



(12)发明专利

(10)授权公告号 CN 106134199 B

(45)授权公告日 2019.05.31

(21)申请号 201580013365.0

(22)申请日 2015.03.12

(65)同一申请的已公布的文献号
申请公布号 CN 106134199 A

(43)申请公布日 2016.11.16

(30)优先权数据
14/216,369 2014.03.17 US

(85)PCT国际申请进入国家阶段日
2016.09.12

(86)PCT国际申请的申请数据
PCT/US2015/020237 2015.03.12

(87)PCT国际申请的公布数据
W02015/179010 EN 2015.11.26

(73)专利权人 高通股份有限公司
地址 美国加利福尼亚州

(72)发明人 钟仁肃 于洋 王翔林
马尔塔·卡切维奇

(74)专利代理机构 北京律盟知识产权代理有限
责任公司 11287
代理人 宋献涛

(51)Int.Cl.
H04N 19/60(2014.01)
H04N 19/42(2014.01)

(56)对比文件
CN 1535030 A, 2004.10.06, 全文.
CN 1545211 A, 2004.11.10, 全文.
CN 101860661 A, 2010.10.13, 全文.
US 2002/0014455 A1, 2012.01.19, 全文.
US 2007/0297502 A1, 2007.12.27, 全文.

审查员 刘晶

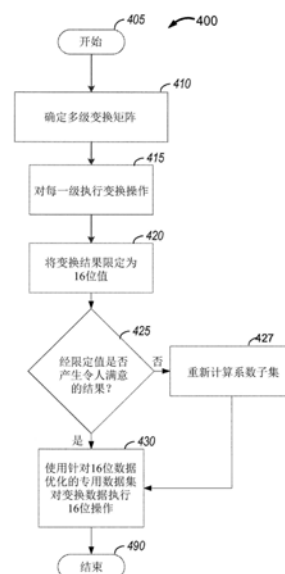
权利要求书3页 说明书20页 附图5页

(54)发明名称

用于使用基于网格的计算的低复杂度正变换的系统和方法

(57)摘要

本文中描述用于使用基于网格的计算的低复杂度正变换的系统和方法。本发明中所描述的标的物的一个方面提供包括经配置以存储视频信息的存储器的视频编码器。所述视频编码器进一步包括与所述存储器通信的处理器。所述处理器经配置以将变换分解成多个变换级。所述处理器经进一步配置以使用所述多个级变换所述视频信息,从而确定每一变换级处的变换级输出。所述处理器经进一步配置以将每一变换级处的所述变换级输出限定到预定位深度。所述处理器经进一步配置以对所述多个级中的最末级的经限定变换输出执行操作,其中所述操作仅可供与具有所述预定位深度的数据一起使用。



1. 一种视频编码器,其包括:
经配置以存储视频数据块的存储器;及
与所述存储器通信的处理器,所述处理器经配置以:
将用于将所述视频数据块从像素域变换到系数域的变换分解成多个变换级;
通过将所述多个变换级应用于所述视频数据块来将所述视频数据块从所述像素域变换到所述系数域,从而在每一变换级处产生输出系数集合;
将每一变换级处的所述输出系数集合限定到预定位深度;
确定每一变换级处的经限定系数集合与所述输出系数集合之间的偏差;
当经确定偏差大于预定阈值时重新计算变换级的所述输出系数集合的子集;及
对所述多个变换级中的最末级的经限定输出系数执行操作,其中所述操作的位深度等于所述预定位深度。
2. 根据权利要求1所述的视频编码器,其中所述处理器经进一步配置以使用基于网格的方法、蝶形方法或奇偶分解来分解所述变换。
3. 根据权利要求1所述的视频编码器,其中所述处理器经进一步配置以通过削减所述系数来限定每一变换级处的所述输出系数集合,使得经削减系数必须在预定范围内。
4. 根据权利要求1所述的视频编码器,其中所述预定位深度为16位。
5. 根据权利要求1所述的视频编码器,其中所述偏差包括所述经限定系数集合与所述输出系数集合之间的差。
6. 根据权利要求1所述的视频编码器,其中所述处理器经进一步配置以使用至少经重新计算系数子集来变换所述视频数据块。
7. 一种编码视频的方法,所述方法包括:
存储视频数据块;
将用于将所述视频数据块从像素域变换到系数域的变换分解成多个变换级;
通过将所述多个变换级应用于所述视频数据块来将所述视频数据块从所述像素域变换到所述系数域,从而在每一变换级处产生输出系数集合;
将每一变换级处的所述输出系数集合限定到预定位深度;
确定每一变换级处的经限定系数集合与所述输出系数集合之间的偏差;
当经确定偏差大于预定阈值时重新计算变换级的所述输出系数集合的子集;及
对所述多个变换级中的最末级的经限定输出系数执行操作,其中所述操作的位深度等于所述预定位深度。
8. 根据权利要求7所述的方法,其进一步包括使用基于网格的方法、蝶形方法或奇偶分解来分解所述变换。
9. 根据权利要求7所述的方法,其进一步包括通过削减所述系数来限定每一变换级处的所述输出系数集合,使得经削减系数必须在预定范围内。
10. 根据权利要求7所述的方法,其中所述预定位深度为16位。
11. 根据权利要求7所述的方法,其中所述偏差包括所述经限定系数集合与所述输出系数集合之间的差。
12. 根据权利要求7所述的方法,其进一步包括使用至少经重新计算系数子集来变换所述视频数据块。

13. 一种包括代码的非暂时性计算机可读媒体, 所述代码在经执行时使得设备进行以下操作:

存储视频数据块;

将用于将所述视频数据块从像素域变换到系数域的变换分解成多个变换级;

通过将所述多个变换级应用于所述视频数据块来将所述视频数据块从所述像素域变换到所述系数域, 从而在每一变换级处产生输出系数集合;

将每一变换级处的所述输出系数集合限定到预定位深度;

确定每一变换级处的经限定系数集合与所述输出系数集合之间的偏差;

当经确定偏差大于预定阈值时重新计算变换级的所述输出系数集合的子集; 及

对所述多个变换级中的最末级的经限定输出系数执行操作, 其中所述操作的位深度等于所述预定位深度。

14. 根据权利要求13所述的计算机可读媒体, 其进一步包括代码, 所述代码在经执行时使得所述设备使用基于网格的方法、蝶形方法或奇偶分解来分解所述变换。

15. 根据权利要求13所述的计算机可读媒体, 其进一步包括代码, 所述代码在经执行时使得所述设备通过削减所述系数来限定每一变换级处的所述输出系数集合, 使得经削减系数必须在预定范围内。

16. 根据权利要求13所述的计算机可读媒体, 其中所述偏差包括所述经限定系数集合与所述输出系数集合之间的差。

17. 根据权利要求13所述的计算机可读媒体, 其进一步包括代码, 所述代码在经执行时使得所述设备使用至少经重新计算系数子集来变换所述视频数据块。

18. 一种用于编码视频的设备, 其包括:

用于存储视频数据块的装置;

用于将变换分解成多个变换级的装置, 其中所述变换用于将所述视频数据块从像素域变换到系数域;

用于通过将所述多个变换级应用于所述视频数据块来将所述视频数据块从所述像素域变换到所述系数域, 从而在每一变换级处产生输出系数集合的装置;

用于将每一变换级处的所述输出系数集合限定到预定位深度的装置;

用于确定每一变换级处的经限定系数集合与所述输出系数集合之间的偏差的装置;

用于当经确定偏差大于预定阈值时重新计算变换级的所述输出系数集合的子集的装置; 及

用于对所述多个变换级中的最末级的经限定输出系数执行操作的装置, 其中所述操作的位深度等于所述预定位深度。

19. 根据权利要求18所述的设备, 其进一步包括用于使用基于网格的方法、蝶形方法或奇偶分解来分解所述变换的装置。

20. 根据权利要求18所述的设备, 其进一步包括用于通过削减所述系数来限定每一变换级处的所述输出系数集合的装置, 使得经削减系数必须在预定范围内。

21. 根据权利要求18所述的设备, 其中所述偏差包括所述经限定系数集合与所述输出系数集合之间的差。

22. 根据权利要求18所述的设备, 其进一步包括用于使用至少经重新计算系数子集来

变换所述视频数据块的装置。

用于使用基于网格的计算的低复杂度正变换的系统和方法

技术领域

[0001] 本发明涉及视频编码。

背景技术

[0002] 数字视频能力可以并入到多种多样的装置中,包含数字电视、数字直播系统、无线广播系统、个人数字助理(PDA)、膝上型或桌上型计算机、平板计算机、电子图书阅读器、数码摄像机、数字记录装置、数字媒体播放器、视频游戏装置、视频游戏控制台、蜂窝式或卫星无线电电话、智能电话、视频电话会议装置、视频流装置及其类似者。数字视频装置实施视频译码技术,例如描述于以下各者中的那些技术:由MPEG-2、MPEG-4、ITU-T H.263、ITU-T H.264/MPEG-4第10部分高级视频译码(AVC)定义的标准、高效率视频译码(HEVC)及此些标准的扩展。视频装置可通过实施此类视频译码技术而更有效率地发射、接收、编码、解码和/或存储数字视频信息。

[0003] 视频译码技术包含空间(图片内)预测和/或时间(图片间)预测以减少或移除视频序列中固有的冗余。对于基于块的视频译码来说,视频切片(例如,视频帧或视频帧的一部分)可分割成视频块,视频块还可被称作树块、译码单元(CU)和/或译码节点。CU可进一步分割成一或多个预测单元(PU)以确定针对CU的预测视频数据。视频压缩技术还可将CU分割为残余视频块数据的一或多个变换单元(TU),所述残余视频块数据表示待译码视频块与预测视频数据之间的差。例如二维离散余弦变换(DCT)等线性变换可应用于TU以将残余视频块数据从像素域变换到频域以实现进一步压缩。此外,图片的经帧内译码(I)切片中的视频块可使用相对于同一图片中的相邻块中的参考样本的空间预测来编码。图片的经帧间译码(P或B)切片中的视频块可使用关于同一图片中的相邻块中的参考样本的空间预测或关于其它参考图片中的参考样本的时间预测。图片可被称作帧,且参考图片可被称作参考帧。

[0004] 空间或时间预测产生待译码块的预测块。残余数据表示待译码原始块与预测块之间的像素差。经帧间译码块是根据指向形成预测块的参考样本块的运动向量和指示经译码块与预测块之间的差的残余数据编码的。根据帧内译码模式和残余数据而对经帧内译码块进行编码。为了进一步压缩,可将残余数据从像素域变换到变换域,从而产生残余变换系数,可接着量化所述残余变换系数。可扫描最初布置成二维阵列的经量化变换系数,以便产生变换系数的一维向量,且可应用熵编码以实现更多压缩。

[0005] 在更早期视频标准(例如,AVC)中,正变换及反变换大小(例如, 4×4 及 8×8)并不充当视频编码性能的障碍物。但是,更现代的HEVC标准利用达 16×16 及 32×32 正变换及反变换大小,所述大小充当HEVC过程的限制因素。当从像素域变换到系数域时,更大变换需要更多复杂度及周期来处理。在所关注的译码效率中,标准将得益于将视频编码器中的较大正变换向量分解成多个级(例如,“基于网格的方法”,“蝶形方法”或“奇偶分解”)并限定每一级处的内部深度度的过程。本文中所揭示的技术的一些优点涉及在视频编码期间通过将视频编码器中的较大正变换向量分解成多个级并限定每一级处的内部深度度而改善译码效率并降低计算资源要求。

发明内容

[0006] 一般来说,本发明描述涉及通过将较大正变换分解成多个级(例如,实施正变换的基于网格的方法)且将每一级处的内部位深度限定到可适应计算上有效的指令集的水平来改善视频编码性能的技术。举例来说,虽然变换的输入位深度可为9位且起始内部位深度可大于16位,但视频编码器可经配置以使用饱和逻辑将变换的内部位深度限定到16位(例如,将内部值削减到16位)。为了防止视频质量的降低,视频编码器可经进一步配置以测量偏差(例如,误差)水平,比较所述水平与阈值,且在偏差水平超出阈值的情况下重新计算经变换系数的子集。

[0007] 附图和以下描述中陈述一或多个实例的细节。其它特征、目标及优点将从所述描述及图式以及权利要求书而显而易见。

[0008] 本发明中所描述的标的物的一个方面提供包括经配置以存储视频信息的存储器的视频编码器。视频编码器进一步包括与存储器通信的处理器。处理器经配置以将变换分解成多个变换级。处理器经进一步配置以使用多个级变换视频信息,从而确定每一变换级处的变换级输出。处理器经进一步配置以将每一变换级处的变换级输出限定到预定位深度。处理器经进一步配置以对多个级中的最末级的经限定变换输出执行操作,其中所述操作仅可供与具有预定位深度的数据一起使用。

[0009] 本发明中描述的标的物的另一方面提供一种编码视频的方法。所述方法包含存储视频信息。所述方法进一步包含将变换分解成多个变换级。所述方法进一步包含使用多个级变换视频信息以确定每一变换级处的变换级输出。所述方法进一步包含将每一变换级处的变换级输出限定到预定位深度。所述方法进一步包含对多个级中的最末级的经限定变换输出执行操作,其中所述操作仅可供与具有预定位深度的数据一起使用。

[0010] 本发明中描述的标的物的另一方面提供非暂时性计算机可读媒体。所述媒体包括当经执行时使得设备存储视频信息的代码。所述媒体进一步包括当经执行时使得设备将变换分解成多个变换级的代码。所述媒体进一步包括当经执行时使得设备使用多个级变换视频信息以确定每一变换级处的变换级输出的代码。所述媒体进一步包括当经执行时使得设备将每一变换级处的变换级输出限定到预定位深度的代码。所述媒体进一步包括当经执行时使得设备对多个级中的最末级的经限定变换输出执行操作的代码,其中所述操作仅可供与具有预定位深度的数据一起使用。

[0011] 本发明中描述的标的物的另一方面提供一种用于编码视频的设备。所述设备包括用于存储视频信息的装置。所述设备进一步包括用于将变换分解成多个变换级的装置。所述设备进一步包括用于使用多个级变换视频信息以确定每一变换级处的变换级输出的装置。所述设备进一步包括用于将每一变换级处的变换级输出限定到预定位深度的装置。所述设备进一步包括用于对多个级中的最末级的经限定变换输出执行操作的装置,其中所述操作仅可供与具有预定位深度的数据一起使用。

附图说明

[0012] 图1是说明可利用根据本发明中所描述的方面的技术的实例视频编码和解码系统的框图。

[0013] 图2是说明可实施根据本发明中所描述的方面的技术的视频编码器的实例的框

图。

[0014] 图3是说明可实施本发明中所描述的方面的技术的视频解码器的实例的框图。

[0015] 图4说明(例如)使用基于网格的计算的低复杂度正变换的方法的流程图。

[0016] 图5说明用于低复杂度正变换的方法的流程图。

[0017] 图式中所说明的各种特征可能未按比例绘制。因此,为了清楚起见,可能任意扩大或减小各种特征的尺寸。另外,图式中的一些图式可能并未描绘给定系统、方法或装置的所有组件。最后,在整个说明书和图式中,可使用相同参考标号来表示相同特征。

具体实施方式

[0018] 本发明中描述的技术通常涉及在视频编码期间的正变换,尤其高效率视频译码(HEVC)标准及其扩展。

[0019] 视频译码标准包括ITU-T H.261、ISO/IEC MPEG-1Visual、ITU-T H.262或ISO/IEC MPEG-2Visual、ITU-T H.263、ISO/IEC MPEG-4Visual和ITU-T H.264(也被称为ISO/IEC MPEG-4AVC),包括其可缩放视频译码(SVC)和多视图视频译码(MVC)扩展。此外,存在一种新的视频译码标准,高效率视频译码(HEVC),其由ITU-T视频译码专家组(VCEG)及ISO/IEC动画专家组(MPEG)的视频译码联合合作小组(JCT-VC)进行开发。

[0020] 如上文所提及,HEVC标准利用大小达 32×32 的正变换(例如,在HEVC编码器中)及反变换,而AVC标准仅利用达 8×8 的变换大小。更大的变换大小增加HEVC中的大译码块的译码效率;但是,相较于使用更小的变换大小,其也增大复杂度、计算周期及处理时间。本发明中所描述的方法可减小编码器将视频信息从像素域变换到系数域时所需要的增加的复杂度及周期。举例来说,某些方法包含将视频编码器中的较大正变换向量分解成多个级(例如,基于网格方法)并限定每一级的内部位深度。

[0021] 在一些实施中,出于从残余域转换到系数域的目的,将较大正变换分解成多个级(例如,使用实施正变换的基于网格的或“蝶形”方法)可导致处理比使用矩阵乘法方法更有效。基于网格的 $N \times N$ 变换实施的一个实例在下文附录A的代码中加以说明。在一个实施中,变换单元(例如, 16×16 变换)可以两百五十六个残余源像素(例如,附录A中的pSrc)开始,所述残余源像素中的每一者可表示明度值。处理器或编码器(例如,图2的编码器20的变换处理单元52)可接着确定两百五十六个输出系数(例如,附录A中的pDst),一个系数用于每一源像素。在第一级之后,处理器或编码器(例如,变换处理单元52)可接着确定两像素的一百二十八个和及两像素的一百二十八个差(例如,附录A的nE和nO的)。在第二处理级期间,处理器或编码器可使用nE与nO和及差来确定对和及差和。在四个级之后,处理器或编码器可产生输出系数(例如,pDst)。使用此基于网格的方法(如附录A中所展示),全大小变换可分解成若干更小、更复杂的变换,如果这些变换一起相乘又将产生全大小变换。

[0022] 在执行网格方法(例如,上文所论述的网格方法)或任何其它分解方法之后,本发明中所描述的方法可配置处理器或编码器以在每一级处将内部位深度限定到某一水平,从而使得处理器或编码器能够利用所述水平的其它计算上有效的指令集。实际上,某些编码器及解码器包含仅可供与具有某些位深度的输入一起使用的计算上有效的指令集。举例来说,虽然输入到变换的位深度可为9位并且起始内部位深度可大于16位,但视频编码器可经配置以将变换的每一分解级的内部位深度(例如,在变换操作期间的位深度而不是仅变换

输出位深度)限定到16位。由于每一级的内部位深度经限定到16位,处理器可利用计算上有有效的指令集,所述计算上有有效的指令集经设计特别供与16位操作(例如,ARM结构、高级SIMD (NEON)、数字信号处理(DSP)等)一起使用。在一个实施中,视频编码器可使用饱和逻辑限定每一级的内部位深度(例如,将内部值削减到16位)。

[0023] 在一些情况下,限定变换级的位深度可导致最终视频质量降低。为防止此结果,本发明中所描述的方法可测量由限定过程引起的偏差(例如,误差)水平。所述方法可接着将偏差水平与预定阈值(例如,持久极限)相比较且接着如果偏差水平超出预定阈值重新计算变换系数的子集。通过以此方式限定位深度并且重新计算系数的子集,本发明中所描述的方法可允许视频编码器的变换处理单元使用更少的计算资源同时还保留视频质量。

[0024] 在使用基于块的处理(例如,HEVC,其中视频帧可分割成视频块或译码单元)的视频编解码器中,可从原始像素减去预测块或预测单元(例如,从帧间预测或帧内预测)。如上文进一步解释,残余数据可接着使用正变换(例如,离散余弦变换)变换为残余变换系数,经量化,经熵编码(例如,以实现进一步压缩)。可使用下文进一步描述的各种熵译码引擎(例如,CAVLC、CABAC等)来执行熵编码。然后,且如下文还进一步描述,解码器可接着对系数进行熵解码、解量化及反变换。最终,系数可加回到预测块从而形成经重建像素。

[0025] 在视频译码的一个实施例中,可首先使用经重建的在时间上和/或空间上相邻的块中的像素来预测图像块。可接着对预测误差(有时称为“残差”)进行变换及量化。举例来说,如果S为大小 $N \times N$ 的残差块,那么可使用如下的矩阵乘法导出经变换块K:

$$[0026] \quad K = A * S * B$$

[0027] 其中K、A和B也具有大小 $N \times N$ 。A为垂直变换矩阵,且B为水平变换矩阵。在一些实施例中,A和B为彼此的转置(例如, $B = A'$ 其中“'”意味着转置)。在其它实施例中,A及B不为彼此的转置。当A与B为彼此的转置时,先前等式变为:

$$[0028] \quad K = A * S * A'$$

[0029] 每一变换(A及B)可包含多种变换中的任一者。在一些实施例中,变换包含离散余弦变换(DCT)、离散正弦变换(DST)、哈达玛(Hadamard)变换、哈尔(Haar)变换等中的一者。

[0030] 在SVC扩展中,可存在多个视频信息层。底层可充当基础层(BL),且顶层可充当增强型层(EL)或“增强层”。顶层与底层之间的所有层可充当EL或BL中的一者或两者。SVC可用于提供质量可扩展性(或信噪比,SNR)、空间可扩展性和/或时间可扩展性。增强型层可具有与基础层不同的空间分辨率。可使用针对SVC提供的不同层来执行当前块的预测。此预测可称作层间预测。在SVC中可利用层间预测方法以便减少层间冗余。层间预测的一些实例可包含层间帧内预测、层间运动预测和层间残余预测。层间帧内预测使用基础层中的协同定位块的重建来预测增强层中的当前块。层间运动预测使用基础层的运动来预测增强层中的运动。层间残余预测使用基础层的残差来预测增强层的残差。被称作“帧内BL模式”的用于增强层的一个特定译码模式包含可使用基础层中的对应(有时被称作“协同定位”,例如,定位于相同空间位置处)块的织构来预测织构。

[0031] 在层间残余预测中,基础层的残差可用于预测增强层中的当前块。可将所述残差定义为对视频单元与源视频单元之间的时间预测的差。在残余预测中,还在预测当前块中考虑基础层的残差。举例来说,可使用从增强层的残差、从增强层的时间预测及从基础层的残差来重建当前块。可根据以下等式重建当前块:

[0032] $\hat{I}_e = re + Pe + rb$

[0033] 其中 \hat{I}_e 表示当前块的重建, re 表示来自增强层的残差, Pe 表示来自增强层的时间预测, 且 rb 表示来自基础层的残差预测。

[0034] 对于使用差域的帧间译码, 基于增强层参考图片中的对应经预测块样本与经缩放基础层参考图片中的对应经预测块样本之间的差值来确定当前经预测块。所述差值可被称为差预测块。将协同定位的基础层经重建样本添加到差预测块以便获得增强层预测样本。

[0035] 本发明中描述的技术可解决与HEVC中的正变换的矩阵乘法期间的复杂计算要求相关的问题。技术可改善编码器和/或变换处理单元可执行正变换矩阵乘法的速度、效率和功效。

[0036] 下文参看附图更充分地描述新颖系统、设备和方法的各种方面。然而, 本发明可以许多不同形式来体现, 且不应将其解释为限于在整个本发明中所呈现的任何特定结构或功能。实际上, 提供这些方面以使得本发明将为透彻且完整的, 且将向所属领域的技术人员充分传达本发明的范围。基于本文中的教导, 所属领域的技术人员应了解, 本发明的范围既定涵盖无论是独立于本发明的任何其它方面而实施还是与之组合而实施的本文中所揭示的新颖系统、设备及方法的任何方面。举例来说, 可使用本文中所阐述的任何数目个方面来实施设备或实践方法。此外, 本发明的范围旨在涵盖用作本文中所阐述的本发明的各种方面的补充或替代的其它结构、功能性或结构与功能性来实践的此设备或方法。应理解, 可通过权利要求书的一或多个要素来体现本文中所揭示的任何方面。

[0037] 尽管本文中描述了特定方面, 但这些方面的许多变化和排列属于本发明的范围内。虽然提及了优选方面的一些好处及优点, 但本发明的范围不欲限于特定好处、用途或目标。相反地, 本发明的方面希望广泛地适用于不同无线技术、系统配置、网络及发射协议, 其中的一些是作为实例而在图中及在优选方面的以下描述中加以说明。具体实施方式和图式仅说明本发明, 而不是限制由所附权利要求书和其等效物定义的本发明的范围。

[0038] 图1是说明可利用根据本发明中所描述的方面的技术的实例视频编码和解码系统的框图。如图1中所展示, 视频编码和解码系统10包含源装置12, 所述源装置12提供稍后由目的地装置14解码的经编码视频数据。确切地说, 源装置12经由计算机可读媒体16将视频数据提供到目的地装置14。源装置12和目的地装置14可包括广泛范围的装置中的任一者, 包含桌上型计算机、笔记型(例如, 膝上型)计算机、平板计算机、机顶盒、电话手持机(例如, 智能电话)、电视机、摄像机、显示装置、数字媒体播放器、视频游戏控制台、视频流装置等。在一些情况下, 源装置12和目的地装置14可经装备以用于无线通信。

[0039] 如上文所提及, 目的地装置14可经由计算机可读媒体16接收待解码的经编码视频数据。计算机可读媒体16可包括能够将经编码视频数据从源装置12移动到目的地装置14的任何类型的媒体或装置。在一个实例中, 计算机可读媒体16可包括通信媒体(未绘制)以使得源装置12能够及时将经编码视频数据直接发射到目的地装置14。经编码视频数据可根据通信标准(例如, 无线通信协议)来调制, 且被发射到目的地装置14。通信媒体可包括任何无线或有线通信媒体, 例如射频(RF)频谱或一或多个物理发射线。通信媒体还可形成分组网络(例如, 局域网、广域网或例如因特网的全球网络)的部分。通信媒体可包括路由器、交换机、基站或可用于促进从源装置12到目的地装置14的通信的任何其它设备。

[0040] 在一些实例中, 经编码数据可从输出接口22输出到存储装置(未绘制)。类似地, 可

通过输入接口28从存储装置存取经编码数据。存储装置可包含多种分布式或本地存取的数据存储媒体中的任一者,例如硬盘驱动器、蓝光光盘、DVD、CD-ROM、快闪存储器、易失性或非易失性存储器或用于存储经编码视频数据的任何其它合适的数字存储媒体。在另一实例中,存储装置可对应于文件服务器或可存储由源装置12产生的经编码视频的另一中间存储装置。目的地装置14可经由流式发射或下载从存储装置存取经存储的视频数据。文件服务器可为能够存储经编码视频数据且将所述经编码视频数据发射到目的地装置14的任何类型的服务器。实例文件服务器包含网络服务器(例如,用于网站)、FTP服务器、网络附接存储(NAS)装置或本地磁盘驱动器。目的地装置14可通过任何标准数据连接(包含因特网连接)来存取经编码视频数据。数据连接可包含适合于存取存储于文件服务器上的经编码视频数据的无线信道(例如,Wi-Fi连接)、有线连接(例如,DSL、电缆调制解调器等)或两者的组合。来自存储装置的经编码视频数据的发射可以是流式发射、下载发射或其组合。

[0041] 本发明的技术未必限于无线应用或环境。所述技术可以应用于支持多种多媒体应用中的任一者的视频译码,例如空中电视广播、有线电视发射、卫星电视发射、因特网流式视频发射(例如,经由HTTP的动态自适应流式发射(DASH))、经编码到数据存储媒体上的数字视频,存储在数据存储媒体上的数字视频的解码,或其它应用。在一些实例中,所述系统10可经配置以支持用于例如视频流式发射、视频回放、视频广播、视频电话等应用的单向或双向视频发射。

[0042] 在图1的实例中,源装置12包含视频源18、视频编码器20和输出接口22。目的地装置14包含输入接口28、视频解码器30和显示装置32。根据本发明,源装置12的视频编码器20可经配置以应用用于对包含符合多个标准或标准扩展的视频数据的位流进行译码的技术。在其它实例中,源装置12和目的地装置14可包含其它组件或布置。举例来说,源装置12可从外部视频源(例如,外部摄像机)接收视频数据。同样,目的地装置14可与外部显示装置介接,而不是集成显示装置32。

[0043] 尽管本发明的技术通常通过视频编码装置来执行,但是所述技术还可通过视频编码器/解码器(通常被称作“编解码器”)执行。此外,本发明的技术还可由视频预处理器来执行。源装置12及目的地装置14仅为此类译码装置的实例,其中源装置12产生用于发射到目的地装置14的经译码视频数据。在一些实例中,源装置12及目的地装置14可以大致上对称的方式操作使得其各自包含视频编码和解码组件。因此,系统10可支持源装置12与目的地装置14之间的单向或双向视频发射,例如,针对视频流式发射、视频回放、视频广播、视频电话等。

[0044] 源装置12的视频源18可包含视频捕获装置(未绘制),例如,视频摄像机、含有先前所捕获的视频的视频存档、从视频内容提供者接收视频的视频馈入接口等。作为另一替代方案,视频源18可产生基于计算机图形的数据,或实况视频、所存档视频和计算机产生的视频的组合。在一些情况下,如果视频源18是摄像机,那么源装置12和目的地装置14可为摄像机电话或视频电话。在另一实施例中,本发明中描述的技术一般可适用于视频译码,且可应用于无线和/或有线应用。在每一种情况下,可由视频编码器20编码所捕获、经预先捕获或计算机产生的视频。经编码视频信息可接着通过输出接口22输出到计算机可读媒体16上。

[0045] 计算机可读媒体16可包含瞬时媒体,例如无线广播或有线网络发射,或存储媒体(即,非暂时性存储媒体),所述存储媒体例如硬盘、快闪驱动器、压缩光盘、数字视频光盘、

蓝光光盘或其它计算机可读媒体。在一些实例中,网络服务器(未展示)可从源装置12接收经编码视频数据及例如经由网络发射、直接有线通信等将经编码视频数据提供到目的地装置14。类似地,媒体生产设施(例如光盘冲压设施)的计算器件可以从源器件12接收经编码视频数据并且生产含有经编码视频数据的光盘。因此,计算机可读媒体16可包含各种形式的一或多个计算机可读媒体。

[0046] 目的地装置14的输入接口28可从计算机可读媒体16接收信息。计算机可读媒体16的信息可包含由视频编码器20定义的语法信息。语法信息还可由视频解码器30使用,所述语法信息可包含描述块及其它经译码单元的特性和/或处理的语法元素。显示装置32可将经解码视频数据显示给用户,且可包括多种显示装置中的任一者,例如阴极射线管(CRT)、液晶显示器(LCD)、等离子显示器、有机发光二极管(OLED)显示器或另一类型的显示装置。

[0047] 视频编码器20和视频解码器30可根据视频译码标准来操作,所述视频译码标准例如高效率视频译码(HEVC)标准或其变体中的任一者(例如,HEVC测试模型(HM))。替代地,视频编码器20及视频解码器30可根据其它所有权或行业标准来操作,例如,ITU-T H.264标准(MPEG-4)第10部分高级视频译码(AVC)、ITU-T H.263、ITU-T H.262(ISO/IEC MPEG-2 Visual)、ISO/IEC MPEG-1 Visual、ITU-T H.261或任何此些标准的扩展。在一些方面中,视频编码器20和视频解码器30可与音频编码器、音频解码器、多路复用器-多路分用器单元(未绘制)或其它硬件和软件集成,以处置共同数据流或单独数据流中的音频和视频两者的编码。

[0048] 视频编码器20和视频解码器30各自可实施为多种合适的编码器电路中的任一者,例如一或多个微处理器、数字信号处理器(DSP)、专用集成电路(ASIC)、现场可编程门阵列(FPGA)、离散逻辑、软件、硬件、固件或其任何组合。如果必须要执行本发明的技术,视频编码器20和/或视频解码器30可存储用于合适的非暂时性计算机可读媒体中的软件的指令且使用一或多个处理器执行硬件中的指令。视频编码器20和视频解码器30中的每一者可包含在一或多个编码器或解码器中,所述编码器或解码器中的任一者可集成为相应装置中的组合式编码器/解码器(编解码器)的部分。包含视频编码器20和/或视频解码器30的装置可包括集成电路、微处理器和/或无线通信装置,例如,手机。

[0049] HEVC标准指定视频帧或图片可划分成树块序列或包含明度样本及色度样本两者的最大译码单元(LCU)。位流内的语法数据可定义LCU(就像素数目来说,其为最大译码单元)的大小。切片包含按译码次序的许多连续树块。视频帧或图片可被分割成一或多个切片。每一树块可根据四分树而分裂成多个译码单元(CU)。一般来说,四叉树数据结构包含每个CU一个节点,其中根节点对应于所述树块。如果CU分割成四个子CU,那么对应于CU的节点包含四个叶节点,所述叶节点中的每一者对应于所述子CU中的一者。

[0050] 四叉树数据结构的每一节点可提供对应CU的语法数据。举例来说,四叉树中的节点可包括分裂旗标,指示对应于所述节点的CU是否分裂成子CU。用于CU的语法元素可以递归式定义,且可取决于CU是否分裂成子CU。如果CU不进一步分裂,那么将其称作叶CU。在本发明中,叶CU的四个子CU也将被称作叶CU,即使不存在原始叶CU的明确分裂时也是如此。举例来说,如果 16×16 大小的CU不进一步分裂,那么四个 8×8 子CU也将被称作叶CU,虽然 16×16 CU从未分裂。

[0051] CU具有类似于H.264标准的宏块的目的,但是CU并不具有大小区别。举例来说,树

块可以分裂成四个子节点(还被称作子CU),并且每一子节点又可以是父节点并且可以分裂成另外四个子节点。最后的未经分裂子节点(被称作四叉树的叶节点)包括译码节点,也称为叶CU。与经译码位流相关联的语法数据可定义树块可以分裂的最大次数(被称作最大CU深度),并且还可定义译码节点的最小大小。因此,位流还可定义最小译码单元(SCU)。本发明使用术语“块”来指HEVC的上下文中的CU、PU或TU中的任一者,或者其它标准的上下文中的类似数据结构(例如,其在H.264/AVC中的宏块及子块)。

[0052] CU包含译码节点及与所述译码节点相关联的预测单元(PU)及变换单元(TU)。CU的大小对应于译码节点的大小且形状必须是正方形。CU的大小可在 8×8 像素多到具有最大值 64×64 像素或在一些情况下更大树块大小的范围内。每一CU可含有一或多个PU和一或多个TU。举例来说,与CU相关联的语法数据可描述将CU分割成一或多个PU。分割模式可在跳过CU或经直接模式编码、经帧内预测模式编码或经帧间预测模式编码之间有区别。PU可分割为非正方形形状。举例来说,与CU相关联的语法数据还可描述CU根据四叉树分割成一或多个TU。TU可为正方形或非正方形(例如,矩形)形状。

[0053] HEVC标准允许根据TU变换,TU可因不同CU而有所不同。TU通常是基于针对经分割LCU定义的给定CU内的PU的大小而定大小,但是情况可能并不总是如此。TU通常与PU大小相同或小于PU。在一些实例中,对应于CU的残余样本可使用被称为“残余四叉树”(RQT)的四叉树结构而细分成较小单元。RQT的叶节点可被称作变换单元(TU)。可变换与TU相关联的像素差值以产生变换系数,所述变换系数可经量化。

[0054] 叶CU可包含一或多个预测单元(PU)。一般来说,PU表示对应于对应CU的全部或一部分的空间区域,且可包含用于检索PU的参考样本的数据。此外,PU包含与预测有关的数据。举例来说,当PU经帧内模式编码时,用于PU的数据可包含在残余四叉树(RQT)中,残余四叉树可包含描述用于对应于PU的TU的帧内预测模式的数据。作为另一实例,当PU经帧间模式编码时,PU可包含定义用于PU的一或多个运动向量的数据。定义PU的运动向量的数据可描述(例如)运动向量的水平分量、运动向量的垂直分量、运动向量的分辨率(例如,四分之一像素精度或八分之一像素精度)、运动向量所指向的参考图片,及/或运动向量的参考图片列表(例如,列表0、列表1或列表C)。

[0055] 具有一或多个PU的叶CU还可包含一或多个变换单元(TU)。变换单元可使用RQT(也称为TU四叉树结构)来指定,如上文所论述。举例来说,分裂旗标可指示叶CU是否分裂成四个变换单元。接着,每一变换单元可以进一步分裂成其它子TU。当TU不进一步分裂时,其可被称为叶TU。一般来说,对于帧内译码,所有属于叶CU的叶TU共享相同的帧内预测模式。即,通常应用相同的帧内预测模式来计算叶CU的所有TU的预测值。对于帧内译码,视频编码器可使用帧内预测模式将每一叶TU的残余值计算为CU的对应于TU的部分与原始块之间的差。TU不一定限于PU的大小。因此,TU可比PU大或小。对于帧内译码,PU可以与相同CU的对应叶TU并置。在一些实例中,叶TU的最大大小可对应于对应叶CU的大小。

[0056] 此外,叶CU的TU还可与相应四叉树数据结构(被称作残余四叉树(RQT))相关联。即,叶CU可包含指示叶CU如何分割成TU的四叉树。TU四叉树的根节点通常对应于叶CU,而CU四叉树的根节点通常对应于树块(或LCU)。RQT的未分裂的TU被称作叶TU。一般来说,除非另有说明,本发明分别使用术语CU和TU指代叶CU和叶TU。

[0057] 视频序列通常包含一系列视频帧或图片。图片群组(GOP)通常包括一系列的

图片中的一或多者。GOP可包含GOP的标头、图片中的一或多者的标头或其它地方中的语法数据,所述语法数据描述GOP中包含的图片的数目。图片的每一切片可包含描述用于相应切片的编码模式的切片语法数据。图1的视频编码器20可对个别视频切片内的视频块操作以便编码视频数据。视频块可对应于CU内的译码节点。视频块可具有固定或变化的大小,且可根据指定译码标准而大小不同。

[0058] HEVC支持各种PU大小的预测。假设特定CU的大小为 $2N \times 2N$,那么HEVC支持 $2N \times 2N$ 或 $N \times N$ 的PU大小的帧内预测,及 $2N \times 2N$ 、 $2N \times N$ 、 $N \times 2N$ 或 $N \times N$ 的对称PU大小的帧间预测。HEVC还支持用于 $2N \times nU$ 、 $2N \times nD$ 、 $nL \times 2N$ 及 $nR \times 2N$ 的PU大小的帧间预测的不对称分割。在不对称分割中,不分割CU的一个方向,而将另一方向分割成25%和75%。CU的对应于25%分区的一部分由“n”跟随有“上方”、“下方”、“左侧”或“右侧”的指示来指示。因此,例如,“ $2N \times nU$ ”指经水平分割的 $2N \times 2N$ CU,其中顶部为 $2N \times 0.5N$ PU,而底部为 $2N \times 1.5N$ PU。

[0059] 在本发明中,“ $N \times N$ ”与“N乘N”可互换地使用以指就垂直维度和水平维度来说的视频块的像素尺寸,例如,16×16像素或16乘16像素。一般来说,16×16块将在垂直方向上具有16像素($y=16$),且在水平方向上具有16像素($x=16$)。同样, $N \times N$ 块一般来说在垂直方向上可具有N像素,且在水平方向上可具有N像素,其中N表示非负整数值。块中的像素可布置成行和列。此外,块可能不一定在水平方向与垂直方向上具有相同数目的像素。举例来说,块可包括 $N \times M$ 像素,其中M未必等于N。

[0060] 在使用CU的PU的帧内预测性或帧间预测性译码之后,视频编码器20可计算CU的TU的残余数据。PU可包括描述在空间域(也被称作像素域)中产生预测性像素数据的方法或模式的语法数据,并且TU可包括在对残余视频数据应用了变换(例如,离散余弦变换(DCT)、整数变换、小波变换或概念上类似的变换)的变换域中的系数。残余数据可对应于未编码图片的像素与对应于PU的预测值之间的像素差。视频编码器20可形成包含用于CU的残余数据的TU,且接着变换所述TU以产生用于CU的变换系数。

[0061] 在任何用于产生变换系数的变换之后,视频编码器20可执行变换系数的量化。量化为既定具有其最广泛一般含义的广义术语。在一个实施例中,量化是指变换系数经量化以可能减少用于表示系数的数据量从而提供进一步压缩的过程。量化过程可减小与系数中的一些或全部相关联的位深度。例如, n 位值可在量化期间被下舍入到 m 位值,其中 n 大于 m 。

[0062] 在量化之后,视频编码器可以扫描变换系数,从包括经量化变换系数的二维矩阵产生一维向量。扫描可经设计以将较高能量(并且因此较低频率)的系数放置在阵列正面,并且将较低能量(并且因此较高频率)的系数放置在阵列的背面。在一些实例中,视频编码器20可利用预定义扫描次序来扫描经量化的变换系数以产生可经熵编码的串行化向量。在其它实例中,视频编码器20可执行自适应扫描。在扫描经量化变换系数以形成一维向量之后,视频编码器20可(例如)根据上下文自适应可变长度译码(CAVLC)、上下文自适应二进制算术译码(CABAC)、基于语法的上下文自适应二进制算术译码(SBAC)、概率区间分割熵(PIPE)译码或另一熵编码方法对一维向量进行熵编码。视频编码器20还可以对与经编码视频数据相关联的语法元素进行熵编码以供视频解码器30在解码视频数据时使用。

[0063] 视频编码器20可进一步(例如)在帧标头、块标头、切片标头或GOP标头中将例如基于块的语法数据、基于帧的语法数据及基于GOP的语法数据等语法数据发送到视频解码器30。GOP语法数据可描述相应GOP中的多个帧,及帧语法数据可指示用以编码对应帧的编码/

预测模式。

[0064] 图2是说明可实施根据本发明中所描述的方面的技术的视频编码器的实例的框图。视频编码器20的单元中的一或多个者可经配置以执行本发明的技术中的任一者或全部。作为一个实例,变换处理单元52可经配置以执行本发明中描述的变换技术中的任一者或全部。然而,本发明的方面不限于此。在一些实例中,本发明中所描述的技术可在视频编码器20的各种组件间共享。在一些实例中,处理器(未展示)可经配置以执行本发明中描述的技术中的任一者或全部。

[0065] 视频编码器20可对视频切片内的视频块执行帧内译码及帧间译码。帧内译码依靠空间预测来减少或移除给定视频帧或图片内的视频中的空间冗余。帧间译码依靠时间预测来减少或移除视频序列的邻近帧或图片内的视频中的时间冗余。帧内模式(I模式)可指代若干基于空间的译码模式中的任一者。例如单向预测(P模式)或双向预测(B模式)的帧间模式可以指若干基于时间的译码模式中的任一者。

[0066] 视频编码器20可接收待编码视频帧内的当前视频块。在图2的实例中,视频编码器20包含模式选择单元40、参考帧存储器64、求和器50、变换处理单元52、量化单元54及熵编码单元56。模式选择单元40包含运动估计单元42、运动补偿单元44、帧内预测单元46和分割单元48。为进行视频块重建,视频编码器20还可包含反量化单元58、反变换单元60和求和器62。还可包含解块过滤器(未绘制)以将块边界过滤,以从经重建视频移除成块性假影。必要时,所述解块过滤器通常将对求和器62的输出进行过滤。除了解块过滤器之外,还可使用额外过滤器(环路内或环路后)。为简洁起见未展示此些过滤器,但是必要时,此些过滤器可对求和器50的输出进行过滤(作为环路内过滤器)。

[0067] 在编码过程期间,视频编码器20可接收待译码的视频帧或切片。所述帧或切片可被划分成多个视频块。运动估计单元42和运动补偿单元44可相对于一或多个参考帧中的一或多个块执行所接收视频块的帧间预测性译码以提供时间预测。帧内预测单元46可替代地相对于与待译码块相同的帧或切片中的一或多个相邻块执行所接收视频块的帧内预测性译码以提供空间预测。视频编码器20可执行多个译码遍次,(例如)以针对每一视频数据块选择一种适当的译码模式。

[0068] 此外,分割单元48可基于前述译码遍次中的前述分割方案的评估将视频数据块分割成子块。举例来说,分割单元48最初可将帧或切片分割成LCU,并且基于速率失真分析(例如,速率失真优化)将LCU中的每一者分割成子CU。模式选择单元40可进一步产生指示LCU分割成子CU的四叉树数据结构。四叉树的叶节点CU可包含一或多个PU和一或多个TU。

[0069] 模式选择单元40可(例如)基于误差结果选择译码模式(帧内或帧间)中的一者,且将所得经帧内或帧间译码块提供到求和器50以产生残余块数据,且提供到求和器62以重建用作参考帧的经编码块。模式选择单元40还可将语法元素(例如运动向量、帧内模式指示符、分割信息和其它此类语法信息)提供到熵译码单元56。

[0070] 运动估计单元42和运动补偿单元44可高度集成,但出于概念目的单独地加以说明。由运动估计单元42执行的运动估计是产生运动向量的过程,所述过程估计视频块的运动。举例来说,运动向量可指示当前视频帧或图片内的视频块的PU相对于参考帧(或其它经译码单元)内的预测块相对于当前帧(或其它经译码单元)内正经译码的当前块的位移。预测块是经发现在像素差方面与待译码块紧密匹配的块,其可通过绝对差总和(SAD)、平方差

总和 (SSD) 或其它差量度来确定。在一些实例中, 视频编码器20可计算存储在参考帧存储器64中的参考图片的子整数像素位置的值。举例来说, 视频编码器20可内插参考图片的四分之一像素位置、八分之一像素位置或其它分数像素位置的值。因此, 运动估计单元42可相对于整数像素位置和分数像素位置执行运动搜索并且输出具有分数像素精确度的运动向量。

[0071] 运动估计单元42通过比较PU的位置与参考图片的预测块的位置来计算用于经帧间译码切片中的视频块的PU的运动向量。参考图片可选自第一参考图片列表(列表0)或第二参考图片列表(列表1), 所述参考图片列表中的每一者识别存储在参考帧存储器64中的一或多个参考图片。运动估计单元42将计算出来的运动向量发送到熵编码单元56和运动补偿单元44。

[0072] 运动补偿单元44执行的运动补偿可包括基于由运动估计单元42确定的运动向量来取出或产生预测性块。又, 在一些实例中, 运动估计单元42和运动补偿单元44可在功能上集成。在接收到当前视频块的PU的运动向量后, 运动补偿单元44可在参考图片列表中的一者中定位所述运动向量指向的预测性块。求和器50可通过从经译码的当前视频块的像素值减去预测块的像素值从而形成像素差值来形成残余视频块, 如下文所论述。运动估计单元42可相对于明度分量执行运动估计, 且运动补偿单元44可针对色度分量和明度分量两者使用基于亮度分量计算的运动向量。模式选择单元40还可产生与视频块和视频切片相关联的语法元素供视频解码器30在解码视频切片的视频块时使用。语法元素可表示处于视频序列层级、视频帧层级、视频切片层级、视频CU层级或视频PU层级中的一或多者的预测信息。举例来说, 运动补偿单元44可产生指示包含CU、PU及TU的大小的视频块信息及用于帧内模式预测的运动向量信息的语法元素。

[0073] 如上文所描述, 作为由运动估计单元42和运动补偿单元44执行的帧间预测的替代方案, 帧内预测单元46可对当前块进行帧内预测或计算。明确地说, 帧内预测单元46可确定用来编码当前块的帧内预测模式。在一些实例中, 帧内预测单元46可(例如) 在单独的编码编次期间使用各种帧内预测模式对当前块进行编码, 并且帧内预测单元46(或在一些实例中为模式选择单元40) 可从测试模式中选择适当的帧内预测模式来使用。

[0074] 举例来说, 帧内预测单元46可使用速率失真分析计算各种受测帧内预测模式的速率失真值, 并且从所述受测模式当中选择具有最佳速率失真特性的帧内预测模式。速率失真分析通常确定经编码块与经编码以产生所述经编码块的原始未经编码块之间的失真(或误差) 的量, 以及用于产生经编码块的位速率(即, 位数目)。帧内预测单元46可针对各种经编码块从失真及速率计算比率, 以确定哪个帧内预测模式展现用于所述块的最佳速率失真值。

[0075] 在选择用于块的帧内预测模式之后, 帧内预测单元46可将指示用于所述块的选定帧内预测模式的信息提供到熵编码单元56。熵编码单元56可对指示选定帧内预测模式的信息进行编码。视频编码器20可在经发射位流中包含配置数据, 所述配置数据可包含多个帧内预测模式索引表及多个经修改的帧内预测模式索引表(还称为码字映射表), 对用于各种块的编码上下文的定义, 及对将用于所述上下文中的每一者中的最可能帧内预测模式、帧内预测模式索引表及经修改的帧内预测模式索引表的指示。

[0076] 视频编码器20通过从经译码的原始视频块减去来自模式选择单元40的预测数据而形成残余视频块。求和器50可执行此减法运算。变换处理单元52可将变换(例如离散余弦

变换(DCT)或概念上类似的变换)应用于残余块,从而产生包含残余变换系数值的视频块。变换处理单元52可执行概念上类似于DCT的其它变换。也可使用小波变换、整数变换、子带变换或其它类型的变换。变换处理单元52可随后将变换应用于残余块,从而产生残余变换系数的块。变换处理单元52可将残余信息从像素值域转换到变换域,例如频域。更确切地说,在应用变换之前,TU可包括像素域中的残余视频数据,并且在应用变换之后,TU可包括表示频域中的残余视频数据的变换系数。

[0077] 常规地,视频编码器20保持用于由经实施视频压缩标准支持的不同大小的TU中的每一者的单独上下文模型。对于HEVC标准,额外变换单元大小(例如,32×32至128×128)可用于改善视频译码效率,但是额外TU大小也导致增大的存储器及用以针对额外变换单元大小中的每一者保持上下文模型的计算要求。在一些情况下,较大TU大小可使用更多上下文,所述上下文可导致增大的存储器及保持用于较大TU大小的上下文的增加的数目的计算要求。为了减小此问题的影响,变换处理单元52可经进一步配置以执行上文和下文关于在矩阵乘法期间简化变换(有时被称作“正变换”)并限定其位深度(例如,下文及关于图4至5所描述的内部位深度限定方法)所描述的方法中的任一者。

[0078] 变换处理单元52可将所得变换系数发送到量化单元54。量化单元54可随后量化变换系数以进一步减小位速率。量化过程可减小与系数中的一些或全部相关联的位深度。可通过调整量化参数来修改量化程度。在一些实例中,量化单元54可随后执行对包含经量化变换系数的矩阵的扫描。替代地,熵编码单元56可执行所述扫描。

[0079] 作为一实例,变换处理单元52可将变换结果限定为预定位深度值(例如,16位位深度或其它位深度值)。在一个实施中,变换处理单元52可限定一或多个内部变换级的变换结果。视频编码器20可随后利用针对预定位深度值优化的专用指令集。以此方式,变换处理单元52可体验更快的处理速度。此过程在图4中经进一步描述及表明。

[0080] 在上文所描述的实例中,变换处理单元52经配置以将变换结果限定为16位值。在其它情况下,变换处理单元52可经进一步配置以确定经限定值将产生令人满意的结果。此过程在图5中经进一步描述及表明。

[0081] 在量化之后,熵编码单元56可对经量化变换系数进行熵译码。举例来说,熵编码单元56可执行上下文自适应可变长度译码(CAVLC)、上下文自适应二进制算术译码(CABAC)、基于语法的上下文自适应二进制算术译码(SBAC)、概率区间分割熵(PIPE)译码或另一熵编码技术。在基于上下文的熵编码的情况下,上下文可基于相邻块。在由熵编码模块56进行的熵编码之后,可将经编码位流发射到另一装置(例如,视频解码器30),或将经编码位流存档以供稍后发射或检索。

[0082] 反量化单元58和反变换单元60可分别应用反量化和反变换以在像素域中重建残余块,例如以供稍后用作参考块。运动补偿单元44可通过将残余块添加到参考帧存储器64的帧中的一者的预测性块来计算参考块。运动补偿单元44还可将一或多个内插过滤器应用于经重建残余块以计算子整数像素值用于运动估计。求和器62可将经重建残余块添加到由运动补偿单元44产生的运动补偿预测块以产生经重建视频块用于存储在参考帧存储器64中。经重建视频块可随后由运动估计单元42和运动补偿单元44用作参考块以对后续视频帧中的块进行帧间译码。

[0083] 图3是说明可实施本发明中所描述的方面的技术的视频解码器的实例的框图。本

发明中描述的技术可利用视频解码器30的各种组件。在一些实例中,处理器(未展示)可经配置以执行所述技术中的任一者或全部。

[0084] 在图3的实例中,视频解码器30包含熵解码单元70,所述熵解码单元进一步包含运动补偿单元72和帧内预测单元74的预测单元81、反量化单元76、反变换单元78、参考图片(帧)存储器82和求和器80。视频解码器30可执行通常与关于视频编码器20(例如,见图1和图2)描述的编码遍次互逆的解码遍次。运动补偿单元72可基于从熵解码单元70接收的运动向量而产生预测数据,而帧内预测单元74可基于从熵解码单元70接收的帧内预测模式指示符而产生预测数据。

[0085] 常规地,视频解码器30将针对由经实施视频压缩标准支持的不同TU大小中的每一者保持单独上下文模型。对于HEVC标准,额外变换单元大小(例如,32×32至128×128)可用于改善视频译码效率,但是额外TU大小还导致增大的存储器及用以针对额外变换单元大小中的每一者保持上下文模型的计算要求。

[0086] 在解码过程期间,视频解码器30可从视频编码器20接收表示经解码视频切片的视频块及相关联的语法元素的经编码视频位流。视频解码器30的熵解码单元70对位流进行熵解码,以产生经量化系数、运动向量或帧内预测模式指示符和其它语法元素。熵解码单元70可随后将运动向量和其它语法元素转发到运动补偿单元72。视频解码器30可接收视频切片层级及/或视频块层级的语法元素。

[0087] 当视频切片经译码为经帧内译码(I)切片时,帧内预测单元74可基于所传信帧内预测模式和来自当前帧或图片的先前经解码块的数据产生用于当前视频切片的视频块的预测数据。当视频帧经译码为经帧间译码(例如,B、P或GPB)切片时,运动补偿单元72可基于从熵解码单元70接收的运动向量和和其它语法元素产生用于当前视频切片的视频块的预测块。可从参考图片列表中的一者内的参考图片中的一者产生预测块。视频解码器30可使用基于存储于参考图片(帧)存储器82中的参考图片的默认建构技术来建构参考帧列表,列表0及列表1。运动补偿单元72可通过剖析运动向量和和其它语法元素确定用于当前视频切片的视频块的预测信息,并且可使用所述预测信息产生用于经解码当前视频块的预测性块。举例来说,运动补偿单元72可使用所接收语法元素中的一些确定用于对视频切片的视频块进行译码的预测模式(例如,帧内预测或帧间预测)、帧间预测切片类型(例如,B切片、P切片或GPB切片)、切片的参考图片列表中的一或多者的构造信息、切片的每一经帧间编码的视频块的运动向量、切片的每一经帧间译码的视频块的帧间预测状态和用以对当前视频切片中的视频块进行解码的其它信息。

[0088] 运动补偿单元72还可基于内插过滤器执行内插。运动补偿单元72可使用如由视频编码器20在视频块的编码期间使用的内插过滤器来计算参考块的子整数像素的经内插值。在此情况下,运动补偿单元72可所接收语法元素确定由视频编码器20使用的内插滤波器且使用所述内插滤波器来产生预测性块。

[0089] 反量化单元76可将在位流中提供且由熵解码单元70解码的经量化变换系数反量化,例如,解量化。反量化过程可包含使用视频解码器30针对视频切片中的每一视频块计算以确定应应用的量化程度和同样反量化程度的量化参数QPY。

[0090] 反变换单元78可对变换系数应用反变换,例如反DCT、反整数变换或概念上类似的反变换过程,以便产生像素域中的残余块。在运动补偿单元72基于运动向量和和其它语法元

素产生了当前视频块的预测性块之后,视频解码器30可通过将来自反变换单元78的残余块与由运动补偿单元72产生的对应的预测性块求和来形成经解码视频块。求和器80可执行此求和运算。还可应用解块过滤器以对经解码块进行过滤以便去除成块性假影。其它环路过滤器(在译码环路中或在译码环路之后)也可用于使像素转变平滑或者以其它方式改善视频质量。给定帧或图片中的经解码视频块可随后存储于参考图片(帧)存储器82中,所述参考图片(帧)存储器82可存储参考图片以用于后续运动补偿。参考图片(帧)存储器82还可存储经解码视频以用于稍后在显示装置上呈现,例如,图1的显示装置32。

[0091] 图4说明(例如)使用基于网格的计算的低复杂度正变换的方法400的流程图。举例来说,方法400可由处理器或编码器(例如,图2的编码器20)来执行。在一个实施例中,图2的编码器20的变换处理单元52经配置以执行方法400。虽然方法的各种框经描述为由变换处理单元执行,但应理解,所述方法的各种块可由其它处理器、编码器或其单元执行。

[0092] 方法400允许编码器将变换结果限定为16位值以使得能够使用针对16位数据(例如,ARM结构、高级SIMD(NEON)、数字信号处理(DSP)等)优化的专用指令集来执行16位操作。如上文所描述,限定位深度可改善译码效率并减小视频编码期间的计算资源要求。但是,在一些情况下,限定位深度还可降低变换输出的质量。因此,在执行16位操作之前,编码器(例如,编码器20的变换处理单元52)可经进一步配置以确定经限定值将产生令人满意的结果并因此调整,此进一步关于图5进行描述。

[0093] 方法400开始于框405处。随后在框410处,变换处理单元52将全大小正变换矩阵分解成多个不复杂的级(例如,确定多级变换矩阵)。可使用基于网格的分解方法发生分解。在一个实施中,不是使用计算资源来分解全大小变换矩阵,变换处理单元52可实际上从存储器或经译码位流检索多个级。在一些实施中,最初正变换矩阵可为 32×32 变换矩阵。在其它实施中,变换矩阵可为另一大小。

[0094] 随后在框415处,变换处理单元52可对多个级中的每一者执行适当的变换操作。在一些情况下,在已经执行了变换操作之后,所述级中的一或多者可含有大于16位的变换结果。这些变换结果可需要针对大于16位(例如,32位操作)系统而设计的内部计算操作,所述内部计算操作可减小变换处理单元52的效率。

[0095] 因此,在框420处,变换处理单元52可将多个级中的每一者的变换结果的内部位深度限定为16位值。变换处理单元52可使用饱和逻辑执行此经限定过程,例如,通过将内部值削减为16位值。更确切地说,变换处理单元52可在变换操作期间限定内部位深度而不是仅限定变换输出位深度。为了实现此限定操作,变换处理单元52可在每一中间变换级的输出时执行削减函数。在一个实施中,变换处理单元52可使用随附于附录A中说明的代码的削减函数来执行所述限定过程,其中a和b表示与经限定到16位范围内的总和相加的值。一个实例削减函数可表示为 $y = \text{CLIP}_3(x, \text{min}_{\text{val}}, \text{max}_{\text{val}})$ 。在此实例函数中,x可表示字节的数目。如果x小于 min_{val} ,那么y可设定成 min_{val} 。此外,如果x大于 max_{val} ,那么y可设定成 max_{val} 。如果x在 min_{val} 与 max_{val} 范围之间,那么y可设定成x。更特定的示例性削减函数可表示为 $(a+b)_{\text{sat}} = \text{CLIP}_3(-32768, 32767, a+b)$ 或 $(a_0*b_0+a_1*b_1+\dots)_{\text{sat}} = \text{CLIP}_3(-32768, 32767, a_0*b_0+a_1*b_1+\dots)$ 中的一者,其遵循与上述削减实例类似的逻辑树。可实施削减函数的其它结构以执行所述限定过程。在任何情况下,对于加法、减法和/或乘法与加法运算,饱和逻辑可将结果限定到16位。

[0096] 在每一级限定内部位深度为16位可使得变换处理单元52能够利用用于所述层级的计算上有效的指令集。举例来说,输入到变换的位深度可为9位并且起始内部位深度可大于16位。在一个实例中,变换处理单元52可仅能够执行特定层级操作(例如,16位操作、32位操作或64位操作等)。因此,当内部位深度大于16位时,可需要变换处理单元52执行至少32位层级的内部操作。为了避免此结果,变换处理单元52可将变换的每一分解级的内部位深度限定为16位(例如,以使得不需要32位内部操作)。换句话说,由于每一级的内部位深度限定为16位,处理器可利用计算上有效的指令集,所述计算上有效的指令集经设计特别供与16位操作(例如,ARM结构、高级SIMD (NEON)、数字信号处理 (DSP) 等)一起使用。

[0097] 随后在任选框425处,变换处理单元52可视情况通过将每一级的经限定值偏差(例如,误差)与阈值相比较而确定经限定值是否将产生令人满意的结果。举例来说,在框425处,方法500可通过执行下文关于图5所论述的方法500的框510和520而确定经限定值是否将产生令人满意的结果。如果经限定值将产生令人满意的结果,那么方法400继续进行至框430。否则的话,在框427处,变换处理单元52可重新计算系数的子集直至偏差小于阈值(例如,如下文关于方法500的框530所论述)。此方法500可允许变换处理单元52增大译码效率同时还保留视频质量。变换处理单元52已经确定经限定值将产生令人满意的结果之后,那么在框430处,变换处理单元52可使用针对16位数据(例如,ARM结构、高级SIMD (NEON)、数字信号处理 (DSP) 等)优化的专用指令集继续对变换数据执行16位操作。在框490处,所述方法结束。

[0098] 图5说明用于低复杂度正变换的方法500的流程图。所述方法500可由处理器或编码器(例如,上文关于图2所描述的编码器20)来执行。在一个实施例中,编码器的变换处理单元(例如,图2的编码器20的变换处理单元52)可用于执行所述方法500。实际上,虽然描述方法500为由编码器20的变换处理单元52执行,但应理解,所述方法500可由不同处理器、编码器或编码器的处理单元来执行。在一个实施例中,方法500首先确定限定值(例如,从图4的方法的经限定值)是否将产生令人满意的结果并因此而调整。如上文关于图4所解释,当变换处理单元52限定内部位深度时,其还可降低变换输出的质量。关于图5所描述的方法确定可发生质量降低并且因此而调整直至偏差小于预定阈值的程度。

[0099] 方法500开始于框505处。在方法500的开始处,变换处理单元52已经将全大小正变换矩阵分解成多个不复杂的级。举例来说,正变换矩阵可已经根据关于图4所描述的方法而分解。每一级的内部位深度可已经限定为特定深度值(例如,16位深度值),如关于图4进一步论述。

[0100] 随后在框510处,变换处理单元52可确定经限定值(例如,最终系数)与原始值的偏差。作为一实例,当使用饱和逻辑时,最终系数值可偏离原始值(例如,含有误差层级)。

[0101] 在计算所述偏差之后,在框520处,变换处理单元52可确定所述偏差是否大于预定阈限。如果是这样,那么方法500进行到框530。在框530处,变换处理单元52可重新计算系数的子集(例如,通过部分应用变换来重新计算AC层级系数的子集)并且使用重新计算的结果来替换同一经限定值子集。此过程可补偿最终系数值与原始值的偏差同时也将计算要求(例如,周期)和复杂度最小化。在一个实施中,系数的子集可仅含有DC层级系数,因为DC层级系数具有比AC层级系数更大的动态范围;因此,其更可能偏离原始值。在其它实施中,且为了进一步增大精确性,系数的子集可含有具有接近除含有DC层级系数外的DC层级的值的

AC层级系数。

[0102] 在重新计算了如上文所解释的小的系数子集之后,变换处理单元可随后返回至框510并且又检查与阈值的偏差。当变换处理单元52已经确定所述偏差小于预定阈值持久极限时(在框520处),方法500进行到框590并结束。在框590处,变换处理单元52可确定经限定值将产生令人满意的结果,并且提供经重新计算的系数子集。变换处理单元52可随后利用在方法400的框427处重新计算的系数子集,如上文关于图4所论述。

[0103] 上述所有方法是关于 $N \times N$ 变换描述的。然而,本发明中所描述的方法不限于 $N \times N$ 变换。方法还可关于任何大小的变换而实施,无论变换中之维度的数目。

[0104] 应认识到,取决于实例,本文中所描述的技术中的任一者的某些动作或事件可用不同顺序执行、可添加、合并或全部省略(例如,实践所述技术并不需要所有的所描述动作或事件)。此外,在某些实例中,可例如经由多线程处理、中断处理或多个处理器同时而非循序执行动作或事件。

[0105] 在一或多个实例中,所描述的功能可以硬件、软件、固件或其任何组合来实施。如果以软件实施,那么所述功能可作为一或多个指令或代码在计算机可读媒体上存储或发射,且由基于硬件的处理单元执行。计算机可读媒体可包含计算机可读存储媒体,其对应于有形媒体(例如,数据存储媒体),或包含促进将计算机程序从一处传送到另一处(例如,根据通信协议)的任何媒体的通信媒体。以此方式,计算机可读媒体通常可对应于(1)有形计算机可读存储媒体,其为非暂时性的,或(2)通信媒体,例如信号或载波。数据存储媒体可为可由一或多个计算机或一或多个处理器存取以检索用于实施本发明中所描述的技术的指令、代码和/或数据结构的任何可用媒体。计算机程序产品可包含计算机可读媒体。

[0106] 借助于实例而非限制,此类计算机可读存储媒体可包括RAM、ROM、EEPROM、CD-ROM或其它光盘存储装置、磁盘存储器或其它磁性存储装置、快闪存储器或可用于存储指令或数据结构的形式的期望程序代码并且可以通过计算机存取的任何其它媒体。又,可恰当地将任何连接称作计算机可读媒体。举例来说,如果使用同轴缆线、光纤缆线、双绞线、数字订户线(DSL)或无线技术(例如,红外线、无线电及微波)从网站、服务器或其它远程源发射指令,那么将同轴电缆、光纤缆线、双绞线、DSL或无线技术(例如,红外线、无线电及微波)包含在媒体的定义中。但是,应理解,所述计算机可读存储媒体和数据存储媒体并不包含连接、载波、信号或其它暂时性媒体,而是实际上针对于非暂时性有形存储媒体。如本文所使用,磁盘和光盘包含压缩光盘(CD)、激光光盘、光学光盘、数字多功能光盘(DVD)、软性磁盘及蓝光光盘,其中磁盘通常以磁性方式再现数据,而光盘用激光以光学方式再现数据。以上各者的组合也应包含在计算机可读媒体的范围内。

[0107] 可由一或多个处理器(例如,一或多个数字信号处理器(DSP)、通用微处理器、专用集成电路(ASIC)、现场可编程逻辑阵列(FPGA)或其它等效集成或离散逻辑电路)来执行所述指令。因此,如本文中所使用的术语“处理器”可指前述结构或适合于实施本文中所描述的技术的任何其它结构中的任一者。另外,在一些方面,本文所述的功能性可在经配置用于编码及解码的专用硬件及/或软件模块内提供,或者并入在组合式编解码器中。又,所述技术可完全实施于一或多个电路或逻辑元件中。

[0108] 本发明的技术可在多种多样的装置或设备中实施,包含无线手持机、集成电路(IC)或IC集合(例如,芯片组)。在本发明中描述各种组件、模块或单元是为了强调经配置以

执行所揭示的技术的装置的功能方面,但未必需要由不同硬件单元来实现。实际上,如上文所描述,各种单元可配合合适的软件和/或固件组合在一个编解码器硬件单元中,或者通过互操作硬件单元的集合来提供,所述硬件单元包含如上文所描述的一或多个处理器。

[0109] 已描述了各种实例。这些及其它实例属于所附权利要求书的范围内。

[0110] 附录A: 16×16 正变换的基于网格的实施方案的实例

```

for (j=0; j<16; j++)
{
    /* E and O*/
    for (k=0; k<8; k++)
    {
        nE[k] = pSrc[k] + pSrc[15-k];
        nO[k] = pSrc[k] - pSrc[15-k];
    }
    /* EE and EO */
    for (k=0; k<4; k++)
    {
        nEE[k] = nE[k] + nE[7-k];
        nEO[k] = nE[k] - nE[7-k];
    }
    /* EEE and EEO */
    nEEE[0] = nEE[0] + nEE[3];
    nEEO[0] = nEE[0] - nEE[3];
    nEEE[1] = nEE[1] + nEE[2];
    nEEO[1] = nEE[1] - nEE[2];

    pDst[ 0 ] = (anTransCoef16[ 0][0]*nEEE[0] + anTransCoef16[ 0][1]*nEEE[1] + 4)>>3;
    pDst[ 8 ] = (anTransCoef16[ 8][0]*nEEE[0] + anTransCoef16[ 8][1]*nEEE[1] + 4)>>3;
    pDst[ 4 ] = (anTransCoef16[ 4][0]*nEEO[0] + anTransCoef16[ 4][1]*nEEO[1] + 4)>>3;
    pDst[12] = (anTransCoef16[12][0]*nEEO[0] + anTransCoef16[12][1]*nEEO[1] + 4)>>3;

    for (k=2; k<16; k+=4)
    {
        pDst[ k ] = (anTransCoef16[k][0]*nEO[0] + anTransCoef16[k][1]*nEO[1] +
            anTransCoef16[k][2]*nEO[2] + anTransCoef16[k][3]*nEO[3] + 4)>>3;
    }

    for (k=1; k<16; k+=2)
    {
        pDst[ k ] = (anTransCoef16[k][0]*nO[0] + anTransCoef16[k][1]*nO[1] +

```

```

        anTransCoef16[k][2]*nO[2] + anTransCoef16[k][3]*nO[3] +
        anTransCoef16[k][4]*nO[4] + anTransCoef16[k][5]*nO[5] +
        anTransCoef16[k][6]*nO[6] + anTransCoef16[k][7]*nO[7] + 4)>>3;
    }

    pSrc += 16;
    pDst += 16;

}

pSrc = pCoef;
pDst = pRes;
for (j=0; j<16; j++)
{
    /* E and O */
    for (k=0; k<8; k++)
    {
[0112]    nE[k] = pSrc[k*16] + pSrc[(15-k)*16];
        nO[k] = pSrc[k*16] - pSrc[(15-k)*16];
    }
    /* EE and EO */
    for (k=0; k<4; k++)
    {
        nEE[k] = nE[k] + nE[7-k];
        nEO[k] = nE[k] - nE[7-k];
    }
    /* EEE and EEO */
    nEEE[0] = nEE[0] + nEE[3];
    nEEO[0] = nEE[0] - nEE[3];
    nEEE[1] = nEE[1] + nEE[2];
    nEEO[1] = nEE[1] - nEE[2];

    pDst[ 0 ] = (anTransCoef16[ 0][0]*nEEE[0] + anTransCoef16[ 0][1]*nEEE[1] +
    512)>>10;
    pDst[ 8*16 ] = (anTransCoef16[ 8][0]*nEEE[0] + anTransCoef16[ 8][1]*nEEE[1] +

```

[0113]

512)>>10;

pDst[4*16] = (anTransCoef16[4][0]*nEEO[0] + anTransCoef16[4][1]*nEEO[1] +
512)>>10;

pDst[12*16] = (anTransCoef16[12][0]*nEEO[0] + anTransCoef16[12][1]*nEEO[1] +
512)>>10;

for (k=2;k<16;k+=4)

{

pDst[k*16] = (anTransCoef16[k][0]*nEO[0] + anTransCoef16[k][1]*nEO[1] +
anTransCoef16[k][2]*nEO[2] + anTransCoef16[k][3]*nEO[3] + 512)>>10;

}

for (k=1;k<16;k+=2)

{

pDst[k*16] = (anTransCoef16[k][0]*nO[0] + anTransCoef16[k][1]*nO[1] +
anTransCoef16[k][2]*nO[2] + anTransCoef16[k][3]*nO[3] +
anTransCoef16[k][4]*nO[4] + anTransCoef16[k][5]*nO[5] +
anTransCoef16[k][6]*nO[6] + anTransCoef16[k][7]*nO[7] + 512)>>10;

}

pSrc ++;

pDst ++;

}

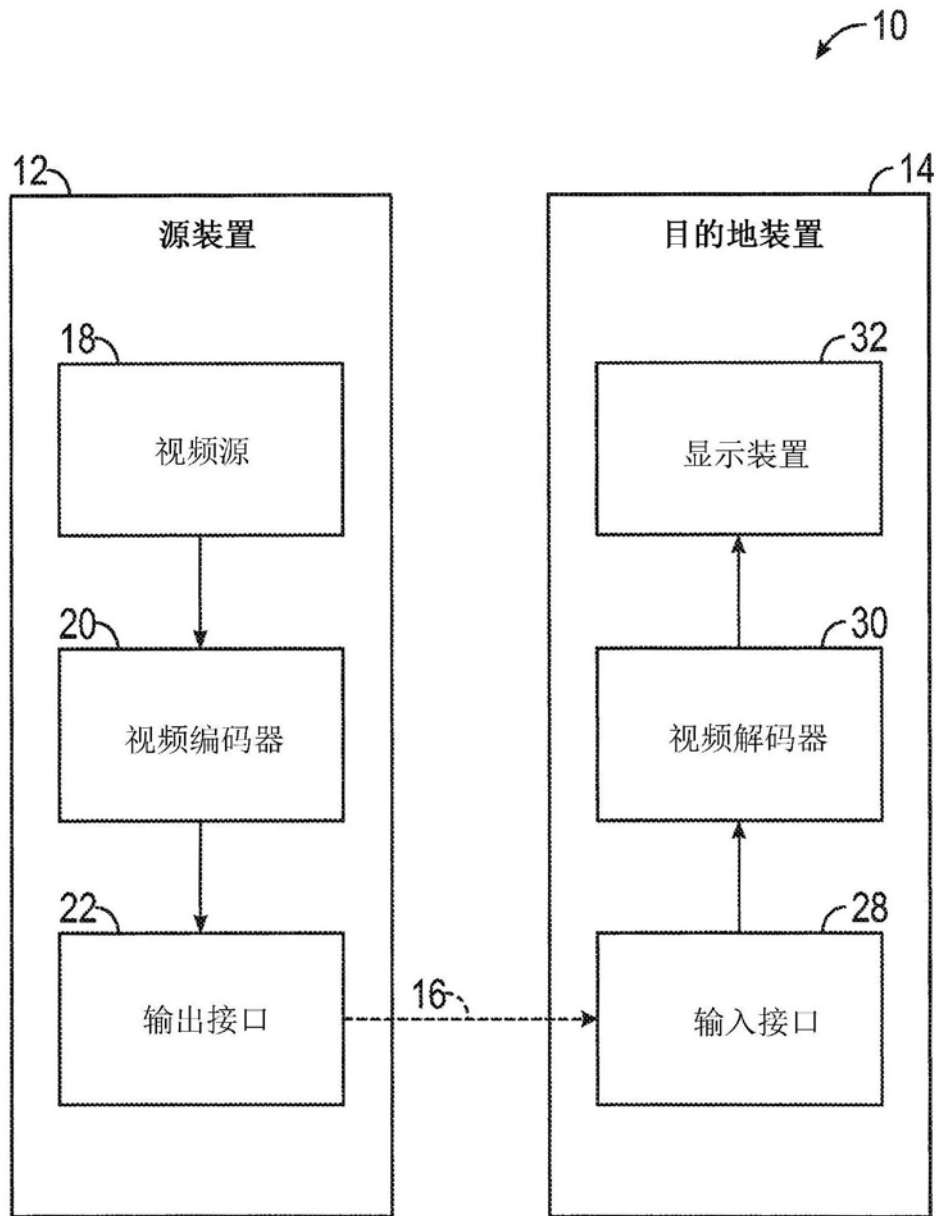


图1

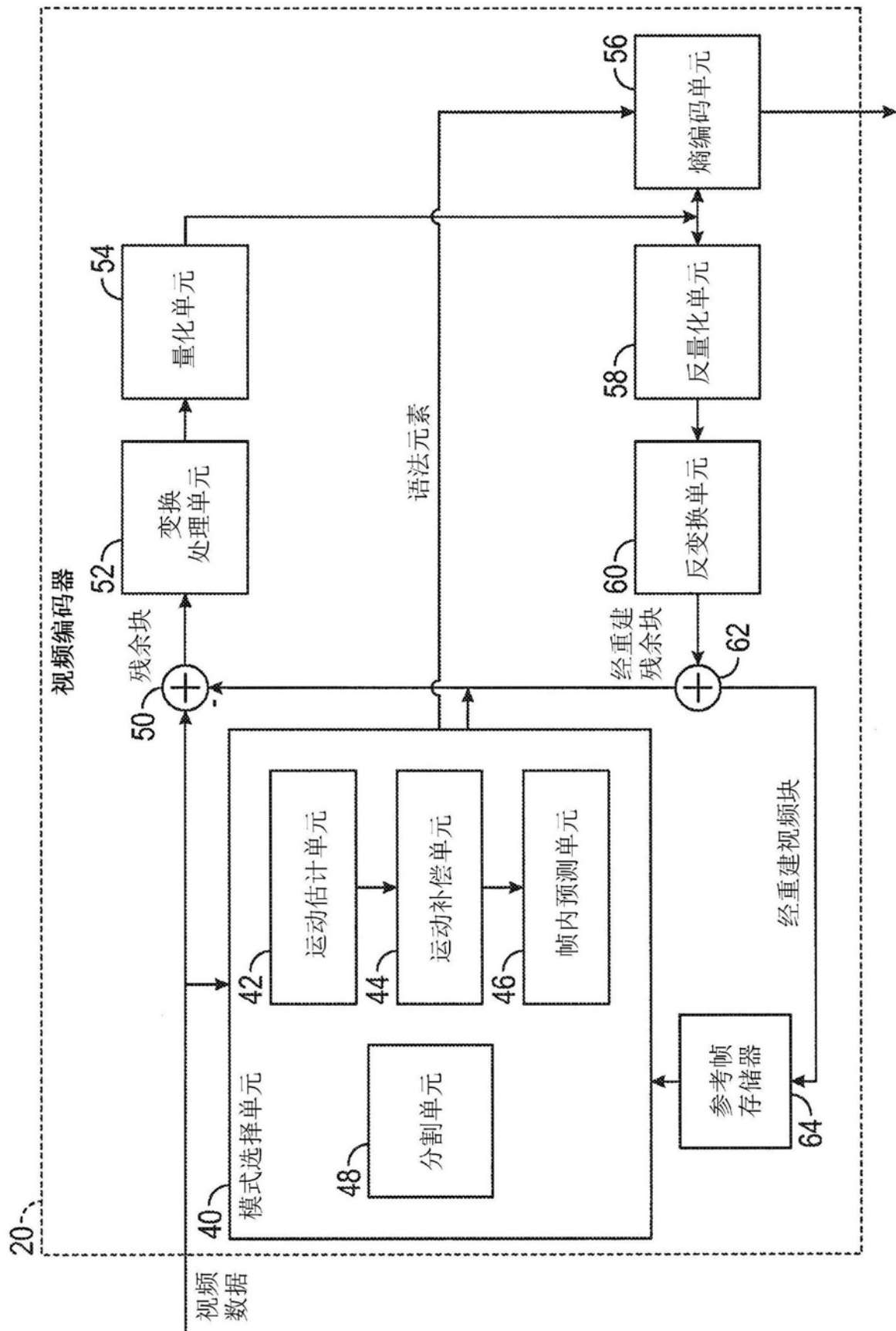


图2

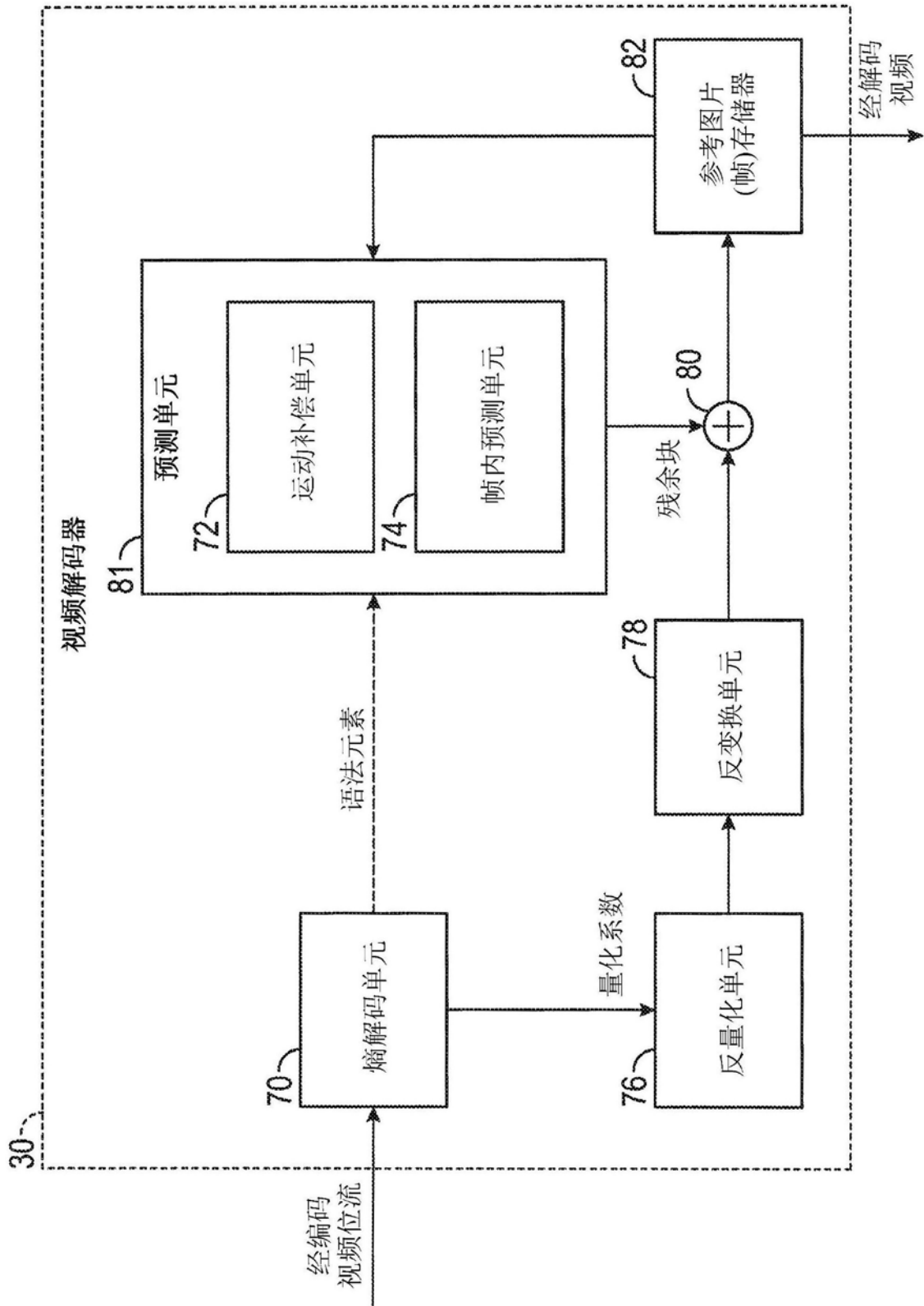


图3

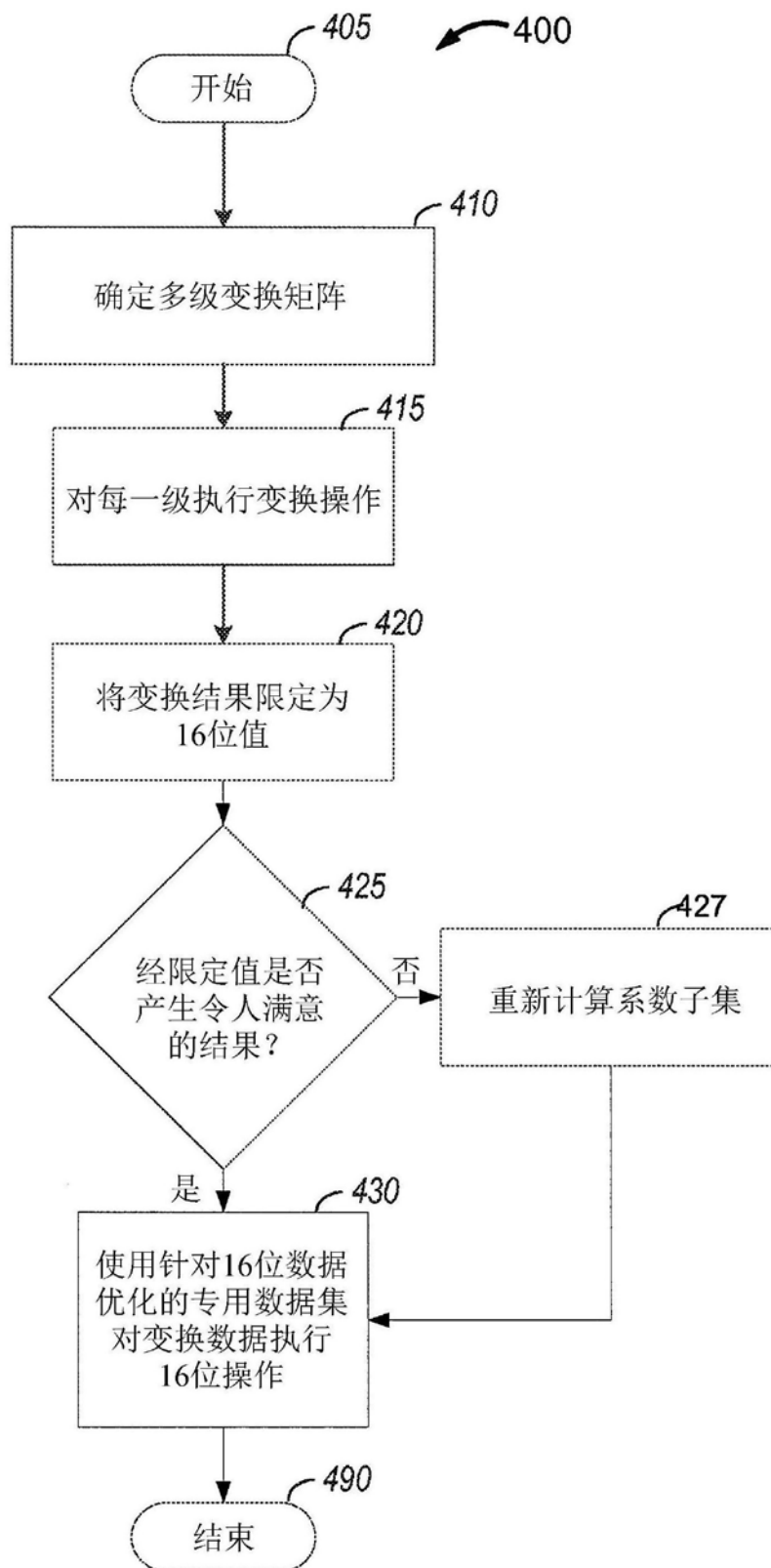


图4

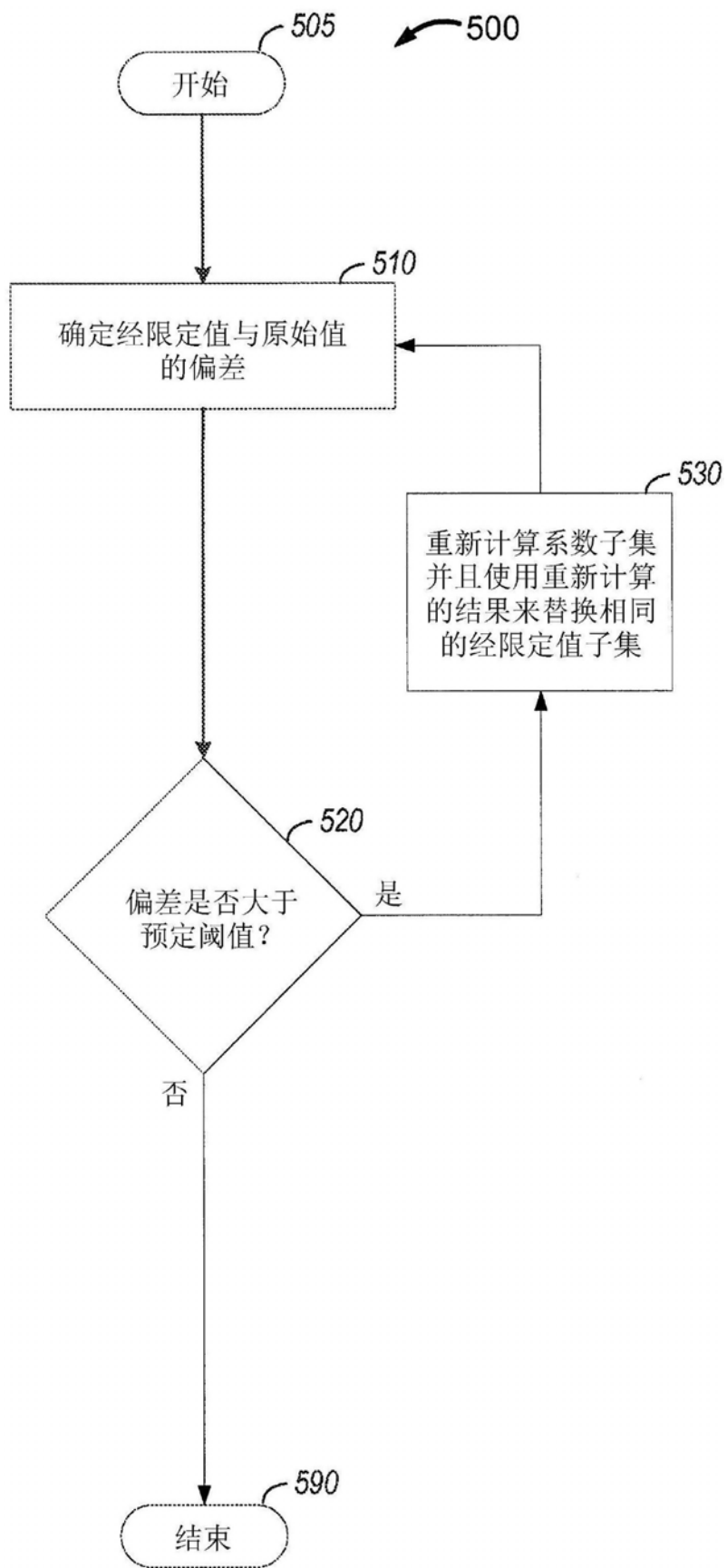


图5