



(12) 发明专利

(10) 授权公告号 CN 102484709 B

(45) 授权公告日 2015. 06. 10

(21) 申请号 201080038907. 7

(22) 申请日 2010. 06. 29

(30) 优先权数据

61/222, 177 2009. 07. 01 US

(85) PCT国际申请进入国家阶段日

2012. 03. 01

(86) PCT国际申请的申请数据

PCT/US2010/001862 2010. 06. 29

(87) PCT国际申请的公布数据

W02011/002504 EN 2011. 01. 06

(73) 专利权人 汤姆森特许公司

地址 法国伊西莱穆利诺

(72) 发明人 郑云飞 许茜 吕小安 尹鹏

J. 索尔 A. 阿巴斯

(74) 专利代理机构 北京市柳沈律师事务所

11105

代理人 吕晓章

(51) Int. Cl.

H04N 19/176(2014. 01)

H04N 19/70(2014. 01)

H04N 19/147(2014. 01)

H04N 19/196(2014. 01)

H04N 19/96(2014. 01)

H04N 19/11(2014. 01)

H04N 19/463(2014. 01)

H04N 19/136(2014. 01)

(56) 对比文件

KR 20090065401 A, 2009. 06. 22,

CN 101207810 A, 2008. 06. 25,

WO 2009051719 A2, 2009. 04. 23,

Mathias Wien. Variable Block-Size

Transforms for H. 264/AVC. 《IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY》. 2003, 第 13 卷 (第 7 期),

审查员 黄海云

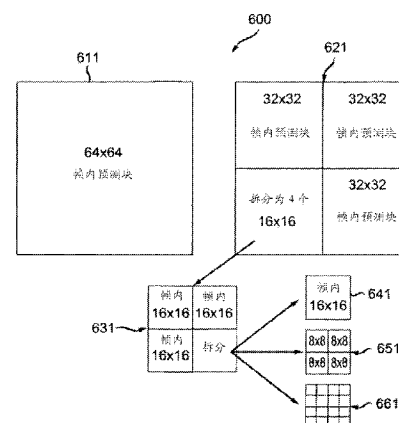
权利要求书3页 说明书16页 附图9页

(54) 发明名称

用于视频编码器和解码器的对大块的帧内预测进行信令的方法和装置

(57) 摘要

提供了用于视频编码器和解码器的对大块的帧内预测进行信令的方法和装置。装置包括视频编码器 (400), 所述视频编码器通过对用于画面中的至少一个大块的帧内预测进行信令来编码所述至少一个大块的画面数据。通过选择基本编码单元尺寸并且分配用于基本编码单元尺寸的单个空间帧内划分类型来对帧内预测进行信令。该单个空间帧内划分类型是从多个空间帧内划分类型中选择的。所述至少一个大块具有比基本编码单元的块尺寸大的大块尺寸。帧内预测是分层级的帧内预测并且通过以下操作中的至少一个而对至少一个大块执行: 将大块尺寸拆分为基本编码单元尺寸以及从基本编码单元尺寸合并到大块尺寸。



1. 一种视频编码装置,包括:

视频编码器(400),通过对用于画面中的至少一个大块的帧内预测进行信令来编码所述至少一个大块的画面数据,其中通过选择基本编码单元尺寸并且分配用于基本编码单元尺寸的单个空间帧内划分类型来对帧内预测进行信令,该单个空间帧内划分类型是从多个空间帧内划分类型中选择的,

其中,所述至少一个大块具有比基本编码单元的块尺寸大的大块尺寸,所述大块尺寸等于或大于 32×32 ,

其中所述帧内预测是分层级的帧内预测并且通过以下操作中的至少一个而对所述至少一个大块执行:将大块尺寸拆分为基本编码单元尺寸的块以及从基本编码单元尺寸的块合并到大块尺寸。

2. 根据权利要求1所述的装置,其中对于多个空间帧内划分类型的每一个,向多个可用的帧内预测模式中最频繁使用的特定的帧内预测模式分配较高的优先级。

3. 根据权利要求1所述的装置,其中自适应地选择所述大块尺寸。

4. 根据权利要求1所述的装置,其中使用一个或多个高级语法元素来执行信令。

5. 根据权利要求1所述的装置,其中空间帧内划分类型表和帧内预测模式表中的至少一个被所述视频编码器(400)预先存储并且使用以便编码所述至少一个大块,并且其中该空间帧内划分类型表和帧内预测模式表中的至少一个被安排为被对应的视频解码器预先存储并且使用以便解码该至少一个大块。

6. 根据权利要求1所述的装置,其中空间帧内划分类型表和帧内预测模式表中的至少一个被所述视频编码器(400)用于编码所述至少一个大块,并且被所述视频编码器使用一个或多个高级语法元素来传送。

7. 一种视频编码器中的方法,包括:

通过对用于画面中的至少一个大块的帧内预测进行信令来编码所述至少一个大块的画面数据(775, 780, 755, 742),其中通过选择基本编码单元尺寸并且分配用于基本编码单元尺寸的单个空间帧内划分类型来对帧内预测进行信令,该单个空间帧内划分类型是从多个空间帧内划分类型中选择的,

其中,所述至少一个大块具有比基本编码单元的块尺寸大的大块尺寸,所述大块尺寸等于或大于 32×32 ,

其中所述帧内预测是分层级的帧内预测并且通过以下操作中的至少一个而对所述至少一个大块执行:将大块尺寸拆分为基本编码单元尺寸的块(720, 725)以及从基本编码单元尺寸的块合并到大块尺寸(742, 750, 755, 770, 775, 780)。

8. 根据权利要求7所述的方法,其中对于多个空间帧内划分类型的每一个,向多个可用的帧内预测模式中最频繁使用的特定的帧内预测模式分配较高的优先级(785, 790, 730)。

9. 根据权利要求7所述的方法,其中自适应地选择所述大块尺寸(750, 770)。

10. 根据权利要求7所述的方法,其中使用一个或多个高级语法元素来执行信令(742, 755, 775, 780, 785, 790, 797)。

11. 根据权利要求7所述的方法,其中空间帧内划分类型表和帧内预测模式表中的至少一个被视频编码器预先存储并且使用以便编码所述至少一个大块,并且其中该空间帧内

划分类型表和帧内预测模式表中的至少一个被安排为被对应的视频解码器预先存储并且使用以便解码该至少一个大块 (710)。

12. 根据权利要求 7 所述的方法, 其中空间帧内划分类型表和帧内预测模式表中的至少一个被视频编码器用于编码所述至少一个大块, 并且被视频编码器使用一个或多个高级语法元素来传送 (797)。

13. 一种视频解码装置, 包括:

视频解码器 (500), 通过确定要为画面中的至少一个大块执行的帧内预测来解码所述至少一个大块的画面数据, 其中通过确定基本编码单元尺寸并且确定用于基本编码单元尺寸的单个空间帧内划分类型来确定帧内预测, 该单个空间帧内划分类型是可从多个空间帧内划分类型中确定的,

其中, 至少一个块具有比基本编码单元的块尺寸大的大块尺寸, 所述大块尺寸等于或大于 32×32 ,

其中所述帧内预测是分层级的帧内预测并且通过以下操作中的至少一个而对所述至少一个大块执行: 将大块尺寸拆分为基本编码单元尺寸的块以及从基本编码单元尺寸的块合并到大块尺寸。

14. 根据权利要求 13 所述的装置, 其中对于多个空间帧内划分类型的每一个, 向多个可用的帧内预测模式中最频繁使用的特定的帧内预测模式分配较高的优先级。

15. 根据权利要求 13 所述的装置, 其中自适应地选择所述大块尺寸。

16. 根据权利要求 13 所述的装置, 其中使用一个或多个高级语法元素来确定所述帧内预测。

17. 根据权利要求 13 所述的装置, 其中空间帧内划分类型表和帧内预测模式表中的至少一个被所述视频解码器 (500) 预先存储并且使用以便解码所述至少一个大块, 并且其中所述至少一个大块之前被对应的编码器使用在其中预先存储的空间帧内划分类型表和帧内预测模式表中的至少一个进行编码。

18. 根据权利要求 13 所述的装置, 其中空间帧内划分类型表和帧内预测模式表中的至少一个通过所述视频解码器 (500) 使用一个或多个高级语法元素进行接收并且被所述视频解码器 (500) 用于解码所述至少一个大块。

19. 一种视频解码器中的方法, 包括:

通过确定要为画面中的至少一个块执行的帧内预测来解码至少一个大块的画面数据 (820, 830), 其中通过确定基本编码单元尺寸并且确定用于基本编码单元尺寸的单个空间帧内划分类型来确定所述帧内预测, 单个空间帧内划分类型是可从多个空间帧内划分类型中确定的,

其中, 所述至少一个大块具有比基本编码单元的块尺寸大的大块尺寸, 所述大块尺寸等于或大于 32×32 ,

其中所述帧内预测是分层级的帧内预测并且通过以下操作中的至少一个而对所述至少一个大块执行: 将大块尺寸拆分为基本编码单元尺寸的块以及从基本编码单元尺寸的块合并到大块尺寸 (820, 830)。

20. 根据权利要求 19 所述的方法, 其中对于多个空间帧内划分类型的每一个, 向多个可用的帧内预测模式中最频繁使用的特定的帧内预测模式分配较高的优先级

(855, 835, 850)。

21. 根据权利要求 19 所述的方法, 其中自适应地确定所述大块尺寸 (820, 830)。

22. 根据权利要求 19 所述的方法, 其中使用一个或多个高级语法元素来确定所述帧内预测 (820, 830, 835, 885, 850)。

23. 根据权利要求 19 所述的方法, 其中空间帧内划分类型表和帧内预测模式表中的至少一个被所述视频解码器预先存储并且使用以便解码所述至少一个大块, 并且其中所述至少一个大块之前被对应的编码器使用在其中预先存储的空间帧内划分类型表和帧内预测模式表中的至少一个进行编码 (808)。

24. 根据权利要求 19 所述的方法, 其中空间帧内划分类型表和帧内预测模式表中的至少一个通过所述视频解码器使用一个或多个高级语法元素进行接收并且被所述视频解码器用于解码所述至少一个大块 (810)。

用于视频编码器和解码器的对大块的帧内预测进行信令的方法和装置

[0001] 相关申请的交叉引用

[0002] 本申请要求 2009 年 7 月 1 日提交的美国临时申请序列号 No. 61/222, 177 (代理案号 No. PU090082) 的权益, 通过引用将其内容全部并入于此。

技术领域

[0003] 本原理一般地涉及视频编码和解码, 并且更具体地涉及用于视频编码器和解码器的对大块的帧内预测进行信令 (signal) 的方法和装置。

背景技术

[0004] 多数现代视频编码标准采用各种编码模式来有效地减少空间域和时间域中的相关度。例如, 在国际标准化组织 / 国际电工委员会 (ISO/IEC) 运动画面专家组 -4 (MPEG-4) 第 10 部分高级视频编码 (AVC) 标准 / 国际电信联盟电信分部 (ITU-T) H. 264 推荐 (下文的“MPEG-4AVC 标准”) 中, 可以帧内编码或者帧间编码画面。在帧内画面中, 以帧内模式编码所有宏块, 由此利用画面内的空间相关度。帧内模式可以被归类为以下三种类型: INTRA4×4; INTRA8×8; INTRA16×16。INTRA4×4 和 INTRA8×8 支持 9 种帧内预测模式, INTRA16×16 支持 4 种帧内预测模式。

[0005] INTRA 4×4 和 INTRA8×8 支持以下 9 种帧内预测模式: 垂直预测、水平预测、DC 预测、对角下 / 左预测、对角下 / 右预测、垂直 - 左预测、水平 - 下预测、垂直 - 右预测, 以及水平 - 上预测。INTRA16×16 支持以下 4 种帧内预测模式: 垂直预测、水平预测、DC 预测, 以及平面预测。转到图 1, 由参考标号 100 总地指示 INTRA4×4 和 INTRA8×8 预测模式。在图 1 中, 参考标号 0 指示垂直预测模式、参考标号 1 指示水平预测模式、参考标号 3 指示对角下 / 左预测模式、参考标号 4 指示对角下 / 右预测模式、参考标号 5 指示垂直 - 右预测模式、参考标号 6 指示水平 - 下预测模式、参考标号 7 指示垂直 - 左预测模式、参考标号 8 指示垂直 - 上预测模式。未示出作为 INTRA4×4 和 INTRA8×8 预测模式一部分的 DC 模式。转到图 2, 由参考标号 200 总地指示 INTRA16×16 预测模式。在图 2 中, 参考标号 0 指示垂直预测模式、参考标号 1 指示水平预测模式、参考标号 3 指示平面预测模式。未示出作为 INTRA16×16 预测模式一部分的 DC 模式。

[0006] INTRA4×4 使用 4×4 离散余弦变换 (DCT)。INTRA8×8 使用 8×8 变换。INTRA16×16 使用级联的 4×4 变换。为了进行信令, INTRA4×4 和 INTRA8×8 共享相同的宏块类型 (mb_type) 0 并且通过变换尺寸标志 (transform_8×8_size_flag) 来区分。然后, 通过最可能的模式 (如果必要, 可能利用其余模式) 来对在 INTRA4×4 或 INTRA8×8 中帧内预测模式的选取进行信令。对于 INTRA16×16, 在 mb_type 中对所有帧内预测模式连同编码块图案 (cbp) 类型进行信令, 其使用 1 到 24 的 mb_type 值。表 1 示出了用于帧内编码码片 (I 码片) 的宏块类型的详细的信令。如果尺寸大于 16×16 的更大的块用于帧内预测, 则面对如下的若干可能的问题。

[0007] (1) 如果通过在 MPEG-4AVC 标准中简单地扩展 mb_type 来增加 INTRA 32×32 或者 INTRA 64×64 预测, 则其将对这两种新模式造成太多的开销, 并且另外, 将不允许帧内预测的分级类型。如下解释帧内预测的分级类型的示例。如果 32×32 块用作大块并且允许子划分为 16×16 , 则对于每个 16×16 子划分, 应允许 INTRA 4×4 、INTRA 8×8 、INTRA 16×16 。

[0008] (2) 如果更大的变换 (诸如 16×16 变换) 而不是级联的变换用于 INTRA 16×16 , 则不能应用当前的信令。

[0009] (3) 应对一个帧内划分类型内部的帧内预测模式给出不同的优先级。

[0010] 表 1

[0011]

mb_type	mb_type 的名称	transform_size _8x8_flag	Mb 划分预测模式 (mb_type, 0)	Intra16x16- 预测模式	编码块 图案色度	编码块 图案色度
0	I_NxN	0	Intra_4x4	na	方程式 7-33	方程式 7-33
0	I_NxN	1	Intra_8x8	na	方程式 7-33	方程式 7-33
1	I_16x16_0_0_0	na	Intra_16x16	0	0	0
2	I_16x16_1_0_0	na	Intra_16x16	1	0	0
3	I_16x16_2_0_0	na	Intra_16x16	2	0	0
4	I_16x16_3_0_0	na	Intra_16x16	3	0	0
5	I_16x16_0_1_0	na	Intra_16x16	0	1	0
6	I_16x16_1_1_0	na	Intra_16x16	1	1	0
7	I_16x16_2_1_0	na	Intra_16x16	2	1	0
8	I_16x16_3_1_0	na	Intra_16x16	3	1	0
9	I_16x16_0_2_0	na	Intra_16x16	0	2	0
10	I_16x16_1_2_0	na	Intra_16x16	1	2	0
11	I_16x16_2_2_0	na	Intra_16x16	2	2	0
12	I_16x16_3_2_0	na	Intra_16x16	3	2	0
13	I_16x16_0_0_1	na	Intra_16x16	0	0	15
14	I_16x16_1_0_1	na	Intra_16x16	1	0	15
15	I_16x16_2_0_1	na	Intra_16x16	2	0	15
16	I_16x16_3_0_1	na	Intra_16x16	3	0	15
17	I_16x16_0_1_1	na	Intra_16x16	0	1	15
18	I_16x16_1_1_1	na	Intra_16x16	1	1	15
19	I_16x16_2_1_1	na	Intra_16x16	2	1	15
20	I_16x16_3_1_1	na	Intra_16x16	3	1	15
21	I_16x16_0_2_1	na	Intra_16x16	0	2	15
22	I_16x16_1_2_1	na	Intra_16x16	1	2	15
23	I_16x16_2_2_1	na	Intra_16x16	2	2	15
24	I_16x16_3_2_1	na	Intra_16x16	3	2	15
25	I_PCM	na	na	na	na	na

[0012] MPEG-4AVC 标准的扩展中存在与对大的运动（帧间）划分进行信令有关的一些现有技术方法。关于第一种现有技术方法来描述在 MPEG-4AVC 标准的扩展中怎样对大的运动（帧间）划分进行信令的一个示例。第一种现有技术方法描述了怎样为使用分级编码结构的 32×32 块或者 64×64 块进行信令。

[0013] 此外，除了 MPEG-4AVC 标准中现有的运动划分尺寸（ 16×16 、 16×8 、 8×16 、 8×8 、 8×4 、 4×8 和 4×4 ）之外，也已经提出使用 32×32 、 32×16 和 16×32 划分的用于 MPEG-4AVC

标准的扩展的帧间编码。转到图 3,通过参考标号 300 总地指示用于 32×32 块中的运动划分。划分包括 32×32 、 32×16 、 16×32 和 16×16 。 16×16 划分可以进一步被划分为尺寸 16×16 、 16×8 、 8×16 和 8×8 的划分。此外, 8×8 划分可以进一步被划分为尺寸 8×8 、 8×4 、 4×8 和 4×4 的划分。

[0014] 对于每个 32×32 的块,以对 MPEG-4AVC 标准的其它模式执行的方式类似的方式使用 mb32_skip_flag 来对 SKIP 模式或者 DIRECT 模式进行信令。另外,MPEG-4AVC 标准中的 $M \times N$ ($M = 8$ 或 16 而 $N = 8$ 或 16) 划分的原始 mb_type 还用于对 32×32 块中的 $2M \times 2N$ 划分进行信令。如果 32×32 的 mb32_type 指示使用 16×16 划分,则通过使用与 MPEG-4 AVC 标准中的 macroblock_layer() 相同的语法元素,以光栅扫描顺序来对四个 16×16 块进行信令。可以进一步以二叉树方式将每个 16×16 块从尺寸 16×16 向下划分为尺寸 4×4 。

[0015] 对于宏块尺寸 64×64 ,在 32×32 块中使用的划分之上添加了以下划分: 64×64 、 64×32 和 32×64 。由此,在块尺寸 32×32 之上的宏块划分中添加了超过一个分级的层。MPEG-4AVC 标准中的 $M \times N$ ($M = 8$ 或 16 而 $N = 8$ 或 16) 宏块划分的原始的 mb_ttype 用于对 64×64 宏块中的 $4M \times 4N$ 宏块划分进行信令。如果 32×32 宏块划分用于 64×64 块,则每个 32×32 块将以与上述相同的方式被处理。

[0016] 然而,现有文献没有解决怎样对大的帧内模式进行信令,其中大的帧内模式被定义为意指涉及具有等于或者大于 32×32 的尺寸的划分块的帧内预测。

发明内容

[0017] 通过本原理解决现有技术的这些和其它缺陷和缺点,本原理针对用于视频编码器和解码器的对大块的帧内预测进行信令的方法和装置。

[0018] 根据本原理的一个方面,提供了一种装置。该装置包括视频编码器,所述视频编码器通过对用于画面中的至少一个大块的帧内预测进行信令来编码所述至少一个大块的画面数据。通过选择基本编码单元尺寸并且分配用于基本编码单元尺寸的单个空间帧内划分类型来对帧内预测进行信令。该单个空间帧内划分类型是可从多个空间帧内划分类型中选择的。所述至少一个大块具有比基本编码单元的块尺寸大的大块尺寸。帧内预测是分层级的帧内预测并且通过以下操作中的至少一个而对至少一个大块执行:将大块尺寸拆分为基本编码单元尺寸以及从基本编码单元尺寸合并到大块尺寸。

[0019] 根据本原理的另一方面,提供了一种视频编码器中的方法。该方法包括通过对用于画面中的至少一个大块的帧内预测进行信令来编码所述至少一个大块的画面数据。通过选择基本编码单元尺寸并且分配用于基本编码单元尺寸的单个空间帧内划分类型来对帧内预测进行信令。该单个空间帧内划分类型是可从多个空间帧内划分类型中选择的。所述至少一个大块具有比基本编码单元的块尺寸大的大块尺寸。帧内预测是分层级的帧内预测并且通过以下操作中的至少一个而对至少一个大块执行:将大块尺寸拆分为基本编码单元尺寸以及从基本编码单元尺寸合并到大块尺寸。

[0020] 根据本原理的又一方面,提供了一种装置。该装置包括视频解码器,所述视频解码器通过确定要为画面中的至少一个大块执行的帧内预测来解码所述至少一个大块的画面数据。通过确定基本编码单元尺寸并且确定用于基本编码单元尺寸的单个空间帧内划分类型来确定帧内预测。该单个空间帧内划分类型是可从多个空间帧内划分类型中确定的。所

述至少一个大块具有比基本编码单元的块尺寸大的大块尺寸。帧内预测是分层级的帧内预测并且通过以下操作中的至少一个而对至少一个大块执行：将大块尺寸拆分为基本编码单元尺寸以及从基本编码单元尺寸合并到大块尺寸。

[0021] 根据本原理的再一方面，提供了一种视频解码器中的方法。该方法包括通过确定要为画面中的至少一个大块执行的帧内预测来解码所述至少一个大块的画面数据。通过确定基本编码单元尺寸并且确定用于基本编码单元尺寸的单个空间帧内划分类型来确定帧内预测。该单个空间帧内划分类型是可从多个空间帧内划分类型中确定的。所述至少一个大块具有比基本编码单元的块尺寸大的大块尺寸。帧内预测是分层级的帧内预测并且通过以下操作中的至少一个而对至少一个大块执行：将大块尺寸拆分为基本编码单元尺寸以及从基本编码单元尺寸合并到大块尺寸。

[0022] 本原理的这些和其它方面、特征和优点将从示例实施例的以下具体描述中变得明显，将结合附图阅读以下具体描述。

附图说明

[0023] 依据以下示例性图将更好地理解本原理，其中：

[0024] 图 1 是示出可以应用本原理的 INTRA4×4 和 INTRA8×8 预测模式的图；

[0025] 图 2 是示出可以应用本原理的 INTRA16×16 预测模式的图；

[0026] 图 3 是示出可以应用本原理的用于 32×32 块的运动划分的图；

[0027] 图 4 是依据本原理实施例的可以应用本原理的示例性视频编码器的框图；

[0028] 图 5 是依据本原理实施例的可以应用本原理的示例性视频解码器的框图；

[0029] 图 6 是依据本原理实施例的可以应用本原理的示例性分级划分的框图；

[0030] 图 7A 和 7B 表示依据本原理实施例的通过对用于大块的帧内预测进行信令来编码所述大块的画面数据的示例性方法的流程图；以及

[0031] 图 8A 和 8B 表示依据本原理实施例的通过确定要被应用到大块的帧内预测来解码所述大块的画面数据的示例性方法的流程图。

具体实施方式

[0032] 本原理针对用于视频编码器和解码器的对大块的帧内预测进行信令的方法和装置。

[0033] 本描述例示了本原理。因此，将理解，本领域技术人员将能够开发未在这里明确描述或示出但是体现本原理并且被包括在本原理的精神和范围之内的各种布置。

[0034] 在此叙述的所有示例和条件性语言意欲用于教导的目的以便帮助读者理解本原理以及由（多个）发明人贡献以推动本领域发展的构思，并且应该被解释为不局限于这样具体叙述的示例和条件。

[0035] 另外，在这里叙述本原理的原理、方面和实施例的所有陈述，及其具体示例意欲包括其结构和功能上的等同物。另外，意图是：这样的等同物包括当前已知的等同物以及将来开发的等同物二者，即所开发的执行相同功能的任何元件，而不论其结构如何。

[0036] 因此，例如，本领域技术人员将认识到：在此呈现的框图表示体现本原理的例示性电路的概念性视图。类似地，将认识到：任何流程图示 (flow chart)、流程图 (flow

diagram)、状态转换图、伪代码等表示实质上可以表示在计算机可读介质中并因此由计算机或处理器执行的各种处理,而不管是否明确地示出这样的计算机或处理器。

[0037] 可以通过使用专用硬件、以及与适当的软件相关联的能够执行软件的硬件来提供图中示出的各种元件的功能。当利用处理器来提供所述功能时,可以利用单个专用处理器、利用单个共享处理器、或者利用其中一些可被共享的多个独立处理器来提供所述功能。另外,术语“处理器”或“控制器”的明确使用不应该被解释为排他性地指代能够执行软件的硬件,而是可以隐含地不受限制地包括数字信号处理器(“DSP”)硬件、用于存储软件的只读存储器(“ROM”)、随机存取存储器(“RAM”)、和非易失性存储装置。

[0038] 还可以包括其它传统的和/或定制的硬件。类似地,图中示出的任何开关只是概念性的。它们的功能可以通过程序逻辑的运行、通过专用逻辑、通过程序控制和专用逻辑的交互、或者甚至手动地来执行,如从上下文更具体地理解的,实施者可选择具体技术。

[0039] 在其权利要求中,被表示为用于执行指定功能的部件的任何元件意欲包含执行那个功能的任何方式,例如包括:a) 执行那个功能的电路元件的组合或者 b) 与适当电路相组合的任何形式的软件,所述软件因此包括固件或微代码等,所述适当电路用于执行该软件以执行所述功能。由这种权利要求限定的本发明在于如下事实,即,以权利要求所要求的方式将由各种所叙述的部件提供的功能组合和集合到一起。因此认为可以提供那些功能的任何部件与在此示出的那些部件等同。

[0040] 在本说明书中提到的本原理的“一个实施例”或“实施例”及其其它变型意味着:结合所述实施例描述的具体特征、结构、特性等被包括在本原理的至少一个实施例中。因此,在说明书各处出现的短语“在一个实施例中”和“在实施例中”、以及任何其它变型不一定都指代相同的实施例。

[0041] 应当认识到,“/”、“和/或”以及“至少一个”任何一个的使用,例如在“A/B”、“A和/或B”和“A和B中的至少一个”的情况中,意欲包括仅仅对第一个列出的选项(A)的选择、或仅仅对第二个列出的选项(B)的选择、或者对于两个选项(A和B)的选择。作为另一示例,在“A、B和/或C”以及“A、B和C中的至少一个”的情况中,这种措辞意欲包括仅仅对第一个列出的选项(A)的选择、或仅仅对第二个列出的选项(B)的选择、或仅仅对第三个列出的选项(C)的选择、或仅仅对第一个和第二个列出的选项(A和B)的选择、或仅仅对第一个和第三个列出的选项(A和C)的选择、或仅仅对第二个和第三个列出的选项(B和C)的选择、或者对于全部三个选项(A和B和C)的选择。如本领域和相关领域普通技术人员容易认识到的,这可以被扩展用于很多列出的条目。

[0042] 此外,应理解,尽管这里关于 MPEG-4AVC 标准的扩展来描述本原理的一个或多个实施例,但本原理不仅仅限于该扩展或者该标准,并且因此可以关于其它视频编码标准、推荐、及其扩展而被利用,同时保持本原理的精神。

[0043] 如这里所使用的,“高级语法”指代在分级上高于宏块层驻留的比特流中存在的语法。例如,如这里所使用的,高级语法可以指代但不限于:码片首标级、补充增强信息(SEI)级、画面参数集(PPS)级、序列参数集(SPS)级、和网络抽象层(NAL)单元首标级处的语法。

[0044] 而且,如这里所使用的,词语“画面”和“图像”被可互换地使用,并且指代静止图像或者来自视频序列的画面。如已知的,画面可以是帧或场。

[0045] 此外,如这里所使用的,词语“信令”指代向对应解码器指示某些内容

(something)。例如,编码器可以对被指定用于特定的大块(如在此定义的)的帧内预测进行信令以便使得解码器得知在编码器侧使用了特定的预测类型(例如,帧内或者帧间)。以此方式,可以在编码器侧和解码器侧使用相同的预测类型。由此,例如,编码器可以对特定的大块传送关于在该大块上要执行帧内预测的指示(例如,信号),以便简单地使得解码器知道并且选择用于该大块的相同的预测类型。应理解,可以以多种方式来实现该信令。例如,可以使用一个或多个语法元素、标志等来向对应解码器对信息进行信令。

[0046] 转到图 4,由参考标号 400 总地指示依据本原理实施例的可以应用本原理的示例性视频编码器。

[0047] 视频编码器 400 包括帧排序缓冲器 410,该帧排序缓冲器 410 具有与组合器 485 的非反相输入端进行信号通信的输出端。组合器 485 的输出端以信号通信方式与变换器和量化器 425 的第一输入端连接。变换器和量化器 425 的输出端以信号通信方式与熵编码器 445 的第一输入端以及逆变换器和逆量化器 450 的第一输入端连接。熵编码器 445 的输出端以信号通信方式与组合器 490 的第一非反相输入端连接。组合器 490 的输出端以信号通信方式与输出缓冲器 435 的第一输入端连接。

[0048] 编码器控制器 405 的第一输出端以信号通信方式与帧排序缓冲器 410 的第二输入端、逆变换器和逆量化器 450 的第二输入端、画面类型判定模块 415 的输入端、宏块类型(MB 类型)判定模块 420 的输入端、超级帧内预测模块 460 的第二输入端、去块滤波器 465 的第二输入端、运动补偿器 470 的第一输入端、运动估计器 475 的第一输入端、和参考画面缓冲器 480 的第二输入端连接。

[0049] 编码器控制器 405 的第二输出端以信号通信方式与补充增强信息(SEI)插入器 430 的第一输入端、变换器和量化器 425 的第二输入端、熵编码器 445 的第二输入端、输出缓冲器 435 的第二输入端、以及序列参数集(SPS)和画面参数集(PPS)插入器 440 的输入端连接。

[0050] 画面类型判定模块 415 的第一输出端以信号通信方式与帧排序缓冲器 410 的第三输入端连接。画面类型判定模块 115 的第二输出端以信号通信方式与宏块类型判定模块 420 的第二输入端连接。

[0051] 序列参数集(SPS)和画面参数集(PPS)插入器 440 的输出端以信号通信方式与组合器 490 的第三非反相输入端连接。

[0052] 逆量化器和逆变换器 450 的输出端以信号通信方式与组合器 419 的第一非反相输入端连接。组合器 419 的输出端以信号通信方式与超级帧内预测模块 460 的第一输入端和去块滤波器 465 的第一输入端连接。去块滤波器 465 的输出端以信号通信方式与参考画面缓冲器 480 的第一输入端连接。参考画面缓冲器 480 的输出端以信号通信方式与运动估计器 475 的第二输入端连接。运动估计器 475 的第一输出端以信号通信方式与运动补偿器 470 的第二输入端连接。运动估计器 475 的第二输出端以信号通信方式与熵编码器 445 的第三输入端连接。

[0053] 运动补偿器 470 的输出端以信号通信方式与开关 497 的第一输入端连接。超级帧内预测模块 460 的输出端以信号通信方式与开关 497 的第二输入端连接。宏块类型判定模块 420 的输出端以信号通信方式与开关 497 的第三输入端连接。开关 497 的第三输入端确定开关的“数据”输入(与控制输入(即,第三输入端)相对)是由运动补偿器 470 提供还

是由超级帧内预测模块 460 提供。开关 497 的输出端以信号通信方式与组合器 419 的第二非反相输入端和组合器 485 的反向输入端连接。

[0054] 帧排序缓冲器 410 和编码器控制器 405 的输入端可用作编码器 400 的用于接收输入画面 401 的输入端。此外,补充增强信息 (SEI) 插入器 430 的输入端可用作编码器 400 的用于接收元数据的输入端。输出缓冲器 435 的输出端可用作编码器 400 的用于输出比特流的输出端。

[0055] 转到图 5,通过参考标号 500 总地指示依据本原理实施例的可以应用本原理的示例性视频解码器。

[0056] 视频解码器 500 包括输入缓冲器 510,该输入缓冲器 510 具有以信号通信方式与熵解码器 545 的第一输入端连接的输出端。熵解码器 545 的第一输出端以信号通信方式与逆变换器和逆量化器 550 的第一输入端连接。逆变换器和逆量化器 550 的输出端以信号通信方式与组合器 525 的第二非反相输入端连接。组合器 525 的输出端以信号通信方式与去块滤波器 565 的第二输入端和超级帧内预测模块 560 的第一输入端连接。去块滤波器 565 的第二输出端以信号通信方式与参考画面缓冲器 580 的第一输入端连接。参考画面缓冲器 580 的输出端以信号通信方式与运动补偿器 570 的第二输入端连接。

[0057] 熵解码器 545 的第二输出端以信号通信方式与运动补偿器 570 的第三输入端和去块滤波器 565 的第一输入端连接。熵解码器 545 的第三输出端以信号通信方式与解码器控制器 505 的输入端连接。解码器控制器 505 的第一输出端以信号通信方式与熵解码器 545 的第二输入端连接。解码器控制器 505 的第二输出端以信号通信方式与逆变换器和逆量化器 550 的第二输入端连接。解码器控制器 505 的第三输出端以信号通信方式与去块滤波器 565 的第三输入端连接。解码器控制器 505 的第四输出端以信号通信方式与超级帧内预测模块 560 的第二输入端、运动补偿器 570 的第一输入端、以及参考画面缓冲器 580 的第二输入端连接。

[0058] 运动补偿器 570 的输出端以信号通信方式与开关 597 的第一输入端连接。超级帧内预测模块 560 的输出端以信号通信方式与开关 597 的第二输入端连接。开关 597 的输出端以信号通信方式与组合器 525 的第一非反相输入端连接。

[0059] 输入缓冲器 510 的输入端可用作解码器 500 的用于接收输入比特流的输入端。去块滤波器 565 的第一输出端可用作解码器 500 的用于对输出画面进行输出的输出端。

[0060] 如上注意的,本原理针对用于视频编码器和解码器的对大块的帧内预测进行信令的方法和装置。此外,如上注意的,可以应用本原理的大块被定义为意指具有等于或者大于 32×32 的尺寸的块。

[0061] 在实施例中,为了易于标注,把帧内预测的信令拆分为以下两部分:sip_type(空间帧内划分类型,其可以是 INTRA 4×4 、INTRA 8×8 、INTRA 16×16 等等);以及每个 sip_type 内的 intra_pred_mode(诸如,例如 INTRA 4×4 和 INTRA 8×8 内的 9 种帧内预测模式)。关于特定实施例进一步的详述中,提出了用于本原理的以下三个规则:(1) 选择基本编码单元;(2) 通过从最大的帧内预测类型中拆分或者从基本编码单元中合并而允许分层级的帧内预测;以及(3) 对于每个 sip_type,向最频繁使用的 intra_pred_mode 分配较高的优先级。关于规则(1),允许若干 sip_type 用于基本的编码单元。

[0062] 一实施例

[0063] 在一实施例中,将基本编码单元设置为 16×16 。在该编码单元中,允许 sip_type 为 INTRA 4×4 、INTRA 8×8 、INTRA 16×16 。还允许分层级的帧内预测,如图 6 所示。

[0064] 转到图 6,通过参考标号 600 总地指示可以应用本原理的示例性的分级划分。在该实施例中,如果最大的块尺寸被设置为 64×64 ,则使用“拆分信令”来允许分层级的帧内预测。也就是说,在实施例中,添加了 intra64f_flag。如果 intra64_flag 等于 1,则使用 INTRA 64×64 。否则,如果 intra64_flag 等于 0,则将 64×64 块 611 拆分为四个 32×32 块 621。对于每个 32×32 块 621,添加 intra32_flag。如果 intra32_flag 等于 1,则使用 INTRA 32×32 。否则,如果 intra32_flag 等于 0,则在此(例如对于 32×32 块 621)同样允许在 16×16 基本编码单元中所允许的所有 sip_type。对于 INTRA 16×16 中的 intra_pred_mode,具有 DC 模式和定向模式(directional mode),后者通过发送模式信息而允许不同类型的定向预测。由此, 32×32 帧内预测块 621 可以被进一步拆分为 4 个 16×16 帧内预测块 631。4 个 16×16 帧内预测块 631 中的一个或多个可以进一步被拆分为 DC 模式(未示出)、 16×16 模式 641、 8×8 模式 651,和 4×4 模式 661。在该实施例中,假设具有以下四个 16×16 帧内预测模式:DC;水平的(HOR);垂直的(VER);和多定向的(Multi_DIR)。通过考虑每个模式的优先级来对 intra_pred_mode 进行信令。在 INTRA 16×16 中,由于 DC 模式比其它模式更经常地使用,所以在 INTRA 16×16 之前在 sip_type 表中添加 INTRA 16×16 DC。然后,移除用于 INTRA 16×16 的 intra_pred_mode 中的 most_probable_mode 指示。作为替代,绝对地(absolutely)指示其它 3 个模式(16×16 、 8×8 和 4×4)。

[0065] 语法

[0066] 在表 2 和表 3 中例示用于该实施例的语法示例。具体地,表 2 示出了依据本原理实施例的用于 16×16 编码单元的 sip_type 的示例性规范,表 3 示出了依据本原理实施例的示例性的 INTRA 16×16 预测模式。对于 INTRA 32×32 /INTRA 64×64 ,使用与 INTRA 16×16 相同的模式。对于信令,利用 intra32_DC_flag 和 intra64_DC_flag 来替换 most_probable_mode 指示,这是由于更频繁地使用 DC。然后,绝对地编码其它 intra_pred_mode。

[0067] 可以与在 MPEG-4AVC 标准中完全相同地执行对于 INTRA 4×4 和 INTRA 8×8 的 intra_pred_mode 信令,因此在任何表中不列出这些模式。

[0068] 表 2

[0069]

Sip_type	索引	二进制比特
SIP 8×8	0	0
SIP 16×16 DC	1	10
SIP 16×16	2	110
SIP 4×4	3	1110

[0070] 表 3

[0071]

帧内预测模式	索引	二进制比特
VER	0	0
HOR	1	10
Multi-DIR	2	11

[0072] 表 4 示出了依据本原理实施例的示例性的宏块层语法。

[0073] 表 4

[0074]

macroblock_layer() {	C	描述符
...		
intra64_flag	2	u(1)
if(intra64_flag == 1) {		
intra64_DC_flag	2	u(1)
if(intra64_DC_flag == 0) {		
intra_pred_mode_64	2	ue(v)/se(v)
if(intra_pred_mode_64 == Multi-DIR) {		
intra64_multidir_index	2	ue(v)/se(v)
}		
}		
}		
else {		
for(i32 = 0; i32 < 4; i32++) {		
intra32_flag[i32]	2	u(1)
if(intra32_flag[i32] == 1) {		
intra32_DC_flag[i32]	2	u(1)
if(intra32_DC_flag[i32] == 0) {		

intra_pred_mode_32[i32]	2	ue(v)/se(v)
if(intra_pred_mode_32[i32] == Multi-DIR) {		
intra32_multidir_index[i32]	2	ue(v)/se(v)
}		
}		
}		
else{		
for(i16 = 0; i16 < 4; i16++) {		
sip_type[i16]	2	ue(v)/se(v)
if(sip_type[i16] == SIP16×16) {		
intra_pred_mode_16[i16]	2	ue(v)/se(v)
if(intra_pred_mode_16[i16] == Multi-DIR) {		
intra16_multidir_index[i16]	2	ue(v)/se(v)
}		
}		
else if(sip_type[i16] != SIP16×16_DC) {		
mb_intra_prediction_syntax() ;/* 与 H.264 相同 1*/		
}		
}		
}		
...		
}		

[0075] 表 4 的一些语法元素的语义如下：

[0076] Intra64_flag 等于 1 规定使用 INTRA64×64。Intra64_flag 等于 0 规定 64×64

的大块被进一步拆分为 32×32 的划分。

[0077] Intra64_DC_flag 等于 1 规定 intra_pred_mode 是用于 INTRA64×64 的 DC 模式。Intra64_DC_flag 等于 0 规定 intra_pred_mode 不是用于 INTRA64×64 的 DC 模式。

[0078] intra_pred_mode_64 规定用于 INTRA64×64 的帧内预测模式（不包括 DC 模式）。

[0079] intra_multidir_index 规定用于 INTRA64×64 中的 Multi_Dir 模式的角度的索引。

[0080] Intra32_flag[i] 等于 1 规定对于第 i 个 32×32 的大块使用 INTRA32×32。intra32_flag[i] 等于 0 规定第 i 个 32×32 的大块被进一步拆分为 16×16 的划分。

[0081] Intra32_DC_flag[i] 等于 1 规定对于第 i 个 32×32 块 intra_pred_mode 是用于 INTRA32×32 的 DC 模式。Intra32_DC_flag[i] 等于 0 规定对于第 i 个 32×32 块 intra_pred_mode 不是用于 INTRA32×32 的 DC 模式。

[0082] intra_pred_mode_32[i] 规定用于第 i 个 32×32 大块的 INTRA32×32 的帧内预测模式（不包括 DC 模式）。

[0083] intra_multidir_index 规定用于 INTRA32×32 中的 Multi_Dir 模式的角度的索引。

[0084] sip_type[i] 规定在第 i 个 16×16 块中用于基本块编码单元的空间帧内划分类型。

[0085] intra_pred_mode_16[i] 规定用于第 i 个 16×16 块的 INTRA16×16 的帧内预测模式（不包括 DC 模式）。

[0086] intra_multidir_index 规定用于第 i 个 16×16 块的 INTRA16×16 中的 Multi_Dir 模式的角度的索引。

[0087] 另一实施例

[0088] 在另一实施例中，自适应地选择大块单元为 32×32 或者 64×64 。可以使用一个或多个高级语法元素来对该选择进行信令。在实施例中，如果选择 32×32 ，则仅仅移除与 64×64 有关的所有语法。

[0089] 在另一实施例中，分层级的帧内预测可以包含从基本编码单元合并。例如，如果最大的块单元是 64×64 并且基本编码单元是 16×16 ，则使用一个标志 (is_all_16×16_coding) 来指示一个 64×64 块内部的所有 16×16 块是否都属于 16×16 编码类型。如果 is_all_16-16_coding 等于 1，则这指示使用 16×16 编码类型并且停止信令。否则，使用一个标志 (is_all_32×32_coding) 来指示一个 64×64 块内部的所有 32×32 块是否都属于 32×32 编码类型。如果 is_all_32×32_coding 等于 1，则这指示一个 64×64 块内部的所有 32×32 块都属于 32×32 编码类型。否则，如果 is_all_32×32_coding 和 is_all_16×16_coding 等于 0，则这指示使用 INTRA64×64。

[0090] 在另一实施例中，引入了用于具有不小于 16×16 尺寸的块单元的 SIP 类型 (large_sip_type)。三种类型如下被称为：large_intra_16×16；large_intra_32×32；和 large_intra_64×64。large_intra_16×16 意味着一个大块内部的所有 16×16 块都属于 16×16 编码类型。large_intra_32×32 意味着一个大块内部的所有 32×32 块都属于 32×32 编码类型。在一实施例中，large_intra_32×32 可以与以上利用 intra32_flag 描述的实施例进行组合以允许分层级的帧内预测。large_intra_64×64 意味着一个大块内部

的所有 64×64 块都被编码为 INTRA 64×64 。

[0091] 在另一实施例中,可以引入若干 sip/mode 表。可以将这些表预先存储在编码器和解码器二者中,或者这些表可以是用户指定的并且使用一个或多个高级语法元素进行传送。表 5 示出了依据本原理实施例的示例性的宏块层语法。

[0092] 表 5

[0093]

macroblock_layer() {	C	描述符
...		
is_all_16×16_coding	2	u(1)
if(is_all_16×16_coding == 0) {		
is_all_32×32_coding	2	u(1)
if(is_all_32×32_coding == 0) {		
decode_with_64×64_coding_type()		
}		
else{		
decode_with_32×32_coding_type()		
}		
}		
else{		
decode_with_16×16_coding_type()		
}		
...		
}		

[0094] 表 5 的一些语法元素的语义如下：

[0095] is_all_16×16_coding 等于 1 规定大块内部的所有 16×16 块都通过 16×16 编码类型编码。is_all_16×16_coding 等于 0 规定所述大块不通过 16×16 编码类型进行编码。

[0096] is_all_32×32_coding 等于 1 规定大块内部的所有 32×32 块都通过 32×32 编码类型编码。is_all_32×32_coding 等于 0 规定所述大块不通过 32×32 编码类型进行编

码。

[0097] 转到图 7A 和 7B,其一起表示由参考标号 700 总地指示的、通过对用于大块的帧内预测进行信令而编码所述大块的画面数据的示例性方法。方法 700 包括开始块 705,其将控制传递到功能块 710。功能块 710 执行初始化,并且将控制传递到循环限制块 715。循环限制块 715 对 64×64 块(即,具有 64×64 块尺寸的块)执行循环(下文也称为循环 1),并且将控制传递到功能块 785 和循环限制块 720。

[0098] 功能块 785 执行帧内 64×64 模式判定,基于 RD64(即,从 Intra 64×64 模式判定产生的率失真)设置 Intra64_DC_flag,并且将控制传递到判定块 770。

[0099] 循环限制块 720 对四个 32×32 块(即,具有 32×32 块尺寸并且从由循环 1 处理的当前 64×64 块中获得的四个块)执行循环(下文也称为循环 2),并且将控制传递到功能块 790 和循环限制块 725。

[0100] 功能块 790 执行帧内 32×32 模式判定,基于 RD32(即,从 Intra 32×32 模式判定产生的率失真)设置 Intra32_DC_flag,并且将控制传递到判定块 750。

[0101] 循环限制块 725 对四个 16×16 块(即,具有 16×16 块尺寸并且从由循环 2 处理的当前 32×32 块中获得的四个块)执行循环(下文也称为循环 3),并且将控制传递到功能块 730 和功能块 735。

[0102] 功能块 730 评估 Intra 16×16 _DC 模式,并且将控制传递到功能块 740。功能块 735 评估其它的 16×16 模式(即,除了 Intra 16×16 _DC 以外的)和以下的模式(例如, 8×8 , 4×4 等等),并且将控制传递到功能块 740。

[0103] 功能块 740 基于 RD16(即,从 Intra 16×16 模式判定产生的率失真)执行 16×16 模式判定,然后累计每个 16×16 块的 RD16 以获得 TotRD16(其指示当由四个 16×16 块编码时整个 32×32 块的总的率失真),并且将控制传递到循环限制块 745。循环限制块 745 结束对 16×16 块的循环(即,循环 3),并且将控制传递给判定块 750。

[0104] 判定块 750 确定 RD32 是否小于 TotRD16(即,当前 32×32 块的率失真成本是否小于从当前 32×32 块中获得的四个 16×16 块的总的率失真成本)。如果是这样,则控制被传递给功能块 755。否则,控制被传递给功能块 742。

[0105] 功能块 755 将 Intra32_flag 设置为等于 1,并且将控制传递给功能块 760。功能块 742 将 Intra32_flag 设置为等于 0,并且将控制传递给功能块 760。

[0106] 功能块 760 将每个 32×32 块的 RD32 的累计设置至 TotRD32 以便指示在通过四个 32×32 块进行编码时整个 64×64 块的总的率失真,并且将控制传递给循环限制块 765。循环限制块 765 结束对 32×32 块的循环(即,循环 2),并且将控制传递给判定块 770。

[0107] 判定块 770 确定 RD64 是否小于 TotRD32(即,当前 64×64 块的率失真成本是否小于从当前 64×64 块中获得的四个 32×32 块的总的率失真成本)。如果是这样,则控制被传递给功能块 775。否则,控制被传递给功能块 780。

[0108] 功能块 775 将 Intra64_flag 设置为等于 1,并且将控制传递给循环限制块 795。功能块 780 将 Intra64_flag 设置为等于 0,并且将控制传递给循环限制块 795。

[0109] 循环限制块 795 结束对 64×64 块的循环(即,循环 1),并且将控制传递给功能块 797。功能块 797 熵编码标志、intra_pred_mode 和残差,并且将控制传递给结束块 799。

[0110] 转到图 8A 和 8B,其一起表示由参考标号 800 总地指示的、通过确定要被应用于大

块的帧内预测而解码所述大块的画面数据的示例性方法。方法 800 包括开始块 805, 其将控制传递到功能块 808。功能块 808 初始化解码器并且然后将控制传递到功能块 810。功能块 810 解析比特流, 并且将控制传递给循环限制块 815。循环限制块 815 对 64×64 块执行循环 (下文也称为循环 1), 并且将控制传递判定块 820。判定块 820 确定 Intra64_flag 是否被设置为等于 1。如果是这样, 则控制被传递给功能块 885。否则, 控制被传递给循环限制块 825。

[0111] 功能块 885 确定 Intra64_DC_flag 是否被设置为等于 1。如果是这样, 则控制被传递给功能块 887。否则, 控制被传递给功能块 888。功能块 887 执行 Intra64 $\times$ 64DC 预测, 并且然后将控制传递到功能块 890。功能块 888 执行除了 Intra64 $\times$ 64DC 模式以外的 Intra 64 $\times$ 64 预测, 并且然后将控制传递给功能块 890。功能块 890 解码当前 64 $\times$ 64 块, 并且将控制传递给循环限制块 880。循环限制块 880 结束对 64 $\times$ 64 块的循环 (即, 循环 1), 并且将控制传递给结束块 899。

[0112] 循环限制块 825 对四个 32 $\times$ 32 块执行循环 (下文也称为循环 2), 并且将控制传递给判定块 830。判定块 830 确定 Intra32_flag 是否等于 1。如果是这样, 则控制被传递给功能块 835。否则, 控制被传递给循环限制块 845。

[0113] 功能块 835 确定 Intra32_DC_flag 是否等于 1。如果是这样, 则控制被传递给功能块 837。否则, 控制被传递给功能块 838。功能块 837 执行 Intra32 $\times$ 32DC 预测, 并且将控制传递到功能块 840。功能块 838 执行除了 Intra32 $\times$ 32DC 模式以外的帧内预测模式, 并且然后将控制传递给功能块 840。功能块 840 解码 32 $\times$ 32 块, 并且将控制传递给循环限制块 875。

[0114] 循环限制块 875 结束对 32 $\times$ 32 块的循环 (即, 循环 2), 并且将控制传递给循环限制块 880。

[0115] 循环限制块 845 对四个 16 $\times$ 16 块执行循环 (下文也称为循环 3), 并且将控制传递判定块 850。判定块 850 确定是否 sip_type = Intra16_DC。如果是这样, 则控制被传递给功能块 855。否则, 控制被传递给功能块 860。

[0116] 功能块 855 执行 Intra16 $\times$ 16_DC 模式预测, 并且将控制传递到功能块 865。功能块 860 使用其它帧内预测模式 (即, 除了 Intra16 $\times$ 16_DC 模式以外的) 执行模式预测, 并且将控制传递给功能块 865。

[0117] 功能块 865 解码 16 $\times$ 16 块, 并且将控制传递给循环限制块 870。循环限制块 870 结束对 16 $\times$ 16 块的循环 (即, 循环 3), 并且将控制传递给循环限制块 875。

[0118] 现在将描述本发明的许多伴随优点 / 特征中的一些, 其中的一些已经在上面提及。例如, 一个优点 / 特征是一种具有视频编码器的装置, 所述视频编码器通过对用于画面中的至少一个大块的帧内预测进行信令来编码用于所述至少一个大块的画面数据。通过选择基本编码单元尺寸并且分配用于基本编码单元尺寸的单个空间帧内划分类型来对帧内预测进行信令。该单个空间帧内划分类型是从多个空间帧内划分类型中选择的。所述至少一个大块具有比基本编码单元的块尺寸大的大块尺寸。帧内预测是分层级的帧内预测并且通过以下操作中的至少一个而对至少一个大块执行: 将大块尺寸拆分为基本编码单元尺寸以及从基本编码单元尺寸合并到大块尺寸。

[0119] 另一优点 / 特征是具有如上所述的视频编码器的装置, 其中对于多个空间帧内划

分类型的每一个,向多个可用的帧内预测模式中最频繁使用的特定的帧内预测模式分配较高的优先级。

[0120] 又一优点 / 特征是具有如上所述的视频编码器的装置,其中自适应地选择大块尺寸。

[0121] 再一优点 / 特征是具有如上所述的视频编码器的装置,其中,使用一个或多个高级语法元素来执行信令。

[0122] 此外,另一优点 / 特征是具有如上所述的视频编码器的装置,其中空间帧内划分类型表和帧内预测模式表中的至少一个被视频编码器预先存储并且使用以便编码至少一个大块。该空间帧内划分类型表和帧内预测模式表中的至少一个被安排为被对应的视频解码器预先存储并且使用以便解码该至少一个大块。

[0123] 此外,另一优点 / 特征是具有如上所述的视频编码器的装置,其中空间帧内划分类型表和帧内预测模式表中的至少一个被视频编码器用于编码至少一个大块,并且被视频编码器使用一个或多个高级语法元素来传送。

[0124] 基于这里的教导,本领域普通技术人员可以容易确定本原理的这些和其它特征和优点。应理解,本原理的教导可以以硬件、软件、固件、专用处理器、或其组合的各种形式来实现。

[0125] 更优选地,本原理的教导被实现为硬件与软件的组合。此外,软件可以实现为有形地体现在程序存储单元上的应用程序。应用程序可以被上载到包括任何适当架构的机器并由该机器执行。优选地,在具有诸如一个或多个中央处理单元 (“CPU”)、随机存取存储器 (“RAM”)、以及输入 / 输出 (“I/O”) 接口等的硬件的计算机平台上实现该机器。计算机平台还可以包括操作系统和微指令代码。这里描述的各种处理与功能可以是可能由 CPU 执行的微指令代码的一部分或是应用程序的一部分、或者是其任何组合。另外,各种其它的诸如附加数据存储单元以及打印单元之类的外设单元可以连接到计算机平台。

[0126] 还应理解,由于在附图中描绘的一些组成系统组件和方法优选地以软件实现,因此这些系统组件或处理功能块之间的实际连接可能取决于本原理被编程的方式而有所不同。给出这里的教导,本领域普通技术人员将能够预期本原理的这些和类似的实现方式或配置。

[0127] 尽管这里已经参考附图描述了示例实施例,应理解本原理不限于那些确切的实施例,并且本领域普通技术人员可以在其中进行各种改变和修改,而不偏离本原理的范围和精神。所有这些改变和修改意在被包括在所附权利要求阐述的本原理的范围之内。

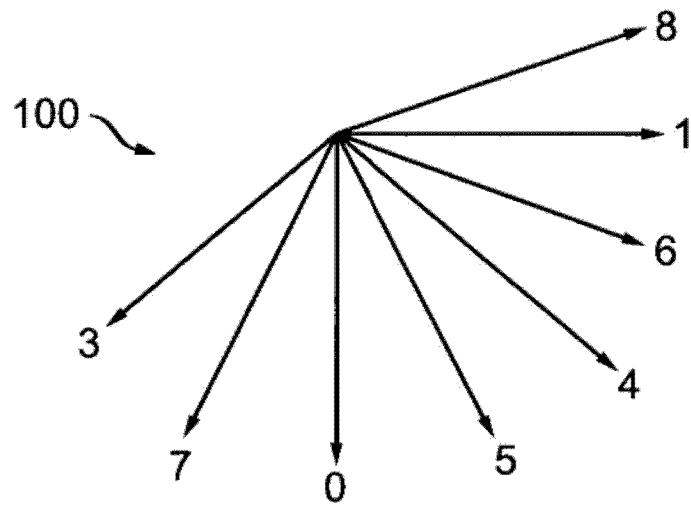


图 1

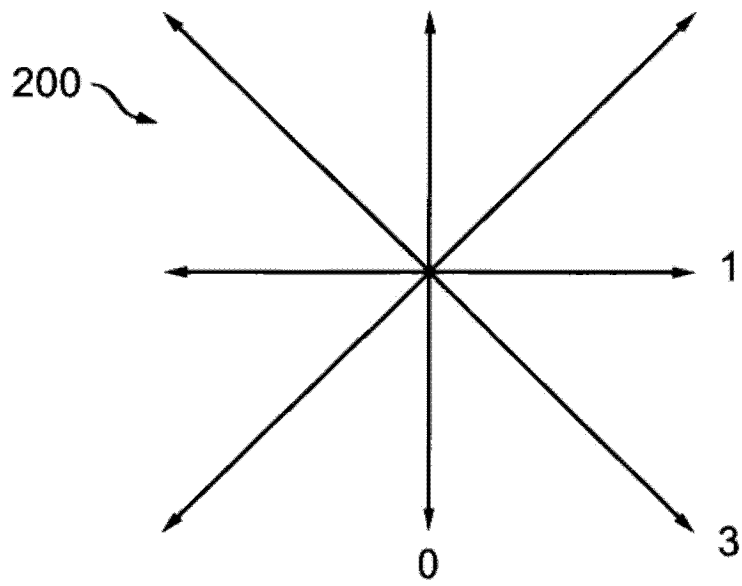


图 2

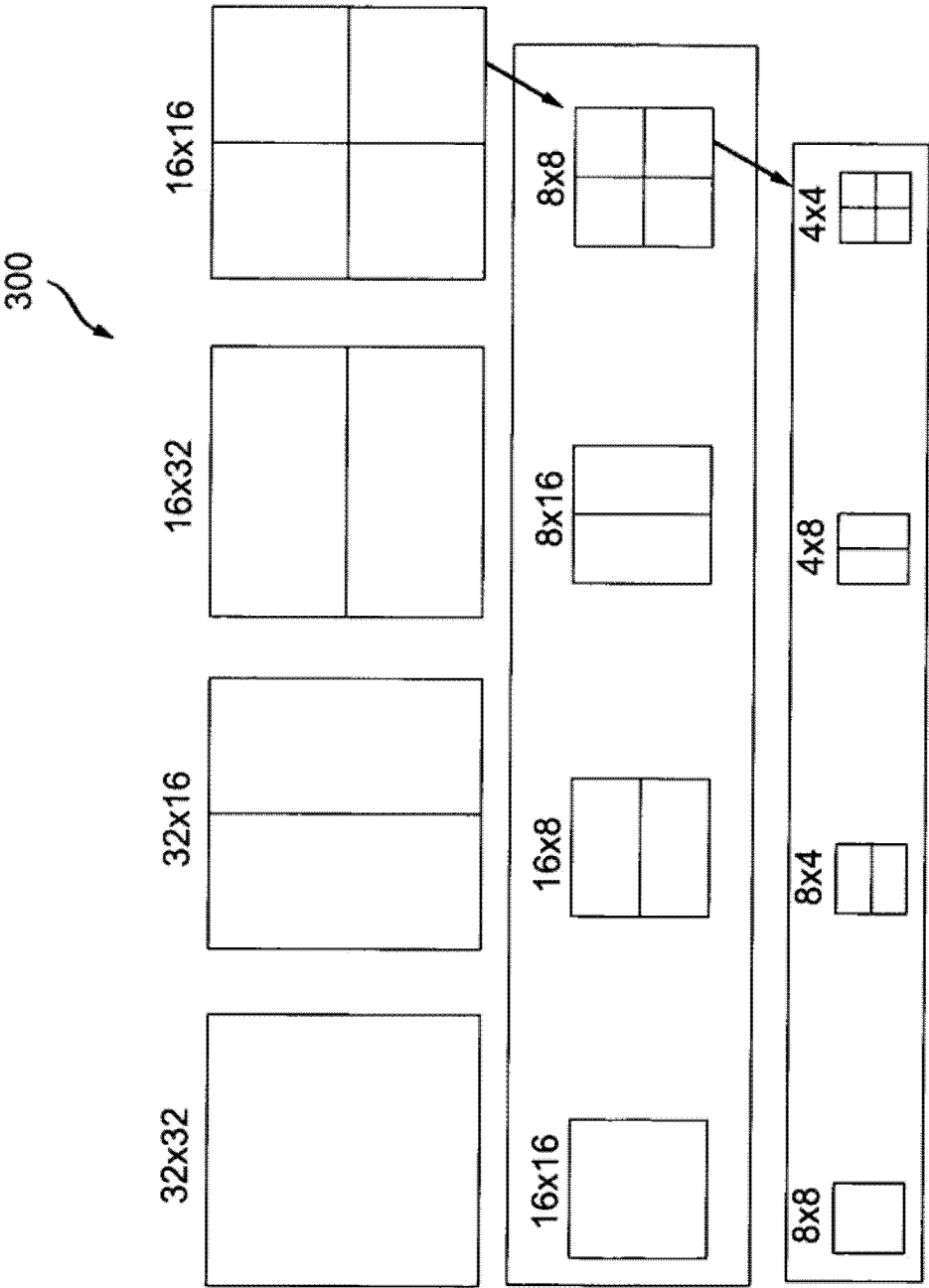


图 3

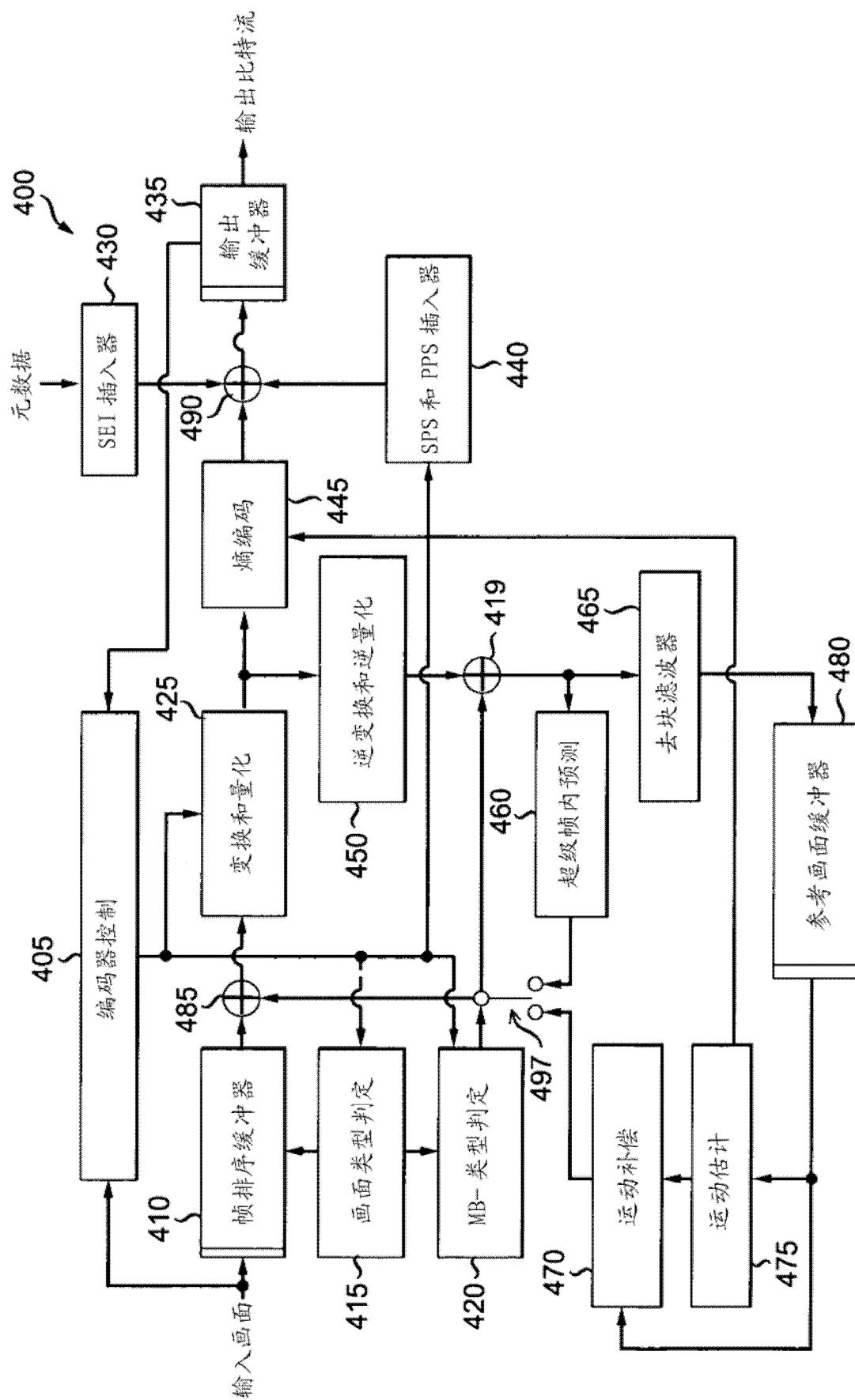


图 4

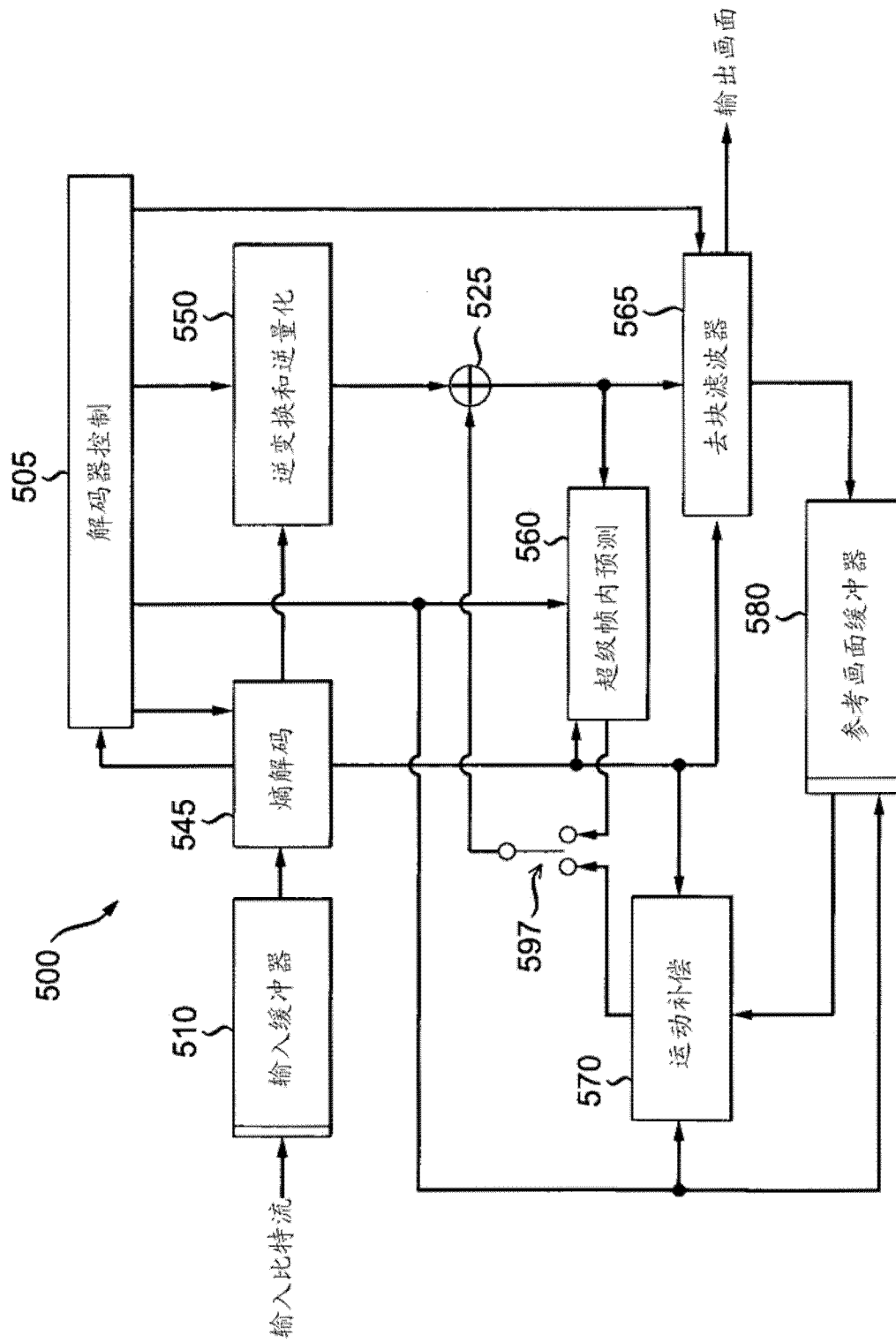


图 5

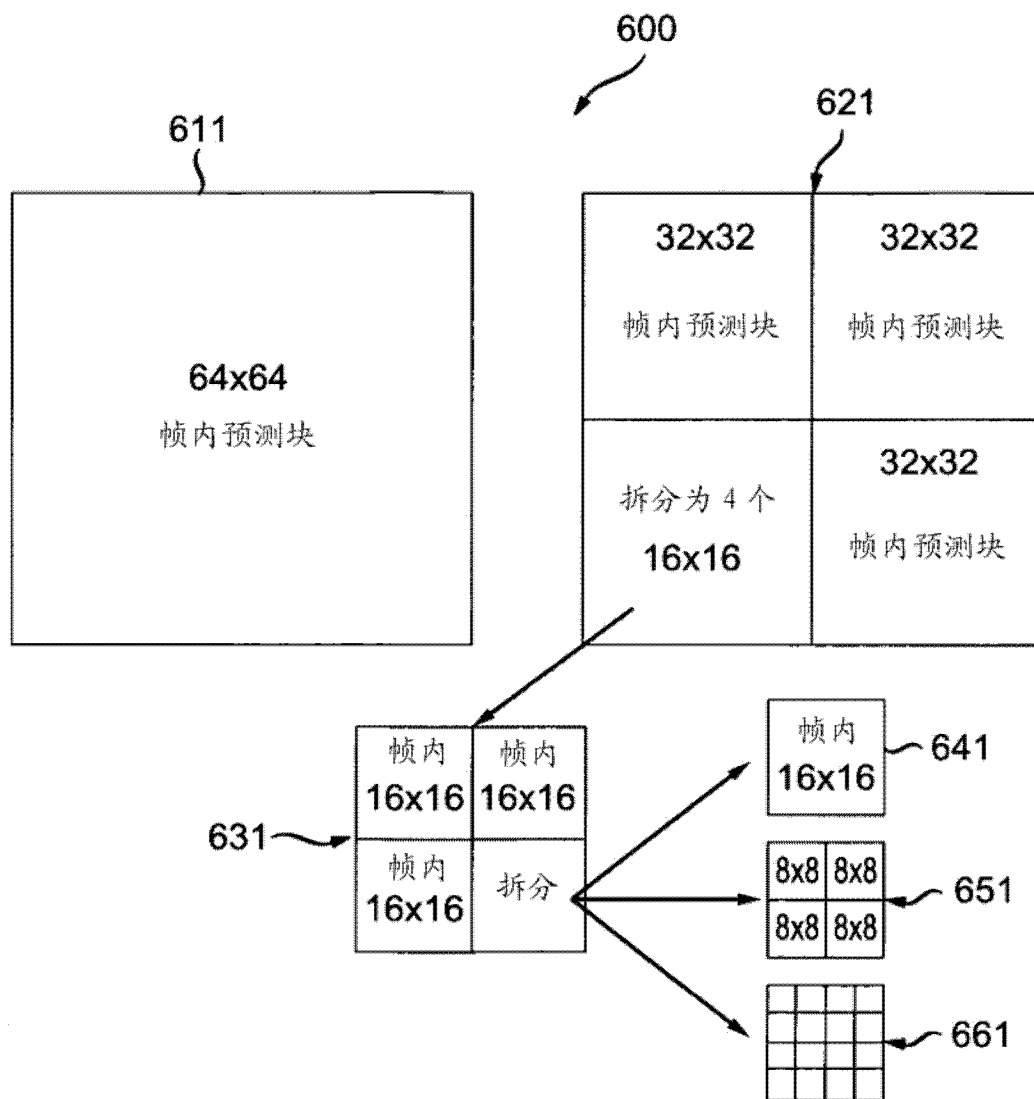


图 6

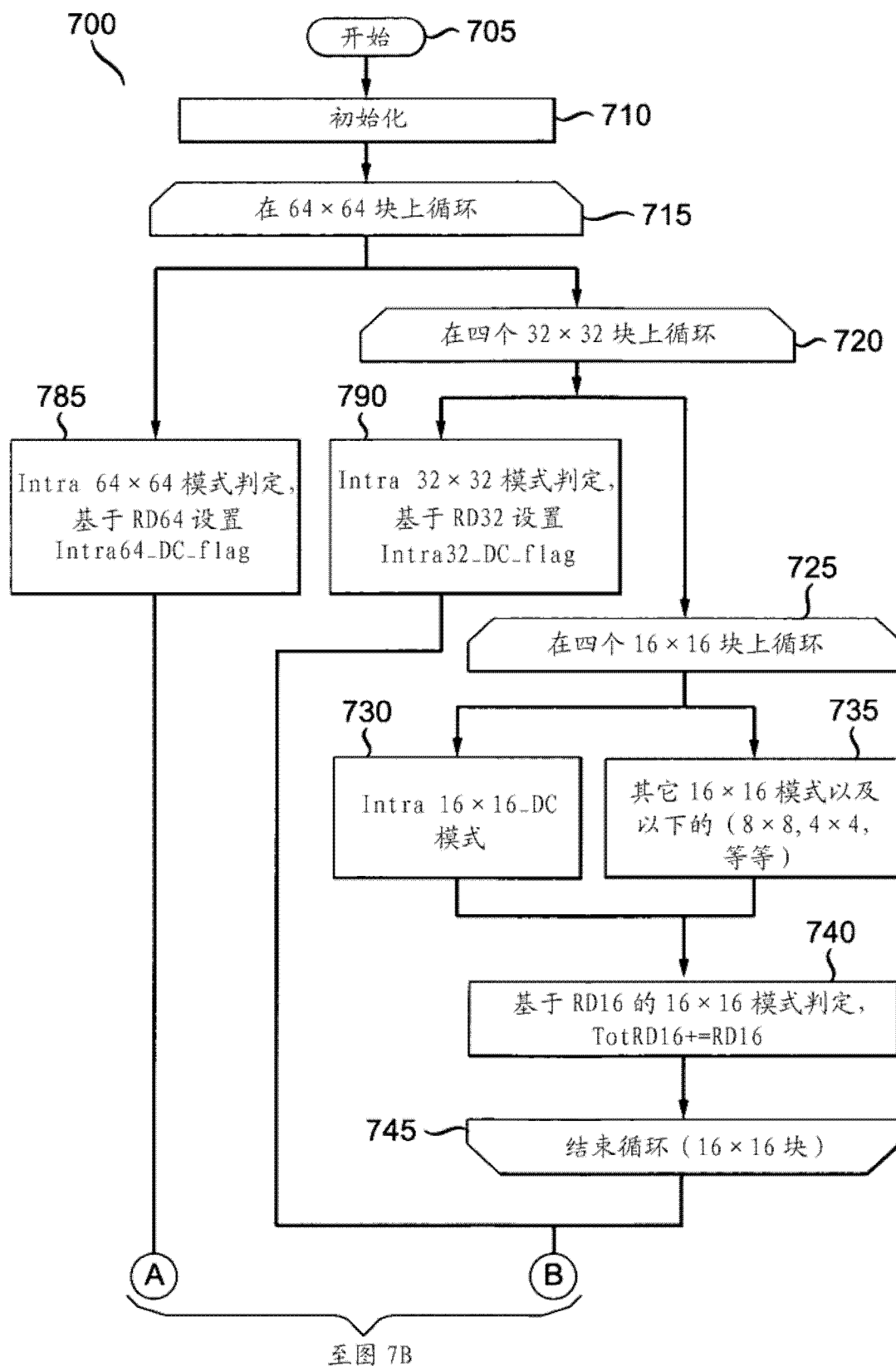


图 7A

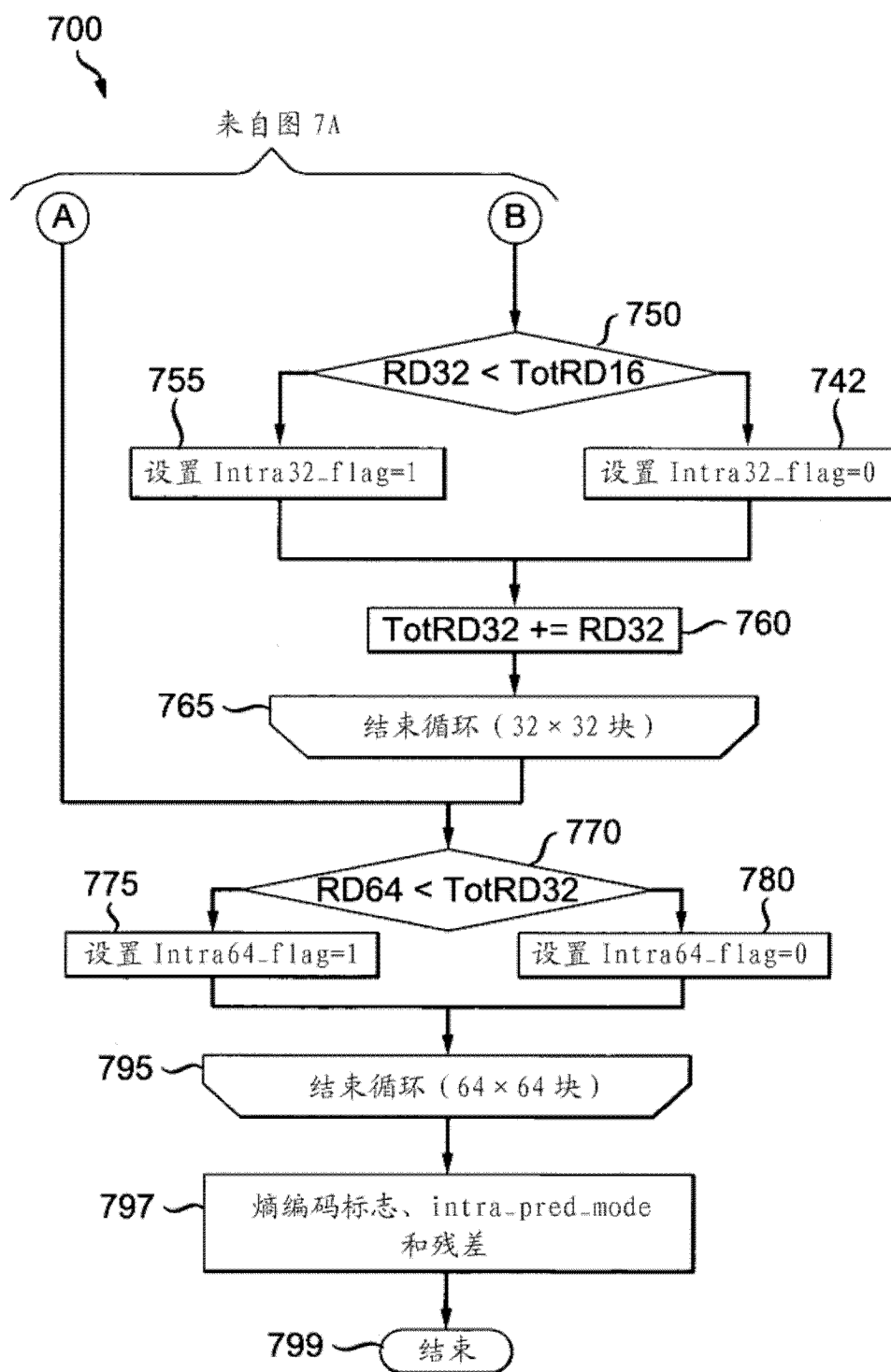


图 7B

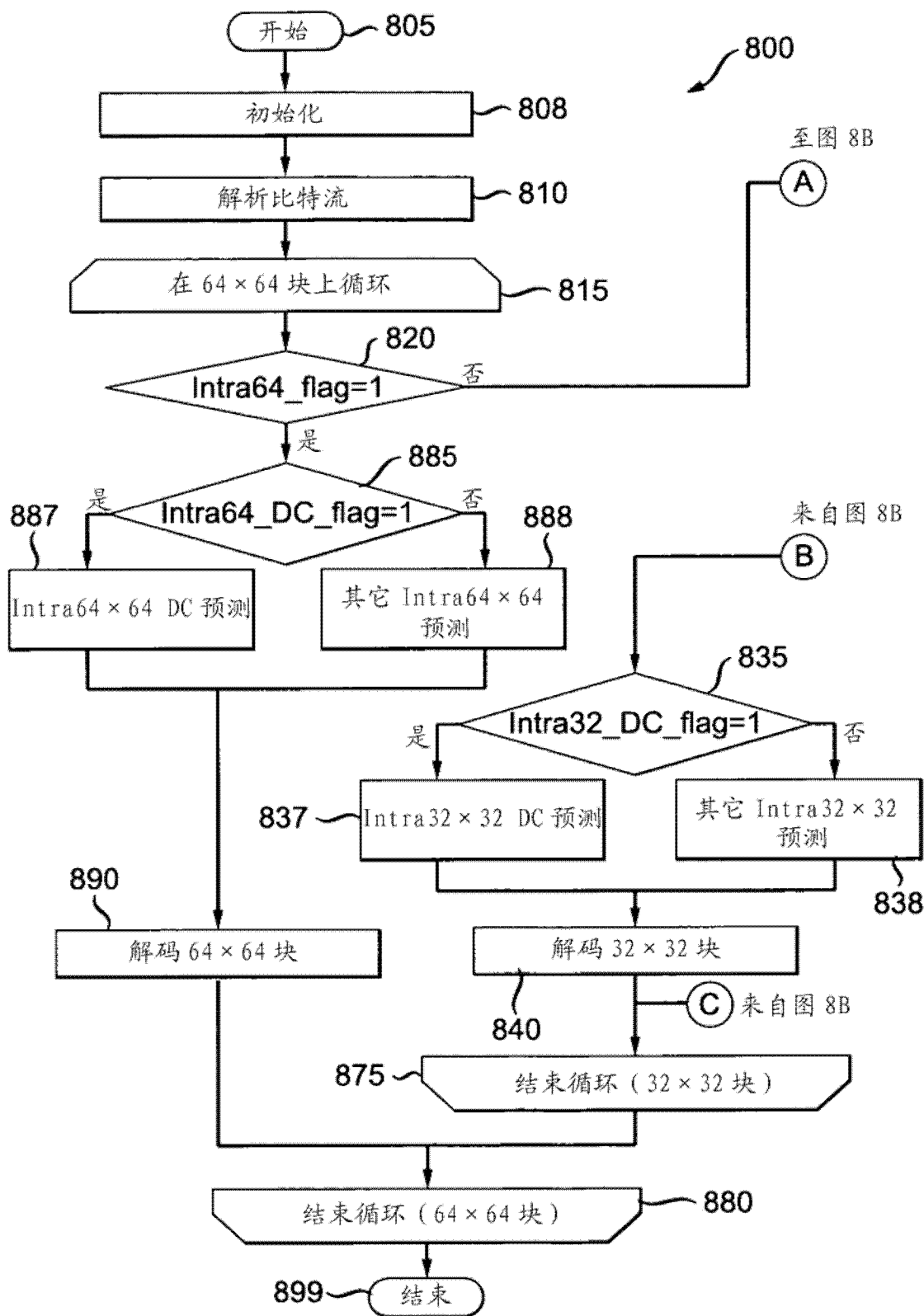


图 8A

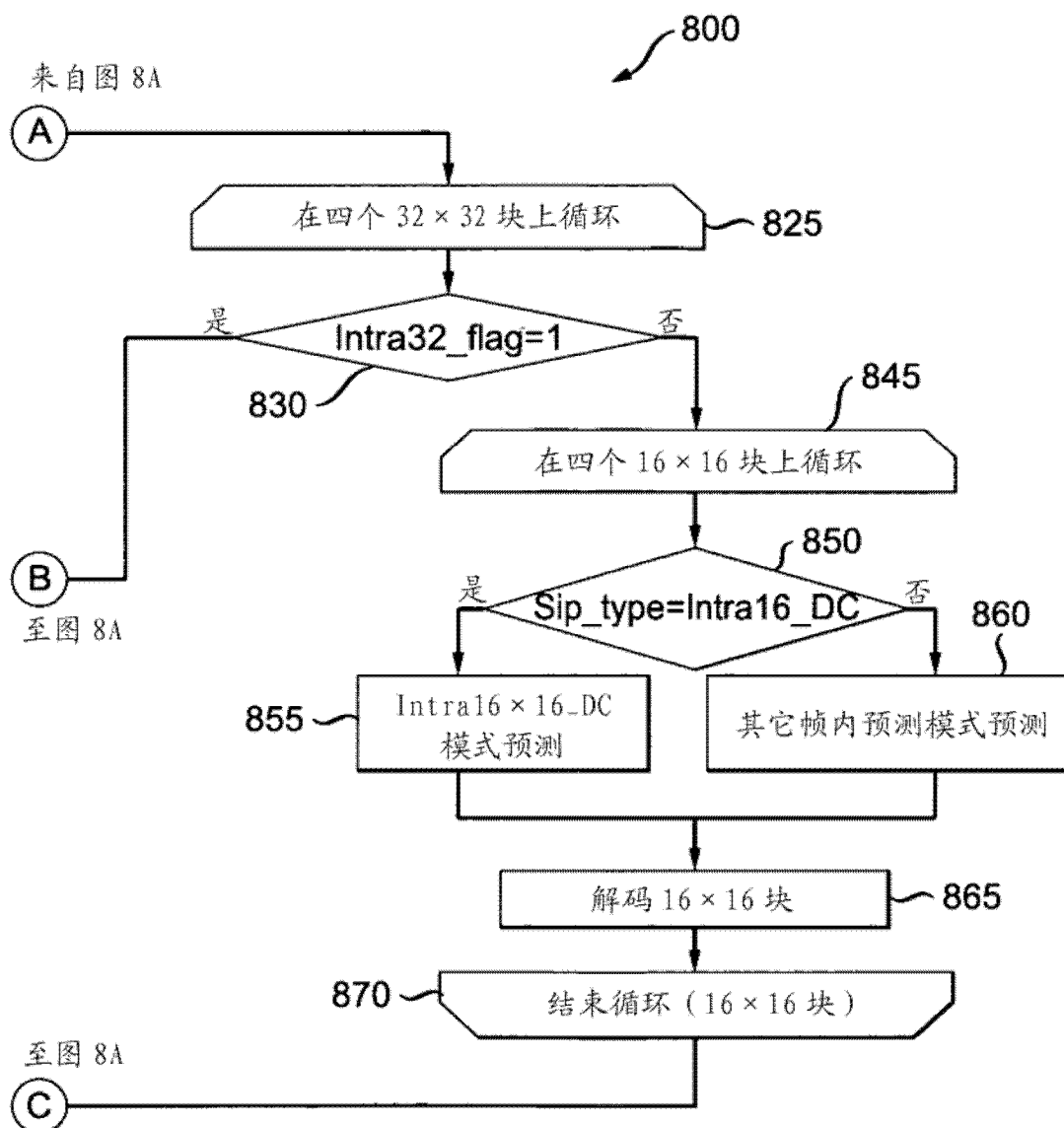


图 8B