



(12) 发明专利申请

(10) 申请公布号 CN 113170100 A

(43) 申请公布日 2021.07.23

(21) 申请号 201980069230.4

波夫

(22) 申请日 2019.12.30

瓦西里·亚历斯维奇·拉夫特斯基

陈建乐

(30) 优先权数据

62/786,349 2018.12.29 US

62/821,422 2019.03.20 US

(51) Int.Cl.

H04N 19/105 (2006.01)

(85) PCT国际申请进入国家阶段日

2021.04.20

(86) PCT国际申请的申请数据

PCT/RU2019/050262 2019.12.30

(87) PCT国际申请的公布数据

W02020/139173 EN 2020.07.02

(71) 申请人 华为技术有限公司

地址 518129 广东省深圳市龙岗区坂田华

为总部办公楼

(72) 发明人 阿列克谢·康斯坦丁诺维奇·菲利

权利要求书4页 说明书41页 附图23页

(54) 发明名称

用于帧内预测的方法和装置

(57) 摘要

本发明提供了对块进行帧内预测的装置和方法。所述对当前块进行帧内预测的方法包括：使用帧内预测模式对一个或多个参考像素点值进行帧内预测以获取预测像素点值；根据所述帧内预测模式，获取至少一个额外参考像素点值；根据所述额外参考像素点值，获取阈值化的额外参考像素点值。所述方法还包括：根据所述阈值化的额外参考像素点值，计算额外值；将所述预测像素点值与像素点加权因子相乘，得到加权预测像素点值；将所述加权预测像素点值与所述额外值相加，得到非归一化的预测像素点值；对所述非归一化的预测像素点值进行归一化，得到归一化的预测像素点值。本发明提高了所述帧内预测的准确性。

1. 一种对图像中的块进行帧内预测的方法,其特征在于,所述方法包括:
使用帧内预测模式对一个或多个参考像素点值进行帧内预测以获取预测像素点值;
获取位于所述预测像素点上方和左侧的最近参考像素点的值;
根据所述帧内预测模式,获取至少一个额外参考像素点值;
根据所述额外参考像素点值,获取阈值化的额外参考像素点值(thresholded additional reference sample value);
根据所述阈值化的额外参考像素点值,计算额外值;
将所述预测像素点值与像素点加权因子相乘,得到加权预测像素点值;
将所述加权预测像素点值与所述额外值相加,得到非归一化的预测像素点值;
对所述非归一化的预测像素点值进行归一化,得到归一化的预测像素点值。
2. 根据权利要求1所述的方法,其特征在于,所述帧内预测模式为垂直帧内预测模式,将所述额外参考像素点值设置为所述位于预测像素点上方的最近参考像素点的值与左上方参考像素点的值之间的差值。
3. 根据权利要求1所述的方法,其特征在于,所述帧内预测模式为水平帧内预测模式,将所述额外参考像素点值设置为所述位于预测像素点左侧的最近参考像素点的值与左上方参考像素点的值之间的差值。
4. 根据权利要求1至3中任一项所述的方法,其特征在于,所述获取阈值化的额外参考像素点值包括:
当所述帧内预测模式为水平帧内预测模式或垂直帧内预测模式时,对所述额外参考像素点值进行阈值化,得到所述阈值化的额外参考像素点值。
5. 根据权利要求1至3中任一项所述的方法,其特征在于,所述获取阈值化的额外参考像素点值包括:
获取左上方参考像素点值;
通过检查所述额外参考像素点值是否大于上限值或小于下限值来更新所述额外参考像素点值,以得到所述阈值化的额外参考像素点值,其中,根据所述左上方参考像素点值是否大于所述预测像素点值,选择所述上限值或所述下限值。
6. 根据权利要求5所述的方法,其特征在于,所述方法还包括:
推导所述预测像素点的最小值和最大值。
7. 根据权利要求6所述的方法,其特征在于,当所述左上方参考像素点值大于或等于所述预测像素点值时,所述上限值为所述预测像素点的最大值减去所述预测像素点值得到的差值,将所述阈值化的额外参考像素点值设置为第一值和第二值中的最大值,其中,
所述第一值为所述额外参考像素点值,
所述第二值为所述上限值。
8. 根据权利要求6所述的方法,其特征在于,当所述左上方参考像素点值小于所述预测像素点值时,所述下限值为所述预测像素点的最小值减去所述预测像素点值得到的差值,将所述阈值化的额外参考像素点值设置为第一值和第二值中的最小值,其中,
所述第一值为所述额外参考像素点值,
所述第二值为所述下限值。
9. 根据权利要求6至8中任一项所述的方法,其特征在于,根据图像参数集(picture

parameter set,PPS)值,推导所述预测像素点的最大值和最小值。

10. 根据权利要求6至9中任一项所述的方法,其特征在于,在条带头(slice header)中指示所述预测像素点的最大值和最小值。

11. 根据权利要求1至10中任一项所述的方法,其特征在于,所述根据所述阈值化的额外参考像素点值,计算额外值包括:

当所述帧内预测模式为水平帧内预测模式或垂直帧内预测模式时,通过将所述阈值化的额外参考像素点值与加权因子相乘来计算所述额外值。

12. 根据权利要求1至11中任一项所述的方法,其特征在于,当所述帧内预测模式为水平帧内预测模式或垂直帧内预测模式时,将所述像素点加权因子设置为1。

13. 根据权利要求1所述的方法,其特征在于,所述帧内预测模式为DC帧内预测模式,所述至少一个额外参考像素点值包括第一额外参考像素点值和第二额外参考像素点值;

通过以下方式获取所述第一额外参考像素点值和所述第二额外参考像素点值:

将所述第一额外参考像素点值设置为所述位于预测像素点左侧的最近参考像素点的值,

将所述第二额外参考像素点值设置为所述位于预测像素点上方的最近参考像素点的值。

14. 根据权利要求13所述的方法,其特征在于,所述计算额外值包括:

当所述帧内预测模式为DC帧内预测模式时,计算所述第一额外参考像素点值和所述第二额外参考像素点值的加权和,其中,将所述加权和设置为所述额外值。

15. 根据权利要求1至14中任一项所述的方法,其特征在于,所述图像为视频序列的一部分。

16. 一种对图像中的块进行帧内预测的方法,其特征在于,所述方法包括:

获取当前块的帧内预测模式;

当所述帧内预测模式为DC模式或平面模式时,根据上方像素点或左侧像素点,执行位置相关预测组合(position-dependent prediction combination,PDPC),其中,左上方像素点不用于位置相关预测组合(position-dependent prediction combination,PDPC)。

17. 根据权利要求16所述的方法,其特征在于,当predModeIntra等于INTRA_PLANAR或INTRA_DC时,

$$wT[y] = 32 \gg ((y \ll 1) \gg nScale),$$

$$wL[x] = 32 \gg ((x \ll 1) \gg nScale),$$

$$wTL[x][y] = 0,$$

其中,predModeIntra用于表示所述帧内预测模式,{x,y}限定预测像素点的位置,wT、wL和wTL为根据所述限定的位置与参考像素点相关的权重,nScale为缩放参数。

18. 一种对图像中的块进行帧内预测的方法,其特征在于,所述方法包括:

获取当前块的帧内预测模式;

当所述帧内预测模式为水平帧内预测模式或垂直帧内预测模式时,执行位置相关预测组合(position-dependent prediction combination,PDPC)中的修正运算,或者当所述帧内预测模式为DC模式或平面模式时,不执行位置相关预测组合(position-dependent prediction combination,PDPC)中的修正运算。

19. 根据权利要求18所述的方法, 其特征在于, 仅当所述帧内预测模式为水平帧内预测模式或垂直帧内预测模式时, 才执行PDPC中的修正运算。

20. 根据权利要求18或19所述的方法, 其特征在于, 当predModeIntra等于INTRA_ANGULAR18或INTRA_ANGULAR50时,

$$\text{predSamples}[x][y] = \text{clip1Cmp}((\text{refL}[x][y] \times \text{wL}[x] + \text{refT}[x][y] \times \text{wT}[y] - \text{p}[-1][-1] \times \text{wTL}[x][y] + (64 - \text{wL}[x] - \text{wT}[y] + \text{wTL}[x][y]) \times \text{predSamples}[x][y] + 32) \gg 6),$$

其中, predModeIntra用于表示所述帧内预测模式, $\{x, y\}$ 限定预测像素点的位置, wT、wL和wTL为根据所述限定的位置与参考像素点相关的权重。

21. 根据权利要求18或19所述的方法, 其特征在于, 当predModeIntra不等于INTRA_ANGULAR18或INTRA_ANGULAR50时,

$$\text{predSamples}[x][y] = (\text{refL}[x][y] \times \text{wL}[x] + \text{refT}[x][y] \times \text{wT}[y] + (64 - \text{wL}[x] - \text{wT}[y]) \times \text{predSamples}[x][y] + 32) \gg 6,$$

其中, predModeIntra用于表示所述帧内预测模式, $\{x, y\}$ 限定预测像素点的位置, wT和wL为根据所述限定的位置与参考像素点相关的权重。

22. 一种对图像中的块进行帧内预测的装置, 其特征在于, 所述装置包括:

获取单元, 用于: 使用帧内预测模式对一个或多个参考像素点值进行帧内预测以获取预测像素点值; 根据所述帧内预测模式, 获取至少一个额外参考像素点值; 根据所述额外参考像素点值, 获取阈值化的额外参考像素点值;

计算单元, 用于: 根据所述阈值化的额外参考像素点值, 计算额外值; 将所述预测像素点值与像素点加权因子相乘, 得到加权预测像素点值; 将所述加权预测像素点值与所述额外值相加, 得到非归一化的预测像素点值; 对所述非归一化的预测像素点值进行归一化, 得到归一化的预测像素点值。

23. 根据权利要求22所述的装置, 其特征在于, 所述帧内预测模式为垂直帧内预测模式, 将所述额外参考像素点值设置为位于预测像素点上方的最近参考像素点的值与左上方参考像素点的值之间的差值。

24. 根据权利要求22所述的装置, 其特征在于, 所述帧内预测模式为水平帧内预测模式, 将所述额外参考像素点值设置为位于预测像素点左侧的最近参考像素点的值与左上方参考像素点的值之间的差值。

25. 根据权利要求22所述的装置, 其特征在于, 所述帧内预测模式为DC帧内预测模式, 所述至少一个额外参考像素点值包括第一额外参考像素点值和第二额外参考像素点值;

通过以下方式获取所述第一额外参考像素点值和所述第二额外参考像素点值:

将所述第一额外参考像素点值设置为所述位于预测像素点左侧的最近参考像素点的值,

将所述第二额外参考像素点值设置为所述位于预测像素点上方的最近参考像素点的值。

26. 根据权利要求22至25中任一项所述的装置, 其特征在于, 当所述左上方参考像素点值大于或等于所述预测像素点值时, 上限值为所述预测像素点的最大值减去所述预测像素

点值得到的差值,将所述阈值化的额外参考像素点值设置为第一值和第二值中的最大值,其中,

所述第一值为所述额外参考像素点值,

所述第二值为所述上限值。

27. 根据权利要求22至25中任一项所述的装置,其特征在于,当所述左上方参考像素点值小于所述预测像素点值时,下限值为所述预测像素点的最小值减去所述预测像素点值得到的差值,将所述阈值化的额外参考像素点值设置为第一值和第二值中的最小值,其中,

所述第一值为所述额外参考像素点值,

所述第二值为所述下限值。

28. 一种对图像进行编码或解码的设备,其特征在于,所述设备包括处理电路,用于执行根据权利要求1至21中任一项所述的方法。

29. 根据权利要求28所述的设备,其特征在,所述处理电路包括一个或多个处理器以及与所述一个或多个处理器连接的非瞬时性计算机可读介质,其中,所述非瞬时性计算机可读介质包括程序代码;当所述一个或多个处理器执行所述程序代码时,所述设备执行所述方法。

30. 一种包括程序代码的非瞬时性计算机可读存储介质,其特征在于,当计算机设备执行所述程序代码时,所述计算机设备执行根据权利要求1至21中任一项所述的方法。

31. 一种解码器,其特征在于,所述解码器包括:

一个或多个处理器;

非瞬时性计算机可读存储介质,与所述一个或多个处理器耦合并存储由所述一个或多个处理器执行的程序,其中,当所述一个或多个处理器执行所述程序时,所述解码器用于执行根据权利要求1至21中任一项所述的方法。

32. 一种编码器,其特征在于,所述编码器包括:

一个或多个处理器;

非瞬时性计算机可读存储介质,与所述一个或多个处理器耦合并存储由所述一个或多个处理器执行的程序,其中,当所述一个或多个处理器执行所述程序时,所述编码器用于执行根据权利要求1至21中任一项所述的方法。

33. 一种非瞬时性记录介质,其特征在于,所述非瞬时性记录介质包括由一种装置解码的经预测编码的码流,所述码流是根据权利要求1至21中任一项生成的。

用于帧内预测的方法和装置

[0001] 相关申请交叉引用

[0002] 本申请要求于2018年12月29日提交的申请号为62/786,349、发明名称为“用于帧内预测的方法和装置(METHOD AND APPARATUS FOR INTRA PREDICTION)”的美国临时专利申请的权益,要求于2019年3月20日提交的申请号为62/821,422、发明名称为“用于帧内预测的方法和装置(METHOD AND APPARATUS FOR INTRA PREDICTION)”的美国临时专利申请的权益,其全部内容通过引用的方式并入本文中。

技术领域

[0003] 本发明涉及图像和/或视频编码和解码的技术领域,尤其涉及一种用于帧内预测的方法和装置。

背景技术

[0004] 自从DVD光盘问世以来,数字视频得到了广泛使用。视频在发送之前进行编码,然后通过传输介质进行发送。观看者接收视频,并使用观看设备来解码和显示视频。多年来,由于分辨率、色彩深度和帧率等的提高,视频质量已经得到了提高。这使得目前通过互联网和移动通信网络传输的数据流更大。

[0005] 但是,更高分辨率视频通常具有更多信息,因此需要更多带宽。为了降低带宽要求,便引入了涉及视频压缩的视频译码标准。当对视频进行编码时,带宽要求(或存储时对应的内存要求)会降低。这种降低往往牺牲了质量。因此,视频译码标准试图在带宽要求和质量之间找到平衡。

[0006] 高效视频译码(High Efficiency Video Coding,HEVC)是本领域技术人员所熟知的视频译码标准的一个示例。在HEVC中,译码单元(coding unit,CU)被划分为预测单元(prediction unit,PU)或变换单元(transform unit,TU)。通用视频编码(Versatile Video Coding,VVC)下一代标准是ITU-T视频编码专家组(Video Coding Experts Group,VCEG)和ISO/IEC运动图像专家组(Moving Picture Experts Group,MPEG)最近的联合视频项目。这两个标准化组织共同合作,其伙伴关系被称为联合视频探索小组(Joint Video Exploration Team,JVET)。VVC也称为ITU-T H.266/下一代视频译码(Next Generation Video Coding,NGVC)标准。VVC中去掉了多重分割类型的概念,即不区分CU、PU和TU概念(除非CU的大小对于最大变换长度而言太大),并支持更灵活的CU分割形状。

[0007] 对这些译码单元(coding unit,CU)(也称为块)的处理取决于它们的大小、空间位置和编码器指定的编码模式。根据预测类型,编码模式可以分为两类:帧内预测模式和帧间预测模式。帧内预测模式使用同一个图像(picture/image)(也称为帧)的像素点(sample)来生成参考像素点,计算重建块的像素点的预测值。帧内预测也称为空间预测。帧间预测模式设计用于时间预测,并使用先前或后续图像的参考像素点来预测当前图像中的块的像素点。

[0008] 需要用于视频译码的装置和方法,从而可以高效地处理帧内预测过程。

发明内容

[0009] 本发明提供了对图像中的块进行帧内预测的方法,编码设备,解码设备和计算机可读介质。

[0010] 在本发明第一方面的一个示例中,提供了一种对图像中的块进行帧内预测的方法。所述方法包括:

[0011] 使用帧内预测模式对一个或多个参考像素点值进行帧内预测以获取预测像素点值,其中,所述帧内预测模式为DC帧内预测模式、水平帧内预测模式或垂直帧内预测模式中的一种;

[0012] 获取位于预测像素点上方和左侧的最近参考像素点的值;

[0013] 根据所述帧内预测模式,获取一个或多个额外参考像素点值,其中,

[0014] -当所述帧内预测模式为垂直帧内预测模式时,将额外参考值设置为所述位于预测像素点上方的最近参考像素点的值与左上方参考像素点的值之间的差值,

[0015] -当所述帧内预测模式为水平帧内预测模式时,将额外参考值设置为所述位于预测像素点左侧的最近参考像素点的值与左上方参考像素点的值之间的差值,

[0016] -当所述帧内预测模式为DC帧内预测模式时,通过以下方式获取第一额外参考像素点值和第二额外参考像素点值:

[0017] ◦将所述第一额外参考像素点值设置为所述位于预测像素点左侧的最近参考像素点的值,

[0018] ◦将所述第二额外参考像素点值设置为所述位于预测像素点上方的最近参考像素点的值;

[0019] 当所述帧内预测模式为水平帧内预测模式或垂直帧内预测模式时,对所述额外参考像素点值进行阈值化;

[0020] 根据以下情况计算额外值:

[0021] ◦当所述帧内预测模式为DC帧内预测模式时,将所述额外值计算为所述第一额外参考像素点和所述第二额外参考像素点的加权和,或者

[0022] ◦当所述帧内预测模式为水平帧内预测模式或垂直帧内预测模式时,通过将所述额外参考值与加权因子相乘来计算所述额外值;

[0023] 将所述预测像素点值与像素点加权因子相乘,得到加权预测像素点值,其中,当所述帧内预测模式为水平帧内预测模式或垂直帧内预测模式时,将所述像素点加权因子设置为1;

[0024] 将所述加权预测像素点值与所述额外值相加,得到非归一化的预测像素点值;

[0025] 通过对所述非归一化的预测像素点值的整数表示进行算术右移,对所述非归一化的预测像素点值进行归一化,得到归一化的预测像素点值。

[0026] 在所述第一方面的一种实现方式中,所述方法还包括:推导所述预测像素点的最小值和最大值,其中,所述对所述额外参考像素点值进行阈值化包括:

[0027] 获取左上方参考像素点值;

[0028] 更新所述额外参考像素点值,包括检查所述额外参考像素点值是否大于上限值或小于下限值,其中,

[0029] -当所述左上方参考像素点值大于所述预测像素点值时,所述上限值为所述预测

像素点的最大值减去所述预测像素点值得到的差值,将所述更新后的额外参考像素点值设置为两个值中的最大值,其中,

[0030] o所述第一值为所述额外参考像素点值,

[0031] o所述第二值为所述上限值,

[0032] -否则,所述下限值为所述预测像素点的最小值减去所述预测像素点值得到的差值,

[0033] 将所述更新后的额外参考像素点值设置为两个值中的最小值,其中,

[0034] o所述第一值为所述额外参考像素点值,

[0035] o所述第二值为所述下限值。

[0036] 在本发明第二方面的一个示例中,提供了一种对图像中的块进行帧内预测的方法。所述方法包括:获取当前块的帧内预测模式;当所述帧内预测模式为DC模式或平面模式时,根据上方像素点或左侧像素点,执行位置相关预测组合(position-dependent prediction combination,PDPC),其中,左上方像素点不用于位置相关预测组合(position-dependent prediction combination,PDPC)。

[0037] 在本发明第三方面的一个示例中,提供了一种对图像中的块进行帧内预测的方法。所述方法包括:获取当前块的帧内预测模式;当所述帧内预测模式为水平帧内预测模式或垂直帧内预测模式时,执行位置相关预测组合(position-dependent prediction combination,PDPC)中的修正运算,或者当所述帧内预测模式为DC模式或平面模式时,不执行位置相关预测组合(position-dependent prediction combination,PDPC)中的修正运算。

[0038] 在所述第三方面的一种实现方式中,仅当所述帧内预测模式为水平帧内预测模式或垂直帧内预测模式时,才执行PDPC中的修正运算。例如,当predModeIntra等于INTRA_ANGULAR18或INTRA_ANGULAR50时,

[0039] $\text{predSamples}[x][y] = \text{clip1Cmp}((\text{refL}[x][y] \times wL[x] + \text{refT}[x][y] \times wT[y] - p[-1][-1] \times wTL[x][y] + (64 - wL[x] - wT[y] + wTL[x][y]) \times \text{predSamples}[x][y] + 32) \gg 6)$,

[0040] 其中,predModeIntra用于表示所述帧内预测模式,{x,y}限定预测像素点的位置,wT、wL和wTL为根据所述限定的位置与参考像素点相关的权重。

[0041] 所述第一方面或所述第一方面的任一可能实施例、所述第二方面或所述第二方面的任一可能实施例、所述第三方面或所述第三方面的任一可能实施例提供的方法可以由本发明第四方面提供的装置执行。

[0042] 根据第五方面,本发明涉及一种对视频流进行解码的装置。所述装置包括处理器和存储器。所述存储器存储指令,所述指令使得所述处理器执行所述第一方面或所述第一方面的任一可能实施例、所述第二方面或所述第二方面的任一可能实施例、所述第三方面或所述第三方面的任一可能实施例提供的方法。

[0043] 根据第六方面,本发明涉及一种对视频流进行编码的装置。所述装置包括处理器和存储器。所述存储器存储指令,所述指令使得所述处理器执行所述第一方面或所述第一方面的任一可能实施例、所述第二方面或所述第二方面的任一可能实施例、所述第三方面或所述第三方面的任一可能实施例提供的方法。

[0044] 根据第七方面,提供了一种存储有指令的计算机可读存储介质。所述指令在执行

时使得一个或多个处理器用于对视频数据进行译码。所述指令使得所述一个或多个处理器执行所述第一方面或所述第一方面的任一可能实施例、所述第二方面或所述第二方面的任一可能实施例、所述第三方面或所述第三方面的任一可能实施例提供的方法。

[0045] 根据第八方面,本发明涉及一种计算机程序。所述计算机程序包括程序代码,用于执行所述第一方面或所述第一方面的任一可能实施例、所述第二方面或所述第二方面的任一可能实施例、所述第三方面或所述第三方面的任一可能实施例提供的方法。

[0046] 以下附图和说明书详细阐述了一个或多个实施例。其它特征、目的和优点在说明书、附图和权利要求中是显而易见的。

附图说明

[0047] 下面将参照所附附图和示意图更加详细地描述本发明实施例,其中:

[0048] 图1A为用于实现本发明实施例的视频译码系统的一个示例的框图;

[0049] 图1B为用于实现本发明实施例的视频译码系统的另一个示例的框图;

[0050] 图2为用于实现本发明实施例的视频编码器的一个示例的框图;

[0051] 图3为用于实现本发明实施例的视频解码器的一种示例性结构的框图;

[0052] 图4为编码装置或解码装置的一个示例的框图;

[0053] 图5为编码装置或解码装置的另一个示例的框图;

[0054] 图6示出了角度帧内预测方向和模式以及垂直预测方向的 p_{ang} 的相关值;

[0055] 图7示出了 4×4 块的 p_{ref} 变换到 $p_{1,\text{ref}}$ 的一个示例;

[0056] 图8示出了构建用于水平角度预测的 $p_{1,\text{ref}}$ 的一个示例;

[0057] 图9示出了构建用于垂直角度预测的 $p_{1,\text{ref}}$ 的一个示例;

[0058] 图10A示出了JEM和BMS-1中的角度帧内预测方向和相关帧内预测模式的一个示例;

[0059] 图10B示出了VTM-2中的角度帧内预测方向和相关帧内预测模式的一个示例;

[0060] 图10C示出了VTM-3中的角度帧内预测方向和相关帧内预测模式的一个示例;

[0061] 图11示出了HEVC中的角度帧内预测方向和相关帧内预测模式的一个示例;

[0062] 图12示出了QTBT的一个示例;

[0063] 图13A示出了垂直和水平帧内预测模式的坐标的一个示例;

[0064] 图13B示出了 4×4 块内的位置(0,0)和(1,0)的DC模式PDPC权重的一个示例;

[0065] 图14示出了根据主参考边上的参考像素点对块进行帧内预测的一个示例;

[0066] 图15示出了 4×4 块内的位置(0,0)和(1,0)的DC模式PDPC权重的一个示例;

[0067] 图16示出了对像素点块进行帧内预测的一个示例;

[0068] 图17示出了帧内预测过程使用参考像素点来生成预测像素点的一个示例;

[0069] 图18为进行帧内预测来产生预测像素点的装置的一种示例性结构的框图;

[0070] 图19为实现内容分发服务的内容供应系统3100的一种示例性结构的框图;

[0071] 图20为终端设备的一个示例的结构的框图。

[0072] 在下文,相同附图标记表示相同特征或至少在功能上等价的特征,除非另有明确规定。

具体实施方式

[0073] 缩略语和术语定义

[0074] JEM Joint Exploration Model 联合探索模型(未来视频译码探索的软件代码库)

[0075] JVET Joint Video Experts Team 联合视频专家组

[0076] LUT Look-Up Table 查找表

[0077] PDPC Positiondependent prediction combination 位置相关预测组合

[0078] PPS Picture parameter set 图像参数集

[0079] QT QuadTree 四叉树

[0080] QTBT QuadTree plus Binary Tree 四叉树结合二叉树

[0081] RDO Rate-distortion Optimization 率失真优化

[0082] ROM Read-Only Memory 只读存储器

[0083] SPS Sequence parameter set 序列参数集

[0084] VTM Read-Only Memory VVC测试模型

[0085] VVC Versatile Video Coding 通用视频译码(JVET开发的标准化项目)

[0086] CTU/CTB Coding Tree Unit/Coding Tree Block 译码树单元/译码树块

[0087] CU/CB Coding Unit/Coding Block 译码单元/译码块

[0088] PU/PB Prediction Unit/Prediction Block 预测单元/预测块

[0089] TU/TB Transform Unit/Transform Block 变换单元/变换块

[0090] HEVC High Efficiency Video Coding 高效视频译码

[0091] 以下描述中,参考形成本发明一部分并以说明的方式示出本发明实施例的具体方面或可以使用本发明实施例的具体方面的附图。应理解,本发明实施例可以在其它方面中使用,并且可以包括附图中未描绘的结构变化或逻辑变化。因此,以下详细描述不应以限制性的意义来理解,且本发明的范围由所附权利要求书界定。

[0092] 例如,应理解,与描述方法有关的公开内容可以对用于执行所述方法的对应设备或系统也同样适用,反之亦然。例如,如果描述一个或多个具体方法步骤,则对应的设备可以包括一个或多个单元(例如功能单元)来执行所描述的一个或多个方法步骤(例如,一个单元执行一个或多个步骤,或多个单元分别执行多个步骤中的一个或多个),即使附图中未明确描述或说明这样的—个或多个单元。另一方面,例如,如果根据一个或多个单元(例如功能单元)来描述具体装置,则对应的方法可以包括一个步骤来执行一个或多个单元的功能(例如,一个步骤执行一个或多个单元的功能,或多个步骤分别执行多个单元中的一个或多个单元的功能),即使附图中未明确描述或说明这样的—个或多个步骤。此外,应理解,除非另外明确说明,本文中描述的各种示例性实施例和/或方面的特征可以相互组合。

[0093] 视频译码通常是指处理形成视频或视频序列的图像序列。在视频译码领域中,术语“帧(frame)”与“图像(picture/image)”可以用作同义词。视频译码(或总称为译码)包括视频编码和视频解码两部分。视频编码在源侧执行,通常包括处理(例如通过压缩)原始视频图像,以减少表示视频图像所需的数据量(从而更高效存储和/或发送)。视频解码在目的地侧执行,通常包括相对于编码器作逆处理,以重建视频图像。实施例涉及的视频图像(或总称为图像)的“译码”应理解为视频图像或相应视频序列的“编码”或“解码”。编码部分和

解码部分也合称为编解码(编码和解码, CODEC)。

[0094] 在无损视频译码情况下,可以重建原始视频图像,即重建视频图像与原始视频图像具有相同的质量(假设存储或发送期间没有传输损耗或其它数据丢失)。在有损视频译码情况下,通过量化等执行进一步压缩,来减少表示视频图像的数据量,而解码器侧无法完全重建视频图像,即重建视频图像的质量比原始视频图像的质量低或差。

[0095] 几个视频译码标准属于“有损混合型视频编解码器”组(即,将像素域中的空间预测和时间预测与变换域中用于应用量化的2D变换译码相结合)。视频序列中的每个图像通常分割成不重叠的块集合,通常在块级处执行译码。换句话说,在编码器侧,通常在块(视频块)级处对视频进行处理(即编码),例如,通过空间(帧内)预测和/或时间(帧间)预测来生成预测块;从当前块(当前处理的块/待处理块)中减去预测块,得到残差块;在变换域中变换残差块并量化残差块,以减少待发送(压缩)的数据量,而解码器侧对经编码或压缩的块进行相对于编码器的逆处理,以重建当前块进行表示。此外,编码器与解码器的处理步骤相同,使得编码器和解码器生成相同的预测块(例如帧内预测和帧间预测块)和/或重建块,以对后续块进行处理(即译码)。

[0096] 在以下视频译码系统10的实施例中,视频编码器20和视频解码器30根据图1至图3进行描述。

[0097] 图1A为示例性译码系统10的示意性框图,例如可以利用本申请技术的视频译码系统10(或简称为译码系统10)。视频译码系统10中的视频编码器20(或简称为编码器20)和视频解码器30(或简称为解码器30)为一种示例,可以为使用本申请中描述的各种示例来执行技术的设备。

[0098] 如图1A所示,译码系统10包括源设备12,源设备12用于将经编码的图像数据21提供给目标设备14等,以对经编码的图像数据13进行解码。

[0099] 源设备12包括编码器20,并且可以另外(即可选地)包括图像源16、预处理器(或预处理单元)18(例如图像预处理器18)和通信接口或通信单元22。

[0100] 图像源16可以包括或可以是任何类型的用于捕获真实世界图像等的图像捕获设备;和/或任何类型的图像生成设备(例如用于生成计算机动画图像的计算机图形处理器);或者任何类型的用于获取和/或提供真实世界图像、计算机动画图像(例如屏幕内容、虚拟现实(virtual reality, VR)图像)和/或其任何组合(例如增强现实(augmented reality, AR)图像)的设备。图像源可以为任何类型的存储任一上述图像的存储器(memory/storage)。

[0101] 为了区分预处理器18和预处理单元18执行的处理,图像或图像数据17也可以称为原始图像或原始图像数据17。

[0102] 预处理器18用于接收(原始)图像数据17,并对图像数据17执行预处理,得到预处理图像19或预处理图像数据19。预处理器18执行的预处理可以包括修剪、颜色格式转换(例如从RGB转换为YCbCr)、调色或去噪等。可以理解的是,预处理单元18可以为可选组件。

[0103] 视频编码器20用于接收预处理图像数据19并提供经编码的图像数据21(结合图2等描述更多细节)。

[0104] 源设备12中的通信接口22可以用于接收经编码的图像数据21,并通过通信信道13将经编码的图像数据21(或对经编码的图像数据21进一步处理后得到的数据)发送到另一

设备(例如目标设备14)或任何其它设备,以便进行存储或直接重建。

[0105] 目标设备14包括解码器30(例如视频解码器30),并且可以另外(即可选地)包括通信接口或通信单元28、后处理器32(或后处理单元32)和显示设备34。

[0106] 目标设备14中的通信接口28用于(例如)直接从源设备12或从存储设备(例如经编码的图像数据存储设备)等任何其它源,接收经编码的图像数据21(或对经编码的图像数据21进一步处理后得到的数据),并将经编码的图像数据21提供给解码器30。

[0107] 通信接口22和通信接口28可以用于经由源设备12与目标设备14之间的直接通信链路(例如直接有线或无线连接)或者经由任何类型的网络(例如有线网络、无线网络或其任何组合,或者任何类型的私网和公网或其任何类型的组合)发送或接收经编码的图像数据21或经编码的数据13。

[0108] 例如,通信接口22可以用于将经编码的图像数据21封装成合适的格式(例如数据包),和/或通过任何类型的传输编码或处理方式来处理经编码的图像数据,以便通过通信链路或通信网络进行传输。

[0109] 例如,与通信接口22对应的通信接口28可以用于接收传输数据,并通过任何类型的对应传输解码或处理和/或解封装方式来处理传输数据,得到经编码的图像数据21。

[0110] 通信接口22和通信接口28都可以配置为图1A中从源设备12指向目标设备14的通信信道13的箭头所指示的单向通信接口,或者配置为双向通信接口,并且可以用于发送和接收消息等,以建立连接、确认并交换与通信链路和/或数据传输(例如经编码的图像数据传输)相关的任何其它信息,等等。

[0111] 解码器30用于接收经编码的图像数据21并提供解码图像数据31或解码图像31(下文结合图3或图5等描述更多细节)。

[0112] 目标设备14中的后处理器32用于对解码图像数据31(也称为重建图像数据)(例如解码图像31)进行后处理,得到后处理图像数据33(例如后处理图像33)。后处理单元32执行的后处理可以包括颜色格式转换(例如从YCbCr转换为RGB)、调色、修剪或重采样,或者任何其它处理,以便提供解码图像数据31以供显示设备34等显示,等等。

[0113] 目标设备14中的显示设备34用于接收后处理图像数据33,以便向用户或观看者等显示图像。显示设备34可以为或可以包括任何类型的显示器(例如集成或外部显示器或显示屏),以表示重建图像。例如,显示器可以包括液晶显示器(liquid crystal display, LCD)、有机发光二极管(organic light emitting diode, OLED)显示器、等离子显示器、投影仪、微型LED显示器、硅基液晶显示器(liquid crystal on silicon, LCoS)、数字光处理器(digital light processor, DLP)或任何类型的其它显示器。

[0114] 尽管图1A示出了源设备12和目标设备14作为单独的设备,但是在实施例,设备还可以同时包括源设备12和目标设备14或同时包括源设备12和目标设备14的功能,即源设备12或对应功能以及目标设备14或对应功能。在这些实施例中,可以使用相同的硬件和/或软件或使用单独的硬件和/或软件或其任何组合来实现源设备12或对应功能以及目标设备14或对应功能。

[0115] 根据描述,图1A所示的源设备12和/或目的设备14中的不同单元或功能的存在和(精确)划分可以根据实际设备和应用而不同,这对技术人员来说是显而易见的。

[0116] 编码器20(例如视频编码器20)和解码器30(例如视频解码器30)可以分别实现为

图1B所示的各种合适电路中的任一种,例如一个或多个微处理器、一个或多个数字信号处理器(digital signal processor,DSP)、一个或多个专用集成电路(application-specific integrated circuit,ASIC)、一个或多个现场可编程门阵列(field-programmable gate array,FPGA)、一个或多个离散逻辑、一个或多个硬件或其任意组合。如果上述技术部分在软件中实现,则一种设备可以将该软件的指令存储在合适的非瞬时性计算机可读介质中,并且可以使用一个或多个处理器在硬件中执行这些指令,以执行本发明技术。上述各项(包括硬件、软件、硬件和软件的组合等)中的任一种可以认为是一个或多个处理器。视频编码器20和视频解码器30都可以包括在一个或多个编码器或解码器中,编码器或解码器可以作为组合编码器/解码器(encoder/decoder,CODEC)的一部分集成在相应的设备中。

[0117] 源设备12和目标设备14可以包括多种设备中的任一种,包括任何类型的手持设备或固定设备,例如笔记本电脑/膝上型电脑、手机、智能手机、平板或平板电脑、相机、台式电脑、机顶盒、电视机、显示设备、数字媒体播放器、视频游戏机、视频流设备(例如内容业务服务器或内容分发服务器)、广播接收器设备等,并且可以不使用或可以使用任何类型的操作系统。在一些情况下,源设备12和目标设备14可以用于无线通信。因此,源设备12和目标设备14可以是无线通信设备。

[0118] 在一些情况下,图1A所示的视频译码系统10仅仅是示例性的,本申请技术可以适用于编码设备与解码设备之间不一定包括任何数据通信的视频译码(例如视频编码或视频解码)设置。在其它示例中,从本地存储器中检索数据,通过网络发送,等等。视频译码设备可以对数据进行编码并将数据存储到存储器中,和/或视频解码设备可以从存储器中检索数据并对数据进行解码。在一些示例中,编码和解码由相互不通信而只是将数据编码到存储器和/或从存储器中检索数据并对数据进行解码的设备来执行。

[0119] 图1B为一个示例性实施例提供的示例性视频译码系统40的示意图。视频译码系统40包括图2中的编码器20和/或图3中的解码器30。系统40可以实现本申请中描述的各种示例提供的技术。在所说明的实现方式中,视频译码系统40可以包括一个或多个成像设备41、视频编码器100、视频解码器30(和/或通过一个或多个处理单元46中的逻辑电路47实现的视频译码器)、天线42、一个或多个处理器43、一个或多个存储器44和/或显示设备45。

[0120] 如图所示,一个或多个成像设备41、天线42、一个或多个处理单元46、逻辑电路47、视频编码器20、视频解码器30、一个或多个处理器43、一个或多个存储器44和/或显示设备45能够相互通信。如上所述,虽然示出了视频译码系统40包括视频编码器20和视频解码器30,但是在各种示例中,视频译码系统40可以只包括视频编码器20或只包括视频解码器30。

[0121] 如图所示,在一些示例中,视频译码系统40可以包括天线42。例如,天线42可以用于发送或接收视频数据的经编码的码流。此外,在一些示例中,视频译码系统40可以包括显示设备45。显示设备45可以用于呈现视频数据。如图所示,在一些示例中,逻辑电路47可以通过一个或多个处理单元46实现。一个或多个处理单元46可以包括专用集成电路(application-specific integrated circuit,ASIC)逻辑、一个或多个图形处理器、一个或多个通用处理器等。视频译码系统40还可以包括一个或多个可选的处理器43,这些处理器43可以类似地包括专用集成电路(application-specific integrated circuit,ASIC)逻辑、一个或多个图形处理器、一个或多个通用处理器等。在一些示例中,逻辑电路47可以

通过硬件或视频译码专用硬件等实现,一个或多个处理器43可以通过通用软件或操作系统等实现。另外,一个或多个存储器44可以是任何类型的存储器,例如易失性存储器(例如静态随机存取存储器(Static Random Access Memory,SRAM)、动态随机存取存储器(Dynamic Random Access Memory,DRAM)等)或非易失性存储器(例如闪存等),等等。在非限制性示例中,一个或多个存储器44可以通过超速缓存内存实现。在一些示例中,逻辑电路47可以访问一个或多个存储器44(用于实现图像缓冲器等)。在其它示例中,逻辑电路47和/或一个或多个处理单元46可以包括存储器(例如高速缓存等),用于实现图像缓冲器等。

[0122] 在一些示例中,通过逻辑电路实现的视频编码器20可以包括(例如通过一个或多个处理单元46或一个或多个存储器44实现的)图像缓冲器和(例如通过一个或多个处理单元46实现的)图形处理单元。该图形处理单元可以与图像缓冲器以通信方式耦合。该图形处理单元可以包括通过逻辑电路47实现的视频编码器20,以包含参照图2描述的各种模块和/或本文描述的任何其它编码器系统或子系统。逻辑电路可以用于执行本文描述的各种操作。

[0123] 视频解码器30可以类似地通过逻辑电路47实现,以包含参照图3中的解码器30描述的各种模块和/或本文描述的任何其它解码器系统或子系统。在一些示例中,通过逻辑电路实现的视频解码器30可以包括(例如通过一个或多个处理单元420或一个或多个存储器44实现的)图像缓冲器和(例如通过一个或多个处理单元46实现的)图形处理单元。该图形处理单元可以与图像缓冲器以通信方式耦合。该图形处理单元可以包括通过逻辑电路47实现的视频解码器30,以包含参照图3描述的各种模块和/或本文描述的任何其它解码器系统或子系统。

[0124] 在一些示例中,视频译码系统40中的天线42可以用于接收视频数据的经编码的码流。如上所述,经编码的码流可以包括与本文描述的与视频帧编码相关的数据、指示符、索引值、模式选择数据等,例如与译码分割相关的数据(例如变换系数或量化变换系数、可选指示符(如上所述)和/或定义译码分割的数据)。视频译码系统40还可以包括与天线42耦合并用于对经编码的码流进行解码的视频解码器30。显示设备45用于呈现视频帧。

[0125] 为便于描述,本文(例如)参考由ITU-T视频译码专家组(Video Coding Experts Group,VCEG)和ISO/IEC运动图像专家组(Motion Picture Experts Group,MPEG)的视频译码联合协作团队(Joint Collaboration Team on Video Coding,JCT-VC)开发的高效视频译码(High-Efficiency Video Coding,HEVC)或下一代视频译码标准通用视频译码(Versatile Video Coding,VVC)参考软件来描述本发明实施例。本领域普通技术人员理解本发明实施例不限于HEVC或VVC。

[0126] 编码器和编码方法

[0127] 图2为用于实现本申请技术的示例性视频编码器20的示意性框图。在图2的示例中,视频编码器20包括输入端201(或输入接口201)、残差计算单元204、变换处理单元206、量化单元208、反量化单元210、逆变换处理单元212、重建单元214、环路滤波器单元220、解码图像缓冲器(decoded picture buffer,DPB)230、模式选择单元260、熵编码单元270和输出端272(或输出接口272)。模式选择单元260可以包括帧间预测单元244、帧内预测单元254和分割单元262。帧间预测单元244可以包括运动估计单元和运动补偿单元(未示出)。图2所示的视频编码器20也可以称为混合视频编码器或基于混合型视频编解码器的视频编码器。

[0128] 残差计算单元204、变换处理单元206、量化单元208和模式选择单元260可以组成编码器20的前向信号路径,而反量化单元210、逆变换处理单元212、重建单元214、缓冲器216、环路滤波器220、解码图像缓冲器(decoded picture buffer,DPB)230、帧间预测单元244和帧内预测单元254可以组成视频编码器20的后向信号路径,其中,视频编码器20的后向信号路径对应于解码器(参见图3中的视频解码器30)的信号路径。反量化单元210、逆变换处理单元212、重建单元214、环路滤波器220、解码图像缓冲器(decoded picture buffer,DPB)230、帧间预测单元244和帧内预测单元254还组成视频编码器20的“内置解码器”。

[0129] 图像和图像分割(图像和块)

[0130] 编码器20可以用于通过输入端201等接收图像17(或图像数据17)。图像17可以是形成视频或视频序列的图像序列中的图像。接收到的图像或图像数据也可以是预处理图像19(或预处理图像数据19)。为了简单起见,以下描述使用图像17。图像17也可以称为当前图像或待译码图像(尤其是在视频译码中将当前图像与同一视频序列(也就是同样包括当前图像的视频序列)中的其它图像(例如先前的编码和/或解码图像)区分开)。

[0131] (数字)图像为或可以视为具有强度值的像素点组成的二维阵列或矩阵。阵列中的像素点也可以称为像素(pixel/pel)(图像元素的简称)。阵列或图像的水平方向和垂直方向(或轴线)上的像素点数量限定了图像的大小和/或分辨率。为了表示颜色,通常采用三个颜色分量,即图像可以表示为或包括三个像素点阵列。在RGB格式或颜色空间中,图像包括对应的红色、绿色和蓝色像素点阵列。但是,在视频译码中,每个像素通常以亮度和色度格式或颜色空间表示,例如YCbCr,包括Y表示的亮度分量(有时也用L表示)以及Cb和Cr表示的两个色度分量。亮度(luminance,简称为luma)分量Y表示亮度或灰度级强度(例如在灰度等级图像中两者相同),而两个色度(chrominance,简称为chroma)分量Cb和Cr表示色度或颜色信息分量。相应地,YCbCr格式的图像包括亮度像素点值(Y)的亮度像素点阵列和色度值(Cb和Cr)的两个色度像素点阵列。RGB格式的图像可以转换或变换为YCbCr格式,反之亦然。该过程也称为颜色变换或转换。如果图像是黑白的,则该图像可以只包括亮度像素点阵列。相应地,图像可以为例如黑白格式的亮度像素点阵列或4:2:0、4:2:2和4:4:4颜色格式的亮度像素点阵列和两个对应的色度像素点阵列。

[0132] 在实施例中,视频编码器20可以包括图像分割单元(图2中未示出),用于将图像17分割成多个(通常不重叠)图像块203。这些块也可以称为根块、宏块(H.264/AVC),或译码树块(coding tree block,CTB)或译码树单元(coding tree unit,CTU)(H.265/HEVC和VVC)。图像分割单元可以用于对视频序列中的所有图像使用相同的块大小和使用限定块大小的对应网格,或者在图像或图像子集或图像组之间改变块大小,并将每个图像分割成对应块。

[0133] 在其它实施例中,视频编码器可以用于直接接收图像17中的块203,例如组成图像17的一个、几个或所有块。图像块203也可以称为当前图像块或待译码图像块。

[0134] 与图像17一样,图像块203同样是或可以视为具有强度值(像素点值)的像素点组成的二维阵列或矩阵,但是图像块203的尺寸比图像17的尺寸小。换句话说,块203可以包括(例如)一个像素点阵列(例如黑白图像17情况下的亮度阵列或者彩色图像情况下的亮度阵列或色度阵列)或三个像素点阵列(例如彩色图像17情况下的一个亮度阵列和两个色度阵列)或根据所采用的颜色格式的任何其它数量和/或类型的阵列。块203的水平方向和垂直

方向(或轴线)上的像素点数量限定了块203的大小。相应地,一个块可以为 $M \times N$ (M 列 \times N 行)个像素点阵列,或 $M \times N$ 个变换系数阵列等。

[0135] 在实施例中,图2所示的视频编码器20可以用于逐块对图像17进行编码,例如对每个块203执行编码和预测。

[0136] 残差计算

[0137] 残差计算单元204用于通过如下方式根据图像块203和预测块265(后续详细介绍了预测块265)来计算残差块205(也称为残差205)以得到像素域中的残差块205:例如,逐个像素点(逐个像素)从图像块203的像素点值中减去预测块265的像素点值。

[0138] 变换

[0139] 变换处理单元206可以用于对残差块205的像素点值进行离散余弦变换(discrete cosine transform,DCT)或离散正弦变换(discrete sine transform,DST)等变换,得到变换域中的变换系数207。变换系数207也可以称为变换残差系数,表示变换域中的残差块205。

[0140] 变换处理单元206可以用于应用DCT/DST的整数近似,例如为H.265/HEVC指定的变换。与正交DCT变换相比,这种整数近似通常通过某一因子按比例缩放(scale)。为了维持经过正变换和逆变换处理的残差块的范数,使用其它缩放因子作为变换过程的一部分。缩放因子通常是某些约束条件来选择的,例如缩放因子是用于移位运算的2的幂、变换系数的位深度、准确性与实现成本之间的权衡等。例如,通过逆变换处理单元212等为逆变换(以及在视频解码器30侧通过逆变换处理单元312等为对应的逆变换)指定具体的缩放因子;相应地,可以在编码器20侧,通过变换处理单元206等为正变换指定对应的缩放因子。

[0141] 在实施例中,视频编码器20(对应地,变换处理单元206)可以用于输出一种或多种变换的类型等变换参数,例如直接输出或由熵编码单元270进行编码或压缩后输出,使得(例如)视频解码器30可以接收并使用变换参数进行解码。

[0142] 量化

[0143] 量化单元208用于通过标量量化或矢量量化等对变换系数207进行量化,得到量化变换系数209。量化系数209也可以称为量化变换系数209或量化残差系数209。

[0144] 量化过程可以减少与部分或全部变换系数207相关的位深度。例如,可以在量化期间将 n 位变换系数向下舍入到 m 位变换系数,其中, n 大于 m 。可以通过调整量化参数(quantization parameter,QP)修改量化程度。例如,对于标量量化,可以应用不同程度的缩放来实现较细或较粗的量化。较小量化步长对应于较细量化,而较大量化步长对应于较粗量化。可以通过量化参数(quantization parameter,QP)表示合适的量化步长。例如,量化参数可以为合适的量化步长的预定义集合的索引。例如,较小的量化参数可以对应于精细量化(较小量化步长),较大的量化参数可以对应于粗糙量化(较大量化步长),反之亦然。量化可以包括除以量化步长,而反量化单元210等执行的对应反量化可以包括乘以量化步长。根据HEVC等一些标准的实施例可以使用量化参数来确定量化步长。一般而言,可以根据量化参数使用包含除法的等式的定点近似来计算量化步长。可以引入其它缩放因子来进行量化和解量化,以恢复可能由于在量化步长和量化参数的等式的定点近似中使用的缩放而修改的残差块的范数。在一种示例性实现方式中,可以合并逆变换和解量化的缩放。或者,可以使用自定义量化表,并在码流等中将量化表从编码器通过信号发送到解码器。量化是

有损操作,其中,量化步长越大,损耗越大。

[0145] 在实施例中,视频编码器20(对应地,量化单元208)可以用于输出量化参数(quantization parameters,QP),例如直接输出或由熵编码单元270进行编码后输出,使得(例如)视频解码器30可以接收并使用量化参数进行解码。

[0146] 反量化

[0147] 反量化单元210用于对量化系数进行量化单元208的反量化,得到解量化系数211,例如根据或使用与量化单元208相同的量化步长,执行与量化单元208所执行的量化方案相反的反量化方案。解量化系数211也可以称为解量化残差系数211,对应于变换系数207,但是由于量化造成损耗,解量化系数211通常与变换系数不同。

[0148] 逆变换

[0149] 逆变换处理单元212用于进行变换处理单元206进行的变换的逆变换,例如逆离散余弦变换(discrete cosine transform,DCT)或逆离散正弦变换(discrete sine transform,DST),得到像素域中的重建残差块213(或对应的解量化系数213)。重建残差块213也可以称为变换块213。

[0150] 重建

[0151] 重建单元214(例如加法器或求和器214)用于通过如下方式将变换块213(即重建残差块213)添加到预测块265以得到像素域中的重建块215:例如,逐个像素点将重建残差块213的像素点值和预测块265的像素点值相加。

[0152] 滤波

[0153] 环路滤波器单元220(或简称“环路滤波器”220)用于对重建块215进行滤波,得到滤波块221,或一般用于对重建像素点进行滤波,得到滤波像素点。例如,环路滤波器单元用于顺利进行像素转变或提高视频质量。环路滤波器单元220可以包括一个或多个环路滤波器,例如去块效应滤波器、像素点自适应偏移(sample-adaptive offset,SAO)滤波器或者一个或多个其它滤波器,例如双边滤波器、自适应环路滤波器(adaptive loop filter,ALF)、锐化或平滑滤波器、协同滤波器,或其任意组合。虽然环路滤波器单元220在图2中示为环内滤波器,但是在其它配置中,环路滤波器单元220可以实现为环后滤波器。滤波块221也可以称为滤波重建块221。在环路滤波器单元220对重建译码块进行滤波操作之后,解码图像缓冲器230可以存储重建译码块。

[0154] 在实施例中,视频编码器20(对应地,环路滤波器单元220)可以用于输出环路滤波器参数(例如像素点自适应偏移信息),例如直接输出或由熵编码单元270进行编码后输出,使得(例如)解码器30可以接收并使用相同的环路滤波器参数或相应的环路滤波器进行解码。

[0155] 解码图像缓冲器

[0156] 解码图像缓冲器(decoded picture buffer,DPB)230可以是存储参考图像或一般存储参考图像数据以供视频编码器20在对视频数据进行编码时使用的存储器。DPB 230可以由多种存储器设备中的任一种形成,例如动态随机存取存储器(dynamic random access memory,DRAM),包括同步DRAM(synchronous DRAM,SDRAM)、磁阻RAM(magnetoresistive RAM,MRAM)、电阻RAM(resistive RAM,RRAM)或其它类型的存储器设备。解码图像缓冲器(decoded picture buffer,DPB)230可以用于存储一个或多个滤波块221。解码图像缓冲器

230还可以用于存储同一个当前图像或不同图像(例如先前的重建图像)中的其它先前滤波块(例如先前滤波重建块221),并可以提供先前完整的重建(即解码)图像(和对应的参考块和像素点)和/或部分重建的当前图像(和对应的参考块和像素点),以进行帧间预测等。如果重建块215未由环路滤波器单元220进行滤波,则解码图像缓冲器(decoded picture buffer,DPB)230还可以用于存储一个或多个未经滤波的重建块215,或一般存储未经滤波的重建像素点,或未进行任何其它处理的重建块或重建像素点。

[0157] 模式选择(分割和预测)

[0158] 模式选择单元260包括分割单元262、帧间预测单元244和帧内预测单元254,并且用于从解码图像缓冲器230或其它缓冲器(例如行缓冲器,图中未示出)接收或获取原始块203(当前图像17中的当前块203)等原始图像数据以及重建图像数据(例如同一个(当前)图像和/或一个或多个先前的解码图像中的经过滤波和/或未经滤波的重建像素点或块)。重建图像数据用作帧间预测或帧内预测等预测所需的参考图像数据,得到预测块265或预测值265。

[0159] 模式选择单元260可以用于为当前块预测模式(包括不分割)确定或选择一种分割以及确定或选择一种预测模式(例如帧内预测模式或帧间预测模式),生成对应的预测块265,以对残差块205进行计算和对重建块215进行重建。

[0160] 在实施例中,模式选择单元260可以用于选择分割和预测模式(例如从模式选择单元260支持的或可用的预测模式中选择),所述预测模式提供最佳匹配或者说最小残差(最小残差是指传输或存储中更好的压缩),或者提供最小信令开销(最小信令开销是指传输或存储中更好的压缩),或者同时考虑或平衡以上两者。模式选择单元260可以用于根据率失真优化(rate distortion optimization,RDO)确定分割和预测模式,即选择提供最小率失真的预测模式。本文中的“最佳”、“最小”、“最优”等术语不一定指总体上“最佳”、“最小”、“最优”等,但也可以指满足终止或选择标准的情况,例如,超过或低于阈值的值或其它约束条件可能导致“次优选择”,但会降低复杂度且减少处理时间。

[0161] 换句话说,分割单元262可以用于通过如下方式将块203分割成较小的块分割部分或子块(再次形成块):例如,通过迭代使用四叉树(quad-tree,QT)分割、二叉树(binary-tree,BT)分割或三叉树(triple-tree,TT)分割或其任意组合,并且用于(例如)对块分割部分或子块中的每一个执行预测,其中,模式选择包括选择分割块203的树结构和选择块分割部分或子块中的每一个所使用的预测模式。

[0162] 下文将详细地描述由示范性视频编码器20执行的分割(例如由分割单元260执行)和预测处理(由帧间预测单元244和帧内预测单元254执行)。

[0163] 分割

[0164] 分割单元262可以将当前块203分割(或划分)成更小的分割部分,例如正方形或矩形大小的较小块。可以将这些较小块(也可以称为子块)进一步分割成甚至更小的分割部分。这也称为树分割或分层树分割。在根树级别0(层次级别0、深度0)等的根块可以递归地分割成两个或更多下一个较低树级别的块,例如树级别1(层次级别1、深度1)的节点。这些块可以又分割成两个或更多下一个较低级别的块,例如树级别2(层次级别2、深度2)等,直到分割结束(因为满足结束标准,例如达到最大树深度或最小块大小)。未进一步分割的块也称为树的叶块或叶节点。分割成两个分割部分的树称为二叉树(binary-tree,BT),分割

成三个分割部分的树称为三叉树 (ternary-tree, TT), 分割成四个分割部分的树称为四叉树 (quad-tree, QT)。

[0165] 如上所述, 本文使用的术语“块”可以是图像的一部分, 特别是正方形或矩形部分。例如, 参照HEVC和VVC, 块可以为或可以对应于译码树单元 (coding tree unit, CTU)、译码单元 (coding unit, CU)、预测单元 (prediction unit, PU) 和变换单元 (transform unit, TU) 和/或对应的块, 例如译码树块 (coding tree block, CTB)、译码块 (coding block, CB)、变换块 (transform block, TB) 或预测块 (prediction block, PB)。

[0166] 例如, 译码树单元 (coding tree unit, CTU) 可以为或可以包括具有三个像素点阵列的图像中的亮度像素点的一个CTB、该图像中的色度像素点的两个对应CTB, 或者黑白图像中的或使用三个单独颜色平面和语法结构进行译码的图像中的像素点的一个CTB。这些语法结构用于对像素点进行译码。相应地, 译码树块 (coding tree block, CTB) 可以为 $N \times N$ 个像素点块, 其中, N 可以设为某个值, 使得一个分量划分为CTB, 这就是分割。译码单元 (coding unit, CU) 可以为或可以包括具有三个像素点阵列的图像中的亮度像素点的一个译码块、该图像中的色度像素点的两个对应译码块, 或者黑白图像中的或使用三个单独颜色平面和语法结构进行译码的图像中的像素点的一个译码块。这些语法结构用于对像素点进行译码。相应地, 译码块 (coding block, CB) 可以为 $M \times N$ 个像素点块, 其中, M 和 N 可以设为某个值, 使得一个CTB划分为译码块, 这就是分割。

[0167] 在实施例中, 例如根据HEVC, 可以使用表示为译码树的四叉树结构将译码树单元 (coding tree unit, CTU) 划分为多个CU。在CU级决定是否使用帧间 (时间) 预测或帧内 (空间) 预测对图像区域进行译码。每个CU可以根据PU划分类型进一步划分为一个、两个或四个PU。一个PU内执行相同的预测过程, 并以PU为单位向解码器发送相关信息。在根据PU划分类型执行预测过程得到残差块之后, 可以根据类似于用于CU的译码树的其它四叉树结构将CU分割成变换单元 (transform unit, TU)。

[0168] 在实施例中, 例如根据当前开发的最新视频译码标准 (称为通用视频译码 (Versatile Video Coding, VVC)), 使用四叉树结合二叉树 (quad-tree and binary-tree, QTBT) 分割来分割译码块。在QTBT块结构中, 一个CU可以为正方形或矩形。例如, 译码树单元 (coding tree unit, CTU) 首先通过四叉树结构进行分割。四叉树叶节点进一步通过二叉树或三叉 (ternary/triple) 树结构进行分割。分割树叶节点称为译码单元 (coding unit, CU), 这样的分段用于预测和变换处理, 无需任何进一步分割。这表示在QTBT译码块结构中, CU、PU和TU的块大小相同。与此同时, 还提出将三叉树分割等多重分割与QTBT块结构一起使用。

[0169] 在一个示例中, 视频编码器20中的模式选择单元260可以用于执行本文描述的分割技术的任意组合。

[0170] 如上所述, 视频编码器20用于从 (预定的) 预测模式集合中确定或选择最佳或最优的预测模式。预测模式集合可以包括帧内预测模式和/或帧间预测模式等。

[0171] 帧内预测

[0172] 帧内预测模式集合可以包括35种不同的帧内预测模式, 例如像DC (或均值) 模式和平面模式的非方向性模式或者如HEVC中定义的方向性模式, 或可以包括67种不同的帧内预测模式, 例如像DC (或均值) 模式和平面模式的非方向性模式或者如VVC中定义的方向性模

式。

[0173] 帧内预测单元254用于根据帧内预测模式集合中的帧内预测模式,使用同一个当前图像中的邻块的重建像素点来生成帧内预测块265。

[0174] 帧内预测单元254(或总称为模式选择单元260)还用于将帧内预测参数(或总称为表示块的选定帧内预测模式的信息)以语法元素266的形式输出到熵编码单元270,以包含到经编码的图像数据21中,使得(例如)视频解码器30可以接收并使用预测参数进行解码。

[0175] 帧间预测

[0176] (可能的)帧间预测模式集合取决于可用参考图像(即(例如)前述存储在DPB 230中的至少部分经过解码的图像)和其它帧间预测参数,例如取决于是否使用整个参考图像或只使用参考图像的一部分(例如当前块的区域周围的搜索窗口区域)来搜索最佳匹配参考块,和/或例如取决于是否进行像素插值(例如二分之一/半像素插值和/或四分之一像素插值)。

[0177] 除上述预测模式外,还可以使用跳过模式和/或直接模式。

[0178] 帧间预测单元244可以包括运动估计(motion estimation,ME)单元和运动补偿(motion compensation,MC)单元(两者在图2中未示出)。运动估计单元可以用于接收或获取图像块203(当前图像17中的当前图像块203)和解码图像231,或者至少一个或多个先前的重建块(例如一个或多个其它/不同先前的解码图像231中的重建块),以进行运动估计。例如,视频序列可以包括当前图像和先前解码图像231,或换句话说,当前图像和先前解码图像231可以为组成视频序列中的图像序列的一部分或组成该图像序列。

[0179] 例如,编码器20可以用于从多个其它图像中的同一个或不同图像的多个参考块中选择一个参考块,并将参考图像(或参考图像索引)和/或参考块的位置(x坐标、y坐标)与当前块的位置之间的偏移(空间偏移)作为帧间预测参数提供给运动估计单元。这种偏移也称为运动矢量(motion vector,MV)。

[0180] 运动补偿单元用于获取(例如接收)帧间预测参数,并根据或使用帧间预测参数执行帧间预测,得到帧间预测块265。由运动补偿单元执行的运动补偿可以包括根据通过运动估计确定的运动/块矢量来提取或生成预测块,还可以包括对子像素精度执行插值。插值滤波可以根据已知像素的像素点生成其它像素的像素点,从而潜在地增加可以用于对图像块进行译码的候选预测块的数量。一旦接收到当前图像块对应的PU的运动矢量时,运动补偿单元可以在其中一个参考图像列表中定位运动矢量指向的预测块。

[0181] 运动补偿单元还可以生成与块和视频条带(slice)相关的语法元素,以供视频解码器30在解码视频条带的图像块时使用。

[0182] 熵编码

[0183] 熵编码单元270用于将熵编码算法或方案(例如可变长度译码(variable length coding,VLC)方案、上下文自适应VLC(context adaptive VLC scheme,CAVLC)方案、算术译码方案、二值化,上下文自适应二进制算术译码(context adaptive binary arithmetic coding,CABAC)、基于语法的上下文自适应二进制算术译码(syntax-based context-adaptive binary arithmetic coding,SBAC)、概率区间分割熵(probability interval partitioning entropy,PIPE)译码或其它熵编码方法或技术)等应用于或不应用于(无压缩)量化系数209、帧间预测参数、帧内预测参数、环路滤波器参数和/或其它语法元素,得到

可以通过输出端272以经编码的码流21等形式输出的经编码的图像数据21,使得(例如)视频解码器30可以接收并使用这些参数进行解码。可以将经编码的码流21发送到视频解码器30,或者将其存储在存储器中稍后由视频解码器30发送或检索。

[0184] 视频编码器20的其它结构变型可以用于对视频流进行编码。例如,基于非变换的编码器20可以在某些块或帧没有变换处理单元206的情况下直接量化残差信号。在另一种实现方式中,编码器20可以包括组合成单个单元的量化单元208和反量化单元210。

[0185] 解码器和解码方法

[0186] 图3示出了用于实现本申请技术的视频解码器30的一个示例。视频解码器30用于接收(例如)由编码器20编码的经编码的图像数据21(例如经编码的码流21),得到解码图像331。经编码的图像数据或码流包括用于解码该经编码的图像数据的信息,例如表示经编码的视频条带中的图像块的数据和相关的语法元素。

[0187] 在图3的示例中,解码器30包括熵解码单元304、反量化单元310、逆变换处理单元312、重建单元314(例如求和器314)、环路滤波器320、解码图像缓冲器(decoded picture buffer,DPB)330、帧间预测单元344和帧内预测单元354。帧间预测单元344可以为或可以包括运动补偿单元。在一些示例中,视频解码器30可以执行大体上与参照图2中的视频编码器100描述的编码回合互逆的解码回合。

[0188] 如参照编码器20所述,反量化单元210、逆变换处理单元212、重建单元214、环路滤波器220、解码图像缓冲器(decoded picture buffer,DPB)230、帧间预测单元344和帧内预测单元354还组成视频编码器20的“内置解码器”。相应地,反量化单元310在功能上可以与反量化单元110相同,逆变换处理单元312在功能上可以与逆变换处理单元212相同,重建单元314在功能上可以与重建单元214相同,环路滤波器320在功能上可以与环路滤波器220相同,解码图像缓冲器330在功能上可以与解码图像缓冲器230相同。因此,视频编码器20的相应单元和功能的解释相应地适用于视频解码器30的相应单元和功能。

[0189] 熵解码

[0190] 熵解码单元304用于解析码流21(或一般为经编码的图像数据21)并对经编码的图像数据21执行熵解码等,得到量化系数309和/或解码译码参数(图3中未示出)等,例如帧间预测参数(例如参考图像索引和运动矢量)、帧内预测参数(例如帧内预测模式或索引)、变换参数、量化参数、环路滤波器参数和/或其它语法元素中的任一个或全部。熵解码单元304可以用于应用与参照编码器20中的熵编码单元270描述的编码方案对应的解码算法或方案。熵解码单元304还可以用于向模式选择单元360提供帧间预测参数、帧内预测参数和/或其它语法元素,以及向解码器30的其它单元提供其它参数。视频解码器30可以接收视频条带级和/或视频块级的语法元素。

[0191] 反量化

[0192] 反量化单元310可以用于从经编码的图像数据21(例如通过熵解码单元304等解析和/或解码)接收量化参数(quantization parameters,QP)(或一般为与反量化相关的信息)和量化系数,并根据这些量化参数对解码量化系数309进行反量化,得到解量化系数311。解量化系数311也可以称为变换系数311。反量化过程可以包括使用视频编码器20对视频条带中的每个视频块确定的量化参数来确定量化程度,同样也确定需要进行的反量化的程度。

[0193] 逆变换

[0194] 逆变换处理单元312可以用于接收解量化系数311(也称为变换系数311),并对解量化系数311进行变换,得到像素域中的重建残差块213。重建残差块213也可以称为变换块313。变换可以为逆变换,例如逆DCT、逆DST、逆整数变换或概念上类似的逆变换过程。逆变换处理单元312还可以用于从经编码的图像数据21(例如通过熵解码单元304等解析和/或解码)接收变换参数或相应的信息,以确定要对解量化系数311进行的变换。

[0195] 重建

[0196] 重建单元314(例如加法器或求和器314)可以用于通过如下方式将重建残差块313添加到预测块365以得到像素域中的重建块315:例如,将重建残差块313中的像素点值和预测块365中的像素点值相加。

[0197] 滤波

[0198] 环路滤波器单元320(在译码环路中或之后)用于对重建块315进行滤波,得到滤波块321,从而顺利进行像素转变或提高视频质量等。环路滤波器单元320可以包括一个或多个环路滤波器,例如去块效应滤波器、像素点自适应偏移(sample-adaptive offset,SAO)滤波器或者一个或多个其它滤波器,例如双边滤波器、自适应环路滤波器(adaptive loop filter,ALF)、锐化或平滑滤波器、协同滤波器,或其任意组合。虽然环路滤波器单元320在图3中示为环内滤波器,但是在其它配置中,环路滤波器单元320可以实现为环后滤波器。

[0199] 解码图像缓冲器

[0200] 随后将一个图像中的解码视频块321存储在解码图像缓冲器330中,解码图像缓冲器330存储作为参考图像的解码图像331,后续对其它图像进行运动补偿和/或输出或显示。

[0201] 解码器30用于通过输出端312等输出解码图像311,向用户显示或供用户查看。

[0202] 预测

[0203] 帧间预测单元344可以与帧间预测单元244(特别是与运动补偿单元)相同,帧内预测单元354在功能上可以与帧内预测单元254相同,并根据从经编码的图像数据21(例如通过熵解码单元304等解析和/或解码)接收的分割和/或预测参数或相应的信息来决定划分或分割和执行预测。模式选择单元360可以用于根据重建图像、块或相应的像素点(经过滤波或未经滤波)执行每个块的预测(帧内预测或帧间预测),得到预测块365。

[0204] 当将视频条带译码为经帧内译码(I)条带时,模式选择单元360中的帧内预测单元354用于根据指示(signal)的帧内预测模式和来自当前图像中的先前解码块的数据生成用于当前视频条带的图像块的预测块365。当视频图像译码为经帧间译码(即,B或P)条带时,模式选择单元360中的帧间预测单元344(例如运动补偿单元)用于根据运动矢量和从熵解码单元304接收的其它语法元素生成用于当前视频条带的视频块的预测块365。对于帧间预测,可以根据其中一个参考图像列表内的其中一个参考图像产生这些预测块。视频解码器30可以根据存储在DPB 330中的参考图像,使用默认构建技术来构建参考帧列表0和列表1。

[0205] 模式选择单元360用于通过解析运动矢量和其它语法元素,确定当前视频条带的视频块的预测信息,并使用预测信息产生用于正在解码的当前视频块的预测块。例如,模式选择单元360使用接收到的一些语法元素确定用于对视频条带的视频块进行译码的预测模式(例如帧内预测或帧间预测)、帧间预测条带类型(例如B条带、P条带或GPB条带)、用于条带的一个或多个参考图像列表的构建信息、用于条带的每个经帧间编码的视频块的运动矢

量、用于条带的每个经帧间译码的视频块的帧间预测状态,以及其它信息,以对当前视频条带中的视频块进行解码。

[0206] 视频解码器30的其它变型可以用于对经编码的图像数据21进行解码。例如,解码器30可以在没有环路滤波器单元320的情况下产生输出视频流。例如,基于非变换的解码器30可以在某些块或帧没有逆变换处理单元312的情况下直接对残差信号进行反量化。在另一种实现方式中,视频解码器30可以包括组合成单个单元的反量化单元310和逆变换处理单元312。

[0207] 图4为本发明实施例提供的视频译码设备400的示意图。视频译码设备400适用于实现本文描述的公开实施例。在一个实施例中,视频译码设备400可以是解码器(例如图1A中的视频解码器30)或编码器(例如图1A中的视频编码器20)。

[0208] 视频译码设备400包括:用于接收数据的入端口410(或输入端口410)和接收单元(Rx)420;用于处理所述数据的处理器、逻辑单元或中央处理器(central processing unit,CPU)430;用于发送所述数据的发送单元(Tx)440和出端口450(或输出端口450);用于存储所述数据的存储器460。视频译码设备400还可以包括与入端口410、接收单元420、发送单元440和出端口450耦合的光电(optical-to-electrical,OE)组件和电光(electrical-to-optical,E0)组件,用作光信号或电信号的出口或入口。

[0209] 处理器430通过硬件和软件实现。处理器430可以实现为一个或多个CPU芯片、一个或多个核(例如多核处理器)、一个或多个FPGA、一个或多个ASIC和一个或多个DSP。处理器430与入端口410、接收单元420、发送单元440、出端口450和存储器460通信。处理器430包括译码模块470。译码模块470实现上文描述的公开实施例。例如,译码模块470执行、处理、准备或提供各种译码操作。因此,将译码模块470包含在内为视频译码设备400的功能提供了实质性的改进,并且影响了视频译码设备400到不同状态的转换。或者,以存储在存储器460中并由处理器430执行的指令来实现译码模块470。

[0210] 存储器460可以包括一个或多个磁盘、磁带机和固态硬盘,并且可以用作溢出数据存储设备,以在选择程序来执行时存储这些程序以及存储在执行程序过程中读取的指令和数据。例如,存储器460可以是易失性和/或非易失性的,并且可以是只读存储器(read-only memory,ROM)、随机存取存储器(random access memory,RAM)、三态内容寻址存储器(ternary content-addressable memory,TCAM)和/或静态随机存取存储器(static random-access memory,SRAM)。

[0211] 图5为示例性实施例提供的装置500的简化框图。装置500可以用作图1的源设备12和目标设备14中的任一个或两个。装置500可以实现上文描述的本申请技术。装置500可以是包括多个计算设备的计算系统,也可以是单个计算设备,例如手机、平板电脑、膝上型电脑、笔记本电脑、台式电脑等。

[0212] 装置500中的处理器502可以是中央处理器。或者,处理器502可以是现有的或今后将研发出的能够操控或处理信息的任何其它类型的设备或多个设备。虽然可以使用如图所示的处理器502等单个处理器来实施所公开的实现方式,但使用多个处理器可以提高速度和效率。

[0213] 在一种实现方式中,装置500中的存储器504可以是只读存储器(read only memory,ROM)设备或随机存取存储器(random access memory,RAM)设备。任何其它合适类

型的存储设备都可以用作存储器504。存储器504可以包括处理器502通过总线512访问的代码和数据506。存储器504还可以包括操作系统508和应用程序510,应用程序510包括至少一个程序,这个程序使得处理器502执行本文所述方法。例如,应用程序510可以包括应用1至应用N,还包括执行本文所述方法的视频译码应用。装置500还可以包括辅助存储器514形式的其它存储器,辅助存储器514可以是与移动计算设备一起使用的存储卡等。由于视频通信会话可以包括大量信息,因此它们可以全部或部分地存储在辅助存储器514中,并需要根据加载到存储器504中进行处理。

[0214] 装置500还可以包括一个或多个输出设备,例如显示器518。在一个示例中,显示器518可以是将显示器与触敏元件组合的触敏显示器,该触敏元件能够用于感测触摸输入。显示器518可以通过总线512与处理器502耦合。除显示器518之外,还可以提供使得用户对装置500进行编程或以其它方式使用装置500的其它输出设备。当输出设备是显示器或者包括显示器时,该显示器可以为液晶显示器(liquid crystal display,LCD)、阴极射线管(cathode-ray tube,CRT)显示器、等离子显示器或发光二极管(light emitting diode,LED)显示器(例如有机LED(organic LED,OLED)显示器)。

[0215] 装置500还可以包括图像传感设备520或者与图像传感设备520通信。图像传感设备520可以是相机,或现有的或今后将研发出的能够感测图像(例如操作装置500的用户的图像)的任何其它图像传感设备520等。可以将图像传感设备520朝向操作装置500的用户放置。在一个示例中,可以设置图像传感设备520的位置和光轴,使得视野包括与显示器518紧邻的区域,从这个区域可以看到显示器518。

[0216] 装置500还可以包括声音传感设备522或者与声音传感设备通信。声音传感设备522可以是麦克风,或现有的或今后将研发出的能够感测装置500附近声音的任何其它声音传感设备等。可以将声音传感设备522朝向操作装置500的用户放置。声音传感设备522可以用于接收用户在操作装置500时发出的语音或其它话语等声音。

[0217] 虽然图5描述了装置500中的处理器502和存储器504集成在单个设备中,但是可以使用其它构造。处理器502的操作可以分布在多台机器(每台机器包括一个或多个处理器)上,这些机器可以直接耦合或者通过局域网或其它网络耦合。存储器504可以分布在多台机器中,例如基于网络的存储器或者执行装置500的操作的多台机器中的存储器。虽然装置500的总线512在这里示为单个总线,但是总线512可以有多个。此外,辅助存储器514可以直接与装置500中的其它组件耦合或可以通过网络被访问,并且可以包括单个集成单元(例如一个存储卡)或多个单元(例如多个存储卡)。因此,装置500可以具有各种各样的构造。

[0218] H.264/AVC和HEVC等视频译码方案是根据基于块的混合视频译码的成功原理进行设计的。利用这个原理,首先将图像分割成块,然后通过帧内或帧间预测对每个块进行预测。

[0219] 本文使用的术语“块”可以是图像或帧的一部分。为便于描述,本文参考由ITU-T视频译码专家组(Video Coding Experts Group,VCEG)和ISO/IEC运动图像专家组(Motion Picture Experts Group,MPEG)的视频译码联合协作团队(Joint Collaboration Team on Video Coding,JCT-VC)开发的高效视频译码(High-Efficiency Video Coding,HEVC)或通用视频译码(Versatile Video Coding,VVC)参考软件来描述本发明实施例本领域普通技术人员理解本发明实施例不限于HEVC或VVC。块可以指CU、PU和TU。在HEVC中,可以使用表示

为译码树的四叉树结构将CTU划分为多个CU。在CU级决定是否使用帧间(时间)预测或帧内(空间)预测对图像区域进行译码。每个CU可以根据PU划分类型进一步划分为一个、两个或四个PU。一个PU内执行相同的预测过程,并以PU为单位向解码器发送相关信息。在根据PU划分类型执行预测过程得到残差块之后,可以根据类似于用于CU的译码树的其它四叉树结构将CU分割成变换单元(transform unit, TU)。在视频压缩技术的最新开发中,使用四叉树结合二叉树(quad-tree and binary-tree, QTBT)分割来分割译码块。在QTBT块结构中,一个CU可以为正方形或矩形。例如,译码树单元(coding tree unit, CTU)首先通过四叉树结构进行分割。四叉树叶节点进一步通过二叉树结构进行分割。二叉树叶节点称为译码单元(coding unit, CU),这样的分段用于预测和变换处理,无需任何进一步分割。这表示在QTBT译码块结构中, CU、PU和TU的块大小相同。与此同时,还提出将三叉树分割等多重分割与QTBT块结构一起使用。

[0220] ITU-T VCEG (Q6/16) 和ISO/IEC MPEG (JTC 1/SC 29/WG 11) 正在研究未来视频译码技术标准化的潜在需求,其中,未来视频译码技术的压缩能力将大大超过当前HEVC标准的压缩能力(包括针对屏幕内容译码和高动态范围译码的当前扩展版本和近期扩展版本)。这两个专家组正在联合开展这一探索活动,称为联合视频探索小组(Joint Video Exploration Team, JVET),以评估其专家在这一领域提出的压缩技术设计。

[0221] 方向性帧内预测可以采用帧内预测模式,表示从上对角线到下对角线的不同预测角度。为了定义预测角度,定义了32像素点网格上的偏移值 p_{ang} 。图6示出了垂直预测模式下的 p_{ang} 与对应帧内预测模式之间的关联关系。水平预测模式下的方案是翻转到垂直方向,然后相应地赋值 p_{ang} 。如上所述,所有角度预测模式可用于所有适当的帧内预测块大小。所有角度预测模式都使用相同的32像素点网格来定义预测角度。图6所示的 p_{ang} 值在32像素点网格上的分布表明,垂直方向周围的预测角度的分辨率增加,而对角线方向的预测角度的分辨率较低。水平方向也是如此,即水平方向周围的预测角度的分辨率增加,而对角线方向的预测角度的分辨率较低。这种设计是因为观察到在很多视频内容中,与对角线结构相比,近似水平和垂直的结构起着重要作用。

[0222] 虽然在水平和垂直预测方向上选择预测要使用的像素点很简单,但如果是角度预测,这项任务需要更多的操作。对于模式11至25,当根据角度方向上的预测像素点集 p_{ref} (也称为主参考边)对当前块 B_c 进行预测时, p_{ref} 的垂直部分和水平部分中的像素点都会涉及。由于确定各个像素点在 p_{ref} 的任一分支上的位置需要一定的计算操作,因此针对HEVC帧内预测设计了统一的1维预测参考。该方案如图7所示。在执行实际预测操作之前,将参考像素点集 p_{ref} 映射到1维向量 $p_{1,ref}$ 。用于这种映射的投影根据相应帧内预测模式的帧内预测角度指示的方向而定。只将预测要使用的那部分 p_{ref} 中的参考像素点映射到 $p_{1,ref}$ 。图8和图9分别示出了水平和垂直角度预测方向上的每个角度预测模式下的参考像素点到 $p_{1,ref}$ 的实际映射。为预测像素点块构建一次参考像素点集 $p_{1,ref}$ 。然后根据该集合中的两个相邻参考像素点推导预测,如下详述。从图8和图9可见,在所有帧内预测模式中,1维参考像素点集都没有完全填满。集合中只包括在对应帧内预测方向的投影范围内的位置。

[0223] 水平和垂直预测模式下的预测以相同的方式执行,只是互换块的x坐标和y坐标。根据 $p_{1,ref}$ 执行预测的精度为1/32像素精度。根据角度参数 p_{ang} 的值,确定 $p_{1,ref}$ 的像素点偏移 i_{idx} 和位置(x,y)上的像素点的加权因子 i_{fact} 。这里提供了垂直模式下的推导方式。水平

模式下的推导方式以此类推,即互换x和y。

$$[0224] \quad i_{\text{idx}} = (y + 1) \cdot \frac{P_{\text{ang}}}{32}, \quad i_{\text{fact}} = [(y + 1) \cdot p_{\text{ang}}] \bmod 32。$$

[0225] 如果 i_{fact} 不等于0,即进行预测的位置正好不是 $p_{1,\text{ref}}$ 中的完整像素点位置,则 $p_{1,\text{ref}}$ 中的两个相邻像素点位置之间的线性加权执行如下:

$$[0226] \quad B_c(x, y) = \frac{32 - i_{\text{fact}}}{32} \cdot p_{1,\text{ref}}(x + i_{\text{idx}} + 1) + \frac{i_{\text{fact}}}{32} \cdot p_{1,\text{ref}}(x + i_{\text{idx}} + 2),$$

[0227] 其中, $0 \leq x, y < N_c$ 。需要说明的是, i_{idx} 和 i_{fact} 的值只取决于y,因此只需要每行计算一次(在垂直预测模式下)。

[0228] 通用测试模型(Versatile Test Model,VTM)-1.0使用35种帧内模式,而基准集(Benchmark Set,BMS)使用67种帧内模式。许多视频译码框架中都使用帧内预测,以便能够在只能涉及某个帧的情况下提高压缩效率。

[0229] 图10A示出了VVC中提出的67种帧内预测模式的一个示例。67种帧内预测模式中的多种帧内预测模式包括:平面模式(索引0)、DC模式(索引1)和角度模式(索引2到66),其中,图10A中的左下方角度模式指的是索引2,而且这些索引的编号递增到索引66,即表示图10A中的最右上方角度模式。

[0230] 如图10B至图10C所示,JEM的最新版本有一些模式与倾斜(skew)帧内预测方向对应。在这些模式中的任一模式下,为了预测一个块内的像素点,如果块边上的对应位置为分数位置,则需要对相邻参考像素点集执行插值。HEVC和VVC在两个相邻参考像素点之间进行线性插值。JEM使用更复杂的4抽头插值滤波器。根据宽度值或高度值,选择滤波器系数为高斯(Gaussian)滤波器系数或立方(Cubic)滤波器系数。确定是否使用宽度或高度与选择主参考边相协调:当帧内预测模式大于或等于对角线模式时,选择参考像素点的顶边作为主参考边,并选择宽度值来确定所使用的插值滤波器。否则,选择块的左侧边作为主参考边,而且通过高度来控制滤波器选择过程。具体地,如果选定的边长度小于或等于8个像素点,则使用三次插值4抽头滤波器。否则,插值滤波器为4抽头高斯滤波器。

[0231] JEM中使用的具体滤波器系数如表1所示。根据子像素偏移和滤波器类型,通过与从表1中选择的系数进行卷积来计算预测像素点,如下所示:

$$[0232] \quad s(x) = \left(\sum_{i=0}^{i<4} (\text{ref}_{i+x} \cdot c_i) + 128 \right) \gg 8$$

[0233] 在该等式中,“ \gg ”表示按位右移运算。

[0234] 如果选择的是立方滤波器,则将预测像素点进一步修正(clip)到允许的取值范围内。该允许取值范围在序列参数集(sequence parameter set,SPS)中定义或根据选定分量的位深度进行推导。

[0235] 表1:JEM中使用的帧内预测插值滤波器

[0236]

子像素偏移	立方滤波器				高斯滤波器			
	c ₀	c ₁	c ₂	c ₃	c ₀	c ₁	c ₂	c ₃
0 (整数)	0	256	0	0	47	161	47	1
1	-3	252	8	-1	43	161	51	1
2	-5	247	17	-3	40	160	54	2
3	-7	242	25	-4	37	159	58	2
4	-9	236	34	-5	34	158	62	2
5	-10	230	43	-7	31	156	67	2
6	-12	224	52	-8	28	154	71	3
7	-13	217	61	-9	26	151	76	3
8	-14	210	70	-10	23	149	80	4
9	-15	203	79	-11	21	146	85	4
10	-16	195	89	-12	19	142	90	5
11	-16	187	98	-13	17	139	94	6
12	-16	179	107	-14	16	135	99	6
13	-16	170	116	-14	14	131	104	7
14	-17	162	126	-15	13	127	108	8
15	-16	153	135	-16	11	123	113	9
16 (半像素)	-16	144	144	-16	10	118	118	10
17	-16	135	153	-16	9	113	123	11
18	-15	126	162	-17	8	108	127	13
19	-14	116	170	-16	7	104	131	14
20	-14	107	179	-16	6	99	135	16
21	-13	98	187	-16	6	94	139	17
22	-12	89	195	-16	5	90	142	19
23	-11	79	203	-15	4	85	146	21
24	-10	70	210	-14	4	80	149	23
25	-9	61	217	-13	3	76	151	26
26	-8	52	224	-12	3	71	154	28

[0237]

子像素偏移	立方滤波器				高斯滤波器			
	c ₀	c ₁	c ₂	c ₃	c ₀	c ₁	c ₂	c ₃
27	-7	43	230	-10	2	67	156	31
28	-5	34	236	-9	2	62	158	34
29	-4	25	242	-7	2	58	159	37
30	-3	17	247	-5	2	54	160	40
31	-1	8	252	-3	1	51	161	43

[0238] 表2示出了具有6位精度的另一插值滤波器组。

[0239] 表2:具有6位精度的插值滤波器组

[0240]

子像素偏移	统一的帧内/帧间滤波器				高斯滤波器			
	c ₀	c ₁	c ₂	c ₃	c ₀	c ₁	c ₂	c ₃
0 (整数)	0	64	0	0	16	32	16	0
1	-1	63	2	0	15	29	17	3
2	-2	62	4	0	14	29	18	3
3	-2	60	7	-1	14	29	18	3
4	-2	58	10	-2	14	28	18	4
5	-3	57	12	-2	13	28	19	4
6	-4	56	14	-2	12	28	20	4
7	-4	55	15	-2	12	27	20	5
8	-4	54	16	-2	11	27	21	5
9	-5	53	18	-2	11	27	21	5
10	-6	52	20	-2	10	26	22	6
11	-6	49	24	-3	10	26	22	6
12	-6	46	28	-4	9	26	23	6
13	-5	44	29	-4	9	26	23	6
14	-4	42	30	-4	8	25	24	7
15	-4	39	33	-4	8	25	24	7
16 (半像素)	-4	36	36	-4	7	25	25	7
17	-4	33	39	-4	7	24	25	8
18	-4	30	42	-4	7	24	25	8
19	-4	29	44	-5	6	23	26	9

[0241]

子像素偏移	统一的帧内/帧间滤波器				高斯滤波器			
	c ₀	c ₁	c ₂	c ₃	c ₀	c ₁	c ₂	c ₃
20	-4	28	46	-6	6	23	26	9
21	-3	24	49	-6	6	22	26	10
22	-2	20	52	-6	6	22	26	10
23	-2	18	53	-5	5	21	27	11
24	-2	16	54	-4	5	21	27	11
25	-2	15	55	-4	5	20	27	12
26	-2	14	56	-4	4	20	28	12
27	-2	12	57	-3	4	19	28	13
28	-2	10	58	-2	4	18	28	14
29	-1	7	60	-2	3	18	29	14
30	0	4	62	-2	3	18	29	14
31	0	2	63	-1	3	17	29	15

[0242] 根据子像素偏移和滤波器类型,通过与从表1中选择的系数进行卷积来计算帧内预测像素点,如下所示:

$$[0243] \quad s(x) = \left(\sum_{i=0}^{i<4} (\text{ref}_{i+x} \cdot c_i) + 32 \right) \gg 6$$

[0244] 在该等式中,“>>”表示按位右移运算。

[0245] 表3示出了具有6位精度的另一插值滤波器组。

[0246] 表3:具有6位精度的插值滤波器组

[0247]

子像素偏移	色度 DCT-IF 滤波器				高斯滤波器			
	c ₀	c ₁	c ₂	c ₃	c ₀	c ₁	c ₂	c ₃
0 (整数)	0	64	0	0	16	32	16	0
1	-1	63	2	0	15	29	17	3
2	-2	62	4	0	15	29	17	3
3	-2	60	7	-1	14	29	18	3
4	-2	58	10	-2	13	29	18	4
5	-3	57	12	-2	13	28	19	4
6	-4	56	14	-2	13	28	19	4
7	-4	55	15	-2	12	28	20	4

子像素偏移	色度 DCT-IF 滤波器				高斯滤波器			
	c ₀	c ₁	c ₂	c ₃	c ₀	c ₁	c ₂	c ₃
8	-4	54	16	-2	11	28	20	5
9	-5	53	18	-2	11	27	21	5
10	-6	52	20	-2	10	27	22	5
11	-6	49	24	-3	9	27	22	6
12	-6	46	28	-4	9	26	23	6
13	-5	44	29	-4	9	26	23	6
14	-4	42	30	-4	8	25	24	7
15	-4	39	33	-4	8	25	24	7
16 (半像素)	-4	36	36	-4	8	24	24	8
17	-4	33	39	-4	7	24	25	8
18	-4	30	42	-4	7	24	25	8
19	-4	29	44	-5	6	23	26	9
20	-4	28	46	-6	6	23	26	9
21	-3	24	49	-6	6	22	27	9
22	-2	20	52	-6	5	22	27	10
23	-2	18	53	-5	5	21	27	11
24	-2	16	54	-4	5	20	28	11
25	-2	15	55	-4	4	20	28	12
26	-2	14	56	-4	4	19	28	13
27	-2	12	57	-3	4	19	28	13
28	-2	10	58	-2	4	18	29	13
29	-1	7	60	-2	3	18	29	14
30	0	4	62	-2	3	17	29	15
31	0	2	63	-1	3	17	29	15

[0248]

[0249] 图11示出了HEVC UIP方案中使用的多种帧内预测模式的示意图。对于亮度块，帧内预测模式可以包括最多36种帧内预测模式，这些模式可以包括3种非方向性模式和33种方向性模式。非方向性模式可以包括平面预测模式、均值(DC)预测模式和基于亮度的色度(LM)预测模式。平面预测模式可以通过假设块幅度表面具有根据块的边界推导的水平和垂直斜率来执行预测。DC预测模式可以通过假设平坦块表面具有与块边界的平均值一致的值来执行预测。LM预测模式可以通过假设块的色度值与块的亮度值一致来执行预测。方向性模式可以根据相邻块执行预测，如图11所示。

[0250] H.264/AVC和HEVC中规定，在帧内预测过程中使用参考像素点之前，可以先对这些参考像素点使用低通滤波器。是否使用参考像素点滤波器由帧内预测模式和块大小决定。这种机制可以称为模式相关帧内平滑 (Mode Dependent Intra Smoothing, MDIS)。还存在

与MDIS相关的多种方法。例如,自适应参考像素点平滑(Adaptive Reference Sample Smoothing, ARSS)方法可以显式地(即,码流中包括一个标志)或隐式地(即,例如通过数据隐藏来避免将标志放入到码流中,以减少指示开销)指示是否对预测像素点进行滤波。在这种情况下,编码器可以通过测试所有潜在帧内预测模式的速率失真(Rate-Distortion, RD)代价确定是否进行平滑处理。

[0251] VVC中使用基于四叉树和二叉树的分割机制,称为QTBT。如图12所示,QTBT分割不仅可以得到正方形块,还可以得到矩形块。当然,与HEVC/H.265标准中使用的基于四叉树的传统分割相比,QTBT分割会在编码器侧产生一些指示开销,并且会增加编码器侧的计算复杂度。虽然如此,与传统的四叉树相比,基于QTBT的分割具有更好的分段特性,因此译码效率明显更高。

[0252] 进行分割所使用的树的叶节点按Z形扫描顺序进行处理,使得与当前叶节点对应的当前块具有在编码或解码过程中已经重建(除非当前块位于条带的边界处)的左邻块和上邻块。图12也示出了这一点。图12中的右侧部分所示的树的叶节点从左到右扫描对应于图12中的左侧部分所示的块的空间Z形扫描顺序。四叉树或多类型树采用相同的扫描方式。

[0253] 对于方向性帧内预测,根据先前的重建相邻块的像素点获取参考像素点。根据块大小和帧内预测模式,在使用参考像素点获取预测像素点的值之前,可以先对这些参考像素点使用滤波器。

[0254] 在边界平滑和PDPC的情况下,将预测块的前几列或前几行与根据相邻像素点生成的其它预测信号相组合。

[0255] 当使用水平帧内预测模式(例如,在图10C中示为模式18)来执行帧内预测时,按列复制预测块左边上的参考像素点,从而一行中的所有像素点被设置为预测块左边的这一行中的参考像素点。

[0256] 当使用垂直帧内预测模式(例如,在图10C中示为模式50)来执行帧内预测时,按行复制预测块顶边上的参考像素点,从而一列中的所有像素点被设置为预测块顶边的这一列中的参考像素点。

[0257] 2011年7月,A.Minzawa、K.Sugimoto和S.Sekiguchi在意大利都灵的第6次JCT-VC会议的文献JCTVC-F172中发表了“An improved intra vertical and horizontal prediction(改进的帧内垂直和水平预测)”,其中的用于垂直和水平帧内预测模式的推导过程详述如下:

[0258] 垂直预测:

$$[0259] \quad S'(x,y) = S(x,-1) + (S(-1,y) - S(-1,-1)) / 2^{x+1}$$

[0260] 水平预测:

$$[0261] \quad S'(x,y) = S(-1,y) + (S(x,-1) - S(-1,-1)) / 2^{y+1}$$

[0262] 其中, (x,y) 表示预测像素点在亮度预测块中的位置,如图13A所示, $S'(x,y)$ 表示预测像素点, $S(x,y)$ 表示传统预测要使用的参考像素点。

[0263] 由于 $S'(x,y)$ 的值可能会超出预测像素点的最小值 p_{MIN} 和最大值 P_{MAX} 的范围,因此需要使用修正函数,(例如)如下所示:

$$[0264] \quad S''(x,y) = \text{Clip3}(p_{\text{MIN}}, P_{\text{MAX}}, S'(x,y)),$$

[0265] 其中,Clip3()表示修正函数。进一步给出了该函数的示例性定义。

[0266] 对于预测像素点 $S(x,y)$,位于所述预测像素点上方的最近参考像素点可以表示为 $S(x,-1)$ 。该最近参考像素点与所述预测像素点位于同一列。

[0267] 类似地,位于所述预测像素点左侧的最近参考像素点可以表示为 $S(-1,y)$ 。该最近参考像素点与所述预测像素点位于同一行。

[0268] 根据帧内预测模式不同,可以按不同的方式具体实现简化版PDPC:

[0269] 对于平面、DC、HOR/VER帧内预测模式(在图10B和图10C中分别表示为0、1、18和50),执行以下步骤:

[0270] 位于 (x,y) 上的预测像素点 $P(x,y)$ 计算如下:

$$[0271] \quad P(x,y) = \text{Clip1Cmp}((wL \times R_{-1,y} + wT \times R_{x,-1} - wTL \times R_{-1,-1} + (64 - wL - wT \mp wTL) \times P(x,y) + 32) \gg 6) \quad (1)$$

[0272] 其中, $R_{x,-1}$ 和 $R_{-1,y}$ 表示位于当前像素点 (x,y) 上方和左侧的参考像素点, $R_{-1,-1}$ 表示左上方像素点,即位于当前块左上角的参考像素点,函数clip1Cmp设置如下:

[0273] -如果cIdx等于0,则将clip1Cmp设置为Clip1_Y;

[0274] -否则,将clip1Cmp设置为Clip1_C。

[0275] $\text{Clip1}_Y(x) = \text{Clip3}(0, (1 \ll \text{BitDepth}_Y) - 1, x)$

[0276] $\text{Clip1}_C(x) = \text{Clip3}(0, (1 \ll \text{BitDepth}_C) - 1, x)$

$$[0277] \quad \text{Clip3}(x,y,z) = \begin{cases} x & ; \quad z < x \\ y & ; \quad z > y \\ z & ; \quad \text{otherwise} \end{cases} .$$

[0278] BitDepth_Y 表示亮度像素点的位深度。

[0279] BitDepth_C 表示色度像素点的位深度。

[0280] BitDepth_Y 和 BitDepth_C 可以在码流的序列参数集(sequence parameter set,SPS)中指示。

[0281] $\text{Clip1}_Y(x)$ 和 $\text{Clip1}_C(x)$ 还可以有其它定义。具体地,如F.Galpin、P.Bordes和F.Le Lannec在文献JVET-C0040“Adaptive Clipping in JEM2.0(JEM2.0中的自适应修正)”中所述, $\text{Clip1Cmp}(x) = \text{Clip3}(\min_c, \max_c, x)$,

[0282] 其中, \min_c 表示当前条带中针对分量ID C使用的修正下限,

[0283] \max_c 表示当前条带中针对分量ID C使用的修正上限,

[0284] C表示颜色分量(例如,Y表示亮度,Cb和Cr表示色度),

[0285] “ $x \gg y$ ”表示x的补码整数表示向右算术移动y个二进制位。只有y为非负整数值时才有这个函数定义。右移的结果是移进最高有效位(most significant bit,MSB)的比特位等于移位操作之前的x的MSB。

[0286] DC模式权重计算如下:

[0287] $wT = 32 \gg ((y \ll 1) \gg \text{shift})$, $wL = 32 \gg ((x \ll 1) \gg \text{shift})$,

[0288] $wTL = -(wL \gg 4) - (wT \gg 4)$,

[0289] 其中, $\text{shift} = (\log_2(\text{width}) + \log_2(\text{height}) + 2) \gg 2$ 。

[0290] 对于平面模式, $wTL = 0$,而对于水平模式, $wTL = wT$,对于垂直模式, $wTL = wL$ 。图13B示出了一个 4×4 块内的位置(0,0)和(1,0)的DC模式PDPC权重(wL, wT, wTL)。从图13可见,必须执行等式(1)中的修正运算,但最新PDPC实现方案存在一个问题。以下示例说明了修正运

算的结果可能超出由 BitDepth_y 或 BitDepth_c 确定的范围的情况。

[0291] 假设 $R_{-1,y}=0, R_{x,-1}=0, R_{-1,-1}=100, P(x,y)=0$, 根据等式(1)可以得出, 对于 4×4 预测块内的位置 $(0, 0)$, $P(x,y) = \text{Clip1Cmp}((wL \times R_{-1,y} + wT \times R_{x,-1} \pm wTL \times R_{-1,-1} + (64 - wL - wT \mp wTL) \times P(x,y) + 32) \gg 6) = \text{Clip1Cmp}(wTL \times R_{-1,-1} + 32) \gg 6) = \text{Clip1Cmp}(-4 \times 100 + 32) \gg 6)$,

[0292] 其中, $wTL=4$, 如图13B所示。

[0293] 从上面的示例可见, 使用算术位移将负值“ $-4 \times 100 + 32 = -368$ ”向右移动。根据实现方式不同, 将负值向右算术移动可能会得到不同的输出(例如, 在C/C++编程语言的情况下), 因此不能保证 $\text{Clip1Cmp}()$ 的输出始终为0, 因为将负值向右移动得到的结果在具体实现方式中可以具有负号和非零大小。

[0294] 对于对角线模式(在图10B和图10C中表示为2和66)和相邻模式(图10B或图10C中不小于58或不大于10的方向性模式), 根据相同的公式(1)执行处理如下。

[0295] 图14A示出了PDPC向右上方对角线模式扩展的参考像素点 $R_{x,-1}, R_{-1,y}$ 和 $R_{-1,-1}$ 的定义。预测像素点 $\text{pred}(x', y')$ 位于预测块内的 (x', y') 上。参考像素点 $R_{x,-1}$ 的坐标 x 由以下等式给出: $x = x' + y' + 1$, 类似地, 参考像素点 $R_{-1,y}$ 的坐标 y 由以下等式给出: $y = x' + y' + 1$ 。

[0296] 右上方对角线模式的PDPC权重为:

[0297] $wT = 16 \gg ((y' \ll 1) \gg \text{shift}), wL = 16 \gg ((x' \ll 1) \gg \text{shift}), wTL = 0$ 。

[0298] 类似地, 图14B示出了PDPC向左下方对角线模式扩展的参考像素点 $R_{x,-1}, R_{-1,y}$ 和 $R_{-1,-1}$ 的定义。参考像素点 $R_{x,-1}$ 的坐标 x 由以下等式给出: $x = x' + y' + 1$, 参考像素点 $R_{-1,y}$ 的坐标 y 为: $y = x' + y' + 1$ 。右上方对角线模式的PDPC权重为: $wT = 16 \gg ((y' \ll 1) \gg \text{shift}), wL = 16 \gg ((x' \ll 1) \gg \text{shift}), wTL = 0$ 。图14C示出了右上方相邻对角线模式的情况。右上方相邻对角线模式的PDPC权重为: $wT = 32 \gg ((y' \ll 1) \gg \text{shift}), wL = 0, wTL = 0$ 。类似地, 图14D示出了左下方相邻对角线模式的情况。左下方相邻对角线模式的PDPC权重为: $wL = 32 \gg ((x' \ll 1) \gg \text{shift}), wT = 0, wTL = 0$ 。后两种情况下的参考像素点坐标使用已经用于角度模式帧内预测的表格进行计算。如果计算出分数参考像素点坐标, 则对参考像素点进行线性插值。

[0299] 简化版PDPC可以按照VVC规范执行。此外, 采用以下表示:

[0300] $\text{invAngle} = \text{Round}\left(\frac{256 \times 32}{\text{intraPredAngle}}\right)$ 表示反角度的值,

[0301] $\text{Round}(x) = \text{Sign}(x) \times \text{Floor}(\text{Abs}(x) + 0.5)$,

[0302] $\text{Sign}(x) = \begin{cases} 1 & ; x > 0 \\ 0 & ; x == 0, \\ -1 & ; x < 0 \end{cases}$

[0303] $\text{Floor}(x)$ 表示小于或等于 x 的最大整数,

[0304] $\text{Log}_2(x)$ 表示 x 的以2为底的对数,

[0305] intraPredAngle 表示在表6中详述的角度参数,

[0306] $A = C ? B : D$ 为三元赋值运算, 其中, 如果条件 C 为真(true), 则将 A 设置为 B ; 否则, 如果条件 C 为假(false), 则将 A 设置为 D ,

[0307] INTRA_PLANAR 表示平面帧内预测模式,

[0308] INTRA_DC 表示DC帧内预测模式,

[0309] INTRA_ANGULARXX表示其中一种方向性帧内预测模式,其中,XX表示该模式的编号和对应方向,如图4B所示。

[0310] 如果某个术语在此未进行说明,则应理解,可以在VVC规范或HEVC/H.265标准规范中找到该术语的定义。

[0311] 鉴于上述表示,简化版PDPC的步骤可以定义如下:

[0312] 该过程的输入包括:

[0313] -帧内预测模式predModeIntra;

[0314] -变量nTbW,表示变换块宽度;

[0315] -变量nTbH,表示变换块高度;

[0316] -变量refW,表示参考像素点宽度;

[0317] -变量refH,表示参考像素点高度;

[0318] -预测像素点predSamples[x][y],其中, $x=0\cdots nTbW-1$, $y=0\cdots nTbH-1$;

[0319] -相邻像素点p[x][y],其中, $x=-1$, $y=-1\cdots refH-1$; $x=0\cdots refW-1$, $y=-1$;

[0320] -变量cIdx,表示当前块的颜色分量。

[0321] 该过程的输出为修改后的预测像素点predSamples[x][y],其中, $x=0\cdots nTbW-1$, $y=0\cdots nTbH-1$ 。

[0322] 根据cIdx的值,将函数clip1Cmp设置如下:

[0323] -如果cIdx等于0,则将clip1Cmp设置为Clip1_y;

[0324] -否则,将clip1Cmp设置为Clip1_c。

[0325] 将变量nScale设置为 $((\text{Log}_2(nTbW) + \text{Log}_2(nTbH) - 2) \gg 2)$ 。

[0326] 预测像素点阵列mainRef[x]和sideRef[y](其中, $x=0\cdots nTbW-1$, $y=0\cdots nTbH-1$)的推导方式如下:

[0327] $refT[x][y] = p[x][-1]$

[0328] $sideRef[y] = p[-1][y]$

[0329] 变量refL[x][y]、refT[x][y]、wT[y]、wL[x]和wTL[x][y](其中, $x=0\cdots nTbW-1$, $y=0\cdots nTbH-1$)的推导方式如下:

[0330] -如果predModeIntra等于INTRA_PLANAR、INTRA_DC、INTRA_ANGULAR18或

[0331] INTRA_ANGULAR50,则

[0332] $refL[x][y] = p[-1][y]$

[0333] $refT[x][y] = p[x][-1]$

[0334] $wT[y] = 32 \gg ((y \ll 1) \gg nScale)$

[0335] $wL[x] = 32 \gg ((x \ll 1) \gg nScale)$

[0336] $wTL[x][y] = (\text{predModeIntra} == \text{INTRA_DC}) ? ((wL[x] \gg 4) + (wT[y] \gg 4)) : 0$

[0337] {x,y}限定预测像素点的位置,wT、wL和wTL为根据所述限定的位置与参考像素点相关的权重,nScale为控制传播距离(权重wT和wL的衰减)的缩放参数。

[0338] -否则,如果predModeIntra等于INTRA_ANGULAR2或INTRA_ANGULAR66,则

[0339] $refL[x][y] = p[-1][x+y+1]$

[0340] $refT[x][y] = p[x+y+1][-1]$

[0341] $wT[y] = (32 \gg 1) \gg ((y \ll 1) \gg nScale)$

[0342] $wL[x] = (32 \gg 1) \gg ((x \ll 1) \gg nScale)$

[0343] $wTL[x][y] = 0$

[0344] - 否则, 如果predModeIntra小于或等于INTRA_ANGULAR10, 则按顺序执行以下步骤:

[0345] 1. 变量dXPos[y]、dXFrac[y]、dXInt[y]和dX[x][y]通过invAngle推导如下:

[0346] $dXPos[y] = ((y+1) \times invAngle+2) \gg 2$

[0347] $dXFrac[y] = dXPos[y] \& 63$

[0348] $dXInt[y] = dXPos[y] \gg 6$

[0349] $dX[x][y] = x + dXInt[y]$

[0350] 2. 变量refL[x][y]、refT[x][y]、wT[y]、wL[x]和wTL[x][y]的推导方式如下:

[0351] $refL[x][y] = 0$

[0352] $refT[x][y] = (dX[x][y] \ll refW-1) ? ((64-dXFrac[y]) \times mainRef[dX[x][y]] + dXFrac[y] \times mainRef[dX[x][y]+1] + 32) \gg 60$ (等式1)

[0353] $wT[y] = (dX[x][y] \ll refW-1) ? 32 \gg ((y \ll 1) \gg nScale) : 0$

[0354] $wL[x] = 0$

[0355] $wTL[x][y] = 0$

[0356] - 否则, 如果predModeIntra大于或等于INTRA_ANGULAR58 (参见图4B), 则按顺序执行以下步骤:

[0357] 1. 变量dYPos[x]、dYFrac[x]、dYInt[x]和dY[x][y]通过invAngle (如下文根据intraPredMode详述) 推导如下:

[0358] $dYPos[x] = ((x+1) \times invAngle+2) \gg 2$

[0359] $dYFrac[x] = dYPos[x] \& 63$

[0360] $dYInt[x] = dYPos[x] \gg 6$

[0361] $dY[x][y] = y + dYInt[x]$

[0362] 2. 变量refL[x][y]、refT[x][y]、wT[y]、wL[x]和wTL[x][y]的推导方式如下:

[0363] $refL[x][y] = (dY[x][y] \ll refH-1) ? ((64-dYFrac[x]) \times sideRef[dY[x][y]] + dYFrac[x] \times sideRef[dY[x][y]+1] + 32) \gg 6 : 0$ (等式2)

[0364] $refT[x][y] = 0$

[0365] $wT[y] = 0$

[0366] $wL[x] = (dY[x][y] \ll refH-1) ? 32 \gg ((x \ll 1) \gg nScale) : 0$

[0367] $wTL[x][y] = 0$

[0368] - 否则, 将refL[x][y]、refT[x][y]、wT[y]、wL[x]和wTL[x][y]都设置为0。

[0369] 修改后的预测像素点predSamples[x][y] (其中, $x=0 \dots nTbW-1, y=0 \dots nTbH-1$) 的值推导如下:

[0370] $predSamples[x][y] = clip1Cmp((refL[x][y] \times wL[x] + refT[x][y] \times wT[y] - p[-1][-1] \times wTL[x][y] + (64-wL[x]-wT[y]+wTL[x][y]) \times predSamples[x][y] + 32) \gg 6)$

[0371] 在上文的赋值等式1中, 简化版PDPC可以使用最近邻插值代替线性插值:

[0372] $refT[x][y] = (dX[x][y] \ll refW-1) ? mainRef[dX[x][y]] : 0$

[0373] 类似地, 赋值等式2还可以使用最近邻插值:

[0374] $\text{refL}[x][y] = (\text{dY}[x][y] < \text{refH} - 1) ? \text{sideRef}[\text{dY}[x][y]] : 0$

[0375] 只有若干种模式可以省去修正。具体地,对于DC帧内预测模式,可以将wTL设置为0。因此,如果帧内预测模式为DC帧内预测模式,则PDPC不需要clip1Cmp()运算。可以采用以下规则或方法:

[0376] -DC模式下的PDPC不使用左上方(top-left,TL)像素点;

[0377] -对于除HOR和VER模式之外的任何帧内预测模式,不执行PDPC中的修正运算;

[0378] -HOR和VER模式下的PDPC使用修正运算,而不进行本申请中描述的定限运算。

[0379] 根据上文的描述,可以按以下方式修改规范:

[0380] -如果predModeIntra等于INTRA_PLANAR或INTRA_DC,则

[0381] $\text{refL}[x][y] = p[-1][y]$ (8-227)

[0382] $\text{refT}[x][y] = p[x][-1]$ (8-228)

[0383] $wT[y] = 32 \gg ((y < 1) \gg nScale)$ (8-229)

[0384] $wL[x] = 32 \gg ((x < 1) \gg nScale)$ (8-230)

[0385] $wTL[x][y] = 0$ (8-231)

[0386] -否则,如果predModeIntra等于INTRA_ANGULAR18或INTRA_ANGULAR50,则

[0387] $\text{refL}[x][y] = p[-1][y]$ (8-232)

[0388] $\text{refT}[x][y] = p[x][-1]$ (8-233)

[0389] $wT[y] = (\text{predModeIntra} == \text{INTRA_ANGULAR18}) ? 32 \gg ((y < 1) \gg nScale) : 0$ (8-234)

[0390] $wL[x] = (\text{predModeIntra} == \text{INTRA_ANGULAR50}) ? 32 \gg ((x < 1) \gg nScale) : 0$ (8-235)

[0391] $wTL[x][y] = (\text{predModeIntra} == \text{INTRA_ANGULAR18}) ? wT[y] : wL[x]$ (8-236)

[0392] -否则,如果predModeIntra等于INTRA_ANGULAR2或INTRA_ANGULAR66,则

[0393] $\text{refL}[x][y] = p[-1][x+y+1]$ (8-237)

[0394] $\text{refT}[x][y] = p[x+y+1][-1]$ (8-238)

[0395] $wT[y] = (32 \gg 1) \gg ((y < 1) \gg nScale)$ (8-239)

[0396] $wL[x] = (32 \gg 1) \gg ((x < 1) \gg nScale)$ (8-240)

[0397] $wTL[x][y] = 0$ (8-241)

[0398] -否则,如果predModeIntra小于或等于INTRA_ANGULAR10,则执行以下步骤:

[0399] 1. 变量dXPos[y]、dXFrac[y]、dXInt[y]和dX[x][y]通过invAngle(根据intraPredMode在VVC规范草案中详述)推导如下:

$\text{dXPos}[y] = ((y+1) \times \text{invAngle} + 2) \gg 2$

[0400] $\text{dXFrac}[y] = \text{dXPos}[y] \& 63$ (8-242)

$\text{dXInt}[y] = \text{dXPos}[y] \gg 6$

$\text{dX}[x][y] = x + \text{dXInt}[y]$

[0401] 2. 变量refL[x][y]、refT[x][y]、wT[y]、wL[x]和wTL[x][y]的推导方式如下:

[0402] $\text{refL}[x][y] = 0$ (8-243)

[0403] $\text{refT}[x][y] = (\text{dX}[x][y] < \text{refW} - 1) ? \text{mainRef}[\text{dX}[x][y] + (\text{dXFrac}[y] \gg 5)] : 0$ (8-244)

[0404] $wT[y] = (dX[x][y] \ll (\text{refW}-1) ? 32) \gg ((y \ll 1) \gg nScale) : 0$ (8-245)

[0405] $wL[x] = 0$ (8-246)

[0406] $wTL[x][y] = 0$ (8-247)

[0407] - 否则, 如果 predModeIntra 大于或等于 INTRA_ANGULAR58 , 则按顺序执行以下步骤:

[0408] 1. 变量 $dXPos[y]$ 、 $dXFrac[y]$ 、 $dXInt[y]$ 和 $dX[x][y]$ 通过 invAngle (根据 intraPredMode 在 VVC 规范草案中详述) 推导如下:

[0409]
$$\begin{aligned} dYPos[x] &= ((x+1) \times \text{invAngle} + 2) \gg 2 \\ dYFrac[x] &= dYPos[x] \& 63 \\ dYInt[x] &= dYPos[x] \gg 6 \\ dY[x][y] &= y + dYInt[x] \end{aligned}$$
 (8-248)

[0410] 2. 变量 $\text{refL}[x][y]$ 、 $\text{refT}[x][y]$ 、 $wT[y]$ 、 $wL[x]$ 和 $wTL[x][y]$ 的推导方式如下:

[0411] $\text{refL}[x][y] = (dY[x][y] \ll (\text{refH}-1) ? \text{sideRef}[dY[x][y] + (dYFrac[x] \gg 5)] : 0$ (8-249)

[0412] $\text{refT}[x][y] = 0$ (8-250)

[0413] $wT[y] = 0$ (8-251)

[0414] $wL[x] = (dY[x][y] \ll (\text{refH}-1) ? 32) \gg ((x \ll 1) \gg nScale) : 0$ (8-252)

[0415] $wTL[x][y] = 0$ (8-253)

[0416] - 否则, 将 $\text{refL}[x][y]$ 、 $\text{refT}[x][y]$ 、 $wT[y]$ 、 $wL[x]$ 和 $wTL[x][y]$ 都设置为 0。

[0417] 修改后的预测像素点 $\text{predSamples}[x][y]$ (其中, $x=0 \dots nTbW-1$, $y=0 \dots nTbH-1$) 的值推导如下:

[0418] - 如果 predModeIntra 等于 INTRA_ANGULAR18 或 INTRA_ANGULAR50 , 则 $\text{predSamples}[x][y] = \text{clip1Cmp}((\text{refL}[x][y] \times wL[x] + \text{refT}[x][y] \times wT[y] - p[-1][-1] \times wTL[x][y] + (64 - wL[x] - wT[y] + wTL[x][y]) \times \text{predSamples}[x][y] + 32) \gg 6) (1*)$

[0419] - 否则,

[0420] $\text{predSamples}[x][y] = (\text{refL}[x][y] \times wL[x] + \text{refT}[x][y] \times wT[y] + (64 - wL[x] - wT[y]) \times \text{predSamples}[x][y] + 32) \gg 6$ 。

[0421] 根据上文的规范文本, 对 PDPC 方法进行以下修改:

[0422] - 当执行 DC 模式下的 PDPC 时, 不考虑使用左上方像素点;

[0423] - 只在水平和垂直帧内预测模式下, 对预测像素点的 PDPC 输出值进行修正。在所有其它帧内预测模式下, 不执行修正。

[0424] 在编码器和解码器两侧, 所提供的方法使用以下内容作为输入数据:

[0425] • 方向性帧内预测模式 (还表示为 predModeIntra , 如图 10B 和图 10C 所示);

[0426] • 块大小参数 $nTbS$, 设置为 $(\log_2(nTbW) + \log_2(nTbH)) \gg 1$, 其中, $nTbW$ 和 $nTbH$ 分别表示预测块的宽度和高度, “ \gg ” 表示右移运算。

[0427] VVC 规范经过修改后能够使用所提供的方法, 修改内容可以包括在描述简化版 PDPC 的章节中使用 “参考像素点 $p[x][y]$ ” 代替 “相邻像素点 $p[x][y]$ ”。

[0428] 角度参数 intraPredAngle 表示定点表示的相邻两行预测像素点之间的子像素偏移, 其分数部分的长度等于 5 位。该参数可以根据帧内预测模式 (predModeIntra) 推导。可以

[0446] 其中, $R_{x,-1}$ 和 $R_{-1,y}$ 表示位于当前像素点 (x,y) 上方和左侧的参考像素点。值得注意的是, 等式 (2) 中不使用函数 `clip1Cmp`, 因为所述预测像素点 $P(x,y)$ 的值始终在有效值的范围内, 即在像素值的最小值和最大值之间。

[0447] “ $x \gg y$ ”表示 x 的补码整数表示向右算术移动 y 个二进制位。只有 y 为非负整数值时才有这个函数定义。右移的结果是移进最高有效位 (most significant bit, MSB) 的比特位等于移位操作之前的 x 的 MSB。

[0448] DC模式权重计算如下:

[0449] $wT = 32 \gg ((y \ll 1) \gg \text{shift})$, $wL = 32 \gg ((x \ll 1) \gg \text{shift})$,

[0450] 其中, $\text{shift} = (\log_2(\text{width}) + \log_2(\text{height}) + 2) \gg 2$ 。

[0451] 图15示出了一个 4×4 块内的位置 $(0,0)$ 和 $(1,0)$ 的DC模式PDPC权重 (wL, wT)。可以注意到, 与图13B相比, 未使用左上方参考像素点, 所有没有指定该像素点的权重。

[0452] 本替换方法可以用一部分VVC规范表示:

[0453] 位置相关帧内预测像素点滤波过程

[0454] 该过程的输入包括:

[0455] - 帧内预测模式 `predModeIntra`;

[0456] - 变量 `nTbW`, 表示变换块宽度;

[0457] - 变量 `nTbH`, 表示变换块高度;

[0458] - 变量 `refW`, 表示参考像素点宽度;

[0459] - 变量 `refH`, 表示参考像素点高度;

[0460] - 预测像素点 `predSamples[x][y]`, 其中, $x = 0 \cdots nTbW - 1, y = 0 \cdots nTbH - 1$;

[0461] - 相邻像素点 `p[x][y]`, 其中, $x = -1, y = -1 \cdots refH - 1; x = 0 \cdots refW - 1, y = -1$;

[0462] - 变量 `cIdx`, 表示当前块的颜色分量。

[0463] 该过程的输出为修改后的预测像素点 `predSamples[x][y]`, 其中, $x = 0 \cdots nTbW - 1, y = 0 \cdots nTbH - 1$ 。

[0464] 根据 `cIdx` 的值, 将函数 `clip1Cmp` 设置如下:

[0465] - 如果 `cIdx` 等于 0, 则将 `clip1Cmp` 设置为 `Clip1y`;

[0466] - 否则, 将 `clip1Cmp` 设置为 `Clip1c`。

[0467] 将变量 `nScale` 设置为 $((\text{Log}_2(nTbW) + \text{Log}_2(nTbH) - 2) \gg 2)$ 。

[0468] 参考像素点阵列 `mainRef[x]` 和 `sideRef[y]` (其中, $x = 0 \cdots nTbW - 1, y = 0 \cdots nTbH - 1$) 的推导方式如下:

[0469] $\text{refT}[x][y] = p[x][-1]$

[0470] $\text{sideRef}[y] = p[-1][y]$

[0471] 变量 `refL[x][y]`、`refT[x][y]`、`wT[y]` 和 `wL[x]` (其中, $x = 0 \cdots nTbW - 1, y = 0 \cdots nTbH - 1$) 的推导方式如下:

[0472] - 如果 `predModeIntra` 等于 `INTRA_PLANAR`、`INTRA_DC`、`INTRA_ANGULAR18` 或 `INTRA_ANGULAR50`, 则

[0473] $\text{refL}[x][y] = p[-1][y]$

[0474] $\text{refT}[x][y] = p[x][-1]$

[0475] $wT[y] = 32 \gg ((y \ll 1) \gg nScale)$

[0476] $wL[x] = 32 \gg ((x \ll 1) \gg nScale)$

[0477] - 否则, 如果predModeIntra等于INTRA_ANGULAR2或INTRA_ANGULAR66, 则

[0478] $refL[x][y] = p[-1][x+y+1]$

[0479] $refT[x][y] = p[x+y+1][-1]$

[0480] $wT[y] = (32 \gg 1) \gg ((y \ll 1) \gg nScale)$

[0481] $wL[x] = (32 \gg 1) \gg ((x \ll 1) \gg nScale)$

[0482] - 否则, 如果predModeIntra小于或等于INTRA_ANGULAR10, 则按顺序执行以下步骤:

[0483] 1. 变量dXPos[y]、dXFrac[y]、dXInt[y]和dX[x][y]通过invAngle(根据intraPredMode而定)推导如下:

[0484] $dXPos[y] = ((y+1) \times invAngle+2) \gg 2$

[0485] $dXFrac[y] = dXPos[y] \& 63$

[0486] $dXInt[y] = dXPos[y] \gg 6$

[0487] $dX[x][y] = x + dXInt[y]$

[0488] invAngle可以在VVC规范草案中详述。

[0489] 2. 变量refL[x][y]、refT[x][y]、wT[y]和wL[x]的推导方式如下:

[0490] $refL[x][y] = 0$

[0491] $refT[x][y] = (dX[x][y] \ll refW-1) ? ((64-$

[0492] $dXFrac[y]) \times mainRef[dX[x][y]] + dXFrac[y] \times mainRef[dX[x][y]+1] + 32) \gg 6 : 0$

[0493] $wT[y] = (dX[x][y] \ll refW-1) ? 32 \gg ((y \ll 1) \gg nScale) : 0$

[0494] $wL[x] = 0$

[0495] - 否则, 如果predModeIntra大于或等于INTRA_ANGULAR58, 则按顺序执行以下步骤:

[0496] 1. 变量dYPos[x]、dYFrac[x]、dYInt[x]和dY[x][y]通过invAngle(根据intraPredMode而定)推导如下:

[0497] $dYPos[x] = ((x+1) \times invAngle+2) \gg 2$

[0498] $dYFrac[x] = dYPos[x] \& 63$

[0499] $dYInt[x] = dYPos[x] \gg 6$

[0500] $dY[x][y] = y + dYInt[x]$

[0501] invAngle可以在VVC规范草案中详述。

[0502] 2. 变量refL[x][y]、refT[x][y]、wT[y]和wL[x]的推导如下:

[0503] $refL[x][y] = (dY[x][y] \ll refH-1) ? ((64 - dYFrac[x]) \times sideRef[dY[x][y]] +$

[0504] $dYFrac[x] \times sideRef[dY[x][y]+1] + 32) \gg 6 : 0$

[0505] $refT[x][y] = 0$

[0506] $wT[y] = 0$

[0507] $wL[x] = (dY[x][y] \ll refH-1) ? 32 \gg ((x \ll 1) \gg nScale) : 0$

[0508] - 否则, 将refL[x][y]、refT[x][y]、wT[y]和wL[x]都设置为0。修改后的预测像素点predSamples[x][y](其中, $x=0 \cdots nTbW-1$, $y=0 \cdots nTbH-1$)的值推导如下:

[0509] $predSamples[x][y] = (refL[x][y] \times wL[x] + refT[x][y] \times wT[y] + (64 - wL[x] - wT$

[y]) \times predSamples[x][y]+32) \gg 6)

[0510] 这里, $(64-wL[x]-wT[y])$ 表示像素点加权因子。

[0511] 图16示出了上文描述的方法。虚线示出了修正步骤,此步骤在最新PDPC中执行,但是由于只有负系数wTL不使用,因此不需要该步骤,所以所提供的方法中不执行该步骤。

[0512] 在图17中,帧内预测过程使用参考像素点来生成预测像素点。此外,使用像素点加权因子对每个预测像素点进行加权。所述像素点加权因子可以等于 $(64-wL[x]-wT[y])$ 等。使用相同的参考像素点根据x和y计算每个预测像素点的额外值,其中,x和y限定预测像素点在预测块中的位置。将这些额外值与相应的加权预测像素点相加。然后,根据像素点加权因子的预定精度,通过右移将本操作得到的每个像素点进行归一化。例如,如果所述像素点加权因子定义为 $(64-wL[x]-wT[y])$,则精度为6位。因此,在本步骤中,向右移动6位,以保证输出值的可能最小值和最大值与参考像素点的可能最小值和最大值相同。

[0513] 所提供的技术方案的一个有益效果在于,可以复用平面帧内预测机制来计算额外值。具体地,平面帧内预测使用以下等式来推导水平和垂直预测像素点值:

$$[0514] \quad \text{predV}[x][y] = ((nTbH-1-y) \times p[x][-1] + (y+1) \times p[-1][nTbH]) \ll \text{Log2}(nTbW) \quad (8-82)$$

$$[0515] \quad \text{predH}[x][y] = ((nTbW-1-x) \times p[-1][y] + (x+1) \times p[nTbW][-1]) \ll \text{Log2}(nTbH) \quad (8-83)$$

[0516] 从上述两个等式可以看出,predV[x][y]使用与predV[x][y]位于同一列的参考像素点 $p[x][-1]$,predH[x][y]使用与predH[x][y]位于同一行的参考像素点 $p[-1][y]$ 。除此之外,最后一个步骤是执行左移运算,由于左移运算不会影响复用的中间计算,所以可以跳过。 $nTbW$ 、 $nTbH$ 、变量x和y为平面帧内预测方法的输入,因此可以相应地调整。由此,可以使用输入变量 D_x 代替 $(nTbW-1-x)$ 以及使用输入变量 D_y 代替 $(nTbH-1-y)$ 。由于左下方和右上方参考像素为不使用的参数,所以可以设置为0。

[0517] 根据上述观察结果,上述等式可以根据预先确定的输入进行改写:

$$[0518] \quad V_y = D_y \times p[x][-1]$$

$$[0519] \quad V_x = D_x \times p[-1][y]$$

[0520] 因此,可以统一执行以下步骤:

[0521] -水平模式(模式18)情况下的额外值可以计算为 $V_y = D_y \times p[x][-1]$,其中,将 D_y 设置为 $wT[y]$;

[0522] -垂直模式(模式50)情况下的额外值可以计算为 $V_x = D_x \times p[-1][y]$,其中,将 D_x 设置为 $wL[y]$;

[0523] -DC模式(模式1)情况下的额外值可以计算为 $V_y + V_x$,其中,与前两种情况相同,将 D_y 设置为 $wT[y]$,将 D_x 设置为 $wL[y]$ 。

[0524] 通过改变参考像素点选择,可以表明,为PDPC指定的所有帧内预测模式可以执行统一处理。

[0525] 当指定帧内预测模式为DC、水平或垂直帧内预测模式时,需要对得到的更新后预测像素点进行修正,(例如)如图16中带虚线边的方框所示。

[0526] 在水平或垂直帧内预测模式的情况下,更好的方法是,当加权预测像素点与额外值相加时,将修正运算放置在简化版PDPC的最后一个步骤之前。对于如上所述的不同模式,

按不同的方式获取该额外值。

[0527] 本发明实施例提出,对中间计算的结果进行阈值化,执行这些中间计算是为了在帧内预测中使用垂直或水平帧内预测模式时获取额外值。对于这些情况,可以通过与加权预测像素点相加来计算额外值,这样将不需要修正运算。

[0528] 最新PDPC方法对更新后的预测像素点进行修正(即使在水平和垂直模式下)。具体地, $\text{predSamples}[x][y] = \text{clip1Cmp}((\text{refL}[x][y] \times wL[x] + \text{refT}[x][y] \times wT[y] - p[-1][-1] \times wTL[x][y] + (64 - wL[x] - wT[y] + wTL[x][y]) \times \text{predSamples}[x][y] + 32) \gg 6)$

[0529] 对于水平帧内预测模式,将 $wTL[x][y]$ 设置为 $wT[y]$,因此得到简化版表达式:

[0530] $\text{predSamples}[x][y] = \text{clip1Cmp}(((\text{refT}[x][y] - p[-1][-1]) \times wT[y] + 64 \times \text{predSamples}[x][y] + 32) \gg 6)$

[0531] 在该表达式中,“ $\text{refT}[x][y] - p[-1][-1]$ ”实际上表示额外参考像素点值。即,对于帧内预测模式为水平帧内预测模式的情况,将额外参考值设置为位于预测像素点上方的最近参考像素点($\text{refT}[x][y]$)的值与左上方参考像素点($p[-1][-1]$)的值之间的差值。

[0532] 对于帧内预测模式为垂直帧内预测模式的情况,将额外参考值设置为位于预测像素点左侧的最近参考像素点($\text{refL}[x][y]$)的值与左上方参考像素点($p[-1][-1]$)的值之间的差值,即“ $\text{refT}[x][y] - p[-1][-1]$ ”。

[0533] 因此,可以使用对额外参考像素点值(即“ $\text{refT}[x][y] - p[-1][-1]$ ”)进行的定限运算来替换 clipCmp 函数。

[0534] 对于水平帧内预测模式,

[0535] $\text{predSamples}[x][y] = (T(\text{refT}[x][y] - p[-1][-1]) \times wT[y] + 64 \times \text{predSamples}[x][y] + 32) \gg 6,$

[0536] 其中, $T()$ 为定限运算。

[0537] 对于垂直帧内预测模式,

[0538] $\text{predSamples}[x][y] = (T(\text{refL}[x][y] - p[-1][-1]) \times wT[y] + 64 \times \text{predSamples}[x][y] + 32) \gg 6,$

[0539] 其中, $T()$ 为定限运算。

[0540] 图17给出了 $T()$ 的示例性实现方式。额外参考像素点值的定限运算 $T()$ 实际上为根据左上方参考像素点值是否大于预测像素点值来更新额外参考像素点值。具体地,

[0541] -当所述左上方参考像素点值大于所述预测像素点值时,下限值为所述预测像素点的最大值减去预测像素点值得到的差值,将所述更新后的额外参考像素点值设置为两个值中的最大值,其中,

[0542] o所述第一值为所述额外参考像素点值,

[0543] o所述第二值为所述下限值;

[0544] -否则,上限值为所述预测像素点的最小值减去预测像素点值得到的差值,将所述更新后的额外参考像素点值设置为两个值中的最小值。

[0545] o所述第一值为所述额外参考像素点值,

[0546] o所述第二值为所述上限值。

[0547] 在图17中,当帧内预测为水平帧内预测时,最接近参考像素点为位于所述预测像素点左侧的最近参考像素点。当帧内预测为垂直帧内预测时,最接近参考像素点为位于所

述预测像素点上方的最近参考像素点。从图17可以注意到,比较结果用于在以下值之间切换:

[0548] -定限函数的结果,即最小值和最大值,

[0549] -定限函数中的其中一个参数。

[0550] 值得注意的是,Clip3()函数提供了两个定限运算。一个阈值为所述预测像素点的最小值,另一个阈值为所述预测像素点的最大值。与应用到更新后的预测像素点的Clip3()函数相比(图16),所提出的定限运算(图17)只使用一个阈值:所述预测像素点的最小值或所述预测像素点的最大值。

[0551] 还可以注意到,上文描述的改变定限运算的位置能够使用相同的等式,无需对PDPC适用的所有帧内预测模式进行修正。

[0552] 图18为对图像中的块进行帧内预测的装置1800的一种示例性结构的框图。装置1800用于执行上述方法并可以包括:

[0553] 获取单元1810,用于:使用帧内预测模式对一个或多个参考像素点值进行帧内预测以获取预测像素点值;根据所述帧内预测模式,获取至少一个额外参考像素点值;根据所述额外参考像素点值,获取阈值化的额外参考像素点值;

[0554] 计算单元1820,用于:根据所述阈值化的额外参考像素点值,计算额外值;将所述预测像素点值与像素点加权因子相乘,得到加权预测像素点值;将所述加权预测像素点值与所述额外值相加,得到非归一化的预测像素点值;对所述非归一化的预测像素点值进行归一化,得到归一化的预测像素点值。

[0555] 当所述帧内预测模式为垂直帧内预测模式时,可以将所述额外参考像素点值设置为位于预测像素点上方的最近参考像素点的值与左上方参考像素点的值之间的差值。

[0556] 当所述帧内预测模式为水平帧内预测模式时,可以将所述额外参考像素点值设置为位于预测像素点左侧的最近参考像素点的值与左上方参考像素点的值之间的差值。

[0557] 当所述帧内预测模式为DC帧内预测模式时,所述至少一个额外参考像素点值包括第一额外参考像素点值和第二额外参考像素点值;可以通过以下方式获取所述第一额外参考像素点值和所述第二额外参考像素点值:

[0558] 将所述第一额外参考像素点值设置为所述位于预测像素点左侧的最近参考像素点的值,

[0559] 将所述第二额外参考像素点值设置为所述位于预测像素点上方的最近参考像素点的值。

[0560] 当所述左上方参考像素点值大于或等于所述预测像素点值时,上限值为所述预测像素点的最大值减去所述预测像素点值得到的差值,将所述阈值化的额外参考像素点值设置为第一值和第二值中的最大值,其中,

[0561] o所述第一值为所述额外参考像素点值,

[0562] o所述第二值为所述上限值。

[0563] 当所述左上方参考像素点值小于所述预测像素点值时,下限值为所述预测像素点的最小值减去所述预测像素点值得到的差值,将所述阈值化的额外参考像素点值设置为第一值和第二值中的最小值,其中,

[0564] o所述第一值为所述额外参考像素点值,

[0565] o所述第二值为所述下限值。

[0566] 下面对上述实施例中所示的编码方法和解码方法的应用以及使用这些应用的系统进行解释说明。

[0567] 图19为用于实现内容分发服务的内容供应系统3100的框图。内容供应系统3100包括捕获设备3102、终端设备3106,并且可选地包括显示器3126。捕获设备3102通过通信链路3104与终端设备3106通信。通信链路可以包括上文描述的通信信道13。通信链路3104包括但不限于Wi-Fi、以太网、有线、无线(3G/4G/5G)、USB或其任何种类的组合等。

[0568] 捕获设备3102生成数据,并可以通过如上述实施例中所示的编码方法对数据进行编码。可替换地,捕获设备3102可以将数据分发到流媒体服务器(图中未示出),该服务器对数据进行编码并将编码数据发送到终端设备3106。捕获设备3102包括但不限于相机、智能手机或平板电脑、计算机或笔记本电脑、视频会议系统、PDA、车载设备,或其中任何一个的组合等。例如,捕获设备3102可以包括上文所述的源设备12。当数据包括视频时,包括在捕获设备3102中的视频编码器20可以实际执行视频编码处理。当数据包括音频(即语音)时,包括在捕获设备3102中的音频编码器可以实际执行音频编码处理。对于一些实际场景,捕获设备3102通过将经编码的视频数据和经编码的音频数据一起复用来分发经编码的视频数据和经编码的音频数据。对于其它实际场景,例如在视频会议系统中,不复用经编码的音频数据和经编码的视频数据。捕获设备3102将经编码的音频数据和经编码的视频数据分别分发到终端设备3106。

[0569] 在内容供应系统3100中,终端设备310接收并再生成经编码的数据。终端设备3106可以是具有数据接收和恢复能力的设备,例如智能手机或平板电脑3108、计算机或笔记本电脑3110、网络视频录像机(network video recorder,NVR)/数字视频录像机(digital video recorder,DVR)3112、电视3114、机顶盒(set top box,STB)3116、视频会议系统3118、视频监控系统3120、个人数字助理(personal digital assistant,PDA)3122、车载设备3124,或能够对上述经编码的数据进行解码的以上设备中任何一个的组合等。例如,终端设备3106可以包括如上所述的目标设备14。当经编码的数据包括视频时,包括在终端设备中的视频解码器30优先进行视频解码。当经编码的数据包括音频时,包括在终端设备中的音频解码器优先进行音频解码处理。

[0570] 对于具有显示器的终端设备,例如智能手机或平板电脑3108、计算机或笔记本电脑3110、网络视频录像机(network video recorder,NVR)/数字视频录像机(digital video recorder,DVR)3112、电视3114、个人数字助理(personal digital assistant,PDA)3122或车载设备3124,终端设备可以将解码数据馈送到其显示器。对于不配备显示器的终端设备,例如STB 3116、视频会议系统3118或视频监控系统3120,在其中连接外部显示器3126以接收和显示解码数据。

[0571] 当此系统中的每个设备执行编码或解码时,可以使用如上述实施例中所示的图像编码设备或图像解码设备。

[0572] 图20为终端设备3106的一个示例的结构的示意图。在终端设备3106从捕获设备3102接收到流之后,协议处理单元3202分析该流的传输协议。所述协议包括但不限于实时流媒体协议(Real Time Streaming Protocol,RTSP)、超文本传输协议(Hyper Text Transfer Protocol,HTTP)、HTTP直播流媒体协议(HTTP Live streaming protocol,HLS)、

MPEG-DASH、实时传输协议 (Real-time Transport protocol, RTP)、实时消息传输协议 (Real Time Messaging Protocol, RTMP), 或其任何种类的组合等。

[0573] 在协议处理单元3202对流进行处理之后, 生成流文件。文件被输出到解复用单元3204。解复用单元3204可以将复用数据分离为经编码的音频数据和经编码的视频数据。如上文所述, 对于一些实际场景, 例如在视频会议系统中, 不复用经编码的音频数据和经编码的视频数据。在这种情况下, 经编码数据被发送到视频解码器3206和音频解码器3208, 而不通过解复用单元3204。

[0574] 通过解复用处理, 生成视频基本流 (elementary stream, ES)、音频ES和可选的字幕。视频解码器3206, 包括如上述实施例中说明的视频解码器30, 通过如上述实施例中所示的解码方法对视频ES进行解码以生成视频帧, 并将此数据馈送到同步单元3212。音频解码器3208对音频ES进行解码以生成音频帧, 并将此数据馈送到同步单元3212。可替换地, 在将视频帧馈送到同步单元3212之前可以将视频帧存储在缓冲器 (图Y中未示出) 中。类似地, 在将音频帧馈送到同步单元3212之前可以将音频帧存储在缓冲器 (图Y中未示出) 中。

[0575] 同步单元3212同步视频帧和音频帧, 并将视频/音频提供给视频/音频显示器3214。例如, 同步单元3212同步视频和音频信息的呈现。信息可以使用经译码的音频和可视数据的呈现有关的时间戳以及与数据流本身的传送有关的时间戳而以语法进行译码。

[0576] 如果流中包括字幕, 则字幕解码器3210对字幕进行解码, 使字幕与视频帧和音频帧同步, 并将视频/音频/字幕提供给视频/音频/字幕显示器3216。

[0577] 本发明并不限于上述系统, 并且上述实施例中的图像编码设备或图像解码设备都可以结合到其它系统 (例如汽车系统) 中。

[0578] 尽管本发明实施例主要根据视频译码进行了描述, 但需要说明的是, 译码系统10、编码器20和解码器30 (相应地, 系统10) 的实施例以及本文描述的其它实施例也可以用于静止图像处理或译码, 即对视频译码中独立于任何先前或连续图像的单个图像进行处理或译码。一般而言, 如果图像处理译码仅限于单个图像17, 则只有帧间预测单元244 (编码器) 和344 (解码器) 不可用。视频编码器20和视频解码器30的所有其它功能 (也称为工具或技术) 同样可以用于静止图像处理, 例如残差计算204/304、变换206、量化208、反量化210/310、(逆) 变换212/312、分割262/362、帧内预测254/354和/或环路滤波220/320、熵编码270和熵解码304。

[0579] 尽管本发明实施例主要根据视频译码进行了描述, 但需要说明的是, 译码系统10、编码器20和解码器30 (相应地, 系统10) 的实施例以及本文描述的其它实施例也可以用于静止图像处理或译码, 即对视频译码中独立于任何先前或连续图像的单个图像进行处理或译码。一般而言, 如果图像处理译码仅限于单个图像17, 则只有帧间预测单元244 (编码器) 和344 (解码器) 不可用。视频编码器20和视频解码器30的所有其它功能 (也称为工具或技术) 同样可以用于静止图像处理, 例如残差计算204/304、变换206、量化208、反量化210/310、(逆) 变换212/312、分割262/362、帧内预测254/354和/或环路滤波220/320、熵编码270和熵解码304。

[0580] 例如, 编码器20和解码器30的实施例以及本文中描述的功能 (例如参照编码器20和解码器30) 可以在硬件、软件、固件或其任意组合中实现。如果在软件中实现, 则这些功能可以作为一个或多个指令或代码存储在计算机可读介质中或通过通信介质发送, 且由基于

硬件的处理单元执行。计算机可读介质可以包括计算机可读存储介质,对应于有形介质(例如数据存储介质),或者包括任何根据通信协议等促进将计算机程序从一个地方传递到另一个地方的通信介质。通过这种方式,计算机可读介质一般可以对应于(1)非瞬时性的有形计算机可读存储介质或(2)信号或载波等通信介质。数据存储介质可以通过一个或多个计算机或一个或多个处理器访问的任何可用介质,以检索用于实施本发明所述技术的指令、代码和/或数据结构。计算机程序产品可以包括计算机可读介质。

[0581] 作为示例而非限制,这类计算机可读存储介质可以包括RAM、ROM、EEPROM、CD-ROM或其它光盘存储器、磁盘存储器或其它磁性存储设备、闪存或可以用于存储指令或数据结构形式的所需程序代码并且可以由计算机访问的任何其它介质。此外,任何连接都可以适当地称为计算机可读介质。例如,如果使用同轴缆线、光纤缆线、双绞线、数字用户线(digital subscriber line,DSL)或红外线、无线电和微波等无线技术从网站、服务器或其它远程源传输指令,则同轴缆线、光纤缆线、双绞线、DSL或红外线、无线电和微波等无线技术包含在介质的定义中。但是,应理解,计算机可读存储介质和数据存储介质并不包括连接、载波、信号或其它瞬时性介质,而是涉及非瞬时性有形存储介质。本文所使用的磁盘和光盘包含压缩光盘(compact disc,CD)、激光光盘、光学光盘、数字多功能光盘(digital versatile disc,DVD)和蓝光光盘,其中,磁盘通常以磁性方式再现数据,而光盘利用激光以光学方式再现数据。以上各项的组合也应包括在计算机可读介质的范围内。

[0582] 可以通过一个或多个数字信号处理器(digital signal processor,DSP)、一个或多个通用微处理器、一个或多个专用集成电路(application specific integrated circuit,ASIC)、一个或多个现场可编程逻辑阵列(field programmable logic array,FPGA)或其它同等集成或离散逻辑电路等一或多个处理器来执行指令。因此,本文所使用的术语“处理器”可以指上述结构中的任一种或适于实施本文所述技术的任何其它结构。另外,在一些方面中,本文描述的各种功能可以提供在用于编码和解码的专用硬件和/或软件模块内,或者并入在组合译码器中。而且,这些技术可以在一个或多个电路或逻辑元件中完全实现。本发明提及的处理电路可以包括硬件和软件。

[0583] 本发明技术可以在多种设备或装置中实现,这些设备或装置包括无线手机、集成电路(integrated circuit,IC)或一组IC(例如芯片组)。本发明描述了各种组件、模块或单元,以强调用于执行所公开技术的设备的功能方面,但未必需要由不同的硬件单元实现。相反,如上所述,各种单元可以结合合适的软件和/或固件组合在译码器硬件单元中,或者通过包括如上所述的一个或多个处理器的互操作硬件单元的集合来提供。

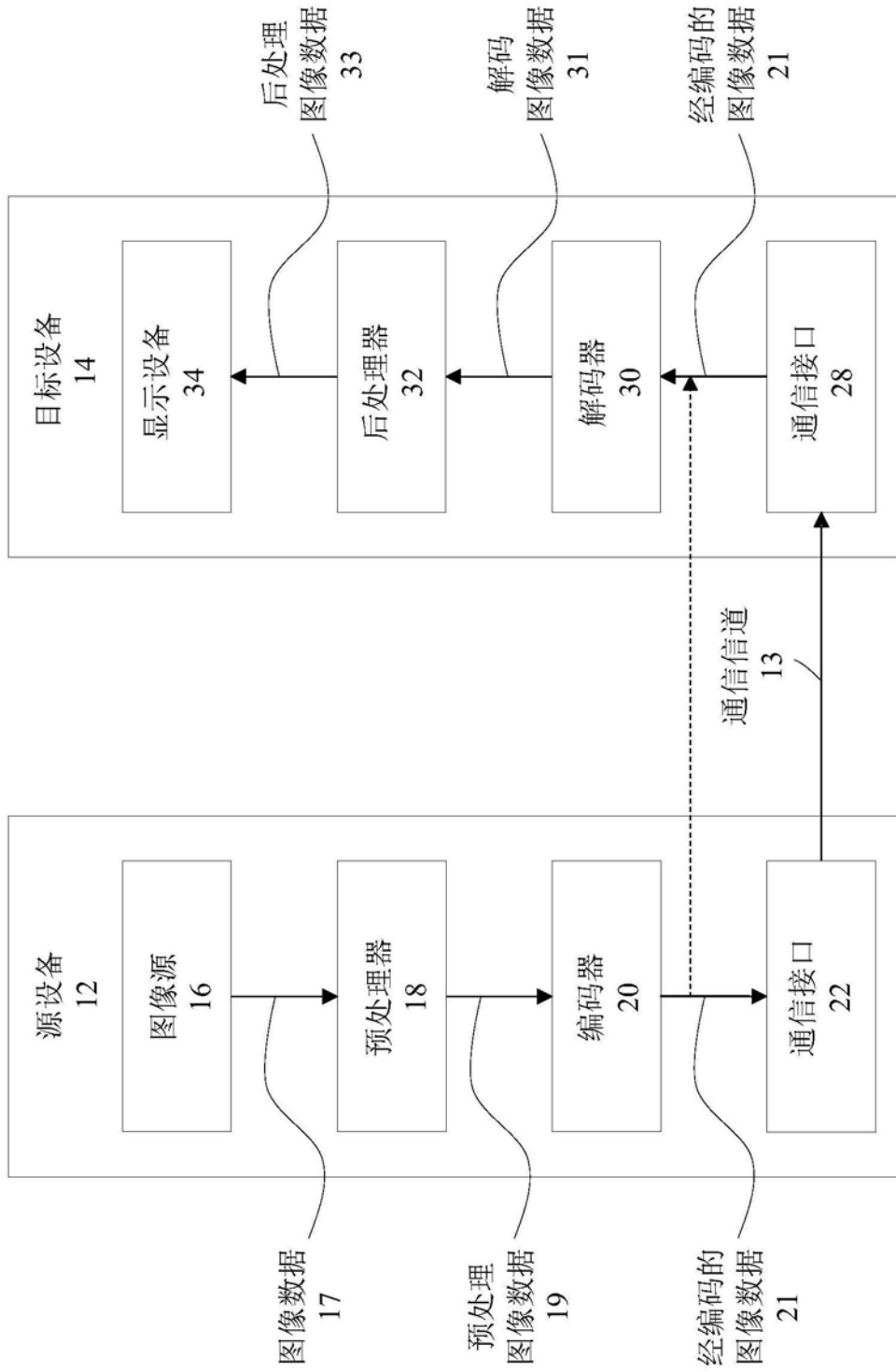


图1A

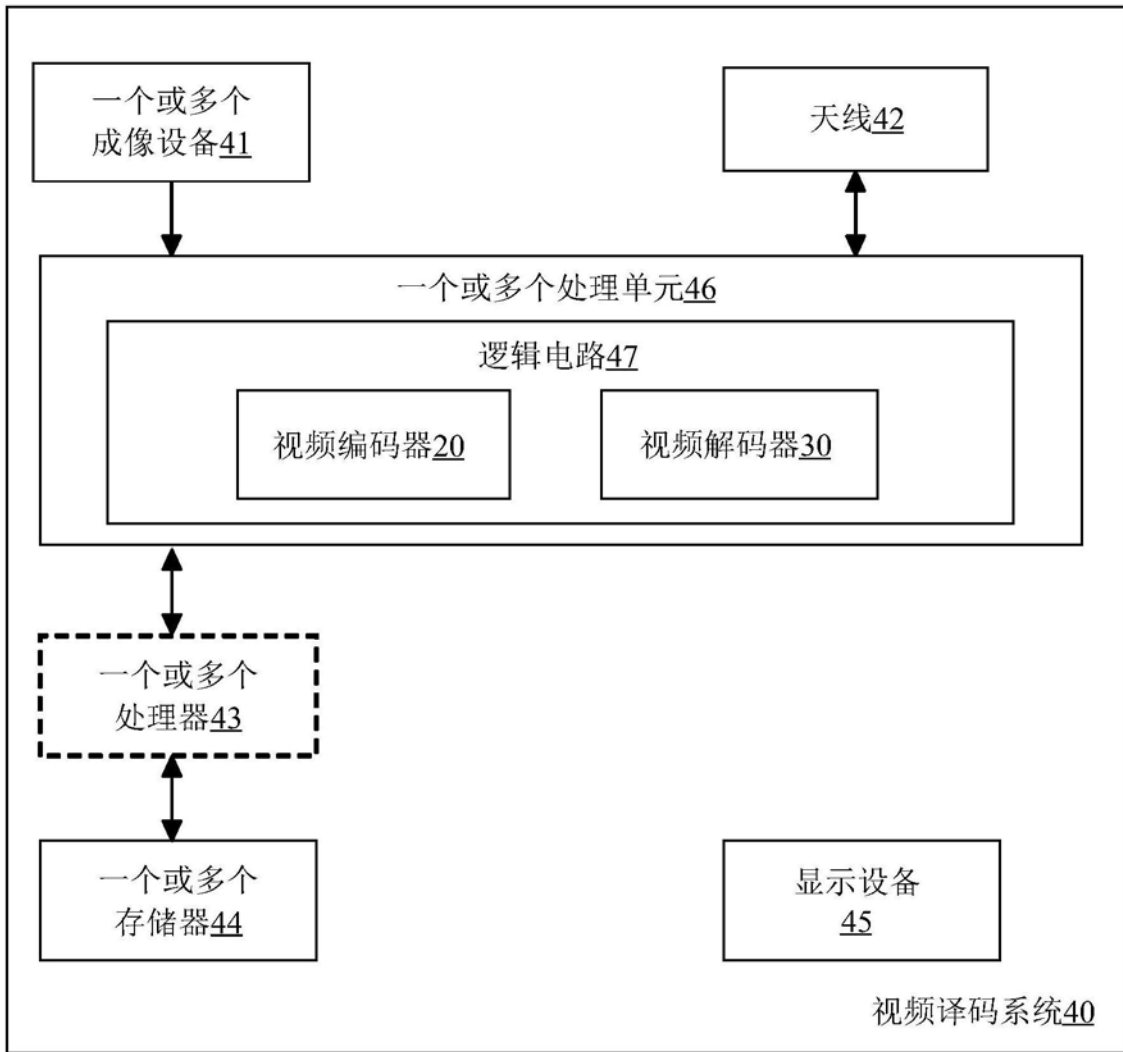


图1B

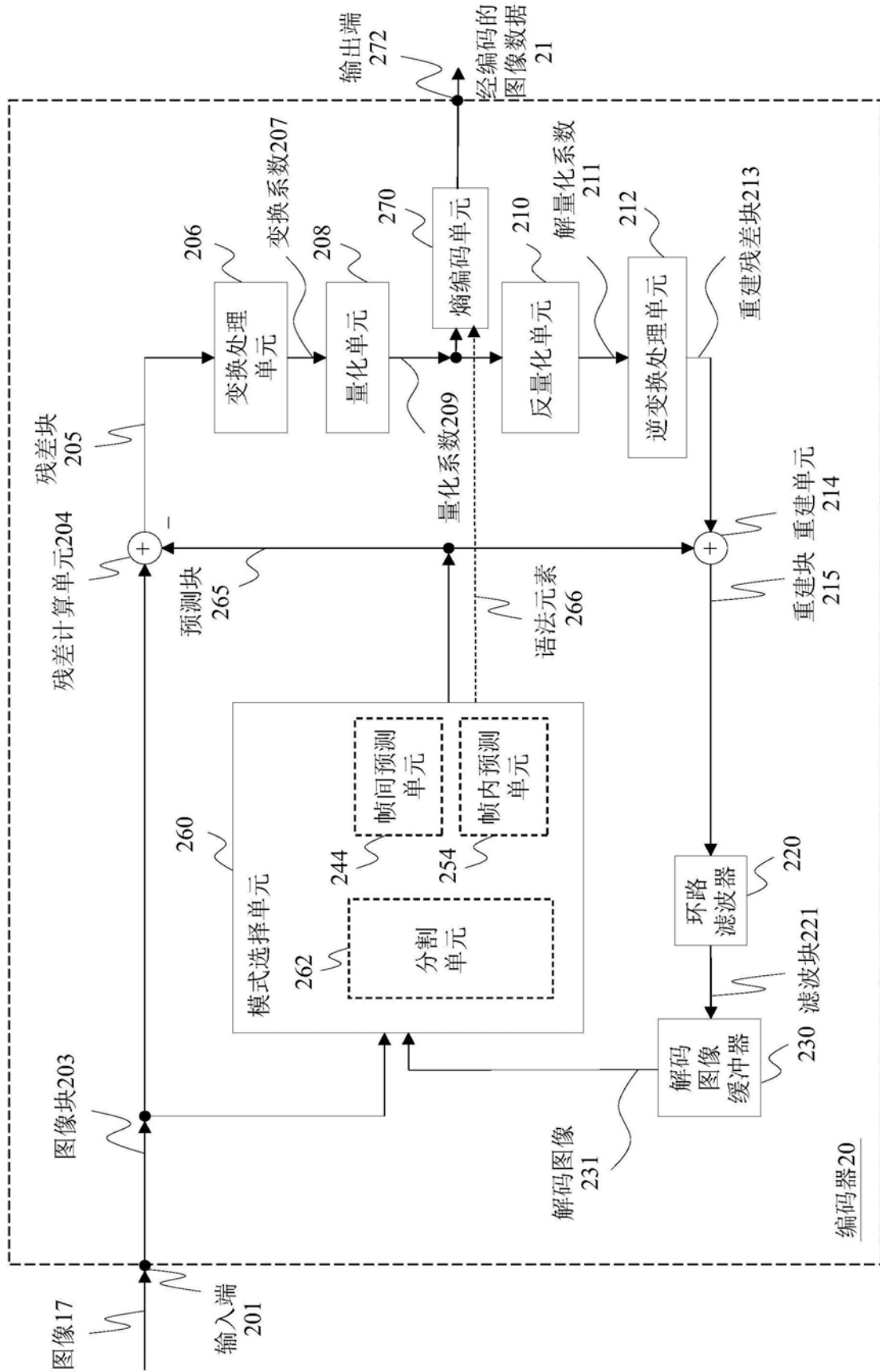


图2

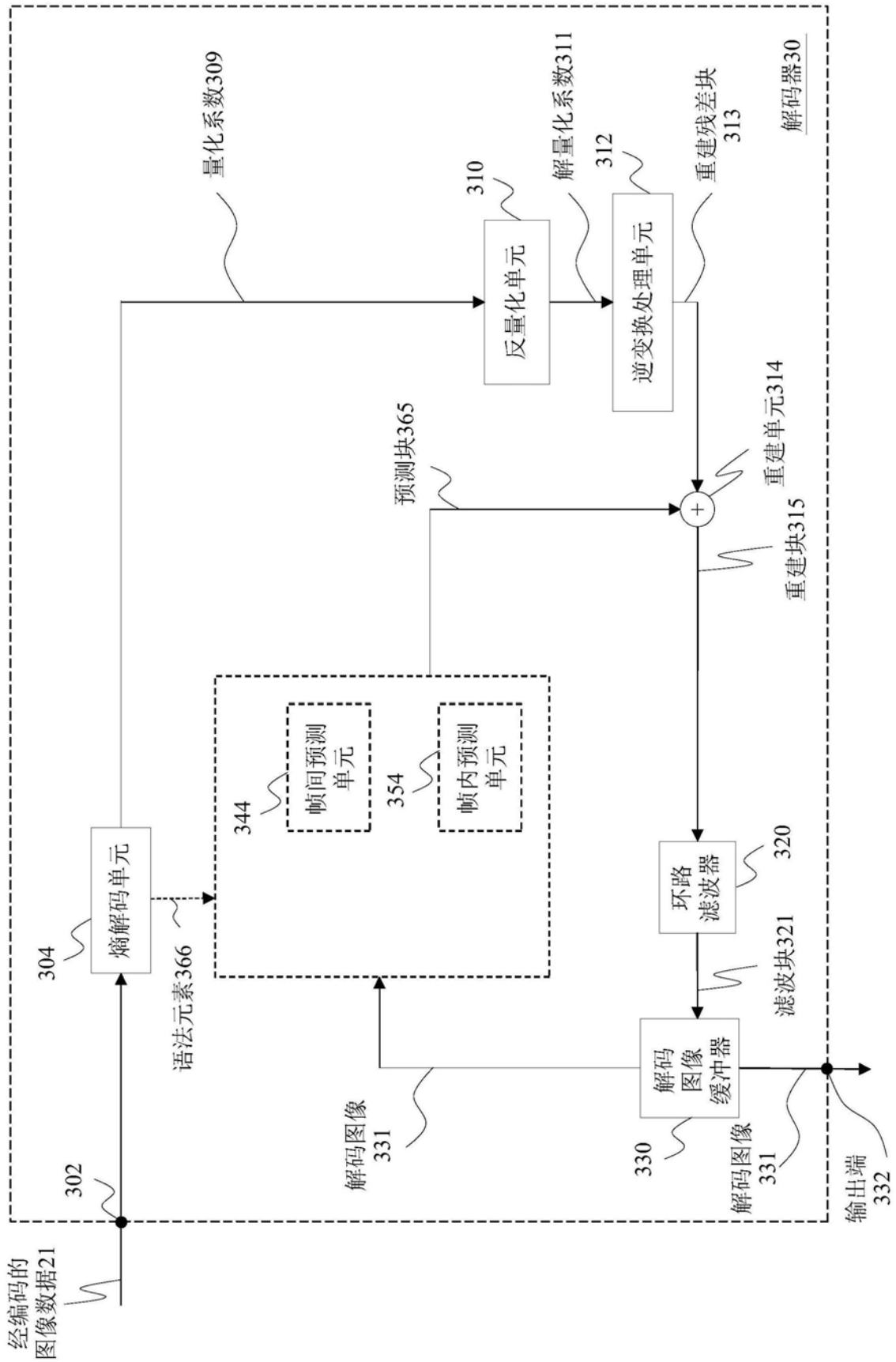


图3

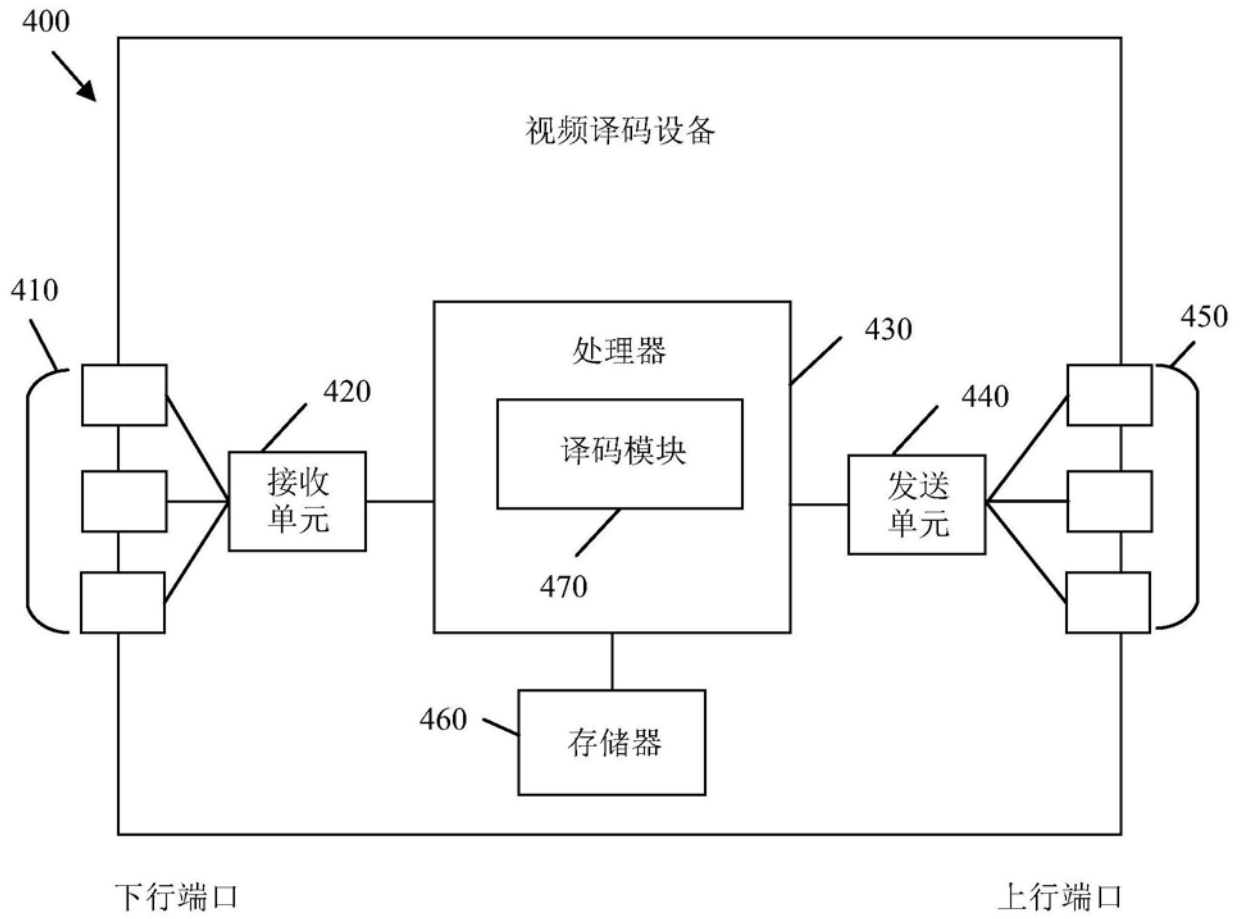


图4

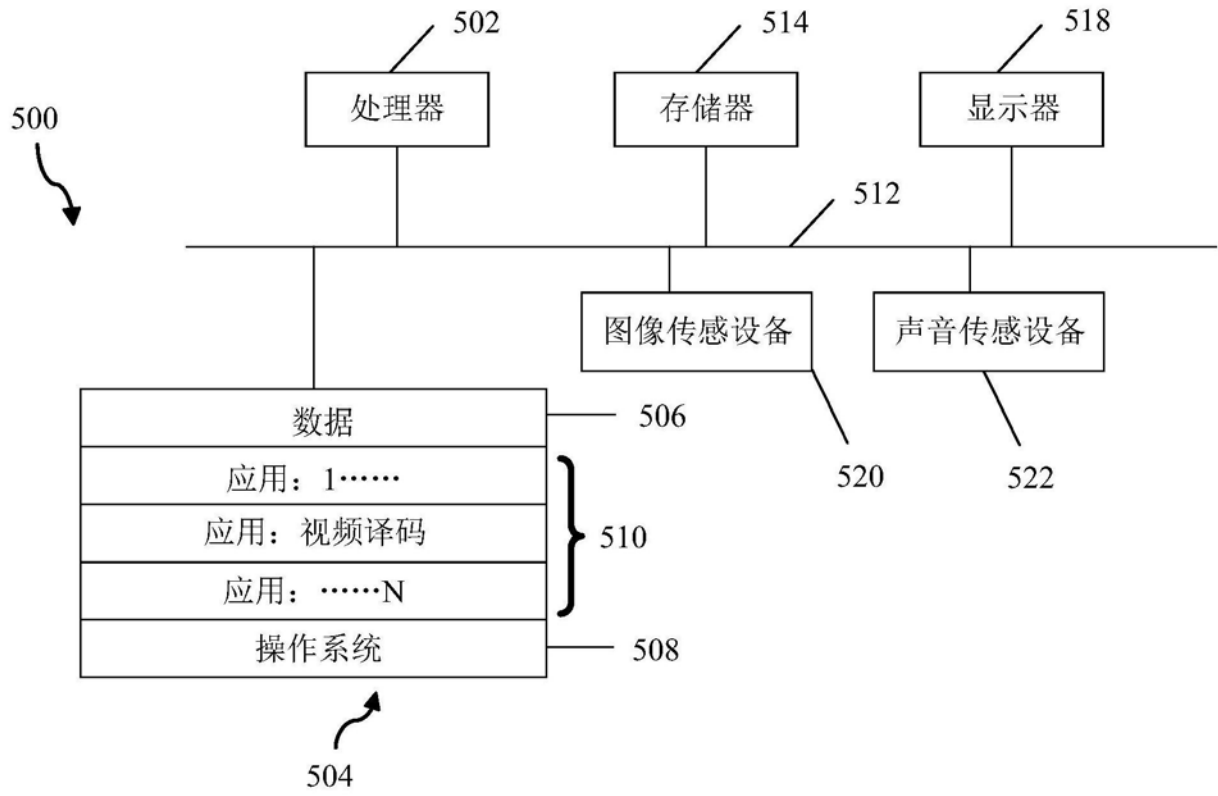


图5

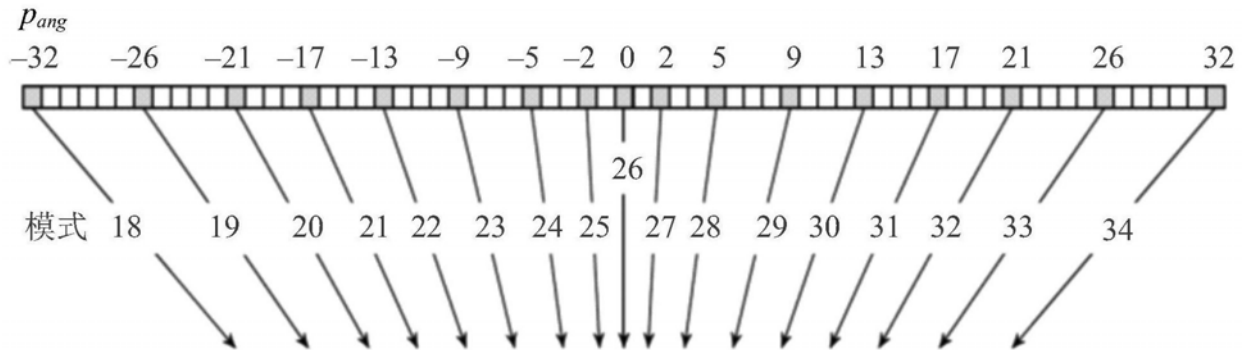


图6

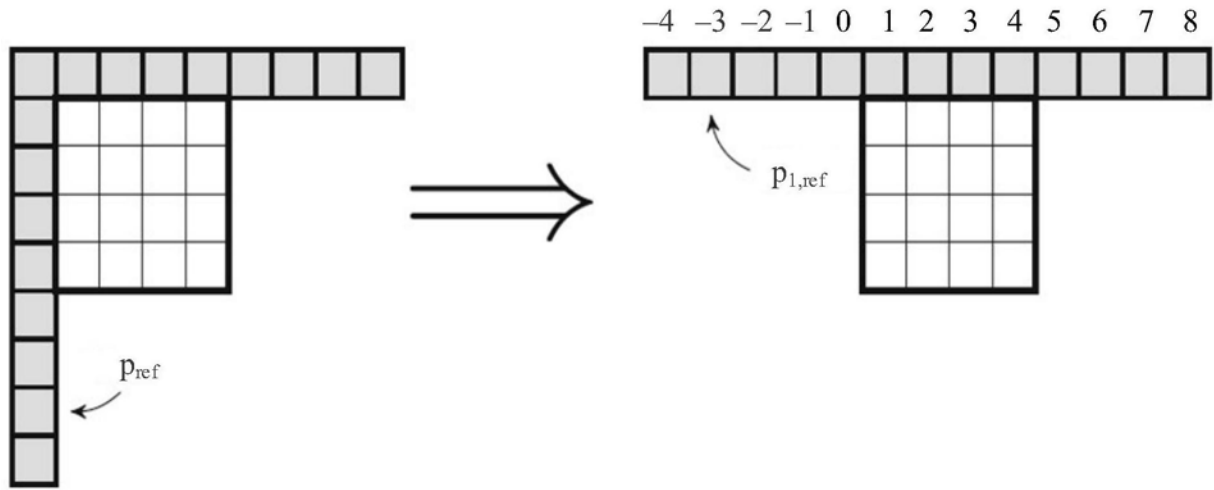


图7

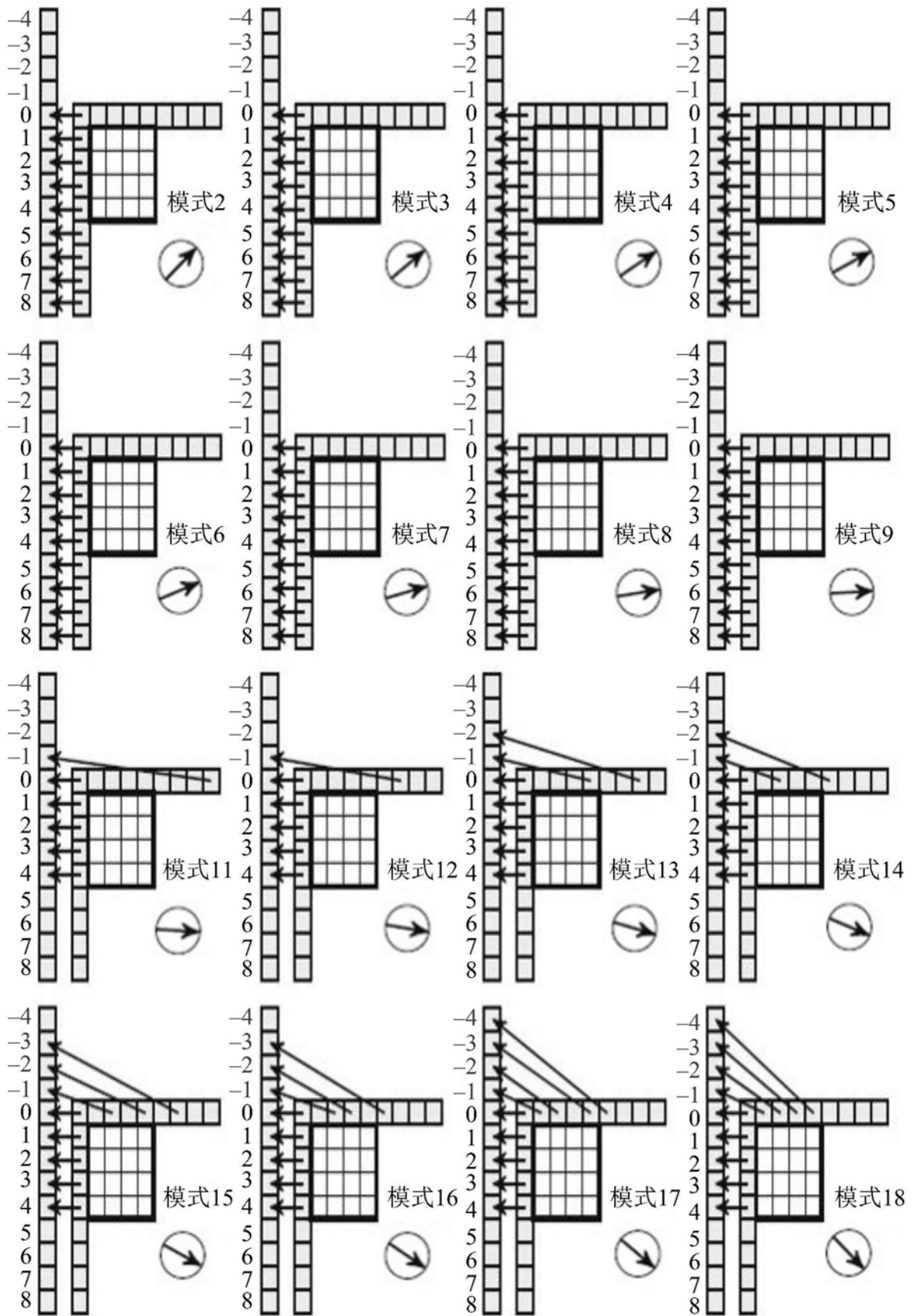


图8

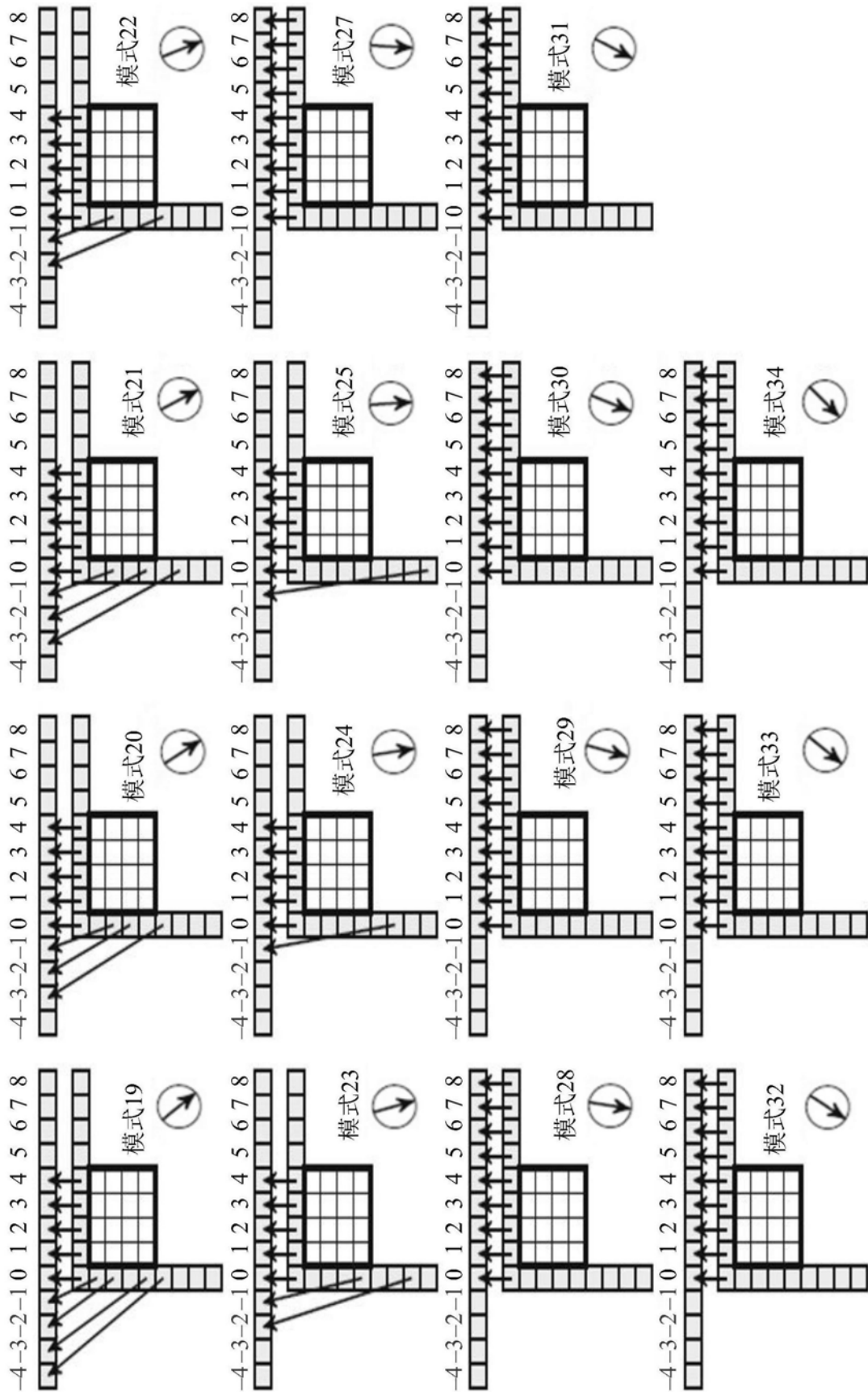


图9

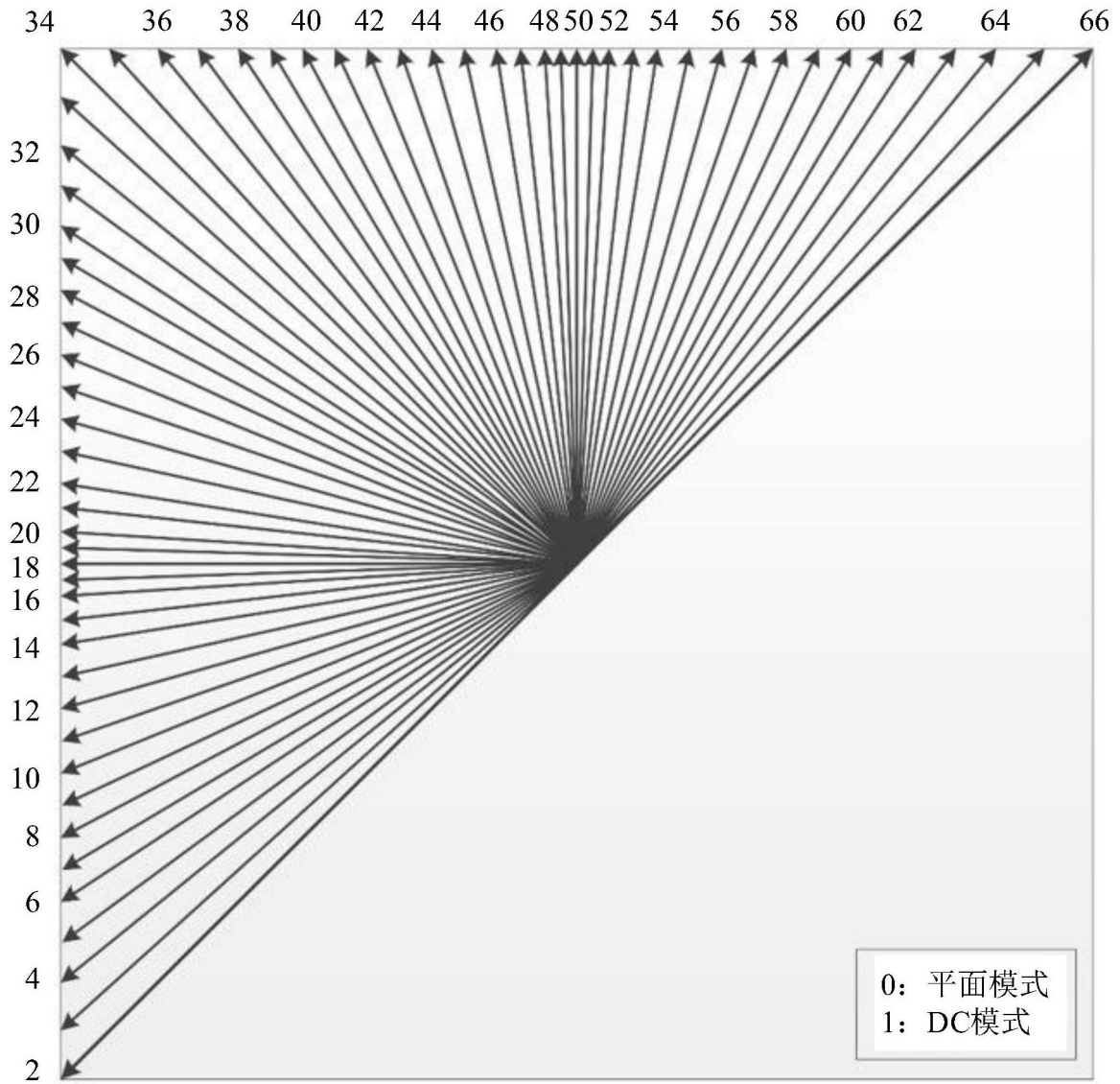


图10A

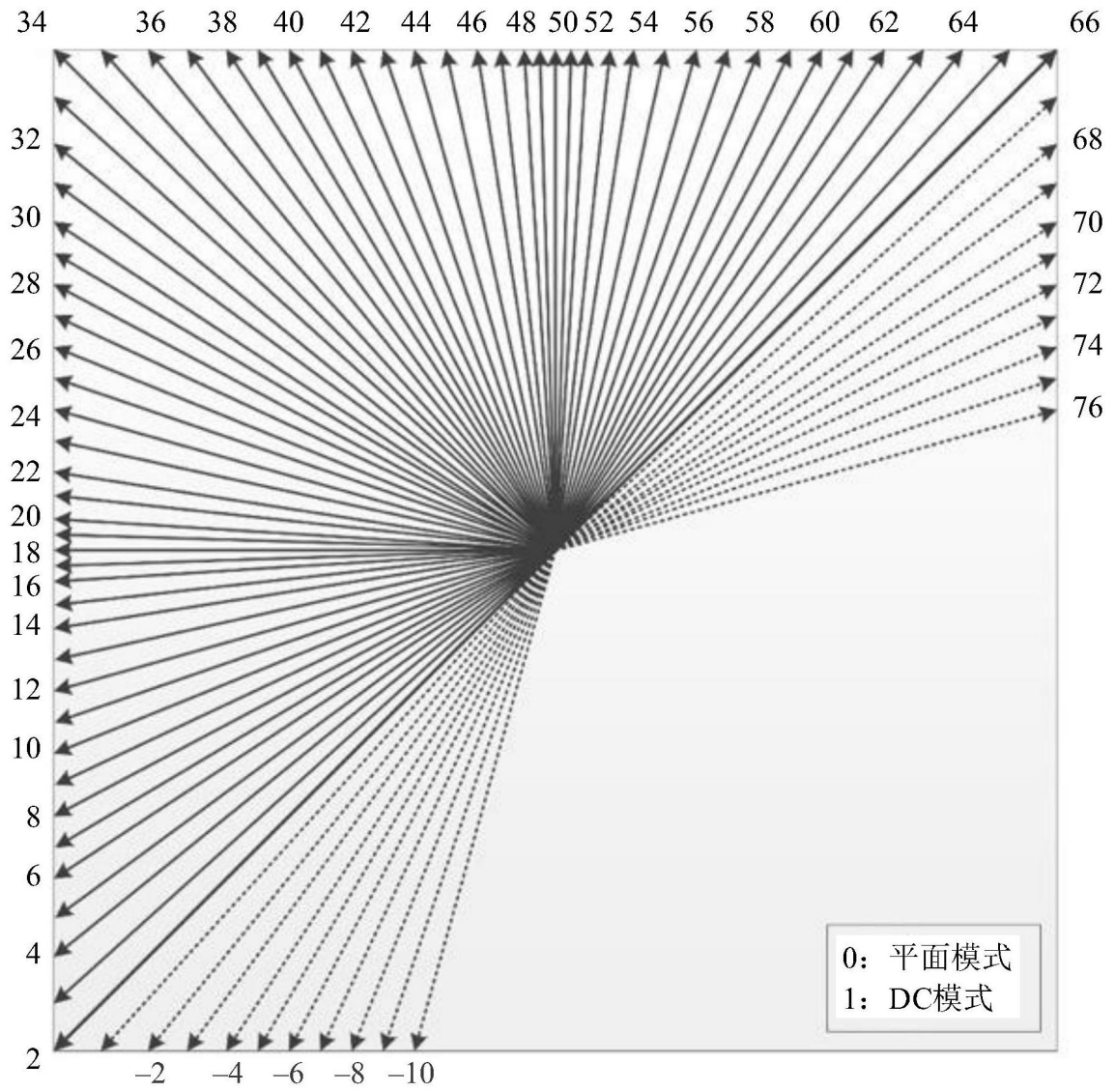


图10B

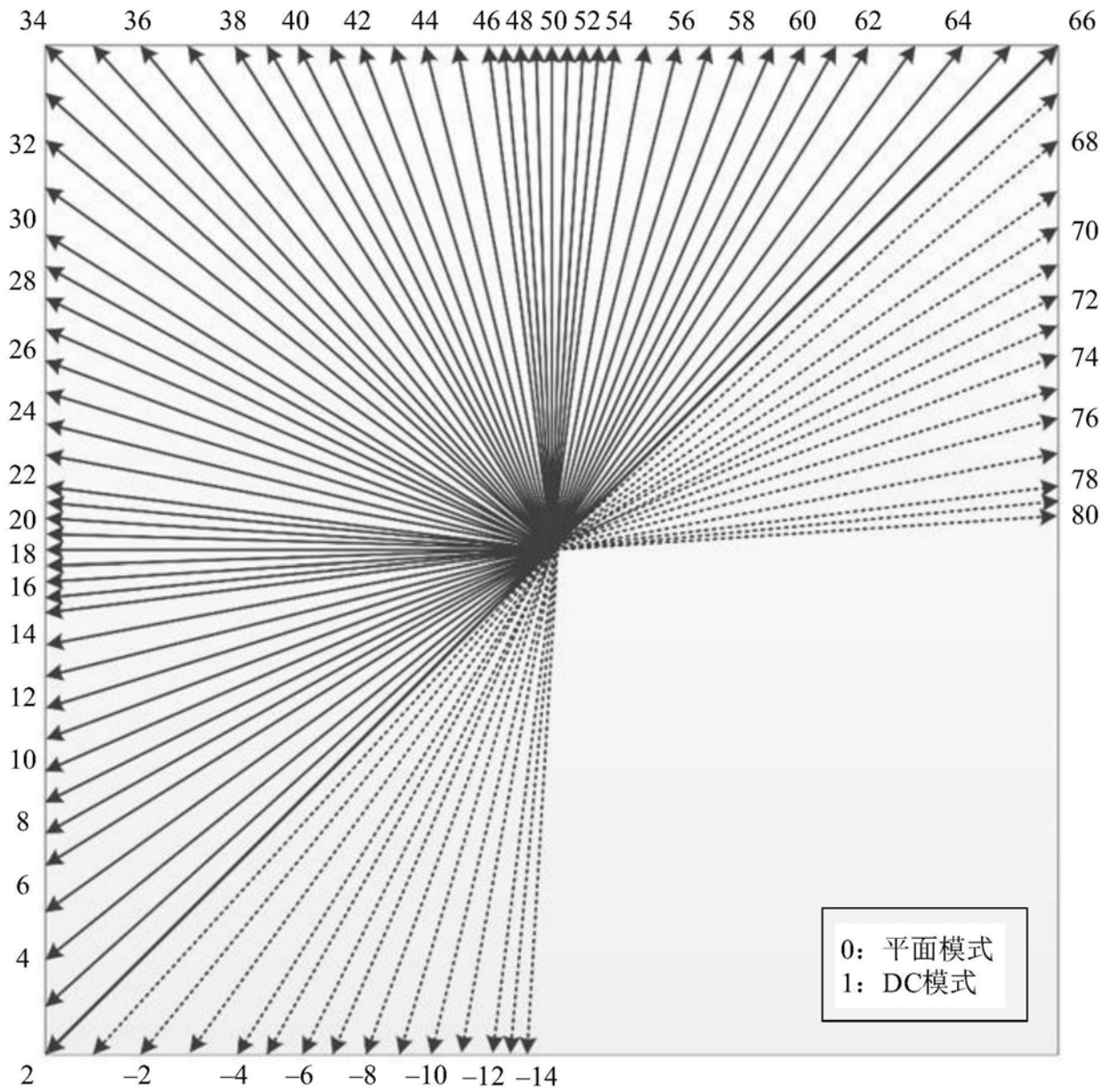


图10C

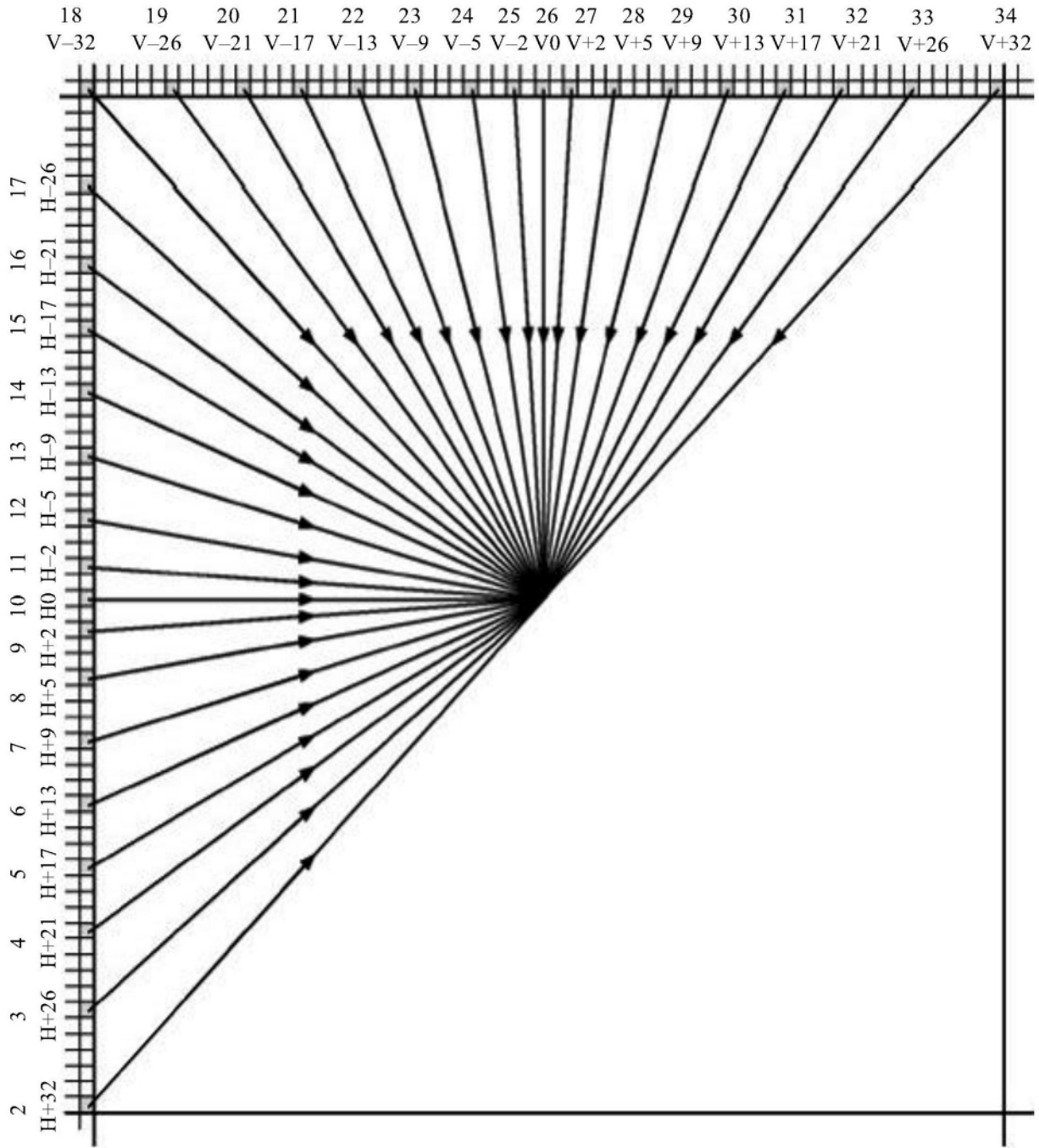
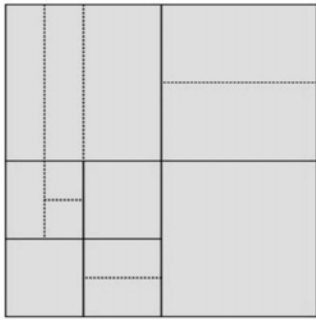


图11

空间分割



用于对空间分割进行译码的四叉树结合二叉树

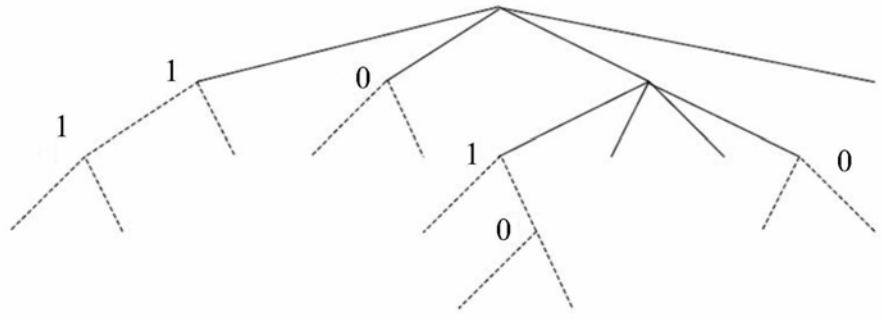
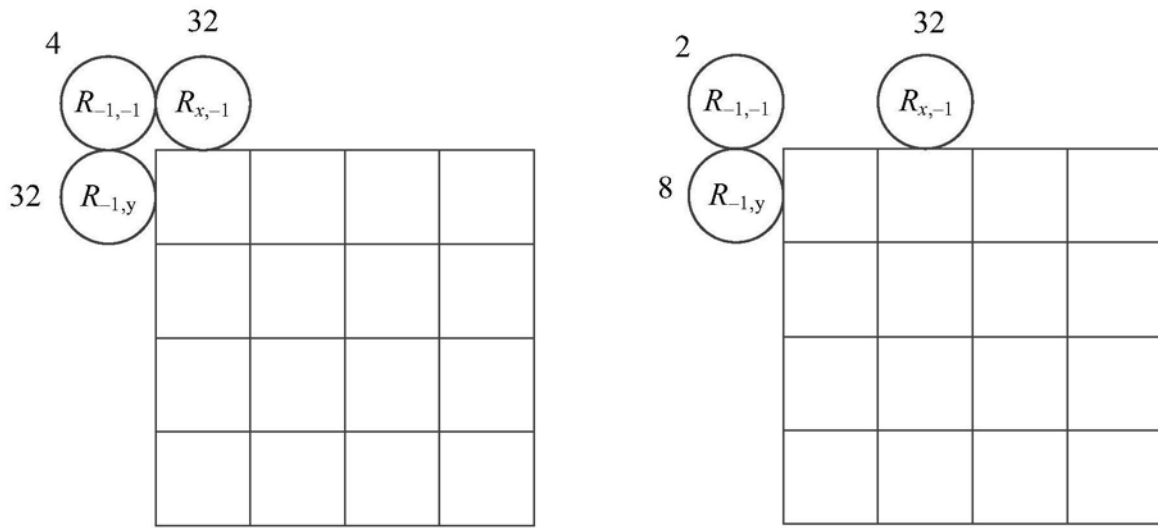
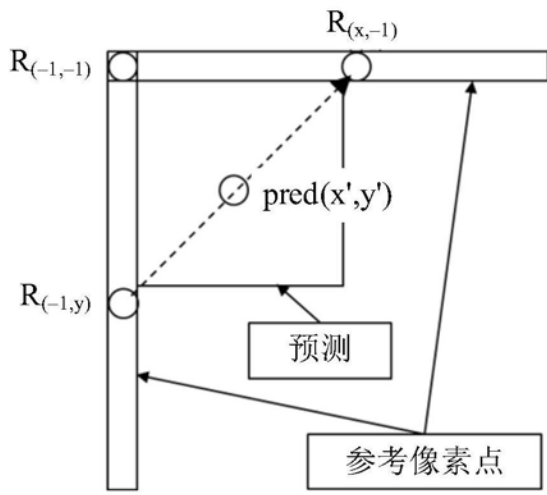


图12

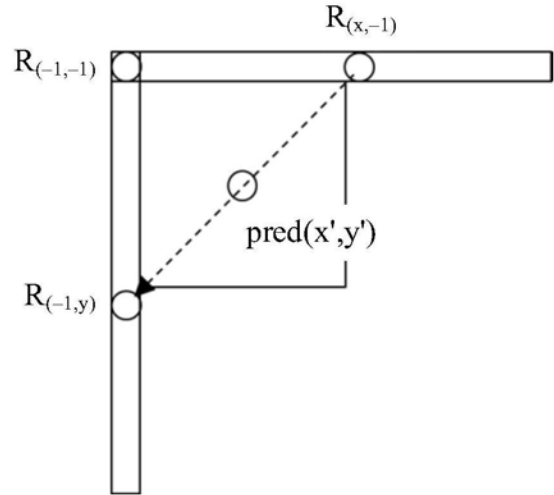


一个4×4块内的位置 (0, 0) 和 (1, 0) 的DC模式PDPC权重

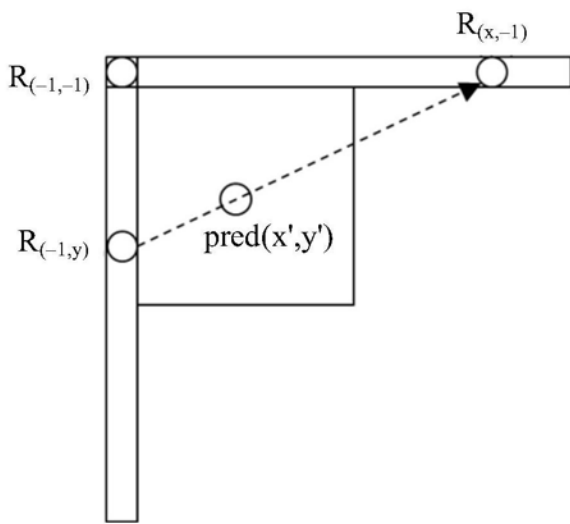
图13B



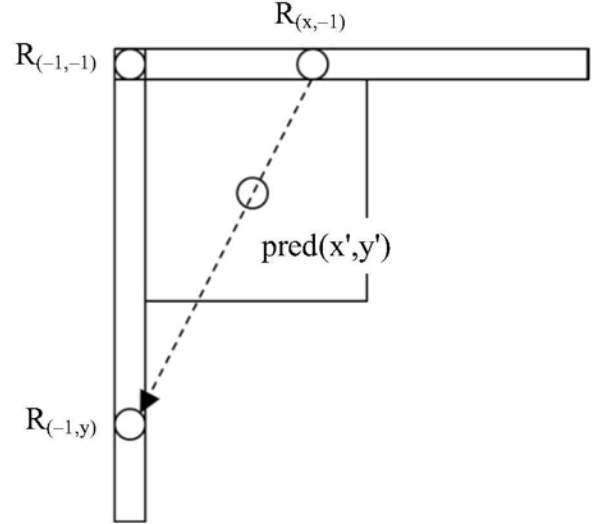
(A) 右上方对角线模式



(B) 左下方对角线模式



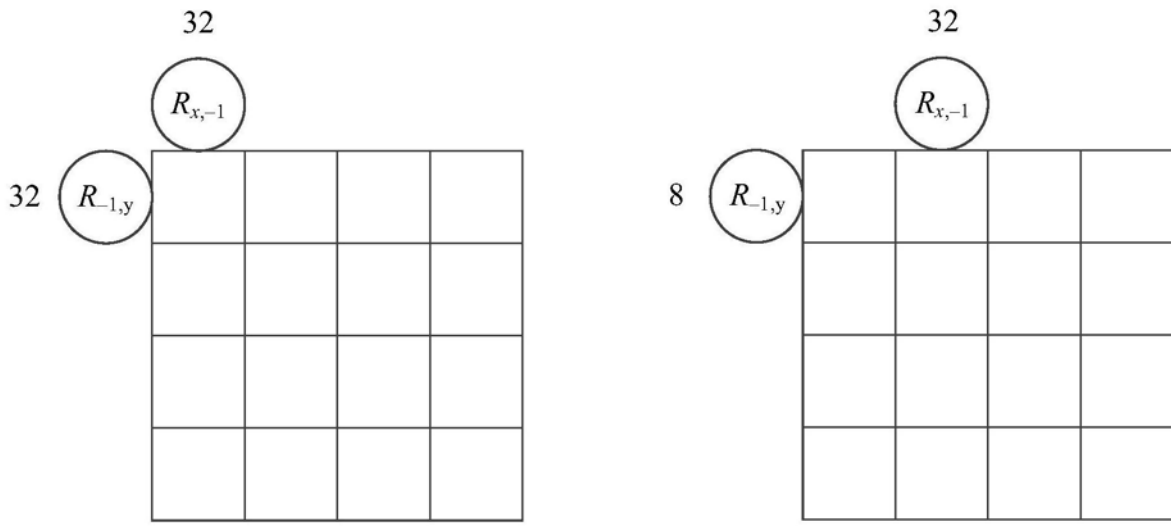
(C) 右上方相邻对角线模式



(D) 左下方相邻对角线模式

PDPC到对角线和相邻角度帧内模式扩展所使用的像素点的定义

图14



一个4×4块内的位置 (0, 0) 和 (1, 0) 的DC模式PDPC权重

图15

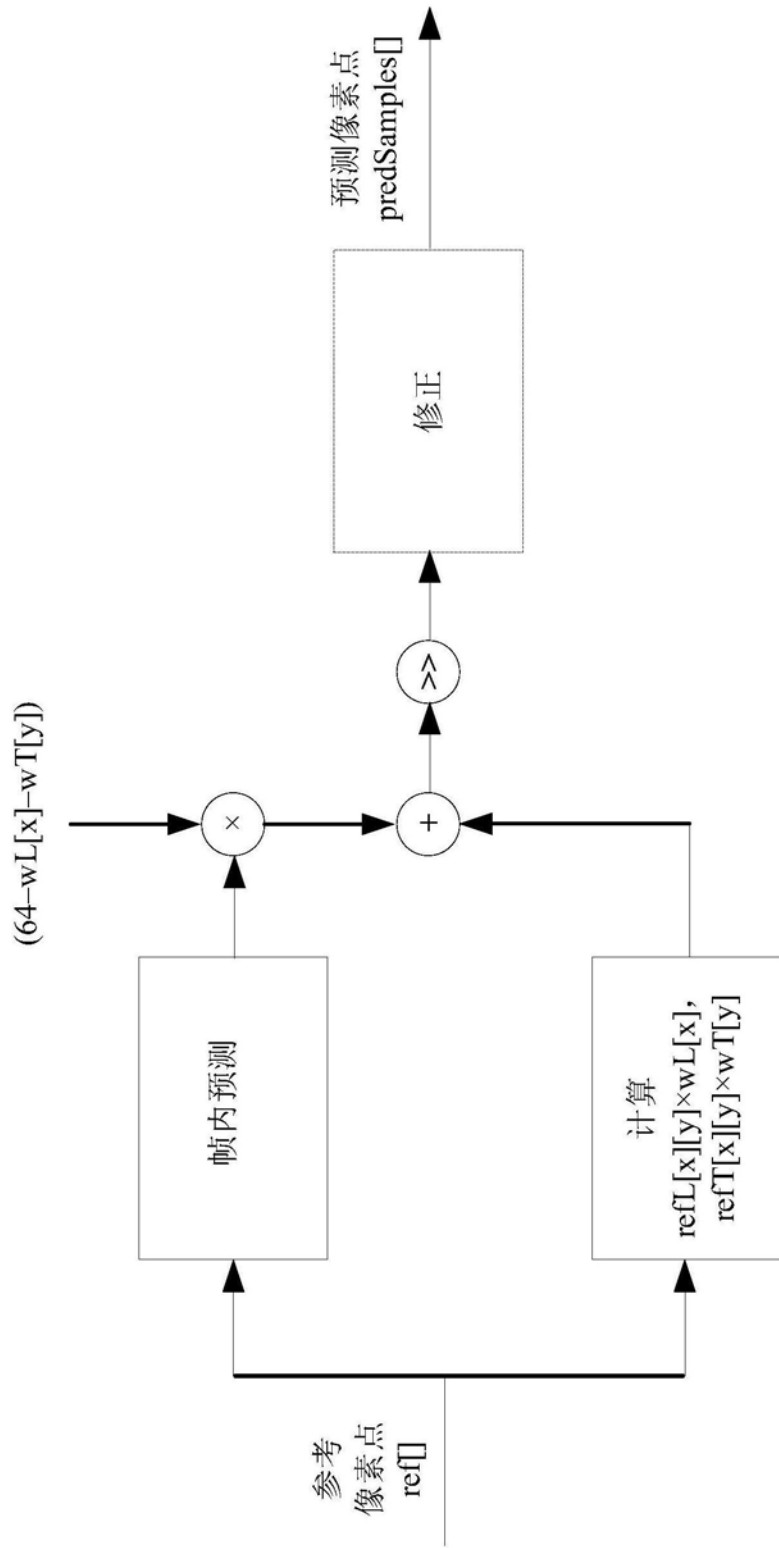


图16

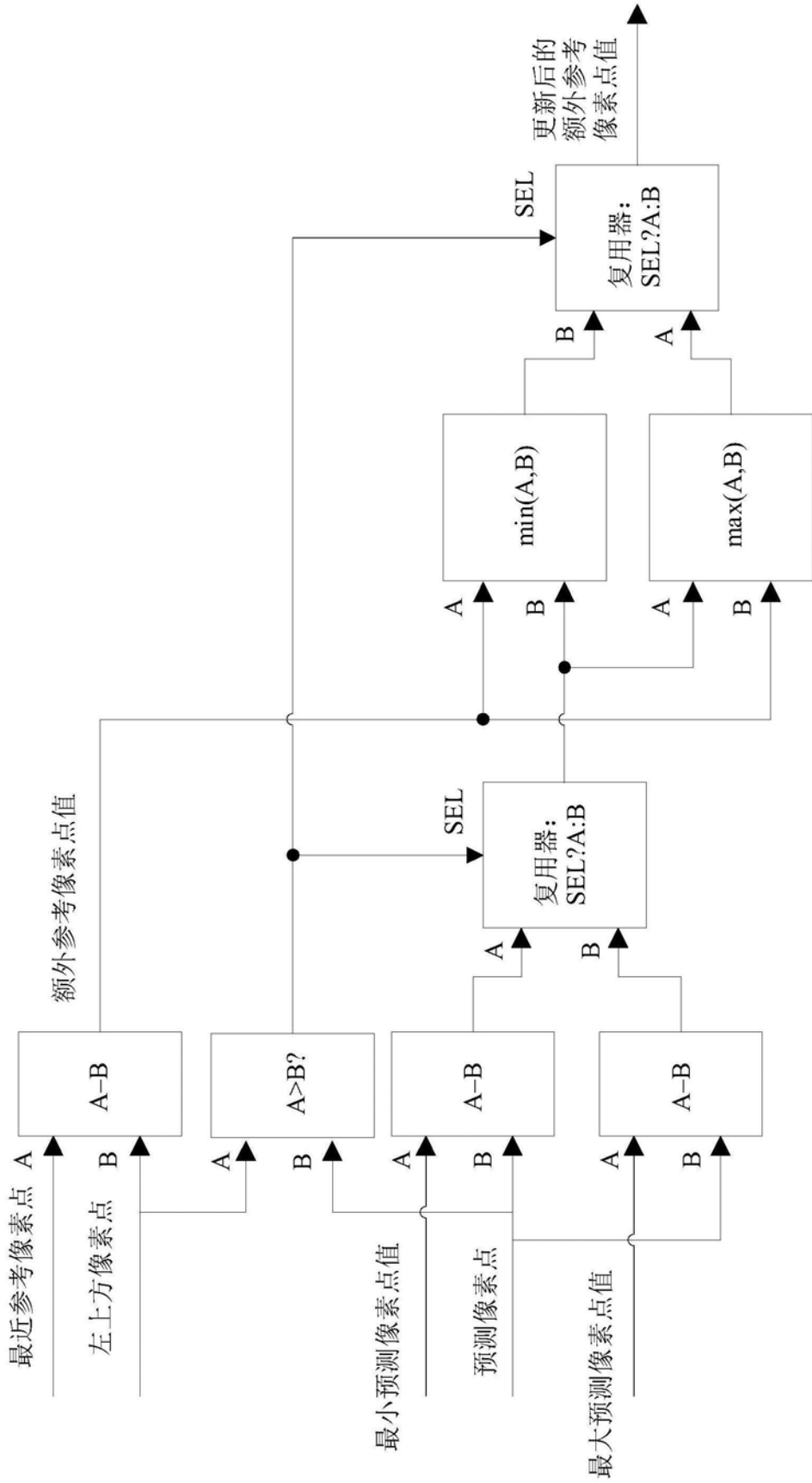


图17

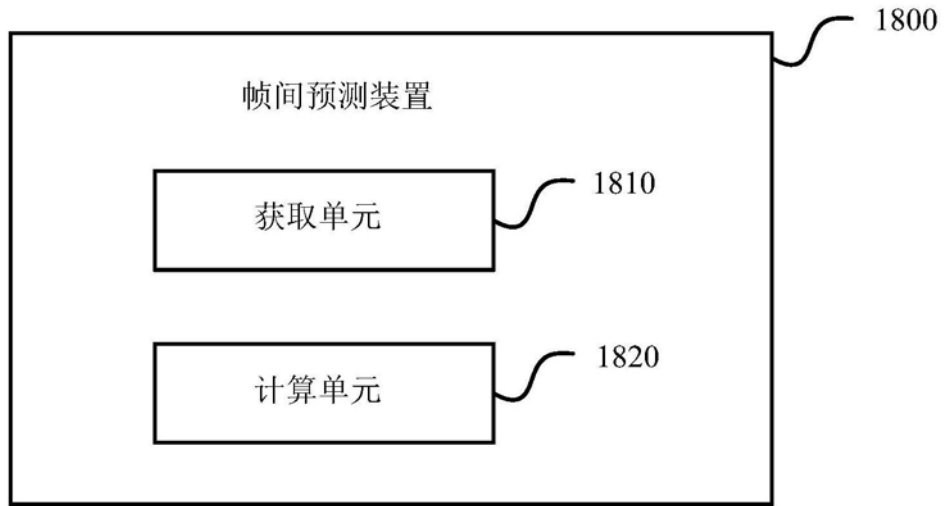


图18

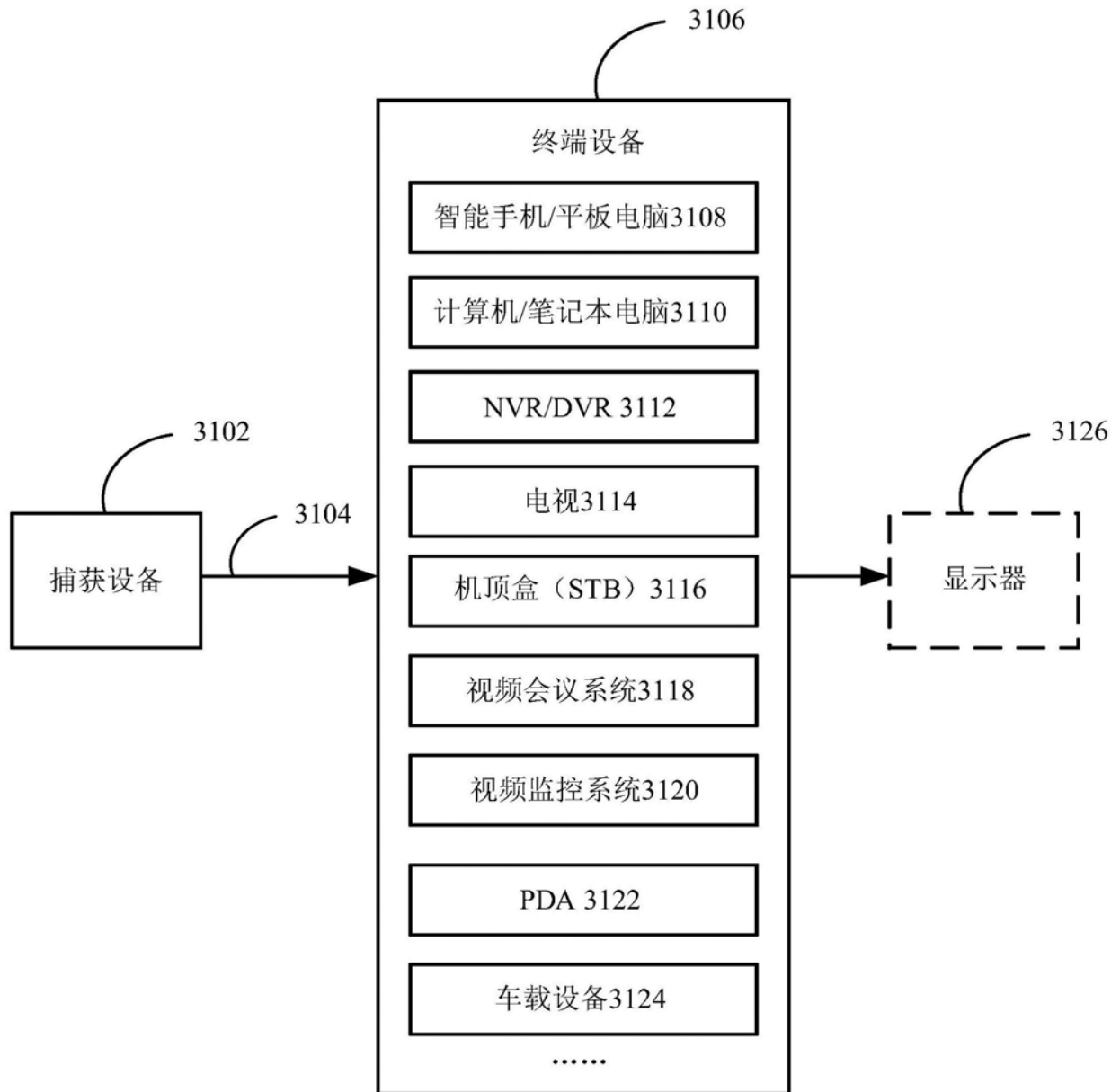


图19

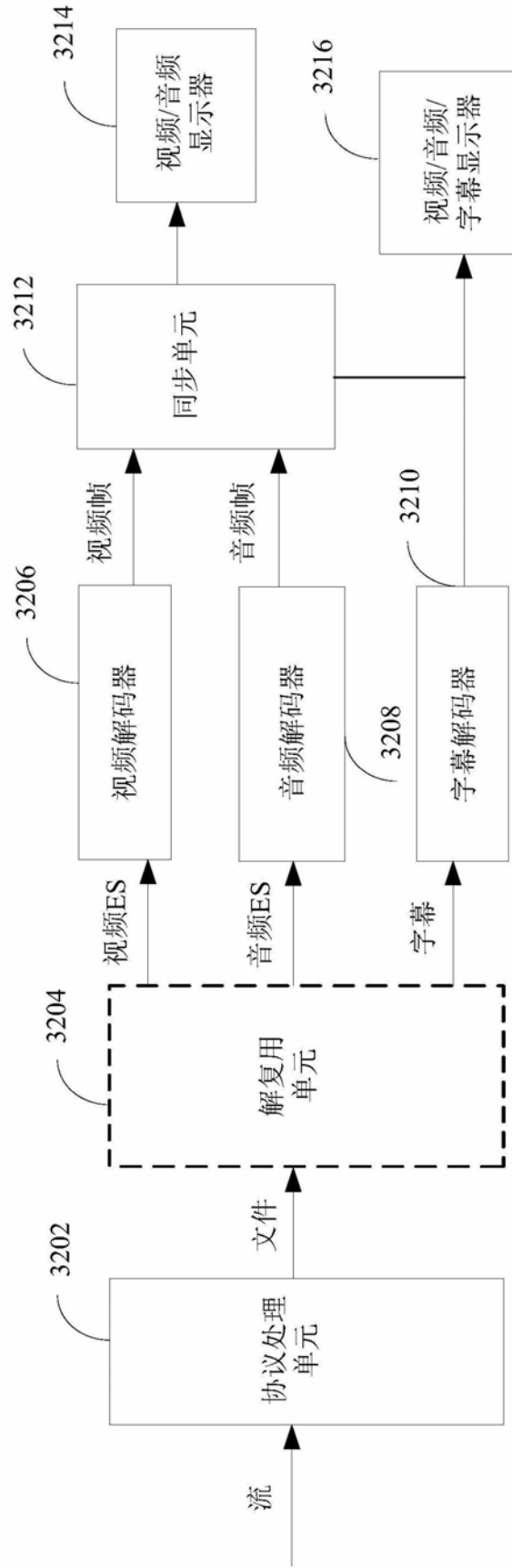


图20