



(12) 发明专利

(10) 授权公告号 CN 101203882 B

(45) 授权公告日 2011. 07. 13

(21) 申请号 200680012737. 9

(22) 申请日 2006. 05. 19

(30) 优先权数据

60/683, 240 2005. 05. 20 US

11/437, 515 2006. 05. 18 US

(85) PCT申请进入国家阶段日

2007. 10. 17

(86) PCT申请的申请数据

PCT/US2006/019641 2006. 05. 19

(87) PCT申请的公布数据

W02006/127546 EN 2006. 11. 30

(73) 专利权人 株式会社 NTT 都科摩

地址 美国加利福尼亚

(72) 发明人 丹·莱列斯库 弗兰克·博森

(74) 专利代理机构 中科专利商标代理有限责任
公司 11021

代理人 王波波

(51) Int. Cl.

G06T 5/00 (2006. 01)

H04N 7/30 (2006. 01)

(56) 对比文件

CN 1236461 A, 1999. 11. 24, 权利要求 1、说明书第 4 页末段至第 5 页首段。

US 2005/0078872 A1, 2005. 04. 14, 全文。

R. A. Gopinath, M. Lang H. Guo, J. E. Odegard. WAVELET-BASED POST-PROCESSING OF LOW BITRATETTRANSFORM CODED IMAGES. PROCEEDINGS OF THE INTERNATIONAL CONFERENCE ON IMAGE PROCESSING. 1994, 913-914.

Xin Li, Michael T. Orchard. SPATIALLY ADAPTIVE IMAGEDENOISINGUNDEROVERCOMPLETE EXPANSION. CONFERENCE RECORD OF THE 37TH. ASILOMAR CONFERENCE ON SIGNALS, SYSTEMS & COMPUTERS. . 2000, 301.

审查员 慈丽雁

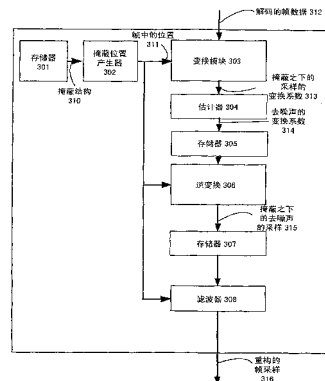
权利要求书 4 页 说明书 17 页 附图 11 页

(54) 发明名称

视频编码中的噪声滤波方法和设备

(57) 摘要

这里公开一种方法和设备,用于对视频帧数据进行编码和/或解码。在一个实施例中,视频编码器包括噪声滤波模块,对变换后的帧数据进行操作,并通过估计多个变换系数的每一个的信号功率,将所述每个系数的信号功率与至少一个阈值相比较,并至少部分基于信号功率与至少一个阈值的比较结果来设置所述每个变换系数的值,来对多个变换系数进行信号估计。



1. 一种用于对视频帧中的数据进行编码的视频编码器,所述视频编码器包括:
噪声滤波模块,用于对变换后的帧数据进行操作,并通过下面的步骤来对多个变换系数进行信号估计:

估计多个变换系数的每一个的信号功率,

将所述每个变换系数的信号功率与至少一个阈值相比较,以及

至少根据信号功率与至少一个阈值的比较结果来设置所述每个变换系数的值,

其中如果所述至少一个阈值是单个阈值,则通过以下步骤计算每个变换系数:

当信号功率小于单个阈值时,将所述变换系数的值设置为零,并且

当信号功率不小于单个阈值时,基于信号功率、噪声功率和单个阈值来调整所述变换系数的值;以及

其中如果所述至少一个阈值是第一阈值和第二阈值,则通过以下步骤计算每个变换系数:

当信号功率小于第一阈值时,将所述变换系数的值设置为零,

当信号功率不小于第一阈值但小于第二阈值时,基于信号功率、噪声功率和所述第一阈值和第二阈值来调整所述变换系数的值,并且

当信号功率不小于第一阈值也不小于第二阈值时,保持所述变换系数的值不变。

2. 一种用于对视频编码器中的多个变换系数执行信号估计的方法,包括以下步骤:

估计多个变换系数的每一个的信号功率,

将所述每个变换系数的信号功率与至少一个阈值相比较,以及

至少根据信号功率与至少一个阈值的比较结果来设置所述每个变换系数的值,

其中如果所述至少一个阈值是单个阈值,则通过以下步骤计算每个变换系数:

当信号功率小于单个阈值时,将所述变换系数的值设置为零,并且

当信号功率不小于单个阈值时,基于信号功率、噪声功率和单个阈值来调整所述变换系数的值;以及

其中如果所述至少一个阈值是第一阈值和第二阈值,则通过以下步骤计算每个变换系数:

当信号功率小于第一阈值时,将所述变换系数的值设置为零,

当信号功率不小于第一阈值但小于第二阈值时,基于信号功率、噪声功率和所述第一阈值和第二阈值来调整所述变换系数的值,并且

当信号功率不小于第一阈值也不小于第二阈值时,保持所述变换系数的值不变。

3. 一种对视频帧数据去噪声的方法,包括以下步骤:

从存储器中取得帧数据;

对在掩蔽之下的帧中的采样应用变换,以创建变换系数;

对变换系数去噪声;

对去噪声后的变换系数应用逆变换;以及

对与帧中位置相对应的去噪声后的帧采样进行滤波,以获得该位置处的滤波后的数据采样,

其中对变换系数去噪声包括:

至少根据变换系数的信号功率与至少一个阈值的比较结果来设置所述每个变换系数

的值,

其中如果所述至少一个阈值是单个阈值,则通过以下步骤计算每个变换系数:

当信号功率小于单个阈值时,将所述变换系数的值设置为零,并且

当信号功率不小于单个阈值时,基于信号功率、噪声

功率和单个阈值来调整所述变换系数的值;以及

其中如果所述至少一个阈值是第一阈值和第二阈值,则通过以下步骤计算每个变换系数:

当信号功率小于第一阈值时,将所述变换系数的值设置为零,

当信号功率不小于第一阈值但小于第二阈值时,基于信号功率、噪声功率和所述第一阈值和第二阈值来调整所述变换系数的值,并且

当信号功率不小于第一阈值也不小于第二阈值时,保持所述变换系数的值不变。

4. 一种噪声滤波模块,包括:

变换模块,用于将帧空间数据变换到适用于信号处理的另一个域;

估计器模块,用于通过估计信号功率、将信号功率与至少一个阈值相比较,以及根据比较结果调整变换系数,来产生与帧数据相对应的变换系数的去噪声后的估计值,

其中如果所述至少一个阈值是单个阈值,则通过以下步骤计算每个变换系数:

当信号功率小于单个阈值时,将所述变换系数的值设置为零,并且

当信号功率不小于单个阈值时,基于信号功率、噪声功率和单个阈值来调整所述变换系数的值;以及

如果所述至少一个阈值是第一阈值和第二阈值,则通过以下步骤计算每个变换系数:

当信号功率小于第一阈值时,将所述变换系数的值设置为零,

当信号功率不小于第一阈值但小于第二阈值时,基于信号功率、噪声功率和所述第一阈值和第二阈值来调整所述变换系数的值,并且

当信号功率不小于第一阈值也不小于第二阈值时,保持所述变换系数的值不变;

逆变换模块,用于向去噪声后的变换系数应用逆变换,并使掩蔽下的数据回到其原始域;以及

滤波模块,用于通过将估计器模块的处理所产生的估计值组合,来获得每个帧位置处的滤波后的采样。

5. 一种估计器模块,包括:

估计多个变换系数中的每一个的信号功率的装置;

将信号功率与至少一个阈值相比较的装置;以及

根据信号功率与至少一个阈值的比较结果来调整变换系数值的装置,

其中如果所述至少一个阈值是单个阈值,则通过以下步骤计算每个变换系数:

当信号功率小于单个阈值时,将所述变换系数的值设置为零,并且

当信号功率不小于单个阈值时,基于信号功率、噪声功率和单个阈值来调整所述变换系数的值;以及

其中如果所述至少一个阈值是第一阈值和第二阈值,则通过以下步骤计算每个变换系数:

当信号功率小于第一阈值时,将所述变换系数的值设置为零,

当信号功率不小于第一阈值但小于第二阈值时,基于信号功率、噪声功率和所述第一阈值和第二阈值来调整所述变换系数的值,并且

当信号功率不小于第一阈值也不小于第二阈值时,保持所述变换系数的值不变。

6. 一种用于对视频帧中的数据进行解码的视频解码器,所述视频解码器包括:

噪声滤波模块,用于对变换后的帧数据进行操作,并通过以下步骤对多个变换系数执行信号估计:

估计所述多个变换系数中的每个的信号功率,

将每个变换系数的信号功率与至少一个阈值相比较,以及

至少根据信号功率与至少一个阈值的比较结果来设置所述每个变换系数的值,

其中如果所述至少一个阈值是单个阈值,则通过以下步骤计算每个变换系数:

当信号功率小于单个阈值时,将所述变换系数的值设置为零,并且

当信号功率不小于单个阈值时,基于信号功率、噪声功率和单个阈值来调整所述变换系数的值;以及

其中如果所述至少一个阈值是第一阈值和第二阈值,则通过以下步骤计算每个变换系数:

当信号功率小于第一阈值时,将所述变换系数的值设置为零,

当信号功率不小于第一阈值但小于第二阈值时,基于信号功率、噪声功率和所述第一和第二阈值来调整所述变换系数的值,并且

当信号功率不小于第一阈值也不小于第二阈值时,保持所述变换系数的值不变。

7. 一种解码方法,包括:

对帧数据块的预测误差进行解码;

对与所述块相关联的运动矢量进行解码;

使用运动矢量和参考帧,对所述块执行运动补偿,以确定块预测值;

使用块预测值和预测误差,重构所述块;以及

通过从帧数据中产生多个变换系数、响应于变换系数而将变换系数的估计信号功率与至少一个阈值相比较以产生去噪声后的变换系数、以及对去噪声后的采样进行滤波以创建重构的帧采样,来对重构的块采样进行滤波,

其中将变换系数的估计信号功率与至少一个阈值相比较以产生去噪声后的变换系数还包括:

至少根据变换系数的信号功率与至少一个阈值的比较结果来设置所述每个变换系数的值,

其中如果所述至少一个阈值是单个阈值,则通过以下步骤计算每个变换系数:

当信号功率小于单个阈值时,将所述变换系数的值设置为零,并且

当信号功率不小于单个阈值时,基于信号功率、噪声功率和单个阈值来调整所述变换系数的值;以及

其中如果所述至少一个阈值是第一阈值和第二阈值,则通过以下步骤计算每个变换系数:

当信号功率小于第一阈值时,将所述变换系数的值设置为零,

当信号功率不小于第一阈值但小于第二阈值时,基于信号功率、噪声功率和所述第一

和第二阈值来调整所述变换系数的值,并且

当信号功率不小于第一阈值也不小于第二阈值时,保持所述变换系数的值不变。

视频编码中的噪声滤波方法和设备

技术领域

[0001] 本发明涉及视频编码领域；更具体地，本发明涉及根据对系数的信号功率的估计以及这些估计值与一个或多个阈值的比较来在视频编码中执行噪声滤波。

背景技术

[0002] 混合视频编码的传统方法对视频帧本身进行编码，而不参考序列中的其它帧，或者通过根据序列中已经编码且重构的其它帧来预测视频帧，而对视频帧进行编码。该处理通常在图像的块级别上进行操作。在任何一种情况下，编码过程之后重构的帧包含该过程所引入的失真，可将所述失真建模为噪声。该噪声可作为例如块和边缘振荡伪像 (artifacts) 的编码伪像而可见。典型地由视频编码器和解码器使用的存储在帧存储器中的一些重构帧还用作预测序列中的其它帧的参考，而其它帧则不是。在前一种情况下，存储在帧存储器中的重构帧的质量不仅对于其显示是重要的，而且通过其在预测过程中的使用，影响将该帧用作参考的随后编码的帧的质量。在后一种情况下，对于并未用作参考的帧，其质量对于显示目的是重要的。因此，在任意一种情况下，去除编码噪声的过程对于重构视频序列的质量是有益的，并提高客观编码效率。

[0003] 多数现代编解码器包括某种滤波以降低所产生的编码噪声的程度。滤波可工作在编解码器的编码环路之外或之内。对于工作在编码环路之内的环路滤波器，通常在编码器和在解码器中使用相同可用数据来执行相同的操作，因而不需要明确地从编码器将该滤波特有的边信息 (side information) 发送到解码器。通常使用的噪声滤波方法包括：对重构帧数据的简单的低通滤波，以平滑图像，并降低编码伪像的可视度；以及根据本地图像特征（例如块边缘）来改变其强度的环路滤波器。

[0004] 诸如在解码器处对重构（有噪声）帧的低通滤波之类的编码环路之外的滤波技术通常不足以改善解码的视频序列的整体图像质量。参见 H. C. Reeve and J. S. Lim, "Reduction of Blocking Effects in Image Coding," Opt. Eng., Vol. 23, no. 1, pp. 34-37. Jan/Feb. 1984。分别在编码器和解码器中的编码和解码环路内部执行滤波的环路内滤波器具有极好的性能。关于更多的信息，参见 Wiegand, et al., "Overview of the H.264/AVC Video Coding Standard," IEEE Transactions on Circuits and Systems for Video Technology, on page(s) :560-576, vol. 13, Issue :7, July 2003。它们通常以试探的方式，根据视频编解码器所用的量化方法和要滤波的信号的特征，来改变其滤波强度。此外，由于它们工作在编码环路内，所以它们改善了用于预测的参考帧的质量，因而提高了编码过程的效率。

[0005] 存在与试探式工作相反的滤波技术，它们基于使用特定信号模型的对噪声的信号估计。关于更多的信息，参见 A. Papoulis, "Probability, Random Variables, and Stochastic Processes", 3rd edition, New York, McGraw-Hill, 1991。

[0006] 工作在变换域中的过完备 (overcomplete) 去噪声技术额外提供了以下优点：确定帧中相同采样位置的多个重构实例（估计值），然后可组合（例如平均）该多个重构

实例以便改善估计质量。参见 R. R. Coifman and D. L. Donoho, "Translation Invariant Denoising," in Wavelets and Statistics, Springer Lecture Notes in Statistics 103, pp. 125-150, New York, Springer Verlag。试图从编码噪声中提取信号的现有去噪声技术的核心在于信号估计器。由于通常并不精确地知道问题的统计数据,所以必须对未知数据做出一些假设,并选择估计信号的策略。一种最常用的策略是针对未知数据的最差可能选择来优化信号估计,从而产生鲁棒的估计器。关于更多的信息,参见 Y. C. Eldar, A. B. -Tal, and A. Nemirovski, "Linear Minimax Regret Estimation of Deterministic Parameters with Bounded Data Uncertainties," IEEE Transactions on Signal Processing, Vol. 52, No. 8, Aug. 2004。众所周知的是,在该设计的限制下,由于针对最差情况而优化了性能,所以估计器趋于太保守。

[0007] 编码噪声滤波的现有方法的性能受到这些方法的一些固有特征以及对噪声本质做出的假设的影响。用于减少编码伪像的简单的低通滤波技术对于处理视频帧中可视信息的多样性无效,并对于这些序列具有极大变化的性能(没有性能控制)。视频编码中的自适应、环路内的边界滤波使用试探法,在某种明确意义下并不确保最佳解决方案。此外,使用信号模型的现有去噪声技术在信号和噪声的本质的非理想假设下确定估计器,尤其是通过假设信号和噪声是不相关的。对于图像和视频编码中出现的信号和编码噪声,情况并不是这样的。由于这些滤波技术与现有问题条件匹配较差,所以这些滤波技术的性能受到损害。

发明内容

[0008] 这里公开一种方法和设备,用于对视频帧数据进行编码和/或解码。在一个实施例中,视频编码器包括噪声滤波模块,对变换的帧数据进行操作,并通过估计多个变换系数的每一个的信号功率,将所述每个系数的信号功率与至少一个阈值相比较,并至少部分基于信号功率与至少一个阈值的比较结果来设置所述每个变换系数的值,来对多个变换系数进行信号估计。

附图说明

[0009] 从下面给出的详细说明和本发明多个实施例的附图中,可更完全地理解本发明,然而,并不应该将本发明局限于特定实施例,而是这些实施例仅用于解释和理解。

[0010] 图 1 是视频编码器的一个实施例的框图。

[0011] 图 2 是视频编码过程的一个实施例的流程图。

[0012] 图 3 是噪声滤波器模块的一个实施例的框图。

[0013] 图 4 是噪声滤波过程的一个实施例的流程图。

[0014] 图 5 是噪声滤波过程的另一个实施例的流程图。

[0015] 图 6 是噪声滤波过程的另一个实施例的流程图。

[0016] 图 7 是估计过程的一个实施例的流程图。

[0017] 图 8 是估计过程的另一个实施例的流程图。

[0018] 图 9 是视频解码器的一个实施例的框图。

[0019] 图 10 是视频解码过程的一个实施例的流程图。

[0020] 图 11 是计算机系统的一个实施例的框图。

具体实施方式

[0021] 公开了用于实现视频编码和解码的信号去噪声技术的设备和方法。在一个实施例中,滤波设备对视频编解码器所产生的重构帧数据进行操作,并且滤波过程在与帧数据相对应的过完备变换域中进行。在该变换域中执行从由编码过程引入的噪声中估计信号。在一个实施例中,该估计过程包括估计与变换系数相对应的信号功率、将所估计的信号概率与一个或多个阈值相比较、以及根据比较结果来调整变换系数。因此,估计过程使用所估计的信号功率、噪声功率以及阈值。

[0022] 在一个实施例中,给定过完备表示,对于被处理的帧中的每个数据采样位置,该位置的采样对应于多个估计值,因为它们是由估计过程所产生的。进一步对每个数据采样的这些初始估计值进行滤波,以便获得所处理的视频帧中给定位置处的数据采样的最终估计值。这种滤波可包括平均。

[0023] 因此,将过完备变换域中的自适应估计用于从编码过程期间引入的噪声中提取信号。在一个实施例中,用于估计信号功率的噪声滤波器中的估计器根据与编码过程中的信号和噪声有关的假定来进行工作。此外,在一个实施例中,信号估计器设计用于实现鲁棒性和性能之间的平衡,以避免所述有关现有技术中的估计器的缺陷。

[0024] 与视频帧滤波的有关现有技术相比,该信号去噪声技术的使用实现了解码视频帧的提高了的客观速率-失真性能以及更好的主观(视觉)质量。

[0025] 在下面的说明中,给出多个细节,以提供对本发明的更详尽的解释。然而,对于本领域技术人员而言显而易见的是,可在没有这些具体细节的情况下实现本发明。在其它情况下,以框图形式而不是详细示出了公知结构和设备,以避免混淆本发明。

[0026] 下面详细说明的一些部分以计算机存储器内数据比特的运算的算法和符号表示来给出。这些算法说明和表示是数据处理领域中的技术人员用于更有效地向本领域技术人员表达其工作实质的手段。这里通常将算法看作产生所希望的结果的自洽步骤序列。所述步骤是需要对物理量进行物理操作的步骤。通常,尽管并不必须,这些量具有能够被存储、传输、组合、比较以及其它操作的电或磁信号的形式。主要是由于通常使用的原因,有时将这些信号称为比特、值、元素、符号、字符、项、数字等被证明是方便的。

[0027] 然而,应该记住,所有这些术语和类似术语都与适当的物理量相关联,并仅仅是运用于这些量的适当标签而已。除非特别指出,否则从下面的讨论中显而易见的是,在整个说明书中,使用诸如“处理”或“计算”或“运算”或“确定”或“显示”等术语的讨论是指计算机系统或类似电子计算设备的动作和处理,用于将由计算机系统的寄存器和存储器内的物理(电子)量表示的数据操作并变换为类似地由计算机系统存储器或寄存器或其它这些信息存储器、传输或显示设备内的物理量表示的其它数据。

[0028] 本发明还涉及用于执行这里所述操作的设备。该设备可为所需目的而特别构成,或者可包括由存储在计算机中的计算机程序选择性地激活或重新配置的通用计算机。这种计算机程序可存储在计算机可读存储介质中,计算机可读存储介质是例如但不限于:包括软盘、光盘、CD-ROM 和磁光盘的任意类型的盘、只读存储器 (ROM)、随机访问存储器 (RAM)、EPROM、EEPROM、磁或光卡或者适用于存储电子指令的任意类型的介质,并且每个均与计算机系统总线相耦合。

[0029] 这里所示的算法和显示本质上并不与任意特定的计算机或其它设备相关联。根据这里的教导,可与程序一起使用各种通用系统,或者构造更加专门化的设备来执行所需方法步骤被证明是方便的。从下面的说明中可显示各种这样的系统的所需结构。此外,并不参考任意特定编程语言来描述本发明。可以认识到,可使用各种编程语言来实现这里所述的本发明的教导。

[0030] 计算机可读介质包括用于以机器(例如计算机)可读的形式存储或传输信息的任何机制。例如,机器可读介质包括只读存储器(“ROM”);随机访问存储器(“RAM”);磁盘存储介质;光存储介质;闪存设备;电、光、声或其它传播信号形式(例如载波、红外信号、数字信号等)等等。

[0031] 视频编码器实施例的概述

[0032] 在一个实施例中,视频编解码器包括滤波模块,用于处理帧数据以便减小或消除编码噪声,因此提高编码过程的客观和主观质量。噪声滤波器采用在由过完备基底中的帧数据表示所构成的域中工作的自适应估计技术,以从有噪声的帧数据中提取视频信号。这里所述的设备和方法以与现有技术不同的方式,在帧数据的过完备基底表示中执行估计和滤波操作,并且与现有技术相比,它们的使用产生了优越的性能。与现有技术的一些不同包括但不限于:用于从噪声中提取信号的估计器的结构、该估计器所用的其它变量的估计、以及用于处理估计器所产生的信号估计值以便确定用于上述的显示或者参考和显示目的的最终滤波数据采样的滤波器结构。

[0033] 与现有技术的方法相比,本发明在这里所述的估计技术和/或滤波过程的使用产生了提高的数据编码效率(反映为编码过程的更好的速率-失真概图)以及通过减小或消除编码伪像而改善的解码视频序列的视觉质量。

[0034] 图1是视频编码器的一个实施例的框图。参考图1,视频编码器包括运动估计模块(MEM)129、运动补偿预测模块(MCPM)131、存储器123、变换编码模块(TCM)114、运动数据处理模块(MDPM)127、噪声滤波模块(NFM)125、帧存储器(FS)(例如存储器)126、以及开关118、124、128、141和142。变换编码器114则包括变换模块111、量化模块112以及熵编码模块113。变换编码器114的输入处的帧数据可包括视频帧或移位帧间差(DFD)帧。DFD帧是在视频编码器中通过采用视频帧中数据与在MCPM 131的输出处产生的对它的预测之间的差值而获得的。从接收视频帧101和I帧或内块(intra-block)(在开关141闭合时)的减法器103输出该差值。MCPM131根据包括存储在帧存储器126中的运动补偿重构视频帧的数据来产生预测。使用运动估计模块129所产生的运动信息来进行运动补偿。通过下面将讨论的图1中开关的配置能够进行该过程。可选地,当视频编码器(VC)与传统编码器的内编码模式相对应地,直接对视频帧进行编码而不使用预测时,并不使用MEM129、MCPM131和MDPM127。

[0035] 视频编码器通常依次对划分为块的帧数据进行工作,所述帧数据包括块中每个位置处的亮度和色度值。

[0036] 图2是视频编码过程的一个实施例的流程图。由可包括硬件(电路、专用逻辑电路等)、软件(例如运行在通用计算机系统或专用机器上的软件)或其组合的处理逻辑来执行该过程。

[0037] 在图2中,如下面过程所述的,依次读取并处理帧中的块,直到处理了当前帧中的

所有块为止。参考图 2, 该过程开始于处理逻辑将用于对当前帧中的块进行索引的索引变量 i 设置等于 1 (处理块 201)。接下来, 处理逻辑从当前帧中读取块 $c(i)$ (处理块 202)。

[0038] 在视频编码器操作的第一阶段, 图 1 所示的开关配置如下。开关 141、142 和 128 基于所处理的块的类型而打开或闭合 (对于内编码的块是闭合的, 对于传统视频过程中的预测块是闭合的)。开关 124 打开。

[0039] 处理逻辑还将索引 k 初始化为等于 1, 其中索引 k 是与当前块 $c(i)$ 相对应的搜索区域中的位置 (处理块 203), 并且处理逻辑在参考帧中检索与运动估计搜索区域中的位置 k 相对应的参考块 $b(k)$ (处理块 204)。执行运动估计过程, 从而依次搜索帧存储器中的参考帧中的搜索区域内的候选位置, 以识别当前块 $c(i)$ 的最佳匹配参考块。将匹配度量的良好程度用于对块 $c(i)$ 和 $b(k)$ 之间的差值进行量化。运动估计过程试图在参考帧的搜索区域中找到块 $b(k)$, 以便使当前块 $c(i)$ 与参考块 $b(k)$ 之间的差值 $d(i, k)$ 的度量最小化。该位置由起始于当前块 $c(i)$ 并指向参考帧中参考块 $b(k)$ 的运动矢量 $mv(i, k)$ 指定。

[0040] 处理逻辑计算块 $c(i)$ 和 $b(k)$ 之间的差值的度量 $d(i, k)$, 并将其存储在存储器中 (处理块 205)。在一个实施例中, 例如, 通过 $c(i)$ 和 $b(k)$ 的采样之间的绝对差值之和来量化该差值 $d(i, k)$ 。因此, 对于当前块 $c(i)$, 通过检索相应参考块 $b(k)$ 并计算 $d(i, k)$ 来测试搜索区域内的位置 k , 并将误差 $d(i, k)$ 存储在存储器中。

[0041] 处理逻辑然后使变量 k 递增 1 (处理块 206), 并测试变量 k 是否小于或等于 K , K 是搜索区域中位置的总数 (处理块 207)。如果尚未检查参考帧中与当前块 $c(i)$ 相对应的所有搜索区域位置 (即 $k < K$), 则处理逻辑跳转到处理块 204, 并从该块处继续进行处理。如果已经检查了上述所有搜索区域位置 (即 k 不小于 K), 则处理逻辑跳转到处理块 208, 在处理块 208 处, 处理逻辑找到 $c(i)$ 和 $b(k)$ 之间的 $d(i, k)$ 最小意义下的最佳匹配 $b(n)$ 。在一个实施例中, 在调查了与当前块 $c(i)$ 相对应的参考帧内所有 K 个搜索区域位置之后, MEM 129 的处理逻辑找到存储器中存储的值 $d(i, k)$ 阵列中的最小误差 $d_{\min}$, 其对应于由运动矢量 $mv(i)$ 指定坐标的用 $b_{\min}$ 所表示的特定参考块。

[0042] 其后, 处理逻辑在存储器中存储运动矢量 $mv(i, n)$ (处理块 209), 决定并存储块 $c(i)$ 的编码模式 $m(i)$, 其中, 块 $c(i)$ 的编码模式 $m(i)$ 是 INTRA 或预测 (处理块 210)。在一个实施例中, MEM 129 的处理逻辑通过将在对当前块 $c(i)$ 的采样进行内编码时当前块 $c(i)$ 的编码成本与在相对于块 $b_{\min}$ 预测地对块 $c(i)$ 进行编码的成本相比较, 来确定当前块 $c(i)$ 的编码模式 $cmode(i)$ 。如果前一成本更小, 则将块 $c(i)$ 的模式标记为 INTRA; 否则, 将块标记为预测块。

[0043] 如果块 $c(i)$ 的编码模式不是 INTRA, 则将运动矢量 $mv(i)$ 存储在存储器中。还将当前块的模式 $cmode(i)$ 写入存储器中, 并且通过由处理逻辑使索引变量 i 递增 1 (处理块 211), 并测试 i 是否小于 N (N 是当前帧中块的总数) (处理块 212), 过程前进到当前帧中的下一块 $c(i)$ 。如果索引 i 小于 N , 则处理逻辑跳转到处理块 202, 并从该块处继续进行处理。如果不小于, 则处理逻辑跳转到处理块 213, 在处理块 213 处, 将索引变量 i 设置为等于 1。

[0044] 一旦如上所述处理了当前帧中的所有块, 则启动运动补偿过程。运动补偿过程开始于处理逻辑从当前帧中读取初始当前块 $c(i)$ (处理块 214)。在一个实施例中, 依次读取块。处理逻辑测试块 $c(i)$ 的编码模式是否等于 INTRA (处理块 215)。在一个实施例中, 通过 MCPM 131 中的处理逻辑来执行该过程。如果块 $c(i)$ 的编码模式等于 INTRA, 则处理逻辑

辑跳转到处理块 219。如果不等于,则处理逻辑执行以下操作:使用 $mv(i,n)$ 来执行运动补偿,以便从参考帧中获取块预测 $b(i)$ (处理块 216);根据以下公式来计算预测误差 $e(i)$:

$$[0045] \quad e(i) = c(i) - p(i)$$

[0046] 其中,预测块 $p(i)$ 由 MCPM 131 使用块 $c(i)$ 的存储在存储器中的运动矢量 $mv(i)$ 来产生 (处理块 217);对运动信息进行熵编码 (处理块 218);并跳转到处理块 219。在一个实施例中,将与当前块相对应的运动矢量数据发送到 MDPM 127,在 MDPM 127 处,对其进行处理 (例如相对于相邻块的运动矢量来进行 DPCM 编码),并将其发送到熵编码器模块 113 处编码。

[0047] 在处理块 219 处,处理逻辑对 $c(i)$ 的编码模式进行熵编码。处理逻辑然后基于模式 $m(i)$ 来对数据块 $c(i)$ 或 $e(i)$ 进行代码变换 (处理块 220),重构当前块 (处理块 221),并将其写入存储器 (例如存储器 123) (处理块 222)。

[0048] 针对当前帧中所有的块重复上述过程。为此,处理逻辑使索引 i 递增 1 (处理块 223),并测试索引 i 是否小于 N (N 是当前帧中块的总数) (处理块 224)。如果小于,则处理逻辑跳转到处理块 214,并从该块继续进行处理。如果不小于,则处理逻辑对存储器 (例如存储器 123) 中有噪声的帧数据进行滤波 (处理块 225),并将滤波后的数据写入帧存储器 (例如帧存储器 126) 中 (处理块 226)。其后,该过程结束。

[0049] 在处理了当前帧中所有的块之后,开关 124 闭合。噪声滤波模块 125 如下所述来处理存储在存储器 123 中的重构的帧数据。噪声滤波模块 125 将其处理的数据存储在帧存储器 126 中。

[0050] 噪声滤波

[0051] 图 3 是噪声滤波模块 (例如图 1 的噪声滤波模块 125) 的一个实施例的框图。参考图 3,噪声滤波模块包括存储器 301、空间掩蔽 (mask) 位置产生模块 (MPGM) 302、变换模块 (TM) 303、估计器模块 (EM) 304、存储器 305、逆变换模块 (ITM) 306、存储器 307 以及滤波器模块 (FM) 308。存储器 301 向掩蔽位置产生器 302 提供掩蔽结构 310。响应于此,掩蔽位置产生器 302 给出帧 311 中的位置,并将这些位置提供给变换模块 303、逆变换 306 以及滤波器 308。变换模块 303 向解码后的帧数据 312 应用变换,以产生在掩蔽位置产生器 302 所给出的掩蔽之下的采样的变换系数 313。估计器 305 接收变换系数 313,并响应于此,产生去噪声的变换系数 314。将去噪声的变换系数 314 存储在存储器 305 中。随后,逆变换模块 306 根据掩蔽位置产生器 302 的掩蔽,向去噪声的变换系数应用逆变换,以产生在掩蔽之下的去噪声的采样 315。将这些去噪声采样存储在存储器 307 中。滤波模块 308 根据掩蔽位置产生器 302 的掩蔽,对去噪声采样 315 执行滤波,以产生重构的帧采样 316。

[0052] 噪声滤波模块的输入包括存储在存储器模块 123 中的重构的帧数据 312 (例如解码后的帧数据)。噪声滤波模块向帧存储器 126 输出滤波后的帧数据 316 (例如重构的帧采样)。由图 4、5 和 6 中所示的流程图示出了噪声滤波模块进行的操作。

[0053] 图 4 是噪声滤波过程的实施例的流程图。由可包括硬件 (电路、专用逻辑等)、软件 (例如运行在通用计算机系统或专用机器上的软件) 或其组合的处理逻辑来执行该过程。在一个实施例中,处理逻辑是图 3 的噪声滤波器模块的估计器 304 的一部分。

[0054] 噪声滤波器模块所执行的过程开始于从存储器 (例如存储器 123) 中检索其中存储的重构的帧数据。使用重构的帧数据,处理逻辑计算一组变量 V (处理块 401)。将在估计

过程中进一步使用该组辅助变量V。根据编解码器所使用的量化方法,估计器计算编码过程所引入的量化噪声的度量。为此,由q表示编码过程中使用的量化步骤。在一个实施例中,处理逻辑计算辅助变量

$$[0055] \quad v^2 = \alpha \frac{q^2}{4}$$

[0056] 其中 α 是常数, v^2 表示噪声功率的估计。估计器中的处理逻辑还根据编码器中使用的编码范围的特性,确定变量b的值。在这种情况下,b被计算为 $b = \beta v^2$,其中 β 是常数。在另一个实施例中,处理逻辑如下计算附加的辅助变量:

$$[0057] \quad t_1 = \mu v^2 - (\mu - 1)b$$

$$[0058] \quad x_1 = \sqrt{b}$$

$$[0059] \quad x_2 = \sqrt{t_1}$$

$$[0060] \quad z_1 = \frac{1}{x_2 - x_1}$$

$$[0061] \quad z_2 = x_1 z_1$$

[0062] 其中 μ 是可选常数。

[0063] 在另一个实施例中,处理逻辑计算下面的辅助变量:

$$[0064] \quad t_1 = \mu v^2 - (\mu - 1)b$$

$$[0065] \quad x_1 = \sqrt{b}$$

$$[0066] \quad x_2 = \sqrt{t_1}$$

$$[0067] \quad y_2 = \frac{\mu}{\mu + 1} x_2,$$

$$[0068] \quad h_1 = \frac{y_2}{x_2 - x_1},$$

$$[0069] \quad h_2 = \frac{y_2 x_1}{x_2 - x_1}$$

[0070] 其中 μ 是可选常数。

[0071] 处理逻辑然后设置具有支持区域R的掩蔽S(处理块402)。例如,可使掩蔽大小(支持区域R)等于编码过程所使用的块的大小。

[0072] 其后,处理逻辑将(i,j)初始化为帧中块的位置(处理块403)。从存储器(例如存储器123)中的当前被滤波的帧中选择位于视频编解码器通常用于编码/解码帧的块隔栅上的块位置(i,j)。

[0073] 处理逻辑还将(m,n)初始化为围绕(i,j)以及包含(i,j)的位置(处理块404)。位置(m,n)表示掩蔽S相对于(i,j)偏移的位置。处理逻辑然后将掩蔽S移动到帧中位置(m,n)(处理块405)。

[0074] 处理逻辑对在掩蔽S之下的采样运用变换T(例如离散余弦变换(DCT))(处理块406),并使用估计器中的处理逻辑来对变换的系数去噪声(处理块407)。这里由s(k)(k=1...N)表示与由掩蔽S的当前位置所覆盖的数据采样相对应的N个变换系数。由估计器模块中的处理逻辑来处理所产生的变换系数。在图7和8的流程图中示出了该操作的不同实施例,下面将更详细地进行描述。

[0075] 在图 7 的流程图中,估计器模块中的处理逻辑取得变换系数 $s(k)$ 并通过计算值 $s^2(k)$ 来估计信号功率。处理逻辑然后将信号功率与阈值相比较,并如下确定与系数 $s(k)$ 相对应的估计值 $a(k)$:

[0076] if $s^2(k) \leq b$

[0077] $a(k) = 0$;

[0078] else

$$[0079] \quad a(k) = \frac{s^2(k) - b}{s^2(k) + v^2 - 2b}$$

[0080] 估计器中的处理逻辑然后根据下式来确定处理后的变换系数 $p(k)$:

[0081] $p(k) = a(k) s(k)$

[0082] 在可选实施例中,使用图 8 的流程图中所示的过程,估计器中的处理逻辑使用上文先前定义的辅助变量 V ,并如下针对每个系数 $s(k)$ 计算估计器的值 :

[0083] if $s^2(k) \leq b$

[0084] $a(k) = 0$;

[0085] else if $b < s^2(k) \leq t_1$

[0086] $a(k) = z_1 s(k) - z_2$

[0087] else

[0088] $a(k) = 1$

[0089] 估计器中的处理逻辑然后如下确定处理后的变换系数 $p(k)$:

[0090] $p(k) = a(k) s(k)$

[0091] 在可选实施例中,同样使用图 8 的流程图所示的过程,针对每个系数 $s(k)$,估计器中的处理逻辑如下计算处理后的系数 $p(k)$ 的值 :

[0092] if $s^2(k) \leq b$

[0093] $p(k) = 0$;

[0094] else if $b < s^2(k) \leq t_1$

[0095] $p(k) = h_1 s(k) - h_2$;

[0096] else

[0097] $p(k) = s(k)$ 。

[0098] 下面将更详细地描述图 8。

[0099] 处理逻辑将系数 $p(k)$ 存储在存储器 (例如存储器 305) 中,不论使用了图 7 还是图 8 的过程。如果尚未处理所有的系数,则重复上述过程。

[0100] 接下来,处理逻辑向在掩蔽 S 之下的处理后的变换系数 $p(k)$ 运用逆变换 T^{-1} (例如逆 DCT 变换) (处理块 408),产生与帧中的掩蔽 S 的当前位置 (m, n) 相对应的采样 $r(k)$ 。处理逻辑将这些在掩蔽 S 之下的处理后的采样存储在存储器 (例如存储器 307) 中 (处理块 409)。

[0101] 处理逻辑然后将 (m, n) 设置为下一个位置 (处理块 410),并测试是否检查了支持区域 R 中的所有位置 (m, n) (处理块 411)。如果尚未处理块位置 (i, j) 周围的掩蔽 S 的所有位置,则处理跳转到处理块 405,选择掩蔽 S 相对于 (i, j) 的下一个偏移位置 (m, n) ,并重复上述过程。

[0102] 如果处理了块位置 (i, j) 周围掩蔽 S 的所有位置, 则处理块跳转到处理块 412, 在处理块 412 处, 处理逻辑在帧的块隔栅上选择下一个块位置 (i, j), 测试是否处理了帧中要处理的所有块位置 (i, j) (处理块 413)。如果否, 则处理逻辑跳转到处理块 404, 过程从处理块 404 处继续。如果是, 则处理跳转到处理块 414。按照这种方式, 重复上述过程, 直到检查了所有块位置 (i, j) 为止。

[0103] 在处理块 414 处, 处理逻辑使用滤波器 (例如滤波器 308), 对与帧中每个采样位置相对应的重构版本进行滤波 (处理块 414), 并将所产生的被处理的当前帧的滤波后数据采样 $f(x, y)$ 写入帧存储器 (例如帧存储器 126) 中 (处理块 415)。在一个实施例中, 处理逻辑对存储在存储器 (例如存储器 307) 中的与当前帧中每个采样相对应的处理后采样进行滤波。在一个实施例中, 由噪声滤波模块 125 的滤波器 308 执行该滤波。因此, 对于噪声滤波模块 125 处理的当前帧中的每个位置 (x, y), 存在由估计器 304 存储在存储器 307 中的多个经处理并重构的采样 $r^l(x, y)$, 其中 $l = 1 \dots L$ 对帧中块隔栅位置周围的掩蔽 S 可占据的位置进行索引, 并且在每个这种位置处, 掩蔽 S 的支持区域 R 均包含采样位置 (x, y)。在一个实施例中, 滤波器对与帧位置 (x, y) 相对应的输入数据 $r^l(x, y)$ 执行加权平均运算:

$$[0104] \quad f(x, y) = \sum_l w_l r^l(x, y),$$

[0105] 其中 w_l 是基于估计器中的处理逻辑所产生的统计数据而选择的权重。在一个实施例中, 选择所有权重均相等, $w_l = \frac{1}{L}$, 并对应于平均运算。在可选实施例中, 选择权重与在掩蔽 S 的位置 1 处应用估计器之后产生的零值处理后系数 $p(k)$ 的个数成比例。

[0106] 然后该过程结束。

[0107] 图 5 是噪声滤波模块所执行的噪声滤波过程的可选实施例的流程图。由可包括硬件 (电路、专用逻辑等)、软件 (例如运行在通用计算机系统或专用机器上的软件) 或其组合的处理逻辑来执行该过程。在一个实施例中, 处理逻辑是图 3 的噪声滤波模块的估计器 304 的一部分。

[0108] 参考图 5, 该过程开始于处理逻辑产生具有支持区域 R 的掩蔽 S (处理块 501)。例如, 可使掩蔽大小 (支持区域 R) 等于编码过程中使用的块的大小。

[0109] 处理逻辑计算一组变量 $V(k)$ (处理块 502)。由于向在掩蔽 S 之下的采样应用了变换 (例如 DCT) (在处理块 303 处), 所以可按照惯例针对这种变换排列变换系数。该排列由 k 表示。在本实施例中, 变量 $v(k)$ 基于在掩蔽 S 之下产生的变换系数的排列 k。在估计过程中将进一步使用这组辅助变量 $v(k)$ 。基于编解码器所使用的量化方法, 处理逻辑计算由编码过程所引入的量化噪声的度量。为此, 由 q 表示在编码过程中使用的量化步骤。在这种情况下, 处理逻辑计算变量

$$[0110] \quad v^2(k) = \alpha(k) \frac{q^2}{4}。$$

[0111] 其中, 可根据诸如掩蔽 S 内系数 $s(k)$ 的排列 k 之类的各种因素来改变 α , 在这种情况下 $\alpha = \alpha(k)$ 。例如, 系数排序越高, α 越大。如果 v 是 k 的函数, 则 b 也成为系数排列 k 的函数, $b(k) = \beta v^2(k)$ 。在另一个实施例中, 处理逻辑基于在掩蔽 S 之下 (S 内的位置) 的变换系数的排列 k, 如下计算附加辅助变量:

$$[0112] \quad t_1(k) = \mu v^2(k) - (\mu - 1)b(k)$$

$$[0113] \quad x_1(k) = \sqrt{b(k)}$$

$$[0114] \quad x_2(k) = \sqrt{t_1(k)}$$

$$[0115] \quad z_1(k) = \frac{1}{x_2(k) - x_1(k)}$$

$$[0116] \quad z_2(k) = x_1(k) z_1(k)$$

[0117] 其中 μ 是可选常数。

[0118] 可选地, 在另一个实施例中, 处理逻辑计算下面的辅助变量:

$$[0119] \quad t_1(k) = \mu v^2(k) - (\mu - 1)b(k)$$

$$[0120] \quad x_1(k) = \sqrt{b(k)}$$

$$[0121] \quad x_2(k) = \sqrt{t_1(k)}$$

$$[0122] \quad y_2(k) = \frac{\mu}{\mu + 1} x_2(k),$$

$$[0123] \quad h_1(k) = \frac{y_2(k)}{x_2(k) - x_1(k)},$$

$$[0124] \quad h_2(k) = \frac{y_2(k)x_1(k)}{x_2(k) - x_1(k)}$$

[0125] 其中 μ 是可选常数。

[0126] 处理逻辑然后将 (i, j) 初始化为帧中块的位置 (处理块 503)。在一个实施例中, 从存储器 (例如存储器 123) 中的当前被滤波的帧中选择位于视频编解码器通常用于编码/解码帧的块隔栅上的块位置 (i, j) 。

[0127] 处理逻辑还将 (m, n) 初始化为围绕 (i, j) 以及包含 (i, j) 的位置 (处理块 504)。位置 (m, n) 是掩蔽 S 相对于 (i, j) 偏移的位置。处理逻辑将掩蔽 S 移动到帧中的位置 (m, n) (处理块 505)。

[0128] 处理逻辑对在掩蔽 S 之下的采样运用变换 T (例如 DCT) (处理块 506), 并对变换系数去噪声 (处理块 507)。这里由 $s(k)$ ($k = 1 \dots N$) 表示与由掩蔽 S 的当前位置所覆盖的数据采样相对应的 N 个变换系数。由估计器模块中的处理逻辑来处理所产生的变换系数。在图 7 和 8 的流程图中示出了该操作的不同实施例, 下面将更详细地进行描述。

[0129] 在图 7 的流程图中, 估计器模块中的处理逻辑取得变换系数 $s(k)$ 并通过计算值 $s^2(k)$ 来估计信号功率。处理逻辑然后将信号功率与阈值相比较, 并如下确定系数 $s(k)$ 的估计器值:

$$[0130] \quad \text{if } s^2(k) \leq b(k)$$

$$[0131] \quad a(k) = 0;$$

$$[0132] \quad \text{else}$$

$$[0133] \quad a(k) = \frac{s^2(k) - b(k)}{s^2(k) + v^2(k) - 2b(k)}.$$

[0134] 估计器中的处理逻辑然后根据下式来确定处理后的变换系数 $p(k)$:

$$[0135] \quad p(k) = a(k)s(k)$$

[0136] 在可选实施例中,使用图 8 的流程图中所示的过程,估计器中的处理逻辑使用辅助变量,并如下针对每个系数 $s(k)$ 计算估计器的值:

[0137] if $s^2(k) \leq b(k)$

[0138] $a(k) = 0$;

[0139] else if $b(k) < s^2(k) \leq t_1(k)$

[0140] $a(k) = z_1(k)s(k) - z_2(k)$

[0141] else

[0142] $a(k) = 1$ 。

[0143] 估计器中的处理逻辑然后如下确定处理后的变换系数 $p(k)$:

[0144] $p(k) = a(k)s(k)$

[0145] 在可选实施例中,同样使用图 8 的流程图所示的过程,针对每个系数 $s(k)$,估计器中的处理逻辑如下计算处理后的系数 $p(k)$ 的值:

[0146] if $s^2(k) \leq b(k)$

[0147] $p(k) = 0$;

[0148] else if $b(k) < s^2(k) \leq t_1(k)$

[0149] $p(k) = h_1(k)s(k) - h_2(k)$;

[0150] else

[0151] $p(k) = s(k)$ 。

[0152] 下面将更详细地描述图 8。

[0153] 处理逻辑存储系数 $p(k)$ (例如在存储器 305) 中,不论使用了图 7 还是图 8 的过程。如果尚未处理所有的系数,则重复上述过程。

[0154] 在去噪声之后,处理逻辑向在掩蔽 S 之下的变换系数运用逆变换 T^{-1} (例如逆 DCT) (处理块 508),产生与帧中的掩蔽 S 的当前位置 (m, n) 相对应的采样 $r(k)$,并将在掩蔽 S 之下的重构的采样存储在存储器 (例如存储器 307) 中 (处理块 509)。

[0155] 处理逻辑然后将 (m, n) 设置为下一个位置 (处理块 510),并测试是否检查了支持区域 R 中的所有位置 (m, n) (处理块 511)。如果尚未处理块位置 (i, j) 周围的掩蔽 S 的所有位置,则处理跳转到处理块 505,选择掩蔽 S 相对于 (i, j) 的下一个位置 (m, n) ,并重复上述过程。

[0156] 如果处理了块位置 (i, j) 周围掩蔽 S 的所有位置,则处理块跳转到处理块 512,在处理块 512 处,处理逻辑在帧的块隔栅上选择下一个块位置 (i, j) ,测试是否处理了帧中被处理的所有块位置 (i, j) (处理块 513)。如果否,则处理逻辑跳转到处理块 504,过程从处理块 504 处继续。如果是,则处理跳转到处理块 514。按照这种方式,重复上述过程,直到检查了所有块位置 (i, j) 为止。

[0157] 在处理块 514 处,处理逻辑使用滤波器,对与帧中每个采样位置相对应的重构版本进行滤波 (处理块 514)。在一个实施例中,所使用的滤波器是滤波器 308。因此,对于噪声滤波模块 125 处理的当前帧中的每个位置 (x, y) ,存在由估计器存储在存储器 307 中的多个被处理并重构的采样 $r^l(x, y)$,其中 $l = 1 \dots L$ 对帧中块隔栅位置周围的掩蔽 S 可占据的位置进行索引,并且在每个这种位置处,掩蔽的支持区域 R 均包含采样位置 (x, y) 。在一个实施例中,滤波器 308 对与帧位置 (x, y) 相对应的其输入数据 $r^l(x, y)$ 执行加权平均运

算：

$$[0158] \quad f(x, y) = \sum_l w_l r^l(x, y),$$

[0159] 其中 w_l 是基于估计器所产生的统计数据而选择的权重。在一个实施例中,选择所有权重均相等, $w_l = \frac{1}{L}$ 。在可选实施例中,选择权重从而与在掩蔽 S 的位置 1 处应用估计器之后产生的零值处理后系数 $p(k)$ 的个数成比例。

[0160] 处理逻辑将所产生的被处理的当前帧的滤波后数据采样 $f(x, y)$ 写入帧存储器(例如帧存储器 126)中(处理块 515)。

[0161] 然后该过程结束。

[0162] 图 6 是噪声滤波模块所执行的噪声滤波过程的另一个可选实施例的流程图。由可包括硬件(电路、专用逻辑等)、软件(例如运行在通用计算机系统或专用机器上的软件)或其组合的处理逻辑来执行该过程。

[0163] 参考图 6,该过程开始于处理逻辑产生具有支持区域 R 的掩蔽 S(处理块 601)。在一个实施例中,可使掩蔽大小(支持区域 R)等于编码过程中使用的块的大小。

[0164] 处理逻辑然后使 (i, j) 初始化为帧中的块位置(处理块 602)。在一个实施例中,从存储器(例如存储器 123)中的当前被滤波的帧中选择位于视频编解码器通常用于编码/解码帧的块隔栅上的块位置 (i, j) 。

[0165] 处理逻辑还将 (m, n) 初始化为围绕 (i, j) 以及包含 (i, j) 的位置(处理块 603)。位置 (m, n) 是掩蔽 S 相对于 (i, j) 偏移的位置。处理逻辑将掩蔽 S 移动到帧中位置 (m, n) (处理块 604)。

[0166] 估计器中的处理逻辑计算不仅基于在当前掩蔽位置处由 303 所产生的变换系数的排列 k (处理块 605)、而且基于帧中掩蔽 S 的位置 (m, n) 的一组变量 $V(m, n, k)$ 。在估计过程中将进一步使用这组辅助变量。基于编解码器所使用的量化方法,处理逻辑计算编码过程所引入的量化噪声的度量。为此,由 q 表示编码过程中使用的量化步骤。在这种情况下,处理逻辑计算变量

$$[0167] \quad v^2(m, n, k) = \alpha(m, n, k) \frac{q^2}{4}$$

[0168] 其中,可根据诸如掩蔽 S 内系数 $s(k)$ 的排列 k 以及 (i, j) 周围掩蔽 S 的位置 (m, n) 之类的各种因素来改变 α , 在这种情况下 $\alpha = \alpha(m, n, k)$ 。例如,系数的排序越高, α 越大。如果 v 是 (m, n, k) 的函数,则 b 也成为系数排列 m, n, k 的函数, $b(m, n, k) = \beta v^2(m, n, k)$ 。在另一个实施例中,处理逻辑基于映射到掩蔽 S(S 内的位置)的变换系数的排列 m, n, k , 如下计算附加辅助变量:

$$[0169] \quad t_1(m, n, k) = \mu v^2(m, n, k) - (\mu - 1)b(m, n, k)$$

$$[0170] \quad x_1(m, n, k) = \sqrt{b(m, n, k)}$$

$$[0171] \quad x_2(m, n, k) = \sqrt{t_1(m, n, k)}$$

$$[0172] \quad z_1(m, n, k) = \frac{1}{x_2(m, n, k) - x_1(m, n, k)}$$

$$[0173] \quad z_2(m, n, k) = x_1(m, n, k) z_1(m, n, k)$$

[0174] 其中 μ 是可选常数。

[0175] 可选地,在另一个实施例中,处理逻辑计算下面的辅助变量:

$$[0176] \quad t_1(m, n, k) = \mu v^2(m, n, k) - (\mu - 1)b(m, n, k)$$

$$[0177] \quad x_1(m, n, k) = \sqrt{b(m, n, k)}$$

$$[0178] \quad x_2(m, n, k) = \sqrt{t_1(m, n, k)}$$

$$[0179] \quad y_2(m, n, k) = \frac{\mu}{\mu + 1} x_2(m, n, k),$$

$$[0180] \quad h_1(m, n, k) = \frac{y_2(m, n, k)}{x_2(m, n, k) - x_1(m, n, k)},$$

$$[0181] \quad h_2(m, n, k) = \frac{y_2(m, n, k)x_1(m, n, k)}{x_2(m, n, k) - x_1(m, n, k)}$$

[0182] 其中 μ 是可选常数。

[0183] 处理逻辑对在掩蔽 S 之下的采样运用变换 T (例如 DCT) (处理块 606), 并使用估计器中的处理逻辑对变换系数去噪声 (处理块 607)。这里由 $s(m, n, k)$ ($k = 1 \dots N$) 表示与由掩蔽 S 的当前位置所覆盖的数据采样相对应的 N 个变换系数。由估计器模块中的处理逻辑来处理所产生的变换系数。在图 7 和 8 的流程图中示出了该操作的不同实施例, 下面将更详细地进行描述。

[0184] 在图 7 的流程图中, 估计器模块中的处理逻辑取得变换系数 $s(m, n, k)$ 并通过计算值 $s^2(m, n, k)$ 来估计信号功率。处理逻辑然后将信号功率与阈值相比较, 并如下确定系数 $s(m, n, k)$ 的估计器值:

$$[0185] \quad \text{if } s^2(m, n, k) \leq b(m, n, k)$$

$$[0186] \quad a(m, n, k) = 0;$$

[0187] else

$$[0188] \quad a(m, n, k) = \frac{s^2(m, n, k) - b(m, n, k)}{s^2(m, n, k) + v^2(m, n, k) - 2b(m, n, k)}。$$

[0189] 估计器中的处理逻辑然后根据下式来确定处理后的变换系数 $p(m, n, k)$:

$$[0190] \quad p(m, n, k) = a(m, n, k) s(m, n, k)$$

[0191] 在可选实施例中, 使用图 8 的流程图中所示的过程, 估计器中的处理逻辑使用辅助变量, 并如下针对每个系数 $s(m, n, k)$ 计算估计器的值:

$$[0192] \quad \text{if } s^2(m, n, k) \leq b(m, n, k)$$

$$[0193] \quad a(m, n, k) = 0;$$

$$[0194] \quad \text{else if } b(m, n, k) < s^2(m, n, k) \leq t_1(m, n, k)$$

$$[0195] \quad a(m, n, k) = z_1(m, n, k) s(m, n, k) - z_2(m, n, k)$$

[0196] else

$$[0197] \quad a(m, n, k) = 1。$$

[0198] 估计器中的处理逻辑然后如下确定处理后的变换系数 $p(m, n, k)$:

$$[0199] \quad p(m, n, k) = a(m, n, k) s(m, n, k)。$$

[0200] 在可选实施例中, 同样使用图 8 的流程图所示的过程, 针对每个系数 $s(m, n, k)$, 估

计器中的处理逻辑如下计算处理后的系数 $p(m, n, k)$ 的值：

[0201] if $s^2(m, n, k) \leq b(m, n, k)$

[0202] $p(m, n, k) = 0$;

[0203] else if $b(m, n, k) < s^2(m, n, k) \leq t_1(m, n, k)$

[0204] $p(m, n, k) = h_1(m, n, k) s(m, n, k) - h_2(m, n, k)$;

[0205] else

[0206] $p(m, n, k) = s(m, n, k)$ 。

[0207] 下面将更详细地描述图 8。

[0208] 处理逻辑将系数 $p(k)$ 存储在存储器（例如存储器 305）中，不论使用了图 7 还是图 8 的过程。如果尚未处理所有的系数，则重复上述过程。

[0209] 接下来，处理逻辑向在掩蔽 S 之下的变换系数运用逆变换 T^{-1} （处理块 608），产生与帧中的掩蔽 S 的当前位置 (m, n) 相对应的采样 $r(m, n, k)$ 。处理逻辑将这些在掩蔽 S 之下的重构采样存储在存储器（例如存储器 307）中（处理块 609）。

[0210] 处理逻辑然后将 (m, n) 设置为下一个位置（处理块 610），并测试是否检查了支持区域 R 中的所有位置 (m, n) （处理块 611）。如果尚未处理块位置 (i, j) 周围的掩蔽 S 的所有位置，则处理跳转到处理块 605，选择掩蔽 S 相对于 (i, j) 的下一个位置 (m, n) ，并重复上述过程。

[0211] 如果处理了块位置 (i, j) 周围掩蔽 S 的所有位置，则处理块跳转到处理块 612，在处理块 612 处，处理逻辑在帧的块隔栅上选择下一个块位置 (i, j) ，测试是否处理了帧中被处理的所有块位置 (i, j) （处理块 613）。如果否，则处理逻辑跳转到处理块 604，过程从处理块 604 处继续。如果是，则处理跳转到处理块 614。按照这种方式，重复上述过程，直到检查了所有块位置 (i, j) 为止。

[0212] 在处理块 614 处，处理逻辑使用滤波器（例如滤波器 308），对与帧中每个采样位置相对应的重构版本进行滤波（处理块 614），并将所产生的被处理的当前帧的滤波后数据采样 $f(x, y)$ 写入帧存储器（例如帧存储器 126）中（处理块 615）。在一个实施例中，处理逻辑对在存储器（例如存储器 307）中存储的与当前帧中每个采样相对应的估计采样进行滤波。在一个实施例中，该滤波由噪声滤波模块 125 的滤波器 308 执行。因此，对于噪声滤波模块 125 处理的当前帧中的每个位置 (x, y) ，存在由估计器 304 存储在存储器中的多个被处理并重构的采样 $r^l(x, y)$ ，其中 $l = 1 \dots L$ 对帧中块隔栅位置周围的掩蔽 S 可占据的位置进行索引，并且在每个这种位置处，掩蔽的支持区域 R 均包含采样位置 (x, y) 。在一个实施例中，滤波器对与帧位置 (x, y) 相对应的其输入数据 $r^l(x, y)$ 执行加权平均运算：

$$[0213] \quad f(x, y) = \sum_l w_l r^l(x, y),$$

[0214] 其中 w_l 是基于估计器中的处理逻辑所产生的统计数据而选择的权重。在一个实施例中，选择所有权重均相等， $w_l = \frac{1}{L}$ 。在可选实施例中，选择权重与在掩蔽 S 的位置 1 处应用估计器之后产生的零值处理后系数 $p(k)$ 的个数成比例。

[0215] 然后该过程结束。

[0216] 估计过程的示例

[0217] 图 7 是估计过程的一个实施例的流程图。由可包括硬件（电路、专用逻辑等）、软

件（例如运行在通用计算机系统或专用机器上的软件）或其组合的处理逻辑来执行该过程。

[0218] 参考图 7, 该过程开始于处理逻辑取得变换系数（处理块 701）并估计变换系数的信号功率（处理块 702）。在一个实施例中, 通过取变换系数的平方来估计信号功率。

[0219] 处理逻辑然后测试信号概率是否小于阈值（处理块 703）。在上文第 52 段中已经描述了阈值的一个实施例。如果是, 则处理逻辑将系数设置为 0（处理块 704）, 并且处理跳转到处理块 706, 在处理块 706 处, 存储系数。如果否, 则处理逻辑根据信号功率、噪声功率和阈值, 通过执行上述逻辑过程, 来调整系数值（处理块 705）, 并且处理跳转到处理块 706, 在处理块 706 处存储系数。

[0220] 在存储系数之后, 处理逻辑测试是否处理了所有系数（处理块 707）。如果否, 则处理逻辑跳转到处理块 701, 并且从该点开始继续进行处理。如果是, 则该过程结束。

[0221] 图 8 是估计过程的可选实施例的流程图。由可包括硬件（电路、专用逻辑等）、软件（例如运行在通用计算机系统或专用机器上的软件）或其组合的处理逻辑来执行该过程。

[0222] 参考图 8, 该过程开始于处理逻辑取得变换系数（处理块 801）并估计变换系数的信号功率（处理块 802）。在一个实施例中, 通过取变换系数的平方来估计信号功率。处理逻辑然后测试信号功率是否小于阈值（处理块 803）。在上文第 71 段中描述了阈值的一个实施例。如果是, 则处理逻辑将系数设置为 0（处理块 804）, 并且处理跳转到处理块 806, 在处理块 806 处, 存储系数。如果否, 则处理逻辑测试信号功率是否小于第二阈值（处理块 805）。在上文第 93 段中描述该阈值的实施例。如果是, 则处理逻辑根据信号功率、噪声功率和阈值, 通过执行上述逻辑过程, 来调整系数值（处理块 806）, 并且处理跳转到处理块 808, 在处理块 808 处存储系数。如果否, 则处理逻辑保持系数不变（处理块 807）, 并跳转到处理块 808, 在处理块 808 处, 存储系数。

[0223] 在存储系数之后, 处理逻辑测试是否处理了所有系数（处理块 809）。如果否, 则处理逻辑跳转到处理块 701, 并且从该点开始继续进行处理。如果是, 则该过程结束。

[0224] 视频解码器的示例

[0225] 图 9 是视频解码器（VD）的一个实施例的框图。参考图 9, 视频解码器包括变换解码器 913、加法器 904、存储器模块 905、运动补偿和预测模块（MCPM）907、运动数据处理模块（MDPM）908、帧存储器 909、噪声滤波模块（NFM）906、以及开关 941-943。在一个实施例中, 变换解码器 913 包括熵解码器 901、逆量化器 902、以及逆变换模块 903。

[0226] 视频解码器根据下面结合 10 的流程图所规定的过程来重构视频帧数据。开关 942 和 943 与传统编码过程的操作类似地, 根据解码块的类型（内编码或预测）而打开或闭合。

[0227] 图 10 是视频解码过程的一个实施例的流程图。由可包括硬件（电路、专用逻辑等）、软件（例如运行在通用计算机系统或专用机器上的软件）或其组合的处理逻辑来执行该过程。

[0228] 参考图 10, 该过程开始于处理逻辑将索引变量 i 初始化等于 1（处理块 1001）。

[0229] 处理逻辑然后对被解码的视频帧中的当前块 $c(i)$ 的编码模式进行解码（处理块 1002）, 并测试块 $c(i)$ 的编码模式是否等于 INTRA（处理块 1003）。如果是, 则处理逻辑根据传统过程对当前块 $c(i)$ 进行内解码（处理块 1004）, 并跳转到处理块 1009。如果否, 则处理

逻辑使用变换解码器 913,对当前块 $c(i)$ 的块预测误差 $e(i)$ 进行解码(处理块 1005),使用变换解码器 913 和 MDPM 908 对当前块 $c(i)$ 的运动矢量 $mv(i)$ 进行解码(处理块 1006),使用 $mv(i)$ 和帧存储器中的参考帧来对当前块 $c(i)$ 进行运动补偿,以确定块预测值 $u(i)$ (处理块 1007)。在一个实施例中,解码的运动矢量 $mv(i)$ 由 MCPM 908 用于从存储在帧存储器 909 中的参考帧中获取参考块(预测值) $u(i)$ 。

[0230] 接下来,处理逻辑根据下面的方程来重构当前块(处理块 1008):

[0231] $rc(i) = u(i) + e(i)$

[0232] 换言之,通过将预测误差 $e(i)$ 与预测值采样 $u(i)$ 相加来重构当前块 $rc(i)$ 。

[0233] 其后,处理逻辑将重构的数据写入存储器(例如存储器 905)中(处理块 1009)。即,将与重构的块 $rc(i)$ 相对应的数据写入存储器。处理逻辑然后通过使 i 递增 1,移动到被解码的当前帧中的下一个块位置(处理块 1010)。

[0234] 处理逻辑然后测试索引变量 i 是否小于作为被处理的当前帧中块数目的 N (处理块 1011)。如果是,则处理逻辑跳转到处理块 1002,处理从该点处继续进行。按照这种方式,针对解码的当前帧中的所有块,重复上述过程。如果否,则开关 941 闭合,并且处理逻辑使用噪声滤波模块(例如噪声滤波模块 906),对存储器(例如存储器 905)中的帧数据进行滤波(处理块 1012)。在一个实施例中,由环路滤波模块读取存储在存储器模块 905 中的解码的帧数据,环路滤波模块执行与针对视频编码器中的环路滤波模块 125 所述的过程相同的过程。然后,处理逻辑将滤波后的数据写入帧存储器(例如帧存储器 909)(处理块 1013)。

[0235] 其后,该过程结束。

[0236] 因此,这里所述的方法和设备用于确定视频帧数据滤波,与现有技术方案相比,所述视频帧数据滤波产生极好的客观和主观(图像质量)性能。

[0237] 示例的计算机系统

[0238] 图 11 是可执行这里所述的一个或多个操作的示范性计算机系统的框图。参考图 11,计算机系统 1100 可包括示范性的客户端或服务器计算机系统。计算机系统 1100 包括用于传送信息的通信机制或总线 1111、以及用于处理信息的与总线 1111 相连的处理器 1112。处理器 1112 包括微处理器,但是并不局限于微处理器,例如可以是 Pentium™、PowerPC™、Alpha™ 等。

[0239] 系统 1100 还包括用于存储要由处理器 1112 执行的指令和信息的与总线 1111 耦合的随机访问存储器(RAM)或其它动态存储设备 1104(称为主存储器)。主存储器 1104 还可以用于在处理器 1112 执行指令期间存储暂时变量或其它中间信息。

[0240] 计算机系统 1100 还可包括用于存储处理器 1112 的静态信息和指令的与总线 1111 耦合的只读存储器(ROM)和/或其它静态存储设备 1106、以及数据存储设备 1107,例如磁盘或光盘及其相应的盘驱动器。数据存储设备 1107 与总线 1111 耦合,用于存储信息和指令。

[0241] 计算机系统 1100 还可与诸如阴极射线管(CRT)或液晶显示器(LCD)之类的显示设备 1121 耦合,并与总线 1111 耦合,以便向计算机用户显示信息。包括字母数字和其它键的字母输入设备 1122 也可与总线 1111 耦合,用于向处理器 1112 传送信息和命令选择。附加的用户输入设备是光标控制器 1123,例如鼠标、轨迹球、触摸板、手写笔或光标方向键,与总线 1111 耦合,以便向处理器 1112 传送方向信息和命令选择,并用于控制显示器 1121 上

的光标运动。

[0242] 可与总线 1111 耦合的另一设备是硬拷贝设备 1124, 硬拷贝设备 1124 可用于在诸如纸、膜或类似介质上标记信息。可与总线 1111 耦合的另一设备是有线 / 无线通信工具 1125, 用于与电话或手持掌上设备进行通信。

[0243] 注意, 可在本发明中使用系统 1100 的任意或所有组件以及有关硬件。然而, 可以认识到, 计算机系统的其它配置可包括这些设备中的一些或所有。

[0244] 毋庸置疑, 本发明的多个替换和修改对于本领域技术人员而言, 在阅读了上述说明之后是显而易见的, 因此要理解, 作为示例示出并描述的任何特定实施例绝不意味着限制。因此, 对各个实施例的细节的参考并不意欲限制权利要求的范围, 权利要求本身仅叙述了对于本发明必要的特征。

[0245] 本申请要求 2005 年 5 月 20 日递交的题为 “Method and Apparatus For Noise Filtering in Video Coding” 的相应临时专利申请 No. 60/683, 240 的优先权, 将其内容通过引用结合于此。

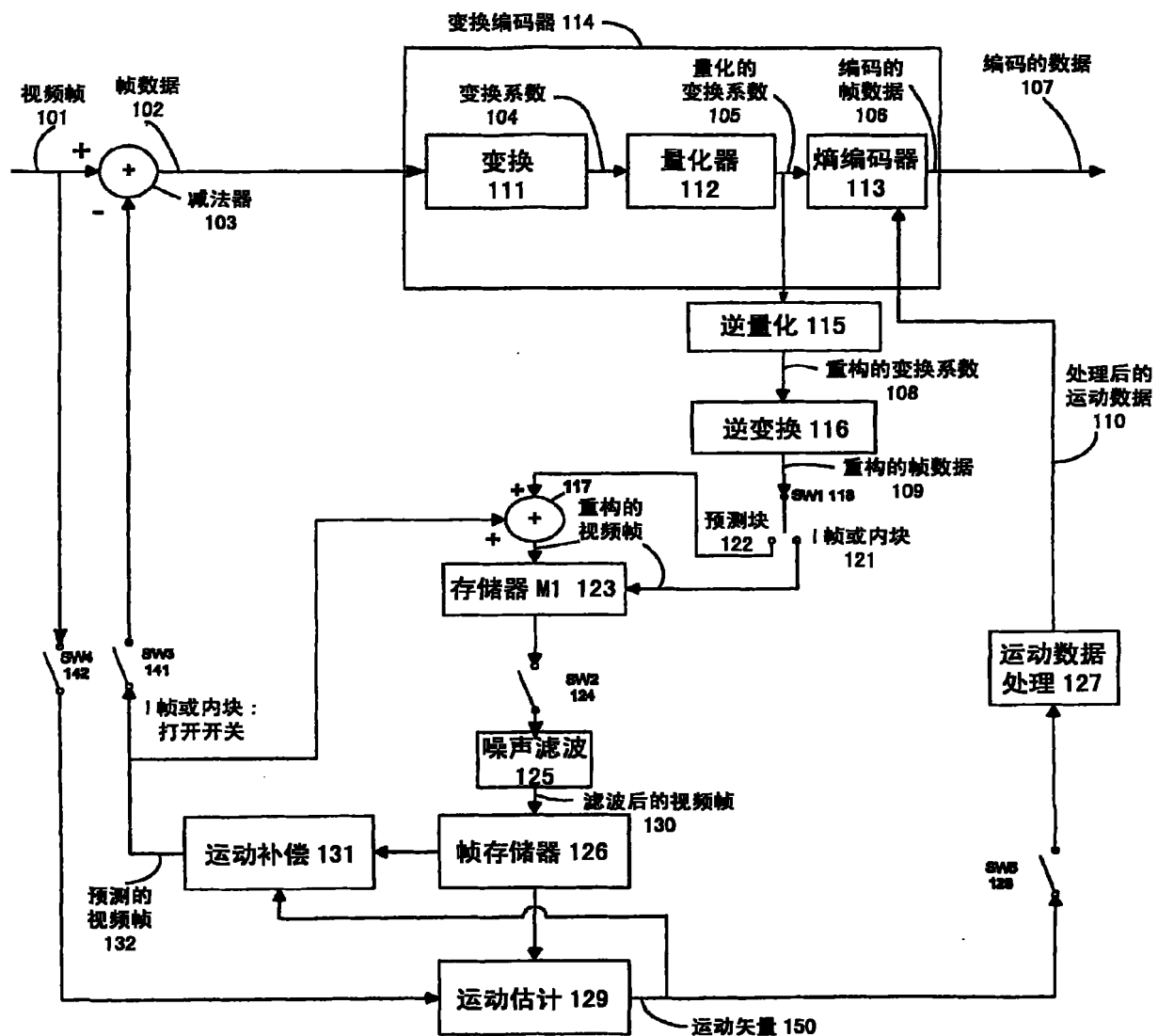


图 1

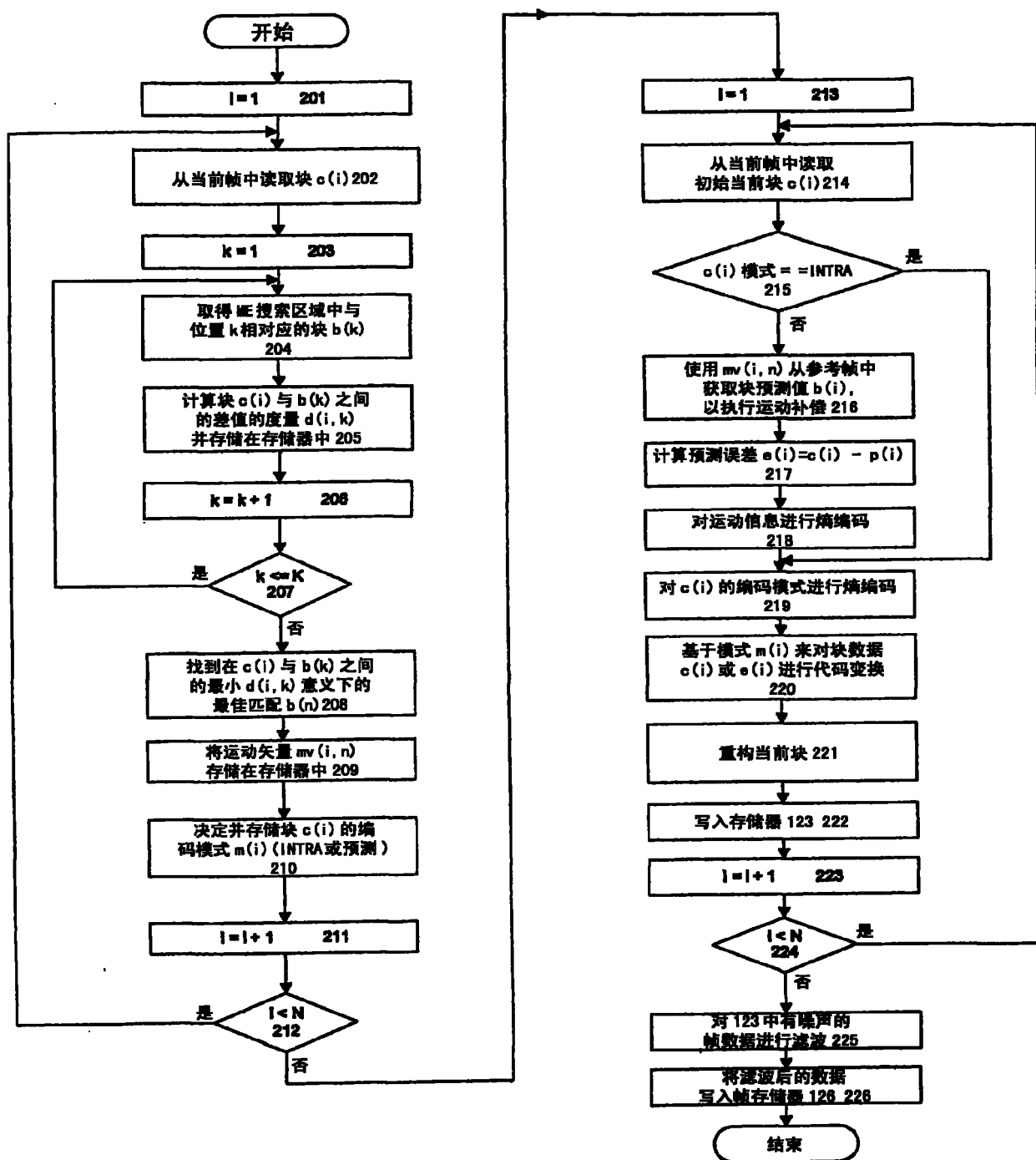


图 2

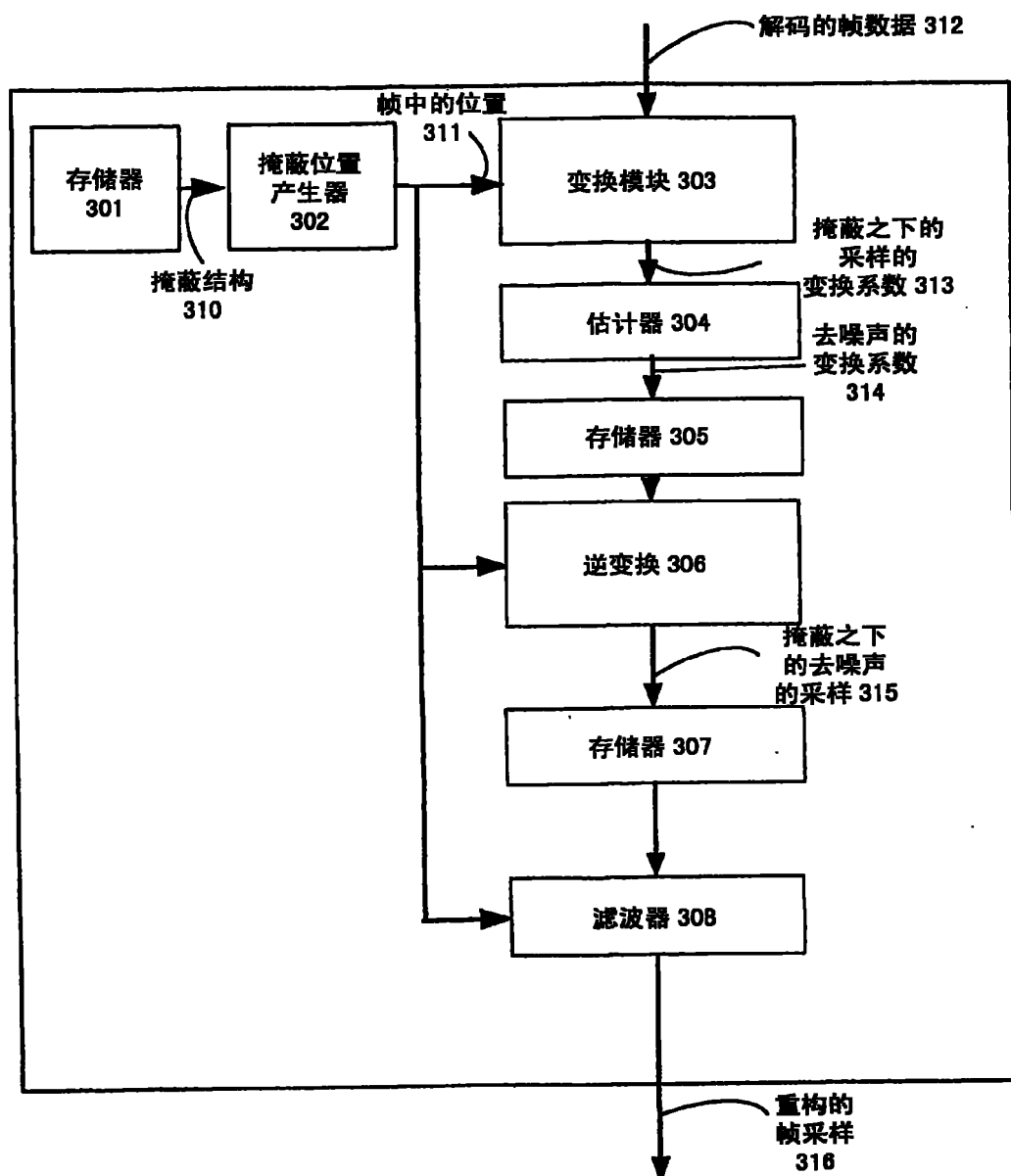


图 3

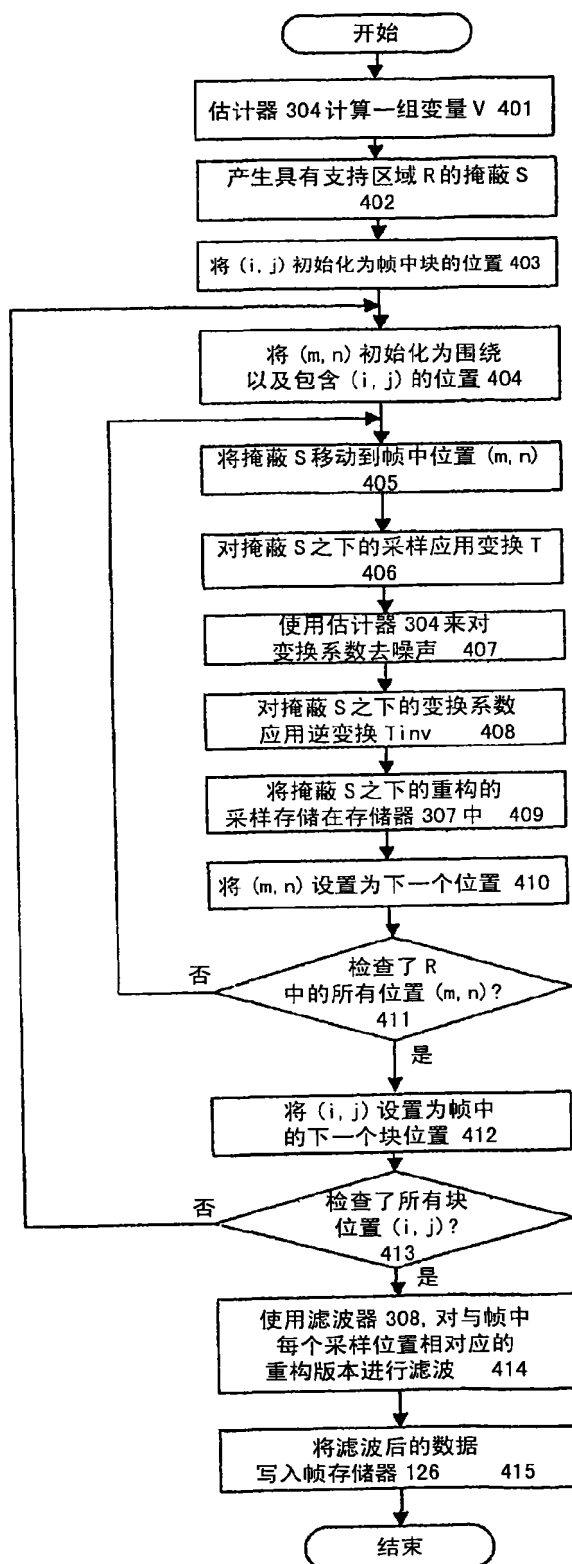


图 4

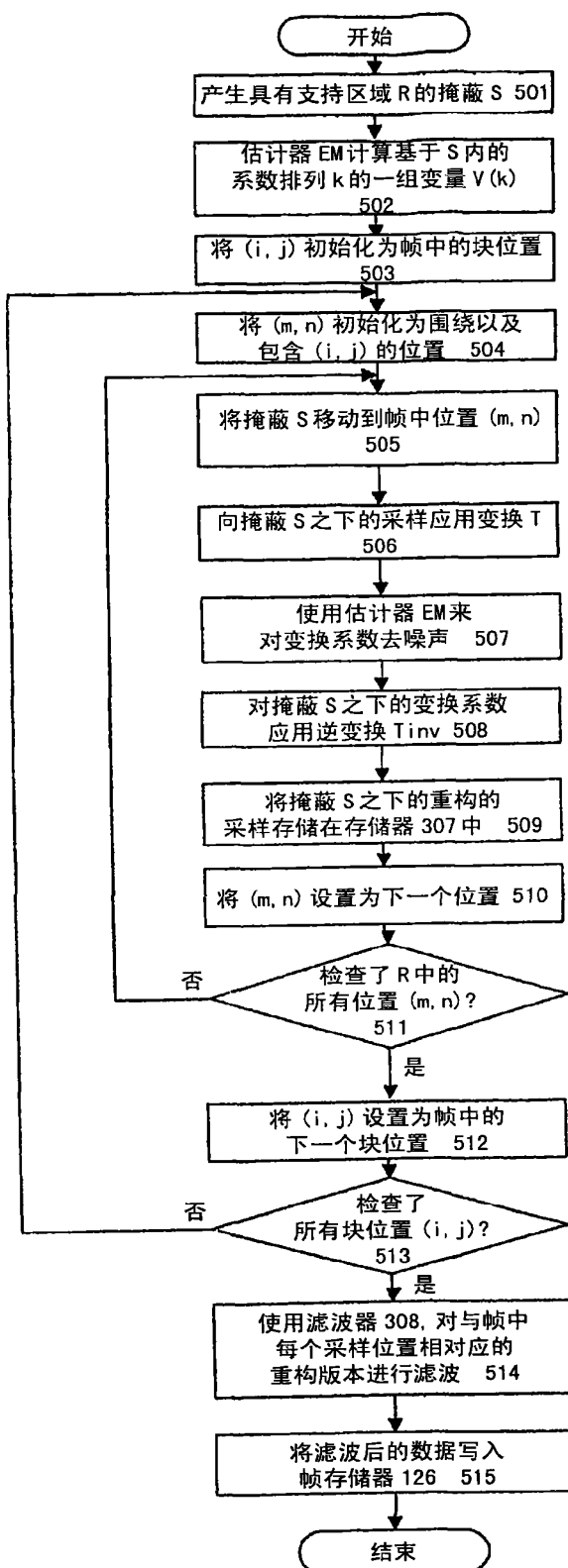


图 5

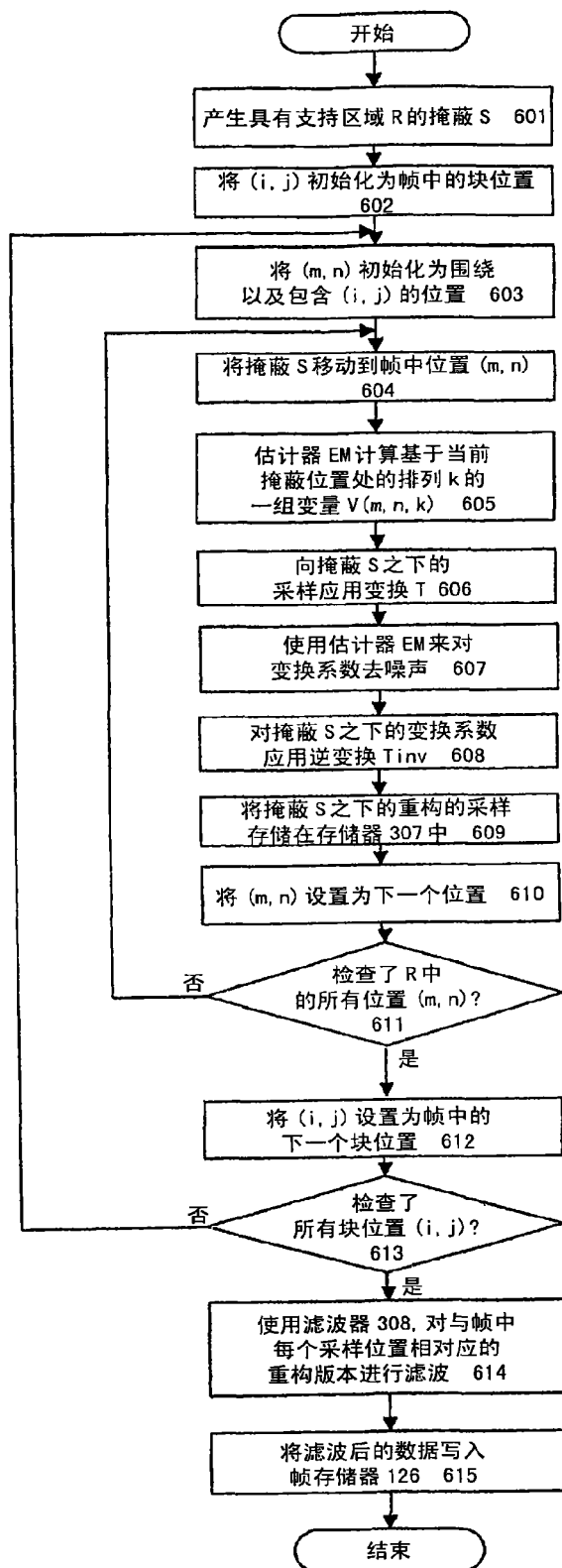


图 6

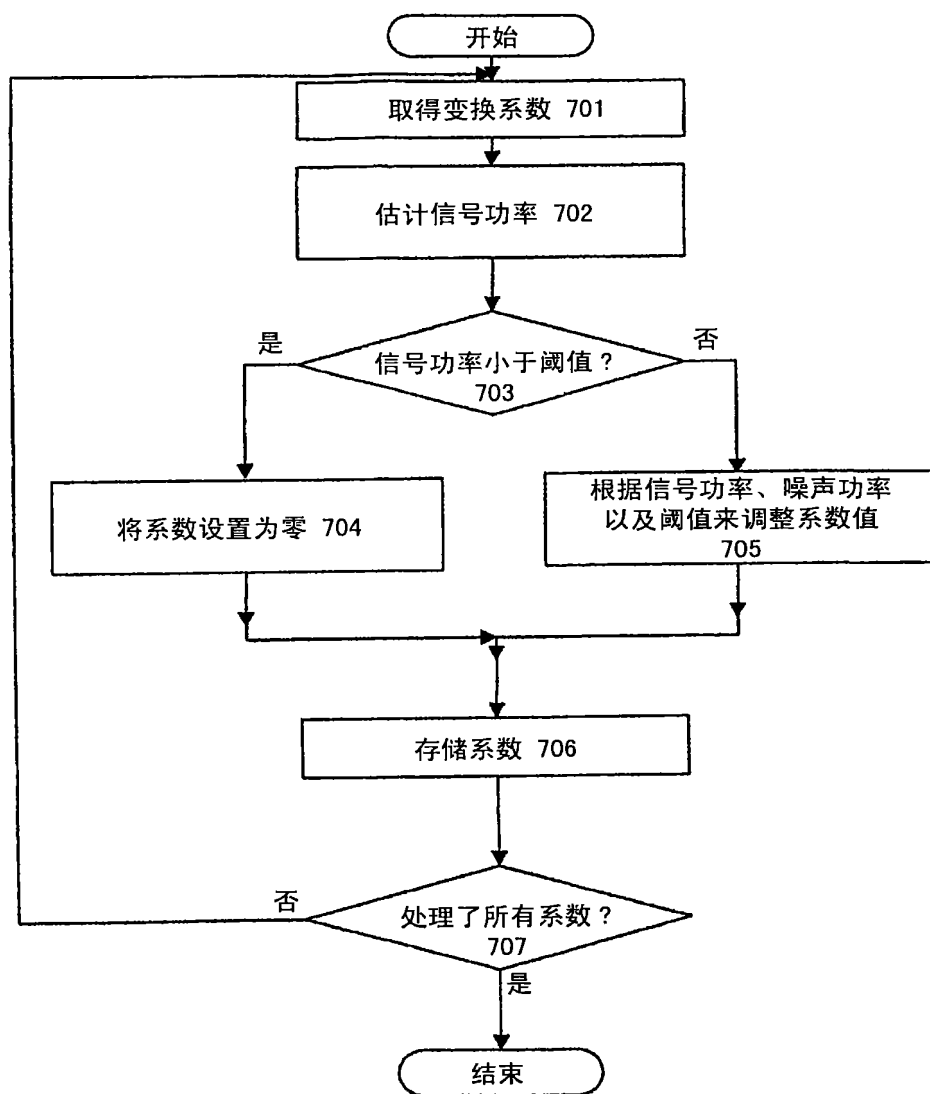


图 7

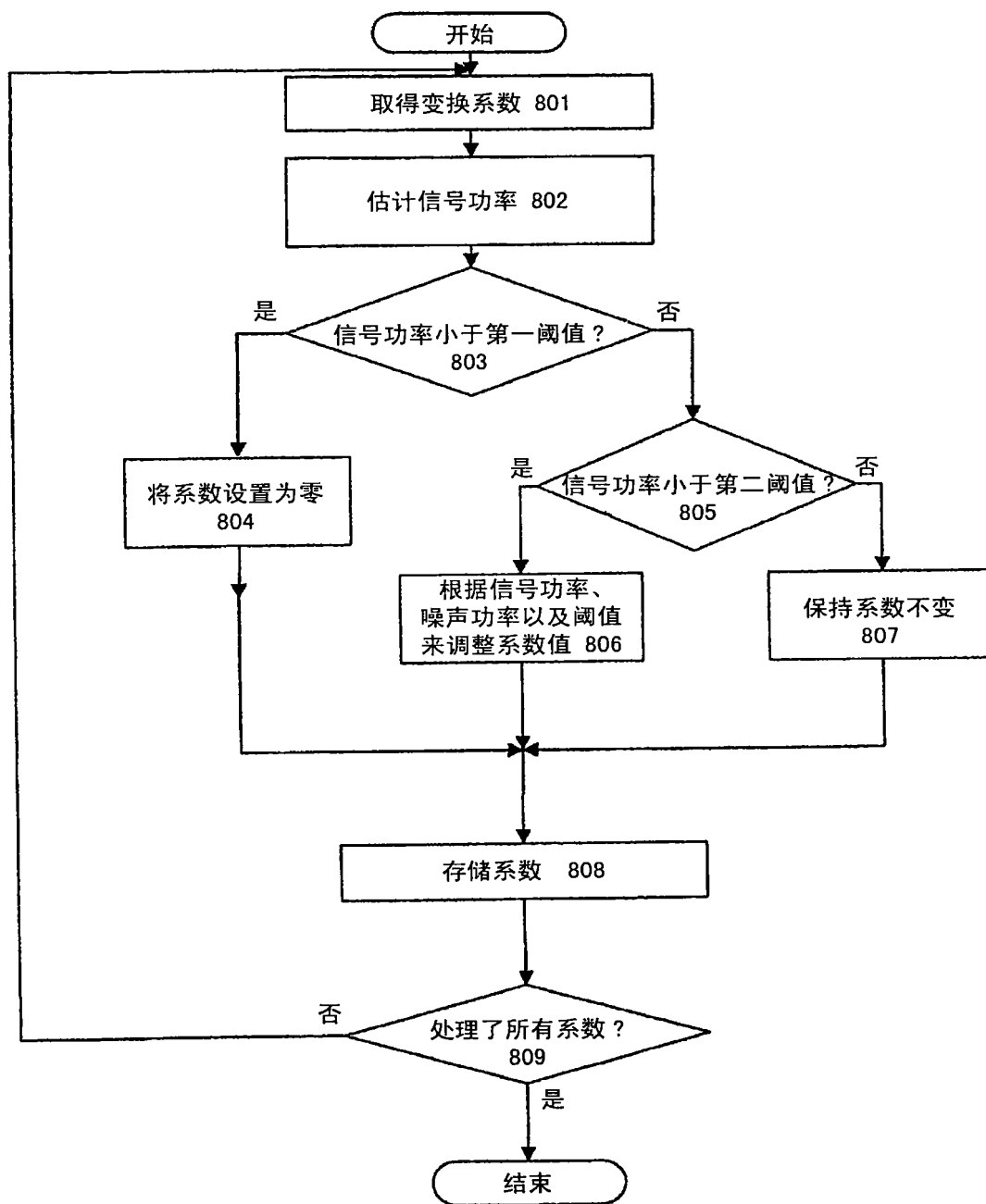


图 8

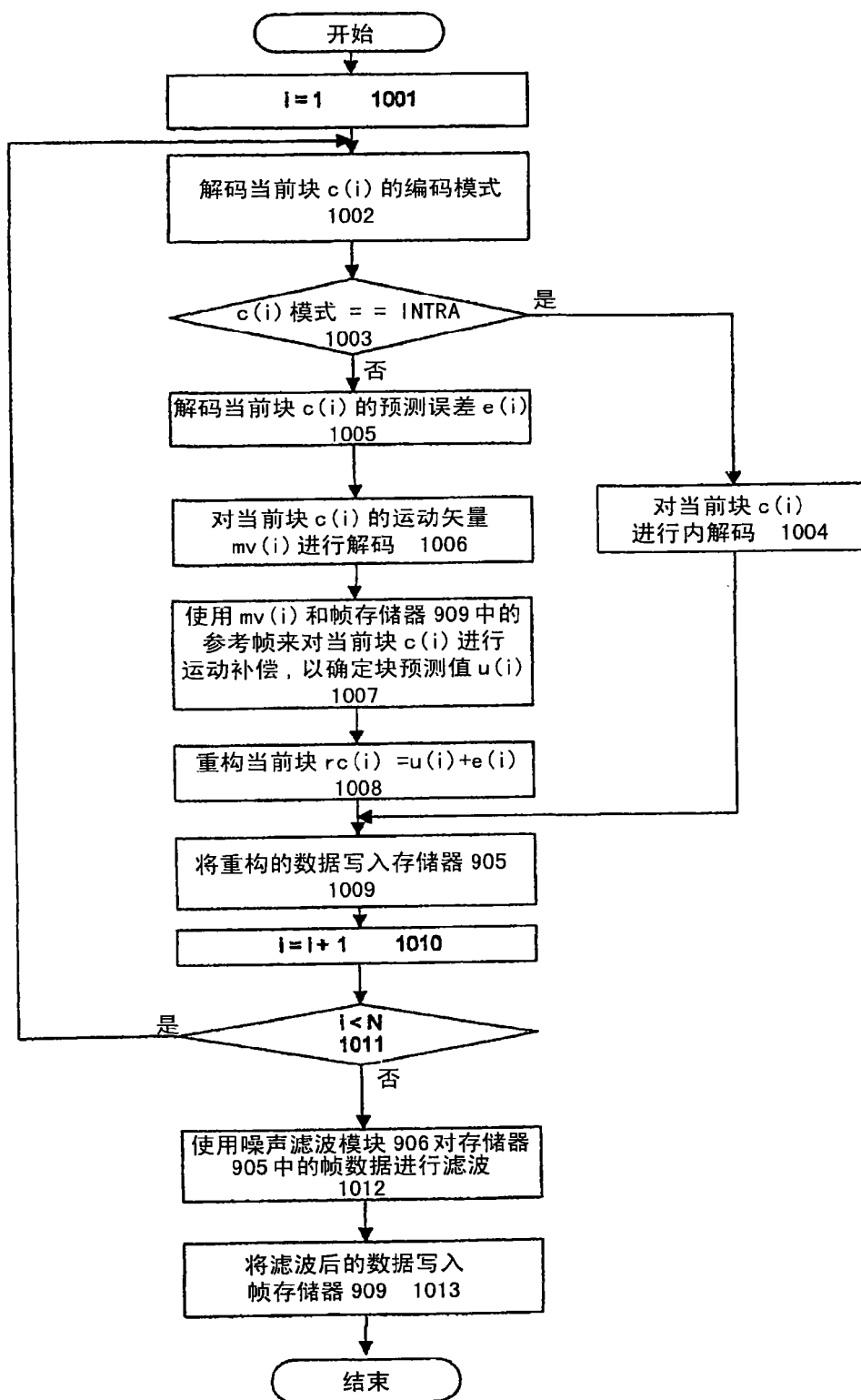


图 10

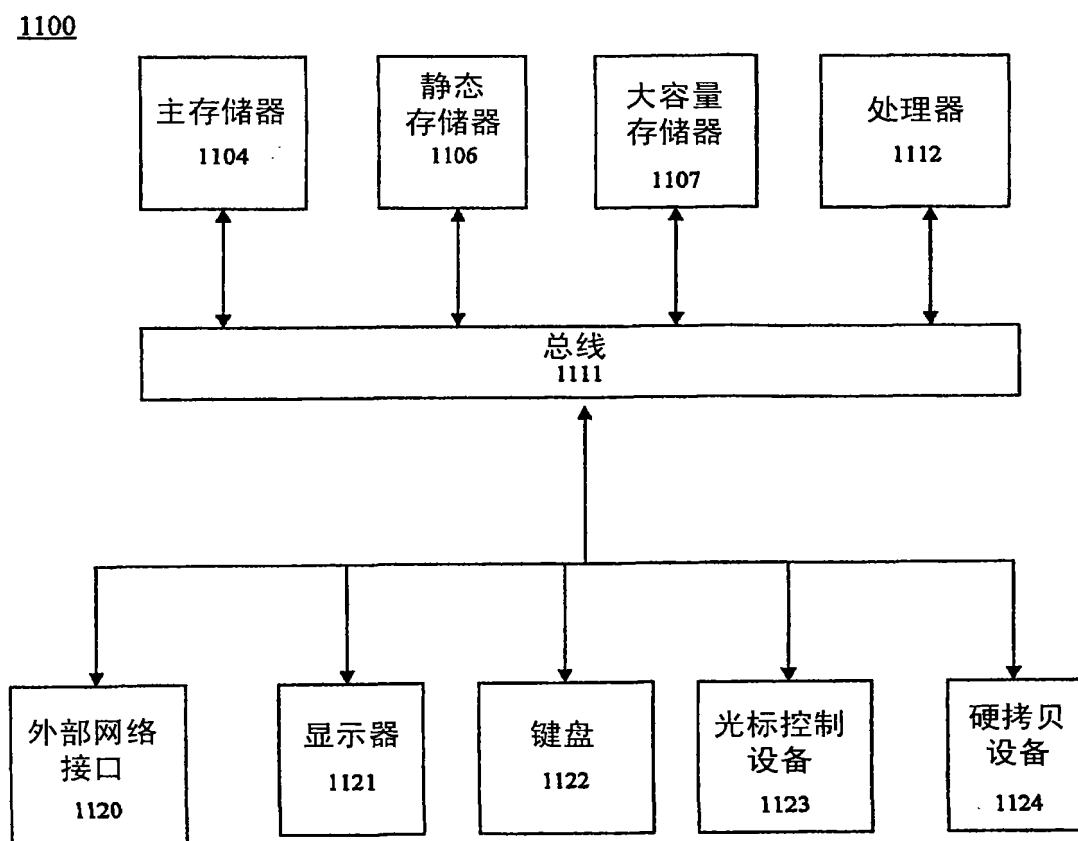


图 11