



# (12) 发明专利申请

(10) 申请公布号 CN 117296323 A

(43) 申请公布日 2023. 12. 26

(21) 申请号 202280030418.X

(74) 专利代理机构 北京市柳沈律师事务所  
11105

(22) 申请日 2022.04.21

专利代理师 张亮 刘文洁

(66) 本国优先权数据

PCT/CN2021/088961 2021.04.22 CN

(51) Int.Cl.

H04N 19/463 (2006.01)

(85) PCT国际申请进入国家阶段日

2023.10.23

(86) PCT国际申请的申请数据

PCT/CN2022/088116 2022.04.21

(87) PCT国际申请的公布数据

W02022/222985 EN 2022.10.27

(71) 申请人 抖音视界有限公司

地址 100041 北京市石景山区实兴大街30  
号院3号楼2层B-0035房间

申请人 字节跳动有限公司

(72) 发明人 崔静 刘鸿彬 张莉 王悦

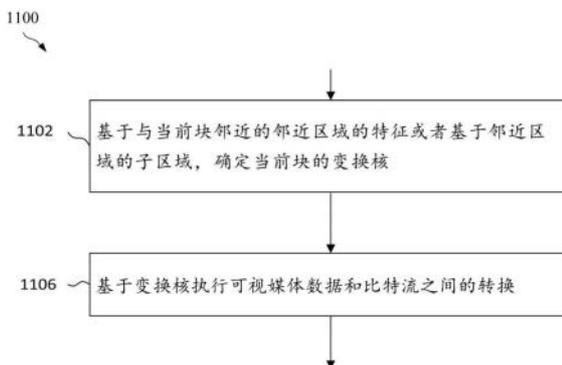
权利要求书2页 说明书21页 附图12页

(54) 发明名称

基于特征的变换选择

(57) 摘要

在用于处理可视数据的方法中,基于与当前块邻近的邻近区域的特征或者基于邻近区域的子区域,确定可视媒体的当前块的变换核。基于变换核执行可视媒体和可视媒体的比特流之间的转换。



1. 一种用于处理可视数据的方法,包括:  
基于与当前块邻近的邻近区域的特征或者基于所述邻近区域的子区域,确定可视媒体的所述当前块的变换核,以及  
基于所述变换核执行所述可视媒体和所述可视媒体的比特流之间的转换。
2. 根据权利要求1所述的方法,其中,所述邻近区域与所述当前块紧邻。
3. 根据权利要求1所述的方法,其中,所述邻近区域不与所述当前块紧邻。
4. 根据权利要求1所述的方法,其中,所述邻近区域是上邻近区域。
5. 根据权利要求1所述的方法,其中,所述邻近区域是左邻近区域。
6. 根据权利要求1所述的方法,其中,所述邻近区域是左上邻近区域。
7. 根据权利要求1所述的方法,其中,所述邻近区域是右上邻近区域。
8. 根据权利要求1-7中任一项所述的方法,其中所述邻近区域的尺寸是预先确定的。
9. 根据权利要求1-7中任一项所述的方法,其中,根据所述当前块的尺寸来分配所述邻近区域的尺寸。
10. 根据权利要求1-7中任一项所述的方法,其中,根据所述当前块的帧内模式来分配所述邻近区域的尺寸。
11. 根据权利要求1-10中任一项所述的方法,其中,基于所述子区域的左上样点,所述子区域被包括在所述邻近区域中。
12. 根据权利要求1-11中任一项所述的方法,还包括从所述子区域中的重建样点中提取所述特征。
13. 根据权利要求1-12中任一项所述的方法,其中所述特征基于所述子区域的分类信息。
14. 根据权利要求13所述的方法,其中,所述分类信息是:所述子区域是噪声区域、平滑区域或边缘区域。
15. 根据权利要求13-14中任一项所述的方法,其中所述特征基于一个区域中的第一样点和围绕所述第一样点的第二样点之间的差、一组样点的平均值或边缘信息。
16. 根据权利要求14所述的方法,还包括:  
基于所述子区域中跳跃点的标识将所述子区域分类为所述噪声区域。
17. 根据权利要求14所述的方法,还包括:  
基于所述子区域中正常点的标识将所述子区域分类为所述平滑区域。
18. 根据权利要求14所述的方法,还包括:  
基于所述子区域中边缘像素的标识将所述子区域分类为所述边缘区域。
19. 根据权利要求1-12中任一项所述的方法,其中所述特征基于所述子区域的辅助信息。
20. 根据权利要求1-19中任一项所述的方法,其中所述辅助信息是块尺寸。
21. 根据权利要求1-19中任一项所述的方法,其中所述辅助信息是预测模式。
22. 根据权利要求1-21中任一项所述的方法,其中,所述变换核是离散余弦变换类型二(DCT2)。
23. 根据权利要求1-21中任一项所述的方法,其中所述变换核是离散余弦变换类型七(DCT7)。

24. 根据权利要求1-23中任一项所述的方法,还包括:

基于所述比特流中的块维度或信令通知来确定是否使用基于特征的变换选择 (FTS)。

25. 根据权利要求1-23中任一项所述的方法,还包括:

基于所述比特流中的块维度或信令通知来确定如何使用基于特征的变换选择 (FTS)。

26. 根据权利要求1-25中任一项所述的方法,其中,所述转换包括将所述可视媒体编码成所述比特流。

27. 根据权利要求1-25中任一项所述的方法,其中所述转换包括从所述比特流中解码所述可视媒体。

28. 一种非暂时性计算机可读介质,包含供视频编解码设备使用的计算机程序产品,所述计算机程序产品包含存储在所述非暂时性计算机可读介质上的计算机可执行指令,使得所述计算机可执行指令在由处理器执行时使所述视频编解码设备执行权利要求1-27中任一项所述的方法。

29. 一种用于处理可视数据的装置,包括:处理器;以及其上具有指令的非暂时性存储器,其中所述指令在由所述处理器执行时使得所述处理器实施权利要求1-27中任一项所述的方法。

30. 一种存储有可视媒体的比特流的非暂时性计算机可读记录介质,所述比特流由视频处理装置执行的方法生成,其中所述方法包括:

基于与当前块邻近的邻近区域的特征或基于所述邻近区域的子区域来确定所述当前块的变换核;以及

基于所述变换核生成所述比特流。

31. 一种用于存储视频的比特流的方法,包括:

基于与当前块邻近的邻近区域的特征或基于所述邻近区域的子区域来确定所述当前块的变换核;

基于所述变换核生成所述比特流;以及

将所述比特流存储在非暂时性计算机可读记录介质中。

## 基于特征的变换选择

[0001] 相关申请的交叉引用

[0002] 本专利申请要求Jing Cui等人于2021年4月22日提交的标题为“基于特征的变换选择”的国际申请No. PCT/CN2021/088961的权益,该申请通过引用并入本文。

### 技术领域

[0003] 本专利文档涉及文件格式的数字音频视频媒体信息的生成、存储和消耗。

### 背景技术

[0004] 数字视频在互联网和其他数字通信网络上占据着最大的带宽使用。随着能够接收和显示视频的连接用户设备数量的增加,预计数字视频使用的带宽需求将继续增长。

### 发明内容

[0005] 第一方面涉及一种用于处理视频数据的方法,包括:基于与当前块邻近的邻近区域的特征或基于邻近区域的子区域来确定当前块的变换核;以及基于变换核执行可视媒体数据和比特流之间的转换。

[0006] 可选地,邻近区域与当前块紧邻。

[0007] 可选地,邻近区域不与当前块紧邻。

[0008] 可选地,邻近区域是上邻近区域。

[0009] 可选地,邻近区域是左邻近区域。

[0010] 可选地,邻近区域是左上邻近区域。

[0011] 可选地,邻近区域是右上邻近区域。

[0012] 可选地,邻近区域的尺寸是预先确定的。

[0013] 可选地,根据当前块的尺寸来分配邻近区域的尺寸。

[0014] 可选地,根据当前块的帧内模式来分配邻近区域的尺寸。

[0015] 可选地,基于子区域的左上样点,子区域被包括在邻近区域中。

[0016] 可选地,该方法还包括从子区域中的重建样点中提取特征。

[0017] 可选地,特征基于子区域的分类信息。

[0018] 可选地,分类信息是:子区域是噪声区域、平滑区域或边缘区域。

[0019] 可选地,特征基于一个区域中的第一样点和围绕第一样点的第二样点之间的差、一组样点的平均值或边缘信息。

[0020] 可选地,该方法还包括基于子区域中跳跃点的标识将子区域分类为噪声区域。

[0021] 可选地,该方法还包括基于子区域中正常点的标识将子区域分类为平滑区域。

[0022] 可选地,该方法还包括基于子区域中边缘像素的标识将子区域分类为边缘区域。

[0023] 可选地,特征基于子区域的辅助信息。

[0024] 可选地,辅助信息是块尺寸。

[0025] 可选地,辅助信息是预测模式。

- [0026] 可选地,变换核是离散余弦变换类型二(DCT2)。
- [0027] 可选地,变换核是离散余弦变换类型七(DCT7)。
- [0028] 可选地,该方法还包括基于比特流中的块维度或信令通知来确定是否使用基于特征的变换选择(FTS)。
- [0029] 可选地,该方法还包括基于比特流中的块维度或信令通知来确定如何使用基于特征的变换选择(FTS)。
- [0030] 第二方面涉及一种非暂时性计算机可读介质,包含供视频编解码设备使用的计算机程序产品,计算机程序产品包含存储在非暂时性计算机可读介质上的计算机可执行指令,使得计算机可执行指令在由处理器执行时使视频编解码设备执行前述方面中任一方面的方法。
- [0031] 第三方面涉及一种用于处理视频数据的装置,包括:处理器;以及其上具有指令的非暂时性存储器,其中指令在由处理器执行时使得处理器执行前述方面中任一方面的方法。
- [0032] 第四方面涉及一种存储有视频的比特流的非暂时性计算机可读记录介质,比特流由视频处理装置执行的方法生成的,其中方法包括:基于与当前块邻近的邻近区域的特征或基于邻近区域的子区域来确定当前块的变换核;以及基于变换核执行可视媒体数据和比特流之间的转换。
- [0033] 第五方面涉及一种用于存储视频的比特流的方法,包括:基于与当前块邻近的邻近区域的特征或基于邻近区域的子区域来确定当前块的变换核;以及基于变换核执行可视媒体数据和比特流之间的转换。

## 附图说明

- [0034] 为了更全面地理解本公开内容,现结合附图和详细说明参考以下简要说明,其中相同的参考数字代表相同的部件。
- [0035] 图1是VVC的编码器框图。
- [0036] 图2是HEVC中的变换和逆变换的示意图。
- [0037] 图3是联合探索模型(JEM)中二次变换的示意图。
- [0038] 图4是示例低频不可分变换(LFNST)的示意图。
- [0039] 图5是正逆简化变换(RT)的示意图。
- [0040] 图6是具有 $16 \times 48$ 矩阵的正LFNST  $8 \times 8$ 过程的示例的示意图。
- [0041] 图7是用于支持LFNST的信令通知的示例扫描机制的示意图。
- [0042] 图8是当前块及其邻近区域和非紧邻区域的示意图。
- [0043] 图9是示出可实现本公开的各种技术的示例视频处理系统的框图。
- [0044] 图10是示例视频处理装置的框图。
- [0045] 图11是视频处理的示例方法的流程图。
- [0046] 图12是可利用本公开的技术的示例视频编解码系统的框图。
- [0047] 图13是示出视频编码器的示例的框图,该视频编码器可为图12所示系统中的视频编码器。
- [0048] 图14是示出视频解码器的示例的框图,该视频解码器可为图12所示系统中的视频

解码器。

[0049] 图15是示例编码器的示意图。

### 具体实施方式

[0050] 首先应理解,尽管下文提供了一个或多个实施例的说明性实现方式,但可使用任何数量的技术实现所公开的系统 and/或方法,无论是当前已知的还是待开发的。本公开不应以任何方式限于以下示出的示例性实现方式、附图和技术,包括在此示出和描述的示例性设计和实现方式,而是可以在所附权利要求及其等同物的全部范围内进行修改。

[0051] 多功能视频编解码(VVC),也称为H.266,在某些描述中使用术语仅是为了便于理解,而不是为了限制所公开技术的范围。因此,本文描述的技术也适用于其他视频编解码器协议和设计。在本文档中,相对于VVC规范或国际标准化组织(ISO)基本媒体文件格式(ISOBMFF)文件格式规范,文本的编辑更改通过粗体斜体表示取消的文本和粗体下划线表示添加的文本。

[0052] 本公开涉及图像/视频编解码技术。更具体地,涉及视频编解码中的变换编解码。可以应用于现有的视频编解码标准,如HEVC,或VVC(多功能视频编解码),或即将完成的标准,也可以应用于未来的视频编解码标准或视频编解码器。

[0053] 视频编解码标准主要是随着众所周知的ITU-T和ISO/IEC标准的发展而演变。ITU-T制定了H.261和H.263,ISO/IEC制定了MPEG-1和MPEG-4视觉,并且这两个组织联合制定了H.262/MPEG-2视频和H.264/MPEG-4高级视频编解码(AVC)和H.265/HEVC标准。自H.262以来,视频编解码标准是基于利用了时域预测加变换编解码的混合视频编解码结构。

[0054] 图1是VVC的编码器框图100的示例。该编码器框图100包含三个环路滤波块:去块滤波器(DF)、样点自适应偏移(SAO)和ALF。与使用预定义滤波器的DF不同,SAO和ALF利用当前图片的原始样点,分别通过添加偏移和通过应用有限脉冲响应(FIR)滤波器来减少原始样点和重建样点之间的均方误差,其中编解码辅助信息信令通知偏移和滤波器系数。ALF位于每个图片的最后一个处理阶段,并且可以被视为试图捕捉和修复前几个阶段产生的伪影的工具。

[0055] 离散正弦变换族包括众所周知的离散傅立叶、余弦、正弦和Karhunen-Loeve(在一阶马尔可夫条件下)变换。在所有成员中,有8种基于余弦函数的变换(DCT-I、DCT-II、……、DCT-VIII)和8种基于正弦函数的变换(DST-I、DST-II、……、DST-VIII)。离散余弦和正弦变换的这些变体源自它们对应的对称周期序列的不同对称性。在所提出的方法中使用的选定类型的DCT和DST的变换基函数如表1所示。

变换类型	基函数 $T_i(j)$ , $i, j=0, 1, \dots, N-1$
[0056] DCT-II	$T_i(j) = \omega_0 \cdot \sqrt{\frac{2}{N}} \cdot \cos\left(\frac{\pi \cdot i \cdot (2j+1)}{2N}\right) \text{ where } \omega_0 = \begin{cases} \sqrt{\frac{2}{N}} & i = 0 \\ 1 & i \neq 0 \end{cases}$

[0057]	$T_i(j) = \omega_0 \cdot \omega_1 \cdot \sqrt{\frac{2}{2N-1}} \cdot \cos\left(\frac{2\pi \cdot i \cdot j}{2N-1}\right),$ $\text{where } \omega_0 = \begin{cases} \sqrt{\frac{2}{N}} & i = 0 \\ 1 & i \neq 0 \end{cases}, \omega_1 = \begin{cases} \sqrt{\frac{2}{N}} & j = 0 \\ 1 & j \neq 0 \end{cases}$
	$T_i(j) = \sqrt{\frac{4}{2N+1}} \cdot \cos\left(\frac{\pi \cdot (2i+1) \cdot (2j+1)}{4N+2}\right)$
	$T_i(j) = \sqrt{\frac{2}{N+1}} \cdot \sin\left(\frac{\pi \cdot (i+1) \cdot (j+1)}{N+1}\right)$
	$T_i(j) = \sqrt{\frac{4}{2N+1}} \cdot \sin\left(\frac{\pi \cdot (2i+1) \cdot (2j+1)}{2N+1}\right)$

[0058] 表1用于N点输入的DCT-II/V/VIII和DSTI/VII的变换基函数

[0059] HEVC标准规定了尺寸为 $4 \times 4$ 、 $8 \times 8$ 、 $16 \times 16$ 和 $32 \times 32$ 的核心变换矩阵,用于基于块的运动补偿视频压缩的上下文中的二维变换。像先前视频编解码标准一样,HEVC为所有变换尺寸规定了类似于逆离散余弦变换(IDCT)的二维变换。多种变换尺寸提高了压缩性能,但也增加了实现方式的复杂性。

[0060] 沿着预测方向,离散正弦变换类型VII(DST-VII)比DCT-II更接近最佳变换。HEVC中已经采用 $4 \times 4$ DST-VII来编解码亮度帧内预测残差。

[0061] 图2是HEVC中的变换和逆变换的示意图200。 $N \times N$ 块的变换过程可表示为:

$$[0062] \quad w_i = \sum_{j=0}^{N-1} u_j c_{ij},$$

[0063] 其中 $i=0, \dots, N-1$ 。DCT变换矩阵C的元素 $c_{ij}$ 被定义为:

$$[0064] \quad c_{ij} = \frac{A}{\sqrt{N}} \cdot \cos\left[\frac{\pi \cdot i \cdot (2j+1)}{2N}\right],$$

[0065] 其中 $i=0, \dots, N-1, j=0, \dots, N-1$ ,并且针对 $i=0$ 和 $i>0$ ,A分别等于1和 $2^{\frac{1}{2}}$ 。此外,DCT的基矢量 $c_i$ 被定义为 $c_i = [c_{i0}, c_{i1}, \dots, c_{i(N-1)}]^T$ ,其中 $i=0, \dots, N-1$ 。在图2中, $M = \log_2(N)$ , $S_{T1} = 2^{-(B+M-9)}$ , $S_{T2} = 2^{-(M+6)}$ ,并且B表示位深度。在第一逆变换阶段之后, $S_{IT1} = 2^{-7}$ ,在第二逆变换阶段之后 $S_{IT2} = 2^{-(20-B)}$ 。

[0066] 对于正变换的输出样点系数(coeff),简单的量化方案可以实现如下:

$$[0067] \quad level = \left( (coeff * f_{QP\%6} + offset_Q) \gg \frac{QP}{6} \right) \gg shift2,$$

[0068] 在图2中, $shift2 = 29 - M - B$ , $S_Q = 2^{-shift2}$ 。此外,量化过程中应用以下设置:

$$[0069] \quad Qstep(QP) = (2^{1/6})^{QP-4} = G_{QP\%6} \ll (QP/6)$$

$$[0070] \quad F = [f_0, f_1, \dots, f_5] = [26214, 23302, 20560, 18396, 16384, 14564]$$

$$[0071] \quad G = [G_0, G_1, \dots, G_5] = [2^{-4/6}, 2^{-3/6}, 2^{-2/6}, 2^{-1/6}, 2^0, 2^{1/6}]$$

[0072]  $f_{QP\%6} \approx 2^{14} / G_{QP\%6}$ 。对于量化器输出级别(level),去量化器在HEVC标准中规定为:

$$[0073] \quad coeff_Q = \left( (level * (g_{QP\%6} \ll \frac{QP}{6})) + offset_{IQ} \right) \gg shift1,$$

[0074] 其中,在图2中, $shift1 = M - 9 + B$ 和 $S_{IQ} = 2^{-shift1}$ 。此外,在去量化过程中应用以下设

置:

[0075]  $g_{QP\%6} = \text{round}(2^6 \times G_{QP\%6})$ ,

[0076]  $g = [g_0, g_1, \dots, g_5] = [40, 45, 51, 57, 64, 71]$ 。

[0077] 在HEVC中,只允许正方形的变换块,即变换块的宽度和高度总是相等的。因此,对于 $2^M \times 2^M$ 块,图2中的移位和乘数在表2a和2b中定义。

[0078]

	缩放因子
第一正变换阶段	$2^{(6+M/2)}$
在第一正变换阶段之后 ( $S_{T1}$ )	$2^{-(M+B-9)}$
第二正变换阶段	$2^{(6+M/2)}$
在第二正变换阶段之后 ( $S_{T2}$ )	$2^{-(M+6)}$
正变换的总缩放	$2^{(15-B-M)}$

[0079] 表2a正变换的HEVC的变换过程中的参数定义

[0080]

	缩放因子
第一逆变换阶段	$2^{(6+M/2)}$
在第一逆变换阶段之后 ( $S_{IT1}$ )	$2^{-7}$
第二逆变换阶段	$2^{(6+M/2)}$
在第二逆变换阶段之后 ( $S_{IT2}$ )	$2^{-(20-B)}$
逆变换的总缩放	$2^{-(15-B-M)}$

[0081] 表2b逆变换的HEVC的变换过程中的参数定义

[0082] 在VVC,变换/量化/去量化/逆变换的过程类似于图2所示的HEVC的过程。与HEVC不同的是VVC支持矩形块,即宽度和高度可以不同。假设变换块的宽度和高度分别为W和H,则:

[0083]  $S_{T1} = 2^{-(\lfloor \log_2 W \rfloor + B - 9)}$

[0084]  $S_{T2} = 2^{-(\lfloor \log_2 H \rfloor + 6)}$ 。

[0085] 然后如表3a和3b所示,相应地修改图2中针对VVC的移位和乘数。

[0086]

	缩放因子
第一正变换阶段	$2^{(6 + \lfloor \log_2 W \rfloor / 2)}$
在第一正变换阶段之后 ( $S_{T1}$ )	$2^{-(\lfloor \log_2 W \rfloor + B - 9)}$
第二正变换阶段	$2^{(6 + \lfloor \log_2 H \rfloor / 2)}$
在第二正变换阶段之后 ( $S_{T2}$ )	$2^{-(\lfloor \log_2 H \rfloor + 6)}$
正变换的总缩放	$2^{15-B - (\lfloor \log_2 W \rfloor + \lfloor \log_2 H \rfloor) / 2}$

[0087] 表3a正变换的VVC的变换过程中的参数定义

	缩放因子
第一逆变换阶段	$2^{(6+\lfloor \log_2 W \rfloor/2)}$
在第一逆变换阶段之后 ( $S_{IT1}$ )	$2^{-7}$
第二逆变换阶段	$2^{(6+\lfloor \log_2 H \rfloor/2)}$
在第二逆变换阶段之后 ( $S_{IT2}$ )	$2^{-(20-B)}$
逆变换的总缩放	$2^{-(15-B-(\lfloor \log_2 W \rfloor+\lfloor \log_2 H \rfloor)/2)}$

[0089] 表3b逆变换的VVC的变换过程中的参数定义

[0090] 与HEVC相比,当 $\lfloor \log_2 W \rfloor + \lfloor \log_2 H \rfloor$ 是偶数时,相同的量化/去量化因子可以被使用。如果 $\lfloor \log_2 W \rfloor + \lfloor \log_2 H \rfloor$ 是奇数,则必须在量化/去量化阶段补偿因子 $2^{1/2}$ 。如果 $\lfloor \log_2 W \rfloor + \lfloor \log_2 H \rfloor$ 是奇数,则用 $f' = [f'_0, f'_1, \dots, f'_5] = [18396, 16384, 14564, 13107, 11651, 10280]$ 代替 $f$ ,并且用 $g' = [g'_0, g'_1, \dots, g'_5] = [57, 64, 72, 80, 90, 102]$ 代替 $g$ 。 $f' \approx f \times 2^{-1/2}$ 和 $g' \approx g \times 2^{1/2}$ 。

[0091] 除了已经在HEVC中采用的DCT-II之外,多重变换选择(MTS)方案被用于帧间和帧内编解码块两者的残差编解码。MTS使用DCT8/DST7中的多个选定变换。新引入的变换矩阵是DST-VII (DST7)和DCT-VIII (DCT8)。表1示出了所选DST/DCT的基函数。启用MTS有两种方式,一种是隐式MTS,另一种是显式MTS。

[0092] 隐式MTS是VVC中采用的一种工具。是否启用隐式MTS取决于变量implicitMtsEnabled的值。当implicitMtsEnabled被设置为等于1时,应用隐式MTS。隐式MTS仅适用于帧内编解码块。变量implicitMtsEnabled的导出如下:如果sps\_mts\_enabled\_flag等于1,且下列一个或多个条件为真,则implicitMtsEnabled被设置为等于1: IntraSubPartitionSplitType不等于ISP\_NO\_SPLIT;cu\_sbt\_flag等于1,并且Max(nTbW, nTbH)小于等于32;或者sps\_explicit\_mts\_intra\_enabled\_flag等于0,以及CuPredMode[0][xTbY][yTbY]等于MODE\_INTRA,以及lfnst\_idx等于0,以及IntraMipFlag[x0][y0]等于0。否则,implicitMtsEnabled被设置为等于0。

[0093] 规定水平变换核的变量trTypeHor和规定垂直变换核的变量trTypeVer被导出如下:如果下列一个或多个条件为真, trTypeHor和trTypeVer被设置为等于0: cIdx大于0;或者IntraSubPartitionsSplitType不等于ISP\_NO\_SPLIT且lfnst\_idx不等于0。否则,如果implicitMtsEnabled等于1,则适用以下情况:如果cu\_sbt\_flag等于1,则根据cu\_sbt\_horizontal\_flag和cu\_sbt\_pos\_flag规定trTypeHor和trTypeVer,如表4所示。

[0094]	cu_sbt_horizontal_flag	cu_sbt_pos_flag	trTypeHor	trTypeVer
	0	0	2	1
	0	1	1	1
	1	0	1	2
	1	1	1	1

[0095] 表4取决于cu\_sbt\_horizontal\_flag和cu\_sbt\_pos\_flag的trTypeHor和

trTypeVer的规范

[0096] 否则 (cu\_sbt\_flag等于0), trTypeHor和trTypeVer导出如下:

[0097]  $\text{trTypeHor} = (\text{nTbW} \geq 4 \& \& \text{nTbW} \leq 16) ? 1 : 0$

[0098]  $\text{trTypeVer} = (\text{nTbH} \geq 4 \& \& \text{nTbH} \leq 16) ? 1 : 0$

[0099] 否则,根据mts\_idx规定trTypeHor和trTypeVer,如表5所示。

[0100] mts_idx	0	1	2	3	4
trTypeHor	0	1	2	1	2
trTypeVer	0	1	1	2	2

[0101] 表5取决于mts\_idx的trTypeHor和trTypeVer的规范

[0102] 规定水平变换核的变量trTypeHor和规定垂直变换核的变量trTypeVer导出如下:如果以下一个或多个条件为真, trTypeHor和trTypeVer被设置为等于0: cIdx大于0;或者IntraSubPartitionsSplitType不等于ISP\_NO\_SPLIT且lfnst\_idx不等于0。否则,如果implicitMtsEnabled等于1,则适用以下情况:如果cu\_sbt\_flag等于1,则根据cu\_sbt\_horizontal\_flag和cu\_sbt\_pos\_flag规定trTypeHor和trTypeVer,如表4所示。否则 (cu\_sbt\_flag等于0), trTypeHor和trTypeVer导出如下:

[0103]  $\text{trTypeHor} = (\text{nTbW} \geq 4 \& \& \text{nTbW} \leq 16) ? 1 : 0$

[0104]  $\text{trTypeVer} = (\text{nTbH} \geq 4 \& \& \text{nTbH} \leq 16) ? 1 : 0$

[0105] 否则,根据mts\_idx规定trTypeHor和trTypeVer,如表5所示。变量nonZeroW和nonZeroH导出如下:如果ApplyLfnstFlag[cIdx]等于1,则以下情况适用:

[0106]  $\text{nonZeroW} = (\text{nTbW} == 4 \mid \mid \text{nTbH} == 4) ? 4 : 8$

[0107]  $\text{nonZeroH} = (\text{nTbW} == 4 \mid \mid \text{nTbH} == 4) ? 4 : 8$

[0108] 否则,以下情况适用:

[0109]  $\text{nonZeroW} = \text{Min}(\text{nTbW}, (\text{trTypeHor} > 0) ? 16 : 32)$

[0110]  $\text{nonZeroH} = \text{Min}(\text{nTbH}, (\text{trTypeVer} > 0) ? 16 : 32)$

[0111] 为了控制MTS方案,使用一个标志来规定比特流中是否存在帧内/帧间的显式MTS。此外,在序列参数集 (SPS) 级别,为帧内和帧间分别规定了两个单独的启用标志,以指示是否启用显式MTS。当在SPS级别启用MTS时,可以信令通知编解码单元 (CU) 级别的变换索引,以指示是否应用MTS。这里,MTS仅适用于亮度。当满足以下条件时,MTS CU级别索引 (由mts\_idx表示) 被信令通知:宽度和高度都小于或等于32;CBF亮度标志等于1;非TS;非ISP;非SBT;LFNST被禁用;存在不在DC位置 (块的左上位置) 的非零系数;在左上的16×16区域之外没有非零系数。

[0112] 如果mts\_idx的第一个二进制位等于零,则DCT2在两个方向上应用。然而,如果mts\_idx的第一个二进制位等于1,则另外信令通知另外两个二进制位,以分别指示水平和垂直方向的变换类型。转换和信令通知映射如表6所示。

[0113]	第 0 个	第 1 个	第 2 个	帧内/帧间		mts_idx
	二进制制位	二进制制位	二进制制位	水平	垂直	
	0			DCT2		0
	1	0	0	DST7	DST7	1
		0	1	DCT8	DST7	2
		1	0	DST7	DCT8	3
		1	1	DCT8	DCT8	4

[0114] 表6MTS的信令通知

[0115] 8位初级变换核心用于变换矩阵精度。因此,HEVC中使用的所有变换核心保持不变,包括4点DCT-2和DST-7、8点、16点和32点DCT-2。此外,包括64点DCT-2、4点DCT-8、8点、16点、32点DST-7和DCT-8的其他变换核心使用8位初级变换核心。

[0116] 为了降低大尺寸DST-7和DCT-8的复杂性,对于尺寸(宽度或高度,或宽度和高度)等于32的DST-7和DCT-8块,高频变换系数被归零。仅保留16x16低频区域内的系数。

[0117] 如在HEVC,块的残差可以用变换跳过模式来编解码。为了避免语法编解码的冗余,当CU级别MTS\_CU\_flag不等于零时,不信令通知变换跳过标志。变换跳过的块尺寸限制与JEM4中MTS的块尺寸限制相同,这表明当块宽度和高度都等于或小于32时,变换跳过适用于CU。

[0118] 在VTM8中,启用了尺寸高达64×64的大块尺寸变换,其主要用于更高分辨率的视频,例如1080p和4K序列。对于尺寸(宽度或高度,或宽度和高度)不小于64的变换块,应用DCT2变换的块的高频变换系数被归零,从而仅保留低频系数。所有其他系数被强制归零且不被信令通知。例如,对于M×N变换块,M为块宽,N为块高,当M不小于64时,只保留变换系数的左32列。类似地,当N不小于64时,只保留变换系数的上32行。

[0119] 对于尺寸(宽度或高度,或宽度和高度)不小于32的变换块,应用DCT8或DST7变换的块的高频变换系数被归零,从而仅保留低频系数,所有其他系数被强制归零且不被信令通知。例如,对于M×N变换块,M为块宽,N为块高,当M不小于32时,只保留变换系数的左16列。类似地,当N不小于32时,只保留变换系数的上16行。

[0120] 图3是JEM中的二次变换300的示意图。在JEM中,二次变换应用于正初级变换和量化之间(在编码器处)以及去量化和逆初级变换之间(在解码器侧)。如图3所示,执行4×4二次变换还是8×8二次变换取决于块尺寸。例如,对小块(即,min(宽度,高度)<8)应用4×4二次变换,并且对每个8×8块的较大块(即,min(宽度,高度)>4)应用8×8二次变换。

[0121] 使用来自输入图片的输入块作为示例,不可分变换的应用描述如下。为了应用不可分变换,4×4输入块X首先被表示为向量 $\vec{X}$ ,如下所示:

$$[0122] \quad X = \begin{bmatrix} X_{00} & X_{01} & X_{02} & X_{03} \\ X_{10} & X_{11} & X_{12} & X_{13} \\ X_{20} & X_{21} & X_{22} & X_{23} \\ X_{30} & X_{31} & X_{32} & X_{33} \end{bmatrix}$$

$$[0123] \quad \vec{X} = [X_{00} X_{01} X_{02} X_{03} X_{10} X_{11} X_{12} X_{13} X_{20} X_{21} X_{22} X_{23} X_{30} X_{31} X_{32} X_{33}]^T。$$

[0124] 不可分变换计算如下:  $\vec{F} = T \cdot \vec{X}$ , 其中  $\vec{F}$  表示变换系数矢量,  $T$  是  $16 \times 16$  变换矩阵。使用  $4 \times 4$  块的扫描顺序 (水平、垂直或对角线),  $16 \times 1$  系数矢量  $\vec{F}$  随后被重新组织为  $4 \times 4$  块。具有较小索引的系数将被放置在  $4 \times 4$  系数块中具有较小扫描索引的位置。总共有 35 个变换集, 并且每个变换集使用 3 个不可分变换矩阵 (核)。从帧内预测模式到变换集的映射是预定义的。对于每个变换集, 所选择的不可分二次变换候选进一步地被显式信令通知的二次变换索引规定。在变换系数之后, 每个帧内 CU 在比特流中信令通知一次索引。

[0125] 图 4 是示例 LFNST 400 的示意图。JVET-K0099 中引入了 LFNST, 并且 JVET-L0133 中引入了 4 变换集 (而不是 35 变换集) 映射。在 JVET-N0193 中,  $16 \times 64$  (可以进一步简化为  $16 \times 48$ ) 和  $16 \times 16$  矩阵分别用于  $8 \times 8$  块和  $4 \times 4$  块。为了便于表示, 将  $16 \times 64$  (可以进一步简化为  $16 \times 48$ ) 变换表示为 LFNST  $8 \times 8$ , 将  $16 \times 16$  变换表示为 LFNST  $4 \times 4$ 。

[0126] 图 5 是正逆简化变换 (RT) 500 的示意图。RT 的主要思想是将  $N$  维矢量映射为不同空间的  $R$  维矢量, 其中  $R/N$  ( $R < N$ ) 为简化因子。RT 矩阵是  $R \times N$  矩阵, 如下所示:

$$[0127] \quad T_{R \times N} = \begin{bmatrix} t_{11} & t_{12} & t_{13} & \cdots & t_{1N} \\ t_{21} & t_{22} & t_{23} & \cdots & t_{2N} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ t_{R1} & t_{R2} & t_{R3} & \cdots & t_{RN} \end{bmatrix},$$

[0128] 其中该变换的  $R$  行是  $N$  维空间的  $R$  基。RT 的逆变换矩阵是其正变换的转置矩阵。

[0129] 这里应用了简化因子为 4 (1/4 尺寸) 的 LFNST  $8 \times 8$ 。因此, 使用  $16 \times 64$  直接矩阵而不是  $64 \times 64$  (传统的  $8 \times 8$  不可分变换矩阵尺寸)。换句话说, 解码器侧使用  $64 \times 16$  逆 LFNST 矩阵来生成  $8 \times 8$  左上区域的核心 (初级) 变换系数。正 LFNST  $8 \times 8$  使用  $16 \times 64$  (或者对于  $8 \times 8$  块是  $8 \times 64$ ) 矩阵, 因此仅在给定  $8 \times 8$  区域内的左上  $4 \times 4$  区域中产生非零系数。换句话说, 如果应用 LFNST, 那么除了左上的  $4 \times 4$  区域之外的  $8 \times 8$  区域将只有零系数。对于 LFNST  $4 \times 4$ , 应用  $16 \times 16$  (或者对于  $4 \times 4$  块是  $8 \times 16$ ) 的直接矩阵乘法。

[0130] 当满足以下两个条件时, 有条件地应用逆 LFNST: 1) 块尺寸大于或等于给定阈值 ( $W \geq 4$  且  $H \geq 4$ ), 以及 2) 变换跳过模式标志等于零。如果变换系数块的宽度 ( $W$ ) 和高度 ( $H$ ) 都大于 4, 则 LFNST  $8 \times 8$  被应用于变换系数块的左上  $8 \times 8$  区域。否则, LFNST  $4 \times 4$  被应用于变换系数块的左上  $\min(8, W) \times \min(8, H)$  区域。如果 LFNST 索引等于 0, 则不应用 LFNST。否则, 将应用 LFNST, 其中使用 LFNST 索引选择 LFNST 的核。LFNST 选择方法和 LFNST 索引的编解码如下所述。

[0131] 此外, LFNST 适用于帧内和帧间条带两者中的帧内 CU, 以及适用于亮度和色度两者。如果启用了双树, 则亮度和色度的 LFNST 索引会分开被信令通知。对于帧间条带 (双树被禁用) 或双树禁用的 I 条带 (即单树), 单个 LFNST 索引被信令通知并且仅用于亮度。

[0132] 在第 13 届 JVET 会议上, 帧内子分割 (ISP) 作为一种新的帧内预测模式被采用。当选

择ISP模式时,LFNST被禁用,并且LFNST索引不被信令通知,因为即使LFNST被应用于每个可行的分割块,性能改善也是微不足道的。此外,对ISP预测的残差禁用LFNST可以降低编码复杂度。

[0133] 从四个变换集中选择LFNST矩阵,每个变换集包括两个变换。应用哪个变换集由帧内预测模式确定如下:如果三种CCLM模式之一被指示,则选择变换集0。否则,根据表7执行变换集选择。

IntraPredMode	Tr. 集索引
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	1

[0135] 表7变换集选择表

[0136] 访问表7的索引被表示为IntraPredMode,具有[-14,83]的范围,这是用于广角帧内预测的变换模式索引。

[0137] 图6是具有 $16 \times 48$ 矩阵的正LFNST  $8 \times 8$ 过程600的示例的示意图。如图所示,作为进一步的简化,应用 $16 \times 48$ 矩阵而不是具有相同变换集配置的 $16 \times 64$ 矩阵,每个矩阵从左上 $8 \times 8$ 块中的三个 $4 \times 4$ 块(不包括右下的 $4 \times 4$ 块)中获取48个输入数据。

[0138] 图7是用于支持LFNST的信令通知的示例扫描机制700的示意图。 $R=16$ 的正LFNST  $8 \times 8$ 使用 $16 \times 64$ 矩阵,因此仅在给定的 $8 \times 8$ 区域内的左上 $4 \times 4$ 区域中产生非零系数。换句话说,如果应用LFNST,则 $8 \times 8$ 区域中除了左上 $4 \times 4$ 区域之外的区域仅生成零系数。因此,如图所示,当在 $8 \times 8$ 块区域中除左上 $4 \times 4$ 之外的区域中检测到任何非零元素时,LFNST索引不被编解码,因为这意味着LFNST不被应用。在这种情况下,LFNST索引被推断为零。

[0139] 通常,在对 $4 \times 4$ 子块应用逆LFNST之前, $4 \times 4$ 子块中的任何系数可以是非零的。然而,可以应用约束,即在某些情况下,在对子块应用逆LFNST之前, $4 \times 4$ 子块中的某些系数必须为零。设nonZeroSize为变量。规则可能要求,在应用逆LFNST之前,当系数被重新排列成1-D数组时,索引不小于nonZeroSize的任何系数必须为零。当nonZeroSize等于16时,左上 $4 \times 4$ 子块中的系数没有归零约束。当当前块尺寸为 $4 \times 4$ 或 $8 \times 8$ 时,nonZeroSize被设置为等于8。对于其他块维度,nonZeroSize被设置为等于16。

[0140] 当前隐式MTS设计仅考虑编解码块(变换块)的宽度和高度。其使用DST7作为给定块(宽度或高度大于或等于4且小于或等于16)的变换核。这种设计易于实现,但是可能效率不高,因为它没有考虑编解码块的内容特性或残差特性。

[0141] 本文公开了解决上述一个或多个问题的机制。所公开的实施例实现了基于特征的变换选择(FTS)。在FTS中,根据从当前块的邻近区域提取的特征来选择当前块的变换核。邻近区域可以与当前块紧邻或不紧邻。在下面的描述中,当前块的宽度和高度分别由W和H表示。例如,当前块可指当前块的上邻近区域、左邻近区域、左上邻近区域、右上邻近区域或非紧邻区域。例如,当前块可以指位于邻近区域中的子区域。例如,可以通过分类信息将子区

域分类为特定类别,并且分类信息可以用于为当前块选择变换核。例如,分类过程可以取决于从子区域提取的特征。例如,子区域可以被分类为噪声区域、平滑块或边缘区域。例如,除了分类信息之外,根据诸如块尺寸或预测模式的其他相关辅助信息来选择当前块的变换核。因此,为了解决上述问题,公开了几种方法。这些方法能实现更好的编解码性能。

[0142] 以下详细实施例应视为解释一般概念的示例。这些实施例不应被狭义地解释。此外,实施例可以以任何方式组合。

[0143] 示例1

[0144] 当前块是指当前块的邻近区域。邻近区域是当前块的上邻近区域、左邻近区域、左上邻近区域、右上邻近区域或非紧邻区域。邻近区域可以是宽度为 $W_1$ 和高度为 $H_1$ 的矩形。 $W_1$ 和 $H_1$ 可以相同。当讨论诸如邻近区域的区域时,可以使用块来代替。

[0145] 图8是当前块及其邻近区域和非紧邻区域的示意图800。当前块在中心,并且被上邻近区域、左邻近区域、左上邻近区域、右上邻近区域和非紧邻区域包围。上邻近区域内是子区域。虽然图8将当前块及其邻近和非紧邻区域显示为正方形,但是它们也可以是矩形。

[0146] 示例1a

[0147]  $W_1$ 和 $H_1$ 是预先确定的。例如, $W_1$ 和 $H_1$ 被设置为等于4、8、16、32等。

[0148] 示例1b

[0149] 根据当前块的尺寸分配邻近区域的尺寸。例如,尺寸是 $W_1 * H_1$ (即,与当前块相同的维度)、 $W_1 * 2 * H_1$ (即,宽度与当前块相同,但是高度是当前块的两倍)、 $2 * W_1 * H_1$ (即,宽度是当前块的两倍,但是高度与当前块相同)等。

[0150] 示例1c

[0151] 根据当前块的帧内模式自适应分配邻近区域的尺寸。

[0152] 示例2

[0153] 基于子区域的左上样点,子区域可被包括在邻近区域中。子区域可以位于每个邻近区域内。子区域应该是矩形区域。 $w$ 和 $h$ 分别表示子区域的宽度和高度,并且 $w \leq W_1, h \leq H_1$ 。

[0154] 示例3

[0155] 从子区域中的重建样点提取的特征用于确定变换核。

[0156] 示例3a

[0157] 可通过分类信息将子区域分为特定类别。分类信息可用于为当前块选择变换核。

[0158] 示例3b

[0159] 子区域可分为噪声区域、平滑区域或边缘区域。

[0160] 示例4

[0161] 分类过程可取决于从子区域提取的特征。特征可以由一个区域中的第一样点和围绕第一样点的第二样点之间的差、一组样点的平均值或边缘信息来定义。可以首先检测邻近区域的边缘,然后将样点分类为边缘样点或非边缘样点。例如,如果样点和边缘之间的距离小于阈值,则该样点可以被分类为边缘样点。

[0162] 示例5

[0163] 可基于跳跃点的标识将子区域分类为噪声区域。

[0164] 示例5a

[0165] 对于上邻近区域中宽度为 $W_1$ 、高度为 $H_1$ 的子区域、左邻近区域中宽度为 $W_2$ 、高度为

$H_2$ 的子区域、左上邻近区域中宽度为 $W_3$ 、高度为 $H_3$ 的子区域或右上邻近区域中宽度为 $W_4$ 、高度为 $H_4$ 的子区域,如果位置 $(i, j)$ 处的重建像素值 $P_{cur}$ 及其位置 $(i-1, j)$ 、 $(i+1, j)$ 、 $(i, j-1)$ 和 $(i, j+1)$ 处的可用周围像素( $P_l$ 、 $P_r$ 、 $P_t$ 和 $P_d$ )分别满足其中一个条件,则将位置 $(i, j)$ 处的当前像素标记为跳转点:

[0166] 条件1:如果 $P_{cur}$ 和 $P_l$ 、 $P_r$ 、 $P_t$ 或 $P_d$ 之间的差大于阈值 $Thr1_1$ ,即 $\max\{P_{cur}-P_l, P_{cur}-P_r, P_{cur}-P_t, P_{cur}-P_d\}$ 大于 $Thr1_1$ ,则位置 $(i, j)$ 处的像素为跳转点。

[0167] 条件2:如果 $P_{cur}$ 和 $P_l$ 、 $P_r$ 、 $P_t$ 或 $P_d$ 之间的绝对差之一大于阈值 $Thr1_2$ ,即 $\max\{abs(P_{cur}-P_l), abs(P_{cur}-P_r), abs(P_{cur}-P_t), abs(P_{cur}-P_d)\}$ 大于 $Thr1_2$ ,则位置 $(i, j)$ 处的像素为跳转点。

[0168] 条件3:如果 $P_{cur}$ 和 $P_{avg}$ 之间的差小于阈值 $Thr1_3$ ,则位置 $(i, j)$ 处的像素是跳转点。 $P_{avg}$ 是子区域的平均像素值。

[0169] 术语“像素”和“样点”可互换使用。

[0170] 示例5b

[0171] 当以下条件之一为真时,可将子区域分类为噪声区域:

[0172] 条件1:子区域中的所有像素都是跳转点。

[0173] 条件2:子区域中跳转点的总数大于 $Thr1_4*(W_1*H_1+W_2*H_2+W_3*H_3+W_4*H_4)$ 。

[0174] 示例5c

[0175]  $Thr1_4$ 可设置为通过数据训练实现的固定值,或可自适应地取决于子区域尺寸。

[0176] 示例6

[0177] 基于正常点的标识,可将子区域分类为平滑区域。

[0178] 示例6a

[0179] 对于上、左、左上、右上或非紧邻邻近区域中宽度为 $w$ 、高度为 $h$ 的子区域,如果位置 $(i, j)$ 处的重建像素值 $P_{cur}$ 及其位置 $(i-1, j)$ 、 $(i+1, j)$ 、 $(i, j-1)$ 和 $(i, j+1)$ 处的可用周围像素( $P_l$ 、 $P_r$ 、 $P_t$ 和 $P_d$ )满足以下条件之一,则将位置 $(i, j)$ 处的当前像素标记为正常点:

[0180] 条件1:如果 $P_{cur}$ 和 $P_l$ 、 $P_r$ 、 $P_t$ 和 $P_d$ 之间的最小绝对差小于阈值 $Thr2_1$ ,意味着 $\min\{abs(P_{cur}-P_l), abs(P_{cur}-P_r), abs(P_{cur}-P_t), abs(P_{cur}-P_d)\}$ 小于 $Thr2_1$ ,则位置 $(i, j)$ 处的像素为正常点。

[0181] 条件2:如果 $P_{cur}$ 和 $P_{avg}$ 之间的差小于阈值 $Thr2_2$ ,则位置 $(i, j)$ 处的像素为正常点。 $P_{avg}$ 是子区域的平均像素值。

[0182] 示例6b

[0183] 当以下条件之一为真时,子区域可被分类为平滑区域:

[0184] 条件1:子区域中的所有像素都是正常点。

[0185] 条件2:子区域中正常点的总数大于 $Thr2_3*(W_1*H_1+W_2*H_2)$ 。

[0186] 示例7

[0187] 可基于边缘像素的标识将子区域分类为边缘区域。

[0188] 示例7a

[0189] 对于上、左、左上、右上或非紧邻邻近区域中宽度为 $w$ 、高度为 $h$ 的子区域,当以下条件之一为真时,可将该子区域分类为边缘区域:

[0190] 条件1:子区域中的所有像素都是边缘像素。边缘像素可被定义为邻接诸如块边界或子区域边界等边界的像素。

- [0191] 条件2:子区域中的边缘像素总数大于阈值 $\text{Thr}3_1 * (W_1 * H_1, W_2 * H_2 \text{ 或 } W_3 * H_3)$ 。
- [0192] 示例8
- [0193] 根据子区域的分类信息或子区域的其他辅助信息,如块尺寸或预测模式,选择当前块的变换核。
- [0194] 示例8a
- [0195] 亮度分量的水平变换或垂直变换的变换核可以是DCT2或DCT7。
- [0196] 示例8b
- [0197] 当以下条件之一为真时,亮度分量的水平变换或垂直变换的变换核为DCT2:
- [0198] 条件1:上邻近区域和左上邻近区域中的所有子区域都是噪声区域。
- [0199] 条件2:上邻近区域中的所有子区域都是噪声区域。
- [0200] 条件3:上邻近区域中噪声区域与子区域总数的比率以及左上邻近区域中噪声区域与子区域总数的比率均大于阈值 $\text{Thr}4_1$ 。
- [0201] 条件4:上邻近区域中噪声区域与全部子区域的比率大于阈值 $\text{Thr}4_2$ 。
- [0202] 条件5:上邻近区域和左上邻近区域中的所有子区域都是边缘区域,当前块是帧内块,并且当前块的块尺寸小于上邻近区域中子区域的块尺寸。
- [0203] 条件6:上邻近区域和左上邻近区域中的所有子区域均为边缘区域,当前块为帧内块,且被分割以创建当前块的CTU的总分割深度大于上邻近区域中任何子区域的块尺寸。
- [0204] 示例8c
- [0205] 当以下条件之一为真时,亮度分量的水平变换或垂直变换的变换核为DCT7:
- [0206] 条件1:左邻近区域和上邻近区域中的所有子区域都是平滑区域,并且当前块的当前预测模式是DC模式。
- [0207] 条件2:左邻近区域和上邻近区域中的所有子区域都是平滑区域,并且当前块的当前预测模式是平面模式。
- [0208] 示例9
- [0209] 邻近区域中的亮度、Cb或Cr样点可用于确定当前块中的亮度、Cb或Cr样点的变换核。
- [0210] 示例9a
- [0211] 邻近区域中的亮度样点可用于确定当前块的所有分量的变换核。
- [0212] 示例10
- [0213] 是否或如何使用FTS可取决于块维度或比特流中信令通知的语法元素。例如,在比特流中信令通知是否应用FTS的指示。
- [0214] 图9是实现本文中所公开的各种技术的示例视频处理系统900的框图。各种实现方式可以包括系统900中的一些或全部组件。系统900可以包括用于接收视频内容的输入902。视频内容可以以原始或未压缩的格式(例如8或10比特多分量像素值)接收,或者可以以压缩或编码的格式接收。输入902可以代表网络接口、外围总线接口或存储接口。网络接口的示例包括有线接口(诸如以太网、无源光网络(PON)等)和无线接口(诸如Wi-Fi或蜂窝接口)。
- [0215] 系统900可以包括可以实现本文档中描述的各种编解码或编码方法的编解码组件904。编解码组件904可以减少从输入902到编解码组件904的输出的视频的平均比特率,以

产生视频的编解码表示。因此,编解码技术有时称为视频压缩或视频转码技术。编解码组件904的输出可以被存储或经由所连接的通信来发送,如组件906所表示的。在输入902处接收的视频的存储或通信的比特流(或编解码)表示可以由组件908使用,以生成被发送到显示接口910的像素值或可显示视频。从比特流表示中生成用户可见的视频的过程有时称为视频解压缩。此外,尽管某些视频处理操作被称为“编解码”操作或工具,但是应当理解,在编码器处使用编码工具或操作,相应的解码工具或操作将由解码器执行,以反转编码的结果。

[0216] 外围总线接口或显示接口的示例可以包括通用串行总线(USB)或高清晰度多媒体接口(HDMI)或Displayport等。存储接口的示例包括串行高级技术附件(SATA)、外围组件互连(PCI)、集成驱动电子设备(IDE)接口等。本文档中描述的技术可以实施在各种电子设备中,诸如移动电话、膝上型计算机、智能电话或其它能够进行数字数据处理和/或视频显示的设备。

[0217] 图10示例视频处理装置1000的框图。装置1000可以用于实现本文中所述的方法中的一个或多个。装置1000可以实施在智能电话、平板电脑、计算机、物联网(IoT)接收器等中。装置1000可以包括一个或多个处理器1002、一个或多个存储器1004和视频处理硬件1006。(多个)处理器1002可以配置为实现本文档中所述的一个或多个方法。(多个)存储器1004可以用于存储数据和代码,该代码用于实现本文所描述的方法和技术。视频处理硬件1006可以用于在硬件电路中实现本文档中所描述的一些技术。在一些实施例中,视频处理硬件1006可以至少部分地包括在处理器1002中,例如图形协处理器。

[0218] 图11是视频处理的示例方法1100的流程图。在步骤1102,基于与当前块邻近的邻近区域的特征或者基于邻近区域的子区域,确定(例如,信令通知)当前块的变换核。在步骤1104,基于变换核执行可视媒体数据和比特流之间的转换。

[0219] 邻近区域可与当前块紧邻或不紧邻。邻近区域可以是上邻近区域、左邻近区域、左上邻近区域或右上邻近区域。邻近区域的尺寸可被预先确定、根据当前块的尺寸被分配或根据当前块的帧内模式被分配。基于子区域的左上样点,子区域可被包括在邻近区域中。

[0220] 方法1100还可包括从子区域中的重建样点中提取特征。特征可以基于子区域的分类信息。分类信息可以是:子区域是噪声区域、平滑区域或边缘区域。特征可以基于一个区域中的第一样点和围绕第一样点的第二样点之间的差、一组样点的平均值或、边缘信息。

[0221] 方法1100还可包括基于子区域中跳跃点的标识将子区域分类为噪声区域、基于子区域中正常点的标识将子区域分类为平滑区域、或者基于子区域中边缘像素的标识将子区域分类为边缘区域。特征可以基于子区域的辅助信息。辅助信息可以是块尺寸。辅助信息可以是预测模式。变换核可以是离散余弦变换类型二(DCT2)。变换核可以是离散余弦变换类型七(DCT7)。

[0222] 该方法还可包括基于比特流中的块维度或信令通知来确定是否使用基于特征的变换选择(FTS)。该方法还可包括基于上述比特流中的块维度或信令通知来确定如何使用基于特征的变换选择(FTS)。

[0223] 图12是示出可利用本公开的技术的示例视频编解码系统1200的框图。如图12所示,视频编解码系统1200可以包括源设备1210和目标设备1220。源设备1210生成编码的视频数据,其可以被称为视频编码设备。目标设备1220可以解码由源设备1210生成的编码的视频数据,该目标设备1220可以被称为视频解码设备。

[0224] 源设备1210可以包括视频源1212、视频编码器1214和输入/输出(I/O)接口1216。视频源1212可以包括诸如视频捕获设备的源、从视频内容提供者接收视频数据的接口、和/或生成视频数据的计算机图形系统,或这些源的组合。视频数据可以包括一个或多个图片。视频编码器1214对来自视频源1212的视频数据进行编码以生成比特流。比特流可以包括形成视频数据的编解码表示的比特序列。比特流可以包括编解码图片和相关联的数据。编解码图片是图片的编解码表示。相关联的数据可以包括序列参数集、图片参数集和其他语法元素。I/O接口1216包括调制器/解调器(调制解调器)和/或发送器。可以将编码的视频数据经由I/O接口1216通过网络1230直接发送到目标设备1220。还可以将编码的视频数据存储到存储介质/服务器1240上,用于由目标设备1220访问。

[0225] 目标设备1220可以包括I/O接口1226、视频解码器1224和显示设备1222。I/O接口1226可以包括接收器和/或调制解调器。I/O接口1226可以从源设备1210或存储介质/服务器1240获取编码的视频数据。视频解码器1224可以对编码的视频数据进行解码。显示设备1222可以向用户显示解码的视频数据。显示设备1222可以与目标设备1220集成,或可以在配置为与外置显示设备相接的目标设备1220外部。

[0226] 视频编码器1214和视频解码器1224可以根据视频压缩标准(诸如,高效视频编解码(HEVC)标准、多功能视频编解码(VVC)标准和其他当前和/或其他标准)进行操作。

[0227] 图13是示出视频编码器1300的示例的框图,该视频编码器1300可以是图12中示出的系统1200中的视频编码器1214。视频编码器1300可以被配置为执行本公开的任何或全部技术。在图13的示例中,视频编码器1300包括多个功能组件。本公开所描述的技术可以在视频编码器1300的各种组件之间共享。在一些示例中,处理器可以配置为进行本公开中描述的任何或全部技术。

[0228] 视频编码器1300的功能组件可以包括分割单元1301、预测单元1302(其可以包括模式选择单元1303、运动估计单元1304、运动补偿单元1305、帧内预测单元1306)、残差生成单元1307、变换处理单元1308、量化单元1309、逆量化单元1310、逆变换单元1311、重建单元1312、缓冲器1313和熵编码单元1314。

[0229] 在其他示例中,视频编码器1300可以包括更多、更少或不同的功能组件。在一个示例中,预测单元1302可以包括帧内块复制(IBC)单元。IBC单元可以以IBC模式进行预测,其中至少一个参考图片是当前视频块所位于的图片。

[0230] 此外,诸如运动估计单元1304和运动补偿单元1305的一些组件可以被高度集成,但是出于解释的目的在图13的示例中分开表示。

[0231] 分割单元1301可以将图片分割成一个或多个视频块。视频编码器1200和视频解码器1300可以支持各种视频块尺寸。

[0232] 模式选择单元1303可以例如基于错误结果选择帧内或帧间的编解码模式中的一个,并且将得到的帧内或帧间编解码块提供到残差生成单元1307来生成残差块数据而且提供到重建单元1312来重建编解码块以用作参考图片。在一些示例中,模式选择单元1303可以选择帧内和帧间预测的组合(CIIP)模式,其中预测是基于帧间预测信号和帧内预测信号。模式选择单元1303还可以为帧间预测情况下的块选择运动矢量的分辨率(例如子像素或整像素精度)。

[0233] 为了对当前视频块进行帧间预测,运动估计单元1304可以通过将来自缓冲器1313

的一个或多个参考帧与当前视频块进行比较,生成当前视频块的运动信息。运动补偿单元1305可以基于来自缓冲器1313的图片(而不是与当前视频块相关联的图片)的运动信息和解码样点来为当前视频块确定预测的视频块。

[0234] 运动估计单元1304和运动补偿单元1305可以为当前视频块进行不同操作,例如执行不同操作取决于当前视频块是在I条带、P条带还是B条带中。

[0235] 在一些示例中,运动估计单元1304可以进行当前视频块的单向预测,并且运动估计单元1304可以在列表0或列表1的参考图片中搜索当前视频块的参考视频块。运动估计单元1304然后可以生成指示列表0或列表1的参考图片中含有参考视频块的参考索引以及指示在当前视频块与参考视频块之间的空域位移的运动矢量。运动估计单元1304可以输出参考索引、预测方向指示符、和运动矢量作为当前视频块的运动信息。运动补偿单元1305可以基于由当前视频块的运动信息指示的参考视频块来生成当前块的预测视频块。

[0236] 在其他示例中,运动估计单元1304可以进行当前视频块的双向预测,运动估计单元1304可以在列表0的参考图片中搜索当前视频块的参考视频块并且还可以在列表1的参考图片中搜索当前视频块的另一个参考视频块。运动估计单元1304然后可以生成指示列表0或列表1的参考图片中含有参考视频块的参考索引以及指示在参考视频块与当前视频块之间的空域位移的运动矢量。运动估计单元1304可以输出参考索引和当前视频块的运动矢量作为当前视频块的运动信息。运动补偿单元1305可以基于由当前视频块的运动信息指示的参考视频块来生成当前视频块的预测视频块。

[0237] 在一些示例中,运动估计单元1304可以输出运动信息的全部集合,用于解码器的解码处理。在一些示例中,运动估计单元1304可以不输出当前视频的运动信息的全部集合。而是,运动估计单元1304可以参考另一个视频块的运动信息来信令通知当前视频块的运动信息。例如,运动估计单元1304可以确定当前视频块的运动信息与邻近视频块的运动信息足够相似。

[0238] 在一个示例中,运动估计单元1304可以在与当前视频块相关联的语法结构中指示:图14的视频解码器1400指示当前视频块具有与另一个视频块相同的运动信息的值。

[0239] 在另一个示例中,运动估计单元1304可以在与当前视频块相关联的语法结构中标识另一个视频块和运动矢量差(MVD)。运动矢量差指示当前视频块的运动矢量与指示视频块的运动矢量之间的差。视频解码器1400可以使用指示视频块的运动矢量和运动矢量差来确定当前视频块的运动矢量。

[0240] 如上所讨论的,视频编码器1300可以预测性地信令通知运动矢量。可以由视频编码器1300实现的预测性的信令通知技术的两个示例包括高级运动矢量预测(AMVP)和merge模式信令通知。

[0241] 帧内预测单元1306可以对当前视频块进行帧内预测。当帧内预测单元1306对当前视频块进行帧内预测时,帧内预测单元1306可以基于相同图片中其他视频块的解码样点来生成当前视频块的预测数据。当前视频块的预测数据可以包括预测视频块和各种语法元素。

[0242] 残差生成单元1307可以通过从当前视频块中减去当前视频块的(多个)预测视频块来生成当前视频块的残差数据。当前视频块的残差数据可以包括对应于当前视频块中样点的不同样点分量的残差视频块。

[0243] 在其他示例中,例如在跳过模式下,对于当前视频块可能不存在当前视频块的残差数据,并且残差生成单元1307可以不进行减去操作。

[0244] 变换处理单元1308可以通过将一个或多个变换应用于与当前视频块相关联的残差视频块来生成当前视频块的一个或多个变换系数视频块。

[0245] 在变换处理单元1308生成与当前视频块相关联的变换系数视频块之后,量化单元1309可以基于与当前视频块相关联的一个或多个量化参数(QP)值来量化与当前视频块相关联的变换系数视频块。

[0246] 逆量化单元1310和逆变换单元1311可以将逆量化和逆变换分别应用于变换系数视频块,来从变换系数视频块重建残差视频块。重建单元1312可以将重建的残差视频块添加到来自预测单元1302生成的一个或多个预测视频块的对应样点,以产生与当前块相关联的重建视频块用于存储在缓冲器1313中。

[0247] 在重建单元1312重建视频块之后,可以进行环路滤波操作以降低视频块中视频块量化伪影。

[0248] 熵编码单元1314可以从视频编码器1300的其他功能组件接收数据。当熵编码单元1314接收数据时,熵编码单元1314可以进行一个或多个熵编码操作以生成熵编码数据并且输出包括熵编码数据的比特流。

[0249] 图14是示出视频解码器1400的示例的框图,该视频解码器1400可以是图12中示出的系统1200中的视频解码器1224。

[0250] 视频解码器1400可以被配置为进行本公开的任何或全部技术。在图14的示例中,视频解码器1400包括多个功能组件。本公开所描述的技术可以在视频解码器1400的各种组件之间共享。在一些示例中,处理器可以配置为进行本公开中描述的任何或全部技术。

[0251] 在图14的示例中,视频解码器1400包括熵解码单元1401、运动补偿单元1402、帧内预测单元1409、逆量化单元1404、逆变换单元1405、重建单元1406和缓冲器1407。在一些示例中,视频解码器1400可以进行与关于视频编码器1300(图13)所描述的编码过程总体反演的解码过程。

[0252] 熵解码单元1401可以检索编码比特流。编码比特流可以包括熵编解码视频数据(例如,视频数据的编解码块)。熵解码单元1401可以对熵编解码视频进行解码,并且根据熵编解码视频数据,运动补偿单元1402可以确定包括运动矢量、运动矢量精度、参考图片列表索引和其他运动信息的运动信息。运动补偿单元1402例如可以通过进行AMVP和merge模式确定此类信息。

[0253] 运动补偿单元1402可以产生运动补偿块,可能地基于插值滤波器进行插值。要以子像素精度使用的插值滤波器的标识符可以包括在语法元素中。

[0254] 运动补偿单元1402可以使用由视频编码器1300在编码视频块的期间所使用的插值滤波器,来计算出参考块的子整数个像素的插值的值。运动补偿单元1402可以根据接收的语法信息确定由视频编码器1300所使用的插值滤波器并且使用插值滤波器来产生预测块。

[0255] 运动补偿单元1402可以使用一些语法信息来确定:用于对编码视频序列的(多个)帧和/或(多个)条带进行编码的块的尺寸,描述编码视频序列的每个宏块如何被分割的分割信息,指示如何编码每个分割的模式,每个帧间编码块的一个或多个参考帧(和参

考帧列表),以及对编码视频序列进行解码的其他信息。

[0256] 帧内预测单元1403可以使用例如在比特流中接收的帧内预测模式来从空域相邻块形成预测块。逆量化单元1404逆量化(即去量化)在比特流中提供的且由熵解码单元1401解码的量化的视频块系数。逆变换单元1405应用逆变换。

[0257] 重建单元1406可以用由运动补偿单元1402或帧内预测单元1403生成的对应预测块求和残差块,以形成解码块。如所期望的,去块滤波器还可以应用于滤波解码块以便移除块效应伪影。解码视频块然后存储在缓冲器1407中,该缓冲器1407提供用于随后的运动补偿/帧内预测的参考块,并且还产生用于在显示设备上呈现的解码视频。

[0258] 图15是示例编码器1500的示意图。编码器1500适合于实现VVC技术。编码器1500包括三个环路滤波器,即去块滤波器(DF)1502、样点自适应偏移(SAO)1504和自适应环路滤波器(ALF)2906。与使用预定义滤波器的DF 2902不同,SAO 2904和ALF 2906利用当前图片的原始样点,通过分别添加偏移和应用有限脉冲响应(FIR)滤波器,利用编解码的辅助信息信令通知偏移和滤波器系数,来减小原始样点和重建样点之间的均方误差。ALF 1506位于每个图片的最后处理阶段,并且可以被视为试图捕捉和修复由先前阶段产生的伪影的工具。

[0259] 编码器1500还包括帧内预测组件1508和运动估计/补偿(ME/MC)组件1510,配置为接收输入视频。帧内预测组件1508被配置成执行帧内预测,而ME/MC组件1510被配置成利用从参考图片缓冲器1512获得的参考图片来执行帧间预测。来自帧间预测或帧内预测的残差块被馈送到变换(T)组件1514和量化(Q)组件1516中,以生成量化的残差变换系数,这些系数被馈送到熵编解码组件1518中。熵编解码组件1518对预测结果和量化的变换系数进行熵编解码,并将其向视频解码器(未示出)发送。从量化组件1516输出的量化组件可以被馈送到逆量化(IQ)组件1520、逆变换(IT)组件1522和重建(REC)组件1524。REC组件1524能够将图像输出到DF 1502、SAO 1504和ALF 1506,以便在这些图片被存储在参考图片缓冲器1512中之前进行滤波。

[0260] 以下解决方案显示了本文讨论的技术的示例。

[0261] 1.一种媒体数据处理的方法,包括:为视频的当前块和视频的比特流之间的转换,根据规则确定用于该转换的变换,并根据该确定执行该转换;其中该规则基于当前块的邻近区域的一个或多个特征。

[0262] 2.根据权利要求1所述的方法,其中,邻近区域包括与当前块不紧邻的一个或多个块。

[0263] 3.根据权利要求1-2中任一项所述的方法,其中邻近区域包括与当前块紧邻的一个或多个块。

[0264] 4.根据权利要求1所述的方法,其中,邻近区域包括当前块的上区域或左区域或左上区域或右上区域。

[0265] 5.根据权利要求4所述的方法,其中上区域或左区域或左上区域或右上区域的形状为矩形。

[0266] 6.根据权利要求1-5中任一项所述的方法,其中,邻近区域的一个或多个特征来自邻近区域的一个或多个子区域。

[0267] 7.根据权利要求1-6中任一项所述的方法,其中,通过将邻近的一个或多个子区域分类并从中选择变换核来确定邻近区域的一个或多个特征。

- [0268] 8. 根据权利要求7所述的方法,其中,分类取决于从当前块的邻近区域的子区域提取的特征。
- [0269] 9. 根据权利要求8所述的方法,其中特征包括邻近区域的子区域的边缘特征或平均样点值。
- [0270] 10. 根据权利要求7-9中任一项所述的方法,其中类别中的一类对应于噪声区域类别。
- [0271] 11. 根据权利要求7-10中任一项所述的方法,其中分类的类别是基于确定子区域的当前像素是否是跳转点。
- [0272] 12. 根据权利要求7-11中任一项所述的方法,其中类别中的一类对应于平滑区域类别。
- [0273] 13. 根据权利要求7-12中任一项所述的方法,其中分类的类别是基于确定子区域的当前像素是否是正常点。
- [0274] 14. 根据权利要求7-13中任一项所述的方法,其中类别中的一类别对应于边缘区域类别。
- [0275] 15. 根据权利要求7-14中任一项所述的方法,其中分类的类别是基于确定子区域的当前像素是否是边缘像素。
- [0276] 16. 根据权利要求1-15中任一项所述的方法,其中,规则基于邻近区域的辅助信息。
- [0277] 17. 根据权利要求16所述的方法,其中,辅助信息包括邻近区域的块尺寸或预测模式。
- [0278] 18. 根据权利要求16所述的方法,其中辅助信息用于确定水平和/或垂直方向上的变换。
- [0279] 19. 根据权利要求1-18中任一项所述的方法,其中,规则定义了用于第一颜色分量的转换的变换基于第一颜色分量的邻近区域的一个或多个特征。
- [0280] 20. 根据权利要求1-18中任一项所述的方法,其中,规则定义了用于第一颜色分量的转换的变换基于与第一颜色分量不同的第二颜色分量的邻近区域的一个或多个特征。
- [0281] 21. 根据权利要求1-19中任一项所述的方法,其中,规则还规定响应于当前块的维度来执行该确定。
- [0282] 22. 根据权利要求1-19中任一项所述的方法,其中,规则还规定指示关于转换的规则的应用的语法元素。
- [0283] 23. 根据权利要求1-21中任一项所述的方法,其中,该转换包括从比特流中生成视频或者从视频中生成比特流。
- [0284] 24. 一种在计算机可读介质上存储比特流的方法,包括根据权利要求1-22中任一项或多项所述的方法生成比特流,并将该比特流存储在计算机可读介质上。
- [0285] 25. 一种其上存储有视频的比特流的计算机可读介质,该比特流在被视频解码器的处理器处理时使得视频解码器生成视频,其中该比特流是根据权利要求1-23中的一项或多项所述的方法生成的。
- [0286] 26. 一种视频解码装置,包括被配置为实现权利要求1-23中一项或多项所述的方法的处理器。

[0287] 27.一种视频编码装置,包括被配置为实现权利要求1-23中一项或多项所述的方法的处理器。

[0288] 28.一种其上存储有计算机代码的计算机程序产品,该代码在由处理器执行时使处理器实现权利要求1-23中任一项所述的方法。

[0289] 29.一种计算机可读介质,其上存储有符合根据权利要求1-23中任一项生成的比特流格式的比特流。

[0290] 30.一种方法、一种装置、一种根据本文档中描述的公开方法或系统生成的比特流。

[0291] 在本文所述的解决方案中,编码器可通过根据格式规则产生编解码表示来符合格式规则。在本文描述的解决方案中,解码器可以使用格式规则来解析编解码表示中的语法元素,根据格式规则知道语法元素的存在和不存在,以产生解码的视频。

[0292] 在本文件中,术语“视频处理”可以指视频编码、视频解码、视频压缩或视频解压缩。例如,视频压缩算法可以在从视频的像素表示转换为对应的比特流表示期间应用,反之亦然。例如,如语法所定义的,当前视频块的比特流表示可以对应于在比特流内共位的或分布在不同位置的比特。例如,宏块可以根据变换和编解码的误差残差值进行编码,并且还可以在比特流中的标头和其他字段中使用比特。此外,在转换期间,解码器可以基于该确定,在知道一些字段可能存在或不存在的条件下解析比特流,如以上解决方案中所述。类似地,编码器可确定包括或不包括某些语法字段,并通过从编解码表示中包括或排除语法字段来相应地生成编解码表示。

[0293] 本文件中描述的公开和其他解决方案、示例、实施例、模块和功能操作可以在数字电子电路、或计算机软件、固件或硬件中实现,包括本文件中所公开的结构及其结构等效体,或其中一个或多个的组合。公开的内容和其他实施例可以实施为一个或多个计算机程序产品,即一个或多个编码在有形的且非易失的计算机可读介质上的计算机程序指令的模块,以供数据处理装置执行或控制数据处理装置的操作。计算机可读介质可以是机器可读存储设备、机器可读存储基板、存储设备、影响机器可读传播信号的物质组成或其中一个或多个的组合。术语“数据处理单元”或“数据处理装置”包括用于处理数据的所有装置、设备和机器,包括例如可编程处理器、计算机或多处理器或计算机组。除硬件外,该装置还可以包括为计算机程序创建执行环境的代码,例如,构成处理器固件的代码、协议栈、数据库管理系统、操作系统或其中一个或多个的组合。传播的信号是人为生成的信号,例如,机器生成的电、光或电磁信号,其被生成以编码信息以传输到合适的接收器设备。

[0294] 计算机程序(也称为程序、软件、软件应用、脚本或代码)可以用任何形式的编程语言(包括编译语言或解释语言)编写,并且可以以任何形式部署,包括作为独立程序或作为模块、组件、子程序或其他适合在计算环境中使用的单元。计算机程序不一定与文件系统中的文件对应。程序可以存储在保存其他程序或数据的文件的部分中(例如,存储在标记语言文档中的一个或多个脚本)、专用于该程序的单个文件中、或多个协调文件(例如,存储一个或多个模块、子程序或部分代码的文件)中。计算机程序可以部署在一台或多台计算机上来执行,这些计算机位于一个站点上或分布在多个站点上,并通过通信网络互连。

[0295] 本文件中描述的处理和逻辑流可以通过一个或多个可编程处理器执行,该处理器执行一个或多个计算机程序,通过在输入数据上操作并生成输出来执行功能。处理和逻辑

流也可以通过特殊用途的逻辑电路来执行,并且装置也可以实现为特殊用途的逻辑电路,例如,FPGA(现场可编程门阵列)或ASIC(专用集成电路)。

[0296] 例如,适于执行计算机程序的处理器包括通用和专用微处理器,以及任何类型数字计算机的任何一个或多个。通常,处理器将从只读存储器或随机存取存储器或两者接收指令和数据。计算机的基本组件是执行指令的处理器和存储指令和数据的一个或多个存储设备。通常,计算机还将包括一个或多个用于存储数据的大容量存储设备,例如,磁盘、磁光盘或光盘,或者可操作地耦合以从大容量存储设备接收数据或者将数据传输到大容量存储设备,或者以上两者。然而,计算机不一定具有这样的设备。适用于存储计算机程序指令和数据的计算机可读介质包括所有形式的非易失性存储器、介质和存储器设备,包括例如半导体存储器设备,例如可擦除可编程只读存储器 (EPROM)、电可擦除可编程只读存储器 (EEPROM) 和闪速存储器设备;磁盘,例如内部硬盘或可移动磁盘;磁光盘;和光盘只读存储器 (CD ROM) 和数字多功能光盘只读存储器 (DVD-ROM) 盘。处理器和存储器可以由专用逻辑电路补充,或者合并到专用逻辑电路中。

[0297] 虽然本专利文件包含许多细节,但不应将其解释为对任何发明或权利要求范围的限制,而应解释为对特定发明的特定实施例的特征的描述。本专利文件在单独实施例的上下文描述的某些特征也可以在单个实施例中组合实施。相反,在单个实施例的上下文中描述的各种功能也可以在多个实施例中单独实施,或在任何合适的子组合中实施。此外,尽管上述特征可以描述为在某些组合中起作用,甚至最初要求是这样,但在某些情况下,可以从组合中移除权利要求组合中的一个或多个特征,并且权利要求的组合可以指向子组合或子组合的变体。

[0298] 同样,尽管附图中以特定顺序描述了操作,但这不应理解为要获得想要的结果必须按照所示的特定顺序或顺序执行此类操作,或执行所有说明的操作。此外,本专利文件实施例中各种系统组件的分离不应理解为在所有实施例中都需要这样的分离。

[0299] 仅描述了一些实现和示例,其他实现、增强和变体可以基于本专利文件中描述和说明的内容做出。

[0300] 当除了第一组件和第二组件之间的线、迹线或另一种介质之外,不存在中间组件时,第一组件直接耦合至第二组件。当在第一组件和第二组件之间存在除了线、迹线或另一种介质之外的中间组件时,第一组件间接耦合到第二组件。术语“耦合”及其变体包括直接耦合和间接耦合。除非另有说明,术语“约”的使用意味着包括数值的上下10%的范围。

[0301] 虽然本公开中提供了若干实施例,但应理解,在不脱离本公开的精神或范围的情况下,所公开的系统和方法可体现为许多其他特定形式。当前的示例被认为是说明性的而非限制性的,并且不限于其给出的细节。例如,各种元件或组件可以组合或集成在另一个系统中,或者可以省略或不实现某些特征。

[0302] 此外,在不脱离本公开的范围的情况下,各种实施例中描述和说明的离散或单独的技术、系统、子系统和方法可与其他系统、模块、技术或方法组合或集成。被示出或讨论为耦合的其他项目可以直接连接,或者可以通过一些接口、设备或中间组件以电、机械或其他方式间接耦合或通信。本领域的技术人员可以在不脱离这里公开的精神和范围的情况下确定改变、替换和变更的其他示例。

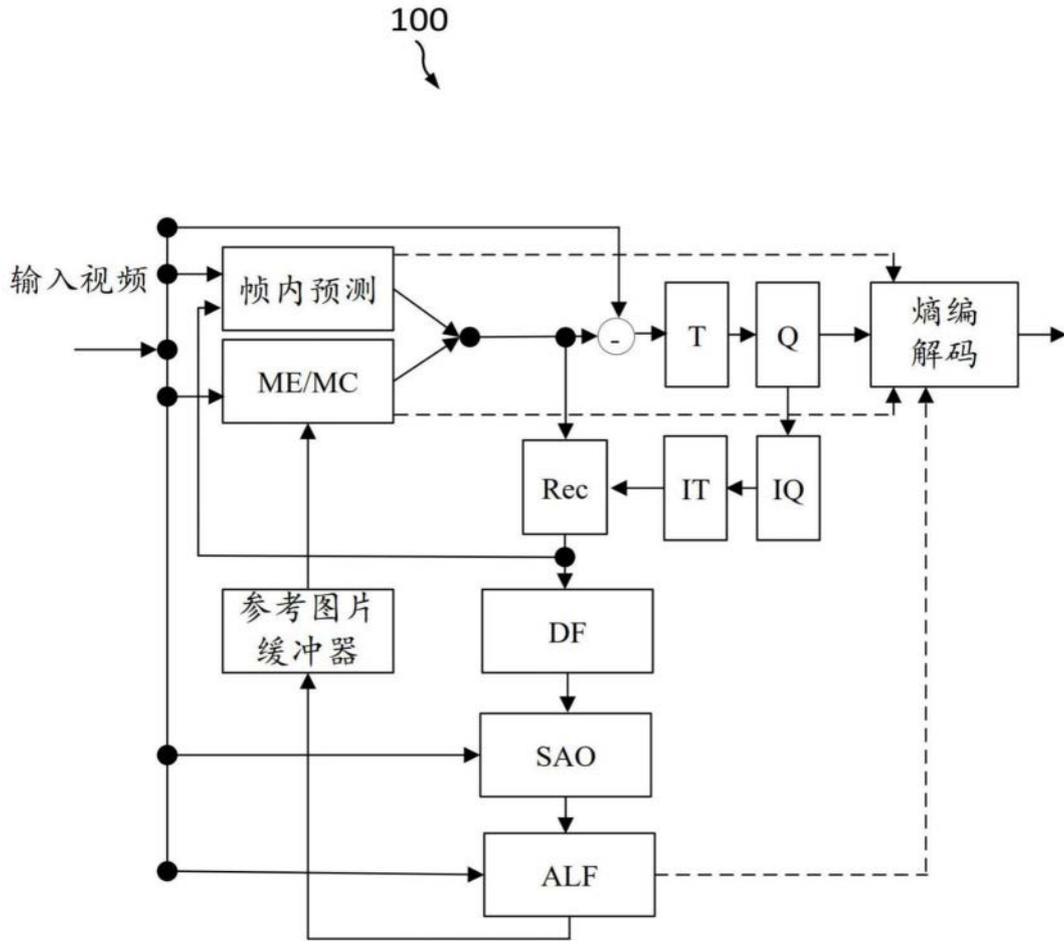


图1

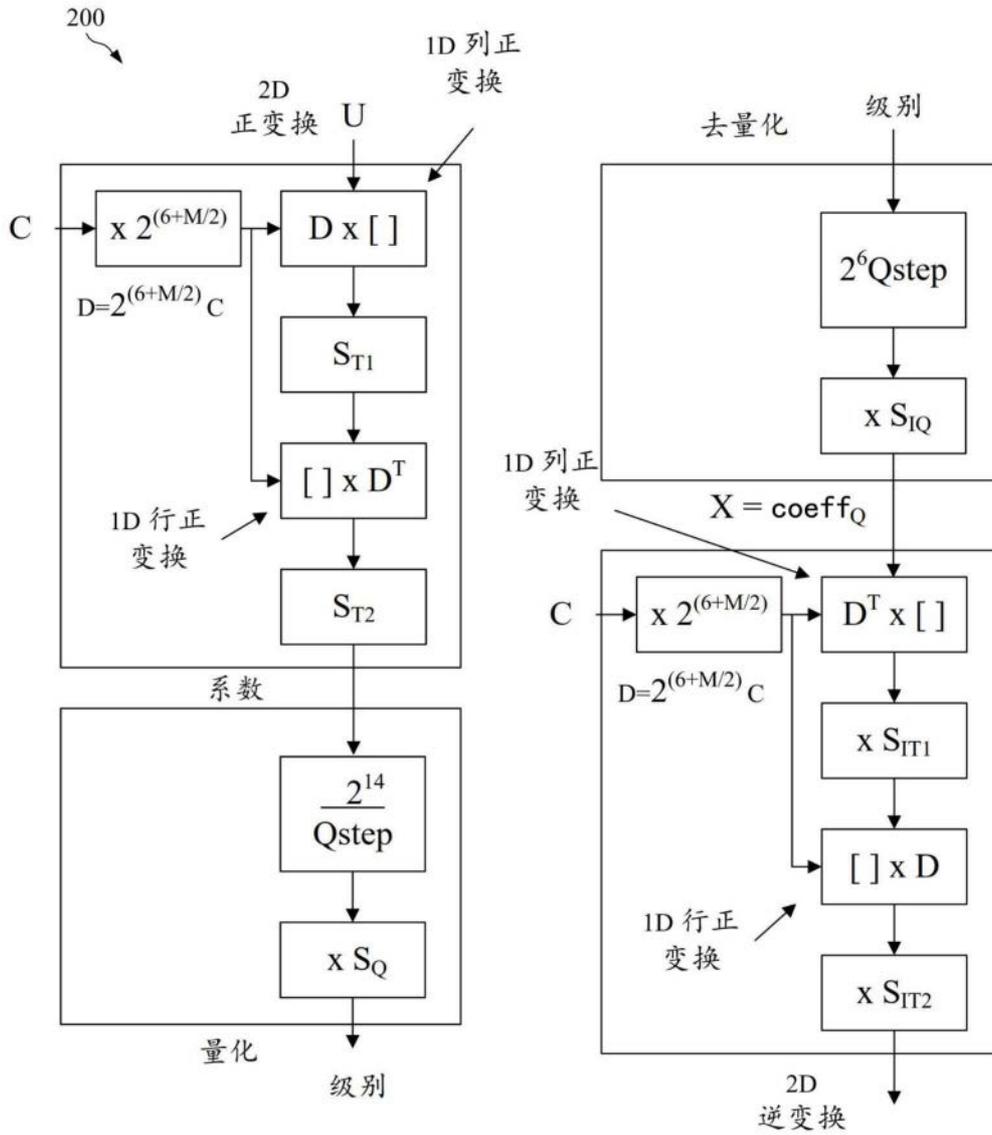


图2

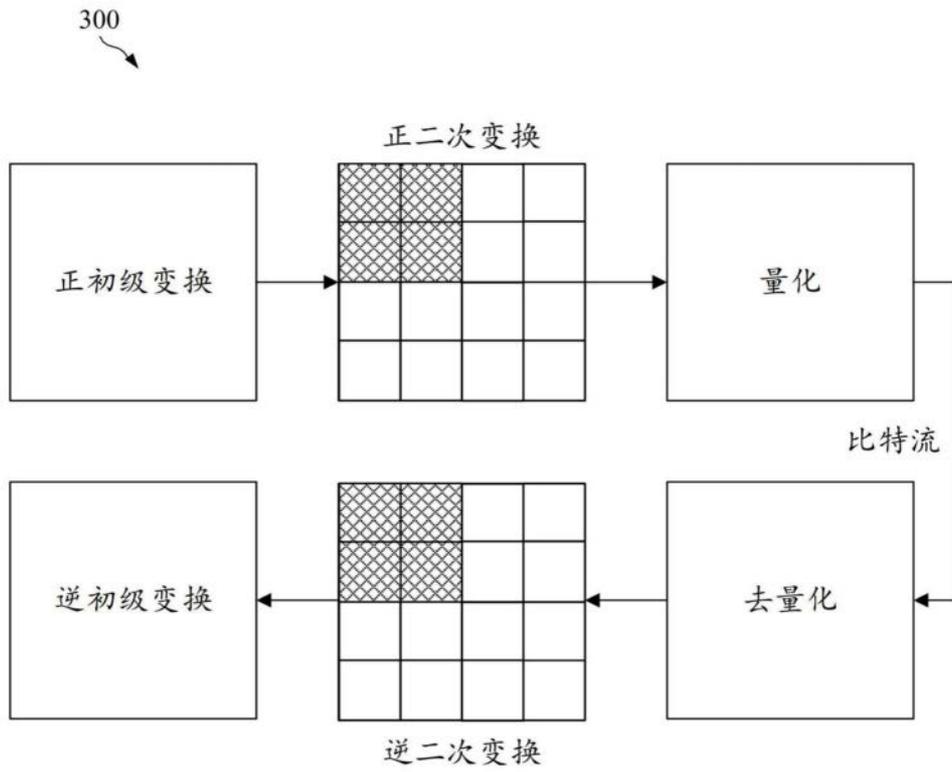


图3

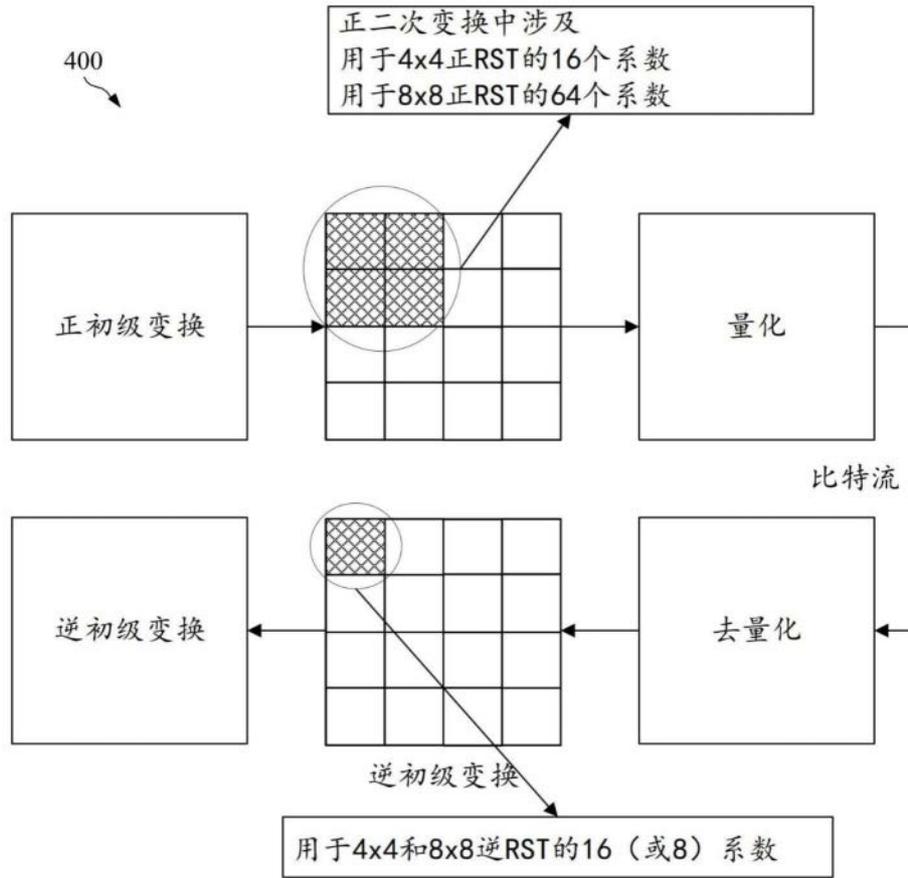
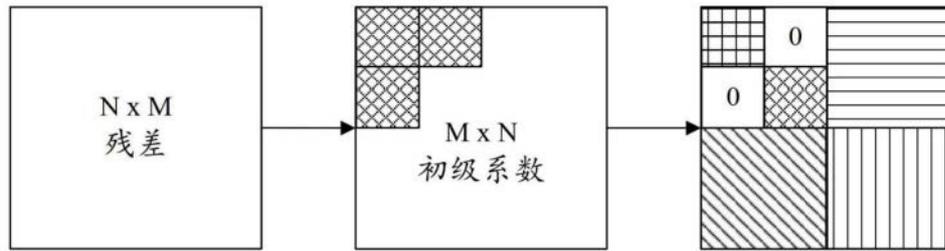


图4



图5

600



2D 正初级  
变换

正二次  
变换

-  4x4 初级系数
-  4x4 二次系数
-  (M-8)x8 右上初级系数
-  8x(N-8) 左下初级系数
-  (M-8)x(N-8) 右下初级系数

图6

700  
↘

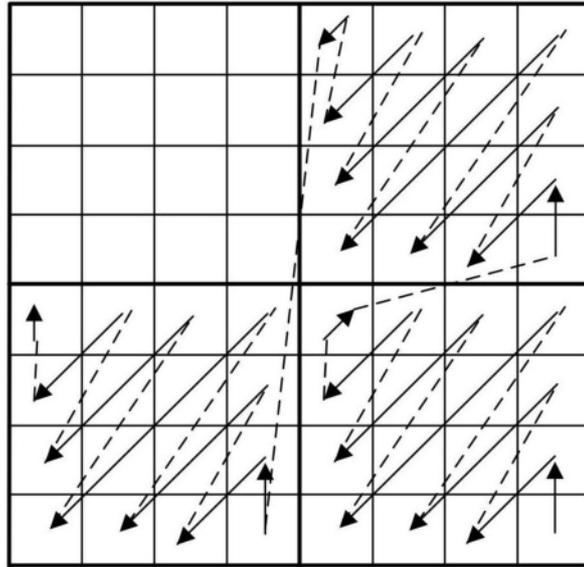


图7

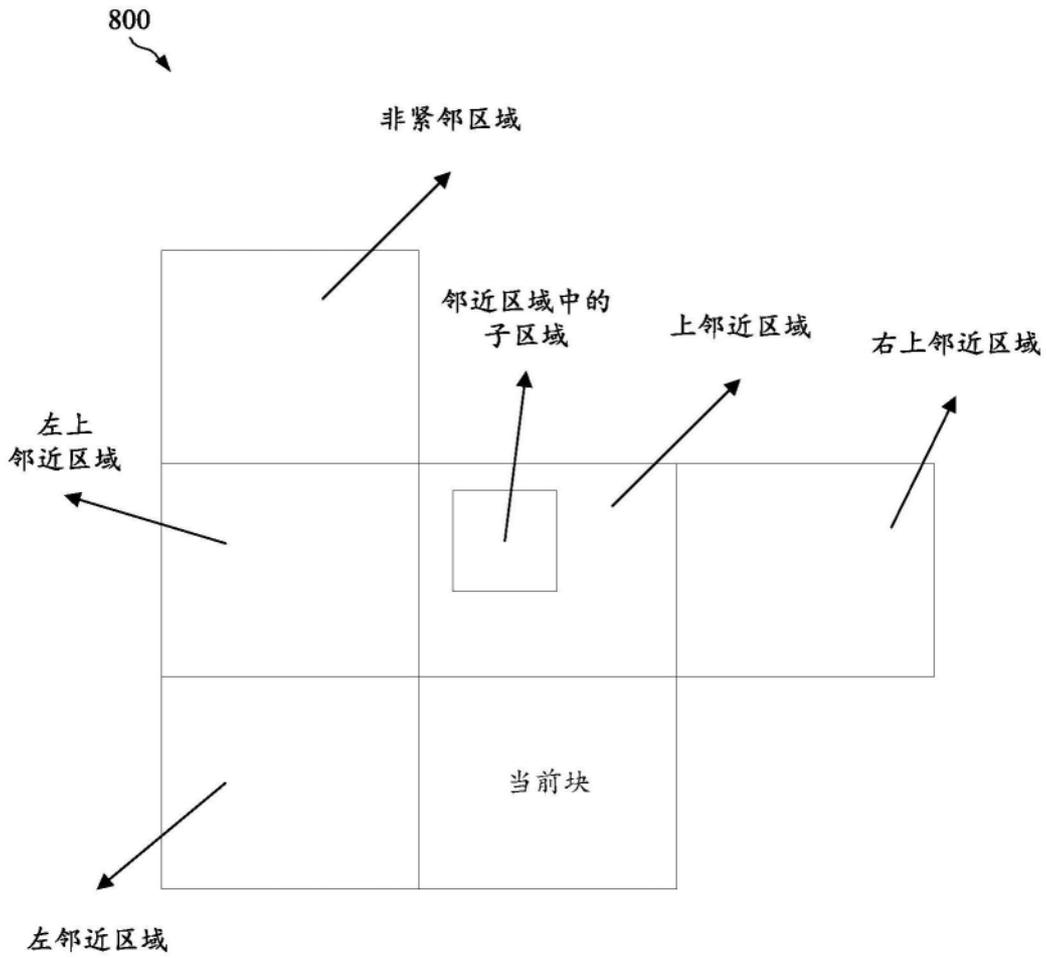


图8

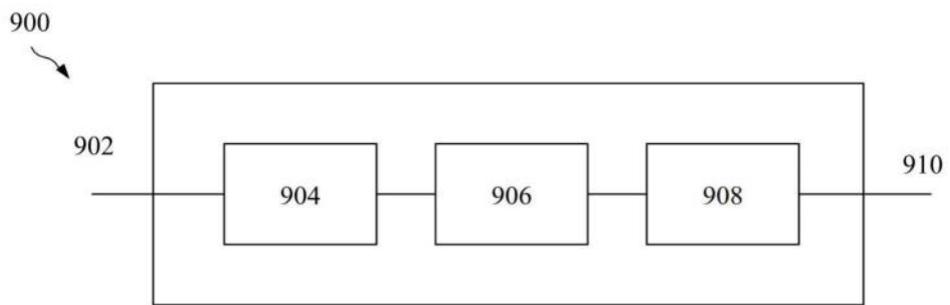


图9

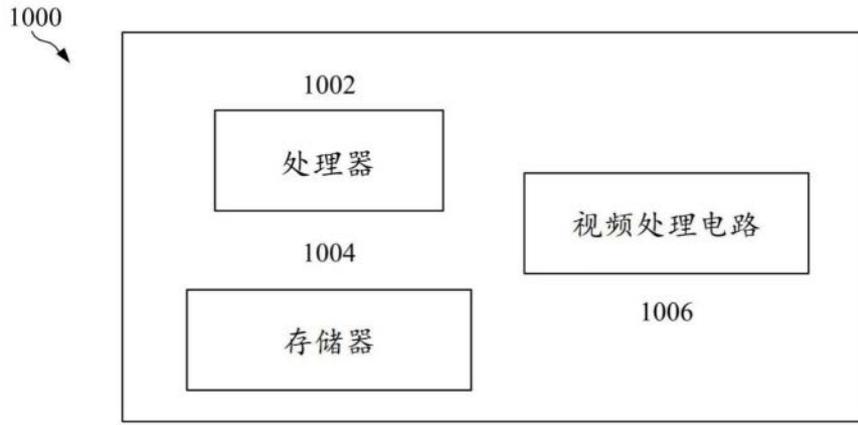


图10

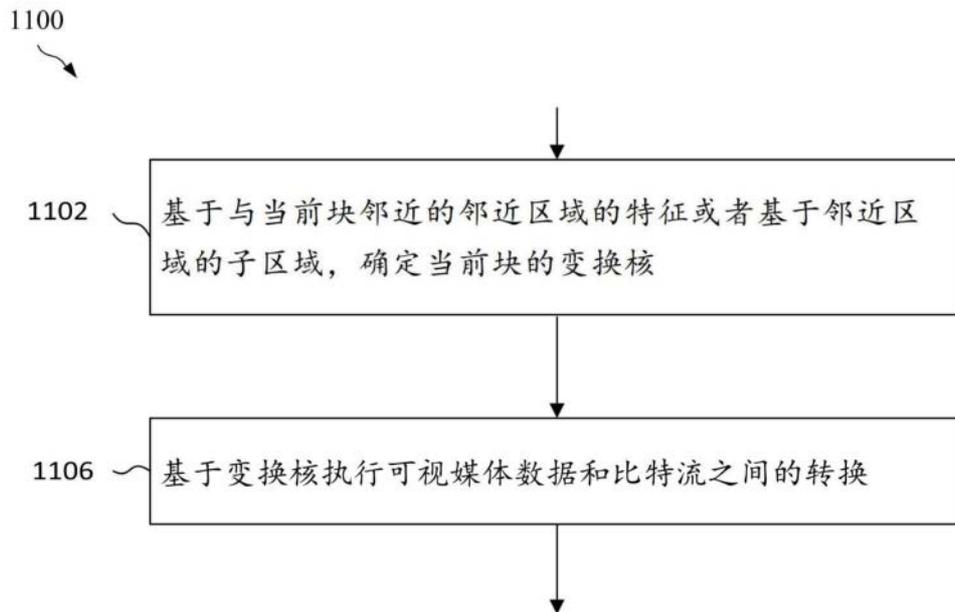


图11

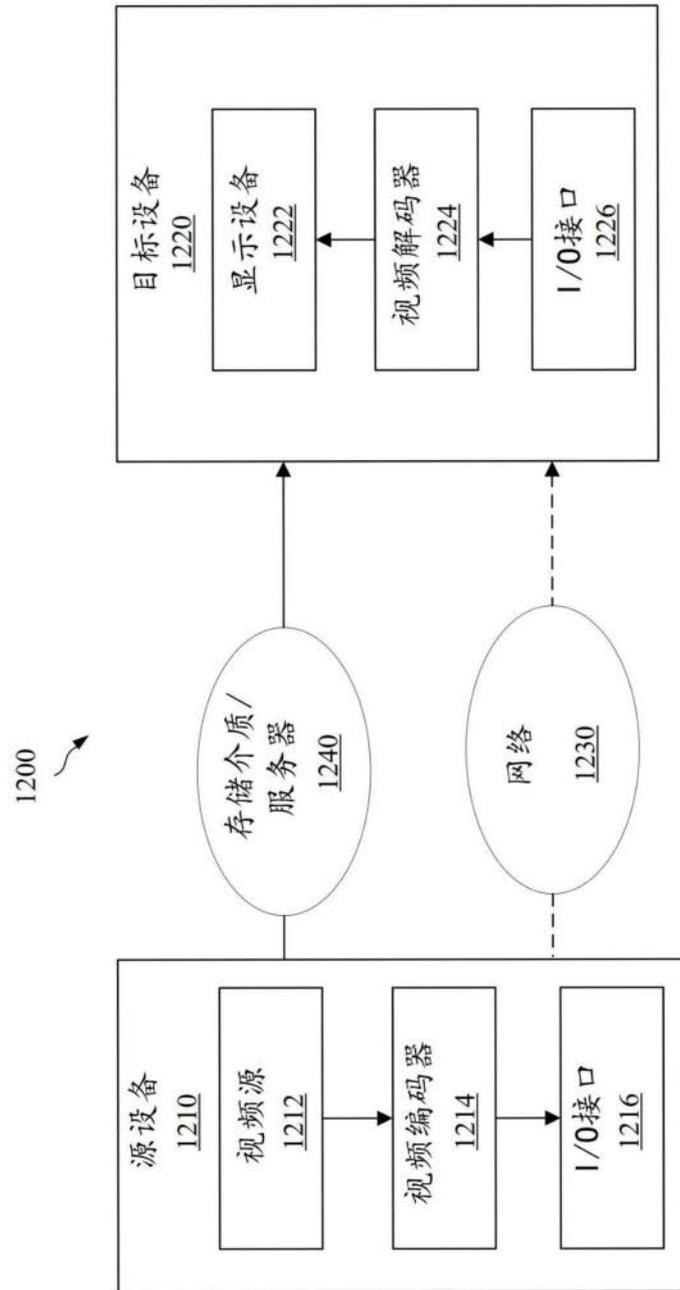


图12

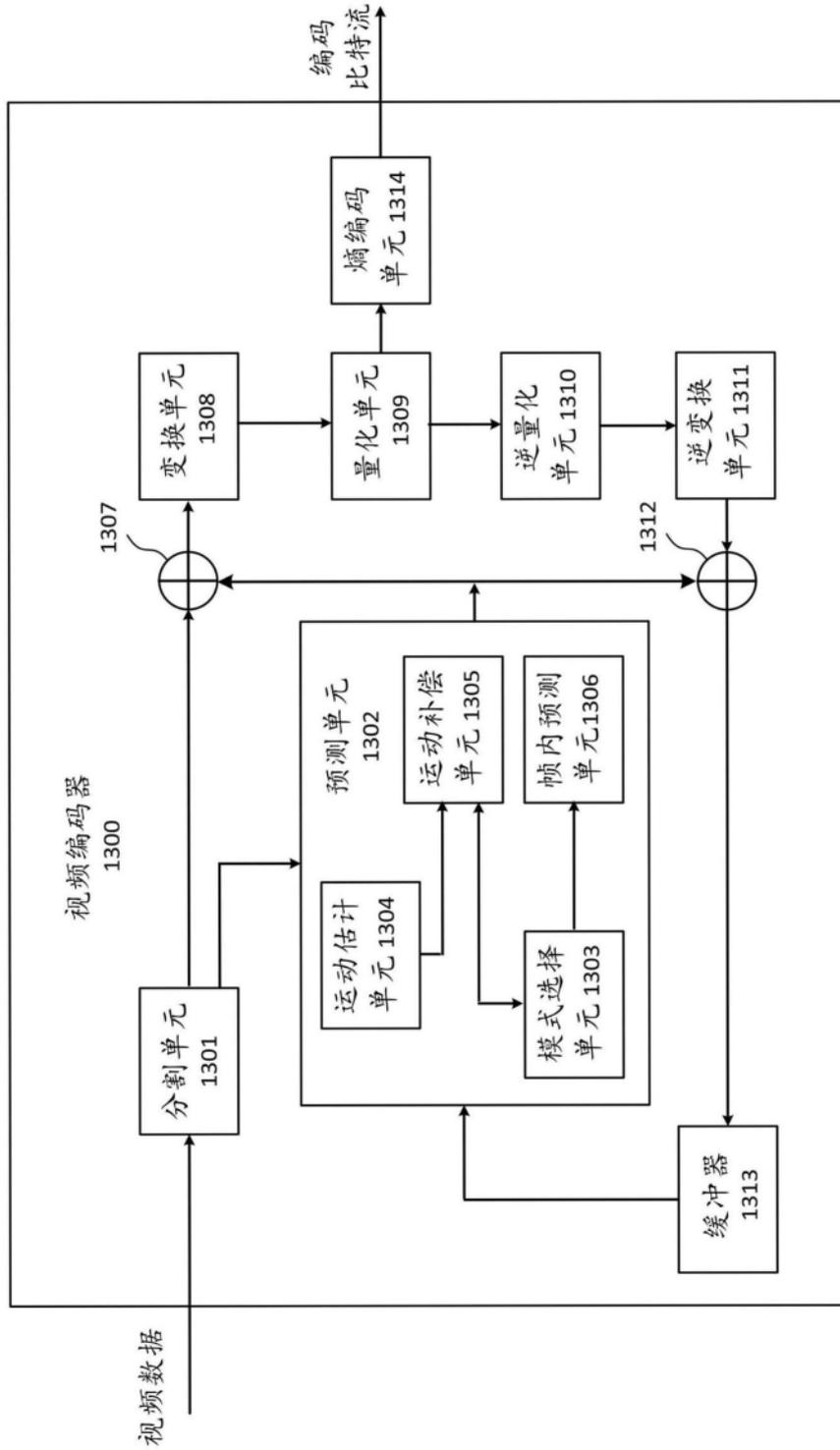


图13

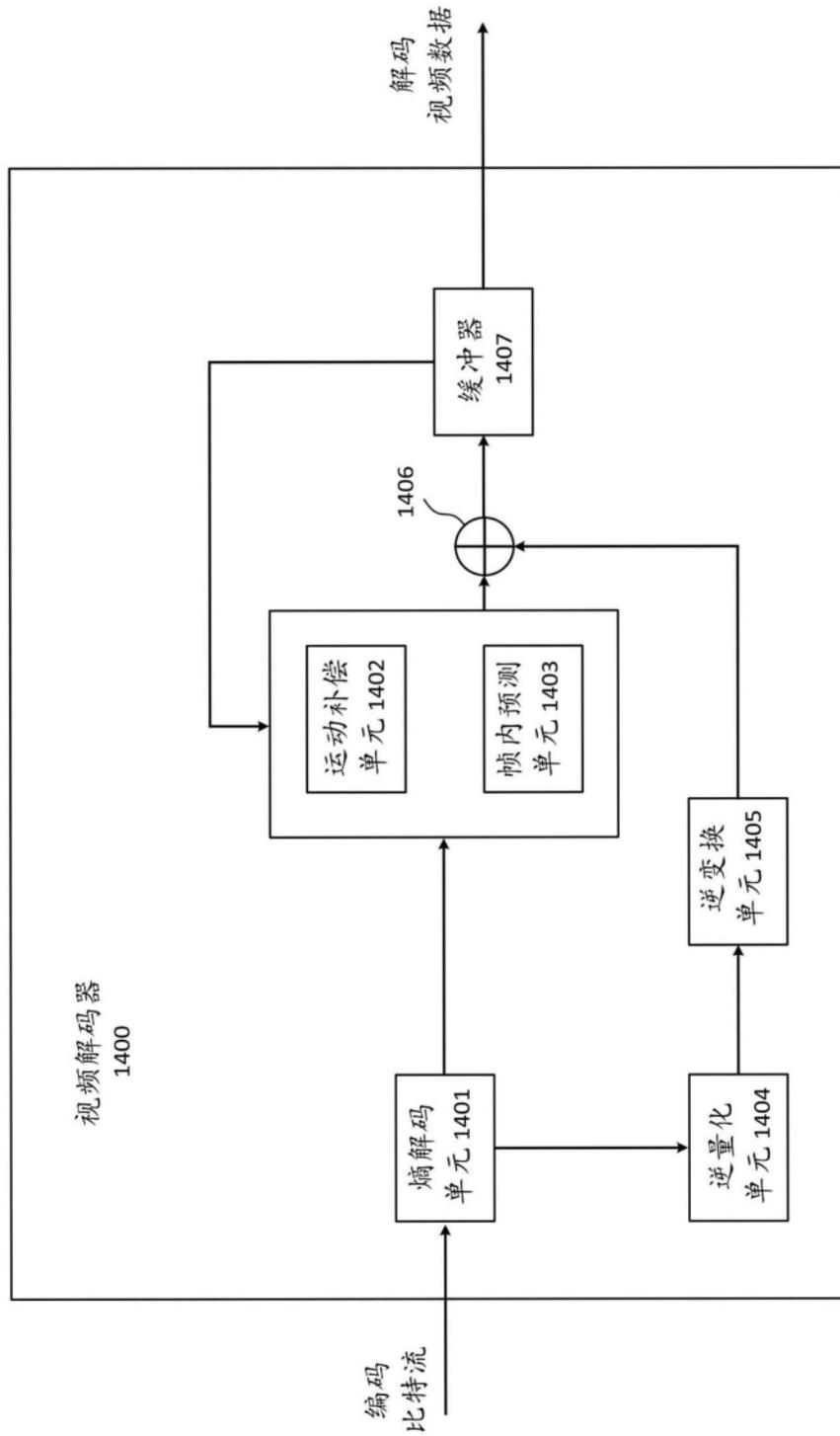


图14

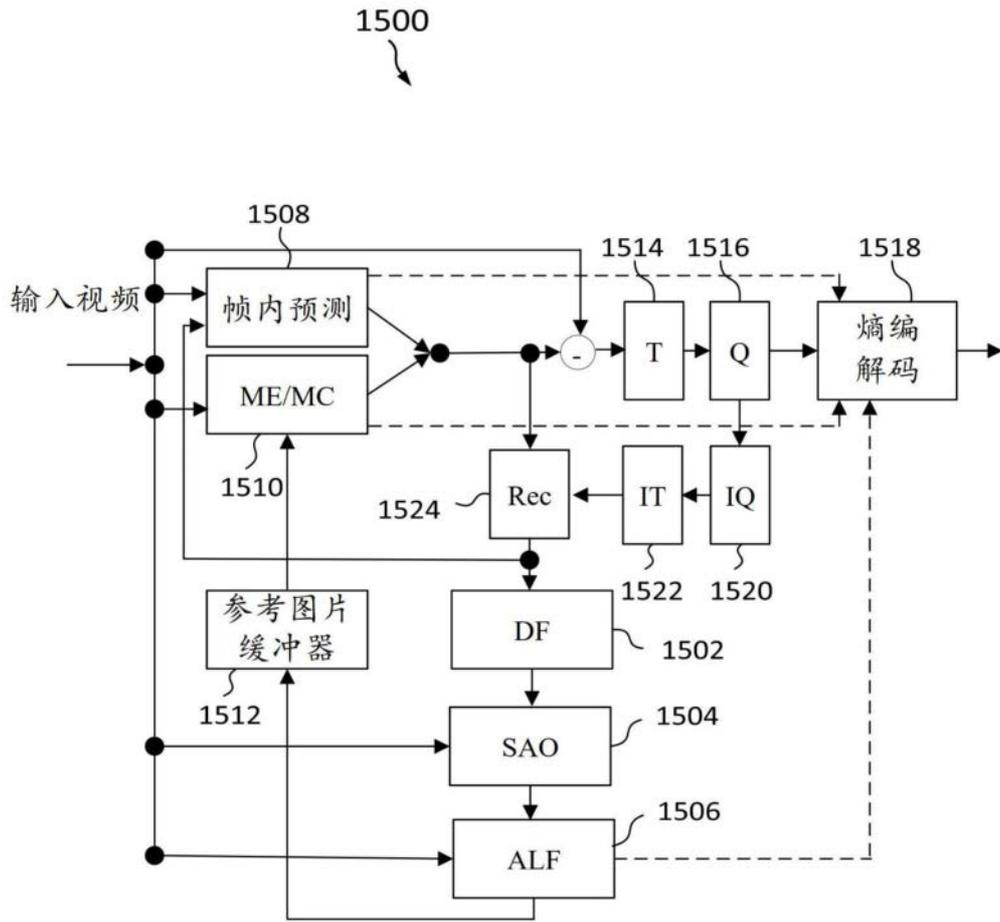


图15