(19)国家知识产权局

(12)发明专利

(10)授权公告号 CN 118631988 B
(45)授权公告日 2025.05.16

(21)申请号 202410744572.4

(22)申请日 2020.03.11

(65)同一申请的已公布的文献号
申请公布号 CN 118631988 A

(43)申请公布日 2024.09.10

(30)优先权数据
62/816,751 2019.03.11 US
62/826,659 2019.03.29 US

(62)分案原申请数据
202080020571.5 2020.03.11

(73)专利权人 华为技术有限公司
地址 518129 广东省深圳市龙岗区坂田华为总部办公楼

(72)发明人 王业奎 陈建乐 弗努·亨德里

(74)专利代理机构 北京中博世达专利商标代理有限公司 11274
专利代理师 申健

(51)Int.Cl.
H04N 19/105(2014.01)
H04N 19/117(2014.01)
H04N 19/119(2014.01)
H04N 19/122(2014.01)
H04N 19/132(2014.01)
H04N 19/137(2014.01)
H04N 19/159(2014.01)
H04N 19/169(2014.01)
H04N 19/172(2014.01)
H04N 19/174(2014.01)
H04N 19/176(2014.01)
H04N 19/184(2014.01)
H04N 19/186(2014.01)
H04N 19/46(2014.01)
H04N 19/52(2014.01)
H04N 19/593(2014.01) (续)

(56)对比文件
Jianle Chen et al..Document: JVET-M0136-v1:AHG12: Treating tile and tile group boundariesas picture boundaries.《Joint Video Experts Team (JVET) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11 13th Meeting: Marrakech, MA, 9-18 Jan. 2019》.2019,正文第1节.

审查员 于典

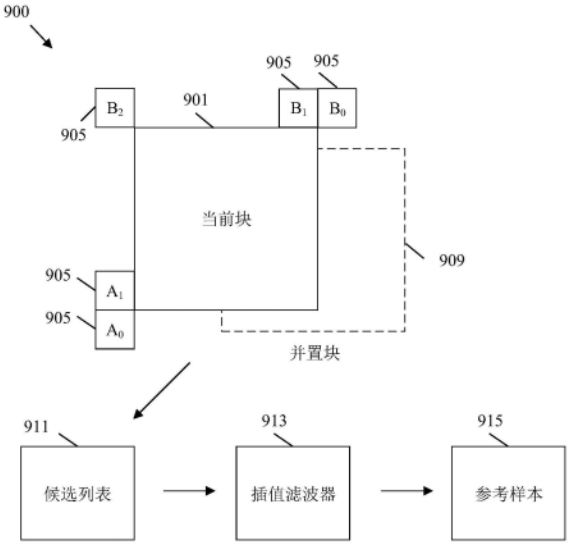
权利要求书2页 说明书53页 附图14页

(54)发明名称

用于子图像运动矢量的插值滤波器限幅

(57)摘要

本发明公开了一种视频译码机制。所述机制包括:接收包括当前图像的码流,其中,所述当前图像包括被根据帧间预测进行译码的子图像;确定所述子图像中的块的运动矢量;当所述运动矢量指向所述子图像外部时以及当一个标志被设置为表示所述子图像当作图像处理时,将限幅函数应用于参考块中的样本位置,以支持应用插值滤波器;将所述插值滤波器应用于所述限幅函数的结果,以获得预测样本值;根据所述预测样本值对所述块进行解码;转发所述块,用于显示为经解码视频序列的一部分。



CN 118631988 B

[转续页]

[接上页]

(51)Int.Cl.

H04N 19/70(2014.01)
H04N 19/82(2014.01)
H04N 19/86(2014.01)
H04N 19/96(2014.01)
G06V 10/62(2022.01)
G06V 20/40(2022.01)

1.一种在解码器中实现的方法,其特征在于,所述方法包括:

接收包括当前图像的码流,其中,所述当前图像包括根据帧间预测进行译码的子图像,所述子图像包括多个分块,且所述子图像不是所述分块;

解析所述码流,以获得量化变换系数;

对所述量化变换系数进行逆量化和逆变换,以得到残差块;

确定所述子图像中的块的运动矢量;

当所述运动矢量指向所述子图像外部时以及当一个标志表示所述子图像当作图像处理时,将限幅函数(clipping function)应用于参考块中的样本位置;

将插值滤波器应用于所述限幅函数的结果,以获得预测样本值;

根据所述残差块和预测块,得到重建的图像块,所述预测块包括所述预测样本值;

其中,所述插值滤波器包括亮度样本双线性插值过程,所述块包括亮度样本块,所述预测样本值包括预测亮度样本值;

所述亮度样本双线性插值过程接收包括具有整数个样本单位的亮度位置(xIntL,yIntL)的输入,所述亮度样本双线性插值过程输出预测亮度样本值(predSampleLXL),所述限幅函数根据以下内容应用于所述样本位置:

当subpic_treated_as_pic_flag[SubPicIdx]等于1时,以下内容适用:

$xInti = Clip3(SubPicLeftBoundaryPos, SubPicRightBoundaryPos, xInt_L + i)$,

$yInti = Clip3(SubPicTopBoundaryPos, SubPicBotBoundaryPos, yInt_L + i)$,

其中,subpic_treated_as_pic_flag表示所述标志,SubPicIdx表示所述子图像的索引,xInti和yInti表示索引i处的限幅后样本位置,SubPicRightBoundaryPos表示所述子图像的右边界的位置,SubPicLeftBoundaryPos表示所述子图像的左边界的位置,SubPicTopBoundaryPos表示所述子图像的上边界的位置,SubPicBotBoundaryPos表示所述子图像的下边界的位置,Clip3表示根据下式的所述限幅函数:

$$Clip3(x,y,z)=\begin{cases} x & ; \quad z<x \\ y & ; \quad z>y, \\ z & ; \quad 其它 \end{cases}$$

其中,x、y和z是数字输入值。

2.根据权利要求1中所述的方法,其特征在于,所述插值滤波器包括亮度样本8抽头插值滤波过程,所述块包括亮度样本块,所述预测样本值包括预测亮度样本值。

3.根据权利要求1中所述的方法,其特征在于,所述插值滤波器包括色度样本插值过程,所述块包括色度样本块,所述预测样本值包括预测色度样本值。

4.一种视频译码设备,其特征在于,所述视频译码设备包括:

处理器、与所述处理器耦合的接收器、与所述处理器耦合的存储器和与所述处理器耦合的发送器,其中,所述处理器、接收器、存储器和发送器用于执行根据权利要求1至3中任一项所述的方法。

5.一种非瞬时性计算机可读介质,其特征在于,所述非瞬时性计算机可读介质包括供视频译码设备使用的计算机程序产品,所述计算机程序产品包括存储在所述非瞬时性计算机可读介质中的计算机可执行指令,当处理器执行所述计算机可执行指令时,使得所述视频译码设备执行根据权利要求1至3中任一项所述的方法。

6.一种解码器,其特征在于,所述解码器包括:

接收模块,用于接收包括当前图像的码流,其中,所述当前图像包括根据帧间预测进行译码的子图像,所述子图像包括多个分块,且所述子图像不是所述分块;

解析模块,用于解析所述码流,以获得量化变换系数;

处理模块,用于对所述量化变换系数进行逆量化和逆变换,以得到残差块;

确定模块,用于确定所述子图像中的块的运动矢量;

应用模块,用于:

当所述运动矢量指向所述子图像外部时以及当一个标志表示所述子图像当作图像处理时,将限幅函数应用于参考块中的样本位置;

将插值滤波器应用于所述限幅函数的结果,以获得预测样本值;

重建模块,用于根据所述残差块和预测块,得到重建的图像块,所述预测块包括所述预测样本值;

其中,所述插值滤波器包括亮度样本双线性插值过程,所述块包括亮度样本块,所述预测样本值包括预测亮度样本值;

所述亮度样本双线性插值过程接收包括具有整数个样本单位的亮度位置(xIntL,yIntL)的输入,所述亮度样本双线性插值过程输出预测亮度样本值(predSampleLXL),所述限幅函数根据以下内容应用于所述样本位置:

当subpic_treated_as_pic_flag[SubPicIdx]等于1时,以下内容适用:

$xInti = Clip3(SubPicLeftBoundaryPos, SubPicRightBoundaryPos, xInt_L + i)$,

$yInti = Clip3(SubPicTopBoundaryPos, SubPicBotBoundaryPos, yInt_L + i)$,

其中,subpic_treated_as_pic_flag表示所述标志,SubPicIdx表示所述子图像的索引,xInti和yInti表示索引i处的限幅后样本位置,SubPicRightBoundaryPos表示所述子图像的右边界的位置,SubPicLeftBoundaryPos表示所述子图像的左边界的位置,SubPicTopBoundaryPos表示所述子图像的上边界的位置,SubPicBotBoundaryPos表示所述子图像的下边界的位置,Clip3表示根据下式的所述限幅函数:

$$Clip3(x,y,z)=\begin{cases} x & ; \quad z<x \\ y & ; \quad z>y, \\ z & ; \quad 其它 \end{cases}$$

其中,x、y和z是数字输入值。

7.根据权利要求6所述的解码器,其特征在于,所述解码器还用于执行根据权利要求2或3所述的方法。

8.一种计算机程序产品,其特征在于,所述计算机程序产品包括计算机可执行指令,当处理器执行所述计算机可执行指令时,使得所述处理器执行根据权利要求1至3中任一项所述的方法。

用于子图像运动矢量的插值滤波器限幅

[0001] 本申请是分案申请,原申请的申请号是202080020571.5,原申请日是2020年03月11日,原申请的全部内容通过引用结合在本申请中。

[0002] 相关申请交叉引用

[0003] 本专利申请要求王业奎(Ye-Kui Wang)等人于2019年3月11日提交的申请号为62/816,751、发明名称为“基于子图像的视频译码(Sub-Picture Based Video Coding)”的美国临时专利申请的权益以及要求王业奎(Ye-Kui Wang)等人于2019年3月29日提交的申请号为62/826,659、发明名称为“基于子图像的视频译码(Sub-Picture Based Video Coding)”的美国临时专利申请的权益,其全部内容通过引用结合在本申请中。

技术领域

[0004] 本发明大体上涉及视频译码,具体涉及在视频译码中对图像中的子图像进行译码。

背景技术

[0005] 即使视频相对较短,也需要大量的视频数据来描述,当数据要在带宽容量受限的通信网络中进行流式传输或以其它方式传输时,这样可能会造成困难。因此,视频数据通常要先压缩,然后通过现代电信网络进行传输。由于内存资源可能有限,当在存储设备上存储视频时,该视频的大小也可能是一个问题。视频压缩设备通常在源侧使用软件和/或硬件对视频数据进行编码,然后进行传输或存储,从而减少表示数字视频图像所需的数据量。然后,对视频数据进行解码的视频解压缩设备在目的地侧接收压缩数据。在网络资源有限以及对更高视频质量的需求不断增长的情况下,需要改进压缩和解压缩技术,这些改进的技术在几乎不影响图像质量的情况下能够提高压缩比。

发明内容

[0006] 在一个实施例中,本发明包括一种在解码器中实现的方法。所述方法包括:所述解码器的接收器接收包括当前图像的码流,其中,所述当前图像包括被根据帧间预测进行译码的子图像;所述解码器的处理器确定所述子图像中的块的运动矢量;当所述运动矢量指向所述子图像外部时以及当一个标志被设置为表示所述子图像当作图像处理时,所述处理器将限幅函数应用于参考块中的样本位置,以支持应用插值滤波器;所述处理器将所述插值滤波器应用于所述限幅函数的结果,以获得预测样本值;所述处理器根据所述预测样本值对所述块进行解码。帧间预测可以根据几种帧间预测模式中的一种来执行。某些帧间预测模式在编码器和解码器侧都生成运动矢量预测值的候选列表。这样,编码器可以通过指示候选列表中的索引来指示运动矢量,而不是指示运动矢量本身。此外,一些系统对子图像进行解码以进行独立提取。独立提取使得当前子图像可以在其它子图像中的信息没有进行解码的情况下进行解码并显示。这种情况可能会在使用指向子图像外部的运动矢量时产生错误,因为运动矢量指向的数据可能没有解码,因此不能使用。本发明包括一种表示子图像

可以当作图像处理的标志。在当前子图像当作图像处理时,当前子图像可以在不参考其它子图像的情况下提取。具体地,本示例使用在应用插值滤波器时应用的限幅函数。这种限幅函数确保插值滤波器不依赖于相邻子图像中的数据,以保持子图像分开,从而支持单独提取。因此,当设置有该标志且运动矢量指向当前子图像外部时,应用限幅函数。然后,将插值滤波器应用于限幅函数的结果。因此,本示例为视频编解码器提供了额外功能,即防止在执行子图像提取时出现错误。

[0007] 可选地,根据任一上述方面,在所述方面的另一种实现方式中,所述插值滤波器包括亮度样本双线性插值过程,所述块包括亮度样本块,所述预测样本值包括预测亮度样本值。

[0008] 可选地,根据任一上述方面,在所述方面的另一种实现方式中,所述亮度样本双线性插值过程接收包括具有整数个样本单位的亮度位置(xIntL,yIntL)的输入,所述亮度样本双线性插值过程输出预测亮度样本值(predSampleLXL),所述限幅函数根据以下内容应用于所述样本位置:当subpic_treated_as_pic_flag[SubPicIdx]等于1时,以下内容适用:

[0009] xInti=Clip3(SubPicLeftBoundaryPos,SubPicRightBoundaryPos,xIntL+i),

[0010] yInti=Clip3(SubPicTopBoundaryPos,SubPicBotBoundaryPos,yIntL+i),

[0011] 其中,subpic_treated_as_pic_flag表示所述被设置为表示子图像当作子图像处理的标志,SubPicIdx表示子图像的索引,xInti和yInti表示索引i处的限幅后样本位置,SubPicRightBoundaryPos表示子图像的右边界的位置,SubPicLeftBoundaryPos表示子图像的左边界的位置,SubPicTopBoundaryPos表示子图像的上边界的位置,SubPicBotBoundaryPos表示子图像的下边界的位置,Clip3表示根据下式的限幅函数:

[0012] $$\mathrm{Clip3}(x,y,z)=\begin{cases}x & ; \ z<x\\ y & ; \ z>y,\\ z & ; \ \text{其它}\end{cases}$$

[0013] 其中,x、y和z是数字输入值。

[0014] 可选地,根据任一上述方面,在所述方面的另一种实现方式中,所述插值滤波器包括亮度样本8抽头插值滤波过程,所述块包括亮度样本块,所述预测样本值包括预测亮度样本值。

[0015] 可选地,根据任一上述方面,在所述方面的另一种实现方式中,所述亮度样本8抽头插值滤波过程接收包括具有整数个样本单位的亮度位置(xIntL,yIntL)的输入,所述亮度样本8抽头插值滤波过程输出预测亮度样本值(predSampleLXL),所述限幅函数根据以下内容应用于所述样本位置:当subpic_treated_as_pic_flag[SubPicIdx]等于1时,以下内容适用:

[0016] xInti=Clip3(SubPicLeftBoundaryPos,SubPicRightBoundaryPos,xIntL+i-3),

[0017] yInti=Clip3(SubPicTopBoundaryPos,SubPicBotBoundaryPos,yIntL+i-3),

[0018] 其中,subpic_treated_as_pic_flag表示所述被设置为表示子图像当作子图像处理的标志,SubPicIdx表示子图像的索引,xInti和yInti表示索引i处的限幅后样本位置,SubPicRightBoundaryPos表示子图像的右边界的位置,SubPicLeftBoundaryPos表示子图像的左边界的位置,SubPicTopBoundaryPos表示子图像的上边界的位置,SubPicBotBoundaryPos表示子图像的下边界的位置,Clip3为根据下式的所述限幅函数:

[0019] $$Clip3(x,y,z)=\begin{cases}x & ; \quad z<x\\ y & ; \quad z>y,\\ z & ; \quad 其它\end{cases}$$

[0020] 其中,x、y和z是数字输入值。

[0021] 可选地,根据任一上述方面,在所述方面的另一种实现方式中,所述插值滤波器包括色度样本插值过程,所述块包括色度样本块,所述预测样本值包括预测色度样本值。

[0022] 可选地,根据任一上述方面,在所述方面的另一种实现方式中,所述色度样本插值过程接收包括具有整数个样本单位的色度位置(xIntC,yIntC)的输入,所述色度样本插值过程输出预测色度样本值(predSampleLXC),所述限幅函数根据以下内容应用于所述样本位置:当subpic_treated_as_pic_flag[SubPicIdx]等于1时,以下内容适用:

[0023] xInti＝Clip3(SubPicLeftBoundaryPos/SubWidthC,SubPicRightBoundaryPos/SubWidthC,xIntC+i),

[0024] yInti＝Clip3(SubPicTopBoundaryPos/SubHeightC,SubPicBotBoundaryPos/SubHeightC,yIntC+i),

[0025] 其中,subpic_treated_as_pic_flag表示所述被设置为表示所述子图像当作图像处理的标志,SubPicIdx表示所述子图像的索引,xInti和yInti表示索引i处的限幅后样本位置,SubPicRightBoundaryPos表示所述子图像的右边界的位置,SubPicLeftBoundaryPos表示所述子图像的左边界的位置,SubPicTopBoundaryPos表示所述子图像的上边界的位置,SubPicBotBoundaryPos表示所述子图像的下边界的位置,SubWidthC和SubHeightC表示亮度样本与色度样本之间的水平采样率比值和垂直采样率比值,Clip3表示根据下式的所述限幅函数:

[0026] $$Clip3(x,y,z)=\begin{cases}x & ; \quad z<x\\ y & ; \quad z>y,\\ z & ; \quad 其它\end{cases}$$

[0027] 其中,x、y和z是数字输入值。

[0028] 在一个实施例中,本发明包括一种在编码器中实现的方法。所述方法包括:所述编码器的处理器将当前图像分割成子图像并将所述子图像分割成块;所述处理器确定根据帧间预测对所述块进行编码;所述处理器选择运动矢量对所述块进行编码;当所述运动矢量指向所述子图像外部时以及当一个标志被设置为表示所述子图像当作图像处理时,所述处理器将限幅函数应用于参考块中的样本位置,以支持应用插值滤波器;所述处理器将所述插值滤波器应用于所述限幅函数的结果,以获得预测样本值;所述处理器根据所述预测样本值和所述运动矢量将所述块编码到码流中;与所述处理器耦合的存储器存储所述码流,用于发送给解码器。帧间预测可以根据几种帧间预测模式中的一种来执行。某些帧间预测模式在编码器和解码器侧都生成运动矢量预测值的候选列表。这样,编码器可以通过指示候选列表中的索引来指示运动矢量,而不是指示运动矢量本身。此外,一些系统对子图像进行编码以进行独立提取。独立提取使得当前子图像可以在其它子图像中的信息没有进行编码的情况下进行解码并显示。这种情况可能会在使用指向子图像外部的运动矢量时产生错误,因为运动矢量指向的数据可能没有编码,因此不能使用。本发明包括一种表示子图像可以当作图像处理的标志。在当前子图像当作图像处理时,当前子图像可以在不参考其它子图像的情况下提取。具体地,本示例使用在应用插值滤波器时应用的限幅函数。这种限幅函

数确保插值滤波器不依赖于相邻子图像中的数据,以保持子图像分开,从而支持单独提取。因此,当设置有该标志且运动矢量指向当前子图像外部时,应用限幅函数。然后,将插值滤波器应用于限幅函数的结果。因此,本示例为视频编解码器提供了额外功能,即防止在执行子图像提取时出现错误。

[0029] 可选地,根据任一上述方面,在所述方面的另一种实现方式中,所述插值滤波器包括亮度样本双线性插值过程,所述块包括亮度样本块,所述预测样本值包括预测亮度样本值。

[0030] 可选地,根据任一上述方面,在所述方面的另一种实现方式中,所述亮度样本双线性插值过程接收包括具有整数个样本单位的亮度位置(xIntL,yIntL)的输入,所述亮度样本双线性插值过程输出预测亮度样本值(predSampleLXL),所述限幅函数根据以下内容应用于所述样本位置:当subpic_treated_as_pic_flag[SubPicIdx]等于1时,以下内容适用:

[0031] xInti=Clip3(SubPicLeftBoundaryPos,SubPicRightBoundaryPos,xIntL+i),

[0032] yInti=Clip3(SubPicTopBoundaryPos,SubPicBotBoundaryPos,yIntL+i),

[0033] 其中,subpic_treated_as_pic_flag表示所述被设置为表示子图像当作子图像处理的标志,SubPicIdx表示子图像的索引,xInti和yInti表示索引i处的限幅后样本位置,SubPicRightBoundaryPos表示子图像的右边界的位置,SubPicLeftBoundaryPos表示子图像的左边界的位置,SubPicTopBoundaryPos表示子图像的上边界的位置,SubPicBotBoundaryPos表示子图像的下边界的位置,Clip3表示根据下式的所述限幅函数:

[0034] $$\mathrm{Clip3}(x,y,z)=\begin{cases}x & ; \ z<x\\ y & ; \ z>y,\\ z & ; \ \text{其它}\end{cases}$$

[0035] 其中,x、y和z是数字输入值。

[0036] 可选地,根据任一上述方面,在所述方面的另一种实现方式中,所述插值滤波器包括亮度样本8抽头插值滤波过程,所述块包括亮度样本块,所述预测样本值包括预测亮度样本值。

[0037] 可选地,根据任一上述方面,在所述方面的另一种实现方式中,所述亮度样本8抽头插值滤波过程接收包括具有整数个样本单位的亮度位置(xIntL,yIntL)的输入,所述亮度样本8抽头插值滤波过程输出预测亮度样本值(predSampleLXL),所述限幅函数根据以下内容应用于所述样本位置:当subpic_treated_as_pic_flag[SubPicIdx]等于1时,以下内容适用:

[0038] xInti=Clip3(SubPicLeftBoundaryPos,SubPicRightBoundaryPos,xIntL+i-3),

[0039] yInti=Clip3(SubPicTopBoundaryPos,SubPicBotBoundaryPos,yIntL+i-3),

[0040] 其中,subpic_treated_as_pic_flag表示所述被设置为表示子图像当作子图像处理的标志,SubPicIdx表示子图像的索引,xInti和yInti表示索引i处的限幅后样本位置,SubPicRightBoundaryPos表示子图像的右边界的位置,SubPicLeftBoundaryPos表示子图像的左边界的位置,SubPicTopBoundaryPos表示子图像的上边界的位置,SubPicBotBoundaryPos表示子图像的下边界的位置,Clip3表示根据下式的所述限幅函数:

[0041] $$\mathrm{Clip3}(x,y,z)=\begin{cases}x & ; \ z<x\\ y & ; \ z>y,\\ z & ; \ \text{其它}\end{cases}$$

[0042] 其中,x、y和z是数字输入值。

[0043] 可选地,根据任一上述方面,在所述方面的另一种实现方式中,所述插值滤波器包括色度样本插值过程,所述块包括色度样本块,所述预测样本值包括预测色度样本值。

[0044] 可选地,根据任一上述方面,在所述方面的另一种实现方式中,所述色度样本插值过程接收包括具有整数个样本单位的色度位置(xIntC,yIntC)的输入,所述色度样本插值过程输出预测色度样本值(predSampleLXC),所述限幅函数根据以下内容应用于所述样本位置:当subpic_treated_as_pic_flag[SubPicIdx]等于1时,以下内容适用:

[0045] xInti=Clip3(SubPicLeftBoundaryPos/SubWidthC,SubPicRightBoundaryPos/SubWidthC,xIntC+i),

[0046] yInti=Clip3(SubPicTopBoundaryPos/SubHeightC,SubPicBotBoundaryPos/SubHeightC,yIntC+i),

[0047] 其中,subpic_treated_as_pic_flag表示所述被设置为表示所述子图像当作图像处理的标志,SubPicIdx表示所述子图像的索引,xInti和yInti表示索引i处的限幅后样本位置,SubPicRightBoundaryPos表示所述子图像的右边界的位置,SubPicLeftBoundaryPos表示所述子图像的左边界的位置,SubPicTopBoundaryPos表示所述子图像的上边界的位置,SubPicBotBoundaryPos表示所述子图像的下边界的位置,SubWidthC和SubHeightC表示亮度样本与色度样本之间的水平采样率比值和垂直采样率比值,Clip3表示根据下式的所述限幅函数:

[0048] $$\text{Clip3}(x,y,z)=\begin{cases}x & ; \quad z<x\\ y & ; \quad z>y,\\ z & ; \quad \text{其它}\end{cases}$$

[0049] 其中,x、y和z是数字输入值。

[0050] 在一个实施例中,本发明包括一种视频译码设备。所述视频译码设备包括处理器、与所述处理器耦合的接收器、与所述处理器耦合的存储器和与所述处理器耦合的发送器,其中,所述处理器、接收器、存储器和发送器用于执行根据任一上述方面所述的方法。

[0051] 在一个实施例中,本发明包括一种非瞬时性计算机可读介质。所述非瞬时性计算机可读介质包括供视频译码设备使用的计算机程序产品,所述计算机程序产品包括存储在所述非瞬时性计算机可读介质中的计算机可执行指令,当处理器执行所述计算机可执行指令时,使得所述视频译码设备执行根据任一上述方面所述的方法。

[0052] 在一个实施例中,本发明包括一种解码器。所述解码器包括:接收模块,用于接收包括当前图像的码流,其中,所述当前图像包括被根据帧间预测进行译码的子图像;确定模块,用于确定所述子图像中的块的运动矢量;应用模块,用于:当所述运动矢量指向所述子图像外部时以及当一个标志被设置为表示所述子图像当作图像处理时,将限幅函数应用于参考块中的样本位置,以支持应用插值滤波器;将所述插值滤波器应用于所述限幅函数的结果,以获得预测样本值;解码模块,用于根据所述预测样本值对所述块进行解码;转发模块,用于转发所述块,用于显示为经解码视频序列的一部分。

[0053] 可选地,根据任一上述方面,在所述方面的另一种实现方式中,所述解码器还用于执行根据任一上述方面所述的方法。

[0054] 在一个实施例中,本发明包括一种编码器。所述编码器包括:分割模块,用于将当

前图像分割成子图像并将所述子图像分割成块;确定模块,用于确定根据帧间预测对所述块进行编码;选择模块,用于选择运动矢量对所述块进行编码;应用模块,用于:当所述运动矢量指向所述子图像外部时以及当一个标志被设置为表示所述子图像当作图像处理时,将限幅函数应用于参考块中的样本位置,以支持应用插值滤波器;将所述插值滤波器应用于所述限幅函数的结果,以获得预测样本值;编码模块,用于根据所述预测样本值和所述运动矢量将所述块编码到码流中;存储模块,用于存储所述码流,用于发送给解码器。

[0055] 可选地,根据任一上述方面,在所述方面的另一种实现方式中,所述编码器还用于执行根据任一上述方面所述的方法。

[0056] 可选地,根据任一上述方面,在所述方面的另一种实现方式中,所述编码器还用于执行根据任一上述方面所述的方法。

[0057] 为了清楚起见,任一上述实施例可以与上述其它任一或多个实施例组合以创建在本发明范围内的新实施例。

[0058] 根据以下结合附图和权利要求书进行的具体实施方式,将更清楚地理解这些和其它特征。

附图说明

[0059] 为了更全面地理解本发明,现在参考以下结合附图和具体实施方式进行的简要描述,其中,相似的附图标记表示相似的部件。

[0060] 图1为对视频信号进行译码的示例性方法的流程图。

[0061] 图2为用于视频译码的示例性编码和解码(编解码)系统的示意图。

[0062] 图3为示例性视频编码器的示意图。

[0063] 图4为示例性视频解码器的示意图。

[0064] 图5A为分割成子图像的示例性图像的示意图。

[0065] 图5B为分割成条带的示例性子图像的示意图。

[0066] 图5C为分割成分块的示例性条带的示意图。

[0067] 图5D为分割成编码树单元(coding tree unit,CTU)的示例性条带的示意图。

[0068] 图6为单向帧间预测的一个示例的示意图。

[0069] 图7为双向帧间预测的一个示例的示意图。

[0070] 图8为根据来自相邻经译码块的候选运动矢量对当前块进行译码的一个示例的示意图。

[0071] 图9为用于确定运动矢量候选列表的示例性模式的示意图。

[0072] 图10为示例性环内滤波器的框图。

[0073] 图11为包括译码工具参数以支持对图像中的子图像进行解码的示例性码流的示意图。

[0074] 图12为示例性视频译码设备的示意图。

[0075] 图13为在子图像当作图像处理时将限幅函数应用于插值滤波器时将视频序列编码到码流中的示例性方法的流程图。

[0076] 图14为在子图像当作图像处理时将限幅函数应用于插值滤波器时从码流中解码出视频序列的示例性方法的流程图。

[0077] 图15为在子图像当作图像处理时将限幅函数应用于插值滤波器时对码流中的图像组成的视频序列进行译码的示例性系统的示意图。

具体实施方式

[0078] 首先应当理解,尽管下文提供一个或多个实施例的说明性实现方式,但所公开的系统和/或方法可以使用任意数量的技术来实施,无论这些技术是当前已知的还是现有的。本发明决不应限于下文所说明的说明性实现方式、附图和技术,包括本文所说明并描述的示例性设计和实现方式,而是可以在所附权利要求书的范围以及其等效物的完整范围内修改。

[0079] 本文使用以下缩略语:自适应环路滤波器(Adaptive Loop Filter,ALF)、编码树块(Coding Tree Block,CTB)、编码树单元(Coding Tree Unit,CTU)、编码单元(Coding Unit,CU)、编码视频序列(Coded Video Sequence,CVS)、联合视频专家组(Joint Video Experts Team,JVET)、运动约束分块集(Motion-Constrained Tile Set,MCTS)、最大传输单元(Maximum Transfer Unit,MTU)、网络抽象层(Network Abstraction Layer,NAL)、图像顺序编号(Picture Order Count,POC)、原始字节序列载荷(Raw Byte Sequence Payload,RBSP)、样本自适应偏移(Sample Adaptive Offset,SAO)、序列参数集(Sequence Parameter Set,SPS)、时间运动矢量预测(Temporal Motion Vector Prediction,TMVP)、通用视频编码(Versatile Video Coding,VVC)和工作草案(Working Draft,WD)。

[0080] 许多视频压缩技术可以用于在产生最少数据丢失的情况下减小视频文件的大小。例如,视频压缩技术可以包括执行空间(例如,帧内)预测和/或时间(例如,帧间)预测来减少或去除视频序列中的数据冗余。对于基于块的视频译码,视频条带(slice)(例如,视频图像或视频图像中的一部分)可以分割成视频块,这些视频块还可以称为树块(treeblock)、编码树块(coding tree block,CTB)、编码树单元(coding tree unit,CTU)、编码单元(coding unit,CU)和/或编码节点。图像中的经帧内译码(I)条带中的视频块是相对于同一图像中的相邻块中的参考样本使用空间预测进行译码的,而图像中的经帧间译码的单向预测(P)或双向预测(B)条带中的视频块可以是相对于同一图像中的相邻块中的参考样本使用空间预测进行译码的,也可以是相对于其它参考图像中的参考样本使用时间预测进行译码的。图像(picture/image)可以称为帧(frame),参考图像可以称为参考帧。空间预测或时间预测会产生表示图像块的预测块。残差数据表示原始图像块与预测块之间的像素差。因此,经帧间译码块是根据运动矢量和残差数据进行编码的,其中,运动矢量指向形成预测块的参考样本组成的块,残差数据表示经译码块与预测块之间的差值;而经帧内译码块是根据帧内译码模式和残差数据进行编码的。为了进一步压缩,残差数据可以从像素域变换到变换域。这些产生可以量化的残差变换系数(简称为量化变换系数)。量化变换系数最初可以排列为二维阵列。量化变换系数可以扫描,目的是产生变换系数的一维矢量。熵译码可以用于实现进一步压缩。下文更详细地论述了这些视频压缩技术。

[0081] 为了确保经编码视频能够正确解码,视频根据对应的视频编码标准进行编码和解码。视频编码标准包括国际电信联盟(International Telecommunication Union,ITU)标准化部门(ITU Standardization Sector,ITU-T)H.261、国际标准化组织/国际电工委员会(International Organization for Standardization/International

Electrotechnical Commission,ISO/IEC)运动图像专家组(Motion Picture Experts Group,MPEG)-1第2部分、ITU-T H.262或ISO/IEC MPEG-2第2部分、ITU-T H.263、ISO/IEC MPEG-4第2部分、高级视频编码(Advanced Video Coding,AVC)(还称为ITU-T H.264或ISO/IEC MPEG-4第10部分)以及高效视频编码(High Efficiency Video Coding,HEVC)(还称为ITU-T H.265或MPEG-H第2部分)。AVC包括可分级的视频编码(Scalable Video Coding,SVC)、多视图视频编码(Multiview Video Coding,MVC)和多视图视频编码加深度(Multiview Video Coding plus Depth,MVC+D)、三维(three dimension,3D)AVC(3D-AVC)等扩展版本。HEVC包括可分级的HEVC(Scalable HEVC,SHVC)、多视图HEVC(Multiview HEVC,MV-HEVC)、3D HEVC(3D-HEVC)等扩展版本。ITU-T和ISO/IEC的联合视频专家组(joint video experts team,JVET)已经着手开发一种称为通用视频编码(Versatile Video Coding,VVC)的视频编码标准。VVC包括在工作草案(Working Draft,WD)中,该工作草案包括JVET-M1001-v6,提供了算法描述、VVC WD中的编码端描述以及参考软件。

[0082] 为了对视频图像进行译码,该图像首先进行分割,分割部分(partition)然后译码到码流中。存在各种图像分割方案。例如,图像可以分割成常规条带(regular slice)、非独立条带、分块(tile),和/或根据波前并行处理(Wavefront Parallel Processing,WPP)进行分割。为了简单起见,HEVC对编码器做了限制,使得在将条带分割成CTB组,以用于视频译码时,只能使用常规条带、非独立条带、分块、WPP及其组合。这种分割可以用于支持最大传输单元(maximum transfer unit,MTU)大小匹配、并行处理和减少端到端时延。MTU表示可以在单个数据包中发送的最大数据量。如果数据包载荷超过MTU,则该载荷通过称为分割(fragmentation)的过程划分为两个数据包。

[0083] 常规条带还简称为条带,是图像中的分割部分,可以独立于同一图像中的其它常规条带重建,但是由于存在环路滤波操作,所以还是存在相互依赖关系。每个常规条带封装在其自身的网络抽象层(Network Abstraction Layer,NAL)单元中,以用于传输。此外,为了支持独立重建,帧内预测(in-picture prediction/intra sample prediction)(运动信息预测、译码模式预测)和跨条带边界的熵译码依赖关系可以被禁用。这种独立重建支持并行处理。例如,基于常规条带的并行处理减少了处理器间或核间通信。但是,由于每个常规条带是独立的,所以每个条带都与单独的条带头相关。由于每个条带的条带头都存在比特成本,以及缺乏跨条带边界预测,所以使用常规条带可能会带来大量的译码开销。此外,常规条带可以用于支持MTU大小匹配需求。具体地,由于常规条带封装在单独的NAL单元中并且可以独立译码,所以每个常规条带需要小于MTU方案中的MTU,以避免将条带拆分为多个数据包。因此,为了实现并行处理和MTU大小匹配,图像中的条带布局可能互相矛盾。

[0084] 非独立条带与常规条带类似,但条带头更短,并且可以在不影响帧内预测的情况下实现图像树块边界的分割。因此,非独立条带可以将常规条带分割成多个NAL单元,这样通过先完成整个常规条带的编码,然后将该常规条带的一部分发送出去,降低端到端时延。

[0085] 分块(tile)是图像中由水平边界和垂直边界产生的分割部分,这些边界产生了分块列和分块行。分块可以按照光栅扫描顺序(从右到左,从上到下)进行译码。CTB的扫描顺序是在一个分块内部执行扫描的顺序。因此,第一分块中的CTB首先按照光栅扫描顺序进行译码,然后进行到下一分块中的CTB。与常规条带类似,分块影响了帧内预测依赖关系以及熵解码依赖关系。但是,分块可能不包括在各个NAL单元中,因此分块可能不会用于MTU大小

匹配。每个分块可以由一个处理器/核处理,对相邻分块进行的处理单元之间的帧内预测所采用的处理器间/核间通信可能只限于发送共享的条带头(当相邻分块位于同一条带中时)以及执行重建样本和元数据的环路滤波相关共享。当条带中包括多个分块时,除条带中的第一个入口点(entry point)偏移以外,每个分块的入口点字节偏移可以指示(signal)在条带头中。对于每个条带和分块,需要满足以下至少一个条件:(1)条带中的所有经译码树块都属于同一分块;(2)分块中的所有经译码树块都属于同一条带。

[0086] 在WPP中,图像分割成多个单行CTB。熵解码和预测机制可以使用其它行中的CTB中的数据。并行处理可能通过CTB行的并行解码实现。例如,当前行可以与前一行并行解码。但是,当前行的解码比前几行的解码过程延迟两个CTB。这种延迟确保了在当前CTB进行解码之前,与当前行中的当前CTB的上方CTB和右上方CTB相关的数据可用。这种方法用图形表示时,看起来像波面(wavefront)。这种交错式开始解码可以使用与图像中包括的CTB行一样多的处理器/核来实现并行处理。由于支持在图像中的各行相邻树块之间进行帧内预测,所以可能需要大量处理器间/核间通信来实现帧内预测。WPP分割不考虑NAL单元大小。因此,WPP不支持MTU大小匹配。但是,常规条带可以与WPP结合使用,产生一定的译码开销,从而根据需要实现MTU大小匹配。

[0087] 分块还可以包括运动约束分块集。运动约束分块集(motion constrained tile set,MCTS)是一种分块集,设计使得相关运动矢量被限制为指向MCTS内的整样本位置以及指向只需要MCTS内的整样本位置进行插值的分数样本位置。此外,从MCTS外部的块中推导出的用于时间运动矢量预测的运动矢量候选被禁止使用。这样一来,每个MCTS可以在不需要MCTS外部的分块的情况下独立解码。时间MCTS补充增强信息(supplemental enhancement information,SEI)消息可以用于表示MCTS存在于码流中并指示这些MCTS。MCTS SEI消息提供补充信息(表示为SEI消息语义的一部分),该补充信息可以用于MCTS子码流提取,以生成用于MCTS集的一致性码流。该信息包括多个提取信息集,每个提取信息集定义多个MCTS集并且包括要在MCTS子码流提取过程中使用的替换视频参数集(video parameter set,VPS)、序列参数集(sequence parameter set,SPS)和图像参数集(picture parameter set,PPS)的原始字节序列载荷(raw bytes sequence payload,RBSP)字节。当根据MCTS子码流提取过程提取子码流时,由于一个或所有条带地址相关的语法元素(包括first_slice_segment_in_pic_flag和slice_segment_address)可以在提取到的子码流中采用不同的值,所以参数集(VPS、SPS和PPS)可以被重写或替换,条带头可以被更新。

[0088] 图像还可以分割成一个或多个子图像。将图像分割成子图像可以实现,从译码观点以不同的方式处理图像中的不同部分。例如,子图像可以在不提取其它子图像的情况进行提取和显示。又如,不同的子图像可以按不同的分辨率显示,相对于彼此重新定位(例如,在电话会议应用中),或者即使子图像共同包括同一图像中的数据,也可以译码为单独的图像。

[0089] 子图像的示例性实现方式如下所述。图像可以分割成一个或多个子图像。子图像是以地址等于0的条带/分块组开始的矩形或方形条带集/分块集。每个子图像可以参考不同的PPS,因此每个子图像可以采用不同的分割机制。在解码过程中,子图像可以当作图像处理。对当前子图像进行解码所使用的当前参考图像可以通过从解码图像缓冲区中的参考图像中提取与当前子图像并置的区域来生成。提取到的区域可以是经解码子图像,因此帧

间预测可以发生在图像中的相同大小和相同位置的子图像之间。分块组可以是子图像的分块光栅扫描下的一系列分块。以下内容可以推导用来确定子图像在图像中的位置。每个子图像可以包括在图像中的按照CTU光栅扫描顺序排列的下一个未占用位置上,该位置足够大以适合图像边界处的子图像。

[0090] 各种视频译码系统采用的子图像方案存在降低译码效率和/或功能的各种问题。本发明包括各种技术方案,以解决这些问题。在第一示例性问题中,帧间预测可以根据几种帧间预测模式中的一种来执行。某些帧间预测模式在编码器和解码器侧都生成运动矢量预测值的候选列表。这样,编码器可以通过指示候选列表中的索引来指示运动矢量,而不是指示运动矢量本身。此外,一些系统对子图像进行编码以进行独立提取。独立提取使得当前子图像可以在其它子图像中的信息没有进行解码的情况下进行解码并显示。这种情况可能会在使用指向子图像外部的运动矢量时产生错误,因为运动矢量指向的数据可能没有解码,因此不能使用。

[0091] 因此,在第一示例中,本文公开了一种表示子图像可以当作图像处理的标志。该标志被设置为支持子图像的单独提取。当设置有该标志时,从并置块(collocated block)中获得的运动矢量预测值只包括指向子图像内部的运动矢量,而指向子图像外部的任何运动矢量预测值排除在外。这样确保指向子图像外部的运动矢量不被选择,避免了相关错误。并置块是与当前图像不同的图像中的块。来自当前图像中的块(非并置块)的运动矢量预测值可以指向子图像外部,因为插值滤波器等其它过程可以避免这些运动矢量预测值导致的错误。因此,本示例为视频编码器/解码器(编解码器)提供了额外功能,即防止在执行子图像提取时出现错误。

[0092] 在第二示例中,本文公开了一种表示子图像可以当作图像处理的标志。在当前子图像当作图像处理时,当前子图像可以在不参考其它子图像的情况下提取。具体地,本示例使用在应用插值滤波器时使用的限幅函数。这种限幅函数确保插值滤波器不依赖于相邻子图像中的数据,以保持子图像分开,从而支持单独提取。因此,当设置有该标志且运动矢量指向当前子图像外部时,应用限幅函数。然后,将插值滤波器应用于限幅函数的结果。因此,本示例为视频编解码器提供了额外功能,即防止在执行子图像提取时出现错误。这样一来,第一示例和第二示例解决了第一示例性问题。

[0093] 在第二示例性问题中,视频译码系统将图像分割成子图像、条带(slice)、分块(tile)和/或编码树单元(coding tree unit),然后它们分割成块。接着对这些块进行编码,用于发送给解码器。对这些块进行解码可能会产生包括各种噪声的经解码图像。为了解决这些问题,视频译码系统可以跨块边界应用各种滤波器。这些滤波器可以去除块效应(blocking)、量化噪声和其它不好的译码伪影。如上所述,一些系统对子图像进行编码以进行独立提取。独立提取使得当前子图像可以在其它子图像中的信息没有进行编码的情况下进行编码并显示。在这些系统中,子图像可以分割成块,以用于编码。因此,沿子图像边缘的块边界可以与子图像边界对齐。在一些情况下,块边界也可以与分块边界对齐。滤波器可以跨这些块边界应用,因此也可以跨子图像边界和/或分块边界应用。这样可能会在当前子图像被独立提取时产生错误,因为当相邻子图像中的数据不可用时,滤波过程可能会以意想不到的方式执行。

[0094] 在第三示例中,本文公开了一种控制子图像级滤波的标志。当该标志针对子图像

设置时,滤波器可以跨子图像边界应用。当未设置有该标志时,滤波器不会跨子图像边界应用。这样,滤波器可以对于编码以用于单独提取的子图像禁用,而对于编码以用于成组显示的子图像启用。因此,本示例为视频编解码器提供了额外功能,即防止在执行子图像提取时出现滤波器相关错误。

[0095] 在第四示例中,本文公开了一种可以设置为控制分块级滤波的标志。当该标志针对分块设置时,滤波器可以跨分块边界应用。当未设置有该标志时,滤波器不会跨分块边界应用。这样,所述滤波器可以禁用或启用,以便在分块边界处使用(例如,同时继续滤波所述分块的内部部分)。因此,本实施例为视频编解码器提供了额外功能,即支持跨分块边界的选择性滤波。这样一来,第三示例和第四示例解决了第二示例性问题。

[0096] 在第三示例性问题中,视频译码系统可以将图像分割成子图像。这样,不同的子图像可以在对视频进行译码时以不同的方式处理。例如,子图像可以单独提取和显示,根据应用级更改独立调整大小,等等。在一些情况下,子图像可以通过将图像分割成分块并将分块分配给子图像来创建。一些视频译码系统根据子图像中包括的分块来描述子图像边界。但是,分块方案可能不适用于一些图像中。因此,这些边界描述可以限制采用分块的图像使用子图像。

[0097] 在第五示例中,本文公开了一种用于根据CTB和/或CTU指示子图像边界的机制。具体地,子图像的宽度和高度可以按CTB为单位指示。此外,子图像中的左上方CTU的位置可以指示为与图像中的左上方CTU的偏移,测量单位是CTB。CTU和CTB的大小可以被设置为预定值。因此,根据CTB和CTU指示子图像尺寸和位置提供足够信息给解码器来定位子图像,以用于显示。这样,子图像还可以用于未采用分块的情况下。此外,这种指示机制既避免了复杂性,又可以使用相对较少的比特进行译码。因此,本示例为视频编解码器提供了额外功能,即将子图像与分块独立使用。此外,本示例提高了译码效率,因此减少了编码器和解码器侧处理器资源、内存资源和/或网络资源的使用。这样一来,第五示例解决了第三示例性问题。

[0098] 在第四示例性问题中,图像可以分割成多个条带,以用于编码。在一些视频译码系统中,这些条带根据它们相对于图像的位置寻址。还有一些视频译码系统采用子图像的概念。如上所述,从译码角度来看,子图像可以与其它子图像以不同的方式处理。例如,子图像可以独立于其它子图像提取和显示。在这种情况下,由于省略了大量预期条带地址,根据图像位置生成的条带地址可能会停止正常起作用。一些视频译码系统解决这个问题的方式是根据请求动态重写条带头以更改条带地址来支持子图像提取。这种过程可能需要很多资源,因为每次用户请求查看子图像时,这个过程都可能会发生。

[0099] 在第六示例中,本文公开了相对于包括条带的子图像寻址的条带。例如,条带头可以包括子图像标识符(identifier,ID)和子图像中包括的每个条带的地址。此外,序列参数集(sequence parameter set,SPS)可以包括子图像ID可以参考的子图像的维度。因此,当请求单独提取子图像时,不需要重写条带头。条带头和SPS包括足够信息来支持定位子图像中的条带,以用于显示。因此,本示例提高了译码效率和/或避免了条带头重写冗余,从而减少了编码器和/或解码器侧处理器资源、内存资源和/或网络资源的使用。这样一来,第六示例解决了第四示例性问题。

[0100] 图1为对视频信号进行译码的示例性操作方法100的流程图。具体地,视频信号是在编码器侧编码的。编码过程通过采用各种机制来压缩视频信号,以减小视频文件。文件较

小,压缩视频文件可用于发送给用户,同时降低相关带宽开销。然后,解码器对压缩视频文件进行解码,以重建原始视频信号,向终端用户显示。解码过程通常是编码过程的逆过程,以使解码器重建的视频信号可以与编码器侧的视频信号保持一致。

[0101] 在步骤101中,将视频信号输入到编码器中。例如,视频信号可以是存储在存储器中的未压缩视频文件。又如,视频文件可以由摄像机等视频捕获设备捕获,并且进行编码以支持视频的直播流传输。视频文件可以包括音频分量和视频分量。视频分量包括一系列图像帧。这些图像帧按顺序观看时,给人以运动的视觉效果。这些帧包括以光表示的像素,在本文中称为亮度分量(或亮度样本),还包括以颜色表示的像素,称为色度分量(或色度样本)。在一些示例中,这些帧还可以包括深度值,以支持三维观看。

[0102] 在步骤103中,将视频分割成块。分割包括将每一帧中的像素细分成方块和/或矩形块进行压缩。例如,在高效视频编码(High Efficiency Video Coding,HEVC)(还称为H.265和MPEG-H第2部分)中,首先将帧分成编码树单元(coding tree unit,CTU),它们是预定义大小(例如,64个像素×64个像素)的块。这些CTU包括亮度样本和色度样本。编码树可以用于将CTU分成块,然后,重复细分这些块,直到获得支持进一步编码的配置。例如,帧的亮度分量可以细分到各个块包括相对均匀的亮度值。此外,对帧的色度分量可以细分到各个块包括相对均匀的色值。因此,分割机制因视频帧的内容而异。

[0103] 在步骤105中,采用各种压缩机制对在步骤103中分割得到的图像块进行压缩。例如,可以采用帧间预测和/或帧内预测。帧间预测是为了利用场景中的对象往往出现在连续帧中这一事实而设计的。这样一来,描述参考帧中的对象的块不需要在相邻帧中重复描述。具体地,一个对象(例如一张桌子)可以在多个帧中保持在固定位置上。因此,该桌子被描述一次,而且相邻帧可以参考回参考帧。模式匹配机制可以用于在多个帧上匹配对象。此外,由于对象移动或相机移动等原因,移动对象可以跨多个帧表示。在一个具体示例中,视频可以在多个帧上显示跨屏幕移动的汽车。运动矢量可以用于描述这种移动。运动矢量是一个二维矢量,提供对象在一个帧中的坐标到该对象在参考帧中的坐标的偏移。因此,帧间预测可以将当前帧中的图像块编码为运动矢量集,表示当前帧中的图像块与参考帧中的对应块的偏移。

[0104] 帧内预测对公共帧中的块进行编码。帧内预测利用亮度分量和色度分量往往聚集在一个帧中这一事实。例如,一棵树某个部分的一片绿色往往与类似的几片绿色相邻。帧内预测采用多种方向性预测模式(例如,HEVC中有33种)、平面模式和直流(direct current,DC)模式。这些方向性模式表示当前块的样本与对应方向上的相邻块的样本相似/相同。平面模式表示一行/列(例如,平面)上的一系列块可以根据该行的边缘上的相邻块进行插值。平面模式实际上通过采用变化值的相对恒定的斜率表示光/颜色跨行/列的平稳过渡。DC模式用于边界平滑,并表示块与所有相邻块的样本的平均值相似/相同,这些相邻块与方向性预测模式的角度方向相关。因此,帧内预测块可以将图像块表示为各种关系预测模式值而不是表示为实际值。此外,帧间预测块可以将图像块表示为运动矢量值而不是表示为实际值。在任一种情况下,预测块在一些情况下可能都无法准确表示图像块。所有差值都存储在残差块中。可以对残差块应用变换以进一步压缩文件。

[0105] 在步骤107中,可以应用各种滤波技术。在HEVC中,滤波器是根据环内滤波方案应用的。上文描述的基于块的预测可能会在解码器侧产生块状图像。此外,基于块的预测方案

可以对块进行编码,然后重建经编码块,以便后续用作参考块。环内滤波方案迭代地将噪声抑制滤波器、去块效应滤波器、自适应环路滤波器和样本自适应偏移(sample adaptive offset,SAO)滤波器应用于块/帧。这些滤波器减少了块伪影,这样可以准确重建经编码文件。此外,这些滤波器减少了重建参考块中的伪影,这样伪影不太可能在根据重建参考块编码的后续块中产生其它伪影。

[0106] 一旦视频信号完成分割、压缩和滤波,则在步骤109中,将所得数据编码到码流中。码流包括上文描述的数据以及支持在解码器侧进行适当的视频信号重建所需要的任何指示数据。例如,这些数据可以包括分割数据、预测数据、残差块和提供译码指令给解码器的各种标志。码流可以存储在存储器中,以便根据请求向解码器发送。码流还可以广播和/或组播到多个解码器。创建码流是一个迭代过程。因此,步骤101、步骤103、步骤105、步骤107和步骤109可以在多个帧和块中连续和/或同时执行。图1所示的顺序是为了清楚和便于论述的目的,并非旨在将视频译码过程限于特定顺序。

[0107] 在步骤111中,解码器接收码流并开始解码过程。具体地,解码器采用熵解码方案将码流转换为对应的语法数据和视频数据。在步骤111中,解码器采用码流中的语法数据来确定帧的分割部分。分割应该与步骤103中的块分割的结果匹配。下面描述在步骤111中采用的熵编码/解码。编码器在压缩过程中做出许多选择,例如,根据一个或多个输入图像中的值的空间放置从若干个可能选择中选择块分割方案。指示确切的选择可能会采用大量的位元(bin)。本文所使用的“位元”是一个作为变量的二进制值(例如,可能因内容而异的比特值)。熵编码使得编码器丢弃任何明显不适合特定情况的选项,从而留下一组可用选项。然后,为每个可用选项分配一个码字。码字的长度取决于可允许选项的数量(例如,一个位元对应两个选项,两个位元对应三个或四个选项,以此类推)。然后,编码器对选定的选项对应的码字进行编码。这种方案减小了码字,这是因为码字与预期的一样大,从而唯一地指示从可允许选项的小子集中进行选择,而不是唯一地指示从所有可能选项的可能大集合中进行选择。然后,解码器通过以与编码器类似的方式确定可允许选项集对该选择进行解码。通过确定可允许选项集,解码器可以读取码字并确定编码器做出的选择。

[0108] 在步骤113中,解码器执行块解码。具体地,解码器采用逆变换,以生成残差块。然后,解码器采用残差块和对应的预测块,以根据分割来重建图像块。预测块可以包括编码器在步骤105中生成的帧内预测块和帧间预测块。接着,根据在步骤111中确定的分割数据将重建图像块放置在重建视频信号的帧中。还可以通过上文描述的熵编码在码流中指示用于步骤113的语法。

[0109] 在步骤115中,以类似于编码器侧的步骤107的方式对重建视频信号的帧执行滤波。例如,噪声抑制滤波器、去块效应滤波器、自适应环路滤波器和SAO滤波器可以应用于帧,以去除块效应。一旦对帧进行了滤波,则在步骤117中,可以将视频信号输出到显示器,以供终端用户观看。

[0110] 图2为用于视频译码的示例性编码和解码(编解码)系统200的示意图。具体地,编解码系统200提供功能来支持实现操作方法100。编解码系统200广义地用于描述编码器和解码器侧都使用的组件。编解码系统200接收视频信号并对视频信号进行分割,如结合操作方法100中的步骤101和步骤103所述,得到分割后的视频信号201。然后,编解码系统200在充当编码器时将分割后的视频信号201压缩到经编码码流中,如结合方法100中的步骤105、

步骤107和步骤109所述。编解码系统200在充当解码器时从码流中生成输出视频信号,如结合操作方法100中的步骤111、步骤113、步骤115和步骤117所述。编解码系统200包括通用译码器控制组件211、变换缩放和量化组件213、帧内估计组件215、帧内预测组件217、运动补偿组件219、运动估计组件221、缩放和逆变换组件229、滤波器控制分析组件227、环内滤波器组件225、解码图像缓冲区组件223以及标头格式以及上下文自适应二进制算术编码(context adaptive binary arithmetic coding,CABAC)组件231。这些组件如图所示耦合。在图2中,黑线表示待编码/待解码数据的运动,而虚线表示控制其它组件操作的控制数据的运动。编解码系统200中的组件都可以存在于编码器中。解码器可以包括编解码系统200中的组件的子集。例如,解码器可以包括帧内预测组件217、运动补偿组件219、缩放和逆变换组件229、环内滤波器组件225以及解码图像缓冲区组件223。下面对这些组件进行描述。

[0111] 分割后的视频信号201是一个捕获到的已经通过编码树分割成像素块的视频序列。编码树采用各种划分模式将像素块细分成较小的像素块。这些块然后可以进一步细分成较小的块。这些块可以称为编码树上的节点。较大的父节点划分成较小的子节点。节点进行细分的次数称为节点/编码树的深度。在一些情况下,划分得到的块包括在编码单元(coding unit,CU)中。例如,CU可以是CTU的子部分,包括亮度块、一个或多个红色差色度(Cr)块和一个或多个蓝色差色度(Cb)块以及CU的对应语法指令。划分模式可以包括二叉树(binary tree,BT)、三叉树(triple tree,TT)和四叉树(quad tree,QT),用于根据所采用的划分模式将节点分别分割成不同形状的两个、三个或四个子节点。将分割后的视频信号201转发给通用译码器控制组件211、变换缩放和量化组件213、帧内估计组件215、滤波器控制分析组件227以及运动估计组件221进行压缩。

[0112] 通用译码器控制组件211用于根据应用约束条件做出与将视频序列中的图像编码到码流中相关的决策。例如,通用译码器控制组件211管理码率/码流大小相对于重建质量的优化。可以根据存储空间/带宽可用性和图像分辨率请求做出这些决策。通用译码器控制组件211还根据传输速度管理缓冲区利用率,以缓解缓存欠载和超载问题。为了解决这些问题,通用译码器控制组件211管理由其它组件进行的分割、预测和滤波。例如,通用译码器控制组件211可以动态提高压缩复杂度以提高分辨率和增加带宽利用率,或者降低压缩复杂度以降低分辨率和带宽利用率。因此,通用译码器控制组件211控制编解码系统200中的其它组件来平衡视频信号重建质量与码率问题。通用译码器控制组件211生成控制数据,这些控制数据用于控制其它组件的操作。还将控制数据转发给标头格式和CABAC组件231,以编码到码流中,从而指示解码器进行解码使用的参数。

[0113] 还将分割后的视频信号201发送给运动估计组件221和运动补偿组件219进行帧间预测。可以将分割后的视频信号201的帧或条带分成多个视频块。运动估计组件221和运动补偿组件219相对于一个或多个参考帧中的一个或多个块对所接收到的视频块进行帧间预测译码,以提供时间预测。编解码系统200可以执行多个译码回合,以便为每个视频数据块选择合适的译码模式,等等。

[0114] 运动估计组件221和运动补偿组件219可以高度集成,但出于概念目的,单独说明。由运动估计组件221执行的运动估计是生成运动矢量的过程,其中,这些运动矢量用于估计视频块的运动。例如,运动矢量可以表示经译码对象相对于预测块的位移。预测块是在像素

差方面发现与待译码块高度匹配的块。预测块还可以称为参考块。这种像素差可以通过绝对差异和(sum of absolute difference,SAD)、平方差异和(sum of square difference,SSD)或其它差异度量来确定。HEVC采用若干个经译码对象,包括CTU、编码树块(coding tree block,CTB)和CU。例如,可以将CTU分成多个CTB,然后可以将CTB分成CB,以包括在CU中。可以将CU编码为包括预测数据的预测单元(prediction unit,PU)和/或包括CU的变换残差数据的变换单元(transform unit,TU)。运动估计组件221使用率失真分析作为率失真优化过程的一部分来生成运动矢量、PU和TU。例如,运动估计组件221可以确定当前块/帧的多个参考块、多个运动矢量等,并且可以选择具有最佳率失真特性的参考块、运动矢量等。最佳率失真特性平衡了视频重建的质量(例如,压缩造成的数据丢失量)和译码效率(例如,最终编码的大小)。

[0115] 在一些示例中,编解码系统200可以计算存储在解码图像缓冲区组件223中的参考图像的子整数像素位置的值。例如,视频编解码系统200可以对参考图像的四分之一像素位置、八分之一像素位置或其它分数像素位置的值进行插值。因此,运动估计组件221可以相对于整像素位置和分数像素位置执行运动搜索,并输出具有分数像素精度的运动矢量。运动估计组件221通过将PU的位置与参考图像的预测块的位置进行比较,计算经帧间译码条带中的视频块的PU的运动矢量。运动估计组件221将计算到的运动矢量作为运动数据输出到标头格式和CABAC组件231进行编码,并作为运动数据输出到运动补偿组件219。

[0116] 运动补偿组件219执行的运动补偿可以涉及根据运动估计组件221所确定的运动矢量获取或生成预测块。同样,在一些示例中,运动估计组件221和运动补偿组件219可以在功能上集成。在接收到当前视频块的PU的运动矢量时,运动补偿组件219可以定位运动矢量指向的预测块。然后,从正在译码的当前视频块的像素值中减去预测块的像素值,得到像素差,如此形成残差视频块。一般而言,运动估计组件221相对于亮度分量执行运动估计,而运动补偿组件219将根据亮度分量计算到的运动矢量用于色度分量和亮度分量。将预测块和残差块转发给变换缩放和量化组件213。

[0117] 还将分割后的视频信号201发送给帧内估计组件215和帧内预测组件217。与运动估计组件221和运动补偿组件219一样,帧内估计组件215和帧内预测组件217可以高度集成,但出于概念目的,单独说明。帧内估计组件215和帧内预测组件217相对于当前帧中的各块对当前块进行帧内预测,以替代如上所述的由运动估计组件221和运动补偿组件219在各帧之间执行的帧间预测。具体地,帧内估计组件215确定帧内预测模式对当前块进行编码。在一些示例中,帧内估计组件215从多个测试的帧内预测模式中选择合适的帧内预测模式来对当前块进行编码。然后,将选定的帧内预测模式转发给标头格式和CABAC组件231进行编码。

[0118] 例如,帧内估计组件215对各种测试的帧内预测模式进行率失真分析来计算率失真值,并在测试的模式中选择具有最佳率失真特性的帧内预测模式。率失真分析通常确定经编码块与经编码以产生经编码块的原始未编码块之间的失真(或误差)量,以及确定用于产生经编码块的码率(例如,比特数)。帧内估计组件215根据各种经编码块的失真和速率计算比率,以确定表现出块的最佳率失真值的帧内预测模式。另外,帧内估计组件215可以用于根据率失真优化(rate-distortion optimization,RDO),使用深度建模模式(depth modeling mode,DMM)对深度图像的深度块进行译码。

[0119] 帧内预测组件217在编码器上实现时可以根据由帧内估计组件215确定的选定帧内预测模式从预测块中生成残差块,或者在解码器上实现时,可以从码流中读取残差块。残差块包括预测块与原始块之间的差值,表示为矩阵。然后,将残差块转发给变换缩放和量化组件213。帧内估计组件215和帧内预测组件217可以对亮度分量和色度分量进行操作。

[0120] 变换缩放和量化组件213用于进一步压缩残差块。变换缩放和量化组件213将离散余弦变换(discrete cosine transform,DCT)、离散正弦变换(discrete sine transform,DST)等变换或者进行概念上类似的变换应用于残差块,从而产生包括残差变换系数值的视频块。还可以使用小波变换、整数变换、子频带变换或其它类型的变换。变换可以将残差信息从像素值域转换到变换域,例如频域。变换缩放和量化组件213还用于根据频率等对变换残差信息进行缩放。这种缩放涉及将缩放因子应用于残差信息,以便在不同的粒度下量化不同的频率信息,这可能会影响重建视频的最终视觉质量。变换缩放和量化组件213还用于量化变换系数以进一步降低码率。量化过程可以减小与部分或全部系数相关的位深度。量化程度可以通过调整量化参数来修改。在一些示例中,变换缩放和量化组件213随后可以对包括量化变换系数的矩阵进行扫描。将量化变换系数转发给标头格式和CABAC组件231,以编码到码流中。

[0121] 缩放和逆变换组件229进行与变换缩放和量化组件213相反的操作以支持运动估计。缩放和逆变换组件229应用逆缩放、逆变换和/或反量化以重建像素域中的残差块,例如,后续用作参考块。该参考块可以成为另一当前块的预测块。运动估计组件221和/或运动补偿组件219可以通过将残差块添加回对应的预测块来计算参考块,以用于后续块/帧的运动估计。将滤波器应用于重建参考块,以减少在缩放、量化和变换期间产生的伪影。当预测后续块时,这些伪影可能会使预测不准确(并产生额外的伪影)。

[0122] 滤波器控制分析组件227和环内滤波器组件225将滤波器应用于残差块和/或重建图像块。例如,可以将来自缩放和逆变换组件229的变换残差块与来自帧内预测组件217和/或运动补偿组件219的对应预测块组合以重建原始图像块。然后,可以将滤波器应用于重建图像块。在一些示例中,滤波器可以转而应用于残差块。与图2中的其它组件一样,滤波器控制分析组件227和环内滤波器组件225高度集成,可以一起实现,但是出于概念目的,单独描述。将应用于重建参考块的滤波器应用于特定空间区域,这些滤波器包括多个参数以调整使用这些滤波器的方式。滤波器控制分析组件227对重建参考块进行分析,以确定需要使用这些滤波器的位置并设置对应的参数。将这些数据作为滤波器控制数据转发给标头格式和CABAC组件231进行编码。环内滤波器组件225根据滤波器控制数据应用这些滤波器。这些滤波器可以包括去块效应滤波器、噪声抑制滤波器、SAO滤波器和自适应环路滤波器。这些滤波器可以根据示例应用于空域/像素域(例如,针对重建像素块)或频域中。

[0123] 当作为编码器操作时,将经滤波波的重建图像块、残差块和/或预测块存储在解码图像缓冲区组件223中,以供后续用于运动估计,如上所述。当作为解码器操作时,解码图像缓冲区组件223存储经重建和经滤波的块并将其作为输出视频信号的一部分转发给显示器。解码图像缓冲区组件223可以是任何能够存储预测块、残差块和/或重建图像块的存储设备。

[0124] 标头格式和CABAC组件231从编解码系统200的各种组件接收数据,并将这些数据编码到经译码码流中,用于发送给解码器。具体地,标头格式和CABAC组件231生成各种标头

以对控制数据(例如通用控制数据和滤波器控制数据)进行编码。此外,将预测数据(包括帧内预测数据和运动数据)以及以量化变换系数数据为形式的残差数据都编码到码流中。最终的码流包括解码器重建原始分割后的视频信号201所需要的所有信息。这些信息还可以包括帧内预测模式索引表(也称为码字映射表)、各种块的编码上下文的定义、最可能帧内预测模式的指示、分割信息的指示等。这些数据可以采用熵编码来编码。例如,这些信息可以采用上下文自适应可变长度编码(context adaptive variable length coding,CAVLC)、CABAC、基于语法的上下文自适应二进制算术编码(syntax-based context-adaptive binary arithmetic coding,SBAC)、概率区间分割熵(probability interval partitioning entropy,PIPE)编码或其它熵编码技术来编码。在熵编码之后,可以将经编码码流发送给另一设备(例如,视频解码器)或存档以供后续发送或检索。

[0125] 图3为示例性视频编码器300的框图。视频编码器300可以用于实现编解码系统200的编码功能和/或执行操作方法100中的步骤101、步骤103、步骤105、步骤107和/或步骤109。编码器300对输入视频信号进行分割,得到基本上类似于分割后的视频信号201的分割后的视频信号301。然后,通过编码器300中的组件压缩分割后的视频信号301并编码到码流中。

[0126] 具体地,将分割后的视频信号301转发给帧内预测组件317进行帧内预测。帧内预测组件317可以基本上类似于帧内估计组件215和帧内预测组件217。还将分割后的视频信号301转发给运动补偿组件321,根据解码图像缓冲区组件323中的参考块进行帧间预测。运动补偿组件321可以基本上类似于运动估计组件221和运动补偿组件219。将来自帧内预测组件317和运动补偿组件321的预测块和残差块转发给变换和量化组件313进行残差块的变换和量化。变换和量化组件313可以基本上类似于变换缩放和量化组件213。将变换量化残差块和对应的预测块(连同相关的控制数据)转发给熵编码组件331,以译码到码流中。熵编码组件331可以基本上类似于标头格式和CABAC组件231。

[0127] 还将变换量化残差块和/或对应的预测块从变换和量化组件313转发给逆变换和量化组件329,以重建为参考块供运动补偿组件321使用。逆变换和量化组件329可以基本上类似于缩放和逆变换组件229。根据示例,还将环内滤波器组件325中的环内滤波器应用于残差块和/或重建参考块。环内滤波器组件325可以基本上类似于滤波器控制分析组件227和环内滤波器组件225。环内滤波器组件325可以包括多个滤波器,如结合环内滤波器组件225所述。然后,将经滤波的块存储在解码图像缓冲区组件323中,以作为参考块供运动补偿组件321使用。解码图像缓冲区组件323可以基本上类似于解码图像缓冲区组件223。

[0128] 图4为示例性视频解码器400的框图。视频解码器400可以用于实现编解码系统200的解码功能和/或执行操作方法100中的步骤111、步骤113、步骤115和/或步骤117。解码器400从编码器300等接收码流,并根据码流生成重建输出视频信号,以向终端用户显示。

[0129] 码流由熵解码组件433接收。熵解码组件433用于执行熵解码方案,例如CAVLC、CABAC、SBAC、PIPE解码或其它熵解码技术。例如,熵解码组件433可以使用标头信息来提供上下文以解析在码流中编码为码字的附加数据。经解码信息包括对视频信号进行解码所需的任何信息,例如,通用控制数据、滤波器控制数据、分割信息、运动数据、预测数据和残差块中的量化变换系数。将量化变换系数转发给逆变换和量化组件429,以重建成残差块。逆变换和量化组件429可以类似于逆变换和量化组件329。

[0130] 将重建残差块和/或预测块转发给帧内预测组件417,以根据帧内预测操作重建为图像块。帧内预测组件417可以类似于帧内估计组件215和帧内预测组件217。具体地,帧内预测组件417使用预测模式来定位帧中的参考块,并将残差块添加到上述结果中以重建帧内预测图像块。将重建帧内预测图像块和/或残差块以及对应的帧间预测数据通过环内滤波器组件425转发给解码图像缓冲区组件423。解码图像缓冲区组件423和环内滤波器组件425可以基本上分别类似于解码图像缓冲区组件223和环内滤波器组件225。环内滤波器组件425对重建图像块、残差块和/或预测块进行滤波。这些信息存储在解码图像缓冲区组件423中。将来自解码图像缓冲区组件423的重建图像块转发给运动补偿组件421,以进行帧间预测。运动补偿组件421可以基本上类似于运动估计组件221和/或运动补偿组件219。具体地,运动补偿组件421使用参考块的运动矢量来生成预测块,并将残差块应用于上述结果中以重建图像块。还可以通过环内滤波器组件425将所得到的重建块转发给解码图像缓冲区组件423。解码图像缓冲区组件423继续存储其它重建图像块。这些重建图像块可以通过分割信息重建为帧。这些帧还可以放置在一个序列中。该序列作为重建输出视频信号输出到显示器。

[0131] 图5A为分割成子图像510的示例性图像500的示意图。例如,图像500可以由编解码系统200和/或编码器300分割以用于编码以及由编解码系统200和/或解码器400分割以用于解码。又如,图像500可以由编码器在方法100的步骤103中分割,以供解码器在步骤111中使用。

[0132] 图像500是描述视频序列在指定时间位置上的完整视觉部分的图像。图像(picture/image)500还可以称为帧。图像500可以由图像顺序编号(picture order count,POC)表示。POC是表示图像500在视频序列中的输出顺序/显示顺序的索引。图像500可以分割成子图像510。子图像510是图像500中的一个或多个条带/分块组的矩形区域或方形区域。子图像510是可选的,因此一些视频序列包括子图像510,而其它视频序列不包括。虽然示出了四个子图像510,但图像500可以分割成任意数量的子图像510。子图像510的分割可以在整个编码视频序列(coded video sequence,CVS)上保持一致。

[0133] 子图像510可以用于以不同的方式处理图像500中的不同区域。例如,指定的子图像510可以独立提取并发送给解码器。在一个具体示例中,用户使用虚拟现实(virtual reality,VR)头盔可以看到图像500的子集,这可以给用户带来身临图像500描述的空间中的感觉。在这种情况下,对很可能显示给用户的子图像510进行流式传输可能会提高译码效率。又如,不同的子图像510可以在某些应用中以不同的方式处理。在一个具体示例中,电话会议应用可以在比当前未说话的用户更突出的位置上以更高的分辨率显示活动的说话者。将不同的用户放置在不同的子图像510中支持实时配置所显示图像以支持这种功能。

[0134] 每个子图像510可以由唯一的子图像ID标识,这可以对于整个CVS保持一致。例如,位于图像500左上方的子图像510可以具有为0的子图像ID。在这种情况下,序列中的任何图像500的左上方子图像510可以由为0的子图像ID参考。此外,每个子图像510可以包括定义好的配置,这可以对于整个CVS保持一致。例如,子图像510可以包括高度、宽度和/或偏移。高度和宽度描述子图像510的大小,偏移描述子图像510的位置。例如,一行中所有子图像510的宽度之和等于图像500的宽度。此外,一列中所有子图像510的高度之和等于图像500的高度。另外,偏移表示子图像510的左上角相对于图像500的左上角的位置。子图像510的

高度、宽度和偏移提供足够信息来定位图像500中的对应子图像510。由于子图像510的分割可以在整个CVS上保持一致,因此与子图像相关的参数可以包括在序列参数集(sequence parameter set,SPS)中。

[0135] 图5B为分割成条带(slice)515的示例性子图像510的示意图。如图所示,图像500中的子图像510可以包括一个或多个条带515。条带515包括整数个完整的分块或一个分块内的整数个连续完整的CTU行,这些行只包括在单个网络抽象层(network abstraction layer,NAL)单元中。虽然示出了四个条带515,但子图像510可以包括任意数量的条带515。条带515包括指定POC的图像500特有的视觉数据。因此,与条带515相关的参数可以包括在图像参数集(picture parameter set,PPS)和/或条带头中。

[0136] 图5C为分割成分块(tile)517的示例性条带515的示意图。如图所示,图像500中的条带515可以包括一个或多个分块517。分块517可以通过将图像500分割成矩形的各行各列来创建。因此,分块517是图像中特定分块列和特定分块行内的CTU组成的矩形区域或方形区域。分块是可选的,因此一些视频序列包括分块517,而其它视频序列不包括。虽然示出了四个分块517,但条带515可以包括任意数量的分块517。分块517可以包括指定POC的图像500中的条带515特有的视觉数据。在一些情况下,条带515也可以包括在分块517中。因此,与块517相关的参数可以包括在PPS和/或条带头中。

[0137] 图5D为分割成CTU 519的示例性条带515的示意图。如图所示,图像500中的条带515(或条带515中的分块517)可以包括一个或多个CTU 519。CTU 519是图像500中的一个区域,其通过编码树进行细分,产生译码块进行编码/解码。CTU 519可以包括黑白图像500中的亮度样本或彩色图像500中的亮度样本和色度样本的组合。可以通过编码树分割的一组亮度样本或色度样本称为编码树块(coding tree block,CTB)518。因此,CTU 519包括具有三个样本阵列的图像500中的亮度样本组成的CTB 518和色度样本组成的两个对应CTB 518,或者黑白图像中或使用三个单独彩色平面和语法结构进行译码的图像中的样本组成的CTB 518,其中,这些语法结构用于对这些样本进行译码。

[0138] 如上所述,图像500可以分割成子图像510、条带515、分块517、CTU 519和/或CTB 518,然后它们分割成块。接着对这些块进行编码,用于发送给解码器。对这些块进行解码可能会产生包括各种噪声的经解码图像。为了解决这些问题,视频译码系统可以跨块边界应用各种滤波器。这些滤波器可以去除块效应(blocking)、量化噪声和其它不好的译码伪影。如上所述,子图像510可以在执行独立提取时采用。在这种情况下,当前子图像510可以在其它子图像510中的信息没有进行解码的情况下进行解码并显示。因此,沿子图像510边缘的块边界可以与子图像边界对齐。在一些情况下,块边界也可以与分块边界对齐。滤波器可以跨这些块边界应用,因此也可以跨子图像边界和/或分块边界应用。这样可能会在当前子图像510被独立提取时产生错误,因为当相邻子图像510中的数据不可用时,滤波过程可能会以意想不到的方式执行。

[0139] 为了解决这些问题,一种标志可以用于控制子图像510级滤波。例如,该标志可以表示为loop_filter_across_subpic_enabled_flag。当该标志针对子图像510设置时,滤波器可以跨对应的子图像边界应用。当未设置有该标志时,滤波器不会跨对应的子图像边界应用。这样,滤波器可以对于编码以用于单独提取的子图像510禁用,而对于编码以用于成组显示的子图像510启用。另一种标志可以被设置为控制分块517级滤波。该标志可以表示

为loop_filter_across_tiles_enabled_flag。当该标志针对分块517设置时,滤波器可以跨分块边界应用。当未设置有该标志时,滤波器不会跨分块边界应用。这样,所述滤波器可以禁用或启用,以便在分块边界处使用(例如,同时继续滤波所述分块的内部部分)。如本文所述,滤波器在应用于边界两侧的样本时跨子图像510或分块517的边界应用。

[0140] 同样如上所述,分块(tiling)是可选的。但是,一些视频译码系统根据子图像510中包括的分块517描述子图像边界。在这些系统中,根据分块517的子图像边界描述限制采用分块517的图像500使用子图像510。为了扩大子图像510的适用性,子图像510可以根据边界、CTB 518和/或CTU 519描述。具体地,子图像510的宽度和高度可以按CTB 518为单位指示。此外,子图像510中的左上方CTU 519的位置可以指示为与图像500中的左上方CTU 519的偏移,测量单位是CTB 518。CTU 519和CTB 518的大小可以被设置为预定值。因此,根据CTB 518和CTU 519指示子图像尺寸和位置提供足够信息给解码器来定位子图像510,以用于显示。这样,子图像510还可以用于未采用分块517的情况下。

[0141] 另外,一些视频译码系统根据条带515相对于图像500的位置寻址这些条带。当子图像510进行译码以用于独立提取和显示时,这会产生一个问题。在这种情况下,条带515和与省略的子图像510相关的对应地址也被省略掉。省略条带515的地址可能会妨碍解码器正确定位条带515。一些视频译码系统通过动态重写与条带515相关的条带头中的地址来解决这个问题。由于用户可以请求任何子图像,因此用户每次请求视频时都会发生这种重写,这需要大量资源。为了解决这个问题,当采用子图像510时,条带515相对于包括条带515的子图像510进行寻址。例如,条带515可以由包括条带515的子图像510特有的索引或其它值标识。条带地址可以译码到与条带515相关的条带头中。包括条带515的子图像510的子图像ID也可以编码到条带头中。此外,子图像510的尺寸/配置可以与子图像ID一起译码到SPS中。因此,解码器可以根据子图像ID从SPS中获取子图像510的配置,并在不参考完整图像500的情况下,在子图像510中定位条带515。因此,当提取到子图像510时,可以省略条带头重写,这大大减少了编码器、解码器和/或对应条带器侧的资源使用。

[0142] 一旦图像500分割成CTB 518和/或CTU 519,CTB 518和/或CTU 519就可以进一步划分为译码块。译码块然后可以根据帧内预测和/或帧间预测进行译码。本发明还包括与帧间预测机制相关的改进点。帧间预测可以在几种不同的模式下执行,这些模式可以根据单向帧间预测和/或双向帧间预测运行。

[0143] 图6为单向帧间预测600的一个示例的示意图。例如,在块压缩步骤105、块解码步骤113、运动估计组件221、运动补偿组件219、运动补偿组件321和/或运动补偿组件421处,执行单向帧间预测600来确定运动矢量(motion vector,MV)。例如,单向帧间预测600可以用于确定在分割图像(例如图像500)时产生的经编码块和/或解码块的运动矢量。

[0144] 单向帧间预测600采用包括参考块631的参考帧630来预测当前帧610中的当前块611。参考帧630可以如图所示在时间上位于当前帧610之后(例如,作为下一个参考帧),但在一些示例中,也可以在时间上位于当前帧610之前(例如,作为前一个参考帧)。当前帧610是在特定时间进行编码/解码的示例性帧/图像。当前帧610包括当前块611中的一个对象,该对象与参考帧630中的参考块631中的一个对象匹配。参考帧630是对当前帧610进行编码所使用的参考帧,参考块631是参考帧630中的一个块,这个块包括的对象也包括在当前帧610中的当前块611中。

[0145] 当前块611是译码过程中在指定时间点进行编码/解码的任何译码单元。当前块611可以是整个分割块,也可以是采用仿射帧间预测时的子块。当前帧610与参考帧630间隔一定时间距离(temporal distance,TD)633。TD 633表示视频序列中的当前帧610与参考帧630之间的时间量,测量单位可以是帧。当前块611的预测信息可以通过表示各帧之间的方向和时间距离的参考索引对参考帧630和/或参考块631参考。在TD 633表示的时间段内,当前块611中的对象从当前帧610中的一个位置移动到参考帧630中的另一个位置(例如,参考块631的位置)。例如,该对象可以沿着运动路径(trajectory)613移动,运动路径613表示对象随时间移动的方向。运动矢量635描述该对象在TD 633内沿着运动路径613移动的方向和幅度。因此,经编码运动矢量635、参考块631和包括当前块611与参考块631之差的残差提供足够信息来重建当前块611并且在当前帧610中定位当前块611。

[0146] 图7为双向帧间预测700的一个示例的示意图。例如,在块压缩步骤105、块解码步骤113、运动估计组件221、运动补偿组件219、运动补偿组件321和/或运动补偿组件421处,执行双向帧间预测700以确定MV。例如,双向帧间预测700可以用于确定在分割图像(例如图像500)时产生的经编码块和/或解码块的运动矢量。

[0147] 双向帧间预测700类似于单向帧间预测600,但采用一对参考帧来预测当前帧710中的当前块711。因此,当前帧710和当前块711分别基本上类似于当前帧610和当前块611。当前帧710在时间上位于前一个参考帧720与下一个参考帧730之间,其中,前一个参考帧720在视频序列中出现在当前帧710之前,下一个参考帧730在视频序列中出现在当前帧710之后。前一个参考帧720和下一个参考帧730在其它方面基本上类似于参考帧630。

[0148] 当前块711与前一个参考帧720中的前一个参考块721和下一个参考帧730中的下一个参考块731匹配。这种匹配表示,在播放视频序列的过程中,一个对象沿着运动路径713从前一个参考块721中的位置经过当前块711移动到下一个参考块731中的位置。当前帧710与前一个参考帧720间隔一定的前一个时间距离(TD0)723,且与下一个参考帧730间隔一定的下一个时间距离(TD1)733。TD0 723表示视频序列中的前一个参考帧720与当前帧710之间的时间量,以帧为单位。TD1 733表示视频序列中的当前帧710与下一个参考帧730之间的时间量,以帧为单位。因此,该对象在由TD0 723表示的时间段内沿着运动路径713从前一个参考块721移动到当前块711。该对象还在由TD1 733表示的时间段内沿着运动路径713从当前块711移动到下一个参考块731。当前块711的预测信息可以通过一对表示各帧之间的方向和时间距离的参考索引来参考前一个参考帧720和/或前一个参考块721以及下一个参考帧730和/或下一个参考块731。

[0149] 前一个运动矢量(MV0)725描述该对象在TD0 723(例如,在前一个参考帧720与当前帧710之间)内沿着运动路径713移动的方向和幅度。下一个运动矢量(MV1)735描述该对象在TD1 733(例如,在当前帧710与下一个参考帧730之间)内沿着运动路径713移动的方向和幅度。因此,在双向帧间预测700中,当前块711可以通过前一个参考块721和/或下一个参考块731、MV0 725和MV1 735进行译码和重建。

[0150] 在融合(merge)模式和高级运动矢量预测(advanced motion vector prediction,AMVP)模式中,候选列表都是通过按照候选列表确定模式定义好的顺序将候选运动矢量添加到候选列表中生成的。这些候选运动矢量可以包括根据单向帧间预测600、双向帧间预测700或其组合的运动矢量。具体地,运动矢量是在相邻块进行编码时针对这些块

生成的。这些运动矢量被添加到当前块的候选列表中,而且当前块的运动矢量是从该候选列表中选择的。然后,运动矢量可以指示为选定运动矢量在候选列表中的索引。解码器可以使用与编码器相同的过程构建候选列表,并可以根据指示的索引从候选列表中确定选定的运动矢量。因此,候选运动矢量包括根据单向帧间预测600和/或双向帧间预测700生成的运动矢量,具体取决于在编码这些相邻块时使用哪种方法。

[0151] 图8为根据来自相邻经译码块802的候选运动矢量对当前块801进行译码的示例800的示意图。编码器300和/或解码器400操作方法100和/或采用编解码器系统200的功能,可以采用相邻经译码块802来生成候选列表。这种候选列表可以用于根据单向帧间预测600和/或双向帧间预测700的帧间预测中。然后,候选列表可以用于对当前块801进行编码/解码,当前块801可以通过分割图像(例如图像500)生成。

[0152] 当前块801是在指定时间内根据示例在编码器侧编码或在解码器侧解码的块。经译码块802是在指定时间已经编码的块。因此,经译码块802很可能用于生成候选列表。当前块801和经译码块802可以包括在相同帧中和/或可以包括在时间相邻帧中。当经译码块802与当前块801包括在相同帧中时,经译码块802包括与当前块801的边界紧邻(例如,邻接)的边界。当经译码块802包括在时间相邻帧中时,经译码块802在时间相邻帧中的位置与当前块801在当前帧中的位置相同。候选列表可以是通过将来自经译码块802的运动矢量添加为候选运动矢量生成的。然后,当前块801可以通过从候选列表中选择候选运动矢量并指示选定候选运动矢量的索引进行译码。

[0153] 图9为用于确定运动矢量候选列表的示例性模式900的示意图。具体地,编码器300和/或解码器400操作方法100和/或采用编解码器系统200的功能,可以采用候选列表确定模式900来生成候选列表911,从而对从图像500中分割到的当前块801进行编码。得到的候选列表911可以是融合候选列表或AMVP候选列表,都可以用于根据单向帧间预测600和/或双向帧间预测700的帧间预测中。

[0154] 当对当前块901进行编码时,候选列表确定模式900在当前块901所在的相同图像/帧中的位置905(表示为A0、A1、B0、B1和/或B2)中搜索有效的候选运动矢量。候选列表确定模式900还可以在并置块909中搜索有效的候选运动矢量。并置块909是与当前块901位于相同位置的块,但包括在时间相邻图像/帧中。然后,候选运动矢量可以按照预定检查顺序放置在候选列表911中。因此,候选列表911是一列按程序生成的带索引的候选运动矢量。

[0155] 候选列表911可以用于选择运动矢量以执行当前块901的帧间预测。例如,编码器可以获取由候选列表911中的候选运动矢量指向的参考块中的样本。然后,编码器可以选择指向与当前块901最匹配的参考块的候选运动矢量。然后,选定的候选运动矢量的索引可以进行编码以表示当前块901。在一些情况下,一个或多个候选运动矢量指向包括部分参考样本915的参考块。在这种情况下,插值滤波器913可以用于重建完整的参考样本915以支持运动矢量选择。插值滤波器913是一种能够对信号进行上采样的滤波器。具体地,插值滤波器913是一种能够接受部分/低质量信号作为输入并确定完整/高质量信号的近似值的滤波器。因此,插值滤波器913可以在某些情况下用于获取参考样本915的完整集合,以用于选择当前块901的参考块,因此用于选择运动矢量以对当前块901进行编码。

[0156] 上述根据帧间预测使用候选列表对块进行译码的机制在采用子图像(例如子图像510)时可能会导致某些错误。具体地,这些问题可能会出现在当前块901包括在当前子图像

中,但运动矢量指向至少部分位于相邻子图像中的参考块时。在这种情况下,当前子图像可以在没有相邻子图像的情况提取进行呈现。当这种情况发生时,相邻子图像中的参考块的各部分可能不发送给解码器,因此参考块可能不可用于对当前块901进行解码。当这种情况发生时,解码器无法访问足够的数据来对当前块901进行解码。

[0157] 本发明提供了解决这种问题的机制。在一个示例中,一种标志用于表示当前子图像可以当作图像处理。该标志可以设置为支持子图像的单独提取。具体地,当设置有该标志时,当前子图像可以在不参考其它子图像中的数据的情况下进行编码。在这种情况下,当前子图像当作图像处理,因为当前子图像与其它子图像分开译码且可用于显示为单独的图像。因此,该标志可以表示为subpic_treated_as_pic_flag[i],其中,i表示当前子图像的索引。当设置有该标志时,从并置块909中获得的运动矢量候选(也称为运动矢量预测值)只包括指向当前子图像内部的运动矢量。指向当前子图像外部的任何运动矢量预测值排除在候选列表911之外。这样确保指向当前子图像外部的运动矢量不被选择,避免了相关错误。本示例具体应用于来自并置块909的运动矢量。来自同一图像/帧中的搜索位置905的运动矢量可以通过如下所述的不同机制修改。

[0158] 另一个示例可以用于在当前子图像当作图像处理时(例如,当设置有subpic_treated_as_pic_flag[i]时)寻址搜索位置905。在当前子图像当作图像处理时,当前子图像可以在不参考其它子图像的情况下提取。示例机制涉及插值滤波器913。插值滤波器913可以应用于一个位置上的样本,以对另一个位置上的相关样本进行插值(例如,预测)。在本示例中,只要插值滤波器913可以只根据当前子图像中的参考样本915对当前子图像外部的参考样本915进行插值,则来自经译码块的位于搜索位置905上的运动矢量可以指向参考样本915。因此,本示例采用在将插值滤波器913应用于来自同一图像中的搜索位置905的运动矢量候选时使用的限幅函数。在确定由运动矢量候选指向的参考样本915时,该限幅函数对相邻子图像中的数据进行限幅并因此去除输入到插值滤波器913的数据。这种方法在编码过程中保持子图像分开,以支持当子图像当作图像处理时的单独提取和解码。限幅函数可以应用于亮度样本双线性插值过程、亮度样本8抽头插值滤波过程和/或色度样本插值过程。

[0159] 图10为示例性环内滤波器1000的框图。环内滤波器1000可以用于实现环内滤波器组件225、325和/或425。此外,在执行方法100时,环内滤波器1000可以应用于编码器和解码器侧。另外,环内滤波器1000可以用于滤波从图像500中分割到的当前块801,当前块801可以根据单向帧间预测600和/或双向帧间预测700使用根据模式900生成的候选列表进行译码。环内滤波器1000包括去块效应滤波器1043、SAO滤波器1045和自适应环路滤波器(adaptive loop filter,ALF)1047。环内滤波器1000中的滤波器依次应用于编码器侧(例如,在用作参考块之前)和解码器侧(在显示之前)的重建图像块。

[0160] 去块效应滤波器1043用于去除由于基于块的帧间和帧内预测产生的块状边缘。去块效应滤波器1043扫描图像部分(例如,图像条带),以确定在分割边界处出现的色度值和/或亮度值的不连续性。然后,去块效应滤波器1043将平滑函数应用于块边界,以去除这些不连续性。去块效应滤波器1043的强度可以因发生在与块边界相邻的区域内的空间活动(例如,亮度/色度分量的变化)而异。

[0161] SAO滤波器1045用于去除与编码过程导致的样本失真相关的伪影。编码器侧的SAO

滤波器1045根据相关的去块效应滤波边缘形状和/或方向将重建图像中的经去块效应滤波的样本分类为几个类别。然后,根据类别确定偏移,并将偏移添加到样本中。偏移接着编码到码流中,解码器侧的SAO滤波器1045使用该偏移。SAO滤波器1045去除带状伪影(值带而不是平滑过渡)和振铃伪影(清晰边缘附近的杂波信号)。

[0162] 编码器侧的ALF 1047用于将重建图像与原始图像进行比较。ALF 1047通过基于维纳(Wiener)的自适应滤波器等确定描述重建图像与原始图像之间差异的系数。这些系数编码到码流中,并由解码器侧的ALF 1047用来去除重建图像与原始图像之间的差异。

[0163] 经过环内滤波器1000滤波的图像数据输出到图像缓冲区1023,其基本上类似于解码图像缓冲区组件223、323和/或423。如上所述,去块效应滤波器1043、SAO滤波器1045和/或ALF 1047可以通过loop_filter_across_subpic_enabled_flag和/或loop_filter_across_tiles_enabled_flag等标志在子图像边界和/或分块边界处禁用。

[0164] 图11为包括译码工具参数以支持对图像中的子图像进行解码的示例性码流1100的示意图。例如,码流1100可以由编解码系统200和/或编码器300生成,以由编解码系统200和/或解码器400进行解码。又如,码流1100可以由编码器在方法100的步骤109中生成,以供解码器在步骤111中使用。此外,码流1100可以包括经编码图像500、对应的子图像510和/或相关的经译码块,例如当前块801和/或901,当前块801和/或901可以根据单向帧间预测600和/或双向帧间预测700使用根据模式900生成的候选列表进行译码。码流1100还可以包括用于配置环内滤波器1000的参数。

[0165] 码流1100包括序列参数集(sequence parameter set,SPS)1110、多个图像参数集(picture parameter set,PPS)1111、多个条带头1115以及图像数据1120。SPS1110包括码流1100中包括的视频序列中的所有图像共用的序列数据。这些数据可以包括图像大小、位深度、译码工具参数、码率限制条件等。PPS 1111包括应用于整个图像的参数。因此,视频序列中的每个图像可以参考PPS1111。需要说明的是,虽然每个图像都参考PPS1111,但在一些示例中,单个PPS1111可以包括多个图像的数据。例如,多个类似的图像可以根据类似的参数进行译码。在这种情况下,单个PPS1111可以包括这样类似图像的数据。PPS1111可以表示可用于对应图像中条带的译码工具、量化参数、偏移等。条带头1115包括图像中的每个条带特有的参数。因此,视频序列中的每个条带可以存在一个条带头1115。条带头1115可以包括条带类型信息、图像顺序编号(picture order count,POC)、参考图像列表、预测权重、分块入口点、去块效应滤波参数等。需要说明的是,在一些上下文中,条带头1115还可以称为分块组头。

[0166] 图像数据1120包括根据帧间预测和/或帧内预测进行编码的视频数据以及对应的变换量化残差数据。例如,视频序列包括译码为图像数据的多个图像。图像是视频序列中的单帧,因此在显示视频序列时通常显示为单个单元。但是,子图像可用于显示用来实现某些技术,例如虚拟现实、画中画等。图像都是参考PPS1111。如上所述,图像分成子图像、分块和/或条带。在一些系统中,条带称为包括分块的分块组。条带和/或包括分块的分块组参考条带头1115。这些条带进一步分成CTU和/或CTB。CTU/CTB根据编码树进一步分成译码块。译码块然后可以根据预测机制进行编码/解码。

[0167] 码流1100中的参数集包括可以用于实现本文所述示例的各种数据。为了支持第一示例性实现方式,码流1100中的SPS1110包括与指定子图像相关的子图像当作图像处理标

志(sub-pic treated as a pic flag)1131。在一些示例中,子图像当作图像处理标志1131表示为subpic_treated_as_pic_flag[i],其中,i表示与该标志相关的子图像的索引。例如,子图像当作图像处理标志1131可以被设置为1,表示编码视频序列中(图像数据1120中)的每个经译码图像的第i个子图像在除环内滤波操作以外的解码过程中当作图像处理。子图像当作图像处理标志1131可以用于当前图像中的当前子图像已经根据帧间预测进行译码的情况下。当子图像当作图像处理标志1131被设置为表示当前子图像作为图像处理时,当前块的候选运动矢量的候选列表可以通过将并置块中包括的且指向当前子图像外部的并置运动矢量排除在候选列表之外来确定。这样确保在当前子图像与其它子图像分开提取时,指向当前子图像外部的运动矢量不被选择,避免了相关错误。

[0168] 在一些示例中,当前块的运动矢量的候选列表是根据时间亮度运动矢量预测确定的。例如,时间亮度运动矢量预测可以用于以下情况中:当前块是亮度样本组成的亮度块,当前块的选定当前运动矢量是指向参考块中的参考亮度样本的时间亮度运动矢量,以及当前块是根据参考亮度样本进行译码的。在这种情况下,时间亮度运动矢量预测根据以下内容执行:

[0169] xColBr=xCb+cbWidth;

[0170] yColBr=yCb+cbHeight;

[0171] rightBoundaryPos=subpic_treated_as_pic_flag[SubPicIdx]?

[0172] SubPicRightBoundaryPos:pic_width_in_luma_samples-1;

[0173] botBoundaryPos=subpic_treated_as_pic_flag[SubPicIdx]?

[0174] SubPicBotBoundaryPos:pic_height_in_luma_samples-1,

[0175] 其中,xColBr和yColBR表示并置块的位置,xCb和yCb表示相对于当前图像的左上方样本的当前块的左上方样本,cbWidth表示当前块的宽度,cbHeight表示当前块的高度,SubPicRightBoundaryPos表示子图像的右边界的位置,SubPicBotBoundaryPos表示子图像的下边界的位置,pic_width_in_luma_samples表示以亮度样本为单位测量的当前图像的宽度,pic_height_in_luma_samples表示以亮度样本为单位测量的当前图像的高度,botBoundaryPos表示子图像的下边界的计算位置,rightBoundaryPos表示子图像的右边界的计算位置,SubPicIdx表示子图像的索引;当yCb>>CtbLog2SizeY不等于yColBr>>CtbLog2SizeY时,并置运动矢量排除在外,其中,CtbLog2SizeY表示编码树块的大小。

[0176] 子图像当作图像处理标志1131还可以用于第二示例性实现方式。如在第一示例中一样,子图像当作图像处理标志1131可以用于当前图像中的当前子图像已经根据帧间预测进行译码的情况下。在本示例中,运动矢量可以针对子图像中的当前块(例如,从候选列表中)确定。当设置有子图像当作图像处理标志1131时,限幅函数可以应用于参考块中的样本位置。样本位置是图像中的位置,可以包括亮度值和/或一对色度值组成的单个样本。然后,插值滤波器可以应用于运动矢量指向当前子图像外部的情况下。这种限幅函数确保插值滤波器不依赖于相邻子图像中的数据,以保持子图像分开,从而支持单独提取。

[0177] 限幅函数可以应用于亮度样本双线性插值过程中。亮度样本双线性插值过程可以接收包括具有整数个样本单位的亮度位置(xIntL,yIntL)的输入。亮度样本双线性插值过程输出预测亮度样本值(predSampleLXL)。限幅函数应用于样本位置,如下所述。当subpic_treated_as_pic_flag[SubPicIdx]等于1时,以下内容适用:

[0178] xInti＝Clip3(SubPicLeftBoundaryPos,SubPicRightBoundaryPos,xIntL+i),

[0179] yInti＝Clip3(SubPicTopBoundaryPos,SubPicBotBoundaryPos,yIntL+i),

[0180] 其中,subpic_treated_as_pic_flag表示设置为表示子图像当作子图像处理的标志,SubPicIdx表示子图像的索引,xInti和yInti表示索引i处的限幅后样本位置,SubPicRightBoundaryPos表示子图像的右边界的位置,SubPicLeftBoundaryPos表示子图像的左边界的位置,SubPicTopBoundaryPos表示子图像的上边界的位置,SubPicBotBoundaryPos表示子图像的下边界的位置,Clip3表示根据下式的限幅函数:

[0181] $$\mathrm{Clip3}(x,y,z)=\begin{cases}x & ; \ z<x\\ y & ; \ z>y,\\ z & ; \ \text{其它}\end{cases}$$

[0182] 其中,x、y和z是数字输入值。

[0183] 限幅函数还可以应用于亮度样本8抽头插值滤波过程中。亮度样本8抽头插值滤波过程接收包括具有整数个样本单位的亮度位置(xIntL,yIntL)的输入。亮度样本8抽头插值滤波过程输出预测亮度样本值(predSampleLXL)。限幅函数应用于样本位置,如下所述。当subpic_treated_as_pic_flag[SubPicIdx]等于1时,以下内容适用:

[0184] xInti＝Clip3(SubPicLeftBoundaryPos,SubPicRightBoundaryPos,xIntL+i-3),

[0185] yInti＝Clip3(SubPicTopBoundaryPos,SubPicBotBoundaryPos,yIntL+i-3),

[0186] 其中,subpic_treated_as_pic_flag表示设置为表示子图像当作图像处理的标志,SubPicIdx表示子图像的索引,xInti和yInti表示索引i处的限幅后样本位置,SubPicRightBoundaryPos表示子图像的右边界的位置,SubPicLeftBoundaryPos表示子图像的左边界的位置,SubPicTopBoundaryPos表示子图像的上边界的位置,SubPicBotBoundaryPos表示子图像的下边界的位置,Clip3如上所述。

[0187] 限幅函数还可以应用于色度样本插值过程中。色度样本插值过程接收包括具有整数个样本单位的色度位置(xIntC,yIntC)的输入。色度样本插值过程输出预测色度样本值(predSampleLXC)。限幅函数应用于样本位置,如下所述。当subpic_treated_as_pic_flag[SubPicIdx]等于1时,以下内容适用:

[0188] xInti＝Clip3(SubPicLeftBoundaryPos/SubWidthC,SubPicRightBoundaryPos/SubWidthC,xIntC+i),

[0189] yInti＝Clip3(SubPicTopBoundaryPos/SubHeightC,SubPicBotBoundaryPos/SubHeightC,yIntC+i),

[0190] 其中,subpic_treated_as_pic_flag表示设置为表示子图像当作图像处理的标志,SubPicIdx表示子图像的索引,xInti和yInti表示索引i处的限幅后样本位置,SubPicRightBoundaryPos表示子图像的右边界的位置,SubPicLeftBoundaryPos表示子图像的左边界的位置,SubPicTopBoundaryPos表示子图像的上边界的位置,SubPicBotBoundaryPos表示子图像的下边界的位置,SubWidthC和SubHeightC表示亮度样本与色度样本之间的水平采样率比值和垂直采样率比值,Clip3如上所述。

[0191] SPS1110中的跨子图像环路滤波器启用标志1132可以用于第三示例性实现方式。跨子图像环路滤波器启用标志1132可以被设置为控制是否跨指定子图像的边界采用滤波。例如,跨子图像环路滤波器启用标志1132可以表示为loop_filter_across_subpic_

enabled_flag。跨子图像环路滤波器启用标志1132在表示环内滤波操作可以跨子图像的边界执行时,可以被设置为1,或者在表示环内滤波操作不跨子图像的边界执行时,可以被设置为0。因此,根据跨子图像环路滤波器启用标志1132的值,滤波过程可以跨子图像的边界执行,也可以不跨子图像的边界执行。滤波操作可以包括应用去块效应滤波器1043、ALF 1047和/或SAO滤波器1045。这样,滤波器可以对于编码以用于单独提取的子图像禁用,而对于编码以用于成组显示的子图像启用。

[0192] PPS1111中的跨分块环路滤波器启用标志1134可以用于第四示例性实现方式。跨分块环路滤波器启用标志1134可以被设置为控制是否跨指定分块的边界采用滤波。例如,跨分块环路滤波器启用标志1134可以表示为loop_filter_across_tiles_enabled_flag。跨分块环路滤波器启用标志1134在表示环内滤波操作可以跨分块的边界执行时,可以被设置为1,或者在表示环内滤波操作不跨分块的边界执行时,可以被设置为0。因此,根据跨分块环路滤波器启用标志1134的值,滤波过程可以跨分块的边界执行,也可以不跨分块的边界执行。滤波操作可以包括应用去块效应滤波器1043、ALF 1047和/或SAO滤波器1045。

[0193] SPS1110中的子图像数据1133可以用于第五示例性实现方式。子图像数据1133可以包括图像数据1120中的每个子图像的宽度、高度和偏移。例如,每个子图像的宽度和高度可以在子图像数据1133中以CTB为单位描述。在一些示例中,子图像的宽度和高度分别作为subpic_width_minus1和subpic_height_minus1存储在子图像数据1133中。此外,每个子图像的偏移可以在子图像数据1133中以CTU为单位描述。例如,每个子图像的偏移可以表示为子图像中的左上方CTU的垂直位置和水平位置。具体地,子图像的偏移可以表示为图像中的左上方CTU与子图像中的左上方CTU之间的差值。在一些示例中,子图像中的左上方CTU的垂直位置和水平位置分别作为subpic_ctu_top_left_y和subpic_ctu_top_left_x存储在子图像数据1133中。本示例性实现方式以CTB/CTU为单位而不是以分块为单位描述子图像数据1133中的子图像。这样,子图像还可以用于分块未用于对应的图像/子图像中的情况下。

[0194] SPS1110中的子图像数据1133、条带头1115中的条带地址1136和条带头1115中的条带子图像ID 1135可以用于第六示例性实现方式。子图像数据1133可以如第五示例性实现方式中所述实现。条带地址1136可以包括与条带头1115相关的条带(例如,在图像数据1120中)的子图像级条带索引。例如,条带是根据条带在子图像中的位置而不是根据条带在图像中的位置进行索引的。条带地址1136可以存储在变量slice_address中。条带子图像ID 1135包括子图像的ID,该子图像包括与条带头1115相关的条带。具体地,条带子图像ID 1135可以参考子图像数据1133中的对应子图像的描述(例如,宽度、高度和偏移)。条带子图像ID 1135可以存储在变量slice_subpic_id中。因此,根据条带在如子图像数据1133中描述的条带子图像ID 1135表示的子图像中的位置,条带地址1136指示为一个索引。这样,条带在子图像中的位置还可以在子图像被单独提取并且从码流1100中省略其它子图像的情况下确定。这是因为这种寻址方案将每个子图像的地址与其它子图像的地址分开。因此,条带头1115在子图像提取时不需要重写,而在条带根据条带在图像的位置进行寻址的寻址方案中,必须要重写条带头。需要说明的是,这种方法可以用于条带是矩形/方形条带(与光栅扫描条带相对)的情况下。例如,PPS1111中的rect_slice_flag可以被设置为1,表示条带是矩形条带。

[0195] 在一些视频译码系统中使用的子图像的示例性实现方式如下所述。与CVS中可能

存在的子图像相关的信息可以指示在SPS中。这种指示可以包括以下信息。CVS的每个图像中存在的子图像的数量可以包括在SPS中。在SPS或CVS的上下文中,所有接入单元(access unit,AU)的并置子图像可以统称为子图像序列。用于进一步表示与每个子图像的属性相关的信息的环路也可以包括在SPS中。这些信息可以包括子图像标识、子图像的位置(例如,子图像中的左上角亮度样本与图像中的左上角亮度样本之间的偏移距离)以及子图像的大小。另外,SPS还可以用于指示每个子图像是否为运动约束子图像,其中,运动约束子图像是包括MCTS的子图像。每个子图像的档次(profile)、层次(tier)和级别(level)信息可以包括在码流中,除非这些信息是可导出的。这些信息可以用于通过从包括整个图像的原始码流中提取子图像而产生的提取码流的档次、层次和级别信息。每个子图像的档次和层次可以推导为与原始码流的档次和层次相同。每个子图像的级别可以显式指示。这种指示可以出现在上述环路中。序列级假设参考解码(hypothetical reference decoder,HRD)参数可以指示在每个子图像(或每个子图像序列)的SPS的视频可用信息(video usability information,VUI)部分中。

[0196] 当图像没有分割成两个或两个以上子图像时,子图像的属性(例如,位置、大小等),除子图像ID以外,可能不会指示在码流中。当提取CVS的图像中的子图像时,新码流中的每个接入单元可能不包括子图像,因为新码流中的每个AU中所产生的图像数据没有分割成多个子图像。因此,位置和大小等子图像属性可以从SPS中省略,因为这些信息可以从图像属性中推导出。但是,子图像标识仍然可以指示,因为这个ID可以由包括在提取子图像中的视频编码层(video coding layer,VCL)NAL单元/分块组参考。在提取子图像时,需要避免更改子图像ID,以减少资源使用。

[0197] 例如,子图像在图像中的位置(x偏移和y偏移)可以按亮度样本为单位指示,并且可以表示子图像中的左上角亮度样本与图像中的左上角亮度样本之间的距离。又如,子图像在图像中的位置可以按最小译码亮度块大小(MinCbSizeY)为单位指示,并且可以表示子图像中的左上角亮度样本与图像中的左上角亮度样本之间的距离。又如,子图像位置偏移的单位可以由参数集中的语法元素显式指示,该单位可以是CtbSizeY、MinCbSizeY、亮度样本或其它值。编解码器可能要求,当子图像的右边界与图像的右边界不重合时,子图像的宽度应该是亮度CTU大小(CtbSizeY)的整数倍。同理,编解码器还可能要求,当子图像的下边界与图像的下边界不重合时,子图像的高度应该是CtbSizeY的整数倍。编解码器还可能要求,当子图像的宽度不是亮度CTU大小的整数倍时,子图像位于图像最右边的位置上。同理,编解码器也可能要求,当子图像的高度不是亮度CTU大小的整数倍时,子图像位于图像最下方的位置上。当子图像的宽度以亮度CTU大小为单位指示,且子图像的宽度不是亮度CTU大小的整数倍时,以亮度样本为单位的实际宽度可以根据子图像的偏移位置、以亮度CTU大小为单位的子图像宽度和以亮度样本为单位的图像宽度推导出。同理,当子图像的高度以亮度CTU大小为单位指示,且子图像的高度不是亮度CTU大小的整数倍时,以亮度样本为单位的实际高度可以根据子图像的偏移位置、以亮度CTU大小为单位的子图像高度和以亮度样本为单位的图像高度推导出。

[0198] 对于任何子图像,子图像ID可以与子图像索引不同。子图像索引可以是指示在SPS中的子图像环路中的子图像的索引。可选地,子图像索引可以是相对于图像按照子图像光栅扫描顺序分配的索引。当每个子图像的子图像ID的值与其子图像索引相同时,子图像ID

可以指示或推导出。当每个子图像的子图像ID与其子图像索引不同时,子图像ID显式指示。用于指示子图像ID的比特数可以指示在包括子图像属性的同一参数集中(例如,在SPS中)。出于某些目的,子图像ID的一些值可以保留。这种值保留可以如下所述。当分块组/条带头包括子图像ID来表示哪个子图像包括分块组时,值0可以保留并且可以不用于子图像,以确保分块组/条带头起始处的前几个比特不全为0,从而避免生成防伪码。当图像中的子图像没有无重叠、无间隙地覆盖图像的整个区域时,一个值(例如,值1)可以针对不属于任何子图像的分块组保留。可选地,剩余区域的子图像ID可以显式指示。用于指示子图像ID的比特数可以约束如下。取值范围需要足够唯一地标识图像中的所有子图像,包括子图像ID的保留值。例如,用于子图像ID的最小比特数可以为Ceil(Log2(图像中子图像的数量+保留子图像ID的数量)的值。

[0199] 环路中的子图像的合并可能需要无间隙、无重叠地覆盖整个图像。当应用这种约束条件时,每个子图像存在一个标志,表示子图像是否为运动约束子图像,这表示可以提取该子图像。可选地,子图像的合并可能不覆盖整个图像。但是,图像中的子图像之间可能没有重叠。

[0200] 子图像ID可以紧跟在NAL单元头之后,以辅助子图像提取过程,所以提取器不需要知道其余的NAL单元比特。对于VCL NAL单元,子图像ID可以存在于分块组头的前几个比特中。对于非VCL NAL单元,以下内容可以适用。子图像ID可能不需要紧跟在SPS的NAL单元头之后。关于PPS,当同一图像中的所有分块组都约束为参考同一PPS时,子图像ID不需要紧跟在NAL单元头之后。另一方面,如果同一图像中的分块组可以参考不同PPS,则子图像ID可以存在于PPS的前几个比特中(例如,紧跟在PPS NAL单元头之后)。在这种情况下,一个图像中的任意两个分块组不可以共用同一PPS。可选地,当同一图像中的分块组可以参考不同PPS,且同一图像中的不同分块组还可以共用同一PPS时,PPS语法中不存在子图像ID。可选地,当同一图像中的分块组可以参考不同PPS,且同一图像中的不同分块组还可以共用同一PPS时,PPS语法中存在子图像ID列表。该列表表示PPS应用到的子图像。对于其它非VCL NAL单元,如果非VCL单元适用于图像级(例如,接入单元分隔符、序列结束、码流结束等)或以上级别,则子图像ID不需要紧跟在其NAL单元头之后。否则,子图像ID可以紧跟在NAL单元头之后。

[0201] 各个子图像内的分块分割可以指示在PPS中,但同一图像中的分块组可以参考不同PPS。在这种情况下,分块在每个子图像中进行分组,而不是跨图像进行分组。因此,这样情况下的分块分组概念包括将子图像分割成分块的方式。可选地,子图像参数集(Sub-Picture Parameter Set,SPPS)可以用于描述各个子图像内的分块分割。SPPS通过采用参考SPS ID的语法元素来参考SPS。SPPS可以包括子图像ID。出于子图像提取目的,参考子图像ID的语法元素是SPPS中的第一个语法元素。SPPS包括分块结构,表示列数、行数、均匀分块间距等。SPPS可以包括标志来表示环路滤波器是否跨相关子图像边界启用。可选地,每个子图像的子图像属性可以指示在SPPS中,而不是指示在SPS中。各个子图像内的分块分割可以指示在PPS中,但同一图像中的分块组可以参考不同PPS。SPPS一旦激活可以按照解码顺序持续一系列连续AU,但可以在不是CVS起始的AU处去激活/激活。在包括多个子图像的单层码流的解码过程中,多个SPPS可以在任何时刻处于活动状态,而且SPPS可以被AU的不同子图像共用。可选地,SPPS和PPS可以合并为一个参数集。为了实现这种情况,包括在同一子

图像中的所有分块组可能约束为参考SPPS和PPS合并得到的同一参数集。

[0202] 用于指示子图像ID的比特数可以指示在NAL单元头中。这种信息存在时有助于子图像提取过程解析NAL单元载荷起始的子图像ID值(例如,紧跟在NAL单元头之后的前几个比特)。对于这种指示,NAL单元头中的一些保留位可以用于避免增加NAL单元头的长度。用于这样指示的比特数可以覆盖sub-picture-ID-bit-len的值。例如,VVC NAL单元头中的7个保留比特中的4个比特可以用于此目的。

[0203] 当对子图像进行解码时,每个编码树块的位置,表示为垂直CTB位置(xCtb)和水平CTB位置(yCtb),被调整为图像中的实际亮度样本位置,而不是子图像中的亮度样本位置。这样,由于所有事物都解码为好像位于图像中而不是子图像中,所以可以避免从每个参考图像中提取并置子图像。为了调整编码树块的位置,变量SubpictureXOffset和SubpictureYOffset可以根据子图像位置(subpic_x_offset和subpic_y_offset)推导出。这两个变量的值可以分别与子图像中的每个编码树块的亮度样本位置x和y坐标的值相加。子图像提取过程可以定义如下。该过程的输入包括待提取的目标子图像。待提取的目标子图像可以按照子图像ID或子图像位置的形式输入。当该输入是子图像的位置时,相关子图像ID可以通过解析SPS中的子图像信息解析出。对于非VCL NAL单元,以下内容适用。SPS中与图像大小和级别相关的语法元素可以随着子图像的大小和级别信息而更新。以下非VCL NAL单元不会通过提取改变:PPS、接入单元分隔符(access unit delimiter,AUD)、序列结束(end of sequence,EOS)、码流结束(end of bitstream,EOB)以及任何其它适用于图像级或以上级别的非VCL NAL单元。子图像ID不等于目标子图像ID的剩余非VCL NAL单元可以被去除。子图像ID不等于目标子图像ID的VCL NAL单元也可以被去除。

[0204] 子图像嵌套SEI消息可以用于嵌套子图像集的AU级或子图像级SEI消息。子图像嵌套SEI消息携带的数据可以包括缓冲周期、图像定时和非HRD SEI消息。这个SEI消息的语法和语义可以如下所述。对于系统操作,例如,全向媒体格式(omnidirectional media format,OMAF)环境,覆盖视角的子图像序列集可以由OMAF播放器请求和解码。因此,序列级SEI消息可以携带统一包括矩形或方形图像区域的子图像序列集的信息。该信息可以由系统使用,该信息表示最小解码能力以及子图像序列集的码率。该信息包括只有子图像序列集的码流的级别、码流的码率,可选地包括为子图像序列集指定的子码流提取过程。

[0205] 上述实现方式存在几个问题。图像的宽度和高度和/或子图像的宽度/高度/偏移的指示效率不高。指示这些信息可以节省更多比特。当子图像大小和位置信息指示在SPS中时,PPS包括分块配置。此外,PPS可以由同一图像中的多个子图像共用。因此,num_tile_columns_minus1和num_tile_rows_minus1的取值范围可以更明确地说明。此外,表示子图像是否是运动约束子图像的标志的语义没有明确说明。每个子图像序列都必须指示级别。但是,当子图像序列不能独立解码时,指示子图像的级别没有用处。此外,在一些应用中,一些子图像序列可以与至少一个其它子图像序列一起解码和显示。因此,指示这些子图像序列中的单个子图像序列的级别可能没有用处。此外,确定每个子图像的定级别值可能会给编码器带来负担。

[0206] 随着独立可解码的子图像序列的引入,需要独立提取和解码图像中某些区域的方案可能无法根据分块组实现。因此,分块组ID的显式指示可能没有用处。此外,PPS语法元素pps_seq_parameter_set_id和loop_filter_across_tiles_enabled_flag中的每个语法元

素的值在经译码图像的分块组头参考的所有PPS中可以是相同的。这是因为活动SPS不可以在CVS中改变,并且loop_filter_across_tiles_enabled_flag的值对于根据分块用于并行处理的图像中的所有分块可以是相同的。是否可以在图像中混合矩形分块组和光栅扫描分块组需要明确说明。属于不同图像并在CVS中采用相同子图像ID的子图像是否可以使用不同的分块组模式也需要说明。时间亮度运动矢量预测的推导过程可能无法将子图像边界当作时间运动矢量预测(temporal motion vector prediction,TMVP)中的图像边界处理。此外,亮度样本双线性插值过程、亮度样本8抽头插值滤波过程和色度样本插值过程可以不将子图像边界当作运动补偿中的图像边界处理。另外,用于控制子图像边界处的去块效应滤波操作、SAO滤波操作和ALF滤波操作的机制也需要说明。

[0207] 随着独立可解码的子图像序列的引入,loop_filter_across_tile_groups_enabled_flag可能不太有用。这是因为出于并行处理目的禁用环内滤波操作也可以通过将loop_filter_across_tile_groups_enabled_flag设置为0来满足。此外,禁用环内滤波操作以实现图像中的某些区域的独立提取和解码也可以通过将loop_filter_across_sub_pic_enabled_flag设置为0来满足。因此,进一步说明根据loop_filter_across_tile_groups_enabled_flag禁用跨分块组边界的环内滤波操作的过程会不必要地给解码器带来负担并浪费比特。另外,如上所述的解码过程可能无法禁用跨分块边界的ALF滤波操作。

[0208] 因此,本发明包括支持基于子图像的视频译码的设计。子图像是图像中的矩形区域或方形区域,可以或可以不使用与图像相同的解码过程进行独立解码。这些技术的描述以通用视频编码(Versatile Video Coding,VVC)标准为基础。然而,这些技术还适用于其它视频编解码规范。

[0209] 在一些示例中,为图像宽度和高度语法元素以及语法元素子图像宽度/高度/offset_x/offset_y的列表指示大小单位。所有语法元素都以xxx_minus1的形式指示。例如,当大小单位为64个亮度样本时,宽度值为99表示图像宽度为6400个亮度样本。同样的示例适用于这些语法元素中的其它元素。又如,以下一个或多个大小单位适用。可以为语法元素图像宽度和高度指示大小单位,以xxx_minus1为形式。为语法元素子图像宽度/高度/offset_x/offset_y的列表指示的大小单位可以是xxx_minus1的形式。又如,以下一个或多个大小单位适用。语法元素图像宽度和子图像宽度/offset_x的列表的大小单位可以按xxx_minus1的形式指示。语法元素图像高度和子图像高度/offset_y的列表的大小单位可以按xxx_minus1的形式指示。又如,以下一个或多个大小单位适用。以xxx_minus1为形式的语法元素图像宽度和高度可以按最小编码单元为单位指示。以xxx_minus1为形式的语法元素子图像宽度/高度/offset_x/offset_y可以按CTU或CTB为单位指示。图像右边界处的每个子图像的子图像宽度可以推导出。图像下边界处的每个子图像的子图像高度可以推导出。子图像宽度/高度/offset_x/offset_y的其它值都可以指示在码流中。在其它示例中,对于子图像具有均匀大小的情况,可以添加用于指示子图像的宽度和高度以及它们在图像中的位置的模式。当子图像包括相同的子图像行和子图像列时,子图像具有均匀大小。在这种模式下,子图像行的数量、子图像列的数量、每个子图像列的宽度和每个子图像行的高度都可以指示。

[0210] 又如,子图像宽度和高度的指示可以不包括在PPS中。num_tile_columns_minus1和num_tile_rows_minus1的取值范围可以为0到整数值,例如1024,包括端值。又如,当参考

PPS的子图像包括一个以上分块时,通过存在标志(presence flag)调整的两个语法元素可以指示在PPS中。这些语法元素用于以CTB为单位指示子图像宽度和高度,并表示参考PPS的所有子图像的大小。

[0211] 又如,描述各个子图像的附加信息也可以指示。可以为每个子图像序列指示sub_pic_treated_as_pic_flag[i]等标志,以表示子图像序列中的子图像是否在解码过程中当作图像处理,以用于除环内滤波操作以外的目的。当sub_pic_treated_as_pic_flag[i]等于1时,只可以指示子图像序列符合的级别。子图像序列是具有相同子图像ID的子图像组成的CVS。当sub_pic_treated_as_pic_flag[i]等于1时,还可以指示子图像序列的级别。这可以由所有子图像序列的标志控制,也可以由每个子图像序列的一个标志控制。又如,可以在不改变VCL NAL单元的情况下实现子码流提取。这可以通过从PPS中去除显式分块组ID指示来实现。当rect_tile_group_flag等于1,表示矩形分块组时,规定tile_group_address的语义。tile_group_address可以包括子图像内的多个分块组中的分块组的分块组索引。

[0212] 又如,PPS语法元素pps_seq_parameter_set_id和loop_filter_across_tiles_enabled_flag中的每个语法元素的值在经译码图像的分块组头参考的所有PPS中可以是相同的。其它PPS语法元素对于经译码图像的分块组头参考的不同PPS可以不同。single_tile_in_pic_flag的值对于经译码图像的分块组头参考的不同PPS可以不同。这样,CVS中的一些图像可能只包括一个分块,而CVS中的其它一些图像可能包括一个以上分块。这样,图像中的一些子图像(例如,非常大的子图像)可以包括多个分块,而同一图像中的其它子图像(例如,非常小的子图像)只包括一个分块。

[0213] 又如,图像可以包括矩形扫描分块和光栅扫描分块组的混合。因此,图像中的一些子图像使用矩形分块组模式,而其它子图像使用光栅扫描分块组模式。这种灵活性有益于码流融合场景。可选地,约束条件可能要求图像中的所有子图像都应该使用相同的分块组模式。CVS中具有相同子图像ID的不同图像中的子图像可以不使用不同的分块组模式。CVS中具有相同子图像ID的不同图像中的子图像可以使用不同的分块组模式。

[0214] 又如,当子图像的sub_pic_treated_as_pic_flag[i]等于1时,用于子图像的时间运动矢量预测的并置运动矢量被限制为来自子图像的边界内。因此,处理子图像的时间运动矢量预测,就好像子图像边界是图像边界。此外,限幅操作被指定为亮度样本双线性插值过程、亮度样本8抽头插值滤波过程和色度样本插值过程的一部分,以在sub_pic_treated_as_pic_flag[i]等于1的子图像的运动补偿中,将子图像边界当做图像边界处理。

[0215] 又如,每个子图像都与loop_filter_across_sub_pic_enabled_flag等指示标志相关。该标志用于控制子图像边界处的环内滤波操作,并用于控制对应解码过程中的滤波操作。去块效应滤波过程可能不用于对与loop_filter_across_sub_pic_enabled_flag等于0的子图像的边界重合的子块边缘和变换块边缘译码。可选地,去块效应滤波过程不用于对与loop_filter_across_sub_pic_enabled_flag等于0的子图像的上边界或左边界重合的子块边缘和变换块边缘译码。可选地,去块效应滤波过程不用于对与sub_pic_treated_as_pic_flag[i]等于1或零的子图像的边界重合的子块边缘和变换块边缘译码。可选地,去块效应滤波过程不用于对与子图像的上边界或左边界重合的子块边缘和变换块边缘译码。当子图像的loop_filter_across_sub_pic_enabled_flag等于零时,限幅操作可以被指定来禁用跨子图像边界的SAO滤波操作。当子图像的loop_filter_across_sub_pic_enabled_

flag等于0时,限幅操作可以被指定来禁用跨子图像边界的ALF滤波操作。loop_filter_across_tile_groups_enabled_flag也可以从PPS中去除。因此,当loop_filter_across_tiles_enabled_flag等于0时,跨非子图像边界的分块组边界的环内滤波操作不会被禁用。环路滤波器操作可以包括去块效应滤波操作、SAO滤波操作和ALF滤波操作。又如,当分块的loop_filter_across_tiles_enabled_flag等于0时,限幅操作被指定来禁用跨分块边界的ALF滤波操作。

[0216] 上述一个或多个示例中可以实现如下。子图像可以被定义为图像中的一个或多个分块组或条带组成的矩形区域或方形区域。处理元素的以下划分可以形成空间或分量分割:将每个图像划分为分量,将每个分量划分为CTB,将每个图像划分为子图像,将每个子图像划分为子图像中的分块列,将每个子图像划分为子图像中的分块行,将子图像中的每个分块列划分为分块,将子图像中的每个分块行划分为分块,以及将每个子图像划分为分块组。

[0217] 子图像内的CTB光栅和分块扫描过程可以如下所述。列表ColWidth[i](i的取值范围为0到num_tile_columns_minus1,包括端值)表示以CTB为单位的第i个分块列的宽度,可以推导如下。

[0218]
```
if( uniform_tile_spacing_flag )
for( i = 0; i <= num_tile_columns_minus1; i++ )
    ColWidth[ i ] = ( ( i + 1 ) * SubPicWidthInCtbsY ) /
( num_tile_columns_minus1 + 1 ) –
                ( i * SubPicWidthInCtbsY ) / ( num_tile_columns_minus1 + 1 )
else {
ColWidth[ num_tile_columns_minus1 ] = SubPicWidthInCtbsY                    (6-1)
for( i = 0; i < num_tile_columns_minus1; i++ ) {
    ColWidth[ i ] = tile_column_width_minus1[ i ] + 1
    ColWidth[ num_tile_columns_minus1 ] –= ColWidth[ i ]
}
}
```

[0219] 列表RowHeight[j](j的取值范围为0到num_tile_rows_minus1,包括端值)表示以CTB为单位的第j个分块行的高度,推导如下。

[0220]
```
if( uniform_tile_spacing_flag )
for( j = 0; j <= num_tile_rows_minus1; j++ )
    RowHeight[ j ] = ( ( j + 1 ) * SubPicHeightInCtbsY ) / ( num_tile_rows_minus1 + 1 ) –
                ( j * SubPicHeightInCtbsY ) / ( num_tile_rows_minus1 + 1 )
else {
RowHeight[ num_tile_rows_minus1 ] = SubPicHeightInCtbsY                     (6-2)
for( j = 0; j < num_tile_rows_minus1; j++ ) {
    RowHeight[ j ] = tile_row_height_minus1[ j ] + 1
    RowHeight[ num_tile_rows_minus1 ] –= RowHeight[ j ]
}
}
```

[0221] 列表ColBd[i](i的取值范围为0到num_tile_columns_minus1+1,包括端值)表示以CTB为单位的第i个分块列边界的位置,推导如下。

[0222] for(ColBd[0]=0,i=0;i<=num_tile_columns_minus1;i++)

[0223] ColBd[i + 1] = ColBd[i] + ColWidth[i] (6-3)

[0224] 列表RowBd[j](j的取值范围为0到num_tile_rows_minus1+1,包括端值)表示以CTB为单位的第j个分块行边界的位置,推导如下。

[0225] for(RowBd[0]＝0,j＝0;j<＝num_tile_rows_minus1;j++)

[0226] RowBd[j + 1] ＝ RowBd[j] + RowHeight[j] (6-4)

[0227] 列表CtbAddrRsToTs[ctbAddrRs](ctbAddrRs的取值范围为0到SubPicSizeInCtbsY-1,包括端值)表示从子图像中的CTB光栅扫描下的CTB地址到子图像中的分块扫描下的CTB地址的转换,推导如下。

[0228]
```
for( ctbAddrRs = 0; ctbAddrRs < SubPicSizeInCtbsY; ctbAddrRs++ ) {
tbX = ctbAddrRs % SubPicWidthInCtbsY
tbY = ctbAddrRs / SubPicWidthInCtbsY
for( i = 0; i <= num_tile_columns_minus1; i++ )
    if( tbX >= ColBd[ i ] )
        tileX = i
for( j = 0; j <= num_tile_rows_minus1; j++ )                    (6-5)
    if( tbY >= RowBd[ j ] )
        tileY = j
```

[0229]
```
CtbAddrRsToTs[ ctbAddrRs ] = 0
for( i = 0; i < tileX; i++ )
    CtbAddrRsToTs[ ctbAddrRs ] += RowHeight[ tileY ] * ColWidth[ i ]
for( j = 0; j < tileY; j++ )
    CtbAddrRsToTs[ ctbAddrRs ] += SubPicWidthInCtbsY * RowHeight[ j ]
CtbAddrRsToTs[ ctbAddrRs ] += ( tbY – RowBd[ tileY ]      )    *
ColWidth[ tileX ] + tbX – ColBd[ tileX ]
}
```

[0230] 列表CtbAddrTsToRs[ctbAddrTs](ctbAddrTs的取值范围为0到SubPicSizeInCtbsY-1,包括端值)表示从分块扫描下的CTB地址到子图像中的CTB光栅扫描下的CTB地址的转换,推导如下。

[0231] for(ctbAddrRs ＝ 0; ctbAddrRs < SubPicSizeInCtbsY; ctbAddrRs++) (6-6)

[0232] CtbAddrTsToRs[CtbAddrRsToTs[ctbAddrRs]]＝ctbAddrRs

[0233] 列表TileId[ctbAddrTs](ctbAddrTs的取值范围为0到SubPicSizeInCtbsY-1,包括端值)表示从子图像中的分块扫描下的CTB地址到分块ID的转换,推导如下。

[0234]
```
for( j = 0, tileIdx = 0; j <= num_tile_rows_minus1; j++ )
for( i = 0; i <= num_tile_columns_minus1; i++, tileIdx++ )
    for( y = RowBd[ j ]; y < RowBd[ j + 1 ]; y++ )                (6-7)
        for( x = ColBd[ i ]; x < ColBd[ i + 1 ]; x++ )
            TileId[ CtbAddrRsToTs[ y * SubPicWidthInCtbsY+ x ] ] = tileIdx
```

[0235] 列表NumCtusInTile[tileIdx](tileIdx的取值范围为0到NumTilesInSubPic-1,包括端值)表示从分块索引到分块中的CTU数量的转换,推导如下:

[0236]
```
for( j = 0, tileIdx = 0; j <= num_tile_rows_minus1; j++ )
for( i = 0; i <= num_tile_columns_minus1; i++, tileIdx++ )      (6-8)
    NumCtusInTile[ tileIdx ] = ColWidth[ i ] * RowHeight[ j ]
```

[0237] 列表FirstCtbAddrTs[tileIdx](tileIdx的取值范围为0到NumTilesInSubPic-1,

包括端值)表示从分块ID到分块中的第一个CTB的分块扫描下的CTB地址的转换,推导如下。

[0238]
```
for( ctbAddrTs = 0, tileIdx = 0, tileStartFlag = 1; ctbAddrTs < SubPicSizeInCtbsY; ctbAddrTs++ ) {
if( tileStartFlag ) {
    FirstCtbAddrTs[ tileIdx ] = ctbAddrTs                                    (6-9)
    tileStartFlag = 0
}
tileEndFlag = ctbAddrTs = = SubPicSizeInCtbsY – 1 | | TileId[ ctbAddrTs + 1 ]
!= TileId[ ctbAddrTs ]
if( tileEndFlag ) {
    tileIdx++
    tileStartFlag = 1
}
}
```

[0239] ColumnWidthInLumaSamples[i]的值表示以亮度样本为单位的第i个分块列的宽度,被设置为ColWidth[i]<<CtbLog2SizeY(i的取值范围为0到num_tile_columns_minus1,包括端值)。RowHeightInLumaSamples[j]的值表示以亮度样本为单位的第j个分块行的高度,被设置为RowHeight[j]<<CtbLog2SizeY(j的取值范围为0到num_tile_rows_minus1,包括端值)。

[0240] 示例性序列参数集RBSP语法如下所示。

[0241]

seq_parameter_set_rbsp( ) {	描述符
sps_max_sub_layers_minus1	u(3)
sps_reserved_zero_5bits	u(5)

[0242]

seq_parameter_set_rbsp( ) {	描述符
profile_tier_level(sps_max_sub_layers_minus1)	
sps_seq_parameter_set_id	ue(v)
chroma_format_idc	ue(v)
if(chroma_format_idc = = 3)	
separate_colour_plane_flag	u(1)
pic_width_in_luma_samples	ue(v)
pic_height_in_luma_samples	ue(v)
bit_depth_luma_minus8	ue(v)
bit_depth_chroma_minus8	ue(v)
num_sub_pics_minus1	ue(v)
sub_pic_id_len_minus1	ue(v)
if(num_sub_pics_minus1 > 0)	
sub_pic_level_present_flag	u(1)
for (i = 0; i <= num_sub_pics_minus1; i++) {	
sub_pic_id[i]	u(v)
if(num_subpics_minus1 > 0) {	
sub_pic_treated_as_pic_flag[i]	u(1)
if(sub_pic_treated_as_pic_flag[i] && sub_pic_level_present_flag)	
sub_pic_level_idc[i]	u(8)
sub_pic_x_offset[i]	ue(v)
sub_pic_y_offset[i]	ue(v)
sub_pic_width_in_luma_samples[i]	ue(v)
sub_pic_height_in_luma_samples[i]	ue(v)
}	
}	
log2_max_pic_order_cnt_lsb_minus4	ue(v)
...	

[0243] 示例性图像参数集RBSP语法如下所述。

[0244]

pic_parameter_set_rbsp( ) {	描述符
pps_pic_parameter_set_id	ue(v)

[0245]

pic_parameter_set_rbsp( ) {	描述符
pps_seq_parameter_set_id	ue(v)
loop_filter_across_sub_pic_enabled_flag	u(1)
single_tile_in_sub_pic_flag	u(1)
if(!single_tile_in_sub_pic_flag) {	
num_tile_columns_minus1	ue(v)
num_tile_rows_minus1	ue(v)
uniform_tile_spacing_flag	u(1)
if(!uniform_tile_spacing_flag) {	
for(i = 0; i < num_tile_columns_minus1; i++)	
tile_column_width_minus1[i]	ue(v)
for(i = 0; i < num_tile_rows_minus1; i++)	
tile_row_height_minus1[i]	ue(v)
}	
single_tile_per_tile_group_flag	u(1)
if(!single_tile_per_tile_group_flag)	
rect_tile_group_flag	u(1)
if(rect_tile_group_flag && !single_tile_per_tile_group_flag) {	
num_tile_groups_in_sub_pic_minus1	ue(v)
for(i = 0; i <= num_tile_groups_in_sub_pic_minus1; i++) {	
if(i > 0)	
top_left_tile_idx[i]	u(v)
bottom_right_tile_idx[i]	u(v)
}	
}	
loop_filter_across_tiles_enabled_flag	u(1)
}	
for(i = 0; i < 2; i++)	
num_ref_idx_default_active_minus1[i]	ue(v)

[0246] 示例性通用分块组头语法如下所示。

[0247]

tile_group_header( ) {	描述符
tile_group_pic_parameter_set_id	ue(v)
tile_group_sub_pic_id	u(v)
if(NumTilesInSubPic > 1 | | !(rect_tile_group_flag && num_tile_group_in_sub_pic_minus1 = = 0))	
tile_group_address	u(v)

[0248] 示例性编码树单元语法如下所示。

[0249]

coding_tree_unit( ) {	描述符
xCtb = ((CtbAddrInRs % SubPicWidthInCtbsY) << CtbLog2SizeY) + sub_pic_x_offset[SubPicIdx[tile_group_subpic_id]]	
yCtb = ((CtbAddrInRs / SubPicWidthInCtbsY) << CtbLog2SizeY) + sub_pic_y_offset[SubPicIdx[tile_group_subpic_id]]	
...	

[0250] 示例性序列参数集RBSP语义如下所述。

[0251] bit_depth_chroma_minus8表示色度阵列BitDepthC中的样本的位深度,色度量化参数范围偏移QpBdOffsetC的值如下所述:

[0252] BitDepthC = 8 + bit_depth_chroma_minus8 (7-4)

[0253] QpBdOffsetC = 6 * bit_depth_chroma_minus8 (7-5)

[0254] bit_depth_chroma_minus8的取值范围应该为0到8,包括端值。

[0255] num_sub_pics_minus1+1表示CVS中的每个经译码图像中的子图像的数量。num_sub_pics_minus1的取值范围应该为0到1024,包括端值。sub_pic_id_len_minus1+1表示用于表示SPS中的语法元素sub_pic_id[i]和分块组头中的语法元素tile_group_sub_pic_id的比特数量。sub_pic_id_len_minus1的取值范围应该为Ceil(Log2(num_sub_pic_minus1+1)-1到9,包括端值。sub_pic_level_present_flag被设置为1,表示语法元素sub_pic_level_idc[i]可能存在。sub_pic_level_present_flag被设置为0,表示语法元素sub_pic_level_idc[i]不存在。sub_pic_id[i]表示CVS中每个经译码图像中的第i个子图像的子图像ID。sub_pic_id[i]的长度为(sub_pic_id_len_minus1+1)个比特。

[0256] sub_pic_treated_as_pic_flag[i]被设置为1,表示CVS中每个经译码图像的第i个子图像在不包括环内滤波操作的解码过程中当作图像处理。sub_pic_treated_as_pic_flag[i]被设置为0,表示CVS中每个经译码图像中的第i个子图像在不包括环内滤波操作的解码过程中不当作图像处理。sub_pic_level_idc[i]表示第i个子图像序列符合的级别,其中,第i个子图像序列只包括CVS中的子图像ID等于sub_pic_id[i]的子图像的VCL NAL单元及其相关的非VCL NAL单元。sub_pic_x_offset[i]表示第i个子图像中的左上角亮度样本相对于CVS中每个图像中的左上角亮度样本的水平偏移,以亮度样本为单位。当sub_pic_x_offset[i]不存在时,ref_pic_list_idx[i]的值被推断为0。sub_pic_y_offset[i]表示第i个子图像中的左上角亮度样本相对于CVS中每个图像中的左上角亮度样本的垂直偏移,以亮度样本为单位。当sub_pic_y_offset[i]不存在时,sub_pic_y_offset[i]的值被推断为

0。sub_pic_width_in_luma_samples[i]表示CVS中每个图像中的第i个子图像的宽度,以亮度样本为单位。当sub_pic_x_offset[i]和sub_pic_width_in_luma_samples[i]之和小于pic_width_in_luma_samples时,sub_pic_width_in_luma_samples[i]的值应该为CtbSizeY的整数倍。当sub_pic_width_in_luma_samples[i]不存在时,则sub_pic_width_in_luma_samples[i]的值被推断为pic_width_in_luma_samples。sub_pic_height_in_luma_samples[i]表示CVS中每个图像中的第i个子图像的高度,以亮度样本为单位。当sub_pic_y_offset[i]与sub_pic_height_in_luma_samples[i]之和小于pic_height_in_luma_samples时,sub_pic_height_in_luma_samples[i]的值应该为CtbSizeY的整数倍。当sub_pic_height_in_luma_samples[i]不存在时,则sub_pic_height_in_luma_samples[i]的值被推断为pic_height_in_luma_samples。

[0257] 对于码流一致性,以下约束条件适用。对于i和j的任何整数值,当i等于j时,sub_pic_id[i]和sub_pic_id[j]的值应该不相同。对于任意两个子图像subpicA和subpicB,当subpicA的子图像ID小于subpicB的子图像ID时,subPicA的任何经译码分块组NAL单元应该在解码顺序上位于subPicB的任何经译码分块组NAL单元之后。子图像的形状应该确保每个子图像在解码时使其整个左边界和整个上边界由图像边界组成或由先前解码的一个或多个子图像的边界组成。

[0258] spId值等于sub_pic_id[i](i的取值范围为0到num_sub_pics_minus1,包括端值)的列表SubPicIdx[spId]表示从子图像ID到子图像索引的转换,推导如下:

[0259] for(i＝0;i<＝num_sub_pics_minus1;i++)

[0260] SubPicIdx[sub_pic_id[i]] ＝ i (7-5)

[0261] log2_max_pic_order_cnt_lsb_minus4表示用于图像顺序编号的解码过程中的变量MaxPicOrderCntLsb的值,如下所述:

[0262] MaxPicOrderCntLsb ＝ 2(log2_max_pic_order_cnt_lsb_minus4 + 4) (7-5)

[0263] log2_max_pic_order_cnt_lsb_minus4的取值范围应该为0到12,包括端值。

[0264] 示例性图像参数集RBSP语义如下所述。

[0265] 当PPS语法元素pps_seq_parameter_set_id和loop_filter_across_tiles_enabled_flag存在时,其中每个语法元素的值在经译码图像的分块组头参考的所有PPS中应该是相同的。pps_pic_parameter_set_id标识SPPS,以供其它语法元素参考。pps_pic_parameter_set_id的取值范围应该为0到63,包括端值。pps_seq_parameter_set_id表示活动SPS的sps_seq_parameter_set_id的值。pps_seq_parameter_set_id的取值范围应该为0到15,包括端值。loop_filter_across_sub_pic_enabled_flag被设置为1,表示环内滤波操作可以跨参考PPS的子图像的边界执行。loop_filter_across_sub_pic_enabled_flag被设置为0,表示环内滤波操作不跨参考PPS的子图像的边界执行。

[0266] single_tile_in_sub_pic_flag被设置为1,表示参考PPS的每个子图像中只包括一个分块。single_tile_in_pic_flag被设置为0,表示参考PPS的每个子图像中包括一个以上分块。num_tile_columns_minus1+1表示分割子图像的分块列的数量。num_tile_columns_minus1的取值范围应该为0到1024,包括端值。当num_tile_columns_minus1不存在时,num_tile_columns_minus1的值被推断为0。num_tile_rows_minus1+1表示分割子图像的分块行的数量。num_tile_rows_minus1的取值范围应该为0到1024,包括端值。当num_

tile_rows_minus1不存在时,num_tile_rows_minus1的值被推断为0。变量NumTilesInSubPic被设置为(num_tile_columns_minus1+1)*(num_tile_rows_minus1+1)。当single_tile_in_sub_pic_flag等于0时,NumTilesInSubPic应该大于1。

[0267] uniform_tile_spacing_flag被设置为1,表示分块列边界和分块行边界是跨子图像均匀分布的。uniform_tile_spacing_flag被设置为0,表示分块列边界和分块行边界不是跨子图像均匀分布的,而是使用语法元素tile_column_width_minus1[i]和tile_row_height_minus1[i]显式指示的。当uniform_tile_spacing_flag不存在时,则uniform_tile_spacing_flag的值被推断为1。tile_column_width_minus1[i]+1表示第i个分块列的宽度,以CTB为单位。tile_row_height_minus1[i]+1表示第i个分块行的高度,以CTB为单位。single_tile_per_tile_group被设置为1,表示参考PPS的每个分块组包括一个分块。single_tile_per_tile_group被设置为0,表示参考PPS的分块组可以包括一个以上分块。

[0268] rect_tile_group_flag被设置为0,表示子图像中的每个分块组内的分块处于光栅扫描顺序下,并且分块组信息未指示在PPS中。rect_tile_group_flag被设置为1,表示每个分块组内的分块覆盖子图像中的矩形区域或方形区域,并且分块组信息指示在PPS中。当single_tile_per_tile_group_flag被设置为1时,rect_tile_group_flag被推断为1。num_tile_groups_in_sub_pic_minus1+1表示参考PPS的每个子图像中的分块组的数量。num_tile_groups_in_sub_pic_minus1的取值范围应该为0到NumTilesInSubPic-1,包括端值。当num_tile_groups_in_sub_pic_minus1不存在且single_tile_per_tile_group_flag等于1时,则num_tile_groups_in_sub_pic_minus1的值被推断为NumTilesInSubPic-1。

[0269] top_left_tile_idx[i]表示位于子图像中的第i个分块组的左上角的分块的分块索引。对于任何不等于j的i,top_left_tile_idx[i]的值应该不等于top_left_tile_idx[j]的值。当top_left_til_idx[i]不存在时,top_left_til_idx[i]的值被推断为i。语法元素top_left_til_idx[i]的长度为Ceil(Log2(NumTilesInSubPic)个比特。bottom_right_tile_idx[i]表示位于子图像中的第i个分块组的右下角的分块的分块索引。当single_tile_per_tile_group_flag被设置为1时,bottom_right_tile_idx[i]被推断为top_left_tile_idx[i]。语法元素bottom_right_tile_idx[i]的长度为Ceil(Log2(NumTilesInSubPic))个比特。

[0270] 码流一致性的要求是,任何特定分块应该只包括在一个分块组中。变量NumTilesInTileGroup[i]表示子图像中的第i个分块组中的分块的数量,该变量与相关变量推导如下:

[0271] deltaTileIdx=bottom_right_tile_idx[i]-top_left_tile_idx[i]

[0272] NumTileRowsInTileGroupMinus1[i]=deltaTileIdx/(num_tile_columns_minus1+1)

[0273] NumTileColumnsInTileGroupMinus1[i] = deltaTileIdx % (num_tile_columns_minus1 + 1) (7-33)

[0274] NumTilesInTileGroup[i]=(NumTileRowsInTileGroupMinus1[i]+1)*

[0275] (NumTileColumnsInTileGroupMinus1[i]+1)

[0276] loop_filter_across_tiles_enabled_flag被设置为1,表示环内滤波操作可以跨参考PPS的子图像中的分块边界执行。loop_filter_across_tiles_enabled_flag被设置为

0,表示环内滤波操作不跨参考PPS的子图像中的分块边界执行。环内滤波操作包括去块效应滤波器操作、样本自适应偏移滤波器操作和自适应环路滤波器操作。当loop_filter_across_tiles_enabled_flag不存在时,则loop_filter_across_tiles_enabled_flag的值被推断为1。num_ref_idx_default_active_minus1[i]+1在i等于0时,表示num_ref_idx_active_override_flag等于0的P条带组或B条带组的变量NumRefIdxActive[0]的推断值;在i等于1时,表示num_ref_idx_active_override_flag等于0的B条带组的NumRefIdxActive[1]的推断值。num_ref_idx_default_active_minus1[i]的取值范围应该为0到14,包括端值。

[0277] 示例性通用分块组头语义如下所述。当分块组头语法元素tile_group_pic_order_cnt_lsb和tile_group_temporal_mvp_enabled_flag存在时,其中每个语法元素的值在经译码图像的所有分块组头中应该是相同的。当tile_group_pic_parameter_set_id存在时,tile_group_pic_parameter_set_id的值在经译码子图像的所有分块组头中应该是相同的。tile_group_pic_parameter_set_id表示当前使用的PPS的pps_pic_parameter_set_id的值。tile_group_pic_parameter_set_id的取值范围应该为0到63,包括端值。码流一致性的要求是,当前图像的TemporalId的值应该大于或等于当前图像中的分块组参考的每个PPS的TemporalId的值。tile_group_sub_pic_id标识分块组头所属的子图像。tile_group_sub_pic_id的长度为(sub_pic_id_len_minus1+1)个比特。tile_group_sub_pic_id的值对于经译码子图像的所有分块组头应该是相同的。

[0278] 变量SubPicWidthInCtbsY、SubPicHeightInCtbsY和SubPicSizeInCtbsY推导如下:

[0279] i＝SubPicIdx[tile_group_subpic_id]

[0280] SubPicWidthInCtbsY＝

[0281] Ceil(sub_pic_width_in_luma_samples[i]÷CtbSizeY)(7-34)

[0282] SubPicHeightInCtbsY＝Ceil(sub_pic_height_in_luma_samples[i]÷CtbSizeY)

[0283] SubPicSizeInCtbsY＝SubPicWidthInCtbsY*SubPicHeightInCtbsY

[0284] 以下变量通过调用CTB光栅和分块扫描转换过程推导出:列表ColWidth[i](i的取值范围为0到num_tile_columns_minus1,包括端值),表示第i个分块列的宽度,以CTB为单位;列表RowHeight[j](j的取值范围为0到num_tile_rows_minus1,包括端值),表示第j个分块行的高度,以CTB为单位;列表ColBd[i](i的取值范围为0到num_tile_columns_minus1+1,包括端值),表示第i个分块列边界的位置,以CTB为单位;列表RowBd[j](j的取值范围为0到num_tile_rows_minus1+1,包括端值),表示第j个分块行边界的位置,以CTB为单位;列表CtbAddrRsToTs[ctbAddrRs](ctbAddrRs的取值范围为0到SubPicSizeInCtbsY-1,包括端值),表示从子图像中的CTB光栅扫描下的CTB地址到子图像中的分块扫描下的CTB地址的转换;列表CtbAddrTsToRs[ctbAddrTs](ctbAddrTs的取值范围为0到SubPicSizeInCtbsY-1,包括端值),表示子图像中的分块扫描下的CTB地址到子图像中的CTB光栅扫描下的CTB地址的转换;列表TileId[ctbAddrTs](ctbAddrTs的取值范围为0到SubPicSizeInCtbsY-1,包括端值),表示子图像中的分块扫描下的CTB地址到分块ID的转换;列表NumCtusInTile[tileIdx](tileIdx的取值范围为0到NumTilesInSubPic-1,包括端值),表示从分块索引到

分块中的CTU数量的转换;列表FirstCtbAddrTs[tileIdx](tileIdx的取值范围为0到NumTilesInSubPic-1,包括端值),表示从分块ID到分块中的第一个CTB的分块扫描下的CTB地址的转换;列表ColumnWidthInLumaSamples[i](i的取值范围为0到num_tile_columns_minus1,包括端值),表示第i个分块列的宽度,以亮度样本为单位;列表RowHeightInLumaSamples[j](j的取值范围为0到num_tile_rows_minus1,包括端值),表示第j个分块行的高度,以亮度样本为单位。

[0285] ColumnWidthInLumaSamples[i](i的取值范围为0到num_tile_columns_minus1,包括端值)和RowHeightInLumaSamples[j](j的取值范围为0到num_tile_rows_minus1,包括端值)的值应该均大于0。变量SubPicLeftBoundaryPos、SubPicTopBoundaryPos、SubPicRightBoundaryPos和SubPicBotBoundaryPos推导如下:

[0286]
```
i = SubPicIdx[ tile_group_subpic_id ]
if( sub_pic_treated_as_pic_flag[ i ] ) {
SubPicLeftBoundaryPos = sub_pic_x_offset[ i ]
SubPicRightBoundaryPos = SubPicLeftBoundaryPos + sub_pic_width_in_luma_samples[ i ] – 1
SubPicTopBoundaryPos = sub_pic_y_offset[ i ]
SubPicBotBoundaryPos = SubPicTopBoundaryPos + sub_pic_height_in_luma_samples[ i ] – 1        (7-34)
}
```

[0287] 对于当前子图像中的每个分块,索引i=0...NumTilesInSubPic-1,变量TileLeftBoundaryPos[i]、TileTopBoundaryPos[i]、TileRightBoundaryPos[i]和TileBotBoundaryPos[i]推导如下:

[0288]
```
tileColIdx = i % ( num_tile_columns_minus1 + 1 )
tileRowIdx = i / ( num_tile_columns_minus1 + 1 )
 TileLeftBoundaryPos[ i ] = SubPicLeftBoundaryPos + ( ColBd[ tileColIdx ] << CtbLog2SizeY ) )
TileRightBoundaryPos[ i ] = SubPicLeftBoundaryPos +
        ( ( ColBd[ tileColIdx ] + ColWidth[ tileColIdx ] ) << CtbLog2SizeY ) – 1
if( TileRightBoundaryPos[ i ] > pic_width_in_luma_samples – 1)                    (7-41)
    TileRightBoundaryPos[ i ] = pic_width_in_luma_samples – 1
 TileTopBoundaryPos[ i ] = SubPicTopBoundaryPos + ( RowBd[ tileRowIdx ] << CtbLog2SizeY )
TileBotBoundaryPos[ i ] = SubPicTopBoundaryPos +
    ( ( RowBd[ tileRowIdx ] + RowHeight[ tileRowIdx ] ) << CtbLog2SizeY ) – 1
if( TileBotBoundaryPos[ i ] > pic_height_in_luma_samples – 1 )
TileBotBoundaryPos[ i ] = pic_height_in_luma_samples – 1
```

[0289] tile_group_address表示分块组中的第一个分块的分块地址。当tile_group_address不存在时,tile_group_address的值被推断为0。如果rect_tile_group_flag等于0,则以下内容适用:分块地址为分块ID;tile_group_address的长度为Ceil(Log2(NumTilesInSubPic))个比特;tile_group_address的取值范围应该为0到NumTilesInSubPic-1,包括端值。否则(rect_tile_group_flag等于1),以下内容适用:分块地址为子图像中的多个分块组中的分块组的分块组索引;tile_group_address的长度为Ceil(Log2(num_tile_groups_in_sub_pic_minus1+1))个比特;tile_group_address的取值范围应该为0到num_tile_groups_in_sub_pic_minus1,包括端值。

[0290] 码流一致性的要求为以下约束条件适用。tile_group_address的值不应该等于同一经译码图像的任何其它经译码分块组NAL单元的tile_group_address的值。子图像中的分块组应该按照它们tile_group_address值的递增顺序排列。子图像中的分块组的形状应

该确保每个分块在解码时使其整个左边界和整个上边界由子图像边界组成或由先前解码的一个或多个分块的边界组成。

[0291] num_tiles_in_tile_group_minus1在存在时表示分块组中的分块的数量减1。num_tiles_in_tile_group_minus1的取值范围应该为0到NumTilesInSubPic-1,包括端值。当num_tiles_in_tile_group_minus1不存在时,则num_tiles_in_tile_group_minus1的值被推断为0。变量NumTilesInCurrTileGroup表示当前分块组中的分块数量,TgTileIdx[i]表示当前分块组中的第i个分块的分块索引,两者推导如下:

[0292]
```
if( rect_tile_group_flag ) {
NumTilesInCurrTileGroup = NumTilesInTileGroup[ tile_group_address ]
tileIdx = top_left_tile_idx[ tile_group_address ]
for( j = 0, tIdx = 0; j <
    NumTileRowsInTileGroupMinus1[ tile_group_address ] + 1;                    (7-35)
            j++, tileIdx += num_tile_columns_minus1 + 1 )
    for( i = 0, currTileIdx = tileIdx; i < NumTileColumnsInTileGroupMinus1[ tile_group_address ] + 1;
                    i++, currTileIdx++, tIdx++ )
        TgTileIdx[ tIdx ] = currTileIdx
```

[0293]
```
} else {
NumTilesInCurrTileGroup = num_tiles_in_tile_group_minus1 + 1
TgTileIdx[ 0 ] = tile_group_address
for( i = 1; i < NumTilesInCurrTileGroup; i++ )
    TgTileIdx[ i ] = TgTileIdx[ i – 1 ] + 1
}
```

[0294] tile_group_type表示分块组的译码类型。

[0295] 时间亮度运动矢量预测的示例性推导过程如下所述。变量mvLXCol和availableFlagLXCol推导如下:如果tile_group_temporal_mvp_enabled_flag等于0,则mvLXCol的两个分量都被设置为0,且availableFlagLXCol被设置为0。否则(tile_group_temporal_mvp_enabled_flag等于1),以下按顺序执行的步骤适用:右下角并置运动矢量和右下边界样本位置推导如下:

[0296] xColBr = xCb + cbWidth (8-414)

[0297] yColBr = yCb + cbHeight (8-415)

[0298] rightBoundaryPos=sub_pic_treated_as_pic_flag[SubPicIdx[tile_group_subpic_id]]?

[0299] SubPicRightBoundaryPos:pic_width_in_luma_samples-1(8-415)

[0300] botBoundaryPos=sub_pic_treated_as_pic_flag[SubPicIdx[tile_group_subpic_id]]?

[0301] SubPicBotBoundaryPos:pic_height_in_luma_samples-1(8-415)

[0302] 如果yCb>>CtbLog2SizeY等于yColBr>>CtbLog2SizeY,yColBr小于或等于botBoundaryPos,且xColBr小于或等于rightBoundaryPos,则以下内容适用:变量colCb表示亮度译码块,其覆盖ColPic表示的并置图像内的由((xColBr>>3)<<3,(yColBr>>3)<<3)给出的修改位置。亮度位置(xColCb,yColCb)被设置为由colCb表示的并置亮度译码块的左上样本相对于由ColPic表示的并置图像的左上亮度样本的位置。调用并置运动矢量的推导过程,其中,currCb、colCb、(xColCb,yColCb)、refIdxLX和设置为0的sbFlag作为输入,输出

被赋给mvLXCol和availableFlagLXCol。否则,mvLXCol的两个分量都被设置为0,且availableFlagLXCol被设置为0。

[0303] 示例性亮度样本双线性插值过程如下所述。对于i＝0..1,具有整数个样本单位的色度位置(xInti,yInti)推导如下:如果sub_pic_treated_as_pic_flag[SubPicIdx[tile_group_subpic_id]]等于1,则以下内容适用:

[0304] xInti = Clip3(SubPicLeftBoundaryPos, SubPicRightBoundaryPos, xIntL + i) (8-458)

[0305] yInti = Clip3(SubPicTopBoundaryPos, SubPicBotBoundaryPos, yIntL + i) (8-458)

[0306] 否则(sub_pic_treated_as_pic_flag[SubPicIdx[tile_group_subpic_id]]等于0),则以下内容适用:

[0307] xInti＝sps_ref_wraparound_enabled_flag?

[0308] ClipH((sps_ref_wraparound_offset_minus1+1)*MinCbSizeY,picW,(xIntL+i)):

[0309] (8-459)

[0310] Clip3(0,picW-1,xIntL+i)

[0311] yInti = Clip3(0, picH - 1, yIntL + i) (8-460)

[0312] 亮度样本8抽头插值滤波过程如下所述。对于i＝0..7,具有整数个样本单位的色度位置(xInti,yInti)推导如下:如果sub_pic_treated_as_pic_flag[SubPicIdx[tile_group_subpic_id]]等于1,则以下内容适用:

[0313] xInti = Clip3(SubPicLeftBoundaryPos, SubPicRightBoundaryPos, xIntL + i - 3) (8-830)

[0314] yInti = Clip3(SubPicTopBoundaryPos, SubPicBotBoundaryPos, yIntL + i - 3) (8-830)

[0315] 否则(sub_pic_treated_as_pic_flag[SubPicIdx[tile_group_subpic_id]]等于0),则以下内容适用:

[0316] xInti＝sps_ref_wraparound_enabled_flag?

[0317] ClipH((sps_ref_wraparound_offset_minus1+1)*MinCbSizeY,picW,xIntL+i-3):

[0318] (8-831)

[0319] Clip3(0,picW-1,xIntL+i-3)

[0320] yInti = Clip3(0, picH - 1, yIntL + i - 3) (8-832)

[0321] 示例性色度样本插值过程如下所述。变量xOffset被设置为(sps_ref_wraparound_offset_minus1+1)*MinCbSizeY)/SubWidthC。对于i＝0..3,具有整数个样本单位的色度位置(xInti,yInti)推导如下:

[0322] 如果sub_pic_treated_as_pic_flag[SubPicIdx[tile_group_subpic_id]]等于1,则以下内容适用:

[0323] xInti＝Clip3(SubPicLeftBoundaryPos/SubWidthC,SubPicRightBoundaryPos/SubWidthC,xIntL+i)

[0324] (8-844)

[0325] yInti＝Clip3(SubPicTopBoundaryPos/SubHeightC,SubPicBotBoundaryPos/SubHeightC,yIntL+i)

[0326] (8-844)

[0327] 否则(sub_pic_treated_as_pic_flag[SubPicIdx[tile_group_subpic_id]]等于0),则以下内容适用:

[0328] xInti = sps_ref_wraparound_enabled_flag ? ClipH(xOffset, picWC, xIntC + i - 1) : (8-845)

[0329] Clip3(0,picWC-1,xIntC+i-1)

[0330] yInti = Clip3(0, picHC - 1, yIntC + i - 1) (8-846)

[0331] 示例性去块效应滤波器过程如下所述。去块效应滤波器过程应用于图像中的所有译码子块边缘和变换块边缘,但以下类型的边缘除外:位于图像边界的边缘;与loop_filter_across_sub_pic_enabled_flag等于0的子图像的边界重合的边缘;与loop_filter_across_tiles_enabled_flag等于0的分块的边界重合的边缘;与tile_group_deblocking_filter_disabled_flag等于1的分块组中的或内的上边界或左边界重合的边缘;与所考虑分量的8×8样本网格边界不对应的边缘;边缘两侧都使用帧间预测的色度分量内的边缘;不是相关变换单元的边缘的色度变换块的边缘;跨IntraSubPartitionsSplit值不等于ISP_NO_SPLIT的译码单元的亮度变换块的边缘。

[0332] 一个方向的示例性去块效应滤波器过程如下所述。对于译码块宽度为log2CbW、译码块高度为log2CbH和译码块左上样本的位置为(xCb,yCb)的每个译码单元,当edgeType等于EDGE_VER且xCb％8等于0时,或当edgeType等于EDGE_HOR且yCb％8等于0时,边缘通过以下按顺序执行的步骤进行滤波。译码块宽度nCbW被设置为1<<log2CbW,译码块高度nCbH被设置为1<<log2CbH。变量filterEdgeFlag推导如下:如果edgeType等于EDGE_VER且以下一个或多个条件为真(true),则filterEdgeFlag被设置为0:当前译码块的左边界是图像的左边界。当前译码块的左边界是子图像的左边界或右边界,loop_filter_across_sub_pic_enabled_flag等于0。当前译码块的左边界是分块的左边界,loop_filter_across_tiles_enabled_flag等于0。如果edgeType等于EDGE_HOR且以下一个或多个条件为真,则变量filterEdgeFlag被设置为0。当前亮度译码块的上边界是图像的上边界。当前译码块的上边界是子图像的上边界或下边界,loop_filter_across_sub_pic_enabled_flag等于0。当前译码块的上边界是分块的上边界,loop_filter_across_tiles_enabled_flag等于0。否则,filterEdgeFlag被设置为1。

[0333] 示例性CTB修改过程如下所述。对于所有样本位置(xSi,ySj)和(xYi,yYj),其中,i＝0..nCtbSw-1,j＝0..nCtbSh-1,取决于包括覆盖recPicture[xSi][ySj]的译码块的译码单元的pcm_loop_filter_disabled_flag、pcm_flag[xYi][yYj]和cu_transquant_bypass_flag的值,以下内容适用。如果以下一个或多个条件对于所有样本位置(xSik',ySjk')和(xYik',yYjk')(k＝0..1)为真,则edgeIdx被设置0。位置(xSik',ySjk')上的样本在图像边界之外。位置(xSik',ySjk')上的样本属于不同的子图像,样本recPicture[xSi][ySj]所属的分块组中的loop_filter_across_sub_pic_enabled_flag等于0。loop_filter_across_tiles_enabled_flag等于0,位置(xSik',ySjk')上的样本属于不同的分块。

[0334] 亮度样本的示例性编码树块滤波过程如下所述。为了推导经滤波的重建亮度样本alfPictureL[x][y],当前亮度编码树块内的每个重建亮度样本recPictureL[x][y]滤波如下,其中,x,y=0..CtbSizeY-1。亮度样本组成的给定阵列recPicture内的每个对应的亮度样本(x,y)的位置(hx,vy)推导如下。如果包括位置(hx,vy)上的亮度样本的分块tileA的loop_filter_across_tiles_enabled_flag等于0,则假设变量tileIdx为tileA的分块索引,则以下内容适用:

[0335] hx=Clip3(TileLeftBoundaryPos[tileIdx],TileRightBoundaryPos[tileIdx],xCtb+x)

[0336] (8-1140)

[0337] vy=Clip3(TileTopBoundaryPos[tileIdx],TileBotBoundaryPos[tileIdx],yCtb+y)

[0338] (8-1141)

[0339] 如果包括位置(hx,vy)上的亮度样本的子图像中的loop_filter_across_sub_pic_enabled_flag等于0,则以下内容适用:

[0340] hx = Clip3(SubPicLeftBoundaryPos, SubPicRightBoundaryPos, xCtb + x) (8-1140)

[0341] vy = Clip3(SubPicTopBoundaryPos, SubPicBotBoundaryPos, yCtb + y) (8-1141)

[0342] 否则,以下内容适用:

[0343] hx = Clip3(0, pic_width_in_luma_samples - 1, xCtb + x) (8-1140)

[0344] vy = Clip3(0, pic_height_in_luma_samples - 1, yCtb + y) (8-1141)

[0345] 亮度样本的ALF转置和滤波器索引的示例性推导过程如下所述。亮度样本组成的给定阵列recPicture内的每个对应的亮度样本(x,y)的位置(hx,vy)推导如下。如果包括位置(hx,vy)上的亮度样本的分块tileA的loop_filter_across_tiles_enabled_flag等于0,则假设分块tileIdx为分块A的分块索引,以下内容适用:

[0346] hx = Clip3(TileLeftBoundaryPos[tileIdx], TileRightBoundaryPos[tileIdx], x) (8-1140)

[0347] vy = Clip3(TileTopBoundaryPos[tileIdx], TileBotBoundaryPos[tileIdx], y) (8-1141)

[0348] 否则,如果包括位置(hx,vy)上的亮度样本的子图像的loop_filter_across_sub_pic_enabled_flag等于0,则以下内容适用:

[0349] hx = Clip3(SubPicLeftBoundaryPos, SubPicRightBoundaryPos, x) (8-1140)

[0350] vy = Clip3(SubPicTopBoundaryPos, SubPicBotBoundaryPos, y) (8-1141)

[0351] 否则,以下内容适用:

[0352] hx = Clip3(0, pic_width_in_luma_samples - 1, x) (8-1145)

[0353] vy = Clip3(0, pic_height_in_luma_samples - 1, y) (8-1146)

[0354] 色度样本的示例性编码树块滤波过程如下所述。为了推导经滤波的重建色度样本alfPicture[x][y],当前色度编码树块内的每个重建色度样本recPicture[x][y]滤波如

下,其中,x,y=0..ctbSizeC-1。色度样本组成的给定阵列recPicture内的每个对应的色度样本(x,y)的位置(hx,vy)推导如下。如果包括位置(hx,vy)上的色度样本的分块tileA的loop_filter_across_tiles_enabled_flag等于0,则假设tileIdx为tileA的分块索引,以下内容适用:

[0355] hx=Clip3(TileLeftBoundaryPos[tileIdx]/SubWidthC,

[0356] TileRightBoundaryPos[tileIdx]/SubWidthC,xCtb+x)(8-1140)

[0357] vy=Clip3(TileTopBoundaryPos[tileIdx]/SubWidthC,

[0358] TileBotBoundaryPos[tileIdx]/SubWidthC,yCtb+y)(8-1141)

[0359] 否则,如果包括位置(hx,vy)上的色度样本的子图像的loop_filter_across_sub_pic_enabled_flag等于0,则以下内容适用:

[0360] hx=Clip3(SubPicLeftBoundaryPos/SubWidthC,

[0361] SubPicRightBoundaryPos/SubWidthC,xCtb+x)(8-1140)

[0362] vy=Clip3(SubPicTopBoundaryPos/SubWidthC,

[0363] SubPicBotBoundaryPos/SubWidthC,yCtb+y)(8-1141)

[0364] 否则,以下内容适用:

[0365] hx = Clip3(0, pic_width_in_luma_samples / SubWidthC - 1, xCtbC + x) (8-1177)

[0366] vy = Clip3(0, pic_height_in_luma_samples / SubHeightC - 1, yCtbC + y) (8-1178)

[0367] 变量sum推导如下:

[0368]
$$\begin{aligned} sum = {} & AlfCoeffC[0] * (recPicture[hx, vy+2] + recPicture[hx, vy-2]) + \\ & AlfCoeffC[1] * (recPicture[hx+1, vy+1] + recPicture[hx-1, vy-1]) + \\ & AlfCoeffC[2] * (recPicture[hx, vy+1] + recPicture[hx, vy-1]) + \\ & AlfCoeffC[3] * (recPicture[hx-1, vy+1] + recPicture[hx+1, vy-1]) + \\ & AlfCoeffC[4] * (recPicture[hx+2, vy] + recPicture[hx-2, vy]) + \\ & AlfCoeffC[5] * (recPicture[hx+1, vy] + recPicture[hx-1, vy]) + \\ & AlfCoeffC[6] * recPicture[hx, vy] \end{aligned} \quad (8\text{-}1179)$$

[0369] sum = (sum + 64) >> 7 (8-1180)

[0370] 修改后的经滤波重建色度图像样本alfPicture[xCtbC+x][yCtbC+y]推导如下:

[0371] alfPicture[xCtbC + x][yCtbC + y] = Clip3(0, (1 << BitDepthC) - 1, sum) (8-1181)

[0372] 图12为示例性视频译码设备1200的示意图。视频译码设备1200适用于实现本文描述的公开示例/实施例。视频译码设备1200包括下行端口1220、上行端口1250和/或收发单元(Tx/Rx)1210。收发单元1210包括发送器和/或接收器,用于通过网络在上行和/或下行进行数据通信。视频译码设备1200还包括处理器1230和存储器1232。处理器1230包括逻辑单元和/或中央处理单元(central processing unit,CPU),以处理数据。存储器1232用于存储数据。视频译码设备1200还可以包括与上行端口1250和/或下行端口1220耦合的电组件、光电(optical-to-electrical,OE)组件、电光(electrical-to-optical,EO)组件和/或无线通信组件,用于通过电通信网络、光通信网络或无线通信网络进行数据通信。视频译码设备1200还可以包括输入和/或输出(input/output,I/O)设备1260,用于与用户进行数据通

信。I/O设备1260可以包括输出设备,例如显示视频数据的显示器、输出音频数据的扬声器等。I/O设备1260还可以包括键盘、鼠标、轨迹球等输入设备和/或与这些输出设备交互的对应接口。

[0373] 处理器1230通过硬件和软件实现。处理器1230可以实现为一个或多个CPU芯片、一个或多个核(例如,实现为多核处理器)、一个或多个现场可编程门阵列(field-programmable gate array,FPGA)、一个或多个专用集成电路(application specific integrated circuit,ASIC)以及一个或多个数字信号处理器(digital signal processor,DSP)。处理器1230与下行端口1220、Tx/Rx 1210、上行端口1250和存储器1232通信。处理器1230包括译码模块1214。编码模块1214实现本文描述的公开实施例,例如方法100、方法1300和方法1400,它们可以采用环内滤波器1000、码流1100、图像500和/或当前块801和/或901,当前块801和/或901可以根据单向帧间预测600和/或双向帧间预测700使用根据模式900生成的候选列表译码。译码模块1214还可以实现本文描述的任何其它方法/机制。此外,译码模块1214可以实现编解码系统200、编码器300和/或解码器400。例如,译码模块1214可以实现如上所述的第一、第二、第三、第四、第五和/或第六示例性实现方式。因此,译码模块1214使视频译码设备1200在对视频数据进行译码时提供额外功能和/或提高译码效率。因此,译码模块1214改进了视频译码设备1200的功能,同时解决了视频译码领域特有的问题。此外,译码模块1214影响了视频译码设备1200到不同状态的转换。可选地,译码模块1214可以实现为存储在存储器1232中并由处理器1230执行的指令(例如,实现为存储在非瞬时性介质中的计算机程序产品)。

[0374] 存储器1232包括一种或多种存储器类型,例如磁盘、磁带机、固态硬盘、只读存储器(read only memory,ROM)、随机存取存储器(random access memory,RAM)、闪存、三态内容寻址存储器(ternary content-addressable memory,TCAM)、静态随机存取存储器(static random-access memory,SRAM)等。存储器1232可以用作溢出数据存储设备,以在选择程序来执行时存储这些程序以及存储在执行程序过程中读取的指令和数据。

[0375] 图13为在子图像(例如子图像510)当作图像(例如图像500)处理时将限幅函数应用于插值滤波器(例如插值滤波器913)时将视频序列编码到码流(例如码流1100)中的示例性方法1300的流程图。方法1300可以由编解码系统200、编码器300和/或视频译码设备1200在执行方法100以根据单向帧间预测600和/或双向帧间预测700使用环内滤波器1000和/或根据模式900生成的候选列表对当前块801和/或901进行编码时采用。

[0376] 方法1300可以开始于:编码器接收包括多个图像的视频序列,并根据用户输入等确定将所述视频序列编码到码流中。在步骤1301中,所述编码器将当前图像分割成子图像。所述编码器还将所述子图像分割成块。在步骤1303中,所述编码器确定根据帧间预测对所述块进行编码。相应地,所述编码器选择运动矢量对所述块进行编码。

[0377] 在步骤1305中,所述编码器将限幅函数应用于所述运动矢量指向的参考块中的样本位置。应用所述限幅函数是为了支持在所述运动矢量指向所述子图像外部时应用插值滤波器。这个过程发生在一个标志被设置为表示所述子图像当作图像处理的情况下,因此所述子图像可以进行译码以支持独立于所述图像中的其它子图像进行提取。在这个上下文中,子图像在不参考其它子图像中的数据的情况下进行译码时当作子图像处理,因此可以单独提取。

[0378] 在步骤1307中,所述编码器将所述插值滤波器应用于所述限幅函数的结果,以获得预测样本值。在一个示例中,所述插值滤波器包括亮度样本双线性插值过程。在这种情况下,所述块包括亮度样本块。此外,所述预测样本值包括预测亮度样本值。在这种情况下,所述亮度样本双线性插值过程接收包括具有整数个样本单位的亮度位置(xIntL,yIntL)的输入,所述亮度样本双线性插值过程输出预测亮度样本值(predSampleLXL)。在这种情况下,将步骤1305中的所述限幅函数应用于所述样本位置,如下所述。

[0379] 当subpic_treated_as_pic_flag[SubPicIdx]等于1时,以下内容适用:

[0380] xInti=Clip3(SubPicLeftBoundaryPos,SubPicRightBoundaryPos,xIntL+i),

[0381] yInti=Clip3(SubPicTopBoundaryPos,SubPicBotBoundaryPos,yIntL+i),

[0382] 其中,subpic_treated_as_pic_flag表示所述被设置为表示所述子图像当作子图像处理的标志,SubPicIdx表示子图像的索引,xInti和yInti表示索引i处的限幅后样本位置,SubPicRightBoundaryPos表示子图像的右边界的位置,SubPicLeftBoundaryPos表示子图像的左边界的位置,SubPicTopBoundaryPos表示子图像的上边界的位置,SubPicBotBoundaryPos表示子图像的下边界的位置,Clip3表示根据下式的所述限幅函数:

[0383] $$Clip3(x,y,z)=\begin{cases}x & ; \ z<x\\ y & ; \ z>y,\\ z & ; \ 其它\end{cases}$$

[0384] 其中,x、y和z是数字输入值。

[0385] 在另一个示例中,所述插值滤波器包括亮度样本8抽头插值滤波过程。在这种情况下,所述块包括亮度样本块,所述预测样本值包括预测亮度样本值。此外,所述亮度样本8抽头插值滤波过程接收包括具有整数个样本单位的亮度位置(xIntL,yIntL)的输入,所述亮度样本8抽头插值滤波过程输出预测亮度样本值(predSampleLXL)。此外,将步骤1305中的所述限幅函数应用于所述样本位置,如下所述。当subpic_treated_as_pic_flag[SubPicIdx]等于1时,以下内容适用:

[0386] xInti=Clip3(SubPicLeftBoundaryPos,SubPicRightBoundaryPos,xIntL+i-3),

[0387] yInti=Clip3(SubPicTopBoundaryPos,SubPicBotBoundaryPos,yIntL+i-3),

[0388] 其中,subpic_treated_as_pic_flag表示所述被设置为表示所述子图像当作子图像处理的标志,SubPicIdx表示子图像的索引,xInti和yInti表示索引i处的限幅后样本位置,SubPicRightBoundaryPos表示子图像的右边界的位置,SubPicLeftBoundaryPos表示子图像的左边界的位置,SubPicTopBoundaryPos表示子图像的上边界的位置,SubPicBotBoundaryPos表示子图像的下边界的位置,Clip3表示根据下式的所述限幅函数:

[0389] $$Clip3(x,y,z)=\begin{cases}x & ; \ z<x\\ y & ; \ z>y,\\ z & ; \ 其它\end{cases}$$

[0390] 其中,x、y和z是数字输入值。

[0391] 在另一个示例中,所述插值滤波器包括色度样本插值过程。在这种情况下,所述块包括色度样本块,所述预测样本值包括预测色度样本值。在这种情况下,所述色度样本插值过程接收包括具有整数个样本单位的色度位置(xIntC,yIntC)的输入。此外,所述色度样本插值过程输出预测色度样本值(predSampleLXC)。在这种情况下,将步骤1305中的所述限幅函数应用于所述样本位置,如下所述。当subpic_treated_as_pic_flag[SubPicIdx]等于1

时,以下内容适用:

[0392] xInti＝Clip3(SubPicLeftBoundaryPos/SubWidthC,SubPicRightBoundaryPos/SubWidthC,xIntC+i),

[0393] yInti＝Clip3(SubPicTopBoundaryPos/SubHeightC,SubPicBotBoundaryPos/SubHeightC,yIntC+i),

[0394] 其中,subpic_treated_as_pic_flag表示所述被设置为表示所述子图像当作图像处理的标志,SubPicIdx表示所述子图像的索引,xInti和yInti表示索引i处的限幅后样本位置,SubPicRightBoundaryPos表示所述子图像的右边界的位置,SubPicLeftBoundaryPos表示所述子图像的左边界的位置,SubPicTopBoundaryPos表示所述子图像的上边界的位置,SubPicBotBoundaryPos表示所述子图像的下边界的位置,SubWidthC和SubHeightC表示亮度样本与色度样本之间的水平采样率比值和垂直采样率比值,Clip3表示根据下式的所述限幅函数:

[0395] $$\mathrm{Clip3}(x,y,z)=\begin{cases}x & ; \quad z<x\\ y & ; \quad z>y,\\ z & ; \quad \text{其它}\end{cases}$$

[0396] 其中,x、y和z是数字输入值。

[0397] 在步骤1309中,所述编码器根据所述预测样本值和所述运动矢量将所述块编码到码流中。可以存储所述码流,用于发送给解码器。

[0398] 图14为在子图像(例如子图像510)当作图像(例如图像500)处理时将限幅函数应用于插值滤波器(例如插值滤波器913)的同时从码流(例如码流1100)中解码出视频序列的示例性方法1400的流程图。方法1400可以由编解码系统200、解码器400和/或视频译码设备1200在执行方法100以根据单向帧间预测600和/或双向帧间预测700使用环内滤波器1000和/或根据模式900生成的候选列表对当前块801和/或901进行解码时采用。

[0399] 方法1400可以开始于:解码器开始接收表示视频序列的经译码数据的码流(例如,方法1300的结果)。在步骤1401中,所述解码器接收包括当前图像的码流,其中,所述当前图像包括子图像。图像、子图像、条带(slice)、分块(tile)、CTU和/或它们中的其它子区域是根据帧间预测进行译码的。在步骤1402中,所述解码器确定所述子图像中的块的运动矢量。

[0400] 在步骤1403中,所述解码器将限幅函数应用于所述运动矢量指向的参考块中的样本位置。应用所述限幅函数是为了支持在所述运动矢量指向所述子图像外部时应用插值滤波器。这个过程发生在一个标志被设置为表示所述子图像当作图像处理的情况下,因此所述子图像可以进行译码以支持独立于所述图像中的其它子图像进行提取。

[0401] 在步骤1405中,所述解码器将所述插值滤波器应用于所述限幅函数的结果,以获得预测样本值。在一个示例中,所述插值滤波器包括亮度样本双线性插值过程。在这种情况下,所述块包括亮度样本块。此外,所述预测样本值包括预测亮度样本值。在这种情况下,所述亮度样本双线性插值过程接收包括具有整数个样本单位的亮度位置(xIntL,yIntL)的输入,所述亮度样本双线性插值过程输出预测亮度样本值(predSampleLXL)。在这种情况下,将步骤1403中的所述限幅函数应用于所述样本位置,如下所述。

[0402] 当subpic_treated_as_pic_flag[SubPicIdx]等于1时,以下内容适用:

[0403] xInti＝Clip3(SubPicLeftBoundaryPos,SubPicRightBoundaryPos,xIntL+i),

[0404] yInti＝Clip3(SubPicTopBoundaryPos,SubPicBotBoundaryPos,yIntL+i),

[0405] 其中,subpic_treated_as_pic_flag表示所述被设置为表示所述子图像当作子图像处理的标志,SubPicIdx表示子图像的索引,xInti和yInti表示索引i处的限幅后样本位置,SubPicRightBoundaryPos表示子图像的右边界的位置,SubPicLeftBoundaryPos表示子图像的左边界的位置,SubPicTopBoundaryPos表示子图像的上边界的位置,SubPicBotBoundaryPos表示子图像的下边界的位置,Clip3表示根据下式的所述限幅函数:

[0406] $$Clip3(x,y,z)=\begin{cases}x & ; \ z<x\\ y & ; \ z>y,\\ z & ; \ 其它\end{cases}$$

[0407] 其中,x、y和z是数字输入值。

[0408] 在另一个示例中,所述插值滤波器包括亮度样本8抽头插值滤波过程。在这种情况下,所述块包括亮度样本块,所述预测样本值包括预测亮度样本值。此外,所述亮度样本8抽头插值滤波过程接收包括具有整数个样本单位的亮度位置(xIntL,yIntL)的输入,所述亮度样本8抽头插值滤波过程输出预测亮度样本值(predSampleLXL)。此外,将步骤1403中的所述限幅函数应用于所述样本位置,如下所述。当subpic_treated_as_pic_flag[SubPicIdx]等于1时,以下内容适用:

[0409] xInti＝Clip3(SubPicLeftBoundaryPos,SubPicRightBoundaryPos,xIntL+i-3),

[0410] yInti＝Clip3(SubPicTopBoundaryPos,SubPicBotBoundaryPos,yIntL+i-3),

[0411] 其中,subpic_treated_as_pic_flag表示所述被设置为表示所述子图像当作子图像处理的标志,SubPicIdx表示子图像的索引,xInti和yInti表示索引i处的限幅后样本位置,SubPicRightBoundaryPos表示子图像的右边界的位置,SubPicLeftBoundaryPos表示子图像的左边界的位置,SubPicTopBoundaryPos表示子图像的上边界的位置,SubPicBotBoundaryPos表示子图像的下边界的位置,Clip3表示根据下式的所述限幅函数:

[0412] $$Clip3(x,y,z)=\begin{cases}x & ; \ z<x\\ y & ; \ z>y,\\ z & ; \ 其它\end{cases}$$

[0413] 其中,x、y和z是数字输入值。

[0414] 在另一个示例中,所述插值滤波器包括色度样本插值过程。在这种情况下,所述块包括色度样本块,所述预测样本值包括预测色度样本值。在这种情况下,所述色度样本插值过程接收包括具有整数个样本单位的色度位置(xIntC,yIntC)的输入。此外,所述色度样本插值过程输出预测色度样本值(predSampleLXC)。在这种情况下,将步骤1403中的所述限幅函数应用于所述样本位置,如下所述。当subpic_treated_as_pic_flag[SubPicIdx]等于1时,以下内容适用:

[0415] xInti＝Clip3(SubPicLeftBoundaryPos/SubWidthC,SubPicRightBoundaryPos/SubWidthC,xIntC+i),

[0416] yInti＝Clip3(SubPicTopBoundaryPos/SubHeightC,SubPicBotBoundaryPos/SubHeightC,yIntC+i),

[0417] 其中,subpic_treated_as_pic_flag表示所述被设置为表示所述子图像当作图像处理的标志,SubPicIdx表示所述子图像的索引,xInti和yInti表示索引i处的限幅后样本

位置,SubPicRightBoundaryPos表示所述子图像的右边界的位置,SubPicLeftBoundaryPos表示所述子图像的左边界的位置,SubPicTopBoundaryPos表示所述子图像的上边界的位置,SubPicBotBoundaryPos表示所述子图像的下边界的位置,SubWidthC和SubHeightC表示亮度样本与色度样本之间的水平采样率比值和垂直采样率比值,Clip3表示根据下式的所述限幅函数:

[0418] $$\mathrm{Clip3}(x,y,z)=\begin{cases} x & ; \ z<x \\ y & ; \ z>y, \\ z & ; \ \text{其它} \end{cases}$$

[0419] 其中,x、y和z是数字输入值。

[0420] 在步骤1407中,所述解码器根据所述预测样本值对所述块进行解码。然后,所述解码器可以转发所述块,用于显示为经解码视频序列的一部分。

[0421] 图15为在子图像(例如子图像510)当作图像(例如图像500)处理时将限幅函数应用于插值滤波器(例如插值滤波器913)的同时对码流(例如码流1100)中的图像组成的视频序列进行译码的示例性系统1500的示意图。系统1500可以由编解码系统200、编码器300、解码器400和/或视频译码设备1200等编码器和解码器实现。此外,系统1500可以在实现方法100、方法1300和/或方法1400以根据单向帧间预测600和/或双向帧间预测700使用环内滤波器1000和/或根据模式900生成的候选列表对当前块801和/或901进行译码时采用。

[0422] 系统1500包括视频编码器1502。视频编码器1502包括分割模块1503,用于将当前图像分割成子图像并将所述子图像分割成块。视频编码器1502还包括确定模块1504,用于确定根据帧间预测对所述块进行编码。视频编码器1502还包括选择模块1505,用于选择运动矢量对所述块进行编码。视频编码器1502还包括应用模块1506,用于:当所述运动矢量指向所述子图像外部时以及当一个标志被设置为表示所述子图像当作图像处理时,将限幅函数应用于参考块中的样本位置,以支持应用插值滤波器。应用模块1506还用于将所述插值滤波器应用于所述限幅函数的结果,以获得预测样本值。视频编码器1502还包括编码模块1507,用于根据所述预测样本值和所述运动矢量将所述块编码到码流中。视频编码器1502还包括存储模块1508,用于存储所述码流,用于发送给解码器。视频编码器1502还包括发送模块1509,用于将所述码流发送给视频解码器1510。视频编码器1502还可以用于执行方法1300的任一步骤。

[0423] 系统1500还包括视频解码器1510。视频解码器1510包括接收模块1511,用于接收包括当前图像的码流,其中,所述当前图像包括被根据帧间预测进行译码的子图像。视频解码器1510还包括确定模块1512,用于确定所述子图像中的块的运动矢量。视频解码器1510还包括应用模块1513,用于:当所述运动矢量指向所述子图像外部时以及当一个标志被设置为表示所述子图像当作图像处理时,将限幅函数应用于参考块中的样本位置,以支持应用插值滤波器。应用模块1513还用于将所述插值滤波器应用于所述限幅函数的结果,以获得预测样本值。视频解码器1510还包括解码模块1514,用于根据所述预测样本值对所述块进行解码。视频解码器1510还包括转发模块1515,用于转发所述块,用于显示为经解码视频序列的一部分。视频解码器1510还可以用于执行方法1400的任一步骤。

[0424] 当第一组件与第二组件之间除线、迹线或其它介质之外不存在中间组件时,第一组件与第二组件直接耦合。当第一组件与第二组件之间除线、迹线或其它介质之外还存在

中间组件时,第一组件与第二组件间接耦合。术语“耦合”及其变体包括直接耦合和间接耦合。除非另有说明,否则使用术语“约”表示包括后续数字±10%的范围。

[0425] 还应当理解,本文中阐述的示例性方法的步骤不一定需要按照所描述的顺序执行,并且这些方法的步骤的顺序应当理解为仅仅是示例性的。同理,在与本发明各种实施例相一致的方法中,这些方法可以包括其它步骤,并且某些步骤可以省略或组合。

[0426] 虽然本发明提供了若干个实施例,但应当理解,在不脱离本发明的精神或范围的情况下,所公开的系统和方法可以通过其它多种具体形式体现。本发明的示例将被视为说明性而非限制性的,且本发明并不限于本文中所给出的详细内容。例如,各种元件或组件可以组合或集成在另一系统中,或者一些特征可以省略或不实施。

[0427] 另外,在不脱离本发明范围的情况下,各种实施例中描述和说明为离散或单独的技术、系统、子系统和方法可以与其它系统、组件、技术或方法组合或集成。其它变更、替换、更改示例能够由本领域技术人员确定,并且可以在不脱离本文公开的精神和范围的情况下进行。

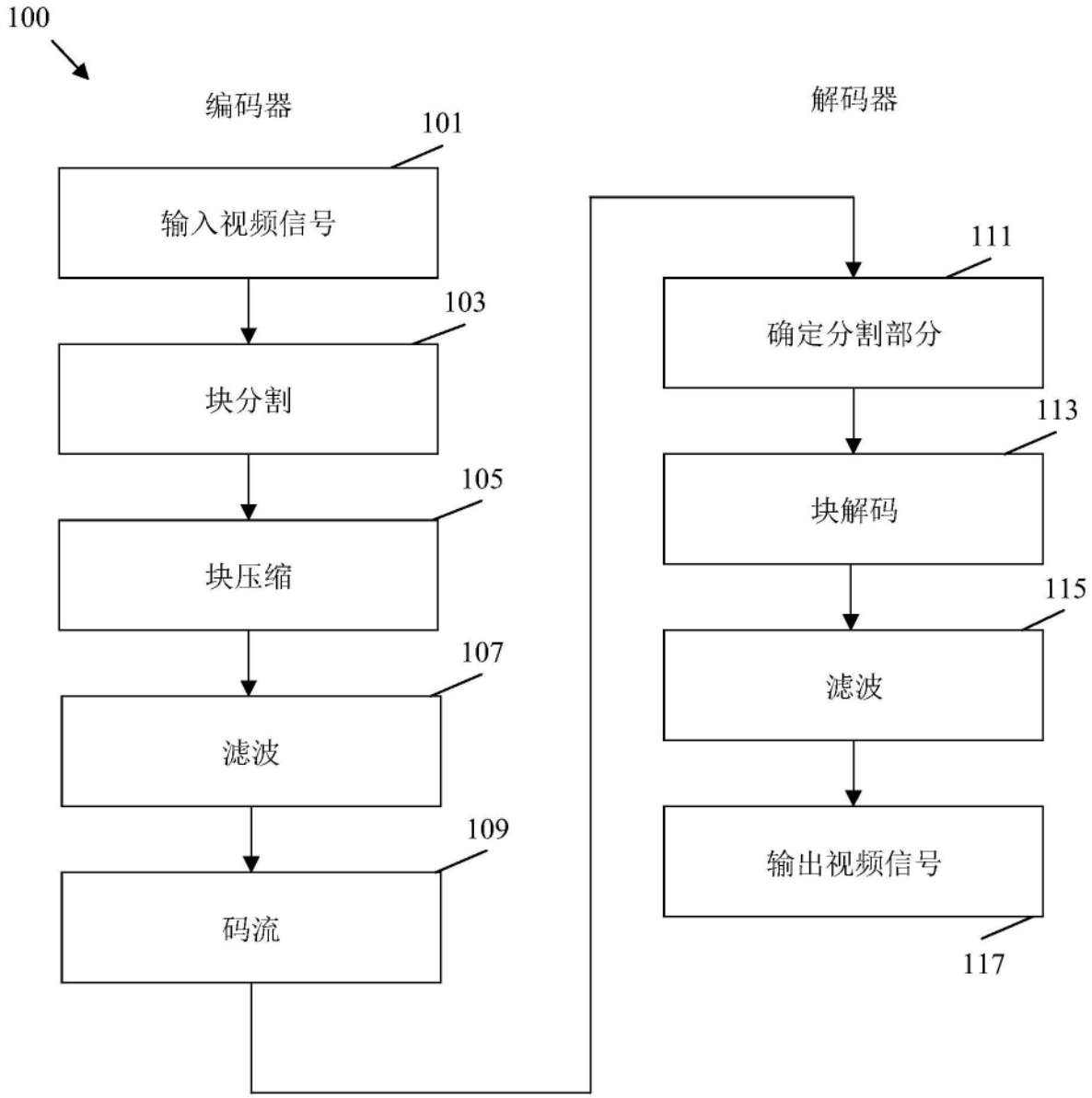

100
编码器
101
输入视频信号
103
块分割
105
块压缩
107
滤波
109
码流
解码器
111
确定分割部分
113
块解码
115
滤波
输出视频信号
117

图1

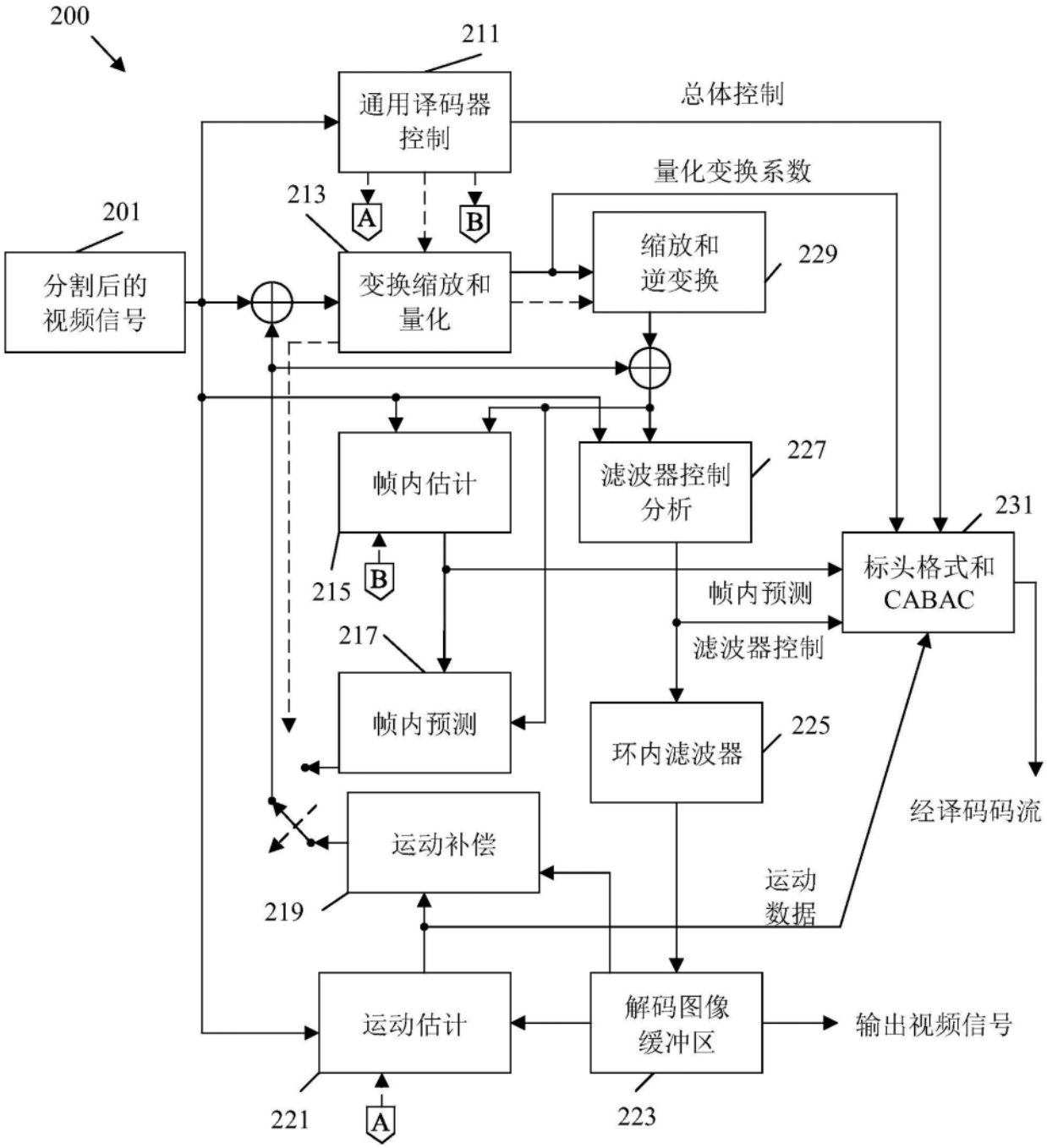

200
211
通用译码器控制
总体控制
量化变换系数
213
A
B
201
分割后的视频信号
变换缩放和量化
缩放和逆变换
229
帧内估计
滤波器控制分析
227
231
标头格式和CABAC
215
B
帧内预测
滤波器控制
217
帧内预测
环内滤波器
225
经译码码流
运动补偿
219
运动数据
运动估计
解码图像缓冲区
输出视频信号
221
A
223

图2

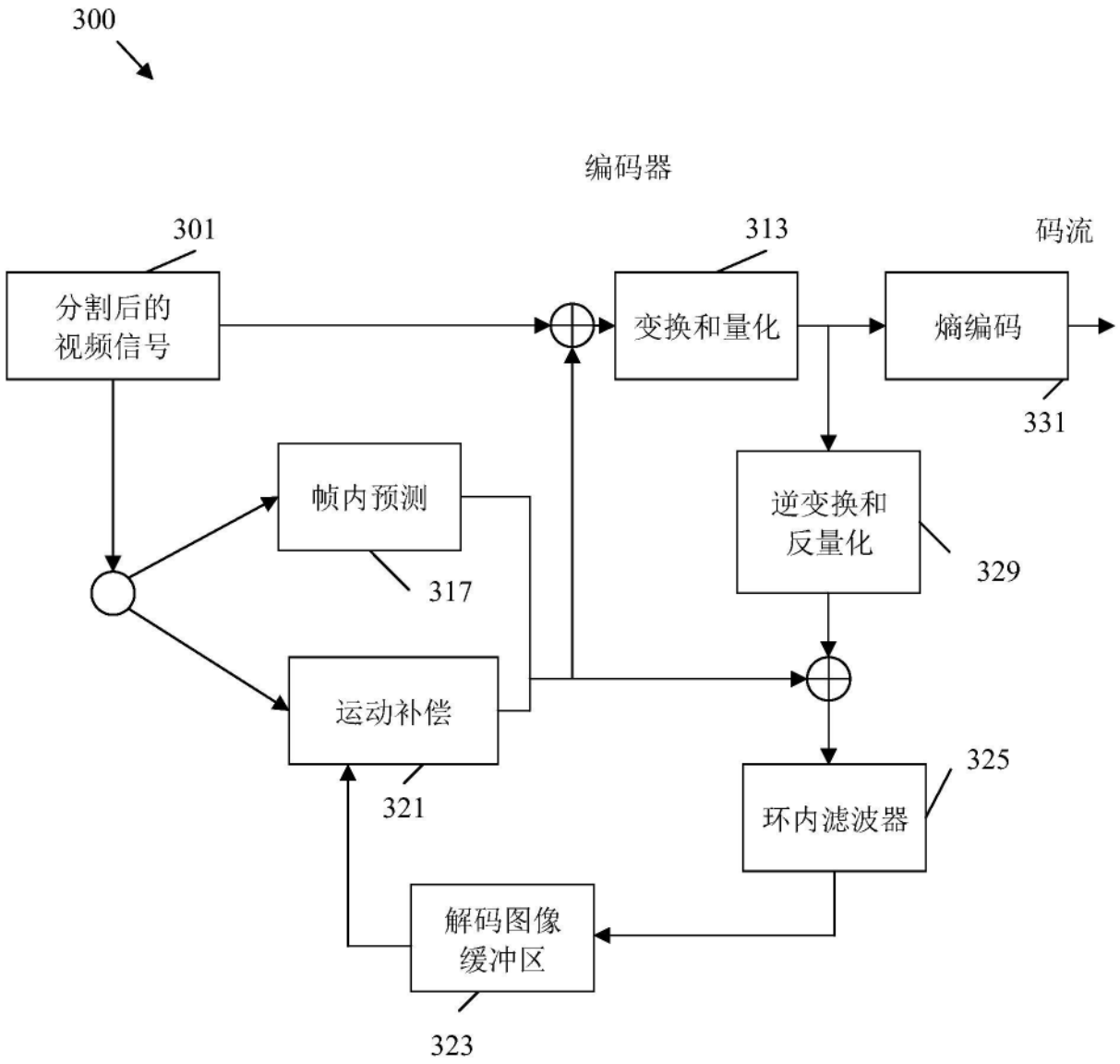

300
编码器
301
分割后的
视频信号
313
变换和量化
码流
熵编码
331
帧内预测
317
逆变换和
反量化
329
运动补偿
321
325
环内滤波器
解码图像
缓冲区
323

图3

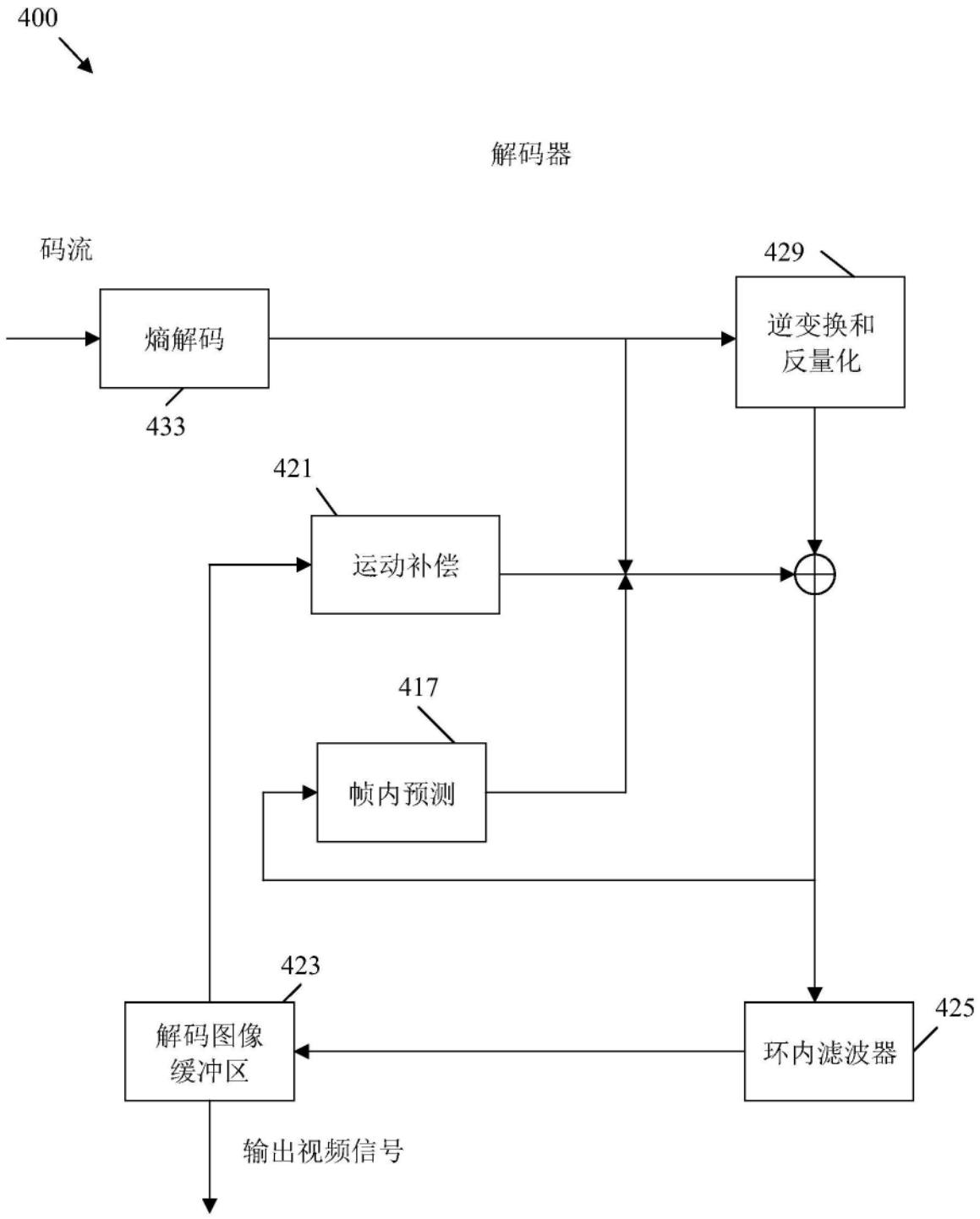

400
解码器
码流
熵解码
433
逆变换和
反量化
429
421
运动补偿
417
帧内预测
423
解码图像
缓冲区
环内滤波器
425
输出视频信号

图4

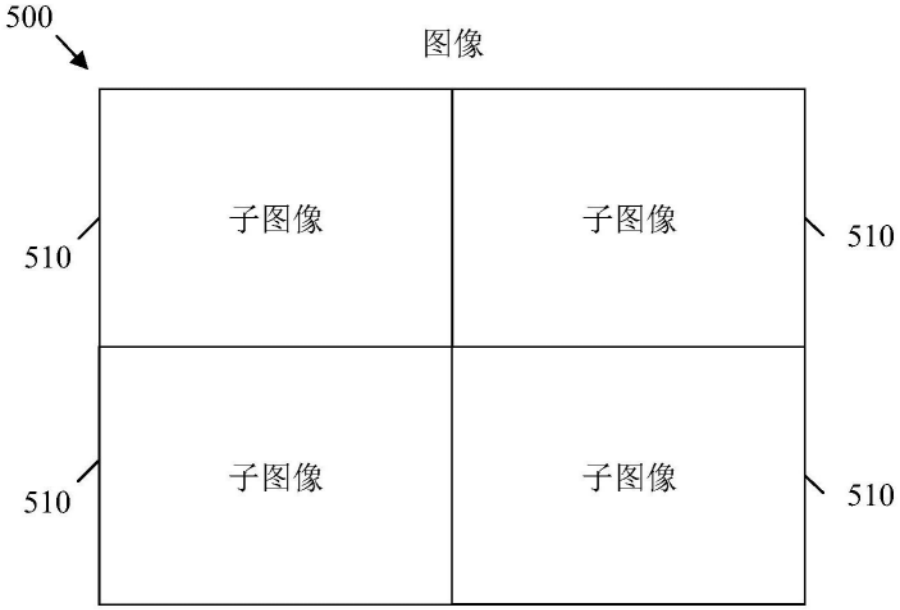

500
图像
子图像
子图像
子图像
子图像
510
510
510
510

图5A

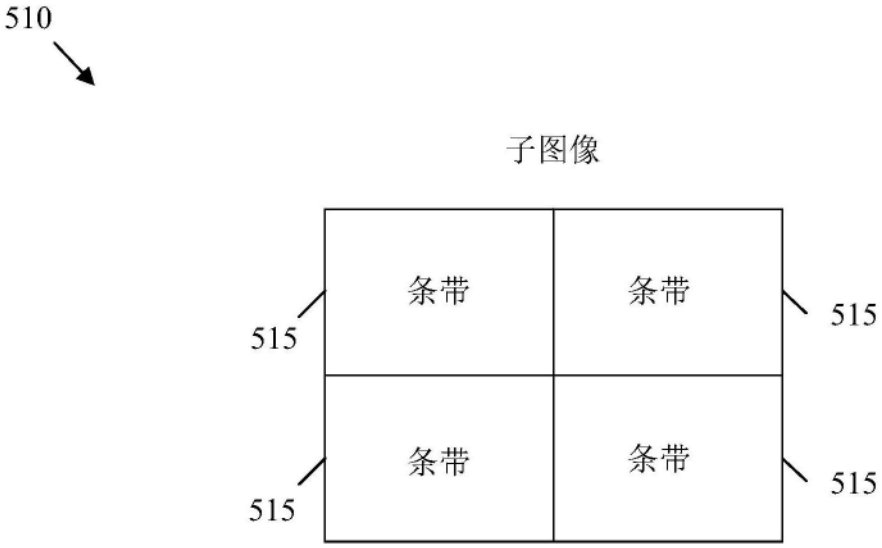

510
子图像
条带
条带
条带
条带
515
515
515
515

图5B

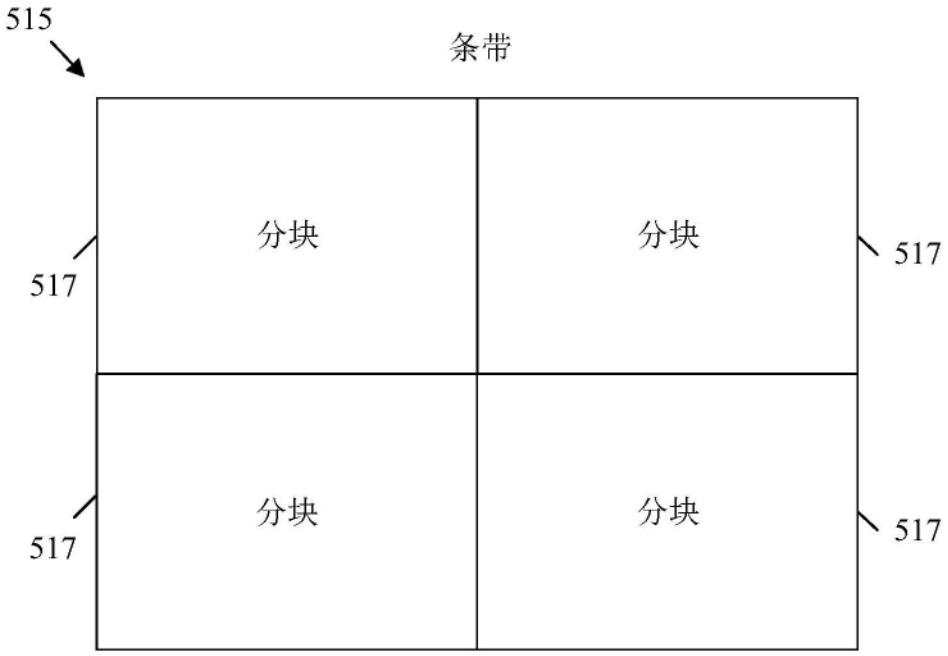


图5C

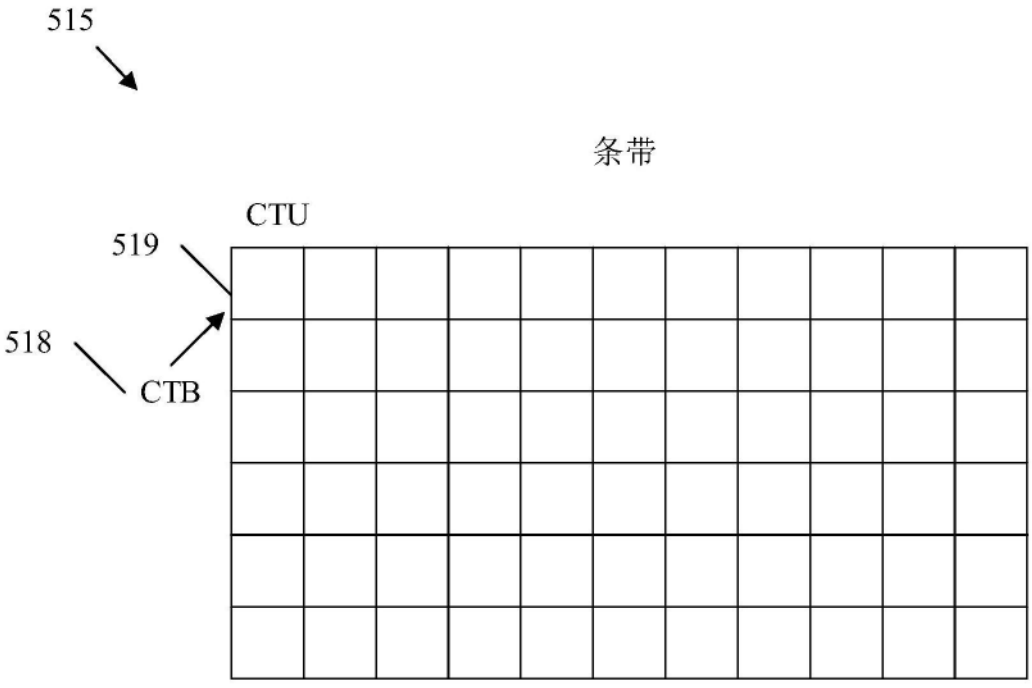


图5D

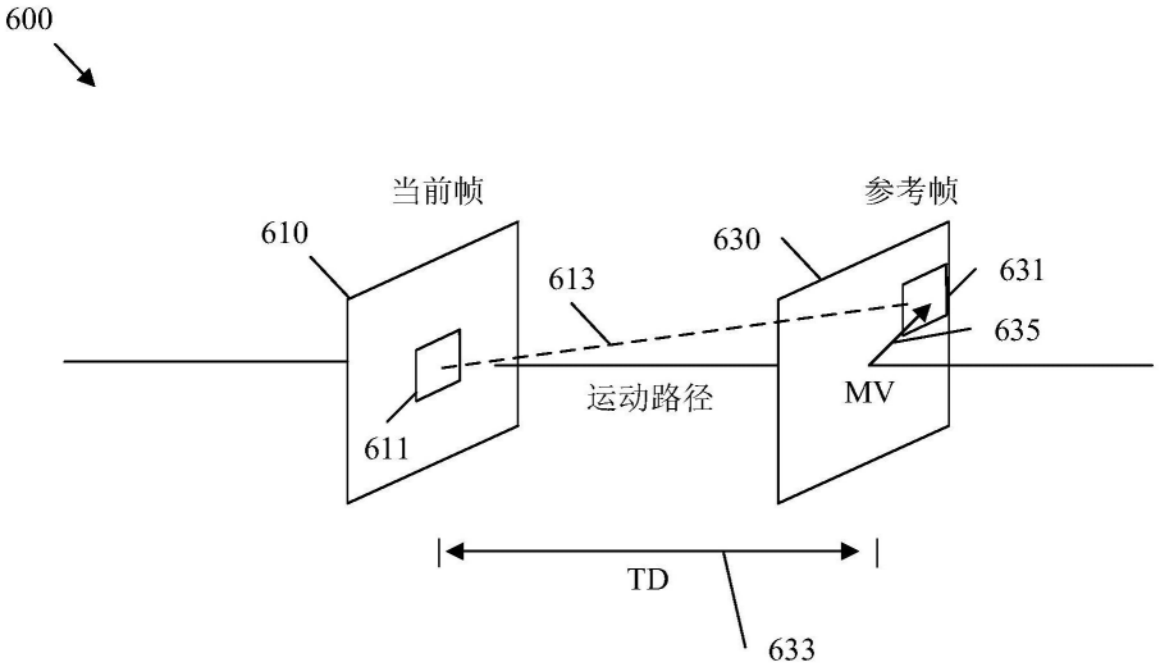

600
当前帧
参考帧
610
630
613
631
635
611
运动路径
MV
TD
633

图6

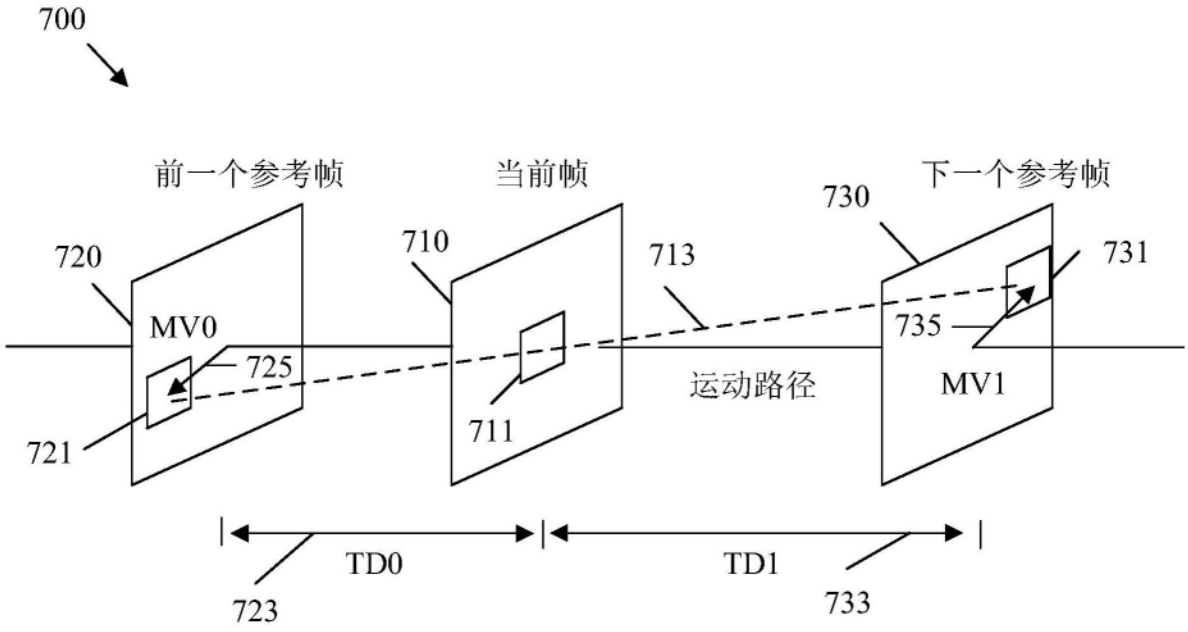

700
前一个参考帧
当前帧
下一个参考帧
720
710
730
713
731
MV0
735
725
运动路径
MV1
721
711
TD0
TD1
723
733

图7

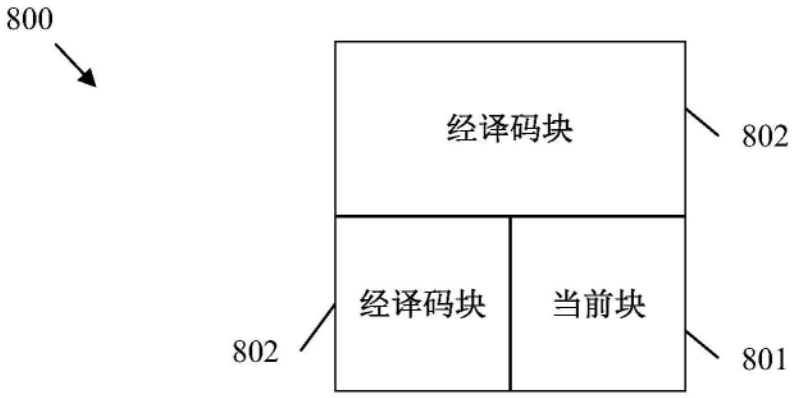

800
经译码块
802
经译码块
当前块
802
801

图8

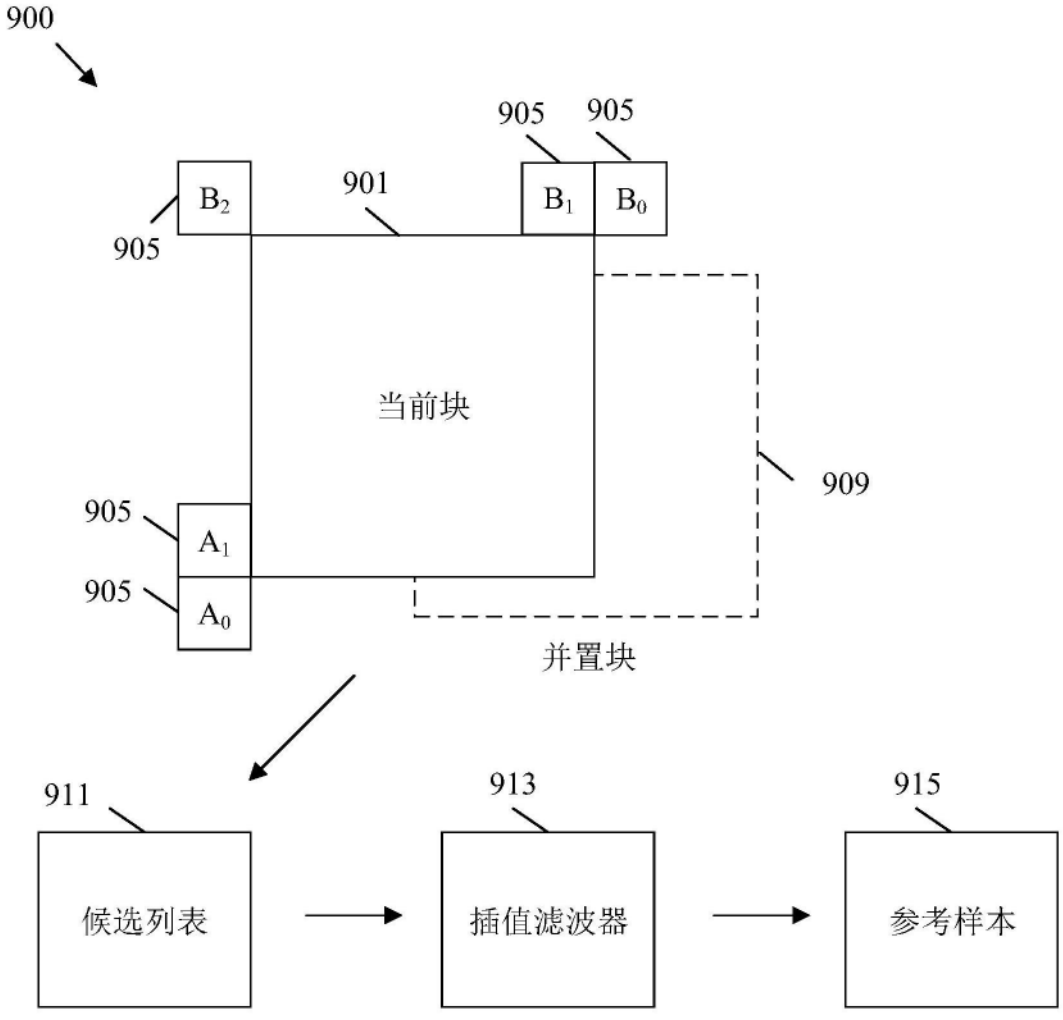

900
905
905
B2
901
B1
B0
905
当前块
909
905
A1
905
A0
并置块
911
913
915
候选列表
插值滤波器
参考样本

图9

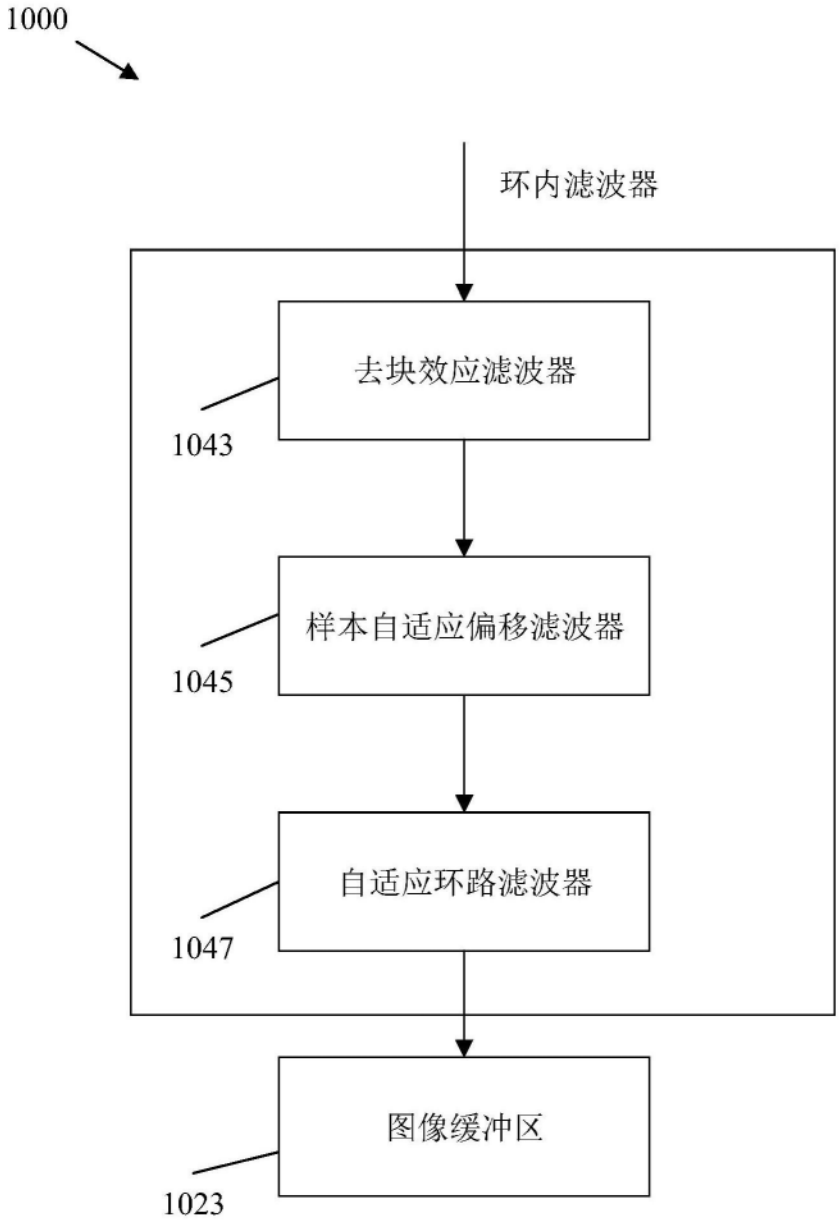

1000
环内滤波器
去块效应滤波器
1043
样本自适应偏移滤波器
1045
自适应环路滤波器
1047
图像缓冲区
1023

图10

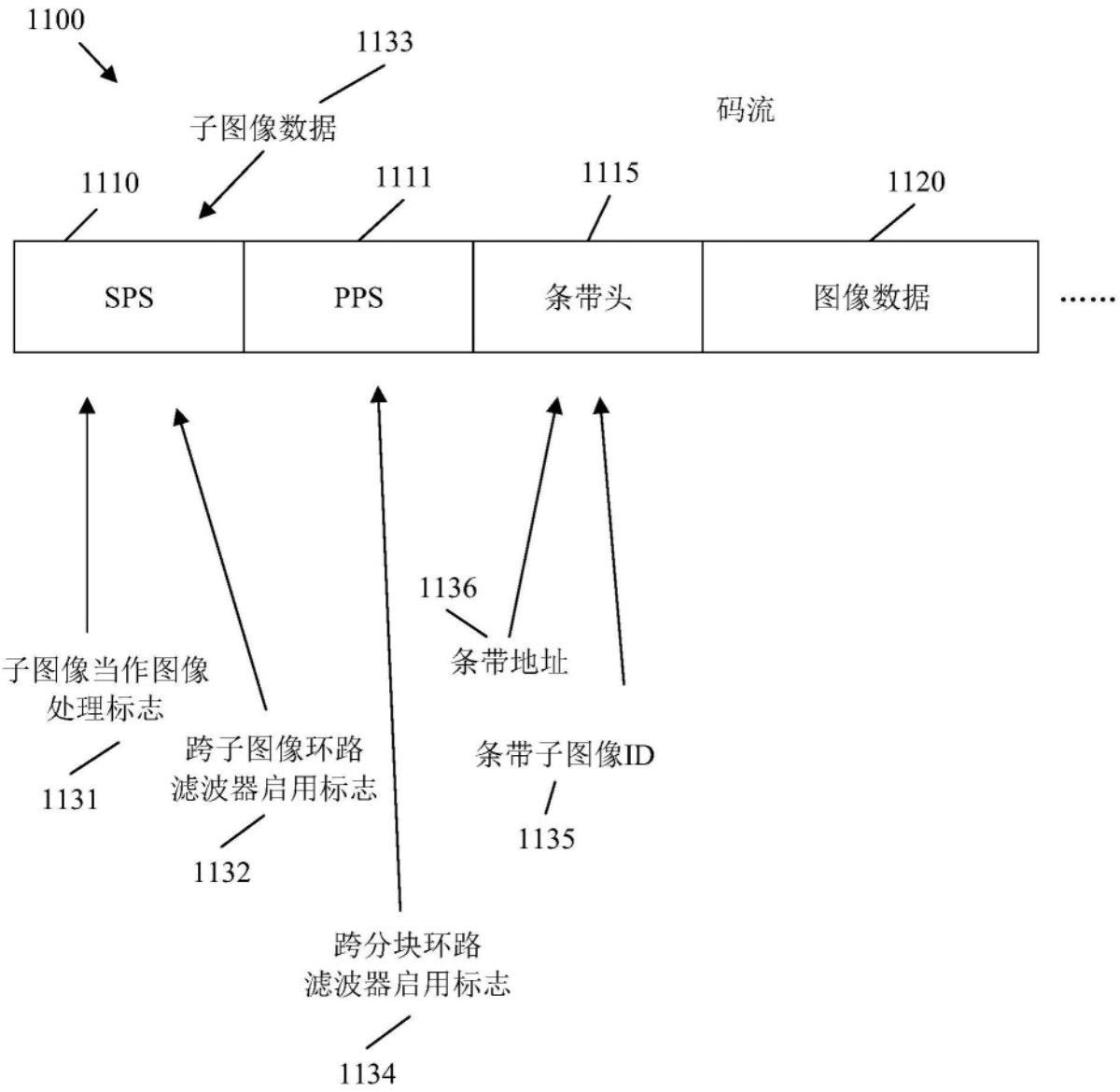

1100
1133
子图像数据
码流
1110
1111
1115
1120
SPS
PPS
条带头
图像数据
……
1136
条带地址
子图像当作图像
处理标志
1131
跨子图像环路
滤波器启用标志
1132
条带子图像ID
1135
跨分块环路
滤波器启用标志
1134

图11

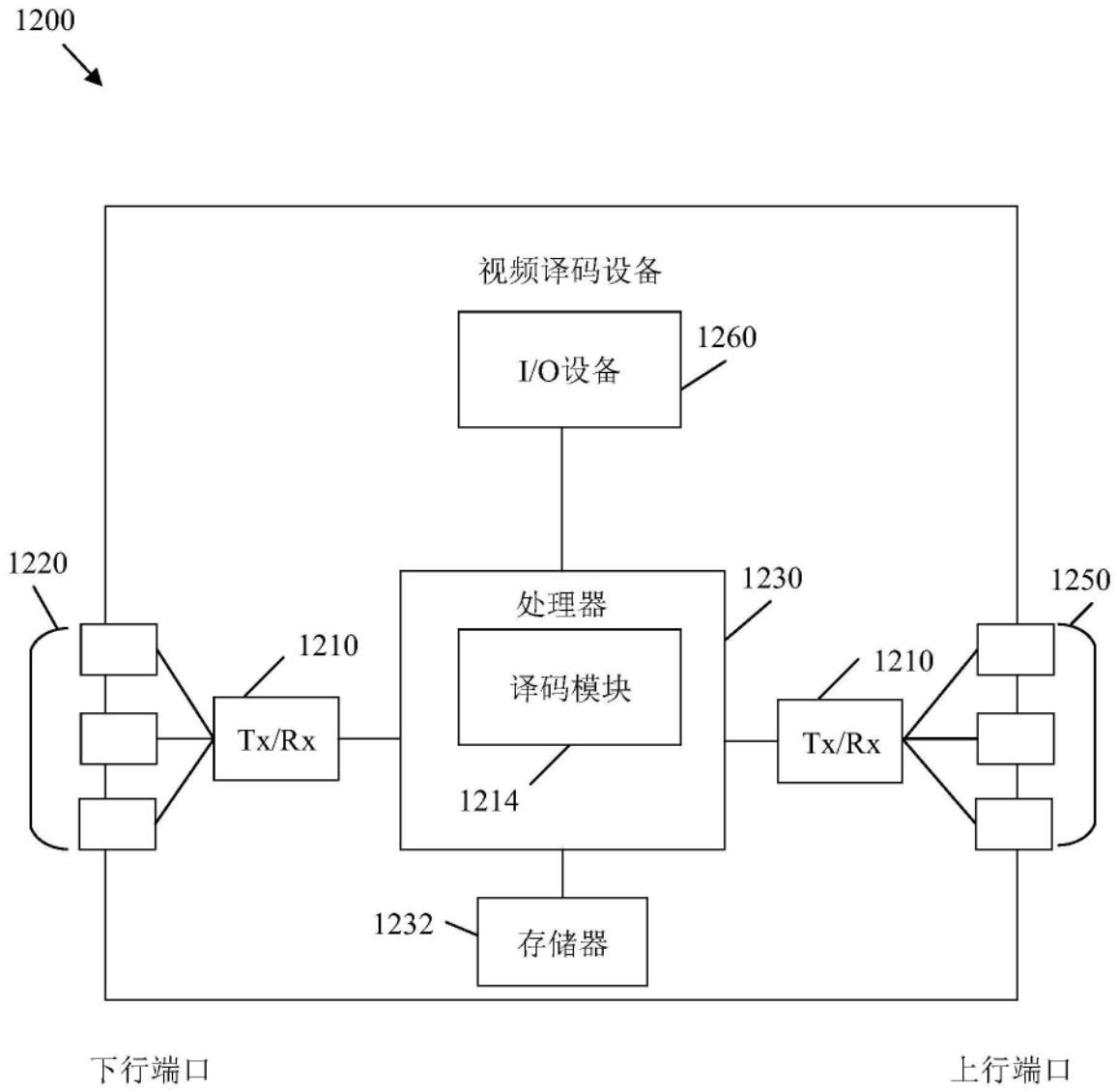

1200
视频译码设备
I/O设备
1260
1220
处理器
1230
1250
1210
译码模块
1210
Tx/Rx
Tx/Rx
1214
1232
存储器
下行端口
上行端口

图12

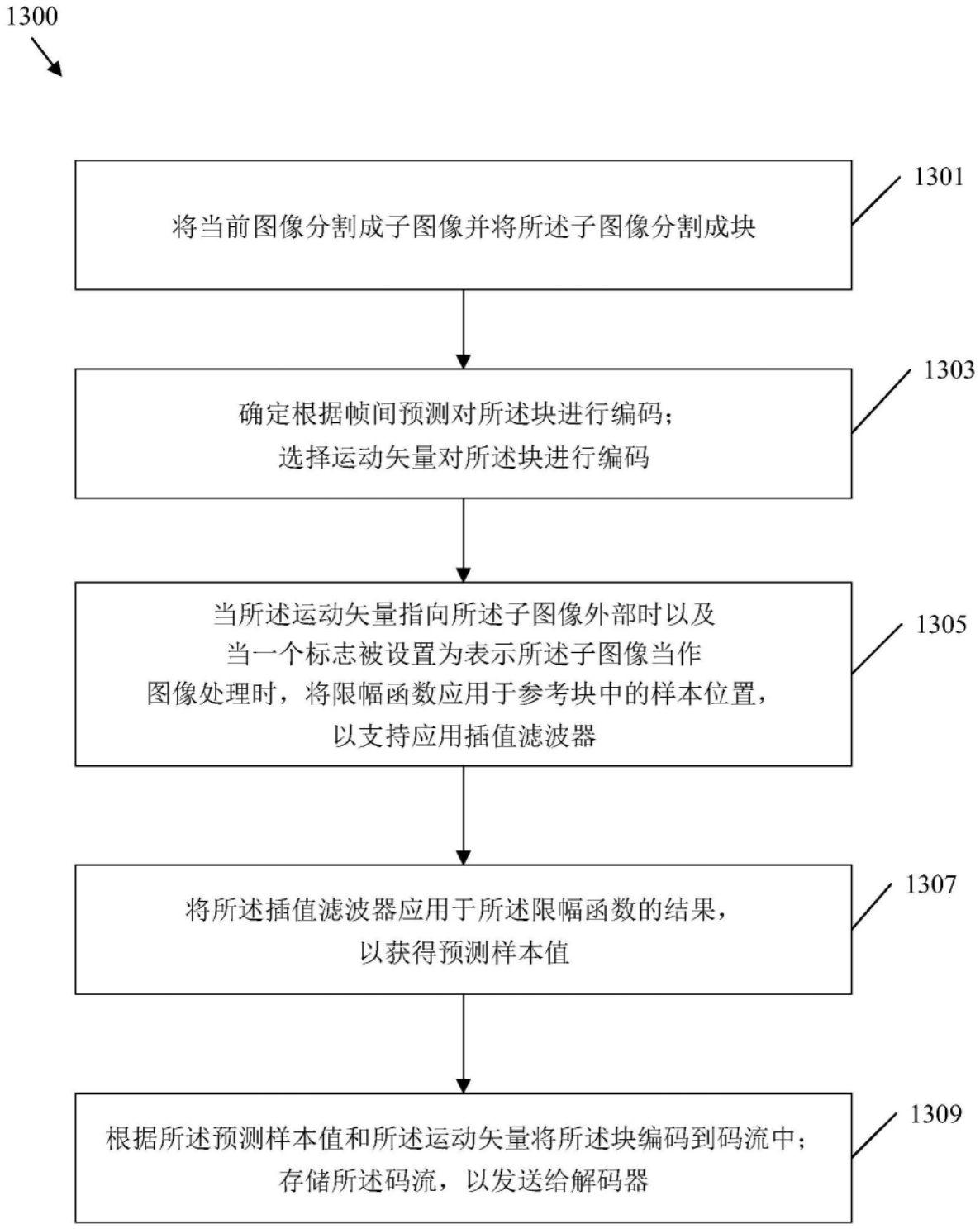

1300
将当前图像分割成子图像并将所述子图像分割成块
1301
确定根据帧间预测对所述块进行编码；
选择运动矢量对所述块进行编码
1303
当所述运动矢量指向所述子图像外部时以及
当一个标志被设置为表示所述子图像当作
图像处理时，将限幅函数应用于参考块中的样本位置，
以支持应用插值滤波器
1305
将所述插值滤波器应用于所述限幅函数的结果，
以获得预测样本值
1307
根据所述预测样本值和所述运动矢量将所述块编码到码流中；
存储所述码流，以发送给解码器
1309

图13

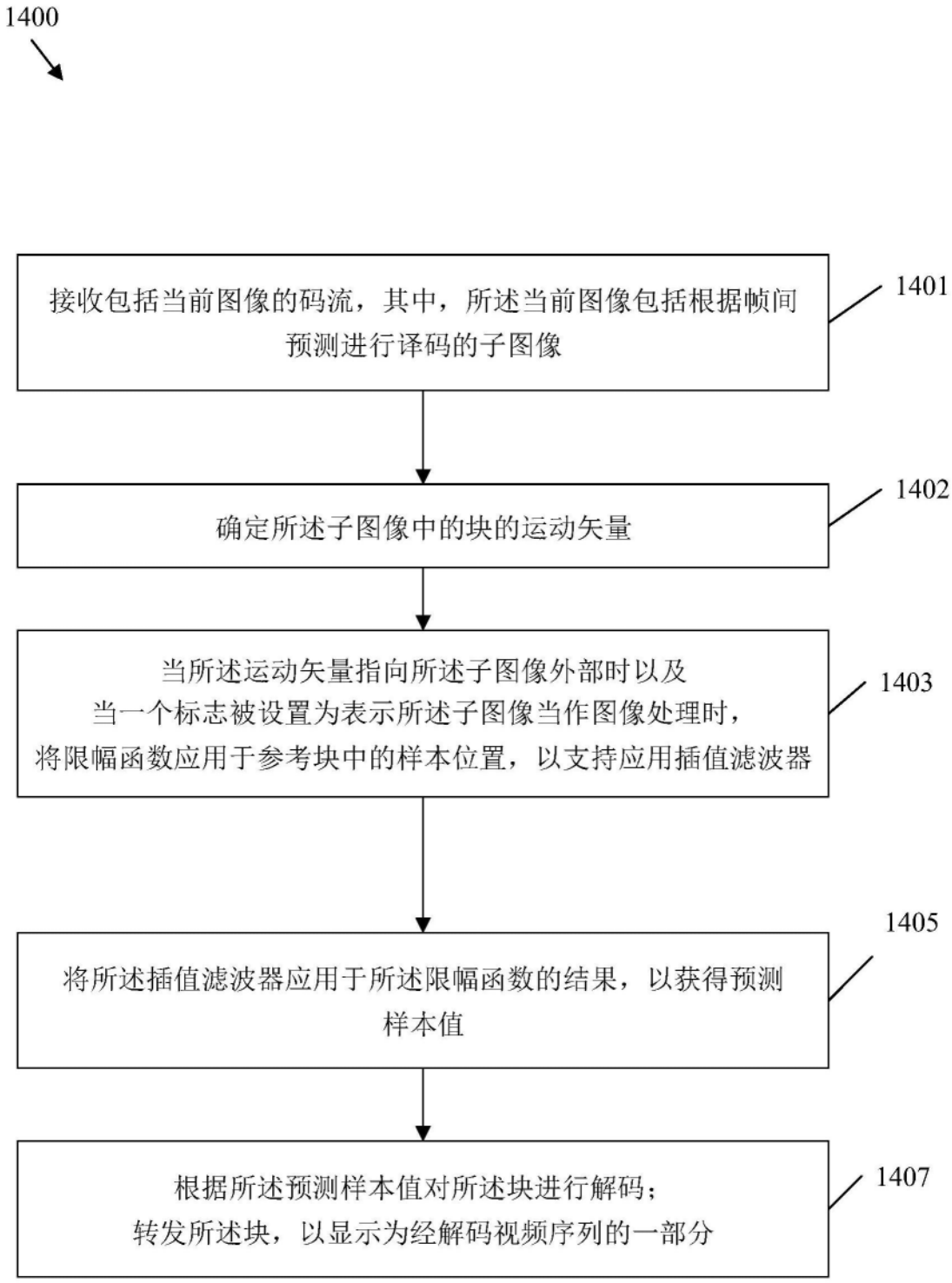

1400
接收包括当前图像的码流，其中，所述当前图像包括根据帧间预测进行译码的子图像
1401
确定所述子图像中的块的运动矢量
1402
当所述运动矢量指向所述子图像外部时以及
当一个标志被设置为表示所述子图像当作图像处理时，
将限幅函数应用于参考块中的样本位置，以支持应用插值滤波器
1403
将所述插值滤波器应用于所述限幅函数的结果，以获得预测样本值
1405
根据所述预测样本值对所述块进行解码；
转发所述块，以显示为经解码视频序列的一部分
1407

图14

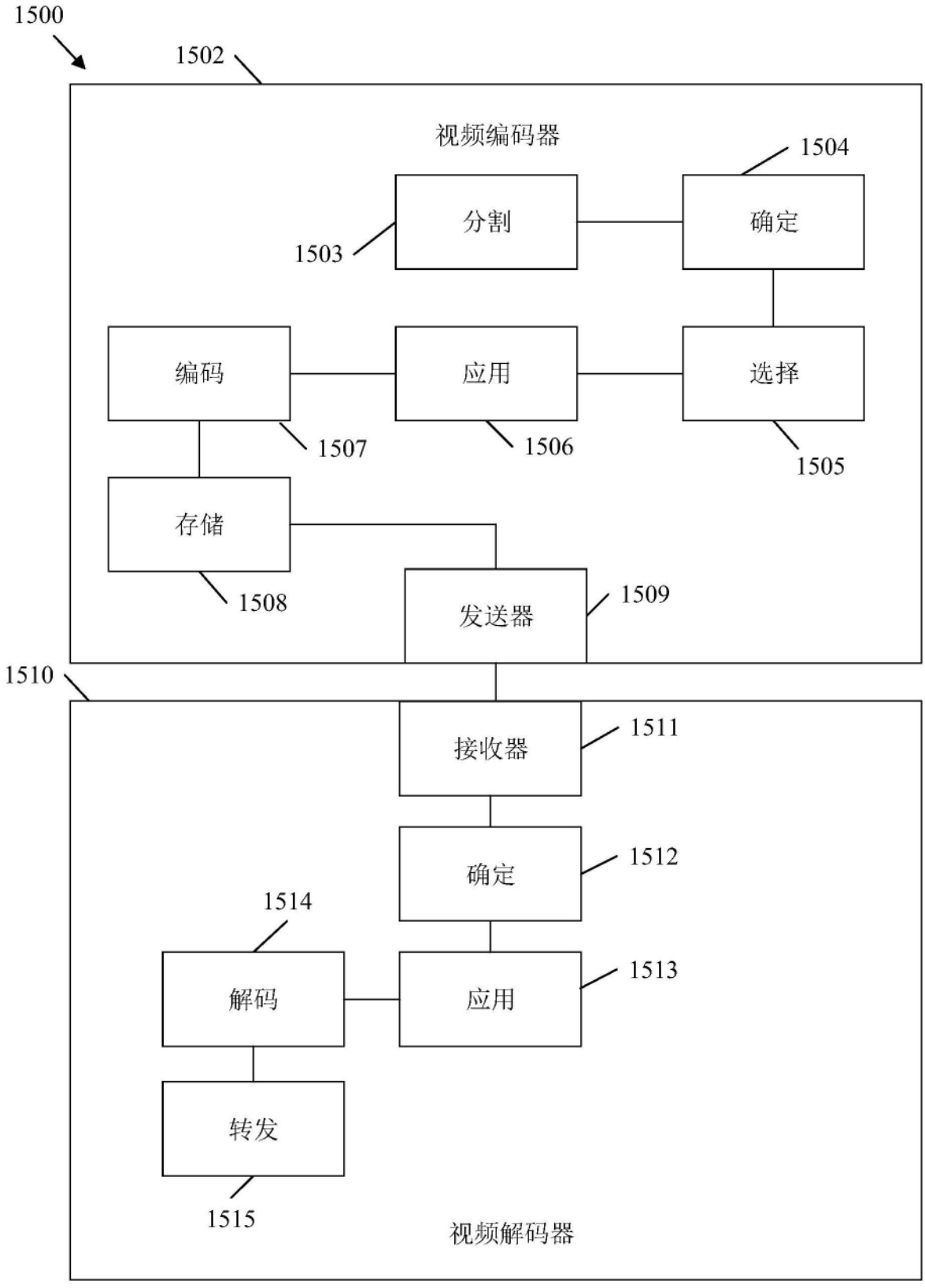

1500
1502
视频编码器
1504
分割
确定
1503
编码
应用
选择
1507
1506
1505
存储
1508
发送器
1509
1510
接收器
1511
确定
1512
1514
解码
应用
1513
转发
1515
视频解码器

图15