



(12) 发明专利申请

(10) 申请公布号 CN 120051987 A

(43) 申请公布日 2025. 05. 27

(21) 申请号 202380072845.9

(22) 申请日 2023.10.12

(66) 本国优先权数据

PCT/CN2022/124841 2022.10.12 CN

(85) PCT国际申请进入国家阶段日

2025.04.14

(86) PCT国际申请的申请数据

PCT/CN2023/124196 2023.10.12

(87) PCT国际申请的公布数据

W02024/078566 EN 2024.04.18

(71) 申请人 抖音视界有限公司

地址 100041 北京市石景山区实兴大街30

号院3号楼2层B-0035房间

申请人 字节跳动有限公司

(72) 发明人 尹文斌 张凯 张莉

(74) 专利代理机构 北京市柳沈律师事务所

11105

专利代理师 张亮 刘文洁

(51) Int.Cl.

H04N 19/117 (2006.01)

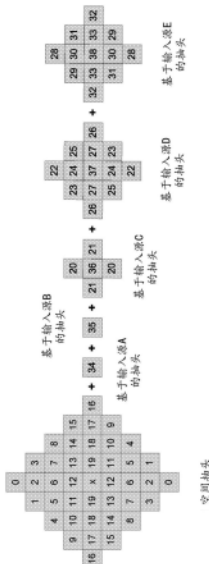
权利要求书3页 说明书38页 附图18页

(54) 发明名称

视频编解码中用于自适应环路滤波器的基于多个输入源的扩展抽头

(57) 摘要

公开了一种用于处理视频数据的机制。该机制包括确定将具有扩展抽头的自适应环路滤波器 (ALF) 应用于视频中的图片。第二滤波器的中间滤波结果用作扩展抽头的输入。基于该ALF执行可视媒体数据与比特流之间的转换。



1. 一种用于处理视频数据的方法,包括:
确定将具有扩展抽头的自适应环路滤波器(ALF)应用于视频中的图片,其中第二滤波器的中间滤波结果用作所述扩展抽头的输入;以及
基于所述ALF执行可视媒体数据与比特流之间的转换。
2. 如权利要求1所述的方法,其中ALF的经离线训练的滤波器、ALF的经在线训练的滤波器、预定义滤波器或其他经在线训练的滤波器的中间滤波结果用作所述扩展抽头的输入。
3. 如权利要求1至2中任一项所述的方法,其中所述中间滤波结果是通过所述ALF之前的重建和所述ALF的经离线训练的滤波器生成的。
4. 如权利要求1至3中任一项所述的方法,其中所述中间滤波结果是通过去方块滤波器DBF之前的重建和所述ALF的所述经离线训练的滤波器生成的。
5. 如权利要求1至4中任一项所述的方法,其中所述预定义滤波器包括高斯滤波器。
6. 如权利要求1至5中任一项所述的方法,其中所述预定义滤波器包括双边滤波器、导向滤波器、中值滤波器、具有低通属性的滤波器或具有高通属性的滤波器。
7. 如权利要求1至6中任一项所述的方法,其中ALF的经在线训练的滤波器的中间滤波结果用作所述扩展抽头的输入。
8. 如权利要求1至7中任一项所述的方法,其中用于生成所述中间滤波结果的输入包括不同编解码阶段处的重建样点。
9. 如权利要求1至8中任一项所述的方法,其中所述中间滤波结果是使用参考帧的ALF之前或之后的重建样点生成的。
10. 如权利要求1至9中任一项所述的方法,其中所述中间滤波结果是使用参考帧的去方块滤波器(DBF)之前或之后的重建样点生成的。
11. 如权利要求1至10中任一项所述的方法,其中所述中间滤波结果是使用参考帧的样点自适应偏移(SAO)或跨分量SAO(CCSAO)之前或之后的重建样点、或参考帧的双边滤波器(BIF)之前或之后的重建样点生成的。
12. 如权利要求1至11中任一项所述的方法,其中当前帧的不同编解码阶段处的重建样点用作所述扩展抽头的输入。
13. 如权利要求1至12中任一项所述的方法,其中所述当前帧的DBF之前或之后的重建样点用作所述扩展抽头的输入。
14. 如权利要求1至13中任一项所述的方法,其中所述当前帧的ALF之前的重建样点用作所述扩展抽头的输入,或者其中所述当前帧的SAO或CCSAO之前或之后的重建样点用作所述扩展抽头的输入,或者其中所述当前帧的BIF之前或之后的重建样点用作所述扩展抽头的输入。
15. 如权利要求1至14中任一项所述的方法,其中编解码图片内部的样点用作所述扩展抽头的输入源。
16. 如权利要求1至15中任一项所述的方法,其中与当前块、当前条带或当前帧相关联的参考图片列表(RPL)或参考图片集(RPS)中的参考帧内部的样点用作所述扩展抽头的输入源。
17. 如权利要求1至16中任一项所述的方法,其中所述参考帧是当前图片、当前条带或当前块的长期参考图片或短期参考图片。

18. 如权利要求1至17中任一项所述的方法,其中解码图片缓存区中的帧内部的样点用作所述扩展抽头的输入源。

19. 如权利要求1至18中任一项所述的方法,其中通过信号传输指示符以指示包含用作所述扩展抽头的输入源的样点的编解码图片。

20. 如权利要求1至19中任一项所述的方法,其中所述指示符指示参考图片列表或参考索引,并且其中根据参考图片列表中的参考图片的数量或解码图片缓存区中的解码图片的数量有条件地通过信号传输所述指示符。

21. 如权利要求1至20中任一项所述的方法,其中编解码图片包含用作所述扩展抽头的输入源的样点,并且其中所述编解码图片是被动态确定的。

22. 如权利要求1至21中任一项所述的方法,其中所述扩展抽头从解码图片缓存区中的帧、参考图片列表0中的帧、参考图片列表1中的帧、最接近当前帧的参考帧、具有特定索引的参考帧、同位帧或以上帧的组合接收输入。

23. 如权利要求1至22中任一项所述的方法,其中基于解码信息来确定用作所述扩展抽头的输入的帧。

24. 如权利要求1至23中任一项所述的方法,其中是否从先前编解码的帧中获取信息以用作所述扩展抽头的输入源取决于待滤波块的至少一个区域的解码信息。

25. 如权利要求1至24中任一项所述的方法,其中语法元素指示是否从先前编解码的帧中获取信息以用作所述扩展抽头的输入源。

26. 如权利要求1至25中任一项所述的方法,其中是否从先前编解码的帧中获取信息以用作所述扩展抽头的输入源取决于条带类型或图片类型。

27. 如权利要求1至26中任一项所述的方法,其中从先前编解码的帧中获取信息以用作所述扩展抽头的输入源仅适用于帧间编解码的条带或图片。

28. 如权利要求1至27中任一项所述的方法,其中是否从先前编解码的帧中获取信息以用作所述扩展抽头的输入源取决于参考图片信息或解码图片缓存区中的图片信息。

29. 如权利要求1至28中任一项所述的方法,其中是否从先前编解码的帧中获取信息以用作所述扩展抽头的输入源取决于时间层索引、量化参数或图片的尺寸。

30. 如权利要求1至29中任一项所述的方法,其中当块包含以非帧间模式编解码的样点时,所述扩展抽头不使用来自先前编解码的帧的信息来对所述块进行滤波。

31. 如权利要求1至30中任一项所述的方法,其中当前块与匹配块之间的失真用于确定是否使用来自先前编解码的帧的信息来对所述当前块进行滤波。

32. 如权利要求1至31中任一项所述的方法,其中所述信息包含当前块的两个参考块或同位块,其中一个块来自列表-0中的第一参考帧,并且另一个块来自列表-1中的第一参考帧。

33. 如权利要求1至32中任一项所述的方法,进一步包括:确定在环路滤波器中应用所述扩展抽头,以及基于所述环路滤波器执行所述可视媒体数据与所述比特流之间的转换。

34. 如权利要求1至33中任一项所述的方法,其中所述环路滤波器包括跨分量ALF (CCALF) 滤波器。

35. 如权利要求1至34中任一项所述的方法,其中所述转换包括从所述比特流解码所述可视媒体数据。

36. 如权利要求1至34中任一项所述的方法,其中所述转换包括将所述可视媒体数据编码为所述比特流。

37. 一种用于处理视频数据的装置,包括:处理器;以及上面具有指令的非暂时性存储器,其中所述指令在由所述处理器执行时使所述处理器执行如权利要求1至36中任一项所述的方法。

38. 一种非暂时性计算机可读介质,包括供视频编解码设备使用的计算机程序产品,所述计算机程序产品包括存储在所述非暂时性计算机可读介质上的计算机可执行指令,使得在由处理器执行时,使所述视频编解码设备执行如权利要求1至36中任一项所述的方法。

39. 一种非暂时性计算机可读记录介质,其存储视频的比特流,该视频的比特流由通过视频处理装置执行的方法生成,其中所述方法包括:

确定将自适应环路滤波器(ALF)中的扩展抽头应用于所述视频中的图片,其中第二滤波器的中间滤波结果用作所述扩展抽头的输入;以及

基于所述确定生成所述比特流。

40. 一种用于存储视频的比特流的方法,包括:

确定将自适应环路滤波器(ALF)中的扩展抽头应用于所述视频中的图片,其中第二滤波器的中间滤波结果用作所述扩展抽头的输入;

基于所述确定生成所述比特流;以及

将所述比特流存储在非暂时性计算机可读记录介质中。

视频编解码中用于自适应环路滤波器的基于多个输入源的扩展抽头

[0001] 相关申请的交叉引用

[0002] 本申请要求于2022年10月12日提交的国际专利申请号PCT/CN2022/124841的优先权和权益。前述专利申请的内容以引用方式整体并入本文中。

技术领域

[0003] 本公开涉及文件格式的数字音频视频媒体信息的生成、存储和消费。

背景技术

[0004] 数字视频占在互联网和其他数字通信网络上使用的最大的带宽。随着能够接收和显示视频的连接用户设备的数量的增加,数字视频使用的带宽需求有可能继续增长。

发明内容

[0005] 第一方面涉及一种用于处理视频数据的方法,其包括:确定将具有扩展抽头的自适应环路滤波器(ALF)应用于视频中的图片,其中第二滤波器的中间滤波结果用作该扩展抽头的输入;以及基于该ALF执行可视媒体数据与比特流之间的转换。

[0006] 第二方面涉及一种用于处理视频数据的装置,其包括:处理器;以及上面具有指令的非暂时性存储器,其中这些指令在由处理器执行时使处理器执行前述任一方面。

[0007] 第三方面涉及一种非暂时性计算机可读介质,其包括供视频编解码设备使用的计算机程序产品,该计算机程序产品包括存储在非暂时性计算机可读介质上的计算机可执行指令,使得在由处理器执行时使视频编解码设备执行前述任一方面的方法。

[0008] 第四方面涉及一种非暂时性计算机可读记录介质,其存储由通过视频处理装置执行的方法生成的视频的比特流,其中该方法包括:确定将具有扩展抽头的自适应循环滤波器(ALF)应用于视频中的图片,其中第二滤波器的中间滤波结果用作该扩展抽头的输入;以及基于该确定生成该比特流。

[0009] 第五方面涉及一种用于存储视频的比特流的方法,其包括:确定将具有扩展抽头的自适应环路滤波器(ALF)应用于视频中的图片,其中第二滤波器的中间滤波结果用作该扩展抽头的输入;基于该确定生成该比特流;以及将比特流存储在非暂时性计算机可读记录介质中。

[0010] 第六方面涉及一种在本专利文档中描述的方法、装置或系统。

[0011] 为了清楚起见,前述实施例中的任一者都可与其他前述实施例中的一者或多者相结合以在本公开的范围内产生新的实施例。

[0012] 根据下面结合附图和权利要求的详细描述,可更清楚地理解这些和其他特征。

附图说明

[0013] 为了更全面地理解本公开,现在参考结合附图和具体实施方式的以下简要描述,

其中相同的附图标记表示相同部件。

- [0014] 图1示出了图片中的4:2:2亮度和色度样点的标称垂直和水平位置的示例。
- [0015] 图2示出了示例编码器框图。
- [0016] 图3示出了被分割成光栅扫描条带的示例图片。
- [0017] 图4示出了被分割成矩形扫描条带的示例图片。
- [0018] 图5示出了分区成砖块(brick)的示例图片。
- [0019] 图6示出了跨图片边界的编解码树块(CTB)的示例。
- [0020] 图7示出了帧内预测模式的示例。
- [0021] 图8示出了图片中的块边界的示例。
- [0022] 图9示出了滤波器使用中所涉及的像素的示例。
- [0023] 图10示出了ALF的滤波器形状的示例。
- [0024] 图11示出了 5×5 菱形滤波器支持的变换系数的示例。
- [0025] 图12示出了 5×5 菱形滤波器支持的相对坐标的示例。
- [0026] 图13和图14示出了空域抽头的示例形状。
- [0027] 图15至图18示出了具有扩展抽头的示例ALF滤波器。
- [0028] 图19是示出示例视频处理系统的框图。
- [0029] 图20是示例视频处理装置的框图。
- [0030] 图21是视频处理的示例方法的流程图。
- [0031] 图22是示出示例视频编解码系统的框图。
- [0032] 图23是示出示例编码器的框图。
- [0033] 图24是示出示例解码器的框图。
- [0034] 图25是示例编码器的示意图。

具体实施方式

[0035] 首先应当理解,尽管下文提供了一个或多个实施例的说明性实施方式,但是可使用任何数量的技术实施所公开的系统 and/或方法,无论是当前已知的还是有待开发的。本公开不应以任何方式限于以下示出的示例性实施方式、附图和技术,包括本文示出和描述的示例性设计和实施方式,而是可以在所附权利要求及其等同物的全部范围内进行修改。

[0036] 本文档中使用的章节标题是为了便于理解,并且不限制在每个章节中公开的技术和实施例仅对该章节的适用性。此外,本文中描述的技术适用于其他视频编解码器协议和设计。

[0037] 1. 初始讨论

[0038] 本文档涉及视频编解码技术。具体地,它涉及图像/视频编解码中的环路滤波器和其他编解码工具。这些想法可以单独或以各种组合应用于视频编解码器,诸如高效视频编解码(HEVC)、多功能视频编解码(VVC)或其他视频编解码技术。

[0039] 2. 缩写

[0040] 本公开包括以下缩写。高级视频编解码(Rec. ITU-T H.264|ISO/IEC 14496-10)(AVC)、编解码图像缓存区(CPB)、纯随机接入(CRA)、编解码树单元(CTU)、编解码视频序列(CVS)、解码图片缓存区(DPB)、解码参数集(DPS)、通用约束信息(GCI)、国际标准化组织

(ISO)、国际电工委员会(IEC)、高效视频编解码(也称为Rec. ITU-T H.265|ISO/IEC 23008-2, (HEVC))、联合探索模型(JEM)、运动约束片集(MCTS)、网络抽象层(NAL)、输出层集(OLS)、图片头(PH)、图片参数集(PPS)、档次、层和级别(PTL)、图片单元(PU)、参考图片重采样(RPR)、原始字节序列有效负载(RBSP)、补充增强信息(SEI)、条带头(SH)、序列参数集(PPS)、视频编解码层(VCL)、视频参数集(VPS)、多功能视频编解码(也称为Rec. ITU-T H.266|ISO/IEC 23090-3) (VVC)、VVC测试模型(VTM)、视频可用性信息(VUI)、变换单元(TU)、编解码单元(CU)、去方块滤波器(DF)、样点自适应偏移(SAO)、自适应环路滤波器(ALF)、编解码块标志(CBF)、量化参数(QP)、率失真优化(RDO)和双边滤波器(BF)。

[0041] 3. 视频编解码标准

[0042] 视频编解码标准主要通过国际电信联盟-电信标准化部门(ITU-T)和ISO/IEC标准的发展而演变。ITU-T制定了H.261和H.263, ISO/IEC制定了运动图片专家组(MPEG-1)和MPEG-4视觉,并且这两个组织联合制定了H.262/MPEG-2视频和H.264/MPEG-4高级视频编解码(AVC)和H.265/HEVC标准。自H.262起,视频编解码标准是基于混合视频编解码结构,其中利用了时域预测加变换编解码。为了探索HEVC之外的未来视频编解码技术,视频编解码专家组(VCEG)和MPEG联合成立了联合视频探索团队(JVET)。JVET采纳了许多方法,并将其置于名为联合探索模型(JEM)的参考软件中。在通用视频编解码(VVC)项目正式启动时,联合视频探索团队(JVET)更名为联合视频专家组(JVET)。VVC是一种编码标准,其目标是与HEVC相比将比特率降低50%。VVC工作草案和VVC测试模型(VTM)被不断地更新。

[0043] VVC草案的示例版本(即,通用视频编解码(草案10))可在以下网址找到:https://jvet-experts.org/doc_end_user/documents/19_Teleconference/wg11/JVET-S2001-v17.zip。VVC的参考软件的示例版本(名为VTM)可在以下网址找到:https://vcgit.hhi.fraunhofer.de/jvet-u-ee2/VVCSoftware_VTM/-/tree/VTM-11.2。

[0044] ITU-T VCEG和ISO/IEC MPEG联合技术委员会(JTC)1/小组委员会(SC)29/工作组(WG)11正在研究对压缩能力将大幅超过当前的VVC标准的未来视频编解码技术标准化的潜在需求。这种未来的标准化行动可采取VVC的扩展的扩展或全新标准的形式。这些小组在被称为JVET的联合协同工作尝试中共同开展这项开发活动,以评估其专家在该领域提出的压缩技术设计。第一次探索实验(EI)由JVET建立,并且名为增强压缩模型(ECM)的参考软件正在使用中。测试模型ECM被不断地更新。

[0045] 3.1 颜色空间和色度下采样

[0046] 颜色空间(也称为颜色模型(或颜色系统))是一种数学模型,它将颜色范围描述为数字元组,例如作为3或4个值或颜色分量(例如,RGB)。一般来说,颜色空间是坐标系和子空间的具体化形式。对于视频压缩,最常使用的颜色空间是亮度、蓝色差色度以及红色差色度(YCbCr)和红、绿、蓝(RGB)。

[0047] YCbCr、Y'CbCr或Y Pb/Cb Pr/Cr(也写作YCBCR或Y'CBCR)是一系列色彩空间的总称,其作为彩色图像处理流程的一部分,应用于视频系统和数字摄影领域。Y'是亮度分量,并且Cb和Cr是蓝色差色度分量和红色差色度分量。Y'(带撇号)与Y不同,这意味着光强度是基于经伽马校正的RGB原色进行非线性编码的。

[0048] 色度下采样是利用人类视觉系统对色差的敏锐度低于对亮度的敏锐度,通过对色度信息实现比亮度信息更低的分辨率来对图像进行编码的做法。

[0049] 3.1.1.4:4:4

[0050] 在4:4:4中,Y'CbCr三个分量中的每一者具有相同的采样率。因此没有色度下采样。这种方案有时用于高端胶片扫描仪和电影后期制作。

[0051] 3.1.2.4:2:2

[0052] 在4:2:2中,两个色度分量以亮度的采样率的一半被采样。水平色度分辨率减半,而垂直色度分辨率不变。这将未压缩视频信号的带宽减少了三分之一,但几乎没有视觉差异。图1中示出了图片中的4:2:2颜色格式下的标称垂直和水平位置的示例。

[0053] 3.1.3 4:2:0

[0054] 在4:2:0中,与4:1:1相比,水平采样提高了一倍,但是由于在这种方案中Cb和Cr通道仅在交替的行上采样,因此垂直分辨率减半,数据率因此保持不变。Cb和Cr各自在水平和垂直两个方向上均以因数2进行下采样。4:2:0方案有三种变体,每个变体都具有不同的水平和垂直采样位置。在MPEG-2中,Cb和Cr在水平方向上是同位对齐的。Cb和Cr在垂直方向上位于像素之间(即,交错排列)。在JPEG/JFIF、H.261和MPEG-1中,Cb和Cr以交错方式排列,位于交替亮度样点之间的中间位置。在4:2:0DV中,Cb和Cr在水平方向上同位对齐。在垂直方向上,它们在交替行上同位对齐。

[0055]	chroma_format_idc	separate_colour_plane_flag	色度格式	SubWidthC	SubHeightC
	0	0	单色	1	1
[0056]	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

[0057] 表1.从chroma_format_idc和Separate_colour_plane_flag推导的SubWidthC和SubHeightC值

[0058] 3.2视频编解码器的示例编解码流程

[0059] 图2示出了VVC的编码器框图的示例,其包含三个环路滤波模块:去方块滤波器(DF)、样点自适应偏移(SAO)和ALF。与使用预定义滤波器的DF不同,SAO和ALF利用当前图片的原始样点以通过分别添加偏移和通过应用有限脉冲响应(FIR)滤波器来减小原始样点与重建样点之间的均方误差,编解码的边信息(side information)用于通过信号传输该偏移和滤波器系数。ALF位于每张图片的最后处理阶段,并且可以被视为试图捕捉和修复由先前阶段产生的伪影的工具。

[0060] 3.3视频/编解码单元的定义

[0061] 图片被划分为一个或多个片行和一个或多个片列。片是覆盖图片的矩形区域的CTU序列。片可以被划分为一个或多个砖块,每个砖块包括片内的一定数量个CTU行。没有被分割为多个砖块的一个片也可以被称为一个砖块。然而,作为片的真子集的砖块不能称为片。条带包含一个图片的若干个片或一个片的若干个砖块。

[0062] 支持两种条带模式,即,光栅扫描条带模式和矩形条带模式。在光栅扫描条带模式下,条带在图片的片光栅扫描中包含一系列片。在矩形条带模式中,条带包含图片的多个砖

块,这些砖块共同形成图片的矩形区域。矩形条带内的砖块按照条带的砖块光栅扫描的顺序排列。图3示出了(具有18乘12个亮度CTU的)图片的光栅扫描条带分割的示例,其中图片被划分为12个片和3个光栅扫描条带。

[0063] 图4示出了(具有18乘12个亮度CTU的)图片的矩形条带分割的示例,其中图片被划分为24个片(6个片列和4个片行)和9个矩形条带。

[0064] 图5示出了图片被分割成片、砖块和矩形条带的示例,其中图片被划分为4个片(2个片列和2个片行)、11个砖块(左上角片包含1个砖块,右上角片包含5个砖块,左下角片包含2个砖块,并且右下角片包含3个砖块)和4个矩形条带。

[0065] 3.3.1CTU/CTB尺寸

[0066] 在VVC中,CTU尺寸(通过语法元素log2_ctu_size_minus2在序列参数集(SPS)中通过信号传输)可以小至 4×4 。

[0067] 7.3.2.3序列参数集RBSP语法

[0068]	seq_parameter_set_rbsp() {	描述符
--------	----------------------------	-----

[0069]

sps_decoding_parameter_set_id	u(4)
sps_video_parameter_set_id	u(4)
sps_max_sub_layers_minus1	u(3)
sps_reserved_zero_5bits	u(5)
profile_tier_level(sps_max_sub_layers_minus1)	
gra_enabled_flag	u(1)
sps_seq_parameter_set_id	ue(v)
chroma_format_idc	ue(v)
if(chroma_format_idc == 3)	
separate_colour_plane_flag	u(1)
pic_width_in_luma_samples	ue(v)
pic_height_in_luma_samples	ue(v)
conformance_window_flag	u(1)
if(conformance_window_flag) {	
conf_win_left_offset	ue(v)
conf_win_right_offset	ue(v)
conf_win_top_offset	ue(v)
conf_win_bottom_offset	ue(v)
}	
bit_depth_luma_minus8	ue(v)
bit_depth_chroma_minus8	ue(v)
log2_max_pic_order_cnt_lsb_minus4	ue(v)
sps_sub_layer_ordering_info_present_flag	u(1)
for(i = (sps_sub_layer_ordering_info_present_flag ? 0 : sps_max_sub_layers_minus1); i <= sps_max_sub_layers_minus1; i++) {	
sps_max_dec_pic_buffering_minus1[i]	ue(v)
sps_max_num_reorder_pics[i]	ue(v)
sps_max_latency_increase_plus1[i]	ue(v)
}	
long_term_ref_pics_flag	u(1)
sps_idr_rpl_present_flag	u(1)
rpl1_same_as_rpl0_flag	u(1)

[0070]

for(i = 0; i < !rpl1_same_as_rpl0_flag ? 2 : 1; i++) {	
num_ref_pic_lists_in_sps[i]	ue(v)
for(j = 0; j < num_ref_pic_lists_in_sps[i]; j++)	
ref_pic_list_struct(i, j)	
}	
qtbtt_dual_tree_intra_flag	u(1)
log2_ctu_size_minus2	ue(v)
log2_min_luma_coding_block_size_minus2	ue(v)
partition_constraints_override_enabled_flag	u(1)
sps_log2_diff_min_qt_min_cb_intra_slice_luma	ue(v)
sps_log2_diff_min_qt_min_cb_inter_slice	ue(v)
sps_max_mtt_hierarchy_depth_inter_slice	ue(v)
sps_max_mtt_hierarchy_depth_intra_slice_luma	ue(v)
if(sps_max_mtt_hierarchy_depth_intra_slice_luma != 0) {	
sps_log2_diff_max_bt_min_qt_intra_slice_luma	ue(v)
sps_log2_diff_max_tt_min_qt_intra_slice_luma	ue(v)
}	
if(sps_max_mtt_hierarchy_depth_inter_slices != 0) {	
sps_log2_diff_max_bt_min_qt_inter_slice	ue(v)
sps_log2_diff_max_tt_min_qt_inter_slice	ue(v)
}	
if(qtbtt_dual_tree_intra_flag) {	
sps_log2_diff_min_qt_min_cb_intra_slice_chroma	ue(v)
sps_max_mtt_hierarchy_depth_intra_slice_chroma	ue(v)
if (sps_max_mtt_hierarchy_depth_intra_slice_chroma !=	
0) {	
sps_log2_diff_max_bt_min_qt_intra_slice_chroma	ue(v)
sps_log2_diff_max_tt_min_qt_intra_slice_chroma	ue(v)
}	
}	
...	
rbsp_trailing_bits()	
}	

[0071] log2_ctu_size_minus2加2规定每个CTU的亮度编解码树块尺寸。log2_min_luma_coding_block_size_minus2加2规定最小亮度编解码块尺寸。变量CtbLog2SizeY、

CtbSizeY、MinCbLog2SizeY、MinCbSizeY、MinTbLog2SizeY、MaxTbLog2SizeY、MinTbSizeY、MaxTbSizeY、PicWidthInCtbsY、PicHeightInCtbsY、PicSizeInCtbsY、PicWidthInMinCbsY、PicHeightInMinCbsY、PicSizeInMinCbsY、PicSizeInSamplesY、PicWidthInSamplesC和PicHeightInSamplesC推导如下:

[0072] $CtbLog2SizeY = \log_2_ctu_size_minus2 + 2$ (7-9)

[0073] $CtbSizeY = 1 \ll CtbLog2SizeY$ (7-10)

[0074] $MinCbLog2SizeY = \log_2_min_luma_coding_block_size_minus2 + 2$ (7-11)

[0075] $MinCbSizeY = 1 \ll MinCbLog2SizeY$ (7-12)

[0076] $MinTbLog2SizeY = 2$ (7-13)

[0077] $MaxTbLog2SizeY = 6$ (7-14)

[0078] $MinTbSizeY = 1 \ll MinTbLog2SizeY$ (7-15)

[0079] $MaxTbSizeY = 1 \ll MaxTbLog2SizeY$ (7-16)

[0080] $PicWidthInCtbsY = \text{Ceil}(pic_width_in_luma_samples \div CtbSizeY)$ (7-17)

[0081] $PicHeightInCtbsY = \text{Ceil}(pic_height_in_luma_samples \div CtbSizeY)$ (7-18)

[0082] $PicSizeInCtbsY = PicWidthInCtbsY * PicHeightInCtbsY$ (7-19)

[0083] $PicWidthInMinCbsY = pic_width_in_luma_samples / MinCbSizeY$ (7-20)

[0084] $PicHeightInMinCbsY = pic_height_in_luma_samples / MinCbSizeY$ (7-21)

[0085] $PicSizeInMinCbsY = PicWidthInMinCbsY * PicHeightInMinCbsY$ (7-22)

[0086] $PicSizeInSamplesY = pic_width_in_luma_samples * pic_height_in_luma_samples$ (7-23)

[0087] $PicWidthInSamplesC = pic_width_in_luma_samples / SubWidthC$ (7-24)

[0088] $PicHeightInSamplesC = pic_height_in_luma_samples / SubHeightC$ (7-25)

[0089] 3.3.2 一张图片中的CTU

[0090] 假设CTB/LCU尺寸用 $M \times N$ (通常 M 等于 N)表示,并且对于位于图片边界(或片或条带或其他类型的边界,以图片边界为例)的CTB, $K \times L$ 个样点位于图片边界内,其中 $K < M$ 或 $L < N$ 。对于图6A至图6C中示出的那些CTB,CTB尺寸仍然等于 $M \times N$ 。然而,如图6A所示,CTB的下边界在图片之外,或者如图6B所示,CTB的右边界在图片之外,或者如图6C所示,CTB的下边界/右边界在图片之外。

[0091] 3.4 帧内预测

[0092] 为了捕获自然视频中呈现的任意边方向,方向帧内模式的数量从HEVC中使用的33种扩展到65种。图7中示出了扩展的方向模式,并且平面和DC模式保持不变。这些更密集的方向帧内预测模式适用于所有块尺寸以及亮度和色度帧内预测。

[0093] 角度帧内预测方向可以被定义为在顺时针方向上从45度到-135度,如图7所示。在VTM中,对于非正方形块,几个角度帧内预测模式被适应性地被替换为广角帧内预测模式。被替换的模式被通过信号传输并在解析之后重新映射到广角模式的索引。帧内预测模式的总数保持不变,例如,67种,并且帧内模式编解码不变。

[0094] 在HEVC中,每个帧内编解码块都具有正方形形状,并且块的每个边的长度都是2的幂。因此,不需要使用除法运算来使用DC模式生成帧内预测器。在VVC中,块可以具有矩形形状,在一般情况下必须对每个块使用除法运算。为了避免DC预测的除法运算,只有较长边用

于计算非正方形块的平均值。

[0095] 3.5帧间预测

[0096] 对于每个帧间预测的CU,运动参数包括运动矢量、参考图片索引、参考图片列表使用索引以及用于帧间预测的样点生成的VVC的新编解码特征的扩展信息。可以显式或隐式方式通过信号传输运动参数。当使用跳过模式对CU进行编解码时,CU与一个PU相关联,并且没有显著的残差系数、没有编解码的运动矢量增量和/或参考图片索引。Merge模式被规定为,从邻近的CU获得当前CU的运动参数,包括空域和时域候选以及在VVC中引入的扩展调度(schedule)。Merge模式可以应用于任何帧间预测的CU,而不仅仅适用于跳过模式。Merge模式的替代方案是显式发送运动参数,其中对于每个CU,显式地通过信号传输运动矢量、每个参考图片列表的对应参考图片索引、参考图片列表使用标志以及其他有用信息。

[0097] 3.6去方块滤波器

[0098] 去方块滤波是视频编解码器中的示例环路滤波器。在VVC中,去方块滤波过程应用于CU边界、变换子块边界和预测子块边界。预测子块边界包括由基于子块的时间运动矢量预测(SbTMVP)和仿射模式引入的预测单元边界。变换子块边界包括由子块变换(SBT)和帧内子分割(ISP)模式引入的变换单元边界以及由于大CU的隐式划分而进行的变换。去方块滤波器的处理顺序被定义为首先对整个图片的垂直边缘进行水平滤波,然后对水平边缘进行垂直滤波。这种特定顺序使得能够在并行线程中应用多个水平滤波或垂直滤波过程。滤波过程也可逐CTB地实现,且处理延迟很小。

[0099] 首先对图片中的垂直边缘进行滤波。然后,用经过垂直边缘滤波过程修改的样点作为输入,对图片中的水平边缘进行滤波。每个CTU的CTB中的垂直和水平边缘根据编解码单元进行分开处理。按照编解码块的几何顺序从编解码块左侧的边缘开始,向编解码块右侧的边缘进行,对编解码单元中的编解码块的垂直边缘进行滤波。按照编解码块的几何顺序从编解码块顶部的边缘开始,向编解码块底部的边缘进行,对编解码单元中的编解码块的水平边缘进行滤波。

[0100] 图8是样点804的 8×8 块内的样点802的图示800。如图所示,图示800分别包括 8×8 网格806、808上的水平块边界和垂直块边界。另外,图示800示出了 8×8 样点810的非重叠块,这些块可被并行去方块。

[0101] 3.6.1边界决策

[0102] 滤波被应用于 8×8 块边界。另外,这种边界必须是变换块边界或者编解码子块边界,例如,由于使用仿射运动预测(ATMVP)而导致的变换块边界或者编解码子块边界。对于其他边界,禁用去方块滤波器。

[0103] 3.6.2边界强度计算

[0104] 对于变换块边界/编解码子块边界,如果边界位于 8×8 网格内,则可以对边界进行滤波,并且该边缘的 $bS[xDi][yDj]$ 的设置(其中 $[xDi][yDj]$ 表示坐标)分别如表2和表3所定义。

[0105]	优先级	条件	Y	U	V
	5	相邻块中的至少一者是帧内预测的	2	2	2
	4	TU 边界和至少一个相邻块具有非零变换系数	1	1	1
	3	相邻块的参考图片或 MV 数量(单向预测为 1, 双向预测为 2)是不同的	1	N/A	N/A
	2	同一参考图片的属于相邻块的运动矢量之间的绝对差大于或等于一个整数亮度样点	1	N/A	N/A
	1	其他	0	0	0

[0106] 表2. 边界强度(当SPS IBC被禁用时)

[0107]	优先级	条件	Y	U	V
	8	相邻块中的至少一者是帧内预测的	2	2	2
	7	TU 边界和至少一个相邻块具有非零变换系数	1	1	1
	6	相邻块的预测模式不同(例如, 一个是 IBC, 一个是帧间预测)	1		
	5	IBC 和属于相邻块的运动矢量之间的绝对差两者都大于或等于一个整数亮度样点	1	N/A	N/A
	4	相邻块的参考图片或 MV 数量(单向预测为 1, 双向预测为 2)是不同的	1	N/A	N/A
	3	同一参考图片的属于相邻块的运动矢量之间的绝对差大于或等于一个整数亮度样点	1	N/A	N/A
	1	其他	0	0	0

[0108] 表3. 边界强度(当SPS IBC被启用时)

[0109] 3.6.3亮度分量的去方块决策

[0110] 图9是滤波器开/关决策和强/弱滤波器选择所涉及的像素的示例。仅当条件1、条件2和条件3都为真(TRUE)时,才会使用更宽更强的亮度滤波器。条件1是“大块条件”。该条件检测P侧和Q侧的样点是否属于大块,分别由变量bSidePisLargeBlk和bSideQisLargeBlk表示。bSidePisLargeBlk和bSideQisLargeBlk定义如下。

[0111] $bSidePisLargeBlk = ((\text{边缘类型为垂直且 } p_0 \text{ 属于宽度} \geq 32 \text{ 的CU}) \mid \mid (\text{边缘类型为水平且 } p_0 \text{ 属于高度} \geq 32 \text{ 的CU})) ? \text{真} : \text{假}$

[0112] $bSideQisLargeBlk = ((\text{边缘类型为垂直且 } q_0 \text{ 属于宽度} \geq 32 \text{ 的CU}) \mid \mid (\text{边缘类型为水平且 } q_0 \text{ 属于高度} \geq 32 \text{ 的CU})) ? \text{真} : \text{假}$

[0113] 基于bSidePisLargeBlk和bSideQisLargeBlk,条件1定义如下:

- [0114] 条件1 = (bSidePisLargeBlk || bSidePisLargeBlk) ? 真 : 假
- [0115] 接下来, 如果条件1为真, 则进一步检查条件2。首先, 推导以下变量:
- [0116] 首先按照HEVC的方式推导dp0、dp3、dq0、dq3
- [0117] if (p侧大于或等于32)
- [0118] $dp0 = (dp0 + Abs(p50 - 2 * p40 + p30) + 1) \gg 1$
- [0119] $dp3 = (dp3 + Abs(p53 - 2 * p43 + p33) + 1) \gg 1$
- [0120] if (q侧大于或等于32)
- [0121] $dq0 = (dq0 + Abs(q50 - 2 * q40 + q30) + 1) \gg 1$
- [0122] $dq3 = (dq3 + Abs(q53 - 2 * q43 + q33) + 1) \gg 1$
- [0123] 条件2 = (d < β) ? 真 : 假
- [0124] 其中d = dp0 + dq0 + dp3 + dq3。
- [0125] 如果条件1和条件2都满足, 则进一步检查任意块是否使用子块:
- ```

If (bSidePisLargeBlk)
{
 If (模式块 P == SUBBLOCKMODE)
 Sp = 5
 else
[0126] Sp = 7
}
else
 Sp = 3

If (bSideQisLargeBlk)
{
 If (模式块 Q == SUBBLOCKMODE)
 Sq = 5
 else
[0127] Sq = 7
}
else
 Sq = 3

```
- [0128] 最后, 如果条件1和条件2两者都满足, 则去方块方法将检查条件3 (大块强滤波条件), 其定义如下。在条件3StrongFilterCondition中, 推导以下变量:
- [0129] 按照HEVC的方式推导dpq。
- [0130] 按照HEVC的方式推导sp3 = Abs (p3 - p0)

if(p 侧大于或等于 32)

if(Sp==5)

[0131] sp3=(sp3+Abs(p5-p3)+1)>>1

else

sp3=(sp3+Abs(p7-p3)+1)>>1

[0132] 按照HEVC的方式推导sq3=Abs(q0-q3)

if(q 侧大于或等于 32)

If(Sq==5)

[0133] sq3=(sq3+Abs(q5-q3)+1)>>1

else

sq3=(sq3+Abs(q7-q3)+1)>>1

[0134] 按照HEVC, StrongFilterCondition= (dpq小于( $\beta \gg 2$ ), sp3+sq3小于( $3*\beta \gg 5$ ), 且 Abs(p0-q0) 小于 ( $5*tC+1 \gg 1$ ) ?真:假。

[0135] 3.6.4用于亮度的更强的去方块滤波器

[0136] 当边界任一侧的样点属于大块时,使用双线性滤波器。当垂直边缘的宽度 $\geq 32$ 时并且当水平边缘的高度 $\geq 32$ 时,将样点定义为属于大块。双线性滤波器列出如下。块边界样点 $p_i$  ( $i=0$ 至 $Sp-1$ )和 $q_i$  ( $i=0$ 至 $Sq-1$ ),在上述HEVC去方块中, $p_i$ 和 $q_i$ 是用于对垂直边缘进行滤波的行内的第 $i$ 个样点,或用于对水平边缘进行滤波的列内的第 $i$ 个样点,然后通过线性插值替换,如下所示:

[0137]  $p_i' = (f_i * Middle_{s,t} + (64 - f_i) * P_s + 32) \gg 6$ , 限幅至 $p_i \pm tcPD_i$

[0138]  $q_j' = (g_j * Middle_{s,t} + (64 - g_j) * Q_s + 32) \gg 6$ , c限幅至 $q_j \pm tcPD_j$

[0139] 其中 $tcPD_i$ 和 $tcPD_j$ 项是上文描述的位置相关的限幅(clipping),并且 $g_j$ 、 $f_i$ 、 $Middle_{s,t}$ 、 $P_s$ 和 $Q_s$ 给出如下。

[0140] 3.6.5色度去方块决策

[0141] 块边界两侧都使用色度强滤波器。这里,当色度边缘两侧都大于或等于8(色度位置)时,选择色度滤波器,并且满足以下三个条件的决策:第一决策是边界强度以及大块的决策。当与块边缘正交的块宽度或高度在色度样点域中等于或大于8时,可应用该滤波器。第二决策和第三决策与HEVC亮度去方块决策基本相同,分别是开/关决策和强滤波器决策。

[0142] 在第一决策中,针对色度滤波,对边界强度(bS)进行修改,并且循序地检查条件。如果满足条件,则跳过优先级较低的其余条件。当bS等于2时执行色度去方块,或者当检测到大块边界时bS等于1。第二条件和第三条件与HEVC亮度强滤波器决策基本相同,如下所示。

[0143] 在第二条件下,按照HEVC亮度去方块推导d。当d小于 $\beta$ 时,第二条件将为真。在第三条件下,StrongFilterCondition推导如下:

[0144] 按照HEVC的方式推导dpq。

[0145] 按照HEVC的方式推导 $sp_3 = Abs(p_3 - p_0)$

[0146] 按照HEVC的方式推导 $sq_3 = Abs(q_0 - q_3)$

[0147] 按照HEVC设计,  $\text{StrongFilterCondition} = (\text{dpq} < (\beta \gg 2), \text{sp3} + \text{sq3} < (\beta \gg 3),$   
且  $\text{Abs}(\text{p0} - \text{q0}) < (5 * \text{tC} + 1) \gg 1$  )。

[0148] 3.6.6用于色度的强去方块滤波器

[0149] 用于色度的强去方块滤波器定义如下:

[0150]  $\text{p2}' = (3 * \text{p3} + 2 * \text{p2} + \text{p1} + \text{p0} + \text{q0} + 4) \gg 3$

[0151]  $\text{p1}' = (2 * \text{p3} + \text{p2} + 2 * \text{p1} + \text{p0} + \text{q0} + \text{q1} + 4) \gg 3$

[0152]  $\text{p0}' = (\text{p3} + \text{p2} + \text{p1} + 2 * \text{p0} + \text{q0} + \text{q1} + \text{q2} + 4) \gg 3$

[0153] 示例色度滤波器对  $4 \times 4$  色度样点网格执行去方块。

[0154] 3.6.7位置相关限幅

[0155] 位置相关限幅  $\text{tcPD}$  应用于亮度滤波过程的输出样点, 涉及强且长的滤波器, 这些滤波器在边界处修改7、5和3个样点。假设量化误差分布, 可以增大样点的限幅值, 这些样点预期具有更高量化噪声, 因此预期重建样点值与真实样点值的偏差更大。

[0156] 对于用非对称滤波器滤波的每个P或Q边界, 根据决策制定过程的结果, 从两个表(例如, 下面列出的  $\text{Tc7}$  和  $\text{Tc3}$ ) 中选择位置相关的阈值表, 这两个表作为辅助信息提供给解码器:

[0157]  $\text{Tc7} = \{6, 5, 4, 3, 2, 1, 1\}; \text{Tc3} = \{6, 4, 2\};$

[0158]  $\text{tcPD} = (\text{Sp} == 3) ? \text{Tc3} : \text{Tc7};$

[0159]  $\text{tcQD} = (\text{Sq} == 3) ? \text{Tc3} : \text{Tc7};$

[0160] 对于用短对称滤波器滤波的P或Q边界, 应用较低幅度的位置相关阈值:

[0161]  $\text{Tc3} = \{3, 2, 1\};$

[0162] 在定义阈值之后, 根据  $\text{tcP}$  和  $\text{tcQ}$  限幅值对滤波后的  $\text{p}'_i$  和  $\text{q}'_i$  样点值进行限幅:

[0163]  $\text{p}''_i = \text{Clip3}(\text{p}'_i + \text{tcPi}, \text{p}'_i - \text{tcPi}, \text{p}'_i);$

[0164]  $\text{q}''_j = \text{Clip3}(\text{q}'_j + \text{tcQj}, \text{q}'_j - \text{tcQj}, \text{q}'_j);$

[0165] 其中  $\text{p}'_i$  和  $\text{q}'_j$  是滤波后的样点值,  $\text{p}''_i$  和  $\text{q}''_j$  是限幅后的输出样点值, 并且  $\text{tcPi}$  是从VVC  $\text{tc}$  参数以及  $\text{tcPD}$  和  $\text{tcQD}$  推导的限幅阈值。函数  $\text{Clip3}$  是如VVC中规定的限幅函数。

[0166] 3.6.8.子块去方块调整

[0167] 为了使用长滤波器和子块去方块两者实现并行友好地去方块, 长滤波器被限制为在使用子块去方块 (AFFINE或ATMVP或解码器侧运动矢量细化 (DMVR)) 的一侧最多修改5个样点, 如长滤波器的亮度控制中所示。扩展地, 调整子块去方块, 使得靠近CU或隐式TU边界的  $8 \times 8$  网格上的子块边界被限制为在每侧最多修改两个样点。

[0168] 以下内容适用于与CU边界不对齐的子块边界。

[0169] If (模式块Q == SUBBLOCKMODE && edge != 0) {

[0170] if (! (implicitTU && (edge == (64/4))))

[0171] if (edge == 2 || edge == (orthogonalLength - 2) || edge == (56/4) || edge == (72/4))

[0172] Sp = Sq = 2;

[0173] else

[0174] Sp = Sq = 3;

[0175] else

[0176]  $Sp = Sq = bSideQisLargeBlk?5:3$

[0177] }

[0178] 其中边缘等于0对应于CU边界,边缘等于2或等于orthogonalLength-2对应于距CU边界8个样点的子块边界等。如果使用TU的隐式划分,则隐式TU为真。

[0179] 3.7.样点自适应偏移

[0180] 通过使用由编码器为每个CTB规定的偏移,将样点自适应偏移(SAO)应用于去方块滤波器之后的重建信号。视频编码器首先决定是否对当前条带应用SAO处理。如果将SAO应用于条带,则每个CTB将被分类为五种SAO类型中的一者,如表4所示。SAO的概念是将像素分类为多种类别,并通过为每个类别的像素添加偏移来减少失真。SAO操作包括:边缘偏移(E0),在SAO类型1至4中使用边缘属性进行像素分类;以及频带偏移(B0),在SAO类型5中使用像素强度进行像素分类。每个适用的CTB都具有SAO参数,包括sao\_merge\_left\_flag、sao\_merge\_up\_flag、SAO类型和四个偏移。如果sao\_merge\_left\_flag等于1,则当前CTB将重新使用左侧CTB的SAO类型和偏移。如果sao\_merge\_up\_flag等于1,则当前CTB将重新使用上面的CTB的SAO类型和偏移。

[0181]

| SAO类型 | 要使用的样点自适应偏移类型  | 类别数量 |
|-------|----------------|------|
| 0     | 无              | 0    |
| 1     | 1-D 0度模式边缘偏移   | 4    |
| 2     | 1-D 90度模式边缘偏移  | 4    |
| 3     | 1-D 135度模式边缘偏移 | 4    |
| 4     | 1-D 45度模式边缘偏移  | 4    |
| 5     | 频带偏移           | 4    |

[0182] 表4.SAO类型的规定

[0183] 3.8.自适应环路滤波器

[0184] 用于视频编解码的自适应环路滤波是通过使用基于维纳的(Wiener-based)自适应滤波器来将原始样点与解码的样点之间的均方误差最小化。ALF位于每张图片的最后处理阶段,并且可以被视为捕捉和修复来自先前阶段的伪影的工具。合适的滤波器系数由编码器确定并显式地通过信号传输至解码器。为了实现更好的编解码效率,特别是对于高分辨率视频,通过对图片中的不同区域或块应用不同的滤波器,对亮度信号使用局部自适应。除了滤波器自适应之外,编解码树单元(CTU)级别的滤波器开/关控制也有助于提高编解码效率。从语法上讲,滤波器系数在称为自适应参数集的图片级别的头部信息中发送,并且CTU的滤波器开/关标志在条带数据中呈CTU级别交错。这种语法设计不仅支持图片级别优化,而且还实现较低的编码延迟。

[0185] 3.8.1.参数的信号传输

[0186] 根据VTM中的ALF设计,滤波器系数和限幅索引在ALF自适应参数集(APS)中运载。ALF APS最多可包括8个色度滤波器和一个亮度滤波器集,共最多25个滤波器。25个亮度类中的每一类还包括一个索引。具有相同索引的类共用相同的滤波器。通过合并不同的类,减少了表示滤波器系数所需的位数。滤波器系数的绝对值使用0阶指数Golomb码、后跟非零系数的符号位来表示。当启用限幅时,还会使用两位定长码为每个滤波器系数通过信号传输限幅索引。解码器最多可同时使用8个ALF APS。

[0187] VTM中ALF的滤波器控制语法元素包括两种类型的信息。首先,以序列、图片、条带和CTB级别通过信号传输ALF开/关标志。仅当以对应级别启用了亮度ALF时,才可以图片和条带级别启用色度ALF。其次,如果以图片、条带和CTB级别启用了ALF,则以该级别通过信号传输滤波器使用信息。如果图片内的所有条带都使用相同的APS,则参考的ALF APS ID以条带级别或图片级别进行编解码。亮度分量最多可参考7个ALF APS,而色度分量最多可参考1个ALF APS。对于亮度CTB,通过信号传输索引来指示使用哪个ALF APS或经离线训练的亮度滤波器集。对于色度CTB,该索引指示使用参考的APS中的哪个滤波器。

[0188] 与VTM中的LUMA分量相关联的ALF数据语法元素如下所列:

|        |                                                                           |       |
|--------|---------------------------------------------------------------------------|-------|
| [0189] | alf_data() {                                                              | 描述符   |
|        | alf_luma_filter_signal_flag                                               | u(1)  |
|        | if( alf_luma_filter_signal_flag ) {                                       |       |
|        | alf_luma_clip_flag                                                        | u(1)  |
|        | alf_luma_num_filters_signalled_minus1                                     | uc(v) |
|        | if( alf_luma_num_filters_signalled_minus1 > 0 )                           |       |
|        | for( filtIdx = 0; filtIdx < NumAlfFilters; filtIdx++ )                    |       |
|        | alf_luma_coeff_delta_idx[ filtIdx ]                                       | u(v)  |
|        | for( sfIdx = 0; sfIdx <= alf_luma_num_filters_signalled_minus1; sfIdx++ ) |       |
|        | for( j = 0; j < 12; j++ ) {                                               |       |
|        | alf_luma_coeff_abs[ sfIdx ][ j ]                                          | uc(v) |
|        | if( alf_luma_coeff_abs[ sfIdx ][ j ] )                                    |       |
|        | alf_luma_coeff_sign[ sfIdx ][ j ]                                         | u(1)  |
|        | }                                                                         |       |
|        | if( alf_luma_clip_flag )                                                  |       |
|        | for( sfIdx = 0; sfIdx <= alf_luma_num_filters_signalled_minus1; sfIdx++ ) |       |
|        | for( j = 0; j < 12; j++ )                                                 |       |
|        | alf_luma_clip_idx[ sfIdx ][ j ]                                           | u(2)  |
|        | }                                                                         |       |

[0190] alf\_luma\_filter\_signal\_flag等于1规定通过信号传输亮度滤波器集。alf\_luma\_filter\_signal\_flag等于0规定不通过信号传输亮度滤波器集。alf\_luma\_clip\_flag等于0规定将线性自适应环路滤波应用于亮度分量。alf\_luma\_clip\_flag等于1规定可将非线性自适应环路滤波应用于亮度分量。alf\_luma\_num\_filters\_signalled\_minus1加1规定可通过信号传输亮度系数的自适应环路滤波器类的数量。alf\_luma\_num\_filters\_signalled\_minus1的值应在0至NumAlfFilters-1的范围内(包括端点)。alf\_luma\_coeff\_delta\_idx[filtIdx]指示由filtIdx指示的滤波器类的通过信号传输的自适应环路滤波器

亮度系数增量的索引, `filtIdx`的范围为0至`NumAlfFilters-1`。当`alf_luma_coeff_delta_idx[filtIdx]`不存在时,将其推断为等于0。`alf_luma_coeff_delta_idx[filtIdx]`的长度是 $\text{Ceil}(\text{Log2}(\text{alf\_luma\_num\_filters\_signalled\_minus1}+1))$ 位。`alf_luma_coeff_delta_idx[filtIdx]`的值应在0至`alf_luma_num_filters_signalled_minus1`的范围内(包括端点)。

[0191] `alf_luma_coeff_abs[sfIdx][j]`规定由`sfIdx`指示的通过信号传输的亮度滤波器第`j`个系数的绝对值。当`alf_luma_coeff_abs[sfIdx][j]`不存在时,将其推断为等于0。`alf_luma_coeff_abs[sfIdx][j]`的值应在0至128的范围内(包括端点)。`alf_luma_coeff_sign[sfIdx][j]`规定由`sfIdx`指示的滤波器的第`j`个亮度系数的符号,如下所示:

[0192] 如果`alf_luma_coeff_sign[sfIdx][j]`等于0,则对应的亮度滤波器系数具有正值。

[0193] 否则(`alf_luma_coeff_sign[sfIdx][j]`等于1),对应的亮度滤波器系数具有负值。

[0194] 当`alf_luma_coeff_sign[sfIdx][j]`不存在时,将其推断为等于0。

[0195] `alf_luma_clip_idx[sfIdx][j]`规定在乘以由`sfIdx`指示的通过信号传输的亮度滤波器的第`j`个系数之前要使用的限幅值的限幅索引。当`alf_luma_clip_idx[sfIdx][j]`不存在时,将其推断为等于0。与VTM中的亮度分量相关联的编解码树语法元素如下所列:

|        |                                                                    |                    |
|--------|--------------------------------------------------------------------|--------------------|
| [0196] | <code>coding_tree_unit() {</code>                                  | 描述符                |
|        | <code>  xCtb = CtbAddrX &lt;&lt; CtbLog2SizeY</code>               |                    |
|        | <code>  yCtb = CtbAddrY &lt;&lt; CtbLog2SizeY</code>               |                    |
|        | <code>  if( sh_alf_enabled_flag ){</code>                          |                    |
|        | <code>    alf_ctb_flag[ 0 ][ CtbAddrX ][ CtbAddrY ]</code>         | <code>ae(v)</code> |
|        | <code>    if( alf_ctb_flag[ 0 ][ CtbAddrX ][ CtbAddrY ] ) {</code> |                    |
|        | <code>      if( sh_num_alf_aps_ids_luma &gt; 0 )</code>            |                    |
|        | <code>        alf_use_aps_flag</code>                              | <code>ae(v)</code> |
|        | <code>        if( alf_use_aps_flag ) {</code>                      |                    |
|        | <code>          if( sh_num_alf_aps_ids_luma &gt; 1 )</code>        |                    |
|        | <code>            alf_luma_prev_filter_idx</code>                  | <code>ae(v)</code> |
|        | <code>        } else</code>                                        |                    |
|        | <code>          alf_luma_fixed_filter_idx</code>                   | <code>ae(v)</code> |
|        | <code>        }</code>                                             |                    |
|        | <code>  }</code>                                                   |                    |

[0197] `alf_ctb_flag[cIdx][xCtb>>CtbLog2SizeY][yCtb>>CtbLog2SizeY]`等于1规定自适应环路滤波器应用于由亮度位置(`xCtb`,`yCtb`)处的编解码树单元的`cIdx`指示的颜色分量的编解码树块。`alf_ctb_flag[cIdx][xCtb>>CtbLog2SizeY][yCtb>>CtbLog2SizeY]`等于0规定自适应环路滤波器不应用于由亮度位置(`xCtb`,`yCtb`)处的编解码树单元的`cIdx`指示的颜色分量的编解码树块。

[0198] 当`alf_ctb_flag[cIdx][xCtb>>CtbLog2SizeY][yCtb>>CtbLog2SizeY]`不存在时,将其推断为等于0。`alf_use_aps_flag`等于0规定将固定滤波器集中的一者应用于亮度CTB。`alf_use_aps_flag`等于1规定将来自APS的滤波器集应用于亮度CTB。当`alf_use_aps_flag`不存在时,将其推断为等于0。`alf_luma_prev_filter_idx`规定应用于亮度CTB的前一个滤波器。`alf_luma_prev_filter_idx`的值应在0至`sh_num_alf_aps_ids_luma-1`的范围内(包括端点)。当`alf_luma_prev_filter_idx`不存在时,将其推断为等于0。

[0199] 变量`AlfCtbFiltSetIdxY[xCtb>>CtbLog2SizeY][yCtb>>CtbLog2SizeY]`规定位置(`xCtb,yCtb`)处的亮度CTB的滤波器集索引推导如下:

[0200] 如果`alf_use_aps_flag`等于0,则`AlfCtbFiltSetIdxY[xCtb>>CtbLog2SizeY][yCtb>>CtbLog2SizeY]`被设置为等于`alf_luma_fixed_filter_idx`。

[0201] 否则,`AlfCtbFiltSetIdxY[xCtb>>CtbLog2SizeY][yCtb>>CtbLog2SizeY]`被设置为等于`16+alf_luma_prev_filter_idx`。

[0202] `alf_luma_fixed_filter_idx`规定应用于亮度CTB的固定滤波器。`alf_luma_fixed_filter_idx`的值应在0至15的范围内(包括端点)。

[0203] 基于VTM的ALF设计,ECM的ALF设计进一步将替代滤波器集的概念引入到亮度滤波器中。基于每个替代方案/轮次的更新后的亮度CTU ALF开/关决策,对亮度滤波器进行多个替代方案/轮次的训练。这样,将有多个滤波器集与每个训练替代方案相关联,并且每个滤波器集的分类合并结果可能不同。每个CTU可选择由RDO设定的最佳滤波器,并且将通过信号传输相关的替代方案信息。与ECM中的亮度分量相关联的ALF的数据语法元素如下所列:

|        |                                                                                  |                    |
|--------|----------------------------------------------------------------------------------|--------------------|
| [0204] | <code>alf_data( ) {</code>                                                       | 描述符                |
|        | <code>alf_luma_filter_signal_flag</code>                                         | <code>u(1)</code>  |
|        | <code>if( alf_luma_filter_signal_flag ) {</code>                                 |                    |
|        | <code>alf_luma_num_alts_minus1</code>                                            | <code>ue(v)</code> |
|        | <code>for(altIdx = 0; altIdx &lt; alf_luma_num_alts_minus1 +1; altIdx++){</code> |                    |
|        | <code>alf_luma_clip_flag[altIdx]</code>                                          | <code>u(1)</code>  |
|        | <code>alf_luma_num_filters_signalled_minus1[altIdx]</code>                       | <code>ue(v)</code> |
|        | <code>if(alf_luma_num_filters_signalled_minus1[altIdx] &gt; 0){</code>           |                    |
|        | <code>for( filtIdx = 0; filtIdx &lt; NumAlfFilters; filtIdx++ )</code>           |                    |
|        | <code>alf_luma_coeff_delta_idx[altIdx][filtIdx]</code>                           | <code>u(v)</code>  |
|        | <code>}</code>                                                                   |                    |
|        | <code>for(sfIdx = 0; sfIdx &lt;=</code>                                          |                    |
|        | <code>alf_luma_num_filters_signalled_minus1[altIdx]; sfIdx++){</code>            |                    |
|        | <code>for(j = 0; j &lt; 19; j++){</code>                                         |                    |
|        | <code>alf_luma_coeff_abs[altIdx][ sfIdx ][ j ]</code>                            | <code>ue(v)</code> |

|        |                                                          |      |
|--------|----------------------------------------------------------|------|
| [0205] | if( alf_luma_coeff_abs[altIdx][ sfIdx ][ j ] )           |      |
|        | alf_luma_coeff_sign[altIdx][ sfIdx ][ j ]                | u(1) |
|        | }                                                        |      |
|        | }                                                        |      |
|        | if( alf_luma_clip_flag [altIdx])                         |      |
|        | for( sfIdx = 0; sfIdx <=                                 |      |
|        | alf_luma_num_filters_signalled_minus1[altIdx]; sfIdx++ ) |      |
|        | for( j = 0; j < 19; j++ )                                |      |
|        | alf_luma_clip_idx[altIdx][ sfIdx ][ j ]                  | u(2) |
|        | }                                                        |      |
|        | }                                                        |      |

[0206] alf\_luma\_num\_alts\_minus1加1规定亮度分量的替代滤波器集的数量。alf\_luma\_num\_alts\_minus1的值应在0至3的范围中(包括端点)。alf\_luma\_clip\_flag[altIdx]等于0规定将线性自适应环路滤波应用于具有索引altIdx的亮度分量的替代亮度滤波器集。alf\_luma\_clip\_flag[altIdx]等于1规定可将非线性自适应环路滤波应用于具有索引altIdx的亮度分量的替代亮度滤波器集。alf\_luma\_num\_filters\_signalled\_minus1[altIdx]加1规定可以向具有索引altIdx的替代亮度滤波器集通过信号传输亮度系数的自适应环路滤波器类的数量。alf\_luma\_num\_filters\_signalled\_minus1[altIdx]的值应在0至NumAlfFilters-1的范围内(包括端点)。

[0207] alf\_luma\_coeff\_delta\_idx[altIdx][filtIdx]规定对于具有索引altIdx的替代亮度滤波器集由filtIdx表示的用于滤波器类的通过信号传输的自适应环路滤波器亮度系数增量的索引,filtIdx范围为0至NumAlfFilters-1。当alf\_luma\_coeff\_delta\_idx[filtIdx][altIdx]不存在时,将其推断为等于0。alf\_luma\_coeff\_delta\_idx[altIdx][filtIdx]的长度是Ceil(Log2(alf\_luma\_num\_filters\_signalled\_minus1[altIdx]+1))位。alf\_luma\_coeff\_delta\_idx[altIdx][filtIdx]的值应在0至alf\_luma\_num\_filters\_signalled\_minus1[altIdx]范围内(包括端点)。alf\_luma\_coeff\_abs[altIdx][sfIdx][j]规定由具有索引altIdx的替代亮度滤波器集的sfIdx指示的通过信号传输的亮度滤波器的第j个系数的绝对值。当alf\_luma\_coeff\_abs[altIdx][sfIdx][j]不存在时,将其推断为等于0。alf\_luma\_coeff\_abs[altIdx][sfIdx][j]的值应在0至128的范围内(包括端点)。

[0208] alf\_luma\_coeff\_sign[altIdx][sfIdx][j]规定由具有索引altIdx的替代亮度滤波器集的sfIdx指示的滤波器的第j个亮度系数的符号,如下所示:

[0209] 如果alf\_luma\_coeff\_sign[altIdx][sfIdx][j]等于0,则对应的亮度滤波器系数具有正值。

[0210] 否则(alf\_luma\_coeff\_sign[altIdx][sfIdx][j]等于1),对应的亮度滤波器系数具有负值。

[0211] 当alf\_luma\_coeff\_sign[altIdx][sfIdx][j]不存在时,将其推断为等于0。

[0212] alf\_luma\_clip\_idx[altIdx][sfIdx][j]规定在乘以索引为altIdx的替代亮度滤

波器集的由sfIdx表示的被通过信号传输的亮度滤波器的第j个系数之前要使用的限幅值的限幅索引。当alf\_luma\_clip\_idx[altIdx][sfIdx][j]不存在时,将其推断为等于0。与ECM中的亮度分量相关联的编解码树语法元素如下所列:

|        |                                                                        |       |
|--------|------------------------------------------------------------------------|-------|
| [0213] | <code>coding_tree_unit() {</code>                                      | 描述符   |
|        | <code>  xCtb = CtbAddrX &lt;&lt; CtbLog2SizeY</code>                   |       |
|        | <code>  yCtb = CtbAddrY &lt;&lt; CtbLog2SizeY</code>                   |       |
|        | <code>  if( sh_alf_enabled_flag ){</code>                              |       |
|        | <code>    alf_ctb_flag[ 0 ][ CtbAddrX ][ CtbAddrY ]</code>             | ae(v) |
|        | <code>    if( alf_ctb_flag[ 0 ][ CtbAddrX ][ CtbAddrY ] ) {</code>     |       |
|        | <code>      if( sh_num_alf_aps_ids_luma &gt; 0 )</code>                |       |
|        | <code>        alf_use_aps_flag</code>                                  | ae(v) |
|        | <code>        if( alf_use_aps_flag ) {</code>                          |       |
|        | <code>          if( sh_num_alf_aps_ids_luma &gt; 1 )</code>            |       |
|        | <code>          alt_ctb_luma_filter_alt_idx[CtbAddrX][CtbAddrY]</code> | ae(v) |
|        | <code>          alf_luma_prev_filter_idx</code>                        | ae(v) |
|        | <code>        } else</code>                                            |       |
|        | <code>          alf_luma_fixed_filter_idx</code>                       | ae(v) |
|        | <code>        }</code>                                                 |       |
|        | <code>  }</code>                                                       |       |

[0214] `alf_ctb_luma_filter_alt_idx[xCtb>>CtbLog2SizeY][yCtb>>CtbLog2SizeY]`规定应用于亮度位置(xCtb,yCtb)处的编解码树单元的亮度分量的编解码树块的替代亮度滤波器的索引。当`alf_ctb_luma_filter_alt_idx[xCtb>>CtbLog2SizeY][yCtb>>CtbLog2SizeY]`不存在时,将其推断为等于零。

### [0215] 3.8.2. 滤波器形状

[0216] 在JEM中,可为亮度分量选择最多三种菱形滤波器形状(如图10所示)。以图片级别通过信号传输索引来指示用于亮度分量的滤波器形状。每个正方形表示一个样点,并且Ci(i为0~6(左)、0~12(中)、0~20(右))表示要应用于该样点的系数。对于图片中的色度分量,始终使用5×5菱形形状。在VVC中,7×7菱形形状始终用于亮度,而5×5菱形形状始终用于色度。

### [0217] 3.8.3 ALF的分类

[0218] 每个2×2(或4×4)块被归类为25个类中的一者。分类索引C是基于其方向性D和活动的量化值 $\hat{A}$ 推导的,如下所示:

$$[0219] \quad C = 5D + \hat{A}.$$

[0220] 为了计算D和 $\hat{A}$ ,首先使用1维拉普拉斯算子计算水平、垂直和两个对角线方向的梯度:

$$\begin{aligned}
g_v &= \sum_{k=i-2}^{i+3} \sum_{l=j-2}^{j+3} V_{k,l}, V_{k,l} = |2R(k,l) - R(k,l-1) - R(k,l+1)|, \\
g_h &= \sum_{k=i-2}^{i+3} \sum_{l=j-2}^{j+3} H_{k,l}, H_{k,l} = |2R(k,l) - R(k-1,l) - R(k+1,l)|, \\
[0221] \quad g_{d1} &= \sum_{k=i-2}^{i+3} \sum_{l=j-3}^{j+3} D1_{k,l}, D1_{k,l} = |2R(k,l) - R(k-1,l-1) - R(k+1,l+1)| \\
g_{d2} &= \sum_{k=i-2}^{i+3} \sum_{l=j-2}^{j+3} D2_{k,l}, D2_{k,l} = |2R(k,l) - R(k-1,l+1) - R(k+1,l-1)|
\end{aligned}$$

[0222] 索引*i*和*j*是指 $2 \times 2$ 块中的左上样点的坐标,并且*R*(*i*,*j*)指示坐标(*i*,*j*)处的重建样点。水平和垂直方向的梯度的D最大值和最小值被设定为:

$$[0223] \quad g_{h,v}^{max} = \max(g_h, g_v), \quad g_{h,v}^{min} = \min(g_h, g_v),$$

[0224] 并且两个对角线方向的梯度的最大值和最小值被设定为:

$$[0225] \quad g_{d0,d1}^{max} = \max(g_{d0}, g_{d1}), \quad g_{d0,d1}^{min} = \min(g_{d0}, g_{d1}),$$

[0226] 为了推导方向性D的值,将这些值彼此进行比较并与两个阈值*t*<sub>1</sub>和*t*<sub>2</sub>进行比较:

[0227] 步骤1.如果 $g_{h,v}^{max} \leq t_1 \cdot g_{h,v}^{min}$ 并且 $g_{d0,d1}^{max} \leq t_1 \cdot g_{d0,d1}^{min}$ 两者都为真,则将D设定为0。

[0228] 步骤2.如果 $g_{h,v}^{max} / g_{h,v}^{min} > g_{d0,d1}^{max} / g_{d0,d1}^{min}$ ,则从步骤3继续;否则从步骤4继续;

[0229] 步骤3.如果 $g_{h,v}^{max} > t_2 \cdot g_{h,v}^{min}$ ,则将D设定为2;否则,将D设定为1。

[0230] 步骤4.如果 $g_{d0,d1}^{max} > t_2 \cdot g_{d0,d1}^{min}$ ,则将D设定为4;否则,将D设定为3。

[0231] 活动值A被计算为:

$$[0232] \quad A = \sum_{k=i-2}^{i+3} \sum_{l=j-2}^{j+3} (V_{k,l} + H_{k,l}).$$

[0233] A被进一步量化为0至4的范围(包括端点),并且量化值被表示为 $\hat{A}$ 。对于图片中的两个色度分量,不采用任何分类方法(即,对每个色度分量应用一组ALF系数)。

[0234] 3.8.4.滤波器系数的几何变换

[0235] 在对每个 $2 \times 2$ 块进行滤波之前,根据为该块计算的梯度值,对与坐标(*k*,*l*)相关的滤波器系数*f*(*k*,*l*)应用诸如旋转或对角线和垂直翻转等几何变换。这相当于将这些变换应用于滤波器支持区域中的样点。这个想法是通过对齐ALF所应用的不同块的方向性使得这些块更加相似。

[0236] 引入三种几何变换,包括对角线变换、垂直翻转和旋转:

$$[0237] \quad \text{对角线: } f_D(k,l) = f(l,k),$$

$$[0238] \quad \text{竖直翻转: } f_V(k,l) = f(k, K-1-l),$$

$$[0239] \quad \text{旋转: } f_R(k,l) = f(K-1-l, k).$$

[0240] 其中K是滤波器的尺寸,并且 $0 \leq k, l \leq K-1$ 是系数坐标,使得位置(0,0)位于左上角,并且位置(K-1,K-1)位于右下角。根据为该块计算的梯度值,将变换应用于滤波器系数*f*(*k*,*l*)。表5总结了变换与四个方向的四个梯度之间的关系。图11示出了基于 $5 \times 5$ 菱形的每

个位置的变换系数。

|        |                                  |      |
|--------|----------------------------------|------|
| [0241] | 梯度值                              | 变换   |
|        | $g_{d2} < g_{d1}$ 并且 $g_h < g_v$ | 无变换  |
|        | $g_{d2} < g_{d1}$ 并且 $g_v < g_h$ | 对角线  |
|        | $g_{d1} < g_{d2}$ 并且 $g_h < g_v$ | 竖直翻转 |
|        | $g_{d1} < g_{d2}$ 并且 $g_v < g_h$ | 旋转   |

[0242] 表5. 为一个块计算的梯度和变换的映射。

[0243] 3.8.5. 滤波过程

[0244] 在解码器侧, 当为块启用ALF时, 将对块内的每个样点 $R(i, j)$ 进行滤波, 得到如下所示的样点值 $R'(i, j)$ , 其中 $L$ 表示滤波器长度,  $f_{m,n}$ 表示滤波器系数, 并且 $f(k, l)$ 表示解码后的滤波器系数。

$$[0245] \quad R'(i, j) = \sum_{k=-L/2}^{L/2} \sum_{l=-L/2}^{L/2} f(k, l) \times R(i+k, j+l)$$

[0246] 图12示出了用于 $5 \times 5$ 菱形滤镜支持的相对坐标的示例, 假设当前样点的坐标 $(i, j)$ 为 $(0, 0)$ 。用相同颜色填充的不同坐标中的样点与相同的滤波系数相乘。

[0247] 3.8.6. 非线性滤波重构(reformulation)

[0248] 线性滤波可以被重构为以下表达式, 而不会影响编解码效率:

$$[0249] \quad O(x, y) = I(x, y) + \sum_{(i,j) \neq (0,0)} w(i, j) \cdot (I(x+i, y+j) - I(x, y))$$

[0250] 其中 $w(i, j)$ 是相同的滤波器系数。

[0251] VVC引入非线性以在近邻样点值 $(I(x+i, y+j))$ 与被滤波的当前样点值 $(I(x, y))$ 相差太大时通过使用简单的限幅函数降低这些相邻样点值的影响来使ALF更加高效。更具体地, ALF滤波器修改如下:

$$[0252] \quad O'(x, y) = I(x, y) + \sum_{(i,j) \neq (0,0)} w(i, j) \cdot K(I(x+i, y+j) - I(x, y), k(i, j))$$

[0253] 其中 $K(d, b) = \min(b, \max(-b, d))$ 是限幅函数, 并且 $k(i, j)$ 是限幅参数, 它们取决于 $(i, j)$ 滤波器系数。编码器执行优化以找到最佳 $k(i, j)$ 。

[0254] 为每个ALF滤波器指定限幅参数 $k(i, j)$ , 每个滤波器系数都通过信号传输一个限幅值。这意味着每个亮度滤波器的比特流中最多可通过信号传输12个限幅值, 并且每个色度滤波器的比特流中最多可通过信号传输6个限幅值。为了限制信令成本和编码器复杂度, 仅使用4个固定值, 这些固定值对于帧间和帧内条带是相同的。

[0255] 因为亮度的局部差异的方差通常高于色度的局部差异的方差, 所以应用了用于亮度滤波器和色度滤波器的两个不同集合。还引入了每个集合中的最大样点值(这里对于10位的位深度为1024), 使得在不需要时可禁用限幅。这4个值已通过在对数域中粗略平均地划分亮度的样点值的全部范围(以10位编解码)以及色度的4至1024的范围而选择的。更准确地, 限幅值的亮度表是通过以下公式获得的:

[0256]  $\text{AlfClip}_L = \left\{ \text{round} \left( \left( (M)^{\frac{1}{N}} \right)^{N-n+1} \right) \text{ for } n \in 1..N \right\}$ , 其中  $M=2^{10}$  并且  $N=4$

[0257] 类似地, 限幅值的色度表可根据以下公式获得:

[0258]  $\text{AlfClip}_C = \left\{ \text{round} \left( A \cdot \left( \left( \frac{M}{A} \right)^{\frac{1}{N-1}} \right)^{N-n} \right) \text{ for } n \in 1..N \right\}$ , 其中  $M=2^{10}$ ,  $N=4$  并且  $A=4$

[0259] 3.9. 双边环路滤波器

[0260] 3.9.1. 双边图像滤波器

[0261] 双边图像滤波器是在保留边缘结构的同时将噪声平滑化的非线性滤波器。双边滤波是使滤波器权重不仅随着样点之间的距离而减小, 而且随着强度差的增大而减小的技术。这样, 可改善边缘过度平滑化。权重被定义为

[0262]  $w(\Delta x, \Delta y, \Delta I) = e^{-\frac{\Delta x^2 + \Delta y^2}{2\sigma_d^2} - \frac{\Delta I^2}{2\sigma_r^2}}$

[0263] 其中  $\Delta x$  和  $\Delta y$  分别是垂直和水平方向上的距离, 并且  $\Delta I$  是样点之间的强度差。

[0264] 边缘保持去噪双边滤波器对域滤波器和范围滤波器两者均采用低通高斯滤波器。域低通高斯滤波器赋予空间上靠近中心像素的像素更高的权重。范围低通高斯滤波器赋予类似于中心像素的像素更高的权重。结合范围滤波器和域滤波器, 边缘像素处的双边滤波器变为沿着边缘取向并且在梯度方向上大幅减小的细长高斯滤波器。这就是双边滤波器可在保留边缘结构的同时将噪声平滑化的原因。

[0265] 3.9.2. 视频编解码中的双边滤波器

[0266] 视频编解码中的双边滤波器是用于VVC的编解码工具。滤波器与样点自适应偏移(SAO)滤波器并行充当环路滤波器。双边滤波器和SAO两者都作用于相同的输入样点, 每个滤波器产生一个偏移, 然后这些偏移被添加到输入样点以产生输出样点, 该输出样点在限幅之后进入下一阶段。空间滤波强度 $\sigma_d$ 由块尺寸决定, 其中块越小, 滤波强度就越大, 并且强度滤波强度 $\sigma_r$ 由量化参数决定, 其中较强的滤波用于较高的QPs。仅使用四个最接近的样点, 因此滤波后的样点强度 $I_F$ 可被计算为

[0267]  $I_F = I_C + \frac{w_A \Delta I_A + w_B \Delta I_B + w_L \Delta I_L + w_R \Delta I_R}{w_C + w_A + w_B + w_L + w_R}$

[0268] 其中 $I_C$ 表示中心样点的强度、 $\Delta I_A = I_A - I_C$ 表示中心样点与上方样点之间的强度差。 $\Delta I_B$ ,  $\Delta I_L$ 和 $\Delta I_R$ 表示中心样点分别与下方、左侧和右侧样点之间的强度差。

[0269] 4. 通过所公开的技术解决方案解决的技术问题

[0270] 视频编解码中的自适应环路滤波器(ALF)的示例设计具有以下问题:

[0271] 在一些ALF设计中, 仅经过其他滤波(诸如去方块滤波(DBF)、样点自适应偏移(SAO)和双边滤波(BF))之后的空域重建样点被用于滤波器训练和滤波。然而, 可潜在地利用其他有价值的信息, 诸如由一个或多个预定义滤波器滤波/生成的样点。

[0272] 在一些ALF设计中, 仅经过其他滤波(诸如去方块滤波、SAO和BF)之后的空域重建样点用于滤波器训练和滤波。然而, 可潜在地利用其他有价值的信息, 诸如DBF、SAO或其他阶段之前的样点。

[0273] 5. 解决方案和实施例的列表

[0274] 为了解决上述问题,公开了如下文总结的方法。实施例应被视为解释一般概念的范例,而不应狭隘地加以解释。此外,这些实施例可以单独地应用或以任何方式组合。应注意,所公开的方法可用作环路滤波器或后处理。在本公开中,视频单元可以指代序列、图片、子图片、条带、CTU、块和/或区域。视频单元可包括一个颜色分量或多个颜色分量。在本公开中,ALF处理单元可以指代序列、图片、子图片、条带、CTU、块、区域或样点。ALF处理单元可包括一个颜色分量或多个颜色分量。

[0275] 在以下公开内容中,滤波后的样点值被表示为BF的输出。例如,在如下描述的过程中:

$$[0276] \quad I_F = I_C + \frac{w_A \Delta I_A + w_B \Delta I_B + w_L \Delta I_L + w_R \Delta I_R}{w_C + w_A + w_B + w_L + w_R},$$

[0277] 其中 $I_F$ 表示BF的输出。

[0278] 在以下公开内容中,BF表示视频编解码中的双边滤波的示例,其通常使用预定义的滤波参数来产生BF的输出。例如,没有在线训练或通过信号传输用于BF中的滤波参数。

[0279] 在以下公开内容中,自适应BF表示在视频编解码中的双边滤波之上的BF的改进版本。自适应BF涉及在线参数训练和参数的信号传输。例如,基于BF生成的多个滤波后的样点进一步与通过信号传输并且经在线训练的参数组合。

[0280] 示例1

[0281] 提出使用滤波器的中间滤波结果作为一个或多个扩展抽头的输入。在一个示例中,经离线训练的ALF滤波器的中间滤波结果可用作一个或多个扩展抽头的输入。在一个示例中,中间滤波结果可以由ALF之前的重建和ALF的经离线训练的滤波器生成。在一个示例中,中间滤波结果可以由DBF之前的重建和ALF的经离线训练的滤波器生成。在一个示例中,经在线训练的ALF滤波器的中间滤波结果可用作一个或多个扩展抽头的输入。在一个示例中,其他预定义滤波器的中间滤波结果可用作一个或多个扩展抽头的输入。

[0282] 在一个示例中,可应用高斯滤波器。在一个示例中,可应用双边滤波器。在一个示例中,可应用导向滤波器。在一个示例中,可应用中值滤波器。在一个示例中,可应用局部中值滤波器。在一个示例中,可应用非局部中值滤波器。在一个示例中,可应用具有低通性质的滤波器。在一个示例中,可应用具有高通性质的滤波器。

[0283] 在一个示例中,其他经在线训练的滤波器的中间滤波结果可用作一个或多个扩展抽头的输入。在一个示例中,中间滤波的输入可以是不同编解码阶段处的重建样点。在一个示例中,当前/参考帧的ALF之前/之后的重建可用于生成中间滤波结果。在一个示例中,当前/参考帧的SAO/跨分量样点自适应偏移 (CCSAO) 之前/之后的重建可用于生成中间滤波结果。在一个示例中,当前/参考帧的双边滤波器 (BIF) 之前/之后的重建可用于生成中间滤波结果。在一个示例中,当前/参考帧的DBF之前/之后的重建可用于生成中间滤波结果。在一个示例中,当前/参考帧的任何其他阶段之前/之后的重建可用于生成中间滤波结果。

[0284] 示例2

[0285] 提出使用当前帧的不同编解码阶段之前/之后的重建样点作为一个或多个扩展抽头的输入。在一个示例中,当前帧的DBF之前/之后的重建可用作一个或多个扩展抽头的输入。在一个示例中,当前帧的SAO/CCSAO之前/之后的重建可用作一个或多个扩展抽头的输入。在一个示例中,当前帧的BIF之前/之后的重建可用作一个或多个扩展抽头的输入。在一

个示例中,当前帧的其他阶段之前/之后的重建可用作一个或多个扩展抽头的输入。

[0286] 示例3

[0287] 提出使用一个或多个编解码图片内部的样点作为一个或多个扩展抽头的输入源。在一个示例中,先前编解码的帧可以是与块/当前条带/帧相关联的参考图片列表(RPL)或参考图片集(RPS)中的参考帧。在一个示例中,先前编解码的帧可以是块/当前条带/帧的短期参考图片。在一个示例中,先前编解码的帧可以是块/当前条带/帧的长期参考图片。

[0288] 示例4

[0289] 在一个示例中,先前编解码的帧可能不是参考帧,但是它仍存储在解码图片缓存区(DPB)中。

[0290] 示例5

[0291] 在一个示例中,通过信号传输至少一个指示符来指示要使用哪一个(哪些)先前编解码的帧。在一个示例中,通过信号传输一个指示符来指示要使用哪个参考图片列表。在一个示例中,可通过信号传输至少一个指示符以指示参考索引。在一个示例中,可以有条件地通过信号传输指示符,例如,取决于RPL/RPS中包括多少个参考图片。在一个示例中,可以有条件地通过信号传输指示符,例如,取决于DPB中包括多少个先前解码的图片。

[0292] 示例6

[0293] 在一个示例中,即时确定要利用哪些帧。在一个示例中,扩展抽头可以从DPB中的一个或多个先前编解码的帧获取信息。在一个示例中,扩展抽头可以从列表0中的一个/多个参考帧获取信息。在一个示例中,扩展抽头可以从列表1中的一个/多个参考帧获取信息。在一个示例中,扩展抽头可以从列表0和列表1两者中的参考帧获取信息。在一个示例中,扩展抽头可以从最接近当前帧(例如,与当前条带/帧具有最小的图片顺序计数(POC)距离)的参考帧获取信息。

[0294] 在一个示例中,扩展抽头可以从参考列表中参考索引等于K(例如, $K=0$ )的参考帧获取信息。在一个示例中,K可以是预定义的。在一个示例中,可以根据参考图片信息即时推导K。在一个示例中,可以通过信号传输K。

[0295] 在一个示例中,扩展抽头可以从同位帧获取信息。在一个示例中,可以通过解码信息来确定要利用哪个帧。在一个示例中,要利用的帧可以被定义为当前条带/帧内的样点的前N个(例如, $N=1$ )最常使用的参考图片。在一个示例中,要利用的帧可以被定义为针对当前条带/帧内的样点的每个参考图片列表(如果可用)的前N个(例如, $N=1$ )最常使用的参考图片。在一个示例中,要利用的帧可以被定义为相对于当前图片具有前N个(例如, $N=1$ )最小POC距离/绝对POC距离的图片。

[0296] 示例7

[0297] 在一个示例中,是否从先前编解码的帧获取信息可以取决于待滤波块的至少一个区域的解码信息(例如,编解码模式/统计数据/特性)。在一个示例中,编码器可以向解码器信号通知是否从待滤波块的至少一个区域的先前编解码的帧中获取信息。在一个示例中,是否从先前编解码的帧获取信息可以取决于条带/图片类型。在一个示例中,它可能仅适用于帧间编解码的条带/图片(例如,P或B条带/图片)。

[0298] 在一个示例中,是否从先前编解码的帧获取信息可以取决于参考图片的可用性。在一个示例中,是否从先前编解码的帧获取信息可以取决于参考图片信息或DPB中的图片

信息。在一个示例中,如果最小POC距离(例如,参考图片/DPB中的图片与当前图片之间的最小POC距离)大于阈值,则禁用它。在一个示例中,是否从先前编解码的帧中获取信息可取决于时间层索引和/或QP和/或图片的维度。在一个示例中,它可能适用于具有给定的时间层索引(例如,最高时间层)的块。在一个示例中,如果待滤波块包含以非帧间模式编解码的样点的一部分,则扩展抽头不能使用来自先前编解码的帧的信息来对该块进行滤波。

[0299] 在一个示例中,非帧间模式可以被定义为帧内模式。在一个示例中,非帧间模式可以被定义为包括但不限于帧内/IBC/调色板模式的一组编解码模式。在一个示例中,计算当前块与匹配块之间的失真,并将其用于决定是否从先前编解码的帧获取信息来对当前块进行滤波。

[0300] 在一个示例中,可以使用先前编解码的帧中的同位块与当前块之间的失真来决定是否从先前编解码的帧获取信息来对当前块进行滤波。在一个示例中,可以首先使用运动估计来从至少一个先前编解码的帧中找到匹配块。在一个示例中,当失真大于预定义阈值时,不能使用来自先前编解码的帧的信息。

[0301] 示例8

[0302] 在一个示例中,该信息包含当前块的两个参考块和/或同位块,其中一个来自列表-0中的第一参考帧,并且另一个来自列表-1中的第一参考帧。

[0303] 示例9

[0304] 在一个示例中,所公开的方法可用于后处理和/或预处理。

[0305] 示例10

[0306] 在一个示例中,可以联合使用上文提及的方法。

[0307] 示例11

[0308] 在一个示例中,可以单独使用上文提及的方法。

[0309] 示例12

[0310] 在一个示例中,用于ALF方法的所提出/描述的扩展抽头可应用于视频编解码中的任何环路滤波工具、预处理或后处理滤波方法(包括但不限于ALF/跨分量自适应环路滤波器(CCALF)或任何其他滤波方法)。在一个示例中,所提出的扩展抽头方法可应用于环路滤波方法。在一个示例中,所提出的扩展抽头方法可应用于ALF。在一个示例中,所提出的扩展抽头方法可以应用于CCALF。在一个示例中,所提出的扩展抽头方法可应用于其他环路滤波方法。在一个示例中,所提出的扩展抽头方法可应用于预处理滤波方法。在一个示例中,所提出的扩展抽头方法可应用于后处理滤波方法。

[0311] 示例13

[0312] 在以上示例中,视频单元可以指代序列/图片/子图片/条带/片/编解码树单元(CTU)/CTU行/CTU组/编解码单元(CU)/预测单元(PU)/变换单元(TU)/编解码树块(CTB)/编解码块(CB)/预测块(PB)/变换块(TB)/包含超过一个亮度或色度样点/像素的任何其他区域。

[0313] 示例14

[0314] 可以在比特流中通过信号传输是否和/或如何应用上文公开的方法。在一个示例中,它们可以序列级别/图片组级别/图片级别/条带级别/片组级别通过信号传输,例如在序列头、图片头、SPS、VPS、DPS、解码器能力信息(DCI)、PPS、APS、条带头和片组头中通过信

号传输。在一个示例中,它们可以在包含超过一个样点或像素的PB/TB/CB/PU/TU/CU/虚拟管道数据单元(VPDU)/CTU/CTU行/条带/片/子图片/其他类型的区域处通过信号传输。

[0315] 示例15

[0316] 是否和/或如何应用上文公开的方法可以取决于编解码信息,诸如块尺寸、颜色格式、单/双树分割、颜色分量、条带/图片类型。

[0317] 其他示例

[0318] 在一个示例中,ALF滤波器的至少一个扩展抽头进一步增强ALF的效率。在一个示例中,至少一个扩展抽头可不同于ALF滤波器中的空域抽头,该空域抽头仅利用目标分量的空域邻近样点的信息。在一个示例中,空域邻近样点可来自DBF/SAO/BF之后的重建。在一个示例中,空域抽头可仅使用空域邻近亮度样点来对一个ALF滤波器内部的中心亮度样点进行滤波。在一个示例中,空域抽头可仅使用空域邻近色度样点来对一个ALF滤波器内部的中心色度样点进行滤波。

[0319] 在一个示例中,至少一个扩展抽头和至少一个空域抽头共存于一个ALF滤波器内。在一个示例中,ALF滤波器可以包括空域抽头和扩展抽头两者。在一个示例中,ALF滤波器可以包括M(例如, $M>0$ )个空域抽头和N(例如, $N>0$ )个扩展抽头。

[0320] 在一个示例中,ALF滤波器的一个或多个扩展抽头可以使用一个或多个输入源。在一个示例中,ALF滤波器的一个或多个扩展抽头可以使用一个输入源。(例如,DBF之前的重建或预定义滤波器的中间滤波结果或SAO/BF之前的重建)。在一个示例中,ALF滤波器的一个或多个扩展抽头可以使用多个输入源。(例如,DBF之前的重建和预定义滤波器的中间滤波结果)。扩展抽头的输入源也可以被ALF中的其他抽头使用。扩展抽头的输入源不能被ALF中的其他抽头使用。扩展抽头的输入源可以是预测样点。扩展抽头的输入源可以是滤波器预测样点。扩展抽头的输入源可以是多个源的加权和或滤波。扩展抽头的输入源可以是多个源的函数的输出。扩展抽头的输入源可以基于不同颜色分量中的样点推导。

[0321] 在一个示例中,对于不同的颜色格式和/或不同的颜色分量,是否和/或如何应用具有至少一个扩展抽头的滤波器可能有所不同。在一个示例中,具有至少一个扩展抽头的ALF滤波器可仅应用于处理亮度分量。在一个示例中,具有至少一个扩展抽头的ALF滤波器可仅应用于处理色度分量中的一者(例如,Cb或Cr分量)。在一个示例中,具有至少一个扩展抽头的ALF滤波器可应用于处理所有色度分量(例如,Cb和Cr分量)。在一个示例中,具有至少一个扩展抽头的ALF滤波器可应用于对亮度和色度分量进行滤波(例如,Y、Cb和Cr分量)。

[0322] 在一个示例中,ALF滤波器内部的扩展抽头的系数可对应于一个或多个输入样点。在一个示例中,ALF滤波器内部的扩展抽头的系数可仅对应于一个输入样点。在一个示例中,ALF滤波器内部的扩展抽头的系数可对应于N个输入样点(例如, $N=2$ )。在一个示例中,N个输入样点可以对称方式进行设计。在一个示例中,N个输入样点可以非对称方式进行设计。在一个示例中,ALF滤波器内部的扩展抽头的系数可基于几何信息被多个输入共享。在一个示例中,多个输入样点可位于对称位置处。

[0323] 在一个示例中,具有至少一个扩展抽头的ALF滤波器可以使用不同形状或尺寸。在一个示例中,ALF滤波器可包含不同形状的空域抽头和扩展抽头。在一个示例中,在ALF滤波器内部,用于一个或多个空域抽头的形状/尺寸可与用于一个或多个扩展抽头的形状/尺寸不同。在一个示例中,在ALF滤波器内部,用于一个或多个空域抽头的形状/尺寸可与用于一

个或多个扩展抽头的形状/尺寸相同。在一个示例中,在具有至少一个扩展抽头的ALF滤波器内部,用于一个或多个空域抽头的滤波器形状可如下所述。在一个示例中,用于空域抽头的滤波器形状可以是菱形形状。在一个示例中,用于空域抽头的滤波器形状可以是正方形形状。在一个示例中,用于空域抽头的滤波器形状可以是十字形。在一个示例中,用于空域抽头的滤波器形状可以是对称形状。在一个示例中,用于空域抽头的滤波器形状可以是非对称形状。在一个示例中,用于空域抽头的滤波器形状可以是其他设计形状。在一个示例中,可以即时确定/通过信号传输/推导用于空域抽头的滤波器形状。

[0324] 在一个示例中,在具有至少一个扩展抽头的ALF滤波器内部,用于一个或多个扩展抽头的滤波器形状可如下所述。在一个示例中,用于扩展抽头的滤波器形状可以是菱形形状。在一个示例中,用于扩展抽头的滤波器形状可以是正方形形状。在一个示例中,用于扩展抽头的滤波器形状可以是十字形。在一个示例中,用于扩展抽头的滤波器形状可以是对称形状。在一个示例中,用于扩展抽头的滤波器形状可以是非对称形状。在一个示例中,用于扩展抽头的滤波器形状可以是其他设计形状。在一个示例中,可以即时确定/通过信号传输/推导用于扩展抽头的滤波器形状。

[0325] 在一个示例中,包含至少一个空域抽头和至少一个扩展抽头的ALF滤波器可以设计如下。在一个示例中,一个或多个空域抽头可用于滤波(空域抽头可被视为使用ALF之前的重建作为输入)。图13至图14示出了空域抽头的示例形状。在一个示例中,可以按图13设计空域抽头的形状。在一个示例中,可以按图14设计空域抽头的形状。

[0326] 在一个示例中,一个或多个扩展抽头可应用于滤波。在一个示例中,可以如下设计具有一个或多个空域抽头和一个或多个扩展抽头的ALF滤波器。图15至图18示出了具有扩展抽头的示例ALF滤波器。在一个示例中,可以按图15设计具有空域抽头和扩展抽头的ALF滤波器。在一个示例中,可以按图16设计具有空域抽头和扩展抽头的ALF滤波器。在一个示例中,可以按图17设计具有空域抽头和扩展抽头的ALF滤波器。在一个示例中,可以按图18设计具有空域抽头和扩展抽头的ALF滤波器。

[0327] 在一个示例中,可应用基于几何信息的变换。在一个示例中,基于几何信息的变换可以独立地应用于一个或多个空域抽头。在一个示例中,基于几何信息的变换可以独立地应用于一个或多个扩展抽头。在一个示例中,基于几何信息的变换可以联合应用于一个或多个空域抽头和一个或多个扩展抽头。

[0328] 在一个示例中,一个或多个输入源可以用于一个或多个扩展抽头。在一个示例中,图15至图18的示例中所示的输入源A、B、C、D、E可以是不同的。在另一个示例中,图15至图18的示例中所示的输入源A、B、C、D、E可以是相同的。在一个示例中,一个或多个扩展抽头的一个可能输入源可以是ALF之前的重建。在一个示例中,一个或多个扩展抽头的一个可能输入源可以是DBF之前重建。在一个示例中,一个或多个扩展抽头的一个可能输入源可以由ALF之前的重建生成的中间滤波结果和ALF的经离线训练的滤波器。

[0329] 在一个示例中,可以应用特定的经离线训练的滤波器组。在一个示例中,可以即时通过信号传输/预定义/推导经离线训练的滤波器组指示符。在一个示例中,可以应用特定的经离线训练的滤波器分类器。在一个示例中,可以即时通过信号传输/预定义/推导经离线训练的滤波器分类器指示符。在一个示例中,一个或多个扩展抽头的一个可能输入源可以由DBF之前的重建生成的中间滤波结果和ALF的经离线训练的滤波器。

[0330] 在一个示例中,可以应用特定的经离线训练的滤波器组。在一个示例中,可以即时通过信号传输/预定义/推导经离线训练的滤波器组指示符。在一个示例中,可以应用特定的经离线训练的滤波器分类器。在一个示例中,可以即时通过信号传输/预定义/推导经离线训练的滤波器分类器指示符。在一个示例中,可以在APS中/即时预定义/推导用于输入源的指示符。

[0331] 在一个示例中,图15至图18中表示的ALF滤波器的输入源可以按顺序排列如下。在一个示例中,通过将ALF之前的重建馈送到分类器-0和与通过信号传输的滤波器组指示符相对应的经离线训练的滤波器组中而生成的中间滤波结果可以被应用为输入源A。在一个示例中,通过将ALF之前的重建馈送到分类器-1和与通过信号传输的滤波器组指示符相对应的经离线训练的滤波器组中而生成的中间滤波结果可以被应用为输入源B。在一个示例中,当前图片的DBF之前的重建可以被应用为输入源C。在一个示例中,通过将DBF之前的重建馈送到分类器-0和经离线训练的滤波器组-0中而生成的中间滤波结果可以被应用为输入源D。在一个示例中,通过将DBF之前的重建馈送到分类器-0和经离线训练的滤波器组-1中而生成的中间滤波结果可以被应用为输入源E。在一个示例中,前向参考图片列表(参考列表0)内部名为参考图像0的编解码图像可以被应用为输入源D。在一个示例中,后向参考图片列表(参考列表1)内部名为参考图像0的编解码图像可以被应用为输入源E。在一个示例中,通过将ALF之前的重建馈送到分类器-0和经离线训练的滤波器组-0中而生成的中间滤波结果可以被应用为输入源D。在一个示例中,通过将ALF之前的重建馈送到分类器-0和经离线训练的滤波器组-1中而生成的中间滤波结果可以被应用为输入源E。在一个示例中,通过将ALF之前的重建馈送到分类器-0和与通过信号传输的滤波器组指示符的相反指示符相对应的经离线训练的滤波器组中而生成的中间滤波结果可以被应用为输入源D。在一个示例中,通过将ALF之前的重建馈送到分类器-1和与通过信号传输的滤波器组指示符的相反指示符相对应的经离线训练的滤波器组中而生成的中间滤波结果可以被应用为输入源E。在一个示例中,可以基于形状、滤波器长度、对称约束联合推导ALF滤波器内部的扩展抽头的总数。

[0332] 在一个示例中,是否和/或如何应用ALF中的至少一个扩展抽头可取决于ALF中的分类。在一个示例中,分类可以基于输入源的梯度信息。在一个示例中,输入源可以是ALF之前的重建。在一个示例中,输入源可以是DBF之前的重建。在一个示例中,输入源可以由经离线训练的滤波器生成的中间滤波结果和ALF之前的重建。在一个示例中,输入源可以由经离线训练的滤波器生成的中间滤波结果和DBF之前的重建。在一个示例中,分类可以基于输入源的频带信息。在一个示例中,输入源可以是ALF之前的重建。在一个示例中,输入源可以是DBF之前的重建。在一个示例中,输入源可以由经离线训练的滤波器生成的中间滤波结果和ALF之前的重建。在一个示例中,输入源可以由经离线训练的滤波器生成的中间滤波结果和DBF之前的重建。

[0333] 在一个示例中,扩展抽头的输入样点可以被填充在图片/条带/片/CTU边界之间。在一个示例中,填充可在编码/解码过程期间应用于任何边界。在一个示例中,填充可应用于图片/子图片边界。在一个示例中,填充可应用于条带/片边界。在一个示例中,填充可应用于CTU/CTB边界。在一个示例中,填充可应用于CU/TU/PU边界。在一个示例中,填充可应用于块边界。在一个示例中,填充可应用于单元边界。在一个示例中,填充可应用于虚拟边界。

在一个示例中,填充可应用于任何其他类型的边界。在一个示例中,不同的填充方法可应用于边界。在一个示例中,可应用扩展填充。在一个示例中,可应用镜像填充。在一个示例中,可应用重复填充。在一个示例中,可应用其他填充方法。

[0334] 在一个示例中,可以通过信号传输第一语法元素以指示是否启用了具有至少一个扩展抽头的滤波器。在一个示例中,可以通过算术编解码对第一语法元素进行编解码。在一个示例中,第一语法元素可以用至少一个上下文来编解码。上下文可以取决于当前块或相邻块的编解码信息。上下文可以取决于至少一个相邻块的滤波形状。在一个示例中,可以用旁路编解码对第一语法元素进行编解码。在一个示例中,可以通过一元码、或舍位一元码、或定长码、或指数哥伦布码、舍位指数哥伦布码等对第一语法元素进行二进制定码。在一个示例中,可以有条件地通过信号传输第一语法元素。例如,仅当扩展抽头可用时才可以通过信号传输第一语法元素。可以预测方式对第一语法元素进行编解码。可以通过至少一个相邻块的扩展抽头的开/关决策来预测第一语法元素。可以针对不同的颜色分量独立地通过信号传输第一语法元素。在示例中,可以针对不同的颜色分量通过信号传输并共享第一语法元素。在示例中,可以针对第一颜色分量通过信号传输第一语法元素,但是针对第二颜色分量不通过信号传输第一语法元素。可在SPS/PPS/图片头/条带头/APS/CTU/CU/等中通过信号传输语法元素。

[0335] 在一个示例中,可以通过信号传输第一语法元素以指示哪些/何种输入源用于ALF滤波器内部的扩展抽头。在一个示例中,可以通过算术编解码对第一语法元素进行编解码。在一个示例中,第一语法元素可以用至少一个上下文来编解码。上下文可以取决于当前块或相邻块的编解码信息。上下文可以取决于至少一个相邻块的滤波形状。在一个示例中,可以用旁路编解码对第一语法元素进行编解码。在一个示例中,可以通过一元码、或舍位一元码、或定长码、或指数哥伦布码、舍位指数哥伦布码等对第一语法元素进行二进制定码。在一个示例中,可以有条件地通过信号传输第一语法元素。例如,仅当扩展抽头可用时才可以通过信号传输第一语法元素。可以预测方式对第一语法元素进行编解码。可以通过至少一个相邻块的扩展抽头的开/关决策来预测第一语法元素。可以针对不同的颜色分量独立地通过信号传输第一语法元素。在示例中,可以针对不同的颜色分量通过信号传输并共享第一语法元素。在示例中,可以针对第一颜色分量通过信号传输第一语法元素,但是针对第二颜色分量不通过信号传输第一语法元素。可在SPS/PPS/图片头/条带头/APS/CTU/CU/等中通过信号传输语法元素。在一个示例中,可以在用于通过信号传输的ALF滤波器的APS中通过信号传输语法元素。

[0336] 可以在语法元素结构(诸如APS)中通过信号传输ALF滤波器内部的至少一个扩展抽头的系数。在一个示例中,扩展抽头的系数可以包含在APS中。在一个示例中,扩展抽头的限幅参数可以包含在APS中。在一个示例中,扩展抽头的类合并结果可以包含在APS中。在一个示例中,可以预测方式对扩展抽头的系数进行编解码。在一个示例中,可以使用至少一个上下文通过算术编解码来对扩展抽头的系数进行编解码。在一个示例中,可使用旁路编解码来对扩展抽头的系数进行编解码。在一个示例中,扩展抽头的系数可以与空域抽头的系数联合编解码。在一个示例中,扩展抽头的其他参数可以包含在APS中。在一个示例中,扩展抽头的系数可以是预定义固定值。

[0337] 图19是示出可以在其中实施本文公开的各种技术的示例视频处理系统4000的框

图。各种实施方式可以包括系统4000的部分或全部组件。系统4000可以包括用于接收视频内容的输入4002。视频内容可以原始或未压缩的格式(例如,8位或10位多分量像素值)接收,或者可以压缩格式或编码格式接收。输入4002可以表示网络接口、外围总线接口或存储接口。网络接口的示例包括有线接口,诸如以太网、无源光网络(PON)等,以及无线接口,诸如Wi-Fi或蜂窝接口。

[0338] 系统4000可以包括编解码组件4004,其可以实施本文档中描述的各种编解码或编码方法。编解码组件4004可以降低从到编解码组件4004的输入4002到输出的视频的平均比特率,以产生视频的编解码表示。因此,编解码技术有时被称为视频压缩或视频转码技术。编解码组件4004的输出可以被存储或者经由所连接的通信发送,如组件4006所示。在输入4002处接收的视频的存储的或传送的比特流表示(或编解码表示)可以由组件4008使用以生成像素值或可显示视频,其被发送到显示接口4010。从比特流表示生成用户可视视频的过程有时被称为视频解压缩。此外,尽管某些视频处理操作被称为“编码”操作或工具,但是应当理解,编码工具或操作被用在编码器中,并且解码器将执行编码的反过程的对应解码工具或操作。

[0339] 外围总线接口或显示接口的示例可以包括通用串行总线(USB)或高清晰度多媒体接口(HDMI)或显示端口等。存储接口的示例包括串行高级技术附件(SATA)、外围组件互连(PCI)、集成驱动电子设备(IDE)接口等。本文档中描述的技术可以体现在各种电子设备(诸如移动电话、膝上型计算机、智能手机或能够执行数字数据处理和/或视频显示的其他设备)中。

[0340] 图20是示例视频处理设备4100的框图。设备4100可以用于实施本文中描述的一种或多种方法。设备4100可以实现为智能手机、平板计算机、计算机、物联网(IoT)接收器等。设备4100可以包括一个或多个处理器4102、一个或多个存储器4104和视频处理电路4106。处理器4102可以被配置为实施本文档中描述的一种或多种方法。存储器(多个存储器)4104可以用于存储用于实施本文描述的方法和技术的代码和数据。视频处理电路4106可以用于在硬件电路中实施本文档中描述的一些技术。在一些实施例中,视频处理电路4106可以至少部分地包括在处理器4102(例如,图形协处理器)中。

[0341] 图21是用于视频处理的示例方法4200的流程图。方法4200包括在步骤4202处确定将具有扩展抽头的ALF应用于视频中的图片。第二滤波器的中间滤波结果用作扩展抽头的输入。在步骤4204处基于ALF执行可视媒体数据与比特流之间的转换。根据示例,步骤4204的转换可以包括在编码器处进行编码或在解码器处进行解码。

[0342] 应当注意,方法4200可以在用于处理视频数据的设备中实施,该设备包括处理器和其上具有指令的非暂时性存储器,该设备,诸如,视频编码器4400、视频解码器4500和/或编码器4600。在这种情况下,该指令在由处理器执行时使处理器执行方法4200。此外,方法4200可以由包括供视频编解码设备使用的计算机程序产品的非暂时性计算机可读介质来执行。计算机程序产品包括存储在非暂时性计算机可读介质上的计算机可执行指令,使得当由处理器执行时使视频编解码设备执行方法4200。

[0343] 图22是示出可以利用本公开的技术的示例视频编解码系统4300的框图。视频编解码系统4300可以包括源设备4310和目标设备4320。源设备4310生成编码的视频数据,该源设备可以被称为视频编码设备。目标设备4320可以对由源设备4310生成的编码的视频数据

进行解码,该目标设备可以被称为视频解码设备。

[0344] 源设备4310可以包括视频源4312、视频编码器4314和输入/输出(I/O)接口4316。视频源4312可以包括诸如视频拍摄设备、用于从视频内容提供者接收视频数据的接口和/或用于生成视频数据的计算机图形系统的源,或此类源的组合。视频数据可以包括一个或多个图片。视频编码器4314对来自视频源4312的视频数据进行编码以生成比特流。比特流可以包括一系列的比特,这些比特形成视频数据的编解码表示。比特流可以包括编解码图片和相关联的数据。编解码图片是图片的编解码表示形式。关联数据可以包括序列参数集、图片参数集和其他语法结构。I/O接口4316可以包括调制器/解调器(调制解调器)和/或发送器。编码的视频数据可以通过网络4330经由I/O接口4316直接发送到目标设备4320。编码的视频数据还可以被存储到存储介质/服务器4340上以供目标设备4320访问。

[0345] 目标设备4320可以包括I/O接口4326、视频解码器4324和显示设备4322。I/O接口4326可以包括接收器和/或调制解调器。I/O接口4326可以从源设备4310或存储介质/服务器4340获取编码的视频数据。视频解码器4324可以对编码的视频数据进行解码。显示设备4322可以向用户显示解码的视频数据。显示设备4322可以与目标设备4320集成,或者可以在目标设备4320的外部,该目标设备可以被配置为与外部显示设备通过接口连接。

[0346] 视频编码器4314和视频解码器4324可以根据视频压缩标准操作,该视频压缩标准诸如HEVC标准、VVC标准和其他当前和/或进一步的标准。

[0347] 图23是示出视频编码器4400的示例的框图,该视频编码器可以是图22所示的系统4300中的视频编码器4314。视频编码器4400可以被配置为执行本公开的任何或全部技术。视频编码器4400包括多个功能组件。可以在视频编码器4400的各个组件之间共享本公开中描述的技术。在一些示例中,处理器可以被配置为执行本公开中描述的任何或所有技术。

[0348] 视频编码器4400的功能组件可以包括分割单元4401;预测单元4402,该预测单元可包括模式选择单元4403、运动估计单元4404、运动补偿单元4405和帧内预测单元4406;残差生成单元4407;变换处理单元4408;量化单元4409;反量化单元4410;逆变换单元4411;重建单元4412;缓存4413和熵编码单元4414。

[0349] 在其他示例中,视频编码器4400可以包括更多、更少或不同的功能组件。在一个示例中,预测单元4402可以包括帧内块复制(IBC)单元。IBC单元可以在IBC模式下执行预测,其中至少一个参考图片是当前视频块所在的图片。

[0350] 此外,一些组件,诸如运动估计单元4404和运动补偿单元4405,可以是高度集成的,但是在视频编码器4400的示例中出于解释目的而单独地示出。

[0351] 分割单元4401可以将图片分割成一个或多个视频块。视频编码器4400和视频解码器4500可以支持各种视频块尺寸。

[0352] 模式选择单元4403可以例如基于误差结果选择编解码模式中的一者,并将所得的帧内或帧间编解码块提供给残差生成单元4407以生成残差块数据并向重建单元4412提供以重建编解码块以用作参考图片。在一些示例中,模式选择单元4403可以选择帧内和帧间预测组合(CIIP)模式,其中该预测是基于帧间预测信号和帧内预测信号。模式选择单元4403还可以在帧间预测的情况下为块选择运动矢量的分辨率(例如,子像素或整数像素精度)。

[0353] 为了对当前视频块执行帧间预测,运动估计单元4404可以通过将来自缓存4413的

一个或多个参考帧与当前视频块进行比较来为当前视频块生成运动信息。运动补偿单元4405可以基于运动信息和来自缓存4413的图片(而不是与当前视频块关联的图片)的解码样点来确定用于当前视频块的预测视频块。

[0354] 运动估计单元4404和运动补偿单元4405可以对当前视频块执行不同的操作,例如,根据当前视频块是在I条带、P条带还是B条带中。

[0355] 在一些示例中,运动估计单元4404可以对当前视频块执行单向预测,并且运动估计单元4404可以在列表0或列表1的参考图片中搜索用于当前视频块的参考视频块。然后,运动估计单元4404可以生成参考索引,该索引指示列表0或列表1中的参考图片,该参考图片包含参考视频块和运动矢量,该运动矢量指示当前视频块与参考视频块之间的空间位移。运动估计单元4404可以输出参考索引、预测方向指示器和运动矢量作为当前视频块的运动信息。运动补偿单元4405可以基于由当前视频块的运动信息指示的参考视频块来生成当前块的预测视频块。

[0356] 在其他示例中,运动估计单元4404可以对当前视频块执行双向预测,运动估计单元4404可以在列表0中的参考图片中搜索当前视频块的参考视频块,并且还可以在列表1中的参考图片中搜索当前视频块的另一个参考视频块。然后,运动估计单元4404可以生成参考索引,该参考索引指示列表0和列表1中的参考图片,该参考图片包含参考视频块和运动矢量,该运动矢量指示参考视频块与当前视频块之间的空间位移。运动估计单元4404可以输出当前视频块的参考索引和运动矢量作为当前视频块的运动信息。运动补偿单元4405可以基于由当前视频块的运动信息指示的参考视频块来生成当前视频块的预测视频块。

[0357] 在一些示例中,运动估计单元4404可以输出一组完整的运动信息以用于解码器的解码处理。在一些示例中,运动估计单元4404可以不输出用于当前视频的一组完整的运动信息。更确切地,运动估计单元4404可以参考另一个视频块的运动信息来通过信号传输当前视频块的运动信息。例如,运动估计单元4404可以确定当前视频块的运动信息与邻近视频块的运动信息充分相似。

[0358] 在一个示例中,运动估计单元4404可以在与当前视频块关联的语法结构中指示某个值,该值向视频解码器4500指示当前视频块与另一个视频块具有相同的运动信息。

[0359] 在另一个示例中,运动估计单元4404可以在与当前视频块关联的语法结构中识别另一个视频块和运动矢量差(MVD)。运动矢量差指示当前视频块的运动矢量与指示的视频块的运动矢量之间的差。视频解码器4500可以使用指示的视频块的运动矢量和运动矢量差来确定当前视频块的运动矢量。

[0360] 如上文所讨论,视频编码器4400可以预测地通过信号传输运动矢量。可以由视频编码器4400实施的预测信令技术的两个示例包括高级运动矢量预测(AMVP)和Merge模式信令。

[0361] 帧内预测单元4406可以对当前视频块执行帧内预测。当帧内预测单元4406对当前视频块执行帧内预测时,帧内预测单元4406可以基于同一图片中的其他视频块的解码样点作为当前视频块生成预测数据。当前视频块的预测数据可以包括预测的视频块和各种语法元素。

[0362] 残差生成单元4407可以通过从当前视频块中减去当前视频块的预测视频块来为当前视频块生成残差数据。当前视频块的残差数据可以包括与当前视频块中的样点的不同

样点分量相对应的残差视频块。

[0363] 在其他示例中,例如在跳过模式下,对于当前视频块的当前视频块可以没有残差数据,并且残差生成单元4407可以不执行减法运算。

[0364] 变换处理单元4408可以通过对与当前视频块关联的残差视频块应用一个或多个变换来生成用于当前视频块的变换系数视频块。

[0365] 在变换处理单元4408生成与当前视频块关联的变换系数视频块之后,量化单元4409可以基于与当前视频块关联的一个或多个量化参数(QP)值来量化与当前视频块关联的变换系数视频块。

[0366] 反量化单元4410和逆变换单元4411可以分别对变换系数视频块应用反量化和逆变换,以从变换系数视频块重建残差视频块。重建单元4412可以将重建的残差视频块添加到来自预测单元4402生成的一个或多个预测视频块的对应样点,以产生与当前块关联的重建视频块,以便存储在缓存4413中。

[0367] 在重建单元4412重建视频块之后,可以执行环路滤波操作,以减少视频块中的视频方块伪影。

[0368] 熵编码单元4414可以从视频编码器4400的其他功能组件接收数据。当熵编码单元4414接收数据时,熵编码单元4414可以执行一个或多个熵编码操作以生成熵编码数据并输出包括熵编码数据的比特流。

[0369] 图24是示出视频解码器4500的示例的框图,该视频解码器可以是图22所示的系统4300中的视频解码器4324。视频解码器4500可以被配置为执行本公开的任何或全部技术。在所示的示例中,视频解码器4500包括多个功能组件。可以在视频解码器4500的各个组件之间共享本公开中描述的技术。在一些示例中,处理器可以被配置为执行本公开中描述的任何或所有技术。

[0370] 在所示的示例中,视频解码器4500包括熵解码单元4501、运动补偿单元4502、帧内预测单元4503、反量化单元4504、逆变换单元4505,以及重建单元4506、缓存4507。在一些示例中,视频解码器4500可以执行与关于视频编码器4400描述的编码过程大致互逆的解码过程。

[0371] 熵解码单元4501可以取出编码的比特流。编码的比特流可以包括熵编解码的视频数据(例如,编码的视频数据块)。熵解码单元4501可以对熵编解码的视频数据进行解码,并且运动补偿单元4502可以根据熵解码的视频数据确定运动信息,该运动信息包括运动矢量、运动矢量精度、参考图片列表索引以及其他运动信息。运动补偿单元4502可以例如通过执行AMVP和Merge模式来确定此类信息。

[0372] 运动补偿单元4502可以产生运动补偿块,从而可能基于插值滤波器执行插值。以子像素精度使用的插值滤波器的标识可以包括在语法元素中。

[0373] 运动补偿单元4502可以使用由视频编码器4400在视频块的编码期间使用的插值滤波器来计算参考块的亚整数像素的插值。运动补偿单元4502可以根据接收到的语法信息确定由视频编码器4400使用的插值滤波器,并使用插值滤波器来产生预测块。

[0374] 运动补偿单元4502可以使用一些语法信息来确定用于对编码的视频序列的帧和/或条带进行编码的块的大小、描述编码的视频序列的图像的每个宏块如何分割的分割信息、指示如何对每个分割进行编码的模式、每个帧间编解码块的一个或多个参考帧(和参考

帧列表),以及用于对编码的视频序列进行解码的其他信息。

[0375] 帧内预测单元4503可以使用例如在比特流中接收的帧内预测模式,从空间上相邻的块形成预测块。反量化单元4504对提供在比特流中并由熵解码单元4501解码量化的视频块系数进行反量化,即,去量化。逆变换单元4505应用逆变换。

[0376] 重建单元4506可以将残差块与由运动补偿单元4502或帧内预测单元4503生成的对应预测块相加,以形成解码块。如果需要,还可以应用去方块滤波器来对解码块进行滤波以消除方块效应。然后,解码的视频块被存储在缓存4507中,该缓存为后续运动补偿/帧内预测提供参考块,并且还产生解码视频以呈现在显示设备上。

[0377] 图25是示例编码器4600的示意图。编码器4600适合于实施VVC技术。编码器4600包括三个环路滤波器,即,去方块滤波器(DF) 4602、样点自适应偏移(SAO) 4604和自适应环路滤波器(ALF) 4606。与使用预定义滤波器的DF 4602不同,SAO 4604和ALF 4606利用当前图片的原始样点以通过分别添加偏移和应用有限脉冲响应(FIR)滤波器并利用编解码的边信息通过信号传输偏移和滤波器系数来减小原始样点与重建样点之间的均方误差。ALF 4606位于每张图片的最后处理阶段,并且可以被视为试图捕捉和修复由先前阶段创建的伪影的工具。

[0378] 编码器4600还包括被配置为接收输入视频的帧内预测组件4608和运动估计/补偿(ME/MC)组件4610。帧内预测组件4608被配置为执行帧内预测,而ME/MC组件4610被配置为利用从参考图片缓存4612获取的参考图片来执行帧间预测。来自帧间预测或帧内预测的残差块被馈送到变换(T)组件4614和量化(Q)组件4616中,以生成量化的残差变换系数,这些系数被馈送到熵编解码组件4618中。熵编解码组件4618对预测结果和量化的变换系数进行熵编解码,并将其发送到视频解码器(未示出)。从量化组件4616输出的量化分量可以被馈送到反量化(IQ)组件4620、逆变换组件4622和重建(REC)组件4624。REC组件4624能够将图像输出到DF 4602、SAO 4604和ALF 4606,以便在这些图片被存储在参考图片缓存4612中之前进行滤波。

[0379] 接下来提供由一些示例优选的解决方案的列表。

[0380] 以下解决方案示出了本文讨论的技术的示例。

[0381] 1.一种用于处理视频数据的方法,其包括:确定将具有扩展抽头的自适应环路滤波器(ALF)应用于视频中的图片,其中第二滤波器的中间滤波结果用作该扩展抽头的输入;以及基于该ALF执行可视媒体数据与比特流之间的转换。

[0382] 2.如解决方案1所述的方法,其中离线训练的ALF、经在线训练的ALF、预定义滤波器或其他经在线训练的滤波器的中间滤波结果用作该扩展抽头的输入。

[0383] 3.如解决方案1至2中任一项所述的方法,其中该中间滤波结果是通过该ALF之前的重建和该经离线训练的ALF生成的。

[0384] 4.如解决方案1至3中任一项所述的方法,其中该中间滤波结果是通过去方块滤波器DBF之前的重建和该经离线训练的ALF生成的。

[0385] 5.如解决方案1至4中任一项所述的方法,其中应用高斯滤波器、双边滤波器、导向滤波器、中值滤波器、具有低通属性的滤波器或具有高通属性的滤波器。

[0386] 6.如解决方案1至5中任一项所述的方法,其中经在线训练的ALF滤波器的中间滤波结果用作该扩展抽头的输入。

[0387] 7. 如解决方案1至6中任一项所述的方法,其中用于中间滤波结果的输入包括不同编解码阶段处的重建样点。

[0388] 8. 如解决方案1至7中任一项所述的方法,其中该中间滤波结果通过以下操作生成:参考帧的ALF之前的重建、参考帧的样点自适应偏移(SAO)或跨分量SAO(CCSAO)之前的重建、参考帧的双边滤波器(BIF)之前的重建、或参考帧的去方块滤波器(DBF)之前的重建。

[0389] 9. 如解决方案1至8中任一项所述的方法,其中该中间滤波结果通过以下操作生成:参考帧的ALF之后的重建、参考帧的SAO或CCSAO之后的重建、参考帧的BIF之后的重建、或参考帧的DBF之后的重建。

[0390] 10. 如解决方案1至9中任一项所述的方法,其中该扩展抽头接收来自以下各者的输入:当前帧的ALF之前的重建、当前帧的样点自适应偏移(SAO)或跨分量SAO(CCSAO)之前的重建、或当前帧的双边滤波器(BIF)之前的重建。

[0391] 11. 如解决方案1至10中任一项所述的方法,其中该扩展抽头接收来自以下各者的输入:当前帧的ALF之后的重建、当前帧的SAO或CCSAO之后的重建、或当前帧的BIF之后的重建。

[0392] 12. 如解决方案1至11中任一项所述的方法,其中编解码图片内部的样点用作该扩展抽头的输入源。

[0393] 13. 如解决方案1至12中任一项所述的方法,其中与当前块、当前条带或当前帧相关联的参考图片列表(RPL)或参考图片集(RPS)中的参考帧内部的样点用作该扩展抽头的输入源。

[0394] 14. 如解决方案1至13中任一项所述的方法,其中该参考帧是当前图片、当前条带或当前块的长期参考图片或短期参考图片。

[0395] 15. 如解决方案1至14中任一项所述的方法,其中解码图片缓存区中的帧内部的样点用作该扩展抽头的输入源。

[0396] 16. 如解决方案1至15中任一项所述的方法,其中通过信号传输指示符以指示包含用作该扩展抽头的输入源的样点的编解码图片。

[0397] 17. 如解决方案1至16中任一项所述的方法,其中该指示符指示参考图片列表或参考索引,并且其中根据参考图片列表中的参考图片的数量或解码图片缓存区中的解码图片的数量有条件地通过信号传输该指示符。

[0398] 18. 如解决方案1至17中任一项所述的方法,其中编解码图片包含用作该扩展抽头的输入源的样点,并且其中该编解码图片是被动态确定的。

[0399] 19. 如解决方案1至18中任一项所述的方法,其中该扩展抽头从解码图片缓存区中的帧、参考图片列表0中的帧、参考图片列表1中的帧、最接近当前帧的参考帧、具有特定索引的参考帧、同位帧或它们的组合获取输入。

[0400] 20. 如解决方案1至19中任一项所述的方法,其中基于解码信息来确定用作该扩展抽头的输入的帧。

[0401] 21. 如解决方案1至20中任一项所述的方法,其中是否从先前编解码的帧中获取信息以用作该扩展抽头的输入源取决于待滤波块的至少一个区域的解码信息。

[0402] 22. 如解决方案1至21中任一项所述的方法,其中语法元素指示是否从先前编解码的帧中获取信息以用作该扩展抽头的输入源。

[0403] 23.如解决方案1至22中任一项所述的方法,其中是否从先前编解码的帧中获取信息以用作该扩展抽头的输入源取决于条带类型或图片类型。

[0404] 24.如解决方案1至23中任一项所述的方法,其中是否从先前编解码的帧中获取信息以用作该扩展抽头的输入源取决于参考图片信息或解码图片缓存区中的图片信息。

[0405] 25.如解决方案1至24中任一项所述的方法,其中是否从先前编解码的帧中获取信息以用作该扩展抽头的输入源取决于时间层索引、量化参数或图片的尺寸。

[0406] 26.如解决方案1至25中任一项所述的方法,其中当块包含以非帧间模式编解码的样点时,该扩展抽头不使用来自先前编解码的帧的信息来对该块进行滤波。

[0407] 27.如解决方案1至26中任一项所述的方法,其中当前块与匹配块之间的失真用于确定是否使用来自先前编解码的帧的信息来对该当前块进行滤波。。

[0408] 28.如解决方案1至27中任一项所述的方法,其中该信息包含当前块的两个参考块或同位块,其中一个块来自列表-0中的第一参考帧,并且另一个块来自列表-1中的第一参考帧。

[0409] 29.一种用于处理视频数据的装置,其包括:处理器;以及上面具有指令的非暂时性存储器,其中所述指令在由所述处理器执行时使所述处理器执行如解决方案1至28中任一项所述的方法。

[0410] 30.一种非暂时性计算机可读介质,其包括供视频编解码设备使用的计算机程序产品,所述计算机程序产品包括存储在所述非暂时性计算机可读介质上的计算机可执行指令,使得在由处理器执行时使所述视频编解码设备执行如解决方案1至28中任一项所述的方法。

[0411] 31.一种非暂时性计算机可读记录介质,其存储由通过视频处理装置执行的方法生成的视频的比特流,其中所述方法包括:确定在自适应环路滤波器(ALF)中应用扩展抽头;以及基于所述确定生成所述比特流。

[0412] 32.一种用于存储视频的比特流的方法,其包括:确定在自适应环路滤波器(ALF)中应用扩展抽头;基于所述确定生成所述比特流;以及将所述比特流存储在非暂时性计算机可读记录介质中。

[0413] 33.一种在本专利文档中描述的方法、装置或系统。

[0414] 在本文中描述的解决方案中,编码器可以通过根据格式规则产生编解码表示来符合格式规则。在本文中描述的解决方案中,解码器可以使用格式规则来解析编解码表示中的语法元素,并根据格式规则已知语法元素的存在和不存在,以产生解码的视频。

[0415] 在本文档中,术语“视频处理”可以指代视频编码、视频解码、视频压缩或视频解压缩。例如,视频压缩算法可以在从视频的像素表示转换为对应的比特流表示(反之亦然)期间应用。例如,当前视频块的比特流表示可以对应于同位或分布在比特流中的不同位置处的位,如语法所定义的那样。例如,宏块可以根据变换和编码的错误残差值进行编码,并且还使用标头中的位和比特流中的其他字段进行编码。此外,在转换期间,解码器可基于确定来解析比特流,并已知某些字段可能存在或不存在,如上述解决方案中所描述。类似地,编码器可以确定是否包括某些语法字段,并通过在编码表示中包括或排除语法字段来相应地生成编码表示。

[0416] 本文档中描述的所公开的和和其他解决方案、示例、实施例、模块和功能操作可以在

数字电子电路或计算机软件、固件或硬件或其中一者或多者的组合中实现,该计算机软件、固件或硬件包括本文档中公开的结构及其等同结构。所公开的和他实施例可以被实现为被编码在计算机可读介质上以由数据处理装置执行或控制数据处理装置的操作的一个或多个计算机程序产品,即,计算机程序指令的一个或多个模块。该计算机可读介质可以是机器可读存储设备、机器可读存储基板、存储器设备、影响机器可读传播信号的物质组合物或其中一者或多者的组合。术语“数据处理装置”涵盖用于处理数据的所有装置、设备和机器,包括例如可编程处理器、计算机或多个处理器或计算机。除了硬件之外,该装置还可以包括为所讨论的计算机程序创建执行环境的代码,例如,构成处理器固件、协议栈、数据库管理系统、操作系统、或其一个或多个的组合的代码。传播信号是人为生成的信号,例如,机器生成的电、光或电磁信号,其经生成以对信息进行编码以便发送到合适的接收器装置。

[0417] 计算机程序(也被称为程序、软件、软件应用程序、脚本或代码)可以任何形式的编程语言(包括编译语言或解释语言)编写,并且它可以任何形式(包括作为独立程序或作为模块、组件、子程序或适于用于计算环境中的任何其他单元)进行部署。计算机程序不一定与文件系统中的文件相对应。程序可以存储在保存其他程序或数据的文件的一部分中(例如,存储在标记语言文档中的一个或多个脚本)、专用于所讨论程序的单个文件中、或多个协调的文件中(例如,存储一个或多个模块、子程序或代码的部分的文件)。计算机程序可以被部署为在一个计算机或多个计算机上执行,该多个计算机位于一个站点处或跨多个站点分布并由通信网络互连。

[0418] 本文档中描述的过程或逻辑流程可以由执行一个或多个计算机程序的一个或多个可编程处理器来执行以便通过操作输入数据并生成输出来执行功能。过程和逻辑流程还可以由专用逻辑电路(例如现场可编程门阵列(FPGA)或专用集成电路(ASIC))执行,并且设备还可以被实现为专用逻辑电路(例如现场可编程门阵列(FPGA)或专用集成电路(ASIC))。

[0419] 适用于执行计算机程序的处理器包含,例如,通用及专用微处理器两者及任何种类的数字计算机的任何一个或多个处理器。通常,处理器将从只读存储器或随机存取存储器或两者接收指令及数据。计算机的基本元件是用于执行指令的处理器以及用于存储指令和数据的一个或多个存储器设备。通常,计算机还将包括用于存储数据的一个或多个大容量存储设备(例如,磁盘、磁光盘或光盘),或者可操作地耦合以从一个或多个大容量存储设备接收数据或将数据传递到一个或多个大容量存储设备或这两者。然而,计算机不必具有此类设备。适合于存储计算机程序指令和数据的计算机可读介质包括所有形式的非易失性存储器、介质和存储器设备,包括例如半导体存储器设备,例如,可擦可编程只读存储器(EPROM)、电可擦可编程只读存储器(EEPROM)和闪存存储器设备;磁盘,例如内部硬盘或可移动磁盘;磁光盘;以及光盘只读存储器(CD ROM)和数字多功能光盘只读存储器(DVD-ROM)磁盘。处理器和存储器可以由专用逻辑电路补充或结合在其中。

[0420] 尽管本公开包含许多细节,但是这些细节不应被解释为对任何主题范围或可能要求保护的内容的限制,而应被解释为对可能特定于特定技术的特定实施例的特征的描述。在本公开中在单独的实施例的背景下描述的某些特征还可以在单个实施例中组合地实现。相反地,在单个实施例的背景下描述的几个特征还可以在多个实施例中单独地或者以任何适当的子组合来实现。此外,尽管上文可能将特征描述为以某些组合来起作用以及甚至最初如此要求保护,但是在一些情况下,来自所要求保护的组合的一个或多个特征可以从该

组合中去除,以及所要求保护的组合可以涉及子组合或者子组合的变形。

[0421] 类似地,尽管在图中以特定的次序示出了操作,但是这并不应当被理解为要求这样的操作以所示出的特定次序或者以顺序次序来执行或者执行所有示出的操作来实现期望的结果。此外,本公开中描述的各种系统组件的分离不应被理解为要求在所有实施例中进行这种分离。

[0422] 仅描述了几个实施方式和示例,并且可以基于本公开中描述和示出的内容做出其他实施方式、增强和更改。

[0423] 当除了第一组件与第二组件之间的线、迹线或另一种介质之外,不存在中间组件时,第一组件直接耦合到第二组件。当在第一组件与第二组件之间存在除了线、迹线或另一种介质之外的中间组件时,第一组件间接耦合到第二组件。术语“耦合”及其变体包括直接耦合和间接耦合。除非另有说明,否则术语“约”的使用意味着包括后续数值的 $\pm 10\%$ 的范围。

[0424] 虽然本公开中提供了若干实施例,但是应当理解,在不脱离本公开的精神或范围的情况下,所公开的系统和方法可被体现为许多其他特定形式。本示例被认为是说明性的而不是限制性的,并且意图不限于本文给出的细节。例如,各种元件或组件可以组合或集成在另一个系统中,或者可以省略或不实现某些特征。

[0425] 另外,在不脱离本公开的范围的情况下,各种实施例中描述和示出为离散或单独的技术、系统、子系统和方法可与其他系统、模块、技术或方法组合或集成。被示出或讨论为耦合的其他项目可以直接连接,或者可以通过一些接口、设备或中间组件间接耦合或通信,而不论是电、机械还是其他耦合或通信。本领域技术人员可确认改变、替换和变更的其他示例,并且可以在不脱离本文公开的精神和范围的情况下进行这些改变、替换和变更。

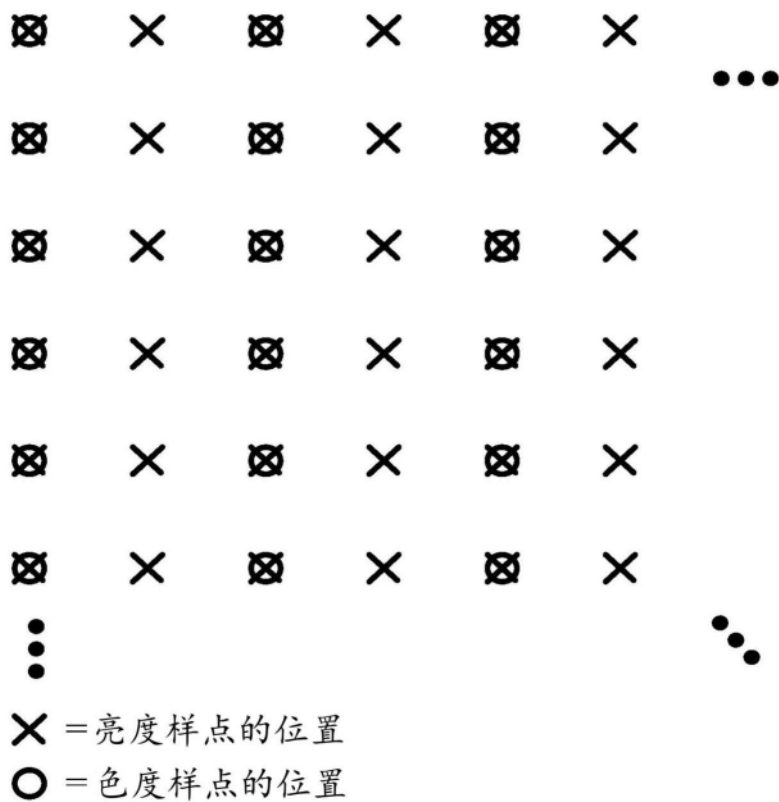


图1

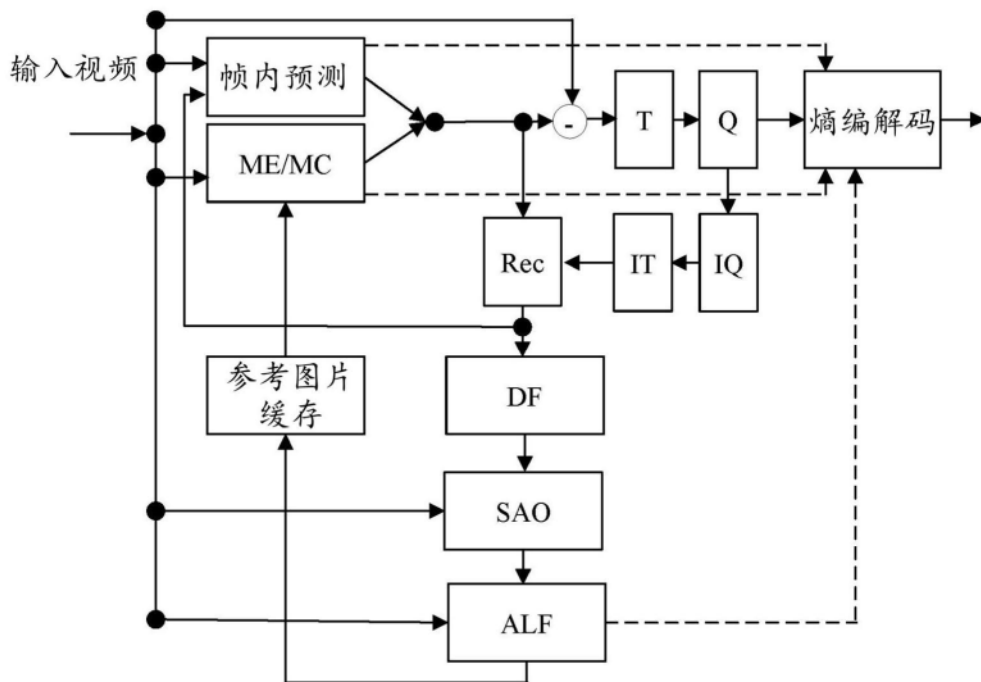


图2

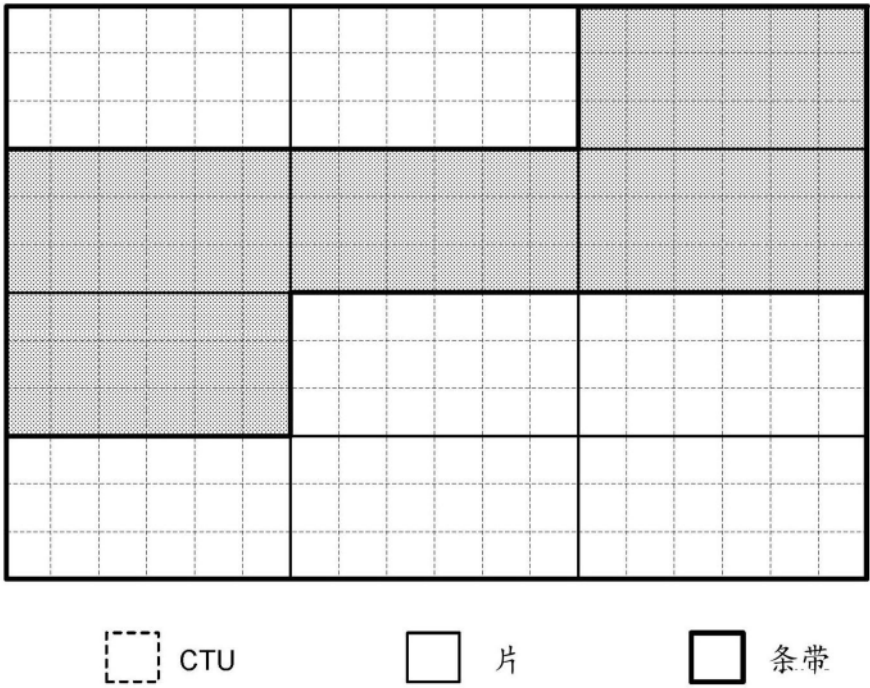


图3

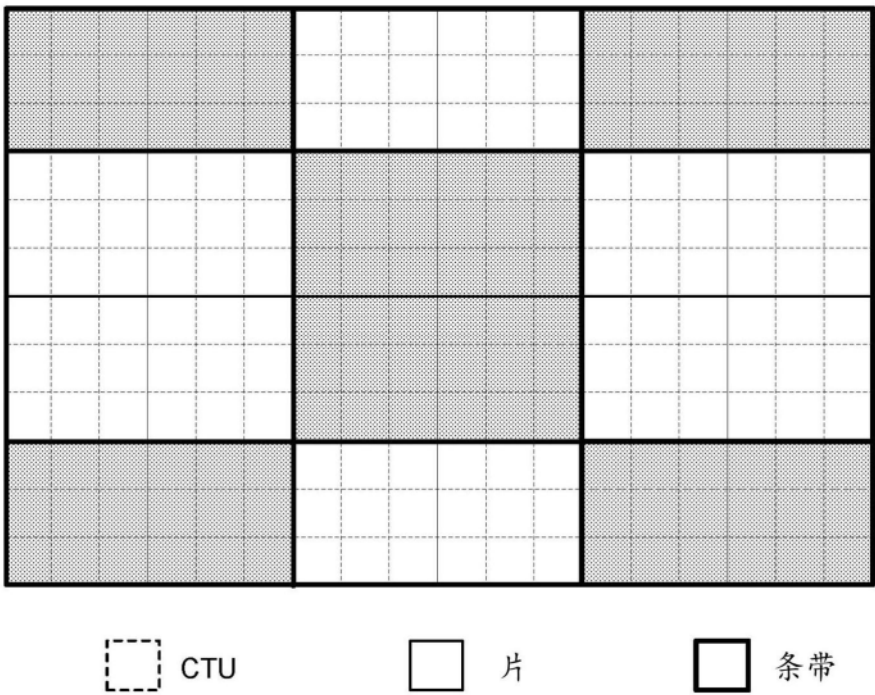


图4

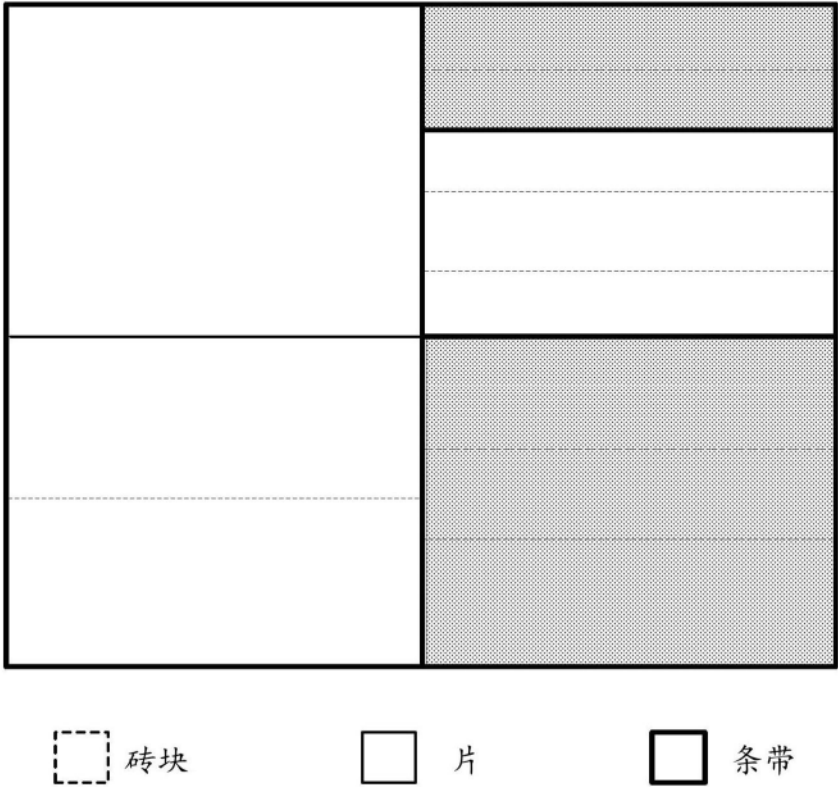


图5

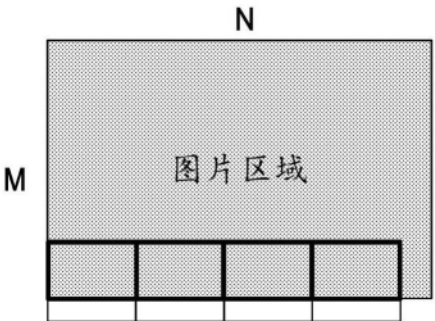


图6A

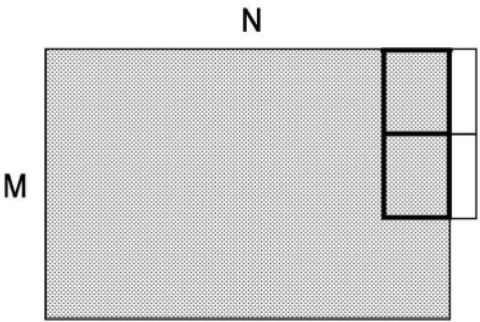


图6B

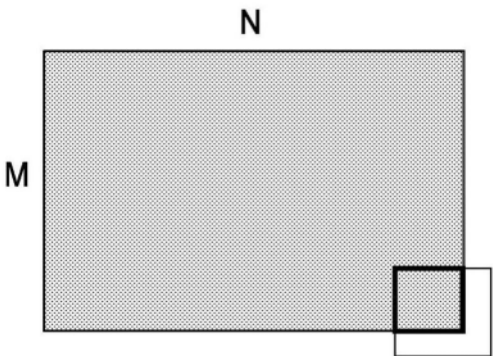


图6C

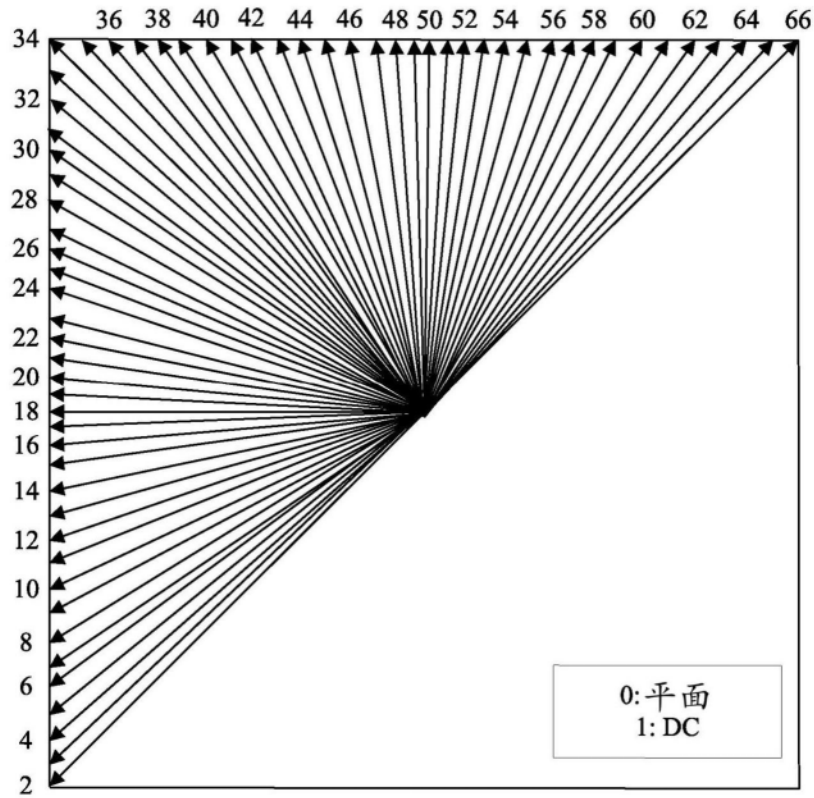


图7

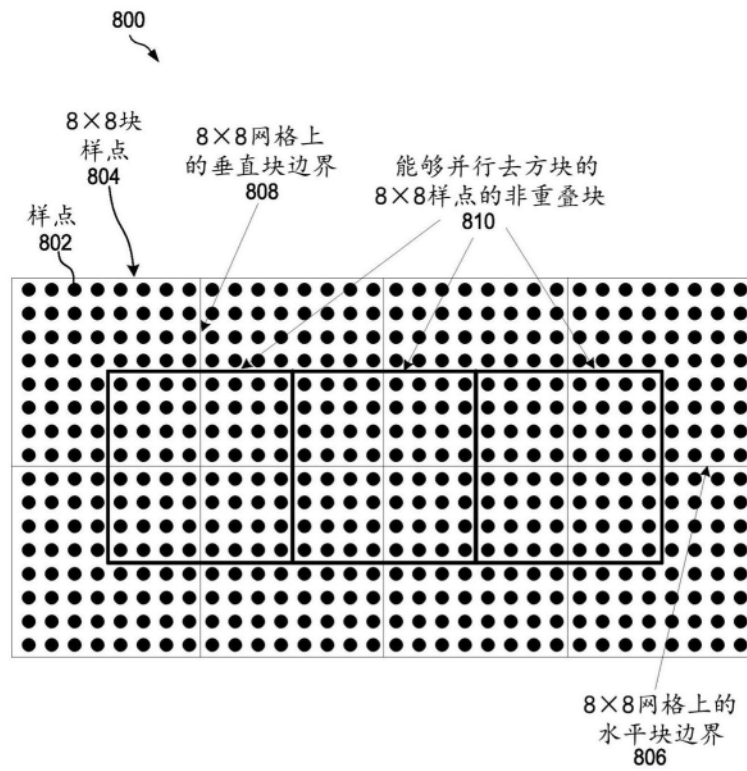


图8

|                 |                 |                 |                 |                 |                 |                 |                 |
|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|
| p3 <sub>0</sub> | p2 <sub>0</sub> | p1 <sub>0</sub> | p0 <sub>0</sub> | q0 <sub>0</sub> | q1 <sub>0</sub> | q2 <sub>0</sub> | q3 <sub>0</sub> |
| p3 <sub>1</sub> | p2 <sub>1</sub> | p1 <sub>1</sub> | p0 <sub>1</sub> | q0 <sub>1</sub> | q1 <sub>1</sub> | q2 <sub>1</sub> | q3 <sub>1</sub> |
| p3 <sub>2</sub> | p2 <sub>2</sub> | p1 <sub>2</sub> | p0 <sub>2</sub> | q0 <sub>2</sub> | q1 <sub>2</sub> | q2 <sub>2</sub> | q3 <sub>2</sub> |
| p3 <sub>3</sub> | p2 <sub>3</sub> | p1 <sub>3</sub> | p0 <sub>3</sub> | q0 <sub>3</sub> | q1 <sub>3</sub> | q2 <sub>3</sub> | q3 <sub>3</sub> |
| p3 <sub>4</sub> | p2 <sub>4</sub> | p1 <sub>4</sub> | p0 <sub>4</sub> | q0 <sub>4</sub> | q1 <sub>4</sub> | q2 <sub>4</sub> | q3 <sub>4</sub> |
| p3 <sub>5</sub> | p2 <sub>5</sub> | p1 <sub>5</sub> | p0 <sub>5</sub> | q0 <sub>5</sub> | q1 <sub>5</sub> | q2 <sub>5</sub> | q3 <sub>5</sub> |
| p3 <sub>6</sub> | p2 <sub>6</sub> | p1 <sub>6</sub> | p0 <sub>6</sub> | q0 <sub>6</sub> | q1 <sub>6</sub> | q2 <sub>6</sub> | q3 <sub>6</sub> |
| p3 <sub>7</sub> | p2 <sub>7</sub> | p1 <sub>7</sub> | p0 <sub>7</sub> | q0 <sub>7</sub> | q1 <sub>7</sub> | q2 <sub>7</sub> | q3 <sub>7</sub> |

图9

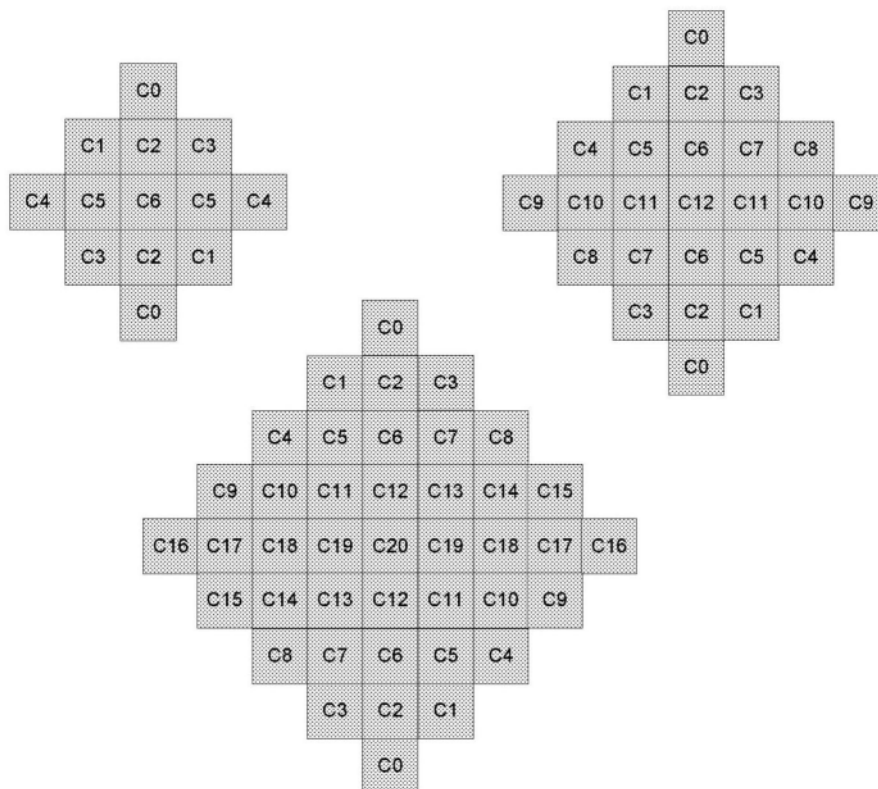


图10

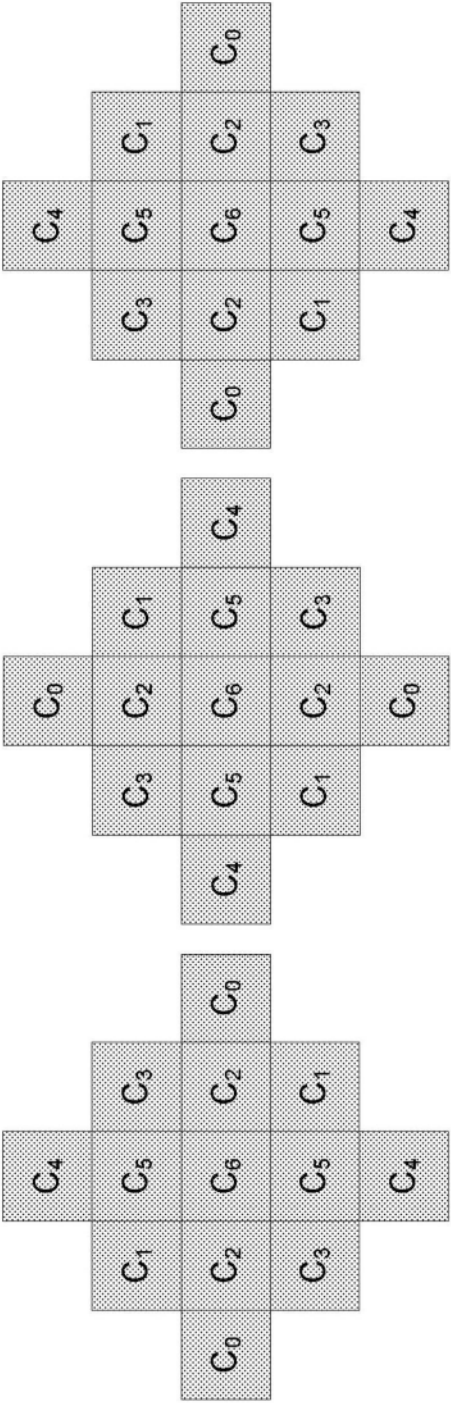


图11

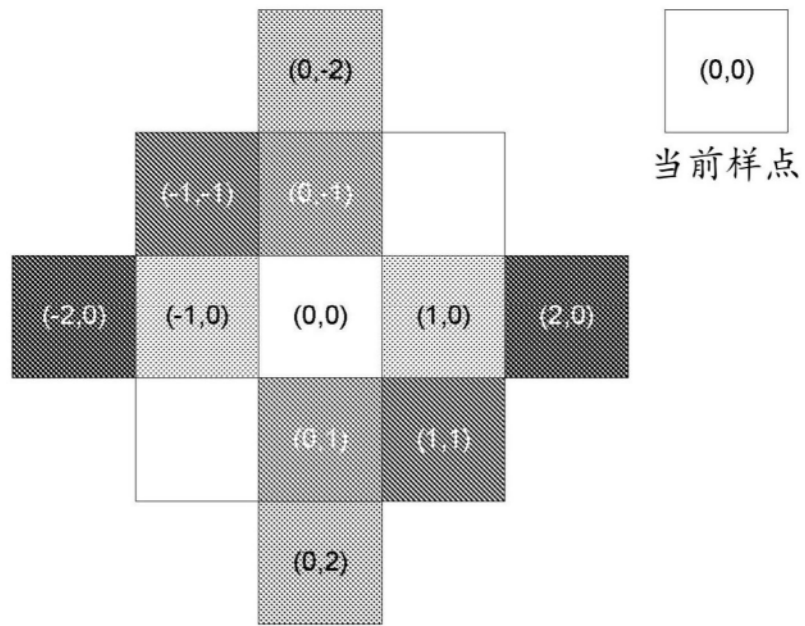


图12

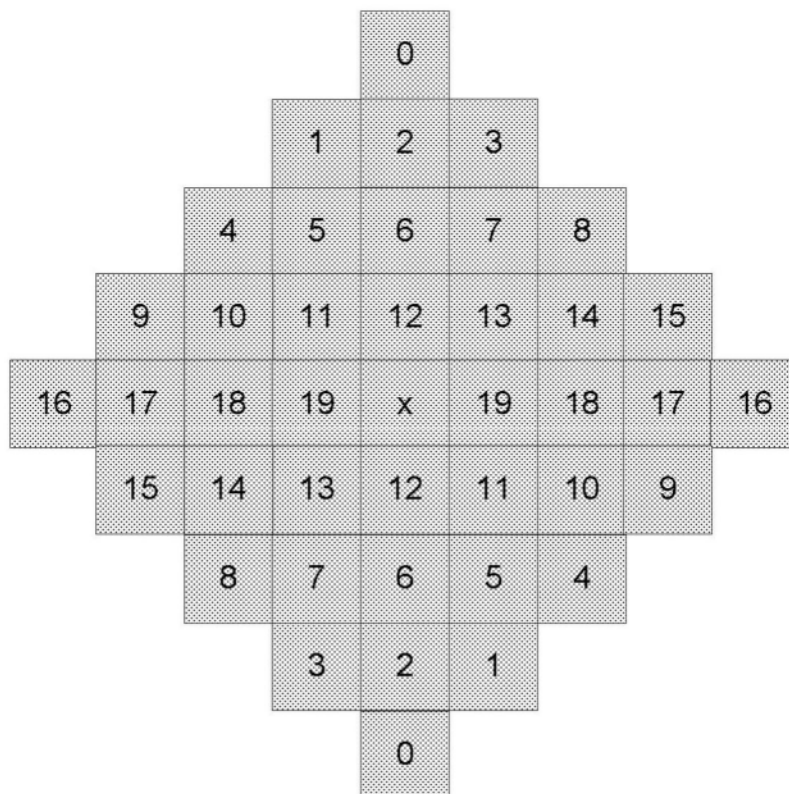


图13

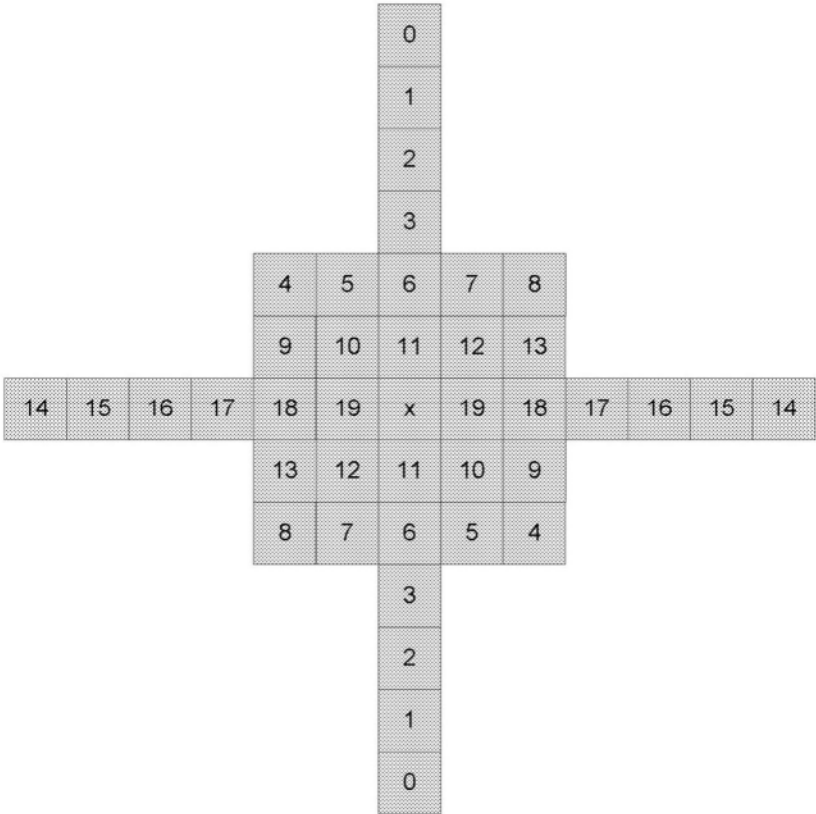


图14

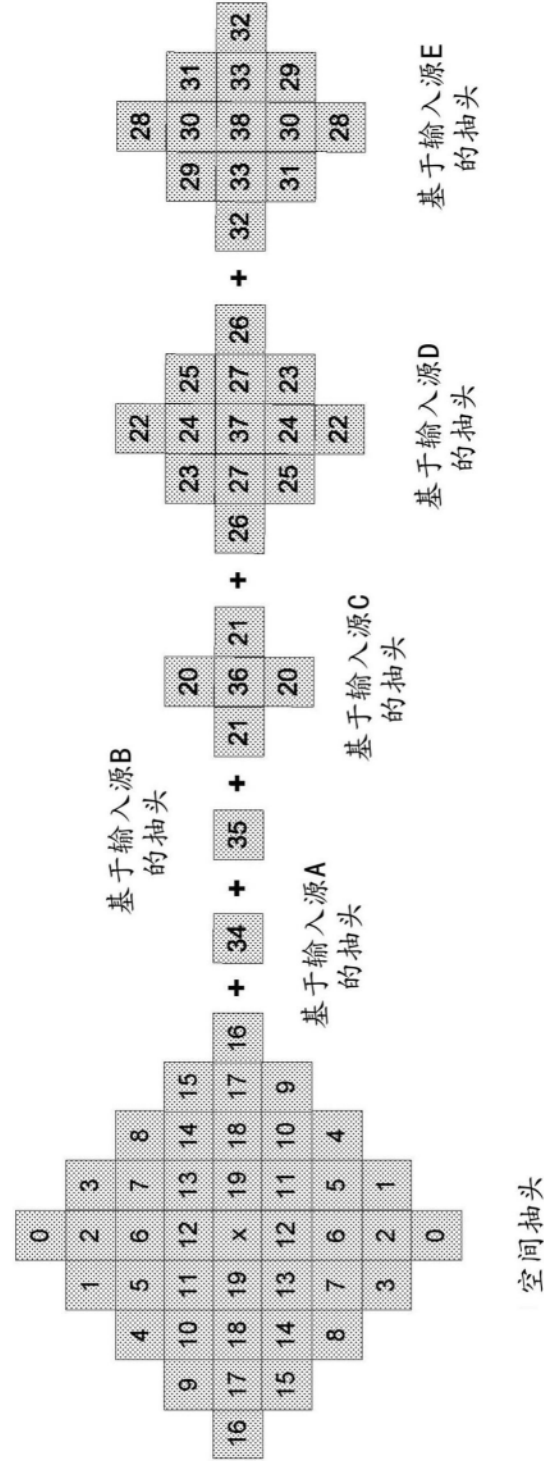


图15

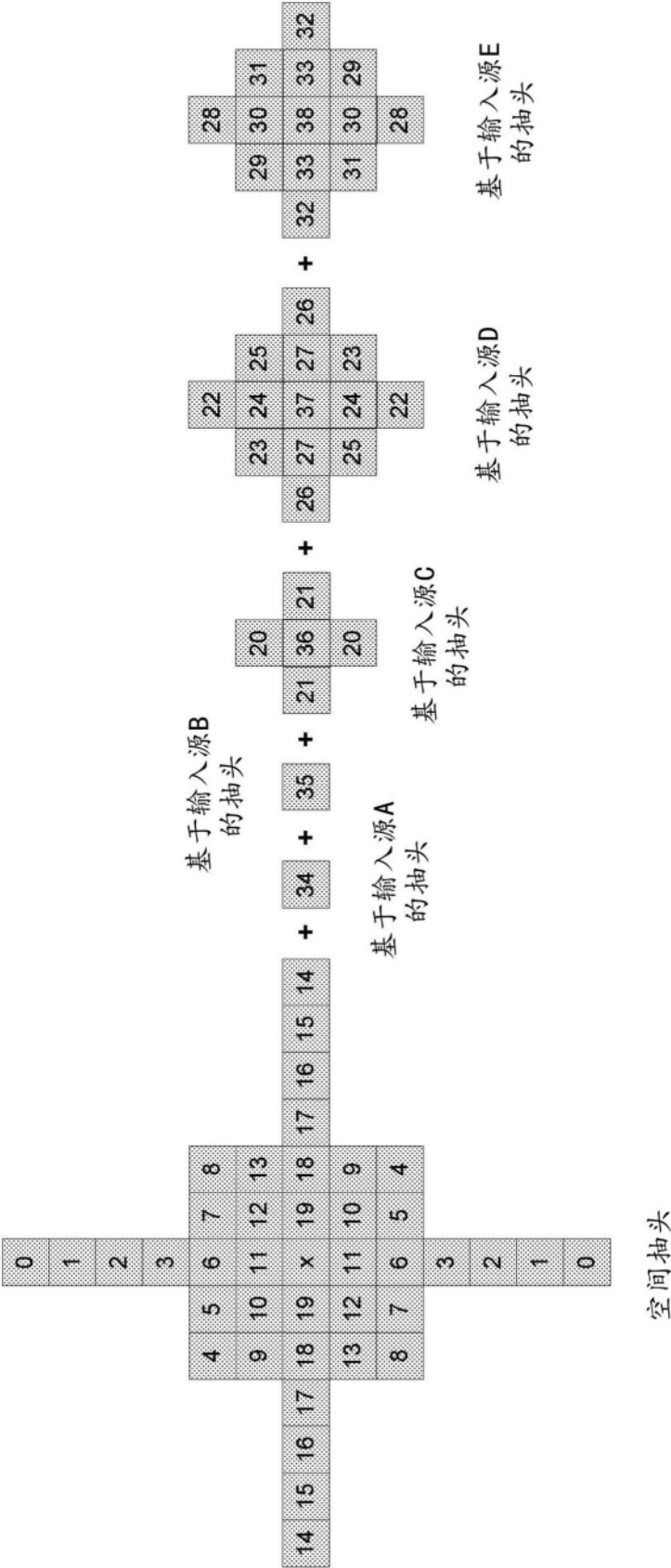


图16

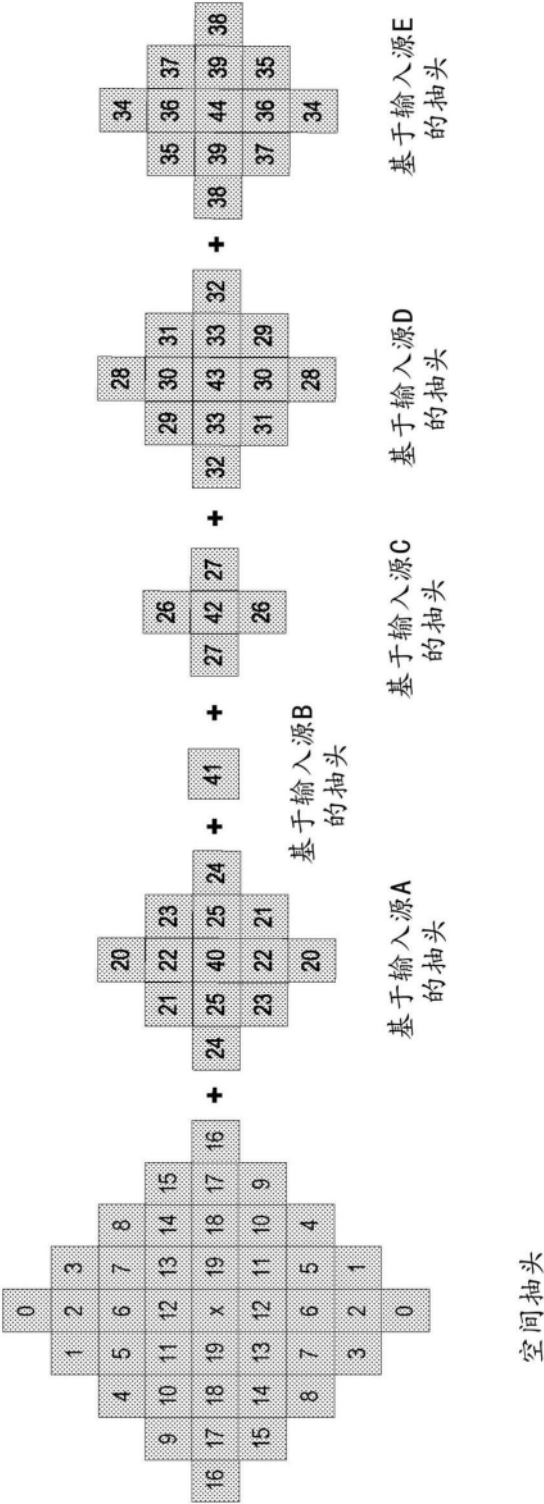


图17

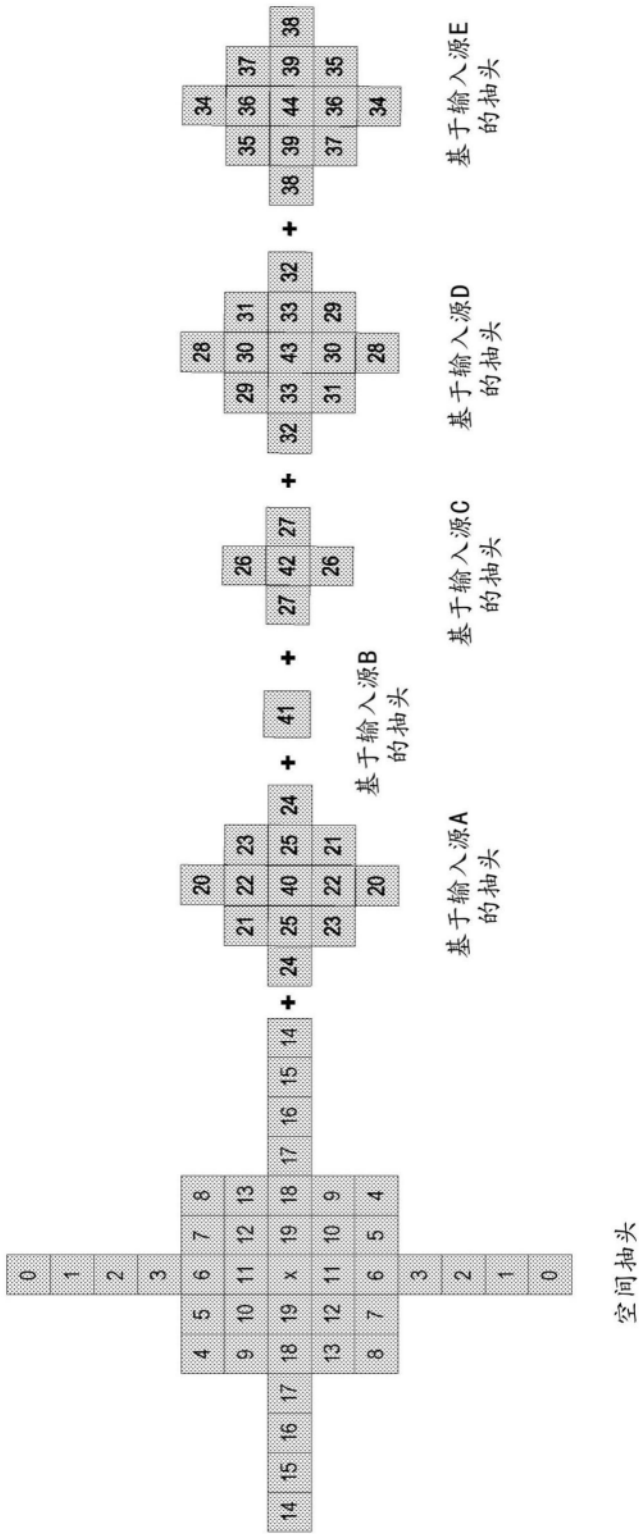


图18

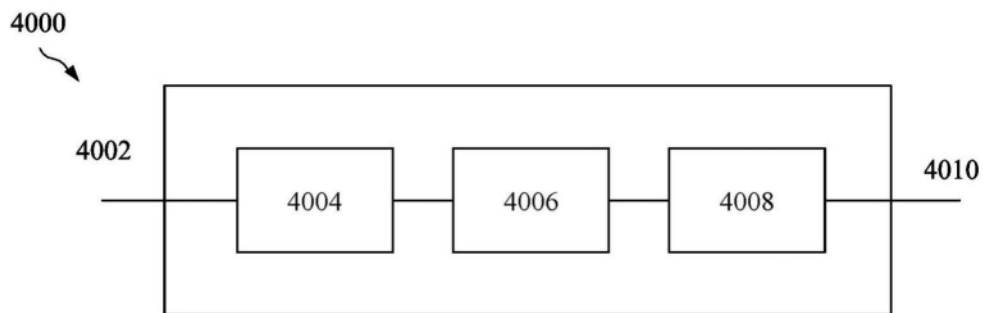


图19

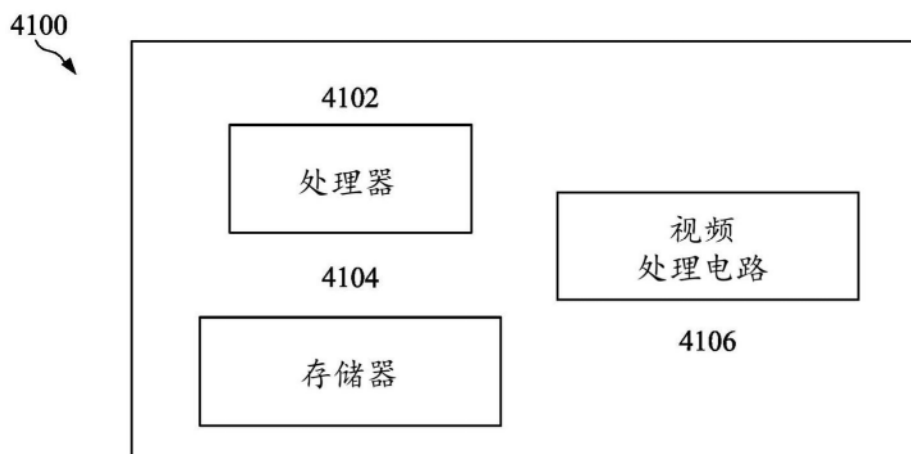


图20

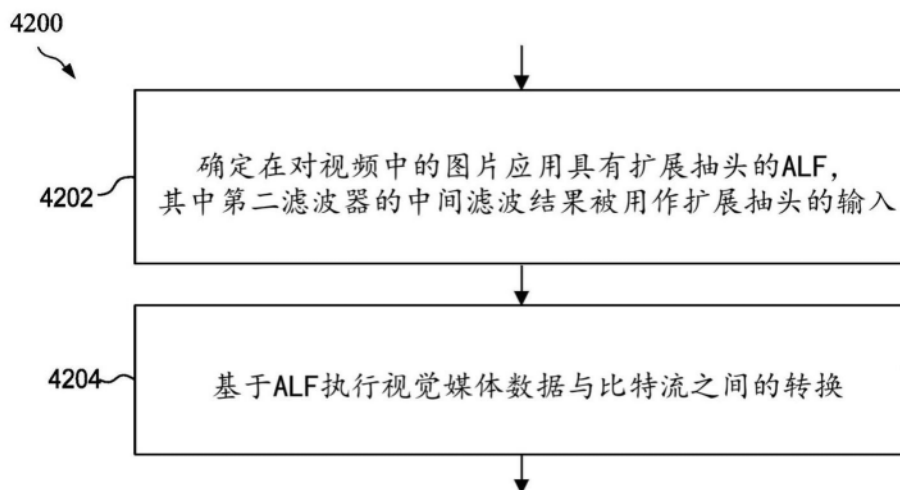


图21

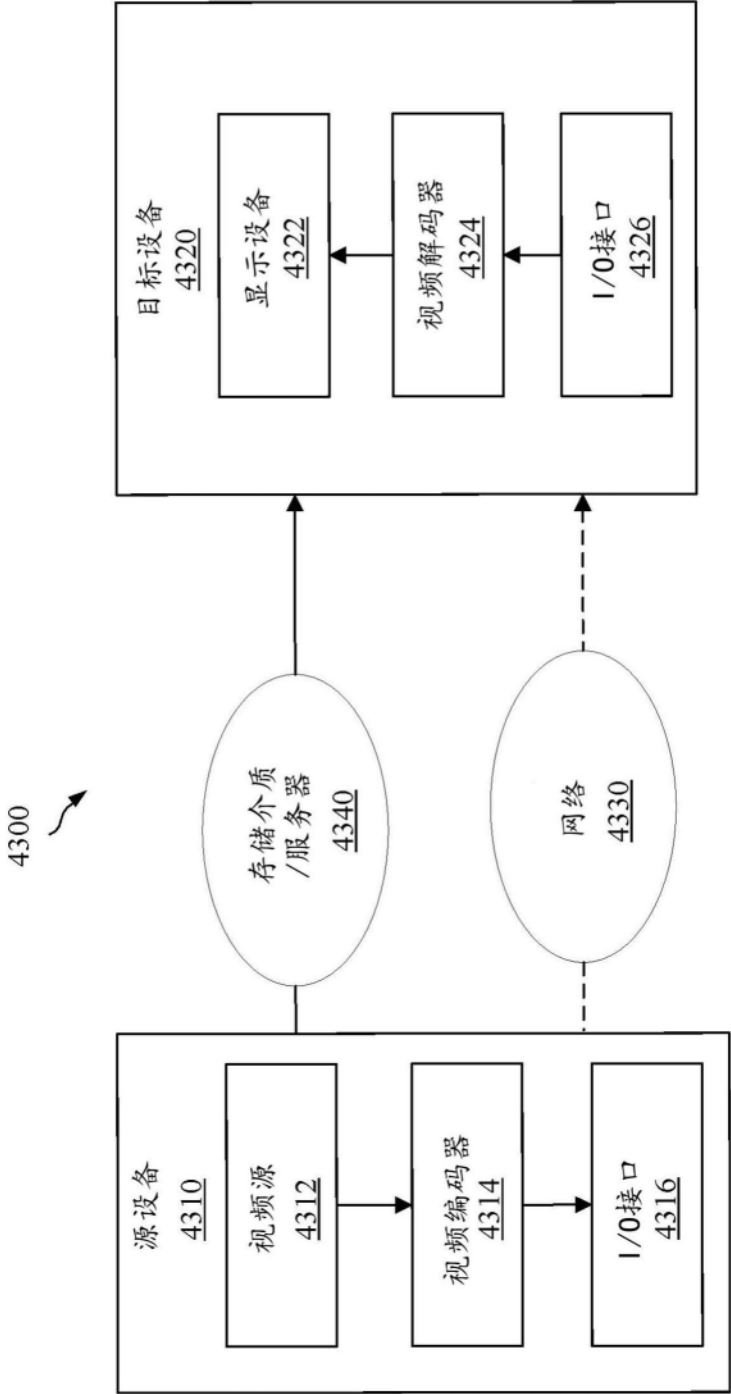


图22

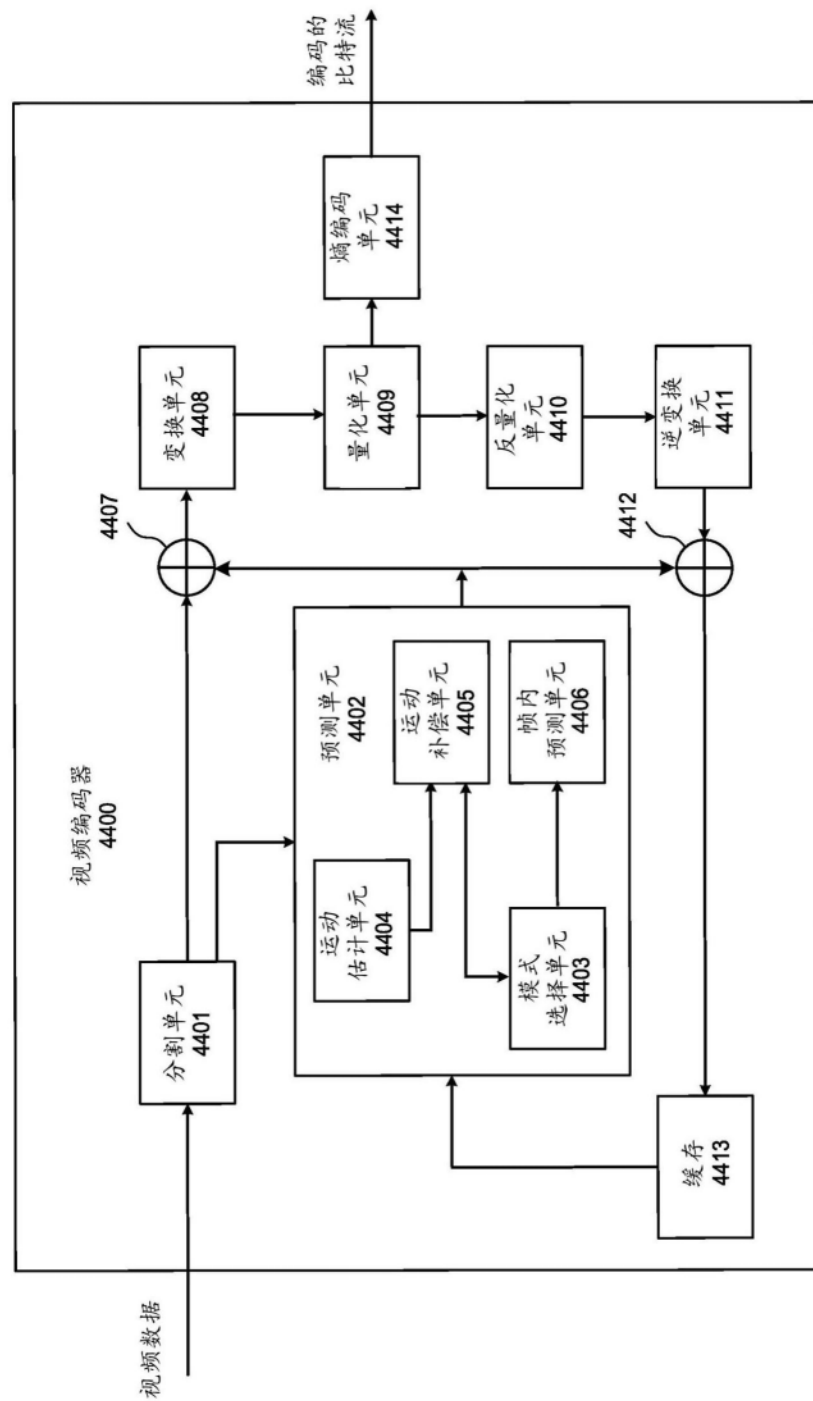


图23

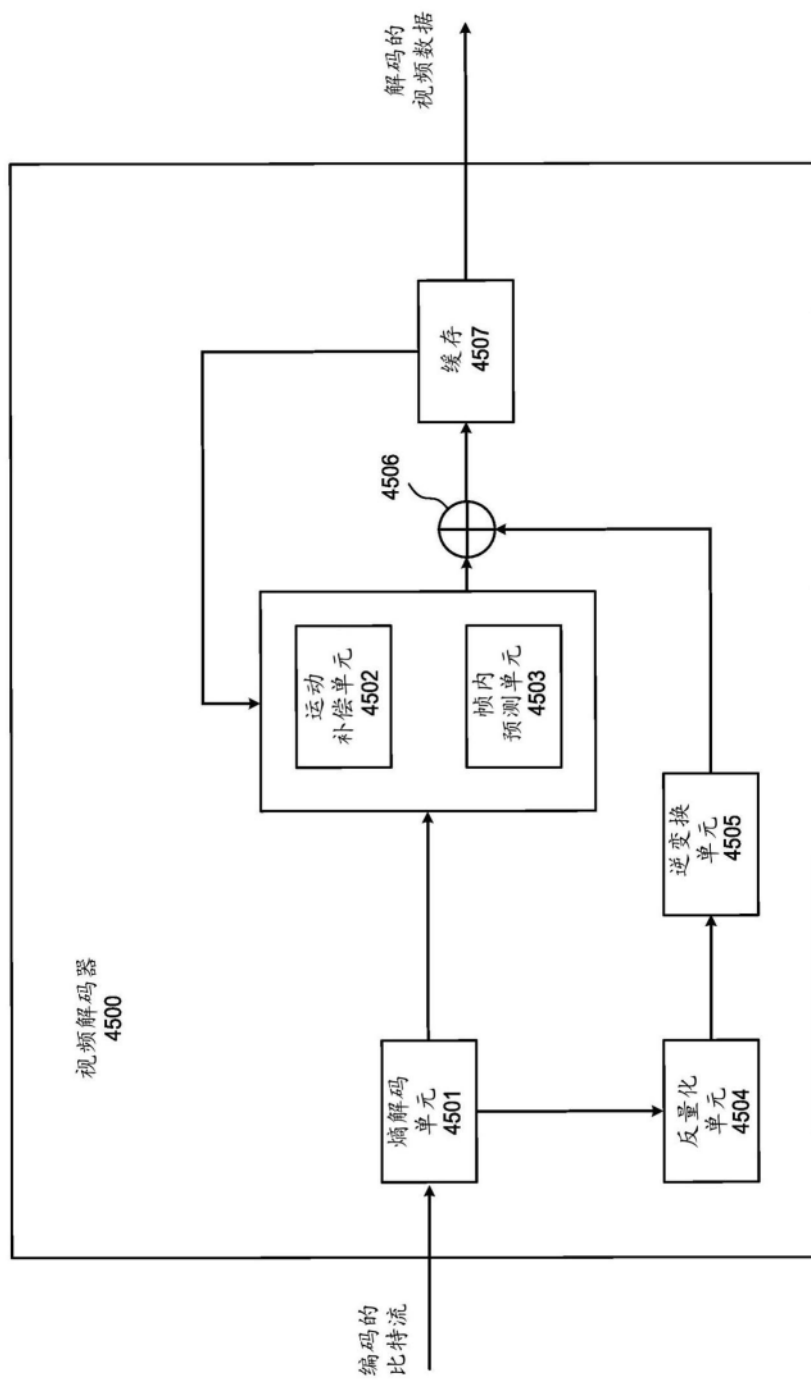


图24

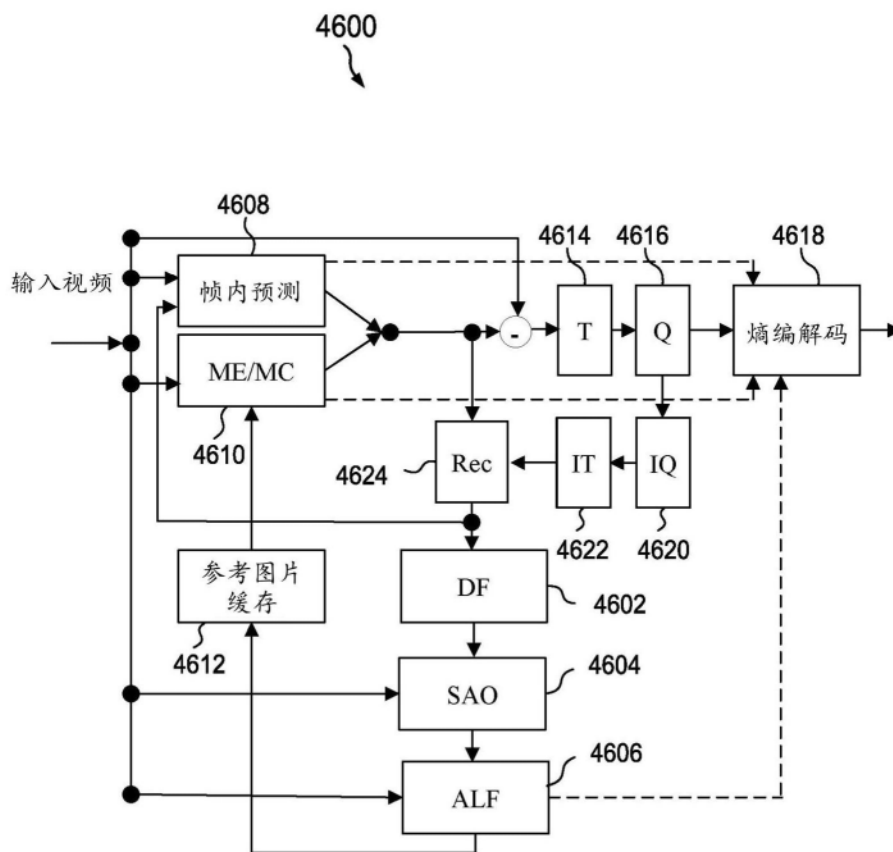


图25