(19)中华人民共和国国家知识产权局

(12)发明专利

(10)授权公告号 CN 102833539 B
(45)授权公告日 2015.03.25

(21)申请号 201210271659.1

(22)申请日 2005.06.24

(30)优先权数据
60/583,418 2004.06.27 US
60/643,918 2005.01.09 US
11/118,604 2005.04.28 US
11/118,616 2005.04.28 US

(62)分案原申请数据
200580006363.5 2005.06.24

(73)专利权人 苹果公司
地址 美国加利福尼亚

(72)发明人 童歆 吴锡荣 托马斯·彭
安德里亚那·杜米特拉
巴林·哈斯凯尔 吉姆·诺米勒

(74)专利代理机构 中国国际贸易促进委员会专利商标事务所 11038
代理人 董莘

(51)Int.Cl.
H04N 19/176(2014.01)
H04N 19/172(2014.01)
H04N 19/15(2014.01)
H04N 19/126(2014.01)
H04N 19/14(2014.01)
H04N 19/137(2014.01)
H04N 19/142(2014.01)
H04N 19/152(2014.01)
H04N 19/154(2014.01)
H04N 19/177(2014.01)
H04N 19/192(2014.01)

(56)对比文件
US 2002/0057739 A1,2002.05.16,
US 2003/0061038 A1,2003.03.27,
CN 1251004 A,2000.04.19,
CN 1176562 A,1998.03.18,
US 2003/0115050 A1,2003.06.19,

审查员 姜丹

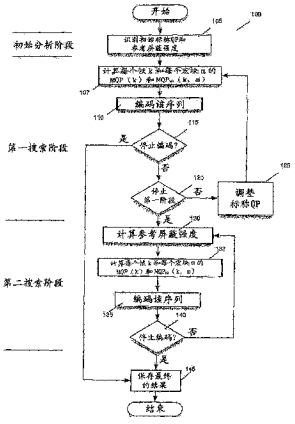
权利要求书2页 说明书19页 附图9页

(54)发明名称

多通路视频编码

(57)摘要

本发明的一些实施例提供了一种编码多个图像(例如,视频序列的几帧)的多通路编码方法。该方法重复执行编码这些图像的编码操作。该编码操作基于标称量化参数,该方法使用该标称量化参数计算该图像的量化参数。在该编码操作的几次不同的迭代操作期间,该方法使用了几种不同的标称量化参数。该方法在达到了终结准则(例如,其识别到一个可接受的图像编码)时停止其迭代过程。



CN 102833539 B

1. 一种编码视频图像序列的方法，所述方法包括：

通过使用多个编码参数编码所述视频图像序列以产生当前编码方案；

从通过所述当前编码方案编码的所述视频图像序列中，通过连续耗尽参考解码器的输入缓冲区直到所述输入缓冲区中的下溢条件改进为止，识别出导致所述输入缓冲区的所述下溢条件的多个视频图像，其中所述参考解码器的输入缓冲区在编码期间被使用以模拟解码器输入缓冲区；

调整用于所识别出的多个视频图像的一组编码参数；

使用调整后的编码参数组来编码所识别出的多个视频图像以产生被指定为当前编码方案的新编码方案；以及

重复所述识别、所述调整和所述编码以产生所述新编码方案，直到所述视频图像序列的编码不会导致所述参考解码器的所述输入缓冲区下溢为止。

2. 根据权利要求 1 的方法，还包括：当所述新编码方案不会导致输入缓冲区下溢时，编码所述视频图像序列的从在导致所述下溢的所述多个图像之后的第一图像开始的剩余图像。

3. 根据权利要求 1 的方法，其中所述编码所述视频图像序列包括：

使用所述参考解码器的所述输入缓冲区来模拟所述解码器的解码器输入缓冲区；以及

利用所述模拟来选择多个比特以编码所述视频图像序列，同时防止所述参考解码器的所述输入缓冲区的下溢。

4. 根据权利要求 3 的方法，其中模拟所述解码器输入缓冲区还包括考虑所述解码器经其接收编码数据的网络的条件。

5. 根据权利要求 3 的方法，其中模拟所述解码器输入缓冲区还包括考虑所述解码器输入缓冲区的大小。

6. 根据权利要求 3 的方法，其中模拟所述解码器输入缓冲区还包括考虑来自所述解码器输入缓冲区的初始移除延迟。

7. 根据权利要求 1 的方法，还包括：对于所述多个视频图像的每一次编码都满足目标比特率，其中满足所述目标比特率是多个质量准则之一，其中每个产生的编码方案都满足所述质量准则的全部。

8. 一种编码图像序列的方法，所述方法包括：

使用一组参数编码所述图像序列；

模拟解码器的输入缓冲区；

在所述图像序列中的多个图像导致所述解码器的所述输入缓冲区下溢条件时，通过连续耗尽所述输入缓冲区直到所述输入缓冲区中的所述下溢条件改进为止，调整所述参数组；以及

使用调整后的参数组编码导致所述下溢的多个图像，其中调整所述参数组和编码所述多个图像被迭代地执行，直到所述多个图像导致的所述输入缓冲区的下溢被消除为止。

9. 根据权利要求 8 的方法，其中模拟所述输入缓冲区包括：考虑连接速度和缓冲区大小之一。

10. 根据权利要求 8 的方法，其中模拟所述输入缓冲区包括：考虑所述解码器的解码过程的速度。

11. 根据权利要求 8 的方法，还包括：在所述下溢被消除之后，编码所述图像序列中的在导致所述下溢的所述多个图像之后的下一组图像。

12. 一种编码设备，包括用于执行如权利要求 1-11 中的任一项所述的方法中的步骤的装置。

多通路视频编码

[0001] 本申请是申请日为 2005 年 6 月 24 日、申请号为 200580006363.5、发明名称为“多通路视频编码”的发明专利申请的分案申请。

背景技术

[0002] 视频编码器通过利用多种编码方案编码视频图像序列(例如,视频帧)。视频编码方案典型地是以内帧或帧间的方式编码视频帧或视频帧的各部分(例如,视频帧内的像素集)。内帧编码的帧或像素集是独立于其他帧或其他帧内的像素集来编码的。帧间编码的帧或像素集是通过参考一个或多个其他帧或其他帧内的像素集来编码的。

[0003] 当压缩视频帧时,一些编码器实现了“速率控制器”,其为将要编码的视频帧或视频帧的集合提供“比特预算”。比特预算指定已经分配给编码该视频帧或视频帧集合的比特数量。通过有效分配比特预算,速率控制器试图生成考虑到某种限制(例如,目标比特率等)的最高质量压缩的视频流。

[0004] 迄今为止,已经提出了多种单通路和多通路速率控制器。单通路速率控制器为在单个通路中编码一系列视频图像的编码方案提供比特预算,而多通路速率控制器为在多个通路中编码一系列视频图像的编码方案提供比特预算。

[0005] 单通路速率控制器在实时编码条件下是有效的。另一方面,多通路速率控制器基于一组限制为特定比特率优化编码。迄今为止,并没有很多的速率控制器在控制它们的比特率中考虑到帧或帧内像素集的空间或时间的复杂度。同样,大多数多通路速率控制器没有为虑及所期望比特率而对帧和 / 或帧内像素集使用最优量化参数的编码解决方案充分搜索解空间。

[0006] 因此,现有技术中存在对使用新颖技术的速率控制器的需求,以便在控制用于编码一组视频图像的比特率的同时,考虑视频图像和 / 或视频图像各部分的空间或时间复杂度。现有技术中还存在对多通路速率控制器的需求,其充分检查各种编码方案以识别出针对视频图像和 / 或视频图像各部分使用最优量化参数集的编码方案。

发明内容

[0007] 本发明的一些实施例提供一种编码多个图像(例如,视频序列的多个帧)的多通路编码方法。该方法重复执行编码这些图像的编码操作。该编码操作是基于标称量化参数,该方法使用该标称量化参数计算这些图像的量化参数。在该编码操作的几次不同的迭代过程中,该方法使用了几种不同的标称量化参数。该方法在达到了终结准则(例如,其识别到一个可接受的图像编码)时停止其迭代过程。

[0008] 本发明的一些实施例提供一种用于编码视频序列的方法。该方法识别量化视频中的第一图像的复杂度的第一属性。它还基于所述识别的第一属性为编码第一图像识别量化参数。该方法接着基于所述识别的量化参数编码第一图像。在一些实施例中,这种方法为视频中的多个图像执行这三项操作。

[0009] 本发明的一些实施例基于视频图像和 / 或视频图像的各部分的“视觉掩蔽”属性

编码视频图像序列。图像或图像各部分的视觉掩蔽是对在图像或图像各部分中能够忍受多少编码人工因素的指示。为了表达图像或图像各部分的视觉掩蔽属性，一些实施例计算了量化图像或图像各部分的亮度能量的视觉掩蔽强度。在一些实施例中，该亮度能量测量作为图像或图像各部分的平均 luma 或像素能量的函数。

[0010] 替代该亮度能量或与之结合，图像或图像各部分的视觉掩蔽强度也可以量化图像或图像各部分的活动性能量。活动性能量表示图像或图像各部分的复杂度。在一些实施例中，活动性能量包括量化图像或图像各部分空间复杂度的空间组件，和 / 或量化由于图像之间的移动而能够忍受 / 掩蔽的失真数量的运动组件。

[0011] 本发明的一些实施例提供一种用于编码视频序列的方法。该方法识别视频中的第一图像的视觉掩蔽属性。其还识别用于基于所述识别的视觉掩蔽属性编码第一图像的量化参数。该方法接着基于所述识别的量化参数编码第一图像。

附图说明

[0012] 本发明的新颖特征在所附权利要求书中阐述。然而，出于解释的目的，在以下附图中阐述本发明的多个实施例。

[0013] 图 1 给出了概念性举例说明本发明一些实施例的编码方法的过程 ；

[0014] 图 2 概念性举例说明了一些实施例的编解码系统 ；

[0015] 图 3 为举例说明一些实施例的编码过程的流程图 ；

[0016] 图 4a 为一些实施例中图像的标称移除时间和最终到达时间之间的区别与举例说明下溢条件的图像数量之间关系的曲线图 ；

[0017] 图 4b 举例说明了在消除下溢条件之后，对如图 4a 中所示的同一图像标称移除时间和最终到达时间的区别与图像数量之间的关系曲线图 ；

[0018] 图 5 举例说明了一些实施例中编码器用于执行下溢检测的过程 ；

[0019] 图 6 举例说明了一些实施例中编码器用于消除图像的单个片段中的下溢条件的过程 ；

[0020] 图 7 举例说明了视频流应用中缓冲器下溢管理的应用 ；

[0021] 图 8 举例说明了 HD-DVD 系统中缓冲器下溢管理的应用。

[0022] 图 9 给出了利用其实现了本发明的一个实施例的计算机系统。

具体实施方式

[0023] 在以下对本发明的详细描述中，提出并描述了本发明的众多细节、实例及实施例。然而，对本领域技术人员明确并显而易见的是，本发明并不局限于所述的实施例，并且本发明可以无需一些指定细节和所讨论实例而实施。

[0024] I. 定义

[0025] 此部分为这个文档中使用的多个符号提供了定义。

[0026] R_T 代表目标比特率，它是用于编码帧序列所期望的比特率。通常，这个比特率以比特 / 秒为单位表述，并且是从所期望的最终的文件尺寸、序列中帧的数量、以及帧速率计算得出的。

[0027] Rp 代表通路 p 的结束处所编码比特流的比特率。

[0028] Ep 代表在通路 p 的结束处比特率中的错误百分比。在一些情况下，这个百分比计算为$100\times\frac{(R_T-R_p)}{R_T}$。

[0029] ε 代表最终比特率中的误差容许范围。

[0030] ε_C代表针对第一 QP 搜索阶段的比特率中的误差容许范围。

[0031] QP 代表量化参数。

[0032] $QP_{Nom(p)}$代表为帧序列编码的通路 p 中所使用的标称量化参数。$QP_{Nom(p)}$的值由本发明的多通路编码器在第一 QP 调整阶段中调整以达到目标比特率。

[0033] $MQP_p(k)$代表屏蔽帧 QP，其是通路 p 中帧 k 的量化参数(QP)。一些实施例通过利用标称 QP 和帧级视觉掩蔽计算该值。

[0034] $MQP_{MB(p)}(k,m)$代表屏蔽宏块 QP，其是帧 k 和通路 p 的单个宏块(具有宏块索引 m)的量化参数(QP)。一些实施例通过利用 $MQP_p(k)$和宏块级视觉掩蔽计算 $MQP_{MB(p)}(k,m)$。

[0035] $\Phi_F(k)$ 代表成为帧 k 掩蔽强度的值。掩蔽强度 $\Phi_F(k)$ 是对该帧的复杂度度量，在一些实施例中，这个值被用于确定视觉编码人工因素 / 噪声将如何呈现以及用于计算帧 k 的 $MQP_p(k)$。

[0036] $\Phi_{R(p)}$代表通路 p 中的参考屏蔽强度。该参考屏蔽强度用于计算帧 k 的 $MQP_p(k)$，并且其由本发明的多通路编码器在第二阶段中调整以达到目标比特率。

[0037] $\Phi_{MB}(k,m)$ 代表帧 k 中具有索引号为 m 的宏块的屏蔽强度。屏蔽强度 $\Phi_{MB}(k,m)$ 为该宏块复杂度的度量，并且在一些实施例中，其被用于确定视觉编码人工因素 / 噪声将如何呈现以及用于计算 $MQP_{MB(p)}(k,m)$。AMQPp 代表通路 p 中的帧之上的平均屏蔽 QP。在一些实施例中，该值作为通路 p 中的所有帧之上的平均 $MQP_p(k)$ 计算。

[0038] II. 概述

[0039] 本发明的一些实施例提供了实现以给定比特率编码帧序列的最佳视觉质量的编码方法。在一些实施例中，该方法使用为每一个宏块分配量化参数 QP 的视觉掩蔽过程。这种分配基于图像或视频帧中较亮或空间上较复杂区域中的编码人工因素 / 噪声不如较暗或平面区域中的编码人工因素 / 噪声明显的认识。

[0040] 在一些实施例中，这种视觉掩蔽过程作为发明的多通路编码过程的部分执行。为了使最终编码比特流达到目标比特率，这种编码过程调整标称量化参数并通过参考屏蔽强度参数 Φ_R控制视觉掩蔽过程。如以下的进一步描述，调整标称量化参数和控制屏蔽算法调整每幅图片(即，通常是视频编码方案中的每个帧)和每幅图片内的每个宏块的 QP 值。

[0041] 在一些实施例中，多通路编码过程全局调整整个序列的标称 QP 和 Φ_R。在其他实施例中，这个过程将视频序列划分为片段，利用标称 QP 和 Φ_R调整每个片段。下面的描述涉及其上应用了多通路编码处理的帧序列。普通技术人员将意识到在一些实施例中这个序列包括整个序列，而在其他实施例中其仅包括序列的一个片段。

[0042] 在一些实施例中，本方法具有三个编码阶段。这三个阶段为：(1)在通路 0 中执行的初始分析阶段，(2) 在通路 1 到通路 N_1中执行的第一搜索阶段，以及(3) 在通路 N_1+1 到 N_1+N_2中执行的第二搜索阶段。

[0043] 在初始分析阶段中(即，在通路 0 期间)，本方法识别用于标称 QP ($QP_{Nom(1)}$，将在编码的通路 1 中使用)的初始值。在初始分析阶段期间，该方法还识别参考屏蔽强度 Φ_R的值，

它在第一搜索阶段中的所有通路中使用。

[0044] 在第一搜索阶段中，本方法执行编码过程的 N_1 迭代（即，N_1 通路）。在通路 p 中对每一个帧 k，该过程通过使用特定量化参数 MQP_p（k）和帧 k 内的各个宏块 m 的特定量化参数 $MQP_{MB(p)}$（k，m）编码该帧，在此 $MQP_{MB(p)}$（k，m）是利用 MQP_p（k）计算的。

[0045] 在第一搜索阶段中，量化参数 MQP_p（k）在通路之间变化，因为其是由在通路之间变化的标称量化参数 $QP_{Nom(p)}$ 得到的。换言之，在第一搜索阶段期间每个通路 p 的结束时，该过程计算用于通路 p+1 的标称 $QP_{Nom(p+1)}$。在一些实施例中，标称 $QP_{Nom(p+1)}$ 是基于来自之前的通路的标称 QP 值和比特率错误。在其他的实施例中，标称 $QP_{Nom(p+1)}$ 值在第二搜索阶段中的每个通路的结束时不同地计算。

[0046] 在第二搜索阶段中，本方法执行编码过程的 N_2 迭代（即，N_2 通路）。正如在第一搜索阶段中的那样，该过程通过使用特定量化参数 MQP_p（k）和帧 k 内的各个宏块 m 的特定量化参数 $MQP_{MB(p)}$（k，m）在每个通路 p 期间编码每个帧 k，在此由 MQP_p（k）得到 $MQP_{MB(p)}$（k，m）。

[0047] 同样，正如在第一搜索阶段中的那样，量化参数 MQP_p（k）在通路间变化。然而，在第二搜索阶段期间，这个参数改变是由于其是利用在通路之间变化的参考屏蔽强度 $\Phi_{R(p)}$ 计算的。在一些实施例中，参考屏蔽强度 $\Phi_{R(p)}$ 是基于来自之前通路的比特率中的错误和 Φ_R 值计算的。在其他的实施例中，该参考屏蔽强度在第二搜索阶段中的每个通路的结束时计算为不同的值。

[0048] 尽管是结合视觉掩蔽过程描述了多通路编码过程，本领域的普通技术人员将意识到的是编码器无需同时一起使用这些两种处理过程。例如，在一些实施例中，通过忽略 Φ_R 并省略以上所述的第二搜索阶段，多通路编码过程被用于编码给定目标比特率附近的比特流而无需视觉掩蔽。

[0049] 在本申请的第 III 和 IV 部分进一步描述了视觉掩蔽和多通路编码过程。

[0050] III. 视觉掩蔽

[0051] 给定一个标称量化参数，视觉掩蔽处理首先利用参考屏蔽强度（Φ_R）和该帧屏蔽强度（Φ_F）计算每个帧的屏蔽帧量化参数（MQP）。该过程接着基于该帧和宏块级屏蔽强度（Φ_F 和 Φ_{MB}）计算每个宏块的屏蔽宏块量化参数（MQP_{MB}）。当在多通路编码过程中应用视觉掩蔽处理时，一些实施例中的参考屏蔽强度（Φ_R）如上所述以及以下进一步的描述在第一编码通路中被识别。

[0052] A. 计算帧级屏蔽强度

[0053] 1. 第一种方法

[0054] 为了计算帧级屏蔽强度 $\Phi_F(k)$，一些实施例使用以下公式（A）：

[0055] $\Phi_F(k) = C*power(E*avgFrameLuma(k), \beta)*power(D*avgFrameSAD(k), \alpha_F)$，(A)

[0056] 其中：

[0057] ● avgFrameLuma（k）为利用 bxb 区域计算的帧 k 中的平均像素强度，其中 b 为大于或等于 1 的整数（例如，b=1 或 b=4）；

[0058] ● avgFrameSAD（k）为帧 k 内所有宏块的 MbSAD（k，m）的平均值；

[0059] ● MbSAD（k，m）为由函数 Calc4x4MeanRemovedSAD（4x4_block_pixel_value）给出的具有索引为 m 的宏块中所有 4x4 块的值的总和；

[0060] ● α_F, C, D, 和 E 为常数和 / 或根据本地统计而调整 ; 以及

[0061] ● power (a, b) 意为 a^b。

[0062] 用于函数 Calc4x4MeanRemovedSAD 的伪码如下 :

[0063]

```
Calc4x4MeanRemovedSAD(4x4_block_pixel_values)
{
        calculate the mean of pixel values in the given 4x4 block;
        subtract the mean from pixel values and compute their absolute values;
        sum the absolute values obtained in the previous step;
        return the sum;
}
```

[0064] 2. 第二种方法

[0065] 其他的实施例以不同的方式计算帧级屏蔽强度。例如, 上述的公

[0066] 式 (A) 基本如下所示计算帧屏蔽强度 :

[0067] $\Phi_F(k)$ = C*power(E*Brightness_Attribute, exponent0)*

[0068] power(scalar*Spatial_Activity_Attribute, exponentl).

[0069] 在公式 (A) 中, 帧的 Brightness_Attribute 等于 avgFrameLuma (k), 而 Spatial_Activity_Attribute 等于 avgFrameSAD (k), 其是帧内的所有宏块的平均宏块 SAD (MbSAD (k, m)) 值, 在此平均宏块 SAD 等于宏块内所有 4x4 块的平均移除 4x4 像素变更 (如由 Calc4x4MeanRemovedSAD 给出) 的绝对值之和。该 Spatial_Activity_Attribute 度量了正被编码的帧之内的像素区域中的空间修正的数量。

[0070] 其他的实施例将活动度量扩展到包含穿过许多连续帧的像素区域中的时间修正的数量。特别的, 这些实施例如下所示计算帧屏蔽强度 :

[0071] $\Phi_F(k)$ = C*power(E*Brightness_Attribute, expouent0)*

[0072] power(scalar*Activity_Attribute, exponentl) (B)

[0073] 在这个公式中, Activity_Attribute 由以下公式 (C) 给出 :

[0074] E*power(F*Temporal_Activity_Attribuc, exponent_delta) (C)

[0075] 在一些实施例中, Temporal_Activity_Attribute 量化了能够忍受 (即, 屏蔽) 由于帧之间的移动而引起失真的数量。在这些实施例的一些中, 帧的 Temporal_Activity_Attribute 等于该帧内所定义的像素区域的移动补偿错误信号的绝对值之和的常数倍。在另外一些实施例中, Temporal_Activity_Attribute 由以下公式 (D) 提供 :

[0076] **Temporal_Activity_Attribute =**

[0077] $$\sum_{j=-1}^{-N}\left(W_j \bullet avgFrameSAD(j)\right)+\sum_{j=1}^{M}\left(W_j \bullet avgFrameSAD(j)\right)+W_0 \bullet avgFrameSAD(0) \qquad (D)$$

[0078] 在公式 (D) 中, "avgFrameSAD" 代表 (如上所述) 帧内的平均宏块 SAD (MbSAD (k, m)) 值, avgFrameSAD (0) 为当前帧的 avgFrameSAD, 并且负的 j 指向当前帧之前的时间实例, 而正的 j 指向当前帧之后的时间实例。由此, avgFrameSAD (j=-2) 表示当前帧之前的两个帧的平均帧 SAD, avgFrameSAD (j=3) 表示当前帧之后的三个帧的平均帧 SAD。

[0079] 同样, 在公式 (D) 中, 变量 N 和 M 分别指当前帧之前和之后的帧的数量。代替简单的基于特定数量的帧选择值 N 和 M, 一些实施例基于当前时间帧的时间的之前或之后特定

时间周期计算值 N 和 M。将移动屏蔽与空间持续时间相关联比将移动屏蔽与一组数量的帧相关联更具优势。这是因为将移动屏蔽与时间周期相关联直接符合观察者基于时间的视觉感觉。另一方面，将这样的屏蔽与帧的数量相关联由于不同的显示装置以不同帧速率呈现视频而要忍受可变的显示持续时间。

[0080] 在公式(D)中，“W”代指权重因数，在一些实施例中，当帧 j 进一步离开当前帧时其会减少。同样，在这个公式中，第一求和表示能够在当前帧之前屏蔽的移动数量。第二求和表示能够在当前正之后屏蔽的移动数量，而最后的表达式(avgFrameSAD (0)) 表示当前帧的帧 SAD。

[0081] 在一些实施例中，权重因数被调整以说明场景变化。例如，一些实施例解决先行范围内(即，在 M 帧内) 即将来临的场景变化，但在场景变化之后没有任何帧。例如，这些实施例可以设置场景变化之后的先行范围内的帧的权重因数为零。同样，一些实施例不解决向后看范围内(即，在 N 帧之内)先于或位于场景变化的帧。例如，这些实施例可以设置涉及前面场景或落到先前场景变化之前的向后看范围内的帧的权重因数为零。

[0082] 3. 第二方法的变异

[0083] a) 限制过去帧和将来帧对 Temporal_Activity_Attribute 的影响

[0084] 以上的公式(D) 基本上从以下条件表述 Temporal_Activity_Attribute：

[0085] Temporal_Activity_Attribute = Past_Frame_Activity+Future_Frame_Activity+

[0086] Current_Frame_Activity,

[0087] 在此 Past_Frame_Activity (PFA)等于 $\sum_{i=1}^{N}(W_i \bullet avgFrameSAD(i))$，Future_Frame_Activity (FFA) 等于 $\sum_{j=1}^{M}(W_j \bullet avgFrameSAD(j))$，而 Current_Frame_Activity (CFA) 等于 avgFrameSAD (current)。

[0088] 一些实施例修改 Temporal_Activity_Attribute 的计算以便 Past_Frame_Activity 和 Future_Frame_Activity 均不会过度控制 Temporal_Activity_Attribute 的值。例如，一些实施例初始定义 PFA 等于 $\sum_{i=1}^{N}(W_i \bullet avgFrameSAD(i))$，而 FFA 等于 $\sum_{j=1}^{M}(W_j \bullet avgFrameSAD(j))$。

[0089] 这些实施例接着判断 PFA 是否大于标量时间 FFA。如果是的话，这些实施例就将 PFA 设置为等于 PFA 上限值(例如，标量时间 FFA)。除了设置 PFA 等于 PFA 上限值，一些实施例可以执行将 FFA 设置为零以及将 CFA 设置为零的组合设置。其他的实施例可以将 PFA 和 CFA 之一或二者设置为 PFA、CFA、以及 FFA 的加权组合。

[0090] 与之类似，在基于加权总和初始定义了 PFA 和 FFA 值之后，一些实施例还判断 FFA 值是否大于标量时间 PFA。如果是的话，这些实施例就将 FFA 设置为等于 FFA 上限值(例如，标量时间 PFA)。除了设置 FFA 等于 FFA 上限值，一些实施例可以执行将 PFA 设置为零以及将 CFA 设置为零的组合设置。其他的实施例可以将 FFA 和 CFA 之一或二者设置为 FFA、CFA、

以及 PFA 的加权组合。

[0091] PFA 和 FFA 值的潜在后续调整(在基于加权总和对这些值进行初始估算之后)防止了这些值的任一个对 Temporal_Activity_Attribute 的过度控制。

[0092] b) 限制 Spatial_Activity_Attribute 和

[0093] Temporal_Activity_Attribute 对 Activity_Attribute 的影响

[0094] 以上的公式(C)基本从以下条件表述 Activity_Attribute :

[0095] Activity_Attribute = Spatial_Activity+Temporal_Activity

[0096] 其中, Spatial_Activity 等于 scalar* (scalar*Spatial_Activity_Attribute)$^{\beta}$, 而 Temporal_Activity 等于 scalar* (scalar*Temporal_Activity_Attribute)$^{\Delta}$。

[0097] 一些实施例修改 Activity_Attribute 的计算以便 Spatial_Activity 和 Temporal_Activity 任一个都不会过度控制 Activity_Attribute 的值。例如,一些实施例初始定义 Spatial_Activity(SA)等于 scalar*(scalar*Spatial_Activity_Attribute)$^{\beta}$, 以及定义 Temporal_Activity(TA)等于 scalar*(scalar*Temporal_Activity_Attribute)$^{\Delta}$。

[0098] 这些实施例接着判断 SA 是否大于标量时间 TA。如 果是的话,这些实施例就将 SA 设置为等于 SA 上限值(例如,标量时间 TA)。除了设置 SA 等于 SA 上限的这种情况之外,一些实施例还可以将 TA 值设置为零或设置为 TA 和 SA 的加权组合。

[0099] 与之类似,在基于指数方程初始定义 SA 和 TA 值之后,一些实施例还判断 TA 值是否大于标量时间 SA。如果是的话,这些实施例就将 TA 设置为等于 TA 上限值(例如,标量时间 SA)。除了设置 TA 等于 TA 上限的这种情况之外,一些实施例还可以将 SA 值设置为零或设置为 SA 和 TA 的加权组合。

[0100] SA 和 TA 值的潜在后续调整(在基于指数方程对这些值进行初始计算之后)防止了这些值之一对 Activity_Attribute 的过度控制。

[0101] B. 计算宏块级屏蔽强度

[0102] 1. 第一种方法

[0103] 在一些实施例中,宏块级屏蔽强度 $\Phi_{MB}(k,m)$ 如下计算 :

[0104] $\Phi_{MB}(k,m)$ = A*power(C*avgMbLuma(k,m), β)*power(B*MbSAD(k,

[0105] m), α_{MB}), (F)

[0106] 其中 :

[0107] avgMbLuma (k,m) 为帧 k、宏块 m 内的平均像素强度 ;

[0108] α_{MB}、β、A、B、和 C 为常数和 / 或适合于本地统计。

[0109] 2. 第二种方法

[0110] 以上所述的公式(F)基本上如下计算宏块屏蔽强度 :

[0111] $\Phi_{MB}(k,m)$ = D*power(E*Mb_Brightness_Attribute, exponent0)*

[0112] power(scalar*Mb_Spatial_Activity_Attribute, cxponentl)

[0113] 在公式(F)中,宏块的 Mb_Brightness_Attribute 等于 avgMbLuma (k, m),而 Mb_Spatial_Activity_Attribute 等于 avgMbSAD (k)。该 Mb_Spatial_Activity_Attribute 度量了正被编码的宏块内的像素区域中的空间修正的数量。

[0114] 正如在帧屏蔽强度的情况下一样,一些实施例可以扩展宏块屏蔽强度中的活动度

量以包含穿过许多连续帧的像素区域中的时间修正的数量。特别的，这些实施例将如下所示计算宏块屏蔽强度：

[0115] $\Phi_{MB}(k,m) = D*power(E*Mb\_Brightness\_Attribute, exponent0)*$

[0116] power(scalar*Mb_Activity_Attribute, exponent1), (G)

[0117] 其中 Mb_Activity_Attribute 由以下公式(H)给出：

[0118] Mb_Activity_Attribute = F*power(D*Mb_Spatial_Activity_Attribute, exponent_beta)+

[0119] G*power(F*Mb_Temporal_Activity_Attribue, exponent_dclta)(H)

[0120] 宏块的 Mb_Temporal_Activity_Attribute 的计算可以与以上所述帧的 Mb_Temporal_Activity_Attribute 的计算相类似。例如，在这些实施例的一些中，Mb_Temporal_Activity_Attribute 由以下公式(I)提供：

[0121] **Mb_Temporal_Activity_Attribute =**

[0122] $$\sum_{i=1}^{N}\left(W_i \bullet MbSAD(i,m)\right) + \sum_{j=1}^{M}\left(W_j \bullet MbSAD(j,m)\right) + MbSAD(m) \qquad (I)$$

[0123] 公式(I)中的变量在第 III 部分中定义。在公式(F)中，帧 I 或 j 中的宏块 m 可以是如与当前帧中宏块 m 的相同位置中的宏块，或可以是初始预测为对应当前帧中的宏块 m 的帧 i 或 j 中的宏块。

[0124] 由公式(I)提供的 Mb_Temporal_Activity_Attribute 可以以与公式(D)所提供的帧 Temporal_Activity_Attribute 的修改(在以上第 III.A.3 部分中所讨论的)相类似的方式进行修改。特别的，可以修改由公式(I)提供的 Mb_Temporal_Activity_Attribute 以限制过去和将来帧中的宏块的过度影响。

[0125] 类似的，由公式(H)所提供的 Mb_Activity_Attribute 可以以与公式(C)所提供的帧 Activity_Attribute 的修改(在以上第 III.A.3 部分中所讨论的)相类似的方式进行修改。特别的，可以修改由公式(H)提供的 Mb_Activity_Attribute 以限制 Mb_Spatial_Activity_Attribute 和 Mb_Temporal_Activity_Attribute 的过度影响。

[0126] C. 计算屏蔽的 QP 值

[0127] 基于屏蔽强度(Φ_F和 Φ_{MB})值和参考屏蔽强度(Φ_R)值，视觉掩蔽处理可通过使用两个函数 CalcMQP 和 CalcMQPforMB 计算帧级和宏块级的屏蔽 QP 值。这两个函数的伪码如下：

[0128]

```
CalcMQP(nominalQP, φR, φF(k), maxQPFrameAdjustment)
{
        QPFrameAdjustment = βF *(φF(k) - φR) / φR;
        clip QPFrameAdjustment to lie within [minQPFrameAdjustment,,
maxQPFrameAdjustment];
        maskedQPofFrame = nominalQP + QPFrameAdjustment;
        clip maskedQPofFrame to lie in the admissible range;
        return maskedQPofFrame (for frame k);
}

CalcMQPforMB(maskedQPofFrame, φF(k), φMB(k, m),
maxQPMacroblockAdjustment)
{
        if (φF(k) >T)          where T is a suitably chosen threshold
                QPMacroblockAdjustment= βMB * (φMB(k, m) - φF(k)) /
            φF(k);
        else
                QPMacroblockAdjustment= 0;
        clip QPMacroblockAdjustment so that it lies within
[minQPMacroblockAdjustment, maxQPMacroblockAdjustment ];
        maskedQPofMacroblock = maskedQPofFrame +
QPMacroblockAdjustment;
        clip maskedQPofMacroblock so that it lies within the valid QP value
range;
        return maskedQPofMacroblock;
}
```

[0129] 在以上函数中，β_F和 β_{MB}可以是预先设定的常数或适合于本地统计。

[0130] IV. 多通路编码

[0131] 图 1 展示了过程 100，其概念性地举例说明了本发明一些实施例的多通路编码方法。正如该图所示，过程 100 有三个阶段，在以下三个部分中描述。

[0132] A. 分析和初始 QP 选择

[0133] 如图 1 所示，过程 100 最初在多通路编码过程的初始分析阶段（即，在通路 0 期间）计算参考屏蔽强度（$\Phi_{R(1)}$）的初始值和标称量化参数（$QP_{Nom(1)}$）的初始值（步骤 105）。初始参考强度（$\Phi_{R(1)}$）在第一搜索阶段期间使用，而初始标称量化参数（$QP_{Nom(1)}$）在第一搜索阶段的第一通路期间使用（即，多通路编码过程的通路 1 期间）。

[0134] 在通路 0 之初，$\Phi_{R(0)}$可以是某些任意值或基于实验结果选择的值（例如，Φ_R值的典型范围的中间值）。在序列的分析期间，针对每帧计算屏蔽强度 $\Phi_F(k)$，然后在通路 0 的结束设置参考屏蔽强度 $\Phi_{R(1)}$等于 avg（$\Phi_F(k)$）。对参考屏蔽强度 Φ_R的其他判定也是可能的。例如，它可以计算作为值 $\Phi_F(k)$ 的中间值或其他算术函数，例如值 $\Phi_F(k)$ 的加权平均值。

[0135] 存在使用变化的复杂度进行初始 QP 选择的几种方法。例如，初始标称 QP 可以选择为如任意值（例如 26）。可选的，可以基于编码实验选择已知的值以针对目标比特率生成可接受的质量。

[0136] 初始标称 QP 值也可以基于空间解决方案、帧速率、空间 / 时间复杂度、以及目标比

特率从查询表中选择。在一些实施例中，该初始标称 QP 值使用依赖于这些参数中的每一个的距离度量从表中选择，或者它可以利用这些参数的加权距离度量选择。

[0137] 该初始标称 QP 值还可以如它们在使用速率控制器快速编码期间（无屏蔽）所选择的那样设置为帧 QP 值的调整平均值，其中该平均值已经基于通路 0 的比特率百分比速率误差 E_0 调整。类似的，初始标称 QP 也可以设置为帧 QP 值的加权调整平均值，其中每个帧的权重由没有编码为跳跃宏块的宏块在这个帧中的百分比确定。可选的，初始标称 QP 可以如它们在使用速率控制器快速编码期间（带屏蔽）所选择的那样设置为帧 QP 值的调整平均值或调整加权平均值，同时考虑了参考屏蔽强度从 $\Phi_{R(0)}$ 改变到 $\Phi_{R(1)}$ 的效应。

[0138] B. 快速搜索阶段：标称 QP 调整

[0139] 步骤 105 之后，多通路编码过程 100 进入第一搜索阶段。在第一搜索阶段，过程 100 执行序列的 N_1 编码，其中 N_1 代表通过第一搜索阶段的通路数。在第一阶段的每个通路期间，该过程使用具有恒定参考屏蔽强度的变动标称量化参数。

[0140] 特别的，在第一级搜索阶段的每个通路 p 期间，过程 100 计算（步骤 107）每个帧 k 的特定量化参数 $MQP_p(k)$，以及计算帧 k 内的每个单独宏块 m 的特定量化参数 $MQP_{MB(p)}(k, m)$。给定标称量化参数 $QP_{Nom(p)}$ 和参考屏蔽强度 $\Phi_{R(p)}$ 的参数 $MQP_p(k)$ 和 $MQP_{MB(p)}(k, m)$ 的计算在第 III 部分中描述（其中 $MQP_p(k)$ 和 $MQP_{MB(p)}(k, m)$ 是通过利用函数 CalcMQP 和 CalcMQPforMB 计算的，这在以上的部分 III 中描述）。在通过步骤 107 的第一通路（即，通路 1）中，标称量化参数和第一阶段参考屏蔽强度为参数 $QP_{Nom(1)}$ 和参考屏蔽强度 $\Phi_{R(1)}$，它们在初步分析阶段 105 期间计算。

[0141] 步骤 107 之后，该过程基于在步骤 107 计算的量化参数值编码该序列（步骤 110）。接下来，编码过程 100 判断其是否应该结束（步骤 115）。不同的实施例具有结束整个编码过程的不同条件。完全结束多通路编码过程的退出条件的例子包括：

[0142] ● $|Ep| < \varepsilon$，其中 ε 为最终比特率中的误差容许范围。

[0143] ● $QP_{Nom(p)}$ 为 QP 值有效范围的上边界和下边界。

[0144] ●通路的数量超过了允许的最大通路数 P_{MAX}。

[0145] 一些实施例可能使用所有的这些退出条件，而其他实施例可能仅使用它们中的一些。然而其他的实施例可能使用其他的用于结束编码过程的退出条件。

[0146] 当多通路编码过程决定结束（步骤 115），过程 100 省略第二搜索阶段并转移到步骤 145。在步骤 145，该过程保存来自最后的通路 p 的比特流作为最终结果，然后结束。

[0147] 另一方面，当该过程确定（步骤 115）不能结束，其接着确定（步骤 120）是否应当结束第一搜索阶段。同样，不同的实施例具有结束第一搜索阶段的不同条件。结束多通路编码过程的第一搜索阶段的退出条件的例子包括：

[0148] ● $QP_{Nom(p+1)}$ 与 $QP_{Nom(q)}$ 相同，并且 $q \leq p$，（在此情况下，比特率中的误差不能再通过修改标称 QP 进一步降低）。

[0149] ● $|Ep| < \varepsilon_C, \varepsilon_C > \varepsilon$，其中 ε_C 为第一搜索阶段的比特率中的误差允许范围。

[0150] ●通路的数量已超过了 P_1，其中 P_1 小于 P_{MAX}。

[0151] ●通路的数量已超过了 P_2，其小于 P_1，并且 $|Ep| < \varepsilon_2, \varepsilon_2 > \varepsilon_C$。

[0152] 一些实施例可能使用所有这些退出条件，而其实施例可能仅使用它们中的一些。然而其他的实施例可能使用其他的用于结束第一搜索阶段的退出条件。

[0153] 当多通路编码过程决定(步骤 120)结束第一搜索阶段时,过程 100 继续到第二搜索阶段,其在以下部分中描述。另一方面,当过程确定(步骤 120)其不应结束第一搜索阶段时,它就在第一搜索阶段中更新(步骤 125)下一通路的标称 QP (即,定义 $QP_{Nom\ (p+1)}$)。在一些实施例中,标称 $QP_{Nom\ (p+1)}$ 如下更新。在通路 1 的结束,这些实施例定义 :

[0154] $QP_{Nom\ (p+1)}=QP_{Nom\ (p)}+x\,E_p$,

[0155] 其中 x 为常数。在从通路 2 到通路 N_1 的每个通路的结束,这些实施例于是定义 :

[0156] $QP_{Nom\ (p+1)}=$ InterpExtrap (0, E_{q1}, E_{q2}, $QP_{Nom\ (q1)}$, $QP_{Nom\ (q2)}$),

[0157] 其中 InterpExtrap 为如下进一步描述的函数。同样,在以上公式中,q1 和 q2 为对应具有直到通路 p 的所有通路中比特误差最低的通路数,而且 q1、q2 和 p 具有以下关系 :

[0158] $1 \leqslant q_1 < q_2 \leqslant p$

[0159] 以下为 InterpExtrap 函数的伪码。注意,如果 x 不在 x1 和 x2 之间,这个函数就为外推函数。否则,其为插值函数。

[0160]

```
InterpExtrap(x, x1, x2, y1, y2)
{
        if (x2 != x1) y = y1 + (x – x1) * (y2 - y1)/(x2 - x1);
        else  y = y1;
        return y;
}
```

[0161] 标称 QP 值通常四舍五入为整数值并限制在 QP 值的有效范围之内。本领域普通技术人员将认识到其他实施例可以以不同于以上所述的方法来计算标称 $QP_{Nom\ (p+1)}$。

[0162] 在步骤 125 之后,该过程转移回到步骤 107 以开始下一通路(即, p:=p+1),并且对于这个通路,针对当前通路 p 计算每个帧 k 的特定量化参数 $MQP_p(k)$,以及帧 k 内的每个单独的宏块 m 的特定量化参数 $MQP_{MB\ (p)}(k,m)$(步骤 107)。接下来,该过程基于这些新近计算的量化参数编码帧序列(步骤 110)。该过程接着由步骤 110 转移步骤 115,其已在上面描述。

[0163] C. 第二搜索阶段 :参考屏蔽强度调整

[0164] 当过程 100 确定其应当结束第一搜索阶段时(步骤 120),它转移到步骤 130。在第二搜索阶段,过程 100 执行序列的 N_2 编码,在此 N_2 代表通过第二搜索阶段的通路数。在每个通路期间,该过程使用相同的标称量化参数和变化的参考屏蔽强度。

[0165] 在步骤 130,过程 100 计算下一通路,即通路 p+1,其为通路 N_1+1,的参考屏蔽强度 $\Phi_{R(p+1)}$。在通路 N_1+1 中,过程 100 在步骤 135 中编码帧序列。不同的实施例以不同的方式在通路 p 的结束计算参考屏蔽强度 $\Phi_{R(p+1)}$ (步骤 130)。以下描述了两种可选的实现方法。

[0166] 一些实施例基于来自先前的通路的比特率中的误差和 Φ_R 的值计算参考屏蔽强度 $\Phi_{R(p)}$。例如,在通路 N_1 的结束,一些实施例定义 :

[0167] $\Phi_{R(N1+1)}=\Phi_{R(N1)}+\Phi_{R(N1)}\times Konst\times E_{N1}$.

[0168] 在通路 N1+m 的结束处,此处 m 为大于 1 的整数,一些实施例定义

[0169] $\Phi_{R(N1+m)}=$ InterpExtrap(0, E_{N1+m-2}, E_{N1+m-1}, $\Phi_{R(N1+m-2)}$, $\Phi_{R(N1+m-1)}$)

[0170] 或者,一些实施例定义 :

[0171] $\Phi_{R(N1+m)}=$ InterpExtrap(0, $E_{N1+m-q2}$, $E_{N1+m-q1}$, $\Phi_{R(N1+m-q2)}$, $\Phi_{R(N1+m-q1)}$)

[0172] 其中 q1 和 q2 为之前给出最优误差的通路。

[0173] 其他实施例通过利用 AMQP 在第二搜索阶段在每个通路的结束计算参考屏蔽强度，其在第 I 部分中定义。以下将参考函数 GetAvgMaskedQP 的伪码描述给定标称 QP 和 Φ_R 的一些值用于计算 AMQP 的一种方式：

[0174]

```
GetAvgMaskedQP(nominalQP, φR)
{
    sum=0;
    for(k=0;k<numframes;k++) {
        MQP(k) = maskedQP for frame k calculated using
    CalcMQP(nominalQP, φR, φF(k), maxQPFrameAdjustment); // see
    above
        sum += MQP(k);
    }
    return sum/numframes;
}
```

[0175] 一些使用 AMQP 的实施例基于来自之前通路的比特率中的误差和 AMQP 的值计算通路 p+1 所期望的 AMQP。对应于这个 AMQP 的 $\Phi_{R(p+1)}$ 于是通过由函数 Search（$AMQP_{(p+1)}$，$\Phi_{R(p)}$）给出的搜索过程而找到，该函数的伪码在本部分的最后给出。

[0176] 例如，一些实施例在通路 N_1 的结束计算 $AMQP_{N1+1}$，其中：

[0177] $AMQP_{N1+1}$ = InterpExtrap(0, E_{N1-1}, E_{N1}, $AMQP_{N1-1}$, $AMQP_{N1}$)，when $N_1 > 1$，

[0178] 并且

[0179] $AMQP_{N1+1} = AMQP_{N1}$，when $N_1 = 1$，

[0180] 这些实施例于是定义：

[0181] $\Phi_{R(N1+1)}$ = Search($AMQP_{N1+1}$, $\Phi_{R(N1)}$)

[0182] 在通路 N_1+m（其中 m 为大于 1 的整数）的结束，一些实施例定义：

[0183] $AMQP_{N1+m}$ = InterpExtrap(0, E_{N1+m-2}, E_{N1+m-1}, $AMQP_{N1+m-2}$, $AMQP_{N1+m-1}$)，

[0184] 以及

[0185] $\Phi_{R(N1+m)}$ = Search($AMQP_{N1+m}$, $\Phi_{R(N1+m-1)}$)

[0186] 给定所期望的 AMQP 和 Φ_R 的一些默认值，对应于所期望的 AMQP 的 Φ_R 可以利用 Search 函数找到，该函数在一些实施例中具有以下伪码：

[0187]

```
Search(AMQP, φR)
{
        interpolateSuccess=True;      //until set otherwise

        refLumaSad0=refLumaSad1=refLumaSadx=φR;
        errorInAvgMaskedQp = GetAvgMaskedQp(nominalQp, refLumaSadx) -
        AMQP;
        if(errorInAvgMaskedQp>0){
                ntimes=0;
                do{
                        ntimes++;
                        refLumaSad0 = (refLumaSad0 * 1.1);
                        errorInAvgMaskedQp = GetAvgMaskedQp(nominalQp,refLumaSad0) -
                        amqp;
                }while(errorInAvgMaskedQp>0 && ntimes<10);
                if(ntimes>=10)interpolateSuccess=False;
        }
        else{  //errorInAvgMaskedQp<0
                ntimes=0;
                do{
                        ntimes++;
                        refLumaSad1 = (refLumaSad1 * 0.9);
                        errorInAvgMaskedQp = GetAvgMaskedQp(nominalQp,refLumaSad1) -
                        amqp;
                }while(errorInAvgMaskedQp<0 && ntimes<10);
                if(ntimes>=10)interpolateSuccess=False;
        }
        ntimes=0;
        do{
                ntimes++;
                refLumaSadx = (refLumaSad0+refLumaSad1)/2; //simple successive
                approximation
                errorInAvgMaskedQp = GetAvgMaskedQp(nominalQp,refLumaSadx) - AMQP;
                  if(errorInAvgMaskedQp>0)refLumaSad1=refLumaSadx;
                  else refLumaSad0=refLumaSadx;
          }while( ABS (errorInAvgMaskedQp) > 0.05 && ntimes<12 );
          if(ntimes>=12)interpolateSuccess=False;
          }
          if (interpolateSuccess) return   refLumaSadx;
          else   return  φR;
}
```

[0188] 在以上伪码中，数字 10、12 和 0.05 可以使用适当选择的阈值代替。

[0189] 在通过编码帧序列计算了下一通路（通路 p+1）的参考屏蔽强度之后，过程 100 就转移到步骤 132 并开始下一个通路（即，p:=p+1）。在每个编码通路 p 期间，对于每个帧 k 和每个宏块 m，该过程计算每个帧 k 的特定量化参数 $MQP_p(k)$ 以及帧 k 中的单独宏块 m 的特定量化参数 $MQP_{MB(p)}(k, m)$（步骤 132）。给定标称量化参数 $QP_{Nom(p)}$ 和参考屏蔽强度 $\Phi_{R(p)}$ 的参数 $MQP_p(k)$ 和 $MQP_{MB(p)}(k,m)$ 的计算在第 III 部分中描述（其中 $MQP_p(k)$ 和 $MQP_{MB(p)}(k, m)$ 通过利用函数 CalcMQP 和 CalcMQPforMB 计算，这在以上第 III 部分中描述）。在通过步

骤 132 的第一通路期间，参考屏蔽强度正是在步骤 130 处计算的数值。同样，在第二搜索阶段期间，标称 QP 在整个第二搜索阶段保持为常数。在一些实施例中，第二搜索阶段之内的标称 QP 为第一搜索阶段期间由最优编码解决方案（即，在具有最低比特率误差的编码解决方案中）所得到的标称 QP。

[0190] 在步骤 132 之后，该过程利用在步骤 130 处计算的量化参数编码帧序列（步骤 135）。在步骤 135 之后，该过程确定（步骤 140）是否应当结束第二搜索阶段。不同的实施例使用不同的条件用于在通路 p 的结束处结束第一搜索阶段。这种条件的例子为：

[0191] ● $|Ep|<\varepsilon$，其中 ε 为最终比特率中的误差容许范围。

[0192] ●通路的数量超过了所允许的最大通路数 P_{MAX}。

[0193] 一些实施例可能使用所有的这些退出条件，而其他实施例可能仅使用它们中的一些。然而其他的实施例可能使用其他的用于结束第一搜索阶段的退出条件。

[0194] 当过程 100 确定（步骤 140）不应当结束第二搜索阶段时，其返回到步骤 130 以重新计算下一编码通路的参考屏蔽强度。该过程从步骤 130 转移到步骤 132 以计算量化参数，然后转移到步骤 135 以通过利用新近计算的量化参数编码视频序列。

[0195] 另一方面，当该过程决定（步骤 140）结束第二搜索阶段时，则其转移到步骤 145。在步骤 145，过程 100 保存来自最后一个通路 p 的比特流作为最终结果，然后就结束。

[0196] V. 解码器输入缓冲区下溢控制

[0197] 本发明的一些实施例提供对目标比特率检查视频序列的各种编码的多通路编码过程，为了识别有关由解码器使用的输入缓冲区的使用的最优编码方案。在一些实施例中，这种多通路过程遵循图 1 的多通路编码过程 100。

[0198] 由于各种因素的变化，例如已编码图像的大小、解码器接收已编码数据所使用的速度、解码器缓冲区的大小、解码过程的速度等方面的变动，解码器输入缓冲区（“解码器缓冲区”）的使用在解码已编码图片序列（例如，帧）的过程中在一定程度上变动。

[0199] 解码器缓冲区下溢在图像已经完全到达解码器端之前解码器准备解码下一图像的情况下颇为重要。一些实施例的多通路编码器模拟解码器缓冲区并重新编码序列中所选择的片段以防止解码器缓冲区下溢。

[0200] 图 2 概念性举例说明了本发明一些实施例的编码系统 200。该系统包括解码器 205 和编码器 210。在该图中，编码器 210 具有多个使其能够模拟解码器 205 的类似组件的操作的组件。

[0201] 特别的，解码器 205 具有输入缓冲区 215、解码过程 220、以及输出缓冲区 225。解码器 210 通过维护模拟解码器输入缓冲区 230、模拟解码过程 235、以及模拟解码器输出缓冲区 240 来模拟这些模块。为了不妨碍本发明的描述，简化图 2 以将解码过程 220 和编码过程 245 显示为单个的块。同样，在一些实施例中，没有利用模拟解码过程 235 和模拟解码器输出缓冲区 240 用于缓冲区下溢管理，从而在本图中仅出于举例而示意。

[0202] 解码器维护输入缓冲区 215 以消除输入的编码图像的速率和到达时间的变化。如果解码器用完了数据（下溢）或填满了输入缓冲区（上溢）的话，就会有例如图片解码中断或输入的数据被丢弃的可视的解码中断。这两种情况都是不期望的。

[0203] 为了消除下溢条件，在一些实施例中编码器 210 首先编码图像序列并将它们存储到存储器 255。例如，编码器 210 使用多通路编码过程 100 以获取图像序列的第一编码。然

后它模拟解码器输入缓冲区 215 并且重新编码可能导致缓冲区下溢的图像。在所有缓冲区下溢条件都消除之后，通过连接 255 将重新编码的图像提供给解码器 205，连接 255 可以是网络连接（因特网、电缆、PSTN 线路等），非网络直接连接，媒体（DVD 等）等。

[0204] 图 3 举例说明了一些实施例的编码器的编码过程 300。该过程试图找到不会导致解码器缓冲区下溢的最优编码方案。如图 3 所示，过程 300 识别（步骤 302）满足所期望目标比特率（例如，序列中满足所期望平均目标比特率的每个图像的平均比特率）的图像序列的第一编码。例如，过程 300 可以使用（步骤 302）多通路编码过程 100 以获取图像序列的第一编码。

[0205] 在步骤 302 之后，编码过程 300 通过考虑各种因素，如连接速度（即，解码器用于接收编码数据的速度）、解码器输入缓冲区的大小、所编码图像的大小、解码处理速度等，的变化模拟解码器输入缓冲区 215（步骤 305）。在步骤 310，过程 300 确定所编码图像的任何片段是否会导致解码器输入缓冲区下溢。编码器用于确定（并随后消除）下溢条件的技术在下面进一步描述。

[0206] 如果过程 300 确定（步骤 310）所编码图像没有造成下溢条件，该过程结束。另一方面，如果过程 300 确定（步骤 310）在所编码图像的任何片段中存在缓冲区下溢条件的话，其就基于来自先前编码通路的这些参数的值改进编码参数（步骤 315）。然后该过程重新编码（步骤 320）具有下溢的片段以减小该片段的比特大小。在重新编码该片段之后，过程 300 检查（步骤 325）该片段以确定是否消除了下溢条件。

[0207] 当该过程确定（步骤 325）该片段仍会导致下溢时，过程 300 就转移到步骤 315 以进一步改进编码参数以消除下溢。可选的，当该过程确定（步骤 325）该片段不会导致任何下溢时，该过程就指定（步骤 330）用于重新检查并重新编码该视频序列的起始点作为步骤 320 的上一次迭代中重新编码的片段的结束之后的帧。接下来，在步骤 335，该过程重新编码在步骤 330 所指定的视频序列部分，直到（并排除）在步骤 315 和 320 指定的下溢片段随后的第一 IDR 帧。在步骤 335 之后，该过程转移回到步骤 305 以模拟解码器缓冲区以确定余下的视频序列在重新编码之后是否仍就会导致缓冲区下溢。以上描述了过程 300 从步骤 305 开始的流程。

[0208] A. 确定已编码图像序列中的下溢片段

[0209] 如上所述，编码器模拟解码器缓冲区条件以确定已编码或重新编码的图像的序列中的任何片段是否会导致解码器缓冲区中的下溢。在一些实施例中，编码器使用考虑了编码图像的大小、诸如带宽的网络条件、解码器因素（例如，输入缓冲区大小，移除图像的初始和标称时间，解码处理时间，每个图像的显示时间等）的模拟模型。

[0210] 在一些实施例中，使用 MPEG-4AVC 编码图片缓冲区（CPB）模型模拟解码器输入缓冲区条件。CPB 是在 MPEG-4H.264 标准中使用的术语，指理想基准解码器（HRD）的模拟输入缓冲区。HRD 为指定编码过程可能产生的合格数据流的可变性方面的限制的理想解码器模型。CPB 模型是众所周知的，并且出于方便在以下部分 1 中描述。CPB 和 HRD 的更为详细的描述可以在 ITU-T 推荐草案和 International Standard of Joint Video Specification 最终草案（ITU-TRec.H.264/ISO/IEC14496-10AVC）中找到。

[0211] 1. 使用 CPB 模型模拟解码器缓冲区

[0212] 以下段落描述了在一些实施例中是如何使用 CPB 模型模拟解码器输入缓冲区的。

图像 n 的第一个比特开始进入 CPB 的时间被称为初始到达时间 $t_{ai}(n)$，其推导如下：

[0213] ● $t_{ai}(0)=0$，当图像为第一图像时（即，图像 0）；

[0214] ● $t_{ai}(n)=Max(t_{af}(n-1), t_{ai}, earliest(n))$，当图像不是正编码或重新编码的序列中的第一图像时（即，n>0）。

[0215] 在以上公式中：

[0216] ● t_{ai}，$earliest(n)=t_{r,n}(n)-initial\ cpb\ removal\ delay$，

[0217] 其中 $t_{r,n}(n)$ 为如下面所指定的图像 n 从 CPB 中移除的标称移除时间，而 initial_cpb_removal_delay 为初始缓冲周期。

[0218] 图像 n 的最终到达时间通过下式推导：

[0219] $t_{af}(n)=t_{ai}(n)+b(n)/BitRate$，

[0220] 其中 b(n) 为图像 n 以比特为单位的大小。

[0221] 在一些实施例中，编码器如下所述进行自身标称移除时间的计算，而非如 H.264 规范中的那样从比特流的可选部分读取。对于图像 0，图像从 CPB 移除的标称移除时间指定为：

[0222] $t_{r,n}(0)=initial\_cpb\_removal\_delay$

[0223] 对于图像 n(n>0)，图像从 CPB 移除的标称移除时间指定为：

[0224] $t_{r,n}(n)=t_{r,n}(0)+sum_{i=0\ to\ n-1}(ti)$

[0225] 其中 $t_{r,n}(n)$ 为图像 n 的标称移除时间，而 t_i 为图片 i 的显示持续时间。

[0226] 图像 n 的移除时间如下指定：

[0227] ● $t_r(n)=t_{r,n}(n)$，当 $t_{r,n}(n)>=t_{af}(n)$ 时，

[0228] ● $t_r(n)=t_{af}(n)$，当 $t_{r,n}(n)<t_{af}(n)$ 时

[0229] 后一种情况指示图像 n 的大小 b(n) 非常大以至于它阻止了在标称移除时间时移除。

[0230] 2. 下溢片段的检测

[0231] 如在前面的部分中的描述，编码器能够模拟解码器输入缓冲区状态并在立即给定的时间瞬间获取缓冲区中的比特数量。可选的，编码器能够跟踪每个单独的图像是如何通过其标称移除时间与最终到达时间之间的差异（即，$t_b(n)=t_{r,n}(n)-t_{af}(n)$）来改变解码器输入缓冲区状态的。当 $t_b(n)$ 小于 0 时，缓冲区就会在时间瞬间 $t_{r,n}(n)$ 和 $t_{af}(n)$ 之间，并且可能会在 $t_{r,n}(n)$ 之前和 $t_{af}(n)$ 之后遭遇下溢。

[0232] 通过测试 $t_b(n)$ 是否小于 0 能够容易地发现直接陷入下溢的图像。然而，$t_b(n)$ 小于 0 的图像并非必然导致下溢，反之导致下溢的图像的 $t_b(n)$ 不一定小于 0。一些实施例通过连续不停地耗尽解码器输入缓冲区直至下溢达到其最低点将下溢片段定义为导致下溢的连续图像（以解码顺序）的伸展。

[0233] 图 4 为一些实施例中图像 $t_b(n)$ 与图像数量的标称移除时间与最终到达时间之间的差别的曲线图。该曲线针对 1500 个编码图像序列而绘制。图 4a 示意了以箭头标记其开始和结束的下溢片段。注意图 4a 中在第一下溢片段之后还发生了另外一个下溢片段，出于简化没有对其使用箭头明显标注。

[0234] 图 5 举例说明了编码器用于执行步骤 305 处的下溢检测操作的过程 500。过程 500 首先通过如上述的解释模拟解码器输入缓冲区条件确定(步骤 505)每个图像的最终到达时

间 t_{af}和标称移除时间 $t_{r,n}$。注意，由于该过程在缓冲区下溢管理的迭代过程中可能被称为若干时间，其接收图像号作为起始点并从该给定的起始点开始检查图像序列。显而易见的是，对于第一次迭代，该起始点为序列中的第一个图像。

[0235] 在步骤 510，过程 500 通过解码器将解码器输入缓冲区处的每个图像的最终到达时间与该图像的标称移除时间相比较。如果该过程确定在标称移除时间之后没有具有最终到达时间的图像(即，不存在下溢条件)，该过程就退出。另一方面，当找到了其最终到达时间在标称移除时间之后的图像时，该过程就确定存在下溢并转移到步骤 515 以识别下溢片段。

[0236] 在步骤 515，过程 500 将下溢片段识别为解码器缓冲区开始连续耗尽直至下一全局最小值的图像的片段，在此下溢条件开始改进(即，t_b（n）在图像伸展期间不会更多的负值)。过程 500 于是退出。在一些实施例中，下溢片段的开始被进一步调整为以 I 帧开始，其是标记一组相关内编码图像的开始的内编码图像。一旦识别出一个或多个导致下溢的片段，编码器就继续消除下溢。以下部分 B 描述了单个片段情况下(即，当编码整个图像序列仅包含单个下溢片段时）下溢的消除。然后部分 C 描述用于多个片段下溢的情况下的下溢消除。

[0237] B. 单个片段下溢消除

[0238] 参考图 4（a)，如果 t_b（n）与 n 的曲线具有下降斜率仅穿过 n 轴一次的话，那么在整个序列中就仅有一个下溢片段。该下溢片段开始于先前零交叉点的最近的本地最大值处，结束于零交叉点与序列结束之间的下一个全局最小值点。如果缓冲区从下溢中恢复的话，片段的结束点能够跟随具有上升斜率的曲线的另一个零交叉点。

[0239] 图 6 举例说明了在一些实施例中在图像的单个片段内解码器用于(步骤 315、320 和 325）消除下溢条件的过程 600。在步骤 605，过程 600 通过计算进入到缓冲区中的输入比特率的产出和在片段的结束处找到的最长延迟(例如，最小值 t_b（n））估算下溢片段内要减少的比特总数（ΔB)。

[0240] 接着，在步骤 610，过程 600 使用平均屏蔽帧 QP（AMQP)以及来自上一编码通路(或多个通路）的当前片段中的比特总数估算用于实现该片段所期望的比特数的期望的 AMQP，$B_T=B-\Delta B_p$，其中 p 为该片段的过程 600 的当前迭代次数。如果该迭代为该特定片段的过程 600 的首次迭代的话，AMQP 和比特的总数就是在步骤 302 处所识别的由初始编码解决方案推导得到的该片段的 AMQP 和比特总数。另一方面，当该迭代不是过程 600 的首次迭代的话，这些参数就可以由编码解决方案或在过程 600 的最后一个通路或最后多个通路中获得的解决方案推导得到。

[0241] 接下来，在步骤 615，过程 600 基于屏蔽强度 $\Phi_{F(n)}$使用所期望的 AMQP 修正平均屏蔽帧 QP，MQP（n)，以便能够忍受更多屏蔽的图像得到更多得比特扣除。该过程接着基于在步骤 315 定义的参数重新编码(步骤 620）视频片段。该过程接着检查(步骤 625）该片段以判断下溢条件是否被消除。图 4（b)举例说明了在将过程 600 应用于下溢片段以对其重新编码之后图 4（a)的下溢条件的消除情况。当消除了下溢条件时，该过程就退出。否则，过程转移回到步骤 605 以进一步调整编码参数以减少总比特大小。

[0242] C. 多下溢片段的下溢消除

[0243] 当序列中有多个下溢片段时，片段的重新编码改变了所有确保帧的缓冲区充满度

时间 t_b（n）。为了解决修改的缓冲区条件，编码器从具有下降斜率的第一个零交叉点（即，在最低点 n 处）开始，一次搜索一个下溢片段。

[0244] 下溢片段开始于先于该零交叉点的最近的本地最大值处，并结束于零交叉点和下一零交叉点（或如果没有更多零交叉点的话在序列的结束点）之间的下一全局最小值处。在找到一个片段之后，编码器理想地移除这个片段内的下溢并通过在片段结束处设置 t_b（n）为 0 以及对所有序列帧重新进行缓冲区模拟估算更新的缓冲区充满度。

[0245] 编码器接着利用修改后的缓冲区充满度继续搜索下一片段。一旦如上所述的识别了所有的下溢片段，编码器就导出 AMQP 并正如在单个片段的情况下的那样独立于其他片段修改每个片段的屏蔽帧 QP。

[0246] 普通技术人员会认识到可以以不同方式实现其他的实施例。例如，一些实施例不会识别多个导致解码器的输入缓冲区下溢的片段。一些实施例而是会如上所述执行缓冲区模拟以识别导致下溢的第一片段。在识别这样的片段之后，这些实施例就修改该片段以校正那个片段内的下溢条件，然后继续执行随后的校正部分的编码。在编码了序列的剩余部分之后，这些实施例将对下一下溢片段重复这个过程。

[0247] D. 缓冲区下溢管理的应用

[0248] 以上所述的解码器缓冲区下溢技术应用于众多编码和解码系统。以下描述了此类系统的多个例子。

[0249] 图 7 举例说明了将视频数据流服务器 710 与几台客户端解码器 715-725 相连接的网络 705。客户端通过具有诸如 300Kb/ 秒和 3Mb/ 秒的不同带宽的链路连接到网络 705。视频数据流服务器 710 控制从编码器 730 到客户端解码器 715-725 的编码视频图像流。

[0250] 流视频服务器可以决定使用网络中的最低带宽（即，300Kb/ 秒）和最小客户端缓冲区大小流动编码视频图像。在此情况下，流服务器 710 仅需要为 300Kb/ 秒的目标比特率优化的一组编码的图像。另一方面，服务器可以生成并存储针对不同带宽和不同客户端缓冲区条件优化的不同编码。

[0251] 图 8 举例说明了解码器下溢管理的另一个应用实例。在这个例子中，HD-DVD 播放器 805 从已经存储了来自视频编码器 810 的已编码视频数据的 HD-DVD 840 接收编码视频图像。HD-DVD 播放器 805 具有输入缓冲区 815、出于简化显示为一个部件 820 的一组解码模块、以及输出缓冲区 825。

[0252] 播放器 805 的输出被发送到诸如 TV 830 或计算机显示终端 835 的显示装置。HD-DVD 播放器可以具有很高的带宽，例如 29.4Mb/ 秒。为了在显示装置上维持高质量的图像，编码器确保视频图像以某种方式编码，其中图像序列中不会有太大以致不能按时传递到解码器输入缓冲区的片段。

[0253] VI. 计算机系统

[0254] 图 9 展示了所实现的本发明的一个实施例的计算机系统。计算机系统 900 包括总线 905、处理器 910、系统存储器 915、只读存储器 920、永久存储装置 925、输入装置 930、和输出装置 935。总线 905 集中表示所有的系统、外围设备、和畅通连接计算机系统 900 的众多内部设备的芯片集总线。例如，总线 905 将处理器 910 与只读存储器 920、系统存储器 915、和永久存储器装置 925 畅通连接。

[0255] 为了执行本发明的各个过程，处理器 910 从这些各种各样的存储单元中检索要执

行的指令和要处理的数据。只读存储器(ROM)920 存储了处理器 910 和计算机系统的其他模块所需的静态数据和指令。

[0256] 另一方面,永久存储器装置 925 为读 - 写存储器装置。该装置是即使是当计算机系统 900 关闭时也存储指令和数据的非易失存储器单元。本发明的一些实施例使用大容量存储装置(如磁盘或光盘及其对应的盘驱动器)作为永久存储装置 925。

[0257] 其他的实施例使用可移动存储装置(如软盘或压缩盘,及其对应的盘驱动器)作为永久存储装置。与永久存储装置 925 相类似,系统存储器 915 为读 - 写存储器装置。然而,与存储装置 925 不同的是,系统存储器为非永久性读 - 写存储器,如随机存取存储器。系统存储器存储了处理器在运行时间所需的一些指令和数据。在一些实施例中,本发明的各种处理过程保存在系统存储器 915、永久存储装置 925、和 / 或只读存储器 920 中。

[0258] 总线 905 还连接到输入和输出装置 930 和 935。输入装置使用户能够与计算机系统沟通信息并选择到计算机系统的命令。输入装置 930 包括字母数字键盘和光标控制器。输出装置 935 显示由计算机系统生成的图像。输出装置包括打印机和显示设备,如阴极射线管(CRT)或液晶显示器(LCD)。

[0259] 最后,如图 9 所示,总线 905 还通过网络适配器(未示出)将计算机 900 与网络 965 相连。在这种方式下,计算机可以是计算机网络(如局域网(“LAN”),广域网(“WAN”),或内部网)的一部分或网络(诸如因特网)的网络的一部分。计算机系统 900 的任何或所有组件都可以结合本发明使用。然而,本领域普通技术人员将理解的是,也可以结合本发明使用任何其他系统配置。

[0260] 尽管已经参考各种特定细节描述了本发明,本领域普通技术人员将认识到的是,可以不偏离本发明的精神而以其他指定的方式实施本发明。例如,不是使用模拟解码器输入缓冲区的 H264 方法,也可以使用考虑到缓冲区大小、缓冲区中图像的到达和移除时间、以及图像的解码和显示次数的其他模拟方法。

[0261] 以上所述的多个实施例计算了平均移除 SAD 以获得宏块中图像变化的指示。然而,其他实施例可以以不同的方式识别图像变化。例如,一些实施例可以预测宏块的像素的期望图像值。这些实施例接着通过从宏块的像素的亮度值中扣除该预测值,并加上该扣除部分的绝对值生成宏块 SAD。在一些实施例中,该预测值不仅基于宏块内的像素值,而且基于一个或多个相邻宏块内的像素值。

[0262] 同样,以上所述的实施例直接使用推导得出的空间和时间屏蔽值。其他的实施例为了挑出视频图像之中连续空间屏蔽值和 / 或连续时间屏蔽值的总体趋势而在使用它们之前对这些值应用平滑过滤。由此,本领域内普通技术人员将理解的是,本发明并不局限于前面所举例的细节。

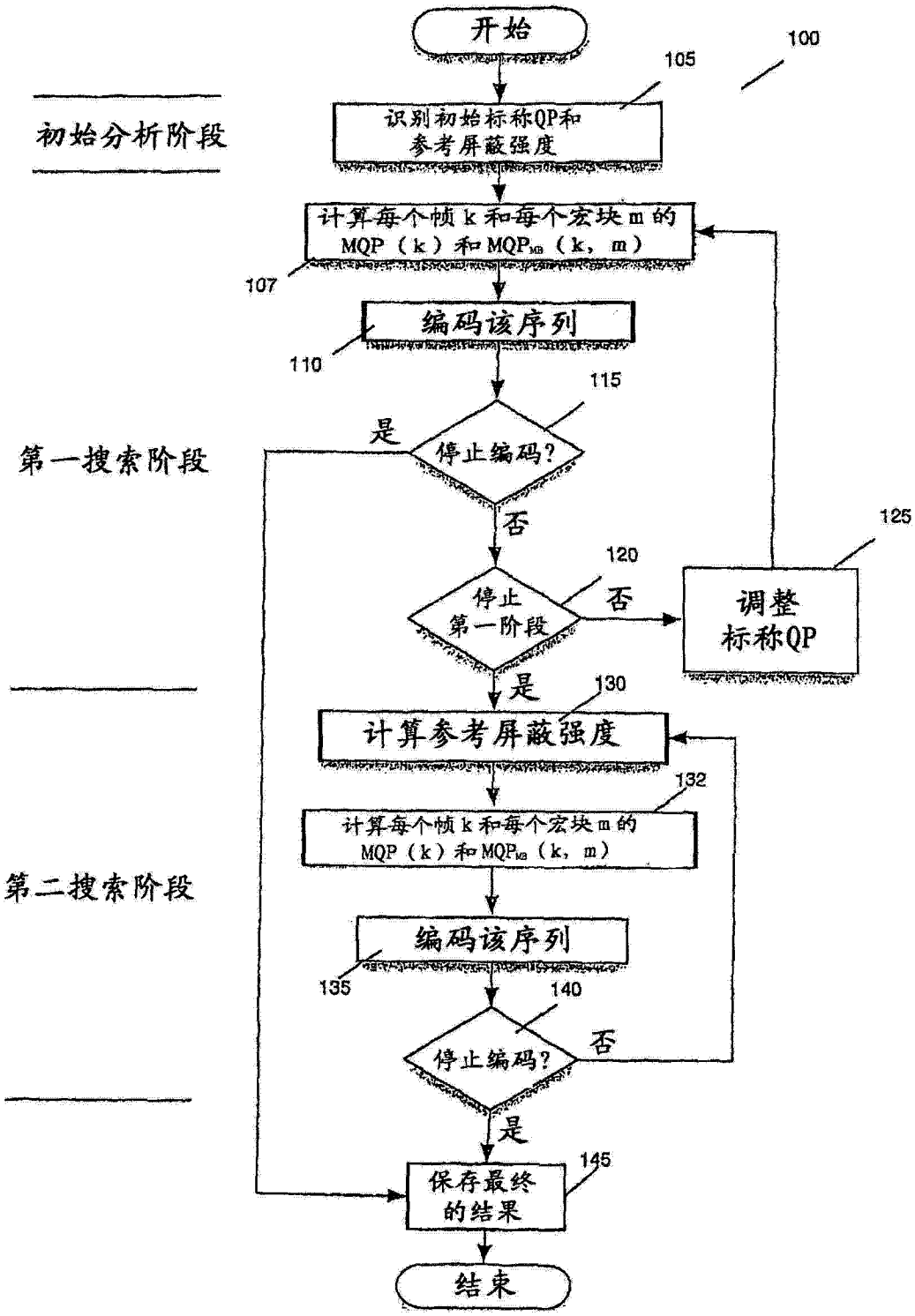

开始
100
105
识别初始标称QP和参考屏蔽强度
初始分析阶段
计算每个帧 k 和每个宏块 m 的 MQP（k）和 MQP_MB（k，m）
107
编码该序列
110
115
是
停止编码?
第一搜索阶段
否
125
120
停止第一阶段
否
调整标称QP
是
130
计算参考屏蔽强度
132
计算每个帧 k 和每个宏块 m 的 MQP（k）和 MQP_MB（k，m）
第二搜索阶段
编码该序列
135
140
停止编码?
否
是
145
保存最终的结果
结束

图 1

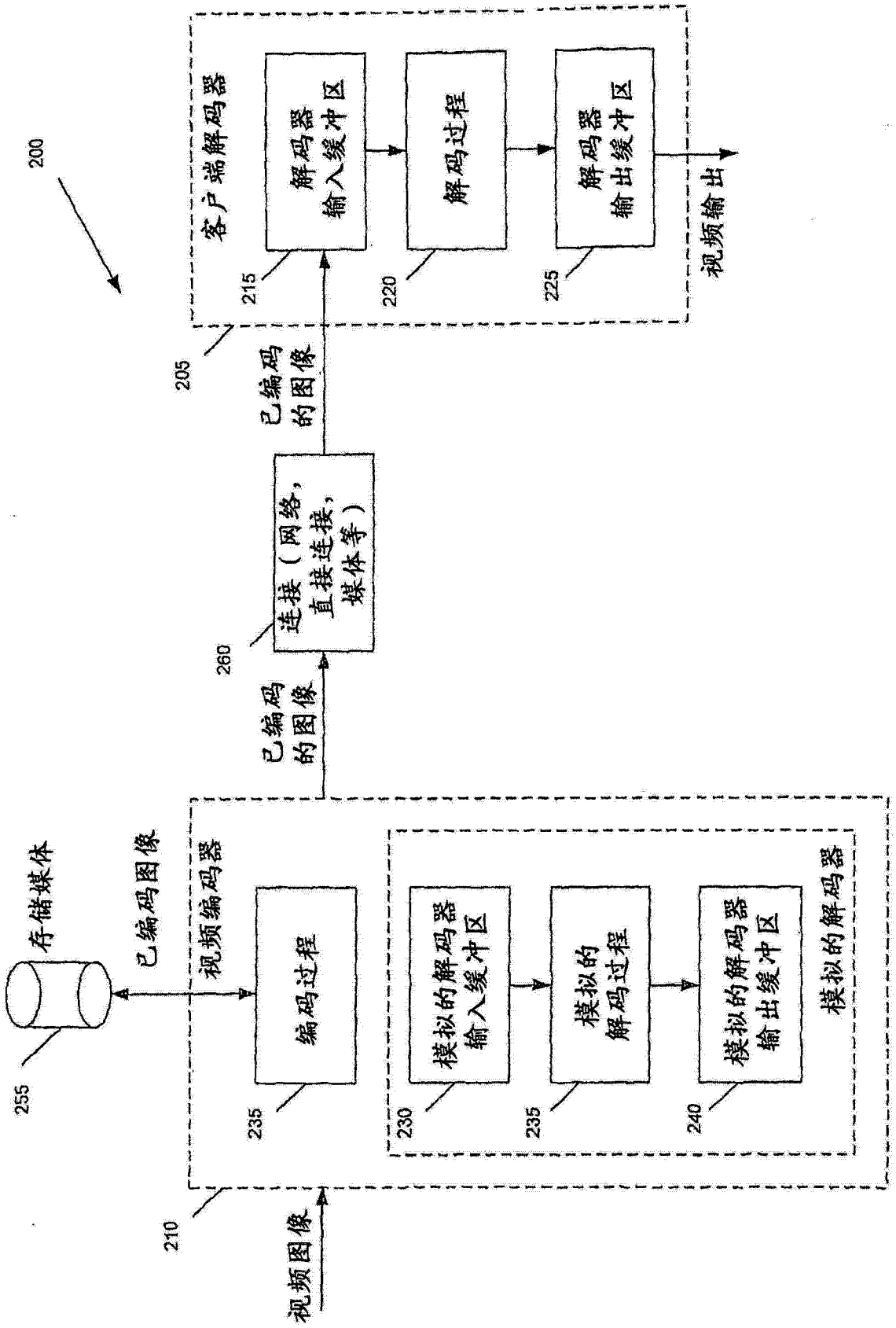

200
客户端解码器
解码器输入缓冲区
解码过程
解码器输出缓冲区
视频输出
215
220
225
205
已编码的图像
连接（网络，直接连接，媒体等）
260
已编码的图像
存储媒体
已编码图像
视频编码器
编码过程
模拟的解码器输入缓冲区
模拟的解码过程
模拟的解码器输出缓冲区
模拟的解码器
255
235
230
235
240
210
视频图像

图 2

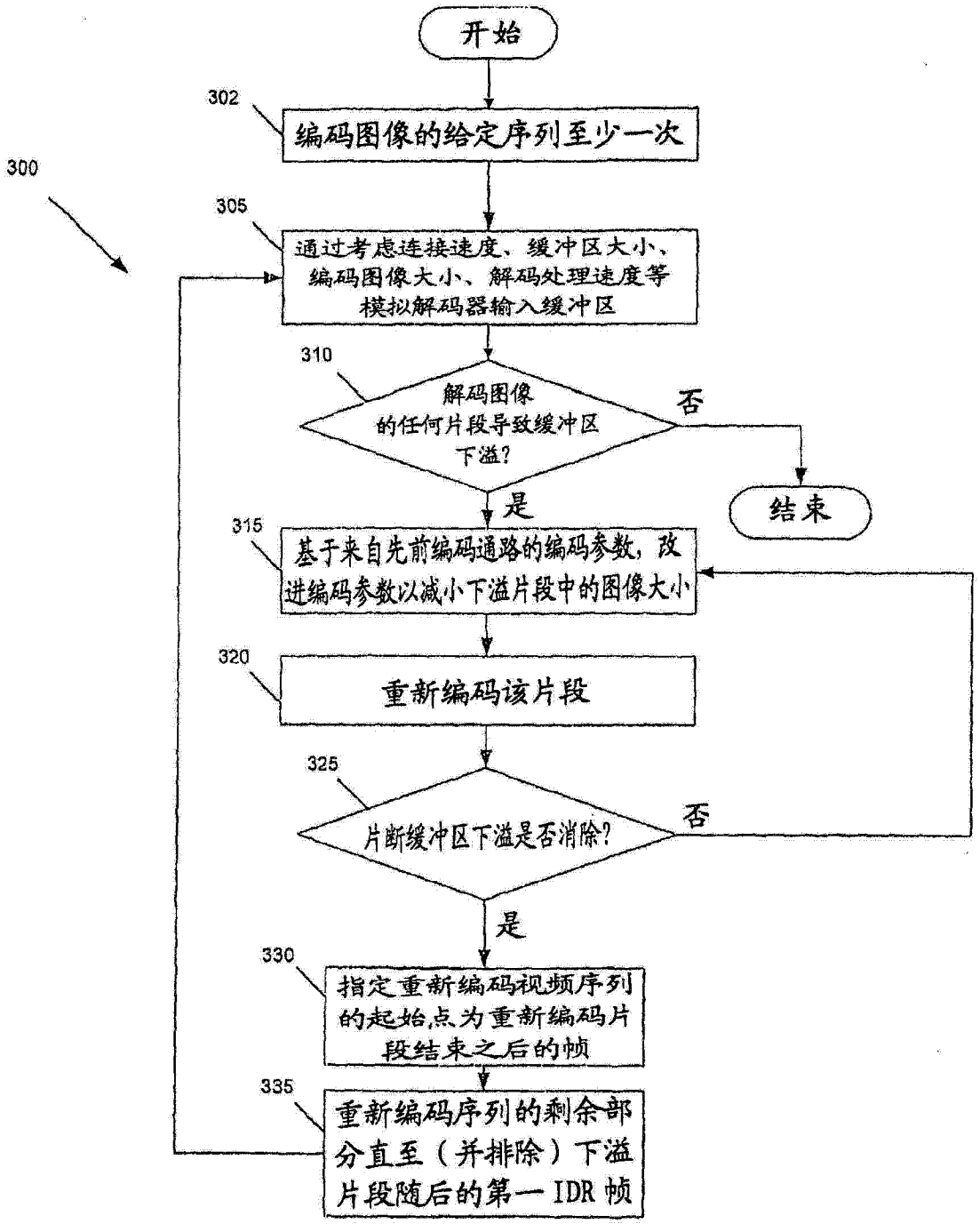

开始
302
编码图像的给定序列至少一次
300
305
通过考虑连接速度、缓冲区大小、编码图像大小、解码处理速度等模拟解码器输入缓冲区
310
解码图像的任何片段导致缓冲区下溢?
否
是
结束
315
基于来自先前编码通路的编码参数，改进编码参数以减小下溢片段中的图像大小
320
重新编码该片段
325
片断缓冲区下溢是否消除?
否
是
330
指定重新编码视频序列的起始点为重新编码片段结束之后的帧
335
重新编码序列的剩余部分直至（并排除）下溢片段随后的第一 IDR 帧

图 3

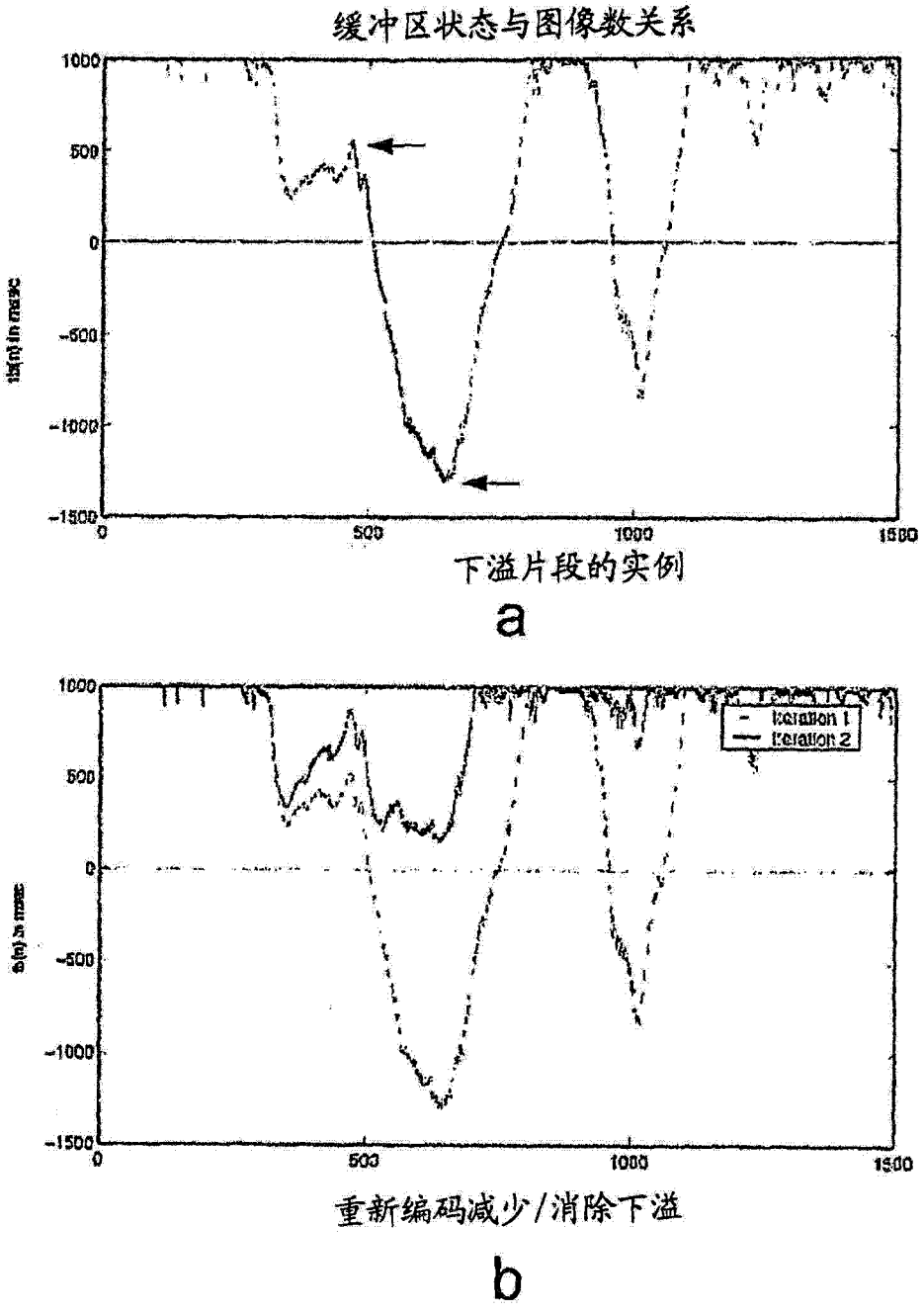

缓冲区状态与图像数关系
1000
500
0
-500
-1000
-1500
0
500
1000
1500
下溢片段的实例
a
Iteration 1
Iteration 2
重新编码减少/消除下溢
b

图 4

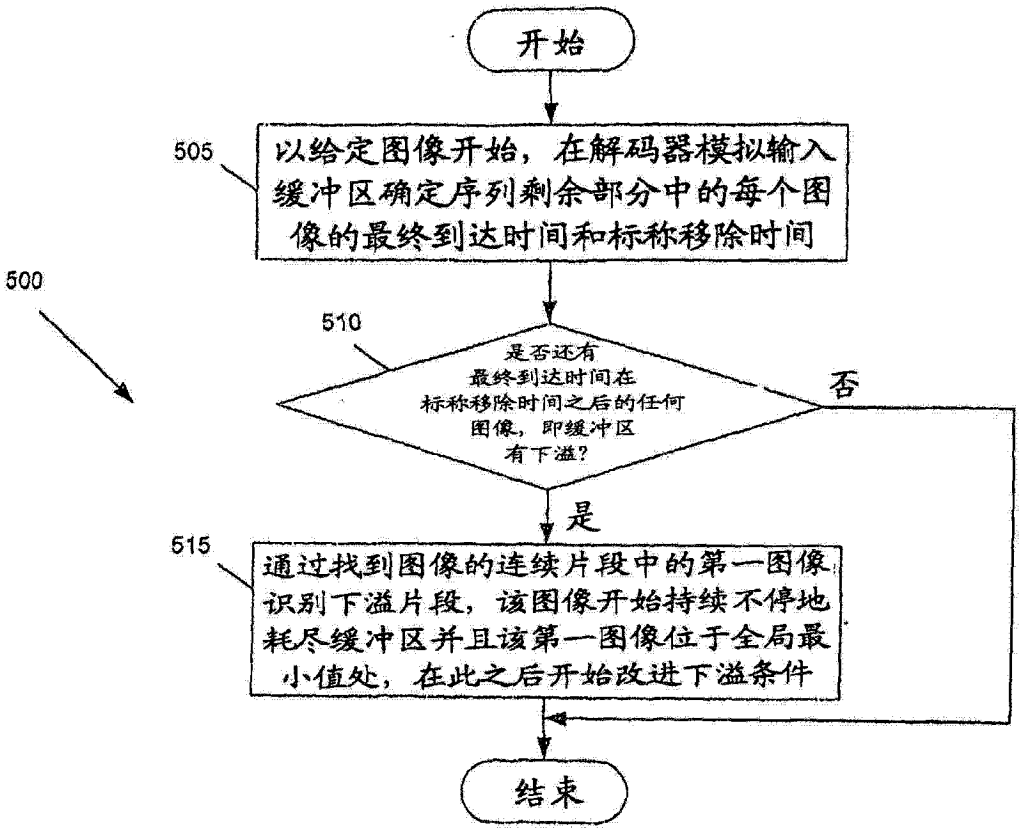

开始
505
以给定图像开始，在解码器模拟输入缓冲区确定序列剩余部分中的每个图像的最终到达时间和标称移除时间
500
510
是否还有最终到达时间在标称移除时间之后的任何图像，即缓冲区有下溢？
否
是
515
通过找到图像的连续片段中的第一图像识别下溢片段，该图像开始持续不停地耗尽缓冲区并且该第一图像位于全局最小值处，在此之后开始改进下溢条件
结束

图 5

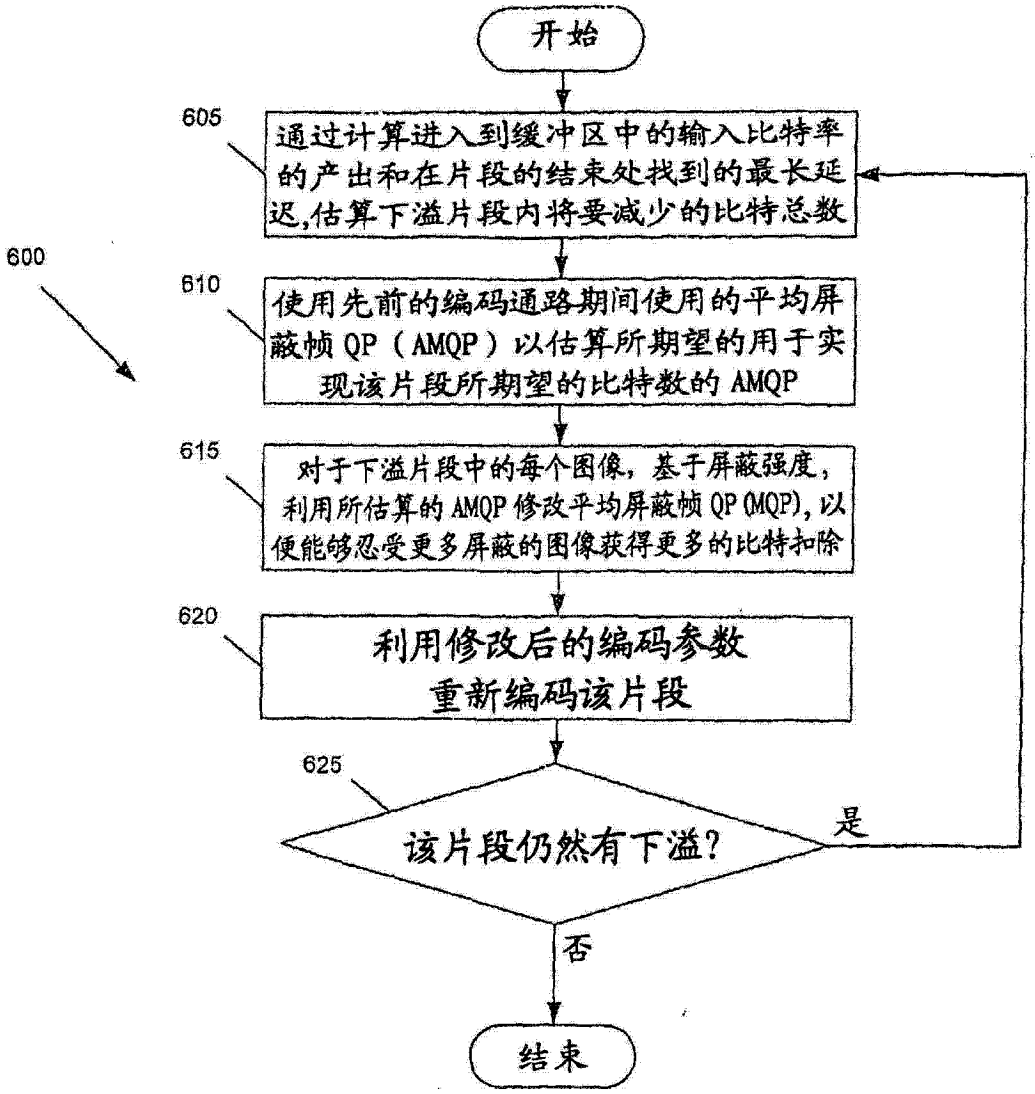

开始
605
通过计算进入到缓冲区中的输入比特率的产出和在片段的结束处找到的最长延迟,估算下溢片段内将要减少的比特总数
600
610
使用先前的编码通路期间使用的平均屏蔽帧 QP（AMQP）以估算所期望的用于实现该片段所期望的比特数的 AMQP
615
对于下溢片段中的每个图像，基于屏蔽强度，利用所估算的 AMQP 修改平均屏蔽帧 QP (MQP)，以便能够忍受更多屏蔽的图像获得更多的比特扣除
620
利用修改后的编码参数重新编码该片段
625
该片段仍然有下溢?
是
否
结束

图 6

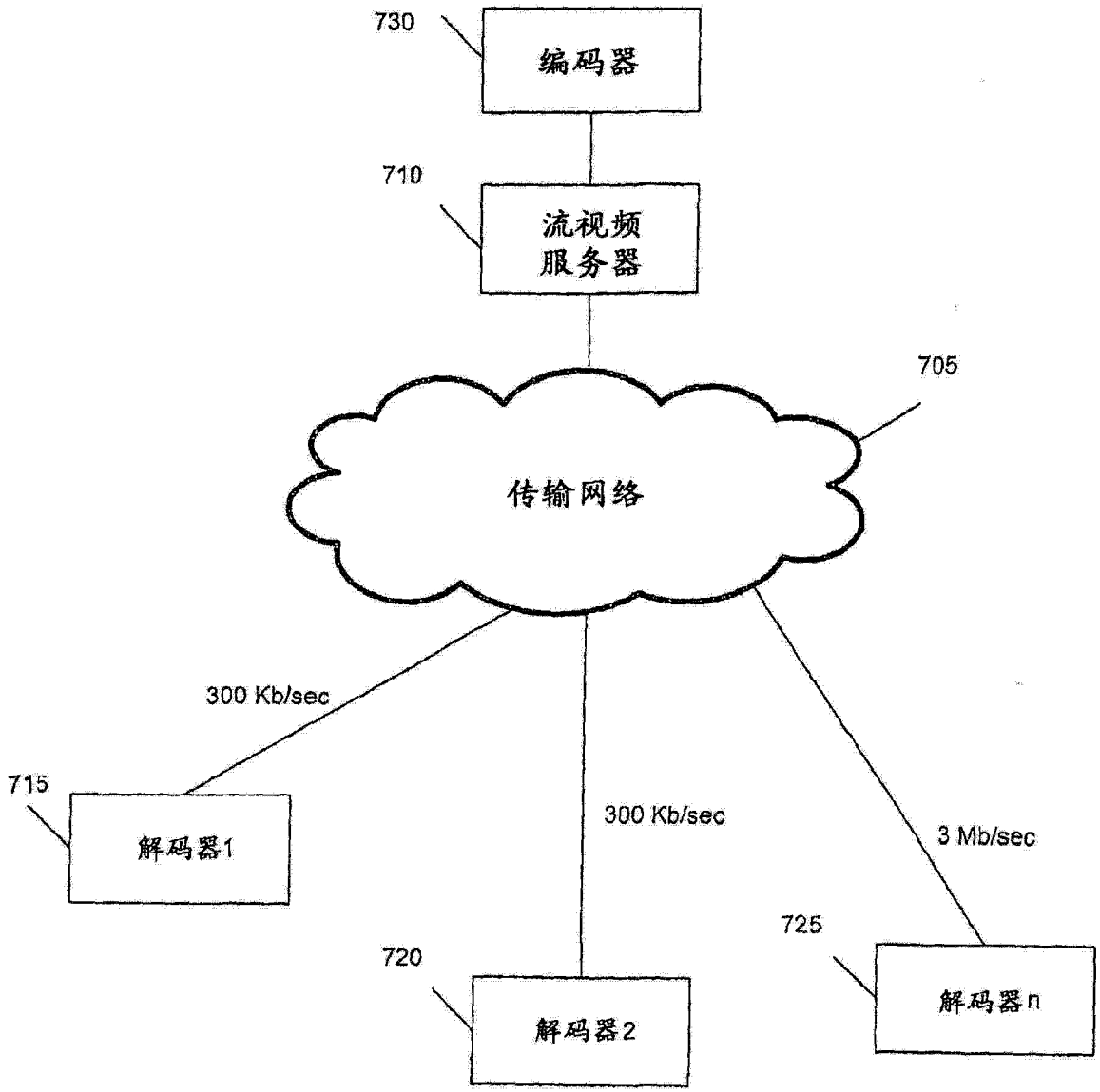

730
编码器
710
流视频
服务器
705
传输网络
300 Kb/sec
715
解码器1
300 Kb/sec
3 Mb/sec
725
720
解码器2
解码器n

图 7

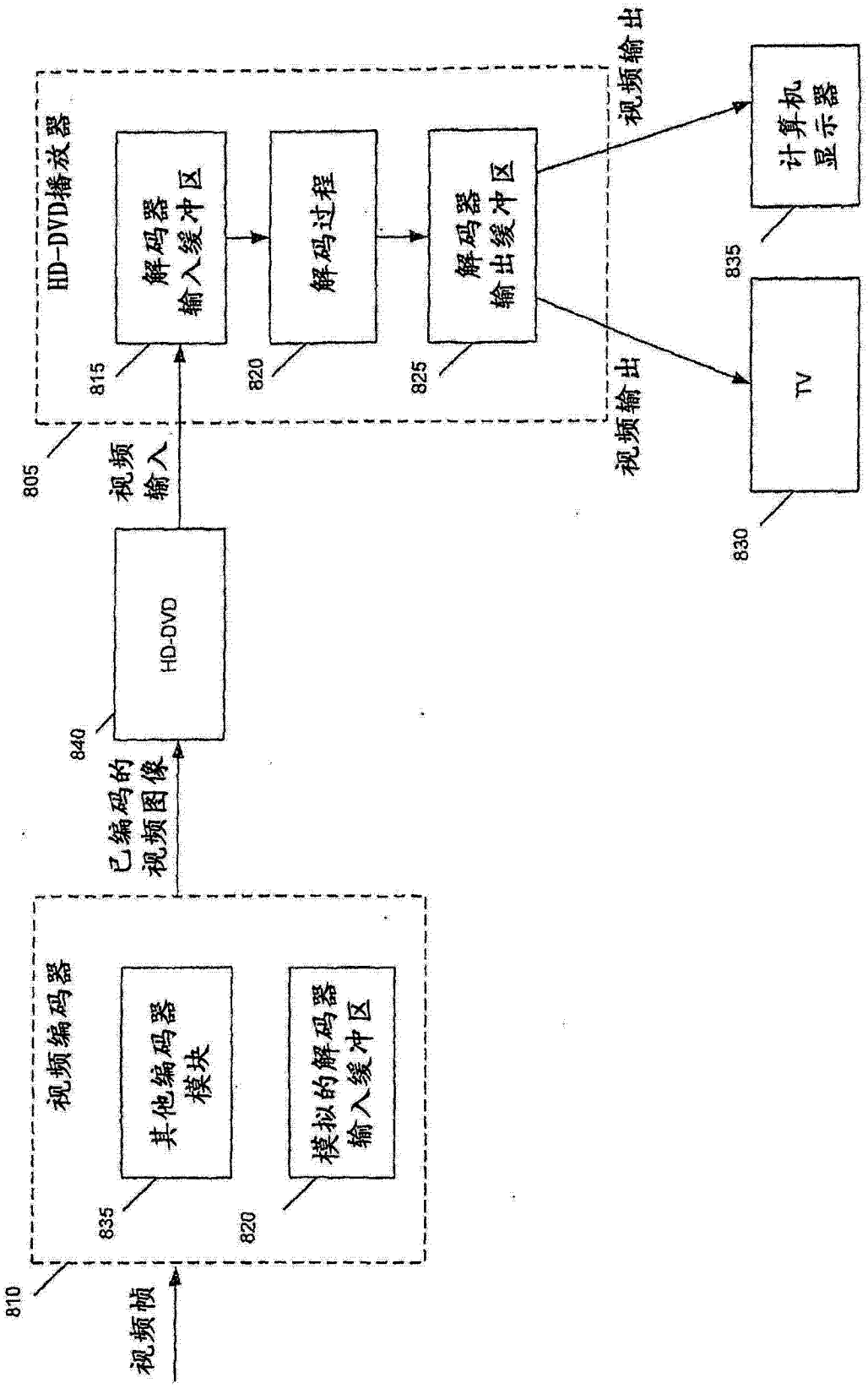

视频帧
810
视频编码器
其他编码器模块
835
模拟的解码器输入缓冲区
820
已编码的视频图像
840
HD-DVD
视频输入
805
HD-DVD播放器
解码器输入缓冲区
815
解码过程
820
解码器输出缓冲区
825
视频输出
视频输出
计算机显示器
835
TV
830

图 8

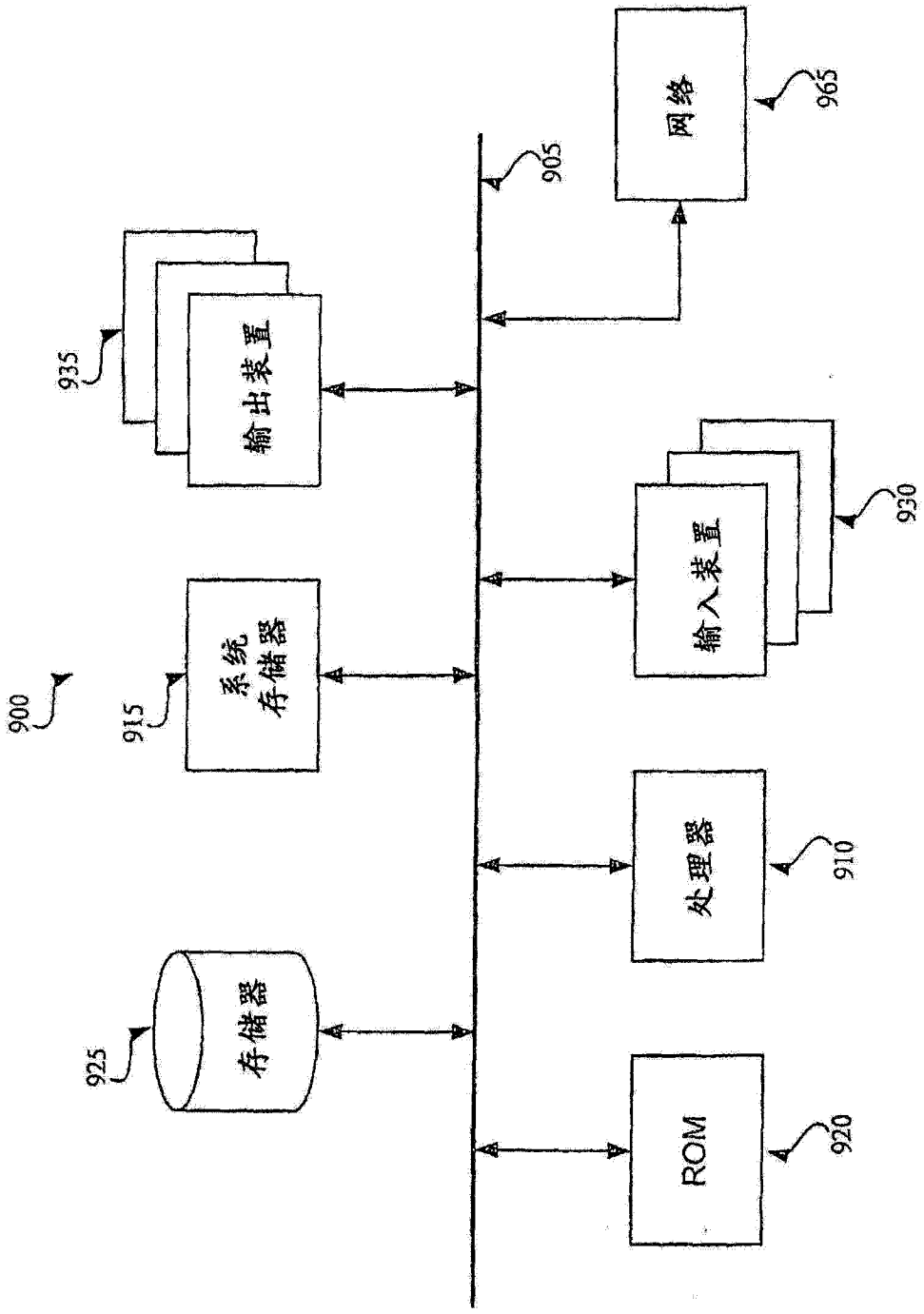

网络
965
905
输出装置
935
输入装置
930
系统存储器
915
900
处理器
910
存储器
925
ROM
920

图 9