



(12) 发明专利申请

(10) 申请公布号 CN 117793377 A

(43) 申请公布日 2024.03.29

(21) 申请号 202410005772.8

(22) 申请日 2018.06.29

(30) 优先权数据

62/526,577 2017.06.29 US

62/561,561 2017.09.21 US

62/629,313 2018.02.12 US

62/680,710 2018.06.05 US

62/686,738 2018.06.19 US

(62) 分案原申请数据

201880012069.2 2018.06.29

(71) 申请人 杜比实验室特许公司

地址 美国加利福尼亚州

(72) 发明人 吕陶然 浦方君 尹鹏 陈涛

W·J·胡萨克

(74) 专利代理机构 北京市汉坤律师事务所

11602

专利代理人 魏小微 吴丽丽

(51) Int.Cl.

H04N 19/61 (2014.01)

H04N 19/117 (2014.01)

H04N 19/34 (2014.01)

H04N 19/44 (2014.01)

H04N 19/107 (2014.01)

H04N 19/503 (2014.01)

H04N 19/159 (2014.01)

H04N 19/98 (2014.01)

H04N 19/182 (2014.01)

H04N 19/124 (2014.01)

H04N 19/136 (2014.01)

H04N 19/463 (2014.01)

H04N 19/176 (2014.01)

H04N 19/82 (2014.01)

H04N 19/174 (2014.01)

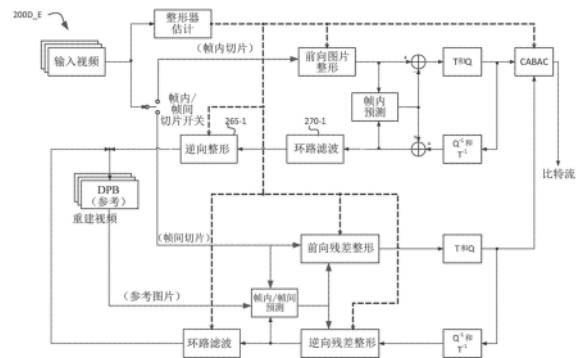
权利要求书2页 说明书41页 附图19页

(54) 发明名称

整合的图像整形和视频编码

(57) 摘要

本公开涉及整合的图像整形和视频编码。在给定了采用第一码字表示的图像序列的情况下，提出了用于将整形整合到用于对图像进行编码和解码的下一代视频编解码器中的方法、过程和系统，其中，整形允许采用第二码字表示对所述图像的一部分进行编码，所述第二码字表示允许比使用所述第一码字表示更高效的压缩。讨论了各种架构，包括：环外整形架构、仅用于帧内图片的环内整形架构、用于预测残差的环内架构以及混合环内整形架构。还提出了用于用信号传送整形参数的语法方法以及关于整形而优化的图像编码方法。



1. 一种用于对编码比特流进行解码以生成采用第一码字表示的输出图像的装置,所述装置包括:

输入部,所述输入部用于接收部分采用第二码字表示编码的编码图像;以及

处理器,其中,所述处理器:

接收所述编码图像的整形信息;

基于所述整形信息生成前向整形函数,所述前向整形函数将像素从所述第一码字表示映射到所述第二码字表示;

基于所述整形信息生成逆向整形函数,其中,所述逆向整形函数将像素从所述第二码字表示映射到所述第一码字表示;

针对所述编码图像的区域:

生成经解码经整形的残差区域;

基于参考像素缓冲器中的或先前经解码的空间邻域中的像素来生成预测区域;

基于所述经解码经整形的残差区域、所述预测区域、所述前向整形函数和所述逆向整形函数来生成重建的像素区域,其中,生成所述重建的像素区域包括计算:

$Reco\_sample(i) = Inv(Res\_d(i) + Fwd(Pred\_sample(i)))$ ,其中, $Reco\_sample(i)$ 表示所述重建的像素区域的像素, $Res\_d(i)$ 表示所述经解码经整形的残差区域的像素, $Inv()$ 表示所述逆向整形函数, $Fwd()$ 表示所述前向整形函数,并且 $Pred\_sample(i)$ 表示所述预测区域的像素;

基于所述重建的像素区域生成所述输出图像的输出像素区域;以及

将所述输出像素区域存储在所述参考像素缓冲器中。

2. 如权利要求1所述的装置,其中,所述编码图像的区域包括图像感兴趣区域。

3. 一种用于利用处理器对编码比特流进行解码以生成采用第一码字表示的输出图像的装置,所述装置包括:

输入部,所述输入部用于接收部分采用第二码字表示编码的编码图像;以及

处理器,其中,所述处理器:

接收所述编码图像的整形信息;

基于所述整形信息生成整形缩放函数;

针对所述编码图像的区域:

生成经解码经整形的残差区域;

基于参考像素缓冲器中的或先前经解码的空间邻域中的像素来生成预测区域;

基于所述经解码经整形的残差区域、所述预测区域和所述整形缩放函数来生成重建的像素区域;

基于所述重建的像素区域生成所述输出图像的输出像素区域;以及

将所述输出像素区域存储在所述参考像素缓冲器中。

4. 如权利要求3所述的装置,其中,生成所述重建的像素区域包括计算:

$Reco\_samde(i) = Pred\_sample(i) + (1/a(Pred\_sample(i))) * Res\_d(i)$ ,

其中, $Reco\_sample(i)$ 表示所述重建的像素区域的像素, $Res\_d(i)$ 表示所述经解码经整形的残差区域的像素, $a()$ 表示所述整形缩放函数,并且 $Pred\_sample(i)$ 表示所述预测区域的像素。

5. 如权利要求3所述的装置,其中,所述编码图像的区域包括图像感兴趣区域。

## 整合的图像整形和视频编码

[0001] 本申请是申请日为2018年6月29日、申请号为201880012069.2、发明名称为“整合的图像整形和视频编码”的发明专利申请的分案申请。

[0002] 相关申请的交叉引用

[0003] 本申请要求于2018年6月19日提交的美国临时专利申请序列号62/686,738、于2018年6月5日提交的序列号62/680,710、于2018年2月12日提交的序列号62/629,313、于2017年9月21日提交的序列号62/561,561、以及于2017年6月29日提交的序列号62/526,577的优先权,这些美国专利申请中的每一个通过引用以其全文结合于本文中。

### 技术领域

[0004] 本发明总体上涉及图像和视频编码。更具体地,本发明的实施例涉及整合的图像整形和视频编码。

### 背景技术

[0005] 2013年,国际标准化组织(ISO)中的MPEG专家组连同国际电信联盟(ITU)发布了HEVC(也称为H.265)视频编码标准的初稿。最近,MPEG专家组发布了一项证据用于支持下一代编码标准的开发,所述下一代编码标准提供了与现有视频编码技术相比改进的编码性能。

[0006] 如本文所使用的,术语‘比特深度(bit depth)’表示用于表示图像的颜色分量之一的像素数量。传统上,图像以每个像素每颜色分量8比特(例如,每像素24比特)来编码;然而,现代架构现在可以支持更高的比特深度,诸如10比特、12比特或更多。

[0007] 在传统图像流水线(pipeline)中,使用非线性光电函数(OETF)来量化所捕获的图像,所述非线性光电函数将线性场景光转换为非线性视频信号(例如,伽马编码的RGB或YCbCr)。然后,信号在被显示在显示器上之前在接收器上通过电光转换函数(EOTF)来处理,所述电光转换函数将视频信号值转化成输出屏幕颜色值。此类非线性函数包括在ITU-R Rec.BT.709和BT.2020中记载的传统“伽马”曲线以及SMPTE ST 2084和Rec.ITU-R BT.2100中描述的“PQ”(感知量化)曲线。

[0008] 如本文所使用的,术语“前向整形(forward reshaping)”表示数字图像从其原始比特深度和原始码字分布或表示(例如,伽马或PQ等)到相同或不同比特深度和不同码字分布或表示的图像的样本到样本映射或码字到码字映射的过程。整形允许以固定的比特率改善可压缩性或改善图像质量。例如,非限制性地,可以将整形应用于10比特或12比特PQ编码的HDR视频,以提高10比特视频编码架构中的编码效率。在接收器中,在解压缩经整形的信号之后,接收器可以应用“逆向整形函数”来将信号恢复到其原始码字分布。如在此由本发明人理解的,随着针对下一代视频编码标准的开发开始,期望用于图像的整合的整形和编码的改进技术。本发明的方法可以适用于各种视频内容,包括但不限于标准动态范围(SDR)和/或高动态范围(HDR)中的内容。

[0009] 在这一部分中描述的方法是可以寻求的方法,但不一定是之前已经设想到或寻求

的方法。因此,除非另有指明,否则不应认为本节中所述的任何方法仅凭其纳入本节就可称为现有技术。类似地,除非另有指明,否则关于一种或多种方法所认定的问题不应基于本节而认为在任何现有技术中被认定。

## 附图说明

[0010] 在附图中以举例而非限制的方式来展示本发明的实施例,并且其中相似的附图标记指代相似的元件,并且在附图中:

[0011] 图1A描绘了视频传输流水线的示例过程;

[0012] 图1B描绘了根据现有技术的使用信号整形进行数据压缩的示例过程;

[0013] 图2A描绘了根据本发明的实施例的使用规范的环外整形的编码器的示例架构;

[0014] 图2B描绘了根据本发明的实施例的使用规范的环外整形的解码器的示例架构;

[0015] 图2C描绘了根据本发明的实施例的使用规范的仅帧内环内整形的编码器的示例架构;

[0016] 图2D描绘了根据本发明的实施例的使用规范的仅帧内环内整形的解码器的示例架构;

[0017] 图2E描绘了根据本发明的实施例的使用用于预测残差的环内整形的编码器的示例架构;

[0018] 图2F描绘了根据本发明的实施例的使用用于预测残差的环内整形的解码器的示例架构;

[0019] 图2G描绘了根据本发明的实施例的使用混合环内整形的编码器的示例架构;

[0020] 图2H描绘了根据本发明的实施例的使用混合环内整形的解码器的示例架构;

[0021] 图3A描绘了根据本发明的实施例的用于使用环外整形架构对视频进行编码的示例过程;

[0022] 图3B描绘了根据本发明的实施例的用于使用环外整形架构对视频进行解码的示例过程;

[0023] 图3C描绘了根据本发明的实施例的用于使用环内仅帧内整形架构对视频进行编码的示例过程;

[0024] 图3D描绘了根据本发明的实施例的用于使用环内仅帧内整形架构对视频进行解码的示例过程;

[0025] 图3E描绘了根据本发明的实施例的用于使用用于预测残差的环内整形架构对视频进行编码的示例过程;

[0026] 图3F描绘了根据本发明的实施例的用于使用用于预测残差的环内整形架构对视频进行解码的示例过程;

[0027] 图4A描绘了根据本发明的实施例的用于使用三种基于整形的架构中的任一种或其组合对视频进行编码的示例过程;

[0028] 图4B描绘了根据本发明的实施例的用于使用三种基于整形的架构中的任一种或其组合对视频进行解码的示例过程;

[0029] 图5A和图5B描绘了根据本发明的实施例的在视频解码器中的整形函数重建过程;

[0030] 图6A和图6B描绘了根据本发明的实施例的色度QP偏移值如何根据PQ编码信号和

HLG编码信号的亮度量化参数(QP)而改变的示例;并且

[0031] 图7描绘了根据本发明的实施例的整形函数的基于枢轴(pivot)表示的示例。

### 具体实施方式

[0032] 本文描述了用于压缩图像的规范的环外和环内整合的信号整形和编码技术。在以下说明中,出于解释的目的,阐述了许多具体细节以便提供对本发明的透彻理解。然而,明显的是,可以在没有这些具体细节的情况下实施本发明。在其他情形中,为了避免不必要的遮蔽、模糊或混淆本发明,没有详尽地描述众所周知的结构和设备。

[0033] 概述

[0034] 本文描述的示例实施例涉及对视频进行整合的信号整形和编码。在编码器中,处理器接收采用第一码字表示的输入图像,所述第一码字表示由输入比特深度N和输入码字映射(例如,伽马(gamma)、PQ等)表示。处理器从用于使用允许比第一码字表示更高效的压缩的第二码字表示来压缩输入图像的两个或更多个候选编码器架构中选择编码器架构(其中,整形器是编码器的组成部分),其中,所述两个或更多个候选编码器架构包括环外整形架构、仅用于帧内图片的环内整形架构、或用于预测残差的环内架构,并且处理器根据所选编码器架构来压缩输入图像。

[0035] 在另一实施例中,用于生成采用第一码字表示的输出图像的解码器接收编码比特流,其中,编码图像中的至少一部分以第二码字表示被压缩。所述解码器还接收相关联的整形信息。处理器接收指示来自用于解压缩输入编码比特流的两个或更多个候选解码器架构的解码器架构的信令,其中,所述两个或更多个候选解码器架构包括环外整形架构、仅用于帧内图片的环内整形架构、或用于预测残差的环内架构,并且处理器根据所接收的整形架构来对编码图像进行解压缩以生成输出图像。

[0036] 在另一实施例中,在用于根据用于预测残差的环内架构来压缩图像的编码器中,处理器访问采用第一码字表示的输入图像,并生成将输入图像的像素从第一码字表示映射到第二码字表示的前向整形函数。所述处理器基于前向整形函数来生成将由第二码字表示的像素映射到采用第一码字表示的像素的逆向整形函数。然后,针对输入图像中的输入像素区域:所述处理器进行以下操作:

[0037] 基于参考帧缓冲器中的或先前编码的空间邻域中的像素数据来计算至少一个预测区域;

[0038] 基于输入像素区域、预测区域和前向整形函数来生成经整形的残差区域;

[0039] 基于经整形的残差区域来生成经编码的(经变换和经量化的)残差区域;

[0040] 基于经编码的残差区域来生成经解码的(经逆量化的和经逆变换的)残差区域;

[0041] 基于经解码的残差区域、预测区域、前向整形函数和逆向整形函数来生成重建的像素区域;以及

[0042] 基于重建的像素区域来生成要存储在参考帧缓冲器上的参考像素区域。

[0043] 在另一实施例中,在用于根据用于预测残差的环内架构来生成采用第一码字表示的输出图像的解码器中,处理器接收部分采用第二码字表示编码的编码比特流。所述处理器还接收相关联的整形信息。处理器基于整形信息来生成前向整形函数和逆向整形函数,其中,所述前向整形函数将像素从第一码字表示映射到第二码字表示,并且所述逆向整形

函数将像素从第二码字表示映射到第一码字表示。针对编码图像的区域,处理器进行以下操作:

[0044] 基于编码图像来生成经解码经整形的残差区域;

[0045] 基于参考像素缓冲器中的或先前解码的空间邻域中的像素来生成预测区域;

[0046] 基于经解码经整形的残差区域、预测区域、前向整形函数和逆向整形函数来生成重建的像素区域;

[0047] 基于重建的像素区域来生成输出像素区域;以及,将输出像素区域存储在参考像素缓冲器中。

[0048] 示例视频传输处理流水线

[0049] 图1A描绘了传统视频传输流水线(100)的示例过程,其示出了从视频捕获到视频内容显示的各个阶段。使用图像生成框(105)来捕获或生成视频帧(102)序列。视频帧(102)可以被(例如,由数码相机)数字地捕获或者由计算机(例如,使用计算机动画)生成以提供视频数据(107)。可替代地,视频帧(102)可以由胶片摄像机捕获在胶片上。胶片被转换为数字格式以提供视频数据(107)。在制作阶段(110),对视频数据(107)进行编辑以提供视频制作流(112)。

[0050] 制作流的视频数据(112)然后在框(115)处被提供给处理器以进行后期制作编辑。框(115)后期制作编辑可以包括调整或修改图像的特定区中的颜色或明亮度(brightness),以根据视频创作者的创作意图来增强图像质量或实现图像的特定外观。这有时被称为“颜色调整(color timing)”或“颜色分级(color grading)”。可以在框(115)处执行其他编辑(例如,场景选择和排序、图像裁剪、添加计算机生成的视觉特效等)以产生用于分发的产品的最终版本(117)。在后期制作编辑(115)期间,在参考显示器(125)上观看视频图像。

[0051] 在后期制作(115)之后,可以将最终产品(117)的视频数据传输到编码框(120),以便向下游传输到诸如电视机、机顶盒、电影院等解码和重放设备。在一些实施例中,编码框(120)可以包括诸如由ATSC、DVB、DVD、蓝光和其他传输格式定义的那些音频编码器和视频编码器,以生成编码比特流(122)。在接收器中,编码比特流(122)由解码单元(130)解码,以生成表示信号(117)的完全相同或接近近似(close approximation)的经解码信号(132)。接收器可以附接到目标显示器(140),所述目标显示器可以具有与参考显示器(125)完全不同的特性。在这种情况下,显示管理框(135)可以用于通过生成显示器映射信号(137)来将经解码信号(132)的动态范围映射到目标显示器(140)的特性。

[0052] 信号整形

[0053] 图1B描绘了根据现有技术(参考文献[1])的用于信号整形的示例过程。在给定了输入帧(117)的情况下,前向整形框(150)分析输入约束和编码约束,并生成将输入帧(117)映射到重新量化的输出帧(152)的码字映射函数。例如,可以根据某种电光转换函数(EOTF)(例如,伽马)来对输入(117)进行编码。在一些实施例中,可以使用元数据将关于整形过程的信息传送至下游设备(诸如解码器)。如本文所使用的,术语“元数据”涉及作为编码比特流的一部分传输并且辅助解码器渲染经解码图像的任何辅助信息。这样的元数据可以包括但不限于如本文所描述的那些颜色空间或色域信息、参考显示器参数和辅助信号参数。

[0054] 在编码(120)和解码(130)之后,经解码帧(132)可以通过将重新量化的帧(132)转

换回原始EOTF域(例如,伽马)的后向(或逆向)整形函数(160)来处理以供进一步的下游处理,诸如前面讨论的显示管理过程(135)。在一些实施例中,可以将后向整形函数(160)与解码器(130)中的去量化器整合,例如作为AVC或HEVC视频解码器中的去量化器的一部分。

[0055] 如本文所使用的,术语“整形器”可以表示在对数字图像进行编码和/或解码时要使用的前向整形函数或逆向整形函数。参考文献[1]和[2]中讨论了整形函数的示例。出于本发明的目的,假设本领域技术人员可以根据输入视频信号的特性以及编码架构和解码架构的可用比特深度来得到合适的前向整形函数和逆向整形函数。

[0056] 在参考文献[1]中,提出了一种用于高动态范围视频编码的基于块的环内图像整形方法。这种设计允许在编码环路内进行基于块的整形,但代价是增加了复杂度。具体而言,所述设计需要维持两组解码图像缓冲器:一组用于经逆向整形(或非整形)的解码图片,所述图片可以用于在不进行整形的情况下进行预测以及用于输出到显示器;并且另一组用于经前向整形的解码图片,所述图片仅用于在整形的情况下进行预测。尽管可以实时地计算经前向整形的解码图片,但是复杂度成本非常高,尤其是对于帧间预测(利用子像素插值的运动补偿)。通常,显示图片缓冲器(DPB)管理是复杂的并且需要非常小心,因此,如本领域人所理解的,期望用于对视频进行编码的简化方法。

[0057] 本文提出的基于整形的编解码器架构的实施例可以划分如下:具有外部环外整形器的架构、具有环内仅帧内整形器的架构以及具有用于预测残差的环内整形器(也简称为“环内残差整形器”)的架构。视频编码器或解码器可以支持这些架构中的任何一个或其组合。这些架构中的每一个还可以单独应用或与其他架构中的任何一个组合应用。每个架构可以应用于明度(luminance)分量、色度分量、或者亮度(luma)与一个或多个色度分量的组合。

[0058] 除了这三种架构之外,附加实施例描述了用于与整形相关的元数据的高效信令方法以及用于在应用整形时提高编码效率的若干基于编码器的优化工具。

#### [0059] 规范的环外整形器

[0060] 图2A和图2B描绘了用于具有“规范的(normative)”环外整形器的视频编码器(200A\_E)和相应视频解码器(200A\_D)的架构。术语“规范的”表示不同于其中整形被认为是预处理步骤并且因此在诸如AVC、HEVC等编码标准的规范描述之外的先前设计,在这个实施例中,前向整形和逆向整形是规范的要求的一部分。与图1B的在解码(130)之后根据标准来测试比特流一致性的架构不同,在图2B中,在逆向整形框(265)(例如,在图1B中的输出162处)之后测试一致性。

[0061] 在编码器(200A\_E)中,两个新框被添加到传统的基于块的编码器(例如,HEVC):用于估计前向整形函数的框(205),以及对输入视频(117)的一个或多个颜色分量应用前向整形的前向图片整形框(210)。在一些实施例中,这两个操作可以作为单个图像整形框的一部分来执行。与在解码器中确定逆向整形函数相关的参数(207)可以被传递到视频编码器的无损编码器块(例如,CABAC 220),使得所述参数可以嵌入到编码比特流(122)中。使用存储在DPB(215)中的经整形图片来执行与帧内或帧间预测(225)、变换和量化(T和Q)、逆变换和逆量化( $Q^{-1}$ 和 $T^{-1}$ )以及环路滤波相关的所有操作。

[0062] 在解码器(200A\_D)中,两个新的规范的框被添加到传统的基于块的解码器:用于基于经编码的整形函数参数(207)来重建逆向整形函数的框(250),以及用于将逆向整形函

数应用于经解码数据(262)以生成经解码的视频信号(162)的框(265)。在一些实施例中,与框250和265相关的操作可以组合到单个处理框中。

[0063] 图3A描绘了根据本发明的实施例的用于使用环外整形架构(200A\_E)对视频进行编码的示例过程(300A\_E)。如果没有启用整形(路径305),则编码如现有技术编码器(例如,HEVC)中已知的那样进行。如果启用了整形(路径310),则编码器可以具有以下选项:要么应用预定(默认)整形函数(315),要么基于图片分析(320)(例如,如参考文献[1]至[3]中所描述的)来自适应地确定新的整形函数(325)。在前向整形(330)之后,编码的其余部分遵循传统的编码流水线(335)。如果采用自适应整形(312),则生成与逆向整形函数相关的元数据作为“编码整形器”步骤(327)的一部分。

[0064] 图3B描绘了根据本发明的实施例的用于使用环外整形架构(200A\_D)对视频进行解码的示例过程(300A\_D)。如果没有启用整形(路径355),则在对图片进行解码(350)之后,如在传统解码流水线中那样生成(390)输出帧。如果启用了整形(路径360),则在步骤(370)中,解码器确定是应用预定(默认)整形函数(375),还是基于所接收的参数(例如,207)来自适应地确定逆向整形函数(380)。在逆向整形(385)之后,解码的其余部分遵循传统的解码流水线。

#### [0065] 规范的环内仅帧内整形器

[0066] 图2C描绘了根据本发明的实施例的用于使用规范的仅帧内环内整形的编码器(200B\_E)的示例架构。所述设计与参考文献[1]中提出的设计非常相似;然而,为了降低复杂度,尤其是在涉及DPB存储器(215和260)的使用时,仅使用该架构对帧内图片进行编码。

[0067] 编码器200B\_E与环外整形(200A\_E)相比的主要差异在于:DPB(215)存储经逆向整形的图片而不是经整形的图片。换言之,需要在将经解码的帧内图片存储到DPB之前对其进行逆向整形(通过逆向整形单元265)。这种方法背后的原因在于,如果利用整形对帧内图片进行编码,则即使帧间图片不利用整形来编码,对帧内图片进行编码的改进性能也将传播以(隐式地)改进对帧间图片的编码。以这种方式,人们可以利用整形而无需处理对帧间图片的环内整形的复杂度。由于逆向整形(265)是内部环路的一部分,因此可以在环内滤波器(270)之前实施逆向整形。在环内滤波器之前添加逆向整形的优点在于:在这种情况下,可以基于原始图片的特性而不是基于经前向整形的图片的特性来优化环内滤波器的设计。

[0068] 图2D描绘了根据本发明的实施例的用于使用规范的仅帧内环内整形的解码器(200B\_D)的示例架构。如图2D中描绘的,现在在环内滤波(270)之前执行确定逆向整形函数(250)和应用逆向整形(265)。

[0069] 图3C描绘了根据本发明的实施例的用于使用环内仅帧内整形架构对视频进行编码的示例过程(300B\_E)。如所描绘的,图3C中的操作流程与图3A中的操作流程共享许多要素。现在,默认情况下,不会对帧间编码应用整形。对于帧内编码图片,如果启用了整形,则编码器再次具有使用默认整形曲线或者应用自适应整形(312)的选项。如果图片被整形,则逆向整形(385)是所述过程的一部分,并且在步骤(327)中对相关联参数进行编码。在图3D中描绘了相应的解码过程(300B\_D)。

[0070] 如图3D中描绘的,仅对所接收的帧内图片、并且仅当在编码器上应用帧内整形时才启用整形相关操作。

#### [0071] 用于预测残差的环内整形器

[0072] 在编码中,术语‘残差’表示对样本或数据元素的预测与其原始或经解码值之间的差。例如,在给定了来自输入视频(117)的原始样本(表示为Orig\_sample)的情况下,帧内或帧间预测(225)可以生成表示为Pred\_sample的相应预测样本(227)。如果没有整形,则未整形的残差(Res\_u)可以定义为:

$$[0073] \quad \text{Res}_u = \text{Orig\_sample} - \text{Pred\_sample}. \quad (1)$$

[0074] 在一些实施例中,将整形应用于残差域可能是有益的。图2E描绘了根据本发明的实施例的用于使用用于预测残差的环内整形的编码器(200C\_E)的示例架构。令Fwd()表示前向整形函数,并且令Inv()表示相应的逆向整形函数。在实施例中,经整形的残差(232)可以被定义为:

$$[0075] \quad \text{Res}_r = \text{Fwd}(\text{Orig\_sample}) - \text{Fwd}(\text{Pred\_sample}). \quad (2)$$

[0076] 相应地,在逆向整形器(265)的输出(267)处,表示为Reco\_sample的重建样本(267)可以表达为:

$$[0077] \quad \text{Reca\_sample} = \text{Inv}(\text{Res}_d + \text{Fwd}(\text{Pred\_sample})), \quad (3)$$

[0078] 其中,Res\_d表示在200C\_E中的环内编码和解码之后的残差(234),即Res\_r的接近近似。

[0079] 注意的是,尽管将整形应用于残差,但实际输入视频像素不被整形。图2F描绘了相应的解码器(200C\_D)。注意的是,如图2F中描绘的、并且基于等式(3),解码器需要访问前向整形函数和逆向整形函数两者,这些函数可以使用所接收的元数据(207)和“整形器解码”框(250)来提取。

[0080] 在实施例中,为了降低复杂度,可以简化等式(2)和(3)。例如,假设前向整形函数可以通过分段线性函数来近似并且Pred\_sample与Orig\_sample之间的绝对差相对较小,则等式(2)可以近似为:

$$[0081] \quad \text{Res}_r = a(\text{Pred\_sample}) * (\text{Orig\_sample} - \text{Pred\_sample}), \quad (4)$$

[0082] 其中,a(Pred\_sample)表示基于Pred\_sample的值的缩放因子。根据等式(3)和(4),等式(3)可以近似为:

$$[0083] \quad \text{Reco\_sample} = \text{Pred\_sample} + (1/a(\text{Pred\_sample})) * \text{Res}_r, \quad (5)$$

[0084] 因此,在实施例中,仅需要向解码器传送用于分段线性模型的缩放因子a(Pred\_sample)。

[0085] 图3E和图3F描绘了用于使用预测残差的环内整形对视频进行编码(300C\_E)和解码(300C\_D)的示例过程流程。所述过程与图3A和图3B中描述的那些过程非常相似,并且因此不需加以说明。

[0086] 表1总结了所提出的这三种架构的关键特征。

[0087] 表1:所考虑整形架构的关键特征

	架构	环外	环内仅帧内	环内残差
	<b>DPB 存储</b>	经整形的图片	帧内模式：经逆向整形的图片 帧间模式：没有整形	未整形的图片
	<b>帧内预测执行对象</b>	经整形的图片	经整形的图片	未整形的图片
	<b>帧间预测（运动估计）执行对象</b>	经整形的图片	未整形的图片	未整形的图片
	<b>是否需要额外的图片缓冲器</b>	是（需要缓冲器来将经整形的图片保存在DPB中并且保存未整形的图片以进行输出）	否（实时替换图片样本）	否（实时替换残差样本）
[0088]	<b>自适应整形估计的位置/频率</b>	不受限制（可以是仅帧内、基于场景、或可配置的）	仅帧内图片	不受限制（可以是仅帧内、基于场景、或可配置的）
	<b>样本修改（整形）的复杂度</b>	处理所有图片	仅处理帧内图片（最低复杂度）	处理残差，不管帧内还是帧外
	<b>环路滤波器交互</b>	使用经整形的图片作为参考进行优化	使用原始图片作为参考进行优化	使用原始图片作为参考进行优化
	<b>自适应的块级/区域级整形的可能性</b>	否	否	是
	<b>其他方面</b>	如果参考帧具有不同的整形函数，则帧间预测性能可能受到影响		帧间预测可以将整形器用于当前图片以处理与参考帧（其本身可能具有不同的整形器）的残差
		解码器侧仅需要逆向整形函数	解码器侧仅需要逆向整形函数	解码器需要前向整形函数和逆向整形函数两者

[0089] 图4A和图4B描绘了用于使用所提出的这三种架构的组合来进行编码和解码的示例编码处理流程和解码处理流程。如图4A中描绘的，如果未启用整形，则根据已知的视频编码技术（例如，HEVC等）来对输入视频进行编码，而不使用任何整形。否则，编码器可以根据目标接收器的能力和/或输入特性来选择主要提出的这三种方法中的任何一种。例如，在实施例1中，编码器可以在场景级在这些方法之间切换，其中，‘场景’被表示为具有相似明度特性的连续帧序列。在另一实施例中，在序列参数集（SPS）级中定义了高级参数。

[0090] 如图4B中描绘的，解码器可以根据所接收的整形信息信令来调用相应解码过程中的任何解码过程，用于对传入的编码比特流进行解码。

#### [0091] 混合环内整形

[0092] 图2G描绘了用于使用混合环内整形架构的编码器的示例架构（200D\_E）。这个架构组合了来自前面讨论的环内仅帧内整形架构（200B\_E）和环内残差架构（200C\_E）两者的要素。在这个架构下，根据环内帧内整形编码架构（例如，图2C中的200B\_E）来对帧内切片进行编码，只是有一点不同：对于帧内切片，在环路滤波（270-1）之后执行逆向图片整形（265-1）。在另一实施例中，可以在逆向整形之后对帧内切片执行环内滤波；然而，实验结果表明，

这样的布置与在环路滤波之后执行逆向整形相比可能产生更差的编码效率。其余操作保持与前面讨论的相同。

[0093] 如前面所讨论的,根据环内残差编码架构(例如,图2E中的200C\_E)来对帧间切片进行编码。如图2G中描绘的,帧内/帧间切片切换允许根据要编码的切片类型而在这两种架构之间切换。

[0094] 图2H描绘了用于使用混合环内整形的解码器的示例架构(200D\_D)。再次,根据环内帧内整形解码器架构(例如,图2D中的200B\_D)来对帧内切片进行解码,其中,再次对于帧内切片,环路滤波(270-1)在逆向图片整形(265-1)之前。根据环内残差编码架构(例如,图2F中的200C\_D)来对帧间切片进行解码。如图2H中描绘的,帧内/帧间切片切换允许根据经编码视频图片中的切片类型而在这两种架构之间切换。

[0095] 通过调用图2G中描绘的编码过程300D-E,图4A可以容易地扩展为还包括混合环内整形编码方法。类似地,通过调用图2H中描绘的解码过程300D-D,图4B可以容易地扩展为还包括混合环内整形解码方法。

#### [0096] 切片级的整形

[0097] 本发明的实施例允许各种切片级的适配。例如,为了减少计算,可以仅对帧内切片或仅对帧间切片启用整形。在另一实施例中,可以基于时间ID(例如,HEVC的变量TemporalId(参考文献[11]),其中,TemporalId=nuh\_temporal\_id\_plus1-1)的值来允许整形。例如,如果当前切片的TemporalId小于或等于预定义值,则可以将当前切片的slice\_resaper\_enable\_flag设置为1,否则,slice\_resaper\_enable\_flag将为0。为了避免针对每个切片发送slice\_resaper\_enable\_flag参数,可以在SPS级指定sps\_resaper\_temporal\_id参数,因此可以推断slice\_resaper\_enable\_flag参数的值。

[0098] 对于启用了整形的切片,解码器需要知道要使用哪个整形模型。在一个实施例中,可以始终使用在SPS级定义的整形模型。在另一实施例中,可以始终使用在切片报头中定义的整形模型。如果在当前切片中没有定义整形模型,则可以应用在最近解码的已使用整形的切片中使用的整形模型。在另一实施例中,无论是否将整形用于帧内切片,都可以始终在帧内切片中指定整形模型。在这样的实施方式中,需要对参数slice\_resaper\_enable\_flag和slice\_resaper\_model\_present\_flag进行解除关联。表5中描绘了这种切片语法的示例。

#### [0099] 整形信息的信令

[0100] 与前向和/或逆向整形相关的信息可以存在于不同的信息层,例如,存在于视频参数集(VPS)、序列参数集(SPS)、图片参数集(PPS)、切片报头、补充信息(SEI)、或任何其他高级语法。作为示例而非限制性地,表2提供了SPS中的用于发信号通知是否启用了整形、整形是否是自适应的、以及正在使用这三种架构中的哪一种的高级语法的示例。

[0101] 表2:SPS中的整形信息的示例

	SPS()	描述符
	.....	
	<b>sps_resaper_enable_flag</b> /* 1: 打开整形, 否则关闭*/	u(1)
[0102]	if (sps_resaper_enable_flag) {	
	<b>sps_resaper_adaptive_flag</b> /* 1: 打开自适应整形, 否则关闭*/	u(1)
	<b>sps_resaper_architecture</b> /*例如: 0: 环外, 1: 环内帧内, 2: 环内残差*/	ue(v)
	}	
	.....	

[0103] 还可以在某个其他层(比如在切片报头中)携带附加信息。整形函数可以通过查找表(LUT)、分段多项式或其他种类的参数化模型来描述。可以通过附加语法元素(例如, `reshaping_model_type`标志)来发信号通知用于传送整形函数的整形模型的类型。例如,考虑使用以下两个不同表示的系统:`model_A`(例如,`reshaping_model_type=0`)将整形函数表示为一组分段多项式(例如,参见参考文献[4]),而在`model_B`(例如,`reshaping_model_type=1`)中,整形函数是通过基于图片明度特性和视觉重要性(例如,参见参考文献[3])来将码字分派给不同的明度频带而自适应地得到的。表3提供了图片的切片报头中的语法元素的示例,用于辅助解码器确定正在使用的适当整形模型。

[0104] 表3:用于切片报头中的整形信令的示例语法

	slice_segment_header()	描述符
	.....	
	if (sps_resaper_adaptive_flag) {	
	<b>reshaping_model_type</b>	ue(v)
[0105]	if (reshaping_model_type == model_A) {	
	reshaping_sliceheader_table_model_A()	
	}	
	else if (reshaping_model_type == model_B) {	
	reshaping_sliceheader_table_model_B()	
	}	
	else ...	
	}	
	.....	

[0106] 以下三个表描述了用于在序列层、切片层或编码树单元 (CTU) 层处进行信号整形的比特流语法的替代性示例。

[0107] 表4:SPS中的整形信息的示例

SPS()		描述符
.....		
	<b>sps_reshaper_enable_flag</b> /* 1: 打开整形, 否则关闭*/	u(1)
	if(sps_reshaper_enable_flag) {	
	<b>sps_reshaper_signal_type</b> /* 0: SDR, 1: PQ, 2: HLG */	u(2)
[0108]	<b>sps_reshaper_ILF_opt</b> /*在哪个域中的环路滤波器: 2 比特 帧间/帧内*/	u(2)
	<b>sps_reshaper_chromaAdj</b> /* 1: 色度 DQP; 2: 色度缩放*/	u(2)
	<b>sps_reshaper_model_present_flag</b> /* 1: 存在*/	u(1)
	if(sps_reshaper_model_present_flag)	
	sps_reshaper_model()	
	}	

[0109] 表5:用于切片报头中的整形信令的示例语法

slice_header()		描述符
.....		
	<b>slice_reshaper_model_present_flag</b>	u(1)
	if ( slice_reshaper_model_present_flag )	
[0110]	slice_reshaper_model()	
	<b>slice_reshaper_enable_flag</b>	u(1)
	if ( slice_reshaper_enable_flag ) {	
	<b>reshaper_CTU_control_flag</b> /* 1: 开, CTU级开/关标志*/	u(1)
	}	
	.....	

[0111] 表6:用于CTU中的整形信令的示例语法

coding_tree_unit()		描述符
.....		
[0112]	if ( reshape_CTU_control_flag ) {	
	<b>reshaper_CTU_flag</b>	ae(v)
	}	
	.....	

[0113] 对于表4至表6,示例语义可以表示为:

[0114] sps\_reshaper\_enable\_flag等于1指定在编码视频序列(CVS)中使用了整形器。sps\_reshaper\_enabled\_flag等于0指定在CVS中未使用整形器。

[0115] slice\_reshaper\_enable\_flag等于1指定当前切片启用了整形器。slice\_reshaper\_enable\_flag等于0指定当前切片未启用整形器。

[0116] sps\_reshaper\_signal\_type表明原始码字分布或表示。作为示例而非限制性地,sps\_reshaper\_signal\_type等于0指定SDR(伽马);sps\_reshaper\_signal\_type等于1指定PQ;并且sps\_reshaper\_signal\_type等于2指定HLG。

[0117] reshaper\_CTU\_control\_flag等于1表明允许整形器被适配用于每个CTU。reshaper\_CTU\_control\_flag等于0表明不允许整形器被适配用于每个CTU。当reshaper\_

CUT\_control\_flag不存在时,应推断该值为0。

[0118] reshaper\_CTU\_flag等于1指定对当前CTU使用整形器。reshaper\_CUT\_flag等于0指定当前CTU不使用整形器。当reshaper\_CTU\_flag不存在时,应推断该值为等于slice\_reshaper\_enabled\_flag。

[0119] sps\_reshaper\_model\_present\_flag等于1表明sps中存在sps\_reshaper\_model()。sps\_reshaper\_model\_present\_flag等于0表明SPS中不存在sps\_reshaper\_model()。

[0120] slice\_reshaper\_model\_present\_flag等于1表明切片报头中存在slice\_reshaper\_model()。slice\_reshaper\_model\_present\_flag等于0表明SPS中不存在slice\_reshaper\_model()。sps\_reshaper\_chromaAdj等于1表明使用色度DQP完成了色度QP调整。sps\_reshaper\_chromaAdj等于2表明使用色度缩放完成了色度QP调整。

[0121] sps\_reshaper\_ILF\_opt表明:对于帧内切片和帧间切片,是在原始域还是整形域中应用环内滤波器。例如,使用两位语法,其中,最低有效位是指帧内切片:

sps_reshaper_ILF_opt	环内滤波器操作
0 0	对于帧内和帧间均在原始域中
0 1	对于帧间在原始域中,对于帧内在整形域中
1 0	对于帧间在整形域中,对于帧内在原始域中
1 1	对于帧内和帧间均在整形域中

[0123] 在一些实施例中,可以在切片级调整这个参数。例如,在实施例中,当slice\_reshaper\_enable\_flag被设置为1时,切片可以包括slice\_reshape\_ILFOPT\_flag。在另一实施例中,在SPS中,如果启用sps\_reshaper\_ILF\_opt,则可以包括sps\_reshaper\_ILF\_Tid参数。如果当前切片的TemporalID≤sps\_reshaper\_ILF\_Tid并且slice\_reshaper\_enable\_flag被设置为1,则在整形域中应用环内滤波器。否则,在未整形域中应用环内滤波器。

[0124] 在表4中,在SPS级控制色度QP调整。在实施例中,还可以在切片级控制色度QP调整。例如,在每个切片中,当slice\_reshaper\_enable\_flag被设置为1时,可以添加语法元素slice\_reshape\_chromaAdj\_flag。在另一实施例中,在SPS中,如果启用sps\_reshaper\_chromaAdj,则可以添加语法元素sps\_reshaper\_ChromaAdj\_Tid。如果当前切片的TemporalID≤sps\_reshaper\_ChromaAdj\_Tid并且slice\_reshaper\_enable\_flag被设置为1,则应用色度调整。否则,不应用色度调整。表4B使用前面描述的语法描绘了表4的示例变型。

[0125] 表4B: 用于SPS中使用时间ID的整形信令的示例语法

	SPS()	描述符
	.....	
	<b>sps_resaper_enable_flag</b> /* 1: 打开整形, 否则关闭*/	u(1)
	if(sps_resaper_enable_flag) {	
	<b>sps_resaper_signal_type</b> /* 0: HDR, 1: PQ, 2: HLG */	u(2)
	<b>sps_resaper_ILF_opt</b> /*在哪个域中的环路滤波器: 2 比特 帧间/帧内*/	u(2)
	if(sps_resaper_ILF_opt == 3)	
[0126]	<b>sps_resaper_ILF_Tid</b>	u(3)
	<b>sps_resaper_chromaAdj</b> /* 1: 色度 DQP; 2: 色度缩放*/	u(2)
	if(sps_resaper_chromaAdj)	
	<b>sps_resaper_chromaAdj_Tid</b>	u(3)
	<b>sps_resaper_model_present_flag</b> /* 1: 存在*/	u(1)
	if(sps_resaper_model_present_flag)	
	sps_resaper_model()	
	}	

[0127] sps\_resaper\_ILF\_Tid指定对整形域中的经整形切片应用环内滤波器的最高TemporalID。sps\_resaper\_chromaAdj\_Tid指定对经整形切片应用色度调整的最高TemporalID。

[0128] 在另一实施例中,可以使用整形模型ID(例如,reshape\_model\_id)来定义整形模型,例如,作为slice\_resaper\_model()函数的一部分。可以在SPS级、PPS级或切片报头级用信号传送整形模型。如果在SPS或PPS中发出信号,则还可以从sps\_seq\_parameter\_set\_id或pps\_pic\_parameter\_set\_id推断出reshape\_model\_id的值。在下面的表5B中示出了如何对不携带slice\_resaper\_model() (例如,slice\_resaper\_model\_present\_flag等于0)的切片使用reshape\_model\_id的示例,表5B是表5的变型。

[0129] 表5B:用于切片报头中使用reshape\_model\_id的整形信令的示例语法

	slice_header()	描述符
	.....	
	<b>slice_resaper_model_present_flag</b>	u(1)
	if ( slice_resaper_model_present_flag )	
	slice_resaper_model ( )	
	else	
[0130]	<b>reshape_model_id</b>	ue(v)
	slice_resaper_enable_flag	u(1)
	if ( slice_resaper_enable_flag ) {	
	<b>resaper_CTU_control_flag</b> /*1: 开, CTU 级开/关标志*/	u(1)
	}	
	.....	

[0131] 在示例语法中,参数reshape\_model\_id指定正在使用的reshape\_model的值。reshape\_model\_id的值应当在0到15的范围内。

[0132] 作为使用所提出的语法的示例,考虑使用PQ EOTF编码的HDR信号,其中,在SPS级使用整形,在切片级不使用特定的整形(对所有切片使用整形),并且仅对帧间切片允许CTU

适配。则：

[0133] sps\_reshaper\_signal\_type=1(PQ)；

[0134] sps\_reshaper\_model\_present\_flag=1；

[0135] //注释：对于帧间切片，可以操纵slice\_reshaper\_enable\_flag来启用和禁用整形器。

```
slice_reshaper_enable_flag = 1;
```

```
if (CTUAdp)
```

```
{
```

```
    if (I_slice)
```

```
        slice_reshaper_model_present_flag = 0;
```

```
        reshaper_CTU_control_flag = 0;
```

[0136] else

```
        slice_reshaper_model_present_flag = 0;
```

```
        reshaper_CTU_control_flag = 1;
```

```
    }
```

```
else
```

```
{
```

```
    slice_reshaper_model_present_flag = 0;
```

```
    reshaper_CTU_control_flag = 0;
```

```
}
```

[0137] 在另一示例中，考虑SDR信号，其中仅在切片级并且仅针对帧内切片应用整形。仅对帧间切片允许CTU整形适配。则：

```

sps_reshaper_signal_type = 0 (SDR);
sps_reshaper_model_present_flag = 0;
slice_reshaper_enable_flag = 1;
if (L_slice)
{
    slice_reshaper_model_present_flag = 1;
    reshaper_CTU_control_flag = 0;
}
[0138] else
{
    slice_reshaper_model_present_flag = 0;
    if (CTUAdp)
        reshape_CTU_control_flag = 1;
    else
        reshaper_CTU_control_flag = 0;
}

```

[0139] 在CTU级,在实施例中,可以基于CTU的明度特性来启用CTU级整形。例如,对于每个CTU,可以计算平均明度(例如,CTU\_avg\_lum\_value),将平均明度与一个或多个阈值进行比较,并基于这些比较的结果来决定是打开还是关闭整形。例如,

[0140] 如果CTU\_avg\_lum\_value < THR1,或者

[0141] 如果CTU\_avg\_lum\_value > THR2,或者

[0142] 如果THR3 < CTU\_avg\_lum\_value < THR4,

[0143] 则对于该CTU,reshaper\_CTU\_Flag = 1。

[0144] 在实施例中,代替使用平均明度,可以使用CTU的某个其他明度特性,诸如最小明度、最大明度、或平均明度、方差等。还可以应用CTU的基于色度的特性,或者可以将明度特性和色度特性与阈值进行组合。

[0145] 如前所述(例如,关于图3A、图3B和图3C中的步骤),实施例可以支持默认或静态整形函数、或自适应整形。“默认整形器”可以用于执行预定义的整形函数,因此降低了在得到整形曲线时分析每个图片或场景的复杂度。在这种情况下,不需要在场景、图片或切片级用信号传送逆向整形函数。默认整形器可以通过使用存储在解码器中的固定映射曲线来实施以避免任何信令,或者其可以作为序列级参数集的一部分用信号传送一次。在另一实施例中,先前解码的自适应整形函数可以按编码顺序重新用于后面的图片。在另一实施例中,可以与先前解码的方式不同的方式用信号传送整形曲线。在其他实施例中,(例如,对于需要Inv()函数和Fwd()函数两者来执行逆向整形的环内残差整形),可以在比特流中仅用信号传送Inv()函数或Fwd()函数中的一个,或者可替代地,用信号传送这两个函数以降低解码器复杂度。表7和表8提供了用于用信号传送整形信息的两个示例。

[0146] 在表7中,将整形函数作为一组二阶多项式进行传送。其是探索性测试模型(ETM)(参考文献[5])的简化语法。还可以在参考文献[4]中找到更早的变型。

[0147] 表7:用于整形函数的分段表示(model\_A)的示例语法

	描述符
reshaping_sliceheader_table_model_A() {	
reshape_input_luma_bit_depth_minus8	ue(v)
coeff_log2_offset_minus2	ue(v)
reshape_num_ranges_minus1	ue(v)
reshape_equal_ranges_flag	u(1)
reshape_global_offset_val	u(v)
if( !reshape_equal_ranges_flag)	
for ( i = 0; i < reshape_num_ranges_minus1+ 1; i++ )	
reshape_range_val[ i ]	u(v)
[0148] reshape_continuity_flag	u(1)
for( i = 0; i < reshape_num_ranges_minus1 + 2; i++ ) {	
reshape_poly_coeff_order0_int[ i ]	ue(v)
reshape_poly_coeff_order0_frac[ i ]	u(v)
}	
if( reshape_continuity_flag == 1 ) {	
reshape_poly_coeff_order1_int	se(v)
reshape_poly_coeff_order1_frac	u(v)
}	
}	

[0149] reshape\_input\_luma\_bit\_depth\_minus8指定整形过程的输入亮度分量的样本比特深度。coeff\_log2\_offset\_minus2指定用于亮度分量的整形相关系数计算的小数位数。coeff\_log2\_offset\_minus2的值应当在0到3的范围内(包括端点)。

[0150] reshape\_num\_ranges\_minus1加1,指定分段整形函数中的范围的数量。当reshape\_num\_ranges\_minus1不存在时,推断其值为0。reshape\_num\_ranges\_minus1应当在0到7的范围内(对于亮度分量包括端点)。

[0151] reshape\_equal\_ranges\_flag等于1,指定分段整形函数被分割为具有几乎相等长度的NumberRanges段,并且未明确地发信号通知每个范围的长度。reshape\_equal\_ranges\_flag等于0指定明确地发信号通知了每个范围的长度。

[0152] reshape\_global\_offset\_val用于得到用于指定第0个范围的起始点的偏移值。

[0153] reshape\_range\_val[i]用于得到亮度分量的第i个范围的长度。

[0154] reshape\_continuity\_flag指定亮度分量的整形函数的连续性属性。如果reshape\_continuity\_flag等于0,则将零阶连续性应用于连续枢轴点之间的分段线性逆向整形函数。如果reshape\_continuity\_flag等于1,则使用一阶平滑度来得到连续枢轴点之间的完整二阶多项式逆向整形函数。

[0155] reshape\_poly\_coeff\_order0\_int[i]指定亮度分量的第i段0阶多项式系数的整数值。

[0156] reshape\_poly\_coeff\_order0\_frac[i]指定亮度分量的第i段0阶多项式系数的小数值。

[0157] reshape\_poly\_coeff\_order1\_int指定亮度分量的1阶多项式系数的整数值。

[0158] reshape\_poly\_coeff\_order1\_frac指定亮度分量的1阶多项式系数的小数值。

[0159] 表8描述了根据前面讨论的model\_B(参考文献[3])的替代性参数化表示的示例实施例。

[0160] 表8:用于整形函数的参数化表示(model\_B)的示例语法

	描述符
reshaping_sliceheader_table_model_B() {	
reshape_model_profile_type	ue(v)
reshape_model_scale_idx	u(2)
reshape_model_min_bin_idx	u(5)
reshape_model_max_bin_idx	u(5)
reshape_model_num_band	u(4)
for ( i = 0; i < reshape_model_num_band; i++ ) {	
reshape_model_band_profile_delta [ i ]	u(1)
}	
}	

[0162] 在表8中,在实施例中,语法参数可以被定义为:reshape\_model\_profile\_type指定在整形器构建过程中要使用的分布类型。

[0163] reshape\_model\_scale\_idx指定在整形器构建过程中要使用的缩放因子(表示为ScaleFactor)的索引值。ScaleFactor的值允许改进对整形函数的控制,以提高整体编码效率。关于对整形函数重建过程的讨论(例如,如图5A和图5B中描绘的)提供了关于使用此ScaleFactor的附加细节。作为示例而非限制性地,reshape\_model\_scale\_idx的值应当在0到3的范围内(包括端点)。在实施例中,如下表所示的scale\_idx与ScaleFactor之间的映射关系由下式给出:

[0164]  $ScaleFactor = 1.0 - 0.05 * reshape\_model\_scale\_idx.$

[0165]

reshape_model_scale_idx	ScaleFactor
0	1.0
1	0.95
2	0.9
3	0.85

[0166] 在另一实施例中,对于更高效的定点实施方式,

[0167]  $ScaleFactor = 1 - 1/16 * reshape\_model\_scale\_idx.$

[0168]

reshape_model_scale_idx	ScaleFactor
0	1.0
1	0.9375
2	0.875
3	0.8125

[0169] reshape\_model\_min\_bin\_idx指定在整形器构建过程中要使用的最小区间索引。

[0170] reshape\_model\_min\_bin\_idx的值应当在0到31的范围内,包括端点。

[0171] reshape\_model\_max\_bin\_idx指定在整形器构建过程中要使用的最大区间索引。

[0172] reshape\_model\_max\_bin\_idx的值应当在0到31的范围内,包括端点。

[0173] reshape\_model\_num\_band指定在整形器构建过程中要使用的频带数。

[0174] reshape\_model\_num\_band的值应当在0到15的范围内,包括端点。

[0175] reshape\_model\_band\_profile\_delta[i]指定在整形器构建过程中要用于调整第i个频带的分布的增量值。reshape\_model\_band\_profile\_delta[i]的值应当在0到1的范围

内,包括端点。

[0176] 与参考文献[3]相比,通过定义一组“默认分布类型”(比如高光、中间调和黑暗),表8中的语法远更高效。在实施例中,每种类型都具有预定义的视觉频带重要性分布。预定义的频带和相应的分布可以在解码器中实施为固定值,或者它们还可以使用高级语法(诸如序列参数集)来用信号传送。在编码器处,首先分析每个图像并将图像分类为分布类型之一。分布类型通过语法元素“`reshape_model_profile_type`”来用信号传送。在自适应整形中,为了捕获全范围的图像动态,通过针对每个明度频带或明度频带子集的增量来进一步调整默认分布。增量值是基于明度频带的视觉重要性得到的,并且通过语法元素“`reshape_model_band_profile_delta`”来用信号传送。

[0177] 在一个实施例中,增量值可以仅取值0或1。在编码器处,通过将整个图像中的频带像素的百分比与“主频带”内的频带像素的百分比进行比较来确定视觉重要性,其中,可以使用局部直方图来检测主频带。如果频带内的像素集中在小的局部框中,则所述频带很可能在所述框是视觉重要的。对主频带的计数进行求和并归一化以形成有意义的比较从而获得每个频带的增量值。

[0178] 在解码器中,必须调用整形器函数重建过程以基于参考文献[3]中描述的方法得到整形LUT。因此,与仅需要评估分段多项式函数来计算LUT的更简单的分段近似模型相比,复杂度更高。使用参数化模型语法的好处是可以显著降低使用整形器的比特率。例如,基于典型的测试内容,表7中描绘的模型需要200至300比特来用信号传送整形器,而参数化模型(如表8中所示)仅使用大约40比特。

[0179] 在另一实施例中,如表9中描绘的,可以根据dQP值的参数化模型得到前向整形查找表。例如,在实施例中,

[0180]  $dQP = \text{clip3}(\min, \max, \text{scale} * X + \text{offset})$ ,

[0181] 其中, $\min$ 和 $\max$ 表示dQP的边界, $\text{scale}$ 和 $\text{offset}$ 是模型的两个参数,并且 $X$ 表示基于信号明度得到的参数(例如,像素的明度值,或者对于框而言,框明度的度量(例如,其最小值、最大值、平均值、方差、标准偏差等))。例如,非限制性地,

[0182]  $dQP = \text{clip3}(-3, 6, 0.015 * X - 7.5)$ 。

[0183] 表9:用于整形函数的参数化表示(模型C)的示例语法

	描述符
sps_reshaper_model_C() {	
<b>full_range_input_flag</b>	u(1)
<b>dQP_model_scale_int_prec</b>	ue(v)
if (dQP_model_scale_int_prec > 0) {	
<b>dQP_model_scale_int</b>	u(v)
}	
<b>dQP_model_scale_frac_prec_minus16</b>	ue(v)
<b>dQP_model_scale_frac</b>	u(v)
if (dQPModelScaleAbs) {	
<b>dQP_model_scale_sign</b>	u(1)
}	
<b>dQP_model_offset_int_prec_minus3</b>	ue(v)
<b>dQP_model_offset_int</b>	u(v)
[0184] <b>dQP_model_offset_frac_prec_minus1</b>	ue(v)
<b>dQP_model_offset_frac</b>	u(v)
if (dQPModelOffsetAbs) {	
<b>dQP_model_offset_sign</b>	u(1)
}	
<b>dQP_model_abs_prec_minus3</b>	ue(v)
<b>dQP_model_max_abs</b>	u(v)
if (dQP_model_max_abs) {	
<b>dQP_model_max_sign</b>	u(1)
}	
<b>dQP_model_min_abs</b>	u(v)
if (dQP_model_min_abs) {	
<b>dQP_model_min_sign</b>	u(1)
}	
}	

[0185] 在实施例中,表9中的参数可以定义如下:

[0186] **full\_range\_input\_flag**指定输入视频信号的范围。**full\_range\_input\_flag**为0对应于标准动态范围输入视频信号。**full\_range\_input\_flag**为1对应于全范围输入视频信号。当**full\_range\_input\_flag**不存在时,其被推断为0。

[0187] 注意:如本文所使用的,术语“全范围视频”表示视频中的有效码字不是“受限的”。例如,对于10比特全范围视频,有效码字在0与1023之间,其中,0被映射到最低明度级。相反,对于10比特“标准范围视频”,有效码字在64与940之间,并且64被映射到最低明度级。

[0188] 例如,“全范围”和“标准范围”的计算可以计算如下:

[0189] 对于[0 1]中的归一化明度值 $E_{y'}$ ,以BD比特(例如,BD=10、12等)进行编码:

[0190] 全范围: $Y = \text{clip3}(0, (1 \ll \text{BD}) - 1, E_{y'} * ((1 \ll \text{BD}) - 1))$

[0191] 标准范围: $Y = \text{clip3}(0, (1 \ll \text{BD}) - 1, \text{round}(1 \ll (\text{BD} - 8) * (219 * E_{y'} + 16)))$

[0192] 该语法类似于如HEVC(H.265)规范(参考文献[11])的章节E.2.1中描述的HEVCVUI参数中的“**video\_full\_range\_flag**”语法。

[0193] **dQP\_model\_scale\_int\_prec**指定用于表示**dQP\_model\_scale\_int**的比特数。**dQP\_model\_scale\_int\_prec**等于0表明**dQP\_model\_scale\_int**未被用信号传送并且被推断为0。

[0194] **dQP\_model\_scale\_int**指定dQP模型缩放的整数值。

[0195] **dQP\_model\_scale\_frac\_prec\_minus16**加16,指定用于表示**dQP\_model\_scale**

frac的比特数。

[0196] dQP\_model\_scale\_frac指定dQP模型缩放的小数值。

[0197] 如下得到变量dQPModelScaleAbs:

[0198]  $dQPModelScaleAbs = dQP\_model\_scale\_int \ll (dQP\_model\_scale\_frac\_prec\_minus16+16) + dQP\_model\_scale\_ffrac$

[0199] dQP\_model\_scale\_sign指定dQP模型缩放的符号。当dQPModelScaleAbs等于0时, dQP\_model\_scale\_sign未被用信号传送并且其被推断为0。

[0200] dQP\_model\_offset\_int\_prec\_minus3加3,指定用于表示dQP\_model\_offset\_int的比特数。

[0201] dQP\_model\_offset\_int指定dQP模型偏移的整数值。

[0202] dQP\_model\_offset\_frac\_prec\_minus1加1,指定用于表示dQP\_model\_offset\_frac的比特数。

[0203] dQP\_model\_offset\_frac指定dQP模型偏移的小数值。

[0204] 如下得到变量dQPModelOffsetAbs:

[0205]  $dQPModelOffsetAbs = dQP\_model\_offset\_int \ll (dQP\_model\_offset\_frac\_prec\_minus1+1) + dQP\_model\_offset\_fiac$

[0206] dQP\_model\_offset\_sign指定dQP模型偏移的符号。当dQPModelOffsetAbs等于0时, dQP\_model\_offset\_sign未被用信号传送并且被推断为0。

[0207] dQP\_model\_abs\_prec\_minus3加3,指定用于表示dQP\_model\_max\_abs和dQP\_model\_min\_abs的比特数。

[0208] dQP\_model\_max\_abs指定dQP模型最大值的整数值。

[0209] dQP\_model\_max\_sign指定dQP模型最大值的符号。当dQP\_model\_max\_abs等于0时, dQP\_model\_max\_sign未被用信号传送并且被推断为0。

[0210] dQP\_model\_min\_abs指定dQP模型最小值的整数值。

[0211] dQP\_model\_min\_sign指定dQP模型最小值的符号。当dQP\_model\_min\_abs等于0时, dQP\_model\_min\_sign未被用信号传送并且被推断为0。

[0212] 模型C的解码过程

[0213] 在给定了表9的语法元素的情况下,可以如下得到整形LUT。

[0214] 如下得到变量dQPModelScaleFP:

[0215]  $dQPModelScaleFP = ((1-2*dQP\_model\_scale\_sign)*dQPModelScaleAbs) \ll (dQP\_model\_offset\_frac\_prec\_minus1+1)$ .

[0216] 如下得到变量dQPModelOffsetFP:

[0217]  $dQPModelOffsetFP = ((1-2*dQP\_model\_offset\_sign)*dQPModelOffsetAbs) \ll (dQP\_model\_scale\_frac\_prec\_minus16+16)$ .

[0218] 如下得到变量dQPModelShift:

[0219]  $dQPModelShift = (dQP\_model\_offset\_frac\_prec\_minus1+1) + (dQP\_model\_scale\_frac\_prec\_minus16+16)$ .

[0220] 如下得到变量dQPModelMaxFP:

[0221]  $dQPModelMaxFP = ((1-2*dQP\_model\_max\_sign)*dQP\_model\_max\_abs) \ll$

dQPModelShift.

[0222] 如下得到变量dQPModelMinFP:

[0223]  $dQPModelMinFP = ((1 - 2 * dQP\_model\_min\_sign) * dQP\_model\_min\_abs) \ll dQPModelShift.$

[0224] for Y=0:maxY//例如,对于10比特视频,maxY=1023

[0225] {

[0226]  $dQP[Y] = clip3(dQPModelMinFP, dQPModelMaxFP, dQPModelScaleFP * Y + dQPModelOffsetFP);$

[0227]  $slope[Y] = exp2((dQP[Y] + 3) / 6);$  //exp2定点实施方式,其中,  $exp2(x) = 2^x$  ;}

[0228] If (full\_range\_input\_flag == 0) //如果输入是标准范围视频

[0229] 对于在标准范围之外的Y (即Y = [0:63] and [940:1023]), 设置slope[Y] = 0;

[0230] CDF[0] = slope[0];

[0231] for Y=0:maxY-1

[0232] {

[0233]  $CDF[Y+1] = CDF[Y] + slope[Y];$  //CDF[Y]是slope[Y]的积分

[0234] }

[0235] for Y=0:maxY

[0236] {

[0237]  $FwdLUT[Y] = round(CDF[Y] * maxY / CDF[maxY]);$  //舍入并归一化以获得FwdLUT

[0238] }

[0239] 在另一实施例中,如表10中描绘的,前向整形函数可以表示为亮度枢轴点(In\_Y)及其相应码字(Out\_Y)的集合。为了简化编码,根据起始枢轴和等间隔后续枢轴的序列、使用线性分段表示来描述输入亮度范围。在图7中描绘了表示用于10比特输入数据的前向整形函数的示例。

[0240] 表10:用于整形函数的基于枢轴表示(模型D)的示例语法

	描述符
<b>sps_resaper_model_D() {</b>	
<b>full_range_input_flag</b>	u(1)
<b>bin_pivot_start</b>	u(v)
<b>bin_cw_start</b>	u(v)
<b>log2_num_equal_bins_minus3</b>	ue(v)
<b>equal_bin_pivot_delta</b>	u(v)
<b>bin_cw_in_first_equal_bin</b>	u(v)
[0241] <b>bin_cw_delta_abs_prec_minus4</b>	ue(v)
for(i = 0 ; i < NumEqualBins - 1 ; i++) {	
<b>bin_cw_delta_abs[ i ]</b>	u(v)
if ( bin_cw_delta_abs[ i ] ) {	
<b>bin_cw_delta_sign[ i ]</b>	u(1)
}	
}	
}	

[0242] 在实施例中,表10中的参数可以定义如下:

[0243] full\_range\_input\_flag指定输入视频信号的范围。full\_range\_input\_flag为0

对应于标准范围输入视频信号。full\_range\_input\_flag为1对应于全范围输入视频信号。当full\_range\_input\_flag不存在时,其被推断为0。

[0244] bin\_pivot\_start指定第一等长区间的枢轴值(710)。当full\_range\_input\_flag等于0时,bin\_pivot\_start应当大于或等于最小标准范围输入,并且应当小于最大标准范围输入。(例如,对于10比特SDR输入,bin\_pivot\_start(710)应当在64与940之间)。

[0245] bin\_cw\_start指定bin\_pivot\_start(710)的映射值(715)(例如,bin\_cw\_start=FwdLUT[bin\_pivot\_start])。

[0246] log2\_num\_equal\_bins\_minus3加3,指定在起始枢轴(710)之后的等长区间的数量。变量NumEqualBins和NumTotalBins由以下定义:

[0247]  $\text{NumEqualBins} = 1 \ll (\text{log2\_num\_equal\_bins\_minus3} + 3)$

[0248] 如果full\_range\_input\_flag == 0

[0249]  $\text{NumTotalBins} = \text{NumEqualBins} + 4$

[0250] 否则

[0251]  $\text{NumTotalBins} = \text{NumEqualBins} + 2$

[0252] 注意:实验结果表明,大多数前向整形函数可以使用八个等长片段来表示;然而,复杂的整形函数可能需要更多的片段(例如,16个片段或更多片段)。

[0253] equal\_bin\_pivot\_delta指定等长区间的长度(例如,720-1,720-N)。NumEqualBins\*equal\_bin\_pivot\_delta应当小于或等于有效输入范围。(例如,如果full\_range\_input\_flag为0,则对于10比特输入,有效输入范围应当为940-64=876;如果full\_range\_input\_flag为1,则对于10比特输入,有效输入范围应当为0到1023。)

[0254] bin\_cw\_in\_first\_equal\_bin指定第一等长区间(720-1)中的映射码字的数量(725)。

[0255] Bin\_cw\_delta\_abs\_prec\_minus4加4,指定用于表示每个后续相等区间的bin\_cw\_delta\_abs[i]的比特数。

[0256] bin\_cw\_delta\_abs[i]指定每个后续等长区间的bin\_cw\_delta\_abs[i]的值。bin\_cw\_delta[i](例如,735)是当前等长区间i(例如,720-N)中的码字(例如,740)与前一等长区间i-1中的码字(例如,730)相比的差异。

[0257] bin\_cw\_delta\_sign[i]指定bin\_cw\_delta\_abs[i]的符号。当bin\_cw\_delta\_abs[i]等于0时,bin\_cw\_delta\_sign[i]未被用信号传送并且被推断为0。变量bin\_cw\_delta[i] = (1-2\*bin\_cw\_delta\_sign[i])\*bin\_cw\_delta\_abs[i]

[0258] 模型D的解码过程

[0259] 在给定了表10的语法元素的情况下,对于10比特输入,可以如下得到整形LUT:定义常量:

[0260] minIN=minOUT=0;

[0261] maxIN=maxOUT= $2^{\text{BD}} - 1 = 1023$ 对于10比特的情况//BD=比特深度

[0262] minStdIN=64对于10比特的情况

[0263] maxStdIN=940对于10比特的情况

[0264] 步骤1:对于j=0至NumTotalBins,得到枢轴值In\_Y[j]

```
In_Y [ 0 ] = 0;
In_Y [ NumTotalBins ] = maxIN;

if ( full_range_input_flag == 0 )
{
    In_Y [ 1 ] = minStdIN;
    In_Y [ 2 ] = bin_pivot_start;

    for ( j = 3: NumTotalBins - 2 )
[0266]     In_Y [ j ] = In_Y [ j - 1 ] + equal_bin_pivot_delta;
    In_Y [ NumTotalBins - 1 ] = maxStdIN;
}
else
{
    In_Y [ 1 ] = bin_pivot_start;
    for j = 2: NumTotalBins - 1
        In_Y [ j ] = In_Y [ j - 1 ] + equal_bin_pivot_delta;
    }
[0266] 步骤2:对于j=0至NumTotalBins,得到映射值Out_Y[j]
Out_Y [ 0 ] = 0;
Out_Y [ NumTotalBins ] = maxOUT;

if ( full_range_input_flag == 0 )
{
[0267]     Out_Y [ 1 ] = 0;
    Out_Y [ 2 ] = bin_cw_start;
    Out_Y [ 3 ] = bin_cw_start + bin_cw_in_first_equal_bin;
    bin_cw [ 3 ] = bin_cw_in_first_equal_bin;
    for j = ( 4 : NumTotalBins - 2 )

        bin_cw [ j ] = bin_cw [ j - 1 ] + bin_cw_delta [ j - 4 ]; // bin_cw_delta[i]从 idx 0 开始
```

```

        for j = ( 4 : NumTotalBins - 2 )
            Out_Y [ j ] = Out_Y [ j - 1 ] + bin_cw [ j ];
        Out_Y [ NumTotalBins - 1 ] = maxOUT;
    }
    else
    {
        Out_Y [ 1 ] = bin_cw_start;
[0268] Out_Y [ 2 ] = bin_cw_start + bin_cw_in_first_equal_bin;
        bin_cw [ 2 ] = bin_cw_in_first_equal_bin;
        for j = ( 3 : NumTotalBins - 1 )

            bin_cw [ j ] = bin_cw [ j - 1 ] + bin_cw_delta [ j - 3 ]; // bin_cw_delta[i]从 idx 0 开始
        for j = 3: NumTotalBins - 1
            Out_Y [ j ] = Out_Y [ j - 1 ] + bin_cw [ j ];
        }
[0269] 步骤3:线性插值以获得所有LUT条目
[0270] 初始化:FwdLUT [ ]
        for ( j = 0: NumTotalBins - 1 )
        {
            InS   = In_Y [ j ];
            InE   = In_Y [ j + 1 ];
            OutS  = Out_Y [ j ];
            OutE  = Out_Y [ j + 1 ];

[0271]
            for i = In_Y [ j ] : In_Y [ j + 1 ] - 1
            {
                FwdLUT [ i ] = OutS + round ( ( OutE - OutS ) * ( i - InS ) / ( InE - InS ) );
            }
        }

        FwdLUT [ In_Y [ NumTotalBins ] ] = Out_Y [ NumTotalBins ];

```

[0272] 通常,可以为每个切片打开或关闭整形。例如,可以仅对帧内切片启用整形,并且对帧间切片禁用整形。在另一示例中,可以对具有最高时间级别的帧间切片禁用整形。(注意:作为示例,如本文所使用的,时间子层可以匹配HEVC中的时间子层的定义。)在定义整形器模型时,在一个示例中,可以仅在SPS中用信号传送整形器模型,但是在另一示例中,可以在帧内切片中用信号传送切片整形器模型。可替代地,可以在SPS中用信号传送整形器模型并允许切片整形器模型针对所有切片更新SPS整形器模型,或者可以仅允许切片整形器模型针对帧内切片更新SPS整形器模型。对于跟随帧内切片的帧间切片,可以应用SPS整形器

模型或帧内切片整形器模型。

[0273] 作为另一示例,图5A和5B描绘了根据实施例的在解码器中的整形函数重建过程。所述过程使用本文和参考文献[3]中描述的方法,其中视觉评级范围为[0 5]。

[0274] 如图5A所示,首先(步骤510),解码器提取reshape\_model\_profile\_type变量,并为每个区间设置适当的初始频带分布(步骤515、520和525)。例如,在伪代码中:

[0275] 如果(reshape\_model\_profile\_type==0)  $R[b_i]=R_{\text{bright}}[b_i]$ ;

[0276] 否则如果(reshape\_model\_profile\_type==1)  $R[b_i]=R_{\text{dark}}[b_i]$ ;

[0277] 否则 $R[b_i]=R_{\text{mid}}[b_i]$ 。

[0278] 在步骤530中,解码器使用所接收的reshape\_model\_band\_profile\_delta[b<sub>i</sub>]值来调整每个频带分布,如下所示:

[0279] for(i=0:reshape\_model\_num\_band-1)

[0280]          $\{R[b_i]=R[b_i]+\text{reshape\_model\_band\_profile\_delta}[b_i]\}$ 。

[0281] 在步骤535中,解码器将调整后的值传播到每个区间分布(binprofile),如下所示:如果bin[j]属于频带b<sub>i</sub>, $R_{\text{bin}}[j]=R[b_i]$ 。

[0282] 在步骤540中,修改区间分布,如下所示:

[0283] 如果(j>reshape\_model\_max\_bin\_idx)或者(j<reshape\_model\_min\_bin\_idx)

[0284] 则 $\{R_{\text{bin}}[j]=0\}$ 。

[0285] 并行地,在步骤545和550中,解码器可以提取参数以计算每个bin[j]的缩放因子值和候选码字,如下所示:

[0286] ScaleFactor=1.0-0.05\*reshape\_model\_scale\_idx

[0287] CW\_dft[j]=如果使用默认整形的情况下的区间中的码字

[0288] CW\_PQ[j]=TotalCW/TotalNumBins。

[0289] 在计算ScaleFactor值时,对于定点实施方式,可以替代地使用 $1/16=0.0625$ 而不是使用缩放因子0.05。

[0290] 继续图5B,在步骤560中,解码器基于区间分布开始对每个区间的码字(CW)预分派,如下所示:

[0291] 如果 $R_{\text{bin}}[j]==0$ , $CW[j]=0$

[0292] 如果 $R_{\text{bin}}[j]==1$ , $CW[j]=CW_{\text{dft}}[j]/2$ ;

[0293] 如果 $R_{\text{bin}}[j]==2$ , $CW[j]=\min(CW_{\text{PQ}}[j],CW_{\text{dft}}[j])$ ;

[0294] 如果 $R_{\text{bin}}[j]==3$ , $CW[j]=(CW_{\text{PQ}}[j]+CW_{\text{dft}}[j])/2$ ;

[0295] 如果 $R_{\text{bin}}[j]\geq 4$ , $CW[j]=\max(CW_{\text{PQ}}[j],CW_{\text{dft}}[j])$ ;

[0296] 在步骤565中,计算总使用的码字并细化/完成码字(CW)分派,如下所示:

[0297]  $CW_{\text{used}}=\text{Sum}(CW[j])$ ;

[0298] 如果 $CW_{\text{used}}>\text{TotalCW}$ ,重新平衡(rebalance) $CW[j]=CW[j]/(CW_{\text{used}}/\text{TotalCW})$ ;

[0299] 否则

[0300] {

[0301]      $CW_{\text{remain}}=\text{TotalCW}-CW_{\text{used}}$ ;

[0302]      $CW_{\text{remain}}$ 被分派给具有最大 $R_{\text{bin}}[j]$ 的区间;

[0303] }

[0304] 最后,在步骤565中,解码器:a)通过累加CW[j]值来生成前向整形函数(例如,FwdLUT),b)将ScaleFactor值与FwdLUT值相乘以形成最终的FwdLUT(FFwdLUT),并且c)其基于FFwdLUT来生成逆向整形函数InvLUT。

[0305] 在定点实施方式中,计算ScaleFactor和FFwdLUT可以表达为:

[0306]  $ScaleFactor = (1 \ll SF\_PREC) - reshape\_model - scale - idx$

[0307]  $FFwdLUT = (FwdLUT * ScaleFactor + (1 \ll (FP\_PREC + SF\_PREC - 1))) \gg (FP\_PREC + SF\_PREC)$ ,

[0308] 其中,SF\_PREC和FP\_PREC是预定义的精度相关的变量(例如,SF\_PREC=4,并且FP\_PREC=14)，“c=a<<n”表示对a进行二进制移位运算左移n比特(或c=a\*(2<sup>n</sup>)),并且“c=a>>n”表示对a进行二进制移位运算右移n比特(或者c=a/(2<sup>n</sup>))。

[0309] 色度QP推导

[0310] 色度编码性能与亮度编码性能密切相关。例如,在AVC和HEVC中,定义用于指定亮度分量与色度分量的量化参数(QP)之间、或明度与色度之间的关系表。规范还允许使用一个或多个色度QP偏移来更灵活地定义亮度与色度之间的QP关系。当使用整形时,亮度值被修改,因此,亮度与色度之间的关系也可以被修改。为了在整形期间维持并进一步提高编码效率,在实施例,在编码单元(CU)级,基于整形曲线得到色度QP偏移。需要在解码器和编码器两者处执行此操作。

[0311] 如本文所使用的,术语“编码单元”(CU)表示编码框(例如,宏框等)。例如,非限制性地,在HEVC中,CU被定义为“亮度样本编码框、具有三个样本阵列的图像的两个相应的色度样本编码框、或者单色图像的样本编码框或使用三个单独的颜色平面和用于编码样本的语法结构来编码的图像的样本编码框”。

[0312] 在实施例,可以如下得到色度量参数(QP)(chromaQP)值:

[0313] 1) 基于整形曲线,得到等效的亮度dQP映射,dQPLUT:

[0314] for CW=0:MAX\_CW\_VALUE-1

[0315]  $dQPLUT[CW] = -6 * \log_2(\text{slope}[CW]);$

[0316] 其中,slope[CW]表示每个CW(码字)点处的前向整形曲线的斜率,并且MAX\_CW\_VALUE是给定比特深度的最大码字值,例如,对于10比特信号,MAX\_CW\_VALUE=1024(2<sup>10</sup>)。

[0317] 然后,对于每个编码单元(CU):

[0318] 2) 计算编码单元的平均明度,表示为AvgY:

[0319] 3) 基于dQPLUT[ ]、AvgY、整形架构、逆向整形函数Inv()和切片类型来计算chromaDQP值,如下表11所示:

[0320] 表11:根据整形架构的示例chromaDQP值

整形架构	帧内切片	帧间切片
环外	dQPLUT[Inv(AvgY)]	dQPLUT[Inv(AvgY)]
环内仅帧内整形器	dQPLUT[Inv(AvgY)]	0
用于残差的环内整形器	dQPLUT[AvgY]	dQPLUT[AvgY]
混合环内整形	dQPLUT[Inv(AvgY)]	dQPLUT[AvgY]

[0321]

[0322] 4) 将chromaQP计算为:

[0323]  $\text{chromaQP} = \text{QP\_luma} + \text{chromaQPOffset} + \text{chromaDQP}$ ;

[0324] 其中,chromaQPOffset表示色度QP偏移,QP\_luma表示编码单元的亮度QP。注意,色度QP偏移的值对于每个色度分量(比如,Cb和Cr)可以是不同的,并且色度QP偏移值作为编码比特流的一部分被传送到解码器。

[0325] 在实施例中,dQPLUT[]可以被实施为预定义的LUT。假设将所有码字划分为N个区间(例如,N=32),并且每个区间包含 $M = \text{MAX\_CW\_VALUE}/N$ 个码字(例如, $M = 1024/32 = 32$ )。当为每个区间分派新的码字时,它们可以将码字的数量限制为1到 $2 * M$ ,因此它们可以预先计算dQPLUT[1... $2 * M$ ]并将计算结果保存为LUT。这种方法可以避免任何浮点计算或避免定点计算的近似。所述方法还可以节省编码/解码时间。对于每个区间,一个固定的chromaQPOffset用于该区间中的所有码字。DQP值被设置为等于dQPLUT[L],其中,L是该区间的码字数,其中, $1 \leq L \leq 2 * M$ 。

[0326] dQPLUT值可以被预先计算如下:

[0327] for  $i = 1 : 2 * M$

[0328]         $\text{slope}[i] = i/M$ ;

[0329]         $\text{dQPLUT}[i] = -6 * \log_2(\text{slope}[i])$ ;

[0330] 结束

[0331] 在计算dQPLUT[x]时,可以使用不同的量化方案来获得整数QP值,诸如:round()、ceil()、floor()或它们的混合。例如,可以设置阈值TH,并且如果 $Y < TH$ ,则使用floor()来量化dQP值,否则,当 $Y \geq TH$ 时,使用ceil()来量化dQP值。此类量化方案的使用和相应参数可以在编解码器中预定义、或者可以在比特流中用信号传送以进行适配。允许如前面所讨论的将量化方案与一个阈值混合的示例语法如下所示:

	quant_scheme_signal_table() {	描述符
	if (sps_resaper_chromaAdj > 0) {	
	<b>quant_scheme_idc</b> // 0: round(), 1: ceil(), 2: floor(), 3: 混合	u(2)
	if (quant_scheme_idc == 3) { // 混合	
[0332]	<b>quant_change_threshold</b>	u(v)
	<b>first_quant_scheme_idc</b>	u(2)
	<b>second_quant_scheme_idc</b>	u(2)
	}	
	}	
	}	

[0333] 可以根据适配需要在整形语法的不同级(例如,序列级、切片级等)定义quant\_scheme\_signal\_table()函数。

[0334] 在另一实施例中,可以通过将缩放因子应用于每个编码单元(或更具体地变换单元)中的残差信号来计算chromaDQP值。此缩放因子可以是依赖于亮度的值并且可以如下计算:a)数字地,例如,作为前向整形LUT的一阶导数(斜率)(参见例如下一节中的等式(6)),或者b)计算为:

[0335] 
$$\text{Slope}(x) = 2^{\left(\frac{\text{dQP}(x)}{6}\right)}$$

[0336] 当使用dQP(x)来计算Slope(x)时,dQP可以保持浮点精度而不进行整数量化。可替

代地,可以使用各种不同的量化方案来计算量化的整数dQP值。在一些实施例中,可以在像素级而不是在框级执行这种缩放,其中每个色度残差可以通过使用该色度样本的共同定位的亮度预测值得到的不同缩放因子来进行缩放。因此,

[0337] 表12:针对混合环内整形架构使用缩放的示例色度dQP值

更新	帧内切片	帧间切片
[0338] 基于 CU 的色度缩放 (CU 中所有样本共享的同一 $S_{cu}$ 缩放因子)	$Scu=SlopeLUT[Inv(AvgY)]$ $C\_Res\_scaled = C\_Res * S_{cu}$	$Scu=SlopeLUT[AvgY]$ $C\_Res\_scaled = C\_Res * S_{cu}$
基于像素的色度缩放 (每个样本中不同的 $S_{px}$ 缩放因子)	$Spx=SlopeLUT[Inv(ColPredY)]$ $C\_Res\_scaled = C\_Res * S_{px}$	$Spx=SlopeLUT[ColPredY]$ $C\_Res\_scaled = C\_Res * S_{px}$

[0339] 例如,如果  $CSCALE\_FP\_PREC=16$

[0340] • 前向缩放:在生成色度残差之后,在进行变换和量化之前:

[0341]  $-C\_Res=C\_orig-C\_pred$

[0342]  $-C\_Res\_scaled=C\_Res*S+(1 \ll (CSCALE\_FP\_PREC-1)) \gg CSCALE\_FP\_PREC$

[0343] • 逆向缩放:在进行色度逆量化和逆变换之后,但在重建之前:

[0344]  $-C\_Res\_inv=(C\_Res\_scaled \ll CSCALE\_FP\_PREC) / S$

[0345]  $-C\_Reco=C\_Pred+C\_Res\_inv;$

[0346] 其中,S是 $S_{cu}$ 或 $S_{px}$ 。

[0347] 注意:在表12中,在计算 $Scu$ 时,框的平均亮度(AvgY)是在应用逆向整形之前计算的。可替代地,可以在计算平均亮度之前应用逆向整形,例如, $Scu=SlopeLUT[Avg(Inv[Y])]$ 。这种替代性计算顺序也适用于计算表11中的值;即,计算 $Inv(AvgY)$ 可以替换为计算 $Avg(Inv[Y])$ 值。后一种方法可能被认为更准确,但增加了计算复杂度。

[0348] 关于整形的编码器优化

[0349] 本节讨论了当整形是标准解码过程的一部分(如三种候选架构之一所述)时通过共同优化整形参数和编码器参数来提高编码器的编码效率的多种技术。通常,编码器优化和整形在解决不同地方的编码问题时具有其自身的局限性。在传统的成像和编码系统中,存在两种类型的量化:a)基带信号中的样本量化(例如,伽马或PQ编码),以及b)变换相关的量化(压缩的一部分)。整形处于它们之间。基于图片的整形通常是基于图片来更新的,并且仅允许基于其明度级的样本值映射,而不考虑任何空间信息。在基于块的编解码器(诸如,HEVC)中,变换量化(例如,针对亮度)在空间框内被应用并且可以在空间上进行调整,因此编码器优化方法必须对包含具有不同明度值的样本的整个框应用同一参数集。如本发明人所理解并在本文描述的,联合整形和编码器优化可以进一步提高编码效率。

[0350] 帧间/帧内模式决策

[0351] 在传统编码中,帧间/帧内模式决策是基于计算原始样本与预测样本之间的失真函数( $dfiunc()$ )。此类函数的示例包括平方误差之和(SSE)、绝对差之和(SAD)等。在实施

例中,可以使用经整形的像素值来使用此类失真度量。例如,如果原始dfunct()使用Orig\_sample(i)和Pred\_sample(i),则在应用整形时,dfunct()可以使用其相应的经整形值Fwd(Orig\_sample(i))和Fwd(Pred\_sample(i))。这种方法允许更准确的帧间/帧内模式决策,从而提高编码效率。

[0352] 利用整形的LumaDQP

[0353] 在JCTVC HDR通用测试条件(CTC)文档(参考文献[6])中,lumaDQP和chromaQPoffsets是用于修改亮度分量和色度分量的量化(QP)参数以提高HDR编码效率的两个编码器设置。在本发明中,提出了几种新的编码器算法以进一步改进原始提议。对于每个lumaDQP适配单元(例如,64×64CTU),基于单元的平均输入亮度值来计算dQP值(如参考文献[6]的表3中所示)。应当通过减去该dQP来调整用于该lumaDQP适配单元内的每个编码单元的最终量化参数QP。dQP映射表在编码器输入配置中是可配置的。该输入配置被表示为 $dQP_{inp}$ 。

[0354] 如参考文献[6]和[7]中所讨论的,在现有编码方案中,同一lumaDQP LUT  $dQP_{inp}$ 用于帧内图片和帧间图片两者。帧内图片和帧间图片可能具有不同的属性和质量特性。在本发明中,提出了基于图片编码类型来调整lumaDQP设置。因此,在编码器输入配置中有两个dQP映射表是可配置的,并且被表示为 $dQP_{inpIntra}$ 和 $dQP_{inpInter}$ 。

[0355] 如前面所讨论的,当使用环内帧内整形方法时,由于不对帧间图片执行整形,因此重要的是将一些lumaDQP设置应用于帧间编码图片以实现类似的质量,就好像通过与用于帧内图片的相同整形器来对帧间图片进行整形一样。在一个实施例中,用于帧间图片的lumaDQP设置应当与用于帧内图片的整形曲线的特性相匹配。

[0356] 令

[0357]  $Slope(x) = Fwd'(x) = (Fwd(x+dx) - Fwd(x-dx)) / (2dx)$ , (6)

[0358] 表示前向整形函数的一阶导数,则在实施例中,表示自动得到的 $dQP_{auto}(x)$ 值可以计算如下:

[0359] 如果 $Slope(x) = 0$ ,则 $dQP_{auto}(x) = 0$ ,否则

[0360]  $dQP_{auto}(x) = 6 \log_2(Slope(x))$ , (7)

[0361] 其中, $dQP_{auto}(x)$ 可以被限幅在合理的范围内,例如[-6 6]。

[0362] 如果对利用整形的帧内图片启用lumaDQP(即,设置了外部 $dQP_{inpIntra}$ ),则用于帧间图片的lumaDQP应当考虑到这一点。在实施例中,可以通过将从整形器得到的 $dQP_{auto}$ (等式(7))与用于帧内图片的 $dQP_{inpIntra}$ 设置相加来计算最终帧间 $dQP_{final}$ 。在另一实施例中,为了利用帧内质量传播,可以将用于帧间图片的 $dQP_{final}$ 设置为 $dQP_{auto}$ 或仅以小的增量设置(通过设置 $dQP_{inpInter}$ )并将其添加到 $dQP_{auto}$ 。

[0363] 在实施例中,当启用整形时,以下用于设置亮度dQP值的一般规则可以适用:

[0364] (1) 可以为帧内图片和帧间图片独立设置亮度dQP映射表(基于图片编码类型);

[0365] (2) 如果编码环路内的图片处于整形域(例如,环内帧内整形架构中的帧内图片或者环外整形架构中的所有图片)中,则同样需要将输入亮度到增量QP映射 $dQP_{inp}$ 转化到整形域 $dQP_{rsp}$ 。即

[0366]  $dQP_{rsp}(x) = dQP_{inp}[Inv(x)]$ . (8)

[0367] (3) 如果编码环路内的图片处于未整形域(例如,经逆向整形的或未整形的,例如,

环内帧内整形架构中的帧间图片或者环内残差整形架构中的所有图片)中,则输入亮度到增量QP映射不需要转化并且可以直接使用。

[0368] (4) 自动帧间增量QP推导仅对环内帧内整形架构有效。在这种情况下用于帧间图片的实际增量QP是自动得到的和输入的总和值:

$$[0369] \quad dQP_{final}[x] = dQP_{inp}[x] + dQP_{auto}[x], \quad (9)$$

[0370] 并且 $dQP_{final}[x]$ 可以被限幅到合理的范围,例如 $[-12 \ 12]$ ;

[0371] (5) 可以在每个图片中更新、或者当整形LUT发生变化时更新亮度到dQP映射表。实际dQP适配(根据框的平均亮度值,获得对该框的量化的相应dQP)可以在CU级(编码器可配置)发生。

[0372] 表13总结了所提出的这三种架构中每一种的dQP设置。

[0373] 表13:dQP设置

架构	环外	环内仅帧内	环内残差
[0374] 帧内图片dQP	$dQP_{final}(x) = dQP_{rsp}(x) = dQP_{intra}[Inv(x)]$	$dQP_{final}(x) = dQP_{rsp}(x) = dQP_{intra}[Inv(x)]$	$dQP_{final}(x) = dQP_{intra}(x)$
帧间图片dQP	$dQP_{final}(x) = dQP_{rsp}(x) = dQP_{intra}[Inv(x)]$	$dQP_{final}(x) = dQP_{auto}(x) + dQP_{intra}(x)$	$dQP_{final}(x) = dQP_{intra}(x)$

[0375] 率失真优化(RDO)

[0376] 在JEM6.0软件(参考文献[8])中,当启用lumaDQP时,使用基于RDO(率失真优化)像素的加权失真。权重表是基于明度值而固定的。在实施例,应当基于如上一节中所提出的那样计算的lumaDQP设置来自适应地调整权重表。平方误差之和(SSE)及绝对差之和(SAD)的两个权重提出如下:

$$[0377] \quad weight_{SSE}[x] = 2^{\left(\frac{dQP[x]}{3}\right)}, \quad (10a)$$

$$[0378] \quad weight_{SAD}[x] = 2^{\left(\frac{dQP[x]}{6}\right)}, \quad (10b)$$

[0379] 通过等式(10a)或等式(10b)计算的权重是基于最终dQP的总权重,所述最终dQP包括输入lumaDQP和从前向整形函数得到的dQP两者。例如,基于等式(9),等式(10a)可以写为:

$$[0380] \quad weight_{SSE}[x] = 2^{\left(\frac{dQP_{final}[x]}{3}\right)} = 2^{\left(\frac{dQP_{inp}[x] + dQP_{auto}[x]}{3}\right)} \\ = 2^{\left(\frac{dQP_{inp}[x]}{3}\right)} * 2^{\left(\frac{dQP_{auto}[x]}{3}\right)}.$$

[0381] 总权重可以分离为由输入lumaDQP计算的权重:

$$[0382] \quad weight_{SSE}[x]_{inp} = 2^{\left(\frac{dQP_{inp}[x]}{3}\right)},$$

[0383] 以及来自整形的权重:

$$[0384] \quad weight_{SSE}[x]_{reshape} = 2^{\left(\frac{dQP_{auto}[x]}{3}\right)} = 2^{\left(\frac{6 \log_2(Slope(x))}{3}\right)} = (Slope(x))^2.$$

[0385] 当通过首先计算来自整形的权重而使用总dQP来计算总权重时,由于限幅操作以获得整数 $dQP_{auto}$ 而损失了精度。相反,直接使用斜率函数来计算来自整形的权重可以保持更高的权重精度,并且因此更有利。

[0386] 从输入lumaDQP得到的权重表示为 $w_{dQP}$ 。令 $f'(x)$ 表示前向整形曲线的一阶导数(或

斜率)。在实施例中,总权重同时考虑了dQP值和整形曲线的形状,因此总权重值可以表达为:

$$[0387] \quad \text{weight}_{\text{total}} = \text{Clip3}(0.0, 30.0, W_{\text{dQP}} * f'(x)^2)。 \quad (11)$$

[0388] 类似的方法也可以应用于色度分量。例如,在实施例中,对于色度,可以根据表13来定义dQP[x]。

#### [0389] 与其他编码工具的交互

[0390] 在启用整形时,本节提供了其他编码工具所需的建议更改的几个示例。对于下一代视频编码标准中要包括的任何可能的现有或未来编码工具,可能存在交互。以下给出的示例不是限制性的。通常,需要标识在编码步骤期间的视频信号域(经整形的、未整形的、经逆向整形的),并且在每个步骤处处理视频信号的操作需要考虑整形效果。

#### [0391] 跨分量线性模型预测

[0392] 在CCLM(跨分量线性模型预测)(参考文献[8])中,可以使用亮度重建信号 $\text{rec}_L'(i, j)$ 来得到预测的色度样本 $\text{pred}_c(i, j)$ :

$$[0393] \quad \text{pred}_c(i, j) = \alpha \cdot \text{rec}_L'(i, j) + \beta。 \quad (12)$$

[0394] 在启用整形时,在实施例中,可能需要判别亮度重建信号是处于整形域(例如,环外整形器或环内帧内整形器)中还是处于未整形域(例如,环内残差整形器)中。在一个实施例中,可以按原样隐式地使用重建亮度信号而无需任何附加信令或操作。在其他实施例中,如果重建信号处于未整形域中,则可以将重建亮度信号转化成也处于未整形域中,如下所示:

$$[0395] \quad \text{pred}_c(i, j) = \alpha \cdot \text{Inv}(\text{rec}_L'(i, j)) + \beta。 \quad (13)$$

[0396] 在其他实施例中,可以添加比特流语法元素以发信号通知哪个域是期望的(经整形的或未整形的),这可以由RDO过程决定,或者可以基于解码的信息得到所述决策,从而节省显式信令所需的开销。可以基于所述决策来对重建信号执行相应的操作。

#### [0397] 具有残差预测工具的整形器

[0398] 在HEVC范围扩展配置文件中,包括残差预测工具。根据编码器侧的亮度残差信号将色度残差信号预测为:

$$[0399] \quad \Delta r_c(x, y) = r_c(r, y) - (\alpha \times r'_L(x, y)) \gg 3, \quad (14)$$

[0400] 并且色度残差信号在解码器侧被补偿为:

$$[0401] \quad r'_c(x, y) = \Delta r'_c(x, y) + (\alpha \times r'_L(x, y)) \gg 3, \quad (15)$$

[0402] 其中, $r_c$ 表示位置(x,y)处的色度残差样本, $r'_L$ 表示亮度分量的重建残差样本, $\Delta r_c$ 表示使用色间预测的预测信号, $\Delta r'_c$ 表示在对 $\Delta r_c$ 进行编码和解码之后的重建信号,并且 $r_c$ 表示重建的色度残差。

[0403] 在启用整形时,可能需要考虑使用哪个亮度残差来进行色度残差预测。在一个实施例中,可以按原样使用“残差”(可以是基于整形器架构经整形的或未整形的)。在另一实施例中,可以使亮度残差强制处于一个域中(诸如在未整形域中)并执行适当的映射。在另一实施例中,可以由解码器得到适当的处理,或者可以如前所述明确地发信号通知适当的处理。

#### [0404] 具有自适应限幅的整形器

[0405] 自适应限幅(参考文献[8])是一种新工具,其被引入用于发信号通知关于内容动

态的原始数据范围,并在压缩工作流程中(例如,在变换/量化、环路滤波、输出中)发生限幅的每个步骤处进行自适应限幅而不是固定限幅(基于内部比特深度信息)。令

$$[0406] \quad T_{\text{clip}} = \text{Clip}_{\text{BD}}(T, \text{bitdepth}, C) = \text{Clip3}(\min_c, \max_c, T), \quad (16)$$

[0407] 其中,  $x = \text{Clip3}(\min, \max, c)$  表示:

$$[0408] \quad x = \begin{cases} \min, & \text{如果 } c \leq \min \\ \max, & \text{如果 } c \geq \max, \\ c, & \text{否则} \end{cases}$$

[0409] 并且

[0410] • C是分量ID(通常为Y、Cb或Cr)

[0411] •  $\min_c$ 是分量ID C的当前切片中使用的限幅下限

[0412] •  $\max_c$ 是分量ID C的当前切片中使用的限幅上限

[0413] 在启用整形时,在实施例,可能需要找出数据流当前所处的域并正确地执行限幅。例如,如果在整形域数据中处理限幅,则需要将原始限幅边界转化到整形域:

$$[0414] \quad T_{\text{clip}} = \text{Clip}_{\text{BD}}(T, \text{bitdepth}, c) =$$

$$[0415] \quad = \text{Clip3}(\text{Fwd}(\min_c), \text{Fwd}(\max_c), T)。 \quad (17)$$

[0416] 通常,需要关于整形架构正确地处理每个限幅步骤。

[0417] 整形器和环路滤波

[0418] 在HEVC和JEM 6.0软件中,诸如ALF和SAO等环路滤波器需要使用重建的亮度样本和未压缩的“原始”亮度样本来估计最佳滤波器参数。在启用整形时,在实施例,可以(明确地或隐式地)指定想要执行滤波器优化的域。在一个实施例,可以估计整形域上的滤波器参数(当重建处于整形域中时,相对于经整形的原件)。在其他实施例,可以估计未整形域上的滤波器参数(当重建处于未整形域中或经逆向整形的域中时,相对于原件)。

[0419] 例如,取决于环内整形架构,可以通过表14和表15描述环内滤波器优化(ILFOPT)选项和操作。

[0420] 表14. 在环内仅帧内整形架构和环内混合整形中的环路滤波优化

	帧内	帧间
[0421] 选项 1: 在整形域中估计并执行环路滤波(LF);	<ul style="list-style-type: none"> <li>• <b>编码器侧:</b> <ul style="list-style-type: none"> <li>○ 使用已经整形的原始帧内图像作为 LF 参考</li> <li>○ 使用重建的帧内图像(在整形域中)和 LF 参考来估计 LF 参数; 取决于 LF 模框相对于逆向整形模框(图 2C 中的框 265 和 270)的位置, 存在两种情况:</li> </ul> </li> </ul>	<ul style="list-style-type: none"> <li>• <b>编码器侧:</b> <ul style="list-style-type: none"> <li>○ 对原始帧间图像进行前向整形作为 LF 参考</li> <li>○ 对重建的帧间图像进行前向整形</li> <li>○ 估计 LF 参数</li> <li>○ 将 LF 应用于整形域中的重建帧间图像</li> </ul> </li> </ul>

[0422]

	<ul style="list-style-type: none"> <li>▪ 如果在 LF 之前执行逆向整形, 则需要对重建的帧内图像应用前向整形</li> <li>▪ 如果要在 LF 之后执行逆向整形, 则直接使用重建的帧内图像             <ul style="list-style-type: none"> <li>○ 将 LF 应用于整形域中的重建帧内图像</li> <li>○ 对整个重建的帧内图像进行逆向整形</li> <li>○ 存储在 DPB 中</li> </ul> </li> <li>• <b>解码器侧:</b> <ul style="list-style-type: none"> <li>○ 将 LF 应用于整形域中的重建帧内图像; 取决于 LF 模框相对于逆向整形模框 (图 2D 中的框 265 和 270) 的位置, 存在两种情况:                 <ul style="list-style-type: none"> <li>▪ 如果在 LF 之前执行逆向整形, 则需要应用 LF 之前对重建的帧内图像进行前向整形</li> <li>▪ 如果要在 LF 之后执行逆向整形, 则直接对重建的帧内图像应用 LF</li> </ul> </li> <li>○ 对整个重建的帧内图像进行逆向整形</li> <li>○ 存储在 DPB 中</li> </ul> </li> </ul>	<ul style="list-style-type: none"> <li>○ 对整个重建的帧间图像进行逆向整形</li> <li>○ 存储在 DPB 中</li> <li>• <b>解码器侧:</b> <ul style="list-style-type: none"> <li>○ 对重建的帧间图像进行前向整形</li> <li>○ 将 LF 应用于整形域中的重建帧间图像</li> <li>○ 对整个重建的帧间图像进行逆向整形</li> <li>○ 存储在 DPB 中</li> </ul> </li> </ul>
<p>选项 2: 在未整形域中估计并执行 LF</p>	<ul style="list-style-type: none"> <li>• <b>编码器侧:</b> <ul style="list-style-type: none"> <li>○ 如果已经在原始图像缓冲器上执行了就地整形, 则对原始图像进行逆向整形作为 LF 参考; 或者取出未整形的原始图像作为 LF 参考</li> <li>○ 对重建的帧内图像进行逆向整形</li> <li>○ 估计 LF 参数</li> <li>○ 对整个重建的图像进行逆向整形</li> <li>○ 将 LF 应用于重建的帧内图像</li> <li>○ 存储在 DPB 中</li> </ul> </li> </ul>	<p>(精确的传统 LF 工作流程)</p> <ul style="list-style-type: none"> <li>• <b>编码器侧:</b> <ul style="list-style-type: none"> <li>○ 使用重建的帧间图像和原始图像作为 LF 参考来估计 LF 参数</li> <li>○ 将 LF 应用于重建的帧间图像</li> <li>○ 存储在 DPB 中</li> </ul> </li> <li>• <b>解码器侧:</b> <ul style="list-style-type: none"> <li>○ 将 LF 应用于重建的帧间图像</li> <li>○ 存储在 DPB 中</li> </ul> </li> </ul>

[0423]	<ul style="list-style-type: none"> <li>• <b>解码器侧:</b> <ul style="list-style-type: none"> <li>○ 对整个重建的图像进行逆向整形</li> <li>○ 将 LF 应用于重建的帧内图像</li> <li>○ 存储在 DPB 中</li> </ul> </li> </ul>	
--------	--	--

[0424] 表15. 环内残差整形架构中的环路滤波优化

[0425]	环内残差整形	帧内和帧间 (与表 14 中的 <b>帧间</b> 操作相同)
--------	--------	------------------------------------

[0426] 虽然本文的大多数详细讨论涉及对明度分量执行的方法,但是本领域技术人员将理解,可以对色度颜色分量和色度相关参数(诸如chromaQPoffset(例如,参见参考文献[9]))执行类似的方法。

[0427] 环内整形和感兴趣区域(ROI)

[0428] 在给定了图像的情况下,如本文所使用的,术语‘感兴趣区域’(ROI)表示被认为是特别感兴趣的图像区域。在本节中,提出了仅支持对感兴趣区域的环内整形的新颖实施例。即,在实施例中,可以仅在ROI内部而不是在外部应用整形。在另一实施例中,可以在感兴趣区域中和感兴趣区域之外应用不同的整形曲线。

[0429] ROI的使用受到需要使比特率与图像质量平衡的推动。例如,考虑日落的视频序列。在图像的上半部分,可以使太阳位于相对均匀的颜色天空(因此天空背景中的像素可以具有非常低的方差)上。相反,图像的下半部分可以描绘移动波。从观看者的角度来看,可能认为上部远比下部重要。另一方面,由于移动波的像素的方差较大,因此移动波难以压缩,每个像素需要更多的比特;然而,可能想要在太阳部分上分配比波部分更多的比特。在这种情况下,上半部分可以表示为感兴趣区域。

[0430] ROI描述

[0431] 如今,大多数编解码器(例如,AVC、HEVC等)是基于块的。为了简化实施方式,可以以框为单位指定区域。非限制地,使用HEVC作为示例,区域可以被定义为多个编码单元(CU)或编码树单元(CTU)。可以指定一个ROI或多个ROI。多个ROI可以是不同的或重叠的。ROI不一定是矩形的。可以在任何感兴趣的级(诸如切片级、图片级、视频流级等)提供用于ROI的语法。在实施例中,首先在序列参数集(SPS)中指定ROI。然后在切片报头中,可以允许很小的ROI变化。表16描绘了其中一个ROI被指定为矩形区域中的多个CTU的语法的示例。表17描述了在切片级的修改的ROI的语法。

[0432] 表16:用于ROI的SPS语法

	SPS() {	描述符
	...	
	<b>sps_reshaper_enable_flag</b>	u(1)
	if (sps_reshaper_enable_flag) {	
	.....	
	<b>sps_reshaper_active_ROI_flag</b>	u(1)
[0433]	if (sps_reshaper_active_ROI_flag) {	
	<b>reshaper_active_ROI_in_CTUsize_left</b>	ue(v)
	<b>reshaper_active_ROI_in_CTUsize_right</b>	ue(v)
	<b>reshaper_active_ROI_in_CTUsize_top</b>	ue(v)
	<b>reshaper_active_ROI_in_CTUsize_bottom</b>	ue(v)
	}	
	}	
	....	
	}	

[0434] 表17:用于ROI的切片报头语法

	reshaping_sliceheader_table() {	描述符
	...	
	if (sps_reshaper_active_ROI_flag) {	
	<b>reshape_model_ROI_modification_flag</b>	u(1)
	if (reshape_model_ROI_modification_flag) {	
[0435]	<b>reshaper_ROI_mod_offset_left</b>	se(v)
	<b>reshaper_ROI_mod_offset_right</b>	se(v)
	<b>reshaper_ROI_mod_offset_top</b>	se(v)
	<b>reshaper_ROI_mod_offset_bottom</b>	se(v)
	}	
	}	
	....	
	}	

[0436] sps\_reshaper\_active\_ROI\_flag等于1,指定编码视频序列(CVS)中存在ROI。sps\_reshaper\_active\_ROI\_flag等于0,指定CVS中不存在ROI。

[0437] reshaper\_active\_ROI\_in\_CTUsize\_left、reshaper\_active\_ROI\_in\_CTUsize\_right、reshaper\_active\_ROI\_in\_CTUsize\_top和reshaper\_active\_ROI\_in\_CTUsize\_bottom各自根据图片坐标中指定的矩形区域来指定ROI中的图片样本。对于左侧和上部,坐标等于offset\*CTUsize,而对于右侧和下部,坐标等于offset\*CTUsize-1。

[0438] reshape\_model\_ROI\_modification\_flag等于1,指定在当前切片中修改了ROI。reshape\_model\_ROI\_modification\_flag等于0,指定在当前切片中未修改ROI。

[0439] reshaper\_ROI\_mod\_offset\_left、reshaper\_ROI\_mod\_offset\_right、reshaper\_ROI\_mod\_offset\_top和reshaper\_ROI\_mod\_offset\_bottom各自指定与reshaper\_active\_ROI\_in\_CTUsize\_left、reshaper\_active\_ROI\_in\_CTUsize\_right、reshaper\_active\_ROI\_in\_CTUsize\_top和reshaper\_active\_ROI\_in\_CTUsize\_bottom的左/右/上/下偏移值。

[0440] 对于多个ROI,表16和表17的针对单个ROI的示例语法可以使用针对每个ROI的索引(或ID)进行扩展,类似于HEVC中用于使用SEI消息定义多个全扫描(pan-scan)矩形的方案(参见HEVC规范,参考文献[11],D.2.4节)。

[0441] 环内仅帧内整形中的ROI处理

[0442] 对于仅帧内整形,首先对图片的ROI部分进行整形,然后对其应用编码。由于整形仅应用于ROI,因此可能会看到图片的ROI部分与非ROI部分之间的边界。由于环路滤波器(例如,图2C或图2D中的270)可能跨越边界,因此必须特别注意ROI来进行环路滤波器优化(ILFOPT)。在实施例中,提出仅在整个解码图片处于相同域中的情况下才应用该环路滤波器。即,整个图片要么全部处于整形域中要么全部处于未整形域中。在一个实施例中,在解码器侧,如果在未整形域上应用环路滤波,则应当首先对经解码图片的ROI部分应用逆向整形,并且然后应用环路滤波器。接下来,将经解码图片存储到DPB中。在另一实施例中,如果在整形域上应用环路滤波器,则应当首先对经解码图片的非ROI部分应用整形,并且然后应用环路滤波器,并且然后对整个图片进行逆向整形。接下来,将经解码图片存储到DPB中。在又一个实施例中,如果在整形域上应用环路滤波,则可以首先对经解码图片的ROI部分进行逆向整形,然后对整个图片进行整形,然后应用环路滤波器,然后对整个图片进行逆向整形。接下来,将经解码图片存储到DPB中。表18中总结了这三种方法。从计算的角度来看,方法“A”更简单。在实施例中,ROI的启用可以用于指定执行逆向整形与环路滤波(LF)的顺序。例如,如果主动使用ROI(例如,SPS语法标志=真),则在逆向整形(图2C和图2D中的框265)之后执行LF(图2C和图2D中的框270)。如果没有主动使用ROI,则在逆向整形之前执行LF。

## [0443] 表18. 使用ROI的环路滤波(LF)选项

[0444]	方法 A	方法 B	方法 C
[0445]	<ul style="list-style-type: none"> <li>• 对经解码图片的 ROI 进行逆向整形</li> <li>• 对整个图片应用环路滤波 (LF)</li> <li>• 存储在 DPB 中</li> </ul>	<ul style="list-style-type: none"> <li>• 对经解码图片的非 ROI 部分应用整形</li> <li>• 对整个图片应用 LF</li> <li>• 对整个图片进行逆向整形</li> <li>• 存储在 DPB 中</li> </ul>	<ul style="list-style-type: none"> <li>• 对经解码图片的 ROI 进行逆向整形</li> <li>• 对整个图片进行整形</li> <li>• 对整个图片应用 LF</li> <li>• 对整个图片进行逆向整形</li> <li>• 存储在 DPB 中</li> </ul>

[0446] 环内预测残差整形中的ROI处理

[0447] 对于环内(预测)残差整形架构(例如,参见图2F中的200C\_D),在解码器处,使用等式(3),处理可以表达为:

[0448] 如果(当前CTU属于ROI)

[0449]  $Reco\_sample = Inv(Res\_d + Fwd(Pred\_sample))$ , (参见等式(3))

[0450] 否则

[0451]  $Reco\_sample = Res\_d + Pred\_sample$

[0452] 结束

[0453] ROI和编码器考虑因素

[0454] 在编码器中,需要检查每个CTU是否属于ROI。例如,对于环内预测残差整形,基于等式(3)的简单检查可以执行以下内容:

[0455] 如果(当前CTU属于ROI)

[0456] 对亮度应用RDO中的加权失真。权重基于等式(10)得到

[0457] 否则

- [0458] 对亮度应用RDO中的未加权失真
- [0459] 结束
- [0460] 在整形期间考虑ROI的示例编码 workflow 可以包括以下步骤:
- [0461] -对于帧内图片:
- [0462] -对原始图片的ROI区应用前向整形
- [0463] -对帧内帧进行编码
- [0464] -在环路滤波器 (LF) 之前对重建图片的ROI区应用逆向整形
- [0465] -在未整形域中如下执行环路滤波 (例如, 参见表18中的方法“C”), 其包括以下步骤:
- [0466] • 对原始图片的非ROI区应用前向整形 (以便使整个原始图片被整形用于环路滤波器参考)
- [0467] • 对重建图片的整个图片区应用前向整形
- [0468] • 得到环路滤波器参数并应用环路滤波
- [0469] • 对重建图片的整个图片区应用逆向整形并将其存储在DPB中
- [0470] 在编码器侧, 由于LF需要具有用于滤波器参数估计的非压缩参考图像, 因此每种方法的LF参考的处理如表19所示:
- [0471] 表19. 关于ROI的LF参考的处理

	方法 A	方法 B	方法 C
[0472]	<ul style="list-style-type: none"> <li>将未整形的原始输入图片用于 LF 参考;</li> </ul>	<ul style="list-style-type: none"> <li>对整个原始输入图片 (非 ROI 部分和 ROI 部分两者) 应用整形, 以获得 LF 参考;</li> </ul>	<ul style="list-style-type: none"> <li>对整个原始输入图片 (非 ROI 部分和 ROI 部分两者) 应用整形, 以获得 LF 参考;</li> </ul>

- [0473] -对于帧间图片:
- [0474] -当对帧间帧进行编码时, 对于ROI内的每个CU, 对亮度应用预测残差整形和加权失真; 对于ROI之外的每个CU, 不应用任何整形
- [0475] -与以前一样 (就好像没有使用ROI一样) 执行环路滤波优化 (选项1):
- [0476] • 对原始图片的整个图片区进行前向整形
- [0477] • 对重建图片的整个图片区进行前向整形
- [0478] • 得到环路滤波器参数并应用环路滤波
- [0479] • 对重建图片的整个图片区应用逆向整形并将其存储在DPB中
- [0480] 对HLG编码内容的整形
- [0481] 术语HybridLog-Gamma或HLG表示在Rec. BT. 2100中定义的用于映射高动态范围信号的另一个传递函数。HLG的开发是为了维持与使用传统伽马函数编码的传统标准动态范围信号的后向兼容性。当比较PQ编码内容与HLG编码内容之间的码字分布时, PQ映射倾向于在暗区和亮区中分配更多码字, 而大多数HLG内容码字显现为被分配到中间范围中。可以使用两种方法来进行HLG亮度整形。在一个实施例中, 可以简单地将HLG内容转换为PQ内容, 并且然后应用前面讨论的所有PQ相关的整形技术。例如, 可以应用以下步骤:
- [0482] 1) 将HLG明度 (例如, Y) 映射到PQ明度。令变换的函数或LUT表示为HLG2PQLUT (Y)

[0483] 2) 分析PQ明度值并得到基于PQ的前向整形函数或LUT。将其表示为PQAdpFLUT(Y)

[0484] 3) 将这两个函数或LUT合并为单个函数或LUT:HLGAdpFLUT[i]=PQAdpFLUT

[0485] [HLG2PQLUT[i]]。

[0486] 由于HLG码字分布与PQ码字分布完全不同,因此这种方法可能产生次优的整形结果。在另一实施例中,HLG整形函数是直接来自HLG样本得到的。可以应用与用于PQ信号的框架相同的框架,但是改变CW\_Bins\_Dft表以反映HLG信号的特性。在实施例中,使用用于HLG信号的中间调分布,能够根据用户偏好来设计几个CW\_Bins\_Dft表。例如,当优选保留高光时,对于alpha=1.4,

[0487] g\_DftHLGCWBin0={8,14,17,19,21,23,24,26,27,28,29,30,31,32,33,34,35,36,36,37,38,39,39,40,41,41,42,43,43,44,44,30}。

[0488] 当优选保留中间调(或中间范围)时:

[0489] g\_DftHLGCWBin1={12,16,16,20,24,28,32,32,32,32,36,36,40,44,48,52,56,52,48,44,40,36,36,32,32,32,26,26,20,16,16,12}。

[0490] 当优选保留皮肤色调时:

[0491] g\_DftHLGCWBin2={12,16,16,24,28,32,56,64,64,64,64,56,48,40,32,32,32,32,32,32,28,28,24,24,20,20,20,20,20,16,16,12};

[0492] 从比特流语法的角度来看,为了区分基于PQ的整形与基于HLG的整形,添加了表示为sps\_reshaper\_signal\_type的新参数,其中,值sps\_reshaper\_signal\_type表明被整形的信号类型(例如,0为基于伽马的SDR信号,1为PQ编码信号,并且2为HLG编码信号)。

[0493] 在表20和表21中示出了针对PQ和HLG两者SPS和切片报头中的用于HDR整形的语法表的示例,所述HDR整形具有前面讨论的所有特征(例如,ROI、环路滤波器优化(ILFOPT)和ChromaDQPAdjustment)。

[0494] 表20:用于整形的示例SPS语法

SPS()	描述符
.....	
<b>sps_reshaper_enable_flag</b> /*1: 打开整形, 否则关闭*/	u(1)
if (sps_reshaper_enable_flag) {	
<b>sps_reshaper_adaptive_flag</b> /* 1: 打开自适应整形, 否则关闭*/	u(1)
<b>sps_reshaper_signal_type</b> /*例如: 0: SDR, 1:PQ, 2: HLG*/	u(2)
<b>sps_in_loop_filter_opt_flag</b> /* ILFOPT 标志*/	u(1)
<b>sps_luma_based_chroma_qp_offset_flag</b> /*chromaDQPAdjustment 标志*/	u(1)
<b>sps_reshaper_active_ROI_flag</b>	u(1)
if (sps_reshaper_active_ROI_flag) {	
<b>reshaper_active_ROI_in_CTUsize_left</b>	ue(v)
<b>reshaper_active_ROI_in_CTUsize_right</b>	ue(v)
<b>reshaper_active_ROI_in_CTUsize_top</b>	ue(v)
<b>reshaper_active_ROI_in_CTUsize_bottom</b>	ue(v)
}	
}	

[0495] sps\_in\_loop\_filter\_opt\_flag等于1指定要在编码视频序列(CVS)中的整形域中执行的环内滤波器优化。

[0497] sps\_in\_loop\_filter\_opt\_flag等于0指定要在CVS中的未整形域中执行的环内滤

波器优化。sps\_luma\_based\_chroma\_qp\_offset\_flag等于1指定得到(例如,根据表11或表12)基于亮度的色度QP偏移并且将其应用于编码视频序列(CVS)中的每个CU的色度编码。sps\_luma\_based\_chroma\_qp\_offset\_flag等于0指定在CVS中未启用基于亮度的色度QP偏移。

[0498] 表21:用于在切片级进行整形的示例语法

	描述符
reshaping_sliceheader_table_model() {	
<b>reshape_model_profile_type</b>	ue(v)
<b>reshape_model_scale_idx</b>	u(2)
<b>reshape_model_min_bin_idx</b>	u(5)
<b>reshape_model_max_bin_idx</b>	u(5)
<b>reshape_model_num_band</b>	u(4)
for ( i = 0; i < reshape_model_num_band; i++ ) {	
<b>reshape_model_band_profile_delta [ i ]</b>	u(1)
}	
[0499] if (sps_resaper_active_ROI_flag) {	
<b>reshape_model_ROI_modification_flag</b>	u(1)
if (reshape_model_ROI_modification_flag) {	
<b>resaper_ROI_mod_offset_left</b>	se(v)
<b>resaper_ROI_mod_offset_right</b>	se(v)
<b>resaper_ROI_mod_offset_top</b>	se(v)
<b>resaper_ROI_mod_offset_bottom</b>	se(v)
}	
}	

[0500] 提高色度质量

[0501] 基于HLG的编码的支持者认为基于HLG的编码提供了与SDR信令更好的后向兼容性。因此,理论上,基于HLG的信号可以采用与传统SDR信号相同的编码设置。但是,当在HDR模式下查看HLG编码信号时,仍然可以观察到一些颜色伪像(artifact),尤其是在非彩色(achromatic)区域中(诸如白色和灰色)。在实施例中,可以通过在编码期间调整chromaQPoffset值来减少此类伪像。建议对于HLG内容,应用比在编码PQ信号时使用的较不强烈的chromaQP调整。例如,在参考文献[10]中,基于亮度QP来为Cb和Cr分派QP偏移、并基于捕获颜色原色和表示颜色原色来分派因子的模型被描述为:

$$[0502] \quad QPoffsetCb = Clip3(-12, 0, Round(c_{cb} * (k * QP + 1))), \quad (18a)$$

$$[0503] \quad QPoffsetCr = Clip3(-12, 0, Round(c_{cr} * (k * QP + 1))), \quad (18b)$$

[0504] 其中,如果捕获颜色原色与表示颜色原色相同,则 $c_{cb} = 1$ ,如果捕获颜色原色等于P3D65原色并且表示颜色原色等于Rec. ITU-R BT.2020原色,则 $c_{cb} = 1.04$ ,并且如果捕获颜色原色等于Rec. ITU-R BT.709原色并且表示原色等于Rec. ITU-R BT.2020原色,则 $c_{cb} = 1.14$ 。类似地,如果捕获颜色原色与表示颜色原色相同,则 $c_{cr} = 1$ ,如果捕获颜色原色等于P3D65原色并且表示颜色原色等于Rec. ITU-R BT.2020原色,则 $c_{cr} = 1.39$ ,并且如果捕获颜色原色等于Rec. ITU-R BT.709原色并且表示原色等于Rec. ITU-R BT.2020原色,则 $c_{cr} = 1.78$ 。最后, $k = -0.46$ 并且 $l = 0.26$ 。

[0505] 在实施例中,建议使用相同的模型但具有不同的参数,以产生较不强烈的chromaQPoffset变化。例如,非限制性地,在实施例中,对于等式(18a)中的Cb,  $c_{cb} = 1$ ,  $k = -$

0.2且 $l=7$ ,并且对于等式(18b)中的 $C_r, c_{c_r}=1, k=-0.2$ ,且 $l=7$ 。图6A和图6B描绘了chromaQPoffset值如何根据针对PQ(Rec.709)和HLG的亮度量化参数(QP)而改变的示例。PQ相关值的变化比HLG相关值更显著。图6A对应于Cb(等式(18a)),而图6B对应于Cr(等式(18b))。

[0506] 参考文献

[0507] 本文列出的参考文献中的每一篇都通过引用以其全文被并入。

[0508] [1]于2016年3月30日提交的In-Loop Block-Based Image Reshaping in High Dynamic Range Video Coding(高动态范围视频编码中的基于块的环内图像整形)的PCT申请PCT/US2016/025082,也由G-M.Su公布为WO 2016/164235。

[0509] [2]D.Baylon、Z.Gu、A.Luthra、K.Minoo、P.Yin、F.Pu、T.Lu、T.Chen、W.Husak、Y.He、L.Kerofsky、Y.Ye、B.Yi的“Response to Call for Evidence for HDR and WCG Video Coding:Arris,Dolby and InterDigital(对HDR视频编码和WCG视频编码的证据支持的响应:Arris、Dolby和InterDigital)”,文档m36264,2015年7月,波兰华沙。

[0510] [3]于2017年1月19日由T.Lu等人提交的Content-Adaptive Reshaping for High Codeword representation Images(用于高码字表示图像的内容自适应整形)的美国专利申请15/410,563。

[0511] [4]于2016年7月14日提交的Signal Reshaping and Coding for HDR and Wide Color Gamut Signals(用于HDR和宽色域信号的信号整形和编码)的PCT申请PCT/US2016/042229,也由P.Yin等人公布为WO 2017/011636。

[0512] [5]K.Minoo等人的“Exploratory Test Model for HDR extension of HEVC(用于HEVC的HDR扩展的探索性测试模型)”,MPEG输出文档,JCTVC-W0092(m37732),2016年,美国圣地亚哥。

[0513] [6]E.Francois、J.Sole、J.Ström、P.Yin的“Common Test Conditions for HDR/WCG video coding experiments(用于HDR/WCG视频编码实验的通用测试条件)”,JCTVC文档Z1020,日内瓦,2017年1月。

[0514] [7]A.Segall、E.Francois和D.Rusanovskyy的“JVET common test conditions and evaluation procedures for HDR/WCG Video(用于HDR/WCG视频的JVET通用测试条件和评估程序)”,JVET-E1020,ITU-T会议,日内瓦,2017年1月。

[0515] [8]JEM 6.0软件:<https://jvet.hhi.fraunhofer.de/svn/svnHMJEMSoftware/tags/HM-16.6-JEM-6.0>

[0516] [9]T.Lu等人的于2016年10月11日提交的“Adaptive Chroma Quantization in Video Coding for Multiple Color Imaging Formats(用于多种颜色成像格式的视频编码中的自适应色度量化)”的美国临时专利申请序列号62/406,483,也作为美国专利申请序列号15/728,939提交,公布为美国专利申请公开US2018/0103253。

[0517] [10]J.Samuelsson等人(编辑)的“Conversion and coding practices for HDR/WCG Y'CbCr 4:2:0Video with PQ Transfer Characteristics(用于具有PQ传输特性的HDR/WCG Y'CbCr 4:2:0视频的转化和编码实践)”JCTVC-Y1017,ITU-T/ISO会议,成都,2016年10月。

[0518] [11]ITU-T H.265,“High efficiency video coding(高效视频编码)”,ITU,4.0

版, (12/2016)。

[0519] 示例计算机系统实施方式

[0520] 本发明的实施例可以利用计算机系统、以电子电路和部件来配置的系统、整合的电路 (IC) 设备 (诸如微控制器、现场可编程门阵列 (FPGA)、或其他可配置或可编程逻辑器件 (PLD)、离散时间或数字信号处理器 (DSP)、专用 IC (ASIC))、和/或包括一个或多个这样的系统、设备或部件的装置来实施。计算机和/或 IC 可以执行、控制或实行与整合的信号整形和图像编码相关的指令, 诸如本文所述的那些指令。计算机和/或 IC 可以计算与本文描述的信号整形和编码过程相关的各种参数或值中的任何参数或值。图像和视频实施例可以以硬件、软件、固件及其各种组合来实施。

[0521] 本发明的某些实施方式包括执行软件指令的计算机处理器, 所述软件指令使处理器执行本发明的方法。例如, 显示器、编码器、机顶盒、转码器等中的一个或多个处理器可以通过执行所述处理器可访问的程序存储器中的软件指令来实施与如上所述的整合的信号整形和图像编码相关的方法。还可以以程序产品的形式提供本发明。程序产品可以包括携带一组计算机可读信号的任何非暂态介质, 所述一组计算机可读信号包括指令, 所述指令当由数据处理器执行时使数据处理器执行本发明的方法。根据本发明的程序产品可以采用各种形式中的任何一种。程序产品可以包括例如物理介质, 诸如包括软盘、硬盘驱动器的磁性数据存储介质, 包括 CD ROM、DVD 的光学数据存储介质, 包括 ROM、闪存 RAM 的电子数据存储介质等。程序产品上的计算机可读信号可以可选地被压缩或加密。

[0522] 在上面提到部件 (例如, 软件模框、处理器、组件、设备、电路等) 的情况下, 除非另有说明, 否则对此部件的引用 (包括对“装置”的引用) 都应被解释为包括为此部件的执行所描述部件的功能的任何部件的等效物 (例如, 功能上等同的), 包括在结构上不等同于执行在本发明的所展示示例实施例中的功能的所公开结构的部件。

[0523] 等效物、扩展、替代品及其他

[0524] 如此描述了与高效的整合的信号整形和图像编码相关的示例实施例。在前述说明书中, 已经参考许多具体细节描述了本发明的实施例, 这些细节可以根据实施方式而变化。因此, 作为本发明的、并且申请人意图作为本发明的唯一且独有的指示是从本申请中以这套权利要求发布的具体形式发布的权利要求, 包括任何后续修改。本文中明确阐述的针对这样的权利要求中包含的术语的任何定义应当支配着权利要求中使用的这样的术语的含义。因此, 权利要求中未明确引用的限制、要素、特性、特征、优点或属性不应以任何方式限制这样的权利要求的范围。因此, 说明书和附图应被视为说明性的而非具有限制性意义。

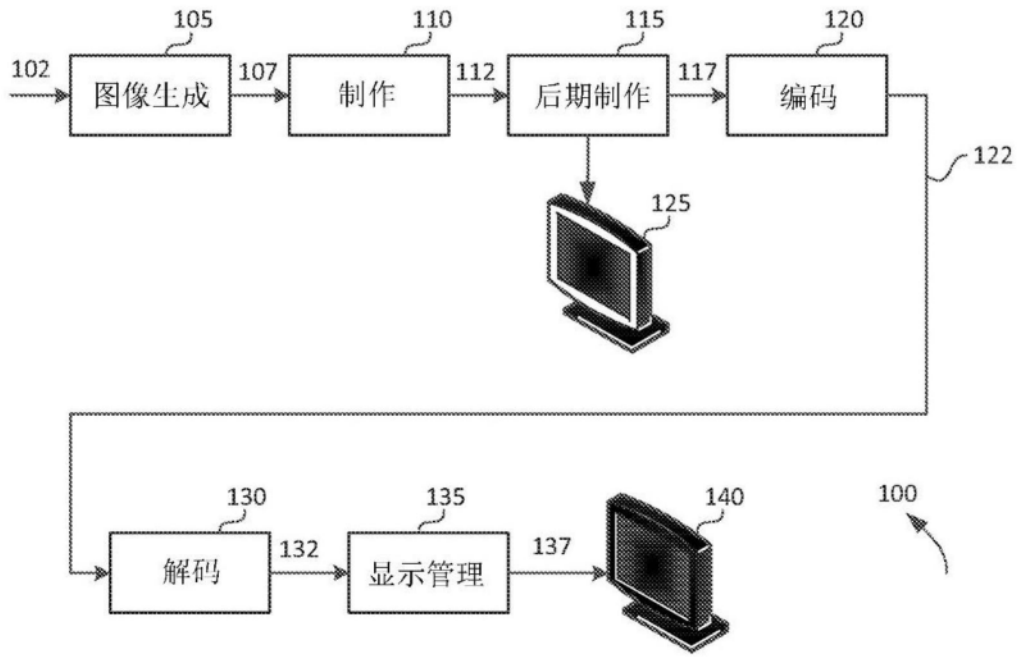
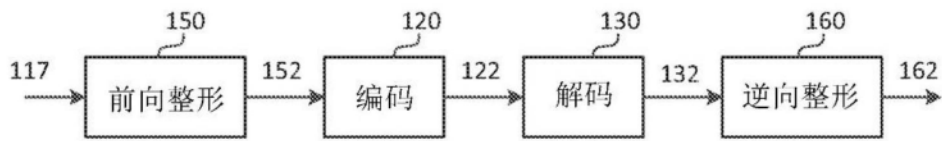


图1A



现有技术

图1B

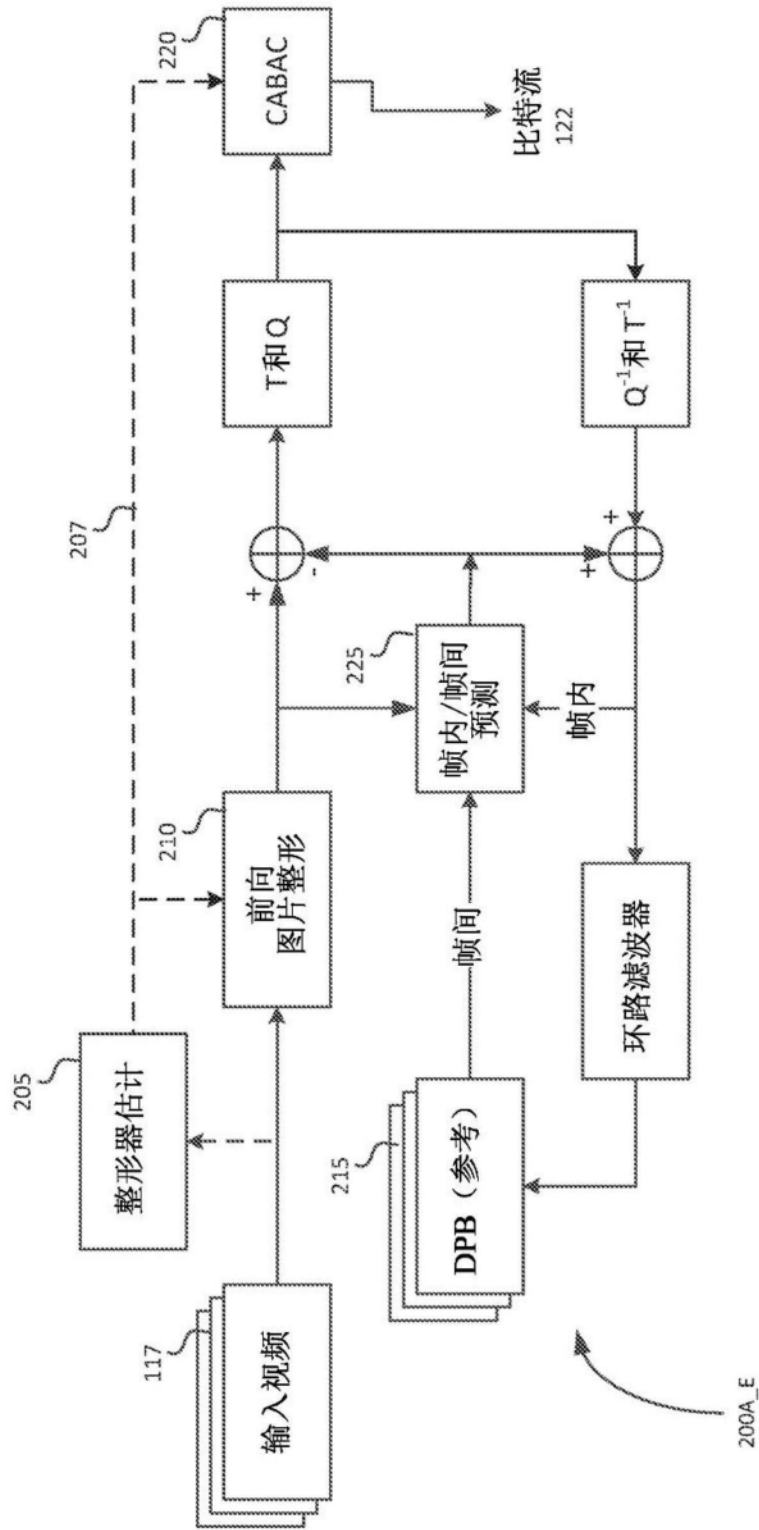


图2A

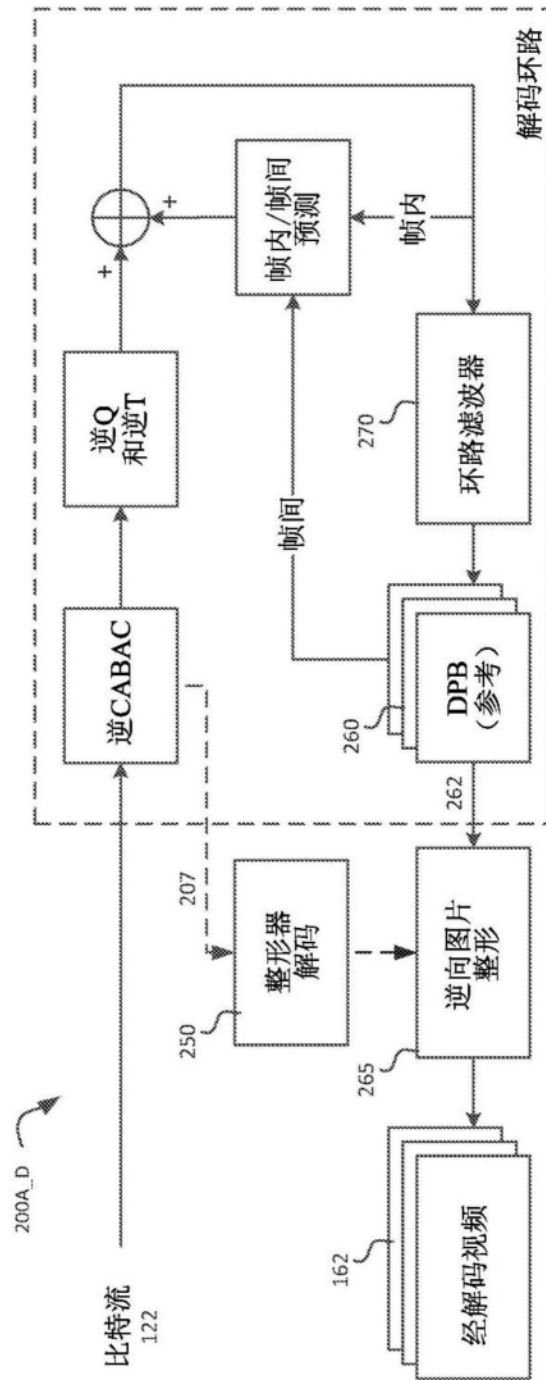


图2B

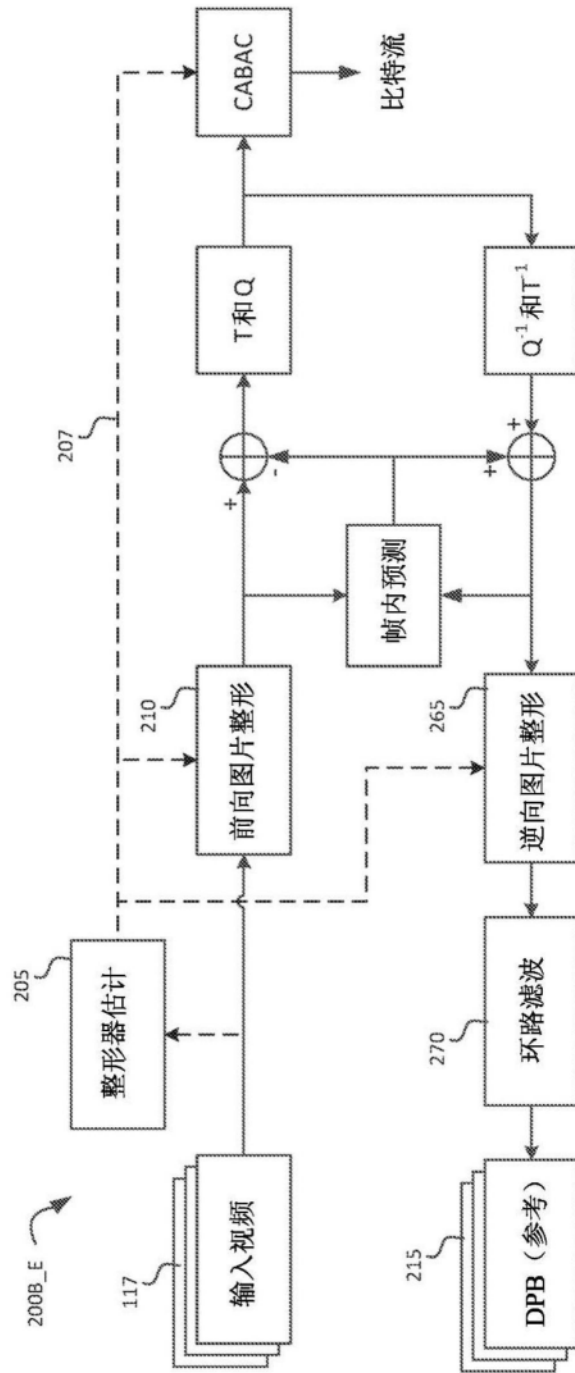


图2C



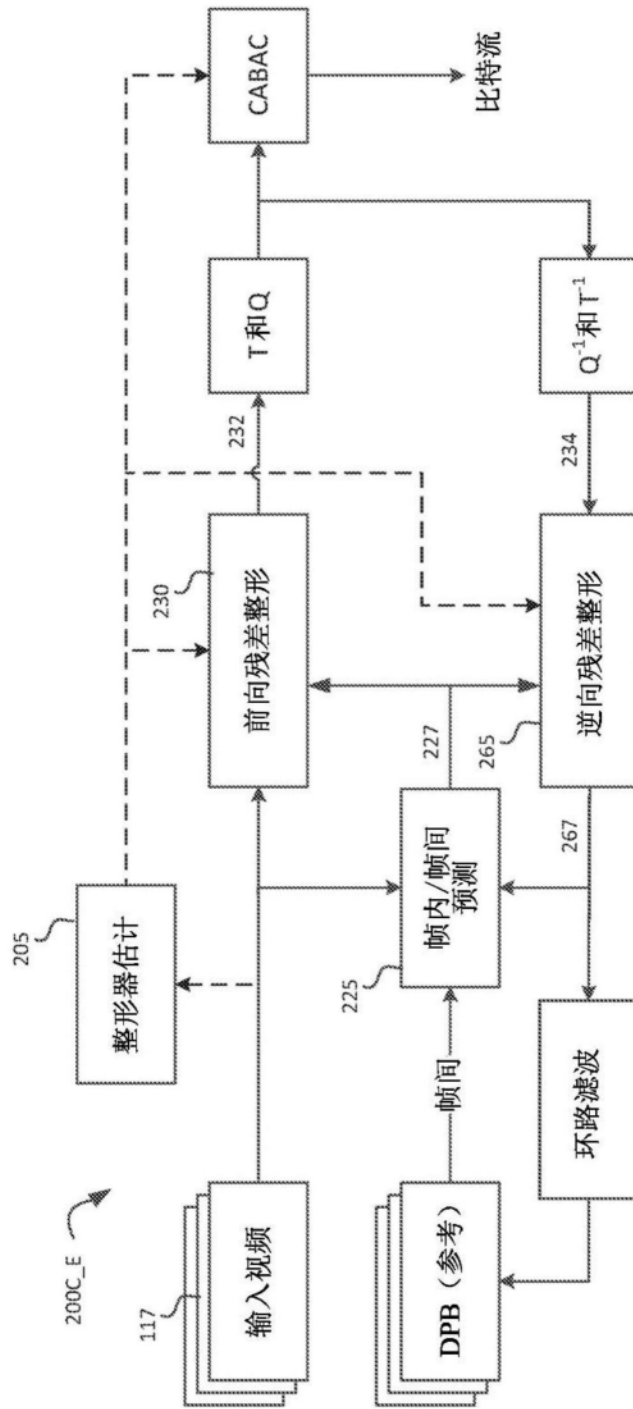


图2E

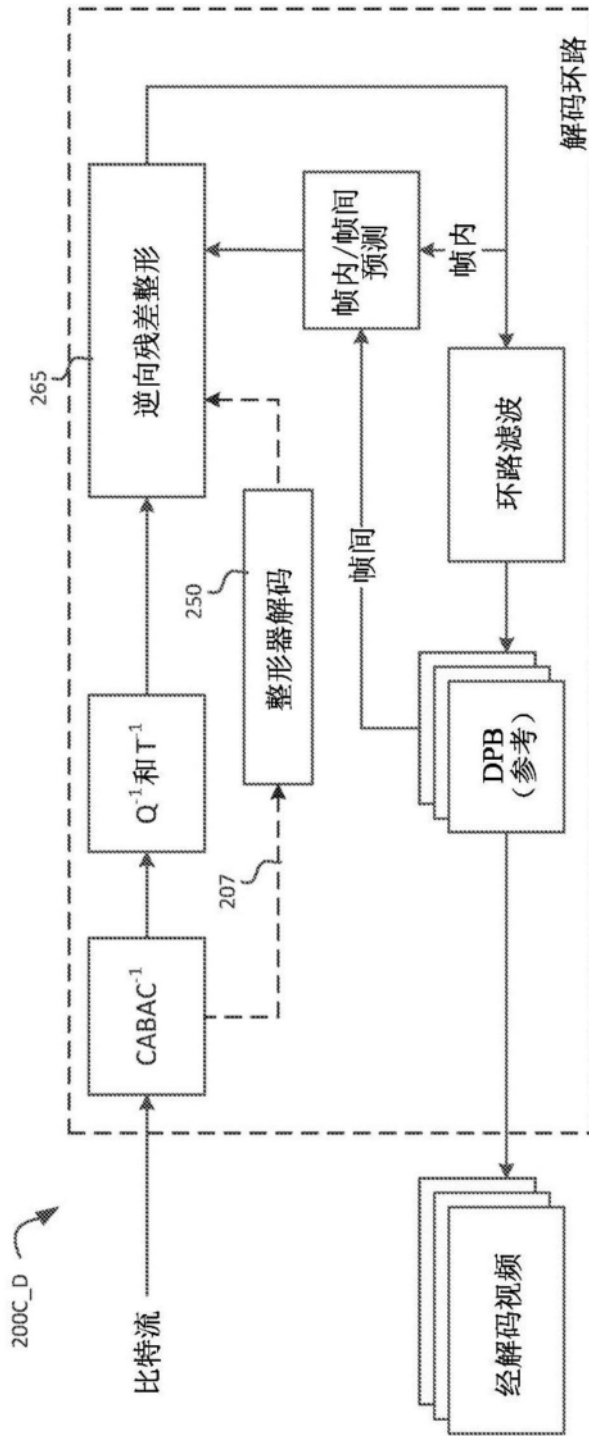


图2F



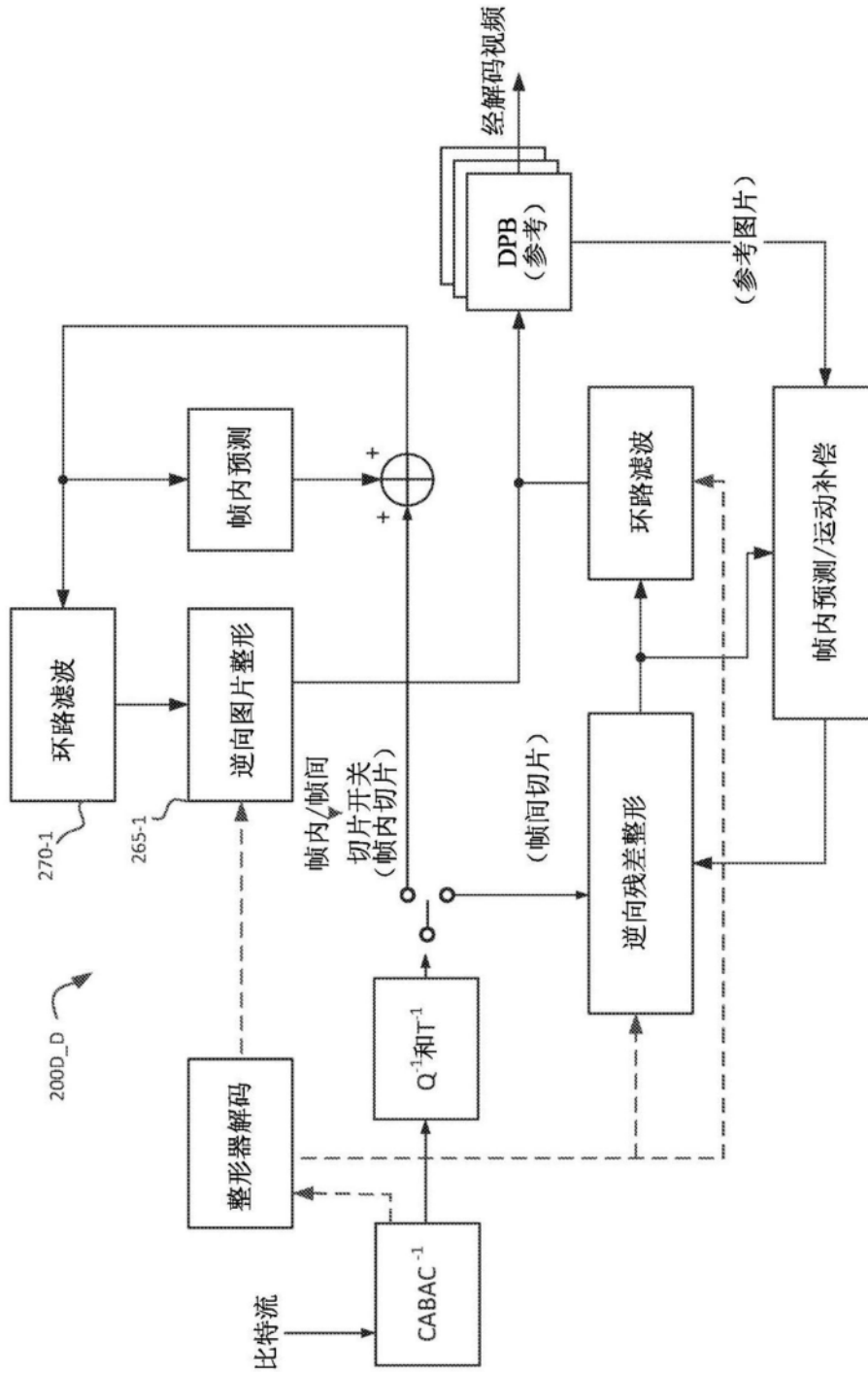


图2H

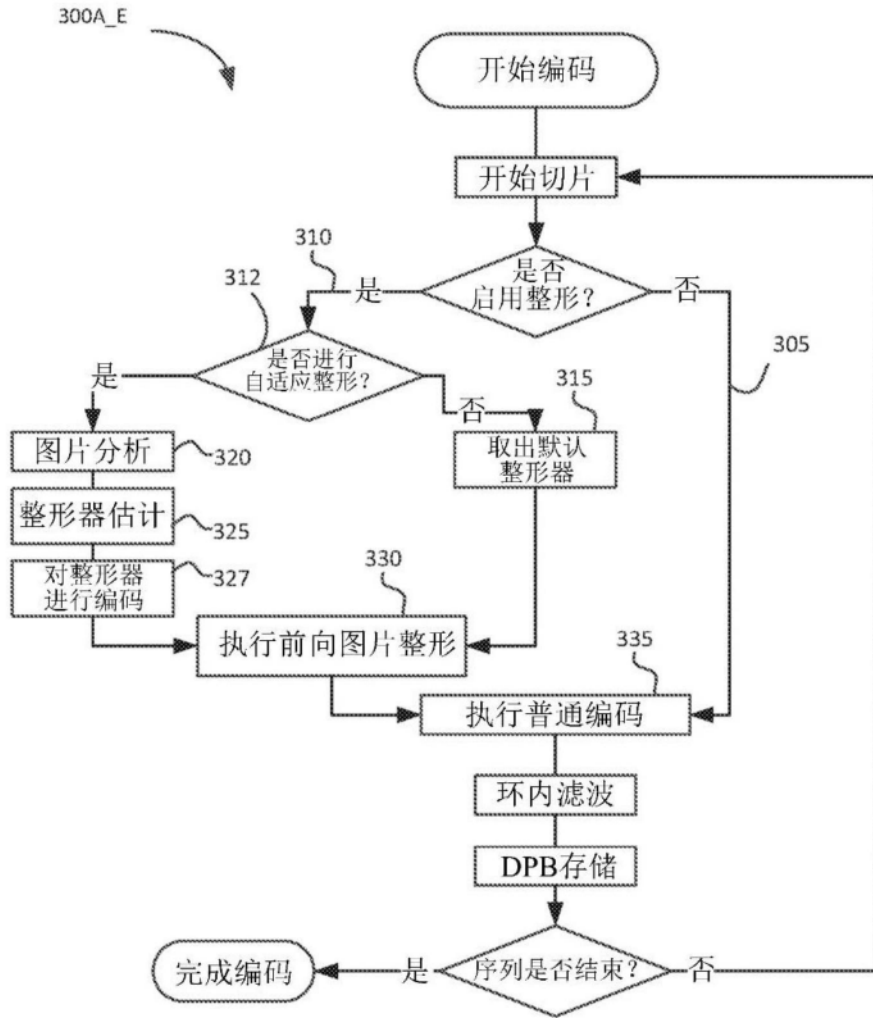


图3A

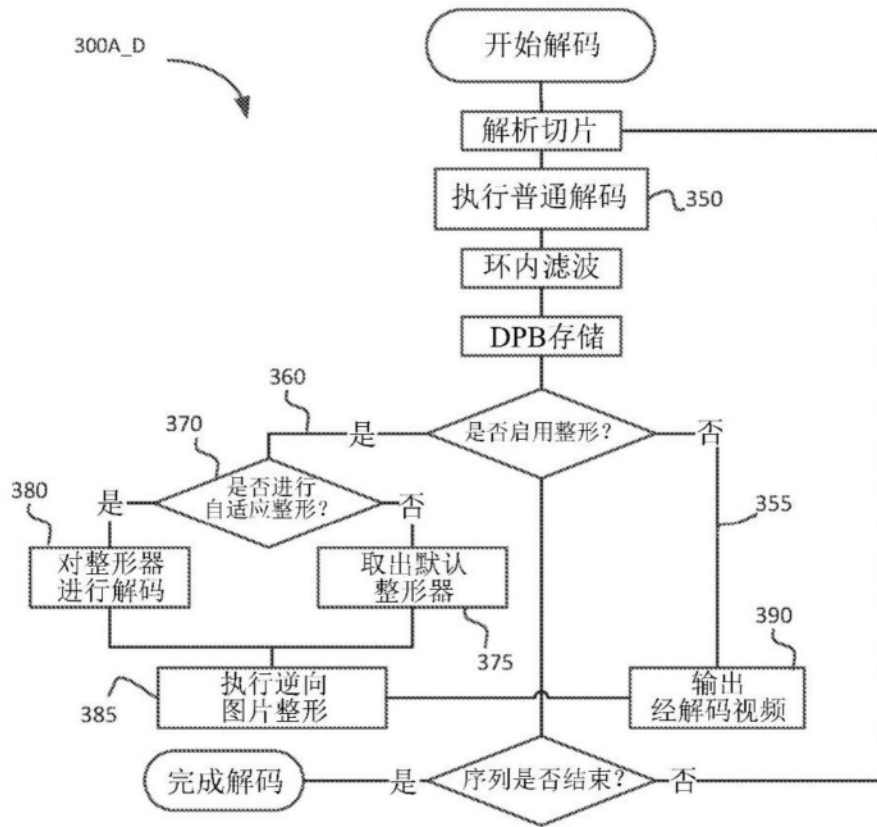


图3B

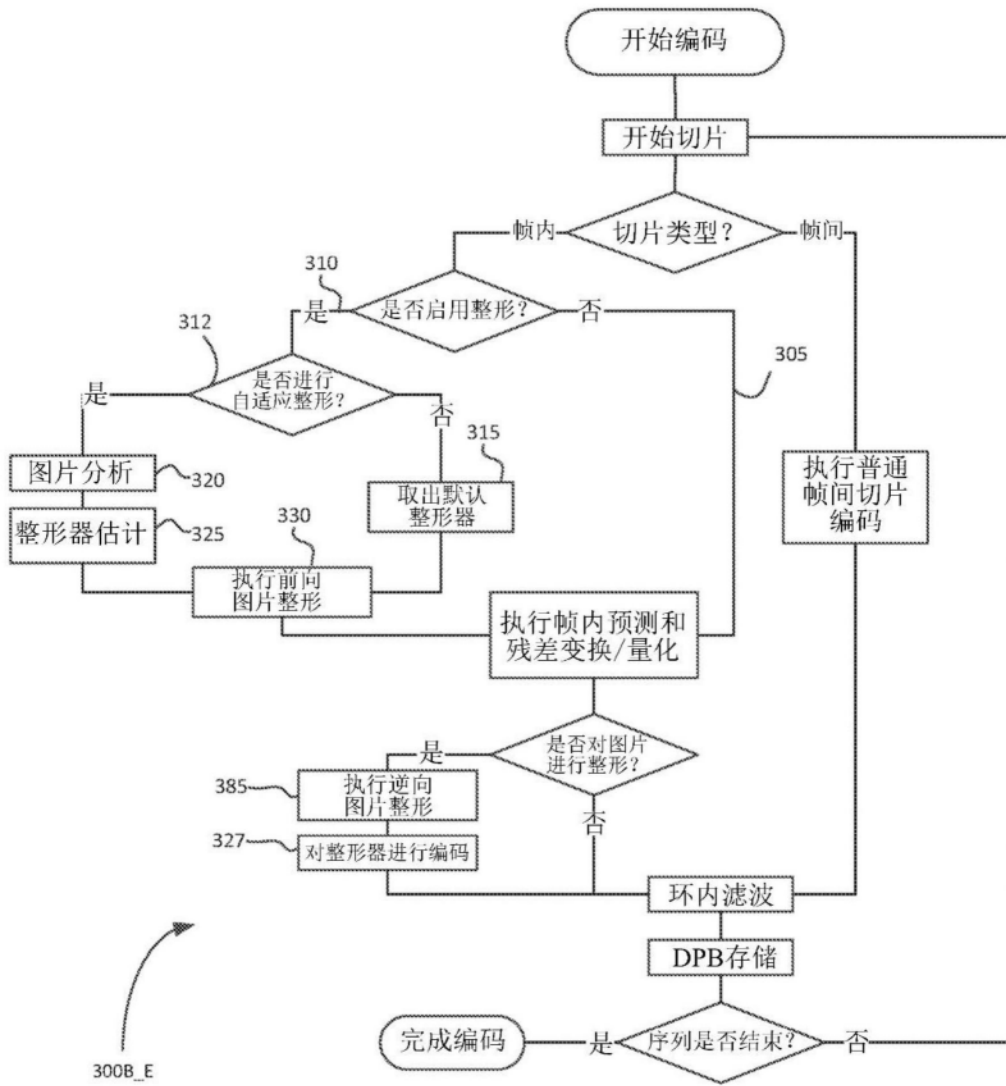


图3C

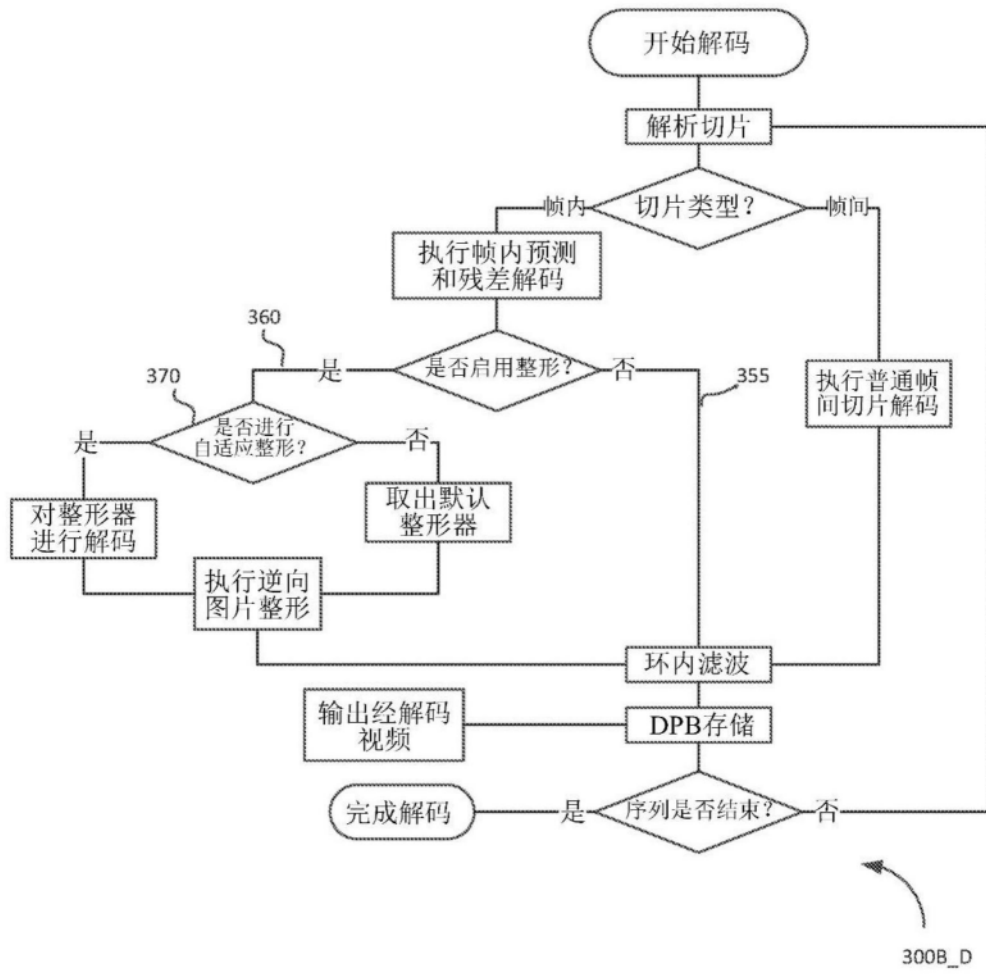


图3D

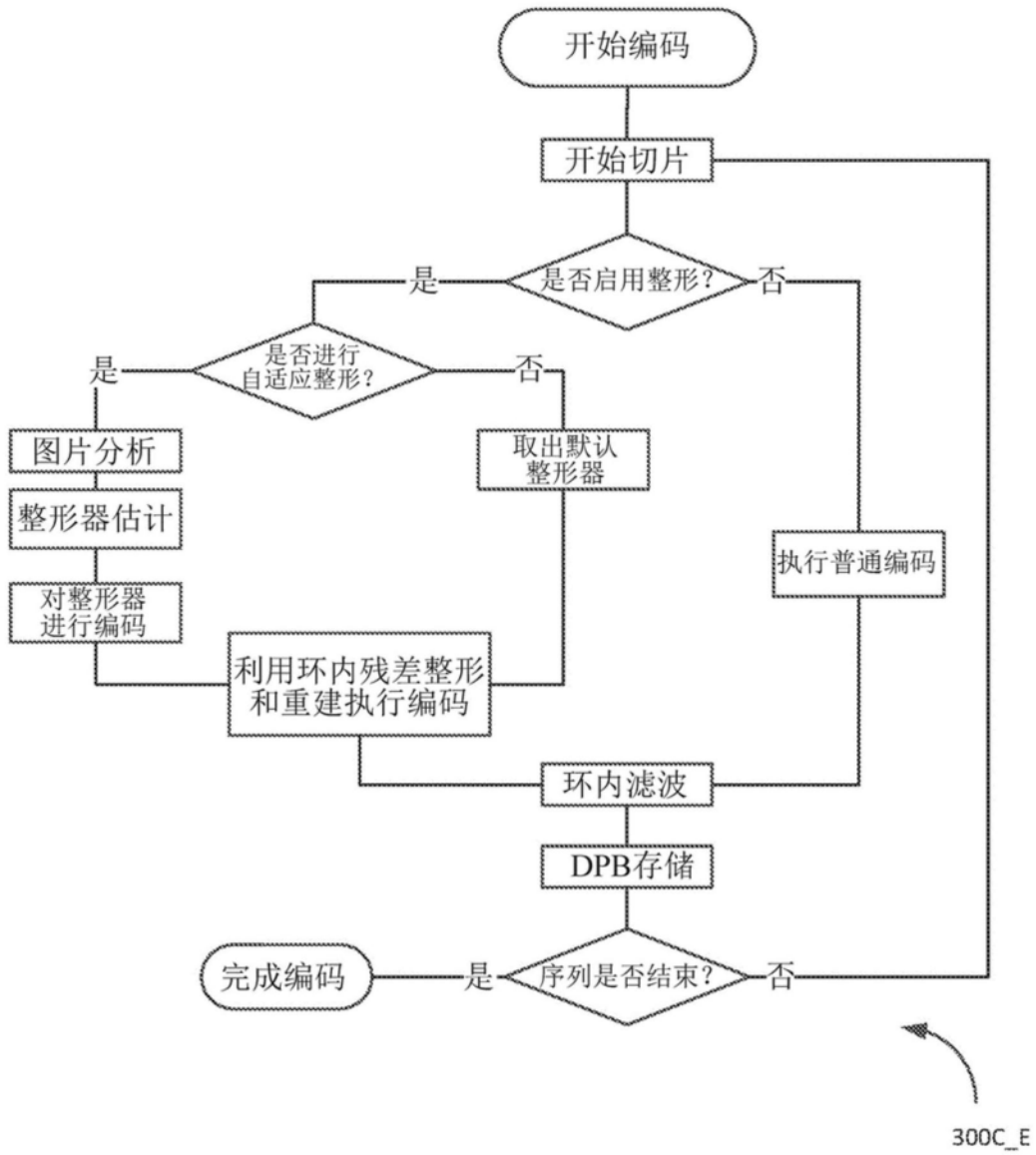


图3E

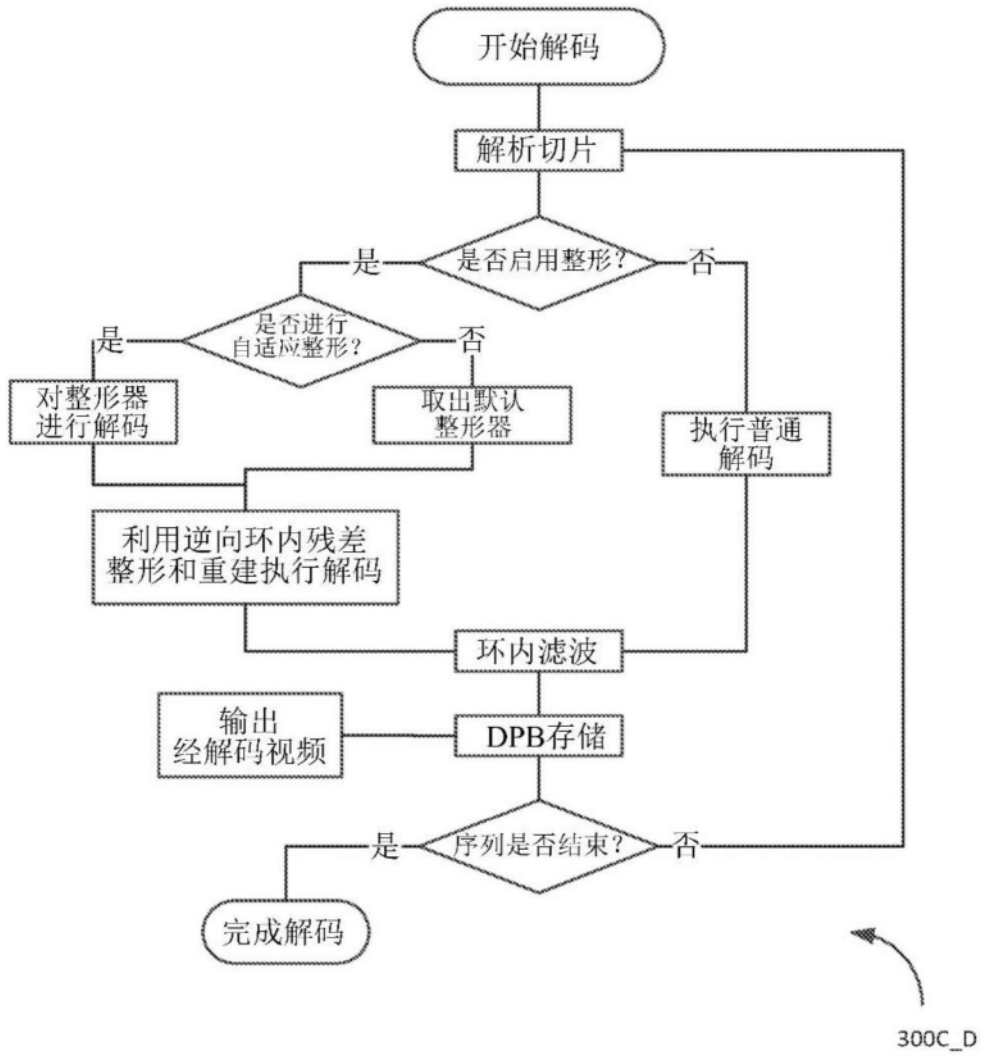


图3F

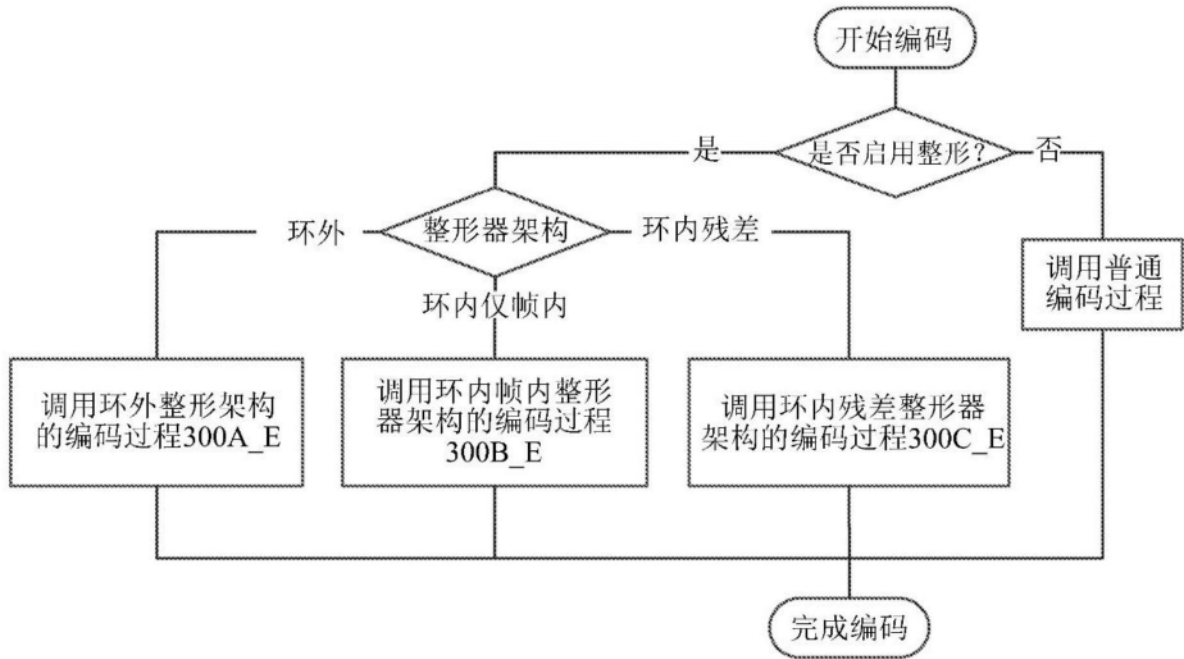


图4A

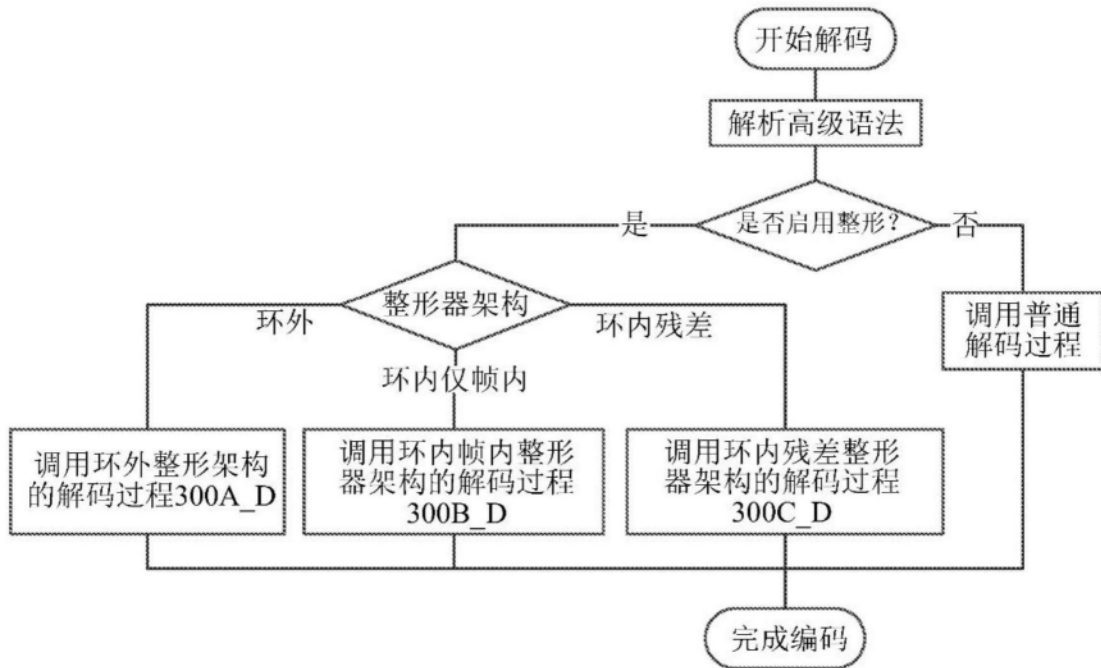


图4B

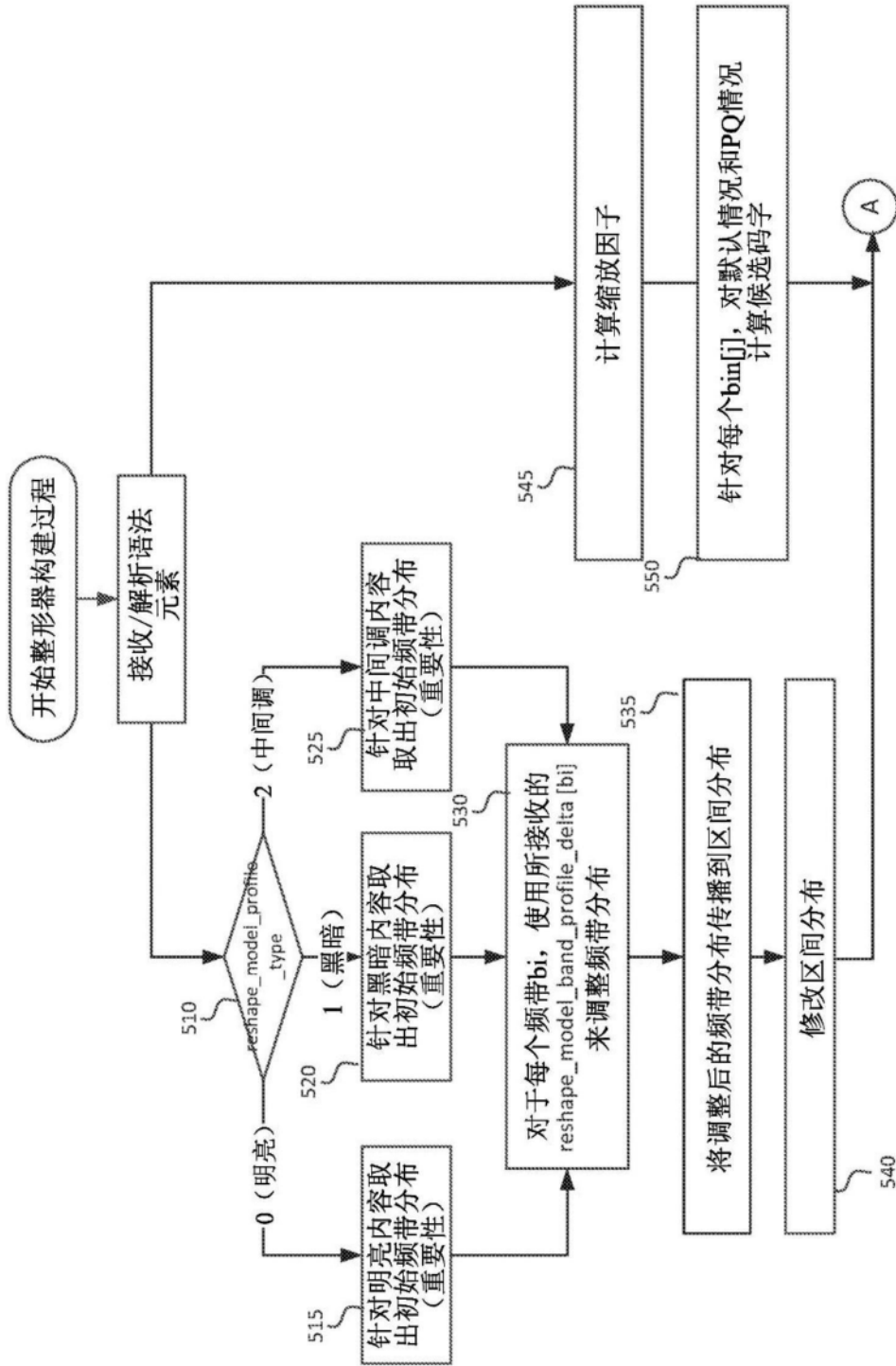


图5A

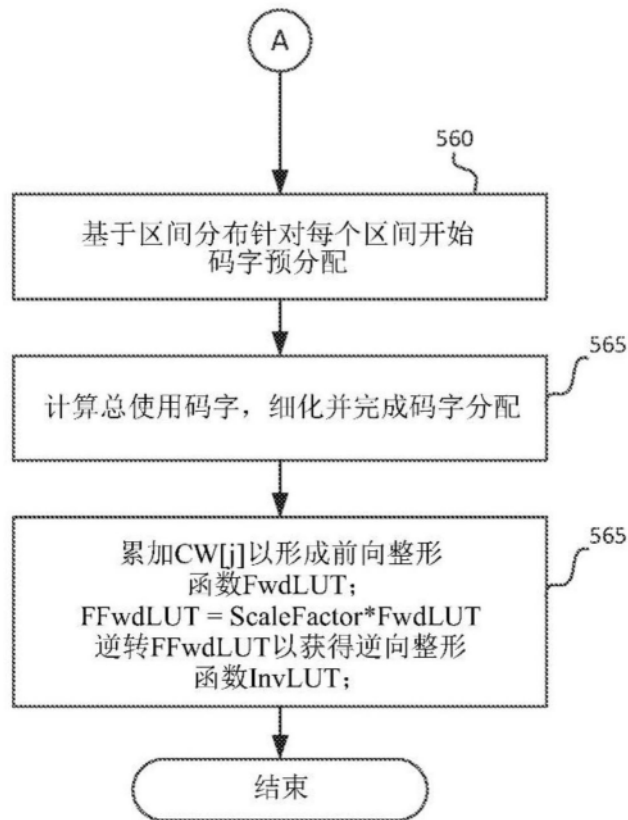


图5B

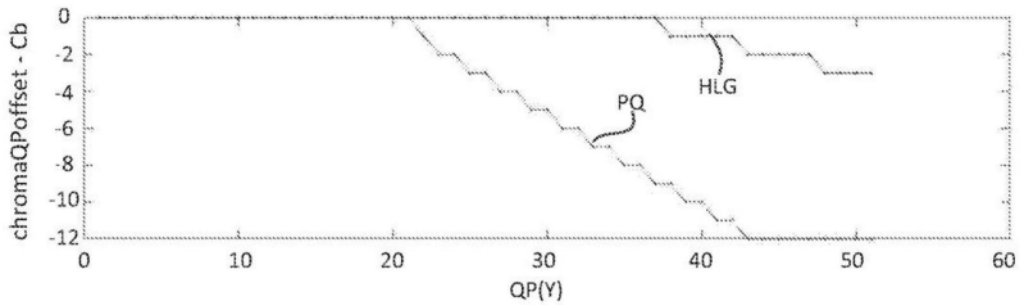


图6A

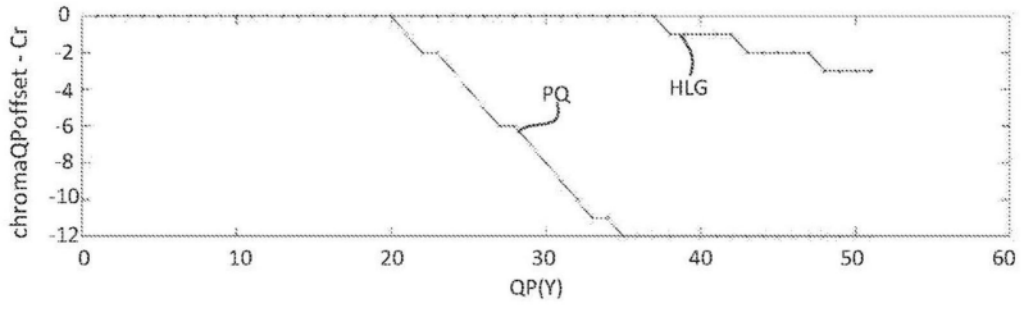


图6B

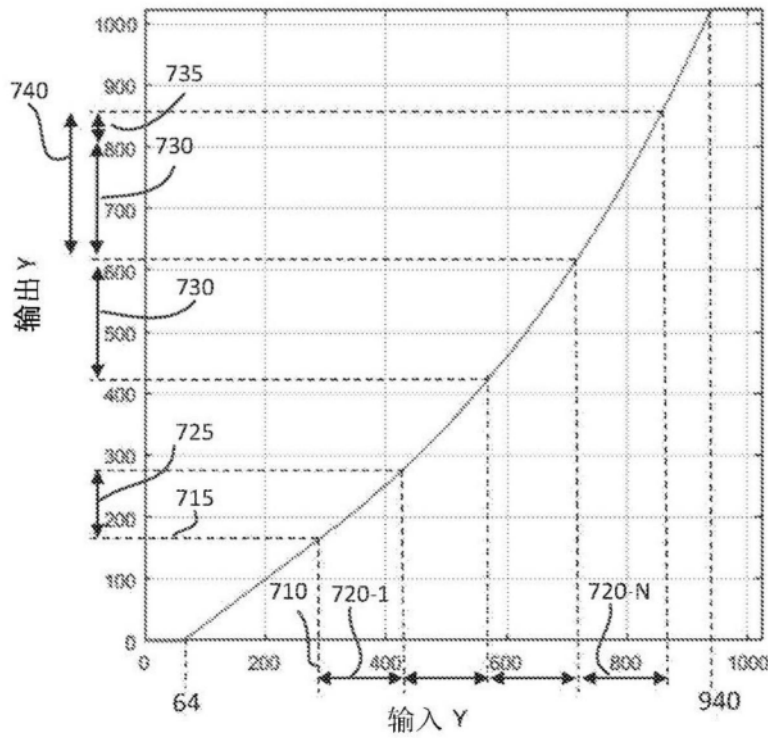


图7