



(12) 发明专利

(10) 授权公告号 CN 102217315 B

(45) 授权公告日 2016. 03. 09

(21) 申请号 200980145004. 6

(22) 申请日 2009. 11. 10

(30) 优先权数据

61/199, 028 2008. 11. 12 US

(85) PCT国际申请进入国家阶段日

2011. 05. 12

(86) PCT国际申请的申请数据

PCT/US2009/006056 2009. 11. 10

(87) PCT国际申请的公布数据

W02010/056310 EN 2010. 05. 20

(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 杨华

(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 11/04(2006. 01)

(56) 对比文件

US 7023924 B1, 2006. 04. 04,

US 2008192830 A1, 2008. 08. 14,

US 2006193388 A1, 2006. 08. 31,

US 2008175439 A1, 2008. 07. 24,

审查员 裴暑云

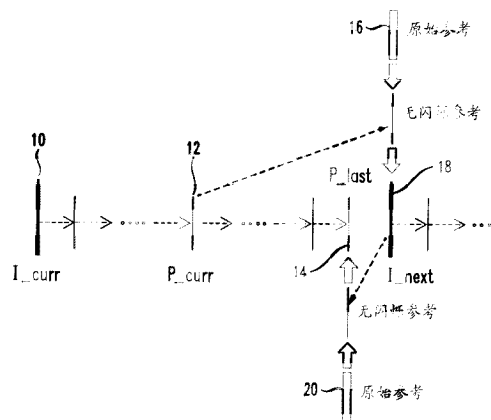
权利要求书1页 说明书8页 附图7页

(54) 发明名称

用于画面组的并行多线程视频编码的 I 帧去闪烁

(57) 摘要

提出了一种编码视频的方法,其中形成并在并行线程中编码多个画面组(GOP)。每个编码GOP具有初始的 I 帧以及接下来的一系列 P 帧。每个 I 帧被利用从之前 GOP 的最接近的编码帧导出的第一无闪烁参考来进行去闪烁编码,而该之前 GOP 的一系列 P 帧中的最后 P 帧被利用从去闪烁编码的 I 帧导出的第二无闪烁参考来进行去闪烁编码。



1. 一种编码视频的方法,包括以下步骤:  
形成多个画面组 (GOP);  
开始 GOP 的并行线程的多个编码,每个 GOP 具有初始的 I 帧,其后跟随着一系列仅有的 P 帧;  
如果在将要对之前 GOP 的最后 P 帧进行编码之前对接下来 GOP 的第一 I 帧进行编码,则利用接下来 GOP 的第一编码 I 帧来对该之前 GOP 的一系列 P 帧中的最后 P 帧进行去闪烁编码;以及,  
如果在将要对之前 GOP 的最后 P 帧进行编码之前对接下来 GOP 的第一 I 帧不进行编码,则在没有无闪烁参考的情况下对该之前 GOP 的一系列 P 帧中的最后 P 帧进行编码。
2. 如权利要求 1 所述的方法,其中,在编码 I 帧时采用小量化参数 (QP) 以便非常接近无闪烁参考。
3. 如权利要求 2 所述的方法,其中,在编码最后的 P 帧时采用中等的 QP。
4. 如权利要求 1 所述的方法,其中,P 帧去闪烁编码包括以下步骤:对于在第一 I 帧和之前 GOP 的最接近的编码帧之间的低相关性,应用较大的运动搜索范围。
5. 如权利要求 1 所述的方法,其中,该 P 帧去闪烁编码包括以下步骤:对于在第一 I 帧和之前 GOP 的最接近的编码帧之间的高相关性,应用较小的运动搜索范围。
6. 如权利要求 1 所述的方法,其中,P 帧去闪烁编码包括在模式选择中放弃跳跃模式检查。
7. 如权利要求 1 所述的方法,其中,P 帧去闪烁编码包括以下步骤:仅检查 P 16x 16 模式,使用较小的运动搜索范围,编码当前帧 MB 和预测参考 MB 之间的失真匹配,以及在 RDO-MS 中修改 RD 成本,由此防止或阻止跳跃模式和帧内模式。
8. 如权利要求 4 所述的方法,其中,通过总的帧间复杂度来确定相关性。
9. 如权利要求 5 所述的方法,其中,通过总的帧间复杂度来确定相关性。

## 用于画面组的并行多线程视频编码的 I 帧去闪烁

### 技术领域

[0001] 本发明涉及视频编码,并且更具体地涉及视频被编码为画面组(GOP)情况下的 I 帧闪烁伪像消除。

### 背景技术

[0002] 在播放 GOP 编码视频时,对于同一场景中的 GOP 而言,恼人的跳动(annoying pulsing)或者所谓的闪烁伪像通常将在周期性的 I 帧处看到。尤其对于低比特率或中等比特率视频编码,非常明显地看到该 I 帧闪烁,这大大损害了编码视频的整体感知质量。

[0003] 原始的视频信号具有原本平滑的光流。然而,在质量差的视频编码之后,在编码视频信号中原来的光流将失真。然后,作为结果的、在编码帧上的时间不一致性/不连贯性将被感知为闪烁伪像。实际上,更经常在编码视频的静止的或低运动区域/部分处感知到闪烁。例如,几个连续帧可能共享相同的静止背景。因此,在这些帧上在静止背景中所有的位于同一位置的像素在原始输入视频中具有相同或相似的像素值。然而,在视频编码中,位于同一位置的像素可能从不同帧中的不同参考像素预测而来,并且因此在对残差进行量化之后,产生不同的重构值。在视觉上,在编码视频播放期间,在这些帧上增加的帧间差异将被感知为闪烁。

[0004] 这样,由于粗糙量化,对于低比特率编码或中等比特率编码而言,闪烁伪像更密集。而且,在 I 帧上比 P 帧或 B 帧上更明显地观察到闪烁伪像。这主要是因为:对于相同的静止区域,在 P 帧或 B 帧中由帧间预测产生的预测残差通常比在 I 帧中由帧内预测或无预测产生的预测残差更小。因此,利用粗糙量化,I 帧中重构的静止区域展示了更容易注意到的、与之前的 P 帧或 B 帧中位于同一位置的区域的差异,并且因此展示了更容易注意到的闪烁伪像。因此,如何减少 I 帧闪烁是大大影响整体感知视频编码质量的关键问题。

[0005] 大多数现有的基于编码器的 I 帧去闪烁机制被设计用于 GOP 顺序单线程视频编码情况,在该情况下,当编码 I 帧时,紧接在该 I 帧之前的帧已经被编码。因此,可以容易地使用重构的之前帧来导出当前帧的无闪烁参考,其然后可以被用于当前 I 帧的去闪烁。

[0006] 图 1 图示了常用的用于 GOP 顺序单线程编码的两回合(pass) I 帧去闪烁方法。在此情况下,在编码 I<sub>next</sub> 8 之前,P<sub>last</sub> 4 已经被编码,由此总可以被用来导出 I<sub>next</sub> 8 的无闪烁参考以用于其去闪烁。由于 P<sub>last</sub> 4 紧接在 I<sub>next</sub> 8 之前,因此通常这两帧是高度相关的,并且因而所导出的无闪烁参考通常很好地用于去闪烁。

[0007] 使用多编码线程而不使用一个单线程,是用于在实时视频编码系统中大大加速计算密集的视频编码处理的常用的有效并行化策略。尽管在实践中可以以许多不同的方式来采用多线程,一种直接的并且因而通常被采用的方法是使多线程分别且同时编码多个 GOP。这是用于 GOP 并行编码的情形。注意,贯穿该说明书,术语“GOP 并行”和“多线程”将被互换地使用,并且“GOP 顺序”和“单线程”将同样地被互换地使用。

[0008] 多线程编码使 I 帧闪烁消除成为比在 GOP 顺序单线程编码的情况下的 I 帧闪烁消除更具挑战的任务。在单线程编码中,当编码 I 帧时,紧接在其之前的帧已经被编码,可以

容易地采用该之前帧的重构来导出用于当前 I 帧的去闪烁编码的良好的无闪烁参考（例如，经由用于第一编码回合的穷尽性（exhaustive）或简化 P 帧编码）。然而，在 GOP 并行多线程编码情况下，最有可能的是：当编码 I 帧时，紧接在其之前的帧尚未被编码，这是因为这两帧可能属于两个不同的 GOP，而这两个不同的 GOP 被两个不同的编码线程编码。在此情况下，一种解决方案是使用最靠近当前 I 帧的之前 GOP 中的编码帧来产生其无闪烁参考以用于去闪烁。然而，如果该帧距当前帧太远使得这两帧没有很好地相关，则可能不能从该帧中导出良好的无闪烁参考，因而可能不能实现合适的闪烁消除。

[0009] 通常，通过适当地修改编码处理或者通过在解码器处增加一些有效的后处理，可能消除或减少 I 帧闪烁以及任何其它编码伪像。然而，在实际的视频编码应用中，基于后处理的去闪烁通常不是好的解决方案，这是因为编码视频比特流可能被来自大量不同制造商的解码器 / 播放器解码，所述大量不同制造商中的一些可能没有采用该特定的后处理技术（例如，为了降低产品成本）。

### 发明内容

[0010] 提出了一种编码视频的方法，其中形成并在以并行线程编码多个画面组 (GOP)。每个编码 GOP 具有一个初始的 I 帧以及接下来的一系列 P 帧。每个 I 帧被利用从之前 GOP 的最靠近的编码帧导出的第一无闪烁参考来进行去闪烁编码，而该之前 GOP 的一系列 P 帧中的最后 P 帧被利用从去闪烁编码的 I 帧导出的第二无闪烁参考来进行去闪烁编码。可以在编码 I 帧时采用小量化参数 (QP) 以便非常接近该第一无闪烁参考。可以在编码最后的 P 帧时采用中等的 QP。在该方法中，所导出的第一无闪烁参考可以由一个回合的简化 P 帧编码来生成。该简化的 P 帧编码可以包括以下步骤：对于在 I 帧和之前 GOP 的最接近的编码帧之间的低相关性，应用较大的运动搜索范围。该简化的 P 帧编码还可以包括以下步骤：对于在 I 帧和之前 GOP 的最接近的编码帧之间的高相关性，应用较小的运动搜索范围；或者包括在模式选择中放弃跳跃模式检查，其中可以通过总的帧间复杂度、或者可以通过总的帧间复杂度来确定该相关性。简化的 P 帧编码还可能包括以下步骤：仅检查 P 16x16 模式，使用较小的运动搜索范围，编码当前帧 MB 和预测参考 MB 之间的失真匹配，以及在 RDO-MS 中修改 RD 成本，由此防止或阻止跳跃模式和帧内模式。

### 附图说明

[0011] 现在将参考附图借助示例来描述本发明，在附图中：

[0012] 图 1 是现有的用于 GOP 顺序单线程编码的两回合 I 帧去闪烁方法的示意图；

[0013] 图 2 是根据本发明的用于 GOP 并行多线程编码的 I 帧去闪烁解决方案的示意图；

[0014] 图 3 是图 2 的多线程 I 帧去闪烁解决方案的作为结果的去闪烁性能的曲线图；

[0015] 图 4 是多线程 I 帧去闪烁框架的框图；

[0016] 图 5 是示出来自图 4 的去闪烁缓存器的合适参考帧加载的框图；

[0017] 图 6 是示出将当前帧编码结果缓存到图 4 的去闪烁缓存器中的框图；

[0018] 图 7 是示出 I<sub>next</sub> MB 的去闪烁编码的框图；以及

[0019] 图 8 是示出 P<sub>last</sub> MB 的去闪烁编码的框图。

## 具体实施方式

[0020] 在 GOP 并行多线程视频编码情形下, GOP 以 IDR 帧开始并且以 P 帧结束。注意, 尽管 GOP 间预测 (即, 跨 GOP 边界的预测) 产生或多或少改进的编码效率, 但是难以在该 GOP 并行多线程编码架构下支持 GOP 间预测。因此, 上面的假设一般常常保持为真。不丧失一般性, 假设每个 GOP 仅具有一个 I 帧, 其也是其第一帧。

[0021] 在以下描述中, 关注点在于同一场景中的两个连续 GOP 的编码, 以及因而关注点在于第二 GOP 中的第一 I 帧的去闪烁。第一 GOP 和第二 GOP 中的最初 I 帧分别作为“ $I_{curr}$ ”以及“ $I_{next}$ ”。我们将第一 GOP 中的最后 P 帧表示为“ $P_{last}$ ”。不丧失一般性, 假设两个 GOP 分别由两个不同的编码线程编码, 并且当一个线程将要开始编码  $I_{next}$  时, 另一线程仅仅部分地编码了之前 GOP。具有最高显示顺序的第一 GOP 中的编码帧被表示为“ $P_{curr}$ ”。注意,  $P_{curr}$  帧实际上可以是除了 I 帧之外的任何帧类型。这里, 使用  $P_{curr}$  仅仅是为了表示方便。还注意,  $P_{curr}$  仅仅是之前 GOP 中最接近  $I_{next}$  的编码帧。这些表示如在图 1 和图 2 中图示。

[0022] 参考图 2, 在 GOP 并行多线程编码的情况下, 由于两个 GOP 可能由两个不同的编码线程分别编码, 因此, 在编码  $I_{next}$  18 时,  $P_{last}$  14 非常有可能尚未被编码。因此,  $I_{next}$  18 去闪烁不得不诉诸于最接近的之前编码帧, 即  $P_{curr}$  12。这里的挑战在于: 由于  $P_{curr}$  12 比  $P_{last}$  14 距  $I_{next}$  18 更远, 因此其可能具有与  $I_{next}$  18 的非常低的相关性。在该情况下, 如何从  $P_{curr}$  12 中导出用于  $I_{next}$  18 的良好的无闪烁参考是比从  $P_{last}$  14 中导出良好的无闪烁参考更为困难的任务。在至少一个实现方式中, 提出了用于解决该问题的新的简化 P 帧编码方案。如下面详细解释的, 所提出的方案与之前的简化 P 帧编码方案具有许多显著差异, 其对于良好的多线程去闪烁性能是重要的。

[0023] 除了  $I_{next}$  18 的新的去闪烁编码之外, 我们的解决方案中的第二种技术是所提出的  $P_{last}$  14 的去闪烁编码。在多线程编码中, 非常有可能: 当一线程将要编码当前 GOP 中的最后帧 (即,  $P_{last}$  14) 时, 下一 GOP 中的第一 I 帧 (即,  $I_{next}$  18) 已经被另一线程编码。在此情况下, 我们提出同样进行  $P_{last}$  14 的去闪烁编码。注意, 在  $I_{next}$  18 的去闪烁编码中, 经常将更多的比特分配给该帧, 使得可以利用小量化参数 (QP) 来编码  $I_{next}$  18, 并且因而  $I_{next}$  18 非常靠近其无闪烁参考。然而, 在新的  $P_{last}$  的去闪烁编码中, 不再期望非常接近无闪烁参考。这是因为: 尽管  $P_{last}$  14 和  $I_{next}$  18 可能高度相关, 但  $P_{curr}$  12 和  $P_{last}$  14 可能并且因而在时间上不连贯, 即在  $P_{last}$  14 的之前帧和  $I_{next}$  18 之间可能存在闪烁伪像。因此, 在该情况下, 对于  $P_{last}$  14 而言, 更优选地在其之前的编码帧和编码  $I_{next}$  18 之间进行很好的权衡以得到最佳的整体去闪烁性能, 而不是非常接近从其之前的编码帧或编码  $I_{next}$  18 导出的无闪烁参考。因此, 在所提出的  $P_{last}$  14 的去闪烁编码方案中, 仍经由与在  $I_{next}$  去闪烁编码中相同的新提出的简化 P 帧编码从编码  $I_{next}$  18 中导出其无闪烁参考。然而, 仅将中等量的附加比特进一步分配给  $P_{last}$  14。因此, 作为结果的重构  $P_{last}$  14 表示其之前帧和  $I_{next}$  18 的适当混合, 其产生它们之间的更平滑的过渡。

[0024] 在图 2 和图 3 中分别图示了所提出的整体去闪烁解决方案以及期望的去闪烁性能。

[0025] 在图 4-8 中进一步详细解释了所提出的去闪烁方案的实现方式。图 4 示出了所提

出的用于编码每个帧的方案的整体流程图。这示出了由所有的编码线程分别进行所提出的帧编码方案。在步骤 20, 当一线程正在编码一帧时, 其首先检查是否是合格的 (qualified) P<sub>last</sub> 14 或 I<sub>next</sub> 帧 18。如果是, 则该线程将从 deflicker\_buffer (去闪烁缓存器) 中加载适当参考帧以用于该帧的去闪烁编码。

[0026] 在该实现方式中, deflicker\_buffer 是重要的并且是有用的缓存机制, 其有助于所有的多个线程缓存并且共享它们的编码结果以用于 I<sub>next</sub> 18 或 P<sub>last</sub> 14 去闪烁。在我们的当前实现方式中, deflicker\_buffer 包括三个部分:

[0027] 1) deflicker\_var\_buffer: 每个编码线程一个, 由线程 ID 来编制索引, 记录线程的编码状态变量, 例如, 当前的编码帧号 (由“CurrFrmNumber”表示), 从当前帧到 GOP 结束的累计的帧编码复杂度 (由“SumComplexityFromCurrFrmToGOPEnd”表示) 等。

[0028] 2) deflicker\_frm\_buffer: 所有线程一个, 缓存最近的 P<sub>last</sub> 或 I<sub>next</sub> 及其相关的其它信息以用于可能的去闪烁编码;

[0029] 3) prev\_frm\_buffer: 每个编码线程一个, 对于每个线程缓存具有最高显示顺序的编码帧及其相关的其它信息。

[0030] 在图 4-8 中解释这些缓存器的使用。注意, 为了简洁, 未示出缓存器的初始化。在步骤 24, 传统的 MB 编码处理采取原始视频帧作为目标帧, 并且然后通常基于最小化的率失真 (RD) 成本准则从所有的 MB 编码模式选项 (即, 包括所有的帧间和帧内预测模式) 中选择最佳编码模式。这是所谓的 RD 优化模式选择 (RDO-MS)。然后, 将利用所选择的最佳编码模式将 MB 编码为输出比特流。在步骤 78 和 96 中也解释了分别用于 I 帧和 P 帧中的 MB 的传统的 MB 编码。对于步骤 26, 在图 8 中详细解释了 P<sub>last</sub> MB 的去闪烁编码。在图 5 中的步骤 42、44、46、48 和 49 中以及在图 6 中分别解释了其参考帧缓存器加载和更新。图 6 提供了步骤 28 的细节, 其中, 如图 5 中的步骤 49、44 和 40 中所示地管理所涉及的变量 SaveCurrFrm。

[0031] 图 5 解释了从 deflicker\_buffer 加载适当参考帧。“curr\_thread\_ID”是标识当前编码线程的索引。在步骤 30, “SumComplexityToGOPEnd”是每个帧的被用来度量当前帧和 I<sub>next</sub> 之间的相关性的量。在当前实现方式中, 两个连续帧之间的复杂度被计算如下。

$$[0032] \quad Cmpl = \bar{R}_{mv} + \overline{MAD} \quad (1)$$

[0033] 这里, Cmpl 表示后一帧的复杂度。 $\bar{R}_{mv}$  表示一帧中所有 MB 上的平均 MV 编码比特, 而  $\overline{MAD}$  表示一帧中所有 MB 上的 MB 运动估计误差的平均的亮度绝对平均差 (MAD)。注意, 图 6 中的 TH1 和图 7 中的 TH2 是与特定复杂度度量有关的阈值。利用 (1) 中的复杂度度量, TH1 = 250, TH2 = 20。可以看到更高的复杂度意味着帧之间更低的相关性。

[0034] 图 5 示出了当编码 P<sub>last</sub> 14 时该处理检查 I<sub>next</sub> 18 是否已被编码 (步骤 32、34)。如果是, 则在步骤 36 等待 P<sub>last</sub> 编码完成, 然后从 deflicker\_buffer 加载 I<sub>next</sub> 18 (步骤 40) 以用于 P<sub>last</sub> 14 的去闪烁编码。否则, P<sub>last</sub> 14 将在步骤 42 和 44 经历传统的 P 帧编码处理。当编码 I<sub>next</sub> 时, 首先在步骤 38 检查 P<sub>last</sub> 是否可用。如果是, 则加载 P<sub>last</sub> 以对 I<sub>next</sub> 进行去闪烁编码 (步骤 40)。否则, 在步骤 42 进一步检查有用的 P<sub>curr</sub> 是否可用。这里, 有用的 P<sub>curr</sub> 被定义为具有 SumComplexityToGOPEnd < TH1 的 P<sub>curr</sub> 帧, 即可能与 I<sub>next</sub> 很好地相关的 P<sub>curr</sub>。如果是, 则在步骤 44 将加载 P<sub>curr</sub> 以用于 I<sub>next</sub> 去闪烁。在步骤 46, 由于多线程编码, 在步骤 46 一个线程正在编码 P<sub>last</sub> 的同时,

可能将  $I_{next}$  分配为另一线程,并且已经被编码、或者尚未开始编码、或者正在编码中。步骤 46 检查  $I_{next}$  是否正在编码中。如果是,则当前的编码线程将等待,直到其它线程结束  $I_{next}$  编码。因此,在步骤 46 之后, $I_{next}$  已经被完全编码或者尚未开始编码。然后,步骤 48 将检查哪种情况为真。如果  $I_{next}$  已经被编码,则将前进到步骤 49。否则,前进到步骤 42。如在步骤 49 中所解释的,当  $I_{next}$  被编码,将利用其来生成用于当前  $P_{last}$  帧的 MB 去闪烁编码的无闪烁参考。在图 5 中,将原始的之前帧和重构的之前帧表示为  $PrevFrmOrig$  和  $PrevFrmRecon$ 。关于  $P_{last}$  MB 编码,在图 8 的步骤 82 中使用  $PrevFrmRecon$ ,并且在步骤 92 中使用这两者来计算  $P_{16 \times 16}$  预测参考 MB 所包含的重构失真。 $DeflickerCurrFrm$  是在当前实现方式中所使用的标志,其指示去闪烁编码是否被用于当前帧编码。 $SaveCurrFrm$  是图 6 的步骤 50 中所检查的用于更新  $deflicker\_buffer$  的标志。

[0035] 图 6 示出了利用当前帧编码结果的  $deflicker\_buffer$  更新。在步骤 50,如果  $SaveCurrFrm$  为真 (true),则在步骤 54,  $I_{next}$  18 或  $P_{last}$  14 帧将记录在  $deflicker\_frm\_buffer$  中以便稍后分别用于  $P_{last}$  14 或  $I_{next}$  18 的去闪烁编码。否则,如果当前编码帧到目前为止是对于  $I_{next}$  去闪烁最有用的帧,则将在步骤 52、53 将当前帧结果记录在  $prev\_frm\_buffer[curr\_thread\_ID]$  中,其稍后将被加载为用于  $I_{next}$  去闪烁的  $P_{curr}$ 。注意,仅当满足图 6 中的所有四个条件时,才需要缓存当前帧结果。

[0036] 图 7 示出了  $I_{next}$  MB 的去闪烁编码。这里,  $QP$  表示当前 MB 编码  $QP$ 。 $QP_{PrevFrm}$  表示所加载的参考帧的 MB 平均  $QP$ 。 $ME\_range$  表示运动矢量搜索范围。这里,  $ME\_SAD$  表示运动估计后所选择的运动矢量的预测残差的绝对差之和。 $TH3 = 10$ 。该条件是在步骤 60 和 62 检查 MB 是运动的还是静止的。 $QP_{CurrMB}$  表示从率控制计算的当前 MB 编码  $QP$ 。注意,在率控制中,将向  $I_{next}$  分配更多比特以确保其低  $QP$  编码,从而使得编码的  $I_{next}$  非常接近其无闪烁参考。在图 7 中,如果  $P_{last}$  在步骤 63 被用来生成用于  $I_{next}$  去闪烁编码的  $I_{next}$  无闪烁参考,则将预期去闪烁编码与如在步骤 64 和 78 中所示的 GOP 顺序单线程方案相同。该情况实际上表示 GOP 并行多线程编码的最佳可实现去闪烁性能。否则,  $P_{curr}$  而不是  $P_{last}$  将被用来生成无闪烁参考并被用于  $I_{next}$  的去闪烁编码,在步骤 66-76 中对其进行解释。图 7 示出了新提出的简化 P 帧编码的至少一个实现方式的细节,并且该实现方式包含许多与用于单线程编码的简化 P 帧编码方案的显著差异。将这些差异总结如下:

[0037] 1) 自适应的 ME 搜索范围:如果  $P_{curr}$  与  $I_{next}$  高度相关,则使用较小的搜索范围 (例如,5)。否则,使用较大的搜索范围 (例如,10)。

[0038] 2) 在简化的 RD 优化模式选择 (RDO-MS) 中无跳跃模式检查

[0039] 3) 如果当前 MB 不是静止 MB 或者如果经由 RDO-MS 选择了帧间模式,总使用具有质量匹配  $QP$  的  $P_{16 \times 16}$  模式来生成无闪烁参考。

[0040] 除了这些差异之外,步骤 66-76 中的经由  $P_{curr}$  的  $I_{next}$  去闪烁编码遵循与传统的单线程 I 帧去闪烁编码几乎相同的方案。简言之,如果 MB 的  $ME\_SAD$  大于阈值并且最佳 RD 优选模式是帧内预测模式,则 MB 被识别为高运动,并且因此是闪烁不敏感的 MB,对于其而言,去闪烁编码不是必须的,并且因此其将以如步骤 90 中所示的将原始 MB 取为目标 MB 的传统方式编码。否则,MB 被识别为低运动,并且因此是倾向于闪烁的 MB,其将被编码以便去闪烁。在此情况下,将如步骤 92 中所示地首先生成无闪烁参考 MB,然后其将被取为用于

当前 MB 编码的目标 MB。

[0041] 图 8 示出了 P\_last MB 的去闪烁编码。与图 7 中 I\_next MB 的去闪烁编码的差异在于：

[0042] 1) 当 P\_last 紧接在 I\_next 之前时，它们之间的高度相关区域必定是低运动的从而是倾向于闪烁的。因此，在步骤 80 设置较小的 ME 搜索范围是合适的。类似于步骤 66-76，步骤 84-90 遵循与传统的单线程 I 帧去闪烁编码几乎相同的方案。

[0043] 2) 如步骤 92 和 94 中所示，来自 P\_last MB 去闪烁的率控制的 QP\_CurrMB 具有中间值。这是因为，如先前讨论的，P\_last 的中间编码质量是优选的，从而为其重构产生编码 I\_next 及其之前的编码帧之间的适当权衡或混合。

[0044] 3) 在实际编码的第二回合中，不使用跳跃模式。相反，将在步骤 96 使用安全跳跃 (Safe\_Skip) 模式。安全跳跃模式实际上是具有与跳跃模式相同的 MV 的替代 P 16x16 模式，即不招致 MV 编码比特。注意，在该模式中，将编码预测残差，从而防止跳跃模式编码的不期望的差质量。跳跃模式是大数目的最近视频编码标准（例如，H. 264/AVC）中的标准化的 MB 编码模式，其表明将使用帧间预测来编码 MB，然而其将简单地使用依据运动补偿而从相邻的编码 MB 的运动矢量预测的精确运动矢量，并且不包括预测残差的编码。因此，其表示最少比特消耗的 MB 编码模式，然而，往往是所有编码模式中具有最大编码失真的模式。安全跳跃模式是我们提出的对于跳跃模式的新的替代模式，其使用与跳跃模式的运动矢量相同的运动矢量，然而，其与 P16x16 模式一样地编码预测残差。因此，与其它帧间预测模式（例如 P16x8、8x16、8x8、8x4、4x8、4x4 等）相比，其在运动矢量编码上不耗费比特，而同时由于所包含的残差编码而产生类似的编码失真。

[0045] 而且，注意，在 P\_last 或 I\_next MB 无闪烁生成中的简化 RDO-MS 都包含每个候选模式的修改后的 RD 成本，这对于最终的显著的和可靠的去闪烁性能而言也是关键的。基本上，经由在 RDO-MS 中修改 RD 成本，更阻止跳跃模式和帧内模式，而帧间预测模式是更受欢迎的。这证明是用于实现更好的去闪烁性能的有效手段。具体地，在无闪烁参考生成中，对于增加的偏好和对于 P\_last MB 而言，帧间模式的 RD 成本被乘以 0.7，在无闪烁参考生成以及实际编码两者中，对于降低的偏好，帧内模式的 RD 成本被乘以 2.5。

[0046] 最后一点也很重要，如先前所述，率控制必须与 I\_next 18 和 P\_last 14 的去闪烁编码良好地协调。基本上，在帧级别的率控制中，需要将更多比特分配给 I\_next 去闪烁，而需要将中等量的比特分配给 P\_last 去闪烁。这通常可以通过在进行帧级别比特分配时为帧分配合适的 QP 偏移来实现。在我们的当前实现方式中，我们分别向 I\_next 和 P\_last 分配 -6 和 -2 的 QP 偏移。

[0047] 已经进行了实验来评估上面提出的 GOP 并行多线程去闪烁解决方案的性能。结果表明所提出的方案能够有效的降低多线程编码情况下的 I 帧闪烁伪像，而所招致的附加计算复杂度不对实时编码的实施施加严重挑战。尤其，我们发现较短的 GOP 长度（例如，< 60）比较大的 GOP 长度（例如，> 90）更有希望得到更好的去闪烁性能，这是因为：利用较短的 GOP 长度，P\_curr 12 和 I\_next 18 之间的间距也更可能地短，这对于良好的去闪烁而言是高度受欢迎的。

[0048] 这里，提供了具有特定特征和方面的一个或多个实现方式。然而，所描述的实现方式的特征和方面还可以被适配于其它实现方式。例如，可以使用一个、两个、或多个回合来



执行实现方式,即使这里参考特定数目的回合来进行描述。另外,QP 可以对于给定画面或帧而变化,诸如例如,基于 MB 的特性而变化。尽管这里描述的实现方式可能是在特定背景下描述的,但是这样的描述将决不应被当作将所述特征和构思限制于这样的实现方式或背景。

[0049] 这里描述的实现方式可以以例如方法或处理、装置、或软件程序来实现。即使仅仅在单个实现方式形式的背景下进行讨论(例如,仅作为方法讨论),所讨论的实现方式或特征也可以以其它形式(例如,装置或程序)来实现。装置可以以例如适当硬件、软件和固件来实现。方法可以以例如在诸如例如计算机或其它处理设备的装置中实现。另外,方法可以通过被处理设备或其它装置执行的指令来实现,并且这样的指令可以存储在诸如例如 CD 的计算机可读介质上、或其它计算机可读存储设备上或者可以存储在集成电路上。此外,计算机可读介质可以存储由实现方式产生的数据值。

[0050] 如对于本领域技术人员而言明显的,实现方式还可以产生被格式化为承载可以例如被存储或传输的信息的信号。该信息可以包括例如用于执行方法的指令、或者由所描述的实现方式之一产生的数据。

[0051] 另外,许多实现方式可以在编码器、编码器的预处理器、解码器、或解码器的后处理器中的一个或多个中实现。

[0052] 此外,根据该公开可预期其它实现方式。例如,通过组合、删除、修改或补充所公开的实现方式中的各种特征,可以创建另外的实现方式。

[0053] 以下列表提供了各种实现方式的简短列表。该列表不意图是穷举的而仅仅是将许多可能实现方式中一小部分的简短描述提供如下:

[0054] 1. 一种利用多个编码线程对 GOP 并行实时编码的视频编码器,其通过利用从之前 GOP 中的最靠近的编码帧导出的无闪烁参考来首先对 I 帧进行去闪烁编码、并且然后利用从去闪烁编码的 I 帧导出的无闪烁参考来对该之前 GOP 中的最后 P 帧进行去闪烁编码,来降低 I 帧闪烁。

[0055] 2. 在实现方式 1 中,在第一 I 帧的实际编码中使用小 QP 以便非常接近其无闪烁参考,而在最后 P 帧的实际编码中使用中等 QP 以便为编码帧产生下一 GOP 中的编码 I 帧和当前 GOP 中的之前编码帧的权衡混合。

[0056] 3. 在实现方式 1 中,在一帧的去闪烁编码中,经由简化 P 帧编码的一个回合生成无闪烁参考。

[0057] 4. 在实现方式 3 中,简化 P 帧编码包含:(i) 对于当前 I 帧和之前 GOP 中最靠近的编码帧之间较低的相关性,较大的运动搜索范围,反之亦然,(ii) 在模式选择中无跳跃模式检查,(iii) RDO-MS 中修改的 RD 成本阻止跳跃模式和帧内模式。

[0058] 5. 在实现方式 1 中,使用总的帧间复杂度来确定当前 I 帧和之前 GOP 中最靠近的编码帧之间的相关性级别。

[0059] 6. 在实现方式 1 中,对于 GOP 中最后 P 帧的去闪烁编码,在实际 MB 编码中,使用如在本公开的一个或多个实现方式中定义的安全跳跃模式,而不是传统的跳跃模式。

[0060] 7. 在实现方式 1 中,采用如在本公开的一个或多个实现方式中定义的多线程缓存和通信机制,其分离地对于每个编码线程缓存具有 GOP 中最高显示顺序的重构编码帧、对于所有线程缓存 GOP 的重构的最后 P 帧和最初 I 帧。

- [0061] 8. 从所描述的实现方式中任一个产生的信号。
- [0062] 9. 根据在本公开中描述的一个或多个实现方式创建、组合、存储、传输、接收和/或处理用于 I 帧和 P 帧的视频编码信息以便降低闪烁。
- [0063] 10. 一种能够根据所描述的实现方式之一或者与所描述的实现方式之一沟通地操作的设备（诸如例如编码器、解码器、预处理器、或后处理器）。
- [0064] 11. 一种设备（例如，计算机可读介质），其用于存储根据在本公开中描述的一个或多个实现方式的 I 帧或 P 帧之一或编码、或者执行 I 帧或 P 帧的编码的一组指令。
- [0065] 12. 一种信号，其被格式化为包括涉及根据在本公开中描述的一个或多个实现方式的、I 帧或 P 帧的编码的信息。
- [0066] 13. 在实现方式 12 中，其中信号表示数字信息。
- [0067] 14. 在实现方式 12 中，其中信号是电磁波。
- [0068] 15. 在实现方式 12 中，其中信号是基带信号。
- [0069] 16. 在实现方式 12 中，其中信息包括残差数据、运动矢量数据、以及参考指示符数据中的一个或多个。
- [0070] 17. 一种对于视频的多线程编码降低闪烁的处理、或者一种用于实现对于视频的多线程编码降低闪烁的处理的设备、或者一组用于实现对于视频的多线程编码降低闪烁的处理的指令。
- [0071] 所描述的实施例提出了用于 GOP 并行多线程视频编码的有效的 I 帧去闪烁方案。所提出的方案可以降低紧接在前的重构帧的不可用性对当前 I 帧去闪烁的影响。该方案也是高效的，这是因为其招致少量的附加计算和存储器成本，并且因此在实时视频编码系统中很好地适应。
- [0072] 总之，这里提出了适当地改变编码器的手段及其方法，其中，以更直接和一般的方式进行编码以便解决上面讨论的各种伪像消除问题。
- [0073] 尽管一些方案对于所有的帧内编码视频（利用运动 JPEG2000 标准或者利用 H. 264/AVC 标准的）都解决去闪烁问题，但是本公开中的至少一个实现方式提供与主流视频编码标准（即，公知的具有运动补偿和变换编码的混合编码模式）兼容的去闪烁技术范例。此外，本申请涉及按 GOP 编码的视频，其中每个 GOP 以 I 帧开始。

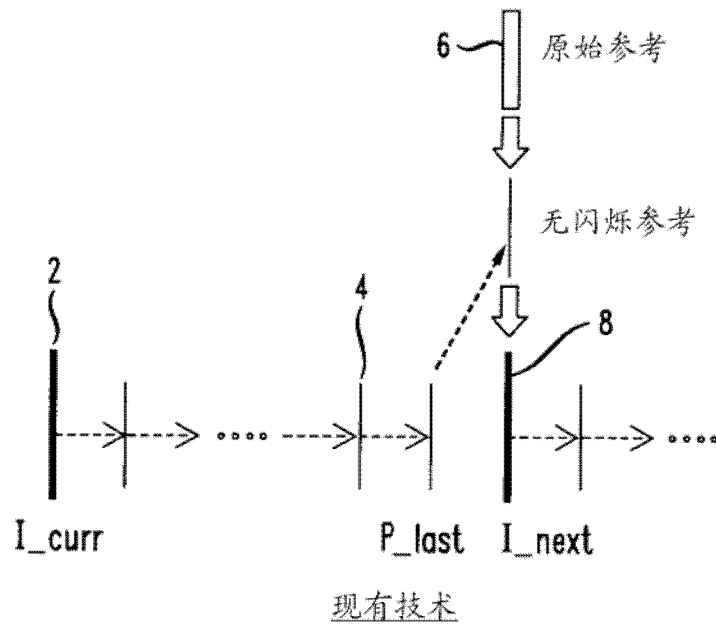


图 1

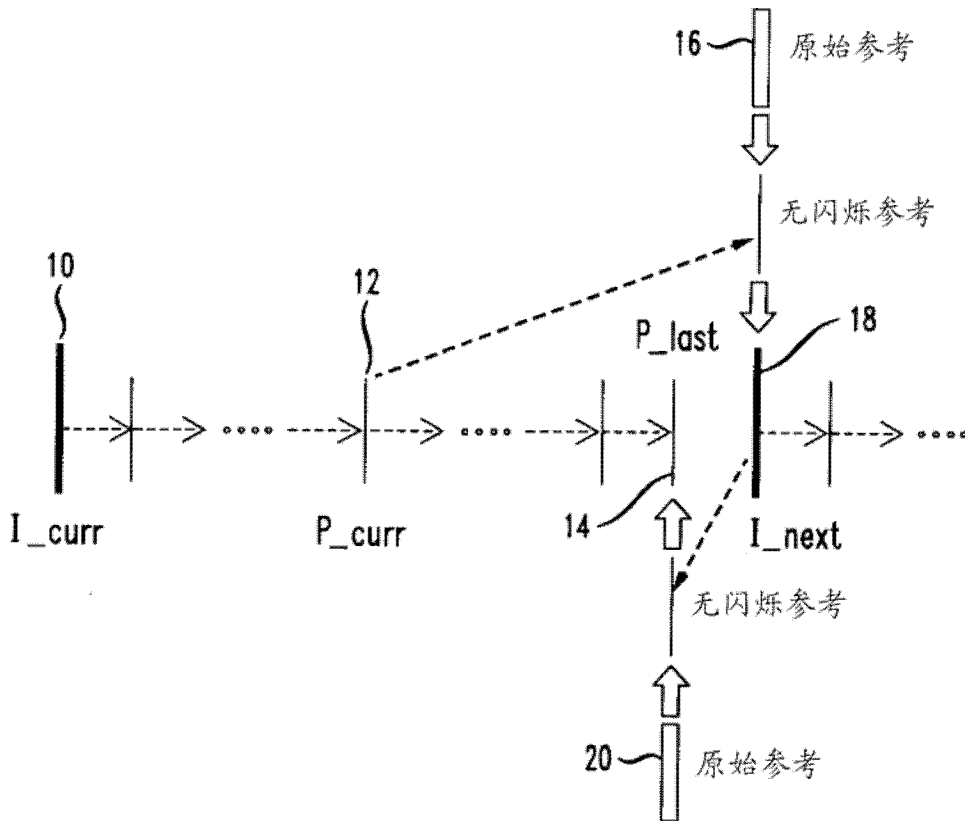
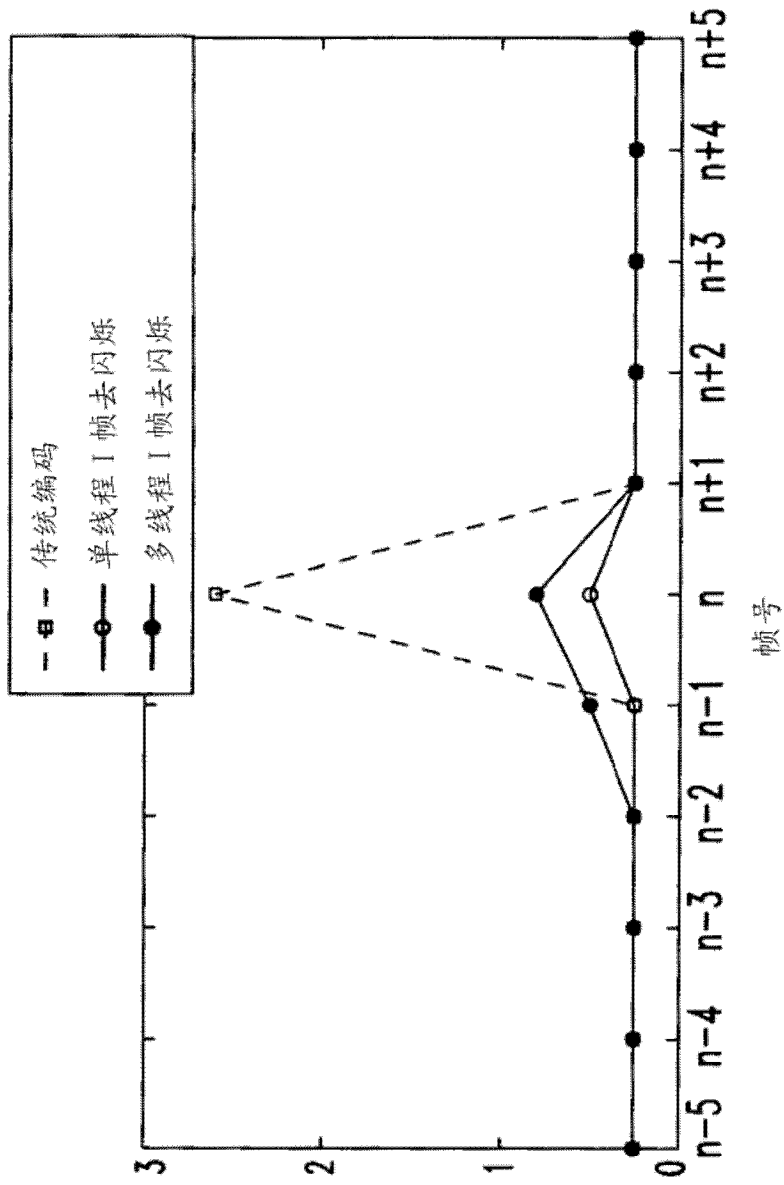


图 2



闪烁失真

图 3

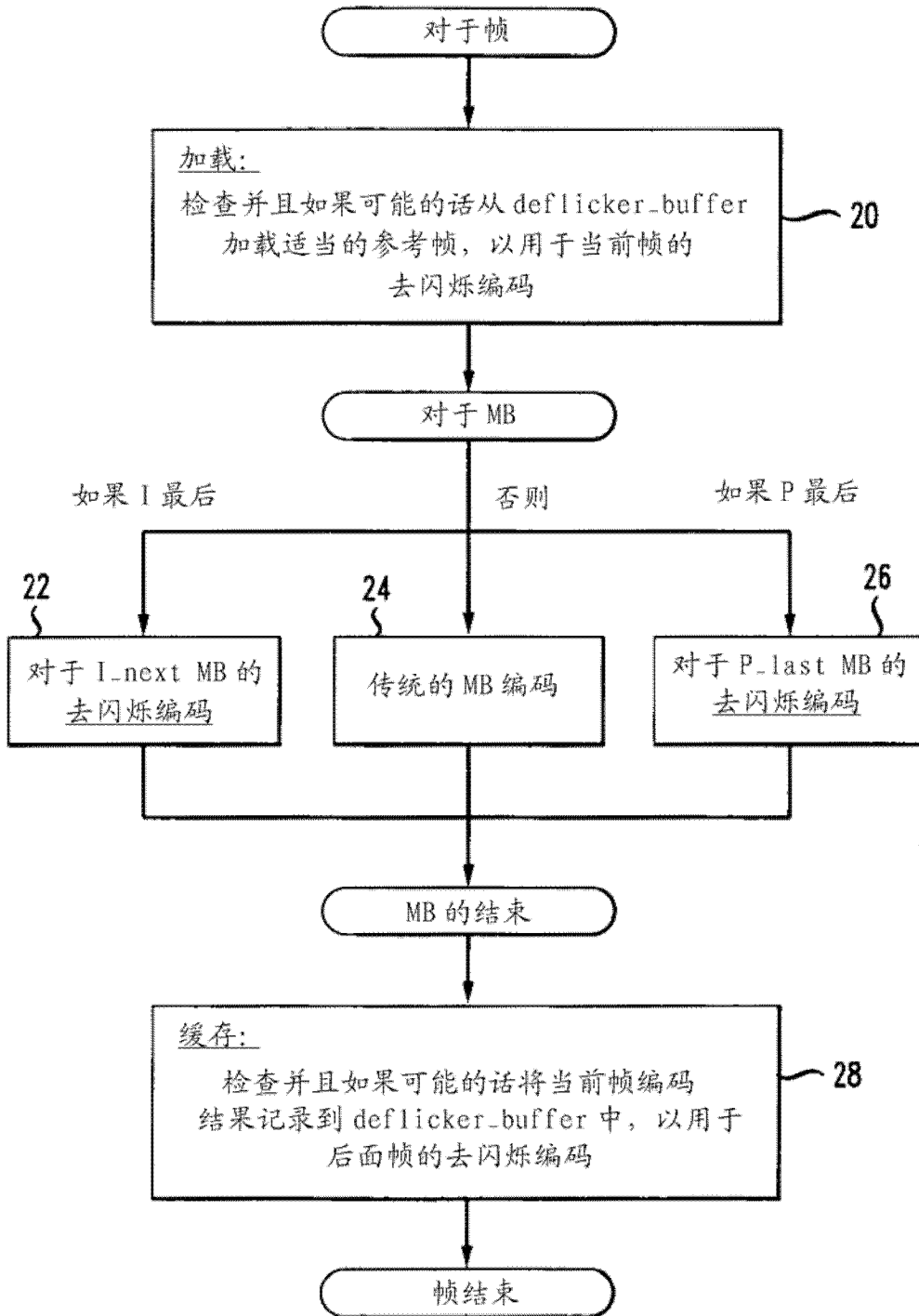


图 4

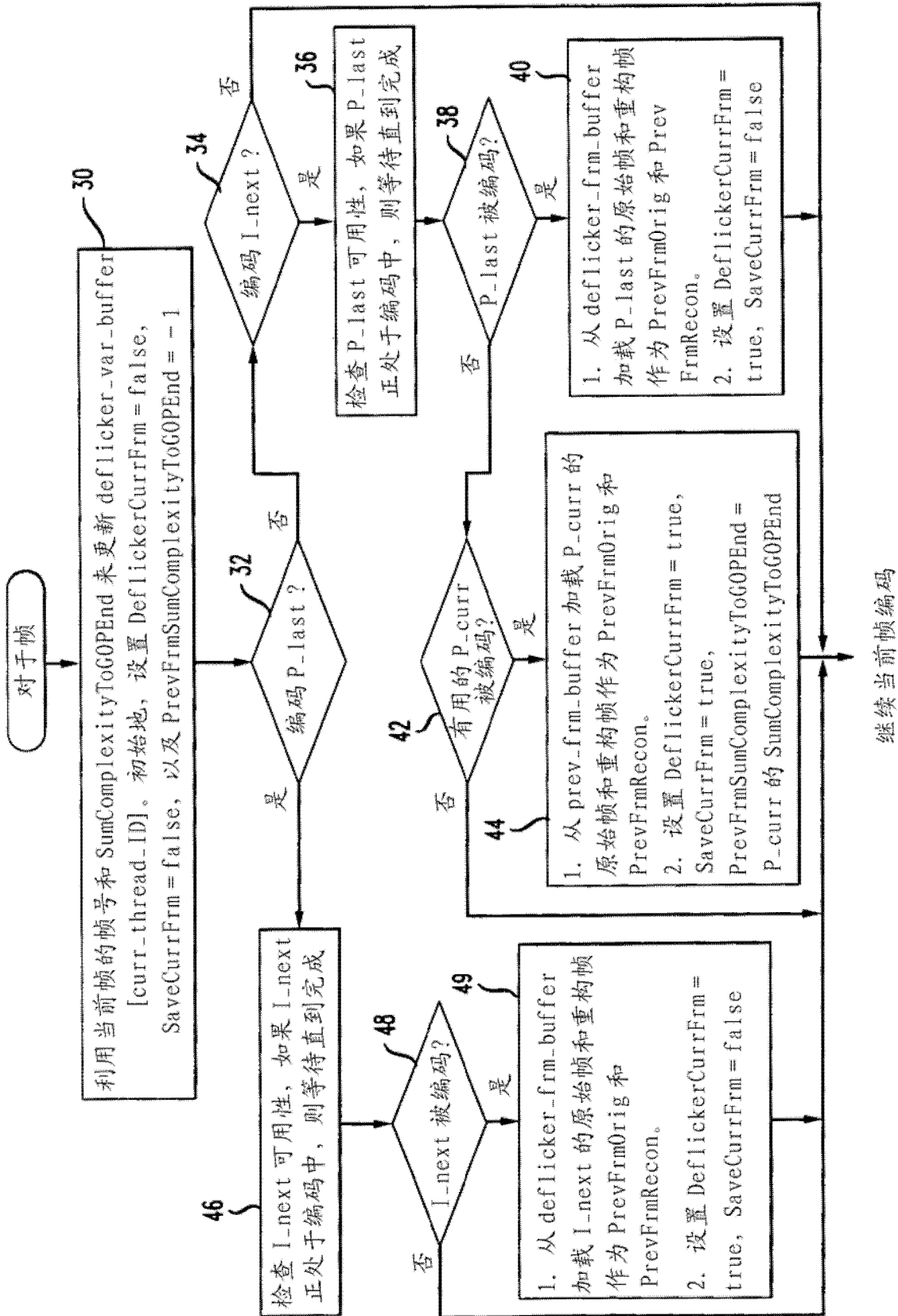


图 5

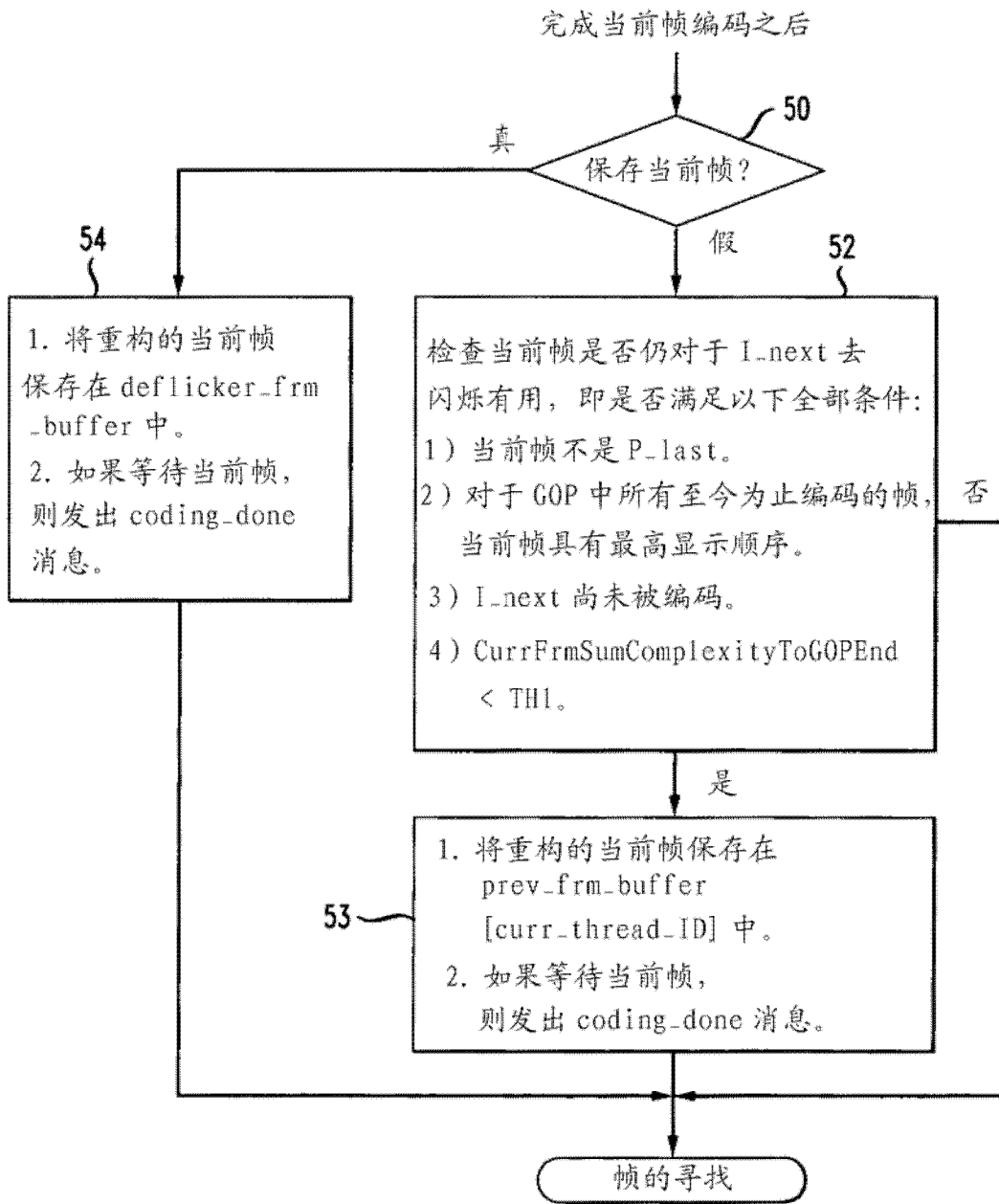


图 6

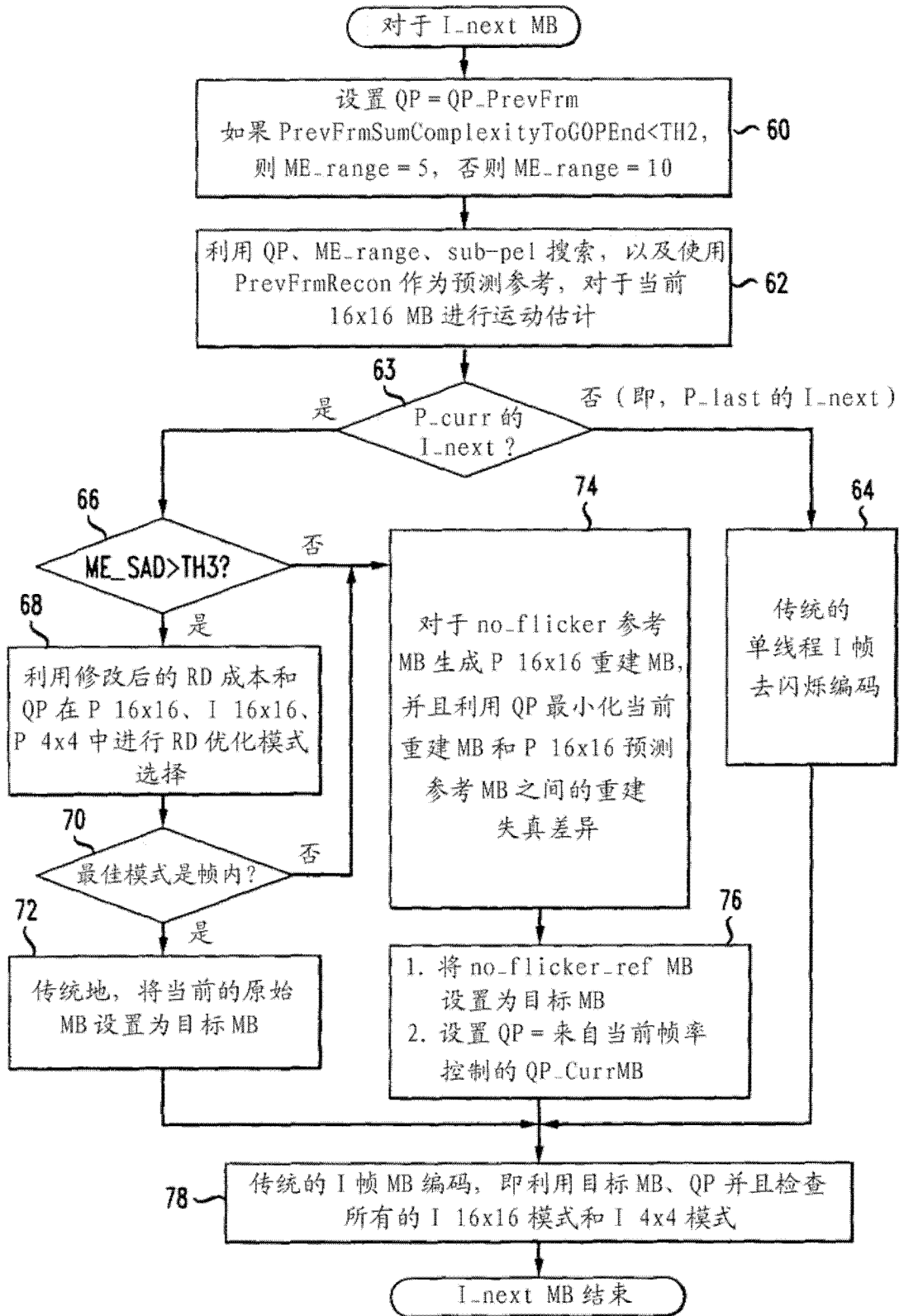


图 7



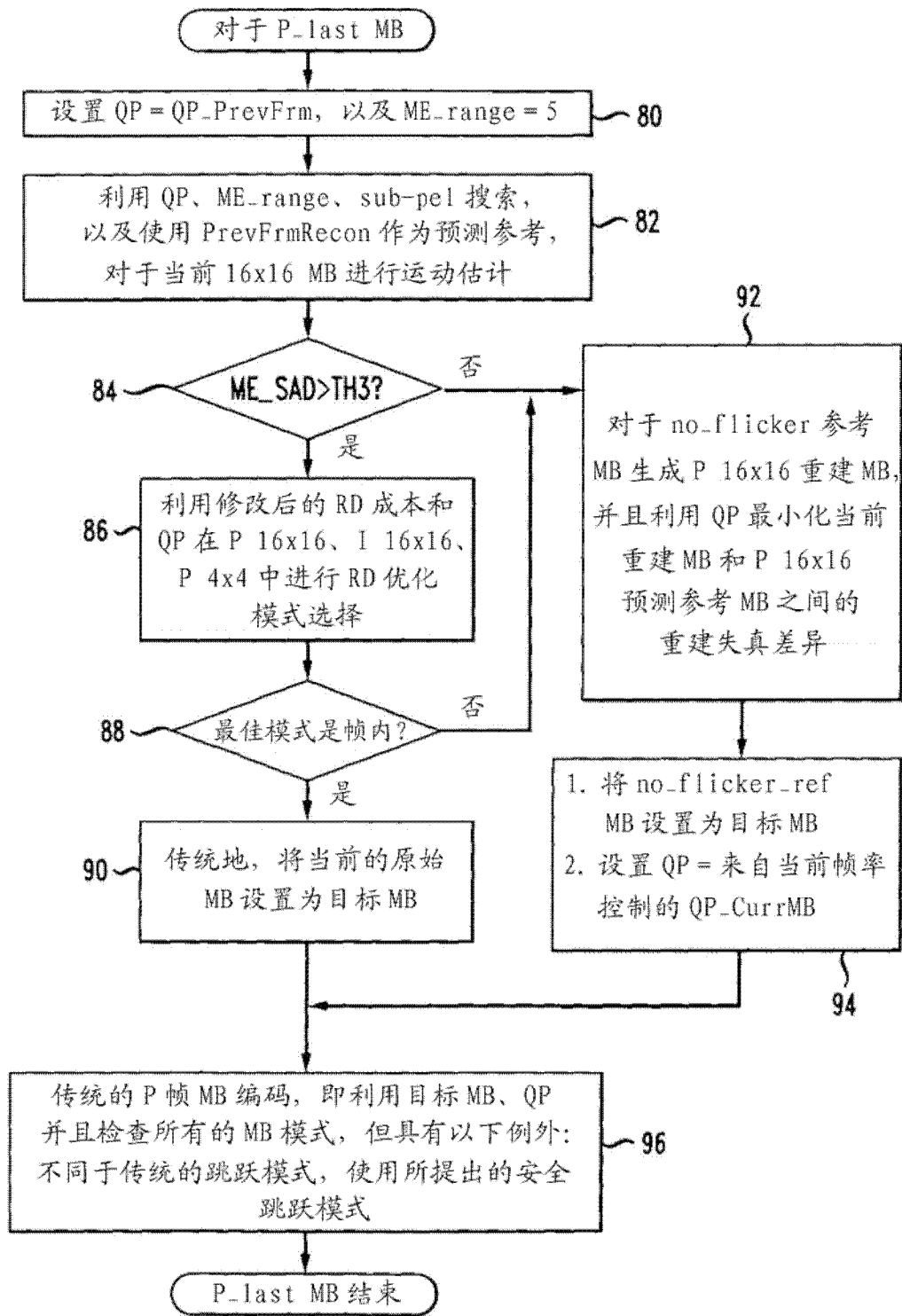


图 8