



(12)发明专利申请

(10)申请公布号 CN 106454352 A

(43)申请公布日 2017.02.22

(21)申请号 201611042229.7 *H04N 19/176*(2014.01)

(22)申请日 2011.04.05 *H04N 19/119*(2014.01)

(30)优先权数据 *H04N 19/46*(2014.01)

10-2010-0102506 2010.10.20 KR *H04N 19/122*(2014.01)

61/320,826 2010.04.05 US *H04N 19/96*(2014.01)

(62)分案原申请数据
201180027746.6 2011.04.05

(71)申请人 三星电子株式会社
地址 韩国京畿道水原市

(72)发明人 崔雄一 韩宇镇 李泰美 李善一

(74)专利代理机构 北京铭硕知识产权代理有限公司 11286

代理人 曾世骅 孔敏

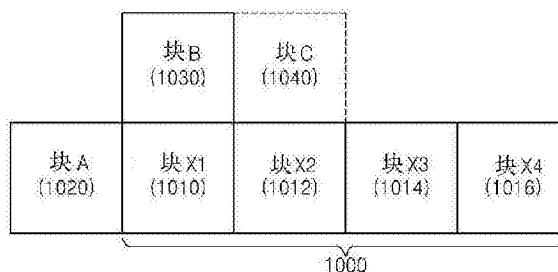
(51)Int.Cl.
H04N 19/159(2014.01)

权利要求书1页 说明书13页 附图9页

(54)发明名称
低复杂度熵编码/解码方法和设备

(57)摘要

提供了一种低复杂度熵编码/解码方法和设备。通过基于先前编码或解码的块的上下文模型对图像数据的块的上下文单元执行上下文建模，分别执行图像数据的熵编码和熵解码。



1. 一种对图像数据进行熵解码的方法,所述方法包括:
通过使用第一概率信息来对在层内的第一块的相邻块的语法元素进行熵解码;
通过更新第一概率信息来获得第二概率信息;
基于第二概率信息对在所述层内的第一位置处的第一块的语法元素和在所述层内的第二位置处的第二块的语法元素进行熵解码;
其中,第一块的所述相邻块在第一块和第二块之前被解码,
其中,第一块的在所述层内的第一位置和第二块的在所述层内的第二位置是在所述层内彼此空间分离的位置,
其中,所述层是图像中所包括的多个层之一,
其中,第二概率信息是通过更新包括在第一概率信息中的指示最可能符号的信息和指示所述最可能符号的发生概率的信息中的至少一个而获得的。

低复杂度熵编码/解码方法和设备

[0001] 本申请是申请日为2011年4月5日,申请号为“201180027746.6”,标题为“低复杂度熵编码/解码方法和设备”的发明专利申请的分案申请。

技术领域

[0002] 示例性实施例涉及一种熵编码和熵解码方法和设备,更具体地讲,涉及一种用于对通过对图像进行编码而产生的预定的语法元素进行熵编码和熵解码的方法和设备。

背景技术

[0003] 一种编解码器(诸如,运动图像专家组(MPEG)-4H.264/MPEG-4高级视频编码(AVC))通过使用基于上下文的自适应可变长编码(CAVLC)和基于上下文的自适应二进制算术编码(CABAC)对语法元素进行熵编码/解码。然而,在以上描述的基于上下文的自适应熵编码/解码中,将会频繁地执行存储器访问。

发明内容

[0004] 技术方案

[0005] 示例性实施例包括有效地执行熵编码/解码的方法和设备、以及记录有用于执行所述方法的计算机程序的计算机可读记录介质。

[0006] 有益效果

[0007] 如上所述,可通过使用低复杂度硬件以高的速度对图像数据进行熵编码或熵解码,而在编码期间图像的压缩率不会显著地降低。

附图说明

[0008] 通过参考附图详细描述以上和其它方面的示例性实施例,以上和其它方面将会变得更加清楚,在附图中:

[0009] 图1是根据示例性实施例的对图像进行编码的设备的框图;

[0010] 图2是根据示例性实施例的对图像进行解码的设备的框图;

[0011] 图3示出根据示例性实施例的分层编码单元;

[0012] 图4是根据示例性实施例的基于编码单元的图像编码器的框图;

[0013] 图5是根据示例性实施例的基于编码单元的图像解码器的框图;

[0014] 图6示出根据示例性实施例的最大编码单元、子编码单元和预测单元;

[0015] 图7示出根据示例性实施例的编码单元和变换单元;

[0016] 图8A到图8D示出根据示例性实施例的编码单元、预测单元和变换单元的划分形状;

[0017] 图9是根据示例性实施例的熵编码设备的框图;

[0018] 图10示出根据示例性实施例的共享上下文模型的方法;

[0019] 图11示出根据示例性实施例的共享上下文模型的方法;

[0020] 图12是根据示例性实施例的熵解码设备的框图；

[0021] 图13是示出根据示例性实施例的熵编码方法的流程图；

[0022] 图14是示出根据示例性实施例的熵解码方法的流程图。

[0023] 最佳模式

[0024] 根据示例性实施例的一方面，提供一种对图像数据进行熵编码的方法，所述方法包括：将用于对图像数据的第一块的语法元素进行熵编码的当前上下文模型以及用于对图像数据的与第一块相邻的第二块的语法元素进行熵编码的第二上下文模型设置为与用于对与第一块相邻的先前编码的块的语法元素进行熵编码的先前上下文模型相同；基于所述当前上下文模型对第一块的语法元素和第二块的语法元素进行熵编码。

[0025] 所述方法还可包括：将第一块和第二块组合为上下文单元，并且第一块是所述上下文单元中的在第二块之前被编码的块。

[0026] 设置当前上下文模型的步骤可包括：将用于对第一块的语法元素进行熵编码的上下文索引设置为与用于对先前编码的块的语法元素进行熵编码的上下文索引相同。

[0027] 对第一块的语法元素进行熵编码的步骤可包括：基于所述上下文索引对第一块的语法元素执行基于上下文的自适应二进制算术编码。

[0028] 对第一块的语法元素进行熵编码的步骤可包括：基于指示与所述上下文索引相应的最可能的符号的信息以及指示最可能的符号的发生概率的信息对第一块的语法元素执行算术编码。

[0029] 对第一块的语法元素进行熵编码的步骤还可包括：基于算术编码的结果，更新指示最可能的符号的信息以及指示最可能的符号的发生概率的信息中的至少一个。

[0030] 对第一块的语法元素进行熵编码的步骤还可包括：基于所述上下文索引对第一块的语法元素执行基于上下文的自适应可变长编码 (CAVLC)。

[0031] 根据示例性实施例的另一方面，提供一种对图像数据进行熵解码的方法，所述方法包括：将用于对图像数据的第一块的语法元素进行熵解码的当前上下文模型以及用于对图像数据的与第一块相邻的第二块的语法元素进行熵解码的第二上下文模型设置为与用于对与第一块相邻的先前解码的块的语法元素进行熵解码的先前上下文模型相同；基于所述当前上下文模型对第一块的语法元素和第二块的语法元素进行熵解码。

[0032] 根据示例性实施例的另一方面，提供一种用于对图像数据进行熵编码的设备，所述设备包括：上下文建模单元，将用于对图像数据的第一块的语法元素进行熵编码的当前上下文模型以及用于对图像数据的与第一块相邻的第二块的语法元素进行熵编码的第二上下文模型设置为与用于对与第一块相邻的先前编码的块的语法元素进行熵编码的先前上下文模型相同；熵编码器，基于所述当前上下文模型对第一块的语法元素和第二块的语法元素进行熵编码。

[0033] 根据示例性实施例的另一方面，提供一种用于对图像数据进行熵解码的设备，所述设备包括：上下文建模单元，将用于对图像数据的第一块的语法元素进行熵解码的当前上下文模型以及用于对图像数据的与第一块相邻的第二块的语法元素进行熵解码的第二上下文模型设置为与用于对与第一块相邻的先前解码的块的语法元素进行熵解码的先前上下文模型相同；熵解码器，基于所述当前上下文模型对第一块的语法元素和第二块的语法元素进行熵解码。

[0034] 根据示例性实施例的另一方面,提供一种计算机可读记录介质,所述计算机可读记录介质上具有用于执行对图像数据进行熵编码的方法或对图像数据进行熵解码的方法的计算机程序。

具体实施方式

[0035] 以下,将参照附图更全面地描述一个或多个示例性实施例。当诸如“…中的至少一个”的表述位于一系列元素之后时,其修饰整个一系列元素,而不是修饰列出的单个元素。在本说明书中,“图像”可表示视频的静止图像或运动图像(即,视频本身)。

[0036] 图1是根据示例性实施例的对图像进行编码的设备100的框图。

[0037] 参照图1,对图像进行编码的设备100包括最大编码单元划分器110、编码深度确定器120、图像数据编码器130和编码信息编码器140。

[0038] 最大编码单元划分器110可基于最大编码单元(即,当前帧或当前条带的编码单元中具有最大尺寸的编码单元)划分当前帧或当前条带。也就是说,最大编码单元划分器110可将当前帧或当前条带划分为至少一个最大编码单元。

[0039] 根据示例性实施例,可使用最大编码单元和深度表示编码单元。如上所述,最大编码单元表示当前帧的编码单元中具有最大尺寸的编码单元,深度表示分层减小编码单元的程度。随着深度的增加,编码单元可从最大编码单元减小到最小编码单元,其中,最大编码单元的深度是最小深度并且最小编码单元的深度是最大深度。由于编码单元的尺寸随着深度的增加从最大编码单元的尺寸减小,因此第k深度的子编码单元可包括多个第(k+n)深度的子编码单元(k和n是等于或大于1的整数)。

[0040] 根据将被编码的帧的尺寸的增加,按照更大的编码单元对图像进行编码可产生更高的图像压缩率。然而,如果仅使用大的编码单元对图像进行编码,则由于连续改变图像特征而不能有效地对图像进行编码。

[0041] 例如,当对平滑区域(诸如,海面或天空)进行编码时,编码单元的尺寸越大,图像的压缩率可增加的越多。然而,当对复杂区域(诸如,人或建筑物)进行编码时,编码单元的尺寸越小,图像的压缩率可增加的越多。

[0042] 因此,根据示例性实施例,可针对每个帧或每个条带设置最大图像编码单元和最大深度。因此,针对第一帧或第一条带的最大编码单元和最大深度可与针对第二帧或第二条带的最大编码单元和最大深度不同。由于最大深度表示编码单元可减小的最大次数,因此可根据最大深度可变地设置包括在最大图像编码单元中的每个最小编码单元的尺寸。可针对每个帧或每个条带、或针对每个最大编码单元单独地确定最大深度。

[0043] 编码深度确定器120确定最大编码单元的划分形状。可基于率失真(RD)代价的计算确定划分形状。确定的最大编码单元的划分形状被提供给编码信息编码器140,根据最大编码单元的图像数据被提供给图像数据编码器130。

[0044] 最大编码单元可被划分为子编码单元,子编码单元根据不同的深度具有不同的尺寸,并且可基于具有不同的尺寸的处理单元对包括在最大编码单元中的具有不同尺寸的子编码单元进行预测或频率变换。换句话说,设备100可基于具有多种尺寸和多种形状的处理单元执行关于图像编码的多个处理操作。为了对图像数据进行编码,执行处理操作(诸如,预测、变换和熵编码中的至少一个),其中,具有相同的尺寸或不同的尺寸的处理单元可被

用于每个操作。

[0045] 例如,用于对图像进行编码的设备100可选择与编码单元不同的处理单元,以预测所述编码单元。

[0046] 当编码单元的尺寸是 $2N \times 2N$ (其中, N 是正整数)时,用于预测的处理单元可以是 $2N \times 2N$ 、 $2N \times N$ 、 $N \times 2N$ 和 $N \times N$ 。换句话说,可基于具有这样的形状的处理单元执行运动预测:编码单元的高度和宽度中的至少一个被二等分。以下,作为预测的基础的处理单元被称为“预测单元”。

[0047] 预测模式可以是帧内模式、帧间模式、跳过模式中的至少一个,可仅针对具有特定尺寸或特定形状的预测单元执行特定预测模式。例如,可仅针对形状是正方形的尺寸为 $2N \times 2N$ 和 $N \times N$ 的预测单元执行帧内模式。另外,可仅针对尺寸为 $2N \times 2N$ 的预测单元执行跳过模式。如果在编码单元中存在多个预测单元,则可在针对每个预测单元执行预测之后,选择具有最小编码错误的预测模式。

[0048] 可选地,用于对图像进行编码的设备100可基于具有与编码单元的尺寸不同的尺寸的处理单元对图像数据执行变换。对于编码单元中的变换,可基于具有与编码单元的尺寸相同的尺寸或小于编码单元的尺寸的尺寸的处理单元执行所述变换。以下,作为变换的基础的处理单元被称为“变换单元”。变换可以是离散余弦变换(DCT)或Karhunen Loeve变换(KLT)、或任何其它定点空间变换。

[0049] 编码深度确定器120可使用基于拉格朗日乘子的RD优化来确定包括在最大编码单元中的子编码单元。换句话说,编码深度确定器120可确定从最大编码单元划分的多个子编码单元的形状,其中,多个子编码单元具有根据它们的深度的尺寸。图像数据编码器130通过基于由编码深度确定器120确定的划分形状对最大编码单元进行编码来输出比特流。

[0050] 编码信息编码器140对关于由编码深度确定器120确定的最大编码单元的编码模式的信息进行编码。换句话说,编码信息编码器140通过对关于最大编码单元的划分形状的信息、关于最大深度的信息和关于每个深度的子编码单元的编码模式的信息进行编码来输出比特流。关于子编码单元的编码模式的信息可包括:关于子编码单元的预测单元的信息、关于每个预测单元的预测模式的信息和关于子编码单元的变换单元的信息。

[0051] 关于最大编码单元的划分形状的信息可以是这样的信息,例如,表示每个编码单元是否被划分的标志信息。例如,当对最大编码单元进行划分和编码时,对表示最大编码单元是否被划分的信息进行编码。另外,当对从最大编码单元划分的子编码单元进行划分和编码时,对表示子编码单元是否被划分的信息进行编码。

[0052] 由于针对每个最大编码单元存在具有不同的尺寸的子编码单元,并且必须针对每个子编码单元确定关于编码模式的信息,所以可针对一个最大编码单元确定表示至少一个编码模式的信息。

[0053] 设备100可通过根据深度的增加将最大编码单元的高度和宽度两者二等分(即,将最大编码单元四等分)来产生子编码单元。也就是说,当第 k 深度的编码单元的尺寸是 $2N \times 2N$ 时,第 $(k+1)$ 深度的编码单元的尺寸是 $N \times N$ 。

[0054] 因此,用于对图像进行编码的设备100可考虑图像特征,基于最大编码单元的尺寸和最大深度确定每个最大编码单元的最佳划分形状。通过考虑图像特征动态地调整最大编码单元的尺寸并通过将最大编码单元划分为不同深度的子编码单元对图像进行编码,可对

具有各种分辨率的图像进行更有效的编码。

[0055] 图2是根据示例性实施例的对图像进行解码的设备200的框图。

[0056] 参照图2,对图像进行解码的设备200包括:图像数据获取单元210、编码信息提取器220和图像数据解码器230。

[0057] 图像数据获取单元210获得由设备200接收的比特流,对接收的比特流进行解析,以获得根据最大编码单元的图像数据,并将图像数据输出到图像数据解码器230。图像数据获取单元210可从当前帧或当前条带的头提取关于当前帧或当前条带的最大编码单元的信息。换句话说,图像数据获取单元210按照最大编码单元对比特流进行划分,从而图像数据解码器230可根据最大编码单元对图像数据进行解码。

[0058] 编码信息提取器220通过对接收的比特流进行解析来从当前帧的头提取以下信息:表示最大编码单元的信息、表示最大深度的信息、表示最大编码单元的划分形状的信息、表示子编码单元的编码模式的信息。表示划分形状的信息和表示编码模式的信息被提供给图像数据解码器230。

[0059] 表示最大编码单元的划分形状的信息可包括在最大编码单元中包括的表示根据深度具有不同尺寸的子编码单元的信息,并且可以是表示每个编码单元是否被划分的信息(例如,标志信息)。

[0060] 表示编码模式的信息可包括关于根据子编码单元的预测单元的信息、关于预测模式的信息和关于变换单元的信息。

[0061] 图像数据解码器230通过基于由编码信息提取器220提取的信息对每个最大编码单元的图像数据进行解码来恢复当前帧。

[0062] 图像数据解码器230可基于关于最大编码单元的划分形状的信息对包括在最大编码单元中的子编码单元进行解码。解码处理可包括:包括帧内预测和运动补偿的预测处理和逆变换处理。

[0063] 图像数据解码器230可基于关于预测单元的信息和关于预测模式的信息执行帧内预测或帧间预测,以对预测单元进行预测。图像数据解码器230还可基于关于子编码单元的变换单元的信息针对每个子编码单元执行逆变换。

[0064] 图3示出根据示例性实施例的分层编码单元。

[0065] 参照图3,分层编码单元可包括宽度×高度是 64×64 、 32×32 、 16×16 、 8×8 和 4×4 的编码单元。除具有完全正形状的这些编码单元以外,可还存在宽度×高度是 64×32 、 32×64 、 32×16 、 16×32 、 16×8 、 8×16 、 8×4 和 4×8 的编码单元。

[0066] 参照图3,针对分辨率为 1920×1080 的图像数据310,最大编码单元的尺寸被设置为 64×64 ,最大深度被设置为2。

[0067] 针对分辨率为 1920×1080 的图像数据320,最大编码单元的尺寸被设置为 64×64 ,最大深度被设置为3。针对分辨率为 352×288 的图像数据330,最大编码单元的尺寸被设置为 16×16 ,最大深度被设置为2。

[0068] 当分辨率高或者数据量大时,编码单元的最大尺寸可以相对大以增加压缩率并精确地反映图像特征。因此,针对具有比图像数据330更高的分辨率的图像数据310和图像数据320, 64×64 可被选择为最大编码单元的尺寸。

[0069] 最大深度表示分层编码单元中的层的总数。由于图像数据310的最大深度是2,所

以图像数据310的编码单元315根据深度的增加可包括长轴尺寸为64的最大编码单元和长轴尺寸为32和16的子编码单元。

[0070] 另一方面,由于图像数据330的最大深度是1,因此图像数据330的编码单元335根据深度的增加可包括长轴尺寸为16的最大编码单元和长轴尺寸为8和4的编码单元。

[0071] 然而,由于图像数据320的最大深度是3,因此图像数据320的编码单元325根据深度的增加可包括长轴尺寸为64的最大编码单元和长轴尺寸为32、16、8和4的子编码单元。由于随着深度增加基于更小的子编码单元对图像进行编码,因此示例性实施例适于对包括多个更小场景的图像进行编码。

[0072] 图4是根据示例性实施例的基于编码单元的图像编码器400的框图。

[0073] 帧内预测单元410在当前帧405中对帧内模式的预测单元执行帧内预测.运动估计器420和运动补偿器425使用当前帧405和参考帧495对帧间模式的预测单元执行帧间预测和运动补偿。

[0074] 基于从帧内预测单元410、运动估计器420和运动补偿器425输出的预测单元产生残差值,并且产生的残差值通过经过变换器430和量化器440被输出为量化的变换系数。

[0075] 量化的变换系数通过经过反量化器460和逆变换器470被恢复为残差值,通过过去块单元480和环路滤波单元490对恢复的残差值进行后处理,并且恢复的残差值被输出为参考帧495。量化的变换系数可通过经过熵编码器450被输出为比特流455。

[0076] 为了基于根据示例性实施例的编码方法执行编码,图像编码器400的组件(即,帧内预测单元410、运动估计器420、运动补偿器425、变换器430、量化器440、熵编码器450、反量化器460、逆变换器470、去块单元480和环路滤波单元490)基于最大编码单元、根据深度的子编码单元、预测单元和变换单元执行图像编码处理。

[0077] 图5是根据示例性实施例的基于编码单元的图像解码器500的框图。

[0078] 参照图5,比特流505经过解析器510,其中,解析器510对比特流进行解析,以提取将被解码的编码的图像数据和所述解码所必须的编码信息。编码的图像数据通过经过熵解码器520和反量化器530被输出为反量化的数据,并通过经过逆变换器540被恢复为残差值。通过将残差值添加到帧内预测单元550的帧内预测结果或运动补偿单元560的运动补偿结果来根据编码单元恢复残差值。恢复的编码单元通过过去块单元570和环路滤波单元580被用于下一个编码单元或下一个帧585、595的预测。

[0079] 为了基于根据示例性实施例的解码方法执行解码,图像解码器500的组件(即,解析器510、熵解码器520、反量化器530、逆变换器540、帧内预测单元550、运动补偿器560、去块单元570和环路滤波单元580)基于最大编码单元、根据深度的子编码单元、预测单元和变换单元执行图像解码处理。

[0080] 具体来讲,帧内预测单元550和运动补偿器560通过考虑最大编码单元和深度来确定子编码单元中的预测单元和预测模式,逆变换器540通过考虑变换单元的尺寸执行逆变换。

[0081] 图6示出根据示例性实施例的最大编码单元、子编码单元和预测单元。

[0082] 图1中示出的用于对图像进行编码的设备100和图2中示出的用于对图像进行解码的设备200考虑图像特征,使用分层编码单元执行编码和解码。可根据图像特征适应性地设置最大编码单元和最大深度,或者可根据用户的输入设置最大编码单元和最大深度。

[0083] 在图6中,分层编码单元结构600具有最大编码单元610,其中,最大编码单元610的高度和宽度是64且最大深度是4。深度沿着分层编码单元结构600的纵轴增加,并且随着深度增加,子编码单元620到650的高度和宽度减小。沿着分层编码单元结构600的横轴示出最大编码单元610和子编码单元620到子编码单元650的预测单元。

[0084] 最大编码单元610的深度为0且编码单元的尺寸(即,高度和宽度)为 64×64 。深度沿着纵轴增加,存在尺寸为 32×32 且深度为1的子编码单元620、尺寸为 16×16 且深度为2的子编码单元630、尺寸为 8×8 且深度为3的子编码单元640和尺寸为 4×4 且深度为4的子编码单元650。尺寸为 4×4 且深度为4的子编码单元650是最小编码单元,并且最小编码单元可被划分为预测单元,其中,每个预测单元小于最小编码单元。

[0085] 参照图6,根据每个深度沿着横轴示出预测单元的示例。也就是说,深度为0的最大编码单元610的预测单元可以是尺寸与编码单元610相等(即, 64×64)的预测单元、或尺寸小于尺寸为 64×64 的编码单元610的以下预测单元:尺寸为 64×32 的预测单元612、尺寸为 32×64 的预测单元614、或尺寸为 32×32 的预测单元616。

[0086] 深度为1且尺寸为 32×32 的编码单元620的预测单元可以是尺寸与编码单元620相等(即, 32×32)的预测单元、或尺寸小于尺寸为 32×32 的编码单元620的以下预测单元:尺寸为 32×16 的预测单元622、尺寸为 16×32 的预测单元624、或尺寸为 16×16 的预测单元626。

[0087] 深度为2且尺寸为 16×16 的编码单元630的预测单元可以是尺寸与编码单元630相等(即, 16×16)的预测单元、或尺寸小于尺寸为 16×16 的编码单元630的以下预测单元:尺寸为 16×8 的预测单元632、尺寸为 8×16 的预测单元634、或尺寸为 8×8 的预测单元636。

[0088] 深度为3且尺寸为 8×8 的编码单元640的预测单元可以是尺寸与编码单元640相等(即, 8×8)的预测单元、或尺寸小于尺寸为 8×8 的编码单元640的以下预测单元:尺寸为 8×4 的预测单元642、尺寸为 4×8 的预测单元644、或尺寸为 4×4 的预测单元646。

[0089] 最后,深度为4且尺寸为 4×4 的编码单元650是最小编码单元和最大深度的编码单元,并且编码单元650的预测单元可以是尺寸为 4×4 的预测单元650、尺寸为 4×2 的预测单元652、尺寸为 2×4 的预测单元654或尺寸为 2×2 的预测单元656。

[0090] 图7示出根据示例性实施例的编码单元和变换单元。

[0091] 图1中示出的用于对图像进行编码的设备100和图2中示出的用于对图像进行解码的设备200使用最大编码单元或从最大编码单元划分的等于或小于最大编码单元的子编码单元来执行编码和解码。在编码和解码过程中,将用于变换的变换单元的尺寸选择不大于相应的编码单元的尺寸。例如,参照图7,当当前编码单元710的尺寸为 64×64 时,可使用尺寸为 32×32 的变换单元720执行变换。

[0092] 图8A到图8D示出根据示例性实施例的编码单元、预测单元和变换单元的划分形状。

[0093] 图8A和图8B分别示出根据示例性实施例的编码单元和预测单元。

[0094] 图8A示出由图1中示出的用于对图像进行编码的设备100选择的以对最大编码单元810进行编码的划分形状。设备100将最大编码单元810划分为多种形状,执行编码,通过基于R-D代价对多种划分形状的编码结果进行相互比较来选择最佳划分形状。当对最大编码单元810进行编码最佳时,可对最大编码单元810进行编码而不需如图8A到图8D所示划分

最大编码单元810。

[0095] 参照图8A,通过将最大编码单元810划分为深度等于或大于1的子编码单元对深度为0的最大编码单元810进行编码。也就是说,将最大编码单元810划分为深度为1的4个子编码单元,并且将深度为1的所有或一些子编码单元划分为深度为2的子编码单元。

[0096] 将深度为1的子编码单元中位于右上侧的子编码单元和位于左下侧的子编码单元划分为深度等于或大于2的子编码单元。可将深度等于或大于2的一些子编码单元划分为深度为等于或大于3的子编码单元。

[0097] 图8B示出最大编码单元的预测单元的划分形状。

[0098] 参照图8B,可以以与最大编码单元810的划分不同的配置,来划分最大编码单元810的预测单元860。换句话说,每个子编码单元的预测单元可小于相应的子编码单元。

[0099] 例如,深度为1的子编码单元中位于右下侧的子编码单元854的预测单元可小于子编码单元854。另外,深度为2的子编码单元814、816、818、828、850和852中的一些子编码单元814、816、850和852的预测单元可分别小于子编码单元814、816、850和852。

[0100] 另外,深度为3的子编码单元822、832和848的预测单元可分别小于子编码单元822、832和848。所述预测单元可具有这样的形状:各个子编码单元沿着高度或宽度的方向被二等分;或者所述预测单元可具有这样的形状:各个子编码单元沿着高度和宽度的方向被四等分。

[0101] 图8C和图8D分别示出根据示例性实施例的预测单元和变换单元。

[0102] 图8C示出图8B中示出的最大编码单元810的预测单元的划分形状,图8D示出最大编码单元810的变换单元的划分形状。

[0103] 参照图8D,可与预测单元860不同地设置变换单元870的划分形状。

[0104] 例如,即使在图8C中深度为1的编码单元854的预测单元被选择为具有编码单元854的高度被二等分的形状,也可在图8D中选择具有与编码单元854相同的尺寸的变换单元。也就是说,图8A中的编码单元854的尺寸和形状与图8D的变换单元854的尺寸和形状相同,但图8B和图8C中的预测单元854的尺寸和形状与图8A中的编码单元854和图8D中的变换单元的尺寸和形状不同。同样地,即使深度为2的编码单元814和编码单元850的预测单元被选择为具有编码单元814和850的每个的高度被二等分的形状,也可选择具有与编码单元814和编码单元850中的每个编码单元的原始尺寸相同的尺寸的变换单元。

[0105] 可选择具有小于预测单元的尺寸的变换单元。例如,当深度为2的编码单元852的预测单元被选择为具有编码单元852的宽度被二等分的形状时,可选择具有沿高度和宽度方向将编码单元852四等分的形状的变换单元,其中,所述变换单元具有小于预测单元的形状的尺寸。

[0106] 图9是根据示例性实施例的熵编码设备900的框图。

[0107] 参照图9,熵编码设备900包括上下文建模单元910和熵编码器920。熵编码设备900可相应于图4中示出的熵编码器450。

[0108] 上下文建模单元910设置用于对关于当前块的数据进行熵编码的上下文模型。关于当前块的数据包括与指示对当前块进行编码的方法的信息和残差数据有关的各种语法元素。指示对当前块进行编码的方法的信息可以是用于描述对当前块进行编码的方法的头信息(诸如,指示块类型的信息、指示预测模式的信息等),残差数据可以是指示通过对残差

值执行变换而产生的变换系数的信息(诸如,编码块模式(CBP)、有效的映射(significance map)、级别信息(level information)等)。

[0109] 分别对语法元素执行熵编码和熵解码。详细地,使用针对每个语法元素的不同上下文函数计算上下文索引,使用基于上下文的自适应二进制算数编码(CABAC)或基于上下文的自适应可变长编码(CAVLC)执行熵编码和熵解码。因此,在下文中,当前块的熵编码或熵解码是指针对当前块的预定的语法元素的熵编码或熵解码,并且预定的语法元素是指多种语法元素之一。

[0110] 一种现有技术的编解码器(诸如,运动图像专家组(MPEG)-4H.264/MPEG-4高级视频编码(AVC))通过参考与当前块相邻的先前编码的块的语法元素来执行上下文建模,从而对当前块的预定的语法元素进行熵编码。上下文建模是指设置用于对当前块进行熵编码的上下文模型,设置上下文模型的操作包括计算上下文索引的操作。

[0111] 然而,当通过参考先前编码的块的语法元素来对当前进行熵编码时,发生存储器的频繁访问。另外,由于为了参考先前编码的块的语法元素,语法元素应被连续地存储在存储器中,因此熵编码需要大容量的存储器。当所有的块被熵编码时,上下文建模单元910不是重复地执行上下文建模,而是通过对预定数量的块进行组合来设置作为用于执行上下文建模的单元的上下文单元,并对上下文单元仅执行一次上下文建模。将参照图10和图11详细描述上述处理。

[0112] 图10示出根据示例性实施例的共享上下文模型的方法。

[0113] 一种现有技术的编解码器(诸如,MPEG-4H.264/MPEG-4AVC)参考与当前块左侧相邻的先前编码的块和/或与当前块的上侧相邻的先前编码的块,从而对当前块进行熵编码。换句话说,为了对块X1 1010的语法元素A进行熵编码,将与块X1 1010的左侧相邻的先前编码的块A 1020的语法元素A和/或与块X1 1010的上侧相邻的先前编码的块B 1030的语法元素A代入到预定的上下文函数,并基于所述代入计算用于对块X1 1010的语法元素A进行熵编码的上下文索引。上下文函数是用于确定上下文索引的函数,并可针对每个语法元素被确定。

[0114] 另外,在现有技术的编解码器中,当完成块X1 1010的熵编码并且块X2 1012的语法元素被熵编码时,将与块X2 1012相邻的先前编码的块X1 1010的语法元素A和块C 1040的语法元素A代入到上下文函数,从而可基于所述代入计算上下文索引。

[0115] 然而,上下文建模单元910将块X1 1010、块X2 1012、块X3 1014和块X4 1016组合为一个上下文单元1000,并对该上下文单元1000执行上下文建模。例如,当通过对上下文单元1000中的被最初编码的块X1 1010执行上下文建模来计算预定的上下文索引时,上下文建模单元910不单独计算用于执行上下文建模的上下文单元1000的其它块1012至1016的上下文索引。上下文单元1000中的其它块1012至1016的上下文索引被设置为与块X1 1010的上下文索引相同(即,相等)。

[0116] 不需要重复地执行上下文建模以对块X2 1012进行熵编码,并且可通过使用用于对块X1 1010进行熵编码的上下文索引来对块X2 1012进行熵编码。上下文单元1000的设置中没有任何限制,并且将连续编码的块组合为一个组的所有方法可用于设置上下文单元1000。指示设置上下文单元1000的方法的信息可连同熵编码的结果一起被编码,并可被插入到图像数据中。然而,当通过使用相同的方法由编码器和解码器设置上下文单元1000时,

可不必对指示设置上下文单元1000的方法的信息进行编码。

[0117] 由于如图10所示对上下文单元1000中的四个块执行上下文建模,因此当执行上下文建模时,避免发生延时,并且可以以更高的速度执行熵编码。另外,由于相邻块可具有相似的特性的概率很高,因此即使相邻的块共享上下文索引并被熵编码,编码的压缩率也不会降低。

[0118] 图11示出根据示例性实施例的共享上下文模型的方法。

[0119] 参照图11,作为用于执行上下文建模的单元的上下文单元1100可被设置为与以上参照图1至图8描述的最大编码单元相同。换句话说,为了对图11中示出的块1110至1130进行熵编码,上下文建模单元910仅执行一次上下文建模。

[0120] 为了对最大编码单元中被最初编码的块X1 1110的语法元素A进行熵编码,将块A1140的语法元素A和块B 1150的语法元素A代入到预定的上下文函数,并且基于所述代入计算块X1 1110的上下文索引。随后,将用于对其它块1112至1130的语法元素A进行熵编码的上下文索引设置为与块X1 1110的上下文索引相同。对包括在最大编码单元中的多个块1110至1130仅执行一次上下文建模,以避免由于频繁的上下文建模引起的低效率。

[0121] 返回参照图9,熵编码器920基于由上下文建模单元910设置的上下文模型对关于当前块的数据进行熵编码,从而产生比特流。熵编码器920对当前块的预定的语法元素进行熵编码。

[0122] 当当前块是上下文单元1000或1100的初始块时,上下文建模单元910基于与当前块相邻的至少一个先前编码的块设置用于对当前块进行熵编码的上下文模型,并且熵编码器920根据设置的上下文模型对当前块进行熵编码。

[0123] 当当前块不是上下文单元1000或1100的初始块1010或1110时,上下文建模单元910将用于对当前块进行熵编码的上下文模型设置为与用于对初始块1010或1110进行熵编码的上下文模型相同,熵编码器920根据设置的上下文模型对当前块进行熵编码。熵编码是基于上下文的自适应熵编码,并且基于上下文的自适应熵编码包括CABAC和/或CAVLC。

[0124] 关于CABAC,熵编码器920基于关于最可能的符号的信息并基于关于最可能的符号的发生概率的信息执行算术编码,其中,最可能的符号对应于由上下文建模单元910计算的上下文索引。当上下文索引在上下文单元1000或1100中被设置时,与上下文索引相应的最可能的符号以及最可能的符号的发生概率被确定。熵编码器920基于确定的最可能的符号和确定的最可能的符号的发生概率,来对上下文单元1000或1100的所有块执行熵编码。

[0125] 当完成当前块的熵编码时,熵编码器920可更新最可能的符号和最可能符号的发生概率中的至少一个。如果最可能的符号已根据当前块的熵编码的结果被改变,则熵编码器920改变关于与由上下文建模单元910设置的上下文索引相应的最可能的符号的信息。另外,熵编码器920根据当前块的熵编码的结果更新最可能的符号的发生概率,其中,所述最可能的符号相应于由上下文建模单元910设置的上下文索引。

[0126] 每当对块进行熵编码时,或者在每个预定的更新时间段中,可执行更新。当在每个预定的更新时间段中执行更新时,熵编码器920可基于上下文单元1000或1100的预定的块的熵编码的结果来执行更新。在图10的上下文单元1000的情况下,熵编码器920可对上下文单元1000的初始块1010进行熵编码,并随后可基于熵编码的结果执行更新,或者,熵编码器920可对上下文单元1000的最后块1016进行熵编码,并随后可基于熵编码的结果执行更新。

[0127] 在CAVLC中,熵编码器920可基于由上下文建模单元910设置的上下文模型选择可变长编码表。根据CAVLC,可通过参考与当前块的左侧相邻的先前编码的块或与当前块的上侧相邻的先前编码的块,来选择用于对当前块进行熵编码的可变长编码表。对于更有效的可变长编码,可通过参考先前编码的块来选择多个可变长编码表之一,从而可执行适应于上下文的熵编码。

[0128] 然而,根据示例性实施例,由于不是通过针对每个块反复地选择可变长编码表,而是通过针对包括在上下文单元1000或1100中的所有的块仅选择一次可变长编码表,对上下文单元1000或1100仅执行一次上下文建模,因此熵编码器920可根据上下文单元1000或1100中选择的可变长编码表对当前块执行熵编码。

[0129] 图12是根据本发明的实施例的熵解码设备1200的框图。

[0130] 参照图12,根据示例性实施例的熵解码设备1200包括上下文建模单元1210和熵解码器1220。熵解码设备1200可相应于图5中示出的熵解码器520。图12的熵解码设备1200是用于对以上参照图9至图11描述的熵编码设备900的熵编码执行逆操作的设备。

[0131] 上下文建模单元1210设置用于对关于当前块的比特流进行熵解码的上下文模型。关于当前块的比特流是通过多种语法元素进行熵编码而产生的比特流,其中,所述多种语法元素包括指示对当前块进行编码的方法的信息和残差数据。

[0132] 用于熵解码的上下文建模是与用于熵编码的上下文建模相应的操作。因此,当当前块是上下文单元1000或1100中的初始熵解码的块1010或1110时,上下文建模单元1210基于与当前块相邻的至少一个先前解码的块,来设置用于对当前块进行熵解码的上下文模型。另外,当当前块不是上下文单元1000或1100中的初始熵编码的块1010或1110时,上下文建模单元1210将上下文模型设置为与上下文单元1000或1100的初始熵解码的块1010或1110的上下文模型相同。如上所述,上下文模型可以是上下文索引。对于设置上下文单元的方法没有限制。然而,如以上参照图11所述,上下文单元1100可被设置为与最大编码单元相同。

[0133] 熵解码器1220根据由上下文建模单元1210设置的上下文模型对当前块进行熵解码。熵解码器1220根据由上下文建模单元1210计算的上下文索引,对当前块执行基于上下文的自适应熵解码。基于上下文的自适应熵解码包括基于上下文的二进制算术解码或基于上下文的自适应可变长解码。

[0134] 当当前块是上下文单元1000或1100的初始块时,上下文建模单元1210基于与当前块相邻的至少一个先前解码的块设置用于对当前块进行熵解码的上下文模型,并且熵解码器1220根据设置的上下文模型对当前块进行熵解码。

[0135] 当当前块不是上下文单元1000或1100的初始块1010或1110时,上下文建模单元1210将用于对当前块进行熵解码的上下文模型设置为与用于对初始块1010或1110进行熵解码的上下文模型相同,熵解码器1220根据设置的上下文模型对当前块进行熵解码。

[0136] 在基于上下文的自适应二进制算术解码中,熵解码器1220基于这样的信息执行算术解码,其中,所述信息是指指示与由上下文建模单元1210计算的上下文索引相应的最可能的符号的信息以及指示最可能的符号的发生概率的信息。当上下文索引被设置为上下文单元1000或1100时,与上下文索引相应的最可能的符号以及最可能的符号的发生概率被确定。

[0137] 熵解码器1220基于确定的最可能的符号和确定的最可能的符号的发生概率,来对上下文单元1000或1100的所有块执行熵解码。

[0138] 当完成当前块的熵解码时,熵解码器1220可更新最可能的符号和最可能的符号的发生概率中的至少一个。如果最可能的符号已根据当前块的熵解码的结果被改变,则熵解码器1220改变指示与由上下文建模单元1210设置的上下文索引相应的最可能的符号的信息。另外,熵解码器1220根据当前块的熵解码的结果更新最可能的符号的发生概率,其中,所述最可能的符号相应于由上下文建模单元1210设置的上下文索引。以与熵编码器920相同的方式执行更新。换句话说,每当执行熵解码时,或者在每个预定的更新时间段中,可执行更新。

[0139] 在基于上下文的自适应可变长解码中,熵解码器1220可基于由上下文建模单元1210设置的上下文模型选择可变长解码表。由于对上下文单元1000或1100仅执行一次上下文建模,因此不是对所有的块重复执行可变长解码表的选择,而是针对包括在上下文单元1000或1100中的所有的块仅选择一次可变长解码表,使得熵解码器1220可根据上下文单元1000或1100中选择的可变长解码表对当前块执行熵解码。

[0140] 图13是示出根据示例性实施例的熵编码方法的流程图。

[0141] 参照图13,在操作1310,图9的熵编码设备900确定是否执行上下文建模,以便对当前块进行熵编码。当当前块是上下文单元1000或1100中的初始熵编码的块时,上下文建模应被执行,以便对当前块的预定的语法元素进行熵编码,并且当当前块不是上下文单元1000或1100中的初始熵编码的块时,上下文建模不需要被执行。

[0142] 作为在操作1310的确定的结果,如果确定为了对当前块进行熵编码需要执行上下文建模,则在操作1320,熵编码设备900设置用于对当前块进行熵编码的上下文模型。熵编码设备900将与当前块相邻的至少一个先前编码的块的语法元素代入到预定的上下文函数,从而计算上下文索引。

[0143] 作为在操作1310的确定的结果,如果确定为了对当前块进行熵编码不需要执行上下文建模,则在操作1330,熵编码设备900将当前块的上下文模型设置为和与当前块相邻的先前编码的块的上下文模型相同。可将用于对当前块进行熵编码的上下文索引设置为与包括当前块的上下文单元1000或1100中的初始熵编码的块1010或1110的上下文索引相同。

[0144] 在操作1340,熵编码设备900根据在操作1320或1330设置的上下文模型对当前块进行熵编码。熵编码可以是CABAC或CAVLC。

[0145] 图14是示出根据示例性实施例的熵解码方法的流程图。

[0146] 参照图14,在操作1410,图12的熵解码设备1200确定是否执行上下文建模,以便对当前块进行熵解码。当当前块是上下文单元1000或1100中的初始熵解码的块时,上下文建模应被执行,以便对当前块的预定的语法元素进行熵解码,当当前块不是上下文单元1000或1100中的初始熵解码的块时,上下文建模不需要被执行。

[0147] 作为在操作1410的确定的结果,如果确定为了对当前块进行熵解码需要执行上下文建模,则在操作1420,熵解码设备1200设置用于对当前块进行熵解码的上下文模型。熵解码设备900将与当前块相邻的至少一个先前解码的块的语法元素代入到预定的上下文函数,从而计算上下文索引。

[0148] 作为在操作1410的确定的结果,如果确定为了对当前块进行熵解码不需要执行上

下文建模,则在操作1430,熵解码设备900将当前块的上下文模型设置为和与当前块相邻的先前解码的块的上下文模型相同。可将用于对当前块进行熵编码的上下文索引设置为与包括当前块的上下文单元1000或1100中的初始熵解码的块1010或1110的上下文索引相同。

[0149] 在操作1440,熵解码设备1200根据在操作1420或1430设置的上下文模型对当前块进行熵解码。熵解码可以是基于上下文的自适应二进制算术解码或基于上下文的自适应可变长解码。

[0150] 如上所述,可通过使用低复杂度硬件以高的速度对图像数据进行熵编码或熵解码,而在编码期间图像的压缩率不会显著地降低。

[0151] 虽然已参考附图具体地示出和描述了示例性实施例,但本领域的普通技术人员将理解,在不脱离由权利要求及其等同物限定的本发明的精神和范围的情况下,可在形式和细节上对示例性实施例进行各种改变。

[0152] 图1、图2、图4、图5、图9和图12中示出的用于对图像进行编码的设备、用于对图像进行解码的设备、熵编码设备和熵解码设备可包括连接到设备或编码器的每个单元的总线、与所述总线连接的至少一个处理器,用于执行命令的处理器以及与总线连接的用于存储命令、接收的消息和产生的消息的存储器。

[0153] 示例性实施例还可被实施为计算机可读记录介质上的计算机可读代码。计算机可读记录介质是可存储数据的任何数据存储装置,其中,所述数据可在被存储之后被计算机系统读取,以执行计算机可读代码。计算机可读记录介质的示例包括只读存储器(ROM)、随机存取存储器(RAM)、CD-ROM、磁带、软盘和光数据存储装置。计算机可读记录介质还可分布于联网的计算机系统,从而以分布的方式存储和执行计算机可读代码。可选择地,示例性实施例可被实施为计算机可读传输介质(诸如,信号或载波),以在网络(诸如,互联网)上传输。

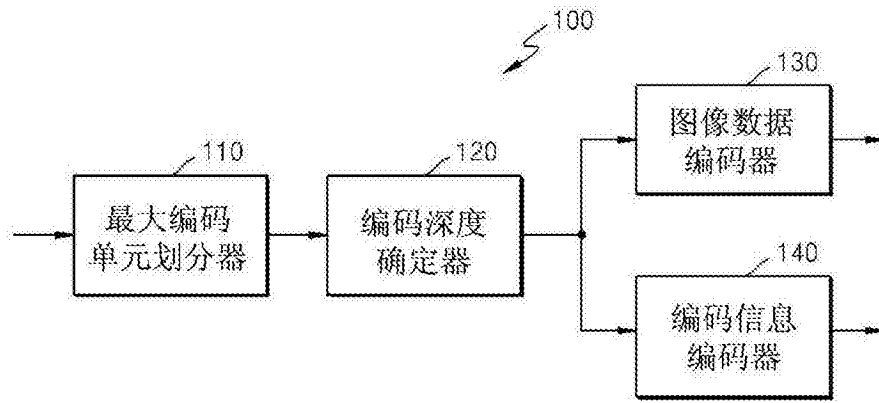


图1

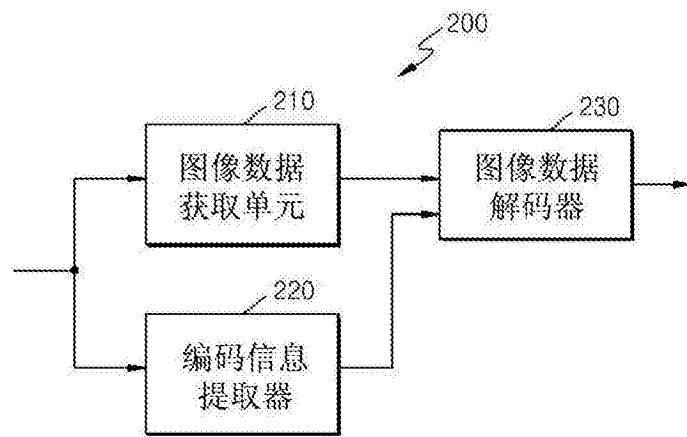


图2

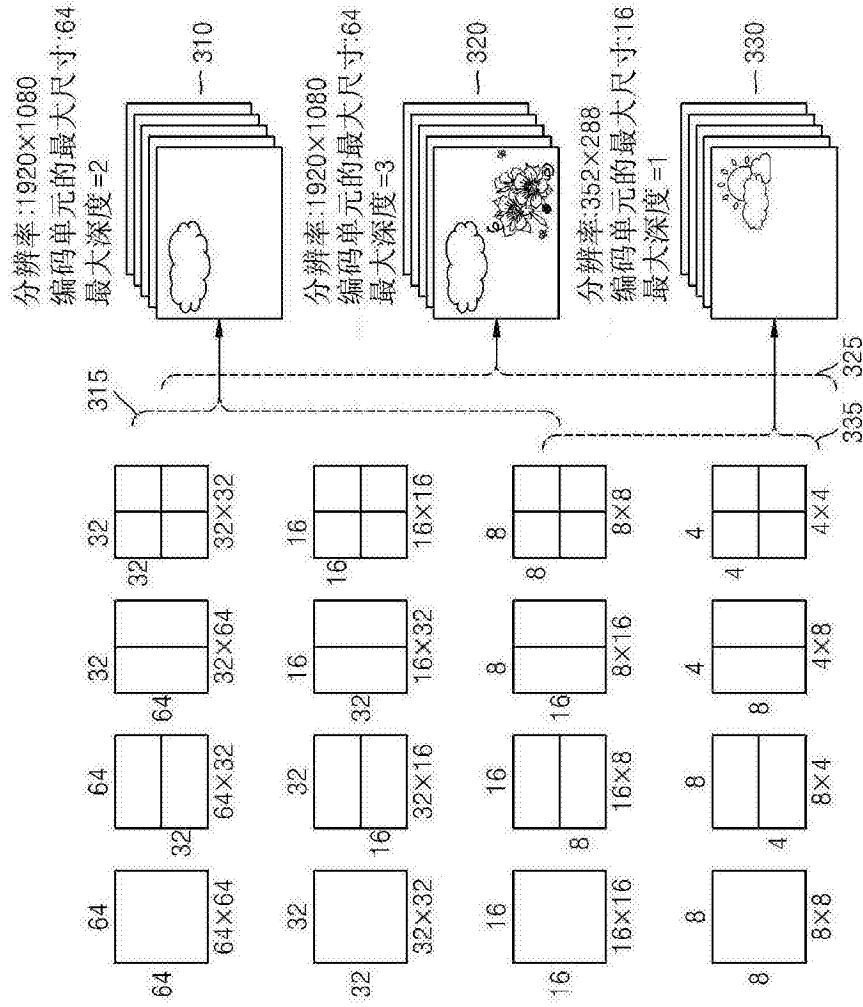


图3

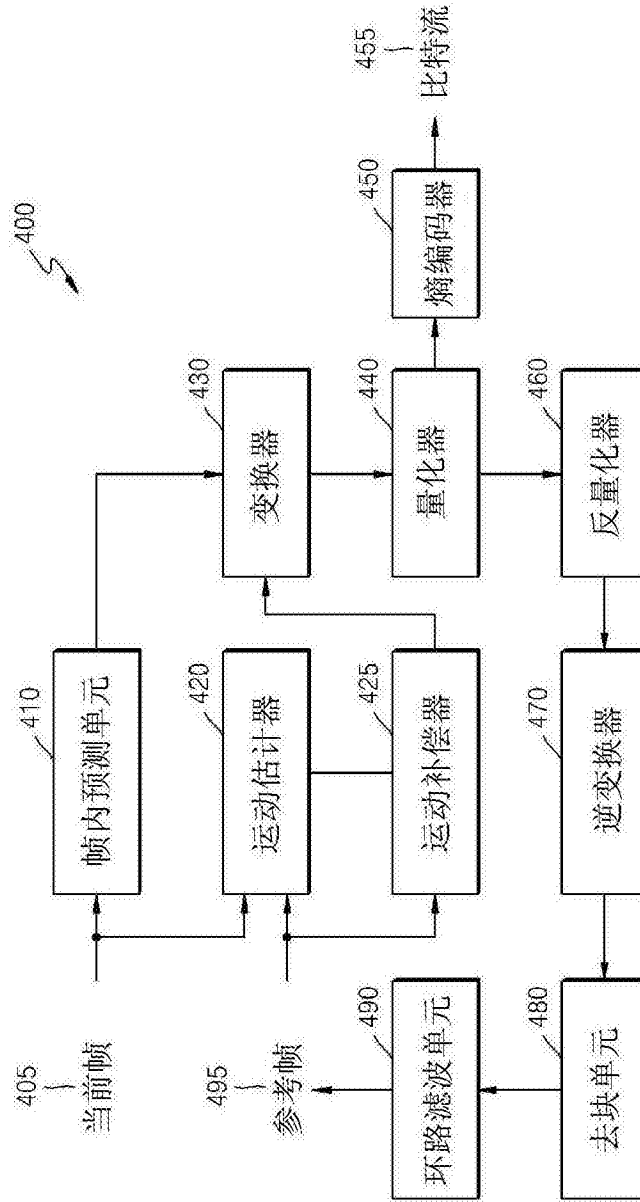


图4

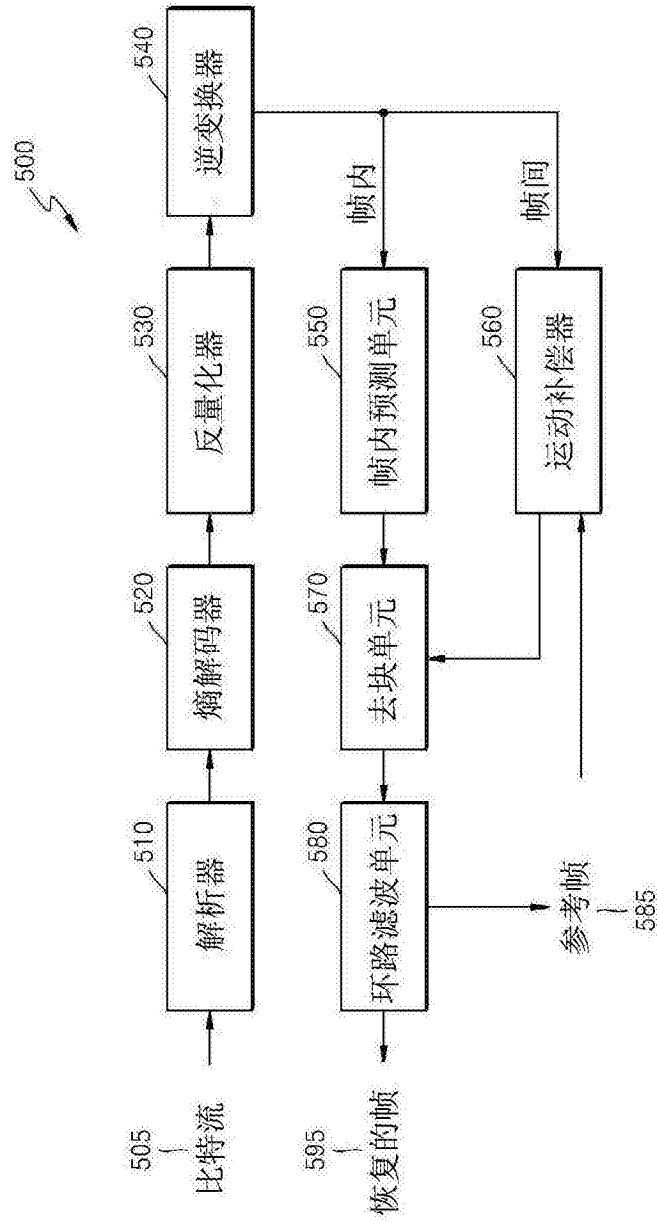


图5

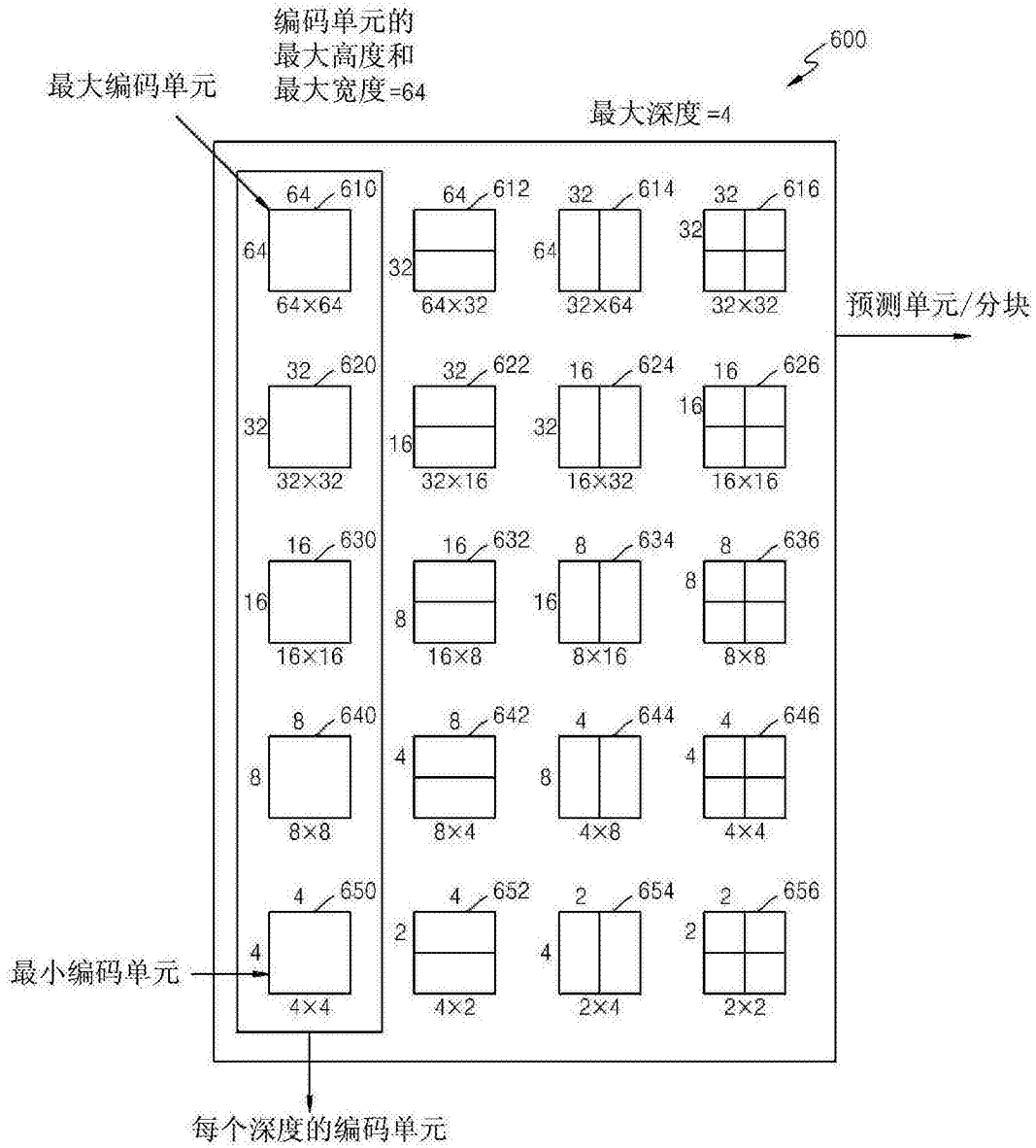


图6

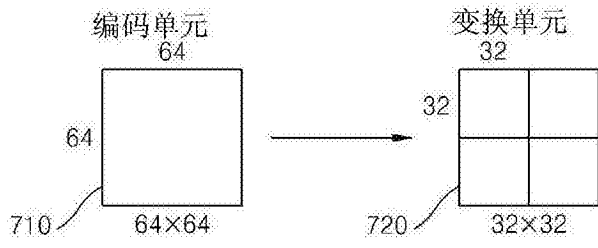
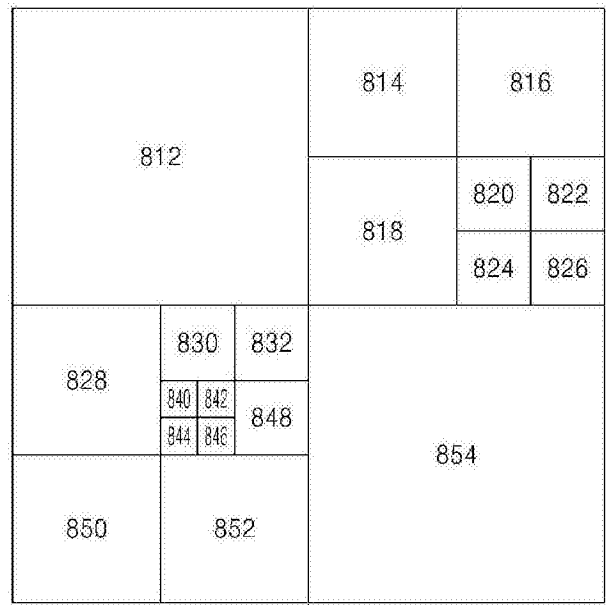
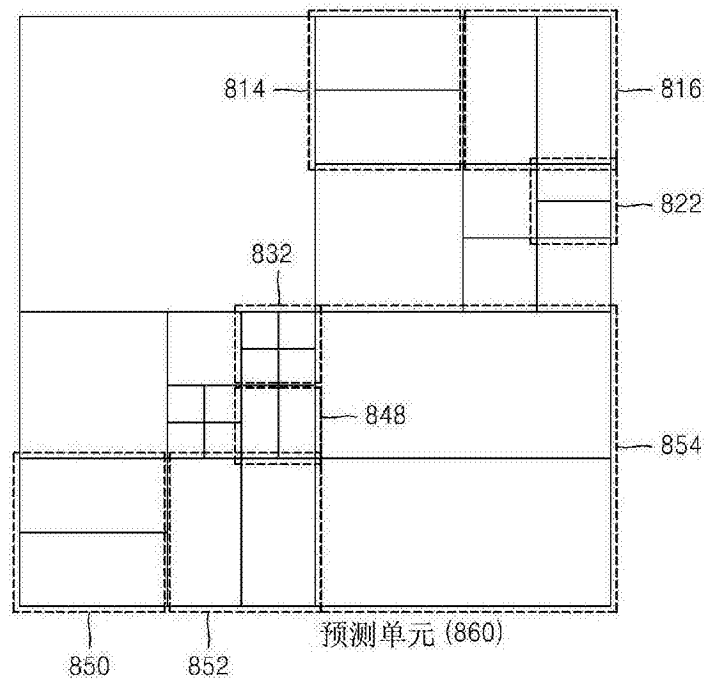


图7



编码单元 (810)

图8a



预测单元 (860)

图8b

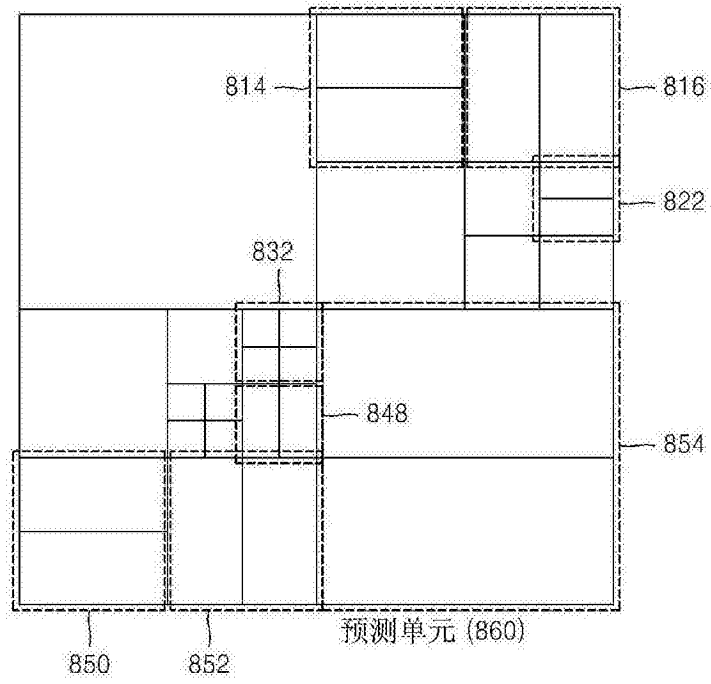


图8c

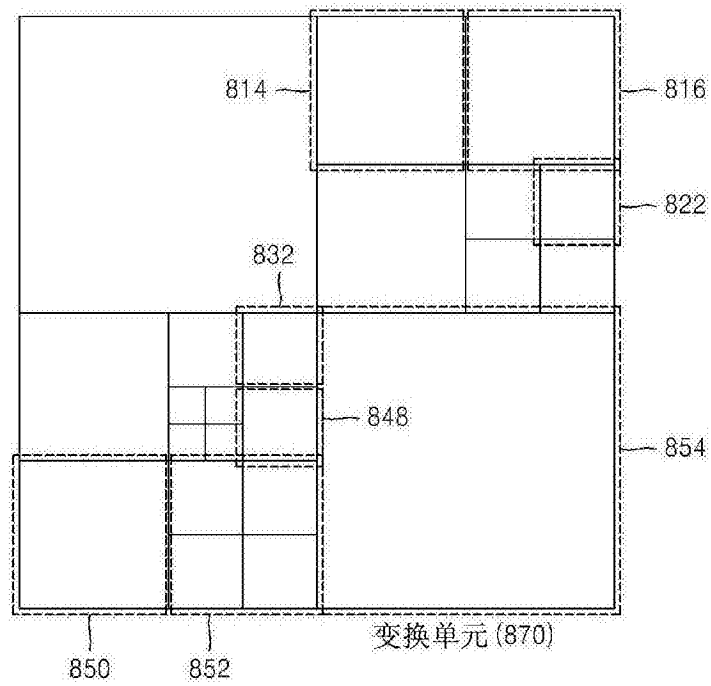


图8d

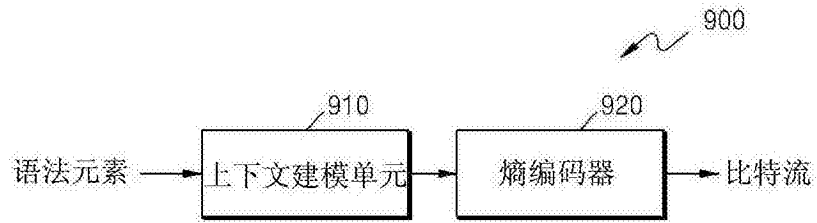


图9

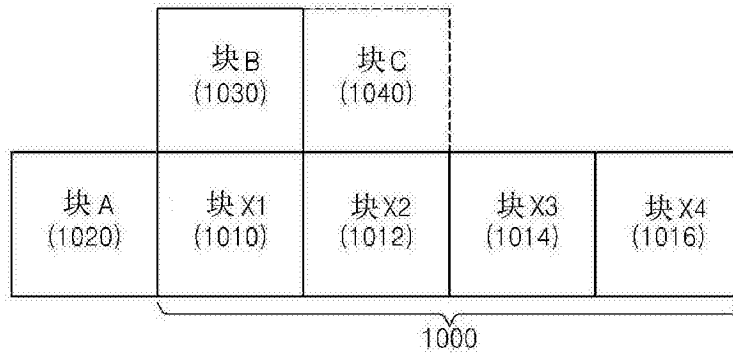


图10

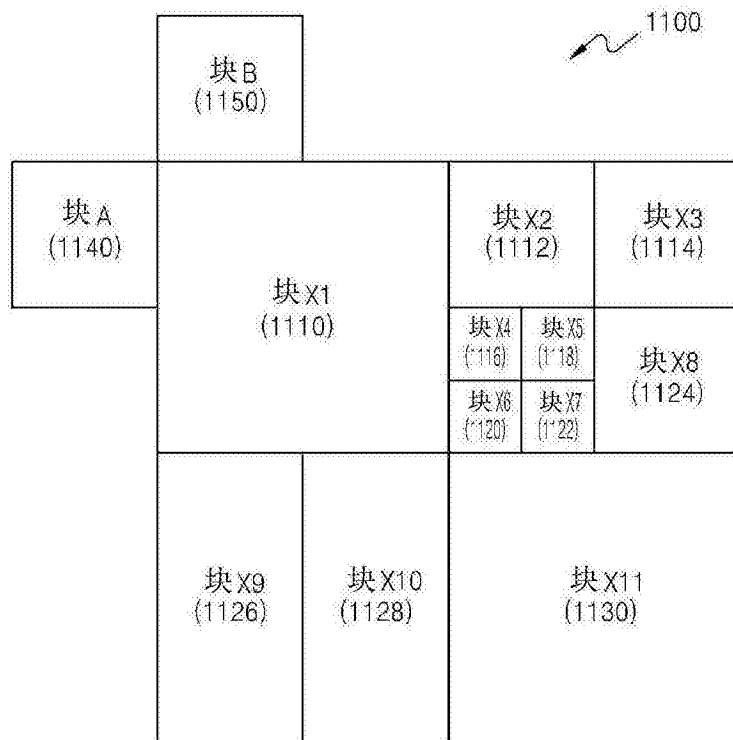


图11

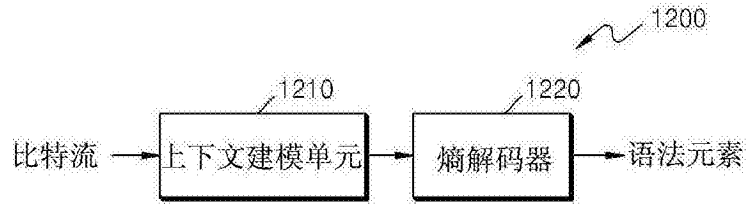


图12

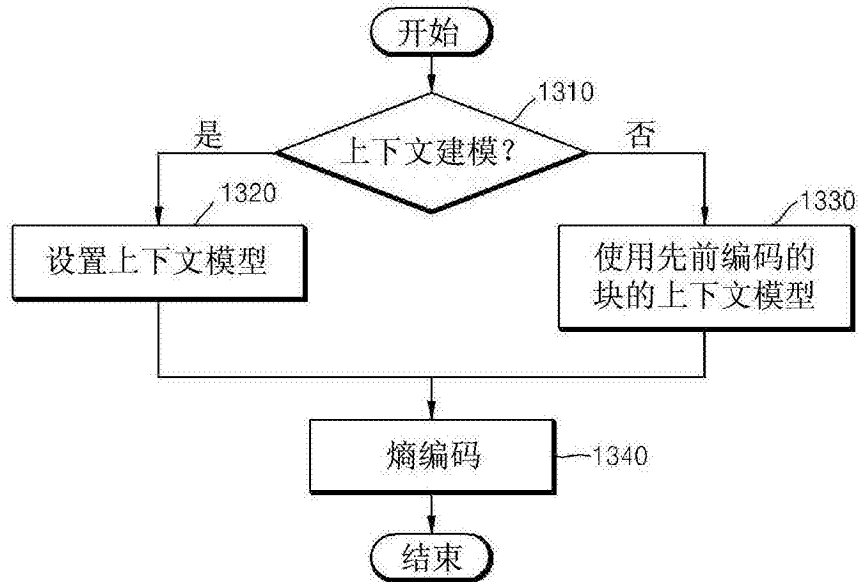


图13

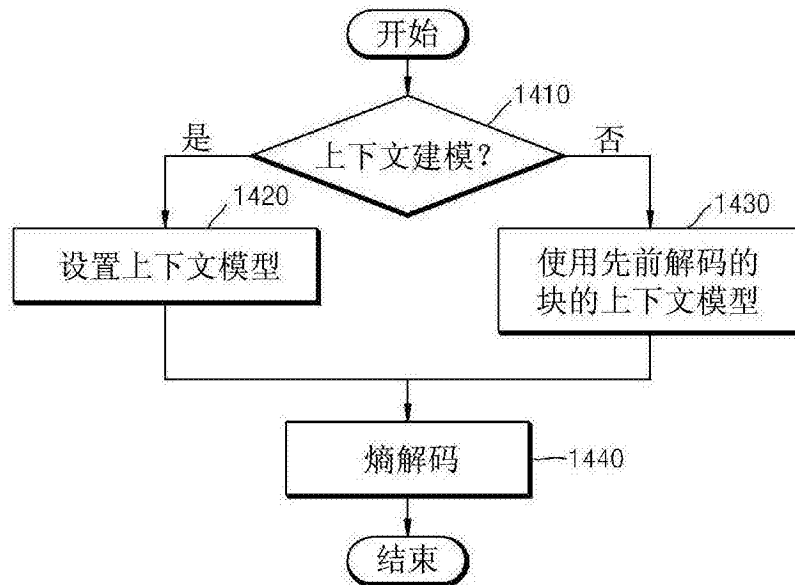


图14