

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。

在视频编码标准中包括帧内预测和帧间预测。其中，帧内预测是根据当前块周边已重建的图像来预测当前块，帧间预测是根据相同物体在不同时间的图像来预测当前块，两者残差的统计规律是有区别的，导致在变换过程中虽然可以使用低频不可分离变换（Low Frequency Non-Separable Transform, LFNST）或不可分离的基础变换（Non-Separable Primary Transform, NSPT），但是压缩效率较低。

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：

确定当前块的预测模式；

根据当前块的预测模式，确定至少一个上下文模型；

根据至少一个上下文模型解码码流，确定当前块的二元符号串；其中，二元符号串包括至少一个二元符号；

20 对二元符号串进行反二值化处理，确定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，确定当前块的变换核，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：

确定当前块的预测模式；

25 根据当前块的预测模式，确定至少一个上下文模型；

确定第一语法标识信息的取值，并对第一语法标识信息的取值进行二值化处理，确定当前块的二元符号串；其中，二元符号串包括至少一个二元符号；

根据至少一个上下文模型对二元符号串中的至少一个二元符号分别进行编码处理，将所得到的编码比特写入码流。

30 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、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，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，第二语法标识信息用于当前块是否使用多变换选择模式，第三语法标识信息用于指示当前块是否使用不可分离基础变换模式，第四语法标识信息用于指示当前块是否使用低频不可分离变换模式。

35 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、二值化单元和编码单元，其中：第一确定单元，配置为确定当前块的预测模式；以及根据当前块的预测模式，确定至少一个上下文模型；

二值化单元，配置为确定第一语法标识信息的取值，并对第一语法标识信息的取值进行二值化处理，

40 确定当前块的二元符号串；其中，二元符号串包括至少一个二元符号；

编码单元，配置为根据至少一个上下文模型对二元符号串中的至少一个二元符号分别进行编码处理，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；

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、解码单元、反二值化单元和反变换单元,其中:

5 第二确定单元,配置为确定当前块的预测模式;以及根据当前块的预测模式,确定至少一个上下文模型;

解码单元,配置为根据至少一个上下文模型解码码流,确定当前块的二元符号串;其中,二元符号串包括至少一个二元符号;

反二值化单元,配置为对二元符号串进行反二值化处理,确定第一语法标识信息的取值;

10 反变换单元,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,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;

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。

15 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,确定当前块的预测模式;根据当前块的预测模式,确定至少一个上下文模型;确定第一语法标识信息的取值,并对第一语法标识信息的取值进行二值化处理,确定当前块的二元符号串;其中,二元符号串包括至少一个二元符号;根据至少一个上下文模型对二元符号串中的至少一个二元符号分别进行编码处理,将所得到的编码比特写入码流。在解码端,确定当前块的预测模式;根据当前块的预测模式,确定至少一个上下文模型;根据至少一个上下文模型解码码流,确定当前块的二元符号串;其中,二元符号串包括至少一个二元符号;对二元符号串进行反二值化处理,确定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,确定当前块的变换核,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,都是首先根据当前块的预测模式来确定至少一个上下文模型,然后根据所确定的至少一个上下文模型对第一语法标识信息对应的二元符号串进行编解码处理。也就是说,针对帧内预测模式或者帧间预测模式可以选择不同的上下文模型,使得能够对帧间预测模式下 LFNST/NSPT 的二元符号串使用符合其分布规律的编解码方法,从而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

- 30 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;
35 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;
40 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;
45 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;
50 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;
5 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;
10 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;
15 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 为本申请实施例提供的一种帧内预测模式对应的 $lfnst_nspt_idx$ 的统计结果示意图;
20 图 41 为本申请实施例提供的一种帧间预测模式对应的 $lfnst_nspt_idx$ 的统计结果示意图;
图 42 为本申请实施例提供的一种单一纹理特征与双向纹理特征的示意图;
图 43 为本申请实施例提供的一种解码方法的流程示意图二;
图 44 为本申请实施例提供的一种解码方法的流程示意图三;
图 45 为本申请实施例提供的一种解码方法的流程示意图四;
25 图 46 为本申请实施例提供的一种编码方法的流程示意图一;
图 47 为本申请实施例提供的一种编码方法的流程示意图二;
图 48 为本申请实施例提供的一种语法元素值写入码流的框图示意图;
图 49 为本申请实施例提供的一种从码流中解码语法元素值的框图示意图;
图 50 为本申请实施例提供的一种编码器的组成结构示意图;
30 图 51 为本申请实施例提供的一种编码器的硬件结构示意图;
图 52 为本申请实施例提供的一种解码器的组成结构示意图;
图 53 为本申请实施例提供的一种解码器的硬件结构示意图;
图 54 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

- 35 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。

- 40 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。

- 45 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:

- 50 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);
5 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);
10 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);
15 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);
20 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
25 (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
30 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
35 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
40 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
45 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
50 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, 一般可以理解为一帧是一
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。为了使眼看不出卡顿，一般的视频的帧率会有 30 帧每秒，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 IBBB.....，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,并非是具体的块划分。

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: $tmvp_x = col_mv_x * tb / td$, $tmvp_y = col_mv_y * tb / td$ 。

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。

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。

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。

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。

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 x[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

其中，MMVD 的 MVD 按如下方法得到：

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

5 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:

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

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, 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做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
5 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
10 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
15 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
20 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
25 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
30 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。
35 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
40 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。

45 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
50 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:

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

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

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

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，然后第二三分量再编解码，这样第二三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。

（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 个组，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 (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 $\text{pred}_c(i,j) = \alpha \cdot \text{grad}_L(i,j) + \beta$;

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

10 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。

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。

20 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。

30 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。

40 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	$T_i(j) = \omega_0 \cdot \sqrt{\frac{2}{N}} \cdot \cos\left(\frac{\pi \cdot i \cdot (2j+1)}{2N}\right)$ $\text{where, } \omega_0 = \begin{cases} \sqrt{\frac{2}{N}} & i = 0 \\ 1 & i \neq 0 \end{cases}$

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。

5 (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。

10 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。

15 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。

20 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。

25 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。

30 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。

35 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。

40 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。

5 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

10 （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。

15 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。

20 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。

25 综上所述，LFNST 和 NSPT 可以应用于帧内编码的块，或者也可以应用于帧间编码的块。虽然对变换而言，这里提供了残差块、变换系数，只需进行变换与反变换就是。但实际上当前块可能会使用 LFNST/NSPT，也可能不会使用 LFNST/NSPT，而且 LFNST/NSPT 的一个变换核组中是有多个变换核可以选择的。也就是说，编码器是可以选择使用或不使用 LFNST/NSPT，具体使用 LFNST/NSPT 的哪一个变换核。如果不使用 LFNST/NSPT，那么可以选择其他的变换方法，例如从 DCT2、DCT8、DST7 30 中选择水平和竖直的基础变换。

另外，帧内预测和帧间预测产生的残差在统计上是差别很大的。帧内预测可以根据当前块周边已重建的图像来预测当前块，如果当前块的原始图像有周边已重建的图像没有的变化，帧内预测很难得到很好的预测块，这也意味着残差比较多。而帧间预测是用相同物体在不同时间的图像来预测当前块，如果当前块中的物体在参考图像和当前图像的时间间隔中没有发生明显的变化、或者遮挡等，帧间预测能得到很好的预测块，这也意味着残差通常比较少。当然帧间预测也会出现预测不好的情况，例如物体发生了明显的变化、以及存在遮挡等，从根本上来说从其他已编码的图像中找不到很好的这个内容，这样残差也会比较明显。简单来说，因为帧内预测和帧间预测存在本质的区别，它们的残差的统计规律是不同的；这也就导致虽然 LFNST/NSPT 都可以用于帧内预测和帧间预测，但仍是有所区别的。

40 本申请实施例提供了一种编码方法，确定当前块的预测模式；根据当前块的预测模式，确定至少一个上下文模型；确定第一语法标识信息的取值，并对第一语法标识信息的取值进行二值化处理，确定当前块的二元符号串；其中，二元符号串包括至少一个二元符号；根据至少一个上下文模型对二元符号串中的至少一个二元符号分别进行编码处理，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，确定当前块的预测模式；根据当前块的预测模式，确定至少一个上下文模型；根据至少一个

上下文模型解码码流,确定当前块的二元符号串;其中,二元符号串包括至少一个二元符号;对二元符号串进行反二值化处理,确定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,确定当前块的变换核,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。

- 5 这样,无论是编码端还是解码端,都是首先根据当前块的预测模式来确定至少一个上下文模型,然后根据所确定的至少一个上下文模型对第一语法标识信息对应的二元符号串进行编解码处理。也就是说,针对帧内预测模式或者帧间预测模式可以选择不同的上下文模型,使得能够对帧间预测模式下 LFNST/NSPT 的二元符号串使用符合其分布规律的编解码方法,从而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。

- 10 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。

15 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 的输出可以是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。

30 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。

50 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 还可以是具有一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：确定当前块的预测模式。

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可以应用于帧内预测模式和/或帧间预测模式，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，以提高压缩效率。

还需要说明的是，在本申请实施例中，当前块的预测模式可以包括：帧间预测模式和/或帧内预测模式。其中，帧内预测模式主要是根据当前块周边的已重建区域来预测当前块，帧间预测模式主要是根据当前块的参考图像来预测当前块。

S3902：根据当前块的预测模式，确定至少一个上下文模型。

需要说明的是,在本申请实施例中,确定帧间预测模式对应的至少一个上下文模型与帧内预测模式对应的至少一个上下文模型不同。也就是说,帧间预测模式与帧内预测模式不共用上下文模型。

S3903: 根据至少一个上下文模型解码码流,确定当前块的二元符号串;其中,二元符号串包括至少一个二元符号。

5 S3904: 对二元符号串进行反二值化处理,确定第一语法标识信息的取值。

还需要说明的是,在本申请实施例中,上下文模型的数量与二元符号的数量有关。如果当前块的二元符号串包括n个二元符号,那么这里根据当前块的预测模式,需要确定n个上下文模型。其中,每一个二元符号在解码时使用一个上下文模型。在这里,n为正整数。

10 在一种可能的实现方式中,针对帧内预测模式和帧间预测模式,二元符号串可以使用相同的反二值化方式,但是不同的上下文模型。

在一些实施例中,根据至少一个上下文模型解码码流,确定当前块的二元符号串,可以包括:在当前块的预测模式为帧内预测模式时,根据至少一个第一上下文模型解码码流,确定当前块的第一二元符号串;或者,在当前块的预测模式为帧间预测模式时,根据至少一个第二上下文模型解码码流,确定当前块的第二二元符号串。

15 在本申请实施例中,至少一个第一上下文模型与至少一个第二上下文模型不同,即帧间预测模式与帧内预测模式不共用上下文模型。

在一些实施例中,对二元符号串进行反二值化处理,确定第一语法标识信息的取值,可以包括:在当前块的预测模式为帧内预测模式和/或帧间预测模式时,基于第一方式对二元符号串进行反二值化处理,确定第一语法标识信息的取值。

20 在本申请实施例中,无论是帧间预测模式还是帧内预测模式均可以使用第一方式进行解码处理。其中,第一方式可以为定长解码方式,也可称为固定长度(Fixed-Length, FL)解码方式,即采用相同数目的二元符号表示第一语法标识信息的不同取值;或者,第一方式也可以为截断一元码方式,即采用不同数目的二元符号表示第一语法标识信息的不同取值;或者,第一方式甚至还可以为其他方式,如一元码方式、K阶指数哥伦布解码方式等,这里不作具体限定。

25 在本申请实施例中,第一语法标识信息可以用lfnst_nspt_idx表示。示例性地,lfnst_nspt_idx的取值可以为0、1、2、3。如果第一方式为定长解码方式,可以使用表6所示的二值化表格。在表6中,第一语法标识信息的取值都是使用2个二元符号,这2个二元符号可以都是使用上下文模型进行解码。其中,第一个(binIdx等于0)二元符号使用一个上下文模型,第二个(binIdx等于1)二元符号使用一个上下文模型。

30 表6

lfnst_nspt_idx 的取值	BinIdx	
	0	1
0	0	0
1	1	0
2	0	1
3	1	1

如果第一方式为截断一元码方式,可以使用表7所示的二值化表格。在表7中,第一语法标识信息的取值可以使用不同的二元符号串长度。其中,表7中的3个二元符号可以都是上下文模型进行解码。其中,第一个(binIdx等于0)二元符号使用一个上下文模型,第二个(binIdx等于1)二元符号使用一个上下文模型,第三个(binIdx等于2)二元符号使用一个上下文模型。

35 表7

lfnst_nspt_idx 的取值	BinIdx		
	0	1	2
0	0		
1	1	0	
2	1	1	0
3	1	1	1

需要说明的是,在本申请实施例中,一元码的解码规则是,对于待解码的二元符号“x”>=0,那么解码的二元符号串可以由x个“1”再加一个“0”组成。例如,若x=5,则根据一元码方式进行解码后,二元符号串为“111110”。

40 还需要说明的是,在本申请实施例中,截断一元码属于一元码的变体,用在已知待解码的语法元素的最大值Max的情况下。假设待解码符号为x,如果0<x<Max,那么x对应的二元符号串采用一元

码方式；如果 $x = \text{Max}$ ，那么 x 对应的二元符号串全部由 1 组成，长度为 Max 。例如，假设 $\text{Max} = 6$ ，若 $x = 6$ ，则根据截断一元码方式进行解码后，二元符号串为“111111”；若 $x = 3$ ，则根据截断一元码方式进行解码后，二元符号串为“1110”。

可以理解地，在本申请实施例中，NSPT 和 LFNST 都是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，垂直渐变纹理，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，在本申请实施例中，NSPT 和 LFNST 都包括有多个变换核组，每一个变换核组包括 3 个变换核可选，再加上不使用 LFNST/NSPT 的情况，也就是一个块针对 LFNST/NSPT 有 4 种可能的选择。在申请实施例中可以使用语法元素 `lfnst_idx` 表示是否使用 LFNST 以及使用哪一个 LFNST 变换核。`lfnst_idx` 等于 0，表示当前块不使用 LFNST；`lfnst_idx` 等于 1，表示当前块使用 LFNST 的第一个变换核；`lfnst_idx` 等于 2，表示当前块使用 LFNST 的第二个变换核。需要注意的是，变换核组可以由帧内预测模式或纹理特征索引确定，此时不需要语法元素指示。另外，对于一个变换核组，在一些技术中只能选择 2 个变换核，在另一些技术中可以选择 3 个变换核。而且对一定尺寸的小块则使用 NSPT，也可以是 3 个变换核。在本申请实施例中，这里可以使用 `lfnst_nspt_idx` 来指示是否使用 LFNST/NSPT 以及使用哪一个 LFNST/NSPT 的变换核。因为在一种可能的实现方式中它对较小的块指示的是 NSPT 的变换核索引，对较大的块指示的是 LFNST 的变换核索引。也就是说，`lfnst_nspt_idx` 的值可以为 0, 1, 2, 3。当然，本申请实施例也可以使用 `lfnst_idx` 这个语法元素，或者还可以使用 `lfnst_idx` 和 `nspt_idx` 两个语法元素，`lfnst_idx` 用于指示是否使用 LFNST 以及使用哪一个 LFNST 的变换核，`nspt_idx` 用于指示是否使用 NSPT 以及使用哪一个 NSPT 的变换核；这里实质上是一样的，在此不作具体限定。

示例性地，对于帧内预测模式下 `lfnst_nspt_idx` 的取值，其二值化方式可以如前述表 6 所示。可以看到帧内预测模式的 `lfnst_nspt_idx` 的二值化方式都使用 2 个二元符号，这 2 个二元符号都是上下文模型进行解码的。第一个（`binIdx` 等于 0）二元符号使用一个上下文模型，第二个（`binIdx` 等于 1）二元符号使用一个上下文模型。

需要说明的是，这样设置的二值化方式及上下文模型配置和它的统计规律是相关的。如图 40 所示，这里示出了一些技术中帧内预测模式对应的 `lfnst_nspt_idx` 的统计结果示意图。具体地，图 40 是一种帧内 `lfnst_nspt_idx` 的取值在测试序列 MarketPlace 在 all intra QP 32 下的统计结果。根据图 40 可以看到这个条件下，对于 `lfnst_nspt_idx` 的取值，4 种选择的可能性是差不多的，1 和 2 的可能性甚至比 0 还要多。需要注意的是，这仅是一个示例，不同的测试序列、不同的 QP 对应的统计结果是不同的。

进一步地，使用同样的二值化方式，`lfnst_nspt_idx` 统计出来的一些技术中帧间预测模式对应的 `lfnst_nspt_idx` 的统计结果如图 41 所示。具体地，图 41 是一种帧间 `lfnst_nspt_idx` 在测试序列 BasketballDrive 在 random access QP 32 下的统计结果。根据图 41 可以看到这个条件下，对于 `lfnst_nspt_idx` 的取值，4 种选择的可能性有一定差距。0 明显高于 1、2、3，1 高于 2、3。当然也需要注意的是，这仅是一个示例，不同的测试序列、不同的 QP 对应的统计结果是不同的。

基于对帧间 `lfnst_nspt_idx` 的概率分布的理论分析和统计数据，本申请实施例提出了一种帧间 `lfnst_nspt_idx` 的二值化方式，具体如前述表 7 所示。根据表 7 可以看到帧间 `lfnst_nspt_idx` 的二值化方式使用不同的二元符号串长度，是一种截断一元码方式。在这里，3 个二元符号可以都是上下文模型进行解码。第一个（`binIdx` 等于 0）二元符号使用一个上下文模型，第二个（`binIdx` 等于 1）二元符号使用一个上下文模型，第三个（`binIdx` 等于 2）二元符号使用一个上下文模型。

也就是说，在另一种可能的实现方式中，针对帧内预测模式和帧间预测模式，二元符号串可以使用不同的反二值化方式，以及使用不同的上下文模型。

在一些实施例中，根据至少一个上下文模型解码码流，确定当前块的二元符号串，可以包括：在当前块的预测模式为帧内预测模式时，根据至少一个第一上下文模型和第一方式解码码流，确定当前块的第一二元符号串；或者，在当前块的预测模式为帧间预测模式时，根据至少一个第二上下文模型和第二方式解码码流，确定当前块的第二二元符号串。

在一些实施例中，对二元符号串进行反二值化处理，确定第一语法标识信息的取值，可以包括：在当前块的预测模式为帧内预测模式时，基于第一方式对二元符号串进行反二值化处理，确定第一语法标识信息的取值；或者，在当前块的预测模式为帧间预测模式时，基于第二方式对二元符号串进行反二值化处理，确定第一语法标识信息的取值。

在本申请实施例中，至少一个第一上下文模型与至少一个第二上下文模型不同，即帧间预测模式与帧内预测模式不共用上下文模型。另外，第一方式与第二方式不同，即帧间预测模式与帧内预测模式使用不同的二值化方式。

在一种具体的实施例中，第一方式可以为定长解码方式，第二方式可以为截断一元码方式。如此，在当前块的预测模式为帧内预测模式时，根据至少一个第一上下文模型和定长解码方式解码码流，确定当前块的第一二元符号串；或者，在当前块的预测模式为帧间预测模式时，根据至少一个第二上下文模型和截断一元码方式解码码流，确定当前块的第二二元符号串。在这种实现方式下，对于第一语法标识信息 `lfnst_nspt_idx` 的不同取值，第一二元符号串的长度不变，第二二元符号串的长度则是变化的。

简单来说，针对帧内预测模式和帧间预测模式的变换核索引（即 `lfnst_nspt_idx` 的取值），这里可以使用不同的解码方法，而且也可以使用不同的上下文模型。具体地，帧间预测模式的 `lfnst_nspt_idx` 使用的上下文模型和帧内预测模式的 `lfnst_nspt_idx` 使用的上下文模型不同，或者说帧间预测模式的 `lfnst_nspt_idx` 和帧内预测模式的 `lfnst_nspt_idx` 不共用上下文模型。其中，一种可能的实现方式是帧间预测模式的 `lfnst_nspt_idx` 和帧内预测模式的 `lfnst_nspt_idx` 使用相同的二值化方式，但是不同的上下文模型。另一种可能的实现方式是帧间预测模式的 `lfnst_nspt_idx` 和帧内预测模式的 `lfnst_nspt_idx` 使用不同的二值化方式，以及不同的上下文模型。

还需要说明的是，在本申请实施例中，LFNST/NSPT 可以共用一个语法元素 `lfnst_nspt_idx`，具体是对较小的块指示的是 NSPT 的变换核索引，对较大的块指示的是 LFNST 的变换核索引。或者，LFNST/NSPT 也可能各自使用不同的语法元素，例如 LFNST 使用语法元素 `lfnst_idx`，NSPT 使用语法元素 `nspt_idx`，但是针对 `lfnst_idx` 和 `nspt_idx` 的二值化处理，上述方法仍然适用。

S3905：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，并根据变换核对待当前块的变换系数进行反变换，确定当前块的残差块。

需要说明的是，在本申请实施例中，确定当前块的变换核，可以包括：确定当前块的变换核组；根据变换核组与第一语法标识信息的取值，确定当前块的变换核。

在这里，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用于指示当前块的变换核在变换核组的编号。这样，根据变换核组与第一语法标识信息的取值，可以确定出当前块的变换核。

还需要说明的是，在本申请实施例中，对于当前块进行帧内预测时，可以根据帧内预测模式与变换核组之间的对应关系，确定出当前块的变换核组。对于当前块进行帧间预测时，可以根据纹理特征索引（即虚拟的帧内预测模式）与变换核组之间的对应关系，确定出当前块的变换核组。其中，可以使用当前块的预测块或参考块或当前块周边的重建区域来推导当前块的纹理特征索引。

在一些实施例中，确定当前块的变换核组，可以包括：在当前块的预测模式为帧内预测模式时，确定变换核组中包括 M 个候选变换核；或者，在当前块的预测模式为帧间预测模式时，确定变换核组中包括 N 个候选变换核。其中，M、N 均为正整数，且 M 大于或等于 N。

也就是说，在本申请实施例中，LFNST/NSPT 可能有更多或者更少的选择，帧间预测的 LFNST/NSPT 可用的变换核数量可能和帧内预测的 LFNST/NSPT 可用的变换核数量可以不同。示例性地，由于帧内预测的残差更多，帧间预测的残差更少，帧内预测的一个变换核组有 3 个变换核，而帧间预测的一个变换核组有 2 个变换核。在这种情况下，针对帧间预测的 `lfnst_nspt_idx` 的取值，可以为 0，1，2。

示例性地，表 8 提供了另一种帧间预测的 `lfnst_nspt_idx` 的二值化方式，如下所示。

表 8

lfnst_nspt_idx 的取值	BinIdx	
	0	1
0	0	
1	1	0
2	1	1

根据表 8 可以看到，帧间预测的 `lfnst_nspt_idx` 的二值化方式可以使用不同的二元符号串长度，是一种截断一元码方式。这 2 个二元符号可以都是上下文模型进行解码的。第一个（`binIdx` 等于 0）二元符号使用一个上下文模型，第二个（`binIdx` 等于 1）二元符号使用一个上下文模型。

还需要说明的是，帧间预测的 `lfnst_nspt_idx` 使用的上下文模型和帧内预测的 `lfnst_nspt_idx` 使用的上下文模型不同，或者说帧间预测的 `lfnst_nspt_idx` 和帧内预测的 `lfnst_nspt_idx` 不共用上下文模型。另外，LFNST/NSPT 也可能各自使用不同的语法元素，上述方法仍然适用。

在一些实施例中，该方法还包括：在 M 大于 N 时，将 N 个候选变换核设置为 M 个候选变换核的一个子集；或者，在 M 等于 N 时，将 N 个候选变换核设置为与 M 个候选变换核中至少部分不同。

需要说明的是，在本申请实施例中，帧间 LFNST/NSPT 可用的变换核可以和帧内 LFNST/NSPT 可

用的变换核相同。如果帧间 LFNST/NSPT 可用的变换核数量小于帧内 LFNST/NSPT 可用的变换核数量，那么帧内 LFNST/NSPT 可用的变换核可以是帧内 LFNST/NSPT 可用的变换核的一个子集。示例性地，帧间预测可以使用 2 个变换核，帧内预测可以使用 3 个变换核，帧间预测可用的变换核可以是帧内预测可用的变换核的前 2 个或者后 2 个或者第 1、3 个，这里不作具体限定。

5 还需要说明的是，在本申请实施例中，帧间 LFNST/NSPT 可用的变换核可以和帧内 LFNST/NSPT 可用的变换核不同。即针对帧间预测和帧内预测的残差特征分别设计变换核，这样能达到更高的压缩效率。示例性地，帧间预测可以使用 3 个变换核，帧内预测也可以使用 3 个变换核，但是帧间预测和帧内预测的 3 个变换核部分或全部不同。

也就是说，在本申请实施例中，帧间预测使用的变换核和帧内预测使用的变换核可以相同，也可以部分或全部不同。另外，在本申请实施例中，所述可用的变换核是指一个变换核组可用的变换核。

10 在一些实施例中，在确定当前块的变换核组时，该方法还可以包括：确定当前块的至少一个纹理特征；根据至少一个纹理特征进行变换核训练，确定当前块的变换核组。

需要说明的是，相关技术中帧内预测可以根据帧内预测模式对应一个变换核组，在本申请实施例中，帧间预测可以用预测块导出虚拟的帧内预测模式（纹理特征索引）对应一个变换核组。帧间预测也可以导出和帧内预测不同的纹理特征，对应地以这样的纹理特征来训练变换核。

15 示例性地，目前 LFNST 和 NSPT 所训练的都是比较单一的纹理特征，比如说某一个特定方向的纹理，如图 42 中的（a）和（b）所示。应注意的是，这只是一个示意图，但是对帧间预测来说，残差更多的出现在物体的边缘，比如说如果帧间会经常出现如图 42 中的（c）所示的有两个方向纹理的残差，那么也可以训练这样的 LFNST/NSPT 的变换核，在分析当前块的纹理特征（如上述推导虚拟的帧内预测模式的过程）时，如果发现当前块的预测块或者周边重建区域有 2 种方向的纹理，那么就可以匹配具有这样纹理特征的变换核。

下面将结合第二标识信息、第三语法标识信息和第四语法标识信息针对语法元素的解码顺序进行详细介绍。

在一种可能的实现方式中，针对帧间预测模式，参见图 43，该方法可以包括：

25 S4301：解码码流，确定当前块的第二语法标识信息的取值。

S4302：在第二语法标识信息指示当前块不使用多变换选择模式且当前块满足第一变换模式的使用条件时，解码码流，确定当前块的第一语法标识信息的取值。

S4303：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，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。

30 需要说明的是，在本申请实施例中，对于第一语法标识信息和第二语法标识信息都可以使用上下文模型进行解码。具体地，针对步骤 S4302 来说，解码码流，确定当前块的第一语法标识信息的取值，可以包括：根据至少一个上下文模型解码码流，确定第一语法标识信息的二元符号串；以及对第一语法标识信息的二元符号串进行反二值化处理，确定第一语法标识信息的取值。针对步骤 S4301 来说，解码码流，确定当前块的第二语法标识信息的取值，可以包括：根据至少一个上下文模型解码码流，确定第二语法标识信息的二元符号串；以及对第二语法标识信息的二元符号串进行反二值化处理，确定第二语法标识信息的取值。在这里，二元符号串中的每个二元符号使用一个上下文模型。

35 还需要说明的是，在本申请实施例中，第一语法标识信息用 `lfnst_nspt_idx` 表示，第二语法标识信息用 `mts_idx` 表示。其中，`lfnst_nspt_idx` 用于指示当前块是否使用第一变换模式（即 LFNST/NSPT），`mts_idx` 用于指示当前块是否使用多变换选择模式（即 MTS）。也就是说，在解码码流时，LFNST/NSPT 的语法元素 `lfnst_nspt_idx` 可以在 MTS 的语法元素 `mts_idx` 之后。

40 在一些实施例中，如果第一语法标识信息的取值为第一值，那么可以确定当前块不使用 LFNST/NSPT；如果第一语法标识信息的取值为第二值，那么可以确定当前块使用 LFNST/NSPT 以及对应使用的哪种变换核。其中，第一值可以设置为 0，第二值可以设置为非 0，例如 1、2、3 等。

45 在一些实施例中，如果第二语法标识信息的取值为第一值，那么可以确定当前块不使用多变换选择模式；如果第二语法标识信息的取值为第二值，那么可以确定当前块使用多变换选择模式以及对应使用的哪种变换核。其中，第一值可以设置为 0，第二值可以设置为非 0，例如 1、2、3 等。

50 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指示当前块使用多变换选择模式时，那么不再需要解码 `lfnst_nspt_idx`，此时可以确定当前块的多个变换核，并根据多个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。例如，`mts_idx` 用于指示当前块的水平/竖直方向分别使用哪种（基础）变换核。

在另一种可能的实现方式中，针对帧内预测模式，参见图 44，该方法可以包括：

S4401：解码码流，确定当前块的第一语法标识信息的取值。

S4402: 在第一语法标识信息指示当前块不使用第一变换模式且当前块满足多变换选择模式的使用条件时, 解码码流, 确定当前块的第二语法标识信息的取值。

S4403: 在第二语法标识信息指示当前块使用多变换选择模式时, 确定当前块的多个变换核, 并根据多个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。

5 需要说明的是, 在本申请实施例中, 针对步骤 S4401 来说, 解码码流, 确定当前块的第一语法标识信息的取值, 可以包括: 根据至少一个上下文模型解码码流, 确定第一语法标识信息的二元符号串; 以及对第一语法标识信息的二元符号串进行反二值化处理, 确定第一语法标识信息的取值。针对步骤 S4402 来说, 解码码流, 确定当前块的第二语法标识信息的取值, 可以包括: 根据至少一个上下文模型解码码流, 确定第二语法标识信息的二元符号串; 以及对第二语法标识信息的二元符号串进行反二值化处理, 确定第二语法标识信息的取值。在这里, 二元符号串中的每个二元符号使用一个上下文模型。

10 还需要说明的是, 在本申请实施例中, 第一语法标识信息用 `lfnst_nspt_idx` 表示, 第二语法标识信息用 `mts_idx` 表示。其中, `lfnst_nspt_idx` 用于指示当前块是否使用第一变换模式 (即 LFNST/NSPT), `mts_idx` 用于指示当前块是否使用多变换选择模式 (即 MTS)。也就是说, 在解码码流时, LFNST/NSPT 的语法元素 `lfnst_nspt_idx` 可以在 MTS 的语法元素 `mts_idx` 之前。

15 还需要说明的是, 在本申请实施例中, 如果第一语法标识信息指示当前块使用第一变换模式时, 那么不再需要解码 `mts_idx`, 此时可以确定 LFNST/NSPT 的变换核, 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。

也就是说, 在本申请实施例中, 帧间预测和帧内预测 LFNST/NSPT 和 MTS 的解码顺序不同。具体地, 在解码码流时, LFNST/NSPT 的语法元素 `lfnst_nspt_idx` 在 MTS 的语法元素 `mts_idx` 之前。也就是先解析 `lfnst_nspt_idx`, 如果 `lfnst_nspt_idx` 的取值为 0, 而且当前情况符合 MTS 的应用条件, 那么继续解析 `mts_idx`。在这里, `mts_idx` 指示当前块的水平垂直方向分别使用哪种 (基础) 变换核。另外, 在帧内预测时同样地, `lfnst_nspt_idx` 的解析可以在 `mts_idx` 之前。换句话说, `mts_idx` 是否解析依赖于 `lfnst_nspt_idx` 的取值。

20 在本申请实施例中, 针对当前块的帧间预测, `mts_idx` 的解析可以在 `lfnst_nspt_idx` 之前。换句话说, `lfnst_nspt_idx` 是否解析依赖于 `mts_idx` 的取值。也就是先解析 `mts_idx`, 如果 `mts_idx` 为 0, 而且当前情况符合 LFNST/NSPT 的应用条件, 那么继续解析 `lfnst_nspt_idx`。

需要注意的是, 本申请实施例中所述帧间或帧内预测可以理解成帧间编码的块或编码单元或变换单元, 所述帧内或帧内预测可以理解成帧内编码的块或编码单元或变换单元。

30 还需要说明的是, 在本申请实施例中, 这里也可以使用两个语法标识信息 (第三语法标识信息和第四语法标识信息) 来替代 `lfnst_nspt_idx`。其中, 第三语法标识信息用 `nspt_idx` 表示, 可以用于指示当前块是否使用 NSPT 以及对应使用的哪一个 NSPT 变换核, 第四语法标识信息用 `lfnst_idx` 表示, 可以用于指示当前块是否使用 LFNST 以及对应使用的哪一个 LFNST 变换核。

35 在一些实施例中, 该方法还包括: 根据至少一个第三上下文模型解码码流, 确定当前块的第三二元符号串; 对第三二元符号串进行反二值化处理, 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块使用不可分离基础变换模式时, 确定当前块的变换核, 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。

40 在一些实施例中, 该方法还包括: 根据至少一个第四上下文模型解码码流, 确定当前块的第四二元符号串; 对第四二元符号串进行反二值化处理, 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块使用低频不可分离变换模式时, 确定当前块的变换核, 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。

45 也就是说, 在本申请实施例中, LFNST/NSPT 可以各自使用不同的语法元素, 但是上述方法仍然适用。具体地, 帧内预测和帧间预测对 LFNST 语法元素和 NSPT 语法元素使用不同的解码方法, 帧内预测和帧间预测对 LFNST 语法元素和 NSPT 语法元素使用不同的上下文模型, 帧内预测和帧间预测 LFNST 和 NSPT 可使用的变换核数不同, 帧内预测和帧间预测 LFNST 和 NSPT 可使用的变换核不同, 帧内预测和帧间预测 LFNST/NSPT 和 MTS 的解码顺序不同。

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”。

还可以理解地, 在本申请实施例中, 除了将 LFNST 和 NSPT 应用于帧内预测/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, 具体是将帧间预测替换为 IBC 即可。

在又一种可能的实现方式中, `lfnst_nspt_idx` 可以在当前块的最后一个非零系数位置符合一定条件下才进行解析的。参见图 45, 该方法可以包括:

S4501: 解码码流, 确定当前块的量化系数。

S4502: 对当前块的量化系数进行反量化, 确定当前块的变换系数。

5 S4503: 在当前块的变换系数中的最后一个非零系数位置满足预设条件时, 解码码流, 确定当前块的第一语法标识信息的取值。

S4504: 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, 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。

需要说明的是, 在本申请实施例, 对于步骤 S4503 来说, 解码码流, 确定当前块的第一语法标识信息的取值, 可以包括: 根据至少一个上下文模型解码码流, 确定第一语法标识信息的二元符号串; 以及对第一语法标识信息的二元符号串进行反二值化处理, 确定第一语法标识信息的取值。

还需要说明的是, 在本申请实施例, 在第一语法标识信息指示当前块不使用第一变换模式时, 确定当前块使用的第二变换模式的变换核, 并根据第二变换模式的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。示例性地, 第一变换模式可以为 LFNST/NSPT, 第二变换模式可以为 MTS, 但是不作具体限定。

15 在相关技术中, `lfnst_nspt_idx` 是在符合一定条件下才会解析的。其中一个条件是, 最后一个非零系数位置为大于或等于 1。量化系数的扫描是按分层对角线扫描方式 (hierarchical diagonal scan) 进行的。一个系数块可以分成 4×4 的子块, 4×4 的子块之间是按对角线扫描方式 (diagonal scan) 进行扫描的, 4×4 的子块内部也是按照 diagonal scan 进行扫描的。当前块左上角 (0, 0) 的位置是扫描顺序的 0, 最后一个非零系数位置为大于或等于 1, 也就意味着当前块有 (0, 0) 以外的系数。对 DCT2 变换来说, (0, 0) 就是 DC 系数。也就是说, 如果当前块的系数只有 (0, 0) 位置有非零系数或者没有非零系数, 那么 `lfnst_nspt_idx` 这个语法元素不需要解析, 默认为 0。因为在这种情况下 LFNST/NSPT 能使用到的可能性很小, 而通过这个限制可以明显减少 `lfnst_nspt_idx` 的不必要的开销。而且相关技术要求所有颜色分量的最后一个非零系数位置需要大于或等于 1。比如说 YUV 格式下单树划分的块, Y、U、V 三个颜色分量的最后一个非零系数位置都需要大于或等于 1。

这样, 针对帧间预测的块来说, 本来残差就比帧内预测的块少, 又考虑到一般色度分量的质量要求比亮度分量低得多, 在帧间预测的块中会出现较多的亮度分量有一定系数而色度分量没有系数的情况。由于 LFNST/NSPT 的这个条件是为帧内预测设计的, 而目前帧间预测也可以使用 LFNST/NSPT, 但它们的系数、残差规律并不相同, 因而本申请实施例可以对这个限制条件进行修改。

30 在一种可能的实现方式中, 当前块的变换系数中的最后一个非零系数位置满足预设条件, 可以包括: 在当前块的预测模式为帧内预测模式时, 当前块的变换系数中所有颜色分量的最后一个非零系数位置大于或等于第一阈值; 或者, 在当前块的预测模式为帧间预测模式时, 当前块的变换系数中亮度分量的最后一个非零系数位置大于或等于第一阈值。

35 在另一种可能的实现方式中, 当前块的变换系数中的最后一个非零系数位置满足预设条件, 可以包括: 在当前块的划分类别为双树划分时, 当前块的变换系数中所有颜色分量的最后一个非零系数位置大于或等于第一阈值; 或者, 在当前块的划分类别为单树划分时, 当前块的变换系数中亮度分量的最后一个非零系数位置大于或等于第一阈值。

需要说明的是, 在本申请实施例, 第一阈值也可以称为最小阈值。其中, 第一阈值可以设置为 1, 或者也可以更大的值, 如 2, 3 等, 这里不作具体限定。

40 还需要说明的是, 本申请实施例对于最后一个非零系数位置等于第一阈值的情况不作限定。也就是说, 当前块的变换系数中的最后一个非零系数位置满足预设条件, 也可以包括: 在当前块的预测模式为帧内预测模式或双树划分时, 当前块的变换系数中所有颜色分量的最后一个非零系数位置大于第一阈值; 或者, 在当前块的预测模式为帧间预测模式或单树划分时, 当前块的变换系数中亮度分量的最后一个非零系数位置大于第一阈值。

45 示例性地, 假定最小阈值等于 1, 那么对帧内预测的块, 仍然使用相关技术。即所有颜色分量的最后一个非零系数位置需要大于或等于 1 的情况下才解析 `lfnst_nspt_idx`; 对帧间预测的块, 在 Y 分量的最后一个非零系数位置大于或等于 1 的情况下才解析 `lfnst_nspt_idx`, 不再要求 U、V 分量。或者, 对双树划分的块, 在它所有颜色分量的最后一个非零系数位置大于或等于 1 的情况下才解析 `lfnst_nspt_idx`; 对单树划分的块, 在 Y 分量的最后一个非零系数位置大于或等于 1 的情况下才解析 `lfnst_nspt_idx`, 不再要求 U、V 分量。

50 还可以理解地, 在相关技术中, `lfnst_nspt_idx` 是在符合一定条件下才会解析的。其中一个条件是, 最后一个非零系数位置不能大于一个阈值, 这个阈值是 LFNST 或 NSPT 最大可能的系数个数减一, 这

里可称之为最大阈值，最大阈值可以根据当前块的尺寸参数，即长度和宽度确定。示例性地，编码端 LFNST 对 4×4 的块进行二次变换时，LFNST 的输入是 4×4 的 DCT2 变换后的系数，输出是 8 个系数，也就是对 4×4 的块 LFNST 输出最多可能只有 8 个系数，那么对解码器来说，如果一个 4×4 的块的最后一个非零系数位置超过 7，即 (8-1)，那么它肯定不是 LFNST 输出的，这样解码器就可以知道当前块肯定没有使用 LFNST，也就不需要再解析它的语法元素。通过这个限制可以明显减少 `lfnst_nspt_idx` 的不必要的开销。而且相关技术要求所有颜色分量的最后一个非零系数位置不能大于其最大阈值。比如说 YUV 格式下单树划分的块，Y、U、V 三个颜色分量的最后一个非零系数位置都不能大于其最大阈值。

这样，考虑到 LFNST/NSPT 的这个条件是为帧内预测设计的，而目前帧间预测也可以使用 LFNST/NSPT，但它们的系数、残差规律并不相同，因而可以对这个限制条件进行修改。

在又一种可能的实现方式中，当前块的变换系数中的最后一个非零系数位置满足预设条件，可以包括：在当前块的预测模式为帧内预测模式时，当前块的变换系数中所有颜色分量的最后一个非零系数位置小于或等于第二阈值；或者，在当前块的预测模式为帧间预测模式时，当前块的变换系数中亮度分量的最后一个非零系数位置小于或等于第二阈值。

在又一种可能的实现方式中，当前块的变换系数中的最后一个非零系数位置满足预设条件，可以包括：在当前块的划分类别为双树划分时，当前块的变换系数中所有颜色分量的最后一个非零系数位置小于或等于第二阈值；或者，在当前块的划分类别为单树划分时，当前块的变换系数中亮度分量的最后一个非零系数位置小于或等于第二阈值。

需要说明的是，在本申请实施例中，第二阈值也可以称为最大阈值。一种具体的实施例中，可以根据当前块的尺寸参数，确定第二阈值。例如，第二阈值可以根据当前块的长度和宽度确定。另一种具体的实施例中，可以将第二阈值设置为当前块的变换系数中系数个数最大值与一的差值。例如，如果当前块 LFNST 输出最多可能只有 8 个系数，那么第二阈值可以设置为 7。

还需要说明的是，本申请实施例对于最后一个非零系数位置等于第二阈值的情况也不作限定。也就是说，当前块的变换系数中的最后一个非零系数位置满足预设条件，还可以包括：在当前块的预测模式为帧内预测模式或双树划分时，当前块的变换系数中所有颜色分量的最后一个非零系数位置小于第二阈值；或者，在当前块的预测模式为帧间预测模式或单树划分时，当前块的变换系数中亮度分量的最后一个非零系数位置小于第二阈值。

示例性地，对帧内预测的块，仍然使用相关技术。即所有颜色分量的最后一个非零系数位置都不大于其最大阈值的情况下才解析 `lfnst_nspt_idx`；对帧间预测的块，在 Y 分量的最后一个非零系数位置不大于其最大阈值的情况下才解析 `lfnst_nspt_idx`，不再要求 U、V 分量。或者，对双树划分的块，在它所有分量的最后一个非零系数位置不大于最大阈值的情况下才解析 `lfnst_nspt_idx`；对单树划分的块，在 Y 分量的最后一个非零系数位置不大于其最大阈值的情况下才解析 `lfnst_nspt_idx`，不再要求 U、V 分量。

也就是说，在本申请实施例中，可以分情况在 Y 分量的最后一个非零系数位置大于或等于最小阈值的情况下才解析 `lfnst_nspt_idx`，不再要求 U、V 分量。或者，也可以分情况在 Y 分量的最后一个非零系数位置不大于最大阈值的情况下才解析 `lfnst_nspt_idx`，不再要求 U、V 分量。

进一步地，在一些实施例中，该方法还包括：确定当前块的预测块；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，如果当前块的预测模式为帧内预测模式，那么对当前块进行帧内预测，确定当前块的预测块；然后对当前块的预测块和当前块的残差块进行加法运算，确定当前块的重建块。还需要说明的是，在本申请实施例中，如果当前块的预测模式为帧间预测模式，那么对当前块进行帧间预测，确定当前块的预测块；然后对当前块的预测块和当前块的残差块进行加法运算，确定当前块的重建块。

本实施例提供了一种解码方法，确定当前块的预测模式；根据当前块的预测模式，确定至少一个上下文模型；根据至少一个上下文模型解码码流，确定当前块的二元符号串；其中，二元符号串包括至少一个二元符号；对二元符号串进行反二值化处理，确定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，确定当前块的变换核，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，首先根据当前块的预测模式来确定至少一个上下文模型，然后根据所确定的至少一个上下文模型对第一语法标识信息对应的二元符号串进行解码处理；如此，针对帧内预测模式或者帧间预测模式可以选择不同的上下文模型，使得能够对帧间预测模式下 LFNST/NSPT 的二元符号串使用符合其分布规律的解码方法，从而不仅可以提高压缩效率，而且还可以提升编解码性能。

在本申请的一实施例中，参见图 46，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 46 所示，该方法可以包括：

S4601: 确定当前块的预测模式。

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可以应用于帧内预测模式和/或帧间预测模式, 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, 以提高压缩效率。

5 还需要说明的是, 在本申请实施例中, 当前块的预测模式可以包括: 帧间预测模式和/或帧内预测模式。其中, 帧内预测模式主要是根据当前块周边的已重建区域来预测当前块, 帧间预测模式主要是根据当前块的参考图像来预测当前块。

S4602: 根据当前块的预测模式, 确定至少一个上下文模型。

需要说明的是, 在本申请实施例中, 确定帧间预测模式对应的至少一个上下文模型与帧内预测模式对应的至少一个上下文模型不同。也就是说, 帧间预测模式与帧内预测模式不共用上下文模型。

S4603: 确定第一语法标识信息的取值, 并对所述第一语法标识信息的取值进行二值化处理, 确定所述当前块的二元符号串; 其中, 所述二元符号串包括至少一个二元符号。

S4604: 根据所述至少一个上下文模型对所述二元符号串中的至少一个二元符号分别进行编码处理, 将所得到的编码比特写入码流。

15 还需要说明的是, 在本申请实施例中, 上下文模型的数量与二元符号的数量有关。如果当前块的二元符号串包括 n 个二元符号, 那么这里根据当前块的预测模式, 需要确定 n 个上下文模型。其中, 每一个二元符号在编码时使用一个上下文模型。在这里, n 为正整数。

20 还需要说明的是, 在本申请实施例中, 确定第一语法标识信息的取值, 可以包括: 若当前块不使用第一变换模式, 则确定第一语法标识信息的取值为第一值; 和/或, 若当前块使用第一变换模式, 则确定第一语法标识信息的取值为第二值。

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, 第一值可以设置为 0, 第二值可以设置为非零, 例如 1、2、3 等。

25 可以理解地, 在本申请实施例中, NSPT 和 LFNST 都是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, 垂直渐变纹理, 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。

30 还需要说明的是, 在本申请实施例中, NSPT 和 LFNST 都包括有多个变换核组, 每一个变换核组包括 3 个变换核可选, 再加上不使用 LFNST/NSPT 的情况, 也就是一个块针对 LFNST/NSPT 有 4 种可能的选择。在本申请实施例中可以使用语法元素 `lfnst_idx` 表示是否使用 LFNST 以及使用哪一个 LFNST 变换核。`lfnst_idx` 等于 0, 表示当前块不使用 LFNST; `lfnst_idx` 等于 1, 表示当前块使用 LFNST 的第一个变换核; `lfnst_idx` 等于 2, 表示当前块使用 LFNST 的第二个变换核。需要注意的是, 变换核组可以由帧内预测模式或纹理特征索引确定, 此时不需要语法元素指示。另外, 对于一个变换核组, 在一些技术中只能选择 2 个变换核, 在另一些技术中可以选择 3 个变换核。而且对一定尺寸的小块则使用 NSPT, 也可以是 3 个变换核。在本申请实施例中, 这里可以使用 `lfnst_nspt_idx` 来指示是否使用 LFNST/NSPT 以及使用哪一个 LFNST/NSPT 的变换核。因为在一种可能的实现方式中它对较小的块指示的是 NSPT 的变换核索引, 对较大的块指示的是 LFNST 的变换核索引。也就是说, `lfnst_nspt_idx` 的值可以为 0, 1, 2, 3。

40 在本申请实施例中, 第一语法标识信息可以用 `lfnst_nspt_idx` 表示, 也可以继续使用 `lfnst_idx` 这个语法元素, 或者还可以使用 `lfnst_idx` 和 `nspt_idx` 两个语法元素, 其中, `lfnst_idx` 用于指示是否使用 LFNST 以及使用哪一个 LFNST 的变换核, `nspt_idx` 用于指示是否使用 NSPT 以及使用哪一个 NSPT 的变换核; 这里实质上是一样的, 在此不作具体限定。

45 在一种可能的实现方式中, 针对帧内预测模式和帧间预测模式, 第一语法标识信息对应的二元符号串可以使用相同的二值化方式, 但是不同的上下文模型。

在一些实施例中, 对第一语法标识信息的取值进行二值化处理, 确定当前块的二元符号串, 可以包括: 在当前块的预测模式为帧内预测模式或帧间预测模式时, 基于第一方式对第一语法标识信息的取值进行二值化处理, 确定当前块的二元符号串。

50 在一些实施例中, 根据至少一个上下文模型对二元符号串中的至少一个二元符号分别进行编码处理, 可以包括: 在当前块的预测模式为帧内预测模式时, 根据至少一个第一上下文模型对二元符号串中的至少一个二元符号分别进行编码处理; 或者, 在当前块的预测模式为帧间预测模式时, 根据至少一个第二

上下文模型对二元符号串中的至少一个二元符号分别进行编码处理。

需要说明的是,在本申请实施例中,至少一个第一上下文模型与至少一个第二上下文模型不同,即帧间预测模式与帧内预测模式不共用上下文模型。

还需要说明的是,在本申请实施例中,无论是帧间预测模式还是帧内预测模式均可以使用第一方式进行编码处理。其中,第一方式可以为定长编码方式,也可称为固定长度(Fixed-Length, FL)编码方式,即采用相同数目的二元符号表示第一语法标识信息的不同取值;或者,第一方式也可以为截断一元码方式,即采用不同数目的二元符号表示第一语法标识信息的不同取值;或者,第一方式甚至还可以为其他方式,如一元码方式、K阶指数哥伦布编码方式等,这里不作具体限定。

示例性地, `lfnst_nspt_idx` 的取值可以为 0、1、2、3。如果第一方式为定长编码方式,可以使用前述表 6 所示的二值化表格。在表 6 中,第一语法标识信息的取值都是使用 2 个二元符号,这 2 个二元符号可以都是使用上下文模型进行编码。其中,第一个(`binIdx` 等于 0)二元符号使用一个上下文模型,第二个(`binIdx` 等于 1)二元符号使用一个上下文模型。或者,如果第一方式为截断一元码方式,可以使用表 7 所示的二值化表格。在表 7 中,第一语法标识信息的取值可以使用不同的二元符号串长度。其中,表 7 中的 3 个二元符号可以都是上下文模型进行编码。其中,第一个(`binIdx` 等于 0)二元符号使用一个上下文模型,第二个(`binIdx` 等于 1)二元符号使用一个上下文模型,第三个(`binIdx` 等于 2)二元符号使用一个上下文模型。

在另一种可能的实现方式中,针对帧内预测模式和帧间预测模式,第一语法标识信息对应的二元符号串可以使用不同的二值化方式,以及使用不同的上下文模型。

在一些实施例中,对第一语法标识信息的取值进行二值化处理,确定当前块的二元符号串,可以包括:在当前块的预测模式为帧内预测模式时,基于第一方式对第一语法标识信息的取值进行二值化处理,确定当前块的第一二元符号串;或者,在当前块的预测模式为帧间预测模式时,基于第二方式对第一语法标识信息的取值进行二值化处理,确定当前块的第二二元符号串。

在一些实施例中,根据至少一个上下文模型对二元符号串中的至少一个二元符号分别进行编码处理,可以包括:在当前块的预测模式为帧内预测模式时,根据至少一个第一上下文模型和第一方式对第一二元符号串中的至少一个二元符号分别进行编码处理;或者,在当前块的预测模式为帧间预测模式时,根据至少一个第二上下文模型和第二方式对第二二元符号串中的至少一个二元符号分别进行编码处理。

需要说明的是,在本申请实施例中,至少一个第一上下文模型与至少一个第二上下文模型不同,即帧间预测模式与帧内预测模式不共用上下文模型。另外,第一方式与第二方式不同,即帧间预测模式与帧内预测模式使用不同的二值化方式。

在一种具体的实施例中,第一方式为定长编码方式,第二方式为截断一元码方式。

如此,在当前块的预测模式为帧内预测模式时,根据至少一个第一上下文模型和定长编码方式对第一二元符号串进行编码处理;或者,在当前块的预测模式为帧间预测模式时,根据至少一个第二上下文模型和截断一元码方式对第二二元符号串进行编码处理。在这种实现方式下,对于第一语法标识信息 `lfnst_nspt_idx` 的不同取值,第一二元符号串的长度不变,第二二元符号串的长度则是变化的。

简单来说,针对帧内预测模式和帧间预测模式的变换核索引(即 `lfnst_nspt_idx` 的取值),这里可以使用不同的编码方法,而且也可以使用不同的上下文模型。具体地,帧间预测模式的 `lfnst_nspt_idx` 使用的上下文模型和帧内预测模式的 `lfnst_nspt_idx` 使用的上下文模型不同,或者说帧间预测模式的 `lfnst_nspt_idx` 和帧内预测模式的 `lfnst_nspt_idx` 不共用上下文模型。其中,一种可能的实现方式是帧间预测模式的 `lfnst_nspt_idx` 和帧内预测模式的 `lfnst_nspt_idx` 使用相同的二值化方式,但是不同的上下文模型。另一种可能的实现方式是帧间预测模式的 `lfnst_nspt_idx` 和帧内预测模式的 `lfnst_nspt_idx` 使用不同的二值化方式,以及不同的上下文模型。

还需要说明的是,在本申请实施例中,LFNST/NSPT 可以共用一个语法元素 `lfnst_nspt_idx`,具体是对较小的块指示的是 NSPT 的变换核索引,对较大的块指示的是 LFNST 的变换核索引。或者,LFNST/NSPT 也可能各自使用不同的语法元素,例如 LFNST 使用语法元素 `lfnst_idx`,NSPT 使用语法元素 `nspt_idx`,但是针对 `lfnst_idx` 和 `nspt_idx` 的二值化处理,上述方法仍然适用。

进一步地,如果当前块使用 LFNST/NSPT,那么在一些实施例中,参见图 47,该方法还包括:

S4701: 在当前块使用第一变换模式时,确定当前块的变换核。

S4702: 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。

S4703: 对当前块的变换系数进行量化,确定当前块的量化系数。

S4704: 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。

需要说明的是,在本申请实施例中,在当前块使用 LFNST/NSPT 时,首先确定当前块的变换核和残差块,然后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,最终对当前块的变换系

数进行编码处理,将所得到的编码比特写入码流。

还需要说明的是,在本申请实施例中,确定当前块的变换核,可以包括:确定当前块的变换核组;根据变换核组与第一语法标识信息的取值,确定当前块的变换核。

在这里,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用于指示当前块的变换核在变换核组的编号。这样,根据变换核组与第一语法标识信息的取值,可以确定出当前块的变换核。

还需要说明的是,在本申请实施例中,对于当前块进行帧内预测时,可以根据帧内预测模式与变换核组之间的对应关系,确定出当前块的变换核组。对于当前块进行帧间预测时,可以根据纹理特征索引(即虚拟的帧内预测模式)与变换核组之间的对应关系,确定出当前块的变换核组。其中,可以使用当前块的预测块或参考块或当前块周边的重建区域来推导当前块的纹理特征索引。

在一些实施例中,确定当前块的变换核组,可以包括:在当前块的预测模式为帧内预测模式时,确定变换核组中包括M个候选变换核;或者,在当前块的预测模式为帧间预测模式时,确定变换核组中包括N个候选变换核。其中,M、N均为正整数,且M大于或等于N。

也就是说,在本申请实施例中,LFNST/NSPT可能有更多或者更少的选择,帧间预测的LFNST/NSPT可用的变换核数量可能和帧内预测的LFNST/NSPT可用的变换核数量可以不同。示例性地,由于帧内预测的残差更多,帧间预测的残差更少,帧内预测的一个变换核组有3个变换核,而帧间预测的一个变换核组有2个变换核。在这种情况下,针对帧间预测的`lfnst_nspt_idx`的取值,可以为0,1,2。

示例性地,前述表8提供了另一种帧间预测的`lfnst_nspt_idx`的二值化方式。根据表8可以看到,帧间预测的`lfnst_nspt_idx`的二值化方式可以使用不同的二元符号串长度,是一种截断一元码方式。这2个二元符号可以都是上下文模型进行编码的。第一个(`binIdx`等于0)二元符号使用一个上下文模型,第二个(`binIdx`等于1)二元符号使用一个上下文模型。

还需要说明的是,帧间预测的`lfnst_nspt_idx`使用的上下文模型和帧内预测的`lfnst_nspt_idx`使用的上下文模型不同,或者说帧间预测的`lfnst_nspt_idx`和帧内预测的`lfnst_nspt_idx`不共用上下文模型。另外,LFNST/NSPT也可能各自使用不同的语法元素,上述方法仍然适用。

在一些实施例中,该方法还包括:在M大于N时,将N个候选变换核设置为M个候选变换核的一个子集;或者,在M等于N时,将N个候选变换核设置为与M个候选变换核中至少部分不同。

需要说明的是,在本申请实施例中,帧间LFNST/NSPT可用的变换核可以和帧内LFNST/NSPT可用的变换核相同。如果帧间LFNST/NSPT可用的变换核数量小于帧内LFNST/NSPT可用的变换核数量,那么帧间LFNST/NSPT可用的变换核可以是帧内LFNST/NSPT可用的变换核的一个子集。示例性地,帧间预测可以使用2个变换核,帧内预测可以使用3个变换核,帧间预测可用的变换核可以是帧内预测可用的变换核的前2个或者后2个或者第1、3个,这里不作具体限定。

还需要说明的是,在本申请实施例中,帧间LFNST/NSPT可用的变换核可以和帧内LFNST/NSPT可用的变换核不同。即针对帧间预测和帧内预测的残差特征分别设计变换核,这样能达到更高的压缩效率。示例性地,帧间预测可以使用3个变换核,帧内预测也可以使用3个变换核,但是帧间预测和帧内预测的3个变换核部分或全部不同。

也就是说,在本申请实施例中,帧间预测使用的变换核和帧内预测使用的变换核可以相同,也可以部分或全部不同。另外,在本申请实施例中,所述可用的变换核是指一个变换核组可用的变换核。

在一些实施例中,在确定当前块的变换核组时,该方法还可以包括:确定当前块的至少一个纹理特征;根据至少一个纹理特征进行变换核训练,确定当前块的变换核组。

需要说明的是,相关技术中帧内预测可以根据帧内预测模式对应一个变换核组,在本申请实施例中,帧间预测可以用预测块导出虚拟的帧内预测模式(纹理特征索引)对应一个变换核组。帧间预测也可以导出和帧内预测不同的纹理特征,对应地以这样的纹理特征来训练变换核。

在一些实施例中,该方法还包括:确定当前块的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。

在这里,第二语法标识信息可以用`mts_idx`表示,用于指示当前块是否使用多变换选择模式,以及当前块的水平/竖直方向分别使用哪种(基础)变换核。

还需要说明的是,如果当前块不使用MTS模式,那么确定第二语法标识信息的取值为第一值;如果当前块使用MTS模式,那么确定第二语法标识信息的取值为第二值。其中,第一值可以设置为0,第二值可以设置为非0,例如1、2、3等。在这里,第二值的大小与当前块对应使用的哪种变换核有关。

在一些实施例中,该方法还包括:在当前块不使用多变换选择模式且当前块满足第一变换模式的使用条件时,执行确定第一语法标识信息的取值的步骤。

需要说明的是,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第一值;如果当前块使用LFNST/NSPT,那么确定第一语法标识信息的取值为第二值。其中,第一值可以设置为0,

第二值可以设置为非 0，例如 1、2、3 等。第二值的大小与当前块对应使用的哪种变换核有关。

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和第二语法标识信息都可以使用上下文模型进行编码。另外，第一语法标识信息用 `lfnst_nspt_idx` 表示，第二语法标识信息用 `mts_idx` 表示。其中，`lfnst_nspt_idx` 用于指示当前块是否使用第一变换模式（即 LFNST/NSPT），`mts_idx` 用于指示当前块是否使用多变换选择模式（即 MTS）。也就是说，在写入码流时，LFNST/NSPT 的语法元素 `lfnst_nspt_idx` 可以在 MTS 的语法元素 `mts_idx` 之后。

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指示当前块使用多变换选择模式时，那么不再需要编码 `lfnst_nspt_idx`，此时可以确定当前块的多个变换核，并根据多个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。例如，`mts_idx` 可以用于指示当前块的水平/竖直方向分别使用哪种（基础）变换核。

在一些实施例中，该方法还包括：在当前块不使用第一变换模式且当前块满足多变换选择模式的使用条件时，执行确定当前块的第二语法标识信息的取值的步骤。

需要说明的是，在本申请实施例中，对于第一语法标识信息和第二语法标识信息也都可以使用上下文模型进行编码。其中，第一语法标识信息用 `lfnst_nspt_idx` 表示，第二语法标识信息用 `mts_idx` 表示。其中，`lfnst_nspt_idx` 用于指示当前块是否使用第一变换模式（即 LFNST/NSPT），`mts_idx` 用于指示当前块是否使用多变换选择模式（即 MTS）。也就是说，在写入码流时，LFNST/NSPT 的语法元素 `lfnst_nspt_idx` 可以在 MTS 的语法元素 `mts_idx` 之前。

还需要说明的是，在本申请实施例中，如果当前块使用第一变换模式时，那么不再需要编码 `mts_idx`，此时可以确定 LFNST/NSPT 的变换核，并根据该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。

也就是说，在本申请实施例中，帧间预测和帧内预测 LFNST/NSPT 和 MTS 的编码顺序不同。具体地，在编码过程时，LFNST/NSPT 的语法元素 `lfnst_nspt_idx` 在 MTS 的语法元素 `mts_idx` 之前。也就是后续在解码端先解析 `lfnst_nspt_idx`，如果 `lfnst_nspt_idx` 的取值为 0，而且当前情况符合 MTS 的应用条件，那么继续解析 `mts_idx`。在这里，`mts_idx` 指示当前块的水平/竖直方向分别使用哪种（基础）变换核。另外，在帧内预测时同样地，`lfnst_nspt_idx` 的编码可以在 `mts_idx` 之前。换句话说，`mts_idx` 是否编码依赖于 `lfnst_nspt_idx` 的取值。

在本申请实施例中，针对当前块的帧间预测，`mts_idx` 的编码可以在 `lfnst_nspt_idx` 之前。换句话说，`lfnst_nspt_idx` 是否编码依赖于 `mts_idx` 的取值。也就是后续在解码端先解析 `mts_idx`，如果 `mts_idx` 为 0，而且当前情况符合 LFNST/NSPT 的应用条件，那么继续解析 `lfnst_nspt_idx`。

需要注意的是，本申请实施例中所述帧间或帧内预测可以理解成帧间编码的块或编码单元或变换单元，所述帧内或帧内预测可以理解成帧内编码的块或编码单元或变换单元。

还需要说明的是，在本申请实施例中，这里也可以使用两个语法标识信息（第三语法标识信息和第四语法标识信息）来替代 `lfnst_nspt_idx`。其中，第三语法标识信息用 `nspt_idx` 表示，可以用于指示当前块是否使用 NSPT 以及对应使用的哪一个 NSPT 变换核，第四语法标识信息用 `lfnst_idx` 表示，可以用于指示当前块是否使用 LFNST 以及对应使用的哪一个 LFNST 变换核。

在一些实施例中，该方法还包括：确定当前块的第三语法标识信息的取值；其中，第三语法标识信息用于指示当前块是否使用不可分离基础变换模式；对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。

在本申请实施例中，如果当前块不使用不可分离基础变换模式，那么确定第三语法标识信息的取值为第一值；如果当前块使用不可分离基础变换模式，那么确定第三语法标识信息的取值为第二值。

在本申请实施例中，在当前块使用不可分离基础变换模式时，确定当前块的变换核，并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。

在一些实施例中，该方法还包括：确定当前块的第四语法标识信息的取值；其中，第四语法标识信息用于指示当前块是否使用低频不可分离变换模式；对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。

在本申请实施例中，如果当前块不使用低频不可分离变换模式，那么确定第四语法标识信息的取值为第一值；如果当前块使用低频不可分离变换模式，那么确定第四语法标识信息的取值为第二值。

在本申请实施例中，在当前块使用低频不可分离变换模式时，确定当前块的变换核，并根据变换核对当前块的残差块进行 LFNST 变换，确定当前块的变换系数。

还需要说明的是，在本申请实施例中，第一值可以设置为 0，第二值可以设置为非 0，例如 1、2、3 等。在这里，第二值的大小与当前块对应使用的哪种变换核有关。

也就是说，在本申请实施例中，LFNST/NSPT 可以各自使用不同的语法元素，但是上述方法仍然适

用。具体地，帧内预测和帧间预测对 LFNST 语法元素和 NSPT 语法元素使用不同的编码方法，帧内预测和帧间预测对 LFNST 语法元素和 NSPT 语法元素使用不同的上下文模型，帧内预测和帧间预测 LFNST 和 NSPT 可使用的变换核数不同，帧内预测和帧间预测 LFNST 和 NSPT 可使用的变换核不同，帧内预测和帧间预测 LFNST/NSPT 和 MTS 的编码顺序不同。

5 还可以理解地，在本申请实施例中，除了将 LFNST 和 NSPT 应用于帧内预测/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，具体是将帧间预测替换为 IBC 即可。

在又一种可能的实现方式中，`lfnst_nspt_idx` 可以在当前块的最后一个非零系数位置符合一定条件下才进行编码入码流的。在一些实施例中，该方法还包括：在当前块的变换系数中的最后一个非零系数位置满足预设条件时，执行确定第一语法标识信息的取值，并对所述第一语法标识信息的取值进行二值化处理，确定所述当前块的二元符号串的步骤。

10 在相关技术中，`lfnst_nspt_idx` 是在符合一定条件下才会编码的。其中一个条件是，最后一个非零系数位置为大于或等于 1。量化系数的扫描是按 hierarchical diagonal scan 进行的。一个系数块可以分成 4×4 的子块， 4×4 的子块之间是按 diagonal scan 进行扫描的， 4×4 的子块内部也是按照 diagonal scan 进行扫描的。当前块左上角 $(0, 0)$ 的位置是扫描顺序的 0，最后一个非零系数位置为大于或等于 1，也就意味着当前块有 $(0, 0)$ 以外的系数。对 DCT2 变换来说， $(0, 0)$ 就是 DC 系数。也就是说，如果当前块的系数只有 $(0, 0)$ 位置有非零系数或者没有非零系数，那么 `lfnst_nspt_idx` 这个语法元素不需要编码，默认为 0。因为在这种情况下 LFNST/NSPT 能使用到的可能性很小，而通过这个限制可以明显减少 `lfnst_nspt_idx` 的不必要的开销。而且相关技术要求所有颜色分量的最后一个非零系数位置需要大于或等于 1。比如说 YUV 格式下单树划分的块，Y、U、V 三个颜色分量的最后一个非零系数位置都需要大于或等于 1。

20 这样，针对帧间预测的块来说，本来残差就比帧内预测的块少，又考虑到一般色度分量的质量要求比亮度分量低得多，在帧间预测的块中会出现较多的亮度分量有一定系数而色度分量没有系数的情况。由于 LFNST/NSPT 的这个条件是为帧内预测设计的，而目前帧间预测也可以使用 LFNST/NSPT，但它们的系数、残差规律并不相同，因而本申请实施例可以对这个限制条件进行修改。

25 在一种可能的实现方式中，当前块的变换系数中的最后一个非零系数位置满足预设条件，可以包括：在当前块的预测模式为帧内预测模式时，当前块的变换系数中所有颜色分量的最后一个非零系数位置大于或等于第一阈值；或者，在当前块的预测模式为帧间预测模式时，当前块的变换系数中亮度分量的最后一个非零系数位置大于或等于第一阈值。

30 在另一种可能的实现方式中，当前块的变换系数中的最后一个非零系数位置满足预设条件，可以包括：在当前块的划分类别为双树划分时，当前块的变换系数中所有颜色分量的最后一个非零系数位置大于或等于第一阈值；或者，在当前块的划分类别为单树划分时，当前块的变换系数中亮度分量的最后一个非零系数位置大于或等于第一阈值。

需要说明的是，在本申请实施例中，第一阈值也可以称为最小阈值。其中，第一阈值可以设置为 1，或者也可以更大的值，如 2、3 等，这里不作具体限定。

35 还需要说明的是，本申请实施例对于最后一个非零系数位置等于第一阈值的情况不作限定。也就是说，当前块的变换系数中的最后一个非零系数位置满足预设条件，也可以包括：在当前块的预测模式为帧内预测模式或双树划分时，当前块的变换系数中所有颜色分量的最后一个非零系数位置大于第一阈值；或者，在当前块的预测模式为帧间预测模式或单树划分时，当前块的变换系数中亮度分量的最后一个非零系数位置大于第一阈值。

40 示例性地，假定最小阈值等于 1，那么对帧内预测的块，仍然使用相关技术。即所有颜色分量的最后一个非零系数位置需要大于或等于 1 的情况下才编码 `lfnst_nspt_idx`；对帧间预测的块，在 Y 分量的最后一个非零系数位置大于或等于 1 的情况下才编码 `lfnst_nspt_idx`，不再要求 U、V 分量。或者，对双树划分的块，在它所有颜色分量的最后一个非零系数位置大于或等于 1 的情况下才编码 `lfnst_nspt_idx`；对单树划分的块，在 Y 分量的最后一个非零系数位置大于或等于 1 的情况下才编码 `lfnst_nspt_idx`，不再要求 U、V 分量。

45 还可以理解地，在相关技术中，`lfnst_nspt_idx` 是在符合一定条件下才会编码的。其中一个条件是，最后一个非零系数位置不能大于一个阈值，这个阈值是 LFNST 或 NSPT 最大可能的系数个数减一，这里可称之为最大阈值，最大阈值可以根据当前块的尺寸参数，即长度和宽度确定。示例性地，编码端 LFNST 对 4×4 的块进行二次变换时，LFNST 的输入是 4×4 的 DCT2 变换后的系数，输出是 8 个系数，也就是对 4×4 的块 LFNST 输出最多可能只有 8 个系数，那么对解码器来说，如果一个 4×4 的块的最后一个非零系数位置超过 7，即 $(8-1)$ ，那么它肯定不是 LFNST 输出的，这样解码器就可以知道当前块肯定没有使用 LFNST，也就不需要再编码它的语法元素。通过这个限制可以明显减少 `lfnst_nspt_idx` 的

不必要的开销。而且相关技术要求所有颜色分量的最后一个非零系数位置不能大于其最大阈值。比如说 YUV 格式下单树划分的块，Y、U、V 三个颜色分量的最后一个非零系数位置都不能大于其最大阈值。

这样，考虑到 LFNST/NSPT 的这个条件是为帧内预测设计的，而目前帧间预测也可以使用 LFNST/NSPT，但它们的系数、残差规律并不相同，因而可以对这个限制条件进行修改。

- 5 在又一种可能的实现方式中，当前块的变换系数中的最后一个非零系数位置满足预设条件，可以包括：在当前块的预测模式为帧内预测模式时，当前块的变换系数中所有颜色分量的最后一个非零系数位置小于或等于第二阈值；或者，在当前块的预测模式为帧间预测模式时，当前块的变换系数中亮度分量的最后一个非零系数位置小于或等于第二阈值。

- 10 在又一种可能的实现方式中，当前块的变换系数中的最后一个非零系数位置满足预设条件，可以包括：在当前块的划分类别为双树划分时，当前块的变换系数中所有颜色分量的最后一个非零系数位置小于或等于第二阈值；或者，在当前块的划分类别为单树划分时，当前块的变换系数中亮度分量的最后一个非零系数位置小于或等于第二阈值。

- 15 需要说明的是，在本申请实施例中，第二阈值也可以称为最大阈值。一种具体的实施例中，可以根据当前块的尺寸参数，确定第二阈值。例如，第二阈值可以根据当前块的长度和宽度确定。另一种具体的实施例中，可以将第二阈值设置为当前块的变换系数中系数个数最大值与一的差值。例如，如果当前块 LFNST 输出最多可能只有 8 个系数，那么第二阈值可以设置为 7。

- 20 还需要说明的是，本申请实施例对于最后一个非零系数位置等于第二阈值的情况也不作限定。也就是说，当前块的变换系数中的最后一个非零系数位置满足预设条件，还可以包括：在当前块的预测模式为帧内预测模式或双树划分时，当前块的变换系数中所有颜色分量的最后一个非零系数位置小于第二阈值；或者，在当前块的预测模式为帧间预测模式或单树划分时，当前块的变换系数中亮度分量的最后一个非零系数位置小于第二阈值。

- 25 示例性地，对帧内预测的块，仍然使用相关技术。即所有颜色分量的最后一个非零系数位置都不大于其最大阈值的情况下才编码 `lfnst_nspt_idx`；对帧间预测的块，在 Y 分量的最后一个非零系数位置不大于其最大阈值的情况下才编码 `lfnst_nspt_idx`，不再要求 U、V 分量。或者，对双树划分的块，在它所有分量的最后一个非零系数位置不大于最大阈值的情况下才编码 `lfnst_nspt_idx`；对单树划分的块，在 Y 分量的最后一个非零系数位置不大于其最大阈值的情况下才编码 `lfnst_nspt_idx`，不再要求 U、V 分量。

- 也就是说，在本申请实施例中，可以分情况在 Y 分量的最后一个非零系数位置大于或等于最小阈值的情况下才编码 `lfnst_nspt_idx`，不再要求 U、V 分量。或者，也可以分情况在 Y 分量的最后一个非零系数位置不大于最大阈值的情况下才编码 `lfnst_nspt_idx`，不再要求 U、V 分量。

- 30 进一步地，在一些实施例中，该方法还包括：确定当前块的预测块；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。

- 需要说明的是，在本申请实施例中，如果当前块的预测模式为帧内预测模式，那么对当前块进行帧内预测，确定当前块的预测块；然后对当前块的原始块和当前块的预测块进行减法运算，确定当前块的残差块。还需要说明的是，在本申请实施例中，如果当前块的预测模式为帧间预测模式，那么对当前块进行帧间预测，确定当前块的预测块；然后对当前块的原始块和当前块的预测块进行减法运算，确定当前块的残差块。

在一种可能的实现方式中，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，可以包括：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。

- 40 在另一种可能的实现方式中，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，可以包括：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；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。

- 45 也就是说，在本申请实施例中，如果 `lfnst_nspt_idx` 指示当前块使用第一变换模式，那么在根据变换核对当前块的残差块进行变换时，可以包括：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。

- 50 在这里，当前块的尺寸参数满足第一条件，可以包括：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。当前块的尺寸参数满足第二条件，可以包括：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。

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”。

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、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,第二语法标识信息用于指示当前块是否使用多变换选择模式,第三语法标识信息用于指示当前块是否使用不可分离基础变换模式,第四语法标识信息用于指示当前块是否使用低频不可分离变换模式。

本实施例提供了一种编码方法,确定当前块的预测模式;根据当前块的预测模式,确定至少一个上下文模型;根据至少一个上下文模型解码码流,确定当前块的二元符号串;其中,二元符号串包括至少一个二元符号;对二元符号串进行反二值化处理,确定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,确定当前块的变换核,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,首先根据当前块的预测模式来确定至少一个上下文模型,然后根据所确定的至少一个上下文模型对第一语法标识信息对应的二元符号串进行编码处理;如此,针对帧内预测模式或者帧间预测模式可以选择不同的上下文模型,使得能够对帧间预测模式下 LFNST/NSPT 的二元符号串使用符合其分布规律的编码方法,从而不仅可以提高压缩效率和编码效率,而且还可以提升编码性能。

在本申请的又一实施例中,基于前述实施例所述的编解码方法,考虑到相关技术中, 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,也就是一个块针对 LFNST/NSPT 有 4 种可能的选择。在本申请实施例中可以用语法元素 `lfnst_idx` 表示是否使用 LFNST 变换核以及使用哪一个 LFNST 变换核。`lfnst_idx` 等于 0 表示当前块不使用 LFNST, `lfnst_idx` 等于 1 表示当前块使用 LFNST 的第一个变换核, `lfnst_idx` 等于 2 表示当前块使用 LFNST 的第二个变换核,需要注意的是,变换核组由帧内预测模式确定,不需要语法元素指示。示例性地,在一些技术中只能选择 2 个变换核,在另一些技术中可以使用 3 个变换核。而且对一定尺寸的小块使用 NSPT,也是 3 个变换核。这里可以用 `lfnst_nspt_idx` 来指示是否使用 LFNST/NSPT 变换核以及使用哪一个 LFNST/NSPT 的变换核。因为在一种实现方式中它对较小的块指示的是 NSPT 的变换核的索引,对较大的块指示的是 LFNST 的索引。也就是说, `lfnst_nspt_idx` 的值可以为 0, 1, 2, 3。当然也可以继续使用 `lfnst_idx` 这个语法元素,或者使用 `lfnst_idx` 和 `nspt_idx` 两个语法元素,实质上是一样的,这里不作具体限定。

图 48 示出了一种语法元素值写入码流的框图示意图,图 49 示出了一种从码流中解码语法元素值的框图示意图。其中,在编码时,语法元素值(syntax element value)经过二值化(binization)变成二元符号串(bin string),二元符号串再经过 CABAC 变成码流(bit stream)。在解码时,解码器要解析某个语法元素时,从读取码流经过 CABAC 解码出二元符号串,二元符号串经过反二值化得到语法元素值。CABAC(context-based adaptive binary arithmetic coding)编码时,如果一个二元符号使用了上下文模型,则使用对应上下文模型的概率进行编码,否则,使用等概率进行编码,解码时亦然。需要注意的是,这里的语法元素值即为前述实施例中语法标识信息的取值。

在相关技术中,原来的 `lfnst_nspt_idx` 的编码方法是针对帧内预测来优化的,所以这里可以直接称之为帧内的 `lfnst_nspt_idx` 的二值化方法,具体如前述的表 6。据此可以看到帧内的 `lfnst_nspt_idx` 的二值化方法都使用 2 个二元符号,这 2 个二元符号都是上下文模式编码的。第一个(`binIdx` 等于 0)二元符号使用一个上下文模型,第二个(`binIdx` 等于 1)二元符号使用一个上下文模型。

这样设置二值化方法及上下文模型配置和它的统计规律是相关的。图 40 是一种帧内 `lfnst_nspt_idx` 的取值在测试序列 MarketPlace 在 all intra QP 32 下的统计结果。根据图 40 可以看到这个条件下,对于 `lfnst_nspt_idx` 的取值,4 种选择的可能性是差不多的,1 和 2 的可能性甚至比 0 还要多。需要注意的是,这仅是一个示例,不同的测试序列、不同的 QP 对应的统计结果是不同的。

进一步地,使用同样的二值化方式,图 41 是一种帧间 `lfnst_nspt_idx` 在测试序列 BasketballDrive 在 random access QP 32 下的统计结果。据此可以看到这个条件下,对于 `lfnst_nspt_idx` 的取值,4 种选择的可能性有一定差距。0 明显高于 1、2、3,1 高于 2、3。当然也需要注意的是,这仅是一个示例,不同的测试序列、不同的 QP 对应的统计结果是不同的。

基于对帧间 `lfnst_nspt_idx` 的概率分布的理论分析和统计数据,本申请实施例提出了一种帧间 `lfnst_nspt_idx` 的二值化方式,具体如前述表 7 所示。据此可以看到帧间的 `lfnst_nspt_idx` 的二值化方式使用不同的二元符号串长度,是一种截断一元码。这 3 个二元符号可以都是上下文模式编码的。第一个(`binIdx` 等于 0)二元符号使用一个上下文模型,第二个(`binIdx` 等于 1)二元符号使用一个上下文模

型, 第三个 (binIdx 等于 2) 二元符号使用一个上下文模型。

也就是说, 帧间的 `lfnst_nspt_idx` 使用的上下文模型和帧内的 `lfnst_nspt_idx` 使用的上下文模型不同, 或者说帧间的 `lfnst_nspt_idx` 和帧内的 `lfnst_nspt_idx` 不共用上下文模型。一种可能的实现方式是帧间的 `lfnst_nspt_idx` 和帧内的 `lfnst_nspt_idx` 使用相同的二值化方法, 但是不同的上下文模型。另一种可能的实现方式是帧间的 `lfnst_nspt_idx` 和帧内的 `lfnst_nspt_idx` 使用不同的二值化方法, 不同的上下文模型。

还需要说明的是, LFNST/NSPT 可能各自使用不同的语法元素, 上述方法的原则不变。

在本申请实施例中, LFNST/NSPT 可能有更多或者更少的选择, 帧间的 LFNST/NSPT 可用的变换核数可能和帧内的 LFNST/NSPT 可用的变换核数不同。示例性地, 由于帧内的残差更多, 帧间的残差更少, 帧内的一个变换核组有 3 个变换核, 而帧间的一个变换核组有 2 个变换核。那么帧间的 `lfnst_nspt_idx` 的编码方法如前述表 8。据此可以看到帧间的 `lfnst_nspt_idx` 的二值化方法使用不同的二元符号串长度, 是一种截断一元码。这 2 个二元符号可以都是上下文模式编码的。第一个 (binIdx 等于 0) 二元符号使用一个上下文模型, 第二个 (binIdx 等于 1) 二元符号使用一个上下文模型。

在这里, 帧间的 `lfnst_nspt_idx` 使用的上下文模型和帧内的 `lfnst_nspt_idx` 使用的上下文模型不同, 或者说帧间的 `lfnst_nspt_idx` 和帧内的 `lfnst_nspt_idx` 不共用上下文模型。另外, LFNST/NSPT 可能各自使用不同的语法元素, 上述方法的原则不变。

进一步地, 帧间 LFNST/NSPT 可用的变换核可以和帧内 LFNST/NSPT 可用的变换核相同, 如果帧间 LFNST/NSPT 可用的变换核数量小于帧内 LFNST/NSPT 可用的变换核数量, 帧间 LFNST/NSPT 可用的变换核可以是帧内 LFNST/NSPT 可用的变换核的一个子集。一个例子比如帧间可以使用 2 个变换核, 帧内可以使用 3 个变换核, 帧间可用的变换核是帧内可用的变换核的前 2 个, 或后 2 个或第 1, 3 个。

进一步地, 帧间 LFNST/NSPT 可用的变换核可以和帧内 LFNST/NSPT 可用的变换核不同, 即针对帧间和帧内的残差特征分别设计变换核, 这样能达到更高的压缩效率。一个例子是帧间可以使用 3 个变换核, 帧内可以使用 3 个变换核, 但是帧间和帧内的 3 个变换核部分或全部不同。

可选地, 所述可用的变换核指一个变换核组可用的变换核。在相关技术中, 帧内根据帧内预测模式对应一个变换核组, 在本方案中, 帧间用预测块导出虚拟的帧内预测模式 (纹理特征索引) 对应一个变换核组。帧间也可以导出和帧内不同的纹理特征, 对应地以这样的纹理特征训练变换核。

示例性地, 目前 LFNST 和 NSPT 所训练的都是比较单一的纹理特征, 比如说某一个特定方向的纹理, 如图 42 中的 (a) 和 (b) 所示。应注意的是, 这只是一个示意图, 但是对帧间来说, 残差更多的出现在物体的边缘, 比如说如果帧间会经常出现如图 42 中的 (c) 所示的有两个方向纹理的残差, 那么也可以训练这样的 LFNST/NSPT 的变换核, 在分析当前块的纹理特征 (如上述推导虚拟的帧内预测模式的过程) 时, 如果发现当前块的预测块或者周边重建区域有 2 种方向的纹理, 那么就可以匹配具有这样纹理特征的变换核。

在本申请实施例中, 在解析码流时, LFNST 的语法元素 `lfnst_nspt_idx` 在 MTS 的语法元素 `mts_idx` 之前。也就是先解析 `lfnst_nspt_idx`, 如果 `lfnst_nspt_idx` 为 0, 而且当前情况符合 MTS 的应用条件, 解析 `mts_idx`。在这里, `mts_idx` 用于指示当前块的水平垂直方向分别使用哪种 (基础) 变换核。如此, 在帧内编码时同样地, `lfnst_nspt_idx` 的解析也可以在 `mts_idx` 之前, `mts_idx` 是否解析依赖于 `lfnst_nspt_idx` 的值。

在本申请实施例中, 在帧间编码时, `mts_idx` 的解析在 `lfnst_nspt_idx` 之前, `lfnst_nspt_idx` 是否解析依赖于 `mts_idx` 的值。也就是说, 先解析 `mts_idx`, 如果 `mts_idx` 为 0, 而且当前情况符合 LFNST/NSPT 的应用条件, 解析 `lfnst_nspt_idx`。

需要注意的是, 本申请实施例中所述帧间可以理解为帧间编码的块或编码单元, 变换单元, 帧内可以理解为帧内编码的块或编码单元, 变换单元。

还需要注意的是, 本技术方案也可应用于 IBC 模式, 将上文中帧间换为 IBC 即可。

在本申请实施例中, `lfnst_nspt_idx` 是在符合一定条件下才会解析的。其中一个条件是, 最后一个非零系数位置大于等于 1。量化系数的扫描是按 hierarchical diagonal scan 进行的。一个系数块, 会分成 4×4 的子块, 4×4 的子块之间是按 diagonal scan 进行扫描的, 4×4 的子块内部也是按照 diagonal scan 进行扫描的。当前块左上角 (0, 0) 的位置是扫描顺序的 0, 最后一个非零系数位置大于等于 1, 也就意味着当前块有 (0, 0) 以外的系数。对 DCT2 变换来说, (0, 0) 就是 DC 系数。也就是说如果当前块的系数只有 (0, 0) 位置有非零系数或者没有非零系数, 那么 `lfnst_nspt_idx` 这个语法元素不需要解析, 默认为 0。因为这种情况下 LFNST/NSPT 能用到的可能性很小, 而通过这个限制可以明显减少 `lfnst_nspt_idx` 的不必要的开销。而且相关技术要求所有分量的最后一个非零系数位置大于等于 1。比如说 YUV 格式下 single tree 划分的块, Y, U, V 三个分量的最后一个非零系数位置都要大于等于 1。

对帧间编码的块来说, 本来残差就比帧内编码的块少, 又考虑到一般色度的质量要求比亮度低得多,

在帧间编码的块中会出现较多的亮度有一定系数而色度没有系数的情况。因为 LFNST/NSPT 的这个条件是为帧内编码设计的,而现在帧间编码也可以使用 LFNST/NSPT,但它们的系数,残差规律并不相同,因而可以对这个限制条件进行修改。

一种可能的实现方式为:对帧内编码的块,仍然使用相关技术。即所有分量的最后一个非零系数位置大于等于 1 的情况下才解析 `lfnst_nspt_idx`。对帧间编码的块,在 Y 分量的最后一个非零系数位置大于等于 1 的情况下才解析 `lfnst_nspt_idx`。不再要求 UV 分量。

另一种可能的实现方式为:对 dual tree 划分的块,在它所有的分量的最后一个非零系数位置大于等于 1 的情况下才解析 `lfnst_nspt_idx`。对 single tree 划分的块,在 Y 分量的最后一个非零系数位置大于等于 1 的情况下才解析 `lfnst_nspt_idx`。不再要求 UV 分量。

更进一步地,相关技术中判断条件设为最后一个非零系数位置大于等于 1,即这个最小可能的阈值是 1,这里称它为最小阈值,最小阈值也可以设置更大的值,如 2, 3 等。如最后一个非零系数位置大于等于 2 的情况下才解析 `lfnst_nspt_idx`。

在相关技术中, `lfnst_nspt_idx` 是在符合一定条件下才会解析的。其中一个条件是,最后一个非零系数位置不能大于一个阈值,这个阈值是 LFNST 或 NSPT 最大可能的系数个数减一,这里称之为最大阈值,最大阈值可以根据块的尺寸,即长和宽确定。示例性地,编码端 LFNST 对 4×4 的块进行二次变换时, LFNST 的输入是 4×4 的 DCT2 变换后的系数,输出是 8 个系数,也就是对 4×4 的块 LFNST 输出最多可能只有 8 个系数,那么对解码器来说,如果一个 4×4 的块的最后一个非零系数位置超过 7,即 (8-1),那么它肯定不是 LFNST 输出的,这样解码器就知道当前块肯定没用 LFNST,也就不需要解析它的语法元素了。通过这个限制可以明显减少 `lfnst_nspt_idx` 的不必要的开销。

而且相关技术要求所有分量的最后一个非零系数位置不能大于其最大阈值。比如说 YUV 格式下 single tree 划分的块, Y, U, V 三个分量的最后一个非零系数位置都不能大于其最大阈值。因为 LFNST/NSPT 的这个条件是为帧内编码设计的,而现在帧间编码也可以使用 LFNST/NSPT,但它们的系数,残差规律并不相同,因而可以对这个限制条件进行修改。

一种可能的实现方式为:对帧内编码的块,仍然使用相关技术。即所有分量的最后一个非零系数位置都不大于其最大阈值的情况下才解析 `lfnst_nspt_idx`。对帧间编码的块,在 Y 分量的最后一个非零系数位置不大于其最大阈值的情况下才解析 `lfnst_nspt_idx`。不再要求 UV 分量。

另一种可能的实现方式为:对 dual tree 划分的块,在它所有的分量的最后一个非零系数位置不大于最大阈值的情况下才解析 `lfnst_nspt_idx`。对 single tree 划分的块,在 Y 分量的最后一个非零系数位置不大于其最大阈值的情况下才解析 `lfnst_nspt_idx`。不再要求 UV 分量。

在本申请实施例中,通过上述实施例对前述实施例的具体实现进行详细阐述,从中可以看出,根据前述实施例的技术方案,一方面,帧内和帧间对 LFNST/NSPT 的变换核索引使用不同的编码方法,帧内和帧间对 LFNST/NSPT 的索引使用不同的上下文模型,帧内和帧间 LFNST/NSPT 可使用的变换核数不同,帧内和帧间 LFNST/NSPT 可使用的变换核不同,帧间和帧内 LFNST/NSPT 和 MTS 的编码顺序不同;另一方面,分情况在 Y 分量的最后一个非零系数位置大于等于 1 的情况下才解析 `lfnst_nspt_idx`,不再要求 UV 分量;以及分情况在 Y 分量的最后一个非零系数位置不大于最大阈值的情况下才解析 `lfnst_nspt_idx`,不再要求 UV 分量;从而对帧间的 LFNST/NSPT 的变换核索引使用符合其分布规律的编码方法,提高了压缩效率,进而提高了编解码性能。

在本申请的再一实施例中,基于前述实施例相同的发明构思,参见图 50,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 50 所示,该编码器 500 可以包括第一确定单元 5001、二值化单元 5002 和编码单元 5003,其中:

第一确定单元 5001,配置为确定当前块的预测模式;以及根据当前块的预测模式,确定至少一个上下文模型;

二值化单元 5002,配置为确定第一语法标识信息的取值,并对第一语法标识信息的取值进行二值化处理,确定当前块的二元符号串;其中,二元符号串包括至少一个二元符号;

编码单元 5003,配置为根据至少一个上下文模型对二元符号串中的至少一个二元符号分别进行编码处理,将所得到的编码比特写入码流。

在一些实施例中,当前块的预测模式包括:帧间预测模式和/或帧内预测模式;相应地,第一确定单元 5001,还配置为确定帧间预测模式对应的至少一个上下文模型与所述帧内预测模式对应的至少一个上下文模型不同。

在一些实施例中,二值化单元 5002,还配置为在当前块的预测模式为帧内预测模式和/或帧间预测模式时,基于第一方式对所述第一语法标识信息的取值进行二值化处理,确定当前块的二元符号串。

在一些实施例中, 二值化单元 5002, 还配置为在所述当前块的预测模式为帧内预测模式时, 基于第一方式对第一语法标识信息的取值进行二值化处理, 确定当前块的第一二元符号串; 或者, 在当前块的预测模式为帧间预测模式时, 基于第二方式对第一语法标识信息的取值进行二值化处理, 确定当前块的第二二元符号串; 其中, 第一方式与第二方式不同。

5 在一些实施例中, 第一方式为定长编码方式, 第二方式为截断一元码方式。

在一些实施例中, 第一确定单元 5001, 还配置为若当前块不使用第一变换模式, 则确定第一语法标识信息的取值为第一值; 若当前块使用第一变换模式, 则确定第一语法标识信息的取值为第二值。

在一些实施例中, 参见图 50, 编码器 500 还可以包括变换单元 5004, 其中:

第一确定单元 5001, 还配置为在所述当前块使用第一变换模式时, 确定当前块的变换核;

10 变换单元 5004, 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 5003, 还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。

15 在一些实施例中, 第一确定单元 5001, 还配置为确定当前块的变换核组; 以及根据变换核组与第一语法标识信息的取值, 确定当前块的变换核; 其中, 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用于指示当前块的变换核在所述变换核组的编号。

在一些实施例中, 第一确定单元 5001, 还配置为在当前块的预测模式为帧内预测模式时, 确定变换核组中包括 M 个候选变换核; 或者, 在当前块的预测模式为帧间预测模式时, 确定变换核组中包括 N 个候选变换核; 其中, M、N 均为正整数, 且 M 大于或等于 N。

20 在一些实施例中, 第一确定单元 5001, 还配置为在 M 大于 N 时, 将 N 个候选变换核设置为 M 个候选变换核的一个子集; 或者, 在 M 等于 N 时, 将 N 个候选变换核设置为与 M 个候选变换核中至少部分不同。

在一些实施例中, 第一确定单元 5001, 还配置为确定当前块的至少一个纹理特征; 以及根据至少一个纹理特征进行变换核训练, 确定当前块的变换核组。

25 在一些实施例中, 第一确定单元 5001, 还配置为确定当前块的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块是否使用多变换选择模式; 编码单元 5003, 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 5001, 还配置为在当前块不使用多变换选择模式且当前块满足第一变换模式的使用条件时, 执行确定第一语法标识信息的取值的步骤。

30 在一些实施例中, 第一确定单元 5001, 还配置为在当前块不使用第一变换模式且当前块满足多变换选择模式的使用条件时, 执行确定当前块的第二语法标识信息的取值的步骤。

在一些实施例中, 第一确定单元 5001, 还配置为确定当前块的第三语法标识信息的取值; 其中, 第三语法标识信息用于指示当前块是否使用不可分离基础变换模式; 编码单元 5003, 还配置为对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。

35 在一些实施例中, 第一确定单元 5001, 还配置为确定当前块的第四语法标识信息的取值; 其中, 第四语法标识信息用于指示当前块是否使用低频不可分离变换模式; 编码单元 5003, 还配置为对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。

在一些实施例中, 参见图 50, 编码器 500 还可以包括量化单元 5005, 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 5003, 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。

40 在一些实施例中, 二值化单元 5002, 还配置为在当前块的变换系数中的最后一个非零系数位置满足预设条件时, 执行确定第一语法标识信息的取值, 并对第一语法标识信息的取值进行二值化处理, 确定当前块的第二元符号串的步骤。

45 在一些实施例中, 第一确定单元 5001, 还配置为在当前块的预测模式为帧内预测模式时, 当前块的变换系数中所有颜色分量的最后一个非零系数位置大于或等于第一阈值; 或者, 在当前块的预测模式为帧间预测模式时, 当前块的变换系数中亮度分量的最后一个非零系数位置大于或等于第一阈值。

在一些实施例中, 第一确定单元 5001, 还配置为在当前块的划分类型为双树划分时, 当前块的变换系数中所有颜色分量的最后一个非零系数位置大于或等于第一阈值; 或者, 在当前块的划分类型为单树划分时, 当前块的变换系数中亮度分量的最后一个非零系数位置大于或等于第一阈值。

50 在一些实施例中, 第一确定单元 5001, 还配置为在当前块的预测模式为帧内预测模式时, 当前块的变换系数中所有颜色分量的最后一个非零系数位置小于或等于第二阈值; 或者, 在当前块的预测模式为帧间预测模式时, 当前块的变换系数中亮度分量的最后一个非零系数位置小于或等于第二阈值。

在一些实施例中, 第一确定单元 5001, 还配置为在当前块的划分类型为双树划分时, 当前块的变

换系数中所有颜色分量的最后一个非零系数位置小于或等于第二阈值；或者，在当前块的划分类别为单树划分时，当前块的变换系数中亮度分量的最后一个非零系数位置小于或等于第二阈值。

在一些实施例中，第一确定单元 5001，还配置为根据当前块的尺寸参数，确定第二阈值。

5 在一些实施例中，第一确定单元 5001，还配置为将第二阈值设置为当前块的变换系数中系数个数最大值与一的差值。

在一些实施例中，第一确定单元 5001，还配置为确定当前块的预测块；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，变换单元 5004，还配置为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，确定当前块的变换系数。

10 在一些实施例中，变换单元 5004，还配置为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。

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。

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

因此，本申请实施例提供了一种计算机可读存储介质，应用于编码器 500，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。

25 基于编码器 500 的组成以及计算机可读存储介质，参见图 51，其示出了本申请实施例提供的编码器 500 的具体硬件结构示意图。如图 51 所示，编码器 500 可以包括：第一通信接口 5101、第一存储器 5102 和第一处理器 5103；各个组件通过第一总线系统 5104 耦合在一起。可理解，第一总线系统 5104 用于实现这些组件之间的连接通信。第一总线系统 5104 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 51 中将各种总线都标为第一总线系统 5104。其中，

30 第一通信接口 5101，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 5102，用于存储能够在第一处理器 5103 上运行的计算机程序；

第一处理器 5103，用于在运行所述计算机程序时，执行：

确定当前块的预测模式；根据当前块的预测模式，确定至少一个上下文模型；确定第一语法标识信息的取值，并对第一语法标识信息的取值进行二值化处理，确定当前块的二元符号串；其中，二元符号串包括至少一个二元符号；根据至少一个上下文模型对二元符号串中的至少一个二元符号分别进行编码处理，将所得到的编码比特写入码流。

可以理解，本申请实施例中的第一存储器 5102 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, SLD RAM）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 5102 旨在包括但不限于这些和任意其它适合类型的存储器。

而第一处理器 5103 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 5103 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 50 5103 可以是通用处理器、数字信号处理器（Digital Signal Processor, DSP）、专用集成电路（Application Specific Integrated Circuit, ASIC）、现成可编程门阵列（Field Programmable Gate Array, FPGA）或者其他可编程逻辑器件、分立门或者晶体管逻辑器件、分立硬件组件。可以实现或者执行本申请实施例中的

公开的各方法、步骤及逻辑框图。通用处理器可以是微处理器或者该处理器也可以是任何常规的处理器等。结合本申请实施例所公开的方法的步骤可以直接体现为硬件译码处理器执行完成，或者用译码处理器中的硬件及软件模块组合执行完成。软件模块可以位于随机存储器，闪存、只读存储器，可编程只读存储器或者电可擦写可编程存储器、寄存器等本领域成熟的存储介质中。该存储介质位于第一存储器 5102，第一处理器 5103 读取第一存储器 5102 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 5103 还配置为在运行所述计算机程序时，执行前述实施例中任一项所述的方法。

本实施例提供了一种编码器，首先根据当前块的预测模式来确定至少一个上下文模型，然后根据所确定的至少一个上下文模型对第一语法标识信息对应的二元符号串进行编码处理。也就是说，针对帧内预测模式或者帧间预测模式可以选择不同的上下文模型，使得能够对帧间预测模式下 LFNST/NSPT 的二元符号串使用符合其分布规律的编码方法，从而不仅可以提高压缩效率，而且还可以提升编解码性能。

基于前述实施例相同的发明构思，参见图 52，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 52 所示，该解码器 520 可以包括第二确定单元 5201、解码单元 5202、反二值化单元 5203 和反变换单元 5204，其中：

第二确定单元 5201，配置为确定当前块的预测模式；以及根据当前块的预测模式，确定至少一个上下文模型；

解码单元 5202，配置为根据至少一个上下文模型解码码流，确定当前块的二元符号串；其中，二元符号串包括至少一个二元符号；

反二值化单元 5203，配置为对二元符号串进行反二值化处理，确定第一语法标识信息的取值；

反变换单元 5204，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，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，当前块的预测模式包括：帧间预测模式和/或帧内预测模式；相应地，第二确定单元 5201，还配置为确定帧间预测模式对应的至少一个上下文模型与所述帧内预测模式对应的至少一个上下文模型不同。

在一些实施例中，反二值化单元 5203，还配置为在当前块的预测模式为帧内预测模式和/或帧间预测模式时，基于第一方式对所述二元符号串进行反二值化处理，确定第一语法标识信息的取值。

在一些实施例中，反二值化单元 5203，还配置为在当前块的预测模式为帧内预测模式时，基于第一方式对二元符号串进行反二值化处理，确定第一语法标识信息的取值；或者，在当前块的预测模式为帧间预测模式时，基于第二方式对所述二元符号串进行反二值化处理，确定第一语法标识信息的取值；其中，第一方式与第二方式不同。

在一些实施例中，第二确定单元 5201，还配置为在所述当前块的预测模式为帧内预测模式时，根据至少一个第一上下文模型和定长解码方式解码码流，确定当前块的第一二元符号串；或者，在当前块的预测模式为帧间预测模式时，根据至少一个第二上下文模型和截断一元码方式解码码流，确定当前块的第二二元符号串；其中，所述至少一个第一上下文模型与所述至少一个第二上下文模型不同。

在一些实施例中，第二确定单元 5201，还配置为确定当前块的变换核组；以及根据变换核组与第一语法标识信息的取值，确定当前块的变换核；其中，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用于指示当前块的变换核在变换核组的编号。

在一些实施例中，第二确定单元 5201，还配置为在当前块的预测模式为帧内预测模式时，确定变换核组中包括 M 个候选变换核；或者，在当前块的预测模式为帧间预测模式时，确定变换核组中包括 N 个候选变换核；其中，M、N 均为正整数，且 M 大于或等于 N。

在一些实施例中，第二确定单元 5201，还配置为在 M 大于 N 时，将 N 个候选变换核设置为 M 个候选变换核的一个子集；或者，在 M 等于 N 时，将 N 个候选变换核设置为与 M 个候选变换核中至少部分不同。

在一些实施例中，第二确定单元 5201，还配置为确定当前块的至少一个纹理特征；根据至少一个纹理特征进行变换核训练，确定所述当前块的变换核组。

在一些实施例中, 解码单元 5202, 还配置为解码码流, 确定当前块的第二语法标识信息的取值; 反二值化单元 5203, 还配置为在第二语法标识信息指示当前块不使用多变换选择模式且所述当前块满足第一变换模式的使用条件时, 根据至少一个上下文模型解码码流, 确定当前块的二元符号串; 以及对二元符号串进行反二值化处理, 确定第一语法标识信息的取值的步骤。

- 5 在一些实施例中, 解码单元 5202, 还配置为在第一语法标识信息指示当前块不使用第一变换模式且当前块满足多变换选择模式的使用条件时, 解码码流, 确定当前块的第二语法标识信息的取值; 反变换单元 5204, 还配置为在第二语法标识信息指示当前块使用多变换选择模式时, 确定当前块的多个变换核, 并根据多个变换核对所述当前块的变换系数进行反变换, 确定当前块的残差块。

- 10 在一些实施例中, 第二确定单元 5201, 还配置为根据至少一个第三上下文模型解码码流, 确定当前块的第三二元符号串; 反二值化单元 5203, 还配置为对第三二元符号串进行反二值化处理, 确定第三语法标识信息的取值;

反变换单元 5204, 还配置为在第三语法标识信息指示当前块使用不可分离基础变换模式时, 确定当前块的变换核, 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。

- 15 在一些实施例中, 第二确定单元 5201, 还配置为根据至少一个第四上下文模型解码码流, 确定当前块的第四二元符号串; 反二值化单元 5203, 还配置为对第四二元符号串进行反二值化处理, 确定第四语法标识信息的取值;

反变换单元 5204, 还配置为在第四语法标识信息指示当前块使用低频不可分离变换模式时, 确定当前块的变换核, 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 在一些实施例中, 参见图 52, 解码器 500 还包括反量化单元 5005, 其中:

解码单元 5202, 还配置为解码码流, 确定所述当前块的量化系数;

反量化单元 5005, 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。

在一些实施例中, 反二值化单元 5203, 还配置为在当前块的变换系数中的最后一个非零系数位置满足预设条件时, 执行根据所述至少一个上下文模型解码码流, 确定当前块的二元符号串; 以及对二元符号串进行反二值化处理, 确定第一语法标识信息的取值的步骤。

- 25 在一些实施例中, 第二确定单元 5201, 还配置为在当前块的预测模式为帧内预测模式时, 当前块的变换系数中所有颜色分量的最后一个非零系数位置大于或等于第一阈值; 或者, 在当前块的预测模式为帧间预测模式时, 当前块的变换系数中亮度分量的最后一个非零系数位置大于或等于第一阈值。

- 30 在一些实施例中, 第二确定单元 5201, 还配置为在当前块的划分类型为双树划分时, 当前块的变换系数中颜色分量的最后一个非零系数位置大于或等于第一阈值; 或者, 在当前块的划分类型为单树划分时, 当前块的变换系数中亮度分量的最后一个非零系数位置大于或等于第一阈值。

在一些实施例中, 第二确定单元 5201, 还配置为在当前块的预测模式为帧内预测模式时, 当前块的变换系数中所有颜色分量的最后一个非零系数位置小于或等于第二阈值; 或者, 在当前块的预测模式为帧间预测模式时, 当前块的变换系数中亮度分量的最后一个非零系数位置小于或等于第二阈值。

- 35 在一些实施例中, 第二确定单元 5201, 还配置为在当前块的划分类型为双树划分时, 当前块的变换系数中所有颜色分量的最后一个非零系数位置小于或等于第二阈值; 或者, 在当前块的划分类型为单树划分时, 当前块的变换系数中亮度分量的最后一个非零系数位置小于或等于第二阈值。

在一些实施例中, 第二确定单元 5201, 还配置为根据当前块的尺寸参数, 确定第二阈值。

在一些实施例中, 第二确定单元 5201, 还配置为将第二阈值设置为当前块的变换系数中系数个数最大值与一的差值。

- 40 在一些实施例中, 第二确定单元 5201, 还配置为确定当前块的预测块; 以及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。

- 45 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。

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 520, 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。

- 50 基于解码器 520 的组成以及计算机可读存储介质, 参见图 53, 其示出了本申请实施例提供的解码器 520 的具体硬件结构示意图。如图 53 所示, 解码器 520 可以包括: 第二通信接口 5301、第二存储器 5302 和第二处理器 5303; 各个组件通过第二总线系统 5304 耦合在一起。可理解, 第二总线系统 5304

用于实现这些组件之间的连接通信。第二总线系统 5304 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 53 中将各种总线都标为第二总线系统 5304。其中，

第二通信接口 5301，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 5302，用于存储能够在第二处理器 5303 上运行的计算机程序；

5 第二处理器 5303，用于在运行所述计算机程序时，执行：

确定当前块的预测模式；根据当前块的预测模式，确定至少一个上下文模型；根据至少一个上下文模型解码码流，确定当前块的二元符号串；其中，二元符号串包括至少一个二元符号；对二元符号串进行反二值化处理，确定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，确定当前块的变换核，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。

10 可选地，作为另一个实施例，第二处理器 5303 还配置为在运行所述计算机程序时，执行前述实施例中任一项所述的方法。

可以理解，第二存储器 5302 与第一存储器 5102 的硬件功能类似，第二处理器 5303 与第一处理器 5103 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，首先根据当前块的预测模式来确定至少一个上下文模型，然后根据所确定的至少一个上下文模型对第一语法标识信息对应的二元符号串进行解码处理。也就是说，针对帧内预测模式或者帧间预测模式可以选择不同的上下文模型，使得能够对帧间预测模式下 LFNST/NSPT 的二元符号串使用符合其分布规律的解码方法，从而不仅可以提高压缩效率，而且还可以提升编解码性能。

在本申请的再一实施例中，参见图 54，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 54 所示，编解码系统 540 可以包括编码器 5401 和解码器 5402。

20 在本申请实施例中，编码器 5401 可以是前述实施例中任一项所述的编码器，解码器 5402 可以是前述实施例中任一项所述的解码器。

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

上述本申请实施例序号仅仅为了描述，不代表实施例的优劣。

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。

30 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。

35 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

本申请实施例中，在编码端，确定当前块的预测模式；根据当前块的预测模式，确定至少一个上下文模型；确定第一语法标识信息的取值，并对第一语法标识信息的取值进行二值化处理，确定当前块的二元符号串；其中，二元符号串包括至少一个二元符号；根据至少一个上下文模型对二元符号串中的至少一个二元符号分别进行编码处理，将所得到的编码比特写入码流。在解码端，确定当前块的预测模式；根据当前块的预测模式，确定至少一个上下文模型；根据至少一个上下文模型解码码流，确定当前块的二元符号串；其中，二元符号串包括至少一个二元符号；对二元符号串进行反二值化处理，确定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，确定当前块的变换核，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，都是首先根据当前块的预测模式来确定至少一个上下文模型，然后根据所确定的至少一个上下文模型对第一语法标识信息对应的二元符号串进行编解码处理。也就是说，针对帧内预测模式或者帧间预测模式可以选择不同的上下文模型，使得能够对帧间预测模式下 LFNST/NSPT 的二元符号串使用符合其分布规律的编解码方法，从而不仅可以提高压缩效率，而且还可以提升编解码性能。

权利要求书

- 1、一种解码方法，应用于解码器，所述方法包括：
确定当前块的预测模式；
根据所述当前块的预测模式，确定至少一个上下文模型；
5 根据所述至少一个上下文模型解码码流，确定所述当前块的二元符号串；其中，所述二元符号串包括至少一个二元符号；
对所述二元符号串进行反二值化处理，确定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，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。
- 10 2、根据权利要求1所述的方法，其中，所述当前块的预测模式包括：帧间预测模式和/或帧内预测模式，所述方法还包括：
确定所述帧内预测模式对应的至少一个上下文模型与所述帧内预测模式对应的至少一个上下文模型不同。
- 15 3、根据权利要求2所述的方法，其中，所述对所述二元符号串进行反二值化处理，确定第一语法标识信息的取值，包括：
在所述当前块的预测模式为帧内预测模式和/或帧间预测模式时，基于第一方式对所述二元符号串进行反二值化处理，确定第一语法标识信息的取值。
- 4、根据权利要求2所述的方法，其中，所述对所述二元符号串进行反二值化处理，确定第一语法标识信息的取值，包括：
20 在所述当前块的预测模式为帧内预测模式时，基于第一方式对所述二元符号串进行反二值化处理，确定第一语法标识信息的取值；或者，
在所述当前块的预测模式为帧间预测模式时，基于第二方式对所述二元符号串进行反二值化处理，确定第一语法标识信息的取值；
其中，所述第一方式与所述第二方式不同。
- 25 5、根据权利要求1所述的方法，其中，所述根据所述至少一个上下文模型解码码流，确定所述当前块的二元符号串，包括：
在所述当前块的预测模式为帧内预测模式时，根据至少一个第一上下文模型和定长解码方式解码码流，确定所述当前块的第一二元符号串；或者，
在所述当前块的预测模式为帧间预测模式时，根据至少一个第二上下文模型和截断一元码方式解码
30 码流，确定所述当前块的第二二元符号串；
其中，所述至少一个第一上下文模型与所述至少一个第二上下文模型不同。
- 6、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核，包括：
确定所述当前块的变换核组；
根据所述变换核组与所述第一语法标识信息的取值，确定所述当前块的变换核；其中，在所述当前
35 块使用第一变换模式时，所述第一语法标识信息的取值还用于指示所述当前块的变换核在所述变换核组的编号。
- 7、根据权利要求6所述的方法，其中，所述确定所述当前块的变换核组，包括：
在所述当前块的预测模式为帧内预测模式时，确定所述变换核组中包括M个候选变换核；或者，
在所述当前块的预测模式为帧间预测模式时，确定所述变换核组中包括N个候选变换核；其中，
40 M、N均为正整数，且M大于或等于N。
- 8、根据权利要求7所述的方法，其中，所述方法还包括：
在M大于N时，将所述N个候选变换核设置为所述M个候选变换核的一个子集；
或者，
在M等于N时，将所述N个候选变换核设置为与所述M个候选变换核中至少部分不同。
- 45 9、根据权利要求6所述的方法，其中，所述确定当前块的变换核组，包括：
确定所述当前块的至少一个纹理特征；
根据所述至少一个纹理特征进行变换核训练，确定所述当前块的变换核组。
- 10、根据权利要求1至9中任一项所述的方法，其中，所述方法还包括：
解码码流，确定当前块的第二语法标识信息的取值；
50 在所述第二语法标识信息指示当前块不使用多变换选择模式且所述当前块满足第一变换模式的使

用条件时,执行根据所述至少一个上下文模型解码码流,确定所述当前块的二元符号串;以及对所述二元符号串进行反二值化处理,确定第一语法标识信息的取值的步骤。

11、根据权利要求1至9中任一项所述的方法,其中,所述方法还包括:

5 在所述第一语法标识信息指示当前块不使用第一变换模式且所述当前块满足多变换选择模式的使用条件时,解码码流,确定当前块的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块使用多变换选择模式时,确定所述当前块的多个变换核,并根据所述多个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。

12、根据权利要求1至9中任一项所述的方法,其中,所述方法还包括:

根据至少一个第三上下文模型解码码流,确定所述当前块的第三二元符号串;

10 对所述第三二元符号串进行反二值化处理,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所述当前块使用不可分离基础变换模式时,确定所述当前块的变换核,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。

13、根据权利要求1至9中任一项所述的方法,其中,所述方法还包括:

根据至少一个第四上下文模型解码码流,确定所述当前块的第四二元符号串;

15 对所述第四二元符号串进行反二值化处理,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所述当前块使用低频不可分离变换模式时,确定所述当前块的变换核,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。

14、根据权利要求1至9中任一项所述的方法,其中,所述方法还包括:

解码码流,确定所述当前块的量化系数;

20 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。

15、根据权利要求14所述的方法,其中,所述方法还包括:

在所述当前块的变换系数中的最后一个非零系数位置满足预设条件时,执行根据所述至少一个上下文模型解码码流,确定所述当前块的二元符号串;以及对所述二元符号串进行反二值化处理,确定第一语法标识信息的取值的步骤。

25 16、根据权利要求15所述的方法,其中,所述当前块的变换系数中的最后一个非零系数位置满足预设条件,包括:

在所述当前块的预测模式为帧内预测模式时,所述当前块的变换系数中所有颜色分量的最后一个非零系数位置大于或等于第一阈值;或者,

30 在所述当前块的预测模式为帧间预测模式时,所述当前块的变换系数中亮度分量的最后一个非零系数位置大于或等于第一阈值。

17、根据权利要求15所述的方法,其中,所述当前块的变换系数中的最后一个非零系数位置满足预设条件,包括:

在所述当前块的划分类型为双树划分时,所述当前块的变换系数中所有颜色分量的最后一个非零系数位置大于或等于第一阈值;或者,

35 在所述当前块的划分类型为单树划分时,所述当前块的变换系数中亮度分量的最后一个非零系数位置大于或等于第一阈值。

18、根据权利要求15所述的方法,其中,所述当前块的变换系数中的最后一个非零系数位置满足预设条件,包括:

40 在所述当前块的预测模式为帧内预测模式时,所述当前块的变换系数中所有颜色分量的最后一个非零系数位置小于或等于第二阈值;或者,

在所述当前块的预测模式为帧间预测模式时,所述当前块的变换系数中亮度分量的最后一个非零系数位置小于或等于第二阈值。

19、根据权利要求15所述的方法,其中,所述当前块的变换系数中的最后一个非零系数位置满足预设条件,包括:

45 在所述当前块的划分类型为双树划分时,所述当前块的变换系数中所有颜色分量的最后一个非零系数位置小于或等于第二阈值;或者,

在所述当前块的划分类型为单树划分时,所述当前块的变换系数中亮度分量的最后一个非零系数位置小于或等于第二阈值。

20、根据权利要求18或19所述的方法,其中,所述方法还包括:

50 根据所述当前块的尺寸参数,确定所述第二阈值。

21、根据权利要求18或19所述的方法,其中,所述方法还包括:

将所述第二阈值设置为所述当前块的变换系数中系数个数最大值与一的差值。

- 22、根据权利要求 1 至 21 中任一项所述的方法，其中，所述方法还包括：
确定所述当前块的预测块；
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。
- 23、一种编码方法，应用于编码器，所述方法包括：
- 5 确定当前块的预测模式；
根据所述当前块的预测模式，确定至少一个上下文模型；
确定第一语法标识信息的取值，并对所述第一语法标识信息的取值进行二值化处理，确定所述当前块的二元符号串；其中，所述二元符号串包括至少一个二元符号；
根据所述至少一个上下文模型对所述二元符号串中的至少一个二元符号分别进行编码处理，将所得
- 10 到的编码比特写入码流。
- 24、根据权利要求 23 所述的方法，其中，所述当前块的预测模式包括：帧间预测模式和/或帧内预测模式，所述方法还包括：
确定所述帧间预测模式对应的至少一个上下文模型与所述帧内预测模式对应的至少一个上下文模型不同。
- 15 25、根据权利要求 24 所述的方法，其中，所述对所述第一语法标识信息的取值进行二值化处理，确定所述当前块的二元符号串，包括：
在所述当前块的预测模式为帧内预测模式和/或帧间预测模式时，基于第一方式对所述第一语法标识信息的取值进行二值化处理，确定所述当前块的二元符号串。
- 26、根据权利要求 24 所述的方法，其中，所述对所述第一语法标识信息的取值进行二值化处理，
- 20 确定所述当前块的二元符号串，包括：
在所述当前块的预测模式为帧内预测模式时，基于第一方式对所述第一语法标识信息的取值进行二值化处理，确定所述当前块的第一二元符号串；或者，
在所述当前块的预测模式为帧间预测模式时，基于第二方式对所述第一语法标识信息的取值进行二值化处理，确定所述当前块的第二二元符号串；
- 25 其中，所述第一方式与所述第二方式不同。
- 27、根据权利要求 26 所述的方法，其中，所述第一方式为定长编码方式，所述第二方式为截断一元码方式。
- 28、根据权利要求 23 所述的方法，其中，所述确定第一语法标识信息的取值，包括：
若所述当前块不使用第一变换模式，则确定所述第一语法标识信息的取值为第一值；
- 30 若所述当前块使用第一变换模式，则确定所述第一语法标识信息的取值为第二值。
- 29、根据权利要求 23 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，确定所述当前块的变换核；
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
- 35 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。
- 30、根据权利要求 29 所述的方法，其中，所述确定所述当前块的变换核，包括：
确定所述当前块的变换核组；
根据所述变换核组与所述第一语法标识信息的取值，确定所述当前块的变换核；其中，在所述当前块使用第一变换模式时，所述第一语法标识信息的取值还用于指示所述当前块的变换核在所述变换核组
- 40 的编号。
- 31、根据权利要求 30 所述的方法，其中，所述确定所述当前块的变换核组，包括：
在所述当前块的预测模式为帧内预测模式时，确定所述变换核组中包括 M 个候选变换核；或者，
在所述当前块的预测模式为帧间预测模式时，确定所述变换核组中包括 N 个候选变换核；其中，M、N 均为正整数，且 M 大于或等于 N。
- 45 32、根据权利要求 31 所述的方法，其中，所述方法还包括：
在 M 大于 N 时，将所述 N 个候选变换核设置为所述 M 个候选变换核的一个子集；
或者，
在 M 等于 N 时，将所述 N 个候选变换核设置为与所述 M 个候选变换核中至少部分不同。
- 33、根据权利要求 30 所述的方法，其中，所述确定当前块的变换核组，包括：
- 50 确定所述当前块的至少一个纹理特征；
根据所述至少一个纹理特征进行变换核训练，确定所述当前块的变换核组。
- 34、根据权利要求 23 至 33 中任一项所述的方法，其中，所述方法还包括：

确定当前块的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所述当前块是否使用多变换选择模式；

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。

35、根据权利要求 34 所述的方法，其中，所述方法还包括：

- 5 在所述当前块不使用多变换选择模式且所述当前块满足第一变换模式的使用条件时，执行确定第一语法标识信息的取值的步骤。

36、根据权利要求 23 至 33 中任一项所述的方法，其中，所述方法还包括：

在所述当前块不使用第一变换模式且所述当前块满足多变换选择模式的使用条件时，执行确定当前块的第二语法标识信息的取值的步骤。

- 10 37、根据权利要求 23 至 33 中任一项所述的方法，其中，所述方法还包括：

确定当前块的第三语法标识信息的取值；其中，所述第三语法标识信息用于指示所述当前块是否使用不可分离基础变换模式；

对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。

38、根据权利要求 23 至 33 中任一项所述的方法，其中，所述方法还包括：

- 15 确定当前块的第四语法标识信息的取值；其中，所述第四语法标识信息用于指示所述当前块是否使用低频不可分离变换模式；

对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。

39、根据权利要求 29 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：

- 20 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。

40、根据权利要求 29 所述的方法，其中，所述方法还包括：

在所述当前块的变换系数中的最后一个非零系数位置满足预设条件时，执行确定第一语法标识信息的取值，并对所述第一语法标识信息的取值进行二值化处理，确定所述当前块的二元符号串的步骤。

- 25 41、根据权利要求 40 所述的方法，其中，所述当前块的变换系数中的最后一个非零系数位置满足预设条件，包括：

在所述当前块的预测模式为帧内预测模式时，所述当前块的变换系数中所有颜色分量的最后一个非零系数位置大于或等于第一阈值；或者，

- 30 在所述当前块的预测模式为帧间预测模式时，所述当前块的变换系数中亮度分量的最后一个非零系数位置大于或等于第一阈值。

42、根据权利要求 40 所述的方法，其中，所述当前块的变换系数中的最后一个非零系数位置满足预设条件，包括：

在所述当前块的划分类型为双树划分时，所述当前块的变换系数中所有颜色分量的最后一个非零系数位置大于或等于第一阈值；或者，

- 35 在所述当前块的划分类型为单树划分时，所述当前块的变换系数中亮度分量的最后一个非零系数位置大于或等于第一阈值。

43、根据权利要求 40 所述的方法，其中，所述当前块的变换系数中的最后一个非零系数位置满足预设条件，包括：

- 40 在所述当前块的预测模式为帧内预测模式时，所述当前块的变换系数中所有颜色分量的最后一个非零系数位置小于或等于第二阈值；或者，

在所述当前块的预测模式为帧间预测模式时，所述当前块的变换系数中亮度分量的最后一个非零系数位置小于或等于第二阈值。

44、根据权利要求 40 所述的方法，其中，所述当前块的变换系数中的最后一个非零系数位置满足预设条件，包括：

- 45 在所述当前块的划分类型为双树划分时，所述当前块的变换系数中所有颜色分量的最后一个非零系数位置小于或等于第二阈值；或者，

在所述当前块的划分类型为单树划分时，所述当前块的变换系数中亮度分量的最后一个非零系数位置小于或等于第二阈值。

- 46、根据权利要求 43 或 44 所述的方法，其中，所述方法还包括：

- 50 根据所述当前块的尺寸参数，确定所述第二阈值。

46、根据权利要求 43 或 44 所述的方法，其中，所述方法还包括：

将所述第二阈值设置为所述当前块的变换系数中系数个数最大值与一的差值。

- 47、根据权利要求 29 所述的方法，其中，所述确定所述当前块的残差块，包括：
确定所述当前块的预测块；
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。
- 48、根据权利要求 29 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，
5 确定所述当前块的变换系数，包括：
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。
- 49、根据权利要求 29 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：
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；
10 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。
- 50、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、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；
15 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，所述第二语法标识信息用于指示所述当前块是否使用多变换选择模式，所述第三语法标识信息用于指示所述当前块是否使用不可分离基础变换模式，所述第四语法标识信息用于指示所述当前块是否使用低频不可分离变换模式。
- 51、一种编码器，所述编码器包括第一确定单元、二值化单元和编码单元，其中：
所述第一确定单元，配置为确定当前块的预测模式；以及根据所述当前块的预测模式，确定至少一个
20 个上下文模型；
所述二值化单元，配置为确定第一语法标识信息的取值，并对所述第一语法标识信息的取值进行二值化处理，确定所述当前块的二元符号串；其中，所述二元符号串包括至少一个二元符号；
所述编码单元，配置为根据所述至少一个上下文模型对所述二元符号串中的至少一个二元符号分别进行编码处理，将所得到的编码比特写入码流。
- 52、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 23 至 49 中任一项所述的方法。
- 53、一种解码器，所述解码器包括第二确定单元、解码单元、反二值化单元和反变换单元，其中：
所述第二确定单元，配置为确定当前块的预测模式；以及根据所述当前块的预测模式，确定至少一个
30 个上下文模型；
所述解码单元，配置为根据所述至少一个上下文模型解码码流，确定所述当前块的二元符号串；其中，所述二元符号串包括至少一个二元符号；
所述反二值化单元，配置为对所述二元符号串进行反二值化处理，确定第一语法标识信息的取值；
所述反变换单元，配置为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
35 当前块的变换核，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。
- 54、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 22 中任一项所述的方法。
- 55、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 22 中任一项所述的方法、或者被第二处理器执行时实现如
40 权利要求 23 至 49 中任一项所述的方法。