}-1))$ 作为输入, 并且输出是修改的滤波图片 $\text{alfPicture}_{\text{Cr}}$ 。

[0732] 7.3.5.1 一般的条带标头语法

	slice_header() {
	...
	if(sps_sao_enabled_flag) {
	slice_sao_luma_flag
	if(ChromaArrayType != 0)
	slice_sao_chroma_flag
	}
[0733]	if(sps_alf_enabled_flag) {
	slice_alf_enabled_flag
	if(slice_alf_enabled_flag) {
	num_alf_aps_ids
	for(i = 0; i < num_alf_aps_ids; i++)
	slice_alf_aps_id_luma[i]
	if(ChromaArrayType != 0){
	slice_alf_chroma_idc
	if(slice_alf_chroma_idc && (slice_type != I num_alf_aps_ids != 1))
	slice_alf_aps_id_chroma
	}
	}
[0734]	}
	...
	}

[0735] $\text{slice_alf_chroma_idc}$ 等于 0 指定不将自适应环路滤波器应用于Cb和Cr色彩分量。 $\text{slice_alf_chroma_idc}$ 等于 1 指示将自适应环路滤波器应用于Cb色彩分量。 $\text{slice_alf_chroma_idc}$ 等于 2 指示将自适应环路滤波器应用于Cr色彩分量。 $\text{slice_alf_chroma_idc}$ 等于 3 指示将自适应环路滤波器应用于Cb和Cr色彩分量。当不存在 $\text{slice_alf_chroma_idc}$ 时, 推断它等于 0。当 ChromaArrayType 等于 0 时, $\text{slice_alf_chroma_idc}$ 因为不用于解码过程中而不存在。

- [0736] -截断的一元二值化 $tu(v)$ 的最大值 $maxVal$ 设置为等于3。
- [0737] - $slice_alf_aps_id_chroma$ 指定条带的色度分量所指代的 $adaptation_parameter_set_id$ 。当不存在 $slice_alf_aps_id_chroma$ 时,推断它等于 $slice_alf_aps_id_luma[0]$ 。具有 $adaptation_parameter_set_id$ 等于 $slice_alf_aps_id_chroma$ 的ALF APS NAL单元的TemporalId应当小于或等于编解码的条带NAL单元的TemporalId。
- [0738] -对于帧内条带和IRAP图片中的条带, $slice_alf_aps_id_chroma$ 不得指代与其他图片相关联的ALF APS,而不是包括帧内条带或IRAP图片的图片。
- [0739] -当 $ChromaArrayType$ 等于0时, $slice_alf_aps_id_chroma$ 因为不用于解码过程中而不存在。

[0740] 5.2. 色度增量QP的信令通知的实施例2

[0741] 添加的改变以下划线表示,并且移除的部分以[[]]标记。

[0742] 7.3.2.2. 图片参数集RBSP语法

[0743]	<pre>pic_parameter_set_rbsp() { ... cabac_init_present_flag</pre>	<p>描述符</p> <p>u(1)</p>
[0744]	<pre>for(i = 0; i < 2; i++) num_ref_idx_default_active_minus1[i] rpl1_idx_present_flag init_qp_minus26 transform_skip_enabled_flag if(transform_skip_enabled_flag) log2_transform_skip_max_size_minus2 cu_qp_delta_enabled_flag if(cu_qp_delta_enabled_flag) cu_qp_delta_subdiv <u>if(ChromaArrayType != 0) {</u> pps_cb_qp_offset pps_cr_qp_offset pps_slice_chroma_qp_offsets_present_flag } ... }</pre>	<p>ue(v)</p> <p>u(1)</p> <p>se(v)</p> <p>u(1)</p> <p>ue(v)</p> <p>u(1)</p> <p>ue(v)</p> <p>se(v)</p> <p>se(v)</p> <p>u(1)</p>

[0745] $pps_cb_qp_offset$ 和 $pps_cr_qp_offset$ 指定分别用于推导出 Qp'_{Cb} 和 Qp'_{Cr} 的亮度量化参数 Qp'_y 的偏移量。 $pps_cb_qp_offset$ 和 $pps_cr_qp_offset$ 的值应在-12至+12的范围内(含端值)。当不存在[[ChromaArrayType等于0]]时, $pps_cb_qp_offset$ 和 $pps_cr_qp_offset$ 被推断为0[[不用于在解码过程中,解码器应忽略它们的值]]。

[0746] pps_slice_chroma_qp_offsets_present_flag等于1指示slice_cb_qp_offset和slice_cr_qp_offset语法元素存在于相关联的条带标头中。pps_slice_chroma_qp_offsets_present_flag等于0表示这些语法元素不存在于相关联的条带标头中。当不存在[[ChromaArrayType等于0]]时,则pps_slice_chroma_qp_offsets_present_flag应当等于0。

[0747] 7.3.5条带标头语法

[0748] 7.3.5.1一般的条带标头语法

	slice_header() {	描述符
[0749]	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_qp_delta	se(v)
	if(ChromaArrayType != 0 && pps_slice_chroma_qp_offsets_present_flag) {	
[0750]	slice_cb_qp_offset	se(v)
	slice_cr_qp_offset	se(v)
	}	
	if(sps_sao_enabled_flag) {	
	slice_sao_luma_flag	u(1)
	if(ChromaArrayType != 0)	
	slice_sao_chroma_flag	u(1)
	}	
	...	
	byte_alignment()	
	}	

[0751] 5.3联合色度残差编解码的信令通知的实施例#3

[0752] 添加的改变以下划线表示,并且移除的部分以[[]]标记。

[0753] 7.3.2.2图片参数集RBSP语法

	pic_parameter_set_rbsp() {	描述符
	...	
	pps_cb_qp_offset	se(v)
	pps_cr_qp_offset	se(v)
[0754]	if (ChromaArrayType != 0) {	
	pps_joint_cbr_qp_offset	se(v)
	}	
	...	

[0755]	}	
--------	---	--

[0756] 7.3.4.1一般的片组标头语法

	tile_group_header() {	描述符
	...	
	if(pps_tile_group_chroma_qp_offsets_present_flag) {	
	tile_group_cb_qp_offset	se(v)
	tile_group_cr_qp_offset	se(v)
[0757]	if (ChromaArrayType != 0) {	
	tile_group_joint_cbr_qp_offset	se(v)
	}	
	}	
	...	
	}	

[0758] 7.3.6.12残差编解码语法

	residual_coding(x0, y0, log2TbWidth, log2TbHeight, cIdx) {	描述符
	if(ChromaArrayType != 0 && cIdx == 2 && tu_cbf_cb[x0][y0]) {	
	tu_joint_cbr_residual[x0][y0]	ae(v)
[0759]	if(tu_joint_cbr_residual [x0][y0]) {	
	return	
	}	
	...	
	}	

[0760] 7.4.3.2图片参数集RBSP语义

[0761] pps_joint_cbr_qp_offset指定用于推导出 Qp'_{CbCr} 的亮度量化参数 Qp'_Y 的偏移。 pps_joint_cbr_qp_offset的值应在-12至+12的范围内(含端值)。如果不存在[[ChromaArrayType等于0]]，则pps_joint_cbr_qp_offset被推断为0[[不用于解码过程

中,解码器应忽略其值]]。

[0762] 5.4LMCS上的实施例#4

[0763] 是否启用依赖于亮度的色度残差缩放(例如,slice_chroma_residual_scale_flag)取决于ChromaArrayType。添加的改变以下划线表示,并且移除的部分以[[]]标记。

[0764] 7.3.5条带标头语法

[0765] 7.3.5.1一般的条带标头语法

	描述符
slice_header() {	
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)
...	
if(pps_slice_chroma_qp_offsets_present_flag) {	
slice_cb_qp_offset	se(v)
slice_cr_qp_offset	se(v)
}	
[0766] if(sps_sao_enabled_flag) {	
slice_sao_luma_flag	u(1)
if(ChromaArrayType != 0)	
slice_sao_chroma_flag	u(1)
}	
if(sps_alf_enabled_flag) {	
slice_alf_enabled_flag	u(1)
if(slice_alf_enabled_flag) {	
num_alf_aps_ids_minus1	ue(v)
for(i = 0; i <= num_alf_aps_ids_minus1; i++)	
slice_alf_aps_id[i]	u(5)
}	
}	
dep_quant_enabled_flag	u(1)
if(!dep_quant_enabled_flag)	

	sign_data_hiding_enabled_flag	u(1)
	if(deblocking_filter_override_enabled_flag)	
	deblocking_filter_override_flag	u(1)
	if(deblocking_filter_override_flag) {	
	slice_deblocking_filter_disabled_flag	u(1)
	if(!slice_deblocking_filter_disabled_flag) {	
	slice_beta_offset_div2	se(v)
	slice_tc_offset_div2	se(v)
	}	
	}	
	if(sps_lmcs_enabled_flag) {	
	slice_lmcs_enabled_flag	u(1)
	if(slice_lmcs_enabled_flag) {	
[0767]	slice_lmcs_aps_id	
	if(<u>ChromaArrayType</u> != 0 && !(qtbtt_dual_tree_intra_flag && slice_type == I))	
	slice_chroma_residual_scale_flag	u(1)
	}	
	if(entropy_coding_sync_enabled_flag)	
	num_entry_point_offsets	ue(v)
	if(NumEntryPoints > 0) {	
	offset_len_minus1	ue(v)
	for(i = 0; i < NumEntryPoints; i++)	
	entry_point_offset_minus1[i]	u(v)
	}	
	byte_alignment()	
	}	

[0768] slice_chroma_residual_scale_flag等于1指定为当前条带启用色度残差缩放。slice_chroma_residual_scale_flag等于0指定为当前条带不启用色度残差缩放。当不存在slice_chroma_residual_scale_flag时,推断它等于0。当ChromaArrayType等于0时, slice_chroma_residual_scale_flag因为不用于解码过程中而不存在。

[0769] 5.5分组语法的实施例#5

[0770] 添加的改变以下划线表示,并且移除的部分以[[]]标记。

[0771] 7.3.2.3序列参数集RBSP语法

[0772]

	描述符
seq_parameter_set_rbsp() {	
sps_decoding_parameter_set_id	u(4)
sps_max_sub_layers_minus1	u(3)
sps_reserved_zero_5bits	u(5)
profile_tier_level(sps_max_sub_layers_minus1)	
gra_enabled_flag	u(1)
sps_seq_parameter_set_id	ue(v)
chroma_format_idc	ue(v)
if(chroma_format_idc == 3)	
separate_colour_plane_flag	u(1)
pic_width_in_luma_samples	ue(v)
pic_height_in_luma_samples	ue(v)
conformance_window_flag	u(1)
if(conformance_window_flag) {	
conf_win_left_offset	ue(v)
conf_win_right_offset	ue(v)
conf_win_top_offset	ue(v)
conf_win_bottom_offset	ue(v)
}	
bit_depth_luma_minus8	ue(v)
[[bit_depth_chroma_minus8	ue(v)]]
log2_max_pic_order_cnt_lsb_minus4	ue(v)
sps_sub_layer_ordering_info_present_flag	u(1)
for(i = (sps_sub_layer_ordering_info_present_flag ? 0 : sps_max_sub_layers_minus1); i <= sps_max_sub_layers_minus1; i++) {	
sps_max_dec_pic_buffering_minus1[i]	ue(v)
sps_max_num_reorder_pics[i]	ue(v)
sps_max_latency_increase_plus1[i]	ue(v)
}	

[0773]

long_term_ref_pics_flag	u(1)
sps_idr_rpl_present_flag	u(1)
rpl1_same_as_rpl0_flag	u(1)
for(i = 0; i < !rpl1_same_as_rpl0_flag ? 2 : 1; i++) {	
num_ref_pic_lists_in_sps[i]	ue(v)
for(j = 0; j < num_ref_pic_lists_in_sps[i]; j++)	
ref_pic_list_struct(i, j)	
}	
[[qtbtt_dual_tree_intra_flag	u(1)]]
log2_ctu_size_minus2	ue(v)
log2_min_luma_coding_block_size_minus2	ue(v)
partition_constraints_override_enabled_flag	u(1)
sps_log2_diff_min_qt_min_cb_intra_slice_luma	ue(v)
sps_log2_diff_min_qt_min_cb_inter_slice	ue(v)
sps_max_mtt_hierarchy_depth_inter_slice	ue(v)
sps_max_mtt_hierarchy_depth_intra_slice_luma	ue(v)
if(sps_max_mtt_hierarchy_depth_intra_slice_luma != 0) {	
sps_log2_diff_max_bt_min_qt_intra_slice_luma	ue(v)
sps_log2_diff_max_tt_min_qt_intra_slice_luma	ue(v)
}	
if(sps_max_mtt_hierarchy_depth_inter_slices != 0) {	
sps_log2_diff_max_bt_min_qt_inter_slice	ue(v)
sps_log2_diff_max_tt_min_qt_inter_slice	ue(v)
}	
[[if(qtbtt_dual_tree_intra_flag) {	
sps_log2_diff_min_qt_min_cb_intra_slice_chroma	ue(v)
sps_max_mtt_hierarchy_depth_intra_slice_chroma	ue(v)
if(sps_max_mtt_hierarchy_depth_intra_slice_chroma != 0) {	
sps_log2_diff_max_bt_min_qt_intra_slice_chroma	ue(v)
sps_log2_diff_max_tt_min_qt_intra_slice_chroma	ue(v)
}	
}]]	
sps_sao_enabled_flag	u(1)
sps_alf_enabled_flag	u(1)

[0774]

sps_pcm_enabled_flag	u(1)
if(sps_pcm_enabled_flag) {	
pcm_sample_bit_depth_luma_minus1	u(4)
[[pcm_sample_bit_depth_chroma_minus1	u(4)]]
log2_min_pcm_luma_coding_block_size_minus3	ue(v)
log2_diff_max_min_pcm_luma_coding_block_size	ue(v)
pcm_loop_filter_disabled_flag	u(1)
}	
if((CtbSizeY / MinCbSizeY + 1) <= (pic_width_in_luma_samples / MinCbSizeY - 1)) {	
sps_ref_wraparound_enabled_flag	u(1)
if(sps_ref_wraparound_enabled_flag)	
sps_ref_wraparound_offset_minus1	ue(v)
}	
sps_temporal_mvp_enabled_flag	u(1)
if(sps_temporal_mvp_enabled_flag)	
sps_sbtmvp_enabled_flag	u(1)
sps_amvr_enabled_flag	u(1)
sps_bdof_enabled_flag	u(1)
sps_affine_amvr_enabled_flag	u(1)
sps_dmvr_enabled_flag	u(1)
<u>if(ChromaArrayType != 0){</u>	
<u>Bit depth chroma minus8</u>	
<u>qtbtt dual tree intra flag</u>	<u>u(1)</u>
<u>if(qtbtt dual tree intra flag) {</u>	
<u>sps log2 diff min qt min cb intra slice chroma</u>	<u>ue(v)</u>
<u>sps max mtt hierarchy depth intra slice chroma</u>	<u>ue(v)</u>
<u>if(sps max mtt hierarchy depth intra slice chroma != 0) {</u>	
<u>sps log2 diff max bt min qt intra slice chroma</u>	<u>ue(v)</u>
<u>sps log2 diff max tt min qt intra slice chroma</u>	<u>ue(v)</u>
<u> }</u>	
<u> }</u>	
<u>if(sps pcm enabled flag) {</u>	
<u> pcm sample bit depth chroma minus1</u>	<u>u(4)</u>

[0775]

_____}	
sps_cclm_enabled_flag	u(1)
if(sps_cclm_enabled_flag && chroma_format_idc == 1)	
sps_cclm_colocated_chroma_flag	u(1)
}	
sps_mts_enabled_flag	u(1)
if(sps_mts_enabled_flag) {	
sps_explicit_mts_intra_enabled_flag	u(1)
sps_explicit_mts_inter_enabled_flag	u(1)
}	
sps_sbt_enabled_flag	u(1)
if(sps_sbt_enabled_flag)	
sps_sbt_max_size_64_flag	u(1)
sps_affine_enabled_flag	u(1)
if(sps_affine_enabled_flag)	
sps_affine_type_flag	u(1)
sps_gbi_enabled_flag	u(1)
sps_ibc_enabled_flag	u(1)
sps_ciip_enabled_flag	u(1)
sps_fpel_mmvd_enabled_flag	u(1)
sps_triangle_enabled_flag	u(1)
sps_lmcs_enabled_flag	u(1)
sps_ladf_enabled_flag	u(1)
if(sps_ladf_enabled_flag) {	
sps_num_ladf_intervals_minus2	u(2)
sps_ladf_lowest_interval_qp_offset	se(v)
for(i = 0; i < sps_num_ladf_intervals_minus2 + 1; i++) {	
sps_ladf_qp_offset[i]	se(v)
sps_ladf_delta_threshold_minus1[i]	ue(v)
}	
}	
sps_extension_flag	u(1)
if(sps_extension_flag)	
while(more_rbsp_data())	

	sps_extension_data_flag	u(1)
[0776]	rbsp_trailing_bits()	
	}	

[0777] 5.6实施例#6

[0778] 添加的改变以下划线表示,并且移除的部分以[[]]标记。

[0779] 7.3.5.3自适应环路滤波器数据语法

	alf_data(adaptation_parameter_set_id) {	描述符
	alf_luma_filter_signal_flag	u(1)
	if(ChromaArrayType != 0)	
	alf_chroma_filter_signal_flag	u(1)
	if(alf_luma_filter_signal_flag) {	
	alf_luma_clip_flag	u(1)
	alf_luma_num_filters_signaled_minus1	tb(v)
	if(alf_luma_num_filters_signaled_minus1 > 0) {	
	for(filtIdx = 0; filtIdx < NumAlfFilters; filtIdx++)	
	alf_luma_coeff_delta_idx[filtIdx]	tb(v)
	}	
	alf_luma_use_fixed_filter_flag	u(1)
[0780]	if(alf_luma_use_fixed_filter_flag) {	
	alf_luma_fixed_filter_set_idx	tb(v)
	alf_luma_fixed_filter_pred_present_flag	u(1)
	if(alf_luma_fixed_filter_pred_present_flag) {	
	for(i = 0; i < NumAlfFilters; i++)	
	alf_luma_fixed_filter_pred_flag[i]	u(1)
	}	
	}	
	alf_luma_coeff_delta_flag	u(1)
	if(!alf_luma_coeff_delta_flag &&	
	alf_luma_num_filters_signaled_minus1 > 0)	
	alf_luma_coeff_delta_prediction_flag	u(1)
	alf_luma_min_eg_order_minus1	ue(v)
	for(i = 0; i < 3; i++)	
[0781]	alf_luma_eg_order_increase_flag[i]	u(1)

[0782]

if(alf_luma_coeff_delta_flag) {	
for(sIdx = 0; sIdx <= alf_luma_num_filters_signaled_minus1; sIdx++)	
alf_luma_coeff_flag[sIdx]	u(1)
}	
for(sIdx = 0; sIdx <= alf_luma_num_filters_signaled_minus1; sIdx++) {	
if(alf_luma_coeff_flag[sIdx]) {	
for(j = 0; j < 12; j++) {	
alf_luma_coeff_delta_abs[sIdx][j]	uek(v)
if(alf_luma_coeff_delta_abs[sIdx][j])	
alf_luma_coeff_delta_sign[sIdx][j]	u(1)
}	
}	
}	
if(alf_luma_clip_flag) {	
alf_luma_clip_min_eg_order_minus1	ue(v)
for(i = 0; i < 3; i++)	
alf_luma_clip_eg_order_increase_flag[i]	u(1)
for(sIdx = 0; sIdx <= alf_luma_num_filters_signaled_minus1; sIdx++) {	
if(alf_luma_coeff_flag[sIdx]) {	
for(j = 0; j < 12; j++) {	
if(filtCoeff[sIdx][j])	
alf_luma_clip_idx[sIdx][j]	uek(v)
}	
}	
}	
}	
}	
if(alf_chroma_filter_signal_flag) {	
alf_chroma_clip_flag	u(1)
alf_chroma_min_eg_order_minus1	ue(v)
for(i = 0; i < 2; i++)	
alf_chroma_eg_order_increase_flag[i]	u(1)

	for(j = 0; j < 6; j++) {	
	alf_chroma_coeff_abs[j]	uek(v)
	if(alf_chroma_coeff_abs[j] > 0)	
	alf_chroma_coeff_sign[j]	u(1)
	}	
	if(alf_chroma_clip_flag) {	
	alf_chroma_clip_min_eg_order_minus1	ue(v)
	for(i = 0; i < 2; i++)	
[0783]	alf_chroma_clip_eg_order_increase_flag[i]	u(1)
	for(j = 0; j < 6; j++) {	
	if(alf_chroma_coeff_abs[j])	
	alf_chroma_clip_idx[j]	uek(v)
	}	
	}	
	}	
	}	

[0784] alf_chroma_filter_signal_flag等于1指定信令通知色度滤波器。alf_chroma_filter_signal_flag等于0指定不信令通知色度滤波器。当ChromaArrayType等于0时,alf_chroma_filter_signal_flag应等于0。

[0785] 5.7实施例#7

[0786] 添加的改变以下划线表示,并且移除的部分以[[]]标记。

	slice_header() {	描述符
	...	
	if(sps_sao_enabled_flag) {	
	slice_sao_luma_flag	u(1)
	if(ChromaArrayType != 0)	
	slice_sao_chroma_flag	u(1)
[0787]	}	
	if(sps_alf_enabled_flag) {	
	slice_alf_enabled_flag	u(1)
	if(slice_alf_enabled_flag) {	
	num_alf_aps_ids	tb(v)
	for(i = 0; i < num_alf_aps_ids; i++)	

[0788]

slice_alf_aps_id_luma[i]	u(5)
<u>if(ChromaArrayType != 0)</u> {	
slice_alf_chroma_idc	tu(v)
if(slice_alf_chroma_idc && (slice_type != I num_alf_aps_ids != 1))	
slice_alf_aps_id_chroma	u(5)
}	
}	
}	
dep_quant_enabled_flag	u(1)
if(!dep_quant_enabled_flag)	
sign_data_hiding_enabled_flag	u(1)
if(deblocking_filter_override_enabled_flag)	
deblocking_filter_override_flag	u(1)
if(deblocking_filter_override_flag) {	
slice_deblocking_filter_disabled_flag	u(1)
if(!slice_deblocking_filter_disabled_flag) {	
slice_beta_offset_div2	se(v)
slice_tc_offset_div2	se(v)
}	
}	
if(sps_lmcs_enabled_flag) {	
slice_lmcs_enabled_flag	u(1)
if(slice_lmcs_enabled_flag) {	
slice_lmcs_aps_id	u(5)
<u>if(ChromaArrayType != 0)</u> {	
if(!(qtbt_dual_tree_intra_flag && slice_type == I))	
slice_chroma_residual_scale_flag	u(1)
}	
}	
if(entropy_coding_sync_enabled_flag)	
num_entry_point_offsets	ue(v)
if(NumEntryPoints > 0) {	
offset_len_minus1	ue(v)

	for(i = 0; i < NumEntryPoints; i++)	
	entry_point_offset_minus1[i]	u(v)
[0789]	}	
	byte_alignment()	
	}	

[0790] bit_depth_chroma_minus8指定色度阵列BitDepth_c的样本的比特深度以及色度量化参数范围偏移QpBdOffset_c的值,如下所示:

$$[0791] \quad \text{BitDepth}_c = 8 + \text{bit_depth_chroma_minus8} \quad (7-5)$$

$$[0792] \quad \text{QpBdOffset}_c = 6 * \text{bit_depth_chroma_minus8} \quad (7-6)$$

[0793] bit_depth_chroma_minus8应在0至8的范围内(含端值)。当ChromaArrayType等于0时,bit_depth_chroma_minus8不用于解码过程中,并且解码器应忽略其值。

[0794] slice_alf_chroma_idc等于0指定不将自适应环路滤波器应用于Cb和Cr色彩分量。slice_alf_chroma_idc等于1指定将自适应环路滤波器应用于Cb色彩分量。slice_alf_chroma_idc等于2指定将自适应环路滤波器应用于Cr色彩分量。slice_alf_chroma_idc等于3指定将自适应环路滤波器应用于Cb和Cr色彩分量。当slice_alf_chroma_idc不存在时,将其推断为等于0。当ChromaArrayType等于0时,slice_alf_chroma_idc因为不用于解码过程中而不存在,并且解码器应忽略其值。

[0795] 截断的一元二值化tu(v)的最大值maxVal被设置为等于3。

[0796] slice_alf_aps_id_chroma指定条带的色度分量所指代的adaptation_parameter_set_id。当不存在slice_alf_aps_id_chroma时,推断它等于slice_alf_aps_id_luma[0]。具有adaptation_parameter_set_id等于slice_alf_aps_id_chroma的ALF APS NAL单元的TemporalId应当小于或等于编解码的条带NAL单元的TemporalId。

[0797] 对于帧内条带和IRAP图片中的条带,slice_alf_aps_id_chroma不应指代与除了包括帧内条带的图片或IRAP图片以外的其他图片相关联的ALF APS。

[0798] 当ChromaArrayType等于0时,slice_alf_aps_id_chroma因为不用于解码过程中而不存在,并且解码器应忽略其值。

[0799] slice_chroma_residual_scale_flag等于1指定为当前条带启用色度残差缩放。slice_chroma_residual_scale_flag等于0指定为当前条带不启用色度残差缩放。当不存在slice_chroma_residual_scale_flag时,推断它等于0。当ChromaArrayType等于0时,slice_chroma_residual_scale_flag因为不用于解码过程中而不存在。

[0800] alf_chroma_filter_signal_flag等于1指定信令通知色度滤波器。alf_chroma_filter_signal_flag等于0指定不信令通知色度滤波器。当ChromaArrayType等于0时,alf_chroma_filter_signal_flag应等于0。

[0801] 6. 所公开技术的示例实施方式

[0802] 图16是视频处理设备1600的框图。设备1600可以用于实现本文描述的一种或多种方法。设备1600可以实施在智能手机、平板计算机、计算机、物联网(IoT)接收器等中。设备1600可以包括一个或多个处理器1602、一个或多个存储器1604和视频处理硬件1606。(一个或多个)处理器1602可以配置为实现本文档中描述的一种或多种方法。(一个或多个)存储

器1604可以用于存储用于实现本文描述的方法和技术的数据和代码。视频处理硬件1606可以用于在硬件电路中实现本文档中描述的一些技术,并且可以部分地或完全地是处理器1602(例如,图形处理器核心GPU或其他信令处理电路)的一部分。

[0803] 在本文档中,术语“视频处理”可以指代视频编码、视频解码、视频压缩或视频解压缩。例如,可以在从视频的像素表示到对应的比特流表示的转换期间应用视频压缩算法,反之亦然。如语法所定义,当前视频块的比特流表示可以例如对应于比特流内共位的位或是不同位置中扩展的比特。例如,可以根据变换的和编解码的误差残差值而且还使用标头中的比特和比特流中的其他字段来对块进行编码。这里,视频块可以是与正在执行的处理操作对应的逻辑单元,例如,编解码单元、变换单元、预测单元等。

[0804] 将会理解,所公开的方法和技术通过允许使用本文档中公开的技术,将有益于合并在诸如智能手机、膝上型计算机、台式计算机和类似设备之类的视频处理装置内的视频编码器和/或解码器实施例。

[0805] 图17是视频处理的示例方法1700的流程图。该方法1700包括:在1710,在当前视频块和当前视频块的比特流表示之间执行转换,其中该转换包括响应于检测到至少一个条件选择性地启用或禁用与以下中的一个或多个的使用相关联的语法标志中的一个或多个的信令通知:联合色度残差编解码步骤、依赖于亮度的色度残差缩放(LDCRS)步骤或自适应环路滤波器步骤。

[0806] 可以使用以下基于条款的格式来描述一些实施例。

[0807] 1.一种视频处理的方法,包括:

[0808] 在当前视频块和当前视频块的比特流表示之间执行转换,其中该转换包括响应于检测到至少一个条件,选择性地启用或禁用与以下中的一个或多个的使用相关联的语法标志中的一个或多个的信令通知:联合色度残差编解码步骤、依赖于亮度的色度残差缩放(LDCRS)步骤、二次变换步骤、量化步骤或自适应环路滤波器步骤。

[0809] 2.如条款1的方法,其中,一个或多个语法标志与以下任何一个相关:色度增量量化参数,指示是否执行联合色度残差编解码的二进制值,二次变换步骤的使用,用于表示色度分量的PCM样本值的比特数,色度二次变换矩阵的索引,指示是否执行联合色度残差编解码的二进制值,或指示是否执行双树编解码步骤的二进制值。

[0810] 3.如条款1-2中任一项所述的方法,其中,一个或多个语法标志对应于以下中的一个或多个的值:pps_joint_cbr_qp_offset,tu_cb_cr_joint_residual,st_idx,pps_cb_qp_offset,slice_chroma_residual_scale_flag,pcm_sample_bit_depth_chroma_minus1,qtbt_dual_tree_intra_flag,sps_cclm_enabled_flag或sps_st_enabled_flag。

[0811] 4.如条款1-3中任一项所述的方法,其中,一个或多个语法标志与序列、视图、图片、条带、片、片组、编解码块、变换块、预测单元或其他视频数据单元相关联。

[0812] 5.如条款1-4中任一项所述的方法,其中,至少一个条件与以下相关联:色度格式、分开的平面编解码步骤的存在,ChromaArrayType变量的值、或一个或多个语法标志中标志的值,或对当前视频块应用二次转换。

[0813] 6.如条款1-5中任一项的方法,还包括:

[0814] 基于一个或多个语法标志,确定将解码步骤应用于当前视频块。

[0815] 7.一种视频处理的方法,包括:

[0816] 在当前视频块和当前视频块的比特流表示之间执行转换,其中该转换包括响应于检测到至少一个条件,选择性地启用或禁用与以下中的一个或多个的使用相关联的一个或多个语法标志的信令通知:联合色度残差编解码步骤、依赖于亮度的色度残差缩放(LDCRS)步骤、二次变换步骤、量化步骤或自适应环路滤波器步骤,其中当前视频块的第一色度分量与第一约化的二次变换(RST)矩阵相关联,并且当前视频块的第二色度分量与第二RST矩阵相关联。

[0817] 8.如条款7所述的方法,其中,第一标志与第一RST矩阵相关联,并且第二标志与第二RST矩阵相关联,其中,第一标志和第二标志包括在一个或多个语法标志中。

[0818] 9.如条款7所述的方法,其中,以比特流表示发送第一RST矩阵,并且推导出第二RST矩阵。

[0819] 10.如条款7所述的方法,其中,第二标志至少部分地基于第一标志。

[0820] 11.如条款7所述的方法,其中,第二RST矩阵是从代表性亮度块推导出的。

[0821] 12.如条款7所述的方法,其中,第二RST矩阵是从第一RST矩阵推导出的,其中,第一RST矩阵不等于第二RST矩阵。

[0822] 13.如条款7所述的方法,其中,第二RST矩阵是从色度帧内模式推导出的。

[0823] 14.如条款8所述的方法,还包括:

[0824] 响应于确定为当前视频块启用联合色度残差编解码步骤,禁用第一RST标志和/或第二标志的信令通知。

[0825] 15.如条款7-14中任一项所述的方法,还包括:

[0826] 响应于确定为当前视频块启用联合色度残差编解码步骤,应用第一RST矩阵和/或第二矩阵。

[0827] 16.一种视频处理的方法,包括:

[0828] 在当前视频块和当前视频块的比特流表示之间执行转换,其中该转换包括响应于检测到至少一个条件,选择性地启用或禁用与以下中的一个或多个的使用相关联的一个或多个语法标志的信令通知:联合色度残差编解码步骤、依赖于亮度的色度残差缩放(LDCRS)步骤、二次变换步骤、量化步骤或自适应环路滤波器步骤,其中,当前视频块的第一色度分量与第一约化的二次变换(RST)矩阵相关联,并且当前视频块的第二色度分量与第二RST矩阵相关联;以及

[0829] 响应于确定为当前视频块启用联合色度残差编解码步骤,在当前视频块的一个或多个色度分量上禁用与CCLM的使用相关的标志的信令通知,其中,一个或多个语法标志中的标志与跨分量线性模型(CCLM)的使用相关。

[0830] 17.如条款16所述的方法,还包括:

[0831] 响应于确定为当前视频块启用联合色度残差编解码步骤,在当前视频块上禁用CCLM的使用。

[0832] 18.如条款16所述的方法,还包括:

[0833] 响应于确定将CCLM应用于当前视频块,在当前视频块上禁用联合色度残差编解码步骤。

[0834] 19.如条款16所述的方法,还包括:

[0835] 在当前视频块上启用CCLM的使用和联合色度残差编解码步骤的使用。

[0836] 20.如条款16所述的方法,其中,联合色度残差编解码步骤跨越至少两个色度残差分量,其中,两个色度残差分量之间的相关性至少部分地基于与CCLM相关的系数。

[0837] 21.如条款16-20中任一项所述的方法,其中,CLLM的使用包括LM-T模式或LM-L模式的使用。

[0838] 22.如条款1-21中任一项所述的方法,其中,所述一个或多个语法标志被单独地信令通知或组合为单个标志。

[0839] 23.如条款1-22中任一项所述的方法,还包括:

[0840] 响应于确定启用联合色度残差编解码步骤,禁用依赖于亮度的色度残差缩放(LDCRS)步骤。

[0841] 24.如条款1-23中任一项所述的方法,其中,一个或多个语法标志被单独地信令通知或组合为单个标志。

[0842] 25.一种视频处理的方法,包括:为当前视频块做出关于在当前视频块的比特流表示中选择性包括一个或多个编解码块标志的决定,以及基于该决定来执行在当前视频块和当前视频块的比特流表示之间的转换,其中,该选择性包括是基于色彩格式、当前视频块的分量编解码方法或当前视频块的编解码模式。

[0843] 26.如条款25所述的方法,其中,当色彩格式是4:0:0或分量编解码方法是分开的平面编解码方法时,不信令通知用于编解码单元的编解码块标志(cu_cbf/cu_coded_flag)。

[0844] 27.如条款25所述的方法,其中,当色彩格式是4:0:0或分量编解码方法是分开的平面编解码方法时,不信令通知当前视频块的亮度分量的编解码块标志(tu_cbf_luma/tu_y_coded_flag)。

[0845] 28.如条款25所述的方法,其中,当色彩格式是4:0:0且当前视频块以非跳过帧间模式编解码时,不信令通知当前视频块的亮度分量的编解码块标志(tu_cbf_luma/tu_y_coded_flag)。

[0846] 29.如条款25所述的方法,其中,当色彩格式是4:0:0或分量编解码方法是分开的平面编解码方法时,不信令通知当前视频块的一个或多个色度分量的编解码块标志(tu_cbf_cr/tu_cr_coded_flag或tu_cbf_cb/tu_cb_coded_flag)。

[0847] 30.如条款25所述的方法,其中,信令通知当前视频块的一个或多个色度分量的编解码块标志(tu_cbf_cr/tu_cr_coded_flag或tu_cbf_cb/tu_cb_coded_flag)是基于当前视频块的亮度分量的编解码块标志。(tu_cbf_luma/tu_y_coded_flag)。

[0848] 31.一种视频处理的方法,包括:基于当前视频块的一个或多个色彩分量中的非零系数的数目,为当前视频块,做出关于将约化的二次变换(RST)选择性地应用于当前视频块的决定,以及基于该决定在当前视频块和当前视频块的比特流表示之间执行转换。

[0849] 32.如条款31所述的方法,其中,做出决定还包括将非零系数的数目与取决于当前视频块的色彩格式或分量编解码方法的阈值进行比较。

[0850] 33.如条款32所述的方法,其中阈值是1,其中色彩格式是4:0:0或分量编解码方法是分开的平面编解码方法。

[0851] 34.如条款32所述的方法,其中阈值是2,其中色彩格式不同于4:0:0,并且分量编解码方法不同于分开的平面编解码方法。

[0852] 35.如条款31所述的方法,其中,选择性应用仅基于当前视频块的亮度分量中的非零系数的数目。

[0853] 36.一种视频处理的方法,包括:在当前视频块和当前视频块的比特流表示之间执行转换,其中,比特流表示是基于根据色度的特性选择性包括指示关于色度编解码的信息的语法元素。

[0854] 37.如条款36所述的方法,其中,关于色度编解码的信息包括用于色度阵列样本的比特深度信息和/或色度量参数范围偏移的值,并且其中,色度的特性包括色度格式。

[0855] 38.如条款36所述的方法,其中,关于色度编解码的信息包括关于在转换期间使用的自适应环路滤波的信息,且其中,色度的特性对应于ChromaArrayType语法元素是否等于特定值。

[0856] 39.如条款36所述的方法,其中,关于色度编解码的信息包括关于在转换期间使用的自适应环路滤波的信息,且其中,色度的特性对应于色度格式是否是4:0:0。

[0857] 40.如条款36所述的方法,其中,关于色度编解码的信息包括关于色度的调色板模式或帧内块复制模式的信息,并且其中,色度的特性对应于ChromaArrayType语法元素是否等于特定值。

[0858] 在前一节的项目5、7、8和12中提供了条款36至40的其他示例和实施例。

[0859] 41.一种视频系统中的设备,包括处理器和其上具有指令的非暂时性存储器,其中,所述指令在由处理器执行时使处理器实现条款1至40中任一项的方法。

[0860] 42.一种计算机程序产品,存储在非暂时性计算机可读介质上,该计算机程序产品包括用于执行条款1至40中任一项的方法的程序代码。

[0861] 图22是示出其中可实现本文中所公开的各种技术的示例性视频处理系统2200的框图。各种实施方式可以包括系统2200的一些或所有组件。系统2200可以包括用于接收视频内容的输入2202。视频内容可以以原始或未压缩的格式(例如8或10位多分量像素值)被接收,或者可以为压缩或编码的格式。输入2202可以代表网络接口、外围总线接口或存储接口。网络接口的示例包括有线接口(诸如以太网、无源光网络(PON)等)和无线接口(诸如Wi-Fi或蜂窝接口)。

[0862] 系统2200可以包括可实现本文档中所描述的各种编解码或编码方法的编解码组件2204。编解码组件2204可以减少从输入2202到编解码组件2204的输出的视频的平均比特率,以产生视频的编解码表示。因此,编解码技术有时称为视频压缩或视频转码技术。如组件2206所表示的,编解码组件2204的输出可以被存储或经由所连接的通信来发送。在输入2202处接收的视频的存储或通信的比特流(或编解码)表示可以由组件2208使用,用于生成向显示接口2210发送的像素值或可显示视频。从比特流表示中生成用户可视的视频的过程有时称为视频解压缩。此外,虽然某些视频处理操作被称为“编解码”操作或工具,但是应当理解,编码器处使用编解码工具或操作,并且解码器将执行反演编解码结果的对应解码工具或操作。

[0863] 外围总线接口或显示接口的示例可以包括通用串行总线(USB)或高清多媒体接口(HDMI)或Displayport等。存储接口的示例包括SATA(串行高级技术附件)、PCI、IDE接口等。本文档中所描述的技术可以实施在各种电子装置中,诸如移动电话、膝上型计算机、智能电话或其他能够执行数字数据处理和/或视频显示的装置。

[0864] 图23是根据本技术的视频解码的方法2300的流程图表示。该方法2300包括,操作2310处,基于规则确定在视频的编解码表示的语法结构中是否包括适用于视频的区域的一个或多个语法标志。该一个或多个语法标志指示用于在编解码表示中表示区域的联合色度残差编解码步骤。该方法2300还包括,操作2320处,根据编解码表示中一个或多个语法标志的存在或不存在,通过解析编解码表示来生成区域的一个或多个解码视频块。

[0865] 图24是根据本技术的视频编码的方法2400的流程图表示。该方法2400包括,操作2410处,基于规则对视频的编解码表示的语法结构中适用于视频的区域的一个或多个语法标志进行条件编码。该一个或多个语法标志指示用于在编解码表示中表示区域的联合色度残差编解码步骤。

[0866] 在一些实施例中,基于视频的区域的颜色格式来指示规则。在一些实施例中,确定颜色格式包括确定颜色分量是否被单独编解码。在一些实施例中,在颜色分量被单独编解码的情况下,编解码表示中不存在一个或多个语法标志。在一些实施例中,在颜色分量不被单独编解码的情况下,编解码表示中存在一个或多个语法标志。

[0867] 在一些实施例中,规则指示,在颜色格式是4:0:0的情况下,编解码表示中不存在一个或多个语法标志。在一些实施例中,规则指示,在颜色格式不是4:0:0的情况下,编解码表示中存在一个或多个语法标志。在一些实施例中,规则指示,颜色格式由编解码表示中的变量ChromaArrayType指示。在一些实施例中,规则指示,在变量ChromaArrayType等于预定值的情况下,编解码表示中不存在一个或多个语法标志。在一些实施例中,规则指示,在变量ChromaArrayType不等于预定值的情况下,编解码表示中存在一个或多个语法标志。在一些实施例中,预定值是0。

[0868] 在一些实施例中,区域包括序列。在一些实施例中,在序列参数集中信令通知一个标志。在一些实施例中,区域包括图片。在一些实施例中,在图片参数集中信令通知多个标志,多个标志与量化参数偏移相关。在一些实施例中,该区域包括变换单元。在一些实施例中,与变换单元相关联的一个标志与联合色度残余编解码步骤的使用相关。在一些实施例中,区域包括条带。在一些实施例中,区域包括视图或片组。

[0869] 图25是根据本技术的视频解码的方法2500的流程图表示。方法2500包括,操作2510处,确定视频的区域的颜色格式。方法2500包括,操作2520处,基于颜色格式确定适用于视频的区域的一个或多个语法标志是否包括在视频的编解码表示的语法结构中。该一个或多个语法标志表示使用量化参数偏移,用于在编解码表示中表示区域。方法2500还包括,操作2530处,通过根据一个或多个语法标志的存在或不存在来解析编解码表示,生成视频区域的一个或多个解码视频块。

[0870] 图26是根据本技术的视频解码的方法2600的流程图表示。方法2600包括,操作2610处,确定视频的区域的颜色格式。方法2600包括,操作2620处,基于颜色格式,确定视频的编解码表示的语法结构中不存在一个或多个语法标志,该一个或多个语法标志指示适用于视频区域的二次变换的使用。在去量化步骤和逆一次变换之间应用二次变换。方法2600还包括,操作2630处,通过根据不存在一个或多个语法标志来解析编解码表示,生成视频区域的一个或多个解码的视频块。

[0871] 在一些实施例中,基于颜色格式的确定来生成一个或多个解码的视频块。在一些实施例中,方法包括,在编解码表示中存在一个或多个语法标志的情况下,丢弃用于生成一

个或多个解码视频块的一个或多个语法标志。

[0872] 图27是根据本技术的视频编码的方法2700的流程图表示。方法2700包括,操作2710处,确定与视频的区域相关联的色度格式。方法2700还包括,操作2720处,基于色度格式对视频的编解码表示的语法结构中的一个或多个语法标志进行条件编码。该一个或多个语法标志指示,用于编解码表示中表示区域的量化参数偏移的使用是启用的。

[0873] 图28是根据本技术的视频编码的方法2800的流程图表示。方法2800包括,操作2810处,确定视频的区域的颜色格式。方法2800还包括,操作2820处,基于颜色格式,通过对区域进行编码而在视频的编解码表示的语法结构中不包括指示二次变换的使用的一个或多个语法标志的情况下,生成视频的编解码表示。在正向一次变换和量化步骤之间应用二次变换。

[0874] 在一些实施例中,确定颜色格式包括确定颜色格式的色彩分量是否被单独编解码。在一些实施例中,区域包括图片。

[0875] 在一些实施例中,在(1)颜色格式是4:0:0或(2)颜色格式的色彩分量被单独编解码的情况下,编解码表示中不存在一个或多个语法标志。在一些实施例中,在(1)颜色格式不是4:0:0或(2)颜色格式的色彩分量不被单独编解码的情况下,编解码表示中存在一个或多个语法标志。

[0876] 在一些实施例中,颜色格式由比特流表示中的变量ChromaArrayType指示。在一些实施例中,在变量ChromaArrayType等于预定值的情况下,编解码表示中不存在一个或多个语法标志。在一些实施例中,在变量ChromaArrayType不等于预定值的情况下,编解码表示中存在一个或多个语法标志。在一些实施例中,预定值是0。

[0877] 在一些实施例中,对应于区域的级别包括序列级别。在一些实施例中,颜色格式的确定包括确定变量ChromaArrayType的单个步骤。在一些实施例中,在变量ChromaArrayType不等于预定值的情况下,在编解码表示中对一个或多个语法标志进行排序。在一些实施例中,预定义值是0。在一些实施例中,一个或多个语法标志包括(1)qtbttdual_tree_intra_flag、(2)sps_cclm_enabled_flag或(3)pcm_sample_bit_depth_chroma_minus1中的至少一个。

[0878] 在一些实施例中,在(1)变量ChromaArrayType等于预定值,(2)颜色格式是4:0:0,或(3)颜色格式的色彩分量被单独编解码的情况下,而没有对视频的块应用自适应环路滤波器、二次变换、色度残差缩放步骤或量化缩放矩阵,生成一个或多个解码视频块。在一些实施例中,生成一个或多个解码视频块包括在(1)变量ChromaArrayType不等于预定值的情况下,对视频的块应用自适应环路滤波器、二次变换、色度残差缩放步骤或量化缩放矩阵。

[0879] 在一些实施例中,在颜色格式是4:0:0或颜色格式的色彩分量被单独编解码的情况下,在编解码表示中将联合色度残差编解码步骤指示为禁用。在一些实施例中,在颜色格式是4:0:0或颜色格式的色彩分量被单独编解码的情况下,在编解码表示中将色度自适应环路滤波过程指示为禁用。在一些实施例中,在颜色格式是4:0:0或颜色格式的色彩分量被单独编解码的情况下,在编解码表示中将色度二次变换指示为禁用。在一些实施例中,在颜色格式是4:0:0或颜色格式的色彩分量被单独编解码的情况下,在编解码表示中将色度二次变换矩阵的索引设置为预定值。在一些实施例中,在颜色格式是4:0:0或颜色格式的色彩分量被单独编解码的情况下,在编解码表示中将联合色度残差编解码步骤中使用的色度量

化参数偏移设置为预定值。在一些实施例中,在色度格式是4:0:0或色度格式的色彩分量被单独编解码的情况下,在编解码表示中将亮度相关的色度残差缩放的使用设置为预定值。在一些实施例中,在色度格式是4:0:0或色度格式的色彩分量被单独编解码的情况下,在编解码表示中将图片级别的色度量化参数偏移设置为预定值。在一些实施例中,预定值为0。

[0880] 图29A是根据本技术的视频编码的方法2900的流程图表示。方法2900包括,操作2910处,基于包括区域的视频的色度格式,确定在视频的编解码表示的语法结构中是否缺少适用于视频区域的一个或多个语法标志。该一个或多个语法标志指示使用依赖于亮度的色度残差缩放编解码步骤,用于在编解码表示中表示区域。依赖于亮度的色度残差缩放编解码步骤包括乘法过程,该乘法过程用视频的色度信令补偿视频的亮度信令。方法2900包括,操作2920处,通过根据确定解析编解码表示来生成视频区域的一个或多个解码视频块。

[0881] 图29B是根据本技术的视频编码的方法2950的流程图表示。方法2950包括,操作2960处,确定包括区域的视频的色度格式。方法2950包括,操作2970处,基于色度格式,通过对区域进行编码而在视频的编解码表示的语法结构中不包括一个或多个语法标志的情况下,生成视频的编解码表示,其中该一个或多个语法标志指示使用了依赖于亮度的色度残差缩放编解码步骤,用于在编解码表示中表示区域,其中依赖于亮度的色度残差缩放编解码步骤包括乘法过程,该乘法过程用视频的色度信令补偿视频的亮度信令。

[0882] 图30A是根据本技术的视频编码的方法3000的流程图表示。方法3000包括,操作3010处,基于包括区域的视频的色度格式,确定在视频的编解码表示的语法结构中不存在适用于视频的区域的一个或多个语法标志。该一个或多个语法标志指示用于表示色度分量的每个脉冲编解码调制样本值的比特数。方法3000包括,操作3020处,通过根据确定解析编解码表示来生成视频区域的一个或多个解码视频块。

[0883] 图30B是根据本技术的视频编码的方法3050的流程图表示。方法3050包括,操作3060处,确定包括区域的视频的色度格式。方法3050还包括,操作3070处,基于色度格式,通过对区域进行编码而在视频的编解码表示的语法结构中不包括一个或多个语法标志的情况下,生成视频的编解码表示,其中该一个或多个语法标志指示用于表示色度分量的每个脉冲编解码调制样本值的比特数。

[0884] 图31A是根据本技术的视频编码的方法3100的流程图表示。方法3100包括,操作3110处,基于包括区域的视频的色度格式,确定在视频的编解码表示的语法结构中是否包括适用于视频的区域的一个或多个语法标志。该一个或多个语法标志与用于色度分量的自适应环路滤波器(ALF)时域预测步骤相关。方法3100还包括,操作3120处,通过根据确定解析编解码表示来生成视频区域的一个或多个解码视频块。

[0885] 图31B是根据本技术的视频编码的方法3150的流程图。方法3150包括,操作3160处,确定包括区域的视频的色度格式。方法3150还包括,操作3170处,基于色度格式,对视频的编解码表示的语法结构中的一个或多个语法标志进行条件编码,其中该一个或多个语法标志与色度分量的自适应环路滤波器(ALF)时域预测步骤相关。

[0886] 在一些实施例中,基于色度格式的确定来生成一个或多个解码的视频块。在一些实施例中,方法包括:在编解码表示中存在一个或多个语法标志的情况下,丢弃用于生成一个或多个解码视频块的一个或多个语法标志。

[0887] 在一些实施例中,确定色度格式包括确定色度格式的色彩分量是否被单独编解

码。在一些实施例中,该区域包括条带。在一些实施例中,在(1)色度格式是4:0:0,或者(2)色度格式的色彩分量被单独编解码的情况下,编解码表示中不存在一个或多个语法标志。在一些实施例中,在(1)色度格式不是4:0:0,或者(2)色度格式的色彩分量不被单独编解码的情况下,编解码表示中存在一个或多个语法标志。在一些实施例中,色度格式由比特流表示中的变量ChromaArrayType指示。在一些实施例中,在变量ChromaArrayType等于预定值的情况下,编解码表示中不存在一个或多个语法标志。在一些实施例中,在变量ChromaArrayType不等于预定值的情况下,编解码表示中存在一个或多个语法标志。在一些实施例中,预定值是0。

[0888] 在一些实施例中,一个或多个语法标志包括以下中的至少一个:(1)滤波器索引、(2)自适应环路滤波器(ALF)自适应参数集(APS)索引或(3)控制标志。在一些实施例中,对应于区域的级别包括序列级别。在一些实施例中,色度格式的确定包括确定变量ChromaArrayType的单个步骤。在一些实施例中,在变量ChromaArrayType不等于预定义值的情况下,在编解码表示中对一个或多个语法标志进行排序。在一些实施例中,预定义值是0。在一些实施例中,一个或多个语法标志包括以下中的至少一个:(1)qtbtt_dual_tree_intra_flag、(2)sps_cclm_enabled_flag或(3)pcm_sample_bit_depth_chroma_minus1。

[0889] 在一些实施例中,在(1)变量ChromaArrayType等于预定值,(2)色度格式是4:0:0,或(3)色度格式的色彩分量被单独编解码的情况下,而没有对视频的块应用自适应环路滤波器、二次变换、色度残差缩放步骤或量化缩放矩阵,生成一个或多个解码视频块。二次变换适用于正向一次变换与量化步骤之间或去量化步骤与逆一次变换之间。

[0890] 在一些实施例中,生成一个或多个解码视频块包括,在(1)ChromaArrayType不等于预定值的情况下,对视频块应用自适应环路滤波器、二次变换、色度残差缩放步骤或量化缩放矩阵。二次变换适用于正向一次变换与量化步骤之间或去量化步骤与逆一次变换之间。

[0891] 在一些实施例中,在色度格式是4:0:0或色度格式的色彩分量被单独编解码的情况下,在编解码表示中将联合色度残差编解码步骤指示为禁用。联合色度残差编解码步骤包括确定联合残差,该联合残差是与色度分量相关联的残差的平均值。

[0892] 在一些实施例中,在色度格式是4:0:0或色度格式的色彩分量被单独编解码的情况下,在编解码表示中将色度自适应环路滤波过程指示为禁用。在一些实施例中,在色度格式是4:0:0或色度格式的色彩分量被单独编解码的情况下,在编解码表示中将色度二次变换指示为禁用。色度二次变换适用于正向一次变换与量化步骤之间或去量化步骤与逆一次变换之间。

[0893] 在一些实施例中,在色度格式是4:0:0或色度格式的色彩分量被单独编解码的情况下,在编解码表示中将色度二次变换矩阵的索引设置为预定值。色度二次变换适用于正向一次变换与量化步骤之间或去量化步骤与逆一次变换之间。

[0894] 在一些实施例中,在色度格式是4:0:0或色度格式的色彩分量被单独编解码的情况下,在编解码表示中将联合色度残差编解码步骤中使用的色度量化的参数偏移设置为预定值。联合色度残差编解码步骤包括确定联合残差,该联合残差是与色度分量相关联的残差的平均值。在一些实施例中,在色度格式是4:0:0或色度格式的色彩分量被单独编解码的情况下,在编解码表示中将依赖于亮度的色度残差缩放编解码步骤的使用设置为预定值。

在一些实施例中,在色度格式是4:0:0或色度格式的色彩分量被单独编解码的情况下,在编解码表示中将图片级别的色度量化的参数偏移设置为预定值。在一些实施例中,预定值是0。

[0895] 图32是根据本技术的视频处理的方法3200的流程图表示。方法3200包括,操作3210处,对于视频的块和视频的比特流表示之间的转换,确定对应于块的多个色度色彩分量的多个约化的二次变换(RST)矩阵。二次变换适用于在正向一次变换和量化步骤之间或者在去量化步骤和逆一次变换之间。方法3200还包括,操作3220处,基于确定执行转换。

[0896] 在一些实施例中,对于比特流表示中的对应色度色彩分量,信令通知多个RST矩阵中的每一个。在一些实施例中,用于第二色度色彩分量的第二RST矩阵的信令通知是基于用于第一色度色彩分量的第一RST矩阵的信令通知。

[0897] 在一些实施例中,多个RST矩阵包括第一RST矩阵和第二RST矩阵,在比特流表示中信令通知第一RST矩阵,并且在比特流表示中不存在第二RST矩阵。在一些实施例中,基于用于亮度块的RST矩阵来推导出第二RST矩阵。在一些实施例中,基于第一RST矩阵推导出第二RST矩阵,第一RST矩阵和第二RST矩阵是不同的。在一些实施例中,基于色度帧内模式编解码步骤来推导出第二RST矩阵。

[0898] 在一些实施例中,在比特流表示中信令通知多个RST矩阵的子集。在一些实施例中,基于块的色彩子采样格式来确定多个RST矩阵的子集的数目。在一些实施例中,基于是否为块启用双树编解码步骤来确定多个RST矩阵的子集的数目。在一些实施例中,基于是否使用联合色度残差编解码步骤来确定多个RST矩阵的子集的数目。在一些实施例中,在使用联合色度残差编解码步骤的情况下,在比特流表示中信令通知单个RST矩阵。

[0899] 图33是根据本技术的视频处理的方法3300的流程图表示。方法3300包括,操作3310处,对于视频的块与视频的比特流表示之间的转换,确定在将联合色度残差编解码步骤应用于块的情况下,比特流表示中不存在用于二次变换的一个或多个矩阵索引。二次变换适用于正向一次变换与量化步骤之间或去量化步骤与逆一次变换之间的块。联合色度残差编解码步骤包括确定联合残差,该联合残差是与色度分量相关联的残差的平均值。方法3300还包括,操作3320处,基于确定执行转换。

[0900] 在一些实施例中,对块的色度分量禁用二次变换。在一些实施例中,使用一个或多个预定义矩阵索引来应用二次变换。在一些实施例中,相同的预定义矩阵索引用于块的色度分量。在一些实施例中,将不同的预定义矩阵索引应用于块的不同色度分量。在一些实施例中,预定义的矩阵索引用于块的第一色度分量,并且其中对于块的第二色度分量禁用二次变换。

[0901] 图34是根据本技术的视频处理的方法3400的流程图表示。方法3400包括,操作3410处,对于在视频的块和视频的比特流表示之间的转换,确定在以下情况下比特流表示中缺少指示联合色度残差编解码步骤的使用的语法标志:在正向一次变换和量化步骤之间或在去量化步骤和逆一次变换之间将二次变换应用于该块。联合色度残差编解码步骤包括确定联合残差,该联合残差是与色度分量相关联的残差的平均值。方法3400还包括,操作3420处,基于确定执行转换。在一些实施例中,为块禁用联合色度残差编解码步骤。

[0902] 图35是根据本技术的视频处理的方法3500的流程图表示。方法3500包括,操作3510处,对于视频的块和视频的比特流表示之间的转换,确定在以下情况下比特流表示中缺少指示跨分量线性模型编解码步骤的使用的语法标志:联合色度残差编解码步骤应用于

该块。联合色度残差编解码步骤包括确定联合残差,该联合残差是与色度分量相关联的残差的平均值。方法3500还包括,操作3520处,基于确定执行转换。在一些实施例中,为块的色度分量禁用跨分量线性模型编解码步骤。

[0903] 图36是根据本技术的视频处理的方法3600的流程图表示。方法3600包括,操作3610处,对于在视频的块和视频的比特流表示之间的转换,确定在以下情况下比特流表示中缺少指示联合色度残差编解码步骤的使用的语法标志:跨分量线性模型应用于该块。联合色度残差编解码步骤包括确定联合残差,该联合残差是与色度分量相关联的残差的平均值。方法3600还包括,操作3620处,基于确定执行转换。在一些实施例中,为该块禁用联合色度残差编解码步骤。

[0904] 图37是根据本技术的视频处理的方法3700的流程图表示。方法3700包括,操作3710处,对于视频的块与视频的比特流表示之间的转换,基于块的跨分量线性模型编解码步骤的系数,确定在联合色度残差编解码步骤中使用的两个色度残差之间的相关性。联合色度残差编解码步骤包括确定联合残差,该联合残差是与色度分量相关联的残差的平均值。方法3700包括,操作3720处,基于确定执行转换。

[0905] 在一些实施例中,第一色度分量是Cb,第二色度分量是Cr,a1、a2、b1和b2是跨分量线性模型编解码步骤的系数,并且 $Cb = a1 * luma + b1$ and $Cr = a2 * luma + b2$ 。在一些实施例中,第一色度分量的第一残差resiCb与第二色度分量的第二残差resiCr之间的相关性被表示为 $resiCb = a2 * (resiCr - b1) / a1 + b2$ 。在一些实施例中,第一色度分量的第一残差resiCb与第二色度分量的第二残差resiCr之间的相关性表示为 $resiCb = a2 * (resiCr - b1) / a1$ 。在一些实施例中,第一色度分量的第一残差resiCb与第二色度分量的第二残差resiCr之间的相关性被表示为 $resiCb = a2 * resiCr / a1$ 。

[0906] 在一些实施例中,该方法包括基于跨分量线性模型编解码步骤的系数来确定两个色度残差之间的一个或多个附加相关性。在一些实施例中,跨分量线性模型编解码步骤包括线性模型(LM)模式、LM-T模式或LM-L模式。

[0907] 图38是根据本技术的视频处理的方法3800的流程图表示。方法3800包括,操作3810处,对于视频块和视频的比特流表示之间的转换,确定在将联合色度残差编解码步骤应用于块的情况下禁用依赖于亮度的色度残差缩放编解码步骤。依赖于亮度的色度残差缩放编解码步骤包括乘法过程,该乘法过程用视频的色度信令补偿视频的亮度信令。方法3800还包括,操作3820处,基于确定执行转换。

[0908] 图39是根据本技术的视频处理的方法3900的流程图表示。方法3900包括,操作3910处,对于在视频的块和视频的编解码表示之间的转换,基于与视频的编解码特性相关的规则确定与指示至少一个非零系数的存在的块相关联的语法标志是否包括在编解码表示中的语法结构中。方法3900还包括,操作3920处,根据确定来执行转换。

[0909] 在一些实施例中,编解码特性包括视频的色彩格式。在一些实施例中,编解码特性包括使用分量编解码技术以在编解码表示中表示视频。在一些实施例中,分量编解码技术包括对视频的色彩分量进行单独编解码。

[0910] 在一些实施例中,语法结构用于编解码单元。在一些实施例中,规则指定,在色彩格式是4:0:0或视频的色彩分量被单独编解码的情况下,在语法结构中省略语法标志。在一些实施例中,色彩格式由编解码表示中的变量ChromaArrayType指示,并且其中规则指定在

ChromaArrayType等于预定义值的情况下省略语法标志。在一些实施例中,在语法结构中省略语法标志的情况下,认为编解码单元用至少一个非零系数编解码。

[0911] 在一些实施例中,将块划分为一个或多个变换单元,并且其中语法结构用于该块的变换单元。在一些实施例中,块属于视频的亮度分量。在一些实施例中,规则指定,在色彩格式是4:0:0或块的色彩分量被单独编解码的情况下,在语法结构中省略语法标志。在一些实施例中,规则指定,在色彩格式是4:0:0且使用帧间模式编解码工具对块进行编解码而不使用跳过模式编解码的情况下,在语法结构中省略语法标志。在一些实施例中,规则指定,在块的色彩分量被单独编解码并且使用帧间模式编解码工具对块进行编解码而不使用跳过模式编解码的情况下,在编解码表示中省略语法标志。在一些实施例中,色彩格式由编解码表示中的变量ChromaArrayType指示,并且其中规则指定在ChromaArrayType等于预定义值并且使用帧间模式编解码工具对块进行编解码而不使用跳过模式编解码的情况下不存在语法标志。

[0912] 在一些实施例中,块属于视频的色度分量。在一些实施例中,规则指定,在色彩格式是4:0:0或块的色彩分量被单独编解码的情况下,语法结构中不存在语法标志。在一些实施例中,色彩格式由编解码表示中的变量ChromaArrayType指示,并且规则指定在ChromaArrayType等于预定义值的情况下语法结构中不存在语法标志。在一些实施例中,预定义值是0。

[0913] 在一些实施例中,规则指定语法标志是否存在于语法结构中是基于属于视频的其他分量的其他块的一个或多个语法标志。在一些实施例中,块属于视频的Cr分量,并且一个或多个语法标志用于属于视频的Cb分量和/或亮度分量的其他块。在一些实施例中,在一个或多个语法标志指示其他块没有用至少一个非零系数编解码的情况下,在语法结构中省略语法标志,并且块被认为用至少一个非零系数编解码。在一些实施例中,块属于视频的Cb分量,并且其中一个或多个语法标志用于属于视频的Cr分量和/或亮度分量的其他块。在一些实施例中,在一个或多个语法标志指示其他块没有用至少一个非零系数编解码的情况下,在语法结构中省略语法标志,并且块被认为用至少一个非零系数编解码。在一些实施例中,块属于视频的亮度分量,并且一个或多个语法标志用于属于视频的Cr分量和/或Cb分量的其他块。在一些实施例中,在一个或多个语法标志指示其他块没有用至少一个非零系数编解码的情况下,在语法结构中省略语法标志,并且块被认为用至少一个非零系数编解码。

[0914] 在一些实施例中,其中处理块和其他块的第一顺序不同于其中将语法标志和一个或多个其他语法标志布置在编解码表示中的第二顺序。在一些实施例中,规则可适用于以下情况:块仅包括一个变换单元,并且块被指示为由编解码单元语法结构中的第二语法标志以至少一个非零系数编解码。在一些实施例中,规则可适用于以下情况:块具有多个变换单元,并且多个变换单元中仅一个具有用于子块变换的非零系数,并且该块被指示为由编解码单元语法结构中的第二语法标志用至少一个非零系数编解码。

[0915] 在一些实施例中,规则指定,在语法结构中是否包括语法标志是基于对包括变换单元的块的子块变换的使用。在一些实施例中,块属于视频的亮度分量,并且在将子块变换应用到块的一部分而不是变换单元的情况下,在语法结构中省略语法标志,并且块被认为以至少一个非零系数编解码。在一些实施例中,块属于视频的色度分量,并且在将子块变换应用到块的一部分而不是变换单元的情况下,在语法结构中省略语法标志,并且块被认为

以至少一个非零系数编解码。在一些实施例中,块属于视频的亮度分量,并且在为块启用子块变换的情况下,语法结构中存在的语法标志指示该块以至少一个非零系数编解码。在一些实施例中,块属于视频的亮度分量、Cr分量或Cb分量中的一个,在语法结构中存在的语法标志指示在为块启用子块变换情况下该块以至少一个非零系数编解码。

[0916] 在一些实施例中,块属于视频的亮度分量,并且其中,在编解码单元语法结构中的第一语法标志指示该块以至少一个非零系数编解码的情况下,变换单元语法结构中的第二语法标志指示该块以至少一个非零系数编解码。在一些实施例中,块属于视频的亮度分量、Cr分量或Cb分量之一,并且在编解码单元语法结构中的第一语法标志指示该块以至少一个非零系数编解码的情况下,变换单元语法结构中的第二语法标志指示该块以至少一个非零系数编解码。

[0917] 图40是根据本技术的视频处理的方法4000的流程图表示。方法4000包括,操作4010处,对于视频的块与视频的编解码表示之间的转换,基于与视频的编解码特性相关联的规则,确定关于具有降低维度的二次变换的信息。二次变换适用于正向一次变换和量化步骤之间或去量化步骤和逆一次变换之间,并且基于块的维度来减小降低的维度。方法4000还包括,操作4020处,根据确定执行转换。

[0918] 在一些实施例中,关于二次变换的信息包括是否为块启用二次变换。在一些实施例中,关于二次变换的信息包括将二次变换应用于块的方式。在一些实施例中,关于二次变换的信息指定适用于块的二次变换核。

[0919] 在一些实施例中,编解码特性包括视频的色彩格式。在一些实施例中,编解码特性包括使用分量编解码技术以在编解码表示中表示视频。在一些实施例中,分量编解码技术包括对视频的色彩分量进行单独编解码。

[0920] 在一些实施例中,使用单树编解码结构对块进行编解码,并且阈值是基于色彩格式是否是4:0:0或者视频的色彩分量是否被单独编解码。在一些实施例中,在色彩格式是4:0:0或色彩分量被单独编解码的情况下,阈值为1。在一些实施例中,如果色彩格式不是4:0:0并且色彩分量被一起编解码,则阈值为2。在一些实施例中,色彩格式由编解码表示中的变量ChromaArrayType指示。在一些实施例中,在ChromaArrayType等于预定义值的情况下,阈值为1。在一些实施例中,在ChromaArrayType不等于预定义值的情况下,阈值为2。在一些实施例中,预定义值是0。

[0921] 在一些实施例中,规则指定,在编解码表示中是否包括关于二次变换的信息仅是基于与属于视频的亮度分量的块相关联的非零系数的数目。在一些实施例中,在与块相关联的非零系数的数量大于阈值的情况下,在编解码表示中包括关于二次变换的信息。在一些实施例中,阈值为1。在一些实施例中,在与块相关联的非零系数的数目为0并且认为对块禁用二次变换的情况下,在编解码表示中省略关于二次变换的信息。

[0922] 在一些实施例中,规则指定,在块的语法标志指示该块没有以至少一个非零系数编解码的情况下,在编解码表示中省略关于二次变换的信息。块属于视频的Cr分量、Cb分量或亮度分量,并且认为对块禁用二次变换。在一些实施例中,规则适用于使用单树编解码结构对块进行编解码的情况。在一些实施例中,视频的色彩格式不是4:0:0。在一些实施例中,视频的色彩分量被单独编解码。在一些实施例中,规则指定,在编解码表示中是否包括关于二次变换的信息是基于块的语法标志。块属于视频的色彩分量。在一些实施例中,块属于视

频的色度分量,并且在变换单元语法中的一个语法标志指示该块未以至少一个非零系数编解码的情况下,在编解码表示中省略关于二次变换的信息。在一些实施例中,编解码特性包括视频的色彩格式,用于在编解码表示中表示视频的分量编解码技术的使用,或者指示块是否以至少一个非零系数编解码的变换单元语法结构中的语法标志。在一些实施例中,分量编解码技术包括对视频的色彩分量单独编解码。

[0923] 在一些实施例中,块属于视频的Cr分量、Cb分量或亮度分量,并且规则指定,在语法标志指示块没有以至少一个非零系数编解码并且变换单元语法结构中的其他语法标志指示属于视频的其他分量的其他块没有以至少一个非零系数编解码的情况下,在编解码表示中省略关于二次变换的信息。在一些实施例中,块属于视频的亮度分量,并且规则指定,在语法标志指示块没有以至少一个非零系数编解码的情况下,在编解码表示中省略关于二次变换的信息。在一些实施例中,块属于Cr分量或Cb分量,并且规则指定,在语法标志指示块没有以至少一个非零系数编解码并且变换单元语法结构中的第二语法标志指示属于视频的另一色度分量的第二块没有以至少一个非零系数编解码的情况下,在编解码表示中省略关于二次变换的信息。在一些实施例中,规则适用于色彩格式是4:0:0或色彩分量被单独编解码的情况。在一些实施例中,块包括与多个色彩分量相关联的样本。

[0924] 图41是根据本技术的视频处理的方法4100的流程图表示。方法4100包括,操作4110处,对于视频和视频的编解码表示之间的转换,基于条件确定与视频的色度分量的自适应环路滤波(ALF)操作相关联的语法元素是否包括在视频的编解码表示中。方法4100包括,操作4120处,根据确定执行转换。

[0925] 在一些实施例中,语法元素包括标志,该标志指示在编解码表示中是否至少包括用于自适应环路滤波操作的系数。在一些实施例中,在与ALF操作相对应的自适应参数集(APS)中信令通知语法元素。在一些实施例中,编解码表示包括与用于自适应环路滤波操作的滤波系数相关的多个语法元素。

[0926] 在一些实施例中,语法元素包括用于自适应参数集的索引、该自适应参数集包括用于一个或多个自适应环路滤波器的信息。在一些实施例中,在条带标头中信令通知语法元素。

[0927] 在一些实施例中,语法元素包括是否为任何色度分量启用ALF操作的指示。在一些实施例中,在条带标头中信令通知语法元素。

[0928] 在一些实施例中,视频的色彩格式是否存在由编解码表示中的变量ChromaArrayType指示,并且条件包括ChromaArrayType是否不等于预定义值。在一些实施例中,预定义值是0。在一些实施例中,语法元素至少包括alf_chroma_filter_signal_flag、slice_alf_chroma_idc或slice_alf_aps_id_chroma。

[0929] 在一些实施例中,语法元素至少包括限幅标志、限幅参数或滤波器系数。在一些实施例中,在满足条件的情况下,在编解码表示中省略语法元素。在一些实施例中,条件指定:不存在色度分量,视频的色彩格式是4:0:0或视频的色彩分量被单独编解码。在一些实施例中,在满足条件的情况下,在编解码表示中用编解码值来信令通知语法元素,并且无论编解码值如何,语法元素的实际值都被设置为默认值。在一些实施例中,语法元素包括alf_chroma_filter_signal_flag,并且默认值为0。

[0930] 在一些实施例中,基于条件禁用自适应环路滤波操作的时域预测。在一些实施例

中,语法元素指示自适应环路滤波操作的时域预测,并且在满足条件的情况下,在编解码表示中省略语法元素。该条件指定视频的色彩格式是4:0:0或视频的色彩分量被单独编解码。

[0931] 图42是根据本技术的视频处理的方法4200的流程图表示。方法4200包括,操作4210处,对于视频和视频的编解码表示之间的转换,基于条件确定编解码表示中是否包括与视频的色彩分量的编解码技术相关联的语法元素。方法4200包括,操作4220处,根据确定执行转换。

[0932] 在一些实施例中,编解码技术至少包括:调色板模式编解码技术,其中使用代表像素值的调色板来表示视频块的样本值;或帧内块复制编解码模式编解码技术。在一些实施例中,语法元素包括指示是否为色度分量启用编解码技术的标志或索引。在一些实施例中,语法元素包括pred_mode_ibc_flag。

[0933] 在一些实施例中,视频的色彩格式由编解码表示中的变量ChromaArrayType表示,其中条件包括ChromaArrayType是否不等于预定义值。在一些实施例中,预定义值为0。在一些实施例中,在满足条件的情况下,在编解码表示中省略与调色板模式编解码技术相关联的语法元素。该条件指定视频的色彩格式是4:0:0或视频的色彩分量被单独编解码。

[0934] 在一些实施例中,在满足条件的情况下,在编解码表示中用编解码值信令通知语法元素,并且无论编解码值如何,语法元素的实际值都被设置为默认值。在一些实施例中,语法元素包括指示调色板模式编解码技术的使用的标志,并且默认值为0,指示禁用调色板模式编解码技术。

[0935] 图43是根据本技术的视频处理的方法4300的流程图表示。方法4300包括,操作4310处,对于视频和视频的编解码表示之间的转换,基于与视频的色彩格式相关联的条件,确定指示视频的色彩分量的特性的语法元素是否包括在编解码表示中。方法4300包括,操作4320处,根据确定来执行转换。

[0936] 在一些实施例中,特性包括视频的色彩阵列的样本比特深度。在一些实施例中,特性包括色度量参数范围偏移。在一些实施例中,语法元素包括bit_depth_chroma_minus8。在一些实施例中,语法元素包括pps_joint_cbcr_qp_offset。

[0937] 在一些实施例中,在满足指定视频的色彩格式是4:0:0或视频的色彩分量被单独编解码的条件的情况下,在编解码表示中省略语法元素。在一些实施例中,视频的色彩格式由编解码表示中的变量ChromaArrayType指示,并且在满足指定ChromaArrayType等于预定义值的条件的情况下,在编解码表示中省略语法元素。在一些实施例中,在满足指定视频的色彩格式不是4:0:0或视频的色彩分量没有被单独编解码的条件的情况下,在编解码表示中包括语法元素。在一些实施例中,视频的色彩格式由编解码表示中的变量ChromaArrayType指示,并且在满足指定ChromaArrayType不等于预定义值的条件的情况下,在编解码表示中包括语法元素。在一些实施例中,预定义值为0。

[0938] 在一些实施例中,在满足条件的情况下,在编解码表示中包括语法元素,该条件指定视频的色彩格式是4:0:0,视频的色彩分量被单独编解码,或者指示视频的色彩格式的变量ChromaArrayType等于预定义值。在转换过程中忽略语法元素。在一些实施例中,与条件是否满足相关联的至少一个其他语法元素在处理或忽略该语法元素之前被处理。

[0939] 在一些实施例中,执行转换包括基于视频的块生成比特流表示。在一些实施例中,执行转换包括从比特流表示生成视频的块。

[0940] 本文档中所描述的所公开的和和其他解决方案、示例、实施例、模块和功能操作可以在数字电子电路中,或在计算机软件、固件或硬件中实现,包括本文档中公开的结构及其结构等同物,或者它们中的一个或多个的组合。所公开的和和其他实施例可以实现为一个或多个计算机程序产品,例如,编码在计算机可读介质上的一个或多个计算机程序指令模块,用于由数据处理设备执行或控制数据处理设备的操作。计算机可读介质可以是机器可读存储装置、机器可读存储基板、存储装置、实现机器可读传播信令的物质的组合,或者它们中的一个或多个的组合。术语“数据处理设备”涵盖用于处理数据的所有设备、装置和机器,例如包括可编程处理器、计算机、或多个处理器或计算机。除硬件外,设备还可以包括为所提及的计算机程序创建执行环境的代码,例如,构成处理器固件、协议栈、数据库管理系统、操作系统或它们中的中一个或多个的组的代码。传播的信令是人工生成的信令,例如,机器生成的电、光或电磁信令,其被生成以编码信息来传输到合适的接收器设备。

[0941] 计算机程序(也称为程序、软件、软件应用程序、脚本或代码)可以用任何形式的编程语言(包括编译或解释语言)来编写,并且也可以以任何形式进行部署,包括作为独立程序或作为模块、组件、子例程,或其他适合在计算环境中使用的单元进行部署。计算机程序不一定对应于文件系统中的文件。程序可以存储在保存其他程序或数据的文件的一部分(例如,存储在标记语言文档中的一个或多个脚本)、专用于所提及程序的单个文件或多个协调文件(例如,存储一个或多个模块、子程序或部分代码的文件)中。计算机程序可以部署为在一台计算机上执行,也可以部署为在位于一个站点或分布在多个站点并通过通信网络互连的多台计算机上执行。

[0942] 本文档中描述的过程和逻辑流程可以由执行一个或多个计算机程序以通过对输入数据执行操作并生成输出来执行功能的一个或多个可编程处理器来执行。过程和逻辑流程也可以由专用逻辑电路执行,并且设备也可以实现为专用逻辑电路,例如FPGA(现场可编程门阵列)或ASIC(专用集成电路)。

[0943] 例如,适合于执行计算机程序的处理器包括通用和专用微处理器,以及任何种类的数字计算机的任何一个或多个处理器。通常,处理器将从只读存储器或随机存取存储器或二者接收指令和数据。计算机的基本元件是用于执行指令的处理器和用于存储指令和数据的一个或多个存储装置。通常,计算机还将包括或可操作地耦接以从一个或多个用于存储数据的大容量存储装置(例如,磁性,磁光盘或光盘)接收数据或将数据传输到一个或多个大容量存储装置或既接收也传输。但是,计算机不必具有此类装置。适合于存储计算机程序指令和数据的计算机可读介质包括所有形式的非易失性存储器、介质和存储装置,例如包括半导体存储装置,例如EPROM、EEPROM和闪存装置;磁盘,例如内部硬盘或可移动磁盘;磁光盘;以及CD ROM和DVD-ROM磁盘。处理器和存储器可以由专用逻辑电路补充或并入专用逻辑电路中。

[0944] 尽管该专利文件包括许多细节,但是这些细节不应被解释为对任何主题或所要求保护的范围的限制,而是针对特定技术的特定实施例的特征的描述。在单独的实施例的上下文中在该专利文件中描述的某些特征也可以在单个实施例组合实现。相反,在单个实施例的上下文中描述的各种特征也可以分别在多个实施例中以任何合适的子组合来实现。而且,尽管以上可以将特征描述为以某些组合起作用并且甚至最初是这样声称的,但是在某些情况下可以从该组合中切除所要求保护的组合中的一个或多个特征,并且所要求

保护的组合可以针对子组合或子组合的变型。

[0945] 类似地,虽然在附图中以特定顺序描绘了操作,但是这不应理解为要求以所示的特定顺序或以连续的顺序执行这样的操作,或者执行所有图示的操作以实现期望的结果。此外,在该专利文件中描述的实施例中的各种系统组件的分离不应被理解为在所有实施例中都需要这种分离。

[0946] 仅描述了一些实施方式和示例,并且基于该专利文件中描述和示出的内容可以产生其他实施方式、增强和变型。

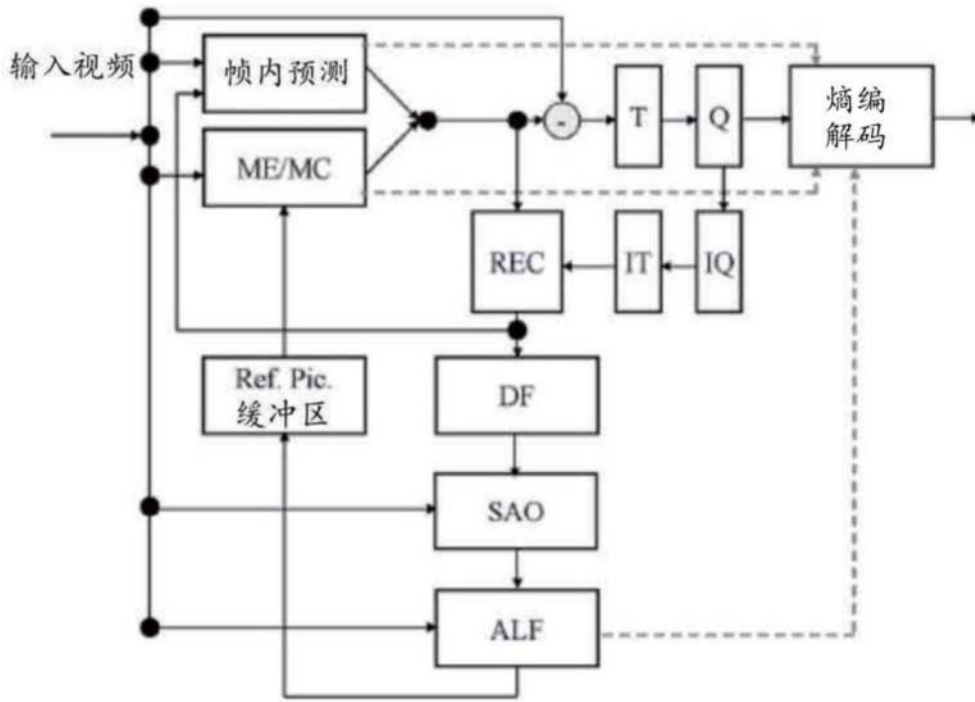


图1

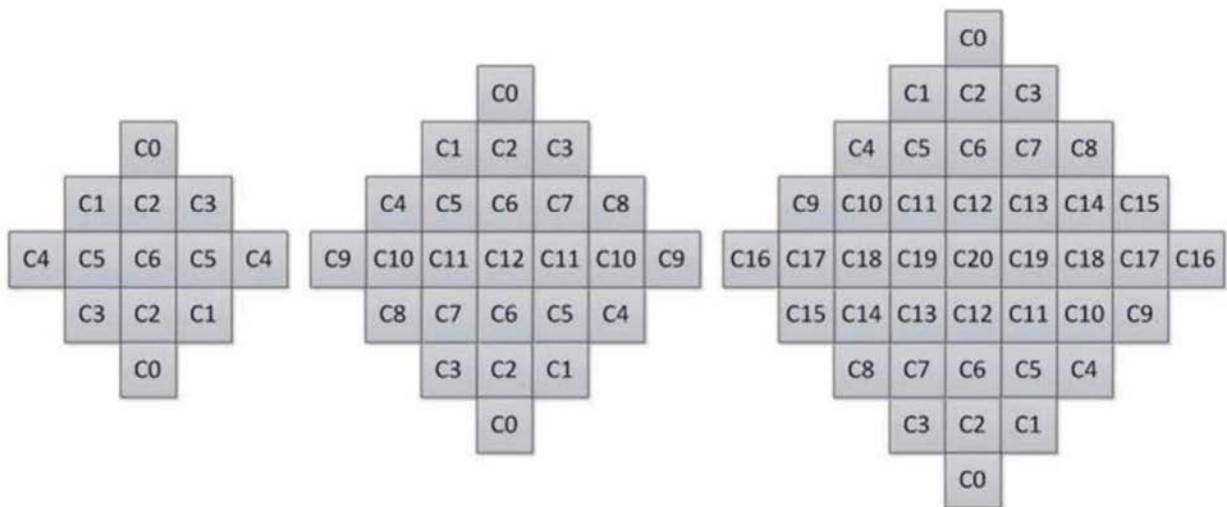
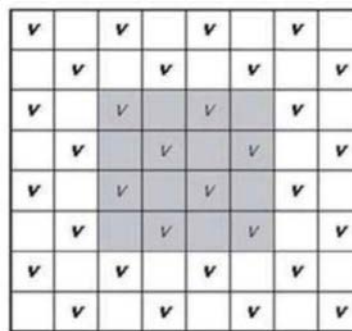


图2



GALF编码器决策的流程图

图3



垂直梯度的子采样位置

图4A

<i>H</i>		<i>H</i>		<i>H</i>		<i>H</i>	
	<i>H</i>		<i>H</i>		<i>H</i>		<i>H</i>
<i>H</i>		<i>H</i>		<i>H</i>		<i>H</i>	
	<i>H</i>		<i>H</i>		<i>H</i>		<i>H</i>
<i>H</i>		<i>H</i>		<i>H</i>		<i>H</i>	
	<i>H</i>		<i>H</i>		<i>H</i>		<i>H</i>
<i>H</i>		<i>H</i>		<i>H</i>		<i>H</i>	
	<i>H</i>		<i>H</i>		<i>H</i>		<i>H</i>

水平梯度的子采样位置

图4B

<i>D1</i>		<i>D1</i>		<i>D1</i>		<i>D1</i>	
	<i>D1</i>		<i>D1</i>		<i>D1</i>		<i>D1</i>
<i>D1</i>		<i>D1</i>		<i>D1</i>		<i>D1</i>	
	<i>D1</i>		<i>D1</i>		<i>D1</i>		<i>D1</i>
<i>D1</i>		<i>D1</i>		<i>D1</i>		<i>D1</i>	
	<i>D1</i>		<i>D1</i>		<i>D1</i>		<i>D1</i>
<i>D1</i>		<i>D1</i>		<i>D1</i>		<i>D1</i>	
	<i>D1</i>		<i>D1</i>		<i>D1</i>		<i>D1</i>

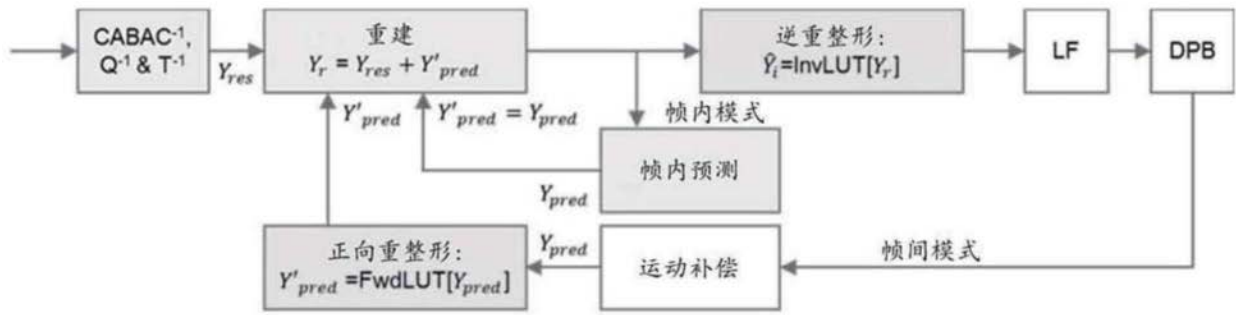
对角线梯度的子采样位置

图4C

<i>D2</i>		<i>D2</i>		<i>D2</i>		<i>D2</i>	
	<i>D2</i>		<i>D2</i>		<i>D2</i>		<i>D2</i>
<i>D2</i>		<i>D2</i>		<i>D2</i>		<i>D2</i>	
	<i>D2</i>		<i>D2</i>		<i>D2</i>		<i>D2</i>
<i>D2</i>		<i>D2</i>		<i>D2</i>		<i>D2</i>	
	<i>D2</i>		<i>D2</i>		<i>D2</i>		<i>D2</i>
<i>D2</i>		<i>D2</i>		<i>D2</i>		<i>D2</i>	
	<i>D2</i>		<i>D2</i>		<i>D2</i>		<i>D2</i>

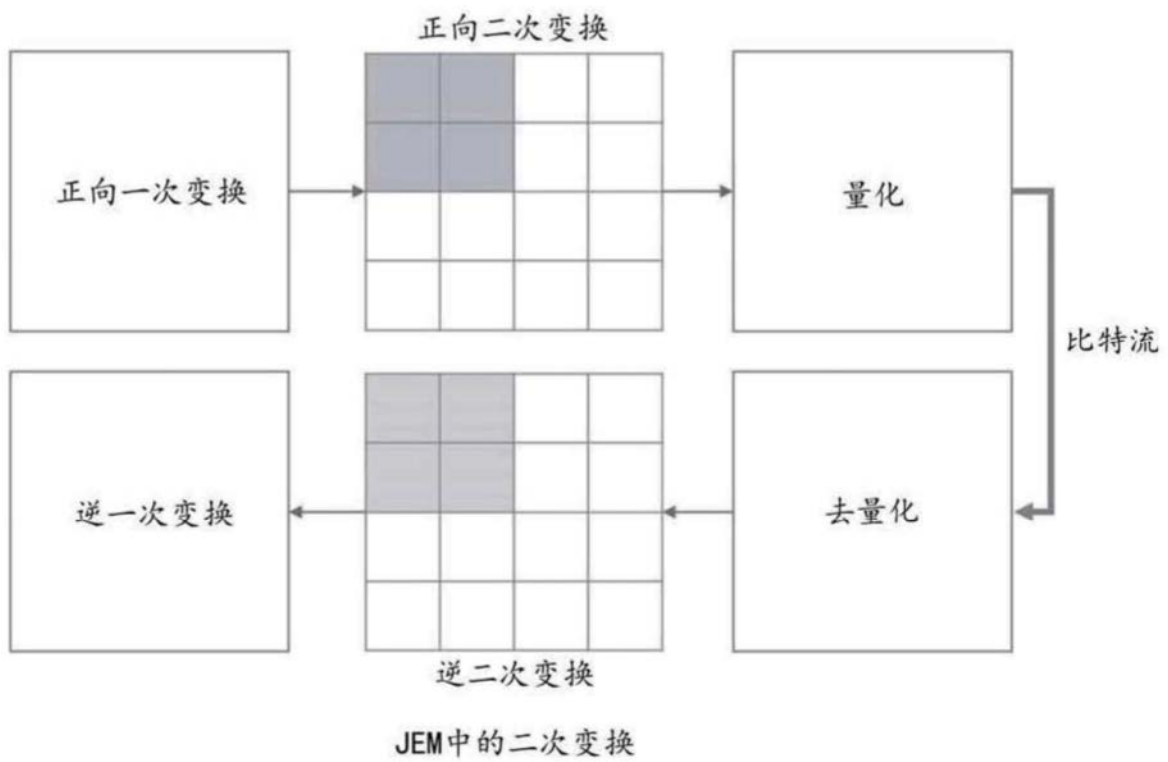
对角线梯度的子采样位置

图4D



具有重整形的解码流程的流程图

图5



JEM中的二次变换

图6

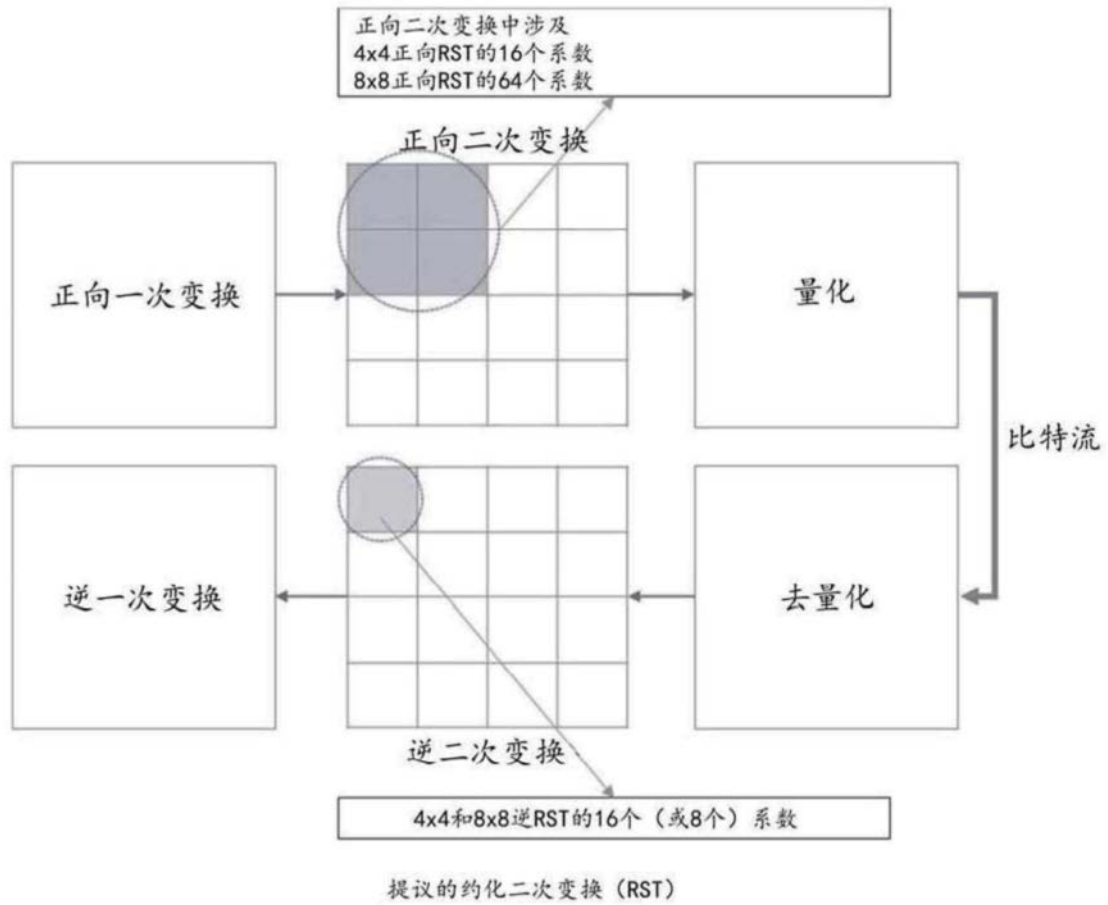
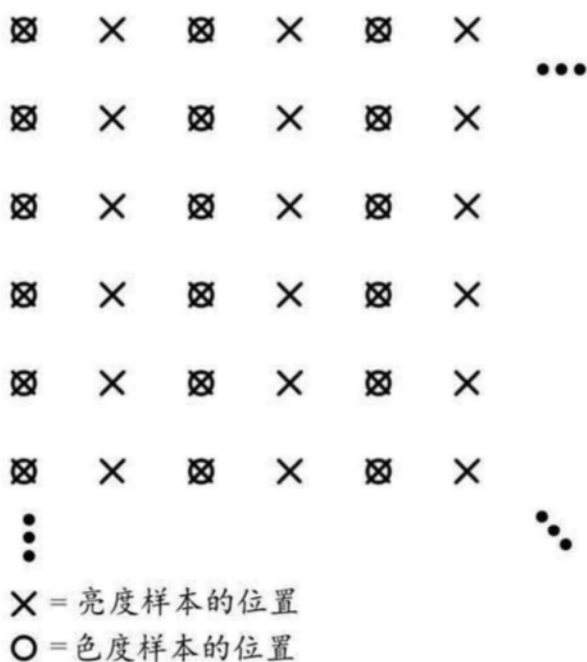


图7



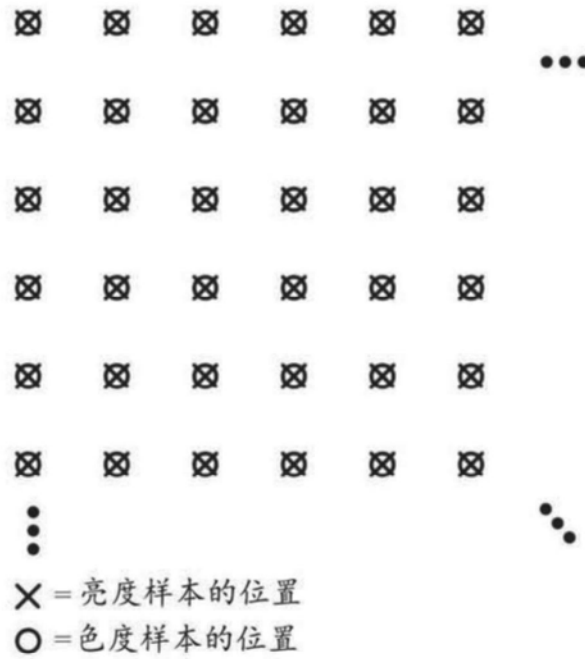
图片中4:2:0亮度和色度样本的标称垂直和水平位置

图8



图片中4:2:2亮度和色度样本的标称垂直和水平位置

图9



图片中4:2:4亮度和色度样本的标称垂直和水平位置

图10

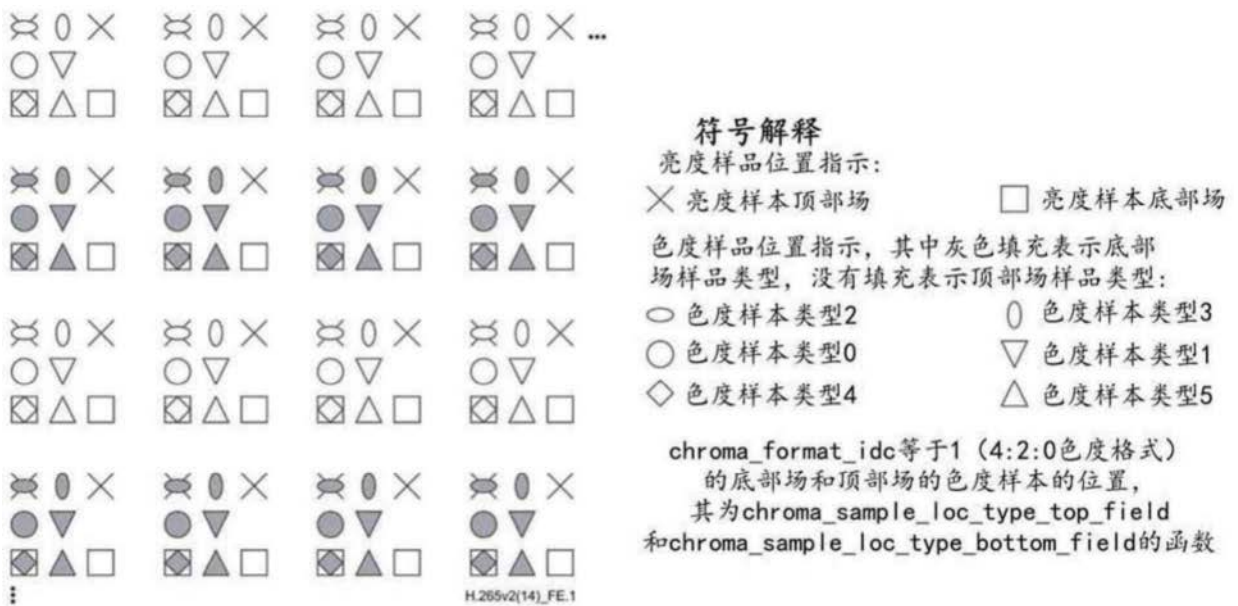
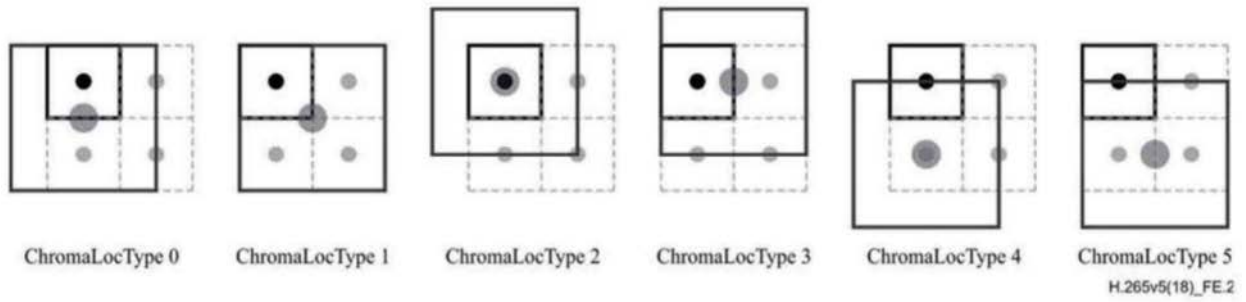
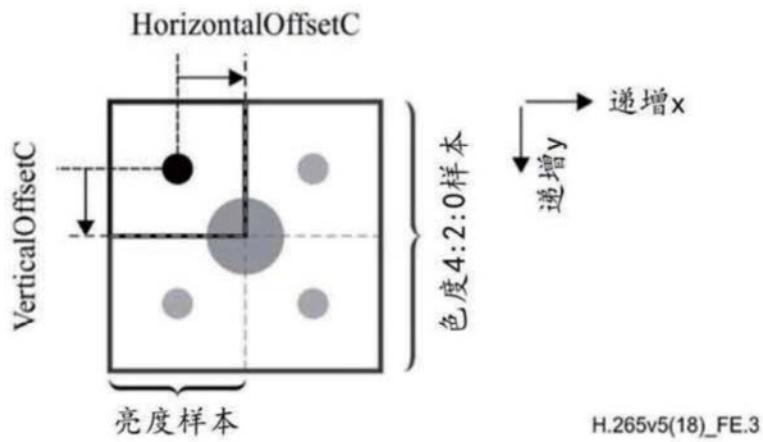


图11



当chroma_format_idc等于1 (4:2:0色度格式) 时的左上色度样本的位置, 其为ChromaLocType的函数

图12



当ChromaLocType等于1时, 当chroma_format_idc等于1 (4:2:0色度格式) 时的左上色度样本的位置

图13

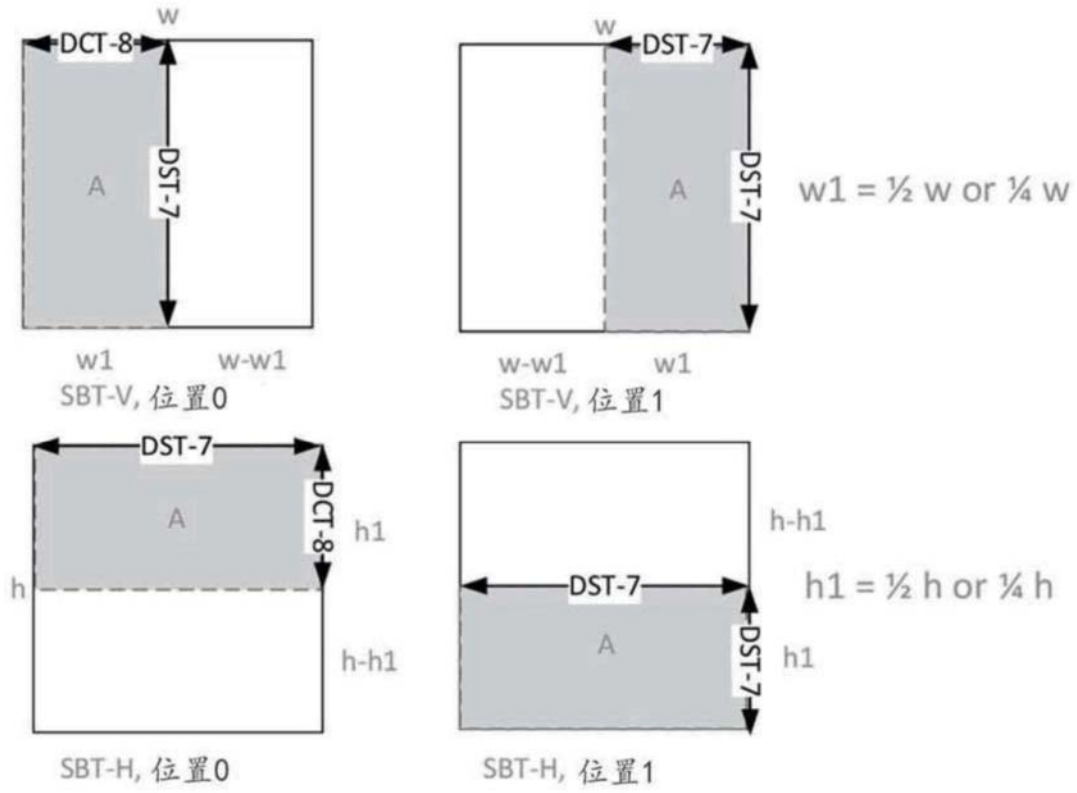


图14

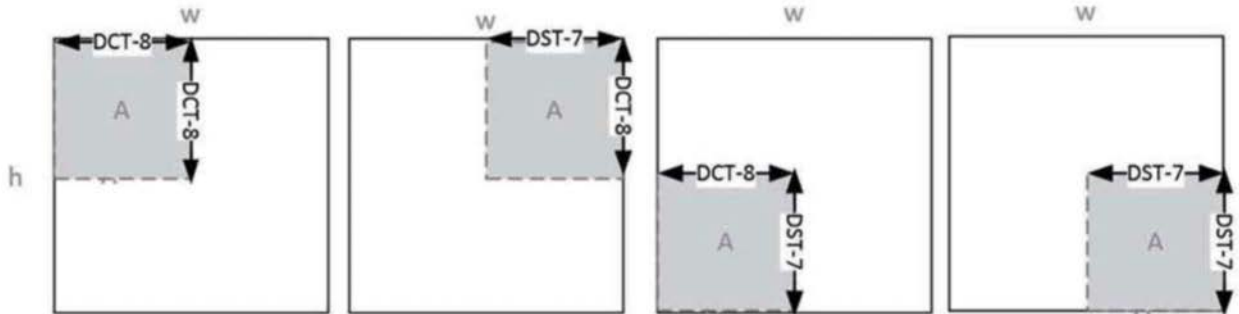


图15

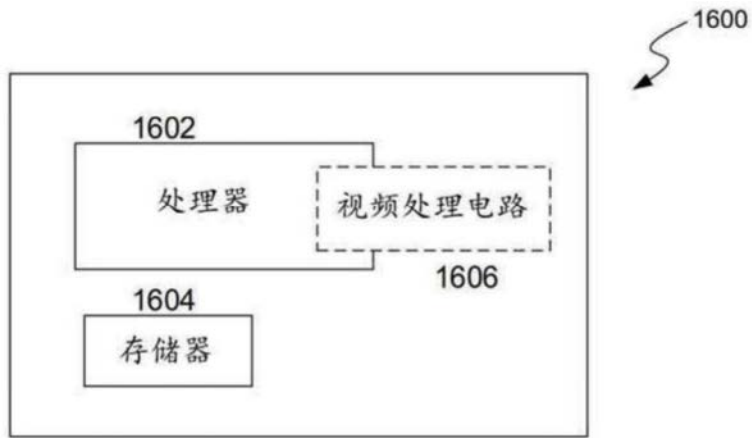


图16

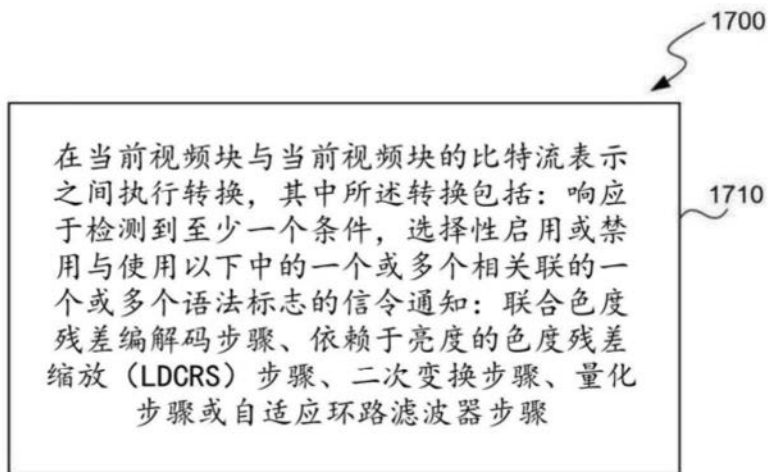


图17

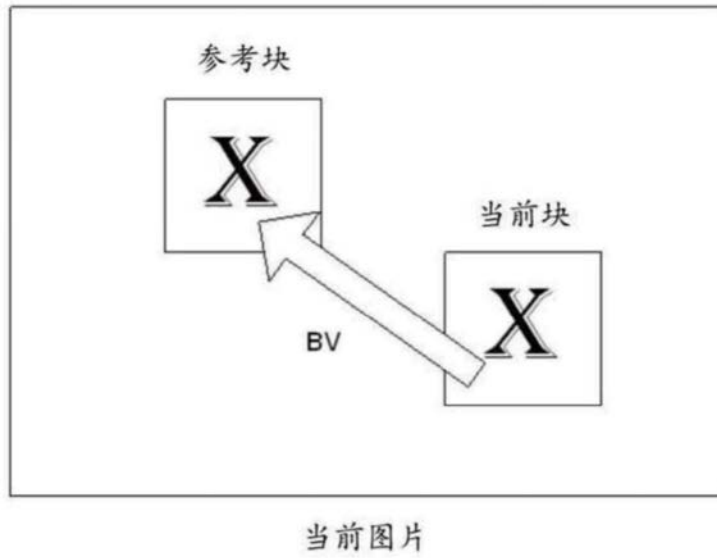


图18

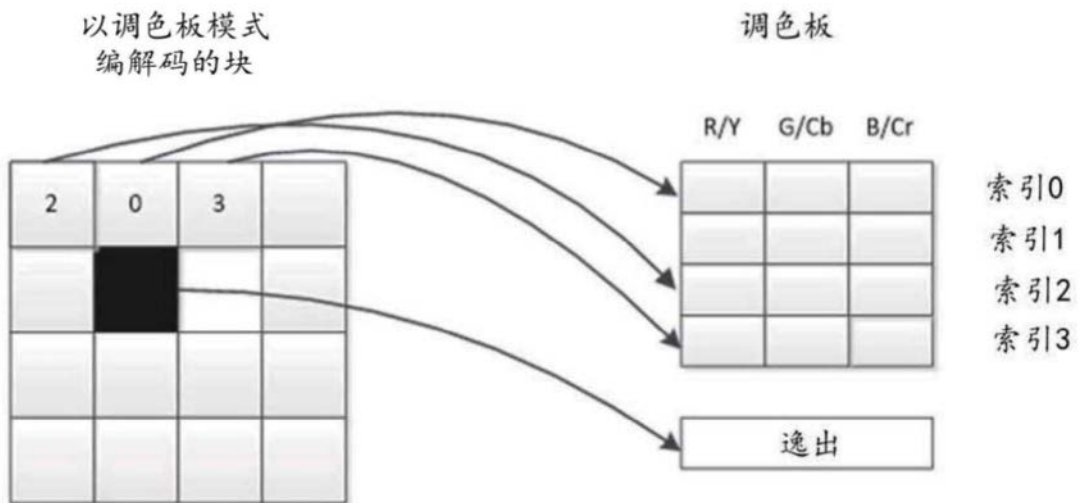


图19

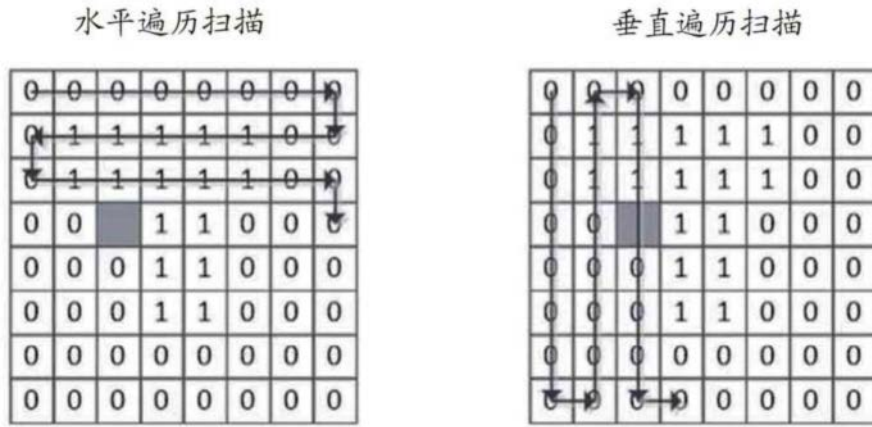


图20

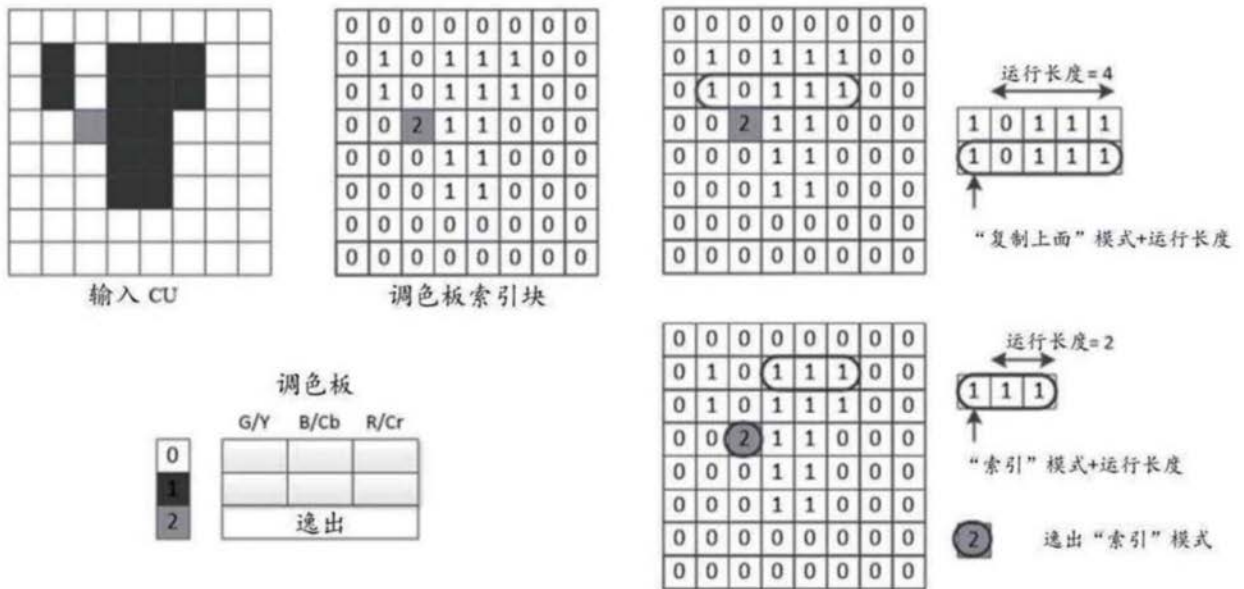


图21

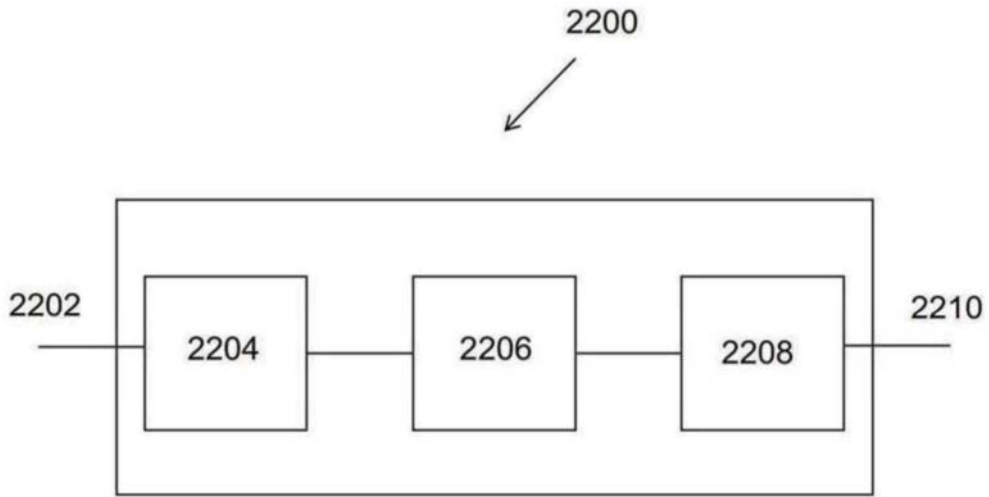


图22

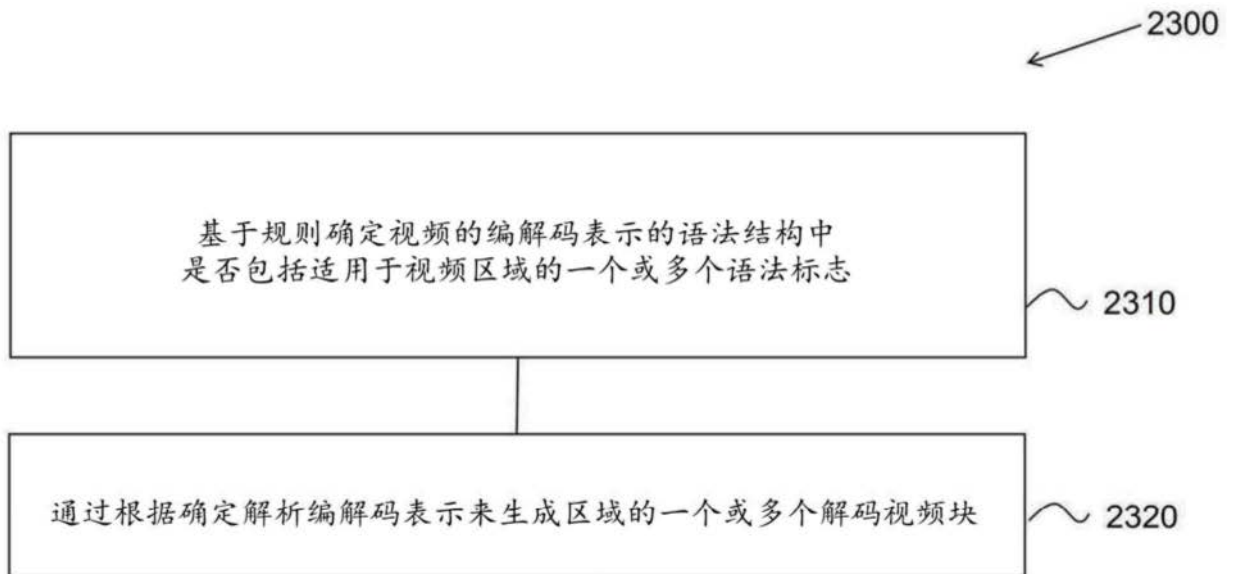


图23

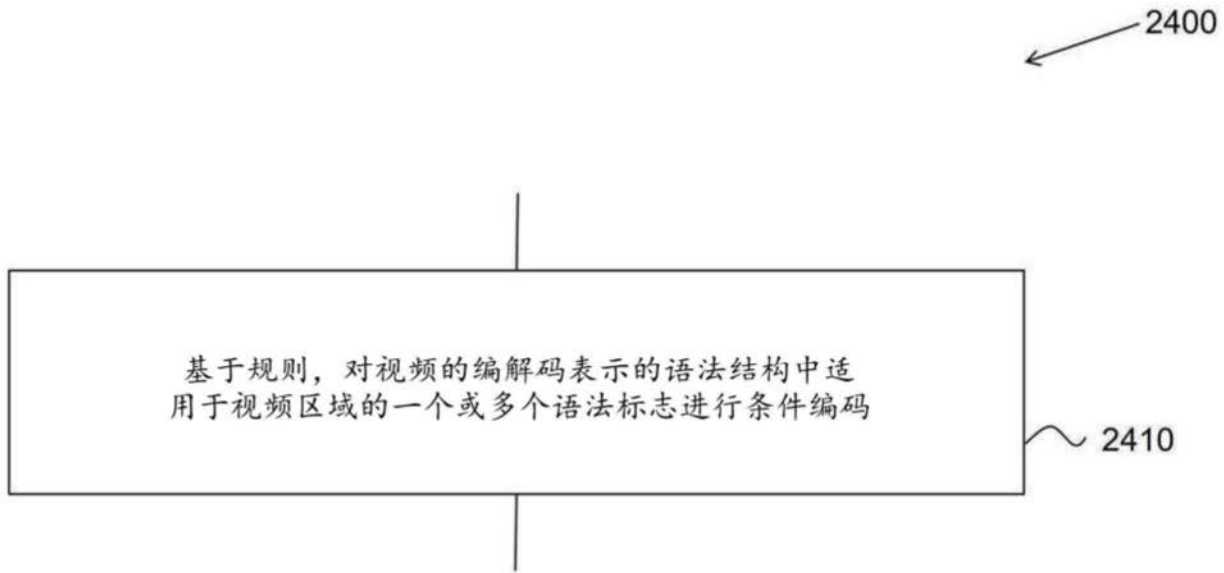


图24

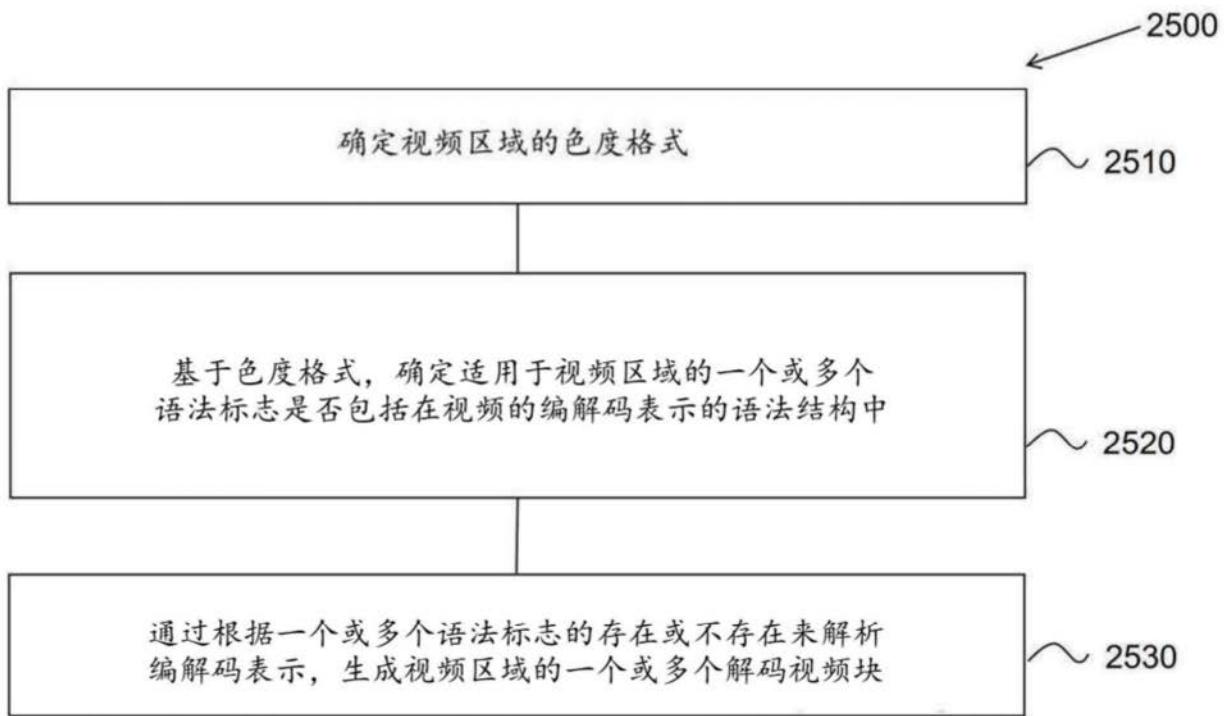


图25

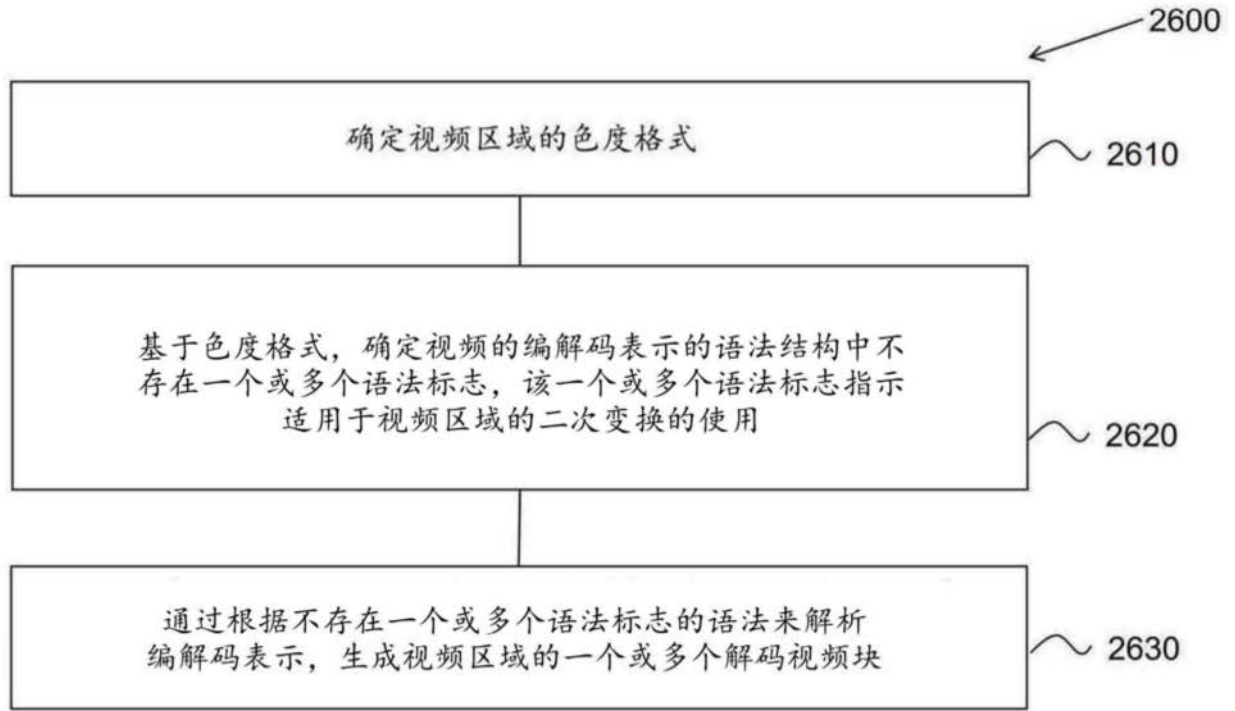


图26

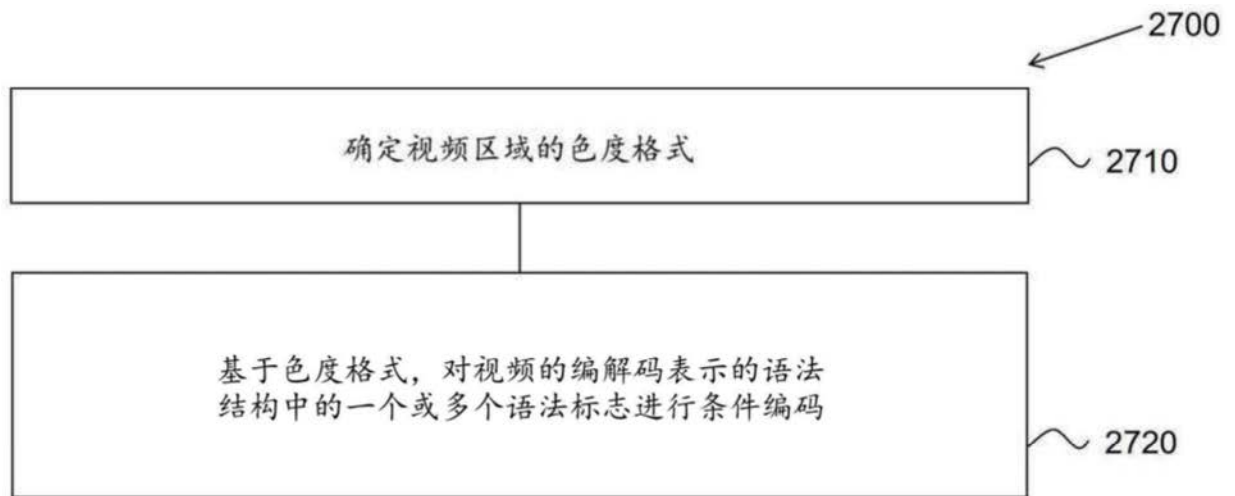


图27

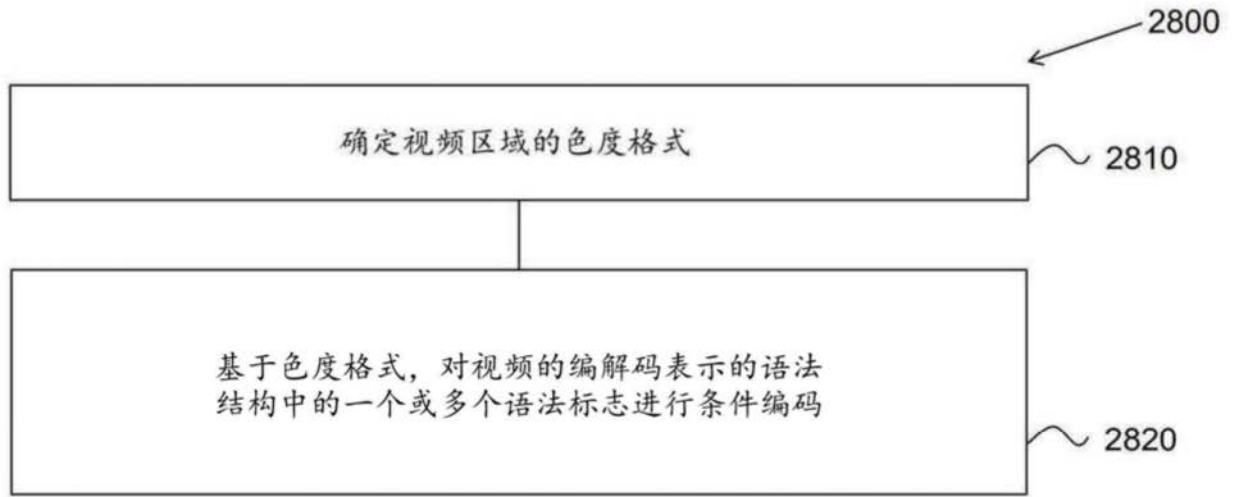


图28

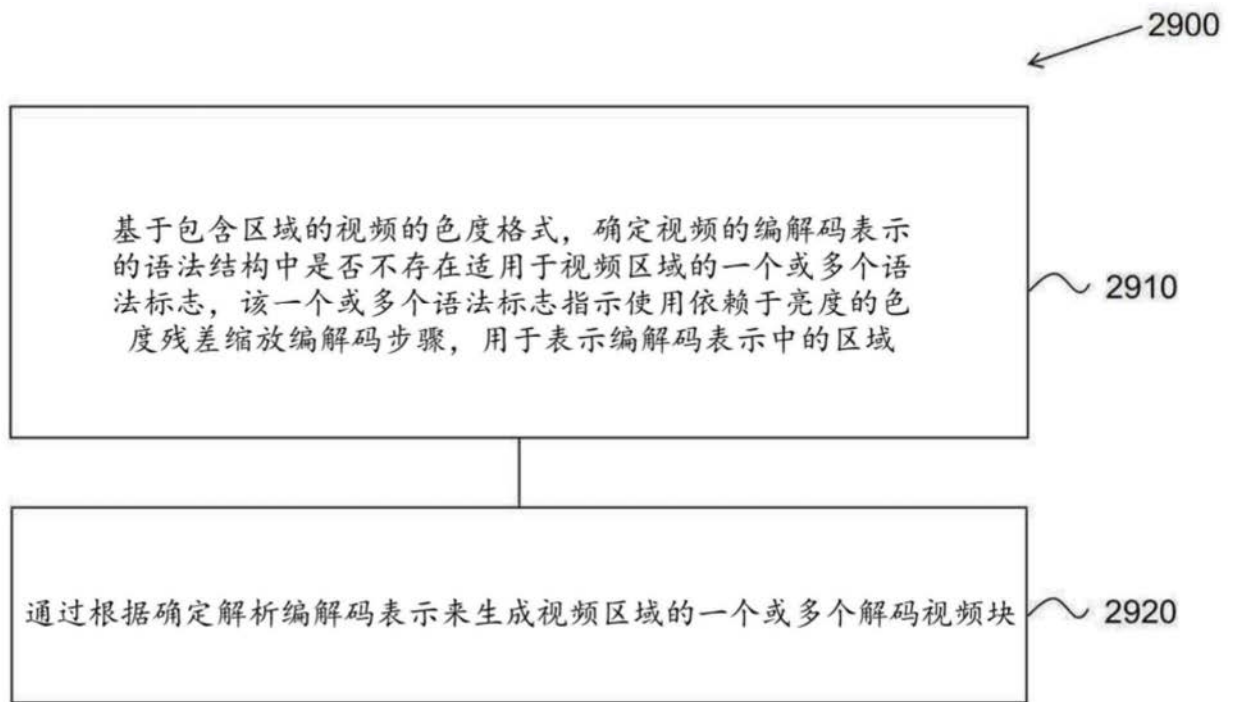


图29A

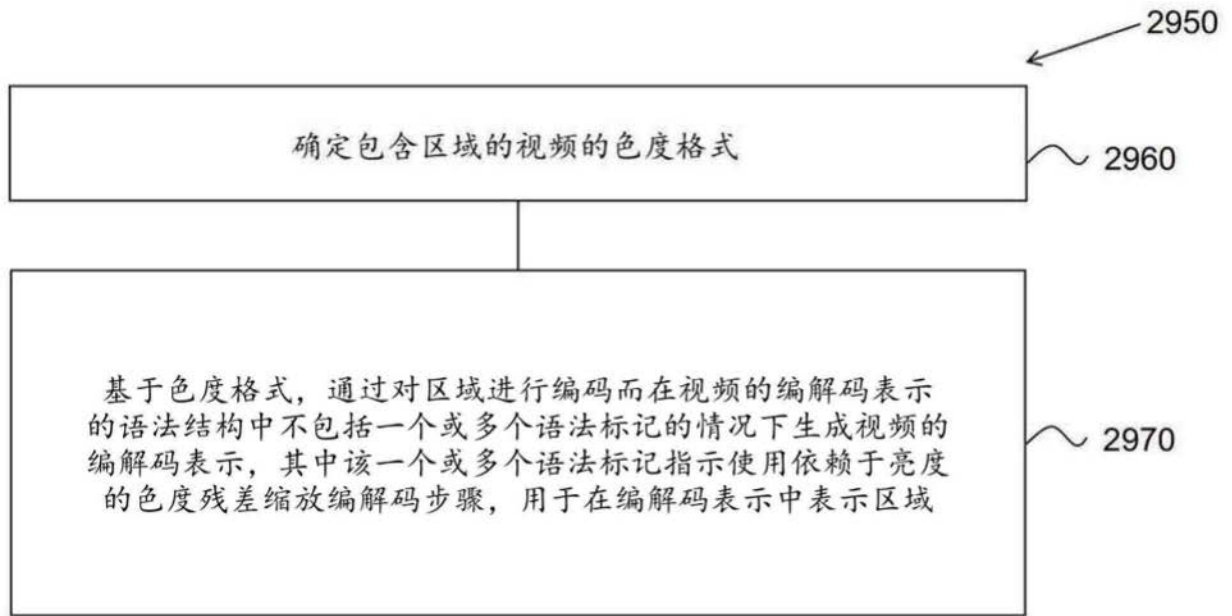


图29B

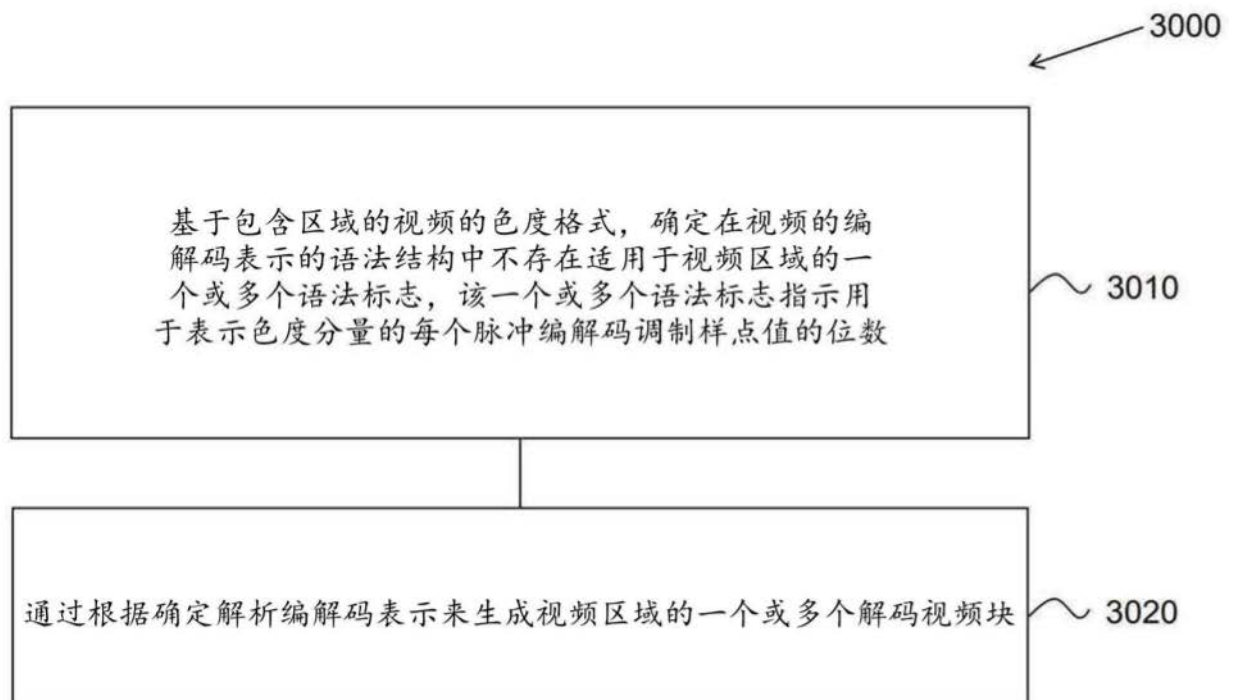


图30A

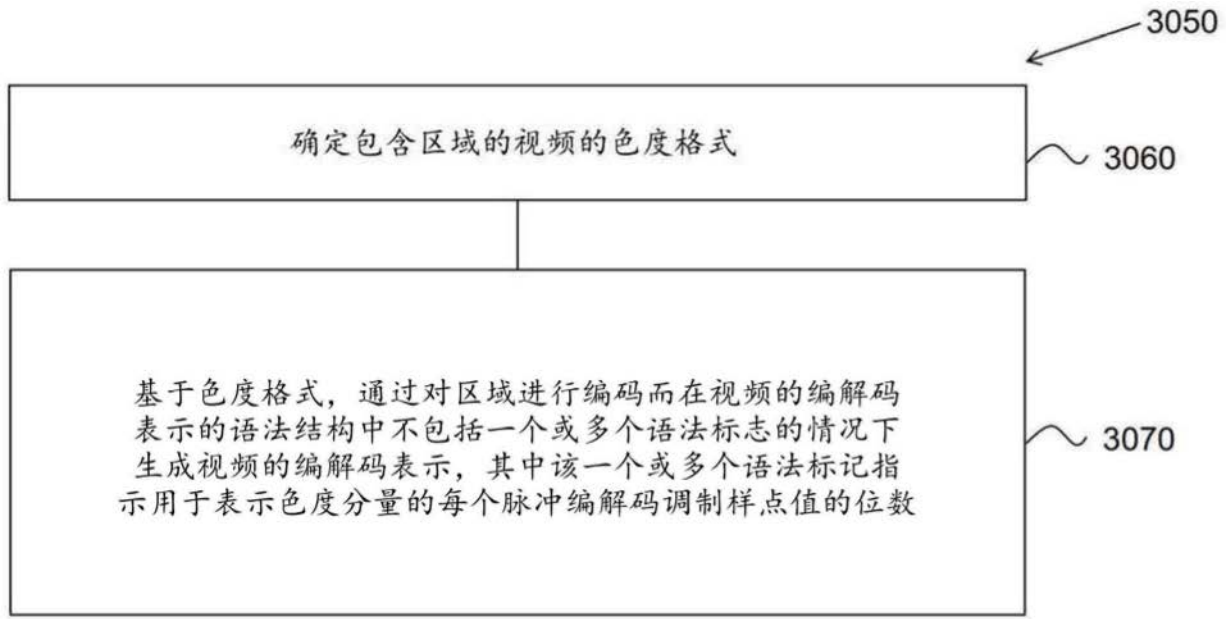


图30B

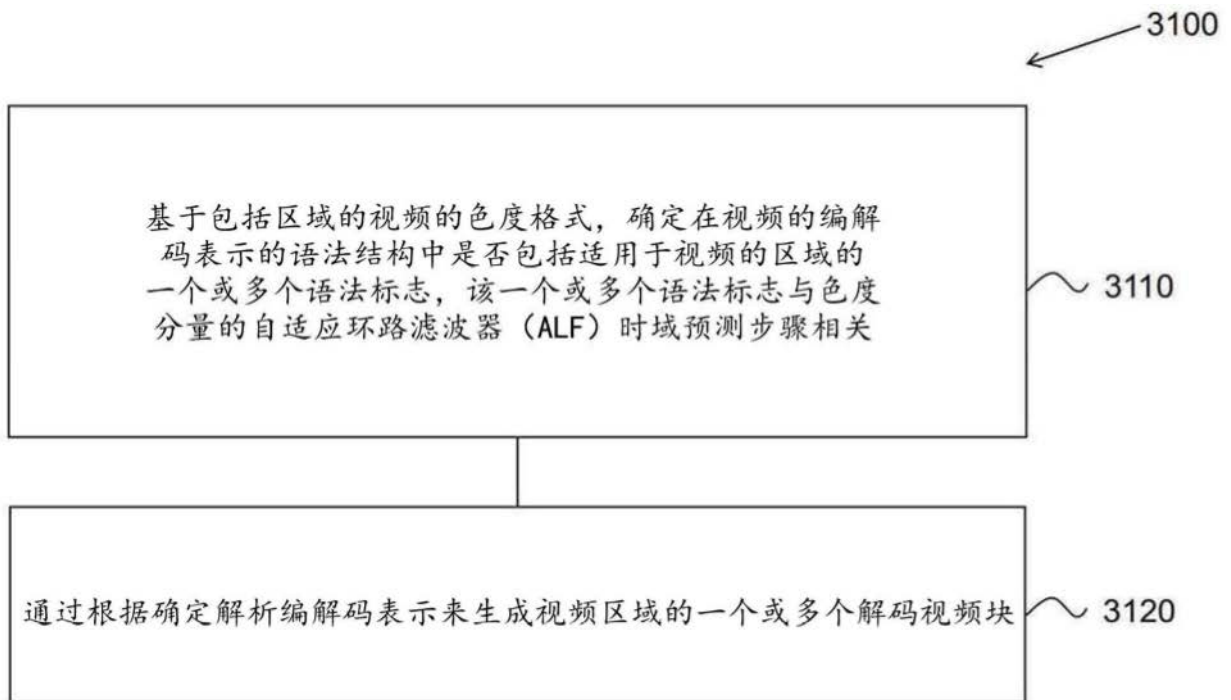


图31A

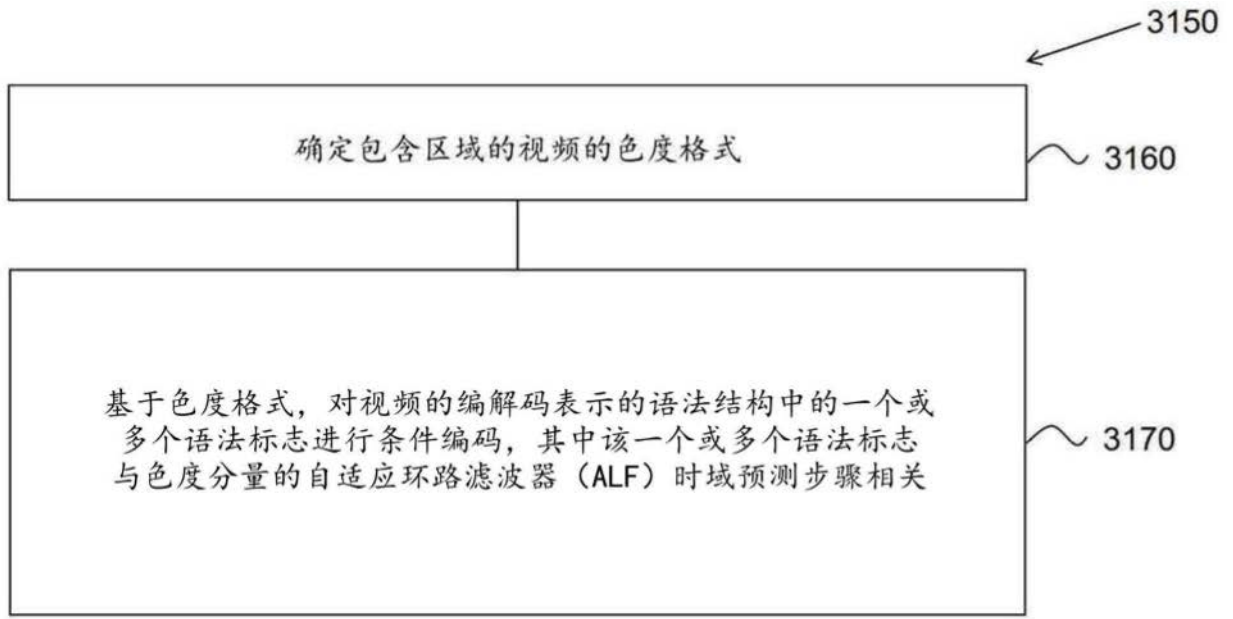


图31B

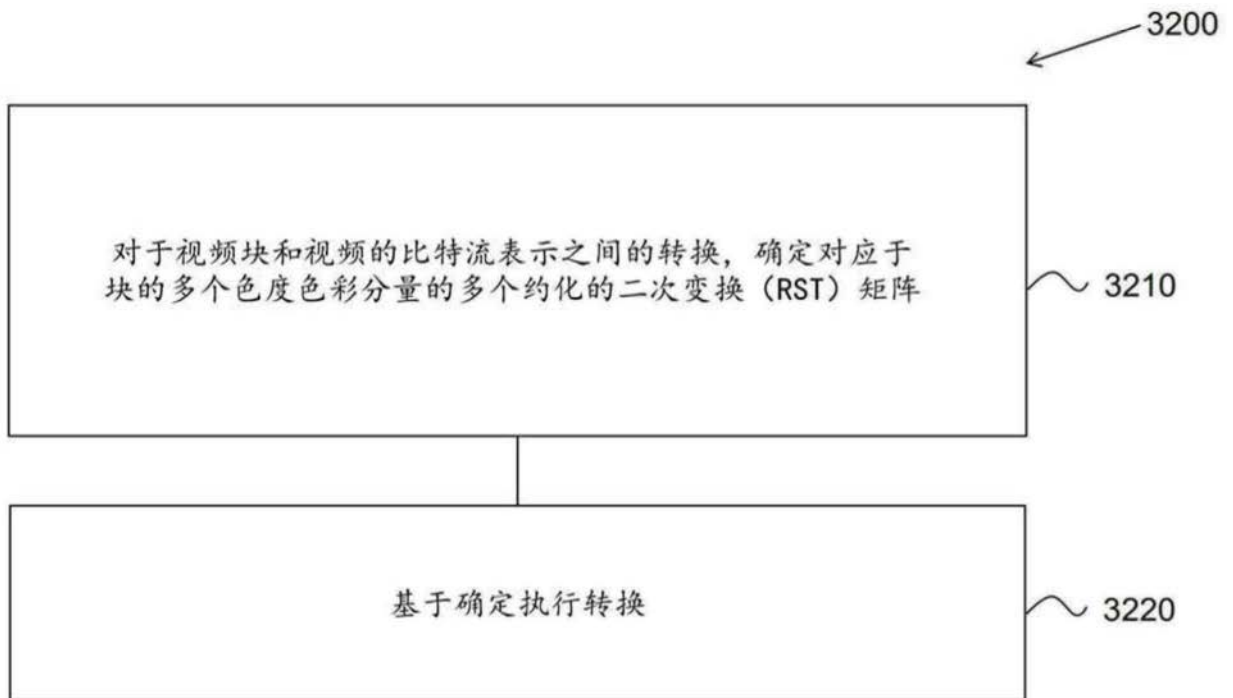


图32

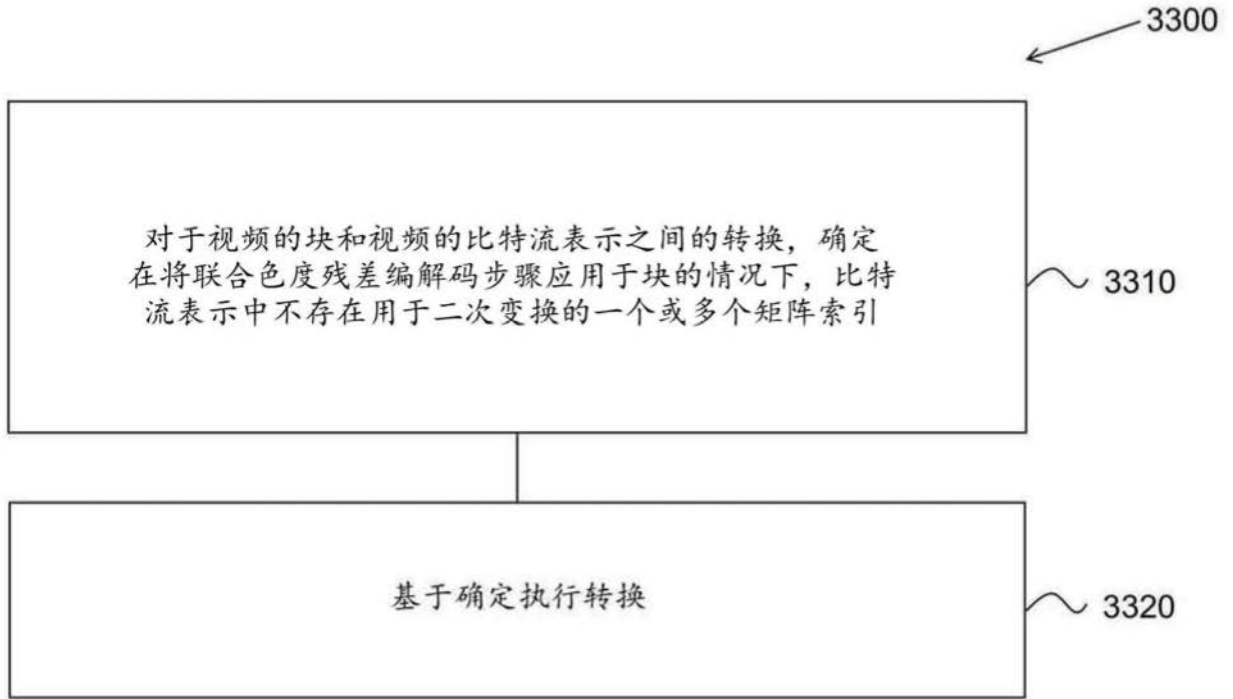


图33

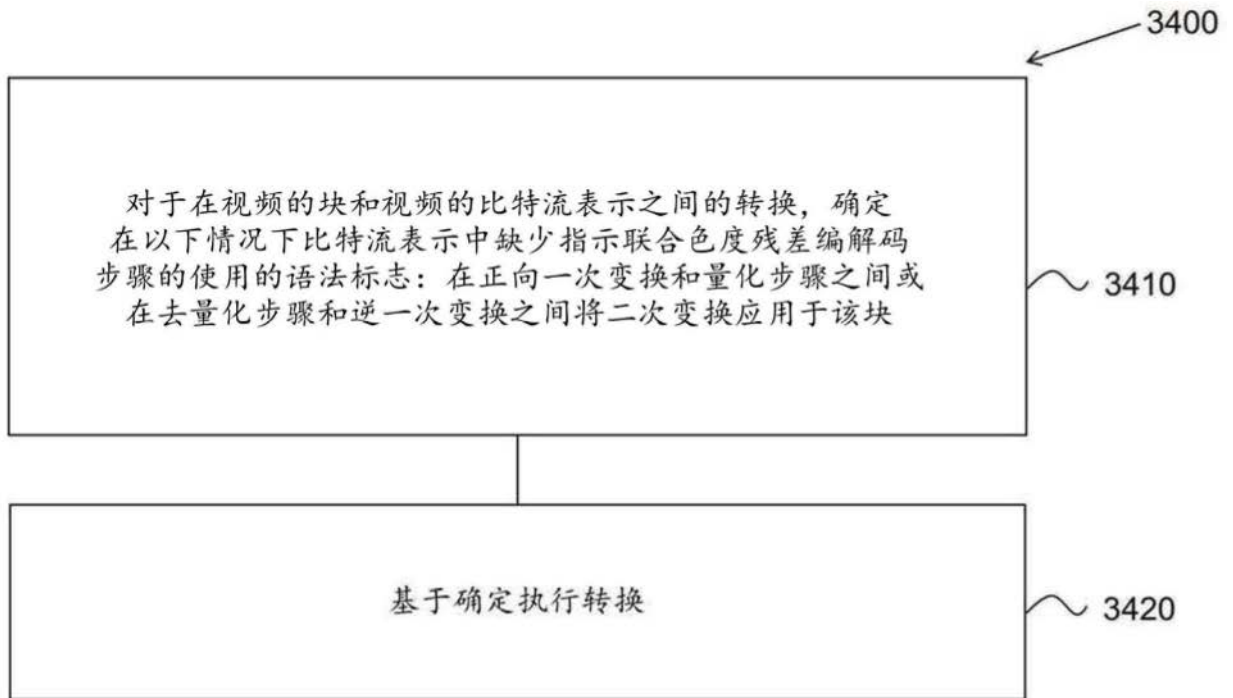


图34

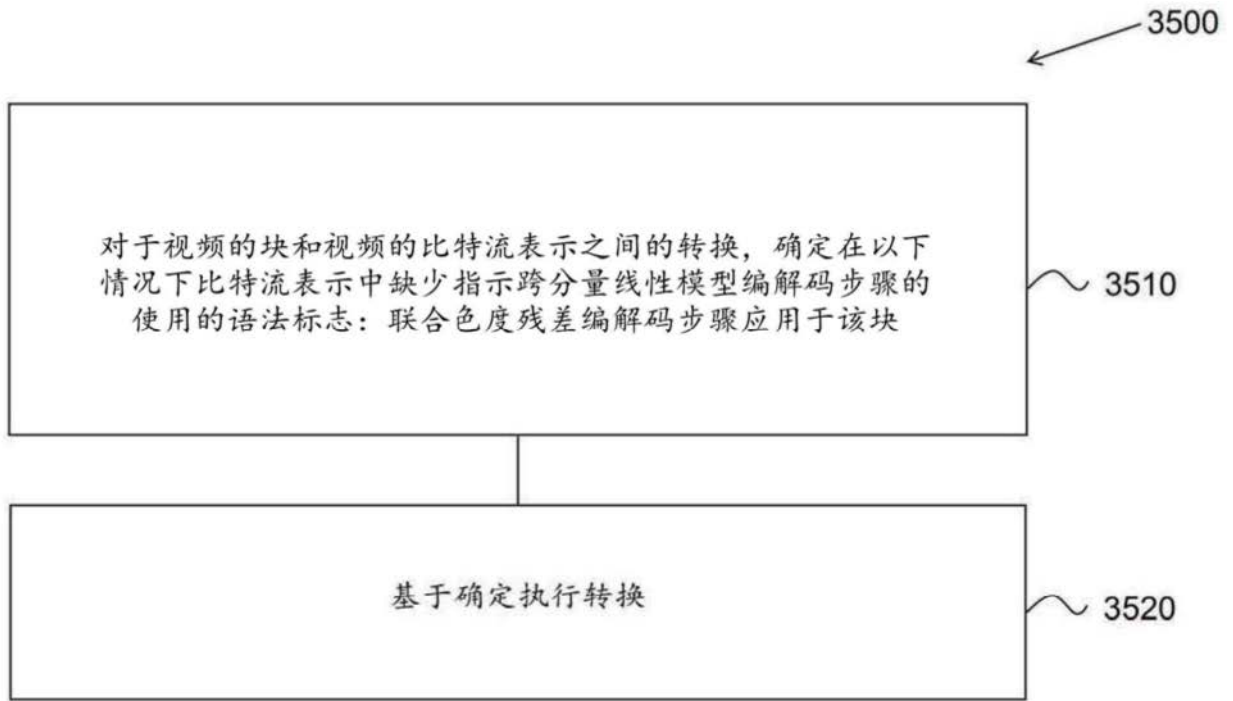


图35

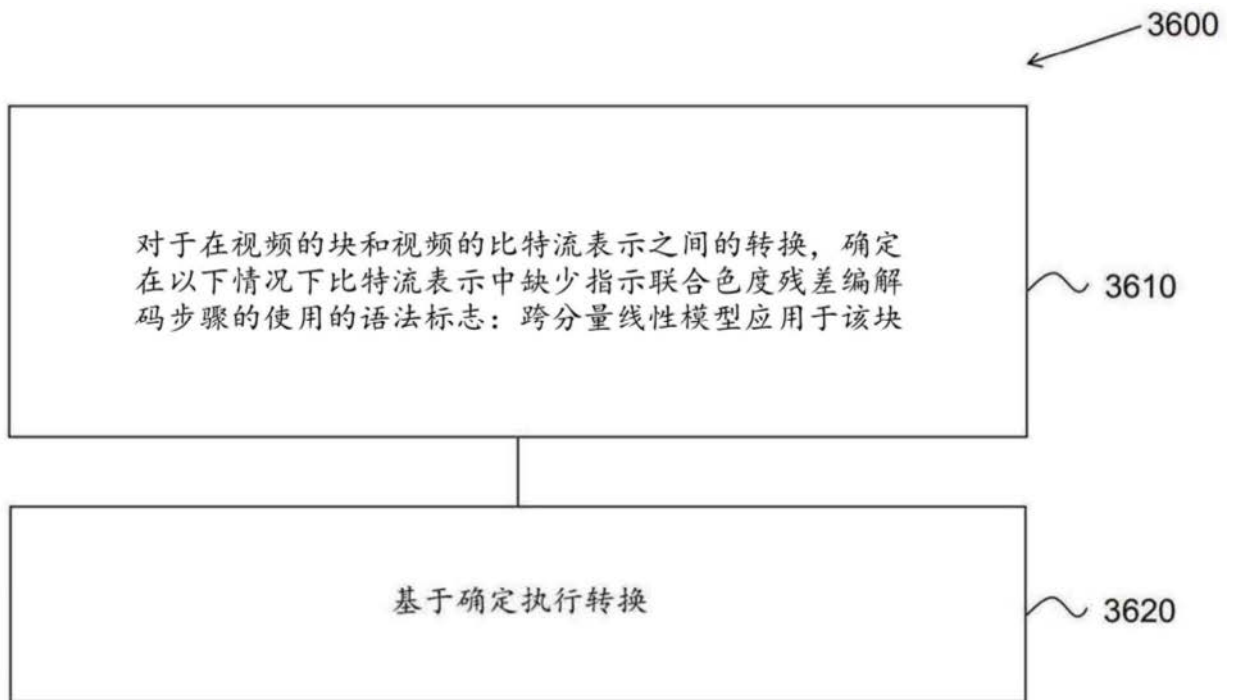


图36

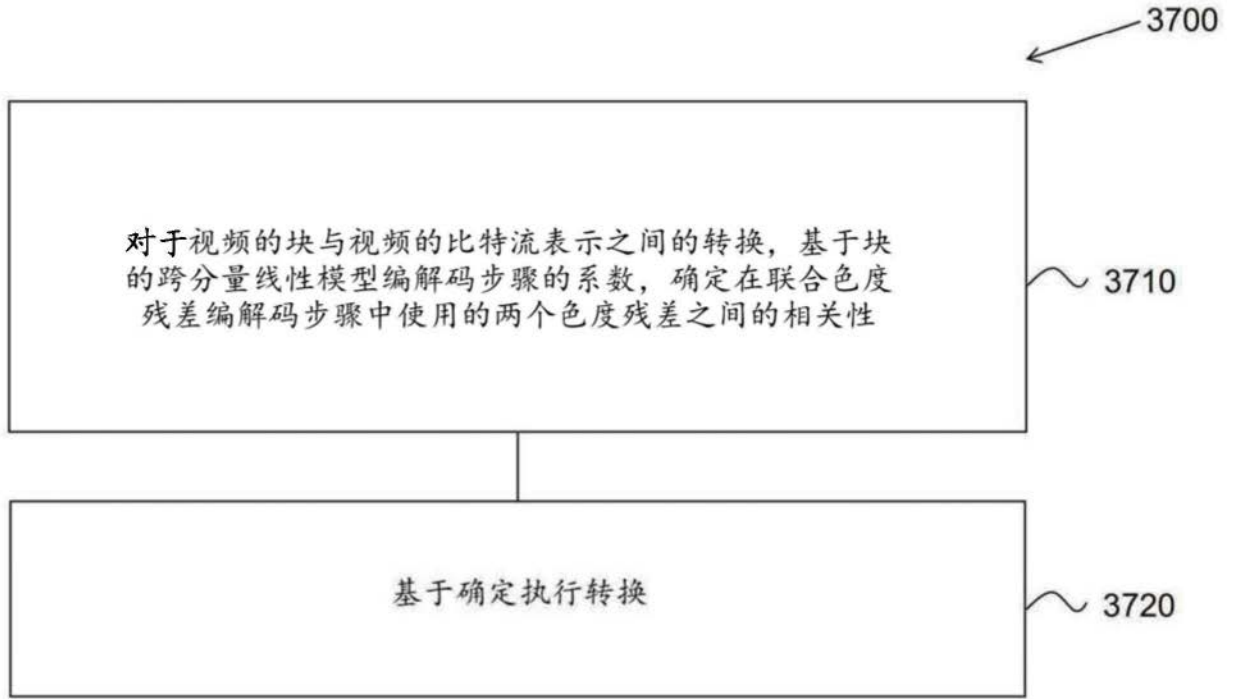


图37

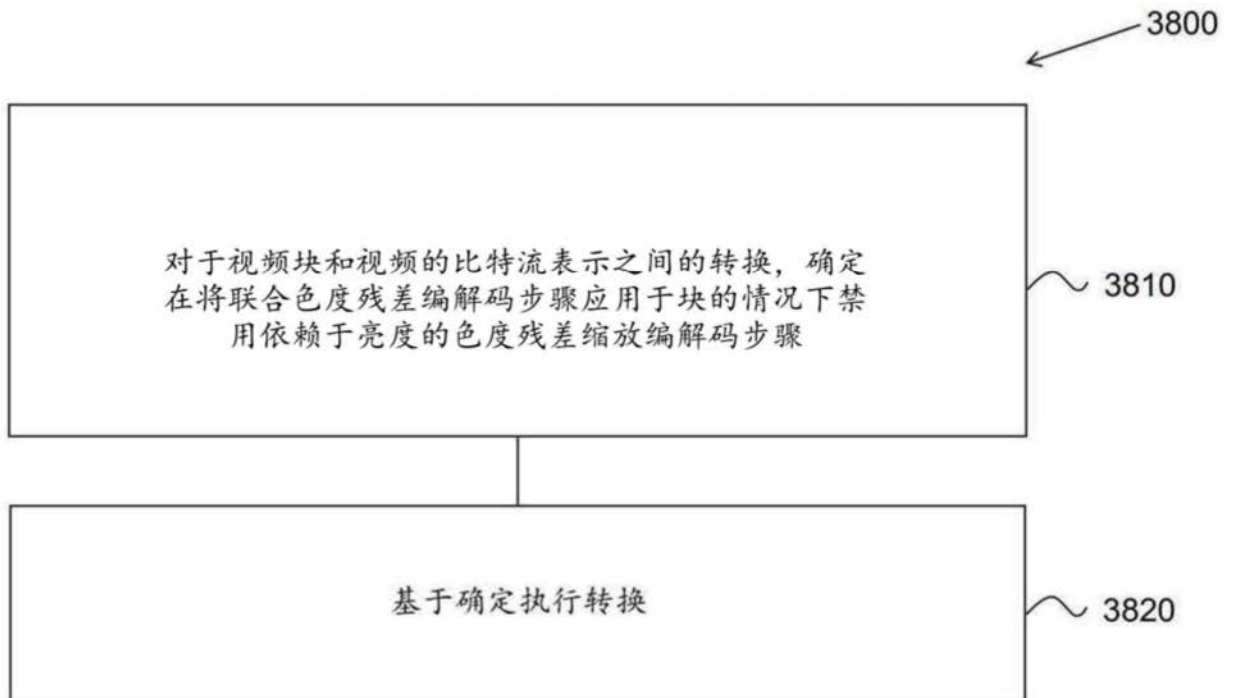


图38

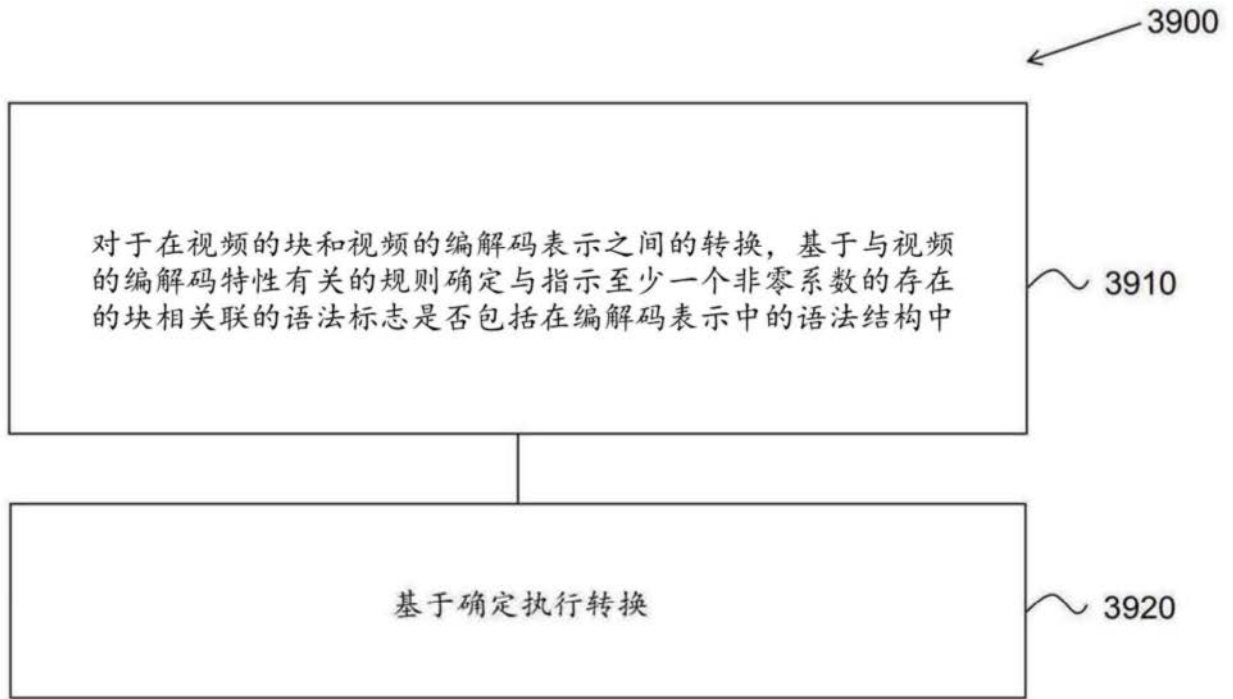


图39

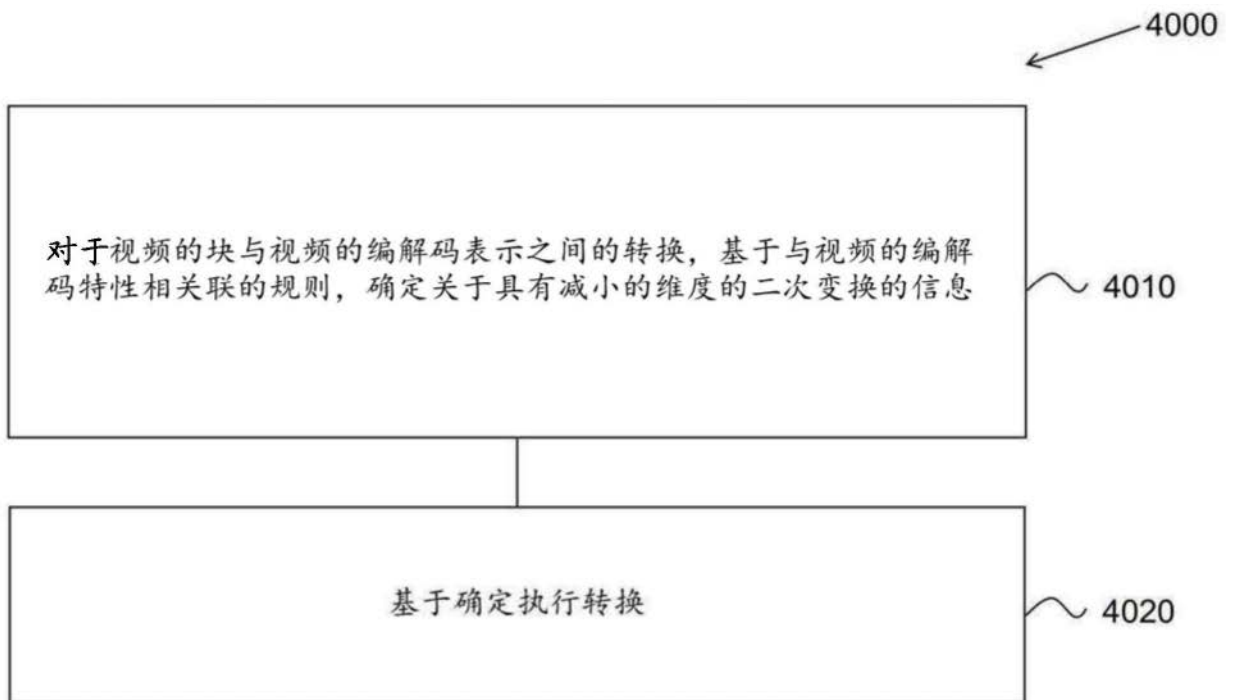


图40

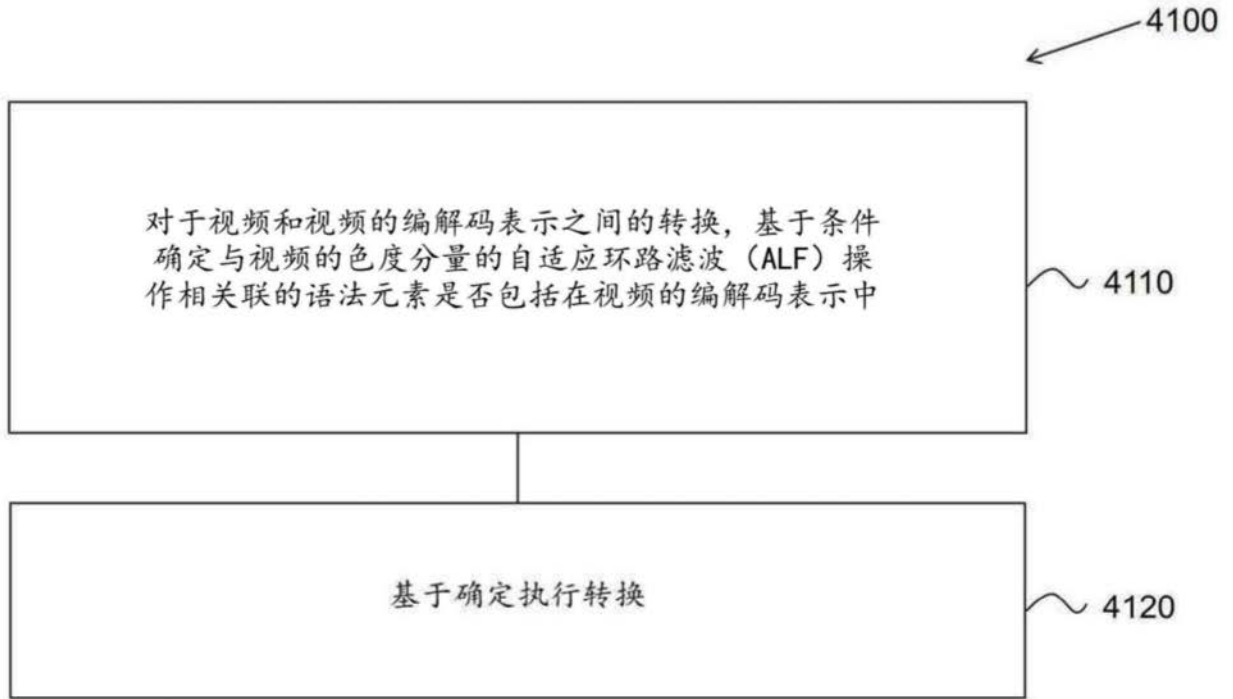


图41

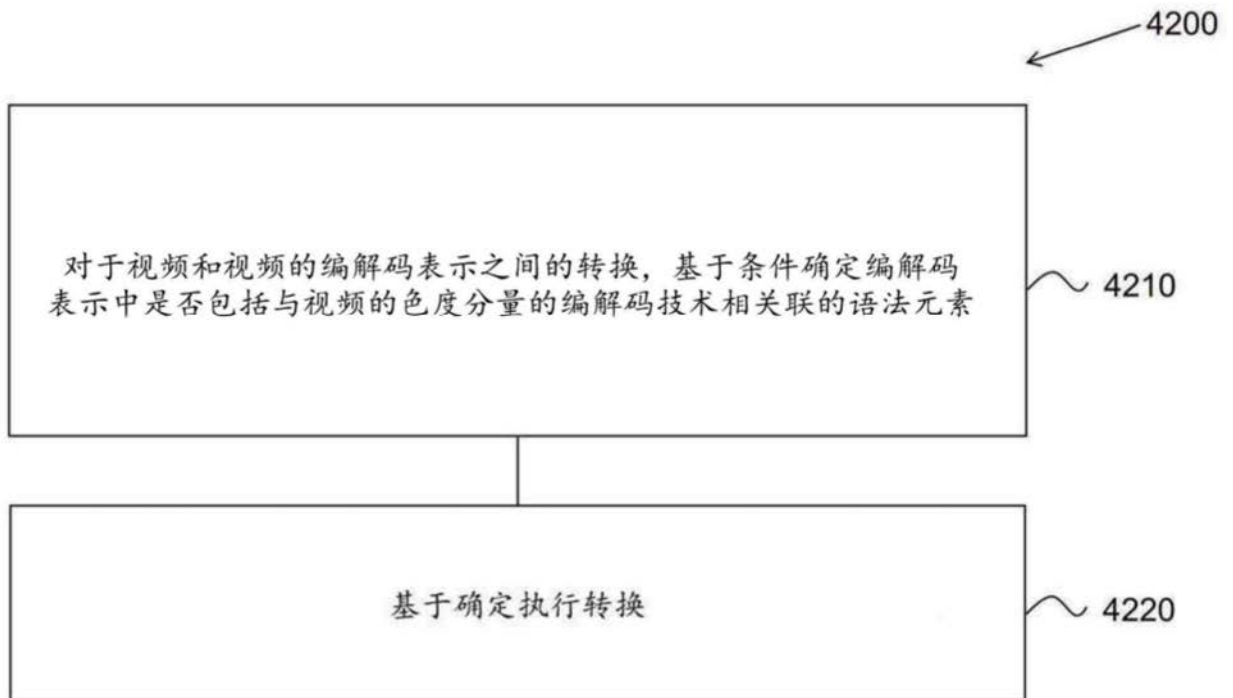


图42

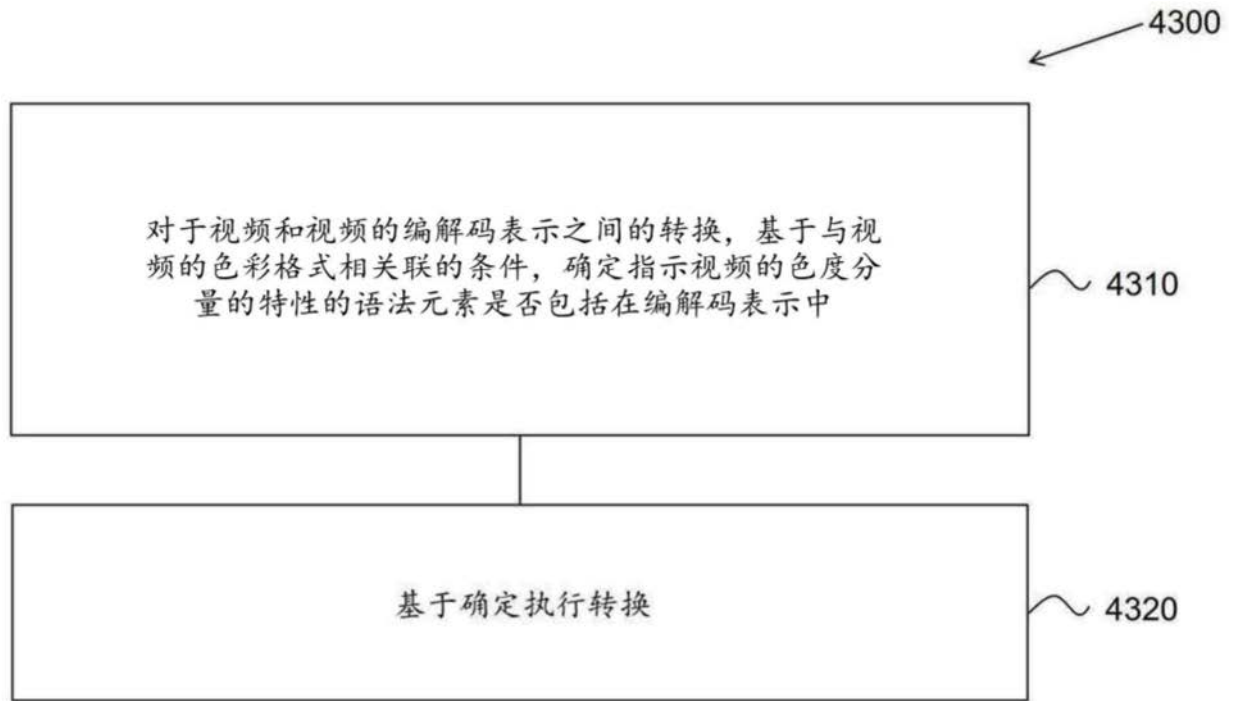


图43