(19)中华人民共和国国家知识产权局

(12)发明专利申请

(10)申请公布号 CN 105120277 A
(43)申请公布日 2015.12.02

(21)申请号 201510562762.5
(22)申请日 2012.06.20
(30)优先权数据
1155606 2011.06.24 FR
(62)分案原申请数据
201280031335.9 2012.06.20
(71)申请人 杜比国际公司
地址 瑞典斯德哥尔摩
(72)发明人 F·亨利 S·帕特克斯 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 宋岩
(51)Int.Cl.
H04N 19/13(2014.01)
H04N 19/174(2014.01)
H04N 19/176(2014.01)
H04N 19/196(2014.01)
H04N 19/436(2014.01)

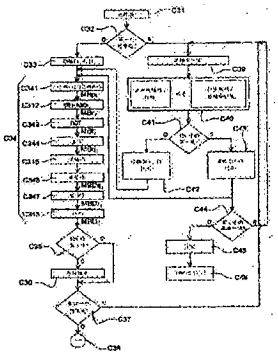
权利要求书3页 说明书16页 附图9页

(54)发明名称

图像编码和解码的方法、编码和解码设备以及计算机程序

(57)摘要

公开了用于对图像编码和解码的方法、编码和解码设备以及相应的计算机程序。本发明涉及一种编码方法，包括：从可以包含属于预定符号集的符号的多个块切割出图像，将所述块分组为预定数量的块子集，使用熵模块，通过将数字信息与所述子集中的每个块的符号进行组合来对所考虑的块子集中的每个进行编码。在当前块是所述块子集中被编码的第一块的情形下，所述第一当前块的符号出现的概率是针对至少一个其他子集的预定编码和解码块来确定的那些概率。在当前单元是所考虑的子集中最后编码的单元的情形下，在所考虑的所述子集中的块的编码期间与所述符号关联的所有数字信息被写入到子流中，该子流至少表示所考虑的所述子集，并实现初始化了步骤。



CN 105120277 A

1. 一种存储指令的计算机可读介质，所述指令在被一个或更多个处理器执行时使得所述一个或更多个处理器执行包括以下项的操作：

接收代表至少一个编码图像的数据流，该数据流包括：所述至少一个编码图像的子图像的多个上下文适应熵编码的子流，所述多个上下文适应熵编码的子流中的每个子流包括变换后的残留值的量化系数的连续块的行，

从数据流中识别所述多个上下文适应熵编码的子流；

对所述多个上下文适应熵编码的子流进行解码以获得相应的解码后的子图像；以及

基于解码后的子图像中的每个子图像来重构解码后的图像并且输出解码后的图像，

其中，所述多个上下文适应熵编码的子流中的子流在数据流中与子图像的光栅次序不同地排序。

2. 如权利要求 1 所述的计算机可读介质，其中，数据流进一步包括：

包括变换后的残留值的量化系数的连续块的第一行的第一上下文适应熵编码的子流；以及

包括变换后的残留值的量化系数的连续块的第二行的第二上下文适应熵编码的子流，在子图像的光栅次序中，连续块的第二行紧接在连续块的第一行之后，

其中，第二上下文适应熵编码的子流在数据流中被排序为与第一上下文适应熵编码的子流相距预定数量的上下文适应熵编码的子流。

3. 如权利要求 1 所述的计算机可读介质，其中，数据流进一步包括：

包括变换后的残留值的量化系数的连续块的第三行的第三上下文适应熵编码的子流，在子图像的光栅次序中，连续块的第三行紧接在连续块的第二行之后，

其中，第三上下文适应熵编码的子流在数据流中被排序为在第二上下文适应熵编码的子流之前。

4. 如权利要求 1 所述的计算机可读介质，其中，所述多个上下文适应熵编码的子流对应于在数据流的解码中使用的多个解码单元。

5. 如权利要求 1 所述的计算机可读介质，其中，所述多个上下文适应熵编码的子流包括代表子图像的光栅次序中的相应位置的指示符。

6. 如权利要求 1 所述的计算机可读介质，其中，数据流进一步包括：

包括所述多个上下文适应熵编码的子流的第一子集的第一子流组，其中第一子集中的每个上下文适应熵编码的子流具有在第一子图像的光栅次序中的位置；以及

包括所述多个上下文适应熵编码的子流的第二子集的第二子流组，其中第二子集中的每个上下文适应熵编码的子流具有在第二子图像的光栅次序中的位置，其中在数据流的次序中，第二子流组在第一子流组之后。

7. 一种用于解码代表至少一个编码图像的数据流的系统，包括：

一个或更多个处理器以及一个或更多个存储指令的存储设备，所述指令可操作为在被所述一个或更多个处理器执行时使得所述一个或更多个处理器执行包括以下项的操作：

接收所述数据流，所述数据流包括：所述至少一个编码图像的子图像的多个上下文适应熵编码的子流，所述多个上下文适应熵编码的子流中的每个子流包括变换后的残留值的量化系数的连续块的行，

从数据流中识别所述多个上下文适应熵编码的子流；

对所述多个上下文适应熵编码的子流进行解码以获得相应的解码后的子图像；以及

基于解码后的子图像中的每个子图像来重构解码后的图像并且输出解码后的图像，

其中，所述多个上下文适应熵编码的子流中的子流在数据流中与子图像的光栅次序不同地排序。

8. 如权利要求 7 所述的系统，其中，数据流进一步包括：

包括变换后的残留值的量化系数的连续块的第一行的第一上下文适应熵编码的子流；以及

包括变换后的残留值的量化系数的连续块的第二行的第二上下文适应熵编码的子流，在子图像的光栅次序中，连续块的第二行紧接在连续块的第一行之后，

其中，第二上下文适应熵编码的子流在数据流中被排序为与第一上下文适应熵编码的子流相距预定数量的上下文适应熵编码的子流。

9. 如权利要求 7 所述的系统，其中，数据流进一步包括：

包括变换后的残留值的量化系数的连续块的第三行的第三上下文适应熵编码的子流，在子图像的光栅次序中，连续块的第三行紧接在连续块的第二行之后，

其中，第三上下文适应熵编码的子流在数据流中被排序为在第二上下文适应熵编码的子流之前。

10. 如权利要求 7 所述的系统，其中，所述多个上下文适应熵编码的子流对应于在数据流的解码中使用的多个解码单元。

11. 如权利要求 7 所述的系统，其中，所述多个上下文适应熵编码的子流包括代表子图像的光栅次序中的相应位置的指示符。

12. 如权利要求 7 所述的系统，其中，数据流进一步包括：

包括所述多个上下文适应熵编码的子流的第一子集的第一子流组，其中第一子集中的每个上下文适应熵编码的子流具有在第一子图像的光栅次序中的位置；以及

包括所述多个上下文适应熵编码的子流的第二子集的第二子流组，其中第二子集中的每个上下文适应熵编码的子流具有在第二子图像的光栅次序中的位置，其中在数据流的次序中，第二子流组在第一子流组之后。

13. 一种用于解码代表至少一个编码图像的数据流的方法，包括：

接收所述数据流，所述数据流包括：所述至少一个编码图像的子图像的多个上下文适应熵编码的子流，所述多个上下文适应熵编码的子流中的每个子流包括变换后的残留值的量化系数的连续块的行，

从数据流中识别所述多个上下文适应熵编码的子流；

对所述多个上下文适应熵编码的子流进行解码以获得相应的解码后的子图像；以及

基于解码后的子图像中的每个子图像来重构解码后的图像并且输出解码后的图像，

其中，所述多个上下文适应熵编码的子流中的子流在数据流中与子图像的光栅次序不同地排序。

14. 如权利要求 13 所述的方法，其中，数据流进一步包括：

包括变换后的残留值的量化系数的连续块的第一行的第一上下文适应熵编码的子流；以及

包括变换后的残留值的量化系数的连续块的第二行的第二上下文适应熵编码的子流，

在子图像的光栅次序中，连续块的第二行紧接在连续块的第一行之后，

其中，第二上下文适应熵编码的子流在数据流中被排序为与第一上下文适应熵编码的子流相距预定数量的上下文适应熵编码的子流。

15. 如权利要求 13 所述的方法，其中，数据流进一步包括：

包括变换后的残留值的量化系数的连续块的第三行的第三上下文适应熵编码的子流，在子图像的光栅次序中，连续块的第三行紧接在连续块的第二行之后，

其中，第三上下文适应熵编码的子流在数据流中被排序为在第二上下文适应熵编码的子流之前。

16. 如权利要求 13 所述的方法，其中，所述多个上下文适应熵编码的子流对应于在数据流的解码中使用的多个解码单元。

17. 如权利要求 13 所述的方法，其中，所述多个上下文适应熵编码的子流包括代表子图像的光栅次序中的相应位置的指示符。

18. 如权利要求 13 所述的方法，其中，数据流进一步包括：

包括所述多个上下文适应熵编码的子流的第一子集的第一子流组，其中第一子集中的每个上下文适应熵编码的子流具有在第一子图像的光栅次序中的位置；以及

包括所述多个上下文适应熵编码的子流的第二子集的第二子流组，其中第二子集中的每个上下文适应熵编码的子流具有在第二子图像的光栅次序中的位置，其中在数据流的次序中，第二子流组在第一子流组之后。

图像编码和解码的方法、编码和解码设备以及计算机程序

[0001] 本申请是申请号为201280031335.9，申请日为2012年6月20日，题为“用于对图像编码和解码的方法、编码和解码设备以及相应的计算机程序”的中国发明专利申请的分案申请。

技术领域

[0002] 本发明一般属于图像处理的领域，且更准确地属于数字图像和数字图像序列的编码和解码。

[0003] 本发明由此可特别应用于在目前的数字视频编码器(MPEG，H.264等）或未来的视频编码器（ITU-T/VCEG(H.265）或ISO/MPEG(HVC)）中实现的视频编码。

背景技术

[0004] 目前的视频编码器(MPEG、H264等）使用视频序列的逐块表示。图像被分为宏块，每个宏块本身可以被分为块，并且每个块或宏块通过图像内或图像间预测来编码。于是，特定的图像通过空间预测（内预测）来编码，而其他图像在相对于一个或多个被编码-解码的参考图像的时间预测（间预测）、在本领域技术人员已知的运动补偿的帮助下进行编码。此外，针对每个块，可以对与原始块减去预测相对应的残留块进行编码。该块的系数可以在变换之后被量化，然后被熵编码器编码。

[0005] 内预测和间预测需要之前已经被编码和解码的特定块可用，从而在解码器上或在编码器上被用于预测当前块。图1示出了这样的预测编码的示例性例子，其中，图像I_N被分为块，该图像的当前块MB_i正在关于预订数量的之前被编码和解码的块MBr_1、MBr_2和MBr_3进行预测编码，例如如阴影箭头所示。上述三个块特别包含紧接当前块MB_i左侧的块MBr_1，以及分别仅紧接当前块MB_i上方和右上方的两个块MBr_2和MBr_3。

[0006] 熵编码器在这里更为感兴趣。熵编码器以其到达的顺序来编码信息。典型地实现“光栅扫描”类型的块的逐行遍历，如图1所示通过引用PRS，从图像左上角的块开始。对于每个块，用来表示块所必须的各个信息项（块的类型、预测模式、残留系数等）被顺序分发到熵编码器。

[0007] 已知在AVC压缩标准（也被称为ISO-MPEG4第10部分和ITU-T H.264）中引入的被称为“CABAC”（上下文适应二进制算术编码器）的足够复杂的有效的算术编码器。

[0008] 熵编码器实现各个概念：

[0009] -算术编码：编码器例如初始在文档J.Rissanen and G.G.Langdon Jr，“Universal modeling and coding，”IEEE Trans.Inform.Theory，vol.IT-27，pp.12-23，Jan.1981中描述的编码器使用符号出现概率来对该符号进行编码；

[0010] -上下文适应：在这里，这包括适应要编码的符号的出现概率。一方面，快速实现学习。另一方面，依赖于之前被编码的信息的状态，特定的上下文被用于编码。对于每个上下文，存在固有的符号出现概率与之对应。例如，上下文对应于根据给定配置来编码的符号的类型（残留系数的表示、编码模式的信号等），或者邻居的状态（例如在邻居中选择的

“内”模式的数量等）。

[0011] -二进制化：实现要编码的符号的比特序列的成形。随后，这些各个比特被相继分发到二进制熵编码器。

[0012] 于是，针对使用的每个上下文，该熵编码器实现一种系统，针对所考虑的上下文来快速学习关于之前被编码的符号的概率。该学习基于对这些符号进行编码的顺序。典型地，根据如上所述的“光栅扫描”类型的顺序来遍历图像。

[0013] 在可以是 0 或 1 的给定符号 b 的编码期间，通过下列方式来更新当前块 MB_i 的该符号的出现概率 p_i 的学习：

[0014]

$$p_i(b=0)=\alpha\, p_{i-1}(b=0)+\begin{cases}(1-\alpha) & \text{if coded bit is 0}\\ 0 & \text{otherwise}\end{cases}$$

[0015] 其中，α 是预定值例如 0.95，p_{i-1} 是在该符号最后出现时计算的符号出现概率。

[0016] 图 1 示出了这样的熵编码的示例性例子，其中，图像 I_N 的当前块 MB_i 被熵编码。在块 MB_i 的熵编码开始时，所使用的符号出现概率是之前被编码和解码的块的编码之后获得的概率，根据上述□光栅扫描□类型的块的逐行遍历，该块正好在当前块 MB_i 的前面。仅为了图的清楚，在图 1 中通过细箭头 针对特定的块来表示这样的基于块和块的依赖性的学习。

[0017] 该类型的熵编码的缺点在于，考虑到块的□光栅扫描□遍历，在对位于一行开始的符号进行编码时，使用的概率主要对应于在前一行结束位置的符号所看到的那些概率。现在，考虑到符号概率的可能的空间变化（例如，对于和运动信息项相关的符号，位于图像右侧部分的运动可以和在左侧部分看到的不同，并且因此对于随后的局部概率来说也是类似的），可以看到概率的局部一致性的缺失，由此可能增加编码期间的效率损失。

[0018] 为了限制该现象，已经提出了对块的遍历顺序的调整，目标是确保更好的局部一致性，但编码和解码仍保持顺序。

[0019] 该类型的熵编码器还有另一个缺点。确实，符号的编码和解码依赖于就此学习的概率，符号的解码可以仅以与在编码期间使用的顺序相同的顺序来实现。典型地，于是解码可以只是顺序的，由此阻止若干个符号的并行解码（例如从多核架构中受益）。

[0020] 文档 Thomas Wiegand,Gary J.Sullivan,Gisle Bjontegaard,and Ajay Luthra, "Overview of the H.264/AVC Video Coding Standard",IEEE Transactions on Circuits and Systems for Video Technology,Vol.13,No.7,pp.560-576,July 2003 还指出，CABAC 熵编码器具有将非整数数量的比特分配给要编码的当前字母表的每个符号的特殊特征，这对于大于 0.5 的符号出现概率是有利的。特别地，CABAC 编码器等待直到它已经读取了若干个符号，然后将预定数量的比特分配给读取的该符号集，编码器将该比特写到要发送到解码器的压缩流中。这样的规定由此使其可能使若干个符号上的比特“交互作用”，并对分数数量的比特上的符号进行编码，该数量反映了与通过符号实际发送的信息更接近的信息。与读取的符号关联的其他比特未在压缩流中发送，而是保持等候，等待被分配给 CABAC 编码器读取的一个或多个新的符号，使其可能再次使这些其他比特交互作用。通过已知的方式，熵编码器在给定的时刻“清空”这些未发送的比特。除非另外说明，在所述给定的时刻，编码器提取还未被发送的比特，并将它们写入到去往解码器的压缩流中。该清空例如在

已经读取要编码的最后一个符号的时刻进行，以确保压缩流确实包含所有比特，该比特将允许解码器对字母表中的所有符号进行解码。通过更一般的方式，作为专用于给定编码器/解码器的性能和功能的函数来确定进行清空的时刻。

[0021] 在2011年4月15号的互联网地址http://research.microsoft.com/en-us/um/people/jinl/paper_2002/msri_jpeg.htm上可用的文档描述了一种对符合JPEG 2000压缩标准的静态图像进行编码的方法。该静态图像经过离散小波变换，然后被量化，由此使其可能获得量化的小波系数，量化索引分别与之关联。在熵编码器的帮助下对获取的量化索引进行编码。量化系数之前被分组为称为代码块的矩形块，大小典型为64x 64或32x 32。每个代码块然后被熵编码独立编码。于是，在对当前的代码块进行编码时，熵编码器不会使用在之前的代码块的编码期间计算的符号出现概率。熵编码器由此在每次开始代码块的编码时处于初始化状态。该方法展示了对代码块的数据进行解码而不用对相邻的代码块进行解码的好处。于是，例如，一个客户端软件可以请求一个服务器软件提供仅客户需要的压缩代码块来对图像中识别的子部分进行解码。该方法还展示了允许代码块的并行编码和/或解码的优势。于是，代码块的大小越小，并行化级别就越高。例如，对于固定为2的并行化级别，两个代码块将被并行编码和/或解码。理论上，并行化级别的值等于图像中要编码的代码块的数量。但是，考虑到该编码没有利用从当前代码块的中间环境出现的概率，该方法获得的压缩性能不是最优的。

发明内容

[0022] 本发明的一个目标是修复上述现有技术的缺陷。

[0023] 为此，本发明的主题涉及一种对至少一个图像进行编码的方法，包括下列步骤：

[0024] -将图像分割为可以包含属于预定符号集的符号的多个块，

[0025] -将所述块分组为预定数量的块子集，

[0026] -通过熵编码模块、通过将数字信息与所考虑的子集中的每个块的符号进行关联来对所述块子集中的每个进行编码，该编码步骤包括针对图像的第一块来初始化熵编码模块的状态变量的子步骤，

[0027] -生成表示被编码的块子集中的至少一个的至少一个数据子流，

[0028] 根据本发明的方法值得注意，在于：

[0029] -在当前块是所考虑的子集中要被编码的第一块的情形下，确定该第一当前块的符号出现概率，该概率是针对至少一个其他子集的编码和解码的预 定块而已经确定的那些概率，

[0030] -在当前块是所考虑的子集中最后编码的块的情形下：

[0031] ●将在所考虑的所述子集中的块的编码期间与所述符号关联的所有数字信息写入到子流中，该子流表示所考虑的子集，

[0032] ●实现初始化子步骤。

[0033] 上述写入步骤相当于，一旦块子集中的最后一块已被编码，清空还未被发送的数字信息（比特），如上描述所解释。

[0034] 上述写入步骤和重新初始化熵编码模块的步骤的耦合使其可能产生包含各个数据子流的被编码的数据流，该数据子流分别对应于该至少一个被编码的块子集，所述流适

合根据各种并行化级别来编码，并且这与应用于块子集的编码类型不管是顺序还是并行无关。于是，在编码时可以在并行级别的选择上有很大的自由度，该并行级别作为期望的编码/解码性能的函数。解码的并行级别可变，且甚至可以与编码的并行级别不同，因为在开始块子集的解码时，解码器总是处于初始化状态。

[0035] 根据第一例子，熵编码模块的状态变量是表示预定符号集的符号中的符号出现概率的区间的两个边界。

[0036] 根据第二例子，熵编码模块的状态变量是本领域技术人员熟知并且在2011年6月21日在下列互联网地址http://en.wikipedia.org/wiki/Lempel%E2%80%93Ziv%E2%80%93Welch中描述的LZW(Lempel-Ziv-Welch)熵编码器的转换表中包含的符号的串。

[0037] 使用在所考虑的块子集的第一当前块的熵编码期间使用针对所述其他子集的第一块来确定的符号出现概率的主要好处是，通过在后者中仅存储所述符号出现概率的更新而不用考虑通过所述其他子集中的其他连续块来学习的符号出现概率，来节省编码器的缓冲存储器。

[0038] 在所考虑的块子集中的第一当前块的熵编码期间使用针对所述子集中除了第一块以外的块例如第二块来确定的符号出现概率的好处是获得更准确且由此更好地学习符号出现概率，由此提升更好的视频压缩性能。

[0039] 在特定的实施例中，块子集被顺序或者并行编码。

[0040] 子集块被顺序编码的事实所具有的好处是展示符合H.264/MPEG-4AVC标准的根据本发明的编码方法。

[0041] 子集块被并行编码的事实所具有的好处是加速编码器处理时间且受益 于图像编码的多平台架构。

[0042] 在另一特定实施例中，当至少两个块子集与至少一个其他的块子集并行编码时，该至少两个被编码的块子集被包含在相同的数据子流中。

[0043] 该规定使其特别可能节省数据子流的信令。确实，为了使解码单元能够尽可能早地对子流进行解码，需要在压缩文件中指示所考虑的子流在何处开始。当若干个块子集被包含在相同的数据子流中时，需要单个指示符，由此降低压缩文件的大小。

[0044] 在又一特定实施例中，当所述被编码的块子集想要以预定的顺序来并行编码时，在对该块子集中的每个分别编码之后给出的数据子流在为了解码而被发送之前根据该预定顺序来预先排序。

[0045] 该规定使其可能使被编码的数据子流适应特定类型的解码，而不需要对图像进行解码然后再次编码。

[0046] 相关地，本发明还涉及一种对至少一个图像进行编码的设备，包括：

[0047] -将图像分割为多个块的装置，该多个块可以包含属于预定符号集的符号，

[0048] -用于将所述块分组为预定数量的块子集的装置，

[0049] -对该块子集中的每个进行编码的装置，该编码装置包括熵编码模块，能够将数字信息与所考虑的子集中的每个块的符号相关联，该编码装置包含针对图像中的第一块来初始化熵编码模块的状态变量的子装置，

[0050] -用于生成数据的至少一个数据子流的装置，该数据子流表示被编码的块子集中

的至少一个。

[0051] 该编码装置值得注意，在于它包括：

[0052] -用于确定当前块的符号出现概率的装置，在当前块是所考虑的子集中要被编码的第一块的情形下，该装置确定第一块的符号出现概率为针对至少一个其他子集的编码和解码的预定块所已经确定的那些概率，

[0053] -写入装置，在当前块是所考虑的子集中最后编码的块的情形下，其被激活以将在所考虑的子集中的块的编码期间已经与所述符号关联的所有数字信息写入到子流中，该子流至少表示所考虑的子集，

[0054] 初始化子装置还被激活以重新初始化熵编码模块的状态变量。

[0055] 通过相应的方式，本发明还涉及一种用于对表示至少一个被编码的图像的流进行解码的方法，包括下列步骤：

[0056] -在所述流中识别与要解码的至少一个块子集分别对应的预定数量的数据子流，所述块能够包含属于预定符号集的符号，

[0057] -通过熵解码模块，通过在至少一个被识别的子流中读取与对应于该至少一个所述被识别的子流的子集中的每个块的符号相关联的数字信息，来对所述被识别的块子集进行解码，该解码步骤包括针对图像中要被解码的第一块来初始化所述熵解码模块的状态变量的子步骤，

[0058] 该解码方法值得注意，在于：

[0059] -在当前块是所考虑的子集中要被解码的第一块的情形下，确定所考虑的子集中的第一块的符号出现概率，该概率是针对至少一个其他子集的被解码的预定块而已经确定的那些概率，

[0060] -在当前块是所考虑的子集中最后解码的块的情形下，实现所述初始化子步骤。

[0061] 在特定的实施例中，所述块子集被顺序或并行地解码。

[0062] 在另一特定的实施例中，当至少两个块子集与至少一个其他的块子集并行解码时，被识别的数据子流中的一个表示所述至少两个块子集。

[0063] 在又一特定的实施例中，当被编码的块子集想要以预定的顺序来并行解码时，与被编码的块子集分别对应的数据子流之前在所述要解码的流中以该预定顺序来排序。

[0064] 相关地，本发明还涉及一种用于对表示至少一个被编码的图像的流进行解码的设备，包括：

[0065] -用于在所述流中识别与要解码的至少一个块子集分别对应的预定数量的数据子流的装置，所述块能够包含属于预定符号集的符号，

[0066] -用于对被识别的块子集进行解码的装置，该解码装置包括熵解码模块，其能够在至少一个所述被识别的子流中读取与对应于该至少一个被识别的子流的子集中的每个块的符号相关联的数字信息，该解码装置包括针对图像中要被解码的第一块来初始化熵解码模块的状态变量的子装置，

[0067] 该解码装置值得注意，在于它包括用于确定当前块的符号出现概率的装置，在当前块是所考虑的子集中要被解码的第一块的情形下，该装置确定该第一块的符号出现概率，作为针对至少一个其他子集的被解码的预定块而已经确定的那些概率，

[0068] 并且在于，在当前块是所考虑的子集中最后解码的块的情形下，初始化 子装置被

激活以重新初始化熵解码模块的状态变量。

[0069] 本发明的目标还在于一种包含指令的计算机程序，当程序被计算机执行时，用于执行上述编码或解码方法的步骤。

[0070] 该程序可以使用任意编程语言，并且可以采用源代码、目标代码或介于源代码和目标代码之间的中间代码的形式，例如部分编译的形式或任意其他想要的形式。

[0071] 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，并且包含如上所述的计算机程序指令。

[0072] 该记录介质可以是能存储程序的实体或装置。例如，该介质可以包括存储装置例如 ROM 如 CD ROM 或微电子电路 ROM，或者磁存储装置如磁盘（软盘）或硬盘。

[0073] 此外，该记录介质可以是可传输介质例如电或光信号，其可以通过电或光缆、通过无线电或其他方式来传递。根据本发明的程序特别地可以从互联网类型的网络上下载。

[0074] 或者，该记录介质可以是其中包含程序的集成电路，该电路适于执行所考虑的方法或者在后者执行时使用。

[0075] 上述编码装置、解码方法、解码装置和计算机程序展示了与根据本发明的编码方法所具有的相同的优势。

附图说明

[0076] 阅读参考附图来描述的两个优选实施例，其他特征和优势将变得明显，在附图中：

[0077] - 图 1 表示现有技术的图像编码图，

[0078] - 图 2A 表示根据本发明的编码方法的主要步骤，

[0079] - 图 2B 详细地表示在图 2A 的编码方法中实现的编码，

[0080] - 图 3A 表示根据本发明的编码装置的第一实施例，

[0081] - 图 3B 表示图 3A 中的编码装置的编码单元，

[0082] - 图 3C 表示根据本发明的编码装置的第二实施例，

[0083] - 图 4A 表示根据第一优选实施例的图像编码 / 解码图，

[0084] - 图 4B 表示根据第二优选实施例的图像编码 / 解码图，

[0085] - 图 5A 表示根据本发明的解码方法的主要步骤，

[0086] - 图 5B 详细地表示在图 5A 的解码方法中实现的解码，

[0087] - 图 6A 表示根据本发明的解码装置的实施例，

[0088] - 图 6B 表示图 6A 中的解码装置的解码单元，

[0089] - 图 7A 表示实现顺序类型的编码和并行类型的解码的图像编码 / 解码图，

[0090] - 图 7B 表示以各自不同的并行级别来实现并行类型的编码 / 解码的图像编码 / 解码图。

具体实施方式

[0091] 现在将描述本发明的实施例，其中，根据例如符合 H. 264/MPEG-4AVC 标准的编码而获得的二进制流，依据根据本发明的编码方法被用于对图像序列进行编码。在该实施例中，通过对初始符合 H. 264/MPEG-4AVC 标准的编码器的调整，根据本发明的编码方法例如

以软件或硬件的方式来实现。根据本发明的编码方法以包含步骤C1到C5的算法的形式来表示，如图2A所示。

[0092] 根据本发明的实施例，根据本发明的编码方法在编码装置CO中实现，其两个实施例分别在图3A和3C中表示。

[0093] 参考图2A，第一编码步骤C1是将要编码的图像序列中的图像IE划分为多个块或宏块MB，如图4A或4B所示。所述宏块可以包含一个或多个符号，所述符号形成预定符号集的一部分。在所示例子中，所述块MB具有正方形并且都具有相同的大小。作为图像大小的函数，该图像不必是块大小的倍数，左侧的最后一块和底部的最后一块可以不是正方形。在替代的实施例中，块例如可以是长方形大小并且/或者互相不对齐。

[0094] 每个块或宏块自身可以进一步被分为子块，该子块本身可以再细分。

[0095] 该划分是通过图3A所示的划分模块PCO来执行的，其使用众所周知的划分算法。

[0096] 参考图2A，第二编码步骤C2是将上述块分组为要被顺序或并行编码的预订数量P的连续块的子集SE1，SE2，…，SEk，…，SEP。在图4A和4B所示的例子中，为了图的清楚，仅展示了四个子集SE1、SE2、SE3、SE4。这四个块子集每个用虚线表示，并且分别包含图像IE的前四行的块。

[0097] 该分组是通过图3A中所示的计算模块GRCO在本身众所周知的算法的辅助下执行的。

[0098] 参考图2A，第三编码步骤C3包括对所述块子集SE1到SE6中的每个进行编码，根据例如顺序类型的预定遍历次序PS来对所考虑的子集中的块进行编码。在图4A和4B所示的例子中，当前块SEk(1 ≤ k ≤ P)中的块如箭头PS所示从左到右被依次编码。

[0099] 根据第一变体，该编码是顺序类型，并且通过如图3A所示的单个编码单元UC来实现。通过本身已知的方式，编码器CO包括缓冲存储器MT，其被适配为包含在当前块编码时被逐渐重复更新的符号出现的概率。

[0100] 如图3B中更详细地展示，编码单元UC包括：

[0101] *关于至少一个之前被编码和解码的块对当前块进行预测编码的模块，被表示为MCP；

[0102] *使用针对所述之前被编码和解码的块来计算的至少一个符号出现的概率来对所述当前块进行熵编码的模块，被表示为MCE。

[0103] 预测编码模块MCP是软件模块，其能够根据传统的预测技术例如以内和/或间模式对当前块进行预测编码。

[0104] 熵编码模块MCE本身是CABAC类型，但根据本发明来调整，在描述中将进一步描述。

[0105] 作为变体，熵编码模块MCE可以是本身已知的霍夫曼编码器。

[0106] 在图4A和4B所示的例子中，单元UC对第一行SE1中的块从左到右进行编码。当它到达第一行SE1的最后一块时，它传递到第二行SE2的第一块。当它到达第二行SE2的最后一块时，它传递到第三行SE3的第一块。当它到达第三行SE3的最后一块时，它传递到第四行SE4的第一块，等等，直到图像IE的最后一块被编码。

[0107] 与以上刚才描述的不同的其他类型的遍历当然是可能的。于是，可以将图像IE划分为若干个子图像并将该类型的划分独立应用于每个子图像。编码单元还可以不和以上解

释的那样相继处理各行，而是相继（处理）各列。还可以以任意方向来遍历行或列。

[0108] 根据第二变体，该编码是并行类型，并且，仅通过由预订数量 R 的编码单元 UCk(1 ≤ k ≤ R) 来实现的事实（在图 3C 所示的例子中 R = 2），它与顺序编码的第一变体相区分。已知该并行编码给编码方法带来了明显的加速。

[0109] 编码单元 UCk 中的每个等价于图 3B 所示的编码单元 UC。通过相应的方式，编码单元 UCk 包括预测编码模块 MCPk 和熵编码模块 MCEk。

[0110] 再次参考图 4A 和 4B，第一单元 UC1 对奇数排名的行中的块进行编码，而第二单元 UC2 对偶数排名的行中的块进行编码。更准确地说，第一单元 UC1 从左到右对第一行 SE1 中的块进行编码。当它达到第一行 SE1 的最后一块时，它传递第 (2n+1) 行即第三行 SE3 等的第一块。与第一单元 UC1 执行的处理并行，第二单元 UC2 从左到右对第二行 SE2 中的块进行编码。当它到达第二行 SE2 的最后一块，它传递到第 (2n) 行在这里是第四行 SE4 等的第一块。上述两个遍历被重复，直到图像 IE 的最后一块被编码。

[0111] 参考图 2A，第四编码步骤 C4 是产生比特的 L 个子流 F1，F2，…，Fm，…，FL(1 ≤ m ≤ L ≤ P) 来表示通过上述编码单元 UC 或者上述编码单元 UCk 中的每个来压缩的被处理的块以及每个子集 SEk 中被处理的块的解码版本。根据在描述中将进一步细化的同步机制，所考虑的子集中用 SED1，SED2，…，SEDk，…，SEDP 表示的被解码的处理的块可以被图 3A 所示的编码单元 UC 或图 3C 所示的编码单元 UCk 中的每个复用。

[0112] 参考图 3B，产生 L 个子流的步骤是通过流生成软件模块 MGSF 或 MSGFk 来实现的，其被适配为产生数据流例如比特。

[0113] 参考图 2A，第五编码步骤 C5 包括基于上述 L 个子流 F1，F2，…，Fm，…，FL 来构造全局流 F。根据一个实施例，子流 F1，F2，…，Fm，…，FL 被简单地并置，用额外的信息项来向解码器指示全局流 F 中的每个子流 Fm 的位置。后者之后被通信网络（未示出）发送到远程终端。后者包括图 5A 所示的解码器 DO。根据另一实施例，这尤其有利，因为不需要对图像解码然后再次编码，在将流 F 发送到解码器 DO 之前，解码器 DO 之前以预定的顺序来对 L 个子流 F1，F2，…，Fm，…，FL 进行排序，该顺序对应于 DO 能够对子流进行解码的顺序。

[0114] 于是，如描述中更详细地描述，根据本发明的解码器能够在全局流 F 中分离子流 F1，F2，…，Fm，…，FL，并将它们分配给组成解码器的一个或多个解码单元。需要注意，全局流中的子流的该分解独立于使用单个编码单元或并行操作的若干个编码单元的选择，并且用该方法，可以只有编码器或只有解码器，其包含并行操作的单元。

[0115] 全局流 F 的该构造是在例如图 3A 和图 3C 所示的流构造模块 CF 中实现的。

[0116] 现在将参考图 2B 来描述本发明的各个特定子步骤，例如在上述编码步 骤 C3 期间在编码单元 UC 或 UCk 中实现的子步骤。

[0117] 在步骤 C31 的过程中，编码单元 UC 或 UCk 选择图 4A 或 4B 所示的当前块 SEk 例如第一行 SE1 中要编码的第一块作为当前块。

[0118] 在步骤 C32 的过程中，单元 UC 或 UCk 测试当前块是否是图像 IE 的第一块（位于顶部且位于左侧），该图像在上述步骤 C1 中已被划分为块。

[0119] 如果是这种情况，在步骤 C33 的过程中，熵编码模块 MCE 或 MCEk 对其状态变量进行初始化。根据所示例子，其使用如前所述的算术编码，这包含表示预定符号集中包含的符

号的出现的概率的间隔的初始化。通过本身已知的方式，用两个边界 L 和 H，分别为下边界和上边界，来初始化该间隔。下边界 L 的值被固定为 0，而上边界的值被固定为 1，由此对应于预定符号集的所有符号中的第一符号的出现的概率。该间隔的大小 R 因此在此刻由 R = H-L = 1 来定义。被初始化的间隔被传统地进一步划分为多个预定的子间隔，其分别表示预定符号集中的符号的出现的概率。

[0120] 作为变体，如果使用的熵编码是 LZW 编码，符号的串的转换表被初始化，从而它包含所有可能的符号一次且仅一次。

[0121] 如果在上述步骤 C32 之后，当前块不是图像 IE 的第一块，则在后续描述中之后被描述的步骤 C40 的过程中确定必要的之前被编码和解码的块的可用性。

[0122] 在步骤 C34 的过程中，对图 4A 或 4B 中表示的第一行 SE1 中的第一当前块 MB1 进行编码。该步骤 C34 包括以下将描述的多个子步骤 C341 到 C348。

[0123] 在图 2B 所示的第一子步骤 C341 的过程中，通过内和 / 或间预测的已知技术对当前块 MB1 进行预测编码，在其过程中，关于至少一个之前被编码和解码的块来对块 MB1 进行预测。

[0124] 毋庸置疑，例如 H. 264 标准中提出的其他模式的内预测也是可以的。

[0125] 还可以以内模式对当前块 MB1 进行预测编码，在其过程中，关于在之前被编码和解码的图像中出现的块来对当前块进行预测。其他类型的预测当然也是可能的。在对当前块的可能的预测中，根据对于本领域技术人员来说众所周知的比特率失真准则来选择最优预测。

[0126] 上述预测编码步骤使其可能构造预测的块 MBp_1，这是对当前块 MB_1 的近似。与预测编码相关的信息后续将被写入到流 F 中，该流 F 被发送到解码 器 DO。该信息特别包括预测类型（间或内（预测）），并且如果合适，包括内预测的类型、块或宏块（如果后者已经被细分）的划分类型、在间预测模式中使用的参考图像索引和位移矢量。该信息被编码器 CO 压缩。

[0127] 在之后的子步骤 C342 的过程中，从当前块 MB_1 中减去预测块 MBp_1 以产生残留块 MBr_1。

[0128] 在之后的子步骤 C343 的过程中，根据传统的直接变换操作例如 DCT 类型的离散余弦变换来对残留块 MBr_1 进行变换，以产生变换后的块 MBt_1。

[0129] 在之后的子步骤 C344 的过程中，根据传统的量化操作例如标量量化来对变换后的块 MBt_1 进行量化。然后得到系数被量化的块 MBq_1。

[0130] 在之后的子步骤 C345 的过程中，对系数被量化的块 MBq_1 进行熵编码。在优选的实施例中，这包括 CABAC 熵编码。该步骤包括：

[0131] a) 读取与所述当前块关联的符号或预定符号集中的符号，

[0132] b) 将数字信息例如比特与读取的符号进行关联。

[0133] 在上述变体中（根据该变体，使用的编码是 LZW 编码），与当前转换表中的符号的代码相对应的数字信息项与要编码的符号关联，并根据本身已知的过程来更新该转换表。

[0134] 在之后的子步骤 C346 的过程中，根据传统的反量化操作即与步骤 C344 中执行的量化相反的操作来对块 MBq_1 进行反量化。然后得到系数被反量化的块 $MBDq_1$。

[0135] 在之后的子步骤 C347 的过程中，对系数被反量化的块 $MBDq_1$ 进行逆变换，这是与

以上步骤 C343 中执行的直接变换相反的操作。然后得到被解码的残留块 $MBDr_i$。

[0136] 在之后的子步骤 C348 的过程中，通过将解码的残留块 $MBDr_i$ 加到预测块 MBp_i 来构造被解码的块 MBD_i。需要注意，后一块与在描述中将进一步描述的对图像 IE 进行解码的方法完成之后得到的被解码的块相同。被解码的块 MBD_i 由此被呈现为可用，以被构成预定数量 R 的编码单元的一部分的编码单元 UCk 或任意其他编码单元使用。

[0137] 在完成上述编码步骤 C34 之后，例如如图 3B 所示的熵编码模块 MCE 或 MCEk 包含在第一块的编码时被逐渐重复更新的所有概率。这些概率对应于可能的语法的各个元素或者各个关联的编码上下文。

[0138] 在上述编码步骤 C34 之后，在步骤 C35 的过程中执行测试以确定当前 块是否是同一行的第 j 块，其中 j 是编码器 CO 已知的预定值，其至少等于 1。

[0139] 如果是该情况，在图 2B 所示的步骤 C36 的过程中，针对第 j 块来计算的概率集合被存储在例如图 3A 或 3B 以及图 4A 和 4B 中所示的编码器 CO 的缓冲存储器 MT 中，所述存储器的大小被适配为存储所计算的概率数量。

[0140] 在图 2B 所示的步骤 C37 的过程中，编码单元 UC 或 UCk 测试刚才被编码的行 SEk 的当前块是否是图像 IE 中的最后一块。如果在步骤 C35 的过程中当前块不是行 SE1 的第 j 块，该步骤也会被实现。

[0141] 在当前块是图像 IE 的最后一块时，在步骤 C38 的过程中，编码方法结束。

[0142] 如果不是该情况，在步骤 C39 的过程中，根据如图 4A 或 4B 中的箭头 PS 所示的遍历顺序来选择要编码的下一块 MB_i。

[0143] 在图 2B 所示的步骤 C40 的过程中，确定对于当前块 MB_i 的编码来说必须的之前被编码和解码的块的可用性。

[0144] 如果这是第一行 SE1，该步骤包括验证位于要编码的 MB_i 的左侧的至少一个块的可用性。但是，针对图 4A 或 4B 所示的实施例中选择的遍历顺序 PS，在所考虑的行 SEk 中依次对各个块进行编码。结果，左侧被编码和解码的块总是可用的（除了一行的第一块）。在图 4A 或 4B 所示的例子中，这是直接位于要编码的当前块的左侧的块。

[0145] 如果这是与第一行不同的行 SEk，所述确定步骤还包括验证位于前一行 SEk-1 中的预订数量 N' 的块例如分别位于当前块的上方和右上方的两个块是否可用于当前块的编码，即，它们是否已经被编码单元 UC 或 UCk-1 编码且然后被解码。

[0146] 由于该测试步骤容易降低编码方法的速度，通过根据本发明的替代方式，在行的编码是并行类型的情形下，图 3C 所示的时钟 CLK 被适配为同步块编码的进展以确保分别位于当前块的上方和右上方的两个块的可用性，而不需要验证这两块的可用性。于是，编码单元 UCk 总是以被用于当前块的编码的前一行 SEk-1 中的预订数量 N'（例如 $N' = 2$）的被编码和解码的块的偏移来开始第一块的编码。从软件的角度来看，这样的时钟的实现使其可能显著加速编码器 CO 中对图像 IE 中的块的处理时间。

[0147] 在图 2B 所示的步骤 C41 的过程中，执行测试以确定当前块是否是所考虑的行 SEk 中的第一块。

[0148] 如果是这种情况，在步骤 C42 的过程中，在缓冲存储器 MT 中仅读取在前一行 SEk-1 的第 j 块的编码期间计算的符号出现的概率。

[0149] 根据图 4A 所示的第一变体，第 j 块是前一行 SEk-1 的第一块（$j = 1$）。该读取包

括用缓冲存储器 MT 中存在的概率来替换 CABAC 编码器的概率。为使其处理第二、第三和第四行 SE2、SE3 和 SE4 中的各个第一块，如细线所示的箭头在图 4A 中示出该读取步骤。

[0150] 根据图 4B 所示的上述步骤 C43 的第二变体，第 j 块是前一行 SEk-1 的第二块（j = 2）。该读取包括用在缓冲存储器 MT 中存在的概率来替换 CABAC 编码器的概率。为使其处理第二、第三和第四行 SE2、SE3 和 SE4 中的各个第一块，如细虚线所示的箭头在图 4B 中示出该读取步骤。

[0151] 在步骤 C42 之后，通过重复上述步骤 C34 到 C38 来对当前块进行编码然后进行解码。

[0152] 如果在上述步骤 C41 之后，当前块不是所考虑的行 SEk 的第一块，有利地不用读取在位于同一行 SEk 中的之前被编码和解码的块即在所示例子中直接位于当前块的左侧的被编码和解码的块中出现的概率。确实，考虑到如图 4A 或 4B 所示的用于读取位于同一行中的块的遍历顺序 PS，在开始当前块的编码时在 CABAC 编码器中存在的符号出现的概率正好是在该同一行中的前一块的编码 / 解码后存在的那些概率。

[0153] 结果，在图 2B 所示的步骤 C43 的过程中，针对所述当前块的熵编码来学习符号出现的概率，这些（概率）仅对应于如图 4A 或 4B 中的双实线箭头所示的针对同一行中的所述前一块来计算的那些概率。

[0154] 在步骤 C43 之后，重复上述步骤 C34 到 C38 来对当前块进行编码然后进行解码。

[0155] 在步骤 C44 的过程之后执行测试以确定当前块是否是所考虑的行 SEk 中的最后一块。

[0156] 如果不是该情况，在步骤 C44 之后，再次实现对要编码的下一块 MB_i 进行选择的步骤 C39。

[0157] 如果当前块是所考虑的行 SEk 的最后一块，在步骤 C45 的过程中，图 3A 或 3C 的编码装置 CO 执行如上描述中提到的清空。为此，编码单元 UCk 将已经与在所考虑的所述行中的每个块的编码期间读取的符号关联的所有比特发送到相应的子流生成模块 MGSFk，通过这样的方式从而模块 MGSFk 将所有比特写入到子流 Fm 中，该子流包含的二进制串表示所考虑的所述行 SEk 中被编码的块。这样的情况在图 4A 和 4B 中用每行 SEk 的末端的三角形来表示。

[0158] 在图 2B 所示的步骤 C46 的过程中，编码单元 UC 或 UCk 执行与上述步骤 C33 相同的步骤，即再次初始化表示预定符号集中包含的符号的出现的概率的间隔。该重新初始化在图 4A 和 4B 中用每行 SEk 开始的黑点来表示。

[0159] 在该编码级别上执行步骤 C45 和 C46 的好处是，在编码单元 UC 或编码单元 UCk 处理的下一块的编码期间，编码器 CO 处于初始化状态。于是，如下将在描述中进一步描述，并行工作的解码单元可能从该点开始对压缩过的流 F 进行直接解码，因为它满足处于初始化状态。

[0160] 解码部分的实施例的详细描述

[0161] 现在将描述根据本发明的解码方法的实施例，其中，通过对初始符合 H.264/MPEG-4AVC 标准的编码器的调整，根据本发明的解码方法例如以软件或硬件的方式来实现。

[0162] 根据本发明的编码方法以包含步骤 D1 到 D4 的算法的形式来表示，如图 5A 所示。

[0163] 根据本发明的实施例，根据本发明的解码方法在图 6A 所示的解码装置 DO 中实现。

[0164] 参考图 5A，第一解码步骤是在所述流 F 中识别 L 个子流 F1，F2，…，Fm，…，FL，该子流分别包含如图 4A 或 4B 所示之前被编码的块或宏块 MB 的 P 个子集 SE1，SE2，…，SEk，…，SEP。为此，流 F 中的每个子流与指示符关联，该指示符旨在允许解码器 DO 确定每个子流 Fm 在流 F 中的位置。作为变体，在完成上述编码步骤 C3 时，解码器 CO 以解码器 DO 期望的顺序对流 F 中的子流 F1，F2，…，Fm，…，FL 进行排序，由此避免在流 F 中插入子流标识符。该规定由此使其可能降低与数据流 F 的比特率相关的成本。

[0165] 在图 4A 或 4B 所示的例子中，所述块 MB 具有正方形并且都具有相同的大小。取决于图像大小，该图像不必是块大小的倍数，左侧的最后一块和底部的最后一块可以不是正方形。在替代的实施例中，块例如可以是长方形大小并且 / 或者互相不对齐。

[0166] 每个块或宏块自身可以进一步被分为子块，该子块本身可以再细分。

[0167] 该识别是通过例如图 6A 所示的流提取模块 EXDO 来执行的。

[0168] 在图 4A 或 4B 所示的例子中，预定数量 P 等于 6，并且为了图的清楚，仅用虚线示出了四个子集 SE1、SE2、SE3、SE4。

[0169] 参考图 5A，第二解码步骤 D2 是对所述块子集 SE1、SE2、SE3 和 SE4 中的每个进行解码，所考虑的子集中的块是根据预定的遍历顺序 PS 来编码的。在图 4A 或 4B 所示的例子中，当前子集 SEk(1 ≤ k ≤ P) 中的块如箭头 PS 所示从左到右被依次被解码。在步骤 D2 完成时，得到被解码的块子集 SED1，SED2，SED3，…，SEDk，…，SEDP。

[0170] 这样的解码可以是顺序类型的，且因此在单个解码单元的辅助下执行。

[0171] 但是，为了能够受益于多平台解码架构，块子集的解码是并行类型并且通过数量 R 的解码单元 UDk(1 ≤ k ≤ R) 来实现，例如如图 6A 所示 R = 4。该规定由此允许解码方法的明显加速。通过本身已知的方式，解码器 DO 包括缓冲存储器 MT，其被适配为包含在当前块解码时被逐渐重复更新的符号出现的概率。

[0172] 如图 6B 中更详细的表示，每个解码单元 UDk 包括：

[0173] ●通过学习针对至少一个之前被解码的块来计算的至少一个符号出现的概率来对所述当前块进行熵解码的模块，用 MDEk 来表示，

[0174] ●关于所述之前被解码的块来对当前块进行预测解码的模块，用 MDPk 来表示。

[0175] 预测解码模块 SUDPk 能够根据传统的预测技术例如内和 / 或间模式来对当前块进行解码。

[0176] 熵解码模块 MDEk 本身是 CABAC 类型，但根据本发明来调整，如下在描述中将进一步描述。

[0177] 作为变体，熵解码模块 MDEk 可以是本身已知的霍夫曼解码器。

[0178] 在图 4A 或 4B 所示的例子中，第一单元 UD1 从左到右对第一行 SE1 中的块进行解码。当它到达第一行 SE1 的最后一块，它传递到第 (n+1) 行在这里是第 5 行等的第一块。第二单元 UC 从左到右对第二行 SE2 中的块进行解码。当它到达第二行 SE2 的最后一块，它传递到第 (n+2) 行这里是第 6 行等的第一块。该遍历重复直到单元 UD4，其从左到右对第四行进行解码。当它到达第一行的最后一块，它传递到第 (n+4) 行在这里是第 8 行的第一 块，等待，直到最后被识别的子流中的最后一块被解码。

[0179] 与上述刚才描述的不同的遍历类型当然是可能的。例如，每个解码单元可以不处理如上解释的嵌入的行而是嵌入的列。还可以以任意方向来遍历行或列。

[0180] 参考图 5A，第三解码步骤 D3 是基于在解码步骤 D2 中得到的每个被解码的子集 SED1，SED2，…，SEDk，…，SEDP 来重构被解码的图像 ID。更准确地说，每个被解码的子集 SED1，SED2，…，SEDk，…，SEDP 中被解码的块被发送到例如如图 6A 所示的图像重构单元 URI。在步骤 D3 的过程中，当这些变得可用时，单元 URI 将被解码的块写入到被解码的图像。

[0181] 在图 5A 所示的第四解码步骤 D4 的过程中，图 6A 所示的单元 URI 给出被完全解码的图像 ID。

[0182] 现在将参考图 5B 来描述本发明的各个特定子步骤例如在并行解码的上述步骤 D2 期间在解码单元 UDk 中实现的子步骤。

[0183] 在步骤 D21 的过程中，解码单元 UDk 选择图 4A 或 4B 所示的当前块 SEk 中要解码的第一块作为当前块。

[0184] 在步骤 D22 的过程中，解码单元 UDk 测试当前块是否是被解码的图像在的第一块，在该情形下，是否是子流 F1 中的第一块。

[0185] 如果是该情况，在步骤 D23 的过程中，熵解码模块 MDE 或 MDEk 初始化其状态变量。根据所示例子，这需要对表示预定符号集中包含的符号的出现的概率的间隔进行初始化。

[0186] 作为变体，如果使用的熵解码是 LZW 解码，符号的串的转换表被初始化，从而它包含所有可能的符号一次且仅一次。步骤 D23 等同于上述编码步骤 C33，后面不会再被描述。

[0187] 如果在上述步骤 D22 之后，当前块不是被解码的图像 ID 中的第一块，在后续描述中之后被描述的步骤 D30 的过程中确定必要的之前被解码的块的可用性。

[0188] 在步骤 D24 的过程中，对图 4A 或 4B 所示的第一行 SE1 中的第一当前块 MB1 进行解码。该步骤 D24 包括将在下面描述的多个子步骤 D241 到 D246。

[0189] 在第一子步骤 D241 的过程中，对与当前块相关的语法元素进行熵解码。该步骤主要包括：

[0190] a) 读取与所述第一行 SE1 关联的子流中包含的比特，

[0191] b) 基于读取的比特来重构符号。

[0192] 在上述变体中（根据该变体，使用的解码是 LZW 解码），读取与当前转换表中的符号代码相对应的数字信息项，基于读取的代码来重构符号，并根据本身已知的过程来更新转换表。

[0193] 更准确地说，与当前块相关的符号元素被例如如图 6B 所示的 CABAC 熵解码模块 MDE1 解码。后者对压缩文件中的比特子流 F1 进行解码以产生符号元素，并且同时以该方式来重新更新其概率，从而在后者对符号进行解码时，该符号出现的概率等于在上述熵解码步骤 C345 期间对该相同符号进行编码时得到的那些概率。

[0194] 在之后的子步骤 D242 的过程中，通过内和 / 或间预测的已知技术来对当前块 MB1 进行预测解码，在其过程中，关于至少一个之前被解码的块来预测块 MB1。

[0195] 毋庸置疑，例如在 H.264 标准中提出的其他内预测模式也是可以的。

[0196] 在该步骤的过程中，在前一步骤中解码的语法元素的辅助下执行预测解码，该语法元素特别包括（间或内）预测的类型，并且如果合适，包括内预测的模式、块或宏块（如果后者已经被细分）的划分类型、在间预测模式中使用的参考图像索引和位移矢量。

[0197] 上述预测解码步骤使其可能重构预测的块 MBp_1。

[0198] 在之后的子步骤 D243 的过程中，在之前被解码的语法元素的辅助下重构量化的

残留块 MBq_1。

[0199] 在之后的子步骤 D244 的过程中，根据传统的反量化操作即与上述步骤 C344 中执行的量化相反的操作来对被量化的残留块 MBq_1进行反量化，以产生被解码的反量化的块 $MBDt_1$。

[0200] 在之后的子步骤 D245 的过程中，对被反量化的块 $MBDt_1$进行逆变换即与上述步骤 C343 中执行的直接变换相反的操作。于是得到被解码的残留块 $MBDr_1$。

[0201] 在之后的子步骤 D246 的过程中，通过将被解码的残留块 $MBDr_1$加到预测块 MBp_1来构造被解码的块 MBD_1。被解码的块 MBD_1由此被呈现为可用，以被解码单元 UD1 或构成预订数量 N 的解码单元的一部分的任意其他解码单元使用。

[0202] 在完成上述解码步骤 D246 时，例如如图 6B 所示的熵解码模块 MDE1 包含在第一块的解码的同时被逐渐重复更新的所有概率。这些概率对应于可能的语法的各个元素并且对应于各个关联的解码上下文。

[0203] 在上述解码步骤 D24 之后，在步骤 D25 的过程中执行测试来确定当前块是否是同一行的第 j 块，其中 j 是解码器 DO 已知的预定值，其至少等于 1。

[0204] 如果是这种情况，在步骤 D26 的过程中，针对第 j 块计算的概率集合被存储在例如如图 6A 所示以及图 4A 或 4B 中的解码器 DO 中的缓冲存储器 MT，所述存储器的大小被适配为存储所计算的概率数量。

[0205] 在步骤 D27 的过程中，单元 UDk 测试刚才被解码的当前块是否是最后的子流中的最后一块。

[0206] 如果是这种情况，在步骤 D28 的过程中，解码方法结束。

[0207] 如果不是该情况，在步骤 D29 的过程中根据图 4A 或 4B 中的箭头 PS 所示的遍历顺序来选择要解码的下一块 MB_i。

[0208] 如果在上述步骤 D25 的过程中，当前块不是所考虑的行 SEDk 的第 j 块，则执行以上步骤 D27。

[0209] 在上述步骤 D29 之后的步骤 D30 的过程中，确定对于当前块 MB_i的解码来说必要的之前被解码的块的可用性。考虑到这需要通过不同解码单元 UDk 来对块进行并行解码的事实，可能出现这些块没有被指定对这些块进行解码的解码单元来解码，且因此它们还不可用。所述确定步骤包括验证位于前一行 SEk-1 中的预订数量 N' 的块例如分别位于当前块的上方和右上方的两个块是否可用于当前块的解码，即，它们是否已经被指定对其进行解码的解码单元 UDk-1 解码。所述确定步骤还包括确定位于要编码的当前块 MB_i的左侧的至少一块的可用性。但是，考虑到图 4A 或 4B 所示的实施例中选择的遍历顺序 PS，所考虑的行 SEk 中的块被依次解码。结果，左侧被解码的块总是可用的（除了一行中的第一块）。在图 4A 或 4B 所示的例子中，这需要直接位于当前块的左侧的块被解码。为此，只有分别位于当前块的上方和右上方的两个块的可用性被测试。

[0210] 由于该测试步骤容易降低解码方法的速度，通过根据本发明的替代方式，图 6A 所示的时钟 CLK 被适配为同步块解码的进展以确保分别位于当前块的上方和右上方的两个块的可用性，而不需要验证这两块的可用性。于是，如图 4A 或 4B 所示，解码单元 UDk 总是以被用于当前块的解码的前一行 SEk-1 中的预订数量 N'（这里 N' = 2）的被解码的块的偏移来开始第一块的解码。从软件的角度来看，这样的时钟的实现使其可能显著加速解码

器 DO 中对每个子集 SEk 中的块进行处理的时间。

[0211] 在步骤 D31 的过程中，执行测试以确定当前块是否是所考虑的行 SEk 中的第一块。

[0212] 如果是这种情况，在步骤 D32 的过程中，在缓冲存储器 MT 中仅读取在前一行 SEk-1 的第 j 块的解码期间计算的符号出现的概率。

[0213] 根据图 4A 所示的第一变体，第 j 块是前一行 SEk-1 的第一块（j = 1）。该读取包括用缓冲存储器 MT 中存在的概率来替换 CABAC 解码器的概率。为使其处理第二、第三和第四行 SE2、SE3 和 SE4 中的各个第一块，如细线所示的箭头在图 4A 中示出该读取步骤。

[0214] 根据图 4B 所示的上述步骤 D32 的第二变体，第 j 块是前一行 SEk-1 的第二块（j = 2）。该读取包括用在缓冲存储器 MT 中存在的概率来替换 CABAC 解码器的概率。为使其处理第二、第三和第四行 SE2、SE3 和 SE4 中的各个第一块，如细虚线所示的箭头在图 4B 中示出该读取步骤。

[0215] 在步骤 D32 之后，通过重复上述步骤 D24 到 D28 来对当前块进行解码。

[0216] 如果在上述步骤 D31 之后，当前块不是所考虑的行 SEk 的第一块，有利地不用读取在位于同一行 SEk 中的之前被解码的块即在所示例子中直接位于当前块的左侧的被解码的块中出现的概率。确实，考虑到如图 4A 或 4B 所示的用于读取位于同一行中的块的遍历顺序 PS，在开始当前块的解码时在 CABAC 解码器中存在的符号出现的概率正好是在该同一行中的前一块的解码后存在的那些概率。

[0217] 结果，在步骤 D33 的过程中，针对所述当前块的熵解码来学习符号出现的概率，所述概率仅对应于如图 4A 或 4B 中的双实线箭头所示的针对同一行中的所述前一块来计算的那些概率。

[0218] 在步骤 D33 之后，重复上述步骤 D24 到 D28 来对当前块进行解码。

[0219] 在步骤 D34 的过程之后执行测试以确定当前块是否是所考虑的行 SEk 中的最后一块。

[0220] 如果不是该情况，在步骤 D34 之后，再次实现对要解码的下一块 MB_i 进行选择的步骤 D29。

[0221] 如果当前块是所考虑的行 SEk 的最后一块，在步骤 D35 的过程中，解 码装置 UDk 执行与上述步骤 D23 相同的步骤，即再次初始化表示预定符号集中包含的符号的出现的概率的间隔。该重新初始化在图 4A 和 4B 中用每行 SEk 开始的黑点来表示。

[0222] 于是，解码器 DO 在每行开始时处于初始化状态，由此从选择解码的并行级别以及优化解码的处理时间的角度来看允许更大的灵活性。

[0223] 在图 7A 所示的示例性编码 / 解码图中，编码器 CO 包括单个编码单元 UC，如图 3A 所示，而解码器 DO 包括六个解码单元。

[0224] 解码单元对行 SE1、SE2、SE3、SE4、SE5 和 SE6 顺序进行编码。在所示例子中，行 SE1 到 SE4 被完全编码，行 SE5 正在被编码的过程中且行 SE6 还没有被编码。考虑到编码的顺序性，编码单元 UC 被适配为给出流 F，该流包含依次排序的子流 F1、F2、F3、F4，以对行 SE1、SE2、SE3 和 SE4 进行编码。为此，子流 F1、F2、F3 和 F4 用与分别表示被编码的行 SE1、SE2、SE3、SE4 的相同的阴影来表示。通过在所述被编码的行的编码结束时的清空步骤以及在开始对要编码 / 解码的下一行的编码或解码时重新初始化概率的间隔，当解码器 DO 每次读取子流来解码时，解码器处于初始化状态，并且可以以优化的方式用例如可被安装在四个不

同平台上的解码单元 UD1、UD2、UD3 和 UD4 对四个子流 F1、F2、F3 和 F4 进行并行解码。

[0225] 在图 7B 所示的示例性编码 / 解码图中，编码器 CO 包括如图 3C 所示的两个编码单元 UC1 和 UC2，而解码器 DO 包括六个解码单元。

[0226] 编码单元 UC1 对奇数排名的行 SE1、SE3 和 SE5 进行顺序编码，而编码单元 UC2 对偶数排名的行 SE2、SE4 和 SE6 进行顺序编码。为此，行 SE1、SE3 和 SE5 表现为白色背景，而行 SE2、SE4 和 SE6 表现为带点的背景。在所示例子中，行 SE1 到 SE4 被完全编码，行 SE5 正在被编码的过程中而行 SE6 还没有被编码。考虑到所执行的编码的级别为 2 的并行类型的事实，编码单元 UC1 被适配为给出子流 F_{2n}，该子流可被分解为在对行 SE2 和 SE4 分别解码之后得到的两个部分 F2 和 F4。解码器 CO 因此被适配为向解码器 DO 发送流 F，该流包含两个子流 F_{2n+1} 和 F_{2n} 的并置，以及因此与图 7A 所示不同的子流排序 F1、F3、F2、F4。为此，子流 F1、F2、F3 和 F4 用与分别表示被编码的行 SE1、SE2、SE3、SE4 的相同的阴影来表示，子流 F1 和 F3 表现为白色背景（奇数排序的行的编码），且子流 F2 和 F4 表现为带点的背景（偶数排序的行的编码）。

[0227] 关于结合图 7A 所述的优势，该编码 / 解码图还展示了能够清空解码器的优势，该解码器的解码并行级别完全独立于编码的并行级别，由此使其可能进一步优化编码器 / 解码器的操作。

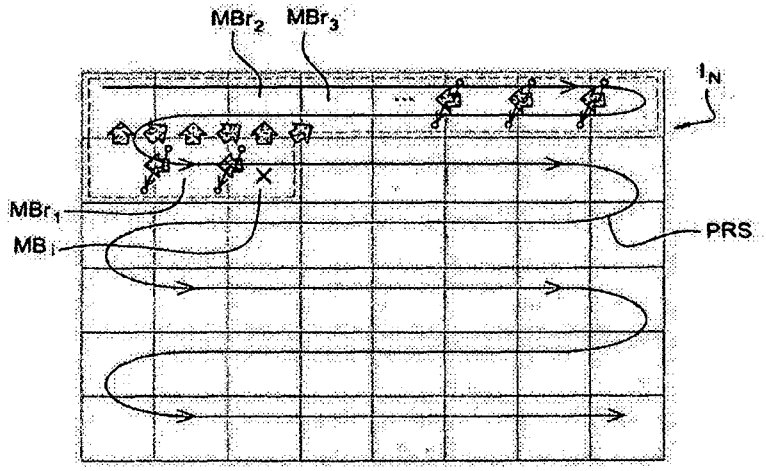


图 1

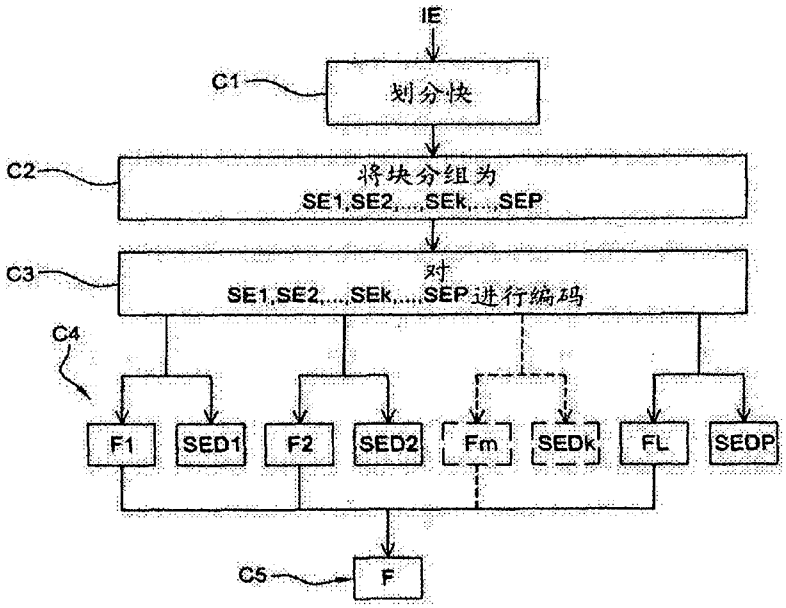


图 2A

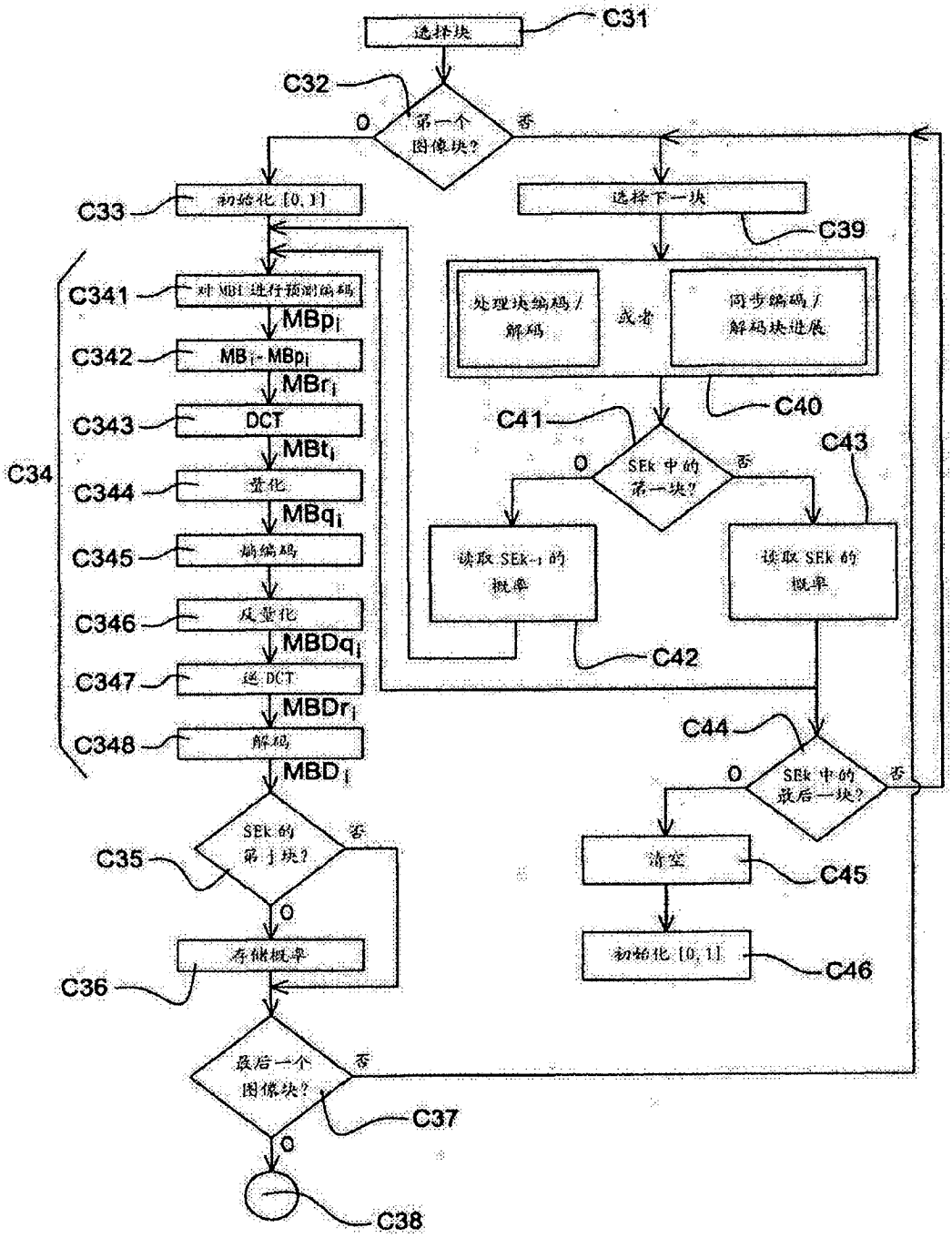


图 2B

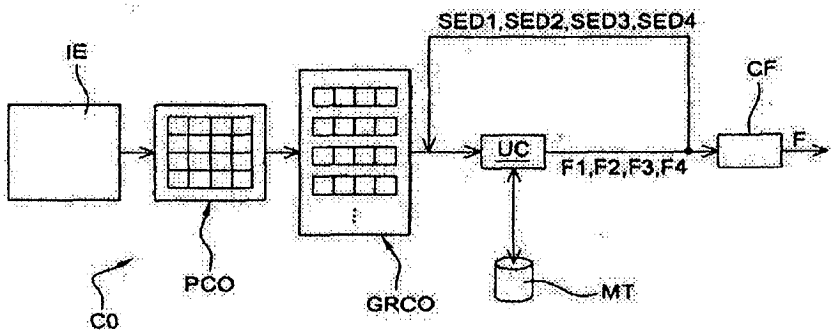


图 3A

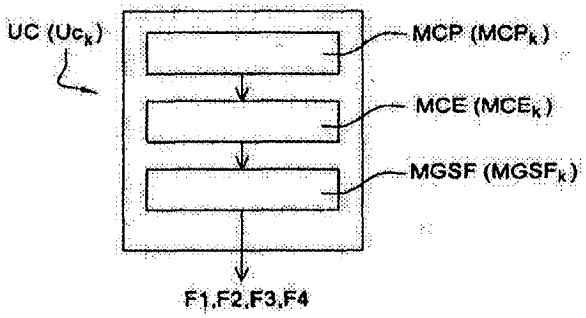


图 3B

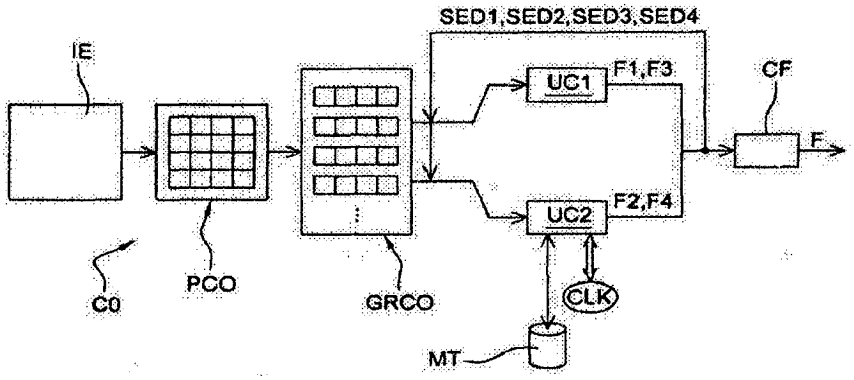


图 3C

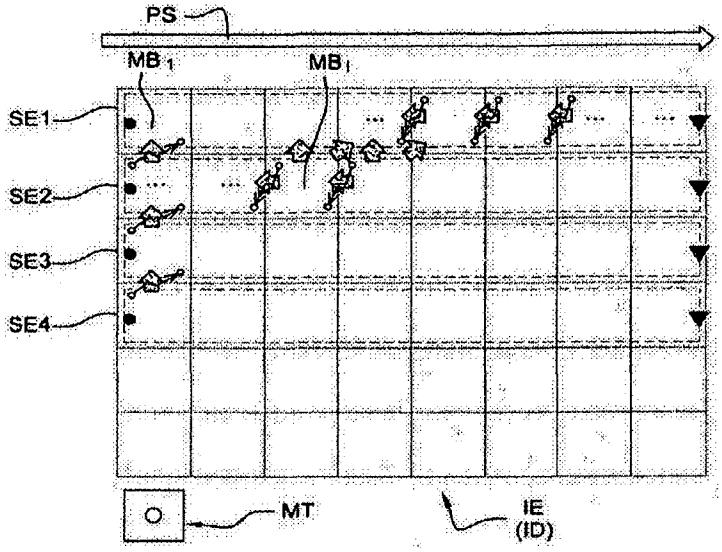


图 4A

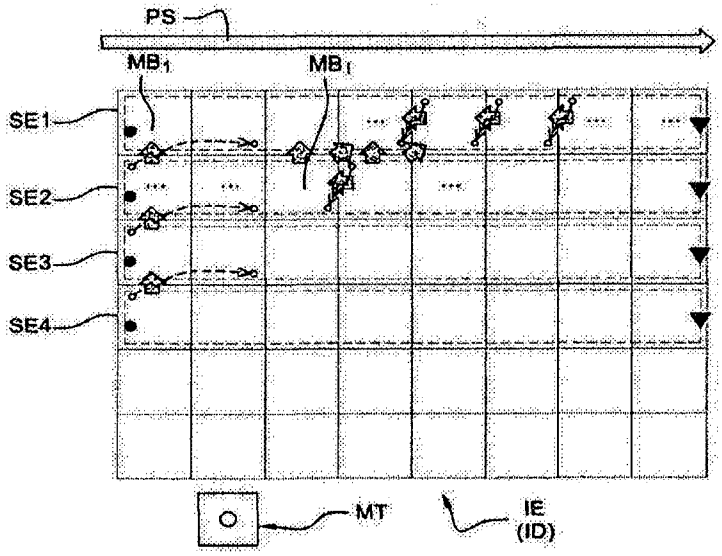


图 4B

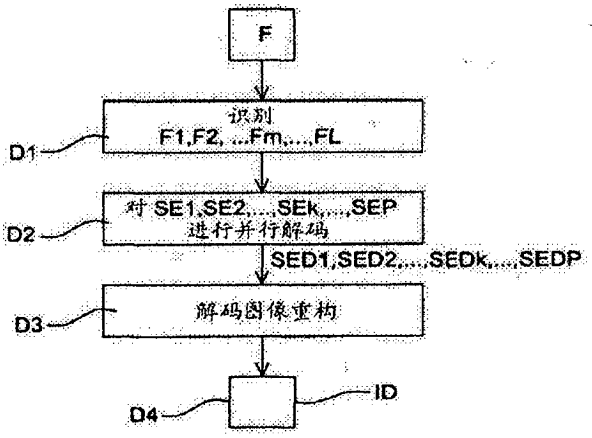


图 5A

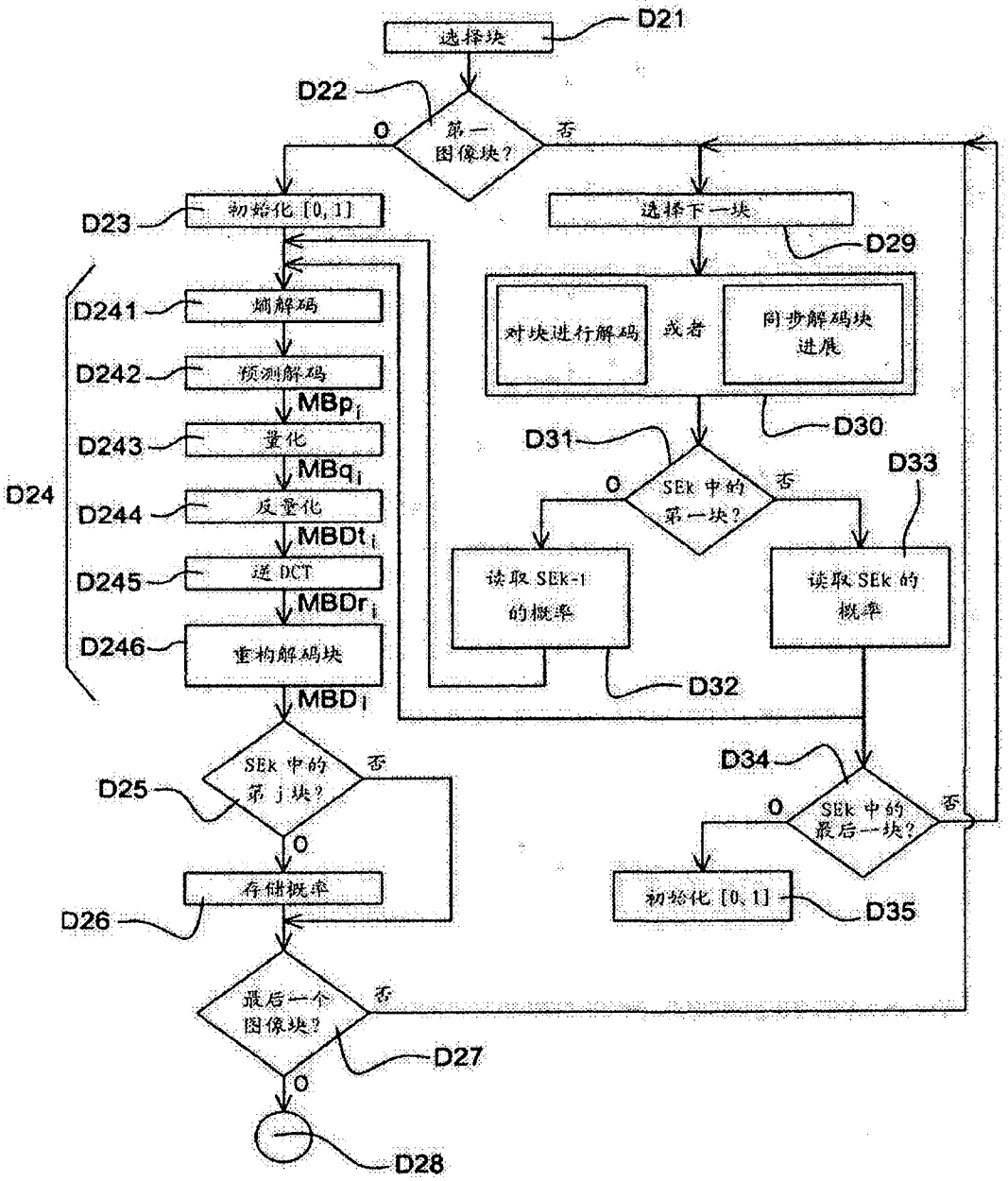


图 5B

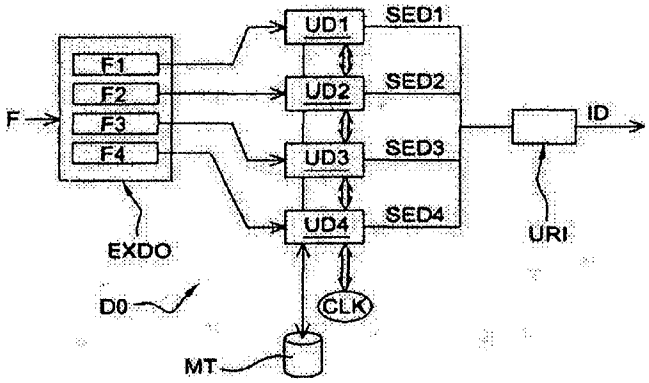


图 6A

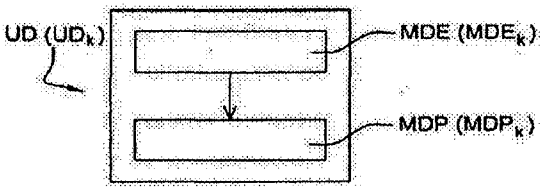


图 6B

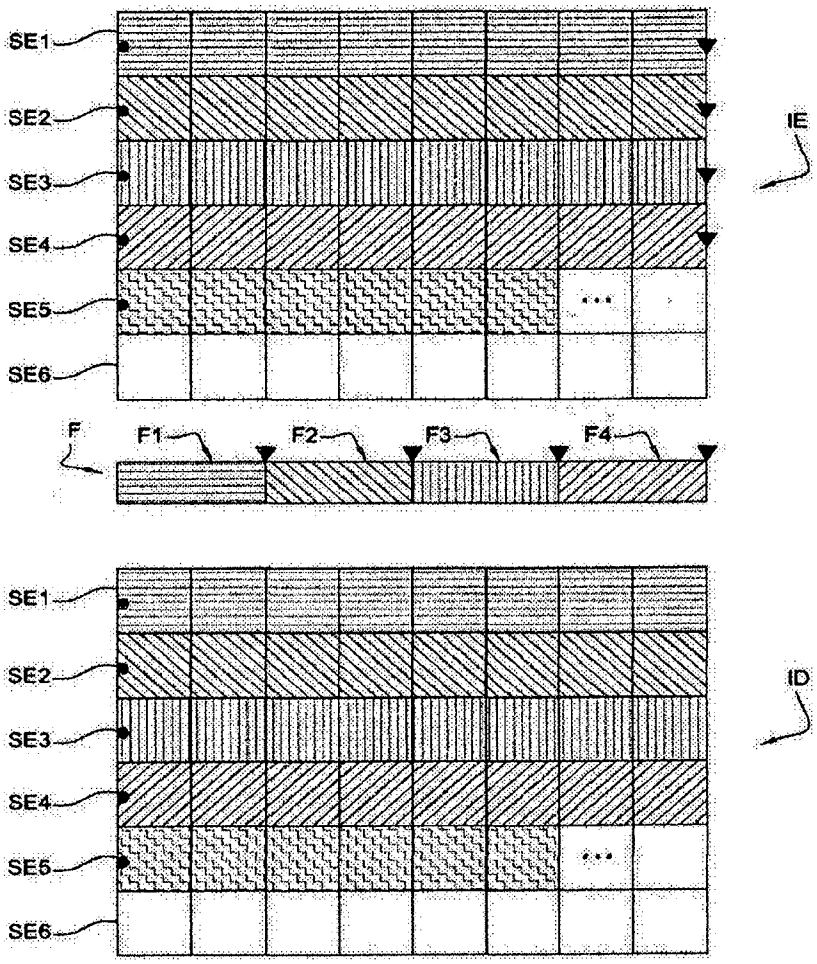


图 7A

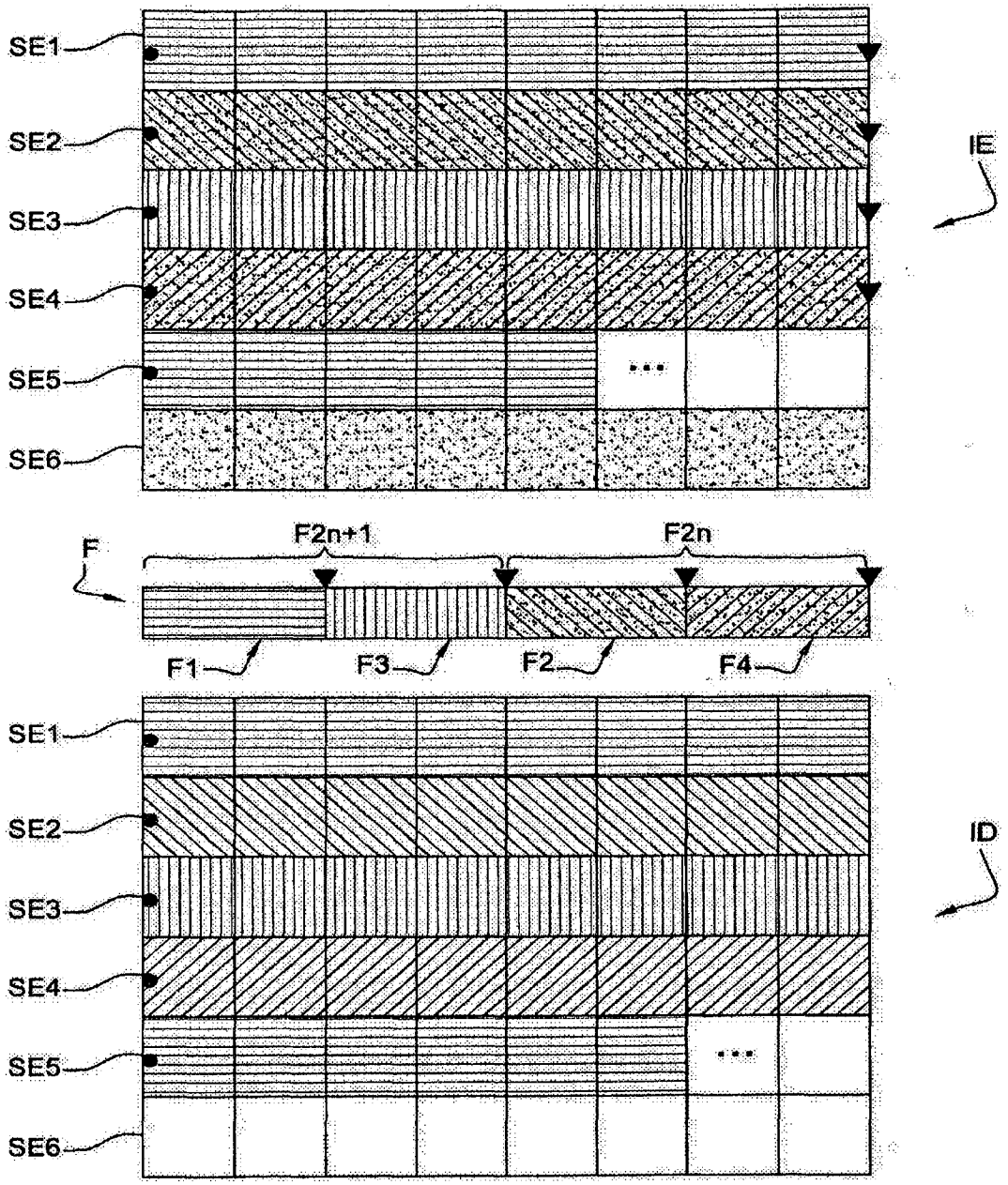


图 7B

IHC160197 ABSTRACT

The present invention discloses a method for encoding and decoding images, encoding and decoding device, and corresponding computer programs. The method for coding includes; segmenting an image into blocks; grouping blocks into a number of subsets; coding, using an entropy coding module, each subset, by associating digital information with symbols of each block of a subset, including, for the first block of the image, initializing state variables of the coding module; and generating a data sub-stream representative of at least one of the coded subsets of blocks. Where a current block is the first block to be coded of a subset, symbol occurrence probabilities for the first current block are determined based on those for a coded and decoded predetermined block of at least one other subset. Where the current block is the last coded block of the subset: writing, in the sub-stream representative of the subset, the entire the digital information associated with the symbols during coding of the blocks of the subset, and implementing the initializing sub-step.