



(12) 发明专利申请

(10) 申请公布号 CN 118900328 A

(43) 申请公布日 2024. 11. 05

(21) 申请号 202411142581.2

H04N 19/103 (2014.01)

(22) 申请日 2019.05.21

H04N 19/176 (2014.01)

(30) 优先权数据

H04N 19/33 (2014.01)

18305653.0 2018.05.29 EP

H04N 19/46 (2014.01)

H04N 19/593 (2014.01)

(62) 分案原申请数据

201980035564.X 2019.05.21

(71) 申请人 交互数字VC控股公司

地址 美国特拉华州

(72) 发明人 F·莱林内克 F·加尔平

T·波伊里尔 G·拉思

(74) 专利代理机构 北京市柳沈律师事务所

11105

专利代理师 赵碧洋

(51) Int. Cl.

H04N 19/119 (2014.01)

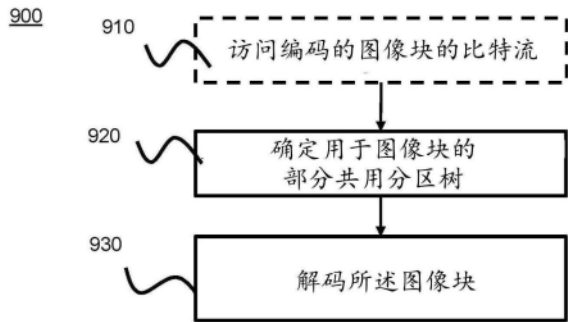
权利要求书2页 说明书13页 附图11页

(54) 发明名称

用于编码和解码的方法、装置和介质

(57) 摘要

提供了一种用于编码和解码的方法、装置和介质。该方法包括：将视频图像分区为多个编码树单元；对指示进行解码，所述指示为：对于I条带，使用隐式二叉树划分将每个编码树单元划分为具有64×64个亮度样本的编码单元，并且所述编码单元是用于亮度和色度的两个独立的编码树语法结构的根；以及基于所述指示对所述多个编码树单元进行解码。



1. 一种方法,包括:

将视频图像分区为多个编码树单元;

对指示进行解码,所述指示为:对于I条带,使用隐式二叉树划分将每个编码树单元划分为具有 $64 \times 64$ 个亮度样本的编码单元,并且所述编码单元是用于亮度和色度的两个独立的编码树语法结构的根;以及

基于所述指示对所述多个编码树单元进行解码。

2. 根据权利要求1所述的方法,还包括:在当前编码单元具有多于 $64 \times 64$ 个亮度样本的情况下,将所述当前编码单元分区为4个大小相等的正方形编码单元。

3. 根据权利要求1所述的方法,还包括:在当前编码单元具有小于或等于 $64 \times 64$ 个亮度样本的情况下,使用亮度编码树结构来对所述当前编码单元的亮度分量进行分区,并且使用色度编码树结构来对所述当前编码单元的色度分量进行分区。

4. 根据权利要求3所述的方法,其中,使用色度编码树结构来对所述当前编码单元的色度分量进行分区还包括:对所述当前编码单元的色度分量被进一步划分的至少指示进行解码。

5. 根据权利要求1所述的方法,其中,所述指示是从在条带级别上信号通知的语法元素中解码的。

6. 一种装置,包括:

存储器,和

一个或多个处理器,被配置为:

将视频图像分区为多个编码树单元;

对指示进行解码,所述指示为:对于I条带,使用隐式二叉树划分将每个编码树单元划分为具有 $64 \times 64$ 个亮度样本的编码单元,并且所述编码单元是用于亮度和色度的两个独立的编码树语法结构的根;以及

基于所述指示对所述多个编码树单元进行解码。

7. 根据权利要求6所述的装置,其中,所述一个或多个处理器还被配置为:在当前编码单元具有多于 $64 \times 64$ 个亮度样本的情况下,将所述当前编码单元分区为4个大小相等的正方形编码单元。

8. 根据权利要求6所述的装置,其中,所述一个或多个处理器还被配置为:在当前编码单元具有小于或等于 $64 \times 64$ 个亮度样本的情况下,使用亮度编码树结构来对所述当前编码单元的亮度分量进行分区,并且使用色度编码树结构来对所述当前编码单元的色度分量进行分区。

9. 根据权利要求8所述的装置,其中,为了进一步使用色度编码树结构来对所述当前编码单元的色度分量进行分区:所述一个或多个处理器还被配置为对所述当前编码单元的色度分量被进一步划分的至少指示进行解码。

10. 根据权利要求6所述的装置,其中,所述指示是从在条带级别上信号通知的语法元素中解码的。

11. 一种方法,包括:

确定编码树结构,所述编码树结构对应于将视频图像的编码树单元分区为多个编码单元;

对指示进行编码,所述指示:对于I条带,使用隐式二叉树划分将每个编码树单元划分为具有 $64 \times 64$ 个亮度样本的编码单元,并且所述编码单元是用于亮度和色度的两个独立的编码树语法结构的根;以及

基于所述指示对所述编码树单元进行编码。

12. 根据权利要求11所述的方法,还包括:在当前编码单元具有多于 $64 \times 64$ 个亮度样本的情况下,确定所述当前编码单元被分区为4个相等大小的正方形编码单元。

13. 根据权利要求11所述的方法,还包括:在当前编码单元具有小于或等于 $64 \times 64$ 个亮度样本的情况下,确定使用亮度编码树结构来分区所述当前编码单元的亮度分量,并且使用色度编码树结构来分区所述当前编码单元的色度分量。

14. 根据权利要求13所述的方法,还包括:对所述当前编码单元的色度分量被进一步划分的至少指示进行编码。

15. 根据权利要求11所述的方法,其中,所述指示被编码为在条带级别上信号通知的语法元素。

16. 一种装置,包括:

存储器,和

一个或多个处理器,被配置为:

确定编码树结构,所述编码树结构对应于将视频图像的编码树单元分区为多个编码单元;

对指示进行编码,所述指示:对于I条带,使用隐式二叉树划分将每个编码树单元划分为具有 $64 \times 64$ 个亮度样本的编码单元,并且所述编码单元是用于亮度和色度的两个独立的编码树语法结构的根;以及

基于所述指示对所述编码树单元进行编码。

17. 根据权利要求16所述的装置,其中,所述一个或多个处理器还被配置为:在当前编码单元具有多于 $64 \times 64$ 个亮度样本的情况下,确定所述当前编码单元被分区为4个相等大小的正方形编码单元。

18. 根据权利要求16所述的装置,其中,所述一个或多个处理器还被配置为:在当前编码单元具有小于或等于 $64 \times 64$ 个亮度样本的情况下,确定使用亮度编码树结构来分区所述当前编码单元的亮度分量,并且使用色度编码树结构来分区所述当前编码单元的色度分量。

19. 根据权利要求18所述的装置,其中,所述一个或多个处理器还被配置为:对所述当前编码单元的色度分量被进一步划分的至少指示进行编码。

20. 根据权利要求16所述的装置,其中,所述指示被编码为在条带级别上信号通知的语法元素。

21. 一种非暂时性计算机可读存储介质,其上存储有计算机程序指令,所述指令由处理器执行以实施根据权利要求1至5或11至15中任一项所述的方法。

## 用于编码和解码的方法、装置和介质

[0001] 本申请是申请日为2019年05月21日、申请号为201980035564.X、发明名称为“利用部分共用亮度和色度编码树的视频编码和解码的方法和装置”的发明专利申请的分案申请。

### 技术领域

[0002] 本实施例中的至少一个一般涉及例如用于视频编码或解码的方法或装置,并且更具体地涉及其中在亮度和色度块之间部分共用分区树结构的方法或装置。

### 背景技术

[0003] 一个或多个实现方式的技术领域一般与视频压缩有关。与现有视频压缩系统相比,诸如HEVC(HEVC是指高效视频编码,也称为“ITU(10/2014)的ITU-T H.265电信标准部门(10/2014),系列H:视听和多媒体系统,视听服务的基础设施-移动视频的编码,高效视频编码,推荐ITU-T H.265”中描述的H.265和MPEG-H部分2),或与开发的视频压缩系统相比,诸如,VVC(多功能视频编码,由联合视频专家小组JVET开发的新标准),至少一些实施例涉及改进压缩效率。

[0004] 为了实现高压压缩效率,图像和视频编码方案通常采用图像的分区,预测(包括运动矢量预测)和变换来利用视频内容中的空间和时间冗余。一般地,使用帧内或帧间预测来利用帧内或帧间相关性,然后变换,量化和熵编码原始图像和预测的图像之间的差(经常表示为预测误差或预测残差)。为了重构视频,通过与熵解码,逆变换,逆量化和预测对应的逆处理来解码压缩的数据。

[0005] 随着新的视频编码方案的出现,分区方案变得更加复杂,并且允许亮度和色度的双树分区/编码以实现高压压缩。然而,似乎双树编码方案在解码阶段产生了硬件实现方式问题。

[0006] 因此,需要一种新的用于编码和解码视频的方法,该方法联合确保高编码效率,同时考虑到一些硬件解码流水线约束。

### 发明内容

[0007] 本发明的目的是克服现有技术的至少一个缺点。为此,根据至少一个实施例的一般方面,提出了一种用于视频编码的方法,包括:确定与视频的图像的块的分区处理对应的分区树结构,其中,分区树结构在亮度和色度块之间部分共用;并且,至少基于分区树结构编码块。

[0008] 根据至少一个实施例的另一一般方面,提出了用于视频解码的方法,包括:确定与视频的图像的块的分区处理对应的分区树结构,其中,分区树结构在亮度和色度块之间部分共用;并且,至少基于所述分区树结构解码块。

[0009] 根据至少一个实施例的另一一般方面,提出了用于视频编码的装置,包括用于实现编码方法的任何一个实施例的部件。

[0010] 根据至少一个实施例的另一一般方面,提出了用于视频解码的装置,包括用于实现解码方法的任何一个实施例的部件。

[0011] 根据至少一个实施例的另一一般方面,提出了用于视频编码的装置,包括一个或多个处理器和至少一个存储器。一个或多个处理器被配置为实现编码方法的实施例中的任何一个。

[0012] 根据至少一个实施例的另一一般方面,提出了用于视频解码的装置,包括一个或多个处理器和至少一个存储器。一个或多个处理器被配置为实现解码方法的实施例中的任何一个。

[0013] 根据至少一个实施例的另一一般方面,在从分区树的根节点向下到切换节点的亮度和色度块之间共用单个分区树结构,并且确定双分区树结构用于从切换节点向下到分区树的叶节点的亮度和色度块。有利地,该实施例通过允许分离的树用于较小的块来优化编码效率,同时限制解码流水线的复杂性。

[0014] 根据至少一个实施例的另一一般方面,部分共用的分区树结构用于编码或者解码属于帧内编码的视频图像的一部分的块,例如,图块或图块组。

[0015] 根据至少一个实施例的另一一般方面,部分共用的分区树结构用于编码或者解码属于帧间编码的视频图像的一部分的块,例如,图块或图块组。

[0016] 根据至少一个实施例的另一一般方面,隐式确定切换节点。根据该实施例的变型,基于分区块大小来确定切换块节点。有利地,分区块大小对应于解码流水线单元大小,并且例如被设置为 $64 \times 64$ 。根据该实施例的另一变型,基于分区深度级别来确定切换节点。有利地,分区深度级别被设置为分区树的二叉树分割的一个深度,并且对应于将 $128 \times 128$  CTU二叉树分割为大小为 $64 \times 64$ 的4个子CU,从而与解码流水线单元大小兼容。

[0017] 解码流水线单元也可以称为虚拟解码流水线单元(VPDU)。

[0018] 根据至少一个实施例的另一一般方面,切换节点被发信号通知。有利地,该实施例允许例如基于RD成本为每个CTU适配切换节点。

[0019] 根据至少一个实施例的另一一般方面,提出了非暂时性计算机可读介质,其包含根据前述任一描述的方法或装置生成的数据内容。

[0020] 根据至少一个实施例的另一一般方面,提供一种信号,该信号包括根据前述描述中的任一个的方法或装置生成的视频数据。根据至少一个实施例的另一一般方面,基于分区块大小来交织信号数据。因此,信号数据有利地被布置用于解码流水线单元。

[0021] 根据至少一个实施例的又一一般方面,提出了一种方法,包括:将视频图像分区为多个编码树单元;对指示进行解码,所述指示为:对于I条带,使用隐式二叉树划分将每个编码树单元划分为具有 $64 \times 64$ 个亮度样本的编码单元,并且所述编码单元是用于亮度和色度的两个独立的编码树语法结构的根;以及基于所述指示对所述多个编码树单元进行解码。

[0022] 根据至少一个实施例的又一一般方面,提出了一种装置,包括:存储器,和一个或多个处理器,被配置为:将视频图像分区为多个编解码编码树单元;对指示进行解码,所述指示为:对于I条带,使用隐式二叉树划分将每个编解码编码树单元划分为具有 $64 \times 64$ 个亮度样本的编解码编码单元,并且所述编解码编码单元是用于亮度和色度的两个独立的编解码编码树语法结构的根;以及基于所述指示对所述多个编解码编码树单元进行解码。

[0023] 根据至少一个实施例的又一一般方面,提出了一种方法,包括:确定编解码编码树

结构,所述编解码编码树结构对应于将视频图像的编解码编码树单元分区为多个编解码编码单元;对指示进行编码,所述指示:对于I条带,使用隐式二叉树划分将每个编解码编码树单元划分为具有 $64 \times 64$ 个亮度样本的编解码编码单元,并且所述编解码编码单元是用于亮度和色度的两个独立的编解码编码树语法结构的根;以及基于所述指示对所述编解码编码树单元进行编码。

[0024] 根据至少一个实施例的又一一般方面,提出了一种装置,包括:存储器,和一个或多个处理器,被配置为:确定编码树结构,所述编码树结构对应于将视频图像的编码树单元分区为多个编码单元;对指示进行编码,所述指示:对于I条带,使用隐式二叉树划分将每个编码树单元划分为具有 $64 \times 64$ 个亮度样本的编码单元,并且所述编码单元是用于亮度和色度的两个独立的编码树语法结构的根;以及基于所述指示对所述编解码编码树单元进行编码。

[0025] 根据至少一个实施例的又一一般方面,提出了一种非暂时性计算机可读存储介质,其上存储有计算机程序指令,所述指令由处理器执行以实施根据上述方面所述的方法。

[0026] 本实施例中的一个或多个还提供一种计算机可读存储介质,其上存储有用于根据上述任何方法编码或解码视频数据的指令。本实施例还提供一种计算机可读存储介质,其上存储有根据上述方法生成的比特流。本实施例还提供一种用于发送根据上述方法生成的比特流的方法和装置。本实施例还提供一种计算机程序产品,该计算机程序产品包括用于执行所描述的任何方法的指令。

## 附图说明

[0027] 图1例示编码树单元(CTU)和编码树(CT)概念以表示压缩的HEVC画面的示例。

[0028] 图2例示在二叉树加二叉树(QTBT)方案中编码单元和相关联的编码树的分区的示例。

[0029] 图3和图4例示一些CU二叉树或三叉树分区的示例。

[0030] 图5和图6例示根据至少一个实施例的一般方面的分别在亮度和色度分量中的CTU的非限制性实施例的分区的示例。

[0031] 图7和图8例示根据至少一个实施例的一般方面的编码方法的各种示例。

[0032] 图9、10、11和12例示根据至少一个实施例的一般方面的解码方法的各种示例。

[0033] 图13例示其中可以实现实施例的各个方面的视频编码器的实施例的框图。

[0034] 图14例示其中可以实现实施例的各个方面的视频解码器的实施例的框图。

[0035] 图15示例示其中可以实现实施例的各个方面的示例装置的框图。

## 具体实施方式

[0036] 要理解,已经简化附图和描述以例示与清楚理解本原理有关的元素,同时为了清楚起见,消除在典型的编码和/或解码设备中发现的许多其他元素。将理解的是,尽管术语第一和第二可以在本文中用于描述各种元素,但是这些元素不应当被这些术语所限制。这些术语仅用于区分一个元素和另一个元素。

[0037] 关于图像的编码/解码描述各种实施例。它们可以应用于编码/解码图像的一部分,诸如条带或图块(tile),图块组或整个图像序列。

[0038] 以上描述各种方法,并且每种方法包括用于实现所描述的方法的一个或多个步骤或动作。除非方法的正确操作要求步骤或动作的特定顺序,否则可以修改或组合特定步骤和/或动作的顺序和/或使用。

[0039] 至少一些实施例涉及用于编码或解码视频的方法,其中树编码/分区结构在亮度和色度块之间部分共用。

[0040] 在HEVC视频压缩标准中,图像被划分为所谓的编码树单元(CTU),其大小典型为64x64、128x128或256x256像素。每个CTU由压缩域中的编码树表示。这是CTU的四叉树划分,其中每个叶片称为编码单元(CU)。然后,给予每个CU一些帧内或帧间预测参数(预测信息)。为此,将其在空间上分区为一个或多个预测单元(PU),每个PU均分配有一些预测信息。帧内或帧间编码模式在CU级别分配,如图1所示。

[0041] 新出现的视频压缩工具包括在压缩域中更灵活的编码树单元表示,与HEVC标准的CU/PU/TU布置相比,该压缩树提供更高的压缩效率。首先,编码树单元表示包括四叉树加二叉树(QTBT),这是编码结构,其中编码单元可以四叉树方式并且以二叉树方式二者分割。图2例示编码树单元的编码树表示的示例。编码单元的分割通过率失真优化过程在编码器端决定,该过程包括确定最小率失真成本的CTU的QTBT表示。在QTBT技术中,CU具有正方形或矩形形状。编码单元的大小始终是2的幂,典型从4到128。

[0042] 除了用于编码单元的多种矩形形状之外,与HEVC相比,这种新的CTU表示具有以下不同的特征。首先,CTU的QTBT分解分为两个阶段:首先以四叉树的方式分割CTU,然后可以以二元方式进一步划分每个四叉树的叶片。这在图2的右侧例示,其中实线表示四叉树分解阶段,并且虚线表示空间上嵌入四叉树叶片的二元分解。

[0043] 其次,在帧内条带中,亮度和色度块分区结构被分离,并且被独立决定。

[0044] 第三,不再采用CU分区成预测单元或变换单元。换句话说,每个编码单元系统地由单个预测单元( $2N \times 2N$ 个预测单元分区类型)和单个变换单元(不划分为变换树)组成。

[0045] 最后,如图3或图4例示,也可以在CTU编码树的表示中采用其他一些CU二叉树或三叉树分区。图3例示非对称二叉树中的CU分区,其中具有大小(w,h)(宽度和高度)的正方形或矩形编码单元通过非对称二元分割模式之一来分割,例如HOR\_UP(水平向上),从而导致具有各自的矩形大小 $(w, \frac{h}{4})$ 和 $(w, \frac{3h}{4})$ 的2个子编码单元。此外,图4表示CU的所谓三叉树分区。

三叉树在于在考虑的取向上将CU分割为相对于父UC大小为(1/4,1/2,1/4)的树子CU。

[0046] 使用上述新拓扑,带来显著编码效率改进。特别地,在色度上获得显著增益。色度分量的这种大增益很大程度上是由于帧内条带中亮度和色度编码树的分离导致。但是,CTU级别的亮度和色度编码树的这种分离在硬件实现方面存在一些问题。在至少一个实施例中,CTU典型尺寸为128x128或256x256。此外,完全分离亮度和色度分量的编码树意味着这些亮度和色度在压缩域中也被完全分离,因此也以分离的方式出现在编码比特流中。这对于至少一种解码器实现方式引起一些问题,其中,这要确保在可能小于CTU尺寸的最大解码单元大小上实现解码流水线。

[0047] 典型地,至少一种解码器实现方式期望基于64x64的解码器流水线。为此,已选择等于64x64的最大变换块大小。另外,移除亮度和色度编码树的分离,因为这意味着在同一CTU中CTU的所有亮度块出现在色度块之前的编码比特流。在这种情况下,解码处理将导致

解析,因此生成将在空间上覆盖整个CTU,即128x128或256x256画面区域的解码的量化块数据。与独立选择、编码和解码这些树的情况相比,移除亮度/色度编码分离的结果是编码效率的损失。

[0048] 因此,期望以这种方式来具有亮度和色度编码树表示:优化编码效率,同时确保解码流水线,例如基于64x64块的解码流水线是可行的。

[0049] 为此,至少一个实施例包括:

[0050] -具有从最大允许块大小(CTU大小)向下到预定义块大小或向上到预定义CU深度级别的联合亮度/色度编码树,这根据流水线解码单元确定。然后,当块大小小于或等于流水线解码单元时,以分离的方式选择、编码和解码亮度和色度编码树。这意味着亮度和色度分量共用整个CTU编码树的一部分,并且然后,超出CTU编码树中的某个深度级别,亮度和色度树是独立的,因此可以不同。在本说明书的下文中,

[0051] 为了提及该方面,使用术语部分共用亮度/色度编码树。

[0052] -根据与处理的编码树节点相关联的块大小,或根据编码/解码的编码树节点的深度级别,从交织的亮度/色度编码处理切换到单独的亮度/色度编码处理。

[0053] 在实践中,假设解码流水线单元为64x64,当对亮度和色度编码树编码时,编码树被联合决定并向下编码为大于64x64的块大小。然后,对于小于64x64的块大小,亮度/色度编码树可能彼此偏离。

[0054] 有利地,该处理还确保在比特流组织中,对于等于所考虑的解码流水线单元(64x64)的每个画面区域,亮度和色度压缩信号一个接一个地出现。因此,当解析比特流时,确保在开始处理下一个流水线解码单元之前,对于亮度和色度分量二者,完全解码与等于流水线解码单元64x64的画面区域相关联的块残差信号。

[0055] 此外,与亮度/色度编码树的完全分离相比,提高了编码效率。

[0056] 在第1节中,公开了部分共用亮度/色度编码树和相关联的比特流组织的若干实施例。

[0057] 在第2节中,公开了编码方法的若干实施例。

[0058] 在第3和4节中,公开了解码方法的若干个实施例。

[0059] 在第5节中,公开了附加信息和通用实施例。

[0060] 1. 部分共用的亮度/色度编码树和相关联的比特流组织的实施例

[0061] 图5和图6例示根据至少一个实施例的一般方面的分别在亮度和色度分量中的CTU的非限制性实施例的分区示例。在两个示例上,考虑的CTU大小为128x128,并且考虑的解码流水线单元为64x64。

[0062] 图5例示第一示例,其中,首先以二叉树的方式分割CTU。这导致亮度分量中的64x64编码单元(如左图所示),其与色度分量中对应的32x32 CU对齐(如右图所示)。可以看出,当要分割的CU的大小大于解码单元大小 $64 \times 64$ 时,两个亮度和色度分割类型是完全相同的。接下来,对于小于或等于64x64的亮度CU,在亮度和色度之间分割模式可能不同。图5就是这种情况,除了右上方的64x64 CU(3和4)。

[0063] 在图5的亮度和色度编码树的情况下,图5的底部示出这里提出的比特流组织。可以看出,根据比特流布置的实施例,比特流由分别包括一系列编码的亮度块和一系列编码的色度块的连续部分形成。以这样的方式组织比特流:一系列编码的亮度或色度块对应于

不大于 $64 \times 64$  (在亮度域中) 的空间区域, 即, 施加的用于解码流水线单元的亮度大小。

[0064] 在特定实施例中, 包括给定的一系列编码的亮度或色度块的比特流部分恰好包含所考虑的画面的空间区域的编码数据, 其大小等于所考虑的解码流水线单元大小。典型地, 该大小为 $64 \times 64$ 。

[0065] 图6例示部分共用的亮度/色度编码树的另一实施例。首先 $128 \times 128$ 经历垂直二元分割, 这导致两个 $64 \times 128$ 编码单元。由于对于大于 $64 \times 64$ 的CU大小, 亮度和色度被同步, 所以CTU在色度分量中以相同的方式分割, 从而导致两个 $64 \times 32$  CU。右侧CU不再分割。左侧 $128 \times 64$ 以亮度和色度分量同步方式通过二元水平分割模式被划分 (根据至少一个实施例的编码树布置的共用/同步部分)。

[0066] 此外, 关于由该第二二元分割阶段产生的CU, 因此在亮度中具有大小 $64 \times 64$ , 它们可以以亮度和色度分量之间的分离方式进一步分割。最后, 在图6的底部示出作为结果的比特流布置和排序。这里要注意的一个特定方面是, 由于最大解码单元大小为 $64 \times 64$ , 所以最大变换大小也为 $64 \times 64$ 。因此, 在图6上的 $64 \times 128$  CU的情况下, CU在亮度分量中分割为2个变换单元 $64 \times 64$ 。同步地, 将其分割成2个色度分量中的两个 $32 \times 32$  CU。然后, 就比特流排序而言, 将这种空间划分产生的 $64 \times 64$ 亮度和 $32 \times 32$ 色度变换块在比特流中交织, 以确保对应于大小为 $64 \times 64$ 的空间区域的所有亮度和色度编码数据都被一起打包在编码比特流中。这对应于图6的编码数据块5、6、7和8。

[0067] 2. 编码的至少一个实施例

[0068] 图7例示根据至少一个实施例的一般方面的编码方法700的示例。

[0069] 在预备步骤710中, 访问用于编码的块。该块 (典型, CU) 属于视频的图像的空间分区, 在前面的节中典型是CTU。编码方法700针对要编码的块递归地确定分区或分割为子块。典型地, 该方法从前面几节中作为CTU的块开始。例如, 根据图2, 图3或图4描述的任何分割模式, 将该块分割为子块。分区树结构 (也称为编码树或分区信息) 指示图像如何被分区。因此, 在步骤720中, 确定与块的分区处理对应的分区树结构。有利地, 分区树结构在亮度和色度块之间部分共用。在亮度和色度块之间部分共用的分区树结构意味着对于分区树中的某些块, 使用单个分区树结构用于亮度和色度块二者, 而对于分区树中的其他块, 使用分开的分区树结构用于亮度块和色度块。例如, 对应于图5中CTU的亮度分量对应的亮度块与对应于CTU的任何色度分量的色度块共用分区树结构中的第一四叉树分割。然后, 对应于图5中CU 1的亮度分量的亮度块1和对应于CU 2的任何色度分量的色度块2以不同的方式分割, 因此确定双分区树结构用于亮度块1和在同一位置的色度块2。在下面, 术语“分离树”和“双树”无差别地用于定义2个树, 而术语公用树, 共用树, 联合树和单个树无差别地用于定义亮度和色度分量二者的公用树。在步骤730中, 至少基于确定的分区树结构来编码块。

[0070] 根据一个实施例, 在从分区树的根节点向下到切换节点的亮度和色度块之间共用单个分区树结构, 并且从切换节点向下到分区树的叶节点确定双分区树结构用于亮度和色度块。因此, 共用编码树用于较大的块, 而分离树用于较小的块, 共用编码树和双编码树之间的切换由切换点或级别定义。例如, 根节点对应于CTU, 而叶节点对应于叶片CU。有利地, 该实施例通过允许分离树用于一些块来优化编码效率, 同时限制解码流水线的复杂性。

[0071] 根据实施例, 部分共用分区树结构用于编码属于帧内编码的视频图像的一部分的块。例如, 视频图像的该部分是图块或图块组, 并且部分共用的亮度和色度编码树仅在帧内

图块或帧内图块组中使用。

[0072] 根据另一实施例,部分共用的分区树结构用于编码属于帧间编码的视频图像的一部分的块。

[0073] 根据实施例,隐式地确定切换节点。有利地,该实施例允许从编码器到解码器不用信号通知切换节点。在

[0074] 第一变型中,基于分区块大小来确定切换块节点。因此,分区块大小是在编码器和解码器中预定义的参数。例如,基于相对于所考虑的解码单元大小的编码单元的大小来决定从共用的亮度/色度编码树到2个分离的亮度和色度子树的切换。根据优选变型,分区块大小被设置为 $64 \times 64$ ,从而允许以 $64 \times 64$ 块为基础的解码流水线。如图5例示,CTU $128 \times 128$ 共用色度和亮度分量的编码树,而对于CU大小小于或等于 $64 \times 64$ 的CU,色度和亮度分量的子树被分离。

[0075] 在第二变型中,基于分区深度级别来确定切换块节点。因此,分区块大小是在编码器和解码器中预定义的参数。根据优选示例,将分区深度级别设置为分区树的二叉树分割的一个深度。如图5例示,CTU的共用树包括第一二叉树分割,子树下面对于CU被分离。例如,基于相对于CU深度阈值的编码单元的深度级别来决定从共用的亮度/色度编码树到2个分离的亮度和色度子树的切换。在特定变型中,推断第一二叉树分割。此变型非常适合于 $128 \times 128$ 的CTU大小,其中第一个QT分割会导致4个大小为 $64 \times 64$ 的CU,其中然后处理双树用于亮度和色度分量。换句话说,仅基于相对于CU二叉树深度阈值的二叉树深度级别来决定从共用的亮度/色度编码树向2个分离的亮度和色度子树的切换。实际上,在QTBT编码结构中,在进入所谓的二叉/三叉树分割处理之前,CTU首先经历二叉树分割处理。因此,可以考虑两种类型的CU深度级别:二叉树相关深度级别和二叉/三叉树深度级别。经历二叉树分割的CU具有等于0的二叉/三叉树级别,并且与父CU相比,二叉树分割导致二叉树相关的深度级别增加了1的4个子CU。

[0076] 在第三变型中,还基于分区深度级别来确定切换块节点,其中,将复合二叉树/二叉树/三叉树深度级别与每个CU相关联。该复合深度示例性地如下计算。如果CU被二叉树分割,则在父和子CU之间复合深度增加3。如果CU被二叉树分割,则在父和子CU之间复合深度增加2。如果CU被三叉树分割,则在父和中间子CU之间,复合深度增加1,而在父和另外两个(非中间)子CU之间,复合深度增加3。在本实施例的范围内,双亮度/色度编码树与分离树之间的切换根据当前CU相对于复合深度阈值的复合深度级别来决定。

[0077] 根据实施例,用信号通知切换节点。有利地,该实施例允许例如基于RD成本为每个CTU适配切换节点。

[0078] 图8例示根据至少一个实施例的编码方法的另一示例。图8示出在编码器处的给定CTU的部分共用的亮度和色度编码树的确定步骤720的实施例。图8的实施例对应于其中基于分区块大小来确定切换节点的实施例。该决定处理涉及递归率失真优化过程,以搜索联合最小化CTU级别的总体率失真成本的两个亮度和色度编码树。为此,在递归处理的每个深度级别,设想候选分割模式的集合(或部分)用于考虑的编码单元(对应于处理开始时的整个CTU)。图8中示出两种主要情况,分别在图的左侧(分支为是)和右侧(分支为否)。

[0079] 左侧对应于当前CU具有严格大于解码流水线单元大小的情况。在那种情况下,根据至少一个实施例,必须选择共同在亮度和色度分量中使用的单个分割模式。为此,

当前CU的允许的候选分割模式的集合中的每个分割模式s或CurrentSplitMode被评估,这导致相关联的率失真成本 $RDcost\_YCbCr(CurrentSplitMode)$ 与所考虑的CU的最佳编码相关联,其可以在选择当前深度级别的分割模式s时执行。因此,分割模式s的评估涉及根据分割模式s的当前CU的分区,并且然后在每个结果子编码单元(sub-CU)上循环。对于每个子CU,以递归的方式调用当前的RDO处理,以便确定所考虑的子CU的分割模式。然后,这种在候选分割模式上的循环导致确定使与当前CU相关联的联合亮度/色度率失真成本最小化的分割模式。

[0080] 如果是128x128 CTU,则二叉树分割操作将128x128 CTU划分成4个64x64编码单元。这种划分要么作为共同的色度/亮度编码树的一部分来发信号通知,要么被推断出从而节省比特。注意,如果针对大于解码流水线单元大小64x64的CU推断二叉树分割模式,则图8的处理的左侧仅在于将输入CU分割为4个子CU,并且依次在每个作为结果的子CU上递归地应用图8的处理。

[0081] 关于图8的算法的右侧,它对应于整个RD搜索处理已到达编码树中的深度级别的情况,其对应于小于或等于所考虑解码流水线单元的大小的编码单元大小,即切换节点。在那种情况下,至少一个实施例提出了依据被优化的整个编码树表示中的考虑的深度级别,从联合亮度/色度编码树切换到分离的亮度和色度子编码树。因此,与当前CU相关联的亮度和色度编码树的确定采用两个分离的、连续的优化过程的形式,其分别找到当前CU的最佳亮度编码树和最佳色度编码树。

[0082] 这对应于图8之后的两个RDO步骤。优化步骤是本领域技术人员已知的。下一步包括计算整个联合亮度/色度率失真成本 $RDcost\_YCbCr(s\_luma, s\_chroma)$ ,其由获得的分离的亮度和色度子编码树得出。 $s\_luma$ 和 $s\_chroma$ 分别代表Luma (Y) 和Chroma (Cb+Cr) 分量中为当前CU找到的最佳分割模式。可以通过将亮度和色度分量的比率相加(得出总比率R),并通过将总失真D计算为两个分离的编码树搜索所产生的亮度和色度失真的总和,来简单地计算出该联合RD成本。然后根据经典公式获得总RD成本,其中 $\lambda$ 是已知的拉格朗日参数:

[0083]  $RDcostD + \lambda \cdot R$

[0084] 注意,在针对大于解码流水线单元大小64x64的CU推断二叉树分割模式的情况下,将率失真最佳分割模式 $s\_luma$ 和 $s\_chroma$ 分别分配给Luma和Chroma通道中的当前CU currCU。实际上,对于大于解码流水线单元大小的CU,分割模式之间不会发生速率失真竞争。

[0085] 来自图8的第一模块/操作(可能共用或不共用亮度/色度编码树)可以位于例如图13(稍后示出)的模块102(图像分区)中。

[0086] 3解码的实施例

[0087] 图9例示根据至少一个实施例的一般方面的解码方法900的示例。

[0088] 在预备步骤910中,访问编码块的比特流或信号用于解码。编码块典型是CU或CTU。解码方法900针对要解码的块递归地确定,分区或分割为子块。例如,根据图2,图3或图4描述的任何分割模式将编码块分割为子块。因此,在步骤920中,确定与该块的分区处理对应的分区树结构。有利地,分区树结构在亮度和色度块之间部分共用。例如,从解码处理以及从与编码块相关联的语法元素获得分区树结构。稍后将使用图10-13的变型实施例对此描述。在步骤930中,至少基于所确定的分区树结构来解码块。

[0089] 本领域技术人员将容易地将针对编码方法描述的变型实施例适配为解码方法900,因此在此不再重复。

[0090] 图10例示根据至少一个实施例的一般方面的解码方法的另一示例。图10示出在解码器处的给定CTU的部分共用的亮度和色度编码树的确定步骤920的实施例。图10的实施例对应于其中基于分区块大小来确定切换节点的实施例。对于编码,此处理也由两部分组成,分别在图10的左侧和右侧示出。图10的左侧例示在宽度或高度大小大于所考虑的解码流水线单元的CU的处理。图10的右侧例示小于或等于所考虑的解码流水线单元的大小的编码单元的解码。因此,在步骤921中,左侧暗示分割模式的递归解析,并且如果CU的大小大于解码单元的大小,则可以进行CU的解码。在图11上描绘这种大CU的解码921。根据变型,四叉树分割操作将CU划分为等于DPU大小的子块(例如,在128x128 CTU的情况下,四叉树分割操作将128x128 CTU划分为4个64x64编码单元)。

[0091] 根据另一变型,关于编码器侧,对于大于DPU大小的CU,可以推断四叉树分割模式。

[0092] 关于图10的右侧,它包括对亮度分量然后对色度分量的依次且独立的解析、解码和重构包含在所考虑的画面区域中的编码单元。

[0093] 来自图10和图11的模块/操作可以例如位于图14(稍后示出)的模块230(包括语法元素的解析的熵解码)和235(分区)之间。

[0094] 4. 解码的替代实施例

[0095] 本节例示节3的解码处理的替代实施例。这里的区别在于,块解码处理在解码流水线的基础上执行,以便显式符合对于解码处理考虑的约束条件。实际上,与本原理的实施例兼容的解码器能够有利地在开始处理下一个解码单元之前执行用于亮度和色度分量的解码单元的所有解码处理。

[0096] 图12示出根据本发明的实施例的用于解码和重构亮度和色度块的基于解码单元的处理。该处理首先在空间上将CTU划分为大小等于解码流水线单元的解码单元,典型为64x64。

[0097] 例如,该划分通过将128x128 CTU划分为4个64x64编码单元的四叉树分割操作执行。

[0098] 该处理涉及所考虑的CTU中包含的所有解码单元(典型大小为64x64)的循环。对于每个解码单元,首先解码包含在解码单元中的亮度块,然后解码并重构包含在解码单元中的色度块。

[0099] 5. 其他实施例和信息

[0100] 本节描述各种方面,包括工具,特征,实施例,模型,方法等。这些方面中的许多方面被专门描述,并且至少为了示出各个特性,被以听起来可能受到限制的方式描述。然而,这是为了描述的清楚,并且不限制那些方面的应用或范围。实际上,所有不同方面都可以组合和互换以提供进一步的方面。此外,这些方面也可以与先前申请中描述的方面组合和互换。

[0101] 本文中描述和设想的方面可以以许多不同的形式实现。下面的图13,图14和图15提供了一些实施例,但是可以设想其他实施例,并且图13、14和15的讨论不限制实施方式的范围。这些方面中的至少一个方面一般涉及视频编码和解码,并且至少另一个方面一般涉及传输所生成或编码的比特流。这些和其他方面可以实现为方法,装置,其上存储有用于根

据所描述的任何方法编码或解码视频数据的指令的计算机可读存储介质,和/或其上存储有根据所描述的任何方法生成的比特流的计算机可读存储介质。

[0102] 在本申请中,术语“重构”和“解码”可以互换使用,术语“像素”和“样本”可以互换使用,术语“图像”,“画面”和“帧”可以互换使用。通常但并非必须,术语“重构”在编码器侧使用,而“解码”在解码器侧使用。

[0103] 以上描述了各种方法,并且每个方法包括用于实现所描述的方法的一个或多个步骤或动作。除非方法的适当操作需要特定的步骤或动作顺序,否则可以修改或组合特定步骤和/或动作的顺序和/或使用。

[0104] 本文中描述的各种方法和其他方面可以用于修改模块,诸如例如,图13和图14示出的JVET或HEVC编码器100和解码器200的分区模块(102,235)和熵解码模块(230)。此外,本方面不限于JVET或HEVC,并且可以应用于例如其他标准和推荐,无论是预先存在的还是将来开发的,以及任何此类标准和推荐的扩展(包括JVET和HEVC)。除非另有说明或技术上排除,否则本文中描述的各个方面可以单独使用或组合使用。

[0105] 在本文档中使用各种数值,例如64x64的解码单元流水线或块分区(CU,TU)大小。特定值是出于示例性目的,并且所描述的方面不限于这些特定值。

[0106] 图13例示示例性编码器100。可以设想该编码器100的变型,但是为了清楚起见,下面描述编码器100,而没有描述所有预期的变型。

[0107] 在被编码之前,视频序列可以经过预编码处理(101),例如,对输入的彩色画面应用颜色变换(例如,从RGB 4:4:4到YCbCr 4:2:0的转换),或者执行输入画面分量的重新映射,以便获得对压缩更有弹性的信号分布(例如,使用颜色分量之一的直方图均衡化)。元数据可以与预处理相关联,并附加到比特流。

[0108] 在示例性编码器100中,画面由编码器元件编码,如下所述。要编码的画面例如以CU为单元分区(102)并处理。例如使用帧内或帧间模式编码每个单元。当单元以帧内模式编码时,其执行帧内预测(160)。在帧间模式中,执行运动估计(175)和补偿(170)。编码器决定(105)帧内模式或帧间模式中的哪一个用于编码该单元,并且例如通过预测模式标志指示帧内/帧间决定。通过从原始图像块中减去(110)预测块来计算预测残差。

[0109] 然后预测残差被变换(125)和量化(130)。熵编码(145)量化的变换系数以及运动矢量和语法元素,以输出比特流。编码器可以跳过该变换,并且将量化直接应用于未变换的残差信号。编码器还可以绕过变换和量化两者,即,在不应用变换或量化处理的情况下直接编码残差。

[0110] 编码器解码编码块,以为进一步的预测提供参考。量化的变换系数被去量化(140)并且逆变换(150)以解码预测残差。组合(155)解码的预测残差和预测块,重构图像块。环内滤波器(165)被应用于重构的画面,以例如执行去块/SAO(样本自适应偏移)滤波以减少编码伪像。滤波图像被存储在参考画面缓冲器(180)中。

[0111] 图14例示示例性视频解码器200的框图。在示例性解码器200中,由解码器元件解码比特流,如下面描述。视频解码器200一般执行与图13中描述的编码通道相反的解码通道。编码器100也一般执行视频解码作为编码视频数据的一部分。

[0112] 具体地,解码器的输入包括可由视频编码器100生成的视频比特流。首先熵解码(230)该比特流,以获得变换系数,运动矢量,和其他编码信息。画面分区信息指示画面如何

被分区。因此,解码器可以根据解码的画面分区信息来划分(235)画面。去量化(240)和逆变换(250)变换系数以解码预测残差。组合(255)解码的预测残差和预测块,重构图像块。可以从帧内预测(260)或运动补偿预测(即,帧间预测)(275)获得(270)预测块。将环内滤波器(265)应用于重构的图像。滤波的图像存储在参考画面缓冲器(280)处。

[0113] 解码的画面可以进一步经历后解码处理(285),例如,逆颜色变换(例如,从YCbCr 4:2:0到RGB 4:4:4的转换)或逆重新映射,其对预编码处理(101)中执行的重新映射处理执行逆过程。后解码处理可以使用在预编码处理中推导出并在比特流中用信号通知的元数据。

[0114] 图15例示其中实现各个方面和示例性实施例的示例系统的框图。系统1000可以体现为包括下面描述的各种组件的设备,并且被配置为执行本文件描述的一个或多个方面。这样的设备的示例包括但不限于个人计算机,膝上型计算机,智能电话,平板计算机,数字多媒体机顶盒,数字电视接收器,个人视频记录系统,连接的家用电器和服务器。系统1000可以通信地耦合到其他类似的系统,并且经由通信信道耦合到显示器,如图15所示,并且如本领域技术人员所知,实现本文件中描述的各个方面。。

[0115] 系统1000包括至少一个处理器1010,其被配置为执行加载在其中的指令,用于实现例如本文件描述的各种方面。处理器1010可以包括嵌入式存储器,输入输出接口和本领域已知的各种其他电路。系统1000还包括至少一个存储器1020(例如,易失性存储器设备,非易失性存储器设备)。系统1000包括存储设备1040,其可以包括非易失性存储器,包括但不限于EEPROM,ROM,PROM,RAM,DRAM,SRAM,闪存,磁盘驱动器和/或光盘驱动器。作为非限制性示例,存储设备1040可以包括内部存储设备,外接的存储设备和/或网络可存取存储设备。系统1000还可以包括编码器/解码器模块1030,其被配置为处理数据以提供编码的视频或解码的视频,并且编码器/解码器模块1030可以包括其本身的处理器和存储器。

[0116] 编码器/解码器模块1030表示可以包括在设备中以执行编码和/或解码功能的(多个)模块。如所知,设备可以包括编码和解码模块中的一个或两个。另外,如本领域技术人员已知的,编码器/解码器模块1030可以实现为系统1000的单独元件,或者可以并入处理器1010内作为本领域技术人员已知的硬件和软件的组合。

[0117] 要加载到处理器1010上以执行本文件描述的各种方面的程序代码可以存储在存储设备1040中,并且随后加载到存储器1020上用于由处理器1010执行。根据示例性实施例,(多个)处理器1010,存储器1020,存储设备1040和编码器/解码器模块1030中的一个或多个可以在执行本文件描述的处理期间存储各种项目中的一个或多个,包括但不限于输入视频,解码视频,比特流,等式,公式,矩阵,变量,操作以及操作逻辑。

[0118] 系统1000包括使得能够经由通信信道1060与其他设备通信的通信接口1050。通信接口1050可以包括但不限于配置为在通信信道1060上发送和接收数据的收发器。通信接口1050可以包括但不限于调制解调器或网卡,并且通信信道1060可以例如在有线和/或无线介质内实现。系统1000的各种组件可以使用各种合适连接(包括但不限于内部总线,电线和印刷电路板)全部连接或者通信耦合在一起。

[0119] 可以通过由处理器1010实现的计算机软件,或者通过硬件,或者通过硬件和软件的组合来执行示例性实施例。作为非限制性示例,示例性实施例可以由一个或多个集成电路实现。作为非限制性示例,存储器1020可以是适合技术环境的任何类型,并且可以使用任

何适当的数据存储技术来实现,诸如,光存储设备,磁存储设备,基于半导体的存储设备,固定存储器和可移动存储器。作为非限制性示例,处理器1010可以是适合于技术环境的任何类型,并且可以包括微处理器,通用计算机,专用计算机和基于多核架构的处理器中的一个或多个。

[0120] 本文描述的实现方式和方面可以在例如方法或处理,装置,软件程序,数据流或信号中实现。即使仅在单个实现形式的上下文中讨论(例如,仅作为方法讨论),讨论的特征的实现方式也可以以其他形式(例如,装置或程序)来实现。装置可以在例如适当的硬件,软件和固件中实现。方法例如可以在例如装置(例如,处理器)中实现,该处理器一般指代处理设备,包括例如计算机,微处理器,集成电路或可编程逻辑设备。处理器也包括通信设备,例如计算机,蜂窝电话,便携/个人数字助理(“PDA”),以及便于终端用户之间的信息通信的其他设备。

[0121] 对“一个实施例”或“实施例”或“一个实现方式”或“实现方式”的引用以及其他变型意味着结合实施例描述的具体特征,结构,特性等包括在至少一个实施例中。因此,在整个文件中出现在各个地方的短语“在一个实施例中”或“在实施例中”或“在一个实现方式中”或“在实现方式中”以及任何其他变型的出现不一定都指代同一个实施例。

[0122] 另外,本文件可以指“确定”各种信息。确定信息可以包括例如估计信息,计算信息,预测信息或从存储器检索信息中的一个或多个。

[0123] 此外,本文件可以指“访问”各种信息。访问信息可以包括例如接收信息,检索信息(例如,从存储器中),存储信息,处理信息,传送信息,移动信息,复制信息,擦除信息,计算信息,确定信息,预测信息或估计信息中的一个或多个。

[0124] 另外,本文件可以指“接收”各种信息。与“访问”一样,接收意图是广义术语。接收信息可以包括例如访问信息或检索信息(例如,从存储器中)中的一个或多个。此外,“接收”典型在操作期间以一种方式或其他方式,涉及例如,存储信息,处理信息,传送信息,移动信息,复制信息,擦除信息,计算信息,确定信息,预测信息或估计信息。

[0125] 另外,该文档可以涉及优化,并且各种实施例不需要被完全优化,一些实施例简单地选择模式而没有通常意义上的优化。

[0126] 将对于本领域技术人员显然的,实现方式可以产生被格式化以携带例如可以存储或传送的信息的各种信号。该信息可以包括例如用于执行方法的指令或由描述的实施方式之一产生数据。例如,信号可以被格式化以承载描述的实施例的比特流。这样的信号可以被格式化为例如电磁波(例如,使用频谱的射频部分)或者作为基带信号。格式化可以包括,例如编码数据流和用编码数据流调制载波。信号携带的信息可以是例如模拟或数字信息。如已知,信号可以通过各种不同的有线或无线链路传送。信号可以存储在处理器可读介质上。

[0127] 前面已经描述了多个实施例。这些实施例提供至少以下通用的发明和权利要求,包括跨越各种不同的权利要求类别和类型的所有组合:

[0128] • 修改在解码器和/或编码器中应用的亮度和色度编码树。

[0129] • 组合在解码器和/或编码器中应用的交织亮度/色度编码和分离的亮度/

[0130] 色度编码。

[0131] • 组合交织的亮度/色度编码和分离的亮度/色度编码,其中,分离的亮度

[0132] /色度编码由与在解码器和/或编码器中应用的对解码单元流水线的约束有关的

信息控制。

[0133] • 从最大允许块大小 (CTU大小) 向下到给定块大小或向上到给定CU深度级别, 联合决定、编码和发信号通知亮度和色度编码树, 并独立决定编码和发信号通知亮度和色度编码树用于大小小于给定块大小的块或超过给定CU深度级别的块。

[0134] • 根据与处理的编码树节点相关联的块大小, 或根据要编码/解码的编码树节点的深度级别, 从交织的亮度/色度编码切换到分离的亮度/色度编码。

[0135] • 在解码器和/或编码器中启用若干种部分共用的亮度和色度编码树方法。

[0136] • 基于某些语法元素, 选择要在解码器上应用的亮度和色度编码树。

[0137] • 创建和/或发送和/或接收和/或解码包括一个或多个描述的编码亮度

[0138] /色度分量或其变型的比特流或信号。

[0139] • 电视, 机顶盒, 手机, 平板电脑或其他电子设备, 根据描述的任何实施例执行启用部分共用的亮度和色度编码树方法的编码/解码。

[0140] • 电视, 机顶盒, 手机, 平板电脑或其他电子设备, 根据描述的任何实施例执行启用部分共用的亮度和色度编码树方法的编码和解码, 并显示 (例如, 使用监视器, 屏幕或其他类型的显示) 作为结果的图像。

[0141] • 电视, 机顶盒, 手机, 平板电脑或其他电子设备, 根据描述的任何实施例, 调谐 (例如, 使用调谐器) 信道以接收包括编码图像的信号,

[0142] 并执行启用部分共用的亮度和色度编码树的解码。

[0143] • 电视, 机顶盒, 手机, 平板电脑或其他电子设备, 根据描述的任何实施例, 通过无线方式接收 (例如, 使用天线) 包括编码图像的信号,

[0144] 并执行启用部分共用的亮度和色度编码树方法的解码。

[0145] 在整个本公开中, 还支持和构想了各种其他广义的和具体化的发明和权利要求。

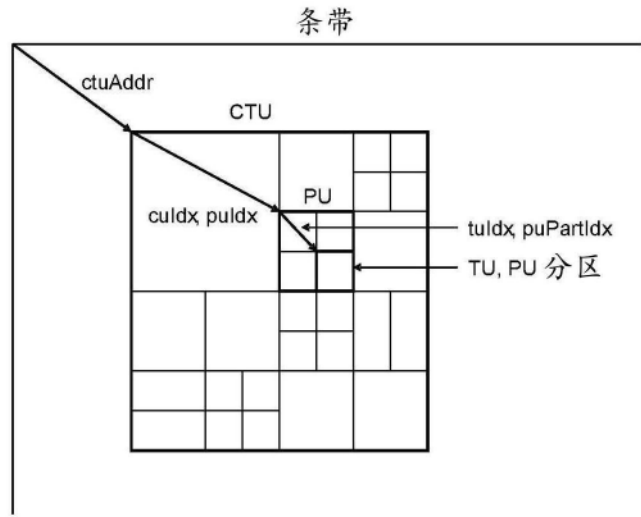


图1

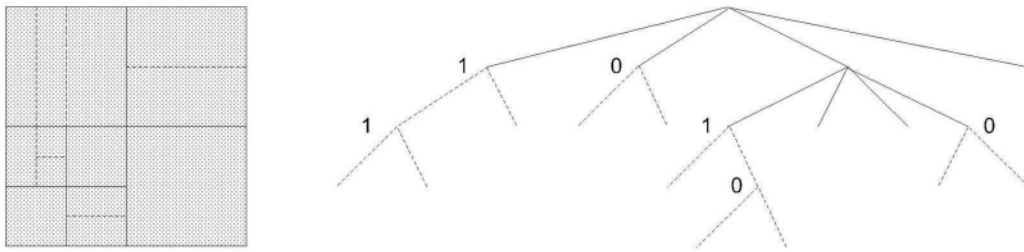


图2

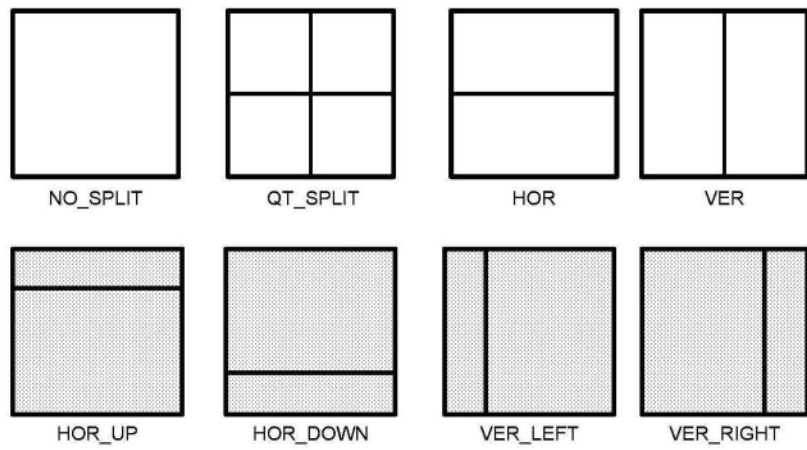


图3

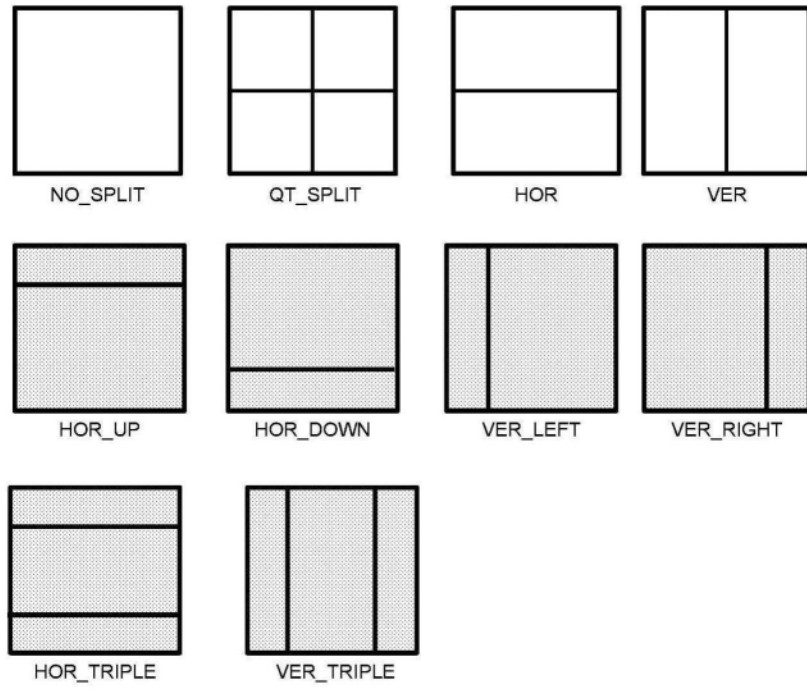


图4

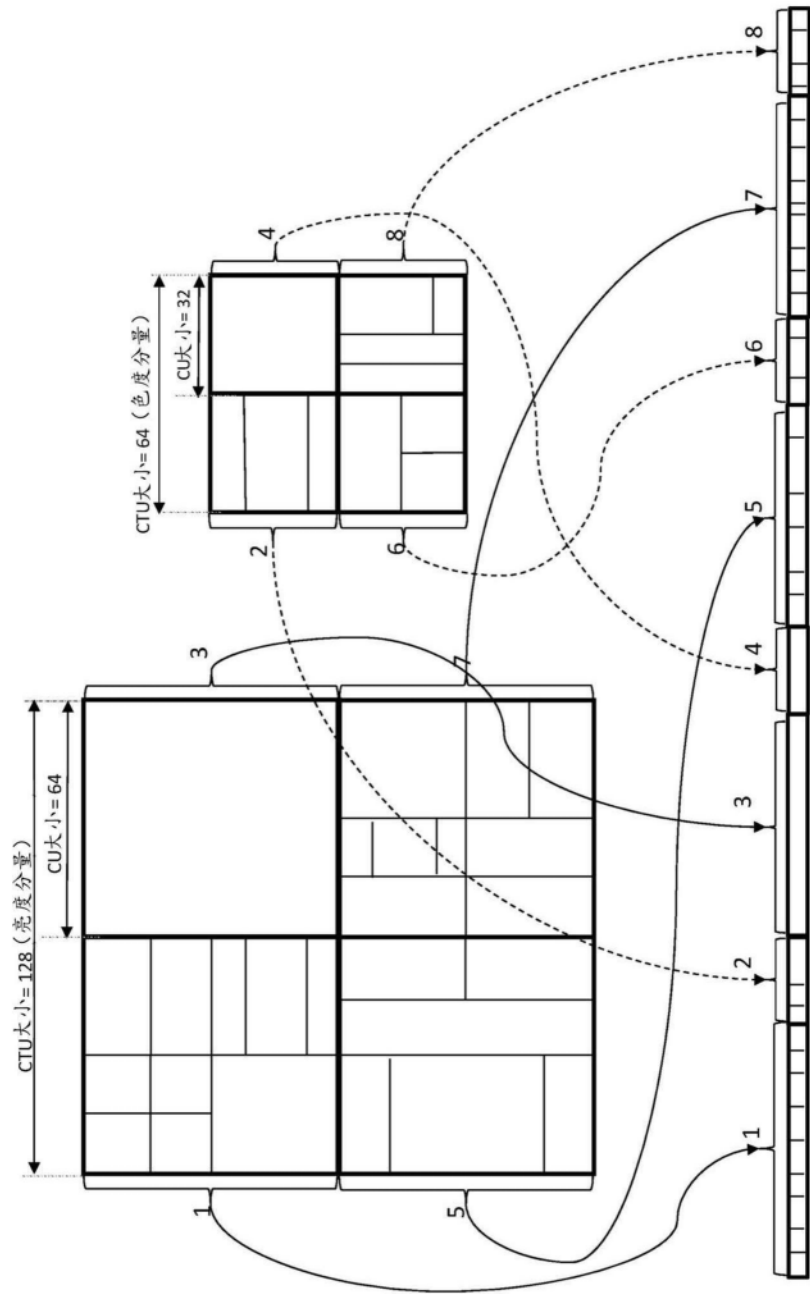


图5

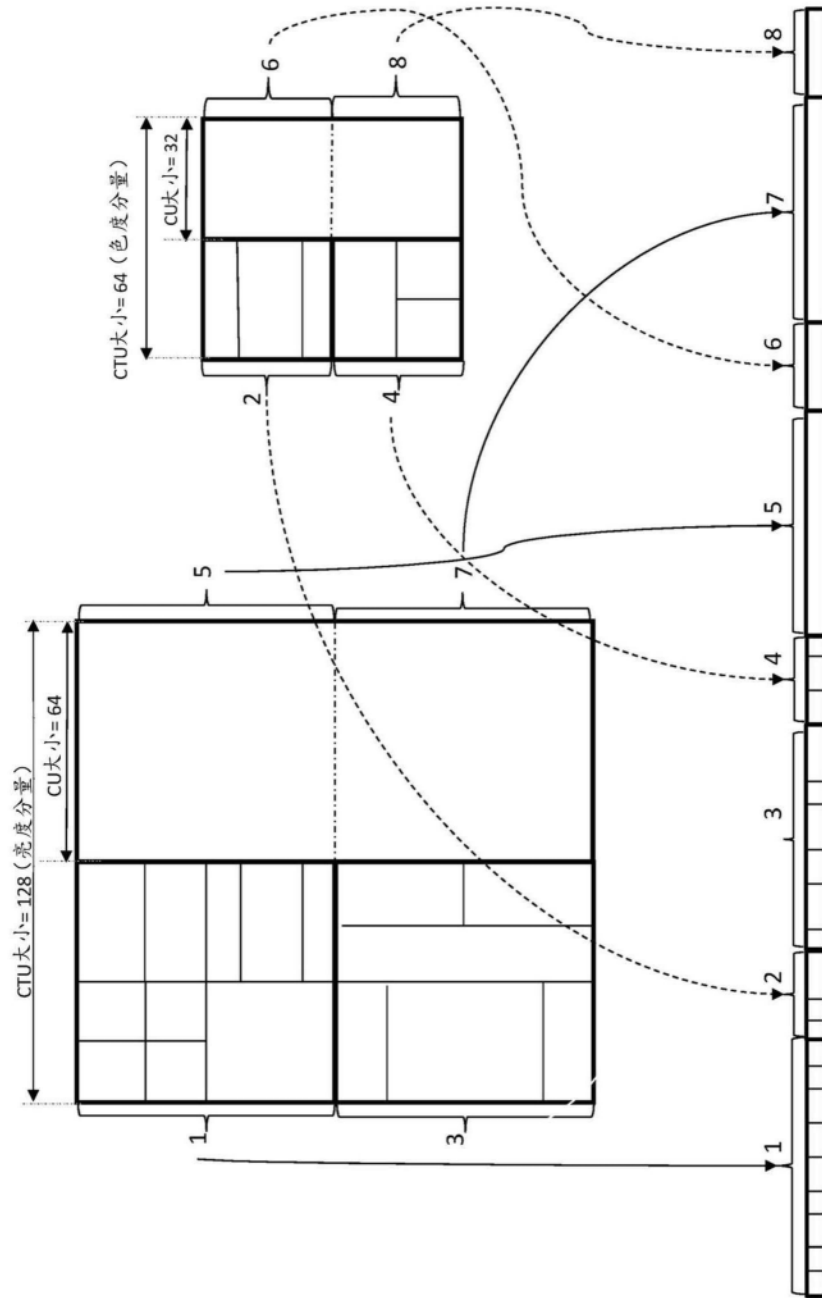


图6

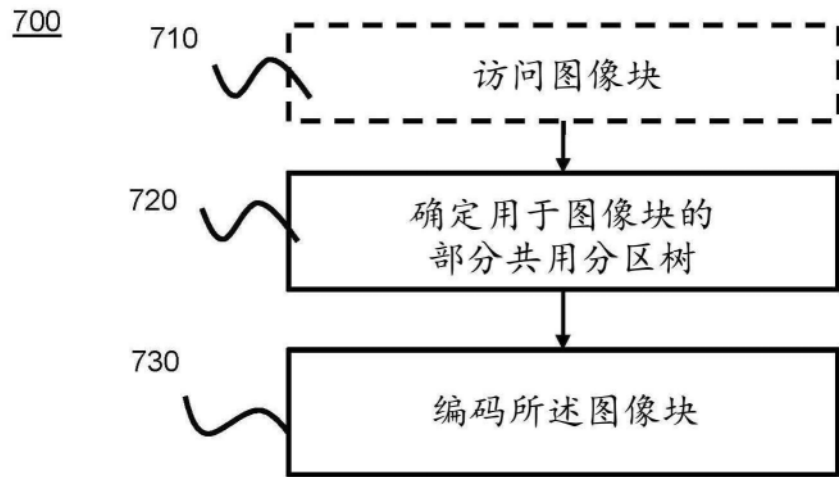


图7

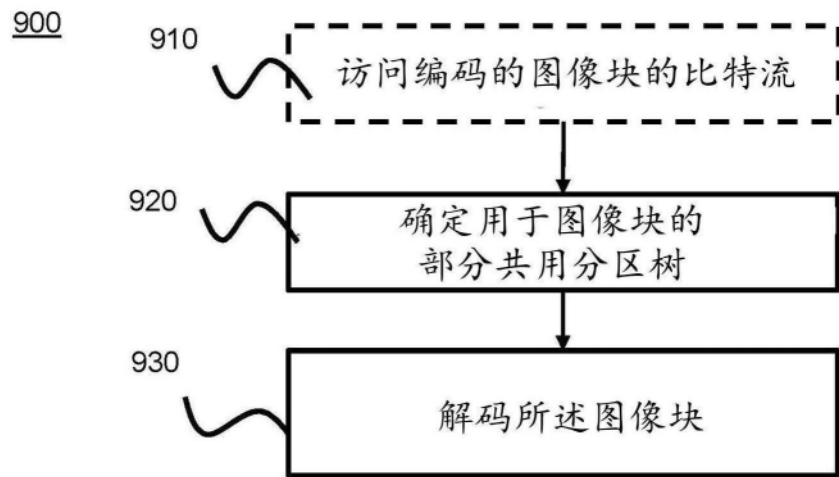


图9

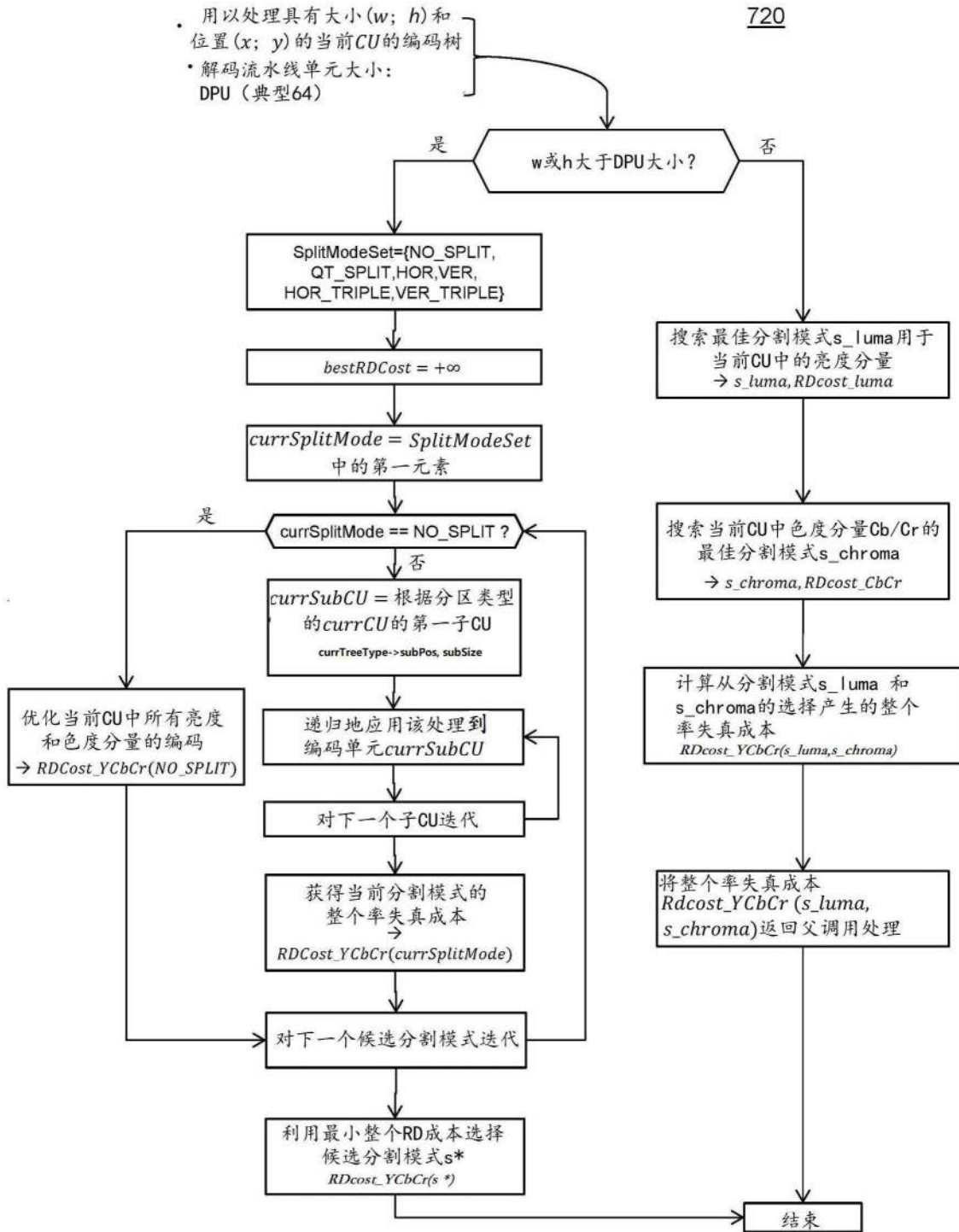


图8

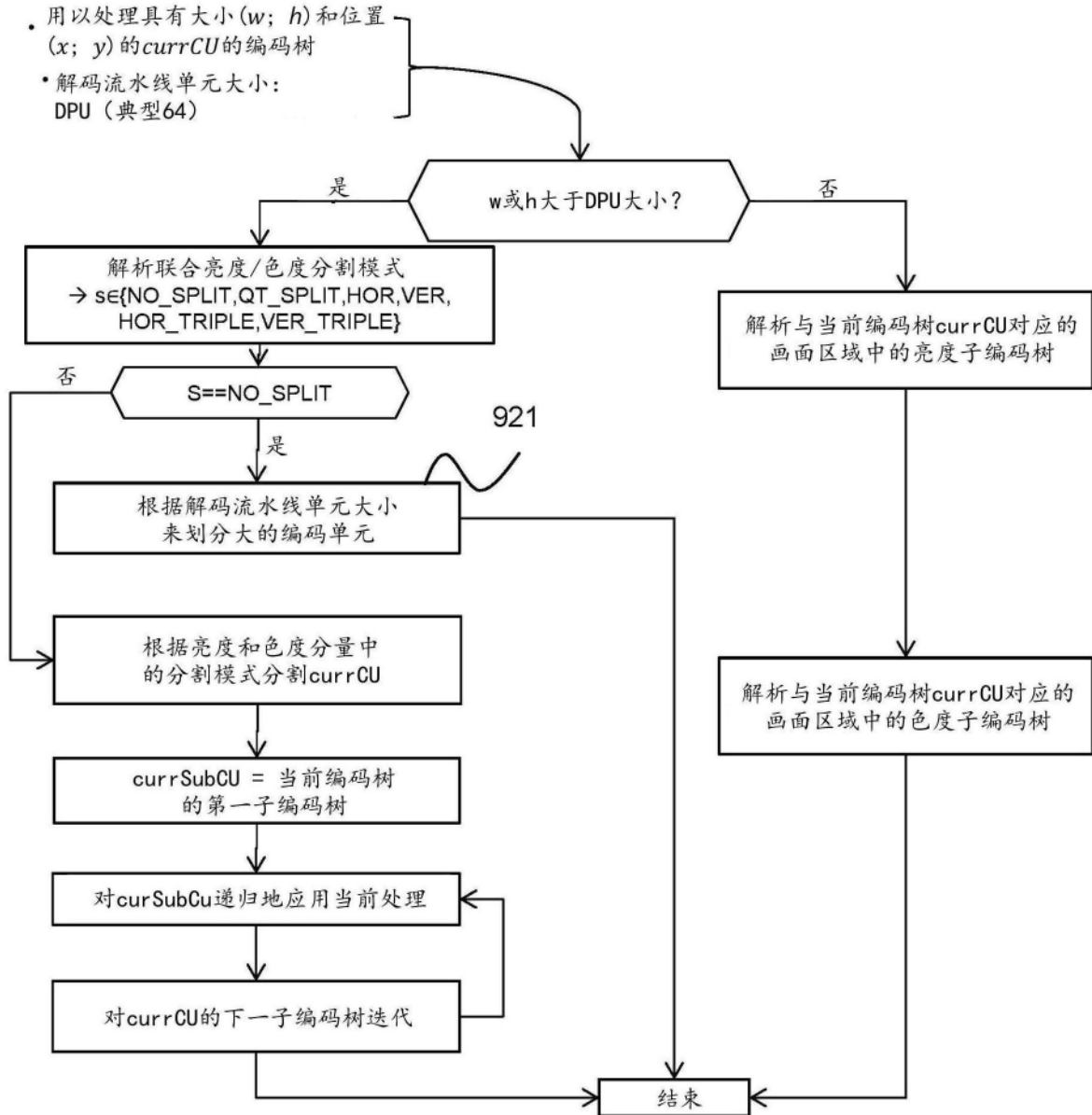


图10

- 用以解码具有大小 ( $w$ ;  $h$ ) 和位置 ( $x$ ;  $y$ ) 的  $currCU$  的大的 CU
- 解码流水线单元大小: DPU (典型 64)
- 与当前 CU 相关联的帧内预测模式

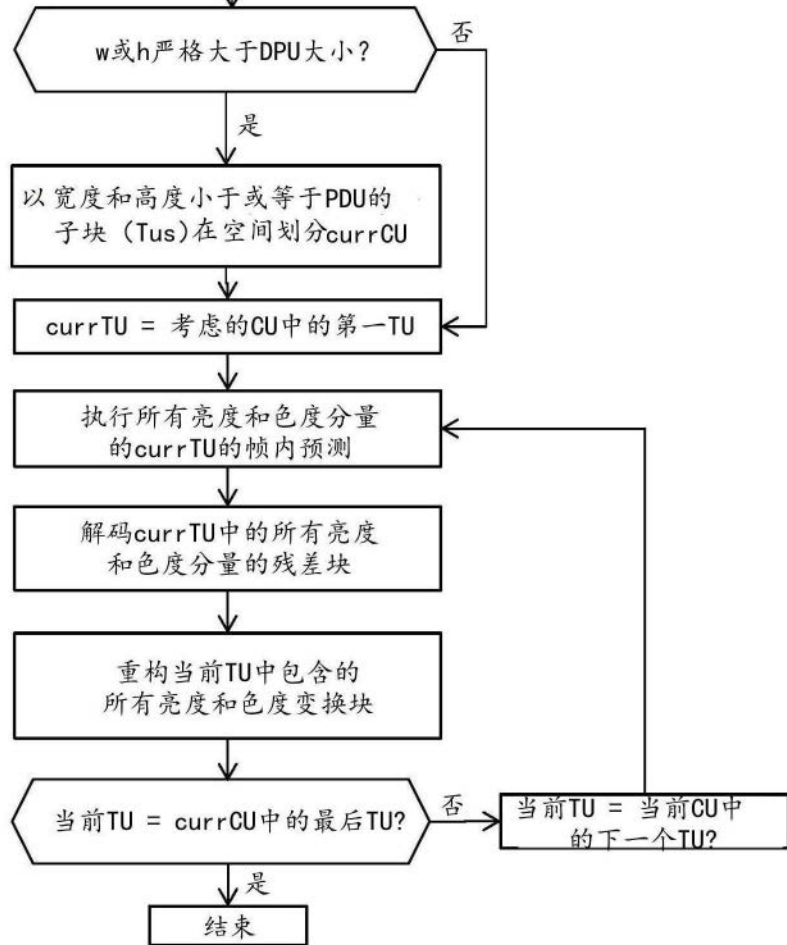


图11

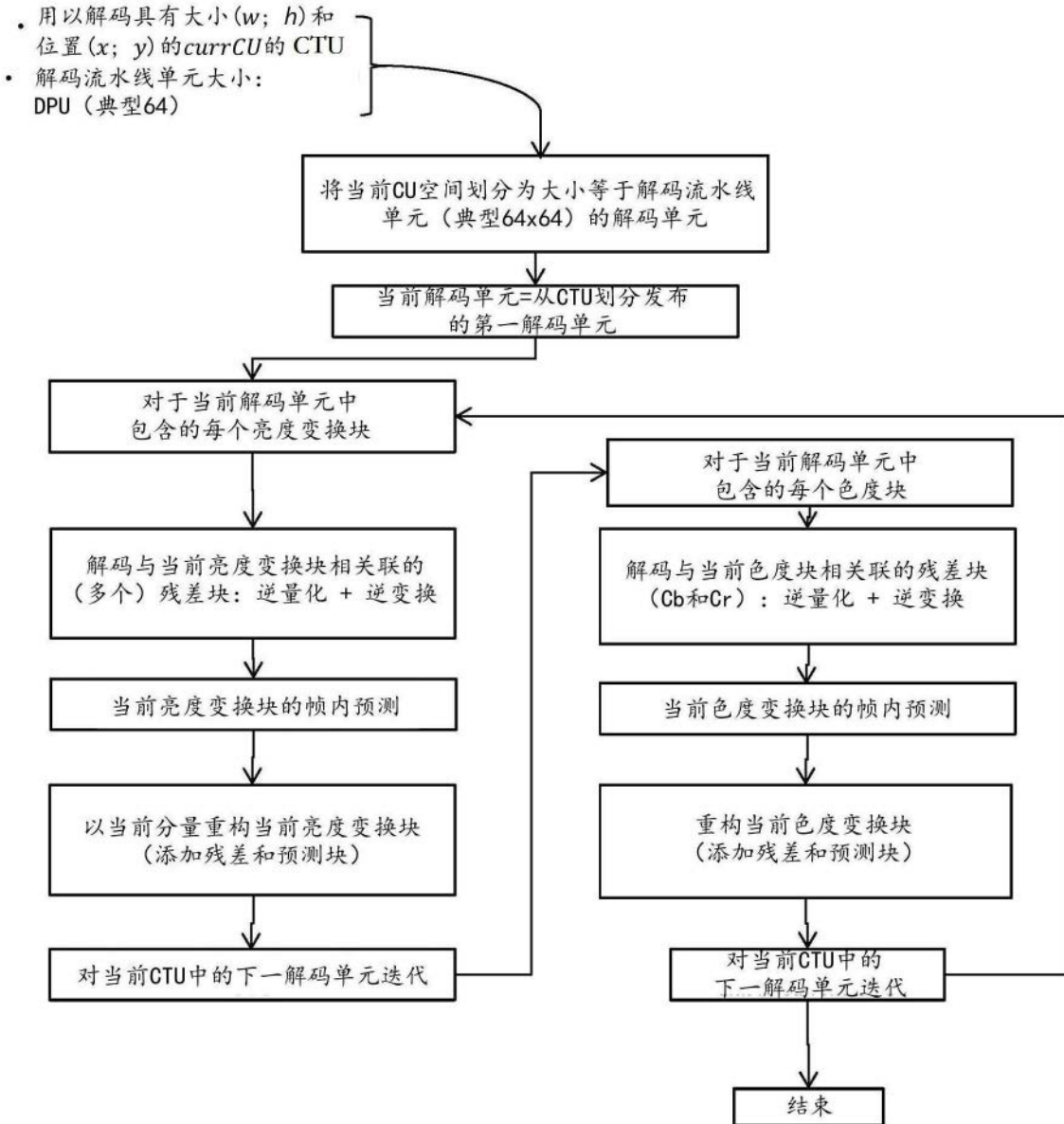


图12

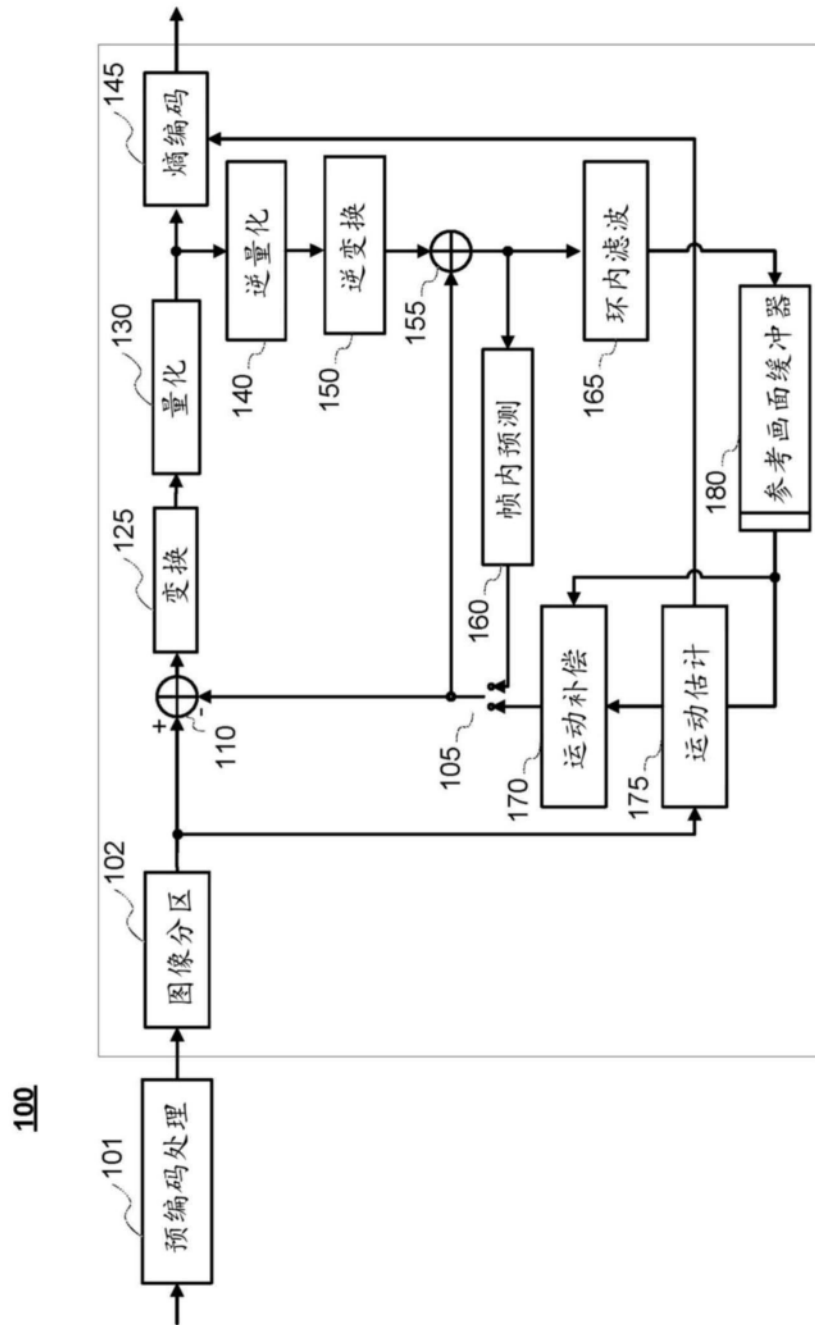


图13

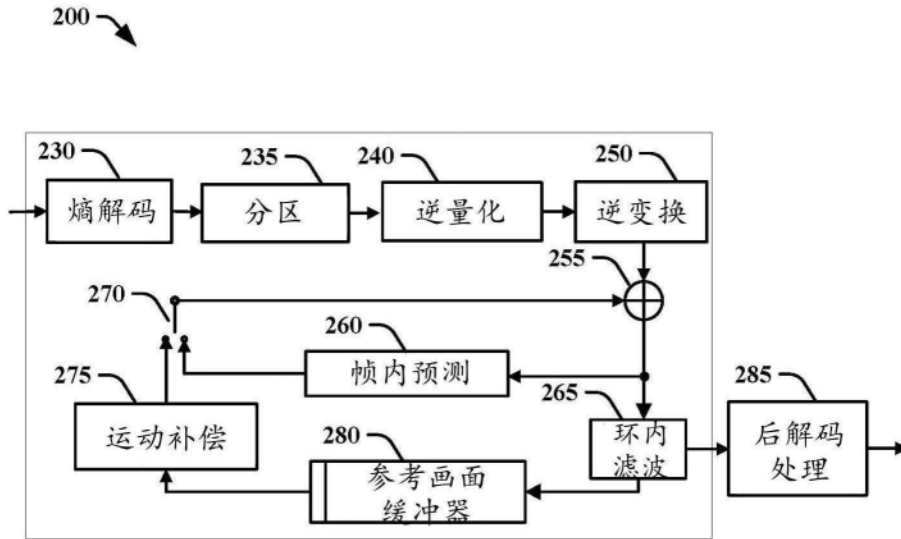


图14

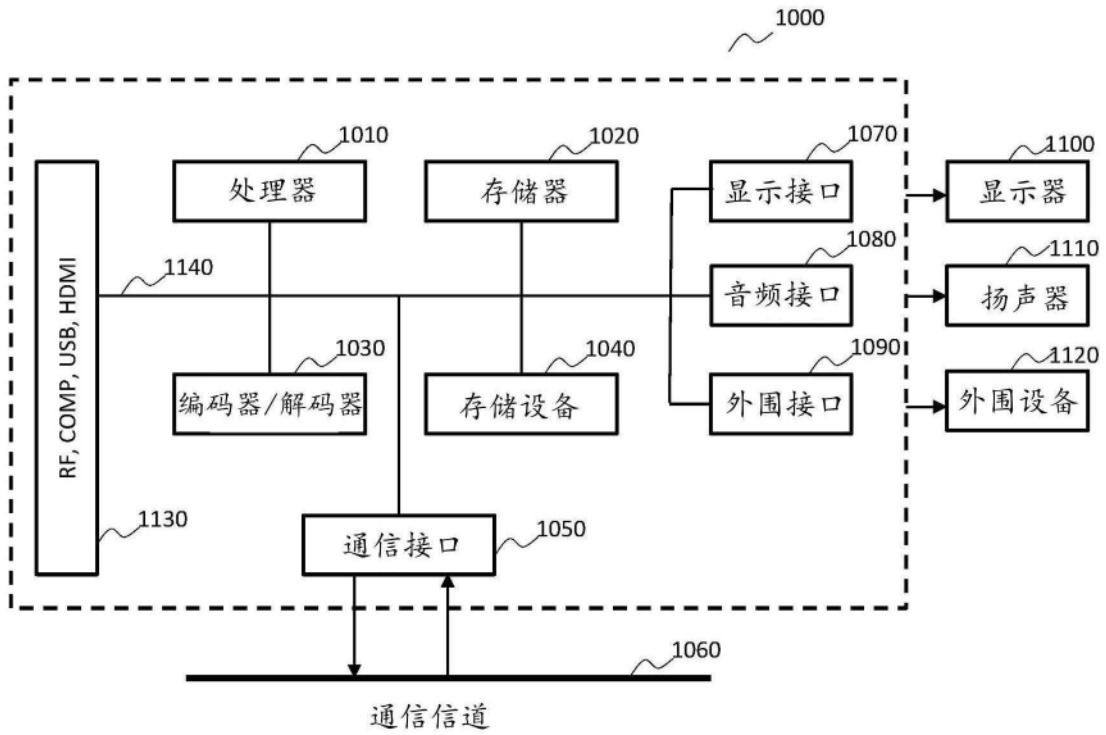


图15