



(12) 发明专利

(10) 授权公告号 CN 109792516 B

(45) 授权公告日 2021.05.25

(21) 申请号 201780059798.9

(22) 申请日 2017.09.28

(65) 同一申请的已公布的文献号
申请公布号 CN 109792516 A

(43) 申请公布日 2019.05.21

(30) 优先权数据
62/403,735 2016.10.04 US
62/420,659 2016.11.11 US

(85) PCT国际申请进入国家阶段日
2019.03.28

(86) PCT国际申请的申请数据
PCT/CN2017/103851 2017.09.28

(87) PCT国际申请的公布数据
W02018/064948 EN 2018.04.12

(73) 专利权人 联发科技股份有限公司

地址 中国台湾新竹市新竹科学工业园区笃行一路1号

(72) 发明人 夜静 许晓中 刘杉

(74) 专利代理机构 深圳市威世博知识产权代理
事务所(普通合伙) 44280
代理人 何青瓦

(51) Int.Cl.
H04N 19/105 (2006.01)
H04N 19/186 (2006.01)
H04N 19/159 (2006.01)

(56) 对比文件
CN 1784015 A, 2006.06.07
CN 103636203 A, 2014.03.12
US 2012287995 A1, 2012.11.15

审查员 王博

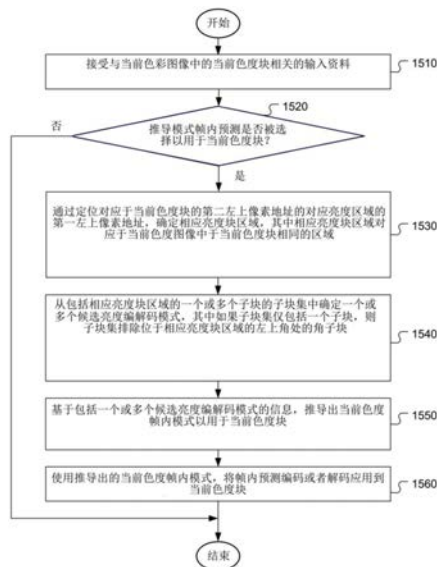
权利要求书2页 说明书15页 附图11页

(54) 发明名称

图像和视频编解码中用于帧内色度编解码的方法及装置

(57) 摘要

本发明提供一种以推导模式帧内预测编解码的色度的视频编解码的方法及装置。如果推导模式帧内预测被选择用于当前色度块,则:通过定位对应于当前色度块相应亮度块区域的第一左上像素地址,确定相应亮度块区域;从包括相应亮度块区域的一个或多个子块的子块集中确定一个或多个候选亮度编解码模式,其中如果子块集仅包括一个子块,则子块集排除位于相应亮度块区域的左上角处的角子块;基于包括一个或多个候选亮度编解码模式的信息,推导出当前色度帧内模式以用于当前色度块;以及使用推导出的当前色度帧内模式,将帧内预测编码或者解码应用到当前色度块。



CN 109792516 B

1. 一种视频编解码方法,其特征在于,分别由视频编码系统和视频解码系统使用,该方法包括:

接收与当前色彩图像中的当前色度块相关的输入数据;

如果推导模式帧内预测被选择用于该当前色度块,则:

通过定位对应于该当前色度块的第二左上像素地址的相应亮度块区域的第一左上像素地址,确定相应亮度块区域,其中该相应亮度块区域对应于该当前色彩图像中与该当前色度块相同的区域;

从包括该相应亮度块区域的一个或多个子块的子块集中确定一个或多个候选亮度编解码模式;

基于包括该一个或多个候选亮度编解码模式的信息,推导出当前色度帧内模式以用于该当前色度块,其中当该相应亮度块区域的右下中间子块(M4)使用帧内块复制(IBC)模式时,使用默认帧内模式作为该当前色度帧内模式;以及

使用推导出的该当前色度帧内模式,将帧内预测编码或者解码应用到该当前色度块。

2. 如权利要求1所述的视频编解码方法,其特征在于,该当前色度块对应于一个色度编码单元,且该相应亮度块区域对应于一个亮度编码单元。

3. 如权利要求1所述的视频编解码方法,其特征在于,该相应亮度块区域的每个子块对应于一个最小亮度编码单元。

4. 如权利要求1所述的视频编解码方法,其特征在于,确定该一个或多个候选亮度编解码模式,包括:

逐个检测与该子块集的n个子块相关的一个或多个亮度编解码模式,直到所检测的亮度编解码模式为帧内角度模式、DC模式或者平面模式;该子块集包括该相应亮度块区域的四个角子块和四个中间子块;

该四个角子块位于该相应亮度块区域的左上角、右上角、左下角和右下角处;

该四个中间子块与该相应亮度块区域的中间位置相邻,且该四个中间子块包括左上中间子块(M1)、右上中间子块(M2)、左下中间子块(M3)和右下中间子块(M4);

以及

使用所检测的该亮度编解码模式作为该当前色度帧内模式;

其中n为小于等于8的正整数。

5. 如权利要求4所述的视频编解码方法,其特征在于,

如果与该子块集的该n个子块相关的该一个或多个亮度编解码模式中没有一个为该帧内角度模式、DC模式或者平面模式,则确定该一个或多个候选亮度编解码模式,包括:

选择默认帧内模式作为该当前色度帧内模式。

6. 如权利要求4所述的视频编解码方法,其特征在于,根据所选择的顺序逐个检测该子块集的该n个子块。

7. 如权利要求6所述的视频编解码方法,其特征在于,该默认帧内模式被发信在序列参数集、图像参数集或者片段头中。

8. 如权利要求6所述的视频编解码方法,其特征在于,该默认帧内模式对应于平面模式、DC模式、垂直模式或者水平模式。

9. 如权利要求6所述的视频编解码方法,其特征在于,
该默认帧内模式自与该当前色度块的一个或多个候选相邻色度块相关的一个或多个候选色度帧内模式而推导出。
10. 如权利要求9所述的视频编解码方法,其特征在于,
该当前色度块的该一个或多个候选相邻色度块包括位于该当前色度块的左上、上方、右上、左侧或者左下处的多个相邻色度块。
11. 如权利要求10所述的视频编解码方法,其特征在于,
该一个或多个候选色度帧内模式中使用最频繁的模式被选择作为该当前色度帧内模式。
12. 如权利要求4所述的视频编解码方法,其特征在于,
如果与该子块集的该n个子块相关的该一个或多个亮度编解码模式中没有一个为该帧内角度模式、DC模式或者平面模式,则该推导模式帧内预测被认为对该当前色度帧内模式不可用。
13. 一种视频编解码装置,其特征在于,分别由视频编码系统和视频解码系统使用,该装置包括一个或多个电子电路或一个或多个处理器,被配置为:
接收与当前色彩图像中的当前色度块相关的输入数据;
如果推导模式帧内预测被选择用于该当前色度块,则:
通过定位对应于该当前色度块的第二左上像素地址的相应亮度块区域的第一左上像素地址,确定相应亮度块区域,其中该相应亮度块区域对应于该当前色彩图像中与该当前色度块相同的区域;
其中当该相应亮度块区域的右下中间子块(M4)使用帧内块复制(IBC)模式时,使用默认帧内模式作为该当前色度帧内模式;以及
使用该当前色度帧内模式,将帧内预测编码或者解码应用到该当前色度块。

图像和视频编解码中用于帧内色度编解码的方法及装置

[0001] 交叉引用

[0002] 本发明主张在2016年10月04日提出的申请号为62/403,735的美国临时专利申请的优先权,主张在2016年11月11日提出的申请号为62/420,659的美国临时专利申请的优先权,其内容以引用方式整体并入本文中。

技术领域

[0003] 本发明涉及图像和视频编解码中的帧内编解码。具体而言,本发明涉及帧内预测模式推导以用于帧内推导模式(derived-mode,DM)编解码的颜色图像或者颜色视频的色度分量,以提高编解码效率。

背景技术

[0004] 高效视频编码(High Efficiency Video Coding,HEVC)标准,是在ITU-T视频编码专家组(Video Coding Experts Group,VCEG)标准组织与运动图像专家组(Moving Picture Experts Group,MPEG)标准组织的联合视频项目下发展的,特别是与视频编码联合小组(Joint Collaborative Team on Video Coding,JCT-VC)存在合作伙伴关系。在HEVC中,一个片段(slice)被分割成多个编码树单元(coding tree unit,CTU)。在主文档中,编码树单元的最小尺寸和最大尺寸由序列参数集(sequence parameter set,SPS)中的语法元素指定。所允许的编码树单元尺寸可以为8x8、16x16、32x32或者64x64。对于每个片段,根据光栅扫描连续处理这个片段内的编码树单元。

[0005] 编码树单元被进一步分割成多个编码单元(coding unit,CU),以适应性各种本地特征。二叉树,表示为编码树,用于将编码树单元分割成多个编码单元。假设编码树单元尺寸为MxM,其中M为64、32或者16的值中的一个。编码树单元可以为单个编码单元(即没有分割),或者可以被分割成四个相同尺寸(即M/2xM/2)的更小的单元,其对应于编码树的节点。如果这些单元为编码树的叶节点,则这些单元变成编码单元。否则,二叉树分割流程可以被重复直到节点的尺寸达到序列参数集中所指定的最小允许编码单元尺寸。

[0006] 此外,根据HEVC,每个编码单元可以被分割成一个或多个预测单元(prediction unit,PU)。结合(couple)编码单元,预测单元用作基础表示块以用于共享预测信息。在每个预测单元内部,相同的预测流程被使用,且基于预测单元,相关信息被发信到解码器。根据预测单元分割类型,编码单元可以被分割成1个、2个或者4个预测单元。HEVC定义了8种形状以用于将编码单元分割成预测单元,其包括2Nx2N分割类型、2NxN分割类型、Nx2N分割类型、NxN分割类型、2NxN分割类型、2NxN分割类型、nLx2N分割类型和nRx2N分割类型。不同于编码单元,预测单元可以仅被分割一次。2NxN分割类型、2NxN分割类型、nLx2N分割类型和nRx2N分割类型的分割对应于非对称分割,其中两个分割的部分具有不同尺寸。

[0007] HEVC编解码包括帧间预测和帧内预测。对于帧内预测,生成帧内预测包括三个部分:帧内平滑滤波、帧内预测和帧内梯度滤波。首先,在计算预测之前,平滑操作被应用到参考样本,作为预处理步骤。这个平滑操作对应于将具有滤波器权重 $[1, 2, 1] \gg 2$ 与低通特性

的FIR (Finite Impulse Response) 滤波器应用到属于当前变换单元(transform unit, TU)的左列和上行的样本。每个变换单元的帧内预测与相邻变换单元的重构样本一起被产生。涉及帧内光滑的样本如图1所示,其中块100对应于当前块,线110对应于水平分界线,线120对应于垂直分界线。是否使用这个光滑操作由变换单元尺寸和帧内预测模式来确定。其次,用某个帧内预测模式,当前块的帧内预测自相邻参考样本推导出,并且帧内预测模式由编码器自DC模式、平面模式和33种方向模式中选择,并被发信在比特流中。再次,如果帧内预测模式为DC模式、水平模式或者垂直模式,则帧内梯度滤波器被进一步应用到位于当前变换单元的左分界线和上分界线处的样本。

[0008] 在HEVC中的所有35种帧内预测模式中,三种模式被考虑成最大概率模式 (most probable mode, MPM), 以用于预测当前预测块内的帧内预测模式。例如,左预测块中和上预测块中使用的帧内预测模式可以作为最大概率模式集的候选。在两个相邻块中的帧内预测模式相同且均为方向的,或者两个相邻块中仅一个可用且以帧内预测编解码,且同时这个帧内预测模式为方向的情况下,紧邻着这个方向的两个相邻方向在最大概率模式集中也被使用。DC模式和平面模式也被考虑在最大概率模式集中,以填充最大概率模式中可用点 (available spot), 特别是如果上相邻块或者顶端相邻块不可用或者不以帧内预测编解码,或者相邻块中的帧内预测模式不是方向的。如果用于当前预测块的帧内预测模式为最大概率模式集中的模式中的一个,则1个二进制代码 (bin) 或者2个二进制代码用于发信其为哪个模式。否则,其与最大概率模式集中的任何条目不相同,则其被编解码成非最大概率模式模式。总共存在32种这样非最大概率模式模式,并且 (5码元) 固定长度编解码方法被应用到发信这个模式。33个方向如图2所示。在图2中,总共存在33种方向模式,即H, H+1, ..., H+8, H-1, ..., H-7, V, V+1, ..., V+8, V-1, ..., V-8。这个系统可以被扩展成常规情况,其中水平模式和垂直模式被表示为H模式和V模式。对于其他方向模式,其可以被表示为H+k模式或者V+k模式,其中k=1, 2等。例如,如图3所示,如果65种方向模式被使用,k可以从1到16、或从-1到-16变化。

[0009] 在HEVC的一些最近发展中,如图3所示,额外的32种方向模式被使用在已有的33种方向模式之间。在这种情况下,存在总共65种模式,包括方向模式和一些非方向模式。

[0010] 在HEVC中,一旦方向模式被确定,沿着预测方向,当前块中所有像素将使用相同的预测子值。如果预测子落入两个重构参考样本之间,则双线性滤波器将被使用以计算预测子作为两个相邻像素的加权平均。例如,根据 $P = [P1 \cdot a + P2 \cdot (32 - a)] / 32$, 预测子发信P可以被推导出,其中P1和P2为两个相邻重构样本,整数a为从预测子P到P2的距离,其范围为0与32之间且包括端点。

[0011] 帧内梯度滤波器的概念将沿着帧内预测方向使用梯度信息来提高帧内预测质量。对于如图2所示的从垂直方向/水平方向 (即v/h) 到垂直+8方向/水平+8方向 (即v+8/h+8) 的帧内预测模式,左侧列相邻样本/上方行相邻样本可以沿着来自于上方行/左侧列的帧内预测方向而放置其相应的参考。用相邻样本计算的梯度可以用于改善帧内预测。用于垂直方向模式的示例如图4A所示,其中 P_{ij} 表示位于i列和j行的预测子。AL表示位于当前块的左上角处的重构样本,而 L_i 表示当前块的左侧列中的重构样本。新的预测子被计算成:

$$[0012] \quad P'_{ij} = P_{ij} + \alpha \cdot (L_i - AL), \quad (1)$$

[0013] 其中 α 为从0到1的小数,且根据j而选择,例如,对于j=0, $\alpha = 1/2$, 对于j=1, $\alpha = 1/$

4. P'_{ij} 被用作最终预测子。对于水平方向模式, 最终预测子 P'_{ij} 被计算成

$$[0014] \quad P'_{ij} = P_{ij+\alpha} \cdot (A_j - AL), \quad (2)$$

[0015] 在上述等式中, A_j 为上方列中的重构样本, 其如图4A所示。对应方向模式 $v+1, \dots, v+8$ 和方向模式 $h+1, \dots, h+8$, L_i 或者 A_j 首先沿着帧内预测的方向获得其相应的参考样本 RL_i 或者 RA_j 。当 RL_i 或者 RA_j 不位于整数像素的位置处时, 其由当前块的上方行或者左侧列中的整数像素的插值来产生。 $v+1$ 方向模式, $\dots, v+8$ 方向模式的示例如图4B所示。最终预测子 P'_{ij} 被计算成

$$[0016] \quad P'_{ij} = P_{ij+\alpha} \cdot (L_i - RL_i). \quad (3)$$

[0017] 同理于垂直方向模式, α 为从0到1的小数, 且根据帧内预测的方向和 j 而选择。对于 $h+1, \dots, h+8$ 方向模式, 最终预测子 P'_{ij} 被计算成

$$[0018] \quad P'_{ij} = P_{ij+\alpha} \cdot (A_j - RA_j), \quad (4)$$

[0019] 其中, α 为从0到1的小数, 且根据帧内预测的方向和 i 而选择。

[0020] 帧内梯度滤波器可以被应用于HEVC中的所有方向模式, 即 $v+1, \dots, v+8$ 和 $h+1, \dots, h+8$ 。但是, 仅当帧内预测模式为DC、水平模式或者垂直模式时, 帧内梯度滤波器被使用。如果帧内预测为DC模式, 则位于第一列和第一行处的样本由帧内梯度滤波器滤波。如果帧内预测为水平模式, 则位于第一行的样本由帧内梯度滤波器滤波。如果帧内预测为垂直模式, 则位于第一列的样本由帧内梯度滤波器进一步滤波。

[0021] 除了帧内梯度滤波, 称为双向帧内预测的另一种方法在JCT-VC会议中也被提出以改善帧内预测的质量。对于对角帧内预测模式, 即 $v+1, \dots, v+8$ 和 $h+1, \dots, h+8$, 上方行的重构样本与来自于左侧列沿着方向的重构样本的加权和用作帧内预测子。例如, 如图5所示, 对于 $v+1, \dots, v+8$ 方向模式, 来自于上方列的相邻样本的 P_{ij} 在左侧行沿着预测方向具有相应的参考样本 F_{ij} 。如果 F_{ij} 不位于整数像素位置处, 则其可以通过插值左侧列中的整数像素而产生。然后, 最终预测子被计算成 P_{ij} 与 F_{ij} 的加权和, 如下:

$$[0022] \quad P'_{ij} = \alpha \cdot P_{ij} + (1-\alpha) \cdot F_{ij} \quad (5)$$

[0023] 其中, α 为从0到1的小数, 且根据帧内预测的方向和 j 而选择 (对于 $v+1, \dots, v+8$ 方向模式) 或者根据帧内预测的方向和 i 而选择 (对于 $h+1, \dots, h+8$ 方向模式)。

[0024] 在产生帧内预测子之后, 预测误差由变换和量化来进一步处理, 并由熵编解码来编码。对于熵编解码, 首先, 量化系数被分割成 4×4 个系数组。不同系数组的编解码顺序和一个系数组中系数的扫描顺序根据帧内预测模式和变换尺寸而选择。如果变换尺寸小于或者等于 8×8 , 则基于帧内模式扫描将用于不同系数组的编解码顺序和一个系数组中的系数的扫描顺序。否则, 对角扫描用于不同系数组的编解码顺序和一个系数组中的系数的扫描顺序。

[0025] 同时, 可能用几个预测子的加权和来产生用于帧内预测的最终预测发信 (即多参数帧内预测或者称为MPI)。位置 (i, j) 的最终预测子 $P_{MPI}[i, j]$ 被定义成如下:

$$[0026] \quad P_{MPI}[i, j] = (\alpha P_{HEVC}[i, j] + \beta P_{MPI}[i-1, j] + \gamma P_{MPI}[i, j-1] + \delta P_{MPI}[i-1, j-1] + 4) \gg 3,$$

[0027] 其中如图6所示, 块的外部 $P_{MPI}[i, j]$ 等于重构发信, 如果 $i < 0 \parallel j < 0$, 则 $P_{MPI}[i, j] = REC[i, j]$ 。

[0028] 图6示出了多参数帧内预测 (multiple parameter Intra prediction, MPI) 流程的示例, 其中输入块由任意方向帧内 (Arbitrary Directional Intra, ADI) 610 处理, 其后

为多参数帧内预测620。这个后处理的强度(即参数 $\alpha+\beta+\gamma+\delta=8$)被控制在编码单元层,并用2个以上码元发信。

[0029] 在ITU-T稿件C1046(A.Said,etc.,“Position dependent Intra prediction combination,”ITU-T SG16 COM 16-C1046-E,Oct.2015)中,如图7所示,一种方法被提出以使用已滤波参考样本和未滤波参考样本的结合来形成最终预测子 $p[x,y]$,以用于未滤波情况(即710)和已滤波情况(即720)。

[0030] 发信 r 和发信 s 用于表示具有已滤波参考和未滤波参考的序列。新的预测 $p[x,y]$ 结合分界线元素 $r[\]$ 和分界线元素 $q[x,y]$ 的加权值(即自己滤波样本 $s[\]$ 推导出的预测子),如下:

$$p[x,y] = \left\{ \left(c_1^{(v)} \gg [y/d] \right) r[x,-1] - \left(c_2^{(v)} \gg [y/d] \right) r[-1,-1] \right. \\ \left. + \left(c_1^{(h)} \gg [x/d] \right) r[-1,y] - \left(c_2^{(h)} \gg [x/d] \right) r[-1,-1] \right. \\ \left. + b[x,y]q[x,y] + 64 \right\} \gg 7$$

[0032] 其中, $c_1^{(v)}$, $c_2^{(v)}$, $c_1^{(h)}$, $c_2^{(h)}$ 为存储的预测参数, $d=1$ 用于块尺寸为 16×16 , $d=2$ 用于更大块,并且

$$b[x,y] = 128 - \left(c_1^{(v)} \gg [y/d] \right) + \left(c_2^{(v)} \gg [y/d] \right) - \left(c_1^{(h)} \gg [y/d] \right) + \left(c_2^{(h)} \gg [y/d] \right)$$

[0034] 为归一化因子。

[0035] 在稿件JVET-C-0061(X.Xiu,etc.,“Decoder-side Intra mode derivation”,JVET-C0061,May,2016)中,公开了用于使用平面模式的帧内预测的插值。根据JVET-C-0061,使用相应的左侧参考样本和上方参考样本的线性平均,位于右下角当前预测块的样本被发信或者估计。因此,如图8所示,使用上方/右下样本组合和左侧/右下样本组合,最右侧列和下方行中的样本被双线性插值(即,示例810所示)。如图8所示,使用相似的双线性插值,预测块中剩余像素被预测(即示例820所示)。

[0036] 四叉树加二叉树(Quadtree Plus Binary Tree,QTBT)结构

[0037] 在稿件m37524/COM16-C966(J.An,et al.,“Block partitioning structure for next generation video coding,”MPEG doc.m37524 and ITU-T SG16 Doc.COM16-C966,Oct.2015)中,公开了QTBT块分割结构。根据QTBT,编码树块(coding tree block,CTB)首先由四叉树结构分割。四叉树叶节点进一步由二叉树结构分割。二叉树叶节点,即编码块(coding block,CB),用于预测和变换,而无需进一步分割。对于P片段和B片段,一个编码树单元(coding tree unit,CTU)中的亮度编码树块和色度编码树块共享相同的QTBT结构。对于I片段,亮度编码树块由一QTBT结构分割成编码块,两个色度编码树块由另一QTBT结构分割成色度编码块。

[0038] 编码树单元(或者用于I片段的编码树块),其为四叉树的根节点,首先由四叉树来分割,其中一个节点的四叉树分割可以被重复直到该节点达到最小允许四叉树叶节点尺寸(即MinQTSIZE)。如果四叉树叶节点尺寸不大于最大允许二叉树根节点尺寸(即MaxBTSIZE),则其可以进一步由二叉树来分割。一个节点的二叉树分割可以被重复直到这个节点达到最小允许二叉树叶节点尺寸(即MinBTSIZE)或者最大允许二叉树深度(即

MaxBTDepth)。二叉树叶节点,即编码单元(或者用于I片段的编码块),将用于预测(即帧内图像预测或者帧间图像预测)和变换,而无需进一步分割。在二叉树分割中存在两个分割类型:对称水平分割和对称垂直分割。

[0039] 图9的块分割910和相应的QTBT结构920示出了通过使用QTBT的块分割示例。实线表示四叉树分割,虚线表示二叉树分割。在二叉树的每个分割节点中,一个标志被发信以指示哪个分割类型(即水平的或者垂直的)被使用,其中0表示水平分割,1表示垂直分割。对于四叉树分割,无需表示分割类型,由于其总是将块水平并垂直地分割成4个相等尺寸的子块。

[0040] 在上述中,联合视频探索队(joint video exploration team, JVET)为国际组织,其由ITU-T VCEG和ISO/IEC MPEG建立,以研究下一代视频编解码技术。称为联合探索模型(joint exploration model, JEM)的参考软件基于HEVC的参考软件(HEVC's reference software, HM)而建立。一些新视频编解码方法,包括QTBT和65种帧内预测方向,被包含在JEM软件中。

[0041] 在QTBT结构中,对于I片段,当使用推导模式(自亮度而推导出的模式)而编解码色度编码单元,编码单元将使用左上色度像素来查找相应的亮度像素。因此,包含亮度像素的亮度块的帧内模式用作色度帧内模式。

[0042] 图10示出了用于色度块的推导模式(derived mode, DM)的示例。在图10中,块1020为自色度编码树块1022分割的色度块。块1010为亮度编码树块1012内相应亮度块区域。黑像素1014表示相应亮度块区域的左上角像素。由于亮度分量和色度分量可以使用单独的块分割,可以不存在同位亮度块(例如,亮度编码块)以用于给定的色度块(例如色度编码块)。因此,色度编码单元的左上像素1024将用于基于相应的亮度编码单元的左上像素1014而查找相应的亮度编码单元区域。相应的亮度编码单元区域1010覆盖在亮度图像中与色度块1020同等的亮度像素区域。相应的亮度编码单元区域1010可以对应于一个或多个亮度编码单元或者部分亮度编码单元。与位于左上角位置处的最小编码单元(即最小编码单元1016)相关的亮度帧内模式用作色度帧内模式。在图10中,每个较小正方形对应一个最小编码单元,其是用于编码或者解码的基础单元。一个编码单元可以由1到M个最小编码单元形成,其中 $M = (\text{最大编码单元宽度} / \text{最小编码单元宽度}) \times (\text{最大编码单元高度} / \text{最小编码单元高度})$ 。

[0043] 根据传统的方法,当推导模式用于色度块时,由相应的亮度编码单元区域的左上最小编码单元使用的亮度帧内模式用作相关色度编码单元的帧内模式。但是,左上最小编码单元可以不是帧内模式编解码的,色度推导模式帧内预测不能被使用。因此,推导模式编解码的色度块可能不具有将使用的有效的亮度帧内模式。本发明的目标之一在于解决这个问题。

发明内容

[0044] 本发明公开了视频编解码色度推导模式帧内预测的方法及装置。根据一个方法,如果推导模式帧内预测被选择用于当前色度块,则:通过定位对应于当前色度块的第二左上像素地址的相应亮度块区域的第一左上像素地址,确定相应亮度块区域,其中相应亮度块区域对应于当前色彩图像中与当前色度块相同的区域;从包括相应亮度块区域的一个或

多个子块的子块集中确定一个或多个候选亮度编解码模式,其中如果子块集仅包括一个子块,则子块集排除位于相应亮度块区域的左上角处的角子块;基于包括一个或多个候选亮度编解码模式的信息,推导出当前色度帧内模式以用于当前色度块;以及使用推导出的当前色度帧内模式,将帧内预测编码或者解码应用到当前色度块。

[0045] 当前色度块可以对应于一个色度编码单元,且相应亮度块区域对应于一个亮度编码单元。相应亮度块区域的每个子块可以对应于一个最小亮度编码单元。

[0046] 确定一个或多个候选亮度编解码模式的步骤包括:逐个检测与子块集的 n 个子块相关的一个或多个亮度编解码模式,直到所检测的亮度编解码模式为帧内角度模式、DC模式或者平面模式,并使用所检测的亮度编解码模式作为当前色度帧内模式,其中 n 为正整数。可以根据所选择的顺序逐个检测子块集的 n 个子块。在一个实施例中,如果与子块集的 n 个子块相关的一个或多个亮度编解码模式中没有一个为帧内角度模式、DC模式或者平面模式,则确定一个或多个候选亮度编解码模式的步骤包括:选择默认帧内模式作为当前色度帧内模式。默认帧内模式被发信在序列参数集、图像参数集或者片段头中。默认帧内模式可以对应于平面模式、DC模式、垂直模式或者水平模式。默认帧内模式也可以自与当前色度块的一个或多个候选相邻色度块相关的一个或多个候选色度帧内模式而推导出。例如,当前色度块的候选相邻色度块包括位于当前色度块的左上、上方、右上、左侧或者左下处的多个相邻色度块。一个或多个候选色度帧内模式中使用最频繁的模式可以被选择作为当前色度帧内模式。在另一实施例中,如果与子块集的 n 个子块相关的一个或多个亮度编解码模式中没有一个为帧内角度模式、DC模式或者平面模式,则推导模式帧内预测被考虑成对当前色度帧内模式不可用。

[0047] 在一个实施例中,子块集包括位于左上角、右上角、左下角和右下角处的四个角子块。在另一实施例中,子块集包括与相应亮度块区域的中间位置相邻的四个中间子块,并且四个中间子块包括左上中间子块(M1)、右上中间子块(M2)、左下中间子块(M3)和右下中间子块(M4)。在又一实施例中,子块集包括相应亮度块区域的四个角子块和四个中间子块。

附图说明

[0048] 图1是根据HEVC的涉及到变换单元(transform unit, TU)的帧内光滑的样本。

[0049] 图2是根据HEVC的用于帧内预测的33种方向,其中这些方向被设计为 $H, H+1, \dots, H+8, H-1, \dots, H-7, V, V+1, \dots, V+8, V-1, \dots, V-8$ 。

[0050] 图3是通过在已有33种方向模式之间添加32中方向模式的用于帧内预测的65种方向的示例。

[0051] 图4A是用于帧内预测的垂直方向模式的示例。

[0052] 图4B是根据HEVC的从 $(v+1)$ 到 $(v+8)$ 用于帧内预测的示例帧内梯度滤波器。

[0053] 图5是根据HEVC的从 $(v+1)$ 到 $(v+8)$ 用于帧内预测的示例双向预测滤波器。

[0054] 图6是多参数帧内预测流程的示例。

[0055] 图7是用于未滤波情况和已滤波情况的 4×4 块中的帧内预测示例。

[0056] 图8是最右侧列和下方行的双线性插值(左边)与剩余样本的双线性插值(右边)的示例。

[0057] 图9是通过使用QTBT的块分割的示例,其中块分割如左边所示,相应的QTBT结构如

右边所示。

[0058] 图10是用于基于相应亮度块区域的左上角最小编码单元的帧内模式的色度块的推导模式的示例。

[0059] 图11是位于相应亮度块区域的四个角处的四个亮度最小块(即TL, TR, BL和BR)的示例。

[0060] 图12是相应亮度块区域中的四个中间亮度最小块(即M1, M2, M3和M4)。

[0061] 图13是用于推导默认帧内模式的当前色度块的相邻块的示例。

[0062] 图14是根据本发明实施例的用于二叉树加四叉树块分割的适应性运动向量分辨率(adaptive motion vector resolution, AMVR)的发信示例。

[0063] 图15是根据本发明实施例的用于色度块的推导模式帧内预测的示例性编解码系统的流程图。

具体实施方式

[0064] 以下描述是实现本发明的最佳实施方式。这一描述是为了说明本发明的一般原理, 而不应被认为是限制性的。本发明的范围最好通过权利要求的范围来确定。

[0065] 如上所述, 在基于HEVC参考软件的称为JEM的参考软件中, 通过复制相应亮度编码单元的帧内模式, 色度编码单元可以使用推导模式(来自于相应亮度分量的推导模式)。然而, 可能的是, 相应亮度块为非帧内编解码的, 例如IBC模式、调色板模式(palette mode)、帧间模式或者除了DC模式、平面模式或者角度模式之外的模式。因此, 这将使得色度帧内模式未定义。因此, 在本发明中公开了克服这个问题的各种方法。

[0066] 用于相应非帧内模式亮度块的色度推导模式分配

[0067] 在本发明中, 当相应亮度块为非帧内编解码的时, 色度推导模式被指定。非帧内模式指的是除了角度模式、DC模式或者平面模式之外的任何模式。因此, 本发明能处理一种情况, 其为色度平面和亮度平面被单独编解码且具有不同块分割结构, 并且相应亮度块可以为非帧内编解码的, 例如IBC模式、调色板模式、PCM模式、帧间模式等。

[0068] 根据本方法, 用于色度块的相应亮度块区域将先基于左上色度像素地址而识别。“相应亮度块区域”的意义已在本发明的背景技术部分中描述。这个亮度块地址将被处理为亮度块的起始。当视频颜色格式为4:2:0时, 这个亮度块的宽度为 $2 \times \text{width_chroma}$, 这个亮度编码单元的高度为 $2 \times \text{height_chroma}$, 其中 width_chroma 为当前色度的宽度, height_chroma 为当前色度的高度。换言之, 当视频颜色格式为4:2:0时, 相应亮度块区域是色度块的两倍宽和两倍高。当视频颜色格式为4:2:2时, 这个亮度块的宽度为 $2 \times \text{width_chroma}$, 这个亮度块的高度等于 height_chroma 。换言之, 当视频颜色格式为4:2:2时, 相应亮度块区域是色度块的两倍宽, 且与色度块相同高度。当视频颜色格式为4:4:4时, 这个亮度块的宽度等于 width_chroma , 这个亮度块的高度等于 height_chroma 。换言之, 当视频颜色格式为4:4:4时, 相应亮度块区域具有与色度块相同的宽度和高度。

[0069] 对于每个相应亮度块区域(例如亮度编码单元/预测单元/变换单元), 存在四个角亮度最小块, 包括左上最小亮度块(即TL)、右上最小亮度块(即TR)、左下最小亮度块(即BL)和右下最小亮度块(即BR)。在本发明中, 最小块也称为子块。图11示出了位于四个角处的四个亮度最小块的示例, 其中块1110对应于亮度编码树块、块1120对应于亮度编码单元区域、

亮度预测单元区域或者亮度变换单元区域。

[0070] 如图12所示,存在由M1,M2,M3和M4表示的四个中间最小块(即子块)。左上中间最小块被设计为M1,右上最小块被设计为M2,左下最小块被设计为M3,右下最小块被设计为M4。每行存在N个最小块,每列存在K个最小块,以用于相应亮度块,其中根据如下确定N和K:

[0071] $N = \text{块宽度} / \text{最小块宽度}$,以及

[0072] $K = \text{块高度} / \text{最小块高度}$ 。

[0073] M1为最小块,其自编码块上分界线具有 $(N/2-1)$ 个最小块的距离,且自编码块左分界线具有 $(M/2-1)$ 个最小块的距离。

[0074] M2为最小块,其自编码块上分界线具有 $N/2$ 个最小块的距离,且自编码块左分界线具有 $(M/2-1)$ 个最小块的距离。

[0075] M3为最小块,其自编码块上分界线具有 $(N/2-1)$ 个最小块的距离,且自编码块左分界线具有 $M/2$ 个最小块的距离。

[0076] M4为最小块,其自编码块上分界线具有 $N/2$ 个最小块的距离,且自编码块左分界线具有 $M/2$ 个最小块的距离。

[0077] 在一个方法中,编码器和解码器将检测8个位置(即TL,TR,BL,BR,M1,M2,M3,M4)中的n个位置,其中n可以为1,2,3,4,5,6,7,8。在n被确定之后,编码器和解码器可以从这8个位置中选择任何n个位置。例如,如果n为4,则编码器、解码器可以选择对应于TL,TR,BL和BR的四个位置。可选地,编码器和解码器可以选择对应于M1,M2,M3和M4的四个位置。编码器和解码器可以逐个检测这n个位置,直到其中一个为帧内角度模式、DC模式或者平面模式编解码的。如果这些模式中没有一个为角度模式、DC模式或者平面模式,则根据本发明,默认帧内模式被填充以用于这个色度块。如本发明的后续部分中所描述,默认帧内模式可以被指定。选择这n个模式的顺序可以有很多变形。

[0078] 在另一实施例中,当色度帧内模式为推导模式时,编码器和解码器将逐个检测TL位置、TR位置、BL位置、BR位置、M1位置、M2位置、M3位置、M4位置,直到其中一个为帧内角度模式、DC模式或者平面模式编解码的。相应亮度块的帧内模式将用于色度块。如果其中没有为帧内角度模式、DC模式或者平面模式编解码的,则默认帧内模式将用于色度块。默认帧内模式将在本发明的后续部分中进行描述。注意的是,这些块的检测顺序可以有很多种变形。例如,编码器和解码器可以依次检测TL位置、M1位置、M2位置、M3位置、M4位置、TR位置、BL位置和BR位置。又例如,编码器和解码器可以依次检测M4位置、TL位置、TR位置、BL位置、BR位置、M3位置、M2位置和M1位置。

[0079] 在又一实施例中,当色度帧内模式为推导模式时,编码器和解码器将逐个检测TL位置、M2位置、M3位置、M4位置,直到其中一个为帧内角度模式、DC模式或者平面模式编解码的。相应亮度块的帧内模式将用于色度块。如果其中没有为帧内角度模式、DC模式或者平面模式编解码的,则默认帧内模式将用于色度块。默认帧内模式将在本发明的后续部分中进行描述。注意的是,这些块的检测顺序可以有很多种变形。例如,编码器和解码器可以依次检测TL位置、M4位置、M2位置、M3位置。又例如,编码器和解码器可以依次检测M4位置、TL位置、M3位置、M2位置。

[0080] 在又一实施例中,当色度帧内模式为推导模式时,编码器和解码器将逐个检测TL位置、TR位置、BR位置和M4位置,直到其中一个为帧内角度模式、DC模式或者平面模式编解

码的。相应块的帧内模式将用于色度块。如果没有为帧内角度模式、DC模式或者平面模式编解码的，则默认帧内模式将用于色度块。默认帧内模式将在本发明的后续部分中进行描述。注意的是，这些块的检测顺序可以有很多种变形。

[0081] 在又一方法中，当色度帧内模式为推导模式时，编码器和解码器将检测8个位置（即TL, TR, BL, BR, M1, M2, M3, M4）中的n个位置，其中n可以为1, 2, 3, 4, 5, 6, 7或者8。在n被确定之后，编码器和解码器可以从这8个位置中选择任何n个位置。例如，如果n为4，则编码器、解码器可以选择对应于TL, TR, BL和BR的四个位置。可选地，编码器和解码器可以选择对应于M1, M2, M3和M4的四个位置。编码器和解码器可以逐个检测所有这n个位置。编码器和解码器可以选择使用最频繁的亮度帧内模式以用于推导模式编解码的色度块。当这些位置中没有一个是帧内角度模式、DC模式或者平面模式编解码的时，默认帧内模式被填充以用于这个色度块。默认帧内模式将在本发明的后续部分中进行描述。这n个模式的顺序可以具有很多变形。

[0082] 在又一方法中，当色度帧内模式为推导模式时，编码器和解码器将检测8个位置（即TL, TR, BL, BR, M1, M2, M3, M4）中的n个位置，其中n可以为1, 2, 3, 4, 5, 6, 7或者8。在n被确定之后，编码器和解码器可以从这8个位置中选择任何n个位置。例如，如果n为4，则编码器、解码器可以选择对应于TL, TR, BL和BR的四个位置。可选地，编码器和解码器可以选择对应于M1, M2, M3和M4的四个位置。编码器和解码器可以逐个检测这n个位置，直到其中一个为帧内角度模式、DC模式或者平面模式编解码的。如果这些位置中没有一个是帧内角度模式、DC模式或者平面模式编解码的，则推导模式被处理为对当前色度块不可用。色度帧内模式列表可以被缩短1，并且相应的码字将被改变。这n个模式的顺序可以具有很多变形。

[0083] 在又一方法中，当色度帧内模式为推导模式时，编码器和解码器将检测8个位置（即TL, TR, BL, BR, M1, M2, M3, M4）中的n个位置，其中n可以为1, 2, 3, 4, 5, 6, 7或者8。在n被确定之后，编码器和解码器可以从这8个位置中选择任何n个位置。例如，当n为4时，则编码器、解码器可以选择对应于TL, TR, BL和BR的四个位置。可选地，编码器和解码器可以选择对应于M1, M2, M3和M4的四个位置。编码器和解码器可以逐个检测这n个位置。编码器和解码器可以选择作为亮度帧内模式使用最频繁的帧内角度模式、DC模式或者平面模式以用于推导模式编解码的色度块。如果这些位置中没有一个是帧内角度模式、DC模式或者平面模式编解码的，则推导模式被处理为对当前色度块不可用。色度帧内模式列表可以被缩短1，并且相应的码字将被改变。这n个模式的顺序可以具有很多变形。

[0084] 填充用于色度帧内推导模式的默认模式

[0085] 在一个方法中，默认模式可以为预定义的帧内模式。这个帧内模式可以是硬件编解码以用于编码器和解码器。这个帧内模式也可以被发信在序列参数集或者图像参数集（Picture Parameter Set, PPS）或者片段头中。

[0086] 例如，默认模式可以是平面模式、DC模式、垂直模式或者水平模式。

[0087] 在另一方法中，默认模式可以被推导出以用于来自于其相邻块的每个色度块。色度块可以检测位于当前块的左上（即AL）、上方（即A）、右上（即AR）、左侧（即L）或者左下（即BL）的相邻块的帧内模式，并查找使用最频繁的相邻块的帧内模式作为用于当前块的帧内模式。如图13所示，是当前块的相邻块的示例。

[0088] 在第三方法中，来自于相邻块的第一可用帧内角度模式、DC模式或者平面模式用

作用于当前块的帧内模式。候选相邻块的顺序可以不同；下面列出了一些示例：

[0089] 1.L,A,AR,BL,AL;

[0090] 2.L,A,AL,AR,BL;以及

[0091] 3.AL,L,A,AR,BL。

[0092] 在上述所有中,L和A的顺序可以被互换,或者AR和BL的顺序可以被互换。在另一示例中,帧间模式运动向量预测子候选清单(空间候选)中的相邻位置的相同顺序可以被使用。在又一示例中,帧内模式合并候选列表(空间候选)中相邻位置的相同顺序可以被使用。

[0093] 适应性运动向量分辨率

[0094] 在第52届VCEG会议中,VCEG-AZ07(Chen,et al.,“Further improvements to HMKTA-1.0”,ITU-T VCEG,document:VCEG-AZ07,19-26June 2015,Warsaw,Poland)中公开了适应性运动向量分辨率。根据适应性运动向量分辨率,预测单元的当前运动向量与预测运动向量之间的运动向量差分(Motion Vector Difference,MVD)可以用四分之一像素分辨率或者整数像素分辨率来编解码。运动向量差分分辨率被控制在编码单元层,且整数运动向量差分分辨率标志被有条件地发信以用于每个编码单元,其具有至少一个非零运动向量差分分量。

[0095] 当整数运动向量差分分辨率标志为假(即表示整数分辨率不被使用)或者不被编解码以用于编码单元时,默认四分之一像素运动向量(Motion Vector,MV)被使用以用于属于这个编码单元的所有预测单元。否则(即整数运动向量分辨率为真以用于编码单元),用高级运动向量预测(advanced motion vector prediction,AMVP)模式编解码且属于这个编码单元的所有预测单元使用整数运动向量分辨率,而用合并模式编解码的预测单元仍然使用四分之一像素运动向量分辨率。当预测单元使用整数运动向量分辨率时,通过将四分之一像素运动向量四舍五入到整数像素运动向量,高级运动向量预测候选清单用整数运动向量来填充。

[0096] 在JEM3.0中,QTBT结构被使能,使得编码单元等于预测单元和变换单元(即编码单元=预测单元=变换单元)。同时,适应性运动向量分辨率被使能以用于每个编码单元。在这种情况下,一些子块可以共享相同的运动向量分辨率。但是,这些块具有其各自的标志,以发信适应性运动向量分辨率。这可能引起额外的发信开销。下面公开了各种方法以降低所需要的发信。

[0097] 方法1:修改用于QTBT结构中的二叉树的适应性运动向量分辨率的发信

[0098] 为了降低发信开销,本发明提出,当使用二进制分割时,在某个二叉树深度(Binary Tree depth,BT depth)(例如AMVR_signal_BT_flag)处发信AMVR_signal_BT_flag。二进制分割的子块共享来自于母二进制分割的相同的适应性运动向量分辨率标志。换句话说,深度大于MIN_BT_DEPTH_SIGNAL_AMVR的二进制块在MIN_BT_DEPTH_SIGNAL_AMVR处共享适应性运动向量分辨率标志。深度小于MIN_BT_DEPTH_SIGNAL_AMVR的二进制块发信其自身的适应性运动向量分辨率标志。当位于MIN_BT_DEPTH_SIGNAL_AMVR处的适应性运动向量分辨率标志为0时,所有子块使用整数运动向量分辨率。当位于MIN_BT_DEPTH_SIGNAL_AMVR处的适应性运动向量分辨率标志为1时,所有子块使用四分之一像素运动向量分辨率。如表1所示,是这个AMVR_signal_BT_flag的二进制表的示例。

[0099] 表1

[0100]	AMVR_signal_BT_flag	注释
	0	所有子树使用整数运动向量分辨率
	1	所有子树使用四分之一像素运动向量分辨率

[0101] 注意的是,上述表中的码字可以被互换。为了编解码AMVR_signal_BT_flag,N个上下文可以被使用,其中N可以为1,2,3,⋯,MAX_INT。

[0102] MIN_BT_DEPTH_SIGNAL_AMVR小于或等于MaxBTSize。其可以为0,1,2,3,⋯,MaxBTSize。MIN_BT_DEPTH_SIGNAL_AMVR可以被定义在编码器和解码器宏中,或者被发信在序列参数集、图像参数集或者片段头中。

[0103] 在一个示例中,MIN_BT_DEPTH_SIGNAL_AMVR对应于为0的二叉树深度,其意味着四叉树(quadtree)的叶和子块均是二进制分割。

[0104] 在另一示例中,MIN_BT_DEPTH_SIGNAL_AMVR对应于为1的二叉树深度,其意味着二进制分割的第一层。因此,当当前块为四叉树分割的时,这个标志将总是被发信。其如图14所示,其中当二叉树深度小于1时适应性运动向量分辨率标志被发信。

[0105] 在图14中,虚线块1410是二叉树深度等于1的二叉树,使得适应性运动向量分辨率标志将被发信以用于这层。当二叉树深度小于1时,运动向量分辨率由位于为1的二叉树深度处的适应性运动向量分辨率标志来表示。换言之,适应性运动向量分辨率标志仅被发信以用于没有分割的编码单元。

[0106] 在另一实施例中,当其为二进制分割的时,在某个二叉树深度(例如MIN_BT_DEPTH_SIGNAL_AMVR)处,AMVR_signal_BT_flag被发信。当位于MIN_BT_DEPTH_SIGNAL_AMVR处的适应性运动向量分辨率标志为0时,所有子块使用整数运动向量分辨率。当位于MIN_BT_DEPTH_SIGNAL_AMVR处的适应性运动向量分辨率标志为1时,所有子块使用四分之一像素运动向量分辨率。当位于MIN_BT_DEPTH_SIGNAL_AMVR处的适应性运动向量分辨率标志为2时,所有子块将发信其自身的适应性运动向量分辨率标志。AMVR_signal_BT_flag可以使用固定长度二进制化、变长度二进制化、k阶指数哥伦布(Exp-Golomb-k)二进制化或者截断二进制化。表2显示了Exp-Golomb-k码字的示例,其中k等于0。

[0107] 表2

	AMVR_signal_BT_flag 值	二进制化	注释
[0108]	0	0	所有子树使用整数运动向量分辨率
	1	10	所有子树使用四分之一像素运动向量分辨率
	2	11	所有子树使用其自身的适应性运动向量分辨率标志

[0109] 注意的是,码字分配可以被改变。为了编解码AMVR_signal_BT_flag,N个上下文可以被使用,其中N可以为1,2,3,⋯,MAX_INT。

[0110] 例如,MIN_BT_DEPTH_SIGNAL_AMVR可以小于或等于MaxBTSize。其可以为0,1,2,3,⋯,MaxBTSize。MIN_BT_DEPTH_SIGNAL_AMVR可以被定义在编码器和解码器宏中,或者被发信在序列参数集、图像参数集或者片段头中。

[0111] 在一个示例中,MIN_BT_DEPTH_SIGNAL_AMVR对应于为0的二叉树深度,其意味着四叉树的叶节点和子块均是二进制分割的。

[0112] 在另一示例中,MIN_BT_DEPTH_SIGNAL_AMVR对应于为1的二叉树深度,其意味着二进制分割的第一层。因此,当当前块为四叉树分割的时,这个标志将总是被发信。

[0113] 在另一实施例中,当其为二进制分割的时,在某个二叉树深度(例如MIN_BT_DEPTH_SIGNAL_AMVR)处,AMVR_signal_BT_flag被发信。当位于MIN_BT_DEPTH_SIGNAL_AMVR处的适应性运动向量分辨率标志为0时,所有子块发信其自身的适应性运动向量分辨率标志。当位于MIN_BT_DEPTH_SIGNAL_AMVR处的适应性运动向量分辨率标志为1时,所有子块使用四分之一像素运动向量分辨率。如表3所示,是这个AMVR_signal_BT_flag的二进制表的示例。

[0114] 表3

AMVR_signal_BT_flag	注释
0	所示子树使用其自身的适应性运动向量分辨率标志
1	所示子树使用四分之一像素运动向量分辨率

[0116] 注意的是,码字可以被互换。为了编解码AMVR_signal_BT_flag,N个上下文可以被使用,其中N可以为1,2,3,⋯,MAX_INT。

[0117] MIN_BT_DEPTH_SIGNAL_AMVR可以小于或等于MaxBTSize。其可以为0,1,2,3,⋯,MaxBTSize。MIN_BT_DEPTH_SIGNAL_AMVR可以被定义在编码器和解码器的宏中,或者被发信在序列参数集、图像参数集或者片段头中。

[0118] 在一个示例中,MIN_BT_DEPTH_SIGNAL_AMVR对应于为0的二叉树深度,其意味着四叉树的叶节点,并且子块均是二进制分割的。

[0119] 在另一示例中,MIN_BT_DEPTH_SIGNAL_AMVR对应于为1的二叉树深度,其意味着二进制分割的第一层。因此,当当前块为四叉树分割的时,这个标志将总是被发信。

[0120] 在另一实施例中,当其为二进制分割的时,在某个二叉树深度(例如MIN_BT_DEPTH_SIGNAL_AMVR)处,AMVR_signal_BT_flag被发信。当位于MIN_BT_DEPTH_SIGNAL_AMVR处的适应性运动向量分辨率标志为0时,所有子块发信其自身的适应性运动向量分辨率标志。当位于MIN_BT_DEPTH_SIGNAL_AMVR处的适应性运动向量分辨率标志为1时,所有子块使用整数像素运动向量分辨率。如表4所示,是这个AMVR_signal_BT_flag的二进制表的示例。

[0121] 表4

AMVR_signal_BT_flag	注释
0	所示子树使用其自身的适应性运动向量分辨率标志
1	所示子树使用整数像素运动向量分辨率

[0124] 注意的是,码字可以被互换。为了编解码AMVR_signal_BT_flag,N个上下文可以被使用,其中N可以为1,2,3,⋯,MAX_INT。

[0125] MIN_BT_DEPTH_SIGNAL_AMVR可以小于或等于MaxBTSize。其可以为0,1,2,3,⋯,

MaxBTSIZE.MIN_BT_DEPTH_SIGNAL_AMVR可以被定义在编码器和解码器的宏中,或者被发信在序列参数集、图像参数集或者片段头中。

[0126] 在一个示例中,MIN_BT_DEPTH_SIGNAL_AMVR对应于为0的二叉树深度,其意味着四叉树的叶节点和子块均是二进制分割的。

[0127] 在另一示例中,MIN_BT_DEPTH_SIGNAL_AMVR对应于为1的二叉树深度,其意味着二进制分割的第一层。因此,当当前块为四叉树分割的时,这个标志将总是被发信。

[0128] 方法2:修改用于QTBT结构中的四叉树的适应性运动向量分辨率的发信

[0129] 在所提出的方法中,当其为二进制分割的时,在某个四叉树深度(例如MIN_QT_DEPTH_SIGNAL_AMVR)处,AMVR_signal_QT_flag被发信。当位于MIN_QT_DEPTH_SIGNAL_AMVR处的AMVR_signal_QT_flag为0时,所有子块使用整数运动向量分辨率。当位于MIN_QT_DEPTH_SIGNAL_AMVR处的AMVR_signal_QT_flag为1时,所有子块使用四分之一像素运动向量分辨率。当位于MIN_QT_DEPTH_SIGNAL_AMVR处的AMVR_signal_QT_flag为2时,所有子块将发信其自身的适应性运动向量分辨率标志。表5显示了Exp-Golomb-k的示例,其中k等于0。

[0130] 表5

AMVR_signal_QT_flag value	二进制化	注释
0	0	所有子树使用整数运动向量分辨率
1	10	所有子树使用四分之一像素运动向量分辨率
2	11	所有子树使用其自身的适应性运动向量分辨率标志

[0132] 注意的是,码字分配可以被改变。为了编解码AMVR_signal_QT_flag,N个上下文可以被使用,其中N可以为1,2,3,⋯,MAX_INT。

[0133] MIN_QT_DEPTH_SIGNAL_AMVR小于或等于MaxQTSIZE。其可以为0,1,2,3,⋯,MaxQTSIZE。MIN_QT_DEPTH_SIGNAL_AMVR可以被定义在编码器和解码器的宏中,或者被发信在序列参数集、图像参数集或者片段头中。

[0134] 在另一实施例中,当其为二进制分割的时,在某个四叉树深度(例如MIN_QT_DEPTH_SIGNAL_AMVR)处,AMVR_signal_QT_flag被发信。当位于MIN_QT_DEPTH_SIGNAL_AMVR处的AMVR_signal_QT_flag为0时,所有子块使用整数运动向量分辨率。当位于MIN_QT_DEPTH_SIGNAL_AMVR处的AMVR_signal_QT_flag为1时,所有子块使用四分之一像素运动向量分辨率。表6示出了显示了Exp-Golomb-k的示例,其中k等于0。

[0135] 表6

AMVR_signal_QT_flag 值	注释
0	所有子树使用整数运动向量分辨率
1	所有子树使用四分之一像素运动向量分辨

[0137]

	率
--	---

[0138] 注意的是,码字可以被互换。为了编解码AMVR_signal_QT_flag,N个上下文可以被使用,其中N可以为1,2,3,⋯,MAX_INT。

[0139] MIN_QT_DEPTH_SIGNAL_AMVR小于或等于MaxQTSIZE。其可以为0,1,2,3,⋯,MaxQTSIZE。MIN_QT_DEPTH_SIGNAL_AMVR可以被定义在编码器和解码器的宏中,或者被发信在序列参数集、图像参数集或者片段头中。

[0140] 在另一实施例中,当其为二进制分割的时,在某个二叉树深度(例如MIN_QT_DEPTH_SIGNAL_AMVR)处,AMVR_signal_QT_flag被发信。当位于MIN_QT_DEPTH_SIGNAL_AMVR处的AMVR_signal_QT_flag为0时,所有子块发信其自身的适应性运动向量分辨率标志。当位于MIN_QT_DEPTH_SIGNAL_AMVR处的AMVR_signal_QT_flag为1时,所有子块使用四分之一像素运动向量分辨率。表7示出了显示了Exp-Golomb-k的示例,其中k等于0。

[0141] 表7

	AMVR_signal_QT_flag 值	注释
[0142]	0	所示子树使用其自身的适应性运动向量分辨率标志
	1	所示子树使用四分之一像素运动向量分辨率

[0143] 注意的是,码字可以被互换。为了编解码AMVR_signal_QT_flag,N个上下文可以被使用,其中N可以为1,2,3,⋯,MAX_INT。

[0144] MIN_QT_DEPTH_SIGNAL_AMVR小于或等于MaxQTSIZE。其可以为0,1,2,3,⋯,MaxQTSIZE。MIN_QT_DEPTH_SIGNAL_AMVR可以被定义在编码器和解码器的宏中,或者被发信在序列参数集、图像参数集或者片段头中。

[0145] 在另一实施例中,当其为二进制分割的时,在某个二叉树深度(例如MIN_QT_DEPTH_SIGNAL_AMVR)处,AMVR_signal_QT_flag被发信。当位于MIN_QT_DEPTH_SIGNAL_AMVR处的AMVR_signal_QT_flag为0时,所有子块使用整数像素运动向量分辨率。当位于MIN_QT_DEPTH_SIGNAL_AMVR处的AMVR_signal_QT_flag为1时,所有子块发信其自身的适应性运动向量分辨率标志。表8示出了显示了Exp-Golomb-k的示例,其中k等于0。

[0146] 表8

	AMVR_signal_QT_flag 值	注释
[0147]	0	所示子树使用整数像素运动向量分辨率
	1	所示子树使用其自身的适应性运动向量分辨率标志

[0148] 注意的是,码字可以被互换。为了编解码AMVR_signal_QT_flag,N个上下文可以被使用,其中N可以为1,2,3,⋯,MAX_INT。

[0149] MIN_QT_DEPTH_SIGNAL_AMVR可以小于或等于MaxQTSize。其可以为0,1,2,3,⋯, MaxQTSize。MIN_QT_DEPTH_SIGNAL_AMVR可以被定义在编码器和解码器的宏中,或者被发信在序列参数集、图像参数集或者片段头中。

[0150] 上述公开的本发明可以以各种形式合并到各种视频编码或者解码系统中。例如,本发明可以使用基于硬件的方法来实现,例如专用集成电路(integrated circuit,IC)、现场可程序设计逻辑数组(field programmable logic array,FPGA)、数据信号处理器(digital signal processor,DSP)、中央处理单元(central processing unit,CPU)等。本发明也可以使用计算机、掌上设备或者例如智能手机的移动设备上可执行的软件代码或者固件代码来实现。此外,软件代码或者固件代码可以在混合型平台上可执行,例如具有专用处理器的CPU(例如视频编解码引擎或者公共处理器)。

[0151] 图15示出了根据本发明实施例的用于色度块的推导模式帧内预测的示例性编解码系统的流程图。根据本方法,在步骤1510中,接收与当前色彩图像中的当前色度块相关的输入数据。在步骤1520中,检测推导模式帧内预测是否被选择以用于当前色度块。如果结果为“是”(即当前色度块为推导模式编解码的块),则执行步骤1530到步骤1560。否则(即“否”路径),跳过步骤1530到步骤1560。在步骤1530中,通过定位对应于当前色度块的第二左上像素地址的相应亮度块区域的第一左上像素地址,确定相应亮度块区域,其中相应亮度块区域对应于当前色彩图像中与当前色度块相同的区域。在步骤1540中,从包括相应亮度块区域的一个或多个子块的子块集中确定一个或多个候选亮度编解码模式,其中如果子块集仅包括一个子块,则子块集排除位于相应亮度块区域的左上角处的角子块。在步骤1550中,基于包括一个或多个候选亮度编解码模式的信息,推导出当前色度帧内模式以用于当前色度块。在步骤1560中,使用推导出的当前色度帧内模式,将帧内预测编码或者解码应用到当前色度块。

[0152] 本发明所示的流程图用于示出根据本发明的视频编解码的示例。在不脱离本发明的精神的情况下,本领域技术人员可以修改每个步骤、重组这些步骤、将一个步骤进行分离或者组合这些步骤而实施本发明。在本发明中,已经使用特定语法和语义来示出不同示例,以实施本发明的实施例。在不脱离本发明的精神的情况下,通过用等价的语法和语义来替换该语法和语义,本领域技术人员可以实施本发明。

[0153] 上述说明被呈现,使得本领域的技术人员能够在特定应用程序的内容及其需求中实施本发明。对技术人员来说,所描述的实施例的各种变形将是显而易见的,并且本文定义的一般原则可以应用于其他实施例中。因此,本发明不限于所示和描述的特定实施例,而是将被赋予与本文所公开的原理和新颖特征相一致的最大范围。在上述详细说明中,说明了各种具体细节,以便透彻理解本发明。不仅如此,将被本领域的技术人员能理解的是,本发明能够被实践。

[0154] 本发明以不脱离其精神或本质特征的其他具体形式来实施。所描述的例子在所有方面仅是说明性的,而非限制性的。因此,本发明的范围由权利要求的范围来表示,而不是前述的描述来表示。权利要求的含义以及相同范围内的所有变化都应纳入其范围内。

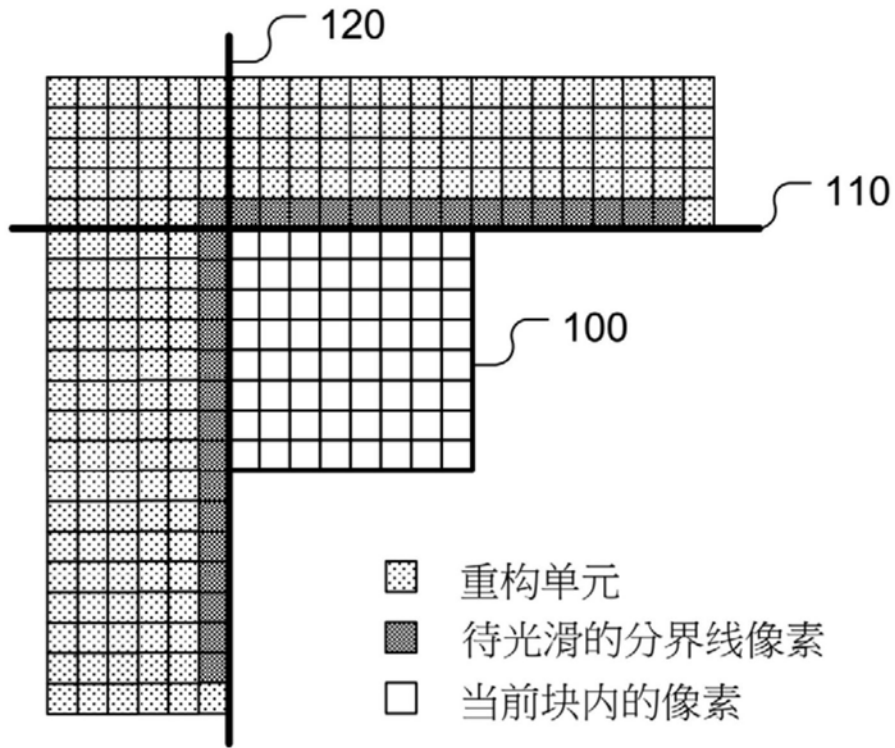


图1

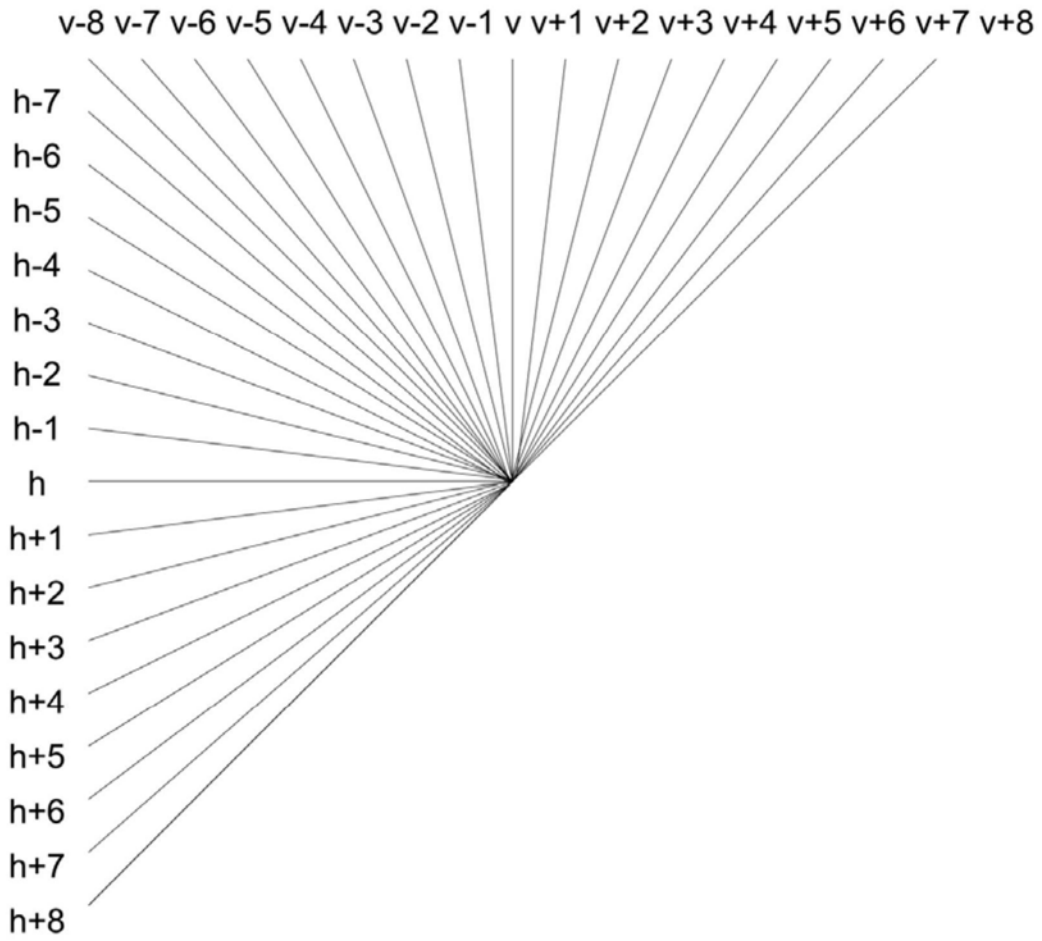


图2

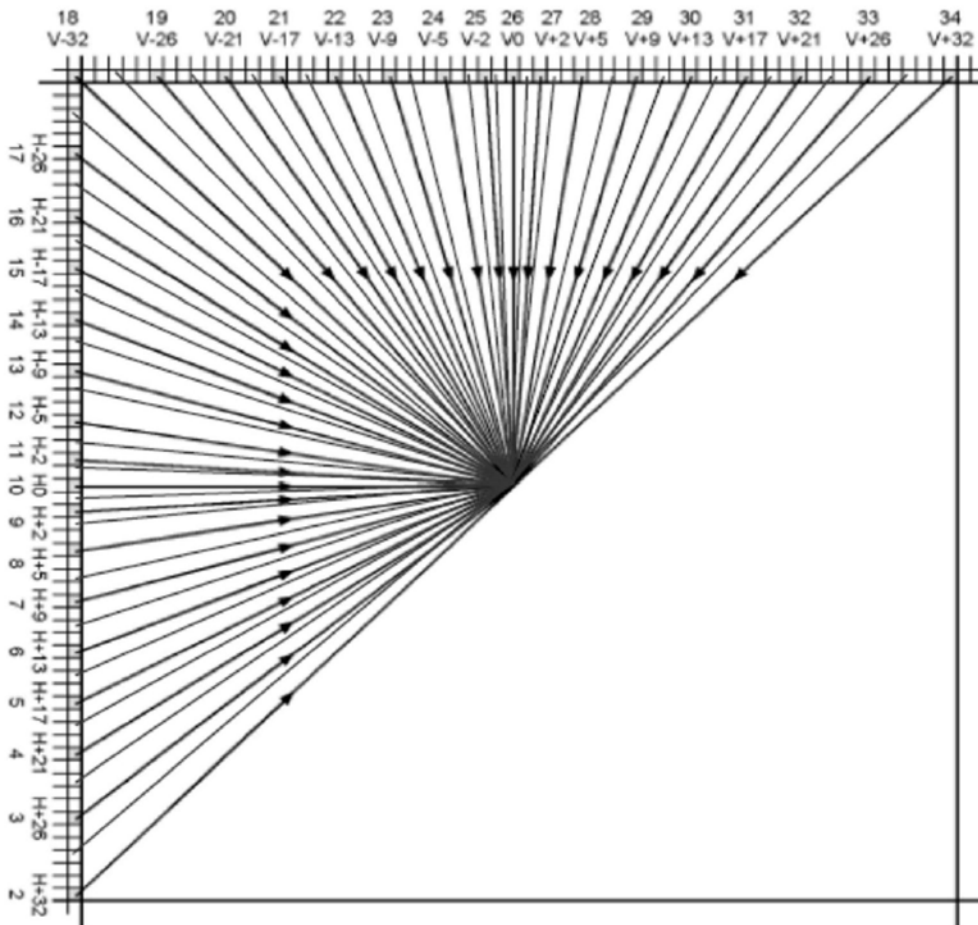


图3

The diagram shows a grid with 7 columns and 7 rows. Above the grid, there are eight downward-pointing arrows, one above each of the seven columns. A thick horizontal line is drawn between the first and second rows. A thick vertical line is drawn to the left of the first column. The grid cells contain the following text:

<i>AL</i>	<i>A0</i>		<i>Aj</i>			<i>Ak</i>
<i>L0</i>	<i>P00</i>	<i>...</i>	<i>P0j</i>	<i>...</i>		<i>P0k</i>
<i>L1</i>	<i>P10</i>		<i>P1j</i>			<i>P1k</i>
	<i>.</i>					
	<i>.</i>					
	<i>.</i>					
<i>Li</i>	<i>Pi0</i>		<i>Pij</i>			<i>Pik</i>
	<i>.</i>					
	<i>.</i>					
	<i>.</i>					
<i>Lk</i>	<i>Pk0</i>		<i>Pkj</i>			<i>Pkk</i>

图4A

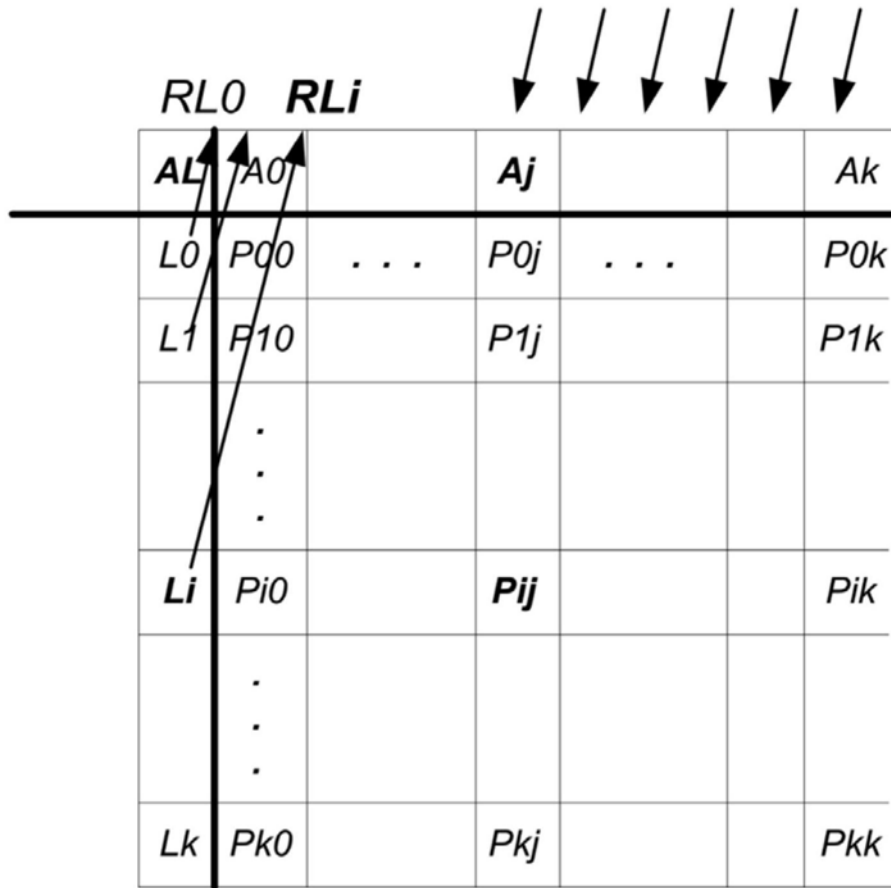


图4B

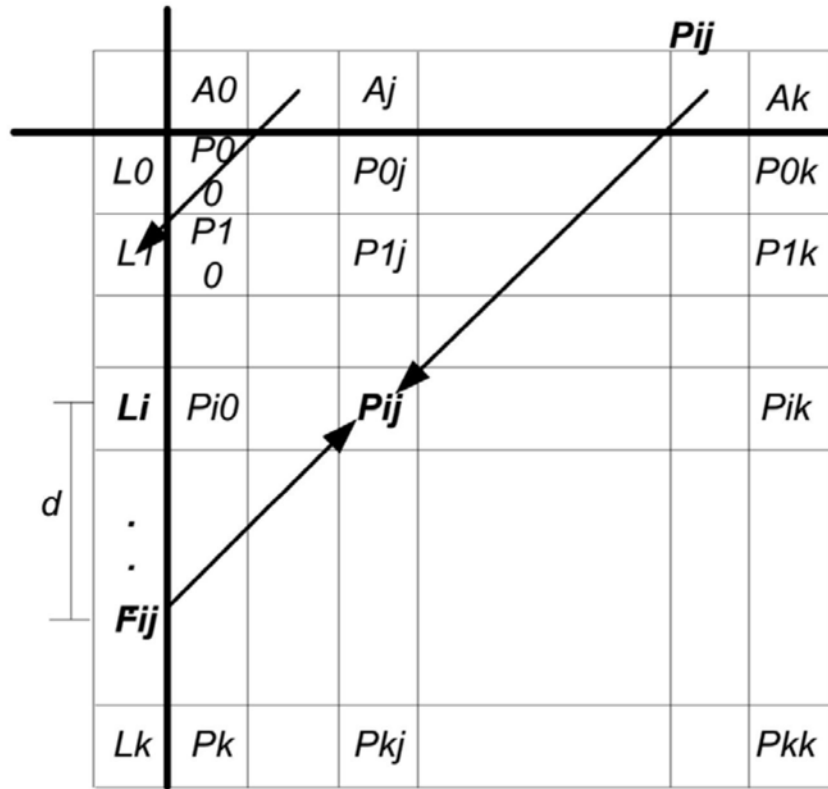


图5

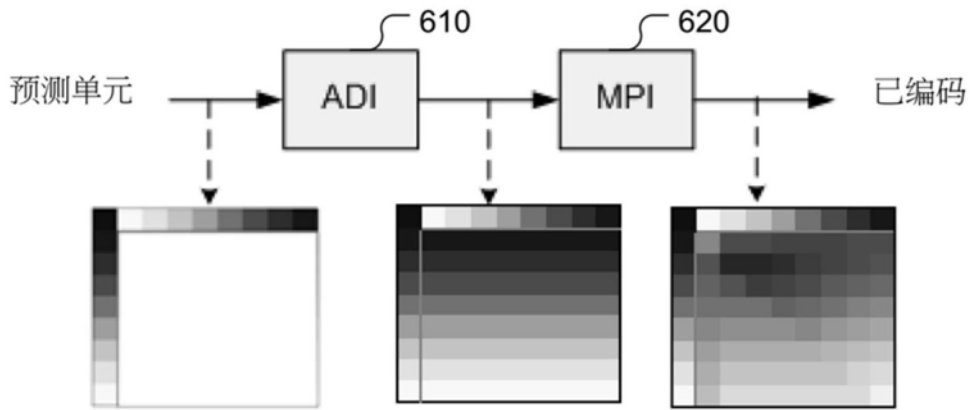


图6

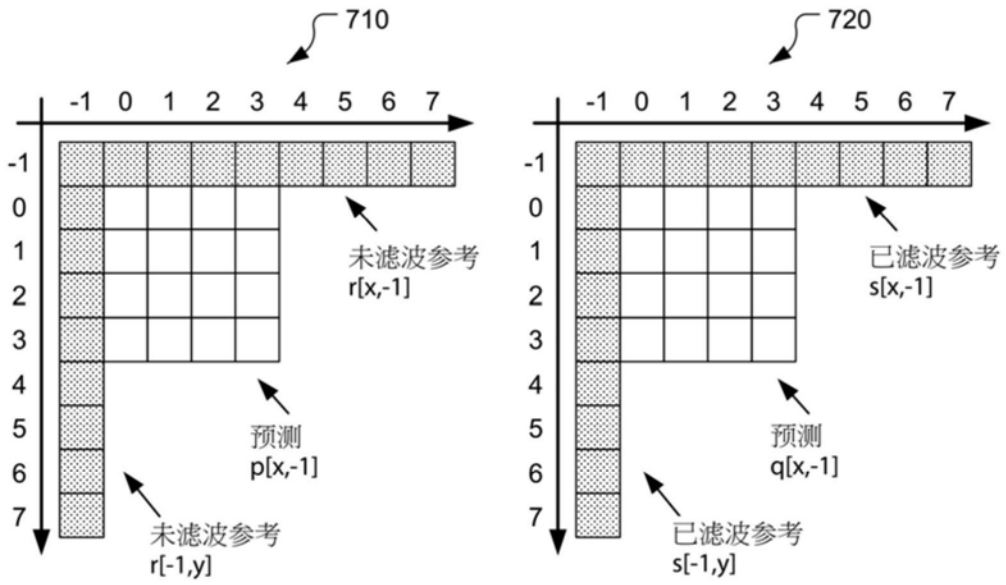


图7

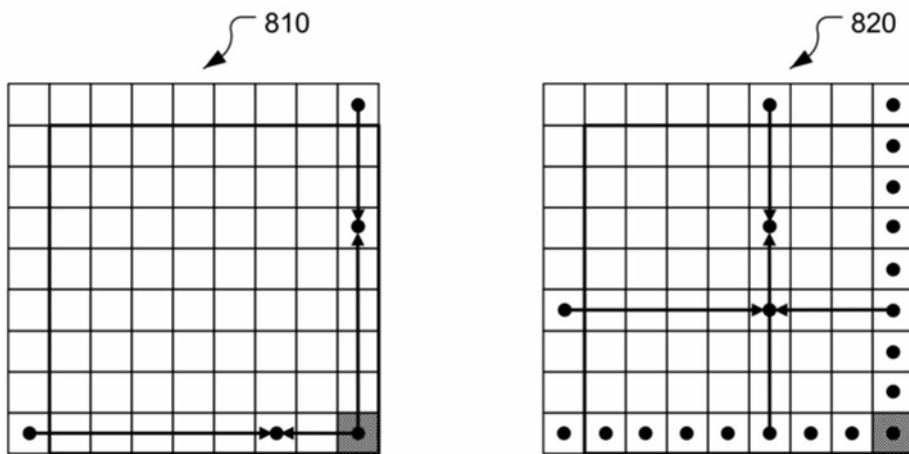


图8

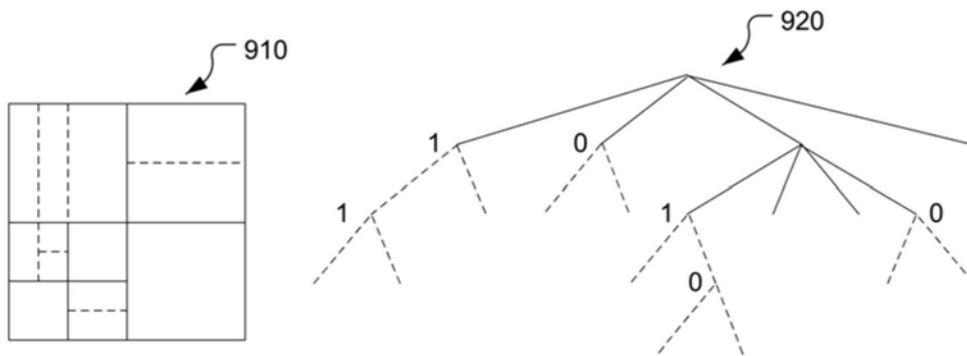


图9

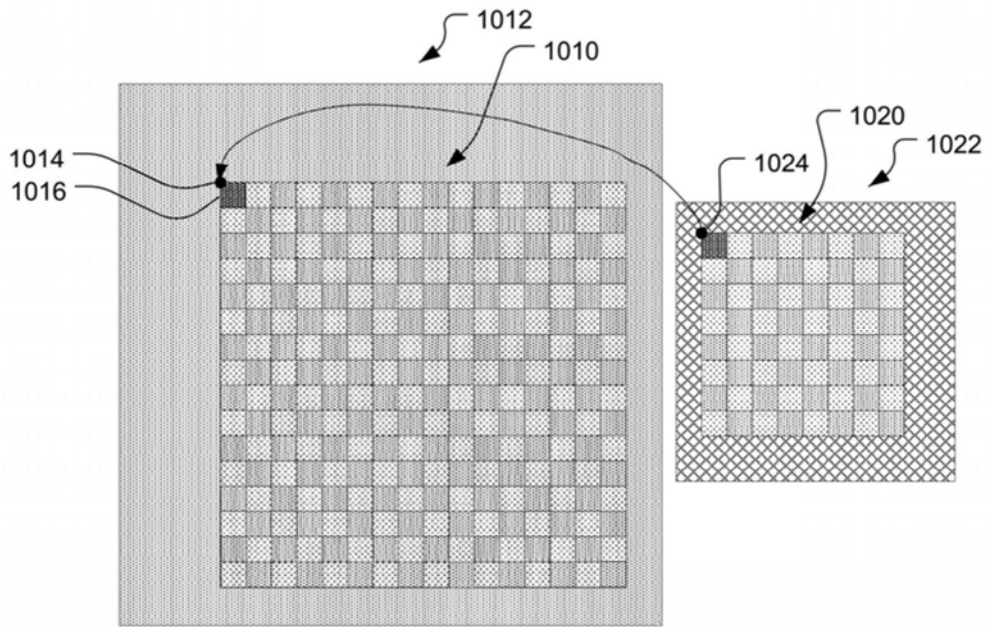


图10

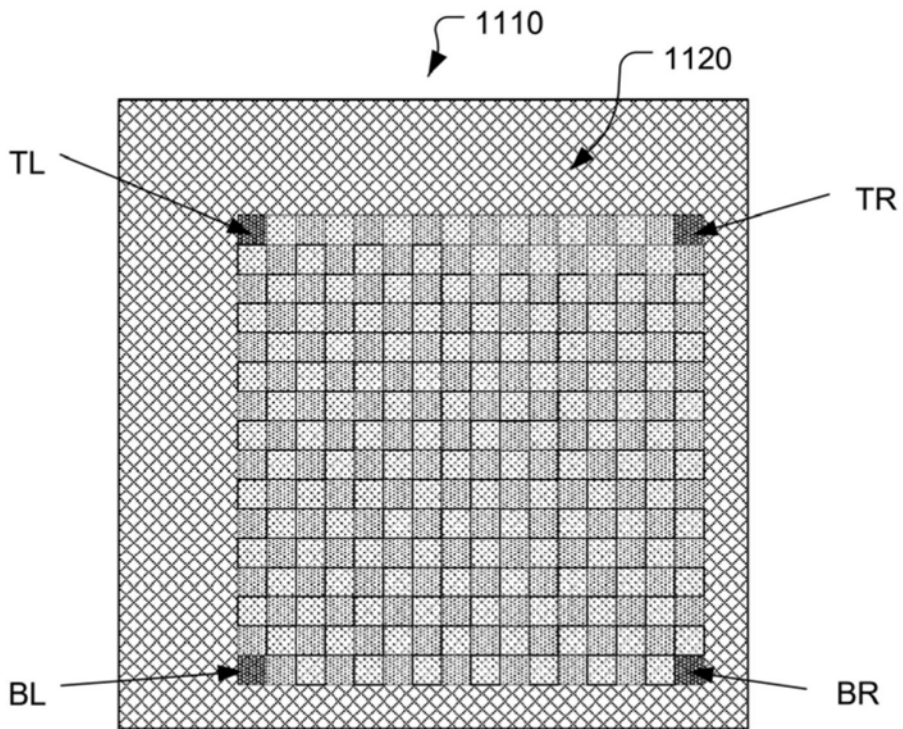


图11

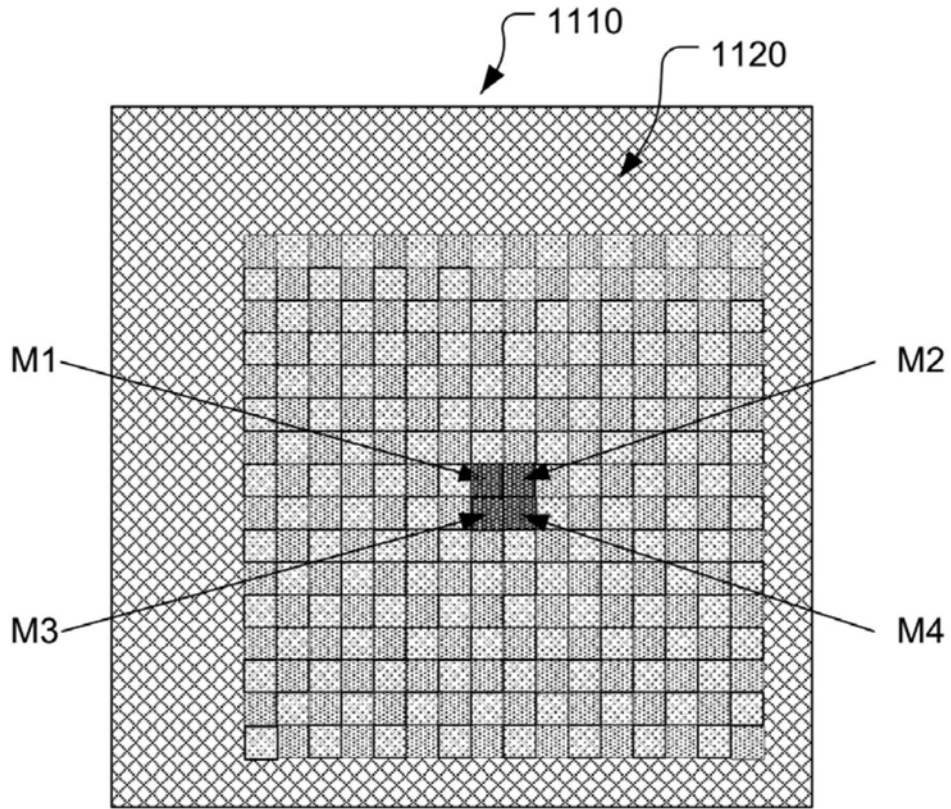


图12

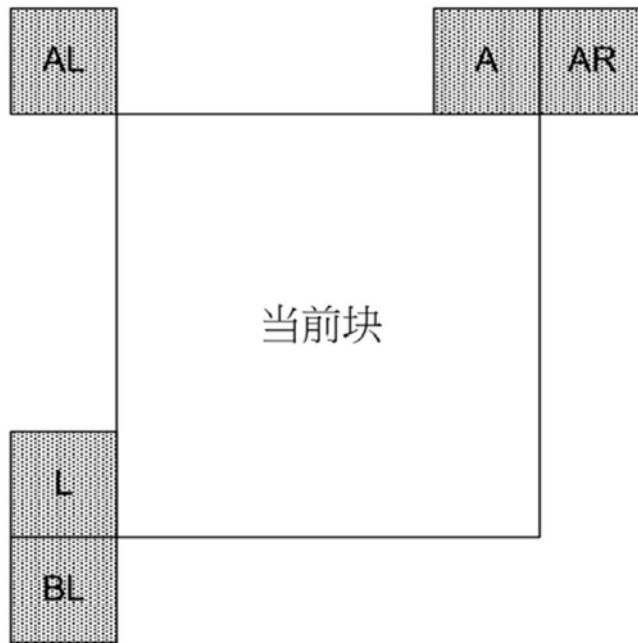


图13

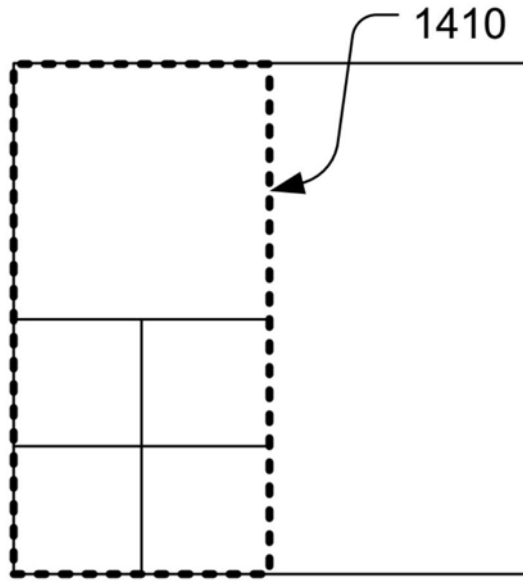


图14

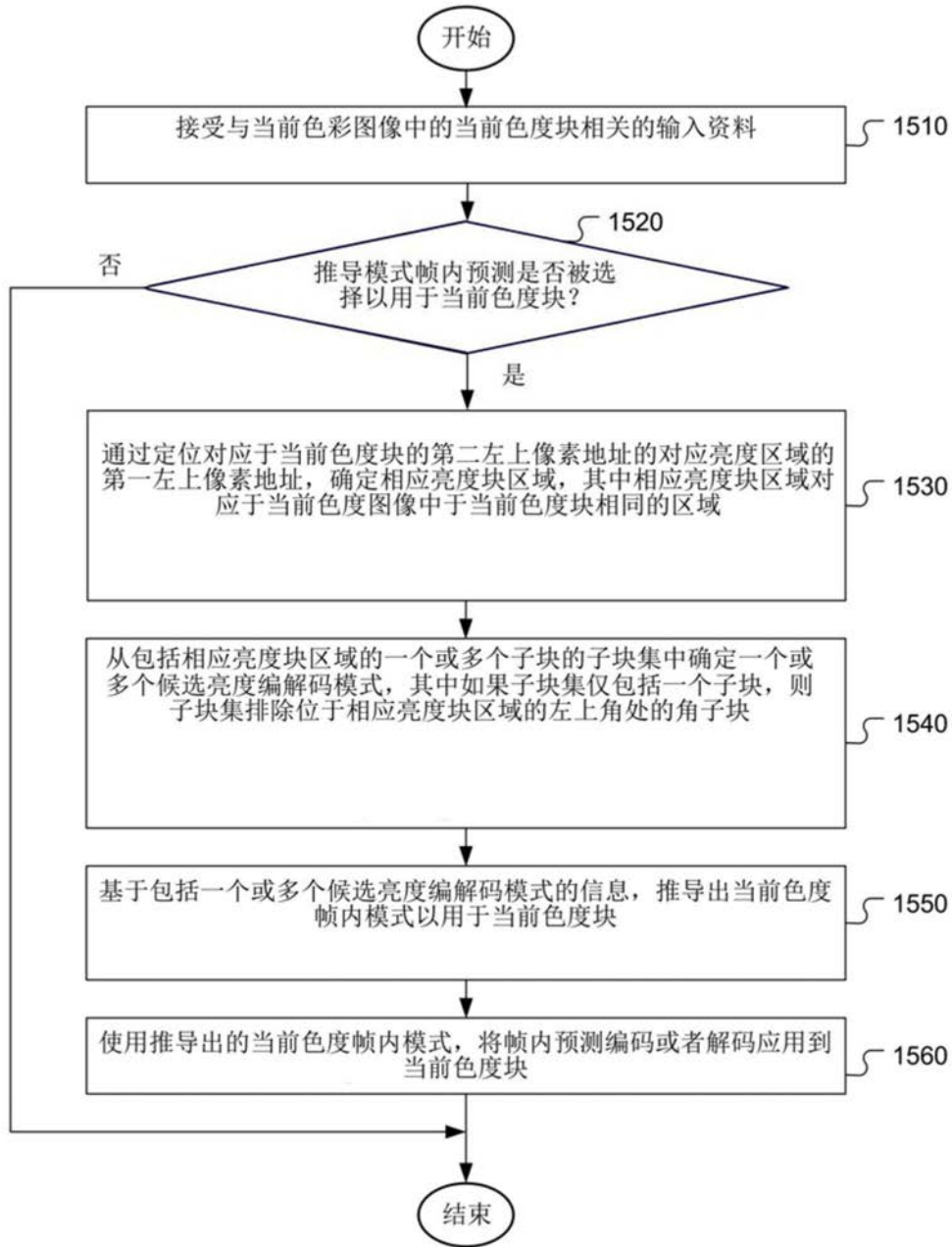


图15