



(12)发明专利

(10)授权公告号 CN 107347154 B

(45)授权公告日 2020.03.17

(21)申请号 201710760368.1

(22)申请日 2012.11.06

(65)同一申请的已公布的文献号

申请公布号 CN 107347154 A

(43)申请公布日 2017.11.14

(30)优先权数据

1160114 2011.11.07 FR

(62)分案原申请数据

201280066229.4 2012.11.06

(73)专利权人 杜比国际公司

地址 荷兰阿姆斯特丹

(72)发明人 F.亨利 G.克莱尔

(74)专利代理机构 中国国际贸易促进委员会专

利商标事务所 11038

代理人 宿小猛

(51)Int.Cl.

H04N 19/117(2014.01)

H04N 19/18(2014.01)

H04N 19/463(2014.01)

H04N 19/467(2014.01)

H04N 19/48(2014.01)

H04N 19/13(2014.01)

H04N 19/176(2014.01)

H04N 19/91(2014.01)

审查员 吴春芳

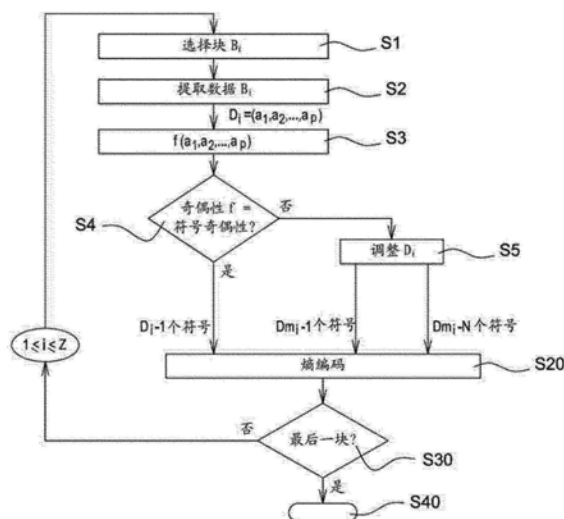
权利要求书2页 说明书16页 附图6页

(54)发明名称

用于编码和解码图像的方法、编码和解码设备、以及相应的计算机程序

(57)摘要

本公开涉及用于编码和解码图像的方法、编码和解码设备、以及相应的计算机程序。一种用于对被划分为多个分区的至少一个图像进行编码的方法，将被编码的当前分区包含数据，所述数据中至少10减1个数据项被分配符号。该编码方法的特征在于，针对所述当前分区，实现下列步骤：计算(S3)表示所述当前分区的除了所述符号外的数据的函数的值；将所述已计算的与所述符号的预定值进行比较(S4)；基于所述比较的结果，调整或不调整(S5)所述当前分区的数据项中的至少一个；以及在调整的情况下，对所述至少一个(20)已调整的数据项进行编码(S20)。



1. 一种计算机实现的用于对图像的符号数据隐藏允许分区进行解码的方法, 包括:

对代表符号数据隐藏允许分区的残留块的一组系数熵解码;

确定所述一组系数中的可调整系数的计数是否大于阈值, 其中所述一组系数中的可调整系数包括所述一组系数中的第一非零系数、所述一组系数中的最后的非零系数、以及所述一组系数中的在所述第一非零系数和所述最后的非零系数之间的系数;

如果系数中的可调整系数的计数不大于所述阈值, 所述第一非零系数的符号不被隐藏;

如果所述一组系数中的可调整系数的计数大于所述阈值, 则所述第一非零系数的符号被隐藏, 并且执行以下操作以确定所述第一非零系数的符号:

计算所述一组系数中的非零系数之和;

使用非零系数之和计算奇偶校验数据;

基于所述奇偶校验数据为所述第一非零系数指定符号;

对包括符号被指定的所述第一非零系数的所述一组系数进行反量化, 以生成一组已反量化系数; 以及

对所述一组已反量化系数进行逆变换。

2. 根据权利要求1所述的方法, 其中, 所述阈值为4。

3. 一种计算机实现的用于对数据项进行编码的方法, 包括:

获得代表所述数据项的一组系数;

对所述一组系数进行变换以生成一组已变换系数;

量化所述一组已变换系数以生成一组已量化系数, 包括具有符号的第一非零已量化系数, 所述符号为正或者负;

确定该组已量化系数中的可调整系数的计数是否大于阈值, 其中该组已量化系数中的可调整系数包括该组系数中的第一非零已量化系数、该组系数中的最后的非零已量化系数、以及该组系数中的在所述第一非零已量化系数和所述最后的非零已量化系数之间的已量化系数;

如果所述一组已量化系数中的可调整系数的计数不大于所述阈值, 所述第一非零已量化系数的符号不被隐藏, 并且被熵编码;

如果所述一组已量化系数中的可调整系数的计数大于所述阈值, 则所述第一非零已量化系数的符号被隐藏, 并且执行以下操作以隐藏所述第一非零已量化系数的符号:

将函数应用于所述一组已量化系数以生成输出数据;

比较所述输出数据与被分配给所述第一非零已量化系数的符号的值;

确定所述比较不满足标准;

响应于确定所述比较不满足标准, 至少调整所述一组已量化系数中的特定非零已量化系数; 以及

熵编码第二组已量化系数, 所述第二组已量化系数包括 (i) 不具有所述符号的第一非零已量化系数, (ii) 调整后的特定非零已量化系数, 以及 (iii) 所述一组已量化系数中的其它系数。

4. 根据权利要求3所述的方法, 其中, 所述阈值为4。

5. 一种非暂态计算机可读存储设备, 包括指令, 所述指令在由处理器执行时导致执行

根据权利要求1-4中任一项所述的方法。

6. 一种用于对图像的符号数据隐藏允许分区进行解码的设备, 包括:

一个或多个处理器, 以及

一个或多个计算机可读存储介质, 存储有指令, 所述指令在被所述一个或多个处理器执行时导致执行根据权利要求1-2中任一项所述的方法。

7. 一种用于对数据项进行编码的设备, 包括:

一个或多个处理器, 以及

一个或多个计算机可读存储介质, 存储有指令, 所述指令在被所述一个或多个处理器执行时导致执行根据权利要求3-4中任一项所述的方法。

8. 一种包括用于执行根据权利要求1-4中任一项所述的方法的部件的装置。

用于编码和解码图像的方法、编码和解码设备、以及相应的计算机程序

[0001] 本申请是申请号为201280066229.4、申请日为2012年11月6日、发明名称为“用于编码和解码图像的方法、编码和解码设备、以及相应的计算机程序”的发明专利申请的分案申请。

技术领域

[0002] 本发明主要属于图像处理领域,且具体而言属于数字图像和数字图像序列的编码和解码。

[0003] 由此,本发明尤其可应用于由目前的视频编码器(MPEG、H.264等)或未来的视频编码器(ITU-T/VCEG(H.265)或ISO/MPEG(HEVC))实现的视频编码。

背景技术

[0004] 目前的视频编码器(MPEG、H.264等)使用视频序列的分块表示。图像被分为多个宏块,每个宏块本身被分为多个块,并且每个块或每个宏块通过图像内或图像间预测来编码。于是,某些图像通过空间预测(帧内预测)来编码,而其它图像借助于本领域技术人员已知的运动补偿通过相对于一个或多个编码-解码参考图像的时间预测(帧间预测)进行编码。

[0005] 对于每个块,存在与由预测所减掉的原始块相对应的、还被称为预测残留的已编码的残留块。残留块通过离散余弦变换(DCT)类型的变换来变换,然后借助于例如标量类型的量化来量化。在量化步骤结束后得到系数,所述系数中一些是正的而其它的是负的。然后以读的顺序,通常为之字形(和JPEG标准一样)来遍历系数,由此可利用高频中的大量零系数。在上述遍历结束后,得到系数的一维列表,其可被称为“量化残留”。然后通过熵编码来对该列表的系数进行编码。

[0006] 熵编码(例如算术编码或霍夫曼编码类型)以如下方式执行:

[0007] -信息项被熵编码,以表示列表中的最后的非零系数的位置,

[0008] -针对位于最后的非零系数之前的每个系数,信息项被熵编码以表示系数是否为零,

[0009] -针对之前所表示的每个非零系数,信息项被熵编码以表示系数是否等于1,

[0010] -针对位于最后的非零系数之前的、不等于1的每个非零系数,幅度信息项(系数的绝对值,其值被减少2)被熵编码,

[0011] -针对每个非零系数,分配给该系数的符号由“0”(用于+符号)或“1”(用于-符号)编码。

[0012] 根据例如H.264技术,在宏块被分为多个块时,与每个块对应的数据信号被发送给解码器。该信号包括:

[0013] -在上述列表中包含的量化残留,

[0014] -表示所使用的编码模式的信息,特别是:

[0015] ●预测模式(帧内预测、帧间预测、默认预测,其进行没有信息项被发送到解码器

的预测(被称为“跳过”);

[0016] ●指定预测类型的信息(方向、参考图像等);

[0017] ●分区类型;

[0018] ●变换类型,例如 4×4 DCT、 8×8 DCT等;

[0019] ●运动信息,如果有必要的话;

[0020] ●等等。

[0021] 逐个图像以及针对每个图像逐个宏块地执行解码。针对宏块的每个分区,读取流的相应元素。执行块的系数的反量化和逆变换,以生成解码的预测残留。然后,计算该分区的预测并通过将预测加到被解码的预测残留来重构该分区。

[0022] 由此,如H.264标准所实现的,通过竞争实现的帧内编码或帧间编码依赖于将例如上述那些编码信息项之类的各个编码信息项与选择最佳模式的目标进行竞争,也就是说,该最佳模式能根据预定的性能标准例如本领域技术人员众所周知的比特率/失真成本来使所讨论分区的编码最优化。

[0023] 表示被选中编码模式的信息被包含在由编码器发送到解码器的数据信号中。解码器由此能够识别在编码器上被选中的编码模式,然后能够根据该模式应用预测。

[0024] 文档“Data Hiding of Motion Information in Chroma and Luma Samples for Video Compression”,J.-M.Thiesse,J.Jung and M.Antonini,International workshop on multimedia signal processing,2011介绍了一种在视频压缩期间实现的数据隐藏方法。

[0025] 更准确地说,建议避免在要被发送到解码器的信号中包含至少一个竞争指数,所述至少一个竞争指数例如出现在要被发送的多个竞争指数中。这样的指数例如是指数MVComp,其表示可标识运动矢量预测器的信息项,该预测器用于在帧间预测模式下预测的块。可等于0或1的该指数未被直接写入编码数据信号中,而是通过量化残留的系数之和的奇偶校验来传输。在量化残留的奇偶性和指数MVComp之间建立关联。通过示例,量化残留的偶数值与值为0的指数MVComp关联,而量化残留的奇数值与值为1的指数MVComp关联。会出现两种情形。在第一种情形中,如果量化残留的奇偶性已经对应于所期望发送的指数MVComp的奇偶性,则以传统方式对量化残留编码。在第二种情形中,如果量化残留的奇偶性与所期望发送的指数MVComp的奇偶性不同,则以使量化残留的奇偶性与指数MVComp的奇偶性相同的方式来对量化残留进行调整。该调整在于将量化残留的一个或多个系数增加或减少奇数值(例如+1、-1、+3、-3、+5、-5等),并且仅保留使预定标准最优化的调整,在该情形中,所述预定标准为前述比特率-失真成本。

[0026] 在解码器上,不从信号读取指数MVComp。解码器通过传统地确定残留来简单地勉强使用。如果该残留的值是偶数,则指数MVComp被设置为0。如果该残留的值是奇数,指数MVComp被设置为1。

[0027] 根据刚才介绍的技术,不会总是以最优的方式选择已调整的系数,从而所应用的调整给要发送到解码器的信号带来了干扰。这样的干扰对于视频压缩的效率必然有害。

[0028] 此外,指数MVComp不会构成要隐藏的信息的最有益项,因为该指数等于0或1的概率是不相等的。因此,如果该指数以传统方式通过熵编码来编码,则在发送到解码器的压缩文件中,将由比所发送的每个指数MVcomp一个比特更少量的数据来表示该指数。于是,如果

在量化残留的奇偶校验中发送指数MVComp,则由此节省的数据量小于每个指数MVComp一个比特时节省的数据量,而残留的奇偶性使其能够传输每个指数一比特的信息项。

[0029] 于是,信令成本的降低以及压缩效率不是最优的。

发明内容

[0030] 本发明的一个目标是纠正上述现有技术中的缺陷。

[0031] 为此,本发明的目标涉及一种用于对被划分为多个分区的至少一个图像进行编码的方法,将被编码的当前分区包含数据,所述数据中的至少一个数据项被分配符号。

[0032] 根据本发明的方法值得注意的是,它针对所述当前分区实现下列步骤:

[0033] -计算表示所述当前分区的除了符号以外的数据的函数的值,

[0034] -将所述已计算的值与所述符号的预定值进行比较,

[0035] -根据比较的结果,调整或不调整所述当前分区的数据中的至少一个,

[0036] -在调整的情形下,对所述至少一个已调整的数据项进行编码。

[0037] 这样的布置使其可有利地将数据隐藏技术应用于要被编码的分区的数据的符号。由于正符号或负符号出现的概率是相同的这一事实,所以该符号实际上是与隐藏特别相关的信息项。因此,假设需要用一比特来对符号编码,由此可能通过隐藏该信息项来节省要发送到解码器的信号中的一个比特,由此充分降低信令成本。

[0038] 需要注意的是,在与图像数据项关联的信息(符号、幅度等)中,它们中的很少是等概率的。符号是等概率的信息项,因此隐藏该类型的信息项有特别的好处,由此可增加压缩性能。

[0039] 在特定的实施例中,在上述比较步骤中考虑多个符号的情形下,该比较步骤包括将表示当前分区的数据的函数的所计算的值与表示所述多个符号的函数的值进行比较。

[0040] 该布置可优化算术编码器的压缩性能,同时优化信令成本的降低,因为它可在要发送到解码器的信号中隐藏若干个符号。

[0041] 相关地,本发明涉及一种用于对被划分为多个分区的至少一个图像进行编码的设备,将被编码的当前分区包含数据,所述数据中至少一个数据项被分配符号。

[0042] 该设备值得注意的是,针对所述要编码的当前分区,它包括处理装置,其能够:

[0043] -计算表示所述当前分区的除了符号以外的数据的函数的值,

[0044] -将所述已计算的值与所述符号的预定值进行比较,

[0045] -根据所述比较的结果,调整或不调整所述当前分区的数据中的至少一个,

[0046] 并且其中所述设备包括在由所述处理装置调整的情形中用于对至少一个已调整的数据项进行编码的装置。

[0047] 相应地,本发明还涉及一种用于对表示之前已被编码的划分为多个分区的至少一个图像的数据信号进行解码的方法,将被解码的当前分区包含数据,所述数据中至少一个数据项被分配符号,

[0048] 该解码方法值得注意的是,针对当前分区,它包括下列步骤:

[0049] -对所述当前分区的除了所述符号以外的数据进行解码,

[0050] -计算表示所述当前分区的已解码数据的函数的值,

[0051] -基于所述已计算的值来获取所述符号的值。

[0052] 在特定的实施例中,基于所计算的值来获取与多个符号分别关联的多个值。

[0053] 相关地,本发明涉及一种用于对表示之前已被编码的被划分为多个分区的至少一个图像的数据信号进行解码的设备,将被解码的当前分区包含数据,所述数据中的至少一个数据项被分配符号。

[0054] 该解码设备值得注意的是,针对被解码的所述当前分区,它包括处理装置,其能够:

[0055] -对所述当前分区的除了所述符号以外的数据进行解码,

[0056] -计算表示所述当前分区的已解码数据的函数的值,

[0057] -基于所述已计算的值来获取所述符号的值。

[0058] 本发明目标还在于一种包含指令的计算机程序,在计算机上执行该程序时,该指令用于执行以上编码或解码方法的步骤。

[0059] 该程序可以使用任何编程语言并且可以是源代码、目标代码或介于源代码与目标代码之间的代码的形式,例如部分编译的形式或任何其它期望的形式。

[0060] 本发明的又一主题还在于一种记录介质,其可被计算机读取并包含如上所述的计算程序的指令。

[0061] 该记录介质可以是能够存储程序的任意实体或设备。例如,该介质可以包括诸如ROM之类的存储介质,例如CD-ROM或微电子电路ROM,或者磁记录介质,例如磁盘(软盘)或硬盘。

[0062] 此外,该记录介质可以是诸如电信号或光信号之类的可传输介质,其可以经电缆或光缆通过无线电或通过其它方式来传递。根据本发明的程序可以特别地从因特网类型的网络来下载。

[0063] 或者,该记录介质可以是其中包含有程序的集成电路,该电路适于执行所讨论的方法或适于在该方法执行时使用。

[0064] 所述编码设备、解码方法、解码设备和前述计算机程序至少表现出与根据本发明的编码方法所具有的优势相同的优势。

附图说明

[0065] 通过阅读参考附图所描述两个优选实施例,其它特征和优点将变得明显,在附图中:

[0066] -图1表示根据本发明的编码方法的一般步骤,

[0067] -图2表示根据本发明的能执行图1中的编码方法的步骤的编码设备,

[0068] -图3表示根据本发明的编码方法的特定实施例,

[0069] -图4表示根据本发明的编码设备的特定实施例,

[0070] -图5表示根据本发明的解码方法的一般步骤,

[0071] -图6表示根据本发明的能够执行图5中的解码方法的步骤的解码设备,

[0072] -图7表示根据本发明的解码方法的特定实施例,

[0073] -图8表示根据本发明的解码设备的特定实施例。

具体实施方式

[0074] 现在将描述本发明的一般实施例,其中,根据本发明的编码方法被用于根据二进制流来对图像序列进行编码,该二进制流与通过根据H.264/MPEG-4AVC标准来编码而获得的二进制流接近。在该实施例中,根据本发明的编码方法例如以软件或硬件形式通过对初始符合H.264/MPEG-4AVC标准的编码器的调整来实现。

[0075] 根据本发明的编码方法以图1所示的包含步骤S1到S40的算法的形式来表示。

[0076] 根据本发明的实施例,根据本发明的编码方法在图2所示的实施例中的编码设备或编码器C0中实现。

[0077] 根据本发明,如图2中所示,在适当的编码步骤之前,以预定顺序将待编码的图像序列中的图像IE划分为多个(Z个)分区 B_1 、 B_2 、 \dots 、 B_i 、 \dots 、 B_Z 。

[0078] 应当注意的是,在本发明的意义上,术语“分区”表示编码单元。该后一术语特别用在当前正在制定的HEVC/H.265标准中,例如用在下列互联网地址可访问的文档中:

[0079] http://phenix.int-evry.fr/jct/doc_end_user/current_document.php?id=3286

[0080] 特别地,这种编码单元将矩形或正方形的像素集合或表现出其它几何形状的别的像素集合分组到一起,所述矩形或正方形的像素集合被称为块、宏块。

[0081] 在图2所示的实例中,所述分区是具有正方形形状并且全部具有相同尺寸的块。根据图像的尺寸,该尺寸不一定是块尺寸的倍数,最左边的块以及最底部的块可以不是正方形。在替代的实施例中,块例如可以是矩形尺寸和/或可以不互相对齐。

[0082] 每个块或每个宏块本身可以进一步被分为子块,子块自己可以再细分。

[0083] 这样的划分是通过图2所示的分区模块PC0来执行的,该分区模块PC0使用例如像这样的分区模块所众所周知的分区算法。

[0084] 在所述划分步骤之后,对所述图像IE的每个当前分区 B_i (其中 i 是大于等于1且小于等于Z的整数)进行编码。

[0085] 在图2所示的实例中,对当前图像IE的块 B_1 到 B_Z 中的每个块相继实施这样的编码。例如根据诸如本领域技术人员众所周知的“光栅扫描”遍历之类的遍历来对块进行编码。

[0086] 根据本发明的编码在如图2所示的编码器C0的编码软件模块MC_C0中实现。

[0087] 在图1所示的步骤S1的过程中,图2的编码模块MC_C0选择当前图像IE的待编码的第一块 B_1 作为当前块 B_i 。如图2所示,第一块 B_1 是图像IE左边的第一块。

[0088] 在图1所示的步骤S2的过程中,以列表 $D_1 = (a_1, a_2, \dots, a_p)$ 的形式抽取当前块 B_1 的数据。该抽取由如图2所示的软件模块EX_C0执行。该数据例如是像素数据,每个非零像素数据被分配有正符号或负符号。

[0089] 列表 D_1 中的每个数据与要被熵编码的各个数字信息项相关联。下面通过示例的方式来描述这样的数字信息项:

[0090] -针对位于列表 D_1 中的最后非零数据项之前的每个数据项,数字信息项例如比特将被熵编码,以表示数据项是否为零:如果数据项为零,则它例如是将被编码的值为0的比特,而如果数据项不为零,它是将被编码的值为1的比特;

[0091] -针对每个非零数据项,数字信息项例如比特要被熵编码,以表示数据项的绝对值是否等于1:如果它等于1,则它例如是将被编码的值为1的比特,而如果它不等于1,则它是

将被编码的值为0的比特。

[0092] 针对每个非零数据项,其绝对值不等于1且位于最后的非零数据项之前,则幅度信息项被熵编码;

[0093] 针对每个非零数据项,分配给它的符号被诸如被设置为“0”的比特的数字信息项编码(用于+符号)或诸如被设置为“1”的比特的数字信息项编码(用于-符号)。

[0094] 现在将参考图1来描述根据本发明的特定编码步骤。

[0095] 根据本发明,决定避免对列表D₁中的所述数据中的一个的至少一个符号进行熵编码。

[0096] 根据优选实施例,要隐藏的是第一非零数据项的符号。该符号例如是正的并且被分配给第一非零数据项,例如数据项a₂。

[0097] 在图1所示的步骤S3的过程中,处理模块MTR_CO计算函数f的值,其表示列表D₁中的数据。

[0098] 在要发送到解码器的信号中隐藏单个符号的优选实施例中,函数f是列表D₁中的数据之和的奇偶校验。

[0099] 在图1所示的步骤S4的过程中,借助之前在编码器C0定义的规则,处理模块MTR_CO验证要隐藏的符号的值的奇偶性是否对应于列表D₁中的数据之和的奇偶性。

[0100] 在所建议的实例中,所述规则是:正符号与值等于零的比特相关联,而负符号与值等于1的比特相关联。

[0101] 根据在根据本发明的编码器C0中采用的规则,如果符号为正,则由此对应于值为零的编码比特,并且如果列表D₁中的数据之和是偶数,则对上述列表D₁中除了第一非零数据项a₂的符号以外的数据进行熵编码的步骤S20。该步骤S20如图1所示。

[0102] 仍然根据在根据本发明的编码器C0中采用的规则,如果符号为负,则由此对应于值为1的编码比特,且如果列表D₁中的数据之和为奇数,则也对上述列表D₁中除了第一非零数据项a₂的符号以外的数据进行熵编码的步骤S20。

[0103] 根据在根据本发明的解码器C0中采用的规则,如果符号为正,则由此对应于值为零的编码比特,并且如果列表D₁的数据之和是奇数,则在图1所示的步骤S5的过程中,对列表D₁中的至少一个可调整数据项进行调整。

[0104] 仍然根据在根据本发明的解码器C0中采用的规则,如果符号为负,则由此对应于值为1的编码比特,并且如果列表D₁中的数据之和是偶数,则也进行对列表D₁中的至少一个可调整数据项的调整的步骤S5。

[0105] 根据本发明,如果数据项的值的调整不会引起解码器的任意去同步化,一旦该调整的数据项被解码器处理,则数据项是可调整的。于是,处理模块MTR_CO被初始配置为不调整:

[0106] 位于第一非零数据项之前的零数据项或零数据,通过这样的方式解码器不会将隐藏符号的值分配给该零数据,

[0107] 并且由于计算复杂度的原因,位于最后的非零数据项之后的零数据项或零数据。

[0108] 该调整操作由图2的处理模块MTR_CO执行。

[0109] 在建议的示例性实施例中,假设列表D₁中的数据总和等于5且因此是奇数。为了使解码器能够重构分配给第一非零数据项a₂的正符号而不用编码器C0将该数据项发送到解

码器,总和的奇偶性必须变成偶数。因此,在所述步骤S5的过程中,处理模块MTR_CO测试列表D₁中的数据的各种调整,所有的目标都是改变数据之和的奇偶性。在优选实施例中,+1或-1被加到每个可调整数据项,并根据预定的标准从执行的那些调整中选择调整。

[0110] 在步骤S5完成时,于是得到调整的列表 $D_{m1} = (a'_1, a'_2, \dots, a'_p)$ 。

[0111] 需要注意的是,在该步骤的过程中,特定的调整是被禁止的。于是,在第一非零数据项等于+1的情形中,不可能将它加上-1,因为它会变成零且然后会丢失它的列表D₁中第一非零数据项的特征。解码器于是随后将被解码的符号(通过计算数据之和的奇偶性)分配给另一数据项,且然后将会出现解码错误。

[0112] 然后经过对上述列表D_{m1}中除了第一非零数据项a₂的正符号以外的数据进行熵编码的步骤S20,该符号被隐藏在数据之和的奇偶校验中。

[0113] 需要注意的是,列表D₁或已调整的列表D_{m1}中的数据的幅度的集合在除了如上解释的未被编码的第一非零数据项的符号以外的符号的集合之前被编码。

[0114] 在图1所示的下列步骤S30的过程中,图2中的编码模块MC_CO测试已编码的当前块是否是图像IE中的最后一块。

[0115] 如果当前块是图像IE中的最后一块,则在图1所示的步骤S40的过程中,编码方法结束。

[0116] 如果不是该情形,则选择下一块B_i,然后根据前述光栅扫描遍历的顺序通过反复执行步骤S1到S20来对 $1 \leq i \leq Z$ 的块B_i进行编码。

[0117] 一旦已经实现所有块B₁到B_Z的熵编码,则构造以二进制形式来表示所述已编码的块的信号F。

[0118] 二进制信号F的构造在图2所示的流构造软件模块CF中实现。

[0119] 然后通过通信网络(未示出)将流F发送到远端。所述远端包括解码器,将在下文的描述中进一步详细描述该解码器。

[0120] 现在将主要参考图1来描述本发明的另一实施例。

[0121] 该另一实施例与前一个实施例的区别仅在于所要隐藏的符号的数量为N,其中N是大于等于2的整数。

[0122] 为此,函数f是列表D₁中的数据之和的模 2^N 的余数。假设在所建议的实例中 $N=2$,要隐藏的两个符号是列表D₁中的前两个非零数据的前两个符号例如a₂和a₃。

[0123] 在图1所示的步骤S4的过程中,处理模块MTR_CO验证N个符号的配置即 2^N 个可能的配置是否对应于列表D₁中的数据之和的模 2^N 的余数的值。

[0124] 在所建议的 $N=2$ 的实例中,存在 $2^2=4$ 个不同的符号配置。

[0125] 这四个配置遵循编码器C0的规则,该规则例如以如下方式确定:

[0126] -等于零的余数对应于两个连续的正符号:+,+;

[0127] -等于1的余数对应于连续的正符号和负符号:+,-;

[0128] -等于2的余数对应于连续的负符号和正符号:-,+;

[0129] -等于3的余数对应于两个连续的负符号:-,-。

[0130] 如果N个符号的配置对应于列表D₁中的数据之和的模 2^N 的余数的值,则对上述列表D₁中除了前两个非零数据a₂和a₃的符号以外的数据进行熵编码的步骤S20,所述符号被隐藏在列表D₁中的数据之和的模 2^N 的奇偶校验中。

[0131] 如果不是该情形,则对列表 D_1 中的至少一个可调整数据项进行调整的步骤S5。该调整由图2所示的处理模块MTR_C0以如下这样的方式执行:列表 D_1 中的可调整数据之和的模 2^N 的余数达到要隐藏的两个符号中的每个符号的值。

[0132] 由此得到已调整的列表 $D_{m1} = (a'_1, a'_2, \dots, a'_p)$ 。

[0133] 然后对上述列表 D_{m1} 中的除了第一非零数据项 a_2 的符号和第二非零数据项 a_3 的符号以外的数据进行熵编码的步骤S20,所述符号被隐藏在数据的模 2^N 之和和奇偶校验中。

[0134] 现在将描述根据本发明的特定实施例,其中,根据本发明的编码方法仍然被用于根据二进制流来对图像序列进行编码,该二进制流与通过根据H.264/MPEG-4AVC标准来编码而获得的二进制流接近。在该实施例中,根据本发明的编码方法例如以软件或硬件形式通过对初始符合H.264/MPEG-4AVC标准的编码器的调整来实现。

[0135] 根据本发明的编码方法以图3所示的包含步骤C1到C40的算法的形式来表示。

[0136] 根据本发明的实施例,编码方法在图4所示的实施例中的编码设备或编码器C01中实现。

[0137] 如图4所示,根据本发明,并且如前面实例所描述的,在适当的编码之前,以预定顺序将待编码的图像序列中的图像IE划分为多个(Z 个)分区 $B'_1, B'_2, \dots, B'_i, \dots, B'_Z$ 。

[0138] 在图4所示的实例中,所述分区是具有正方形形状并且全部具有相同尺寸的块。根据图像的尺寸,该尺寸不一定是块尺寸的倍数,最左边的块以及最底部的块可以不是正方形。在替代的实施例中,块例如可以是矩形尺寸和/或可以不互相对齐。

[0139] 每个块或每个宏块本身可以进一步被分为子块,子块自己可以再细分。

[0140] 这样的划分是通过图4所示的分区软件模块PC01来执行的,该分区软件模块PC01与图2所示的划分模块PC0相同。

[0141] 在所述划分步骤之后,对所述图像IE的每个当前分区 B'_i (其中 i 是大于等于1且小于等于 Z 的整数)进行编码。

[0142] 在图4所示的实例中,对当前图像IE的块 B'_1 到 B'_Z 中的每个块相继实施这样的编码。例如根据诸如本领域技术人员众所周知的“光栅扫描”遍历之类的遍历来对块进行编码。

[0143] 根据本发明的编码在如图4所示的编码器C01的编码软件模块MC_C01中实现。

[0144] 在图3所示的步骤C1的过程中,图4的编码模块MC_C01选择当前图像IE的待编码的第一块 B'_1 作为当前块 B'_i 。如图4所示,这是图像IE左边的第一块。

[0145] 在图3所示的步骤C2的过程中,通过已知的帧内预测和/或帧间预测技术对当前块 B'_1 进行预测编码,在该预测编码过程中,相对于至少一个之前被编码和解码的块来预测块 B'_1 。该预测由图4所示的预测软件模块PRED_C01来执行。

[0146] 毋庸赘述,诸如H.264标准所建议的模式之类的其它帧内预测模式是可能的。

[0147] 当前块 B'_1 还可经过帧间模式的预测编码,在该预测编码过程中,相对于来自之前被编码和解码的图像的块来预测当前块。当然可以想到其它类型的预测。在可能用于当前块的预测中,根据本领域技术人员众所周知的比特率失真标准来选择最优预测。

[0148] 上述预测编码步骤可构造已预测的块 B'_{p1} ,其与当前块 B'_1 近似。与该预测编码相关的信息将被写入要发送到解码器的信号中。该信息特别包括预测类型(帧内预测或帧间预测),并且如果合适的话,包括帧内预测模式、块或宏块(如果宏块已被细分)的分区类型、

参考图像索引以及在帧间预测模式中使用的位移矢量。该信息被编码器C01压缩。

[0149] 在图3所示的下一步骤C3的过程中,预测模块PRED_C01将与当前块 B'_1 相关的数据与已预测的块 B'_{p1} 的数据进行比较。更具体而言,在该步骤的过程中,按照传统从当前块 B'_1 减去已预测的块 B'_{p1} 以产生残留块 B'_{r1} 。

[0150] 在图3所示的下一步骤C4的过程中,根据传统的直接变换操作例如离散余弦变换DCT来变换残留块 B'_{r1} ,以产生已变换的块 B'_{t1} 。该操作由图4所示的变换软件模块MT_C01来执行。

[0151] 在图3所示的下一步骤C5的过程中,根据传统的量化操作例如标量量化来量化已变换的块 B'_{t1} 。然后得到已量化系数的块 B'_{q1} 。该步骤通过图4所示的量化软件模块MQ_C01来执行。

[0152] 在图3所示的下一步骤C6的过程中,以预定顺序来遍历块 B'_{q1} 的量化系数。在所示实例中,这需要传统的之字形遍历。该步骤由图4所示的读取软件模块ML_C01来执行。在步骤C6完成后,得到系数的一维列表 $E_1 = (\epsilon_1, \epsilon_2, \dots, \epsilon_L)$,更一般地被称为“量化残留”,其中L是大于或等于1的整数。列表 E_1 中的每个系数与要经过熵编码的各种数字信息项相关联。下面通过实例来描述这样的数字信息项。

[0153] 假设在所示的实例中, $L=16$ 并且列表 E_1 包括下列16个系数: $E_1 = (0, +9, -7, 0, 0, +1, 0, -1, +2, 0, 0, +1, 0, 0, 0, 0)$ 。

[0154] 在该实例中:

[0155] -针对位于列表 E_1 中的最后非零系数之前的每个系数,数字信息项例如比特将被熵编码,以表示系数是否为零。如果系数为零,则它例如是将被编码的值为0的比特,而如果系数不为零,则它是将被编码的值为1的比特。

[0156] -针对每个非零系数 $+9, -7, +1, -1, +2, +1$,数字信息项例如比特要被熵编码,以表示系数的绝对值是否等于1:如果它等于1,则它例如是将被编码的值为1的比特,而如果它不等于1,则它是将被编码的值为0的比特。

[0157] -针对绝对值不等于1且位于最后一个非零系数之前的每个非零系数,例如值为 $+9, -7, +2$ 的系数,幅度信息项(其系数的绝对值减去了值2)被熵编码;

[0158] -针对每个非零系数,分配给它的符号被诸如被设置为“0”的比特的数字信息项编码(用于+符号)或被诸如被设置为“1”的比特的数字信息项编码(用于-符号)。

[0159] 现在将参考图3描述根据本发明的特定编码步骤。

[0160] 根据本发明,决定避免对至少一个上述数据信息项即列表 E_1 中的所述系数中的一个的至少一个符号进行熵编码。

[0161] 为此,在图3所示的步骤C7的过程中,选择在随后的熵编码步骤的过程中所要隐藏的符号的数量。该步骤由图4所示的处理软件模块MTR_C01来执行。

[0162] 在优选实施例中,所要隐藏的符号的数量是1或0。此外,根据所述优选实施例,所要隐藏的是第一非零系数的符号。因此在所示实例中,它需要隐藏系数 $\epsilon_2 = +9$ 的符号。

[0163] 在替代的实施例中,所要隐藏的符号的数量是0、1、2、3或更多。

[0164] 根据步骤C7的优选实施例,在图3所示的第一子步骤C71的过程中,基于所述列表 E_1 来确定包含可被调整的系数的子列表 SE_1 即 $\epsilon'_1, \epsilon'_2, \dots, \epsilon'_M$,其中 $M < L$ 。在下文的描述中该系数将被称为可调整的系数。

[0165] 根据本发明,如果系数的量化值的调整不会引起解码器的任意去同步化,一旦该调整的系数被解码器处理,则该系数是可调整的。于是,处理模块MTR_C01被初始配置以不调整:

[0166] -位于第一非零系数之前的零系数或多个零系数,通过这种方式解码器不会将隐藏符号的值分配给所述零系数所述多个零系数,

[0167] -以及由于计算复杂度的原因,位于最后的非零系数之后的零系数或多个零系数。

[0168] 在所实例中,在子步骤C71结束后,得到的子列表SE₁由此是SE₁=(9,-7,0,0,1,0,-1,2,0,0,1)。因此,得到11个可调整的系数。

[0169] 在图3所示的下一子步骤C72的过程中,处理模块MTR_C01将可调整的系数的数量和预定阈值TSIG进行比较。在优选实施例中,TSIG等于4。

[0170] 如果可调整的系数的数量小于阈值TSIG,则在图3所示的步骤C20的过程中,对列表E₁中的系数进行传统熵编码,例如在图4中的标记CE_C01所表示的CABAC编码器中所实现的传统熵编码。为此,列表E₁中的每个非零系数的符号被熵编码。

[0171] 如果可调整的系数的数量大于阈值TSIG,则在图3所示的步骤C8的过程中,处理模块MTR_C01计算函数f的值,其表示子列表SE₁中的系数。

[0172] 在要发送到解码器的信号中隐藏单个符号的优选实施例中,函数f是子列表SE₁中的系数之和的奇偶校验。

[0173] 在图3所示的步骤C9的过程中,借助之前在编码器C01定义的规则,处理模块MTR_C01验证要隐藏的符号的值的奇偶性是否对应于子列表SE₁中的系数之和的奇偶性。

[0174] 在所建议的实例中,所述规则是:正符号与值等于零的比特相关联,而负符号与值等于1的比特相关联。

[0175] 根据在根据本发明的编码器C01中采用的规则,如果符号为正,则由此对应于值为零的编码比特,并且如果子列表SE₁中的系数之和是偶数,则对上述列表E₁中除了系数ε₂的符号以外的系数进行熵编码的步骤C20,。

[0176] 仍然根据在根据本发明的编码器C01中采用的规则,如果符号为负,则由此对应于值为1的编码比特,且如果子列表SE₁中的系数之和为奇数,则也对上述列表E₁中除了系数ε₂的符号以外的系数进行熵编码的步骤C20。

[0177] 根据在根据本发明的解码器C01中采用的规则,如果符号为正,则由此对应于值为零的编码比特,并且如果子列表SE₁的系数之和是奇数,则在图3所示的步骤C10的过程中,调整子列表SE₁中的至少一个可调整的系数。

[0178] 仍然根据在根据本发明的解码器C01中采用的规则,如果符号为负,则由此对应于值为1编码比特,并且如果子列表SE₁中的系数之和是偶数,则也进行对子列表SE₁中的至少一个可调整的系数的调整的步骤C10。

[0179] 该调整操作由图4的处理模块MTR_C01执行。

[0180] 在SE₁=(+9,-7,0,0,+1,0,-1,+2,0,0,+1)的示例性实施例中,系数的总和等于5且因此是奇数。为了使解码器能够重构分配给第一非零系数ε₂=+9的正符号而不用编码器C01将该系数发送到解码器,必须将总和的奇偶性变成偶数。因此,在所述步骤C10的过程中,处理模块MTR_C01测试子列表SE₁中的系数的各种调整,所有的目标都是改变系数之和的奇偶性。在优选实施例中,+1或-1被加到每个可调整的系数,并从被执行的所有那些调整

中选择调整。

[0181] 在优选实施例中,这样的选择构成根据性能标准的最优预测,该性能标准例如是本领域技术人员众所周知的比特率失真标准。该标准由下列等式(1)来表示:

[0182] (1) $J = D + \lambda R$

[0183] 其中D表示原始宏块和重构宏块之间的失真,R表示编码信息的比特的编码成本,并且 λ 表示拉格朗日倍数,其值可在编码之前被固定。

[0184] 在所建议的例子中,根据上述比特率失真标准带来最优预测的调整是将值1加到子列表SE₁中的第二系数-7。

[0185] 在步骤C10完成后,于是得到已调整的子列表SE_{m1} = (+9, -6, 0, 0, +1, 0, -1, +2, 0, 0, +1)。

[0186] 应该注意的是,在该步骤的过程中,特定的调整是被禁止的。于是,在第一非零系数 ϵ_2 等于+1的情形下,不可能将该系数加上-1,因为它将变成零,并且然后该系数会丢失列表E₁中的第一非零系数的特征。然后,解码器随后将被解码的符号(通过对系数之和的奇偶性进行计算)分配给另一系数,且然后将会出现解码错误。

[0187] 在图3所示的步骤C11的过程中,处理模块MTR_CO1对列表E₁进行相应的调整。然后得到下列已调整的列表Em₁ = (0, +9, -6, 0, 0, +1, 0, -1, +2, 0, 0, +1, 0, 0, 0, 0)。

[0188] 然后进行对上述列表Em₁中除了系数 ϵ_2 的符号以外的系数的熵编码的步骤C20,在所建议的例子中的系数 ϵ_2 的符号是系数9的+符号,该符号被隐藏在系数之和的奇偶校验中。

[0189] 应注意的是,列表E₁或已调整的列表Em₁中的系数的幅度的集合在除了如上解释的未被编码的第一非零系数 ϵ_2 的符号以外的符号集合之前被编码。

[0190] 在图3所示的下一步骤C30的过程中,图4中的编码模块MC_CO1测试被编码的当前块是否是图像IE中的最后一块。

[0191] 如果当前块是图像IE中的最后一块,则在图3所示的步骤C40的过程中,编码方法结束。

[0192] 如果不是该情形,则选择下一块B'_i,然后根据前述光栅扫描遍历的顺序通过反复执行步骤C1到C20来对 $1 \leq i \leq Z$ 的块B'_i进行编码。

[0193] 一旦已经实现所有块B'₁到B'_Z的熵编码,则构造以二进制形式来表示所述已编码的块的信号F'。

[0194] 二进制符号F'的构造在例如如图4所示的流构造软件模块CF1中实现。

[0195] 然后通过通信网络(未示出)将流F'发送到远端。该远端包括解码器,将在下文的描述中进一步详细描述该解码器。

[0196] 现在将主要参考图3来描述本发明的另一实施例。

[0197] 该另一实施例与前一个实施例的区别仅在于所要隐藏的系数的数量为0或N,其中N是大于等于2的整数。

[0198] 为此,前述比较子步骤C72被图3中的虚线表示的子步骤C72a替换,在子步骤C72a的过程中,将可调整的系数的数量与若干个预定阈值 $0 < \text{TSIG}_1 < \text{TSIG}_2 < \text{TSIG}_3 \cdots$ 进行比较,通过这种方式,如果可调整的系数的数量在TSIG_N与TSIG_{N+1}之间,则N个符号要被隐藏。

[0199] 如果可调整的系数的数量小于第一阈值TSIG₁,则在上述步骤C20的过程中,对列表E₁中的系数进行传统熵编码。为此,列表E₁中的每个非零系数的符号被熵编码。

[0200] 如果可调整的系数的数量在阈值TSIG_N与TSIG_{N+1}之间,则在图3所示的步骤C8的过程中,处理模块MTR_C01计算函数f的值,其表示子列表E₁中的系数。

[0201] 在所述另一实施例中,在解码器上决定隐藏N个符号,函数f是子列表SE₁中的系数之和的模2^N的余数。假设在所建议的例子中N=2,要隐藏的两个符号分别是前两个非零系数即ε₂和ε₃的前两个符号。

[0202] 在图3所示的下一步骤C9的过程中,处理模块MTR_C01验证N个符号的配置即2^N个可能的配置是否对应于子列表SE₁中的系数之和的模2^N的余数的值。

[0203] 在所建议的N=2的实例中,存在2²=4个不同的符号配置。

[0204] 这四个配置遵循编码器C01的规则,该规则例如以如下方式确定:

[0205] -等于零的余数对应于两个连续的正符号:+,+;

[0206] -等于1的余数对应于连续的正符号和负符号:+,-;

[0207] -等于2的余数对应于连续的负符号和正符号:-,+;

[0208] -等于3的余数对应于两个连续的负符号:-,-。

[0209] 如果N个符号的配置对应于子列表SE₁中的系数之和的模2^N的余数的值,则对上述列表E₁中除了系数ε₂和系数ε₃的符号以外的系数进行熵编码的步骤C20,所述符号被隐藏在系数之和的模2^N的奇偶校验中。

[0210] 如果不是该情形,则对子列表SE₁中的至少一个可调整的系数进行调整的步骤C10。该调整由图4所示的处理模块MTR_C01以如下这样的方式执行:子列表SE₁中的可调整的系数之和的模2^N的余数达到要隐藏的两个符号中的每个符号的值。

[0211] 在前述步骤C11的过程中,处理模块MTR_C01对列表E₁进行相应的调整。由此得到已调整的列表E_{m1}。

[0212] 然后进行对上述列表E_{m1}中除了系数ε₂和系数ε₃的符号以外的系数的熵编码的步骤C20,该符号被隐藏在系数之和的模2^N的奇偶校验中。

[0213] 解码部分的详细描述

[0214] 现在将描述根据本发明的解码方法的一般实施例,其中,解码方法以软件或硬件形式通过对初始符合H.264/MPEG-4AVC标准的解码器的调整来实现。

[0215] 根据本发明的解码方法以图5所示的包含步骤SD1到SD7的算法的形式来表示。

[0216] 根据本发明的一般实施例,根据本发明的解码方法在如图6所示的解码设备或解码器D0中实现,其适于接收通过图2中的编码器C0传递的流F。

[0217] 在图5中未示出的预备步骤的过程中,在接收到的数据信号F中,识别之前已由编码器C0编码的分区B₁到B_z。在优选实施例中,所述分区是具有正方形形状并且全部具有相同尺寸的块。根据图像的尺寸,该尺寸不一定是块尺寸的倍数,最左边的块以及最底部的块可以不是正方形。在替代的实施例中,块例如可以是矩形尺寸和/或可以不互相对齐。

[0218] 每个块或每个宏块本身可以进一步被分为子块,子块自己可以再细分。

[0219] 该识别由如图6所示的流分析软件模块EX_D0来执行。

[0220] 在图5所示的步骤SD1的过程中,图6中的模块EX_D0选择待编码的第一块B₁作为当前块B_i。该选择例如包括在在第一块B₁的数据开始时在信号F中放置读指针。

- [0221] 然后对选中的已编码的块中的每个块进行解码。
- [0222] 在图5所示的实例中,对已编码的块 B_1 到 B_Z 中的每个块相继实施这样的解码。例如根据本领域技术人员众所周知的“光栅扫描”遍历来对块进行解码。
- [0223] 根据本发明的解码在如图6所示的解码器D0的解码软件模块MD_D0中实现。
- [0224] 在图5所示的步骤SD2的过程中,首先对已选择的第一当前块 B_1 进行熵解码。该操作由图6所示的例如CABAC类型的熵解码模块DE_D0来执行。在该步骤的过程中,模块DE_D0对与列表 D_1 或已调整的列表 D_{m1} 中每个被编码的数据的幅度相对应的数字信息项执行熵解码。此时,只有列表 D_1 或已调整的列表 D_{m1} 中的数据符号未被解码。
- [0225] 在处理模块MTR_D0接收到列表 $D_1 = (a_1, a_2, \dots, a_P)$ 的情形中,在图5所示的步骤SD3的过程中,对列表 D_1 中的数据的所有符号进行传统的熵解码。该解码由图6中的标记DE_D0所表示的CABAC解码器来执行。为此,列表 D_1 中的每个非零数据项的符号被熵解码。
- [0226] 在处理模块MTR_D0接收到调整的列表 $D_{m1} = (a'_1, a'_2, \dots, a'_P)$ 的情形中,在所述步骤SD3的过程中,对列表 D_{m1} 中的数据除了第一非零数据项 a_2 的符号以外所有符号进行传统的熵解码。
- [0227] 在图5所示的步骤SD4的过程中,处理模块MTR_D0计算表示列表 D_{m1} 的数据的函数 f 的值,以确定所计算的值是偶数还是奇数。
- [0228] 在信号F中隐藏单个符号的优选实施例中,函数 f 是列表 D_{m1} 中的数据之和的奇偶校验。
- [0229] 根据在编码器C0上使用的与解码器D0上相同的规则,列表 D_{m1} 中的数据之和的偶数值表示已调整的列表 D_{m1} 中的第一非零数据项的符号是正的,而列表 D_{m1} 中的数据之和的奇数值表示已调整的列表 D_{m1} 中的第一非零数据项的符号是负的。
- [0230] 在示例性实施例中,数据的总和是偶数。于是,在步骤SD4结束后,处理模块MTR_D0由此推断第一非零数据项 a_2 的隐藏符号是正的。
- [0231] 在图5所示的步骤SD5的过程中,构造已解码的块 BD_1 。该操作由图6所示的重构软件模块MR_D0来执行。
- [0232] 在图5所示的步骤SD6的过程中,解码模块MD_D0测试已解码的当前块是否为在信号F中识别的最后一块。
- [0233] 如果当前块是信号F中的最后一块,则在图5所示的步骤SD7的过程中,解码方法结束。
- [0234] 如果不是该情形,则选择下一块 B_i ,根据前述光栅扫描遍历的顺序通过反复执行步骤SD1到SD5来对 $1 \leq i \leq Z$ 的块 B'_i 进行解码。
- [0235] 现在将主要参考图5来描述本发明的另一实施例。
- [0236] 该另一实施例与前一个实施例的区别仅在于所隐藏的符号的数量现在等于 N , N 是大于等于2的整数。
- [0237] 为此,在前述步骤SD3的过程中,对列表 D_{m1} 中的数据除了所述已调整的列表 D_{m1} 中的前几个非零数据的 N 个各个符号以外的所有符号进行传统的熵解码,所述 N 个符号被隐藏。
- [0238] 在该另一实施例中,处理模块MTR_D0在步骤SD4的过程中计算函数 f 的值,其是列表 D_{m1} 中的数据之和的模 2^N 的余数。假设在所建议的实例中 $N=2$ 。

[0239] 然后,处理模块MTR_D0由此根据在编码时所使用的规则来推断被分别分配给前两个非零数据a2和a3中的每个非零数据的两个隐藏符号的配置。

[0240] 一旦这两个符号已被重构,则执行上述步骤SD5到SD7。

[0241] 现在将描述根据本发明的解码方法的特定实施例,其中,解码方法以软件或硬件形式通过对初始符合H.264/MPEG-4AVC标准的解码器的调整来实现。

[0242] 根据本发明的解码方法以包含图7所示的步骤D1到D12的算法的形式来表示。

[0243] 根据本发明的实施例,根据本发明的解码方法在如图8所示的解码设备或解码器D01中实现,其能够处理通过图4中的编码器C01传递的信号F'。

[0244] 在图7中未示出的预备步骤的过程中,在接收到的数据信号F'中,识别之前已由编码器C01编码的分区B'1到B'z。在优选实施例中,所述分区是具有正方形形状并且全部具有相同尺寸的块。根据图像的尺寸,该尺寸不一定是块大小的倍数,最左边的块以及最底部的块可以不是正方形。在替代的实施例中,块例如可以是矩形尺寸和/或可以不互相对齐。

[0245] 每个块或每个宏块本身可以进一步被分为子块,子块自己可以再细分。

[0246] 该识别由如图8所示的流分析软件模块EX_D01来执行。

[0247] 在图7所示的步骤D1的过程中,图8中的模块EX_D01选择待编码的第一块B'1作为当前块B'i。该选择例如包括在第一块B'1的数据开始时在信号F'中放置读指针。

[0248] 然后对选中的已编码的块中每个块进行解码。

[0249] 在图7所示的实例中,对已解码的块B'1到B'z中的每个块相继实施这样的解码。例如根据本领域技术人员众所周知的“光栅扫描”遍历来对块进行解码。

[0250] 根据本发明的解码在如图8所示的解码器D01的解码软件模块MD_D01中实现。

[0251] 在图7所示的步骤D2的过程中,首先对已选择的第一当前块B'1进行熵解码。该操作由图8所示的例如CABAC类型的熵解码模块DE_D01来执行。在该步骤的过程中,模块DE_D01对与列表E1或已调整的列表Em1中每个被编码的系数的幅度相对应的数字信息执行熵解码。此时,只有列表E1或已调整的列表Em1中的系数的符号未被解码。

[0252] 在图7所示的步骤D3的过程中,确定在之前的熵编码步骤C20的过程中可能已被隐藏的符号的数量。该步骤D3由如图8所示的处理软件模块MTR_D01来执行。步骤D3类似于前述用于确定要被隐藏的符号的数量的步骤C7。

[0253] 在优选实施例中,隐藏的符号的数量是1或0。此外,根据所述优选实施例,被隐藏的是第一非零系数的符号。因此在所示实例中,它需要系数 $\epsilon_2 = +9$ 的正符号。

[0254] 在替代实施例中,隐藏的符号的数量是0、1、2、3或更多。

[0255] 根据步骤D3的优选实施例,在图7所示的第一子步骤D31的过程中,根据所述列表E1或根据已调整的列表Em1来确定包含在编码时已被调整的系数 $\epsilon'_1, \epsilon'_2, \dots, \epsilon'_M$ 的子列表,其中 $M < L$ 。

[0256] 该确定由与前述编码步骤C7相同的方式来执行。

[0257] 和前述处理模块MTR_C01类似,处理模块MTR_D01被初始配置以不调整:

[0258] -位于第一非零系数之前的零系数或多个零系数,

[0259] -以及由于计算复杂的原因,位于最后的非零系数之后的零系数或多个零系数。

[0260] 在所示实例中,在子步骤D31完成后,这需要子列表SEm1从而 $SEm1 = (9, -6, 0, 0, 1, 0, -1, 2, 0, 0, 1)$ 。因此,得到可能已被调整的11个系数。

[0261] 在图7所示的下一子步骤D32的过程中,处理模块MTR_D01将可能已被调整的系数的数量与预定阈值TSIG进行比较。在优选实施例中,TSIG等于4。

[0262] 如果可能已被调整的系数数量小于阈值TSIG,则在图7所示的步骤D4的过程中,对列表E₁中的系数的所有符号进行传统熵解码。该解码由图8中的标记DE_D01表示的CABAC解码器来执行。为此,列表E₁中的每个非零系数的符号被熵解码。

[0263] 如果可能已被调整的系数的数量大于阈值TSIG,则在所述步骤D4的过程中,对列表Em₁中的系数的除了第一非零系数 ϵ_2 的符号以外的所有符号进行传统熵解码。

[0264] 在图7所示的步骤D5的过程中,处理模块MTR_D01计算表示子列表SEm₁的系数的函数f的值,以确定所计算的值是偶数还是奇数。

[0265] 在信号F' 中隐藏单个符号的优选实施例中,函数f是子列表SEm₁中的系数之和的奇偶校验。

[0266] 根据在编码器C01上使用的与解码器D01上相同的规则,子列表SEm₁中的系数之和的偶数值表示已调整的列表Em₁中的第一非零系数的符号是正的,而子列表SEm₁中的系数之和的奇数值表示已调整的列表Em₁中的第一非零系数的符号是负的。

[0267] 在SEm₁ = (+9, -6, 0, 0, +1, 0, -1, +2, 0, 0, +1)的示例性实施例中,系数的总和等于6,且因此是偶数。于是,在步骤D5完成后,处理模块MTR_D01由此推断第一非零系数 ϵ_2 的隐藏符号是正的。

[0268] 在图7所示的步骤D6的过程中,并且借助于步骤D2、D4和D5的过程中所有重构的数字信息项,以预定的顺序来重构块B' q₁的量化系数。在所示实例中,这需要与前述编码步骤C6的过程中执行的之字形遍历相反的之字形遍历。该步骤由如图8所示的读取软件模块ML_D01执行。更具体地说,模块ML_D01使用所述相反的之字形的遍历顺序将列表E₁ (一维)的系数写到块B' q₁ (二维)。

[0269] 在图7所示的步骤D7的过程中,根据传统的反量化操作来对已量化的残留块B' q₁进行反量化,该操作是与编码时的前述步骤C5中执行的量化相反的操作,以生成已解码的反量化块BD' q₁。该步骤通过如图8所示的反量化软件模块MDQ_D01来执行。

[0270] 在图7所示的步骤D8的过程中,对反量化块BD' q₁进行逆变换,所述逆变换是与编码时的前述步骤C4中执行的直接变换相反的操作。然后得到已解码的残留块BD' r₁。该操作由图8所示的逆变换软件模块MTI_D01来执行。

[0271] 在图7所示的步骤D9的过程中,对当前块B₁进行预测解码。诸如所述预测解码这样的预测解码传统上在针对至少一个之前已解码的块来预测块B' ₁的过程中通过已知的帧内和/或帧间预测技术来执行。该操作由如图8所示的预测解码模块PRED_D01来执行。

[0272] 毋庸赘述,诸如H.264标准所建议的模式之类的其它帧内预测模式是可能的。

[0273] 在该步骤的过程中,借助于之前步骤中的被解码的语义元素来执行预测解码,且特别包括预测类型(帧间预测或帧内预测),并且如果合适的话,包括帧内预测模式、块或宏块(如果宏块已被细分)的分区类型、在帧间预测模式中使用的参考图像索引以及位移矢量。

[0274] 上述预测解码步骤可构造已预测的块B' p₁。

[0275] 在图7所示的步骤D10的过程中,通过将已解码的残留块BD' r₁加到已预测的块B' p₁来构造已解码的块BD' ₁。该操作由图8所示的重构软件模块MR_D01来执行。

[0276] 在图7所示的步骤D11的过程中,解码模块MD_D01测试已解码的当前块是否为在信号F' 中识别的最后一块。

[0277] 如果当前块是信号F' 中最后一块,则在图7所示的步骤D12的过程中,解码方法结束。

[0278] 如果不是该情形,则选择下一块B' _i,根据前述光栅扫描遍历的顺序通过反复执行步骤D1到D10来对 $1 \leq i \leq Z$ 的块B' _i进行解码。

[0279] 现在将主要参考图7来描述本发明的另一实施例。

[0280] 该另一实施例与前一个实施例的区别仅在于所隐藏的系数的数量为0或N,其中N是大于等于2的整数。

[0281] 为此,前述比较子步骤D32被图7中的虚线所示的子步骤D32a替换,在比较子步骤D32的过程中,将可能已被调整的系数的数量与若干个预定的阈值 $0 < \text{TSIG}_1 < \text{TSIG}_2 < \text{TSIG}_3 \cdots$ 进行比较,通过这种方式,如果所述系数的数量在 TSIG_N 与 TSIG_{N+1} 之间,则N个信号已被隐藏。

[0282] 如果所述系数的数量小于第一阈值 TSIG_1 ,则在前述步骤D4的过程中,对列表E₁中的系数的所有符号进行传统熵解码。为此,列表E₁中的每个非零系数的符号被熵解码。

[0283] 如果所述系数的数量在阈值 TSIG_N 与 TSIG_{N+1} 之间,则在前述步骤D4中,对列表E₁中的系数的除了所述调整列表Em₁中的第一非零系数的N个各个符号以外的所有符号进行传统熵解码,所述N个符号被隐藏。

[0284] 在该另一实施例中,处理模块MTR_D01在步骤D5的过程中计算函数f的值,其是子列表SEm₁中的系数之和的模 2^N 的余数。假设在所建议的实例中 $N=2$ 。

[0285] 然后,处理模块MTR_D01由此根据在解码时所使用的规则来推断被分别分配给前两个非零系数 ϵ_2 和 ϵ_3 中的每个非零数据的两个隐藏符号的配置。

[0286] 一旦这两个符号已被重构,则执行上述步骤D6到D12。

[0287] 毋庸赘述,本文所描述的实施例仅通过指示的方式给出而不是要限制,并且本领域技术人员可以容易地进行各种调整而不偏离本发明的范围。

[0288] 于是例如,根据关于图4所示的简化实施例,编码器C01可被配置为隐藏至少N' 个预定符号,其中, $N' \geq 1$,而不是0、1或N个预定符号。在该情形下,比较步骤C72或C72a将被移除。通过相应的方式,根据关于图8所示的简化实施例,解码器D01将被配置为重构N' 个预定符号而不是0、1或N个预定符号。在该情形下,比较步骤D32或D32a将被移除。此外,在编码步骤C72和解码步骤D32中应用的判断标准可被另一类型的标准替换。为此,代替与可调整的系数的数量或可能已被调整的系数的数量的阈值进行比较,处理模块MTR_C01或MTR_D01可以应用分别依赖于可调整或可能已被调整的系数的幅度之和的或者可调整或可能已被调整的系数中存在的零的数量的判断标准。

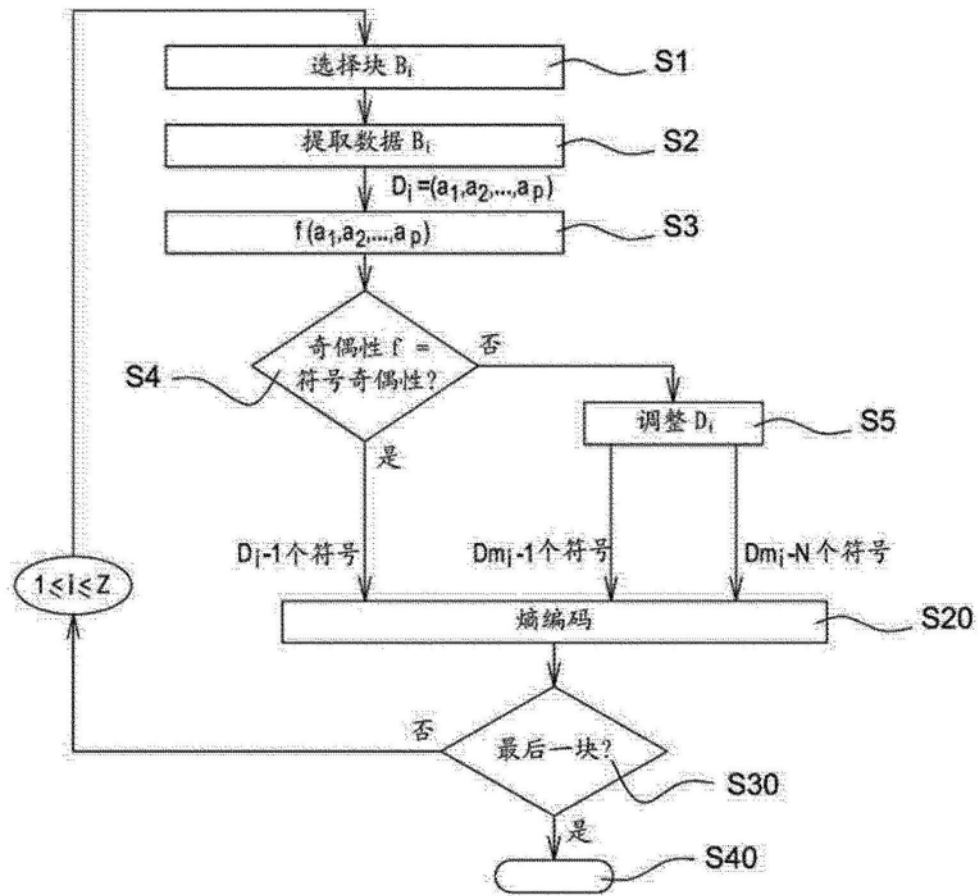


图1

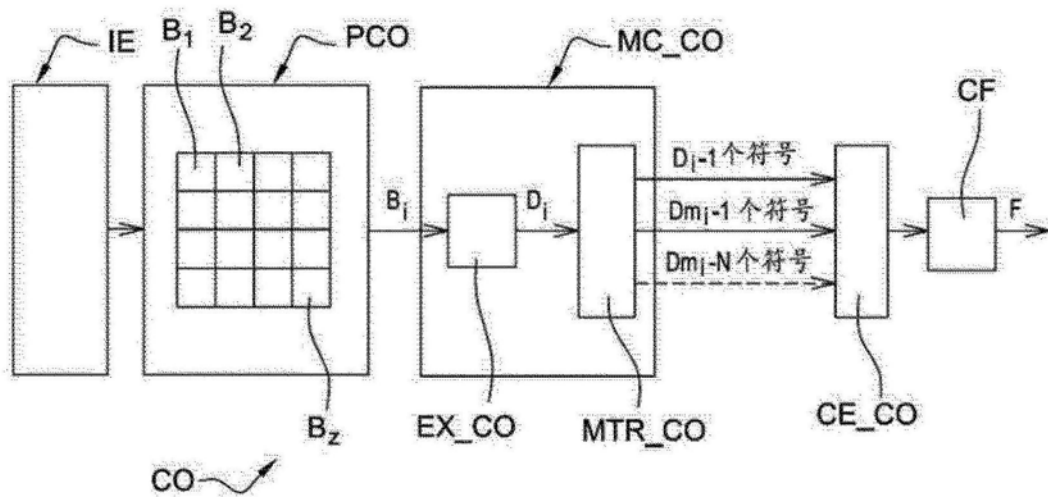


图2

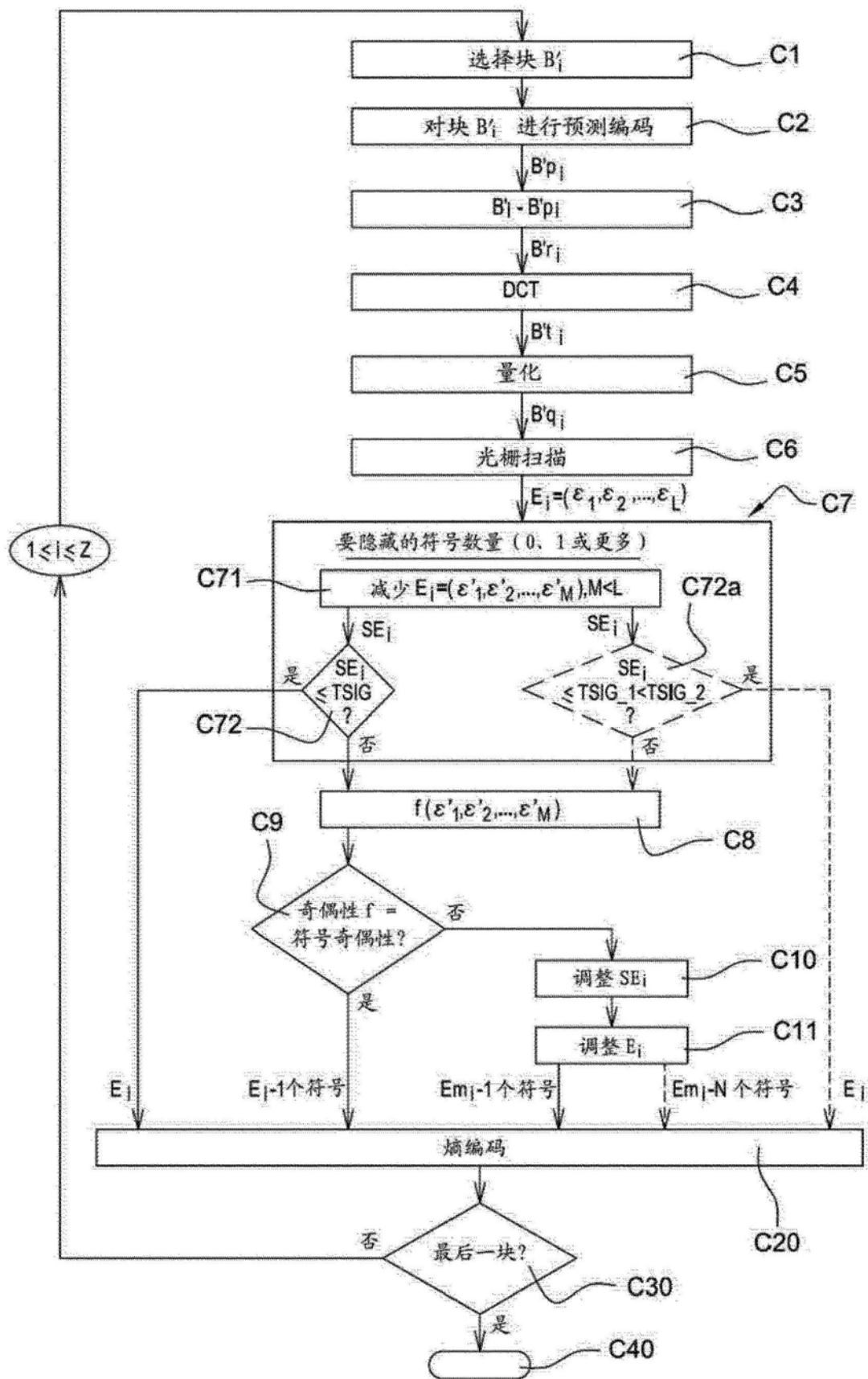


图3

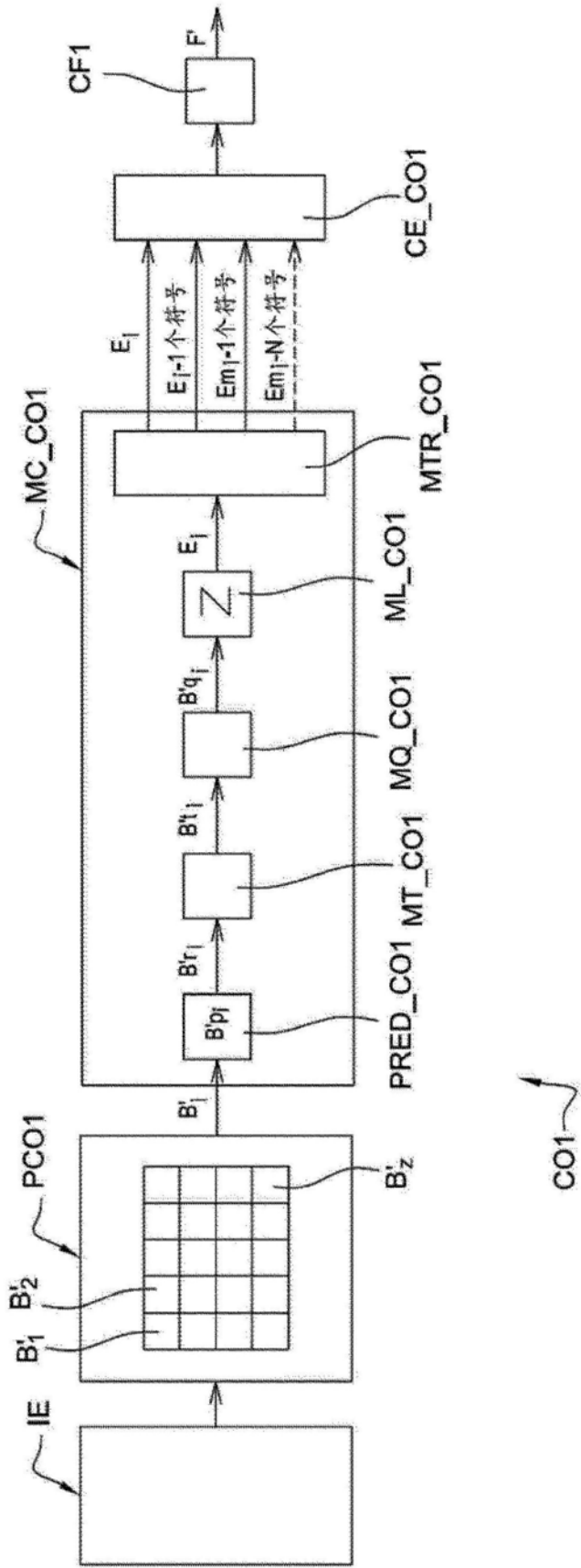


图4

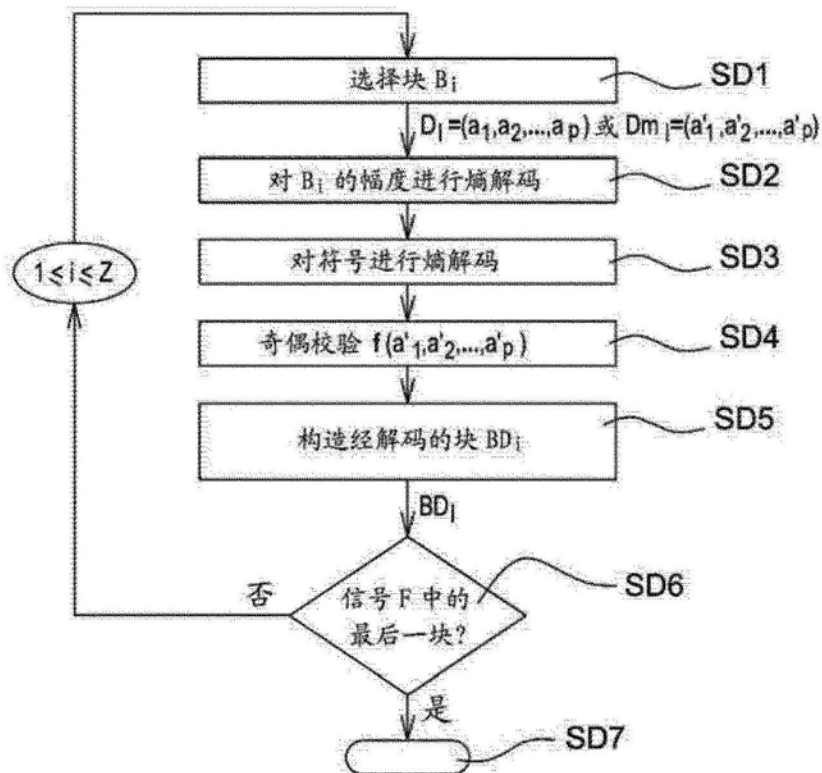


图5

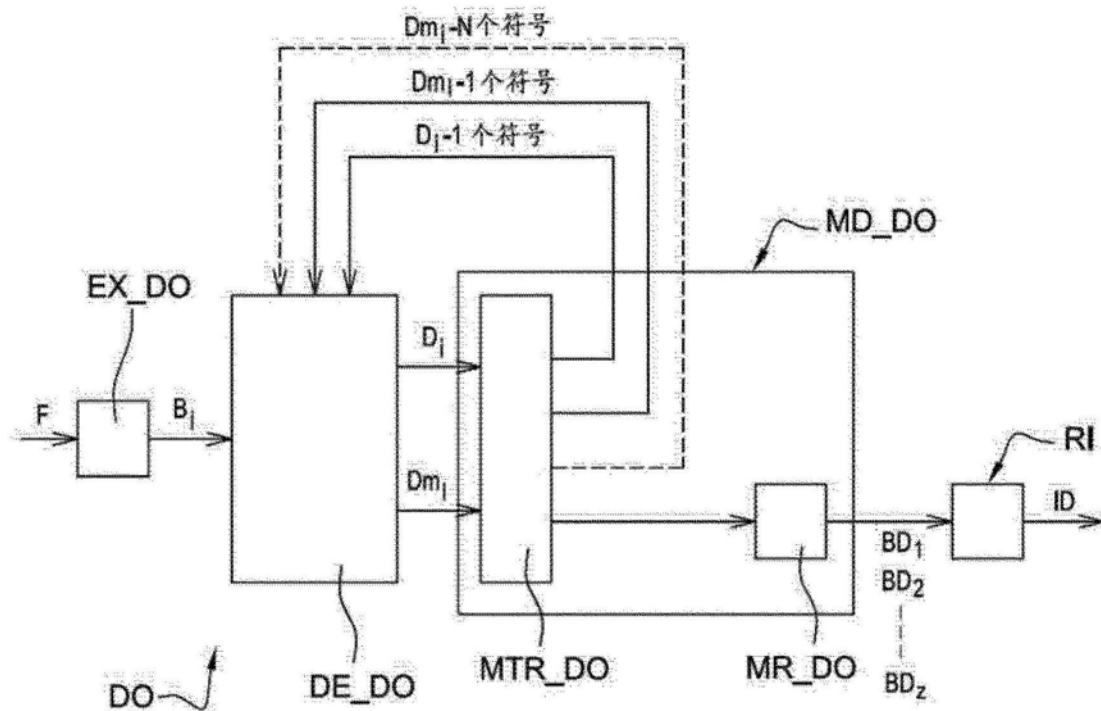


图6

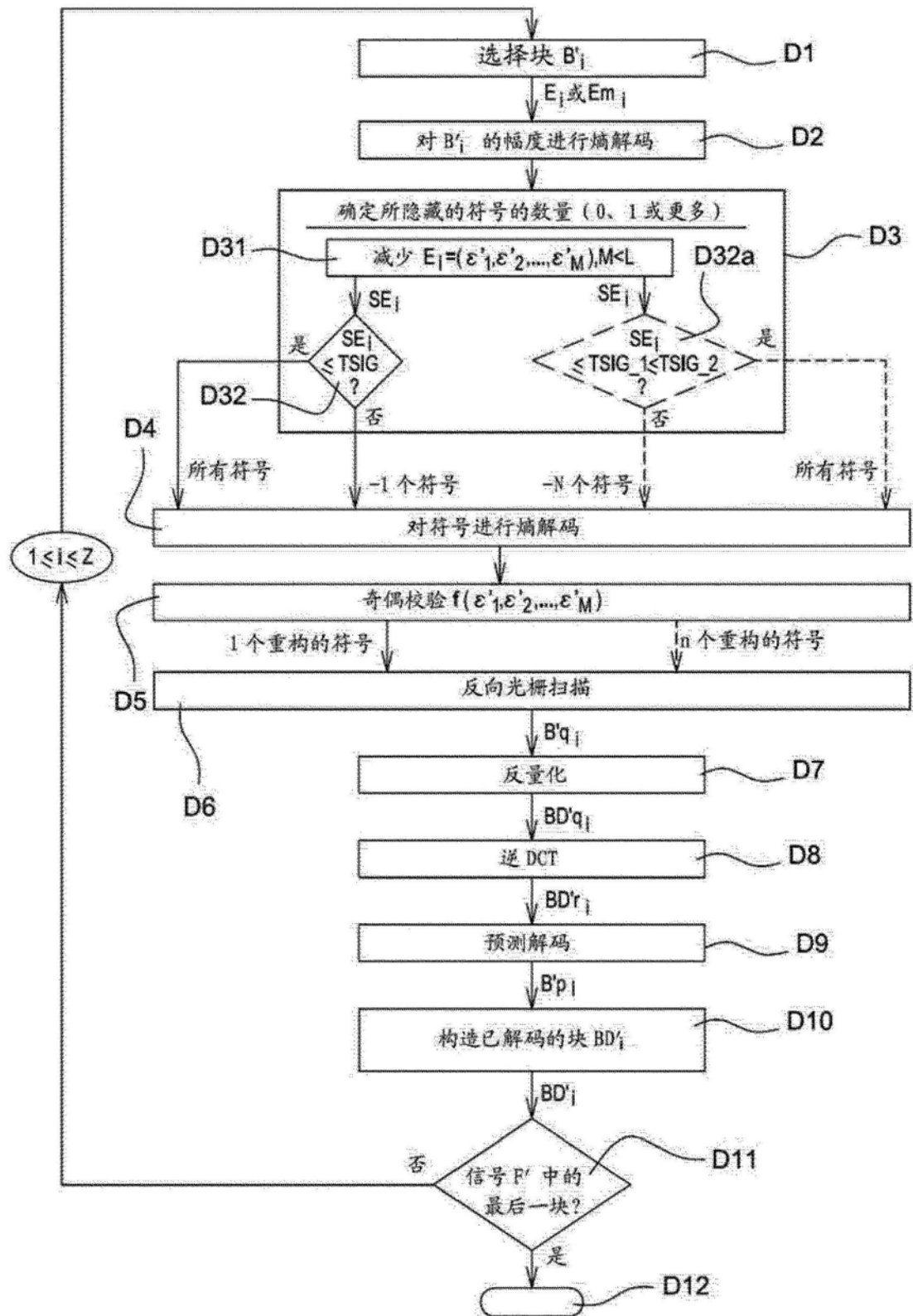


图7

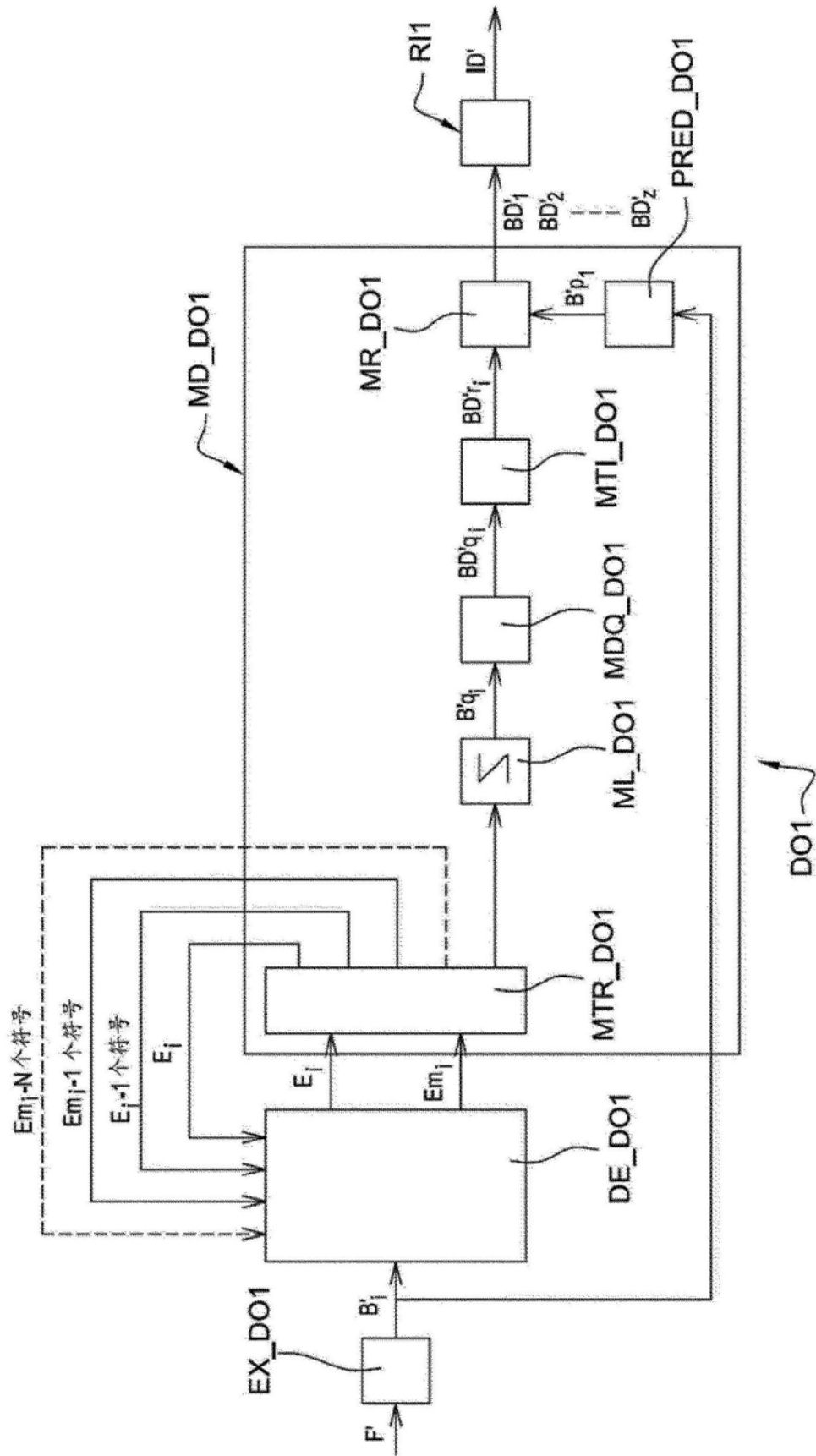


图8