

编解码方法、码流、编码器、解码器以及存储介质

技术领域

本申请实施例涉及视频编解码技术领域，尤其涉及一种编解码方法、码流、编码器、解码器以及存储介质。

5 背景技术

随着人们对视频显示质量要求的提高，高清和超高清等高分辨率视频应运而生。然而，高分辨率视频通常具有更多信息，因此需要更多带宽。为降低带宽要求，已经引入了涉及视频压缩的视频编码标准。

10 在视频编码标准中包括帧内预测和帧间预测。其中，由于很多情况下帧内预测不能完美地预测复杂的纹理，导致帧内编码的块通常有较多的残差；而在帧间预测中，帧间编码的块通常能从参考图像中找到较好的预测块，导致残差较少，但是帧间预测的变换过程考虑不全面，不利于处理各种角度的纹理特征，导致压缩效率低。

发明内容

本申请实施例提供一种编解码方法、码流、编码器、解码器以及存储介质，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：

15 第一方面，本申请实施例提供了一种解码方法，应用于解码器，该方法包括：

确定当前块的纹理特征索引；

根据纹理特征索引，确定当前块的变换核组；

根据变换核组，确定当前块的变换核；

确定当前块的变换系数，并根据变换核对当前块的变换系数进行反变换，确定当前块的残差块。

20 第二方面，本申请实施例提供了一种编码方法，应用于编码器，该方法包括：

确定当前块的纹理特征索引；

根据纹理特征索引，确定当前块的变换核组；

根据变换核组，确定当前块的变换核；

确定当前块的残差块，并根据变换核对当前块的残差块进行变换，确定当前块的变换系数；

25 对当前块的变换系数进行编码处理，将所得到的编码比特写入码流。

第三方面，本申请实施例提供了一种码流，其中，码流是根据待编码信息进行比特编码生成的；其中，待编码信息包括下述至少一项：当前块的量化系数、当前块的特征索引序号、第一语法标识信息的取值、第二语法标识信息的取值、第三语法标识信息的取值和第四语法标识信息的取值；

30 其中，第一语法标识信息用于指示当前块是否使用第一变换模式以及对应使用的变换核索引，第二语法标识信息用于指示当前序列是否允许使用第一变换模式，第三语法标识信息用于指示当前图像是否允许使用第一变换模式；第四语法标识信息用于指示当前片是否允许使用第一变换模式。

第四方面，本申请实施例提供了一种编码器，包括第一确定单元、变换单元和编码单元，其中：

第一确定单元，配置为确定当前块的纹理特征索引；根据纹理特征索引，确定当前块的变换核组；以及根据变换核组，确定当前块的变换核；

35 变换单元，配置为确定当前块的残差块，并根据变换核对当前块的残差块进行变换，确定当前块的变换系数；

编码单元，配置为对当前块的变换系数进行编码处理，将所得到的编码比特写入码流。

第五方面，本申请实施例提供了一种编码器，包括第一存储器和第一处理器，其中：

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；

40 第一处理器，用于在运行所述计算机程序时，执行如第二方面所述的方法。

第六方面，本申请实施例提供了一种解码器，包括第二确定单元和反变换单元，其中：

第二确定单元，配置为确定当前块的纹理特征索引；以及根据纹理特征索引，确定当前块的变换核组；

第二确定单元，还配置为根据变换核组，确定当前块的变换核；

反变换单元,配置为确定当前块的变换系数,并根据变换核对当前块的变换系数进行反变换,确定当前块的残差块。

第七方面,本申请实施例提供了一种解码器,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;

5 第二处理器,用于在运行所述计算机程序时,执行如第一方面所述的方法。

第八方面,本申请实施例提供了一种计算机可读存储介质,该计算机可读存储介质存储有计算机程序,所述计算机程序被至少一个处理器执行时实现如第一方面所述的方法、或者如第二方面所述的方法。

本申请实施例提供了一种编解码方法、码流、编码器、解码器以及存储介质,在编码端,确定当前块的纹理特征索引;根据纹理特征索引,确定当前块的变换核组;根据变换核组,确定当前块的变换核;确定当前块的残差块,并根据变换核对当前块的残差块进行变换,确定当前块的变换系数;对当前块的变换系数进行编码处理,将所得到的编码比特写入码流。在解码端,确定当前块的纹理特征索引;根据纹理特征索引,确定当前块的变换核组;根据变换核组,确定当前块的变换核;确定当前块的变换系数,并根据变换核对当前块的变换系数进行反变换,确定当前块的残差块。这样,无论是编码端还是解码端,都是首先确定当前块的纹理特征索引,然后根据纹理特征索引来确定当前块的变换核组,进而根据变换核组可以确定出当前块的变换核。也就是说,这里建立了纹理特征索引与变换核组之间的对应关系,替代了相关技术中根据帧内预测模式来匹配变换核组的方案,使得 LFNST 和 NSPT 也可以应用于帧间预测模式,从而针对帧间预测模式中的不好预测的块,即残差较大的块,不仅可以提高压缩效率,而且还可以提升编解码性能。

附图说明

- 20 图 1 为一种混合编码框架的流程框图示意图;
图 2 为一种图像组 GOP 的结构示意图;
图 3 为一种块划分结构示意图;
图 4 为一种当前块的空域/时域上的块位置关系示意图;
图 5 为一种当前块与同位块的运动信息示意图;
25 图 6 为一种使用同位参考图像的运动位移示意图;
图 7 为一种使用 MVD 特性的分布示意图;
图 8 为一种使用 2 个或 3 个控制点的仿射运动示意图;
图 9 为一种基于子块推导运动矢量的仿射运动示意图;
图 10 为一种 GPM 模式下的多种模式权重示意图;
30 图 11 为一种双边匹配的运动信息示意图;
图 12 为一种当前块的模板匹配示意图;
图 13 为一种当前块的参考像素示意图;
图 14 为一种当前块的多参考行示意图;
图 15 为一种帧内预测对应的多种预测模式示意图一;
35 图 16 为一种帧内预测对应的多种预测模式示意图二;
图 17 为一种帧内预测对应的多种预测模式示意图三;
图 18 为一种帧内预测对应的多种预测模式示意图四;
图 19 为一种屏幕内容的编码示意图;
图 20 为一种梯度与帧内预测模式的柱状图示意图;
40 图 21 为一种三种帧内预测模式的加权融合示意图;
图 22 为一种色度分量与亮度分量的参考像素示意图;
图 23 为一种多模型 CCLM 的分组示意图;
图 24 为一种多模型 CCLM 的预测示意图;
图 25 为一种帧内 CCCM 的参考区域示意图;
45 图 26 为一种 GLM 的亮度梯度计算示意图;
图 27 为一种帧间 CCCM 的预测示意图;
图 28 为一种 DCT 变换示意图;
图 29 为一种 DCT 变换的基图像示意图;
图 30 为一种无 LFNST 变换的流程示意图;
50 图 31 为一种有 LFNST 变换的流程示意图;

- 图 32 为一种有 LFNST 变换的详细流程示意图;
图 33 为一种多个变换核组的基图像示意图;
图 34 为一种 NSPT 变换的基图像示意图;
图 35 为一种亮度和色度分别划分的结构示意图;
5 图 36 为本申请实施例提供的一种视频编解码的网络架构示意图;
图 37 为本申请实施例提供的一种编码器的系统组成框图示意图;
图 38 为本申请实施例提供的一种解码器的系统组成框图示意图;
图 39 为本申请实施例提供的一种解码方法的流程示意图一;
图 40 为一种 GPM 模式的划分线示意图;
10 图 41 为本申请实施例提供的一种解码方法的流程示意图二;
图 42 为本申请实施例提供的一种解码方法的流程示意图三;
图 43 为本申请实施例提供的一种编码方法的流程示意图一;
图 44 为本申请实施例提供的一种编码方法的流程示意图二;
图 45 为本申请实施例提供的一种编码器的组成结构示意图;
15 图 46 为本申请实施例提供的一种编码器的硬件结构示意图;
图 47 为本申请实施例提供的一种解码器的组成结构示意图;
图 48 为本申请实施例提供的一种解码器的硬件结构示意图;
图 49 为本申请实施例提供的一种编解码系统的组成结构示意图。

具体实施方式

- 20 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。

- 25 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。

- 30 在视频图像中,一般采用第一颜色分量、第二颜色分量和第三颜色分量来表征编码块(Coding Block, CB)。其中,这三个颜色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。

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:

- 35 H.265/高效视频编码(High Efficiency Video Coding, HEVC);
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);
VVC 之后提升压缩性能的平台(Enhanced Compression Model, ECM);
联合视频专家组(Joint Video Experts Team, JVET);
40 编码单元(Coding Unit, CU);
编码树单元(Coding Tree Unit, CTU);
最大编码单元(Largest Coding Unit, LCU);
运动矢量(Motion Vector, MV);
预测单元(Prediction Unit, PU);
45 变换单元(Transform Unit, TU);
融合技术(Merge);
跳过技术(Skip);
量化参数(Quantization Parameter, QP);
使用运动矢量差的融合技术(Merge with Motion Vector Difference, MMVD);
50 运动矢量预测(Motion Vector Prediction, MVP);

时域运动矢量预测 (Temporal Motion Vector Prediction, TMVP);
基于子块的时域运动矢量预测 (Subblock-based Temporal Motion Vector Prediction, SbTMVP);
离散余弦变换 (Discrete Cosine Transform, DCT);
离散正弦变换 (Discrete Sine Transform, DST);
5 多变换选择 (Multiple Transform Selection, MTS);
低频不可分离的变换 (Low Frequency Non-Separable Transform, LFNST);
不可分离的基础变换 (Non-Separable Primary Transform, NSPT);
基于上下文的自适应二进制算术编码 (Context-based Adaptive Binary Arithmetic Coding, CABAC)。

目前,通用的视频编解码标准都采用基于块的混合编码框架。视频中的每一个图像或子图像或一帧
10 (frame)被分割成相同大小(如 256×256 , 128×128 , 64×64 等)的正方形的最大编码单元或编码树
单元。每个最大编码单元或编码树单元可根据规则划分成矩形的编码单元。编码单元可能还会划分为预
测单元、变换单元等。具体地,如图1所示,混合编码框架包括有预测(Prediction)、变换(Transform)
/量化(Quantization)、熵编码(Entropy Coding)、反量化(Inv.quantization)/反变换(Inv.transform)、
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
15 Prediction),帧间预测可以包括运动估计(Motion Estimation)和运动补偿(Motion Compensation)。由
于视频的一个图像中的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的方法消除
相邻像素之间的空间冗余。另外,由于视频中的相邻图像之间存在着很强的相似性,在视频编解码技术
中使用图像间预测方法消除相邻图像之间的时间冗余,从而提高编码效率。

视频编解码器的基本流程如下:在编码端,将一个图像划分成块,对当前块使用帧内预测或帧间预
20 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。在解码端,对当前块使用帧内预测或帧间预测产生
当前块的预测块,另一方面解析码流得到量化系数矩阵,对量化系数矩阵进行反量化与反变换得到残差
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。编码端同样需要和解码端类似的操作获得解码图像。解码图像可以为后续的图像作
25 为帧间预测的参考图像。编码端确定的块划分信息,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在输出到码流中。解码端通过解析码流及根据已有信息进行分析确定与编
码端相同的块划分信息,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
码端获得的解码图像和解码端获得的解码图像相同。编码端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
30 分可以不同。上述是基于块的混合编码框架下的视频编解码器的基本流程,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。

另外,在本申请实施例中,当前块(Current Block, CB)可以是当前编码单元、当前预测单元、或
者当前变换单元等。由于并行处理的需要,图像可以被划分成片(slice)等,同一个图像中的 slice 可
35 以并行处理,也就是说它们之间没有数据依赖。而“帧”是一种常用的说法,一般可以理解为帧是一
个图像。在本申请实施例中所述的帧也可以替换为图像或 slice 等。

可以理解,在预测技术中,下面分别针对帧内预测和帧间预测进行详细描述。

(一) 帧间预测。

帧间预测利用时间上的相关性来消除冗余。为了使人眼看不出卡顿,一般的视频的帧率会有 30 帧
40 每秒, 50 帧每秒, 60 帧每秒, 甚至 120 帧每秒。在这样的视频中,同一个场景下的相邻帧之间的相
关性很高,帧间预测技术利用这种相关性参考已经编解码的帧的内容对当前要编码的内容进行预测。帧间
预测可以极大的提升编码性能。

最基本的帧间预测方法是平移(translational)预测,平移预测假定当前要预测的内容在当前图像与
参考图像之间是平移运动的,比如说当前块(编码单元或预测单元)的内容在当前图像与参考图像之间
45 是平移运动的,那么就可以通过一个运动矢量从参考图像中找到这个内容,并把它作为当前块的预测块。
平移运动在视频中是占很大比例,静止不动的背景、整体平移的物体以及镜头的平移等都可以用平移预
测来处理。

自然视频中的一些内容不是简单的平移,比如说在平移的过程中有一些细微地变化,包括形状,颜
色等的变化。双向预测从参考图像中找到两个参考块,并将这两个参考块进行加权平均,以得到与当前
50 块尽可能像的预测块。比如说对某些场景,从当前帧的前面和后面各找一个参考块进行加权平均,可能
比单独一个参考块更像当前块。基于此双向预测在单向预测的基础上对压缩性能又有提升。

图像播放顺序(Picture Order Count, POC)可以作为图像的一个标识。在一段视频序列中,每一

个图像有唯一的 POC，本申请实施例中认为 POC 的顺序和播放顺序是相同的。P 图像 (P Frame) 是只能使用 POC 在当前图像之前的参考图像进行预测的图像。当前参考图像只有一个参考图像列表 (Reference Picture List, RPL)，记为 RPL0。参考图像列表 RPL0 中都是 POC 在当前图像之前的参考图像。B 图像 (B Frame) 早些时候是可以使用 POC 在当前图像之前的参考图像及 POC 在当前图像之后的参考图像进行预测的图像。B 图像有两个参考图像列表，记为 RPL0 和 RPL1。一种配置方法是 RPL0 中都是 POC 在当前图像之前的参考图像，RPL1 中都是 POC 在当前图像之后的参考图像。对于一个当前块，可以只参考 RPL0 中的某一图像的参考块，这样也叫做前向预测；也可以只参考 RPL1 中的某一图像的参考块，这样也叫做后向预测；也可以同时参考 RPL0 中的某一图像的参考块和 RPL1 中的某一图像的参考块，这样也叫做双向预测。同时参考两个参考块的一种简单的方法是将两个参考块每一个对应位置的像素进行平均得到当前块的预测块。后来 B 图像不再限制 RPL0 中都是 POC 在当前图像之前的参考图像，RPL1 中都是 POC 在当前图像之后的参考图像。所以 RPL0 中也可以有 POC 在当前图像之后的参考图像，RPL1 中也可以有 POC 在当前图像之前的参考图像。当前块也就可以同时参考 POC 在当前图像之前的参考图像或同时参考 POC 在当前图像之后的参考图像。这种 B 图像也叫广义 B 图像。

由于随机访问 (Random Access, RA) 配置的编解码顺序与 POC 顺序不同。这样 B 图像可以参考当前图像之前的信息和当前图像之后的信息从而明显提高了编码性能。示例性地，RA 的一种经典的图像组 (Group Of Pictures, GOP) 结构如图 2 所示。在图 2 中，箭头表示参考关系，由于 I 图像不需要参考图像，那么在 POC 为 0 的 I 图像解码后，将会解码 POC 为 4 的 P 图像，而在解码 POC 为 4 的 P 图像时可以参考 POC 为 0 的 I 图像。在 POC 为 4 的 P 图像解码后，然后再解码 POC 为 2 的 B 图像，而在解码 POC 为 2 的 B 图像时可以参考 POC 为 0 的 I 图像和 POC 为 4 的 P 图像，等等。这样，根据图 2 可以得到，POC 顺序为 {0 1 2 3 4 5 6 7 8} 的情况下，其对应的解码顺序为 {0 3 2 4 1 7 6 8 5}。

低延迟 (Low Delay, LD) 配置的编解码顺序与 POC 顺序相同。所以当前图像只能参考当前图像之前的信息。Low Delay 配置又分 Low Delay P 和 Low Delay B。Low Delay P 即传统的 Low Delay 配置。其典型的结构是 IPPP.....，即先编解码一个 I 图像，之后的图像都是 P 图像。Low Delay B 的典型结构是 IB BB.....，与 Low Delay P 的区别在于每个帧间图像都是 B 图像，即使用两个参考图像列表，当前块可以同时参考 RPL0 中的某一图像的参考块和 RPL1 中的某一图像的参考块。

通常情况下，RA 配置的压缩效率高于 LD 配置，LDB 配置的压缩效率高于 LDP 配置。一方面是因为双向预测可以参考到后向的信息，另一方面是因为双向预测可以通过一些技术减少预测误差，比如说加权平均等。

当前图像的一个参考图像列表最多可以有几个参考图像，如 2 个、3 个或 4 个等。当编码某一个当前图像时，RPL0 和 RPL1 中各有哪几个参考图像是由某种配置或算法决定的，不是本申请实施例所讨论的重点。但是同一个参考图像可能同时出现在 RPL0 和 RPL1 中。即编解码器允许当前块同时参考同一个参考图像的两个参考块。编解码器通常使用参考图像列表里的索引值 (index, idx) 来对应参考图像。如果一个参考图像列表长度为 4，则 index 有 0, 1, 2, 3 四个值。示例性地，如当前图像的 RPL0 有 POC 为 5, 4, 3, 0 的 4 个参考图像。则 RPL0 的 index 0 为 POC 5 的参考图像，RPL0 的 index 1 为 POC 4 的参考图像，RPL0 的 index 2 为 POC 3 的参考图像，RPL0 的 index 3 为 POC 0 的参考图像。

还可以理解，帧间预测使用运动信息 (motion information) 来表示“运动”。基本的运动信息包含参考图像 (reference picture) 的信息和运动矢量 (MV, motion vector) 的信息。一个块为了能使用双向预测，自然需要能找到 2 个参考块，那么就需要 2 组参考图像的信息和运动矢量的信息。可以把它们每一组理解为一个单向运动信息，而把这 2 组组合到一起就形成了一个双向运动信息。在具体实现时，单向运动信息和双向运动信息可以使用相同的数据结构，只是双向运动信息的 2 组参考帧的信息和运动矢量的信息都有效，而单向运动信息的其中一组参考帧的信息和运动矢量的信息是无效的。所述有效也可以说是“使用”，所述无效也可以说是“不使用”。

VVC 支持 2 个参考图像列表，记为 RPL0 与 RPL1。对上述的双向运动信息，VVC 使用 RPL0 对应的参考图像索引 refIdxL0，以及参 RPL0 对应的运动矢量 mvL0，RPL1 对应的参考图像索引 refIdxL1，以及 RPL1 对应的运动矢量 mvL1。这里的 RPL0 对应的参考图像索引，RPL1 对应的参考图像索引就可以理解为上述的参考图像信息。VVC 用两个标识信息来分别表示是否使用 RPL0 对应的运动信息以及是否使用 RPL1 对应的运动信息，分别记为 predFlagL0 和 predFlagL1。也可以理解为 predFlagL0 和 predFlagL1 表示上述单向运动信息“是否有效”。所以 VVC 中虽然没有明确地提到运动信息这种数据结构，但是它用每个参考图像列表对应的参考图像索引，运动矢量以及“是否有效”的标志位一起来表示运动信息。在 VVC 的标准文本中不出现运动信息，而是使用的运动矢量，也可以认为参考图像索引和是否使用对应运动信息的标志是运动矢量的附属。本文中为了描述方便仍然用“运动信息”，但是应当理解，

也可以用“运动矢量”来描述。“运动信息”也可以叫做“运动参数”。

对一个二维的图像来说，运动矢量可以用 (x, y) 来表示，即一个水平方向的分量和一个竖直方向的分量。由于视频都是以像素来表示的，像素之间是有距离的，一个物体的运动在相邻图像之间不一定总是能对应到整像素距离。举个例子，一个远景的视频，2个像素之间的距离在远景的物体上有1米，而这个物体在2帧之间的时间内运动的距离是0.5米，这种场景用整像素的运动矢量就无法很好地表示。因而运动矢量可以做到分像素级别，如1/2像素精度，1/4像素精度，1/8像素精度，1/16像素精度，来把运动表示得更精细。并通过插值的方法来得到参考图像中的分像素位置的像素值。

上述的平移预测中的单向预测和双向预测都是基于块的，如编码单元或预测单元。即把一个像素矩阵作为单位进行预测。最基本的块就是矩形块，如正方形和长方形。视频编解码标准如 HEVC、VVC 允许编码器根据视频的内容来确定编码单元、预测单元的大小及划分方式。纹理或运动简单的区域倾向于使用较大的块，纹理或运动复杂的区域倾向于使用较小的块。块划分的层次越深，越能划分出复杂的、更贴近于实际纹理或运动的块，但相应地用于表征这些划分的开销也就更大。运动信息也可能需要在码流中传输。而且通常情况下，块划分地越细，通常运动信息的开销越大。

最原始的运动信息表示方法是直接写入完整的运动信息。后来，专家发现可以使用运动矢量预测（motion vector prediction, MVP）加运动矢量差（motion vector difference, MVD）来表示运动矢量，即 $MV = MVP + MVD$ 。其中，MVP 越精准，MVD 就越小，这样在码流中占用的开销也就更小。

还可以理解，对于 merge 模式来说，每一个帧间编码的块都需要一个运动信息。为了简化问题，假设 CU 的划分等于 PU 的划分等于 TU 的划分，也就是一个编码单元有一个相同大小相同位置的预测单元、一个相同大小相同位置的变换单元。实际上随着 CU 划分更加灵活，VVC 相对于 HEVC 就有弱化 PU 和 TU 的倾向。预测、变换、量化、熵编码中某一个环节的差异都可能导致 CU 的划分。例如2个区域的运动信息不同，那么编码器可能把这2个区域划分为不同的 CU。又例如2个区域运动信息相同或相似，但是残差特性差别很大，那么编码器也可能把这2个区域划分为不同的 CU。如何划分是根据整体的压缩效率决定的，并不完全取决于某一个因素。因而会出现同一个物体或者说运动相同或相似的区域被划分成不同的 CU。

图3为示意性的 HEVC 的块划分结构。如图3所示，(a)为原始图像，(a)中有一根铁杆在按箭头所指的方向运动，背景区域运动较小。(b)为 HEVC 的块划分情况，(c)为去掉了(b)中运动信息相同的块的边界。可以看出很多相邻的块使用了相同的运动信息。这种情况下，如果对每一个块都单独编码运动信息会产生明显的浪费。上述提到 VVC 的完整的运动信息包括 RPL0 的参考图像索引、MV 和是否使用的标志，RPL1 的参考图像索引、MV 和是否使用的标志。合并 merge 模式的基本原理就是当前块可以继承相邻块的运动信息，包括参考图像的信息和运动矢量的信息。

Merge 模式可以构建一个合并候选列表，如果当前块使用 merge 模式，可以用一个 index 指示当前块合并哪一个运动信息，从而不需要编码完整的运动信息。构建合并候选列表时，可以加入当前块空域（spatial）上相邻块的运动信息，时域（temporal）上的运动信息，空域上不相邻块的运动信息，时域上不相邻块的运动信息，基于历史的运动信息，合成的运动信息等。

所述空域上的相邻块指同一个图像中与当前块相邻的块，所述空域上不相邻块指同一个图像中与当前块不相邻的块。所述时域上的运动信息以及时域上不相邻块的运动信息指在同位（collocated）参考图像上指定位置的运动信息。示例性地，如图4所示，黑色填充块为当前块，其中1、2、3、4、5位置为 merge 使用的空域上相邻的块的位置，其他点填充块对应位置为 merge 使用的空域上不相邻的块的位置；6位置为时域上的运动信息使用的位置，如果当前块右下角的对应的位置不可用，则使用当前块中心对应的位置；其他网格填充块对应位置为时域上的不相邻块的运动信息使用的位置。根据同位参考图像的对应位置上的运动信息导出时域运动信息，具体的导出方法详见下述介绍。需要注意的是，图4中的背景网格仅为像素坐标示意，并非具体的块划分。

基于历史的运动信息与位置无关，编解码器会维护一个先进先出的运动信息列表，每编解码一个块，编解码器就用该块的运动信息更新这个列表，更新列表时保证与列表中已有的运动信息不重复。基于历史的运动信息就从这个列表中获取。

还可以理解，对于时域运动信息（矢量）推导来说，时域运动信息预测被当作空域运动信息预测的补充来使用。通常来说同一个图像上的相邻区域的相关性比不同图像上的相关性更强。但是也有一些情况时域运动信息更好用。举一个简单的例子，比如当前图像中当前块和周边的相邻块属于不同的物体，它们有着截然不同的运动，而在某一个参考图像上与当前块属于同一个物体的块的运动，就能为当前块提供较好的运动信息预测。

同位参考图像上的同位块（这里把获取时域运动信息的那个块叫同位块）的运动矢量是从该同位参考图像 col_pic 到该同位块的参考图像 col_ref 之间的矢量。具体如图5所示，针对当前块，它需要的运

动矢量是从当前图像 curr_pic 到当前块的参考图像 curr_ref 之间的矢量。设 col_pic 和 col_ref 之间的 POC 距离是 td, curr_pic 和 curr_ref 之间的 POC 距离是 tb。假设同位块上的运动到当前块上的运动是不变的,那么可以根据 td, tb 来确定缩放的比例。若同位块的运动矢量为 (col_mv_x, col_mv_y), 则时域运动矢量预测 (tmvp_x, tmvp_y) 可以按如下推导: $tmvp_x = col_mv_x * tb / td$, $tmvp_y = col_mv_y * tb / td$ 。

5 在 VVC 中, 同位参考图像上的存储运动信息的最小单元是 4×4 。也就是每个 4×4 的子块存储一组运动信息。可以理解的是, 如果不考虑硬件实现的代价, 同位参考图像也可以做到每个像素存储一组运动信息。

还可以理解, VVC 中引入了基于子块的时域运动矢量预测, 即 SbTMVP。常规上说 MVP, TMVP 都是对整个块来说的, 也就是整个块共享同一个 MVP。而 SbTMVP 是基于子块的, 这样 SbTMVP 就能对每一个子块获得一个 MVP。这也是 SbTMVP 和 TMVP 的本质区别。

10 另一方面, TMVP 用当前块右下角的位置或当前块中心的位置来定位同位块, 而 SbTMVP 根据周边块的运动找一个运动偏移来确定位置。VVC 中, 如果 A1 位置的块参考了同位参考图像, 那么所述运动位移设为 A1 使用同位参考图像的运动矢量。否则, 所述运动位移设为 (0, 0)。如图 6 所示, 按运动位移找到位置, 然后将“同位块”中的对应于每一个子块的位置的 MV 进行缩放得到每一个子块的 MVP。

15 还可以理解, 对于使用运动矢量差的融合技术, 即 MMVD 来说, Merge 模式直接选中的 merge 候选列表中的运动信息作为当前块的运动信息。在实际的视频中, 当前块实际的运动矢量和选中的 merge 候选列表中的运动矢量有时会有一些差别。MMVD 是 VVC 中的一种特殊的 merge 模式, 它用一种高效的方法编码这种情况下的 MVD。普通的 Merge 不需要编解码 MVD。普通的 inter 模式需要直接编解码 MVD。而 MMVD 利用了 MVD 更多地分布在单水平方向或单竖直方向的特性, 数值小的 MVD 多, 数值越大的 MVD 越少的特性, 如图 7 所示。在图 7 中, 不同形状的圆可以表示不同数值的 MVD。

20 MMVD 只能表示一些特定的方向上的特定数值的 MVD, 它并不能表示任意的 MVD。它用 mmvd_direction_idx 表示 MVD 的方向, 当然也可以理解为 MVD 的 x 和 y 是否非零以及正负号, 用 mmvd_distance_idx 表示 MVD 的 x 和 y 非零的那一个的绝对值的大小 MmvdDistance。

25 在这里, 表 1 示出了一种示意性的 mmvd_distance_idx[x0][y0] 和 MmvdDistance[x0][y0] 的关系。

表 1

mmvd_distance_idx[x0][y0]	MmvdDistance[x0][y0]	
	ph_mmvd_fullpel_only_flag==0	ph_mmvd_fullpel_only_flag==1
0	1	4
1	2	8
2	4	16
3	8	32
4	16	64
5	32	128
6	64	256
7	128	512

其中, ph_mmvd_fullpel_only_flag 是一个图像头 flag, 可以设置 MMVD 的 2 个不同组合。

在这里, 表 2 示出了一种示意性的 mmvd_direction_idx[x0][y0] 和 MmvdSign[x0][y0] 的关系。

表 2

mmvd_direction_idx[x0][y0]	MmvdSign[x0][y0][0]	MmvdSign[x0][y0][1]
0	+1	0
1	-1	0
2	0	+1
3	0	-1

30 其中, MMVD 的 MVD 按如下方法得到:

$MmvdOffset[x0][y0][0] = (MmvdDistance[x0][y0] < 2) * MmvdSign[x0][y0][0]$;

$MmvdOffset[x0][y0][1] = (MmvdDistance[x0][y0] < 2) * MmvdSign[x0][y0][1]$ 。

还可以理解, 对于仿射 Affine 来说, 前面介绍了最简单常用的平移运动, 在现实世界中, 运动不只有平移, 还有很多种形式如缩小, 放大, 旋转, 运动的透视 (透视: 离镜头近的物体显得大, 离镜头远的物体显得小), 还有很多不规则的运动形式。Affine 可以用来表示比平移更复杂的运动。如图 8 所示, Affine 根据 2 个控制点 (4 个参数, 一个运动矢量包括 x, y 两个参数) 或者 3 个控制点 (6 个参数) 的运动矢量, 利用线性模型计算出当前块中每一个子块或者每一个像素的运动矢量。其中, (a) 提供了

2 个控制点的情形, 例如 v_0 、 v_1 ; (b) 提供了 3 个控制点的情形, 例如 v_0 、 v_1 、 v_2 。

对 4 参数 affine 模型, 当前块内 (x, y) 位置的运动矢量按如下公式推导如下:

$$\begin{cases} mv_x = \frac{mv_{1x}-mv_{0x}}{W}x + \frac{mv_{1y}-mv_{0y}}{W}y + mv_{0x} \\ mv_y = \frac{mv_{1y}-mv_{0y}}{W}x + \frac{mv_{1x}-mv_{0x}}{W}y + mv_{0y} \end{cases} \quad (1)$$

对 6 参数 affine 模型, 当前块内 (x, y) 位置的运动矢量按如下公式推导如下:

$$\begin{cases} mv_x = \frac{mv_{1x}-mv_{0x}}{W}x + \frac{mv_{2x}-mv_{0x}}{H}y + mv_{0x} \\ mv_y = \frac{mv_{1y}-mv_{0y}}{W}x + \frac{mv_{2y}-mv_{0y}}{H}y + mv_{0y} \end{cases} \quad (2)$$

其中, (mv_{0x}, mv_{0y}) 为当前块的左上角的控制点的运动矢量, (mv_{1x}, mv_{1y}) 为当前块的右上角的控制点的运动矢量, (mv_{2x}, mv_{2y}) 为当前块的左下角的控制点的运动矢量。

VVC 中使用的 Affine 为了简化硬件实现的复杂度, 将当前块划分为 4×4 的子块, 为每一个子块计算一个 MV 并进行运动补偿。图 9 是 Affine 基于子块推导运动矢量的一个示意图。可以理解的是, 随着硬件处理能力的增强, Affine 也可以做到基于像素的处理。即为每一个像素推导一个运动矢量, 根据该运动矢量对一个像素进行运动补偿。在这里, Affine 只需要几个控制点就可以为每一个子块或每一个像素推导出各自的运动矢量, 相对于基于整块的运动补偿, 它可以做到更精细的预测。而相对于划分更细小的 CU, Affine 的开销要小得多。

还可以理解, 对于几何划分模式 (Geometric Partitioning Mode, GPM) 来说, HEVC 支持最大 64×64 的 CTU, 可以递归地进行四叉树的划分。VVC 支持一套比 HEVC 更灵活的块划分方法, 支持最大 128×128 的 CTU, 包括四叉树、三叉树和二叉树的划分。这些划分方法。虽然块划分越来越灵活, 但不管是 CU、PU 还是 TU, 始终只能划分成矩形块。需要注意的是, VVC 已经弱化了 PU 和 TU 的划分。自然视频中纹理或者运动的边界是多种多样的。比如遇到一个斜向的物体边界, 如果单纯使用矩形块想要逼近边界就会划分出很多的小块, 这样会明显地增加开销。GPM 几何划分模式可以更好地处理自然视频中的纹理和边界。

GPM 使用两个与当前块大小相同的预测块, GPM 的预测块中某些像素位置 100% 使用第一个预测块对应位置的像素值, 某些像素位置 100% 使用第二个预测块对应位置的像素值, 而在交界区域或者称过渡区域, 按一定比例使用这两个预测块对应位置的像素值。交界区域的权重也是逐渐过渡的。当然应对诸如屏幕内容编码地场景, 也可以不使用过渡区域。具体这些权重如何分配, 由 GPM 的“划分”模式决定。根据 GPM 的“划分”模式确定每个像素位置的权重。当然在某些情况下, 比如说块尺寸很小的情况, 可能某些 GPM 的模式下不能保证一定有某些像素位置 100% 使用第一个预测块对应位置的像素值, 某些像素位置 100% 使用第二个预测块对应位置的像素值。也可以认为 GPM 使用两个与当前块大小不相同的预测块, 即各取所需的一部分, 将权重为 0 的部分剔除出来。

图 10 为 VVC 中的 GPM 模式在正方形的块上的 64 种模式的权重示意图。如图 10 所示, 黑色表示第一个预测块对应位置的权重值为 0%, 白色表示第一个预测块对应位置的权重值为 100%, 灰色区域则按颜色深浅的不同表示第一个预测块对应位置的权重值为大于 0% 小于 100% 的某一个权重值。第二个参考块对应位置的权重值则为 100% 减去第一个参考块对应位置的权重值。

GPM 可以说是一种预测模式或预测方法, 因为它最终产生一个预测块。也可以说 GPM 是一种“划分”模式, 它把预测块进行了模拟划分, 它类似于实现了 PU 划分, 但又没有实质地划分。上述 GPM 使用的第一个预测块和第二个预测块既可以是帧内预测产生的预测块, 也可以是帧间的单向预测产生的预测块, 还可以是帧间的双向预测产生的预测块。

还可以理解, 一般消费类的视频码率是受限的, 因而视频压缩通常要去寻求码流开销和失真的折衷。以块划分为例, 对相同的内容, 在一定范围内, 划分地越细开销越大失真越少; 划分越粗开销越少失真越大。以运动信息的编码为例, 对相同的内容, 在一定范围内, 运动信息越精准开销越大失真越小; 运动信息越粗放开销越小失真越大。一些解码端的方法在不占用开销情况下, 利用解码端侧的信息进行处理与计算, 达到改善运动信息, 改善预测效果, 减少失真的效果。不占用开销也就意味着没有编码器的根据原始图像做出的指示, 它是根据可以得到的信息自动处理的。VVC 中两个典型的解码端方法是解码端运动矢量改善 (Decoder side Motion Vector Refinement, DMVR) 和双向光流 (Bi-Directional Optical Flow, BDOF), 下面将对其分别进行详细介绍。

在一种可能的实现方式中, 对于 DMVR 来说, VVC 中的 DMVR 启动的一个条件是当前块的 2 个参考图像分别来自于当前图像的一前一后, 而且 2 个参考图像与当前图像的距离相等。另一个启动条件是当前 CU 使用的是整块的 merge 模式 (包括 skip), 所述整块的即不包含基于子块的 merge 如 SbTMVP 和 affine merge, 因为 merge 模式下运动矢量容易出现不够精确的情况。还有一些其他的条件这里不再

赘述。VVC 中的 DMVR 利用双边匹配 (bilateral matching, BM) 也就是对两边的参考块计算匹配代价, 如绝对差之和 (Sum of Absolute Difference, SAD)。DMVR 搜索原始 MV 周边的 MV 的匹配代价, 在移动时, 2 个参考图像的 MV 是按镜像移动的, 也就是在各自原始 MV 的基础上移动 MV_{diff} , 另一边移动 $-MV_{diff}$, 如图 11 所示。在图 11 中, 2 个参考图像包括参考图像 L0 (refPic in ListL0) 和参考图像 L1 (refPic in ListL1)。搜索时也支持分像素的搜索, 所以 DMVR 有可能找到比原有 MV 精度更高的 MV。按一定的规则进行搜索, 一般先搜索一定范围内的整像素 MV, 找到匹配代价最小的整像素 MV, 再在该整像素 MV 的基础上搜索分像素 MV。如果找到了比原始 MV 匹配代价更小的 MV, 则用匹配代价更小的 MV 做运动补偿预测。DMVR 改善后的 MV 理论上可以用于存储 MV 及周边块使用, 比如在当前块构建 merge 候选列表时, 如果周边块用 DMVR 改善了 MV, 使用改善后的 MV 用于构建 merge 候选列表能达到更好的压缩效果, 但是出于硬件实现的考虑, VVC 并没有这么做。

DMVR 是可以基于子块处理的。实际上在 VVC 中, 如果块的水平方向或竖直方向大于 16 像素, 会以 16 像素的大小分割子块。这一方面是基于硬件实现复杂度的考虑, 因为 DMVR 需要在解码端进行搜索, 而限制子块的大小可以减少缓存的代价。另一方面, 划分成子块处理提供了更好的灵活性, 每个子块可以独立改善 MV, 从一定程度上达到了提高划分精度的效果, 这也提升了压缩效率。

在另一种可能的实现方式中, 对于 BDOF 来说, BDOF 也是一个典型的解码端方法。正如它的名字, BDOF 基于光流原理改善 MV 和预测。光流是空间运动物体在观察成像平面上的像素运动的瞬时速度。光流有一些基本的假设, 如亮度恒定不变, 即同一目标在不同图像之间运动时, 其亮度不发生变化。时间连续或运动是小运动。即时间的变化不会引起目标位置的剧烈变化。

VVC 中的 BDOF 启动的一个条件是当前块的 2 个参考图像分别来自于当前图像的一前一后, 而且 2 个参考图像与当前图像的距离相等。VVC 对每一个 4×4 子块, BDOF 会推导出一个运动矢量偏差 (v_x, v_y) , 这个偏差是通过最小化两个方向的预测值的差值计算得到的。这个运动矢量偏差也被用于调整对应子块里的预测值。其推导过程如下:

首先, 计算 2 个预测块的水平方向和竖直方向的梯度 $\frac{\partial I^{(k)}}{\partial x}(i, j)$ 和 $\frac{\partial I^{(k)}}{\partial y}(i, j)$, $k = 0, 1$; 具体如下:

$$\frac{\partial I^{(k)}}{\partial x}(i, j) = (I^{(k)}(i+1, j) \gg \text{shift1}) - (I^{(k)}(i-1, j) \gg \text{shift1}) \quad (3)$$

$$\frac{\partial I^{(k)}}{\partial y}(i, j) = (I^{(k)}(i, j+1) \gg \text{shift1}) - (I^{(k)}(i, j-1) \gg \text{shift1}) \quad (4)$$

其中, $I^{(k)}(i, j)$ 是参考图像列表 k , $k = 0, 1$ 的坐标 (i, j) 的预测值, shift1 根据亮度的比特深度 bitDepth 计算, $\text{shift1} = \max(6, \text{bitDepth}-6)$ 。

其次, 计算 S_1 、 S_2 、 S_3 、 S_5 和 S_6 , 具体如下:

$$S_1 = \sum_{(i,j) \in \Omega} \text{Abs}(\psi_x(i, j)) \quad (5)$$

$$S_2 = \sum_{(i,j) \in \Omega} \psi_x(i, j) \cdot \text{Sign}(\psi_y(i, j)) \quad (6)$$

$$S_3 = \sum_{(i,j) \in \Omega} \theta(i, j) \cdot \text{Sign}(\psi_x(i, j)) \quad (7)$$

$$S_5 = \sum_{(i,j) \in \Omega} \text{Abs}(\psi_y(i, j)) \quad (8)$$

$$S_6 = \sum_{(i,j) \in \Omega} \theta(i, j) \cdot \text{Sign}(\psi_y(i, j)) \quad (9)$$

其中,

$$\psi_x(i, j) = \left(\frac{\partial I^{(1)}}{\partial x}(i, j) + \frac{\partial I^{(0)}}{\partial x}(i, j) \right) \gg n_a \quad (10)$$

$$\psi_y(i, j) = \left(\frac{\partial I^{(1)}}{\partial y}(i, j) + \frac{\partial I^{(0)}}{\partial y}(i, j) \right) \gg n_a \quad (11)$$

$$\theta(i, j) = (I^{(1)}(i, j) \gg n_b) - (I^{(0)}(i, j) \gg n_b) \quad (12)$$

其中, Ω 是一个当前 4×4 子块周围的 6×6 的窗, n_a 为 $\min(1, \text{bitDepth}-11)$, n_b 为 $\min(4, \text{bitDepth}-8)$ 。

再次, 运动矢量偏差 (v_x, v_y) 计算如下:

$$v_x = S_1 > 0 ? \text{clip3}(-th'_{BIO}, th'_{BIO}, -((S_3 \cdot 2^{n_b-n_a}) \gg \lfloor \log_2 S_1 \rfloor)) : 0 \quad (13)$$

$$v_y = S_5 > 0 ? \text{clip3}(-th'_{BIO}, th'_{BIO}, -((S_6 \cdot 2^{n_b-n_a} - ((v_x S_{2,m}) \ll n_{S_2} + v_x S_{2,s})/2) \gg \lfloor \log_2 S_5 \rfloor)) : 0 \quad (14)$$

其中, $S_{2,m} = S_2 \gg n_{S_2}$, $S_{2,s} = S_2 \& (2^{n_{S_2}} - 1)$, $th'_{BIO} = 2^{\max(5, BD-7)}$, $\lfloor \cdot \rfloor$ 是向下取整, $n_{S_2} = 12$, BD 是比特深度 bitDepth 。

再次, 根据运动矢量偏差和梯度, 该 4×4 子块内的每一个预测值调整如下:

$$b(x, y) = \text{rnd} \left(\left(v_x \left(\frac{\partial I^{(1)}(x, y)}{\partial x} - \frac{\partial I^{(0)}(x, y)}{\partial x} \right) + v_y \left(\frac{\partial I^{(1)}(x, y)}{\partial y} - \frac{\partial I^{(0)}(x, y)}{\partial y} \right) + 1 \right) / 2 \right) \quad (15)$$

最终 BDOF 的预测值计算如下：

$$\text{pred}_{\text{BDOF}}(x, y) = (I^{(0)}(x, y) + I^{(1)}(x, y) + b(x, y) + o_{\text{offset}}) \gg \text{shift} \quad (16)$$

其中， o_{offset} 和 shift 根据亮度的比特深度计算得到。 n_a 、 n_b 和 shift 都是为了降低计算过程中的位宽做的处理。

这样，BDOF 的运动矢量偏差可以做到很高的精度从而使预测更准确，而基于子块的处理也提高了灵活性，这两个方面和 DMVR 是类似的。

还可以理解，DMVR 和 BDOF 都有改善运动矢量的效果。DMVR 基于块匹配，BDOF 基于光流原理。它们是可以组合使用的。示例性地，可以称之为多轮解码端运动矢量改善（Multi-pass Decoder-side Motion Vector Refinement, MDMVR）。

第一步，基于整块的双向匹配的运动矢量改善。第二步，基于子块的双向匹配的运动矢量改善。这一步的字块大小可以是 16×16 。第三步，基于子块的双向光流的运动矢量改善。这一步的子块大小可以是 8×8 。当前在此基础上还可以进一步丰富步骤，如再来一个第四步，基于 4×4 子块的双向光流的运动矢量改善。或者再来一个基于点的双向光流的运动矢量改善等。

还可以理解，模板匹配（Template Matching, TM）的方法最早用在帧间预测中，它利用相邻像素之间的相关性，把当前块周边的一些区域作为模板。在当前块进行编解码时，按照编码顺序其左侧及上侧已经编解码完成。当然在现有的硬件解码器实现时，不一定能保证当前块开始解码时，其左侧和上侧已经解码完成，当然这里说的是帧间块，比如在 HEVC 中帧间编码的块产生预测块时是不需要周边的重建像素的，因而帧间块的预测过程可以并行进行。但是帧内编码的块是一定需要左侧和上侧的重建像素作为参考像素的。理论上左侧和上侧是可得的，也就是说硬件设计做相应的调整是可以实现的。相对来说右侧和下侧在现在标准如 VVC 的编码顺序下是不可得的。

如图 12 所示，把当前块的左侧和上侧的矩形区域设为模板，左侧的模板部分的高度和当前块的高度相同，上侧的模板的部分的宽度一般和当前块的宽度相同，当然也可以不同。在参考图像 L0 中寻找模板的最佳匹配位置从而确定当前块的运动信息或者说运动矢量。这个过程大致可以描述为，在某个参考图像中，从一个起始位置开始，在周边一定范围内进行搜索。可以预先设定好搜索的规则，如搜索范围搜索步长等。每移动一个到位置，计算该位置对应的模板和当前块周边的模板的匹配程度，所谓匹配程度可以用一些失真代价来衡量，比如说绝对差之和（Sum of Absolute Difference, SAD）、绝对变换差之和（Sum of Absolute Transformed Difference, SATD）、均方误差（Mean-Square Error, MSE）等，一般 SATD 使用的变换是哈达玛（Hadamard）变换，另外 SAD、SATD、MSE 等的值越小代表匹配程度越高。用该位置对应的模板的预测块和当前块周边的模板的重建块计算代价。除了整像素位置的搜索还可以进行亚像素位置的搜索，根据搜索到的匹配程度最高的位置来确定当前块的运动信息。利用相邻像素之间的相关性，对模板合适的运动信息可能也是当前块合适的运动信息。当然模板匹配的方法可能并不一定对所有的块都适用，因而可以使用一些方法确定当前块是否使用上述模板匹配的方法，比如在当前块用一个控制开关表示是否使用模板匹配的方法。这种模板匹配的方法的一个名字叫解码端运动矢量推导（Decoder side Motion Vector Derivation, DMVD）。编码器和解码器都可以利用模板进行搜索从而导出运动信息或者在原有的运动信息的基础上找到更好的运动信息。而它不需要传输具体的运动矢量或运动矢量差，而是由编码器和解码器都进行同样规则的搜索从而保证编码和解码的一致。模板匹配的方法可以提高压缩性能，但是它需要在解码端也进行“搜索”，从而带来了一定的解码端复杂度。

（二）帧内预测。

图像内部相邻的部分或相邻像素之间存在着很强的空间相关性，帧内预测就是利用当前块周边已编解码的像素和当前块内部的像素的空间相关性的预测方法。示例性地，如图 13 所示， 4×4 的白色填充像素是当前块，当前块左侧一列和上侧一行的网格填充像素为当前块的参考像素，帧内预测使用这些参考像素对当前块进行预测。这些参考像素可能已经全部可得，即全部已经编解码。也可能有部分不可得，比如当前块是整帧的最左边，那么当前块的左侧的参考像素不可得。或者编解码当前块时，当前块左下方的部分还没有编解码，那么左下方的参考像素也不可得。对于参考像素不可得的情况，可以使用可得的参考像素或某些值或某些方法进行填充，或者不进行填充。

还需要说明的是，多参考行（Multiple reference line, MRL）的帧内预测方法可以使用更多的参考像素从而提高编码效率。如图 14 所示，这里为使用 4 个参考行/列的示意图。

帧内预测有多种预测模式，如图 15 所示，这里是 H.264 中对 4×4 的块进行帧内预测的 9 种模式。其中，模式 0（垂直模式）是将当前块上面的像素按竖直方向复制到当前块作为预测值，模式 1（水平模式）是将左侧的参考像素按水平方向复制到当前块作为预测值，模式 2（DC 模式）是将 A~D 和 I~L

这8个点的平均值作为所有点的预测值,模式3~8分别按某一个角度将参考像素复制到当前块的对应位置,因为当前块某些位置不能正好对应到参考像素,可能需要使用参考像素的加权平均值,或者说是插值的参考像素的分像素。

除此之外,还有 PLANE, PLANAR 等模式,而随着技术的发展以及块的扩大,角度预测模式也越来越多。如 HEVC 使用的帧内预测模式有 PLANAR、DC 和 33 种角度模式共 35 种预测模式,详见图 16。VVC 使用的帧内模式有 PLANAR、DC 和 65 种角度模式共 67 种预测模式,详见图 17。当然,除了上述 67 种模式,VVC 对一些长和宽差距较大的长方形的块还提供了宽角度模式,如图中的虚线所指的模式即-14~-1 和 67~80 两个区间,它们会替换掉一些常规的模式,详见图 18。

还可以理解,对于帧内块复制(Intra Block Copy, IBC)来说,IBC 能够明显提升屏幕内容编码(Screen Content Coding, SCC)的压缩效率,因而 IBC 从 HEVC 到 VVC 都被用于屏幕内容编码。屏幕内容区别于相机采集的内容(camera captured content),它是由计算机生成的,屏幕内容没有噪声,包含文字、计算机图形等,边界清晰。屏幕内容中存在大量重复的内容,如图 19 所示。

在本申请实施例中,可以认为 IBC 是把帧间预测的方法用到了帧内预测。其中,帧间预测把参考图像上的参考块用来生成当前块的预测块,参考图像不是当前图像。而 IBC 则是从当前图像中已编解码的部分或者叫已重建部分找参考块用来生成当前块的预测块。IBC 也可称为帧内块补偿(intra picture block compensation)或当前图像参考(Current Picture Referencing, CPR)。

IBC 可以用块矢量(block vector, BV)来表示当前块和参考块之间的位置差别,这和帧间预测的 MV 是类似的。编码器在搜索范围内通过块匹配的方法确定当前块的最佳的匹配块,并对 BV 进行编码,对 BV 的编码有多种方法,比如可以用 merge 模式,和帧间预测有类似之处,这里不再赘述。

IBC 可以认为是一种帧内预测方法,也可以认为是独立于帧内预测和帧间预测的另一类预测方法。IBC 对屏幕内容编码有很高的效率,在相机采集的自然序列中同样能提升压缩效率。

在 ECM10 中的帧内模板匹配预测(Intra Template Matching Prediction, IntraTMP)可以认为是一种特殊的 IBC, IntraTMP 也是要从当前图像中已编解码的部分中寻找参考块用来生成当前块的预测块。IntraTMP 把当前块左侧和上侧一定范围内的重建像素作为当前块的模板,每搜索一个 BV,该 BV 对应的与当前块相同大小的参考块的左侧和上侧一定范围内的重建像素参考块的模板,计算当前块的模板和参考块的模板的匹配代价,根据匹配代价确定参考块生成预测块。

还可以理解,对于解码端帧内预测模式导出(Decoder-side Intra Mode Derivation, DIMD)来说, DIMD 利用当前块左侧和上侧的重建像素导出预测模式,但是它不是在模板上进行预测,而是分析重建像素的梯度。

如图 20 所示, DIMD 分析黑色点的梯度,如水平梯度和竖直梯度,根据它的梯度适配一种帧内预测模式,对所有需要检查的点分析可以得到一个类似于下面的柱状图的结果。即每种帧内预测模式匹配的点的数量的统计。当然所谓柱状图只是帮助理解,具体实现时可以用多种简单的形式实现。现在的 DIMD 选出柱状图里最高的 2 个帧内预测模式,再加上 PLANAR 模式,共 3 个帧内预测模式的预测值进行加权,权重和分析的结果有关。示例性地,如图 21 所示,3 个帧内预测模式包括 M_1 模式、 M_2 模式和 PLANAR 模式。对于这 3 个帧内预测模式所得到的预测值分别设置为 $Pred_1$ 、 $Pred_2$ 、 $Pred_3$, 这 3 个帧内预测模式的权重值分别设置为 w_1 、 w_2 、 w_3 , 具体计算公式如下:

$$w_1 = \frac{43}{64} \times \frac{ampl(M_1)}{ampl(M_1) + ampl(M_2)} \quad (17)$$

$$w_2 = \frac{43}{64} \times \frac{ampl(M_2)}{ampl(M_1) + ampl(M_2)} \quad (18)$$

$$w_3 = \frac{21}{64} \quad (19)$$

最终的预测块,可以如下述所示:

$$Pred = \sum_{i=1}^3 w_i \times Pred_i \quad (20)$$

总结来说, DIMD 利用重建像素的梯度分析来筛选帧内预测模式,而且可以将 2 个帧内预测模式再加上 planar 根据分析结果进行加权。DIMD 的好处在于如果当前块选择了 DIMD 模式,那么它不需要再去指示具体使用了哪种帧内预测模式,而是由解码器自己通过上述流程导出,一定程度上节省了开销。

还可以理解,对于跨分量预测(Cross-Component Prediction, CCP)来说,由于同一空间的不同分量间具有很强的相关性,视频编解码技术可以利用这种相关性来提高压缩效率。有些情况下,同一空间的第一分量会先编解码,然后第二第三分量再编解码,这样第二第三分量就可以利用第一分量的一些信息。比如在 YUV 格式中,U 和 V 的像素可以用对应位置的 Y 的重建像素来预测。如果是 YUV4:2:0 格式的, U 和 V 的像素位置和 Y 的位置不一一对应,可以通过降采样等方法找到对应的 Y 的重建像素来预测。

VVC 中的 CCP 的一个典型的示例是跨分量线性模型(Cross-Component Linear Model, CCLM)。ECM 中其他跨分量预测技术还有多模型 CCLM (Multi-Model CCLM, MM-CCLM)、卷积跨分量模型

(Convolutional Cross-Component Model, CCCM)、多模型 CCCM (Multi-Model CCCM, MM-CCCM)、梯度线性模型 (Gradient Linear Model, GLM)、inter-CCCM 等, 还有其他衍生技术不再一一列举, 它们都是第二/三颜色分量利用第一颜色分量的一些信息, 如重建值, 进行预测的技术。跨分量预测技术可以用在帧内编码的块也可以用在帧间编码的块。例如 inter-CCCM 就是用在帧间编码的块。

5 (1) 跨分量线性模型 CCLM。

CCLM 顾名思义是用一个第一颜色分量和第二/三颜色分量的线性模型进行预测的技术。具体地, 如下所示:

$$\text{pred}_C(i, j) = \alpha \cdot \text{rec}_L'(i, j) + \beta \quad (21)$$

其中, $\text{pred}_C(i, j)$ 代表 (i, j) 位置的色度像素的预测值, 代表 $\text{rec}_L'(i, j)$ 位置的经过降采样的亮度像素的重建值。CCLM 的模型参数 $(\alpha$ 和 $\beta)$ 是根据当前块的相邻的经过降采样的亮度像素和色度像素导出的。示例性, 如图 22 所示, 这里示出了当前块的亮度分量相邻参考值和色度分量相邻参考值的采样示意图; 其中, 在 (a) 中, 加粗的较大方框用于突出指示色度块 21, 而灰色实心圆圈用于指示色度块 21 的相邻参考值; 在 (b) 中, 加粗的较大方框用于突出指示亮度块 22, 而灰色实心圆圈用于指示亮度块 22 的相邻参考值。其中, 色度块 21 为 $N \times N$ 大小, 亮度块 22 为 $2N \times 2N$ 大小。在这里, 色度块 21 的相邻参考值与亮度块 22 的相邻参考值均是用于推导模型参数 α 和 β 。

基本的原理是利用当前块相邻的亮度和色度像素导出线性模型, 再将这个线性模型用在当前块的上, 根据亮度的重建值和这个线性模型确定色度的预测值。

(2) 多模型 CCLM。

MM-CCLM 是 CCLM 的一个扩展, CCLM 在当前块中只使用一个线性模型, MM-CCLM 顾名思义使用多个线性模型, 具体地在 ECM-10 中使用 2 个线性模型。为了导出 2 个线性模型, 当前块用于导出线性模型的左侧和上侧相邻的亮度像素被分为 2 组, 分组的阈值是这些亮度像素值的中值。对当前块进行预测时, 同样用该阈值把当前块的亮度像素分为 2 组, 分别应用各自的模型进行预测。图 23 为一种 MM-CCLM 分组示意图, 图 24 为一种 MM-CCLM 预测示意图。

(3) 卷积跨分量模型 CCCM。

和 CCLM 类似的是 CCCM 也是根据当前块左侧和上侧已重建的部分导出跨分量模型, 但是 CCCM 使用的重建区域要更大一些, 它可以导出一个非线性的模型。另一个不同的是 CCLM 是用一个降采样后的亮度值预测一个色度值, CCCM 要用到色度像素对应的位置的亮度重建值以及其周边上下左右位置的亮度重建值。

具体地, CCCM 用一个 7 抽头的卷积滤波器, 该滤波器包括 5 个空域相邻的像素, C (center) 是当前色度像素对应的亮度像素, N (north) 是 C 上面的亮度像素, S (south) 是 C 下面的亮度像素, W (west) 是 C 左面的亮度像素, E 是 C 右面的亮度像素。还有一个非线性项 P, $P = (C \times C + \text{midVal}) \gg \text{bitDepth}$ 。其中, midVal 即中值的意思, bitDepth 是位宽。对 10 位的情况, $P = (C \times C + 512) \gg 10$ 。还有一个偏移项 B, 设为中值, 对 10 位的情况是 512。色度像素的预测值 $\text{predChromaVal} = C_0C + C_1N + C_2S + C_3E + C_4W + C_5P + C_6B$ 。

为了确定 C_0 、 C_1 、 C_2 、 C_3 、 C_4 、 C_5 、 C_6 , CCCM 会在如图 25 所示的参考区域中应用同样的模型对色度进行预测并且和重建的色度值比较, 求解使均方误差最小的 C_0 、 C_1 、 C_2 、 C_3 、 C_4 、 C_5 、 C_6 。

(4) 多模型卷积跨分量模型 MM-CCCM。

和 MM-CCLM 类似, 在当前块应用多个模型。其中, 在 ECM-10 中使用的是 2 个, 导出方法也是用一个阈值将已重建的亮度像素和当前块的亮度像素分成 2 组, 分别导出 CCCM 模型, 分别预测并组合成一个预测块。

(5) 梯度线性模型 GLM。

GLM 是根据亮度的梯度来预测色度的方法。这里有 2 种 GLM 模式, 一种模式是 2 参数 GLM, 一种模式是 3 参数 GLM。其中:

2 参数 GLM 的公式为 $\text{pred}_C(i, j) = \alpha \cdot \text{grad}_L(i, j) + \beta$;

45 3 参数 GLM 的公式为 $\text{pred}_C(i, j) = \alpha_0 \cdot \text{grad}_L(i, j) + \alpha_1 \cdot \text{rec}_L'(i, j) + \beta$ 。

在这里, 2 参数 GLM 就是把 CCLM 中的下采样的亮度的重建像素值换成了亮度的梯度。3 参数 GLM 就是在 CCLM 的基础上加上了亮度的梯度。亮度的梯度有如图 26 所示四种计算方法, 滤波器方块对应的是亮度像素, 圆圈对应的是色度像素, 这是一组为 YUV4:2:0 格式设计的滤波器。

(6) 帧间的 CCCM。

50 帧内预测和帧间预测是不同的, 帧内预测使用的是当前块周边已重建的像素来预测当前块内部, 而帧间预测使用的是不同时刻的相同物体的图像来预测。对帧间预测来说最基本的是平移运动, 也就是用一个运动矢量找到一个参考块, 把参考块作为预测块。既然是从不同时刻找到的相同物体, 参考图像中

包含与当前图像相同的分量数,对 YUV 格式的序列来说,YUV 三个分量都可以从参考块中得到预测值。上述跨分量模型如 CCLM,CCCM 都遵循类似帧内的方式,即用周边重建像素来导出模型,而 inter-CCCM 用原有帧间预测的预测块导出跨分量模型(滤波器系数),如图 27 所示。其中, resY、resCb、resCr 分别表示亮度残差值、蓝色色度残差值和红色色度残差值;然后结合对应的预测值进行加法运算,可以得到亮度重建值 Y、蓝色色度重建值 Cb 和红色色度重建值 Cr。也就是说,在亮度进行重建之后,再对亮度重建值应用导出的模型(滤波器系数)得到第二色度预测值,将原来的预测值和 inter-CCCM 得到的第二预测值进行加权作为新的预测值。inter-CCCM 把包含亮度的残差的重建值经过模型转化给色度的预测,从而提升了压缩效率。

进一步地,下面针对变换技术进行相关介绍。
编码时,现在通用的混合编码框架会先进行预测,预测利用空间或者时间上的相关性能得到一个跟当前块相同或相似的图像。对一个块来说,预测块和当前块是完全相同的情况是有可能出现的,但是很难保证一个视频中的所有块都如此,特别是对自然视频,或者说相机拍摄的视频。视频中不规则的运动,扭曲形变,遮挡,亮度等的变化,很难被完全预测。所以混合编码框架会将当前块的原始图像减去预测图像得到残差图像,或者说当前块减去预测块得到残差块。残差块通常要比原始图像简单很多,因而预测可以显著提升压缩效率。对残差块也不是直接进行编码,而是通常先进行变换。变换是把残差图像从空间域变换到频率域,去除残差图像的相关性。残差图像变换到频率域以后,由于能量大多集中在低频区域,变换后的非零系数大多集中在左上角。接下来利用量化来进一步压缩。而且由于人眼对高频不敏感,高频区域可以使用更大的量化步长。

图 28 为一个 DCT 变换的示意图。如图 28 所示,原始图像经过 DCT 变换以后只有左上角区域存在非零系数。当然这个示例是对整幅图像做了 DCT 变换,而在视频编解码中,图像是分割成块来处理的,因而变换也是基于块来进行的。

变换在通常的视频的压缩中非常有用,但是也并非所有的块都必须要做变换,有些情况下变换反而不如不变换压缩效果好,因而在某些标准如 VVC 中,编码器可以选择当前块是否使用变换。其中,DCT2 型(DCT-II)是视频压缩标准中最常用的变换,其变换的基图像如图 29 所示。

另外,VVC 中还可以使用 DCT8 型(DCT-VIII)和 DST7 型(DST-VII)。这些变换的基本公式如表 3 所示,这里示出了 N 个点输入的 DCT2, DCT8 和 DST7 的基本变换公式。

表 3

Transform Type	Basis function $T_i(j), i, j = 0, 1, \dots, N-1$
DCT-II	$T_i(j) = \omega_0 \cdot \sqrt{\frac{2}{N}} \cdot \cos\left(\frac{\pi \cdot i \cdot (2j + 1)}{2N}\right)$ where, $\omega_0 = \begin{cases} \sqrt{\frac{2}{N}} & i = 0 \\ 1 & i \neq 0 \end{cases}$
DCT-VIII	$T_i(j) = \sqrt{\frac{4}{2N + 1}} \cdot \cos\left(\frac{\pi \cdot (2i + 1) \cdot (2j + 1)}{4N + 2}\right)$
DST-VII	$T_i(j) = \sqrt{\frac{4}{2N + 1}} \cdot \sin\left(\frac{\pi \cdot (2i + 1) \cdot (j + 1)}{2N + 1}\right)$

由于图像都是二维的,而直接进行二维的变换运算量和内存开销都是当时硬件条件所不能接受的,因而在标准中使用的上述 DCT2、DCT8、DST7 变换都是拆分成水平方向和竖直方向的一维变换分成两步进行的。如先进行水平方向的变换再进行竖直方向的变换,或者先进行竖直方向的变换再进行水平方向的变换。

(1) 多变换选择 MTS。
VVC 支持 DCT2、DCT8、DST7 等变换核,对一个块,编码器可以选择合适的变换核并将索引传输入到码流中,解码器根据索引确定反变换的变换核。水平方向和竖直方向可以选择不同的变换核,如水平方向用 DCT8,竖直方向用 DST7。这个技术一般称为 MTS。

(2) 低频不可分离的变换 LFNST。
上述变换方法对水平方向和竖直方向的纹理比较有效,但是对斜向的纹理效果就会差一些。确实水平和竖直方向的纹理是最常见的,因而上述的变换方法对提升压缩效率是非常有用的。随着对压缩效率需求的不断提高,如果斜向的纹理能够更有效地处理,可以进一步提升压缩效率。

为了更有效地处理斜向纹理的残差，VVC 中使用了 LFNST 变换。将上述变换诸如 DCT2、DCT8、DST7 称为基础变换（Primary Transform）。在 VVC 的编码端，LFNST 用于 DCT2 变换之后量化之前。在 VVC 的解码端，LFNST 用于反量化之后，反 DCT2 变换之前。因为是在 DCT2（基础变换）的基础上再做变换，LFNST 是一种二次变换。图 30 为无 LFNST（二次变换）的编解码流程示意图，图 31 为有 LFNST（二次变换）的编解码流程示意图。当然编码端可以不通过熵解码而是直接对保存的量化系数进行反量化，因为熵编码是无损的。

图 32 为一种存在 LFNST（二次变换）的详细编解码流程示意图。如图 32 所示，在编码端，首先是正向基础变换，然后 LFNST 对基础变换后的左上角的低频系数进行二次变换。示例性地，对于 4×4 的 LFNST 存在 16 个输入系数，对于 8×8 的 LFNST 存在 64 个输入系数；然后对 LFNST 变换后的系数进行量化处理，并将量化系数写入码流。在解码端，通过解码码流和反量化可以得到变换系数，然后在反 LFNST 变换中，对于 4×4 的反 LFNST 存在 8 个输入系数，对于 8×8 的反 LFNST 存在 16 个输入系数，最后反基础变换可以得到残差块。

也就是说，在编码端，LFNST 对基础变换后的左上角的低频系数进行二次变换。基础变换通过对图像进行去相关性，把能量集中到左上角。而二次变换对基础变换的低频系数再去相关性，结果直观地来说如图 32 所示。在编码端，16 个系数输入到 4×4 的 LFNST，输出是 8 个系数；64 个系数输入到 8×8 的 LFNST，输出是 16 个系数。在解码端，8 个系数输入到 4×4 的反 LFNST，输出是 16 个系数；16 个系数输入到 8×8 的反 LFNST，输出是 64 个系数。

图 33 为 VVC 中 LFNST 的一些基图像。根据图 33 可以看出一些明显的斜向纹理。其中，LFNST 除了有针对某些斜向纹理优化的变换核，也有针对平坦渐变纹理优化的变换核，如 VVC 中的 LFNST 的变换核组 0。

LFNST 只应用于帧内编码的块。角度预测按照指定的角度将参考像素平铺到当前块作为预测值，这意味着预测块会有明显的方向纹理，而当前块经过角度预测后的残差在统计上也会体现出明显的角度特性。因而 LFNST 所选用的变换核可以跟帧内预测模式进行绑定，即确定了帧内预测模式以后，LFNST 只能使用帧内预测模式对应的一组（set）变换核。

具体地，VVC 中的 LFNST 总共有 4 组变换核，每组可选 2 个变换核。表 4 给出了帧内预测模式和变换核组的对应关系。注意到色度帧内预测使用的跨分量预测模式为 81 到 83，亮度帧内预测并没有这几种模式。LFNST 的变换核可以通过转置来用一个变换核组对应处理更多的角度，示例性地，13 到 23 和 45 到 55 的模式都对应变换核组 2，但是 13 到 23 明显是接近于水平的模式而 45 到 55 明显是接近于竖直的模式。

表 4

IntraPredMode	Tr. set index
IntraPredMode < 0	1
0 <= IntraPredMode <= 1	0
2 <= IntraPredMode <= 12	1
13 <= IntraPredMode <= 23	2
24 <= IntraPredMode <= 44	3
45 <= IntraPredMode <= 55	2
56 <= IntraPredMode <= 80	1
81 <= IntraPredMode <= 83	0

VVC 的 LFNST 共有 4 组变换核，根据帧内预测模式指定 LFNST 使用哪一组。这样做利用了帧内预测模式和 LFNST 的变换核之间的相关性，从而减少了选择 LFNST 的变换核在码流中的传输。而当前块是否会使用 LFNST，以及如果使用 LFNST，是使用一个组中的第一个还是第二个，是需要通过码流和一些条件来确定的。

在后续的 ECM 技术演进中，LFNST 进一步扩展。LFNST 有更多的变换核组，在 ECM 中是 35 组，变换核组索引（LFNST set index）和帧内预测模式（Intra pred.mode）的对应关系如表 5。每个变换核组对对应角度的纹理更高效。在这里，每个变换核组可以选择 3 个变换核。

表 5

Intra pred. mode	-14	-13	-12	-11	-10	-9	-8	-7	-6	-5	-4	-3	-2	-1	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
LFNST set index	2	2	2	2	2	2	2	2	2	2	2	2	2	2	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
Intra pred. mode	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49
LFNST set index	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	33	32	31	30	29	28	27	26	25	24	23	22	21	20	19
Intra pred. mode	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	
LFNST set index	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2

(3) 不可分离的基础变换 NSPT。

LFNST 是水平竖直不可分离的变换, 因为有二次变换, 所以 DCT2 可以被称为基础变换。这样先经过 DCT2, 再进行 LFNST, 可以说是一种性能和复杂度折衷的方案, 因为直接进行不可分离的基础变换效率更高, 但是有更高的复杂度, 比如说计算量和变换核的存储空间都更高。

在 ECM10 中, 一些小块可以使用 NSPT, 而大块仍然使用 DCT2+LFNST。小块的尺寸如 4×4 , 4×8 , 8×4 , 8×8 , 4×16 , 16×4 , 8×16 , 16×8 , 8×32 , 32×8 。在 ECM10 中, NSPT 同样根据帧内预测模式匹配变换核组, 匹配方法可以参照 LFNST 的方法, 每个变换核组有 3 个变换核可选。示例性地, ECM10 中的一个 NSPT 的 8×8 基图像如图 34 所示, 图 34 是对应帧间角度预测模式 7 的, 可见其处理对应角度的纹理更优。需要注意的是, NSPT 只应用于帧内编码的块。

进一步地, 对于划分技术来说, 这里可以分为单树 (Single tree) 划分和双树 (Dual tree) 划分。其中, Single tree 中亮度和色度一起划分, 一个 CU 包含亮度块和相同位置的色度块。Dual tree 中亮度和色度单独划分, 一个 CU 只有亮度块或只有色度块。HEVC 中只有 Single tree。VVC 中引入了 Dual tree, 但是 VVC 中的 P 和 B slice 还是只能用 Single tree。VVC 中的 I slice 可以用 Dual tree。如图 35 所示, 亮度和色度分别划分, 可以看出亮度比色度有更精细的划分, 因为亮度比色度具有更多的细节, 而且一般亮度比色度要求更好的质量。

综上可知, 在相关技术中, LFNST 和 NSPT 只应用于帧内编码的块。由于很多情况下帧内预测不能完美地预测复杂的纹理, 帧内编码的块通常有较多的残差, 而且在统计上帧内预测的残差跟预测角度存在一定的相关性, 而帧间编码的块通常能从参考图像中找到较好的预测块, 因而残差较少, 而且没有帧内预测模式去匹配对应的变换核组, 如果在码流中指示变换核组的选择, 开销会很大。这两方面可能是相关技术的 LFNST 和 NSPT 没有用于帧间编码的原因, 不利于帧间编解码的性能提升。

基于此, 本申请实施例提供了一种编码方法, 确定当前块的纹理特征索引; 根据纹理特征索引, 确定当前块的变换核组; 根据变换核组, 确定当前块的变换核; 根据变换核对当前块的残差块进行变换, 确定当前块的变换系数; 以及对当前块的变换系数进行编码处理, 将所得到的编码比特写入码流。本申请实施例还提供了一种解码方法, 确定当前块的纹理特征索引; 根据纹理特征索引, 确定当前块的变换核组; 根据变换核组, 确定当前块的变换核; 以及根据变换核对当前块的变换系数进行反变换, 确定当前块的残差块。

这样, 无论是编码端还是解码端, 都是首先确定当前块的纹理特征索引, 然后根据纹理特征索引来确定当前块的变换核组, 进而根据变换核组可以确定出当前块的变换核。也就是说, 这里建立了纹理特征索引与变换核组之间的对应关系, 替代了相关技术中根据帧内预测模式来匹配变换核组的方案, 使得 LFNST 和 NSPT 也可以应用于帧间预测模式, 从而针对帧间预测模式中的不好预测的块, 即残差较大的块, 不仅可以提高压缩效率, 即视频编解码效率, 同时还可以提升编解码性能。

下面将结合附图对本申请各实施例进行详细说明。

图 36 为本申请实施例提供的一种视频编解码的网络架构示意图。如图 36 所示, 该网络架构包括一个或多个电子设备 13 至 1N 和通信网络 01, 其中, 电子设备 13 至 1N 可以通过通信网络 01 进行视频交互。电子设备在实施的过程中可以为各种类型的具有视频编解码功能的设备, 例如, 所述电子设备可以包括手机、平板电脑、个人计算机、个人数字助理、导航仪、数字电话、视频电话、电视机、传感设备、服务器等, 本申请实施例不作限定。

在本申请实施例中, 这里提供了一种包含解码方法和编码方法的视频编解码系统的网络架构。其中, 本申请实施例中的解码器或编码器就可以为上述电子设备。也就是说, 本申请实施例中的电子设备具有视频编解码功能, 一般包括视频编码器 (即编码器) 和视频解码器 (即解码器)。

图 37 为本申请实施例提供的一种编码器的系统组成框图示意图。如图 37 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、反变换单元 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。反变换单元 111 执行对应于变换单元 108 中的变换的一个或多个反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 37 中相应单元类似的连接和数据交换功能。

图 38 为本申请实施例提供的一种解码器的系统组成框图示意图。如图 38 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、反变换单元 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。

反变换单元 206 具有与编码器 100 中的反变换单元 111 相同的功能。反变换单元 206 执行一个或多个变换操作 (即通过编码器 100 中的反变换单元 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 的预测块和来自反变换单元 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 执行帧间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 38 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”）。

在本申请的一实施例中，参见图 39，其示出了本申请实施例提供的一种解码方法的流程示意图。如图 39 所示，该方法可以包括：

S3901：确定当前块的纹理特征索引。

需要说明的是，在本申请实施例中，该方法应用于解码器。具体来说，基于图 38 所示解码器 200 的组成结构，本申请实施例的解码方法主要应用于帧间解码的块。其中，在帧间解码的块采用帧间预测模式时，这里主要是针对帧间预测模式下的 NSPT 和 LFNST 变换所提出的优化方案，以提高帧间预测的压缩效率。

在这里，NSPT 和 LFNST 都是可以高效处理各种角度的纹理的变换，这里可能有多个变换核，一个变换核可能是专门为某种特定的角度纹理优化的。除了角度的纹理，NSPT 和 LFNST 也包括处理渐变纹理的变换核。其实这些变换核也可以说是训练好的 KL 变换（Karhunen-Loeve Transform, KLT）。也就是说，NSPT 和 LFNST 都可以有多个变换核，每个变换核为特定的纹理设计。其中，特定的纹理包括角度纹理，渐变纹理等。另外，渐变纹理可以进一步扩展如水平渐变纹理，竖直渐变纹理，斜向渐变纹理等。进一步地，本技术方案不局限于只能用于 NSPT 和 LFNST 这样的不可分离变换，针对为特定纹理优化的可分离变换也可以适用本技术方案。

还需要说明的是，在帧内预测的块中，NSPT 和 LFNST 都包括有多个变换核组，每种帧内预测模式可以对应一种变换核组。但是帧间预测的块并没有帧内预测模式，这是将 LFNST 和 NSPT 应用于帧间预测的问题。在本申请实施例中，可以根据帧间预测的块推导出一个虚拟的帧内预测模式（索引），或者可称为纹理特征索引。例如，DC 模式和 PLANAR 模式对应渐变的纹理特征，某一种角度预测模式对应这种角度的纹理特征。也就是说，在本申请实施例中，纹理特征索引一方面可以避免在“帧间”出现帧内预测模式，另一方面也更有利于可能的扩展，例如一种帧内预测模式可以对应多种纹理特征，比如 DC 模式可以对应水平渐变纹理、竖直渐变纹理、斜向渐变纹理等。

在本申请实施例中，为了适配帧间预测的 NSPT 和 LFNST 变换的方案，这里提出了虚拟的帧内预测模式（或称为“纹理特征索引”）。所谓虚拟的帧内预测模式，是因为该模式不参与预测，只是为了选择 NSPT 或 LFNST 的变换核组。下面针对纹理特征索引的推导进行详细描述。

在一些实施例中，确定当前块的纹理特征索引，可以包括：确定用于推导纹理特征索引的候选像素；根据候选像素，确定当前块的纹理特征索引。

在一种可能的实现方式中，对于候选像素来说，可以是确定当前块的预测块；将预测块中的至少部分像素作为候选像素。

在本申请实施例中，如果预测块存在某种纹理，可以认为残差块中存在相同特征的纹理。这样，关于帧间推导纹理特征索引所使用的候选像素，可以是预测块中的全部像素或者预测块中的部分像素。

在另一种可能的实现方式中，对于候选像素来说，可以是确定当前块的已重建区域的相邻像素；将已重建区域的相邻像素作为候选像素。

在本申请实施例中，关于帧间推导纹理特征索引所使用的候选像素，可以是使用当前块的已重建区域的相邻像素，例如当前块左侧和右侧的已重建区域。因为左侧和上侧的已重建区域虽然不是当前块但与当前块相邻，比如说纹理有相连的情况，因此一定程度上可以用于估计当前块的纹理。

在又一种可能的实现方式中，考虑使用更多的像素，对于候选像素来说，可以是已将重建区域的相邻像素和预测块中的至少部分像素共同作为候选像素。

在本申请实施例中，关于帧间推导纹理特征索引所使用的候选像素，也可以是同时使用当前块的预

测块和当前块左侧和上侧的已重建区域的相邻像素,这样用于推导纹理特征索引的像素更多,使得所推导的纹理特征索引更准确。

在再一种可能的实现方式中,考虑硬件的流水线长度,对于候选像素来说,可以是确定当前块的参考块;将参考块中的至少部分像素作为候选像素。

5 在本申请实施例中,对当前块而言,预测块得到以后才能推导纹理特征索引,确定纹理特征索引之后才能确定 LFNST/NSPT 的变换核,然后才能够进行反变换及后续流程。在这里,如果使用参考块来推导纹理特征索引,理论上产生预测块的过程和推导纹理特征索引的过程可以并行进行,这样就比使用预测块的流水线要短。因而,还可以使用参考块来推导纹理特征索引,或者在产生预测块的过程中同步推导纹理特征索引,从而能够缩短流水线长度。

10 在一些实施例中,对于当前块的参考块来说,可以确定参考块为整像素的参考块;或者,确定参考块为分像素的参考块。

在一种可能的实现方式中,参考块时可以直接使用整像素的参考块,因为纹理方向不会有明显的区别。比如一个双向预测的帧间块,可以根据整像素的参考块推导虚拟的帧内预测模式。

15 在另一种可能的实现方式中,参考块也可以使用经过插值滤波(interpolation filter)的分像素的参考块。具体地,确定当前块进行双向预测时的两个参考图像块;对两个参考图像块进行分像素的插值滤波,确定两个分像素的参考图像块;对两个分像素的参考图像块进行加权组合,确定当前块的参考块。

示例性地,对于一个双向预测的帧间块,首先需要对两个参考块进行分像素的插值滤波,然后再将两个分像素的参考块进行加权组合。那么可以在得到两个插值滤波后的参考块后,一方面可以根据插值滤波后的参考块进行加权平均得到预测值,另一方面也可以根据插值滤波后的参考块推导纹理特征索引。

20 还需要说明的是,在本申请实施例中,用于推导纹理特征索引的候选像素的数量可以为至少一个,例如 1 个、2 个、3 个或更多个。在一些实施例中,可以根据当前块的尺寸参数,确定候选像素的数量。

也就是说,在确定当前块的纹理特征索引时,使用多少个候选像素可以由当前块的尺寸参数确定。示例性地,如果当前块的尺寸小,那么可以对所有可用的像素进行统计;如果当前块的尺寸大,那么可以对当前块降采样进行统计,比如水平方向和/或竖直方向每 2 个、或 4 个、或 8 个像素中的一个像素进行统计。或者,如果当前块的水平或竖直某一个方向的尺寸小于或等于 8,那么对该方向上的所有可用的像素进行统计;否则,如果当前块的水平或竖直某一个方向的尺寸小于或等于 16,那么对该方向上的每 2 个像素中的一个像素进行统计;否则,对该方向上的每 4 个像素中的一个像素进行统计,这里不作具体限定。

25 在一些实施例中,根据候选像素,确定当前块的纹理特征索引,可以包括:确定候选像素的水平梯度值和竖直梯度值;根据候选像素的水平梯度值和竖直梯度值,确定候选像素对应的纹理特征索引以及梯度强度值;根据候选像素对应的纹理特征索引以及梯度强度值,构建纹理特征统计表;根据纹理特征统计表,确定当前块的纹理特征索引。

需要说明的是,在本申请实施例中,在根据候选像素的水平梯度值和竖直梯度值,确定候选像素对应的纹理特征索引以及梯度强度值时,可以包括:根据候选像素的水平梯度值和竖直梯度值进行角度映射,确定候选像素对应的纹理特征索引;以及根据候选像素的水平梯度值和竖直梯度值进行梯度强度计算,确定候选像素对应的梯度强度值。

30 在一种具体的实施例中,根据候选像素的水平梯度值和竖直梯度值进行角度映射,确定候选像素对应的纹理特征索引,可以包括:根据候选像素的水平梯度值和竖直梯度值,利用预设查找表确定候选像素对应的纹理特征索引。

40 在本申请实施例中,候选像素的水平梯度值可以用 $grad_x$ 表示,候选像素的竖直梯度值可以用 $grad_y$ 表示。这样,根据 $grad_x$ 和 $grad_y$ 导出虚拟的帧内预测模式可以通过查表来实现。

示例性地,如果 $abs(grad_x)$ 等于 0 且 $abs(grad_y)$ 不等于 0,那么有水平方向纹理,对应一些技术中的帧内预测模式 18。如果 $abs(grad_y)$ 等于 0 且 $abs(grad_x)$ 不等于 0,那么有竖直方向纹理,对应一些技术中的帧内预测模式 50。在 $abs(grad_x)$ 和 $abs(grad_y)$ 都不等于 0 的情况下,如果 $abs(grad_x)$ 等于 $abs(grad_y)$,且 $grad_x$ 和 $grad_y$ 符号相同,对应一些技术中的帧内预测模式 34。如果 $abs(grad_x)$ 等于 2 倍的 $abs(grad_y)$,且 $grad_x$ 和 $grad_y$ 符号相同,对应一些技术中的帧内预测模式 40。另外,其他情况可以按照相同的原則查表确定。

45 在一种具体的实施例中,根据候选像素的水平梯度值和竖直梯度值进行梯度强度计算,确定候选像素对应的梯度强度值,可以包括:对水平梯度值的绝对值和竖直梯度值的绝对值进行加法运算,确定候选像素对应的梯度强度值。

在这里,候选像素对应的梯度强度值可以记为 amp 。示例性地, $amp = abs(grad_x) + abs(grad_y)$ 。

需要说明的是,在本申请实施例中,针对候选像素的水平梯度值和竖直梯度值可以使用索贝尔算子计算得到。示例性地,对于索贝尔算子来说,具体如下所示:

水平梯度值的算子:

$$\begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$$

竖直梯度值的算子:

$$\begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$$

- 5 如此,假设像素位置为 (x, y) 的像素值为 $P_{x,y}$,则水平梯度值 $grad_x$ 和竖直梯度值 $grad_y$ 的计算如下所示:

$$grad_x = P_{x+1,y-1} + 2 * P_{x+1,y} + P_{x+1,y+1} - P_{x-1,y-1} - 2 * P_{x-1,y} - P_{x-1,y+1} \quad (22)$$

$$grad_y = P_{x-1,y+1} + 2 * P_{x,y+1} + P_{x+1,y+1} - P_{x-1,y-1} - 2 * P_{x,y-1} - P_{x+1,y-1} \quad (23)$$

- 10 在一些实施例中,根据候选像素对应的纹理特征索引以及梯度强度值,构建纹理特征统计表,可以包括:在候选像素的数量为至少一个时,确定至少一个纹理特征索引以及对应的至少一个梯度强度值;根据至少一个纹理特征索引确定具有互异特性的至少一种参考纹理特征索引,以及根据至少一个梯度强度值,对归属于同一参考纹理特征索引的梯度强度值进行累加计算,确定至少一种参考纹理特征索引对应的梯度强度累加值;根据至少一种参考纹理特征索引和至少一种参考纹理特征索引对应的梯度强度累加值,构建纹理特征统计表。

- 15 也就是说,在本申请实施例中,以预测块中的至少部分像素作为候选像素为例,对预测块中的全部或部分像素计算其梯度值,一般来说可以计算水平梯度值和竖直梯度值,这里可以使用索贝尔算子来计算梯度值。对某一个像素,根据其水平梯度值和竖直梯度值可以推测该像素的纹理方向,比如水平梯度值非零,竖直梯度值为零,则该像素的纹理是竖直方向。反之,水平梯度值为零,竖直梯度值非零,则该像素的纹理是水平方向。比如水平梯度值和竖直梯度值相等且不为零,则该像素的纹理是45度。当然,本申请实施例还有很多其他情况是水平梯度值和竖直梯度值都不为零,可以根据它们的比值确定该像素的纹理的方向。这样可以根据每一个像素的梯度强度值可以对应到对应的纹理特征索引。从而构建一个纹理特征统计表,将每一个计算的像素的梯度强度值累加到统计表中对应的纹理特征索引项中,以得到最终的纹理特征统计表,进而根据纹理特征统计表可以确定当前块的纹理特征索引。

- 20 在一种可能的实现方式中,在根据纹理特征统计表确定当前块的纹理特征索引时,可以包括:确定纹理特征统计表中的最大梯度强度累加值;将最大梯度强度累加值对应的参考纹理特征索引确定为当前块的纹理特征索引。

- 25 在另一种可能的实现方式中,在根据纹理特征统计表确定当前块的纹理特征索引时,可以包括:确定纹理特征统计表中的最大梯度强度累加值;在最大梯度强度累加值小于第一阈值时,将当前块的纹理特征索引设置为DC模式或PLANANR模式。

- 30 在又一种可能的实现方式中,在根据纹理特征统计表确定当前块的纹理特征索引时,可以包括:确定纹理特征统计表中所有梯度强度累加值的统计和值;在统计和值小于第二阈值时,将当前块的纹理特征索引设置为DC模式或PLANANR模式。

- 也就是说,在本申请实施例中,对于虚拟的帧内预测模式的导出,由于虚拟的帧内预测模式(索引)也可以称为纹理特征索引,因而这里也可以称作纹理特征索引的导出。

- 35 在本申请实施例中,可以利用帧间预测块来导出虚拟的帧内预测模式。如果预测块存在某种纹理,可以想象残差块中存在相同特征的纹理。示例性地,这里可以对预测块中的全部或部分像素计算其梯度值,一般来说可以计算水平梯度值和竖直梯度值,这里可以使用索贝尔算子来计算梯度值。然后根据水平梯度值和竖直梯度值来确定对应的梯度强度值,再根据梯度强度值对应到对应的帧内预测模式,从而构建一个帧内预测模式统计表(即前述的“纹理特征统计表”),需要将每一个计算的候选像素的梯度强度值累加到统计表中对应的帧内预测模式项中。在梯度统计完成后,一种可能的实现方式为帧内预测模式统计表中累计最多的一个帧内预测模式即为当前块的虚拟的帧内预测模式。或者,另一种可能的实现方式为当统计的累计梯度强度最高的帧内预测模式的梯度强度累加值小于某一个阈值时,确定虚拟的帧内预测模式为DC模式或PLANANR模式。或者,又一种可能的实现方式为当所有统计的像素的梯度强度累加值之和小于某一个阈值时,确定虚拟的帧内预测模式为DC模式或PLANANR模式。

- 40 另外,针对DIMD也需要进行类似的统计导出帧内预测模式,但是区别在于DIMD使用的是当前块的周边重建区域,这里使用的是当前块的预测块。在实现上,本方案可以和DIMD复用部分逻辑。

还需要说明的是,关于在帧间导出虚拟帧内预测模式所使用的像素,这里也可以是使用当前块左侧

和上侧的已重建区域，和 DIMD 的做法类似。因为左侧和上侧的已重建区域虽然不是当前块但是与当前块相邻，比如说纹理有相连的情况，因而一定程度上可以用来估计当前块的纹理。另一种可能是同时使用当前块的预测块和当前块左侧和上侧的已重建区域，这样可用来推理的虚拟的帧内预测模式的像素更多。又一种可能是考虑硬件的流水线长度。对当前块而言，预测块得到以后才能导出虚拟的帧内预测模式，确定了虚拟的帧内预测模式才能确定 LFNST/NSPT 的变换核，才能进行反变换及后续流程。如果使用参考块来导出虚拟的帧内预测模式，理论上产生预测块的过程和推导虚拟的帧内预测模式的过程可以并行进行，这样就比上述的设计流水线要短。因而，本申请实施例还可以使用参考块来导出虚拟的帧内预测模式，或者在产生预测块的过程中同步推导虚拟的帧内预测模式。其中，使用参考块时可以直接使用整像素的参考块，因为纹理方向不会有明显的区别。比如一个双向预测的帧间块，一方面根据整像素的参考块推导虚拟的帧内预测模式，另一方面对 2 个参考块进行分像素的插值滤波，然后再将 2 个分像素的参考块进行加权组合。具体地，这里也可以使用经过插值滤波的分像素的参考块，比如一个双向预测的帧间块，需要首先对 2 个参考块进行分像素的插值滤波，然后再将 2 个分像素的参考块进行加权组合。那么可以在得到 2 个插值滤波后的参考块后，一方面根据插值滤波后的参考块进行加权平均得到预测值，另一方面根据插值滤波后的参考块推导虚拟的帧内预测模式。

在一些实施例中，确定当前块的纹理特征索引，还可以包括：确定当前块进行几何划分模式时的角度索引；根据角度索引，确定当前块的纹理特征索引。

在本申请实施例中，对于几何划分模式 GPM 的虚拟的帧内预测模式的导出，GPM 的预测块是把个不同运动的预测块组合到一起得到的，因而它可能包含 2 个物体的内容，通常情况下边界很难被完美地预测，因而 2 个物体的边界处可能存在较大的残差，而这个残差的方向和 GPM 的“划分”方向是一致的。因此对于 GPM 来说，可以使用 GPM 的“划分”模式来确定虚拟的帧内预测模式。

在本申请实施例中，所述划分线可以认为是 GPM 的权重矩阵中权重发生变化的区域中权重相同的点所组成的线。或者说权重为中值的点组成的线，权重为中值的点可能不在整像素位置。以权重为 0 到 8 为例，中值可以是 4。图 40 示出了一种 GPM 模式的划分线示意图。如图 40 所示，加粗实线为这里所述的划分线。

其实 GPM 的权重是根据划分线导出的。下面以一些技术中的 GPM 的一些过程举例，下面的 merge_gpm_partition_idx[xCb][yCb]用来确定“划分”，它就是本申请实施例所说的权重导出模式。根据 merge_gpm_partition_idx[xCb][yCb]利用表 6 确定 GPM 的“划分”角度索引变量 angleIdx 和距离索引变量 distanceIdx。其中，角度索引变量 angleIdx 和距离索引变量 distanceIdx 就可以认为是确定划分线的变量，它们分别确定划分线的角度和偏移量。

表 6 提供了一种 angleIdx 和 distanceIdx 与 on merge_gpm_partition_idx 的对应关系示例，具体如下。

表 6

merge_gpm_partition_idx	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
angleIdx	0	0	2	2	2	2	3	3	3	3	4	4	4	4	5	5
distanceIdx	1	3	0	1	2	3	0	1	2	3	0	1	2	3	0	1
merge_gpm_partition_idx	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
angleIdx	5	5	8	8	11	11	11	11	12	12	12	12	13	13	13	13
distanceIdx	2	3	1	3	0	1	2	3	0	1	2	3	0	1	2	3
merge_gpm_partition_idx	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47
angleIdx	14	14	14	14	16	16	18	18	18	19	19	19	20	20	20	21
distanceIdx	0	1	2	3	1	3	1	2	3	1	2	3	1	2	3	1
merge_gpm_partition_idx	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63
angleIdx	21	21	24	24	27	27	27	28	28	28	29	29	29	30	30	30
distanceIdx	2	3	1	3	1	2	3	1	2	3	1	2	3	1	2	3

因为三个颜色分量（component，如 Y、Cb、Cr）都可以使用 GPM，所以标准文本将一个分量产生 GPM 的预测样本矩阵的过程分装到了一个子流程里面，即 GPM 的加权预测过程（Weighted sample prediction process for geometric partitioning mode），三个颜色分量都会调用这个流程，只是调用的参数不同，这里只用亮度分量举例。当前亮度块的预测矩阵 predSamplesL[xL][yL]（其中 xL=0..cbWidth-1，yL=0..cbHeight-1）由 GPM 的加权预测过程导出。其中的 nCbW 设为 cbWidth，nCbH 设为 cbHeight，两个预测模式做的预测样本矩阵 predSamplesLAL 和 predSamplesLBL，还有 angleIdx，distanceIdx 作为输入。

下面是 GPM 的加权预测过程导出的过程：

这个过程的输入有：

- 当前块的宽度 nCbW 当前块的高度 nCbH;
 - 2个(nCbW)x(nCbH) 的预测样本矩阵 predSamplesLA 和 predSamplesLB;
 - GPM 的“划分”角度索引变量 angleIdx;
 - GPM 的距离索引变量 distanceIdx;
- 5 分量索引变量 cIdx。因为本实施例中只以亮度分量举例，其中 cIdx 为 0，表示亮度分量。
 这个过程的输出是 (nCbW) × (nCbH) 的 GPM 预测样本矩阵 pbSamples。
 变量 nW、nH、shift1、offset1、displacementX、displacementY、partFlip，还有 shiftHor 按如下方法导出：
- 10 $nW = (cIdx == 0) ? nCbW : nCbW * SubWidthC;$
 $nH = (cIdx == 0) ? nCbH : nCbH * SubHeightC;$
 $shift1 = \text{Max}(5, 17 - \text{BitDepth})$ ，其中 BitDepth 是编解码的比特深度;
 $offset1 = 1 \ll (shift1 - 1);$
 $displacementX = angleIdx;$
 $displacementY = (angleIdx + 8) \% 32;$
 15 $partFlip = (angleIdx \geq 13 \ \&\& \ angleIdx \leq 27) ? 0 : 1;$
 $shiftHor = (angleIdx \% 16 == 8 \ || \ (angleIdx \% 16 != 0 \ \&\& \ nH \geq nW)) ? 0 : 1;$
 变量 offsetX and offsetY 按如下方法导出：
 – 如果 shiftHor 的值为 0:
 $offsetX = (-nW) \gg 1;$
 20 $offsetY = ((-nH) \gg 1) + (angleIdx < 16 ? (distanceIdx * nH) \gg 3 : -((distanceIdx * nH) \gg 3));$
 – 否则(即 shiftHor 的值为 1):
 $offsetX = ((-nW) \gg 1) + (angleIdx < 16 ? (distanceIdx * nW) \gg 3 : -((distanceIdx * nW) \gg 3));$
 $offsetY = (-nH) \gg 1;$
 预测样本矩阵 pbSamples[x][y] (其中 $x = 0..nCbW - 1$, $y = 0..nCbH - 1$) 按如下方法导出：
 25 – 变量 xL 和 yL 按如下方法导出：
 $xL = (cIdx == 0) ? x : x * SubWidthC;$
 $yL = (cIdx == 0) ? y : y * SubHeightC;$
 – 表示当前位置预测样本权重的变量 wValue 按如下方法导出，wValue 即为 (x, y) 点的第一
 预测模式的预测矩阵的预测值 predSamplesLA[x][y] 的权重，而 (8 - wValue) 即为 (x, y) 点的第一预
 30 测模式的预测矩阵的预测值 predSamplesLB[x][y] 的权重。
 其中，表 7 提供了一种距离矩阵 disLut 的定义示例。如此，disLut 的取值可以按表 7 确定。

表 7

idx	0	2	3	4	5	6	8	10	11	12	13	14
disLut[idx]	8	8	8	4	4	2	0	-2	-4	-4	-8	-8
idx	16	18	19	20	21	22	24	26	27	28	29	30
disLut[idx]	-8	-8	-8	-4	-4	-2	0	2	4	4	8	8

- 在这里， $weightIdx = (((xL + offsetX) \ll 1) + 1) * disLut[displacementX] +$
 $((yL + offsetY) \ll 1) + 1) * disLut[displacementY];$
- 35 $weightIdxL = partFlip ? 32 + weightIdx : 32 - weightIdx;$
 $wValue = \text{Clip3}(0, 8, (weightIdxL + 4) \gg 3);$
 – 预测样本的值按如下方法导出：
 $pbSamples[x][y] = \text{Clip3}(0, (1 \ll \text{BitDepth}) - 1, (predSamplesLA[x][y] * wValue +$
 $predSamplesLB[x][y] * (8 - wValue) + offset1) \gg shift1);$
- 40 这里需要注意的是，标准文本中对每一个位置来推导一个权重值 wValue，然后计算一个 GPM 的预
 测值 pbSamples[x][y]。因为这种方式权重 wValue 不必写成一个矩阵的形式，但是可以理解如果把每个
 位置的 wValue 都保存到一个矩阵里，那它就是一个权重矩阵。每个点分别计算权重并加权得到 GPM
 的预测值，或者计算出所有的权重再统一加权得到 GPM 的预测样本矩阵其原理是一样的。而本文中的
 诸多描述中使用权重矩阵的说法，是为了表述更容易理解，用权重矩阵画图更直观，其实也可以按每个
 45 位置的权重来描述。比如权重矩阵导出模式也可以说成权重导出模式。
- 这样，根据权重 wValue 的导出过程可以看出 wValue 是根据 angleIdx 和 distanceIdx 导出的。所以在
 确定划分线对应的帧内预测模式时，也可以先确定 angleIdx，然后根据 angleIdx 确定其对应的帧内角

度预测模式。因为根据上面的表 6 也可以看出多个权重导出模式对应一个 angleIdx。一个可能的 angleIdx 和帧内角度预测模式的对应关系如表 8。可以看到有些 angleIdx 对应的是 0，那是因为这些 angleIdx 在 GPM 中没有被用到，也就是如图 10 中的 64 种权重的示例。需要理解的是，如果在某一个版本中某些 angleIdx 发生了变化，比如以后更多的 angleIdx 被使用到，或者帧内角度预测模式发生了变化，比如以后有更多的帧内预测模式，那么这个对应关系表也可以相应变化，这里并不作具体限定。

表 8

angleIdx	0	1	2	3	4	5	6	7
intra pred. mode	50	0	44	41	34	27	0	0
angleIdx	8	9	10	11	12	13	14	15
intra pred. mode	18	0	0	9	66	59	56	0
angleIdx	16	17	18	19	20	21	22	23
intra pred. mode	50	0	44	41	34	27	0	0
angleIdx	24	25	26	27	28	29	30	31
intra pred. mode	18	0	0	9	66	59	56	0

进一步地，在本申请实施例中，当前块的纹理特征索引也可以直接复制参考块的纹理特征索引。在一些实施例中，确定当前块的纹理特征索引，还可以包括：根据参考块的帧内预测模式或参考块的虚拟的帧内预测模式确定当前块的虚拟的帧内预测模式，或者说根据参考块的纹理特征索引确定当前块的纹理特征索引。如果当前块的参考块有帧内解码的块，那么帧内解码的块有帧内预测模式，可以将参考块的帧内预测模式作为当前块的虚拟的帧内预测模式。如果当前块的参考块包含多个帧内解码的块，可以根据坐标来确定某个位置的帧内预测模式，例如以当前块的中心位置对应的参考块的帧内预测模式作为当前块的虚拟的帧内预测模式。可以对每一个帧内解码的块都保存虚拟的帧内预测模式，如果当前块的参考块是帧内解码的块，那么它也可以找到虚拟的帧内预测模式。虚拟的帧内预测模式也可以按一定的粒度来存储，例如每个 4×4 的像素为一个最小存储单元，每个最小存储单元共享同一个帧内预测模式或虚拟的帧内预测模式。如果当前块的参考块包含多个 4×4 的最小存储单元，可以根据坐标来确定某个位置的（虚拟的）帧内预测模式，例如以当前块的中心位置对应的参考块的最小存储单元的（虚拟的）帧内预测模式作为当前块的虚拟的帧内预测模式。

如此，在确定出虚拟的帧内角度预测模式（即纹理特征索引）之后，就可以根据纹理特征索引来确定 NSPT/LFNST 的变换核组。

进一步地，在本申请实施例中，还可以构建候选列表，以此确定虚拟的帧内角度预测模式。因此，在一些实施例中，确定当前块的纹理特征索引，还可以包括：构建候选列表，其中，候选列表包括预设数量的候选纹理特征索引；解码码流，确定当前块的特征索引序号；根据候选列表和特征索引序号，确定当前块的纹理特征索引。其中，特征索引序号表征当前块的纹理特征索引在候选列表中的编号。

需要说明的是，前述实施例提到使用当前块的预测块或参考块或当前块周边的重建图像推导出虚拟的帧内预测模式。在一种具体的实施例中，可以选择梯度强度值最大的一个帧内预测模式作为虚拟的帧内预测模式。除此之外，也可以增加一个步骤，构建一个候选列表，即虚拟的帧内预测模式的候选列表，或者特征索引的候选列表。然后使用一个语法元素写在码流中指示最终选择了哪一个候选。这里所使用的语法元素即当前块的特征索引序号，可以用 `lfnst_nspt_feature_idx` 表示。

还需要说明的是，`lfnst_nspt_feature_idx` 的取值可以为 0、1、2、3 等整数。示例性地，如果 `lfnst_nspt_feature_idx` 的取值为 0，那么可以选择候选列表的第一个元素；如果 `lfnst_nspt_feature_idx` 为 1，那么可以选择候选列表的第二个元素。

在一些实施例中，对于构建候选列表，可以包括：将纹理特征统计表中的至少一种参考纹理特征索引按照对应的梯度强度累加值由大到小进行排序，确定排序靠前的预设数量个参考纹理特征索引；根据排序靠前的预设数量个参考纹理特征索引，构建候选列表。

在一些实施例中，对于构建候选列表，可以包括：确定纹理特征统计表中最大梯度强度累加值对应的第一纹理特征索引，以及确定当前块进行几何划分模式时对应的第二纹理特征索引；根据第一纹理特征索引和第二纹理特征索引，构建候选列表。

也就是说，在本申请实施例中，构建候选列表的一个示例是，仍按照上述计算梯度强度的方法，按梯度强度从大到小选择最靠前的 N 个帧内预测模式（纹理特征索引）构成候选列表。其中，N 的取值可能为 2、3、4 等。另一个示例是假设 N 的取值为 2，也就是梯度强度最大的帧内预测模式（纹理特征索引）为候选列表的第一个元素，梯度强度第二大的帧内预测模式（纹理特征索引）为候选列表的第二

个元素。又一个示例是对于 GPM, 可以设置候选列表第一个元素为 GPM 的“划分”模式对应的帧内预测模式(纹理特征索引), 第二个元素为梯度强度最大的帧内预测模式(纹理特征索引), 也可以反过来; 这里不作具体限定。

5 在一些实施例中, 对于构建候选列表, 还可以包括: 确定当前块的参考块; 根据参考块中至少一个候选位置对应的纹理特征索引, 构建候选列表。

也就是说, 在本申请实施例中, 根据参考块的帧内预测模式(纹理特征索引)构建候选列表, 例如根据参考块中某几个位置对应的 4×4 的最小存储单元的帧内预测模式(纹理特征索引)构建候选列表, 记当前块左上角的坐标为 (x, y) , 当前块的宽度 $nCbW$ 当前块的高度 $nCbH$, 当前块的中心点 $posC$ 为 $(x+nCbW/2, y+nCbH/2)$, 左上角 $posTL(x, y)$, 右上角 $posTR(x+nCbW-1, y)$, 左下角 $posBL(x, y+nCbH-1)$, 右下角 $posBR(x+nCbW-1, y+nCbH-1)$ 。依次用 $posC, posTL, posTR, posBL, posBR$ 对应的最小存储单元的帧内预测模式(纹理特征索引)构建候选列表。在这里, 候选位置可以为 $posC, posTL, posTR, posBL, posBR$ 中的至少一项, 但是并不局限于此。

10 还需要说明的是, 解码端在构建列表的时候需要保证候选列表中的每一项不重复, 如果可用的帧内预测模式(纹理特征索引)无法填满列表, 可以加入默认的帧内预测模式(纹理特征索引), 例如水平方向, 竖直方向的帧内预测模式(纹理特征索引)。

15 还需要说明的是, 这种方法不仅可以适用于帧间预测, 还可以适用于帧内预测。其中, 如果 N 的取值为 2, 那么只需要一个二元符号来表示; 如果 N 的取值大于 2, 那么可以使用截断一元码。在这里, 所有的二元符号都可以使用 CABAC 的上下文模型来解码得到。

20 这样, 在构建候选列表之后, 根据解码码流所得到的特征索引序号, 从而也能够确定出虚拟的帧内角度预测模式(即当前块的纹理特征索引)。

S3902: 根据纹理特征索引, 确定当前块的变换核组。

S3903: 根据变换核组, 确定当前块的变换核。

需要说明的是, 在确定出当前块的纹理特征索引之后, 根据纹理特征索引与变换核组之间的对应关系, 就可以确定当前块的变换核组。示例性地, 一些技术中的 LFNST 共有 4 组变换核组, 帧内预测模式与变换核组之间的对应关系详见表 4; 在一些技术中的 LFNST 也可以有更多的变换核组, 例如 35 组, 帧内预测模式与变换核组之间的对应关系详见表 5。

25 另外, 每一个变换核组可以包括至少两个变换核。示例性地, 变换核组可以包括 2 个变换核, 或者也可以包括 3 个变换核, 甚至也可以包括 4 个可选的变换核, 或者更多数量可选的变换核, 这里均不作具体限定。

30 在一些实施例中, 参见图 41, 在步骤 S3902 之后, 该方法还可以包括:

S4101: 解码码流, 确定当前块的第一语法标识信息的取值。

S4102: 在第一语法标识信息指示当前块使用第一变换模式时, 确定当前块的变换核索引, 并根据变换核组和变换核索引, 确定当前块的变换核。

需要说明的是, 在本申请实施例中, 第一语法标识信息可以用 $lfnst_nspt_idx$ 表示。其中, 第一语法标识信息可以用于指示当前块是否使用第一变换模式, 而且在当前块使用第一变换模式时, 第一语法标识信息的取值还可以用于指示当前块使用的变换核索引。

35 还需要说明的是, 在本申请实施例中, 第一变换模式可以为 LFNST 变换或者 NSPT 变换。也就是说, $lfnst_nspt_idx$ 用于指示当前块是否使用 LFNST 变换或者 NSPT 变换, 而且在当前块使用 LFNST 变换或者 NSPT 变换时, $lfnst_nspt_idx$ 的取值还可以用于指示当前块使用的变换核索引。其中, 可以根据当前块的尺寸参数来确定当前块是使用 LFNST 变换还是使用 NSPT 变换。示例性地, 如果当前块的尺寸参数较小, 那么可以确定 $lfnst_nspt_idx$ 用于指示当前块是否使用 NSPT 变换以及对应使用的 NSPT 的变换核索引; 如果当前块的尺寸参数较大, 那么可以确定 $lfnst_nspt_idx$ 用于指示当前块是否使用 LFNST 变换以及对应使用的 LFNST 的变换核索引。

40 还需要说明的是, 在本申请实施例中, 对于第一语法标识信息 $lfnst_nspt_idx$, 也可以使用 $lfnst_idx$ 和 $nspt_idx$ 两个语法标识信息来替代。其中, $lfnst_idx$ 用于指示当前块是否使用 LFNST 变换以及对应使用的 LFNST 的变换核索引, $nspt_idx$ 用于指示当前块是否使用 NSPT 变换以及对应使用的 NSPT 的变换核索引, 这里不作具体限定。

50 在一些实施例中, 在第一语法标识信息指示当前块使用第一变换模式时, 确定当前块的变换核索引, 可以包括: 若当前块使用帧内预测模式, 则解码码流, 确定当前块的变换核索引; 若当前块使用帧间预测模式, 则根据第一语法标识信息的取值确定当前块的变换核索引。

需要说明的是, 在本申请实施例中, 在当前块使用帧内预测模式时, 第一语法标识信息用于指示当前块是否使用第一变换模式; 在第一语法标识信息指示当前块使用第一变换模式时, 还需要解码码流来

确定当前块的变换核索引,然后根据变换核组和变换核索引来确定当前块的变换核;其中,变换核索引用于指示当前块的变换核在变换核组的编号。示例性地,如果当前块的变换核索引为0,那么当前块的变换核为变换核组的第一个变换核;如果当前块的变换核索引为1,那么当前块的变换核为变换核组的第二个变换核;如果当前块的变换核索引为2,那么当前块的变换核为变换核组的第三个变换核。

5 还需要说明的是,在本申请实施例中,在当前块使用帧内预测模式时,LFNST/NSPT一个变换核组有3个变换核可选,再加上不使用LFNST/NSPT的情况,也就是当前块针对LFNST/NSPT有4种可能的选择。因此,这里可以使用`lfnst_nspt_idx`指示当前块是否使用第一变换模式以及当前块使用的变换核索引。这样,根据第一语法标识信息的取值就可以确定当前块的变换核索引,然后根据变换核组和变换核索引来确定当前块的变换核。示例性地,如果第一语法标识信息的取值为0,那么确定当前块不使用LFNST/NSPT;如果第一语法标识信息的取值为1,那么确定当前块使用LFNST/NSPT,而且当前块的变换核为变换核组的第一个变换核;如果第一语法标识信息的取值为2,那么确定当前块使用LFNST/NSPT,而且当前块的变换核为变换核组的第二个变换核;如果第一语法标识信息的取值为3,那么确定当前块使用LFNST/NSPT,而且当前块的变换核为变换核组的第三个变换核。也就是说,在本申请实施例中,`lfnst_nspt_idx`的取值可以为0、1、2、3等。

15 还需要说明的是,在本申请实施例中,对于第一语法标识信息`lfnst_nspt_idx`,也可以使用另外两个语法标识信息来替代。其中,一个语法标识信息用于指示当前块是否使用LFNST/NSPT,另一个语法标识信息用于指示在当前块使用LFNST/NSPT时对应的变换核索引。这样,在码流中需要传输两个语法元素,导致码流开销大。在这种情况下,针对码流中需要传输变换核索引的情况,如果变换核组中只有一个变换核,那么这时候也不需要传输变换核索引。具体地,在当前块使用LFNST/NSPT时,可以将变换核组中的唯一一个变换核直接作为当前块的变换核。

20 进一步地,针对纹理特征索引通过候选列表来确定的情况,在一些实施例中,该方法还可以包括:解码码流,确定当前块的第一语法标识信息的取值;在第一语法标识信息指示当前块使用第一变换模式时,执行解码码流,确定当前块的特征索引序号的步骤。

也就是说,在本申请实施例中,第一语法标识信息用`lfnst_nspt_idx`表示,特征索引序号用`lfnst_nspt_feature_idx`表示。在这里,`lfnst_nspt_feature_idx`可以用来选择虚拟的帧内预测模式或纹理特征索引。示例性地,如果`lfnst_nspt_feature_idx`为0,那么选择候选列表的第一个元素;如果`lfnst_nspt_feature_idx`为1,那么选择候选列表的第二个元素。另外,这里也可以在当前块使用LFNST或NSPT的情况下再解析`lfnst_nspt_feature_idx`。具体地,如果当前块不使用LFNST/NSPT,那么不解析`lfnst_nspt_feature_idx`。示例性地,由于`lfnst_nspt_idx`用于指示当前块是否使用LFNST/NSPT以及LFNST/NSPT的变换核组中选择哪一个变换核,这时候如果`lfnst_nspt_idx`为0,表示当前块不使用LFNST/NSPT,这种情况下不需要解析`lfnst_nspt_feature_idx`;否则,如果`lfnst_nspt_idx`不为0,表示当前块使用LFNST/NSPT,这种情况下可以解析`lfnst_nspt_feature_idx`。需要注意的是,这种方法不仅可以适用于帧内预测,还可以适用于帧内预测,这里不作具体限定。

30 S3904:确定当前块的变换系数,并根据变换核对当前块的变换系数进行反变换,确定当前块的残差块。

35 需要说明的是,在本申请实施例中,对于确定当前块的变换系数,可以包括:解码码流,确定当前块的量化系数;对当前块的量化系数进行反量化,确定当前块的变换系数。

40 还需要说明的是,在本申请实施例中,在根据变换核对当前块的变换系数进行反变换,确定当前块的残差块时,可以包括:若当前块的尺寸参数满足第一条条件,则根据变换核对当前块的变换系数进行不可分离基础变换的反变换,确定当前块的残差块;若当前块的尺寸参数满足第二条条件,则根据变换核对当前块的变换系数进行低频不可分离变换的反变换,确定当前块的变换块;以及对当前块的变换块进行离散余弦变换的反变换,确定当前块的残差块。

45 在这里,当前块的尺寸参数满足第一条条件,包括:当前块的尺寸参数较小,例如当前块的尺寸参数小于某一阈值。也就是说,针对较小尺寸的块,这里使用NSPT的变换核,即根据变换核对当前块的变换系数进行反NSPT变换,确定当前块的残差块。

在这里,当前块的尺寸参数满足第二条条件,包括:当前块的尺寸参数较大,例如当前块的尺寸参数大于某一阈值。也就是说,针对较大尺寸的块,这里使用LFNST的变换核,即根据变换核对当前块的变换系数进行反LFNST变换,确定当前块的变换块;以及对当前块的变换块进行反DCT2变换,确定当前块的残差块。

50 还需要说明的是,在本申请实施例中,解码端对变换系数的“反变换”,在标准文本中也可称为“变换”。本文中的“变换”和“反变换”对应的是两个相反的过程,如“变换”将空间域的数值转换到频率域的系数,那么“反变换”将频率域的系数转换到空间域的数值。“反”是相对于“正”而言,它们

实质上都是变换。需要注意的是，标准如果只规定解码，那么标准文本中的“变换”是解码的部分，具体指的是本文中的“反变换”。

在一些实施例中，参见图 42，在步骤 S3904 之后，该方法还可以包括：

S4201：对当前块进行帧间预测，确定当前块的预测块。

5 S4202：根据当前块的预测块和当前块的残差块，确定当前块的重建块。

需要说明的是，在本申请实施例中，对于步骤 S4201 来说，该步骤可以是与步骤 S3901~S3903 并行操作，也可以是在步骤 S3901~3903 之前执行，这里对步骤的先后顺序并不作具体限定。

还需要说明的是，在本申请实施例中，在确定当前块的预测块之后，可以对当前块的预测块和当前块的残差块进行加法运算，从而确定出当前块的重建块。

10 简单来说，解码器在确定出预测块后，根据预测块导出虚拟的帧内预测模式，然后根据虚拟的帧内预测模式确定 NSPT/LFNST 的变换核组。如果一个变换核组有多个变换核可选，解码器通过在码流中解码 `lfnst_nspt_idx` 来确定变换核。在码流中解码 `lfnst_nspt_idx` 的过程和确定变换核组的过程没有依赖关系。通过熵解码从码流中解码得到量化系数。

如果是 NSPT 变换，对量化系数进行反量化得到解码的变换系数，对解码的变换系数进行反 NSPT 变换得到解码的残差块，最后根据解码的残差块和预测块得到重建块。

15 如果是 LFNST 变换，对量化系数进行反量化得到解码的变换系数，对解码的变换系数进行反 LFNST 变换，再进行反 DCT2 变换得到解码的残差块，最后根据解码的残差块和预测块得到重建块。

可以理解地，本申请实施例可以使用一个高层语法（high level syntax）控制本技术方案的开关。在一些实施例中，该方法还包括：解码码流，确定第二语法标识信息的取值；在第二语法标识信息指示当前序列允许使用第一变换模式时，执行确定当前块的纹理特征索引的步骤；其中，当前序列包括当前块。

20 需要说明的是，在本申请实施例中，第二语法标识信息可以用 `sps_inter_lfnst_nspt_enabled_flag` 表示，而且第二语法标识信息为序列参数集（Sequence Parameter Set，SPS）中的语法元素。

25 还需要说明的是，在本申请实施例中，如果第二语法标识信息的取值为第一值，则确定第二语法标识信息指示当前序列允许使用第一变换模式；如果第二语法标识信息的取值为第二值，则确定第二语法标识信息指示当前序列不允许使用第一变换模式。

在本申请实施例中，第一值与第二值不同，而且第一值和第二值可以是参数形式，也可以是数字形式。具体地，第一语法标识信息可以是写入在概述（profile）中的参数，也可以是一个标志（flag）的取值，这里对此不作具体限定。示例性地，第一值可以为 1，第二值可以为 0；或者，第一值可以为 0，第二值可以为 1；或者，第一值可以为 true，第二值可以为 false；或者，第一值可以为 false，第二值可以为 true。在一种具体的实施例中，第一值为 1，第二值为 0。

30 也就是说，本申请实施例可以使用一个高层语法（high level syntax）控制本技术方案的开关。比如使用一个序列级的 flag，如在序列参数集（sequence parameter set）中增加语法元素 `sps_inter_lfnst_nspt_enabled_flag`。如果 `sps_inter_lfnst_nspt_enabled_flag` 的取值为 1，当前序列允许使用本技术方案；如果 `sps_inter_lfnst_nspt_enabled_flag` 的取值为 0，当前序列不允许使用本技术方案。其中，在允许使用本技术方案的情况下，解码器在解码帧间编码的块时需要解码 LFNST 或 NSPT 的变换核索引，并进行 LFNST 或 NSPT 的反变换处理。当然本申请实施例也可以分别为 LFNST 和 NSPT 设置各自的语法元素，即 `sps_inter_lfnst_enabled_flag` 和 `sps_inter_nspt_enabled_flag`。`sps_inter_lfnst_enabled_flag` 用于指示当前序列是否允许使用 LFNST，`sps_inter_nspt_enabled_flag` 用于指示当前序列是否允许使用 NSPT。

35 当然本申请实施例也可以为帧间的 LFNST/NSPT 和帧内的 LFNST/NSPT 设置同一个 flag，如 `sps_lfnst_nspt_enabled_flag`。示例性地，如果 `sps_inter_lfnst_nspt_enabled_flag` 的取值为 0，那么可以确定当前序列允许不允许使用 LFNST 和 NSPT；如果 `sps_inter_lfnst_nspt_enabled_flag` 的取值为 1，那么可以确定当前序列允许帧内解码的块使用 LFNST 和 NSPT；如果 `sps_inter_lfnst_nspt_enabled_flag` 的取值为 2，那么可以确定当前序列允许帧间解码的块使用 LFNST 和 NSPT。

进一步地，在一些实施例中，该方法还包括：解码码流，确定第二语法标识信息的取值；在第二语法标识信息指示当前序列允许使用第一变换模式时，解码码流，确定第三语法标识信息的取值；在第三语法标识信息指示当前图像允许使用第一变换模式时，执行确定当前块的纹理特征索引的步骤。

在本申请实施例中，当前序列可以包括当前图像，且当前图像包括当前块。其中，第三语法标识信息可以用 `ph_inter_lfnst_nspt_enabled_flag` 表示，而且第三语法标识信息为图像级别的语法元素。

50 还需要说明的是，在本申请实施例中，如果第三语法标识信息的取值为第一值，则确定第三语法标识信息指示当前图像允许使用第一变换模式；如果第三语法标识信息的取值为第二值，则确定第三语法标识信息指示当前图像不允许使用第一变换模式。

进一步地，在一些实施例中，该方法还包括：解码码流，确定第二语法标识信息的取值；在第一语

法标识信息指示当前序列允许使用第一变换模式时,解码码流,确定第四语法标识信息的取值;在第四语法标识信息指示当前片允许使用第一变换模式时,执行确定当前块的纹理特征索引的步骤。

在本申请实施例中,当前序列可以包括当前片,且当前片包括当前块。其中,第四语法标识信息可以用 `sh_inter_lfnst_nspt_enabled_flag` 表示,而且第四语法标识信息为片(slice)级别的语法元素。

5 还需要说明的是,在本申请实施例中,如果第四语法标识信息的取值为第一值,则确定第四语法标识信息指示当前片允许使用第一变换模式;如果第四语法标识信息的取值为第二值,则确定第四语法标识信息指示当前片不允许使用第一变换模式。

在本申请实施例中,第一值与第二值不同,而且第一值和第二值可以是参数形式,也可以是数字形式。具体地,第一语法标识信息可以是写入在概述(profile)中的参数,也可以是一个标志(flag)的取值,这里对此不作具体限定。示例性地,第一值可以为1,第二值可以为0;或者,第一值可以为0,第二值可以为1;或者,第一值可以为true,第二值可以为false;或者,第一值可以为false,第二值可以为true。在一种具体的实施例中,第一值为1,第二值为0。

也就是说,在本申请实施例中,可以使用一个高层语法(high level syntax)控制本技术方案的开关。比如使用一个序列级的flag,如在序列参数集SPS中增加语法元素 `sps_inter_lfnst_nspt_enabled_flag`。如果 `sps_inter_lfnst_nspt_enabled_flag` 的取值为1,当前序列允许使用本技术方案,如果 `sps_inter_lfnst_nspt_enabled_flag` 的取值为0,当前序列不允许使用本技术方案。如果允许本技术方案,那么解码器在解码帧间编码的块时需要解码 `lfnst_nspt_idx`,并进行LFNST或NSPT的处理。当然也可以分别为LFNST和NSPT设置各自的flag,即 `sps_inter_lfnst_enabled_flag` 和 `sps_inter_nspt_enabled_flag`。

进一步地,本申请实施例也可以使用其他级别的语法元素来实现更灵活的控制,如图像参数集(Picture Parameter Set, PPS)的flag,或图像头(picture header)或片头(slice header)的flag等。示例性地,序列参数集SPS确定当前序列是否可以使用本技术方案,如果当前序列是否使用本技术方案,设置一个片头(slice header)的 `sh_inter_lfnst_nspt_enabled_flag` 确定当前slice是否使用本技术方案,提供更高的灵活性。因为针对帧间编码,特别是RA,不同图像或slice的QP差别较大。其中,作为GOP时域层级低的图像通常QP较低,而GOP时域层级越高的图像通常QP越高,本技术方案在QP特别高或者特别低的情况下对压缩效率提升不明显,因而可以设置图像头(picture header)或片头(slice header)的flag来更灵活地控制。

需要说明的是,在本申请实施例中,可以将LFNST和NSPT应用于帧间预测的块。针对帧间不能很好地预测的块,即残差较大的块,相对于相关技术的MTS能更好地去相关性,提高压缩效率。

30 还需要说明的是,在本申请实施例中,除了将LFNST和NSPT应用于帧间预测的块之外,也可以应用于帧内块复制(Intra Block Copy, IBC)的块。

本实施例提供了一种解码方法,确定当前块的纹理特征索引;根据纹理特征索引,确定当前块的变换核组;根据变换核组,确定当前块的变换核;确定当前块的变换系数,并根据变换核对当前块的变换系数进行反变换,确定当前块的残差块;然后根据帧间预测的预测块与残差块,确定当前块的重建块。也就是说,这里建立了纹理特征索引与变换核组之间的对应关系,替代了相关技术中根据帧内预测模式来匹配变换核组的方案,使得LFNST和NSPT也可以应用于帧间预测模式,从而针对帧间预测模式中的不好预测的块,即残差较大的块,不仅可以提高压缩效率,而且还可以提升编解码性能。

在本申请的另一实施例中,参见图43,其示出了本申请实施例提供的一种编码方法的流程示意图。如图43所示,该方法可以包括:

40 S4301: 确定当前块的纹理特征索引。

需要说明的是,在本申请实施例中,该方法应用于编码器。具体来说,基于图37所示编码器100的组成结构,本申请实施例的编码方法主要应用于帧间编码的块。其中,在帧间编码的块采用帧间预测模式时,这里主要是针对帧间预测模式下的NSPT和LFNST变换所提出的优化方案,以提高帧间预测的压缩效率。

45 在这里,NSPT和LFNST都是可以高效处理各种角度的纹理的变换,这里可能有多个变换核,一个变换核可能是专门为某种特定的角度纹理优化的。除了角度的纹理,NSPT和LFNST也包括处理渐变纹理的变换核。其实这些变换核也可以说是训练好的KL变换。也就是说,NSPT和LFNST都可以有多个变换核,每个变换核为特定的纹理设计。特定的纹理包括角度纹理,渐变纹理等。另外,渐变纹理可以进一步扩展如水平渐变纹理,竖直渐变纹理,斜向渐变纹理等。进一步地,本技术方案不局限于只能用于NSPT和LFNST这样的不可分离变换,针对为特定纹理优化的可分离变换也可以适用本技术方案。

50 还需要说明的是,在帧内预测的块中,NSPT和LFNST都包括有多个变换核组,每种帧内预测模

式可以对应一种变换核组。但是帧间预测的块并没有帧内预测模式，这是将 LFNST 和 NSPT 应用于帧间预测的问题。在本申请实施例中，可以根据帧间预测的块推导出一个虚拟的帧内预测模式（索引），或者可称为纹理特征索引。例如，DC 模式和 PLANAR 模式对应渐变的纹理特征，某一种角度预测模式对应这种角度的纹理特征。也就是说，在本申请实施例中，纹理特征索引一方面可以避免在“帧间”
5 出现帧内预测模式，另一方面也更有利于可能的扩展，例如一种帧内预测模式可以对应多种纹理特征，比如 DC 模式可以对应水平渐变纹理、竖直渐变纹理、斜向渐变纹理等。

在本申请实施例中，为了适配帧间预测的 NSPT 和 LFNST 变换的方案，这里提出了虚拟的帧内预测模式（或称为“纹理特征索引”）。所谓虚拟的帧内预测模式，是因为该模式不参与预测，只是为了选择 NSPT 或 LFNST 的变换核组。下面针对纹理特征索引的推导进行详细描述。

10 在一些实施例中，确定当前块的纹理特征索引，可以包括：确定用于推导纹理特征索引的候选像素；根据候选像素，确定当前块的纹理特征索引。

在一种可能的实现方式中，对于候选像素来说，可以是确定当前块的预测块；将预测块中的至少部分像素作为候选像素。

在本申请实施例中，如果预测块存在某种纹理，可以认为残差块中存在相同特征的纹理。这样，关于帧间推导纹理特征索引所使用的候选像素，可以是预测块中的全部像素或者预测块中的部分像素。

在另一种可能的实现方式中，对于候选像素来说，可以是确定当前块的已重建区域的相邻像素；将已重建区域的相邻像素作为候选像素。

在本申请实施例中，关于帧间推导纹理特征索引所使用的候选像素，可以是使用当前块的已重建区域的相邻像素，例如当前块左侧和右侧的已重建区域。因为左侧和上侧的已重建区域虽然不是当前块但
20 与当前块相邻，比如说纹理有相连的情况，因此一定程度上可以用于估计当前块的纹理。

在又一种可能的实现方式中，考虑使用更多的像素，对于候选像素来说，可以是已将重建区域的相邻像素和预测块中的至少部分像素共同作为候选像素。

在本申请实施例中，关于帧间推导纹理特征索引所使用的候选像素，也可以是同时使用当前块的预测块和当前块左侧和上侧的已重建区域的相邻像素，这样用于推导纹理特征索引的像素更多，使得所推导的纹理特征索引更准确。

25 在再一种可能的实现方式中，考虑硬件的流水线长度，对于候选像素来说，可以是确定当前块的参考块；将参考块中的至少部分像素作为候选像素。

在本申请实施例中，对当前块而言，预测块得到以后才能推导纹理特征索引，确定纹理特征索引之后才能确定 LFNST/NSPT 的变换核，然后才能够进行变换及后续编码流程。在这里，如果使用参考块来推导纹理特征索引，理论上产生预测块的过程和推导纹理特征索引的过程可以并行进行，这样就比使用预测块的流水线要短。因而，还可以使用参考块来推导纹理特征索引，或者在产生预测块的过程中同步推导纹理特征索引，从而能够缩短流水线长度。

30 在一些实施例中，对于当前块的参考块来说，可以确定参考块为整像素的参考块；或者，确定参考块为分像素的参考块。

35 在一种可能的实现方式中，参考块时可以直接使用整像素的参考块，因为纹理方向不会有明显的区别。比如一个双向预测的帧间块，可以根据整像素的参考块推导虚拟的帧内预测模式。

在另一种可能的实现方式中，参考块也可以使用经过插值滤波（interpolation filter）的分像素的参考块。具体地，确定当前块进行双向预测时的两个参考图像块；对两个参考图像块进行分像素的插值滤波，确定两个分像素的参考图像块；对两个分像素的参考图像块进行加权组合，确定当前块的参考块。

40 示例性地，对于一个双向预测的帧间块，首先需要对两个参考块进行分像素的插值滤波，然后再将两个分像素的参考块进行加权组合。那么可以在得到两个插值滤波后的参考块后，一方面可以根据插值滤波后的参考块进行加权平均得到预测值，另一方面也可以根据插值滤波后的参考块推导纹理特征索引。

还需要说明的是，在本申请实施例中，用于推导纹理特征索引的候选像素的数量可以为至少一个，例如 1 个、2 个、3 个或更多个。在一些实施例中，可以根据当前块的尺寸参数，确定候选像素的数量。

45 也就是说，在确定当前块的纹理特征索引时，使用多少个候选像素可以由当前块的尺寸参数确定。示例性地，如果当前块的尺寸小，那么可以对所有可用的像素进行统计；如果当前块的尺寸大，那么可以对当前块降采样进行统计，比如水平方向和/或竖直方向每 2 个、或 4 个、或 8 个像素中的一个像素进行统计。或者，如果当前块的水平或竖直某一个方向的尺寸小于或等于 8，那么对该方向上的所有可用的像素进行统计；否则，如果当前块的水平或竖直某一个方向的尺寸小于或等于 16，那么对该方向上的每 2 个像素中的一个像素进行统计；否则，对该方向上的每 4 个像素中的一个像素进行统计，这里
50 不作具体限定。

在一些实施例中，根据候选像素，确定当前块的纹理特征索引，可以包括：确定候选像素的水平梯

度值和竖直梯度值；根据候选像素的水平梯度值和竖直梯度值，确定候选像素对应的纹理特征索引以及梯度强度值；根据候选像素对应的纹理特征索引以及梯度强度值，构建纹理特征统计表；根据纹理特征统计表，确定当前块的纹理特征索引。

需要说明的是，在本申请实施例中，在根据候选像素的水平梯度值和竖直梯度值，确定候选像素对应的纹理特征索引以及梯度强度值时，可以包括：根据候选像素的水平梯度值和竖直梯度值进行角度映射，确定候选像素对应的纹理特征索引；以及根据候选像素的水平梯度值和竖直梯度值进行梯度强度计算，确定候选像素对应的梯度强度值。

在一种具体的实施例中，根据候选像素的水平梯度值和竖直梯度值进行角度映射，确定候选像素对应的纹理特征索引，可以包括：根据候选像素的水平梯度值和竖直梯度值，利用预设查找表确定候选像素对应的纹理特征索引。

在本申请实施例中，候选像素的水平梯度值可以用 $grad_x$ 表示，候选像素的竖直梯度值可以用 $grad_y$ 表示。这样，根据 $grad_x$ 和 $grad_y$ 导出虚拟的帧内预测模式可以通过查表来实现。

示例性地，如果 $abs(grad_x)$ 等于0且 $abs(grad_y)$ 不等于0，那么有水平方向纹理，对应一些技术中的帧内预测模式18。如果 $abs(grad_y)$ 等于0且 $abs(grad_x)$ 不等于0，那么有竖直方向纹理，对应一些技术中的帧内预测模式50。在 $abs(grad_x)$ 和 $abs(grad_y)$ 都不等于0的情况下，如果 $abs(grad_x)$ 等于 $abs(grad_y)$ ，且 $grad_x$ 和 $grad_y$ 符号相同，对应一些技术中的帧内预测模式34。如果 $abs(grad_x)$ 等于2倍的 $abs(grad_y)$ ，且 $grad_x$ 和 $grad_y$ 符号相同，对应一些技术中的帧内预测模式40。另外，其他情况可以按照相同的原则查表确定。

在一种具体的实施例中，根据候选像素的水平梯度值和竖直梯度值进行梯度强度计算，确定候选像素对应的梯度强度值，可以包括：对水平梯度值的绝对值和竖直梯度值的绝对值进行加法运算，确定候选像素对应的梯度强度值。

在这里，候选像素对应的梯度强度值可以记为 amp 。示例性地， $amp = abs(grad_x) + abs(grad_y)$ 。

需要说明的是，在本申请实施例中，针对候选像素的水平梯度值和竖直梯度值可以使用索贝尔算子计算得到。示例性地，对于索贝尔算子来说，具体如下所示：

水平梯度值的算子：

$$\begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$$

竖直梯度值的算子：

$$\begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$$

如此，假设像素位置为 (x, y) 的像素值为 $P_{x,y}$ ，则水平梯度值 $grad_x$ 和竖直梯度值 $grad_y$ 的计算如下所示：

$$grad_x = P_{x+1,y-1} + 2 * P_{x+1,y} + P_{x+1,y+1} - P_{x-1,y-1} - 2 * P_{x-1,y} - P_{x-1,y+1} \quad (24)$$

$$grad_y = P_{x-1,y+1} + 2 * P_{x,y+1} + P_{x+1,y+1} - P_{x-1,y-1} - 2 * P_{x,y-1} - P_{x+1,y-1} \quad (25)$$

在一些实施例中，根据候选像素对应的纹理特征索引以及梯度强度值，构建纹理特征统计表，可以包括：在候选像素的数量为至少一个时，确定至少一个纹理特征索引以及对应的至少一个梯度强度值；根据至少一个纹理特征索引确定具有互异特性的至少一种参考纹理特征索引，以及根据至少一个梯度强度值，对归属于同一参考纹理特征索引的梯度强度值进行累加计算，确定至少一种参考纹理特征索引对应的梯度强度累加值；根据至少一种参考纹理特征索引和至少一种参考纹理特征索引对应的梯度强度累加值，构建纹理特征统计表。

也就是说，在本申请实施例中，以预测块中的至少部分像素作为候选像素为例，对预测块中的全部或部分像素计算其梯度值，一般来说可以计算水平梯度值和竖直梯度值，这里可以使用索贝尔算子来计算梯度值。对某一个像素，根据其水平梯度值和竖直梯度值可以推测该像素的纹理方向，比如水平梯度值非零，竖直梯度值为零，则该像素的纹理是竖直方向。反之，水平梯度值为零，竖直梯度值非零，则该像素的纹理是水平方向。比如水平梯度值和竖直梯度值相等且不为零，则该像素的纹理是45度。当然，本申请实施例还有很多其他情况是水平梯度值和竖直梯度值都不为零，可以根据它们的比值确定该像素的纹理的方向。这样可以根据每一个像素的梯度强度值可以对应到对应的纹理特征索引。从而构建一个纹理特征统计表，将每一个计算的像素的梯度强度值累加到统计表中对应的纹理特征索引项中，以得到最终的纹理特征统计表，进而根据纹理特征统计表可以确定当前块的纹理特征索引。

在一种可能的实现方式中，在根据纹理特征统计表确定当前块的纹理特征索引时，可以包括：确定

纹理特征统计表中的最大梯度强度累加值;将最大梯度强度累加值对应的参考纹理特征索引确定为当前块的纹理特征索引。

在另一种可能的实现方式中,在根据纹理特征统计表确定当前块的纹理特征索引时,可以包括:确定纹理特征统计表中的最大梯度强度累加值;在最大梯度强度累加值小于第一阈值时,将当前块的纹理特征索引设置为 DC 模式或 PLANANR 模式。

在又一种可能的实现方式中,在根据纹理特征统计表确定当前块的纹理特征索引时,可以包括:确定纹理特征统计表中所有梯度强度累加值的统计和值;在统计和值小于第二阈值时,将当前块的纹理特征索引设置为 DC 模式或 PLANANR 模式。

也就是说,在本申请实施例中,对于虚拟的帧内预测模式的导出,由于虚拟的帧内预测模式(索引)也可以称为纹理特征索引,因而这里也可以称作纹理特征索引的导出。

在本申请实施例中,可以利用帧间预测块来导出虚拟的帧内预测模式。如果预测块存在某种纹理,可以想象残差块中存在相同特征的纹理。示例性地,这里可以对预测块中的全部或部分像素计算其梯度值,一般来说可以计算水平梯度值和竖直梯度值,这里可以使用索贝尔算子来计算梯度值。然后根据水平梯度值和竖直梯度值来确定对应的梯度强度值,再根据梯度强度值对应到对应的帧内预测模式,从而构建一个帧内预测模式统计表(即前述的“纹理特征统计表”),需要将每一个计算的候选像素的梯度强度值累加到统计表中对应的帧内预测模式项中。在梯度统计完成后,一种可能的实现方式为帧内预测模式统计表中累计最多的一个帧内预测模式即为当前块的虚拟的帧内预测模式。或者,另一种可能的实现方式为当统计的累计梯度强度最高的帧内预测模式的梯度强度累加值小于某一个阈值时,确定虚拟的帧内预测模式为 DC 模式或 PLANANR 模式。或者,又一种可能的实现方式为当所有统计的像素的梯度强度累加值之和小于某一个阈值时,确定虚拟的帧内预测模式为 DC 模式或 PLANANR 模式。

另外,针对 DIMD 也需要进行类似的统计导出帧内预测模式,但是区别在于 DIMD 使用的是当前块的周边重建区域,这里使用的是当前块的预测块。在实现上,本方案可以和 DIMD 复用部分逻辑。

还需要说明的是,关于在帧间导出虚拟帧内预测模式所使用的像素,这里也可以是使用当前块左侧和上侧的已重建区域,和 DIMD 的做法类似。因为左侧和上侧的已重建区域虽然不是当前块但是与当前块相邻,比如说纹理有相连的情况,因而一定程度上可以用来估计当前块的纹理。另一种可能是同时使用当前块的预测块和当前块左侧和上侧的已重建区域,这样可用来推理的虚拟的帧内预测模式的像素更多。又一种可能是考虑硬件的流水线长度。对当前块而言,预测块得到以后才能导出虚拟的帧内预测模式,确定了虚拟的帧内预测模式才能确定 LFNST/NSPT 的变换核,才能进行反变换及后续流程。如果使用参考块来导出虚拟的帧内预测模式,理论上产生预测块的过程和推导虚拟的帧内预测模式的过程可以并行进行,这样就比上述的设计流水线要短。因而,本申请实施例还可以使用参考块来导出虚拟的帧内预测模式,或者在产生预测块的过程中同步推导虚拟的帧内预测模式。其中,使用参考块时可以直接使用整像素的参考块,因为纹理方向不会有明显的区别。比如一个双向预测的帧间块,一方面根据整像素的参考块推导虚拟的帧内预测模式,另一方面对 2 个参考块进行分像素的插值滤波,然后再将 2 个分像素的参考块进行加权组合。具体地,这里也可以使用经过插值滤波的分像素的参考块,比如一个双向预测的帧间块,需要首先对 2 个参考块进行分像素的插值滤波,然后再将 2 个分像素的参考块进行加权组合。那么可以在得到 2 个插值滤波后的参考块后,一方面根据插值滤波后的参考块进行加权平均得到预测值,另一方面根据插值滤波后的参考块推导虚拟的帧内预测模式。

在一些实施例中,确定当前块的纹理特征索引,还可以包括:确定当前块进行几何划分模式时的角度索引;根据角度索引,确定当前块的纹理特征索引。

在本申请实施例中,对于几何划分模式 GPM 的虚拟的帧内预测模式的导出,GPM 的预测块是把个不同运动的预测块组合到一起得到的,因而它可能包含 2 个物体的内容,通常情况下边界很难被完美地预测,因而 2 个物体的边界处可能存在较大的残差,而这个残差的方向和 GPM 的“划分”方向是一致的。因此对于 GPM 来说,可以使用 GPM 的“划分”模式来确定虚拟的帧内预测模式。

在本申请实施例中,所述划分线可以认为是 GPM 的权重矩阵中权重发生变化的区域中权重相同的点所组成的线。或者说权重为中值的点组成的线,权重为中值的点可能不在整像素位置。以权重为 0 到 8 为例,中值可以是 4。图 40 示出了一种 GPM 模式的划分线示意图。如图 40 所示,加粗实线为这里所述的划分线。

其实 GPM 的权重是根据划分线导出的。下面以一些技术中的 GPM 的一些过程举例,下面的 merge_gpm_partition_idx[xCb][yCb]用来确定“划分”,它就是本申请实施例所说的权重导出模式。根据 merge_gpm_partition_idx[xCb][yCb]利用前述表 6 确定 GPM 的“划分”角度索引变量 angleIdx 和距离索引变量 distanceIdx。其中,角度索引变量 angleIdx 和距离索引变量 distanceIdx 就可以认为是确定划分线的变量,它们分别确定划分线的角度和偏移量。

因为三个颜色分量 (component, 如 Y、Cb、Cr) 都可以使用 GPM, 所以标准文本将一个分量产生 GPM 的预测样本矩阵的过程分装到了一个子流程里面, 即 GPM 的加权预测过程 (Weighted sample prediction process for geometric partitioning mode), 三个颜色分量都会调用这个流程, 只是调用的参数不同, 这里只用亮度分量举例。当前亮度块的预测矩阵 $\text{predSamplesL}[\text{xL}][\text{yL}]$ (其中 $\text{xL} = 0.. \text{cbWidth} - 1$, $\text{yL} = 0.. \text{cbHeight} - 1$) 由 GPM 的加权预测过程导出。其中的 nCbW 设为 cbWidth , nCbH 设为 cbHeight , 两个预测模式做的预测样本矩阵 predSamplesLAL 和 predSamplesLBL , 还有 angleIdx , distanceIdx 作为输入。

下面是 GPM 的加权预测过程导出的过程:

这个过程的输入有:

- 10 - 当前块的宽度 nCbW 当前块的高度 nCbH ;
- 2 个 $(\text{nCbW}) \times (\text{nCbH})$ 的预测样本矩阵 predSamplesLA 和 predSamplesLB ;
- GPM 的“划分”角度索引变量 angleIdx ;
- GPM 的距离索引变量 distanceIdx ;
- 分量索引变量 cIdx 。因为本实施例中只以亮度分量举例, 其中 cIdx 为 0, 表示亮度分量。

15 这个过程的输出是 $(\text{nCbW}) \times (\text{nCbH})$ 的 GPM 预测样本矩阵 pbSamples 。

变量 nW 、 nH 、 shift1 、 offset1 、 displacementX 、 displacementY 、 partFlip , 还有 shiftHor 按如下方法导出:

```

nW = ( cIdx == 0 ) ? nCbW : nCbW * SubWidthC;
nH = ( cIdx == 0 ) ? nCbH : nCbH * SubHeightC;
20 shift1 = Max( 5, 17 - BitDepth ), 其中 BitDepth 是编解码的比特深度;
offset1 = 1 << ( shift1 - 1 );
displacementX = angleIdx;
displacementY = ( angleIdx + 8 ) % 32;
partFlip = ( angleIdx >= 13 && angleIdx <= 27 ) ? 0 : 1;
25 shiftHor = ( angleIdx % 16 == 8 || ( angleIdx % 16 != 0 && nH >= nW ) ) ? 0 : 1;
变量 offsetX and offsetY 按如下方法导出:
- 如果 shiftHor 的值为 0:
offsetX = ( -nW ) >> 1;
offsetY = ( ( -nH ) >> 1 ) + ( angleIdx < 16 ? ( distanceIdx * nH ) >> 3 : - ( ( distanceIdx * nH ) >> 3 ) );
30 - 否则 (即 shiftHor 的值为 1):
offsetX = ( ( -nW ) >> 1 ) + ( angleIdx < 16 ? ( distanceIdx * nW ) >> 3 : - ( ( distanceIdx * nW ) >> 3 ) );
offsetY = ( -nH ) >> 1;
预测样本矩阵  $\text{pbSamples}[\text{x}][\text{y}]$  (其中  $\text{x} = 0.. \text{nCbW} - 1$ ,  $\text{y} = 0.. \text{nCbH} - 1$ ) 按如下方法导出:
- 变量  $\text{xL}$  和  $\text{yL}$  按如下方法导出:
35  $\text{xL} = ( \text{cIdx} == 0 ) ? \text{x} : \text{x} * \text{SubWidthC}$ ;
 $\text{yL} = ( \text{cIdx} == 0 ) ? \text{y} : \text{y} * \text{SubHeightC}$ ;
- 表示当前位置预测样本权重的变量  $\text{wValue}$  按如下方法导出,  $\text{wValue}$  即为  $(\text{x}, \text{y})$  点的第一
预测模式的预测矩阵的预测值  $\text{predSamplesLA}[\text{x}][\text{y}]$  的权重, 而  $(8 - \text{wValue})$  即为  $(\text{x}, \text{y})$  点的第一预
测模式的预测矩阵的预测值  $\text{predSamplesLB}[\text{x}][\text{y}]$  的权重。其中, 前述表 7 提供了一种距离矩阵  $\text{disLut}$ 
40 的定义示例。如此,  $\text{disLut}$  的取值可以按表 7 确定。
在这里,  $\text{weightIdx} = ( ( ( \text{xL} + \text{offsetX} ) << 1 ) + 1 ) * \text{disLut}[\text{displacementX}] +$ 
 $(( ( \text{yL} + \text{offsetY} ) << 1 ) + 1 ) * \text{disLut}[\text{displacementY}];$ 
 $\text{weightIdxL} = \text{partFlip} ? 32 + \text{weightIdx} : 32 - \text{weightIdx};$ 
 $\text{wValue} = \text{Clip3}( 0, 8, ( \text{weightIdxL} + 4 ) >> 3 );$ 
45 - 预测样本的值按如下方法导出:
 $\text{pbSamples}[\text{x}][\text{y}] = \text{Clip3}( 0, ( 1 << \text{BitDepth} ) - 1, ( \text{predSamplesLA}[\text{x}][\text{y}] * \text{wValue} +$ 
 $\text{predSamplesLB}[\text{x}][\text{y}] * ( 8 - \text{wValue} ) + \text{offset1} ) >> \text{shift1} )$ 。

```

这里需要注意的是, 标准文本中对每一个位置来推导一个权重值 wValue , 然后计算一个 GPM 的预测值 $\text{pbSamples}[\text{x}][\text{y}]$ 。因为这种方式权重 wValue 不必写成一个矩阵的形式, 但是可以理解如果把每个位置的 wValue 都保存到一个矩阵里, 那它就是一个权重矩阵。每个点分别计算权重并加权得到 GPM 的预测值, 或者计算出所有的权重再统一加权得到 GPM 的预测样本矩阵其原理是一样的。而本文中的诸多描述中使用权重矩阵的说法, 是为了表述更容易理解, 用权重矩阵画图更直观, 其实也可以按每个

位置的权重来描述。比如权重矩阵导出模式也可以说成权重导出模式。

这样, 根据权重 wValue 的导出过程可以看出 wValue 是根据 angleIdx 和 distanceIdx 导出的。所以在确定划分线对应的帧内预测模式时, 也可以先确定 angleIdx, 然后根据 angleIdx 确定其对应的帧内角度预测模式。因为根据上面的表 6 也可以看出多个权重导出模式对应一个 angleIdx。一个可能的 angleIdx 和帧内角度预测模式的对应关系如前述表 8。可以看到有些 angleIdx 对应的是 0, 那是因为这些 angleIdx 在 GPM 中没有被用到, 也就是如图 10 中的 64 种权重的示例。需要理解的是, 如果在某一个版本中某些 angleIdx 发生了变化, 比如以后更多的 angleIdx 被使用到, 或者帧内角度预测模式发生了变化, 比如以后有更多的帧内预测模式, 那么这个对应关系表也可以相应变化, 这里并不作具体限定。

进一步地, 在本申请实施例, 当前块的纹理特征索引也可以直接复制参考块的纹理特征索引。在一些实施例中, 确定当前块的纹理特征索引, 还可以包括: 根据参考块的帧内预测模式或参考块的虚拟的帧内预测模式确定当前块的虚拟的帧内预测模式, 或者说根据参考块的纹理特征索引确定当前块的纹理特征索引。如果当前块的参考块有帧内编码的块, 那么帧内编码的块有帧内预测模式, 可以将参考块的帧内预测模式作为当前块的虚拟的帧内预测模式。如果当前块的参考块包含多个帧内编码的块, 可以根据坐标来确定某个位置的帧内预测模式, 例如以当前块的中心位置对应的参考块的帧内预测模式作为当前块的虚拟的帧内预测模式。可以对每一个帧内编码的块都保存虚拟的帧内预测模式, 如果当前块的参考块是帧内编码的块, 那么它也可以找到虚拟的帧内预测模式。虚拟的帧内预测模式也可以按一定的粒度来存储, 例如每个 4×4 的像素为一个最小存储单元, 每个最小存储单元共享同一个帧内预测模式或虚拟的帧内预测模式。如果当前块的参考块包含多个 4×4 的最小存储单元, 可以根据坐标来确定某个位置的(虚拟的)帧内预测模式, 例如以当前块的中心位置对应的参考块的最小存储单元的(虚拟的)帧内预测模式作为当前块的虚拟的帧内预测模式。

如此, 在确定出虚拟的帧内角度预测模式(即纹理特征索引)之后, 就可以根据纹理特征索引来确定 NSPT/LFNST 的变换核组。

进一步地, 在本申请实施例, 还可以构建候选列表, 以此确定虚拟的帧内角度预测模式。因此, 在一些实施例中, 确定当前块的纹理特征索引, 还可以包括: 构建候选列表, 其中, 候选列表包括预设数量的候选纹理特征索引; 对预设数量的候选纹理特征索引分别进行代价值计算, 确定预设数量的候选纹理特征索引各自的代价值; 从预设数量的候选纹理特征索引各自的代价值中确定最小代价值, 将最小代价值对应的候选纹理特征索引确定为当前块的纹理特征索引。

需要说明的是, 在本申请实施例, 预设数量可以为 1 个、2 个、3 个或者更多个。如果候选列表中仅包括 1 个候选纹理特征索引, 那么可以直接将这 1 个候选纹理特征索引作为当前块的纹理特征索引, 无需进行代价值计算。

需要说明的是, 在本申请实施例, 对预设数量的候选纹理特征索引分别进行代价值计算, 确定预设数量的候选纹理特征索引各自的代价值。在这里, 代价值可以根据率失真优化(Rate Distortion Optimization, RDO)的代价结果进行确定, 也可以根据绝对误差和(Sum of Absolute Difference, SAD)的代价结果进行确定, 甚至也可以根据绝对变换差和(Sum of Absolute Transformed Difference, SATD)的代价结果进行确定, 但是这里并不作任何限定。

示例性地, 以率失真优化方式为例, 可以根据预设数量的候选纹理特征索引分别进行率失真代价值计算, 确定预设数量的候选纹理特征索引各自的代价值, 然后从中选取最小代价值, 并将最小代价值对应的候选纹理特征索引确定为当前块的纹理特征索引, 从而能够提升编码效率。

进一步地, 在一些实施例中, 该方法还包括: 根据当前块的纹理特征索引, 确定当前块的特征索引序号; 其中, 特征索引序号用于指示当前块的纹理特征索引在候选列表中的编号; 对当前块的特征索引序号进行编码处理, 将所得到的编码比特写入码流。

需要说明的是, 前述实施例提到使用当前块的预测块或参考块或当前块周边的重建图像推导出虚拟的帧内预测模式。在一种具体的实施例中, 可以选择梯度强度值最大的一个帧内预测模式作为虚拟的帧内预测模式。除此之外, 也可以增加一个步骤, 构建一个候选列表, 即虚拟的帧内预测模式的候选列表, 或者特征索引的候选列表。然后使用一个语法元素写在码流中指示最终选择了哪一个候选。这里所使用的语法元素即当前块的特征索引序号, 可以用 lfnst_nspt_feature_idx 表示。

也就是说, 编码器编码时可以用上述方法构建候选列表, 然后对所有的可能的候选分别尝试进行编码, 取编码代价最小的一个候选确定为当前块使用的变换核, 以及确定 lfnst_nspt_feature_idx 的取值并写入码流。其中, lfnst_nspt_feature_idx 的取值可以为 0、1、2、3 等整数。示例性地, 如果选择候选列表的第一个元素, 那么可以确定 lfnst_nspt_feature_idx 的取值为 0; 如果选择候选列表的第二个元素, 那么可以确定 lfnst_nspt_feature_idx 为 1。

还需要说明的是, 在本申请实施例, 如果候选列表中仅包括 1 个候选纹理特征索引, 那么不仅可

以直接将这 1 个候选纹理特征索引作为当前块的纹理特征索引,无需进行代价值计算;而且也可以不用确定 `lfnst_nspt_feature_idx` 的取值,也不需要将其取值写入码流。

5 在一些实施例中,对于构建候选列表,可以包括:将纹理特征统计表中的至少一种参考纹理特征索引按照对应的梯度强度累加值由大到小进行排序,确定排序靠前的预设数量个参考纹理特征索引;根据排序靠前的预设数量个参考纹理特征索引,构建候选列表。

在一些实施例中,对于构建候选列表,可以方法还包括:确定纹理特征统计表中最大梯度强度累加值对应的第一纹理特征索引,以及确定当前块进行几何划分模式时对应的第二纹理特征索引;根据第一纹理特征索引和第二纹理特征索引,构建候选列表。

10 也就是说,在本申请实施例中,构建候选列表的一个示例是,仍按照上述计算梯度强度的方法,按梯度强度从大到小选择最靠前的 N 个帧内预测模式(纹理特征索引)构成候选列表。其中, N 的取值可能为 2、3、4 等。另一个示例是假设 N 的取值为 2,也就是梯度强度最大的帧内预测模式(纹理特征索引)为候选列表的第一个元素,梯度强度第二大的帧内预测模式(纹理特征索引)为候选列表的第二个元素。又一个示例是对于 GPM,可以设置候选列表第一个元素为 GPM 的“划分”模式对应的帧内预测模式(纹理特征索引),第二个元素为梯度强度最大的帧内预测模式(纹理特征索引),也可以反过来;这里不作具体限定。

15 在一些实施例中,对于构建候选列表,还可以包括:确定当前块的参考块;根据参考块中至少一个候选位置对应的纹理特征索引,构建候选列表。

20 也就是说,在本申请实施例中,根据参考块的帧内预测模式(纹理特征索引)构建候选列表,例如根据参考块中某几个位置对应的 4×4 的最小存储单元的帧内预测模式(纹理特征索引)构建候选列表,记当前块左上角的坐标为 (x, y) ,当前块的宽度 `nCbW` 当前块的高度 `nCbH`,当前块的中心点 `posC` 为 $(x+nCbW/2, y+nCbH/2)$,左上角 `posTL` (x, y) ,右上角 `posTR` $(x+nCbW-1, y)$,左下角 `posBL` $(x, y+nCbH-1)$,右下角 `posBR` $(x+nCbW-1, y+nCbH-1)$ 。依次用 `posC`, `posTL`, `posTR`, `posBL`, `posBR` 对应的最小存储单元的帧内预测模式(纹理特征索引)构建候选列表。在这里,候选位置可以为 `posC`, `posTL`, `posTR`, `posBL`, `posBR` 中的至少一项,但是并不局限于此。

25 还需要说明的是,编码端在构建列表的时候需要保证候选列表中的每一项不重复,如果可用的帧内预测模式(纹理特征索引)无法填满列表,可以加入默认的帧内预测模式(纹理特征索引),例如水平方向,竖直方向的帧内预测模式(纹理特征索引)。

30 还需要说明的是,这种方法不仅可以适用于帧间预测,还可以适用于帧内预测。其中,如果 N 的取值为 2,那么只需要一个二元符号来表示;如果 N 的取值大于 2,那么可以使用截断一元码。在这里,所有的二元符号都可以使用 CABAC 的上下文模型来进行编码。

示例性地,参见表 9,针对当前块的特征索引序号 `lfnst_nspt_feature_idx` 的取值,可以使用截断一元码进行二值化处理,然后基于上下文模型对每一个编码比特进行编码。

表 9

<code>lfnst_nspt_feature_idx</code> 的取值	编码比特
0	0
1	10
2	110
3	111

35 这样,在构建候选列表之后,根据解码码流所得到的特征索引序号,从而也能够确定出虚拟的帧内角度预测模式(即当前块的纹理特征索引)。

S4302:根据纹理特征索引,确定当前块的变换核组。

S4303:根据变换核组,确定当前块的变换核。

40 需要说明的是,在确定出当前块的纹理特征索引之后,根据纹理特征索引与变换核组之间的对应关系,就可以确定当前块的变换核组。示例性地,一些技术中的 LFNST 共有 4 组变换核组,帧内预测模式与变换核组之间的对应关系详见表 4;在一些技术中的 LFNST 也可以有更多的变换核组,例如 35 组,帧内预测模式与变换核组之间的对应关系详见表 5。

另外,每一个变换核组可以包括至少两个变换核。示例性地,变换核组可以包括 2 个变换核,或者也可以包括 3 个变换核。但是一个变换核组也可以包括 4 个可选的变换核,或者更多数量可选的变换核,这里均不作具体限定。

45 在一些实施例中,变换核组包括至少一个候选变换核。相应地,根据变换核组,确定当前块的变换核,可以包括:对至少一个候选变换核分别进行代价值计算,确定至少一个候选变换核各自的代价值;从至少一个候选变换核各自的代价值中确定最小代价值,将最小代价值对应的候选变换核确定为当前块

的变换核。

在一种可能的实现方式中，对至少一个候选变换核分别进行代价值计算，这里以第一候选变换核为例，该方法可以包括：基于第一候选变换核对当前块的残差块进行变换与量化，确定当前块的第一候选量化系数，并对第一候选量化系数进行熵编码处理，确定第一候选变换核的第一代价值；对第一候选量化系数进行反量化与反变换，确定当前块的第一候选残差块，并根据第一候选残差块确定当前块的第一候选预测块；根据第一候选预测块与当前块的原始图像进行代价计算，确定第一候选变换核的第二代价值；根据第一候选变换核的第一代价值和第二代价值，确定第一候选变换核的代价值。

需要说明的是，在本申请实施例中，第一候选变换核为至少一个候选变换核中的任意一个，如此可以确定这至少一个候选变换核各自的代价值。

还需要说明的是，在本申请实施例中，第一代价值可以表征第一候选变换核在码流中开销的代价，第二代价值表征第一候选变换核的失真的代价。在这里，根据第一候选变换核的第一代价值和第二代价值，确定第一候选变换核的代价值，可以包括：对第一候选变换核的第一代价值和第二代价值进行求和运算，将两者的和值作为第一候选变换核的代价值。

还需要说明的是，在本申请实施例中，这里的代价值计算，可以根据 RDO 的代价结果进行确定，也可以根据 SAD 的代价结果进行确定，甚至也可以根据 SATD 的代价结果进行确定，但不作具体限定。

还需要说明的是，在本申请实施例中，变换核组可以包括 1 个、2 个、3 个或者更多个候选变换核。如果变换核组仅包括 1 个候选变换核，那么可以直接将这 1 个候选变换核作为当前块的变换核，无需进行代价值计算。

在一些实施例中，该方法还包括：确定当前块的第一语法标识信息的取值；其中，第一语法标识信息用于指示当前块是否使用第一变换模式以及对应使用的变换核索引；对第一语法标识信息的取值进行编码处理，将所得到的编码比特写入码流。

需要说明的是，在本申请实施例中，第一语法标识信息可以用 `lfnst_nspt_idx` 表示。其中，第一语法标识信息可以用于指示当前块是否使用第一变换模式，而且在当前块使用第一变换模式时，第一语法标识信息的取值还可以用于指示当前块使用的变换核索引。

还需要说明的是，在本申请实施例中，第一变换模式可以为 LFNST 变换或者 NSPT 变换。也就是说，`lfnst_nspt_idx` 用于指示当前块是否使用 LFNST 变换或者 NSPT 变换，而且在当前块使用 LFNST 变换或者 NSPT 变换时，`lfnst_nspt_idx` 的取值还可以用于指示当前块使用的变换核索引。其中，可以根据当前块的尺寸参数来确定当前块是使用 LFNST 变换还是使用 NSPT 变换。示例性地，如果当前块的尺寸参数较小，那么可以确定 `lfnst_nspt_idx` 用于指示当前块是否使用 NSPT 变换以及对应使用的 NSPT 的变换核索引；如果当前块的尺寸参数较大，那么可以确定 `lfnst_nspt_idx` 用于指示当前块是否使用 LFNST 变换以及对应使用的 LFNST 的变换核索引。

还需要说明的是，在本申请实施例中，对于第一语法标识信息 `lfnst_nspt_idx`，也可以使用 `lfnst_idx` 和 `nspt_idx` 两个语法标识信息来替代。其中，`lfnst_idx` 用于指示当前块是否使用 LFNST 变换以及对应使用的 LFNST 的变换核索引，`nspt_idx` 用于指示当前块是否使用 NSPT 变换以及对应使用的 NSPT 的变换核索引，这里不作具体限定。

在一些实施例中，该方法还包括：确定当前块的变换核索引；其中，变换核索引用于指示当前块的变换核在变换核组中的编号；若当前块使用帧内预测模式，则对当前块的变换核索引进行编码处理，将所得到的编码比特写入码流；若当前块使用帧间预测模式，则根据当前块的变换核索引确定第一语法标识信息的取值，对第一语法标识信息的取值进行编码处理，将所得到的编码比特写入码流。

需要说明的是，在本申请实施例中，在当前块使用帧内预测模式时，第一语法标识信息用于指示当前块是否使用第一变换模式；在当前块使用第一变换模式时，还需要确定当前块的变换核索引，并且对当前块的变换核索引进行编码处理，将所得到的编码比特写入码流；以便后续在解码端，根据变换核组和解码得到的变换核索引来确定当前块的变换核；其中，变换核索引用于指示当前块的变换核在变换核组的编号。示例性地，如果当前块的变换核索引为 0，那么当前块的变换核为变换核组的第一个变换核；如果当前块的变换核索引为 1，那么当前块的变换核为变换核组的第二个变换核；如果当前块的变换核索引为 2，那么当前块的变换核为变换核组的第三个变换核。

还需要说明的是，在本申请实施例中，在当前块使用帧间预测模式时，LFNST/NSPT 一个变换核组有 3 个变换核可选，再加上不使用 LFNST/NSPT 的情况，也就是当前块针对 LFNST/NSPT 有 4 种可能的选择。因此，这里可以使用 `lfnst_nspt_idx` 指示当前块是否使用第一变换模式以及当前块使用的变换核索引。这样，在确定当前块的变换核索引之后，根据当前块的变换核索引就可以确定第一语法标识信息的取值，对第一语法标识信息的取值进行编码处理，将所得到的编码比特写入码流；以便后续在解码端，根据变换核组和解码得到的第一语法标识信息的取值来确定当前块的变换核。

在一些实施例中,对于第一语法标识信息的取值,如果当前块不使用第一变换模式,那么确定第一语法标识信息的取值为第一数值;如果当前块使用第一变换模式且当前块使用的变换核索引为0,那么确定第一语法标识信息的取值为第二数值;如果当前块使用第一变换模式且当前块使用的变换核索引为1,那么确定第一语法标识信息的取值为第三数值;如果当前块使用第一变换模式且当前块使用的变换核索引为2,那么确定第一语法标识信息的取值为第四数值。

在一种具体的实施例中,第一数值等于0,第二数值等于1,第三数值等于2,第四数值等于3。

示例性地,如果当前块不使用 LFNST/NSPT,那么确定第一语法标识信息的取值为0;如果当前块使用 LFNST/NSPT,而且当前块的变换核为变换核组的第一个变换核,那么确定第一语法标识信息的取值为1;如果当前块使用 LFNST/NSPT,而且当前块的变换核为变换核组的第二个变换核,那么确定第一语法标识信息的取值为2;如果当前块使用 LFNST/NSPT,而且当前块的变换核为变换核组的第三个变换核,那么确定第一语法标识信息的取值为3。也就是说,在本申请实施例中, `lfnst_nspt_idx` 的取值可以为0、1、2、3等。

还需要说明的是,在本申请实施例中,对于第一语法标识信息 `lfnst_nspt_idx`,也可以使用另外两个语法标识信息来替代。其中,一个语法标识信息用于指示当前块是否使用 LFNST/NSPT,另一个语法标识信息用于指示在当前块使用 LFNST/NSPT 时对应的变换核索引。这样,在码流中需要传输两个语法元素,导致码流开销大。在这种情况下,针对码流中需要传输变换核索引的情况,如果变换核组中只包括一个变换核,那么这时候不仅无需进行代价值计算,而且也不需要传输变换核索引。具体地,在当前块使用 LFNST/NSPT 时,可以将变换核组中的唯一一个变换核直接作为当前块的变换核。

在一些实施例中,在第一语法标识信息指示当前块使用第一变换模式时,确定当前块的变换核索引,可以包括:若当前块使用帧内预测模式,则解码码流,确定当前块的变换核索引;若当前块使用帧间预测模式,则根据第一语法标识信息的取值确定当前块的变换核索引。

进一步地,针对纹理特征索引通过候选列表来确定的情况,在对当前块的特征索引序号进行编码处理之前,该方法还包括:在当前块使用第一变换模式时,执行对当前块的特征索引序号进行编码处理,将所得到的编码比特写入码流的步骤。

也就是说,在本申请实施例中,第一语法标识信息用 `lfnst_nspt_idx` 表示,特征索引序号用 `lfnst_nspt_feature_idx` 表示。在这里, `lfnst_nspt_feature_idx` 可以用来选择虚拟的帧内预测模式或纹理特征索引。示例性地,如果选择候选列表的第一个元素,那么可以确定 `lfnst_nspt_feature_idx` 为0;如果选择候选列表的第二个元素,那么可以确定 `lfnst_nspt_feature_idx` 为1。另外,这里可以是在当前块使用 LFNST 或 NSPT 的情况下再确定是否需要编码 `lfnst_nspt_feature_idx`。具体地,如果当前块不使用 LFNST/NSPT,那么不需要对 `lfnst_nspt_feature_idx` 进行编码。示例性地,由于 `lfnst_nspt_idx` 用于指示当前块是否使用 LFNST/NSPT 以及 LFNST/NSPT 的变换核组中选择哪一个变换核,这时候如果 `lfnst_nspt_idx` 为0,表示当前块不使用 LFNST/NSPT,这种情况下不需要继续编码 `lfnst_nspt_feature_idx`;否则,如果 `lfnst_nspt_idx` 不为0,表示当前块使用 LFNST/NSPT,这种情况下可以继续编码 `lfnst_nspt_feature_idx`,以便后续在解码端,根据解码得到的 `lfnst_nspt_feature_idx` 即可确定出当前块的纹理特征索引。

需要注意的是,上述方法不仅可以适用于帧内预测,还可以适用于帧间预测,这里不作具体限定。

S4304: 确定当前块的残差块,并根据变换核对当前块的残差块进行变换,确定当前块的变换系数。

S4305: 对当前块的变换系数进行编码处理,将所得到的编码比特写入码流。

需要说明的是,在本申请实施例中,在对当前块的变换系数进行编码处理时,该方法可以包括:对当前块的变换系数进行量化,确定当前块的量化系数;对当前块的量化系数进行编码处理,将所得到的编码比特写入码流。

还需要说明的是,在本申请实施例中,编码端对残差块的“变换”,也可称为“正变换”,具体是指从空间域到频率域的变换,以去除残差的相关性。需要注意的是,标准如果只规定解码,那么标准文本中的“变换”是解码的部分,具体指的是本文中的“反变换”。

还需要说明的是,在本申请实施例中,参见图44,对于步骤S4304来说,该方法可以包括:

S4401: 对当前块进行帧间预测,确定当前块的预测块。

S4402: 根据当前块的原始块和当前块的预测块,确定当前块的残差块。

S4403: 根据变换核对当前块的残差块进行变换,确定当前块的变换系数。

需要说明的是,在本申请实施例中,对于步骤S4401~S4402来说,可以是与步骤S4301~S4303并行操作,也可以是在步骤S4301~S4303之前执行,或者还可以在步骤S4301~S4303之后执行,这里对步骤的先后顺序并不作具体限定。

还需要说明的是,在本申请实施例中,在确定当前块的预测块之后,可以对当前块的原始块和当前块的预测块进行减法运算,从而确定出当前块的残差块。

还需要说明的是,在本申请实施例中,在根据变换核对当前块的残差块进行变换,确定当前块的变换系数时,可以包括:若当前块的尺寸参数满足第一条条件,则根据变换核对当前块的残差块进行不可分离基础变换,确定当前块的变换系数;若当前块的尺寸参数满足第二条条件,则对当前块的残差块进行离散余弦变换,确定当前块的变换块;以及根据变换核对当前块的变换块进行低频不可分离变换,确定当前块的变换系数。

在这里,当前块的尺寸参数满足第一条条件,包括:当前块的尺寸参数较小,例如当前块的尺寸参数小于某一阈值。也就是说,针对较小尺寸的块,这里使用 NSPT 的变换核,即根据变换核对当前块的残差块进行 NSPT 变换,确定当前块的变换系数。

在这里,当前块的尺寸参数满足第二条条件,包括:当前块的尺寸参数较大,例如当前块的尺寸参数大于某一阈值。也就是说,针对较大尺寸的块,这里使用 LFNST 的变换核,即先对当前块的残差块进行 DCT2 的基础变换,然后根据变换核对当前块的变换块进行 LFNST 变换,确定当前块的变换系数。

简单来说,编码器在确定出预测块后,根据预测块导出虚拟的帧内预测模式,然后根据虚拟的帧内预测模式来确定 NSPT/LFNST 的变换核组。如果一个变换核组有多个变换核可选,编码器对变换核组中的每一个变换核进行尝试。

如果是 NSPT 变换,可以使用 NSPT 对残差块进行正变换得到变换系数,对变换系数进行量化得到量化系数,然后对量化系数熵编码。通过熵编码可以得到该变换核下的码流中的开销的代价。对量化系数进行反量化得到解码的变换系数,对解码的变换系数进行反 NSPT 变换得到解码的残差块。解码的变换系数和原始的变换系数可能是不同的,因为量化是有损的,同样地,解码的残差块和原始的残差块也可能是不同的。根据解码的残差块和预测块得到重建块。根据重建块和当前块的原始图像可以得到失真的代价。使用当前 NSPT 的变换核进行编码的代价为开销的代价加上失真的代价。比较几个变换核的代价,选最小者作为当前块的 NSPT 的最佳选择。

如果是 LFNST 变换,可以使用 DCT2 对残差块进行正变换,再用 LFNST 进行正变换得到变换系数,对变换系数进行量化得到量化系数,然后对量化系数熵编码。通过熵编码可以得到该变换核下的码流中的开销的代价。对量化系数进行反量化得到解码的变换系数,对解码的变换系数进行反 LFNST 变换,再进行反 DCT2 变换得到解码的残差块。解码的变换系数和原始的变换系数可能是不同的,因为量化是有损的,同样地,解码的残差块和原始的残差块也可能是不同的。根据解码的残差块和预测块得到重建块。根据重建块和当前块的原始图像可以得到失真的代价。使用当前 NSPT 的变换核进行编码的代价为开销的代价加上失真的代价。比较几个变换核的代价,选最小者作为当前块的 NSPT 的最佳选择。

可以理解地,本申请实施例可以使用一个高层语法(high level syntax)控制本技术方案的开关。在一些实施例中,该方法还包括:确定第二语法标识信息的取值;对第二语法标识信息的取值进行编码处理,将所得到的编码比特写入码流。

在本申请实施例中,第二语法标识信息用于指示当前序列是否允许使用第一变换模式。具体地,如果当前序列使用第一变换模式,那么可以确定第二语法标识信息的取值为第一值;如果当前序列不使用第一变换模式,那么可以确定第二语法标识信息的取值为第二值。

进一步地,在一些实施例中,该方法还包括:在当前序列允许使用第一变换模式时,执行确定当前块的纹理特征索引的步骤;其中,当前序列包括当前块。

需要说明的是,在本申请实施例中,第二语法标识信息可以用 `sps_inter_lfnst_nspt_enabled_flag` 表示,而且第二语法标识信息为序列参数集(Sequence Parameter Set, SPS)中的语法元素。

还需要说明的是,在本申请实施例中,第一值与第二值不同,而且第一值和第二值可以是参数形式,也可以是数字形式。具体地,第一语法标识信息可以是写入在概述(profile)中的参数,也可以是一个标志(flag)的取值,这里对此不作具体限定。示例性地,第一值可以为 1,第二值可以为 0;或者,第一值可以为 0,第二值可以为 1;或者,第一值可以为 true,第二值可以为 false;或者,第一值可以为 false,第二值可以为 true。在一种具体的实施例中,第一值为 1,第二值为 0。

也就是说,本申请实施例可以使用一个高层语法(high level syntax)控制本技术方案的开关。比如使用一个序列级的 flag,如在序列参数集(sequence parameter set)中增加语法元素 `sps_inter_lfnst_nspt_enabled_flag`。如果当前序列允许使用本技术方案,那么可以确定 `sps_inter_lfnst_nspt_enabled_flag` 的取值为 1;如果当前序列不允许使用本技术方案,那么可以确定 `sps_inter_lfnst_nspt_enabled_flag` 的取值为 0。其中,在允许使用本技术方案的情况下,编码端可以继续确定当前块的纹理特征索引,以及确定 LFNST 或 NSPT 的变换核索引,并且对变换核索引或者 `lfnst_nspt_idx` 的取值进行编码处理。当然本申请实施例也可以分别为 LFNST 和 NSPT 设置各自的语法元素,即 `sps_inter_lfnst_enabled_flag` 和 `sps_inter_nspt_enabled_flag`。`sps_inter_lfnst_enabled_flag` 用于指示当前序列是否允许使用 LFNST,`sps_inter_nspt_enabled_flag` 用于指示当前序列是否允许使用 NSPT。

还需要说明的是,本申请实施例也可以为帧间的 LFNST/NSPT 和帧内的 LFNST/NSPT 设置同一个 flag,如 `sps_lfnst_nspt_enabled_flag`。示例性地,如果 `sps_inter_lfnst_nspt_enabled_flag` 的取值为 0,那么可以确定当前序列允许不允许使用 LFNST 和 NSPT;如果 `sps_inter_lfnst_nspt_enabled_flag` 的取值为 1,那么可以确定当前序列允许帧内编码的块使用 LFNST 和 NSPT;如果 `sps_inter_lfnst_nspt_enabled_flag` 的取值为 2,那么可以确定当前序列允许帧间编码的块使用 LFNST 和 NSPT。

在一些实施例中,该方法还包括:确定第二语法标识信息的取值和第三语法标识信息的取值;其中,第二语法标识信息用于指示当前序列是否允许使用第一变换模式,第三语法标识信息用于指示当前图像是否允许使用第一变换模式;对第二语法标识信息的取值和第三语法标识信息的取值进行编码处理,将所得到的编码比特写入码流。

进一步地,在一些实施例中,在当前序列允许使用第一变换模式时,确定当前图像是否允许使用第一变换模式;在当前图像允许使用第一变换模式时,执行确定当前块的纹理特征索引的步骤。

在本申请实施例中,当前序列可以包括当前图像,且当前图像包括当前块。其中,第三语法标识信息可以用 `ph_inter_lfnst_nspt_enabled_flag` 表示,而且第三语法标识信息为图像级别的语法元素。

在本申请实施例中,如果当前图像允许使用第一变换模式,那么可以确定第三语法标识信息的取值为第一值;如果当前图像不允许使用第一变换模式,那么可以确定第三语法标识信息的取值为第二值。

在一些实施例中,该方法还包括:确定第二语法标识信息的取值和第四语法标识信息的取值;其中,第二语法标识信息用于指示当前序列是否允许使用第一变换模式,第四语法标识信息用于指示当前片是否允许使用第一变换模式;对第二语法标识信息的取值和第四语法标识信息的取值进行编码处理,将所得到的编码比特写入码流。

进一步地,在一些实施例中,在当前序列允许使用第一变换模式时,确定当前片是否允许使用第一变换模式;在当前片允许使用第一变换模式时,执行确定当前块的纹理特征索引的步骤。

在本申请实施例中,当前序列可以包括当前片,且当前片包括当前块。其中,第四语法标识信息可以用 `sh_inter_lfnst_nspt_enabled_flag` 表示,而且第四语法标识信息为片(slice)级别的语法元素。

在本申请实施例中,如果当前片允许使用第一变换模式,那么可以确定第四语法标识信息的取值为第一值;如果当前片不允许使用第一变换模式,那么可以确定第四语法标识信息的取值为第二值。

还需要说明的是,在本申请实施例中,第一值与第二值不同,而且第一值和第二值可以是参数形式,也可以是数字形式。具体地,第一语法标识信息可以是写入在概述(profile)中的参数,也可以是一个标志(flag)的取值,这里对此不作具体限定。示例性地,第一值可以为 1,第二值可以为 0;或者,第一值可以为 0,第二值可以为 1;或者,第一值可以为 true,第二值可以为 false;或者,第一值可以为 false,第二值可以为 true。在一种具体的实施例中,第一值为 1,第二值为 0。

也就是说,在本申请实施例中,可以使用一个高层语法(high level syntax)控制本技术方案的开关。比如使用一个序列级的 flag,如在序列参数集 SPS 中增加语法元素 `sps_inter_lfnst_nspt_enabled_flag`。如果当前序列允许使用本技术方案,那么可以确定 `sps_inter_lfnst_nspt_enabled_flag` 的取值为 1;如果当前序列不允许使用本技术方案,那么可以确定 `sps_inter_lfnst_nspt_enabled_flag` 的取值为 0。如果允许本技术方案,那么编解码端可以继续确定当前块的纹理特征索引,以及确定 LFNST 或 NSPT 的变换核索引,并且对变换核索引或者 `lfnst_nspt_idx` 的取值进行编码处理。以便解码器在解码帧间编码的块时可以解码 `lfnst_nspt_idx`,并进行 LFNST 或 NSPT 的处理。当然也可以分别为 LFNST 和 NSPT 设置各自的 flag,即 `sps_inter_lfnst_enabled_flag` 和 `sps_inter_nspt_enabled_flag`。

进一步地,本申请实施例也可以使用其他级别的语法元素来实现更灵活的控制,如图像参数集(Picture Parameter Set, PPS)的 flag,或图像头(picture header)或片头(slice header)的 flag 等。示例性地,序列参数集 SPS 确定当前序列是否可以使用本技术方案,如果当前序列是否使用本技术方案,设置一个片头(slice header)的 `sh_inter_lfnst_nspt_enabled_flag` 确定当前 slice 是否使用本技术方案,提供更高的灵活性。因为针对帧间编码,特别是 RA,不同图像或 slice 的 QP 差别较大。其中,作为 GOP 时域层级低的图像通常 QP 较低,而 GOP 时域层级越高的图像通常 QP 越高,本技术方案在 QP 特别高或者特别低的情况下对压缩效率提升不明显,因而可以设置图像头(picture header)或片头(slice header)的 flag 来更灵活地控制。

还需要说明的是,在本申请实施例中,可以将 LFNST 和 NSPT 应用于帧间预测的块。针对帧间不能很好地预测的块,即残差较大的块,相对于相关技术的 MTS 能更好地去相关性,提高压缩效率。

还需要说明的是,在本申请实施例中,除了将 LFNST 和 NSPT 应用于帧间预测的块之外,也可以应用于帧内块复制(Intra Block Copy, IBC)的块。

在本申请的又一实施例中,本申请实施例提供了一种码流,该码流是根据待编码信息进行比特编码生成的;其中,待编码信息包括下述至少一项:当前块的量化系数、当前块的特征索引序号、第一语法

标识信息的取值、第二语法标识信息的取值、第三语法标识信息的取值和第四语法标识信息的取值。

在本申请实施例中，第一语法标识信息用于指示当前块是否使用第一变换模式以及对应使用的变换核索引，第二语法标识信息用于指示当前序列是否允许使用第一变换模式，第三语法标识信息用于指示当前图像是否允许使用第一变换模式；第四语法标识信息用于指示当前片是否允许使用第一变换模式。

- 5 本实施例提供了一种编码方法，确定当前块的纹理特征索引；根据纹理特征索引，确定当前块的变换核组；根据变换核组，确定当前块的变换核；确定当前块的残差块，并根据变换核对当前块的残差块进行变换，确定当前块的变换系数；对当前块的变换系数进行编码处理，将所得到的编码比特写入码流。也就是说，这里建立了纹理特征索引与变换核组之间的对应关系，替代了相关技术中根据帧内预测模式来匹配变换核组的方案，使得 LFNST 和 NSPT 也可以应用于帧内预测模式，从而针对帧内预测模式中的
- 10 的不好预测的块，即残差较大的块，不仅可以提高压缩效率，而且还可以提升编解码性能。

在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 45，其示出了本申请实施例提供的一种编码器的组成结构示意图。如图 45 所示，该编码器 450 可以包括：第一确定单元 4501、变换单元 4502 和编码单元 4503，其中：

- 15 第一确定单元 4501，配置为确定当前块的纹理特征索引；根据纹理特征索引，确定当前块的变换核组；以及根据变换核组，确定当前块的变换核；

变换单元 4502，配置为确定当前块的残差块，并根据变换核对当前块的残差块进行变换，确定当前块的变换系数；

编码单元 4503，配置为对当前块的变换系数进行编码处理，将所得到的编码比特写入码流。

- 20 在一些实施例中，参见图 45，该编码器 450 还可以包括第一预测单元 4504，配置为对当前块进行帧内预测，确定当前块的预测块；

第一确定单元 4501，还配置为根据当前块的原始块和当前块的预测块，确定当前块的残差块。

在一些实施例中，参见图 45，该编码器 450 还可以包括量化单元 4505，配置为对当前块的变换系数进行量化，确定当前块的量化系数；

- 25 编码单元 4503，还配置为对当前块的量化系数进行编码处理，将所得到的编码比特写入码流。

在一些实施例中，第一确定单元 4501，还配置为对当前块的变换系数进行量化，确定当前块的量化系数；编码单元 4503，还配置为对当前块的量化系数进行编码处理，将所得到的编码比特写入码流。

在一些实施例中，第一确定单元 4501，还配置为确定用于推导纹理特征索引的候选像素；以及根据候选像素，确定当前块的纹理特征索引。

- 30 在一些实施例中，第一确定单元 4501，还配置为确定候选像素的水平梯度值和竖直梯度值；根据候选像素的水平梯度值和竖直梯度值，确定候选像素对应的纹理特征索引以及梯度强度值；根据候选像素对应的纹理特征索引以及梯度强度值，构建纹理特征统计表；以及根据纹理特征统计表，确定当前块的纹理特征索引。

在一些实施例中，第一确定单元 4501，还配置为根据当前块的尺寸参数，确定候选像素的数量。

- 35 在一些实施例中，第一确定单元 4501，还配置为确定当前块的预测块；以及将预测块中的至少部分像素作为候选像素。

在一些实施例中，第一确定单元 4501，还配置为确定当前块的已重建区域的相邻像素；以及将已重建区域的相邻像素作为候选像素。

- 40 在一些实施例中，第一确定单元 4501，还配置为将已重建区域的相邻像素和预测块中的至少部分像素作为候选像素。

在一些实施例中，第一确定单元 4501，还配置为确定当前块的参考块；将参考块中的至少部分像素作为候选像素。

在一些实施例中，第一确定单元 4501，还配置为确定参考块为整像素的参考块；或者，确定参考块为分像素的参考块。

- 45 在一些实施例中，第一确定单元 4501，还配置为确定当前块进行双向预测时的两个参考图像块；对两个参考图像块进行分像素的插值滤波，确定两个分像素的参考图像块；以及对两个分像素的参考图像块进行加权组合，确定当前块的参考块。

在一些实施例中，第一确定单元 4501，还配置为根据候选像素的水平梯度值和竖直梯度值进行角度映射，确定候选像素对应的纹理特征索引；以及根据候选像素的水平梯度值和竖直梯度值进行梯度强度计算，确定候选像素对应的梯度强度值。

- 50 在一些实施例中，第一确定单元 4501，还配置为根据候选像素的水平梯度值和竖直梯度值，利用预设查找表确定所述候选像素对应的纹理特征索引。

在一些实施例中, 第一确定单元 4501, 还配置为对水平梯度值的绝对值和竖直梯度值的绝对值进行加法运算, 确定候选像素对应的梯度强度值。

5 在一些实施例中, 第一确定单元 4501, 还配置为在候选像素的数量为至少一个时, 确定至少一个纹理特征索引以及对应的至少一个梯度强度值; 根据至少一个纹理特征索引确定具有互异特性的至少一种参考纹理特征索引, 以及根据所述至少一个梯度强度值, 对归属于同一参考纹理特征索引的梯度强度值进行累加计算, 确定至少一种参考纹理特征索引对应的梯度强度累加值; 以及根据至少一种参考纹理特征索引和至少一种参考纹理特征索引对应的梯度强度累加值, 构建所述纹理特征统计表。

在一些实施例中, 第一确定单元 4501, 还配置为确定纹理特征统计表中的最大梯度强度累加值; 以及将最大梯度强度累加值对应的参考纹理特征索引确定为当前块的纹理特征索引。

10 在一些实施例中, 第一确定单元 4501, 还配置为确定纹理特征统计表中的最大梯度强度累加值; 在最大梯度强度累加值小于第一阈值时, 将当前块的纹理特征索引设置为 DC 模式或 PLANANR 模式。

在一些实施例中, 第一确定单元 4501, 还配置为确定纹理特征统计表中所有梯度强度累加值的统计和值; 在统计和值小于第二阈值时, 将当前块的纹理特征索引设置为 DC 模式或 PLANANR 模式。

15 在一些实施例中, 第一确定单元 4501, 还配置为确定当前块进行几何划分模式时的角度索引; 以及根据角度索引, 确定当前块的纹理特征索引。

在一些实施例中, 第一确定单元 4501, 还配置为确定当前块的参考块; 以及根据参考块的纹理特征索引, 确定当前块的纹理特征索引。

20 在一些实施例中, 第一确定单元 4501, 还配置为构建候选列表, 其中, 候选列表包括预设数量的候选纹理特征索引; 对预设数量的候选纹理特征索引分别进行代价值计算, 确定预设数量的候选纹理特征索引各自的代价值; 以及从预设数量的候选纹理特征索引各自的代价值中确定最小代价值, 将最小代价值对应的候选纹理特征索引确定为当前块的纹理特征索引。

在一些实施例中, 第一确定单元 4501, 还配置为根据当前块的纹理特征索引, 确定当前块的特征索引序号; 其中, 特征索引序号用于指示当前块的纹理特征索引在所述候选列表中的编号; 编码单元 4503, 还配置为对当前块的特征索引序号进行编码处理, 将所得到的编码比特写入码流。

25 在一些实施例中, 第一确定单元 4501, 还配置为将纹理特征统计表中的至少一种参考纹理特征索引按照对应的梯度强度累加值由大到小进行排序, 确定排序靠前的预设数量个参考纹理特征索引; 以及根据排序靠前的预设数量个参考纹理特征索引, 构建候选列表。

30 在一些实施例中, 第一确定单元 4501, 还配置为确定纹理特征统计表中最大梯度强度累加值对应的第一纹理特征索引, 以及确定所述当前块进行几何划分模式时对应的第二纹理特征索引; 根据第一纹理特征索引和第二纹理特征索引, 构建候选列表。

在一些实施例中, 第一确定单元 4501, 还配置为确定当前块的参考块; 以及根据参考块中至少一个候选位置对应的纹理特征索引, 构建所述候选列表。

在一些实施例中, 编码单元 4503, 还配置为在当前块使用第一变换模式时, 执行对当前块的特征索引序号进行编码处理, 将所得到的编码比特写入码流的步骤。

35 在一些实施例中, 第一确定单元 4501, 还配置为确定当前块的第一语法标识信息的取值; 其中, 第一语法标识信息用于指示所述当前块是否使用第一变换模式以及对应使用的变换核索引; 编码单元 4503, 还配置为对第一语法标识信息的取值进行编码处理, 将所得到的编码比特写入码流。

40 在一些实施例中, 第一确定单元 4501, 还配置为确定当前块的变换核索引; 其中, 变换核索引用于指示当前块的变换核在所述变换核组中的编号; 编码单元 4503, 还配置为若当前块使用帧内预测模式, 则对当前块的变换核索引进行编码处理, 将所得到的编码比特写入码流; 以及若当前块使用帧间预测模式, 则根据当前块的变换核索引确定第一语法标识信息的取值, 对第一语法标识信息的取值进行编码处理, 将所得到的编码比特写入码流。

45 在一些实施例中, 变换核组包括至少一个候选变换核; 相应地, 第一确定单元 4501, 还配置为对至少一个候选变换核分别进行代价值计算, 确定至少一个候选变换核各自的代价值; 从至少一个候选变换核各自的代价值中确定最小代价值, 将最小代价值对应的候选变换核确定为所述当前块的变换核。

50 在一些实施例中, 第一确定单元 4501, 还配置为基于第一候选变换核对当前块的残差块进行变换与量化, 确定当前块的第一候选量化系数, 并对第一候选量化系数进行熵编码处理, 确定第一候选变换核的第一代价值; 对第一候选量化系数进行反量化与反变换, 确定当前块的第一候选残差块, 并根据第一候选残差块确定当前块的第一候选预测块; 根据第一候选预测块与所述当前块的原始图像进行代价计算, 确定第一候选变换核的第二代价值; 以及根据第一候选变换核的第一代价值和第二代价值, 确定第一候选变换核的代价值; 其中, 第一候选变换核为至少一个候选变换核中的任意一个。

在一些实施例中, 变换单元 4502, 还配置为若当前块的尺寸参数满足第一条件, 则根据变换核对

当前块的残差块进行不可分离基础变换, 确定当前块的变换系数; 若当前块的尺寸参数满足第二条件, 则对当前块的残差块进行离散余弦变换, 确定当前块的变换块; 以及根据变换核所述当前块的变换块进行低频不可分离变换, 确定当前块的变换系数。

5 在一些实施例中, 第一确定单元 4501, 还配置为确定第二语法标识信息的取值; 其中, 第二语法标识信息用于指示当前序列是否允许使用第一变换模式; 编码单元 4503, 还配置为对第二语法标识信息的取值进行编码处理, 将所得到的编码比特写入码流。

在一些实施例中, 第一确定单元 4501, 还配置为在当前序列允许使用第一变换模式时, 执行确定当前块的纹理特征索引的步骤; 其中, 当前序列包括所述当前块。

10 在一些实施例中, 第一确定单元 4501, 还配置为确定第二语法标识信息的取值和第三语法标识信息的取值; 其中, 第二语法标识信息用于指示当前序列是否允许使用第一变换模式, 第三语法标识信息用于指示当前图像是否允许使用第一变换模式; 编码单元 4503, 还配置为对第二语法标识信息的取值和第三语法标识信息的取值进行编码处理, 将所得到的编码比特写入码流。

15 在一些实施例中, 第一确定单元 4501, 还配置为在当前序列允许使用第一变换模式时, 确定当前图像是否允许使用第一变换模式; 以及在当前图像允许使用第一变换模式时, 执行确定当前块的纹理特征索引的步骤; 其中, 当前序列包括当前图像, 且当前图像包括所述当前块。

在一些实施例中, 第一确定单元 4501, 还配置为确定第二语法标识信息的取值和第四语法标识信息的取值; 其中, 第二语法标识信息用于指示当前序列是否允许使用第一变换模式, 第四语法标识信息用于指示当前片是否允许使用第一变换模式; 编码单元 4503, 还配置为对第二语法标识信息的取值和第四语法标识信息的取值进行编码处理, 将所得到的编码比特写入码流。

20 在一些实施例中, 第一确定单元 4501, 还配置为在当前序列允许使用第一变换模式时, 确定当前片是否允许使用第一变换模式; 以及在当前片允许使用第一变换模式时, 执行确定当前块的纹理特征索引的步骤; 其中, 当前序列包括当前片, 且当前片包括所述当前块。

25 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。

30 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。

35 因此, 本申请实施例提供了一种计算机可读存储介质, 应用于编码器 450, 该计算机可读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。

40 基于编码器 450 的组成以及计算机可读存储介质, 参见图 46, 其示出了本申请实施例提供的编码器 450 的具体硬件结构示意图。如图 46 所示, 编码器 450 可以包括: 第一通信接口 4601、第一存储器 4602 和第一处理器 4603; 各个组件通过第一总线系统 4604 耦合在一起。可理解, 第一总线系统 4604 用于实现这些组件之间的连接通信。第一总线系统 4604 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 46 中将各种总线都标为第一总线系统 4604。其中,

第一通信接口 4601, 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 4602, 用于存储能够在第一处理器 4603 上运行的计算机程序;

第一处理器 4603, 用于在运行所述计算机程序时, 执行:

45 确定当前块的纹理特征索引; 根据纹理特征索引, 确定当前块的变换核组; 根据变换核组, 确定当前块的变换核; 确定当前块的残差块, 并根据变换核对当前块的残差块进行变换, 确定当前块的变换系数; 对当前块的变换系数进行编码处理, 将所得到的编码比特写入码流。

50 可以理解, 本申请实施例中的第一存储器 4602 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, SLDRAM) 和直接内存总线随机存取存储器 (Direct Rambus RAM, DRRAM)。本申请描述的系统和方法的第一存储器 4602 旨在

- 5 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 4603 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 4603 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 4603 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
- 10 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
- 15 4602, 第一处理器 4603 读取第一存储器 4602 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。

可选地, 作为另一个实施例, 第一处理器 4603 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。

- 25 本实施例提供了一种编码器, 在该编码器中, 通过建立纹理特征索引与变换核组之间的对应关系, 替代了相关技术中根据帧内预测模式来匹配变换核组的方案, 使得 LFNST 和 NSPT 也可以应用于帧间预测模式, 从而针对帧间预测模式中的不好预测的块, 即残差较大的块, 不仅可以提高压缩效率, 而且还可以提升编解码性能。

- 30 基于前述实施例相同的发明构思, 参见图 47, 其示出了本申请实施例提供的一种解码器的组成结构示意图。如图 47 所示, 该解码器 470 可以包括第二确定单元 4701 和反变换单元 4702, 其中:

第二确定单元 4701, 配置为确定当前块的纹理特征索引; 以及根据纹理特征索引, 确定当前块的变换核组;

第二确定单元 4701, 还配置为根据变换核组, 确定当前块的变换核;

- 35 反变换单元 4702, 配置为确定当前块的变换系数, 并根据变换核对当前块的变换系数进行反变换, 确定当前块的残差块。

在一些实施例中, 参见图 47, 解码器 470 还可以包括解码单元 4703 和反量化单元 4704, 其中:

解码单元 4703, 配置为解码码流, 确定所述当前块的量化系数;

反量化单元 4704, 还配置为对当前块的量化系数进行反量化, 确定当前块的变换系数。

- 40 在一些实施例中, 参见图 47, 解码器 470 还可以包括第二预测单元 4705, 配置为对当前块进行帧间预测, 确定当前块的预测块;

第二确定单元 4701, 还配置为根据当前块的预测块和当前块的残差块, 确定当前块的重建块。

在一些实施例中, 解码单元 4703, 还配置为解码码流, 确定当前块的第一语法标识信息的取值; 第二确定单元 4701, 还配置为在第一语法标识信息指示当前块使用第一变换模式时, 确定当前块的变换核索引, 并根据变换核组和变换核索引, 确定当前块的变换核。

- 45 在一些实施例中, 解码单元 4703, 还配置为若当前块使用帧内预测模式, 则解码码流, 确定当前块的变换核索引; 第二确定单元 4701, 还配置为若当前块使用帧间预测模式, 则根据第一语法标识信息的取值确定当前块的变换核索引。

- 50 在一些实施例中, 第二确定单元 4701, 还配置为确定用于推导纹理特征索引的候选像素; 以及根据候选像素, 确定当前块的纹理特征索引。

在一些实施例中, 第二确定单元 4701, 还配置为根据候选像素对应的纹理特征索引以及梯度强度值, 构建纹理特征统计表; 以及根据纹理特征统计表, 确定当前块的纹理特征索引。

在一些实施例中,第二确定单元 4701,还配置为根据当前块的尺寸参数,确定候选像素的数量。

在一些实施例中,第二确定单元 4701,还配置为确定当前块的预测块;以及将预测块中的至少部分像素作为候选像素。

5 在一些实施例中,第二确定单元 4701,还配置为确定当前块的已重建区域的相邻像素;以及将已重建区域的相邻像素作为候选像素。

在一些实施例中,第二确定单元 4701,还配置为将已重建区域的相邻像素和预测块中的至少部分像素作为候选像素。

在一些实施例中,第二确定单元 4701,还配置为确定当前块的参考块;以及将参考块中的至少部分像素作为所述候选像素。

10 在一些实施例中,第二确定单元 4701,还配置为确定参考块为整像素的参考块;或者,确定参考块为分像素的参考块。

在一些实施例中,第二确定单元 4701,还配置为确定当前块进行双向预测时的两个参考图像块;对两个参考图像块进行分像素的插值滤波,确定两个分像素的参考图像块;以及对两个分像素的参考图像块进行加权组合,确定当前块的参考块。

15 在一些实施例中,第二确定单元 4701,还配置为根据候选像素的水平梯度值和竖直梯度值进行角度映射,确定候选像素对应的纹理特征索引;以及根据候选像素的水平梯度值和竖直梯度值进行梯度强度计算,确定候选像素对应的梯度强度值。

在一些实施例中,第二确定单元 4701,还配置为根据候选像素的水平梯度值和竖直梯度值,利用预设查找表确定候选像素对应的纹理特征索引。

20 在一些实施例中,第二确定单元 4701,还配置为对水平梯度值的绝对值和竖直梯度值的绝对值进行加法运算,确定候选像素对应的梯度强度值。

在一些实施例中,第二确定单元 4701,还配置为在候选像素的数量为至少一个时,确定至少一个纹理特征索引以及对应的至少一个梯度强度值;根据至少一个纹理特征索引确定具有互异特性的至少一种参考纹理特征索引,以及根据至少一个梯度强度值,对归属于同一参考纹理特征索引的梯度强度值进行累加计算,确定至少一种参考纹理特征索引对应的梯度强度累加值;以及根据至少一种参考纹理特征索引和至少一种参考纹理特征索引对应的梯度强度累加值,构建纹理特征统计表。

25 在一些实施例中,第二确定单元 4701,还配置为确定纹理特征统计表中的最大梯度强度累加值;以及将最大梯度强度累加值对应的参考纹理特征索引确定为当前块的纹理特征索引。

30 在一些实施例中,第二确定单元 4701,还配置为确定纹理特征统计表中的最大梯度强度累加值;在最大梯度强度累加值小于第一阈值时,将当前块的纹理特征索引设置为 DC 模式或 PLANAR 模式。

在一些实施例中,第二确定单元 4701,还配置为确定纹理特征统计表中所有梯度强度累加值的统计和值;在统计和值小于第二阈值时,将当前块的纹理特征索引设置为 DC 模式或 PLANAR 模式。

在一些实施例中,第二确定单元 4701,还配置为确定当前块进行几何划分模式时的角度索引;以及根据角度索引,确定当前块的纹理特征索引。

35 在一些实施例中,第二确定单元 4701,还配置为确定当前块的参考块;以及根据参考块的纹理特征索引,确定当前块的纹理特征索引。

在一些实施例中,第二确定单元 4701,还配置为构建候选列表,其中,候选列表包括预设数量的候选纹理特征索引;解码单元 4703,还配置为解码码流,确定当前块的特征索引序号。

40 在一些实施例中,第二确定单元 4701,还配置为将纹理特征统计表中的至少一种参考纹理特征索引按照对应的梯度强度累加值由大到小进行排序,确定排序靠前的预设数量个参考纹理特征索引;以及根据排序靠前的预设数量个参考纹理特征索引,构建所述候选列表。

在一些实施例中,第二确定单元 4701,还配置为确定纹理特征统计表中最大梯度强度累加值对应的第一纹理特征索引,以及确定当前块进行几何划分模式时对应的第二纹理特征索引;以及根据第一纹理特征索引和第二纹理特征索引,构建候选列表。

45 在一些实施例中,第二确定单元 4701,还配置为确定当前块的参考块;以及根据参考块中至少一个候选位置对应的纹理特征索引,构建候选列表。

在一些实施例中,解码单元 4703,还配置为解码码流,确定当前块的第一语法标识信息的取值;第二确定单元 4701,还配置为在第一语法标识信息指示当前块使用第一变换模式时,执行解码码流,确定当前块的特征索引序号的步骤。

50 在一些实施例中,反变换单元 4702,还配置为若当前块的尺寸参数满足第一条件,则根据变换核对当前块的变换系数进行不可分离基础变换的反变换,确定当前块的残差块;若当前块的尺寸参数满足第二条件,则根据变换核对当前块的变换系数进行低频不可分离变换的反变换,确定当前块的变换块;

以及对当前块的变换块进行离散余弦变换的反变换,确定当前块的残差块。

在一些实施例中,解码单元 4703,还配置为解码码流,确定第二语法标识信息的取值;以及在第二语法标识信息指示当前序列允许使用第一变换模式时,执行确定当前块的纹理特征索引的步骤;其中,当前序列包括当前块。

5 在一些实施例中,解码单元 4703,还配置为解码码流,确定第二语法标识信息的取值;在第二语法标识信息指示当前序列允许使用第一变换模式时,解码码流,确定第三语法标识信息的取值;以及在第三语法标识信息指示当前图像允许使用第一变换模式时,执行确定当前块的纹理特征索引的步骤;其中,当前序列包括当前图像,且当前图像包括当前块。

10 在一些实施例中,解码单元 4703,还配置为解码码流,确定第二语法标识信息的取值;在第一语法标识信息指示当前序列允许使用第一变换模式时,解码码流,确定第四语法标识信息的取值;以及在第四语法标识信息指示当前片允许使用第一变换模式时,执行确定当前块的纹理特征索引的步骤;其中,当前序列包括当前片,且当前片包括所述当前块。

可以理解地,在本实施例中,“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。

15 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,本实施例提供了一种计算机可读存储介质,应用于解码器 470,该计算机可读存储介质存储有计算机程序,所述计算机程序被第二处理器执行时实现前述实施例中任一项所述的方法。

20 基于解码器 470 的组成以及计算机可读存储介质,参见图 48,其示出了本申请实施例提供的解码器 470 的具体硬件结构示意图。如图 48 所示,解码器 470 可以包括:第二通信接口 4801、第二存储器 4802 和第二处理器 4803;各个组件通过第二总线系统 4804 耦合在一起。可理解,第二总线系统 4804 用于实现这些组件之间的连接通信。第二总线系统 4804 除包括数据总线之外,还包括电源总线、控制总线

25 总线和状态信号总线。但是为了清楚说明起见,在图 48 中将各种总线都标为第二总线系统 4804。其中,第二通信接口 4801,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 4802,用于存储能够在第二处理器 4803 上运行的计算机程序;

第二处理器 4803,用于在运行所述计算机程序时,执行:

30 确定当前块的纹理特征索引;根据纹理特征索引,确定当前块的变换核组;根据变换核组,确定当前块的变换核;确定当前块的变换系数,并根据变换核对当前块的变换系数进行反变换,确定当前块的残差块。

可选地,作为另一个实施例,第二处理器 4803 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。

35 可以理解,第二存储器 4802 与第一存储器 4602 的硬件功能类似,第二处理器 4803 与第一处理器 4603 的硬件功能类似;这里不再详述。

本实施例提供了一种解码器,在该解码器中,这里建立了纹理特征索引与变换核组之间的对应关系,替代了相关技术中根据帧内预测模式来匹配变换核组的方案,使得 LFNST 和 NSPT 也可以应用于帧间预测模式,从而针对帧间预测模式中的不好预测的块,即残差较大的块,不仅可以提高压缩效率,而且还可以提升编解码性能。

40 在本申请的再一实施例中,参见图 49,其示出了本申请实施例提供的一种编解码系统的组成结构示意图。如图 49 所示,编解码系统 490 可以包括编码器 4901 和解码器 4902。

在本申请实施例中,编码器 4901 可以是前述实施例中任一项所述的编码器,解码器 4902 可以是前述实施例中任一项所述的解码器。

45 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。

50 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。

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

本申请所提供的几个方法或设备实施例中所揭露的特征，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。

- 5 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

- 10 本申请实施例中，在编码端，确定当前块的纹理特征索引；根据纹理特征索引，确定当前块的变换核组；根据变换核组，确定当前块的变换核；确定当前块的残差块，并根据变换核对当前块的残差块进行变换，确定当前块的变换系数；对当前块的变换系数进行编码处理，将所得到的编码比特写入码流。在解码端，确定当前块的纹理特征索引；根据纹理特征索引，确定当前块的变换核组；根据变换核组，确定当前块的变换核；确定当前块的变换系数，并根据变换核对当前块的变换系数进行反变换，确定当前块的残差块。这样，无论是编码端还是解码端，都是首先确定当前块的纹理特征索引，然后根据纹理特征索引来确定当前块的变换核组，进而根据变换核组可以确定出当前块的变换核。也就是说，这里建立了纹理特征索引与变换核组之间的对应关系，替代了相关技术中根据帧内预测模式来匹配变换核组的方案，使得 LFNST 和 NSPT 也可以应用于帧间预测模式，从而针对帧间预测模式中的不好预测的块，即残差较大的块，不仅可以提高压缩效率，而且还可以提升编解码性能。
- 15

权利要求书

- 1、一种解码方法，应用于解码器，所述方法包括：
确定当前块的纹理特征索引；
根据所述纹理特征索引，确定所述当前块的变换核组；
5 根据所述变换核组，确定所述当前块的变换核；
确定所述当前块的变换系数，并根据所述变换核对所述当前块的变换系数进行反变换，确定所述当前块的残差块。
- 2、根据权利要求1所述的方法，其中，所述确定所述当前块的变换系数，包括：
解码码流，确定所述当前块的量化系数；
10 对所述当前块的量化系数进行反量化，确定所述当前块的变换系数。
- 3、根据权利要求1所述的方法，其中，所述方法还包括：
对所述当前块进行帧间预测，确定所述当前块的预测块；
根据所述当前块的预测块和所述当前块的残差块，确定所述当前块的重建块。
- 4、根据权利要求1所述的方法，其中，所述根据所述变换核组，确定所述当前块的变换核，包括：
15 解码码流，确定所述当前块的第一语法标识信息的取值；
在所述第一语法标识信息指示所述当前块使用第一变换模式时，确定所述当前块的变换核索引，并根据所述变换核组和所述变换核索引，确定所述当前块的变换核。
- 5、根据权利要求4所述的方法，其中，在所述第一语法标识信息指示所述当前块使用第一变换模式时，所述确定所述当前块的变换核索引，包括：
20 若所述当前块使用帧内预测模式，则解码码流，确定所述当前块的变换核索引；
若所述当前块使用帧间预测模式，则根据所述第一语法标识信息的取值确定所述当前块的变换核索引。
- 6、根据权利要求1所述的方法，其中，所述确定当前块的纹理特征索引，包括：
确定用于推导所述纹理特征索引的候选像素；
25 根据所述候选像素，确定所述当前块的纹理特征索引。
- 7、根据权利要求6所述的方法，其中，所述根据所述候选像素，确定所述当前块的纹理特征索引，包括：
确定所述候选像素的水平梯度值和竖直梯度值；
根据所述候选像素的水平梯度值和竖直梯度值，确定所述候选像素对应的纹理特征索引以及梯度强度值；
30 根据所述候选像素对应的纹理特征索引以及梯度强度值，构建纹理特征统计表；
根据所述纹理特征统计表，确定所述当前块的纹理特征索引。
- 8、根据权利要求6所述的方法，其中，所述方法还包括：
根据所述当前块的尺寸参数，确定所述候选像素的数量。
- 9、根据权利要求6所述的方法，其中，所述方法还包括：
35 确定所述当前块的预测块；
将所述预测块中的至少部分像素作为所述候选像素。
- 10、根据权利要求9所述的方法，其中，所述方法还包括：
确定所述当前块的已重建区域的相邻像素；
40 将所述已重建区域的相邻像素作为所述候选像素。
- 11、根据权利要求10所述的方法，其中，所述方法还包括：
将所述已重建区域的相邻像素和所述预测块中的至少部分像素作为所述候选像素。
- 12、根据权利要求6所述的方法，其中，所述方法还包括：
确定所述当前块的参考块；
45 将所述参考块中的至少部分像素作为所述候选像素。
- 13、根据权利要求12所述的方法，其中，所述确定所述当前块的参考块，包括：
确定所述参考块为整像素的参考块；或者，
确定所述参考块为分像素的参考块。
- 14、根据权利要求13所述的方法，其中，在所述参考块为分像素的参考块时，所述确定所述当前块的参考块，包括：
- 50 块的参考块，包括：

- 确定所述当前块进行双向预测时的两个参考图像块；
对所述两个参考图像块进行分像素的插值滤波，确定两个分像素的参考图像块；
对所述两个分像素的参考图像块进行加权组合，确定所述当前块的参考块。
- 15、根据权利要求 7 所述的方法，其中，所述根据所述候选像素的水平梯度值和竖直梯度值，确定
5 所述候选像素对应的纹理特征索引以及梯度强度值，包括：
根据所述候选像素的水平梯度值和竖直梯度值进行角度映射，确定所述候选像素对应的纹理特征索引；
根据所述候选像素的水平梯度值和竖直梯度值进行梯度强度计算，确定所述候选像素对应的梯度强度值。
- 16、根据权利要求 15 所述的方法，其中，所述根据所述候选像素的水平梯度值和竖直梯度值进行
10 角度映射，确定所述候选像素对应的纹理特征索引，包括：
根据所述候选像素的水平梯度值和竖直梯度值，利用预设查找表确定所述候选像素对应的纹理特征索引。
- 17、根据权利要求 15 所述的方法，其中，所述根据所述候选像素的水平梯度值和竖直梯度值进行
15 梯度强度计算，确定所述候选像素对应的梯度强度值，包括：
对所述水平梯度值的绝对值和所述竖直梯度值的绝对值进行加法运算，确定所述候选像素对应的梯度强度值。
- 18、根据权利要求 7 所述的方法，其中，所述根据所述候选像素对应的纹理特征索引以及梯度强度
20 值，构建纹理特征统计表，包括：
在所述候选像素的数量为至少一个时，确定至少一个纹理特征索引以及对应的至少一个梯度强度值；
根据所述至少一个纹理特征索引确定具有互异特性的至少一种参考纹理特征索引，以及根据所述至少一个梯度强度值，对归属于同一参考纹理特征索引的梯度强度值进行累加计算，确定所述至少一种参考纹理特征索引对应的梯度强度累加值；
根据所述至少一种参考纹理特征索引和所述至少一种参考纹理特征索引对应的梯度强度累加值，构建所述纹理特征统计表。
- 19、根据权利要求 18 所述的方法，其中，所述根据所述纹理特征统计表，确定所述当前块的纹理
25 特征索引，包括：
确定所述纹理特征统计表中的最大梯度强度累加值；
将所述最大梯度强度累加值对应的参考纹理特征索引确定为所述当前块的纹理特征索引。
- 20、根据权利要求 18 所述的方法，其中，所述根据所述纹理特征统计表，确定所述当前块的纹理
30 特征索引，包括：
确定所述纹理特征统计表中的最大梯度强度累加值；
在所述最大梯度强度累加值小于第一阈值时，将所述当前块的纹理特征索引设置为 DC 模式或 PLANANR 模式。
- 21、根据权利要求 18 所述的方法，其中，所述根据所述纹理特征统计表，确定所述当前块的纹理
35 特征索引，包括：
确定所述纹理特征统计表中所有梯度强度累加值的统计和值；
在所述统计和值小于第二阈值时，将所述当前块的纹理特征索引设置为 DC 模式或 PLANANR 模式。
- 22、根据权利要求 1 所述的方法，其中，所述确定当前块的纹理特征索引，包括：
40 确定所述当前块进行几何划分模式时的角度索引；
根据所述角度索引，确定所述当前块的纹理特征索引。
- 23、根据权利要求 1 所述的方法，其中，所述确定当前块的纹理特征索引，包括：
确定所述当前块的参考块；
45 根据所述参考块的纹理特征索引，确定所述当前块的纹理特征索引。
- 24、根据权利要求 1 所述的方法，其中，所述确定当前块的纹理特征索引，包括：
构建候选列表，其中，所述候选列表包括预设数量的候选纹理特征索引；
解码码流，确定所述当前块的特征索引序号；
根据所述候选列表和所述特征索引序号，确定所述当前块的纹理特征索引。
- 25、根据权利要求 24 所述的方法，其中，所述构建候选列表，包括：
50 将所述纹理特征统计表中的至少一种参考纹理特征索引按照对应的梯度强度累加值由大到小进行排序，确定排序靠前的预设数量个参考纹理特征索引；

- 根据所述排序靠前的预设数量个参考纹理特征索引, 构建所述候选列表。
- 26、根据权利要求 24 所述的方法, 其中, 所述构建候选列表, 包括:
确定所述纹理特征统计表中最大梯度强度累加值对应的第一纹理特征索引, 以及确定所述当前块进行几何划分模式时对应的第二纹理特征索引;
- 5 根据所述第一纹理特征索引和所述第二纹理特征索引, 构建所述候选列表。
- 27、根据权利要求 24 所述的方法, 其中, 所述构建候选列表, 包括:
确定所述当前块的参考块;
根据所述参考块中至少一个候选位置对应的纹理特征索引, 构建所述候选列表。
- 28、根据权利要求 24 所述的方法, 其中, 所述方法还包括:
- 10 解码码流, 确定所述当前块的第一语法标识信息的取值;
在所述第一语法标识信息指示所述当前块使用第一变换模式时, 执行解码码流, 确定所述当前块的特征索引序号的步骤。
- 29、根据权利要求 1 所述的方法, 其中, 所述根据所述变换核对所述当前块的变换系数进行反变换, 确定所述当前块的残差块, 包括:
- 15 若所述当前块的尺寸参数满足第一条件, 则根据所述变换核对所述当前块的变换系数进行不可分离基础变换的反变换, 确定所述当前块的残差块;
若所述当前块的尺寸参数满足第二条件, 则根据所述变换核对所述当前块的变换系数进行低频不可分离变换的反变换, 确定所述当前块的变换块; 以及对所述当前块的变换块进行离散余弦变换的反变换, 确定所述当前块的残差块。
- 20 30、根据权利要求 1 至 29 中任一项所述的方法, 其中, 所述方法还包括:
解码码流, 确定第二语法标识信息的取值;
在所述第二语法标识信息指示当前序列允许使用第一变换模式时, 执行所述确定当前块的纹理特征索引的步骤; 其中, 所述当前序列包括所述当前块。
- 31、根据权利要求 1 至 29 中任一项所述的方法, 其中, 所述方法还包括:
- 25 解码码流, 确定第二语法标识信息的取值;
在所述第二语法标识信息指示当前序列允许使用第一变换模式时, 解码码流, 确定第三语法标识信息的取值;
在所述第三语法标识信息指示当前图像允许使用第一变换模式时, 执行所述确定当前块的纹理特征索引的步骤; 其中, 所述当前序列包括当前图像, 且所述当前图像包括所述当前块。
- 30 32、根据权利要求 1 至 29 中任一项所述的方法, 其中, 所述方法还包括:
解码码流, 确定第二语法标识信息的取值;
在所述第一语法标识信息指示当前序列允许使用第一变换模式时, 解码码流, 确定第四语法标识信息的取值;
在所述第四语法标识信息指示当前片允许使用第一变换模式时, 执行所述确定当前块的纹理特征索引的步骤; 其中, 所述当前序列包括当前片, 且所述当前片包括所述当前块。
- 35 33、一种编码方法, 应用于编码器, 所述方法包括:
确定当前块的纹理特征索引;
根据所述纹理特征索引, 确定所述当前块的变换核组;
根据所述变换核组, 确定所述当前块的变换核;
- 40 确定所述当前块的残差块, 并根据所述变换核对所述当前块的残差块进行变换, 确定所述当前块的变换系数;
对所述当前块的变换系数进行编码处理, 将所得到的编码比特写入码流。
- 34、根据权利要求 33 所述的方法, 其中, 所述确定所述当前块的残差块, 包括:
对所述当前块进行帧间预测, 确定所述当前块的预测块;
- 45 根据所述当前块的原始块和所述当前块的预测块, 确定所述当前块的残差块。
- 35、根据权利要求 33 所述的方法, 其中, 所述对所述当前块的变换系数进行编码处理, 将所得到的编码比特写入码流, 包括:
对所述当前块的变换系数进行量化, 确定所述当前块的量化系数;
对所述当前块的量化系数进行编码处理, 将所得到的编码比特写入码流。
- 50 36、根据权利要求 33 所述的方法, 其中, 所述确定当前块的纹理特征索引, 包括:
确定用于推导所述纹理特征索引的候选像素;
根据所述候选像素, 确定所述当前块的纹理特征索引。

37、根据权利要求 36 所述的方法，其中，所述根据所述候选像素，确定所述当前块的纹理特征索引，包括：

确定所述候选像素的水平梯度值和竖直梯度值；

5 根据所述候选像素的水平梯度值和竖直梯度值，确定所述候选像素对应的纹理特征索引以及梯度强度值；

根据所述候选像素对应的纹理特征索引以及梯度强度值，构建纹理特征统计表；

根据所述纹理特征统计表，确定所述当前块的纹理特征索引。

38、根据权利要求 36 所述的方法，其中，所述方法还包括：

根据所述当前块的尺寸参数，确定所述候选像素的数量。

10 39、根据权利要求 36 所述的方法，其中，所述方法还包括：

确定所述当前块的预测块；

将所述预测块中的至少部分像素作为所述候选像素。

40、根据权利要求 39 所述的方法，其中，所述方法还包括：

确定所述当前块的已重建区域的相邻像素；

15 将所述已重建区域的相邻像素作为所述候选像素。

41、根据权利要求 40 所述的方法，其中，所述方法还包括：

将所述已重建区域的相邻像素和所述预测块中的至少部分像素作为所述候选像素。

42、根据权利要求 36 所述的方法，其中，所述方法还包括：

确定所述当前块的参考块；

20 将所述参考块中的至少部分像素作为所述候选像素。

43、根据权利要求 42 所述的方法，其中，所述确定所述当前块的参考块，包括：

确定所述参考块为整像素的参考块；或者，

确定所述参考块为分像素的参考块。

25 44、根据权利要求 43 所述的方法，其中，在所述参考块为分像素的参考块时，所述确定所述当前块的参考块，包括：

确定所述当前块进行双向预测时的两个参考图像块；

对所述两个参考图像块进行分像素的插值滤波，确定两个分像素的参考图像块；

对所述两个分像素的参考图像块进行加权组合，确定所述当前块的参考块。

30 45、根据权利要求 37 所述的方法，其中，所述根据所述候选像素的水平梯度值和竖直梯度值，确定所述候选像素对应的纹理特征索引以及梯度强度值，包括：

根据所述候选像素的水平梯度值和竖直梯度值进行角度映射，确定所述候选像素对应的纹理特征索引；

根据所述候选像素的水平梯度值和竖直梯度值进行梯度强度计算，确定所述候选像素对应的梯度强度值。

35 46、根据权利要求 45 所述的方法，其中，所述根据所述候选像素的水平梯度值和竖直梯度值进行角度映射，确定所述候选像素对应的纹理特征索引，包括：

根据所述候选像素的水平梯度值和竖直梯度值，利用预设查找表确定所述候选像素对应的纹理特征索引。

40 47、根据权利要求 45 所述的方法，其中，所述根据所述候选像素的水平梯度值和竖直梯度值进行梯度强度计算，确定所述候选像素对应的梯度强度值，包括：

对所述水平梯度值的绝对值和所述竖直梯度值的绝对值进行加法运算，确定所述候选像素对应的梯度强度值。

48、根据权利要求 37 所述的方法，其中，所述根据所述候选像素对应的纹理特征索引以及梯度强度值，构建纹理特征统计表，包括：

45 在所述候选像素的数量为至少一个时，确定至少一个纹理特征索引以及对应的至少一个梯度强度值；

根据所述至少一个纹理特征索引确定具有互异特性的至少一种参考纹理特征索引，以及根据所述至少一个梯度强度值，对归属于同一参考纹理特征索引的梯度强度值进行累加计算，确定所述至少一种参考纹理特征索引对应的梯度强度累加值；

50 根据所述至少一种参考纹理特征索引和所述至少一种参考纹理特征索引对应的梯度强度累加值，构建所述纹理特征统计表。

49、根据权利要求 48 所述的方法，其中，所述根据所述纹理特征统计表，确定所述当前块的纹理特征索引，包括：

确定所述纹理特征统计表中的最大梯度强度累加值;

将所述最大梯度强度累加值对应的参考纹理特征索引确定为所述当前块的纹理特征索引。

50、根据权利要求 48 所述的方法,其中,所述根据所述纹理特征统计表,确定所述当前块的纹理特征索引,包括:

5 确定所述纹理特征统计表中的最大梯度强度累加值;

在所述最大梯度强度累加值小于第一阈值时,将所述当前块的纹理特征索引设置为 DC 模式或 PLANANR 模式。

51、根据权利要求 48 所述的方法,其中,所述根据所述纹理特征统计表,确定所述当前块的纹理特征索引,包括:

10 确定所述纹理特征统计表中所有梯度强度累加值的统计和值;

在所述统计和值小于第二阈值时,将所述当前块的纹理特征索引设置为 DC 模式或 PLANANR 模式。

52、根据权利要求 33 所述的方法,其中,所述确定当前块的纹理特征索引,包括:

确定所述当前块进行几何划分模式时的角度索引;

15 根据所述角度索引,确定所述当前块的纹理特征索引。

53、根据权利要求 33 所述的方法,其中,所述确定当前块的纹理特征索引,包括:

确定所述当前块的参考块;

根据所述参考块的纹理特征索引,确定所述当前块的纹理特征索引。

54、根据权利要求 33 所述的方法,其中,所述确定当前块的纹理特征索引,包括:

20 构建候选列表,其中,所述候选列表包括预设数量的候选纹理特征索引;

对所述预设数量的候选纹理特征索引分别进行代价值计算,确定所述预设数量的候选纹理特征索引各自的代价值;

从所述预设数量的候选纹理特征索引各自的代价值中确定最小代价值,将所述最小代价值对应的候选纹理特征索引确定为所述当前块的纹理特征索引。

25 55、根据权利要求 54 所述的方法,其中,所述方法还包括:

根据所述当前块的纹理特征索引,确定所述当前块的特征索引序号;其中,所述特征索引序号用于指示所述当前块的纹理特征索引在所述候选列表中的编号;

对所述当前块的特征索引序号进行编码处理,将所得到的编码比特写入码流。

56、根据权利要求 54 所述的方法,其中,所述构建候选列表,包括:

30 将所述纹理特征统计表中的至少一种参考纹理特征索引按照对应的梯度强度累加值由大到小进行排序,确定排序靠前的预设数量个参考纹理特征索引;

根据所述排序靠前的预设数量个参考纹理特征索引,构建所述候选列表。

57、根据权利要求 54 所述的方法,其中,所述构建候选列表,包括:

35 确定所述纹理特征统计表中最大梯度强度累加值对应的第一纹理特征索引,以及确定所述当前块进行几何划分模式时对应的第二纹理特征索引;

根据所述第一纹理特征索引和所述第二纹理特征索引,构建所述候选列表。

58、根据权利要求 54 所述的方法,其中,所述构建候选列表,包括:

确定所述当前块的参考块;

根据所述参考块中至少一个候选位置对应的纹理特征索引,构建所述候选列表。

40 59、根据权利要求 55 所述的方法,其中,所述方法还包括:

在所述当前块使用第一变换模式时,执行对所述当前块的特征索引序号进行编码处理,将所得到的编码比特写入码流的步骤。

60、根据权利要求 33 所述的方法,其中,所述方法还包括:

45 确定所述当前块的第一语法标识信息的取值;其中,所述第一语法标识信息用于指示所述当前块是否使用第一变换模式以及对应使用的变换核索引;

对所述第一语法标识信息的取值进行编码处理,将所得到的编码比特写入码流。

61、根据权利要求 60 所述的方法,其中,所述方法还包括:

确定所述当前块的变换核索引;其中,所述变换核索引用于指示所述当前块的变换核在所述变换核组中的编号;

50 若所述当前块使用帧内预测模式,则对所述当前块的变换核索引进行编码处理,将所得到的编码比特写入码流;

若所述当前块使用帧间预测模式,则根据所述当前块的变换核索引确定所述第一语法标识信息的取

值, 对所述第一语法标识信息的取值进行编码处理, 将所得到的编码比特写入码流。

62、根据权利要求 33 所述的方法, 其中, 所述变换核组包括至少一个候选变换核;

所述根据所述变换核组, 确定所述当前块的变换核, 包括:

对所述至少一个候选变换核分别进行代价值计算, 确定所述至少一个候选变换核各自的代价值;

5 从所述至少一个候选变换核各自的代价值中确定最小代价值, 将所述最小代价值对应的候选变换核确定为所述当前块的变换核。

63、根据权利要求 62 所述的方法, 其中, 所述对所述至少一个候选变换核分别进行代价值计算, 确定所述至少一个候选变换核各自的代价值, 包括:

10 基于第一候选变换核对所述当前块的残差块进行变换与量化, 确定所述当前块的第一候选量化系数, 并对所述第一候选量化系数进行熵编码处理, 确定所述第一候选变换核的第一代价值;

对所述第一候选量化系数进行反量化与反变换, 确定所述当前块的第一候选残差块, 并根据所述第一候选残差块确定所述当前块的第一候选预测块; 根据所述第一候选预测块与所述当前块的原始图像进行代价计算, 确定所述第一候选变换核的第二代价值;

15 根据所述第一候选变换核的第一代价值和第二代价值, 确定所述第一候选变换核的代价值; 其中, 所述第一候选变换核为所述至少一个候选变换核中的任意一个。

64、根据权利要求 33 所述的方法, 其中, 所述根据所述变换核对所述当前块的残差块进行变换, 确定所述当前块的变换系数, 包括:

若所述当前块的尺寸参数满足第一条件, 则根据所述变换核对所述当前块的残差块进行不可分离基础变换, 确定所述当前块的变换系数;

20 若所述当前块的尺寸参数满足第二条件, 则对所述当前块的残差块进行离散余弦变换, 确定所述当前块的变换块; 以及根据所述变换核对所述当前块的变换块进行低频不可分离变换, 确定所述当前块的变换系数。

65、根据权利要求 33 至 64 中任一项所述的方法, 其中, 所述方法还包括:

25 确定第二语法标识信息的取值; 其中, 所述第二语法标识信息用于指示当前序列是否允许使用第一变换模式;

对所述第二语法标识信息的取值进行编码处理, 将所得到的编码比特写入码流。

66、根据权利要求 65 所述的方法, 其中, 所述方法还包括:

在所述当前序列允许使用第一变换模式时, 执行所述确定当前块的纹理特征索引的步骤; 其中, 所述当前序列包括所述当前块。

30 67、根据权利要求 33 至 64 中任一项所述的方法, 其中, 所述方法还包括:

确定第二语法标识信息的取值和第三语法标识信息的取值; 其中, 所述第二语法标识信息用于指示当前序列是否允许使用第一变换模式, 所述第三语法标识信息用于指示当前图像是否允许使用第一变换模式;

35 对所述第二语法标识信息的取值和所述第三语法标识信息的取值进行编码处理, 将所得到的编码比特写入码流。

68、根据权利要求 67 所述的方法, 其中, 所述方法还包括:

在当前序列允许使用第一变换模式时, 确定当前图像是否允许使用第一变换模式;

在所述当前图像允许使用第一变换模式时, 执行所述确定当前块的纹理特征索引的步骤; 其中, 所述当前序列包括当前图像, 且所述当前图像包括所述当前块。

40 69、根据权利要求 33 至 64 中任一项所述的方法, 其中, 所述方法还包括:

确定第二语法标识信息的取值和第四语法标识信息的取值; 其中, 所述第二语法标识信息用于指示当前序列是否允许使用第一变换模式, 所述第四语法标识信息用于指示当前片是否允许使用第一变换模式;

45 对所述第二语法标识信息的取值和所述第四语法标识信息的取值进行编码处理, 将所得到的编码比特写入码流。

70、根据权利要求 69 所述的方法, 其中, 所述方法还包括:

在当前序列允许使用第一变换模式时, 确定当前片是否允许使用第一变换模式;

在所述当前片允许使用第一变换模式时, 执行所述确定当前块的纹理特征索引的步骤; 其中, 所述当前序列包括当前片, 且所述当前片包括所述当前块。

50 71、一种码流, 其中, 所述码流是根据待编码信息进行比特编码生成的; 其中, 待编码信息包括下述至少一项:

当前块的量化系数、所述当前块的特征索引序号、第一语法标识信息的取值、第二语法标识信息的

取值、第三语法标识信息的取值和第四语法标识信息的取值；

其中，所述第一语法标识信息用于指示所述当前块是否使用第一变换模式以及对应使用的变换核索引，所述第二语法标识信息用于指示当前序列是否允许使用第一变换模式，所述第三语法标识信息用于指示当前图像是否允许使用第一变换模式；所述第四语法标识信息用于指示当前片是否允许使用第一变换模式。

5

72、一种编码器，所述编码器包括第一确定单元、变换单元和编码单元，其中：

所述第一确定单元，配置为确定当前块的纹理特征索引；根据所述纹理特征索引，确定所述当前块的变换核组；以及根据所述变换核组，确定所述当前块的变换核；

10

所述变换单元，配置为确定所述当前块的残差块，并根据所述变换核对所述当前块的残差块进行变换，确定所述当前块的变换系数；

所述编码单元，配置为对所述当前块的变换系数进行编码处理，将所得到的编码比特写入码流。

73、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；

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 33 至 70 中任一项所述的方法。

15

74、一种解码器，所述解码器包括第二确定单元和反变换单元，其中：

所述第二确定单元，配置为确定当前块的纹理特征索引；以及根据所述纹理特征索引，确定所述当前块的变换核组；

所述第二确定单元，还配置为根据所述变换核组，确定所述当前块的变换核；

20

所述反变换单元，配置为确定所述当前块的变换系数，并根据所述变换核对所述当前块的变换系数进行反变换，确定所述当前块的残差块。

75、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；

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 32 中任一项所述的方法。

25

76、一种计算机可读存储介质，其中，所述计算机可读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如权利要求 1 至 32 中任一项所述的方法、或者被第二处理器执行时实现如权利要求 33 至 70 中任一项所述的方法。