



(12) 发明专利申请

(10) 申请公布号 CN 119013968 A

(43) 申请公布日 2024. 11. 22

(21) 申请号 202380033410.3

(74) 专利代理机构 北京魏启学律师事务所
11398

(22) 申请日 2023.04.11

专利代理师 陈涛

(30) 优先权数据

2205319.3 2022.04.11 GB

2208470.1 2022.06.09 GB

(51) Int.Cl.

H04N 19/105 (2006.01)

H04N 19/52 (2006.01)

H04N 19/176 (2006.01)

H04N 19/70 (2006.01)

H04N 19/51 (2006.01)

H04N 19/46 (2006.01)

(85) PCT国际申请进入国家阶段日

2024.10.10

(86) PCT国际申请的申请数据

PCT/EP2023/059427 2023.04.11

(87) PCT国际申请的公布数据

W02023/198699 EN 2023.10.19

(71) 申请人 佳能株式会社

地址 日本

(72) 发明人 G·拉罗彻 P·乌诺

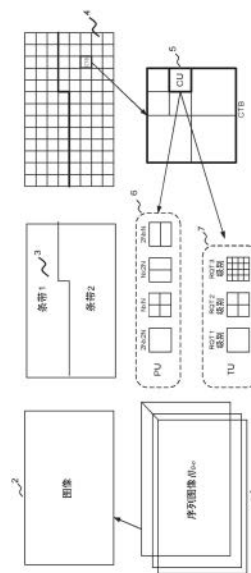
权利要求书5页 说明书38页 附图38页

(54) 发明名称

数据编码和解码

(57) 摘要

公开了对预测器的处理的改进。获得预测器列表,该预测器列表具有至少两个预测器。使用与列表中的第一预测器相关的第一成本和与列表中的第二预测器相关的第二成本、基于准则来确定是否修改预测器列表。基于该确定来修改预测器列表。准则是基于阈值的。



1. 一种用于处理预测器的方法,所述方法包括:
获得预测器的列表,所述预测器的列表具有至少两个预测器;
使用与所述列表中的第一预测器相关的第一成本和与所述列表中的第二预测器相关的第二成本、基于准则来确定是否修改所述预测器的列表;以及
基于所述确定来修改所述预测器的列表,
其中,所述准则是基于阈值的。
2. 根据权利要求1所述的方法,其中,所述确定包括将所述阈值添加到所述第一成本或所述第二成本,以及所述准则与结果同所述第一成本和所述第二成本中的没有添加所述阈值的成本的比较有关。
3. 根据权利要求2所述的方法,其中,所述准则是将所述阈值添加到所述第一成本的结果是否小于所述第二成本。
4. 根据权利要求1所述的方法,其中,所述准则是所述第一成本与所述第二成本之间的差是否越过所述阈值。
5. 根据权利要求4所述的方法,其中,所述差被计算为绝对值,以及所述准则是所述差是否小于所述阈值。
6. 根据权利要求1至5中任一项所述的方法,其中,修改所述列表包括从所述预测器的列表中移除第一候选或第二候选。
7. 根据权利要求1至6中任一项所述的方法,其中,修改所述列表包括改变所述列表中的一个或多个预测器的位置。
8. 根据权利要求7所述的方法,其中,改变一个或多个候选的位置包括将所述第一预测器或所述第二预测器移动到所述列表中的不同位置。
9. 根据权利要求7所述的方法,其中,改变排序包括将所述第一预测器或所述第二预测器移动到被认为是所述列表的末尾的位置。
10. 根据前述权利要求中任一项所述的方法,其中,所述第一预测器和所述第二预测器在被认为是接在所述列表的起始的位置处的第零预测器之后。
11. 根据权利要求10所述的方法,其中,从修改所述预测器的列表的步骤中排除所述第零预测器。
12. 根据权利要求10所述的方法,还包括用于将与所述第零预测器相关的第零成本与第零阈值进行比较的步骤,其中,在所述第零成本超过所述第零阈值的情况下,在用于修改所述预测器的列表的步骤中包括所述第零预测器。
13. 根据权利要求12所述的方法,其中,用于将与所述第零预测器相关的第零成本与第零阈值进行比较的步骤在第一集合的模式中进行,而不在第二集合的模式中进行。
14. 根据权利要求12或13所述的方法,其中,所述第零阈值低于针对所述第一预测器和所述第二预测器所使用的阈值。
15. 根据权利要求1至9中任一项所述的方法,其中,所述第一预测器在被认为是所述列表的起始的位置处,以及其中,所述第一成本被设置为零。
16. 根据权利要求1至14中任一项所述的方法,其中,所述第一成本和所述第二成本包括基于要编码或解码的图像部分的相邻样本和与所述预测器相关联的图像部分的相邻样本的模板匹配成本。

17. 根据权利要求1至14中任一项所述的方法,其中,所述第一成本和所述第二成本包括与所述预测器相关联的两个块预测器的样本值之间的差。

18. 根据权利要求1至14中任一项所述的方法,其中,所述第一成本和所述第二成本包括与所述预测器相关联的样本与所述列表中的至少一个其他预测器之间的差。

19. 根据权利要求1至14中任一项所述的方法,其中,所述第一成本和所述第二成本是基于所述预测器的样本或相邻样本的子集的。

20. 根据权利要求1至14中任一项所述的方法,其中,所述第一成本和所述第二成本是基于与根据不同分辨率的另一图像相对应的样本的。

21. 根据前述权利要求中任一项所述的方法,其中,对用于计算所述第一成本和所述第二成本的样本的值进行预处理。

22. 根据权利要求16至21中任一项所述的方法,其中,所述第一成本和所述第二成本是基于编码速率的。

23. 根据权利要求16至21中任一项所述的方法,其中,所述第一成本和所述第二成本是基于失真成本的。

24. 根据权利要求23所述的方法,其中,所述失真成本是以下项中的任一项:绝对差和即SAD、绝对变换差和即SATD、误差平方和即SSE、以及结构相似性指数度量即SSIM。

25. 根据前述权利要求中任一项所述的方法,其中,所述成本与正被处理的单元中的样本的数量成比例。

26. 根据权利要求23或24所述的方法,其中,所述成本与模板中的样本的数量成比例。

27. 根据权利要求23或24所述的方法,其中,所述成本是基于用于确定失真和块大小的样本的。

28. 根据前述权利要求中任一项所述的方法,还包括:导出所述阈值。

29. 根据权利要求1至27中任一项所述的方法,还包括:接收指示所述阈值的信息。

30. 根据权利要求28或29所述的方法,其中,所述阈值是基于量化参数值即QP值的。

31. 根据权利要求28或29所述的方法,其中,所述阈值是基于当前帧与一个或多个其他帧之间的时间距离或图片顺序计数距离即POC距离的。

32. 根据权利要求28或29所述的方法,其中,所述阈值是基于帧的时间ID的。

33. 根据权利要求28或29所述的方法,其中,所述阈值是基于参考帧的时间方向的。

34. 根据权利要求28或29所述的方法,其中,所述阈值是基于是否使用双向预测的。

35. 根据权利要求28或29所述的方法,其中,所述阈值取决于可用参考帧的一个或多个QP值。

36. 根据权利要求28或29所述的方法,其中,所述阈值取决于用作所述第一成本和所述第二成本其中至少之一的失真的类型。

37. 根据权利要求28或29所述的方法,其中,所述阈值是基于被用于对作为所述第一成本和所述第二成本其中至少之一的度量的失真进行计算的样本的类型的。

38. 根据权利要求28或29所述的方法,其中,所述阈值是基于是帧内预测还是帧间预测被用于对要使用所述预测器的列表而预测的图像部分进行编码或解码的。

39. 根据权利要求28或29所述的方法,其中,所述阈值是基于随机接入配置即RA配置、低延迟配置和仅帧内配置中的一者或多于一者中的哪个被用于进行编码或解码的。

40. 根据权利要求28或29所述的方法,其中,所述阈值是基于用于编码的公式的,并且是由用于计算所述阈值的解码器获得的。

41. 根据权利要求28或29所述的方法,其中,所述阈值是基于移位后的量化参数值即移位后的QP值的。

42. 根据权利要求40所述的方法,其中,所述移位后的QP值被用于确定拉格朗日参数 λ 作为所述阈值。

43. 根据权利要求41至52中任一项所述的方法,其中,所述移位后的量化参数被用于从表中获得所述拉格朗日参数,其中,所述表中的条目将拉格朗日参数值与量化参数值相关联。

44. 根据权利要求1至28中任一项所述的方法,其中,所述阈值被确定为作为所述列表中的连续预测器之间的差的最小成本差,该连续预测器不是所述第一预测器和所述第二预测器。

45. 根据权利要求1至28中任一项所述的方法,其中,所述阈值是使用所述列表中的所有候选或前N个候选的成本值来确定的,其中,N是所述列表中允许的候选的最大数量。

46. 根据权利要求45所述的方法,其中,使用所述成本值包括从用于确定所述阈值的候选中计算连续候选成本的差的平均。

47. 根据权利要求23至43中任一项所述的方法,其中,所述阈值是基于拉格朗日参数 λ 的,以及所述准则是第一失真成本与第二失真成本之间的绝对差是否小于所述拉格朗日参数。

48. 根据权利要求47所述的方法,其中,所述阈值是所述拉格朗日参数乘以所述第一预测器与所述第二预测器之间的速率差。

49. 根据权利要求23至43中任一项所述的方法,其中,将所述第一失真成本和所述第二失真成本其中至少之一乘以所述阈值。

50. 根据权利要求37所述的方法,其中,在第一排序处理之后应用所述准则。

51. 根据权利要求37或50所述的方法,其中,将另外的阈值添加到由乘法产生的所述第一失真成本与所述第二失真成本之间的差的结果。

52. 根据前述权利要求中任一项所述的方法,其中,所述确定是在根据候选的失真成本对候选进行排序的处理期间根据所述准则来进行的。

53. 根据权利要求52所述的方法,其中,确定是否修改所述列表是基于最小失真差和阈值的。

54. 根据权利要求53所述的方法,其中,重新排序的最大数量被限制为预定义数量。

55. 根据权利要求54所述的方法,其中,重新排序的最大数量是所述列表中的候选的最大允许数量。

56. 根据权利要求1至52中任一项所述的方法,其中,所述确定是基于最小失真差的,并且被限制为重新排序的最大数量。

57. 根据权利要求52至56中任一项所述的方法,其中,修改所述列表仅包括移除候选。

58. 根据权利要求57所述的方法,其中,移除之后的候选的数量必须大于或等于所述列表中的候选的最大允许数量。

59. 根据前述权利要求所述的方法,其中,在通过移动候选来修改所述列表时,使所述

列表中允许的候选的最大数量递减。

60. 根据权利要求53至55所述的方法,其中,在满足所述准则的情况下,将所述第二预测器移动到第一位置,在所述第一位置处,针对所述列表中的后续预测器的成本中的差高于所述阈值。

61. 根据权利要求60所述的方法,其中,在具有重新排序的可能性的最大数量大于最终列表中允许的候选的最大数量时,所述修改包括将所述第一预测器或所述第二预测器被移动的位置限制到当前列表的末尾。

62. 根据权利要求60和61所述的方法,其中,所述预测器的处理被应用于常规合并模式、模板匹配合并模式、双边匹配合并模式即BM合并模式、以及帧内块拷贝模式即IBC模式其中至少之一。

63. 根据权利要求1至52中任一项所述的方法,其中,所述第二预测器紧接在所述第一预测器之后,以及其中,在确定为不满足所述准则的情况下,在所述列表中的后续预测器之后插入所述第二预测器。

64. 根据权利要求63所述的方法,其中,在确定为不满足所述准则的情况下,将所述准则应用于所述第二成本和所述列表中的先前候选的个体成本,以及其中,在确定为针对所述第二成本和先前候选的各成本不满足所述准则的情况下,不进行在所述后续预测器之后的所述第二预测器的插入。

65. 根据权利要求52至64中任一项所述的方法,其中,重新排序的最大可能数量被设置为已处理候选的最终列表中的候选的允许数量。

66. 根据权利要求64或65所述的方法,其中,所述预测器的处理被应用于与MVD的合并、与MVD的仿射合并、运动矢量残差的MVD符号预测其中至少之一。

67. 根据前述权利要求中任一项所述的方法,其中,所述预测器的列表用于预测图像部分中的运动。

68. 根据权利要求67所述的方法,其中,所述列表中的预测器是帧内块预测器候选和帧间运动信息预测器候选其中任一者。

69. 根据权利要求1至66中任一项所述的方法,其中,所述列表中的预测器用于以下项中的任一项:预测点云中的位置、对网格进行编码、预测音频数据、预测静止图像数据、以及预测视频数据。

70. 根据权利要求1所述的方法,还包括:将至少一个句法元素与所述预测器的列表中的一个或多个预测器相关联。

71. 一种用于处理预测器的方法,所述方法包括:

获得预测器的列表,所述预测器的列表具有至少两个预测器;

基于与所述列表中的第一预测器相关的第一成本与阈值的比较来确定是否修改所述预测器的列表;以及

基于所述确定来修改所述预测器的列表。

72. 根据权利要求71所述的方法,其中,用于确定是否修改所述预测器的列表的步骤是基于与所述列表中的各个预测器相关的相应成本与阈值的比较的。

73. 根据权利要求71或72所述的方法,其中,修改所述列表包括从所述预测器的列表中移除成本低于所述阈值的所述候选或各个候选。

74. 根据权利要求71或72所述的方法,其中,修改所述列表包括将所述列表中的成本低于所述阈值的所述候选或各个候选移动到所述列表中的接在成本大于所述阈值的候选之后的位置。

75. 根据权利要求71或72所述的方法,其中,修改所述列表包括将所述列表中的成本低于所述阈值的所述候选或各个候选移动到被认为是所述列表的末尾的位置。

76. 根据权利要求71至75中任一项所述的方法,其中,用于修改所述预测器的列表的步骤在第一集合的模式中进行,而不在第二集合的模式中进行。

77. 一种用于将图像数据编码在位流中的方法,其包括根据权利要求1至76中任一项所述的方法来处理预测器的列表。

78. 一种用于从位流中解码图像数据的方法,其包括根据权利要求1至76中任一项所述的方法来处理预测器的列表。

79. 一种用于将图像数据编码在位流中的设备,所述设备被配置为进行根据权利要求77所述的方法。

80. 一种用于从位流中解码图像数据的设备,所述设备被配置为进行根据权利要求78所述的方法。

81. 一种程序,其在执行时使可编程设备进行根据权利要求1至76中任一项所述的方法。

82. 一种计算机可读存储介质,其上存储有根据权利要求81所述的程序。

数据编码和解码

技术领域

[0001] 本发明涉及预测器的编码和解码。

背景技术

[0002] 联合视频专家组 (JVET) (由MPEG和ITU-T第16研究组的VCEG组成的合作团队) 发布了称为多功能视频编码 (VVC) 的新视频编码标准。VVC的目标是在现有HEVC标准上提供压缩性能的显著改进 (即, 通常是以前的两倍)。主要目标应用和服务包括但不限于360度和高动态范围 (HDR) 视频。在超高清 (UHD) 视频测试材料上示出了特定的效果。因此, 针对最终标准, 我们可以预期压缩效率的提高将远超过作为目标的50%。

[0003] 自VVC v1标准结束以来, JVET通过建立探索软件 (ECM) 启动了探索阶段。JVET收集了附加工具并且改进了VVC标准的基础上的现有工具, 以实现更好的编码效率。

[0004] 与HEVC相比, 除了其他修改, VVC具有修改的用于运动矢量预测的“合并模式”的集合, 这以更大的复杂性为代价实现了更好的编码效率。运动矢量预测是通过导出“运动矢量预测器候选”的列表来实现的, 其中所选择的候选的索引在位流中被用信号通知。针对各个编码单元 (CU) 生成合并候选列表。但是CU可以被拆分成更小的块, 以用于解码器侧运动矢量修正 (Decoder-side Motion Vector Refinement (DMVR)) 或其他方法。

[0005] 该列表的组成和顺序可能对编码效率产生显著影响, 因为准确的运动矢量预测器减小了块预测器的残差或失真的大小, 并且在列表的顶端具有这样的候选减少了用信号通知所选择的候选所需的位的数量。本发明旨在改进这些方面中的至少一个。

[0006] 修改被并入VVC v1和ECM中意味着可以有多达10个运动矢量预测器候选; 这实现了候选的多样性, 但是如果选择了列表中较低的候选, 位速率可能增加。本发明广泛涉及对运动矢量预测器候选的列表中的一个或多于一个“成对”运动矢量预测器候选的导出和排序的改进。“成对”运动矢量预测器候选是从候选的列表中的两个或多于两个其他候选组合出或平均出的候选。

发明内容

[0007] 本发明涉及对预测器 (或预测器候选) 的列表的处理的改进, 这实现了以一些增加的复杂性为代价而提供编码效率的预测器的改进排序。例如, 本方法试图通过将候选成本彼此进行比较来防止具有类似成本的冗余候选在候选列表中相邻。在实施例, 在候选的重新排序的量与应用了根据本发明的处理的所得列表的成本多样性的潜在收益之间取得平衡。本发明可以特别地 (但不排他地) 应用于针对图像或视频编码中的图像部分的运动预测的预测器 (或预测器候选)。然而, 当一个数据单元或数据项可以从另一数据单元或数据项被预测并且为此目的生成候选预测器列表时, 本发明对任何编码具有广泛的应用性。例如, 设想, 除了静止和视频图像数据之外, 本发明还可以至少对点云数据、网格数据和音频数据具有应用性。

[0008] 根据本发明的一方面, 提供一种用于处理预测器的方法, 所述方法包括: 获得预测

器的列表,所述预测器的列表具有至少两个预测器;使用与所述列表中的第一预测器相关的第一成本和与所述列表中的第二预测器相关的第二成本、基于准则来确定是否修改所述预测器的列表;基于所述确定来修改(或不修改)所述预测器的列表,其中所述准则是基于阈值的。所述第一预测器和所述第二预测器可以是所述列表中的相邻预测器(换句话说,一个跟在另一个之后)。预测器可以是候选。例如,针对诸如合并模式等的帧间模式的运动信息预测器候选或者帧内模式中的块预测器候选。

[0009] 所述确定可以包括将所述阈值添加到所述第一成本或所述第二成本,以及所述准则与结果同所述第一成本和所述第二成本中的没有添加所述阈值的成本的比较有关。

[0010] 所述准则可以是将所述阈值添加到所述第一成本的结果是否小于所述第二成本。

[0011] 所述准则可以是所述第一成本与所述第二成本之间的差是否越过所述阈值。

[0012] 可选地,所述差被计算为绝对值,以及所述准则是所述差是否小于所述阈值。

[0013] 可选地,修改所述列表包括从所述预测器的列表中移除第一候选或第二候选。

[0014] 可选地,修改所述列表包括改变所述列表中的一个或多于一个预测器的位置。

[0015] 可选地,改变一个或多于一个候选的位置包括将所述第一预测器或所述第二预测器移动到所述列表中的不同位置。

[0016] 改变排序可以包括将所述第一预测器或所述第二预测器移动到被认为是所述列表的末尾的位置。

[0017] 可选地,所述第一预测器和所述第二预测器在被认为是接在所述列表的起始的位置处的第零预测器之后。在一些示例中,可以从修改所述预测器的列表的步骤中排除所述第零预测器(例如,不管列表中的其他预测器的重新排序如何,第零预测器可以被固定在列表的起始处的位置)。在一些示例中,所述方法可以还包括用于将与所述第零预测器相关的第零成本与第零阈值进行比较的步骤,其中,在所述第零成本超过所述第零阈值的情况下,在用于修改所述预测器的列表的步骤中包括所述第零预测器。用于将与所述第零预测器相关的第零成本与第零阈值进行比较的步骤可以在第一集合的模式中进行,而不在第二集合的模式中进行。换句话说,在用于修改预测器的列表的步骤中包括第零预测器可以是模式相关的。在进行用于将与第零预测器相关的第零成本与第零阈值进行比较的步骤的示例中,所述第零阈值可以低于针对所述第一预测器和所述第二预测器所使用的阈值。在其他示例中,阈值可以是相同的。

[0018] 在其他示例中,所述第一预测器可以在被认为是所述列表的起始的位置处。在这样的示例中,所述第一成本可以被设置为零。也就是说,不管与第一预测器相关联的真正计算成本如何,所述方法可以包括用于将该值重写为零成本的步骤。替代地,可以将第一成本设置为零,而根本不计算该成本。

[0019] 可选地,所述第一成本和所述第二成本包括基于要编码或解码的图像部分的相邻样本和与所述预测器相关联的图像部分的相邻样本的模板匹配成本。

[0020] 可选地,所述第一成本和所述第二成本包括与所述预测器相关联的两个块预测器的样本值之间的差。

[0021] 可选地,所述第一成本和所述第二成本包括与所述预测器相关联的样本与所述列表中的至少一个其他预测器之间的差。

[0022] 可选地,所述第一成本和所述第二成本是基于所述预测器的样本或相邻样本的子

集的。

[0023] 可选地,所述第一成本和所述第二成本是基于与根据不同分辨率的另一图像相对应的样本的。

[0024] 可选地,对用于计算所述第一成本和所述第二成本的样本的值进行预处理。

[0025] 可选地,所述第一成本和所述第二成本是基于编码速率的。

[0026] 可选地,所述第一成本和所述第二成本是基于失真成本的。所述失真成本可以是以下项中的任一项:绝对差和(SAD)、绝对变换差和(SATD)、误差平方和(SSE)、以及结构相似性指数度量(SSIM)。

[0027] 所述成本可以与正被处理的单元(图像部分/预测单元/要预测的数据单元)中的样本的数量成比例。所述成本可以与模板(例如,相邻样本的模板)中的样本的数量成比例。所述成本可以基于用于确定失真成本和块大小的样本的。

[0028] 可选地,所述方法还包括导出所述阈值。替代地或附加地,所述方法还包括接收指示所述阈值的信息。

[0029] 所述阈值可以是基于量化参数(QP)值的。根据权利要求22至24中任一项所述的方法,所述阈值是基于当前帧与一个或多个其他帧之间的时间距离或图片顺序计数(POC)距离的。替代地或另外地,所述阈值可以是基于帧的时间ID的。替代地或另外地,所述阈值可以是基于参考帧的时间方向的。替代地或另外地,所述阈值是基于是否针对块使用双向预测的。所述阈值可以取决于可用参考帧的一个或多个QP值。所述阈值可以取决于用作所述第一成本和所述第二成本其中至少之一的失真的类型。所述阈值可以是基于被用于对作为所述第一成本和所述第二成本其中至少之一的度量的失真进行计算的样本的类型的。所述阈值可以是基于是帧内预测还是帧间预测被用于对要使用所述预测器的列表而预测的图像部分进行编码或解码的。所述阈值可以是基于随机接入(RA)配置、低延迟配置和仅帧内配置中的一者或多于一者中的哪个被用于进行编码或解码的。

[0030] 所述阈值可以是基于用于编码的公式的,并且是由用于计算所述阈值的解码器获得的。

[0031] 附加地或替代地,所述阈值可以是基于移位后的量化参数(QP)值的。所述移位后的QP值可以被用于确定拉格朗日参数 λ 作为所述阈值。所述移位后的量化参数可以被用于从表中获得所述拉格朗日参数(例如,Lambda),其中,所述表中的条目将拉格朗日参数值与量化参数值(例如,Lambda)相关联。

[0032] 附加地或替代地,所述阈值可以被确定为作为所述列表中的连续预测器之间的差的最小成本差,该连续预测器不是所述第一预测器和所述第二预测器。

[0033] 附加地或替代地,所述阈值可以是使用所述列表中的所有候选或前N个候选的成本值来确定的,其中,N是所述列表中允许的候选的最大数量。

[0034] 使用所述成本值可以包括从用于确定所述阈值的候选中计算连续候选成本的差的平均。

[0035] 所述阈值可以是基于拉格朗日参数 λ 的,以及所述准则是第一失真成本与第二失真成本之间的绝对差是否小于所述拉格朗日参数。例如,所述阈值可以是所述拉格朗日参数乘以所述第一预测器与所述第二预测器之间的速率差。

[0036] 可选地,将所述第一失真成本和所述第二失真成本其中至少之一乘以所述阈值。

[0037] 可选地,在第一排序处理之后应用所述准则。第一排序处理可以是基于成本的排序的。成本可以是模板匹配失真成本。

[0038] 可选地,将另外的阈值添加到由乘法产生的所述第一失真成本与所述第二失真成本之间的差的结果。

[0039] 可选地,所述确定是在根据候选的失真成本对候选进行排序的处理期间根据所述准则来进行的。

[0040] 可选地,确定是否修改所述列表是基于最小失真差和阈值的。重新排序的最大数量可以被限制为预定义数量。例如,重新排序的最大数量可以是所述列表中的候选的最大允许数量。

[0041] 可选地,所述确定是基于最小失真差的,并且被限制为重新排序的最大数量。

[0042] 可选地,修改所述列表仅包括移除候选。可选地,移除之后的候选的数量必须大于或等于所述列表中的候选的最大允许数量。在实施例中,在通过移动候选来修改所述列表时,使所述列表中允许的候选的最大数量递减。

[0043] 可选地,在满足所述准则的情况下,将所述第二预测器移动到第一位置,在所述第一位置处,针对所述列表中的后续预测器的成本中的差高于所述阈值。在具有重新排序的可能性的最大数量大于最终列表中允许的候选的最大数量时,所述修改可以包括将所述第一预测器或所述第二预测器被移动的位置限制到当前列表的末尾。可选地,所述预测器的处理被应用于常规合并模式、模板匹配合并模式、块匹配(BM)合并模式、以及帧内块拷贝(IBC)模式其中至少之一。

[0044] 在实施例中,所述第二预测器(在所述列表中)紧接在所述第一预测器之后,以及在确定为不满足所述准则的情况下,在所述列表中的后续预测器之后插入所述第二预测器。可选地,在确定为不满足所述准则的情况下,将所述准则应用于所述第二成本和所述列表中的先前候选的个体成本,以及其中,在确定为针对所述第二成本和先前候选的各成本不满足所述准则的情况下,不进行在所述后续预测器之后的所述第二预测器的插入。重新排序的最大可能数量可以被设置为已处理候选的最终列表中的候选的允许数量。附加地,所述预测器的处理可以被应用于与MVD的合并、与MVD的仿射合并、运动矢量残差的MVD符号预测其中至少之一。

[0045] 预测器的列表可以是预测器的子列表(与另一子列表可组合以形成最终列表、中间列表或其他子列表)、预测器的中间列表或最终列表。

[0046] 在实施例中,当使用模板匹配并且至少两个模板之一可用时,可以基于使用可用模板的加权成本作为针对不可用模板的成本来计算针对列表中的所有预测器(候选)的成本。例如,加权可以使得成本与缺失模板的高度和宽度成比例。模板可以是左或上模板。

[0047] 在实施例中,当使用模板匹配并且至少两个模板之一可用时,拉格朗日量或Lambda与缺失模板成比例地被修改。

[0048] 在实施例中,用于导出的运动矢量(MV)阈值对于所有合并预测器(候选)类型是相同的。这样的实施例可以有利地应用于基于预测器(候选)的类型而应用不同运动矢量阈值的合并模式,例如常规、模板匹配(TM)、或块匹配(BM)合并模式。

[0049] 在一个实施例中,当如上面提到的方面和实施例中的任一个中所阐述的本发明的方法可用(或启用)时,针对利用ARMC所导出和排序的所有预测器(候选),应用ARMC处理。

[0050] 在实施例中,当所提出的方法可用(或启用)时,在排序处理期间或之后添加成对预测器(候选)。

[0051] 在本发明的另一方面,提供一种用于处理预测器的列表的方法,所述方法包括:确定列表中的两个连续预测器(或预测器候选)之间的最小成本差。如果该最小差小于阈值(Λ),则将候选之一重新定位在列表中的最后位置处。可选地,最后位置在列表中的被认为冗余的候选之前。成本可以是失真。成本差可以是绝对成本差。

[0052] 根据本发明的另一方面,提供一种用于处理预测器的列表的方法,所述方法包括:确定两个连续预测器(或预测器候选)之间的最小成本差,如果最小差小于阈值(Λ),则将(冗余)预测器之一的失真与列表中(另外的)随后预测器的失真进行比较,识别具有高于 Λ 的失真差的第一随后预测器的位置,将预测器定位在该所识别的位置处。成本可以是失真。成本差可以是绝对成本差。

[0053] 根据本发明的另一方面,提供一种用于处理预测器的列表的方法,所述方法包括:针对候选,确定列表中的第一、随后的非冗余预测器(或预测器候选),在该候选之后插入该非冗余预测器。冗余候选可以是例如已经被确定为具有小于阈值的成本差的候选。成本可以是失真。成本差可以是绝对成本差。

[0054] 根据本发明的另一方面,提供一种用于处理预测器的列表的方法,所述方法包括:针对预测器(或预测器候选),确定列表中的第一、随后的非冗余预测器,除非该非冗余候选被确定为与该候选之前的所有候选相比是冗余的,否则在该候选之后插入该非冗余候选。冗余候选可以是例如已经被确定为具有小于阈值的成本差的候选。成本可以是失真。成本差可以是绝对成本差。

[0055] 根据本发明的另一方面,提供一种用于处理预测器的方法,所述方法包括:获得预测器列表,确定第一预测器的第一失真成本与第二预测器的第二失真成本之间的绝对失真成本差是否小于 Λ 值(例如,拉格朗日量),其中第二预测器紧跟在列表中的第一预测器之后;以及如果绝对失真成本差小于 Λ 值,则移动列表中的第一预测器的位置。第二预测器可以被移动到紧跟在列表中的当前被认为非冗余的最后候选之后并且在任何冗余候选之前的位置。例如,冗余候选可以是先前已经被确定为具有小于 Λ 的与先前候选的绝对失真差的候选。可以基于QP值或移位后的QP值来导出 Λ 值。例如,可以从具有被索引或与移位后的QP值相关联的条目的表中获得 Λ 值。

[0056] 根据本发明的另一方面,提供一种用于处理预测器的列表的方法,所述方法包括:获得预测器的列表,所述预测器的列表具有至少两个预测器,基于与所述列表中的第一预测器相关的第一成本与阈值的比较来确定是否修改所述预测器的列表,以及基于所述确定来修改所述预测器的列表。在一些示例中,用于确定是否修改所述预测器的列表的步骤是基于与所述列表中的各个预测器相关的相应成本与阈值的比较的。也就是说,可以将各自与列表中的相应预测器相关的各成本与阈值进行比较。在一些示例中,用于修改所述列表的步骤包括从所述预测器的列表中移除成本低于所述阈值的所述候选或各个候选。附加地或替代地,用于修改所述列表的步骤可以包括将所述列表中的成本低于所述阈值的所述候选或各个候选移动到所述列表中的接在成本大于所述阈值的候选之后的位置。在另外的示例中,用于修改所述列表的步骤可以包括将所述列表中的成本低于所述阈值的所述候选或各个候选移动到被认为是所述列表的末尾的位置。在所有示例中,用于修改所述预测器的

列表的步骤可以在第一集合的模式中进行,而不在第二集合的模式中进行。也就是说,列表中的预测器的重新排序可以是模式相关的。

[0057] 这些另外的方面可以与上面已经提到的先前陈述或实施例中的任一个有用地组合。

[0058] 以下可选特征可以与上面提到的方面和可选特征中的任一个组合应用。可选地,预测器的列表用于预测图像部分中的运动。列表中的预测器可以是帧内块预测器候选和帧间运动信息预测器候选中的任一个。替代地,预测器列表中的预测器可以用于以下各项中的任一项:预测点云中的位置、对网格进行编码、预测音频数据、预测静止图像数据和预测视频数据。该方法还可以包括将至少一个句法元素与预测器列表中的一或多个预测器相关联。

[0059] 在根据本发明的一方面,提供一种用于将图像数据编码在位流中的方法,该方法包括根据上述任何方面或实施例来处理预测器列表。

[0060] 在根据本发明的一方面,提供一种用于从位流中解码图像数据的方法,该方法包括根据上述任何方面或实施例来处理预测器列表。

[0061] 在根据本发明的一方面,提供一种用于将图像数据编码在位流中的设备,所述设备被配置为根据上述任何方面或实施例来处理预测器列表。

[0062] 在根据本发明的一方面,提供一种用于从位流中解码图像数据的设备,所述设备被配置为根据上述任何方面或实施例来处理预测器列表。

[0063] 在根据本发明的另一方面,提供一种(计算机)程序,该程序在执行时使可编程设备进行根据上述任何方面或实施例的方法。程序可以被存储在计算机可读存储介质上。

[0064] 程序可以独立地提供,或者可以在载体介质上、由载体介质承载或在载体介质中承载。载体介质可以是非暂态的,例如存储介质,特别是计算机可读存储介质。载体介质也可以是暂态的,例如信号或其他传输介质。信号可以经由任何合适的网络(包括因特网)传输。本发明的其他特征由独立权利要求和从属权利要求表征。

[0065] 本发明一个方面中的任何特征可以以任何适当的组合应用于本发明的其他方面。特别地,方法方面可以应用于设备方面,反之亦然。

[0066] 此外,硬件中实现的特征可以在软件中实现,反之亦然。本文中对软件和硬件特征的任何引用都应被相应地解释

[0067] 如本文所述的任何设备特征也可以被提供为方法特征,反之亦然。如本文所使用的,部件加功能特征可以替代地在其相应结构(诸如适当编程的处理器和相关联的存储器)方面来表达。

[0068] 还应当理解,可以独立地实现和/或提供和/或使用在本发明的任何方面中描述和定义的各种特征的特定组合。

附图说明

[0069] 现在将通过示例的方式参考附图,在附图中:

[0070] 图1是用于说明HEVC中使用的编码结构的图;

[0071] 图2是示意性地示出可以实现本发明的一个或多于一个实施例的数据通信系统的框图;

- [0072] 图3是示出可以实现本发明的一个或多个实施例的处理装置的组件的框图；
- [0073] 图4是示出根据本发明实施例的编码方法的步骤的流程图；
- [0074] 图5是示出根据本发明实施例的解码方法的步骤的流程图；
- [0075] 图6和7示出用于描述相对于当前块位于的块的标记方案；
- [0076] 图8的(a)和(b)例示仿射(子块)模式；
- [0077] 图9的(a)、(b)、(c)、(d)例示几何模式；
- [0078] 图10例示VVC的合并候选列表导出的第一步骤；
- [0079] 图11例示VVC的合并候选列表导出的另外步骤；
- [0080] 图12例示成对候选的导出；
- [0081] 图13例示基于相邻样本的模板匹配方法；
- [0082] 图14例示对图10所示的合并候选列表导出的第一步骤的修改；
- [0083] 图15例示对图11所示的合并候选列表导出的另外步骤的修改；
- [0084] 图16例示对图12所示的成对候选的导出的修改；
- [0085] 图17例示列表候选的成本确定；
- [0086] 图18例示合并模式候选的列表的重新排序处理；
- [0087] 图19例示在合并模式候选的列表的重新排序处理期间的成对候选导出；
- [0088] 图20例示基于失真值的预测器的排序处理；
- [0089] 图21例示本发明实施例的基于失真值的预测器的排序处理；
- [0090] 图22例示本发明实施例的基于失真值的预测器的一个重新排序处理；
- [0091] 图23例示本发明实施例的基于失真值的预测器的一个重新排序处理；图24例示本发明实施例的基于失真值的预测器的一个重新排序处理；图25例示本发明实施例的基于失真值的预测器的一个重新排序处理；
- [0092] 图26例示本发明实施例的基于失真值的预测器的一个重新排序处理；
- [0093] 图27例示本发明实施例的基于失真值的预测器的一个重新排序处理；
- [0094] 图28例示本发明实施例的基于失真值的预测器的一个重新排序处理；
- [0095] 图29例示根据本发明实施例的基于失真值的预测器的重新排序处理的图示；
- [0096] 图30例示子块合并列表的ARMC重新排序处理；
- [0097] 图31例示本发明实施例的基于失真值的预测器的一个重新排序处理；
- [0098] 图32例示本发明实施例的基于失真值的预测器的一个重新排序处理；
- [0099] 图33是示出根据本发明实施例的包括编码器或解码器和通信网络的系统的图；
- [0100] 图34是用于实现本发明的一个或多个实施例的计算装置的示意性框图；
- [0101] 图35是例示网络照相机系统的图；
- [0102] 图36是例示智能电话的图。

具体实施方式

[0103] 图1涉及在高效视频编码(HEVC)视频标准和多样化视频编码(VVC)标准中使用的编码结构。视频序列1由一系列数字图像i组成。各个这样的数字图像由一个或多个矩阵表示。矩阵系数表示像素。

[0104] 序列的图像2可以被分割成条带3。在一些情况下,一条带可以构成图像整体。这些

条带被分割成非重叠编码树单元 (CTU)。编码树单元 (CTU) 是高效率视频编码 (HEVC) 视频标准的基本处理单元, 并且概念性地在结构上与若干先前视频标准中使用的宏块单元相对应。CTU有时也被称为最大编码单元 (LCU)。CTU具有亮度和色度分量部分, 各个分量部分被称为编码树块 (CTB)。这些不同的颜色分量未在图1中示出。

[0105] 针对HEVC, CTU通常大小为64像素×64像素, 而针对VVC, 该大小可以为128像素×128像素。可以使用二叉树分解进而将各CTU迭代地分割成较小的可变大小编码单元 (CU) 5。

[0106] 编码单元是基本编码元素, 并且由被称为预测单元 (PU) 和变换单元 (TU) 的两种子单元构成。PU或TU的最大大小等于CU大小。预测单元与CU的用于像素值的预测的分区相对应。将CU分区成PU的各种不同分区是可能的, 如606所示, 包括分成4个正方形PU的分区、以及分成2个矩形PU的两个不同分区。变换单元是使用DCT进行空间变换的基本单元。CU可以基于二叉树表示607分区成TU。

[0107] 各条带嵌入一个网络抽象层 (NAL) 单元中。另外, 视频序列的编码参数存储在称为参数集的专用NAL单元中。在HEVC和H.264/AVC中, 采用两种参数集NAL单元: 第一, 序列参数集 (SPS) NAL单元, 其收集在整个视频序列期间不变的所有参数。通常, 它处理编码配置文件、视频帧的大小和其他参数。第二, 图片参数集 (PPS) NAL单元, 其包括可以从序列的一个图像 (或帧) 改变为其他图像 (或帧) 的参数。HEVC还包括视频参数集 (VPS) NAL单元, 其包含描述位流的总体结构的参数。VPS是HEVC中定义的一种类型的参数集, 并且应用于位流的所有层。层可以包含多个时间子层, 并且所有版本1的位流限定于单个层。HEVC具有用于可缩放性和多视图的某些分层扩展, 并且这些扩展将允许具有向后兼容的版本1的基础层的多个层。

[0108] 在包括子图片的VVC中已经引入了对图像进行拆分其他方式, 所述子图片是一个或多个条带的独立编码组。

[0109] 图2例示可以实现本发明的一个或多个实施例的数据通信系统。数据通信系统包括传输装置 (在这种情况下是服务器201), 其可操作以经由数据通信网络200将数据流的数据包传输至接收装置 (在这种情况下是客户端终端202)。数据通信网络200可以是广域网 (WAN) 或局域网 (LAN)。这种网络可以是例如无线网络 (Wifi/802.11a或b或g)、以太网网络、互联网网络或由若干不同网络组成的混合网络。在本发明的特定实施例中, 数据通信系统可以是数字电视广播系统, 其中服务器201将相同的数据内容发送到多个客户端。

[0110] 由服务器201提供的数据流204可以由表示视频和音频数据的多媒体数据组成。在本发明的一些实施例中, 音频和视频数据流可以分别由服务器201使用麦克风和照相机来捕获。在一些实施例中, 数据流可以存储在服务器201上或由服务器201从其他数据提供商接收, 或在服务器201处生成。服务器201设置有用以对视频和音频流进行编码的编码器, 特别是用以提供用于传输的压缩位流, 该压缩位流是作为编码器的输入所呈现的数据的更紧凑表示。

[0111] 为了获得更好的传输数据的质量与传输数据的量的比率, 可以例如根据HEVC格式或H.264/AVC格式或VVC格式来压缩视频数据。

[0112] 客户端202接收所传输的位流, 并且解码重建的位流, 以在显示装置上再现视频图像和利用扬声器再现音频数据。

[0113] 尽管在图2的示例中考虑了流式传输场景, 但将认识到, 在本发明的一些实施例

中,可以使用例如介质存储装置(诸如光盘等)来进行编码器与解码器之间的数据通信。

[0114] 在本发明的一个或多个实施例中,视频图像与表示要应用到图像的重建像素的补偿偏移的数据一同传输,以在最终图像中提供经滤波的像素。

[0115] 图3示意性地例示被配置为实现本发明的至少一个实施例的处理装置300。处理装置300可以是诸如微计算机、工作站或轻型便携式装置等的装置。装置300包括通信总线313,其连接到:

[0116] -表示为CPU的中央处理单元311,诸如微处理器等;

[0117] -表示为ROM的只读存储器306,其用于存储实现本发明的计算机程序;

[0118] -用于存储本发明实施例的方法的可执行代码的表示为RAM的随机访问存储器312,以及适于记录变量和参数的寄存器,该变量和参数是根据本发明实施例实现对数字图像序列进行编码的方法和/或对位流进行解码的方法所需的;以及

[0119] -连接至通信网络303的通信接口302,通过该通信接口传输或接收要处理的数字数据。

[0120] 可选地,设备300还可以包括以下组件:

[0121] -诸如硬盘等的数据存储部件304,其用于存储实现本发明的一个或多个实施例的方法的计算机程序以及在实现本发明的一个或多个实施例期间所使用或产生的数据;

[0122] -用于盘306的盘驱动器305,该盘驱动器适于从盘306读取数据或将数据写入所述盘;

[0123] -屏幕309,其用于借助于键盘310或任何其他指示装置来显示数据和/或用作与用户交互的图形界面。

[0124] 设备300可以连接到诸如数字照相机320或麦克风308等的各种外围设备,其各自连接到输入/输出卡(未示出)以向设备300提供多媒体数据。

[0125] 通信总线提供设备300中所包括的或连接到设备300的各种元素之间的通信和互操作性。总线的表示不是限制性的,并且特别地,中央处理单元可操作地将指令直接或者借助于设备300的其他元素通信到设备300的任何元素。

[0126] 盘306可以由诸如可重写或不可重写的致密盘(CD-ROM)、ZIP盘或存储卡等的任何信息介质代替,并且一般而言,由微计算机或微处理器可以进行读取的信息存储部件代替,该盘306集成到或不集成到设备中、可能可移动并且适于存储其执行使得能够实现根据本发明的对数字图像序列进行编码的方法和/或对位流进行解码的方法的一个或多个程序。

[0127] 可执行代码可以存储在只读存储器306中、硬盘304上或可移动数字介质(诸如,例如如前述的盘306等)上。根据一变型,程序的可执行代码可以经由接口302借助于通信网络303来接收,以在执行之前存储在设备300的存储部件之一(诸如硬盘304等)中。

[0128] 中央处理单元311适于控制和指导执行根据本发明的一个或多个程序的指令或软件代码的部分、存储在上述存储部件之一中的指令的执行。在通电时,存储在非易失性存储器中(例如,在硬盘304上或在只读存储器306中)的一个或多个程序被传递到随机访问存储器312中(其然后包含一个或多个程序的可执行代码)以及用于存储实现本发明所必需的变量和参数的寄存器。

[0129] 在该实施例中,设备是使用软件来实现本发明的可编程设备。然而,替代地,本发明可以以硬件(例如,以专用集成电路或ASIC的形式)来实现。

[0130] 图4例示根据本发明的至少一个实施例的编码器的框图。编码器由所连接的模块表示,各模块适于例如以由装置300的CPU 311执行的编程指令的形式来实现根据本发明的一个或多个实施例的、用于实现对图像序列中的图像进行编码的至少一个实施例的方法的至少一个相应步骤。

[0131] 编码器400接收数字图像 i_0 至 i_n 的原始序列401作为输入。各数字图像由样本(有时也称为像素)(在下文中,它们被称为像素)的集合表示。

[0132] 编码器400在实现编码处理之后输出位流410。位流410包括多个编码单元或条带,各条带包括用于对条带编码所用的编码参数的编码值进行传输的条带头部、以及包括编码视频数据的条带主体。

[0133] 模块402将输入数字图像 i_0 至 i_n 401分割成像素块。块与图像部分相对应并且可以具有可变大小(例如, 4×4 、 8×8 、 16×16 、 32×32 、 64×64 、 128×128 像素、并且还可以考虑若干矩形块大小)。针对各输入块选择编码模式。提供了两个编码模式族:基于空间预测编码(帧内预测)的编码模式和基于时间预测的编码模式(帧间编码、合并、跳过)。测试了可能的编码模式。

[0134] 模块403实现帧内预测处理,其中,通过根据要编码的给定块的相邻像素计算出的预测器来预测所述要编码的块。如果选择了帧内编码,则对所选择的帧内预测器以及给定块与其预测器之间的差的指示进行编码以提供残差。

[0135] 时间预测由运动估计模块404和运动补偿模块405实现。首先,选择来自参考图像集416的参考图像,并且由运动估计模块404选择参考图像的一部分(也被称为参考区域或图像部分),该部分是与要编码的给定块(在像素值类似性方面)最接近的区域。然后运动补偿模块405使用所选择的区域来预测要编码的块。由运动补偿模块405计算所选择的参考区域与给定块(也称为残差块)之间的差。使用运动矢量指示所选择的参考区域。

[0136] 由此,在这两个情况下(空间和时间预测),通过从原始块减去预测器来计算残差。

[0137] 在由模块403实现的帧内预测中,对预测方向进行编码。在由模块404、405、416、418、417实现的帧间预测中,至少一个运动矢量或用于识别这种运动矢量的数据被编码用于时间预测。

[0138] 如果选择帧间预测,则对与运动矢量和残差块有关的信息进行编码。为了进一步降低位速率,假定运动是同质的,通过相对于运动矢量预测器的差对运动矢量进行编码。由运动矢量预测和编码模块417从运动矢量场418获得运动信息预测器候选的集合中的运动矢量预测器。

[0139] 编码器400还包括选择模块406,该选择模块用于通过应用编码成本准则(诸如,速率-失真准则等)来选择编码模式。为了进一步减少冗余,由变换模块407将变换(诸如DCT等)应用于残差块,然后,所获得的变换数据由量化模块408量化并且由熵编码模块409进行熵编码。最终,正被编码的当前块的编码后的残差块被插入位流410中。

[0140] 编码器400还进行编码图像的解码,以产生用于后续图像的运动估计的参考图像(例如,参考图像/图片416中的参考图像)。这使得接收位流的编码器和解码器能够具有相同的参考帧(使用重建图像或图像部分)。逆量化(去量化)模块411进行量化数据的逆量化

(去量化),之后是逆变换模块412的逆变换。帧内预测模块413使用预测信息来确定对于给定块使用哪个预测器,并且运动补偿模块414实际上将由模块412获得的残差添加到从参考图像集416获得的参考区域。

[0141] 然后,由模块415应用后滤波以对所重建的像素帧(图像或图像部分)进行滤波。在本发明的实施例中,使用SAO环路滤波器,其中补偿偏移被添加到所重建图像的所重建像素的像素值。应理解,并不总是必须进行后滤波。此外,替代SAO环路滤波或除了SAO环路滤波之外,还可以进行任何其他类型的后滤波。

[0142] 图5示出根据本发明实施例的解码器60的框图,解码器60可以用于从编码器接收数据。解码器由所连接的模块表示,各模块适于例如以要由装置300的CPU 311执行的编程指令的形式实现由解码器60实现的方法的相应步骤。

[0143] 解码器60接收包括编码单元(例如,与块或解码单元相对应的数据)的位流61,各编码单元由包含与经编码的参数有关的信息的头部和包含经编码的视频数据的主体组成。如关于图4所说明的,针对给定块,在预定数量的位上,对经编码的视频数据进行熵编码,并且对运动矢量预测器的索引进行编码。所接收的经编码的视频数据由模块62进行熵解码。然后残差数据由模块63去量化,之后由模块64应用逆变换以获得像素值。

[0144] 用于指示编码模式的模式数据也被熵解码,并且基于该模式,对图像数据的编码块(单元/集合/组)进行帧内类型解码或帧间类型解码。

[0145] 在帧内模式的情况下,帧内预测模块65基于在位流中指定的帧内预测模式来确定帧内预测器。

[0146] 如果模式是帧间,则从位流提取运动预测信息以找到(识别)由编码器使用的参考区域。运动预测信息包括参考帧索引和运动矢量残差。通过运动矢量解码模块70将运动矢量预测器添加到运动矢量残差以获得运动矢量。下文参考图6至图10更详细地讨论VVC中使用的各种运动预测器工具。

[0147] 运动矢量解码模块70对通过运动预测编码的各当前块应用运动矢量解码。一旦已获得针对当前块的运动矢量预测器的索引,可以对与当前块相关联的运动矢量的实际值进行解码,并且该实际值用以通过模块66应用运动补偿。从参考图像68提取由经解码的运动矢量指示的参考图像部分以应用运动补偿66。利用经解码的运动矢量更新运动矢量场数据71,以用于后续解码运动矢量的预测。

[0148] 最终,获得经解码的块。在适当的情况下,由后滤波模块67应用后滤波。解码器60最终获得并提供经解码的视频信号69。

[0149] 运动预测(帧间)模式

[0150] HEVC使用3个不同的帧间模式:帧间模式(高级运动矢量预测(AMVP))、“经典”合并模式(即“非仿射合并模式”或也称为“常规”合并模式)和“经典”合并跳过模式(即“非仿射合并跳过”模式或也称为“常规”合并跳过模式)。这些模式之间的主要差异在于位流中的数据的信号通知。对于运动矢量编码,当前的HEVC标准包括用于运动矢量预测的基于竞争的方案,该方案在该标准的早期版本中不存在。这意味着数个候选在编码器侧与速率失真准则竞争,以分别针对帧间或合并模式(即“经典/常规”合并模式或“经典/常规”合并跳过模式)找到最佳运动矢量预测器或最佳运动信息。然后将与运动信息的最佳预测器或最佳候选相对应的索引与表示预测值和实际值之间的差的“残差”一起插入位流中。解码器可以导

出预测器或候选的相同集合,并且根据经解码的索引使用最佳的一个。使用残差,解码器然后可以重新创建原始值。

[0151] 在HEVC的画面内容扩展中,被称为帧内块拷贝(IBC)的新编码工具被用信号通知为这三个帧间模式中的任何一个,IBC和等效帧间模式之间的差异是通过检查参考帧是否是当前帧来确定的。这可以例如通过以下方式来实现:检查列表L0的参考索引,并且如果是该列表中的最后帧,则推断这是帧内块拷贝。另一方式是将当前帧和参考帧的图片顺序计数进行比较:如果相等,则这是帧内块拷贝。

[0152] 预测器和候选的导出的设计对于在不会对复杂性产生不成比例的影响的情况下实现最佳编码效率是重要的。在HEVC中,使用两个运动矢量导出:一个用于帧间模式(高级运动矢量预测(AMVP)),一个用于合并模式(合并导出处理(用于经典合并模式和经典合并跳过模式))。下面描述了VVC中使用的各种运动预测器模式。

[0153] 图6示出本文使用的描述帧之间相对于当前块(即,当前正在编码/解码的块)位于的块的标记方案(图6)。

[0154] VVC合并模式

[0155] 与HEVC相比,在VVC中添加了数个帧间模式。特别地,新的合并模式已被添加到HEVC的常规合并模式。

[0156] 仿射模式(子块模式)

[0157] 在HEVC中,仅针对运动补偿预测(MCP)应用平移运动模型。而在现实世界中,有许多种运动,例如放大/缩小、旋转、透视运动和其他不规则运动。

[0158] 在JEM中,应用了简化的仿射变换运动补偿预测,并且基于2017年7月13-21日在都灵(Torino)举行的JVET会议上提交的文件JVET-G1001的摘录,在下面描述了仿射模式的一般原理。在描述JEM中使用的其他算法的范围内,该整个文档通过引用并入本文。

[0159] 如图8的(a)所示,块的仿射运动场由两个控制点运动矢量描述。

[0160] 仿射模式是如帧间模式(AMVP、“经典”合并或“经典”合并跳过)的运动补偿模式。其原理是根据2或3个相邻的运动信息,生成每像素一个运动信息。在JEM中,仿射模式针对各个 4×4 块导出一个运动信息,如图8的(a)所示(各个正方形是 4×4 块,并且图8的(a)中的整个块是 16×16 块,该整个块被分割成这种 4×4 大小的正方形的16个块(各个 4×4 正方形块具有与其相关联的运动矢量)。通过利用标志启用仿射模式,仿射模式可用于AMVP模式和合并模式(即,也称为“非仿射合并模式”的经典合并模式和也称为“非仿射合并跳过模式”的经典合并跳过模式)。

[0161] 在VVC规范中,仿射模式也称为子块模式;这些术语在本说明书中可互换使用。

[0162] VVC的子块合并模式包含基于子块的时间合并候选,该候选继承由空间运动矢量候选指向的前一帧中的块的运动矢量场。如果已经利用子块合并的帧间仿射模式编码了相邻块,然后在某个零 M_v 候选之前导出了一些作为构建的仿射候选,则该子块候选之后是继承的仿射运动候选。

[0163] CIIP

[0164] 除了常规合并模式和子块合并模式之外,VVC标准还包含组合帧间合并/帧内预测(CIIP),也已知为多假设帧内帧间(Multi-Hypothesis Intra Inter(MHII))合并模式。

[0165] 组合帧间合并/帧内预测(CIIP)合并可以被认为是常规合并模式和帧内模式的组

合,并且下面参考图10进行描述。该模式的针对当前块(1001)的块预测器是合并预测器块和帧内预测器块之间的平均,如图10所示。合并预测器块是利用与合并模式完全相同的处理获得的,因此其是时间块(1002)或两个时间块的双预测器。因此,以与常规合并模式相同的方式针对该模式用信号通知合并索引。基于当前块(1001)的相邻样本(1003)获得帧内预测器块。然而,与帧内块相比,当前块的可用帧内模式的数量是有限的。此外,没有针对CIIP块用信号通知的色度帧内预测器块。色度预测器等于亮度预测器。结果,1、2或3位用于用信号通知CIIP块的帧内预测器。

[0166] 通过合并块预测器和帧内块预测器的加权平均获得CIIP块预测器。加权平均的加权取决于块大小和/或所选择的帧内预测器块。

[0167] 然后将所获得的CIIP预测器添加到当前块的残差,以获得重建块。应该注意的是,CIIP模式仅针对未跳过的块启用。事实上,使用CIIP跳过通常导致压缩性能的损失和编码器复杂性的增加。这是因为CIIP模式通常具有与其他跳过模式相反的块残差。因此,其针对跳过模式的信号通知增加了位速率。当跳过当前CU时,避免CIIP。这种限制的结果是CIIP块不能具有仅包含0值的残差,因为不可能对等于0的VVC块残差进行编码。事实上,在VVC中,对于合并模式,用信号通知块残差等于0的唯一方式是使用跳过模式,这是因为对于合并模式,CU CBF标志被推断为等于真。并且当该CBF标志为真时,块残差不能等于0。

[0168] 以这种方式,在本说明书中,CIIP应该被解释为结合了帧间和帧内预测特征的模式,而不一定是给予一个特定模式的标签。

[0169] CIIP使用与常规合并模式相同的运动矢量候选列表。

[0170] MMVD

[0171] MMVD合并模式是特定的常规合并模式候选导出。它可以被认为是独立的合并候选列表。对于当前CU,通过将运动矢量分量(mvx或mvy)的偏移值添加到初始常规合并候选来获得所选择的MMVD合并候选。偏移值被添加到第一列表L0的运动矢量或第二列表L1的运动矢量,这取决于这些参考帧的配置(这两者都是向后的、这两者都是向前的、或者向前和向后的)。利用索引而用信号通知初始合并候选。利用8个可能的距离(1/4像素、1/2像素、1像素、2像素、4像素、8像素、16像素、32像素)之间的距离索引和给出x或y轴以及偏移符号的方向索引来用信号通知偏移值。

[0172] 在VVC中,仅常规合并列表的前两个候选被用于MMVD导出且通过一个标志用信号通知。

[0173] 几何分区模式

[0174] 几何(GE0)合并模式是特殊的双预测模式。图9例示该特殊的块预测器生成。块预测器包含来自第一块预测器的一个三角形(901或911)和来自第二块预测器的第二三角形(902或912)。但是如图9的(c)和(d)所描述的,块的数个其他可能拆分是可能的。在本说明书中,几何合并应被解释为组合两个帧间非正方形预测器的特征的模式,而不一定是给予一个特定模式的标签。

[0175] 在图9的(a)的示例中,各个分区(901或902)具有作为单向候选的运动矢量候选。并且对于各个分区,用信号通知索引以在解码器处获得单向候选列表中的相应运动矢量候选。并且第一分区和第二分区不能使用相同的候选。该候选列表来自常规合并候选列表,其中对于各个候选,已经移除了2个分量之一(L0或L1)。

[0176] IBC

[0177] 在VVC中,也可以启用帧内块拷贝(IBC)合并模式。IBC具有独立的合并候选导出处理。

[0178] 其他运动信息改进

[0179] DMVR

[0180] VVC中的解码器侧运动矢量导出(DMVR)增加了合并模式的MV的准确性。对于该方法,应用了基于双边匹配(BM)的解码器侧运动矢量修正。在该双预测操作中,围绕参考图片列表L0和参考图片列表L1中的初始MV搜索经修正的MV。BM方法计算参考图片列表L0和列表L1中的两个候选块之间的失真。

[0181] BDOF

[0182] VVC还集成了双向光流(BDOF)工具。BDOF(先前称为BIO)用于在 4×4 子块级别修正CU的双预测信号。如果CU满足数个条件,尤其是如果从两个参考图片到当前图片的距离(即图片顺序计数(POC)差)相同,则将BDOF应用于CU。顾名思义,BDOF模式基于假定物体的运动是平滑的光流概念。对于各个 4×4 子块,通过使L0和L1预测样本之间的差最小化来计算运动修正(v_x, v_y)。然后使用运动修正来调整 4×4 子块中的双预测样本值。

[0183] PROF

[0184] 类似地,针对仿射模式使用具有光流的预测修正(Prediction refinement with optical flow (PROF))。

[0185] AMVR和hpelIfIdx

[0186] VVC还包括自适应运动矢量分辨率(AMVR)。AMVR允许以不同的精度来编码CU的运动矢量差。例如,对于AMVP模式,考虑四分之一亮度样本、半亮度样本、整亮度样本或四亮度样本。VVC规范的下表给出了基于不同句法元素的AMVR移位。

amvr_flag	amvr_precision_idx	AmvrShift		
		inter_affine_flag[x0][y0] == 1	CuPredMode[chType][x0][y0] == MODE_IBC)	inter_affine_flag[x0][y0] == 0 && CuPredMode[chType][x0][y0] != MODE_IBC
[0187] 0	-	2 (1/4亮度样本)	-	2 (1/4亮度样本)
1	0	0 (1/16亮度样本)	4 (1亮度样本)	3 (1/2亮度样本)
1	1	4 (1亮度样本)	6 (4亮度样本)	4 (1亮度样本)
1	2	-	-	6 (4亮度样本)

[0188] AMVR可能对除了使用运动矢量差编码作为不同合并模式的模式之外的其他模式的编码产生影响。事实上,对于一些候选,参数hpelIfIdx(其表示半像素精度的亮度插值滤波器的索引)被传播用于一些合并候选。例如,对于AMVP模式,如下导出hpelIfIdx:

[0189] hpelIfIdx = AmvrShift == 3 ? 1 : 0

[0190] 具有CU级权重的双预测(BCW)

[0191] 在VVC,具有CU级权重的双预测模式(BCW)被扩展到超出简单平均(如在HEVC中进

行的),以允许根据下式的两个预测信号 P_0 和 P_1 的加权平均。

[0192] $P_{bi-pred} = ((8-w)*P_0+w*P_1+4) >> 3$

[0193] 在加权平均双预测中允许五个权重,其中 $w \in \{-2, 3, 4, 5, 10\}$ 。

[0194] 对于非合并CU,在运动矢量差之后用信号通知权重索引bcwIndex。

[0195] 对于合并CU,基于合并候选索引从相邻块推断权重索引。

[0196] BCW仅用于具有256个或更多亮度样本的CU。此外,对于低延迟图片,使用所有5个权重。对于非低延迟图片,仅使用3个权重($w \in \{3, 4, 5\}$)。

[0197] 常规合并列表导出

[0198] 在VVC中,如图10和图11导出常规合并列表。首先,添加空间候选B1 (1002)、A1 (1006)、B0 (1010)、A0 (1014) (如图7所描绘的) (如果它们存在)。并且在A1和B1的运动信息之间进行部分冗余 (1007) 以添加A1 (1008),在B0和B1的运动信息之间进行部分冗余 (1011) 以添加B0 (1012),以及在A0和A1的运动信息之间进行部分冗余 (1015) 以添加A0 (1016)。

[0199] 当添加合并候选时,变量cnt递增 (1015、1009、1013、1017、1023、1027、1115、1108)。

[0200] 如果列表中候选的数量(cnt)严格小于4 (1018),则在候选B2 (1019)不具有与A1和B1相同的运动信息 (1021)的情况下,添加 (1022) 候选B2 (1019)。

[0201] 然后添加时间候选。如果右下候选 (1024) 可用 (1025),则添加该候选 (1026),否则如果中心时间候选 (1028) 存在 (1029),则添加中心时间候选 (1026)。

[0202] 然后,添加基于历史的(HMVP) (1101) (如果它们不具有与A1和B1相同的运动信息 (1103))。此外,基于历史的候选的数量不能超过合并候选列表的候选的最大数量减1 (1102)。因此,在合并候选列表中,在基于历史的候选之后,缺失至少一个位置。

[0203] 然后,如果列表中的候选的数量至少为2,则构建成对候选 (1106) 并将其添加到合并候选列表中 (1107)。

[0204] 然后,如果在合并候选列表中存在空位置 (1109),则添加零候选 (1110)。

[0205] 对于空间候选和基于历史的候选,参数BCWidx和useAltHpelIf被设置为等于候选的相关参数。对于时间候选和零候选,它们被设置为等于默认值0。这些默认值实质上禁用了该方法。

[0206] 对于成对候选,BCWidx被设置为等于0,并且hpelIfIdxp被设置为等于第一候选的hpelIfIdxp (如果第一候选的hpelIfIdxp等于第二候选的hpelIfIdxp,否则被设置为0)。

[0207] 成对候选导出

[0208] 根据图12的算法构建成对候选 (1106)。如所描绘的,当2个候选在列表中时 (1201),如前所述导出hpelIfIdxp (1204、1202、1203)。然后将帧间方向(interDir)设置为等于0 (1205)。对于各个列表L0和L1,如果至少一个参考帧有效 (不同于-1) (1207),则将设置参数。如果这两者都有效 (1208),则导出该候选的mv信息 (1209) 并将其设置为等于第一候选的参考帧,并且运动信息是该列表的两个运动矢量之间的平均,且变量interDir递增。如果候选中仅一个具有该列表的运动信息 (1210),则成对候选的运动信息被设置为等于该候选 (1212、1211),并且帧间方向变量interDir递增。

[0209] ECM

[0210] 自VVC v1标准结束以来,JVET通过建立探索软件(ECM)启动了探索阶段。探索阶段

收集了附加工具并且改进了VVC标准的基础上的现有工具,以实现更好的编码效率。JVET-X2025中描述了与VVC相比的不同附加工具。

[0211] ECM合并模式

[0212] 在添加的所有工具中,添加了一些附加的合并模式。仿射MMVD用信号通知作为常规合并模式的MVVD编码的合并仿射候选的偏移。类似地,还添加了GEO MMVD。CIIP PDPC是CIIP的扩展。并且添加了两个模板匹配合并模式:常规模板匹配和GEO模板匹配。

[0213] 常规模板匹配是基于模板匹配估计,如图13所描绘的。在解码器侧,对于与相关合并索引相对应的候选和对于可用的这两个列表(L0、L1),基于当前块(1301)的相邻样本和基于多个相应块位置的相邻样本进行运动估计,计算成本并选择使成本最小化的运动信息。运动估计受到搜索范围的限制,并且对该搜索范围的数个限制也被用于降低复杂性。

[0214] 在ECM中,常规模板匹配候选列表是基于常规合并列表,但是添加了一些附加的步骤和参数,这意味着可以生成针对同一块的不同合并候选列表。此外,与用于具有JVET定义的共同测试条件的ECM中的针对常规合并候选列表的10个候选相比,仅4个候选可用于模板匹配的常规合并候选列表。

[0215] ECM中的常规合并列表导出

[0216] 在ECM中,更新了常规合并列表导出。图14和图15示出分别基于图10和图11的该更新。但是为了清楚,基于历史的候选(1101)的模块已经在(1501)中被概述。

[0217] 在该图15中,添加了新类型的合并候选:不相邻候选(1540)。这些候选来自空间上位于当前帧的块而不是相邻帧中的块,因为相邻是空间候选。这些候选是根据距离和方向而选择的。对于基于历史的候选,可以添加相邻候选列表,直到该列表达到候选的最大数量减1,使得仍然可以添加成对候选。

[0218] 零候选

[0219] 如果列表仍然尚未达到候选的最大数量(Maxcand),则向列表添加零候选。根据可能的参考帧或参考帧对来添加零候选。以下伪代码给出了这种候选的导出:

```

int iNumRefIdx =slice.isInterB()?min(MaxRefL0, MaxRefL1): MaxRefL0;
int r=0
int refcnt=0;
while (nbCand < Maxcand)
{
    if (slice.isInterB())
    {
        addZero(L0(Mv(0,0),Refix(r)), L1(Mv(0,0),Refix(r)));
    }
    else
[0220] {
        addZero(Mv(0,0),Refix(r));
    }
    nbCand++;
    if (refcnt == iNumRefIdx - 1)
    {
        r = 0;
    }
    else
    {
        ++r;
        ++refcnt;
[0221] }
}
}

```

[0222] 该伪码可以概括为：对于各个参考帧索引(单向)或参考索引对(双预测)，添加零候选。当全部被添加时，仅添加具有参考帧索引0的零候选，直到候选的数量达到其最大值。以这种方式，合并列表可以包括多个零候选。事实上，已经令人惊讶地发现，这在真实视频序列中、特别是在条带或帧和序列的起始处频繁发生。

[0223] 在合并候选的导出的最近修改中，列表中的候选的数量可以大于最终列表中的候选的最大数量Maxcand。然而，初始列表中的候选的该数量(MaxCandInitialList)被用于导出。因此，添加零候选，直到候选的数量是MaxCandInitialList，并且不是直到其是Maxcand。

[0224] BM合并模式

[0225] BM合并是专用于自适应解码器侧运动矢量细化方法的合并模式，该方法是ECM的多遍DMVR的扩展。如JVET-X2025中所述，该模式相当于仅在一个方向上细化MV的2个合并模式。因此，一个合并模式针对L0，且一个合并模式针对L1。因此，仅在DMVR条件可以启用时，才启用BM合并。对于这两个合并模式，仅导出合并候选的一个列表，并且所有候选都遵守DMVR条件。

[0226] 以与常规合并模式类似的方式,从空间相邻编码块、TMVP、不相邻块、HMVP、成对候选导出针对BM合并模式的合并候选。不同之处在于,仅符合DMVR条件的那些才会被添加到候选中。以与针对常规合并模式类似的方式对合并索引进行编码。

[0227] AMVP合并模式

[0228] AVMP合并模式(也称为双预测器)在JVET-X2025中定义如下:其由一个方向上的AMVP预测器和另一方向上的合并预测器组成。当所选择的合并预测器和AMVP预测器满足DMVR条件时,可以对编码块启用该模式,其中相对于当前图片存在来自过去的至少一个参考图片和来自未来的一个参考图片,并且从这两个参考图片到当前图片的距离相同,针对作为起点的合并MV候选和AMVP MVP应用双边匹配MV细化。否则,如果启用模板匹配功能,则将模板匹配MV细化应用于具有较高模板匹配成本的合并预测器或AMVP预测器。

[0229] 用信号通知模式的AMVP部分作为常规单向AMVP,即,用信号通知参考索引和MVD,并且如果使用模板匹配,则AMVP部分具有导出的MVP索引,或者当模板匹配被禁用时用信号通知MVP索引。

[0230] 对于AMVP方向LX(其中X可以是0或1),通过使AMVP预测器和合并预测器(即,针对AMVP和合并运动矢量的对)之间的双边匹配成本最小化,隐含地导出另一方向(1-LX)上的合并部分。对于合并候选列表中具有该另一方向(1-LX)运动矢量的每个合并候选,使用合并候选MV和AMVP MVP来计算双边匹配成本。选择具有最小成本的合并候选。对以所选择的合并候选MV和AMVP MVP为起点的编码块应用双边匹配细化。

[0231] 使得多遍DMVR中的作为多遍DMVR的 8×8 子PU BDOF细化的第三程能够用于AMVP合并模式编码块。

[0232] 模式由标志指示,如果模式被启用,则AMVP方向LX进一步由标志指示。

[0233] MVD符号预测

[0234] 在JVET-X0132中描述符号预测方法。如果运动矢量差包含非零分量,则可以在常规帧间模式中应用运动矢量差符号预测。在当前ECM版本中,针对AMVP、仿射MVD和SMVD模式应用符号预测。根据模板匹配成本对可能的MVD符号组合进行排序,并且导出对应于真正MVD符号的索引,并且用上下文模型对该索引进行编码。在解码器侧,如下导出MVD符号:1/解析MVD分量的大小;2/解析上下文编码的MVD符号预测索引;3/通过创建可能符号与绝对MVD值之间的组合来构建MV候选,并且将其添加到MV预测器;4/基于模板匹配成本和排序针对各个所导出的MV来导出MVD符号预测成本;5/使用MVD符号预测索引来挑选真正MVD符号;6/将真正MVD添加到最终MV的MV预测器。

[0235] TIMD

[0236] 在JVET-X2025中将帧内预测(用于基于模板的帧内模式导出(TIMD)的融合)描述为如下:对于MPM中的各个帧内预测模式,计算模板的预测样本和重建样本之间的SATD。具有最小SATD的前两个帧内预测模式被选择为TIMD模式。这两个TIMD模式在应用PDPC处理之后与权重融合,并且将这样的加权帧内预测用于对当前CU进行编码。位置相关帧内预测组合(PDPC)被包括在TIMD模式的导出中。

[0237] 重复检查

[0238] 在图14和图15中,添加针对各个候选的重复检查(1440、1441、1442、1443、1444、1445和1530)。但是,重复也针对不相邻候选(1540)和基于历史的候选(1501)。重复检查在

于将索引cnt的当前候选的运动信息与先前候选的运动信息进行彼此比较。当该运动信息相等时,被认为是重复的,并且变量cnt不递增。当然,运动信息意味着针对各个列表(L0、L1)的帧间方向、参考帧索引和运动矢量。注意,与不同参考帧相对应的零候选不被认为是重复的。

[0239] MVTH

[0240] 在ECM中,对于该重复检查,引入了运动矢量阈值。该参数通过以下方式来改变相等性检查:如果两个运动矢量对于各个分量的绝对差小于或等于运动矢量阈值MvTh,则认为这两个运动矢量相等。对于常规合并模式,MvTh被设置为等于1,这对应于没有运动矢量阈值的传统重复检查。

[0241] 如下所定义的,对于模板匹配的常规合并模式,MvTh等于取决于当前CU中的亮度样本(nbSamples)的数量的值:

[0242] if (nbSamples < 64) MvTh = 1 << MV_FRACTIONAL = 16

[0243] else if (nbSamples < 256) MvTh = 2 << MV_FRACTIONAL = 32

[0244] else MvTh = 4 << MV_FRACTIONAL = 64

[0245] 其中MV_FRACTIONAL对应于编解码器的帧间分辨率。因此,在当前的ECM中,使用第16分辨率,因而MV_FRACTIONAL等于4。<<是左移位运算符。并且其中nbSamples = Height × Width(即,当前块的高度和宽度)。

[0246] 例如,如下所述,存在用于GEO合并导出并且还用于不相邻候选的重复检查的另一阈值MvThBDMVRMvdThreshold:

if (nbSamples < 64)

MvThBDMVRMvdThreshold = (1 << MV_FRACTIONAL) >> 2 = 4

else if (nbSamples < 256)

[0247] MvThBDMVRMvdThreshold = (1 << MV_FRACTIONAL) >> 1 = 8

else

MvThBDMVRMvdThreshold = (1 << MV_FRACTIONAL) >> 0 = 16

[0248] ARMC

[0249] 在ECM中,为了减少合并索引的位的数量,添加了具有模板匹配的合并候选的自适应重新排序(ARMC)。根据如图13中所计算的模板匹配成本,基于各个候选的成本对候选进行重新排序。在这种方法中,仅针对每个候选计算一个成本。该方法在导出该列表之后应用,并且仅对常规合并候选列表的前5个候选应用。应当理解,选择数量5是为了平衡重新排序处理的复杂性与潜在收益,由此可以对更大的数量(例如,所有候选)进行重新排序。

[0250] 图18给出了适用于如CTC中包含10个候选的常规合并候选列表的该方法的示例。

[0251] 该方法还适用于除时间候选之外的子块合并模式,以及适用于所有4个候选的常规TM模式。

[0252] 在建议中,该方法还被扩展到重新排序并选择要被包括在合并模式候选的最终列表中的候选。例如,在JVET-X0087中,所有可能的不相邻候选(1540)和基于历史的候选

(1501) 与时间不相邻候选一起被考虑,以获得候选的列表。该候选的列表是在不考虑候选的最大数量的情况下构建的。然后对该列表候选进行重新排序。仅来自该列表的正确数量的候选被添加到合并候选的最终列表。正确数量的候选对应于列表中前N个候选。在该示例中,正确数字是候选的最大数量减去最终列表中已经存在的空间和时间候选的数量。换句话说,不相邻候选和基于历史的候选与相邻空间和时间候选被分开地处理。所处理的列表用于补充已经存在于合并候选列表中的相邻空间和时间合并候选,以生成最终合并候选列表。

[0253] 在JVET-X0091中,ARMC用于从3个时间候选bi-dir、L0或L1中选择时间候选。所选择的候选被添加到合并候选列表。

[0254] 在JVET-X0133中,从使用ARMC重新排序的数个时间候选中选择合并时间候选。以同样的方式,所有可能的相邻候选都经受ARMC,并且这些候选中的多达9个可以被添加到合并候选列表。

[0255] 所有这些所建议的方法都使用对合并候选的最终列表进行重新排序的经典的ARMC,以对其进行重新排序。JVET X0087重新使用在不相邻和基于历史的候选的重新排序期间计算的代价,以避免附加的计算成本。JVET-X0133对合并候选的最终列表上的所有候选应用系统的重新排序。

[0256] 新ARMC

[0257] 自ARMC的首次实现以来,该方法被添加到数个其他模式。ARMC被另外应用于常规和模板匹合并模式、子块合并模式,并且还应用于IBC、MMVD、仿射MMVD、CIIP、具有模板匹配的CIIP和BM合并模式。另外,基于模板匹配成本对候选的列表进行重新排序的ARMC的原理也应用于AMVP合并候选导出以及用于帧内方法TIMD以选择可能性最高的预测器。

[0258] 此外,还针对仿射MVD、AMVP和SMVD方法的符号残差预测方法添加了该原理。

[0259] 对于GEO合并模式和具有模板匹配的GEO以及参考帧索引预测,还存在对该重新排序的使用的附加测试。

[0260] 除了ARMC的这种几乎系统的使用之外,还存在对于一些候选的导出处理的数个附加修改。

[0261] 例如,ECM4.0导出包括用于常规、TM和BM合并模式的候选导出的级联ARMC处理,如图19中针对常规和TM合并模式所描绘的。

[0262] 与先前的合并候选导出相比,首先,10个时间位置被检查并且在非重复检查之后被添加到时间候选的列表。该时间列表可以包含最多9个候选。此外,由于与用于合并候选导出的运动阈值相比MV阈值总是1并且不依赖于合并模式,因此对于该时间列表存在特定阈值。基于最多前9个非重复位置的列表,应用ARMC处理,并且如果第一时间候选与先前候选相比不重复,则仅将第一时间候选添加在合并候选的传统列表中。

[0263] 以同样的方式,在59个位置中导出不相邻空间候选。导出可以达到18个候选的第一非重复候选列表。但是运动阈值不同于用于时间导出和用于列表其余部分的运动阈值,并且不依赖于常规或模板合并模式。运动阈值被设置为等于BDMVR的mv。多达18个不相邻候选被重新排序,并且仅前9个不相邻候选被保持并添加在合并候选列表中。然后添加其他候选,除非列表已经包含最大数量的候选。另外,对于TM合并模式,列表MaxCandInitialList中的候选的最大数量高于最终列表可以包含的候选的最大数量Maxcand。

[0264] 然后,如图19所描绘的,针对包含MaxCandInitialList的中间列表的所有候选应用ARMC处理。在候选的最终列表中设置了候选的最大数量Maxcand。

[0265] 如模板匹配合并模式那样,类似的修改也应用于BM合并模式。

[0266] ARMC模板成本算法

[0267] 图17例示ARMC方法的模板成本计算。该处理中考虑的候选的数量NumMergeCandInList大于或等于列表可以包含的最大数Maxcand (1712)。

[0268] 对于列表中的各个候选 (1701),如果在针对时间和不相邻候选的第一ARMC处理期间没有计算成本 (1702),则将成本设置为等于0 (1703)。在实现中,与候选“i”相关联的未计算成本mergeList[i].cost被设置为等于最大值MAXVAL。如果用于当前块的上模板可用 (1704),则计算与当前块模板相比的失真 (1705)并将其添加到当前成本 (1706)。然后或否则,如果当前块的左模板可用 (1707),则计算与当前块模板相比的失真 (1708)并将其添加到当前成本 (1709)。然后,将当前合并候选的成本mergeList[i].cost设置为等于所计算的 (1710),并且更新列表 (1711)。在该示例中,我们考虑当前候选i被设置为关于其成本与其他候选的成本相比的位置。当所有候选都已进行时,列表中的候选的数量NumMergeCandInList被设置为等于列表中可能候选的最大数量Maxcand。

[0269] 图20示出图17的候选列表的更新 (1710)的图示。首先,将变量Shift设置为等于0。然后,在Shift小于图17的当前候选i并且当前候选的相关成本低于先前候选编号i-1-shift的成本时 (2002),使变量Shift递增 (2003)。当该循环结束时,如果变量Shift不同于0 (2007),则在位置i-shift处插入候选编号i (2010)。

[0270] 多假设预测 (MHP)

[0271] ECM中还添加了多假设预测 (MHP)。利用这种方法,可以使用针对每个块的多达四个运动补偿预测信号 (而不是如VVC中的两个)。这些个体的预测信号被叠加以形成整体预测信号。各个附加预测假设的运动参数可以通过指定参考索引、运动矢量预测器索引和运动矢量差来显式地用信号通知,或者通过指定合并索引来隐式地用信号通知。单独的多假设合并标志在这两个信号通知模式之间进行区分。

[0272] 对于空间候选、不相邻合并候选和基于历史的合并候选,多重假设参数值“addHypNeighbours”继承自候选。

[0273] 对于时间候选、零候选和成对候选,不保持多重假设参数值“addHypNeighbours” (它们是清楚的)。

[0274] LIC

[0275] 在ECM中,添加了局部照明补偿 (LIC)。LIC基于照明变化的线性模型。利用当前块的相邻样本和先前块的相邻样本来计算线性模型。

[0276] 在ECM中,仅针对单向预测启用LIC。通过标志的方式用信号通知LIC。对于合并模式,不传输LIC标志,而是以以下方式从合并候选继承LIC标志。

[0277] 对于空间候选、不相邻合并候选和基于历史的合并候选,继承LIC标志的值。

[0278] 对于时间候选和零候选,LIC标志被设置为等于0。

[0279] 对于成对候选,如图16所描绘的那样设置LIC标志的值。该图基于图12,并且添加了模块1620和1621,以及更新了模块1609、1612和1611。变量averageUsed被设置为等于假 (1620),如果对于当前列表,已经计算了针对成对候选的平均,则针对成对候选的LIC标志

LICFlag[cnt]被设置为等于假,并且变量averageUsed等于真(1609)。如果仅候选具有列表的运动信息(16121611),如果不使用平均,则更新LIC标志。并且它被设置为等于其当前值和候选的LICflag的值的OR运算。

[0280] 并且当成对候选是Bidir(例如等于3)时,LIC标志等于假。

[0281] 然而,如图16所示的算法仅在两个候选具有一个列表的运动信息、并且各个候选具有自己的列表的情况下允许LICflag等于不同于真的值。例如,候选0仅具有L0的运动信息,而候选1仅具有L1的运动信息。在这种情况下,LIC标志可以等于不同于0的其他值,但由于LIC仅用于单向,因而永远不会发生。因此成对候选的LIC标志总是等于假。因此,当可能需要LIC时,成对候选不能使用LIC。因此,这降低了候选的效率,并且避免了LIC针对随后的编码块的传播,从而降低了编码效率。

[0282] 此外,ECM软件中的重复检查导致一些效率低下。如图14和图15所描绘的,各个候选被添加到列表中,并且重复检查(1440、1441、1442、1443、1444、1445和1530)仅对变量cnt的递增(1405、1409、1413、1417、1423、1427、1508)有影响。此外,如图16所描述的,没有针对成对候选初始化变量BCWidx。因此,如果列表中添加的最后候选是重复候选,则成对候选的值BCWidx是该先前重复候选的值。这不是VVC中的情况,因为当候选被认为重复时,不添加这些候选。

[0283] 实施例

[0284] 在一个实施例中,当根据针对各个预测器计算的成本值来对预测器列表进行排序时,该成本值包括阈值和基于经解码样本所计算的成本,以获得将被比较的成本。因此,基于不是原始块的样本或基于当前块的最终的经解码样本来计算成本。

[0285] 该实施例在于通过“while((shift<i)AND(mergeList[i].cost+TH<mergeList[i-1-shift].cost))”改变图20的模块2002,其中添加阈值TH。

[0286] 图21例示替代实施例;该图基于图20。在该图中,循环(2102)的条件与图20相同。当已经确定Shift值时(2102)(2103),并且如果针对当前候选mergeList[i].cost的成本加上阈值TH大于或等于位置i-1-shift处的候选的成本(2108),则不将该候选(2109)插入位置i-shift处(2110)。因此,当前候选保持其位置,即使其成本mergeList[i].cost更好也是如此。

[0287] 替代地,并且数学上等价地,在一个实施例中,当根据针对各个预测器计算的成本值对预测器列表进行排序时,将两个成本之间的差值或绝对差值与阈值进行比较。成本包括基于不是原始块的样本的成本。

[0288] 该实施例由图23至图28例示。例如,在图23中,计算成本之间的绝对值(2305)以确定是否需要将候选设置到另一位置。

[0289] 这些实施例的优点是,候选可以被重新排序以改进候选列表的排序中的多样性,从而增加列表中较早的候选被选择的似然性,这实现了编码效率改进。

[0290] 在一个实施例中,当根据针对各个预测器计算的成本值来对预测器列表进行排序,并且将从该候选的初始列表中移除一些候选时,该成本值包括阈值和基于经解码样本所计算的成本,以获得将被比较的成本。因此,基于不是原始块的样本或基于当前块的最终的经解码样本来计算成本。

[0291] 该实施例可以通过修改图21来获得,以将未达到用阈值TH定义的条件候选

(2108) 设置在列表的末尾处或从列表中完全移除。为了将候选设置在列表的末尾处, Shift 值被设置为等于 $-Maxcand+1+i$ (2109), 并且将候选 `mergeList[i]` 插入在位置 $i-shift$ 处 (2110), 因此插入在位置 $Maxcand-1$ 处。

[0292] 替代地, 并且数学上等价地, 在一个实施例中, 根据针对各个预测器计算的成本值来从列表中移除至少一个候选, 将两个成本之间的差值或绝对差值与阈值进行比较。成本包括基于不是原始块的样本的成本。

[0293] 这些实施例的优点是, 在成本方面没有提供足够多样性的候选可以被移除或移动到列表的末尾, 这是因为该候选不太可能被选择, 这优化了列表的排序, 这实现了编码效率改进。

[0294] 预测器列表可以是帧内或帧间。

[0295] 可以应用的一些准则

[0296] 在一个实施例中, 位流中的至少一个句法元素与预测器列表相关联, 并且该排序影响句法元素的值。因此, 当应用排序时并且当该排序不同于在不考虑成本的情况下的排序时, 与预测器列表相关联的句法元素值可以不同。

[0297] 优点是编码效率改进, 这是因为根据这些预测器与表示当前块的数据之间的一些相关性对预测器进行排序, 这通常比预排序更有效率。

[0298] 在一个实施例中, 列表的各个预测器表示句法元素的值。例如, 该方法被应用于仿射、常规和 SMVD 帧间模式的运动矢量残差编码的符号 MVD 预测方法。对于这些模式, 重新排序后的列表被一个或多个句法元素使用。

[0299] 在一个实施例中, 列表的一些预测器表示索引。该实施例与合并 MMVD 和合并仿射 MMVD 方法的 MVD 编码的合并索引编码相对应, 或者与包括 IBC 和 GEO 模式的合并模式的经典合并索引编码相对应。以同样的方式, 也可以考虑帧间预测器的索引作为帧间模式 (AMVP)、仿射、IBC、SMVD。该实施例还可以应用于预测器列表被重新排序的 TIMD 帧内模式。尤其是, 重新排序后的候选的数量高于句法元素能够用信号通知的最大可能索引值。这意味着 $MaxCandInitialList$ 大于 $MaxCand$ 。

[0300] 可能成本计算

[0301] 在解码器侧确定候选的相对“成本”的处理的准确性是重新排序处理的有效性如何的决定因素。以下实施例提供了对成本确定处理的改进, 这得到了更准确的列表、更低的计算成本的复杂性或这两者。

[0302] 重要的是要注意, 用于进行基于重新排序的算法可以被改变以优先考虑重新排序处理的准确性或复杂性 (速度)。

[0303] 以下实施例旨在产生相对成本的准确指示, 同时使操作的整体复杂性最小化。

[0304] 在一个示例中, 重新排序基于包括样本之间的度量的成本。将与各个候选相关联的样本与另一样本进行比较以产生成本的相对度量。

[0305] 例如, 可以基于该预测器块的相邻样本和当前块的相邻样本来计算候选的成本。这样的样本在解码器侧是容易获得的。

[0306] 可以在与列表中的候选相对应的 2 个块预测器的样本之间计算成本。例如, 当候选是双预测候选时, 成本可以是两个块预测器之间的失真。下面更详细地讨论双预测候选。

[0307] 还可以与另一预测器候选进行比较来计算成本。例如, 一个其他预测器候选可以

是可能性最高的候选或预测器。候选的成本是通过其样本和该可能性最高的候选的样本来计算的。

[0308] 可以对相邻样本的子集或预测器的样本的子集来计算成本。例如,如果存在可以用于确定成本的多个相邻样本,则对这些样本进行采样以降低计算的复杂性。

[0309] 可以基于与根据另一分辨率的图像相对应的样本来计算成本。与来自更高分辨率的图像的高相似性是低成本的良好指示(即,良好预测器)。

[0310] 可以对用于计算成本的样本的值进行预处理。假设仅需要相对成本值(即,仅顺序是重要的),预处理值意味着更简单的计算,并且不太可能显著影响重新排序处理的功效。根据预处理,所计算的成改进了重新排序处理。

[0311] 在上述示例中,成本可以是失真的度量,诸如绝对差和(SAD)、绝对变换差和(SATD)、误差平方和(SSE)或结构相似性指数度量(SSIM)。

[0312] 替代地,成本是失真的度量,并且可以将权重应用于该失真。也可以考虑速率或所估计的速率。或者阈值。

[0313] 成本也可以是加权成本,其中权重根据预测器或候选的类型或候选在列表中的初始位置而不同。

[0314] 在实施例中,成本与样本的数量成比例。

[0315] 例如,样本的数量是用于当前块的样本的数量,并且将失真(基于样本值的成本)除以该数量。

[0316] 在替代实施例中,样本的数量是在失真的计算中使用的样本的数量或失真的比较的数量。并且将失真(基于样本值的成本)除以该数量。

[0317] 在一个实施例中,将基于样本值的成本除以用于失真的比较的数量,并且乘以样本的数量是块。

[0318] 例如,对于与块的上行和左排匹配的模板(对于行和排,仅一个样本),与候选I相关联的成本 $\text{mergeList}[i].\text{cost}$ 被设置为等于 $\text{mergeList}[i].\text{cost} = ((\text{height} \times \text{width}) \times \text{mergeList}[i].\text{cost}) / (\text{height} + \text{width})$ 。

[0319] 该实施例的优点是成本与块大小成比例,因此,针对准则的参数可适应于所有块大小。更准确地,这提供了如以下实施例中所定义的Lambda值的更有效率的使用。

[0320] 另外的准则

[0321] 在一个实施例中,阈值是如在速率失真准则 $J = D + \lambda R$ 中使用的拉格朗日参数Lambda(λ)。在编码器侧使用速率失真准则,以确定针对块的最佳编码可能性,或者更一般地,以在编码可能性集合中选择最佳编码可能性。

[0322] 该实施例提供了显著的编码效率改进。如果我们考虑两个候选或预测器cand1(候选1)和cand2(候选2),其分别具有其失真D1和D2。并且如果D1>D2但其RD成本J1和J2是J1<J2,我们可以认为这些候选在RD意义上太接近或太冗余,这是因为用信号通知cand2的速率高于其失真效益。这可以写成:

[0323] $J1 < J2$

[0324] $\Leftrightarrow D1 + \lambda R1 < D2 + \lambda R2$

[0325] $\Leftrightarrow D1 - D2 < \lambda (R2 - R1)$

[0326] $\Leftrightarrow (D1 - D2) / (R2 - R1) < \lambda$

[0327] 由于视频编解码器中的许多索引使用一元max码。我们可以认为 $R2-R1=1$ 。因此 $D1-D2<\lambda$ 。因此,如果 $|D1-D2|<\lambda$ 或 $D1<D2+\lambda$,我们可以认为cand1和cand2是冗余的。

[0328] 因此,在该实施例中,在当前候选的失真和另一候选的失真之间的绝对值小于Lambda时,候选被认为是冗余的。

[0329] 该实施例在图21至图28中例示。例如,在图23中,将最小(min)绝对失真差(2305)与Lambda值进行比较(2308)。

[0330] 该公式特别有效率,并且在候选与其先前候选之间的速率差不为1时也特别有效率。

[0331] 先前实施例不考虑两个候选/预测器之间的速率差来确定它们是否冗余。因此,为了改进编码效率,计算该速率差以确定候选与另一候选相比是否冗余。当与候选/预测器列表相关的句法元素是用一元码以外的东西进行编码时,该实施例是有用的。例如,用哥伦布码对常规MMVD或仿射MMVD的合并运动差进行编码。在这种情况下,一个索引值与其随后索引值之间的bin或位的数量差可以是0、1、2或多于2。

[0332] 在附加实施例中,考虑实际速率。因此,使用CABAC时的实际速率被考虑在内。在这种情况下,当用一元max码对索引进行编码时,也可以应用该实施例。

[0333] 可以通过利用 λ ($R2-R1$) 在图23至图28中改变Lambda值来实现该实施例,或者通过添加Lambda乘以速率而改变针对候选的成本来实现该实施例。类似地,对于图20至图22,准则可以是 $D2+\lambda \times R2 < D1+\lambda \times R1$ 或 $D2+\lambda (R2-R1) < D1$,其中D2是mergeList[i].cost。(在图20中,循环(2002)中的准则寻找比当前候选更好的候选。但是确定候选是否冗余的准则是 $D1 < D2+\lambda (R2-R1)$ 。例如,该实施例在于通过下式来改变图20的模块2002:

[0334] “while((shift<i) AND (mergeList[i].cost+TH*mergeList[i].rate<mergeList[i-1-shift].cost+TH*mergeList[i-1-shift].rate))”

[0335] 在一个实施例中,当候选在第一排序之后被重新排序时,如图23至图28中所描述的,用于确定为候选是冗余的准则(2305)包括一个失真的阈值的乘法,例如, $D2-Th.D1 < 0$,或替代地 $Th.D2-D1 < 0$ 或 $Th2.D2-Th1.D1 < 0$ 。

[0336] 该实施例的优点是编码效率改进。尤其是,该准则能够基于失真的比例来调节两个候选的冗余。由于该准则,具有小失真的两个候选不被认为是冗余的。相反,与先前的准则相比,具有高失真的两个候选将更频繁地被认为是冗余的。

[0337] 在一个实施例中,准则包括平衡2个失真的至少一个阈值,并且添加阈值。例如,如先前定义的准则变为: $|D2-Th.D1| - \lambda < 0$ 或替代地 $|Th2.D2-Th1.D1| - \lambda < 0$ 。

[0338] 该实施例的优点是针对一些情况的编码效率改进,在这些情况中,针对一些候选的失真的惩罚给出了编码效率改进。例如,该惩罚取决于当前块的距离和候选的类型。

[0339] Lambda设置

[0340] 在编码器实现中,Lambda值取决于数个参数,这些参数对失真与速率之间的折衷以及如何计算失真和如何估计速率有影响。

[0341] 在一个实施例中,Lambda取决于一个或多个参数,并且在解码器侧计算Lambda的值。优点是不需要附加的速率来传输Lambda值。

[0342] 在一个实施例中,Lambda值(或阈值)在编码器侧被导出或获得并且在解码器侧(在位流中)被传输。

[0343] 该实施例的优点是,编码器实现可以设置与实现或序列类型更对应的Lambda值。

[0344] 在实施例中,在解码器侧使用的Lambda或传输的Lambda取决于用于当前条带或图片的QP值。在另一实施例中,当在块级别设置QP时,基于块QP值在块级别调整Lambda值。QP值是影响具有块残差的重建块的质量并因此影响速率失真折衷的最重要参数。

[0345] 在实施例中,在解码器侧使用的Lambda或传输的Lambda取决于当前帧与其参考帧之间的图片顺序计数距离或时间距离。作为一些模式,作为常规合并模式,针对一个候选的POC距离可以彼此不同。因此,当数个POC距离是可能的时,优选地考虑当前帧与其参考帧之间的最小POC或最小时间距离。当前帧与其参考帧之间的时间距离或者到最接近参考帧的时间距离影响块预测器的质量。因此,影响具有和不具有块残差的重建块质量。

[0346] 在实施例中,在解码器侧使用的Lambda或传输的Lambda取决于时间ID。在许多视频编码标准中,帧具有指示用于时间多样性的深度的时间ID。因此,该时间ID可以影响帧所需的质量折衷,因此Lambda也能够依赖于该参数。

[0347] 在实施例中,在解码器侧使用的Lambda或传输的Lambda取决于时间方向。时间方向意味着参考帧来自过去或来自未来或来自这两个方向。已知当块预测器来自这两个方向时,平均而言,帧间预测器的质量更好。因此,对速率失真折衷具有影响。作为一些模式,作为常规合并模式,一个候选的时间方向可以彼此不同,优选的是Lambda取决于大量的时间方向,因此,来自一个时间方向或来自两个时间方向。

[0348] 在实施例中,在解码器侧使用的Lambda或传输的Lambda取决于单预测或双预测。已知,当块预测器来自两个块预测器时,平均而言,即使两个参考块来自相同的时间方向,该预测器的质量也更好。因此,对速率失真折衷具有影响。如在诸如常规合并模式等的一些模式中,一个候选可能是双预测的,而另一候选是单预测的,优选的是Lambda取决于是否可以使用双预测。

[0349] 在一个实施例中,在解码器侧使用的Lambda或传输的Lambda取决于当前条带的参考帧的QP值。在实施例中,为了避免针对一些模式的复杂Lambda设置,Lambda取决于各个或这两个方向上最接近的参考帧的QP值或者各个参考帧方向的最小QP值。当与先前实施例组合时,Lambda可以取决于针对参考帧所选择的QP的QP值之间的差。参考的QP影响参考的质量,并因此影响当前帧的QP,因此影响速率失真折衷。

[0350] 在实施例中,在解码器侧使用的Lambda或传输的Lambda取决于所使用的失真的度量。已知,Lambda取决于编码器侧使用的失真(SAD、SSE等)。因此,对于解码器侧,Lambda也取决于失真的类型。

[0351] 在实施例中,在解码器侧使用的Lambda或传输的Lambda取决于用于计算失真的样本的类型。事实上,如前所述,可以基于样本的不同类型针对不同的重新排序或移除算法来应用准则。这也可以被认为是设置Lambda值。例如,由于模板匹配使用比当前块更少的样本,因此针对更少的样本计算失真,因此可以考虑该失真。

[0352] 如前所述,该实施例的替代方案是根据样本的数量来调节失真。

[0353] 在一个实施例中,在解码器侧使用的Lambda或传输的Lambda取决于预测的类型。在编码器侧,Lambda值可以取决于帧内/帧间预测的类型,在解码器侧使用的Lambda也取决于该特性。

[0354] 在实施例中,在解码器侧使用的Lambda或传输的Lambda取决于GOP配置类型。例

如,根据如随机接入配置RA或低延迟配置或仅帧内配置的配置类型,Lambda可以被移位或者是基于除当前帧的QP之外的另一QP的Lambda。

[0355] 在实施例中,在解码器侧使用的Lambda或传输的Lambda基于在编码器处使用以在列表的最佳候选/预测器等中进行选择的Lambda公式,并且在解码器侧使用Lambda。所使用的公式可以例如通过索引在位流中被用信号通知到解码器,该索引指示从可用公式的预定列表中使用哪个公式来计算Lambda。

[0356] 注意,在其他实施例中,阈值或Lambda值可以取决于上述任何因素的组合,其可以被最佳地引入以提供编码效率和复杂性的最佳平衡。

[0357] 设置正确Lambda的一个可能实现是使用传统公式来导出它(其通常取决于QP值),并且根据如先前定义的参数来移位该QP值。在给出显著收益的特定示例中,使用被用于在最佳合并候选中进行选择的Lambda公式。在当前帧处于随机接入配置时,利用QP-4来设置Lambda公式。其中QP是针对当前条带/帧的QP。并且在帧处于低延迟配置时,用于在解码器处获得Lambda值的QP值是当前QP。

[0358] 当然,Lambda可以被存储在预定义的表中,而不是使用公式。在这种情况下,表的各个值与QP值相关。

[0359] 阈值 (TH) 不是Lambda

[0360] 如针对先前实施例所描述的,Lambda值理论上是用于准则的良好值,但是取决于数个参数并且与编码器实现有关。因此,在其他实施例中,其他值可以用作阈值。

[0361] 在实施例中,阈值是如图25所描绘的最小值,这在下面的标题“算法”下描述。

[0362] 在实施例中,基于所有候选 (NumMergeCandInList) 的成本值或根据第一Maxcand候选来计算阈值。

[0363] 在实施例中,阈值是各个候选与其先前的候选相比的失真之间的差的平均值。利用该准则,与图21至图28相关的所有算法都是有效的,并且 λ 值或TH值被该平均值代替。

[0364] 这些实施例的优点是阈值不依赖于数个参数或不需要被传输。

[0365] 可能算法

[0366] 基于所定义的准则对列表进行重新排序或者基于准则移除候选的算法可以是特定的,这是因为成本是在解码器侧从不是当前块的样本的样本计算出的,并且此外,各个模式具有其特定性。

[0367] 在一个实施例中,在如图20所述的排序处理期间应用本发明的准则,这包括改变循环的准则。在实现的另一示例中,图21和图22也涉及该实施例。

[0368] 该实施例的优点是对复杂性的影响小,这是因为仅需要一个循环来应用所提出的准则。

[0369] 图23例示在基于成本的第一排序之后应用的一个算法。因此,在该算法的开始处,列表的所有候选根据其成本/失真被排序。例如,在ARMC方法之后。候选的列表包含NumMergeCandInList候选,并且NumMergeCandInList可以大于或等于最终列表中使用的候选的最大数量Maxcand。

[0370] 在该图中,重新排序的数量MaxNumReorder被限制为列表中的候选的数量NumMergeCandInList减去一(2301)(2302)。例如,如实施例(2302)中定义的,在可能的重新排序的数量大于1并且候选与其随后候选之间的最小差大于或等于Lambda(阈值)值时,应

用该算法。确定候选*i*与其随后的候选*i+1*之间的最小成本差(2304)(2305)(2306),并且变量CandToMove被设置为等于该随后的候选*i+1*。该候选与前驱候选相比被认为是冗余的。如果该最小成本低于Lambda(阈值)(2308),则将该候选从其位置移动(2310)到位置MaxNumReorder-1(2309)。因此,候选被设置在最远的位置,但是在被算法认为是冗余的先前候选之前。然后使MaxNumReorder递减。

[0371] 该算法可以被总结为:

[0372] -确定2个连续候选之间的最小成本差。

[0373] -如果该最小差小于Lambda(或其他阈值),则将该候选设置在列表中的最远位置(并且特别是在被算法认为冗余的先前候选之前)

[0374] 图29给出了通过该算法获得的重新排序的示例。在该示例中,Lambda(即阈值)被设置为等于16。在第一步“步骤1”的开始处,6个合并候选的列表根据其成本(mergeList[i].cost)从最低的成本2264到最高的成本2750被排序。然后计算各个候选与其前驱候选之间的绝对成本差(2305),并且识别最小值(2306)。在图29的该示例中,在步骤1中,最小绝对成本差是5(min=5),并且相关候选CandToMove是候选索引*i*=4。该成本5小于等于16的Lambda值(2308)。因此,在“步骤2”中将候选索引4移动到列表的末尾,并且使MaxNumReorder递减。因此,针对循环的第二步骤不考虑该候选。在步骤2中,所考虑的列表包含5个候选,并且最小绝对成本差是8(与合并候选索引*i*=1相对应)。针对步骤3,该候选被插入在合并候选列表中的位置4,并且针对步骤3不考虑该候选。在“步骤3”期间,最小绝对成本差是18,其大于Lambda值16(2308),因此算法如(2302)中所定义的那样停止。在步骤3结束时,合并列表候选不是根据其成本值(从最小到最大)而是根据算法来排序,该算法通过将可能是冗余的候选(由于其在成本方面与其他候选的相似性)向列表的末尾移动来提供多样性改进。更具体地,冗余候选被移动到跟在目前为止被认为是非冗余的候选之后的位置。

[0375] 与图21的实施例相比,该实施例的一个优点是基于最冗余的候选对冗余进行排序。另一优点是,该重新排序首先通过根据该冗余设置在末尾来移除冗余候选,然后较少冗余候选在最终集合中。此外,移除/重新排序的数量也是有限的。因此,与如图21的实施例相比,该实施例产生了编码效率改进。但是这需要引入更多复杂性的第二重新排序处理。

[0376] 在另一实施例中,MaxNumReorder不被设置为等于候选的数量,而是被设置为Maxcand或替代地被设置为更小的值。事实上,当列表与具有索引的候选的数量(Maxcand)相比显著大(NumMergeCandInList)时,所提出的重新排序设置无效率的候选。因此,优选将可能的重新排序限制为少量值。例如,利用ARMC的合并MMVD,MVD位置列表包含96个候选,并且用信号通知MVD的最大索引仅为12。因此,在这种情况下,优选使用被设置等于12的MaxNumReorder。

[0377] 图25例示在实施例中的图23的算法的替代算法。在该算法中,与先前实施例相对地,不存在与Lambda/阈值(2502)和最小成本差的比较。最冗余的候选CandToMove总是被插入在列表的末尾处。为了获得编码效率,与图23的实施例相比,可能重新排序的最大数量MaxNumReorder由变量LimitReorder限制(2502)。因此,在这种情况下,仅可以应用LimitReorder重新排序。请注意,该实施例类似于图23,其中Lambda/阈值被设置为总是比2个失真之间的可能绝对差大的最大值。

[0378] 该实施例的优点是不需要在解码器侧导出或传输Lambda值,因此其适用于所有编码器选择算法。但是编码效率被降低。然而,对于某些模式,该限制是有效的。

[0379] 在图23的实施例的替代实施例中,算法仅移除候选。这可以通过修改循环的准则(2302)来实现。在这种情况下,当MaxNumReorder大于或等于Maxcand而不是1时,循环停止。另外,为了更好的编码效率,仅在列表的第一部分中确定最小失真差(2304)(2305)(2306)(2307)。例如,将会(当前)在最终列表(Maxcand)中的候选。这可以通过改变2304的准则并且将i限制为Maxcand来实现。

[0380] 与如针对图23所述的重新排序和移除相比,平均而言,仅移除候选是没有效率的,除非影响最终列表中的候选的最大数量Maxcand。

[0381] 在一个实施例中,从列表中移除被认为是冗余的候选,并且使最终列表的最大值相应地减小。

[0382] 图24例示该实施例,在该图中类似于图23,每次将候选设置在列表的末尾(2410)时,最终列表中的候选的最大数量Maxcand被减小(2412)。当相关句法元素没有利用一元max码被编码时或者当Maxcand值小时,这是特别有效率的。但是这产生了解析问题,因为解码器需要完全重建来解析相关句法元素。

[0383] 图26例示图23的算法的一个替代算法。与图23相比,插入冗余候选的位置并不总是MaxNumReorder-1(2309),而是候选变为非冗余的第一位置(2613)(2614)(2615)。这通过在当前冗余候选(2613)(CandToMove)之后的候选中识别第一候选(2614)来实现,其中针对该第一候选,冗余候选(CandToMove)与该候选之间的失真差大于Lambda(阈值)。针对冗余候选的插入位置insertPos被设置为等于该第二候选i的位置(2615)。

[0384] 该算法可以被总结为:

[0385] -确定2个连续候选之间的最小成本差;

[0386] -如果该最小差小于Lambda,则连续候选被认为是冗余的;

[0387] -将冗余候选的失真与列表中的随后候选的失真进行比较;

[0388] -识别具有高于Lambda的失真差的第一随后候选的位置;

[0389] -将冗余候选设置在该识别出的位置处。

[0390] 在附加的实施例中,当可能重新排序的最大数量的值MaxNumReorder大于最终列表中候选的最大数量Maxcand(2612)时,insertPos维持等于MaxNumReorder-1。

[0391] 与图23的实施例相比,该实施例改进了编码效率。事实上,失真成本不是与原始块相比的真实失真,并且第一重新排序通常是正确的,因此对于被认为是冗余的候选的位置改变的该限制通常是正确的。

[0392] 该实施例对于常规合并模式、模板匹配合并模式和BM合并模式以及当对于IBC合并模式启用时特别有效率。

[0393] 请注意,可以调节该图的算法。例如,一个附加步骤可以检查列表中的合并候选的数量NumMergeCandInList是否大于2以应用算法。以避免在确信候选不被重新排序时应用算法。事实上,仅在列表包含多于2个候选时,算法才可以改变候选的顺序。另外,算法中考虑的候选的数量可以是已经被ARMC重新排序的候选的数量,这是因为其他候选不具有相关的失真。在另一可能实现中,步骤2602可以改变为从重新排序候选的最大数量NumMergeCandInList到2的“for”循环(代替“while”循环),并且步骤2611可以被移除。当小

于最小min时,算法停止。所描述的所有算法可以是基于将初始候选位置与其重新排序后的位置相关联的表以及包含各个候选的相关成本的另一表的。在这种情况下,为了避免复杂的交换,可以仅在ARMC算法和多样性重新排序算法结束时进行对最终重新排序候选列表的交换。

[0394] 图27示出在实施例中基于失真的第一重新排序之后的重新排序算法的另一种可能实现。与图23的算法相对地,该算法不是基于最小失真差的,而是专用于识别非冗余候选。

[0395] 对于列表(2701)的各个候选cand(2702),算法寻找第一随后的候选cand2(候选2)(2704),其在本发明中定义的准则的意义上不是冗余的(2705)。如果没有候选是非冗余的,则考虑下一候选,并且应用准则来确定候选是冗余的还是非冗余的。否则,第一非冗余候选(2706)被插入(2707)在候选(insertPos=cand+1)之后。当然,插入意味着将该非冗余候选从其原始位置移除。

[0396] 该算法可以被总结为:

[0397] -对于候选,确定列表中的第一、随后的非冗余候选;

[0398] -在该候选之后插入该非冗余候选。

[0399] 该实施例对于候选几乎都是冗余的列表尤其有效率。因此,例如,该实施例对于MVD残差位置接近的MMVD合并、仿射或常规是有效率的。以同样的方式,出于同样的原因,该算法对于MVD符号预测特别有效率。

[0400] 图28例示图20的附加步骤。在该图中,当候选被认为是非冗余的(2805)时,还测试该候选与当前候选(2809)的所有先前候选(2808)相比是否不冗余。

[0401] 该算法可以被总结为:

[0402] -对于候选,确定列表中的第一、随后的非冗余候选;

[0403] -在该候选之后插入该非冗余候选,除非该非冗余候选与该候选的所有先前候选相比是冗余的。

[0404] 如先前实施例那样,该实施例对于候选几乎都是冗余的列表尤其有效率。因此,例如,该实施例对于MVD残差位置接近的MMVD合并、仿射或常规是有效率的。以同样的方式,出于同样的原因,该算法对于MVD符号预测特别有效率。由于重新排序的这种进一步限制,与先前实施例相比,获得了更好的编码效率。即使图26的算法对于常规、TM、BM合并模式给出了比该实施例更好的结果,该算法对于这些模式也给出了显著的收益。

[0405] 在替代实施例中,SizeCandList没有被设置为等于候选的数量NumMergeCandInList(2701)(2801),而是将其设置为Maxcand或替代地设置为更小的值。事实上,当列表与具有索引的候选的数量相比明显大时,所提出的重新排序将设置没有效率的候选。因此,优选将可能的重新排序限制为少量值。例如,利用ARMC的合并MMVD,针对一个候选的MVD位置列表是96,并且用信号通知MVD的索引仅是12。因此,在这种情况下,优选使用被设置为等于12的MaxNumReorder。

[0406] 请注意,如果NumMergeCandInList与Maxcand之间的差不高,则Maxcand的该限制(2701)(2801)可以是可选的。

[0407] 其他实施例

[0408] 以下特征可以与上述实施例有用地组合以形成本发明的其他实施例。

[0409] 在一个实施例中,当列表仅包含2个候选时,不应用所提出的方法。这相当于将Th设置为等于0。例如,当使用图20的实施例时,这产生编码效率提高。事实上,当候选列表中仅存在2个候选时,优选将具有最佳失真的候选设置为第一候选。

[0410] 对于在第一排序之后应用重新排序的其他实施例,例如对于图23,当仅存在2个候选时,准则不改变任何东西。

[0411] 在实施例中,在当前块的仅一个模板可用时,相对于可用模板的成本成比例地设置针对列表的所有候选的成本。

[0412] 例如,当不可用模板是左模板时,针对当前块的成本为 $\text{cost} = \text{CostUp} + (\text{CostUp} \times \text{height}) / \text{width}$ 。

[0413] 其中高度和宽度对应于块的高度和宽度,因为这里我们认为上模板是块的上侧的上行,且左模板是块的左侧的左行。但是可以在对公式进行调节的情况下考虑更长或更短的模板。乘法和除法可以被移位操作代替以降低复杂性。

[0414] 该实施例的优点是编码效率改进。事实上,对于一些速率失真折衷,Lambda值是固定的。因此,Lambda适合于这种速率失真折衷。利用与对于相同大小的所有块的相等数量的样本相对应的失真,观察到编码改进。

[0415] 替代地,在这种情况下,可以修改Lambda来代替修改各个失真成本。对于相同的示例,当左模板不可用时,Lambda或阈值变为 $\text{Lambda} = \text{Lambda} / (1 + (\text{height}) / \text{width})$ 。该计算可以被修改为仅使用移位而不使用除法以降低复杂性。

[0416] 与先前实施例相比,该实施例的优点是复杂性降低,因为在这种情况下,新的Lambda被计算一次,来代替改变各个失真成本。

[0417] 在当前ECM中,针对常规、TM和BM合并模式的运动矢量阈值对于各个类型的候选是不同的。这在候选子集中创造了一种多样性。

[0418] 在一个实施例中,当应用所提出的发明时,用于导出的MV阈值对于所有合并候选类型是相同的。例如,针对时间和不相邻候选的运动矢量阈值与所有其他类型的候选的运动矢量阈值相同。

[0419] 这改进了编码效率,因为由所提出的本发明的方法创建了多样性,该方法与使用不同的运动矢量阈值相比更有效率。

[0420] 以同样的方式,如图19中所描绘的基于合并类型的ARMC给出了一些收益,这是因为存在基于合并类型的多样性控制。

[0421] 在一个实施例中,当启用本发明的方法时,针对利用ARMC导出和排序的所有候选应用ARMC处理。例如,与图19相比,多达9个时间候选和多达22个(或18个,取决于实现)不相邻候选被添加到其他合并候选(B1、A1等)的列表。为了使用与针对每个合并类型的ARMC处理至少相同数量的ARMC比较,候选的初始列表不限于所定义的MaxCandForFirstList,而是限于MaxCandForFirstList+时间候选的数量和不相邻候选的数量。然而,该数量可以被限制为合并候选的缓冲器可以包含的最大数量。例如,可以固定为30。

[0422] 该实施例的一个优点是编码效率增加,这是因为由于本发明中提出的方法,优选控制多样性。另外,该实施例降低了复杂性,这是因为ARMC不应用于时间列表或不相邻列表中的重复候选。

[0423] 在一个实施例中,时间候选的数量和/或不相邻候选的数量被减少。优点是所有这

些候选的成本计算的复杂性降低。

[0424] 当在ARMC处理之后添加成对候选已经示出编码效率改进。但是,在与基于合并类型的ARMC处理的竞争中,由于可能候选的数量显著增加,收益显著减少。

[0425] 在一个实施例中,当启用所提出的方法时,在排序处理期间或之后添加成对候选。可选地,基于由ARMC的排序给出的第一候选以及列表中的一个或多个其他候选来构建在这种情况下成对候选。可选地,由于本发明中提出的准则,将这些所添加的候选与其成本和列表中的其他候选的成本相比来重新排序,或者在该第一重新排序处理之后应用第二重新排序处理。

[0426] 优点是编码效率改进。事实上,成对候选有时是与列表中的现有候选太接近的候选。因此,在使用ARMC基于成本来排序的情况下,这些太冗余的候选位于列表的顶部,而没有创建足够的多样性。当将本发明的实施例应用于列表时,列表更加多样,并且成对候选被不同地排序,或者当成对候选太冗余时被移除。因此,可以获得编码效率提高。

[0427] 在一个附加的实施例中,当在其他传统候选的重新排序处理之后添加成对候选时,该成对候选在独立的列表中。在这种情况下,表包含各个候选的顺序,并且另一个表包含相关成本。这些表是在ARMC处理和多样性重新排序处理期间设置的。然后根据所确定的顺序,基于来自这两个列表的候选来设置最终列表。优点是复杂性降低,这是因为可以减小缓冲器的最大大小。

[0428] 在一个实施例中,成对候选的数量被限制为固定值。例如,该值为4。

[0429] 该实施例的优点是限制应用ARMC处理的候选的最大数量。

[0430] 在一个实施例中,可以添加的成对候选的最大数量被定义为最终列表中的目标候选的数量或固定值。该成对候选的最大数量被认为限制了初始列表中的候选的最大数量(在ARMC处理之前)。因此,在这种情况下,NumMergeCandInList被设置为等于NumMergeCandInList减去可以添加的成对候选的最大数量。也可以添加偏移(正或负)。应当注意,不可能知道在第一重新排序之后构建成对候选时可以添加的成对候选的真实数量。

[0431] 该实施例的优点是复杂性降低和限制。因为具有ARMC成本计算的候选的最大数量是有限的。

[0432] 在一个实施例中,对于一些模式不添加成对候选。例如,不将成对候选添加到TM合并列表。

[0433] 优点也是复杂性降低。

[0434] 预测性编码应用

[0435] 以上实施例已经被描述为用于在视频或图像编码和解码的上下文中预测图像部分(预测单元)的实现。然而,设想上述方法可应用于可以使用预测器或预测器候选的任何压缩技术。例如,在点云编码中,预测器可以针对云中的点或用于网格的元素。替代地,预测器可以针对从另一音频数据预测要编码或解码的音频数据的音频数据部分。

[0436] 除非另有明确说明,否则可以组合所有所描述的实施例。事实上,许多组合是协同的,并且可以产生大于其部分总和的效率提高。

[0437] 重新排序第一候选

[0438] 对于一些特定的预测器列表,不针对一个或多个第一候选来应用ARMC。如前

所述,例如,不针对子块合并列表的子块时间预测器 (STMVP) 来应用ARMC。多样性准则和相关算法可以适于这些特定情况。

[0439] 另外,针对数个模式应用ARMC,然后这些模式一起竞争。ARMC将最可能的预测器设置在各个列表的起始处。该最可能的预测器可以被认为是所谓的“第零”预测器,这是因为通常期望从重新排序中排除该预测器,使得第零预测器在初始(作为Cand0)和在任何重新排序处理之后(作为NewCand0,参见图30)这两者都处于列表的起始处的位置处。然而,在各个列表的起始处设置类似候选,这在编码效率方面是不期望的,因为在预测模式之间没有足够的多样性。

[0440] 在一个实施例中,基于多样性准则的重新排序被应用于具有成本的候选。对于未被重新排序且未计算成本的STMVP候选的示例,该候选维持在列表的起始处。以同样的方式,利用多样性准则,ARMC(如图31所示)之后获得的候选NewCand1维持在列表的第二位置处。

[0441] 一个替代实施例是计算STMVP候选的成本并且将多样性算法应用于所有子块合并候选。在这种情况下,STMVP候选的计算可以以启用多样性算法或以多样性准则的阈值/Lambda值为条件。如果阈值/Lambda值等于零,则不计算STMVP的成本。

[0442] 在一个实施例中,未评估列表起始处的候选的成本被设置为等于0,并且多样性算法使用该候选作为列表的第零候选。在STMVP候选的示例中,不针对STMVP计算失真成本,而是将相关值设置为等于0,并且对包括该候选的列表中的所有子块候选应用多样性算法。当然,多样性算法不改变STMVP候选的位置,但是可以改变图31的“第一”候选(尽管其处于接在第零候选之后的位置)NewCand1(新候选1)的位置。在该实施例中,当第一候选被认为是被重新排序的时,意味着其成本接近于0。

[0443] 该实施例的优点是与先前实施例相比编码效率改进。在该实施例中,第一候选被认为与其自身成本和多样性准则的阈值相比太冗余。与先前实施例相比,该候选可以被移动到另一位置。

[0444] 该实施例的一个替代方案是计算STMVP候选的成本并且将其设置在列表的起始处,不管计算出的成本如何都是如此。因此,ARMC算法不考虑针对重新排序的该成本。然后可以对列表的所有候选应用多样性准则。

[0445] 与先前实施例相比,多样性准则可以正确地评估第零候选(即STMVP候选)与其后继NewCand1之间的多样性。请注意,在多样性算法中,列表的第二个位置处的候选并不总是相同的,这取决于各个循环。

[0446] 因此,该实施例的优点是编码效率改进,这是因为根据正确的数据来计算准则。

[0447] 在重新排序中考虑第零候选

[0448] 在一个实施例中,多样性算法可以对列表中的所有候选或具有成本的所有候选进行重新排序。对于列表的起始处的候选,由于没有先前的候选,并且为了在多样性算法中考虑该候选,将其成本与阈值进行比较。如果我们考虑阈值是Lambda值,并且如果我们考虑D0是NewCand0(新候选0)的失真,则多样性准则变为: $D0 < \lambda$ 。

[0449] 图31例示基于图26的该实施例。仅在该图中,步骤3103与步骤2603相比被改变。因此,最小成本由列表中的第一候选的成本进行初始化,并且表示可以移动的候选的索引的变量CandToMove由值0进行初始化。

[0450] 该实施例的优点是编码效率改进,这是因为列表的起始处的候选也被考虑用于重新排序。

[0451] 在一个附加的实施例中,将第零候选重新排序的可能性被限于一些模式。例如,将第零候选重新排序的可能性被限于BM合并模式或TM合并模式。事实上,TM和BM在互相不启用时更有效率。这意味着这两个模式以及常规合并模式产生类似的块预测器。在候选导出以针对这些模式在候选集合之间获得一种多样性方面存在差异。但是所提出的方法给出了更好的结果,因为在集合之间产生了多样性。与图31相比,仅针对相关模式应用步骤3103,否则将最小值设置为等于MAX。

[0452] 该实施例的优点是编码效率改进,这是因为在候选集合之间创建的多样性。

[0453] 在一个实施例中,对于列表中的第零候选,阈值不同于在其他候选的多样性算法中使用的阈值。

[0454] 在这种情况下,对于列表的第零候选,准则是 $D_0 < \lambda_1$,而对于其他候选,准则是 $|D_i - D_{(i+1)}| < \lambda_2$ 。对于图31的示例,步骤3108适于在CandToMove等于0或其他情况以及步骤3114时改变Lambda值。

[0455] 优点是参数的更好设置,这是因为对第零候选进行重新排序的可能性不同于对其他预测器进行重新排序。

[0456] 在一个实施例中,针对第零候选的阈值小于多样性准则的阈值。

[0457] 事实上,与其他候选相比,第零候选的移动或移除是被限制的。

[0458] 与阈值或Lambda值相关的先前描述的所有实施例(例如,设置或计算阈值或Lambda值的方法)可以应用于上述实施例。

[0459] 在一个实施例中,可以根据准则 $D_i < \lambda$,基于候选的成本值来移动或移除候选。

[0460] 在一个实施例中,从预测器集合中移除失真小于Lambda的所有候选。例如,在当前列表中的预测器的最大数量(MaxNumReorder)大于最终列表可以包含的预测器的最大数量(Maxcand)时,从列表中移除成本低于Lambda的所有预测器,直到该MaxNumReorder被设置为等于Maxcand。

[0461] 在一个实施例中,失真小于Lambda的所有候选被移动到另一位置,其中在该另一位置处,先前候选具有高于Lambda值的成本失真。

[0462] 图32例示该实施例。对于各个候选(3201),测试其成本是否小于Lambda(3202)。如果是这种情况,则将该候选(3203)移动到候选成本大于或等于Lambda的第一位置(3205),并且将当前候选(3203)插入紧接着的位置(3207)。当然,仅在存在成本值大于或等于Lambda的至少一个候选时,才应用该算法。

[0463] 在一个替代实施例中,成本低于Lambda的候选被移动到列表的末尾处。

[0464] 优点是与先前实施例相比根据准则考虑了所有候选,并且具有小于Lambda的成本的这些候选被更多地惩罚。当预测器列表中的预测器的数量高时,该实施例更有效率。

[0465] 在一个附加实施例中,仅针对一些模式应用所提出的移动或移除。例如,将第一候选重新排序的可能性被限于BM合并模式或TM合并模式。事实上,如前所述,当另一个未被启用时,TM和BM更有效率。

[0466] 该实施例的优点是由于在候选集合之间创建的多样性而提高了编码效率。

[0467] 发明的实现

[0468] 图33示出了根据本发明实施例的系统191、195,其包括编码器150或解码器100中的至少一个以及通信网络199。根据实施例,系统195用于处理并向用户提供内容(例如,用于显示/输出或流式传输视频/音频内容的视频和音频内容),用户例如通过包括解码器100的用户终端或可与解码器100通信的用户终端的用户界面访问解码器100。这样的用户终端可以是计算机、移动电话、平板电脑或能够向用户提供/显示(提供的/流式传输的)内容的任何其他类型的装置。系统195经由通信网络199获得/接收位流101(以连续流或信号的形式(例如,在显示/输出较早的视频/音频时))。根据实施例,系统191用于处理内容并存储经处理的内容,例如经处理以供在稍后的时间显示/输出/流式传输的视频和音频内容。系统191获得/接收包括原始图像序列151的内容,该内容由编码器150接收和处理(包括利用根据本发明的去块滤波器进行滤波),并且编码器150生成将经由通信网络191传送到解码器100的位流101。然后,位流101以多种方式传送到解码器100,例如,可以由编码器150预先生成并作为数据存储于通信网络199中的存储设备中(例如,在服务器或云存储装置上),直到用户从存储设备请求内容(即,位流数据)为止,此时数据从存储设备传送/流式传输到解码器100。系统191还可以包括内容提供设备,以用于向用户提供/流式传输(例如,通过传送要在用户终端上显示的用户界面的数据)存储在存储设备中的内容的内容信息(例如,内容的标题和用于识别、选择和请求内容的其他元/存储位置数据),并且用于接收和处理用户对内容的请求,使得所请求的内容可以从存储设备传送/流式传输到用户终端。替代地,编码器150生成位流101,并且在用户请求内容时将其直接传送/流式传输到解码器100。然后,解码器100接收位流101(或信号),并利用根据本发明的去块滤波器进行滤波,以获得/生成视频信号109和/或音频信号,然后用户终端使用视频信号109和/或音频信号来向用户提供所请求的内容。

[0469] 根据本发明的方法/处理的任何步骤或本文描述的功能可以用硬件、软件、固件或其任何组合来实现。如果以软件实施,则步骤/功能可以作为一个或多个指令或代码或程序或计算机可读介质而存储在一个或多个基于硬件的处理单元上或者经由一个或多个基于硬件的处理单元发送,并且由一个或多个基于硬件的处理单元执行,所述处理单元诸如为可编程计算机器,其可以是PC(“个人计算机”)、DSP(“数字信号处理器”)、电路、电路系统、处理器和存储器、通用微处理器或中央处理单元、微控制器、ASIC(“专用集成电路”)、现场可编程逻辑阵列(FPGA)或其他等效集成或离散逻辑电路系统。因此,如本文中所使用的术语“处理器”可指前述结构或适合于实现本文中所描述的技术的任何其他结构中的任一个。

[0470] 本发明的实施例还可以通过各种装置或设备来实现,包括无线手机、集成电路(IC)或JC集合(例如,芯片集)。本文中描述各种组件、模块或单元以说明被配置为进行这些实施例的装置/设备的功能方面,但未必需要由不同硬件单元实现。而是,各种模块/单元可组合于编解码器硬件单元中或由互操作硬件单元的集合提供,所述硬件单元包括结合合适软件/固件的一个或多个处理器。

[0471] 本发明的实施例可以通过读出并执行记录在存储介质上的计算机可执行指令(例如,一个或多个程序)以进行上述实施例中的一个或多个的模块/单元/功能以及/或者包括用于进行上述实施例中的一个或多个的功能的一个或多个处理单元或电路的系统或设备的计算机来实现,并且可以通过由系统或设备的计算机进行的方法来实现

现,例如,从存储介质读出并执行计算机可执行指令以进行上述实施例中的一个或多个的功能和/或控制一个或多个处理单元或电路以进行上述实施例中的一个或多个的功能。计算机可以包括单独的计算机或单独的处理单元的网络,以读出并执行计算机可执行指令。计算机可执行指令可以例如经由网络或有形存储介质从诸如通信介质等的计算机可读介质提供给计算机。通信介质可以是信号/位流/载波。有形存储介质是“非暂时性计算机可读存储介质”,其可以包括(例如)硬盘、随机访问存储器(RAM)、只读存储器(ROM)、分布式计算系统的存储装置、光盘(例如致密盘(CD)、数字多功能光盘(DVD)或蓝光光盘(BD)TM)、快闪存储器装置、存储卡等中的一个或多个。至少一些步骤/功能也可以由机器或专用组件(诸如FPGA(“现场可编程门阵列”)或ASIC(“专用集成电路”))在硬件中实现。

[0472] 图34是用于实现本发明的一个或多个实施例的计算装置3600的示意性框图。计算装置3600可以是诸如微计算机、工作站或轻型便携式装置等的装置。计算装置3600包括连接到以下各项的通信总线:-中央处理单元(CPU)3601,诸如微处理器等;-用于存储本发明的实施例的方法的可执行代码的随机访问存储器(RAM)3602以及适于记录实现根据本发明的实施例的用于对图像的至少一部分进行编码或解码的方法所需的变量和参数的寄存器,其存储容量例如可以通过连接到扩展端口的可选RAM进行扩展;-用于存储用于实现本发明的实施例的计算机程序的只读存储器(ROM)3603;-网络接口(NET)3604,其通常连接至通信网络,要处理的数字数据通过该通信网络被传输或接收,网络接口(NET)3604可以是单个网络接口,或者由一组不同的网络接口(例如,有线和无线接口,或不同种类的有线或无线接口)组成,在运行在CPU 3601中的软件应用的控制下,数据包被写入网络接口用于传输或者从网络接口读取以进行接收;-用户接口(UI)3605,其可以用于从用户接收输入或向用户显示信息;-硬盘(HD)3606,其可以被设置为大容量存储装置;-输入/输出模块(IO)3607,其可以用于从/向外部装置(诸如视频源或显示器等)接收/发送数据。可执行代码可以存储在ROM 3603中、HD3606上或诸如盘等的可移动数字介质上。根据变型,程序的可执行代码可以经由NET 3604借助于通信网络来接收,以在被执行之前存储在计算装置3600的存储部件(诸如HD 3606等)之一中。CPU 3601适于控制和指导根据本发明的实施例的一个或多个程序的软件代码的指令或部分的执行,该指令被存储在前述存储部件之一中。例如,在通电之后,CPU 3601能够执行来自从程序ROM 3603或HD 3606加载了指令之后的主RAM存储器3602的、与软件应用有关的那些指令。这种软件应用在由CPU 3601执行时使得进行根据本发明的方法的步骤。

[0473] 还应理解,根据本发明的其他实施例,在诸如计算机、移动电话(蜂窝电话)、平板或能够向用户提供/显示内容的任何其他类型的装置(例如,显示设备)等的用户终端中提供根据上述实施例的解码器。根据又一实施例,在图像捕获设备中提供根据上述实施例的编码器,该图像捕获设备还包括用于捕获和提供内容以供编码器进行编码的照相机、摄像机或网络照相机(例如,闭路电视或视频监视照相机)。以下参见图35和图36提供两个这样的示例。

[0474] 图35是例示包括网络照相机3702和客户端设备202的网络照相机系统3700的图。

[0475] 网络照相机3702包括摄像单元3706、编码部3708、通信单元3710和控制单元3712。

[0476] 网络照相机3702和客户端设备202经由网络200相互连接以能够彼此通信。

[0477] 摄像单元3706包括镜头和图像传感器(例如,电荷耦合器件(CCD)或互补金属氧化

物半导体 (CMOS)), 并捕获对象的图像并基于该图像生成图像数据。该图像可以是静止图像或视频图像。

[0478] 编码部3708通过使用以上说明的所述编码方法或以上描述的编码方法的组合来对图像数据进行编码。

[0479] 网络照相机3702的通信单元3710将由编码部3708编码的经编码的图像数据传输至客户端设备202。

[0480] 此外,通信单元3710接收来自客户端设备202的命令。命令包括用于设置用于编码部3708的编码的参数的命令。

[0481] 控制单元3712根据通信单元3710接收到的命令来控制网络照相机3702中的其他单元。

[0482] 客户端设备202包括通信单元3714、解码部3716和控制单元3718。

[0483] 客户端设备202的通信单元3714向网络照相机3702传输命令。

[0484] 此外,客户端设备202的通信单元3714从网络照相机3702接收经编码的图像数据。

[0485] 解码部3716通过使用以上说明的所述解码方法或以上说明的解码方法的组合来对经编码的图像数据进行解码。

[0486] 客户端设备202的控制单元3718根据由通信单元3714接收的用户操作或命令来控制客户端设备202中的其他单元。

[0487] 客户端设备202的控制单元3718控制显示设备2120以显示由解码部3716解码的图像。

[0488] 客户端设备202的控制单元3718还控制显示设备2120以显示用于指定网络照相机3702的参数(包括用于编码部3708的编码的参数)的值的GUI(图形用户界面)。

[0489] 客户端设备202的控制单元3718还根据对显示设备2120所显示的GUI的用户操作输入来控制客户端设备202中的其他单元。

[0490] 客户端设备202的控制单元3718根据对显示设备2120所显示的GUI的用户操作输入来控制客户端设备202的通信单元3714,以将用于指定网络照相机3702的参数的值的命令传输至网络照相机3702。

[0491] 图36是例示智能电话3800的图。

[0492] 智能电话3800包括通信单元3802、解码/编码部3804、控制单元3806和显示单元3808。

[0493] 通信单元3802经由网络200接收经编码的图像数据。

[0494] 解码/编码部3804对通信单元3802接收到的经编码的图像数据进行解码。

[0495] 解码/编码部3804通过使用以上说明的所述解码方法来对经编码的图像数据进行解码/编码。

[0496] 控制单元3806根据通信单元3802接收到的用户操作或命令控制智能电话3800中的其他单元。

[0497] 例如,控制单元3806控制显示单元3808以显示由解码/编码部3804解码的图像。智能电话3800还可以包括传感器3812和图像记录装置3810。以这种方式,智能电话3800可以记录图像,(使用以上描述的方法)对图像进行编码。

[0498] 智能电话3800可以随后(使用以上描述的方法)对经编码的图像进行解码并且将

其经由显示单元3808进行显示,或者经由通信单元3802和网络200将其传输到另一装置。

[0499] 替代和修改

[0500] 虽然已经参考实施例描述了本发明,但是应当理解,本发明不限于所公开的实施例。本领域技术人员将理解,在不脱离所附权利要求限定的本发明的范围的情况下,可以进行各种改变和修改。本说明书(包括任何所附权利要求、摘要和附图)中公开的所有特征、和/或所公开的任何方法或处理的所有步骤,可以以任何组合进行组合,除了这样的特征和/或步骤中的至少一些相互排斥的组合之外。除非另外明确说明,否则本说明书(包括任何所附权利要求、摘要和附图)中所公开的各个特征可以由用于相同、等同或类似目的的替代特征代替。因此,除非另有明确说明,否则所公开的各个特征仅为通用系列等效或类似特征的一个示例。

[0501] 还应理解,上述比较、确定、评估、选择、执行、进行或考虑的任何结果(例如,在编码或滤波处理期间作出的选择)可以在位流中的数据(例如,指示结果的标志或数据)中指示或可从位流中的数据确定/推断,使得所指示或确定/推断的结果可以用于处理而不是实际上例如在解码处理期间进行比较、确定、评估、选择、执行、进行或考虑。

[0502] 在权利要求中,词语“包括”不排除其他元素或步骤,并且不定冠词“a”或“an”不排除多个。仅仅在相互不同的从属权利要求中记载不同特征的事实并不指示这些特征的组合不能被有利地使用。

[0503] 权利要求中出现的附图标记仅作为说明,并且不应对权利要求的范围产生限定作用。

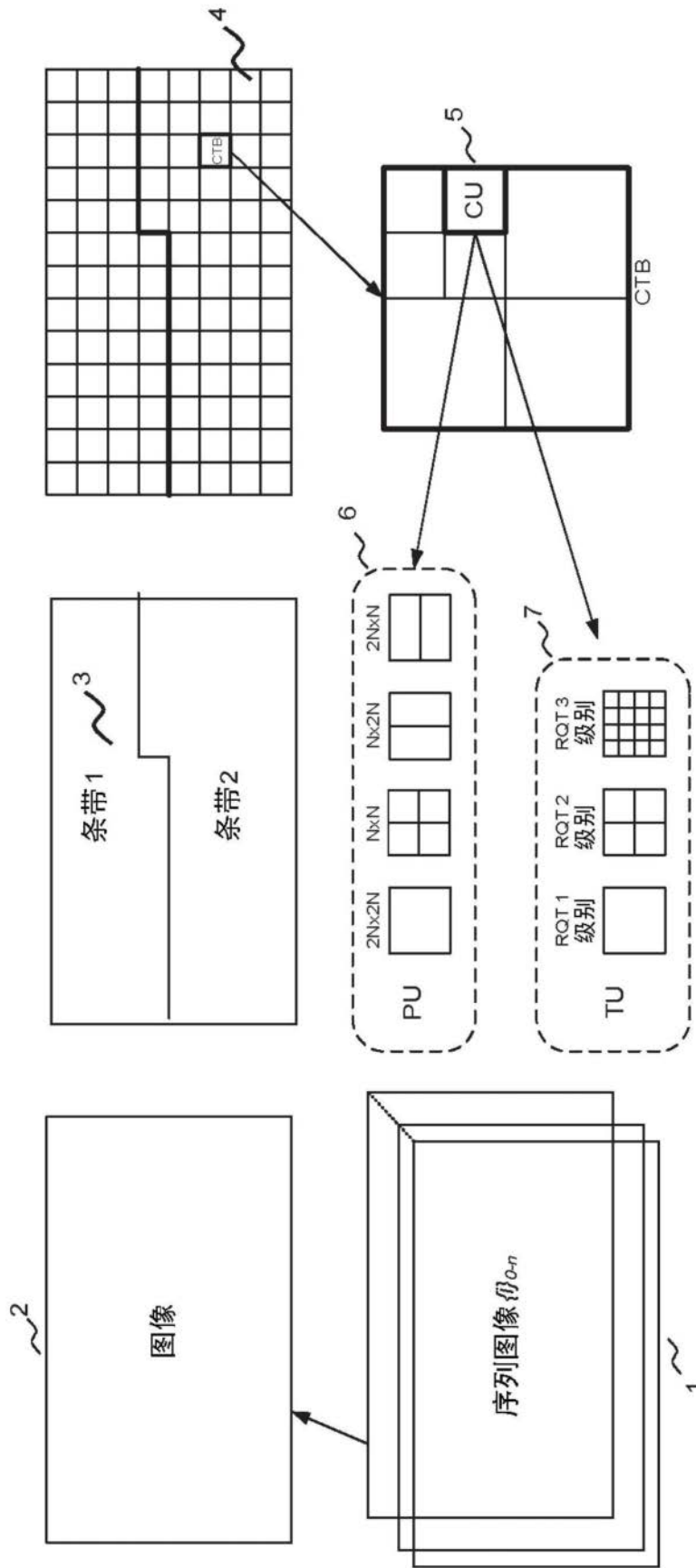


图1

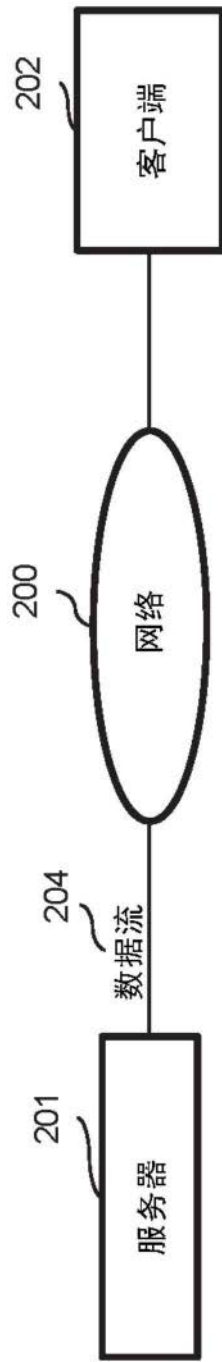


图2

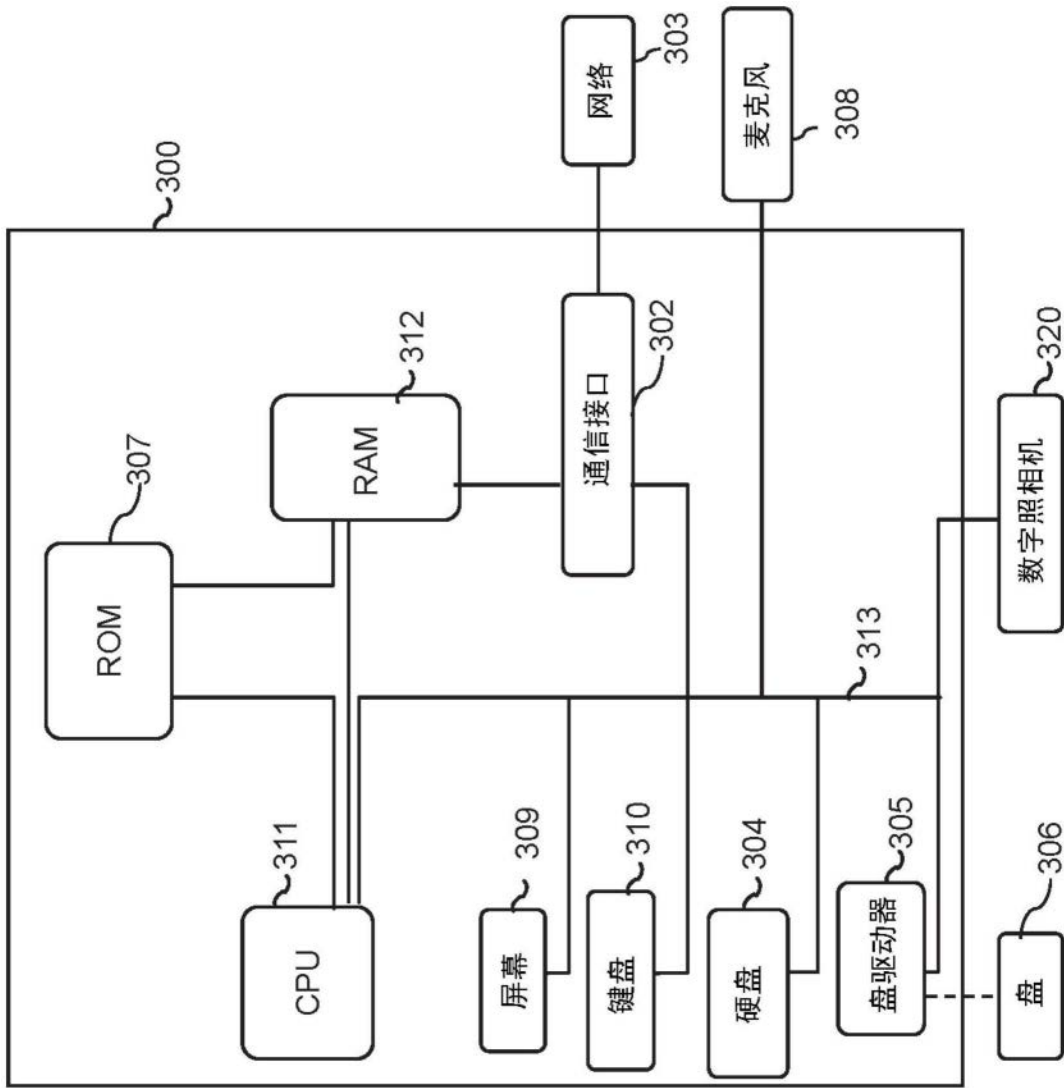


图3

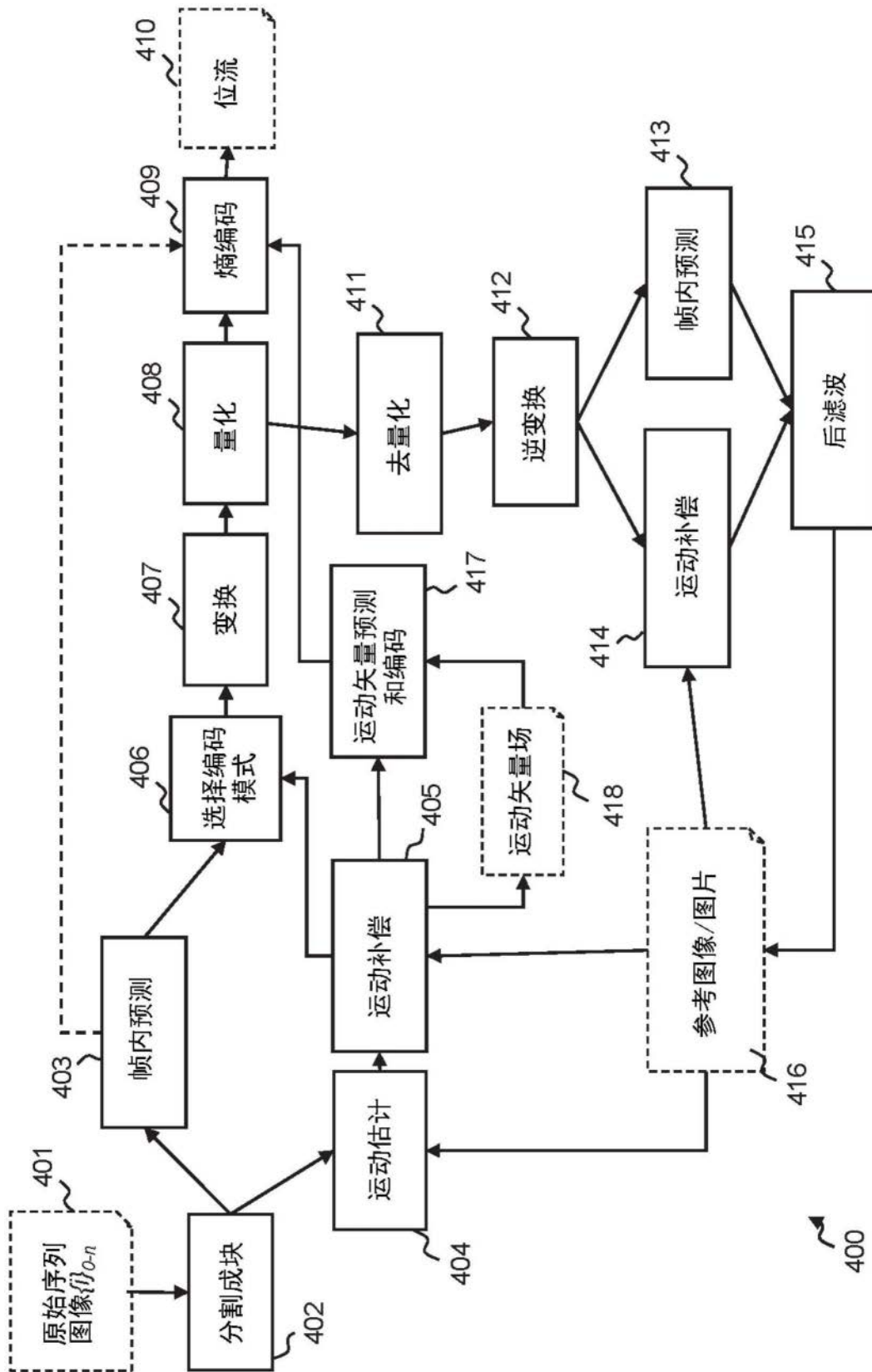


图4

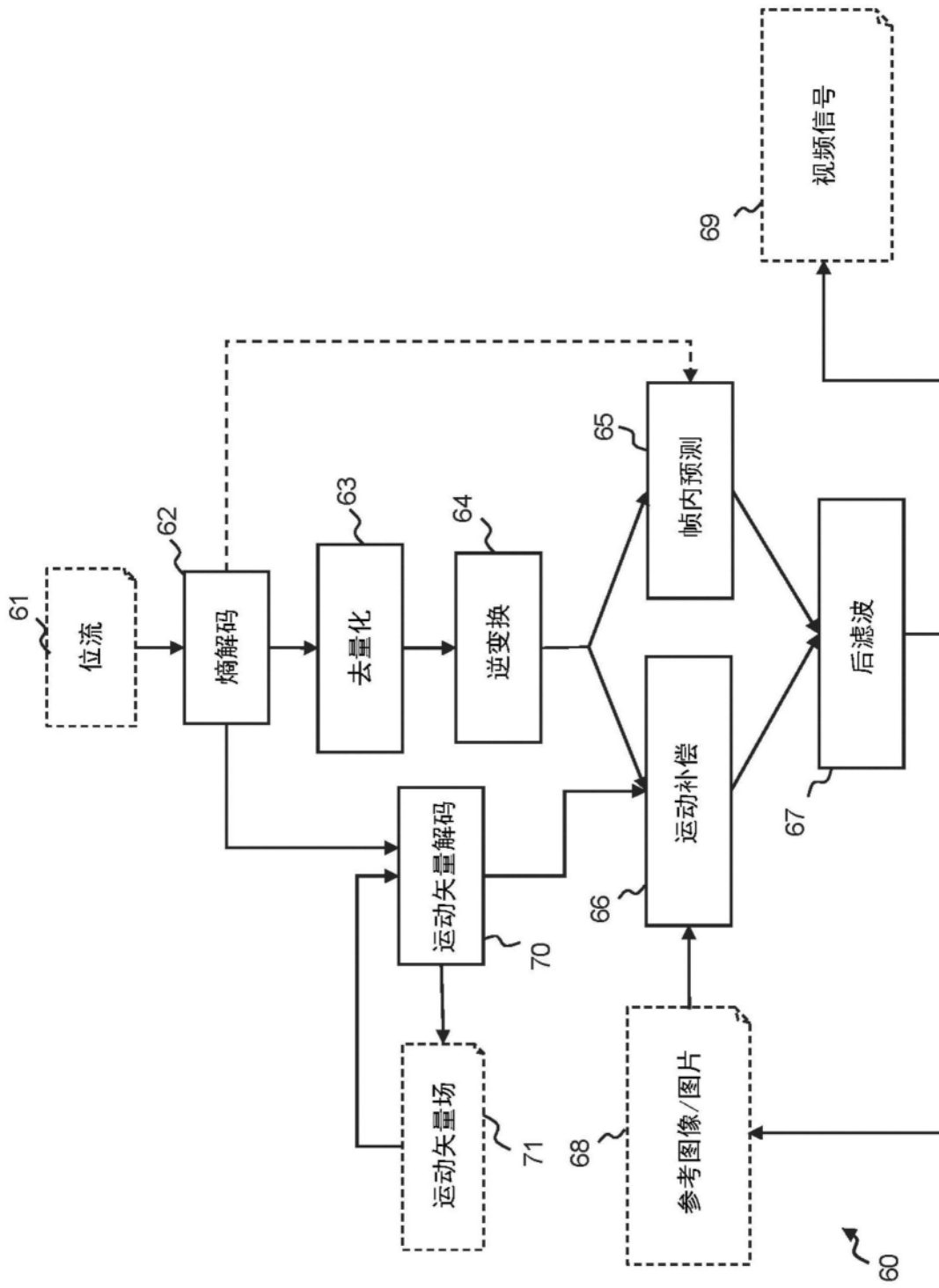


图5

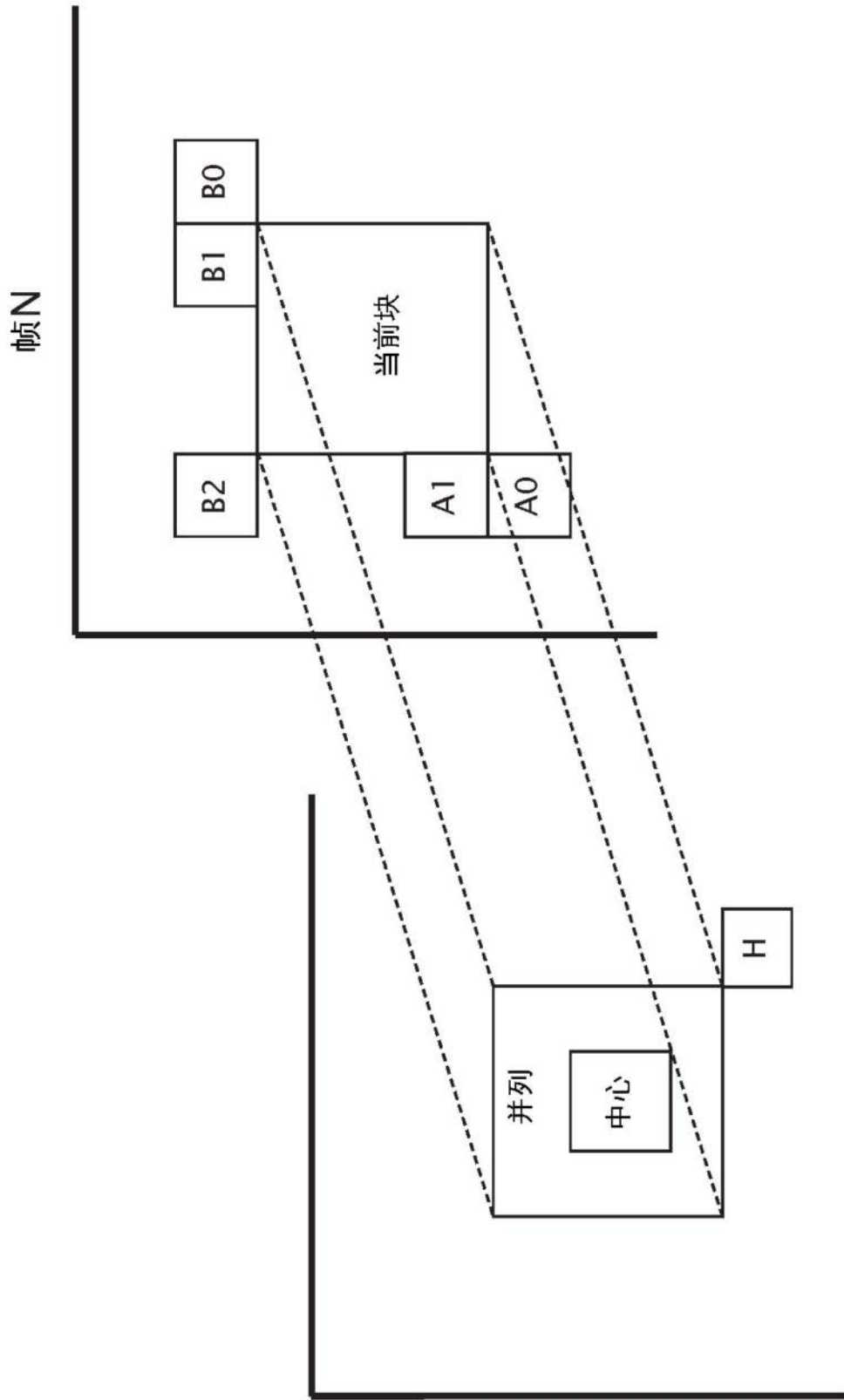


图6

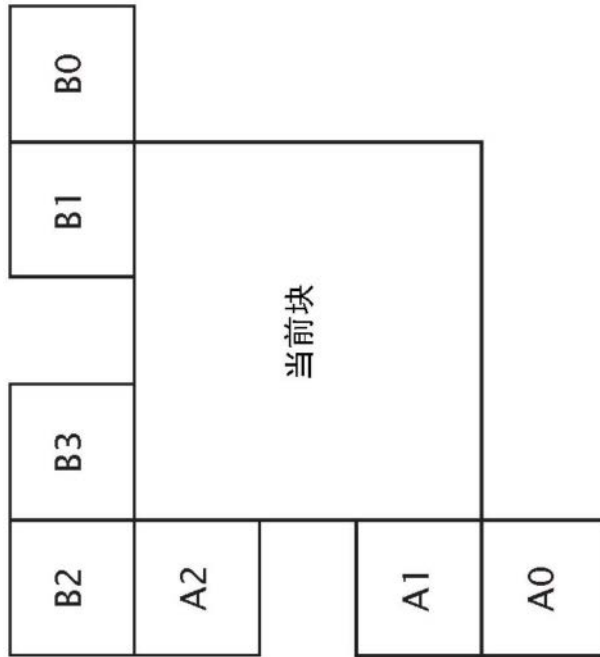


图7

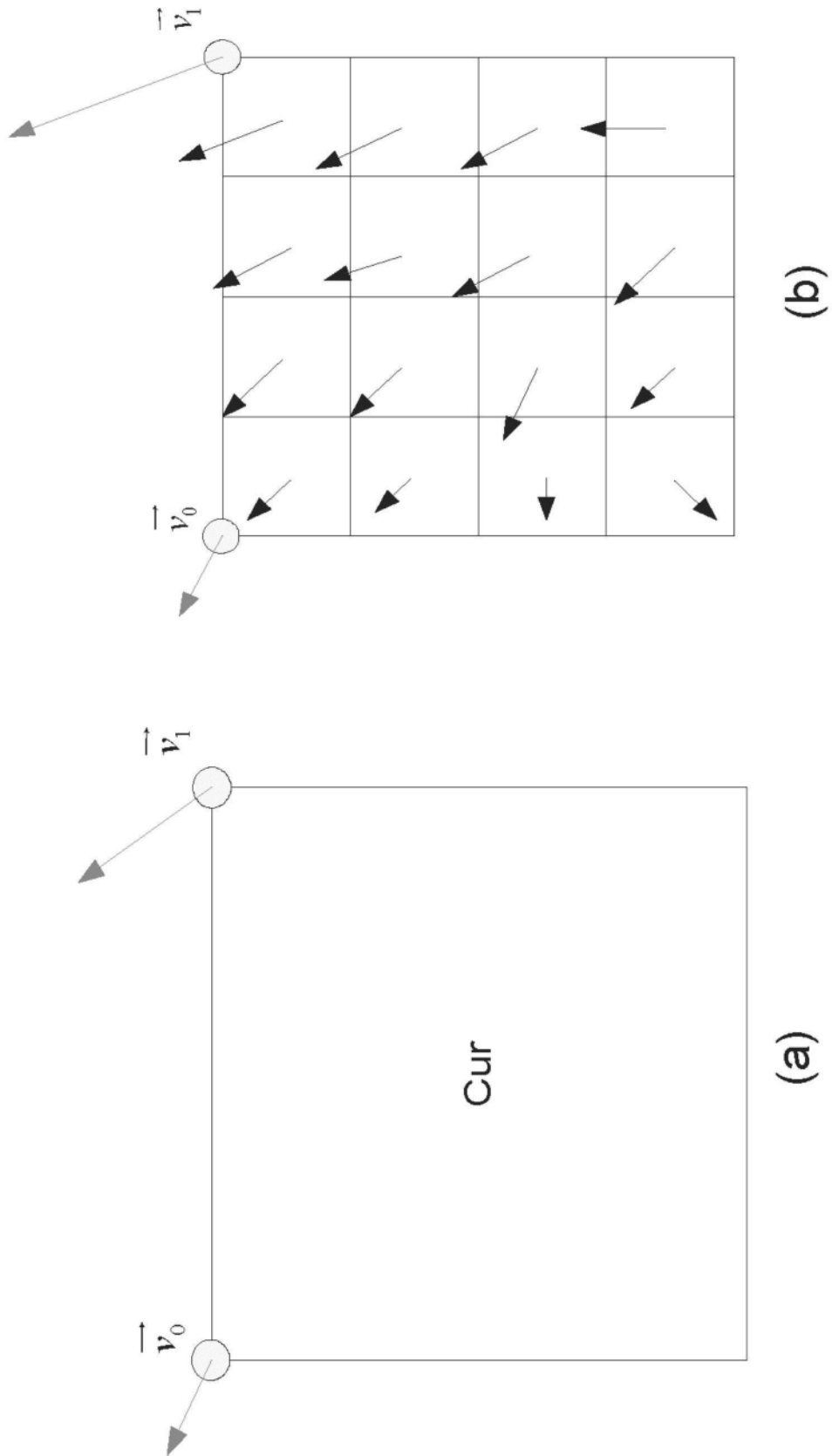


图8

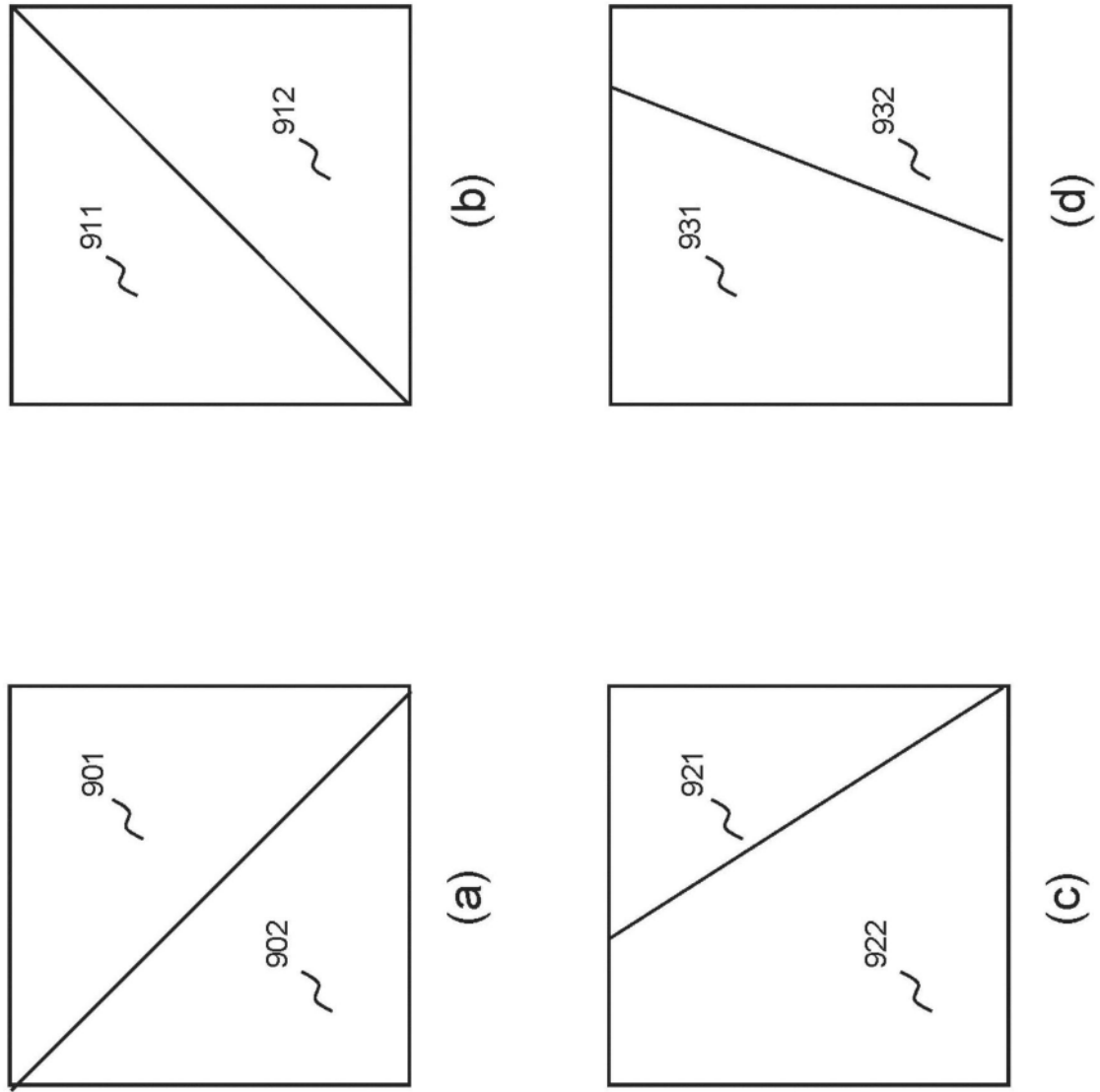


图9

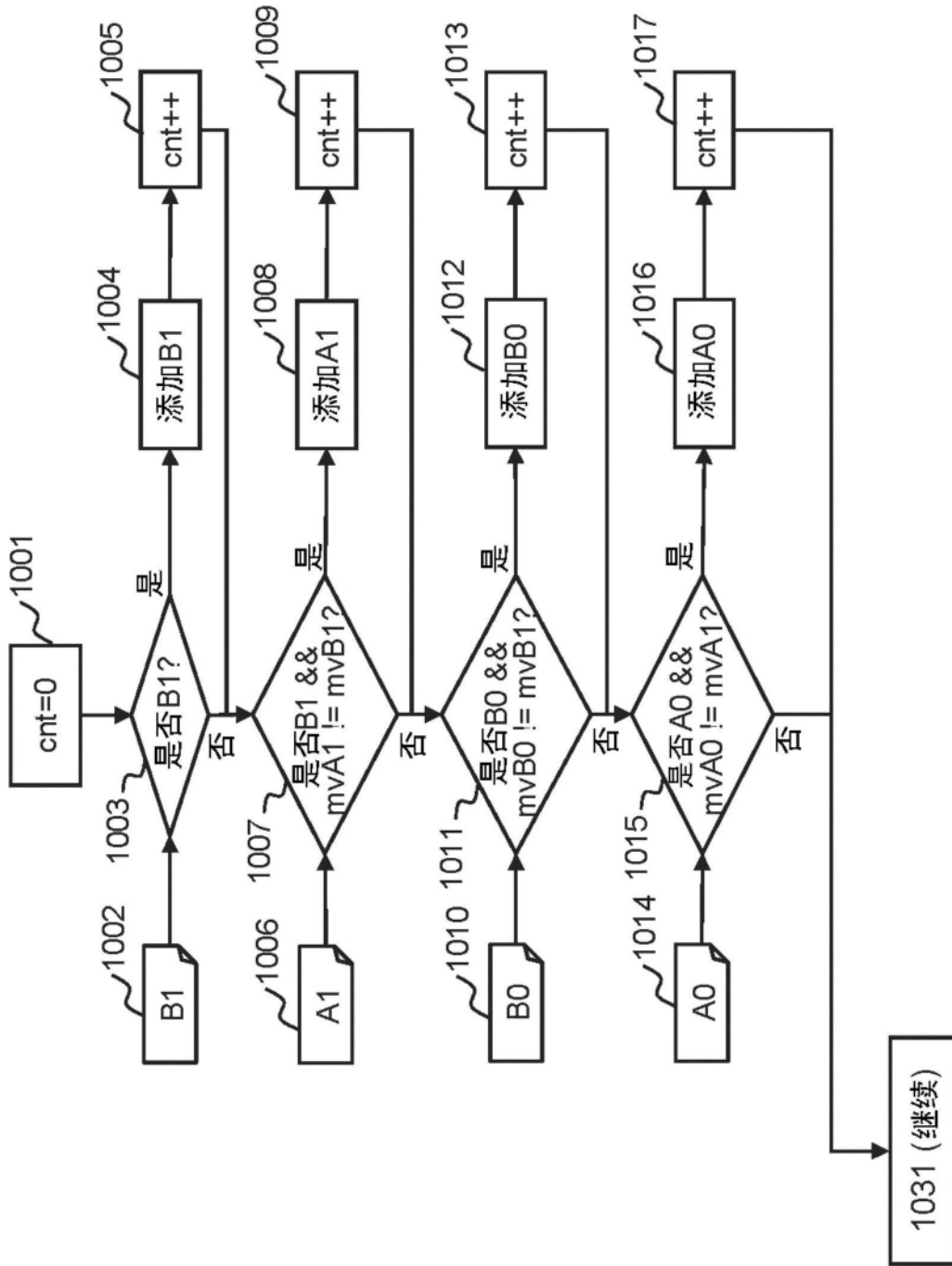


图10

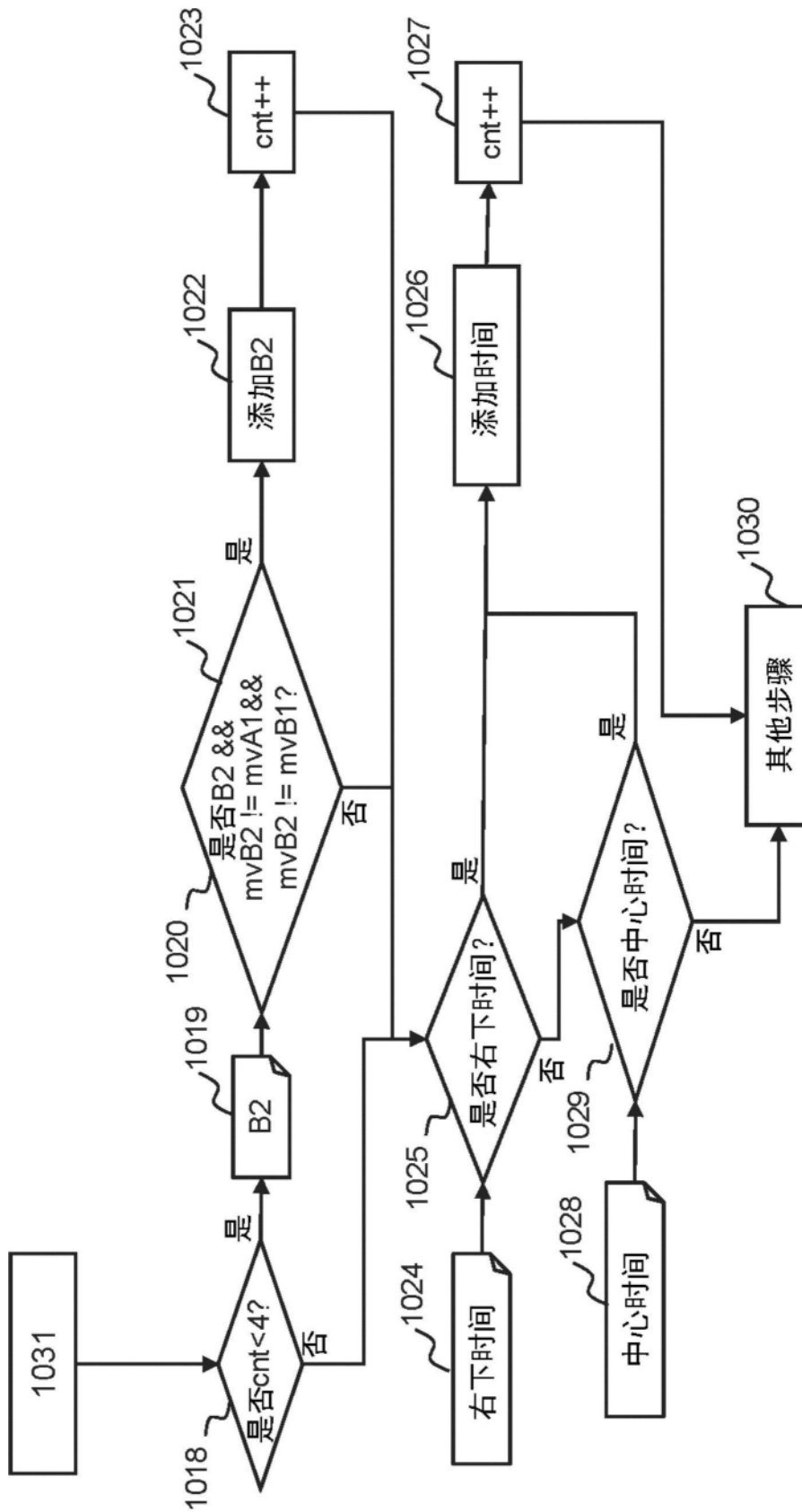


图10(续)

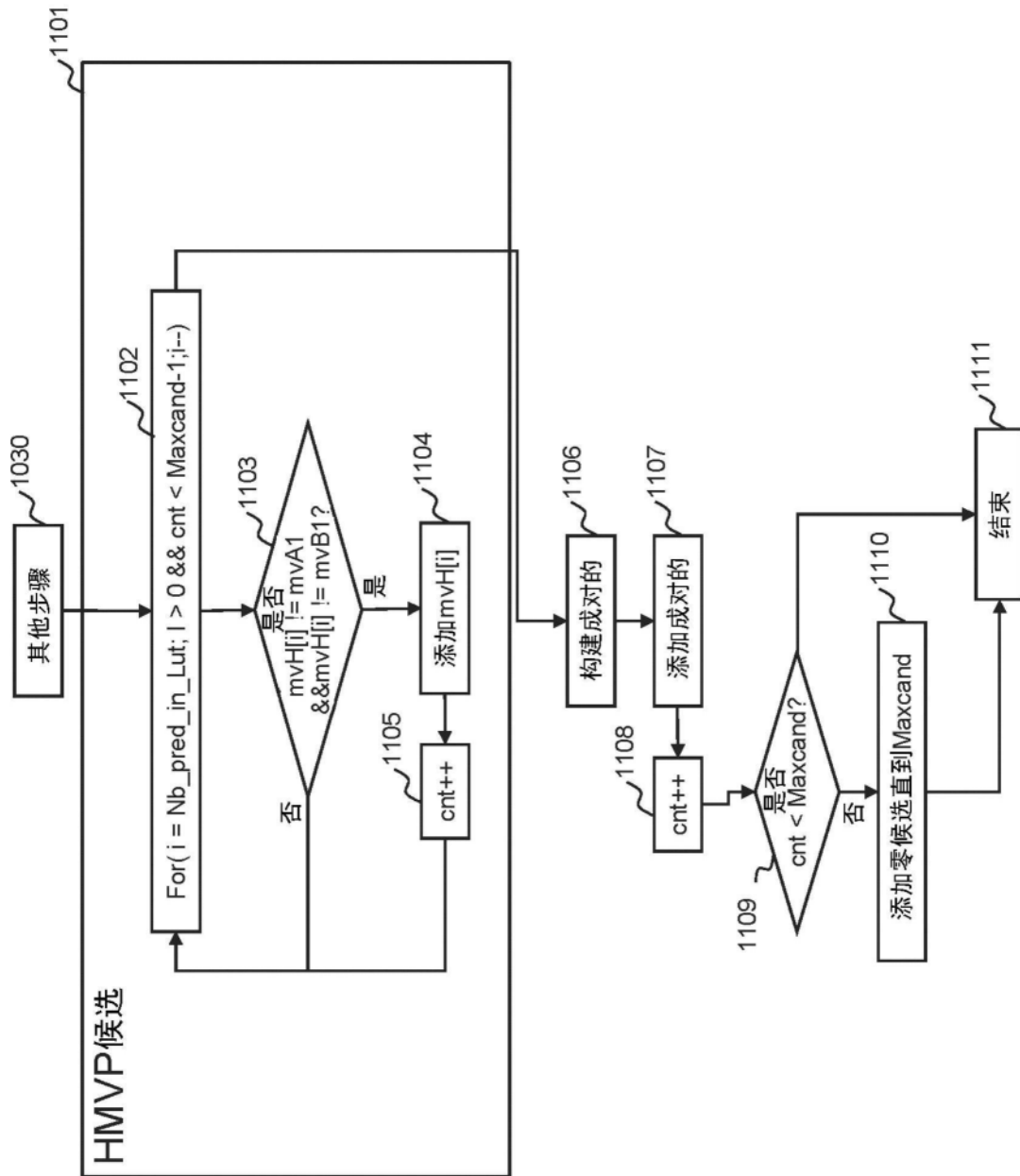


图11

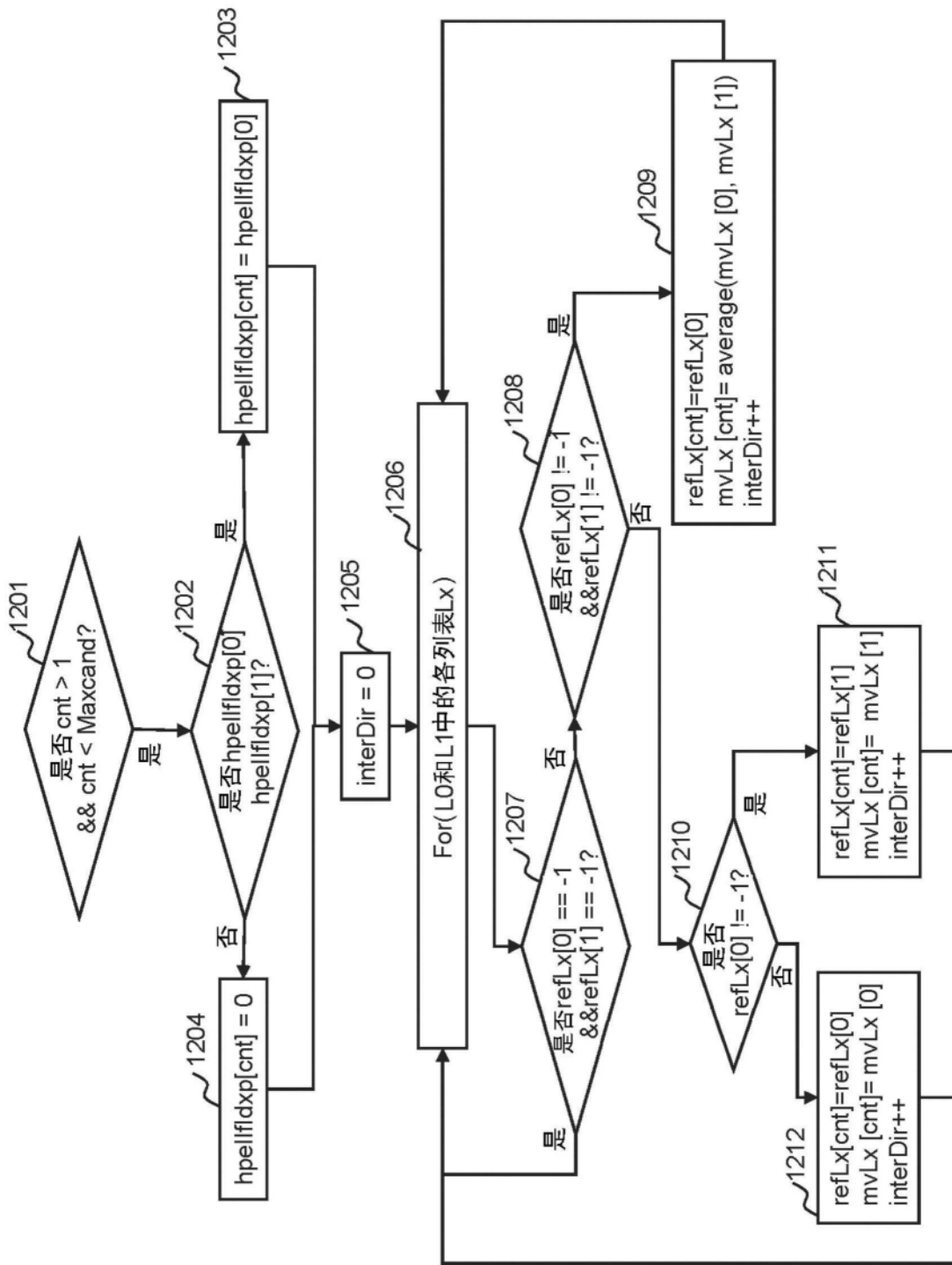


图12

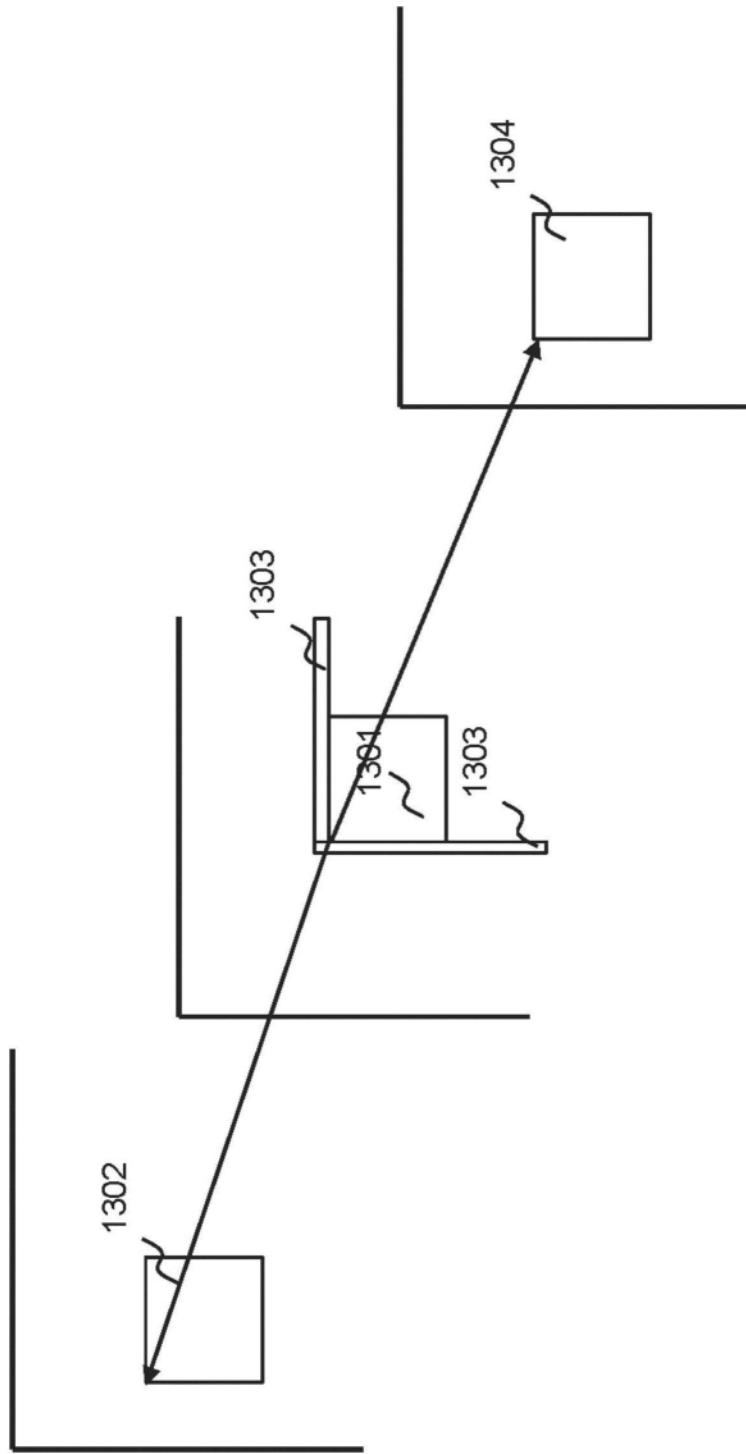


图13

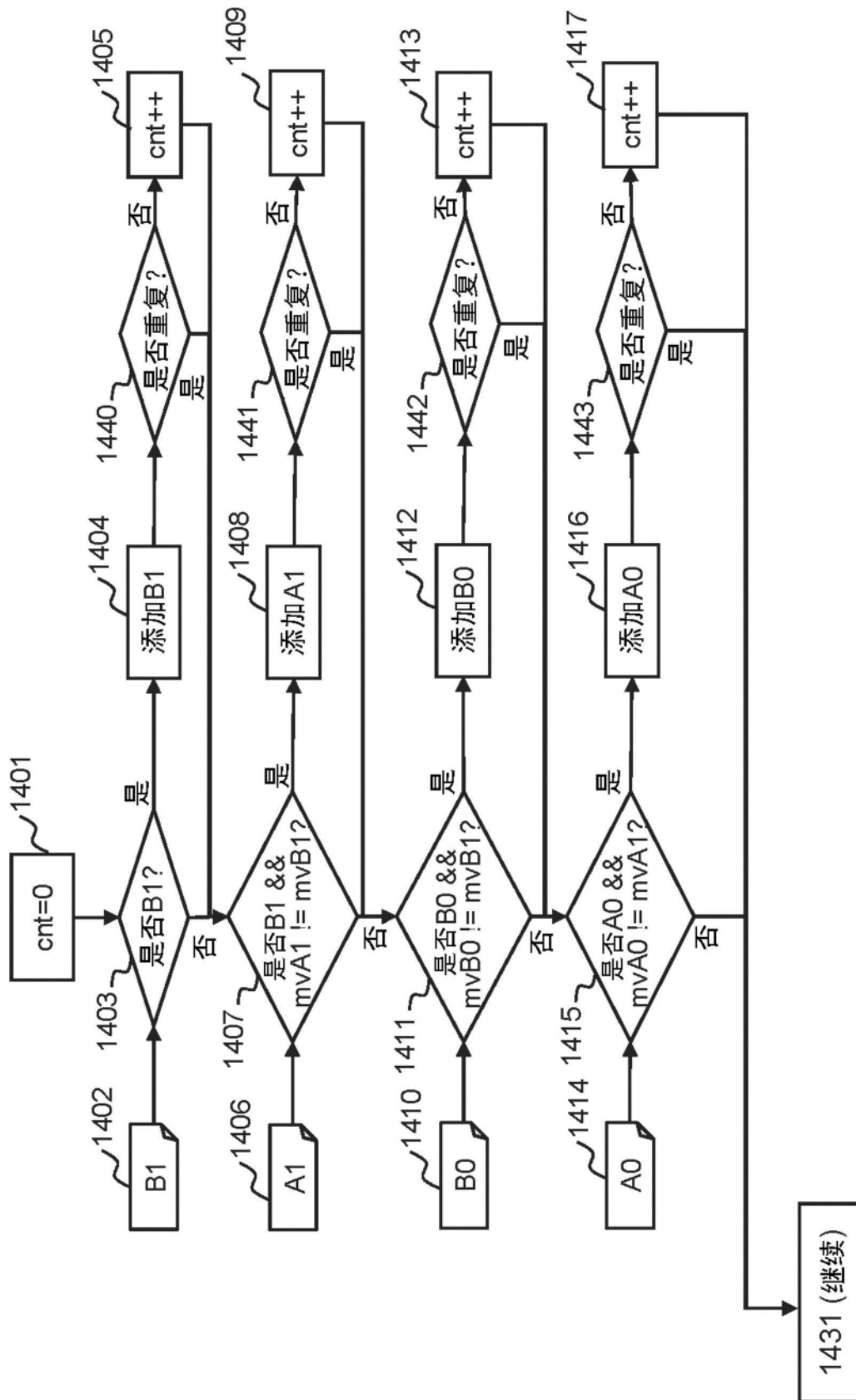


图14

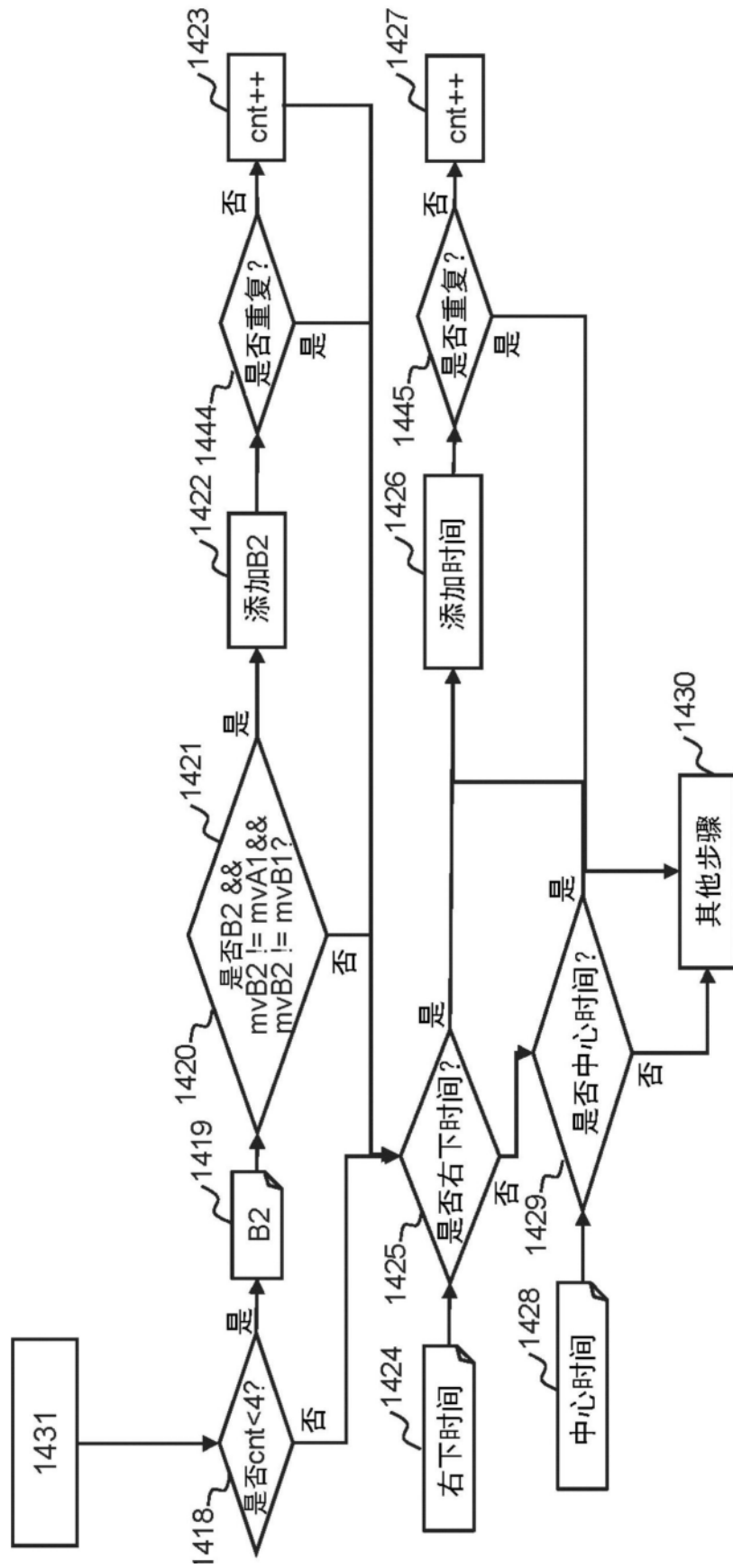


图14(续)

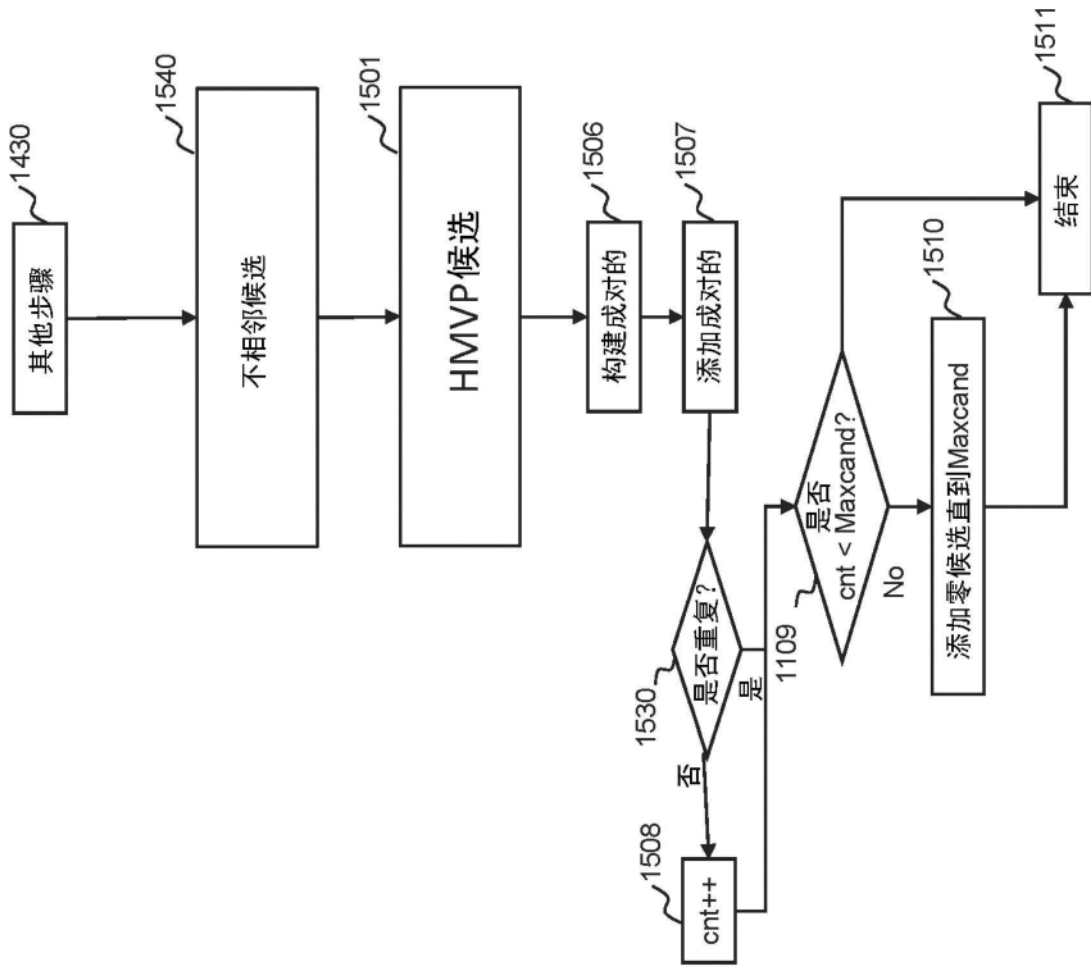


图15

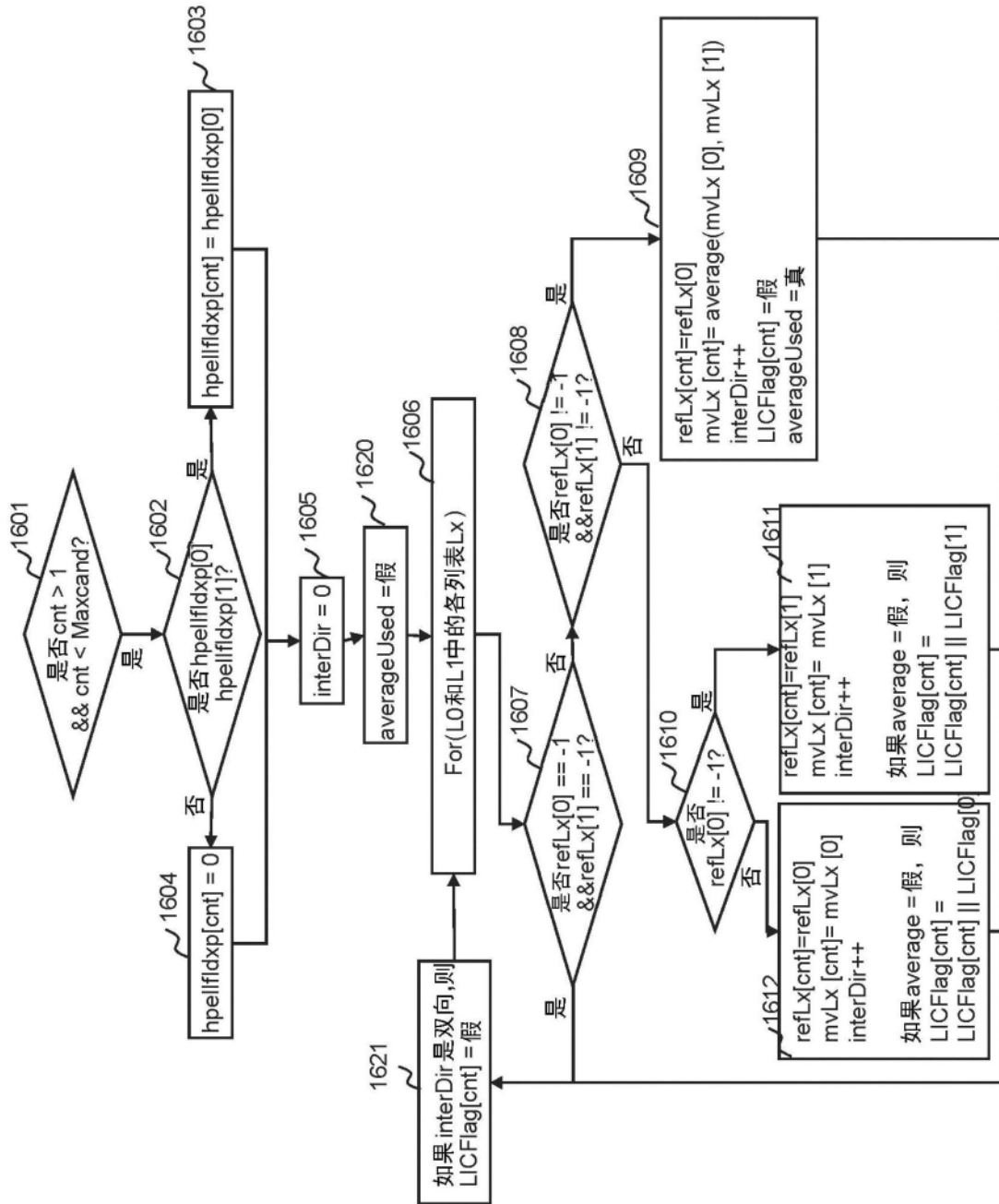


图16

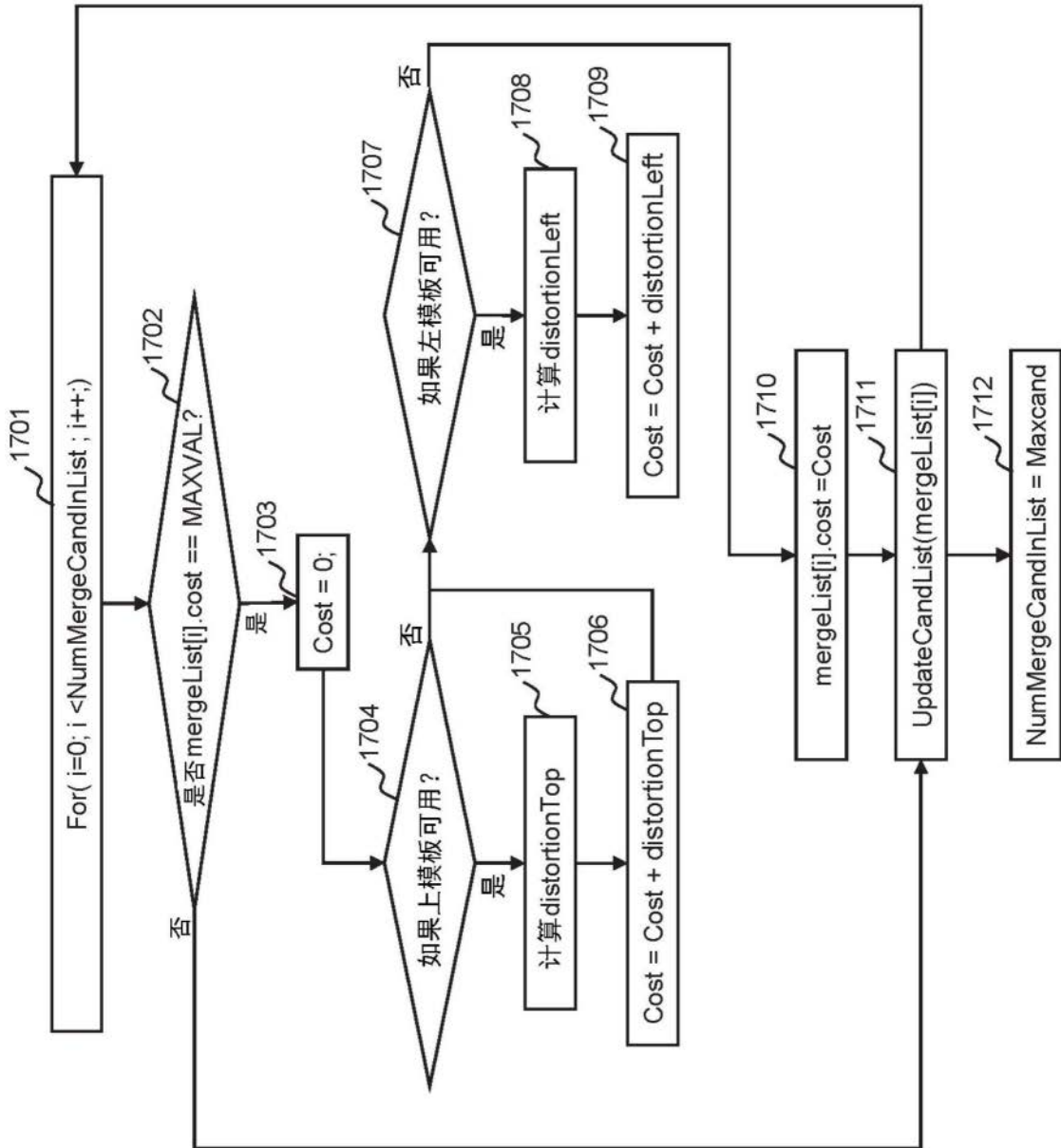


图17

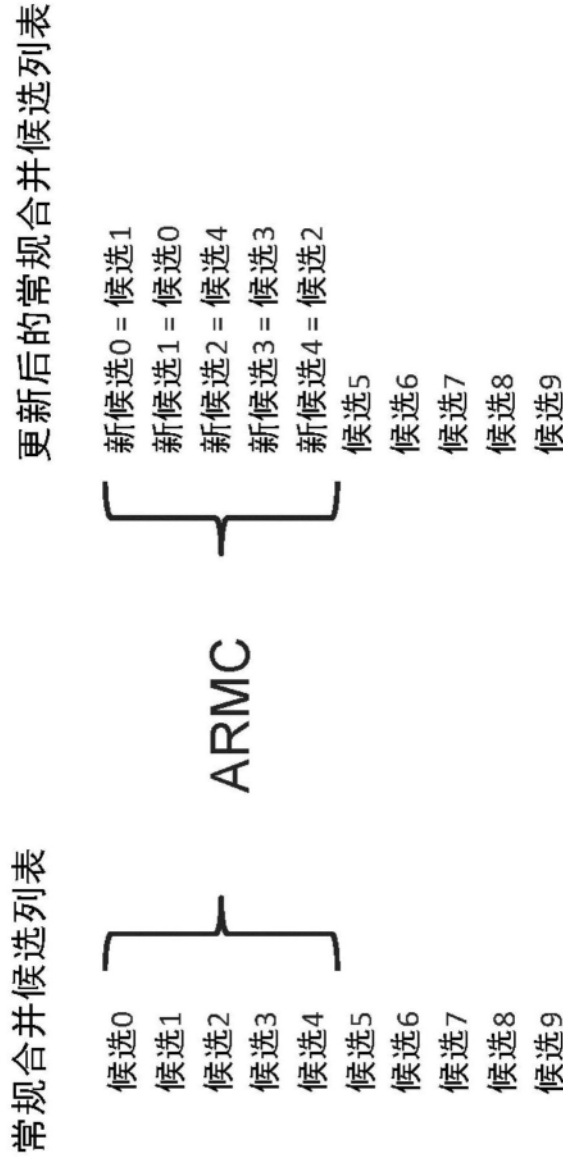


图18

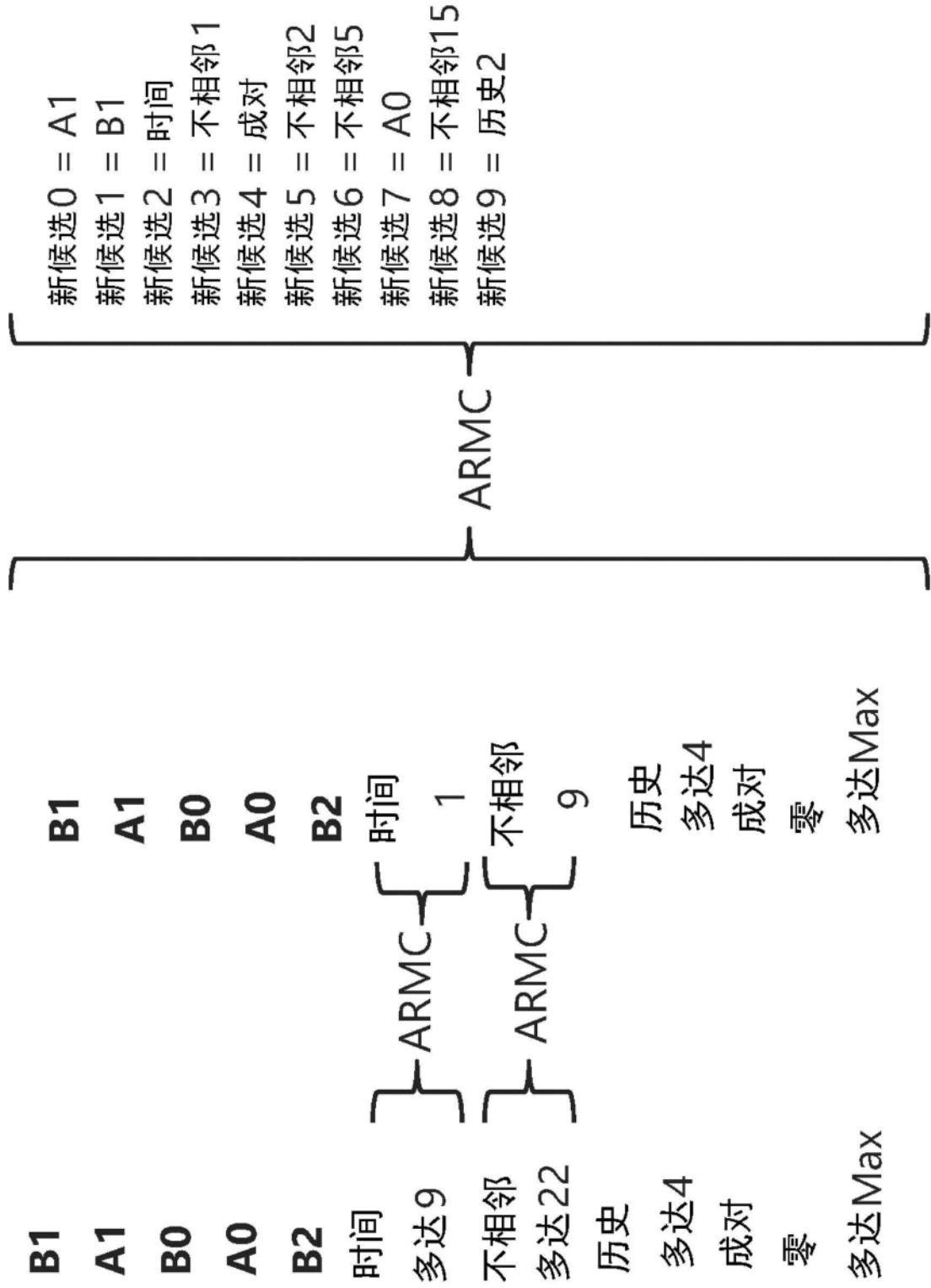


图19

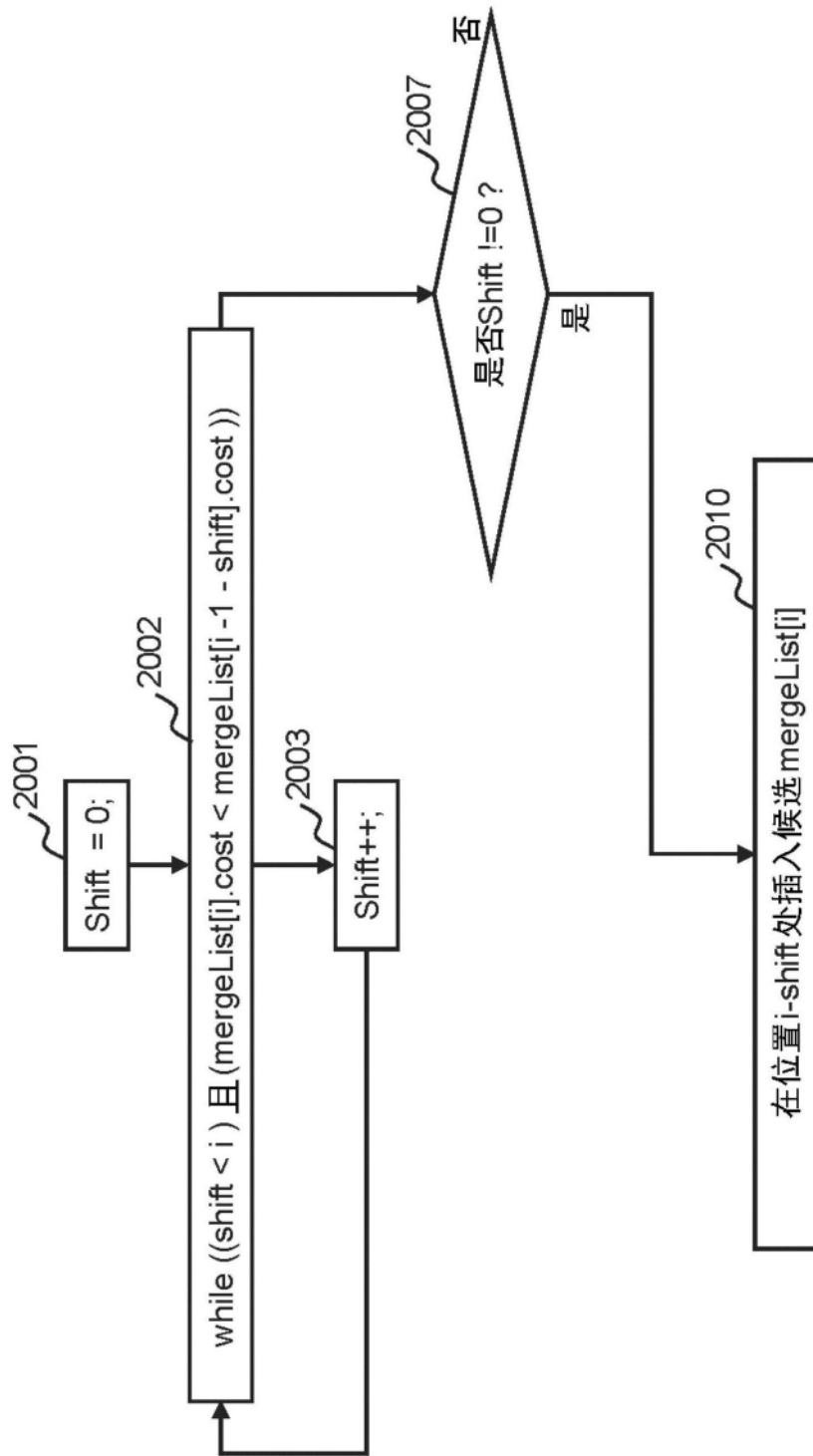


图20

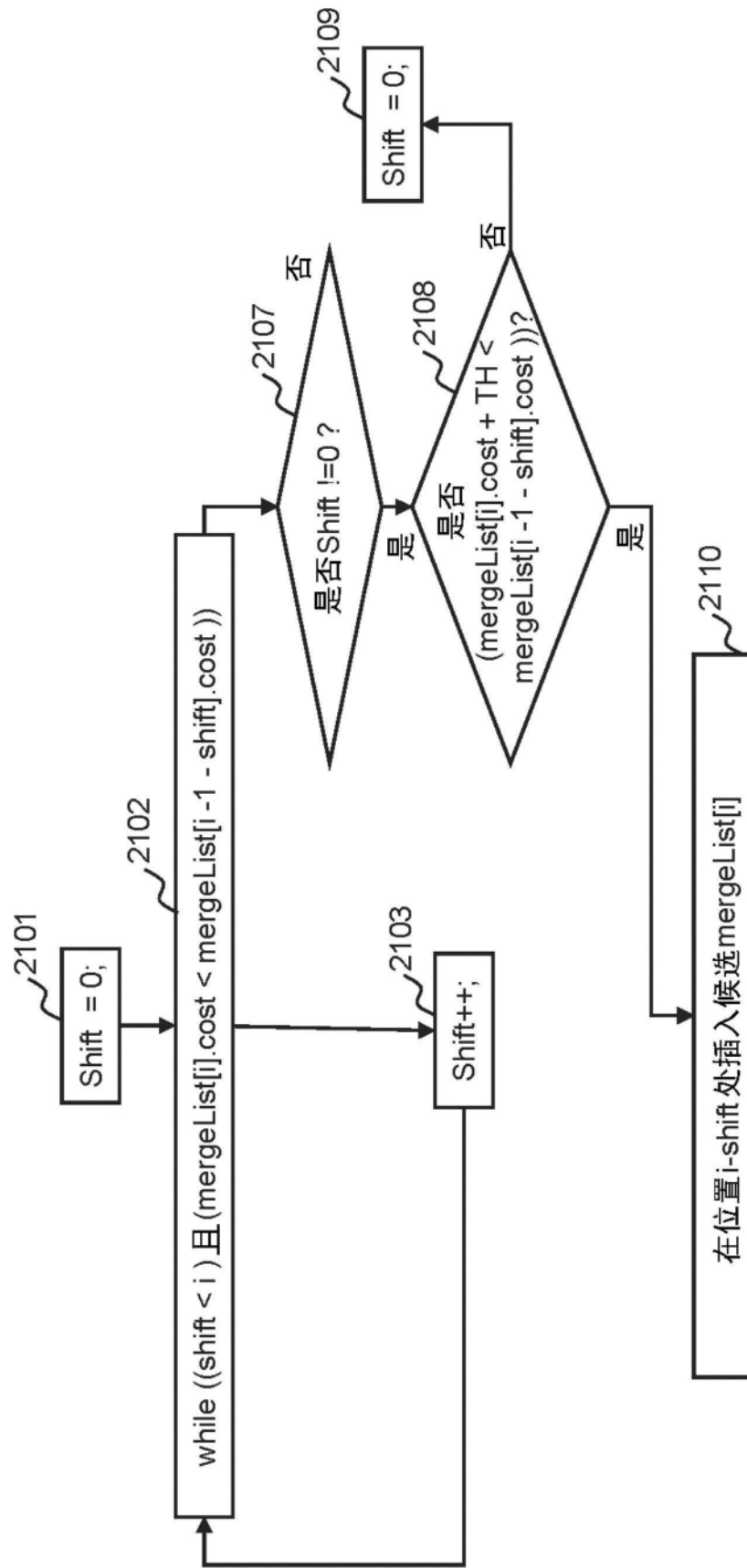


图21

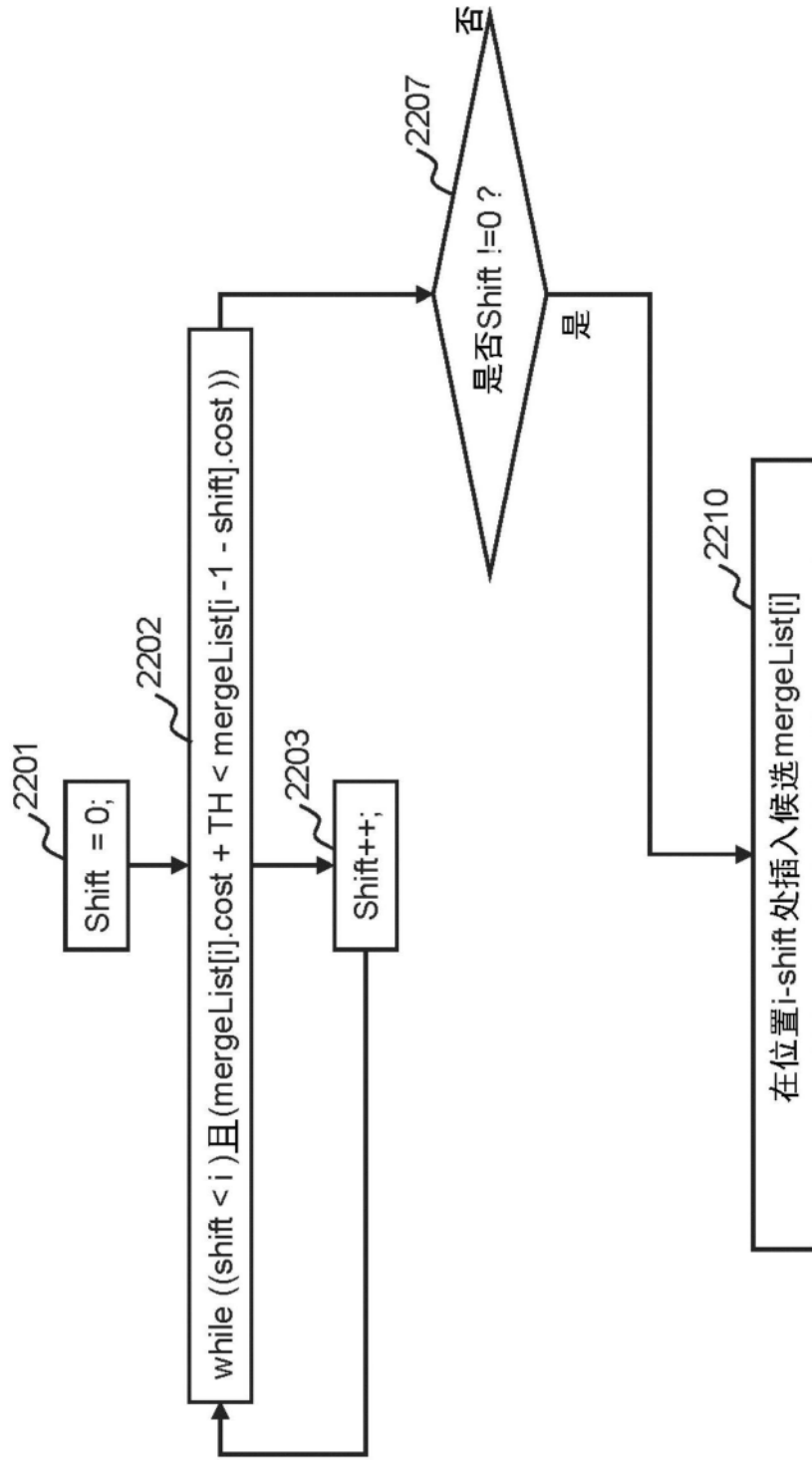


图22

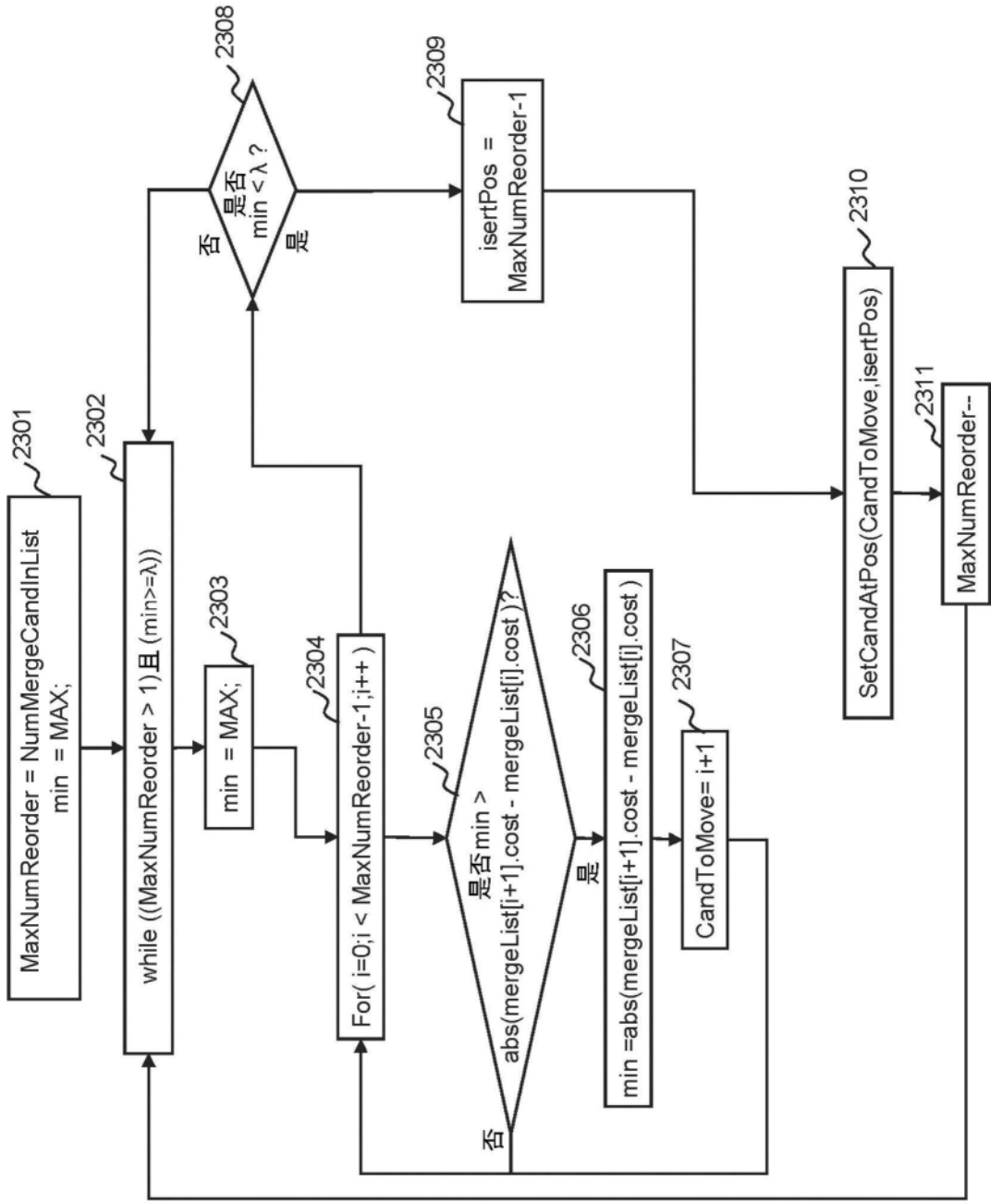


图23

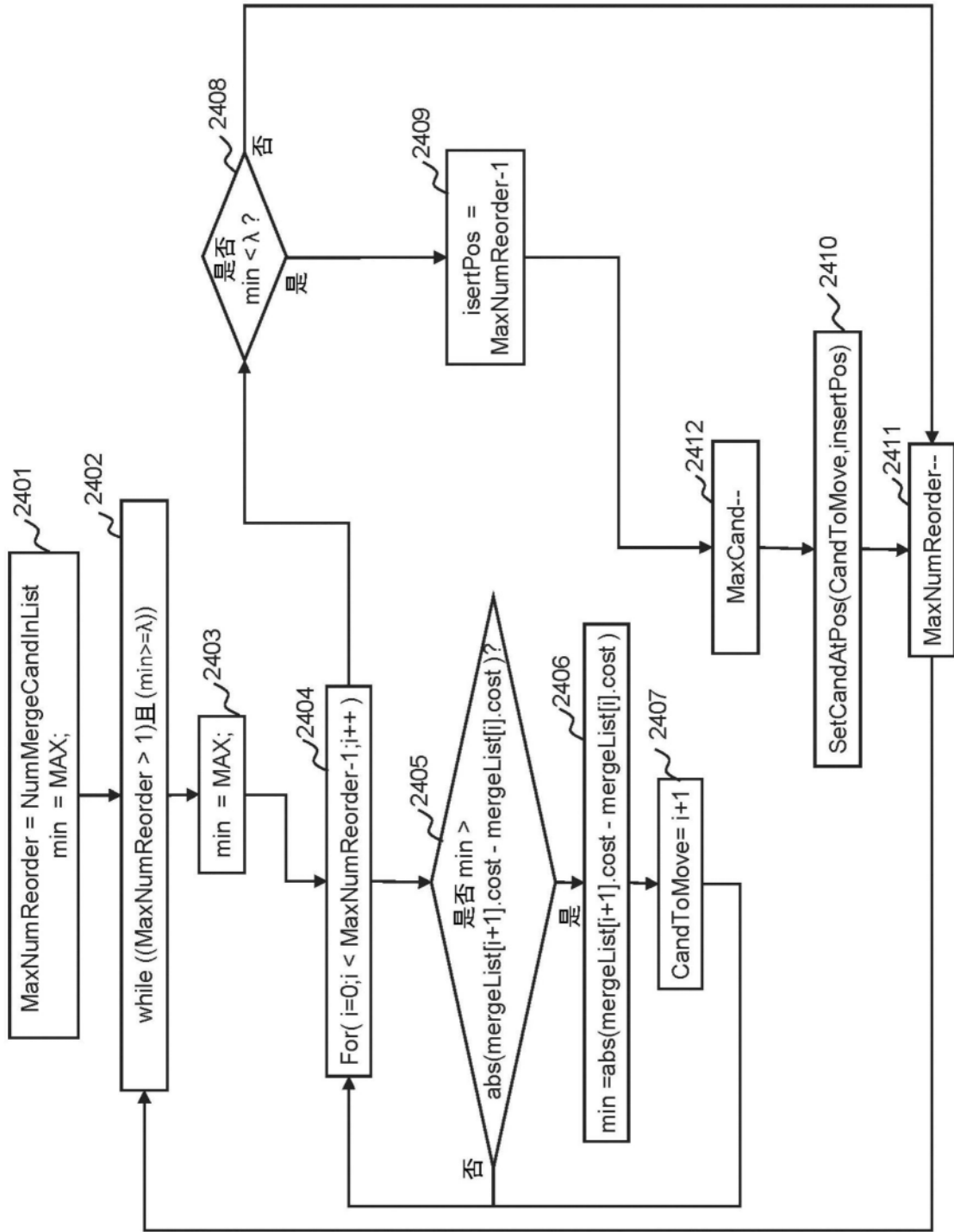


图24

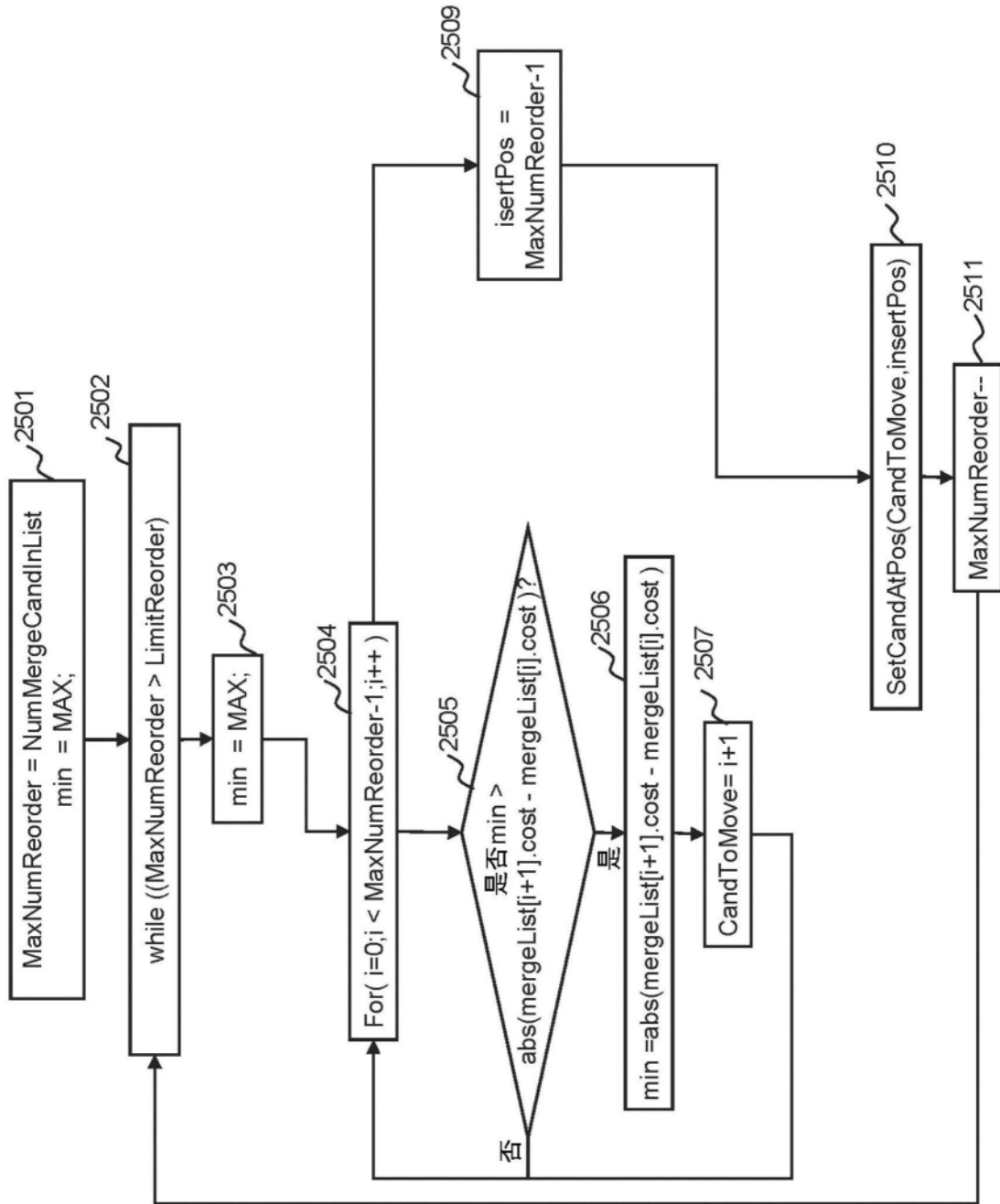


图25

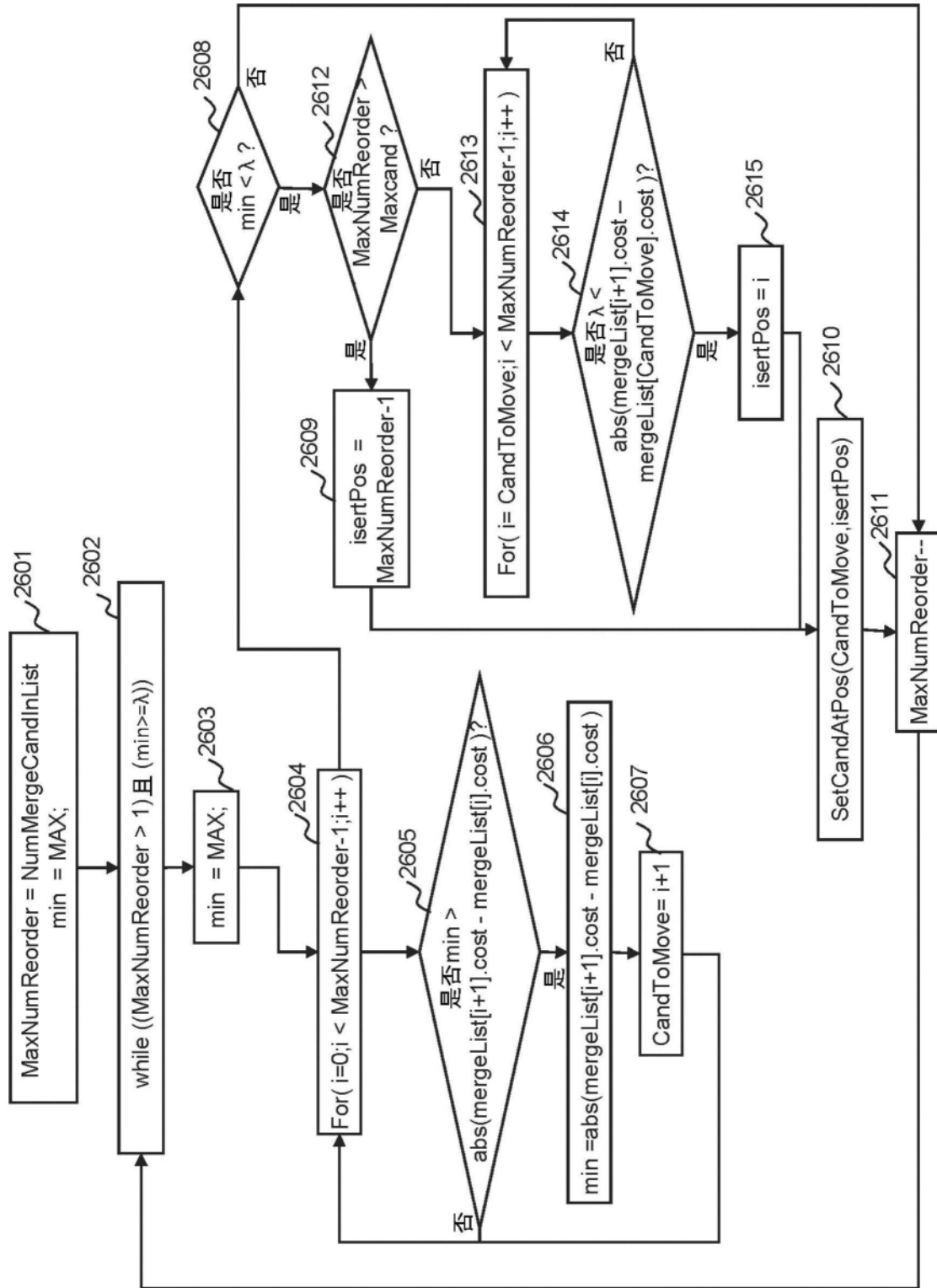


图26

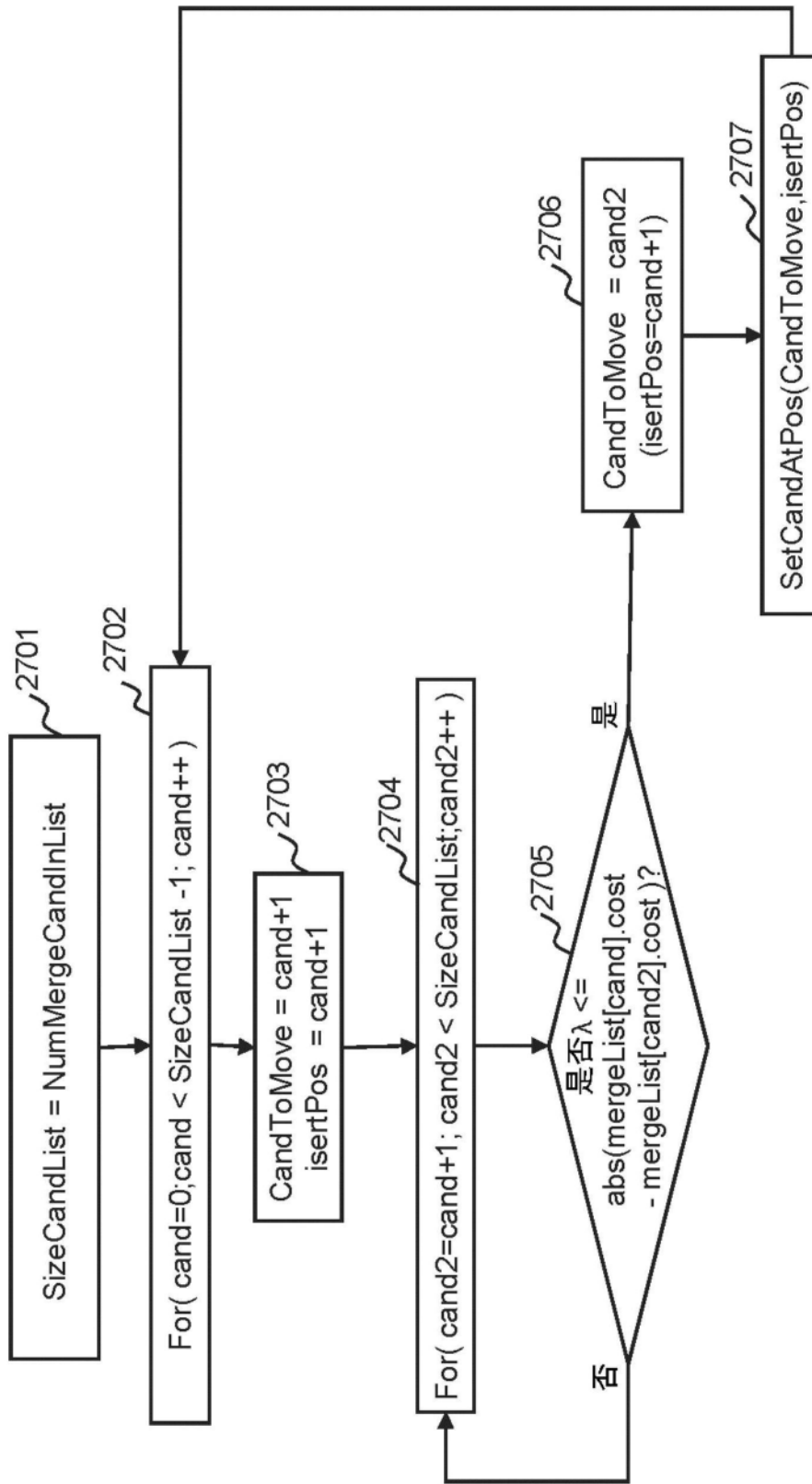


图27

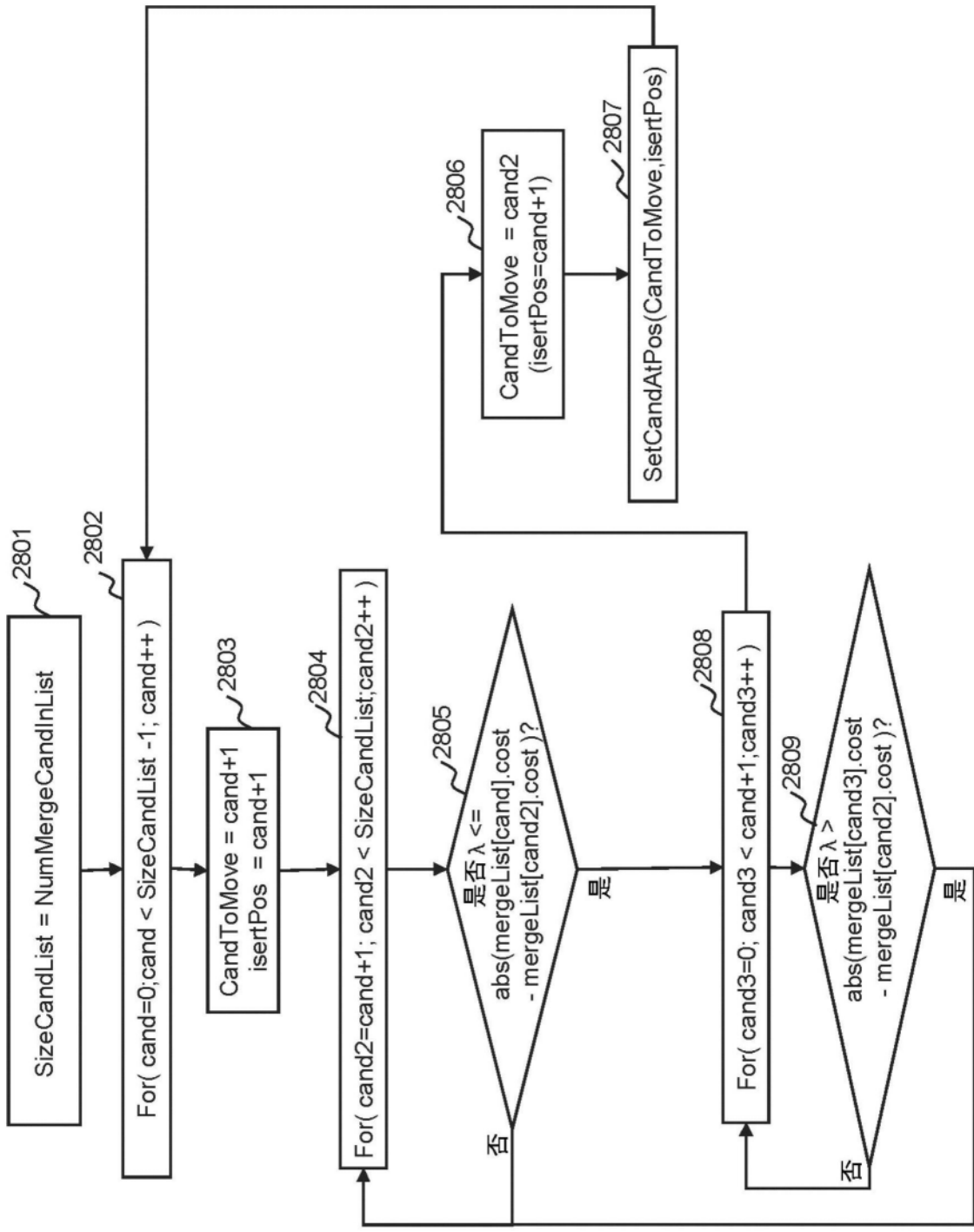


图28

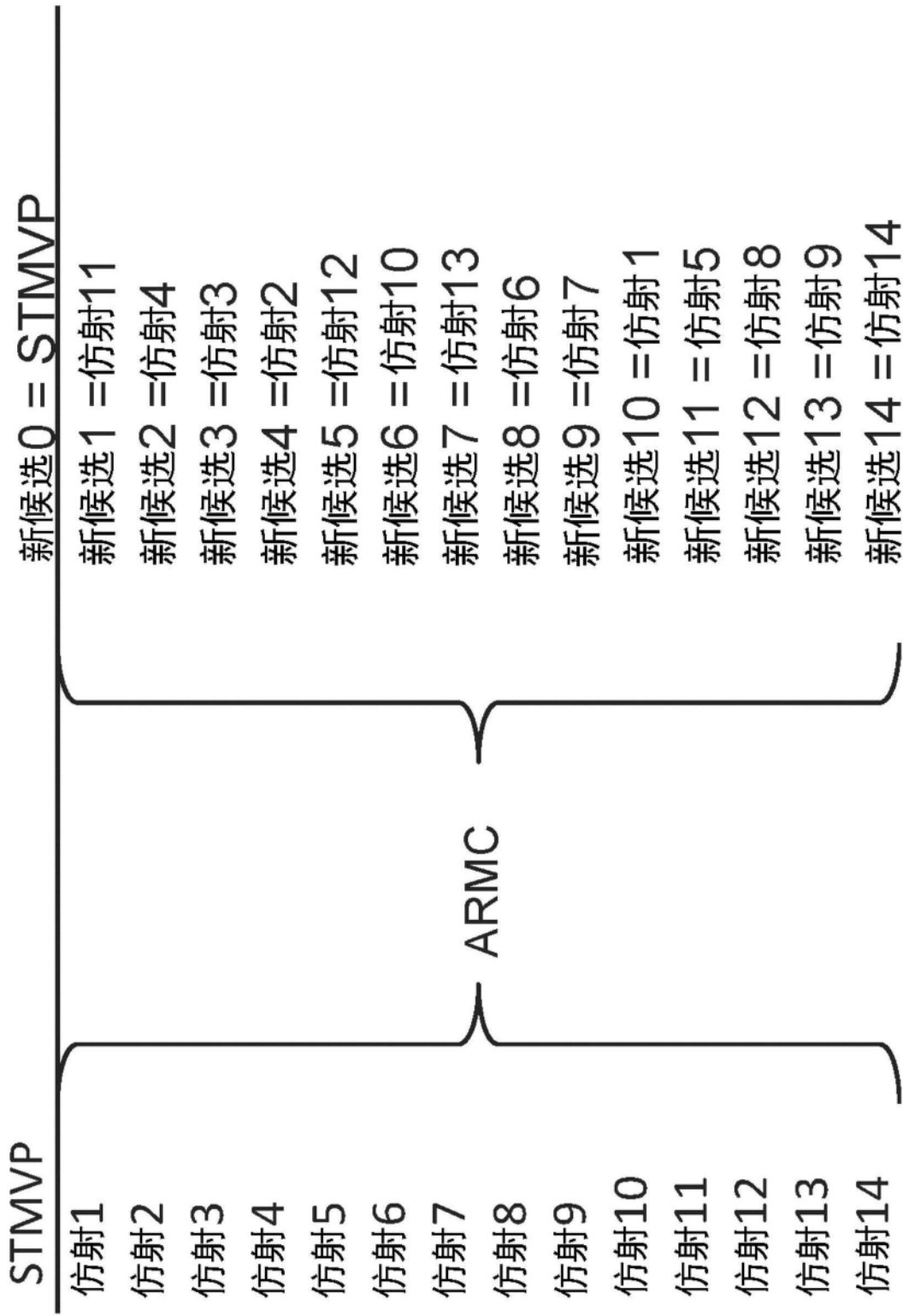


图30

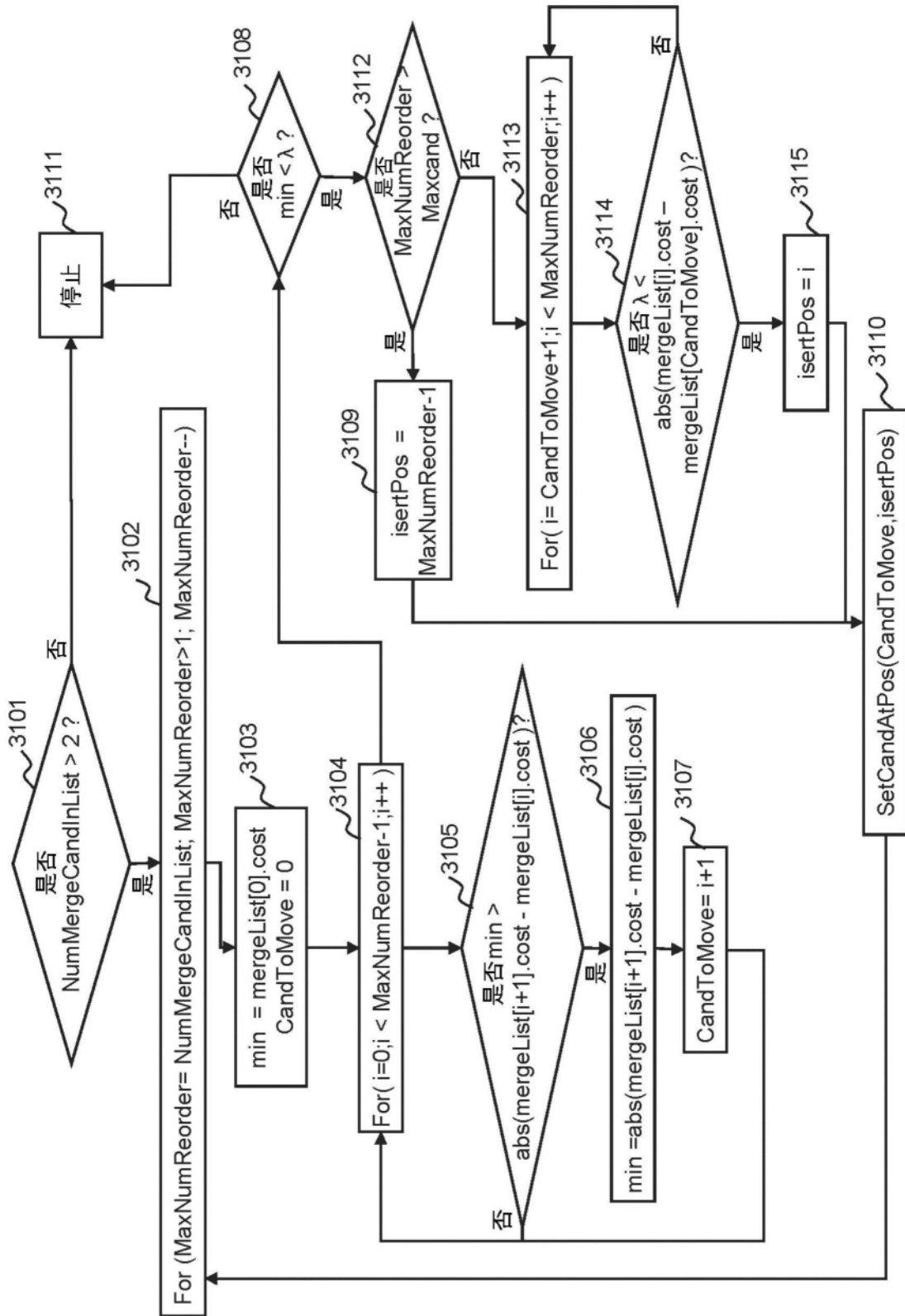


图31

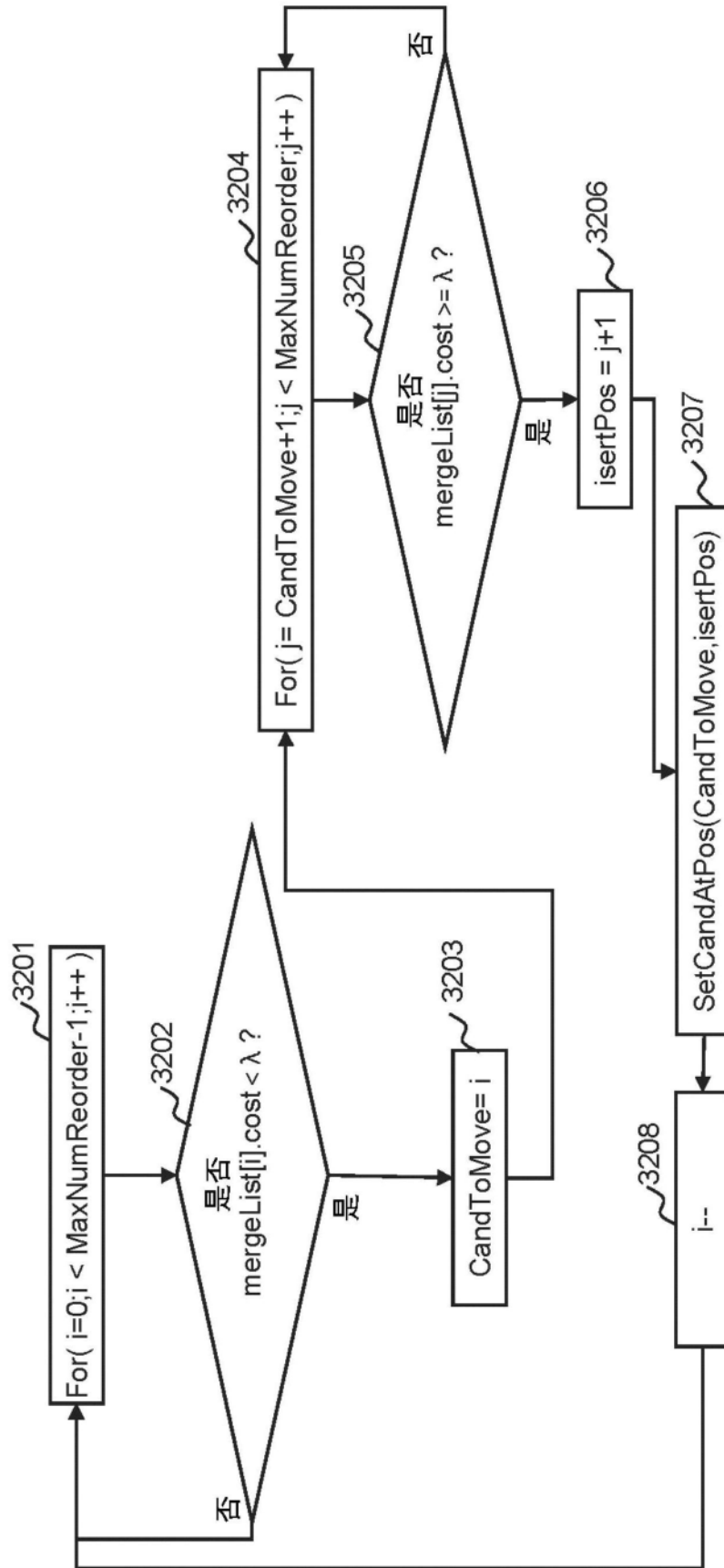


图32

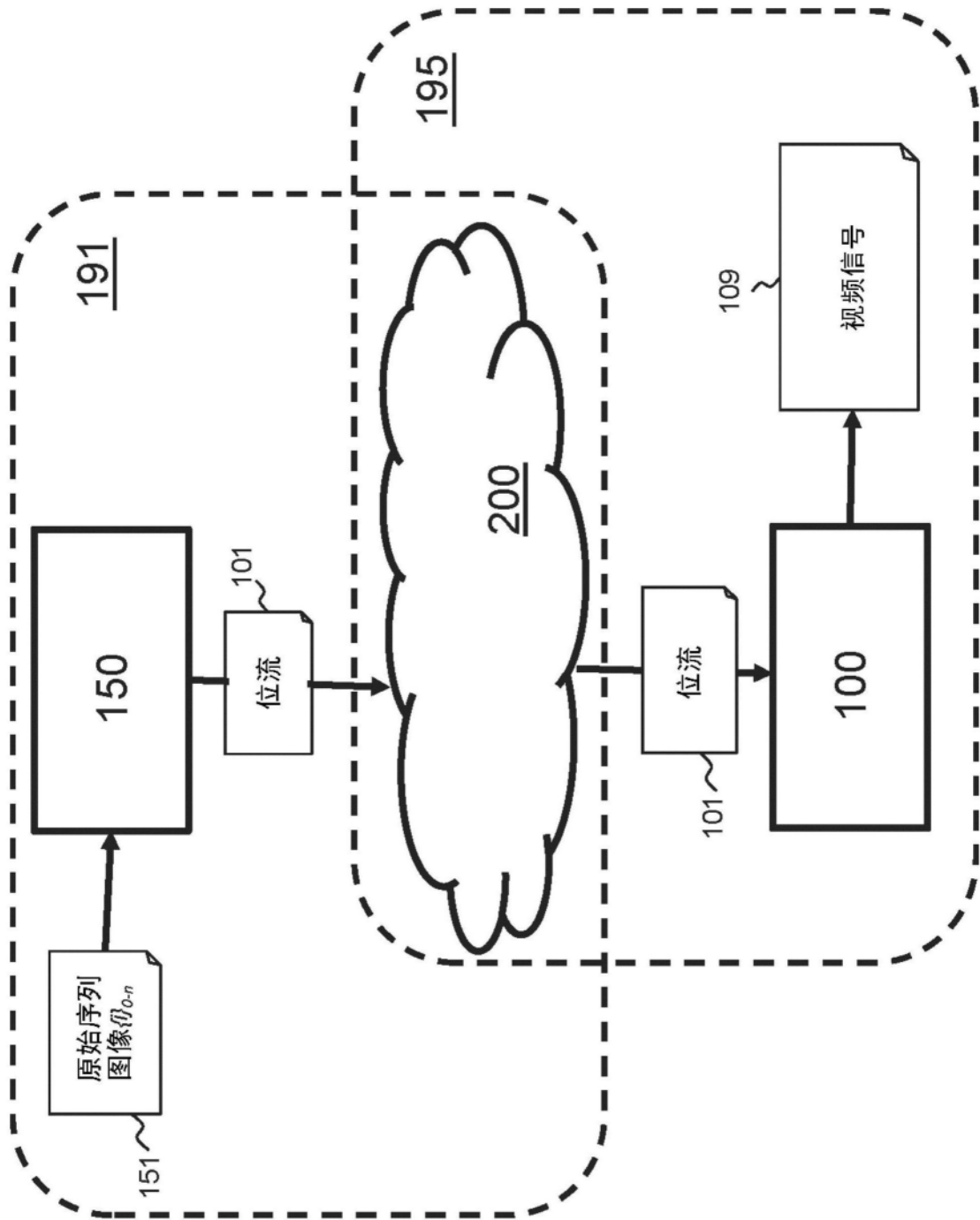


图33

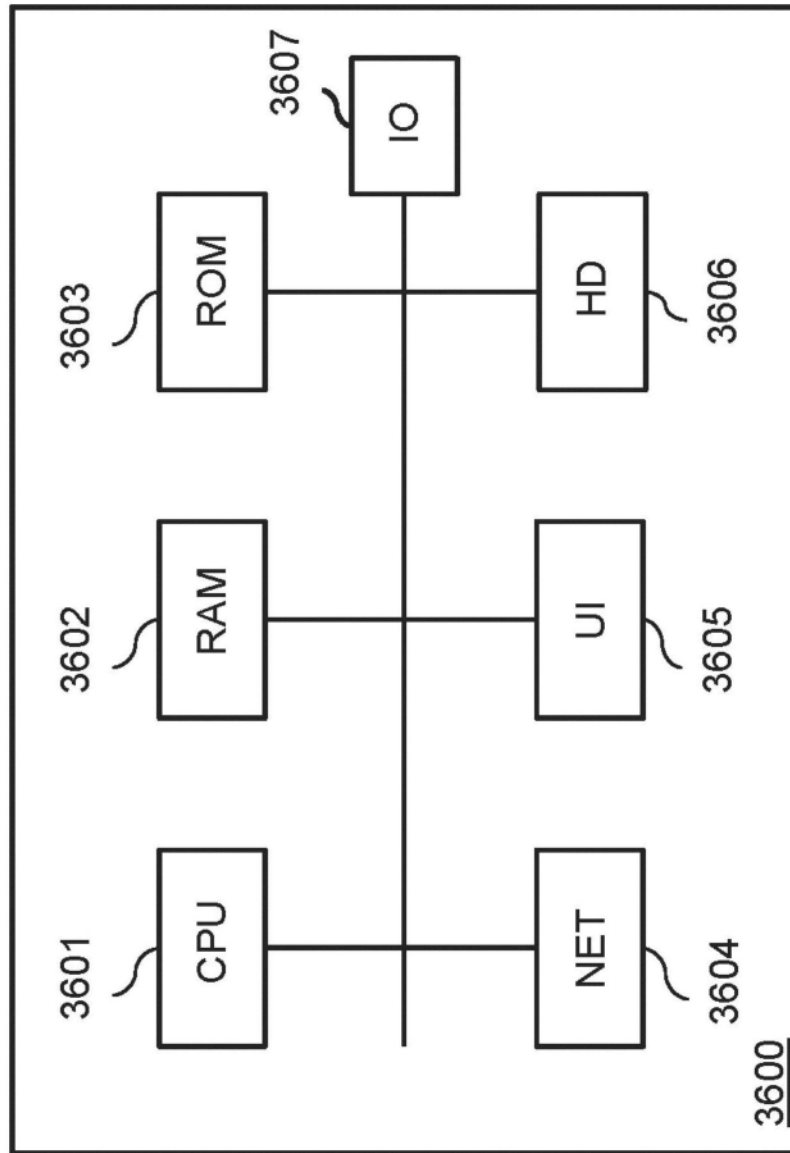


图34

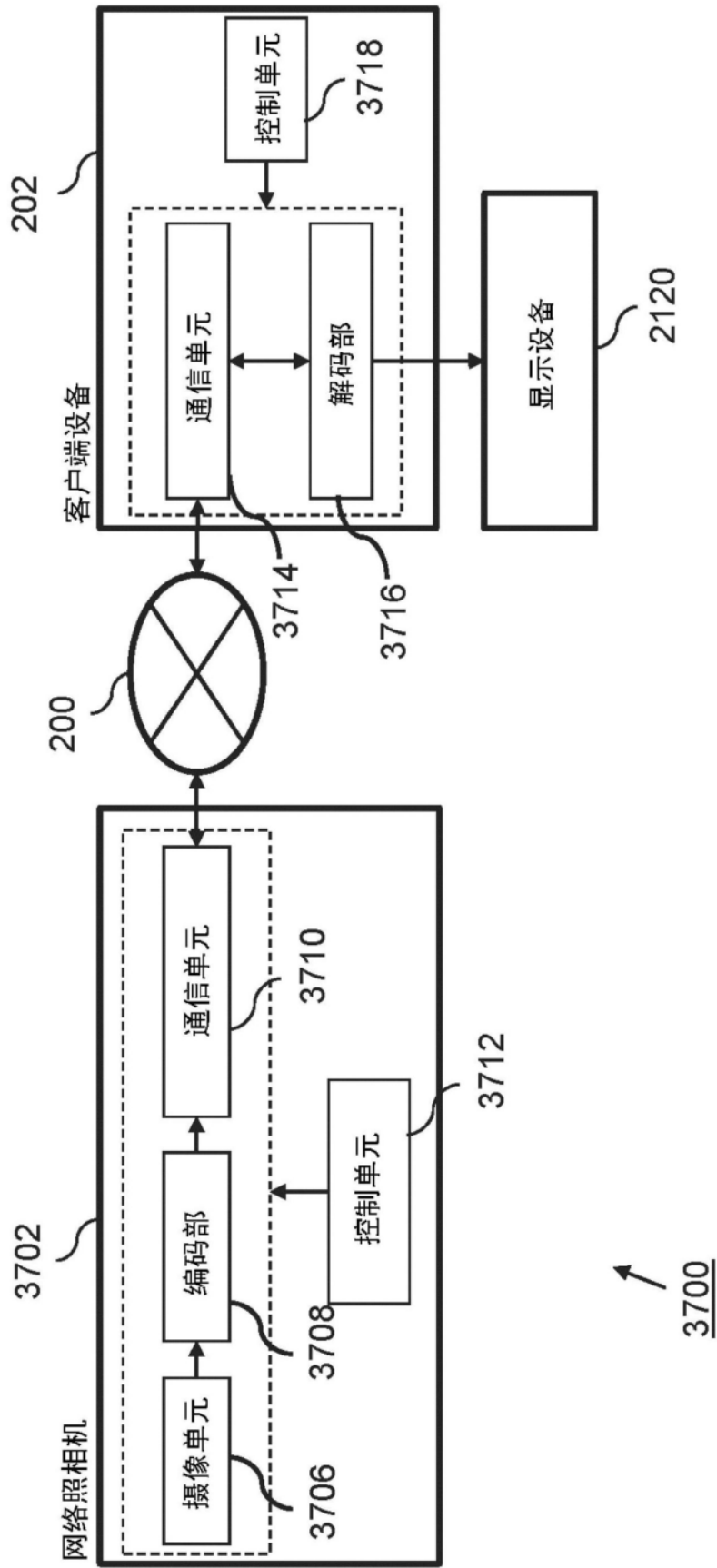


图35

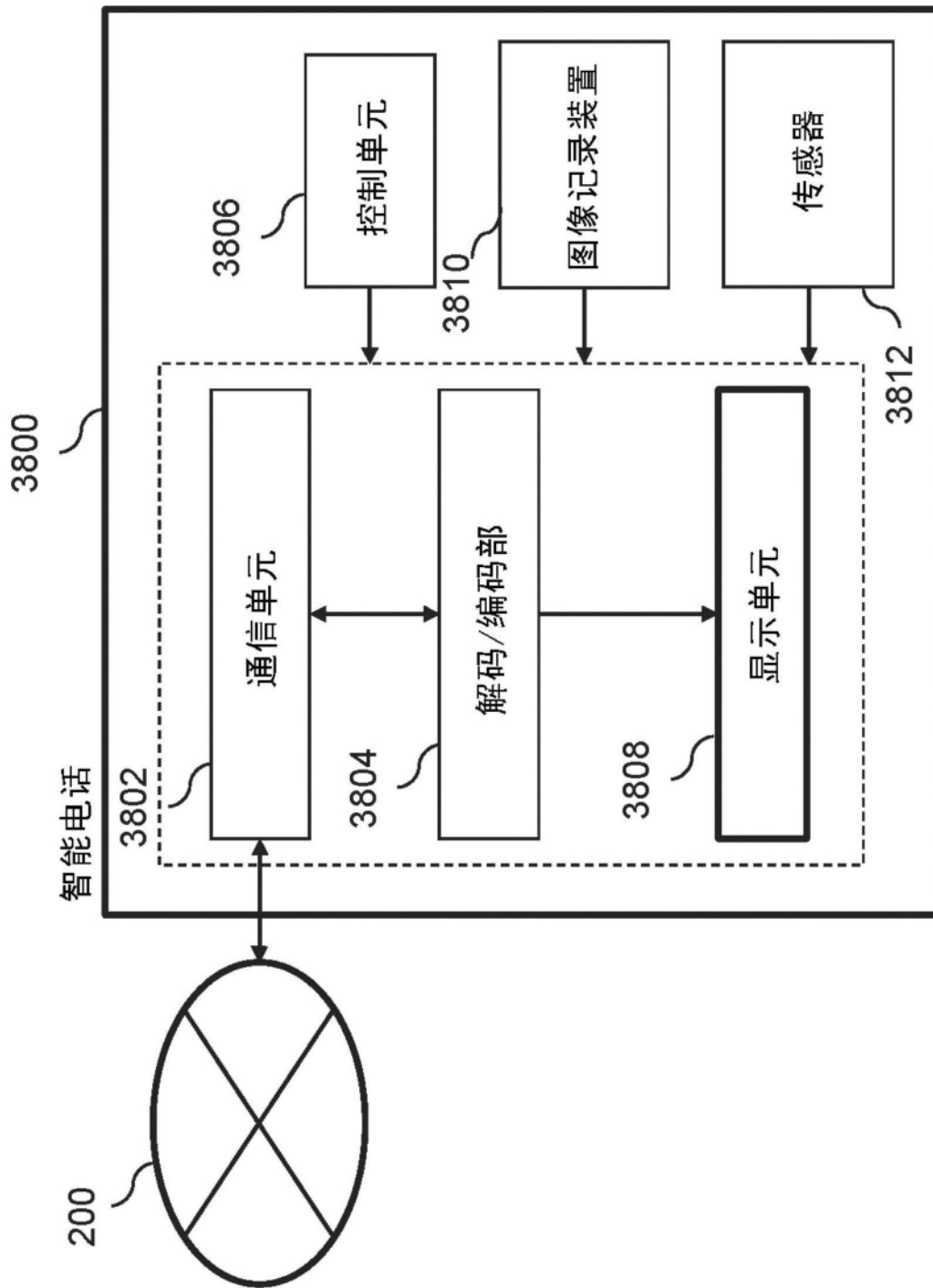


图36