



(12) 发明专利

(10) 授权公告号 CN 101945280 B

(45) 授权公告日 2012. 09. 05

(21) 申请号 201010286538. 5

(22) 申请日 2005. 06. 27

(30) 优先权数据

60/583, 447 2004. 06. 27 US

11/070, 778 2005. 03. 01 US

(62) 分案原申请数据

200510079996. 0 2005. 06. 27

(73) 专利权人 苹果公司

地址 美国加利福尼亚

(72) 发明人 童歆 聂小春

(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 邹姗姗

(51) Int. Cl.

H04N 7/26(2006. 01)

H04N 7/32(2006. 01)

(56) 对比文件

CN 1127578 A, 1996. 07. 24, 全文.

CN 1131877 A, 1996. 09. 25, 全文.

WO 2003/045069 A2, 2003. 05. 30, 全文.

US 6529634 B1, 2003. 03. 04, 全文.

US 5576767 A, 1996. 11. 19, 全文.

审查员 于利娜

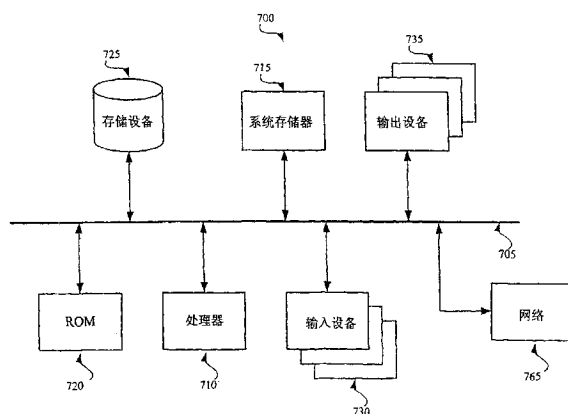
权利要求书 2 页 说明书 14 页 附图 7 页

(54) 发明名称

对编码视频数据选择编码类型和预测模式的方法和设备

(57) 摘要

本发明涉及对编码视频数据选择编码类型和预测模式。在本发明中,提供了一种用于确定视频帧的宏块的编码类型和预测模式的方法。在一些实施方式中,一般的方法 1) 基于容易计算的宏块的属性利用编码类型选择算法选择最初为宏块考虑的编码类型,即 16×16 或 4×4 , 2) 如果在步骤 1 选择了 16×16 编码类型,则基于失真阈值,利用传统的方法或者改进的 16×16 预测模式搜索算法,考虑可以用在宏块上的四种 16×16 预测模式, 3) 如果在步骤 1 中选择了 4×4 编码类型,则基于预测模式之间的位置关系,利用传统的方法或者改进的 4×4 预测模式搜索算法,选择 4×4 预测模式用于宏块的 16 个 4×4 模块的每个。



1. 一种用于编码视频图片中的像素块的方法,其中,所述像素块包括多个子块,所述方法包括:

针对所述像素块中的每个子块,使用第一预测模式来计算失真值,该失真值基于从所述像素块的边界像素组导出的预测值组;

在所述像素块的所有子块的所有所述失真值都满足预定阈值失真值时,使用所述第一预测模式来编码所述像素块;以及

当不是所述像素块的所有子块的所有所述失真值都满足预定阈值失真值时,选择第二预测模式。

2. 根据权利要求1所述的方法,其中,所述失真值是第一失真值,并且所述预测值组是第一预测值组,所述方法还包括:

在确定所述第一失真值不全满足所述预定阈值失真值时,使用所述第二预测模式来计算所述像素块中每个子块的第二失真值;以及

当所述像素块的所有子块的第二失真值满足所述预定阈值失真值时,使用所述第二预测模式来编码所述像素块。

3. 根据权利要求2所述的方法,其中,在确定所述第二失真值不全满足所述预定阈值失真值时,使用第三预测模式来编码所述像素块。

4. 根据权利要求3所述的方法,其中,所述第三预测模式指定的所述像素块的分区尺寸小于所述第二预测模式指定的分区尺寸。

5. 根据权利要求1所述的方法,其中,所述边界像素组包括所述像素块的与所述子块垂直或水平对准的边界上的像素。

6. 根据权利要求1所述的方法,其中,针对每个子块使用所述第一预测模式来计算失真值包括:

基于所述第一预测模式来识别所述像素块的边界像素组;以及

通过使用所识别出的所述像素块的边界像素组,导出子块的预测值组。

7. 根据权利要求1所述的方法,其中,所述像素块的子块中的像素行的所述预测值组包括子块中的所述像素行的同一行的所述像素块的最左列中的像素。

8. 根据权利要求1所述的方法,其中,所述像素块的子块中的像素列的所述预测值组包括子块中的所述像素列的同一列的所述像素块的最上一行中的像素。

9. 根据权利要求1所述的方法,其中,所述预测值组是从所述像素块的最左列和最上一行的像素的平均值导出的。

10. 根据权利要求1所述的方法,其中,所述像素块包括 16×16 个像素,其中所述像素块中的每个子块包括 4×4 个像素。

11. 一种用于编码视频图片中的像素块的设备,其中,所述像素块包括多个子块,所述设备包括:

用于针对所述像素块中的每个子块,使用第一预测模式来计算失真值的装置,该失真值基于从所述像素块的边界像素组导出的该子块的预测值组;

用于在所述像素块的所有子块的所有所述失真值都满足预定阈值失真值时,使用所述第一预测模式来编码所述像素块的装置;以及

用于当不是所述像素块的所有子块的所有所述失真值都满足预定阈值失真值时,选择

第二预测模式的装置。

12. 根据权利要求 11 所述的设备,其中,所述失真值是第一失真值,并且所述预测值组是第一预测值组,所述设备还包括:

用于在确定所述第一失真值不全满足所述预定阈值失真值时,使用所述第二预测模式来计算所述像素块中每个子块的第二失真值的装置;以及

用于当所述像素块的所有子块的第二失真值满足所述预定阈值失真值时,使用所述第二预测模式来编码所述像素块的装置。

13. 根据权利要求 12 所述的设备,还包括:用于在确定所述第二失真值不全满足所述预定阈值失真值时,使用第三预测模式来编码所述像素块的装置。

14. 根据权利要求 13 所述的设备,其中,所述第三预测模式指定的所述像素块的分区尺寸小于所述第二预测模式指定的分区尺寸。

15. 根据权利要求 11 所述的设备,其中,所述边界像素组包括所述像素块的与所述子块垂直或水平对准的边界上的像素。

16. 根据权利要求 11 所述的设备,其中,针对每个子块使用所述第一预测模式来计算失真值包括:

基于所述第一预测模式来识别所述像素块的边界像素组;以及

通过使用所识别出的所述像素块的边界像素组,导出子块的预测值组。

17. 根据权利要求 11 所述的设备,其中,由所述像素块的子块中的像素行的所述预测值组包括子块中的所述像素行的同一行的所述像素块的最左列中的像素。

18. 根据权利要求 11 所述的设备,其中,由所述像素块的子块中的像素列的所述预测值组包括子块中的所述像素列的同一列的所述像素块的最上一行中的像素。

19. 根据权利要求 11 所述的设备,其中,所述预测值组是从所述像素块的最左列和最上一行中的像素的平均值导出的。

20. 根据权利要求 11 所述的设备,其中,所述像素块包括 16×16 个像素,其中所述像素块中的每个子块包括 4×4 个像素。

对编码视频数据选择编码类型和预测模式的方法和设备

[0001] 本发明是申请号为 200510079996.0、申请日为 2005 年 6 月 27 日、发明名称为“对编码视频数据选择编码类型和预测模式”的申请的分案申请。

[0002] 技术领域

[0003] 本发明涉及对于编码视频数据选择编码类型和预测模式。

[0004] 背景技术

[0005] 视频流是由一系列视频帧所组成的,其中每帧由多个宏块组成。每个宏块典型地是 16×16 的像素阵列,尽管其它大小的宏块也是可能的。视频编解码器(压缩器-解压缩器)是压缩算法的软件、硬件或者软件和硬件结合的实现,其中所述的压缩算法用来编码/压缩以及解码/解压缩视频数据流以减小流的大小,为了能更快地传输以及更小的存储空间。虽然有损耗,在压缩视频流的二进制数据的同时,视频编解码器试图保持图像的质量。普通的视频编解码器的实例包括 WMV、RealVideo 以及压缩标准,诸如 MPEG-2、MPEG-4、H. 261、H. 263 和 H. 264 的实现。

[0006] 在 H. 264 压缩标准下,视频帧的宏块可以在内部编码为 16×16 的像素阵列,该阵列的像素值是利用以前编码的宏块计算的值预测的。 16×16 的宏块也可以内部编码为 16 个 4×4 的像素阵列,其中每个 4×4 的阵列中的像素值是利用以前编码的 4×4 的阵列计算的值预测的。存在 4 种可能的 16×16 阵列(亮度(luma)模块)内部预测模式以及 9 种可能的 4×4 阵列(亮度模块)内部预测模式。

[0007] 同样,在编码宏块中,必需做出两个决定(选择):1) 是否该宏块将被编码为 16×16 阵列(称作 16×16 编码)或者被编码为 16 个 4×4 阵列(称作 4×4 编码),以及 2) 用于编码宏块的预测模式。例如,如果决定将宏块编码为 16×16 阵列,也必须决定将使用 16×16 阵列的四个预测模式中的哪一个。如果决定将宏块编码为 16 个 4×4 阵列,也必须决定对于 16 个 4×4 阵列的每个,将使用 4×4 阵列的 9 个预测模式中的哪一个。在这里步骤 1 称作编码类型选择,并且步骤 2 称作预测模式选择。

[0008] 利用成本函数来做出编码类型选择以及预测模式选择。例如,成本函数典型地用于决定是否将宏块编码为 16×16 阵列或编码为 16 个 4×4 阵列,其中选择具有最低成本的编码的类型(16×16 或 4×4 编码)。成本典型地等于失真或者失真的加权平均加上由预测模式生成的比特的数量的估计,其中失真和/或比特数量的增加会增加成本。失真反映了原始像素值和预测的(或编码的)值之间的差异,并且失真可以以不同的方法来测量。例如,失真可以测量为原始像素值和预测的(或编码的)值之间的绝对差值的和。

[0009] 选择宏块的最优编码类型(16×16 或 4×4 编码)以及最优预测模式的穷举搜索方法包括确定宏块中所有 4 个 16×16 预测模式的以及 16 个 4×4 模块的 9 种 4×4 预测模式的所有组合的成本,其中选定成本最低的 16×16 预测模式或 4×4 预测模式的特定组合。对于每个宏块,穷举搜索方法需要考虑 4×4 预测模式的 9^{16} 不同的组合,使得穷举搜索方法实际上不可实行。

[0010] 同样,典型地执行下列的操作以确定宏块的编码类型和预测模式:

[0011] 1) 计算所有 4 种可能 16×16 预测模式的成本。

[0012] 2) 对于每个 16 个 4×4 模块, 选择具有最小成本的预测模式 (在 9 种预测模式中), 然后计算结果的组合的总成本 (例如, 16 种确定的成本的和)。

[0013] 3) 将在步骤 1 确定的成本与在步骤 2 确定的成本进行对比, 并选择最低的一个。该选择提供了编码的类型选择以及预测模式选择。

[0014] 然而传统的方法依然包括确定 4×4 预测模式的 9×16 种不同的组合的成本加上 4 个 16×16 预测模式的成本。

发明内容

[0015] 在一些实施方式中, 提供了一种更有效的方法来确定视频帧中宏块的编码类型的选择以及预测模式的选择。在一些实施方式中, 一般的方法 1) 利用编码类型选择算法 (基于容易计算的宏块的属性) 选择最初为宏块考虑的编码类型 (16×16 或 4×4), 2) 如果在步骤 1 选择了 16×16 编码类型, 则基于失真阈值, 利用传统的方法或者改进的 16×16 预测模式搜索算法, 考虑可以用在宏块上的四种 16×16 预测模式, 3) 如果在步骤 1 中选择了 4×4 编码类型, 基于预测模式之间的位置关系, 利用传统的方法或者改进的 4×4 预测模式搜索算法, 选择用于宏块的 16 个 4×4 模块的每个的 4×4 预测模式。

附图说明

[0016] 图 1 是选择像素宏块的编码类型和预测模式的一般方法的流程图。

[0017] 图 2 是用于选择像素宏块的编码类型的编码类型选择方法的流程图。

[0018] 图 3 是用于选择 16×16 像素宏块的预测模式的预测模式搜索方法的流程图。

[0019] 图 4 描述 4×4 阵列的 9 种预测模式之间的逻辑 / 位置关系的概念图表。

[0020] 图 5 是用于选择宏块 4×4 阵列的预测模式的 4×4 预测模式搜索方法的流程图。

[0021] 图 6 是用于选择宏块 4×4 阵列的预测模式的可替换的 4×4 预测模式搜索方法的流程图。

[0022] 图 7 表示计算机系统, 利用该计算机系统实施本发明的一些实施方式。

具体实施方式

[0023] 序列号为 60/583447, 于 2004 年 6 月 27 日公开的美国临时专利申请“编码和解码图像”在此通过引用清楚地结合。

[0024] 在以下的描述中, 为了解释, 阐述了多个细节。但是, 本领域的普通技术人员将理解不用这些特定的细节也可以实现本发明。在其它的实例中, 为了不使不必要的细节描述模糊本发明的说明, 已知的结构和设备以框图的形式表示。

[0025] 在一些实施方式中, 本发明提供了一种更有效的方法来确定视频帧中宏块的编码类型的选择以及预测模式的选择。在一些实施方式中, 一般的搜索算法执行以下操作

[0026] 1) 利用编码类型选择算法 (基于容易计算的宏块的属性) 选择最初为宏块考虑的编码类型 (16×16 或 4×4)。编码类型选择算法在下面部分 I 中进行讨论。

[0027] 2) 如果在步骤 1 选择了 16×16 编码模式, 基于失真阈值, 利用传统的方法或者改进的 16×16 预测模式搜索算法, 考虑 (即, 测试) 可以用在宏块上的四种 16×16 预测模式。注意在 16×16 预测模式搜索算法的一种特定情形中, 也可以选择 4×4 编码类型。该步骤

在以下部分 II 中进行讨论。

[0028] 3) 如果在步骤 1 中选择了 4×4 编码类型, 基于预测模式之间的位置关系 (在以下部分 III 中对其进行讨论), 利用传统的方法或者改进的 4×4 预测模式搜索算法, 选择 4×4 预测模式用于宏块的每个 16 个 4×4 模块。

[0029] 注意在搜索算法的步骤 1 中, 如果编码类型选择算法选择 16×16 编码类型用于初始考虑, 最终选择的编码类型可以是 16×16 编码类型或 4×4 编码类型。相反, 如果编码类型选择算法选择 4×4 编码类型用于初始考虑, 则将 4×4 编码类型有效地选择为宏块的编码类型。如果选择 16×16 编码类型用于初始考虑, 传统的方法或者改进的 16×16 预测模式搜索算法用于确定宏块的预测模式。如果选择 4×4 编码类型用于初始考虑, 则传统的方法或者改进的 4×4 预测模式搜索算法用于确定宏块的预测模式。同样, 根据由搜索算法选择的用于初始考虑的编码类型, 使用不同的搜索算法 (传统的或改进的) 来确定预测模式。

[0030] 图 1 是选择像素宏块的编码类型和预测模式的一般方法 100 的流程图。在一些实施方式中, 通过设置来编码视频帧的软件和 / 或硬件实现一般方法 100。当其接收 (在 102) 像素宏块时, 该方法开始。在一些实施方式中, 宏块是 16×16 像素阵列, 而在其它的实施方式中, 该宏块是不同尺寸的像素阵列。在一些实施方式中, 将该宏块分成 4 个 8×8 阵列或 16 个 4×4 阵列, 而在其它的实施方式中, 将宏块分成不同尺寸的块 / 阵列。

[0031] 利用编码类型选择方法 200 (以下参照附图 2 进行讨论), 该方法然后选择 (在 105) 最初为宏块考虑的编码类型 (16×16 或 4×4)。该方法然后确定 (在 110) 是否已经选择了 4×4 编码类型。如果是, 则该方法随后利用传统的方法、改进的 4×4 预测模式搜索方法 500 (以下参照附图 5 进行讨论), 或者可选 4×4 预测模式搜索方法 600 (以下参照附图 6 进行讨论), 选择 (在 115) 宏块的 16 个 4×4 模块的每个的预测模式。

[0032] 如果该方法确定 (在 110- 否) 没有选择 4×4 编码类型, 则该方法选择 (在 120) 16×16 预测模式 (利用传统的方法或改进的 16×16 预测模式搜索方法 300, 以下将参照附图 3 进行讨论) 或 4×4 编码类型。该方法然后确定 (在 125) 是否已经选择了 4×4 编码类型 (在 120)。如果是, 该方法继续步骤 115, 其中在步骤 115 选择宏块的 16 个 4×4 模块的每个的预测模式。如果该方法确定 (在 125- 否) 没有选择 4×4 编码类型, 这表示已经选择了特定的 16×16 预测模式。然后该方法结束。

[0033] 部分 I : 编码类型选择算法

[0034] 该部分描述了一般搜索算法的步骤 1 的编码类型选择算法。编码类型选择算法选择最初为宏块考虑的编码类型 (16×16 或 4×4) 并从而确定哪个预测模式 (16×16 模式或 4×4 模式) 是最初为宏块考虑的。编码类型选择算法利用易于确定的宏块的属性。

[0035] 编码类型选择算法执行下列操作:

[0036] 1) 将宏块分成 4 个 8×8 模块。

[0037] 2) 计算每个 8×8 模块的方差, 并将其与预定的阈值进行对比。例如, 可以将预定的阈值设置成等于帧中所有的 8×8 模块平均方差, 在所述帧中包括宏块。可以确定 8×8 模块的方差, 例如通过 a) 确定模块中的所有像素值的平均值, b) 为模块中的每个像素确定: $(\text{像素值} - \text{平均值})^2$, 以及 c) 对在步骤 b 确定的所有值求和 (即, 对模块中的所有像素的确定的值求和)。

[0038] 3) 确定宏块中具有低于预定阈值的方差的 8×8 模块的数量 (其中方差是用于确定是否将宏块编码为 16×16 模块或 4×4 模块的属性)。

[0039] 4) 如果所有的 4 个 8×8 模块具有低于预定阈值的方差, 则为初始考虑选择 16×16 编码类型。注意, 如果 16×16 编码类型被选择为初始考虑, 则最终选择的编码类型可以是 16×16 编码类型或 4×4 编码类型。可以利用传统的方法或改进的 16×16 预测模式搜索算法 (如以下部分 II 中所述), 选择用在宏块上的 16×16 预测模式。

[0040] 5) 如果 4 个 8×8 模块的任一个具有超过预定阈值的方差, 则选择 4×4 编码类型。同样, 利用传统的方法或者改进的 4×4 预测模式搜索算法 (如以下部分 II 所述), 来确定将用在宏块的 16 个 4×4 模块的每个上的 4×4 预测模式。

[0041] 注意当用在编码进程的量化步骤中使用的量化参数值相对高时 (例如, 高于 34), 编码类型选择算法的操作稍微偏离上述的说明。在这些实施方式中, 预定的阈值由取决于量化参数值 (QP) 的因子来依比例确定, 使得当量化参数值增加时, 预定的阈值就增加。在一些实施方式中, 定比的因子由以定比的方程的形式确定: $(1+k(QP-34))$, 其中 k 是常量。在其它的实例中, 定比的因子由另一个定比方程确定。预定的阈值乘以被确定的定比因子以生成成比例的阈值。

[0042] 在这些实施方式中, 如果 4 个 8×8 模块的任一个具有超过成比例的阈值的方差, 则将宏块编码为 16 个 4×4 模块。可以利用传统的方法或本发明的方法 (诸如改进的 4×4 预测模式搜索算法) 来做出选择。然而, 如果确定宏块的所有的 4 个 8×8 模块具有低于成比例的阈值的方差, 可以将宏块编码为 16×16 模块或编码为 16 个 4×4 模块。可以利用传统的方法或改进的 16×16 预测模式搜索算法 (如以下部分 II 所述), 选择 16×16 预测模式用在宏块上。如果选择 4×4 编码类型优于已选定的 16×16 预测模式, 则可以利用传统的方法或本发明的方法 (诸如改进的 4×4 预测模式搜索算法) 来选择 4×4 预测模式。

[0043] 附图 2 是用于选择像素宏块的编码类型的编码类型选择方法 200 的流程图。在一些实施方式中, 方法 200 是由设计用来编码视频帧的软件和 / 硬件来实现的。在一些实施方式中, 编码类型选择方法 200 包括附图 1 的步骤 105。方法 200 从确定宏块的阈值方差 (在 205) 开始。在一些实施方式中, 将阈值方差设置成等于帧中的所有 8×8 模块的平均方差, 在所述的帧中包括有宏块 (如上所述)。在另外的实施方式中, 当用在编码进程的量化步骤的量化参数值相对高时, 预定的阈值方差由取决于量化参数值的因子来依比例确定 (如上所述)。

[0044] 该方法然后将宏块分成 4 个 8×8 模块 (在 210) 并且确定 4 个 8×8 模块的每个的方差 (在 215)。该方法然后确定是否所有的 4 个 8×8 模块的方差低于阈值方差 (在 220)。如果是, 则该方法选择宏块的 16×16 编码类型 (在 225), 并且结束。如果不是, 则该方法选择宏块的 4×4 编码类型 (在 230), 并且结束。

[0045] 部分 II: 16×16 预测模式搜索算法

[0046] 在一般的搜索算法的步骤 1 之后, 如果 16×16 编码类型已被选择为初始考虑, 则然后对 16×16 预测模式进行测试。该选择描述了一般搜索算法的步骤 2。在 H. 264 编解码标准下, 存在四个可能的 16×16 内部预测模式来从: DC、V (垂直)、H (水平) 以及 P (平面的) 中选择。在一些实施方式中, 使用传统的方法来选择 16×16 预测模式 (例如, 通过确定 4 个 16×16 预测模式的每个的成本并选择最低成本的模式)。在其它的实施方式中, 基

于由预测模式导致的失真，•使用改进的 16×16 预测模式搜索算法来考虑（即测试）宏块的 16×16 预测模式。

[0047] 在一些实施方式中， 16×16 预测模式搜索算法使用传统的方法来确定由特定的预测模式导致的失真，所述的特定的预测模式用在了当前处理的宏块中。在传统的方法下，失真的确定是由：1) 从邻接的以前编码的宏块检索像素值，2) 从检索到的像素值获得新的值，3) 利用获得的值计算当前处理的宏块的预测值，以及 4) 确定当前处理的宏块的原始像素值和计算的预测值之间的失真。利用传统的方法来确定失真是耗费时间的。

[0048] 在其它的实施方式中， 16×16 预测模式搜索算法利用更快的估计方法来确定由特定的预测模式导致的失真，所述的特定的预测模式用在了当前处理的宏块中。该估计方法提供对预测值的估计（评估）并且然后确定被估计的预测值的失真，其中所述的预测值是由特定的预测模式生成的。

[0049] 在估计方法下，利用来自于当前处理的宏块自身的像素值来确定预测值，假定当前处理的宏块在边界处的像素值与邻接的宏块的邻近的像素值相似。由于所使用的来自于当前处理宏块的像素值不同于来自于其它宏块的像素值，这减小了检索的时间，所以利用估计方法确定失真耗费的时间较少。利用估计方法显著减小了需要确定精度损失较低的失真值的时间。

[0050] 在一些实施方式中，估计方法的伪码如下：

```
[0051]  choose16×16mode()
[0052]  {
[0053]      step1:
[0054]          //DC 预测
[0055]          pred = 当前宏块的左列和上排的像素平均
[0056]          foreach 4×4block
[0057]          {
[0058]              SAD_DC = sad(4×4block, pred) ;
[0059]              if (SAD_DC > T)
[0060]              {
[0061]                  bail and goto step2;
[0062]              }
[0063]          }
[0064]          choose 16×16DC mode and return
[0065]  step2:
[0066]          // 垂直预测
[0067]          foreach 4×4block
[0068]          {
[0069]              SAD_V = 0 ;
[0070]              foreach column in the 4×4block
[0071]              {
[0072]                  pred = 当前宏块的该列的上排的像素值
```

```

[0073]             SAD_V+ = sad(4×4block column, pred) ;
[0074]         }
[0075]         SAD_V >>= 4 ;
[0076]         if (SAD_V > T)
[0077]         {
[0078]             bail and goto step3 ;
[0079]         }
[0080]     }
[0081]     choose 16×16V and return
[0082] Step 3 :
[0083]     // 水平预测
[0084]     foreach 4×4block
[0085]     {
[0086]         SAD_H = 0 ;
[0087]         foreach row in the 4×4block
[0088]         {
[0089]             pred = 当前宏块的该排的左列的像素值
[0090]             SAD_H+ = sad(4×4block row, pred) ;
[0091]         }
[0092]         SAD_H >>= 4 ;
[0093]         if (SAD_H > T)
[0094]         {
[0095]             choose 16×16planar and 4×4modes and return
[0096]         }
[0097]     }
[0098]     choose 16×16H and return
[0099] }
[0100] 不论用什么方法计算特定预测模式的失真, 16×16 预测模式搜索算法执行下列操作 :
[0101] 1) 测试 16×16DC 预测模式 :
[0102] a) 确定由 16×16DC 预测模式导致的宏块的每个 4×4 模块中的失真。
[0103] b) 如果每个 4×4 模块中的失真低于预定的阈值, 则选择 16×16DC 预测模式作为该宏块的预测模式, 并且 16×16 预测模式搜索算法结束。
[0104] c) 如果任一个 4×4 模块中的失真超过预定的阈值, 则 16×16 预测模式搜索算法继续到步骤 2。
[0105] 2) 测试 16×16V 预测模式 :
[0106] a) 确定由 16×16V 预测模式导致的宏块的每个 4×4 模块中的失真。
[0107] b) 如果每个 4×4 模块中的失真低于预定的阈值, 则选择 16×16V 预测模式作为该宏块的预测模式, 并且 16×16 预测模式搜索算法结束。

```

[0108] c) 如果任一个 4×4 模块中的失真超过预定的阈值, 则 16×16 预测模式搜索算法继续到步骤 3。

[0109] 3) 测试 $16 \times 16H$ 预测模式:

[0110] a) 确定由 $16 \times 16H$ 预测模式导致的宏块中每个 4×4 模块的失真。

[0111] b) 如果每个 4×4 模块的失真低于预定的阈值, 则选择 $16 \times 16H$ 预测模式作为该宏块的预测模式, 并且 16×16 预测模式搜索算法结束。

[0112] c) 如果任一个 4×4 模块中的失真超过预定的阈值, 则 16×16 预测模式搜索算法考虑 $16 \times 16P$ 预测模式以及 4×4 编码类型。在一些实施方式中, 通过利用 4×4 预测模式搜索算法 (在以下部分 III 讨论), 使得两个上述选项之间的选择变得容易。在其它的实施方式中, 利用传统的方法 (例如通过确定 $16 \times 16P$ 预测模式的成本和利用 4×4 编码类型得到的最低成本并且选择具有最低成本的选项) 来在两个上述选项之间做出选择。

[0113] 图 3 是 16×16 预测模式搜索方法 300 的流程图, 用于选择 16×16 像素宏块的预测模式。在一些实施方式中, 该方法 300 是由设置来编码视频帧的软件和 / 或硬件来实现的。在一些实施方式中, 编码类型选择方法 300 包括图 1 的步骤 120。在一些实施方式中, 16×16 预测模式搜索方法利用传统的方法或改进的估计方法来确定宏块中由特定的预测模式 (如上所述) 导致的 4×4 模块的失真。

[0114] 方法 300 从确定由 $16 \times 16DC$ 预测模式导致的宏块的 16 个 4×4 模块中的每个的失真开始 (在 305)。该方法然后确定由 $16 \times 16DC$ 预测模式导致的所有 4×4 模块中的失真是否低于预定的阈值失真 (在 310)。如果是, 就选择 $16 \times 16DC$ 预测模式作为宏块的预测模式 (在 315), 并且该方法结束。

[0115] 如果该方法 300 确定由 $16 \times 16DC$ 预测模式导致的任一个 4×4 模块中的失真不低于预定的阈值失真 (在 310- 否), 则该方法确定由 $16 \times 16V$ 预测模式导致的宏块的 16 个 4×4 模块的每个中的失真 (在 320)。该方法然后确定由 $16 \times 16V$ 预测模式导致的所有 4×4 模块中的失真是否低于预定的阈值失真 (在 325)。如果是, 就选择 $16 \times 16V$ 预测模式作为宏块的预测模式 (在 330), 并且该方法结束。

[0116] 如果方法 300 确定由 $16 \times 16V$ 预测模式导致的任一 4×4 模块中的失真不低于预定的阈值失真 (在 325- 否), 则该方法确定由 $16 \times 16H$ 预测模式导致的宏块的 16 个 4×4 模块中每个的失真 (在 335)。该方法然后确定由 $16 \times 16H$ 预测模式导致的所有 4×4 模块中的失真是否低于预定的阈值失真 (在 340)。如果是, 就选择 $16 \times 16H$ 预测模式作为宏块的预测模式 (在 345), 并且该方法结束。

[0117] 如果方法 300 确定由 $16 \times 16H$ 预测模式导致的任一 4×4 模块中的失真不低于预定的阈值失真 (在 340- 否), 则该方法在设置 $16 \times 16P$ 预测模式作为宏块的预测模式或者为宏块设置 4×4 编码类型之间做出选择 (在 350)。在一些实施方式中, 通过使用 4×4 预测模式搜索方法 500 (以下参照附图 5 讨论), 可使上述的两个选项之间的选择变得容易。在其它的实施方式中, 利用传统的方法 (如上所述) 在上述的两个选项之间做出选择。然后该方法结束。

[0118] 部分 III: 4×4 预测模式搜索算法

[0119] 在一般的搜索算法的步骤 1 之后, 如果 4×4 编码类型已选择为初始考虑, 则然后为宏块的 16 个 4×4 模块的每个选择 4×4 预测模式。该部分描述一般搜索算法的步骤 3。

在一些实施方式中,使用传统的方法来选择每个 4×4 模块的 4×4 预测模式(例如,通过确定 9 个预测模式中每个模式的成本并且选择最低成本的模式)。在其它的实施方式中,改进的 4×4 预测模式搜索算法用于选择 4×4 模块的 4×4 预测模式。

[0120] 如上所述,视频流由一系列视频帧所组成,其中每个帧由多个宏块组成。宏块典型地是 16×16 的像素阵列(尽管其它尺寸的宏块也是可能的)并被分成分区(诸如 4×4 的像素阵列的分区)。在 H. 264 编解码标准下,当内部编码帧时,存在 9 种不同的方法来编码 4×4 阵列(即,存在 9 种内部 4×4 预测模式)。这 9 种模式是:

- [0121] 1. 内部 4×4 水平
- [0122] 2. 内部 4×4 垂直
- [0123] 3. 内部 4×4 斜_下_左
- [0124] 4. 内部 4×4 斜_下_右
- [0125] 5. 内部 4×4 垂直_右
- [0126] 6. 内部 4×4 水平_下
- [0127] 7. 内部 4×4 垂直_左
- [0128] 8. 内部 4×4 水平_上

[0129] 每个 4×4 阵列只以一种预测模式来编码。典型地,将会选择导致最低成本的预测模式。成本典型地等于失真(其中失真反映了原始像素值和编码的预测值之间的差异)或者失真的加权平均和由预测模式生成的比特数(其中失真和/或比特数的增加会增加成本)。可以在所有 9 个预测模式中执行穷举搜索,以确定具有最低成本的最优的预测模式(选定的预测模式)。然而,典型地,在所有 9 个预测模式中的穷举搜索并不是有效的(耗费太多的时间和处理资源)。

[0130] 在一些实施方式中,使用了一种更有效的方法来确定像素阵列的预测模式。在一些实施方式中,使用一种搜索算法,该算法在确定像素阵列的预测模式中考考虑减小数量的预测模式。该搜索算法更有效并且产生相对低的精度损失。

[0131] 该搜索算法是基于 4×4 阵列的 9 种可能的预测模式之间的逻辑/位置关系。图 4 示出 4×4 阵列的 9 种预测模式之间的逻辑/位置关系的概念图表。如图 4 所示,8 种预测模式(0、1、3、4、5、6、7 和 8)围绕圆环设置,其中一个预测模式(2)放置在圆环的中央,预测模式由节点表示。圆环上的两个预测模式相互越接近,两个预测模式之间的逻辑/位置关系就越强。同样,圆环上两个预测模式之间彼此相距越远,两个预测模式之间的逻辑/位置关系就越弱。例如,由于预测模式 0 和 5 在圆环上相互邻接,所以这表示这两个预测模式之间存在较强的逻辑/位置关系。作为又一个实例,由于预测模式 0 和 1 是在圆环上相对的位置,这表示这两个预测模式之间存在较弱的逻辑/位置关系。预测模式 2 位于圆环的中央,表示它与其它任何一个预测模式都没有逻辑/位置关系。

[0132] 图 4 的概念图表是基于内部编码 4×4 阵列的 H. 264 编解码标准。对于当前处理的 4×4 阵列, H. 264 标准下的预测模式表示另外一个 4×4 阵列(在此称作预测阵列)的位置(相对于当前处理的 4×4 阵列),其是基于对当前处理的阵列编码的预测信息。例如,预测模式 0(垂直)表示当前处理的阵列的预测阵列是位于当前处理的阵列之上,预测模式 1(水平)表示当前处理的阵列的预测阵列是位于当前处理的阵列左侧。

[0133] 同样,可以访问预测模式之间的逻辑/位置关系的强弱(基于由与当前处理的

4×4 阵列有关的预测模式表示的位置关系)。例如,由于位于当前处理的阵列之上的预测阵列(由预测模式 0 表示)与位于当前处理的阵列左侧的预测阵列(由预测模式 1 表示)之间具有弱的逻辑/位置关系,所以预测模式 0 与预测模式 1 之间具有弱的逻辑/位置关系。弱的逻辑位置关系反映在了附图 4 中,其中预测模式 0 和 1 在圆环上相对的部分。相反,由于位于当前处理的阵列之上的预测阵列(由预测模式 0 表示)与位于当前处理的阵列上方和右侧的预测阵列(由预测模式 5 表示)之间具有强的逻辑/位置关系,所以预测模式 0 与预测模式 5 之间具有强的逻辑/位置关系。强的逻辑/位置关系反映在了附图 4 中,其中预测模式 0 和 5 在圆环上相邻。

[0134] 围绕圆环设置的 8 个预测模式 (0、1、3、4、5、6、7 和 8) 可以分为 4 组逻辑/位置相关的预测模式:

[0135] 1. 7、0 和 5(垂直_左,垂直,和垂直_右)

[0136] 2. 8、1 和 6(水平_上,水平,和水平_下)

[0137] 3. 7、3 和 8(垂直_左,斜_下_左,和水平_上)

[0138] 4. 5、4 和 6(垂直_右,斜_下_右,和水平_下)

[0139] 相同组内的预测模式相互之间具有强的逻辑/位置关系。在一些实施方式中,搜索算法最初确定每个组中的一个预测模式的成本,连同预测模式 2 的成本。搜索算法然后确定具有最低成本的该组预测模式中的预测模式。如果具有最低成本的预测模式是在特定组的预测模式中,则存在非常大的可能性是所有 9 个可能的预测模式的最低成本就在该特定的组中。同样,该搜索算法然后确定在该特定的组中具有最低成本的预测模式,具有最低成本的预测模式被选作阵列的最优的预测模式。

[0140] 例如,在一些实施方式中,搜索算法最初确定预测模式 0、1、2、3 和 4 的成本,其中预测模式 0、1、3 和 4 分别来自不同的组。如果在预测模式的组 (0, 1, 2, 3, 4) 中预测模式 0、1、3 或 4 具有最低的成本,则搜索算法随后确定预测模式的组中最低的成本,其中包括以前确定的预测模式。例如,如果搜索算法确定在预测模式的组 (0, 1, 2, 3, 4) 中,预测模式 1 具有最低的成本,则搜索算法然后确定预测模式组 (8, 1, 6) 的最低成本,具有最低成本的预测模式被选为最优的预测模式。

[0141] 在一些实施方式中,搜索算法最初考虑并比较预定的预测模式组的成本,该组包括一些但并不是全部的 9 种可能的预测模式。在一些实施方式中,预测模式的预定的组包括一个预测模式,该一个预测模式是来自于 4 个预测模式组的每个(如上参照附图 4 所述)。在以下所述的实施方式中,预测模式的预定的组包括预测模式 0、1、2、3 和 4。然而在其它的实施方式中,预测模式的预定组包括其它的预测模式。搜索算法确定了组中每个预测模式的成本,其中 C0、C1、C2、C3 和 C4 分别是预测模式 0、1、2、3 和 4 的预先确定的成本。可以利用本领域已知的方法确定每个预测模式的成本。搜索算法然后确定具有最低成本的预测模式的序数 (0, 1, 2, 3 或 4),其中变量 n 用于表示确定的序数。这些操作可以由如下的等式来表示:

[0142] $n = \text{GetIn1OfMinCost}(0, 1, 2, 3, 4)$

[0143] 其中 GetIn1OfMinCost 是函数,其确定预测模式组中(例如,0, 1, 2, 3, 和 4) 每个预测模式的成本,并且确定组中具有最低成本的 预测模式的序数。

[0144] 搜索算法然后执行下列操作:

```
[0145]   Ifn == 2
[0146]       x = 2
[0147]   Else If n == 0
[0148]       x == GetInlOfMinCost(7,0,5)
[0149]   Else Ifn == 1
[0150]       x == GetInlOfMinCost(8,1,6)
[0151]   Else If n == 3
[0152]       x == GetInlOfMinCost(7,3,8)
[0153]   Else If n == 4
[0154]       x == GetInlOfMinCost(5,4,6)
```

[0155] 其中 x 是最终选定为像素阵列的最优预测模式（选定的预测模式）的预测模式的序号。

[0156] 同样，搜索算法确定 n 是否等于 2。如果是，选定的搜索模式就是预测模式 2。如果 n 不等于 2，搜索算法然后确定 n 是否等于 0。如果是，选定的预测模式就是具有预测模式组 (7,0,5) 的最低成本的预测模式。如果 n 不等于 0，搜索算法然后确定 n 是否等于 1。如果是，选定的预测模式就是具有预测模式组 (8,1,6) 的最低成本的预测模式。如果 n 不等于 1，搜索算法然后确定 n 是否等于 3。如果是，选定的预测模式就是具有预测模式组 (7,3,8) 的最低成本的预测模式。如果 n 不等于 3，搜索算法然后确定 n 是否等于 4。如果是，选定的预测模式就是具有预测模式组 (5,4,6) 的最低成本的预测模式。

[0157] 图 5 是 4×4 预测模式搜索方法 500 的流程图，该方法选择宏块的 4×4 阵列的预测模式。在一些实施方式中，方法 500 是由设置用来编码视频帧的软件和 / 或硬件来实现的。在一些实施方式中，4×4 预测模式搜索方法 500 包括图 1 的步骤 115。

[0158] 方法 500 从将宏块分成 16 个 4×4 阵列开始（在 505）。该方法然后将第一个 4×4 阵列设置成用于处理的当前的 4×4 阵列（在 510）。对于当前的 4×4 阵列，方法 500 然后确定每个预测模式 0、1、2、3 和 4 的成本（在 515）。该方法确定具有最低成本的预测模式是 2。如果是，该方法将预测模式 2 设置为当前的 4×4 阵列的预测模式（在 525）。该方法然后确定是否在宏块中存在多个 4×4 阵列需要处理（在 585）。如果是，该方法将下一个 4×4 阵列设置为当前的 4×4 阵列（在 590），并且继续步骤 515。如果不是，该方法结束。

[0159] 如果该方法确定具有最低成本的预测模式不是预测模式 2（在 520- 否），则该方法然后确定具有最低成本的预测模式是否是预测模式 0（在 530）。如果是，该方法随后确定每个预测模式 7、0 和 5 的成本（在 535）。该方法然后将（预测模式 7、0 和 5 中）具有最低成本的预测模式设置为（在 540）当前 4×4 阵列的预测模式。该方法然后继续步骤 585。

[0160] 如果该方法确定具有最低成本的预测模式不是预测模式 0（在 530- 否），则该方法然后确定具有最低成本的预测模式是否是预测模式 1（在 545）。如果是，该方法然后确定每个预测模式 8、1 和 6 的成本（在 550）。该方法然后将（预测模式 8、1 和 6 中）具有最低成本的预测模式设置为当前 4×4 阵列的预测模式（在 555）。该方法然后继续步骤 585。

[0161] 如果该方法确定具有最低成本的预测模式不是预测模式 1（在 545- 否），则该方法然后确定具有最低成本的预测模式是否是预测模式 3（在 560）。如果是，则该方法确定每个预测模式 7、3 和 8 的成本（在 565）。该方法然后将（预测模式 7、3 和 8 中）具有最低成本

的预测模式设置为当前 4×4 阵列的预测模式（在 570）。该方法然后继续步骤 585。

[0162] 如果该方法确定具有最低成本的预测模式不是预测模式 3（在 560- 否），则表示具有最低成本的预测模式是预测模式 4。该方法然后确定每个预测模式 5、4 和 6 的成本（在 575）。该方法然后将（预测模式 5、4 和 6 中）具有最低成本的预测模式设置为当前 4×4 阵列的预测模式（在 580）。该方法然后继续步骤 585。

[0163] 在一些实施方式中，方法 500 选择宏块的 4×4 阵列的预测模式，其中每个 4×4 阵列的预测模式选择取决于宏块中一个或多个其它 4×4 阵列的前一个预测模式的选择。在这些实施方式中，宏块的每个 4×4 阵列的预测模式的选择取决于其上方和左侧相邻 4×4 阵列的编码值（以及从而选定的预测模式）。因此，在这些实施方式中，方法 500 选择从左上部的 4×4 阵列开始的预测模式，接下来是右侧的，然后是向下的。给出以前选定的上方和左侧的相邻的 4×4 阵列的预测模式，方法 500 随后选择当前正在处理的 4×4 阵列的具有最低成本的预测模式（在 9 种预测模式中）。为了这样做，方法 500 例如可以计算最后得到的组合的总体成本（即 16 种确定的成本的总和）。

[0164] A：可选的预测模式搜索算法

[0165] 在可选的实施方式中，可选的预测模式搜索算法执行另外一组操作。最初，可选的搜索算法确定预测模式组（0, 1, 3 和 4）中具有最低的成本的预测模式的序数，其中变量 a 用于表示确定的序数。可选的搜索算法也确定预测模式组（0, 1, 3 和 4）中具有次最低的成本的预测模式的序数，其中 b 用于表示确定的序数。而且，可选的搜索算法确定预测模式组（0, 1, 2, 3 和 4）中具有最低的成本的预测模式的序数，其中变量 n 用于表示确定的序数。

[0166] 这些操作可以由以下的等式来表达。

[0167] $a = \text{GetInlOfMinCost_a}(0, 1, 3, 4)$

[0168] $b = \text{GetInlOfSecondMinCost_b}(0, 1, 3, 4)$

[0169] $n = \text{GetInlOfMinCost}(0, 1, 2, 3, 4)$

[0170] 其中 $\text{GetInlOfSecondMinCost}$ 是函数，其确定预测模式组中（例如，0, 1, 3 和 4）中每个预测模式的成本，并且确定组中的具有次最低成本的预测模式的序数。

[0171] 搜索算法然后执行下列操作：

[0172] If $n == 2$

[0173] $x = 2$

[0174] Else If $(a, b) == (0, 4) \text{ or } (4, 0)$

[0175] $x == \text{GetInlOfMinCost}(0, 5, 4)$

[0176] Else If $(a, b) == (4, 1) \text{ or } (1, 4)$

[0177] $x == \text{GetInlOfMinCost}(4, 6, 1)$

[0178] Else If $(a, b) == (1, 3) \text{ or } (3, 1)$

[0179] $x == \text{GetInlOfMinCost}(1, 8, 3)$

[0180] Else If $(a, b) == (3, 0) \text{ or } (0, 3)$

[0181] $x == \text{GetInlOfMinCost}(3, 7, 0)$

[0182] Else

[0183] $x == n$

[0184] 其中 x 是最终选定为像素阵列的最优预测模式（选定的预测模式）的预测模式的

序号。

[0185] 同样,可选的搜索算法确定 n 是否等于 2。如果是,选定的搜索模式就是预测模式 2。如果 n 不等于 2,可选的搜索算法然后确定是否 a 等于 0 并且 b 等于 4 或者反之亦然 (即是否 a 等于 4 并且 b 等于 0)。如果是,选定的预测模式就是具有预测模式组 (0,5,4) 的最低成本的预测模式。

[0186] 如果 a 等于 0 并且 b 等于 4 或者反之亦然都不是真的,可选的搜索算法然后确定是否 a 等于 4 并且 b 等于 1 或者反之亦然。如果是,选定的预测模式就是具有预测模式组 (4,6,1) 的最低成本的预测模式。

[0187] 如果 a 等于 4 并且 b 等于 1 或者反之亦然都不是真的,可选的搜索算法然后确定是否 a 等于 1 并且 b 等于 3 或者反之亦然。如果是,选定的预测模式就是具有预测模式组 (1,8,3) 的最低成本的预测模式。

[0188] 如果 a 等于 1 并且 b 等于 3 或者反之亦然不是真的,可选的搜索算法然后确定是否 a 等于 3 并且 b 等于 0 或者反之亦然。如果是,选定的预测模式就是具有预测模式组 (3,7,0) 的最低成本的预测模式。

[0189] 图 6 是可选的 4×4 预测模式搜索方法 600 的流程图,其中所述的方法选定宏块的 4×4 阵列的预测模式。在一些实施方式中, 4×4 预测模式搜索方法 600 包括图 1 的步骤 115。

[0190] 方法 600 从将宏块分成 16 个 4×4 阵列开始 (在 605)。该方法然后将第一个 4×4 阵列设置成用于处理的当前的 4×4 阵列 (在 610)。对于当前的 4×4 阵列,方法 600 然后确定每个预测模式 0、1、2、3 和 4 的成本 (在 615)。该方法将变量 a 设置成等于具有预测模式 0、1、3 和 4 中最低成本的预测模式的预测指数 (在 616)。该方法将变量 b 设置成等于具有预测模式 0、1、3 和 4 中次最低成本的预测模式的预测模式指数 (在 617)。该方法将变量 n 设置成等于具有预测模式 0、1、2、3 和 4 中最低成本的预测模式的预测模式指数 (在 618)。

[0191] 该方法确定 n 是否等于 2 (在 620)。如果是,则该方法将预测模式 2 设置成当前 4×4 阵列的预测模式 (在 625)。该方法然后确定是否在宏块中存在多个将被处理的 4×4 阵列 (在 685)。如果是,则该方法将下一个 4×4 阵列设置成当前的 4×4 阵列 (在 690),并继续步骤 615。否则,该方法结束。

[0192] 如果该方法确定 n 不等于 2 (在 620- 否),则该方法然后确定条件是否 (a 等于 0 并且 b 等于 4 或者反之亦然) 为真 (在 630)。如果是,则该方法然后确定预测模式 0、5 和 4 的每个的成本 (在 635)。该方法然后将 (预测模式 0、5 和 4 中) 具有最低成本的预测模式设置成当前 4×4 阵列的预测模式 (在 640)。该方法然后继续到步骤 685。

[0193] 如果该方法确定条件 (a 等于 0 并且 b 等于 4 或者反之亦然) 非真 (在 630- 否),则该方法确定条件 (a 等于 4 并且 b 等于 1 或者反之亦然) 是否为真 (在 645)。如果是,该方法然后确定每个预测模式 4、6 和 1 的成本 (在 650)。该方法然后将 (预测模式 4、6 和 1 中) 具有最低成本的预测模式设置成当前 4×4 阵列的预测模式 (在 655)。该方法然后继续到步骤 685。

[0194] 如果该方法确定条件 (a 等于 4 并且 b 等于 1 或者反之亦然) 非真 (在 645- 否),则该方法确定条件 (a 等于 1 并且 b 等于 3 或者反之亦然) 是否为真 (在 660)。如果是,

该方法然后确定每个预测模式 1、8 和 3 的成本（在 665）。该方法然后将（预测模式 1、8 和 3 中）具有最低成本的预测模式设置成当前 4×4 阵列的预测模式（在 670）。该方法然后继续到步骤 685。

[0195] 如果该方法确定条件（a 等于 1 并且 b 等于 3 或者反之亦然）非真（在 660- 否），则该方法确定条件（a 等于 3 并且 b 等于 0 或者反之亦然）是否为真（在 672）。如果是，则该方法然后确定每个预测模式 3、7 和 0 的成本（在 675）。该方法然后将（预测模式 3、7 和 0 中）具有最低成本的预测模式设置成当前 4×4 阵列的预测模式（在 680）。该方法然后继续到步骤 685。如果该方法确定条件（a 等于 3 并且 b 等于 0 或者反之亦然）非真（在 672- 否），该方法然后将该预测模式设置成等于由变量 n（在步骤 618 确定）所指示的预测模式（在 673），即预测模式 0、1、2、3 和 4 中具有最低成本的预测模式。该方法然后继续到步骤 685。

[0196] 图 7 表示计算机系统 700，利用该计算机系统实施本发明的一些实施方式。计算机系统 700 包括总线 705、处理器 710、系统存储器 715、只读存储器 720、永久存储设备 725、输入设备 730 以及输出设备 735。

[0197] 总线 705 总体来说表示所有的系统、外围设备以及芯片组总线，其可通信地连接计算机系统 700 的多个内部设备。例如，总线 705 可通信地将处理器 710 与只读存储器 720、系统存储器 715 以及永久存储设备 725 相连接。

[0198] 只读存储器 (ROM) 720 存储由处理器 710 和计算机系统的其它模块所需要的静态数据和指令。另一方面，永久存储设备 725 是读写存储设备。该设备是非易失性存储器单元，即使计算机系统 700 关闭时其也存储有指令和数据。一些实施方式利用海量存储设备（诸如磁盘或光盘及其相应的磁盘驱动器）作为永久存储设备 725。其它的实施方式利用可移动的存储设备（诸如软盘或 **zip®** 盘，及其相应的磁盘驱动器）作为永久存储设备。

[0199] 和永久存储设备 725 类似，系统存储器 715 是读写存储设备。然而与存储设备 725 不同的是，系统存储器是易失性读写存储器，诸如随机存取存储器 (RAM)。该系统存储器存储处理器在运行期间所需的一些指令和数据。

[0200] 实现一些实施方式所必需的指令和 / 或数据存储的系统存储器 715、永久存储设备 725、只读存储器 720 或者三者的任意组合中。例如，不同的存储器单元可以包含用于根据一些实施方式编码或解码视频数据流的指令，和 / 或包括视频数据。为了执行一些实施方式的处理，处理器 710 从这些不同的存储器单元检索执行的指令和处理的数据。为了执行一些实施方式的处理，处理器 710 从这些不同的存储器单元检索执行的指令和处理的数据。

[0201] 总线 705 也连接到输入和输出设备 730 和 735。输入设备 730 可以使得用户传递信息并选择给计算机系统 700 的命令。输入设备 730 包括字母数字键盘和光标控制器。输出设备 735 显示由计算机系统 700 生成的图像。输出设备包括打印机和显示设备，诸如阴极射线管 (CRT) 或液晶显示器 (LCD)。

[0202] 最后，如图 7 所示，例如通过网络适配器（未示出），总线 705 也将计算机系统 700 连接到网络 765。以这种方式，计算机系统 700 可以是计算机网络（诸如局域网（“LAN”）、广域网（“WAN”）或者企业内部互联网）或者一种网络（诸如因特网）的一部分。计算机系统 700 的任何或所有部件可以结合一些实施方式一起使用。然而，本领域的普通技术人

员将理解任何其它的系统配置也可以用在其它的实施方式中。

[0203] 根据多个特定的细节对本发明进行了说明,本领域的普通技术人员将认识到在不背离本发明的精神的情况下,可以以其它特定的形式来实现本发明。例如,以上通过参照宏块对本发明的多个实施方式进行了描述。本领域的普通技术人员将认识到这些实施方式可以用在任何其它的像素阵列中。

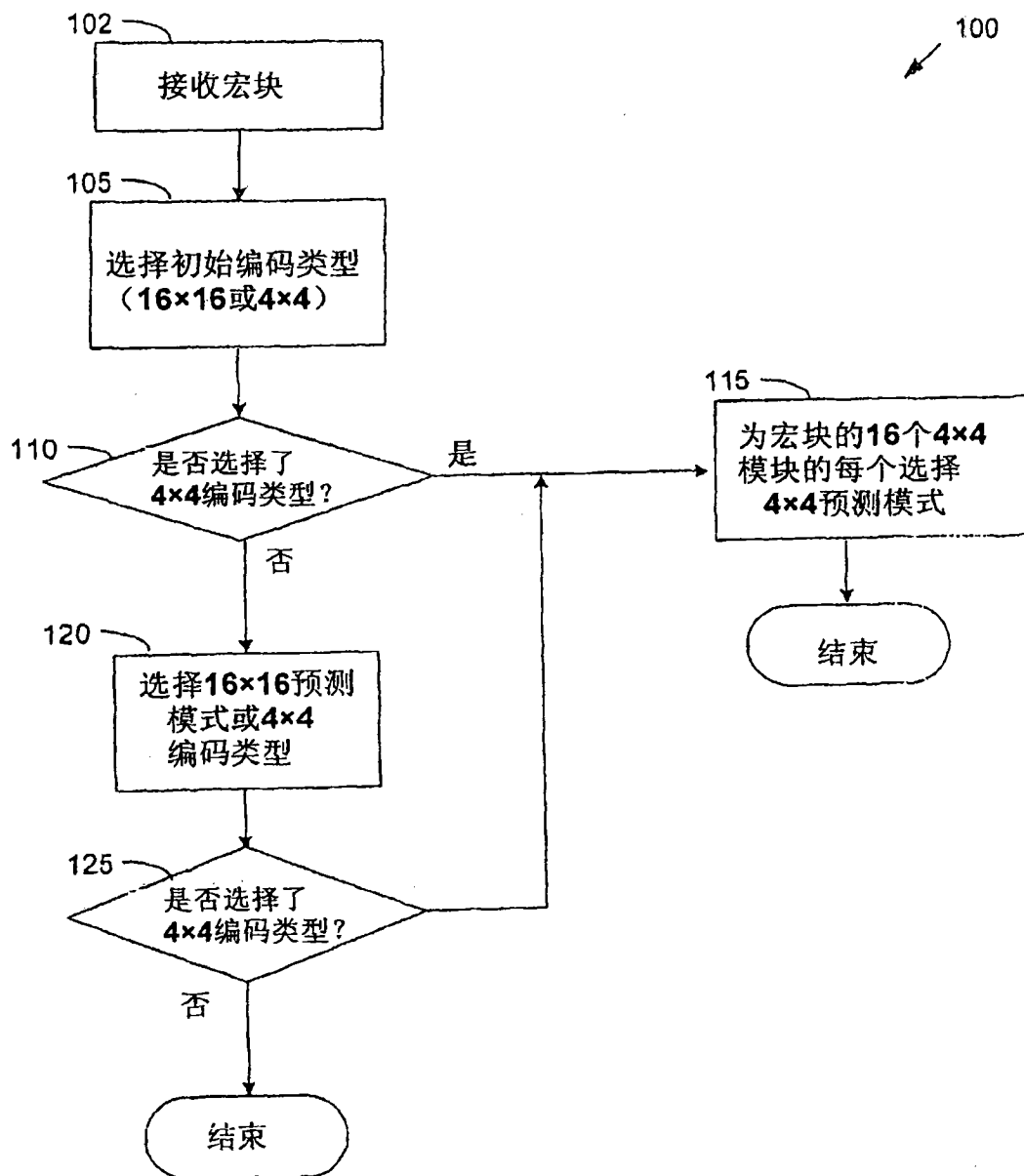


图 1

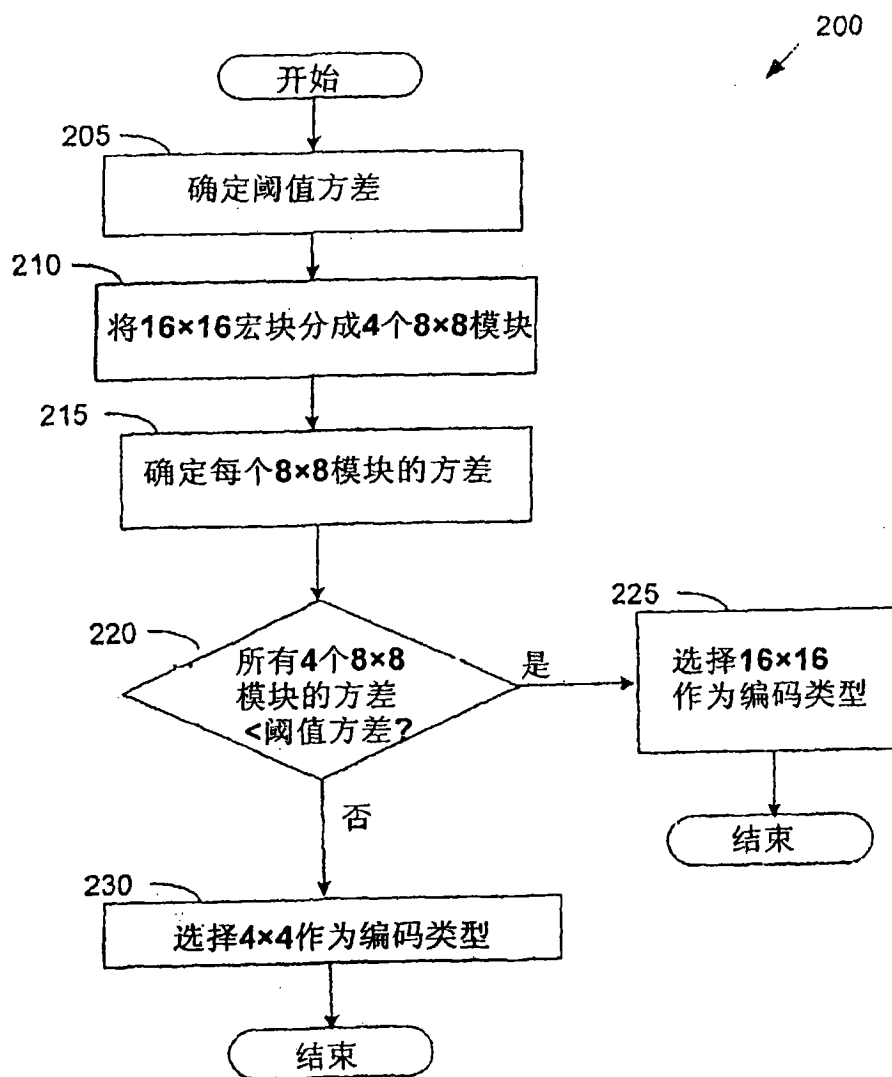


图 2

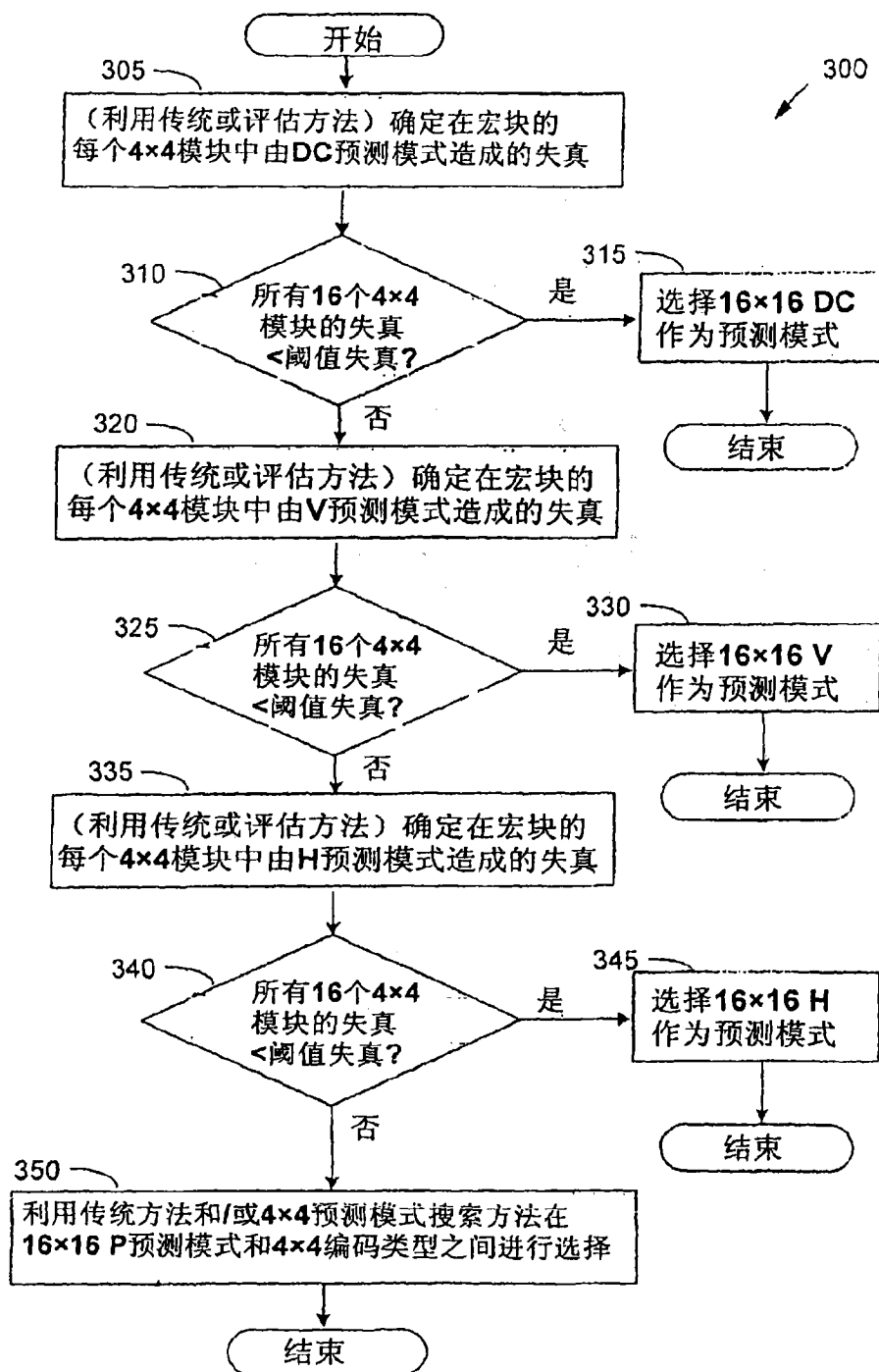


图 3

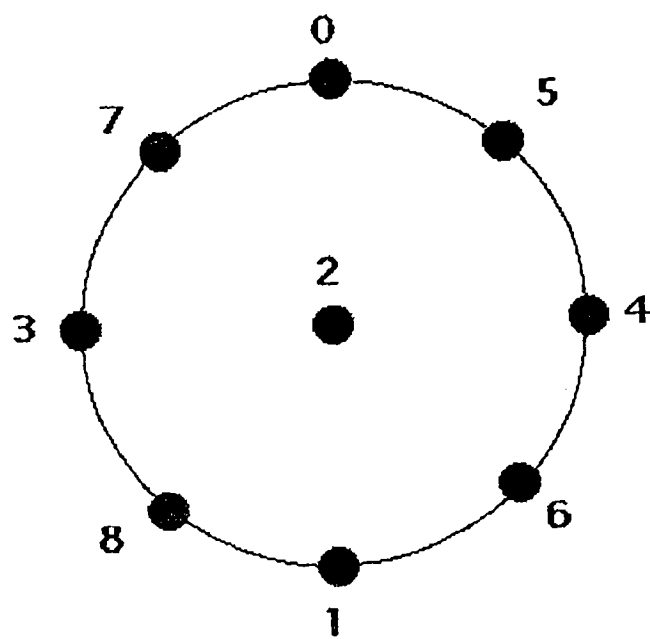


图 4

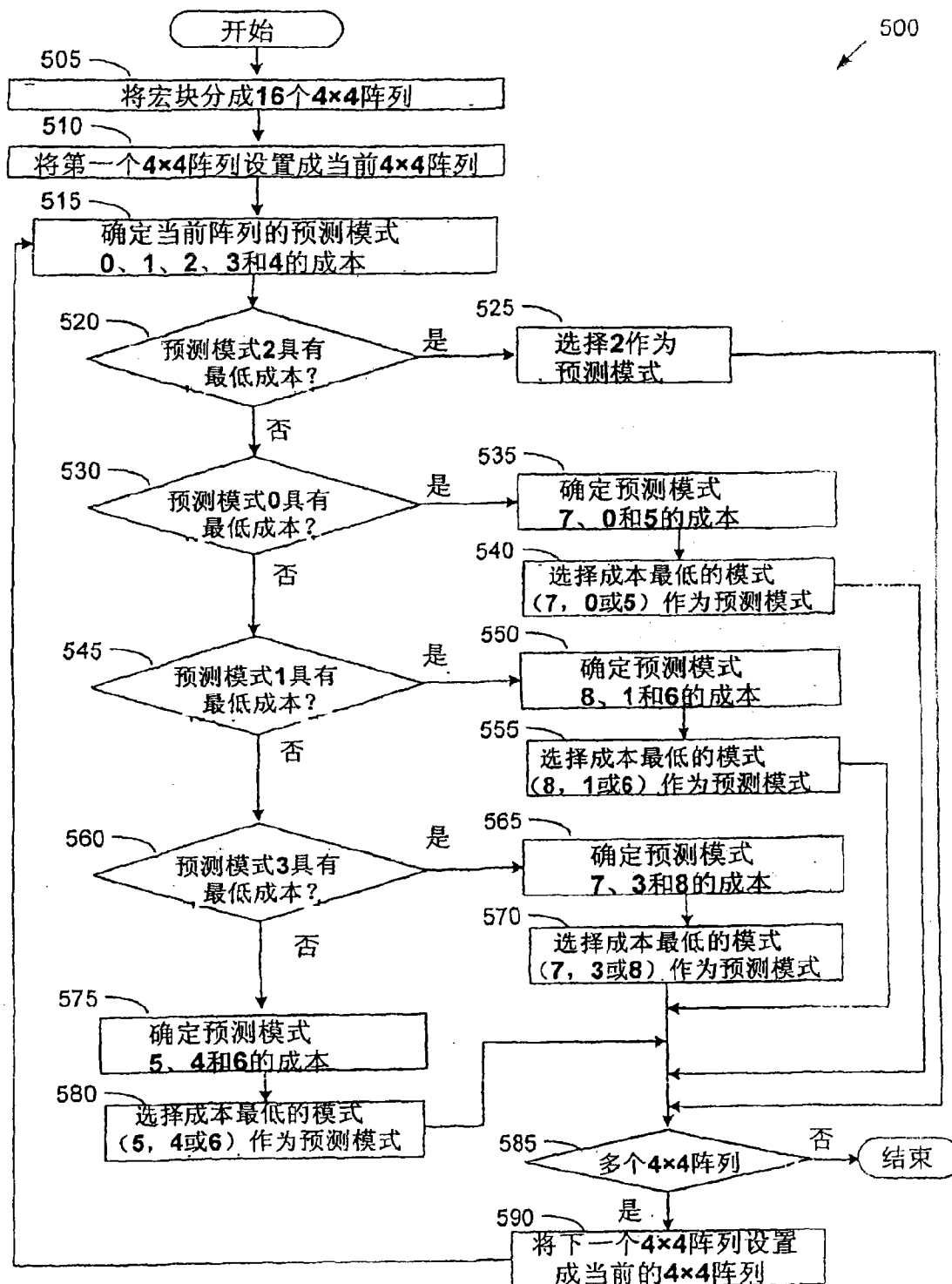


图 5

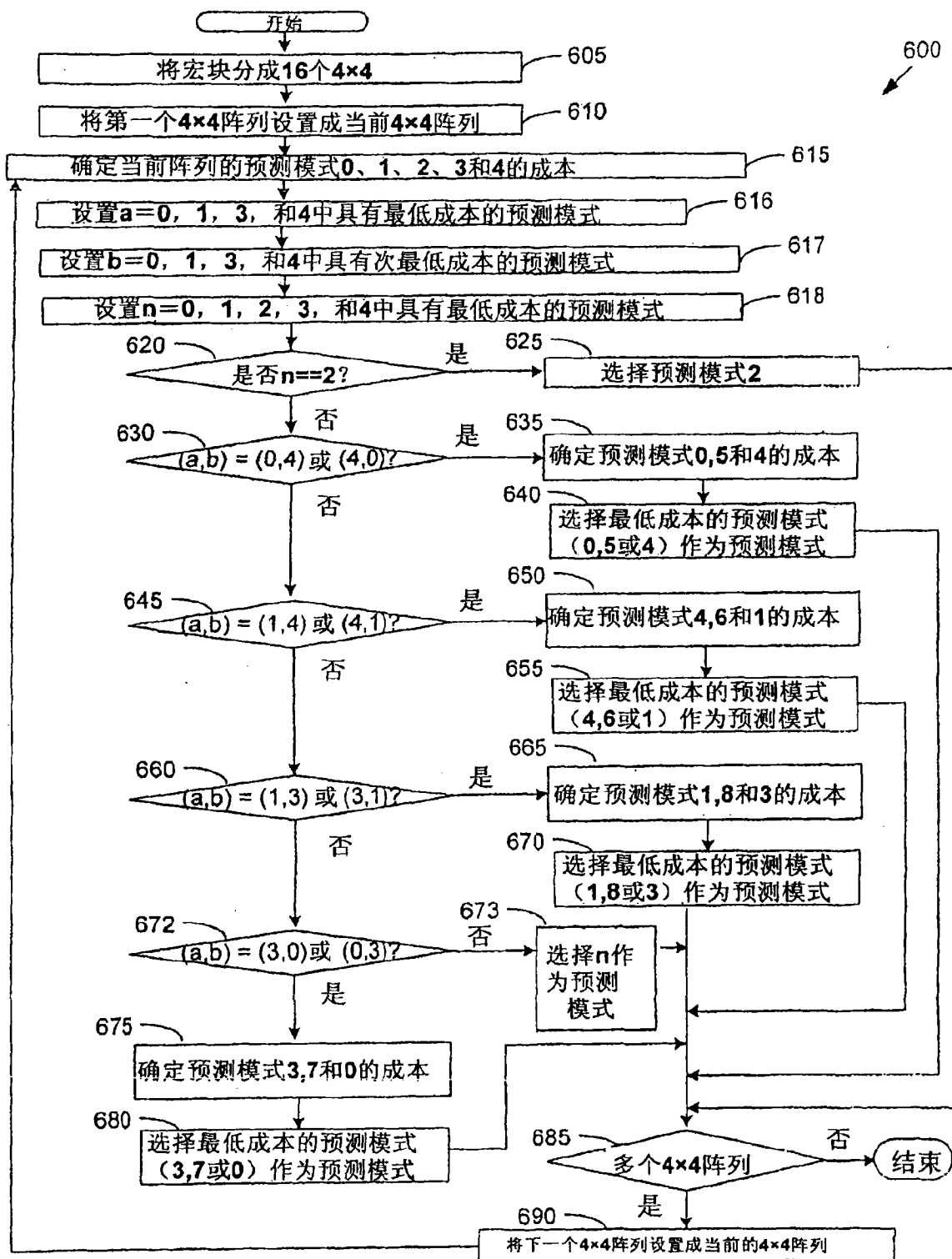


图 6

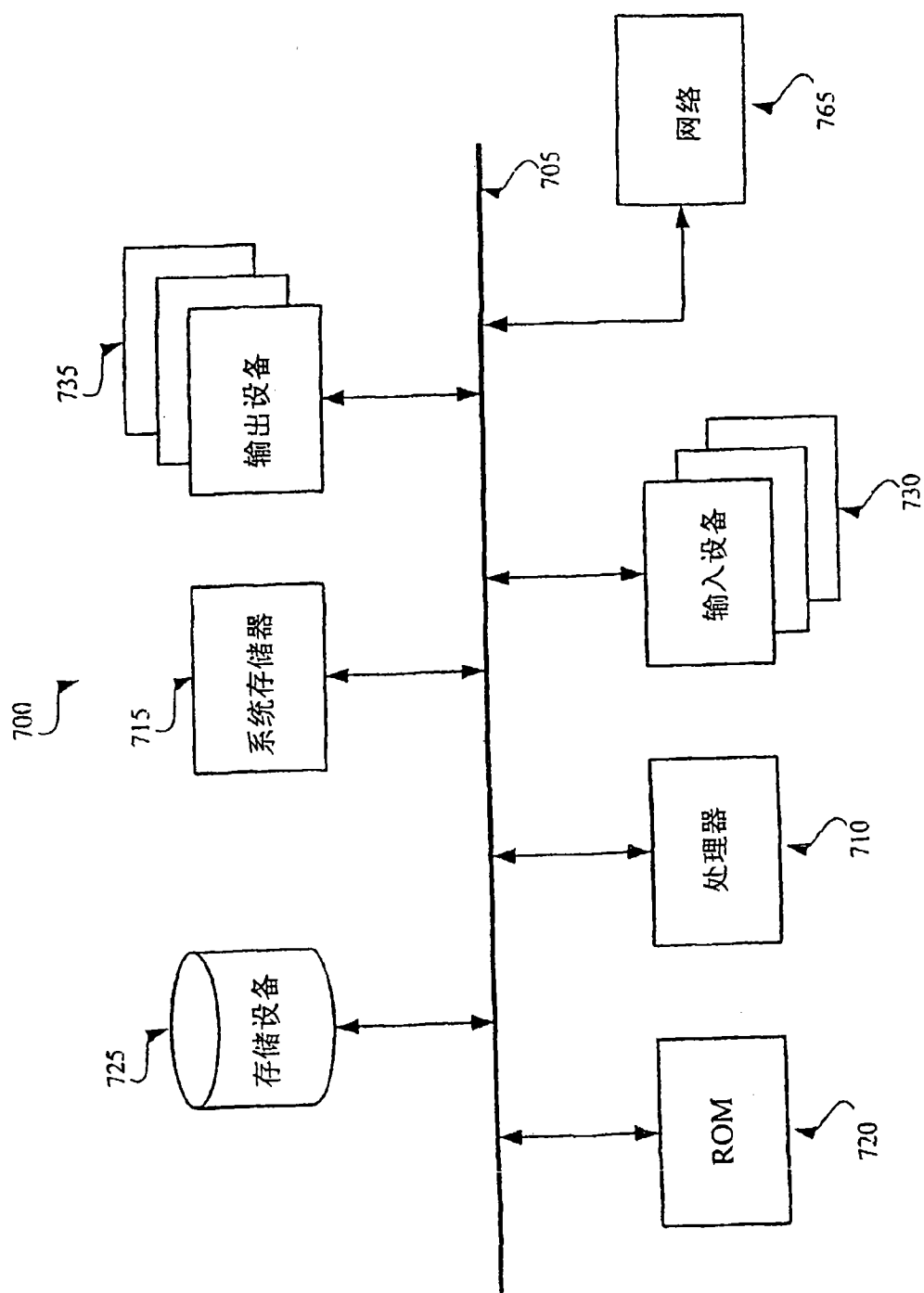


图 7