



(12)发明专利申请

(10)申请公布号 CN 107409226 A

(43)申请公布日 2017. 11. 28

(21)申请号 201680013693.5

(22)申请日 2016.03.02

(30)优先权数据

62/126,983 2015.03.02 US

62/154,373 2015.04.29 US

(85)PCT国际申请进入国家阶段日

2017.09.01

(86)PCT国际申请的申请数据

PCT/CN2016/075279 2016.03.02

(87)PCT国际申请的公布数据

W02016/138854 EN 2016.09.09

(71)申请人 寰发股份有限公司

地址 中国台湾新竹县

(72)发明人 刘杉 许晓中

(74)专利代理机构 北京万慧达知识产权代理有限公司 11111

代理人 王蕊 白华胜

(51)Int.Cl.

H04N 19/59(2006.01)

H04N 19/593(2006.01)

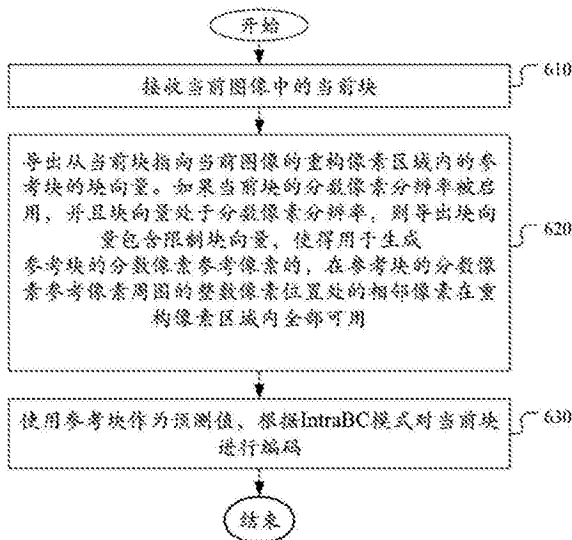
权利要求书3页 说明书17页 附图5页

(54)发明名称

用于视频编解码中的具有分数像素块向量分辨率的IntraBC模式的方法和装置

(57)摘要

揭露了一种使用IntraBC模式(块内复制模式)进行视频编解码的方法和装置。系统导出从当前块指向当前图像的重构像素区域内的参考块的块向量。如果块向量处于分数像素分辨率,则限制块向量,使得用于使用内插滤波器生成参考块的分数像素参考像素的,在参考块的分数像素参考像素周围的整数像素位置处的相邻像素在重构像素区域内全部可用。在另一个实施例中,用于内插以生成分数像素参考像素的不可用参考像素被填充。



1. 一种视频编码方法,使用块内复制模式编解码当前图像,其特征在于,包括:
接收所述当前图像中的当前块;

导出从所述当前块指向所述当前图像的重构像素区域内的参考块的块向量,其中如果所述当前块的分数像素分辨率被启用,并且所述块向量处于所述分数像素分辨率,则所述导出所述块向量包括限制所述块向量以使用于产生所述参考块的多个分数像素参考像素的所述参考块的所述多个分数像素参考像素周围的多个整数像素位置处的多个相邻像素在所述重构像素区域内全部可用;以及

根据块内复制模式使用所述参考块作为预测值对所述当前块进行编码。

2. 根据权利要求1所述的方法,其特征在于,所述参考块的所述多个分数像素参考像素使用具有2L滤波器抽头长度的内插滤波器,从整数像素位置处的所述多个相邻像素生成,其中L是正整数。

3. 根据权利要求2所述的方法,其特征在于,对于亮度块,L等于4,对于色度块,L等于2。

4. 根据权利要求3所述的方法,其特征在于,所述限制所述块向量将可用于生成所述参考块的所述多个分数像素参考像素的多个整数像素位置处的所述多个相邻像素的多个边界从所述重建的像素区域的多个边界向上移动至少(L-1)个像素,以及向左移动至少(L-1)像素。

5. 根据权利要求2所述的方法,其特征在于,根据与所述块向量相关联的一个或多个条件和一个或多个编解码参数是否满足,确定用于产生所述参考块的所述多个分数像素参考像素的所述参考块的所述多个分数像素参考像素周围的多个整数像素位置处的所述多个相邻像素在所述重构像素区域内是否全部可用。

6. 根据权利要求5所述的方法,其特征在于,所述当前块对应于PU(预测单元),并且所述一个或多个编解码参数包括由nPbSw表示的PU宽度,由nPbSh表示的PU高度,由(xPb,yPb)表示的相对于所述当前图像的所述PU的左上角像素的位置,由(xCb,yCb)表示的相对于所述当前图像的当前CU(编码单元)的左上角像素的位置,由CtbSize表示的CTU(编码树单元)的大小,并且其中PU在当前CU内。

7. 根据权利要求6所述的方法,其特征在于,所述一个或多个条件包括

$BV_x+nPbSw+xPb-xCb+(L-1)*2*(subWidthC-1)\leq 0$ 或者

$BV_y+nPbSh+yPb-yCb+(L-1)*2*(subHeightC-1)\leq 0$ 。

8. 根据权利要求7所述的方法,其特征在于,所述一个或多个条件进一步包括

$(xPb+Bv_x+nPbSw-1+(L-1)*2*(subWidthC-1))/CtbSize-xCb/CtbSize\leq$

$yCb/CtbSize-(yPb+Bv_y+nPbSh-1+(L-1)*2*(subHeightC-1))/CtbSize$ 。

9. 根据权利要求1所述的方法,其特征在于,所述当前块包括用于非4:4:4颜色视频数据的亮度块和色度块,与所述亮度块相关联的亮度块向量总是处于整像素分辨率,并且与所述色度块相关联的色度块向量缩放自所述亮度块向量,并且如果所述色度块向量处于所述分数像素分辨率,则将所述限制所述块向量应用于所述色度块向量。

10. 根据权利要求1所述的方法,其特征在于所述当前块对应于亮度块。

11. 一种视频解码方法,使用块内复制模式编解码当前图像,其特征在于,包括:

接收包括用于所述当前图像中当前块的编码数据的视频位流;

从所述视频位流解析当前块向量,其中所述当前块向量从所述当前块指向所述当前图

像的重构像素区域内的参考块；

如果所述当前块向量处于分数像素分辨率，则使用内插滤波器在多个分数像素位置处生成用于所述参考块的多个分数像素参考像素，其中如果所述内插滤波器使用的任何滤波器输入数据不可用，则不可用的滤波器输入数据通过所述内插滤波器填充进行内插；以及根据块内复制模式使用所述参考块作为预测值解码所述当前块。

12. 根据权利要求11所述的方法，其特征在于，通过水平延伸所述参考块的最右边的可用列并且垂直地延伸所述参考块的最底部可用行来填充所述不可用的滤波器输入数据。

13. 根据权利要求12所述的方法，其特征在于，在垂直地延伸所述参考块的所述最底部可用行之前，水平地延伸所述参考块的所述最右边的可用列。

14. 根据权利要求12所述的方法，其特征在于，在垂直地延伸所述参考块的所述最底部可用行之后，水平地延伸所述参考块的所述最右边的可用列。

15. 根据权利要求11所述的方法，其特征在于，使用默认值或预置值填充所述不可用的滤波器输入数据。

16. 根据权利要求15所述的方法，其特征在于，所述默认值或预置值对应于零或 $((1 \ll \text{bit_depth}) \gg 1)$ ，其中 bit_depth 对应于所述当前图像的像素数据位深度。

17. 根据权利要求11所述的方法，其特征在于，所述内插滤波器具有使用滤波像素的每一侧上的 L 个滤波器输入数据的 $2L$ 滤波器抽头长度，其中 L 是正整数。

18. 一种视频编码方法，使用块内复制模式编解码当前图像，其特征在于，包括：

接收所述当前图像中的当前块；

导出从所述当前块指向所述当前图像的重构像素区域内的参考块的当前块向量；

如果所述当前块向量处于分数像素分辨率，则使用内插滤波器在多个分数像素位置处生成用于所述参考块的多个分数像素参考像素，其中如果所述内插滤波器使用的任何滤波器输入数据不可用，则通过所述内插滤波器填充不可用的滤波器输入数据进行内插；以及根据块内复制模式使用所述参考块作为预测值对所述当前块进行编码。

19. 根据权利要求18所述的方法，其特征在于，通过水平延伸所述参考块的最右边的可用列以及垂直地延伸所述参考块的最底部可用行来填充所述不可用的滤波器输入数据。

20. 根据权利要求19所述的方法，其特征在于，在垂直地延伸所述参考块的所述最底部可用行之前，水平地延伸所述参考块的所述最右边的可用列。

21. 根据权利要求19所述的方法，其特征在于，在垂直地延伸所述参考块的所述最底部可用行之后，水平地延伸所述参考块的所述最右边的可用列。

22. 根据权利要求18所述的方法，其特征在于，使用默认值或预置值填充所述不可用的滤波器输入数据。

23. 根据权利要求22所述的方法，其特征在于，所述默认值或预设值对应于零或 $((1 \ll \text{bit_depth}) \gg 1)$ ，其中 bit_depth 对应于所述当前图像的像素数据位深度。

24. 根据权利要求18所述的方法，其特征在于，所述内插滤波器具有使用滤波像素的每一侧上的 L 个滤波器输入数据的 $2L$ 滤波器抽头长度，其中 L 是正整数。

25. 一种视频解码装置，在视频解码系统中使用块内复制模式编解码，包括一个或多个电路，其特征在于，其被布置为：

接收包含用于当前图像中当前块的编码数据的视频位流；

从所述视频位流解析当前块向量,其中所述当前块向量从所述当前块指向所述当前图像的重构像素区域内的参考块;

如果所述当前块向量处于分数像素分辨率,则使用内插滤波器在多个分数像素位置处生成用于所述参考块的多个分数像素参考像素,其中如果所述内插滤波器使用的任何滤波器输入数据不可用,则不可用的滤波器输入数据通过所述内插滤波器填充进行内插;以及根据块内复制模式使用所述参考块作为预测值解码所述当前块。

用于视频编解码中的具有分数像素块向量分辨率的IntraBC模式的方法和装置

[0001] 【交叉引用】

[0002] 本申请要求申请日为2015年3月2日,美国临时申请号为62/126,983的美国临时申请案和申请日为2015年4月29日,美国临时申请号为62/154,373的美国临时申请案的优先权。上述美国临时申请案的内容一并并入本申请。

【技术领域】

[0003] 本发明涉及具有用于IntraBC (Intra Block Copy, 块内复制) 模式的帧间预测和块向量的统一运动向量的视频编解码系统。特别地,本发明涉及当块向量处于分数像素分辨率(fractional-pel resolution)时处理不可用的参考像素。

【背景技术】

[0004] 高效视频编解码(High Efficiency Video coding, 简称为HEVC)是近年来已经开发的新的编码标准。在高效视频编解码(HEVC)系统中,H.264/AVC的固定大小宏块(fixed-size macro block)由称为编解码单元(coding unit, 简称为CU)的灵活块替换。CU中的像素共享相同的编解码参数以提高编解码效率。CU可以以最大CU(LCU)开始,其在HEVC中也被称为已编解码树单元(coded tree unit, 简称为CTU)。除了编解码单元的概念,在HEVC中也引入了预测单元(prediction unit, 简称为PU)的概念。一旦CU层次树的分裂完成,则根据预测类型和PU分割将每个叶CU被进一步拆分成一个或多个预测单元(PU)。

[0005] 伴随着高效率视频编解码(HEVC)标准的开发,HEVC的扩展开始也开始了。HEVC扩展包括屏幕内容编码(screen content coding, 简称为SCC)。由于屏幕内容的特殊性,编解码工具已被开发出来,并且在编解码效率方面取得了显著的进步。其中,颜色索引编解码(亦被称为主要基于颜色的编解码)技术表示使用调色板(主要颜色)的索引的像素块,并通过利用空间冗余来编码调色板和索引。虽然可能的颜色组合的总数是巨大的,但是对于典型的屏幕内容,图像区域中的颜色数量通常非常有限。因此,颜色索引编码对于屏幕内容材料变得非常有效。相关的关键颜色索引编码技术简要回顾如下。

[0006] 在屏幕内容编解码(SCC)开发过程中,已经描述了各种视频编解码工具,包括“内部图像块复制(Intra picture block copy)”(IntraBC)技术。JCTVC-M0350(2013年4月18-26日,韩国仁川举行的视频编码联合协作小组(Joint Collaborative Team on Video编解码,简称为JCT-VC)ITU-TSG16WP3和ISO/IECJTC1/SC29/WG11第13次会议中,Budagavi等人,AHG8:使用帧内运动补偿的视频编码(Video编解码using Intra motion compensation),文献:JCTVC-M0350)中最初提出IntraBC技术。图1中示出了根据JCTVC-M0350的范例,其中使用帧内运动补偿(motion compensation, 简称为MC),对当前编解码单元(CU,110)进行编码。预测块(120)由当前CU和位移向量(displacement vector)(112)来定位。在该示例中,搜索区域限于当前编码树单元(coding tree unit, 简称为CTU)、左CTU和左左CTU。从已经重构的区域获得预测块。然后,对当前CU的位移向量(也称为运动向量(motion vector, 简

写为MV)或块向量(block vector,简称为BV)和残差进行编解码。众所周知,HEVC采用CTU和CU块结构作为用于对视频数据进行编解码的基本单元。每个图像被分成CTU,并且每个CTU被划分成CU。在预测阶段期间,每个CU可以被划分为多个块,称为预测单元(prediction unit,简称为PU),用于执行预测处理。

[0007] 在JCTVC-M0350中,在至少以下方面,块内复制不同于用于帧间预测的运动补偿:

[0008] 对于块内复制(即,水平或垂直),BV被限制为1-D,而帧间预测使用2-D运动估计。

[0009] 对于块内复制,二进制化是固定长度,而帧间预测使用哥伦布指数(exponential-Golomb)。

[0010] 块内复制引入了一个新的语法元素来指示BV是水平还是垂直。

[0011] 基于JCTVC-M0350,在2013年7月25日-8月2日,在维也纳举行的JCT-VC ITU-TSG16WP3和ISO/IECJTC1/SC29/WG11第14次会议中,在非-RCE3:具有2-D MV的帧内运动补偿(Intra Motion Compensation with 2-DMVs)中,Pang等人公开了一些修改,文件:JCTVC-N0256(以下称为JCTVC-N0256)。首先,块内复制被扩展以支持2-D MV,使得垂直和水平MV分量可以同时是非零的。这为块内复制提供了比原始方法更多的灵活性,其中MV被限制为严格水平或垂直。

[0012] 在JCTVC-N0256中,公开了两种BV编码方法:

[0013] 方法1:块向量预测(Block vector prediction)。选择左或上BV作为BV预测值,并对所得到的运动向量差(motion vector difference,简称为BVD)进行编解码。使用标志来指示BVD是否为零。当BVD不为零时,使用3阶的哥伦布指数码来对BVD的剩余绝对水平(remaining absolute level)进行编解码。另一个标志(flag)用于对符号(sign)进行编解码。

[0014] 方法2:无块向量预测。BV使用用于HEVC中的BVD的哥伦布指数码来编解码。

[0015] JCTVC-N0256中公开的另一个差异是,2-D IntraBC进一步与流水线友好方法组合:

[0016] 1. 不使用内插滤波器。

[0017] 2. BV搜索区域受到限制。公开了两种情况:

[0018] a. 搜索区域是当前CTU和左CTU或

[0019] b. 搜索区域是当前CTU和左CTU的最右边4列样本。

[0020] 在JCTVC-N0256中提出的方法中,在HEVC SCC标准草案中采用2-D IntraBC、去除内插滤波器和对当前CTU和左CTU的搜索区域约束。

[0021] 有效的块向量应指向当前图像的已经重建的区域,并且应该在当前CU之外。此外,在SCM-3.0(Joshi等人,屏幕内容编解码测试模型3(SCM3),ITU-TSG16WP3和ISO/IECJTC1/SC29/WG11视频编码联合协作团队(JCT-VC),第19次会议:Strasbourg,FR,2014年10月17日至10月24日,文件:JCTVC-S1014)中,采用梯形IntraBC搜索范围约束,如图2所示,最初用于波前并行处理(Wavefront Parallel processing,简称为WPP)。但是,无论是否用于WPP,它用于HEVC SCC。在图2,每个正方形代表一个CTU。对于当前CTU中的IntraBC块,其可用搜索区域被限制为当前CTU中的点填充CTU和重构块。

[0022] 因此,对于IntraBC编解码块,其块向量 $BV = (BV_x, BV_y)$ 应满足以下位流一致性条件:

[0023] $BV_x+nPbSw+xPb-xCb \leq 0$ 或

[0024] $BV_y+nPbSh+yPb-yCb \leq 0$ (1)

[0025] $(xPb+Bv_x+nPbSw-1)/CtbSize-xCb/CtbSize \leq$

[0026] $yCb/CtbSize-(yPb+Bv_y+nPbSh-1)/CtbSize$ (2)

[0027] 在上述等式中, $nPbSw$ 和 $nPbSh$ 是当前PU的宽度和高度; (xPb, yPb) 是当前PU相对于当前图像的左上像素的位置; (xCb, yCb) 是当前CU相对于当前图像的左上像素的位置; $CtbSize$ 是CTU块的大小。当满足表达式(1)和(2)时,块向量指向的参考块将完全位于当前块的可用搜索区域内。

[0028] 作为帧间编解码模式的IntraBC

[0029] 在最近的编解码标准会议中, IntraBC统一了帧间编解码(inter coding)模式(Pang等人, Non-CE2Test1: Intra block copy and inter signalling unification, ITU-TSG16WP3和ISO/IECJTC1/SC29/WG11视频编码联合协作团队(JCT-VC), 第20次会议: Geneva, CH, 2015年2月10日至2月18日, 文件: JCTVC-T0227)。换句话说,当前图像被视为参考图像并插入到一个或两个参考图像列表中。块向量预测和编解码与帧间运动向量预测和编解码相同。这种统一简化了编解码器设计。但是还有一些问题。对于IntraBC,块向量具有整数分辨率。然而,运动向量具有在切片级切换的整数和四分之一像素(quarter-pel)分辨率。当以四分之一像素分辨率处理块向量时,需要对由块向量指向的块周围的一些额外的像素行和像素列进行内插(interpolated)处理。这些像素在当前的IntraBC-Inter统一方案中可能不可用。

[0030] 在高级语法中,在参考图像列表构造的初始化期间,当前图像的编解码数据被放置在位流中所有短期参考图像和所有其他长期参考图像之后。用于相关参考图像处理的示例性伪码在列表0中示出如下。在每个切片的解码处理开始时,参考图像列表RefPicList0,以及B切片的RefPicList1被导出。

[0031] 将变量NumRpsCurrTempList0设置为等于 $\text{Max}(\text{num_ref_idx_l0_active_minus}+1, \text{NumPicTotalCurr})$,并且如表1所示构造列表RefPicListTemp0。

[0032] 表1

[0033]

```

rIdx = 0
while( rIdx < NumRpsCurrTempList0 ) {
    for( i = 0; i < NumPocStCurrBefore && rIdx < NumRpsCurrTempList0;
        rIdx++, i++ )
        RefPicListTemp0[ rIdx ] = RefPicSetStCurrBefore[ i ]
    for( i = 0; i < NumPocStCurrAfter && rIdx < NumRpsCurrTempList0;
        rIdx++, i++ )
        RefPicListTemp0[ rIdx ] = RefPicSetStCurrAfter[ i ]
}

```

[0034]

```

        for( i = 0; i < NumPocLtCurr && rIdx < NumRpsCurrTempList0; rIdx++, i++
    )
        RefPicListTemp0[ rIdx ] = RefPicSetLtCurr[ i ]
        if( curr_pic_as_ref_enabled_flag )
            RefPicListTemp0[ rIdx++ ] = currPic
    }

```

[0035] 在初始化之后,参考图像列表(即RefPicList0)结构在如下所示的示例伪码中执行:

[0036] for(rIdx=0; rIdx<=num_ref_idx_l0_active_minus1; rIdx++)

[0037] RefPicList0[rIdx]=ref_pic_list_modification_flag_l0?

[0038] RefPicListTemp0[list_entry_l0[rIdx]]:RefPicListTemp0[rIdx]

[0039] 然而,当num_ref_idx_l0_active_minus1小于条目(entry)的索引时,在该条目中RefPicListTemp0阵列存储当前图像,当前图像可能不包括在参考图像列表中。因此,需要开发克服这些问题的技术。

【发明内容】

[0040] 揭露了一种使用IntraBC模式(块内复制模式)进行视频编解码的方法和装置。在一个实施例中,系统出从当前块指向当前图像的重构像素区域内的参考块的块向量。如果当前块的分数像素分辨率被启用,并且块向量处于分数像素分辨率,则导出块向量包括限制块向量以使用于产生参考块的多个分数像素参考像素的参考块的多个分数像素参考像素周围的多个整数像素位置处的多个相邻像素在重构像素区域内全部可用。使用具有2L滤波器抽头长度的内插滤波器,从整数像素位置处的多个相邻像素生成参考块的多个分数像素参考像素,其中L是正整数。对于亮度块,L等于4,对于色度块,L等于2。在一个实施例中,限制块向量将可用于生成参考块的多个分数像素参考像素的多个整数像素位置处的多个相邻像素的多个边界从重建的像素区域的多个边界向上移动至少(L-1)个像素,以及向左移动至少(L-1)像素。

[0041] 根据与块向量相关联的一个或多个条件和一个或多个编解码参数是否满足,确定用于产生参考块的多个分数像素参考像素的参考块的多个分数像素参考像素周围的多个整数像素位置处的多个相邻像素在重构像素区域内是否全部可用。

[0042] 在另一个实施例中,当前块包括用于非4:4:4颜色视频数据的亮度块和色度块,与亮度块相关联的亮度块向量总是处于整像素分辨率,并且与色度块相关联的色度块向量缩放自亮度块向量,并且如果色度块向量处于分数像素分辨率,则将限制块向量应用于色度块向量。在又一个实施例中,当前块对应于亮度块,如果亮度块向量处于分数像素分辨率,则将块向量限制应用于亮度块向量。

[0043] 在另一个实施例中,不可用的参考像素被填充用于与分数像素分辨率相关联的内插。通过水平延伸参考块的最右边的可用列并且垂直地延伸参考块的最底部可用行来填充不可用的滤波器输入数据。可以先执行水平延伸。可选地,可以先执行垂直延伸。可以使用

默认值或预置值填充不可用的滤波器输入数据。例如,默认值或预设值可以对应于零或 $(1 \ll \text{bit_depth}) >> 1$,其中 bit_depth 对应于当前图像的像素数据位深度。

【附图说明】

[0044] 图1中示出了用于屏幕内容编解码的块内复制(IntraBC)编解码的范例。

[0045] 图2示出了可用于IntraBC(块内复制)模式中的参考像素的梯形重建区域的示例。

[0046] 图3示出了可用重建区域中的IntraBC参考块的示例,其中假定了整数块向量。

[0047] 图4示出了用于分数像素块向量的不可用参考像素的示例,其中围绕参考块的右侧和底侧的相邻重建像素不足以进行内插。

[0048] 图5示出了通过首先水平延伸最右列然后垂直延伸最底行来填充用于分数像素内插的不可用区域的示例。

[0049] 图6示出了结合本发明的实施例的用于IntraBC(块内复制模式)编码的视频编码器的示例性流程图。

[0050] 图7示出了使用IntraBC模式(块内复制模式)编解码用于结合本发明的实施例的图像的视频解码的示例性流程图。

【具体实施方式】

[0051] 以下描述是实现本发明的最佳设想模式。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所附权利要求书来确定。

[0052] 用于内插的块向量约束

[0053] 本发明的一个方面涉及与块向量内插有关的问题。对于使用IntraBC模式的当前块,如果块向量(即,指向当前图像内的参考区域的运动向量)处于整数位置,则满足等式(1)和(2)中指定的位流一致性条件的,在当前CU之外的参考块中的像素可用于参考。图3示出了可用重建区域中的IntraBC参考块的示例,其中假定了整数块向量。块向量330从当前块310指向参考块320。线填充区域340表示可用的参考区域,而白色区域350表示不可用的参考区域。

[0054] 当块向量指向分数像素(即子像素)位置时,即使满足位流一致性条件(1)和(2),参考块也可能不完全可用于内插。如果参考块处于子像素位置,则参考块中每个分数像素处的像素值必须从相邻的现有像素内插。因此,除了最接近的整数像素之外,还可能需要一些其他的相邻整数像素进行内插。通常使用对称内插滤波器(symmetric interpolation filter)。如果滤波器具有等于 $2L$ 的滤波器抽头长度,则它需要子像素位置的每一侧有 L 个整数像素可用于内插该子像素位置。在HEVC中,亮度像素的 $L=4$,色度像素的 $L=2$ 。有时,这些邻居可能不适用于可用参考区域边界附近的参考块等情况。

[0055] 图4示出了用于分数像素块向量的不可用参考像素的示例,其中围绕参考块的右侧和底侧的周围相邻重建像素不足以进行内插。从当前块410指向参考块420的块向量430是分数像素分辨率。参考块420内的分数像素参考数据需要通过内插生成。参考块内的分数像素位置由小的菱形符号440表示。厚边界450以上的区域表示可用的参考区域,厚边界450以下的区域表示不可用的参考区域。参考块接近可用参考区域的边界(即右侧2像素,下面2像素)。由于 $L=4$ 需要当前像素的每一侧上的4个相邻像素,所以右侧的两个像素列和底部

仅两个像素行不足以用于内插操作。因此,提出一种限制块向量的方法,以确保所有用于内插的像素数据都可用。

[0056] 方法1.约束块向量

[0057] 根据该方法,对块向量施加约束,使得其只能指向具有用于内插的边界周围有足够可用的重建参考区域的参考块。特别地,在可用的重建参考区域中,右边界的L列和底边界下方的L行应当可用于内插参考块的分数像素位置。如前所述,2L是内插滤波器的长度。

[0058] 实施例1.根据本实施例,如等式(3)和(4)所示,通过修改位流一致性条件(1)和(2)来限制块向量(BV_x,BV_y):

$$[0059] \quad BV_x + nPbSw + xPb - xCb + (L-1) \leq 0 \text{ 或}$$

$$[0060] \quad BV_y + nPbSh + yPb - yCb + (L-1) \leq 0 \quad (3)$$

$$[0061] \quad (xPb + BV_x + nPbSw - 1 + (L-1)) / CtbSize - xCb / CtbSize < =$$

$$[0062] \quad yCb / CtbSize - (yPb + BV_y + nPbSh - 1 + (L-1)) / CtbSize \quad (4)$$

[0063] 在上述等式中,内插滤波器长度为2L,CtbSize为亮度CTU块大小。这两个约束(3)和(4)应适用于块向量。要被内插的像素位置的每一侧的内插滤波器长度(即,L)可以与现有的HEVC SCC草案标准中相同。

[0064] 与位流一致性条件(1)和(2)相比,根据上述实施例的可用参考像素边界在垂直方向上偏移了(L-1)个像素,并且在水平方向上偏移了(L-1)个像素水平方向。

[0065] 实施例2.根据本实施例,如方程(5)和(6)所示,通过修改位流一致性条件(1)和(2)来限制块向量(BV_x,BV_y):

$$[0066] \quad BV_x + nPbSw + xPb - xCb + L \leq 0 \text{ 或}$$

$$[0067] \quad BV_y + nPbSh + yPb - yCb + L \leq 0 \quad (5)$$

$$[0068] \quad (xPb + BV_x + nPbSw - 1 + L) / CtbSize - xCb / CtbSize < =$$

$$[0069] \quad yCb / CtbSize - (yPb + BV_y + nPbSh - 1 + L) / CtbSize \quad (6)$$

[0070] 与位流一致性条件(1)和(2)相比,根据上述实施例的可用参考像素边界在垂直方向上偏移L个像素,并在水平方向上偏移L个像素。

[0071] 实施例3.根据本实施例,如等式(7)和(8)所示,通过修改位流一致条件(1)和(2)来限制块向量(BV_x,BV_y),其中m是正整数。

$$[0072] \quad BV_x + nPbSw + xPb - xCb + (L+m) \leq 0 \text{ 或}$$

$$[0073] \quad BV_y + nPbSh + yPb - yCb + (L+m) \leq 0 \quad (7)$$

$$[0074] \quad (xPb + BV_x + nPbSw - 1 + (L+m)) / CtbSize - xCb / CtbSize < =$$

$$[0075] \quad yCb / CtbSize - (yPb + BV_y + nPbSh - 1 + (L+m)) / CtbSize \quad (8)$$

[0076] 与位流一致性条件(1)和(2)相比,根据上述实施例的可用参考像素边界在垂直方向上偏移(L+m)个像素,并且移位(L+m)个像素水平方向,其中m是具有正值,负值或零的整数。

[0077] 当块向量总是选择整数分辨率时,亮度像素不需要内插。然而,对于非4:4:4颜色格式的色度像素,通过缩放亮度块向量导出色度块向量。缩放后的色度块向量可能指向子像素(sub-pel)位置,并且需要内插。

[0078] 实施例4.根据本实施例,如等式(9)和(10)所示,通过修改以下位流一致性条件(1)和(2)来限制块向量(BV_x,BV_y):

[0079] $BV_x+nPbSw+xPb-xCb+(L-1)*2*(subWidthC-1) \leq 0$ 或

[0080] $BV_y+nPbSh+yPb-yCb+(L-1)*2*(subHeightC-1) \leq 0$ (9)

[0081] $(xPb+Bv_x+nPbSw-1+(L-1)*2*(subWidthC-1))/CtbSize-xCb/CtbSize \leq$

[0082] $yCb/CtbSize-(yPb+Bv_y+nPbSh-1+(L-1)*2*(subHeightC-1))/CtbSize$

[0083] (10)

[0084] 在上述等式中,subHeightC和subWidthC是与HEVC SCC中定义的色度格式相关的变量。subWidthC和subHeightC值是从chroma_format_idc和

[0085] separate_colour_plane_flag导出的,如表2所示。要内插的像素位置的每一侧的内插滤波器长度(即L)可以与现有的HEVC SCC草案标准中相同。

[0086] 表2

[0087]

chroma_f ormat_idc	separate_colou r_plane_flag	色度格 式	sub WidthC	sub HeightC
0	0	单色	1	1
1	0	4:2:0	2	2
2	0	4:2:2	2	1
3	0	4:4:4	1	1
3	1	4:4:4	1	1

[0088] 实施例5。根据本实施例,如等式(11)和(12)所示,通过修改以下位流一致性条件(1)和(2)来限制块向量(BV_x,BV_y):

[0089] $BV_x+nPbSw+xPb-xCb+L*2*(subWidthC-1) \leq 0$ 或

[0090] $BV_y+nPbSh+yPb-yCb+L*2*(subHeightC-1) \leq 0$ (11)

[0091] $(xPb+Bv_x+nPbSw-1+L*2*(subWidthC-1))/CtbSize-xCb/CtbSize \leq$

[0092] $yCb/CtbSize-(yPb+Bv_y+nPbSh-1+L*2*(subHeightC-1))/CtbSize$

[0093] (12)

[0094] 实施例6。根据第六实施例,如等式(13)和(14)所示,通过修改以下位流一致性条件(1)和(2)来限制块向量(BV_x,BV_y):

[0095] $BV_x+nPbSw+xPb-xCb+(L+m)*2*(subWidthC-1) \leq 0$ 或

[0096] $BV_y+nPbSh+yPb-yCb+(L+m)*2*(subHeightC-1) \leq 0$ (13)

[0097] $(xPb+Bv_x+nPbSw-1+(L+m)*2*(subWidthC-1))/CtbSize-xCb/CtbSize \leq$

[0098] $yCb/CtbSize-(yPb+Bv_y+nPbSh-1+(L+m)*2*(subHeightC-1))/CtbSize$ (14)

[0099] 在上述实施例中,只要分数像素块向量被使能,就修改比特比一致性条件。换句话说,无论块向量是指向整数位置还是分数像素位置,都会修改位流一致性条件。上述实施例的变型仅在块向量指向子像素位置时应用位流一致性条件修改。如果块向量具有分数像素分辨率,但实际上指向整数位置(即,小数部分等于0),则不需要上述位流一致性条件修改。

[0100] 方法2.使用相邻可用像素填充不可用像素

[0101] 根据该方法,使用参考区域中的相邻可用像素填充不可用像素。特别地,参考块中可用像素的最右边的列被水平地延伸以便对右侧的相邻像素进行填充。这些不可用像素是内插一些子像素所必需的。类似地,参考块中的可用像素的最底行垂直地延伸以填充底侧相邻像素。这些不可用像素是内插一些子像素所必需的。对于参考块右下角的子像素位置,内插时需要右侧或右下角的总共L个像素。令 $M(x)$ 为右侧可用的整数像素数, $M(y)$ 为右下方可用的整数像素数。如果 $M(x) < L$,则需要填充 $(L - M(x))$ 不可用列。如果 $M(y) < L$,则需要填充 $(L - M(y))$ 不可用行。在一个实施例中,先进行水平填充,然后进行垂直填充。在另一个实施例中,首先进行垂直填充,然后进行水平填充。

[0102] 图5示出了通过首先水平延伸(extend)最右边的列然后垂直延伸最下面的行来填充用于分数像素内插的不可用区域的示例。要内插的子像素参考像素510在图5中被示为菱形符号。内插所需的相邻参考像素由虚线框520包围(enclosed),虚线框520包括每侧4个像素(左、右、上、下)。粗线边界530表示可用和不可用的参考像素之间的边界,其中可用的参考像素在边界的左上侧。如图5所示,用于内插的一些参考像素不可用并且由三角形符号表示。为了填充这些不可用的参考像素,最右列(例如,像素542)中的可用参考像素被延伸到如箭头540所示的右侧,以产生不可用的参考像素(例如,像素544和546)。执行水平延伸后,执行垂直延伸。最下面的行中的可用参考像素(例如,像素552)如箭头550所示向下延伸以生成不可用的参考像素(例如,像素554和556)。在图5中, L 等于4, $M(x)$ 和 $M(y)$ 等于2,其中 $(L - M(x))$ 不可用列的水平填充数等于2, $(L - M(y))$ 不可用行的垂直填充数等于2。尽管图5中的示例示出了首先水平延伸然后是垂直延伸的情况,但是顺序也可以被交换。

[0103] 方法3.使用默认值或预定义值填充不可用像素

[0104] 根据该方法,使用默认值或预定义值填充不可用像素。在一个实施例中,整个当前图像被初始化为固定值 X 。例如,可以将 X 设置为 $(1 \ll \text{bit_depth}) \gg 1$,其中 bit_depth 表示像素的位深度(bit depth)。在上述示例中,将 X 设置为最大水平的一半,例如 $\text{bit_depth} = 8$ 时, $X = 128$ 。在另一示例中,将 X 设置为0。在另一示例中, X 可以是在CU、CTU、切片/图像或序列级别用信号标示的当前值。或者,可以在CU、CTU、切片/图像或序列级别导出 X 。

[0105] 在另一个实施例中,将当前CTU或CU初始化为固定值 X 。例如,可将 X 设置为 $(1 \ll \text{bit_depth}) \gg 1$,其中 bit_depth 是像素的位深度。在上述示例中,将 X 设置为最大电平的一半,例如 $\text{bit_depth} = 8$ 时, $X = 128$ 。在另一示例中,将 X 设置为0。在另一示例中, X 可以是在CU、CTU、切片/图像或序列级别。或者,可以在CU、CTU、切片/图像或序列级别导出 X 。

[0106] 方法4.以非4:4:4格式限制整数亮度BV的整数色度BV

[0107] 根据该方法,对于非4:4:4颜色格式,如果亮度块向量为整数精度,则相应的色度向量保持整数精度。因此,当通过缩放从亮度BV导出色度BV时,色度BV总是被转换为整数位置。因此,如果亮度块向量为整数精度,则不需要对亮度和色度分量进行任何内插。在一个实施例中,通过将色度向量分量四舍五入到 x 和 y 分量的最接近的整数来转换色度BV。例如,值为3.6的导出色度BV分量将四舍五入为4。值为-3.6的导出色度BV分量将四舍五入为-4。在另一个实施例中,对于两个分量,导出的色度向量分量被削减到小于或等于其值的最接近的整数。例如,值为3.6的导出色度BV分量将被四舍五入为3。将导出的值为-3.6的色度BV分量限制为-4。在另一个实施例中,对于两个分量,色度向量分量被削减到大于或等于导出的色度BV值的最接近的整数。例如,3.6将被限制(clip)为4。值为-3.6的导出的色度BV分量

将被限制为-3。对于所有上述操作,当色度向量使用1/8像素精度时,这些整数向左移动以匹配整数位置。例如,数值1以1/8-像素精度由 $(1 \ll 3) = 8$ 表示。

[0108] 方法5.约束编码器以确保参考像素的可用性

[0109] 为了执行像素内插,参考块的相邻像素应该是可用的。根据该方法,施加编码器约束以确保用于内插的那些像素的可用性。

[0110] 实施例1。根据本实施例,将以下位流一致性条件应用于块向量。位流一致性的要求是,当参考图像是当前图像时,亮度运动向量 $mvLX$ 应符合以下约束:

[0111] 当使用JCTVC-T1005 (Joshi等人,屏幕内容编解码测试模型3,ITU-TSG16WP3和ISO/IECJTC1/SC29/WG11视频编解码联合协作团队(JCT-VC),第20次会议:Geneva,CH,2015年2月10日至2月18日,文件:JCTVC-T1005)第6.4.1节中规定的z扫描顺序块可用性的推导过程调用 $(xCurr, yCurr)$ 设置为等于 (xCb, yCb) ,且相邻亮度位置 $(xNbY, yNbY)$ 设置等于 $(xPb + (mvLX[0] \gg 2), (yPb + mvLX[1] \gg 2))$ 作为输入时,输出应等于TRUE。(15)

[0112] 当使用JCTVC-T1005第6.4.1节中规定的z扫描顺序块可用性的推导过程调用 $(xCurr, yCurr)$ 设置为等于 (xCb, yCb) ,且相邻亮度位置 $(xNbY, yNbY)$ 设置等于 $(xPb + (mvLX[0] \gg 2) + nPbW - 1 + (L - 1), yPb + (mvLX[1] \gg 2) + nPbH - 1 + (L - 1))$ 作为输入时,输出应等于TRUE。(16)

[0113] 在上述约束中, $mvLX[0]$ 和 $mvLX[1]$ 是四分之一像素分辨率下的块向量的两个分量。 (xPb, yPb) 是当前预测块的左上角亮度样本位置。 (xCb, yCb) 是当前编解码块的左上角亮度采样位置。 $nPbW$ 和 $nPbH$ 是当前亮度预测块的宽度和高度。内插滤波器长度为 $2L$ 。例如, L 可以被设置为与JCTVC-T1005相同的值(即,亮度内插滤波器的 $L = 4$,色度内插滤波器的 $L = 2$)。

[0114] JCTVC-T1005第6.4.1节公开了一种检查z扫描顺序参考块的可用性的过程。如果进程返回“TRUE”,则表示z扫描顺序参考块可用。否则,z扫描顺序参考块不可用。

[0115] 与JCTVC-T1005的第6.4.1节中规定的约束相比,根据上述实施例的可用参考像素边界在垂直方向上偏移 $(L-1)$ 个像素并在水平方向上偏移 $(L-1)$ 个像素。根据上述实施例的可用参考像素边界在垂直和水平方向上移动多个像素,以确保用于内插或生成子像素像素(sub-pel pixels)的整数像素像素(integer-pel pixels)可用(即重构)。

[0116] 如果内插滤波器是对称的,则根据上述实施例,可用参考像素边界需要移位(在垂直方向上向上并在水平方向上向左)的像素数应至少等于滤波器抽头数量的一半。

[0117] 实施例2。根据本实施例,将以下位流一致性条件应用于块向量。位流一致性的要求是,当参考图像是当前图像时,亮度运动向量 $mvLX$ 应符合以下约束:

[0118] 当使用JCTVC-T1005第6.4.1节中规定的z扫描顺序块可用性的推导过程调用 $(xCurr, yCurr)$ 设置为等于 (xCb, yCb) ,且相邻亮度位置 $(xNbY, yNbY)$ 设置等于 $(xPb + (mvLX[0] \gg 2), (yPb + mvLX[1] \gg 2))$ 作为输入时,输出应等于TRUE。(17)

[0119] 当使用JCTVC-T1005第6.4.1节中规定的z扫描顺序块可用性的推导过程调用 $(xCurr, yCurr)$ 设置为等于 (xCb, yCb) ,且相邻亮度位置 $(xNbY, yNbY)$ 设置等于 $(xPb + (mvLX[0] \gg 2) + nPbW - 1 + L, yPb + (mvLX[1] \gg 2) + nPbH - 1 + L)$ 作为输入时,输出应等于TRUE。(18)

[0120] 与JCTVC-T1005的第6.4.1节中规定的约束相比,根据上述实施例的可用参考像素边界在垂直方向上偏移了 L 个像素,并在水平方向上偏移了 L 个像素。再次,根据上述实施例

的可用参考像素边界在垂直和水平方向上移动多个像素,以确保用于内插或生成子像素像素的整数像素像素可用(即重构)。

[0121] 实施例3。根据本实施例,将以下位流一致性条件应用于块向量。位流一致性的要求是,当参考图像是当前图像时,亮度运动向量 $mvLX$ 应符合以下约束:

[0122] 当使用JCTVC-T1005第6.4.1节中规定的z扫描顺序块可用性的推导过程调用 $(xCurr, yCurr)$ 设置为等于 (xCb, yCb) ,且相邻亮度位置 $(xNbY, yNbY)$ 设置等于 $(xPb+(mvLX[0]>>2), (yPb+mvLX[1]>>2))$ 作为输入时,输出应等于TRUE。(19)

[0123] 当使用JCTVC-T1005第6.4.1节中规定的z扫描顺序块可用性的推导过程调用 $(xCurr, yCurr)$ 设置为等于 (xCb, yCb) ,且相邻亮度位置 $(xNbY, yNbY)$ 设置等于 $(xPb+(mvLX[0]>>2)+nPbW-1+(L+m), yPb+(mvLX[1]>>2)+nPbH-1+(L+m))$ 作为输入时,输出应等于TRUE。(20)

[0124] 与JCTVC-T1005的第6.4.1节中规定的约束相比,根据上述实施例的可用参考像素边界在垂直方向上偏移 $(L+m)$ 个像素,并在水平方向上偏移 $(L+m)$ 个像素。根据上述实施例的可用参考像素边界在垂直和水平方向上移动多个像素,以确保用于内插或生成子像素像素的整数像素像素可用(即重构)。

[0125] 方法6.约束编码器以确保非4:4:4色度内插的参考像素的可用性

[0126] 当块向量总是选择整数分辨率时,亮度像素不需要内插。然而,对于非4:4:4颜色格式的色度像素,需要对亮度块向量进行缩放以导出色度块向量。有时,缩放后的色度块向量指向子像素位置,从而需要内插。

[0127] 为了执行色度像素内插,参考块的相邻像素应该是可用的。在下文中,一些编码器约束被强制以确保那些像素的可用性。

[0128] 实施例1。根据本实施例,将以下位流一致性条件应用于块向量。位流一致性的要求是,当参考图像是当前图像时,亮度运动向量 $mvLX$ 应符合以下约束:

[0129] 当使用JCTVC-T1005第6.4.1节中规定的z扫描顺序块可用性的推导过程调用 $(xCurr, yCurr)$ 设置为等于 (xCb, yCb) ,且相邻亮度位置 $(xNbY, yNbY)$ 设置等于 $(xPb+(mvLX[0]>>2), (yPb+mvLX[1]>>2))$ 作为输入时,输出应等于TRUE。(21)

[0130] 当使用JCTVC-T1005第6.4.1节中规定的z扫描顺序块可用性的推导过程调用 $(xCurr, yCurr)$ 设置为等于 (xCb, yCb) ,且相邻亮度位置 $(xNbY, yNbY)$ 设置等于 $(xPb+(mvLX[0]>>2)+nPbW-1+(L-1)*2*(subWidthC-1), yPb+(mvLX[1]>>2)+nPbH-1+(L-1)*2*(subHeightC-1))$ 作为输入时,输出应等于TRUE。(22)

[0131] 在上述等式中, $subHeightC$ 和 $subWidthC$ 是与HEVC SCC中定义的色度格式相关的变量。 $subWidthC$ 和 $subHeightC$ 值从 $chroma_format_idc$ 和 $separate_colour_plane_flag$ 导出,如表2所示。

[0132] 实施例2。根据本实施例,将以下位流一致性条件应用于块向量。位流一致性的要求是,当参考图像是当前图像时,亮度运动向量 $mvLX$ 应符合以下约束:

[0133] 当使用JCTVC-T1005第6.4.1节中规定的z扫描顺序块可用性的推导过程调用 $(xCurr, yCurr)$ 设置为等于 (xCb, yCb) ,且相邻亮度位置 $(xNbY, yNbY)$ 设置等于 $(xPb+(mvLX[0]>>2), (yPb+mvLX[1]>>2))$ 作为输入时,输出应等于TRUE。(23)

[0134] 当使用JCTVC-T1005第6.4.1节中规定的z扫描顺序块可用性的推导过程调用 $(xCurr, yCurr)$ 设置为等于 (xCb, yCb) ,且相邻亮度位置 $(xNbY, yNbY)$ 设置等于 $(xPb+(mvLX$

[0] >> 2) + nPbW - 1 + L * 2 * (subWidthC - 1), yPb + (mvLX[1] >> 2) + nPbH - 1 + L * 2 * (subHeightC - 1)) 作为输入时, 输出应等于TRUE。 (24)

[0135] 实施例3. 根据本实施例, 将以下位流一致性条件应用于块向量。位流一致性的要求是, 当参考图像是当前图像时, 亮度运动向量mvLX应符合以下约束:

[0136] 当使用JCTVC-T1005第6.4.1节中规定的z扫描顺序块可用性的推导过程调用(xCurr, yCurr) 设置为等于(xCb, yCb), 且相邻亮度位置(xNbY, yNbY) 设置等于(xPb + (mvLX[0] >> 2), (yPb + mvLX[1] >> 2)) 作为输入时, 输出应等于TRUE。 (25)

[0137] 当使用JCTVC-T1005第6.4.1节中规定的z扫描顺序块可用性的推导过程调用(xCurr, yCurr) 设置为等于(xCb, yCb), 且相邻亮度位置(xNbY, yNbY) 设置等于(xPb + (mvLX[0] >> 2) + nPbW - 1 + (L + m) * 2 * (subWidthC - 1), yPb + (mvLX[1] >> 2) + nPbH - 1 + (L + m) * 2 * (subHeightC - 1)) 作为输入时, 输出应等于TRUE。 (26)

[0138] 方法7. 以单色(monochrome color) 颜色格式内插

[0139] 当块向量总是整数, 并且颜色格式是单色(仅亮度分量) 时, 不需要对不可用像素应用填充。在一个实施例中, 当使用常数值A来初始化当前解码图像时(如表3中的伪代码所示), 其中Rec_Pixel[i, j] 是当前图像的位置(i, j) 处的重构像素。

[0140] 表3

```

If(chroma_format_idc != 3 && chroma_format_idc != 0)
{
    for (i = 0; i < pic_width_in_luma_samples / SubWidthC ;
i++)
[0141]   for (j = 0; j < pic_height_in_luma_samples / SubHeightC ; j++)
    {
        Rec_Pixel[i, j] = A;
    }
}

```

[0142] 使用填充的像素值以非4:4:4格式的色度内插

[0143] 当块向量总是选择整数分辨率时, 亮度像素不需要内插。然而, 对于非4:4:4颜色格式的色度像素, 需要对亮度块向量进行缩放以导出色度块向量。有时, 缩放后的色度块向量指向子像素位置, 从而需要内插。为了执行色度像素内插, 参考块的相邻像素应该可用。当参考块位于靠近参考区域边界的位置时, 一些相邻像素可能不可用。在下文中, 使用有效像素值填充不可用像素。一个例子如下所示。

[0144] 当不可用像素(x, y) 在图像边界之外时, 图像边界处的可用像素用于填充不可用像素。

[0145] 如果x在图像边界之外, 但y在图像边界里面, 则使用(x0, y) 处的像素填充这个不可用的像素, 其中x0是图像内部最靠近x的位置。

[0146] 如果y在图像边界之外, 但x在图像边界里面, 则使用(x, y0) 处的像素填充这个不

可用像素,其中y0图像内部是最接近y的位置。

[0147] 如果x和y都在图像边界之外,则使用(x0,y0)处的像素填充不可用像素,其中(x0,y0)是图像内部最接近(x,y)的位置。

[0148] 当不可用像素在参考区域之外但在图像内部时,使用常数值来填充不可用像素。在一个实施例中,使用值0;在另一个实施例中,使用值 $(1 \ll \text{bit_depth}) \gg 1$,其中bit_depth是像素的位深度。当bit_depth为8时,填充中使用的值为128。

[0149] 另一示例如下所示:

[0150] 使用常数值初始化整个当前图像。在一个实施例中,使用值0;在另一个实施例中,使用值 $(1 \ll \text{bit_depth}) \gg 1$,其中bit_depth是像素的位深度。当bit_depth为8时,填充中使用的值为128。

[0151] 在当前图像的解码过程中,重建像素逐渐地逐块填充当前图像,以代替这些常数值。

[0152] 对于指向图像边界的块向量,应用以下规则来在图像边界处使用像素来填充不可用像素:

$$[0153] \quad x_{Bi,j} = \text{Clip3}(0, (\text{pic_width_in_luma_samples}/\text{SubWidthC}) - 1, x_{\text{IntC}} + i) \quad (27)$$

[0154]

$$[0155] \quad y_{Bi,j} = \text{Clip3}(0, (\text{pic_height_in_luma_samples}/\text{SubHeightC}) - 1, y_{\text{IntC}} + j) \quad (28)$$

[0156]

[0157] 在上述等式中,pic_width_in_luma_samples和pic_height_in_luma_samples分别是图像的宽度和高度。SubWidthC和SubHeightC在表2中定义。xIntC和yIntC是参考块左上角像素的左上角的整数位置。i和j分别是通过参考块的两个维度,宽度和高度的索引。

[0158] 自适应运动/块向量分辨率处理

[0159] 本发明的另一方面解决了涉及自适应运动/块向量分辨率的问题。在JCTVC-T1014 (Joshi等人,屏幕内容编解码测试模型4 (SCM4),ITU-TSG16WP3和ISO/IECJTC1/SC29/WG11视频编码联合协作团队(JCT-VC),第20次会议:Geneva,CH,2015年2月10日至2月18日,文件:JCTVC-T1014),运动向量的解码过程描述如下。

[0160] 如果在当前切片中使能整数运动向量分辨率(即,use_integer_mv_flag等于1),则解码的运动向量被视为整数运动向量并以整数分辨率存储。在去块滤波过程中,运动向量处于整数分辨率。在内插过程中,运动向量左移两位,以避免分数像素补偿。

[0161] 如果在当前切片中禁用整数运动向量分辨率(即,use_integer_mv_flag等于0),则解码的运动向量被视为小数像素运动向量并存储为分数像素分辨率。在去块和内插过程中,不需要额外的操作。

[0162] 根据JCTVC-T1014的块向量的解码处理如下所述。

[0163] 解码块向量总是被视为整数块向量。然而,通过在解码之后左移两位,它以分数像素分辨率存储。在去块滤波器和内插滤波器的处理过程中,块向量处于分数像素分辨率,不需要额外的操作。

[0164] 实施例1。根据本实施例,对运动向量和块向量的自适应分辨率的解码处理如下进行修改。

[0165] 当使能整数运动向量分辨率(即,use_integer_mv_flag等于1)时,块向量和运动

向量都以整数分辨率解码。此外,块向量和运动向量都以分数像素分辨率(例如四分之一像素分辨率)存储。这通过将解码的块向量或运动向量左移正整数N(例如N=2)来完成。

[0166] 使用的块向量预测值和运动向量预测值是整数分辨率。这是通过将向量预测值向右移动正整数N(例如N=2)来完成的。

[0167] 当整数运动向量分辨率被禁用(即,use_integer_mv_flag等于0)时,块向量和运动向量都以分数像素分辨率被解码。此外,块向量和运动向量都以分数像素分辨率(例如四分之一像素分辨率)存储。

[0168] 使用的块向量预测值和运动向量预测值是分数像素分辨率(例如四分之一像素分辨率)。

[0169] 实施例2。根据本实施例,如下修改用于运动向量和块向量的自适应分辨率的解码处理。

[0170] 当使能整数运动向量分辨率(即,use_integer_mv_flag等于1)时,块向量和运动向量都以整数分辨率解码。此外,块向量和运动向量都以整数像素分辨率存储。当由去块滤波器用作输入时,块向量和运动向量被修改为分数像素分辨率,例如四分之一像素分辨率。这通过将解码的块向量或运动向量左移正整数N(例如N=2)来完成。

[0171] 当禁用整数运动向量分辨率(use_integer_mv_flag等于0)时,块向量和运动向量都以分数像素分辨率解码。并且块向量和运动向量都以分数像素分辨率(例如四分之一像素分辨率)存储。当块向量和运动向量被解块滤波器用作输入时,不需要进一步修改。

[0172] INTRABC的高级语法

[0173] 本发明的一个方面是IntraBC的高级语法。根据本发明的实施例,当前画面被放置在参考画面列表的最后位置。在以下的实施方式中,以列表0为例进行说明。如果当前图像作为参考图像放在列表1中,则可以对列表1进行类似的更改。

[0174] 实施例1。根据本实施例,在初始化之后,将索引值等于数组RefPicListTemp0中的num_ref_idx_l0_active_minus1的条目设置为当前图像。用于实现该实施例的示例性伪代码示于表4中。变量NumRpsCurrTempList0被设置为等于Max(num_ref_idx_l0_active_minus1+1,NumPicTotalCurr),如表4所示构造。

[0175] 表4

[0176]

```

    rIdx = 0
while( rIdx < NumRpsCurrTempList0 ) {
    for( i = 0; i < NumPocStCurrBefore  &&  rIdx <
NumRpsCurrTempList0; rIdx++, i++ )
        RefPicListTemp0[ rIdx ] = RefPicSetStCurrBefore[ i ]
    for( i = 0;  i < NumPocStCurrAfter  &&  rIdx <
NumRpsCurrTempList0; rIdx++, i++ )
        RefPicListTemp0[ rIdx ] = RefPicSetStCurrAfter[ i ]

    for( i = 0; i < NumPocLtCurr  &&  rIdx <
NumRpsCurrTempList0; rIdx++, i++ )
        RefPicListTemp0[ rIdx ] = RefPicSetLtCurr[ i ]
    if( curr_pic_as_ref_enabled_flag )
        RefPicListTemp0[ rIdx++ ] = currPic
}

```

[0177]

```

    if( curr_pic_as_ref_enabled_flag &&
num_ref_idx_l0_active_minus1 > 0 )
        RefPicListTemp0[ num_ref_idx_l0_active_minus1 ] = currPic

```

[0178] 列表RefPicList0如下构造:

[0179] for(rIdx=0; rIdx<=num_ref_idx_l0_active_minus1; rIdx++)

[0180] RefPicList0[rIdx]=ref_pic_list_modification_flag_l0?

[0181] RefPicListTemp0[list_entry_l0[rIdx]]:RefPicListTemp0[rIdx]

[0182] 实施例2。根据本实施例,在初始化之后,索引值等于阵列RefPicListTemp0中的num_ref_idx_l0_active_minus1的条目与存储当前图像的阵列中的条目交换。用于实现本实施例的示例性伪代码示于表5中。变量NumRpsCurrTempList0设置为等于Max(num_ref_idx_l0_active_minus1+1, NumPicTotalCurr), 并且列表RefPicListTemp0如表5所示构造。

[0183] 表5

[0184]

```

    rIdx = 0
while( rIdx < NumRpsCurrTempList0 ) {
    for( i = 0; i < NumPocStCurrBefore && rIdx <
NumRpsCurrTempList0; rIdx++, i++ )
        RefPicListTemp0[ rIdx ] = RefPicSetStCurrBefore[ i ]
    for( i = 0; i < NumPocStCurrAfter && rIdx <
NumRpsCurrTempList0; rIdx++, i++ )
        RefPicListTemp0[ rIdx ] = RefPicSetStCurrAfter[ i ]
    for( i = 0; i < NumPocLtCurr && rIdx <
NumRpsCurrTempList0; rIdx++, i++ )
        RefPicListTemp0[ rIdx ] = RefPicSetLtCurr[ i ]
    if( curr_pic_as_ref_enabled_flag )
        RefPicListTemp0[ rIdx++ ] = currPic
}
if( curr_pic_as_ref_enabled_flag && rIdx - 1 >
num_ref_idx_l0_active_minus1 && num_ref_idx_l0_active_minus1 > 0
) {

```

[0185]

```

    RefPicListTemp0[ rIdx - 1 ] =
RefPicListTemp0[ num_ref_idx_l0_active_minus1 ]
    RefPicListTemp0[ num_ref_idx_l0_active_minus1 ] = currPic
}

```

[0186] 列表RefPicList0的构建与实施例1相同。

[0187] 实施例3。根据本实施例，在初始化之后，将当前图像插入到索引值等于阵列RefPicListTemp0中的num_ref_idx_l0_active_minus1的条目中。变量NumRpsCurrTempList0设置为等于Max(num_ref_idx_l0_active_minus1+1, NumPicTotalCurr)，并且列表RefPicListTemp0如表6所示构造。

[0188] 表6

[0189]

```

    rIdx = 0
while( rIdx < NumRpsCurrTempList0 ) {
    for( i = 0; i < NumPocStCurrBefore && rIdx <
NumRpsCurrTempList0; rIdx++, i++ )
        RefPicListTemp0[ rIdx ] = RefPicSetStCurrBefore[ i ]
    for( i = 0; i < NumPocStCurrAfter && rIdx <
NumRpsCurrTempList0; rIdx++, i++ )
        RefPicListTemp0[ rIdx ] = RefPicSetStCurrAfter[ i ]

    for( i = 0; i < NumPocLtCurr && rIdx <
NumRpsCurrTempList0; rIdx++, i++ )
        RefPicListTemp0[ rIdx ] = RefPicSetLtCurr[ i ]
    if( curr_pic_as_ref_enabled_flag )
        RefPicListTemp0[ rIdx++ ] = currPic
}
if( curr_pic_as_ref_enabled_flag && rIdx - 1 >
num_ref_idx_l0_active_minus1 && num_ref_idx_l0_active_minus1 > 0
) {
    for( i = num_ref_idx_l0_active_minus1 + 1; i < rIdx; i++ )
        RefPicListTemp0[ i ] = RefPicListTemp0[ i - 1 ]
}
}

```

[0190]

```

    RefPicListTemp0[ num_ref_idx_l0_active_minus1 ] = currPic
}

```

[0191] 列表RefPicList0的构建与实施例1相同。

[0192] 图6示出了结合本发明的实施例的用于IntraBC(块内复制模式)编解码的视频编码器的示例性流程图。系统在步骤610中接收当前图像中的当前块。如步骤620所示,导出从当前块指向当前图像的重构像素区域内的参考块的块向量。如果当前块的分数像素分辨率被启用,并且块向量处于分数像素分辨率,则块向量被限制,使得用于生成参考块的分数像素参考像素的,在参考块的分数像素参考像素周围的整数像素位置处的相邻像素在重构像素区域内全部可用。在步骤630中,使用参考块作为预测值,根据IntraBC模式对当前块进行编码。

[0193] 图7示出了使用IntraBC模式(块内复制模式)编解码用于本发明实施例的图像的视频解码的示例性流程图。在步骤710中,系统接收包括当前图像中的当前块的编码数据的视频位流。在步骤720中,从视频位流中解析当前块向量。当前块向量从当前块指向当前图像的重构的像素区域。如果当前块向量处于分数像素分辨率,则使用内插滤波器,在分数像素位置为参考块生成分数像素参考像素,如步骤730所示。如果内插滤波器使用的任何滤波器输入数据是不可用的,不可用的滤波器输入数据通过内插滤波器填充进行内插。在步骤740中,使用参考块作为预测值,根据IntraBC模式对当前块进行解码。

[0194] 所示的流程图旨在说明根据本发明的IntraBC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、重新排列步骤、拆分步骤或组合步骤来实施本发明。在本公开中,已经使用具体的语法和语义来说明实现本发明的实施例的示例。本领域技术人员可以用等同的语法和语义来代替该语法和语义来实践本发明,而不脱离本发明的精神。

[0195] 呈现上述描述以使得本领域普通技术人员能够在特定应用及其要求的上下文中实施本发明。对所描述的实施例的各种修改对于本领域技术人员将是显而易见的,并且本文定义的一般原理可以应用于其他实施例。因此,本发明并不限于所示出和描述的特定实施例,而是符合与本文所公开的原理和新颖特征相一致的最宽范围。在上述详细描述中,示出了各种具体细节以便提供对本发明的透彻理解。然而,本领域技术人员将理解,可以实施本发明。

[0196] 如上所述的本发明的实施例可以以各种硬件、软件代码或两者的组合来实现。例如,本发明的实施例可以是集成到视频压缩芯片中的一个或多个电路或集成到视频压缩软件中的程序代码,以执行本文所述的处理。本发明的实施例也可以是要在数字信号处理器(DSP)上执行以执行本文所述的处理的程序代码。本发明还可以涉及由计算机处理器、数字信号处理器、微处理器或现场可编程门阵列(FPGA)执行的许多功能。可以通过执行定义本发明所体现的特定方法的机器可读软件代码或固件代码来将这些处理器配置成执行根据本发明的特定任务。软件代码或固件代码可以以不同的编程语言和不同的格式或风格开发。也可以为不同的目标平台编译软件代码。然而,根据本发明的不同的代码格式、软件代码的样式和语言以及配置代码的其他方式将不会脱离本发明的精神和范围。

[0197] 在不脱离本发明的精神或基本特征的情况下,本发明可以以其他具体形式实施。所描述的例子仅在所有方面被认为是说明性的而不是限制性的。因此,本发明的范围由所附权利要求而不是前面的描述来指示。属于权利要求的等同物的含义和范围内的所有变化将被包括在其范围内。

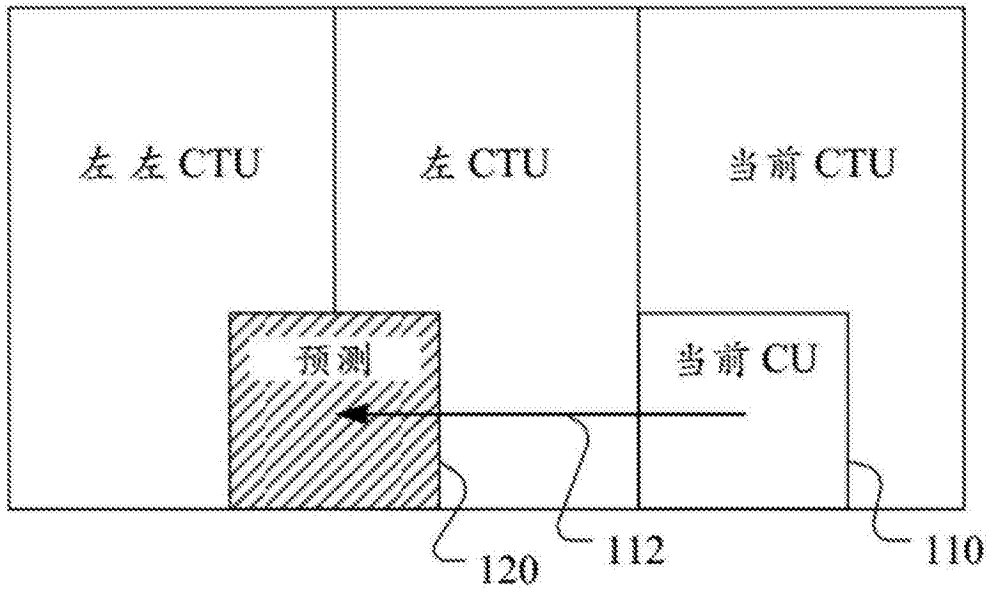


图1

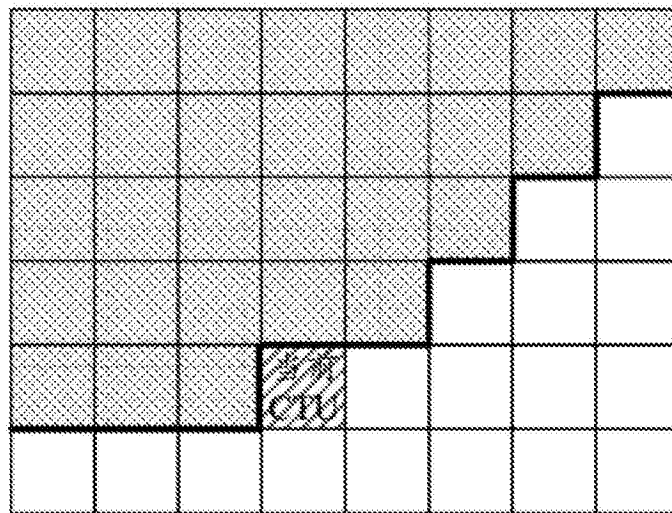


图2

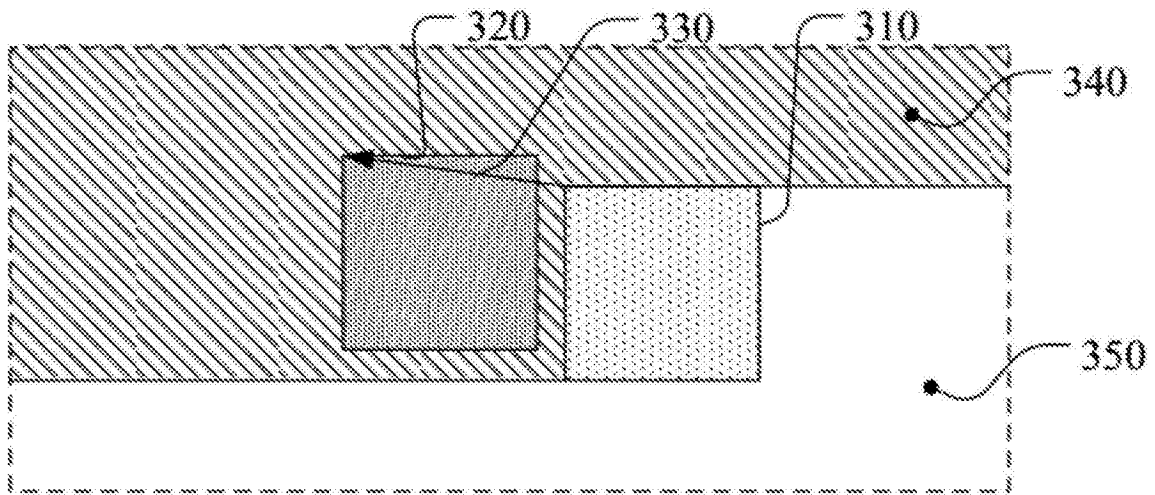


图3

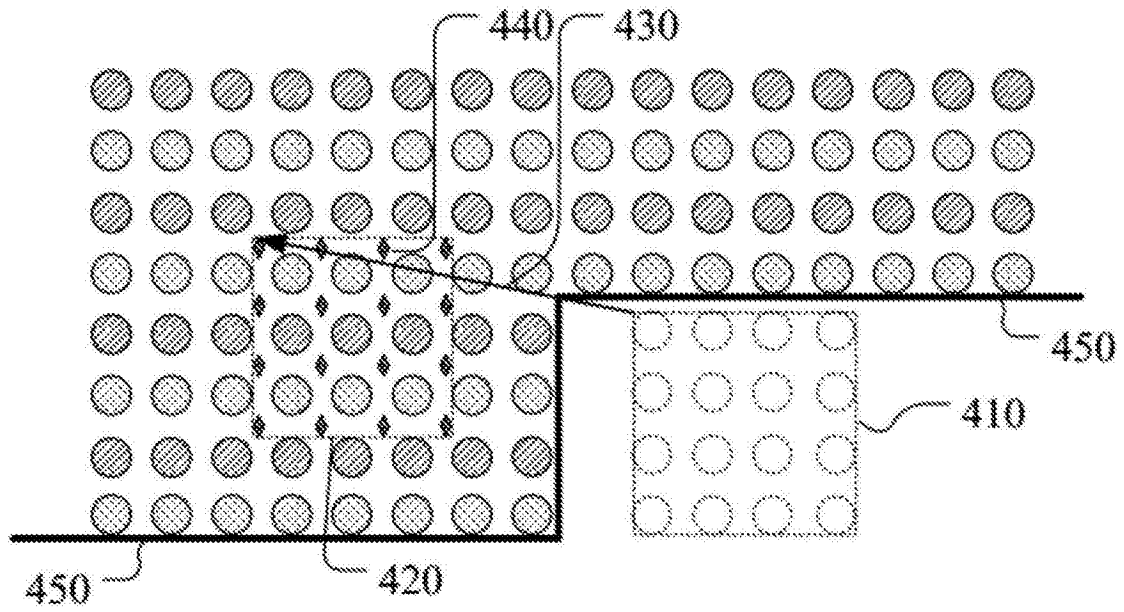


图4

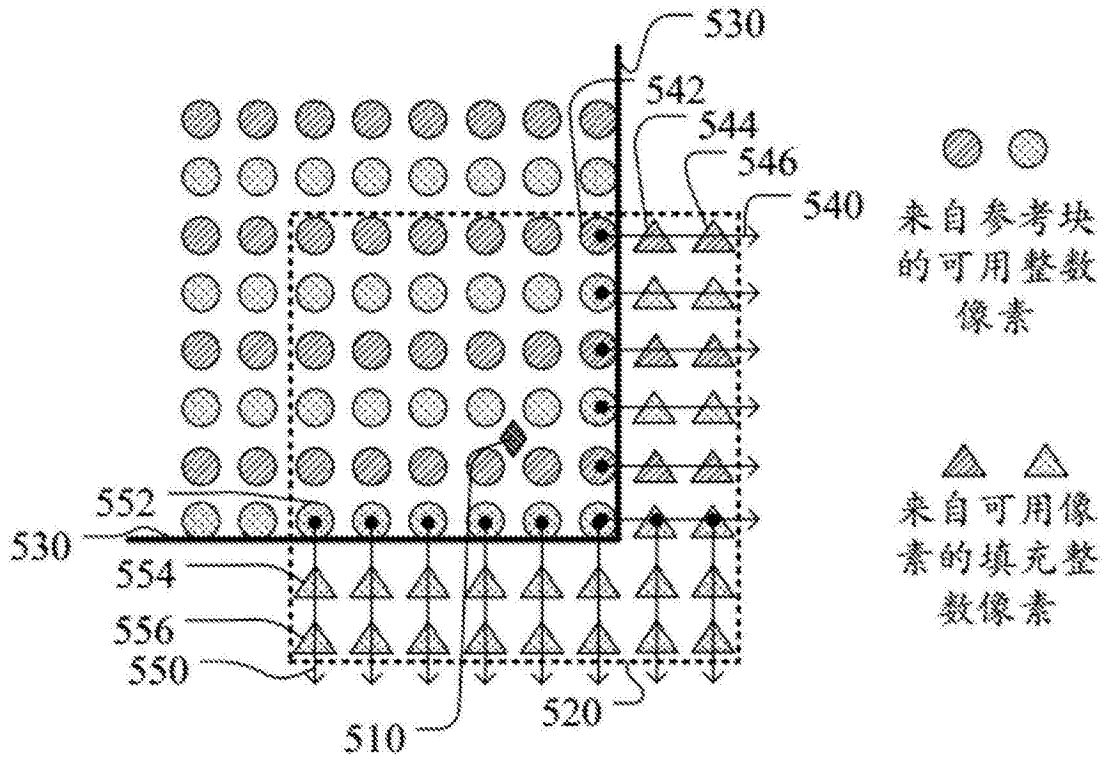


图5

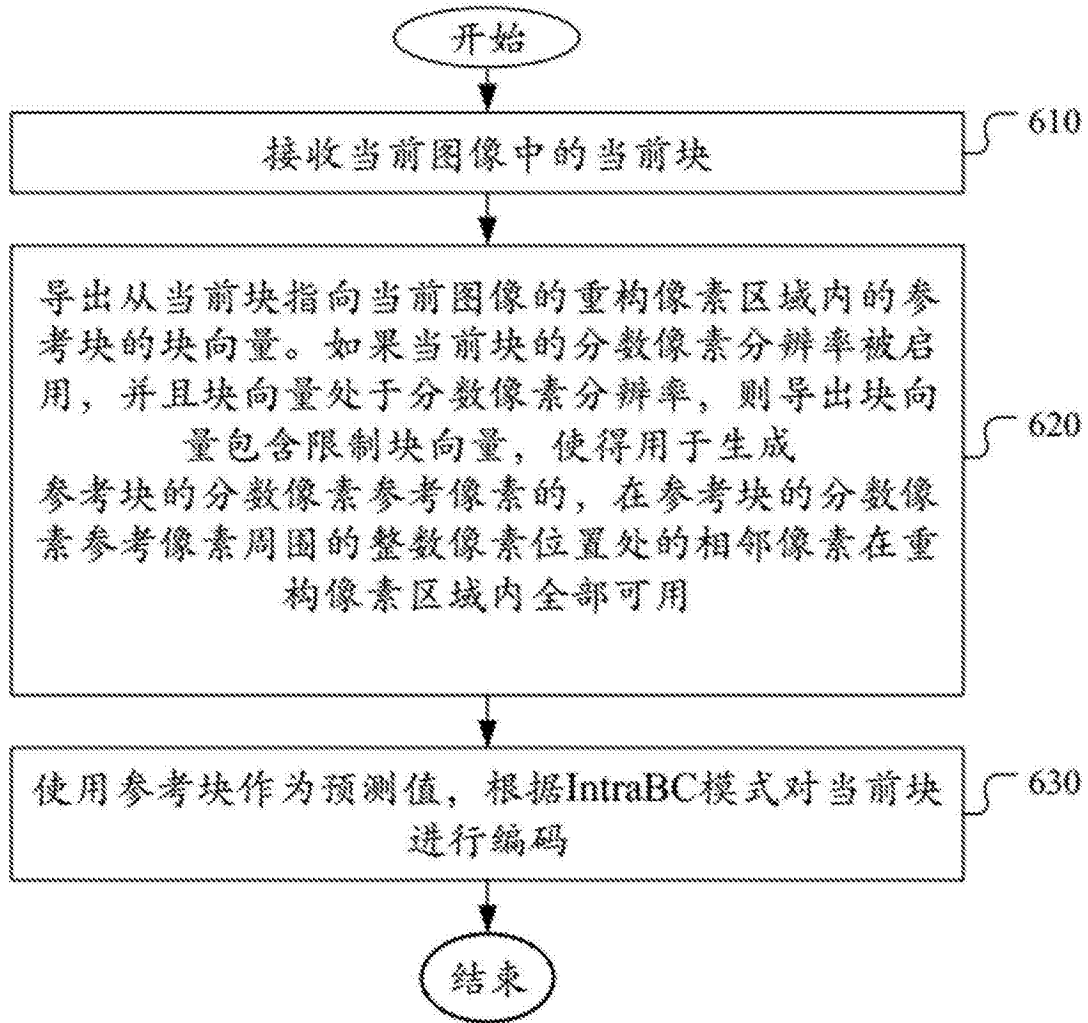


图6

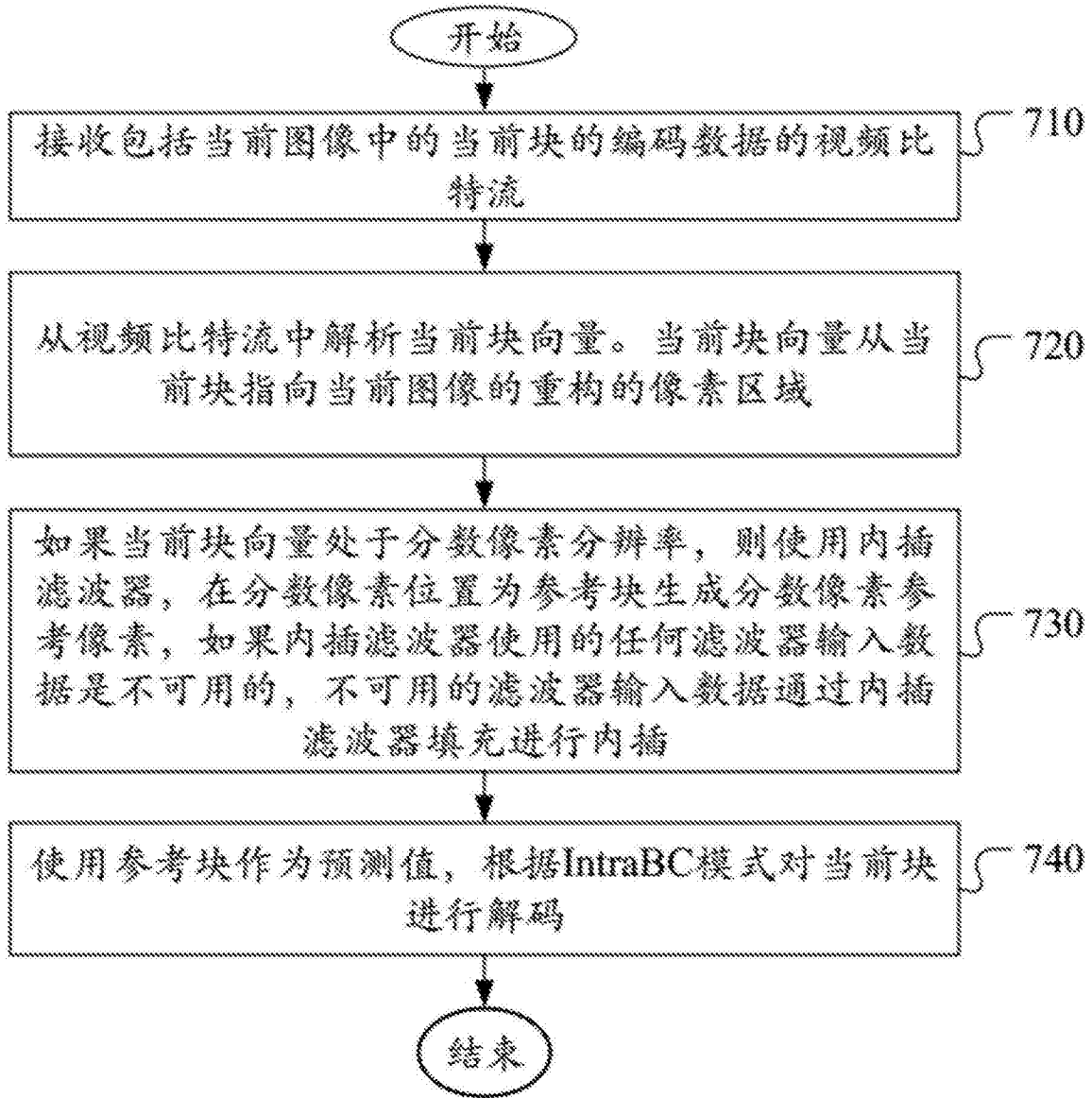


图7