



## (12) 发明专利申请

(10) 申请公布号 CN 118474354 A

(43) 申请公布日 2024. 08. 09

(21) 申请号 202410468092.X

H04N 19/14 (2014.01)

(22) 申请日 2020.02.28

H04N 19/176 (2014.01)

(30) 优先权数据

H04N 19/186 (2014.01)

62/813,728 2019.03.04 US

H04N 19/70 (2014.01)

62/817,546 2019.03.12 US

H04N 19/98 (2014.01)

(62) 分案原申请数据

202080018751.X 2020.02.28

(71) 申请人 阿里巴巴集团控股有限公司

地址 英属开曼群岛大开曼

(72) 发明人 叶琰 陈杰 廖如伶

(74) 专利代理机构 北京三友知识产权代理有限公司

公司 11127

专利代理师 刘爱勤 王小东

(51) Int. Cl.

H04N 19/117 (2014.01)

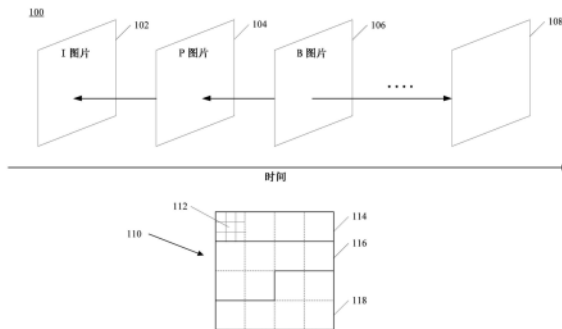
权利要求书4页 说明书29页 附图15页

(54) 发明名称

视频解码方法和装置、处理视频内容的方法和装置及介质

(57) 摘要

本公开提供了视频解码方法和装置、处理视频内容的方法和装置及介质。一种计算机实现的视频解码方法包括以下步骤：重构与图片相关联的多个亮度样本；以及重构与所述图片相关联的色度块，其中，重构色度块包括：确定所述色度块是否具有非零残差；以及响应于确定所述色度块具有一个或多个非零色度残差，在重构所述色度块之前，确定多个重构的亮度样本的平均值，并基于所述平均值来对所述色度块的残差进行缩放。



1. 一种计算机实现的视频解码方法,所述方法包括以下步骤:  
重构与图片相关联的多个亮度样本;以及  
重构与所述图片相关联的色度块,  
其中,重构色度块包括:  
确定所述色度块是否具有非零残差;以及  
响应于确定所述色度块具有一个或更多个非零色度残差,在重构所述色度块之前,确定多个重构的亮度样本的平均值,并且基于所述平均值来对所述色度块的残差进行缩放。
2. 根据权利要求1所述的方法,其中,重构色度块进一步包括:  
响应于确定所述色度块不具有非零色度残差,绕过确定多个重构的亮度样本的平均值。
3. 根据权利要求1所述的方法,其中,基于所述多个重构的亮度样本的平均值来对所述色度块的残差进行缩放包括:  
基于所述多个重构的亮度样本的所述平均值来确定色度缩放因子;以及  
将所述色度缩放因子应用于所述色度块的所述残差。
4. 根据权利要求1所述的方法,其中,所述色度块是否具有非零残差是基于与所述色度块相关联的编码块标志的值来确定的。
5. 根据权利要求4所述的方法,所述方法进一步包括以下步骤:  
响应于所述编码块标志的所述值等于1,确定所述色度块具有一个或更多个非零残差。
6. 根据权利要求4所述的方法,所述方法进一步包括以下步骤:  
响应于所述编码块标志的所述值等于0,确定所述色度块不具有非零残差。
7. 根据权利要求1所述的方法,其中,所述色度块是否具有非零残差是基于以下值来确定的:  
与所述色度块的第一色度分量相关联的第一编码块标志的值,以及  
与所述色度块的第二色度分量相关联的第二编码块标志的值。
8. 根据权利要求7所述的方法,所述方法进一步包括以下步骤:  
响应于所述第一编码块标志的所述值或所述第二编码块标志的所述值中的至少一个等于1,确定所述色度块具有一个或更多个非零残差。
9. 根据权利要求7所述的方法,所述方法进一步包括以下步骤:  
响应于所述第一编码块标志的所述值和所述第二编码块标志的所述值都等于0,确定所述色度块不具有非零残差。
10. 一种视频解码装置,所述视频解码装置包括:  
一个或更多个存储器,所述一个或更多个存储器存储有指令集;以及  
一个或更多个处理器,所述一个或更多个处理器被配置为执行所述指令集,以使所述视频解码装置执行以下操作:  
重构与图片相关联的多个亮度样本;以及  
重构与所述图片相关联的色度块,  
其中,在重构所述色度块时,所述一个或更多个处理器被配置为执行所述指令集,以使所述视频解码装置进一步执行以下操作:  
确定所述色度块是否具有非零残差;以及

响应于确定所述色度块具有一个或更多个非零色度残差,在重构所述色度块之前,确定多个重构的亮度样本的平均值,并且基于所述平均值来对所述色度块的残差进行缩放。

11.根据权利要求10所述的视频解码装置,其中,所述一个或更多个处理器被配置为执行所述指令集,以使所述视频解码装置进一步执行以下操作:

响应于确定所述色度块不具有非零色度残差,绕过确定多个重构的亮度样本的平均值。

12.根据权利要求10所述的视频解码装置,其中,所述一个或更多个处理器被配置为执行所述指令集,以使所述视频解码装置进一步执行以下操作:

基于所述多个重构的亮度样本的所述平均值来确定色度缩放因子;以及  
将所述色度缩放因子应用于所述色度块的所述残差。

13.根据权利要求10所述的视频解码装置,其中,所述一个或更多个处理器被配置为执行所述指令集,以使所述视频解码装置进一步执行以下操作:

基于与所述色度块相关联的编码块标志的值,确定所述色度块是否具有非零残差。

14.根据权利要求13所述的视频解码装置,其中,所述一个或更多个处理器被配置为执行所述指令集,以使所述视频解码装置进一步执行以下操作:

响应于所述编码块标志的所述值等于1,确定所述色度块具有一个或更多个非零残差。

15.根据权利要求13所述的视频解码装置,其中,所述一个或更多个处理器被配置为执行所述指令集,以使所述视频解码装置进一步执行以下操作:

响应于所述编码块标志的所述值等于0,确定所述色度块不具有非零残差。

16.一种非暂时性计算机可读存储介质,所述非暂时性计算机可读存储介质存储有与视频序列相关联的比特流,其中,所述比特流包括用于重构与图片相关联的色度块的编码信息,所述编码信息包括:

指示所述色度块是否具有非零残差的编码块标志,  
其中,根据包括以下步骤的方法来处理所述比特流:  
重构与所述图片相关联的多个亮度样本;以及

当所述编码块标志指示所述色度块具有一个或更多个非零色度残差时,在重构所述色度块之前,确定多个重构的亮度样本的平均值,并基于所述平均值来对所述色度块的残差进行缩放。

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,所述方法进一步包括以下步骤:

当所述编码块标志指示所述色度块不具有非零色度残差时,绕过确定多个重构的亮度样本的平均值。

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,基于所述多个重构的亮度样本的平均值来对所述色度块的残差进行缩放包括:

基于所述多个重构的亮度样本的平均值来确定色度缩放因子;以及  
将所述色度缩放因子应用于所述色度块的所述残差。

19.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,所述编码块标志的值等于1指示所述色度块具有非零残差,并且所述编码块标志的所述值等于0指示所述色度块不具有非零残差。

20. 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, 所述编码块标志包括:

与所述色度块的第一色度分量相关联的第一编码块标志, 以及

与所述色度块的第二色度分量相关联的第二编码块标志, 其中:

当所述第一编码块标志的值或所述第二编码块标志的值中的至少一个等于1时, 指示所述色度块具有一个或多个非零残差; 并且

当所述第一编码块标志的所述值和所述第二编码块标志的所述值都等于0时, 指示所述色度块不具有非零残差。

21. 一种用于处理视频内容的计算机实现的方法, 所述方法包括以下步骤:

接收与图片相关联的色度块, 所述色度块包括多个色度样本;

确定所述图片中的多个亮度样本的平均值; 以及

基于所述平均值来确定色度缩放因子,

其中, 所述色度缩放因子用于处理所述色度块的残差,

其中, 处理所述色度块的残差包括:

确定所述色度块是否具有非零残差; 以及

响应于确定所述色度块具有非零残差, 将所述色度缩放因子应用于所述色度块的所述残差。

22. 根据权利要求21所述的方法, 其中, 所述色度块是否具有非零残差是基于与所述色度块相关联的编码块标志的值确定的。

23. 根据权利要求21所述的方法, 所述方法进一步包括以下步骤:

响应于与所述色度块相关联的编码块标志的值为1, 确定所述色度块具有非零残差。

24. 根据权利要求21所述的方法, 其中, 确定所述多个亮度样本的所述平均值是通过确定满足第一条件来触发的。

25. 根据权利要求24所述的方法, 其中, 所述第一条件包括:

响应于与所述色度块相关联的编码块标志的值为1, 确定所述色度块具有非零残差。

26. 一种用于处理视频内容的装置, 所述装置包括:

存储器, 所述存储器存储有指令集; 以及

处理器, 所述处理器联接至所述存储器并被配置为执行所述指令集, 以使所述装置执行以下操作:

接收与图片相关联的色度块, 所述色度块包括多个色度样本;

确定所述图片中的多个亮度样本的平均值;

基于所述平均值来确定色度缩放因子; 以及

通过将所述色度缩放因子应用于所述色度块的残差来处理所述色度块的所述残差,

其中, 在通过将所述色度缩放因子应用于所述色度块的残差来处理所述色度块的所述残差时, 所述处理器被配置为执行所述指令集, 以使所述装置进一步执行以下操作:

确定所述色度块是否具有非零残差; 以及

响应于确定所述色度块具有非零残差, 将所述色度缩放因子应用于所述色度块的所述残差。

27. 根据权利要求26所述的装置, 其中,

所述色度块是否具有非零残差是基于与所述色度块相关联的编码块标志的值确定的。

28. 根据权利要求26所述的装置, 其中, 所述处理器被配置为执行所述指令集, 以使所述装置进一步执行以下操作:

响应于与所述色度块相关联的编码块标志的值为1, 确定所述色度块具有非零残差。

29. 一种非暂时性计算机可读存储介质, 所述非暂时性计算机可读存储介质存储有指令集, 所述指令集能够由设备的一个或多个处理器执行, 以使所述设备执行用于处理视频内容的方法, 所述方法包括以下步骤:

接收与图片相关联的色度块, 所述色度块包括多个色度样本;

确定所述图片中的多个亮度样本的平均值;

基于所述平均值来确定色度缩放因子; 以及

通过将所述色度缩放因子应用于所述色度块的残差来处理所述色度块的所述残差,

其中, 处理所述色度块的所述残差包括:

确定所述色度块是否具有非零残差; 以及

响应于确定所述色度块具有非零残差, 将所述色度缩放因子应用于所述色度块的所述残差。

30. 根据权利要求29所述的非暂时性计算机可读存储介质, 其中,

所述色度块是否具有非零残差是基于与所述色度块相关联的编码块标志的值确定的。

31. 根据权利要求29所述的非暂时性计算机可读存储介质, 其中, 所述方法进一步包括以下步骤:

响应于与所述色度块相关联的编码块标志的值为1, 确定所述色度块具有非零残差。

## 视频解码方法和装置、处理视频内容的方法和装置及介质

[0001] 本申请是申请号为202080018751.X的发明专利申请(国际申请号:PCT/US2020/020513,申请日:2020年02月28日,发明名称:用于处理视频内容的方法和装置及计算机可读存储介质)的分案申请。

[0002] 相关申请的交叉引用

[0003] 本公开要求于2019年3月4日提交的美国临时申请No.62/813,728以及于2019年3月12日提交的美国临时申请No.62/817,546的优先权的权益,上述两件美国临时申请的全部内容通过引用并入本文。

### 技术领域

[0004] 本公开总体上涉及视频处理,并且更具体地,涉及用于执行环内(in-loop)亮度映射与色度缩放(luma mapping with chroma scaling)的方法和系统。

### 背景技术

[0005] 视频编码系统通常用于压缩数字视频信号,例如,以减少消耗的存储空间或减少与这种信号相关联的传输带宽消耗。随着高清(HD)视频(例如,分辨率为 $1920 \times 1080$ 像素)在视频压缩的各种应用(诸如在线视频流传输、视频会议或视频监控)中越来越受欢迎,不断需要开发可以提高视频数据的压缩效率的视频编码工具。

[0006] 例如,视频监控应用在很多应用场景(例如,安全、交通、环境监测等)中的应用越来越广泛,并且监控设备的数量和分辨率也在不断快速增长。很多视频监控应用场景更喜欢向用户提供HD视频,以捕获更多信息,每帧具有更多像素来捕获这些信息。然而,HD视频比特流可能具有需要高传输带宽和大存储空间的高比特率。例如,平均分辨率为 $1920 \times 1080$ 的监控视频流可能需要高达4Mbps的带宽才能进行实时传输。另外,视频监控通常是 $7 \times 24$ 连续监控,如果要存储视频数据,这对存储系统是很大的挑战。因此,HD视频的高带宽和大存储的需求成为其在视频监控中大规模部署的主要制约因素。

### 发明内容

[0007] 本公开的实施方式提供了一种用于处理视频内容的方法。所述方法可以包括以下步骤:接收与图片相关联的色度块和亮度块;确定与亮度块相关联的亮度缩放信息;基于亮度缩放信息来确定色度缩放因子;以及使用色度缩放因子来处理色度块。

[0008] 本公开的实施方式提供了一种用于处理视频内容的装置。所述装置可以包括:存储器,该存储器存储有指令集;以及处理器,该处理器联接至存储器并被配置为执行指令集以使装置执行以下操作:接收与图片相关联的色度块和亮度块;确定与亮度块相关联的亮度缩放信息;基于亮度缩放信息来确定色度缩放因子;以及使用色度缩放因子来处理色度块。

[0009] 本公开的实施方式提供了一种非暂时性计算机可读存储介质,所述非暂时性计算机可读存储介质存储有指令集,该指令集能够由设备的一个或更多个处理器执行以使该设

备执行用于处理视频内容的方法。该方法包括以下步骤:接收与图片相关联的色度块和亮度块;确定与亮度块相关联的亮度缩放信息;基于亮度缩放信息来确定色度缩放因子;以及使用色度缩放因子来处理色度块。

[0010] 本公开的实施方式提供了一种计算机实现的视频解码方法,所述方法包括以下步骤:重构与图片相关联的多个亮度样本;以及重构与所述图片相关联的色度块,其中,重构色度块包括:确定所述色度块是否具有非零残差;以及响应于确定所述色度块具有一个或更多个非零色度残差,在重构所述色度块之前,确定多个重构的亮度样本的平均值,并基于所述平均值来对所述色度块的残差进行缩放。

[0011] 本公开的实施方式还提供了一种视频解码装置,所述视频解码装置包括:一个或更多个存储器,所述一个或更多个存储器存储有指令集;以及一个或更多个处理器,所述一个或更多个处理器被配置为执行所述指令集,以使所述装置执行以下操作:重构与图片相关联的多个亮度样本;以及重构与所述图片相关联的色度块,其中,在重构所述色度块时,所述一个或更多个处理器被配置为执行所述指令集,以使所述装置进一步执行以下操作:确定所述色度块是否具有非零残差;以及响应于确定所述色度块具有一个或更多个非零色度残差,在重构所述色度块之前,确定多个重构的亮度样本的平均值,并基于所述平均值来对所述色度块的残差进行缩放。

[0012] 本公开的实施方式还提供了一种非暂时性计算机可读存储介质,所述非暂时性计算机可读存储介质存储有与视频序列相关联的比特流,其中,所述比特流包括用于重构与图片相关联的色度块的编码信息,所述编码信息包括:指示所述色度块是否具有非零残差的编码块标志,其中,根据包括以下步骤的方法来处理所述比特流:重构与所述图片相关联的多个亮度样本;以及当所述编码块标志指示所述色度块具有一个或更多个非零色度残差时,在重构所述色度块之前,确定多个重构的亮度样本的平均值,并基于所述平均值来对所述色度块的残差进行缩放。

[0013] 本公开的实施方式还提供了一种用于处理视频内容的计算机实现的方法,所述方法包括以下步骤:接收与图片相关联的色度块,所述色度块包括多个色度样本;确定所述图片中的多个亮度样本的平均值;以及基于所述平均值来确定色度缩放因子,其中,所述色度缩放因子用于处理所述色度块的残差,其中,处理所述色度块的残差包括:确定所述色度块是否具有非零残差;以及响应于确定所述色度块具有非零残差,将所述色度缩放因子应用于所述色度块的所述残差。

[0014] 本公开的实施方式还提供了一种用于处理视频内容的装置,所述装置包括:存储器,所述存储器存储有指令集;以及处理器,所述处理器联接至所述存储器并被配置为执行所述指令集,以使所述装置执行以下操作:接收与图片相关联的色度块,所述色度块包括多个色度样本;确定所述图片中的多个亮度样本的平均值;基于所述平均值来确定色度缩放因子;以及通过将所述色度缩放因子应用于所述色度块的残差来处理所述色度块的所述残差,其中,在通过将所述色度缩放因子应用于所述色度块的残差来处理所述色度块的所述残差的过程中,所述处理器被配置为执行所述指令集,以使所述装置进一步执行以下操作:确定所述色度块是否具有非零残差;以及响应于确定所述色度块具有非零残差,将所述色度缩放因子应用于所述色度块的所述残差。

[0015] 本公开的实施方式还提供了一种非暂时性计算机可读存储介质,所述非暂时性计

计算机可读存储介质存储有指令集,所述指令集能够由设备的一个或多个处理器执行,以使所述设备执行用于处理视频内容的方法,所述方法包括以下步骤:接收与图片相关联的色度块,所述色度块包括多个色度样本;确定所述图片中的多个亮度样本的平均值;基于所述平均值来确定色度缩放因子;以及通过将所述色度缩放因子应用于所述色度块的残差来处理所述色度块的所述残差,其中,处理所述色度块的所述残差包括:确定所述色度块是否具有非零残差;以及响应于确定所述色度块具有非零残差,将所述色度缩放因子应用于所述色度块的所述残差。

## 附图说明

[0016] 以下具体实施方式和附图中例示了本公开的实施方式和各种方面。图中所示的各种特征未按比例绘制。

[0017] 图1例示了根据本公开的一些实施方式的示例视频序列的结构。

[0018] 图2A例示了根据本公开的一些实施方式的示例编码过程的示意图。

[0019] 图2B例示了根据本公开的一些实施方式的另一示例编码过程的示意图。

[0020] 图3A例示了根据本公开的一些实施方式的示例解码过程的示意图。

[0021] 图3B例示了根据本公开的一些实施方式的另一示例解码过程的示意图。

[0022] 图4例示了根据本公开的一些实施方式的用于对视频进行编码或解码的示例装置的框图。

[0023] 图5例示了根据本公开的一些实施方式的示例性的亮度映射与色度缩放(LMCS)过程的示意图。

[0024] 图6例示了根据本公开的一些实施方式的针对LMCS分段线性模型的图块(tile)组级别语法表。

[0025] 图7例示了根据本公开的一些实施方式的用于针对LMCS分段线性模型的另一图块组级别语法表。

[0026] 图8是根据本公开的一些实施方式的编码树单元语法结构的表。

[0027] 图9是根据本公开的一些实施方式的双树分区语法结构的表。

[0028] 图10例示了根据本公开的一些实施方式的简化亮度预测块的平均的示例。

[0029] 图11是根据本公开的一些实施方式的编码树单元语法结构的表。

[0030] 图12是根据本公开的一些实施方式的在图块组级别的用于LMCS分段线性模型的修改信令的语法要素的表。

[0031] 图13是根据本公开的一些实施方式的用于处理视频内容的方法的流程图。

[0032] 图14是根据本公开的一些实施方式的用于处理视频内容的方法的流程图。

[0033] 图15是根据本公开的一些实施方式的用于处理视频内容的另一方法的流程图。

[0034] 图16是根据本公开的一些实施方式的用于处理视频内容的另一方法的流程图。

## 具体实施方式

[0035] 现在将详细参考示例性实施方式,附图例示了所述示例性实施方式的示例。以下描述参考附图,除非另有表示,否则不同附图中的相同附图标记表示相同或相似的要素。在示例性实施方式的以下描述中阐述的实现方式不表示与本发明一致的所有实现方式。相



反,所述实现方式仅仅是与所附权利要求中所述的与本发明相关的方面一致的装置和方法的示例。除非另外特别说明,否则用语“或”涵盖所有可能的组合,除非不可行。例如,如果声明组成部分可以包括A或B,那么,除非另有明确说明或不可行,否则该组成部分可以包括A、或B、或A和B。作为第二示例,如果声明组成部分可以包括A、B或C,那么,除非另有明确说明或不可行,否则该组成部分可以包括A、或B、或C、或A和B、或A和C、或B和C、或A和B和C。

[0036] 视频是按照时间序列排列的静态图片(或“帧”)的集合,用于存储视觉信息。视频捕获设备(例如,摄像头)可以用于按照时间序列捕获和存储这些图片,并且视频播放设备(例如,电视、计算机、智能电话、平板计算机、视频播放器或具有显示功能的任何最终用户终端)可以用于按照时间序列显示这样的图片。此外,在一些应用中,视频捕获设备可以将所捕获的视频实时发送至视频播放设备(例如,具有监控器的电脑),诸如用于监控、会议、实况广播等。

[0037] 为了减少这样的应用所需的存储空间和传输带宽,可以在存储和传输之前对视频进行压缩,并在显示之前进行解压缩。压缩和解压缩可以通过由处理器(例如,通用计算机的处理器)或专用硬件执行的软件来实现。压缩模块一般称为“编码器”,并且解压模块一般称为“解码器”。编码器和解码器可以统称为“编解码器”。编码器和解码器可以实现为多种合适的硬件、软件或其组合中的任何一者。例如,编码器和解码器的硬件实现方式可以包括电路,诸如一个或多个微处理器、数字信号处理器(DSP)、专用集成电路(ASIC)、现场可编程门阵列(FPGA)、离散逻辑、或其任何组合。编码器和解码器的软件实现方式可以包括程序代码、计算机可执行指令、固件或固定在计算机可读介质中的任何合适的计算机实现的算法或过程。视频压缩和解压缩可以通过各种算法或标准(诸如MPEG-1、MPEG-2、MPEG-4、H.26x系列等)来实现。在一些应用中,编解码器可以根据第一编码标准解压缩视频,并使用第二编码标准重新压缩经解压缩的视频,在这种情况下,编解码器可以称为“转码器”。

[0038] 视频编码过程可以识别并保留可以用于重构图片的有用信息,而忽略针对重构而言不重要的信息。如果被忽略的、不重要的信息不能被完全重构,则这样的编码过程可以称为“有损”。否则,它可以称为“无损”。大多数编码过程都是有损的,这是减少所需存储空间和传输带宽的一种权衡。

[0039] 正被编码的图片(称为“当前图片”)的有用信息包括相对于参考图片(例如,先前编码和重构的图片)的变化。这样的变化可以包括像素的位置变化、亮度变化或颜色变化,其中位置变化是最受关注的。表示对象的一组像素的位置变化可以反映该对象在参考图片与当前图片之间的运动。

[0040] 未参考另一图片而编码的图片(即,它是它自己的参考图片)称为“I图片”。使用先前图片作为参考图片而编码的图片称为“P图片”。使用先前图片和未来图片两者作为参考图片(即,参考是“双向的”)而编码的图片称为“B图片”。

[0041] 如前所述,使用HD视频的视频监控面临着高带宽和大存储需求的挑战。为了解决这些挑战,可以降低编码视频的比特率。在I图片、P图片和B图片中,I图片的比特率最高。由于大多数监控视频的背景几乎是静态的,所以降低编码视频整体比特率的一种方法可以是使用更少的I图片进行视频编码。

[0042] 然而,使用更少I图片的改进可能是微不足道的,因为I图片通常在编码视频中不占主导地位。例如,在典型的视频比特流中,I图片、B图片和P图片的比例可以是1:20:9,其

中I图片可以占总比特率的10%以下。换句话说,在这样的示例中,即使去除了所有的I图片,降低的比特率也不会超过10%。

[0043] 本公开提供了用于基于特性的视频处理以进行视频监控的方法、装置和系统。本文的“特性”是指与图片中的视频内容相关联的内容特性、与对图片进行编码或解码的运动估计相关联的运动特性,或者两者。例如,内容特性可以是视频序列的一个或多个连续图片中的像素,所述像素与图片中的对象、场景或环境事件中的至少一者相关。又例如,运动特性可以包括与视频编码过程相关的信息,其示例将在后面详述。

[0044] 在本公开中,在对视频序列的图片进行编码时,可以使用特性分类器对视频序列的图片的一个或多个特性进行检测和分类。不同类的特性可以与不同的优先级级别相关联,所述优先级级别进一步与不同的编码比特率相关联。不同优先级级别可以与不同编码参数集相关联,这会导致不同编码质量级别。优先级级别越高,其关联参数集可以产生的视频质量就越高。通过这种基于特性的视频处理,可以大大降低监控视频的比特率,而不会造成明显的信息丢失。另外,本公开的实施方式可以针对不同应用场景(例如,安全、交通、环境监控等)定制优先级级别与参数集之间的对应关系,从而大大提高视频编码质量,并大大降低带宽和存储的成本。

[0045] 图1例示了根据本公开的一些实施方式的示例视频序列100的结构。视频序列100可以是实况视频或已被捕获和存档的视频。视频100可以是现实视频、计算机生成视频(例如,计算机游戏视频)或其组合(例如,具有增强现实效果的现实视频)。可以从视频捕获设备(例如,摄像头)、包含先前捕获的视频的视频档案(例如,存储在存储设备中的视频文件)或从视频内容提供商接收视频的视频馈送接口(例如,视频广播收发器)输入视频序列100。

[0046] 如图1所示,视频序列100可以包括沿着时间线在时间上排列的一系列图片,包括图片102、图片104、图片106和图片108。图片102至图片106是连续的,并且在图片106与图片108之间存在更多的图片。在图1中,图片102是I图片,其参考图片是图片102本身。图片104是P图片,其参考图片是图片102,如箭头所示。图片106是B图片,其参考图片是图片104和图片108,如箭头所示。在一些实施方式中,图片(例如,图片104)的参考图片可以不紧接在该图片之前或之后。例如,图片104的参考图片可以是图片102之前的图片。应注意,图片102至图片106的参考图片仅是示例,并且本公开不将参考图片的实施方式限制为图1所示的示例。

[0047] 通常,由于这些任务的计算复杂性,所以视频编解码器不会一次对整个图片进行编码或解码。相反,它们可以将图片划分成基本区段(segment),并逐区段地对图片进行编码或解码。这种基本区段在本公开中称为基本处理单元(“BPU”)。例如,图1的结构110示出了视频序列100的图片(例如,图片102至图片108中的任何一者)的示例结构。在结构110中,图片被划分成 $4 \times 4$ 个基本处理单元,其边界显示为虚线。在一些实施方式中,基本处理单元在一些视频编码标准(例如,MPEG系列、H.261、H.263或H.264/AVC)中可以称为“宏块”,或在一些其它视频编码标准(例如,H.265/HEVC或H.266/VVC)中称为“编码树单元”(“CTU”)。在图片中,基本处理单元可以具有可变大小,诸如 $128 \times 128$ 、 $64 \times 64$ 、 $32 \times 32$ 、 $16 \times 16$ 、 $4 \times 8$ 、 $16 \times 32$ 或者像素的任何任意形状和大小。可以基于编码效率与要在基本处理单元中保持的细节级别的平衡来针对图片选择基本处理单元的大小和形状。

[0048] 基本处理单元可以是逻辑单元,其可以包括被存储在计算机存储器中(例如,被存

储在视频帧缓冲部中)的一组不同类型的视频数据。例如,彩色图片的基本处理单元可以包括表示非彩色亮度信息的亮度分量(Y)、表示颜色信息的一个或多个色度分量(例如,Cb和Cr)以及相关语法要素,其中亮度分量和色度分量可以具有与基本处理单元相同的大小。在一些视频编码标准(例如,H.265/HEVC或H.266/VVC)中,亮度分量和色度分量可以称为“编码树块”(“CTB”)。对基本处理单元执行的任何操作都可以对其亮度分量和色度分量中的每一者重复执行。

[0049] 视频编码具有多个操作阶段(stage),其示例将在图2A至图2B以及图3A至图3B中详述。针对各个阶段,基本处理单元的大小仍然可能太大而无法处理,因此可以进一步划分成在本公开中称为“基本处理子单元”的区段。在一些实施方式中,基本处理子单元在一些视频编码标准(例如,MPEG系列、H.261、H.263或H.264/AVC)中可以称为“块”,或在一些其它视频编码标准(例如,H.265/HEVC或H.266/VVC)中称为“编码单元”(“CU”)。基本处理子单元可以具有与基本处理单元相同或更小的大小。与基本处理单元类似,基本处理子单元也是逻辑单元,其可以包括被存储在计算机存储器中(例如,被存储在视频帧缓冲部中)的一组不同类型的视频数据(例如,Y、Cb、Cr以及相关语法要素)。对基本处理子单元执行的任何操作都可以对其亮度分量和色度分量中的每一者重复执行。应注意,这种划分可以根据处理需要进行到更深级别。还应注意,不同阶段可以使用不同方案来划分基本处理单元。

[0050] 例如,在模式决定阶段(其示例将在图2B中详述),编码器可以决定将什么预测模式(例如,图片内预测或图片间预测)用于基本处理单元,该基本处理单元可能太大而无法做出这样的决定。编码器可以将基本处理单元拆分成多个基本处理子单元(例如,H.265/HEVC或H.266/VVC中的CU),并针对各个单独的基本处理子单元决定预测类型。

[0051] 又例如,在预测阶段(其示例将在图2A中详述),编码器可以在基本处理子单元(例如,CU)的级别执行预测操作。然而,在一些情况下,基本处理子单元仍然可能太大而无法处理。编码器可以进一步将基本处理子单元拆分成更小的区段(例如,在H.265/HEVC或H.266/VVC中称为“预测块”或“PB”),可以在该更小的区段的级别执行预测操作。

[0052] 又例如,在变换阶段(其示例将在图2A中详述),编码器可以针对残差基本处理子单元(例如,CU)执行变换操作。然而,在一些情况下,基本处理子单元仍然可能太大而无法处理。编码器可以进一步将基本处理子单元拆分成更小的区段(例如,在H.265/HEVC或H.266/VVC中称为“变换块”或“TB”),可以在该更小的区段的级别执行变换操作。应注意,同一基本处理子单元的划分方案在预测阶段和变换阶段可以是不同的。例如,在H.265/HEVC或H.266/VVC中,同一CU的预测块和变换块可以具有不同大小和数量。

[0053] 在图1的结构110中,基本处理单元112进一步划分为 $3 \times 3$ 个基本处理子单元,其边界用虚线示出。同一图片的不同基本处理单元可以按照不同方案划分成基本处理子单元。

[0054] 在一些实现方式中,为了向视频编码和解码提供并行处理和容错能力,可以将图片划分成多个区域以进行处理,使得针对图片的一区域,编码或解码过程可以不依赖于该图片的任何其它区域的信息。换句话说,可以独立处理图片的各个区域。通过这样做,编解码器可以并行处理图片的不同区域,从而提高编码效率。此外,当区域的数据在处理中损坏或网络传输中丢失时,编解码器可以对同一图片的其它区域进行正确编码或解码,而不依赖于损坏或丢失的数据,从而提供了容错能力。在一些视频编码标准中,图片可以被划分成不同类型的区域。例如,H.265/HEVC和H.266/VVC提供两种类型的区域:“切片(slice)”和

“图块”。还应注意,视频序列100的不同图片可以具有用于将图片划分成多个区域的不同分区方案。

[0055] 例如,在图1中,结构110被划分成三个区域114、116和118,其边界在结构110内部示出为实线。区域114包括四个基本处理单元。区域116和118中的每一者包括六个基本处理单元。应注意,图1中的结构110的基本处理单元、基本处理子单元和区域仅是示例,并且本公开不限制其实施方式。

[0056] 图2A例示了根据本公开的一些实施方式的示例编码过程200A的示意图。编码器可以根据过程200A将视频序列202编码成视频比特流228。类似于图1中的视频序列100,视频序列202可以包括按照时间顺序排列的图片(称为“原始图片”)的集合。类似于图1中的结构110,视频序列202的各个原始图片可以由编码器划分成基本处理单元、基本处理子单元或区域,以进行处理。在一些实施方式中,编码器可以针对视频序列202的各个原始图片在基本处理单元级别执行过程200A。例如,编码器可以以迭代方式执行过程200A,其中编码器可以在过程200A的一次迭代中对基本处理单元进行编码。在一些实施方式中,编码器可以针对视频序列202的各个原始图片的区域(例如,区域114至118)并行地执行过程200A。

[0057] 在图2A中,编码器可以将视频序列202的原始图片的基本处理单元(称为“原始BPU”)馈送至预测阶段204,以生成预测数据206和预测BPU 208。编码器可以从原始BPU减去预测BPU 208,以生成残差BPU 210。编码器可以将残差BPU 210馈送至变换阶段212和量化阶段214,以生成量化变换系数216。编码器可以将预测数据206和量化变换系数216馈送至二进制编码阶段226,以生成视频比特流228。组成部分202、204、206、208、210、212、214、216、226和228可以称为“前向路径”。在过程200A期间,在量化阶段214之后,编码器可以将量化变换系数216馈送至逆量化阶段218和逆变换阶段220,以生成重构残差BPU 222。编码器可以将重构残差BPU 222与预测BPU 208相加,以生成预测参考224,该预测参考224在预测阶段204中用于过程200A的下一迭代。过程200A的组成部分218、220、222和224可以称为“重构路径”。重构路径可以用于确保编码器和解码器使用相同的参考数据进行预测。

[0058] 编码器可以迭代地执行过程200A,以(在前向路径中)对原始图片的各个原始BPU进行编码并且(在重构路径中)生成用于对原始图片的下一原始BPU进行编码的预测参考224。在对原始图片的所有原始BPU进行编码之后,编码器可以继续对视频序列202中的下一图片进行编码。

[0059] 参考过程200A,编码器可以接收由视频捕获设备(例如,摄像头)生成的视频序列202。本文使用的术语“接收”可以指接收、输入、获取、检索、获得、读取、访问或针对输入数据采用的任何方式的任何动作。

[0060] 在预测阶段204,在当前迭代,编码器可以接收原始BPU和预测参考224,并执行预测操作,以生成预测数据206和预测BPU 208。预测参考224可以是根据过程200A的先前迭代的重构路径生成的。预测阶段204的目的是通过提取预测数据206来减少信息冗余,该预测数据206可以用于根据预测数据206和预测参考224将原始BPU重构为预测BPU 208。

[0061] 理想情况下,预测BPU 208可以与原始BPU相同。然而,由于非理想的预测和重构操作,预测BPU 208通常与原始BPU略有不同。为了记录这样的差异,在生成预测BPU 208之后,编码器可以从原始BPU中减去该预测BPU,以生成残差BPU 210。例如,编码器可以从原始BPU的对应像素的值减去预测BPU 208的像素的值(例如,灰度值或RGB值)。残差BPU 210的各个

像素可以具有残差值,该残差值是原始BPU与预测BPU 208的对应像素之间的这种相减的结果。与原始BPU相比,预测数据206和残差BPU 210可以具有更少的位,但是它们可以用于重构原始BPU,而不会显著降低质量。因此,原始BPU被压缩。

[0062] 为了进一步压缩残差BPU 210,在变换阶段212,编码器可以通过将残差BPU 210分解成二维“基本模式”的集合来减少其空间冗余,各个基本模式与“变换系数”相关联。基本模式可以具有相同的大小(例如,残差BPU 210的大小)。各个基本模式可以表示残差BPU 210的变化频率(例如,亮度变化频率)分量。不能根据任何其它基本模式的任何组合(例如,线性组合)再现基本模式。换句话说,分解可以将残差BPU 210的变化分解到频域中。这种分解类似于函数的离散傅里叶变换,其中基本模式类似于离散傅里叶变换的基本函数(例如,三角函数),并且变换系数类似于与基本函数相关联的系数。

[0063] 不同的变换算法可以使用不同的基本模式。可以在变换阶段212使用各种变换算法,例如,诸如离散余弦变换、离散正弦变换等。变换阶段212处的变换是可逆的。即,编码器可以通过变换的逆运算(称为“逆变换”)来恢复残差BPU 210。例如,为了恢复残差BPU 210的像素,逆变换可以是将基本模式的对应像素的值乘以相应关联系数并将乘积相加以产生加权和。针对视频编码标准,编码器和解码器都可以使用相同的变换算法(因此使用同一基本模式)。因此,编码器可以仅记录变换系数,解码器可以根据所述变换系数重构残差BPU 210,而无需从编码器接收基本模式。与残差BPU 210相比,变换系数可以具有更少的位,但是它们可以用于重构残差BPU 210而不会显著降低质量。因此,残差BPU 210被进一步压缩。

[0064] 编码器可以在量化阶段214进一步压缩变换系数。在变换过程中,不同基本模式可以表示不同变化频率(例如,亮度变化频率)。由于人眼通常更擅长识别低频变化,因此编码器可以忽略高频变化信息,而不会导致解码质量显著下降。例如,在量化阶段214,编码器可以通过将各个变换系数除以整数值(称为“量化参数”)并将商四舍五入为其最接近的整数来生成量化变换系数216。在这样的操作之后,高频基本模式的一些变换系数可以被转换为零,而低频基本模式的变换系数可以被转换为更小的整数。编码器可以忽略零值量化变换系数216,由此进一步压缩变换系数。量化过程也是可逆的,其中量化变换系数216可以在量化的逆运算(称为“逆量化”)中重构为变换系数。

[0065] 因为编码器在四舍五入运算中忽略这些除法的余数,所以量化阶段214可能是有损的。通常,量化阶段214可以在过程200A中贡献最多的信息损失。信息损失越大,量化变换系数216可能需要的位越少。为了获得不同级别的信息损失,编码器可以使用量化参数或量化过程的任何其它参数的不同值。

[0066] 在二进制编码阶段226,编码器可以使用二进制编码技术(例如,诸如熵编码、可变长度编码、算术编码、霍夫曼编码、上下文自适应二进制算术编码或任何其它无损或有损压缩算法)对预测数据206和量化变换系数216进行编码。在一些实施方式中,除了预测数据206和量化变换系数216之外,编码器还可以在二进制编码阶段226对其它信息(例如,诸如在预测阶段204使用的预测模式、预测操作的参数、在变换阶段212的变换类型、量化过程的参数(例如,量化参数)、编码器控制参数(例如,比特率控制参数)等)进行编码。编码器可以使用二进制编码阶段226的输出数据来生成视频比特流228。在一些实施方式中,视频比特流228可以被进一步打包,以用于网络传输。

[0067] 参考过程200A的重构路径,在逆量化阶段218,编码器可以对量化变换系数216执

行逆量化,以生成重构变换系数。在逆变换阶段220,编码器可以基于重构变换系数来生成重构残差BPU 222。编码器可以将重构残差BPU 222添加至预测BPU 208,以生成将在过程200A的下一代中使用的预测参考224。

[0068] 应注意,过程200A的其它变型可以用于对视频序列202进行编码。在一些实施方式中,过程200A的阶段可以由编码器按照不同顺序执行。在一些实施方式中,过程200A的一个或更多个阶段可以组合成单个阶段。在一些实施方式中,过程200A的单个阶段可以划分成多个阶段。例如,变换阶段212和量化阶段214可以组合成单个阶段。在一些实施方式中,过程200A可以包括附加阶段。在一些实施方式中,过程200A可以省略图2A中的一个或更多个阶段。

[0069] 图2B例示了根据本公开的一些实施方式的另一示例编码过程200B的示意图。过程200B可以是根据过程200A修改的。例如,过程200B可以由符合混合视频编码标准(例如,H.26x系列)的编码器使用。与过程200A相比,过程200B的前向路径附加地包括模式决定阶段230并将预测阶段204划分成空间预测阶段2042和时间预测阶段2044。过程200B的重构路径附加地包括环路滤波器阶段232和缓冲部234。

[0070] 通常,预测技术可以分为两种类型:空间预测和时间预测。空间预测(例如,图片内预测或“帧内预测(intra prediction)”)可以使用来自同一图片中一个或更多个已经编码的相邻BPU的像素来预测当前BPU。即,空间预测中的预测参考224可以包括相邻BPU。空间预测可以减少图片的固有空间冗余。时间预测(例如,图片间预测或“帧间预测(inter prediction)”)可以使用来自一个或更多个已经编码的图片的区域来预测当前BPU。即,时间预测中的预测参考224可以包括编码图片。时间预测可以减少图片的固有时间冗余。

[0071] 参考过程200B,在前向路径中,编码器在空间预测阶段2042和时间预测阶段2044执行预测操作。例如,在空间预测阶段2042,编码器可以执行帧内预测。针对正在被编码的图片的原始BPU,预测参考224可以包括同一图片中的(在前向路径中)已经编码和(在重构路径中)重构的一个或更多个相邻BPU。编码器可以通过外推相邻BPU来生成预测BPU 208。例如,外推技术可以包括线性外推或内插、多项式外推或内插等。在一些实施方式中,编码器可以在像素级别执行外推,诸如通过外推预测BPU 208的各个像素的对应像素的值。用于外推的相邻BPU可以从各个方向相对于原始BPU定位,诸如在垂直方向(例如,在原始BPU的顶部)、水平方向(例如,到原始BPU的左侧)、对角线方向(例如,到原始BPU的左下侧、右下侧、左上侧或右上侧)或所使用的视频编码标准中限定的任何方向。针对帧内预测,预测数据206可以包括例如所使用的相邻BPU的位置(例如,坐标)、所使用的相邻BPU的大小、外推的参数、所使用的相邻BPU相对于原始BPU的方向等。

[0072] 又例如,在时间预测阶段2044,编码器可以执行帧间预测。针对当前图片的原始BPU,预测参考224可以包括(在前向路径中)已经编码和(在重构路径中)重构的一个或更多个图片(称为“参考图片”)。在一些实施方式中,参考图片可以由BPU编码和重构BPU。例如,编码器可以将重构残差BPU 222与预测BPU 208相加,以生成重构BPU。当生成同一图片的所有重构BPU时,编码器可以生成重构图片作为参考图片。编码器可以执行“运动估计”的操作,以在参考图片的范围(称为“搜索窗口”)中搜索匹配区域。搜索窗口在参考图片中的位置可以是基于原始BPU在当前图片中的位置确定的。例如,搜索窗口可以在如下位置居中,该位置在参考图片中的坐标与原始BPU在当前图片中的坐标相同,并且搜索窗口可以向外

延伸预定距离。当编码器在搜索窗口中(例如,通过使用像素递归算法、块匹配算法等)识别出与原始BPU相似的区域时,编码器可以将这样的区域确定为匹配区域。匹配区域可以具有与原始BPU不同的尺寸(例如,小于、等于、大于或处于不同形状)。因为参考图片和当前图片在时间线上是时间分离的(例如,如图1所示),所以可以认为匹配区域随着时间推移“移动”至原始BPU的位置。编码器可以将这种运动的方向和距离记录为“运动矢量”。当使用多个参考图片时(例如,如图1中的图片106),编码器可以针对各个参考图片搜索匹配区域并确定其关联运动矢量。在一些实施方式中,编码器可以向相应匹配参考图片的匹配区域的像素值分配权重。

[0073] 运动估计可以用于识别各种类型的运动,例如,诸如平移、旋转、缩放等。针对帧间预测,预测数据206可以包括例如匹配区域的位置(例如,坐标)、与匹配区域相关联的运动矢量、参考图片的数量、与参考图片相关联的权重等。

[0074] 为了生成预测BPU 208,编码器可以执行“运动补偿”操作。运动补偿可以用于基于预测数据206(例如,运动矢量)和预测参考224来重构预测BPU 208。例如,编码器可以根据运动矢量移动参考图片的匹配区域,其中,编码器可以预测当前图片的原始BPU。当使用多个参考图片时(例如,如图1中的图片106),编码器可以根据匹配区域的相应运动矢量和平均像素值移动参考图片的匹配区域。在一些实施方式中,如果编码器已向相应匹配参考图片的匹配区域的像素值分配了权重,则编码器可以将所移动的匹配区域的像素值的加权和相加。

[0075] 在一些实施方式中,帧间预测可以是单向的或双向的。单向帧间预测可以使用相对于当前图片在同一时间方向上的一个或更多个参考图片。例如,图1中的图片104是单向帧间预测图片,其中参考图片(即,图片102)在图片104之前。双向帧间预测可以使用相对于当前图片在两个时间方向上的一个或更多个参考图片。例如,图1中的图片106是双向帧间预测图片,其中参考图片(即,图片104和图片108)相对于图片104在两个时间方向上。

[0076] 仍然参考过程200B的前向路径,在空间预测2042和时间预测阶段2044之后,在模式决定阶段230,编码器可以选择针对过程200B的当前迭代的预测模式(例如,帧内预测或帧间预测中的一者)。例如,编码器可以执行率失真优化技术,其中编码器可以根据候选预测模式的比特率和重构参考图片在候选预测模式下的失真来选择预测模式,以最小化成本函数的值。根据所选择的预测模式,编码器可以生成对应预测BPU 208和预测数据206。

[0077] 在过程200B的重构路径中,如果在前向路径中选择了帧内预测模式,则在生成预测参考224(例如,当前图片中已经编码和重构的当前BPU)之后,编码器可以将预测参考224直接馈送至空间预测阶段2042,以供以后使用(例如,用于当前图片的下一BPU的外推)。如果在前向路径中选择了帧间预测模式,则在生成预测参考224(例如,所有BPU都已被编码和重构的当前图片)之后,编码器可以将预测参考224馈送至环路滤波器阶段232,在环路滤波器阶段232,编码器可以将环路滤波器应用于预测参考224,以减少或消除由帧间预测引入的失真(例如,块效应)。编码器可以在环路滤波器阶段232应用各种环路滤波器技术,例如,诸如去块、样本自适应偏移、自适应环路滤波器等。环路滤波参考图片可以存储在缓冲部234(或“解码图片缓冲部”)中,以供以后使用(例如,将用作视频序列202的未来图片的帧间预测参考图片)。编码器可以在缓冲部234中存储一个或更多个参考图片,以在时间预测阶段2044使用。在一些实施方式中,编码器可以在二进制编码阶段226对环路滤波器的参数



(例如,环路滤波器强度)以及量化变换系数216、预测数据206和其它信息进行编码。

[0078] 图3A例示了根据本公开的一些实施方式的示例解码过程300A的示意图。过程300A可以是对应于图2A中的压缩过程200A的解压过程。在一些实施方式中,过程300A可以类似于过程200A的重构路径。解码器可以根据过程300A将视频比特流228解码成视频流304。视频流304可以与视频序列202非常相似。然而,由于压缩过程和解压缩过程(例如,图2A至图2B中的量化阶段214)中的信息丢失,通常,视频流304与视频序列202不同。类似于图2A中的过程200A和图2B中的过程200B,解码器可以在基本处理单元(BPU)级别针对被编码在视频比特流228中的各个图片执行过程300A。例如,解码器可以以迭代方式执行过程300A,其中解码器可以在过程300A的一次迭代中对基本处理单元进行解码。在一些实施方式中,解码器可以针对被编码在视频比特流228中的各个图片的区域(例如,区域114至区域118)并行地执行过程300A。

[0079] 在图3A中,解码器可以将与编码图片的基本处理单元(称为“编码BPU”)相关联的视频比特流228的一部分馈送至二进制解码阶段302。在二进制解码阶段302,解码器可以将该部分解码成预测数据206和量化变换系数216。解码器可以将量化变换系数216馈送至逆量化阶段218和逆变换阶段220,以生成重构残差BPU 222。解码器可以将预测数据206馈送至预测阶段204,以生成预测BPU 208。解码器可以将重构残差BPU 222与预测BPU 208相加,以生成预测参考224。在一些实施方式中,预测参考224可以存储在缓冲部(例如,计算机存储器中的解码图片缓冲部)中。解码器可以将预测参考224馈送至预测阶段204,以在过程300A的下一迭代中执行预测操作。

[0080] 解码器可以迭代地执行过程300A,以对编码图片的各个编码BPU进行解码并生成预测参考224,以用于对编码图片的下一编码BPU进行编码。在对编码图片的所有编码BPU进行解码后,解码器可以将图片输出至视频流304,以供显示,并继续对视频比特流228中的下一编码图片进行解码。

[0081] 在二进制解码阶段302,解码器可以执行由编码器使用的二进制编码技术(例如,熵编码、可变长度编码、算术编码、霍夫曼编码、上下文自适应二进制算术编码或任何其它无损压缩算法)的逆操作。在一些实施方式中,除了预测数据206和量化变换系数216之外,解码器还可以在二进制解码阶段302对其它信息(例如,诸如预测模式、预测操作的参数、变换类型、量化过程的参数(例如,量化参数)、编码器控制参数(例如,比特率控制参数)等)进行解码。在一些实施方式中,如果视频比特流228以包的形式通过网络传输,则解码器可以在将视频比特流228馈送至二进制解码阶段302之前对其进行解包。

[0082] 图3B例示了根据本公开的一些实施方式的另一示例解码过程300B的示意图。过程300B可以是根据过程300A修改的。例如,过程300B可以由符合混合视频编码标准(例如,H.26x系列)的解码器使用。与过程300A相比,过程300B附加地将预测阶段204划分成空间预测阶段2042和时间预测阶段2044,并且附加地包括环路滤波器阶段232和缓冲部234。

[0083] 在过程300B中,针对正在被解码的编码图片(称为“当前图片”)的编码基本处理单元(称为“当前BPU”),由解码器从二进制解码阶段302解码的预测数据206可以包括各种类型的数据,这取决于编码器使用什么预测模式对当前BPU进行编码。例如,如果编码器使用帧内预测来对当前BPU进行编码,则预测数据206可以包括指示帧内预测的预测模式指示符(例如,标志值)、帧内预测操作的参数等。例如,帧内预测操作的参数可以包括用作参考的



一个或多个相邻BPU的位置(例如,坐标)、相邻BPU的大小、外推参数、相邻BPU相对于原始BPU的方向等。又例如,如果编码器使用帧间预测对当前BPU进行编码,则预测数据206可以包括指示帧间预测的预测模式指示符(例如,标志值)、帧间预测操作的参数等。例如,帧间预测操作的参数可以包括与当前BPU相关联的参考图片的数量、分别与参考图片相关联的权重、相应参考图片中一个或多个匹配区域的位置(例如,坐标)、分别与匹配区域相关联的一个或多个运动矢量等。

[0084] 基于预测模式指示符,解码器可以决定是在空间预测阶段2042执行空间预测(例如,帧内预测)还是在时间预测阶段2044执行时间预测(例如,帧间预测)。图2B描述了执行这种空间预测或时间预测的细节,并且下文将不再赘述。在执行这样的空间预测或时间预测之后,解码器可以生成预测BPU 208。解码器可以将预测BPU 208和重构残差BPU 222相加,以生成预测参考224,如图3A中所描述的。

[0085] 在过程300B中,解码器可以将预测参考224馈送至空间预测阶段2042或时间预测阶段2044,以在过程300B的下一代中执行预测操作。例如,如果在空间预测阶段2042使用帧内预测对当前BPU进行解码,则在生成预测参考224(例如,经解码的当前BPU)之后,解码器可以将预测参考224直接馈送至空间预测阶段2042,以供以后使用(例如,用于外推当前图片的下一BPU)。如果在时间预测阶段2044使用帧间预测对当前BPU进行解码,则在生成预测参考224(例如,所有BPU已被解码的参考图片)之后,编码器可以将预测参考224馈送至环路滤波器阶段232,以减少或消除失真(例如,块效应)。解码器可以以图2B中描述的方式将环路滤波器应用于预测参考224。经环路滤波的参考图片可以存储在缓冲部234(例如,计算机存储器中的解码图片缓冲部)中,以供以后使用(例如,将用作视频比特流228的未来编码图片的帧间预测参考图片)。解码器可以将一个或多个参考图片存储在缓冲部234中,以在时间预测阶段2044使用。在一些实施方式中,当预测数据206的预测模式指示符指示使用帧间预测来对当前BPU进行编码时,预测数据可以进一步包括环路滤波器的参数(例如,环路滤波器强度)。

[0086] 图4是根据本公开的一些实施方式的用于对视频进行编码或解码的示例装置400的框图。如图4所示,装置400可以包括处理器402。当处理器402执行本文描述的指令时,装置400可以变成用于视频编码或解码的专用机器。处理器402可以是能够操纵或处理信息的任何类型的电路。例如,处理器402可以包括任何数量的中央处理单元(或“CPU”)、图形处理单元(或“GPU”)、神经处理单元(“NPU”)、微控制器单元(“MCU”)、光学处理器、可编程逻辑控制器、微控制器、微处理器、数字信号处理器、知识产权(IP)核、可编程逻辑阵列(PLA)、可编程阵列逻辑(PAL)、通用阵列逻辑(GAL)、复杂可编程逻辑器件(CPLD)、现场可编程门阵列(FPGA)、片上系统(SoC)、专用集成电路(ASIC)等的任何组合。在一些实施方式中,处理器402还可以是被分组为单个逻辑部件的处理器集合。例如,如图4所示,处理器402可以包括多个处理器,所述多个处理器包括处理器402a、处理器402b和处理器402n。

[0087] 装置400还可以包括被配置为存储数据(例如,指令集、计算机代码、中间数据等)的存储器404。例如,如图4所示,存储数据可以包括程序指令(例如,用于实现过程200A、过程200B、过程300A或过程300B中的阶段的程序指令)和供处理的数据(例如,视频序列202、视频比特流228或视频流304)。处理器402可以(例如,经由总线410)访问程序指令和供处理的数据,并且执行程序指令以对供处理的数据执行操作或操纵。存储器404可以包括高速随

机存取存储设备或非易失性存储设备。在一些实施方式中,存储器404可以包括任何数量的随机存取存储器(RAM)、只读存储器(ROM)、光盘、磁盘、硬盘驱动器、固态驱动器、闪存驱动器、安全数字(SD)卡、记忆棒、紧凑型闪存(CF)卡等的任何组合。存储器404也可以是被分组为单个逻辑部件的一组存储器(图4中未示出)。

[0088] 总线410可以是在装置400内部的部件之间传输数据的通信设备,诸如内部总线(例如,CPU-存储器总线)、外部总线(例如,通用串行总线端口、外围部件互连快速端口)等。

[0089] 为了便于解释而不引起歧义,在本公开中,处理器402和其它数据处理电路统称为“数据处理电路”。数据处理电路可以完全实现为硬件,或实现为软件、硬件或固件的组合。另外,数据处理电路可以是单个独立模块或者可以完全或部分地组合到装置400的任何其它部件中。

[0090] 装置400可以进一步包括网络接口406,以提供与网络(例如,互联网、内联网、局域网、移动通信网络等)的有线通信或无线通信。在一些实施方式中,网络接口406可以包括任何数量的网络接口控制器(NIC)、射频(RF)模块、转发器、收发器、调制解调器、路由器、网关、有线网络适配器、无线网络适配器、蓝牙适配器、红外适配器、近场通信(“NFC”)适配器、蜂窝网络芯片等的任何组合。

[0091] 在一些实施方式中,可选地,装置400可以进一步包括外围接口408,以提供到一个或多个外围设备的连接。如图4所示,外围设备可以包括但不限于光标控制设备(例如,鼠标、触摸板或触摸屏)、键盘、显示器(例如,阴极射线管显示器、液晶显示器或发光二极管显示器)、视频输入设备(例如,摄像头或联接至视频档案的输入接口)等。

[0092] 应注意,视频编解码器(例如,执行过程200A、过程200B、过程300A或过程300B的编解码器)可以实现为装置400中任何软件模块或硬件模块的任何组合。例如,过程200A、过程200B、过程300A或过程300B的一些或所有阶段可以实现为装置400的一个或多个软件模块,诸如可以加载到存储器404中的程序指令。又例如,过程200A、过程200B、过程300A或过程300B的一些或所有阶段可以实现为装置400的一个或多个硬件模块,诸如专用数据处理电路(例如,FPGA、ASIC、NPU等)。

[0093] 图5例示了根据本公开的一些实施方式的示例性的亮度映射与色度缩放(LMCS)过程500的示意图。例如,过程500可以由符合混合视频编码标准(例如,H.26x系列)的解码器使用。LMCS是在图2B的环路滤波器232之前应用的新的处理块。LMCS也可以称为整形器。

[0094] LMCS过程500可以包括基于自适应分段线性模型的亮度分量值的环内映射(in-looping mapping)和色度分量的依赖于亮度的色度残差缩放。

[0095] 如图5所示,基于自适应分段线性模型的亮度分量值的环内映射可以包括前向映射阶段518和逆映射阶段508。色度分量的依赖于亮度的色度残差缩放可以包括色度缩放520。

[0096] 映射前或逆映射后的样本值可以称为原始域中的样本,并且映射后且逆映射前的样本值可以称为映射域中的样本。当启用LMCS时,可以在映射域而不是原始域中执行过程500中的一些阶段。可以理解,可以使用SPS标志在序列级别启用/禁用前向映射阶段518和逆映射阶段508。

[0097] 如图5所示,在映射域中执行 $Q^{-1}$ 与 $T^{-1}$ 阶段504、重构506和帧内预测508。例如, $Q^{-1}$ 与 $T^{-1}$ 阶段504可以包括逆量化和逆变换,重构506可以包括亮度预测和亮度残差的相加,并且帧

内预测508可以包括亮度帧内预测。

[0098] 在原始(即,非映射)域中执行环路滤波器510、运动补偿阶段516和530、帧内预测阶段528、重构阶段522以及解码图片缓冲(DPB) 512和526。在一些实施方式中,环路滤波器510可以包括去块、自适应环路滤波器(ALF)和样本自适应偏移(SAO),重构阶段522可以包括色度预测与色度残差的相加,并且DPB 512和526可以存储解码图片作为参考图片。

[0099] 在一些实施方式中,可以应用利用分段线性模型的亮度映射。

[0100] 亮度分量的环内映射可以通过跨动态范围重新分布码字来调整输入视频的信号统计,以提高压缩效率。亮度映射可以由前向映射函数“FwdMap”和对应逆映射函数“InvMap”执行。使用具有16个相等片段(piece)的分段线性模型来传输“FwdMap”函数。“InvMap”函数不需要传输,而是从“FwdMap”函数获得的。

[0101] 图6的表1和图7的表2示出了分段线性模型的信令。图6的表1例示了图块组报头语法结构。如图6所示,传输整形器模型参数存在标志,以指示当前图块组中是否存在亮度映射模型。如果当前图块组中存在亮度映射模型,则可以使用图7的表2中所示的语法要素在tile\_group\_reshaper\_model()中传输对应分段线性模型参数。分段线性模型将输入信号的动态范围分成16个相等片段。针对16个相等片段中的每一者,所述片段的线性映射参数是使用分配给该片段的码字数量表示的。以10位输入为例。默认情况下,16个片段中的每一者可以具有被分配给该片段的64个码字。传输的码字的数量可以用于计算缩放因子并相应地针对该片段调整映射函数。图7的表2还限定了传输码字数量的最小索引“reshaper\_model\_min\_bin\_idx”和最大索引“reshaper\_model\_max\_bin\_idx”(包括这两者)。如果片段索引小于reshaper\_model\_min\_bin\_idx或大于reshaper\_model\_max\_bin\_idx,则不会传输该片段的码字数量,并且所述码字数量被推断为零(即,未分配码字并且未向该片段应用映射/缩放)。

[0102] 在传输tile\_group\_reshaper\_model()之后,在图块组标头级别,传输另一整形器启用标志“tile\_group\_reshaper\_enable\_flag”,以指示是否向当前图块组应用了如图8所示的LMCS过程。如果针对当前图块组启用了整形器并且如果当前图块组未使用双树分区,则传输另外的色度缩放启用标志,以指示是否针对当前图块组启用了色度缩放。双树分区也可以称为色度分离树。

[0103] 可以基于如下的图7的表2中的传输的语法要素来构建分段线性模型。“FwdMap”分段线性模型的各个第i片段( $i=0,1,\dots,15$ )由两个输入枢轴点InputPivot[]和两个输出(映射)枢轴点MappedPivot[]限定。InputPivot[]和MappedPivot[]是基于如下的传输的语法计算的(不失一般性,我们假定输入视频的位深度为10位):

[0104] 1) OrgCW=64

[0105] 2) 针对 $i=0:16$ , InputPivot[i]= $i*OrgCW$

[0106] 3) 针对 $i=reshaper\_model\_min\_bin\_idx:reshaper\_model\_max\_bin\_idx$ ,  
SignaledCW[i]= $OrgCW+(1-2*reshape\_model\_bin\_delta\_sign\_CW[i])*$

[0107] reshape\_model\_bin\_delta\_abs\_CW[i];

[0108] 4) 针对 $i=0:16$ , MappedPivot[i]计算如下:

[0109] MappedPivot[0]=0;

[0110] for( $i=0;i<16;i++$ )

[0111]  $\text{MappedPivot}[i+1] = \text{MappedPivot}[i] + \text{SignaledCW}[i]$

[0112] 逆映射函数“InvMap”也可以由InputPivot[]和MappedPivot[]限定。与“FwdMap”不同的是,针对“InvMap”分段线性模型,各个片段的两个输入枢轴点可以由MappedPivot[]限定,并且两个输出枢轴点可以由InputPivot[]限定,这与“FwdMap”相反。以这种方式,“FwdMap”的输入被分成相等片段,但“InvMap”的输入不保证被分成相等片段。

[0113] 如图5所示,针对帧间编码块,可以在映射域中执行运动补偿预测。换句话说,在运动补偿预测516之后,基于DPB中的参考信号来计算 $Y_{\text{pred}}$ ,可以应用“FwdMap”函数518,以将原始域中的亮度预测块映射至映射域, $Y'_{\text{pred}} = \text{FwdMap}(Y_{\text{pred}})$ 。针对帧内编码块,不应用“FwdMap”函数,因为帧内预测中使用的参考样本已经在映射域中了。在重构块506之后,可以计算 $Y_r$ 。可以应用“InvMap”函数508,以将映射域中的重构亮度值转换回原始域中的重构亮度值( $\hat{Y}_i = \text{InvMap}(Y_r)$ )。“InvMap”函数508可以应用于帧内编码亮度块和帧间编码亮度块。

[0114] 可以使用查找表(LUT)或使用即时计算来实现亮度映射过程(前向映射或逆映射)。如果使用LUT,则可以预先计算并预先存储表“FwdMapLUT[]”和“InvMapLUT[]”,以供在图块组级别使用,并且前向映射和逆映射可以分别简单地实现为 $\text{FwdMap}(Y_{\text{pred}}) = \text{FwdMapLUT}[Y_{\text{pred}}]$ 和 $\text{InvMap}(Y_r) = \text{InvMapLUT}[Y_r]$ 。另选地,可以使用即时计算。以前向映射函数“FwdMap”为例。为了确定亮度样本所属的片段,可以将样本值右移6位(在假定10位视频的情况下,其对应于16个相等片段),以获得片段索引。然后,该片段的线性模型参数被检索并即时应用,以计算映射亮度值。FwdMap函数估计如下:

[0115]  $Y'_{\text{pred}} = \text{FwdMap}(Y_{\text{pred}}) = ((b2-b1)/(a2-a1)) * (Y_{\text{pred}} - a1) + b1$

[0116] 其中,“i”是片段索引,a1是InputPivot[i],a2是InputPivot[i+1],b1是MappedPivot[i],b2是MappedPivot[i+1]。

[0117] 可以以类似方式即时计算“InvMap”函数,除了在找出样本值所属的片段时需要应用条件检查而不是简单的右位移位,因为不能保证映射域中的片段大小相等。

[0118] 在一些实施方式中,可以执行依赖于亮度的色度残差缩放。

[0119] 色度残差缩放设计为补偿亮度信号与其对应色度信号之间的相互作用。也在图块组级别传输是否启用色度残差缩放。如图6的表1所示,如果启用了亮度映射并且如果双树分区未应用于当前图块组,则传输附加标志(例如,tile\_group\_resampler\_chroma\_residual\_scale\_flag),以指示是否启用了依赖于亮度的色度残差缩放。当不使用亮度映射或在当前图块组中使用了双树分区时,将自动禁用依赖于亮度的色度残差缩放。此外,针对面积小于或等于4的色度块,可以禁用依赖于亮度的色度残差缩放。

[0120] 色度残差缩放取决于对应亮度预测块的平均值(针对帧内编码块和帧间编码块)。avgY' 作为亮度预测块的平均可以计算如下:

[0121] 
$$\text{avgY}' = \frac{\sum_{i=0}^{\text{宽度}-1} \sum_{j=0}^{\text{高度}-1} \text{predSamples}[i][j]}{\text{宽度} * \text{高度}}$$

[0122] 按照以下步骤计算 $C_{\text{ScaleInv}}$ 的值:

[0123] 1) 基于InvMap函数找到avgY' 所属的分段线性模型的索引 $Y_{\text{Idx}}$ 。

[0124] 2)  $C_{\text{ScaleInv}} = c\text{ScaleInv}[Y_{\text{Idx}}]$ ,其中cScaleInv[]是预先计算的16个片段LUT。

[0125] 在VTM4中的当前LMCS方法中,基于64个条目静态LUT ChromaResidualScaleLut和 SignaledCW[i]值,预先计算的LUT cScaleInv[i] (i在0至15的范围内)获得如下:

ChromaResidualScaleLut[64] = {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, 1638, 1489, 1489, 1489, 1489, 1365, 1365, 1365, 1365, 1260, 1260, 1260, 1170, 1170, 1170, 1170, 1092, 1092, 1092, 1092, 1024, 1024, 1024, 1024};

[0126] shiftC = 11

– if ( SignaledCW [ i ] == 0 )

cScaleInv [ i ] = (1 << shiftC)

– Otherwise,

cScaleInv [ i ] = ChromaResidualScaleLut[ (SignaledCW [ i ] >> 1)-1]

[0127] 静态表ChromaResidualScaleLut[]包含64个条目,并且SignaledCW[]在[0,128]范围内(假定输入是10位)。因此,除以2(例如,右移1)用于构建色度缩放因子LUT cScaleInv[]。色度缩放因子LUT cScaleInv[]可以包含多个色度缩放因子。LUT cScaleInv[]是在图块组级别构建的。

[0128] 如果当前块是使用帧内、CIIP或帧内块复制(IBC,也称为当前图片参考或CPR)模式编码的,则将avgY' 计算为帧内预测亮度值、CIIP预测亮度值或IBC预测亮度值的平均。否则,将avgY' 计算为前向映射的帧间预测亮度值(即,图5中的Y'<sub>pred</sub>)的平均。与基于样本执行的亮度映射不同,C<sub>ScaleInv</sub>是整个色度块的常量值。利用C<sub>ScaleInv</sub>,在解码器侧应用色度残差缩放如下:

[0129] 解码器侧:  $\hat{C}_{Res} = \hat{C}_{ResScale} * C_{ScaleInv}$

[0130] 其中 $\hat{C}_{ResScale}$ 是当前块的重构色度残差。在编码器侧,前向色度残差缩放(在变换和量化之前)执行如下:编码器侧:  $C_{ResScale} = C_{Res} * C_{Scale} = C_{Res} / C_{ScaleInv}$ 。

[0131] 在一些实施方式中,可以执行双树分区。

[0132] 在VVC草案4中,编码树方案支持亮度和色度具有单独的块树分区的能力。这也称为双树分区。图8的表3和图9的表4示出了双树分区信令。当开启(turn on)作为在SPS中传输的序列级别控制标志的“qtbtt\_dual\_tree\_intra\_flag”并且当前图块组是帧内编码时,可以首先针对亮度然后针对色度分别传输块分区信息。针对帧间编码图块组(P图块组和B图块组),不允许双树分区。当应用单独的块树模式时,亮度编码树块(CTB)被一种编码树结构分成CU,色度CTB被另一编码树结构分成色度CU,如图9的表4所示。

[0133] 当允许亮度和色度具有不同分区时,针对在不同颜色分量之间具有依赖性的编码工具,可能会出现问题。例如,在LMCS的情况下,使用对应亮度块的平均值,以计算出要应用于当前块的缩放因子。当使用双树时,这可能会造成整个CTU的延迟。例如,如果CTU的亮度块被垂直分割一次,并且CTU的色度块被水平分割一次,那么在对CTU的第一色度块进行解码之前,CTU的两个亮度块都被解码(使得可以计算平均值,这是计算色度缩放因子所需要

的)。在VVC中,CTU可以以亮度样本为单位大到 $128 \times 128$ 。针对硬件解码器流水线设计,如此大的延迟可能是非常有问题的。因此,VVC草案4可以禁止双树分区和依赖于亮度的色度缩放的组合。当针对当前图块组启用双树分区时,可以强制关闭色度缩放。请注意,LMCS的亮度映射部分在双树情况下仍然是允许的,因为它仅对亮度分量进行操作,并且没有交叉颜色分量依赖问题。依赖于颜色分量之间的依赖性来实现更好编码效率的编码工具的另一示例称为交叉分量线性模型(CCLM)。

[0134] 因此,图块组级别色度缩放因子LUT  $cScaleInv[]$  的获得不容易扩展。获得过程目前取决于具有64个条目的恒定色度LUT  $ChromaResidualScaleLut$ 。针对具有16个片段的10位视频,必须应用除以2的附加步骤。当片段数量发生变化时,例如,如果使用8个片段而不是16个片段,那么获得过程必须更改为应用除以4而不是除以2。该附加步骤不仅会导致精度损失,而且是笨拙且不必要的。

[0135] 另外,为了计算当前色度块的片段索引 $Y_{Idx}$  (其用于获得色度缩放因子),可以使用整个亮度块的平均值。这是不可取的,而且很可能是不必要的。考虑 $128 \times 128$ 的最大CTU大小。在这种情况下,平均亮度值是基于16384 ( $128 \times 128$ ) 个亮度样本计算的,这很复杂。此外,如果编码器选择 $128 \times 128$ 的亮度块分区,则该块更有可能包含同质内容。因此,块中亮度样本的子集可能足以计算亮度平均。

[0136] 在双树分区中,可以关闭色度缩放,以避免硬件解码器的潜在流水线问题。然而,如果使用显式信令来指示要应用的色度缩放因子,而不是使用对应亮度样本来获得色度缩放因子,则可以避免这种依赖性。在帧内编码图块组中启用色度缩放可以进一步提高编码效率。

[0137] 可以进一步改进分段线性参数的信令。当前,针对16个片段中的各个片段传输增量(delta)码字值。已经观察到,针对所述16个片段,通常只使用有限数量的不同码字。因此,可以进一步减少信令开销。

[0138] 本公开的实施方式提供了一种通过去除色度缩放LUT来处理视频内容的方法。

[0139] 如上所述,扩展64个条目的色度LUT可能是困难的,并且在使用其它分段线性模型(例如,8个片段、4个片段、64个片段等)时可能会出现问題。这也是不必要的,因为色度缩放因子可以被设置为与对应片段的亮度缩放因子相同,以达到相同的编码效率。在本公开的一些实施方式中,可以基于当前色度块的片段索引“ $Y_{Idx}$ ”来将色度缩放因子“chroma\_scaling”确定如下。

[0140] • 如果 $Y_{Idx} > reshaper\_model\_max\_bin\_idx$ ,  $Y_{Idx} < reshaper\_model\_min\_bin\_idx$ , 或者如果 $SignaledCW[Y_{Idx}] = 0$ , 则将chroma\_scaling设置为默认值,  $chroma\_scaling = 1.0$ 。

[0141] • 否则,将chroma\_scaling设置为 $SignaledCW[Y_{Idx}] / OrgCW$ 。

[0142] 当chroma\_scaling=1.0时,不应用缩放。

[0143] 上面确定的色度缩放因子可以具有小数精度。可以理解,可以应用定点近似,以避免对硬件/软件平台的依赖。此外,在解码器侧,可以执行逆色度缩放。因此,可以通过使用乘法然后是右移的定点算术来实现除法。按照定点精度的逆色度缩放因子

[0144]  $inverse\_chroma\_scaling[Y_{Idx}] = ((1 < (luma\_bit\_depth - \log_2(TOTAL\_NUMBER\_PIECES) + CSCALE\_FP\_PREC)) + (SignaledCW[Y_{Idx}] > 1)) / SignaledCW[Y_{Idx}];$

[0145] “inverse\_chroma\_scaling[ ]”可以基于定点近似中的比特数量“CSCALE\_FP\_PREC”确定如下。

[0146] 其中,luma\_bit\_depth是亮度位深度,TOTAL\_NUMBER\_PIECES是分段线性模型中的片段的总数(其在VVC草案4中被设置为16)。可以理解,“inverse\_chroma\_scaling[ ]”的值可能只需要每图块组计算一次,并且上面的除法是整数除法运算。

[0147] 可以应用进一步的量化来确定色度缩放因子和逆缩放因子。例如,可以针对“SignaledCW”的所有偶数( $2 \times m$ )值计算逆色度缩放因子,“SignaledCW”的奇数( $2 \times m + 1$ )值重新使用相邻偶数值的缩放因子的色度缩放因子。换句话说,可以使用以下内容:

```
for(i = reshaper_model_min_bin_idx; i <= reshaper_model_max_bin_idx; i++)
{
    tempCW = SignaledCW[i] >> 1) << 1;
    inverse_chroma_scaling[i] = ((1 << (luma_bit_depth -
    log2(TOTAL_NUMBER_PIECES) + CSCALE_FP_PREC)) + (tempCW >> 1)) / tempCW;
}
```

[0149] 对色度缩放因子进行量化可以被进一步推广。例如,可以针对“SignaledCW”的每一个第n值计算逆色度缩放因子“inverse\_chroma\_scaling[ ]”,所有其它相邻值共享同一色度缩放因子。例如,“n”可以被设置为4。因此,每4个相邻码字值可以共享相同的逆色度缩放因子值。在一些实施方式中,“n”的值可以是2的幂(power),这允许使用移位来计算除法。将 $\log_2(n)$ 的值表示为LOG2\_n,上式“tempCW=SignaledCW[i] >> 1) << 1”可以修改如下:

[0150] tempCW=SignaledCW[i] >> LOG2\_n) << LOG2\_n。

[0151] 在一些实施方式中,LOG2\_n的值可以是分段线性模型中使用的片段数量的函数。如果使用较少的片段,则使用较大的LOG2\_n可能是有益的。例如,如果TOTAL\_NUMBER\_PIECES的值小于或等于16,则LOG2\_n可以被设置为 $1 + (4 - \log_2(\text{TOTAL\_NUMBER\_PIECES}))$ 。如果TOTAL\_NUMBER\_PIECES大于16,则LOG2\_n可以被设置为0。

[0152] 本公开的实施方式提供了一种通过简化亮度预测块的平均来处理视频内容的方法。

[0153] 如上所述,为了确定当前色度块的片段索引“Y<sub>Idx</sub>”,可以使用对应亮度块的平均值。然而,针对大的块大小,平均过程可能涉及大量亮度样本。在最坏的情况下,平均过程可能涉及 $128 \times 128$ 个亮度样本。

[0154] 本公开的实施方式提供了简化的平均过程以将最坏情况减少到仅使用 $N \times N$ 个亮度样本(N是2的幂)。

[0155] 在一些实施方式中,如果二维亮度块的两个维度不都小于或等于预设阈值M(换句话说,两个维度中的至少一者大于M),则可以应用“下采样”,以仅使用该维度中的M个位置。不失一般性,以水平维度为例。如果宽度大于M,则仅将位置x( $x = i \times (\text{宽度} \gg \log_2(M))$ ),  $i = 0, \dots, M-1$ )处的样本用于平均。

[0156] 图10例示了应用所提出的简化来计算 $16 \times 8$ 亮度块的平均的示例。在该示例中,M被设置为4,并且仅将块中的16个亮度样本(阴影样本)用于平均。可以理解,预设阈值M不限于4,并且M可以被设置为2的幂的任何值。例如,预设阈值M可以是1、2、4、8等。

[0157] 在一些实施方式中,亮度块的水平维度和垂直维度可以具有不同预设阈值M。换

话说,平均操作的最坏情况可以使用 $M1 \times M2$ 个样本。

[0158] 在一些实施方式中,可以在不考虑维度的情况下在平均过程中限制样本数量。例如,最多可以使用16个样本,所述样本可以以 $1 \times 16$ 、 $16 \times 1$ 、 $2 \times 8$ 、 $8 \times 2$ 或 $4 \times 4$ 的形式分布在水平维度或垂直维度上,并且可以选择适合当前块的形状的任何形式。例如,如果块是窄而高的,则可以使用 $2 \times 8$ 个样本的矩阵,如果块是宽且短的,则可以使用 $8 \times 2$ 个样本的矩阵,并且如果块是方形的,则可以使用 $4 \times 4$ 个样本的矩阵。

[0159] 可以理解,当选择大的块大小时,该块内的内容往往是更同质的(homogeneous)。因此,虽然上述简化可以导致整个亮度块的平均值与真实平均之间的差异,但这种差异可以很小。

[0160] 此外,解码器侧运动矢量细化(DMVR)要求在可以应用运动补偿之前,解码器执行运动搜索以获得运动矢量。因此,在VVC标准中,DMVR模式可能很复杂,尤其是针对解码器。VVC标准中的双向光流(BDOF)模式会使情况进一步复杂化,因为BDOF是需要在DMVR之后应用以获得亮度预测块的附加顺序过程。由于色度缩放需要对应亮度预测块的平均值,因此,在可以计算平均值之前,可以应用DMVR和BDOF。

[0161] 为了解决该延迟问题,在本公开的一些实施方式中,在DMVR和BDOF之前使用亮度预测块来计算平均亮度值,并且平均亮度值用于获得色度缩放因子。这允许与DMVR过程和BDOF过程并行应用色度缩放,因此可以显著减少延迟。

[0162] 与本公开一致,可以设想延迟减少的变型。在一些实施方式中,该延迟减少还可以与上文讨论的简化平均过程结合,该简化平均过程仅使用亮度预测块的一部分来计算平均亮度值。在一些实施方式中,可以在DMVR过程之后和BDOF过程之前使用亮度预测块来计算平均亮度值。然后使用平均亮度值来获得色度缩放因子。这种设计允许与BDOF过程并行应用色度缩放,同时保持确定色度缩放因子的准确性。由于DMVR过程可以细化运动矢量,因此在DMVR过程之后使用具有细化运动矢量的预测样本可能比在DMVR过程之前使用具有运动矢量的预测样本更准确。

[0163] 此外,在VVC标准中,CU语法结构“coding\_unit()”包括语法要素“cu\_cbf”,以指示当前CU中是否存在任何非零残差系数。在TU级别,TU语法结构“transform\_unit()”包括语法要素“tu\_cbf\_cb”和“tu\_cbf\_cr”,以指示当前TU中是否存在任何非零色度(Cb或Cr)残差系数。按照惯例,在VVC草案4中,如果在图块组级别启用色度缩放,则始终调用对应亮度块的平均。

[0164] 本公开的实施方式进一步提供了一种通过绕过亮度平均过程来处理视频内容的方法。与所公开的实施方式一致,由于色度缩放过程被应用于残差色度系数,所以如果不存在非零色度系数,则可以绕过亮度平均过程。这可以基于以下条件来确定:

[0165] 条件1:cu\_cbf等于0

[0166] 条件2:tu\_cbf\_cr和tu\_cbf\_cb都等于0

[0167] 如上文所讨论的,“cu\_cbf”可以指示当前CU中是否存在任何非零残差系数,并且“tu\_cbf\_cb”和“tu\_cbf\_cr”可以指示当前TU中是否存在任何非零色度(Cb或Cr)残差系数。当满足条件1或条件2时,可以绕过亮度平均过程。

[0168] 在一些实施方式中,仅使用预测块的 $N \times N$ 个样本来获得平均值,这简化了平均过程。例如,当N等于1时,仅使用预测块的左上侧样本。然而,使用预测块的这种简化平均处理



仍然需要生成预测块,从而导致延迟。

[0169] 在一些实施方式中,可以直接使用参考亮度样本来生成色度缩放因子。这允许解码器与亮度预测过程并行地获得缩放因子,从而减少延迟。下面将分别描述使用参考亮度样本的帧内预测和帧间预测。

[0170] 在示例性帧内预测中,可以将同一图片中的经解码的相邻样本用作参考样本,以生成预测块。这些参考样本可以包括例如当前块顶部、当前块左侧或当前块左上侧的样本。这些参考样本的平均可以用于获得色度缩放因子。在一些实施方式中,可以使用这些参考样本的一部分的平均。例如,仅对最靠近当前块的左上侧位置的K个参考样本(例如,K=3)进行平均。

[0171] 在示例性帧间预测中,可以使用来自时间参考图片的参考样本来生成预测块。这些参考样本由参考图片索引和运动矢量标识。如果运动矢量具有小数精度,则可以应用内插,用于确定参考样本平均的参考样本可以包括内插之前或之后的参考样本。内插之前的参考样本可以包括裁剪(clip)为整数精度的运动矢量。与所公开的实施方式一致,所有参考样本都可以用于计算平均。另选地,可以仅使用参考样本的一部分(例如,与当前块的左上侧位置相对应的参考样本)来计算平均。

[0172] 如图5所示,帧内预测(例如,帧内预测514或528)可以在整形后的(reshape)域中执行,而帧间预测是在原始域中执行的。因此,针对帧间预测,可以对预测块应用前向映射,并使用前向映射后的亮度预测块来计算平均。为了减少延迟,可以使用前向映射之前的预测块来计算平均。例如,可以使用前向映射之前的块、前向映射之前的块的 $N \times N$ 部分或前向映射之前的块的左上侧样本。

[0173] 本公开的实施方式进一步提供了一种利用针对双树分区的色度缩放来处理视频内容的方法。

[0174] 由于对亮度的依赖会导致硬件设计复杂化,因此可以针对启用双树分区的帧内编码图块组关闭色度缩放。然而,这种限制会导致编码效率损失。代替对对应亮度块的样本值求平均以计算 $\text{avgY}$ ,确定片段索引 $Y_{\text{Idx}}$ 并获得色度缩放因子 $\text{inverse\_chroma\_scaling}[Y_{\text{Idx}}]$ ,可以在比特流中明确地传输色度缩放因子,以避免在双树分区的情况下对亮度的依赖。

[0175] 可以在不同级别传输色度缩放索引。例如,可以在编码单元(CU)级别传输色度缩放索引以及色度预测模式,如图11的表5所示。语法要素“ $\text{lmc\_scaling\_factor\_idx}$ ”可以用于确定当前色度块的色度缩放因子。当不存在“ $\text{lmc\_scaling\_factor\_idx}$ ”时,可以推断当前色度块的色度缩放因子在浮点精度上等于1.0或在定点精度上等效地等于1.0( $1 < \text{CSCALE\_FP\_PREC}$ )。“ $\text{lmc\_chroma\_scaling\_idx}$ ”的允许值范围是在图块组级别确定的,稍后将讨论。

[0176] 根据“ $\text{lmc\_chroma\_scaling\_idx}$ ”的可能值,信令成本可能很高,尤其是针对小的块。因此,在本公开的一些实施方式中,图11的表5中的信令条件可以另外包括块大小条件。例如,只有在当前块包含超过给定数量的色度样本或当前块的宽度大于给定宽度W或高度大于给定高度H时,才会传输该语法要素“ $\text{lmc\_chroma\_scaling\_idx}$ ”(以斜体和灰色阴影强调)。针对较小的块,如果未传输“ $\text{lmc\_chroma\_scaling\_idx}$ ”,则可以在解码器侧确定其色度缩放因子。在一些实施方式中,色度缩放因子可以按照浮点精度设置为1.0。在一些实

施方式中,可以在图块组报头级别添加默认“lmcscchroma\_scaling\_idx”值(参见图6的表1)。未传输“lmcscchroma\_scaling\_idx”的小的块可以使用该图块组级别的默认索引来获得对应色度缩放因子。在一些实施方式中,小的块的色度缩放因子可以从其已明确地传输缩放因子的邻居(例如,顶部邻居或左侧邻居)继承的。

[0177] 除了在CU级别传输该语法要素“lmcscchroma\_scaling\_idx”之外,还可以在CTU级别传输该语法要素。然而,鉴于VVC中的最大CTU大小为 $128 \times 128$ ,在CTU级别进行相同的缩放可能过于粗糙。因此,在本公开的一些实施方式中,可以使用固定粒度来传输该语法要素“lmcscchroma\_scaling\_idx”。例如,针对CTU中的各个 $16 \times 16$ 区域,传输一个“lmcscchroma\_scaling\_idx”并将其应用于该 $16 \times 16$ 区域中的所有样本。

[0178] 针对当前图块组的“lmcscchroma\_scaling\_idx”范围取决于当前图块组中允许的色度缩放因子值的数量。可以基于如上文所讨论的64个条目色度LUT来确定当前图块组中允许的色度缩放因子值的数量。另选地,可以使用上文讨论的色度缩放因子计算来确定当前图块组中允许的色度缩放因子值的数量。

[0179] 例如,在“量化”方法中, $\text{LOG}_2 n$ 的值可以被设置为2(即,“n”被设置为4),并且当前图块组的分段线性模型中的各个片段的码字分配可以设置如下: {0,65,66,64,67,62,62,64,64,64,67,64,64,62,61,0}。那么整个图块组只有2个可能的缩放因子值,因为从64到67的任何码字值都可以具有相同的缩放因子值(按照小数精度则为1.0),并且从60到63的任何码字值都可以具有相同的缩放因子值(按照小数精度则为 $60/64=0.9375$ )。针对未被分配任何码字的两个末端片段,色度缩放因子默认设置为1.0。因此,在该示例中,针对当前图块组中的块,一位足以传输“lmcscchroma\_scaling\_idx”。

[0180] 除了使用分段线性模型确定可能的色度缩放因子值的数量之外,编码器可以在图块组报头处传输色度缩放因子值的集合。然后,在块级别,可以使用色度缩放因子值的该集合和针对该块的“lmcscchroma\_scaling\_idx”值来确定针对该块的色度缩放因子值。

[0181] CABAC编码可以应用于编码“lmcscchroma\_scaling\_idx”。块的CABAC上下文可以取决于块的相邻块的“lmcscchroma\_scaling\_idx”。例如,左侧的块或顶部的块可以用于形成CABAC上下文。就“lmcscchroma\_scaling\_idx”的该语法要素的二值化而言,适用于VVC草案4中的ref\_idx\_l0和ref\_idx\_l1语法要素的相同截断莱斯(truncated Rice)二值化可以用于将“lmcscchroma\_scaling\_idx”二值化。

[0182] 传输“chroma\_scaling\_idx”的好处是编码器可以就率失真成本而言选择最佳的“lmcscchroma\_scaling\_idx”。使用率失真优化选择“lmcscchroma\_scaling\_idx”可以提高编码效率,这有助于抵消信令成本的增加。

[0183] 本公开的实施方式进一步提供了一种利用LMCS分段线性模型的信令来处理视频内容的方法。

[0184] 虽然LMCS方法使用了具有16个片段的分段线性模型,但是图块组中“SignaledCW[i]”的唯一值的数量往往远小于16。例如,16个片段中的一些片段可以使用默认数量的码字“OrgCW”,并且所述16个片段中的一些片段可以具有彼此相同的数量的码字。因此,传输LMCS分段线性模型的另选方法可以包括传输多个唯一码字“listUniqueCW[]”,并发送片段中的各个片段的索引,以指示当前片段的“listUniqueCW[]”的要素。

[0185] 图12的表6中提供了修改的语法表。在图12的表6中,新的或修改的语法以斜体和

灰色阴影强调。

[0186] 所公开的信令方法的语义如下(其中变化用下划线表示):

[0187] reshaper\_model\_min\_bin\_idx指定要在整形器构建过程中使用的最小bin(或片段)索引。reshape\_model\_min\_bin\_idx的值应在0到MaxBinIdx的范围内(包括端点(inclusive))。MaxBinIdx的值应等于15。

[0188] reshaper\_model\_delta\_max\_bin\_idx指定最大允许bin(或片段)索引MaxBinIdx减去要在整形器构建过程中使用的最大bin索引。reshape\_model\_max\_bin\_idx的值被设置为等于MaxBinIdx-reshape\_model\_delta\_max\_bin\_idx。

[0189] reshaper\_model\_bin\_delta\_abs\_cw\_prec\_minus1加1指定用于表示语法reshape\_model\_bin\_delta\_abs\_CW[i]的位的数量。

[0190] reshaper\_model\_bin\_num\_unique\_cw\_minus1加1指定码字数组listUniqueCW的大小。

[0191] reshaper\_model\_bin\_delta\_abs\_CW[i]指定第*i*个bin的绝对增量码字值。

[0192] reshaper\_model\_bin\_delta\_sign\_CW\_flag[i]指定

[0193] reshape\_model\_bin\_delta\_abs\_CW[i]的符号如下:

[0194] -如果reshape\_model\_bin\_delta\_sign\_CW\_flag[i]等于0,则对应变量RspDeltaCW[i]是正值。

[0195] -否则(reshape\_model\_bin\_delta\_sign\_CW\_flag[i]不等于0),则对应变量RspDeltaCW[i]是负值。

[0196] 当reshape\_model\_bin\_delta\_sign\_CW\_flag[i]不存在时,推断其等于0。

[0197] 变量RspDeltaCW[i]导出为 $RspDeltaCW[i] = (1 - 2 * \text{reshape\_model\_bin\_delta\_sign\_CW}[i]) * \text{reshape\_model\_bin\_delta\_abs\_CW}[i]$ 。

[0198] 变量listUniqueCW[0]被设置为等于OrgCW。获得变量listUniqueCW[i] ( $i = 1 \cdots \text{reshaper\_model\_bin\_num\_unique\_cw\_minus1}$ ,包括端点)如下:

[0199] -变量OrgCW被设置为等于 $(1 < \text{BitDepthY}) / (\text{MaxBinIdx} + 1)$ 。

[0200] -listUniqueCW[i] = OrgCW + RspDeltaCW[i-1]

[0201] reshaper\_model\_bin\_cw\_idx[i]指定用于获得RspCW[i]的数组listUniqueCW[ ]的索引。reshaper\_model\_bin\_cw\_idx[i]的值应在0到(reshaper\_model\_bin\_num\_unique\_cw\_minus1+1)的范围内,包括端点。

[0202] 获得RspCW[i]如下:

[0203] -如果reshaper\_model\_min\_bin\_idx ≤ *i* ≤ reshaper\_model\_max\_bin\_idx,则

[0204] RspCW[i] = listUniqueCW[reshaper\_model\_bin\_cw\_idx[i]]。

[0205] -否则,RspCW[i] = 0。

[0206] 如果BitDepthY的值等于10,则RspCW[i]的值可以在32到2\*OrgCW-1的范围内。

[0207] 本公开的实施方式进一步提供了一种利用块级别的条件色度缩放来处理视频内容的方法。

[0208] 如图6的表1所示,可以通过在图块组级别传输的“tile\_group\_reshaper\_chroma\_residual\_scale\_flag”来确定是否应用色度缩放。

[0209] 然而,确定是否在块级别应用色度缩放可能是有益的。例如,在一些公开的实施方案

式中,可以传输CU级别标志,以指示色度缩放是否应用于当前块。CU级别标志的存在可以以图块组级别标志“tile\_group\_resaper\_chroma\_residual\_scale\_flag”为条件。即,仅当在图块组级别允许色度缩放时才可以传输CU级别标志。虽然允许编码器基于色度缩放对当前块是否有益来选择是否使用色度缩放,但这也可能导致显著的信令开销。

[0210] 与所公开的实施方式一致,为避免上述信令开销,色度缩放是否应用于块可以以块的预测模式为条件。例如,当块被帧间预测时,预测信号往往是好的,特别是如果它的参考图片在时间距离上更近的话。因此,因为残差预计非常小,所以可以绕过色度缩放。例如,较高时间级别的图片往往具有时间距离接近的参考图片。并且针对块,在使用附近参考图片的图片中,可以禁用色度缩放。当前图片与块的参考图片之间的图片顺序计数(POC)差异可以用于确定是否满足该条件。

[0211] 在一些实施方式中,可以针对所有帧间编码块禁用色度缩放。在一些实施方式中,针对在VVC标准中限定的组合帧内/帧间预测(CIIP)模式,可以禁用色度缩放。

[0212] 在VVC标准中,CU语法结构“coding\_unit()”包括语法要素“cu\_cbf”,以指示当前CU中是否存在任何非零残差系数。在TU级别,TU语法结构“transform\_unit()”包括语法要素“tu\_cbf\_cb”和“tu\_cbf\_cr”,以指示当前TU中是否存在任何非零色度(Cb或Cr)残差系数。色度缩放过程可以以这些标志为条件。如上所述,如果不存在非零残差系数,则可以调用对应亮度色度缩放过程的平均。通过调用平均,可以绕过色度缩放过程。

[0213] 图13例示了用于处理视频内容的计算机实现的方法1300的流程图。在一些实施方式中,方法1300可以由编解码器(例如,图2A至图2B中的编码器或图3A至图3B中的解码器)执行。例如,编解码器可以被实现为用于对视频序列进行编码或转码的装置(例如,装置400)的一个或更多个软件部件或硬件部件。在一些实施方式中,视频序列可以是未压缩视频序列(例如,视频序列202)或被解码的压缩视频序列(例如,视频流304)。在一些实施方式中,视频序列可以是监控视频序列,其可以由与装置的处理器的处理器(例如,处理器402)相关联的监控设备(例如,图4中的视频输入设备)捕获。视频序列可以包括多个图片。所述装置可以在图片级别执行方法1300。例如,所述装置可以采用方法1300一次处理一个图片。又例如,所述装置可以采用方法1300一次处理多个图片。方法1300可以包括以下步骤。

[0214] 在步骤1302,可以接收与图片相关联的色度块和亮度块。可以理解,图片可以与色度分量和亮度分量相关联。因此,图片可以与包含色度样本的色度块和包含亮度样本的亮度块相关联。

[0215] 在步骤1304,可以确定与亮度块相关联的亮度缩放信息。在一些实施方式中,亮度缩放信息可以是在图片的数据流中传输的语法要素或基于在图片的数据流中传输的语法要素获得的变量。例如,亮度缩放信息可以包括上式中描述的“reshape\_model\_bin\_delta\_sign\_CW[i]和reshape\_model\_bin\_delta\_abs\_CW[i]”和/或上式中描述的“SignaledCW[i]”等。在一些实施方式中,亮度缩放信息可以包括基于亮度块确定的变量。例如,可以通过计算与亮度块相邻的亮度样本(诸如亮度块顶部的行和亮度块左侧的列中的亮度样本)的平均值来确定平均亮度值。

[0216] 在步骤1306,可以基于亮度缩放信息来确定色度缩放因子。

[0217] 在一些实施方式中,可以基于亮度缩放信息来确定亮度块的亮度缩放因子。例如,根据上式“inverse\_chroma\_scaling[i]=((1<<(luma\_bit\_depth-log2(TOTAL\_NUMBER\_

$\text{PIECES}) + \text{CSCALE\_FP\_PREC})) + (\text{tempCW} > 1)) / \text{tempCW}$ ”, 可以基于亮度缩放信息 (例如, “tempCW”) 来确定亮度缩放因子。然后可以基于亮度缩放因子的值来进一步确定色度缩放因子。例如, 色度缩放因子可以设置为等于亮度缩放因子的值。可以理解, 在被设置为色度缩放因子之前, 可以对亮度缩放因子的值进行进一步的计算。作为另一示例, 色度缩放因子可以被设置为等于 “ $\text{SignaledCW}[Y_{\text{Idx}}] / \text{OrgCW}$ ”, 其中, 当前色度块的片段索引 “ $Y_{\text{Idx}}$ ” 可以是基于与亮度块相关联的平均亮度值确定的。

[0218] 在步骤1308, 可以使用色度缩放因子来处理色度块。例如, 可以使用色度缩放因子来处理色度块的残差, 以生成色度块的缩放残差。色度块可以是Cb色度分量或Cr色度分量。

[0219] 在一些实施方式中, 如果满足条件, 则可以处理色度块。例如, 所述条件可以包括与图片相关联的目标编码单元不具有非零残差; 或与图片相关联的目标变换单元不具有非零色度残差。可以基于目标编码单元的第一编码块标志的值来确定目标编码单元不具有非零残差。并且可以基于目标变换单元的第一分量的第二编码块标志和第二分量的第三编码块标志的值来确定目标变换单元不具有非零色度残差。例如, 第一分量可以是Cb分量, 而第二分量可以是Cr分量。

[0220] 可以理解, 方法1300的各个步骤可以作为独立的方法来执行。例如, 步骤1308中描述的用于确定色度缩放因子的方法可以作为独立的方法来执行。

[0221] 图14例示了用于处理视频内容的计算机实现的方法1400的流程图。在一些实施方式中, 方法1400可以由编解码器 (例如, 图2A至图2B中的编码器或图3A至图3B中的解码器) 执行。例如, 编解码器可以被实现为用于对视频序列进行编码或转码的装置 (例如, 装置400) 的一个或更多个软件部件或硬件部件。在一些实施方式中, 视频序列可以是未压缩视频序列 (例如, 视频序列202) 或被解码的压缩视频序列 (例如, 视频流304)。在一些实施方式中, 视频序列可以是监控视频序列, 其可以由与装置的处理器的处理器 (例如, 处理器402) 相关联的监控设备 (例如, 图4中的视频输入设备) 捕获。视频序列可以包括多个图片。所述装置可以在图片级别执行方法1400。例如, 所述装置可以采用方法1400一次处理一个图片。又例如, 所述装置可以采用方法1400一次处理多个图片。方法1400可以包括以下步骤。

[0222] 在步骤1402, 可以接收与图片相关联的色度块和亮度块。可以理解, 图片可以与色度分量和亮度分量相关联。因此, 图片可以与包含色度样本的色度块和包含亮度样本的亮度块相关联。在一些实施方式中, 亮度块可以包括  $N \times M$  个亮度样本。 $N$  可以是亮度块的宽度,  $M$  可以是亮度块的高度。如上文所讨论的, 亮度块的亮度样本可以用于确定目标色度块的片段索引。因此, 可以接收与视频序列的图片相关联的亮度块。可以理解,  $N$  和  $M$  可以具有相同的值。

[0223] 在步骤1404, 响应于  $N$  和  $M$  中的至少一者大于阈值, 可以选择  $N \times M$  个亮度样本的子集。为了加快片段索引的确定, 当满足特定条件时, 可以对亮度块进行“下采样”。换句话说, 亮度块中亮度样本的子集可以用于确定片段索引。在一些实施方式中, 特定条件是  $N$  和  $M$  中的至少一者大于阈值。在一些实施方式中, 阈值可以基于  $N$  和  $M$  中的至少一者。阈值可以是2的幂。例如, 阈值可以是4、8、16等。以4为例, 当  $N$  或  $M$  大于4时, 可以选择亮度样本的子集。在图10的示例中, 亮度块的宽度和高度都大于阈值4, 因此选择了  $4 \times 4$  个样本的子集。可以理解, 也可以选择  $2 \times 8$ 、 $1 \times 16$  等的子集进行处理。

[0224] 在步骤1406, 可以确定  $N \times M$  个亮度样本的子集的平均值。

[0225] 在一些实施方式中,确定平均值可以进一步包括:确定是否满足第二条件;响应于确定满足第二条件,确定 $N \times M$ 个亮度样本的子集的平均值。例如,第二条件可以包括与图片相关联的目标编码单元不具有非零残差系数;或者目标编码单元不具有非零色度残差系数。

[0226] 在步骤1408,可以确定基于平均值的色度缩放因子。在一些实施方式中,为了确定色度缩放因子,可以基于平均值来确定色度块的片段索引,可以确定色度块的片段索引是否满足第一条件,然后响应于色度块的片段索引满足第一条件,色度缩放因子可以被设置为默认值。默认值可以指示不应用色度缩放。例如,默认值可以是具有小数精度的1.0。可以理解,可以对默认值应用定点近似。响应于色度块的片段索引不满足第一条件,可以基于平均值确定色度缩放因子。更具体地,色度缩放因子可以被设置为 $\text{SignaledCW}[Y_{\text{Idx}}]/\text{OrgCW}$ ,并且可以基于对应亮度块的平均值来确定目标色度块的片段索引“ $Y_{\text{Idx}}$ ”。

[0227] 在一些实施方式中,第一条件可以包括色度块的片段索引大于传输的码字的最大索引或小于传输的码字的最小索引。传输的码字的最大索引和最小索引可以确定如下。

[0228] 可以使用基于输入信号(例如,亮度样本)的分段线性模型(例如,LMCS)来生成码字。如上文所讨论的,输入信号的动态范围可以被分成多个片段(例如,16个片段),并且输入信号的各个片段可以用于生成码字的bin作为输出。因此,码字的各个bin可以具有与输入信号的片段相对应的bin索引。在该示例中,bin索引的范围可以是0-15。在一些实施方式中,输出(即,码字)的值介于最小值(例如,0)与最大值(例如,255)之间,并且可以传输具有介于最小值与最大值之间的值的多个码字。可以确定传输的多个码字的bin索引。在传输的多个码字的bin索引中,可以进一步确定传输的多个码字的bin的最大bin索引和最小bin索引。

[0229] 除了色度缩放因子之外,方法1400可以进一步基于传输的多个码字的bin来确定亮度缩放因子。亮度缩放因子可以用作逆色度缩放因子。上文已经描述了用于确定亮度缩放因子的公式,这里省略了其描述。在一些实施方式中,多个相邻的传输的码字共享亮度缩放因子。例如,两个或四个相邻的传输的码字可以共享相同的亮度缩放因子,这可以减少确定亮度缩放因子的负担。

[0230] 在步骤1410,可以使用色度缩放因子来处理色度块。如上文参考图5所讨论的,多个色度缩放因子可以在图块组级别构建色度缩放因子LUT,并在解码器侧应用于目标块的重构色度残差。类似地,色度缩放因子也可以应用于编码器侧。

[0231] 可以理解,方法1400的各个步骤可以作为独立的方法来执行。例如,步骤1308中描述的用于确定色度缩放因子的方法可以作为独立的方法来执行。

[0232] 图15例示了用于处理视频内容的计算机实现的方法1500的流程图。在一些实施方式中,方法1500可以由编解码器(例如,图2A至图2B中的编码器或图3A至图3B中的解码器)执行。例如,编解码器可以被实现为用于对视频序列进行编码或转码的装置(例如,装置400)的一个或更多个软件部件或硬件部件。在一些实施方式中,视频序列可以是未压缩视频序列(例如,视频序列202)或被解码的压缩视频序列(例如,视频流304)。在一些实施方式中,视频序列可以是监控视频序列,其可以由与装置的处理器的处理器(例如,处理器402)相关联的监控设备(例如,图4中的视频输入设备)捕获。视频序列可以包括多个图片。所述装置可以在图片级别执行方法1500。例如,所述装置可以采用方法1500一次处理一个图片。又例如,

所述装置可以采用方法1500一次处理多个图片。方法1500可以包括以下步骤。

[0233] 在步骤1502,可以确定所接收的视频数据中是否存在色度缩放索引。

[0234] 在步骤1504,响应于确定所接收的视频数据中不存在色度缩放索引,可以确定没有向所接收的视频数据应用色度缩放。

[0235] 在步骤1506,响应于确定所接收的视频数据中存在色度缩放,可以基于色度缩放索引来确定色度缩放因子。

[0236] 图16例示了用于处理视频内容的计算机实现的方法1600的流程图。在一些实施方式中,方法1600可以由编解码器(例如,图2A至图2B中的编码器或图3A至图3B中的解码器)执行。例如,编解码器可以被实现为用于对视频序列进行编码或转码的装置(例如,装置400)的一个或更多个软件部件或硬件部件。在一些实施方式中,视频序列可以是未压缩视频序列(例如,视频序列202)或被解码的压缩视频序列(例如,视频流304)。在一些实施方式中,视频序列可以是监控视频序列,其可以由与装置的处理器的处理器(例如,处理器402)相关联的监控设备(例如,图4中的视频输入设备)捕获。视频序列可以包括多个图片。所述装置可以在图片级别执行方法1600。例如,所述装置可以采用方法1600一次处理一个图片。又例如,所述装置可以采用方法1600一次处理多个图片。方法1600可以包括以下步骤。

[0237] 在步骤1602,可以接收用于输入视频信号的动态范围的多个唯一码字。

[0238] 在步骤1604,可以接收索引。

[0239] 在步骤1606,可以基于索引来选择多个唯一码字中的至少一个码字。

[0240] 在步骤1608,可以基于所选择的至少一个码字来确定色度缩放因子。

[0241] 在一些实施方式中,还提供了一种包括指令的非暂时性计算机可读存储介质,并且所述指令可以由用于执行上述方法的设备(诸如所公开的编码器和解码器)来执行。例如,非暂时性介质的常见形式包括软盘、软磁盘、硬盘、固态驱动器、磁带或任何其它磁性数据存储介质、CD-ROM、任何其它光学数据存储介质、具有洞图案的任何物理介质、RAM、PROM和EPROM、FLASH-EPROM或任何其它闪存、NVRAM、缓存部、寄存器、任何其它存储器芯片或盒式磁带以及它们的网络版本。设备可以包括一个或更多个处理器(CPU)、输入/输出接口、网络接口和/或存储器。

[0242] 可以理解,上述实施方式可以通过硬件、或软件(程序代码)、或硬件和软件的组合来实现。如果通过软件实现,则该软件可以存储在上述计算机可读介质中。所述软件在由处理器执行时可以执行所公开的方法。本公开中描述的计算单元和其它功能单元可以通过硬件、或软件、或硬件和软件的组合来实现。本领域普通技术人员也可以理解,上述模块/单元中的多个模块/单元可以组合为一个模块/单元,并且上述模块/单元中的各个模块/单元还可以划分成多个子模块/子单元。

[0243] 可以使用以下条款进一步描述实施方式:

[0244] 1.一种用于处理视频内容的计算机实现的方法,所述方法包括以下步骤:

[0245] 接收与图片相关联的色度块和亮度块;

[0246] 确定与所述亮度块相关联的亮度缩放信息;

[0247] 基于所述亮度缩放信息来确定色度缩放因子;以及

[0248] 使用所述色度缩放因子来处理所述色度块。

[0249] 2.根据条款1所述的方法,其中,基于所述亮度缩放信息来确定色度缩放因子的步

骤进一步包括：

[0250] 基于所述亮度缩放信息来确定所述亮度块的亮度缩放因子；

[0251] 基于所述亮度缩放因子的值来确定所述色度缩放因子。

[0252] 3. 根据条款2所述的方法，其中，基于所述亮度缩放因子的值来确定所述色度缩放因子的步骤进一步包括：

[0253] 将所述色度缩放因子设置为等于所述亮度缩放因子的值。

[0254] 4. 根据条款1至3中任一项所述的方法，其中，使用所述色度缩放因子来处理所述色度块的步骤进一步包括：

[0255] 确定是否满足第一条件；以及

[0256] 执行以下项中的一项：

[0257] 响应于确定满足所述第一条件，使用所述色度缩放因子来处理所述色度块；或者

[0258] 响应于确定不满足所述第一条件，绕过使用所述色度缩放因子来处理所述色度块。

[0259] 5. 根据条款4所述的方法，其中，所述第一条件包括：

[0260] 与所述图片相关联的目标编码单元不具有非零残差；或者

[0261] 与所述图片相关联的目标变换单元不具有非零色度残差。

[0262] 6. 根据条款5所述的方法，其中，

[0263] 所述目标编码单元不具有非零残差是基于所述目标编码单元的第一编码块标志的值确定的，并且

[0264] 所述目标变换单元不具有非零色度残差是基于所述目标变换单元的第一色度分量的第二编码块标志和第二亮度色度分量的第三编码块标志的值确定的。

[0265] 7. 根据条款6所述的方法，其中，

[0266] 所述第一编码块标志的值为0；并且

[0267] 所述第二编码块标志和所述第三编码块标志的值为0。

[0268] 8. 根据条款1至7中任一项所述的方法，其中，使用所述色度缩放因子来处理所述色度块的步骤包括：

[0269] 使用所述色度缩放因子来处理所述色度块的残差。

[0270] 9. 一种用于处理视频内容的装置，所述装置包括：

[0271] 存储器，所述存储器存储有指令集；以及

[0272] 处理器，所述处理器联接至所述存储器并被配置为执行所述指令集，以使所述装置执行以下操作：

[0273] 接收与图片相关联的色度块和亮度块；

[0274] 确定与所述亮度块相关联的亮度缩放信息；

[0275] 基于所述亮度缩放信息来确定色度缩放因子；以及

[0276] 使用所述色度缩放因子来处理所述色度块。

[0277] 10. 根据条款9所述的装置，其中，在基于所述亮度缩放信息来确定色度缩放因子时，所述处理器被配置为执行所述指令集以使所述装置进一步执行以下操作：

[0278] 基于所述亮度缩放信息来确定所述亮度块的亮度缩放因子；

[0279] 基于所述亮度缩放因子的值来确定所述色度缩放因子。



[0280] 11. 根据条款10所述的装置, 其中, 在基于所述亮度缩放因子的值来确定所述色度缩放因子时, 所述处理器被配置为执行所述指令集以使所述装置进一步执行以下操作:

[0281] 将所述色度缩放因子设置为等于所述亮度缩放因子的值。

[0282] 12. 根据条款9至11中任一项所述的装置, 其中, 在使用所述色度缩放因子来处理所述色度块时, 所述处理器被配置为执行所述指令集以使所述装置进一步执行以下操作:

[0283] 确定是否满足第一条件; 以及

[0284] 执行以下项中的一项:

[0285] 响应于确定满足所述第一条件, 使用所述色度缩放因子来处理所述色度块; 或者

[0286] 响应于确定不满足所述第一条件, 绕过使用所述色度缩放因子来处理所述色度块。

[0287] 13. 根据条款12所述的装置, 其中, 所述第一条件包括:

[0288] 与所述图片相关联的目标编码单元不具有非零残差; 或者

[0289] 与所述图片相关联的目标变换单元不具有非零色度残差。

[0290] 14. 根据条款13所述的装置, 其中,

[0291] 所述目标编码单元不具有非零残差是基于所述目标编码单元的第一编码块标志的值确定的, 并且

[0292] 所述目标变换单元不具有非零色度残差是基于所述目标变换单元的第一色度分量的第二编码块标志和第二色度分量的第三编码块标志的值确定的。

[0293] 15. 根据条款14所述的装置, 其中,

[0294] 所述第一编码块标志的值为0; 并且

[0295] 所述第二编码块标志和所述第三编码块标志的值为0。

[0296] 16. 根据条款9至15中任一项所述的装置, 其中, 在使用所述色度缩放因子来处理所述色度块时, 所述处理器被配置为执行所述指令集以使所述装置进一步执行以下操作:

[0297] 使用所述色度缩放因子来处理所述色度块的残差。

[0298] 17. 一种非暂时性计算机可读存储介质, 所述非暂时性计算机可读存储介质存储有指令集, 所述指令集能够由设备的一个或多个处理器执行, 以使所述设备执行用于处理视频内容的方法, 所述方法包括以下步骤:

[0299] 接收与图片相关联的色度块和亮度块;

[0300] 确定与所述亮度块相关联的亮度缩放信息;

[0301] 基于所述亮度缩放信息来确定色度缩放因子; 以及

[0302] 使用所述色度缩放因子来处理所述色度块。

[0303] 18. 一种用于处理视频内容的计算机实现的方法, 所述方法包括以下步骤:

[0304] 接收与图片相关联的色度块和亮度块, 所述亮度块包括 $N \times M$ 个亮度样本;

[0305] 响应于 $N$ 和 $M$ 中的至少一者大于阈值, 选择所述 $N \times M$ 个亮度样本的子集;

[0306] 确定所述 $N \times M$ 个亮度样本的所述子集的平均值;

[0307] 基于所述平均值来确定色度缩放因子; 以及

[0308] 使用所述色度缩放因子来处理所述色度块。

[0309] 19. 一种用于处理视频内容的计算机实现的方法, 所述方法包括以下步骤:

[0310] 确定所接收的视频数据中是否存在色度缩放索引;

[0311] 响应于确定所接收的视频数据中不存在所述色度缩放索引,确定色度缩放没有应用于所接收的视频数据;以及

[0312] 响应于确定所接收的视频数据中存在所述色度缩放,基于所述色度缩放索引来确定色度缩放因子。

[0313] 20.一种用于处理视频内容的计算机实现的方法,所述方法包括以下步骤:

[0314] 接收用于输入视频信号的动态范围的多个唯一码字;

[0315] 接收索引;

[0316] 基于所述索引来选择所述多个唯一码字中的至少一个码字;以及

[0317] 基于所选择的至少一个码字来确定色度缩放因子。

[0318] 除了通过使用计算机可读程序代码来实现上述方法之外,上述方法还可以以逻辑门、开关、ASIC、可编程逻辑控制器和嵌入式微控制器的形式来实现。因此,这样的控制器可以被认为是硬件部件,并且被包括在控制器中并被配置用于实现各种功能的装置也可以被认为是硬件部件内部的结构。或者,被配置用于实现各种功能的装置甚至可以被认为是被配置用于实现该方法的软件模块和硬件部件内部的结构两者。

[0319] 本公开可以在由计算机执行的计算机可执行指令(例如,程序模块)的一般上下文中描述。通常,程序模块包括用于执行特定任务或实现特定抽象数据类型的例程、程序、对象、程序集、数据结构、类等。本公开的实施方式还可以在分布式计算环境中实现。在分布式计算环境中,使用通过通信网络连接的远程处理设备来执行任务。在分布式计算环境中,程序模块可以位于本地和远程计算机存储介质中,包括存储设备。

[0320] 应注意,本文中诸如“第一”和“第二”的关系用语仅用于对实体或操作与另一实体或操作进行区分,并不要求或暗示这些实体或操作之间的任何实际关系或顺序。此外,“包括(comprise)”、“具有”、“包含”和“包括(include)”和其它类似形式的词语旨在含义上是等效的并且是开放式的,因为这些词语中的任何一个词语之后的一个或多个项不旨在详尽列出这样的项或多个项,也不旨在仅限于列出的项或多个项。

[0321] 在前述说明书中,已经参考可以随实现方式而变化的许多特定细节描述了实施方式。可以对所描述的实施方式进行某些调整和修改。考虑到本文公开的本公开的说明书和实践,其它实施方式对于本领域技术人员来说是显而易见的。说明书和示例旨在仅被认为是示例,本公开的真正范围和精神由以下权利要求书指示。图中所示的步骤顺序也旨在仅用于例示目的,并不旨在限于任何特定的步骤顺序。因此,本领域技术人员可以理解,在实现同一方法时,可以以不同顺序执行这些步骤。

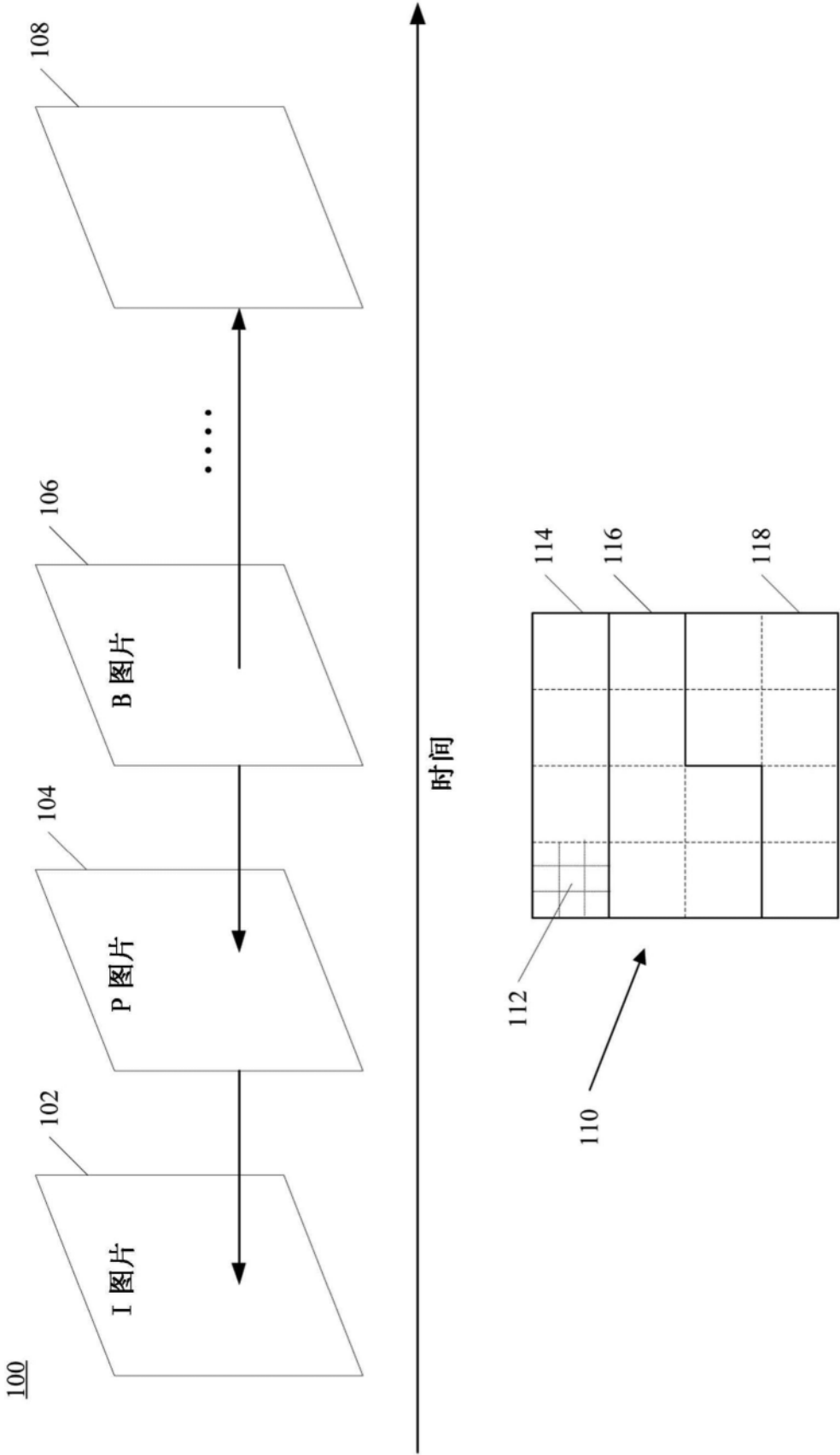


图1

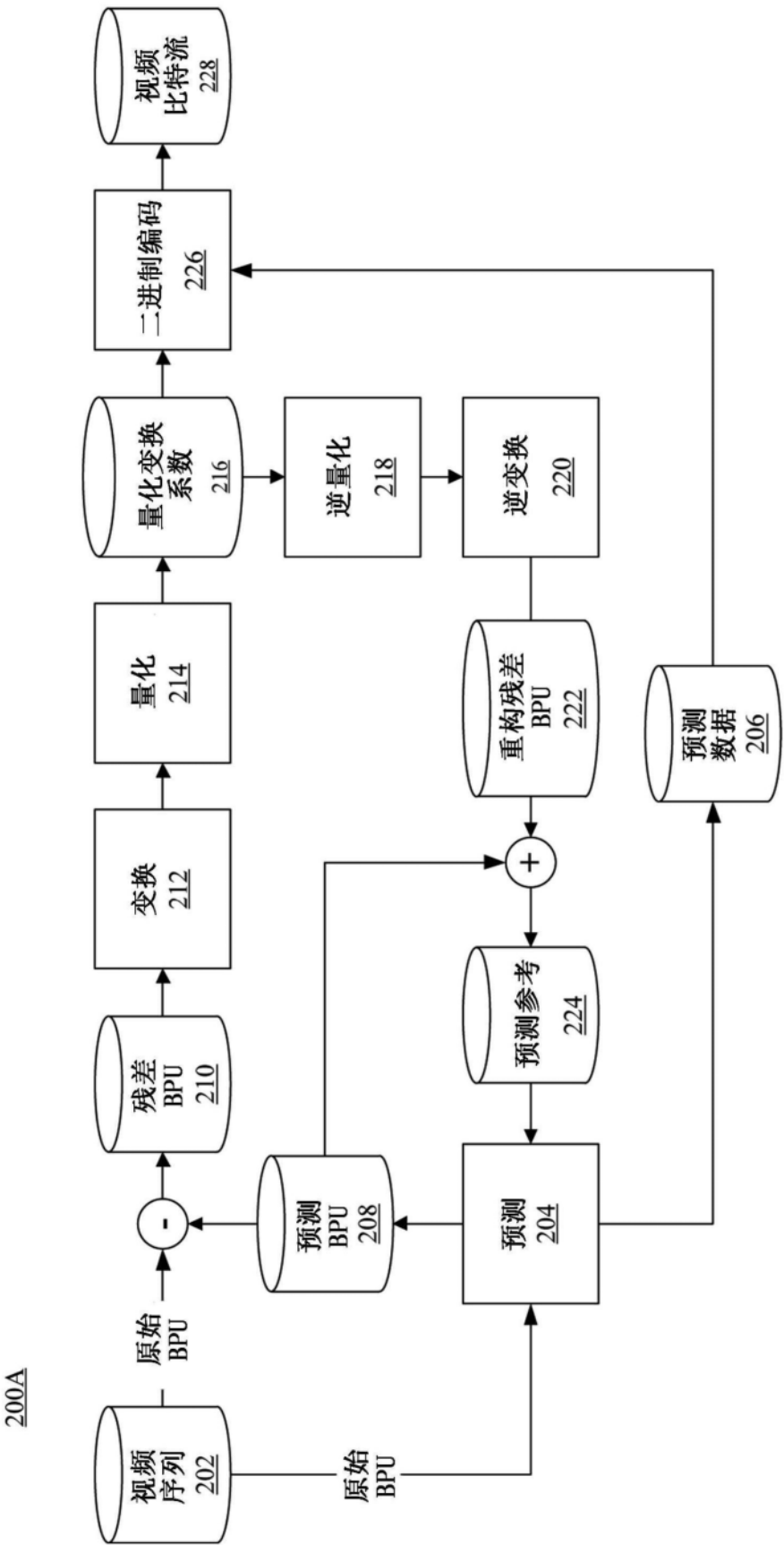


图2A

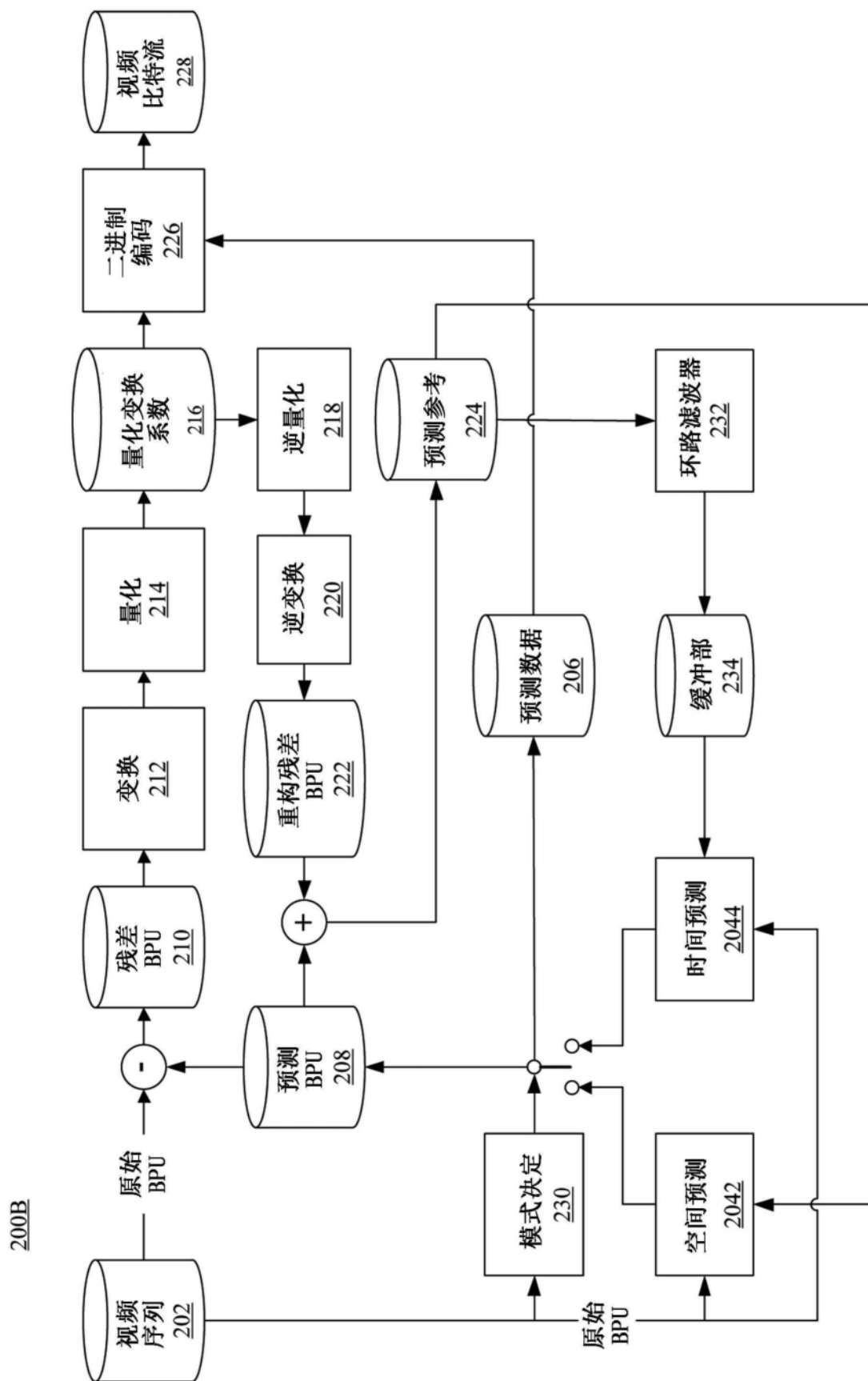


图2B

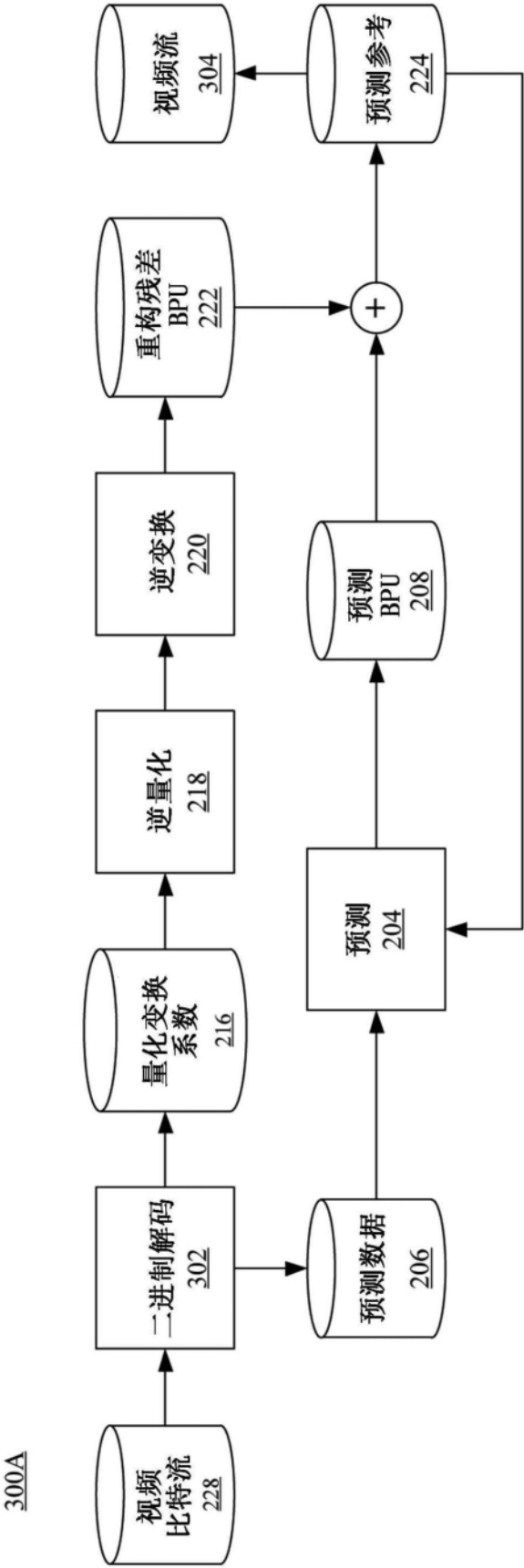


图3A

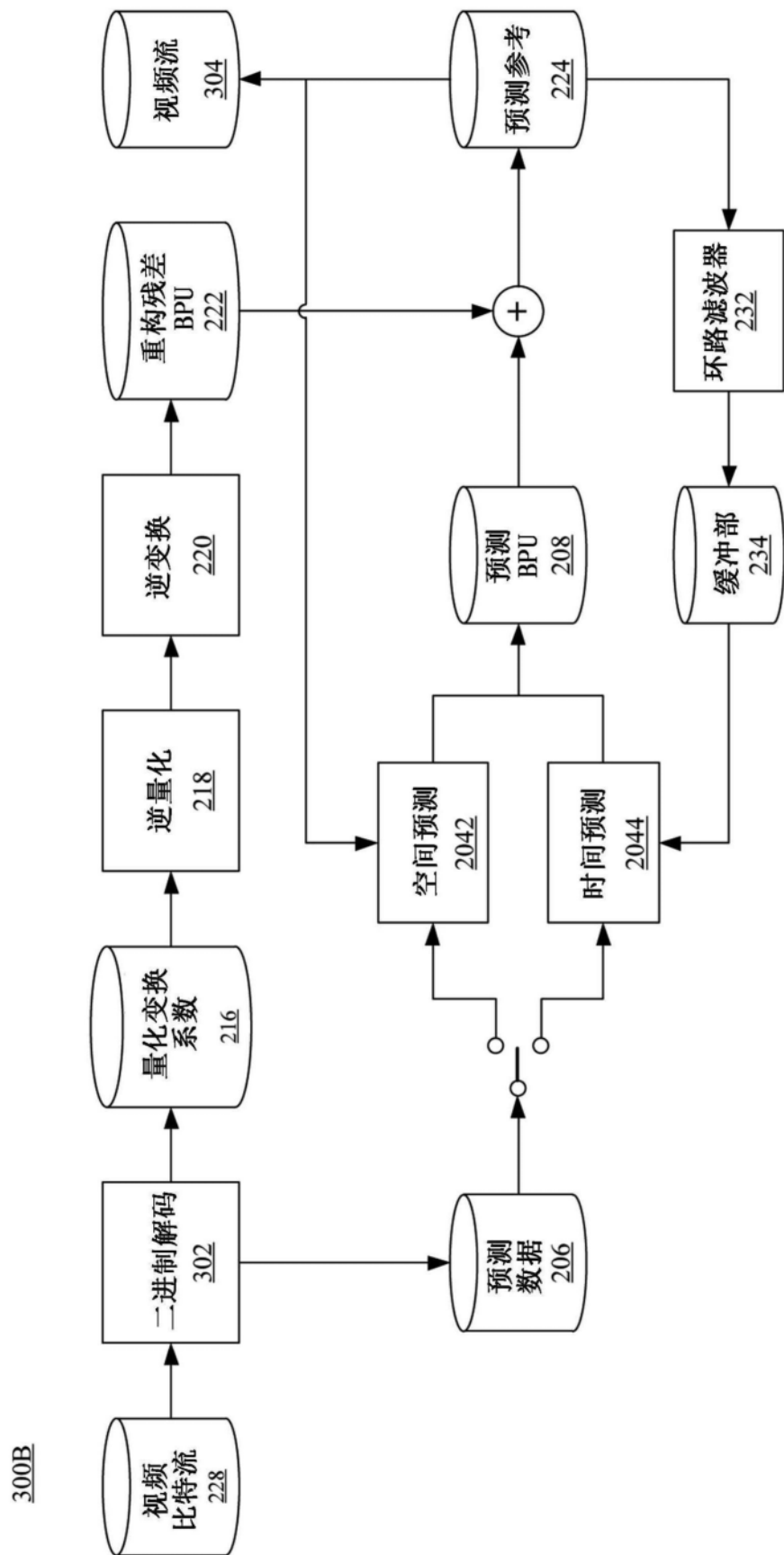


图3B

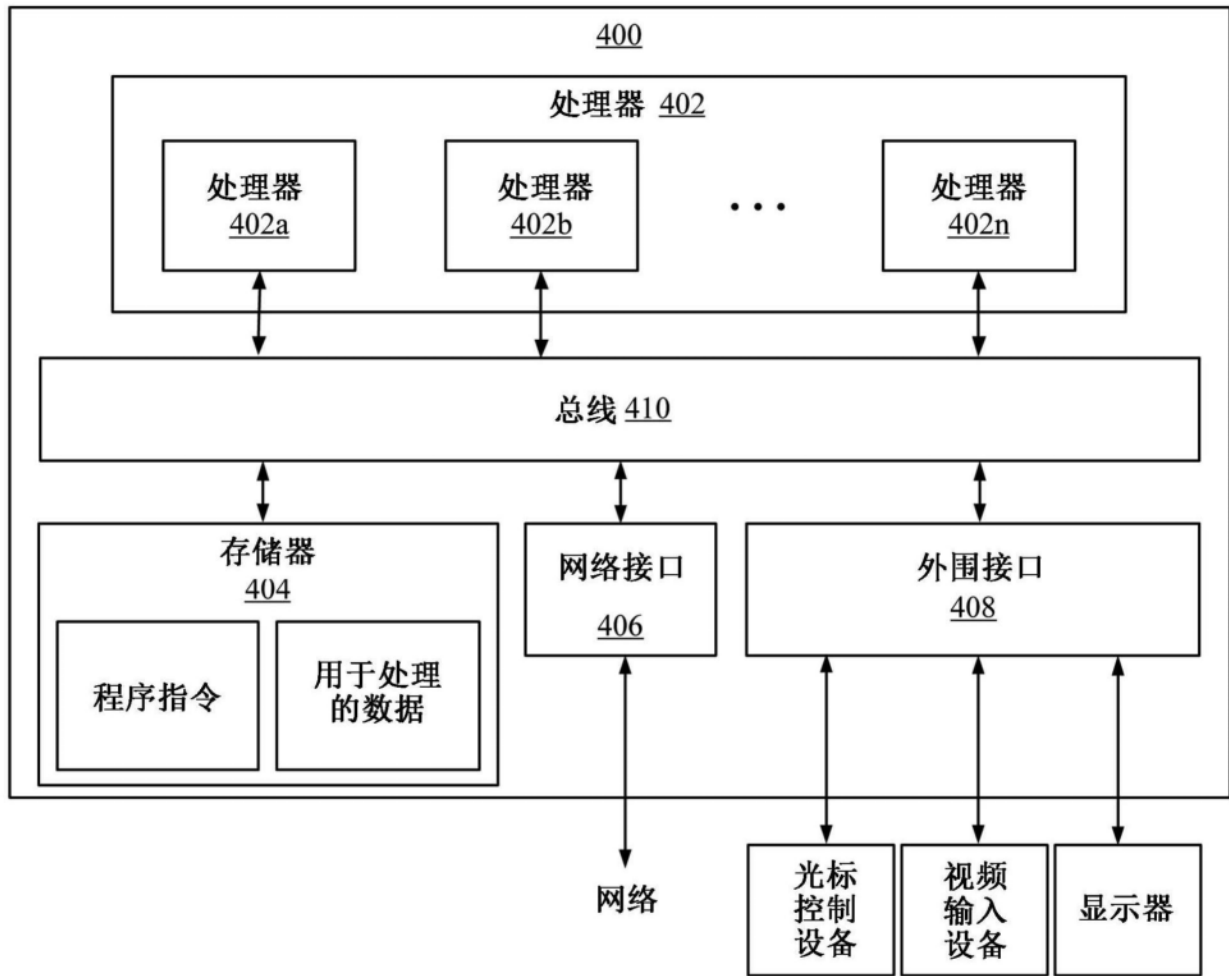
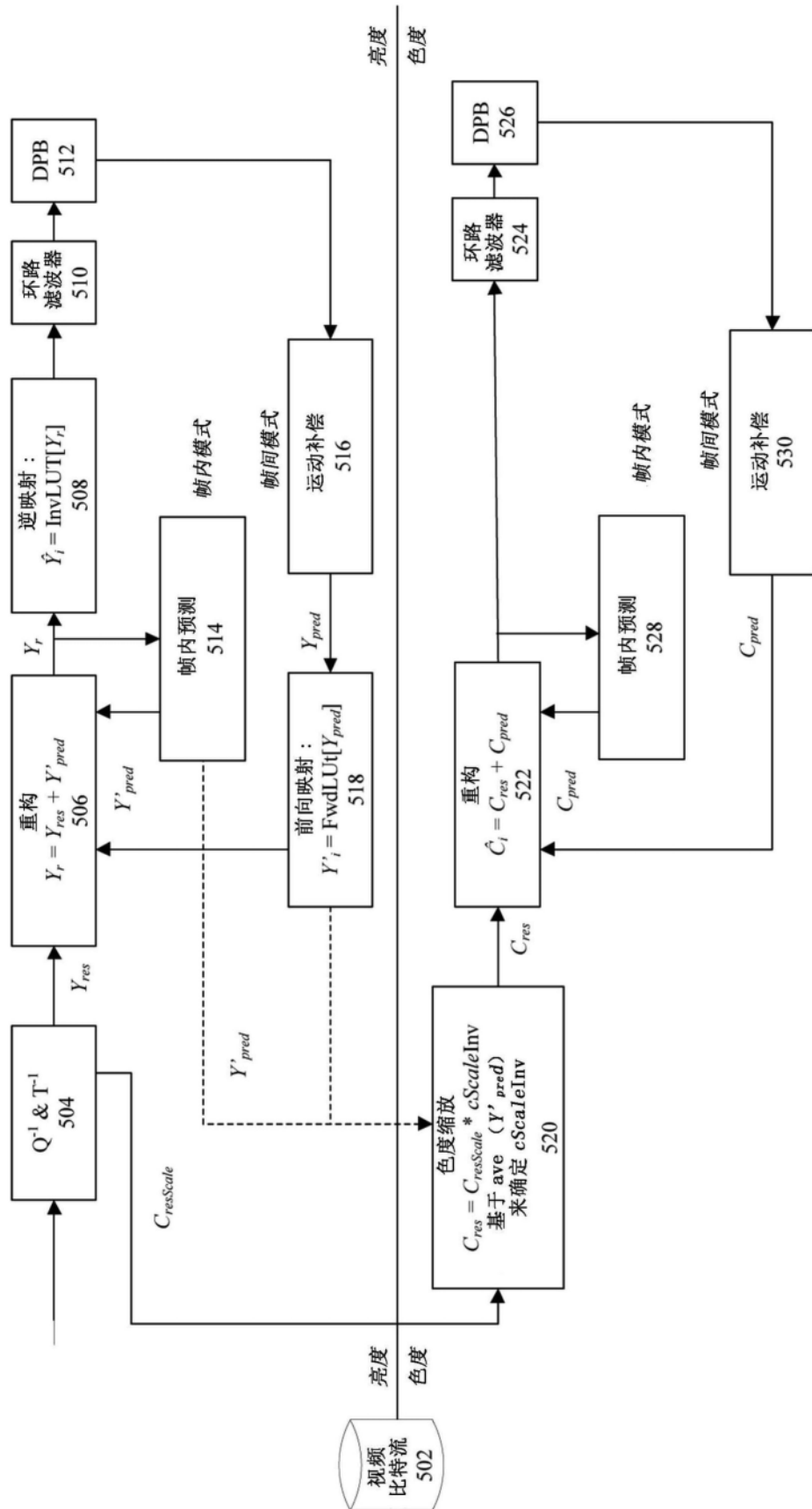


图4





500

图5

表 1

tile_group_header() {	描述符
...	
if ( sps_resaper_enabled_flag ) {	
<b>tile_group_resaper_model_present_flag</b>	u(1)
if ( tile_group_resaper_model_present_flag )	
tile_group_resaper_model ( )	
<b>tile_group_resaper_enable_flag</b>	u(1)
if ( tile_group_resaper_enable_flag && !( qtbt_dual_tree_intra_flag && tile_group_type == I ) ) )	
<b>tile_group_resaper_chroma_residual_scale_flag</b>	u(1)
}	
byte_alignment( )	
}	

图6

表 2

tile_group_resaper_model ( ) {	描述符
<b>reshaper_model_min_bin_idx</b>	ue(v)
<b>reshaper_model_delta_max_bin_idx</b>	ue(v)
<b>reshaper_model_bin_delta_abs_cw_prec_minus1</b>	ue(v)
for ( i = reshaper_model_min_bin_idx; i <= reshaper_model_max_bin_idx; i++ ) {	
<b>reshape_model_bin_delta_abs_CW [ i ]</b>	u(v)
if ( reshaper_model_bin_delta_abs_CW[ i ] > 0 )	
<b>reshaper_model_bin_delta_sign_CW_flag[ i ]</b>	u(1)
}	
}	

图7

表 3

coding_tree_unit( ) {	描述符
xCtb = ( CtbAddrInRs % PicWidthInCtbsY ) << CtbLog2SizeY	
yCtb = ( CtbAddrInRs / PicWidthInCtbsY ) << CtbLog2SizeY	
if( tile_group_sao_luma_flag    tile_group_sao_chroma_flag )	
sao( xCtb >> CtbLog2SizeY, yCtb >> CtbLog2SizeY )	
if( tile_group_alf_enabled_flag ){	
<b>alf_ctb_flag</b> [ 0 ][ xCtb >> Log2CtbSize ][ yCtb >> Log2CtbSize ]	ae(v)
if( alf_chroma_idc == 1    alf_chroma_idc == 3 )	
<b>alf_ctb_flag</b> [ 1 ][ xCtb >> Log2CtbSize ][ yCtb >> Log2CtbSize ]	ae(v)
if( alf_chroma_idc == 2    alf_chroma_idc == 3 )	
<b>alf_ctb_flag</b> [ 2 ][ xCtb >> Log2CtbSize ][ yCtb >> Log2CtbSize ]	ae(v)
}	
if( ( tile_group_type == I    CurrPicIsOnlyRef ) && qtbtt_dual_tree_intra_flag )	
dual_tree_implicit_qt_split ( xCtb, yCtb, CtbLog2SizeY, 0 )	
else	
coding_quadtree( xCtb, yCtb, CtbLog2SizeY, 0, SINGLE_TREE )	
}	

图8

表 4

dual_tree_implicit_qt_split( x0, y0, log2CbSize, cqtDepth ) {	描述符
if( log2CbSize > 6 ) {	
x1 = x0 + ( 1 << ( log2CbSize - 1 ) )	
y1 = y0 + ( 1 << ( log2CbSize - 1 ) )	
dual_tree_implicit_qt_split( x0, y0, log2CbSize - 1, cqtDepth + 1 )	
if( x1 < pic_width_in_luma_samples )	
dual_tree_implicit_qt_split( x1, y0, log2CbSize - 1, cqtDepth + 1 )	
if( y1 < pic_height_in_luma_samples )	
dual_tree_implicit_qt_split( x0, y1, log2CbSize - 1, cqtDepth + 1 )	
if( x1 < pic_width_in_luma_samples && y1 < pic_height_in_luma_samples )	
dual_tree_implicit_qt_split( x1, y1, log2CbSize - 1, cqtDepth + 1 )	
} else {	
coding_quadtree( x0, y0, log2CbSize, cqtDepth, DUAL_TREE_LUMA )	
coding_quadtree( x0, y0, log2CbSize, cqtDepth, DUAL_TREE_CHROMA )	
}	
}	

图9

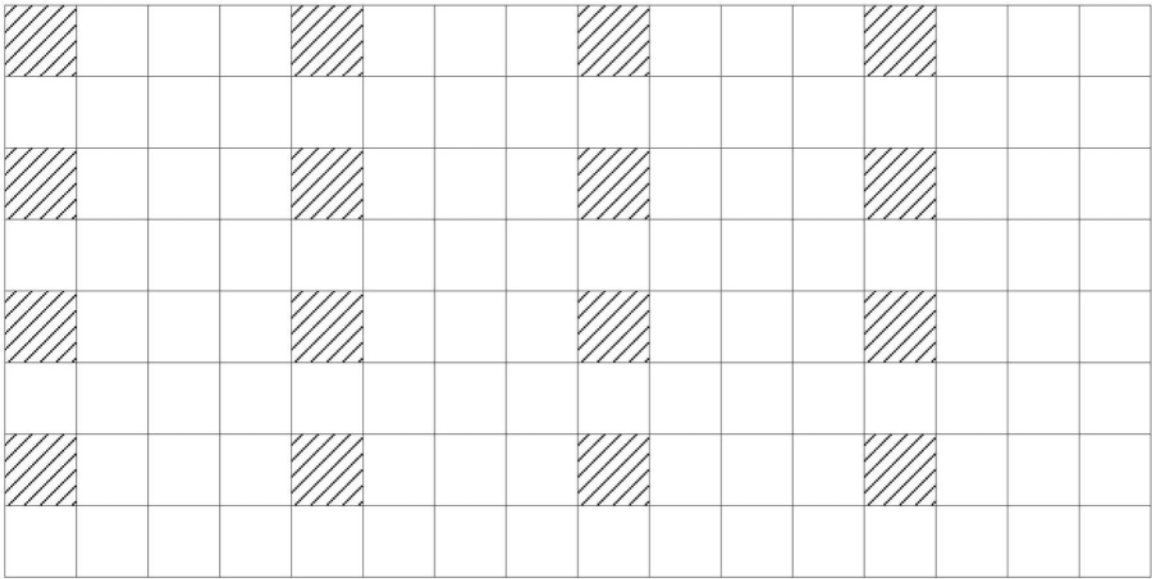


图10

表 5

coding_unit( x0, y0, cbWidth, cbHeight, treeType ) {	描述符
...	
if( CuPredMode[ x0 ][ y0 ] == MODE_INTRA ) {	
....	
if( treeType == SINGLE_TREE    treeType == DUAL_TREE_CHROMA )	
{	
<b>intra_chroma_pred_mode[ x0 ][ y0 ]</b>	ae(v)
<i>if ( tile_group_resaper_chroma_residual_scale_flag )</i>	
<b>lmcs_chroma_scaling_idx[ x0 ][ y0 ]</b>	ae(v)
}	
}	
....	

图11

表 6

tile_group_reshaper_model () {	描述符
reshaper_model_min_bin_idx	ue(v)
reshaper_model_delta_max_bin_idx	ue(v)
reshaper_model_bin_delta_abs_cw_prec_minus1	ue(v)
reshaper_model_bin_num_unique_cw_minus1	ue(v)
for ( i = 0; i <= reshaper_model_bin_num_unique_cw_minus1; i++ ) {	
reshape_model_bin_delta_abs_CW [ i ]	u(v)
if ( reshaper_model_bin_delta_abs_CW [ i ] > 0 )	
reshaper_model_bin_delta_sign_CW_flag [ i ]	u(1)
}	
for ( i = reshaper_model_min_bin_idx; i <= reshaper_model_max_bin_idx; i++ )	
reshaper_model_bin_cw_idx [ i ]	u(v)
}	

图12

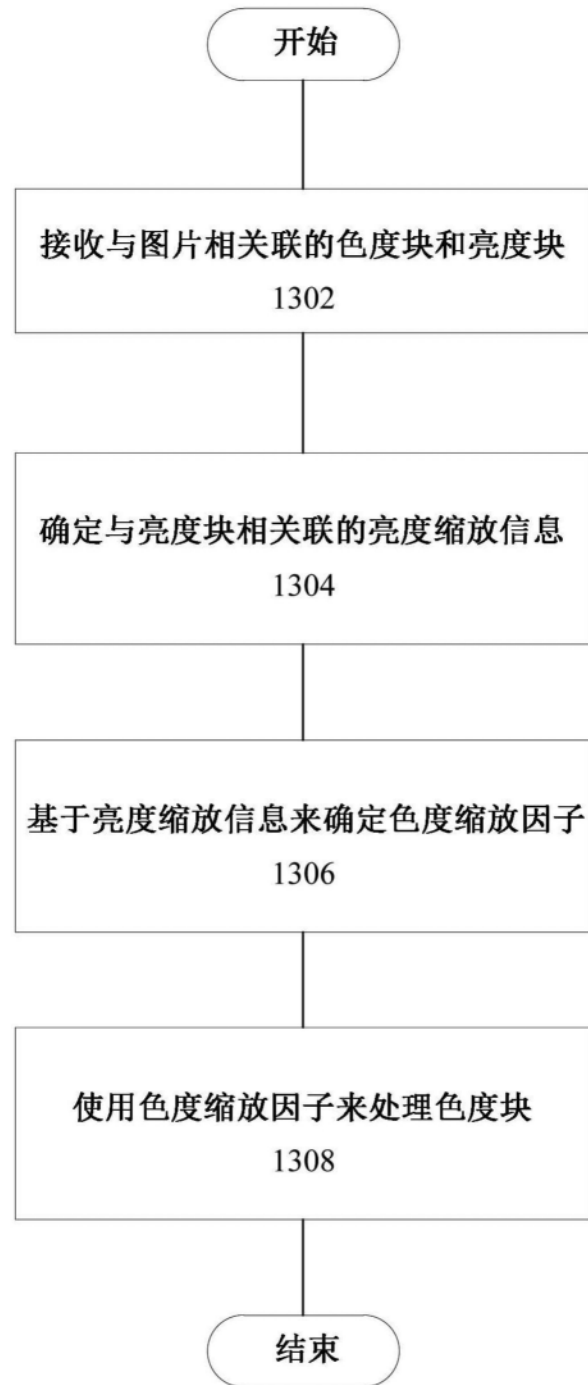
**1300**

图13

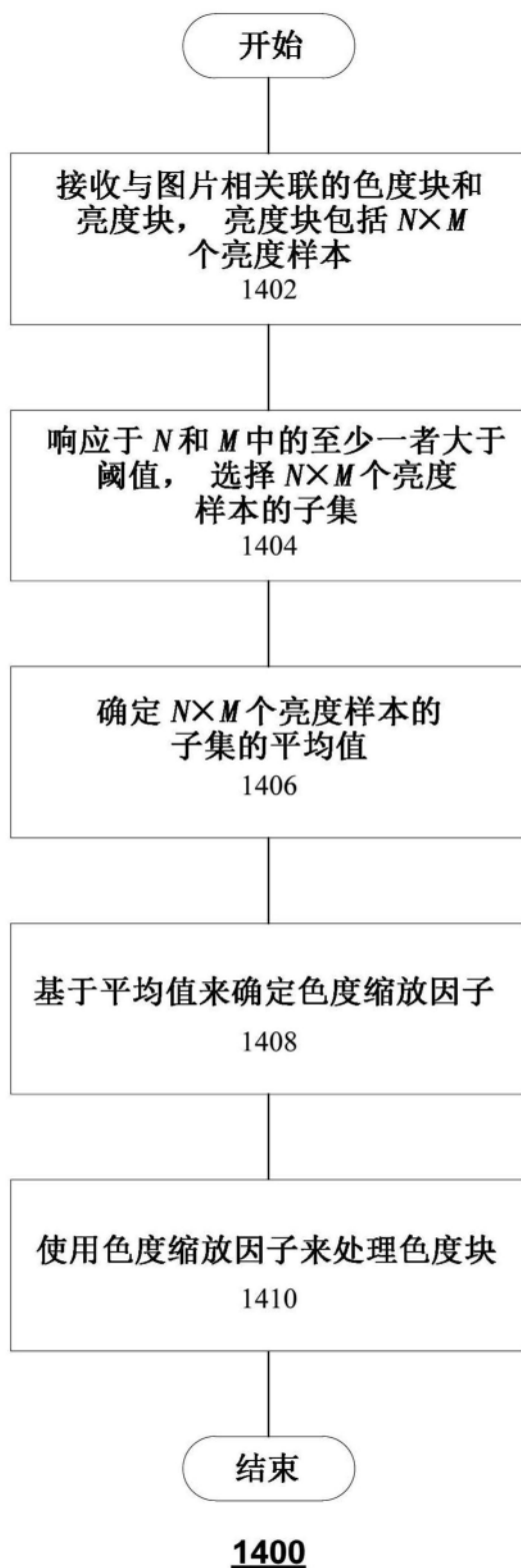


图14

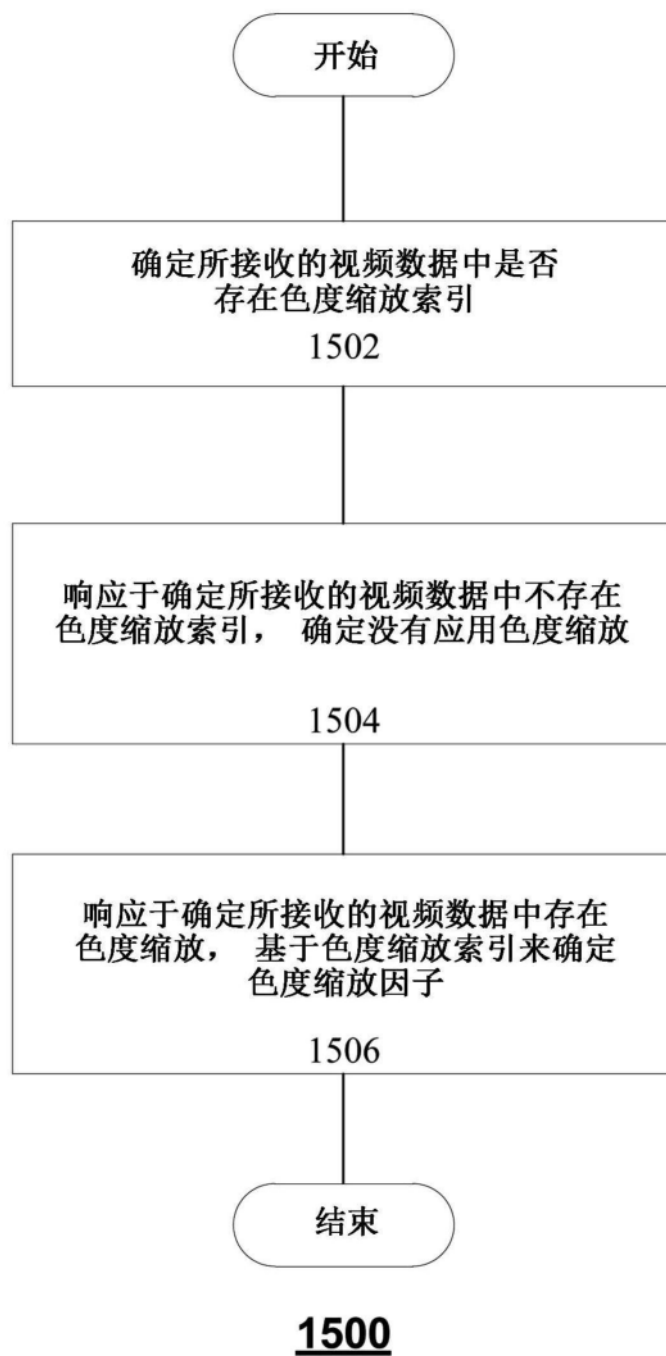


图15



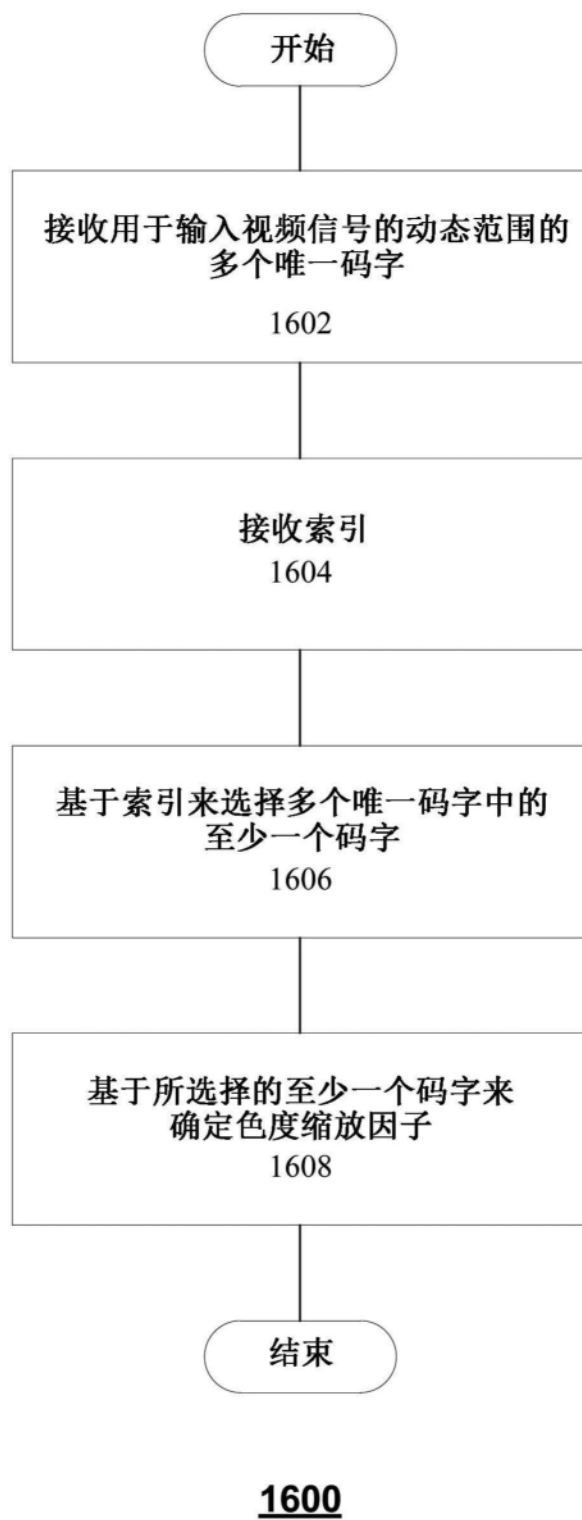


图16