



(12) 发明专利申请

(10) 申请公布号 CN 117156136 A

(43) 申请公布日 2023.12.01

(21) 申请号 202311156237.4

H04N 19/60 (2014.01)

(22) 申请日 2020.02.28

H04N 19/61 (2014.01)

(30) 优先权数据

2019-044275 2019.03.11 JP

H04N 19/126 (2014.01)

H04N 19/157 (2014.01)

(62) 分案原申请数据

202080020388.5 2020.02.28

H04N 19/463 (2014.01)

H04N 19/48 (2014.01)

(71) 申请人 佳能株式会社

H04N 19/176 (2014.01)

地址 日本东京都大田区下丸子3丁目30番2
号

H04N 19/18 (2014.01)

(72) 发明人 志摩真悟

(74) 专利代理机构 北京魏启学律师事务所

11398

专利代理人 魏启学 王小香

(51) Int.Cl.

H04N 19/124 (2014.01)

权利要求书4页 说明书17页 附图13页

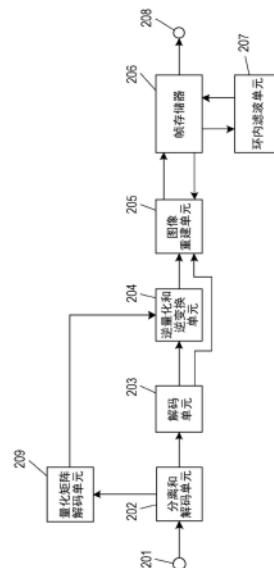
(54) 发明名称

图像解码设备和方法、图像编码设备和方法
以及存储介质

(57) 摘要

本发明提供图像解码设备和方法、图像编码设备和方法以及存储介质。解码单元对与第一量化系数阵列相对应的数据和与第二量化系数阵列相对应的数据进行解码，该第一量化系数阵列包括与第一块相对应的 $N \times M$ 阵列的量化系数，该第二量化系数阵列包括与第二块相对应的 $N \times M$ 阵列的量化系数。逆量化单元通过至少使用 $N \times M$ 阵列的元素的第一量化矩阵来从第一量化系数阵列导出第一正交变换系数阵列，并且通过至少使用 $N \times M$ 阵列的元素的第二量化矩阵来从第二量化系数阵列导出第二正交变换系数阵列。逆正交变换单元对第一正交变换系数阵列进行逆正交变换以生成 $P \times Q$ 阵列的像素的第一预测残差，并且对第二正交变换系数阵列进行逆正交变换以生成 $N \times M$ 阵列的像素的第二预测残差。

A CN 117156136



1. 一种图像解码设备，其能够使用多个块来从位流中解码图像，所述多个块包括 $P \times Q$ 阵列的像素的第一块和 $N \times M$ 阵列的像素的第二块，其中，P和Q是整数，N是满足 $N < P$ 的整数，并且M是满足 $M < Q$ 的整数，所述图像解码设备包括：

解码单元，其被配置为从所述位流中解码与第一量化变换系数阵列相对应的数据和与第二量化变换系数阵列相对应的数据，所述第一量化变换系数阵列与所述第一块相对应，所述第二量化变换系数阵列与所述第二块相对应；

逆量化单元，其被配置为通过使用 $N \times M$ 阵列的元素的第一量化矩阵来从所述第一量化变换系数阵列导出第一变换系数阵列，并且通过使用 $N \times M$ 阵列的元素的第二量化矩阵来从所述第二量化变换系数阵列导出第二变换系数阵列，其中所述第一变换系数阵列表示频率分量，以及其中所述第二变换系数阵列表示频率分量；

逆变换单元，其被配置为通过对所述第一变换系数阵列进行逆变换处理来导出与所述第一块相对应的第一预测残差阵列，并且通过对所述第二变换系数阵列进行逆变换处理来导出与所述第二块相对应的第二预测残差阵列；以及

图片重建单元，其被配置为基于所述第一预测残差阵列和预测图像数据来至少重建图片数据，所述预测图像数据能够通过使用帧内预测而导出，

其中， $N \times M$ 阵列的元素的所述第一量化矩阵是如下的量化矩阵，该量化矩阵包括 $R \times S$ 阵列的元素的第三量化矩阵的一部分元素、但不包括所述第三量化矩阵的其它元素，其中，R是满足 $R \leq N$ 的整数，并且S是满足 $S \leq M$ 的整数，

其中， $N \times M$ 阵列的元素的所述第二量化矩阵是包括 $R \times S$ 阵列的元素的第四量化矩阵的所有元素的量化矩阵，以及

其中，所述第三量化矩阵不同于所述第四量化矩阵。

2. 一种图像解码方法，其能够使用多个块来从位流中解码图像，所述多个块包括 $P \times Q$ 阵列的像素的第一块和 $N \times M$ 阵列的像素的第二块，其中，P和Q是整数，N是满足 $N < P$ 的整数，并且M是满足 $M < Q$ 的整数，所述图像解码方法包括：

从所述位流中解码与第一量化变换系数阵列相对应的数据和与第二量化变换系数阵列相对应的数据，所述第一量化变换系数阵列与所述第一块相对应，所述第二量化变换系数阵列与所述第二块相对应；

通过使用 $N \times M$ 阵列的元素的第一量化矩阵来从所述第一量化变换系数阵列导出第一变换系数阵列，并且通过使用 $N \times M$ 阵列的元素的第二量化矩阵来从所述第二量化变换系数阵列导出第二变换系数阵列，其中所述第一变换系数阵列表示频率分量，以及其中所述第二变换系数阵列表示频率分量；

通过对所述第一变换系数阵列进行逆变换处理来导出与所述第一块相对应的第一预测残差阵列，并且通过对所述第二变换系数阵列进行逆变换处理来导出与所述第二块相对应的第二预测残差阵列；以及

基于所述第一预测残差阵列和预测图像数据来至少重建图片数据，所述预测图像数据能够通过使用帧内预测而导出，

其中， $N \times M$ 阵列的元素的所述第一量化矩阵是如下的量化矩阵，该量化矩阵包括 $R \times S$ 阵列的元素的第三量化矩阵的一部分元素、但不包括所述第三量化矩阵的其它元素，其中，R是满足 $R \leq N$ 的整数，并且S是满足 $S \leq M$ 的整数，

其中, $N \times M$ 阵列的元素的所述第二量化矩阵是包括 $R \times S$ 阵列的元素的第四量化矩阵的所有元素的量化矩阵, 以及

其中, 所述第三量化矩阵不同于所述第四量化矩阵。

3. 根据权利要求2所述的图像解码方法,

其中, 所述第一量化矩阵是除与DC分量相对应的元素以外的元素包括所述第三量化矩阵的所述一部分元素的量化矩阵, 以及

其中, 所述第二量化矩阵是除与DC分量相对应的元素以外的元素包括所述第四量化矩阵的所有元素的量化矩阵。

4. 根据权利要求2所述的图像解码方法,

其中, 所述第一变换系数阵列包括 $N \times M$ 阵列的变换系数, 以及

其中, 所述第二变换系数阵列包括 $N \times M$ 阵列的变换系数。

5. 根据权利要求2所述的图像解码方法, 其中, 所述第一块和所述第二块是非正方形的块。

6. 根据权利要求2所述的图像解码方法,

其中, 所述第一预测残差阵列是 $P \times Q$ 阵列的预测残差, 以及

其中, 所述第二预测残差阵列是 $N \times M$ 阵列的预测残差。

7. 根据权利要求2所述的图像解码方法,

其中, 所述第一变换系数阵列是 $N \times M$ 阵列的变换系数,

其中, 所述第一预测残差阵列是 $P \times Q$ 阵列的预测残差, 以及

其中, 通过对所述第一变换系数阵列进行的逆变换处理来从所述第一变换系数阵列导出所述第一预测残差阵列。

8. 根据权利要求7所述的图像解码方法, 其中, 所述逆变换处理包括所述第一变换系数阵列与 $M \times Q$ 阵列的矩阵的相乘, 以由此导出 $N \times Q$ 阵列的中间值, 并且所述逆变换处理包括 $P \times N$ 阵列的矩阵与所述 $N \times Q$ 阵列的中间值的相乘, 以由此从所述第一变换系数阵列导出所述第一预测残差阵列。

9. 一种存储有程序的非暂时性计算机可读存储介质, 所述程序用于使计算机执行根据权利要求2所述的图像解码方法。

10. 一种图像编码设备, 其能够使用多个块来对图像进行编码, 所述多个块包括 $P \times Q$ 阵列的像素的第一块和 $N \times M$ 阵列的像素的第二块, 其中, P 和 Q 是整数, N 是满足 $N < P$ 的整数, 并且 M 是满足 $M < Q$ 的整数, 所述图像编码设备包括:

变换单元, 其被配置为通过对与所述第一块相对应的第一预测残差阵列进行变换处理来导出第一变换系数阵列, 并且通过对与所述第二块相对应的第二预测残差阵列进行变换处理来导出第二变换系数阵列, 至少所述第一预测残差阵列能够通过使用帧内预测而导出;

量化单元, 其被配置为通过使用 $N \times M$ 阵列的元素的第一量化矩阵对所述第一变换系数阵列进行量化来导出第一量化变换系数阵列, 并且通过使用 $N \times M$ 阵列的元素的第二量化矩阵对所述第二变换系数阵列进行量化来导出第二量化变换系数阵列; 以及

编码单元, 其被配置为对与对应于所述第一块的第一量化变换系数阵列相对应的数据、以及与对应于所述第二块的第二量化变换系数阵列相对应的数据进行编码,

其中, $N \times M$ 阵列的元素的所述第一量化矩阵是如下的量化矩阵, 该量化矩阵包括 $R \times S$ 阵列的元素的第三量化矩阵的一部分元素、但不包括所述第三量化矩阵的其它元素, 其中, R 是满足 $R \leq N$ 的整数, 并且 S 是满足 $S \leq M$ 的整数,

其中, $N \times M$ 阵列的元素的所述第二量化矩阵是包括 $R \times S$ 阵列的元素的第四量化矩阵的所有元素的量化矩阵, 以及

其中, 所述第三量化矩阵不同于所述第四量化矩阵。

11. 一种图像编码方法, 其能够使用多个块来对图像进行编码, 所述多个块包括 $P \times Q$ 阵列的像素的第一块和 $N \times M$ 阵列的像素的第二块, 其中, P 和 Q 是整数, N 是满足 $N < P$ 的整数, 并且 M 是满足 $M < Q$ 的整数, 所述图像编码方法包括:

通过对与所述第一块相对应的第一预测残差阵列进行变换处理来导出第一变换系数阵列, 并且通过对与所述第二块相对应的第二预测残差阵列进行变换处理来导出第二变换系数阵列, 至少所述第一预测残差阵列能够通过使用帧内预测而导出;

通过使用 $N \times M$ 阵列的元素的第一量化矩阵对所述第一变换系数阵列进行量化来导出第一量化变换系数阵列, 并且通过使用 $N \times M$ 阵列的元素的第二量化矩阵对所述第二变换系数阵列进行量化来导出第二量化变换系数阵列; 以及

对与对应于所述第一块的第一量化变换系数阵列相对应的数据、以及与对应于所述第二块的第二量化变换系数阵列相对应的数据进行编码,

其中, $N \times M$ 阵列的元素的所述第一量化矩阵是如下的量化矩阵, 该量化矩阵包括 $R \times S$ 阵列的元素的第三量化矩阵的一部分元素、但不包括所述第三量化矩阵的其它元素, 其中, R 是满足 $R \leq N$ 的整数, 并且 S 是满足 $S \leq M$ 的整数,

其中, $N \times M$ 阵列的元素的所述第二量化矩阵是包括 $R \times S$ 阵列的元素的第四量化矩阵的所有元素的量化矩阵, 以及

其中, 所述第三量化矩阵不同于所述第四量化矩阵。

12. 根据权利要求11所述的图像编码方法,

其中, 所述第一量化矩阵是除与DC分量相对应的元素以外的元素包括所述第三量化矩阵的所述一部分元素的量化矩阵, 以及

其中, 所述第二量化矩阵是除与DC分量相对应的元素以外的元素包括所述第四量化矩阵的所有元素的量化矩阵。

13. 根据权利要求11所述的图像编码方法,

其中, 所述第一变换系数阵列包括 $N \times M$ 阵列的变换系数, 以及

其中, 所述第二变换系数阵列包括 $N \times M$ 阵列的变换系数。

14. 根据权利要求11所述的图像编码方法, 其中, 所述第一块和所述第二块是非正方形的块。

15. 根据权利要求11所述的图像编码方法,

其中, 所述第一预测残差阵列是 $P \times Q$ 阵列的预测残差, 以及

其中, 所述第二预测残差阵列是 $N \times M$ 阵列的预测残差。

16. 根据权利要求11所述的图像编码方法,

其中, 所述第一预测残差阵列是 $P \times Q$ 阵列的预测残差,

其中, 所述第一变换系数阵列是 $N \times M$ 阵列的变换系数, 以及

其中,通过对所述第一预测残差阵列进行的变换处理来从所述第一预测残差阵列导出所述第一变换系数阵列。

17.根据权利要求16所述的图像编码方法,其中,所述变换处理包括所述第一预测残差阵列与 $Q \times M$ 阵列的矩阵的相乘,以由此导出 $P \times M$ 阵列的中间值,并且所述变换处理包括 $N \times P$ 阵列的矩阵与所述 $P \times M$ 阵列的中间值的相乘,以由此从所述第一预测残差阵列导出所述第一变换系数阵列。

18.一种存储有程序的非暂时性计算机可读存储介质,所述程序用于使计算机执行根据权利要求11所述的图像编码方法。

图像解码设备和方法、图像编码设备和方法以及存储介质

[0001] (本申请是申请日为2020年2月28日、申请号为202080020388.5、发明名称为“图像解码设备、图像解码方法和程序”的申请的分案申请。)

技术领域

[0002] 本发明涉及图像编码技术。

背景技术

[0003] 已知有高效率视频编码(HEVC)编码方法(以下称为HEVC)作为用于对运动图像进行压缩的编码方法。HEVC使用大小比传统宏块(16×16 阵列的像素)的大小大的基本块以提高编码效率。大小大的基本块被称为编码树单元(CTU),并且其最大大小为 64×64 阵列的像素。CTU被进一步分割成作为预测和变换的单位的子块。

[0004] 在HEVC中,使用量化矩阵来基于频率分量向正交变换后的系数(以下称为正交变换系数)指派权重。量化矩阵的使用可以通过使与劣化对于人类视觉而言不太明显的高频分量有关的数据比与低频分量有关的数据更多地减少,来在维持图像质量的同时提高压缩效率。日本特开2013-38758论述了用于对指示这种量化矩阵的信息进行编码的技术。

[0005] 最近已经开始作为HEVC的后继者的更加高效编码方法的国际标准化。具体地,由国际标准化组织和国际电工委员会(ISO/IEC)以及ITU电信标准化部门(IUT-T)建立的联合视频专家组(JVET)已经正在推动通用视频编码(VVC)编码方法(以下称为VVC)的标准。对于这种标准化,已经正在研究通过在进行正交变换时的块大小大的情况下将高频分量的正交变换系数强制设置为0来减少代码量以提高编码效率的新技术(以下称为归零)。

[0006] 与HEVC一样,还研究了将量化矩阵引入VVC。然而,HEVC中的量化矩阵基于用于使用具有与传统正交变换的大小相同的大小的量化矩阵来进行量化的方法,并且不能支持作为将正交变换系数的一部分设置为零的新技术的归零。因此,已存在归零的正交变换系数不能进行基于频率分量的量化控制、并且不能提高主观图像质量的问题。

[0007] 现有技术文献

[0008] 专利文献

[0009] 专利文献1:日本特开2013-38758

发明内容

[0010] 例如,提供以下的结构,从而即使使用将正交变换系数的一部分强制设置为0的技术,通过使得能够进行使用与该技术相对应的量化矩阵的量化处理,也能提高主观图像质量。换句话说,一种图像解码设备,其能够以多个块为单位从位流中解码图像,所述多个块包括 $P \times Q$ (P和Q是整数)阵列的像素的第一块和 $N \times M$ (N是满足 $N < P$ 的整数,并且M是满足 $M < Q$ 的整数)阵列的像素的第二块,所述图像解码设备包括:解码单元,其被配置为从所述位流中解码与第一量化系数阵列相对应的数据和与第二量化系数阵列相对应的数据,所述第一量化系数阵列包括与所述第一块相对应的 $N \times M$ 阵列的量化系数,所述第二量化系数阵列包

括与所述第二块相对应的 $N \times M$ 阵列的量化系数；逆量化单元，其被配置为通过至少使用 $N \times M$ 阵列的元素的第一量化矩阵来从所述第一量化系数阵列导出第一正交变换系数阵列，并且通过至少使用 $N \times M$ 阵列的元素的第二量化矩阵来从所述第二量化系数阵列导出第二正交变换系数阵列；以及逆正交变换单元，其被配置为通过对所述第一正交变换系数阵列进行逆正交变换来生成与所述第一块相对应的 $P \times Q$ 阵列的像素的第一预测残差，并且通过对所述第二正交变换系数阵列进行逆正交变换来生成与所述第二块相对应的 $N \times M$ 阵列的像素的第二预测残差，其中， $N \times M$ 阵列的元素的所述第一量化矩阵不同于 $N \times M$ 阵列的元素的所述第二量化矩阵。

[0011] 此外，例如，提供以下的结构。一种图像编码设备，其能够以多个块为单位对图像进行编码，所述多个块包括 $P \times Q$ (P 和 Q 是整数) 阵列的像素的第一块和 $N \times M$ (N 是满足 $N < P$ 的整数，并且 M 是满足 $M < Q$ 的整数) 阵列的像素的第二块，所述图像编码设备包括：正交变换单元，其被配置为通过对与所述第一块相对应的 $P \times Q$ 阵列的像素的预测残差进行正交变换来至少生成第一正交变换系数阵列，并且通过对与所述第二块相对应的 $N \times M$ 阵列的像素的预测残差进行正交变换来生成第二正交变换系数阵列；量化单元，其被配置为通过至少使用 $N \times M$ 阵列的元素的第一量化矩阵对所述第一正交变换系数阵列进行量化来生成包括 $N \times M$ 阵列的量化系数的第一量化系数阵列，并且通过至少使用 $N \times M$ 阵列的元素的第二量化矩阵对所述第二正交变换系数阵列进行量化来生成包括 $N \times M$ 阵列的量化系数的第二量化系数阵列；以及编码单元，其被配置为对与第一量化系数阵列相对应的数据和与第二量化系数阵列相对应的数据进行编码，该第一量化系数阵列包括与所述第一块相对应的 $N \times M$ 阵列的量化系数，该第二量化系数阵列包括与所述第二块相对应的 $N \times M$ 阵列的量化系数，其中， $N \times M$ 阵列的元素的所述第一量化矩阵不同于 $N \times M$ 阵列的元素的所述第二量化矩阵。

附图说明

- [0012] 图1是示出根据第一典型实施例的图像编码设备的结构的框图。
- [0013] 图2是示出根据第二典型实施例的图像解码设备的结构的框图。
- [0014] 图3是示出利用根据第一典型实施例的图像编码设备的图像编码处理的流程图。
- [0015] 图4是示出利用根据第二典型实施例的图像解码设备的图像解码处理的流程图。
- [0016] 图5是示出可应用于根据本发明的图像编码设备和图像解码设备的计算机的硬件结构示例的框图。
- [0017] 图6A是示出在第一典型实施例中输出的位流的示例的图。
- [0018] 图6B是示出在第一典型实施例中输出的位流的示例的图。
- [0019] 图7A是示出第一典型实施例和第二典型实施例中所使用的子块分割的示例的图。
- [0020] 图7B是示出第一典型实施例和第二典型实施例中所使用的子块分割的示例的图。
- [0021] 图7C是示出第一典型实施例和第二典型实施例中所使用的子块分割的示例的图。
- [0022] 图7D是示出第一典型实施例和第二典型实施例中所使用的子块分割的示例的图。
- [0023] 图7E是示出第一典型实施例和第二典型实施例中所使用的子块分割的示例的图。
- [0024] 图7F是示出第一典型实施例和第二典型实施例中所使用的子块分割的示例的图。
- [0025] 图8A是示出第一典型实施例和第二典型实施例中所使用的量化矩阵的示例的图。
- [0026] 图8B是示出第一典型实施例和第二典型实施例中所使用的量化矩阵的示例的图。

- [0027] 图8C是示出第一典型实施例和第二典型实施例中所使用的量化矩阵的示例的图。
- [0028] 图9是示出用于扫描第一典型实施例和第二典型实施例中所使用的量化矩阵的元素的方法的图。
- [0029] 图10是示出第一典型实施例和第二典型实施例中所生成的量化矩阵的差值矩阵的图。
- [0030] 图11A是示出在对量化矩阵的差值进行编码时使用的编码表的示例的图。
- [0031] 图11B是示出在对量化矩阵的差值进行编码时使用的编码表的示例的图。
- [0032] 图12A是示出第一典型实施例和第二典型实施例中所使用的量化矩阵的另一示例的图。
- [0033] 图12B是示出第一典型实施例和第二典型实施例中所使用的量化矩阵的另一示例的图。
- [0034] 图12C是示出第一典型实施例和第二典型实施例中所使用的量化矩阵的另一示例的图。

具体实施方式

[0035] 将参考附图来说明本发明的典型实施例。注意,以下典型实施例中所述的结构仅仅是示例,并且本发明不限于以下典型实施例中所述的结构。为了方便起见,在典型实施例中使用诸如基本块、子块、量化矩阵和基量化矩阵等的术语,并且可以在不改变这些术语的含义的情况下适当地使用其它术语。例如,基本块和子块可被称为基本单元和子单元,或者简称为块或单元。在以下的说明中,如通常所定义的,矩形是指四个内角为直角且两条对角线的长度相同的四边形。如通常所定义的,正方形是指所有的四个角都相同且所有的四条边都相同的矩形或四边形。换句话说,正方形是矩形的一种。

[0036] <第一典型实施例>

[0037] 以下将参考附图来说明本发明的典型实施例。

[0038] 首先将更详细地说明归零。如上所述,归零是用于将要编码的块的正交变换系数的一部分强制设置为0的处理。例如,假定要编码的块是输入图像(图片)中的 64×64 阵列的像素的块。这里,正交变换系数也具有 64×64 阵列的大小。归零是用于例如通过即使正交变换的结果的值不为零也假定 64×64 阵列的正交变换系数的一部分为0、来进行编码的处理。例如,将与二维的正交变换系数中的包括直流(DC)分量的预定左上范围相对应的低频分量从强制归零中排除,并且将与在频率上比低频分量高的频率分量相对应的正交变换系数恒定地设置为0。

[0039] 接着,将说明根据本典型实施例的图像编码设备。图1是示出根据本典型实施例的图像编码设备的框图。在图1中,将图像数据输入到端子101。

[0040] 块分割单元102将输入图像分割成多个基本块,并且将以基本块为单位的图像输出到后级。

[0041] 量化矩阵存储单元103生成并存储量化矩阵。如本文所采用的,量化矩阵旨在基于频率分量来向针对正交变换系数的量化处理指派权重。例如,通过将基于基准参数值(量化参数)的标度值(量化标度)乘以量化矩阵中的各个分量的值,来向以下要说明的量化处理中的针对正交变换系数的量化步长指派权重。

[0042] 没有特别限制用于生成量化矩阵存储单元103中所存储的量化矩阵的方法。例如，用户可以输入指示量化矩阵的信息。图像编码设备可以根据输入图像的特性来计算量化矩阵。可以使用预先指定为初始值的量化矩阵。在本典型实施例中，除图8A所示的 8×8 阵列的基量化矩阵之外，还生成并存储通过扩大该基量化矩阵所生成的图8B和图8C所示的两个类型的 32×32 阵列的二维量化矩阵。图8B的量化矩阵是通过将图8A的 8×8 阵列的基量化矩阵的各元素在垂直和水平方向上重复四次而扩大四倍得到的 32×32 阵列的量化矩阵。另一方面，图8C的量化矩阵是通过将图8A的基量化矩阵的左上方 4×4 阵列部分中的各元素在垂直和水平方向上重复八次而扩大得到的 32×32 阵列的量化矩阵。

[0043] 如上所述，基量化矩阵是不仅用于对 8×8 阵列的像素子块进行量化而且还用于生成在大小上比基量化矩阵大的量化矩阵的量化矩阵。尽管基量化矩阵的大小为 8×8 阵列，但该大小不是限制性的。此外，根据子块大小，可以使用不同的基量化矩阵。例如，如果使用三个类型的子块(即， 8×8 、 16×16 和 32×32 阵列的子块)，则可以使用分别相应的三个类型的基量化矩阵。

[0044] 预测单元104确定与以基本块为单位的图像数据有关的子块分割。更具体地，预测单元104确定是否将基本块分割成子块，并且如果要对基本块进行分割，则确定如何对基本块进行分割。如果不将基本块分割成子块，则如此得到的子块具有与基本块相同的大小。子块可以是正方形或除正方形以外的矩形(非正方形)。

[0045] 然后，预测单元104通过以子块为单位进行作为帧内预测(*intraframe prediction*)的帧内预测(*intra-prediction*)或作为帧间预测(*interframe prediction*)的帧间预测(*inter-prediction*)等来生成预测图像数据。

[0046] 例如，预测单元104从帧内预测和帧间预测中选择要对子块进行的预测方法，并且进行所选择的预测以生成与该子块有关的预测图像数据。注意，要使用的预测方法不限于此，并且可以通过组合帧内预测和帧间预测来进行预测。

[0047] 预测单元104还根据所输入的图像数据和预测图像数据来计算预测残差，并输出这些预测残差。例如，预测单元104计算子块的像素值与通过对该子块的预测所生成的预测图像数据的各个像素值之间的差，并且将这些差计算为预测残差。

[0048] 预测单元104还将预测所需的信息(诸如指示子块的分割状态的信息、指示子块的预测方法的预测模式、以及诸如运动矢量等的信息等)连同预测残差一起输出。预测所需的这些信息在下文将被统称为预测信息。

[0049] 变换和量化单元105以子块为单位对预测单元104所计算出的预测残差进行正交变换，以获得指示预测残差的各个频率分量的正交变换系数。变换和量化单元105还使用量化参数和量化矩阵存储单元103中所存储的量化矩阵来进行量化，以获得作为量化后的正交变换系数的量化系数。进行正交变换的功能和进行量化的功能可被配置为单独的结构。

[0050] 逆量化和逆变换单元106通过使用量化系数和量化矩阵存储单元103中所存储的量化矩阵对从变换和量化单元105输出的量化系数进行逆量化来重建正交变换系数。逆量化和逆变换单元106进一步进行逆正交变换以重建预测残差。用于使用量化矩阵和量化参数来重建(导出)正交变换系数的这种处理将被称为逆量化。进行逆量化的功能和进行逆变换的功能可被配置为单独的结构。图像解码设备导出量化参数所用的信息也由编码单元110编码到位流中。

- [0051] 帧存储器108存储重建的图像数据。
- [0052] 图像重建单元107基于从预测单元104输出的预测信息,通过适当地参考帧存储器108来生成预测图像数据。图像重建单元107根据预测图像数据和所输入的预测残差来生成重建图像数据,并输出该重建图像数据。
- [0053] 环内滤波单元109对重建图像进行诸如解块滤波和采样自适应偏移等的环内滤波处理,并输出滤波处理后的图像。
- [0054] 编码单元110对从变换和量化单元105输出的量化系数以及从预测单元104输出的预测信息进行编码以生成代码数据,并输出该代码数据。
- [0055] 量化矩阵编码单元113对从量化矩阵存储单元103输出的基量化矩阵进行编码,以生成供图像解码设备导出基量化矩阵用的量化矩阵代码数据,并输出该量化矩阵代码数据。
- [0056] 整合和编码单元111通过使用作为来自量化矩阵编码单元113的输出的量化矩阵代码数据来生成头代码数据。整合和编码单元111还将头代码数据与从编码单元110输出的代码数据进行组合以生成位流,并输出该位流。
- [0057] 端子112将整合和编码单元111所生成的位流输出到外部。
- [0058] 现在将说明利用前述图像编码设备的图像编码操作。在本典型实施例中,图像编码设备被配置为以帧为单位输入运动图像数据。此外,在本典型实施例的说明中,块分割单元102被描述为将运动图像数据分割成 64×64 阵列的像素的基本块。然而,这不是限制性的。例如,基本块可以是 128×128 阵列的像素的块。基本块可以是 32×32 阵列的像素的块。
- [0059] 在对图像进行编码之前,图像编码设备生成量化矩阵并对量化矩阵进行编码。在以下的说明中,例如,量化矩阵800或各块中的水平方向将被假定为x轴并且垂直方向将被假定为y轴,其中右方向和下方向为正。量化矩阵800的最左上方元素具有坐标(0, 0)。换句话说, 8×8 阵列的基量化矩阵的最右下方元素具有坐标(7, 7)。 32×32 阵列的量化矩阵的最右下方元素具有坐标(31, 31)。
- [0060] 量化矩阵存储单元103最初生成量化矩阵。基于子块大小、要量化的正交变换系数的大小和预测方法的类型来生成量化矩阵。在本典型实施例中,最初生成要用于生成以下要说明的量化矩阵的图8A所示的 8×8 阵列的基量化矩阵。接着,扩大该基量化矩阵以生成图8B和图8C所示的两个类型的 32×32 阵列的量化矩阵。图8B的量化矩阵是通过将图8A的 8×8 阵列的基量化矩阵在垂直和水平方向上重复四次而放大四倍得到的 32×32 阵列的量化矩阵。
- [0061] 更具体地,在图8B所示的示例中,向 32×32 阵列的量化矩阵中的x坐标为0~3且y坐标为0~3的范围中的各元素指派基量化矩阵的最左上方元素的值1。向 32×32 阵列的量化矩阵中的x坐标为28~31且y坐标为28~31的范围中的各元素指派基量化矩阵的最右下方元素的值15。在图8B的示例中,基量化矩阵中的元素的值各自均被指派给 32×32 阵列的量化矩阵中的元素的一部分。
- [0062] 另一方面,图8C的量化矩阵是通过将图8A的基量化矩阵的左上方 4×4 阵列部分中的各元素在垂直和水平方向上重复八次而放大得到的 32×32 阵列的量化矩阵。
- [0063] 更具体地,在图8C所示的示例中,向 32×32 阵列的量化矩阵中的x坐标为0~7且y

坐标为0~7的范围中的各元素指派基量化矩阵的左上方 4×4 阵列部分中的最左上方元素的值1。向 32×32 阵列的量化矩阵中的x坐标为24~31且y坐标为24~31的范围中的各元素指派基量化矩阵的左上方 4×4 阵列部分中的最右下方元素的值7。在图8C的示例中,基量化矩阵中的元素的值中的仅与左上方 4×4 阵列部分(x坐标为0~3且y坐标为0~3的范围)相对应的元素的值被指派给 32×32 阵列的量化矩阵的元素。

[0064] 注意,要生成的量化矩阵不限于此。如果存在除 32×32 阵列以外的要量化的正交变换系数的大小,则可以生成与要量化的正交变换系数的大小(诸如 16×16 、 8×8 和 4×4 阵列等)相对应的量化矩阵。没有特别限制用于确定构成基量化矩阵和量化矩阵的元素的方法。例如,可以使用预定的初始值。可以单独设置这些元素。可以基于图像特性来生成量化矩阵。

[0065] 如此生成的基量化矩阵和量化矩阵存储在量化矩阵存储单元103中。图8B示出在对与以下要说明的 32×32 阵列的子块相对应的正交变换系数进行量化时要使用的量化矩阵的示例,并且图8C示出在对与 64×64 阵列的子块相对应的正交变换系数进行量化时要使用的量化矩阵的示例。粗框800表示量化矩阵。为了便于说明,各量化矩阵800应包括 32×32 阵列(即,1024个像素),并且粗框中的各正方形应表示构成量化矩阵的元素。在本典型实施例中,图8B和图8C所示的三个类型的量化矩阵是以二维形式存储的。然而,将理解,量化矩阵中的元素不限于此。此外,根据要量化的正交变换系数的大小或者编码对象是亮度块还是色差块,可以针对相同的预测方法存储多个量化矩阵。通常,由于量化矩阵旨在实现基于人类视觉特性的量化处理,因此如图8B和图8C所示,与量化矩阵的左上部分相对应的低频部分中的元素小,并且与右下部分相对应的高频部分中的元素大。

[0066] 量化矩阵编码单元113从量化矩阵存储单元103顺次读取以二维形式存储的基量化矩阵的元素,扫描这些元素以计算差,并将这些差以一维矩阵布置。在本典型实施例中,关于图8A所示的基量化矩阵,使用图9所示的扫描方法来针对各元素计算与按扫描顺序的紧前元素的差。例如,通过图9所示的扫描方法来扫描图8A所示的 8×8 阵列的基量化矩阵。在左上方的第一个元素1之后,接着扫描紧挨地位于下方的元素2,并且计算出差+1。通过计算与预定初始值(例如,8)的差来对量化矩阵的第一个元素(在本典型实施例中为1)进行编码。然而,这不是限制性的,并且可以使用与任意值的差或者第一个元素自身的值。

[0067] 在本典型实施例中,如此通过使用图9的扫描方法根据图8A的基量化矩阵来生成图10所示的差分矩阵。量化矩阵编码单元113进一步对差分矩阵进行编码以生成量化矩阵代码数据。在本典型实施例中,通过使用图11A所示的编码表来进行编码,而编码表不限于此。例如,可以使用图11B所示的编码表。由此生成的量化矩阵代码数据被输出到后级的整合和编码单元111。

[0068] 返回图1,整合和编码单元111对图像数据的编码所需的头信息进行编码,并将量化矩阵上的代码数据与编码后的头信息整合。

[0069] 然后,对图像数据进行编码。从端子101输入的一帧图像数据被输入到块分割单元102。

[0070] 块分割单元102将所输入的图像数据分割成多个基本块,并将以基本块为单位的图像输出到预测单元104。在本典型实施例中,图像是以 64×64 阵列的像素的基本块为单位输出的。

[0071] 预测单元104对从块分割单元102输入的以基本块为单位的图像数据进行预测处理。具体地，预测单元104确定与是否将基本块分割成更小子块有关的子块分割，并且还以子块为单位确定诸如帧内预测和帧间预测等的预测模式。

[0072] 图7A至图7F示出子块分割方法的示例。粗框700表示基本块。为了便于描述，基本块应包括 64×64 阵列的像素，并且该粗框中的各四边形应表示子块。图7B示出四叉树正方形子块分割的示例。将 64×64 阵列的像素的基本块分割成 32×32 阵列的像素子块。图7C至图7F示出矩形子块分割的示例。在图7C中，基本块被分割成 32×64 阵列的像素的纵长矩形子块。在图7D中，基本块被分割成 64×32 阵列的像素的横长矩形子块。在图7E和图7F中，基本块按1:2:1的比被分割成矩形子块。以这种方式，通过不仅使用正方形子块而且还使用非正方形的矩形子块来进行编码处理。此外，可以将基本块进一步分割成多个正方形块，并且可以以分割得到的正方形块为基准进行子块分割。换句话说，基本块的大小不限于 64×64 阵列的像素，并且可以使用多个大小的基本块。

[0073] 在本典型实施例中， 64×64 阵列的像素的基本块如图7A那样未被分割，或者通过仅使用如图7B那样的四叉树分割而被分割。然而，子块分割方法不限于此。可以使用如图7E和图7F那样的三叉树分割以及如图7C和图7D那样的二叉树分割。如果还使用除图7A和图7B的子块分割以外的子块分割，则量化矩阵存储单元103生成与要使用的子块相对应的量化矩阵。如果还生成了与所生成的量化矩阵相对应的一个或多个新的基量化矩阵，则量化矩阵编码单元113也对这一个或多个新的基量化矩阵进行编码。

[0074] 将更详细地说明本典型实施例中所使用的预测单元104的预测方法。在本典型实施例中，例如使用帧内预测和帧间预测这两个类型的预测方法。帧内预测通过使用在空间上位于要编码的块周围的已编码像素来生成要编码的块的预测像素，并且还生成指示诸如水平预测、垂直预测和DC预测等的帧内预测方法中的所使用的帧内预测方法的帧内预测模式信息。帧间预测通过使用在时间上与要编码的块不同的一个或多个帧中的已编码像素来生成要编码的块的预测像素，并且还生成指示要参考的(一个或多个)帧和运动矢量等的运动信息。如上所述，预测单元104可以使用组合帧内预测和帧间预测的预测方法。

[0075] 基于所确定的预测模式和已编码像素来生成预测图像数据。进一步根据所输入的图像数据和预测图像数据来生成预测残差，并将这些预测残差输出到变换和量化单元105。将与子块分割和预测模式等有关的信息作为预测信息输出到编码单元110和图像重建单元107。

[0076] 变换和量化单元105对所输入的预测残差进行正交变换和量化以生成量化系数。变换和量化单元105首先应用与子块大小相对应的正交变换处理以生成正交变换系数。然后，变换和量化单元105通过基于预测模式使用量化矩阵存储单元103中所存储的量化矩阵对正交变换系数进行量化来生成量化系数。以下说明更具体的正交变换和量化处理。

[0077] 如果选择图7B所示的 32×32 阵列的子块分割，则变换和量化单元105将使用 32×32 阵列的正交变换矩阵的正交变换应用于 32×32 阵列的预测残差以生成 32×32 阵列的正交变换系数。具体地，变换和量化单元105进行以离散余弦变换(DCT)为代表的 32×32 阵列的正交变换矩阵与 32×32 阵列的预测残差的相乘，以计算采用 32×32 阵列的矩阵的形式的中间系数。变换和量化单元105进一步进行采用 32×32 阵列的矩阵的形式的中间系数与前述 32×32 阵列的正交变换矩阵的转置的相乘，以生成 32×32 阵列的正交变换系数。变换和

量化单元105通过使用量化参数和图8B所示的 32×32 阵列的量化矩阵对所生成的 32×32 阵列的正交变换系数进行量化来生成 32×32 阵列的量化系数。由于 64×64 阵列的基本块包括 32×32 阵列的四个子块,因此将这种处理重复四次。

[0078] 另一方面,如果选择图7A所示的 64×64 阵列的分割状态(未分割),则对于 64×64 阵列的预测残差,使用通过对 64×64 阵列的正交变换矩阵中的奇数编号的行(以下称为奇数行)进行间隔剔除所生成的 64×32 阵列的正交变换矩阵。换句话说,通过应用使用通过对奇数行进行间隔剔除所生成的 64×32 阵列的正交变换矩阵的正交变换来生成 32×32 阵列的正交变换系数。

[0079] 具体地,变换和量化单元105对 64×64 阵列的正交变换矩阵中的奇数行进行间隔剔除,以生成 64×32 阵列的正交变换矩阵。然后,变换和量化单元105进行 64×32 阵列的正交变换矩阵与 64×64 阵列的预测残差的相乘,以生成采用 64×32 阵列的矩阵的形式的中间系数。变换和量化单元105进行采用 64×32 阵列的矩阵的形式的中间系数与通过对前述 64×32 阵列的正交变换矩阵进行转置所获得的 32×64 阵列的转置的相乘,以生成 32×32 阵列的正交变换系数。然后,变换和量化单元105通过使用所生成的 32×32 阵列的正交变换系数作为 64×64 阵列的正交变换系数的左上部分(x坐标为0~31且y坐标为0~31的范围)中的系数、并将其它系数设置为0,来进行归零。

[0080] 在本典型实施例中,如此通过使用 64×32 阵列的正交变换矩阵和通过对 64×32 阵列的正交变换矩阵进行转置所获得的 32×64 阵列的转置来对 64×64 阵列的预测残差进行正交变换。通过以这种方式生成 32×32 阵列的正交变换系数来进行归零。因此,与如果通过 64×64 阵列的正交变换所生成的 64×64 阵列的正交变换系数的一部分的值不为0、则将该部分强制设置为0的技术相比,可以以较少的计算量生成 32×32 阵列的正交变换系数。换句话说,与以下情况相比,可以减少正交变换的计算量:使用 64×64 阵列的正交变换矩阵来进行正交变换,并且通过不论作为归零的对象的正交变换系数是否为0都假定该正交变换系数为0来对结果进行编码。注意,尽管通过使用正交变换来根据 64×64 阵列的预测残差计算 32×32 阵列的正交变换系数的方法可以减少计算量,但用于归零的方法不限于此,并且可以使用各种方法。

[0081] 在执行归零时,可以对指示归零的对象范围中的正交变换系数为0的信息进行编码,或者可以简单地对指示归零的执行的信息(诸如标志等)进行编码。通过对这种信息进行解码,图像解码设备可以在归零的对象作为0的情况下对块进行解码。

[0082] 接着,变换和量化单元105通过使用量化参数和图8C所示的 32×32 阵列的量化矩阵对如此生成的 32×32 阵列的正交变换系数进行量化,来生成 32×32 阵列的量化系数。

[0083] 在本典型实施例中,对于与 32×32 阵列的子块相对应的 32×32 阵列的正交变换系数,使用图8B的量化矩阵,并且对于与 64×64 阵列的子块相对应的 32×32 阵列的正交变换系数,使用图8C的量化矩阵。换句话说,对于未进行归零的 32×32 阵列的正交变换系数,使用图8B,并且对于与进行了归零的 64×64 阵列的子块相对应的 32×32 阵列的正交变换系数,使用图8C的量化矩阵。注意,要使用的量化矩阵不限于此。所生成的量化系数被输出到编码单元110以及逆量化和逆变换单元106。

[0084] 逆量化和逆变换单元106通过使用量化参数和量化矩阵存储单元103中所存储的量化矩阵对所输入的量化系数进行逆量化来重建正交变换系数。逆量化和逆变换单元106

进一步对重建的正交变换系数进行逆正交变换以重建预测残差。与变换和量化单元105一样,逆量化和逆变换单元106使用与要编码的子块的大小相对应的量化矩阵来进行逆量化处理。以下说明利用逆量化和逆变换单元106的更详细的逆量化和逆正交变换处理。

[0085] 如果选择图7B的 32×32 阵列的子块分割,则逆量化和逆变换单元106通过使用图8B的量化矩阵对变换和量化单元105所生成的 32×32 阵列的量化系数进行逆量化来重建 32×32 阵列的正交变换系数。然后,逆量化和逆变换单元106进行前述 32×32 阵列的转置与 32×32 阵列的正交变换系数的相乘,以计算采用 32×32 阵列的矩阵的形式的中间系数。然后,逆量化和逆变换单元106进行采用 32×32 阵列的矩阵的形式的中间系数与前述 32×32 阵列的正交变换矩阵的相乘,以重建 32×32 阵列的预测残差。逆量化和逆变换单元106对 32×32 阵列的各子块进行类似的处理。另一方面,如果如图7A那样选择未分割,则逆量化和逆变换单元106通过使用图8C的量化矩阵对变换和量化单元105所生成的 32×32 阵列的量化系数进行逆量化来重建 32×32 阵列的正交变换系数。然后,逆量化和逆变换单元106进行前述 32×64 阵列的转置与 32×32 阵列的正交变换系数的相乘,以计算采用 32×64 阵列的矩阵的形式的中间系数。逆量化和逆变换单元106进行采用 32×64 阵列的矩阵的形式的中间系数与前述 64×32 阵列的正交变换矩阵的相乘,以重建 64×64 阵列的预测残差。在本典型实施例中,基于子块的大小,通过使用与在变换和量化单元105中使用的量化矩阵相同的量化矩阵来进行逆量化处理。重建的预测残差被输出到图像重建单元107。

[0086] 图像重建单元107基于从预测单元104输入的预测信息,通过适当地参考帧存储器108中所存储的重建预测图像所需的数据来重建预测图像。然后,图像重建单元107根据重建的预测图像以及从逆量化和逆变换单元106输入的重建的预测残差来重建图像数据,并将该图像数据输入和存储到帧存储器108中。

[0087] 环内滤波单元109从帧存储器108读取重建图像,并进行诸如解块滤波等的环内滤波处理。环内滤波单元109将滤波处理后的图像再次输入并存储到帧存储器108中。

[0088] 编码单元110以块为单位对变换和量化单元105所生成的量化系数以及从预测单元104输入的预测信息进行熵编码,以生成代码数据。没有特别指定熵编码的方法,并且可以使用Golomb编码、算术编码和霍夫曼编码等。所生成的代码数据被输出到整合和编码单元111。

[0089] 整合和编码单元111通过复用前述头代码数据和从编码单元110输入的代码数据等来生成位流。该位流最终从端子112输出到外部。

[0090] 图6A示出在第一典型实施例中输出的位流的示例。序列头包括基量化矩阵上的代码数据,该代码数据包括各个元素的编码结果。注意,对基量化矩阵上的代码数据等进行编码的位置不限于此,并且将理解,位流可被配置成使得将这种数据编码在图片头部或其它头部中。如果在序列内切换量化矩阵,则可以对新的基量化矩阵进行编码以进行更新。在这种情况下,可以重写所有的量化矩阵,或者可以通过指定与要重写的量化矩阵相对应的量化矩阵的子块大小来改变这些量化矩阵的一部分。

[0091] 图3是示出利用根据第一典型实施例的图像编码设备的编码处理的流程图。

[0092] 在步骤S301中,在图像编码之前,量化矩阵存储单元103首先生成二维量化矩阵并存储这些二维量化矩阵。在本典型实施例中,量化矩阵存储单元103生成图8A所示的基量化矩阵以及根据该基量化矩阵所生成的图8B和图8C所示的量化矩阵,并且存储该基量化矩阵

和这些量化矩阵。

[0093] 在步骤S302中,量化矩阵编码单元113扫描在步骤S301中生成量化矩阵时使用的基量化矩阵以计算按扫描顺序邻接的元素之间的差,并且生成一维差分矩阵。在本典型实施例中,关于图8A所示的基量化矩阵,使用图9的扫描方法来生成图10所示的差分矩阵。量化矩阵编码单元113进一步对所生成的差分矩阵进行编码以生成量化矩阵代码数据。

[0094] 在步骤S303中,整合和编码单元111将对图像数据进行编码所需的头信息连同所生成的量化矩阵代码数据一起编码,并输出结果。

[0095] 在步骤S304中,块分割单元102将以帧为单位的输入图像分割成 64×64 阵列的像素的基本块。

[0096] 在步骤S305中,预测单元104通过使用上述预测方法对在步骤S304中生成的以基本块为单位的图像数据进行预测处理,来生成诸如子块分割信息和预测模式等的预测信息、以及预测图像数据。在本典型实施例中,使用两个类型的块大小,即图7B所示的 32×32 阵列的像素子块的块大小和图7A所示的 64×64 阵列的像素子块的块大小。预测单元104进一步根据所输入的图像数据和预测图像数据来计算预测残差。

[0097] 在步骤S306中,变换和量化单元105对在步骤S305中计算出的预测残差进行正交变换以生成正交变换系数。变换和量化单元105进一步通过使用量化参数和在步骤S301中生成并存储的量化矩阵进行量化来生成量化系数。具体地,变换和量化单元105通过使用 32×32 阵列的正交变换矩阵及其转置来对图7B所示的 32×32 阵列的像素子块的预测残差进行相乘,以生成 32×32 阵列的正交变换系数。另一方面,变换和量化单元105通过使用 64×32 阵列的正交变换矩阵及其转置来对图7A所示的 64×64 阵列的像素子块的预测残差进行相乘,以生成 32×32 阵列的正交变换系数。在本典型实施例中,变换和量化单元105通过将图8B的量化矩阵用于图7B所示的 32×32 阵列的子块的正交变换系数、并且通过将图8C的量化矩阵用于图7A所示的 64×64 阵列的子块的正交变换系数,来对 32×32 阵列的正交变换系数进行量化。

[0098] 在步骤S307中,逆量化和逆变换单元106通过使用量化参数和在步骤S301中生成并存储的量化矩阵对在步骤S306中生成的量化系数进行逆量化,来重建正交变换系数。逆量化和逆变换单元106进一步对正交变换系数进行逆正交变换以重建预测残差。在本步骤中,通过使用与步骤S306中所使用的量化矩阵相同的量化矩阵来进行逆量化处理。具体地,对于与图7B的 32×32 阵列的像素子块相对应的 32×32 阵列的量化系数,逆量化和逆变换单元106使用图8B的量化矩阵来进行逆量化处理,以重建 32×32 阵列的正交变换系数。然后,逆量化和逆变换单元106通过使用 32×32 阵列的正交变换矩阵及其转置来对 32×32 阵列的正交变换系数进行相乘,以重建 32×32 阵列的像素的预测残差。另一方面,对于与图7A的 64×64 阵列的像素的子块相对应的 32×32 阵列的量化系数,逆量化和逆变换单元106使用图8C的量化矩阵来进行逆量化处理,以重建 32×32 阵列的正交变换系数。然后,逆量化和逆变换单元106通过使用 64×32 阵列的正交变换矩阵及其转置来对 32×32 阵列的正交变换系数进行相乘,以重建 64×64 阵列的像素的预测残差。

[0099] 在步骤S308中,图像重建单元107基于在步骤S305中生成的预测信息来重建预测图像。图像重建单元107进一步根据重建的预测图像和在步骤S307中生成的预测残差来重建图像数据。

[0100] 在步骤S309中,编码单元110对在步骤S305中生成的预测信息和在步骤S306中生成的量化系数进行编码以生成代码数据。编码单元110通过还包括其它代码数据来生成位流。

[0101] 在步骤S310中,图像编码设备判断帧中的所有基本块是否都被编码。如果所有基本块都被编码,则处理进入步骤S311。如果并非所有基本块都被编码,则处理以下一块作为对象返回到步骤S304。

[0102] 在步骤S311中,环内滤波单元109对在步骤S308中重建的图像数据进行环内滤波处理以生成滤波处理后的图像。处理结束。

[0103] 通过上述结构和操作,可以在减少计算量的同时控制各频率分量的量化以提高主观图像质量。特别地,在步骤S306中,通过减少正交变换系数的数量并使用与减少的正交变换系数相对应的量化矩阵进行量化处理,可以在减少计算量的同时控制各频率分量的量化以提高主观图像质量。此外,如果减少正交变换系数的数量以仅对低频部分进行量化和编码,则可以通过使用如图8C那样的通过仅扩大基量化矩阵的低频部分所获得的量化矩阵来实现对低频部分最佳的量化控制。在图8C的示例中,低频部分在这里是指x坐标为0~3且y坐标为0~3的范围。

[0104] 在本典型实施例中,为了减少代码量,量化矩阵编码单元113被配置为仅对在生成图8B和图8C的量化矩阵时共同使用的图8A的基量化矩阵进行编码。然而,量化矩阵编码单元113可被配置为对图8B和图8C的量化矩阵自身进行编码。由于可以针对量化矩阵的各个频率分量设置独特值,因此这使得能够对各频率分量进行更精细的量化控制。可选地,可以针对图8B和图8C的量化矩阵设置各自不同的基量化矩阵,并且量化矩阵编码单元113可被配置为对各个基量化矩阵进行编码。在这种情况下,可以对 32×32 阵列的正交变换系数和 64×64 阵列的正交变换系数进行各自不同的量化控制,以实现对主观图像质量的更周密的控制。此外,在这种情况下,可以通过代替将 8×8 阵列的基量化矩阵的左上方 4×4 阵列部分扩大八倍、而是将 8×8 阵列的整个基量化矩阵扩大四倍来获得与 64×64 阵列的正交变换系数相对应的量化矩阵。这也使得能够对 64×64 阵列的正交变换系数进行更精细的量化控制。

[0105] 此外,尽管本典型实施例被配置成使得针对要归零的 64×64 阵列的子块的量化矩阵是唯一地确定的,但可以引入标识符以使得能够进行选择。例如,图6B示出如下的位流:新引入量化矩阵编码方法信息代码,以使要归零的 64×64 阵列的子块的量化矩阵编码是可选择的。例如,如果量化矩阵编码方法信息代码指示0,则将作为独立量化矩阵的图8C用于与要归零的 64×64 阵列的像素的子块相对应的正交变换系数。如果编码方法信息代码指示1,则将作为针对不要归零的普通子块的量化矩阵的图8B用于要归零的 64×64 阵列的像素的子块。如果编码方法信息代码指示2,则代替 8×8 阵列的基量化矩阵的元素,对针对要归零的 64×64 阵列的像素的子块所要使用的量化矩阵的所有元素进行编码。这样可以以可选择的方式实现量化矩阵代码的量减少和对要归零的子块的独特量化控制。

[0106] 在本典型实施例中,通过归零所处理的子块仅是 64×64 阵列的子块。然而,通过归零所要处理的子块不限于此。例如,在与图7C或图7D所示的 32×64 或 64×32 阵列的子块相对应的正交变换系数中,可以将下半部分或右半部分的 32×32 阵列的正交变换系数强制设置为0。在这种情况下,仅对上半部分或左半部分的 32×32 阵列的正交变换系数进行量化和

编码。通过使用与图8B的量化矩阵不同的量化矩阵来进行对上半部分或左半部分的 32×32 阵列的正交变换系数的量化处理。

[0107] 此外,可以以与 8×8 阵列的基矩阵的元素的值分开的方式设置和编码与所生成的正交变换系数中的被认为对图像质量的影响最大的左上方的DC系数相对应的量化矩阵的值。图12B和图12C示出与DC分量相对应的最左上方元素的值从图8B和图8C中的值改变的示例。在这种情况下,可以通过除对与图8A的基量化矩阵有关的信息进行编码以外、还对DC部分的位置处的指示“2”的信息进行编码,来设置图12B和图12C所示的量化矩阵。这使得能够针对对图像质量的影响最大的正交变换系数的DC分量进行更精细的量化控制。

[0108] <第二典型实施例>

[0109] 图2是示出根据本发明第二典型实施例的图像解码设备的结构的框图。将通过使用用于对在第一典型实施例中生成的编码数据进行解码的图像解码设备作为示例来说明本典型实施例。

[0110] 将编码的位流输入到端子201。

[0111] 分离和解码单元202将位流分离成与解码处理有关的信息和与系数相关的代码数据,并且对位流的头部中所包括的代码数据进行解码。在本典型实施例中,分离和解码单元202分离出量化矩阵代码,并将该量化矩阵代码输出到后级。分离和解码单元202进行与图1的整合和编码单元111的操作相反的操作。

[0112] 量化矩阵解码单元209进行以下处理,该处理用于从位流中解码量化矩阵代码以重建基量化矩阵,并且进一步根据该基量化矩阵生成量化矩阵。

[0113] 解码单元203对从分离和解码单元202输出的代码数据进行解码,以重建(导出)量化系数和预测信息。

[0114] 逆量化和逆变换单元204与图1的逆量化和逆变换单元106一样,通过使用量化参数和重建的量化矩阵对量化系数进行逆量化来获得正交变换系数,并且进一步进行逆正交变换以重建预测残差。用于导出量化参数的信息也是由解码单元203从位流中解码得到的。进行逆量化的功能和进行逆变换的功能可被配置为单独的结构。

[0115] 帧存储器206存储与重建的图片有关的图像数据。

[0116] 图像重建单元205基于所输入的预测信息通过适当地参考帧存储器206来生成预测图像数据。然后,图像重建单元205根据该预测图像数据和由逆量化和逆变换单元204重建的预测残差来生成重建图像数据,并输出该重建图像数据。

[0117] 环内滤波单元207与图1的109一样对重建图像进行诸如解块滤波等的环内滤波处理,并输出滤波处理后的图像。

[0118] 端子208将重建的图像数据输出到外部。

[0119] 以下将说明前述图像解码设备的图像解码操作。在本典型实施例中,图像解码设备被配置为按帧(以图片为单位)输入在第一典型实施例中生成的位流。

[0120] 在图2中,从端子201输入的1帧位流被输入到分离和解码单元202。分离和解码单元202将位流分离成与解码处理有关的信息和与系数相关的代码数据,并且对位流的头部中所包括的代码数据进行解码。更具体地,分离和解码单元202重建量化矩阵代码数据。在本典型实施例中,分离和解码单元202首先从图6A所示的位流的序列头提取量化矩阵代码数据,并将该量化矩阵代码数据输出到量化矩阵解码单元209。在本典型实施例中,提取并

输出与图8A所示的基量化矩阵相对应的量化矩阵代码数据。然后,分离和解码单元202重建以基本块为单位的图片数据的代码数据,并将该代码数据输出到解码单元203。

[0121] 量化矩阵解码单元209首先对所输入的量化矩阵代码数据进行解码,以重建图10所示的一维差分矩阵。在本典型实施例中,与第一典型实施例一样,通过使用图11A所示的编码表来对量化矩阵代码数据进行解码。然而,编码表不限于此,并且可以使用其它编码表,只要在第一典型实施例中使用相同的编码表即可。量化矩阵解码单元209进一步根据重建的一维差分矩阵重建二维量化矩阵。这里,量化矩阵解码单元209进行与根据第一典型实施例的量化矩阵编码单元113的操作相反的操作。更具体地,在本典型实施例中,关于图10所示的差分矩阵,通过使用图9所示的扫描方法来重建并存储图8A所示的基量化矩阵。具体地,量化矩阵解码单元209将差分矩阵中的差值顺次地与前述初始值相加以重建量化矩阵的元素。然后,量化矩阵解码单元209通过基于图9所示的扫描方法将重建的一维元素按顺序与二维量化矩阵的各个元素关联来重建二维量化矩阵。

[0122] 量化矩阵解码单元209如在第一典型实施例中那样进一步扩大重建的基量化矩阵以生成图8B和图8C所示的两个类型的 32×32 阵列的量化矩阵。图8B的量化矩阵是通过将图8A的 8×8 阵列的基量化矩阵的各元素在垂直和水平方向上重复四次而放大四倍得到的 32×32 阵列的量化矩阵。

[0123] 另一方面,图8C的量化矩阵是通过将图8A的基量化矩阵的左上方 4×4 阵列部分的各元素在垂直和水平方向上重复八次而放大得到的 32×32 阵列的量化矩阵。注意,要生成的量化矩阵不限于此。如果存在除 32×32 以外的、在后级要逆量化的量化系数的任何大小,则可以生成诸如 16×16 、 8×8 和 4×4 阵列等的与要逆量化的量化系数的大小相对应的阵列量化矩阵。所生成的量化矩阵被存储并用在后级的逆量化处理中。

[0124] 解码单元203从位流中解码代码数据,并重建量化系数和预测信息。解码单元203基于解码得到的预测信息来确定(一个或多个)要解码的子块的大小,并且进一步将重建的量化系数输出到逆量化和逆变换单元204,并将重建的预测信息输出到图像重建单元205。在本典型实施例中,不论子块的大小如何(即,大小是如图7A那样的 64×64 阵列还是如图7B那样的 32×32 阵列),对于各要解码的子块都重建 32×32 阵列的量化系数。

[0125] 逆量化和逆变换单元204通过使用量化参数和由量化矩阵解码单元209重建的量化矩阵对所输入的量化系数进行逆量化来生成正交变换系数,并且进一步应用逆正交变换以重建预测残差。以下说明更详细的逆量化和逆正交变换处理。

[0126] 如果选择图7B的 32×32 阵列的子块分割,则通过使用图8B的量化矩阵来对由解码单元203重建的 32×32 阵列的量化系数进行逆量化,以重建 32×32 阵列的正交变换系数。然后,将前述 32×32 阵列的转置和 32×32 阵列的正交变换系数相乘以计算采用 32×32 阵列的矩阵的形式的中间系数。将采用 32×32 阵列的矩阵的形式的中间系数与前述 32×32 阵列的正交变换矩阵相乘以重建 32×32 阵列的预测残差。对 32×32 阵列的各子块进行类似的处理。

[0127] 另一方面,如果如图7A那样选择未分割,则通过使用图8C的量化矩阵来对由解码单元203重建的量化系数的阵列进行逆量化,以重建 32×32 阵列的正交变换系数。将前述 32×64 阵列的转置和 32×32 阵列的正交变换系数相乘,以计算采用 32×64 阵列的矩阵的形式的中间系数。将采用 32×64 阵列的矩阵的形式的中间系数与前述 64×32 阵列的正交变换矩

阵相乘以重建 64×64 阵列的预测残差。

[0128] 重建的预测残差被输出到图像重建单元205。在本典型实施例中,基于通过由解码单元203重建的预测信息所确定的要解码的子块的大小来确定在逆量化处理中要使用的量化矩阵。更具体地,对于图7B中的 32×32 阵列的各个子块,在逆量化处理中使用图8B的量化矩阵。对于图7A的 64×64 阵列的子块,使用图8C的量化矩阵。注意,要使用的量化矩阵不限于此,只要第一典型实施例中的变换和量化单元105以及逆量化和逆变换单元106使用相同的量化矩阵即可。

[0129] 图像重建单元205基于从解码单元203输入的预测信息,通过适当地参考帧存储器206来获得重建预测图像所需的数据,并且重建预测图像。在本典型实施例中,图像重建单元205与第一典型实施例的预测单元104一样使用帧内预测和帧间预测这两个类型的预测方法。如上所述,可以使用组合帧内预测和帧间预测的预测方法。与第一典型实施例一样,以子块为单位进行预测处理。

[0130] 由于具体预测处理与根据第一典型实施例的预测单元104的预测处理类似,因此将省略对预测处理的说明。图像重建单元205根据通过预测处理所生成的预测图像和从逆量化和逆变换单元204输入的预测残差来重建图像数据。具体地,图像重建单元205通过将预测图像和预测残差相加来重建图像数据。重建的图像数据适当地存储在帧存储器206中。在预测其它子块时适当地参考所存储的图像数据。

[0131] 与图1的环内滤波单元109一样,环内滤波单元207从帧存储器206读取重建图像,并进行诸如解块滤波等的环内滤波处理。滤波处理后的图像再次被输入到帧存储器206中。

[0132] 帧存储器206中所存储的重建图像最终从端子208被输出到外部。例如,重建图像被输出到外部显示装置等。

[0133] 图4是示出利用根据第二典型实施例的图像解码设备的图像解码处理的流程图。

[0134] 在步骤S401中,分离和解码单元202首先将位流分离成与解码处理有关的信息和与系数相关的代码数据,并对头部分中的代码数据进行解码。更具体地,分离和解码单元202重建量化矩阵代码数据。

[0135] 在步骤S402中,量化矩阵解码单元209首先对在步骤S401中重建的量化矩阵代码数据进行解码,以重建图10所示的一维差分矩阵。接着,量化矩阵解码单元209根据重建的一维差分矩阵重建二维基量化矩阵。量化矩阵解码单元209进一步扩大重建的二维基量化矩阵以生成量化矩阵。

[0136] 更具体地,在本典型实施例中,量化矩阵解码单元209通过使用图9所示的扫描方法来根据图10所示的差分矩阵重建图8A所示的基量化矩阵。量化矩阵解码单元209进一步扩大重建的基量化矩阵以生成图8B和图8C所示的量化矩阵,并存储这些量化矩阵。

[0137] 在步骤S403中,解码单元203对在步骤S401中分离出的代码数据进行解码,以重建量化系数和预测信息。解码单元203进一步基于解码得到的预测信息来确定(一个或多个)要解码的子块的大小。在本典型实施例中,不论子块的大小如何(即,大小是如图7A那样的 64×64 阵列还是如图7B那样的 32×32 阵列),对于各要解码的子块都重建 32×32 阵列的量化系数。

[0138] 在步骤S404中,逆量化和逆变换单元204通过使用在步骤S402中重建的量化矩阵对量化系数进行逆量化来获得正交变换系数,并且进一步进行逆正交变换以重建预测残

差。在本典型实施例中，基于由在步骤S403中重建的预测信息确定的要解码的子块的大小来确定在逆量化处理中要使用的量化矩阵。更具体地，对于图7B的 32×32 阵列的各个子块，在逆量化处理中使用图8B的量化矩阵。对于图7A的 64×64 阵列的子块，使用图8C的量化矩阵。注意，要使用的量化矩阵不限于此，只要在第一典型实施例的步骤S306和S307中使用相同的量化矩阵即可。

[0139] 在步骤S405中，图像重建单元205根据在步骤S403中生成的预测信息来重建预测图像。在本典型实施例中，与第一典型实施例的步骤S305一样，使用帧内预测和帧间预测这两个类型的预测方法。图像重建单元205进一步根据重建的预测图像和在步骤S404中生成的预测残差来重建图像数据。

[0140] 在步骤S406中，图像解码设备判断帧中的所有基本块是否都被解码。如果所有基本块都被解码，则处理进入步骤S407。如果并非所有基本块都被解码，则处理以下一基本块作为对象返回到步骤S403。

[0141] 在步骤S407中，环内滤波单元207对在步骤S405中重建的图像数据进行环内滤波处理以生成滤波处理后的图像。处理结束。

[0142] 通过上述结构和操作，可以对在第一典型实施例中生成的位流进行解码，其中对于该位流，即使在仅对低频的正交变换系数进行量化和编码的子块中，也使用量化矩阵来控制各频率分量的量化以提高主观图像质量。此外，可以通过将诸如图8C的量化矩阵等的通过仅扩大基量化矩阵的低频部分所获得的量化矩阵用于仅对低频的正交变换系数进行量化和编码的子块，来对将最佳量化控制被应用于其低频部分的位流进行解码。

[0143] 在本典型实施例中，为了减少代码量，仅对要共同地用于生成图8B和图8C的量化矩阵的图8A的基量化矩阵进行解码。然而，图8B和图8C的量化矩阵自身可被解码。由于可以针对量化矩阵的各个频率分量设置独特值，因此这使得能够对针对各频率分量实现更精细的量化控制的位流进行解码。

[0144] 可以针对图8B和图8C设置各自不同的基量化矩阵，并且可以对各个基量化矩阵进行编码。在这种情况下，可以对 32×32 阵列的正交变换系数和 64×64 阵列的正交变换系数进行各自不同的量化控制，以对实现主观图像质量的更周密的控制的位流进行解码。此外，在这种情况下，可以通过代替将 8×8 阵列的基量化矩阵的左上方 4×4 阵列部分扩大八倍、而是将 8×8 阵列的整个基量化矩阵扩大四倍，来获得与 64×64 阵列的正交变换系数相对应的量化矩阵。这使得能够还对 64×64 阵列的正交变换系数进行更精细的量化控制。

[0145] 此外，尽管本典型实施例被配置成使得针对要归零的 64×64 阵列的子块的量化矩阵是唯一地确定的，但可以引入标识符以使得能够进行选择。例如，图6B示出如下的位流：新引入量化矩阵编码方法信息代码，以使要归零的 64×64 阵列的子块的量化矩阵编码是可选择的。例如，如果量化矩阵编码方法信息代码指示0，则将作为独立量化矩阵的图8C用于与要归零的 64×64 阵列的子块相对应的正交变换系数。如果编码方法信息代码指示1，则将作为针对不要归零的普通子块的量化矩阵的图8B用于要归零的 64×64 阵列的子块。如果编码方法信息代码指示2，则代替 8×8 阵列的基量化矩阵的元素，对针对要归零的 64×64 阵列的子块所要使用的量化矩阵的所有元素进行编码。这样使得能够对如下的位流进行解码，其中对于该位流，可以以可选择的方式实现量化矩阵代码的量减少和对要归零的子块的独特量化控制。

[0146] 在本典型实施例中,通过归零所处理的子块仅是 64×64 阵列的子块。然而,通过归零所处理的子块不限于此。例如,在与图7C或图7D所示的 32×64 或 64×32 阵列的子块相对应的正交变换系数中,可以在无需对下半部分或右半部分的 32×32 阵列的正交变换系数进行解码的情况下,仅对上半部分或左半部分的量化系数进行解码。在这种情况下,仅对上半部分或左半部分的 32×32 阵列的正交变换系数进行解码和逆量化。通过使用与图8B的量化矩阵不同的量化矩阵来进行对上半部分或左半部分的 32×32 阵列的正交变换系数的量化处理。

[0147] 此外,可以以与 8×8 阵列的基矩阵的元素的值分开的方式解码和设置与所生成的正交变换系数中的被认为对图像质量的影响最大的左上方的DC系数相对应的量化矩阵的值。图12B和图12C示出与DC分量相对应的最左上方元素的值从图8B和图8C中的值改变的示例。在这种情况下,可以通过除对与图8A的基量化矩阵有关的信息进行解码以外、还对DC部分的位置处的指示“2”的信息进行解码,来设置图12B和图12C所示的量化矩阵。这使得能够对如下的位流进行解码,其中对于该位流,针对对图像质量的影响最大的正交变换系数的DC分量实现了更精细的量化控制。

[0148] <第三实施例>

[0149] 在前述典型实施例中,图1和图2所示的处理单元已被描述为由硬件构成。然而,由这些图中示出的各个处理单元所进行的处理可以由计算机程序构成。

[0150] 图5是示出可应用于根据前述典型实施例的图像编码设备和图像解码设备的计算机的硬件结构示例的框图。

[0151] 中央处理单元(CPU)501通过使用随机存取存储器(RAM)502和只读存储器(ROM)503中所存储的计算机程序和数据来控制整个计算机,并且执行要由根据典型实施例的设备进行的上述处理。换句话说,CPU 501用作图1和图2所示的处理单元。

[0152] RAM 502包括用于临时存储从外部存储装置506加载的计算机程序和数据、以及经由接口(I/F)507从外部获得的数据等的区域。RAM 502还包括供CPU 501在执行各种处理时使用的工作区。换句话说,RAM 502例如可被分配作为帧存储器,并适当地提供其它各种区域。

[0153] ROM 503存储计算机的设置数据和引导程序等。操作单元504包括键盘和鼠标。计算机的用户可以通过对操作单元504进行操作来将各种指令输入到CPU 501中。输出单元505输出利用CPU 501的处理的结果。输出单元505例如包括液晶显示器。

[0154] 外部存储装置506是以硬盘驱动装置为代表的大容量信息存储装置。外部存储设备506存储操作系统(OS)、以及用于使CPU 501实现图1和图2所示的单元的功能的计算机程序。外部存储装置506还可以存储要处理的图像数据。

[0155] 外部存储装置506中所存储的计算机程序和数据基于CPU 501的控制被适当地加载到RAM 502中,并且经过CPU 501的处理。诸如局域网(LAN)和因特网等的网络以及诸如投影装置和显示装置等的其它装置可以连接到I/F507。计算机可以经由I/F 507获得和发送各种类型的信息。总线508连接上述单元。

[0156] 前述结构的活动或前述流程图中描述的活动主要由CPU 501控制。

[0157] (其它典型实施例)

[0158] 典型实施例还可以通过以下来实现:将记录有用于实现前述功能的计算机程序代

码的存储介质供给至系统，并且由系统来读取并执行该计算机程序代码。在这种情况下，从存储介质读取的计算机程序代码自身实现前述典型实施例的功能，并且存储计算机程序代码的存储介质构成本发明。还包括如下的情况：在计算机上运行的操作系统（OS）等基于程序代码中的指令来进行实际处理的一部分或全部，并且通过该处理来实现前述功能。

[0159] 典型实施例还可以通过以下方式来实现。也就是说，将从存储介质读取的计算机程序代码写入插入到计算机中的功能扩展卡或连接到计算机的功能扩展单元中所包括的存储器中。然后，该功能扩展卡或功能扩展单元中所包括的CPU等基于计算机程序代码中的指令来进行实际处理的一部分或全部，并由此实现上述功能。这样的情况也包括在内。

[0160] 如果将本发明应用于上述存储介质，则存储介质存储与上述流程图相对应的计算机程序代码。

[0161] 根据前述典型实施例，尽管使用将正交变换系数的一部分强制设置为0的技术，通过使得能够进行使用与该技术相对应的量化矩阵的量化处理，也可以提高主观图像质量。

[0162] 本发明不限于上述实施例，并且可以在本发明的精神和范围内进行各种改变和修改。因此，为了向公众告知本发明的范围，添加了所附权利要求。

[0163] 本申请要求2019年3月11日提交的日本专利申请2019-044275的优先权，其通过引用而被全部包含于此。

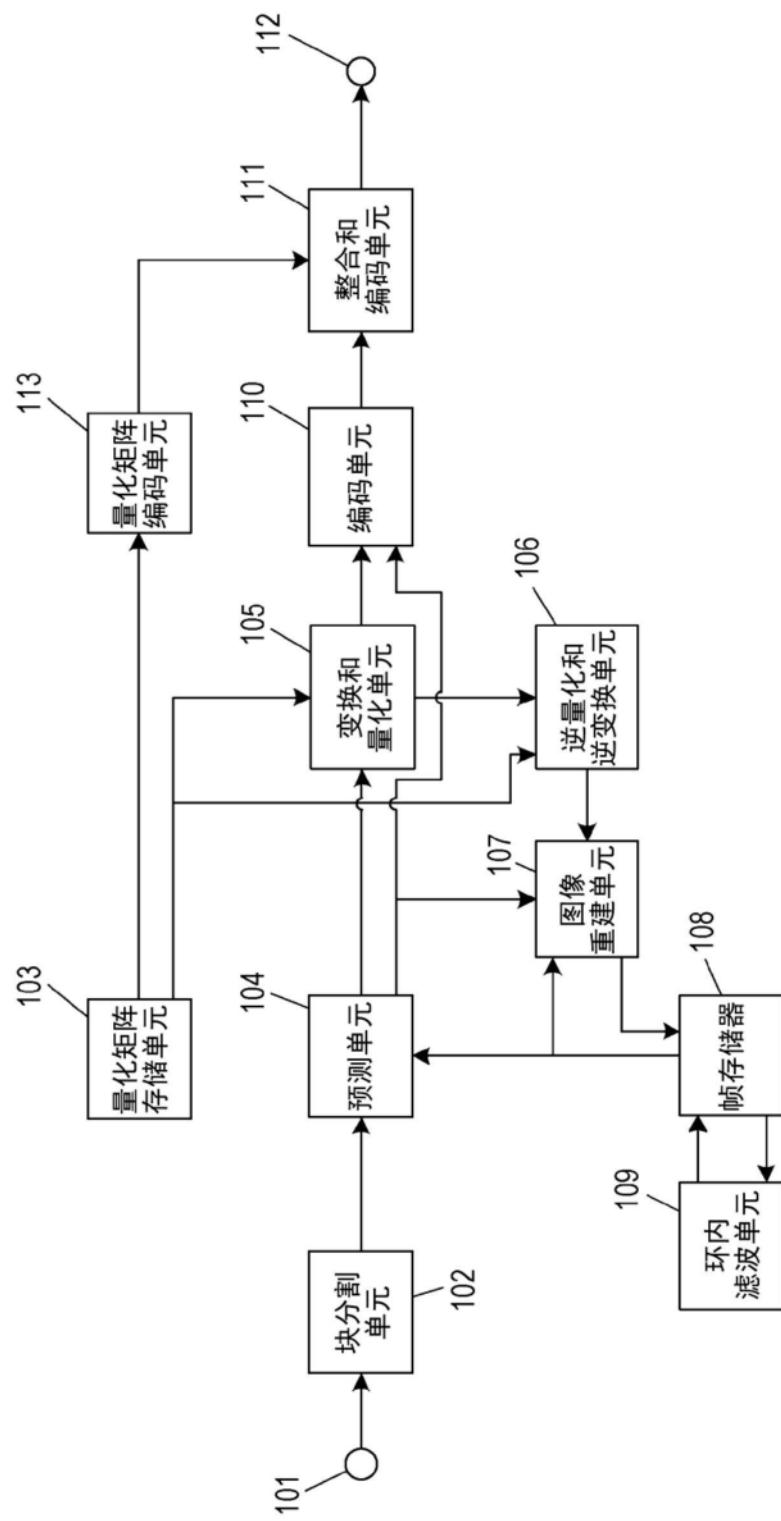


图1

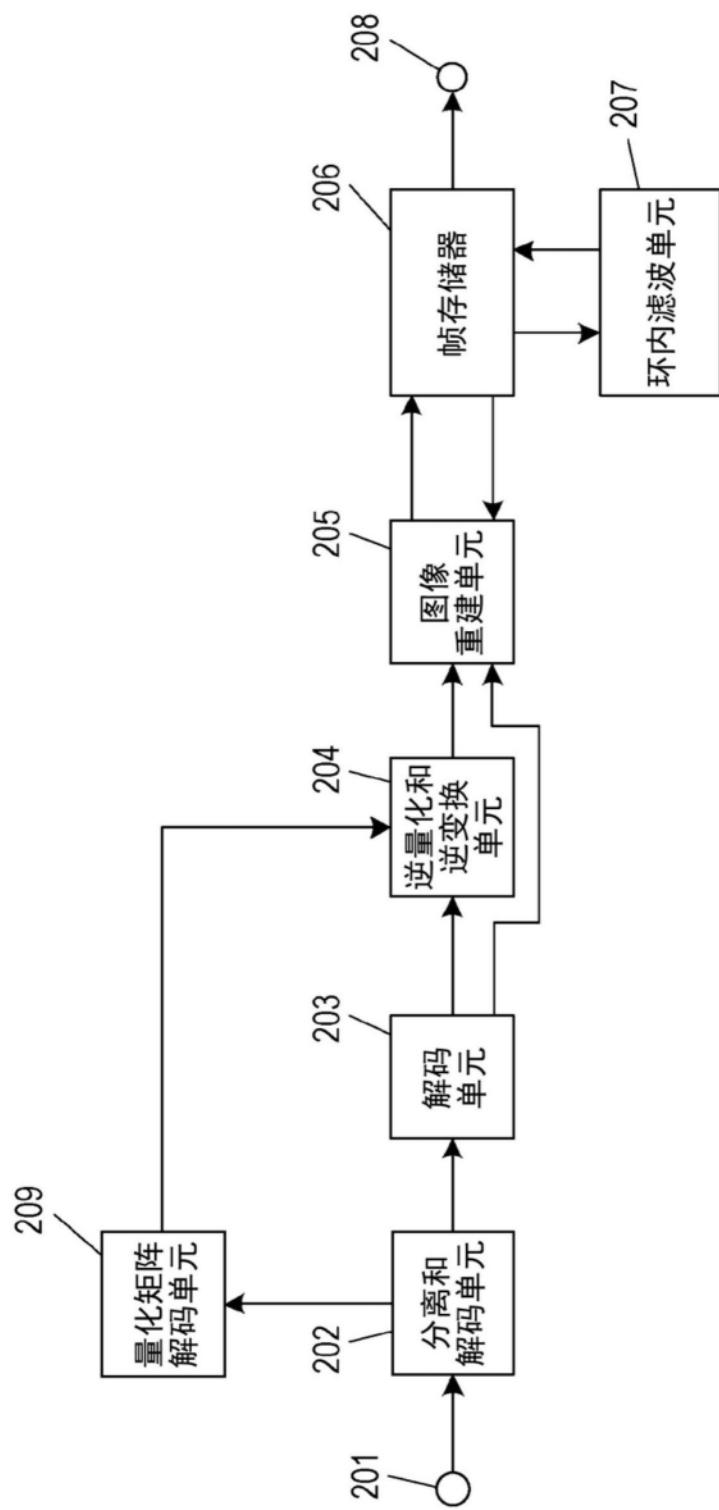


图2

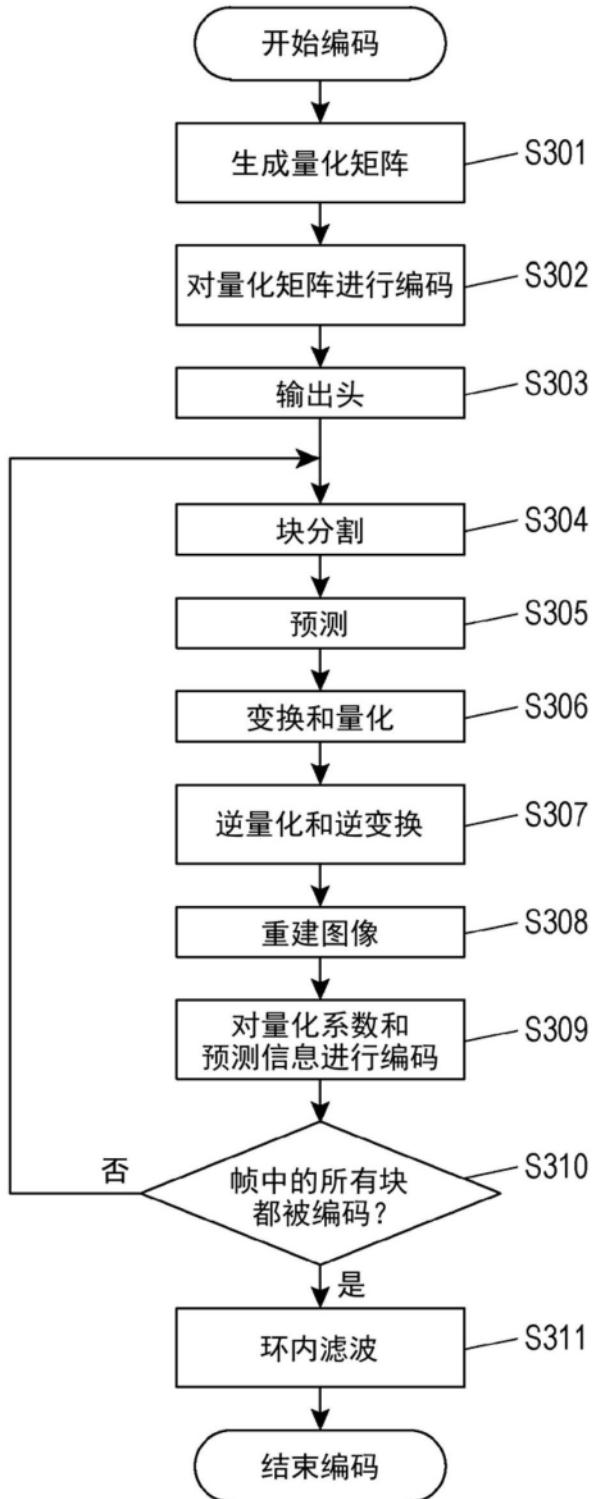


图3

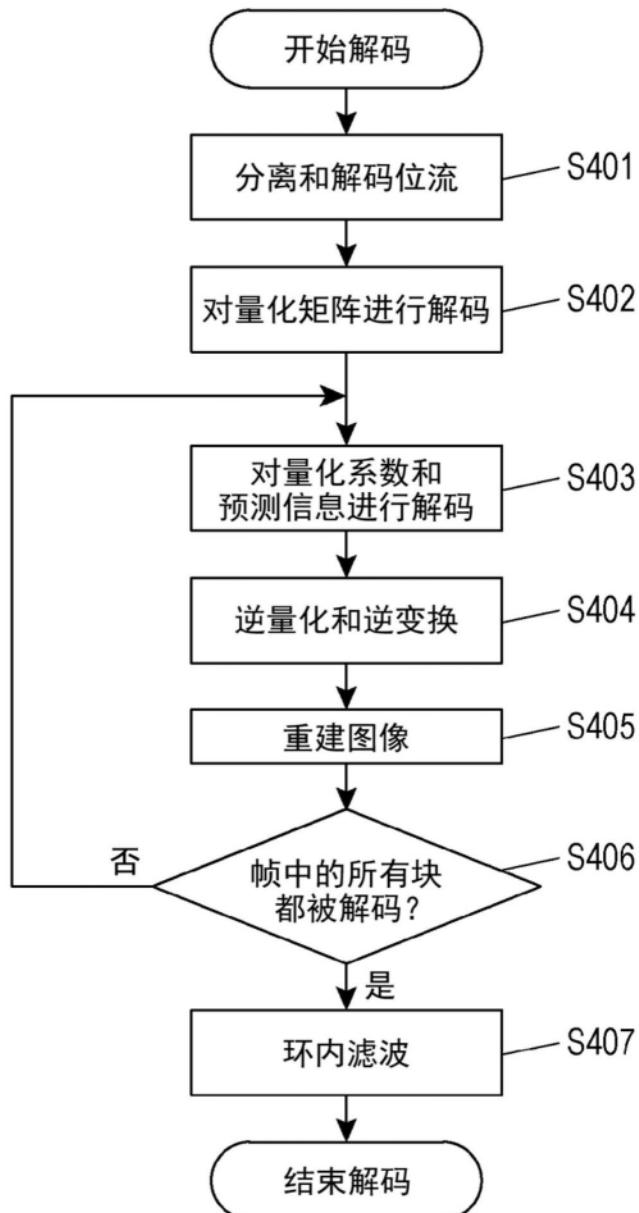


图4

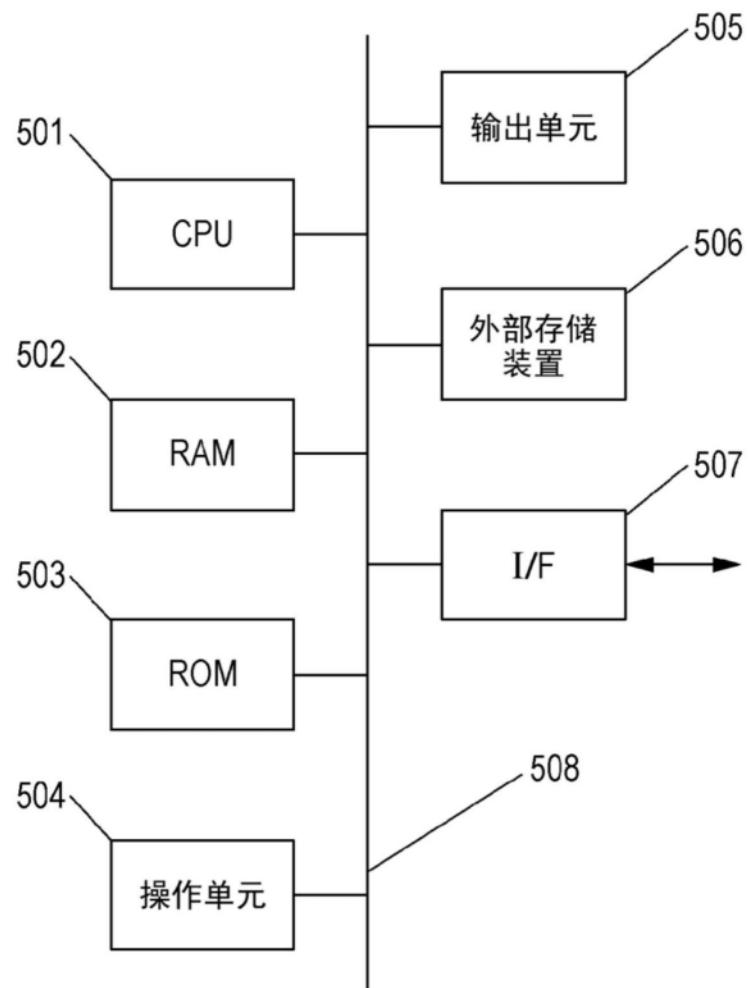


图5

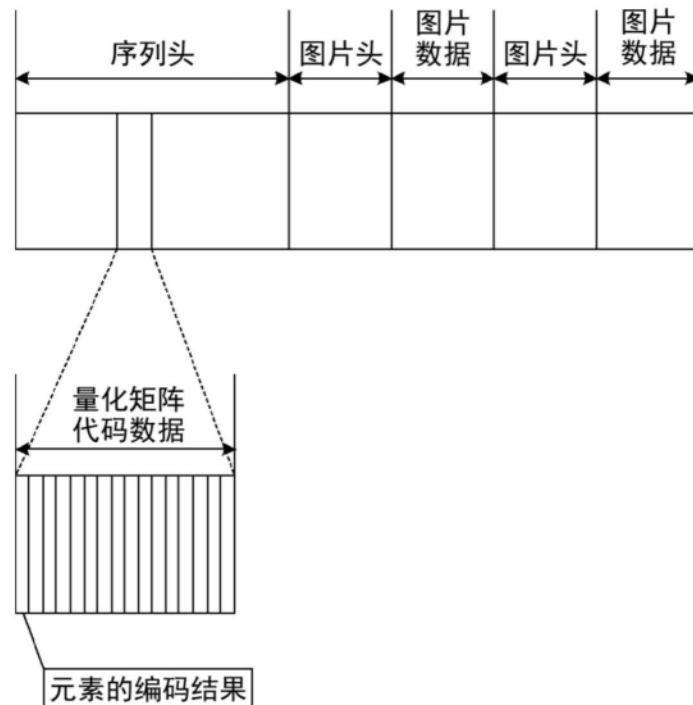


图6A

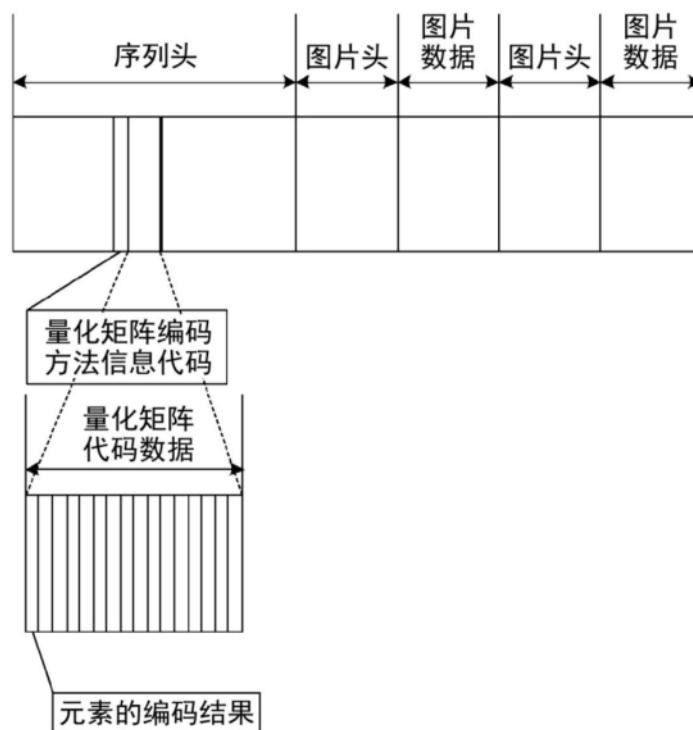


图6B

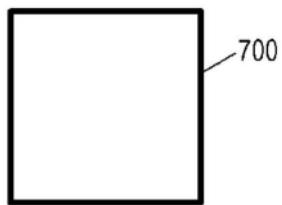


图7A

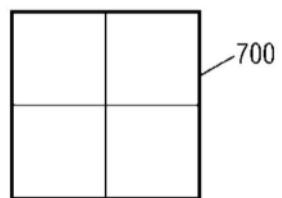


图7B

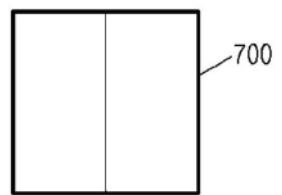


图7C

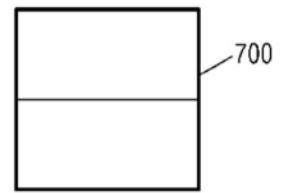


图7D

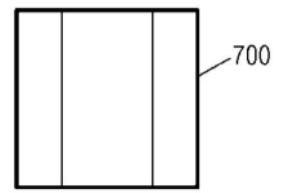


图7E

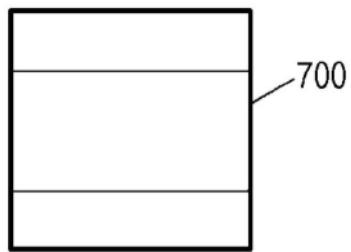


图7F

1	2	3	4	5	6	7	8
2	3	4	5	6	7	8	9
3	4	5	6	7	8	9	10
4	5	6	7	8	9	10	11
5	6	7	8	9	10	11	12
6	7	8	9	10	11	12	13
7	8	9	10	11	12	13	14
8	9	10	11	12	13	14	15

图8A

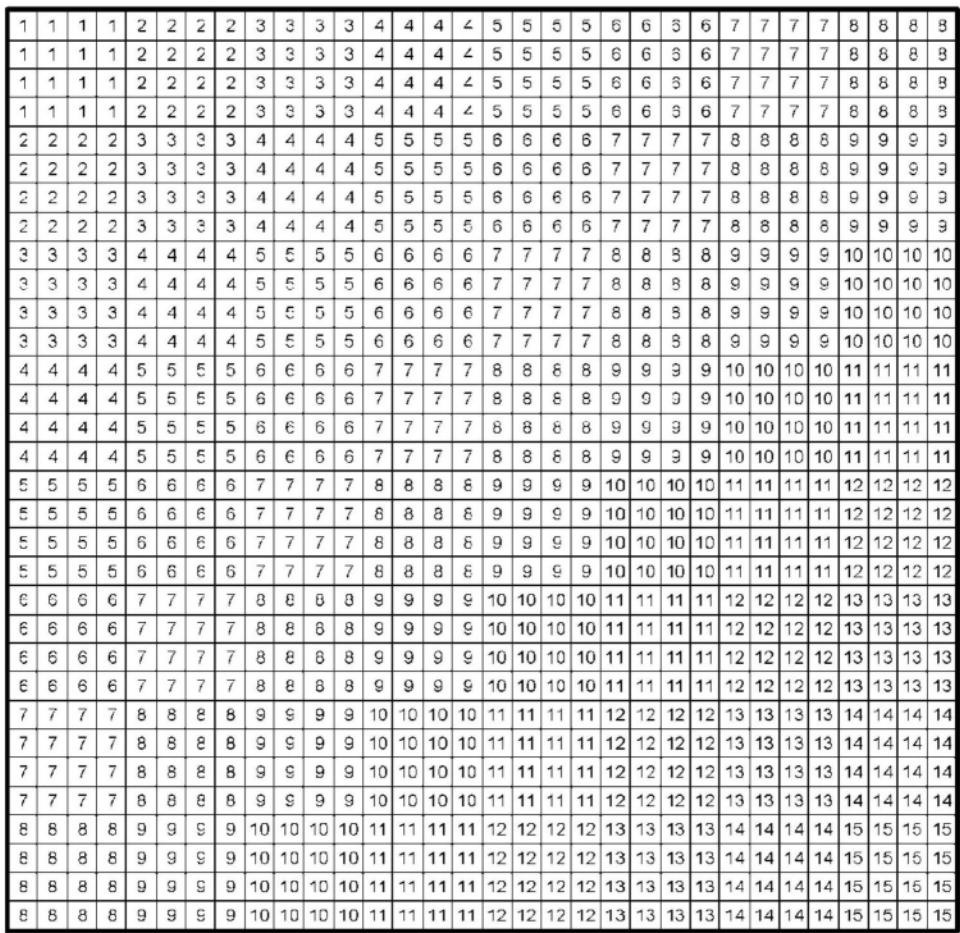


Diagram 8B shows a 15x15 grid of numbers. The numbers are arranged in a pattern where each row and column contains all integers from 1 to 15 exactly once. The grid is bounded by a black border. A callout line with the number '800' points to the cell at the top-right corner of the grid.

1	1	1	1	1	2	2	2	2	3	3	3	3	4	4	4	4	5	5	5	5	6
1	1	1	1	2	2	2	2	2	3	3	3	3	4	4	4	4	5	5	5	5	6
1	1	1	1	2	2	2	2	2	3	3	3	3	4	4	4	4	5	5	5	5	6
1	1	1	1	2	2	2	2	2	3	3	3	3	4	4	4	4	5	5	5	5	6
2	2	2	2	3	3	3	3	3	4	4	4	4	4	4	4	4	5	5	5	5	6
2	2	2	2	3	3	3	3	3	4	4	4	4	4	4	4	4	5	5	5	5	6
2	2	2	2	3	3	3	3	3	4	4	4	4	4	4	4	4	5	5	5	5	6
2	2	2	2	3	3	3	3	3	4	4	4	4	4	4	4	4	5	5	5	5	6
3	3	3	3	4	4	4	4	4	5	5	5	5	6	6	6	6	7	7	7	7	8
3	3	3	3	4	4	4	4	4	5	5	5	5	6	6	6	6	7	7	7	7	8
3	3	3	3	4	4	4	4	4	5	5	5	5	6	6	6	6	7	7	7	7	8
3	3	3	3	4	4	4	4	4	5	5	5	5	6	6	6	6	7	7	7	7	8
4	4	4	4	5	5	5	5	5	6	6	6	6	7	7	7	7	8	8	8	8	9
4	4	4	4	5	5	5	5	5	6	6	6	6	7	7	7	7	8	8	8	9	9
4	4	4	4	5	5	5	5	5	6	6	6	6	7	7	7	7	8	8	8	9	9
5	5	5	5	6	6	6	6	7	7	7	7	7	8	8	8	8	9	9	9	9	10
5	5	5	5	6	6	6	6	7	7	7	7	7	8	8	8	8	9	9	9	9	10
5	5	5	5	6	6	6	6	7	7	7	7	7	8	8	8	8	9	9	9	9	10
6	6	6	6	7	7	7	7	8	8	8	8	9	9	9	9	9	10	10	10	10	11
6	6	6	6	7	7	7	7	7	8	8	8	8	9	9	9	9	10	10	10	10	11
6	6	6	6	7	7	7	7	7	8	8	8	8	9	9	9	9	10	10	10	10	11
7	7	7	7	8	8	8	8	9	9	9	9	10	10	10	10	11	11	11	11	12	12
7	7	7	7	8	8	8	8	9	9	9	9	10	10	10	10	11	11	11	12	12	12
7	7	7	7	8	8	8	8	9	9	9	9	10	10	10	10	11	11	11	12	12	12
8	8	8	8	9	9	9	9	10	10	10	10	11	11	11	11	12	12	12	12	13	13
8	8	8	8	9	9	9	9	10	10	10	10	11	11	11	11	12	12	12	12	13	13
8	8	8	8	9	9	9	9	10	10	10	10	11	11	11	11	12	12	12	12	13	13
8	8	8	8	9	9	9	9	10	10	10	10	11	11	11	11	12	12	12	12	13	13

图8B

Diagram 8C shows a 10x10 grid of numbers. The numbers are arranged in a pattern where they increase from left to right and top to bottom. An arrow labeled '800' points to the number '4' located at the bottom-right corner of the grid.

1	1	1	1	1	1	1	1	2	2
1	1	1	1	1	1	1	1	2	2
1	1	1	1	1	1	1	1	2	2
1	1	1	1	1	1	1	1	2	2
1	1	1	1	1	1	1	1	2	2
2	2	2	2	2	2	2	2	3	3
2	2	2	2	2	2	2	2	3	3
2	2	2	2	2	2	2	2	3	3
2	2	2	2	2	2	2	2	3	3
2	2	2	2	2	2	2	2	3	3
3	3	3	3	3	3	3	3	4	4
3	3	3	3	3	3	3	3	4	4
3	3	3	3	3	3	3	3	4	4
3	3	3	3	3	3	3	3	4	4
3	3	3	3	3	3	3	3	4	4
4	4	4	4	4	4	4	4	5	5
4	4	4	4	4	4	4	4	5	5
4	4	4	4	4	4	4	4	5	5
4	4	4	4	4	4	4	4	5	5
4	4	4	4	4	4	4	4	5	5

图8C

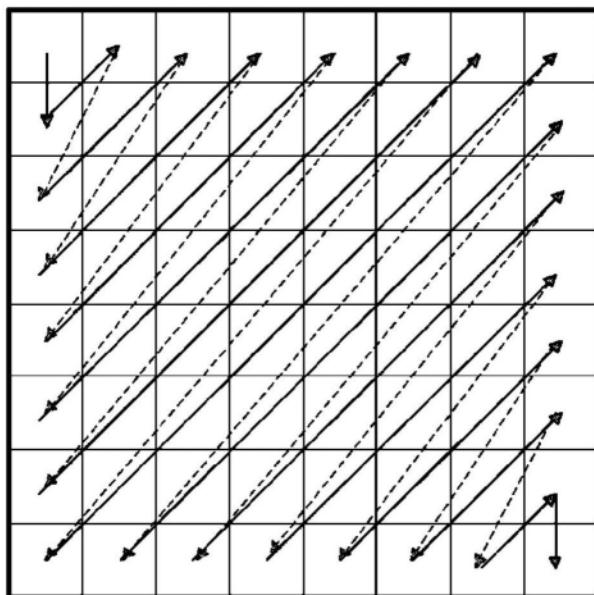


图9

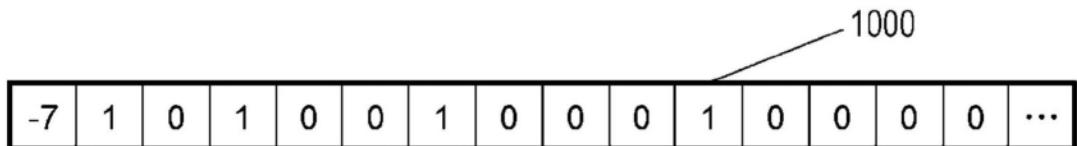


图10

要编码的值	二进制代码
...	...
-5	0001011
-4	0001001
-3	00111
-2	00101
-1	011
0	1
1	010
2	00100
3	00110
4	0001000
5	0001010
...	...

图11A

要编码的值	二进制代码
...	...
-5	0001011
-4	0001001
-3	00111
-2	0010
-1	011
0	11
1	10
2	010
3	00110
4	0001000
5	0001010
...	...

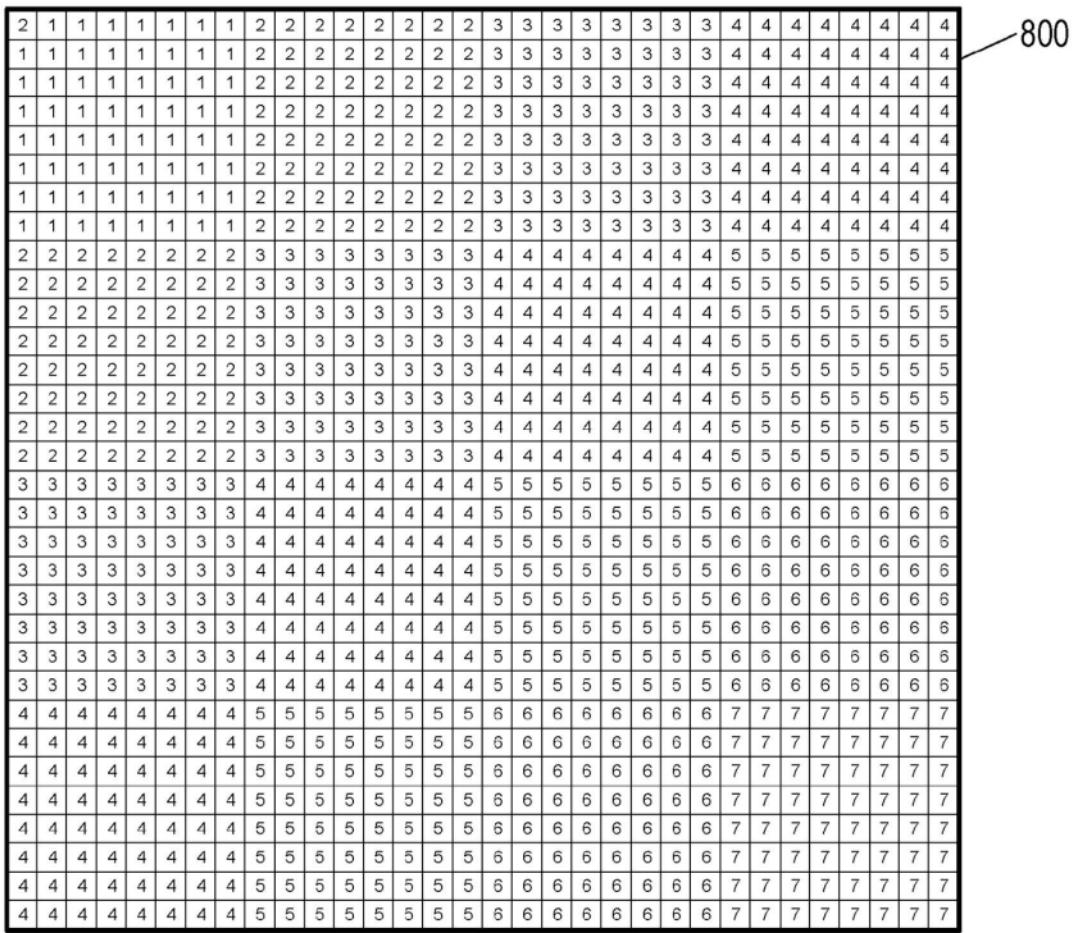
图11B

1	2	3	4	5	6	7	8
2	3	4	5	6	7	8	9
3	4	5	6	7	8	9	10
4	5	6	7	8	9	10	11
5	6	7	8	9	10	11	12
6	7	8	9	10	11	12	13
7	8	9	10	11	12	13	14
8	9	10	11	12	13	14	15

图12A

2	1	1	1	1	2	2	2	2	3	3	3	3	4	4	4	4	5	5	5	5	6	6	6	6	7	7	7	7	7	8	8	8	8	8		
1	1	1	1	1	2	2	2	2	3	3	3	3	4	4	4	4	5	5	5	5	6	6	6	6	7	7	7	7	7	8	8	8	8	8		
1	1	1	1	1	2	2	2	2	3	3	3	3	4	4	4	4	5	5	5	5	6	6	6	6	7	7	7	7	7	8	8	8	8	8		
2	2	2	2	2	3	3	3	3	4	4	4	4	5	5	5	5	6	6	6	6	7	7	7	7	7	8	8	8	9	9	9	9	9	9		
2	2	2	2	2	3	3	3	3	4	4	4	4	5	5	5	5	6	6	6	6	7	7	7	7	7	8	8	8	9	9	9	9	9	9		
2	2	2	2	2	3	3	3	3	4	4	4	4	5	5	5	5	6	6	6	6	7	7	7	7	7	8	8	8	9	9	9	9	9	9		
2	2	2	2	2	3	3	3	3	4	4	4	4	5	5	5	5	6	6	6	6	7	7	7	7	7	8	8	8	9	9	9	9	9	9		
3	3	3	3	4	4	4	4	5	5	5	5	6	6	6	6	7	7	7	7	7	8	8	8	8	9	9	9	9	9	10	10	10	10	10		
3	3	3	3	4	4	4	4	5	5	5	5	6	6	6	6	7	7	7	7	7	8	8	8	8	9	9	9	9	10	10	10	10	10			
3	3	3	3	4	4	4	4	5	5	5	5	6	6	6	6	7	7	7	7	7	8	8	8	8	9	9	9	9	10	10	10	10	10			
3	3	3	3	4	4	4	4	5	5	5	5	6	6	6	6	7	7	7	7	7	8	8	8	8	9	9	9	9	10	10	10	10	10			
4	4	4	4	5	5	5	5	6	6	6	6	7	7	7	7	8	8	8	8	9	9	9	9	9	10	10	10	10	10	11	11	11	11	11		
4	4	4	4	5	5	5	5	6	6	6	6	7	7	7	7	8	8	8	8	9	9	9	9	9	10	10	10	10	10	11	11	11	11	11		
4	4	4	4	5	5	5	5	6	6	6	6	7	7	7	7	8	8	8	8	9	9	9	9	9	10	10	10	10	10	11	11	11	11	11		
4	4	4	4	5	5	5	5	6	6	6	6	7	7	7	7	8	8	8	8	9	9	9	9	9	10	10	10	10	10	11	11	11	11	11		
5	5	5	5	6	6	6	6	7	7	7	7	8	8	8	8	9	9	9	9	10	10	10	10	10	11	11	11	11	12	12	12	12	12	12		
5	5	5	5	6	6	6	6	7	7	7	7	8	8	8	8	9	9	9	9	10	10	10	10	10	11	11	11	11	12	12	12	12	12	12		
5	5	5	5	6	6	6	6	7	7	7	7	8	8	8	8	9	9	9	9	10	10	10	10	10	11	11	11	11	12	12	12	12	12	12		
5	5	5	5	6	6	6	6	7	7	7	7	8	8	8	8	9	9	9	9	10	10	10	10	10	11	11	11	11	12	12	12	12	12	12		
6	6	6	6	7	7	7	7	8	8	8	8	9	9	9	9	10	10	10	10	10	11	11	11	11	11	12	12	12	12	12	13	13	13	13	13	13
6	6	6	6	7	7	7	7	8	8	8	8	9	9	9	9	10	10	10	10	10	11	11	11	11	11	12	12	12	12	12	13	13	13	13	13	13
6	6	6	6	7	7	7	7	8	8	8	8	9	9	9	9	10	10	10	10	10	11	11	11	11	11	12	12	12	12	12	13	13	13	13	13	13
7	7	7	7	8	8	8	8	9	9	9	9	10	10	10	10	11	11	11	11	12	12	12	12	12	13	13	13	13	13	14	14	14	14	14	14	
7	7	7	7	8	8	8	8	9	9	9	9	10	10	10	10	11	11	11	11	12	12	12	12	12	13	13	13	13	13	14	14	14	14	14	14	
7	7	7	7	8	8	8	8	9	9	9	9	10	10	10	10	11	11	11	11	12	12	12	12	12	13	13	13	13	13	14	14	14	14	14	14	
8	8	8	8	9	9	9	9	10	10	10	10	11	11	11	11	12	12	12	12	12	13	13	13	13	13	14	14	14	14	14	15	15	15	15	15	15
8	8	8	8	9	9	9	9	10	10	10	10	11	11	11	11	12	12	12	12	12	13	13	13	13	13	14	14	14	14	14	15	15	15	15	15	15

图12B



The diagram consists of a 16x16 grid of numbers. The numbers are arranged in a pattern where each row and column contains a sequence of 1s, 2s, 3s, and 4s. The sequence starts with 1s, followed by 2s, then 3s, and finally 4s. This pattern repeats across all 16 rows and 16 columns. A callout line with the number '800' points to the top-right corner of the grid.

2	1	1	1	1	1	1	1	1	2	2	2	2	2	2	2	2
1	1	1	1	1	1	1	1	2	2	2	2	2	2	2	2	2
1	1	1	1	1	1	1	1	2	2	2	2	2	2	2	2	3
1	1	1	1	1	1	1	1	2	2	2	2	2	2	2	2	3
1	1	1	1	1	1	1	1	2	2	2	2	2	2	2	2	3
1	1	1	1	1	1	1	1	2	2	2	2	2	2	2	2	3
1	1	1	1	1	1	1	1	2	2	2	2	2	2	2	2	3
1	1	1	1	1	1	1	1	2	2	2	2	2	2	2	2	3
2	2	2	2	2	2	2	2	2	3	3	3	3	3	3	3	4
2	2	2	2	2	2	2	2	2	3	3	3	3	3	3	3	4
2	2	2	2	2	2	2	2	2	3	3	3	3	3	3	3	4
2	2	2	2	2	2	2	2	2	3	3	3	3	3	3	3	4
2	2	2	2	2	2	2	2	2	3	3	3	3	3	3	3	4
2	2	2	2	2	2	2	2	2	3	3	3	3	3	3	3	4
2	2	2	2	2	2	2	2	2	3	3	3	3	3	3	3	4
3	3	3	3	3	3	3	3	4	4	4	4	4	4	4	4	5
3	3	3	3	3	3	3	3	4	4	4	4	4	4	4	4	5
3	3	3	3	3	3	3	3	4	4	4	4	4	4	4	4	5
3	3	3	3	3	3	3	3	4	4	4	4	4	4	4	4	5
3	3	3	3	3	3	3	3	4	4	4	4	4	4	4	4	5
3	3	3	3	3	3	3	3	4	4	4	4	4	4	4	4	5
4	4	4	4	4	4	4	4	5	5	5	5	5	5	5	5	6
4	4	4	4	4	4	4	4	5	5	5	5	5	5	5	5	6
4	4	4	4	4	4	4	4	5	5	5	5	5	5	5	5	6
4	4	4	4	4	4	4	4	5	5	5	5	5	5	5	5	6
4	4	4	4	4	4	4	4	5	5	5	5	5	5	5	5	6
4	4	4	4	4	4	4	4	5	5	5	5	5	5	5	5	6
4	4	4	4	4	4	4	4	5	5	5	5	5	5	5	5	6
4	4	4	4	4	4	4	4	5	5	5	5	5	5	5	5	6

图12C