(19)国家知识产权局

(12)发明专利申请

(10)申请公布号 CN 119545015 A
(43)申请公布日 2025.02.28

(21)申请号 202411702009.7

(22)申请日 2020.06.24

(30)优先权数据
10-2019-0075163 2019.06.24 KR
10-2019-0080749 2019.07.04 KR

(62)分案原申请数据
202080046562.3 2020.06.24

(71)申请人 现代自动车株式会社
地址 韩国首尔
申请人 起亚株式会社
梨花女子大学校产学协力团

(72)发明人 姜制远 朴相孝 朴胜煜 林和平

(74)专利代理机构 北京戈程知识产权代理有限公司 11314
专利代理师 程伟 甄雁翔

(51)Int.Cl.
H04N 19/42(2014.01)
H04N 19/61(2014.01)
H04N 19/159(2014.01)
H04N 19/105(2014.01)
H04N 19/11(2014.01)
H04N 19/44(2014.01)
H04N 19/18(2014.01)
H04N 19/122(2014.01)
H04N 19/186(2014.01)
H04N 19/593(2014.01)
H04N 19/70(2014.01)
H04N 19/132(2014.01)
H04N 19/176(2014.01)

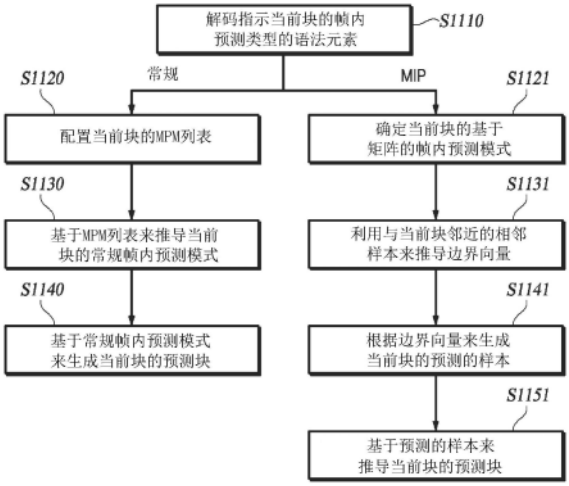
权利要求书3页 说明书23页 附图12页

(54)发明名称

用于视频数据的帧内预测编码的方法和装置

(57)摘要

本发明涉及用于视频数据的帧内预测编码的方法和装置。根据本发明的一个方面,用于对视频数据解码的装置包括解码单元和帧内预测单元,所述解码单元用于从比特流解码指示视频数据的当前块的帧内预测类型的语法元素,所述帧内预测单元用于通过基于由语法元素指示的当前块的帧内预测类型选择性地执行基于矩阵的帧内预测或常规帧内预测来生成当前块的预测块。



CN 119545015 A

1.一种对视频数据解码的方法,包括:

从比特流解码指示视频数据的当前块的帧内预测类型的语法元素,所述帧内预测类型由基于矩阵的帧内预测和常规帧内预测指示;

通过基于由语法元素指示的当前块的帧内预测类型选择性地执行基于矩阵的帧内预测或常规帧内预测来生成当前块的预测块;

从比特流重构变换系数;

通过对变换系数进行逆变换来生成当前块的残差块;

基于预测块和残差块来重构当前块,

其中,通过执行常规帧内预测来生成当前块的预测块包括:

基于与当前块邻近的相邻块的每个的常规帧内预测模式来推导最可能模式候选,并且配置当前块的最可能模式列表;以及

基于最可能模式列表来推导当前块的常规帧内预测模式,

其中,响应于确定出相邻块的帧内预测类型是基于矩阵的帧内预测,将相邻块的常规帧内预测模式设置为平面模式,

其中,通过执行基于矩阵的帧内预测来生成当前块的预测块包括:

从比特流解码指示当前块的基于矩阵的帧内预测模式的语法元素;

基于当前块的宽度和高度,利用与当前块邻近的相邻样本来推导边界向量;

基于边界向量与为基于矩阵的帧内预测模式预定义的矩阵之间的矩阵向量乘法来生成当前块的预测的样本;以及

基于预测的样本来推导当前块的预测块,

其中,利用与当前块邻近的相邻样本来推导边界向量包括:

从比特流解码语法元素,所述语法元素指示了从与当前块邻近的左侧相邻样本推导的边界向量的第一项和从与当前块邻近的上方相邻样本推导的边界向量的第二项的级联顺序;以及

根据级联顺序来级联第一项和第二项,从而生成边界向量。

2.根据权利要求1所述的方法,其中,指示当前块的基于矩阵的帧内预测模式的语法元素是截断的二进制码,所述截断的二进制码指定了对于当前块的宽度和高度所允许的多个基于矩阵的帧内预测模式的一个。

3.根据权利要求1所述的方法,其中,根据当前块的宽度和高度,用从与当前块邻近的左侧相邻样本下采样的值来填充边界向量的项,或者用与当前块邻近的左侧相邻样本来填充边界向量的项。

4.根据权利要求1所述的方法,其中,基于预测的样本来推导当前块的预测块包括:

将预测的样本分配至预测块中的位置;以及

利用对预测的样本、与当前块邻近的左侧相邻样本以及与当前块邻近的上方相邻样本的线性插值,针对预测块中未分配预测的样本的位置来生成预测的样本值。

5.根据权利要求1所述的方法,其中,当前块是包括亮度分量的亮度块。

6.根据权利要求5所述的方法,其中,当对亮度块执行基于矩阵的帧内预测,并且将亮度块的帧内预测模式用作对应于亮度块的色度块的帧内预测模式时,将用于亮度块的常规帧内预测模式设置为平面模式。

7.一种编码视频数据的方法,包括:

将指示视频数据的当前块的帧内预测类型的语法元素编码到比特流中,所述帧内预测类型由基于矩阵的帧内预测和常规帧内预测指示;

通过基于当前块的帧内预测类型选择性地执行基于矩阵的帧内预测或常规帧内预测来生成当前块的预测块;

利用预测块来生成当前块的残差块;

对残差块进行变换以生成变换系数;

对关于变换系数的信息进行编码,

其中,通过执行常规帧内预测来生成当前块的预测块包括:

确定当前块的常规帧内预测模式;

基于与当前块邻近的相邻块的每个的常规帧内预测模式来推导最可能模式候选,以配置当前块的最可能模式列表,其中,当相邻块的帧内预测类型是基于矩阵的帧内预测时,相邻块的常规帧内预测模式设置为平面模式;以及

基于最可能模式列表将指示当前块的常规帧内预测模式的至少一个语法元素编码到比特流中,以及

其中,通过执行基于矩阵的帧内预测来生成当前块的预测块包括:

将指示当前块的基于矩阵的帧内预测模式的语法元素编码到比特流中;

基于当前块的宽度和高度,利用与当前块邻近的相邻样本来推导边界向量;

基于边界向量与为基于矩阵的帧内预测模式预定义的矩阵之间的矩阵向量乘法来生成当前块的预测的样本;以及

基于预测的样本来推导当前块的预测块,

其中,利用与当前块邻近的相邻样本来推导边界向量包括:

将语法元素编码到比特流中,所述语法元素指示了从与当前块邻近的左侧相邻样本推导的边界向量的第一项和从与当前块邻近的上方相邻样本推导的边界向量的第二项的级联顺序;以及根据级联顺序来级联第一项和第二项,从而生成边界向量。

8.一种发送包括对于视频数据的块的编码的数据的比特流的方法,所述方法包括:

基于编码方法通过利用帧内预测对当前块进行编码来生成比特流;

向视频解码装置发送比特流,

其中,基于帧内预测的编码方法包括:

将指示视频数据的当前块的帧内预测类型的语法元素编码到比特流中,所述帧内预测类型由基于矩阵的帧内预测和常规帧内预测指示;

通过基于当前块的帧内预测类型选择性地执行基于矩阵的帧内预测或常规帧内预测来生成当前块的预测块;

利用预测块来生成当前块的残差块;

对残差块进行变换以生成变换系数;

对关于变换系数的信息进行编码,

其中,通过执行常规帧内预测来生成当前块的预测块包括:

确定当前块的常规帧内预测模式;

基于与当前块邻近的相邻块的每个的常规帧内预测模式来推导最可能模式候选,以配

置当前块的最可能模式列表,其中,当相邻块的帧内预测类型是基于矩阵的帧内预测时,相邻块的常规帧内预测模式设置为平面模式;以及

基于最可能模式列表将指示当前块的常规帧内预测模式的至少一个语法元素编码到比特流中,以及

其中,通过执行基于矩阵的帧内预测来生成当前块的预测块包括:

将指示当前块的基于矩阵的帧内预测模式的语法元素编码到比特流中;

基于当前块的宽度和高度,利用与当前块邻近的相邻样本来推导边界向量;

基于边界向量与为基于矩阵的帧内预测模式预定义的矩阵之间的矩阵向量乘法来生成当前块的预测的样本;以及

基于预测的样本来推导当前块的预测块,

其中,利用与当前块邻近的相邻样本来推导边界向量包括:

将语法元素编码到比特流中,所述语法元素指示了从与当前块邻近的左侧相邻样本推导的边界向量的第一项和从与当前块邻近的上方相邻样本推导的边界向量的第二项的级联顺序;以及

根据级联顺序来级联第一项和第二项,从而生成边界向量。

用于视频数据的帧内预测编码的方法和装置

[0001] 本申请是中国专利申请号为202080046562.3,发明名称为“用于视频数据的帧内预测编码的方法和装置”,申请日为2020年6月24日的进入中国的PCT专利申请的分案申请。

[0002] 相关申请的交叉引用

[0003] 本申请要求2019年6月24日在韩国提交的专利申请No.10-2019-0075163和2019年7月4日在韩国提交的专利申请No.10-2019-0080749的优先权,其全部内容通过引用结合于本文中。

技术领域

[0004] 本发明涉及视频数据的编码和解码。

背景技术

[0005] 由于视频数据量大于语音数据量或静止图像数据量,因此在不进行压缩处理的情况下存储或传输视频数据需要大量的硬件资源(包括存储器)。相应地,在存储或传输视频数据时,通常利用编码器来压缩视频数据以进行存储或传输。

[0006] 然后,解码器接收压缩的视频数据,解压并再现视频数据。用于这种视频的压缩技术包括H.264/AVC和高效率视频编码(High Efficiency Video Coding,HEVC),所述高效率视频编码(HEVC)比H.264/AVC的编码效率提高了大约40%。

[0007] 然而,图像大小、分辨率和帧速率逐渐增加,并且相应地,要编码的数据量也在增多。因此,需要一种与现有压缩技术相比具有更好的编码效率和更高的影像质量的新压缩技术。

发明内容

[0008] 本发明提出了一种用于视频数据的块的帧内预测编码的改进技术。

[0009] 根据本发明的一个方面,提供了一种对视频数据解码的方法,其包括:从比特流解码指示视频数据的当前块的帧内预测类型的语法元素,所述帧内预测类型由基于矩阵的帧内预测(matrix based intra-prediction,MIP)和常规帧内预测来指示;以及通过基于由语法元素指示的当前块的帧内预测类型选择性地执行MIP或常规帧内预测来生成当前块的预测块。

[0010] 作为通过执行常规帧内预测来生成当前块的预测块的一部分,该方法包括:基于与当前块邻近的相邻块的每个的常规帧内预测模式来推导最可能模式(Most Probable Mode,MPM)候选,并且配置当前块的MPM列表;以及基于MPM列表来推导当前块的常规帧内预测模式。当相邻块的帧内预测类型是MIP时,将相邻块的常规帧内预测模式设置为平面(PLANAR)模式。

[0011] 作为通过执行常规帧内预测来生成当前块的预测块的一部分,该方法包括:从比特流解码指示当前块的MIP模式的语法元素;基于当前块的大小和形状,利用与当前块邻近的相邻样本来推导边界向量;基于边界向量与为MIP模式预定义的矩阵之间的矩阵向量乘

法来生成当前块的预测的样本;以及基于预测的样本来推导当前块的预测块。

[0012] 根据本发明的另一个方面,提供了一种用于对视频数据解码的装置,包括:解码器和帧内预测器,所述解码器配置为从比特流解码指示视频数据的当前块的帧内预测类型的语法元素,所述帧内预测类型由基于矩阵的帧内预测(MIP)和常规帧内预测指示;所述帧内预测器配置为通过基于由语法元素指示的当前块的帧内预测类型选择性地执行MIP或常规帧内预测来生成当前块的预测块。

[0013] 作为通过执行常规帧内预测来生成当前块的预测块的一部分,帧内预测器配置为:基于与当前块邻近的相邻块的每个的常规帧内预测模式来推导最可能模式(MPM)候选,并且配置当前块的MPM列表;以及基于MPM列表来推导当前块的常规帧内预测模式。在推导MPM候选时,当相邻块的帧内预测类型是MIP时,帧内预测器将相邻块的常规帧内预测模式设置为(视为)平面模式。

[0014] 作为通过执行常规帧内预测来生成当前块的预测块的一部分,帧内预测器可以:从比特流解码指示当前块的MIP模式的语法元素;基于当前块的大小和形状,利用与当前块邻近的相邻样本来推导边界向量;基于边界向量与为MIP模式预定义的矩阵之间的矩阵向量乘法来生成当前块的预测的样本;以及基于预测的样本来推导当前块的预测块。

附图说明

[0015] 图1是能够实现本发明的技术的视频编码装置的示例性框图。

[0016] 图2示例性地示出利用QTBTTT结构的块分区结构。

[0017] 图3a示例性地示出多个帧内预测模式。

[0018] 图3b示例性地示出包括宽角度帧内预测模式的多个帧内预测模式。

[0019] 图4是能够实现本发明的技术的视频解码装置的示例性框图。

[0020] 图5是示出可以在本发明的技术中使用的MIP技术的主要过程的概念图。

[0021] 图6是示出根据本发明实施方案的基于MIP来预测块的示意性过程的流程图。

[0022] 图7a至图7c是示出利用左侧相邻样本构建输入到矩阵向量乘法运算的边界向量的示例性方法的概念图。

[0023] 图8a至图8c是示出利用上方相邻样本构建输入到矩阵向量乘法运算的边界向量的示例性方法的概念图。

[0024] 图9示出左侧相邻样本、边界向量和由此预测的预测的样本。

[0025] 图10示出上方相邻样本、边界向量和由此预测的预测的样本。

[0026] 图11是示出根据本发明实施方案的对视频数据解码的方法的流程图。

具体实施方式

[0027] 在下文中,将参考所附附图对本发明的一些实施方案进行详细描述。应当注意,在将附图标记添加到各个附图中的组成元件时,尽管元件在不同的附图中示出,但是相同的附图标记也表示相同的元件。另外,在以下对本发明的描述中,将省略并入本文的已知功能和配置的详细描述,以避免模糊本发明的主题。

[0028] 图1是能够实现本发明的技术的视频编码装置的示例性框图。在下文中,将参考图1来描述视频编码装置以及该装置的元件。

[0029] 视频编码装置包括:图像分割器110、预测器120、减法器130、变换器140、量化器145、重排单元150、熵编码器155、逆量化器160、逆变换器165、加法器170、滤波单元180和存储器190。

[0030] 视频编码装置的每个元件可以以硬件或软件、或者硬件和软件的组合来实现。各个元件的功能可以实现为软件,并且微处理器可以实现为实施对应于各个元件的软件功能。

[0031] 一个视频包括多个图像。每个图像分割为多个区域,并且对每个区域执行编码。例如,一个图像分割为一个或更多个瓦片(tile)或/和切片(slice)。特别地,一个或更多个瓦片可以被定义为瓦片组。每个瓦片或切片分割为一个或更多个编码树单元(coding tree unit,CTU)。每个CTU通过树结构分割为一个或更多个编码单元(coding unit,CU)。应用于每个CU的信息被编码为CU的语法,并且共同应用于包括在一个CTU中的CU的信息被编码为CTU的语法。另外,共同应用于一个切片中的所有块的信息被编码为切片头的语法,而应用于构成图像的所有块的信息被编码在图像参数集(Picture Parameter Set,PPS)或图像头中。另外,由多个图像共同参考的信息被编码在序列参数集(Sequence Parameter Set,SPS)中。另外,由一个或更多个SPS共同参考的信息被编码在视频参数集(Video Parameter Set,VPS)中。共同应用于一个瓦片或瓦片组的信息可以被编码为瓦片头或瓦片组头的语法。

[0032] 图像分割器110配置为确定编码树单元(CTU)的大小。关于CTU的大小(CTU尺寸)的信息被编码为SPS或PPS的语法,并且被传输至视频解码装置。

[0033] 图像分割器110配置为将构成视频的每个图像分割为具有预定大小的多个CTU,然后利用树结构递归地分割CTU。在树结构中,叶节点用作编码单元(CU),所述编码单元(CU)是编码的基本单元。

[0034] 树结构可以是四叉树(QuadTree,QT)、二叉树(BinaryTree,BT)、三叉树(TernaryTree,TT)、或者由两个或更多个QT结构、BT结构和TT结构的组合形成的结构,所述四叉树(QT)即节点(或父节点)被分割为相同大小的四个从节点(或子节点),所述二叉树(BT)即节点被分割为两个从节点,所述三叉树(TT)即节点以1:2:1的比率被分割为三个从节点。例如,可以利用四叉树加二叉树(QuadTree plus BinaryTree,QTBT)结构,或者可以利用四叉树加二叉树三叉树(QuadTree plus BinaryTree TernaryTree,QTBTTT)结构。这里,BTTT可以统称为多类型树(multiple-type tree,MTT)。

[0035] 图2示例性地示出QTBTTT分割树结构。如图2所示,CTU可以首先分割为QT结构。可以重复QT分割,直到分割块的大小达到QT中允许的叶节点的最小块大小MinQTSize。由熵编码器155对指示QT结构的每个节点是否被分割为下层的四个节点的第一标志(QT_split_flag)进行编码,并将其用信号通知视频解码装置。当QT的叶节点小于BT中允许的根节点的最大块大小(MaxBTSize)时,可以进一步将其分割为一个或更多个BT结构或TT结构。BT结构和/或TT结构可以具有多个分割方向。例如,可以存在两个方向,即,水平地分割节点的块的方向以及竖直地分割块的方向。如图2所示,当MTT分割开始时,通过熵编码器155对指示节点是否被分割的第二标志(mtt_split_flag)、指示分割情况下的分割方向(竖直或水平)的标志、和/或指示分割类型(二叉或三叉)的标志进行编码,并将其用信号通知视频解码装置。替选地,在对指示每个节点是否被分割为下层的4个节点的第一标志(QT_split_flag)

进行编码之前,可以对指示节点是否被分割的CU分割标志(split_cu_flag)进行编码。当CU分割标志(split_cu_flag)的值指示出没有执行分割时,节点的块成为分割树结构中的叶节点,并用作编码单元(CU),编码单元(CU)是编码的基本单元。当CU分割标志(split_cu_flag)的值指示出执行分割时,视频编码装置开始以上述方式从第一标志起对标志进行编码。

[0036] 当利用QTBT作为树结构的另一个示例时,可以存在两种分割类型,即将块水平地分割为相同大小的两个块的类型(即,对称水平分割)和将块竖直地分割为相同大小的两个块的类型(即,对称竖直分割)。由熵编码器155对指示BT结构的每个节点是否被分割为下层的块的分割标志(split_flag)和指示分割类型的分割类型信息进行编码,并将其传输至视频解码装置。可以存在将节点的块分割为两个非对称块的额外类型。非对称分割类型可以包括以1:3的大小比率将块分割为两个矩形块的类型,或者对角线地分割节点的块的类型。

[0037] CU可以根据CTU的QTBT或QTBTTT分割而具有各种大小。在下文中,与要编码或解码的CU(即,QTBTTT的叶节点)相对应的块被称为“当前块”。在采用QTBTTT分割时,当前块的形状可以是正方形或矩形。

[0038] 预测器120对当前块进行预测以生成预测块。预测器120包括帧内预测器122和帧间预测器124。

[0039] 通常,图像中的每个当前块可以分别被预测地编码。通常,利用帧内预测技术(其利用来自包括当前块的图像的数据)或帧间预测技术(其利用在包括当前块的图像之前被编码的图像的数据)来执行当前块的预测。帧间预测包括单向预测和双向预测两者。

[0040] 帧内预测器122配置为利用在包括当前块的当前图像中位于当前块周围的像素(参考像素)来预测当前块中的像素。根据预测方向,存在多个帧内预测模式。例如,如图3a和图3b所示,多个帧内预测模式可以包括2种非方向模式和65种方向模式,所述2种非方向模式包括平面(planar)模式和DC模式。针对每种预测模式不同地定义要使用的相邻像素和等式。下表列出帧内预测模式编号及其名称。

[0041] 表1

[0042]

帧内预测模式	相关名称
0	INTRA_PLANAR
1	INTRA_DC
2…66	INTRA_ANGULAR2…INTRA_ANGULAR66

[0043] 为了对矩形形状的当前块进行有效的方向预测,可以额外地利用由图3b中的虚线箭头指示的方向模式(帧内预测模式67至80和-1至-14)。这些模式可以称为“宽角度帧内预测模式(wide angle intra-prediction modes)”。在图3b中,箭头指示用于预测的相应参考样本,而非指示预测方向。预测方向与由箭头指示的方向相反。宽角度帧内预测模式是在当前块具有矩形形状时在与特定方向模式相反的方向上执行预测而无需额外的比特传输的模式。具体地,在宽角度帧内预测模式中,可以基于矩形当前块的宽度与高度的比率来确定可用于当前块的一些宽角度帧内预测模式。例如,在当前块的矩形形状的高度小于其宽度时,可以利用角度小于大约45度的宽角度帧内预测模式(帧内预测模式67至80)。在当前块的矩形形状的高度大于其宽度时,可以利用角度大于大约-135度的宽角度帧内预测模式(帧内预测模式-1至-14)。

[0044] 帧内预测器122可以配置为确定对当前块进行编码时要使用的帧内预测模式。在一些示例中,帧内预测器122可以配置为利用若干帧内预测模式来对当前块进行编码,并且从测试的模式中选择要使用的适当的帧内预测模式。例如,帧内预测器122可以配置为利用若干测试的帧内预测模式的率失真(rate-distortion)分析来计算率失真值,并且可以配置为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。

[0045] 帧内预测器122配置为从多个帧内预测模式中选择一个帧内预测模式,并且利用根据选择的帧内预测模式所确定的相邻像素(参考像素)和等式来预测当前块。由熵编码器155对关于选择的帧内预测模式的信息进行编码,并将其传输至视频解码装置。

[0046] 另外,帧内预测器122可以配置为利用基于矩阵的帧内预测(Matrix based Intra-Prediction,MIP)来生成当前块的预测块,这将在后面描述。帧内预测器122配置为利用从在当前块的左侧重构的样本和在当前块上方重构的样本推导的边界向量、预定义矩阵以及偏移向量来生成当前块的预测块。

[0047] 另外,帧间预测器124配置为通过运动补偿来生成当前块的预测块。帧间预测器124配置为在比当前图像更早已被编码和解码的参考图像中搜索与当前块最相似的块,并且利用搜索到的块来生成当前块的预测块。然后,帧间预测器配置为生成与当前图像中的当前块和参考图像中的预测块之间的位移(displacement)相对应的运动矢量(motion vector)。通常,对亮度(luma)分量执行运动估计,并且基于亮度分量计算的运动矢量用于亮度分量和色度分量两者。由熵编码器155对包括关于参考图像的信息和有关用于预测当前块的运动矢量的信息的运动信息进行编码,并将其传输至视频解码装置。

[0048] 减法器130配置为将当前块减去由帧内预测器122或帧间预测器124生成的预测块以生成残差块。

[0049] 变换器140可以配置为将残差块分割为一个或更多个变换块,并且对一个或更多个变换块应用变换,从而将变换块的残差值从像素域变换到频域。在频域中,变换块被称为包含一个或更多个变换系数值的系数块。二维变换核可以用于变换,而一维变换核可以分别用于水平变换和竖直变换。变换核可以基于离散余弦变换(DCT)、离散正弦变换(DST)等。

[0050] 变换器140可以配置为利用残差块的整个大小作为变换单元来变换残差块中的残差信号。另外,变换器140可以配置为将残差块在水平方向或竖直方向上分区为两个子块,并且配置为仅变换两个子块的一个。相应地,变换块的大小可以与残差块的大小(进而预测块的大小)不同。非零残差样本值可能不存在,或者在未变换的子块中可能非常罕见。不用信号通知未变换子块的残差样本,并且未变换子块的残差样本可以被视频解码装置视为“0”。根据分区方向和分区比率可以存在多种分区类型。变换器140可以向熵编码器155提供关于残差块的编码模式(或变换模式)的信息(例如,指示是变换残差块还是变换残差子块的信息,指示选择为将残差块分区为子块的分区类型的信息,以及识别被执行变换的子块的信息)。熵编码器155可以配置为对关于残差块的编码模式(或变换模式)的信息进行编码。

[0051] 量化器145配置为对从变换器140输出的变换系数进行量化,并且将量化的变换系数输出到熵编码器155。对于一些块或帧,量化器145可以配置为不经变换直接对相关残差块进行量化。

[0052] 重排单元150可以配置为对量化的残差值的系数值进行重排。重排单元150可以配

置为通过系数扫描(coefficient scanning)来将2维系数阵列改变为1维系数序列。例如,重排单元150可以配置为利用锯齿形扫描(zig-zag scan)或对角线扫描(diagonal scan)从直流(DC)系数向高频区域中的系数对系数进行扫描,以输出1维系数序列。根据变换单元的大小和帧内预测模式,可以利用竖直扫描或水平扫描代替锯齿形扫描,所述竖直扫描即在列方向上对系数的二维阵列进行扫描,所述水平扫描即在行方向上对二维块形状的系数进行扫描。换句话说,可以根据变换单元的大小和帧内预测模式在锯齿形扫描、对角线扫描、竖直扫描和水平扫描中确定要利用的扫描模式。

[0053] 熵编码器155配置为利用诸如基于上下文的自适应二进制算术编码(Context-based Adaptive Binary Arithmetic Code,CABAC)和指数哥伦布(exponential Golomb)的各种编码技术来对从重排单元150输出的一维量化的变换系数进行编码,以生成比特流。

[0054] 另外,熵编码器155配置为对与块分割相关的信息(例如,CTU大小、CU分割标志、QT分割标志、MTT分割类型和MTT分割方向)进行编码,使得视频解码装置可以以与视频编码装置相同的方式来分割块。另外,熵编码器155配置为对关于指示当前块是由帧内预测编码还是由帧间预测编码的预测类型的信息进行编码,并且根据预测类型来对帧内预测信息(即,关于帧内预测模式的信息)或帧间预测信息(即,关于参考图像索引和运动矢量的信息)进行编码。

[0055] 逆量化器160配置为对从量化器145输出的量化的变换系数进行逆量化,以生成变换系数。逆变换器165配置为将从逆量化器160输出的变换系数从频域变换到空域,并且重构残差块。

[0056] 加法器170配置为将重构的残差块和由预测器120生成的预测块相加,以重构当前块。重构的当前块中的像素在执行后续块的帧内预测时用作参考像素。

[0057] 滤波单元180配置为对重构的像素进行滤波,以减少由于基于块的预测和变换/量化而产生的块伪影(blocking artifacts)、振铃伪影(ringing artifacts)和模糊伪影(blurring artifacts)。滤波单元180可以包括去块滤波器182和样本自适应偏移(sample adaptive offset,SAO)滤波器184。

[0058] 去块滤波器180配置为对重构的块之间的边界进行滤波,以去除由逐块编码/解码而引起的块伪影,并且SAO滤波器184配置为对去块滤波的视频执行额外的滤波。SAO滤波器184是用于对由有损编码(lossy coding)引起的重构的像素与原始的像素之间的差进行补偿的滤波器。

[0059] 通过去块滤波器182和SAO滤波器184滤波的重构的块存储在存储器190中。一旦一个图像中的所有块被重构,重构的图像就可以用作后续要被编码的图像中的块的帧间预测的参考图像。

[0060] 图4是能够实现本发明的技术的视频解码装置的示例性功能框图。在下文中,将参考图4来描述视频解码装置和该装置的元件。

[0061] 视频解码装置可以包括:熵解码器410、重排单元415、逆量化器420、逆变换器430、预测器440、加法器450、滤波单元460和存储器470。

[0062] 类似于图1的视频编码装置,视频解码装置的每个元件可以用硬件、软件、或者硬件和软件的组合来实现。另外,每个元件的功能可以用软件来实现,并且微处理器可以实现为实施对应于每个元件的软件功能。

[0063] 熵解码器410配置为通过对由视频编码装置生成的比特流解码并提取与块分割相关的信息来确定要解码的当前块,并且提取重构当前块所需的预测信息和关于残差信号的信息等。

[0064] 熵解码器410配置为从序列参数集(SPS)或图像参数集(PPS)中提取关于CTU大小的信息,确定CTU的大小,并且将图像分割为确定大小的CTU。然后,解码器配置为将CTU确定为树结构的最高层(即,根节点),并且提取关于CTU的分割信息,以利用树结构来分割CTU。

[0065] 例如,当利用QTBTTT结构来分割CTU时,提取与QT的分割相关的第一标志(QT_split_flag),以将每个节点分割为子层的四个节点。对于与QT的叶节点相对应的节点,提取与MTT的分割有关的第二标志(MTT_split_flag)以及关于分割方向(竖直/水平)和/或分割类型(二叉/三叉)的信息,从而以MTT结构来分割对应的叶节点。由此,以BT或TT结构来递归地分割QT的叶节点下方的每个节点。

[0066] 作为另一个示例,当利用QTBTTT结构来分割CTU时,可以提取指示是否分割CU的CU分割标志(split_cu_flag)。当分割对应块时,可以提取第一标志(QT_split_flag)。在分割操作中,在零个或更多个递归QT分割之后,每个节点都可能发生零个或更多个递归MTT分割。例如,CTU可以直接经历MTT分割而不经历QT分割,或者仅经历QT分割多次。

[0067] 作为另一个示例,当利用QTBT结构来分割CTU时,提取与QT分割相关的第一标志(QT_split_flag),并且将每个节点分割为下层的四个节点。然后,提取指示是否以BT进一步分割与QT的叶节点相对应的节点的分割标志(split_flag)以及分割方向信息。

[0068] 一旦通过树结构分割来确定要解码的当前块,熵解码器410就配置为提取关于指示当前块是被帧内预测还是被帧间预测的预测类型的信息。当预测类型信息指示帧内预测时,熵解码器410配置为提取用于当前块的帧内预测信息(帧内预测模式)的语法元素。当预测类型信息指示帧间预测时,熵解码器410配置为提取用于帧间预测信息的语法元素,也就是说,指示运动矢量和由运动矢量参考的参考图像的信息。

[0069] 熵解码器410配置为从比特流提取关于残差块的编码模式的信息(例如,关于是对残差块编码还是仅对残差块的子块编码的信息,指示选择为将残差块分区为子块的分区类型的信息,识别被编码的残差子块的信息,量化参数等)。熵解码器410还配置为提取关于量化的当前块的变换系数的信息作为关于残差信号的信息。

[0070] 重排单元415可以配置为以由视频编码装置执行的系数扫描的相反顺序,将由熵解码器410进行熵解码的一维量化的变换系数的序列改变为2维系数阵列(即,块)。

[0071] 逆量化器420配置为对量化的变换系数进行逆量化。逆变换器430配置为基于关于残差块的编码模式的信息将逆量化的变换系数从频域逆变换到空域,以重构残差信号,从而生成重构的当前块的残差块。

[0072] 当关于残差块的编码模式的信息指示出当前块的残差块已经被视频编码装置编码时,逆变换器430利用当前块的大小(以及进而要重构的残差块的大小)作为逆量化的变换系数的变换单位,以执行逆变换,从而生成当前块的重构的残差块。

[0073] 当关于残差块的编码模式的信息指示出只有残差块的一个子块已经被视频编码装置编码时,逆变换器430利用变换的子块的大小作为逆量化的变换系数的变换单位,以执行逆变换,从而重构变换的子块的残差信号,并且用值“0”来填充未变换的子块的残差信号,以生成重构的当前块的残差块。

[0074] 预测器440可以包括帧内预测器442和帧间预测器444。在当前块的预测类型是帧内预测时,激活帧内预测器442,而在当前块的预测类型是帧间预测时,激活帧间预测器444。

[0075] 帧内预测器442配置为基于从熵解码器410提取的帧内预测模式的语法元素,在多个帧内预测模式中确定当前块的帧内预测模式,并且根据帧内预测模式,利用当前块周围的参考像素来预测当前块。另外,帧内预测器442可以配置为利用基于矩阵的帧内预测(MIP)来生成当前块的预测块,这将在后面描述。帧内预测器442可以配置为利用从在当前块的左侧重构的样本和在当前块的上方重构的样本所推导的边界向量、以及预定义矩阵和偏移向量来生成当前块的预测块。

[0076] 帧间预测器444配置为利用由熵解码器410提取的帧内预测模式的语法元素来确定当前块的运动矢量和由运动矢量参考的参考图像,并且基于运动矢量和参考图像来预测当前块。

[0077] 加法器450配置为通过将从逆变换器430输出的残差块与从帧间预测器444或帧内预测器442输出的预测块相加来重构当前块。在对后续要被解码的块进行帧内预测时,重构的当前块中的像素用作参考像素。

[0078] 滤波单元460可以包括去块滤波器462和SAO滤波器464。去块滤波器462对重构的块之间的边界进行去块滤波,以去除由逐块解码引起的块伪影。SAO滤波器464配置为在对相应偏移进行去块滤波之后对重构的块执行额外的滤波,以补偿由有损编码引起的重构的像素与原始的像素之间的差。通过去块滤波器462和SAO滤波器464滤波的重构的块存储在存储器470中。当一个图像中的所有块被重构时,重构的图像用作对后续要被编码的图像中的块进行帧间预测的参考图像。

[0079] 本发明的技术通常涉及帧内预测编码。下面的描述主要侧重于解码技术,也就是说,视频解码器的操作。由于编码技术与全面描述的解码技术相反,因此简要地描述编码技术。

[0080] 在讨论下一代视频编码标准(即,通用视频编码(Versatile Video Coding,VVC))时,引入了几种新的编码工具,这些编码工具能够实现比高效率视频编码(HEVC)更好的编码性能。多变换选择(Multiple Transform Selection,MTS)(其是在VVC中引入的技术)依赖于三种三角变换,包括DCT-2、DST-7和DCT-8,并且视频编码器可以配置为选择水平变换和竖直变换,这使率失真代价最小化。

[0081] 表2

[0082]

变换类型	基本函数 $T_i(j), i, j = 0, 1, \ldots, N-1$
DCT-II	$T_i(j) = \omega_0 \cdot \sqrt{\frac{2}{N}} \cdot \cos\left(\frac{\pi \cdot i \cdot (2j+1)}{2N}\right)$ 其中，$\omega_0 = \begin{cases} \sqrt{\frac{2}{N}} & i = 0 \\ 1 & i \neq 0 \end{cases}$
DCT-VIII	$T_i(j) = \sqrt{\frac{4}{2N+1}} \cdot \cos\left(\frac{\pi \cdot (2i+1) \cdot (2j+1)}{4N+2}\right)$
DST-VII	$T_i(j) = \sqrt{\frac{4}{2N+1}} \cdot \sin\left(\frac{\pi \cdot (2i+1) \cdot (j+1)}{2N+1}\right)$

[0083] 对于应用了MTS的块,可以对一个或更多个语法元素进行编码,以用信号通知在水平方向和竖直方向上分别使用的变换核。

[0084] 作为示例,三个标志(MTS_CU_flag、MTS_Hor_flag、MTS_Ver_flag)可以与诸如表3的预定义的映射表一起使用。

[0085] 表3

[0086]

MTS_CU_flag	MTS_Hor_flag	MTS_Ver_flag	帧内/帧间	
			水平	竖直
0	-	-	DCT2	
1	0	0	DST7	DST7
	0	1	DCT8	DST7
	1	0	DST7	DCT8
	1	1	DCT8	DCT8

[0087] 首先,可以用信号通知指示是否在水平方向和竖直方向两者上使用DCT-2的一个标志(MTS_CU_flag)。当在这两个方向上没有应用DCT-2时,可以用信号通知指示DST-7与DCT-8之间在水平方向上要应用的变换核的一个标志(MTS_Hor_flag),并且可以用信号通知指示DST-7与DCT-8之间在竖直方向上要应用的变换核的一个标志(MTS_Ver_flag)。DST-7和DCT-8可以用于达到32×32的块大小,而DCT-2可以用于达到64×64的块大小。相应地,当块的大小(宽度和高度)小于或等于32,并且块具有至少一个非零残差信号时,可以用信号通知MTS_CU_flag。当MTS_CU_flag不存在时,MTS_CU_flag可以被推断为0。

[0088] 替选地,指示MTS核索引的语法元素(tu_mts_idx)可以与诸如表4的预定义的映射表一起使用。另外,tu_mts_idx指示MTS核索引,并且根据每个索引值来选择用于水平方向和竖直方向的核。在表4中,trTypeHor和trTypeVer指示用于水平方向和竖直方向的核的类型。特别地,0表示DCT-2,1表示DST-7,2表示DCT-8。例如,当tu_mts_idx＝2时,在水平方向上使用DCT-8,而在竖直方向上使用DST-7。

[0089] 表4

[0090]

tu_mts_idx	0	1	2	3	4
trTypeHor	0	1	2	1	2
trTypeVer	0	1	1	2	2

[0091] 另外,只有当SPS中的sps_explicit_mts_enabled_flag为1时,用信号通知tu_mts_idx。这称为ExplicitMTS,这是因为显式地用信号通知MTS索引。另一方面,tu_mts_idx可能不会被显式地指示,而是通过编码器和解码器被隐式地推导,这称为ImplicitMTS。当SPS中的sps_explicit_mts_enabled_flag为0时,可以执行ImplicitMTS。当利用ImplicitMTS时,可以基于诸如用于块的帧内预测模式、帧间预测模式和块的大小的信息来执行变换核的选择。例如,当块的宽度大于或等于4且小于或等于16时,将DST-7用作水平方向的核。否则,利用DCT-2。另外,当块的高度大于或等于4且小于或等于16时,将DST-7用作垂直方向的核。否则,利用DCT-2。换句话说,当块大小较小时,利用DST-7,这是因为预测残差信号的方向性可能较高。对于相对较大的块,可以将DCT-2用作核,这是因为残差信号相当均匀的特性较强。

[0092] 表5示出VVC草案5中特定的变换单元语法的一部分。在下面的语法中,用灰色来突出显示元素旨在提供理解。

[0093] 表5

[0094]

transform_unit(x0, y0, tbWidth, tbHeight, treeType, subTuIndex) {	描述符
(省略)	

if(tu_cbf_luma[x0][y0] && treeType != DUAL_TREE_CHROMA
&& (tbWidth <= 32) && (tbHeight <= 32)
&& (IntraSubPartitionsSplit[x0][y0] == ISP_NO_SPLIT) && (!cu_sbt_flag)) {	

if(transform_skip_enabled_flag && tbWidth <= MaxTsSize && tbHeight <= MaxTsSize)	
transform_skip_flag[x0][y0]	ae(v)
if(((CuPredMode[x0][y0] != MODE_INTRA && sps_explicit_mts_inter_enabled_flag) || (CuPredMode[x0][y0] == MODE_INTRA && sps_explicit_mts_intra_enabled_flag)) && (!transform_skip_flag[x0][y0]))	
tu_mts_idx[x0][y0]	ae(v)
}	
if(tu_cbf_luma[x0][y0]) {	
if(!transform_skip_flag[x0][y0])	
residual_coding(x0, y0, Log2(tbWidth), Log2(tbHeight), 0)	
else	
residual_ts_coding(x0, y0, Log2(tbWidth), Log2(tbHeight), 0)	
}	
(省略)	

[0095] 可以在无需变换操作的情况下编码变换块,所述变换操作由对于具有至少一个非零系数的变换块用信号通知的语法元素transform_skip_flag指示。变换块的编码模式的高级别控制是通过transform_skip_enabled_flag和log2_transform_skip_max_size_

minus2来实现的,所述transform_skip_enabled_flag和log2_transform_skip_max_size_minus2是诸如序列参数集(SPS)的高级别语法(High-Level Syntax,HLS)中的元素。当transform_skip_enabled_flag为1时,对高度和宽度小于或等于“1<<(log2_transform_skip_max_size_minus2+2)”的变换块编码transform_skip_flag。当transform_skip_flag为1时,以变换跳过模式编码相关的变换块。否则,可以将基于MTS的变换应用于相关的变换块。

[0096] 基于矩阵的帧内预测(MIP)是在VTM5.0中引入的一种新的帧内预测技术。最初的想法是使用基于神经网络的帧内预测技术,也就是说,基于相邻的重构的像素,使用多层神经网络以预测当前PU像素值。然而,由于利用神经网络的预测方法的复杂性较高,引入了一种利用预训练矩阵基于仿射线性变换的帧内预测技术。

[0097] 为了预测具有宽度W和高度H的矩形块PU,MIP将块的左侧的H个重构的样本和块的上方的W个重构的样本作为输入。通过平均化、矩阵向量乘法、线性插值等来获得最终预测的像素。

[0098] 应用MIP的块的大小分为以下三类。

[0099] $$idx(W,H)=\begin{cases}0 & \text{对于 } W=H=4\\ 1 & \text{对于 } max(W,H)=8\\ 2 & \text{对于 } max(W,H)>8\end{cases}$$

[0100] 根据idx(W,H),MIP模式的数量(numModes)、边界大小(boundarySize)和预测块大小(predW、predH、predC)定义如下。在下表中,MipSizeId=idx(W,H)。

[0101] 表6

[0102]

MipSizeId	numModes	boundarySize	predW	predH	predC
0	35	2	4	4	4
1	19	4	4	4	4
2	11	4	Min(nTbW,8)	Min(nTbH,8)	8

[0103] 图5是示出可以在本发明的技术中使用的MIP技术的主要过程的概念图。

[0104] (1)平均化

[0105] 该过程的主要目的是使参考样本归一化。根据块大小和形状(即,MipSizeId),获得4个或8个样本。在当前块的宽度和高度两者均为4时(即,W=H=4),总共获得4个样本,包括左侧的2个和上方的2个(boundarySize=2)。在另一种情况下,总共获得8个样本,包括左侧的4个和上方的4个(boundarySize=4)。

[0106] 如图5所示,上方相邻样本用$bdry^{top}$表示,而左侧相邻样本用$bdry^{left}$表示。通过对$bdry^{top}$和$bdry^{left}$分别执行平均化,获得下采样的样本集$bdry_{red}^{top}$和$bdry_{red}^{left}$。平均化是如下的下采样过程。

[0107] $$\mathrm{redS}[\ x\ ]=\left(\sum_{i=0}^{bDwn-1}\mathrm{refS}[\ x*bDwn+i\ ]+(\ 1\ll(\ \log2(\ bDwn\ )-1\ ))\right)\gg \mathrm{Log2}(\ bDwn\ )$$

[0108] 在上面的等式中,bDwn表示下采样缩放值(nTbs/boundarySize),而refS表示原始参考样本。对于左侧相邻样本,将计算出的redS存储为$bdry_{red}^{left}$,而对于上方相邻样本,将

计算出的redS存储为$bdry_{red}^{top}$。

[0109] 将下采样的参考样本拼接为长度为4或8的向量。输入到向量矩阵乘法的归约的边界向量$bdry_{red}$定义为如下等式。例如,当W＝H＝4且MIP模式小于18时,通过以$bdry_{red}^{left}$和$bdry_{red}^{top}$的顺序拼接来构建边界向量。当W＝H＝4且MIP模式大于或等于18时,以$bdry_{red}^{left}$和$bdry_{red}^{top}$的顺序来拼接$bdry_{red}^{left}$和$bdry_{red}^{top}$。在下面的等式中,“模式”表示MIP模式。

[0110]
$$bdry_{red}=\begin{cases}[bdry_{red}^{top},bdry_{red}^{left}] & \text{对于}W=H=4\ \text{且}\ \text{模式}<18\\ [bdry_{red}^{left},bdry_{red}^{top}] & \text{对于}W=H=4\ \text{且}\ \text{模式}\geqslant 18\\ [bdry_{red}^{top},bdry_{red}^{left}] & \text{对于}\max(W,H)=8\ \text{且}\ \text{模式}<10\\ [bdry_{red}^{left},bdry_{red}^{top}] & \text{对于}\max(W,H)=8\ \text{且}\ \text{模式}\geqslant 10\\ [bdry_{red}^{top},bdry_{red}^{left}] & \text{对于}\max(W,H)>8\ \text{且}\ \text{模式}<6\\ [bdry_{red}^{left},bdry_{red}^{top}] & \text{对于}\max(W,H)>8\ \text{且}\ \text{模式}\geqslant 6\end{cases}$$

[0111] (2)矩阵向量乘法

[0112] 在此过程中,从归约的边界向量$bdry_{red}$生成当前块的下采样预测信号$pred_{red}$。$pred_{red}$是矩阵向量乘积和偏移量的和,并且计算如下。

[0113] $pred_{red}=A\cdot bdry_{red}+b$

[0114] $pred_{red}$的大小为$W_{red}\times H_{red}$。根据当前块的大小和形状来定义W_{red}和H_{red},如下所示。矩阵A的行数与$W_{red}\times H_{red}$一样多,并且当W＝H＝4时为4列,或其它情况下为8列。偏移向量b是大小为$W_{red}\times H_{red}$的向量。

[0115]
$$W_{red}=\begin{cases}4 & \text{对于}\max(W,H)\leqslant 8\\ \min(W,8) & \text{对于}\max(W,H)>8\end{cases}$$
$$H_{red}=\begin{cases}4 & \text{对于}\max(W,H)\leqslant 8\\ \min(H,8) & \text{对于}\max(W,H)>8\end{cases}$$

[0116] 对于每种类别的块大小,预定义可以用于块的矩阵A和偏移向量b的集合S_0、S_1和S_2。根据上述MipSizeId(即,idx(W,H))选择集合S的指数(0,1,2),并且根据应用于当前块的MIP模式从集合S_0、S_1和S_2的一个提取矩阵A和偏移向量b。

[0117] 集合S_0由18个矩阵A_0和18个16维偏移向量b_0组成,并且用于4×4块,每个矩阵A_0具有16行4列。集合S_1由10个矩阵A_1和10个16维偏移向量b_1组成,并且用于4×8、8×4和8×8大小的块,每个矩阵A_1具有16行8列。最后,集合S_2由6个矩阵A_2和6个64维偏移向量b_2组成,并且用于所有其他块形状,每个矩阵A_2具有64行8列。

[0118] (3)像素插值

[0119] 插值是上采样过程。如上所述,$pred_{red}$是原始块的下采样的预测信号。具体地,具有predW和predH大小的下采样的预测块定义如下。

[0120] $pred_{red}[x][y]$,其中x＝0……predW1,y＝0……predH-1

[0121] 通过在每个方向上的剩余位置线性地插值预测信号而生成的具有原始块大小(nTbW,nTbH)的预测块定义如下。

[0122] predSamples[x][y],其中x＝0……nTbW-1,y＝0……nTbH-1

[0123] 根据水平上采样缩放因子upHor(=nTbW/predW)和竖直上采样缩放因子upVer(=nTbH/predH),如下所述从$pred_{red}$填充predSamples的部分或全部。

[0124] $predSamples[(x+1)\times upHor-1][(y+1)\times upVer-1]=pred_{red}[x][y]$

[0125] 当upHor=1时,$pred_{red}$中的predSamples的所有水平位置被填充。当upVer=1时,$pred_{red}$中的predSamples的所有竖直位置被填充。

[0126] 然后,通过双线性插值来填充predSamples的剩余的空白样本。水平方向上的插值和竖直方向上的插值是上采样过程。对于predSamples中左侧和顶部样本的插值,将下采样的样本$bdry_{red}^{top}$分配给predSamples[x][-1]的值,并且将左侧的原始参考样本分配给predSamples[-1][y]的值。

[0127] (4)用信号通知MIP帧内预测模式

[0128] 对于经过帧内预测编码的每个编码单元(CU),传输指示是否应用基于矩阵的帧内预测模式(即,MIP模式)的标志。在VVC草案5中,为了用信号通知MIP模式,以与基于矩阵的帧内预测不同的传统帧内预测模式(在下文中,称为“常规帧内预测模式”)相同的方式使用MPM列表。例如,intra_mip_mpm_flag、intra_mip_mpm_idx和intra_mip_mpm_remainder用于用信号通知MIP模式。利用截断的二进制码对intra_mip_mpm_idx进行编码,利用固定长度编码对intra_mip_mpm_remainder进行编码。

[0129] 根据编码块(CU)的大小,可以支持多达35个MIP模式。例如,对于具有max(W,H)≤8且W×H<32的CU,35个模式是可用的。另外,对于具有max(W,H)=8和max(W,H)>8的CU分别使用19个预测模式和11个预测模式。另外,一对模式(即,两个模式)可以共享矩阵和偏移向量,以减少存储器需求。具体的共享模式计算如下。例如,对于4×4编码块,模式19使用分配给模式2的矩阵的转置矩阵。

[0130]
$$m=\begin{cases}\text{模式} & \text{对于}\ W=H=4\ \text{且}\ \text{模式}<18\\ \text{模式}-17 & \text{对于}\ W=H=4\ \text{且}\ \text{模式}\geqslant 18\\ \text{模式} & \text{对于}\ \max(W,H)=8\ \text{且}\ \text{模式}<10\\ \text{模式}-9 & \text{对于}\ \max(W,H)=8\ \text{且}\ \text{模式}\geqslant 10\\ \text{模式} & \text{对于}\ \max(W,H)>8\ \text{且}\ \text{模式}<6\\ \text{模式}-5 & \text{对于}\ \max(W,H)>8\ \text{且}\ \text{模式}\geqslant 6\end{cases}$$

[0131] 当在应用常规帧内预测模式而不是MIP的块(在下文中,称为“常规块”)附近存在应用MIP的块时,在MIP模式与常规模式之间定义的映射表可以用于常规块的MPM推导。映射表用于从应用MIP的相邻块的MIP模式推导相似特性的常规模式。以这种方式推导的常规模式用于常规块的MPM推导。类似地,即使当MIP应用于在色度DM推导中使用的并置亮度块时,也能够利用映射表推导并置亮度块的常规模式,并且将推导出的常规模式用于色度DM推导。下面的等式利用映射表表达常规模式与MIP模式之间的映射。

[0132] $predmode_{MIP}=map_regular_to_mip_{idx}[predmode_{regular}]$

[0133] $predmode_{regular}=map_mip_to_regular_{idx}[predmode_{MIP}]$

[0134] 如上所述,在VVC草案5中,当基于MIP来预测块时,总是将块左侧的重构的相邻样本$bdry^{left}$和块上方的重构的相邻样本$bdry^{top}$用作参考样本。在块的纹理具有方向特性的块的情况下,该方法可能会使预测性能变差。另外,在VVC草案5中,为了对predSamples中的顶部样本进行插值,将下采样的样本集$bdry_{red}^{top}$分配给predSamples[x][-1]的值,如图5所

示,从而使得插值过程比必要的更复杂。另外,在VVC草案5中,在MPM列表用于用信号通知MIP模式和常规模式两者的情况下,由于许多检查和条件,例如在MIP模式与常规模式之间的映射要求,实施方案可能会非常复杂。

[0135] 鉴于上述,本发明提出了一些用于MPM编码的改进技术。根据本发明的一个方面,提供了这样一种方法,其可以降低MIP模式的实现复杂性并且考虑块的纹理可能具有的方向特性。

[0136] 图6是示出根据本发明实施方案的基于MIP来预测块的示意性过程的流程图。

[0137] 1.解码MIP模式(S610)

[0138] 对于以帧内预测模式编码的CU,视频解码器解码指示帧内预测类型是否是MIP的标志。当将MIP应用于当前CU时,视频解码器从多个可用的MIP模式中解码指示当前CU中使用的MIP模式的语法元素。

[0139] 与传统的帧内预测模式(即,常规的帧内预测模式)不同,MPM列表可能不会用于用信号通知MIP模式。而是,例如,可以使用这样一个语法元素(例如,intra_mip_mode),所述语法元素可以用截断的二进制码进行编码并且指示在多个MIP模式中的当前CU中使用的MIP模式。

[0140] 下面提供基于VVC草案5提出的示例性编码单元语法的一部分。在下面的语法中,灰色的元素用于提供理解。

[0141] 表7

[0142]

```
if( CuPredMode[ x0 ][ y0 ]  ==  MODE_INTRA ) {
  if( treeType == SINGLE_TREE  ||  treeType == DUAL_TREE_LUMA ) {
    if( sps_bdpcm_enabled_flag &&
      cbWidth <= MaxTsSize  &&  cbHeight <= MaxTsSize  )
      intra_bdpcm_flag[ x0 ][ y0 ]
    if( intra_bdpcm_flag[ x0 ][ y0 ] )
      intra_bdpcm_dir_flag[ x0 ][ y0 ]
    else {
      if(  sps_mip_enabled_flag  &&
        ( Abs( Log2( cbWidth ) - Log2( cbHeight ) ) <= 2 )  &&
          cbWidth <= MaxTbSizeY  &&  cbHeight <= MaxTbSizeY )
        intra_mip_flag[ x0 ][ y0 ]
      if( intra_mip_flag[ x0 ][ y0 ] )
        intra_mip_mode[ x0 ][ y0 ]
```

[0143] 当intra_mip_flag[x0][y0]为1时,指示出当前块的帧内预测类型为MIP。当intra_mip_flag[x0][y0]为0时,指示出当前块的帧内预测类型是常规帧内预测,而不是MIP。当intra_mip_flag[x0][y0]不存在时,可以被推断为等于0。intra_mip_mode[x0][y0]指定用于MIP中当前块的MIP模式。

[0144] 可以根据MIP模式不同地确定要用于生成预测信号的当前编码块的相邻样本。作为一个实例,对于具有4×4大小的编码块,35个MIP模式是可用的,并且可以根据模式的范

围来确定要用于生成预测信号的相邻样本,如下所示。

[0145] $$\begin{cases} \text{左侧相邻样本} & \text{对于 模式} < 12 \\ \text{上方相邻样本} & \text{对于 模式} \geqslant 24 \\ \text{左侧相邻样本和上方相邻样本} & \text{否则} \end{cases}$$

[0146] 在当前编码块的像素具有水平特性时,利用左侧相邻样本来生成预测信号可能会是有利的。类似地,在当前编码块的像素具有竖直特性时,利用上方相邻样本来生成预测信号可能会是有利的。

[0147] 为了选择适合于块的方向特性的MIP模式,视频编码器可以配置为将帧内预测的常规模式应用于块,并且对于每个常规模式来计算块的失真值。当水平预测模式(例如,INTRA_ANGULAR_14至INTRA_ANGULAR_22)的失真值小于其他模式的失真值时,可以将水平方向确定为当前编码块的主导方向。相反,当竖直预测模式(例如,INTRA_ANGULAR_46至INTRA_ANGULAR_54)的失真值小于其他模式的失真值时,可以将竖直方向确定为当前编码块的主导方向。

[0148] 当水平方向是当前编码块的主导方向时,编码器将仅使用左侧相邻样本的MIP模式的一个分配给当前编码块。另一方面,当竖直方向是主要方向时,编码器可以将仅使用上方相邻样本的MIP模式的一个分配给当前编码块。

[0149] 在下文中,为了简单起见,将针对利用左侧相邻样本的情况和利用上方相邻样本的情况描述边界向量和根据边界向量来确定预测的样本的方法。当同时利用左侧相邻样本和上方相邻样本二者时,该过程可以与参考图5所描述的基本相同。

[0150] 2-1.边界向量生成——利用左侧相邻样本(S620)

[0151] 图7a至图7c是示出利用左侧相邻样本构建输入到矩阵向量乘法运算的边界向量的示例性方法的概念图。

[0152] 作为一个示例,如图7a所示,在当前编码块的大小(高度)与边界向量$bdry_{red}$的大小相同时,可以利用左侧相邻样本集$bdry^{left}$来填充相同大小的边界向量$bdry_{red}$。例如,在边界向量的项中可以包括左侧相邻样本的每个。

[0153] 作为另一个示例,如图7b所示,可以通过利用与编码块邻近的两个左侧列计算每行中的两个像素的平均值来填充边界向量$bdry_{red}$。

[0154] 作为另一个示例,如图7c所示,可以利用从左侧相邻样本集$bdry^{left}$获得的下采样的样本集$bdry_{red}^{left}$来填充边界向量$bdry_{red}$。可以通过平均每两个样本的$bdry^{left}$来获得$bdry_{red}^{left}$。

[0155] 根据编码块的大小,可以自适应地使用bdryleft和$bdry_{red}^{left}$。例如,如下面的等式所示,当H≤8时,使用bdryleft。否则,可以使用$bdry_{red}^{left}$。

[0156] $$bdry_{red} = \begin{cases} [bdry^{left}] & \text{对于 } H \leqslant 8 \\ [bdry_{red}^{left}] & \text{否则} \end{cases}$$

[0157] 2-2.边界向量生成——利用上方相邻样本(S620)

[0158] 图8a至图8c是示出利用上方相邻样本构建输入到矩阵向量乘法运算的边界向量的示例性方法的概念图。

[0159] 作为示例,如图8a所示,在当前编码块的大小(宽度)与边界向量$bdry_{red}$的大小相同时,可以利用上方相邻样本集来填充相同大小的边界向量$bdry_{red}$。例如,在边界向量的项中可以包括上方相邻样本的每个。

[0160] 作为另一个示例,如图8b所示,可以通过利用与编码块邻近的两个上方行计算每列中的两个像素的平均值来填充边界向量$bdry_{red}$。

[0161] 作为又一个示例,如图8c所示,可以利用从上方相邻样本集$bdry^{top}$获得的下采样的样本集$\underline{bdry_{red}^{top}}$来填充边界向量$bdry_{red}$。另外,可以通过平均$bdry^{top}$的每两个样本来获得$bdry_{red}^{top}$。

[0162] 根据编码块的大小,可以自适应地使用$bdry^{top}$和$bdry_{red}^{top}$。例如,如下面的等式所示,当H≤8时,可以使用$bdry^{top}$。否则,可以使用$bdry_{red}^{top}$。

[0163] $$bdry_{red} = \begin{cases} [bdry^{top}] & \text{对于} H \leqslant 8 \\ [bdry_{red}^{top}] & \text{否则} \end{cases}$$

[0164] 3.矩阵向量乘法和偏移量加法(S630)

[0165] 视频解码器配置为对边界向量$bdry_{red}$执行矩阵向量乘法,并且将偏移向量与其相加。可以根据解码的MIP模式k来确定矩阵A和偏移向量b。作为结果,生成原始块的预测信号或子采样的预测信号$pred_{red}$。$pred_{red}$计算如下。

[0166] $pred_{red} = A \cdot bdry_{red} + b$

[0167] 这里,$pred_{red}$的大小为$W_{red} \times H_{red}$。

[0168] 当利用左侧相邻样本构建输入到矩阵向量乘法的边界向量时,可以根据编码块的大小来确定W_{red}和H_{red},如下所示。

[0169]
$$W_{red} = \begin{cases} 4 & \text{对于} \max(W,H) \leqslant 8 \\ \min(W,8) & \text{对于} \max(W,H) > 8 \end{cases}$$
$$H_{red} = \begin{cases} 4 & \text{对于} H \leqslant 8 \\ 8 & \text{对于} H > 8 \end{cases}$$

[0170] 当利用上方相邻样本构建输入到矩阵向量乘法的边界向量时,可以根据编码块的大小来确定W_{red}和H_{red},如下所示。

[0171]
$$W_{red} = \begin{cases} 4 & \text{对于} W \leqslant 8 \\ 8 & \text{对于} W > 8 \end{cases}$$
$$H_{red} = \begin{cases} 4 & \text{对于} \max(W,H) \leqslant 8 \\ \min(H,8) & \text{对于} \max(W,H) > 8 \end{cases}$$

[0172] 当生成的$pred_{red}$大小少于编码块的样本的数量时,需要插值。

[0173] 4.线性插值(S640)

[0174] 图9示出左侧相邻样本、边界向量和由此获得的预测的样本。在图9中,由用于当前编码块的预测块predSamples[x][y]中的三角形指示的像素具有从生成的$pred_{red}$分配的值。

[0175] 参考图9,可以使用各种插值方法以确定预测块predSamples[x][y]中的空像素。

[0176] 例如,如在像素1的情况下,可以利用左侧相邻样本、上方相邻样本和在空像素下

方分配的$pred_{red}$的值来确定空像素的值。作为另一个示例,如在像素2的情况下,可以利用上方相邻样本和在空像素下方分配的$pred_{red}$的值来确定空像素值。作为另一个示例,如在像素3的情况下,可以利用在空像素上方和下方分配的$pred_{red}$的两个值来确定空像素值。作为另一个示例,如在像素4的情况下,可以利用左侧相邻样本、在空像素的上方和下方分配的$pred_{red}$的两个值来确定空像素值。

[0177] 图10示出上方相邻样本、边界向量和由此获得的预测的样本。在图10中,由用于当前编码块的预测块predSamples[x][y]中的三角形指示的像素具有根据生成的$pred_{red}$分配的值。

[0178] 参考图10,可以使用各种插值方法以确定预测块predSamples[x][y]中的空像素。例如,如在像素1的情况下,可以利用左侧相邻样本、上方相邻样本和在空像素下方分配的$pred_{red}$的值来确定空像素值。作为另一个示例,如在像素2的情况下,可以利用左侧相邻样本、在空像素上方和下方分配的$pred_{red}$的两个值来确定空像素值。作为另一个示例,如在像素3的情况下,可以利用在空像素上方和下方分配的$pred_{red}$的两个值来确定空像素值。作为另一个示例,如在像素4的情况下,可以利用上方相邻样本和在空像素下方分配的$pred_{red}$的值来确定空像素值。

[0179] 与图9和图10中所示的示例不同,为了确定预测块predSamples[x][y]中的空像素,可以将上方原始参考样本$bdry^{top}$分配给predSamples[x][-1],并且可以将左侧原始参考样本$bdry^{left}$分配给predSamples[-1][y]。

[0180] 在上述实施方案中,引入根据MIP模式选择性地利用当前块的相邻样本的方案,使得在基于MIP来生成预测块时可以考虑块的方向性。

[0181] 根据本发明的另一方面,代替根据MIP模式来改变用于生成边界向量的相邻样本集,可以修改应用于边界向量的矩阵A和偏移量b的值。

[0182] 应用于边界向量的矩阵向量乘法和偏移量加法的矩阵A和偏移量b可以分为如下两个部分。

[0183]

$$pred_{red} = A\begin{pmatrix} bdry_{red}^{left} \\ bdry_{red}^{top} \end{pmatrix} + (b)$$

$$pred_{red} = (A1|A2)\begin{pmatrix} bdry_{red}^{left} \\ bdry_{red}^{top} \end{pmatrix} + \begin{pmatrix} b1 \\ b2 \end{pmatrix}$$

[0184] 当A2和b2的值近似为0时,预测信号$pred_{red}$主要受左侧相邻像素$bdry_{red}^{left}$的影响,这对于块的纹理具有水平方向性的块可能会是有用的。当A1和b1的值近似为0时,预测信号$pred_{red}$主要受上方相邻像素$bdry_{red}^{top}$的影响,这对于块的纹理具有竖直方向性的块可能会是有用的。

[0185] 视频编码器和视频解码器可以根据当前块的MIP模式,使用要应用于边界向量的矩阵A和偏移量b的修改的值。例如,对于4×4的CU,35个MIP模式可以是可用的,并且可以根据模式的范围来修改要应用于边界向量的矩阵A和偏移量b的值。

[0186] $$\begin{cases} A2 \text{ 和 } b2 \cong 0 & \text{对于 模式} < 12 \\ A1 \text{ 和 } b2 \cong 0 & \text{对于 模式} \geqslant 24 \\ \text{使用} A \text{ 和 } b & \text{否则} \end{cases}$$

[0187] 如上所述,VVC草案5根据CU的大小和形状支持多达35个MIP模式。例如,对于具有max(W,H)≤8并且W×H<32的CU,35个模式是可用的,对于具有max(W,H)＝8的CU,19个预测模式是可用的,而对于具有max(W,H)>8的CU,11个预测模式是可用的。另外,一对模式(两个模式)可以共享矩阵和偏移向量,以减少存储器需求。例如,对于4×4的编码块,模式19使用分配给模式2的矩阵的转置矩阵。另外,通过以根据MIP模式和块的大小确定的顺序级联$bdry_{red}^{top}$和$bdry_{red}^{left}$,生成要输入到向量矩阵乘法的边界向量$bdry_{red}$。

[0188] 可以使用一种改进的方法,其可以在降低复杂性时实现基本等效的编码效率水平。根据本发明的另一个方面,代替一个模式使用由另一个模式使用的矩阵的转置的现有方案,可以使用配置为改变级联构成边界向量$bdry_{red}$的$bdry_{red}^{top}$和$bdry_{red}^{left}$的顺序的新方案,所述边界向量$bdry_{red}$用于针对每个模式的向量矩阵乘法。视频编码器可以用信号通知语法元素,该语法元素指示将$bdry_{red}^{top}$和$bdry_{red}^{left}$级联以构成用于每个模式的边界向量$bdry_{red}$的顺序。根据该方案,可用的MIP模式的数量可以比传统方法减少一半,并且可以降低边界向量生成和向量矩阵乘法方面的编码复杂性。

[0189] 如关于表5的语法结构所描述的,对于满足特定条件的变换单元的每个,用信号通知作为与跳过变换相关的语法元素的transform_skip_flag。当transform_skip_flag不为真(TRUE)时,必须根据SPS中的标志(sps_explicit_mts_inter_enabled_flag、sps_explicit_mts_intra_enabled_flag)来传输作为与多变换选择相关的语法元素的tu_mts_idx,从而视在比特流中要传输的比特的数量情况而施加负担。

[0190] 然而,多变换选择和变换跳过与残差信号的特性密切相关,因此,无论块中使用的预测模式如何,总是用信号通知变换相关的语法元素可能并不有效。应当注意的是,考虑到包括平均化操作和插值操作的MIP处理,应用MIP的块的残差信号具有在变换域中占主导地位的低频分量,并且可能的是不会将MIP应用于具有像素之间低相关性的块,例如包含复杂运动和纹理的块。因此,可能的是不会将变换跳过应用于使用MIP的块。

[0191] 鉴于上述,无论是否将MIP应用于块,总是用信号通知作为与多变换选择相关的语法元素的tu_mts_idx以及作为与变换跳过相关的语法元素的transform_skip_flag的方法可能不是非常有效的。

[0192] 根据本发明的一个方面,可以根据是否应用MIP来省略用信号通知变换相关的语法元素,并且可以由视频解码器隐式地推断省略的语法元素。

[0193] 在实施方案中,可以根据MIP标志隐式地应用MTS技术。如下表所示,只有当intra_mip_flag不为真时,才可以对transform_skip_flag和tu_mts_idx进行编码。当intra_mip_flag为真时,将transform_skip_flag推断为0,将tu_mts_idx也推断为0,并且推断要应用隐式MTS。相应地,可以根据块大小来确定水平方向和竖直方向上的变换核。在下面的语法中,灰色的元素用于指示语法中的潜在变化或提供其理解。

[0194] 表8

[0195]

transform_unit(x0, y0, tbWidth, tbHeight, treeType, subTuIndex) {	描述符
（省略）	

if(tu_cbf_luma[x0][y0] && treeType != DUAL_TREE_CHROMA
&& (tbWidth <= 32) && (tbHeight <= 32)
&& (IntraSubPartitionsSplit[x0][y0] == ISP_NO_SPLIT) && (!cu_sbt_flag)) {	

if(transform_skip_enabled_flag && tbWidth <= MaxTsSize && tbHeight <= MaxTsSize && (!intra_mip_flag[x0][y0]))	
transform_skip_flag[x0][y0]	ae(v)
if(((CuPredMode[x0][y0] != MODE_INTRA && sps_explicit_mts_inter_enabled_flag) || (CuPredMode[x0][y0] == MODE_INTRA && sps_explicit_mts_intra_enabled_flag)) && (!transform_skip_flag[x0][y0]) && (!intra_mip_flag[x0][y0]))	
tu_mts_idx[x0][y0]	ae(v)
}	
if(tu_cbf_luma[x0][y0]) {	
if(!transform_skip_flag[x0][y0])	
residual_coding(x0, y0, Log2(tbWidth), Log2(tbHeight), 0)	
else	
residual_ts_coding(x0, y0, Log2(tbWidth), Log2(tbHeight), 0)	
}	
（省略）	

[0196] 在另一实施方案中,可以根据MIP标志显式地应用MTS技术。如下表所示,只有当intra_mip_flag不为真时,才可以对transform_skip_flag和tu_mts_idx进行编码。当intra_mip_flag为真时,将transform_skip_flag推断为0,显式地用信号通知tu_mts_idx,并且应用显式的MTS。在下面的语法中,灰色的元素用于指示语法中的潜在变化或提供其理解。

[0197] 表9

[0198]

transform_unit(x0, y0, tbWidth, tbHeight, treeType, subTuIndex) {	描述符
（省略）	

if(tu_cbf_luma[x0][y0] && treeType != DUAL_TREE_CHROMA
&& (tbWidth <= 32) && (tbHeight <= 32)
&& (IntraSubPartitionsSplit[x0][y0] == ISP_NO_SPLIT) && (!cu_sbt_flag)) {	

if(transform_skip_enabled_flag && tbWidth <= MaxTsSize && tbHeight <= MaxTsSize && (!intra_mip_flag[x0][y0]))	
transform_skip_flag[x0][y0]	ae(v)
if(((CuPredMode[x0][y0] != MODE_INTRA && sps_explicit_mts_inter_enabled_flag) || (CuPredMode[x0][y0] == MODE_INTRA && sps_explicit_mts_intra_enabled_flag)) && (!transform_skip_flag[x0][y0]))	
tu_mts_idx[x0][y0]	ae(v)
}	
if(tu_cbf_luma[x0][y0]) {	
if(!transform_skip_flag[x0][y0])	
residual_coding(x0, y0, Log2(tbWidth), Log2(tbHeight), 0)	
else	
residual_ts_coding(x0, y0, Log2(tbWidth), Log2(tbHeight), 0)	
}	
（省略）	

[0199] 在另一个可能的实施方案中,MIP可能不会显式地应用于具有像素之间低相关性的块,例如具有包含复杂运动和纹理的较小尺寸的块。可能不会将变换跳过(Transform skip,TS)应用于应用MIP的4×4的CU。另外,可能不会将MIP应用于应用TS的4×4的CU。另外,可能不会将MIP与TS一起应用于4×4的CU。

[0200] 在传统方法中,可以使用采用最可能模式(Most Probable Mode,MPM)的帧内预测编码。例如,在HEVC中,从左侧块和上方块的帧内预测模式来配置三个MPM的列表。这种方法的缺点是更多的模式(除MPM以外的帧内模式)属于需要用更多比特编码的非MPM。已经提出了数种方法来将MPM的数量扩展到3项或更多项(例如,6个MPM模式)。然而,用更多的项配置这样的MPM列表可能需要更多的检查和条件,这可能会使实施方案更加复杂。

[0201] 如上所述,在VVC草案5中,在每个MPM列表用于用信号通知MIP模式和常规模式的情况下,由于许多检查和条件,例如为了配置MIP列表而在MIP模式与常规模式之间映射的要求,实施方案可能会比较复杂。

[0202] 为了保持MPM列表的配置的低复杂性,可以利用与当前块邻近的左侧相邻块和上方相邻块的帧内预测模式来配置包括六个MPM候选的MPM列表。MPM候选可以包括:默认帧内预测模式(例如,平面模式)、相邻块的帧内预测模式、以及从相邻块的帧内预测模式推导的帧内预测模式。当不使用相邻块的帧内预测模式时(例如,当对相邻块进行帧间预测时,或者相邻块位于不同的切片或另一个瓦片中时),可以将相邻块的帧内预测模式设置为平面模式(Planar)。

[0203] 根据左侧块的模式(左侧模式(Left))和上方块的模式(上方模式(Above))的帧内

预测模式的类型,分为4种情况。当左侧模式和上方模式彼此不同,并且两个模式为方向模式时,可以根据左侧模式和上方模式的不同再包括4种情况,以生成MPM列表。在下表中,Max是指左侧模式和上方模式之间的较大模式,而MIN是指左侧模式和上方模式之间的较小模式。

[0204] 表10

[0205]

条件	具体条件	MPM 模式
左侧模式和上方模式为方向模式并且相同		{Planar, Left, Left-1, Left+1, Left-2, Left+2}
左侧模式和上方模式不同，并且两种模式为方向模式	Max-Min = 1	{Planar, Left, Above, Min-1, Max+1, Min-2}
	Max-Min ≥ 62	{Planar, Left, Above, Min+1, Max-1, Min+2}
	Max-Min ≥ 2	{Planar, Left, Above, Min+1, Min-1, Max+1}
	2 < Max-Min < 62	{Planar, Left, Above, Min-1, Min+1, Max-1}
左侧模式和上方模式不同，并且仅其一为方向模式		{Planar, Max, Max-1, Max+1, Max-2, Max+2}
左侧模式和上方模式两者为非方向模式（即，平面或DC）		{Planar, DC, Angular50, Angular18, Angular46, Angular54}

[0206] 由于包括平均化操作和插值操作的MIP技术的特性,应用MIP的块的残差信号可以具有在变换域中占主导地位的低频分量。注意,该残差特性类似于应用平面模式或DC模式的块的残差信号。因此,在推导以常规帧内预测模式编码的块的MPM列表时,考虑这种残差信号的相似性可能会是有用的。

[0207] 根据本发明的一个方面,在推导用于以常规帧内预测模式编码的块(即,常规块)的MPM列表时,当以MIP模式编码相邻块时,可以将相邻块的帧内预测模式视为平面模式(或DC模式)。作为示例,当将MIP模式应用于相邻块时,可以将平面模式(或DC模式)添加到MPM列表中取代相邻块的MIP模式。

[0208] 类似地,如果将MIP应用于并置的亮度块,即使当推导色度DM(直接模式)时,也可以将亮度块的帧内预测模式视为平面模式(或DC模式),而不是利用MIP模式与常规模式之间的映射表。相应地,编码器解析指定用于色度块的帧内预测模式的语法元素,并且所述语法元素可以指示出色度块的帧内预测模式采用并置的亮度块的帧内预测模式。在这种情况下,当将MIP应用于并置的亮度块时,可以将亮度块的帧内预测模式视为平面模式(或DC模式)。换句话说,当将MIP应用于色度直接模式(DM)中的并置的亮度块时,可以确定出色度块的帧内预测模式是平面模式(或DC模式)。

[0209] 视频编码器可以用信号通知指示当前块的帧内预测模式是否对应于MPM的1位标志(例如,mpm_flag)。通常,在当前块的帧内预测模式对应于MPM时,额外地用信号通知指示6个MPM的一个的MPM索引。注意,在上表中,平面模式始终包括在MPM列表中。因此,对于编码器来说,显式地用信号通知当前块的帧内预测模式是否是平面模式(例如,利用1位标志),并且在当前块的帧内预测模式与其他五个MPM的一个相同时,额外地用信号通知指示其他五个MPM的一个的MPM索引会是有效的。在当前块的帧内预测模式不对应于任何MPM时,可以利用截断的二进制码对指示除6个MPM之外的其余61个非MPM的一个的语法元素进行编码。

[0210] 图11是示出根据本发明实施方案的对视频数据解码的方法的流程图。

[0211] 视频解码器可以配置为从比特流解码指示视频数据的当前块的帧内预测类型的语法元素(S1110)。帧内预测类型包括基于矩阵的帧内预测(MIP)和常规帧内预测。语法元

素可以是指定为对于当前块的大小和形状所允许的多个MIP模式的一个的截断的二进制码。

[0212] 视频解码器可以配置为通过基于当前块的帧内预测类型选择性地执行MIP或常规帧内预测来生成当前块的预测块。

[0213] 作为通过执行常规帧内预测来生成当前块的预测块的一部分,视频解码器可以配置为执行以下操作(S1120至S1140)。视频解码器可以通过基于与当前块邻近的相邻块的常规帧内预测模式推导最可能模式(MPM)候选来配置当前块的MPM列表(S1120),并且基于MPM列表来推导当前块的常规帧内预测模式(S1130)。在基于相邻块的常规帧内预测模式来推导MPM候选时,视频解码器可以配置为当相邻块的帧内预测类型是基于矩阵的帧内预测时,将相邻块的常规帧内预测模式设置为(视为)平面模式。解码器可以配置为基于当前块的常规帧内预测模式来生成当前块的预测块(S1140)。

[0214] 作为通过执行基于矩阵的帧内预测来生成当前块的预测块的一部分,视频解码器可以配置为执行以下操作(S1121至S1151)。视频解码器可以配置为从比特流解码指示当前块的基于矩阵的帧内预测模式的语法元素,以确定当前块的基于矩阵的帧内预测模式(S1121)。视频解码器可以配置为基于当前块的大小和形状利用与当前块邻近的相邻样本来推导边界向量(S1131),并且基于当前块的基于矩阵的帧内预测模式预定义的矩阵与边界向量之间的矩阵向量乘法来生成当前块的预测的样本(S1141)。视频解码器可以配置为通过基于预测的样本执行线性插值和裁剪来推导当前块的预测块(S1151)。

[0215] 视频解码器可以配置为根据当前块的基于矩阵的帧内预测模式,从与当前块邻近的左侧相邻样本集与上方相邻样本集之间选择的一个或两个集中来推导边界向量。

[0216] 视频解码器配置为从比特流解码语法元素,该语法元素指示了从与当前块邻近的左侧相邻样本推导的边界向量的第一项和从与当前块邻近的上方相邻样本推导的边界向量的第二项的级联顺序。视频解码器可以配置为通过根据指示的级联顺序级联第一项和第二项来生成边界向量。

[0217] 边界向量的项可以根据当前块的大小和形状用从与当前块邻近的左侧相邻样本下采样的值填充,或者用与当前块邻近的左侧相邻样本填充。

[0218] 视频解码器可以将预测的样本分配至预测块中的位置,以基于预测的样本推导当前块的预测块。为了针对预测块中未分配预测的样本的位置来生成预测的样本值,可以对预测的样本、与当前块邻近的左侧相邻样本以及与当前块邻近的上方相邻样本执行线性插值。

[0219] 应当理解的是,可以以许多不同的方式实现上述示例性实施方案。在一个或更多个示例中描述的功能或方法可以实现为硬件、软件、或者其任何组合。应当理解的是,本文中描述的功能组件已被标记为“单元”,以进一步强调其实施独立性。本发明中描述的各种功能或方法可以用存储在非易失性记录介质中的指令来实现,所述指令可以由一个或更多个处理器读取和执行。非易失性记录介质包括,例如,所有类型的记录器件,其中数据以计算机系统可读的形式存储。例如,非易失性记录介质包括存储介质,例如可擦除可编程只读存储器(EPROM)、闪存驱动器,光盘驱动器、磁性硬盘驱动器和固态驱动器(SSD)。

[0220] 尽管出于说明的目的已经描述了本发明的示例性实施方案,但是本领域的技术人员应当理解,在不脱离本发明思想和范围的情况下,各种修改和改变是可能的。为了简洁和

清楚起见,已经描述了示例性实施方案。相应地,普通技术人员应当理解,实施方案的范围不受以上明确描述的实施方案限制,而是包括在权利要求书及其等同形式内。

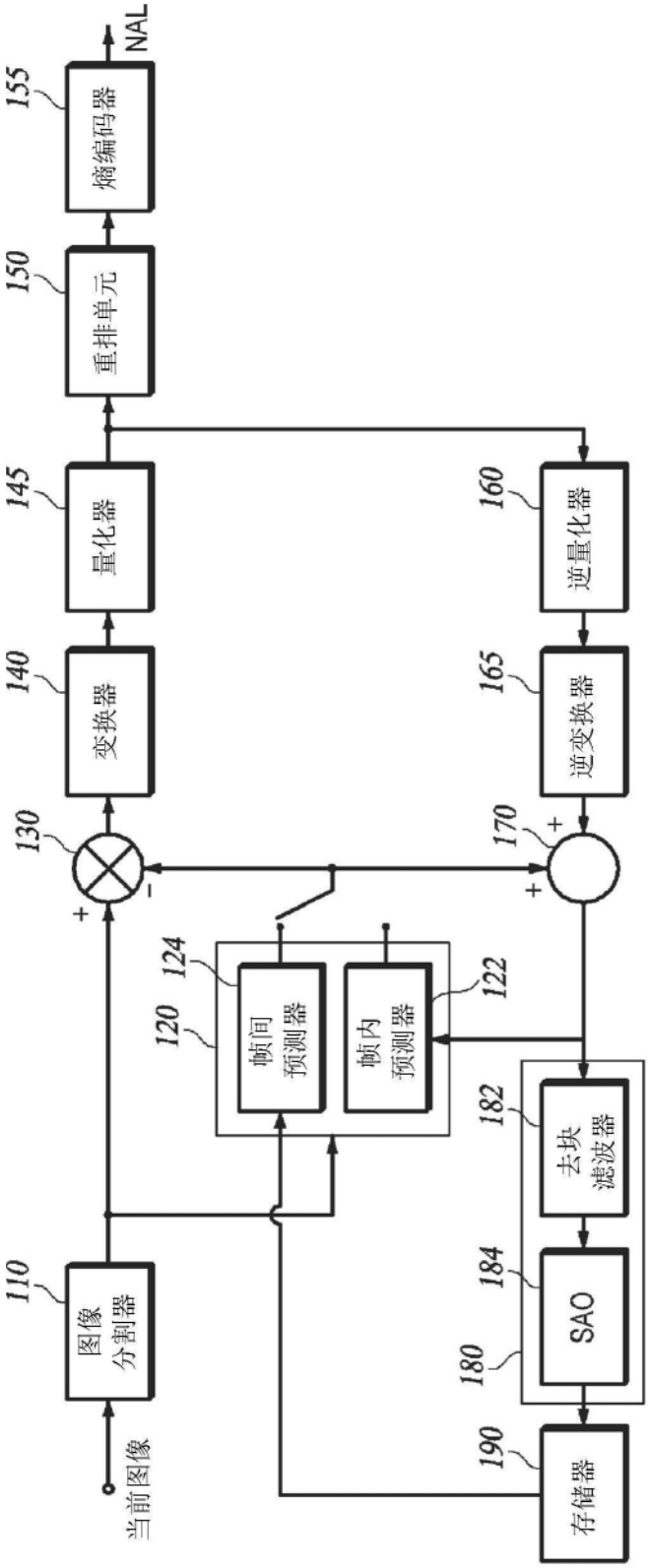

当前图像
110
图像分割器
130
+
-
140
变换器
145
量化器
150
重排单元
155
熵编码器
NAL
120
124
帧间预测器
122
帧内预测器
160
逆量化器
165
逆变换器
170
+
+
180
182
去块滤波器
184
SAO
190
存储器

图1

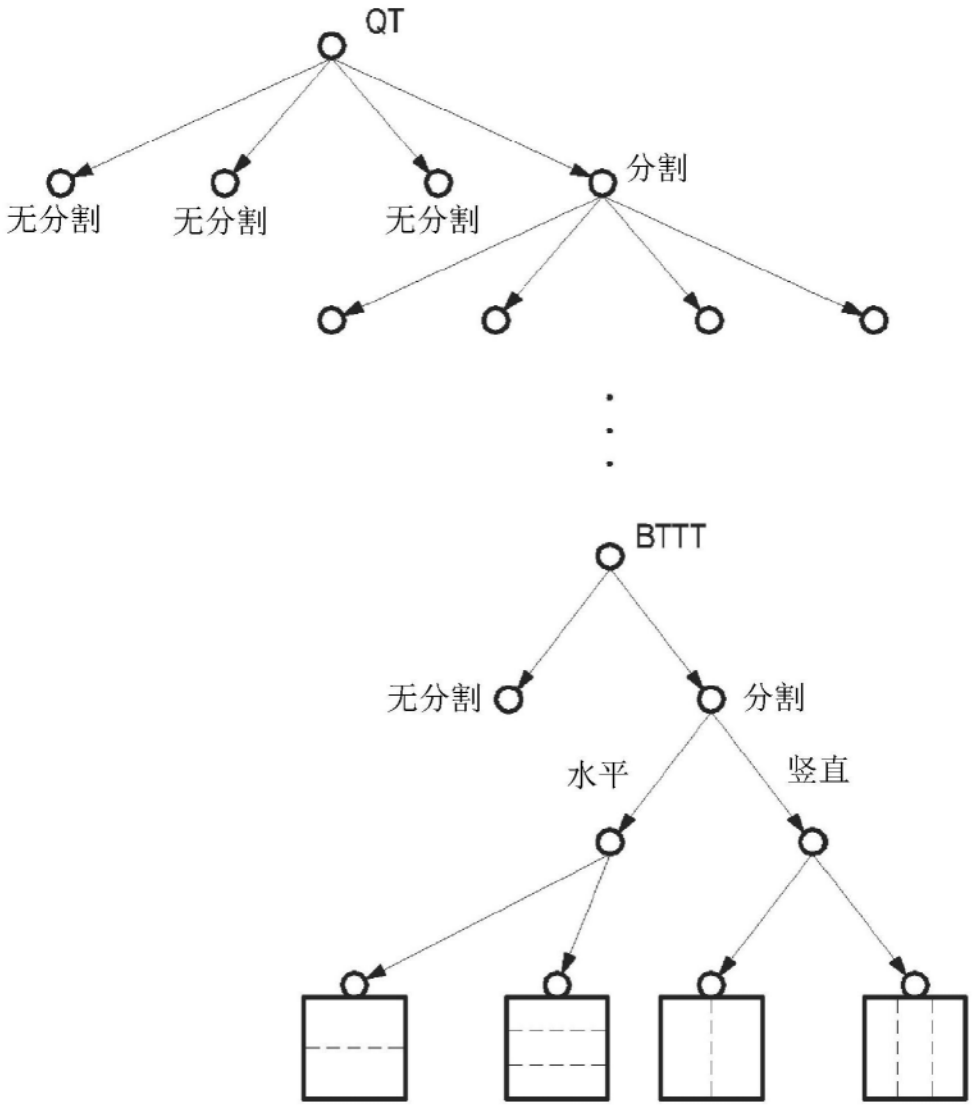


图2

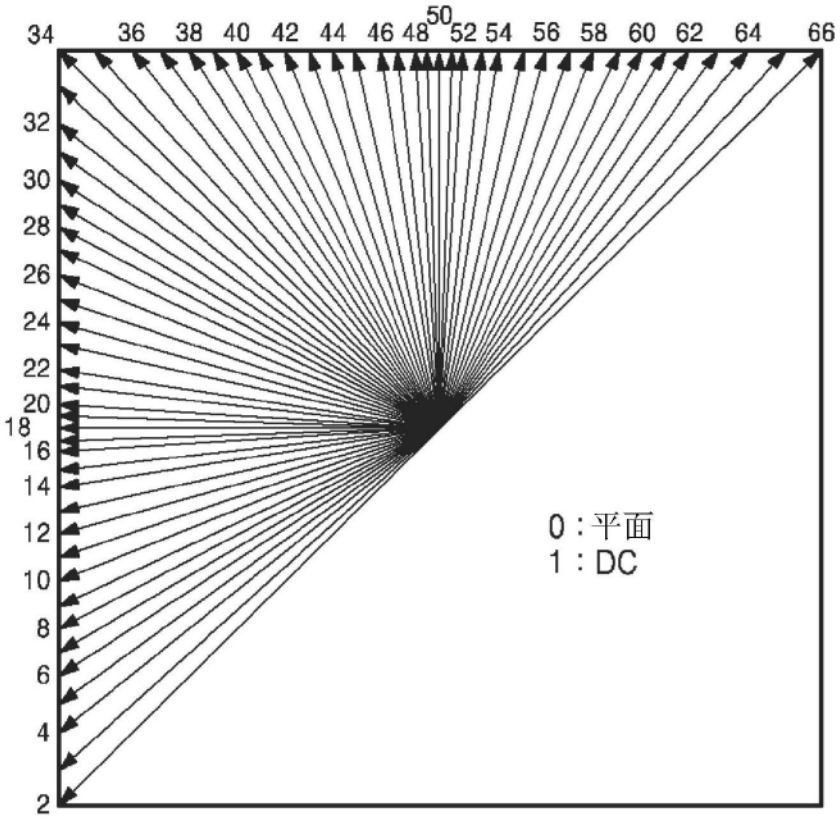

34
36
38
40
42
44
46
48
50
52
54
56
58
60
62
64
66
32
30
28
26
24
22
20
18
16
14
12
10
8
6
4
2
0：平面
1：DC

图3a

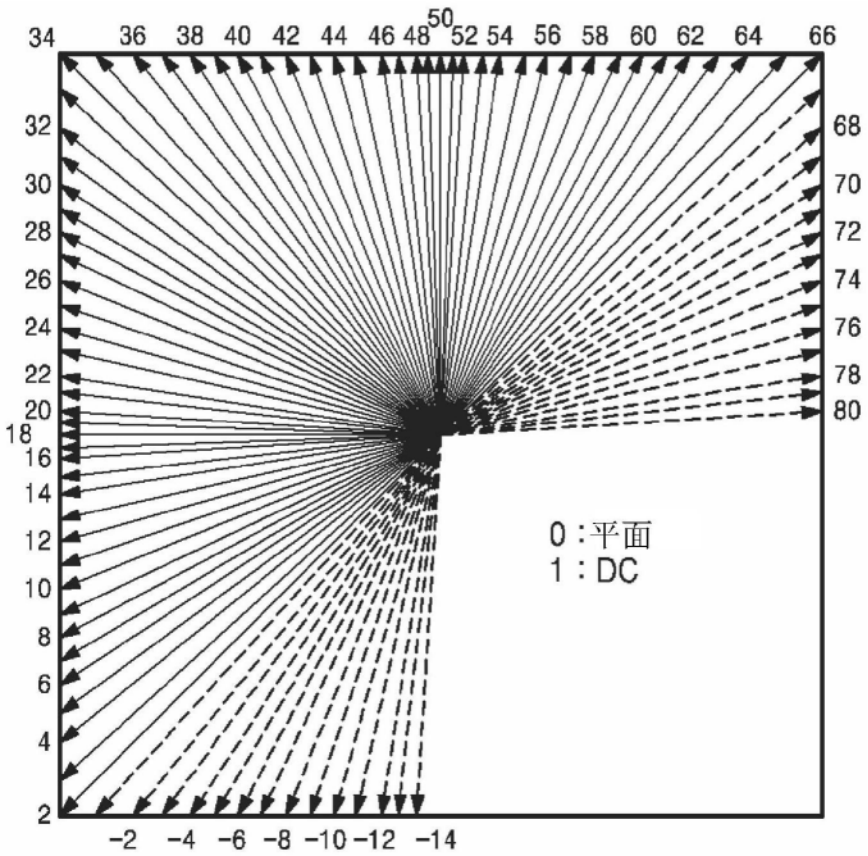

34 36 38 40 42 44 46 48 50 52 54 56 58 60 62 64 66
32 30 28 26 24 22 20 18 16 14 12 10 8 6 4 2
68 70 72 74 76 78 80
-2 -4 -6 -8 -10 -12 -14
0：平面
1：DC

图3b

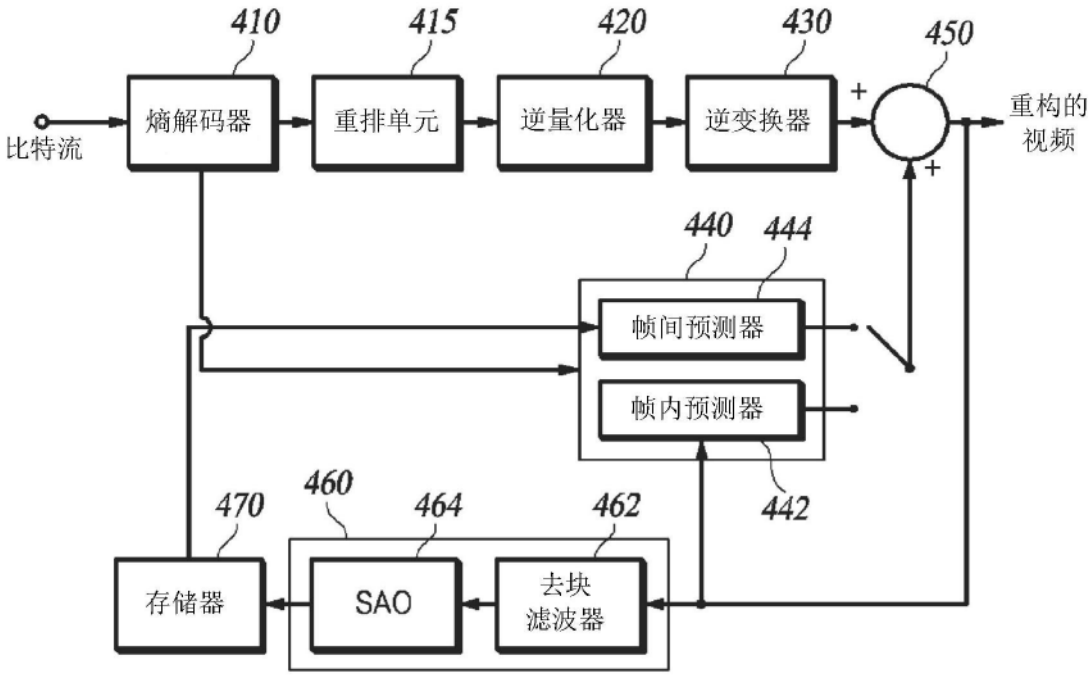

410
415
420
430
450
比特流
熵解码器
重排单元
逆量化器
逆变换器
+
+
重构的视频
440
444
帧间预测器
帧内预测器
442
470
460
464
462
存储器
SAO
去块滤波器

图4

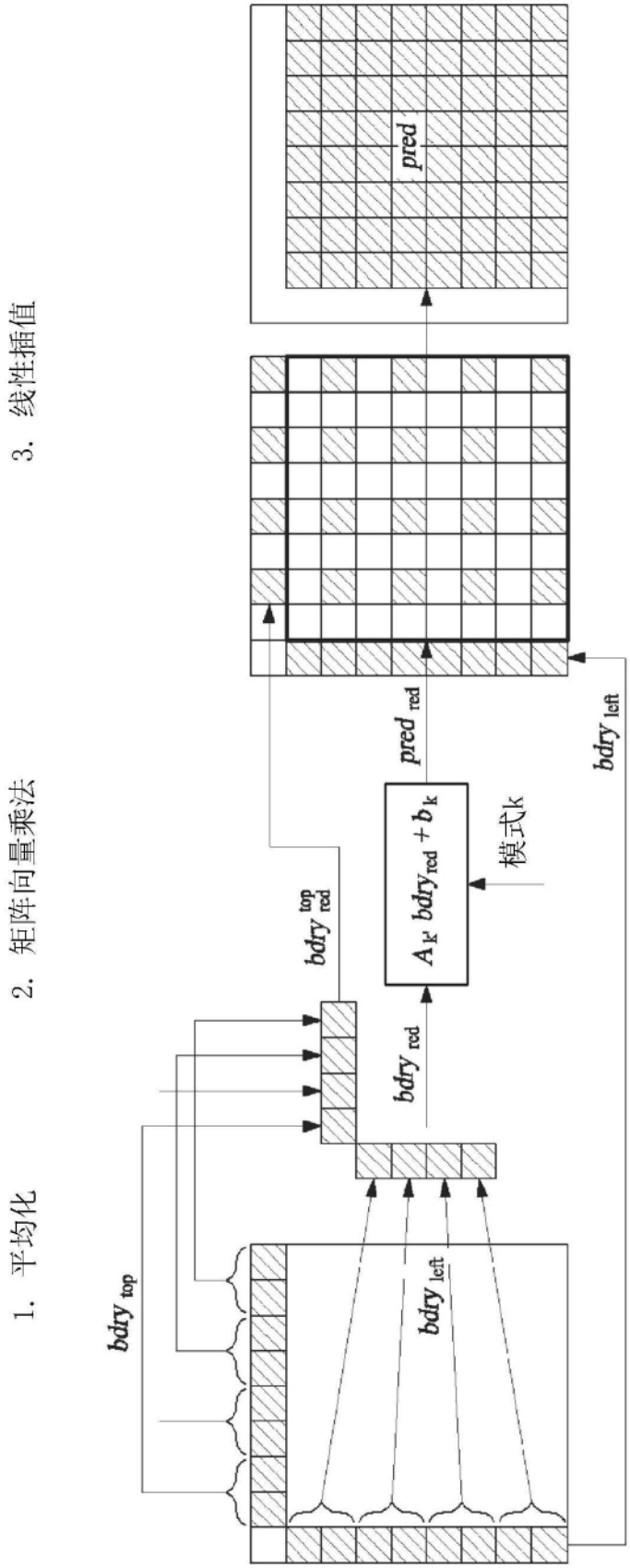

1. 平均化
2. 矩阵向量乘法
3. 线性插值
bdry top
bdry left
bdry red
bdry top red
A k · bdry red + b k
模式k
pred red
pred
bdry left

图5

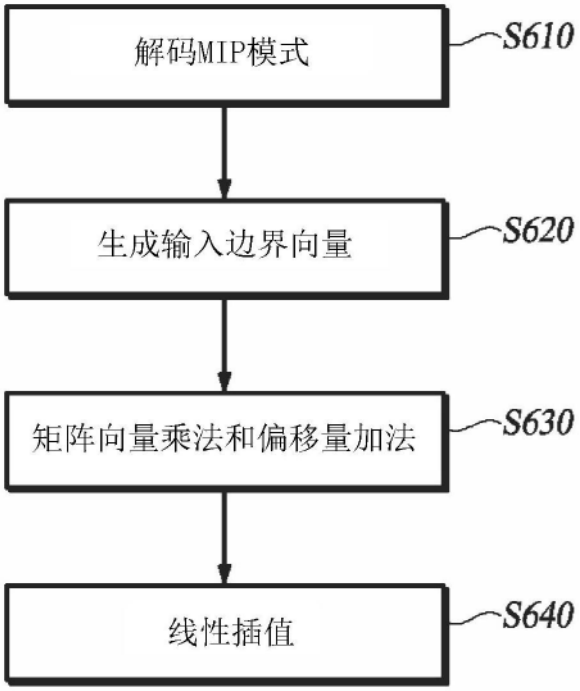

解码MIP模式
S610
生成输入边界向量
S620
矩阵向量乘法和偏移量加法
S630
线性插值
S640

图6

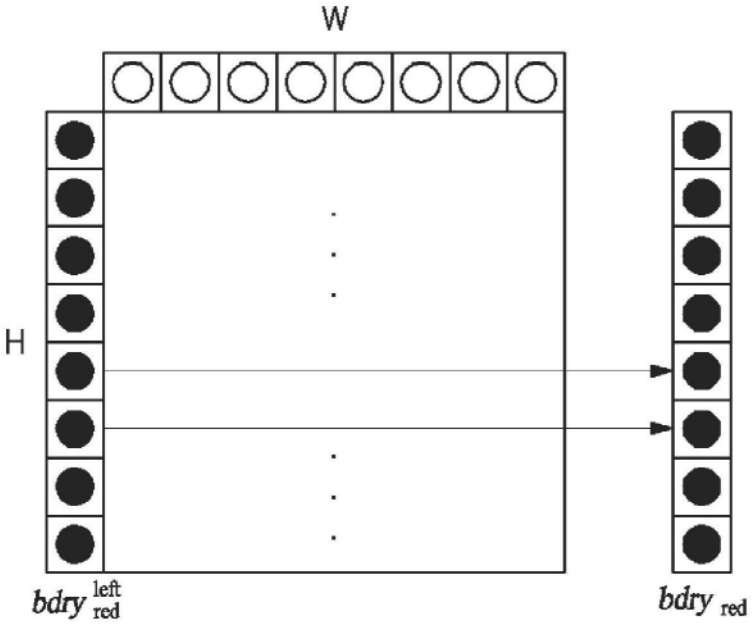

W
H
bdry left red
bdry red

图7a

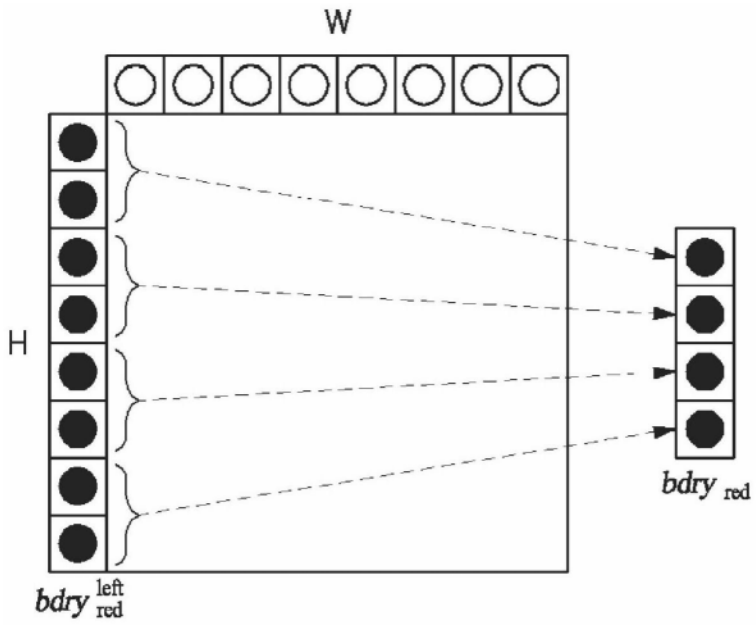


图7b

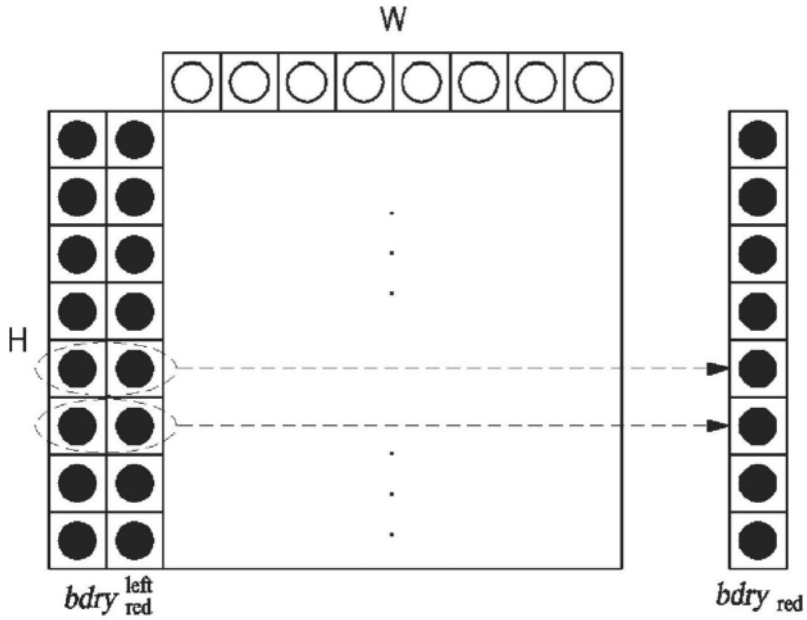


图7c

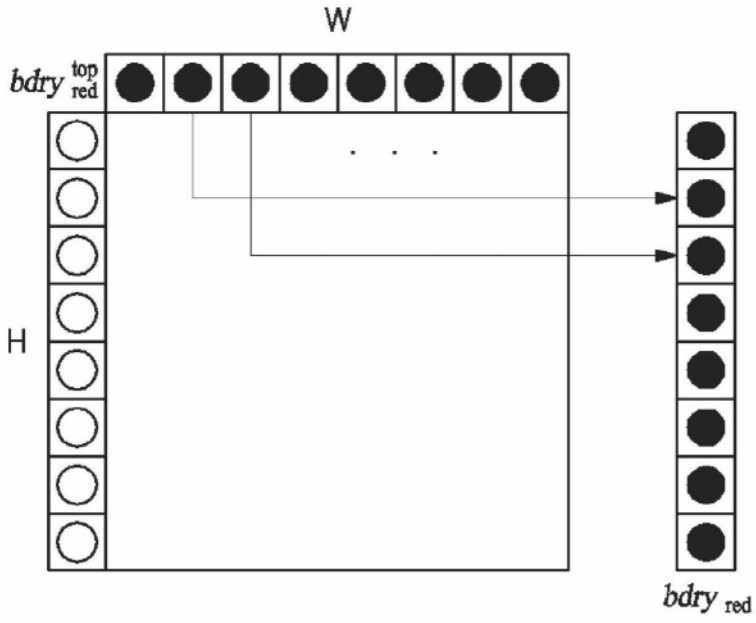


图8a

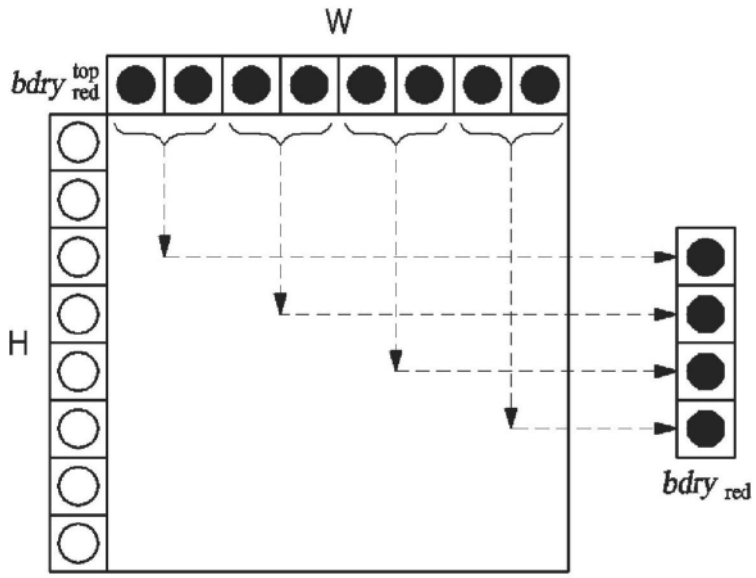


图8b

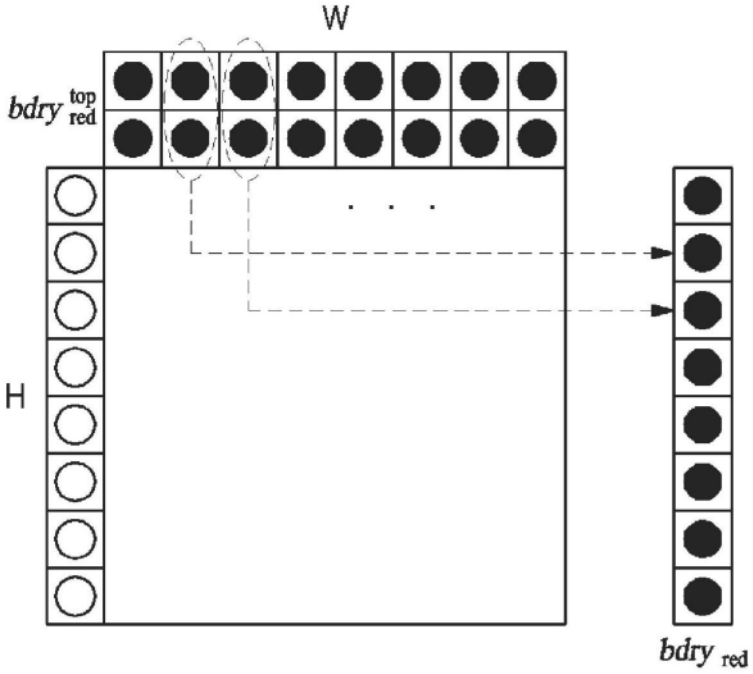


图8c

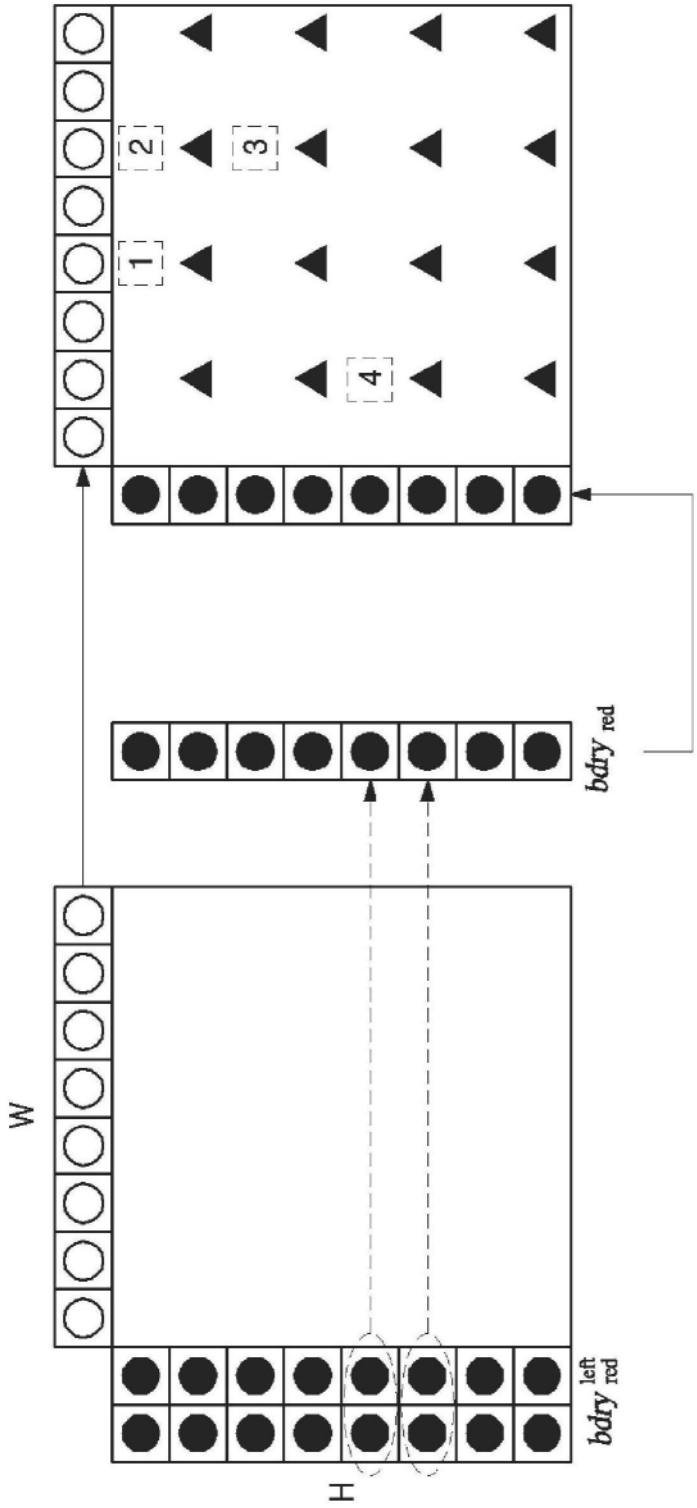


图9

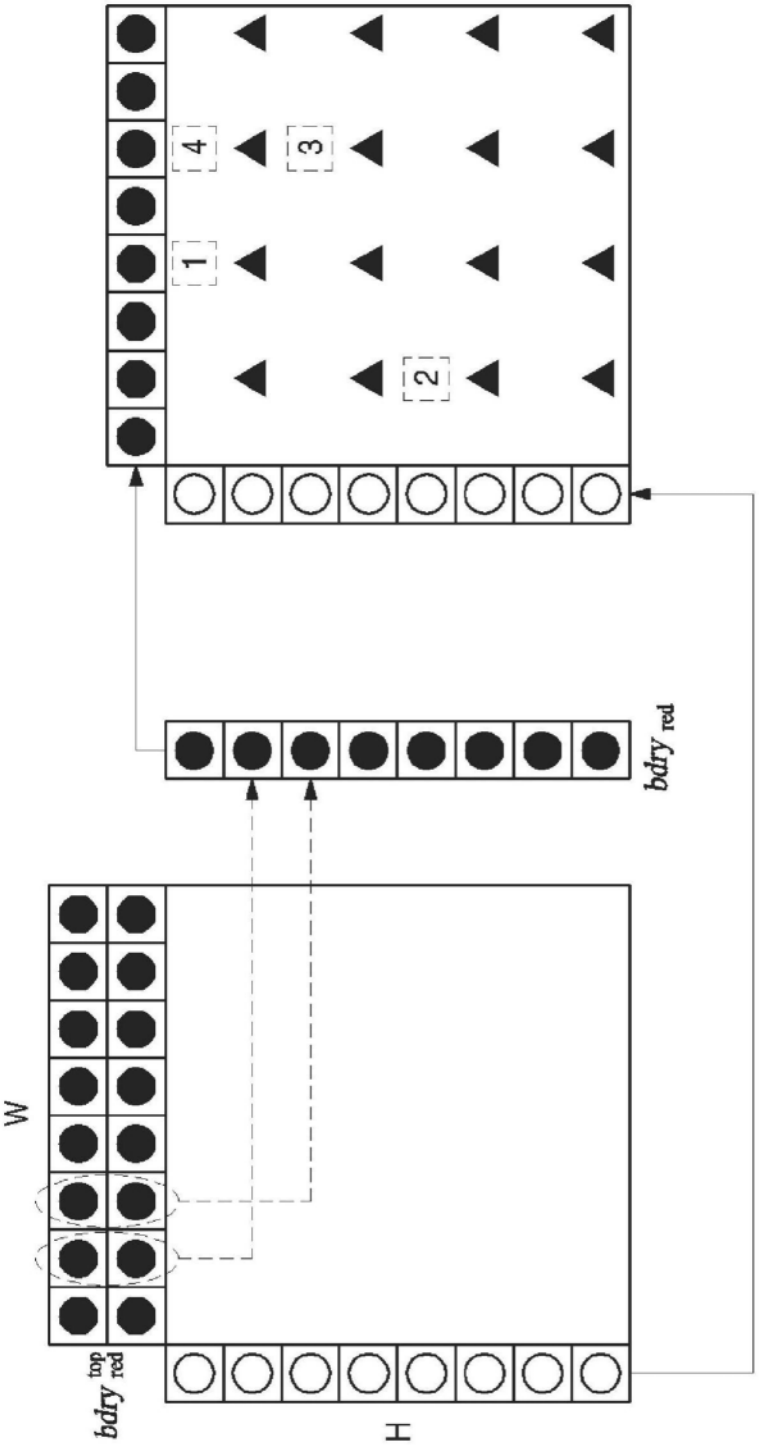


图10

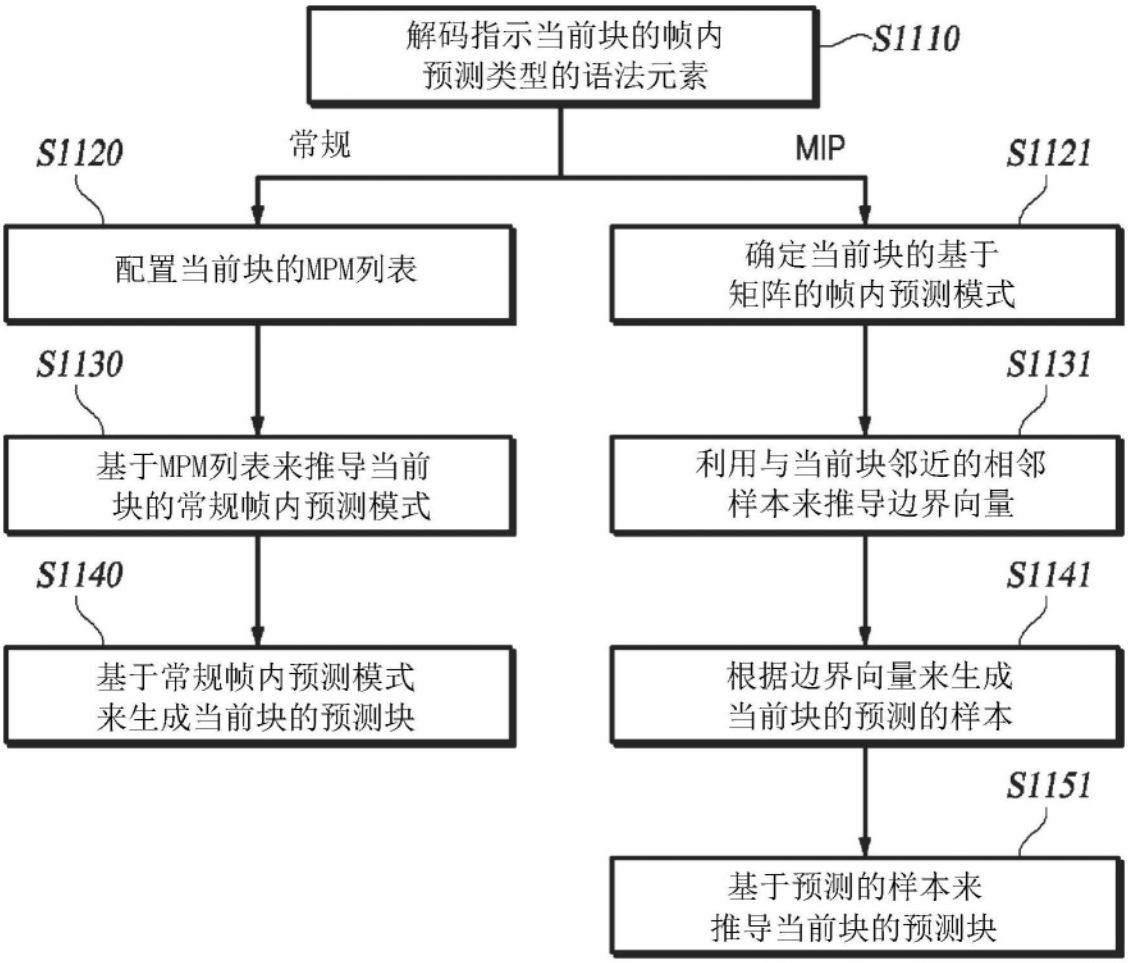

解码指示当前块的帧内预测类型的语法元素
S1110
常规
MIP
S1120
S1121
配置当前块的MPM列表
确定当前块的基于矩阵的帧内预测模式
S1130
S1131
基于MPM列表来推导当前块的常规帧内预测模式
利用与当前块邻近的相邻样本来推导边界向量
S1140
S1141
基于常规帧内预测模式来生成当前块的预测块
根据边界向量来生成当前块的预测的样本
S1151
基于预测的样本来推导当前块的预测块

图11