



(12) 发明专利

(10) 授权公告号 CN 1615653 B

(45) 授权公告日 2010.05.26

(21) 申请号 02827338.9

(22) 申请日 2002.11.07

(30) 优先权数据

09/990,976 2001.11.21 US

(85) PCT申请进入国家阶段日

2004.07.20

(86) PCT申请的申请数据

PCT/CA2002/001713 2002.11.07

(87) PCT申请的公布数据

W02003/047265 EN 2003.06.05

(73) 专利权人 VIXS 系统公司

地址 加拿大安大略

(72) 发明人 尹德拉·莱克索诺

(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 董莘

(51) Int. Cl.

H04N 7/26 (2006.01)

H04N 7/50 (2006.01)

G06T 3/40 (2006.01)

(56) 对比文件

AKAHASHIKETAL: "MOTIONVECTORSYNTNEHSISALGORITMFORMPAKAHASHIKETAL: "MOTIONVECTORSYNTNEHSISALGORITMFORMPEG2-TO-MPEG4TRANSCODER" PROCEEDINGSOFTHESPIE, SPIE, BELLINGHAM, VA, US, XP008000078, PAGES873-877, FIGURES2-5. 2001, 第 873-877 页, 附图 2-5.

审查员 李靖

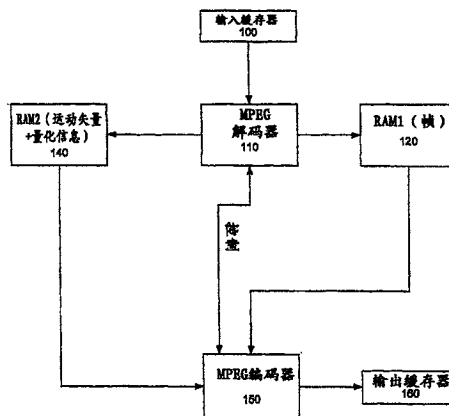
权利要求书 2 页 说明书 21 页 附图 23 页

(54) 发明名称

多通道视频译码的系统和方法

(57) 摘要

公开一种处理通过媒体通道接收的视频数据的系统和方法。所述方法包括利用索引表引用存储的视频数据,包括运动矢量,从而变码器的编码器和解码器能够高效访问存储的视频数据,解码器产生源像素图像,编码器根据存储的运动矢量数据,产生估计的运动矢量。



1. 一种方法,包括下述步骤:
存储多个源宏块的每个源宏块的源宏块信息;
确定索引表,所述索引表基于目的图像信息和源图像信息,其中所述索引表包括多个条目,每个条目包括与存储对应源宏块的源宏块信息的存储器位置相关联的识别符;和
存储所述索引表。
2. 按照权利要求 1 所述的方法,其中所述目的图像信息包括视频目的分辨率,并且其中所述源图像信息包括视频源分辨率。
3. 按照权利要求 1 或 2 所述的方法,还包括下述步骤:
确定将由视频变码器的一部分处理的数据指令分组,其中所述数据指令分组用以识别所述索引表的位置。
4. 按照权利要求 3 所述的方法,其中所述视频变码器的一部分是视频解码器部分和/或视频编码器部分。
5. 按照权利要求 1 或 2 所述的方法,其中其中所述索引表的每个条目具有相同大小。
6. 按照权利要求 1 或 2 所述的方法,其中所述索引表包含用以指示与目的宏块相关的索引表的一部分的宏块指示符的尾部。
7. 按照权利要求 6 所述的方法,其中所述宏块指示符的尾部被存储为索引表条目的字段内的数值。
8. 按照权利要求 1 或 2 所述的方法,其中所述宏块信息包括运动矢量和量化信息。
9. 按照权利要求 8 所述的方法,其中所述索引表用于基于在索引表中识别的源宏块信息产生估计的目的运动矢量。
10. 按照权利要求 1 或 2 所述的方法,包括以下步骤:
产生第一目的宏块的第一估计宏块信息,其中所述第一目的宏块信息至少使用多个源宏块的宏块信息的一部分,所述第一目的宏块可相对于源宏块而被按比例缩小。
11. 按照权利要求 10 所述的方法,包括以下步骤:
存取第二索引表;
在视频解码器,按照第二顺序存取第一批多个宏块信息,以产生第二解码后的图像,其中所述第二顺序基于所述第二索引表,所述第一批多个宏块信息与源宏块相关;
存取第二批多个源宏块信息,以产生第二估计目的宏块信息。
12. 按照权利要求 11 所述的方法,包括以下步骤:
根据所述第一估计目的矢量产生第一宏块,根据所述第二估计目的矢量产生第二宏块,所述第一和第二宏块将被实时地同步显示。
13. 一种系统,包括:
接收源视频数据的输入端口;
与所述输入端口耦接、用以确定对应于所接收的源视频数据的宏块信息数据的控制器部分,其中所述宏块信息包含运动矢量和量化信息;
与所述控制器部分耦接、用以存储对应于所述源视频数据的多个源宏块信息的第一存储器控制部分;和
被配置成用于接收目的图像信息并产生索引表的索引表发生器,其中所述索引表识别将用于产生目的图像的多个源宏块信息的第一部分,所述索引表基于目的图像信息。

14. 按照权利要求 13 所述的方法,其中所述目的图像信息包括目的图像的尺寸指示符。

15. 按照权利要求 13 或 14 所述的方法,其中产生目的图像的步骤包括产生目的源矢量。

16. 按照权利要求 13 或 14 所述的系统,还包括:
耦接地用于根据索引表条目,检索源宏块信息的第二存储器控制部分;
与所述第二存储器控制部分耦接,以根据所检索的源宏块信息产生目的矢量的编码器部分。

17. 按照权利要求 13 或 14 所述的系统,其中利用通用处理器核心实现所述索引表发生器。

多通道视频译码的系统和方法

技术领域

[0001] 本发明涉及处理视频流的系统,更具体地说,涉及存储在视频处理中使用的数据的系统和方法。

背景技术

[0002] 和用于能够利用非常简单的电路,对接收的模拟媒体数据解码的模拟广播系统的模拟调谐器不同,数字媒体数据,例如运动视频,通常需要处理器强化的操作,以便重构数字媒体。如果媒体流具有和媒体流相关的大量数据,例如支持高清晰度电视(HDTV) $1920 \times 1080i$ 格式所需的数据传输速率,那么对多个同时的数字媒体流解码的成本和工作量非常高。目前借助利用帧间内容的连续性产生高度压缩的数据的压缩方案解决这个问题。运动图像专家组(MPEG)已提出包括将运动估计用于帧间的多块图像,以便实现压缩在内的各种方法。

[0003] 压缩视频数据的步骤对处理器和存储带宽要求很高。运动估计(一个压缩步骤)需要大量的计算工作,这么大量的计算工作能够用完大量的可用带宽。在常见的运动估计方法中,一帧图像数据首先被分成多个片段或块。随后,对照另一帧或多帧中的一个或多个片段或块组,比较当前帧图像的一个片段或块组。每个交替帧中的最佳片段或块组可能不在和当前帧相同的位置上。在每个交替帧和当前帧之间,该位置经常不同。片段相对于前一帧的位置被表示成运动矢量。通常使用对处理器和存储带宽要求很高的复杂搜索算法构造每个运动矢量,这种搜索算法必须考虑多种片段组合。

[0004] 在极高的分辨率下,例如 $1920 \times 1080i$ 格式下,这种压缩流的数据速率很高。高数据速率至少造成三个问题。首先,在任意时间长度内记录或存储这样的流需要大量的存储器,成本高得惊人。其次,可用于观看这种流的许多显示设备不能显示这种高分辨率的数据流。第三,在存在具有多个观看或接收设备的数据网络的情况下(这种网络一般具有固定的带宽或容量),这种网络不能同时支持多个观看设备。另外,建立帧的每个片段或宏块通常需要多个运动矢量。这进一步增加了处理和带宽问题。

[0005] 因此,需要一种处理视频流的改进系统和方法。

发明内容

[0006] 根据本发明的一个方面,提供了一种方法,包括下述步骤:

[0007] 存取第一索引表;

[0008] 在视频解码器中按照第一顺序存取第一批多个宏块信息,以产生第一解码后的图像,其中所述第一顺序基于所述第一索引表,所述第一批多个宏块信息与源宏块相关;

[0009] 存取所述第一批多个宏块信息,以产生第一估计的目的运动矢量;和

[0010] 其中所述宏块信息包含运动矢量和量化信息。

[0011] 根据本发明的另一个方面,提供了一种方法,包括下述步骤:

[0012] 存储第一批多个源宏块的每个源宏块的视频源宏块信息;

[0013] 确定具有多个条目的索引表,所述索引表基于视频源分辨率和视频目的分辨率,其中所述索引表的对应条目引用每个宏块的每个源宏块信息的位置;和

[0014] 存储所述索引表。

[0015] 根据本发明的再一个方面,提供了一种系统,包括:

[0016] 接收源视频数据的第一输入端口;

[0017] 与所述第一输入端口耦接、用以确定对应于所接收的源视频数据的宏块信息数据的控制器部分,其中所述宏块信息包含运动矢量和量化信息;

[0018] 与所述控制器部分耦接、用以存储对应于所述源视频数据的多个源宏块信息的第一存储器控制部分;和

[0019] 耦接地用于接收目的图像的尺寸指示符,并产生索引表的索引表发生器,其中所述索引表识别将用于产生第一目的源矢量的多个源宏块信息的第一部分,所述索引表基于目的图像的尺寸指示符。

附图说明

[0020] 图 1 是图解说明根据本发明的至少一个实施例的 MPEG 变码器处理器的方框图;

[0021] 图 2 是图解说明根据本发明的至少一个实施例的 MPEG 变码器处理器的操作的流程图;

[0022] 图 3 是图解说明根据本发明的至少一个实施例的压缩视频译码系统的方框图;

[0023] 图 4 是图解说明根据本发明的至少一个实施例的矢量处理器的方框图;

[0024] 图 5 是图解说明根据本发明的至少一个实施例的压缩视频译码处理器的操作的流程图;

[0025] 图 6 图解说明使用本发明的至少一个实施例公开的译码系统的方法;

[0026] 图 7 图解说明根据本发明一个实施例的数据指令分组;

[0027] 图 8 更详细地图解说明图 3 的一部分;

[0028] 图 9 更详细地图解说明图 3 的一部分;

[0029] 图 10 图解说明源宏块和目的地宏块与图像帧的关系;

[0030] 图 11 图解说明在本发明的一个具体实施例中,存储在线性地址空间中的数据指令分组;

[0031] 图 12 图解说明存储在存储设备中的数据分组信息及其存储在缩放缓存器中的对应宏块信息;

[0032] 图 13 图解说明根据本发明的另一实施例,存储在线性地址空间中的数据指令分组信息;

[0033] 图 14 图解说明存储在设备存储器中的数据指令分组信息和数据指令分组指针及其在缩放缓存器中的对应宏块信息;

[0034] 图 15-19 图解说明根据本发明的具体方法;

[0035] 图 20 更详细地图解说明位操作器的具体实施例;

[0036] 图 21 更详细地图解说明位操作器的线路缓存器和寄存器的具体实施例;

[0037] 图 22 图解说明代表通过位操作器的功能数据流的方框图;

[0038] 图 23 以方框图的形式图解说明产生位偏移指针的具体方法或系统;

- [0039] 图 24 以流程图的形式图解说明根据本发明一个具体实施例的方法；
- [0040] 图 25 以方框图的形式，图解说明覆盖在一帧源宏块上面的一帧目的地宏块的表现；
- [0041] 图 26 以方框图的形式，图解说明在实现本发明的具体实施例中使用的存储器的表现；
- [0042] 图 27 以表格形式，图解说明根据一个实施例，存储在图 26 的存储器的一部分中的数据的表现；
- [0043] 图 28 以流程图的形式，图解说明根据本发明一个具体实施例的方法；
- [0044] 图 29 以方框图的形式，图解说明根据本发明的系统。

具体实施方式

[0045] 根据本发明的至少一个实施例，提供用于译码的系统。该系统包括接收数字视频数据并提供解析视频数据的第一通用处理器。系统还包括与第一处理器耦接，以便获得解析视频数据的第二处理器，第二处理器包括一个视频变码器。根据在图 25-29 中图解说明的本发明的一个具体实施例，运动矢量和量化信息存储在存储器中，并利用索引表存取，以便减少系统使用的存储带宽。本发明的优点在于媒体通道可被有效接收和译码。另一优点在于更有效地支持视频通道的实时重放。

[0046] 图 1-5 图解说明对多个媒体通道译码的系统和方法。该系统包括解析具有一个或多个媒体通道的媒体数据流的第一处理器，和解压缩，缩放随后压缩解析的媒体通道的矢量处理器。在一个实施例中，解析的媒体通道被打包成解码器指令分组，并通过利用排序器，被传送给矢量处理器。矢量处理器对解码器指令分组解压缩，缩放从所述分组产生的宏块，随后压缩缩放后的结果。从而，缩放并且压缩后的输出具有较少的和媒体通道相关的数据，便于更快和 / 或更高效的存储或传输。

[0047] 现在参见图 1，根据本发明的至少一个实施例，图解说明了视频处理系统的高级功能方框图。视频处理系统 100 包括接收代表将提供给 MPEG 解码器 110 的视频图像的数据的输入缓存器 105。MPEG 解码器 110 在存储器 (RAM1 120) 中产生视频图像或者视频图像解压缩帧的随意按比例缩小的表现。另外，MPEG 解码器 110 可从解压缩过程中，获得运动矢量和量化信息。我们将每个宏块的运动矢量和量化信息的集合称为宏块信息。宏块信息存储在存储器 (RAM2 140) 中。注意通常不利用标准 MPEG 解码器，将宏块信息存储在存储器中。另外注意在其它实施例中，存储器 120 和 140 可包括动态随机存取存储器，静态动态随机存取存储器，硬盘驱动器存储设备等。

[0048] 在一个实施例中，MPEG 编码器 150 执行正常 MPEG 编码的许多步骤，但是通过检索先前存储在 RAM 2 140 中的运动矢量，避免计算量大的运动估计步骤。通过检索该组环绕 (surrounding) 运动矢量，并建立一组新的运动矢量，MPEG 编码器 150 能够避免在常规的运动估计中需要的成本极高的搜索。输出缓存器 160 包括在将数据位输出给最终目的地，例如存储器或者耦接的接收这种数据的设备的任意输出端口之前，累积数据位的位存储桶。

[0049] 另外注意在每个图像数据内，每个宏块的量化信息可随着每个宏块的内容而变化。使用更大的量化值会产生进一步压缩的数据，导致较小的数据流，但是也会引起每个宏块中细节的丢失。在适当编码源 MPEG2 流 (接收数据) 中，来自每个宏块的量化信息将被

优化,并且根据本发明,我们可重复使用该量化信息,如同在与本申请同日提出的同时待审的专利申请“METHOD AND SYSTEM FOR RATECONTROL DURING VIDEO TRANSCODING”中公开的那样(该申请作为参考包含于此)。事实上,通过重复使用该量化信息,能够获得更高质量的图像,从而允许宏块的更大量化,所述宏块的更大量化担负得起与过度压缩相关的细节的损失,同时保持应包含更主要细节的宏块细节。

[0050] 参见图 2,根据本发明的至少一个实施例,图解说明了译码方法。参考图 2 中图解说明的具体实施例,附图标记为 310-395 的单元表示变码器的解码器部分的功能,单元 410-495 表示变码器的编码器部分的功能。注意该例子假定 $1/2 \times 1/2$ 的图像缩小比例。在 MPEG 术语学中,宏块是单个像素的 16×16 矩阵。在 MPEG 术语学中,块是单个像素的 8×8 矩阵。当以 $1/2 \times 1/2$ 按比例缩小时,假定一组 2×2 宏块被转换形成单个宏块。一般借助在以较小的位图呈现最终结果的同时,保持尽可能多的原始图像的内容的努力,实现 $1/2 \times 1/2$ 比例缩小操作。众所周知,比例缩小意味着按照某一方式组合一组像素,产生由较少的像素组成的另一组。对于 $1/2 \times 1/2$ 的比例缩小来说,存在几种选择。例如,一种可能的实现是,按照预定方式混合像素。但是,本领域的技术人员会理解存在多个途径混合像素,从而获得相同的结果,或者可能在不进行混合的情况下实现相同的缩放。

[0051] 在步骤 310 中,检索正被解码的宏块并进行位解析。在步骤 320 中,利用霍夫曼解码器进行可变长度解码,所述可变长度解码可以是运行等级或游程长度解码,以便检索和特定块相关的信息。检索的信息包含一系列的运行等级序列,每个运行等级序列代表一个具有由游程含蓄定义的一连串连续零值的 8×8 备用矩阵(称为块)。当在 MPEG 范围内称为运行等级时,游程指的不是等级值的反复行程,而是 0 的连续行程。就 MPEG 来说,按照交错(zigzag)方式建立块。

[0052] 解码之后,该过程在步骤 330 对产生的块去量化(dequantize),这涉及用矩阵中的元素去乘块中的每个元素。本领域中众所周知,去量化是逆量化,可在解码过程的开始,检索或确定固定常数的矩阵,当流被解码时很少发生变化。稀疏矩阵的每个元素被乘以来自该常数矩阵中对应项的常数值。为了理解逆量化过程的重要性,首先应明白的是在压缩过程中,在一个实施例中,块经历离散余弦变换(DCT)步骤,将图像块转换到频域。在图像块的这种表现中,仍然能够无任何损失地检索初始块(up to 源于数字表现局限性的算术舍入)。

[0053] 在频域中,数据块具有有趣的性质。对人眼来说,图像的表现的主要决定因素主要由位于左上角的块(矩阵)的项(始于矩阵的索引 $[0,0]$)决定。对人眼来说,对块右下角的项的变化对重构块的视觉影响较小。编码过程中的量化的目的是利用这种性质,试图将接近于 0,并且在位置上更接近于右下角的项作为 0 来处理,同时尽可能多地保持左上角中的信息。

[0054] 在块被去量化之后,在步骤 340 中对数据块应用反向离散余弦变换(IDCT)方法,以便获得原始形式的块。宏块判定步骤 350 结束之后,循环进行步骤 205-220,直到获得完整的宏块为止。对于 MPEG 来说,该宏块通常由 4(2×2) 块 Y(照度或亮度)信息和 1 块 Cr 及 1 块 Cb 组成。当获得完整的宏块时,过程转到测试解码器运动矢量步骤 360,确定或识别宏块的运动矢量。如果不存在特定宏块的任何运动矢量,那么在一个实施例中,在步骤 390 中,依据某一比例因子,例如 1、2、4 或 8,宏块被比例缩小,并被写出。

[0055] 在存在一组运动矢量的情况下,在步骤 370 中,运动矢量将被存储到存储区,例如存储器 140(图 1)中,所述存储区存储用于建立该帧的所有初始运动矢量。在步骤 380 中,执行运动补偿,建立新的宏块。随后在步骤 390 中被按比例缩小并输出。

[0056] 在步骤 395 中,如果帧已完成,那么在步骤 410,过程初始化帧编码器,这将开始在步骤 420 中对宏块编码。如果当前宏块没有任何运动矢量(在步骤 430 确定),那么在步骤 470 中,从在解码过程中产生的比例缩小和解压缩帧读取该宏块,在步骤 480 中,该宏块中的每块经历离散余弦变换。如果当前的宏块具有运动矢量(在步骤 430 中确定),那么在步骤 440 中,从存储器检索四组相邻的运动矢量,并在步骤 450 和 460 中,将其用于建立初始图像帧。本例中,注意使用 $1/2 \times 1/2$ 的缩放比例。会需要关于其它比例因子的更多运动矢量的检索。例如,如果依据 $1/3 \times 1/3$ 比例缩放,那么理应使用 9 个运动矢量。如果依据 $2/5 \times 2/5$ 比例缩放,那么会使用 4 ~ 9 个运动矢量,取决于如何产生结果的运动矢量。

[0057] 在步骤 450 中,可按照多种途径建立新的运动矢量。在一种方法中,可对来自四组运动矢量中每一组的矢量的各个分量应用 $1/2$ 的简单平均按模计算。在一个备选实施例中,利用破坏联系的任意方法,可选择每组 k 个运动矢量中最频繁发生的运动矢量 (ΔX_k , ΔY_k)。破坏联系的一种可能方法是选择最接近于左上侧的运动矢量的元素。

[0058] 借助在步骤 450 中建立的新的运动矢量,过程转到从存储的解压缩图像帧读取宏块,随后建立包含应用逆运动补偿步骤的结果的 delta 帧,从而在步骤 280 中获得该 delta 帧。此时,delta 宏块被发送给在步骤 485 中对所有块执行离散余弦变换(DCT)的单元(舍入每个元素被矩阵元素的整数除法)。随后在步骤 488 中,所得到的每个块的量化矩阵表现被变长编码。继续该过程,直到在步骤 495 中检测到帧的编码器末端为止,用信号通知解码器开始处理下一帧。注意利用运动矢量的双重缓存,并对其它命令流排队,编码器和解码器步骤可并行进行。

[0059] 至少一个实施例的一个特征在于,在存在运动矢量的情况下,在步骤 370 中,运动矢量将被存储到存储用于建立处理帧的所有初始运动矢量的存储区中。另外,通过检索存储的运动矢量并在步骤 450 中,利用简单的处理步骤建立一组新的运动矢量,避免了计算量大,并且成本高的运动估计步骤。通过避免成本高的运动估计步骤,本发明可提供成本效果更高的解决方案,该解决方案基本上实现和从 scratch 搜索运动矢量的系统相同的译码质量。

[0060] 参见图 3,根据本发明的另一实施例,图解说明了压缩视频译码系统。压缩视频译码系统 300(这里称为译码系统 300)可在各种实现中包括部分、全部或者不包含视频处理系统(图 1)。译码系统 300 包括媒体源 301,变码器设备 303,主中央处理器(CPU)305,总线/存储器控制器 307(即,北桥芯片),主动态随机存取存储器(DRAM)309,系统输入/输出(I/O)总线 315,网络控制器 320 和设备存储器 390。

[0061] 在一个实施例中,变码器设备 303 包括处理器 330,霍夫曼解码器 333,输入位存储桶 335,存储器控制器 340,解码器指令分组(DIP)排序器 345,视频处理器 350 和系统直接存储器存取(DMA)总线接口 380。在至少一个实施例中,处理器 330 是通用处理器,例如 RISC 处理器,而 DIP 排序器 345,和视频处理器 350 是被优化以便高效处理特定任务的专用数据处理器。在一个实施例中,视频处理器 350,排序器 345 和处理器 330 被集成到一个公共基体上,例如半导体基体或插件基体上。在不脱离本发明的精神或范围的情况下,根据需

要,译码系统 300 可包括另外的部件。

[0062] 在至少一个实施例中,媒体数据流 310 代表来自一个或多个源 301 的数字媒体数据,每个源可具有一个或多个媒体数据通道。媒体源 301 可包括媒体数据流的任意源,例如 MPEG 播放器,通过网络控制器 320 与译码系统 300 连接的数据服务器等。在一个实施例中,当在系统 303 接收时,媒体数据流采取字节、或字格式。在其它实施例中,媒体数据流 310 可以是由系统 330 组织成字节或适当字大小的位流。一旦采取恰当的字格式,数据被传送给存储器控制器 340。在一个实施例中,媒体数据流 310 被看作存储器控制器 340 的客户。可通过一组共用或专用输入引线,或者通过系统 I/O 总线 315,媒体数据流可被提供给存储器控制器 340。通常,存储器控制器 340 将根据预定的优先化方案,例如循环方案,或者根据将参考图 5 更详细说明的固定优先化,对其每个客户判优。

[0063] 来自媒体数据流 310 的数据存储在设备存储器 390 中。设备存储器 390 可包括快速存储器,随机存取存储器,高速缓存存储器等。一旦数据流被存储在设备 390 中,那么多个客户机,包括处理器 330 和视频处理器 350 能够通过控制器 340 存取存储器 390,并从存储器 390 获得数据流。这允许系统 303 并行处理数据。

[0064] 在一个实施例中,处理器 330 使用输入位存储桶 335 作为集结区,以支持对存储在存储器 390 中的数据的位级存取,以及缓存数据,从而降低处理器 330 和存储在存储器 390 中的数据之间的等待时间。例如,通过使用输入位存储桶 335,处理器 330 可请求可变范围内的任意位或位的倍数。例如,处理器可请求始于地址位置的位 3 的字节或者恰当大小的字。返回的值将具有恰当对准提供给处理器 330 的实际数据的边界的位 3 的值。

[0065] 在一个实施例中,处理器 330 是通用处理器,例如 MIP 处理器,或者专用处理器,例如 ASIC。通用处理器是执行软件规定的一个或多个功能的数据处理器,这里要明白的是软件理应包括固件。在一个实施例中,处理器 330 存取接收的视频数据,并执行纠错;过滤,例如媒体数据流 310 的解析或解码,产生通过媒体数据流 310 接收的所需媒体数据通道的解码器指令分组 (DIP)。DIP 包括将由 DIP 排序器 220 和矢量处理器 230 用于产生新的宏块的具体指令和 / 或控制信息。一旦产生,那么 DIP 可被存储器控制器 340 存储在设备存储器 390 中。在另一实施例中,纠错、过滤或解析功能中的一个或多个功能由设备 303 的另一组件执行。例如,在存储器控制器 340 将媒体数据流 310 存储在设备存储器 390 中之间,可使用位流过滤器 (未示出) 对媒体数据流 310 进行过滤操作。处理器 330 也可利用专用霍夫曼解码器 333 执行霍夫曼解码过程。霍夫曼解码器 333 可包括硬件或软件中,加速解码过程的简单的可变位查寻表。

[0066] DIP 排序器 345 起视频处理器 350 的控制单元的作用。事实上,DIP 排序器可被认为是视频处理器 350 的一部分。排序器 345 是从其接收数据的存储器控制器 340 的客户机,并将数据连同控制信号一起提供给视频处理器 (变码器) 350。

[0067] 在一个实施例中,视频处理器 350 是接收来自 DIP 排序器 345 的 DIP 和控制信号,随后着手按照流水线方式进行媒体解码,以使通过量达到最大的视频变码器。在至少一个实施例中,视频处理器 350 使用解压缩块 355 解压缩 DIP。解压缩块 355 的数据输出被传送给缩放部件 (scalar block) 360,在缩放部件 360,解压缩视频数据依据比例因子被缩放,随后被输出给压缩部件 365。在一个实施例中,压缩部件 365 压缩缩放部件 360 的输出,产生压缩数据,例如视频帧,并将压缩数据传送给输出位存储桶 370,输出位存储桶 370 存储压

缩视频数据的位,直到整个字准备好由存储器控制器 340 存储为止。下面参考图 4,更详细说明视频处理器 350 的一个具体实现。

[0068] 存储器控制器 340 将视频处理器 350 看作客户机,从输出位存储桶 370 检索其数据,并将其存储在设备存储器 390 中。一旦一帧压缩媒体从视频处理器 350 检索并存储在存储器 390 中,那么系统 DMA 总线接口 380 可存取该帧压缩媒体,并利用系统 I/O 总线 315 将其发送给北桥 307。北桥 307 再将数据发送给主 DRAM 309。在一个实施例中,主 DRAM 309 用作将分配给系统其余部分的最终媒体数据通道的存储器。在一个实施例中,主处理器 350 安排最终媒体数据通道何时将被发送给网络控制器 320,谁负责最终媒体数据通道的分配和广播。在一个实施例中,网络控制器与用于显示传送的信息的多个显示设备连接。注意北桥 307 代表使主 CPU 305 与系统(主)存储器 308 和 IO 总线 315 连接的集成或多芯片解决方案。

[0069] 现在参见图 4,根据本发明的至少一个实施例,图解说明了视频处理器 350 的一种具体实现。回想视频处理器 350 利用解压缩部件 355,对从媒体数据流解析的媒体数据通道(采取 DIP 的形式)解压缩,利用缩放部件 360 缩放输出,随后利用压缩部件 365 压缩缩放后的输出。在本发明的一个实现中,解压缩部件 355 包括去交错(deZigZag)/去量化部件 410,反向离散余弦变换(IDCT)部件 420,运动补偿 430 和宏块缓存器 440,而压缩部件 365 包括缓存运动预测器部件 450,离散余弦变换(DCT)部件 460,量化/交错部件 470 和霍夫曼编码器 480。要认识到可独立实现解压缩部件 355 和/或压缩部件 365 的一个或多个组件。

[0070] 如前所述,在一个实施例中,通过存储器控制器 340,DIP 排序器 345 从设备存储器 390(图 3)检索解码器指令分组(DIP)。这种情况下,DIP 排序器 345 随后能够按照将 DIP 提供给适当位置的方式,将 DIP 转发给变码器 350。例如,通过经变码器 350 本地总线直接寻址并提供数据,能够向各个块提供数据,或者排序器 345 能够将控制 land 和/或数据信息写入寄存器中,变码器 350 部件能够存取所述寄存器中。在正常的的数据流内,排序器将使去交错/去量化部件 410 能够检索数据。

[0071] 在一个实施例中,DIP 排序器根据通过媒体数据流 310 接收的,并存储在存储器 390 中的基本流数据,检索 DIP。回想在至少一个实施例中,媒体数据流 310(图 3)包括一个或多个 MPEG 视频数据通道。这种情况下,视频通道的压缩的 MPEG 算法通常具有离散余弦变换(DCT)阶段,离散余弦变换实现时域-频域变换。作为这种变换的结果,和位于矩阵右下角的元素相比,频域中和所得到的矩阵中索引为(0,0)的左上侧元素最接近的元素 post-DCT 被更大地加权。如果频域中的矩阵使用较低的精度表示元素矩阵右下方一半的元素,那么如果右下方一半中的较小值低于以量化因子为基础的阈值,那么它们被转换成 0。将每个元素除以量化因子是用于产生更多 0 元素的一种方法。MPEG 和相关算法通常应用更大的量化值,以便降低频域中矩阵的精度,导致更多的 0 元素,从而降低数据传输速率。因此,在一个实施例中,去交错/去量化部件 410 对媒体数据通道上 DCT 操作的结果去量化。

[0072] 类似地,虽然通常用(行,列)格式表示矩阵,不过作为 DCT 函数的结果产生的矩阵的独特性质提供一种表示 post-DCT 矩阵的不同方法。在一个实施例中,按照交错方式表示 post-DCT 矩阵,例如(1,1),(2,1),(1,2),(1,3),(2,2),(3,1),(4,1),(3,2),(2,3)等。通过按照这种方式表示 post-DCT 矩阵,可能遇到 post-DCT 矩阵中相当长的一串 0。从而,

利用 (Run, Level) 格式,能够更有效地表示 post-DCT,这里 Run 代表连续 0 的数目,Level 代表 post-DCT 矩阵的下一非 0 元素的值。随后可借助固定的 Huffman 表对 (Run, Level) 对编码,以便进一步压缩 post-DCT 矩阵。

[0073] 在一个实施例中,来自去交错 / 去量化部件 410 的输出被发送给,或者以其它方式提供给将所述输出从频域转换到时域的组件。例如,如果先前根据离散余弦变换处理媒体数据流 310 (由 DIP 表示) 的解析通道,那么来自去交错 / 去量化部件 410 的输出可被发送给反向离散余弦变换 (IDCT) 部件 420,在反向离散余弦变换 (IDCT) 部件 420,利用反向离散余弦变换函数将所述输出从频域转换到时域,从而产生一块数据。在不脱离本发明的精神或范围的情况下,可使用频域和时域之间的其它转换方法。IDCT 部件 420 产生的块可被存储在宏块缓存器 430 中。当产生了足以形成宏块的足够块时,宏块缓存器将数据的宏块连同宏块信息 (包含运动矢量和量化信息) 一起发送给运动补偿器 440。

[0074] 在一个实施例中,运动补偿器 440 可如前参考图 2 的步骤 240 说明的那样工作,运动补偿器 440 存储宏块信息以便以后使用。在一个备选实施例中,存储的宏块信息只包含量化信息。如果传送给运动补偿器 440 的宏块是基准帧 (I-帧) 的一部分,那么运动补偿器 440 对该宏块解码,并利用存储器控制器 340,将解码结果和量化信息写入设备存储器 390 (图 3)。基准帧一般被用于译解必须根据运动矢量构成的其它非基准帧。

[0075] 在一个实施例中,运动补偿器 440 产生的新的宏块被发送给缩放部件 360,在缩放部件 360,利用比例因子缩放该宏块。例如,如果宏块包括块的 4×4 矩阵,并且使用为 2 的比例因子,那么所得到的缩放宏块会包括块的 2×2 矩阵。所得到的缩放宏块随后被传送给缓存运动预测器 450,以便开始压缩过程。

[0076] 在一个实施例中,当产生足以形成一个新的完整宏块的足够的缩放宏块时,诸如缓存运动预测器 440 之类运动矢量发生器处理相关的存储运动矢量,从而产生一组新的运动矢量。在一个备选实施例中,可按照传统方式,根据相邻帧的宏块,产生运动矢量。

[0077] 每次传送一块数据地将基准帧传送给 DCT 460。DCT 460 获得每块数据,并通过例如如前所述,进行离散余弦变换,将该数据块从时域变换到频域。在一个实施例中,如果宏块不具有与其相关的任意运动矢量,例如帧内或 I 帧,那么在不受运动预测器 450 处理的情况下,利用内部数据通路,直接从缩放部件 360 将宏块传送给 DCT 部件 460。

[0078] 在一个实施例中,DCT 460 输出的修改块被发送给量化 / 交错部件 470,在量化 / 交错部件 470,该块从 (行,列) 格式被量化和 / 或变换,或者“交错”成对于运动等级编码来说更佳的格式。来自量化 / 交错部件 470 的量化和 / 或交错输出随后可被发送给霍夫曼编码器 480,霍夫曼编码器 480 压紧或压缩来自量化 / 交错部件 470 的输出,并将压缩结果以位流的形式发送给输出位存储桶 370。在一个实施例中,输出位存储桶能够在等待时间较小的情况下,以位级方式,将数据存储在任何位单元。例如,输出位存储桶能够起累积从霍夫曼编码器 480 输出的位流的缓存器的作用,从而能够高效地将所述位流写入设备存储器 390。

[0079] 变码器 350 还包括 IDCT 部分 475,它对来自量化器的数据进行 IDCT。该数据与来自运动预测器 450 的数据结合,并被存储以便校正有损压缩的效果。在一些实施例中,使用有损压缩的校正,因为变码器产生的帧可被用作形成其它帧的基准帧。通过提供校正有损压缩的步骤,能够限制误差的扩散,从而提高编码器部分产生的数据的准确性。

[0080] 图 5 图解说明了存储器控制器 340 的一个具体实施例。具体地说,图 5 图解说明具有接收来自特定客户机的请求的多个客户机接口 FIFO 510-514(先进先出存储器)的存储器控制器。客户机接口 FIFO 分别接收来自特定客户机的数据存取请求。在图解说明的实施例中,在多路复用器 520,在多条写数据总线接收写数据。判优器 525 确定将服务于哪个待决的写请求,并提供适当的控制信号,以便接收对应的输入通道。存储器排序器 530 按照有效的方式组织读请求。例如,一旦一页存储器打开,那么排序器通常将尝试服务来自同一页的请求,以便降低等待时间开销。除了与存储器,例如双倍数据速率(DDR)存储器连接之外,排序器能够使用与寄存器总线客户机接口连接的寄存器文件。读数据返回通路将获得的数据提供给读返回控制部分 535。读返回控制部分 535 与读数据总线连接,从而将信息返回给多个客户机之一。

[0081] 参见图 6,根据本发明的至少一个实施例说明了利用译码系统 300 的方法。译码方法 600 开始于步骤 610,在步骤 610,媒体数据流 310(图 3)被变码器 303 接收,或者被传送给变码器 303。如前所述,媒体数据流 310 可存储在设备存储器 390 中(图 3),以便以后检索。在步骤 620 中,处理器 330 解析媒体数据流 310,识别所需的数据通道,例如从 MPEG 播放器发送的视频数据通道。处理器 330 也可在步骤 620 对媒体数据流 310 执行其它处理,例如纠错,过滤等。

[0082] 在一个实施例中,在步骤 630 中,解析的数据通道以解码器指令分组(DIP)的形式,被传送给视频处理器 350。如前参考图 4 所述,视频处理器 350 的解压缩部件 355 对 DIP 解压缩。在步骤 640 中,缩放部件 360 利用比例因子缩放解压缩部件 355 的输出,以便减小与媒体数据流 310 的所需通道的元素相关的数据量。例如,通过缩放 MPEG 视频通道的宏块,表现每帧需要的数据减少。在步骤 650 中,如前参考图 4 所述,从缩放部件 360 输出的缩放数据被压缩部件 365 压缩。来自压缩部件 365 的压缩输出随后被输出给一个或多个接收器单元。接收器单元可包括各种媒体设备,例如视频显示终端,音频设备等。

[0083] 作为译码系统 300 执行的处理的结果,媒体数据流 310(图 1)的通道被按比例缩小或者被压缩,导致与所述通道相关的较少数据。例如,视频处理器 350(图 3)能够按比例缩小来自 MPEG 播放器的视频数据流(媒体数据流 310)的通道,导致每帧数据较少的通道。这种按比例缩小随后允许通过网络或者其它媒介,更快和/或更有效地传送视频数据通道。例如,在一个实施例中,媒体数据流 310 包括将被用于实时重放的多个视频数据通道。这种情况下,通过单独按比例缩小一个或多个视频数据通道,能够更快和/或更高效地传送它们,从而支持视频通道的实时重放。

[0084] 这里公开的特殊实施例容许各种修改和备选形式。于是附图和详细说明中举例表示了具体实施例。但是应明白,附图和详细说明并不意图将本发明限制于公开的特定形式,相反,在法律允许的最大范围内,本发明将覆盖落入由权利要求及其等同物限定的本发明的精神和范围内的所有修改,等同物和替换物。例如,在图 1 中图解说明的系统中,解码器 110 和存储器 120 及 140 之间的连接可代表独立总线或共用总线。同样地,编码器 150 和输出缓存器之间的连接,以及解码器和输入缓存器之间的连接可代表相同或不同的连接,就与存储器 120 和 140 的连接来说,甚至可以共用。另外,在本发明的另一实施例中,选择编码器确定运动矢量的标准操作模式和如同这里说明的运动矢量复用模式之一。一种模式的选择通常以先前存储的宏块信息的可用性为基础。注意在另一实施例中,在标准操作模式

内,解码器不存储运动矢量。

[0085] 本发明的一种实现是驻留在如附图中描述那样配置的一个或多个处理系统的随机存取存储器中的多组计算机可读指令。在被处理系统要求之前,该组指令可存储在另一计算机可读存储器中,例如存储在硬盘驱动器或者存储在可拆卸的存储器,例如最终用在 CD 驱动器或 DVD 驱动器中的光盘,或者最终用在软盘驱动器中的软盘中。此外,该组指令可存储在另一图像处理系统的存储器中,并通过局域网或广域网,例如因特网传送,这里传送的信号可以通过诸如 ISDN 线路之类媒介传播的信号,或者信号可通过空中传播,并被本地卫星接收,从而被传送给处理系统。这种信号可以是包含载波信号的复合信号,包含在载波信号内的是包含实现本发明的至少一个计算机程序指令的所需信息,并且当用户需要时,可照这样被下载。本领域的技术人员会认识到各组指令的物理存储和 / 或传送物理改变以电、磁或化学方式存储所述各组指令的媒介,以致所述媒介携带计算机可读信息。

[0086] 数据指令分组 (DIP 或“DIP 分组”) 是能够同时包含指令信息和数据的数据分组。图 7 图解说明了 DIP 分组的一个例子。

[0087] 图 7 中图解说明的 DIP 具有首标部分,配置部分和数据部分。DIP 的首标包括一个操作码和一个尺寸指示符。DIP 的操作码部分指示一个指令,或者将与具体的 DIP 分组相关的其它控制信息。在一个实施例中,首标的尺寸部分指示是否存在与 DIP 相关的尺寸有效负载。例如,尺寸部分可具有指示不存在与 DIP 相关的任何配置或数据,只指示首标大小的值。另一实施例允许尺寸部分值指示 DIP 由 DIP (EOD) 指示符的末端端接。另一实施例允许尺寸部分值指示用后续的字节数规定 DIP 的大小。例如,尺寸部分值可指示接下来的两个或四个字节包含规定 DIP 大小的无符号整数。

[0088] DIP 的配置部分可包含与其具体操作码相关的配置信息。例如,DIP 可具有指示包括图像控制信息的操作码,从而能够预先确定配置部分,以致 DIP 解码器会从位于预定位置的配置部分获得控制信息;注意 DIP 的配置部分是可选的,取决于实际的 DIP。另外,配置部分可包含上面所述的尺寸有效负载数据。

[0089] 指令分组的数据部分可包含与 DIP 分组相关的数据。例如,压缩或非压缩视频数据可包含为具体 DIP 的一部分。注意,数据部分的存在以操作码为基础。在不存在配置或数据部分的模式下,首标部分可规定与配置和操作数据无关的要执行的具体操作。

[0090] 根据公开内容的具体模式,DIP 分组被用于按照视频数据与从中接收视频数据的数据类型无关的方式,分包数字视频数据,例如基本流数据。于是,如果特定的视频标准,例如 MPEG 标准在未来发生变化,或者如果需要支持一种不同的数字视频标准,用于产生 DIPS 的程序可发生变化,以适应接收的视频数据,同时保持 DIP 格式不变。由于能够独立于具体的视频标准设计硬件,因此效率较高。

[0091] 具体 DIP 操作码的例子包括指示 DIP 包含视频时间戳记信息的 DIP 操作码,指示图像配置信息包含在 DIP 内的 DIP 操作码,指示片断或宏块信息包含在 DIP 内的 DIP 操作码,指示缩放宏块信息包含在 DIP 内的 DIP 操作码,指示量化矩阵数据包含在 DIP 内的 DIP 操作码,指示最后的图像 DIP 应被重复的 DIP 操作码,和 / 或指示具体图像位置,例如片段、帧或宏块的末端的 DIP 操作码。要明白可包括另外的操作码,以支持其它功能,并且包含操作码的实际 DIP 通常被存储并且随后被解码。

[0092] 图 8 以方框图的形式图解说明了用于产生 DIP 分组的系统 300 (图 3) 的一部分。

图 8 包括 DIP 发生器部分 710, 在一个实施例中, DIP 发生器部分对应于图 3 的处理器的一部分; 设备存储器 390; 和数据输入控制器 715, 在一个实施例中, 数据输入控制器对应于 DIP 排序器 345 的一部分。要认识到在备选实施例中, 可利用不同于图解说明的其它硬件或软件解决方案, 实现图解说明的各个组件。

[0093] DIP 发生器 710 包括基本流读出器 712, 数据指令分组发生器 711 和指针控制部分 713。在一个实施例中, 可用软件在具有输入端口和输出端口的组合物的通用处理器上实现每个部件 711-713, 所述输入和输出端口用于向单个软件模块提供数据, 以及接收来自单个软件模块的数据。在另一实施例中, 可用具有使组件相互耦接的输入和输出端口的硬件实现部件 711-713。

[0094] 存储器 390 包括存储基本流数据 391、DIP 数据 392 和 DIP 指针数据 393 的各个部分。

[0095] 操作上, 基本流读出器 712 接收来自存储器部分 391 的基本流数据或者其它视频数据。在一个实施例中, 基本流数据 391 首先接收自媒体源 301, 如图 3 中所示, 并被存储在部分 391 中。一般来说, 该数据将被存储为代表视频图像的基本流数据。如同参考图 3 所述那样, 可作为对存储器控制器的客户请求的结果, 接收该数据。基本流读出器接收视频数据, 从而它可由数据指令分组发生器 711 分组。数据指令分组发生器产生实际的 DIP, 所述实际 DIP 包括图 3 的视频处理器部分 350 的后续控制和操作所必需的指令和数据。

[0096] 除了分包视频数据供视频处理器 350 使用之外, 数据指令分组发生器能够产生 DIPS, 从而配置视频处理器 350。在一个实施例中, 数据的每个宏块被存储为单个 DIP。在其它实施例中, 多个宏块可与单个 DIP 相关。例如, 整段数据可作为单个 DIP 的一部分被传送。同样地, 要认识到可实现任意数目的 DIP 变化, 以传送视频信息。

[0097] 当具体的 DIP 由数据指令分组发生器 711 产生时, 它被写回到 DIP 存储部分 392, DIP 存储部分 392 是存储器 390 的一部分。要认识到 DIP 存储部分 392 可以是易失性存储器, 或非易失性存储器, 例如硬盘。在一个实施例中, DIP 存储部分 392 可以是与视频处理器存取的帧缓存器相关的环形缓存器。通常, 相对于正在产生的图像的查看, 实时地发生基本流读出器 712 对基本流数据的存取和 DIP 的产生。一旦 DIP 数据被存储在存储单元 392 中, 那么数据输入控制器 715 能够接收 DIP 数据, 从而将其提供给 DIP 解码器, 如后所述。

[0098] 根据本发明的具体实施例, 指针控制部分 713 跟踪何时产生新的 DIP, 并将 DIP 的指针存储在存储单元 393 中。这里将更详细地说明 DIP 指针数据的使用。

[0099] 图 9 更详细地图解说明设备 303(图 3)的一部分和设备存储器 390。具体地说, 图 9 图解说明存储器控制器 340, DIP 排序器 345, 视频处理器 350 和高速缓存存储器 341。图 9 中图解说明的 DIP 排序器还包括可对应于图 8 的数据输入控制器 715 的 DIP 输入控制模块 346, DIP 解码器模块 347, 和 DIP 输出控制模块 348。DIP 排序器与视频处理器 350 耦接。

[0100] 视频处理器 350 包括和其缩放部件 360 相关的其它细节。具体地说, 缩放部件 360 还包括缩放缓存器 351 和缩放引擎 352。DIP 排序器 345 与存储器控制器 340 耦接, 存储器控制器 340 再与一个或多个存储设备, 例如设备存储器 390 耦接。注意图解说明了能够包括存储在存储器 390 中的数据的高速缓存存储器 341。

[0101] 操作上, DIP 排序器 345 的输入控制模块 346 将通过存储器控制器 340 接收具体的

DIP 信息。在一个实施例中, DIP 输入控制模块将客户读取请求提交给存储器控制器 340。存储器控制器 340 将通过高速缓存存储器 341 或者设备存储器 390 接收 DIP。当收到请求的 DIP 时, 解码器 347 将根据 DIP 分组的操作码部分执行操作。在一个实施例中, DIP 解码器模块 347 将包括与每个 DIP 操作码相关的微代码。根据具体的操作码, DIP 排序器 345 将通过 DIP 输出控制模块 348, 与视频处理器部分 350 连接。

[0102] 例如, 如果具体 DIP 中的操作码指示包含图像配置数据, 那么 DIP 解码器模块 347 将用硬件或软件执行写入视频处理器 350 的恰当寄存器, 从而用具体信息配置视频处理器 350 的功能。例如, 具体的缩放信息, 例如作为 DIP 分组的一部分包括的, 涉及源图像大小和目标图像大小的信息可被存储到控制缩放引擎 352 的功能的一个或多个寄存器中。

[0103] 图 10-14 与一个具体实施例中, 使用视频处理器 350 的缩放过程内 DIP 的使用相关。

[0104] 在本发明的一个实施例中, DIP 排序器存取存储在设备存储器 390 中的 DIP, 并将代表宏块的信息提供给视频处理器 350。一般来说, 提供给变码器 350 的 DIP 数据是代表视频宏块的压缩数据。通过将该数据连同恰当的解压缩信息一起提供给视频处理器 350, 视频处理器能够解压缩该数据, 得到解压缩的宏块数据。要明白的是, 在将非压缩宏缓存器数据存储在缩放缓存器 351 中之前, 可存在由视频处理器 350 执行的数个解压缩过程。但是, 为了便于说明, 将认为 DIP 排序器 345 将宏块数据提供给缩放缓存器。还应明白的是, 在本发明的另一实施例中, DIP 排序器会执行视频处理器执行的一些或全部解压缩功能。

[0105] 一旦宏块数据被存储在缩放部件 360 的缩放缓存器 351 中, 那么缩放引擎 352 能够存取存储在缩放缓存器 351 中的宏块数据, 从而提供缩放的宏块, 缩放引擎能够使用传统的缩放方法, 或者这里公开的缩放方法产生缩放图像。参考图 10 有益于更好地理解本发明, 图 10 图解说明了宏块与图像帧的关系。

[0106] 图 10 图解说明包含宏块 $M(0,0) \sim M(n,m)$ 的一帧视频。为了便于说明, 图 10 的宏块被描述成代表一帧数据, 但是在其它实施例中, 图 10 的宏块实际上可代表数据一块数据, 或者只是一帧或一块数据的一部分。

[0107] 图 10 的帧与特定的分辨率相关。例如, 图 10 的帧可被认为是分辨率为 1920 像素 \times 1080 像素的源帧。由于每个宏块代表该帧的 16 像素 \times 16 像素部分, 因此用于代表一帧的宏块的数目取决于帧的分辨率。应注意的是, 虽然术语宏块具有具体的含义, 为了便于说明和 / 或实现, 在本发明的其它实施例中, 可使用不同于传统宏块的块大小。

[0108] 操作上, 在被缩放引擎 352 存取之前, 图 10 中图解说明的帧的宏块被存储在缩放缓存器 351 中, 缩放引擎 352 产生代表缩放图像的目的宏块。例如, 为了产生分辨率小于 1920 \times 1080, 例如 1280 \times 720 分辨率的目的图像, 缩放引擎 352 必须存取多个源宏块。如图 10 中所示, 为了产生一个新的目的宏块 $DM(DMR0, DMC0)$, 需要四个宏块, 所述目的宏块 $DM(DMR0, DMC0)$ 是与行 $DMR0$ 和列 $DMC0$ 相关的宏块。换句话说, 当图像正被按比例缩小时, 根据一个以上的源宏块产生单个目的宏块。

[0109] 对于缩放引擎 352 来说, 为了有效地进行缩放, 缩放引擎 352 应能够通过缩放缓存器 351 获得源宏块。根据本发明的一个具体实施例, 这是通过装入第一组宏块, 例如一段数据 (即第一行的宏块 $(0,0) \sim (n,0)$) 或者第一批多段数据来实现的。新的目的片断将产生包含由 $DRM0$ 限制的水平数据的新的一行或者多个宏块。于是, 单片源宏块不包含足以产

生用于代表新片段的足够宏块的足够宏块数据。根据本发明的一个具体实施例,还只装入一行宏块的一部分。例如,为了产生目的宏块 DM(DMR0,DMC0),至少需要第二行宏块 M(x,1) 的源宏块 M(0,1) 和 M(1,1)。由于允许通过将少于整段视频数据的视频数据存储到缩放缓存器中,产生目的宏块,从而减小了缩放缓存器的大小,因此该具体实施例的效率高于已知的方法。

[0110] 一旦目的宏块被产生,那么不再需要用于产生后续目的宏块的源宏块可被重写。注意在本实施例中,缩放缓存器的大小将由正在实现的实际比例因子确定,或者由能够实现的最大比例因子确定。例如,参见图 10,可看出为了产生所示目的块大小的目的宏块(参见 M(DMR1,DMC1),需要多达 9 个源宏块。在本实施例中,缩放缓存器需要适应于存储两个片段,加上第三个片段的三个额外的宏块。当希望提供额外的缓存能力,以确保缩放引擎 352 不会停止时,可包括更大部分的第三片段。

[0111] 本实施例的优点在于由于源宏块被顺序存储,例如相对于扫描顺序,因此代表宏块的 DIP 分组被顺序存取。图 11 图解说明了依次存储在地址空间中的各个宏块的 DIP 分组。根据本发明的一个具体实施例,每个 DIP 分组可具有不同的大小。通过顺序存取 DIP 分组,使 DIP 排序器能够确定一个分组始于何处,另一分组止于何处。于是,与具体的一段宏块相关的所有 DIP 分组可被顺序存储在存储器 390 中,从而允许顺序从存储器 391 获得相邻的宏块。

[0112] 图 12 图解说明存储在一部分存储设备 390 中的 DIP 数据 391,存储设备 390 可以是帧存储器或者高速缓存存储器。如存储器表现 391 所示,DIP 数据 391 从 DIP(0,0) ~ DIP(n,m) 被顺序存储在存储器中。在一个实施例中,只有在读取当前 DIP 之后,每个后续 DIP 的位置才是可识别的。通常,这是通过相对于线性存储空间,将 DIP 存储在彼此紧邻的存储单元中来实现的。图解说明存储在缩放缓存器 352 中的数据的存储器表现 352 代表由 DIP 数据 391 产生的宏块数据。注意在图 12 中,图解说明与第一片段相关的宏块,以及宏块 (0,1) ~ 宏块 (3,1) 被存储。虽然这大于产生目的宏块 M(DMR0,DMC0) 所需的宏块的数目,但是它允许存储其它信息,以确保缩放引擎 352 不会停止。要认识到根据本发明选定的等待时间,也可使用另外的缩放缓存器大小。

[0113] 图 13 和 14 图解说明本发明的另一实施例,借助该实施例,通过只存储产生紧接的目的宏块所需的那些源宏块,能够利用具有较小尺寸的缩放缓存器 351。要认识到图 9 的系统必须相对于具体 DIP 的扫描序列,无次序地存取具体 DIP,以便译解特定的目的宏块。图 13 图解说明存储在特排序列 353 中的 DIP 数据,从而指示识别特定 DIP 的位置的指针。如图 14 中进一步所示,代表一个特定宏块的各个 DIP 的指针被存储在存储器 390 的 DIP 指针表部分中。

[0114] 在一个实施例中,以每个指针之间存在已知间隔的方式,将指针值存储在 DIP 指针部分中。例如,指针值可存储在紧邻的字中。通过将指针存储在已知或者易于计算的位置,DIP 输入控制模块 346 能够根据产生的目的宏块,存取特定的 DIP。能够无序存取 DIP 优于现有技术,以及前面描述的方法,因为由于只需容纳缩放引擎 352 产生特定目的宏块所需的那些源宏块,实现了尺寸减小的缩放缓存器。于是,DIP 输入控制模块 346 能够只存取产生所希望目的宏块需要的那些源 DIP。例如,再次参见图 14,为产生目的宏块 M(DMR0,DMC0),可以只存取与宏块 M(0,0),M(1,0),M(1,0) 和 M(2,0) 相关的 DIP。

[0115] 根据本发明的一个实施例,按照实际扫描顺序,将源 DIP 存储在存储器 390 内。在另一实施例中,按照预期的存取顺序,例如缩放顺序,存储源 DIP 分组。术语缩放顺序用于指示为了适应按照特定顺序的目的宏块的产生,需要被存取的源宏块的顺序。例如,虽然宏块的扫描顺序是从第一行开始直到最后一行,从左到右越过每行宏块,不过缩放顺序以目的宏块将被产生的顺序为基础。于是,在要产生的目的宏块的顺序是 M(DMR0, DMC0),之后是 M(DMR0, DMR1) 的情况下,扫描序列的一个例子应是按照下述顺序存取源宏块:M(0,0)、M(1,0), M(0,1), M(1,1), M(2,0), M(3,0), M(2,1), M(3,1)。注意由于宏块 M(0,0) 和 M(1,0) 被用于只产生宏块 M(DMR0, DMC1),因此一旦不再需要它们,那么它们就可被后续宏块重写。

[0116] 由于存储有指示每个顺序 DIP 被存储于何处的指针表,因此对于了解 DIP 存储于何处来说,DIP 本身存储在存储器中的实际顺序并不重要。但是,在另一实施例中,和扫描存取顺序相反,按照缩放存取顺序存储 DIPS,以便提高将紧邻存取的数据被存储在相同一页存储器上的可能性,从而减小与从存储器获得 DIP 数据相关的等待时间。同样地,要认识到存储 DIP 指针的实际顺序可以缩放序列为基础,或者按照除扫描顺序之外的某一顺序方式。例如,可按照相对于扫描存取序列,存取顺序 DIPS 的方式,存储指针本身。

[0117] 图 15-18 图解说明根据本发明的具体方法。图 15 图解说明根据本发明的存储信息的具体方法,在步骤 810,第一组宏块被存储在缩放缓存器中。在一个实施例中,第一组宏块将包含与特定的图像帧相关的一个或多个片段的视频信息。通过存储与一个或多个整个片段相关的宏块,能够确定每个相邻存储的 DIP 的起点和终点。

[0118] 在步骤 811,第二组宏块数据被存储到缩放缓存器中。第二组宏块数据代表一段帧数据的一部分。如前所述,在一个实施例中,第二组数据包括开始产生新的一组缩放宏块所需的视频片段的那些宏块。

[0119] 图 16 根据本发明的一个具体实施例,图解说明了另一方法。图 16 的方法包括已参考图 15 说明的步骤 810 和 811。在步骤 812,另外的宏块被存储到缩放缓存器中。例如,虽然步骤 810 和 811 只存储允许开始新目的宏块的缩放的足够源宏块,不过在缩放缓存器中存储另外的源宏块允许在不停止缩放引擎 352 的情况下,产生另外的目的宏块。例如,如果产生单个新的目的宏块所需的第二组宏块的最大数目为 3,那么步骤 812 可存储足够的其它宏块(即 6 个宏块),以确保缩放缓存器总是具有能够继续产生缩放宏块,而不会停止的足够信息。

[0120] 在步骤 813,确定图 16 的方法是否已将与帧缓存器相关的所有宏块存储在缩放缓存器中。如果所有宏块已被存储,那么结束该流程。如果还没有存储帧中的所有宏块,那么流程进入步骤 814,下一宏块被存储在帧缓存器中。要认识到在步骤 814 存储下一宏块能够重写不再需要的在先宏块。图 16 的方法图解说明使用数据的一个或多个整个片断,和数据后续片段的一部分,避免只存储视频的整个片段的方法。另外,图 16 的方法简化了在不使用指针表的情况下,已按照扫描顺序方式存储的 DIPS 的使用。

[0121] 图 17 的方法和图 16 的方法类似。步骤 818 和 814 被用于将初始宏块装入缩放缓存器。随后,在步骤 820,根据存储的信息产生新的宏块。一旦产生了新的宏块,流程进入步骤 830,确定帧缓存器是否已完成。如果否,那么流程进入步骤 835,装入下一宏块,流程返回步骤 820,产生一个新的宏块。重复该循环,直到已产生整个帧为止。

[0122] 图 18 根据本发明的另一实施例,图解说明了产生 DIP 的方法。在步骤 860,帧的第一宏块被识别成当前宏块。一般来说,帧的第一宏块是包含将在显示设备上显示或扫描的第一像素的宏块。通常,这将是特定帧的左上侧宏块。

[0123] 在一个实施例中,根据存储在存储器中的基本流,识别特定帧的第一宏块。作为接收数字视频数据,例如 MPEG 流的结果,基本流通常存储在存储器中。

[0124] 在步骤 861,产生当前宏块的 DIP。在一个实施例中,通过将 DIP 操作码和从存储器获得的视频数据的表现包含到 DIP 分组的有效负载部分中,产生 DIP,这里操作码指示 DIP 分组包含宏块信息。在另一实施例中,可以首先处理从基本流数据获得的宏块信息,从而在宏块数据被存储到 DIP 中之前,产生宏块数据的特定格式。

[0125] 在步骤 862,当前 DIP 被存储到存储器中。在一个实施例中,当前 DIP 可被存储到与帧缓存器相关的环形缓存器中。在该实施例中,视频处理设备的后续部分易于存取 DIP。在另一实施例中,DIP 被存储到诸如硬盘驱动器之类系统存储资源中,以便视频处理设备以后存取。

[0126] 在步骤 863,当前 DIP 的指针被存储到 DIP 指针单元中。由于 DIP 的大小可变,因此有益的是通过得到指示当前 DIP 存储于何处的指针,跟踪当前 DIP 的位置。这允许随后无序地存取当前 DIP。按照易于确定的已知方式,将指针存储在存储器的指针单元中。例如,每个指针可具有固定大小,从而导致指针之间固定的距离。当最好是只将产生特定宏块所需的那些宏块装入缩放缓存器,从而减小缩放缓存器的所需大小时,这是有利的。

[0127] 在步骤 840,确定图 18 的流程是否已完成。在一个实施例中,当通过根据特定帧的基本流产生 DIP,处理了整个一帧数据时,或者当不存在要处理的其它数据时,图 18 的流程结束。如果所述处理结束,那么流程终止,否则流程进入步骤 864。

[0128] 在步骤 864,帧的下一宏块被识别为当前宏块。在一个实施例中,下一宏块是按照扫描顺序的下一宏块。在另一实施例中,下一宏块是预期要存取的下一宏块。例如,要存取的下一宏块可以是按照扫描顺序的下一宏块。

[0129] 图 19 图解说明根据本发明一个具体实施例的另一方法。具体地说,在步骤 851,按照第一顺序存储一组可变大小的 DIP。第一顺序可由扫描顺序代表,由缩放顺序代表,或者由任意其它排序代表。

[0130] 随后,在步骤 852,按照不同于第一顺序的第二顺序存取存储的 DIPS。例如,如果在步骤 851 中按照扫描顺序存储 DIPS,那么在步骤 852,可按照缩放顺序存取 DIPS。如前所述,通过获得具有指示每个 DIP 存储在存储器中何处的条目的指针表,能够按照不同的顺序存取 DIPS。于是,如果按照第一顺序,例如缩放顺序存储 DIPS,那么将能够无序存取 DIPS,而不必解析多个 DIPS,以确定特定的 DIPS 始于何处。在所有只将产生新的特定宏块所需的那些宏块存储在缩放缓存器中的那些实施例中,实现图 19 中图解说明的具体实施例的实施例理应有利的。

[0131] 参见图 3,位操作器 335 与通用数据处理器 330 耦接。在一个具体实施例中,通用数据处理器 330 是集成到作为这里描述的专用集成电路 (ASIC) 一部分的半导体器件上的微处理器/微控制器核心。同样地,位操作器 335 被集成到 ASIC 上,并且可由处理器 330 存取,以便支持数据的位操作。

[0132] 在一个操作实施例中,处理器 330 将读取请求提供给微控制器 340。处理器 330 能

够响应各种情形,包括当:请求位于新地址的数据;在位操作器中缓存的数据达到预定水印时;以及当处理器 330 确定存在于位操作器 335 中的数据的数据的数量小于预定字节数时,产生读取请求。在一个实施例中,通过读取与位操作器相关的寄存器,处理器 330 能够确定位操作器中可用数据的数量。在另一实施例中,通过从位操作器接收指示特定数量的数据已被使用的中断,处理器 330 能够确定位操作器中可用数据的数量。

[0133] 在图解说明的实现中,处理器 330 请求的所有数据通过位操作器 335 被返回。注意在其它实施例中,处理器 330 能够向存储器控制器 340 指示,哪些数据将通过位操作器 335 返回,哪些数据将直接返回给处理器 330。如图所示,通过总线,例如专用总线,读总线或读/写总线,数据被返回给位操作器 335。

[0134] 在一个备选操作实施例中,位操作器 335 能够与处理器 330 连接,从而它自己的存储器控制逻辑电路能够向存储器控制器 340 请求数据。

[0135] 位操作器 335 接收的数据随后可被处理器 330 顺序存取。在一个实施例中,通过读取与位操作器 335 相关的寄存器,处理器 330 存取所述数据。例如,位操作器 335 可具有一组寄存器,所述一组寄存器返回从 1 到 N 的特定位数,这里 N 是处理器 330 读取总线的总线大小。另外,位操作器可具有一个位偏移量寄存器,位偏移量寄存器指示要返回的数据的第一位的位置。在另一实施例中,通过提供指示数据大小的控制信号,并提供恰当的控制信息,可从位操作器获取数据。

[0136] 图 20 图解说明位操作器 335 的一个具体实现,位操作器 335 包括线路缓存器 910,位存取控制器 905,寄存器 930 和中断逻辑电路 920。

[0137] 在一个实施例中,线路缓存器 910 是充当先进先出缓存器 (FIFO) 的环形缓存器,存储通过存储器控制器 340 从存储器返回的数据。在一个实施例中,接收的数据具有根据来自存储器的数据总线的大小测定大小的字。例如在存储器支持 32 位字的情况下,线路缓存器能够接收 32 位的数据字。在另一实施例中,数据总线的大小以通用处理器 330 的数据总线的大小为基础。

[0138] 位存取控制器 335 接收来自处理器 330 的关于特定数据的请求,并通过处理器的数据总线提供所请求的特定数据。注意处理器 335 和位操作器 335 之间的数据总线的大小可不同于存储器和位操作器之间的数据总线的大小。在图解说明的例子中,处理器 330 的数据总线是 16 位总线。根据来自处理器 330 的请求,将返回 1 ~ 16 位的数据。注意当小于最大数目的数据位被返回给处理器 330 时,根据预定设置(它可以是固定或可变设置),可用 1 或 0 填充未使用的位。

[0139] 寄存器 930 可被处理器 330 存取,在操作上与位存取控制器 905 和线路缓存器 910 耦接。在一个实施例中,如前更详细所述,处理器 330 能够监视位操作器 335 的状态,控制位操作器 335,并通过存取寄存器 930,从位操作器 335 获得数据。

[0140] 中断逻辑电路 920 被用于跟踪位操作器 335 中可用数据的数量。在一个实施例中,每次使用线路缓存器 910 中预定数量的数据时,向处理器 330 提供一个中断。在另一实施例中,通过对寄存器组 930 的一个寄存器编程,以指示每次从线路缓存器 910 读取特定数目的数据时,应产生一个中断,用户可选择预定数量的数据。例如,寄存器可被编程为监视 8 位、16 位、32 位、64 位或 128 位数据已被使用。在另一实施例中,存储在线路缓存器中的数据的当前字节的总数被存储在寄存器组 930 的一个寄存器中。

[0141] 通过向处理器 330 提供监视线路缓存器中可用数据的数量的途径,处理器 330 能够向存储器控制器 340 发出存储器存取请求,从而确保数据被存储在位操作器中。另外,通过向处理器 330 提供监视线路缓存器中可用数据的数量的途径,处理器 330 能够确保在数据请求被发送给位操作器 335 之前,数据适用于高速缓存存储器。这允许处理器 330 延迟给位操作器 335 的数据请求,除非确保对位操作器的读取不会停止将被高速缓存的数据存取请求。

[0142] 图 21 更详细地图解说明了图 20 的各个部分,包括寄存器组 930 和线路缓存器 910。

[0143] 寄存器组包括多个寄存器,包括规定线路缓存器存储单元存在的地址范围的栈顶寄存器和基址寄存器。尾部寄存器指向数据将被存取的线路缓存器的下一有效线路。注意指针可指示数据将被存取的线路缓存器的实际线路,或者线路缓存器的某一线路,所述某一线路偏离数据将被存取的线路缓存器的实际线路。

[0144] 位偏移寄存器指示相对于要存取的下一组位的第一位的偏移量。例如,图 21 指示指向位 3 的位偏移指针。位偏移值指示的位将是由位操作器输出左对齐的位。注意,在备选实施例中,返回的位可被右对齐。

[0145] 水印寄存器存储指示何时需要将另外的数据装入线路寄存器的值。水印值可指示分别与头部或尾部指针相比,确定数据何时被存取的与尾部或头部指针的偏移量。另一方面,水印值可指示与尾部指针的位置相比,确定数据何时被存取的固定位置。不同的实施例使用水印启动中断处理器 330 的过程,或者允许处理器 330 读取水印和其它寄存器,以确定另外的数据何时将被装入位操作器中。

[0146] 如前所述,线路缓存器计数寄存器能够存储指示与线路缓存器相关的有效数据的数量的值。例如,线路缓存器计数寄存器能够指示存储在还要被存取的线路缓存器 910 中的字节的数目。

[0147] 图 22 图解说明表示经过位操作器 335 的功能数据流的方框图。存储单元 955 存储大小至少为处理器 330 的数据总线大小的 2 倍的数据。由于将被提供给处理器 330 的数据字的第一位可被定位于任意位单元,因此要提供的字可被定位于下面的两个当前字内。多路复用器 950 被用于响应从存储单元 955 的一个字转变到下一字的位偏移值,将下一字装到存储单元 955。注意在图解说明的实施例中,多路复用器 955 的输出是 16 位。要认识到可以使用另外的逻辑电路(未示出)将下一 16 位提供给多路复用器 950。

[0148] 图 23 以方框图的形式,图解说明了更新位偏移值的一个具体实施例。该实施例图解说明接收当前位偏移值的数值和要读取的位的数目大小的模加法器。加法器的输出提供新的位偏移值。

[0149] 移位器 960 根据位偏移量,对准存储在单元 955 中的位值,从而移位器 960 输出的数据的最左侧位是由位偏移值指定的那些位。屏蔽器(mask)962 用 0 或 1 填充未被数据请求明确请求的那些位。例如,在一个实施例中,如果数据的 3 位被请求,那么它们将被左对齐,而屏蔽器将用 0 填充右侧的 13 位。

[0150] 在本发明的一个实施例中,响应 GET_BIT(X) 函数存取位操作器 335,这里 X 规定要返回的位数。代替使用内部寄存器和处理器操作,处理器 330 存取对应于位操作器 335 的寄存器的特定地址单元。例如,可使用 16 个特定的地址单元(寄存器)实现 16 个命令 GET_

BIT(1) ~ GET_BIT(16), 这里位操作器 335 返回请求数量的数据, 并恰当地更新寄存器, 包括位偏移值, 所述位偏移值被递增读取数据的数量。在另一实施例中, 可向位操作器提供其它命令, 位操作器存取特定数目的位, 但是并不递增位偏移寄存器或尾部寄存器。例如, 在一个实施例中, 在不递增尾部寄存器的情况下, 可存取整个字的数据。

[0151] 在前面说明的实施例中, 数据指令分组被用于将宏块数据发送给变码器 350。例如, 源宏块信息数据可作为 DIP 的一部分在变码器 350 被接收。参见图 3, 作为媒体数据流 310 一部分的视频数据 (即源运动矢量和量化信息) 在设备 303 被接收, 并被存储在设备存储器 390 中, 以便进一步处理。随后, 系统 303 检索存储的数据, 包括源运动矢量, 产生将由变码器 250 使用的 DIP。在一个实施例中, DIP 在变码器 350 被接收, 控制解码和编码功能的操作。在一个实施例中, 第一 DIP 包含解码器用于产生源像素数据的源宏块信息, 第二 DIP 包含编码器用于允许按照 2001 年 3 月 27 日提交的申请号 09/819147 的同时待审专利申请“Device and Method for Compression of a VideoStream”(该专利申请作为参考包含于此) 中所述那样, 执行运动矢量预测的源信息。

[0152] 当运动矢量和量化信息作为 DIP 的一部分被传送时, 特定宏块的宏块信息数据可在存储器控制器 330 和设备存储器 390 之间传送多达 7 次。当最初收到宏块信息数据时, 传送第一次, 当被检索以便装配解码器 DIP 时, 传送第二次, 当被检索以便装配编码器 DIP 时, 传送第三次, 当存储解码器 DIP 时, 传送第四次, 当存储编码器 DIP 时, 传送第五次, 当解码器存取解码器 DIP 时, 传送第六次, 当编码器存取编码器 DIP 时, 传送第七次。这里说明本发明的一个备选实施例, 该备选实施例减少在存储器控制器 340 和设备存储器 390 之间传送宏块信息的次数, 从而减少存储带宽的数量。在图 24-29 中描述了提高存储带宽利用率的这种备选实施例。

[0153] 图 24 图解说明描述本发明的一个具体实施例的方法。参考图 3、25-27 说明图 24 的方法, 图 25 以方框图的形式图解说明的相互重叠的源宏块和目的宏块的视频帧, 图 26 以方框图的形式图解说明与图 24 的方法相关的数据的内存映像。

[0154] 在图 24 中图解说明的方法的步骤 1010, 接收源视频信息。参考图 3, 源视频信息是媒体数据流 310 的一部分。在一个实施例中, 源视频信息包含宏块信息, 例如与 MPEG 型信息相关的宏块信息。图 25 图解说明用轻实线绘制的源帧, 所述源帧具有构成 $X \times Y$ 阵列的多个宏块, 这里 X 是表示具体宏块的数字, Y 是代表特定一行宏块的数字。例如, 源宏块 (1, 2), 也称为 SMB(1, 2) 是存在于源帧的列 1 和行 2 的源宏块。

[0155] 在步骤 1012, 存储源视频信息。就这里的具体实施例来说, 公开内容将集中于源视频信息的宏块信息部分。图 26 图解说明存储源数据阵列的内存映像。在一个实施例中, 源数据阵列中的每个节点包括对应宏块的特定宏块配置数据 (MBCONFIG), 以及与该特定宏块相关的宏块信息。参考图 3, 源视频信息可存储在存储器 390 中。

[0156] 存储在 MBCONFIG 字段中的宏块配置数据包括和特定宏块相关的各种信息。例如, MBCONFIG 字体能够指示是否将利用正向预测, 反向预测, 或者正向预测和反向预测, 处理运动补偿操作。另外, MBCONFIG 字体能够存储指示运动矢量的第一或第二最低有效位是否识别要使用的半像素信息的数据, 可包括间距指示符的值, 以便指示如何存取源运动矢量阵列中的下一节点, 以及与对接收的源视频解码或编码相关或者有用的任意其它类型的信息。

[0157] 在步骤 1014,接收目的图像信息。通常,目的图像信息包括目的图像的分辨率信息。图 25 图解说明一个目的帧,参见重叠在源帧上的粗体虚线。目的帧具有排列成 $M \times N$ 阵列的多个宏块,这里 M 是代表目的帧中的一列宏块的数字, N 是代表目的帧中一行宏块的数字。例如,目的宏块 (1,2) 或 $DMB(1,2)$ 是存在于目的帧的列 1 和行 2 的目的宏块。

[0158] 在步骤 1016,根据目的图像信息和源图像信息,产生索引表。具体地说,可确定为了产生每个目的宏块,需要存取哪些源宏块。例如,当将相对于源图像按比例缩小目的图像时,如图 25 中所示,来自源帧的多个宏块包含为了产生每个目的宏块,解码器所需的信息。例如,为了产生目的宏块 (0,0),宏块 (0,0) 是图 26 的阵列的左上侧目的宏块,解码器/编码器需要源宏块 : $SMB(0,0)$, $SMB(0,1)$, $SMB(1,0)$ 和 $SMB(1,1)$ 的源宏块信息。需要这四个宏块,因为目的宏块包含来自这些宏块中每个宏块的至少一些数据。

[0159] 根据宏和目的图像分辨率信息,产生具有多个条目的索引表,每个条目具有先前在步骤 1012 存储的宏块的源运动矢量和量化信息的指针。相对于其它索引表条目排列索引表中的条目,以致当按照预定方式,例如从第一条目到最后条目存取索引表时,条目内的指针可被用于按照解码器和编码器产生目的宏块信息所需的顺序,存取在步骤 1012 存储的源运动矢量。

[0160] 在步骤 1016 产生的索引表被存储在存储器中。参见图 26,第一宏块信息(包括运动矢量和量化信息)索引表包括 $POINTER\ 1 \sim POINTER\ LAST$,这里 $POINTER\ LAST$ 指示为了完成视频数据的特定帧的产生,需要寻址的源宏块的最后运动矢量。参见图 25 的实施例,可看出每个目的宏块需要四个源宏块的数据,于是,图 26 的索引表 1 中条目的总数为 $M \times N \times 4$ 。

[0161] 除了包含特定源节点的指针之外,索引表条目还包括指示与要产生的特定目的宏块相关的最后源宏块的字段。该最后源宏块指示符可被解码器或编码器用于确定何时收到足以产生目的宏块的足够源数据。

[0162] 图 27 以表格形式,图解说明了具体索引表的内容的表现。图 27 的第一列表示具体索引表内的条目位置,第二列表示条目指针指向哪个目的宏块信息,第三列表示哪些条目对应于产生新的目的宏块所需的最后源宏块信息。列 3 中的“1”表示该条目是产生特定目的宏块所需的最终源宏块。

[0163] 在一个实施例中,条目位置 1 对应于图 26 中图解说明的源数据阵列的第一项,并包含 $POINTER\ 1$ 。要注意的是特定的源宏块通常被索引表的一个以上的条目引用。例如,在图 27 中,条目 2 和条目 5 都指向源宏块 $SMB(0,1)$ 。这是因为源宏块 $SMB(0,1)$ 包含目的宏块 $DMB(0,0)$ 和 $DMB(0,1)$ 需要的数据。

[0164] 在步骤 1018,产生引用索引表的一个条目的一个或多个数据指令分组。通常,该 DIP 将引用索引表中,解码器和编码器用于存取数据并产生目的宏块的第一条目。注意,如果存在多个目的地目标,那么可产生并存储另外的索引表。通过每次存储宏块信息数据,并产生解码器和编码器通过其能够按照特定顺序存取矢量的索引表,和存储器必须为正在产生的每个目的图像存储两组 DIP 的实施例相比,减小了存储带宽。

[0165] 图 28 图解说明根据本发明的另一方法。在步骤 1050,存取第一索引表。如前所述,该索引表被这样存储,以致索引表的顺序指示为了产生目的宏块,需要被存取的特定宏块信息,例如某一宏块信息的顺序。

[0166] 在步骤 1052, 根据第一索引表条目的顺序, 按照第一顺序存取多个源宏块信息。参见图 27, 根据存储在 INDEX TABLE 1 的条目内的指针值, 存取第一批多个源宏块信息。在步骤 1054, 根据在步骤 1052 获得的数据, 产生特定的图像数据。在一个实施例中, 解码器利用获得的宏块信息, 产生源像素组。在另一实施例中, 变码器的编码器部分根据在步骤 1052 获得的数据, 产生估计的运动矢量。

[0167] 像素组和估计的运动矢量的产生通常与源像素组的产生同步实时进行。注意同步实时意味着通过利用独立的处理模块, 满足系统的实时要求, 这两个功能能够在时间上同步进行 (在完全相同的时间), 或者通过利用一个或多个处理模块, 这两个功能能够顺序进行, 从而处理模块足够快地实现这两个功能, 以适应取决于这两个结果的完成的应用的实时要求。

[0168] 例如, 视频应用的实时同步意味着足够快地产生来自解码器的源像素和来自编码器的估计运动矢量, 以便用于以所需的帧显示速率产生目的宏块。例如, 特定的实时显示速率可以是每秒 20、30 或 60 次。

[0169] 对于第二目的系统来说, 步骤 1056、1058 和 1060 类似于步骤 1050、1052 和 1058。例如, 两个监视器能够接收源图像的表现。要认识到根据该具体实施例, 只将附加源数据, 即源宏块信息的一个副本存储在存储器中, 以便支持一个或多个目的图像。

[0170] 应注意的是, 对于指定图像来说, 索引表通常是静态的, 因为源图像和目的图像之间的缩放很少发生变化。通过使用固定大小的节点存储源数据, 新的源帧的每个宏块可被存储在和前一组宏块相同的宏块覆盖范围 (footprint) 中, 从而在不发生变化的情况下, 允许使用相同的索引表存取多帧的宏块数据。

[0171] 图 29 图解说明了根据本发明的具体系统。在控制器部分 1120 接收视频源数据, 控制器部分 1102 负责解析来自输入源的数据, 并且至少将一部分源宏块信息存储在存储器中。例如, 控制器 1102 负责存储在图 26 中图解说明的源数据阵列信息。注意由于根据索引表条目中的指针值, 单独存取包含宏块信息的节点, 于是所述节点不需要被存储为阵列的形式。在一个实施例中, 利用处理器 330, 可用软件或固件实现控制器 1102。

[0172] 控制器 1102 与存储器控制器 1104 连接, 存储器控制器 1104 与存储器 1111 连接, 以便存储该数据。另外, 控制器 1102 从视频源获得信息, 确定接收的源视频图像的尺寸。源视频图像的尺寸指示用于表现每个源帧的分辨率和 / 或宏块的数目。源图像尺寸被存储在部件 915, 它可以是存储单元, 例如寄存器或存储器 1111 的一部分。另外, 第一目的图像尺寸被存储在部件 1116 中, 第二目的图像尺寸被存储在部件 1117 中。目的图像尺寸指示编码器 1106 产生的理想输出图像的分辨率。为了便于说明, 将只引用存储在部件 1116 中的第一目的图像尺寸。另外, 要认识到, 如果理想的是产生两个以上的目的图像, 那么可包括两个以上的目的图像尺寸部分。在一个实施例中, 控制器 1104 对应于如图 3 中所示的存储器控制器 340。类似地, 存储器 1111 对应于特定实施例中的设备存储器 390。

[0173] 索引表发生器 1109 接收或者以其它方式获得存储在 1115 中的源图像尺寸和存储在 1116 中的目的图像尺寸。根据源图像尺寸和目的图像尺寸, 索引表发生器将产生具有如前所述格式的索引表。存储器控制器 1104 协调将发生器 1109 产生的索引表存储到存储器 1111 中。在一个实施例中, 通过用软件或固件执行代码, 可由处理器 330 实现索引表发生器 1109。

[0174] 编码器 1106 随后通过参考索引表内特定条目的指针,使用存储器控制器 1104 检索存储在存储器 1111 的源宏块信息。按照类似的方式,解码器也可存取存储在存储器 1111 的源宏块信息。注意编码器和解码器可被设计成允许对存储器 1111 的公共读取获得特定的运动矢量。另外,编码器和解码器能够共享公共高速缓存存储器,以便减小存储器总线上的活动。在一个实施例中,编码器 1106 可由一部分压缩部件 365(图 3)实现,解码器 1107 可由一部分解压缩部件 355 实现。

[0175] 在前面的详细说明中,参考了构成本发明一部分的附图,附图中举例表示了可实践本发明的具体优选实施例。足够详细地描述了这些实施例,以使本领域的技术人员能够实践本发明,显然可以利用其它实施例,并且在不脱离本发明的精神或范围的情况下,能够做出逻辑、机械、化学和电学变化。为了避免对于使本领域的技术人员能够实践本发明来说不必要的详细,说明省略了本领域技术人员熟知的一些信息。此外,本领域技术人员易于构成包含本发明教导的许多其它变化的实施例。因此,本发明并不局限于这里陈述的具体形式,相反,本发明意图覆盖能够合理地包括在本发明的精神和范围内的所有这种替换、修改和等同物。于是,前面的详细说明不是对本发明的限制,本发明的范围只由附加的权利要求限定。

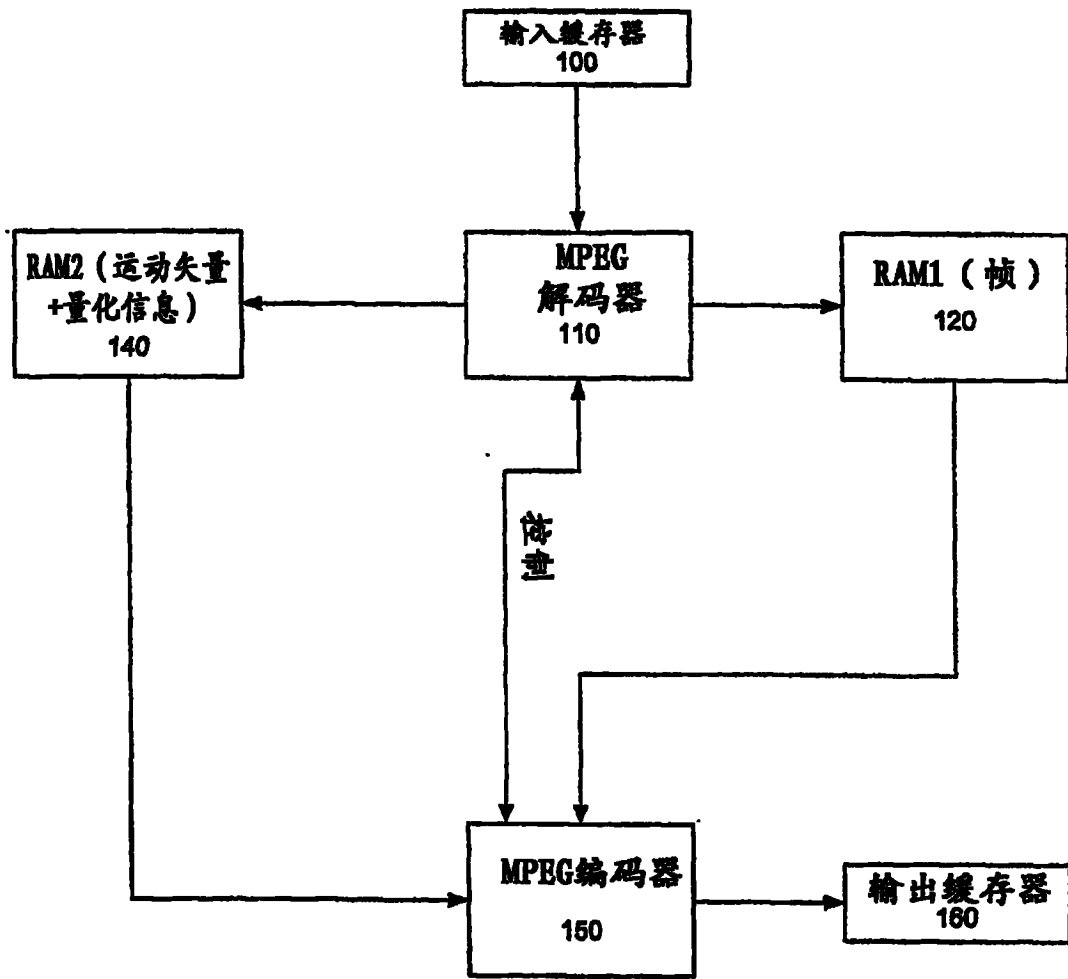


图 1

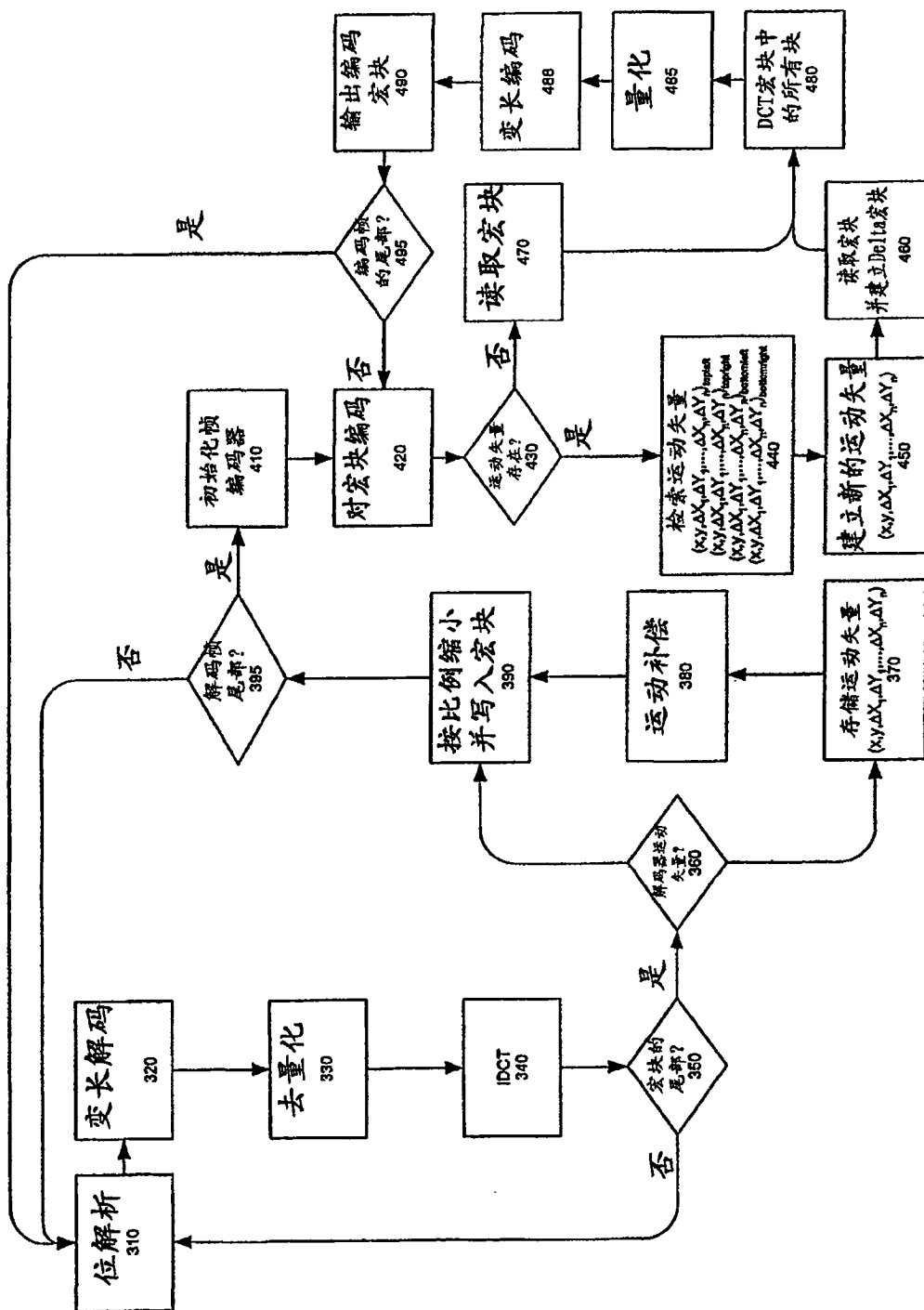


图 2

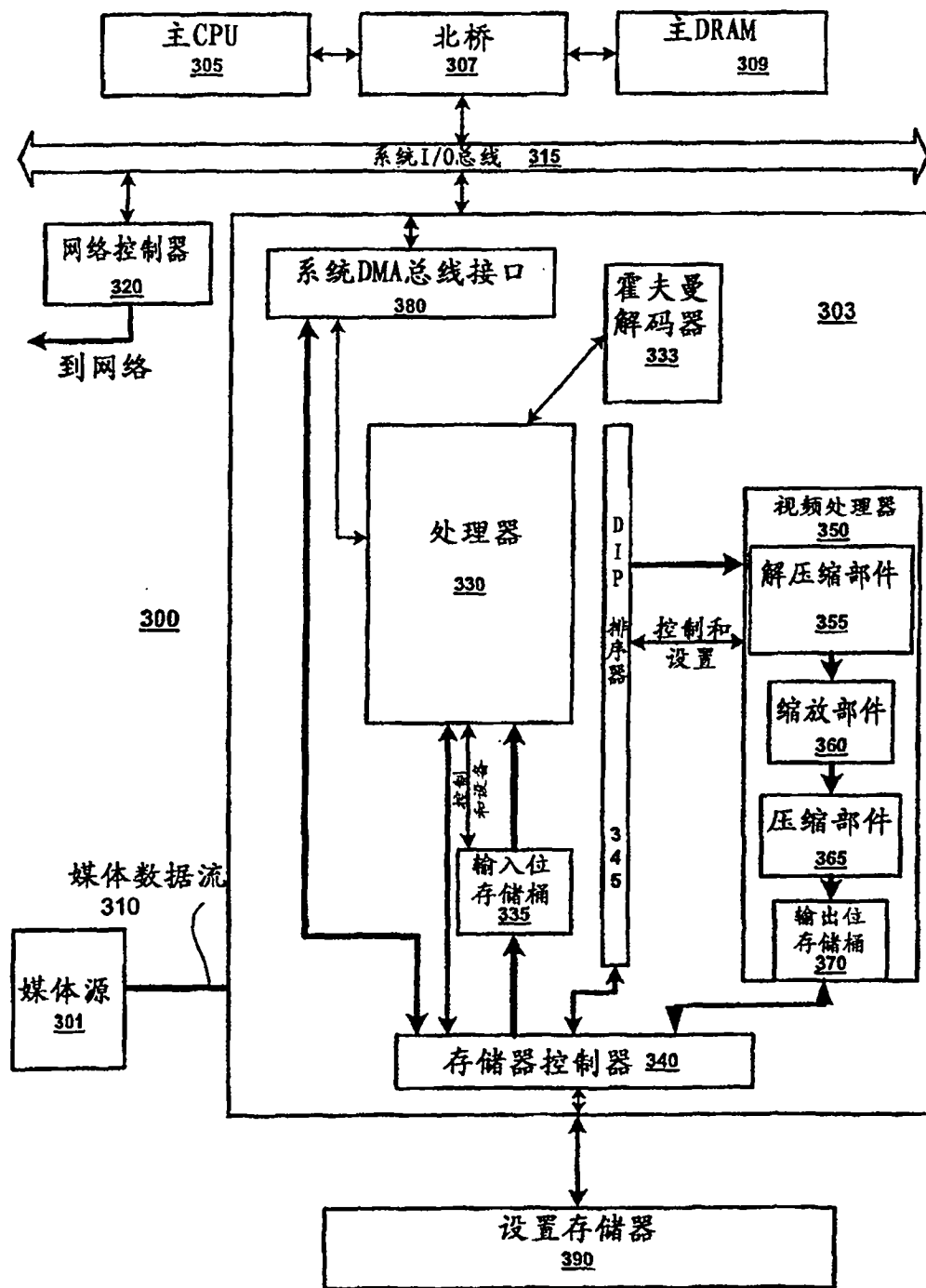


图 3

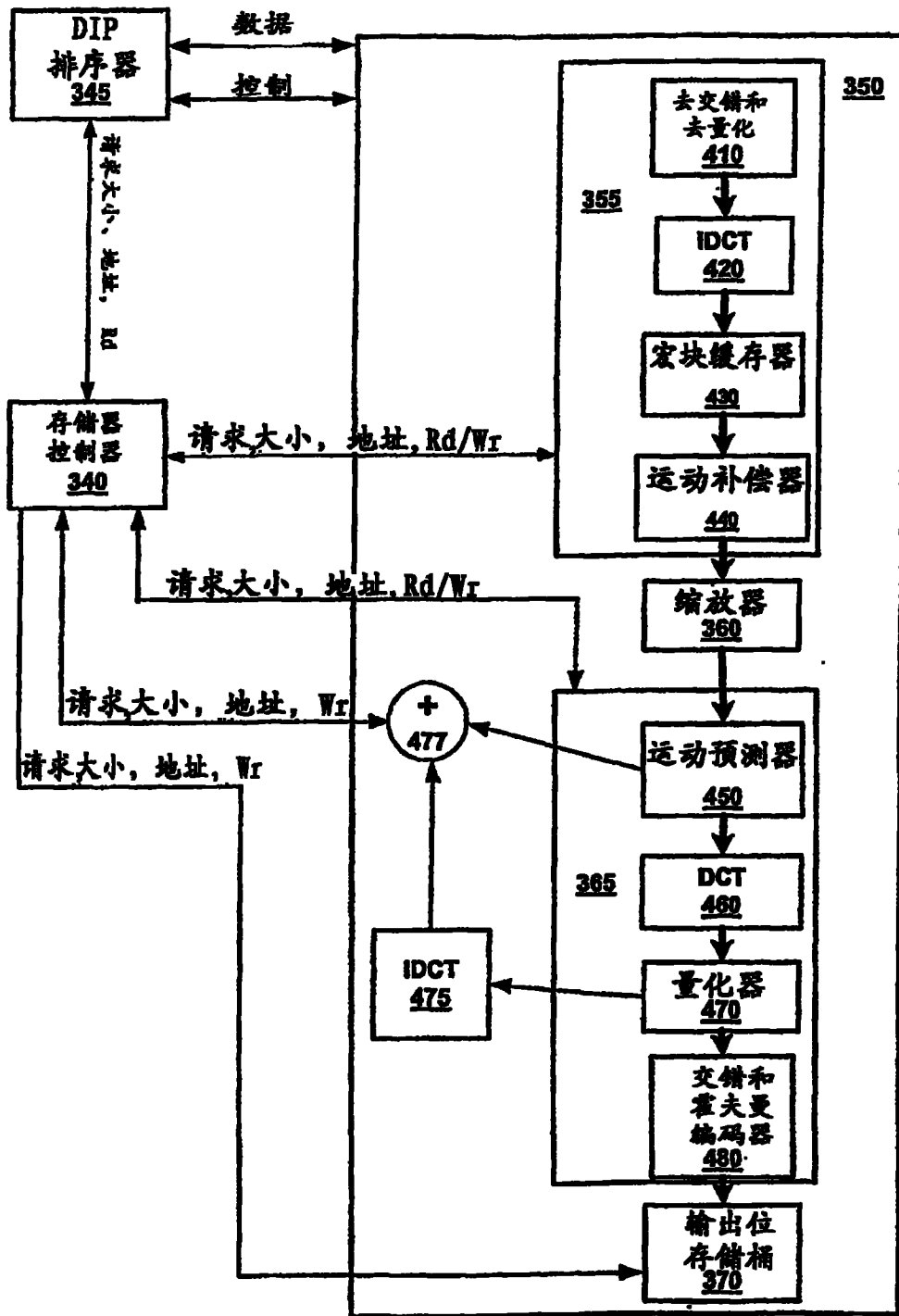


图 4

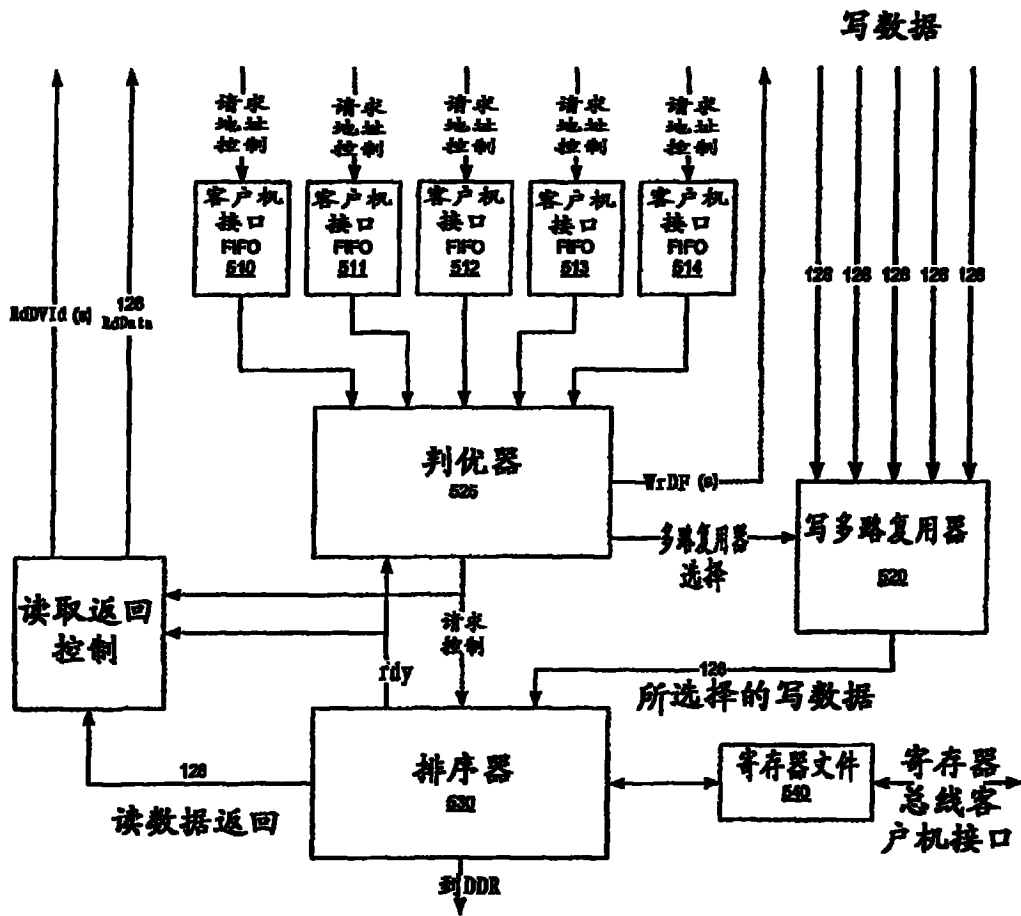


图 5

600

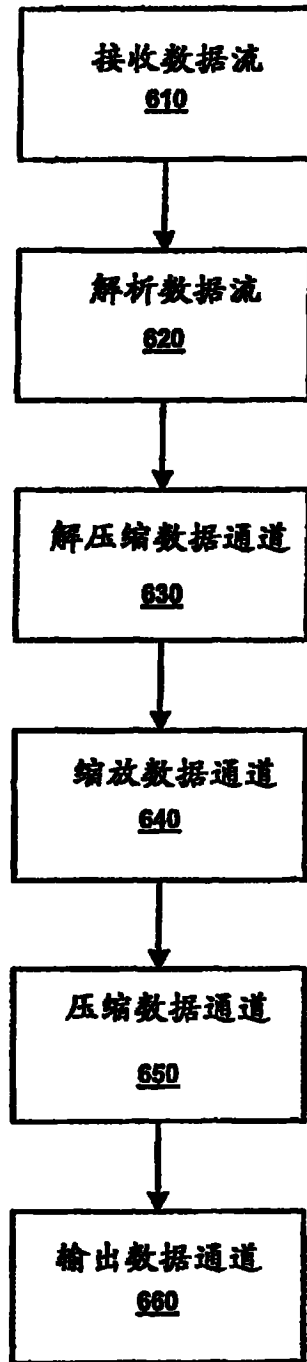


图 6

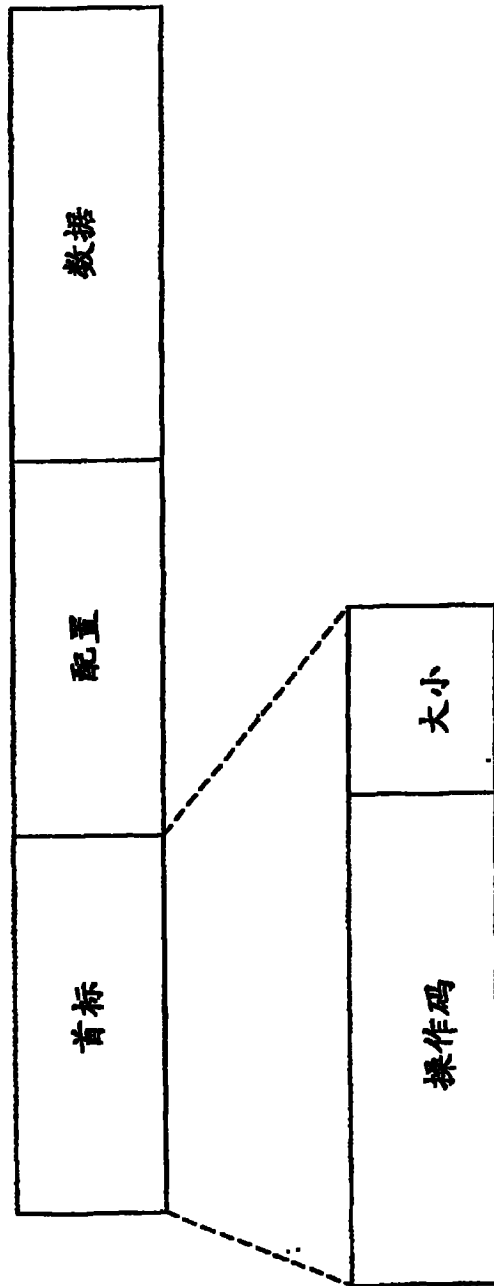


图 7

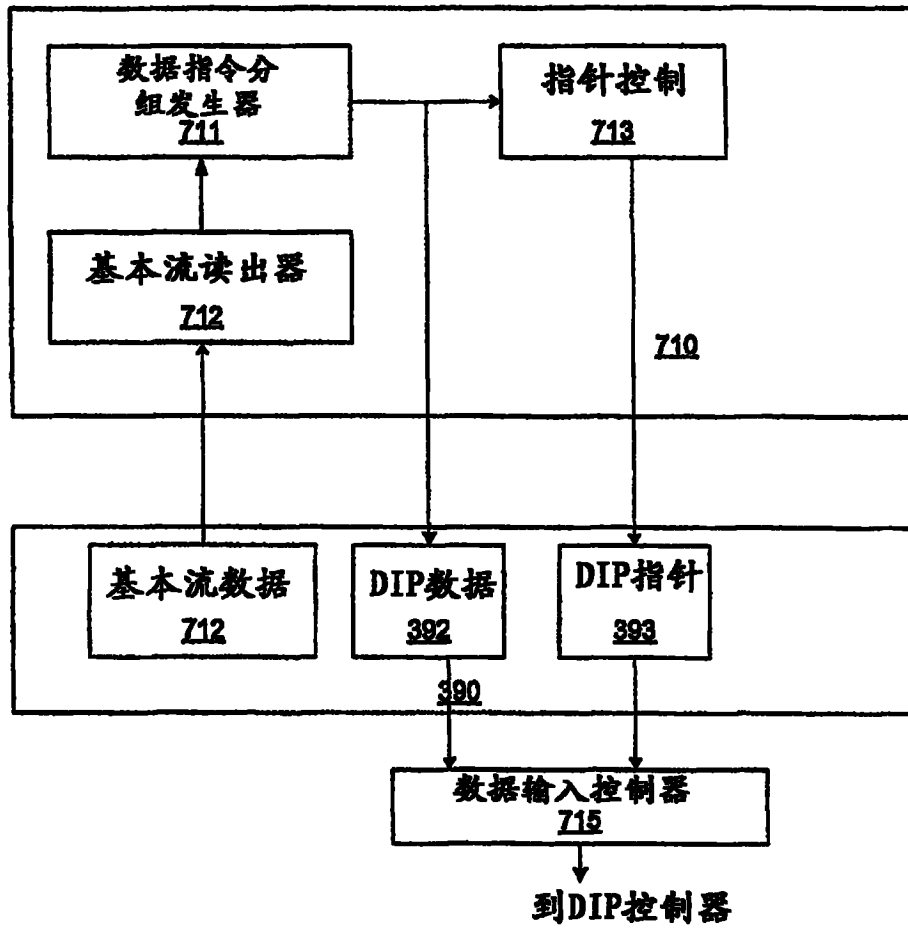


图 8

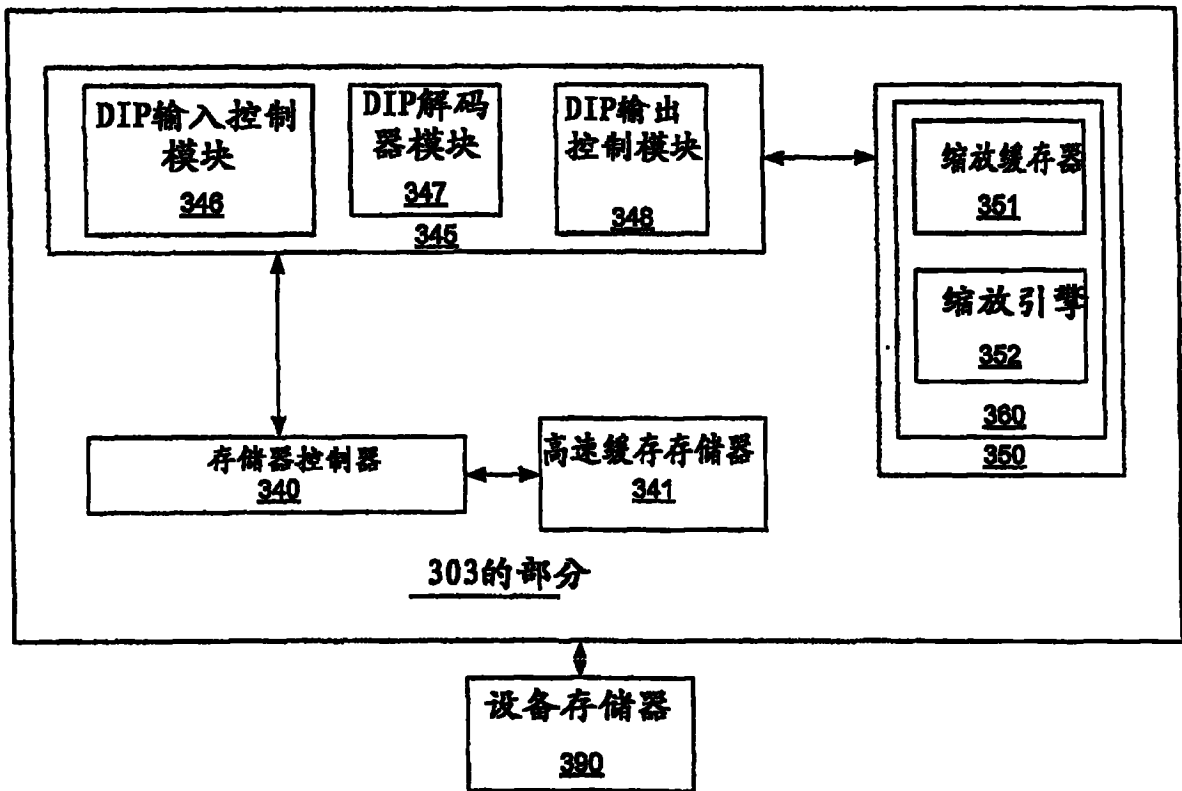


图 9

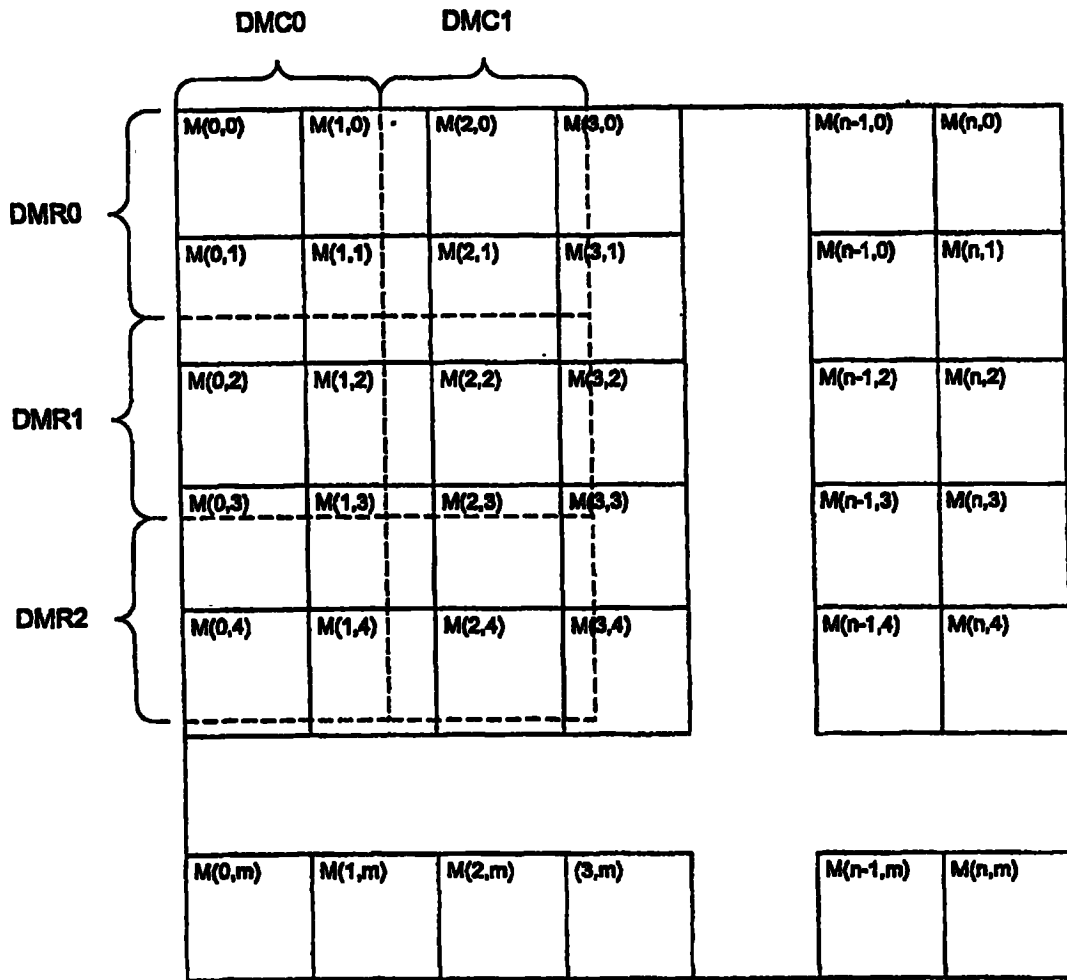


图 10

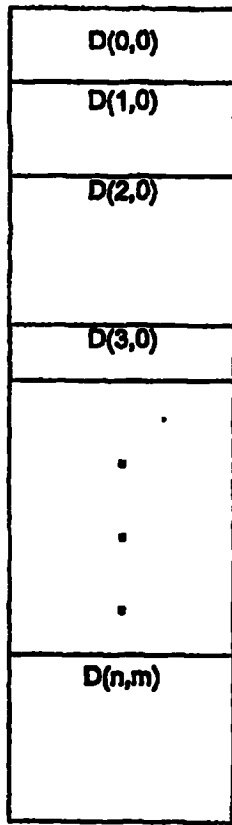


图 11

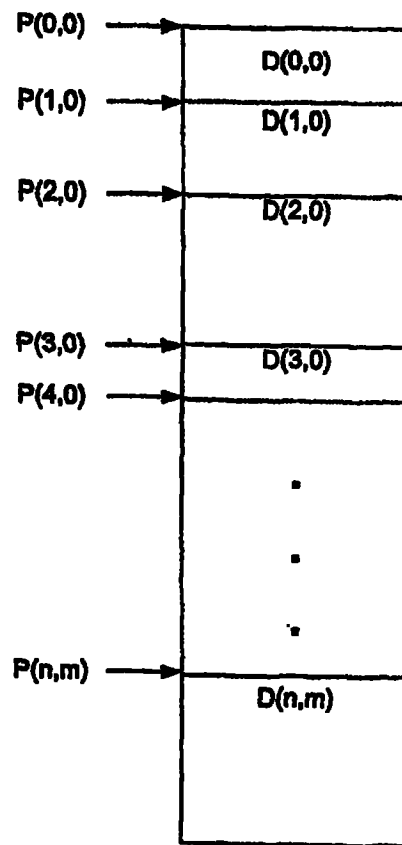


图 13

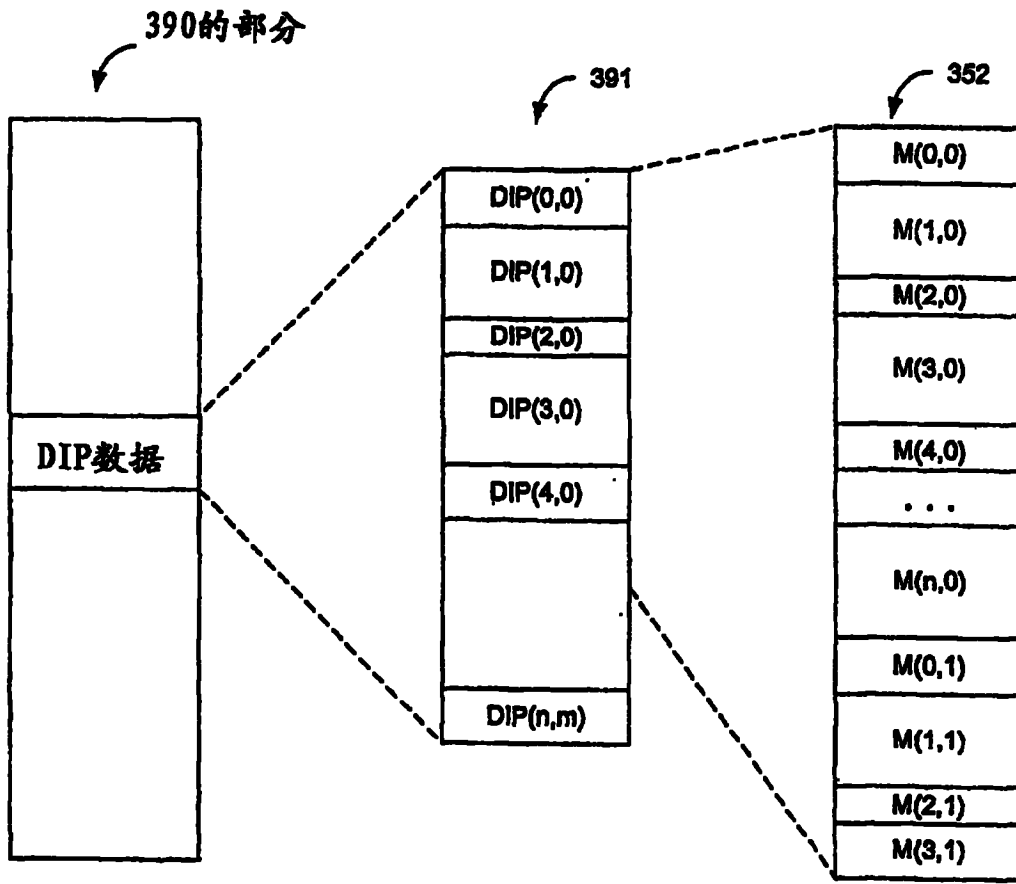


图 12

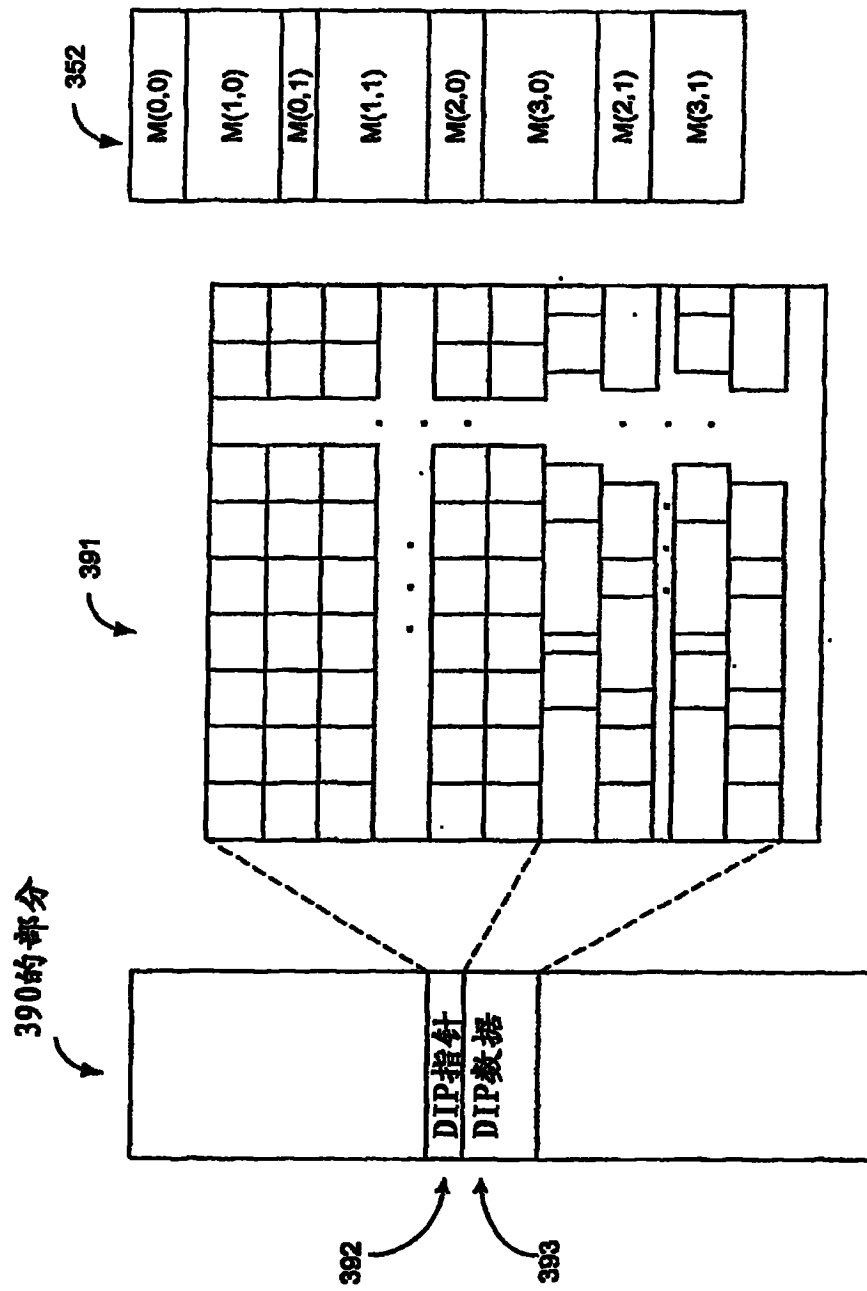


图 14

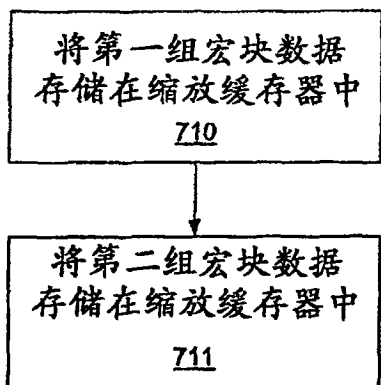


图 15

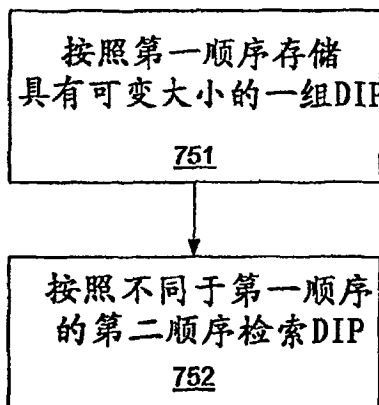


图 19

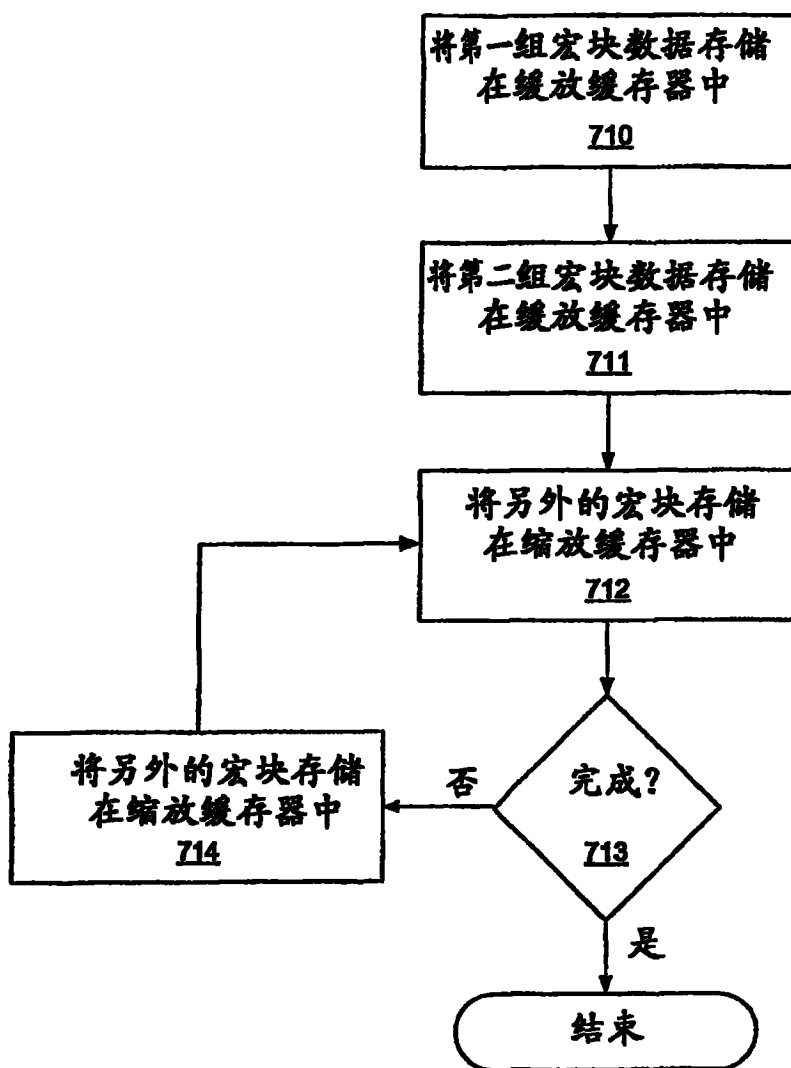


图 16

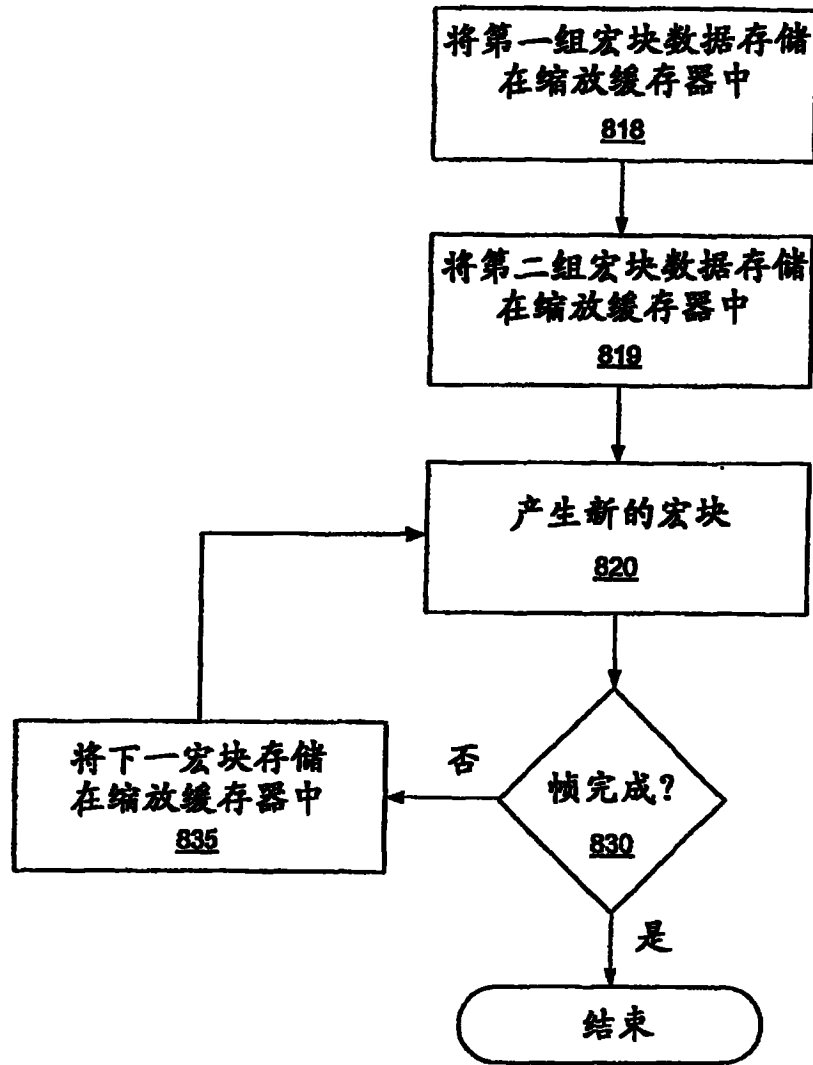


图 17

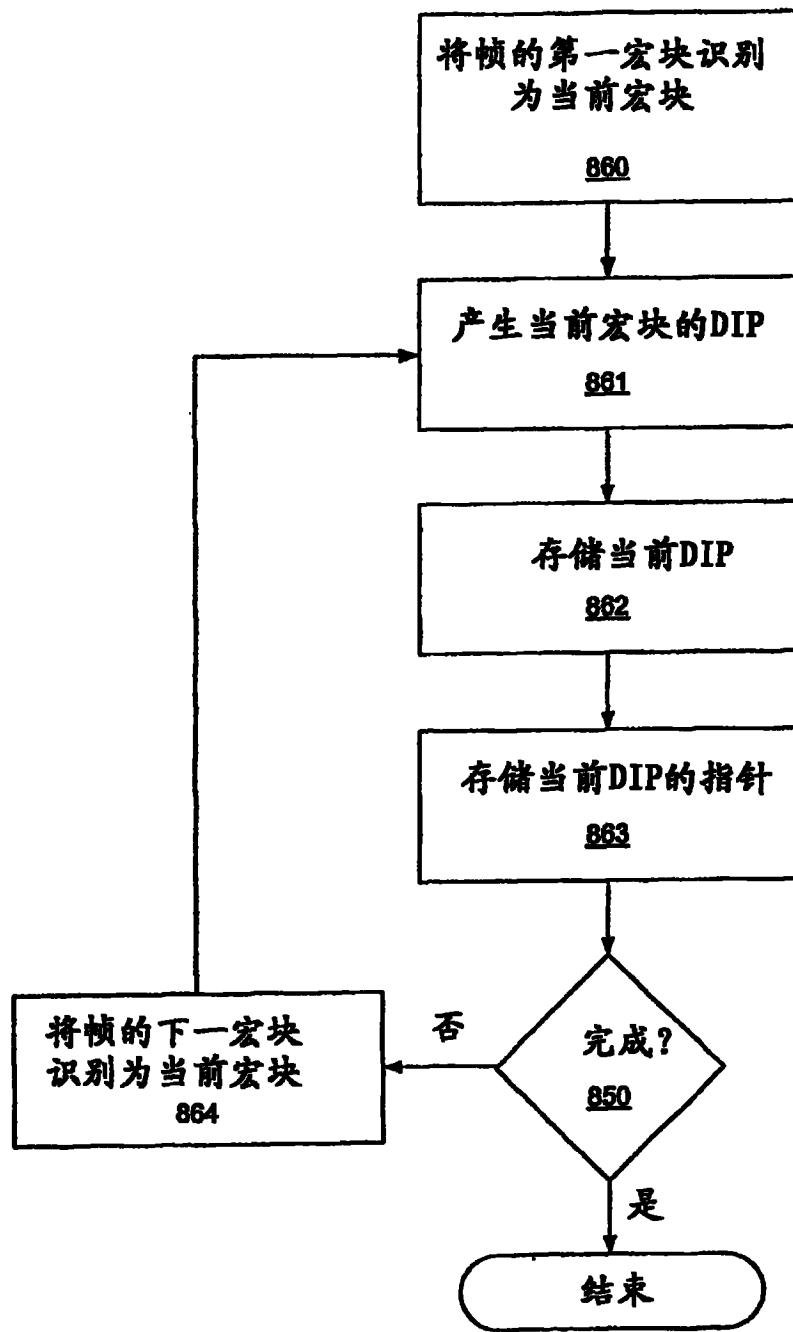


图 18

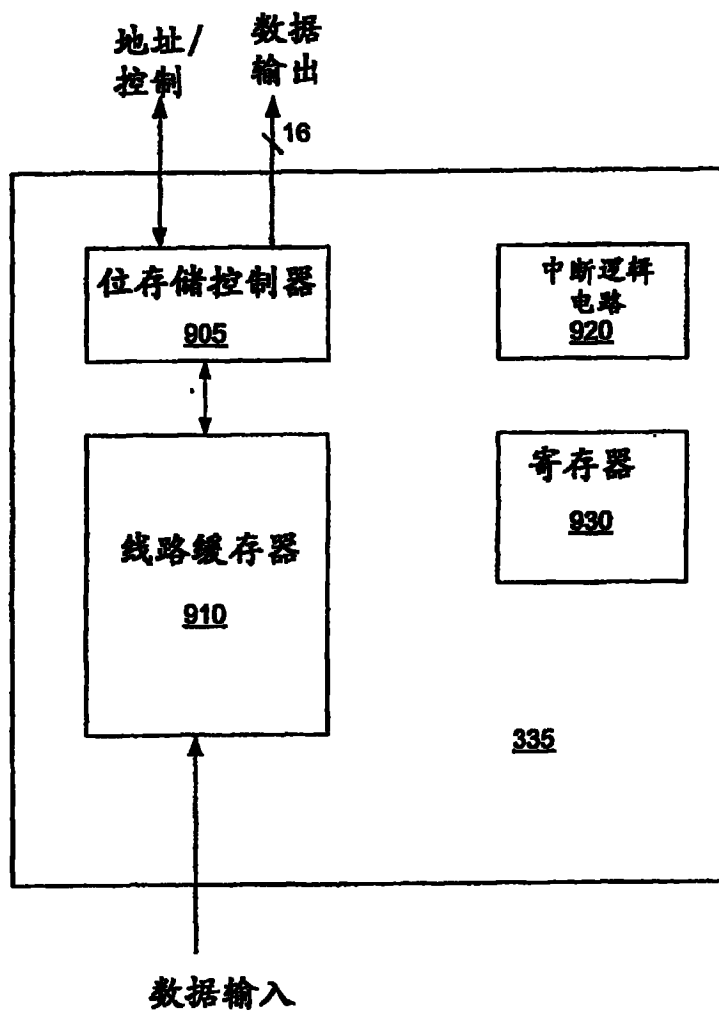


图 20

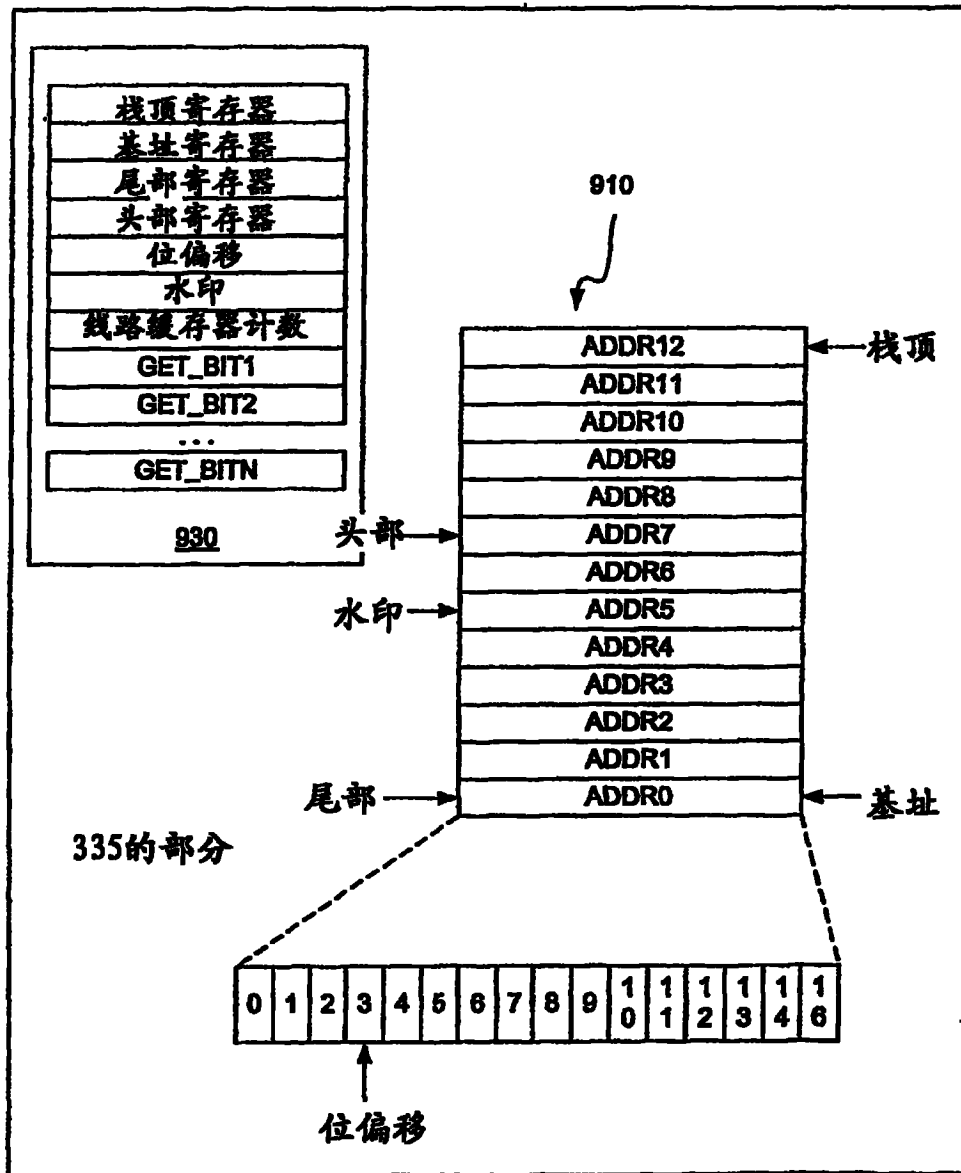


图 21

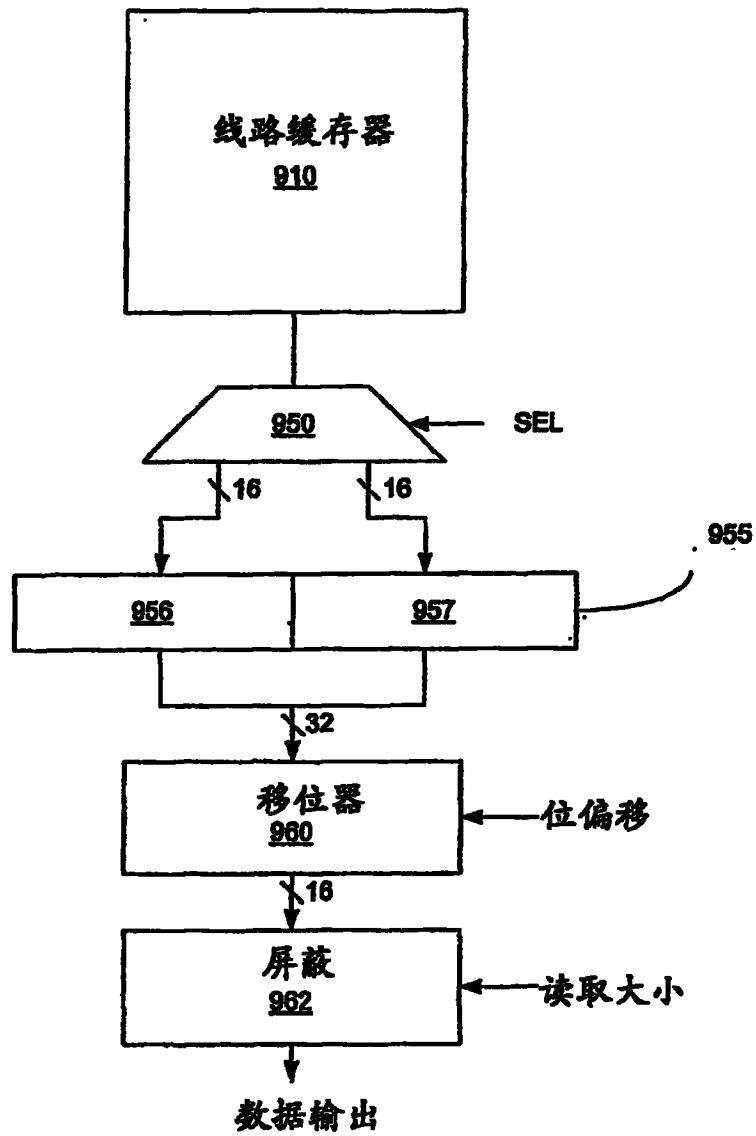


图 22

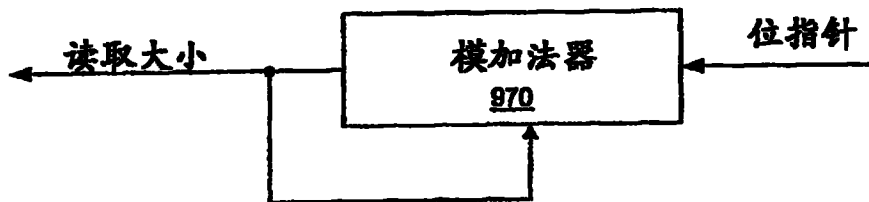


图 23

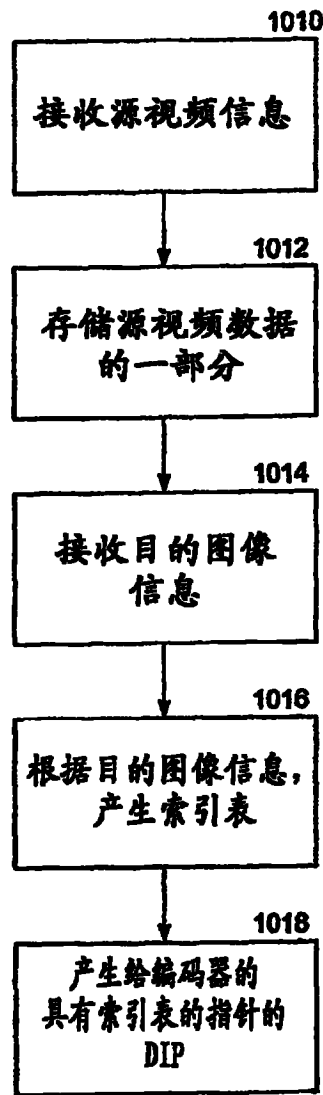


图 24

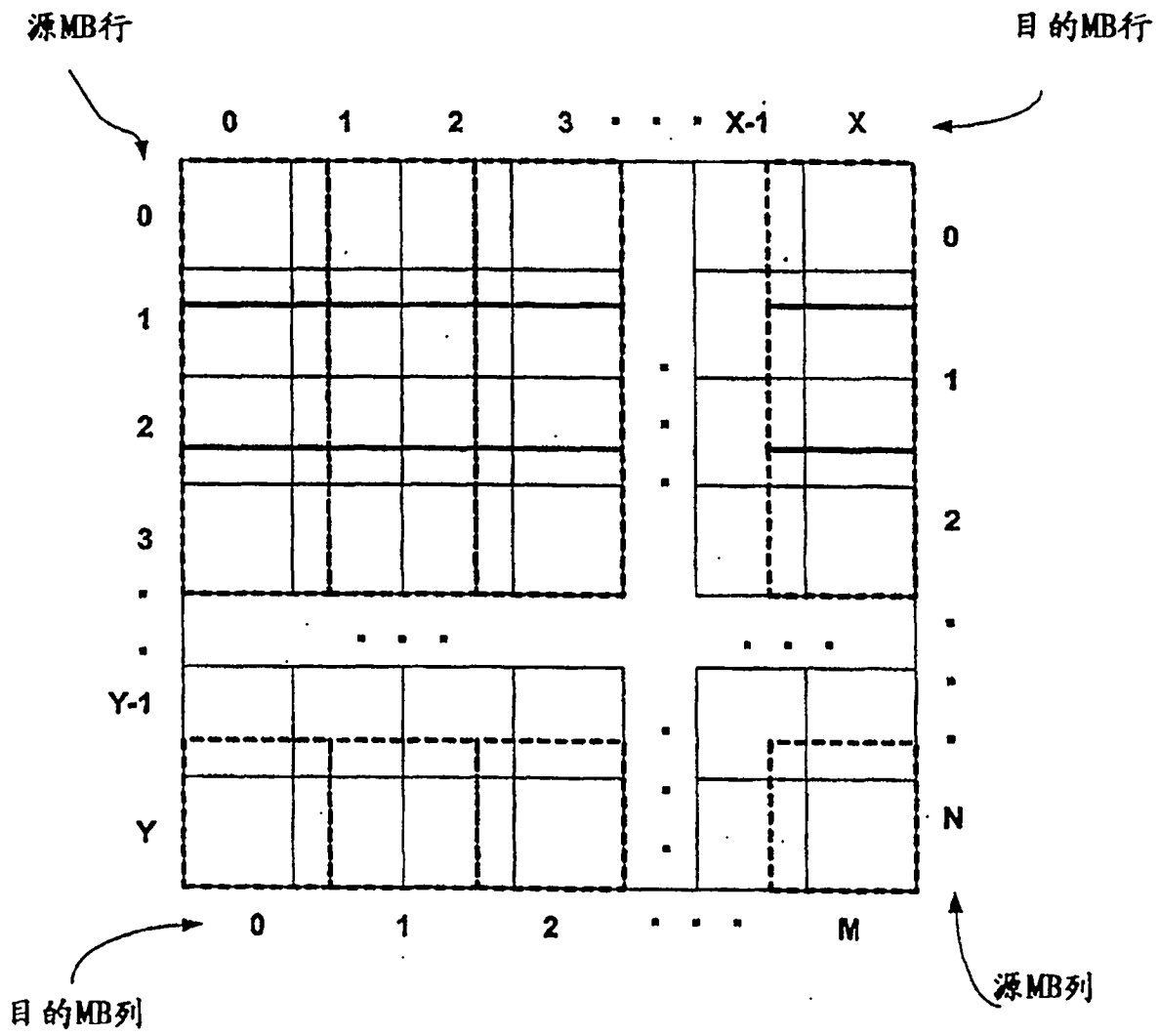


图 25

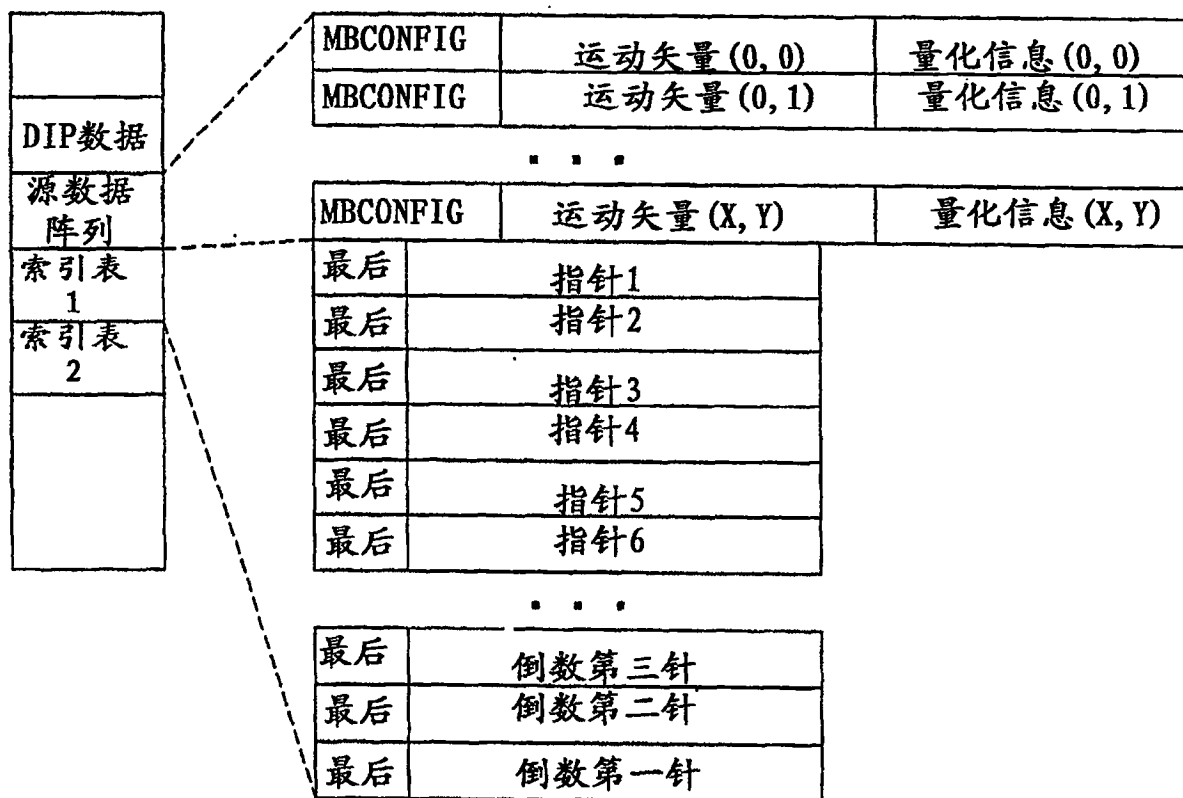


图 26

条目位置	宏块	最后
1	(0,0)	0
2	(0,1)	0
3	(1,0)	0
4	(1,1)	1
5	(0,1)	0
6	(0,2)	0
7	(1,1)	0
8	(1,2)	1
...		
LAST - 3	(M-1,N-1)	0
LAST - 2	(M,N-1)	0
LAST - 1	(M-1,N)	0
LAST	(M,N)	1

图 27

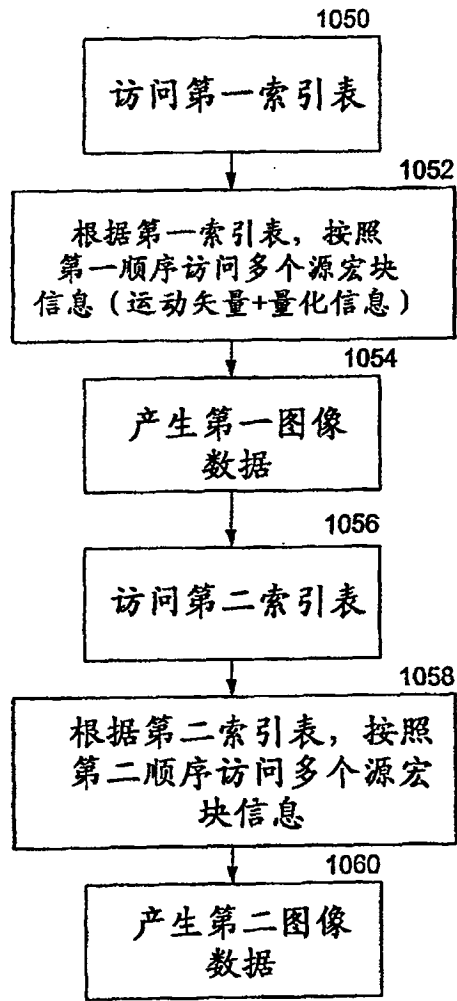


图 28

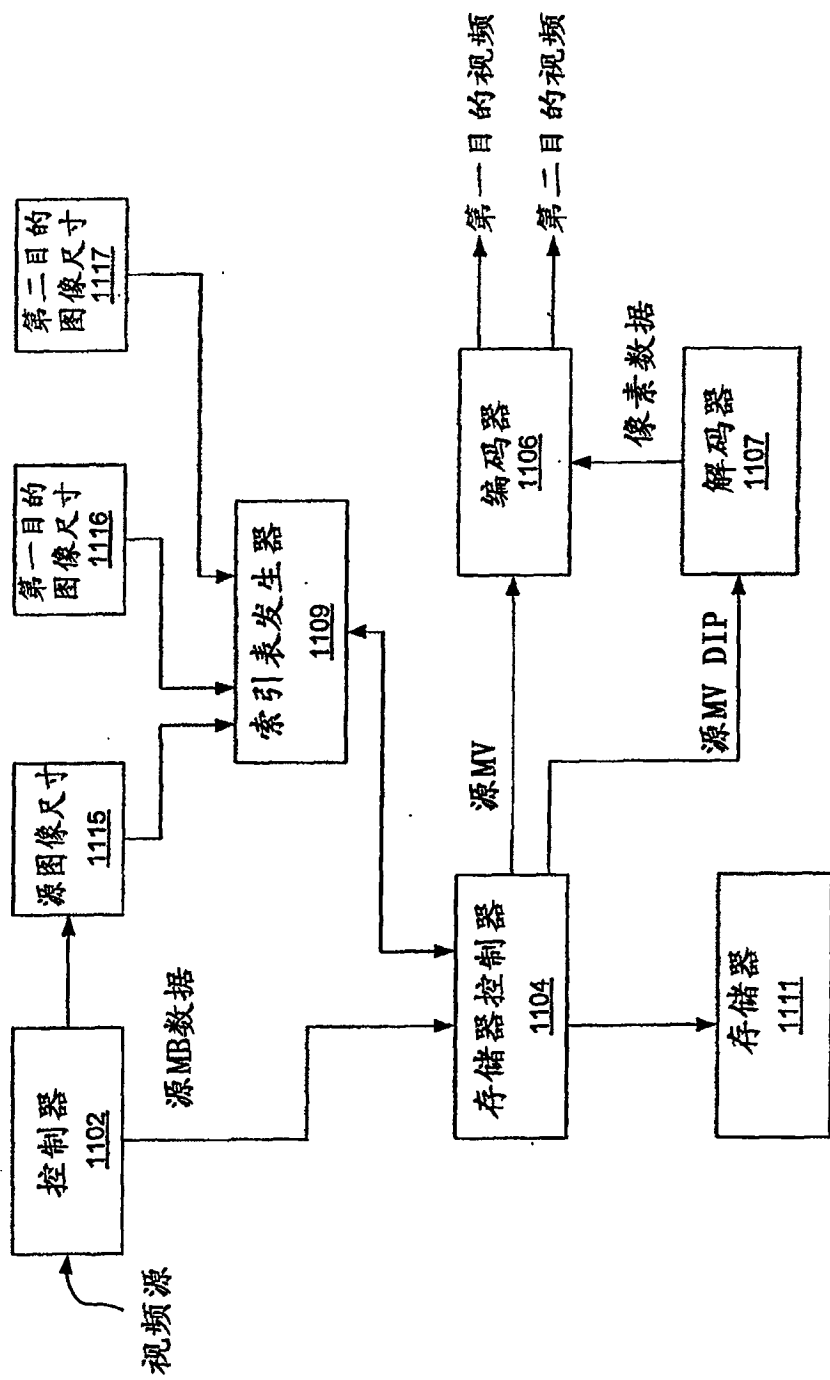


图 29