(19)国家知识产权局

(12)发明专利

(10)授权公告号 CN 113728651 B
(45)授权公告日 2022.10.25

(21)申请号 202080026270.3
(22)申请日 2020.04.02
(65)同一申请的已公布的文献号
申请公布号 CN 113728651 A
(43)申请公布日 2021.11.30
(66)本国优先权数据
PCT/CN2019/080914 2019.04.02 CN
(85)PCT国际申请进入国家阶段日
2021.09.29
(86)PCT国际申请的申请数据
PCT/CN2020/082976 2020.04.02
(87)PCT国际申请的公布数据
WO2020/200277 EN 2020.10.08
(73)专利权人 北京字节跳动网络技术有限公司
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
专利权人 字节跳动有限公司
(72)发明人 刘鸿彬 张莉 张凯 许继征 王悦
(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
(51)Int.Cl.
H04N 19/86(2006.01)
H04N 19/117(2006.01)
H04N 19/82(2006.01)
(56)对比文件
CN 107925773 A,2018.04.17
CN 107925772 A,2018.04.17
WO 2017184970 A1,2017.10.26
CN 109479130 A,2019.03.15
CN 108293118 A,2018.07.17
WO 2017142939 A1,2017.08.24
CN 109076237 A,2018.12.21
Jonathan Taquet et al..Non-Linear Adaptive Loop Filter.《Joint Video Experts Team (JVET) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11》.2019,
审查员 李维维

权利要求书3页 说明书27页 附图19页

(54)发明名称

视频处理中的自适应环路滤波

(57)摘要

描述了用于视频编解码的样点细化和滤波方法的设备、系统和方法。在示例性方面,用于视频处理的方法包括对于视频的当前块和视频的比特流表示之间的转换,基于当前块的相邻像素集合配置滤波过程。该方法还包括基于滤波过程执行转换。

CN 113728651 B

1.一种视频处理方法,包括:

对于视频的当前块和所述视频的比特流表示之间的转换,基于所述当前块的相邻像素集合配置滤波过程;以及

基于所述滤波过程的输出执行所述转换,

其中,所述相邻像素集合包括所述当前块的水平相邻像素,所述滤波过程包括:

$O(x,y)=\sum_{(i,j)}w(i,j)*\sum_{si=-T}^{T}ws(si)*I(x+i+si,y+j)$,或者

$$O(x,y)=I(x,y)+\sum_{(i,j)\neq(0,0)}w(i,j).K\left(\sum_{si=-T}^{T}ws(si)*I(x+i+si,y+j)-I(x,y),k(i,j)\right),$$

或者其中,所述相邻像素集合包括所述当前块的垂直相邻像素,所述滤波过程包括:

$O(x,y)=\sum_{(i,j)}w(i,j)*\sum_{si=-T}^{T}ws(si)*I(x+i,y+j+si)$,或者

$$O(x,y)=I(x,y)+\sum_{(i,j)\neq(0,0)}w(i,j).K\left(\sum_{si=-T}^{T}ws(si)*I(x+i,y+j+si)-I(x,y),k(i,j)\right),$$

其中,I(x,y)是所述当前块的第(x,y)个像素,O(x,y)是第(x,y)个像素的输出值,w(i,j)是加权因子,K(•)是裁剪操作,T表示要用于所述滤波过程的像素数,ws(si)表示与T对应的滤波系数,并且k(i,j)表示裁剪参数。

2.根据权利要求1所述的方法,其中,所述滤波过程包括基于所述当前块的相邻像素集合的一维滤波操作。

3.根据权利要求2所述的方法,其中,所述一维滤波操作包括非线性自适应环路滤波方法。

4.根据权利要求2所述的方法,其中,T针对所述当前块中的不同像素而变化。

5.根据权利要求4所述的方法,其中,在所述当前块的第(x,y)个像素位于所述当前块的边界的情况下,要用于所述滤波过程的像素数T不包括位于所述当前块外部的像素。

6.根据权利要求2所述的方法,其中,对于所述当前块中的不同像素,ws(k)不同。

7.根据权利要求2所述的方法,其中,对于所述当前块中的不同像素,ws(k)相同。

8.根据权利要求2所述的方法,其中,在所述比特流表示中信令通知ws(k)。

9.根据权利要求2所述的方法,其中,ws(k)是预定义的。

10.根据权利要求1所述的方法,其中,所述滤波过程包括沿不同方向应用滤波器。

11.根据权利要求10所述的方法,其中,所述滤波器包括双边滤波器或哈达玛滤波器。

12.根据权利要求11所述的方法,其中,基于与所述当前块相关联的梯度信息,将所述滤波器应用于所述不同方向。

13.根据权利要求12所述的方法,包括:

对于所述当前块内的子块,确定所述子块中像素的水平梯度和垂直梯度;以及

在绝对水平梯度之和大于或等于绝对垂直梯度之和的情况下,将所述滤波器水平应用于所述子块。

14.根据权利要求12所述的方法,包括:

对于所述当前块内的子块,确定所述子块中像素的水平梯度和垂直梯度;以及

在绝对水平梯度之和大于或等于绝对垂直梯度之和的情况下,将所述滤波器垂直应用

于所述子块。

15.根据权利要求12所述的方法，包括：

对于所述当前块内的子块，确定所述子块中像素的水平梯度和垂直梯度；以及

在绝对水平梯度之和小于或等于绝对垂直梯度之和的情况下，将所述滤波器水平应用于所述子块。

16.根据权利要求12所述的方法，包括：

对于所述当前块内的子块，确定所述子块中像素的水平梯度和垂直梯度；以及

在绝对水平梯度之和小于或等于绝对垂直梯度之和的情况下，将所述滤波器垂直应用于所述子块。

17.根据权利要求12所述的方法，包括：

对于所述当前块内的子块，确定所述子块中像素的水平梯度、垂直梯度、45度对角梯度和135度对角梯度；以及

基于所确定的水平梯度、垂直梯度、45度对角梯度和135度对角梯度，将所述滤波器沿绝对梯度之和最大的方向应用于所述子块。

18.根据权利要求12所述的方法，包括：

对于所述当前块内的子块，确定所述子块中像素的水平梯度、垂直梯度、45度对角梯度和135度对角梯度；以及

基于所确定的水平梯度、垂直梯度、45度对角梯度和135度对角梯度，将所述滤波器沿绝对梯度之和最小的方向应用于所述子块。

19.根据权利要求1所述的方法，其中，所述滤波过程是否适用于所述当前块基于所述当前块的颜色格式。

20.根据权利要求19所述的方法，其中，所述滤波过程适用于所述当前块的单个颜色分量。

21.根据权利要求20所述的方法，其中，所述单个颜色分量是亮度颜色分量或绿色分量。

22.根据权利要求19所述的方法，其中，在所述当前块的颜色格式是4:4:4颜色格式的情况下，所述滤波过程适用于所述当前块的所有颜色分量。

23.根据权利要求1至22中任一项或多项所述的方法，其中，执行所述转换包括基于所述视频的当前块来生成所述比特流表示。

24.根据权利要求1至22中任一项或多项所述的方法，其中，执行所述转换包括从所述比特流表示生成所述视频的当前块。

25.一种视频系统中的装置，包括处理器和其上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实现权利要求1至22中任一项所述的方法。

26.一种非暂时性计算机可读存储介质，存储有指令，所述指令使处理器执行权利要求1至22中任一项所述的方法。

27.一种视频的比特流的存储方法，包括：

基于所述视频的当前块的相邻像素集合配置滤波过程；

基于所述滤波过程的输出生成所述视频的比特流；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，

其中,所述相邻像素集合包括所述当前块的水平相邻像素,所述滤波过程包括:

$O(x,y)=\sum_{(i,j)}w(i,j)*\sum_{si=-T}^{T}ws(si)*I(x+i+si,y+j)$,或者

$$O(x,y)=I(x,y)+\sum_{(i,j)\neq(0,0)}w(i,j).K\left(\sum_{si=-T}^{T}ws(si)*I(x+i+si,y+j)-I(x,y),k(i,j)\right),$$

或者其中,所述相邻像素集合包括所述当前块的垂直相邻像素,所述滤波过程包括:

$O(x,y)=\sum_{(i,j)}w(i,j)*\sum_{si=-T}^{T}ws(si)*I(x+i,y+j+si)$,或者

$$O(x,y)=I(x,y)+\sum_{(i,j)\neq(0,0)}w(i,j).K\left(\sum_{si=-T}^{T}ws(si)*I(x+i,y+j+si)-I(x,y),k(i,j)\right),$$

其中,I(x,y)是所述当前块的第(x,y)个像素,O(x,y)是第(x,y)个像素的输出值,w(i,j)是加权因子,K(•)是裁剪操作,T表示要用于所述滤波过程的像素数,ws(si)表示与T对应的滤波系数,并且k(i,j)表示裁剪参数。

视频处理中的自适应环路滤波

[0001] 相关申请的交叉引用

[0002] 根据适用专利法和/或适用于巴黎公约的规则，本申请及时要求2019年4月2日提交的国际专利申请No.PCT/CN2019/080914号的优先权和权益。出于法律的所有目的，上述申请的全部公开作为本申请公开的一部分通过引用并入。

技术领域

[0003] 本专利文档涉及视频编解码和解码技术、设备和系统。

背景技术

[0004] 尽管视频压缩有所进步，数字视频在互联网和其他数字通信网络上仍占最大的带宽使用量。随着能够接收和显示视频的连接用户设备的数量增加，预计数字视频使用所需的带宽将继续增长。

发明内容

[0005] 描述了与数字视频编解码有关的设备、系统和方法，尤其是与视频编解码的样点细化和滤波方法有关的设备、系统和方法。所描述的方法可以应用于现有的视频编解码标准(诸如，高效视频编解码(HEVC))以及未来的视频编解码标准(诸如，通用视频编解码(VVC)或编解码器)。

[0006] 在一个代表性的方面，所公开的技术可以用于提供用于视频处理的方法。该方法包括，对于视频的当前块和视频的比特流表示之间的转换，将裁剪操作应用于当前块中的像素与该像素的相邻像素之间的差的加权平均。该方法还包括基于裁剪操作的输出执行转换。

[0007] 在另一个代表性的方面，所公开的技术可以用于提供用于视频处理的方法。该方法包括，对于视频的当前块和视频的比特流表示之间的转换，基于当前块的相邻像素集合配置滤波过程。该方法还包括基于滤波过程的输出执行转换。

[0008] 在另一个代表性的方面，所公开的技术可以用于提供用于视频处理的方法。该方法包括，对于视频的块和视频的比特流表示之间的转换，通过将裁剪操作应用于细化值来修改该块中的预测样点的细化值。细化值是基于光流编解码处理的梯度值推导的。裁剪操作的输出在范围内。该方法还包括基于修改的细化值来细化预测样点，以及基于细化的预测样点来执行转换。

[0009] 在另一个代表性的方面，所公开的技术可以用于提供用于视频处理的方法。该方法包括，对于视频的块和视频的比特流表示之间的转换，修改光流编解码过程中的梯度值。该方法还包括基于用于转换的修改的梯度值执行后续预测操作。

[0010] 在另一个代表性的方面，所公开的技术可以用于提供用于视频处理的方法。该方法包括，为当前视频块配置一维滤波过程，以及基于一维滤波过程，从对应的比特流表示中重建当前视频块。

[0011] 在又一代表性方面，上述方法以处理器可执行代码的形式实施并存储在计算机可读程序介质中。

[0012] 在又一代表性方面，公开了一种配置或可操作以执行上述方法的设备。该设备可以包括被编程为实现该方法的处理器。

[0013] 在又一代表性方面，视频解码器装置可实现如本文中所描述的方法。

[0014] 在附图、说明书和权利要求中更详细地描述了所公开技术的上述和其他方面和特征。

附图说明

[0015] 图1示出了用于视频编解码的编码器框图的示例。

[0016] 图2A、2B和2C示出了基于几何变换的自适应环路滤波器(GALF)滤波器形状的示例。

[0017] 图3示出了用于GALF编码器决策的流程图的示例。

[0018] 图4A-4D示出了用于自适应环路滤波器(ALF)分类的示例子采样拉普拉斯计算。

[0019] 图5示出了亮度滤波器形状的示例。

[0020] 图6示出了宽视频图形阵列(WVGA)序列的区域划分的示例。

[0021] 图7示出了具有整形(reshaping)的解码流程的示例性流程图。

[0022] 图8示出了双向光流(BDOF或BIO)算法所使用的光流轨迹的示例。

[0023] 图9A和9B示出了使用没有块扩展的双向光流(BIO)算法的示例快照。

[0024] 图10示出了具有光流(PROF)的预测细化的示例。

[0025] 图11A和11B示出了根据所公开的技术的用于视频编解码中的样点细化和滤波的示例方法的流程图。

[0026] 图12是用于实现本文档中描述的视觉媒体解码或视觉媒体编码技术的硬件平台的示例的框图。

[0027] 图13是其中可实现所公开的技术的示例视频处理系统的框图。

[0028] 图14是根据本技术的用于视频处理的方法的流程图表示。

[0029] 图15是根据本技术的用于视频处理的另一种方法的流程图表示。

[0030] 图16是根据本技术的用于视频处理的另一种方法的流程图表示。

[0031] 图17是根据本技术的用于视频处理的又一种方法的流程图表示。

具体实施方式

[0032] 由于对更高分辨率视频的需求的增加，在现代技术中普遍存在视频编解码方法和技术。视频编解码器通常包括压缩或解压缩数字视频的电子电路或软件，并且不断改进以提供更高的编解码效率。视频编解码器将未压缩视频转换为压缩格式，反之亦然。视频质量、用于表示视频的数据量(由比特率确定)、编码和解码算法的复杂度、对数据丢失和错误的敏感性、编辑的简易性、随机访问和端到端延迟(迟延)之间存在复杂的关系。压缩格式通常符合标准视频压缩规范，例如，高效视频编解码(HEVC)标准(也称为H.265或MPEG-H第2部分)、要完成的通用视频编解码(Versatile Video Coding，VVC)标准、或其他当前和/或未来的视频编解码标准。

[0033] 在一些实施例中，使用称为联合探索模型(JEM)的参考软件来探索未来的视频编解码技术。在JEM中，在几种编解码工具中采用基于子块的预测，诸如仿射预测、可选时域运动矢量预测(ATMVP)、空时运动矢量预测(STMVP)、双向光流(BIO)、帧速率上转换(FRUC)、本地自适应运动矢量分辨率(LAMVR)、重叠块运动补偿(OBMC)、局部照明补偿(LIC)和解码器侧运动矢量细化(DMVR)。

[0034] 所公开技术的实施例可应用于现有视频译码标准(例如，HEVC、H.265)和未来标准以改善运行时间性能。在本文档中使用节标题来提高描述的可读性，并且不以任何方式将讨论或实施例(和/或实施方式)仅限于相应节。

[0035] 1.颜色空间和色度子采样的示例

[0036] 颜色空间，也称为颜色模型(或颜色系统)，是一种抽象数学模型，其将颜色范围简单地描述为数字元组，通常为3或4个值或颜色分量(例如RGB)。基本上，颜色空间是对坐标系和子空间的详细说明。

[0037] 对于视频压缩，最常用的颜色空间是YCbCr和RGB。

[0038] YcbCr、Y'CbCr或Y Pb/Cb Pr/Cr，也称为YCBCR或Y'CBCR，是用作视频和数字摄影系统中彩色图像流水线(pipeline)的一部分的一系列颜色空间。Y'是亮度分量，CB和CR是蓝差(blue-difference)和红差(red-difference)色度分量。Y′(带有撇号(prime))区分于Y(亮度)，表示光强度是基于伽玛校正的RGB原色进行非线性编码的。

[0039] 色度子采样是通过对色度信息实现比对亮度信息更低的分辨率来对图像进行编码的实践，利用了人类视觉系统对色差比对亮度的更低敏锐度。

[0040] 1.1 4:4:4颜色格式

[0041] 三个Y'CbCr分量中的每一个具有相同的采样率，因此不存在色度子采样。该方案有时用于高端胶片扫描仪和电影后期制作中。

[0042] 1.2 4:2:2颜色格式

[0043] 两个色度分量以亮度的采样率的一半(例如，水平色度分辨率减半)进行采样。这将未压缩视频信号的带宽减少了三分之一，且几乎没有视觉差异。

[0044] 1.3 4:2:0颜色格式

[0045] 在4:2:0中，水平采样是4:1:1的两倍，但是由于Cb和Cr通道仅在此方案中的每条交替线上采样，因此垂直分辨率减半。因此，数据速率相同。Cb和Cr分别在水平和垂直方向上以2的因子进行子采样。4:2:0方案有三种变体，其具有不同的水平和垂直位置。

[0046] ○在MPEG-2中，Cb和Cr水平共存。Cb和Cr在垂直方向上位于像素之间(间隙地位于)。

[0047] ○在JPEG/JFIF、H.261和MPEG-1中，Cb和Cr间隙地位于交替亮度样点之间的中间。

[0048] ○在4:2:0 DV中，Cb和Cr在水平方向上共存。在垂直方向上，它们共存于交替线上。

[0049] 2.典型视频编解码器的编解码流程的示例

[0050] 图1示出了VVC的编码器框图的示例，其中包含三个环路滤波块：去块滤波器(DF)、采样自适应偏移(SAO)和自适应环路滤波器(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分别通过增加偏移和应用有限脉冲响应(FIR)滤波器来减少原始样点与重建样点之间的均方误差，其中编解码的边信息(side information)信令通知

偏移和滤波器系数。ALF位于每个图片的最后一个处理阶段，可以看作是试图捕获和修复由先前阶段创建的伪像的工具。

[0051] 3.JEM中基于几何变换的自适应环路滤波器的示例

[0052] 在JEM中，应用了具有基于块的滤波器自适应的基于几何变换的自适应环路滤波器(GALF)。对于亮度分量，基于局部梯度的方向和活动，为每个2×2块选择25个滤波器中的一个。

[0053] 3.1滤波器形状的示例

[0054] 在JEM中，可以为亮度分量选择多达三个菱形滤波器形状(分别如图2A、2B和2C所示的5×5菱形、7×7菱形和9×9菱形)。在图片级别信令通知索引，以指示用于亮度分量的滤波器形状。对于图片中的色度分量，使用5×5菱形形状。

[0055] 3.1.1块分类

[0056] 将每个2×2块分类为25个类中的一个。分类索引C基于其方向性D和活动 $\hat{A}$ 的量化值进行推导，如下：

[0057] $$C = 5D + \hat{A}. \tag{1}$$

[0058] 为了计算D和 $\hat{A}$，首先使用一维拉普拉斯变换计算水平、垂直和两个对角方向的梯度：

[0059] $$g_v = \sum_{k=i-2}^{i+3} \sum_{l=j-2}^{j+3} V_{k,l}, \; V_{k,l} \tag{2}$$
$$= |2R(k,l) - R(k,l-1) - R(k,l+1)|,$$

[0060] $$g_h = \sum_{k=i-2}^{i+3} \sum_{l=j-2}^{j+3} H_{k,l}, \; H_{k,l} \tag{3}$$
$$= |2R(k,l) - R(k-1,l) - R(k+1,l)|,$$

[0061] $$g_{d1} = \sum_{k=i-2}^{i+3} \sum_{l=j-3}^{j+3} D1_{k,l}, D1_{k,l} \tag{4}$$
$$= |2R(k,l) - R(k-1,l-1) - R(k+1,l+1)|$$

[0062] $$g_{d2} = \sum_{k=i-2}^{i+3} \sum_{j=j-2}^{j+3} D2_{k,l}, \; D2_{k,l} \tag{5}$$
$$= |2R(k,l) - R(k-1,l+1) - R(k+1,l-1)|$$

[0063] 索引i和j指代2×2块内左上样点的坐标，并且R(i,j)指示在坐标(i,j)处的重建样点。

[0064] 然后将水平和垂直方向的梯度的D个最大值和最小值设置为：

[0065] $$g_{h,v}^{max} = max(g_h, g_v), \quad g_{h,v}^{min} = min(g_h, g_v), \tag{6}$$

[0066] 并且将两个对角方向的梯度的最大值和最小值设置为：

[0067] $$g_{d0,d1}^{max} = max(g_{d0}, g_{d1}), \quad g_{d0,d1}^{min} = min(g_{d0}, g_{d1}), \tag{7}$$

[0068] 为了推导方向性D的值，将这些值相互并使用两个阈值t_1和t_2进行比较：

[0069] 步骤1.如果$g_{h,v}^{max} \le t_1 \cdot g_{h,v}^{min}$ 且$g_{d0,d1}^{max} \le t_1 \cdot g_{d0,d1}^{min}$均为真，则将D设置为0。

[0070] 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。

[0071] 步骤3.如果$g_{h,v}^{max} > t_2 \cdot g_{h,v}^{min}$，则将D设置为2；否则将D设＝设置为1。

[0072] 步骤4.如果$g_{d0,d1}^{max} > t_2 \cdot g_{d0,d1}^{min}$，则将D设置为4；否则将D设置为3。

[0073] 将活动值A计算为：

[0074] $$A = \sum_{k=i-2}^{i+3} \sum_{l=j-2}^{j+3} (V_{k,l} + H_{k,l}). \tag{8}$$

[0075] 将A进一步量化到0到4的范围(包括端值)，并且将量化值表示为$\hat{A}$。对于图片中的两个色度分量，不应用分类方法，即，对每个色度分量应用单个ALF系数集。

[0076] 3.1.2滤波器系数的几何变换

[0077] 在对每个2×2块进行滤波之前，取决于为该块计算的梯度值，将诸如旋转或对角和垂直翻转的几何变换应用于滤波器系数f(k,l)。这等效于将这些变换应用于滤波器支持区域中的样点。思想是通过对齐它们的方向性来使应用了ALF的不同块更加相似。

[0078] 引入了三种几何变换，包括对角、垂直翻转和旋转：

[0079] 对角：$f_D(k,l) = f(l,k)$，

[0080] 垂直翻转：$f_V(k,l) = f(k,K-1-l)$， (9)

[0081] 旋转：$f_R(k,l) = f(K-1-l,k)$。

[0082] 其中，K是滤波器的尺寸，并且$0 \le k,l \le K-1$是系数坐标，使得位置(0,0)在左上角，并且位置(K-1,K-1)在右下角。取决于为该块计算的梯度值，将变换应用于滤波器系数f(k,l)。表1总结了变换与四个方向的四个梯度之间的关系。

[0083] 表1-为一个块计算的梯度和变换之间的映射

[0084]

梯度值	变换
$g_{d2} < g_{d1}$ and $g_h < g_v$	无变换
$g_{d2} < g_{d1}$ and $g_v < g_h$	对角
$g_{d1} < g_{d2}$ and $g_h < g_v$	垂直翻转
$g_{d1} < g_{d2}$ and $g_v < g_h$	旋转

[0085] 3.1.3滤波器参数的信令

[0086] 在JEM中，例如在第一CTU的条带标头之后和SAO参数之前，为第一CTU信令通知GALF滤波器参数。可以信令通知多达25组亮度滤波器系数。为了减少比特开销，可以合并不

同分类的滤波器系数。另外，参考图片的GALF系数被存储并且被允许重新使用作为当前图片的GALF系数。当前图片可以选择使用为参考图片存储的GALF系数，并绕过信令通知GALF系数。在这种情况下，仅信令通知参考图片之一的索引，并且所存储的指示的参考图片的GALF系数被继承用于当前图片。

[0087] 为了支持GALF时域预测，维持GALF滤波器集合的候选列表。在开始解码新序列时，候选列表为空。在解码一个图片之后，可以将对应的滤波器集合添加到候选列表。一旦候选列表的尺寸达到最大可允许值(例如，当前JEM中为6)，新的滤波器集合就会以解码顺序覆盖最旧的集合，即将先进先出(FIFO)规则应用于更新候选列表。为避免重复，只有当对应图片不使用GALF时域预测时，才可以将集合添加到列表中。为了支持时域可伸缩性，存在滤波器集合的多个候选列表，并且每个候选列表都与时域层相关联。更具体地，由时域层索引(TempIdx)分配的每个阵列可以组成具有等于较低的TempIdx的先前解码图片的滤波器集合。例如，第k个阵列被分配为与等于k的TempIdx相关联，并且它仅包含来自具有TempIdx小于或等于k的图片的滤波器集合。在对某个图片进行编解码之后，与该图片相关联的滤波器集合将用于更新与相等的或更高的TempIdx相关联的那些阵列。

[0088] GALF系数的时域预测用于帧间编解码的帧，以最小化信令开销。对于帧内帧，时域预测不可用，并且为每个类别分配了16个固定滤波器的集合。为了指示固定滤波器的用法，将信令通知每个类别的标志，并在必要时信令通知所选固定滤波器的索引。即使为给定类别选择了固定滤波器，仍可以为该类别发送自适应滤波器的系数f(k,l)，在这种情况下，将应用于重建图像的滤波器系数为两组系数的和。

[0089] 亮度分量的滤波过程可以控制在CU级别。信令通知标志以指示是否将GALF应用于CU的亮度分量。对于色度分量，仅在图片级别指示是否应用GALF。

[0090] 3.1.4滤波过程

[0091] 在解码器侧，当对块启用GALF时，对该块内的每个样点R(i,j)进行滤波，得出样点值R′(i,j)，如下所示，其中L表示滤波器长度，$f_{m,n}$表示滤波器系数，f(k,l)表示解码的滤波器系数。

[0092] $$R'(i,j)=\sum_{k=-L/2}^{L/2}\sum_{l=-L/2}^{L/2}f(k,l)\times R(i+k,j+l) \qquad (10)$$

[0093] 3.1.5编码器侧滤波器参数的确定过程

[0094] 在图3中示出了用于GALF的总体编码器决策过程。对于每个CU的亮度样点，编码器决策是否应用GALF以及是否在条带标头中包括适当的信令标记。对于色度样点，基于图片级别而不是CU级别决策是否应用滤波器。此外，仅当为图片启用亮度GALF时，才会检查图片的色度GALF。

[0095] 4.VVC中基于几何变换的自适应环路滤波器的示例

[0096] 与JEM相比，VVC中GALF的当前设计具有以下主要变化：

[0097] (1)去除自适应滤波器形状。亮度分量仅允许7x7滤波器形状，色度分量仅允许5x5滤波器形状。

[0098] (2)去除了ALF参数的时域预测和来自固定滤波器的预测。

[0099] (3)对于每个CTU，无论启用或禁用ALF都信令通知一比特标志。

[0100] (4)类别索引的计算以4x4级别而不是2x2执行。另外，在一些实施例中，利用了用

于ALF分类的子采样拉普拉斯计算方法。更具体地，不需要为一个块内的每个样点计算水平/垂直/45对角/135度梯度。取而代之的是，使用了1:2子采样。

[0101] 5.AVS2中基于区域的自适应环路滤波器的示例

[0102] ALF是环路滤波的最后阶段。此过程中有两个阶段。第一阶段是滤波器系数推导。为了训练滤波器系数，编码器将亮度分量的重建像素分类为16个区域，并使用维纳-霍夫方程对每个类别训练一组滤波器系数，以最小化原始帧和重建帧之间的均方误差。为了减少这16组滤波器系数之间的冗余，编码器将基于率失真性能自适应地合并它们。最多可以为亮度分量分配16个不同的滤波器集合，而为色度分量只能分配一个。第二阶段是滤波器决策，其包括帧级别和LCU级别。首先，编码器决策是否执行帧级别自适应环路滤波。如果帧级别ALF启用，则编码器进一步确定是否执行LCU级别ALF。

[0103] 5.1滤波器形状

[0104] 对于亮度和色度分量，示例性的滤波器形状是叠加了3×3正方形的7×7十字形，如图5所示。图5中的每个正方形对应于样点。因此，总共使用17个样点来推导位置C8的样点的滤波值。考虑到发送系数的开销，使用仅剩九个系数{C0,C1,...,C8}的点对称滤波器，这将滤波器系数的数量减少了一半，并且将滤波中的乘法的数量减少了一半。点对称滤波器还可以将一个滤波的样点的计算减少一半，例如，对于一个滤波的样点仅进行9次乘法和14次加法运算。

[0105] 5.2基于区域的自适应Merge

[0106] 为了适应不同的编解码误差，一些实施例对亮度分量采用基于区域的多个自适应环路滤波器。亮度分量被分成16个尺寸大致相等的基本区域，其中每个基本区域与最大编解码单元(LCU)边界对齐，如图6所示，并且对每个区域推导一个维纳滤波器。使用的滤波器越多，减少的失真就越多，但是用于对这些系数进行编码的比特随着滤波器的数量而增加。为了获得最佳率失真性能，可以将这些区域合并为较少的较大区域，这些区域共享相同的滤波器系数。为了简化合并过程，基于图像先验相关性，根据修改的希尔伯特顺序，为每个区域分配索引。可以基于率失真成本合并具有连续索引的两个区域。

[0107] 区域之间的映射信息应该信令通知给解码器。在AVS-2中，基本区域的数量用于表示Merge结果，并且滤波器系数根据其区域顺序依次压缩。例如，当{0,1}、{2、3、4}、{5、6、7、8、9}和剩余基本区域分别合并为一个区域时，仅对三个整数进行编解码来表示该Merge映射，即2、3、5。

[0108] 5.3边信息的信令

[0109] 还使用了多个开关标志。序列开关标志adaptive_loop_filter_enable用于控制是否对整个序列应用自适应环路滤波器。图像开关标志picture_alf_enble[i]控制是否对对应的第i个图像分量应用ALF。仅当启用picture_alf_enble[i]时，才发送该颜色分量的对应LCU级别标志和滤波器系数。LCU级别标志lcu_alf_enable[k]控制是否对对应的第k个LCU启用ALF，并将该标志交织到条带数据中。不同级别的调节标志的决策全部基于率失真成本。高灵活性进一步使ALF大大提高了编解码效率。

[0110] 在一些实施例中，对于亮度分量，可能有多达16组滤波器系数。

[0111] 在一些实施例中，对于每个色度分量(Cb和Cr)，可以发送一组滤波器系数。

[0112] 6示例GALF

[0113] 在一些实施例中,自适应环路滤波器的滤波过程如下执行:

[0114] $O(x,y)=\sum_{(i,j)}w(i,j).I(x+i,y+j)$ (11)

[0115] 其中样点I(x+i,y+j)是输入样点,O(x,y)是滤波后的输出样点(即滤波结果),并且w(i,j)表示滤波系数。在一些实施例中,它使用用于定点精度计算的整数算法来实现:

[0116] $$O(x,y)=\left(\sum_{i=-\frac{L}{2}}^{\frac{L}{2}}\sum_{j=-\frac{L}{2}}^{\frac{L}{2}}w(i,j).I(x+i,y+j)+64\right)\gg 7 \quad (12)$$

[0117] 其中L表示滤波器长度,并且其中w(i,j)是定点精度的滤波器系数。

[0118] 7非线性自适应环路滤波(ALF)

[0119] 7.1滤波重述(reformulation)

[0120] 可以在以下表达式中重述公式(11),而不影响编解码效率:

[0121] $O(x,y)=I(x,y)+\sum_{(i,j)\neq(0,0)}w(i,j).(I(x+i,y+j)-I(x,y))$ (13)

[0122] 其中,w(i,j)是与公式(11)相同的滤波器系数[除了w(0,0),其在公式(13)中等于1,而在公式(11)中等于$1-\sum_{(i,j)\neq(0,0)}w(i,j)$]。

[0123] 7.2修改的滤波器

[0124] 使用该上述(13)的滤波器公式,可以通过使用简单的裁剪函数来引入非线性以提高ALF的效率,以在邻居样点值与当前正被滤波的样点值(I(x,y)),大不相同时,减少邻居样点值(I(x+i,y+j))的影响。

[0125] 在此提议中,ALF滤波器修改如下:

[0126] $O'(x,y)=I(x,y)+\sum_{(i,j)\neq(0,0)}w(i,j).K(I(x+i,y+j)-I(x,y),k(i,j))$ (14)

[0127] 其中,K(d,b)=min(b,max(-b,d))是裁剪函数,并且k(i,j)是取决于(i,j)滤波器系数的裁剪参数。编码器执行优化以找到最佳k(i,j)。

[0128] 在一些实施例中,为每个ALF滤波器指定裁剪参数k(i,j),对每个滤波器系数信令通知一个裁剪值。这意味着对每个亮度滤波器可以在比特流中信令通知最多12个裁剪值,对色度滤波器可以在比特流中信令通知最多6个裁剪值。

[0129] 为了限制信令成本和编码器复杂度,可以将裁剪值的评估限制为可能值的小集合。只能使用4个与INTER和INTRA片组相同的固定值。

[0130] 由于亮度的局部差异的方差通常比色度的高,因此对于亮度和色度滤波器可以使用两个不同的集合。可以在每个集合中包括最大样点值(在本文中对于10比特的比特深度为1024),以便在不必要时可以禁用裁剪。

[0131] 在表2中提供了在一些实施例中使用的裁剪值的集合。通过在对数域中将亮度的样点值的全范围(以10比特进行编解码)以及色度从4到1024的范围进行大致相等地划分,选择了4个值。

[0132] 更精确地,裁剪值的亮度表已通过以下公式获得:

[0133] $\mathrm{AlfClip_L}=\left\{\mathrm{round}\left(\left((M)^{\frac{1}{N}}\right)^{N-n+1}\right)\ \mathrm{for}\ n\in 1..N\right\}$,其中$M=2^{10}$且N=4。

[0134] 类似地,裁剪值的色度表根据以下公式获得:

[0135] $\mathrm{AlfClip_C}=\left\{\mathrm{round}\left(A.\left(\left(\frac{M}{A}\right)^{\frac{1}{N-1}}\right)^{N-n}\right)\ \mathrm{for}\ n\in\ 1..N\right]\right\}$，其中M＝$2^{10}$，N＝4且A＝4。

[0136] 表2:授权的裁剪值

[0137]

	帧内/帧间片组
亮度	{1024, 181, 32, 6}
色度	{1024, 161, 25, 4}

[0138] 通过使用与上表2中的裁剪值的索引相对应的Golomb编码方案，在“alf_data”语法元素中对所选择的裁剪值进行编解码。该编码方案与滤波器索引的编码方案相同。

[0139] 8示例环路整形(ILR)

[0140] 环路整形(ILR)的基本思想是将原始(在第一域中)信号(预测/重建信号)转换为第二域(整形域)。

[0141] 环路亮度整形器实现为一对查找表(LUT)，但是两个LUT中只有一个需要信令通知，因为另一个可以从信令通知的LUT计算出来。每个LUT是一维10比特1024项映射表(1D-LUT)。一个LUT是前向LUT FwdLUT，它将输入的亮度编解码值Y_i映射为更改的值Y_r：Y_r＝FwdLUT[Y_i]。另一个LUT是反向LUT InvLUT，它将更改的编解码值Y_r映射为$\hat{Y}_i$：$\hat{Y}_i = InvLUT[Y_r]$。($\hat{Y}_i$表示Y_i的重建值。)。

[0142] 8.1示例分段线性(PWL)模型

[0143] 从概念上讲，分段线性(PWL)通过以下方式实现：

[0144] 令x1、x2为两个输入枢轴点，并且y1、y2为一段上它们对应的输出枢轴点。在x1和x2之间的任何输入值x的输出值y可以通过以下公式进行插值：

[0145] y＝((y2-y1)/(x2-x1))*(x-x1)+y1

[0146] 在定点实现中，公式可重写为：

[0147] y＝((m*x+2FP_PREC-1)＞＞FP_PREC)+c

[0148] 其中，m是标量，c是偏移，并且FP_PREC是用于指定精度的常数。

[0149] 注意，在一些实施例中，PWL模型用于预计算1024项的FwdLUT和InvLUT映射表；但是，PWL模型还允许无需预计算LUT即可实时计算相同映射值的实施方式。

[0150] 8.2示例整形

[0151] 8.2.1亮度整形

[0152] 环路亮度整形的一些实施例提供了较低复杂度的流水线，该流水线还消除了在帧间条带重建中分块帧内预测的解码时延。对帧间和帧内条带均在整形域中执行帧内预测。

[0153] 无论条带类型如何，都在整形域中执行帧内预测。通过这种布置，可以在完成先前的TU重建之后立即开始帧内预测。这样的布置还可以为帧内模式提供统一的过程，而不是依赖于条带。图7示出了基于模式的示例解码过程的框图。

[0154] 一些实施例还测试了亮度和色度残差缩放的16段分段线性(PWL)模型，而不是32段PWL模型。

[0155] 在一些实施例中,利用环路亮度整形器进行帧间条带重建(较浅的阴影块指示整形域中的信号:亮度残差;预测的帧内亮度;以及重建的帧内亮度)

[0156] 8.2.2依赖亮度的色度残差缩放

[0157] 依赖亮度的色度残差缩放是用定点整数运算实现的乘法过程。色度残差缩放补偿亮度信号与色度信号的相互作用。在TU级别应用色度残差缩放。更具体地,适用以下内容:

[0158] ○对于帧内,对重建的亮度进行平均。

[0159] ○对于帧间,对预测亮度进行平均。

[0160] 平均值用于标识PWL模型中的索引。该索引标识缩放因子cScaleInv。将色度残差乘以该数。

[0161] 注意,根据前向映射的预测亮度值而不是重建的亮度值来计算色度缩放因子。

[0162] 8.2.3 ILR边信息的信令

[0163] 在片组标头中(当前)发送参数(类似于ALF)。据报告,这些占用了40-100比特。

[0164] 以下说明书基于JVET-L1001的版本9。添加的语法以黄色突出显示。

[0165] 在7.3.2.1中的序列参数集RBSP语法

[0166]

seq_parameter_set_rbsp( ) {	描述符
sps_seq_parameter_set_id	ue(v)
…	
sps_triangle_enabled_flag	u(1)
sps_ladf_enabled_flag	u(1)
if (sps_ladf_enabled_flag) {	
sps_num_ladf_intervals_minus2	u(2)
sps_ladf_lowest_interval_qp_offset	se(v)
for(i = 0; i < sps_num_ladf_intervals_minus2 + 1; i++) {	
sps_ladf_qp_offset[i]	se(v)
sps_ladf_delta_threshold_minus1[i]	ue(v)
}	
}	
sps_reshaper_enabled_flag	u(1)
rbsp_trailing_bits( )	
}	

[0167] 在7.3.3.1中的通用片组标头语法

[0168]

tile_group_header( ) {	描述符

[0169]

…	
if(num_tiles_in_tile_group_minus1 > 0) {	
offset_len_minus1	ue(v)
for(i = 0; i < num_tiles_in_tile_group_minus1; i++)	
entry_point_offset_minus1[i]	u(v)
}	
if (sps_reshaper_enabled_flag) {	
tile_group_reshaper_model_present_flag	u(1)
if (tile_group_reshaper_model_present_flag)	
tile_group_reshaper_model ( )	
tile_group_reshaper_enable_flag	u(1)
if (tile_group_reshaper_enable_flag && (!(qtbtt_dual_tree_intra_flag && tile_group_type == I)))	
tile_group_reshaper_chr oma_residual_scale_flag	u(1)
}	
byte_alignment( )	
}	

[0170] 添加新语法表片组整形器模型：

[0171]

tile_group_reshaper_model () {	**描述符**
reshaper_model_min_bin_idx	ue(v)
reshaper_model_delta_max_bin_idx	ue(v)
reshaper_model_bin_delta_abs_cw_prec_minus1	ue(v)
for (i = reshaper_model_min_bin_idx; i <= reshaper_model_max_bin_idx; i++) {	
reshape_model_bin_delta_abs_CW [i]	u(v)
if (reshaper_model_bin_delta_abs_CW[i]) > 0)	
reshaper_model_bin_delta_sign_CW_flag[i]	u(1)
}	
}	

[0172] 在通用序列参数集RBSP语义中，添加以下语义：

[0173] sps_reshaper_enabled_flag等于1指定在编解码的视频序列(CVS)中使用整形器。sps_reshaper_enabled_flag等于0指定在CVS中不使用整形器。

[0174] 在片组标头语法中，添加以下语义

[0175] tile_group_reshaper_model_present_flag等于1指定片组标头中存在tile_group_reshaper_model()。tile_group_reshaper_model_present_flag等于0指定片组标头中不存在tile_group_reshaper_model()。当不存在tile_group_reshaper_model_present_flag时，将其推断为等于0。

[0176] tile_group_reshaper_enabled_flag等于1指定对当前片组启用整形器。tile_group_reshaper_enabled_flag等于0指定对当前片组不启用整形器。当不存在tile_group_reshaper_enable_flag时，将其推断为等于0。

[0177] tile_group_reshaper_chroma_residual_scale_flag等于1指定对当前片组启用色度残差缩放。tile_group_reshaper_chroma_residual_scale_flag等于0指定对当前片组不启用色度残差缩放。当不存在tile_group_reshaper_chroma_residual_scale_flag时，将其推断为等于0。

[0178] 添加tile_group_reshaper_model()语法

[0179] reshape_model_min_bin_idx指定将要用于整形器构建过程的最小二进制码(或段)索引。reshape_model_min_bin_idx的值应在0到MaxBinIdx(包括端值)范围内。MaxBinIdx的值应等于15。

[0180] reshape_model_delta_max_bin_idx指定最大允许的二进制码(或段)索引MaxBinIdx减去将要用于整形器构建过程的最大位索引。将reshape_model_max_bin_idx的值设置为等于MaxBinIdx-reshape_model_delta_max_bin_idx。

[0181] reshaper_model_bin_delta_abs_cw_prec_minus1加1指定用于表示语法reshape_model_bin_delta_abs_CW[i]的比特数。

[0182] reshape_model_bin_delta_abs_CW[i]指定第i个二进制码的绝对增量(delta)码字值。

[0183] reshaper_model_bin_delta_sign_CW_flag[i]指定reshape_model_bin_delta_abs_CW[i]的符号,如下:

[0184] -如果reshape_model_bin_delta_sign_CW_flag[i]等于0,则对应的变量RspDeltaCW[i]为正值。

[0185] -否则(reshape_model_bin_delta_sign_CW_flag[i]不等于0),则对应的变量RspDeltaCW[i]为负值。

[0186] 当reshape_model_bin_delta_sign_CW_flag[i]不存在时,将其推断为等于0。

[0187] 变量 $\mathrm{RspDeltaCW}\ [i] = (1 - 2 * \mathrm{reshape\_model\_bin\_delta\_sign\_CW}\ [i]) * \mathrm{reshape\_model\_bin\_delta\_abs\_CW}\ [i];$

[0188] 变量RspCW[i]按以下步骤推导:

[0189] 将变量OrgCW设置为等于$(1 << \mathrm{BitDepth}_Y)/(\mathrm{MaxBinIdx}+1)$。

[0190] -如果reshaper_model_min_bin_idx<=i<=reshaper_model_max_bin_idx,则RspCW[i]=OrgCW+RspDeltaCW[i]。

[0191] -否则,则RspCW[i]=0。

[0192] 如果$\mathrm{BitDepth}_Y$的值等于10,则RspCW[i]的值应在32到2*OrgCW-1的范围内。

[0193] 变量InputPivot[i](i在范围0到MaxBinIdx+1中(包括端值))推导如下

[0194] InputPivot[i]=i*OrgCW

[0195] 变量ReshapePivot[i](i在范围0到MaxBinIdx+1中(包括端值))、变量ScaleCoef[i]和InvScaleCoeff[i](i在范围为0到MaxBinIdx中(包括端值))推导如下:

```
        shiftY = 14
        ReshapePivot[ 0 ] = 0;
        for( i = 0; i <= MaxBinIdx ; i++) {
          ReshapePivot[ i + 1 ] = ReshapePivot[ i ] + RspCW[ i ]
          ScaleCoef[ i ] = ( RspCW[ i ] * (1 << shiftY) + (1 << (Log2(OrgCW) - 1)))
```

[0196] `>> (Log2(OrgCW))`

```
          if ( RspCW[ i ] == 0 )
            InvScaleCoeff[ i ] = 0
          else
            InvScaleCoeff[ i ] = OrgCW * (1 << shiftY) / RspCW[ i ]
      }
```

[0197] 变量ChromaScaleCoef[i](i在范围0到MaxBinIdx中(包括端值))推导如下：

[0198] ChromaResidualScaleLut[64]＝{16384,16384,16384,16384,16384,16384,16384,8192,8192,8192,8192,5461,5461,5461,5461,4096,4096,4096,4096,3277,3277,3277,3277,2731,2731,2731,2731,2341,2341,2341,2048,2048,2048,1820,1820,1820,1638,1638,1638,1638,1489,1489,1489,1489,1365,1365,1365,1365,1260,1260,1260,1260,1170,1170,1170,1170,1092,1092,1092,1092,1024,1024,1024,1024}；

[0199] shiftC＝11

[0200] -如果(RspCW[i]＝＝0)

[0201] ChromaScaleCoef[i]＝(1<<shiftC)

[0202] 否则(RspCW[i]!＝0),ChromaScaleCoef[i]＝ChromaResidualScaleLut[RspCW[i]>>1]

[0203] 8.2.4 ILR的使用

[0204] 在编码器侧,首先将每个图片(或片组)转换到整形域。并且所有编解码过程都在整形域中执行。对于帧内预测,相邻块在整形域中;对于帧间预测,首先将参考块(从解码图片缓冲器的原始域生成)转换到整形域。然后生成残差并将其编解码到比特流。

[0205] 在整个图片(或片组)完成编码/解码之后,将整形域中的样点转换到原始域,然后应用去块滤波器和其他滤波器。

[0206] 在以下情况下,禁用对预测信号的前向整形：

[0207] ○当前块是帧内编解码的

[0208] ○当前块被编解码为CPR(当前图片参考,也称为帧内块复制,IBC)

[0209] ○当前块被编解码为组合帧间-帧内模式(CIIP),并且对帧内预测块禁用前向整形

[0210] 9.双向光流(Bi-directional optical flow,BIO或BDOF)

[0211] 9.1 BIO的概述和分析

[0212] 在BDOF(又名BIO)中,首先执行运动补偿以生成当前块的第一预测(在每个预测方向上)。第一预测用于推导块内每个子块/像素的空间梯度、时间梯度和光流,然后使用其生

成第二预测，例如，子块/像素的最终预测。细节描述如下。

[0213] 双向光流方法(BIO)是在双向预测的分块运动补偿基础上执行的分样点运动细化。在一些实施方式中，样点级别运动细化不使用信令。

[0214] 设$I^{(k)}$为块运动补偿之后参考k(k=0,1)的亮度值，并且$\theta I^{(k)}/\theta x$和$\theta I^{(k)}/\theta y$分别表示$I^{(k)}$梯度的水平分量和垂直分量。假设光流是有效的，则运动矢量场(v_x,v_y)由下式给出：

[0215] $$\theta I^{(k)}/\theta t+v_x\theta I^{(k)}/\theta x+v_y\theta I^{(k)}/\theta y=0. \quad (15)$$

[0216] 将此光流公式与每个样点运动轨迹的埃尔米特插值相结合，得到唯一的三阶多项式，该三阶多项式最后匹配函数值$I^{(k)}$和其导数$\theta I^{(k)}/\theta x$，$\theta I^{(k)}/\theta y$两者。该三阶多项式在t=0时的值是BIO预测：

[0217] $$pred_{BIO}=1/2\cdot(I^{(0)}+I^{(1)}+v_x/2\cdot(\tau_1\theta I^{(1)}/\theta x-\tau_0\theta I^{(0)}/\theta x)+v_y/2\cdot(\tau_1\theta I^{(1)}/\theta y-\tau_0\theta I^{(0)}/\theta y)). \quad (16)$$

[0218] 图8示出了双向光流(BIO)方法中的示例光流轨迹。此处，τ_0和τ_1表示到参考帧的距离。基于Ref0和Ref1的POC计算距离τ_0和τ_1：τ_0=POC(当前)-POC(Ref_0)，τ_1=POC(Ref_1)-POC(当前)。如果两个预测都来自相同的时间方向(两者都来自过去或都来自未来)，则符号是不同的(例如，$\tau_0\cdot\tau_1<0$)。在这种情况下，仅当预测不是来自相同的时刻(例如，$\tau_0\neq\tau_1$)时才应用BIO。两个参考区域都具有非零运动(例如，MVx_0，MVy_0，MVx_1，$MVy_1\neq0$)并且块运动矢量与时间距离成比例(例如，$MVx_0/MVx_1=MVy_0/MVVy_1=-\tau_0/\tau_1$)。

[0219] 通过最小化点A和B中的值之间的差Δ来确定运动矢量场(v_x,v_y)。图8示出了运动轨迹和参考帧平面的交叉的示例。模式仅使用Δ的局部泰勒展开的第一线性项：

[0220] $$\Delta=(I^{(0)}-I^{(1)}{}_0+v_x(\tau_1\theta I^{(1)}/\theta x+\tau_0\theta I^{(0)}/\theta x)+v_y(\tau_1\theta I^{(1)}/\theta y+\tau_0\theta I^{(0)}/\theta y)) \quad (17)$$

[0221] 上述公式中的所有值都取决于样点位置，表示为(i′,j′)。假设运动在局部周围区域是一致的，可以在以当前预测点(i,j)为中心的(2M+1)×(2M+1)的方形窗口Ω内最小化Δ，其中M等于2：

[0222] $$(v_x,v_y)=\underset{v_x,v_y}{\operatorname{argmin}}\sum_{[i',j]\in\Omega}\Delta^2[i',j'] \quad (18)$$

[0223] 对于该优化问题，JEM使用简化方法，首先在垂直方向上进行最小化，然后在水平方向上进行最小化。结果如下：

[0224] $$v_x=(s_1+r)>m?\,clip3\left(-thBIO,thBIO,-\frac{s_3}{(s_1+r)}\right):0 \quad (19)$$

[0225] $$v_y=(s_5+r)>m?\,clip3\left(-thBIO,thBIO,-\frac{s_6-v_xs_2/2}{(s_5+r)}\right):0 \quad (20)$$

[0226] 其中，

[0227] $$s_1=\sum_{[i',j]\in\Omega}\left(\tau_1\partial I^{(1)}/\partial x+\tau_0\partial I^{(0)}/\partial x\right)^2;\ s_3=\sum_{[i',j]\in\Omega}\left(I^{(1)}-I^{(0)}\right)\left(\tau_1\partial I^{(1)}/\partial x+\tau_0\partial I^{(0)}/\partial x\right);$$
$$s_2=\sum_{[i',j]\in\Omega}\left(\tau_1\partial I^{(1)}/\partial x+\tau_0\partial I^{(0)}/\partial x\right)\left(\tau_1\partial I^{(1)}/\partial y+\tau_0\partial I^{(0)}/\partial y\right); \quad (21)$$
$$s_5=\sum_{[i',j]\in\Omega}\left(\tau_1\partial I^{(1)}/\partial y+\tau_0\partial I^{(0)}/\partial y\right)^2;\ s_6=\sum_{[i',j]\in\Omega}\left(I^{(1)}-I^{(0)}\right)\left(\tau_1\partial I^{(1)}/\partial y+\tau_0\partial I^{(0)}/\partial y\right)$$

[0228] 为了避免除以零或非常小的值，在公式19和20中可以引入正则化参数r和m，其中：

[0229] $r=500\cdot 4^{d-8}$ (22)

[0230] $m=700\cdot 4^{d-8}$ (23)

[0231] 其中，d是视频样点的比特深度。

[0232] 为了使BIO的存储器访问与常规双向预测运动补偿保持相同，仅针对当前块内的位置计算所有预测和梯度值$I^{(k)}$、$\theta I^{(k)}/\theta x$、$\theta I^{(k)}/\theta y$。图9A示出了块900外部的访问位置的示例。如图9A所示，在公式(17)中，以在预测块的边界上的当前预测点为中心的(2M+1)×(2M+1)方形窗口Ω需要访问块外部的位置。在JEM中，将块外部的$I^{(k)}$、$\theta I^{(k)}/\theta x$、$\theta I^{(k)}/\theta y$的值设置为等于块内最近的可用值。例如，这可以实施为填充区域1901，如图9B所示。

[0233] 利用BIO，可以针对每个样点细化运动场。为了降低计算复杂度，在JEM中使用基于块的BIO设计。可以基于4×4的块计算运动细化。在基于块的BIO中，可以聚合4×4的块中的所有样点的公式21中的s_n的值，然后将s_n的聚合值用于推导4×4块的BIO运动矢量偏移。更具体地，以下公式可以用于基于块的BIO推导：

[0234]
$$s_{1,b_k}=\sum_{(x,y)\in b_k}\sum_{[i',j]\in\Omega(x,y)}\left(\tau_1\partial I^{(1)}/\partial x+\tau_0\partial I^{(0)}/\partial x\right)^2;s_{3,b_k}=\sum_{(x,y)\in b_k}\sum_{[i',j]\in\Omega}\left(I^{(1)}-I^{(0)}\right)\left(\tau_1\partial I^{(1)}/\partial x+\tau_0\partial I^{(0)}/\partial x\right);$$
$$s_{2,b_k}=\sum_{(x,y)\in b_k}\sum_{[i',j]\in\Omega}\left(\tau_1\partial I^{(1)}/\partial x+\tau_0\partial I^{(0)}/\partial x\right)\left(\tau_1\partial I^{(1)}/\partial y+\tau_0\partial I^{(0)}/\partial y\right); \quad (24)$$
$$s_{5,b_k}=\sum_{(x,y)\in b_k}\sum_{[i',j]\in\Omega}\left(\tau_1\partial I^{(1)}/\partial y+\tau_0\partial I^{(0)}/\partial y\right)^2;s_{6,b_k}=\sum_{(x,y)\in b_k}\sum_{[i',j]\in\Omega}\left(I^{(1)}-I^{(0)}\right)\left(\tau_1\partial I^{(1)}/\partial y+\tau_0\partial I^{(0)}/\partial y\right)$$

[0235] 其中，bk表示属于预测块的第k个4×4块的样点集。将公式19和20中的s_n替换为$((s_n,b_k)>>4)$，以推导相关联的运动矢量偏移。

[0236] 在一些情景下，由于噪音或不规则运动，BIO的MV团(MV regiment)可能不可靠。因此，在BIO中，MV团的大小被阈值裁剪。基于当前图片的参考图片是否都来自一个方向来确定阈值。例如，如果当前图片的所有参考图片都来自一个方向，则将阈值的值设置为$12\times 2^{14-d}$；否则，将其设置为$12\times 2^{13-d}$。

[0237] 可以使用与HEVC运动补偿过程(例如，2D可分离有限脉冲响应(FIR))一致的操作通过运动补偿插值来同时计算BIO的梯度。在一些实施例中，该2D可分离FIR的输入是与运动补偿过程和根据块运动矢量的分数部分得到的分数位置(fracX,fracY)相同的参考帧样点。对于水平梯度$\theta I/\theta x$，首先使用与具有去缩放移位d-8的分数位置fracY相对应的BIOfilterS垂直插值信号。然后在水平方向上应用梯度滤波器BIOfilterG，该BIOfilterG与具有去缩放移位18-d的分数位置fracX相对应。对于垂直梯度$\theta I/\theta y$，首先使用与具有去缩放移位d-8的分数位置fracY相对应的BIOfilterG垂直应用梯度滤波器。然后在水平方向上使用BIOfilterS执行信号位移，该BIOfilterS与具有去缩放移位18-d的分数位置fracX相对应。用于梯度计算的插值滤波器BIOfilterG和用于信号位移的插值滤波器BIOfilterS的长度可以较短(例如，6抽头)，以保持合理的复杂度。表3示出了可以用于BIO中块运动矢量的不同分数位置的梯度计算的滤波器的示例。表4示出了可以用于BIO中预测信号生成的插值滤波器的示例。

[0238] 表3：用于BIO中梯度计算的示例性滤波器

[0239]

分数像素位置	梯度的插值滤波器(BIOfilterG)

0	{8,-39,-3,46,-17,5}
1/16	{8,-32,-13,50,-18,5}
1/8	{7,-27,-20,54,-19,5}
3/16	{6,-21,-29,57,-18,5}
1/4	{4,-17,-36,60,-15,4}
5/16	{3,-9,-44,61,-15,4}
3/8	{1,-4,-48,61,-13,3}
7/16	{0,1,-54,60,-9,2}
1/2	{-1,4,-57,57,-4,1}

[0240] 表4:用于BIO中预测信号生成的示例性插值滤波器

[0241]

分数像素位置	预测信号的插值滤波器(BIOfilterS)
0	{0,0,64,0,0,0}
1/16	{1,-3,64,4,-2,0}
1/8	{1,-6,62,9,-3,1}
3/16	{2,-8,60,14,-5,1}
1/4	{2,-9,57,19,-7,2}
5/16	{3,-10,53,24,-8,2}
3/8	{3,-11,50,29,-9,2}
7/16	{3,-11,44,35,-10,3}
1/2	{3,-10,35,44,-11,3}

[0242] 在JEM中,当两个预测来自不同的参考图片时,BIO可以应用于所有双向预测块。当为CU启用局部照明补偿(LIC)时,可以禁用BIO。

[0243] 在一些实施例中,OBMC在正常MC过程之后应用于块。为了降低计算复杂度,在OBMC过程中可以不应用BIO。这意味着BIO仅在使用其自身的MV时才应用于块的MC过程,并且在OBMC过程中使用相邻块的MV时不应用于MC过程。

[0244] 10利用光流的预测细化(PROF)的示例

[0245] 在一些实施例中,可以利用光流来细化基于子块的仿射运动补偿预测。在执行基于子块的仿射运动补偿之后,通过添加由光流公式推导的差来细化预测样点,这被称为利用光流的预测细化(PROF)。这样的技术可以在不增加存储器访问带宽的情况下以像素级别粒度实现帧间预测。

[0246] 为了实现更精细的运动补偿粒度,该贡献提出了利用光流来细化基于子块的仿射运动补偿预测的方法。在执行基于子块的仿射运动补偿之后,通过添加由光流公式推导的差来细化亮度预测样点。将PROF描述为以下四个步骤。

[0247] 步骤1)执行基于子块的仿射运动补偿以生成子块预测I(i,j)。

[0248] 步骤2)使用3抽头滤波器[-1,0,1]在每个样点位置计算子块预测的空域梯度$g_x(i,j)$和$g_y(i,j)$。

[0249] $g_x(i,j)=I(i+1,j)-I(i-1,j)$

[0250] $g_y(i,j)=I(i,j+1)-I(i,j-1)$

[0251] 子块预测在每侧扩展一个像素以进行梯度计算。为了减少存储器带宽和复杂度,

扩展边界上的像素从参考图片中最近的整数像素位置复制。因此，避免了对填充区域的附加插值。

[0252] 步骤3)通过光流公式计算亮度预测细化。

[0253] $\Delta I(i,j)=g_x(i,j)*\Delta v_x(i,j)+g_y(i,j)*\Delta v_y(i,j)$

[0254] 此处，$\Delta v(i,j)$是对采样位置(i,j)计算出的像素MV(由v(i,j)表示)与像素(i,h)所属的子块的子块MV之间的差，如图10所示。

[0255] 由于仿射模型参数和相对于子块中心的像素位置从子块到子块没有变化，因此可以对第一子块计算$\Delta v(i,j)$，并将$\Delta v(i,j)$重用于相同CU中的其他子块。令x和y为从像素位置到子块中心的水平和垂直偏移，则可以通过以下公式推导$\Delta v(x,y)$：

[0256] $$\begin{cases}\Delta v_x(x,y)=c*x+d*y\\ \Delta v_y(x,y)=e*x+f*y\end{cases}$$

[0257] 对于4参数仿射模型，

[0258] $$\begin{cases}c=f=\dfrac{v_{1x}-v_{0x}}{w}\\ e=-d=\dfrac{v_{1y}-v_{0y}}{w}\end{cases}$$

[0259] 对于6参数仿射模型

[0260] $$\begin{cases}c=\dfrac{v_{1x}-v_{0x}}{w}\\ d=\dfrac{v_{2x}-v_{0x}}{h}\\ e=\dfrac{v_{1y}-v_{0y}}{w}\\ f=\dfrac{v_{2y}-v_{0y}}{h}\end{cases}$$

[0261] 此处，(v_{0x},v_{0y})、(v_{1x},v_{1y})、(v_{2x},v_{2y})是左上、右上和左下控制点运动矢量，w和h是CU的宽度和高度。

[0262] 步骤4)最后，将亮度预测细化添加到子块预测I(i,j)。按照以下公式生成最终预测I'：

[0263] $I'(i,j)=I(i,j)+\Delta I(i,j)$

[0264] 11现有实施方式的缺点

[0265] 非线性ALF(NLALF)、BIO(也称为BDOF)和/或PROF设计具有以下问题：

[0266] (1)对于BDOF或PROF，推导的应用于预测样点的偏移值可能与原始样点相距太远，从而导致较大的残差。

[0267] (2)应用了2-D滤波，其对于具有强水平或垂直模式的序列可能是次最优的。

[0268] 12视频编解码中样点细化和滤波的示例性方法

[0269] 当前公开的技术的实施例克服了现有实施方式的缺点，从而提供了具有更高编解码效率的视频编解码。在针对各种实施方式描述的以下示例中阐明了基于所公开的技术的用于视频编解码的样点细化和滤波方法可以增强现有和未来的视频编解码标准。下面提供

的所公开的技术的示例解释了一般概念，并且不意味着被解释为限制性的。在示例中，除非明确指出相反，否则可以对这些示例中描述的各种特征进行组合。

[0270] 注意，在下面的示例中提到的“滤波方法”可以指代自适应环路滤波器/重建后滤波器(诸如双边滤波器\扩散滤波器等)。

[0271] 1.裁剪可以应用于相邻像素的加权平均，而不是对每个相邻像素和当前像素之间的差进行裁剪。

[0272] a.在一些实施例中，裁剪可以如下进行，其中K是裁剪操作。

[0273] $$O(x,y)=I(x,y)+K\left(\sum_{(i,j)\neq(0,0)}w(i,j).\left(I(x+i,y+j)-I(x,y)\right)\right)$$

[0274] 2.可以利用一维滤波方法。

[0275] a.在一些实施例中，1-D滤波器可以仅使用水平相邻样点。

[0276] i.在一些实施例中，可以用以下方式表示滤波过程：

[0277] $$O(x,y)=\sum_{(i,j)}w(i,j)*\sum_{si=-T}^{T}ws(k)*I(x+i+si,y+j)$$

[0278] ii.对于不同的像素，样点数/滤波器抽头数(例如，T)可以不同。例如，对于CTU边界处的像素，不使用不在当前CTU内的像素。

[0279] b.在一些实施例中，短抽头滤波器可以仅使用垂直相邻样点。

[0280] $$O(x,y)=\sum_{(i,j)}w(i,j)*\sum_{si=-T}^{T}ws(k)*I(x+i,y+j+si)$$

[0281] i.对于不同的像素，T可以不同。例如，对于CTU边界处的像素，不使用不在当前CTU内的像素。

[0282] c.在一些实施例中，一维抽头滤波器可以仅使用沿除水平/垂直方向之外的一个方向上的相邻样点。

[0283] d.在一些实施例中，对于不同类别的像素，ws(si)可以不同。

[0284] e.在一些实施例中，对于不同类别的像素，ws(si)可以相同。

[0285] f.在一些实施例中，可以将ws(si)信令通知解码器。

[0286] g.在一些实施例中，ws(si)可以是预定义的。

[0287] 3.可以利用一维非线性自适应环路滤波方法。

[0288] a.在一些实施例中，短抽头滤波器可以仅使用水平相邻样点。

[0289] i.在一些实施例中，非线性滤波方法可以用水平相邻样点以以下方式表示：

[0290]
$$O'(x,y) = I(x,y) + \sum_{(i,j)\neq(0,0)} w(i,j).K\left(\sum_{si=-T}^{T} ws(si) * I(x+i+si, y+j) - I(x,y), k(i,j)\right)$$

[0291] ii.对于不同的像素,T可以不同。例如,对于CTU边界处的像素,不使用不在当前CTU内的像素。

[0292] b.在一些实施例中,一维非线性滤波器可以仅使用垂直相邻样点。

[0293] i.在一些实施例中,非线性滤波方法可以用垂直相邻样点以以下方式表示:

[0294]
$$O'(x,y) = I(x,y) + \sum_{(i,j)\neq(0,0)} w(i,j).K\left(\sum_{si=-T}^{T} ws(si) * I(x+i, y+j+si) - I(x,y), k(i,j)\right)$$

[0295] ii.对于不同的像素,T可以不同。例如,对于CTU边界处的像素,不使用不在当前CTU内的像素。

[0296] c.在一些实施例中,一维非线性抽头滤波器可以仅使用沿除水平/垂直方向之外的一个方向的相邻样点。

[0297] d.在一些实施例中,对于不同类别的像素,ws(si)可以不同。

[0298] e.在一些实施例中,对于不同类别的像素,ws(si)可以相同。

[0299] f.在一些实施例中,可以将ws(si)信令通知解码器。

[0300] g.在一些实施例中,ws(si)可以是预定义的。

[0301] 4.在双边滤波器或哈达玛(Hadamard)滤波器中,可以沿不同方向对像素进行滤波。

[0302] a.在一些实施例中,可以取决于梯度信息,沿不同方向对像素进行滤波。

[0303] b.在一些实施例中,对于每个M×N子块,可以计算水平梯度和垂直梯度,并且如果绝对水平梯度之和大于或/和等于绝对垂直梯度之和,则可以对子块内的像素进行水平滤波。

[0304] i.可替代地,可以对子块内的像素进行垂直滤波。

[0305] c.在一些实施例中,对于每个M×N子块,可以计算水平梯度和垂直梯度,并且如果绝对水平梯度之和小于或/和等于绝对垂直梯度之和,则可以对子块内的像素进行垂直滤波。

[0306] i.可替代地,可以对子块内的像素进行水平滤波。

[0307] d.在一些实施例中,对于每个M×N子块,可以计算水平梯度、垂直梯度、45度对角梯度和135度对角梯度,可以沿绝对梯度之和最大的方向对子块内的像素进行滤波。

[0308] e.在一些实施例中,对于每个M×N子块,可以计算水平梯度、垂直梯度、45度对角

梯度和135度对角梯度,可以沿绝对梯度之和最小的方向对子块内的像素进行滤波。

[0309] 5.在BODF中,可以在将梯度用于样点细化或/和MV偏移推导之前进行修改。

[0310] a.在一些实施例中,可以将在BDOF中计算的空域或/和时域梯度裁剪到范围[min,max]。

[0311] i.对于空域和时域梯度,变量min和max可以不同。

[0312] ii.变量min可以小于0,且变量max可以大于零。

[0313] iii.变量min和max可以取决于样点的输入比特深度。

[0314] iv.变量min和max可以取决于用于生成中间样点的插值滤波器。

[0315] v.可以信令通知变量min和max。

[0316] vi.变量min和max可以是预定义的。

[0317] b.在一些实施例中,可以通过非线性函数来修改在BDOF中计算的空域或/和时域梯度。

[0318] i.例如,可以使用逻辑sigmoid函数。

[0319] c.类似地,在PROF中,可以在将梯度用于样点细化或/和MV偏移推导之前进行修改。

[0320] 6.在BODF/PROF中,在细化的样点值或预测样点与其细化的样点值之间的差(或偏移)用于推导最终重建样点值之前,可以进一步修改细化的样点值或预测样点与其细化的样点值之间的差(或偏移)。

[0321] a.在BODF/PROF中,可以将裁剪操作加到预测样点及其细化样点值之间的差中。

[0322] b.在一些实施例中,该差可以被裁剪到范围[min,max]。

[0323] i.对于空域和时域梯度,变量min和max可以不同。

[0324] ii.变量min可以小于0,且变量max可以大于零。

[0325] iii.变量min和max可以取决于样点的输入比特深度。

[0326] iv.变量min和max可以取决于用于生成中间样点的插值滤波器。

[0327] v.可以信令通知变量min和max。

[0328] vi.变量min和max可以是预定义的。

[0329] c.此外,可替代地,一个预测样点的最终重建值可以取决于裁剪后的差。

[0330] d.此外,可替代地,一个预测样点的最终重建值可以取决于修改的细化的样点/修改的细化的差。

[0331] 7.以上方法中利用的裁剪参数可以以诸如序列级别、图片级别、CTU级别等的一些视频编解码单元级别信令通知。

[0332] a.在一些实施例中,可以在SPS/VPS/PPS/片组标头/CTU行/区域中信令通知它们。

[0333] b.可替代地,可以实时推导参数。

[0334] c.可替代地,可以根据量化参数、片组类型、编解码模式信息、整形参数等来推导参数。

[0335] 8.是否应用以上方法可以取决于颜色格式。

[0336] a.所提出的方法可以仅适用于一个颜色分量,诸如G或亮度颜色分量。

[0337] b.可替代地,所提出的方法可以适用于4:4:4颜色格式的所有颜色分量。

[0338] 可以在下面描述的方法的上下文中包含上述示例,例如方法1100和1150,其可以

在视频解码器或视频编码器处实现。

[0339] 图11A示出了用于视频处理的示例性方法的流程图。方法1100包括，在步骤1102，为当前视频块配置滤波过程，该滤波过程包括应用于当前视频块的相邻像素的加权平均的裁剪操作。

[0340] 在一些实施例中，滤波过程包括

[0341] $O(x,y)=I(x,y)+K(\sum_{(i,j)\neq(0,0)}w(i,j).(I(x+i,y+j)-I(x,y)))$，

[0342] 其中，I(x,y)是当前视频块的第(x,y)个像素，O(x,y)是第(x,y)个像素的输出值，w(i,j)是加权因子，并且K(•)是裁剪操作。

[0343] 方法1100包括，在步骤1104，基于滤波过程，从对应的比特流表示中重建当前视频块。

[0344] 在一些实施例中，在序列参数集(SPS)、视频参数集(VPS)、图片参数集(PPS)、片组标头、编解码树单元(CTU)行或CTU区域中信令通知裁剪操作的一个或多个参数。

[0345] 在一些实施例中，裁剪操作的一个或多个参数是实时推导的。

[0346] 在一些实施例中，基于当前视频块的一个或多个量化参数、一个或多个整形参数、片组类型或编解码模式来推导裁剪操作的一个或多个参数。

[0347] 在一些实施例中，滤波过程对应于自适应环路滤波过程或重建后滤波过程。在示例中，重建后滤波过程使用双边滤波器、哈达玛滤波器或扩散滤波器中的至少一个。

[0348] 图11B示出了用于视频处理的示例性方法的流程图。方法1150包括，在步骤1152，为当前视频块配置一维滤波过程。

[0349] 方法1150包括，在步骤1154，基于一维滤波过程，从对应的比特流表示中重建当前视频块。

[0350] 在一些实施例中，将一维滤波过程应用于当前视频块的水平相邻样点、垂直相邻样点、沿45度对角的相邻样点或沿135度对角的相邻样点中的至少一个。在示例中，应用于编解码树单元(CTU)边界处的像素的一维滤波过程的滤波器抽头的第一集合不同于应用于完全在CTU内的像素的一维滤波过程的滤波器抽头的第二集合。在另一示例中，滤波器抽头的第一集合或滤波器抽头的第二集合是预定义的或在对应的比特流表示中信令通知。

[0351] 在一些实施例中，基于梯度信息来配置一维滤波过程。在示例中，梯度信息包括当前视频块的M×N子块的多个水平梯度和多个垂直梯度。在另一示例中，多个水平梯度的绝对值之和大于或等于多个垂直梯度的绝对值之和。在又一示例中，多个水平梯度的绝对值之和小于或等于多个垂直梯度的绝对值之和。在又一示例中，将一维滤波过程应用于水平相邻样点。在又一示例中，将一维滤波过程应用于垂直相邻样点。

[0352] 在一些实施例中，梯度信息还包括用于M×N子块的多个45度对角梯度和用于M×N子块的多个135度对角梯度。在示例中，沿与多个水平梯度、垂直梯度、45度对角梯度和135度对角梯度的绝对值之和的最大值对应的方向应用一维滤波过程。在另一示例中，沿与多个水平梯度、垂直梯度、45度对角梯度和135度对角梯度的绝对值之和的最小值对应的方向应用一维滤波过程。

[0353] 在一些实施例中，基于梯度信息的修改来配置一维滤波过程。在示例中，修改包括对梯度信息应用范围为[min,max]的裁剪操作。在另一示例中，min和max配置有用于空间梯度的第一值集合，并且其中min和max配置有用于时间梯度的第二值集合，该第二值集合不

同于第一值集合。在又一示例中，min＜0并且max＞0。在又一示例中，基于当前视频块的样点的输入比特深度来选择min和max的值。在又一个示例中，min和max的值是预定义的或在对应的比特流表示中信令通知。

[0354] 在一些实施例中，修改包括对梯度信息应用非线性函数。在示例中，非线性函数是逻辑sigmoid函数。

[0355] 在一些实施例中，一维滤波过程对应于自适应环路滤波过程或重建后滤波过程。在示例中，重建后滤波过程使用双边滤波器、哈达玛滤波器或扩散滤波器中的至少一个。

[0356] 13.所公开的技术的示例实施方式

[0357] 图12是视频处理装置1200的框图。装置1200可用于实现本文所描述的一个或多个方法。装置1200可以实施在智能手机、平板电脑、计算机、物联网(IoT)接收器等中。装置1200可以包括一个或多个处理器1202、一个或多个存储器1204和视频处理硬件1206。(一个或多个)处理器1202可以被配置为实现在本文中描述的一种或多种方法(包括但不限于方法1100和1150)。可以将(一个或多个)存储器1204用于存储用于实现本文所描述的方法和技术的数据和代码。视频处理硬件1206可用于在硬件电路中实现本文档中描述的一些技术。

[0358] 在一些实施例中，视频编解码方法可以使用关于图12所描述的在硬件平台上实现的装置来实现。

[0359] 图13是示出其中可实现本文中所公开的各种技术的示例视频处理系统1300的框图。各种实施方式可以包括系统1300的一些或全部组件。系统1300可以包括用于接收视频内容的输入1302。视频内容可以以原始或未压缩的格式(例如8或10位多分量像素值)接收，或者可以以压缩或编码的格式接收。输入1302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)和无线接口(诸如Wi-Fi或蜂窝接口)。

[0360] 系统1300可以包括可以实现本文中描述的各种编解码或编码方法的编解码组件1304。编解码组件1304可以减少从编解码组件1304的输入1302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，编解码技术有时称为视频压缩或视频转码技术。如组件1306所表示的，编解码组件1304的输出可以被存储或经由所连接的通信来发送。在输入1302处接收的视频的存储或传送的比特流(或编解码)表示可以被组件1308使用，以生成被发送到显示接口1310的像素值或可显示视频。从比特流表示中生成用户可见视频的过程有时称为视频解压缩。此外，尽管某些视频处理操作被称为“编解码”操作或工具，但是应当理解，在编码器处使用编解码工具或操作，并且将由解码器执行反向编解码结果的对应解码工具或操作。

[0361] 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)或高清多媒体接口(HDMI)或Displayport等。存储接口的示例包括SATA(串行高级技术附件)、PCI、IDE接口等。本文中描述的技术可以实施在各种电子设备中，诸如移动电话、膝上型计算机、智能电话或其他能够执行数字数据处理和/或视频显示的设备。

[0362] 图14是根据本技术的用于视频处理的方法1400的流程图表示。方法1400包括，在操作1410处，对于视频的当前块和视频的比特流表示之间的转换，将裁剪操作应用于当前块中的像素与该像素的相邻像素之间的差的加权平均。方法1400包括，在操作1420处，基于

裁剪操作的输出执行转换。

[0363] 在一些实施例中，将裁剪操作应用为：O(x,y)＝I(x,y)+K(∑$_{(i,j)\neq(0,0)}$w(i,j).(I(x+i,y+j)-I(x,y)))，其中I(x,y)是当前块的第(x,y)个像素，O(x,y)是第(x,y)个像素的输出值，w(i,j)是加权因子，并且K(•)是裁剪操作。在一些实施例中，在序列参数集(SPS)、视频参数集(VPS)、图片参数集(PPS)、片组标头、编解码树单元(CTU)行或CTU区域中信令通知裁剪操作的一个或多个参数。在一些实施例中，裁剪操作的一个或多个参数是实时推导的。在一些实施例中，基于当前视频块的一个或多个量化参数、一个或多个整形参数、片组类型或编解码模式来推导裁剪操作的一个或多个参数。

[0364] 图15是根据本技术的用于视频处理的方法1500的流程图表示。方法1500包括，在操作1510处，对于视频的当前块和视频的比特流表示之间的转换，基于当前块的相邻像素集合配置滤波过程。方法1500包括，在操作1520处，基于滤波过程的输出执行转换。

[0365] 在一些实施例中，滤波过程包括基于当前块的相邻像素集合的一维滤波操作。在一些实施例中，一维滤波操作包括非线性自适应环路滤波方法。

[0366] 在一些实施例中，相邻像素集合包括当前块的水平相邻像素。在一些实施例中，滤波过程包括：$O(x,y)=\sum_{(i,j)}w(i,j)*\sum_{si=-T}^{T}ws(si)*I(x+i+si,y+j)$，其中I(x,y)是当前块的第(x,y)个像素，O(x,y)是第(x,y)个像素的输出值，w(i,j)是加权因子，T表示要用于滤波过程的像素数，并且ws(si)表示与T相对应的滤波系数。在一些实施例中，滤波过程包括：$O(x,y)=I(x,y)+\sum_{(i,j)\neq(0,0)}w(i,j).K\left(\sum_{si=-T}^{T}ws(si)*I(x+i+si,y+j)-I(x,y),k(i,j)\right)$，其中I(x,y)是当前块的第(x,y)个像素，O(x,y)是第(x,y)个像素的输出值，w(i,j)是加权因子，K(•)是裁剪操作，T表示要用于滤波过程的像素数，ws(si)表示与T对应的滤波系数，并且k(i,j)表示裁剪参数。

[0367] 在一些实施例中，相邻像素集合包括当前块的垂直相邻像素。在一些实施例中，滤波过程包括：$O(x,y)=\sum_{(i,j)}w(i,j)*\sum_{si=-T}^{T}ws(si)*I(x+i,y+j+si)$，其中I(x,y)是当前块的第(x,y)个像素，O(x,y)是第(x,y)个像素的输出值，w(i,j)是加权因子，T表示要用于滤波过程的像素数，并且ws(si)表示与T对应的滤波系数。在一些实施例中，滤波过程包括：$O(x,y)=I(x,y)+\sum_{(i,j)\neq(0,0)}w(i,j).K\left(\sum_{si=-T}^{T}ws(si)*I(x+i,y+j+si)-I(x,y),k(i,j)\right)$，其中I(x,y)是当前块的第(x,y)个像素，O(x,y)是第(x,y)个像素的输出值，w(i,j)是加权因子，K(•)是裁剪操作，T表示要用于滤波过程的像素数，ws(si)表示与T对应的滤波系数，并且k(i,j)表示裁剪参数。

[0368] 在一些实施例中，T针对当前块中的不同像素而变化。在一些实施例中，在当前块的第(x,y)个像素位于当前块的边界的情况下，要用于滤波过程的像素数T不包括位于当前块外部的像素。在一些实施例中，对于当前块中的不同像素，ws(k)不同。在一些实施例中，对于当前块中的不同像素，ws(k)相同。在一些实施例中，在比特流表示中信令通知ws(k)。在一些实施例中，ws(k)是预定义的。

[0369] 在一些实施例中，一维滤波操作不包括沿水平或垂直方向应用滤波器。

[0370] 在一些实施例中，滤波过程包括沿不同方向应用滤波器。在一些实施例中，滤波器包括双边滤波器或哈达玛滤波器。在一些实施例中，基于与当前块相关联的梯度信息，将滤

波器应用于不同方向。在一些实施例中，该方法包括：对于当前块内的子块，确定子块中像素的水平梯度和垂直梯度；以及在绝对水平梯度之和大于或等于绝对垂直梯度之和的情况下，将滤波器水平应用于子块。在一些实施例中，该方法包括：对于当前块内的子块，确定子块中像素的水平梯度和垂直梯度；以及在绝对水平梯度之和大于或等于绝对垂直梯度之和的情况下，将滤波器垂直应用于子块。在一些实施例中，该方法包括：对于当前块内的子块，确定子块中像素的水平梯度和垂直梯度；以及在绝对水平梯度之和小于或等于绝对垂直梯度之和的情况下，将滤波器水平应用于子块。在一些实施例中，该方法包括：对于当前块内的子块，确定子块中像素的水平梯度和垂直梯度；以及在绝对水平梯度之和小于或等于绝对垂直梯度之和的情况下，将滤波器垂直应用于子块。在一些实施例中，该方法包括：对于当前块内的子块，确定子块中像素的水平梯度、垂直梯度、45度对角梯度和135度对角梯度；以及基于所确定的水平梯度、垂直梯度、45度对角梯度和135度对角梯度，将滤波器沿绝对梯度之和最大的方向应用于子块。在一些实施例中，该方法包括：对于当前块内的子块，确定子块中像素的水平梯度、垂直梯度、45度对角梯度和135度对角梯度；以及基于所确定的水平梯度、垂直梯度、45度对角梯度和135度对角梯度，将滤波器沿绝对梯度之和最大的方向应用于子块。

[0371] 在一些实施例中，滤波过程是否适用于当前块基于当前块的颜色格式。在一些实施例中，滤波过程适用于当前块的单个颜色分量。在一些实施例中，单个颜色分量是亮度分量或绿色分量。在一些实施例中，在当前块的颜色格式是4:4:4颜色格式的情况下，滤波过程适用于当前块的所有颜色分量。

[0372] 图16是根据本技术的用于视频处理的方法1600的流程图表示。方法1600包括，在操作1610处，对于视频的块和视频的比特流表示之间的转换，通过将裁剪操作应用于细化值，来对块中的预测样点的细化值进行修改。基于光流编解码过程的梯度值来推导细化值。裁剪操作的输出在范围内。方法1600包括，在操作1620处，基于修改的细化值来细化预测样点。方法1600还包括，在操作1630处，基于细化的预测样点执行转换。

[0373] 在一些实施例中，光流编解码过程包括具有光流过程的预测细化。在一些实施例中，光流编解码过程包括双向光流过程。

[0374] 在一些实施例中，基于样点的输入比特深度来确定范围。在一些实施例中，范围基于样点的空间梯度或时间梯度可变。在一些实施例中，范围是[min,max]，min小于0并且max大于0。在一些实施例中，范围基于用于生成与样点相关联的中间样点的插值滤波器。在一些实施例中，在比特流表示中信令通知范围。在一些实施例中，范围是预定义的。

[0375] 在一些实施例中，基于裁剪操作的输出来确定样点的最终重建值。在一些实施例中，基于样点的修改的预测值来确定样点的最终重建值。在一些实施例中，块以仿射运动进行编解码。在一些实施例中，对块中的多个样点中的每个推导细化值。

[0376] 图17是根据本技术的用于视频处理的方法1700的流程图表示。方法1700包括，在操作1710处，对于视频的块和视频的比特流表示之间的转换，修改光流编解码过程中的梯度值。方法1700还包括，在操作1720处，基于用于转换的修改的梯度值执行后续预测操作。在一些实施例中，光流编解码过程包括具有光流过程或双向光流过程的预测细化。在一些实施例中，后续预测操作包括样点细化操作或运动矢量偏移推导操作。

[0377] 在一些实施例中，修改梯度值包括对梯度值应用裁剪操作，并且其中裁剪操作的

输出在范围内。在一些实施例中，范围基于梯度值是预测样点的空间梯度还是时间梯度可变。在一些实施例中，范围是[min,max]，min小于0，并且max大于0。在一些实施例中，基于样点的输入比特深度来确定范围。在一些实施例中，范围基于用于生成与样点相关联的中间样点的插值滤波器。在一些实施例中，在比特流表示中用信号通知范围。在一些实施例中，范围是预定义的。在一些实施例中，修改梯度值包括将非线性函数应用于梯度值。在一些实施例中，非线性函数包括逻辑sigmoid函数。

[0378] 在一些实施例中，块以仿射运动进行编解码。在一些实施例中，对块中的多个样点的每个推导梯度值。

[0379] 在一些实施例中，在序列参数集(SPS)、视频参数集(VPS)、图片参数集(PPS)、片组标头、编解码树单元(CTU)行或CTU区域中信令通知裁剪操作的一个或多个参数。在一些实施例中，裁剪操作的一个或多个参数是实时推导的。在一些实施例中，基于当前视频块的一个或多个量化参数、一个或多个整形参数、片组类型或编解码模式来推导裁剪操作的一个或多个参数。

[0380] 在一些实施例中，修改是否适用于当前块基于当前块的颜色格式。在一些实施例中，修改适用于当前块的单个颜色分量。在一些实施例中，单个颜色分量是亮度分量或绿色分量。在一些实施例中，在当前块的颜色格式是4:4:4颜色格式的情况下，修改适用于当前块的所有颜色分量。

[0381] 在一些实施例中，光流编解码过程是其中将块中的运动建模为光流的过程。

[0382] 在一些实施例中，执行转换包括基于视频的当前块来生成比特流表示。在一些实施例中，执行转换包括从比特流表示生成视频的当前块。

[0383] 所公开技术的一些实施例包括做出决策或确定以启用视频处理工具或模式。在示例中，当启用视频处理工具或模式时，编码器将在视频块的处理中使用或实现该工具或模式，但是不一定基于该工具或模式的使用来修改产生的比特流。换言之，从视频块到视频的比特流表示的转换将在基于决策或确定启用视频处理工具或模式时使用视频处理工具或模式。在另一示例中，当启用视频处理工具或模式时，解码器将在知道已经基于视频处理工具或模式修改比特流的情况下处理比特流。换言之，将使用基于决策或确定而启用的视频处理工具或模式来执行从视频的比特流表示到视频块的转换。

[0384] 所公开技术的一些实施例包括作出决策或确定以禁用视频处理工具或模式。在示例中，当禁用视频处理工具或模式时，编码器将在视频块到视频的比特流表示的转换中不使用该工具或模式。在另一示例中，当禁用视频处理工具或模式时，解码器将在知道尚未使用基于决策或确定而启用的视频处理工具或模式修改比特流的情况下处理比特流。

[0385] 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，可以在从视频的像素表示到对应的比特流表示(或编解码表示)的转换期间应用视频压缩算法，反之亦然。如语法所定义，当前视频块的比特流表示可以例如对应于位于比特流内的不同位置中并置或散布的比特。例如，可以根据变换和编解码的误差残差值并且还使用标头中的比特和比特流中的其他字段来对宏块进行编码。此外，如以上解决方案中所描述的，在转换期间，解码器可以在已知存在或不存在某些字段的情况下基于确定来解析比特流。类似地，编码器可以确定是否包括某些语法字段，并通过从编解码表示中包括或排除语法字段来相应地生成编解码表示。

[0386] 从前述内容可以理解，本文已经出于说明的目的描述了当前所公开的技术的具体实施例，但是在不脱离本发明的范围的情况下可以做出各种修改。因此，除了所附权利要求之外，当前所公开的技术不受限制。

[0387] 本专利文档中描述的主题的实施方式和功能性操作可以在各种系统、数字电子电路中实施，或者在计算机软件、固件或硬件中实施，包括本说明书中所公开的结构及其结构等同物，或者以他们中的一个或多个的组合实施。本说明书中描述的主题的实施方式可以被实现为一个或多个计算机程序产品，即，在暂时性和非暂时性计算机可读介质上编码的计算机程序指令的一个或多个模块，以用于由数据处理装置执行或控制数据处理装置的操作。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的合成、或者它们中的一个或多个的组合。术语“数据处理单元”和“数据处理装置”包括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、计算机或者多个处理器或计算机。除了硬件之外，装置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、操作系统及其一个或多个的组合的代码。

[0388] 计算机程序(也称为程序、软件、软件应用、脚本或代码)可以用任何形式的编程语言(包括编译语言或解释语言)编写，并且可以以任何形式部署，包括作为独立程序或作为模块、组件、子程序或其他适合在计算环境中使用的单元。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)、专用于所讨论的程序的单个文件中、或多个协调文件(例如，存储一个或多个模块、子程序或部分代码的文件)中。计算机程序可以部署在一台或多台计算机上来执行，这些计算机位于一个站点或分布在多个站点并通过通信网络互连。

[0389] 本说明书中描述的处理和逻辑流可以由一个或多个可编程处理器执行，该一个或多个处理器执行一个或多个计算机程序，通过对输入数据进行操作并生成输出来执行功能。处理和逻辑流也可以由专用逻辑电路来执行，并且装置也可以实施为专用逻辑电路，例如，FPGA(现场可编程门阵列)或ASIC(专用集成电路)。

[0390] 例如，适用于执行计算机程序的处理器包括通用和专用微处理器、以及任何类型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和数据。计算机的基本元件是执行指令的处理器和存储指令和数据的一个或多个存储设备。通常，计算机还将包括一个或多个用于存储数据的大容量存储设备，例如，磁盘、磁光盘或光盘，或可操作地耦接到一个或多个大容量存储设备，以从其接收数据或向其传送数据，或两者兼有。然而，计算机不一定需要具有这样的设备。适用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EPROM、EEPROM和闪存设备。处理器和存储器可以由专用逻辑电路来补充，或合并到专用逻辑电路中。

[0391] 旨在将说明书与附图一起仅视为示例性的，其中示例性意味着示例。如本文所使用的，单数形式“一(a)”、“一个(an)”和“该(the)”也意图包括复数形式，除非上下文另外明确指出。另外，除非上下文另有明确说明，否则“或”的使用旨在包括“和/或”。

[0392] 虽然本专利文档包含许多细节，但不应将其解释为对任何发明或要求保护的范围的限制，而应解释为特定于特定发明的特定实施例的特征的描述。本专利文档在分离的实

施例的上下文描述的某些特征也可以在单个实施例中组合实现。相反，在单个实施例的上下文中描述的各种特征也可以在多个实施例中单独地实现，或在任何合适的子组合中实现。此外，虽然特征可以被描述为在某些组合中起作用，甚至最初这样要求保护，但在某些情况下，可以从组合中移除来自要求保护的组合的一个或多个特征，并且要求保护的组合可以指向子组合或子组合的变体。

[0393] 同样，尽管在附图中以特定顺序描述了操作，但这不应理解为要获得期望的结果必须按照所示的特定顺序或次序顺序来执行这些操作，或执行所有示出的操作。此外，本专利文档所述实施例中的各种系统组件的分离不应理解为在所有实施例中都需要这样的分离。

[0394] 仅描述了一些实施方式和示例，其他实施方式、增强和变体可以基于本专利文档中描述和说明的内容做出。

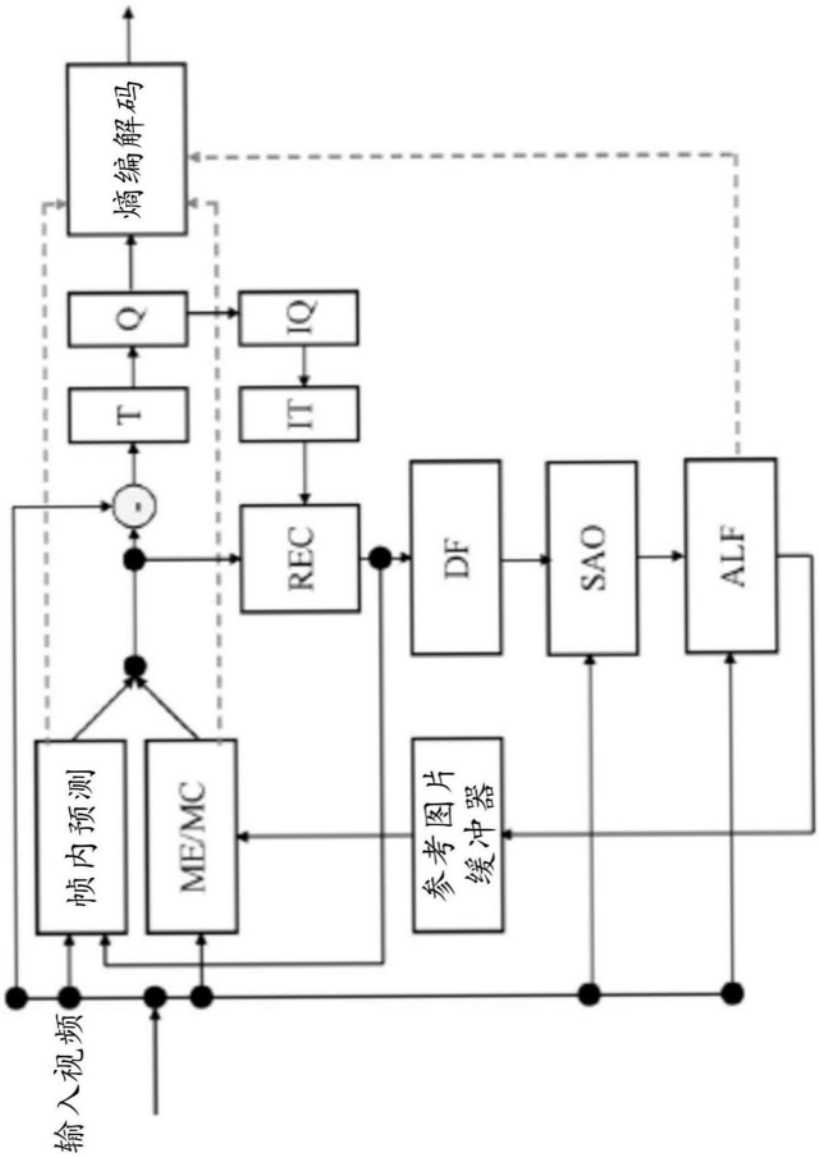
熵编解码
Q
IQ
T
IT
-
REC
DF
SAO
ALF
帧内预测
ME/MC
参考图片缓冲器
输入视频

图1

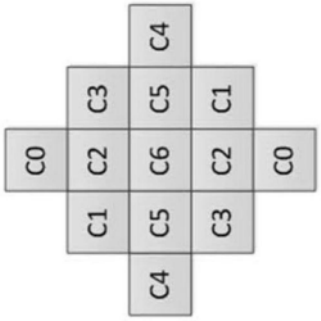
C4
C3
C5
C1
C0
C2
C6
C2
C0
C1
C5
C3
C4

图2A

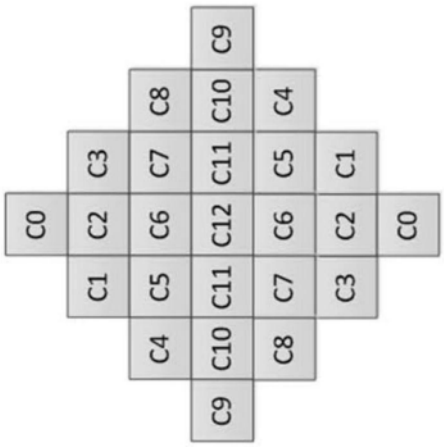
C9
C8 C10 C4
C3 C7 C11 C5 C1
C0 C2 C6 C12 C6 C2 C0
C1 C5 C11 C7 C3
C4 C10 C8
C9

图2B

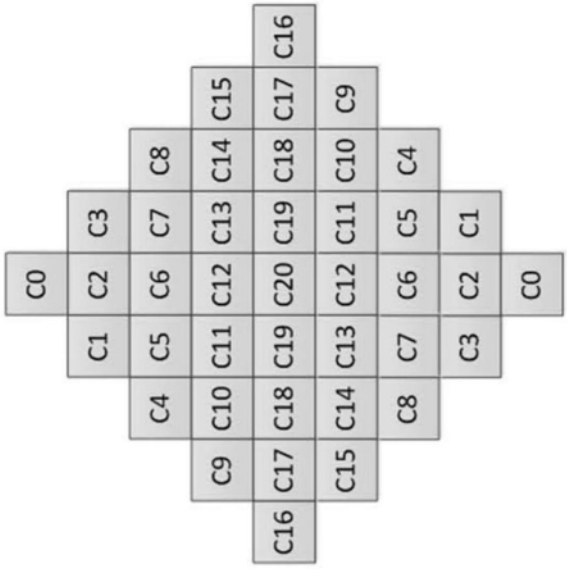
C16
C15 C17 C9
C8 C14 C18 C10 C4
C3 C7 C13 C19 C11 C5 C1
C0 C2 C6 C12 C20 C12 C6 C2 C0
C1 C5 C11 C19 C13 C7 C3
C4 C10 C18 C14 C8
C9 C17 C15
C16

图2C

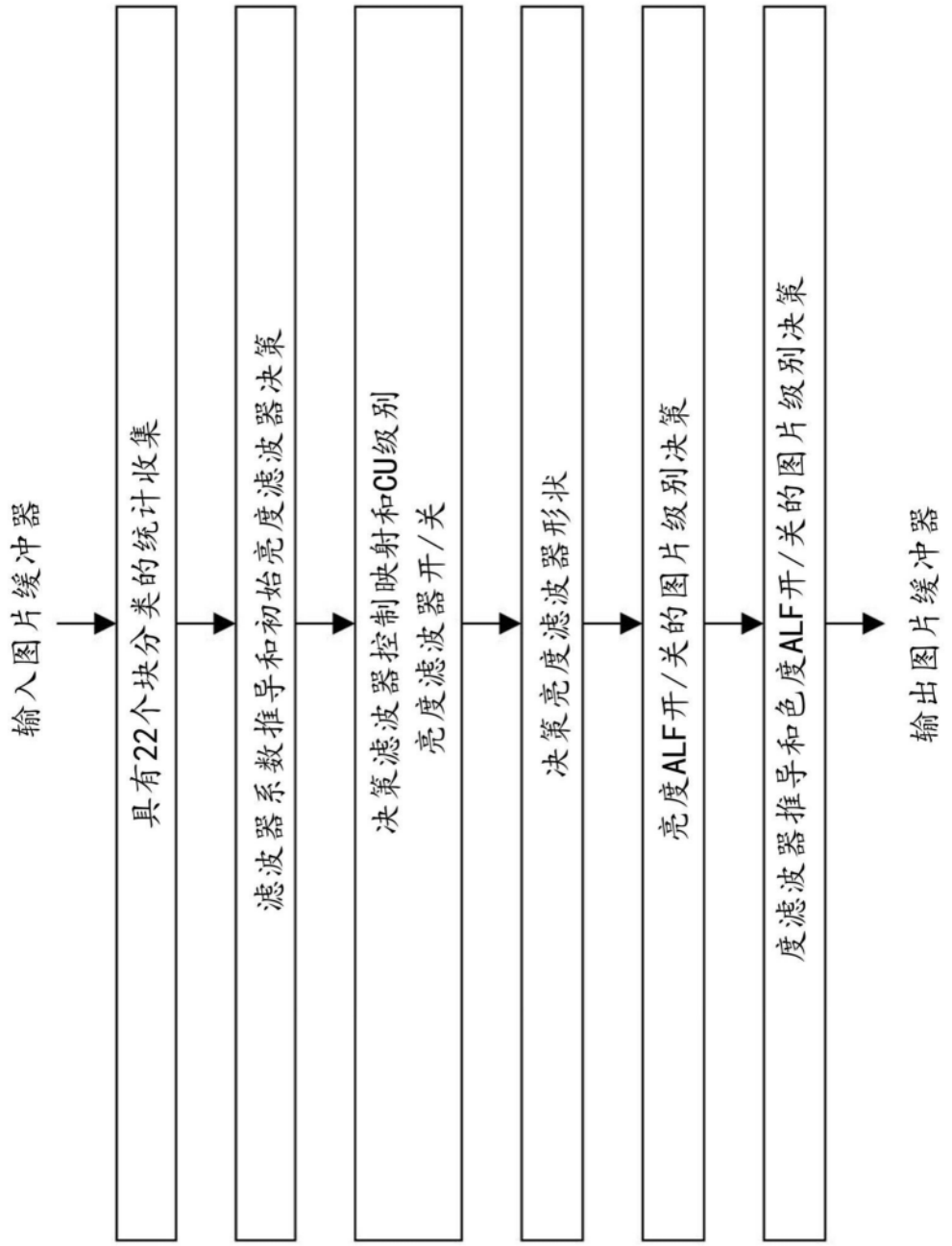
输入图片缓冲器
具有22个块分类的统计收集
滤波器系数推导和初始亮度滤波器决策
决策滤波器控制映射和CU级别
亮度滤波器开/关
决策亮度滤波器形状
亮度ALF开/关的图片级别决策
度滤波器推导和色度ALF开/关的图片级别决策
输出图片缓冲器

图3

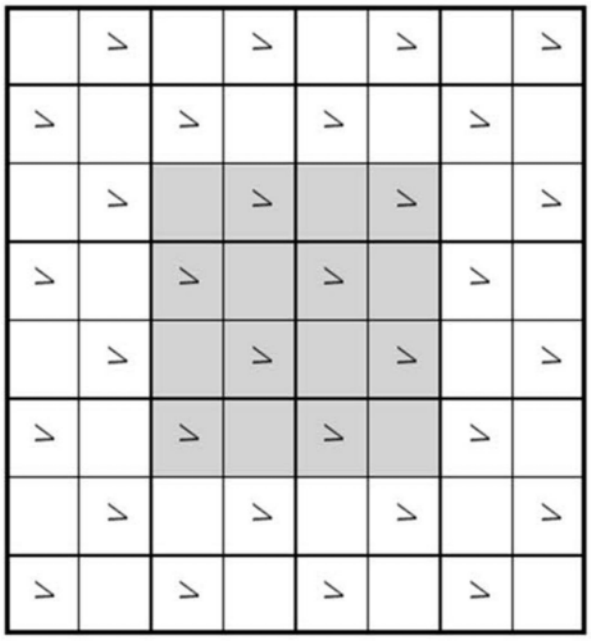

图4A

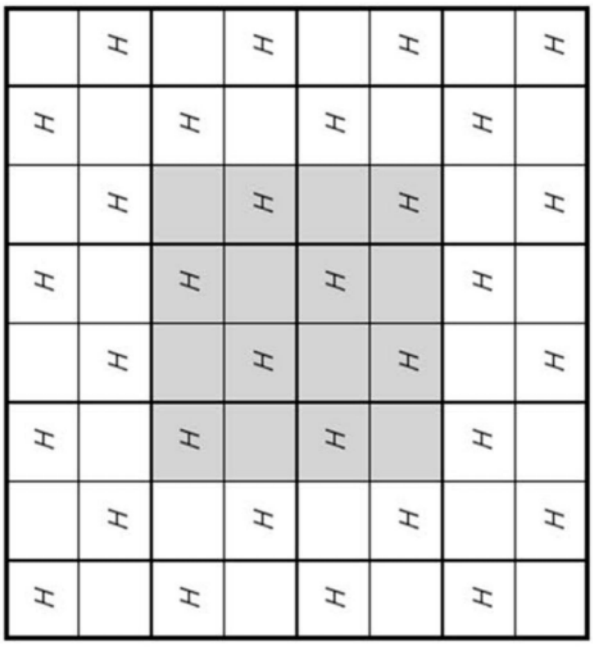

图4B

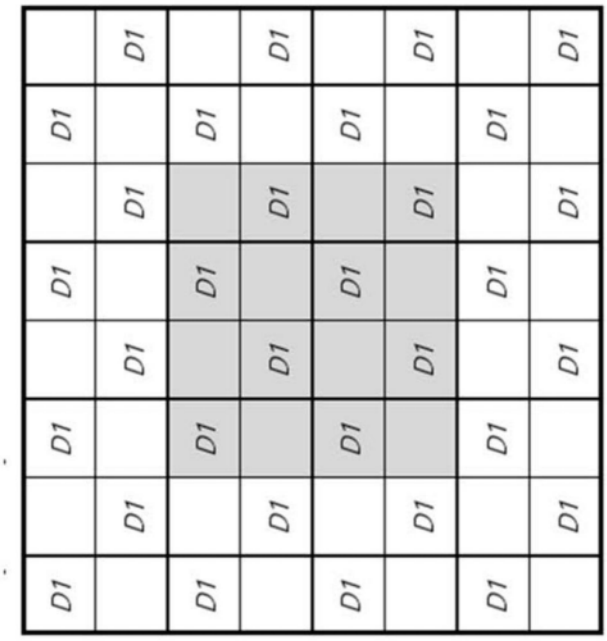

图4C

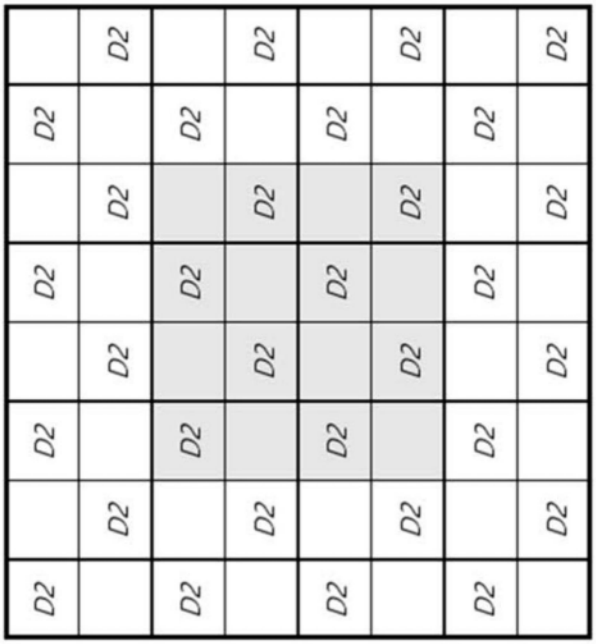

图4D

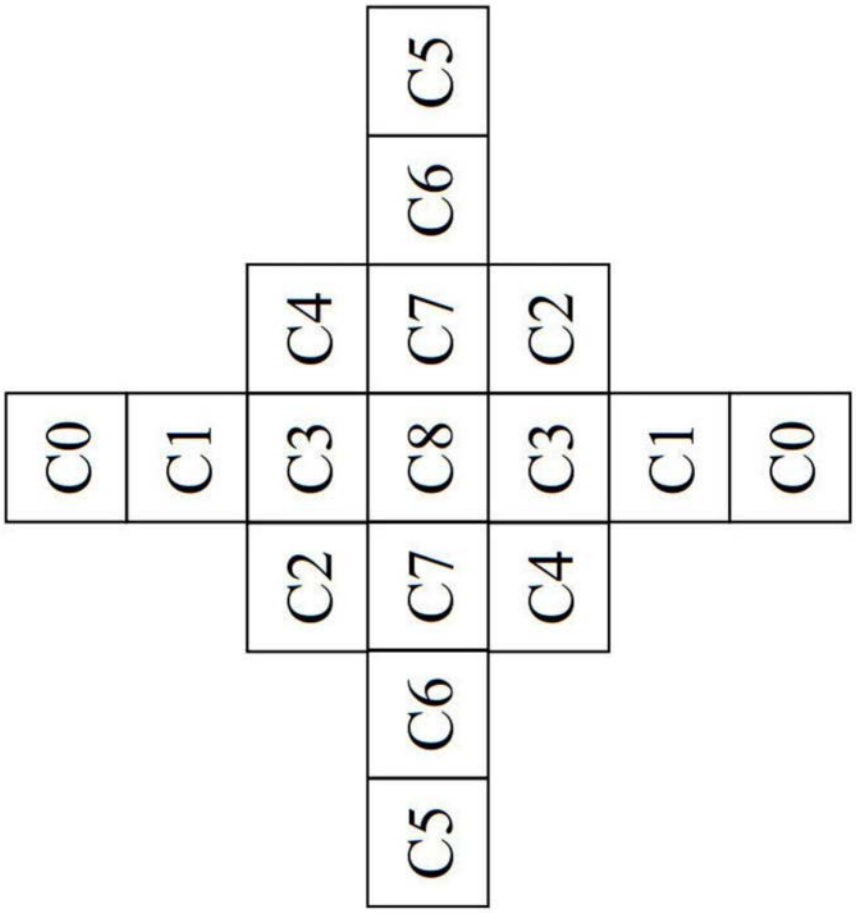

图5

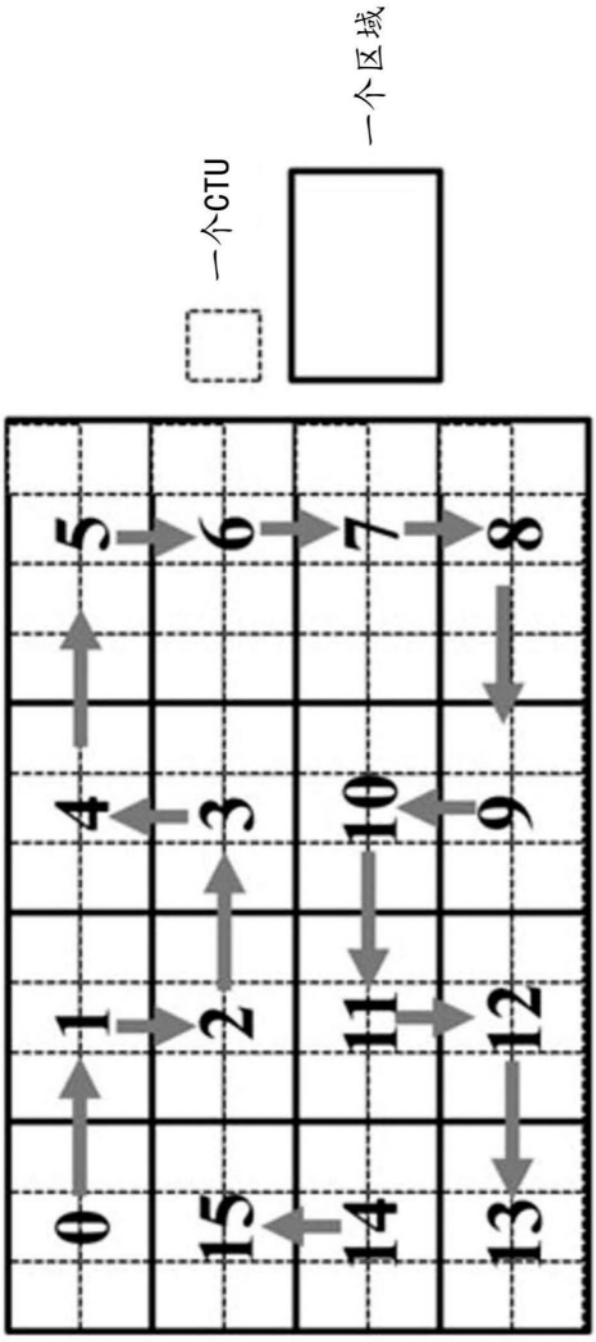

图6

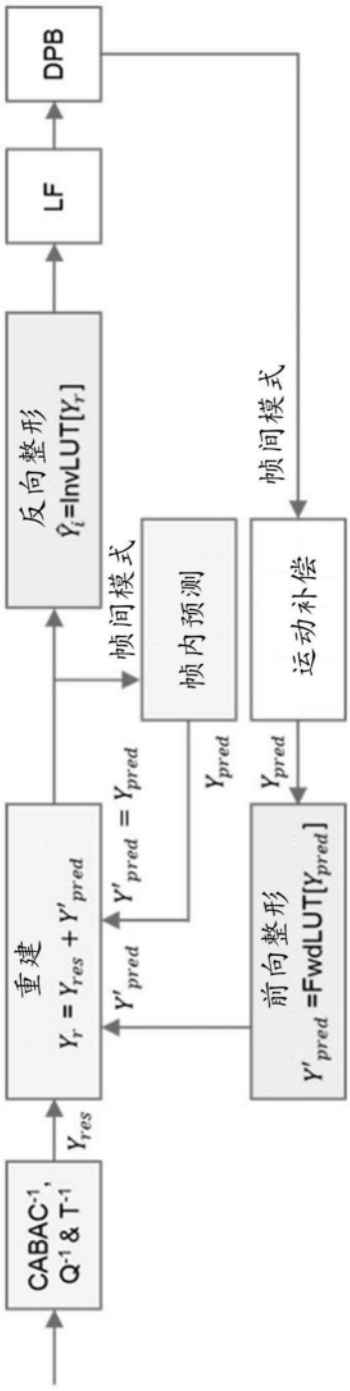
DPB
LF
反向整形
$\hat{Y}_i$=InvLUT[Y_r]
帧间模式
帧内预测
帧间模式
运动补偿
Y_{pred}
Y_{pred}
$Y'_{pred} = Y_{pred}$
重建
$Y_r = Y_{res} + Y'_{pred}$
Y'_{pred}
前向整形
Y'_{pred} =FwdLUT[Y_{pred}]
Y_{res}
CABAC^{-1}, Q^{-1} & T^{-1}

图7

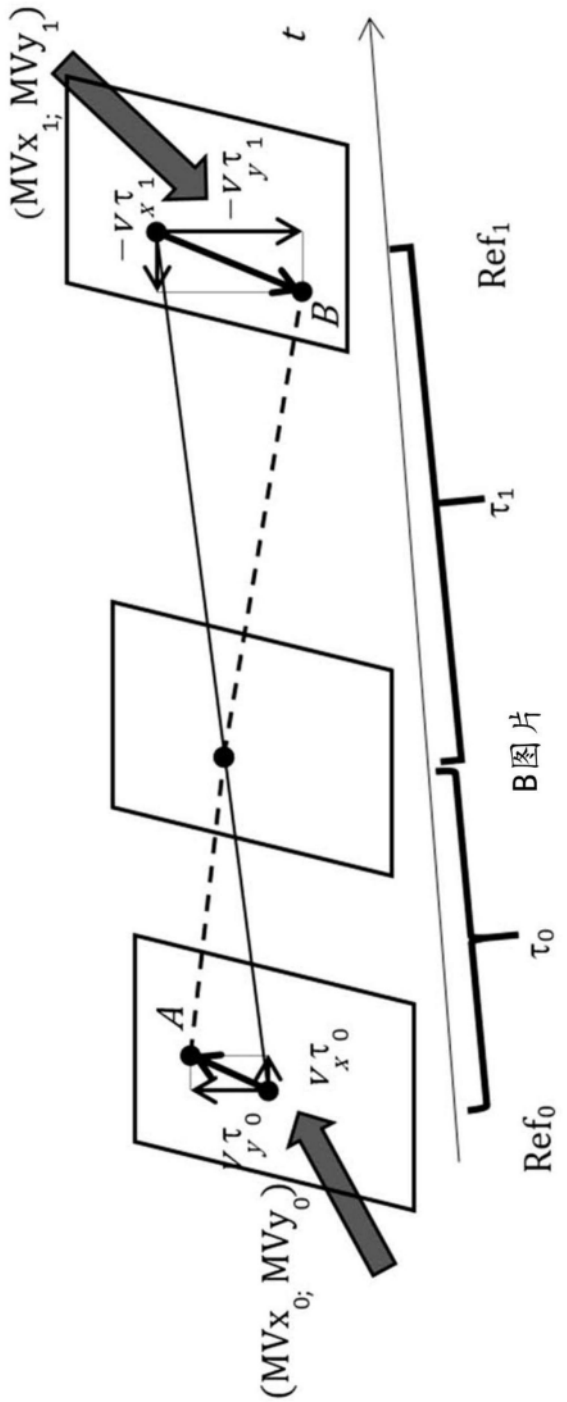

图8

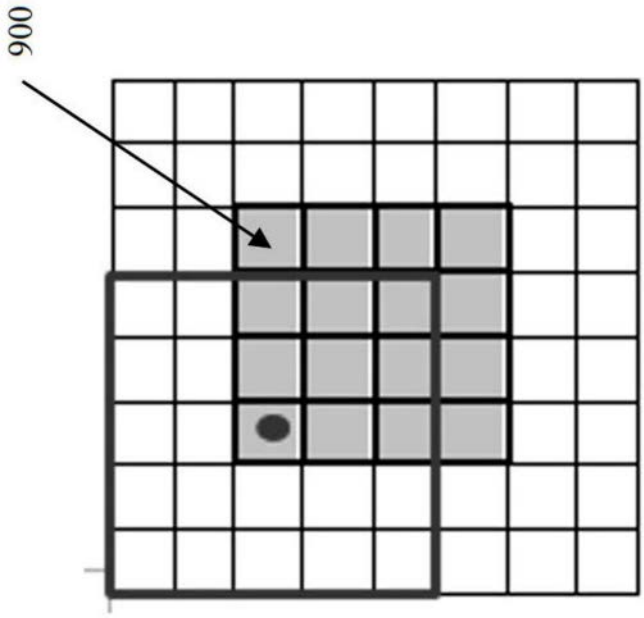

图9A

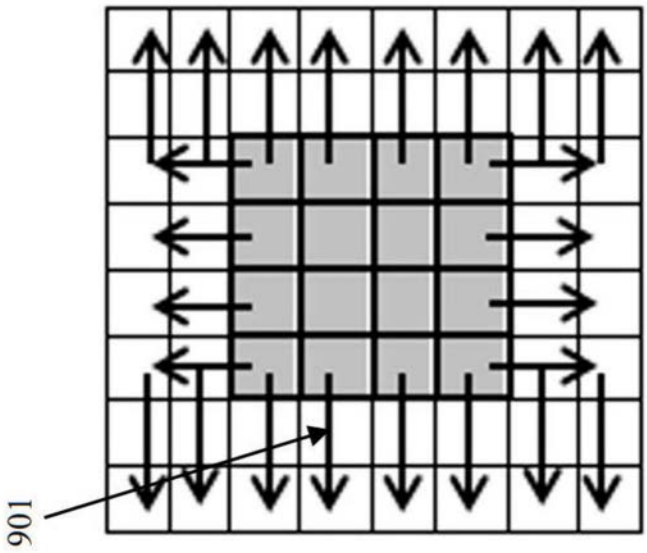

图9B

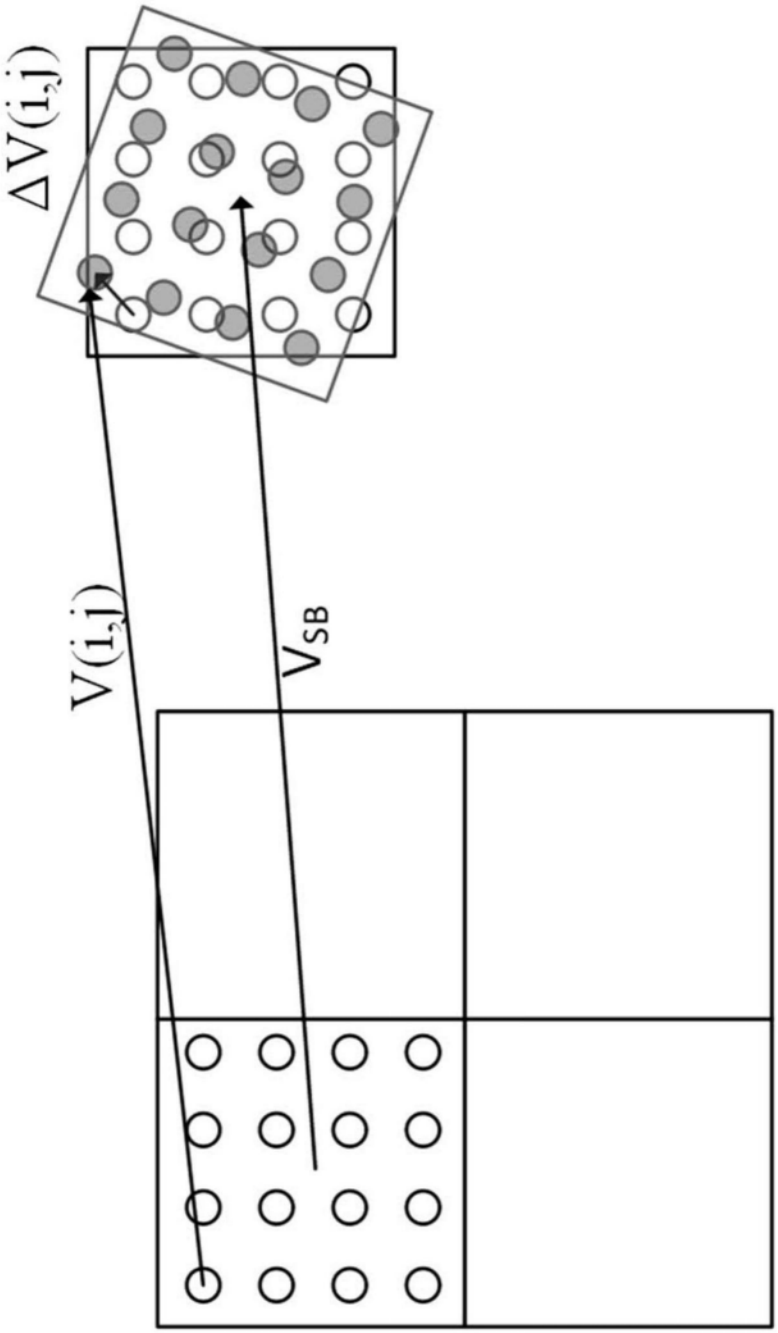

图10

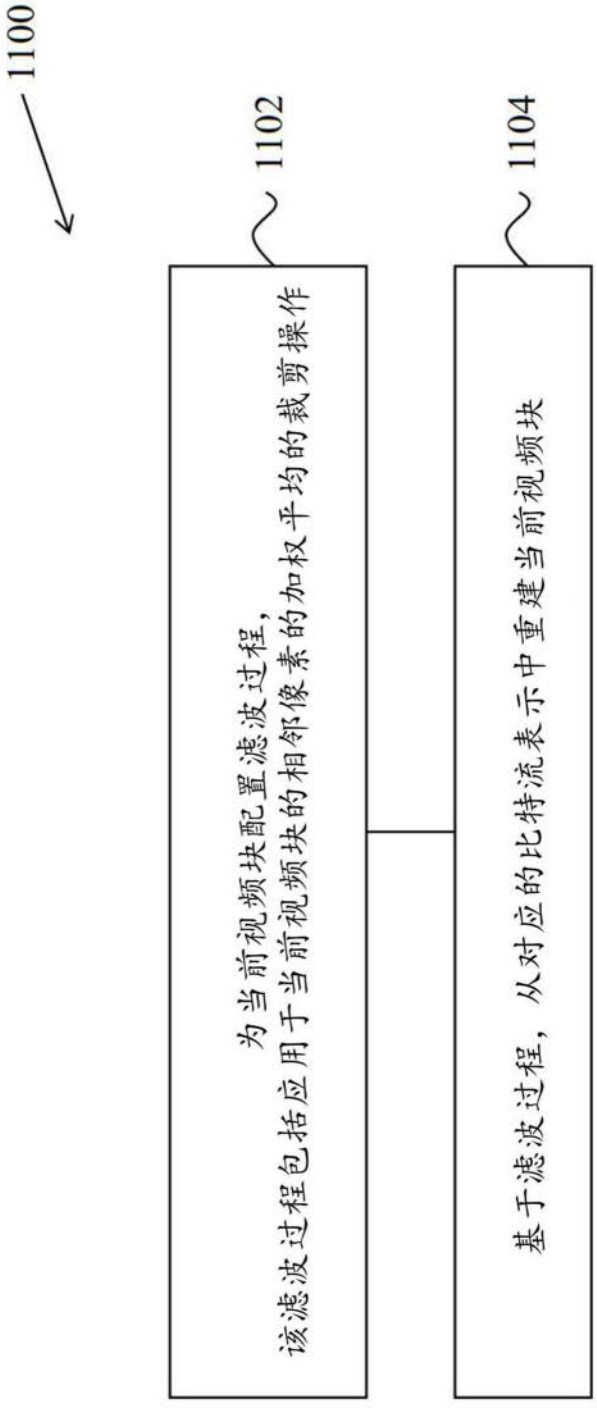
1100
1102
为当前视频块配置滤波过程，
该滤波过程包括应用于当前视频块的相邻像素的加权平均的裁剪操作
1104
基于滤波过程，从对应的比特流表示中重建当前视频块

图11A

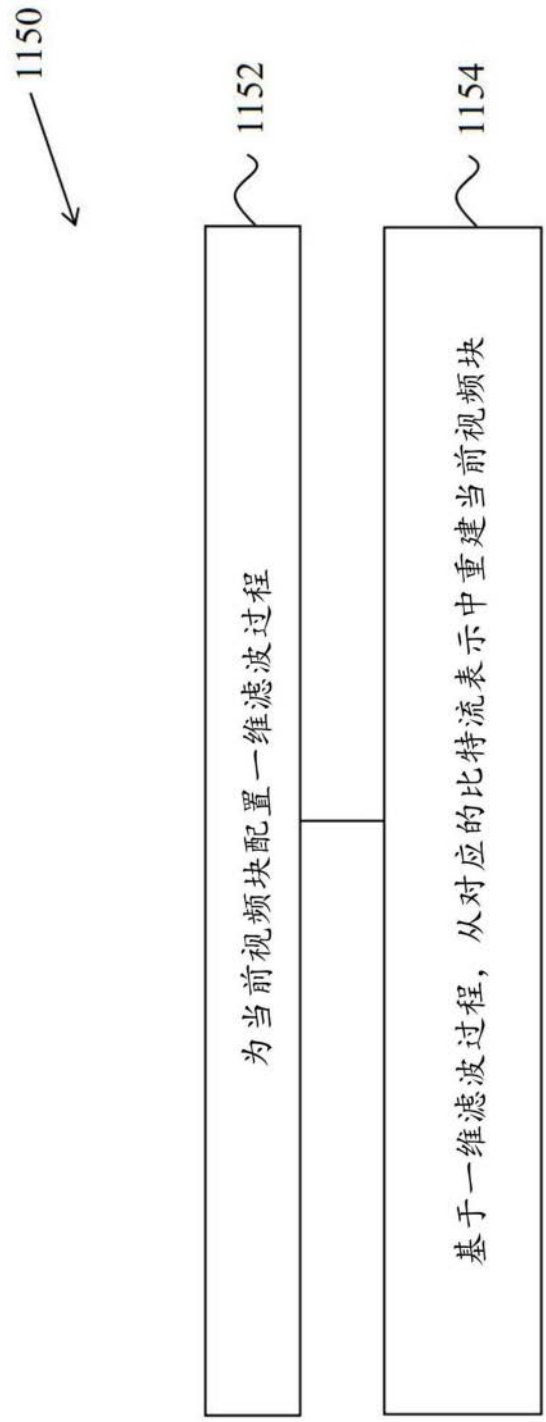
1150
1152
为当前视频块配置一维滤波过程
1154
基于一维滤波过程，从对应的比特流表示中重建当前视频块

图11B

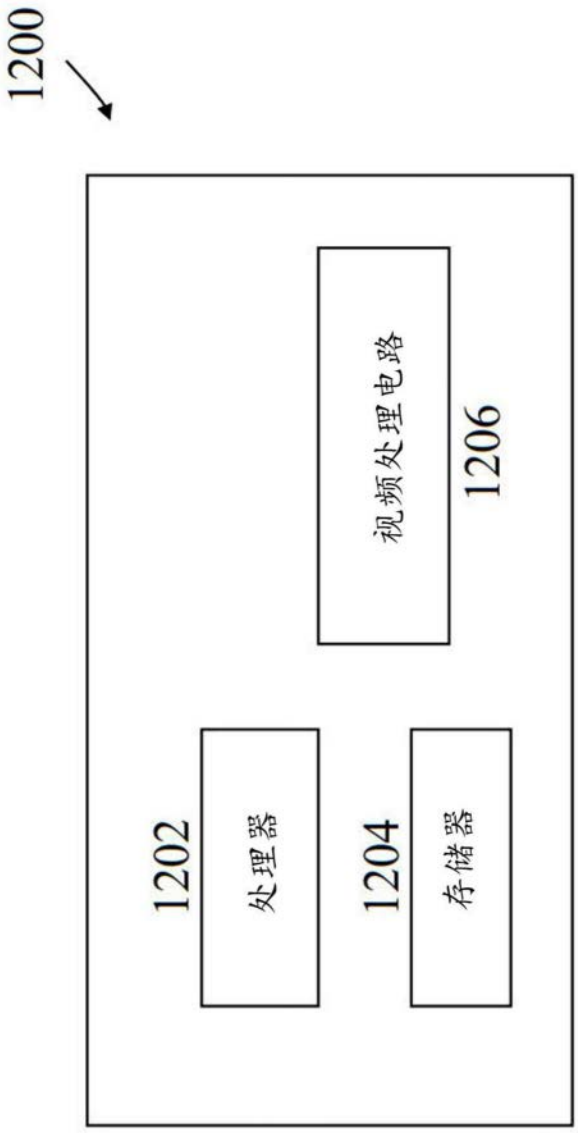

图12

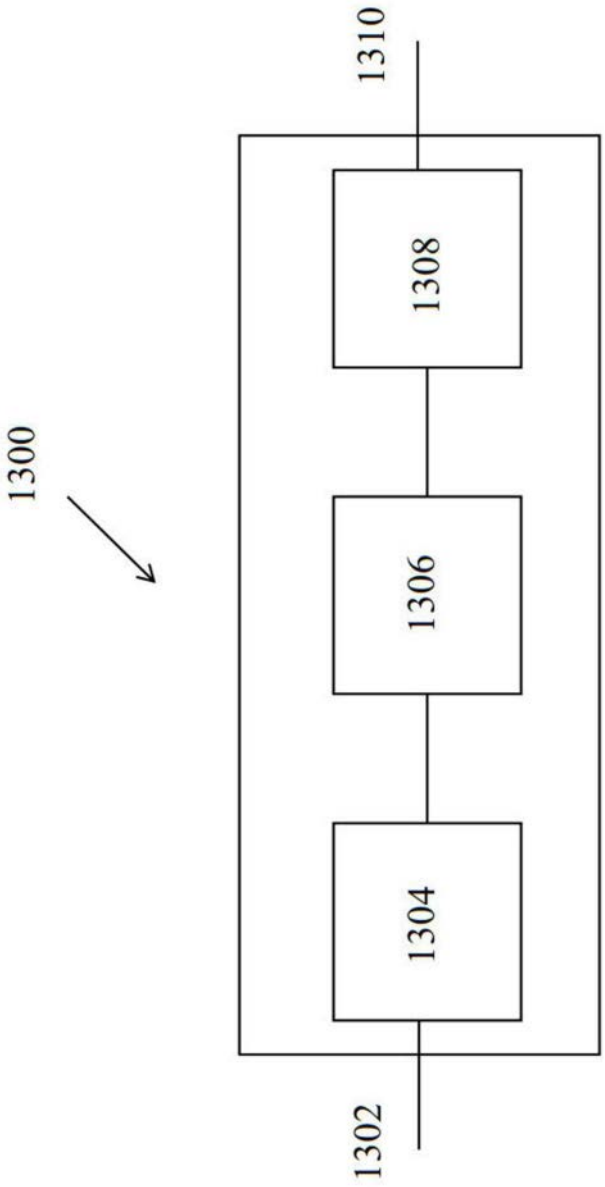

图13

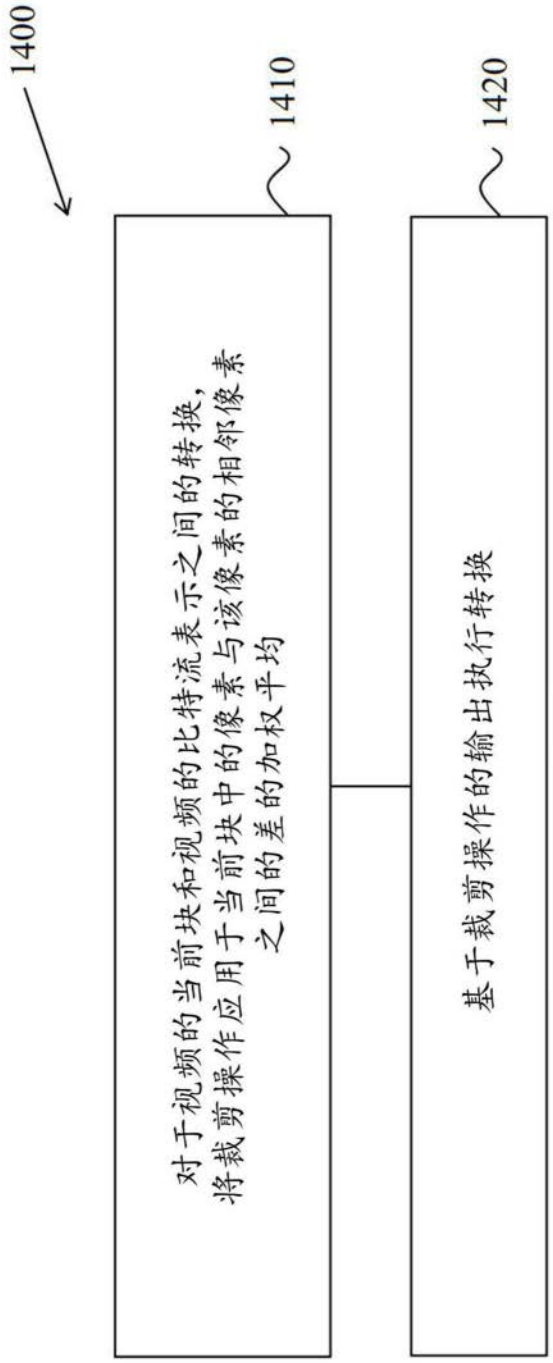
1400
1410
对于视频的当前块和视频的比特流表示之间的转换，将裁剪操作应用于当前块中的像素与该像素的相邻像素之间的差的加权平均
1420
基于裁剪操作的输出执行转换

图14

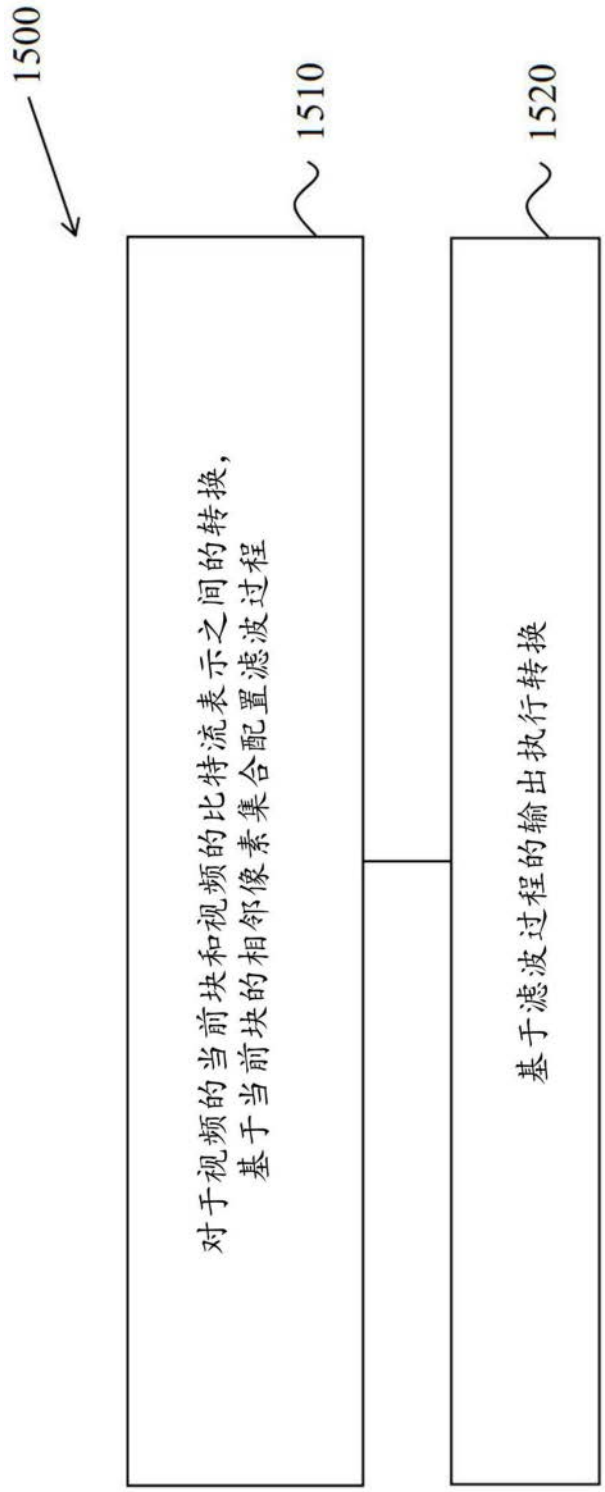
1500
1510
对于视频的当前块和视频的比特流表示之间的转换，基于当前块的相邻像素集合配置滤波过程
1520
基于滤波过程的输出执行转换

图15

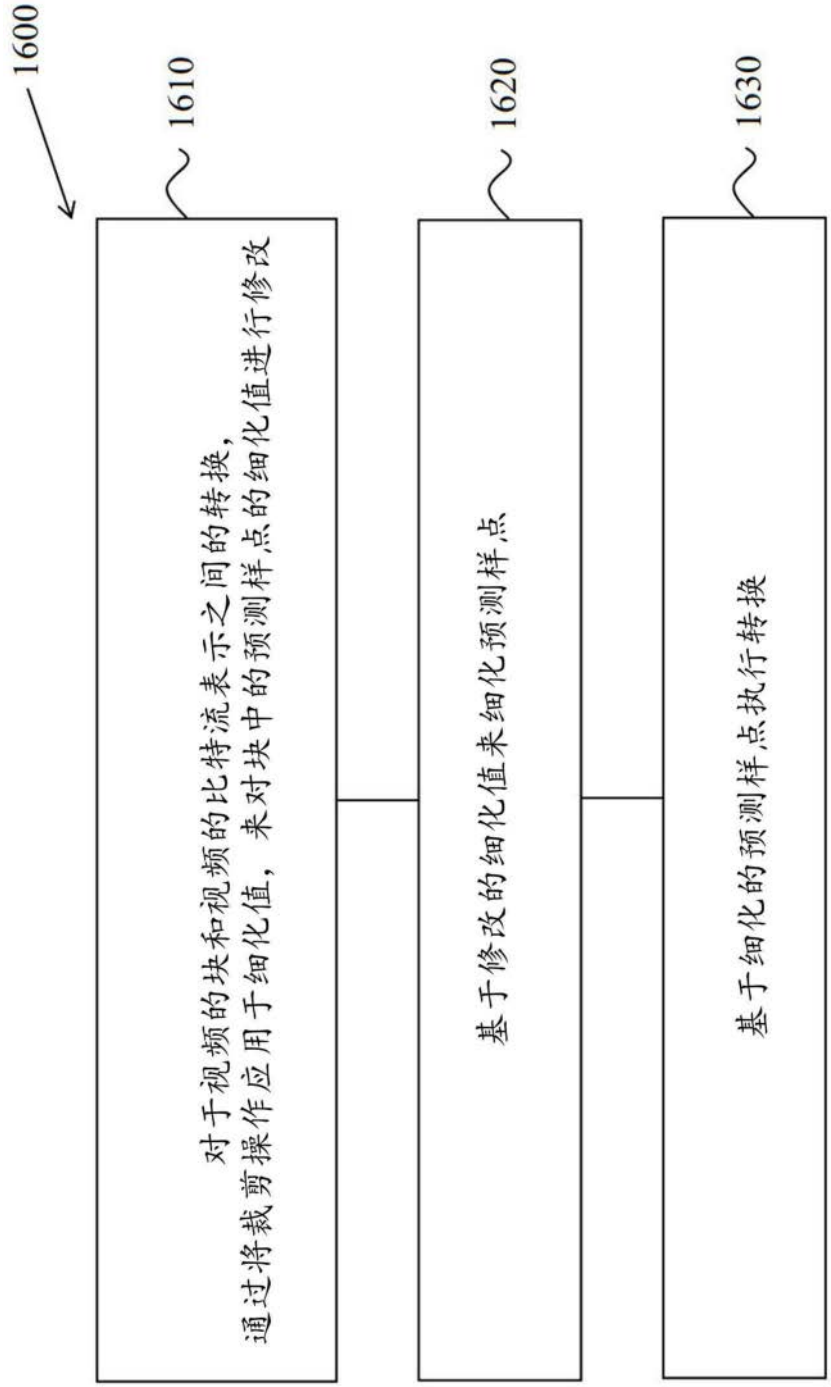
1600
1610
对于视频的块和视频的比特流表示之间的转换，通过将裁剪操作应用于细化值，来对块中的预测样点的细化值进行修改
1620
基于修改的细化值来细化预测样点
1630
基于细化的预测样点执行转换

图16

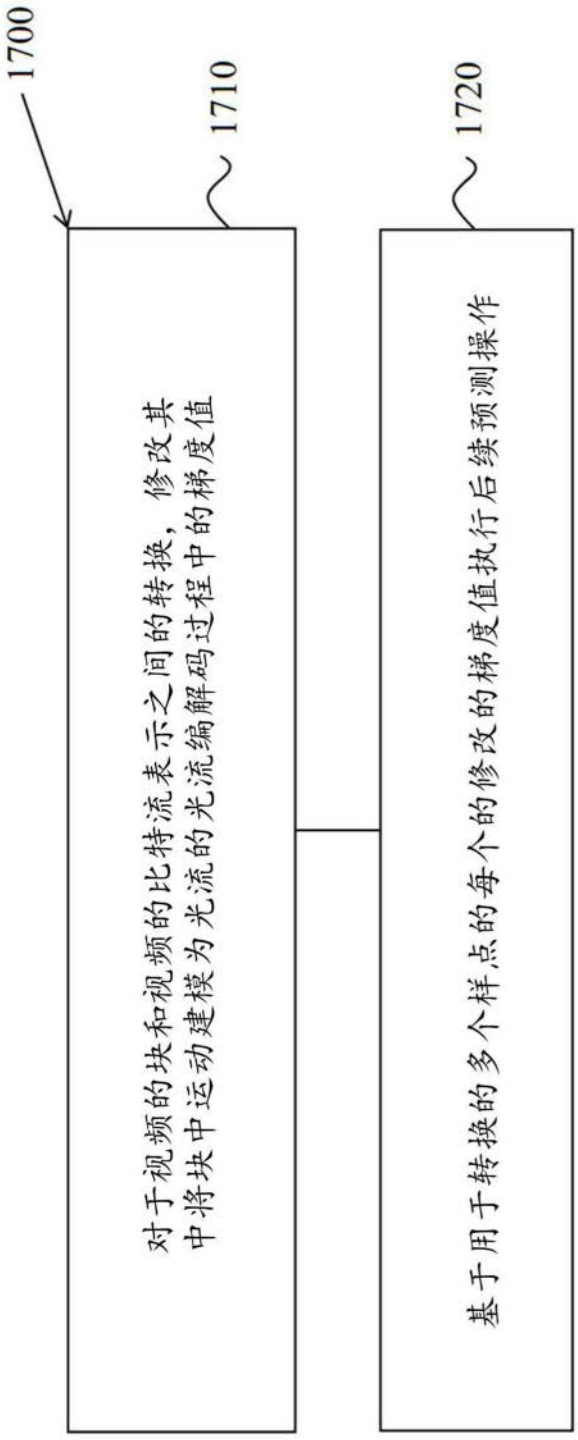
1700
1710
对于视频的块和视频的比特流表示之间的转换，修改其中将块中运动建模为光流的光流编解码过程中的梯度值
1720
基于用于转换的多个样点的每个的修改的梯度值执行后续预测操作

图17