

系数编解码方法、编码器、解码器以及计算机存储介质

技术领域

本申请实施例涉及视频编解码技术领域，尤其涉及一种系数编解码方法、编码器、解码器以及计算机存储介质。

5 背景技术

随着人们对视频显示质量要求的提高，计算机视觉相关领域得到了越来越多的关注。近年来，图像处理技术在各行各业都取得了成功地应用。对于视频图像的编解码过程而言，在编码端，针对待编码的图像数据，在经过变换和量化处理之后将通过熵编码单元进行压缩编码，而熵编码处理后所生成的码流将传输到解码端；然后解析码流，再经过反量化和逆变换处理之后，可以恢复出原始输入的图像数据。

10 目前，高位宽、高质量、高码率的视频编解码（简称为“三高视频”）相较于较低位宽、低质量、低码率的视频编解码（可称为“常规视频”），通常需要编解码的系数更多，而且系数更大。这样，对于三高视频来说，已有的相关方案可能会带来更大的开销，产生浪费，甚至影响编解码的速度和吞吐量。

发明内容

15 本申请实施例提供一种系数编解码方法、编码器、解码器以及计算机存储介质，在高位宽、高码率、高质量或无损的视频编解码场景中，可以提高系数编码的吞吐量和编解码速度，同时还可以提升压缩效率。

本申请实施例的技术方案可以如下实现：

第一方面，本申请实施例提供了一种系数解码方法，应用于解码器，该方法包括：

解析码流，获取视频标识信息；

20 当视频标识信息指示视频满足预设条件时，解析码流，获取最后一个非零系数位置翻转标识信息和最后一个非零系数的坐标信息；

当最后一个非零系数位置翻转标识信息指示当前块使用最后一个非零系数位置翻转时，对最后一个非零系数的坐标信息进行计算，确定最后一个非零系数的位置；

按照预设扫描顺序对最后一个非零系数的位置之前的所有系数进行解码，确定当前块的系数。

25 第二方面，本申请实施例提供了一种系数编码方法，应用于编码器，该方法包括：

确定视频标识信息以及最后一个非零系数的位置；

当视频标识信息指示视频满足预设条件时，确定最后一个非零系数位置翻转标识信息；

根据最后一个非零系数的位置以及最后一个非零系数位置翻转标识信息，确定最后一个非零系数的坐标信息；

30 按照预设扫描顺序对最后一个非零系数的位置之前的所有系数进行编码，并将编码后得到的比特信息、视频标识信息以及最后一个非零系数的坐标信息写入码流。

第三方面，本申请实施例提供了一种编码器，该编码器包括第一确定单元和编码单元；其中，

第一确定单元，配置为确定视频标识信息以及最后一个非零系数的位置；以及当视频标识信息指示视频满足预设条件时，确定最后一个非零系数位置翻转标识信息；

35 第一确定单元，还配置为根据最后一个非零系数的位置以及最后一个非零系数位置翻转标识信息，确定最后一个非零系数的坐标信息；

编码单元，配置为按照预设扫描顺序对最后一个非零系数的位置之前的所有系数进行编码，并将编码后得到的比特信息、视频标识信息以及最后一个非零系数的坐标信息写入码流。

第四方面，本申请实施例提供了一种编码器，该编码器包括第一存储器和第一处理器；其中，

40 第一存储器，用于存储能够在第一处理器上运行的计算机程序；

第一处理器，用于在运行计算机程序时，执行如第二方面所述的方法。

第五方面，本申请实施例提供了一种解码器，该解码器包括解析单元和第二确定单元；其中，

解析单元，配置为解析码流，获取视频标识信息；以及当视频标识信息指示视频满足预设条件时，解析码流，获取最后一个非零系数位置翻转标识信息和最后一个非零系数的坐标信息；

第二确定单元,配置为当最后一个非零系数位置翻转标识信息指示当前块使用最后一个非零系数位置翻转时,对最后一个非零系数的坐标信息进行计算,确定最后一个非零系数的位置;

解析单元,还配置为按照预设扫描顺序对最后一个非零系数的位置之前的所有系数进行解码,确定当前块的系数。

5 第六方面,本申请实施例提供了一种解码器,该解码器包括第二存储器和第二处理器;其中,

第二存储器,用于存储能够在第二处理器上运行的计算机程序;

第二处理器,用于在运行计算机程序时,执行如第一方面所述的方法。

第七方面,本申请实施例提供了一种计算机存储介质,该计算机存储介质存储有计算机程序,所述计算机程序被执行时实现如第一方面所述的方法、或者实现如第二方面所述的方法。

10 本申请实施例提供了一种系数编解码方法、编码器、解码器以及计算机存储介质,在编码器中,确定视频标识信息以及最后一个非零系数的位置;当视频标识信息指示视频满足预设条件时,确定最后一个非零系数位置翻转标识信息;根据最后一个非零系数的位置以及最后一个非零系数位置翻转标识信息,确定最后一个非零系数的坐标信息;按照预设扫描顺序对最后一个非零系数的位置之前的所有系数进行编码,并将编码后得到的比特信息、视频标识信息以及最后一个非零系数的坐标信息写入码流。在解码器中,解析码流,获取视频标识信息;当视频标识信息指示视频满足预设条件时,解析码流,获取最后
15 一个非零系数位置翻转标识信息和最后一个非零系数的坐标信息;当最后一个非零系数位置翻转标识信息指示当前块使用最后一个非零系数位置翻转时,对最后一个非零系数的坐标信息进行计算,确定最后一个非零系数的位置;按照预设扫描顺序对最后一个非零系数的位置之前的所有系数进行解码,确定当前块的系数。这样,在高位宽、高码率、高质量或无损的视频编解码场景中,由于系数分布规律与常规
20 视频场景不同,在系数编码中通过减少甚至消除上下文模式编码的语法元素个数,比如关于最后一个非零系数位置、子块编码标识等语法元素,甚至还可以在最后一个非零系数的坐标信息的值偏大时进行坐标变换,从而可以减少码流中编码所带来的开销,进而提高系数编码的吞吐量和编解码速度;另外,由于减少或消除的语法元素在高位宽、高码率、高质量或无损的视频编解码中的影响较小,还能够提升压缩效率。

25 附图说明

图1为相关技术提供了一种编码框架的应用示意图;

图2为相关技术提供了一种当前系数与相邻系数的位置关系示意图;

图3为相关技术提供了一种bin的算数解码过程的流程示意图;

图4为相关技术提供了一种二元符号的算数解码过程的流程示意图;

30 图5为相关技术提供了一种算数解码引擎的重归一化的流程示意图;

图6为相关技术提供了一种旁路解码过程的流程示意图;

图7为相关技术提供了一种可能有非零系数区域与强制为0区域的位置关系示意图;

图8A为本申请实施例提供了一种编码器的系统组成示意图;

图8B为本申请实施例提供了一种解码器的系统组成示意图;

35 图9为本申请实施例提供了一种系数解码方法的流程示意图;

图10A为本申请实施例提供了一种最后一个非零系数的位置相对于当前块左上角的位置示意图;

图10B为本申请实施例提供了一种最后一个非零系数的位置相对于当前块右下角的位置示意图;

图11为本申请实施例提供了一种系数编码方法的流程示意图;

图12为本申请实施例提供了一种编码器的组成结构示意图;

40 图13为本申请实施例提供了一种编码器的具体硬件结构示意图;

图14为本申请实施例提供了一种解码器的组成结构示意图;

图15为本申请实施例提供了一种解码器的具体硬件结构示意图。

具体实施方式

45 为了能够更加详尽地了解本申请实施例的特点与技术内容,下面结合附图对本申请实施例的实现进行详细阐述,所附附图仅供参考说明之用,并非用来限定本申请实施例。

除非另有定义,本文所使用的所有的技术和科学术语与属于本申请的技术领域的技术人员通常理解的含义相同。本文中所使用的术语只是为了描述本申请实施例的目的,不是旨在限制本申请。

在以下的描述中,涉及到“一些实施例”,其描述了所有可能实施例的子集,但是可以理解,“一些

实施例”可以是所有可能实施例的相同子集或不同子集，并且可以在不冲突的情况下相互结合。还需要指出，本申请实施例所涉及的术语“第一\第二\第三”仅是用于区别类似的对象，不代表针对对象的特定排序，可以理解地，“第一\第二\第三”在允许的情况下可以互换特定的顺序或先后次序，以使这里描述的本申请实施例能够以除了在这里图示或描述的以外的顺序实施。

5 在视频图像中，一般采用第一图像分量、第二图像分量和第三图像分量来表征编码块(Coding Block, CB);其中，这三个图像分量分别为一个亮度分量、一个蓝色色度分量和一个红色色度分量，具体地，亮度分量通常使用符号 Y 表示，蓝色色度分量通常使用符号 Cb 或者 U 表示，红色色度分量通常使用符号 Cr 或者 V 表示;这样，视频图像可以用 YCbCr 格式表示，也可以用 YUV 格式表示。

10 对本申请实施例进行进一步详细说明之前，先对本申请实施例中涉及的名词和术语进行说明，本申请实施例中涉及的名词和术语适用于如下的解释:

动态图像专家组 (Moving Picture Experts Group, MPEG)

国际标准化组织 (International Standardization Organization, ISO)

国际电工委员会 (International Electrotechnical Commission, IEC)

联合视频专家组 (Joint Video Experts Team, JVET)

15 开放媒体联盟 (Alliance for Open Media, AOM)

新一代视频编码标准 H.266/多功能视频编码 (Versatile Video Coding, VVC)

VVC 的参考软件测试平台 (VVC Test Model, VTM)

音视频编码标准 (Audio Video Standard, AVS)

AVS 的高性能测试模型 (High-Performance Model, HPM)

20 基于上下文的自适应二进制算术编码 (Context-based Adaptive Binary Arithmetic Coding, CABAC)

普通残差编码 (Regular Residual Coding, RRC)

变换跳过残差编码 (Transform Skip Residual Coding, TSRC)

可以理解，目前通用的视频编解码标准 (如 VVC) 都采用基于块的混合编码框架。视频图像中的每一帧被分割成相同大小 (如 128×128, 64×64 等) 的正方形的最大编码单元 (Largest Coding Unit, LCU), 每个最大编码单元还可以根据规则划分成矩形的编码单元 (Coding Unit, CU); 而且编码单元可能还会划分成更小的预测单元 (Prediction Unit, PU)、变换单元 (Transform Unit, TU) 等。具体地，如图 1 所示，混合编码框架可以包括有预测 (Prediction)、变换 (Transform)、量化 (Quantization)、熵编码 (Entropy Coding)、环路滤波 (In Loop Filter) 等模块。其中，预测模块可以包括帧内预测 (Intra Prediction) 和帧间预测 (Inter Prediction)，帧间预测可以包括运动估计 (Motion Estimation) 和运动补偿 (Motion Compensation)。由于视频图像的一个帧内相邻像素之间存在很强的相关性，在视频编解码技术中使用帧内预测方式能够消除相邻像素之间的空间冗余;但是由于视频图像中的相邻帧之间也存在着很强的相似性，在视频编解码技术中使用帧间预测方式消除相邻帧之间的时间冗余，从而能够提高编解码效率。

视频编解码器的基本流程如下:在编码器中，将一帧图像划分成块，对当前块使用帧内预测或帧间预测产生当前块的预测块，当前块的原始块减去预测块得到残差块，对残差块进行变换、量化得到量化系数矩阵，对量化系数矩阵进行熵编码输出到码流中。在解码器中，对当前块使用帧内预测或帧间预测产生当前块的预测块，另一方面解码码流得到量化系数矩阵，对量化系数矩阵进行反量化、逆变换得到残差块，将预测块和残差块相加得到重建块。重建块组成重建图像，基于图像或基于块对重建图像进行环路滤波得到解码图像。编码器同样需要和解码器类似的操作获得解码图像。解码图像可以为后续的帧作为帧间预测的参考帧。编码器确定的块划分信息、预测、变换、量化、熵编码、环路滤波等模式信息或者参数信息如果有必要需要输出到码流中;然后解码器通过解析及根据已有信息进行分析确定与编码器相同的块划分信息、预测、变换、量化、熵编码、环路滤波等模式信息或者参数信息，从而保证编码器获得的解码图像和解码器获得的解码图像相同。编码器获得的解码图像通常也叫做重建图像。在预测时可以将当前块划分成预测单元，在变换时可以将当前块划分成变换单元，预测单元和变换单元的划分可以不同。上述是基于块的混合编码框架下的视频编码器和解码器的基本流程，随着技术的发展，该框架或流程的一些模块或步骤可能会被优化，本申请实施例适用于该基于块的混合编码框架下的视频编解码器的基本流程，但不限于该框架及流程。

当前块 (Current Block) 可以是当前编码单元 (CU)、当前预测单元 (PU)、或者当前变换块 (TU) 等。

其中，块划分信息、预测、变换、量化的各种模式及参数信息，系数等通过熵编码写入码流中。假定不同元素的概率是不同的，为出现概率较大的元素分配较短的码字，为出现概率较小的元素分配较长的码字，能获得比定长编码更高的编码效率。但是如果不同元素的概率相近或基本相同，熵编码的压缩空间就有限。CABAC 是一种常用熵编码方法，HEVC 和 VVC 等都使用 CABAC 进行熵编码。CABAC

可以使用上下文模型可以提升压缩效率,但是上下文模式的使用及更新也使得操作更加复杂。CABAC中有一种旁路(bypass)模式,在旁路模式下不需要使用和更新上下文模型,可以达到更高的吞吐量(throughput)。在本申请实施例中,可以把CABAC中需要使用和更新上下文模型的模式叫做上下文模式。

5 一般情况下,需要先根据定义的方法确定上下文模型,在调用定义的二元符号的算数解码过程时,上下文模型的参数可以作为输入。上下文模型的选择在相邻系数之间还存在依赖关系。例如,图2示出了相关技术提供的一种当前系数与相邻系数的位置关系示意图。在图2中,黑色填充的块表示当前系数,网格线填充的块表示相邻系数;如2图所示,当前系数的sig_coeff_flag的选择哪个上下文模型需要根据其右边和下边和右下边相邻的5个系数的信息来确定。根据图2进一步可以看出,上下文模式的操作较旁路模式的操作复杂得多,而且还有相邻系数之间的依赖性。

10 对于CABAC的算数编码引擎,如果需要使用上下文模式,需要调用定义的二元符号的算数解码过程,其中包括状态转移过程也就是上下文模型的更新。而在二元符号的算数解码过程中会调用算数解码引擎的重归一化过程。而在使用旁路模式时需要调用旁路解码过程。

下面以VVC中使用CABAC为例进行介绍:

15 对于CABAC的算数编码引擎,算数解码过程的输入是ctxTable, ctxIdx, bypassFlag以及算数解码引擎的状态变量ivlCurrRange和ivlOffset,算数解码过程的输出是bin的值。

其中,ctxTable是选择上下文模式时使用的表格,ctxIdx是上下文模型索引。

图3示出了相关技术提供的一种bin的算数解码过程的流程示意图。如图3所示,为了解码bin的值,上下文索引表ctxTable, ctxIdx, bypassFlag作为输入被传输到算数解码过程DecodeBin(ctxTable, ctxIdx, bypassFlag),具体如下:

如果bypassFlag的值为1,则调用旁路解码过程DecodeBypass();

否则,如果bypassFlag的值为0, ctxTable的值为0,且ctxIdx的值为0,则调用DecodeTerminate();

25 否则(bypassFlag的值为0且ctxTable的值不为0),调用定义的二元符号的算数解码过程DecodeDecision(ctxTable, ctxIdx)。

进一步地,对于二元符号的算数解码过程,该过程的输入是变量ctxTable, ctxIdx, ivlCurrRange,和ivlOffset,该过程的输出是解码的值binVal,更新后的变量ivlCurrRange和ivlOffset。

图4示出了相关技术提供的一种二元符号的算数解码过程的流程示意图。如图4所示,其中, pStateIdx0和pStateIdx1是当前上下文模型的两个状态。

30 (1)变量ivlLpsRange的值按如下方法导出:

给定ivlCurrRange的当前值,变量qRangeIdx按如下方法导出:

$qRangeIdx = ivlCurrRange \gg 5$

给定qRangeIdx, ctxTable和ctxIdx对应的pStateIdx0和pStateIdx1, valMps和ivlLpsRange按如下方法导出:

35 $pState = pStateIdx1 + 16 \times pStateIdx0$;

$valMps = pState \gg 14$;

$ivlLpsRange = (qRangeIdx \times ((valMps ? 32767 - pState : pState) \gg 9) \gg 1) + 4$ 。

(2)变量ivlCurrRange的值设为ivlCurrRange - ivlLpsRange,并执行如下操作:

40 如果ivlOffset大于或等于ivlCurrRange,那么变量binVal的值为1 - valMps,ivlOffset的值为ivlOffset减去ivlCurrRange,ivlCurrRange的值为ivlLpsRange;

否则(ivlOffset小于ivlCurrRange),变量binVal的值为valMps。

给定binVal的值,执行定义的状态转移(statetransition)。基于ivlCurrRange的当前值,可以执行定义的重归一化(renormalization)。

45 进一步地,对于状态转移过程,该过程的输入是当前的pStateIdx0和pStateIdx1,以及解出的值binVal;输出是更新的ctxTable和ctxIdx对应的上下文变量pStateIdx0和pStateIdx1。其中,变量shift0和shift1由shiftIdx导出,这里定义了shiftIdx与ctxTable和ctxIdx的对应关系,如下:

$shift0 = (shiftIdx \gg 2) + 2$;

$shift1 = (shiftIdx \& 3) + 3 + shift0$ 。

基于解出的值binVal,与ctxTable和ctxIdx对应的两个变量pStateIdx0和pStateIdx1的更新如下:

50 $pStateIdx0 = pStateIdx0 - (pStateIdx0 \gg shift0) + (1023 \times binVal \gg shift0)$;

$pStateIdx1 = pStateIdx1 - (pStateIdx1 \gg shift1) + (16383 \times binVal \gg shift1)$ 。

进一步地,算数解码引擎的重归一化过程的输入是片数据中的bits和变量ivlCurrRange和ivlOffset,

输出是更新的变量 $ivlCurrRange$ 和 $ivlOffset$ 。

图 5 示出了相关技术提供的一种算数解码引擎的重归一化的流程示意图,如图 5 所示, $ivlCurrRange$ 的当前值首选与 256 比较,后续步骤如下:

- 如果 $ivlCurrRange$ 大于或等于 256,那么不需要重归一化, RenormD 过程结束;
- 5 否则 ($ivlCurrRange$ 小于 256),进入重归一化循环。在这个循环中, $ivlCurrRange$ 的值乘 2,也就是左移一位。 $ivlOffset$ 的值乘 2,也就是左移一位。用 $read_bits(1)$ 得到的一位移入到 $ivlOffset$ 中。
- 整个过程中码流中的数据不应该导致 $ivlOffset$ 大于或等于 $ivlCurrRange$ 。
- 进一步地,二元符号的旁路解码过程的输入是片数据的 bits 和变量 $ivlCurrRange$ 和 $ivlOffset$,输出是更新的变量 $ivlOffset$ 和解出的值 $binVal$ 。

- 10 当 $bypassFlag$ 为 1 时旁路解码过程被调用,图 6 示出了相关技术提供的一种旁路解码过程的流程示意图,如图 6 所示,首先 $ivlOffset$ 的值乘 2,也就是左移一位。用 $read_bits(1)$ 得到的一位移入到 $ivlOffset$ 中。然后 $ivlOffset$ 的值会和 $ivlCurrRange$ 的值比较,后续步骤如下:

- 如果 $ivlOffset$ 大于或等于 $ivlCurrRange$,那么 $binVal$ 的值设为 1, $ivlOffset$ 等于 $ivlOffset - ivlCurrRange$;
- 15 否则 ($ivlOffset$ 小于 $ivlCurrRange$), $binVal$ 的值设为 0。
- 整个过程中码流中的数据不应该导致 $ivlOffset$ 大于或等于 $ivlCurrRange$ 。

- 还应理解,在现在的视频编解码标准中,对于残差通常支持一种或多种变换 (transform) 以及变换跳过 (transform skip)。变换有离散余弦变换 (Discrete Cosine Transform, DCT) 等,使用变换的残差块在变换 (及量化) 以后通常体现出一定的特性,如某些变换 (及量化) 后由于能量大多集中在低频区域而使得左上角的区域系数较大,右下角的区域系数较小甚至出现很多 0 系数。而变换跳过,顾名思义不做变换,变换跳过后系数的分布规律与变换后的系数是不同的,因而可以使用不同的系数编码方法。比如 VVC 中对变换跳过后系数的分布使用 RRC,对变换跳过的系数使用 TSRC。
- 20 一般的变换,如 DCT 变换,这时候变换后的块,从左往右代表的频率由低到高,从上到下代表的频率由低到高。左上角代表低频,右下角代表高频。人眼对低频的信息更敏感,而对高频的信息不是特别敏感。利用这种特性,一些高频的信息被做更多的处理或去掉对视觉影响较小。有些技术,如强制置零 (zero-out),可以强制将某些高频信息置 0。比如对 64x64 的块强制将横坐标大于或等于 32、或者纵坐标大于或等于 32 的位置的系数置 0。上述只是一个简单的示例,zero-out 的范围可能会有更多复杂的导出方法,这里不再赘述。如图 7 所示,左上角部分可能存在非零系数 (即为可能有非零系数区域),右下角部分将全部置零 (即为强制为 0 区域)。这样,对于后面的系数编码来说,zero-out 强制置 0 的区域的系数由于肯定是 0 而不需要编码。

- 进一步的,由于对通常的视频的残差进行变换 (及量化) 后系数分布显现出左上角系数较大而右下角很多 0 系数的特性,系数编码时通常会使用一些方法使得左上角一定范围内的系数需要进行编码,而右下角一定范围内的系数不需要编码,也就是默认这些系数为 0。一种方法是在编码一个块的系数时,先确定这个块的按扫描顺序的最后一个非零系数的位置。确定出这个位置以后,按扫描顺序最后一个非零系数的位置后面的系数全部认为是 0,即不需要编码;只有最后一个非零系数位置及之前的系数需要编码。例如在 VVC 中,使用 $last_sig_coeff_x_prefix$ 、 $last_sig_coeff_y_prefix$ 、 $last_sig_coeff_x_suffix$ 、 $last_sig_coeff_y_suffix$ 来确定最后一个非零系数的位置 ($LastSignificantCoeffX$, $LastSignificantCoeffY$)。
- 25 一般的变换,如 DCT 变换,这时候变换后的块,从左往右代表的频率由低到高,从上到下代表的频率由低到高。左上角代表低频,右下角代表高频。人眼对低频的信息更敏感,而对高频的信息不是特别敏感。利用这种特性,一些高频的信息被做更多的处理或去掉对视觉影响较小。有些技术,如强制置零 (zero-out),可以强制将某些高频信息置 0。比如对 64x64 的块强制将横坐标大于或等于 32、或者纵坐标大于或等于 32 的位置的系数置 0。上述只是一个简单的示例,zero-out 的范围可能会有更多复杂的导出方法,这里不再赘述。如图 7 所示,左上角部分可能存在非零系数 (即为可能有非零系数区域),右下角部分将全部置零 (即为强制为 0 区域)。这样,对于后面的系数编码来说,zero-out 强制置 0 的区域的系数由于肯定是 0 而不需要编码。

- 进一步的,由于对通常的视频的残差进行变换 (及量化) 后系数分布显现出左上角系数较大而右下角很多 0 系数的特性,系数编码时通常会使用一些方法使得左上角一定范围内的系数需要进行编码,而右下角一定范围内的系数不需要编码,也就是默认这些系数为 0。一种方法是在编码一个块的系数时,先确定这个块的按扫描顺序的最后一个非零系数的位置。确定出这个位置以后,按扫描顺序最后一个非零系数的位置后面的系数全部认为是 0,即不需要编码;只有最后一个非零系数位置及之前的系数需要编码。例如在 VVC 中,使用 $last_sig_coeff_x_prefix$ 、 $last_sig_coeff_y_prefix$ 、 $last_sig_coeff_x_suffix$ 、 $last_sig_coeff_y_suffix$ 来确定最后一个非零系数的位置 ($LastSignificantCoeffX$, $LastSignificantCoeffY$)。
- 30 (a) $last_sig_coeff_x_prefix$ 确定当前块中按扫描顺序最后一个非零系数的水平 (或列) 坐标的前缀 (prefix)。 $last_sig_coeff_x_prefix$ 的值应该在 0 到 $(\log_2 Z_{oTbWidth} < 1) - 1$, 包含这两个边界值的范围内。

- 40 如果 $last_sig_coeff_x_prefix$ 不存在,那么 $last_sig_coeff_x_prefix$ 的值为 0。

- (b) $last_sig_coeff_y_prefix$ 确定当前块中按扫描顺序最后一个非零系数的竖直 (或行) 坐标的前缀 (prefix)。 $last_sig_coeff_y_prefix$ 的值应该在 0 到 $(\log_2 Z_{oTbHeight} < 1) - 1$, 包含这两个边界值的范围内。

- 45 如果 $last_sig_coeff_y_prefix$ 不存在,那么 $last_sig_coeff_y_prefix$ 的值为 0。

- (c) $last_sig_coeff_x_suffix$ 确定当前块中按扫描顺序最后一个非零系数的水平 (或列) 坐标的后缀 (suffix)。 $last_sig_coeff_x_suffix$ 的值应该在 0 到 $(1 < ((last_sig_coeff_x_prefix > 1) - 1)) - 1$, 包含这两个边界值的范围内。

- 当前变换块中按扫描顺序最后一个非零系数的水平 (或列) 坐标的值 $LastSignificantCoeffX$ 按如下方法导出:
- 50 如果 $last_sig_coeff_x_suffix$ 不存在,那么

$LastSignificantCoeffX = last_sig_coeff_x_prefix$;

- 否则 (last_sig_coeff_x_suffix 存在),

$$\text{LastSignificantCoeffX} = (1 \ll ((\text{last_sig_coeff_x_prefix} \gg 1) - 1)) * (2 + (\text{last_sig_coeff_x_prefix} \& 1)) + \text{last_sig_coeff_x_suffix}.$$
- (d) last_sig_coeff_y_suffix 确定当前变换块中按扫描顺序最后一个非零系数的竖直 (或行) 坐标的后缀 (suffix)。last_sig_coeff_x_suffix 的值应该在 0 到 $(1 \ll ((\text{last_sig_coeff_y_prefix} \gg 1) - 1)) - 1$, 包含这两个边界值的范围内。
- 当前变换块中按扫描顺序最后一个非零系数的竖直 (或行) 坐标的值 LastSignificantCoeffY 按如下方法导出:
- 如果 last_sig_coeff_y_suffix 不存在, 那么
- 10 LastSignificantCoeffY = last_sig_coeff_y_prefix;
 否则 (last_sig_coeff_y_suffix 存在),

$$\text{LastSignificantCoeffY} = (1 \ll ((\text{last_sig_coeff_y_prefix} \gg 1) - 1)) * (2 + (\text{last_sig_coeff_y_prefix} \& 1)) + \text{last_sig_coeff_y_suffix}.$$
- 进一步地, 最后一个非零系数及之前的系数全都要进行编码, 但是在普通视频中, 即使是这些系数中仍然有一定比例的系数是 0, VVC 中使用当前子块是否编码的标志 sb_coded_flag 来确定当前子块中的系数是否需要编码。如果不需要编码则认为当前子块中的系数全为 0。这里, 子块通常是 $n \times n$ 的子块, 如 4×4 的子块。
- sb_coded_flag[xS][yS] 确定当前变换块中位置在 (xS, yS) 的子块的如下信息, 其中, 子块是变换系数值的数组:
- 20 如果 sb_coded_flag[xS][yS] 的值为 0, 那么当前变换块中位置在 (xS, yS) 的子块内的所有变换系数的值为 0;
 如果 sb_coded_flag[xS][yS] 不存在, 那么 sb_coded_flag[xS][yS] 的值为 1。
- 进一步地, 在处理系数编码时, 利用系数的特性可以提升压缩效率。例如, 对通常的视频, 需要编码的系数中, 有一定比例的系数是 0, 因而可以用一个语法元素来表示当前系数是否是 0, 这个语法元素通常是一个二元符号。如果当前系数是 0, 那么意味着对当前系数的编码已经结束; 否则, 还需要继续编码当前系数。又如, 对通常的视频, 在非 0 系数中, 有一定比例的系数的绝对值是 1, 因而可以用一个语法元素来表示当前系数的绝对值是否大于 1, 这个语法元素通常是一个二元符号。如果当前系数的绝对值不大于 1, 那么意味着对当前系数的编码已经结束; 否则, 还需要继续编码当前系数。例如, 在 VVC 中涉及的语法元素如下,
- 30 sig_coeff_flag[xC][yC] 用来确定当前变换块在变换系数位置 (xC, yC) 的对应的变换系数是不是非零系数:
 如果 sig_coeff_flag[xC][yC] 的值是 0, 那么 (xC, yC) 位置的变换系数的值设为 0;
 否则 (sig_coeff_flag[xC][yC] 的值是 1), (xC, yC) 位置的变换系数是非零系数。
 如果 sig_coeff_flag[xC][yC] 不存在, 那么做如下推断:
- 35 如果 transform_skip_flag[x0][y0][cIdx] 的值为 0 或者 sh_ts_residual_coding_disabled_flag 的值为 1:
 如果 (xC, yC) 是最后按扫描顺序一个非零系数的位置 (LastSignificantCoeffX, LastSignificantCoeffY) 或如下条件全为真, 那么 sig_coeff_flag[xC][yC] 的值推断为 1:

$$(xC \& ((1 \ll (\log_2 \text{SbW}) - 1)), yC \& ((1 \ll (\log_2 \text{SbH}) - 1))) \text{ 等于 } (0, 0);$$

 inferSbDcSigCoeffFlag 的值等于 1;
 sb_coded_flag[xS][yS] 的值 1;
 40 否则, sig_coeff_flag[xC][yC] 的值推断为 0;
 否则 (transform_skip_flag[x0][y0][cIdx] 的值为 1 且 sh_ts_residual_coding_disabled_flag 的值为 0):
 如果如下条件全为真, 那么 sig_coeff_flag[xC][yC] 的值推断为 1:

$$(xC \& ((1 \ll (\log_2 \text{SbW}) - 1)), yC \& ((1 \ll (\log_2 \text{SbH}) - 1))) \text{ 等于 } ((1 \ll \log_2 \text{SbW}) - 1, (1 \ll \log_2 \text{SbH}) - 1);$$

 inferSbSigCoeffFlag 的值为 1;
 sb_coded_flag[xS][yS] 的值为 1;
 45 否则, sig_coeff_flag[xC][yC] 的值推断为 0。
- 50 abs_level_gtx_flag[n][j] 用来确定 (扫描顺序第 n 个) 变换系数的绝对值是否大于 $(j \ll 1) + 1$ 。如果 abs_level_gtx_flag[n][j] 不存在, 那么 abs_level_gtx_flag[n][j] 的值为 0。
- 这样, 如果在处理完上述标志 (或称为语法元素) 后, 当前系数还没有编码结束, 那么需要编码系

数绝对值的剩余值。如 VVC 中的 `abs_remainder`。

`abs_remainder[n]` 用来确定扫描顺序第 n 个变换系数的用 Golomb-Rice 编码的剩余绝对值。如果 `abs_remainder[n]` 不存在, 那么 `abs_remainder[n]` 的值为 0。

- 进一步地, 在 VVC 中, `sig_coeff_flag` 和 `abs_level_gtx_flag` 等语法元素使用上下文模式编码, 而 `abs_remainder` 使用旁路模式编码。上述提到了上下文模式编码较旁路模式编码更复杂, 直观理解就是处理起来更慢, 如果需要编码的系数很多, 使用太多的上下文模式编码会影响解码速度。因而可以对使用上下文模式编码的语法元素的个数进行限制, 比如使用上下文模式编码的二元符号的个数超过一个阈值后, 后续的系数编码强制使用旁路模式编码。如 VVC 中的 `dec_abs_level`。

- `dec_abs_level[n]` 是扫描位置 n 处的一个用 Golomb-Rice 编码的中间值。解析 `dec_abs_level[n]` 时可以导出 `ZeroPos[n]`, 在位置 (xC, yC) 处的量化系数的绝对值 `AbsLevel[xC][yC]` 按如下方法导出:

如果 `dec_abs_level[n]` 不存在或 `dec_abs_level[n]` 的值等于 `ZeroPos[n]`, 那么 `AbsLevel[xC][yC]` 的值为 0;

否则, 如果 `dec_abs_level[n]` 的值小于 `ZeroPos[n]`, 那么 `AbsLevel[xC][yC]` 的值为 `dec_abs_level[n]+1`;

否则 (`dec_abs_level[n]` 的值大于 `ZeroPos[n]`), `AbsLevel[xC][yC]` 的值为 `dec_abs_level[n]`。

- 上述涉及的都是系数的绝对值, 非零系数的正负号可以用系数正负号标志 `coeff_sign_flag` 或一些正负号导出的方法来确定。 `coeff_sign_flag[n]` 可以按如下方法确定扫描位置 n 的变换系数的正负号:

如果 `coeff_sign_flag[n]` 的值为 0, 那么对应的变换系数为正值;

否则 (`coeff_sign_flag[n]` 的值为 1), 对应的变换系数为负值。

如果 `coeff_sign_flag[n]` 不存在, 那么 `coeff_sign_flag[n]` 的值为 0; 这时候根据 `CoeffSignLevel[xC][yC]` 确定坐标 (xC, yC) 上的变换系数的正负号:

- 如果 `CoeffSignLevel[xC][yC]` 的值为 0, 那么对应的变换系数为 0;

否则, 如果 `CoeffSignLevel[xC][yC]` 的值为 1, 那么对应的变换系数为正值; 否则 (`CoeffSignLevel[xC][yC]` 的值为 -1), 对应的变换系数为负值。

还需要说明的是, `CoeffSignLevel[xC][yC]` 还可以使用其他一些方法导出, 这里不再赘述。

- 另外, 在 VVC 中还使用了一个系数的值的奇偶性标志 `par_level_flag`。根据这个标志可以知道当前系数的值的奇偶性, 在确定当前系数的值以及依赖量化 (Dependent Quantization) 中会使用到。

`par_level_flag[n]` 确定按扫描顺序位置 n 的变换系数的奇偶性。如果 `par_level_flag[n]` 不存在, 那么 `par_level_flag[n]` 的值为 0。

`par_level_flag` 除了确定变换系数的奇偶性还可以用来和 `abs_level_gtx_flag`、`abs_remainder` 等共同确定系数的大小。

- 在这里, 由于上下文模式编码需要选择、使用并更新上下文模式, 而旁路模式编码不需要选择、使用并更新上下文模式, 通常的做法是在一定范围内把上下文模式编码的语法元素放在一起, 旁路模式编码的语法元素放在一起, 这样对硬件设计更加友好。比如说先把一个块中的所有的上下文模式编码的语法元素处理完, 再处理旁路模式编码的语法元素。当前块中的所有的上下文模式编码的语法元素可能还会分几个组, 一个块中的所有的旁路模式编码的语法元素可能还会分几个组。

- 在一种具体的示例中, RRC 的具体语法如表 1 所示。

表 1

residual_coding ($x0, y0, \log2TbWidth, \log2TbHeight, cIdx$) {	Descriptor
if (<code>sps_mts_enabled_flag</code> && <code>cu_sbt_flag</code> && <code>cIdx==0</code> && <code>log2TbWidth==5</code> && <code>log2TbHeight<6</code>)	
<code>log2ZoTbWidth=4</code>	
else	
<code>log2ZoTbWidth=Min (log2TbWidth , 5)</code>	
if (<code>sps_mts_enabled_flag</code> && <code>cu_sbt_flag</code> && <code>cIdx==0</code> && <code>log2TbWidth<6</code> && <code>log2TbHeight==5</code>)	
<code>log2ZoTbHeight=4</code>	
else	
<code>log2ZoTbHeight=Min (log2TbHeight , 5)</code>	
if (<code>log2TbWidth > 0</code>)	
<code>last_sig_coeff_x_prefix</code>	<code>ae(v)</code>

if (log2TbHeight > 0)	
last_sig_coeff_y_prefix	ae(v)
if (last_sig_coeff_x_prefix > 3)	
last_sig_coeff_x_suffix	ae(v)
if (last_sig_coeff_y_prefix > 3)	
last_sig_coeff_y_suffix	ae(v)
log2TbWidth=log2ZoTbWidth	
log2TbHeight=log2ZoTbHeight	
remBinsPass1= ((1<< (log2TbWidth+log2TbHeight)) × 7) >> 2	
log2SbW= (Min (log2TbWidth , log2TbHeight) < 2 ? 1 : 2)	
log2SbH=log2SbW	
if (log2TbWidth+log2TbHeight > 3)	
if (log2TbWidth < 2) {	
log2SbW=log2TbWidth	
log2SbH=4-log2SbW	
} elseif (log2TbHeight < 2) {	
log2SbH=log2TbHeight	
log2SbW=4-log2SbH	
}	
numSbCoeff=1<< (log2SbW+log2SbH)	
lastScanPos=numSbCoeff	
lastSubBlock= (1<< (log2TbWidth+log2TbHeight- (log2SbW+log2SbH))) -1	
do {	
if (lastScanPos == 0) {	
lastScanPos=numSbCoeff	
lastSubBlock--	
}	
lastScanPos--	
xS=DiagScanOrder[log2TbWidth-log2SbW][log2TbHeight-log2SbH]	
[lastSubBlock][0]	
yS=DiagScanOrder[log2TbWidth-log2SbW][log2TbHeight-log2SbH]	
[lastSubBlock][1]	
xC= (xS<<log2SbW) +DiagScanOrder[log2SbW][log2SbH][lastScanPos][0]	
yC= (yS<<log2SbH) +DiagScanOrder[log2SbW][log2SbH][lastScanPos][1]	
} while ((xC != LastSignificantCoeffX) (yC != LastSignificantCoeffY))	
if (lastSubBlock == 0 && log2TbWidth >= 2 && log2TbHeight >= 2 && !transform_skip_flag[x0][y0][cIdx] && lastScanPos > 0)	
LfstDcOnly=0	
if ((lastSubBlock > 0 && log2TbWidth >= 2 && log2TbHeight >= 2) (lastScanPos > 7 && (log2TbWidth == 2 log2TbWidth == 3) && log2TbWidth == log2TbHeight))	
LfstZeroOutSigCoeffFlag=0	
if ((lastSubBlock > 0 lastScanPos > 0) && cIdx == 0)	

MtsDcOnly=0	
QState=0	
for (i=lastSubBlock ; i >=0 ; i--) {	
startQStateSb=QState	
xS=DiagScanOrder[log2TbWidth-log2SbW][log2TbHeight-log2SbH]	
[i][0]	
yS=DiagScanOrder[log2TbWidth-log2SbW][log2TbHeight-log2SbH]	
[i][1]	
inferSbDcSigCoeffFlag=0	
if (i<lastSubBlock && i>0) {	
sb_coded_flag [xS][yS]	ae(v)
inferSbDcSigCoeffFlag=1	
}	
if (sb_coded_flag[xS][yS] && (xS>3 yS>3) && cIdx==0)	
MtsZeroOutSigCoeffFlag=0	
firstSigScanPosSb=numSbCoeff	
lastSigScanPosSb=-1	
firstPosMode0= (i == lastSubBlock ? lastScanPos : numSbCoeff-1)	
firstPosMode1=firstPosMode0	
for (n=firstPosMode0 ; n>=0 && remBinsPass1>=4 ; n--) {	
xC= (xS<<log2SbW) +DiagScanOrder[log2SbW][log2SbH][n][0]	
yC= (yS<<log2SbH) +DiagScanOrder[log2SbW][log2SbH][n][1]	
if (sb_coded_flag[xS][yS] && (n>0 !inferSbDcSigCoeffFlag) &&	
(xC != LastSignificantCoeffX yC != LastSignificantCoeffY)) {	
sig_coeff_flag [xC][yC]	ae(v)
remBinsPass1--	
if (sig_coeff_flag[xC][yC])	
inferSbDcSigCoeffFlag=0	
}	
if (sig_coeff_flag[xC][yC]) {	
abs_level_gtx_flag [n][0]	ae(v)
remBinsPass1--	
if (abs_level_gtx_flag[n][0]) {	
par_level_flag [n]	ae(v)
remBinsPass1--	
abs_level_gtx_flag [n][1]	ae(v)
remBinsPass1--	
}	
if (lastSigScanPosSb == -1)	
lastSigScanPosSb=n	
firstSigScanPosSb=n	
}	
AbsLevelPass1[xC][yC]=sig_coeff_flag[xC][yC]+par_level_flag[n]+	
abs_level_gtx_flag[n][0]+2×abs_level_gtx_flag[n][1]	
if (sh_dep_quant_used_flag)	
QState=QStateTransTable[QState][AbsLevelPass1[xC][yC]&1]	
firstPosMode1=n-1	
}	

for (n=firstPosMode0 ; n>firstPosMode1 ; n--) {	
xC= (xS<<log2SbW) +DiagScanOrder[log2SbW][log2SbH][n][0]	
yC= (yS<<log2SbH) +DiagScanOrder[log2SbW][log2SbH][n][1]	
if (abs_level_gtx_flag[n][1])	
abs_remainder[n]	ae(v)
AbsLevel[xC][yC]=AbsLevelPass1[xC][yC]+2×abs_remainder[n]	
}	
for (n=firstPosMode1 ; n>=0 ; n--) {	
xC= (xS<<log2SbW) +DiagScanOrder[log2SbW][log2SbH][n][0]	
yC= (yS<<log2SbH) +DiagScanOrder[log2SbW][log2SbH][n][1]	
if (sb_coded_flag[xS][yS])	
dec_abs_level[n]	ae(v)
if (AbsLevel[xC][yC]>0) {	
if (lastSigScanPosSb == -1)	
lastSigScanPosSb=n	
firstSigScanPosSb=n	
}	
if (sh_dep_quant_used_flag)	
QState=QStateTransTable[QState][AbsLevel[xC][yC]&1]	
}	
signHiddenFlag=sh_sign_data_hiding_used_flag && (lastSigScanPosSb-firstSigScanPosSb>3 ?1 : 0)	
for (n=numSbCoeff-1 ; n>=0 ; n--) {	
xC= (xS<<log2SbW) +DiagScanOrder[log2SbW][log2SbH][n][0]	
yC= (yS<<log2SbH) +DiagScanOrder[log2SbW][log2SbH][n][1]	
if ((AbsLevel[xC][yC]>0) && (!signHiddenFlag (n != firstSigScanPosSb)))	
coeff_sign_flag[n]	ae(v)
}	
if (sh_dep_quant_used_flag) {	
QState=startQStateSb	
for (n=numSbCoeff-1 ; n>=0 ; n--) {	
xC= (xS<<log2SbW) +DiagScanOrder[log2SbW][log2SbH][n][0]	
yC= (yS<<log2SbH) +DiagScanOrder[log2SbW][log2SbH][n][1]	
if (AbsLevel[xC][yC]>0)	
TransCoeffLevel[x0][y0][cIdx][xC][yC]= (2×AbsLevel[xC][yC]- (QState>1 ?1 : 0)) × (1-2×coeff_sign_flag[n])	
QState=QStateTransTable[QState][AbsLevel[xC][yC]&1]	
}	
} else{	
sumAbsLevel=0	
for (n=numSbCoeff-1 ; n>=0 ; n--) {	

$xC = (xS \ll \log 2SbW) + \text{DiagScanOrder}[\log 2SbW][\log 2SbH][n][0]$	
$yC = (yS \ll \log 2SbH) + \text{DiagScanOrder}[\log 2SbW][\log 2SbH][n][1]$	
if (AbsLevel[xC][yC]>0) {	
TransCoeffLevel[x0][y0][cIdx][xC][yC] =	
AbsLevel[xC][yC] × (1-2×coeff_sign_flag[n])	
if (signHiddenFlag) {	
sumAbsLevel += AbsLevel[xC][yC]	
if (n == firstSigScanPosSb && sumAbsLevel % 2 == 1)	
TransCoeffLevel[x0][y0][cIdx][xC][yC] =	
-TransCoeffLevel[x0][y0][cIdx][xC][yC]	
}	
}	
}	
}	
}	
}	
}	

其中，数组 AbsLevel[xC][yC] 代表当前变换块的变换系数的绝对值的数组。数组 AbsLevelPass1[xC][yC] 代表当前变换块的变换系数的部分重建的绝对值的数组。数组的索引 xC 和 yC 表示当前变换块中的 (xC, yC) 位置。

- 进入 residual_coding (x0, y0, log2TbWidth, log2TbHeight, cIdx) 函数后，需要确定一些块尺寸的信息，如确定 zero-out 后的块尺寸的对数 log2ZoTbWidth、log2ZoTbHeight，横坐标在 [0, (1<<log2ZoTbWidth) - 1]，纵坐标在 [0, (1<<log2ZoTbHeight) - 1] 范围内的系数可能是非零系数。这里，(1<<log2ZoTbWidth) 表示 zero-out 后的变换块的宽度，(1<<log2ZoTbHeight) 表示 zero-out 后的变换块的高度。然后根据 last_sig_coeff_x_prefix、last_sig_coeff_y_prefix、last_sig_coeff_x_suffix、last_sig_coeff_y_suffix 等确定最后一个非零系数的位置。按扫描顺序最后一个非零系数之前的系数可能是非零系数。然后确定 remBinsPass1 的值，即使用等式 $\text{remBinsPass1} = ((1 \ll (\log 2TbWidth + \log 2TbHeight)) \times 7) \gg 2$ 确定。remBinsPass1 就确定了当前块中使用上下文模式编码的语法元素的个数，本申请实施例可以理解 remBinsPass1 为 remaining binaries inpass1 的意思，也就是第一轮中剩余的符号的个数。按扫描顺序最后一个非零系数之前的系数需要编码。对这些系数所在的子块，依次确定当前子块是否需要编码，如果需要编码，该方法中把一个子块中上下文模式编码的语法元素放在了第一轮中，旁路模式编码的语法元素放在了后面。每个系数最多可能需要处理 4 个上下文模式编码的语法元素 1 个 sig_coeff_flag，2 个 abs_level_gtx_flag，1 个 par_level_flag。在第一轮中，每次处理完一个上下文模式编码的语法元素 remBinsPass1 就会减 1。如果一个系数足够大，在第一轮处理完上下文模式编码的几个语法元素后还需要处理剩余值，即 abs_remainder。而如果 remBinsPass1 已经足够小了 (不满足 $\text{remBinsPass1} \geq 4$)，第一轮就会结束，剩下的系数就会直接用旁路模式处理，即 dec_abs_level。
- 在另一种具体的示例中，TSRC 的具体语法如表 2 所示。

表 2

residual_ts_coding (x0 , y0 , log2TbWidth , log2TbHeight , cIdx) {	Descriptor
log2SbW = (Min (log2TbWidth , log2TbHeight) < 2 ? 1 : 2)	
log2SbH = log2SbW	
if (log2TbWidth + log2TbHeight > 3)	
if (log2TbWidth < 2) {	
log2SbW = log2TbWidth	
log2SbH = 4 - log2SbW	
} elseif (log2TbHeight < 2) {	
log2SbH = log2TbHeight	
log2SbW = 4 - log2SbH	
}	
numSbCoeff = 1 << (log2SbW + log2SbH)	

lastSubBlock= (1<< (log2TbWidth+log2TbHeight- (log2SbW+log2SbH))) -1	
inferSbCbf=1	
RemCbs= ((1<< (log2TbWidth+log2TbHeight)) >>2	
for (i=0 ; i <= lastSubBlock ; i++) {	
xS=DiagScanOrder[log2TbWidth-log2SbW][log2TbHeight-log2SbH][i][0]	
yS=DiagScanOrder[log2TbWidth-log2SbW][log2TbHeight-log2SbH][i][1]	
if (i != lastSubBlock !inferSbCbf)	
sb_coded_flag [xS][yS]	ae(v)
if (sb_coded_flag[xS][yS] && i<lastSubBlock)	
inferSbCbf=0	
/×Firstscanpass ×/	
inferSbSigCoeffFlag=1	
lastScanPosPass1=-1	
for (n=0 ; n <= numSbCoeff-1 && RemCbs>=4 ; n++) {	
xC= (xS<<log2SbW) +DiagScanOrder[log2SbW][log2SbH][n][0]	
yC= (yS<<log2SbH) +DiagScanOrder[log2SbW][log2SbH][n][1]	
lastScanPosPass1=n	
if (sb_coded_flag[xS][yS] &&	
(n != numSbCoeff-1 !inferSbSigCoeffFlag)) {	
sig_coeff_flag [xC][yC]	ae(v)
RemCbs--	
if (sig_coeff_flag[xC][yC])	
inferSbSigCoeffFlag=0	
}	
CoeffSignLevel[xC][yC]=0	
if (sig_coeff_flag[xC][yC]) {	
coeff_sign_flag [n]	ae(v)
RemCbs--	
CoeffSignLevel[xC][yC]= (coeff_sign_flag[n]>0 ?-1 : 1)	
abs_level_gtx_flag [n][0]	ae(v)
RemCbs--	
if (abs_level_gtx_flag[n][0]) {	
par_level_flag [n]	ae(v)
RemCbs--	
}	
}	
AbsLevelPass1[xC][yC]=	
sig_coeff_flag[xC][yC]+par_level_flag[n]+abs_level_gtx_flag[n][0]	
}	
/×Greater than Xscanpass(numGtXFlags=5) ×/	
lastScanPosPass2=-1	
for (n=0 ; n <= numSbCoeff-1 && RemCbs>=4 ; n++) {	
xC= (xS<<log2SbW) +DiagScanOrder[log2SbW][log2SbH][n][0]	
yC= (yS<<log2SbH) +DiagScanOrder[log2SbW][log2SbH][n][1]	
AbsLevelPass2[xC][yC]=AbsLevelPass1[xC][yC]	

for (j=1 ; j<5 ; j++) {	
if (abs_level_gtx_flag[n][j-1]) {	
abs_level_gtx_flag[n][j]	ae(v)
RemCcbs--	
}	
AbsLevelPass2[xC][yC]+=2×abs_level_gtx_flag[n][j]	
}	
lastScanPosPass2=n	
}	
/×remainderscanpass ×/	
for (n=0 ; n <= numSbCoeff-1 ; n++) {	
xC= (xS<<log2SbW) +DiagScanOrder[log2SbW][log2SbH][n][0]	
yC= (yS<<log2SbH) +DiagScanOrder[log2SbW][log2SbH][n][1]	
if ((n <= lastScanPosPass2 && AbsLevelPass2[xC][yC]>=10) (n>lastScanPosPass2 && n <= lastScanPosPass1 && AbsLevelPass1[xC][yC]>= 2) (n>lastScanPosPass1 && sb_coded_flag[xS][yS]))	
abs_remainder[n]	ae(v)
if (n <= lastScanPosPass2)	
AbsLevel[xC][yC]=AbsLevelPass2[xC][yC]+2×abs_remainder[n]	
elseif(n <= lastScanPosPass1)	
AbsLevel[xC][yC]=AbsLevelPass1[xC][yC]+2×abs_remainder[n]	
else{ /×bypass ×/	
AbsLevel[xC][yC]=abs_remainder[n]	
if (abs_remainder[n])	
coeff_sign_flag[n]	ae(v)
}	
if (BdpcmFlag[x0][y0][cIdx]==0 && n <= lastScanPosPass1) {	
absLeftCoeff=xC>0 ? AbsLevel[xC-1][yC] : 0	
absAboveCoeff=yC>0 ? AbsLevel[xC][yC-1] : 0	
predCoeff=Max (absLeftCoeff , absAboveCoeff)	
if (AbsLevel[xC][yC]==1 && predCoeff >0)	
AbsLevel[xC][yC]=predCoeff	
elseif (AbsLevel[xC][yC]>0 && AbsLevel[xC][yC] <= predCoeff)	
AbsLevel[xC][yC]—	
}	
TransCoeffLevel[x0][y0][cIdx][xC][yC]= (1-2×coeff_sign_flag[n]) × AbsLevel[xC][yC]	
}	
}	
}	

进入 residual_ts_coding (x0, y0, log2TbWidth, log2TbHeight, cIdx) 函数后, 需要确定一些块尺寸的信息。然后确定 RemCcbs 的值, 即使用等式 $\text{RemCcbs} = ((1 < (\log_2 \text{TbWidth} + \log_2 \text{TbHeight})) \times 7) >> 2$ 确定。RemCcbs 就确定了当前块中使用上下文模式编码的语法元素的个数, 本申请实施例可以理解 RemCcbs 为 remaining context coded binaries 的意思, 也就是剩余的上下文模式编码的二元符号的个数。

对每一个子块，确定当前子块是否需要编码。如果需要编码，与上述 RRC 不同，TSRC 方法中把一个子块中上下文模式编码的语法元素放在了两轮中，每个系数在第一轮和第二轮中分别最多处理 4 个上下文模式编码的语法元素。旁路模式编码的语法元素放在了后面。在第一轮和第二轮中，每次处理完一个上下文模式编码的语法元素 remBinsPass1 就会减 1。如果一个系数足够大，在第一轮和第二轮处理完上下文模式编码的几个语法元素后还需要处理剩余值，即 abs_remainder。而如果 remBinsPass1 已经足够小了（不满足 remBinsPass1>=4），前两轮就会结束，剩下的系数就会直接用旁路模式处理，这里仍然是 abs_remainder。

简言之，在相关技术中，已有的这种系数编码方法对于目前常用的视频，如消费类的视频，有很好的压缩效率。消费类视频通常每个像素的位宽为 8 位或者 10 位，消费类视频通常码率不会太高，通常为几兆每秒（MB/s）或更低。但是有一些应用的视频的像素需要更高位宽，如每个像素的位宽为 12 位、14 位或 16 位或更多。更高的位宽通常会带来更大的系数，更多的非零系数，从而带来更高的码率。有一些应用的视频需要更高的质量，更高的质量通常也会带来更大的系数，更多的非零系数，从而带来更高的码率。更高的码率对解码器的处理能力如吞吐量就要有更高的要求。

高位宽、高质量、高码率的视频（三高视频）通常较低位宽、低质量、低码率的视频（常规视频）需要编解码的系数更多而且系数更大。比如三高视频中一个块中需要编解码的系数比常规视频中同样大小的一个块中需要编解码的系数多得多。因为常规视频的块中经过预测、变换、量化之后的系数很多都变成了 0，而三高视频的块中经过预测、变换、量化之后的系数很多仍然是非 0 系数。常规视频的块中经过预测、变换、量化之后的系数中需要编码的系数中有很大比例是 0，因而用最后一个非零系数位置（LastSignificantCoeffX，LastSignificantCoeffY）来区分是否要编码的系数区域是非常有效的。即使是在最后一个非零系数位置之前的系数仍让有很大比例是 0，因而用子块是否编码的标志 sb_coded_flag 来进一步区分当前子块是否需要编码是非常有效的。但是在当前块中的非零系数非常多，甚至绝大多数系数甚至所有系数都是非零系数时，上述最后一个非零系数位置以及子块是否编码的标志就不会过滤掉太多非零系数，而且本身在码流中编码非零系数位置以及子块是否编码的标志等会占用一定的开销，会产生浪费。

另一方面，所述最后一个非零系数位置以及子块是否编码的标志等都是用上下文模式编码的，上下文模式编码较旁路模式更复杂，处理这些信息还会影响软件及硬件编解码的速度、吞吐量。

又一方面，目前的最后一个非零系数位置（LastSignificantCoeffX，LastSignificantCoeffY）的编码方法是编码最后一个非零系数位置的坐标。在常规视频中，由于非零系数大多集中在左上角而右下角的大片区域是 0，LastSignificantCoeffX 和 LastSignificantCoeffY 的值通常较小。而在三高视频中，右下角也会出现大量非零系数，这就导致 LastSignificantCoeffX 和 LastSignificantCoeffY 的值通常较大，从而在码流中编码较大的值会带来更大的开销。此外，还有一种可能是在无损压缩时使用本方法，因为无损压缩时不能使用量化，这时候通常系数会比较多，比较大。这时候使用已有的相关方案可能会带来更大的开销，产生浪费，甚至影响编解码的速度和吞吐量。

本申请实施例提供了一种系数解码方法，应用于解码器。解析码流，获取视频标识信息；当视频标识信息指示视频满足预设条件时，解析所述码流，获取最后一个非零系数位置翻转标识信息和最后一个非零系数的坐标信息；当最后一个非零系数位置翻转标识信息指示当前块使用最后一个非零系数位置翻转时，对最后一个非零系数的坐标信息进行计算，确定最后一个非零系数的位置；按照预设扫描顺序对最后一个非零系数的位置之前的所有系数进行解码，确定所述当前块的系数。

本申请实施例还提供了一种系数编码方法，应用于编码器。确定视频标识信息以及最后一个非零系数的位置；当视频标识信息指示视频满足预设条件时，确定最后一个非零系数位置翻转标识信息；根据最后一个非零系数的位置以及最后一个非零系数位置翻转标识信息，确定最后一个非零系数的坐标信息；按照预设扫描顺序对最后一个非零系数的位置之前的所有系数进行编码，并将编码后得到的比特信息、视频标识信息以及最后一个非零系数的坐标信息写入码流。

这样，在高位宽、高码率、高质量或无损的视频编解码场景中，由于系数分布规律与常规视频场景不同，在系数编码中通过减少甚至消除上下文模式编码的语法元素个数，比如关于最后一个非零系数位置、子块编码标识等语法元素，甚至还可以在最后一个非零系数的坐标信息的值偏大时进行坐标变换，从而可以减少码流中编码所带来的开销，进而提高系数编码的吞吐量和编解码速度；另外，由于减少或消除的语法元素在高位宽、高码率、高质量或无损的视频编解码中的影响较小，还能够提升压缩效率。

下面将结合附图对本申请各实施例进行详细说明。

参见图 8A，其示出了本申请实施例提供的一种编码器的系统组成框图示例。如图 8A 所示，编码器 100 可以包括：分割单元 101、预测单元 102、第一加法器 107、变换单元 108、量化单元 109、反量

化单元 110、逆变换单元 111、第二加法器 112、滤波单元 113、解码图片缓存 (Decoded Picture Buffer, DPB) 单元 114 和熵编码单元 115。这里, 编码器 100 的输入可以是由一系列图片或者一张静态图片组成的视频, 编码器 100 的输出可以用于表示输入视频的压缩版本的比特流 (也可以称为“码流”)。

其中, 分割单元 101 将输入视频中的图片分割成一个或多个编码树单元 (Coding Tree Units, CTUs)。分割单元 101 将图片分成多个图块 (或称为瓦片, tiles), 还可以进一步将一个 tile 分成一个或多个砖块 (bricks), 这里, 一个 tile 或者一个 brick 中可以包括一个或多个完整的和/或部分的 CTUs。另外, 分割单元 101 可以形成一个或多个切片 (slices), 其中一个 slice 可以包括图片中按照栅格顺序排列的一个或多个 tiles, 或者覆盖图片中矩形区域的一个或多个 tiles。分割单元 101 还可形成一个或多个子图片, 其中, 一个子图片可以包括一个或多个 slices、tiles 或 bricks。

在编码器 100 的编码过程中, 分割单元 101 将 CTU 传送到预测单元 102。通常, 预测单元 102 可以由块分割单元 103、运动估计 (Motion Estimation, ME) 单元 104、运动补偿 (Motion Compensation, MC) 单元 105 和帧内预测单元 106 组成。具体地, 块分割单元 103 迭代地使用四叉树分割、二叉树分割和三叉树分割而进一步将输入 CTU 划分成更小的编码单元 (Coding Units, CUs)。预测单元 102 可使用 ME 单元 104 和 MC 单元 105 获取 CU 的帧间预测块。帧内预测单元 106 可使用包括 MIP 模式的各种帧内预测模式获取 CU 的帧内预测块。在示例中, 率失真优化的运动估计方式可被 ME 单元 104 和 MC 单元 105 调用以获取帧间预测块, 以及率失真优化的模式确定方式可被帧内预测单元 106 调用以获取帧内预测块。

预测单元 102 输出 CU 的预测块, 第一加法器 107 计算分割单元 101 的输出中的 CU 和 CU 的预测块之间的差值, 即残差 CU。变换单元 108 读取残差 CU 并对残差 CU 执行一个或多个变换操作以获取系数。量化单元 109 对系数进行量化并输出量化系数 (即 levels)。反量化单元 110 对量化系数执行缩放操作以输出重构系数。逆变换单元 111 执行对应于变换单元 108 中的变换的一个或多个逆变换并输出重构残差。第二加法器 112 通过使重构残差和来自预测单元 102 的 CU 的预测块相加而计算出重构 CU。第二加法器 112 还将其输出发送到预测单元 102 以用作帧内预测参考。在图片或子图片中的所有 CU 被重构之后, 滤波单元 113 对重构图片或子图片执行环路滤波。这里, 滤波单元 113 包含一个或多个滤波器, 例如去方块滤波器、采样自适应偏移 (Sample Adaptive Offset, SAO) 滤波器、自适应环路滤波器 (Adaptive Loop Filter, ALF)、亮度映射和色度缩放 (Luma Mapping with Chroma Scaling, LMCS) 滤波器以及基于神经网络的滤波器等。或者, 当滤波单元 113 确定 CU 不用作其它 CU 编码时的参考时, 滤波单元 113 对 CU 中的一个或多个目标像素执行环路滤波。

滤波单元 113 的输出是解码图片或子图片, 这些解码图片或子图片缓存至 DPB 单元 114。DPB 单元 114 根据时序和控制信息输出解码图片或子图片。这里, 存储在 DPB 单元 114 中的图片还可用作预测单元 102 执行帧间预测或帧内预测的参考。最后熵编码单元 115 将来自编码器 100 中解码图片所必需的参数 (比如控制参数和补充信息等) 转换成二进制形式, 并根据每个数据单元的语法结构将这样的二进制形式写入码流中, 即编码器 100 最终输出码流。

进一步地, 编码器 100 可以是具有第一处理器和记录计算机程序的第一存储器。当第一处理器读取并运行计算机程序时, 编码器 100 读取输入视频并生成对应的码流。另外, 编码器 100 还可以是具有一个或多个芯片的计算设备。在芯片上实现为集成电路的这些单元具有与图 8A 中相应单元类似的连接和数据交换功能。

参见图 8B, 其示出了本申请实施例提供的一种解码器的系统组成框图示例。如图 8B 所示, 该解码器 200 可以包括: 解析单元 201、预测单元 202、反量化单元 205、逆变换单元 206、加法器 207、滤波单元 208 和解码图片缓存单元 209。这里, 解码器 200 的输入是用于表示视频或者一张静态图片的压缩版本的比特流, 解码器 200 的输出可以是由一系列图片组成的解码视频或者一张解码的静态图片。

其中, 解码器 200 的输入码流可以是编码器 100 所生成的码流。解析单元 201 对输入码流进行解析并从输入码流获取语法元素的值。解析单元 201 将语法元素的二进制表示转换成数字值并将数字值发送到解码器 200 中的单元以获取一个或多个解码图片。解析单元 201 还可从输入码流解析一个或多个语法元素以显示解码图片。

在解码器 200 的解码过程中, 解析单元 201 将语法元素的值以及根据语法元素的值设置或确定的、用于获取一个或多个解码图片的一个或多个变量发送到解码器 200 中的单元。

预测单元 202 确定当前解码块 (例如 CU) 的预测块。这里, 预测单元 202 可以包括运动补偿单元 203 和帧内预测单元 204。具体地, 当指示帧间解码模式用于对当前解码块进行解码时, 预测单元 202 将来自解析单元 201 的相关参数传递到运动补偿单元 203 以获取帧间预测块; 当指示帧内预测模式 (包括基于 MIP 模式索引值指示的 MIP 模式) 用于对当前解码块进行解码时, 预测单元 202 将来自解析单元 201 的相关参数传递到帧内预测单元 204 以获取帧内预测块。

反量化单元 205 具有与编码器 100 中的反量化单元 110 相同的功能。反量化单元 205 对来自解析单元 201 的量化系数（即 levels）执行缩放操作以获取重构系数。

逆变换单元 206 具有与编码器 100 中的逆变换单元 111 相同的功能。逆变换单元 206 执行一个或多个变换操作（即通过编码器 100 中的逆变换单元 111 执行的一个或多个变换操作的反操作）以获取重构残差。

加法器 207 对其输入（来自预测单元 202 的预测块和来自逆变换单元 206 的重构残差）执行相加操作以获取当前解码块的重构块。重构块还发送到预测单元 202 以用作在帧内预测模式下编码的其它块的参考。

在图片或子图片中的所有 CU 被重构之后，滤波单元 208 对重构图片或子图片执行环路滤波。滤波单元 208 包含一个或多个滤波器，例如去方块滤波器、采样自适应补偿滤波器、自适应环路滤波器、亮度映射和色度缩放滤波器以及基于神经网络的滤波器等。或者，当滤波单元 208 确定重构块不用作对其它块解码时的参考时，滤波单元 208 对重构块中的一个或多个目标像素执行环路滤波。这里，滤波单元 208 的输出是解码图片或子图片，解码图片或子图片缓存至 DPB 单元 209。DPB 单元 209 根据时序和控制信息输出解码图片或子图片。存储在 DPB 单元 209 中的图片还可用作通过预测单元 202 执行帧内预测或帧内预测的参考。

进一步地，解码器 200 可以是具有第二处理器和记录计算机程序的第二存储器。当第一处理器读取并运行计算机程序时，解码器 200 读取输入码流并生成对应的解码视频。另外，解码器 200 还可以是有一个或多个芯片的计算设备。在芯片上实现为集成电路的这些单元具有与图 8B 中相应单元类似的连接和数据交换功能。

还需要说明的是，当本申请实施例应用于编码器 100 时，“当前块”具体是指视频图像中的当前待编码的块（也可以简称为“编码块”）；当本申请实施例应用于解码器 200 时，“当前块”具体是指视频图像中的当前待解码的块（也可以简称为“解码块”）。

本申请的一实施例中，参见图 9，其示出了本申请实施例提供的一种系数解码方法的流程示意图。如图 9 所示，该方法可以包括：

S901：解析码流，获取视频标识信息。

需要说明的是，本申请实施例的系数解码方法应用于解码器。具体来说，基于图 8B 所示解码器 200 的组成结构，本申请实施例的系数解码方法主要应用于解码器 200 中的“解析单元 201”部分，针对该解析单元 201，可以采用基于上下文模型的自适应二进制算术编码模式或者旁路模式进行解码，得到相关标识信息（或语法元素）的取值，进而确定出当前块的系数。

还需要说明的是，视频标准中通常所说的系数编码可以包括编码和解码两部分，因此，系数编码包括编码器侧的系数编码方法和解码器侧的系数解码方法。本申请实施例则针对解码器侧的系数解码方法进行描述。

通常情况下，比如对于常规视频而言，系数解码方法与相关技术中的已有方法相同；但是针对某种情况，比如高位宽或高质量或高码率或无损压缩的视频编解码场景，本申请实施例可以修改最后一个非零系数的位置导出方法。

在本申请实施例中，首先需要判断当前的视频是否满足预设条件，可以用视频标识信息来表示。在一些实施例中，所述解析码流，获取视频标识信息，可以包括：

若视频标识信息的取值为第一值，则确定视频标识信息指示视频满足预设条件；或者，

若视频标识信息的取值为第二值，则确定视频标识信息指示视频不满足预设条件。

在这里，第一值为 1，第二值为 0。

需要说明的是，在另一具体的示例中，第一值还可以设置为 true，第二值还可以设置为 false。甚至在又一具体的示例中，第一值还可以设置为 0，第二值还可以设置为 1；或者，第一值还可以设置为 false，第二值还可以设置为 true。这里不作任何限定。

还需要说明的是，预设条件至少包括下述之一：高位宽、高质量、高码率、高帧率和无损压缩。

也就是说，相比较于常规视频，本申请实施例所述的视频具有高位宽、高质量、高码率、高帧率和无损压缩等特性。

进一步地，视频标识信息可以是序列（sequence）级的标志，甚至也可以是更高级别的标志，比如视频可用性信息（Video Usability Information, VUI）、补充增强信息（Supplemental Enhancement Information, SEI）等。判断视频是否满足预设条件，可以通过判断该视频是否满足高位宽、或者是否满足高码率、或者是否满足高质量、或者是否满足无损压缩等，下面以这四种情况为例分别进行描述。

- 在一些实施例中,当视频标识信息为高位宽标识信息时,该方法还可以包括:
若高位宽标识信息指示所述视频满足高位宽,则确定视频满足预设条件。
在一些实施例中,当视频标识信息为高码率标识信息时,该方法还可以包括:
若高码率标识信息指示所述视频满足高码率,则确定视频满足预设条件。
5 在一些实施例中,当视频标识信息为高质量标识信息时,该方法还可以包括:
若高质量标识信息指示所述视频满足高质量,则确定视频满足预设条件。
在一些实施例中,当视频标识信息为无损压缩标识信息时,该方法还可以包括:
若无损压缩标识信息指示所述视频满足无损压缩,则确定视频满足预设条件。
示例性地,以序列级为例,视频标识信息可以是高位宽标识信息(用 `sps_high_bit_depth_flag` 表示),
10 用于指示当前的视频序列是否为高位宽序列;或者也可以替换为高码率标识信息(用 `sps_high_bit_rate_flag` 表示),用于指示当前的视频序列是否为高码率序列;或者也可以替换为其他指示高位宽、高码率、高质量或者无损压缩的标识信息,本申请实施例不作具体限定。
S902:当视频标识信息指示视频满足预设条件时,解析码流,获取最后一个非零系数位置翻转标识信息和最后一个非零系数的坐标信息。
15 需要说明的是,在视频标识信息指示视频满足预设条件时,这时候可以进一步解析码流,以获取最后一个非零系数位置翻转标识信息和最后一个非零系数的坐标信息。
对于最后一个非零系数的坐标信息而言,可以由 `last_sig_coeff_x_prefix`、`last_sig_coeff_x_suffix`、`last_sig_coeff_y_prefix`、`last_sig_coeff_y_suffix` 确定的。因此,在一些实施例中,所述解析码流,获取最后一个非零系数的坐标信息,可以包括:
20 解析码流,获取最后一个非零系数的水平坐标的前缀信息、最后一个非零系数的竖直坐标的前缀信息、最后一个非零系数的水平坐标的后缀信息和最后一个非零系数的竖直坐标的后缀信息;
根据最后一个非零系数的水平坐标的前缀信息和最后一个非零系数的水平坐标的后缀信息,确定最后一个非零系数的水平坐标;
根据最后一个非零系数的竖直坐标的前缀信息和最后一个非零系数的竖直坐标的后缀信息,确定最后
25 后一个非零系数的竖直坐标;
根据最后一个非零系数的水平坐标和最后一个非零系数的竖直坐标,确定所述最后一个非零系数的坐标信息。
需要说明的是,最后一个非零系数的水平坐标的前缀信息用 `last_sig_coeff_x_prefix` 表示,即确定当前块按照预设扫描顺序最后一个非零系数的水平(或列)坐标的前缀;最后一个非零系数的竖直坐标的前缀信息用 `last_sig_coeff_y_prefix` 表示,即确定当前块按照预设扫描顺序最后一个非零系数的竖直(或行)坐标的前缀;最后一个非零系数的水平坐标的后缀信息用 `last_sig_coeff_x_suffix` 表示,即确定当前块按照预设扫描顺序最后一个非零系数的水平(或列)坐标的后缀;最后一个非零系数的竖直坐标的后缀信息用 `last_sig_coeff_y_suffix` 表示,即确定当前块按照预设扫描顺序最后一个非零系数的竖直(或行)坐标的后缀。
30 还需要说明的是, `last_sig_coeff_x_prefix` 和 `last_sig_coeff_x_suffix` 确定的是最后一个非零系数的横坐标(即水平坐标), `last_sig_coeff_y_prefix` 和 `last_sig_coeff_y_suffix` 确定的是最后一个非零系数的纵坐标(即竖直坐标),也就得到了最后一个非零系数的坐标信息。
对于最后一个非零系数位置翻转标识信息而言,可以用 `reverse_last_sig_coeff_flag` 表示。在本申请实施例中,最后一个非零系数位置翻转标识信息至少可以为下述其中一项的标识信息:序列(sequence)
40 级、图像(picture)级、片(slice)级和块(block)级;甚至也可以为更高级别(如 VUI、SEI 等)的标识信息,这里不作任何限定。
也就是说, `reverse_last_sig_coeff_flag` 可能是一个序列级或者更高级别的标志,或者也可能是图像级的标志或者片级的标志或者块级的标志或者其他级别的标志。另外,块级的标志可以包括最大编码单元(LCU)级的标志或编码单元(CU)级的标志或者其他块级的标志,本申请实施例均不作任何限定。
45 在一些实施例中,该方法还可以包括:
若最后一个非零系数位置翻转标识信息的取值为第一值,则确定最后一个非零系数位置翻转标识信息指示当前块使用最后一个非零系数位置翻转;或者,
若最后一个非零系数位置翻转标识信息的取值为第二值,则确定最后一个非零系数位置翻转标识信息指示当前块不使用最后一个非零系数位置翻转。
50 也就是说,以第一值为 1,第二值为 0 为例,如果 `reverse_last_sig_coeff_flag` 的取值为 1,那么可以确定 `reverse_last_sig_coeff_flag` 指示当前块使用最后一个非零系数位置翻转;或者,如果 `reverse_last_sig_coeff_flag` 的取值为 0,那么可以确定 `reverse_last_sig_coeff_flag` 指示当前块不使用最后

一个非零系数位置翻转。

S903: 当最后一个非零系数位置翻转标识信息指示当前块使用最后一个非零系数位置翻转时, 对最后一个非零系数的坐标信息进行计算, 确定最后一个非零系数的位置。

5 S904: 按照预设扫描顺序对最后一个非零系数的位置之前的所有系数进行解码, 确定当前块的系数。

需要说明的是, 在本申请实施例中, 当最后一个非零系数位置翻转标识信息指示当前块使用最后一个非零系数位置翻转时, 可以确定最后一个非零系数的坐标信息为最后一个非零系数的位置相对于所述当前块的右下角位置之间的水平距离和竖直距离。

10 这时候, 在一些实施例中, 所述对最后一个非零系数的坐标信息进行计算, 确定最后一个非零系数的位置, 可以包括:

确定当前块的宽度和高度;

根据当前块的宽度与最后一个非零系数的位置相对于当前块的右下角位置之间的水平距离进行减法计算, 得到所述最后一个非零系数的水平坐标; 以及

15 根据当前块的高度与最后一个非零系数的位置相对于当前块的右下角位置之间的竖直距离进行减法计算, 得到最后一个非零系数的竖直坐标;

根据最后一个非零系数的水平坐标和最后一个非零系数的竖直坐标, 确定最后一个非零系数的位置。

需要说明的是, 最后一个非零系数的坐标信息通常为最后一个非零系数的位置相对于当前块的左上角位置之间的水平距离和竖直距离。对于常规视频来说, 由于非零系数大多数集中在左上角, 而右下角的大片区域为 0; 但是对于高位宽、高质量、高码率的视频编解码来说, 右下角也会出现大量的非零系数, 使得最后一个非零系数的坐标信息的值通常较大, 这时候为了节省开销, 系数编码的时候需要进行坐标变换(具体可以是坐标翻转计算, 即坐标翻转后最后一个非零系数的坐标信息为最后一个非零系数的位置相对于当前块的右下角位置之间的水平距离和竖直距离), 那么在系数解码时也需要进行坐标翻转计算, 再次翻转后可以恢复出最后一个非零系数的坐标信息为最后一个非零系数的位置相对于当前块的左上角位置之间的水平距离和竖直距离, 从而确定出最后一个非零系数的位置, 以便按照预设扫描顺序对当前块内的最后一个非零系数的位置之前的所有系数进行解码。

25 还需要说明的是, 这里的当前块可以是没有 zero-out 变换的块, 也可以是 zero-out 变换后的块。以 zero-out 变换后的块为例, 这时候当前块的宽度为 $1 < \log_2 \text{ZoTbWidth}$, 当前块的高度是 $1 < \log_2 \text{ZoTbHeight}$; 那么在 reverse_last_sig_coeff_flag 指示当前块使用最后一个非零系数位置翻转(即 reverse_last_sig_coeff_flag 的取值为 1)的情况下,

30 $\text{LastSignificantCoeffX} = (1 < \log_2 \text{ZoTbWidth}) - 1 - \text{LastSignificantCoeffX}$;

$\text{LastSignificantCoeffY} = (1 < \log_2 \text{ZoTbHeight}) - 1 - \text{LastSignificantCoeffY}$ 。

其中, 等式右边的 ($\text{LastSignificantCoeffX}$, $\text{LastSignificantCoeffY}$) 表示通过解码获得的最后一个非零系数的坐标信息, 等式左边的 ($\text{LastSignificantCoeffX}$, $\text{LastSignificantCoeffY}$) 表示最后一个非零系数的位置(也可以看作最后一个非零系数的目标坐标信息)。

35 在本申请实施例中, 在 reverse_last_sig_coeff_flag 的取值为 0 的情况下, 在一些实施例中, 该方法还可以包括:

当最后一个非零系数位置翻转标识信息指示当前块不使用最后一个非零系数位置翻转时, 确定最后一个非零系数的坐标信息为最后一个非零系数的位置相对于当前块的左上角位置之间的水平距离和竖直距离;

40 根据最后一个非零系数的位置相对于当前块的左上角位置之间的水平距离和竖直距离, 确定最后一个非零系数的位置。

需要说明的是, 如果 reverse_last_sig_coeff_flag 指示当前块不使用最后一个非零系数位置翻转, 那么通过解码获取的最后一个非零系数的坐标信息即可以看作最后一个非零系数的目标坐标信息。在本申请实施例中, 最后一个非零系数的目标坐标信息为最后一个非零系数的位置相对于当前块的左上角位置之间的水平距离和竖直距离。

45 进一步地, 在一些实施例中, 该方法还可以包括:

当最后一个非零系数位置翻转标识信息指示当前块不使用最后一个非零系数位置翻转时, 根据最后一个非零系数的坐标信息直接确定最后一个非零系数的位置;

按照预设扫描顺序对最后一个非零系数的位置之前的所有系数进行解码, 确定当前块的系数。

50 需要说明的是, 预设扫描顺序可以是对角线、Zigzag、水平、垂直、4×4 子块扫描或者任何其它扫描顺序, 本申请实施例不作任何限定。

还需要说明的是, 在获取 reverse_last_sig_coeff_flag 之后, 如果 reverse_last_sig_coeff_flag 的取值

为 1, 即需要使用最后一个非零系数位置翻转, 那么在解码得到最后一个非零系数的坐标信息之后, 需要对最后一个非零系数的坐标信息进行计算, 确定出最后一个非零系数的位置; 然后按照预设扫描顺序对最后一个非零系数的位置之前的所有系数进行解码。如果 `reverse_last_sig_coeff_flag` 的取值为 0, 即不需要使用最后一个非零系数位置翻转, 那么在解码得到最后一个非零系数的坐标信息之后, 可以根据最后一个非零系数的坐标信息直接确定出最后一个非零系数的位置; 然后按照预设扫描顺序对最后一个非零系数的位置之前的所有系数进行解码。

如此, 针对某种情况, 在系数编码时, 本申请实施例提供一种修改最后一个非零系数的位置导出方法。也就是说通常情况下, 系数编解码方法还是与相关技术的已有方法一样。某种情况, 比如可以是指高位宽或高质量或高码率的视频或无损压缩的视频编解码。通常情况下, `last_sig_coeff_x_prefix`、`last_sig_coeff_x_suffix` 编码的是最后一个非零系数位置的横坐标, 也就是相对于当前块左上角的水平距离; `last_sig_coeff_y_prefix`、`last_sig_coeff_y_suffix` 编码的是最后一个非零系数位置的纵坐标, 也就是相对于当前块左上角的竖直距离, 如图 10A 所示。而在高位宽或高质量或高码率或无损压缩的视频编解码的情况下, 最后一个非零系数的位置普遍接近于当前块所有可能是非 0 系数的区域的右下角。在这种情况下, `last_sig_coeff_x_prefix`、`last_sig_coeff_x_suffix` 编码的是最后一个非零系数位置相对于当前块所有可能是非 0 系数的区域的右下角的水平距离; `last_sig_coeff_y_prefix`、`last_sig_coeff_y_suffix` 编码的是最后一个非零系数位置相对于当前块所有可能是非 0 系数的区域的右下角的竖直距离, 如图 10B 所示。例如, 如果当前块所有可能是非 0 系数的区域为 $(0, 0)$ 到 $((1 < \log 2ZotbWidth)-1, (1 < \log 2ZotbHeight)-1)$ 的矩形区域, 那么 `last_sig_coeff_x_prefix`、`last_sig_coeff_x_suffix` 编码的是最后一个非零系数位置相对于当前块 $((1 < \log 2ZotbWidth)-1, (1 < \log 2ZotbHeight)-1)$ 的水平距离。`last_sig_coeff_y_prefix`、`last_sig_coeff_y_suffix` 编码的是最后一个非零系数位置相对于当前块 $((1 < \log 2ZotbWidth)-1, (1 < \log 2ZotbHeight)-1)$ 的竖直距离。

针对语义的修改如下:

当前块中按预设扫描顺序最后一个非零系数的水平 (或列) 坐标的值 `LastSignificantCoeffX` 按如下方法导出:

如果 `last_sig_coeff_x_suffix` 不存在, 那么,

$$\text{LastSignificantCoeffX} = \text{last_sig_coeff_x_prefix};$$
 否则 (`last_sig_coeff_x_suffix` 存在),

$$\text{LastSignificantCoeffX} = (1 < ((\text{last_sig_coeff_x_prefix} > 1) - 1)) * (2 + (\text{last_sig_coeff_x_prefix} \& 1)) + \text{last_sig_coeff_x_suffix};$$

如果 `reverse_last_sig_coeff_flag` 的取值为 1, 那么

$$\text{LastSignificantCoeffX} = (1 < \log 2ZotbWidth) - 1 - \text{LastSignificantCoeffX};$$
 当前块中按扫描顺序最后一个非零系数的竖直 (或行) 坐标的值 `LastSignificantCoeffY` 按如下方法导出:

如果 `last_sig_coeff_y_suffix` 不存在, 那么,

$$\text{LastSignificantCoeffY} = \text{last_sig_coeff_y_prefix};$$
 否则 (`last_sig_coeff_y_suffix` 存在):

$$\text{LastSignificantCoeffY} = (1 < ((\text{last_sig_coeff_y_prefix} > 1) - 1)) * (2 + (\text{last_sig_coeff_y_prefix} \& 1)) + \text{last_sig_coeff_y_suffix};$$

如果 `reverse_last_sig_coeff_flag` 的取值为 1, 那么

$$\text{LastSignificantCoeffY} = (1 < \log 2ZotbHeight) - 1 - \text{LastSignificantCoeffY}.$$

其中, `reverse_last_sig_coeff_flag` 是最后一个非零系数位置翻转标志, 表示最后一个非零系数的位置是否需要翻转。如果 `reverse_last_sig_coeff_flag` 的取值为 1, 表示最后一个非零系数的位置需要翻转; 否则, 表示最后一个非零系数的位置不需要翻转。

还需要说明的是, `reverse_last_sig_coeff_flag` 可能是一个序列 (sequence) 级或者更高级别的标志, 也可能是图像 (picture) 级的标志或片 (slice) 级的标志或块 (block) 级的标志或其他级别的标志, 块级的标志包括最大编码单元 (LCU) 级的标志或编码单元 (CU) 级的标志或其他块级的标志。

另外, `reverse_last_sig_coeff_flag` 可能依赖于一些其他的标志, 比如高位宽标识信息或高码率标识信息等。即当高位宽标识信息或高码率标识信息的取值为 1 时, 这时候需要解码 `reverse_last_sig_coeff_flag` 标志; 否则, 不需要解码 `reverse_last_sig_coeff_flag` 标志。

在一种具体的示例中, 以序列级为例, 假定有一个序列级的标志 `sps_high_bit_depth_flag` 指示当前的视频序列是否为高位宽序列。如果 `sps_high_bit_depth_flag` 的取值为 1, 那么表示当前的视频序列是高位宽序列; 否则, 表示当前的视频序列不是高位宽序列。在序列级, 如果 `sps_high_bit_depth_flag` 的

取值为 1, 那么需要解码 `sps_reverse_last_sig_coeff_flag`。这里, `sps_reverse_last_sig_coeff_flag` 是当前序列的最后一个非零系数位置翻转标志。如果 `sps_reverse_last_sig_coeff_flag` 的取值为 1, 那么表示当前序列内的块使用最后一个非零系数位置翻转; 否则 (即 `sps_reverse_last_sig_coeff_flag` 的取值为 0), 表示当前序列内的块不使用最后一个非零系数位置翻转。上述语法表中的 `reverse_last_sig_coeff_flag` 变更为 `sps_reverse_last_sig_coeff_flag`。

- 5 其语法元素如下 (Sequence parameter set RBSP syntax), 参见表 3。

表 3

...	
<code>sps_high_bit_depth_flag</code>	<code>u(1)</code>
<code>if(sps_high_bit_depth_flag)</code>	
<code>sps_default_last_coeff_enabled_flag</code>	<code>u(1)</code>
...	

- 10 在另一种具体的示例中, 以片 (slice) 级为例, 假定有一个序列级的标志 `sps_high_bit_depth_flag` 指示当前的视频序列是否是高位宽序列。如果 `sps_high_bit_depth_flag` 的取值为 1, 那么表示当前的视频序列是高位宽序列; 否则, 表示当前的视频序列不是高位宽序列。在 slice 级, 如果 `sps_high_bit_depth_flag` 的取值为 1, 那么需要解码 `sh_reverse_last_sig_coeff_flag`。这里, `sh_reverse_last_sig_coeff_flag` 是当前片的最后一个非零系数位置翻转标志。如果 `sh_reverse_last_sig_coeff_flag` 的取值为 1, 那么表示当前片内的块使用最后一个非零系数位置翻转; 否则 (即 `sh_reverse_last_sig_coeff_flag` 的取值为 0), 表示当前片内的块不使用最后一个非零系数位置翻转。上述语法表中的 `reverse_last_sig_coeff_flag` 变更为 `sh_reverse_last_sig_coeff_flag`。

- 15 其语法元素如下 (Slice header syntax), 参见表 4。

表 4

...	
<code>if(sps_high_bit_depth_flag)</code>	
<code>sh_reverse_last_sig_coeff_flag</code>	<code>u(1)</code>
...	

- 20 还可以理解, 在视频标识信息指示视频满足预设条件时, 这时候还可默认需要编码所有可能需要编码的系数, 也就是不再使用最后一个非零系数的位置, 而是按预设扫描顺序把当前块所有可能不为 0 的系数都扫描到; 因此, 本申请实施例还可以引入最后一个系数使能标识信息, 用于确定当前块是否使用最后一个系数位置。

在一些实施例中, 当视频标识信息指示视频满足预设条件时, 该方法还可以包括:

解析码流, 获取最后一个系数使能标识信息;

- 25 当最后一个系数使能标识信息指示当前块使用最后一个系数位置时, 按照预设扫描顺序对最后一个系数位置之前的所有系数进行解码, 确定当前块的系数。

需要说明的是, 对于最后一个系数使能标识信息而言, 可以用 `default_last_coeff_enabled_flag` 表示。在本申请实施例中, 最后一个系数使能标识信息至少可以为下述其中一项的标识信息: 序列级、图像级、片级和块级; 甚至也可以为更高级别 (如 VUI、SEI 等) 的标识信息, 这里不作任何限定。

- 30 也就是说, `default_last_coeff_enabled_flag` 可能是一个序列级或者更高级别的标志, 也可能是图像级的标志或者片级的标志或者块级的标志或者其他级别的标志。另外, 块级的标志可以包括最大编码单元 (LCU) 级的标志或编码单元 (CU) 级的标志或者其他块级的标志, 本申请实施例不作任何限定。

在一些实施例中, 该方法还可以包括:

若最后一个系数使能标识信息的取值为第一值, 则确定最后一个系数使能标识信息指示当前块使用最后一个系数位置; 或者,

- 35 若最后一个系数使能标识信息的取值为第二值, 则确定最后一个系数使能标识信息指示当前块不使用最后一个系数位置。

在这里, 第一值为 1, 第二值为 0。

- 40 需要说明的是, 在另一具体的示例中, 第一值还可以设置为 true, 第二值还可以设置为 false。甚至在又一具体的示例中, 第一值还可以设置为 0, 第二值还可以设置为 1; 或者, 第一值还可以设置为 false, 第二值还可以设置为 true。这里不作任何限定。

这样, 以第一值为 1, 第二值为 0 为例, 如果 `default_last_coeff_enabled_flag` 的取值为 1, 那么可以确定 `default_last_coeff_enabled_flag` 指示当前块使用最后一个系数位置; 或者, 如果 `default_last_coeff_enabled_flag` 的取值为 0, 那么可以确定 `default_last_coeff_enabled_flag` 指示当前块不使用最后一个系数

位置。

在当前块使用最后一个系数位置的情况下,这时候可以按照预设扫描顺序对最后一个系数位置之前的所有系数进行解码,进而确定出当前块的系数。

进一步地,在当前块不使用最后一个系数位置的情况下,即最后一个系数使能标识信息的取值为 0, 5 在一些实施例中,该方法还可以包括:

解析码流,获取最后一个非零系数的水平坐标的前缀信息、最后一个非零系数的竖直坐标的前缀信息、最后一个非零系数的水平坐标的后缀信息和最后一个非零系数的竖直坐标的后缀信息;

根据最后一个非零系数的水平坐标的前缀信息、最后一个非零系数的竖直坐标的前缀信息、最后一个非零系数的水平坐标的后缀信息和最后一个非零系数的竖直坐标的后缀信息,确定最后一个非零系数的位置; 10

按照预设扫描顺序对最后一个非零系数的位置之前的所有系数进行解码,确定当前块的系数。

需要说明的是,如果当前块不使用最后一个系数位置,那么这时候需要解码获取最后一个非零系数的位置。具体地,通过解析码流,获取 last_sig_coeff_x_prefix、last_sig_coeff_x_suffix、last_sig_coeff_y_prefix、last_sig_coeff_y_suffix; 15 然后根据 last_sig_coeff_x_prefix、last_sig_coeff_x_suffix、last_sig_coeff_y_prefix、last_sig_coeff_y_suffix,进而确定出最后一个非零系数的位置。否则,如果当前块使用最后一个系数位置,这时候不再需要确定最后一个非零系数的位置,也就不再需要解码获取 last_sig_coeff_x_prefix、last_sig_coeff_x_suffix、last_sig_coeff_y_prefix、last_sig_coeff_y_suffix。

还需要说明的是,如果当前块使用最后一个系数位置,那么可以按照预设扫描顺序对最后一个系数位置之前的所有系数进行解码;如果当前块不使用最后一个系数位置,那么可以按照预设扫描顺序对最后一个非零系数的位置之前的所有系数进行解码。这里,预设扫描顺序可以是 diagonal、Zigzag、水平、垂直、4x4 子块扫描或者任何其它扫描顺序,本申请实施例不作任何限定。 20

进一步地,对于最后一个系数位置而言,在一些实施例中,最后一个系数位置是当前块中所有可能不为零的系数所组成的矩阵的右下角位置;或者,最后一个系数位置是当前块根据预设扫描顺序执行扫描所有可能不为零的系数的最后位置。

需要注意的是,本申请实施例中的最后一个系数位置并不代表最后一个非零系数的位置。因为最后一个系数位置上的系数可能是 0,而最后一个非零系数的位置上的系数一定不是 0。 25

在一种特定示例中,该方法还可以包括:将最后一个非零系数的位置设置在最后一个系数位置。

也就是说,本申请实施例仍然可以使用最后一个非零系数的位置,这时候需要把最后一个非零系数的位置放在按预设扫描顺序的当前块所有可能不为 0 的系数的最后位置。

进一步地,最后一个系数位置可以用 (LastCoeffX, LastCoeffY) 表示,即当前块按照预设扫描顺序所有可能不为 0 的系数的最后位置。在一些实施例中,该方法还可以包括: 30

确定当前块在预设操作后得到变换块的宽度和高度;

根据变换块的宽度和高度进行坐标计算,得到变换块的右下角坐标信息;

根据变换块的右下角坐标信息,确定所述最后一个系数位置。

在这里,预设操作至少包括:强制置零 (zero-out) 操作。 35

需要说明的是, (LastCoeffX, LastCoeffY) 表示 zero-out 后的变换块的右下角坐标信息;其中, (LastCoeffX, LastCoeffY) 的导出方法如下:

$\text{LastCoeffX} = (1 \ll \log_2 \text{ZoTbWidth}) - 1$; $\text{LastCoeffY} = (1 \ll \log_2 \text{ZoTbHeight}) - 1$ 。

这样,如果 default_last_coeff_enabled_flag 的取值为 1,那么根据 (LastCoeffX, LastCoeffY) 即可 40 确定出最后一个系数位置。

在一种特定示例中,仍然使用最后一个非零系数的位置,这时候可以把最后一个非零系数的位置放在按照预设扫描顺序的当前块所有可能为 0 的系数的最后位置。在一些实施例中,该方法还可以包括:

当将最后一个非零系数的位置设置在最后一个系数位置时,根据变换块的右下角坐标信息,确定最后一个非零系数的位置。

也就是说,最后一个非零系数的位置可以用 (LastSignificantCoeffX, LastSignificantCoeffY) 表示, (LastSignificantCoeffX, LastSignificantCoeffY) 的导出方法如下: 45

$\text{LastSignificantCoeffX} = (1 \ll \log_2 \text{ZoTbWidth}) - 1$; $\text{LastSignificantCoeffY} = (1 \ll \log_2 \text{ZoTbHeight}) - 1$ 。

其中, (LastSignificantCoeffX, LastSignificantCoeffY) 表示 zero-out 后的变换块的右下角坐标信息。如果 default_last_coeff_enabled_flag 的取值为 1,那么根据 (LastSignificantCoeffX, LastSignificantCoeffY) 50 即可确定出最后一个非零系数的位置。

如此,针对某种情况,在系数编码时,默认需要编码所有可能需要编码的系数。也就是说通常情况下,系数编解码方法还是与相关技术中的已有方法一样。某种情况,比如可以是指高位宽或高质量或高

码率的视频或无损压缩的视频编解码。默认需要编码所有可能需要编码的系数，也就是说不再使用最后一个非零系数的位置，而是按预设扫描顺序把当前块所有可能不为 0 的系数都扫描到；或者说，把最后一个需要编码的系数位置放在按预设扫描顺序的当前块所有可能不为 0 的系数的最后位置。这个位置通常是指当前块所有可能不为 0 的系数所组成的矩阵的右下角位置。这里使用最后一个需要编码的系数位置而不是最后一个非零系数的位置。因为最后一个需要编码的系数位置上的系数可能是 0，而最后一个非零系数的位置上的系数一定不是 0。

一种特例是仍然使用最后一个非零系数的位置，这时候把最后一个非零系数的位置放在按预设扫描顺序的当前块所有可能不为 0 的系数的最后位置。

另外，按预设扫描顺序的当前块所有可能不为 0 的系数，是因为除了最后一个非零系数之外还有一些其他技术使得一个块中的某些系数默认为 0。例如，可以是上述提到的 zero-out。

针对语义的修改如表 5 所示。

表 5

...	
if(!default_last_coeff_enabled_flag){	
if(log2TbWidth > 0)	
last_sig_coeff_x_prefix	ae(v)
if(log2TbHeight > 0)	
last_sig_coeff_y_prefix	ae(v)
if(last_sig_coeff_x_prefix > 3)	
last_sig_coeff_x_suffix	ae(v)
if(last_sig_coeff_y_prefix > 3)	
last_sig_coeff_y_suffix	ae(v)
}	
...	

本申请实施例中，可以在解码最后一个非零系数所需的信息前加一个条件，即如果 default_last_coeff_enabled_flag 不成立（即 default_last_coeff_enabled_flag 的取值等于 0），那么才需要解码 last_sig_coeff_x_prefix、last_sig_coeff_y_prefix、last_sig_coeff_x_suffix、last_sig_coeff_y_suffix 等语法元素。如果 default_last_coeff_enabled_flag 成立（即 default_last_coeff_enabled_flag 的取值等于 1），那么不需要解码 last_sig_coeff_x_prefix、last_sig_coeff_y_prefix、last_sig_coeff_x_suffix、last_sig_coeff_y_suffix 等语法元素。

在这里，default_last_coeff_enabled_flag 是默认最后一个系数使能标志，用于指示是否使用默认最后一个系数。如果 default_last_coeff_enabled_flag 的取值为 1，表示使用默认最后一个系数位置，即把最后一个需要编码的系数位置放在按预设扫描顺序的当前块所有可能不为 0 的系数的最后位置；否则，表示不使用默认最后一个系数位置。

如果 default_last_coeff_enabled_flag 的取值为 1，那么默认最后一个系数位置（LastCoeffX，LastCoeffY）为当前块按预设扫描顺序所有可能不为 0 的系数的最后位置。按预设扫描顺序（LastCoeffX，LastCoeffY）之前的系数都需要扫描。在本申请实施例中，（LastCoeffX，LastCoeffY）导出方法如下：

LastCoeffX = (1 << log2ZoTbWidth) - 1; LastCoeffY = (1 << log2ZoTbHeight) - 1。

其中，（LastCoeffX，LastCoeffY）是 zero-out 后的变换块的右下角位置的坐标信息。

一种特例是仍然使用最后一个非零系数的位置，把最后一个非零系数的位置放在按预设扫描顺序的当前块所有可能不为 0 的系数的最后位置。在本申请实施例中，最后一个非零系数的位置（LastSignificantCoeffX，LastSignificantCoeffY）导出方法如下：

LastSignificantCoeffX = (1 << log2ZoTbWidth) - 1; LastSignificantCoeffY = (1 << log2ZoTbHeight) - 1。

其中，（LastSignificantCoeffX，LastSignificantCoeffY）是 zero-out 后的变换块的右下角位置的坐标信息。

还需要说明的是，default_last_coeff_enabled_flag 可能是一个序列（sequence）级或者更高级别的标志，也可能是图像（picture）级的标志或片（slice）级的标志或块（block）级的标志或其他级别的标志，块级的标志包括最大编码单元（LCU）级的标志或编码单元（CU）级的标志或其他块级的标志。

另外，default_last_coeff_enabled_flag 可能依赖于一些其他的标志，比如高位宽标识信息或高码率标识信息等。即当高位宽标识信息或高码率标识信息的取值为 1 时，需要解码 default_last_coeff_enabled_flag 标志；否则，不需要解码 default_last_coeff_enabled_flag 标志。

在一种具体的示例中，以序列级为例，假定有一个序列级的标志 sps_high_bit_depth_flag 指示当前

5 的视频序列是否为高位宽序列。如果 `sps_high_bit_depth_flag` 的取值为 1，那么表示当前的视频序列是高位宽序列；否则，表示当前的视频序列不是高位宽序列。在序列级，如果 `sps_high_bit_depth_flag` 的取值为 1，那么需要解码 `sps_default_last_coeff_enabled_flag`。这里，`sps_default_last_coeff_enabled_flag` 是当前序列的默认最后一个系数使能标志。如果 `sps_default_last_coeff_enabled_flag` 的取值为 1，那么表示当前序列内的块使用默认最后一个系数；否则（即 `sps_default_last_coeff_enabled_flag` 的取值为 0），表示当前序列内的块不使用默认最后一个系数。上述语法表中的 `default_last_coeff_enabled_flag` 变更为 `sps_default_last_coeff_enabled_flag`。

其语法元素如下（Sequence parameter set RBSP syntax），参见表 6。

表 6

...	
sps_high_bit_depth_flag	u(1)
if(sps_high_bit_depth_flag)	
sps_default_last_coeff_enabled_flag	u(1)
...	

10 在另一种具体的示例中，以片（slice）级为例，假定有一个序列级的标志 `sps_high_bit_depth_flag` 指示当前的视频序列是否为高位宽序列。如果 `sps_high_bit_depth_flag` 的取值为 1，那么表示当前的视频序列是高位宽序列；否则，表示当前的视频序列不是高位宽序列。在 slice 级，如果 `sps_high_bit_depth_flag` 的取值为 1，那么需要解码 `sh_default_last_coeff_enabled_flag`。这里，`sh_default_last_coeff_enabled_flag` 是当前片的默认最后一个系数使能标志。如果 `sh_default_last_coeff_enabled_flag` 的取值为 1，那么表示当前片内的块使用默认最后一个系数；否则（即 `sh_default_last_coeff_enabled_flag` 的取值为 0），表示当前片内的块不使用默认最后一个系数。上述语法表中的 `default_last_coeff_enabled_flag` 变更为 `sh_default_last_coeff_enabled_flag`。

其语法元素如下（Slice header syntax），参见表 7。

表 7

...	
if(sps_high_bit_depth_flag)	
sh_default_last_coeff_enabled_flag	u(1)
...	

20 还可以理解，在视频标识信息指示视频满足预设条件时，还可以扫描到的子块都默认需要编码，这时候就不需要在码流中传输 `sb_coded_flag`，即编码器/解码器均不需要处理这个标志，以加快编解码速度。因此，本申请实施例还可以引入子块默认编码标识信息，用于确定当前块内的待解码子块是否为默认编码。

在一些实施例中，当视频标识信息指示视频满足预设条件时，该方法还可以包括：

25 解析码流，获取子块默认编码标识信息；

当子块默认编码标识信息指示当前块内的待解码子块默认编码时，确定子块编码标识信息的取值为第一值，对待解码子块内的所有系数进行解码。

需要说明的是，对于子块默认标识信息而言，可以用 `default_sb_coded_flag` 表示。在本申请实施例中，子块默认编码标识信息至少为下述其中一项的标识信息：序列级、图像级、片级和块级；甚至也可以为更高级别（如 VUI、SEI 等）的标识信息，这里不作任何限定。

30 也就是说，`default_sb_coded_flag` 可能是一个序列级或者更高级别的标志，也可能是图像级的标志或者片级的标志或者块级的标志或者其他级别的标志。另外，块级的标志可以包括最大编码单元（LCU）级的标志或编码单元（CU）级的标志或者其他块级的标志，本申请实施例均不作任何限定。

在一些实施例中，该方法还可以包括：

35 若子块默认编码标识信息的取值为第一值，则确定子块默认编码标识信息指示待解码子块默认编码；或者，

若子块默认编码标识信息的取值为第二值，则确定子块默认编码标识信息指示待解码子块不默认编码。

在这里，第一值为 1，第二值为 0。

40 需要说明的是，在另一具体的示例中，第一值还可以设置为 true，第二值还可以设置为 false。甚至在又一具体的示例中，第一值还可以设置为 0，第二值还可以设置为 1；或者，第一值还可以设置为 false，第二值还可以设置为 true。这里不作任何限定。

这样，以第一值为 1，第二值为 0 为例，如果 `default_sb_coded_flag` 的取值为 1，那么可以确定

default_sb_coded_flag 指示待解码子块默认需要编码；或者，如果 default_sb_coded_flag 的取值为 0，那么可以确定 default_sb_coded_flag 指示待解码子块不默认需要编码。

在待解码子块默认需要编码的情况下，这时候 default_sb_coded_flag 的取值为 1，意味着 sb_coded_flag 的取值为 1，即不再需要解码 sb_coded_flag，此时默认需要对待解码子块内的所有系数进行解码。

进一步地，在待解码子块不默认需要编码的情况下，即 default_sb_coded_flag 的取值为 0，在一些实施例中，该方法还可以包括：

解析码流，获取子块编码标识信息；

当子块编码标识信息的取值为第一值时，对待解码子块内的所有系数进行解码。

需要说明的是，如果待解码子块不默认需要编码，那么这时候还需要解码获取子块编码标识信息；然后根据子块编码标识信息来确定是否对待解码子块内的所有系数进行解码。

进一步地，对于子块编码标识信息而言，该方法还可以包括：

若子块编码标识信息的取值为第一值，则确定对待解码子块内的所有系数进行解码；或者，

若子块编码标识信息的取值为第二值，则确定待解码子块内的所有系数均为零。

在本申请实施例中，子块编码标识信息可以用 sb_coded_flag 表示。以第一值为 1，第二值为 0 为例，如果 sb_coded_flag 的取值为 1，那么可以确定需要对待解码子块内的所有系数进行解码；或者，如果 sb_coded_flag 的取值为 0，那么可以确定不需要对待解码子块内的所有系数进行解码，这时候待解码子块内的所有系数均为零。

如此，针对某种情况，在系数编码时，扫描到的子块都默认需要编码，或者说扫描到的子块都默认包含非零系数。也就是说通常情况下，系数编码的方法还是与相关技术中的已有方法一样。某种情况，比如可以是指高位宽或高质量或高码率的视频或无损压缩的视频编解码。这种情况下非零系数很多，扫描到的子块几乎都是需要编码；或者说，扫描到的子块几乎都包含非零系数。这样就不需要在码流中传输 sb_coded_flag，编码器/解码器也就不需要处理这个标志，从而能够加快编解码速度。由于去掉了一个几乎没用的标志，此时还会带来一点压缩性能的提升。

针对语义的修改如表 8 所示。

表 8

...	
if(i<lastSubBlock && i>0) {	
if(!default_sb_coded_flag) {	
sb_coded_flag[xS][yS]	ae(v)
}	
inferSbDcSigCoeffFlag = 1	
}	
...	

其中，default_sb_coded_flag 是默认子块需要编码的标志。如果 default_sb_coded_flag 的取值为 1，那么可以确定 sb_coded_flag[xS][yS] 的取值为 1，这时候不需要再从码流中进行解码；否则（default_sb_coded_flag 的取值为 0），还需要从码流中解码 sb_coded_flag[xS][yS]。

还需要说明的是，default_sb_coded_flag 可能是一个序列（sequence）级或者更高级别的标志，也可能是图像（picture）级的标志或片（slice）级的标志或块（block）级的标志或其他级别的标志，块级的标志包括最大编码单元（LCU）级的标志或编码单元（CU）级的标志或其他块级的标志。

另外，default_sb_coded_flag 可能依赖于一些其他的标志，比如高位宽标识信息或高码率标识信息等。即当高位宽标识信息或高码率标识信息的取值为 1 时，需要解码 default_sb_coded_flag 标志；否则，不需要解码 default_sb_coded_flag 标志。

在一种具体的示例中，以序列级为例，假定有一个序列级的标志 sps_high_bit_depth_flag 指示当前的视频序列是否为高位宽序列。如果 sps_high_bit_depth_flag 的取值为 1，那么表示当前的视频序列是高位宽序列；否则，表示当前的视频序列不是高位宽序列。在序列级，如果 sps_high_bit_depth_flag 的取值为 1，那么需要解码 sps_default_sb_coded_flag。这里，sps_default_sb_coded_flag 是当前序列的默认子块需要编码的标志。如果 sps_default_sb_coded_flag 的取值为 1，那么表示当前序列内的块默认子块需要编码；否则（即 sps_default_sb_coded_flag 的取值为 0），表示当前序列内的块不默认子块需要编码。上述语法表中的 default_sb_coded_flag 变更为 sps_default_sb_coded_flag。

其语法元素如下（Sequence parameter set RBSP syntax），参见表 9。

表 9

...	
sps_high_bit_depth_flag	u(1)
if(sps_high_bit_depth_flag)	
sps_default_sb_coded_flag	u(1)
...	

在另一种具体的示例中，以片（slice）级为例，假定有一个序列级的标志 sps_high_bit_depth_flag 指示当前的视频序列是否为高位宽序列。如果 sps_high_bit_depth_flag 的取值为 1，那么表示当前的视频序列是高位宽序列；否则，表示当前的视频序列不是高位宽序列。在 slice 级，如果 sps_high_bit_depth_flag 的取值为 1，那么需要解码 sh_default_sb_coded_flag。这里，sh_default_sb_coded_flag 是当前片的默认子块需要编码的标志。如果 sh_default_sb_coded_flag 的取值为 1，那么表示当前片内的块默认子块需要编码；否则（即 sh_default_sb_coded_flag 的取值为 0），表示当前片内的块不默认子块需要编码。上述语法表中的 default_sb_coded_flag 变更为 sh_default_sb_coded_flag。

其语法元素如下（Slice header syntax），参见表 10。

表 10

...	
if(sps_high_bit_depth_flag)	
sh_reverse_last_sig_coeff_flag	u(1)
...	

可以理解地，本申请实施例涉及到三种优化方法，分别如下：

方法一：针对某种情况，在系数编码时，默认需要编码所有可能需要编码的系数。也就是说通常情况下，系数编码的方法还是与相关技术中的已有方法一样。某种情况，比如可以是指高位宽或高质量或高码率的视频或无损压缩的视频编解码。默认需要编码所有可能需要编码的系数，也就是说不再使用最后一个非零系数的位置，而是按预设扫描顺序把当前块所有可能不为 0 的系数都扫描到；或者说，把最后一个需要编码的系数位置放在按预设扫描顺序的当前块所有可能不为 0 的系数的最后位置。这里使用最后一个需要编码的系数位置而不是最后一个非零系数的位置。因为最后一个需要编码的系数位置上的系数可能是 0，而最后一个非零系数的位置上的系数一定不是 0。

另外，一种特定示例是仍然使用最后一个非零系数的位置，把最后一个非零系数的位置放在按预设扫描顺序的当前块所有可能不为 0 的系数的最后位置。

方法二：针对某种情况，在系数编码时，修改最后一个非零系数位置的导出方法。也就是说通常情况下，系数编码的方法还是与相关技术中的已有方法一样。某种情况，比如可以是指高位宽或高质量或高码率的视频或无损压缩的视频编解码。通常情况下，last_sig_coeff_x_prefix、last_sig_coeff_x_suffix 编码的是最后一个非零系数位置的横坐标，也就是相对于当前块左上角的水平距离；last_sig_coeff_y_prefix、last_sig_coeff_y_suffix 编码的是最后一个非零系数位置的纵坐标，也就是相对于当前块左上角的竖直距离。而在高位宽或高质量或高码率的视频或无损压缩的视频编解码的情况下，最后一个非零系数的位置普遍接近于当前块所有可能是非 0 系数的区域的右下角。在这种情况下，last_sig_coeff_x_prefix、last_sig_coeff_x_suffix 编码的是最后一个非零系数的位置相对于当前块所有可能是非 0 系数的区域的右下角的水平距离；last_sig_coeff_y_prefix、last_sig_coeff_y_suffix 编码的是最后一个非零系数的位置相对于当前块所有可能是非 0 系数的区域的右下角的竖直距离。例如，如果当前块所有可能是非 0 系数的区域为（0，0）到（ $(1 < \log 2ZoTbWidth)-1$ ， $(1 < \log 2ZoTbHeight)-1$ ）的矩形区域，那么 last_sig_coeff_x_prefix、last_sig_coeff_x_suffix 编码的是最后一个非零系数的位置相对于当前块（ $(1 < \log 2ZoTbWidth)-1$ ， $(1 < \log 2ZoTbHeight)-1$ ）的水平距离；last_sig_coeff_y_prefix、last_sig_coeff_y_suffix 编码的是最后一个非零系数的位置相对于当前块（ $(1 < \log 2ZoTbWidth)-1$ ， $(1 < \log 2ZoTbHeight)-1$ ）的竖直距离。

方法三：针对某种情况，在系数编码时，扫描到的子块都默认需要编码；或者说，扫描到的子块都默认包含非零系数。也就是说通常情况下，系数编码的方法还是与相关技术中的已有方法一样。某种情况，比如可以是指高位宽或高质量或高码率的视频或无损压缩的视频编解码。这种情况下非零系数很多，扫描到的子块几乎都是需要编码，或者说扫描到的子块几乎都包含非零系数。这时候就不需要在码流中传输 sb_coded_flag，编码器/解码器也就不需要处理这个标志。

针对上述三种方法，在高位宽、高码率、高质量或无损的视频编解码场景中，由于系数分布规律与普通视频场景不同，在系数编码中通过减少甚至消除上下文模式编码的语法元素的个数，比如关于最后一个非零系数位置、子块编码标识等语法元素，可以提高系数编码的吞吐量以及编解码速度；同时由于

上述这些标志在高位宽、高码率、高质量或无损的视频编解码中的作用不大,不使用这些标志不会降低压缩效率反而还可以一定程度地提升压缩效率。

除此之外,在本申请实施例中,以序列级为例,序列级标志 `sps_high_bit_depth_flag` 指示当前的视频序列是否为高位宽序列的标志,也可以替换为 `sps_high_bit_rate_flag` 指示当前的视频序列是否为高码率序列的标志;甚至也可以替换为其他指示高位宽、高码率、高质量或无损编码等的标志。

还需要注意的是,本申请实施例的系数解码方法均是以视频中所有分量统一使用该技术为例,所有分量是指如 RGB 格式视频的 R、G、B,或者 YUV 格式的 Y、U、V (Y, Cb, Cr) 等。本申请实施例的系数解码方法也可以仅用于某一个分量,如仅用于 YUV 格式的 Y 分量。本申请实施例的系数解码方法也可以分别用于各个分量,即每一个分量都可单独控制开关。

本实施例提供了一种系数解码方法,应用于解码器。通过解析码流,获取视频标识信息;当视频标识信息指示视频满足预设条件时,解析所述码流,获取最后一个非零系数位置翻转标识信息和最后一个非零系数的坐标信息;当最后一个非零系数位置翻转标识信息指示当前块使用最后一个非零系数位置翻转时,对最后一个非零系数的坐标信息进行计算,确定最后一个非零系数的位置;按照预设扫描顺序对最后一个非零系数的位置之前的所有系数进行解码,确定所述当前块的系数。这样,在高位宽、高码率、高质量或无损的视频编解码场景中,由于系数分布规律与常规视频场景不同,在系数编码中通过减少甚至消除上下文模式编码的语法元素个数,比如关于最后一个非零系数位置、子块编码标识等语法元素,甚至还可以在最后一个非零系数的坐标信息的值偏大时进行坐标变换,从而可以减少码流中编码所带来的开销,进而提高系数编码的吞吐量和编解码速度;另外,由于减少或消除的语法元素在高位宽、高码率、高质量或无损的视频编解码中的影响较小,还能够提升压缩效率。

本申请的另一实施例中,参见图 11,其示出了本申请实施例提供的一种系数编码方法的流程示意图。如图 11 所示,该方法可以包括:

S1101: 确定视频标识信息以及最后一个非零系数的位置。

S1102: 当视频标识信息指示视频满足预设条件时,确定最后一个非零系数位置翻转标识信息。

S1103: 根据最后一个非零系数的位置以及最后一个非零系数位置翻转标识信息,确定最后一个非零系数的坐标信息。

S1104: 按照预设扫描顺序对所述最后一个非零系数的位置之前的所有系数进行编码,并将编码后得到的比特信息、视频标识信息以及最后一个非零系数的坐标信息写入码流。

需要说明的是,本申请实施例的系数编码方法应用于编码器。具体来说,基于图 8A 所示编码器 100 的组成结构,本申请实施例的系数编码方法主要应用于编码器 100 中的“熵编码单元 115”部分,针对该熵编码单元 115,可以采用基于上下文模型的自适应二进制算术编码模式或者旁路模式对相关标识信息(或语法元素)进行熵编码,然后写入码流。

还需要说明的是,视频标准中通常所说的系数编码可以包括编码和解码两部分,因此,系数编码包括编码器侧的系数编码方法和解码器侧的系数解码方法。本申请实施例则针对编码器侧的系数编码方法进行描述。

通常情况下,比如对于常规视频而言,系数编码方法与相关技术中的已有方法相同;但是针对某种情况,比如高位宽或高质量或高码率或无损压缩的视频编解码,本申请实施例可以修改最后一个非零系数的位置导出方法,这时候本申请实施例需要引入视频标识信息和最后一个非零系数位置翻转标识信息,以确定出最后一个非零系数的位置,然后按照预设扫描顺序对当前块内的最后一个非零系数的位置之前的所有系数进行编码。

其中,本申请实施例首先需要判断当前的视频是否满足预设条件,可以用视频标识信息来表示。在一些实施例中,所述确定视频标识信息,可以包括:

若视频满足预设条件,则确定视频标识信息的取值为第一值;或者,

若视频不满足预设条件,则确定视频标识信息的取值为第二值。

在这里,第一值为 1,第二值为 0。

需要说明的是,在另一具体的示例中,第一值还可以设置为 true,第二值还可以设置为 false。甚至在又一具体的示例中,第一值还可以设置为 0,第二值还可以设置为 1;或者,第一值还可以设置为 false,第二值还可以设置为 true。这里不作任何限定。

还需要说明的是,预设条件至少包括下述之一:高位宽、高质量、高码率、高帧率和无损压缩。

进一步地,视频标识信息可以是序列(sequence)级的标志,甚至也可以是更高级别的标志(比如 VUI、SEI 等)。判断视频是否满足预设条件,还可以通过判断该视频是否满足高位宽、或者是否满足高码率、或者是否满足高质量、或者是否满足无损压缩等,下面以这四种情况为例分别进行描述。

- 在一些实施例中,当视频标识信息为高位宽标识信息时,该方法还可以包括:
若视频满足高位宽,则确定高位宽标识信息指示视频满足预设条件。
- 在一些实施例中,当视频标识信息为高码率标识信息时,该方法还可以包括:
若视频满足高码率,则确定高码率标识信息指示视频满足预设条件。
- 5 在一些实施例中,当视频标识信息为高质量标识信息时,该方法还可以包括:
若视频满足高质量,则确定高质量标识信息指示视频满足预设条件。
- 在一些实施例中,当视频标识信息为无损压缩标识信息时,该方法还可以包括:
若视频满足无损压缩,则确定无损压缩标识信息指示视频满足预设条件。
- 示例性地,以序列级为例,视频标识信息可以是高位宽标识信息(用 `sps_high_bit_depth_flag` 表示),
10 用于指示当前的视频序列是否为高位宽序列;或者也可以替换为高码率标识信息(用 `sps_high_bit_rate_flag` 表示),用于指示当前的视频序列是否为高码率序列;或者也可以替换为其他指示高位宽、高码率、高质量或者无损压缩的标识信息,本申请实施例不作具体限定。
- 进一步地,对于最后一个非零系数位置翻转标识信息而言,所述确定最后一个非零系数位置翻转标识信息,可以包括:
- 15 若当前块使用最后一个非零系数位置翻转,则确定最后一个非零系数位置翻转标识信息的取值为第一值;或者,
若当前块不使用最后一个非零系数位置翻转,则确定最后一个非零系数位置翻转标识信息的取值为第二值。
- 在本申请实施例中,最后一个非零系数位置翻转标识信息可以用 `reverse_last_sig_coeff_flag` 表示。
20 这里,最后一个非零系数位置翻转标识信息至少可以为下述其中一项的标识信息:序列(sequence)级、图像(picture)级、片(slice)级和块(block)级;甚至也可以为更高级别(如VUI、SEI等)的标识信息,这里不作任何限定。
- 也就是说, `reverse_last_sig_coeff_flag` 可能是一个序列级或者更高级别的标志,或者也可能是图像级的标志或者片级的标志或者块级的标志或者其他级别的标志。另外,块级的标志可以包括最大编码单元(LCU)级的标志或编码单元(CU)级的标志或者其他块级的标志,本申请实施例均不作任何限定。
- 25 这样,以第一值为1,第二值为0为例,如果确定当前块使用最后一个非零系数位置翻转,那么 `reverse_last_sig_coeff_flag` 的取值为1;或者,如果确定当前块不使用最后一个非零系数位置翻转,那么 `reverse_last_sig_coeff_flag` 的取值为0。
- 进一步地,最后一个非零系数的位置可以包括所述最后一个非零系数的初始水平坐标和初始竖直坐标。当初始水平坐标和初始竖直坐标为最后一个非零系数的位置相对于当前块的左上角位置之间的水平距离和竖直距离时,所述根据最后一个非零系数的位置以及最后一个非零系数位置翻转标识信息,确定最后一个非零系数的坐标信息,可以包括:
- 30 若最后一个非零系数位置翻转标识信息的取值为第一值,则根据最后一个非零系数的初始水平坐标和初始竖直坐标进行计算,确定最后一个非零系数的坐标信息;或者,
35 若最后一个非零系数位置翻转标识信息的取值为第二值,则根据最后一个非零系数的初始水平坐标和初始竖直坐标直接确定最后一个非零系数的坐标信息。
- 换言之,在一些实施例中,该方法还可以包括:
若最后一个非零系数位置翻转标识信息的取值为第一值,则确定最后一个非零系数的坐标信息为最后一个非零系数的位置相对于当前块的右下角位置之间的水平距离和竖直距离;或者,
40 若最后一个非零系数位置翻转标识信息的取值为第二值,则最后一个非零系数的坐标信息为最后一个非零系数的位置相对于当前块的左上角位置之间的水平距离和竖直距离。
- 也就是说,最后一个非零系数的坐标信息通常为最后一个非零系数的位置相对于当前块的左上角位置之间的水平距离和竖直距离。对于常规视频来说,由于非零系数大多数集中在左上角,而右下角的大片区域为0;但是对于高位宽、高质量、高码率的视频编解码来说,右下角也会出现大量的非零系数,
45 使得最后一个非零系数的坐标信息的值通常较大,这时候为了节省开销,系数编码的时候需要进行坐标变换(具体可以是坐标翻转计算,即坐标翻转后最后一个非零系数的坐标信息为最后一个非零系数的位置相对于当前块的右下角位置之间的水平距离和竖直距离),那么后续在解码器中,系数解码时也需要进行坐标翻转计算,再次翻转后可以恢复出最后一个非零系数的坐标信息为最后一个非零系数的位置相对于当前块的左上角位置之间的水平距离和竖直距离,从而确定出最后一个非零系数的位置。
- 50 进一步地,在一些实施例中,所述根据最后一个非零系数的初始水平坐标和初始竖直坐标进行计算,确定最后一个非零系数的坐标信息,可以包括:
确定当前块的宽度和高度;

根据当前块的宽度与最后一个非零系数的初始水平坐标进行减法计算,得到最后一个非零系数的水平坐标;以及根据当前块的高度与最后一个非零系数的初始竖直坐标进行减法计算,得到最后一个非零系数的竖直坐标;

5 根据最后一个非零系数的水平坐标和最后一个非零系数的竖直坐标,确定最后一个非零系数的坐标信息。

需要说明的是,这里的当前块可以是没有 zero-out 变换的块,也可以是 zero-out 变换后的块。以 zero-out 变换后的块为例,这时候当前块的宽度为 $1 < \log_2 \text{ZoTbWidth}$, 当前块的高度是 $1 < \log_2 \text{ZoTbHeight}$; 那么在 reverse_last_sig_coeff_flag 指示使用最后一个非零系数位置翻转 (即 reverse_last_sig_coeff_flag 的取值为 1) 的情况下,

10 $\text{LastSignificantCoeffX} = (1 < \log_2 \text{ZoTbWidth}) - 1 - \text{LastSignificantCoeffX};$

$\text{LastSignificantCoeffY} = (1 < \log_2 \text{ZoTbHeight}) - 1 - \text{LastSignificantCoeffY}.$

其中,等式右边的 ($\text{LastSignificantCoeffX}$, $\text{LastSignificantCoeffY}$) 表示直接确定的最后一个非零系数的坐标信息 (即最后一个非零系数的初始水平坐标和初始竖直坐标), 等式左边的 ($\text{LastSignificantCoeffX}$, $\text{LastSignificantCoeffY}$) 表示坐标翻转后得到的最后一个非零系数的坐标信息 (也即在当前块使用最后一个非零系数位置翻转的情况下,被写入码流的最后一个非零系数的坐标信息)。

15 在一些实施例中,将最后一个非零系数的坐标信息写入码流,可以包括:

根据所述最后一个非零系数的坐标信息,确定最后一个非零系数的水平坐标的前缀信息、最后一个非零系数的竖直坐标的前缀信息、最后一个非零系数的水平坐标的后缀信息和最后一个非零系数的竖直坐标的后缀信息;

20 将所述最后一个非零系数的水平坐标的前缀信息、所述最后一个非零系数的竖直坐标的前缀信息、所述最后一个非零系数的水平坐标的后缀信息和所述最后一个非零系数的竖直坐标的后缀信息写入码流。

需要说明的是,最后一个非零系数的水平坐标的前缀信息用 last_sig_coeff_x_prefix 表示,最后一个非零系数的竖直坐标的前缀信息用 last_sig_coeff_y_prefix 表示,最后一个非零系数的水平坐标的后缀信息用 last_sig_coeff_x_suffix 表示,最后一个非零系数的竖直坐标的后缀信息用 last_sig_coeff_y_suffix 表示;然后将 last_sig_coeff_x_prefix、last_sig_coeff_x_suffix、last_sig_coeff_y_prefix、last_sig_coeff_y_suffix 写入码流,以便解码器能够通过解析码流确定出最后一个非零系数的坐标信息。

如此,本申请实施例提供一种修改最后一个非零系数的位置导出方法。也就是说通常情况下,系数编解码方法还是与相关技术的已有方法一样。某种情况,比如可以是指高位宽或高质量或高码率的视频或无损压缩的视频编解码。由于通常情况下, last_sig_coeff_x_prefix、last_sig_coeff_x_suffix 编码的是最后一个非零系数位置的横坐标,也就是相对于当前块左上角的水平距离; last_sig_coeff_y_prefix、last_sig_coeff_y_suffix 编码的是最后一个非零系数位置的纵坐标,也就是相对于当前块左上角的竖直距离,如图 10A 所示。而在高位宽或高质量或高码率的视频或无损压缩的视频编解码的情况下,最后一个非零系数的位置普遍接近于当前块所有可能是非 0 系数的区域的右下角。在这种情况下, last_sig_coeff_x_prefix、last_sig_coeff_x_suffix 编码的是最后一个非零系数位置相对于当前块所有可能是非 0 系数的区域的右下角的水平距离; last_sig_coeff_y_prefix、last_sig_coeff_y_suffix 编码的是最后一个非零系数位置相对于当前块所有可能是非 0 系数的区域的右下角的竖直距离,如图 10B 所示;因此,本申请实施例可以通过引入 reverse_last_sig_coeff_flag,从而可以解决在码流中编码较大的值会带来更大的开销问题。

进一步地,在视频标识信息指示视频满足预设条件时,这时候还可默认需要编码所有可能需要编码的系数,也就是不再使用最后一个非零系数的位置,而是按预设扫描顺序把当前块所有可能不为 0 的系数都扫描到;因此,本申请实施例还可以引入最后一个系数使能标识信息,用于确定当前块是否使用最后一个系数位置。

45 在一些实施例中,当视频标识信息指示视频满足预设条件时,该方法还可以包括:

确定最后一个系数使能标识信息;

当最后一个系数使能标识信息指示当前块使用最后一个系数位置时,按照预设扫描顺序对所述最后一个系数位置之前的所有系数进行编码,并将编码后得到的比特信息、视频标识信息以及最后一个系数使能标识信息写入码流。

50 需要说明的是,对于最后一个系数使能标识信息而言,可以用 default_last_coeff_enabled_flag 表示。在本申请实施例中,最后一个系数使能标识信息至少可以为下述其中一项的标识信息:序列级、图像级、片级和块级;甚至也可以为更高级别 (如 VUI、SEI 等) 的标识信息,这里不作任何限定。

还需要说明的是,对于最后一个系数使能标识信息而言,在一些实施例中,所述确定最后一个系数使能标识信息,可以包括:

若当前块使用所述最后一个系数位置,则确定最后一个系数使能标识信息的取值为第一值;或者,若当前块不使用所述最后一个系数位置,则确定最后一个系数使能标识信息的取值为第二值。

- 5 也就是说,以第一值为 1,第二值为 0 为例,如果确定当前块使用最后一个系数位置,那么 default_last_coeff_enabled_flag 的取值为 1;或者,如果确定当前块不使用最后一个系数位置,那么 default_last_coeff_enabled_flag 的取值为 0。

- 10 进一步地,对于最后一个系数位置而言,在一些实施例中,最后一个系数位置是当前块中所有可能不为零的系数所组成的矩阵的右下角位置;或者,最后一个系数位置是当前块根据预设扫描顺序执行扫描所有可能不为零的系数的最后位置。

需要注意的是,本申请实施例中的最后一个系数位置并不代表最后一个非零系数的位置。因为最后一个系数位置上的系数可能是 0,而最后一个非零系数的位置上的系数一定不是 0。

在一种特定示例中,该方法还可以包括:将最后一个非零系数的位置设置在所述最后一个系数位置。

- 15 也就是说,本申请实施例仍然可以使用最后一个非零系数的位置,这时候需要把最后一个非零系数的位置放在按预设扫描顺序的当前块所有可能不为 0 的系数的最后位置。

进一步地,最后一个系数位置可以用 (LastCoeffX, LastCoeffY) 表示,即当前块按照预设扫描顺序所有可能不为 0 的系数的最后位置。在一些实施例中,该方法还可以包括:

确定所述当前块在预设操作后得到变换块的宽度和高度;

- 20 根据所述变换块的宽度和高度进行坐标计算,得到所述变换块的右下角坐标信息;

根据所述变换块的右下角坐标信息,确定所述最后一个系数位置。

在这里,预设操作至少包括:强制置零 (zero-out) 操作。

需要说明的是, (LastCoeffX, LastCoeffY) 表示 zero-out 后的变换块的右下角坐标信息;其中, (LastCoeffX, LastCoeffY) 的导出方法如下:

$\text{LastCoeffX} = (1 < \log_2 \text{ZoTbWidth}) - 1$; $\text{LastCoeffY} = (1 < \log_2 \text{ZoTbHeight}) - 1$ 。

- 25 这样,如果 default_last_coeff_enabled_flag 的取值为 1,那么根据 (LastCoeffX, LastCoeffY) 即可确定出最后一个系数位置。

在一种特定示例中,仍然使用最后一个非零系数的位置,这时候可以把最后一个非零系数的位置放在按照预设扫描顺序的当前块所有可能为 0 的系数的最后位置。在一些实施例中,该方法还可以包括:

- 30 当将最后一个非零系数的位置设置在所述最后一个系数位置时,根据所述变换块的右下角坐标信息,确定所述最后一个非零系数的位置。

也就是说,最后一个非零系数的位置可以用 (LastSignificantCoeffX, LastSignificantCoeffY) 表示, (LastSignificantCoeffX, LastSignificantCoeffY) 的导出方法如下:

$\text{LastSignificantCoeffX} = (1 < \log_2 \text{ZoTbWidth}) - 1$; $\text{LastSignificantCoeffY} = (1 < \log_2 \text{ZoTbHeight}) - 1$ 。

- 35 其中, (LastSignificantCoeffX, LastSignificantCoeffY) 表示 zero-out 后的变换块的右下角坐标信息。如果 default_last_coeff_enabled_flag 的取值为 1,那么根据 (LastSignificantCoeffX, LastSignificantCoeffY) 即可确定出最后一个非零系数的位置。

进一步地,在当前块不使用最后一个系数位置的情况下,即最后一个系数使能标识信息的取值为 0,在一些实施例中,该方法还可以包括:

- 40 确定最后一个非零系数的水平坐标的前缀信息、最后一个非零系数的竖直坐标的前缀信息、最后一个非零系数的水平坐标的后缀信息和最后一个非零系数的竖直坐标的后缀信息;

根据所述最后一个非零系数的水平坐标的前缀信息、所述最后一个非零系数的竖直坐标的前缀信息、所述最后一个非零系数的水平坐标的后缀信息和所述最后一个非零系数的竖直坐标的后缀信息,确定所述最后一个非零系数的位置;

- 45 按照预设扫描顺序对所述最后一个非零系数的位置之前的所有系数进行编码,并将所述最后一个非零系数的水平坐标的前缀信息、所述最后一个非零系数的竖直坐标的前缀信息、所述最后一个非零系数的水平坐标的后缀信息和所述最后一个非零系数的竖直坐标的后缀信息写入码流。

需要说明的是,如果当前块不使用最后一个系数位置,那么这时候需要确定最后一个非零系数的位置。具体地,需要确定 last_sig_coeff_x_prefix、last_sig_coeff_x_suffix、last_sig_coeff_y_prefix、last_sig_coeff_y_suffix;然后将 last_sig_coeff_x_prefix、last_sig_coeff_x_suffix、last_sig_coeff_y_prefix、last_sig_coeff_y_suffix 写入码流。

- 50 如此,在系数编码时,默认需要编码所有可能需要编码的系数。也就是说通常情况下,系数编解码方法还是与相关技术中的已有方法一样。某种情况,比如可以是指高位宽或高质量或高码率的视频或无

损压缩的视频编解码，默认需要编码所有可能需要编码的系数，也就是不再使用最后一个非零系数的位置，而是按预设扫描顺序把当前块所有可能不为 0 的系数都扫描到；或者说，把最后一个需要编码的系数位置放在按预设扫描顺序的当前块所有可能不为 0 的系数的最后位置。这个位置通常是指当前块所有可能不为 0 的系数所组成的矩阵的右下角位置。因此，本申请实施例可以通过引入

5 default_last_coeff_enabled_flag，这时候可以减少甚至消除关于最后一个非零系数位置的相关语法元素，能够节省开销，避免造成浪费。

进一步地，在视频标识信息指示视频满足预设条件时，还可以扫描到的子块都默认需要编码，这时候就不需要在码流中传输 sb_coded_flag，即编码器/解码器均不需要处理这个标志，以加快编解码速度。因此，本申请实施例还可以引入子块默认编码标识信息，用于确定当前块内的待解码子块是否为默认编

10 码。

在一些实施例中，当视频标识信息指示视频满足预设条件时，该方法还可以包括：

确定当前块内的待编码子块的子块默认编码标识信息；

当子块默认编码标识信息指示所述待编码子块默认编码时，对待编码子块内的所有系数进行编码，并将编码后得到的比特信息以及所述子块默认编码标识信息写入码流。

需要说明的是，对于子块默认标识信息而言，可以用 default_sb_coded_flag 表示。在本申请实施例中，子块默认编码标识信息至少为下述其中一项的标识信息：序列级、图像级、片级和块级；甚至也可以为更高级别（如 VUI、SEI 等）的标识信息，这里不作任何限定。

还需要说明的是，对于子块默认标识信息而言，在一些实施例中，所述确定待编码子块的子块默认编码标识信息，可以包括：

若待编码子块默认编码，则确定子块默认编码标识信息的取值为第一值；或者，

若待编码子块不默认编码，则确定子块默认编码标识信息的取值为第二值。

这样，以第一值为 1，第二值为 0 为例，如果确定待解码子块默认需要编码，那么 default_sb_coded_flag 的取值为 1；或者，如果确定待解码子块不默认需要编码，那么 default_sb_coded_flag 的取值为 0。

在待解码子块默认需要编码的情况下，这时候 default_sb_coded_flag 的取值为 1，意味着 sb_coded_flag 的取值为 1，即不再需要编码 sb_coded_flag。但是在待解码子块不默认需要编码的情况下，即当子块默认编码标识信息指示待编码子块不默认编码时，在一些实施例中，该方法还可以包括：确定待编码子块的子块编码标识信息，并将子块编码标识信息写入码流。

进一步地，在一些实施例中，所述确定所述待编码子块的子块编码标识信息，可以包括：

若子块内需进行编码，则确定子块编码标识信息的取值为第一值；或者，

若子块内的所有系数均为零，则确定子块编码标识信息的取值为第二值。

在本申请实施例中，子块编码标识信息可以用 sb_coded_flag 表示。以第一值为 1，第二值为 0 为例，如果确定需要对待编码子块进行编码，意味着待编码子块内包含有待编码的非零系数，那么 sb_coded_flag 的取值为 1；或者，如果确定不需要对待编码子块进行编码，意味着待编码子块内的所有系数均为零，那么 sb_coded_flag 的取值为 0。

如此，在系数编码时，扫描到的子块都默认需要编码，或者说扫描到的子块都默认包含非零系数。也就是说通常情况下，系数编码的方法还是与相关技术中的已有方法一样。某种情况，比如可以是指高位宽或高质量或高码率的视频或无损压缩的视频编解码，这种情况下非零系数很多，扫描到的子块几乎都是需要编码；或者说，扫描到的子块几乎都包含非零系数。这样就不需要在码流中传输 sb_coded_flag，编码器也就不需要处理这个标志，从而能够加快编解码速度。由于去掉了一个几乎没有的标志，此时还会带来一点压缩性能的提升。

本实施例还提供了一种系数编码方法，应用于编码器。通过确定视频标识信息以及最后一个非零系数的位置；当视频标识信息指示视频满足预设条件时，确定最后一个非零系数位置翻转标识信息；根据最后一个非零系数的位置以及最后一个非零系数位置翻转标识信息，确定最后一个非零系数的坐标信息；按照预设扫描顺序对最后一个非零系数的位置之前的所有系数进行编码，并将编码后得到的比特信息、视频标识信息以及最后一个非零系数的坐标信息写入码流。这样，在高位宽、高码率、高质量或无损的视频编解码场景中，由于系数分布规律与常规视频场景不同，在系数编码中通过减少甚至消除上下文模式编码的语法元素个数，比如关于最后一个非零系数位置、子块编码标识等语法元素，甚至还可以在最后一个非零系数的坐标信息的值偏大时进行坐标变换，从而可以减少码流中编码所带来的开销，进而提高系数编码的吞吐量和编码速度；另外，由于减少或消除的语法元素在高位宽、高码率、高质量或无损的视频编码中的影响较小，还能够提升压缩效率。

本申请的又一实施例中，基于前述实施例相同的发明构思，参见图 12，其示出了本申请实施例提供的一种编码器 120 的组成结构示意图。如图 12 所示，该编码器 120 可以包括：第一确定单元 1201 和编码单元 1202；其中，

5 第一确定单元 1201，配置为确定视频标识信息以及最后一个非零系数的位置；以及当视频标识信息指示视频满足预设条件时，确定最后一个非零系数位置翻转标识信息；

第一确定单元 1201，还配置为根据最后一个非零系数的位置以及最后一个非零系数位置翻转标识信息，确定最后一个非零系数的坐标信息；

编码单元 1202，配置为按照预设扫描顺序对最后一个非零系数的位置之前的所有系数进行编码，并将编码后得到的比特信息、视频标识信息以及最后一个非零系数的坐标信息写入码流。

10 在一些实施例中，第一确定单元 1201，还配置为若视频满足预设条件，则确定视频标识信息的取值为第一值；或者，若视频不满足预设条件，则确定视频标识信息的取值为第二值。

在一些实施例中，预设条件至少包括下述之一：高位宽、高质量、高码率、高帧率和无损压缩。

15 在一些实施例中，第一确定单元 1201，还配置为若当前块使用最后一个非零系数位置翻转，则确定最后一个非零系数位置翻转标识信息的取值为第一值；或者，若当前块不使用最后一个非零系数位置翻转，则确定最后一个非零系数位置翻转标识信息的取值为第二值。

在一些实施例中，最后一个非零系数的位置包括最后一个非零系数的初始水平坐标和初始竖直坐标，初始水平坐标和初始竖直坐标为最后一个非零系数的位置相对于当前块的左上角位置之间的水平距离和竖直距离；

20 相应地，第一确定单元 1201，还配置为若最后一个非零系数位置翻转标识信息的取值为第一值，则根据最后一个非零系数的初始水平坐标和初始竖直坐标进行计算，确定最后一个非零系数的坐标信息；或者，若最后一个非零系数位置翻转标识信息的取值为第二值，则根据最后一个非零系数的初始水平坐标和初始竖直坐标直接确定最后一个非零系数的坐标信息。

25 在一些实施例中，第一确定单元 1201，还配置为确定当前块的宽度和高度；根据当前块的宽度与最后一个非零系数的初始水平坐标进行减法计算，得到最后一个非零系数的水平坐标；以及根据当前块的高度与最后一个非零系数的初始竖直坐标进行减法计算，得到最后一个非零系数的竖直坐标；以及根据最后一个非零系数的水平坐标和最后一个非零系数的竖直坐标，确定最后一个非零系数的坐标信息。

30 在一些实施例中，第一确定单元 1201，还配置为若最后一个非零系数位置翻转标识信息的取值为第一值，则确定最后一个非零系数的坐标信息为最后一个非零系数的位置相对于当前块的右下角位置之间的水平距离和竖直距离；或者，若最后一个非零系数位置翻转标识信息的取值为第二值，则最后一个非零系数的坐标信息为最后一个非零系数的位置相对于当前块的左上角位置之间的水平距离和竖直距离。

35 在一些实施例中，编码单元 1202，还配置为根据最后一个非零系数的坐标信息，确定最后一个非零系数的水平坐标的前缀信息、最后一个非零系数的竖直坐标的前缀信息、最后一个非零系数的水平坐标的后缀信息和最后一个非零系数的竖直坐标的后缀信息；以及将最后一个非零系数的水平坐标的前缀信息、最后一个非零系数的竖直坐标的前缀信息、最后一个非零系数的水平坐标的后缀信息和最后一个非零系数的竖直坐标的后缀信息写入码流。

在一些实施例中，所述最后一个非零系数位置翻转标识信息至少为下述其中一项的标识信息：序列级、图像级、片级和块级。

40 在一些实施例中，第一确定单元 1201，还配置为当视频标识信息指示视频满足预设条件时，确定最后一个系数使能标识信息；

编码单元 1202，还配置为当最后一个系数使能标识信息指示当前块使用最后一个系数位置时，按照预设扫描顺序对最后一个系数位置之前的所有系数进行编码，并将编码后得到的比特信息、视频标识信息以及最后一个系数使能标识信息写入码流。

45 在一些实施例中，第一确定单元 1201，还配置为若当前块使用最后一个系数位置，则确定最后一个系数使能标识信息的取值为第一值；或者，若当前块不使用最后一个系数位置，则确定最后一个系数使能标识信息的取值为第二值。

在一些实施例中，最后一个系数位置是当前块中所有可能不为零的系数所组成的矩阵的右下角位置；或者，最后一个系数位置是当前块根据预设扫描顺序执行扫描所有可能不为零的系数的最后位置。

50 在一些实施例中，第一确定单元 1201，还配置为将最后一个非零系数的位置设置在所述最后一个系数位置。

在一些实施例中，第一确定单元 1201，还配置为确定所述当前块在预设操作后得到变换块的宽度和高度；根据所述变换块的宽度和高度进行坐标计算，得到所述变换块的右下角坐标信息；以及根据所

述变换块的右下角坐标信息,确定所述最后一个系数位置。

在一些实施例中,预设操作至少包括:强制置零(zero-out)操作。

在一些实施例中,第一确定单元1201,还配置为当将最后一个非零系数的位置设置在最后一个系数位置时,根据变换块的右下角坐标信息,确定最后一个非零系数的位置。

- 5 在一些实施例中,第一确定单元1201,还配置为当所述当前块不使用所述最后一个系数位置时,确定最后一个非零系数的水平坐标的前缀信息、最后一个非零系数的竖直坐标的前缀信息、最后一个非零系数的水平坐标的后缀信息和最后一个非零系数的竖直坐标的后缀信息;以及根据最后一个非零系数的水平坐标的前缀信息、最后一个非零系数的竖直坐标的前缀信息、最后一个非零系数的水平坐标的后缀信息和最后一个非零系数的竖直坐标的后缀信息,确定最后一个非零系数的位置;

- 10 编码单元1202,还配置为按照预设扫描顺序对最后一个非零系数的位置之前的所有系数进行编码,并将最后一个非零系数的水平坐标的前缀信息、最后一个非零系数的竖直坐标的前缀信息、最后一个非零系数的水平坐标的后缀信息和最后一个非零系数的竖直坐标的后缀信息写入码流。

在一些实施例中,最后一个系数使能标识信息至少为下述其中一项的标识信息:序列级、图像级、片级和块级。

- 15 在一些实施例中,第一确定单元1201,还配置为当视频标识信息指示视频满足预设条件时,确定当前块内的待编码子块的子块默认编码标识信息;

编码单元1202,还配置为当子块默认编码标识信息指示所述待编码子块默认编码时,对待编码子块内的所有系数进行编码,并将编码后得到的比特信息以及子块默认编码标识信息写入码流。

- 20 在一些实施例中,第一确定单元1201,还配置为当子块默认编码标识信息指示待编码子块不默认编码时,确定待编码子块的子块编码标识信息,并将子块编码标识信息写入码流。

在一些实施例中,第一确定单元1201,还配置为若待编码子块默认编码,则确定子块默认编码标识信息的取值为第一值;或者,若待编码子块不默认编码,则确定子块默认编码标识信息的取值为第二值。

- 25 在一些实施例中,第一确定单元1201,还配置为若子块内需进行编码,则确定子块编码标识信息的取值为第一值;或者,若子块内的所有系数均为零,则确定子块编码标识信息的取值为第二值。

在一些实施例中,子块默认编码标识信息至少为下述其中一项的标识信息:序列级、图像级、片级和块级。

在一些实施例中,第一值为1,第二值为0。

- 30 在一些实施例中,第一确定单元1201,还配置为当视频标识信息为高位宽标识信息时,若视频满足高位宽,则确定高位宽标识信息指示视频满足预设条件。

在一些实施例中,第一确定单元1201,还配置为当视频标识信息为高码率标识信息时,若视频满足高码率,则确定所述高码率标识信息指示视频满足预设条件。

在一些实施例中,第一确定单元1201,还配置为当视频标识信息为高质量标识信息时,若视频满足高质量,则确定高质量标识信息指示视频满足预设条件。

- 35 在一些实施例中,第一确定单元1201,还配置为当视频标识信息为无损压缩标识信息时,若视频满足无损压缩,则确定无损压缩标识信息指示视频满足预设条件。

可以理解地,在本申请实施例中,“单元”可以是部分电路、部分处理器、部分程序或软件等等,当然也可以是模块,还可以是非模块化的。而且在本实施例中的各组成部分可以集成在一个处理单元中,也可以是各个单元单独物理存在,也可以两个或两个以上单元集成在一个单元中。上述集成的单元既可以采用硬件的形式实现,也可以采用软件功能模块的形式实现。

- 40 所述集成的单元如果以软件功能模块的形式实现并非作为独立的产品进行销售或使用时,可以存储在一个计算机可读取存储介质中,基于这样的理解,本实施例的技术方案本质上或者说对现有技术做出贡献的部分或者该技术方案的全部或部分可以以软件产品的形式体现出来,该计算机软件产品存储在一个存储介质中,包括若干指令用以使得一台计算机设备(可以是个人计算机,服务器,或者网络设备等)或 processor(处理器)执行本实施例所述方法的全部或部分步骤。而前述的存储介质包括:U盘、移动硬盘、只读存储器(Read Only Memory, ROM)、随机存取存储器(Random Access Memory, RAM)、磁碟或者光盘等各种可以存储程序代码的介质。

因此,本申请实施例提供了一种计算机存储介质,应用于编码器120,该计算机存储介质存储有计算机程序,所述计算机程序被第一处理器执行时实现前述实施例中任一项所述的方法。

- 50 基于上述编码器120的组成以及计算机存储介质,参见图13,其示出了本申请实施例提供的编码器120的具体硬件结构示意图。如图13所示,可以包括:第一通信接口1301、第一存储器1302和第一处理器1303;各个组件通过第一总线系统1304耦合在一起。可理解,第一总线系统1304用于实现

这些组件之间的连接通信。第一总线系统 1304 除包括数据总线之外,还包括电源总线、控制总线和状态信号总线。但是为了清楚说明起见,在图 13 中将各种总线都标为第一总线系统 1304。其中,

第一通信接口 1301,用于在与其他外部网元之间进行收发信息过程中,信号的接收和发送;

第一存储器 1302,用于存储能够在第一处理器 1303 上运行的计算机程序;

5 第一处理器 1303,用于在运行所述计算机程序时,执行:

确定视频标识信息以及最后一个非零系数的位置;

当视频标识信息指示视频满足预设条件时,确定最后一个非零系数位置翻转标识信息;

根据最后一个非零系数的位置以及最后一个非零系数位置翻转标识信息,确定最后一个非零系数的坐标信息;

10 按照预设扫描顺序对最后一个非零系数的位置之前的所有系数进行编码,并将编码后得到的比特信息、视频标识信息以及最后一个非零系数的坐标信息写入码流。

可以理解,本申请实施例中的第一存储器 1302 可以是易失性存储器或非易失性存储器,或可包括易失性和非易失性存储器两者。其中,非易失性存储器可以是只读存储器(Read-Only Memory, ROM)、可编程只读存储器(Programmable ROM, PROM)、可擦除可编程只读存储器(Erasable PROM, EPROM)、电可擦除可编程只读存储器(Electrically EPROM, EEPROM)或闪存。易失性存储器可以是随机存取存储器(Random Access Memory, RAM),其用作外部高速缓存。通过示例性但不是限制性说明,许多形式的 RAM 可用,例如静态随机存取存储器(Static RAM, SRAM)、动态随机存取存储器(Dynamic RAM, DRAM)、同步动态随机存取存储器(Synchronous DRAM, SDRAM)、双倍数据速率同步动态随机存取存储器(Double Data Rate SDRAM, DDRSDRAM)、增强型同步动态随机存取存储器(Enhanced SDRAM, ESDRAM)、同步连接动态随机存取存储器(Synchlink DRAM, SDRAM)和直接内存总线随机存取存储器(Direct Rambus RAM, DRAM)。本申请描述的系统和方法的第一存储器 1302 旨在包括但不限于这些和任意其它适合类型的存储器。

而第一处理器 1303 可能是一种集成电路芯片,具有信号的处理能力。在实现过程中,上述方法的各步骤可以通过第一处理器 1303 中的硬件的集成逻辑电路或者软件形式的指令完成。上述的第一处理器 1303 可以是通用处理器、数字信号处理器(Digital Signal Processor, DSP)、专用集成电路(Application Specific Integrated Circuit, ASIC)、现成可编程门阵列(Field Programmable Gate Array, FPGA)或者其他可编程逻辑器件、分立门或者晶体管逻辑器件、分立硬件组件。可以实现或者执行本申请实施例中的公开的各方法、步骤及逻辑框图。通用处理器可以是微处理器或者该处理器也可以是任何常规的处理器等。结合本申请实施例所公开的方法的步骤可以直接体现为硬件译码处理器执行完成,或者用译码处理器中的硬件及软件模块组合执行完成。软件模块可以位于随机存储器、闪存、只读存储器,可编程只读存储器或者电可擦写可编程存储器、寄存器等本领域成熟的存储介质中。该存储介质位于第一存储器 1302,第一处理器 1303 读取第一存储器 1302 中的信息,结合其硬件完成上述方法的步骤。

可以理解的是,本申请描述的这些实施例可以用硬件、软件、固件、中间件、微码或其组合来实现。对于硬件实现,处理单元可以实现在一个或多个专用集成电路(Application Specific Integrated Circuits, ASIC)、数字信号处理器(Digital Signal Processing, DSP)、数字信号处理设备(DSP Device, DSPD)、可编程逻辑设备(Programmable Logic Device, PLD)、现场可编程门阵列(Field-Programmable Gate Array, FPGA)、通用处理器、控制器、微控制器、微处理器、用于执行本申请所述功能的其它电子单元或其组合中。对于软件实现,可通过执行本申请所述功能的模块(例如过程、函数等)来实现本申请所述的技术。软件代码可存储在存储器中并通过处理器执行。存储器可以在处理器中或在处理器外部实现。

40 可选地,作为另一个实施例,第一处理器 1303 还配置为在运行所述计算机程序时,执行前述实施例中任一项所述的方法。

本实施例提供了一种编码器,该编码器可以包括第一确定单元和编码单元。这样,在高位宽、高码率、高质量或无损的视频编解码场景中,由于系数分布规律与常规视频场景不同,在系数编码中通过减少甚至消除上下文模式编码的语法元素个数,从而可以减少码流中编码所带来的开销,进而提高系数编码的吞吐量和编解码速度;另外,由于减少或消除的语法元素在高位宽、高码率、高质量或无损的视频编解码中的影响较小,还能够提升压缩效率。

本申请的再一实施例中,基于前述实施例相同的发明构思,参见图 14,其示出了本申请实施例提供的一种解码器 140 的组成结构示意图。如图 14 所示,该解码器 140 可以包括:解析单元 1401 和第二确定单元 1402;其中,

解析单元 1401,配置为解析码流,获取视频标识信息;以及当视频标识信息指示视频满足预设条件时,解析码流,获取最后一个非零系数位置翻转标识信息和最后一个非零系数的坐标信息;

第二确定单元 1402, 配置为当最后一个非零系数位置翻转标识信息指示当前块使用最后一个非零系数位置翻转时, 对最后一个非零系数的坐标信息进行计算, 确定最后一个非零系数的位置;

解析单元 1401, 还配置为按照预设扫描顺序对最后一个非零系数的位置之前的所有系数进行解码, 确定当前块的系数。

- 5 在一些实施例中, 第二确定单元 1402, 还配置为当最后一个非零系数位置翻转标识信息指示当前块不使用最后一个非零系数位置翻转时, 根据最后一个非零系数的坐标信息直接确定最后一个非零系数的位置;

解析单元 1401, 还配置为按照预设扫描顺序对最后一个非零系数的位置之前的所有系数进行解码, 确定当前块的系数。

- 10 在一些实施例中, 第二确定单元 1402, 还配置为若视频标识信息的取值为第一值, 则确定视频标识信息指示视频满足预设条件; 或者, 若视频标识信息的取值为第二值, 则确定视频标识信息指示视频不满足预设条件。

在一些实施例中, 预设条件至少包括下述之一: 高位宽、高质量、高码率、高帧率和无损压缩。

- 15 在一些实施例中, 第二确定单元 1402, 还配置为若最后一个非零系数位置翻转标识信息的取值为第一值, 则确定最后一个非零系数位置翻转标识信息指示当前块使用最后一个非零系数位置翻转; 或者, 若最后一个非零系数位置翻转标识信息的取值为第二值, 则确定最后一个非零系数位置翻转标识信息指示当前块不使用最后一个非零系数位置翻转。

- 20 在一些实施例中, 解析单元 1401, 还配置为解析码流, 获取最后一个非零系数的水平坐标的前缀信息、最后一个非零系数的竖直坐标的前缀信息、最后一个非零系数的水平坐标的后缀信息和最后一个非零系数的竖直坐标的后缀信息;

第二确定单元 1402, 还配置为根据最后一个非零系数的水平坐标的前缀信息和最后一个非零系数的水平坐标的后缀信息, 确定最后一个非零系数的水平坐标; 以及根据最后一个非零系数的竖直坐标的前缀信息和最后一个非零系数的竖直坐标的后缀信息, 确定最后一个非零系数的竖直坐标; 以及根据最后一个非零系数的水平坐标和最后一个非零系数的竖直坐标, 确定最后一个非零系数的坐标信息。

- 25 在一些实施例中, 第二确定单元 1402, 还配置为当最后一个非零系数位置翻转标识信息指示当前块使用最后一个非零系数位置翻转时, 确定最后一个非零系数的坐标信息为最后一个非零系数的位置相对于当前块的右下角位置之间的水平距离和竖直距离;

- 30 进一步地, 第二确定单元 1402, 还配置为确定当前块的宽度和高度; 根据当前块的宽度与最后一个非零系数的位置相对于当前块的右下角位置之间的水平距离进行减法计算, 得到最后一个非零系数的水平坐标; 以及根据当前块的高度与最后一个非零系数的位置相对于当前块的右下角位置之间的竖直距离进行减法计算, 得到最后一个非零系数的竖直坐标; 以及根据最后一个非零系数的水平坐标和最后一个非零系数的竖直坐标, 确定最后一个非零系数的位置。

- 35 在一些实施例中, 第二确定单元 1402, 还配置为当最后一个非零系数位置翻转标识信息指示当前块不使用最后一个非零系数位置翻转时, 确定最后一个非零系数的坐标信息为最后一个非零系数的位置相对于当前块的左上角位置之间的水平距离和竖直距离; 以及根据最后一个非零系数的位置相对于当前块的左上角位置之间的水平距离和竖直距离, 确定最后一个非零系数的位置。

在一些实施例中, 最后一个非零系数位置翻转标识信息至少为下述其中一项的标识信息: 序列(sequence)级、图像(picture)级、片(slice)级和块(block)级。

- 40 在一些实施例中, 解析单元 1401, 还配置为解析码流, 获取最后一个系数使能标识信息; 以及当最后一个系数使能标识信息指示当前块使用最后一个系数位置时, 按照预设扫描顺序对最后一个系数位置之前的所有系数进行解码, 确定当前块的系数。

在一些实施例中, 第二确定单元 1402, 还配置为若最后一个系数使能标识信息的取值为第一值, 则确定最后一个系数使能标识信息指示当前块使用最后一个系数位置; 或者, 若最后一个系数使能标识信息的取值为第二值, 则确定最后一个系数使能标识信息指示当前块不使用最后一个系数位置。

- 45 在一些实施例中, 解析单元 1401, 还配置为当最后一个系数使能标识信息的取值为第二值时, 解析码流, 获取最后一个非零系数的水平坐标的前缀信息、最后一个非零系数的竖直坐标的前缀信息、最后一个非零系数的水平坐标的后缀信息和最后一个非零系数的竖直坐标的后缀信息;

- 50 第二确定单元 1402, 还配置为根据最后一个非零系数的水平坐标的前缀信息、最后一个非零系数的竖直坐标的前缀信息、最后一个非零系数的水平坐标的后缀信息和最后一个非零系数的竖直坐标的后缀信息, 确定最后一个非零系数的位置; 以及按照预设扫描顺序对最后一个非零系数的位置之前的所有系数进行解码, 确定当前块的系数。

在一些实施例中, 最后一个系数位置是当前块中所有可能不为零的系数所组成的矩阵的右下角位置;

或者,最后一个系数位置是当前块根据预设扫描顺序执行扫描所有可能不为零的系数的最后位置。

在一些实施例中,第二确定单元 1402,还配置为将最后一个非零系数的位置设置在最后一个系数位置。

5 在一些实施例中,第二确定单元 1402,还配置为确定当前块在预设操作后得到变换块的宽度和高度;以及根据变换块的宽度和高度进行坐标计算,得到变换块的右下角坐标信息;以及根据变换块的右下角坐标信息,确定最后一个系数位置。

在一些实施例中,预设操作至少包括:强制置零(zero-out)操作。

在一些实施例中,第二确定单元 1402,还配置为当将最后一个非零系数的位置设置在最后一个系数位置时,根据变换块的右下角坐标信息,确定最后一个非零系数的位置。

10 在一些实施例中,最后一个系数使能标识信息至少为下述其中一项的标识信息:序列级、图像级、片级和块级。

在一些实施例中,解析单元 1401,还配置为当视频标识信息指示视频满足预设条件时,解析码流,获取子块默认编码标识信息;以及当子块默认编码标识信息指示当前块内的待解码子块默认编码时,确定子块编码标识信息的取值为第一值,对待解码子块内的所有系数进行解码。

15 在一些实施例中,解析单元 1401,还配置为当子块默认编码标识信息指示待解码子块不默认编码时,解析码流,获取子块编码标识信息;以及当子块编码标识信息的取值为第一值时,对待解码子块内的所有系数进行解码。

20 在一些实施例中,第二确定单元 1402,还配置为若子块默认编码标识信息的取值为第一值,则确定子块默认编码标识信息指示待解码子块默认编码;或者,若子块默认编码标识信息的取值为第二值,则确定子块默认编码标识信息指示待解码子块不默认编码。

在一些实施例中,第二确定单元 1402,还配置为若子块编码标识信息的取值为第一值,则确定对待解码子块内的所有系数进行解码;或者,若子块编码标识信息的取值为第二值,则确定待解码子块内的所有系数均为零。

25 在一些实施例中,子块默认编码标识信息至少为下述其中一项的标识信息:序列级、图像级、片级和块级。

在一些实施例中,第一值为 1,第二值为 0。

在一些实施例中,第二确定单元 1402,还配置为当视频标识信息为高位宽标识信息时,若高位宽标识信息指示视频满足高位宽,则确定视频满足预设条件。

30 在一些实施例中,第二确定单元 1402,还配置为当视频标识信息为高码率标识信息时,若高码率标识信息指示视频满足高码率,则确定视频满足预设条件。

在一些实施例中,第二确定单元 1402,还配置为当视频标识信息为高质量标识信息时,若高质量标识信息指示视频满足高质量,则确定视频满足预设条件。

在一些实施例中,第二确定单元 1402,还配置为当视频标识信息为无损压缩标识信息时,若无损压缩标识信息指示视频满足无损压缩,则确定视频满足预设条件。

35 可以理解地,在本申请实施例中,“单元”可以是部分电路、部分处理器、部分程序或软件等等,当然也可以是模块,还可以是非模块化的。而且在本实施例中的各组成部分可以集成在一个处理单元中,也可以是各个单元单独物理存在,也可以两个或两个以上单元集成在一个单元中。上述集成的单元既可以采用硬件的形式实现,也可以采用软件功能模块的形式实现。

40 所述集成的单元如果以软件功能模块的形式实现并非作为独立的产品进行销售或使用时,可以存储在一个计算机可读存储介质中,基于这样的理解,本实施例的技术方案本质上或者说对现有技术做出贡献的部分或者该技术方案的全部或部分可以以软件产品的形式体现出来,该计算机软件产品存储在一个存储介质中,包括若干指令用以使得一台计算机设备(可以是个人计算机,服务器,或者网络设备等)或 processor(处理器)执行本实施例所述方法的全部或部分步骤。而前述的存储介质包括:U 盘、移动硬盘、只读存储器(Read Only Memory,ROM)、随机存取存储器(Random Access Memory, RAM)、磁碟或者光盘等各种可以存储程序代码的介质。

因此,本申请实施例提供了一种计算机存储介质,应用于解码器 140,该计算机存储介质存储有计算机程序,所述计算机程序被第一处理器执行时实现前述实施例中任一项所述的方法。

45 基于上述解码器 140 的组成以及计算机存储介质,参见图 15,其示出了本申请实施例提供的解码器 140 的具体硬件结构示意图。如图 15 所示,可以包括:第二通信接口 1501、第二存储器 1502 和第二处理器 1503;各个组件通过第二总线系统 1504 耦合在一起。可理解,第二总线系统 1504 用于实现这些组件之间的连接通信。第二总线系统 1504 除包括数据总线之外,还包括电源总线、控制总线和状态信号总线。但是为了清楚说明起见,在图 15 中将各种总线都标为第二总线系统 1504。其中,

第二通信接口 1501, 用于在与其他外部网元之间进行收发信息过程中, 信号的接收和发送;

第二存储器 1502, 用于存储能够在第二处理器 1503 上运行的计算机程序;

第二处理器 1503, 用于在运行所述计算机程序时, 执行:

解析码流, 获取视频标识信息;

- 5 当视频标识信息指示视频满足预设条件时, 解析码流, 获取最后一个非零系数位置翻转标识信息和最后一个非零系数的坐标信息;

当最后一个非零系数位置翻转标识信息指示当前块使用最后一个非零系数位置翻转时, 对最后一个非零系数的坐标信息进行计算, 确定最后一个非零系数的位置;

按照预设扫描顺序对最后一个非零系数的位置之前的所有系数进行解码, 确定当前块的系数。

- 10 可选地, 作为另一个实施例, 第二处理器 1503 还配置为在运行所述计算机程序时, 执行前述实施例中任一项所述的方法。

可以理解, 第二存储器 1502 与第一存储器 1302 的硬件功能类似, 第二处理器 1503 与第一处理器 1303 的硬件功能类似; 这里不再详述。

- 15 本实施例提供了一种解码器, 该解码器可以包括解析单元和第二确定单元。这样, 在高位宽、高码率、高质量或无损的视频场景中, 由于系数分布规律与常规视频编解码场景不同, 在系数编码中通过减少甚至消除上下文模式编码的语法元素个数, 从而可以减少码流中编码所带来的开销, 进而提高系数编码的吞吐量和编解码速度; 另外, 由于减少或消除的语法元素在高位宽、高码率、高质量或无损的视频编解码中的影响较小, 还能够提升压缩效率。

- 20 需要说明的是, 在本申请中, 术语“包括”、“包含”或者其任何其他变体意在涵盖非排他性的包含, 从而使得包括一系列要素的过程、方法、物品或者装置不仅包括那些要素, 而且还包括没有明确列出的其他要素, 或者是还包括为这种过程、方法、物品或者装置所固有的要素。在没有更多限制的情况下, 由语句“包括一个……”限定的要素, 并不排除在包括该要素的过程、方法、物品或者装置中还存在另外的相同要素。

上述本申请实施例序号仅仅为了描述, 不代表实施例的优劣。

- 25 本申请所提供的几个方法实施例中揭露的方法, 在不冲突的情况下可以任意组合, 得到新的方法实施例。

本申请所提供的几个产品实施例中揭露的特征, 在不冲突的情况下可以任意组合, 得到新的产品实施例。

- 30 本申请所提供的几个方法或设备实施例中揭露的特征, 在不冲突的情况下可以任意组合, 得到新的方法实施例或设备实施例。

以上所述, 仅为本申请的具体实施方式, 但本申请的保护范围并不局限于此, 任何熟悉本技术领域的技术人员在本申请揭露的技术范围内, 可轻易想到变化或替换, 都应涵盖在本申请的保护范围之内。因此, 本申请的保护范围应以所述权利要求的保护范围为准。

工业实用性

- 35 本申请实施例中, 无论是编码器还是解码器, 在高位宽、高码率、高质量或无损的视频编解码场景中, 由于系数分布规律与常规视频场景不同, 在系数编码中通过减少甚至消除上下文模式编码的语法元素个数, 比如关于最后一个非零系数位置、子块编码标识等语法元素, 甚至还可以在最后一个非零系数的坐标信息的值偏大时进行坐标变换, 从而可以减少码流中编码所带来的开销, 进而提高系数编码的吞吐量和编解码速度; 另外, 由于减少或消除的语法元素在高位宽、高码率、高质量或无损的视频编解码中的影响较小, 还能够提升压缩效率。
- 40

权利要求书

- 1、一种系数解码方法，应用于解码器，所述方法包括：
解析码流，获取视频标识信息；
当所述视频标识信息指示视频满足预设条件时，解析所述码流，获取最后一个非零系数位置翻转标识信息和最后一个非零系数的坐标信息；
当所述最后一个非零系数位置翻转标识信息指示当前块使用最后一个非零系数位置翻转时，对所述最后一个非零系数的坐标信息进行计算，确定所述最后一个非零系数的位置；
按照预设扫描顺序对所述最后一个非零系数的位置之前的所有系数进行解码，确定所述当前块的系数。
- 2、根据权利要求1所述的方法，其中，所述方法还包括：
当所述最后一个非零系数位置翻转标识信息指示所述当前块不使用最后一个非零系数位置翻转时，根据所述最后一个非零系数的坐标信息直接确定所述最后一个非零系数的位置；
按照预设扫描顺序对所述最后一个非零系数的位置之前的所有系数进行解码，确定所述当前块的系数。
- 3、根据权利要求1所述的方法，其中，所述方法还包括：
若所述视频标识信息的取值为第一值，则确定所述视频标识信息指示所述视频满足预设条件；或者，若所述视频标识信息的取值为第二值，则确定所述视频标识信息指示所述视频不满足预设条件。
- 4、根据权利要求3所述的方法，其中，所述预设条件至少包括下述之一：高位宽、高质量、高码率、高帧率和无损压缩。
- 5、根据权利要求1所述的方法，其中，所述方法还包括：
若所述最后一个非零系数位置翻转标识信息的取值为第一值，则确定所述最后一个非零系数位置翻转标识信息指示所述当前块使用最后一个非零系数位置翻转；或者，
若所述最后一个非零系数位置翻转标识信息的取值为第二值，则确定所述最后一个非零系数位置翻转标识信息指示所述当前块不使用最后一个非零系数位置翻转。
- 6、根据权利要求1所述的方法，其中，所述解析码流，获取最后一个非零系数的坐标信息，包括：
解析所述码流，获取最后一个非零系数的水平坐标的前缀信息、最后一个非零系数的竖直坐标的前缀信息、最后一个非零系数的水平坐标的后缀信息和最后一个非零系数的竖直坐标的后缀信息；
根据所述最后一个非零系数的水平坐标的前缀信息和所述最后一个非零系数的水平坐标的后缀信息，确定所述最后一个非零系数的水平坐标；
根据所述最后一个非零系数的竖直坐标的前缀信息和所述最后一个非零系数的竖直坐标的后缀信息，确定所述最后一个非零系数的竖直坐标；
根据所述最后一个非零系数的水平坐标和所述最后一个非零系数的竖直坐标，确定所述最后一个非零系数的坐标信息。
- 7、根据权利要求1所述的方法，其中，所述方法还包括：
当所述最后一个非零系数位置翻转标识信息指示所述当前块使用最后一个非零系数位置翻转时，确定所述最后一个非零系数的坐标信息为所述最后一个非零系数的位置相对于所述当前块的右下角位置之间的水平距离和竖直距离；
相应地，所述对所述最后一个非零系数的坐标信息进行计算，确定所述最后一个非零系数的位置，包括：
确定所述当前块的宽度和高度；
根据所述当前块的宽度与所述最后一个非零系数的位置相对于所述当前块的右下角位置之间的水平距离进行减法计算，得到所述最后一个非零系数的水平坐标；以及
根据所述当前块的高度与所述最后一个非零系数的位置相对于所述当前块的右下角位置之间的竖直距离进行减法计算，得到所述最后一个非零系数的竖直坐标；
根据所述最后一个非零系数的水平坐标和所述最后一个非零系数的竖直坐标，确定所述最后一个非零系数的位置。
- 8、根据权利要求1所述的方法，其中，所述方法还包括：
当所述最后一个非零系数位置翻转标识信息指示所述当前块不使用最后一个非零系数位置翻转时，确定所述最后一个非零系数的坐标信息为所述最后一个非零系数的位置相对于所述当前块的左上角位置之间的水平距离和竖直距离；

根据所述最后一个非零系数的位置相对于所述当前块的左上角位置之间的水平距离和竖直距离,确定所述最后一个非零系数的位置。

9、根据权利要求1所述的方法,其中,所述最后一个非零系数位置翻转标识信息至少为下述其中一项的标识信息:序列(sequence)级、图像(picture)级、片(slice)级和块(block)级。

5 10、根据权利要求1至9任一项所述的方法,其中,当所述视频标识信息指示视频满足预设条件时,所述方法还包括:

解析所述码流,获取最后一个系数使能标识信息;

当所述最后一个系数使能标识信息指示所述当前块使用最后一个系数位置时,按照预设扫描顺序对所述最后一个系数位置之前的所有系数进行解码,确定所述当前块的系数。

10 11、根据权利要求10所述的方法,其中,所述方法还包括:

若所述最后一个系数使能标识信息的取值为第一值,则确定所述最后一个系数使能标识信息指示所述当前块使用所述最后一个系数位置;或者,

若所述最后一个系数使能标识信息的取值为第二值,则确定所述最后一个系数使能标识信息指示所述当前块不使用所述最后一个系数位置。

15 12、根据权利要求11所述的方法,其中,当所述最后一个系数使能标识信息的取值为第二值时,所述方法还包括:

解析所述码流,获取最后一个非零系数的水平坐标的前缀信息、最后一个非零系数的竖直坐标的前缀信息、最后一个非零系数的水平坐标的后缀信息和最后一个非零系数的竖直坐标的后缀信息;

20 根据所述最后一个非零系数的水平坐标的前缀信息、所述最后一个非零系数的竖直坐标的前缀信息、所述最后一个非零系数的水平坐标的后缀信息和所述最后一个非零系数的竖直坐标的后缀信息,确定所述最后一个非零系数的位置;

按照预设扫描顺序对所述最后一个非零系数的位置之前的所有系数进行解码,确定所述当前块的系数。

25 13、根据权利要求10所述的方法,其中,所述最后一个系数位置是所述当前块中所有可能不为零的系数所组成的矩阵的右下角位置;或者,

所述最后一个系数位置是所述当前块根据预设扫描顺序执行扫描所有可能不为零的系数的最后位置。

14、根据权利要求10所述的方法,其中,所述方法还包括:

将最后一个非零系数的位置设置在所述最后一个系数位置。

30 15、根据权利要求10所述的方法,其中,所述方法还包括:

确定所述当前块在预设操作后得到变换块的宽度和高度;

根据所述变换块的宽度和高度进行坐标计算,得到所述变换块的右下角坐标信息;

根据所述变换块的右下角坐标信息,确定所述最后一个系数位置。

35 16、根据权利要求15所述的方法,其中,所述预设操作至少包括:强制置零(zero-out)操作。

17、根据权利要求15所述的方法,其中,所述方法还包括:

当将最后一个非零系数的位置设置在所述最后一个系数位置时,根据所述变换块的右下角坐标信息,确定所述最后一个非零系数的位置。

18、根据权利要求10所述的方法,其中,所述最后一个系数使能标识信息至少为下述其中一项的标识信息:序列级、图像级、片级和块级。

40 19、根据权利要求1至18任一项所述的方法,其中,当所述视频标识信息指示视频满足预设条件时,所述方法还包括:

解析所述码流,获取子块默认编码标识信息;

当所述子块默认编码标识信息指示所述当前块内的待解码子块默认编码时,确定子块编码标识信息的取值为第一值,对所述待解码子块内的所有系数进行解码。

45 20、根据权利要求19所述的方法,其中,当所述子块默认编码标识信息指示待解码子块不默认编码时,所述方法还包括:

解析所述码流,获取子块编码标识信息;

当所述子块编码标识信息的取值为第一值时,对所述待解码子块内的所有系数进行解码。

21、根据权利要求19所述的方法,其中,所述方法还包括:

50 若所述子块默认编码标识信息的取值为第一值,则确定所述子块默认编码标识信息指示所述待解码子块默认编码;或者,

若所述子块默认编码标识信息的取值为第二值,则确定所述子块默认编码标识信息指示所述待解码

子块不默认编码。

- 22、根据权利要求 19 所述的方法，其中，所述方法还包括：
若所述子块编码标识信息的取值为第一值，则确定对所述待解码子块内的所有系数进行解码；或者，
若所述子块编码标识信息的取值为第二值，则确定所述待解码子块内的所有系数均为零。
- 5 23、根据权利要求 19 所述的方法，其中，所述子块默认编码标识信息至少为下述其中一项的标识信息：序列级、图像级、片级和块级。
- 24、根据权利要求 3 或 5 或 11 或 21 或 22 所述的方法，其中，所述第一值为 1，所述第二值为 0。
- 25、根据权利要求 1 所述的方法，其中，当所述视频标识信息为高位宽标识信息时，所述方法还包括：
10 若所述高位宽标识信息指示所述视频满足高位宽，则确定所述视频满足预设条件。
- 26、根据权利要求 1 所述的方法，其中，当所述视频标识信息为高码率标识信息时，所述方法还包括：
若所述高码率标识信息指示所述视频满足高码率，则确定所述视频满足预设条件。
- 27、根据权利要求 1 所述的方法，其中，当所述视频标识信息为高质量标识信息时，所述方法还包括：
15 若所述高质量标识信息指示所述视频满足高质量，则确定所述视频满足预设条件。
- 28、根据权利要求 1 所述的方法，其中，当所述视频标识信息为无损压缩标识信息时，所述方法还包括：
若所述无损压缩标识信息指示所述视频满足无损压缩，则确定所述视频满足预设条件。
- 29、一种系数编码方法，应用于编码器，所述方法包括：
确定视频标识信息以及最后一个非零系数的位置；
当所述视频标识信息指示视频满足预设条件时，确定最后一个非零系数位置翻转标识信息；
根据所述最后一个非零系数的位置以及所述最后一个非零系数位置翻转标识信息，确定最后一个非
20 零系数的坐标信息；
- 按照预设扫描顺序对所述最后一个非零系数的位置之前的所有系数进行编码，并将编码后得到的比特信息、所述视频标识信息以及所述最后一个非零系数的坐标信息写入码流。
- 30、根据权利要求 29 所述的方法，其中，所述确定视频标识信息，包括：
若所述视频满足预设条件，则确定所述视频标识信息的取值为第一值；或者，
若所述视频不满足预设条件，则确定所述视频标识信息的取值为第二值。
- 31、根据权利要求 30 所述的方法，其中，所述预设条件至少包括下述之一：高位宽、高质量、高
30 码率、高帧率和无损压缩。
- 32、根据权利要求 29 所述的方法，其中，所述确定最后一个非零系数位置翻转标识信息，包括：
若当前块使用最后一个非零系数位置翻转，则确定所述最后一个非零系数位置翻转标识信息的取
35 值为第一值；或者，
若当前块不使用最后一个非零系数位置翻转，则确定所述最后一个非零系数位置翻转标识信息的取
值为第二值。
- 33、根据权利要求 32 所述的方法，其中，所述最后一个非零系数的位置包括所述最后一个非零系
数的初始水平坐标和初始竖直坐标，所述初始水平坐标和所述初始竖直坐标为所述最后一个非零系数的
位置相对于所述当前块的左上角位置之间的水平距离和竖直距离；
- 40 相应地，所述根据所述最后一个非零系数的位置以及所述最后一个非零系数位置翻转标识信息，确
定最后一个非零系数的坐标信息，包括：
若所述最后一个非零系数位置翻转标识信息的取值为第一值，则根据所述最后一个非零系数的初始
水平坐标和初始竖直坐标进行计算，确定所述最后一个非零系数的坐标信息；或者，
若所述最后一个非零系数位置翻转标识信息的取值为第二值，则根据所述最后一个非零系数的初始
45 水平坐标和初始竖直坐标直接确定所述最后一个非零系数的坐标信息。
- 34、根据权利要求 33 所述的方法，其中，所述根据所述最后一个非零系数的初始水平坐标和初始
竖直坐标进行计算，确定所述最后一个非零系数的坐标信息，包括：
确定所述当前块的宽度和高度；
根据所述当前块的宽度与所述最后一个非零系数的初始水平坐标进行减法计算，得到所述最后一个
50 非零系数的水平坐标；以及
根据所述当前块的高度与所述最后一个非零系数的初始竖直坐标进行减法计算，得到所述最后一个
非零系数的竖直坐标；

根据所述最后一个非零系数的水平坐标和所述最后一个非零系数的竖直坐标,确定所述最后一个非零系数的坐标信息。

35、根据权利要求 33 所述的方法,其中,所述方法还包括:

- 5 若所述最后一个非零系数位置翻转标识信息的取值为第一值,则确定所述最后一个非零系数的坐标信息为所述最后一个非零系数的位置相对于所述当前块的右下角位置之间的水平距离和竖直距离;或者,若所述最后一个非零系数位置翻转标识信息的取值为第二值,则所述最后一个非零系数的坐标信息为所述最后一个非零系数的位置相对于所述当前块的左上角位置之间的水平距离和竖直距离。

36、根据权利要求 29 所述的方法,其中,将所述最后一个非零系数的坐标信息写入码流,包括:

- 10 根据所述最后一个非零系数的坐标信息,确定最后一个非零系数的水平坐标的前缀信息、最后一个非零系数的竖直坐标的前缀信息、最后一个非零系数的水平坐标的后缀信息和最后一个非零系数的竖直坐标的后缀信息;

将所述最后一个非零系数的水平坐标的前缀信息、所述最后一个非零系数的竖直坐标的前缀信息、所述最后一个非零系数的水平坐标的后缀信息和所述最后一个非零系数的竖直坐标的后缀信息写入码流。

- 15 37、根据权利要求 29 所述的方法,其中,所述最后一个非零系数位置翻转标识信息至少为下述其中一项的标识信息:序列级、图像级、片级和块级。

38、根据权利要求 29 至 37 任一项所述的方法,其中,当所述视频标识信息指示视频满足预设条件时,所述方法还包括:

确定最后一个系数使能标识信息;

- 20 当所述最后一个系数使能标识信息指示所述当前块使用最后一个系数位置时,按照预设扫描顺序对所述最后一个系数位置之前的所有系数进行编码,并将编码后得到的比特信息、所述视频标识信息以及所述最后一个系数使能标识信息写入码流。

39、根据权利要求 38 所述的方法,其中,所述确定最后一个系数使能标识信息,包括:

- 25 若所述当前块使用所述最后一个系数位置,则确定所述最后一个系数使能标识信息的取值为第一值;或者,

若所述当前块不使用所述最后一个系数位置,则确定所述最后一个系数使能标识信息的取值为第二值。

40、根据权利要求 38 所述的方法,其中,所述最后一个系数位置是所述当前块中所有可能不为零的系数所组成的矩阵的右下角位置;或者,

- 30 所述最后一个系数位置是所述当前块根据预设扫描顺序执行扫描所有可能不为零的系数的最后位置。

41、根据权利要求 38 所述的方法,其中,所述方法还包括:

将最后一个非零系数的位置设置在所述最后一个系数位置。

42、根据权利要求 38 所述的方法,其中,所述方法还包括:

- 35 确定所述当前块在预设操作后得到变换块的宽度和高度;

根据所述变换块的宽度和高度进行坐标计算,得到所述变换块的右下角坐标信息;

根据所述变换块的右下角坐标信息,确定所述最后一个系数位置。

43、根据权利要求 42 所述的方法,其中,所述预设操作至少包括:强制置零(zero-out)操作。

44、根据权利要求 42 所述的方法,其中,所述方法还包括:

- 40 当将最后一个非零系数的位置设置在所述最后一个系数位置时,根据所述变换块的右下角坐标信息,确定所述最后一个非零系数的位置。

45、根据权利要求 39 所述的方法,其中,当所述当前块不使用所述最后一个系数位置时,所述方法还包括:

- 45 确定最后一个非零系数的水平坐标的前缀信息、最后一个非零系数的竖直坐标的前缀信息、最后一个非零系数的水平坐标的后缀信息和最后一个非零系数的竖直坐标的后缀信息;

根据所述最后一个非零系数的水平坐标的前缀信息、所述最后一个非零系数的竖直坐标的前缀信息、所述最后一个非零系数的水平坐标的后缀信息和所述最后一个非零系数的竖直坐标的后缀信息,确定所述最后一个非零系数的位置;

- 50 按照预设扫描顺序对所述最后一个非零系数的位置之前的所有系数进行编码,并将所述最后一个非零系数的水平坐标的前缀信息、所述最后一个非零系数的竖直坐标的前缀信息、所述最后一个非零系数的水平坐标的后缀信息和所述最后一个非零系数的竖直坐标的后缀信息写入码流。

46、根据权利要求 38 所述的方法,其中,所述最后一个系数使能标识信息至少为下述其中一项的

标识信息：序列级、图像级、片级和块级。

47、根据权利要求 29 至 46 任一项所述的方法，其中，当所述视频标识信息指示视频满足预设条件时，所述方法还包括：

确定所述当前块内的待编码子块的子块默认编码标识信息；

- 5 当所述子块默认编码标识信息指示所述待编码子块默认编码时，对所述待编码子块内的所有系数进行编码，并将编码后得到的比特信息以及所述子块默认编码标识信息写入码流。

48、根据权利要求 47 所述的方法，其中，所述方法还包括：

当所述子块默认编码标识信息指示所述待编码子块不默认编码时，确定所述待编码子块的子块编码标识信息，并将所述子块编码标识信息写入码流。

- 10 49、根据权利要求 47 所述的方法，其中，所述确定待编码子块的子块默认编码标识信息，包括：若所述待编码子块默认编码，则确定所述子块默认编码标识信息的取值为第一值；或者，若所述待编码子块不默认编码，则确定所述子块默认编码标识信息的取值为第二值。

50、根据权利要求 48 所述的方法，其中，所述确定所述待编码子块的子块编码标识信息，包括：若所述子块内需进行编码，则确定所述子块编码标识信息的取值为第一值；或者，

- 15 若所述子块内的所有系数均为零，则确定所述子块编码标识信息的取值为第二值。

51、根据权利要求 47 所述的方法，其中，所述子块默认编码标识信息至少为下述其中一项的标识信息：序列级、图像级、片级和块级。

52、根据权利要求 30 或 32 或 39 或 49 或 50 所述的方法，其中，所述第一值为 1，所述第二值为 0。

- 20 53、根据权利要求 29 所述的方法，其中，当所述视频标识信息为高位宽序列标识信息时，所述方法还包括：

若所述视频满足高位宽，则确定所述高位宽序列标识信息指示所述视频满足预设条件。

54、根据权利要求 29 所述的方法，其中，当所述视频标识信息为高码率标识信息时，所述方法还包括：

若所述视频满足高码率，则确定所述高码率标识信息指示所述视频满足预设条件。

- 25 55、根据权利要求 29 所述的方法，其中，当所述视频标识信息为高质量标识信息时，所述方法还包括：

若所述视频满足高质量，则确定所述高质量标识信息指示所述视频满足预设条件。

56、根据权利要求 29 所述的方法，其中，当所述视频标识信息为无损压缩标识信息时，所述方法还包括：

- 30 若所述视频满足无损压缩，则确定所述无损压缩标识信息指示所述视频满足预设条件。

57、一种编码器，所述编码器包括第一确定单元和编码单元；其中，

所述第一确定单元，配置为确定视频标识信息以及最后一个非零系数的位置；以及当所述视频标识信息指示视频满足预设条件时，确定最后一个非零系数位置翻转标识信息；

- 35 所述第一确定单元，还配置为根据所述最后一个非零系数的位置以及所述最后一个非零系数位置翻转标识信息，确定最后一个非零系数的坐标信息；

所述编码单元，配置为按照预设扫描顺序对所述最后一个非零系数的位置之前的所有系数进行编码，并将编码后得到的比特信息、所述视频标识信息以及所述最后一个非零系数的坐标信息写入码流。

58、一种编码器，所述编码器包括第一存储器和第一处理器；其中，

所述第一存储器，用于存储能够在所述第一处理器上运行的计算机程序；

- 40 所述第一处理器，用于在运行所述计算机程序时，执行如权利要求 29 至 56 任一项所述的方法。

59、一种解码器，所述解码器包括解析单元和第二确定单元；其中，

所述解析单元，配置为解析码流，获取视频标识信息；以及当所述视频标识信息指示视频满足预设条件时，解析所述码流，获取最后一个非零系数位置翻转标识信息和最后一个非零系数的坐标信息；

- 45 所述第二确定单元，配置为当所述最后一个非零系数位置翻转标识信息指示当前块使用最后一个非零系数位置翻转时，对所述最后一个非零系数的坐标信息进行计算，确定所述最后一个非零系数的位置；

所述解析单元，还配置为按照预设扫描顺序对所述最后一个非零系数的位置之前的所有系数进行解码，确定所述当前块的系数。

60、一种解码器，所述解码器包括第二存储器和第二处理器；其中，

所述第二存储器，用于存储能够在所述第二处理器上运行的计算机程序；

- 50 所述第二处理器，用于在运行所述计算机程序时，执行如权利要求 1 至 28 任一项所述的方法。

61、一种计算机存储介质，其中，所述计算机存储介质存储有计算机程序，所述计算机程序被执行时实现如权利要求 1 至 28 任一项所述的方法、或者实现如权利要求 29 至 56 任一项所述的方法。