(19)国家知识产权局

(12)发明专利申请

(10)申请公布号 CN 119996695 A
(43)申请公布日 2025.05.13

(21)申请号 202510171177.6

(22)申请日 2020.06.16

(30)优先权数据
19305802.1 2019.06.21 EP
19305826.0 2019.06.24 EP

(62)分案原申请数据
202080045038.4 2020.06.16

(71)申请人 交互数字CE专利控股有限公司
地址 法国巴黎

(72)发明人 P·德拉格朗日 F·莱纳内克
K·纳赛尔 E·弗兰科伊斯
P·安德鲁万 陈娅

(74)专利代理机构 北京润平知识产权代理有限公司 11283
专利代理师 肖冰滨

(51)Int.Cl.
H04N 19/124(2014.01)
H04N 19/14(2014.01)
H04N 19/159(2014.01)
H04N 19/176(2014.01)
H04N 19/186(2014.01)
H04N 19/463(2014.01)
H04N 19/70(2014.01)

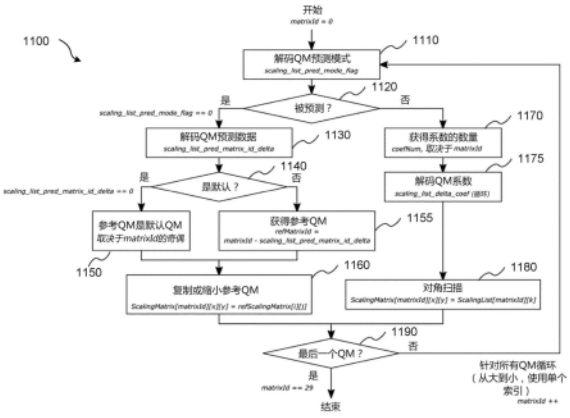
权利要求书2页 说明书24页 附图9页

(54)发明名称

用于视频编码和解码的单索引量化矩阵设计

(57)摘要

本发明涉及用于视频编码和解码的单索引量化矩阵设计,具体涉及可对应于不同块尺寸、颜色分量和预测模式而传送不同量化矩阵。为了更有效地用信号通知量化矩阵的系数,在一个实现中,基于与首先列出的较大尺寸的CU尺寸有关的尺寸标识符(sizeId)和首先列出的亮度QM的矩阵类型(matrixTypeId),使用统一矩阵标识符matrixId。例如,统一标识符被推导为:matrixId=N*sizeId+matrixTypeId,其中N是可能的类型标识符的数量,例如,N=6。该单个标识符允许在使用预测(复制)时参考任何先前传送的矩阵,并且首先传送较大的矩阵避免预测过程中的插值。当块使用帧内块复制预测模式时,可以得到QM标识符,就像该块使用帧间预测模式一样。



CN 119996695 A

1.一种方法,包括:

基于图片中的要解码的块的块大小、颜色分量和预测模式,获得用于量化矩阵的单个标识符,其中,当获得所述单个标识符时,帧内块复制预测模式被认为是帧间预测模式;

解码指示参考量化矩阵的语法元素,其中所述语法元素指定所述参考量化矩阵的标识符和所述量化矩阵的所述单个标识符之间的差;

基于所述参考量化矩阵,获得所述量化矩阵;

基于所述量化矩阵,对所述块的变换系数进行去量化;以及

基于所述去量化的变换系数,对所述块进行解码。

2.根据权利要求1所述的方法,其中,所述单个标识符是基于所述块的所述预测模式是帧内预测模式还是帧间预测模式而被获得的。

3.根据权利要求1所述的方法,针对所述块的亮度分量用信号发送量化矩阵,且通过将所述预测模式视为帧间预测模式而针对色度分量导出量化矩阵。

4.根据权利要求1所述的方法,其中,通过将所述预测模式视为帧内模式来获得另一参考量化矩阵,以及其中所述量化矩阵是作为所述参考量化矩阵和所述另一参考量化矩阵的平均来获得的。

5.一种装置,包括一个或多个处理器,其中所述一个或多个处理器被配置为:

基于图片中的待解码的块的块大小、颜色分量和预测模式,获得量化矩阵的单个标识符,其中,当获得所述单个标识符时,帧内块复制预测模式被认为是帧间预测模式;

解码指示参考量化矩阵的语法元素,其中所述语法元素指定所述参考量化矩阵的标识符与所述量化矩阵的所述单个标识符之间的差;

基于所述参考量化矩阵,获得所述量化矩阵;

基于所述量化矩阵,去量化所述块的变换系数;以及

基于所述去量化的变换系数,对所述块进行解码。

6.根据权利要求5所述的装置,其中,所述单个标识符是基于所述块的所述预测模式是帧内预测模式还是帧间预测模式而被获得的。

7.根据权利要求5所述的装置,针对所述块的亮度分量用信号表示量化矩阵,且通过将所述预测模式视为帧间预测模式而针对色度分量导出量化矩阵。

8.根据权利要求5所述的装置,其中,通过将所述预测模式视为帧内模式来获得另一参考量化矩阵,以及其中所述量化矩阵是作为所述参考量化矩阵和所述另一参考量化矩阵的平均来获得的。

9.一种方法,包括:

访问用于图片中的要被编码的块的量化矩阵;

基于所述块的块大小、颜色分量和预测模式,获得用于所述量化矩阵的单个标识符,其中,当获得所述单个标识符时,帧内块复制预测模式被认为是帧间预测模式;

编码指示参考量化矩阵的语法元素,其中所述语法元素指定所述参考量化矩阵的标识符和所述量化矩阵的所述单个标识符之间的差;

基于所述量化矩阵,量化所述块的变换系数;以及

对所述量化的变换系数进行熵编码。

10.根据权利要求9所述的方法,其中,所述单个标识符是基于所述块的所述预测模式

是帧内预测模式还是帧间预测模式而被获得的。

11.根据权利要求9所述的方法,针对所述块的亮度分量用信号发送量化矩阵,且通过将所述预测模式视为帧间预测模式而针对色度分量导出量化矩阵。

12.根据权利要求9所述的方法,其中,通过将所述预测模式视为帧内模式来获得另一参考量化矩阵,以及其中所述量化矩阵是作为所述参考量化矩阵和所述另一参考量化矩阵的平均来获得的。

13.一种装置,包括一个或多个处理器,其中所述一个或多个处理器被配置为:

访问用于图片中的要被编码的块的量化矩阵;

基于所述块的块大小、颜色分量和预测模式,获得所述量化矩阵的单个标识符,其中,当获得所述单个标识符时,帧内块复制预测模式被认为是帧间预测模式;

编码指示参考量化矩阵的语法元素,其中所述语法元素指定所述参考量化矩阵的标识符和所述量化矩阵的所述单个标识符之间的差;

基于所述量化矩阵,量化所述块的变换系数;以及

对所述量化的变换系数进行熵编码。

14.根据权利要求13所述的装置,其中基于所述块的所述预测模式是帧内预测模式还是帧间预测模式来获得所述单个识别符。

15.根据权利要求13所述的装置,针对所述块的亮度分量用信号表示量化矩阵,且通过将所述预测模式视为帧间预测模式而针对色度分量导出量化矩阵。

16.根据权利要求13所述的装置,其中,通过将所述预测模式视为帧内模式来获得另一参考量化矩阵,以及其中所述量化矩阵是作为所述参考量化矩阵和所述另一参考量化矩阵的平均来获得的。

用于视频编码和解码的单索引量化矩阵设计

本申请为申请日是2020年6月16日、题为“用于视频编码和解码的单索引量化矩阵设计”的中国专利申请No.202080045038.4的分案申请,该中国专利申请的内容被整体引用于此。

技术领域

[0001] 本实施例主要涉及一种视频编码或解码中的量化矩阵设计方法及装置。

背景技术

[0002] 为了实现高压缩效率,图像和视频译码方案通常采用预测和变换以利用视频内容中的空间和时间冗余。通常,使用帧内或帧间预测来利用帧内或帧间图片相关性,然后对通常表示为预测误差或预测残差的原始块与预测块之间的差进行变换、量化和熵译码。为了重构视频,通过与熵译码、量化、变换和预测相对应的逆处理来对压缩数据进行解码。

发明内容

[0003] 根据一实施例,提供一种视频解码方法,包括:基于图片中要被解码的块的块尺寸、颜色分量和预测模式来获得量化矩阵的单个标识符;解码指示参考量化矩阵的语法元素,其中所述语法元素指定所述参考量化矩阵的标识符与所述量化矩阵的所述获得的标识符之间的差;基于所述参考量化矩阵获得所述量化矩阵;响应于所述量化矩阵,对所述块的变换系数进行去量化;以及响应于所述去量化的变换系数来解码所述块。

[0004] 根据另一实施例,提供了一种用于视频编码的方法,包括:访问图片中要被编码的块;访问所述块的量化矩阵;基于所述块的块尺寸、颜色分量及预测模式获得所述量化矩阵的单个标识符;编码指示参考量化矩阵的语法元素,其中所述语法元素指定所述参考量化矩阵的标识符与所述量化矩阵的所述获得的标识符之间的差;响应于所述量化矩阵来量化所述块的变换系数;以及对所述量化的变换系数进行熵编码。

[0005] 根据另一实施例,提供一种用于视频解码的设备,其包括一或多个处理器,其中所述一或多个处理器被配置用于:基于图片中要被解码的块的块尺寸、颜色分量和预测模式来获得用于量化矩阵的单个标识符;解码指示参考量化矩阵的语法元素,其中所述语法元素指定所述参考量化矩阵的标识符与所述量化矩阵的所述获得的标识符之间的差;基于所述参考量化矩阵获得所述量化矩阵;响应于所述量化矩阵,对所述块的变换系数进行去量化;以及响应于所述去量化的变换系数来解码所述块。

[0006] 根据另一实施例,提供一种用于视频编码的设备,其包括一或多个处理器,其中所述一或多个处理器被配置用于:访问图片中的要被编码的块;访问所述块的量化矩阵;基于所述块的块尺寸、颜色分量及预测模式获得所述量化矩阵的单个标识符;编码指示参考量化矩阵的语法元素,其中所述语法元素指定所述参考量化矩阵的标识符和所述量化矩阵的所述获得的标识符之间的差;响应于所述量化矩阵来量化所述块的变换系数;以及对所述量化的变换系数进行熵编码。

[0007] 根据另一实施例,提供了一种视频解码的设备,包括:用于基于图片中要被解码的块的块尺寸、颜色分量和预测模式获得量化矩阵的单个标识符的装置;用于解码指示参考量化矩阵的语法元素的装置,其中所述语法元素指定所述参考量化矩阵的标识符和所述量化矩阵的所述获得的标识符之间的差;用于基于所述参考量化矩阵获得所述量化矩阵的装置;用于响应于所述量化矩阵而对所述块的变换系数进行去量化的装置;以及用于响应于所述去量化的变换系数来解码所述块的装置。

[0008] 根据另一实施例,提供了一种视频编码的设备,包括:用于访问图片中要被编码的块的装置;用于访问所述块的量化矩阵的装置;用于基于所述块的块尺寸、颜色分量及预测模式获得所述量化矩阵的单个标识符的装置;用于对指示参考量化矩阵的语法元素进行编码的装置,其中所述语法元素指定所述参考量化矩阵的标识符与所述量化矩阵的所述获得的标识符之间的差;用于响应于所述量化矩阵而量化所述块的变换系数的装置;以及用于熵编码所述量化的变换系数的装置。

[0009] 一个或多个实施例还提供了一种包括指令的计算机程序,当由一个或多个处理器执行所述指令时,所述指令使所述一个或多个处理器执行根据上述实施例中的任意实施例的编码方法或解码方法。本发明实施例中的一个或多个实施例还提供一种计算机可读存储媒体,其上存储有用于根据上文所描述的方法编码或解码视频数据的指令。一个或多个实施例还提供了一种计算机可读存储介质,其上存储有根据上述方法生成的比特流。一个或多个实施例还提供了一种用于传送或接收根据上述方法生成的比特流的方法和装置。

附图说明

[0010] 图1示出了可以在其中实现本实施例的各方面的系统的框图。

[0011] 图2示出了视频编码器的实施例的框图。

[0012] 图3示出了视频解码器的实施例的框图。

[0013] 图4示出了在VVC草案5中对于大于32的块尺寸将变换系数推断为零。

[0014] 图5示出了JCTVC-H0314中描述的固定预测树。

[0015] 图6示出了根据实施例的从较大尺寸进行预测(抽取)。

[0016] 图7示出了根据实施例的组合根据较大尺寸的预测和矩形块的抽取。

[0017] 图8示出了根据实施例的针对色度的矩形块的QM推导过程。

[0018] 图9示出了根据实施例的针对色度(适应于4:2:2格式)的矩形块的QM推导过程。

[0019] 图10示出了针对色度(适应于4:4:4格式)的矩形块的QM推导过程。

[0020] 图11示出了根据实施例的用于解析缩放列表数据语法结构的流程图。

[0021] 图12示出了根据实施例的用于编码缩放列表数据语法结构的流程图。

[0022] 图13示出了根据实施例的用于QM推导过程的流程图。

具体实施方式

[0023] 图1示出了其中可以实现各个方面和实施例的系统的示例的框图。系统100可以被实现为包括以下描述的各种组件的设备,并且被配置为执行本申请中描述的一个或多个方面。此类设备的示例包括但不限于各种电子设备,诸如个人计算机、膝上型计算机、智能电话、平板计算机、数字多媒体机顶盒、数字电视接收机、个人视频记录系统、连接的家用电器

和服务器。系统100的元件可以单独地或组合地在单个集成电路、多个IC和/或分立组件中实现。例如,在至少一个实施例中,系统100的处理和编码器/解码器元件分布在多个IC和/或分立组件上。在各种实施例中,系统100经由例如通信总线或通过专用输入和/或输出端口通信地耦合到其它系统或其它电子设备。在各种实施例中,系统100被配置为实现本申请中描述的一个或多个方面。

[0024] 系统100包括至少一个处理器110,其被配置为执行加载在其中的指令,以用于实现例如本申请中描述的各个方面。处理器110可以包括嵌入式存储器、输入输出接口和本领域已知的各种其它电路。系统100包括至少一个存储器120(例如,易失性存储器设备和/或非易失性存储器设备)。系统100包括存储设备140,其可以包括非易失性存储器和/或易失性存储器,包括但不限于EEPROM、ROM、PROM、RAM、DRAM、SRAM、闪存、磁盘驱动器和/或光盘驱动器。作为非限制性示例,存储设备140可以包括内部存储设备、附接的存储设备和/或网络可访问的存储设备。

[0025] 系统100包括编码器/解码器模块130,其被配置为例如处理数据以提供编码视频或解码视频,并且编码器/解码器模块130可以包括其自己的处理器和存储器。编码器/解码器模块130表示可以包括在设备中以执行编码和/或解码功能的一个或多个模块。如已知的,设备可以包括编码和解码模块中的一者或两者。另外,编码器/解码器模块130可实施为系统100的单独元件或可并入处理器110内作为如所属领域的技术人员已知的硬件与软件的组合。

[0026] 要加载到处理器110或编码器/解码器130上以执行本申请中描述的各个方面的程序代码可以存储在存储设备140中,并且随后加载到存储器120上以供处理器110执行。根据各种实施例,处理器110、存储器120、存储设备140和编码器/解码器模块130中的一者或一者以上可在执行本申请案中所描述的过程期间存储各种项目中的一者或一者以上。这些存储的项可以包括但不限于输入视频、解码视频或解码视频的部分、比特流、矩阵、变量以及来自等式、公式、运算和运算逻辑的处理的中间或最终结果。

[0027] 在若干实施例中,处理器110和/或编码器/解码器模块130内部的存储器用于存储指令,且提供用于在编码或解码期间需要的处理的工作存储器。然而,在其它实施例中,处理设备外部的存储器(例如,处理设备可以是处理器110或编码器/解码器模块130)被用于这些功能中的一者或多者。外部存储器可以是存储器120和/或存储设备140,例如,动态易失性存储器和/或非易失性闪存。在几个实施例中,外部非易失性闪存用于存储电视的操作系统。在至少一个实施例中,诸如RAM的快速外部动态易失性存储器被用作用于诸如MPEG-2、HEVC或VVC的视频编码和解码操作的工作存储器。

[0028] 如框105所示,可以通过各种输入设备提供对系统100的元件的输入。这样的输入设备包括但不限于(i)接收例如由广播公司通过空中传输的RF信号的RF部分,(ii)复合输入端子,(iii)USB输入端子,和/或(iv)HDMI输入端子。

[0029] 在各种实施例中,框105的输入设备具有本领域已知的相关联的相应输入处理元件。例如,RF部分可以与适于以下的元件相关联:(i)选择期望频率(也称为选择信号,或者将信号频带限制到频带),(ii)将选择的信号下变频,(iii)再次将信号频带限制到较窄频带,以选择(例如)在某些实施例中可以称为信道的信号频带,(iv)解调下变频和频带限制的信号,(v)执行纠错,以及(vi)解复用以选择期望的数据分组流。各种实施例的RF部分包

括一个或多个元件以执行这些功能,例如,频率选择器、信号选择器、限带器、信道选择器、滤波器、下变频器、解调器、纠错器和解复用器。RF部分可以包括执行各种这些功能的调谐器,这些功能包括例如将接收的信号下变频到较低频率(例如,中频或近基带频率)或基带。在一个机顶盒实施例中,RF部分及其相关的输入处理元件接收通过有线(例如,电缆)介质传送的RF信号,并通过滤波、下变频和再次滤波到期望的频带来执行频率选择。各种实施例重新安排上述(和其它)元件的顺序,移除这些元件中的一些,和/或添加执行类似或不同功能的其它元件。添加元件可以包括在现有元件之间插入元件,例如,插入放大器和模数转换器。在各种实施例中,RF部分包括天线。

[0030] 另外,USB和/或HDMI终端可以包括用于通过USB和/或HDMI连接将系统100连接到其它电子设备的相应接口处理器。应当理解,输入处理的各个方面,例如Reed-Solomon纠错,可以根据需要在例如单独的输入处理IC或处理器110中实现。类似地,USB或HDMI接口处理的各方面可根据需要在单独的接口IC内或在处理器110内实现。解调、纠错和解复用的流被提供给各种处理元件,包括例如处理器110和编码器/解码器130,其与存储器和存储元件结合操作以根据需要处理数据流以便在输出设备上呈现。

[0031] 系统100的各种元件可设置在集成壳体内,各种元件可使用合适的连接布置115例如本领域已知的内部总线包括I2C总线、布线和印刷电路板互连并在其间传输数据。

[0032] 系统100包括通信接口150,其使得能够经由通信信道190与其它设备通信。通信接口150可以包括但不限于被配置为通过通信信道190传送和接收数据的收发器。通信接口150可以包括但不限于调制解调器或网卡,并且通信信道190可以例如在有线和/或无线介质内实现。

[0033] 在各种实施例中,使用诸如IEEE 802.11的Wi-Fi网络将数据流式传输到系统100。这些实施例的Wi-Fi信号通过适用于Wi-Fi通信的通信信道190和通信接口150被接收。这些实施例的通信信道190通常连接到接入点或路由器,所述接入点或路由器提供对包括因特网的外部网络的接入以允许流式应用和其它过顶通信。其它实施例使用机顶盒向系统100提供流数据,该机顶盒通过输入框105的HDMI连接传递数据。还有一些实施例使用输入框105的RF连接向系统100提供流数据。

[0034] 系统100可以向各种输出设备提供输出信号,包括显示器165、扬声器175和其他外围设备185。在各种实施例的示例中,其他外围设备185包括独立DVR、盘播放器、立体声系统、照明系统和基于系统100的输出提供功能的其他设备中的一者或多者。在各种实施例中,使用诸如AV.链路的信令、CEC或使得能够在有或没有用户干预的情况下进行设备到设备控制的其他通信协议在系统100和显示器165、扬声器175或其它外围设备185之间传送控制信号。输出设备可以经由通过相应接口160、170和180的专用连接通信地耦合到系统100。或者,输出设备可以使用通信信道190经由通信接口150连接到系统100。显示器165和扬声器175可以与系统100的其它组件一起集成在电子设备(例如电视)中的单个单元中。在各种实施例中,显示接口160包括显示驱动器,例如,定时控制器(T Con)芯片。

[0035] 例如,如果输入105的RF部分是单独机顶盒的一部分,则显示器165和扬声器175可以备选地与其他组件中的一个或多个分离。在显示器165和扬声器175是外部组件的各种实施例中,输出信号可以经由专用输出连接来提供,所述专用输出连接例如包括HDMI端口、USB端口或COMP输出。

[0036] 图2示出了示例视频编码器200,例如高效视频编码(HEVC)编码器。图2还示出了其中对HEVC标准作出改进的编码器或采用类似于HEVC的技术的编码器,例如JVCT(联合视频探索团队)开发中的VVC(通用视频译码)编码器。

[0037] 在本申请中,术语“重构”和“解码”可以互换使用,术语“编码”或“译码”可以互换使用,并且术语“图像”、“图片”和“帧”可以互换使用。通常,但不是必须的,术语“重构”在编码器侧使用,而“解码”在解码器侧使用。

[0038] 在被编码之前,视频序列可以经历预编码处理(201),例如,对输入颜色图片应用颜色变换(例如,从RGB 4:4:4到YCbCr 4:2:0的转换),或者执行输入图片分量的重新映射,以便获得对压缩更有弹性的信号分布(例如,使用颜色分量之一的直方图均衡)。元数据可以与预处理相关联,并且被附加到比特流。

[0039] 在编码器200中,如下所述,由编码器元件对图片进行编码。以例如CU为单位分割(202)并处理要编码的图片。使用例如帧内或帧间模式来编码每个单元。当在帧内模式中对单元进行编码时,其执行帧内预测(260)。在帧间模式中,执行运动估计(275)和补偿(270)。编码器决定(205)使用帧内模式或帧间模式中的哪一者来对单元进行编码,并且通过例如预测模式标志来指示帧内/帧间决定。例如,通过从原始图像块中减去(210)预测块来计算预测残差。

[0040] 然后,对预测残差进行变换(225)和量化(230)。对量化的变换系数以及运动矢量和其它语法元素进行熵译码(245)以输出比特流。编码器可以跳过变换,并直接对未变换的残差信号应用量化。编码器可以绕过变换和量化,即,直接对残差进行译码而不应用变换或量化处理。

[0041] 编码器对编码块进行解码,以提供用于进一步预测的参考。对量化变换系数进行去量化(240)和逆变换(250)以对预测残差进行解码。将解码的预测残差与预测块进行组合(255),重构图像块。环内滤波器(265)被应用于重构的图像,以执行例如解块/SAO(采样自适应偏移)滤波,从而减少编码伪像。将滤波图像存储在参考图片缓冲器(280)中。

[0042] 图3示出了示例视频解码器300的框图。在解码器300中,如下所述,由解码器元件解码比特流。视频解码器300通常执行与图2中所描述的编码回合互逆的解码回合,编码器200通常还执行视频解码作为编码视频数据的一部分。

[0043] 特别地,解码器的输入包括视频比特流,其可以由视频编码器200生成。比特流首先被熵解码(330)以获得变换系数、运动矢量和其它译码信息。图片分割信息指示图片如何被分割。解码器因此可以根据解码的图片分割信息来划分(335)图片。变换系数被去量化(340)和逆变换(350)以解码预测残差。将解码的预测残差与预测块进行组合(355),重构图像块。预测块可以从帧内预测(360)或运动补偿预测(即,帧间预测)(375)获得(370)。将环内滤波器(365)应用于重构图像。将滤波图像存储在参考图片缓冲器(380)中。

[0044] 解码后的图像可以进一步经过后解码处理(385),例如,逆颜色变换(例如,从YCbCr 4:2:0到RGB 4:4:4的转换)或执行在预编码处理(201)中执行的重新映射处理的逆重新映射。后解码处理可以使用在预编码处理中推导并且在比特流中用信号通知的元数据。

[0045] HEVC规范允许在去量化过程中使用量化矩阵,其中变换系数由当前量化步长缩放,且进一步由量化矩阵(QM)缩放,如下:

d[x][y]＝Clip3(coeffMin,coeffMax,((TransCoeffLevel[xTbY][yTbY][cIdx][x][y]*m[x][y]*levelScale[qP%6]<<(qP/6))+(1<<(bdShift-1)))>>bdShift)

其中：

●TransCoefffflevel[…]是由当前块的空间坐标xTbY、yTbY及其分量索引cIdx所标识的当前块的变换系数绝对值。

●x和y是水平/垂直频率索引。

●qP是当前量化参数。

●乘以levelScale[qP%6]并左移(qP/6)等效于乘以量化步长qStep＝(levelScale[qP%6]<<(qP/6))。

●m[…][…]是二维量化矩阵。这里,因为量化矩阵用于缩放,所以它也可以被称为缩放矩阵。

●bdShift是考虑图像采样比特深度的附加缩放因子。项1<<(bdShift-1))用于舍入到最接近的整数的目的。

●d[…]是所得的去量化的变换系数绝对值。

[0046] HEVC用来传输量化矩阵的语法描述如下：

scaling_list_data( ) {	描述符
for(sizeId = 0; sizeId < 4; sizeId++)	
for(matrixId = 0; matrixId < 6; matrixId += (sizeId = = 3) ? 3 : 1) {	
scaling_list_pred_mode_flag[sizeId][matrixId]	u(1)
if(!scaling_list_pred_mode_flag[sizeId][matrixId])	
scaling_list_pred_matrix_id_delta[sizeId][matrixId]	ue(v)
else {	
nextCoef = 8	
coefNum = Min(64, (1 << (4 + (sizeId << 1))))	
if(sizeId > 1) {	
scaling_list_dc_coef_minus8[sizeId − 2][matrixId]	se(v)
nextCoef = scaling_list_dc_coef_minus8[sizeId − 2][matrixId] + 8	
}	
for(i = 0; i < coefNum; i++) {	
scaling_list_delta_coef	se(v)
nextCoef = (nextCoef + scaling_list_delta_coef + 256) % 256	
ScalingList[sizeId][matrixId][i] = nextCoef	
}	
}	
}	
}	

[0047] 可以注意到

●为每个变换尺寸(sizeId)指定不同的矩阵。在缩放列表数据语法结构中,缩放矩阵被扫描到1-D缩放列表(例如,ScalingList)中。

●对于给定的变换尺寸,为帧内/帧间译码和Y/Cb/Cr分量指定六个矩阵。

●矩阵可以是以下任一种

○如果scaling_list_pred_mode_flag为零(参考matrixId作为matrixId-scaling_list_pred_matrix_id_delta获得),则从相同尺寸的先前传送的矩阵复制矩阵

○根据标准中指定的默认值复制矩阵(如果scaling_list_pred_mode_flag和scaling_list_pred_matrix_id_delta两者均为零)

○使用exp-Golomb熵译码,以右上对角扫描顺序在DPCM译码模式中完全指定。

●对于大于8×8的块尺寸,仅传送8×8系数用于用信号通知量化矩阵以便节省译码比特。然后,使用零保持(即,重复)来内插系数,除了明确地传送的DC系数之外。

[0048] 在VVC草案5中,已经基于JVET-N0847的贡献采用了类似于HEVC的量化矩阵的使用(参见O.Chubach等人的“CE7相关:用于VVC的量化矩阵的支持(CE7-related:Support of quantization matrices for VVC)”,JVET-N0847,Geneva,CH,2019年3月)。scaling_list_data语法已经被适配到VVC编解码器,如下所示。

scaling_list_data( ) {	描述符
for(sizeId = 1; sizeId < 7; sizeId++)	
for(matrixId = 0; matrixId < 6; matrixId ++) {	
if(! (((sizeId == 1) && (matrixId % 3 == 0)) || ((sizeId == 6) && (matrixId % 3 != 0)))) {	
scaling_list_pred_mode_flag[sizeId][matrixId]	u(1)
if(!scaling_list_pred_mode_flag[sizeId][matrixId])	
scaling_list_pred_matrix_id_delta[sizeId][matrixId]	ue(v)
else {	
nextCoef = 8	
coefNum = Min(64, (1 << (sizeId << 1)))	
if(sizeId > 3) {	
scaling_list_dc_coef_minus8[sizeId − 4][matrixId]	se(v)
nextCoef = scaling_list_dc_coef_minus8[sizeId − 4][matrixId] + 8	
}	
for(i = 0; i < coefNum; i++) {	
x = DiagScanOrder[3][3][i][0]	
y = DiagScanOrder[3][3][i][1]	
if (!(sizeId==6 && x>=4 && y>=4)) {	
scaling_list_delta_coef	se(v)
nextCoef = (nextCoef + scaling_list_delta_coef + 256) % 256	
ScalingList[sizeId][matrixId][i] = nextCoef	
}	
}	
}	
}	
}	
}	

[0049] 在采用JVET-N0847的VVC草案5的设计中,如在HEVC中,QM由两个参数matrixId和sizeId来标识。这在以下两个表中示出。

表1:块尺寸标识符(JVET-N0847)

亮度	色度	sizeId
-	-	0
-	2x2	1
4x4	4x4	2
8x8	8x8	3
16x16	16x16	4
32x32	32x32	5
64x64	-	6

表2:QM类型标识符(JVET-N0847)

CuPredMode	cIdx(颜色分量)	matrixId
MODE_INTRA	0(Y)	0
MODE_INTRA	1(Cb)	1
MODE_INTRA	2(Cr)	2
MODE_INTER	0(Y)	3
MODE_INTER	1(Cb)	4
MODE_INTER	2(Cr)	5

注:MODE_INTRA QM也用于MODE_IBC(帧内块复制)。

[0050] 两种标识符的组合在下表中示出:

表3:(MatrixId,SizeId)组合(JVET-N0847)

帧内	Y		0,2	0,3	0,4	0,5	0,6
	Cb	1,1	1,2	1,3	1,4	1,5	
	Cr	2,1	2,2	2,3	2,4	2,5	
帧间	Y		3,2	3,3	3,4	3,5	3,6
	Cb	4,1	4,2	4,3	4,4	4,5	
	Cr	5,1	5,2	5,3	5,4	5,5	
块尺寸: max(宽度，高度)		2	4	8	16	32	64
用信号发送的QM系数		2x2	4x4	8x8	8x8 + DC		

[0051] 如在HEVC中,对于大于8×8的块尺寸,仅传送8×8系数和DC系数。使用零保持内插来重构正确尺寸的QM。例如,对于16×16块,每个系数在两个方向上重复两次,然后DC系数被传送的系数代替。

[0052] 对于矩形块,为QM选择保留的尺寸(sizeId)是较大的尺寸,即宽度和高度的最大值。例如,对于4×16块,选择16×16块尺寸的QM。然后,以因子4垂直抽取重构的16×16矩阵,以获得最终的4×16量化矩阵(即,4行中的3行被跳过)。

[0053] 对于以下内容,我们将给定系列的块尺寸(正方形或矩形)的QM称为尺寸-N,其与sizeId及其所用于的正方形块尺寸有关。例如,对于块尺寸16x16或16x4,QM被标识为尺寸-16(VVC草案5中的sizeId 4)。尺寸-N的符号用于区分确切的块形状,以及区分用信号发送的QM系数的数量(限制为8×8,如表3所示)。

[0054] 此外,在VVC草案5中,对于尺寸-64的情况,不传送右下象限的QM系数(将它们推断为0,以下将其称为“零输出”)。这是通过scaling_list_data语法中的“x>＝4&y>＝4”条件来实现的。这避免了传送变换/量化过程从未使用的QM系数。实际上,在VVC中,对于在任何维度(64xN,Nx64,其中N<＝64)中大于32的变换块尺寸,x/y频率坐标大于或等于32的任何变换系数不被传送并且被推断为零,因此,不需要量化矩阵系数来对其进行量化。这在图4中示出,其中阴影区域对应于推断为零的变换系数。

[0055] 与HEVC相比,由于较大数量的块尺寸,VVC需要更多的量化矩阵。然而,在VVC草案5中,QM预测仍然限于相同块尺寸矩阵的复制,这可能导致比特浪费。此外,由于在VVC中仅针对色度使用块尺寸2x2且仅针对亮度使用块尺寸64x64,因此与QMs相关的语法更复杂。此外,如在HEVC中,JVET-N0847描述了针对每个块尺寸的特定矩阵推导过程。

[0056] 在HEVC标准化期间,已经探索了一些QM预测技术,例如,在JCTVC-E073(参见J.Tanaka等人,“HEVC的量化矩阵(Quantization Matrix for HEVC)”,JCTVC-E073,

Geneva,CH,2011年3月)和JCTVC-H0314(参见Y.Wang等人,“分层矩阵量化表示和压缩(Layered quantization matrices representation and compression)”,JCTVC-H0314,San José,CA,USA,2012年2月)中。

[0057] JCTVC-E073:QM是在特定参数集(QMPS)中传送的。在QMPS内,QM以递增的尺寸顺序(与HEVC类似的sizeId/matrixId)传送。提出了根据任何先前译码的QM系数的预测(=复制),包括先前QMPS。使用具有线性内插的上变换来适应较小的参考QM,而使用简单的下采样来适应较大的参考QM。这最终在HEVC标准化期间被拒绝。

[0058] JCTVC-H0314:QM以从大到小的顺序传送。可以使用如图5所示的固定预测树(没有明确的参考索引)复制先前发送的QM而不是传送新的QM,如果参考QM较大,则使用简单的下采样。这最终在HEVC标准化期间被拒绝。

[0059] 这两种提议涉及HEVC,并且不能应付VVC引入的复杂性。

[0060] 本申请提出通过结合以下中的一者或多者来简化VVC草案5的量化矩阵信令和预测过程(在采用JVET-N0847之后),同时增强它们,使得可以从任何先前信令的QM预测任何QM:

-统一QM索引以包含尺寸和类型,使得参考索引差异可以解决任何先前传送的QM索引;

-以块尺寸递减顺序传送量化矩阵;

-根据需要将预测过程指定为复制或抽取过程;

-传送尺寸-64的所有QM系数,以使尺寸-64QM可用作预测符。

[0061] 此外,描述了包含大于8×8的块的上采样和矩形块的下采样的QM推导过程,以根据块参数选择QM索引,并使QM信号通知的尺寸适应于实际块尺寸。

[0062] 为了易于表示,我们将根据默认值或根据先前发送的值而预测量化矩阵的过程视为QM预测过程,并将使发送或预测的QM适应于变换块的尺寸和色度格式的过程视为QM推导过程。QM预测过程可以是例如在图像级别解析缩放列表数据的过程的一部分。推导过程通常在较低的级别,例如在变换块级别。下面进一步详细地呈现了各个方面,之后是草案文本示例和性能结果。

●用于识别QM的单个矩阵索引的推导和使用。单个标识符允许在使用预测(复制)时引用任何先前传送的矩阵,并且传送较大的矩阵首先避免预测过程中的内插。

●QM预测过程在于复制或抽取先前用信号通知的QM(参考QM),其是所传送、预测或默认的参考QM。

●QM推导过程,其对于给定的变换块,在于基于块尺寸、颜色分量和预测模式来选择QM索引,然后使所选择的QM的尺寸适应于块的尺寸。调整尺寸过程基于变换块内x和y坐标的比特移位,以索引所选择的QM的系数。

●对于尺寸-64QM的所有系数的传输。

[0063] 与VVC草案5相比,这些方面简化了规范(与JVET-N0847相比,将文本变化减半),并且带来了显著的比特节省(scaling_list_data的比特成本可以减半)。

[0064] 统一QM索引

[0065] 用于变换块的量化/去量化的QM由一个单个参数matrixId来标识。在一个实施例中,统一matrixId(QM索引)是以下项的组合:

-与CU尺寸(即,CU包围正方形形状,因为仅传送正方形尺寸矩阵)

而非块尺寸相关的尺寸标识符。注意,这里对于亮度或色度,尺寸标识符由亮度块尺寸(例如max(亮度块宽度、亮度块高度))控制。当亮度和色度树被分离时,对于色度,“CU尺寸”将指投影在亮度平面上的块的尺寸。

-首先列出亮度QM的矩阵类型,因为它们可以大于色度(例如,在4:2:0色度格式的情况下)

[0066] 根据该实施例,QM索引推导在表4和5以及等式(1)中示出。

表4:尺寸标识符(所提出的)

亮度	色度	sizeId
64x64	32x32	0
32x32	16x16	1
16x16	8x8	2
8x8	4x4	3
4x4	2x2	4

表5:矩阵类型标识符(所提出的)

CuPredMode	cIdx(颜色分量)	matrixTypeId
MODE_INTRA	0(Y)	0
MODE_INTER	0(Y)	1
MODE_INTRA	1(Cb)	2
MODE_INTER	1(Cb)	3
MODE_INTRA	2(Cr)	4
MODE_INTER	2(Cr)	5

[0067] 统一matrixId推导如下:

$$MatrixId = N*sizeId+MatrixTypeId \quad (1)$$

其中N是可能的类型标识符的数量,例如N=6。

[0068] 在另一个实施例中,如果定义了多于六个QM类型,则sizeId应该乘以正确的数量,该数量是量化矩阵类型的数量。在其它实施例中,其它参数也可以是不同的,例如,特定的块尺寸、用信号通知的矩阵尺寸(这里限于8×8)、或DC系数的存在。这里注意,QM通过减小块尺寸来列出,并且通过单个索引来标识,如表6所示。

表6:统一matrixId(所提出的)

Y	**帧内**	0		6		12		18		24	
	帧间	1		7		13		19		25	
Cb	**帧内**		2		8		14		20		26
	帧间		3		9		15		21		27
Cr	**帧内**		4		10		16		22		28
	帧间		5		11		17		23		29
CU 尺寸:亮度 max(宽度,高度)		**64**		**32**		**16**		**8**		**4**	
块尺寸: max(宽度,高度)		**64**	**32**		**16**		**8**		**4**		**2**
用信号通知的 QM 系数		**8x8 + DC**					**8x8**		**4x4**		**2x2**

[0069] QM预测过程

[0070] 代替传送QM系数,可以根据默认值或根据任何先前传送的值来预测QM。在一个实施例中,当参考QM是相同尺寸时,复制QM,否则按相关比率抽取它,如图6中的示例中所示,其中根据尺寸-8预测尺寸-4的亮度QM。

[0071] 抽取由以下等式描述:

ScalingMatrix[matrixId][x][y]＝refScalingMatrix[i][j] (2)

其中matrixSize＝(matrixId<20)?8:(matrixId<26)?4:2)

x＝0…matrixSize-1,y＝0…matrixSize-1,

i＝x<<(log2(refMatrixSize)-log2(matrixSize)),以及

j＝y<<(log2(refMatrixSize)-log2(matrixSize))。

其中refMatrixSize与refScalingMatrix的尺寸匹配(且因此与i和j变量的范围匹配)。

[0072] 在图6所示的例子中,根据亮度尺寸-8QM预测亮度尺寸-4QM(其是4×4阵列:matrixSize是4),亮度尺寸-8QM是8×8阵列(refMatrixSize是8);两行中的一行和两列中的一列被丢弃以生成4×4阵列(即,参考QM中的元素(2x,2y)被复制到当前QM中的元素(x,y))。

[0073] 等式(3)采用以下形式:

ScalingMatrix[matrix Id][x][y]＝refScalingMatrix[i][j] (3)

x＝0…3,y＝0…3,i＝x<<1,j＝y<<1。

[0074] 当参考QM具有DC值时,如果当前QM需要DC值,则将其复制为DC值;否则,将其复制到左上QM系数。

[0075] 在优选实施例中,该QM预测过程是QM解码过程的一部分,但是在另一实施例中,可以推迟到QM推导过程,其中,用于预测目的抽取将与QM调整尺寸子过程合并。

[0076] QM推导过程

[0077] 所提出的用于量化矩阵的推导过程首先根据如前所述的块参数(统一QM索引)选择正确的QM索引,然后将用于矩形块的抽取、用于大于例如8×8尺寸的块的重复以及色度格式适应的过程统一到一个单个过程中。所提出的过程基于x和y输出坐标的比特移位。为了选择所选择的QM的右行/列,只需要左移,然后右移x/y输出坐标,如下面的等式所示。

m[x][y]＝ScalingMatrix[matrixId][i][j] (4)

其中i＝(x<<log2MatrixSize)>>log2(blkwidth),以及

j＝(y<<log2MatrixSize)>>log2(blkHeight)。

其中log2MatrixSize是ScalingMatrix[matrixId](其为正方形2D阵列)的尺寸的log2,blkWidth和blkHeight分别是当前变换块的宽度和高度,其中x的范围为0到blkWidth-1,y的范围为0到blkHeight-1。

[0078] 下面提供几个例子来说明QM推导过程。在如图7中所示出的实例中,亮度16×8块的QM是根据明度尺寸-16QM推导的,所述QM实际上是8×8阵列加上DC系数。对于该示例,blkWidth等于16,blkHeight等于8,log2MatrixSize等于3,因此等式(5)采用以下形式:

m[x][y]＝ScalingMatrix[matrixId][i][j] (5)

其中i＝(x<<3)>>4,并且j＝(y<<3)>>3,其中x＝0…15并且y＝0…7。

这里,x向右移位1,y不变(即,将选择的QM中的i列复制到当前QM中的2*i和2*i+1列)。另外,由于所选择的QM具有DC系数,因此将其复制到m[0][0]。

[0079] 在如图8中所说明的另一实例中,产生8x4 CU(4:2:0格式)的色度4x2块的QM。这与8×4CU尺寸相匹配,其包围的正方形是8×8。因此,所选择的QM是尺寸-8QM,其中色度QM被译码为4×4阵列。这里,blkWidth等于4,blkHeight等于2,以及log2MatrixSize等于2,因此等式(6)采用以下形式:

$$m[x][y]=ScalingMatrix[matrixId][i][j] \quad (6)$$

其中i=(x<<2)>>2,并且j=(y<<2)>>1,其中x=0…3并且y=0…1。

这里,x未改变,y左移1(即,参考QM中的2y行被复制到当前QM中的y行)。

[0080] 在以下示例中,所提出的对4:2:2和4:4:4格式的适应不同于VVC草案5。代替寻找匹配色度块尺寸的QM(除了不存在色度矩阵的64x64),尺寸匹配基于相同的(亮度)CU尺寸(即,投影在亮度平面上的块的尺寸),并且如果需要的话重复系数。这使得QM设计独立于色度格式。

[0081] 在图9中所示的实例中,产生8x4 CU(4:2:2格式)的色度8x2块的QM。所选择的QM与如图8中所示的上述示例相同,但是4:2:2色度格式需要多两倍的列。这里,重复列,因此x右移1,y仍然左移1。特别地,blkWidth等于8,blkHeight等于2,以及log2MatrixSize等于2,因此等式(7)采取以下形式:

$$m[x][y]=ScalingMatrix[matrix\ Id][i][j] \quad (7)$$

其中i=(x<<2)>>3,并且j=(y<<2)>>1,其中x=0…7并且y=0…1。

[0082] 在图10中所示的实例中,产生8x4 CU(4:4:4格式)的色度8x4块的QM。所选择的QM仍然与图8和9中所示的示例中的相同,但是4:4:4色度格式需要4:2:0色度格式两倍的行和列。这里,必须重复列,因此x被右移1,但是可以跳过行的抽取(由于矩形形状),因此y不被移位。特别地,blkWidth等于8,blkHeight等于4,以及log2MatrixSize等于2,因此公式(8)采用以下形式:

$$m[x][y]=ScalingMatrix[matrixId][i][j] \quad (8)$$

其中i=(x<<2)>>3,并且j=(y<<2)>>2,其中x=0…7并且y=0…3。

[0083] 传送以用于尺寸-64的系数的数量

[0084] 在一个实施例中,为了实现根据尺寸-64的较小QM的预测,在缩放列表语法中传送尺寸-64的所有系数,即使VVC变换和量化过程从未使用右下象限。通常,我们可以发送最大QM的所有系数。

[0085] 然而,值得注意的是,与先前工作(JVET-N0847)相比,与所传送系数的该增加的数量相关的开销在最坏情况下可限于2×16比特,因为语法元素scaling_list_delta_coef在其不被用作预测符时可针对尺寸-64QM的右下象限设定为零:对于右下象限,用信号发送4×4=16个增量系数,如果强制为零(用exp-Golomb译码),则每个耗费1比特,并且存在两个尺寸-64QMs(亮度内部/中间)。

[0086] 表8中描述了测试,表明与预测改进带来的增益相比,该开销是微不足道的。

[0087] 在另一实施例中,尺寸-64QMs右下象限的系数不作为尺寸-64QM的信令的一部分来传送,而是在根据给定尺寸-64QM第一次预测到较小QM时作为补充参数来传送。

[0088] 表7提供了JVET-N0847中描述的方法与所提出的方法的一些比较:

表7

JVET-N0847	所提出的
- sizeId 和 matrixId 之间的复杂规则和依赖关系 - 帧内/帧间的单个默认矩阵 - 矩形块的特定文本 - 新的 Raster2Diag 扫描 - 5 维 ScalingFactor - 64x64 的清零过程	- 单独的 matrixId，单个 ScalingList 阵列 - 根据任何先前发出信号的矩阵利用抽取进行预测 - 单独的帧内/帧间默认矩阵 - 统一过程放大/矩形/4:x:x 色度 - 没有新的扫描 - 无清零流程

[0089] 下面,描述根据实施例的一些语法和语义。

[0090] PPS语法语义(微调整)

pic_parameter_set_rbsp( ) {	描述符
…	
pps_scaling_list_data_present_flag	u(1)
if(pps_scaling_list_data_present_flag)	
scaling_list_data( )	
pps_extension_flag	u(1)
…	
}	

pps_scaling_list_data_present_flag等于1指定用于参考PPS的图片的缩放列表数据是基于由活动SPS指定的缩放列表和由PPS指定的缩放列表推导的。pps_scaling_list_data_present_flag等于0指定用于参考PPS的图片的缩放列表数据被推断为等于由活动SPS指定的那些。当scaling_list_enabled_flag等于0时,pps_scaling_list_data_present_flag的值应等于0。当scaling_list_enabled_flag等于1,sps_scaling_list_data_present_flag等于0且pps_scaling_list_data_present_flag等于0时,默认缩放矩阵用于推导如在第7.4.5款中规定的缩放列表数据语义中描述的阵列ScalingMatrix。

[0091] 请注意,此语法/语义是接近HEVC标准或VVC草案的示例,而不是限制性的。例如,scaling_list_data运输不限于SPS或PPS,并且可以通过其它方式来发送。

[0092] 缩放列表数据语法/语义(简化的)

scaling_list_data( ) {	描述符
for(matrixId = 0; matrixId < 30; matrixId++) {	
scaling_list_pred_mode_flag[matrixId]	u(1)
if (!scaling_list_pred_mode_flag[matrixId])	
scaling_list_pred_matrix_id_delta[matrixId]	ue(v)
else {	
nextCoef = 8	
coefNum = (matrixId < 20) ? 64 : (matrixId < 26) ? 16 : 4	
if (matrixId < 14) {	
scaling_list_dc_coef_minus8[matrixId]	se(v)
nextCoef = scaling_list_dc_coef_minus8[matrixId] + 8	
}	
for(i = 0; i < coefNum; i++) {	
scaling_list_delta_coef	se(v)
nextCoef = (nextCoef + scaling_list_delta_coef + 256) % 256	
ScalingList[matrixId][i] = nextCoef	
}	
}	
}	
}	

scaling_list_pred_mode_flag[matrixId]等于0指定根据参考缩放矩阵的值推导缩放矩阵。参考缩放矩阵由scaling_list_pred_matrix_id_delta[matrixId]指定。scaling_list_pred_mode_flag[matrixId]等于1指定显式地用信号发送缩放列表的值。

scaling_list_pred_matrix_id_delta[matrixId]指定用于推导缩放矩阵的参考缩放矩阵,如下所述。scaling_list_pred_matrix_id_delta[matrixId]的值应当在0到matrixId的范围内,包括端值。

当scaling_list_pred_mode_flag[matrixId]等于零时:

-变量refMatrixSize和阵列refScalingMatrix首先如下推导:

○如果scaling_list_pred_matrix_id_delta[matrixId]等于零,则以下内容适用于设置默认值:

■refMatrixSize设置为等于8,

■如果matrixId是偶数,

refScalingMatrix = (9)

{

{16，16，16，16，16，16，16，16}// 帧内默认值的占位符

{16，16，16，16，16，16，16，16}

{16，16，16，16，16，16，16，16}

{16，16，16，16，16，16，16，16}

{16，16，16，16，16，16，16，16}

{16，16，16，16，16，16，16，16}

{16，16，16，16，16，16，16，16}

{16，16，16，16，16，16，16，16}

},

■否则

refScalingMatrix = (10)

{

{16，16，16，16，16，16，16，16}// 帧间默认值的占位符

{16，16，16，16，16，16，16，16}

{16，16，16，16，16，16，16，16}

{16，16，16，16，16，16，16，16}

{16，16，16，16，16，16，16，16}

{16，16，16，16，16，16，16，16}

{16，16，16，16，16，16，16，16}

{16，16，16，16，16，16，16，16}

},

○否则(如果scaling_list_pred_matrix_id_delta[matrixId]大于零),则以下适用:

refMatrixId＝matrixId-scaling_list_pred_matrix_id_delta[matrixId] (11)

refMatrixSize＝(refMatrixId<20)?8:(refMatrixId<26)?4:2) (12)

refScalingMatrix＝ScalingMatrix[refMatrixId] (13)

-然后如下推导阵列ScalingMatrix[matrixId]:

ScalingMatrix[matrixId][x][y]＝refScalingMatrix[i][j] (14)

其中,matrixSize＝(matrixId<20)?8:(matrixId<26)?4:2)

x＝0…matrixSize-1,y＝0…matrixSize-1,

i＝x<<(log2(refMatrixSize)-log2(matrixSize)),以及

j＝y<<(log2(refMatrixSize)-log2(matrixSize))

scaling_list_dc_coef_minus8[matrixId]加8指定了相关时缩放矩阵的第一个值,如在子句xxx中所述。scaling_list_dc_coef_minus8[matrixId]的值应当在-7到247的范围内,包括端值。

当scaling_list_pred_mode_flag[matrixId]等于零时,scaling_list_pred_matrix_id_delta[matrixId]大于零,且refMatrixId＜14,则以下适用:

-如果matrixId＜14,则scaling_list_dc_coef_minus8[matrixId]被推断为等于scaling_list_dc_coef_minus8[refMatrixId],

-否则,ScalingMatrix[matrixId][0][0]被设置为等于scaling_list_dc_coef_minus8[refMatrixId]+8

当scaling_list_pred_mode_flag[matrixId]等于零时,scaling_list_pred_matrix_id_delta[matrixId]等于零(指示默认值),且matrixId＜14,则scaling_list_dc_coef_minus8[matrixId]被推断为等于8

scaling_list_delta_coef指定当scaling_list_pred_mode_flag[matrixId]等于1时,当前矩阵系数ScalingList[matrixId][i]与先前矩阵系数ScalingList[matrixId][i-1]之间的差。scaling_list_delta_coef的值应当在-128至127的范围内,包括端值。ScalingList[matrixId][i]的值应大于0。当存在时(即scaling_list_pred_mode_flag[matrixId]等于1),阵列ScalingMatrix[matrix_Id]如下推导:

ScalingMatrix[matrixId][i][j]＝ScalingList[matrixId][k] (15)

其中,k＝0…coefNum-1,

i＝diagScanOrder[log2(coefNum)/2][log2(coefNum)/2][k][0],以及

j＝diagScanOrder[log2(coefNum)/2][log2(coefNum)/2][k][1]

[0093] 与JVET-N0847语法相比的主要简化是移除一个for()循环,且从[sizeId][matrixId]到[matrixId]进行索引简化。

[0094] “子句xxx”指的是要在VVC规范中引入的未确定的节编号,其与本文的缩放矩阵推导过程相匹配。

[0095] 请注意,此语法/语义是接近HEVC标准或VVC草案5的示例,而不是限制性的。例如,系数范围不限于1…255,例如可以是1…127(7比特)或-64…63。而且,它不限于30个QM,组织为6种类型×5个尺寸(可以有8种类型,以及更少或更多的尺寸。参见表5,其可以被修改。然后将需要对coefNum和DC系数存在的条件的简单修改)。QM预测的类型(这里仅是复制的)也不是限制性的。例如,可以添加缩放因子或偏移,并且可以在预测之上添加显式译码作为残差。同样的情况代表用于系数传输的方法(这里为DPCM)、DC系数的存在以及固定的系数数量(只有子集可以被传送)。

[0096] 关于默认值,其不限于与MODE_INTRA和MODE_INTER相关联的两个默认值QM,并且在这里类似地用全部16个值填充,直到达成相关默认值为止(例如,可以选择与HEVC相同的默认值QM)。

[0097] 同样,可以数学地表达要用信号通知的系数coefNum的数量,而不是比较序列,具有相同的结果:coefNum=Min(64,4096>>((matrixId+4)/6)*2),这更接近于HEVC或当前的VVC草案方式,但是引入了可能不受欢迎的划分。

[0098] 这里注意,首先传送较大的矩阵,并且使用单个索引,使得预测参考(由scaling_list_pred_matrix_id_delta指示)可以是任何先前传送的矩阵,而不管其预期的块尺寸或类型,或者默认值(例如,如果scaling_list_pred_matrix_id_delta是零)。

[0099] 图11示出了根据实施例的用于解析缩放列表数据语法结构的过程(1100)。对于该实施例,输入是译码比特流,输出是ScalingMatrix阵列。为了清楚起见,省略了关于DC值的细节。具体地,在步骤1110,从比特流解码QM预测模式。如果QM被预测(1120),则解码器进一步根据前述标志来确定是在比特流中推断(预测)QM或还是用信号通知QM。在步骤1130,解码器从比特流解码QM预测数据,当没有信号通知例如QM索引差scaling_list_pred_matrix_id_delta时,需要该QM预测数据来推断QM。然后,解码器确定(1140)是根据默认值(例如,如果scaling_list_pred_matrix_id_delta是零)还是根据先前解码的QM来预测QM。如果参考QM是默认QM,则解码器选择(1150)默认QM作为参考QM。可以有几个默认QM来选择,例如取决于matrixId的奇偶校验。否则,解码器选择(1155)先前解码的QM作为参考QM。参考QM的索引是从矩阵和上述索引差推导的。在步骤1160,解码器根据参考QM预测QM。如果参考QM与当前QM的尺寸相同,则预测由简单的复制组成,或者如果它大于预期,则由抽取组成。结果存储在ScalingMatrix[matrixId]中。

[0100] 如果QM未被预测(1120),则解码器根据matrixId确定(1170)要从比特流解码的QM系数的数量。例如:如果matrixId小于20,则为64,如果matrixId在20和25之间,则为16,否则为4。在步骤1175,解码器从比特流中解码相关数量的QM系数。在步骤1180,解码器根据扫描顺序,例如对角扫描,将解码的QM系数组织在2D矩阵中。结果存储在ScalingMatrix[matrixId]中。使用ScalingMatrix[matrixId],解码器可以使用QM推导过程来获得量化矩阵m[][],用于对可能处于非方形形状和/或处于不同色度格式的变换块进行去量化。

[0101] 在步骤1190,解码器检查当前QM是否是要解析的最后一个QM。如果不是最后一个,则控制返回步骤1110;否则,当从比特流解析了所有QM时,QM解析过程停止。

[0102] 图12示出了根据实施例的在编码器侧对缩放列表数据语法结构进行编码的过程(1200)。在编码器侧,按照所述的顺序,从较大到较小的块尺寸(例如,从0到30的matrixId)扫描QM。在步骤1210,编码器搜索预测偏好,以决定当前QM是否是先前编码的QM的复制(或抽取)。QM可以以优化QM预测效率的方式来设计,例如,如果某QM最初足够接近,或者与默认值QM接近,则它们可以被强制为相等(或者如果尺寸不同,则被抽取)。此外,尺寸-64QM的右下象限中的系数可以被优化以便更好地预测后续QM,或者如果从未被重新用于预测则降低QM比特成本。一旦决定,QM预测模式被编码在比特流中。

[0103] 具体地,如果编码器决定(1220)使用预测,则在步骤1230,编码预测模式(例如,scaling_list_pred_mode_flag=0)。在步骤1240,对预测参数(例如,QM索引差scaling_list_pred_matrix_id_delta)进行编码:对于默认QM值为零索引差,或者如果选择先前的

QM作为预测参考,则为相关索引差。另一方面,如果决定了显式信令,则在步骤1250,对预测模式进行编码(例如,scaling_list_pred_mode_flag＝0)。然后,执行对角扫描(1260),随后执行QM系数编码(1270)。

[0104] 在步骤1280,编码器检查当前QM是否是要编码的最后一个QM。如果不是最后一个,则控制返回步骤1210;否则,当所有QM都被编码到比特流时,QM编码过程停止。

[0105] 图13示出了根据实施例的用于QM推导的过程1300。输入可以包括ScalingMatrix阵列和变换块参数,例如,尺寸(宽度/高度)、预测模式(帧内/帧间/IBC,……)和颜色分量(Y/U/V)。输出是具有与变换块相同尺寸的QM。为了清楚起见,省略了关于DC值的细节。具体地,在步骤1310,解码器根据如前所述的当前变换块尺寸(宽度/高度)、预测模式(帧内/帧间/IBC,……)、颜色分量(Y/U/V)(统一QM索引)来确定QM索引matrixId。在步骤1320,解码器调整所选QM(ScalingMatrix[matrixId])的尺寸以匹配变换块尺寸,如前所述。在一变型中,步骤1320可包含预测所需的抽选。

[0106] QM推导过程在编码器侧是类似的。量化将变换系数除以QM值,而去量化乘以QM值。但是QM是相同的。特别地,在编码器中用于重构的QM将匹配在比特流中用信号通知的那些。

[0107] 概念上,变换系数d[x][y]可以如下量化,其中qStep是量化步长,m[][]是量化矩阵:

TransCoeffLevel[xTbY][yTbY][cIdx][x][y]＝d[x][y]/qStep/m[x][y]

[0108] 但是对于整数计算并且为了避免除法,它通常看起来像:

TransCoeffLevel[xTbY][yTbY][cIdx][x][y]＝((d[x][y]*im[x][y]*ilevelScale[qP%6]>>(qP/6))+(1<<(bdShift-1)))>>bdShift)

其中,例如im[x][y]～＝65536/m[x][y]且ilevelScale[0…5]＝65536/levelScale[0…5],并且bdShift是适当值。实际上,对于软件编码器,im*ilevelScale通常被预先计算并存储在表中。

[0109] 在上述中,QM预测处理和QM推导处理是分开执行的。在另一实施例中,QM预测可以推迟到QM推导过程。该实施例不改变QM信令语法。由于连续的调整尺寸而在功能上不同的本实施例将预测部分(获得参考QM+复制/缩小)和对角线扫描推迟到稍后描述的“QM推导过程”。

[0110] 在一个实施例中,缩放列表数据解析过程1100的输出是ScalingList()的阵列而不是ScalingMatrix的阵列,以及预测标志和有效预测参数:ScalingMatrixPredId阵列总是包含定义的ScalingList的索引(默认的或用信号通知的)。在QM解码期间,通过解释scaling_list_pred_matrix_id_delta递归地构建该阵列,使得QM推导过程可以直接使用该索引来获得实际值,以构建用于对当前变换块进行去量化的QM。

[0111] 在下文中,提供了一示例来说明根据实施例的缩放列表语义。

[0112] 缩放矩阵推导过程(新:部分替换缩放列表语义中的描述;这是XXX节)

该过程的输入是预测模式predMode、颜色分量变量cIdx、块宽度blkWidth和块高度blkHeight。

该过程的输出是(blkWidth)x(blkHeight)阵列m[x][y](缩放矩阵),其中x和y是水平和垂直系数位置。注意,SubWidthC和SubHeightC取决于色度格式,并且指示亮度分量和色度分量中的样本数量的比率。

变量matrixId如下推导:

matrixId＝6*sizeId+matrixTypeId(xxx-1)其中,subWidth＝(cIdx>0)?SubWidthC:1,

subHeight＝(cIdx>0)?SubHeightC:1,

sizeId＝6-max(log2(blkWidth*subWidth),log2(blkHeight*subHeight)),以及

matrixTypeId＝(2*cIdx+(predMode＝＝MODE_INTER?1:0))

变量log2MatrixSize如下推导:

log2MatrixSize＝(matrixId<20)?3:(matrixId<26)?2:1(xxx-2)对于包括从0到blkWidth-1的x范围,以及包括从0到blkHeight-1的y范围,通过应用以下公式推导输出阵列m[x][y]:

m[x][y]＝ScalingMatrix[matrixId][i][j](xxx-3)

其中i＝(x<<log2MatrixSize)>>log2(blkWidth),以及

j＝(y<<log2MatrixSize)>>log2(blkHeight)

如果matrixId低于14,则m[0][0]被进一步修改如下:

m[0][0]＝scaling_list_dc_coef_minus8[matrixId]+8 (xxx-4)

[0113] 请注意,对于scaling_list_data语法和语义,这是示例,而不是限制性的。例如,它不限于与MODE_INTRA和MODE_INTER相关联的两个默认QM,并且在这里类似地用全部16个值填充,直到达到相关默认值为止(例如,可以选择与HEVC相同的默认QM)。可以存在单个默认QM或多于2个默认QM。例如,如果每块尺寸存在多于或少于六种类型的矩阵,则MatrixId计算可以是不同的。重要的是,在单个过程中完成水平和垂直缩小和放大以适应不同于所选QM的块尺寸,优选地是简单的(这里左移之后右移)。

[0114] 对于矩形块,不限于为包围正方形的当前块选择QM标识符:在等式xxx-1中sizeId的推导可以遵循不同的规则。

[0115] 同样,可以用数学方法表达所选择的QM尺寸log2MatrixSize,而不是一系列比较,其中结果相同:log2MatrixSize＝min(3,6-(MatrixId+4)/6),但是这引入了可能不受欢迎的除法。

[0116] 在下文中,提供了根据实施例的描述用于缩放处理的语义的示例。

[0117] 变换系数的缩放处理(适应)

[…]

对于推导的经缩放变换系数d[x][y],其中x＝0…nTbW-1,y＝0…nTbH-1,应用以下:

-(nTbW)x(nTbH)中间缩放因子阵列m如下推导:

-如果以下条件中的一者或多者为真,则m[x][y]被设置为等于16:

-scaling_list_enabled_flag等于0。

-transform_skip_flag[xTbY][yTbY]等于1。

-否则,m是如子句xxx中所指定的缩放矩阵推导过程的输出,其以预测模式CuPredMode[xTbY][yTbY]、颜色分量变量cIdx、块宽度nTbW和块高度nTbH作为输入来调用。

-缩放因子ls[x][y]如下推导:

[…]

[0118] 与VVC草案5相比的主要变化是子句xxx的调用,而不是复制scaling_list_data语义中描述的阵列的一部分。

[0119] 请注意,如上所述,这是一个旨在最小化当前VVC草案的变化的示例,而不是限制性的。例如,用于缩放矩阵推导过程的颜色分量输入可以不同于cIdx。此外,QM不限于用作缩放因子,而是例如可以用作QP偏移。

[0120] 使用与HEVC标准化期间使用的测试集相同的测试集,测试了QM译码性能,该QM译码性能被扩充为从在实际广播中发现的公共标准(JPEG、MPEG2、AVC、HEVC)和QM的推荐或默认QM推导的QM集合。在所有测试中,将一些QM从一种类型复制到另一种类型(例如,亮度到色度,或帧内到帧间),和/或将一个尺寸复制到另一尺寸。

[0121] 下表报告了使用以下三种不同方法编码scaling_list_data所需的比特数:HEVC,JVET-N0847、和所提出的。具体地,HEVC使用HEVC测试集(每个测试24个QM),并且另两个使用推导的测试集合(每个测试30个QM,具有附加尺寸:色度的尺寸-2和亮度的尺寸-64;尺寸-2QM是从尺寸-4下采样的,尺寸-64是从尺寸-32复制的,尺寸-32是从尺寸-16复制的,并且尺寸16、8、4保持原样)。

表8

	HEVC	N0847	N0847 vs HEVC	**所提出的**	**所提出的vs N0847**	**所提出的** vs HEVC
JPEG	3007	3729	+722 (+24%)	**1059**	**-2670 (-72%)**	-1948 (-65%)
MPEG2	1067	1333	+266 (+25%)	**489**	**-844 (-63%)**	-578 (-54%)
H264-类似	950	1224	+274 (+29%)	**480**	**-744 (-61%)**	-470 (-49%)
DTT-类似	2264	2940	+676 (+30%)	**1142**	**-1798 (-61%)**	-1122 (-50%)
HEVC	1281	1554	+273 (+21%)	**535**	**-1019 (-66%)**	-746 (-58%)
symmetry1	2090	2658	+568 (+27%)	**892**	**-1766 (-66%)**	-1198 (-57%)
symmetry2	1272	1620	+348 (+27%)	**997**	**-623 (-38%)**	-275 (-22%)
symmetry3	2660	3220	+560 (+21%)	**2094**	**-1126 (-35%)**	-566 (-21%)
symmetry4	2542	3146	+604 (+24%)	**2064**	**-1082 (-34%)**	-478 (-19%)
symmetry7	3366	4264	+898 (+27%)	**2675**	**-1589 (-37%)**	-691 (-21%)
asymmetry1	2414	3078	+664 (+28%)	**1004**	**-2074 (-67%)**	-1410 (-58%)
asymmetry2	1394	1778	+384 (+28%)	**1057**	**-721 (-41%)**	-337 (-24%)
asymmetry3	2962	3624	+662 (+22%)	**2272**	**-1352 (-37%)**	-690 (-23%)
asymmetry4	3560	4344	+784 (+22%)	**2828**	**-1516 (-35%)**	-732 (-21%)
asymmetry7	3946	5406	+1460 (+37%)	**3268**	**-2138 (-40%)**	-678 (-17%)
平均	*2318*	*2928*	*+610 (+26%)*	***1524***	***-1404 (-48%)***	*-795 (-34%)*

[0122] 对于该测试,可以看出,所提出的技术节省了大量的比特,即使与HEVC相比,而所提出的方法编码了更多的QM。

[0123] 返回JCTVC-E073中的方法,JCTVC-E073中的参考索引(三元组:QMPS,尺寸,类型)比这里提出的更复杂,因为先前的QMPS索引需要存储先前的QMPS。线性插值引入了复杂性。下采样类似于这里所提出的。

[0124] 回到JCTVC-H0314中的方法,从大到小的传输接近于这里所提出的,然而JCTVC-

H0314中的固定预测树比这里所提出的统一索引和明确参考的灵活性要低。

[0125] 用于帧内块复制模式的QM

[0126] 在上面,为两种块预测模式(即帧内和帧间)指定不同的QM。然而,除了帧内和帧间之外,在VVC中还存在新的预测模式:IBC(帧内块复制),其中可以利用适当的位移向量根据相同图片的重构样本预测块。对于在IBC预测模式中的QM选择,JVET-N0847和上述实施例使用与帧内模式相同的QMs。

[0127] 由于相比于帧内预测,IBC模式更接近帧间,因此在一个实施例中,提出了将用信号通知的QM重新用于帧间模式(而不是帧内)。然而,尽管接近于帧间预测,IBC是不同的:位移矢量与对象或相机运动不匹配,但是用于纹理复制。这可能导致特定的伪像,其中特定QM在不同的实施例中可以帮助优化IBC块的译码。下面,我们针对IBC预测模式提出改变QM选择:

●优选实施例是选择相同的QM作为帧间模式(而不是帧内),因为相比于帧内预测,IBC更接近帧间预测。

●另一个选项是具有用于IBC模式的特定QM。

○这些可以在语法中显式地用信号通知,或者推断(例如帧内和帧间QM的平均值)。

[0128] 在优选实施例中,如果块具有IBC预测模式,则用于特定变换块的QM选择或推导过程选择帧间QM。返回参考图12,需要如下所述地调整步骤1210的QM推导过程。

[0129] 在上面提出的草案文本中,QM选择在等式(xxx-1)中描述,其可以如下改变:

matrixId=6*sizeId+matrixTypeId(xxx-1)

其中subWidth=(cIdx>0)?SubWidthC:1,

subHeight=(cIdx>0)?SubHeightC:1,

sizeId=6-max(log2(blkWidth*subWidth),log2(blkHeight*subHeight)),以及

~~matrixTypeId=(2*cIdx+(predMode==MODE_INTER?1:0))~~

matrixTypeId=(2*cIdx+(predMode==MODE_INTRA?0:1))

[0130] 注意,块可以以帧内模式(MODE_INTRA)、帧间模式(MODE_INTER)或帧内块复制模式(MODE_IBC)来编码。当如表6中那样设置matrixTypeId时,或者(matrixTypeId=(2*cIdx+(predMode==MODE_INTER?1:0))),MODE_IBC块选择matrixTypeId就好像它是MODE_INTRA块。其中(xxx-1)中的改变:matrixTypeId=(2*cIdx+(predMode==MODE_INTRA?0:1)),MODE_IBC块选择matrixTypeId就好像它是MODE_INTER块。

[0131] 在JVET-N0847所提出的草案文本中,QM选择在表7-14中描述,其可以如下改变。特别地,MODE_IBC的matrixId以与MODE_INTER相同的方式分配,而不是如JVET-N0847中那样与MODE_INTRA相同。

表9:根据sizeId、预测模式和颜色分量对matrixId的JVET-N0847规范的改变

sizeId	CuPredMode	cIdx (颜色分量)	matrixId
2, 3, 4, 5, 6	MODE_INTRA	0 (Y)	0
1, 2, 3, 4, 5	MODE_INTRA	1 (Cb)	1
1, 2, 3, 4, 5	MODE_INTRA	2 (Cr)	2
2, 3, 4, 5, 6	MODE_INTER	0 (Y)	3
1, 2, 3, 4, 5	MODE_INTER	1 (Cb)	4
1, 2, 3, 4, 5	MODE_INTER	2 (Cr)	5
2, 3, 4, 5, 6	MODE_IBC	0 (Y)	~~0~~ 3
1, 2, 3, 4, 5	MODE_IBC	1 (Cb)	~~1~~ 4
1, 2, 3, 4, 5	MODE_IBC	2 (Cr)	~~2~~ 5

[0132] 变型1:用于IBC的QM的显式信令

[0133] 在该变型中,特定QM用于IBC块(不同于帧内和帧间QM),并且在比特流中显式地用信号通知这些QM。这产生更多QM,其需要scaling_list_data语法和matrixId映射的适配,并且具有比特成本影响。根据该变型,可以如下改变等式(xxx-1)中描述的QM选择

matrixId＝6*sizeId+matrixTypeId(xxx-1)

其中subWidth＝(cIdx>0)?SubWidthC:1,

subHeight＝(cIdx>0)?SubHeightC:1,

sizeId＝6-max(log2(blkWidth*subWidth),log2(blkHeight*subHeight)),以及

~~matrixTypeId = (2 * cIdx + (predMode = = MODE_INTER ? 1 : 0))~~

matrixTypeId＝(3*cIdx+(predMode＝＝MODE_INTRA?0:(predMode＝＝MODE_INTER?1:2)))

[0134] 在JVET-N0847中,QM选择表可以如下改变:

表10:对JVET-N0847改变以增加更多用于IBC的QM

sizeId	CuPredMode	cIdx(颜色分量)	matrixId
2,3,4,5,6	MODE_INTRA	0(Y)	0
1,2,3,4,5	MODE_INTRA	1(Cb)	1
1,2,3,4,5	MODE_INTRA	2(Cr)	2
2,3,4,5,6	MODE_INTER	0(Y)	3
1,2,3,4,5	MODE_INTER	1(Cb)	4
1,2,3,4,5	MODE_INTER	2(Cr)	5
2,3,4,5,6	MODE_IBC	0(Y)	06
1,2,3,4,5	MODE_IBC	1(Cb)	1 7
1,2,3,4,5	MODE_IBC	2(Cr)	2 8

[0135] 变型2:推断的用于IBC模式的QM

[0136] 在这个变型中,特定的QM被用于IBC块(不同于帧内和帧间QM)。然而,这些QM不在比特流中用信号通知,而是推断为:例如帧内和帧间QM的平均值,或者特定的默认值,或者帧间QM的特定变化,如缩放和偏移。

[0137] 变型3:仅用于亮度的显式IBC QM

[0138] 在该变型中,用于IBC的附加QM仅限于亮度;用于IBC的色度QM可以如在变型1中那

样重复使用帧间QM,或者如在变型2中那样推断出新的色度QM。

[0139] 本文描述了各种方法,并且每种方法包括用于实现所描述的方法的一个或多个步骤或动作。除非方法的正确操作需要特定顺序的步骤或动作,否则可修改或组合特定步骤和/或动作的顺序和/或使用。另外,诸如“第一”、“第二”等术语可以在各种实施例中用于修改元件、组件、步骤、操作等,例如,“第一解码”和“第二解码”。除非特别要求,否则这些术语的使用并不意味着对修改后的操作的排序。因此,在该示例中,第一解码不需要在第二解码之前执行,并且可以例如在第二解码之前、期间或在与第二解码重叠的时间段中发生。

[0140] 本申请中描述的各种方法和其它方面可用于修改模块,例如,如图2和图3所示的视频编码器200和解码器300的量化和逆量化模块(230、240、340),此外,本发明的方面不限于VVC或HEVC,并且可应用于例如其它标准和建议,以及任何这种标准和建议的扩展。除非另外指出或在技术上排除,本申请中描述的方面可以单独或组合使用。

[0141] 在本申请中使用各种数值。具体值是出于示例目的,并且所描述的方面不限于这些具体值。

[0142] 各种实现方式涉及解码。如本申请中所使用的“解码”可以包括例如对接收到的编码序列执行的全部或部分过程,以便产生适合于显示的最终输出。在各种实施例中,此类过程包括通常由解码器执行的过程中的一个或多个,例如熵解码、逆量化、逆变换和差分解码。短语“解码过程”是否旨在具体地指代操作的子集或一般地指代更广泛的解码过程基于具体描述的上下文将是清楚的,并且相信是本领域技术人员所充分理解的。

[0143] 各种实现涉及编码。以与以上关于“解码”的讨论类似的方式,如在本申请中使用的“编码”可以包括例如对输入视频序列执行的以便产生编码比特流的过程的全部或部分。

[0144] 注意,这里使用的语法元素是描述性术语。因此,它们不排除使用其它语法元素名称。在上文中,用于PPS和缩放列表的语法元素主要用于说明各种实施例。应当注意,这些语法元素可以被置于其它语法结构中。

[0145] 本文描述的实现和方面可以在例如方法或过程、装置、软件程序、数据流或信号中实现。即使仅在单一形式的实现的上下文中讨论(例如,仅作为方法讨论),所讨论的特征的实现也可以以其他形式(例如,装置或程序)来实现。例如,可以以适当的硬件、软件和固件来实现装置。所述方法可以在例如装置中实现,所述装置例如处理器,其通常指处理设备,包括例如计算机、微处理器、集成电路或可编程逻辑设备。处理器还包括通信设备,例如计算机、蜂窝电话、便携式/个人数字助理(“PDA”)和便于终端用户之间的信息通信的其他设备。

[0146] 对“一个实施例”或“实施例”或“一个实现”或“实现”以及其它变型的提及意味着结合实施例描述的特定特征、结构、特性等包含于至少一个实施例中。因此,在本申请中的各个地方出现的短语“在一个实施例中”或“在实施例中”或“在一个实现中”或“在实现中”以及任何其他变型的出现不一定都指同一实施例。

[0147] 另外,本申请可以涉及“确定”各种信息。确定信息可以包括例如估计信息、计算信息、预测信息或从存储器检索信息中的一者或多者。

[0148] 此外,本申请可以涉及“访问”各种信息。访问信息可以包括例如接收信息、检索信息(例如,从存储器)、存储信息、移动信息、复制信息、计算信息、确定信息、预测信息或估计信息中的一者或多者。

[0149] 另外,本申请可以指“接收”各种信息。如同“访问”一样,接收旨在是广义的术语。接收信息可以包括例如访问信息或(例如从存储器)检索信息中的一者或多者。此外,在例如存储信息、处理信息、传送信息、移动信息、复制信息、擦除信息、计算信息、确定信息、预测信息或估计信息的操作期间,通常以一种方式或另一种方式涉及“接收”。

[0150] 应当理解,例如在“A/B”、“A和/或B”以及“A和B中的至少一者”的情况下,使用以下“/”、“和/或”以及“中的至少一者”中的任何一个旨在涵盖仅对第一列出的选项(A)的选择、或仅对第二列出的选项(B)的选择、或对两个选项(A和B)的选择。作为进一步的例子,在“A、B和/或C”和“A、B和C中的至少一者”的情况下,这样的措词旨在包括仅选择第一个列出的选项(A),或者仅选择第二个列出的选项(B),或者仅选择第三个列出的选项(C),或者仅选择第一个和第二个列出的选项(A和B),或者仅选择第一个和第三个列出的选项(A和C),或者仅选择第二个和第三个列出的选项(B和C),或者选择所有三个选项(A和B和C)。如本领域和相关领域的普通技术人员所清楚的,这可以扩展到所列的多个项目。

[0151] 此外,如本文所使用的,词语“信号/用信号通知”尤其是指向对应的解码器指示某物。例如,在某些实施例中,编码器用信号通知用于去量化的量化矩阵。这样,在实施例中,在编码器侧和解码器侧使用相同的参数。因此,例如,编码器可以向解码器发送(显式信令)特定参数,使得解码器可以使用相同的特定参数。相反,如果解码器已经具有特定参数以及其它参数,则可以使用信令而不进行发送(隐式信令),以简单地允许解码器知道并选择特定参数。通过避免任何实际功能的传输,在各种实施例中实现了比特节省。应当理解,可以以各种方式来实现信令。例如,在各种实施例中,一个或多个语法元素、标志等被用于将信息用信号通知给对应的解码器。虽然前述内容涉及词语“信号/用信号通知”的动词形式,但是词语“信号/用信号通知”在本文中也可以用作名词。

[0152] 如对于本领域普通技术人员将显而易见的,实现可以产生被格式化以携带例如可以被存储或发送的信息的各种信号。该信息可以包括例如用于执行方法的指令,或者由所描述的实现之一产生的数据。例如,信号可以被格式化以携带所描述的实施例的比特流。这种信号可以被格式化为例如电磁波(例如,使用频谱的射频部分)或基带信号。格式化可以包括例如编码数据流和用编码数据流调制载波。信号携带的信息可以是例如模拟或数字信息。如已知的,信号可以通过各种不同的有线或无线链路来传输。该信号可以存储在处理器可读介质上。

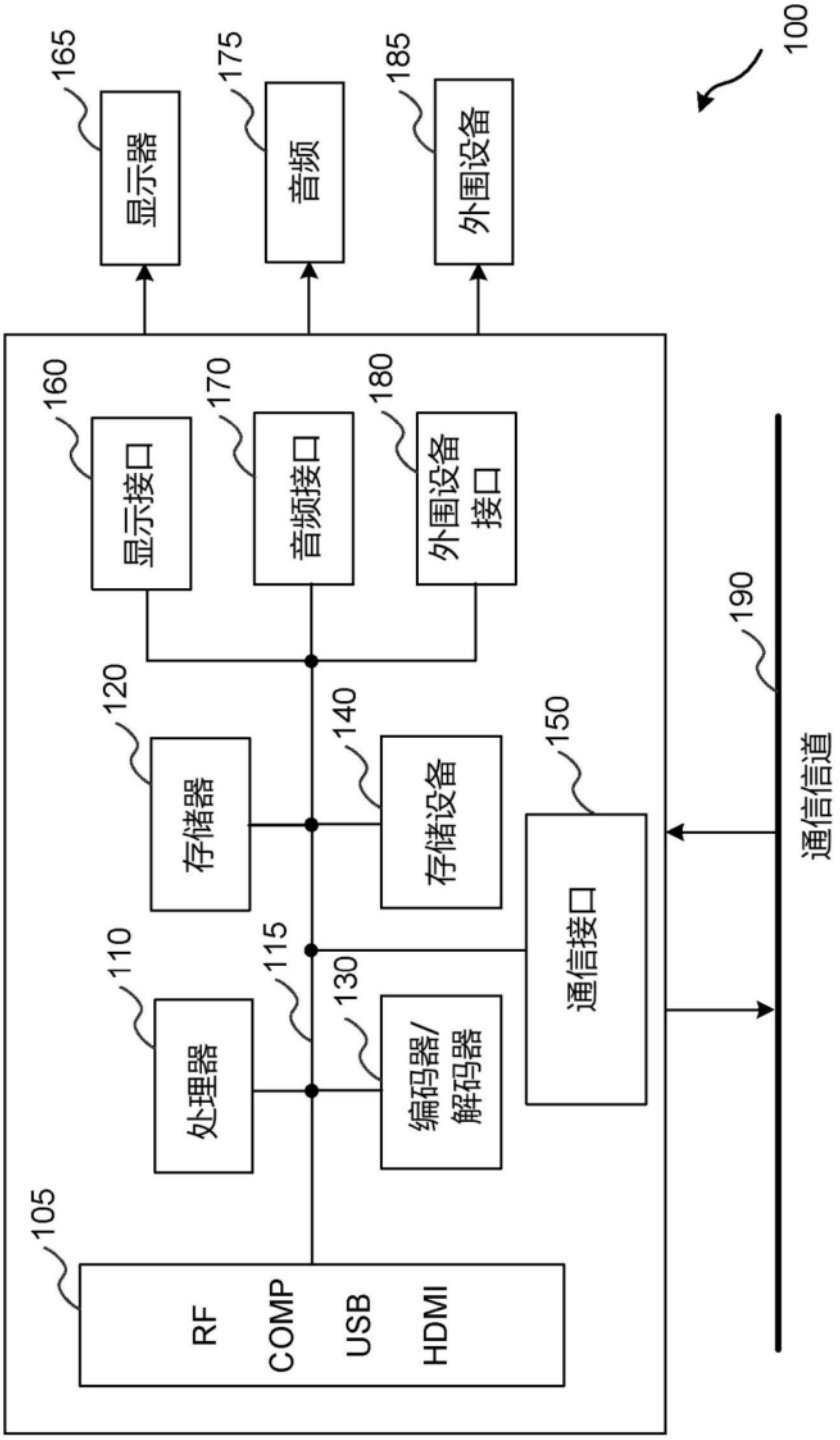

100
165
显示器
175
音频
185
外围设备
160
显示接口
170
音频接口
180
外围设备接口
190
通信信道
120
存储器
140
存储设备
150
通信接口
110
处理器
115
130
编码器/解码器
105
RF
COMP
USB
HDMI

图1

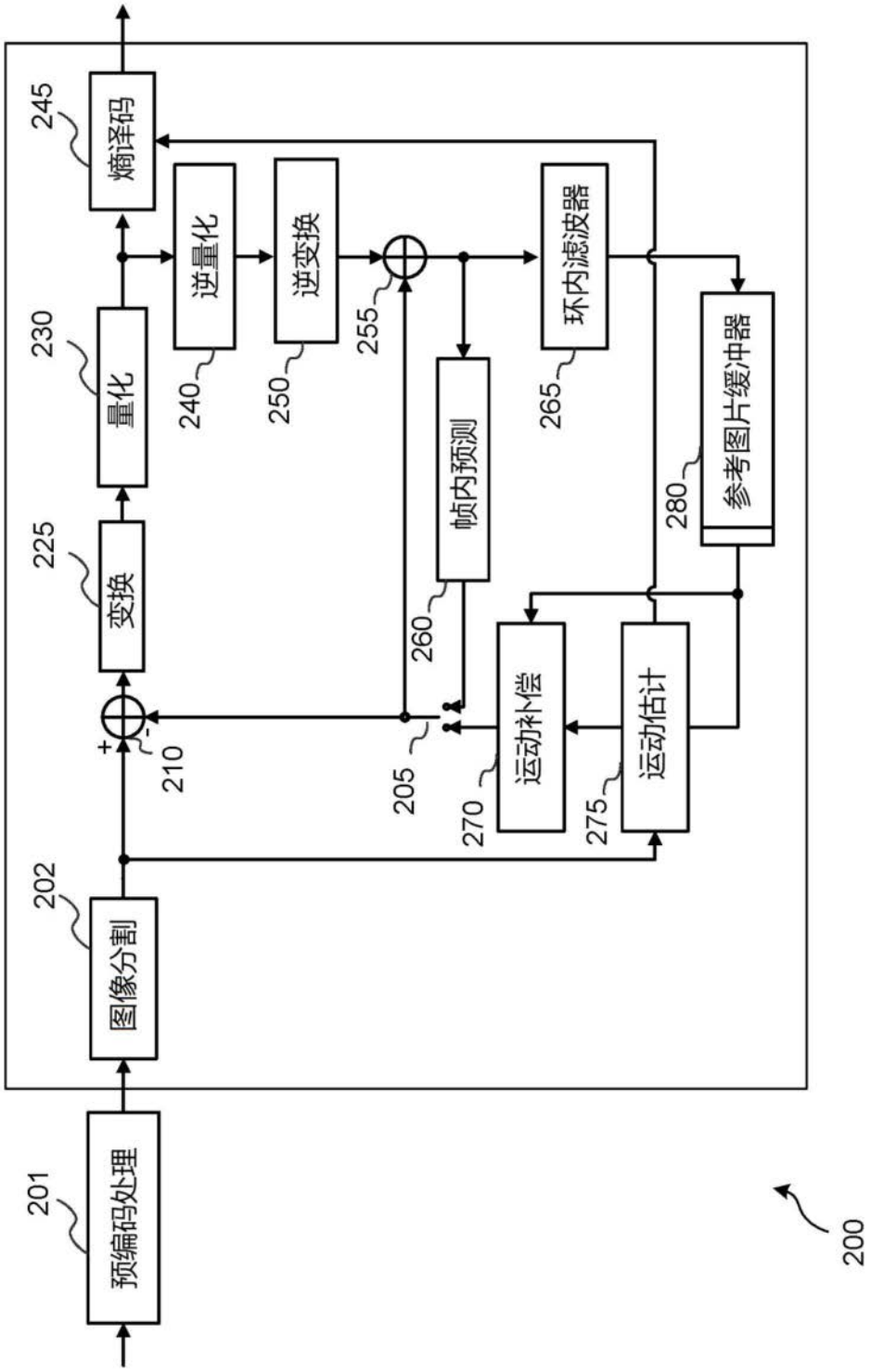

预编码处理
图像分割
变换
量化
熵译码
逆量化
逆变换
环内滤波器
参考图片缓冲器
帧内预测
运动补偿
运动估计
201
202
225
230
245
240
250
255
265
280
260
270
275
205
210
200

图2

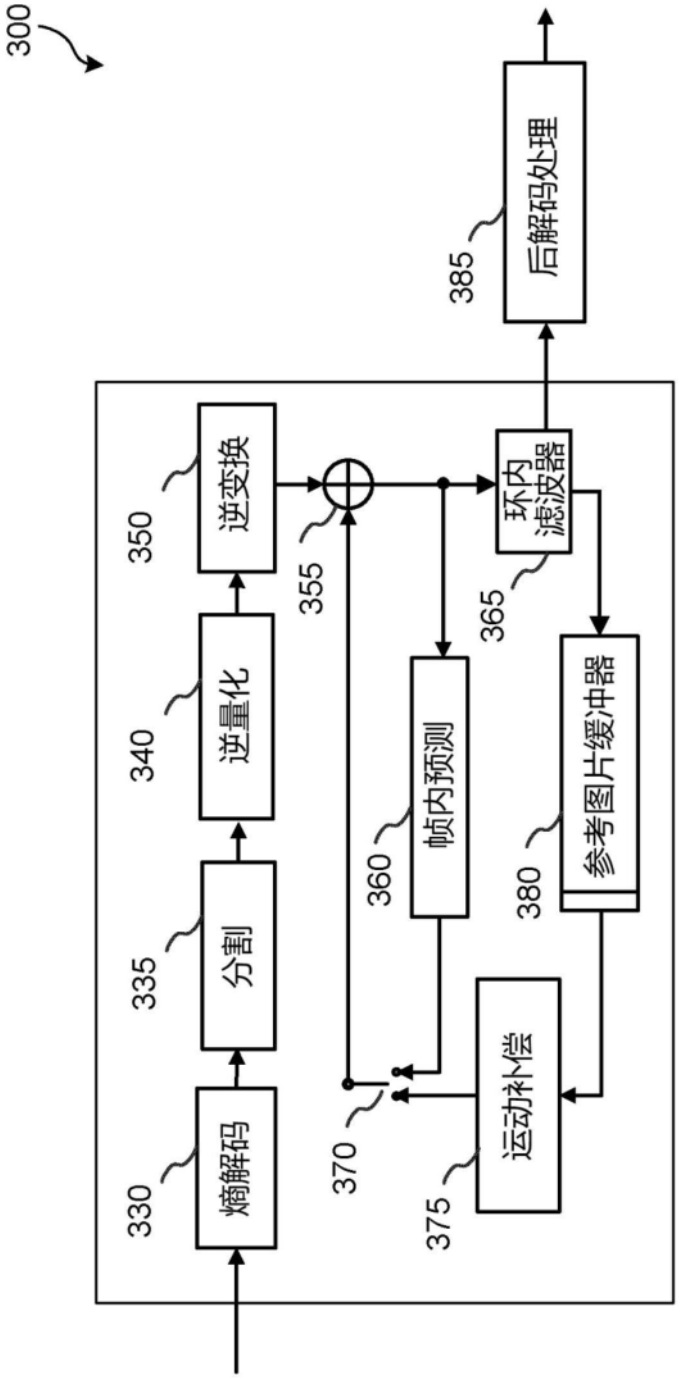

300
385
后解码处理
350
逆变换
355
环内滤波器
365
340
逆量化
360
帧内预测
380
参考图片缓冲器
335
分割
370
375
运动补偿
330
熵解码

图3

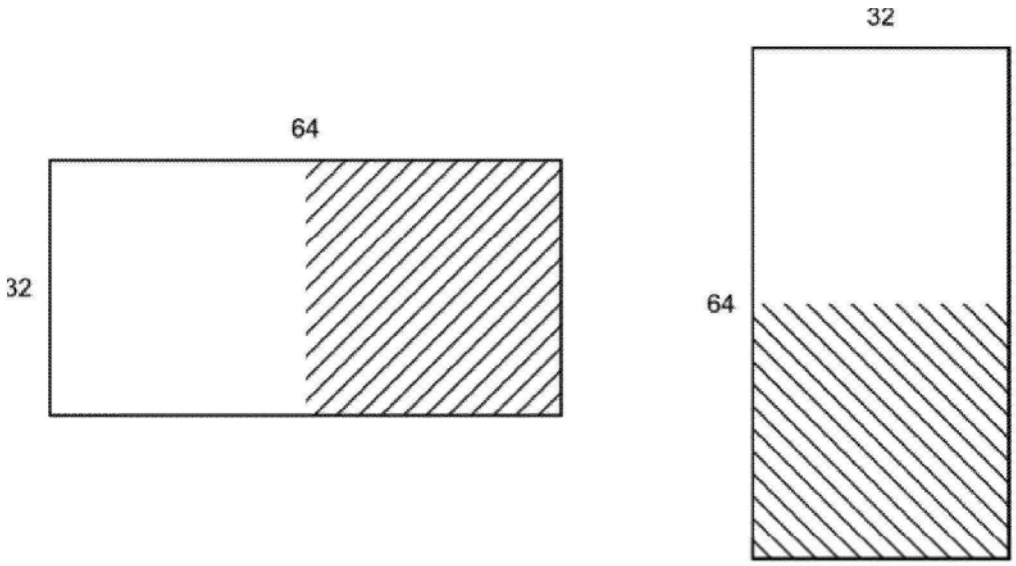


图4

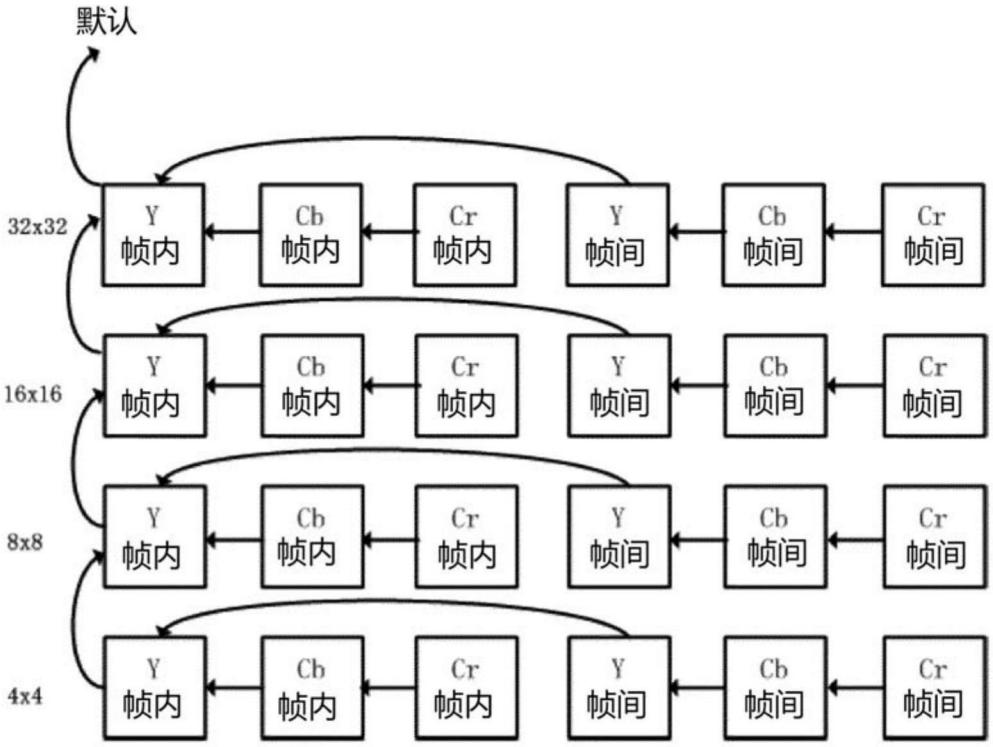


图5

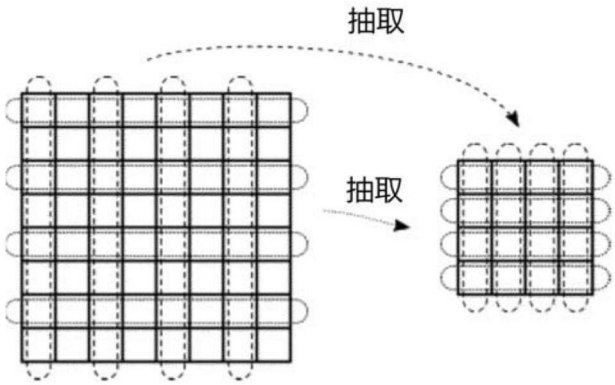


图6

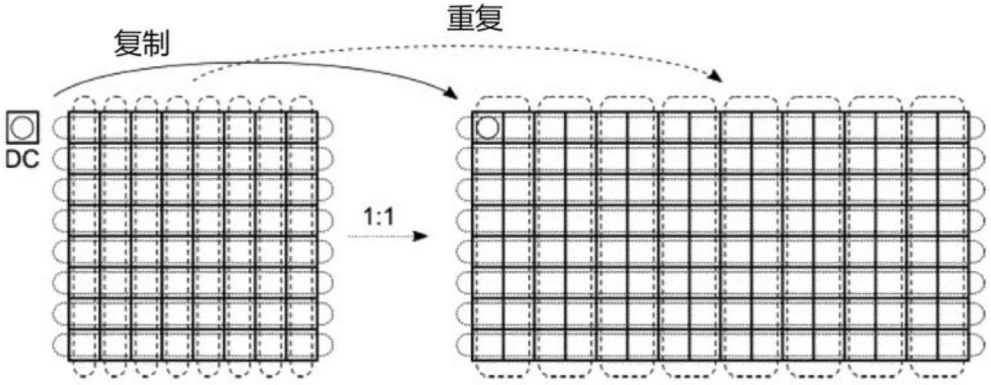


图7

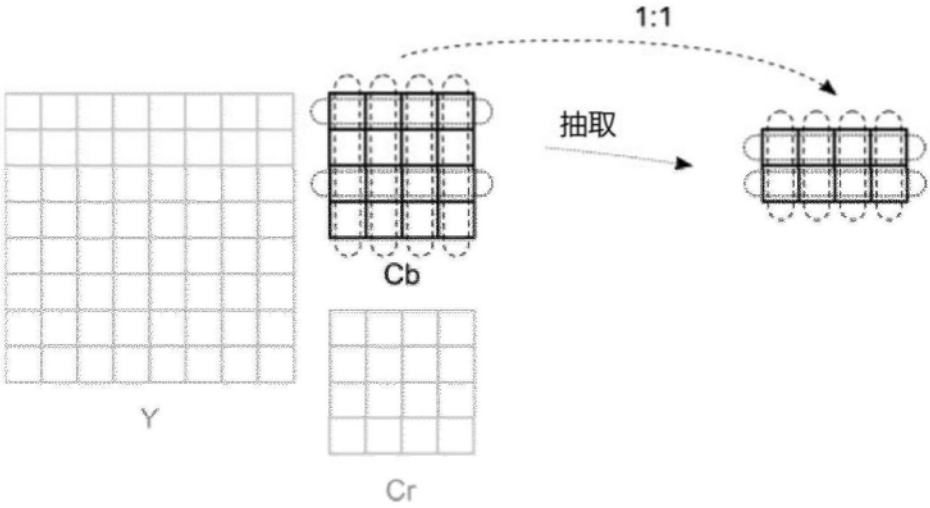


图8

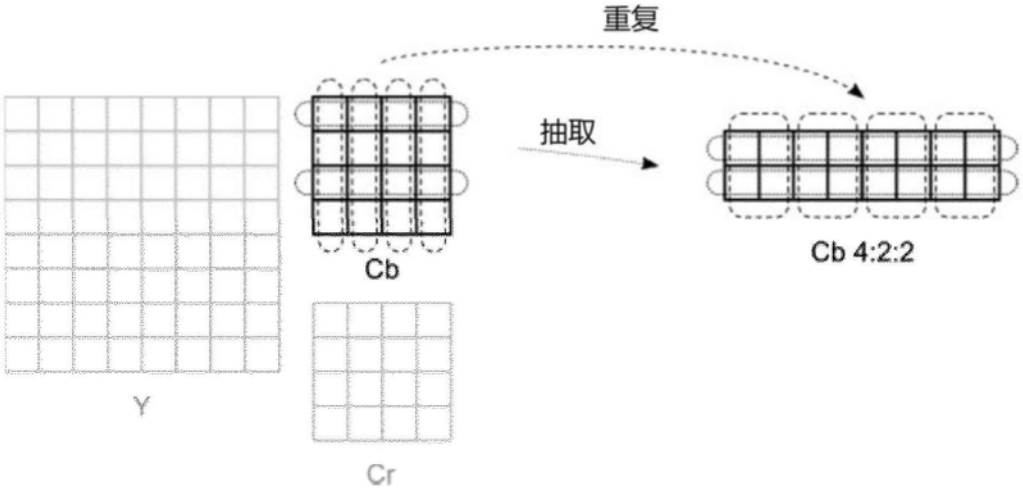


图9

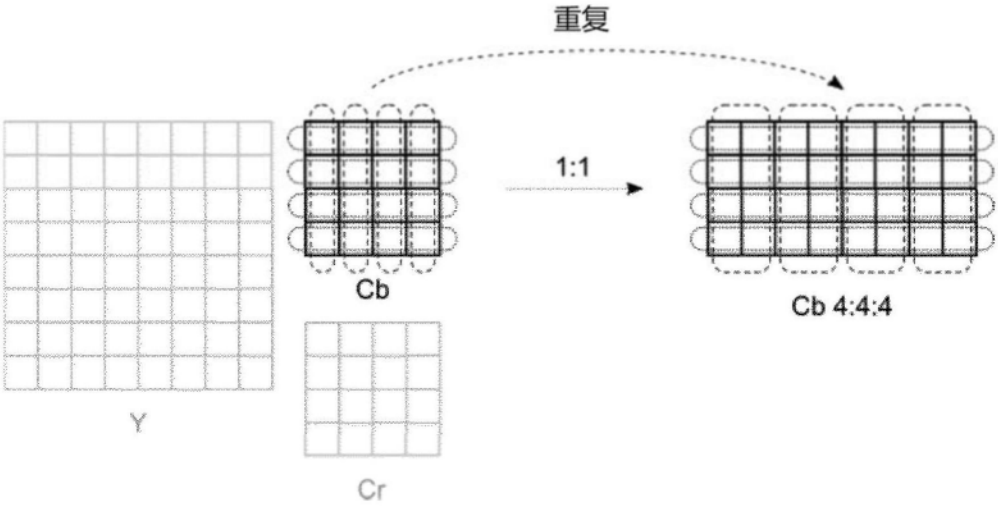


图10

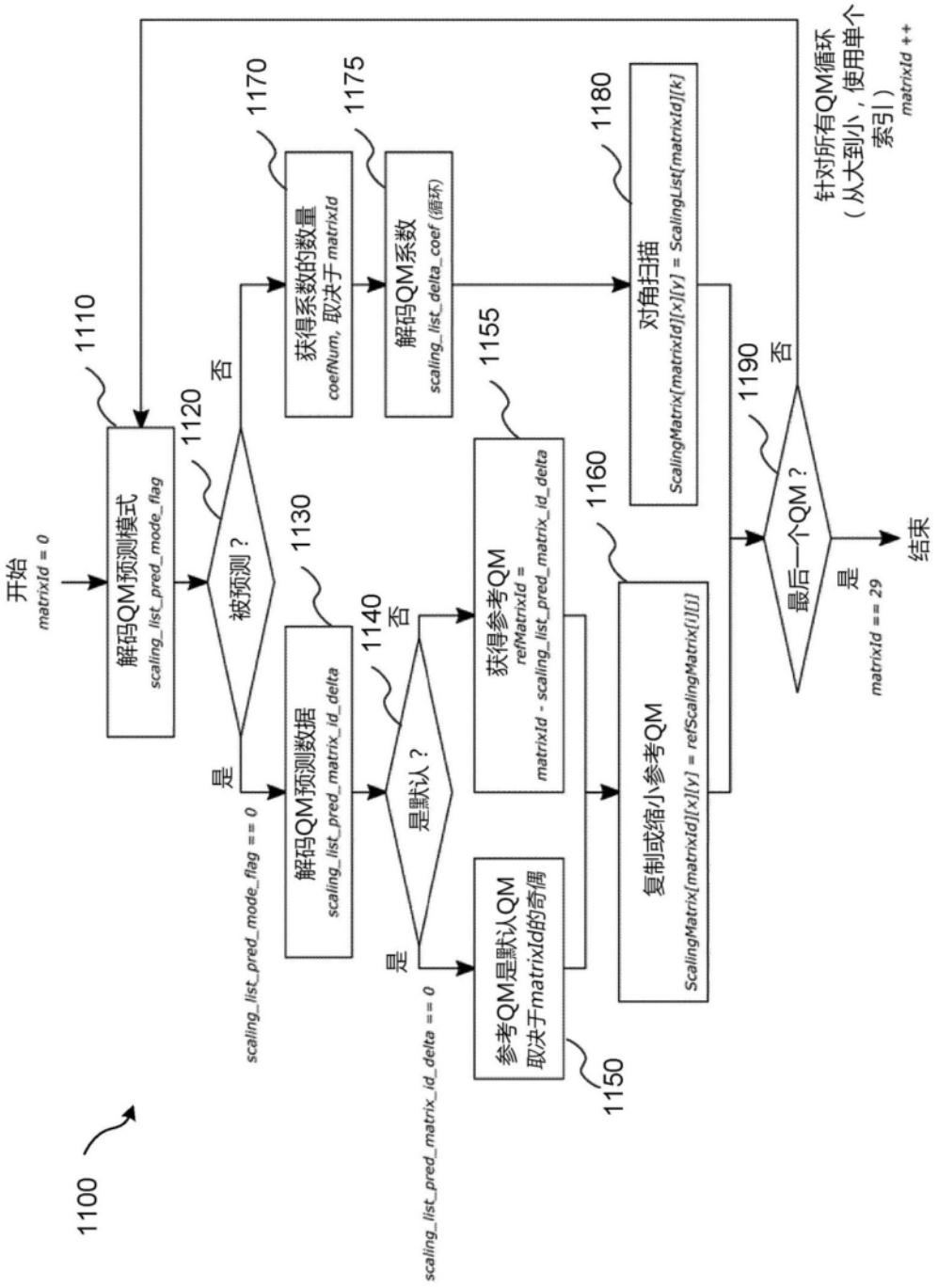

开始
matrixId = 0
解码QM预测模式
scaling_list_pred_mode_flag
1110
被预测 ?
1120
否
是
scaling_list_pred_mode_flag == 0
获得系数的数量
coefNum, 取决于 matrixId
1170
解码QM系数
scaling_list_delta_coef (循环)
1175
解码QM预测数据
scaling_list_pred_matrix_id_delta
1130
是默认 ?
1140
否
是
scaling_list_pred_matrix_id_delta == 0
获得参考QM
refMatrixId =
matrixId - scaling_list_pred_matrix_id_delta
1155
参考QM是默认QM
取决于matrixId的奇偶
1150
复制或缩小参考QM
ScalingMatrix[matrixId][x][y] = refScalingMatrix[i][j]
1160
对角扫描
ScalingMatrix[matrixId][x][y] = ScalingList[matrixId][k]
1180
最后一个QM ?
1190
否
是
matrixId == 29
结束
针对所有QM循环
(从大到小，使用单个索引)
matrixId ++
1100

图11

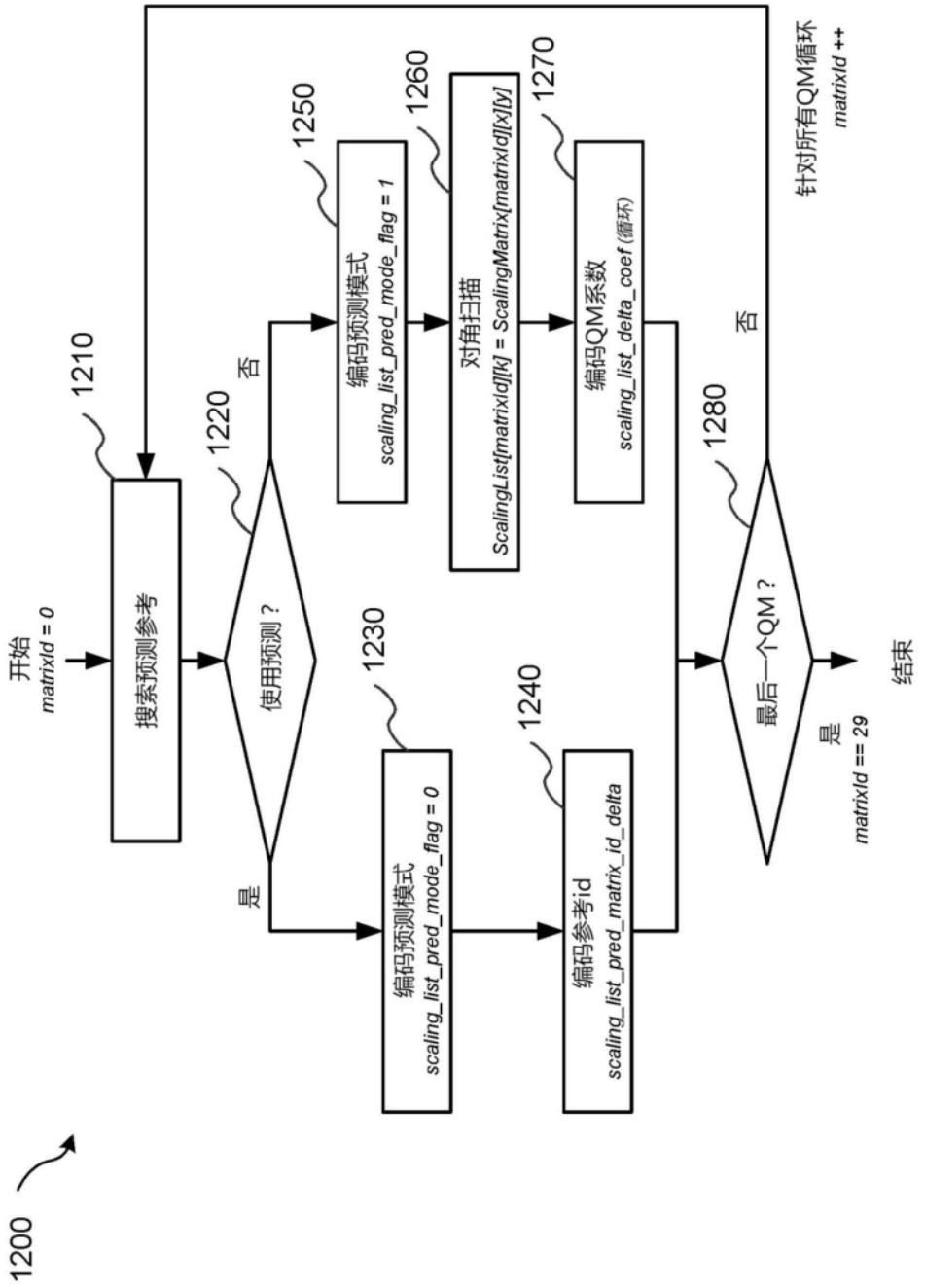

1200
开始
matrixId = 0
1210
搜索预测参考
1220
使用预测？
否
是
1230
编码预测模式
scaling_list_pred_mode_flag = 0
1240
编码参考id
scaling_list_pred_matrix_id_delta
1250
编码预测模式
scaling_list_pred_mode_flag = 1
1260
对角扫描
ScalingList[matrixId][k] = ScalingMatrix[matrixId][x][y]
1270
编码QM系数
scaling_list_delta_coef (循环)
1280
最后一个QM？
否
是
matrixId == 29
针对所有QM循环
matrixId ++
结束

图12

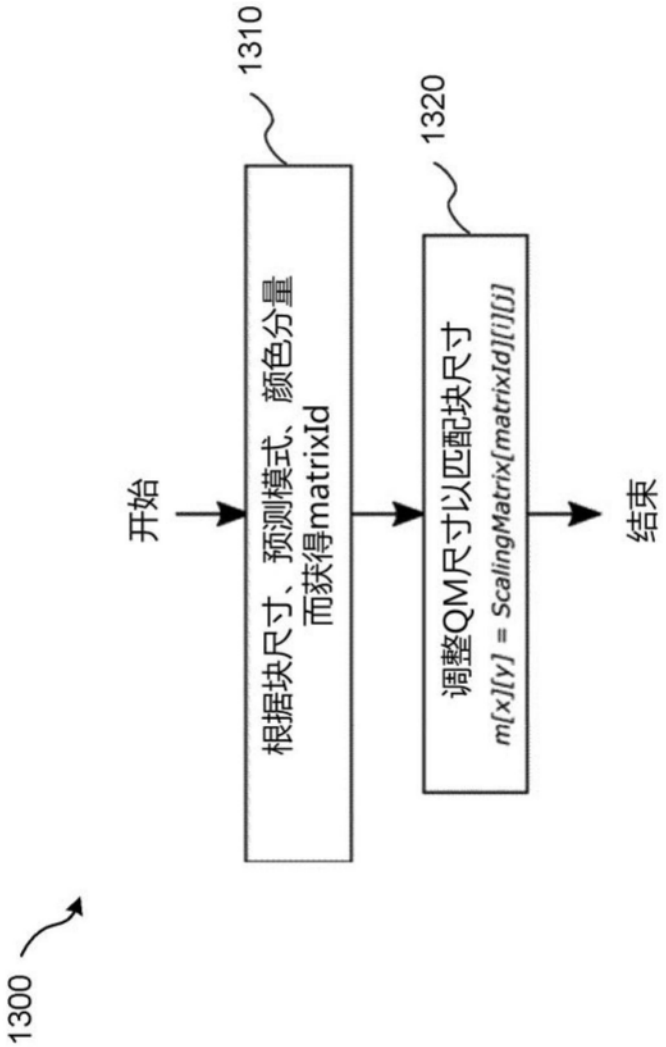

1310
1320
开始
根据块尺寸、预测模式、颜色分量
而获得matrixId
调整QM尺寸以匹配块尺寸
m[x][y] = ScalingMatrix[matrixId][i][j]
结束
1300

图13