



(12) 发明专利申请

(10) 申请公布号 CN 101971630 A

(43) 申请公布日 2011. 02. 09

(21) 申请号 200880110403. 4

代理人 宋鹤 南霆

(22) 申请日 2008. 09. 16

(51) Int. Cl.

(30) 优先权数据

H04N 7/26 (2006. 01)

60/977, 709 2007. 10. 05 US

(85) PCT申请进入国家阶段日

2010. 04. 06

(86) PCT申请的申请数据

PCT/US2008/010775 2008. 09. 16

(87) PCT申请的公布数据

W02009/048502 EN 2009. 04. 16

(71) 申请人 汤姆逊许可证公司

地址 法国布洛尼 - 比扬古市

(72) 发明人 罗建聪 尹鹏

(74) 专利代理机构 北京东方亿思知识产权代理

有限责任公司 11258

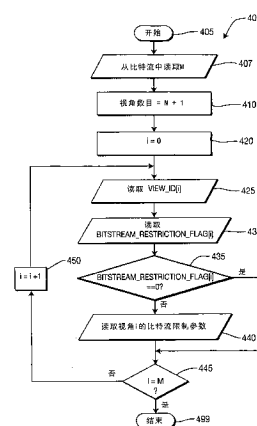
权利要求书 1 页 说明书 19 页 附图 8 页

(54) 发明名称

用于将视频可用性信息 (VUI) 结合在多视角视频 (MVC) 编码系统内的方法和装置

(57) 摘要

提供了用于将视频可用性信息 (VUI) 结合在多视角视频编码 (MVC) 中的方法和装置。一种装置 (100) 包括编码器 (100), 用于通过为从以下各项中选出的至少一种规定视频可用性信息来对多视角视频内容编码: 各个视角 (300)、一视角中的各个时间级别 (500), 以及各个操作点 (700)。另外, 一种装置 (200) 包括解码器, 用于通过为从以下各项中选出的至少一种规定视频可用性信息来对多视角视频内容解码: 各个视角 (400)、一视角中的各个时间级别 (600), 以及各个操作点 (800)。



1. 一种装置,包括:

解码器(200),用于通过为从以下各项中选出的至少一种规定视频可用性信息来对多视角视频内容解码:各个视角、一视角中的各个时间级别,以及各个操作点。

2. 如权利要求1所述的装置,其中参数是在至少一个高级别语法元素中规定的。

3. 如权利要求2所述的装置,其中所述至少一个高级别语法元素包括 `MvcVuiParametersExtension()` 语法元素、`MvcScalabilityInfo` 补充增强信息语法消息、序列参数集的至少一部分、图片参数集以及补充增强信息中的至少一种。

4. 如权利要求1所述的装置,其中所述视频可用性信息的至少一部分包括比特流限制参数。

5. 一种方法,包括:

通过为从以下各项中选出的至少一种规定视频可用性信息来对多视角视频内容解码:各个视角(400)、一视角中的各个时间级别(600),以及各个操作点(800)。

6. 如权利要求5所述的方法,其中参数是在至少一个高级别语法元素中规定的。

7. 如权利要求6所述的方法,其中所述至少一个高级别语法元素包括 `MvcVuiParametersExtension()` 语法元素、`MvcScalabilityInfo` 补充增强信息语法消息、序列参数集的至少一部分、图片参数集以及补充增强信息中的至少一种。

8. 如权利要求5所述的方法,其中所述视频可用性信息的至少一部分包括比特流限制参数。

9. 一种用于视频编码、解码和传输的视频信号结构,包括:

通过为从以下各项中选出的至少一种规定视频可用性信息来编码的多视角视频内容:各个视角、一视角中的各个时间级别,以及各个操作点(800)。

10. 如权利要求9所述的视频信号结构,其中参数是在至少一个高级别语法元素中规定的。

11. 如权利要求10所述的视频信号结构,其中所述至少一个高级别语法元素包括 `MvcVuiParametersExtension()` 语法元素、`MvcScalabilityInfo` 补充增强信息语法消息、序列参数集的至少一部分、图片参数集以及补充增强信息中的至少一种。

12. 如权利要求9所述的视频信号结构,其中所述视频可用性信息的至少一部分包括比特流限制参数。

用于将视频可用性信息 (VUI) 结合在多视角视频 (MVC) 编码系统内的方法和装置

[0001] 与相关申请的交叉引用

[0002] 本申请要求 2007 年 10 月 5 日提交的美国临时申请 No. 60/977,709 的权益,这里通过引用将该临时申请全部结合在此。另外,本申请与标题为“METHODS AND APPARATUS FOR INCORPORATING VIDEOUSABILITY INFORMATION(VUI)WITHIN A MULTI-VIEW VIDEO(MVC) CODING SYSTEM”、代理人案卷号为 No. PU070239 的非临时申请相关,该非临时申请也要求 2007 年 10 月 5 日提交的美国临时申请 No. 60/977,709 的权益,并且被转让给与本申请相同的受让人、通过引用被结合在此并与本申请同时提交。

技术领域

[0003] 本发明总地涉及视频编码和解码,更具体而言涉及用于将视频可用性信息 (video usability information, VUI) 结合在多视角视频编码 (multi-view video coding, MVC) 内的方法和装置。

背景技术

[0004] 国际标准化组织 / 国际电工委员会 (ISO/IEC) 运动图片专家组 -4 (MPEG-4) 第 10 部分高级视频编码 (AVC) 标准 / 国际电信联盟电信部 (ITU-T) H. 264 推荐 (以下称之为“MPEG-4AVC 标准”) 规定了序列参数集的视频可用性信息 (VUI) 参数的语法和语义。视频可用性信息包括纵横比、过扫描、视频信号类型、色度位置、定时、网络抽象层 (NAL) 假设参考解码器 (HRD) 参数、视频编码层 (VCL) 假设参考解码器参数、比特流限制等等的信息。视频可用性信息为相应比特流提供了额外信息,以允许用户进行更宽的应用。例如,在比特流限制信息中,视频可用性信息规定:(1) 运动是否越过图片边界;(2) 每图片的最大字节;(3) 每宏块的最大比特;(4) 最大运动向量长度(水平和垂直);(5) 重排序帧的数目;(6) 最大解码帧缓冲器大小。当解码器看到该信息时,解码器不是使用“级别”(level) 信息来设定解码要求(这一般来说高于比特流实际所要求的),而是可以基于更严格的限度来定制其解码操作。

[0005] 多视角视频编码 (MVC) 是 MPEG-4 AVC 标准的扩展。在多视角视频编码中,通过利用多个视角之间的相关性,可以对多个视角的视频图像进行编码。在所有视角之中,一个视角是基本视角,它是 MPEG-4 AVC 标准兼容的,并且不能从其他视角中预测出。其他视角被称为非基本视角。非基本视角可以从基本视角和其他非基本视角中预测出。可以在时间上对每个视角进行欠采样。视角的时间子集可以由 temporal_id 语法元素来标识。视角的时间级别 (temporal level) 是视频信号的一种表示。在多视角视频编码的比特流中,可以有视角和时间级别的不同组合。每个组合被称为一操作点。可以从比特流中提取出对应于这些操作点的子比特流。

发明内容

[0006] 本发明解决了现有技术的这些和其他缺陷和缺点,本发明涉及用于将视频可用性信息(VUI)结合在多视角视频编码(MVC)内的方法和装置。

[0007] 根据本发明的一个方面,提供了一种装置。该装置包括编码器,用于通过为各个视角、一视角中的各个时间级别以及各个操作点中的至少一种规定视频可用性信息来对多视角视频内容编码。

[0008] 根据本发明的另一方面,提供了一种方法。该方法包括通过为各个视角、一视角中的各个时间级别以及各个操作点中的至少一种规定视频可用性信息来对多视角视频内容编码。

[0009] 根据本发明的另一方面,提供了一种装置。该装置包括解码器,用于通过为各个视角、一视角中的各个时间级别以及各个操作点中的至少一种规定视频可用性信息来对多视角视频内容解码。

[0010] 根据本发明的另一方面,提供了一种方法。该方法包括通过为各个视角、一视角中的各个时间级别以及各个操作点中的至少一种规定视频可用性信息来对多视角视频内容解码。

[0011] 本发明的这些和其他方面、特征和优点将从以下的要结合附图理解的对示例性实施例的详细描述中清楚显现出来。

附图说明

[0012] 根据以下示例性附图可以更好地理解本发明,附图中:

[0013] 图1是根据本发明一个实施例可以应用本发明的示例性多视角视频编码(MVC)编码器的框图;

[0014] 图2是根据本发明一个实施例可以应用本发明的示例性多视角视频编码(MVC)解码器的框图;

[0015] 图3是根据本发明一个实施例用于利用 `mvc_vui_parameters_extension()` 语法元素来对每个视角的比特流限制参数编码的示例性方法的流程图;

[0016] 图4是根据本发明一个实施例用于利用 `mvc_vui_parameters_extension()` 语法元素来对每个视角的比特流限制参数解码的示例性方法的流程图;

[0017] 图5是根据本发明一个实施例用于利用 `mvc_vui_parameters_extension()` 语法元素来对每个视角中的每个时间级别的比特流限制参数编码的示例性方法的流程图;

[0018] 图6是根据本发明一个实施例用于利用 `mvc_vui_parameters_extension()` 语法元素来对每个视角中的每个时间级别的比特流限制参数解码的示例性方法的流程图;

[0019] 图7是根据本发明一个实施例用于利用 `view_scalability_parameters_extension()` 语法元素来对每个操作点的比特流限制参数编码的示例性方法的流程图;并且

[0020] 图8是根据本发明一个实施例用于利用 `view_scalability_parameters_extension()` 语法元素来对每个操作点的比特流限制参数解码的示例性方法的流程图。

具体实施方式

[0021] 本发明涉及用于将视频可用性信息(VUI)结合在多视角视频编码(MVC)内的方法

和装置。

[0022] 本说明书说明了本发明。因此,将会明白,本领域的技术人员将能够设计出各种配置,这些配置虽然在这里没有明确描述或者示出,但却实现了本发明并被包括在其精神和范围之内。

[0023] 这里记载的所有示例和条件语言都意图用于教学目的,以帮助读者理解本发明和发明人对推进现有技术所贡献的思想,并且应当被解释为不限于这种具体记载的示例和条件。

[0024] 另外,这里的所有记载本发明的原理、方面和实施例及其具体示例的陈述,都意图包括其结构和功能等同物。此外,希望这种等同物既包括当前已知的等同物,也包括将来开发出的等同物,即所开发出的任何执行相同功能的元件,不论其结构如何。

[0025] 因此,例如,本领域的技术人员将会明白,这里给出的框图表示了实现本发明的示例性电路的概念性视图。类似地,将会明白,任何流程图、状态转换图、伪代码等等都表示实质上可以表示在计算机可读介质中并可以由计算机或处理器来如此执行的各种过程,不论这种计算机或处理器是否被明确地示出。

[0026] 图中所示的各种元件的功能可通过使用专用硬件以及能够结合适当的软件来执行软件的硬件来提供。当由处理器提供时,这些功能可由单个专用处理器提供、由单个共享处理器提供、或者由多个独立的处理器(其中一些可能被共享)提供。另外,对术语“处理器”或“控制器”的明确使用不应当被解释为只指能够执行软件的硬件,而是可以隐含地包括(但不限于)数字信号处理器(“DSP”)硬件、用于存储软件的只读存储器(“ROM”)、随机存取存储器(“RAM”)和非易失性存储设备。

[0027] 还可以包括其他传统的和/或定制的硬件。类似地,图中所示的任何开关都只是概念性的。它们的功能可通过程序逻辑的操作来实现、通过专用逻辑来实现、通过程序控制和专用逻辑的交互来实现,或者甚至手工实现,具体的技术可由实现者根据对上下文的更具体理解来选择。

[0028] 在其权利要求中,被表达为用于执行指定功能的装置的任何元件都意图涵盖执行该功能的任何方式,例如包括 a) 执行该功能的电路元件的组合或者 b) 任何形式的软件,因此包括固件、微代码等等,这种软件与用于执行该软件的适当电路相组合以执行该功能。这种权利要求所限定的本发明存在于以下事实中:由所记载的各种装置所提供的功能以权利要求所要求的方式被组合到一起。因此,认为任何能够提供这些功能的装置都与这里示出的那些是等同的。

[0029] 说明书中提到本发明的“一个实施例”或“实施例”意指结合该实施例描述的特定特征、结构、特性等等被包括在本发明的至少一个实施例中。因此,说明书中各处出现的短语“在一个实施例中”或者“在实施例”中不一定均指相同实施例。

[0030] 应当明白,例如在“A 和 / 或 B”和“A 和 B 中的至少一个”的情况下使用术语“和 / 或”和“... 中的至少一个”意图涵盖仅对列出的第一选项 (A) 的选择、或者仅对列出的第二选项 (B) 的选择、或者对两个选项 (A 和 B) 的选择。又例如,在“A、B 和 / 或 C”和“A、B 和 C 中的至少一个”的情况下,这种短语意图涵盖仅对列出的第一选项 (A) 的选择、或者仅对列出的第二选项 (B) 的选择、或者仅对列出的第三选项 (C) 的选择、或者仅对列出的第一和第二选项 (A 和 B) 的选择、或者仅对列出的第一和第三选项 (A 和 C) 的选择、或者仅对列出的

第二和第三选项 (B 和 C) 的选择、或者对所有三个选项 (A 和 B 和 C) 的选择。列出了多少项,这就可以扩展到多少项,这对于本领域和相关领域的普通技术人员来说是显而易见的。

[0031] 多视角视频编码 (MVC) 是多视角序列的编码的压缩框图。多视角视频编码 (MVC) 序列是从不同的视角点捕捉同一场景的两个或更多个视频序列的集合。

[0032] 这里可互换使用的“交叉视角”(cross-view) 和“视角间”(inter-view) 都指的是属于除当前视角外的视角的图片。

[0033] 另外,这里使用的“高级别语法”(high level syntax) 指的是存在于比特流中的、在层次上高于宏块层的语法。例如,这里使用的高级别语法可以指(但不限于)片层头部级、补充增强信息 (SEI) 级、图片参数集 (PPS) 级、序列参数集 (SPS) 级和网络抽象层 (NAL) 单元头部级的语法。

[0034] 另外,应当明白,虽然这里出于例示目的针对 MPEG-4 AVC 标准的多视角视频编码扩展描述了本发明的一个或多个实施例,但是本发明并不仅限于该扩展和 / 或该标准,因此,可以针对其他视频编码标准、推荐及其扩展使用,同时仍维持本发明的精神。

[0035] 另外,应当明白,虽然这里出于例示目的针对比特流限制信息描述了本发明的一个或多个实施例,但是本发明并不仅限于使用比特流限制信息作为一类视频可用性信息,因此,根据本发明也可使用其他类型的可被扩展来针对多视角视频编码使用的视频可用性信息,同时仍维持本发明的精神。

[0036] 参考图 1, 示例性的多视角视频编码 (MVC) 编码器由标号 100 总地指示。编码器 100 包括组合器 105, 该组合器 105 具有与变换器 110 的输入发生信号通信连接的输出。变换器 110 的输出与量化器 115 的输入发生信号通信连接。量化器 115 的输出与熵编码器 120 的输入和逆量化器 125 的输入发生信号通信连接。逆量化器 125 的输出与逆变换器 130 的输入发生信号通信连接。逆变换器 130 的输出与组合器 135 的第一同相输入发生信号通信连接。组合器 135 的输出与内预测器 145 的输入和解决块滤波器 150 的输入发生信号通信连接。解决块滤波器 150 的输出与(用于视角 i 的) 参考图片存储装置 155 的输入发生信号通信连接。参考图片存储装置 155 的输出与运动补偿器 175 的第一输入和运动估计器 180 的第一输入发生信号通信连接。运动估计器 180 的输出与运动补偿器 175 的第二输入发生信号通信连接。

[0037] (用于其他视角的) 参考图片存储装置 160 的输出与视差 / 照明估计器 170 的第一输入和视差 / 照明补偿器 165 的第一输入发生信号通信连接。视差 / 照明估计器 170 的输出与视差 / 照明补偿器 165 的第二输入发生信号通信连接。

[0038] 熵编码器 120 的输出可用作编码器 100 的输出。组合器 105 的同相输入可用作编码器 100 的输入, 并且与视差 / 照明估计器 170 的第二输入和运动估计器 180 的第二输入发生信号通信连接。开关 185 的输出与组合器 135 的第二同相输入和组合器 105 的反相输入发生信号通信连接。开关 185 包括与运动补偿器 175 的输出发生信号通信连接的第一输入、与视差 / 照明补偿器 165 的输出发生信号通信连接的第二输入、以及与内预测器 145 的输出发生信号通信连接的第三输入。

[0039] 模式判决模块 140 具有连接到开关 185 的输出, 用于控制开关 185 选择哪个输入。

[0040] 参考图 2, 示例性的多视角视频编码 (MVC) 解码器由标号 200 总地指示。解码器 200 包括熵解码器 205, 该熵解码器 205 具有与逆量化器 210 的输入发生信号通信连接的输

出。该逆量化器的输出与逆变换器 215 的输入发生信号通信连接。逆变换器 215 的输出与组合器 220 的第一同相输入发生信号通信连接。组合器 220 的输出与解块滤波器 225 的输入和内预测器 230 的输入发生信号通信连接。解块滤波器 225 的输出与（用于视角 i 的）参考图片存储装置 240 的输入发生信号通信连接。参考图片存储装置 240 的输出与运动补偿器 235 的第一输入发生信号通信连接。

[0041] （用于其他视角的）参考图片存储装置 245 的输出与视差 / 照明补偿器 250 的第一输入发生信号通信连接。

[0042] 熵解码器 205 的输入可用作解码器 200 的输入,用于接收残余比特流。另外,模式模块 260 的输入也可用作解码器 200 的输入,用于接收控制语法,以控制开关 255 选择哪个输入。此外,运动补偿器 235 的第二输入可用作解码器 200 的输入,用于接收运动向量。另外,视差 / 照明补偿器 250 的第二输入可用作解码器 200 的输入,用于接收视差向量和照明补偿语法。

[0043] 开关 255 的输出与组合器 220 的第二同相输入发生信号通信连接。开关 255 的第一输入与视差 / 照明补偿器 250 的输出发生信号通信连接。开关 255 的第二输入与运动补偿器 235 的输出发生信号通信连接。开关 255 的第三输入与内预测器 230 的输出发生信号通信连接。模式模块 260 的输入与开关 255 发生信号通信连接,用于控制开关 255 选择哪个输入。解块滤波器 225 的输出可用作解码器的输出。

[0044] 在 MPEG-4 AVC 标准中,为视频可用性信息 (VUI) 规定了序列参数集的语法和语义参数。这表示可被插入到比特流中以增强视频对于许多种目的的可用性的附加信息。视频可用性信息包括纵横比、过扫描、视频信号类型、色度位置、定时、网络抽象层 (NAL) 假设参考解码器 (HRD) 参数、视频编码层 (VCL) 假设参考解码器参数、比特流限制等等的信息。

[0045] 根据本发明的一个或多个实施例,我们将现有的视频可用性信息字段用于新的、与现有技术不同的目的,并且还将其使用扩展到多视角视频编码 (MVC)。在我们的多视角视频编码方案中,视频可用性信息被扩展,从而使得其在例如不同视角、一视角中的不同时间级别或者不同操作点之间可以是不同的。因此,根据一个实施例,我们根据以下各项中的一个或多个（但不限于此）来规定视频可用性信息:为各个视角规定视频可用性信息;为一视角中的各个时间级别规定视频可用性信息;以及分别为各个操作点规定视频可用性信息。

[0046] 在 MPEG-4AVC 标准中,包括视频可用性信息 (VUI) 的集合可在序列参数集 (SPS) 中传送。根据一个实施例,我们扩展了视频可用性信息的概念,以用在多视角视频编码 (MVC) 上下文内。有利地,这允许了为多视角视频编码中的不同视角、一视角中的不同时间级别或者不同操作点规定不同的视频可用性信息。在一个实施例中,我们在针对多视角视频编码考虑、修改和使用视频可用性信息中的视频可用性信息方面提供了一种新的方法。

[0047] MPEG-4AVC 标准中的比特流限制信息是在作为 `sequence_parameter_set()` 的一部分的 `vui_parameters()` 语法元素中规定的。表 1 示出了 `vui_parameters()` 的 MPEG-4 AVC 标准语法。

[0048] 表 1

[0049]

vui_parameters() {	C	描述符
aspect_ratio_info_present_flag	0	u(1)
...		
bitstream_restriction_flag	0	u(1)
if(bitstream_restriction_flag) {		
motion_vectors_over_pic_boundaries_flag	0	u(1)
max_bytes_per_pic_denom	0	ue(v)
max_bits_per_mb_denom	0	ue(v)
log2_max_mv_length_horizontal	0	ue(v)
log2_max_mv_length_vertical	0	ue(v)
num_reorder_frames	0	ue(v)
max_dec_frame_buffering	0	ue(v)
}		
}		

[0050] 比特流限制信息的语法元素的语义如下：

[0051] bitstream_restriction_flag 等于 1 规定以下的经编码的视频序列比特流限制参数存在。

[0052] bitstream_restriction_flag 等于 0 规定以下的经编码的视频序列比特流限制参数不存在。

[0053] motion_vectors_over_pic_boundaries_flag 等于 0 表明，不使用在图片边界外的样本和在利用在图片边界外的一个或多个样本得出其值的分数样本位置处的样本来对任何样本进行互预测 (inter prediction)。

[0054] motion_vectors_over_pic_boundaries_flag 等于 1 表明，一个或多个在图片边界外的样本可用于互预测。当 motion_vectors_over_pic_boundaries_flag 语法元素不存在时，motion_vectors_over_pic_boundaries_flag 值应当被推断为等于 1。

[0055] max_bytes_per_pic_denom 指示出与经编码的视频序列中的任何经编码的图片相关联的虚拟编码层 (VCL) 网络抽象层 (NAL) 单元的大小的总和所不超过的字节数目。

[0056] 就此而言,表示网络抽象层单元流中的图片的字节数目被规定为该图片的虚拟编码层网络抽象层单元数据的字节的总数(即,虚拟编码层网络抽象层单元的NumBytesInNALunit变量的总计)。max_bytes_per_pic_denom的值应当在0至16的范围中,包括0和16。

[0057] 取决于max_bytes_per_pic_denom,以下适用:

[0058] - 如果max_bytes_per_pic_denom等于0,则没有指示限度。

[0059] - 否则(max_bytes_per_pic_denom不等于0),则不应当用多于以下数目的字节在经编码视频序列中表示经编码图片:

[0060] $(\text{PicSizeInMbs} * \text{RawMbBits}) \div (8 * \text{max_bytes_per_pic_denom})$

[0061] 当max_bytes_per_pic_denom语法元素不存在时,max_bytes_per_pic_denom的值应当被推断为等于2。变量PicSizeInMbs是图片中的宏块的数目。变量RawMbBits是像MPEG-4 AVC标准的子条款7.4.2.1中那样得出的。

[0062] max_bits_per_mb_denom指示出在经编码视频序列的任何图片中任何宏块的macroblock_layer()数据的经编码比特的最大数目。max_bits_per_mb_denom的值应当在0至16的范围中,包括0和16。

[0063] 取决于max_bits_per_mb_denom,以下适用:

[0064] - 如果max_bits_per_mb_denom等于0,则没有指示限度。

[0065] - 否则(max_bits_per_mb_denom不等于0),则不应当用多于以下数目的比特在比特流中表示经编码的macroblock_layer():

[0066] $(128 + \text{RawMbBits}) \div \text{max_bits_per_mb_denom}$

[0067] 取决于entropy_coding_mode_flag,macroblock_layer()数据的比特如下计数:

[0068] - 如果entropy_coding_mode_flag等于0,则macroblock_layer()数据的比特数目由宏块的macroblock_layer()语法结构中的比特数目给出。

[0069] - 否则(entropy_coding_mode_flag等于1),则宏块的macroblock_layer()数据的比特数目由当解析与该宏块相关联的macroblock_layer()时在MPEG-4 AVC标准的子条款9.3.3.2.2和9.3.3.2.3中调用read_bits(1)的次数给出。

[0070] 当max_bits_permb_denom不存在时,max_bits_per_mb_enom的值应当被推断为等于1。

[0071] log2_max_mv_length_horizontal和log2_max_mv_length_vertical分别指示出,对于经编码视频序列中的所有图片,以1/4亮度样本为单位,经解码的水平和垂直运动向量分量的最大绝对值。值n断言,以1/4亮度样本位移为单位,运动向量分量的值不应当超过从 -2^n 至 2^n-1 的范围(包括 -2^n 和 2^n-1)。log2_max_mv_length_horizontal的值应当在0至16的范围中,包括0和16。log2_max_mv_length_vertical的值应当在0至16的范围中,包括0和16。当log2_max_mv_length_horizontal不存在时,log2_max_mv_length_horizontal和log2_max_mv_length_vertical的值应当被推断为等于16。应当注意,经解码的垂直或水平运动向量分量的最大绝对值也由MPEG-4 AVC标准的附录A中规定的档次(profile)和级别限度来约束。

[0072] num_reorder_frames指示在解码顺序上分别在经编码视频序列中的任何帧、互补场对或非成对场之前、而在输出顺序上则在其之后的帧、互补场对或非成对场的最大数目。

num_reorder_frames 的值应当在 0 至 max_dec_frame_buffering 的范围中,包括 0 和 max_dec_frame_buffering。当 num_reorder_frames 语法元素不存在时,num_reorder_frames 的值应当被如下推断:

[0073] - 如果 profile_idc 等于 44、100、110、122 或 244,并且 constraint_set3_flag 等于 1,则 num_reorder_frames 的值应当被推断为等于 0。

[0074] - 否则 (profile_idc 不等于 44、100、110、122 或 244,或者 constraint_set3_flag 等于 0), num_reorder_frames 的值应当被推断为等于 max_dec_frame_bufferingMaxDpbSize。

[0075] max_dec_frame_buffering 规定以帧缓冲器为单位的假设参考解码器解码图片缓冲器 (DPB) 的必需大小。经编码视频序列不应当需要大小大于 Max(1, max_dec_frame_buffering) 帧缓冲器的经解码图片缓冲器来使得能够在图片定时补充增强信息 (SEI) 消息的 dpb_output_delay 所规定的输出时间输出经解码图片。max_dec_frame_buffering 的值应当在 num_ref_frames 至 MaxDpbSize (在 MPEG-4 AVC 标准中的子条款 A. 3.1 或 A. 3.2 中规定) 的范围中,包括 num_ref_frames 和 MaxDpbSize。当 max_dec_frame_buffering 语法元素不存在时,max_dec_frame_buffering 的值应当被如下推断:

[0076] - 如果 profile_idc 等于 44 或 244,并且 constraint_set3_flag 等于 1,则 max_dec_frame_buffering 的值应当被推断为等于 0。

[0077] - 否则 (profile_idc 不等于 44 或 244,或者 constraint_set3_flag 等于 0), max_dec_frame_buffering 的值应当被推断为等于 MaxDpbSize。

[0078] 在多视角视频编码中,比特流限制参数基于更严格的限度定制了子流的解码操作。因此,应当允许为多视角视频编码的比特流每个可提取的子流规定比特流限制参数。根据一个实施例,我们提出了为每个视角、为一视角中的每个时间级别和 / 或为每个操作点规定比特流限制信息。

[0079] 为每个视角规定比特流限制参数

[0080] 可以为每个视角规定比特流限制参数。我们提出了 mvc_vui_parameters_extension 的语法,其是 subset_sequence_parameter_set 的一部分。表 2 示出了 mvc_vui_parameters_extension 的语法。

[0081] 表 2

[0082]

mvc_vui_parameters_extension() {	C	描述符
num_views_minus1	0	ue(v)
for(i = 0 ; i <= num_views_minus1 ; i++) {		
view_id[i]	0	u(3)
bitstream_restriction_flag[i]	0	u(1)

if(bitstream_restriction_flag[i]) {		
motion_vectors_over_pic_boundaries_flag[i]	0	u(1)
max_bytes_per_pic_denom[i]	0	ue(v)
max_bits_per_mb_denom[i]	0	ue(v)
log2_max_mv_length_horizontal[i]	0	ue(v)
log2_max_mv_length_vertical[i]	0	ue(v)
num_reorder_frames[i]	0	ue(v)
max_dec_frame_buffering[i]	0	ue(v)
}		
}		
}		

[0083] mvc_vui_parameters_extension() 在与此 subset_sequence_parameter_set 相关联的所有视角上循环。每个视角的 view_id 和每个视角的比特流限制参数在该循环内规定。

[0084] 比特流限制语法元素的语义如下：

[0085] bitstream_restriction_flag[i] 规定具有等于 view_id 的 view_id[i] 的视角的 bitstream_restriction_flag 的值。

[0086] motion_vectors_over_pic_boundaries_flag[i] 规定具有等于 view_id 的 view_id[i] 的视角的 motion_vectors_over_pic_boundaries_flag 的值。当 motion_vectors_over_pic_boundaries_flag[i] 语法元素不存在时，具有等于 view_id 的 view_id[i] 的视角的 motion_vectors_over_pic_boundaries_flag 应当被推断为等于 1。

[0087] max_bytes_per_pic_denom[i] 规定具有等于 view_id 的 view_id[i] 的视角的 max_bytes_per_pic_denom 值。当 max_bytes_per_pic_denom[i] 语法元素不存在时，具有等于 view_id 的 view_id[i] 的视角的 max_bytes_per_pic_denom 的值应当被推断为等于 2。

[0088] max_bits_per_mb_denom[i] 规定具有等于 view_id 的 view_id[i] 的视角的 max_bits_per_mb_denom 值。当 max_bits_per_mb_denom[i] 不存在时，具有等于 view_id 的 view_id[i] 的视角的 max_bits_per_mb_denom 的值应当被推断为等于 1。

[0089] log2_max_mv_length_horizontal[i] 和 log2_max_mv_length_vertical[i] 分别规定具有等于 view_id 的 view_id[i] 的视角的 log2_max_mv_length_horizontal 和 log2_

max_mv_length_vertical 的值。当 log2_max_mv_length_horizontal[i] 不存在时,具有等于 view_id 的 view_id[i] 的视角的 log2_max_mv_length_horizontal 和 log2_max_mv_length_vertical 的值应当被推断为等于 16。

[0090] num_reorder_frames[i] 规定具有等于 view_id 的 view_id[i] 的视角的 num_reorder_frames 的值。num_reorder_frames[i] 的值应当在 0 至 max_dec_frame_buffering 的范围中,包括 0 和 max_dec_frame_buffering。当 num_reorder_frames[i] 语法元素不存在时,具有等于 view_id 的 view_id[i] 的视角的 num_reorder_frames 的值应当被推断为等于 max_dec_frame_buffering。

[0091] max_dec_frame_buffering[i] 规定具有等于 view_id 的 view_id[i] 的视角的 max_dec_frame_buffering 的值。max_dec_frame_buffering[i] 的值应当在 num_ref_frames[i] 至 MaxDpbSize(在 MPEG-4 AVC 标准中的子条款 A. 3. 1 或 A. 3. 2 中规定)的范围中,包括 num_ref_frames[i] 和 MaxDpbSize。当 max_dec_frame_buffering[i] 语法元素不存在时,具有等于 view_id 的 view_id[i] 的视角的 max_dec_frame_buffering 的值应当被推断为等于 MaxDpbSize。

[0092] 参考图 3,用于利用 mvc_vui_parameters_extension() 语法元素来对每个视角的比特流限制参数编码的示例性方法由标号 300 总地指示。

[0093] 方法 300 包括开始块 305,该开始块将控制传递到功能块 310。功能块 310 设定变量 M 等于视角数目减 1,并将控制传递到功能块 315。功能块 315 将该变量 M 写入到比特流,并将控制传递到功能块 320。功能块 320 设定变量 i 等于零,并将控制传递到功能块 325。功能块 325 写入 view_id[i] 语法元素,并将控制传递到功能块 330。功能块 330 写入 bitstream_restriction_flag[i] 语法元素,并将控制传递到判决块 335。判决块 335 判定 bitstream_restriction_flag[i] 语法元素是否等于零。如果是,则控制被传递到判决块 345。否则,控制被传递到功能块 340。

[0094] 功能块 340 写入视角 i 的比特流限制参数,并将控制传递到判决块 345。判决块 345 判定变量 i 是否等于变量 M。如果是,则控制被传递到结束块 399。否则,控制被传递到功能块 350。

[0095] 功能块 350 设定变量 i 等于 i 加 1,并将控制返回到功能块 325。

[0096] 参考图 4,用于利用 mvc_vui_parameters_extension() 语法元素来对每个视角的比特流限制参数解码的示例性方法由标号 400 总地指示。

[0097] 方法 400 包括开始块 405,该开始块将控制传递到功能块 407。功能块 407 从比特流中读取变量 M,并将控制传递到功能块 410。功能块 410 设定视角数目等于变量 M 加 1,并将控制传递到功能块 420。功能块 420 设定变量 i 等于零,并将控制传递到功能块 425。功能块 425 读取 view_id[i] 语法元素,并将控制传递到功能块 430。功能块 430 读取 bitstream_restriction_flag[i] 语法元素,并将控制传递到判决块 435。判决块 435 判定 bitstream_restriction_flag[i] 语法元素是否等于零。如果是,则控制被传递到判决块 445。否则,控制被传递到功能块 440。

[0098] 功能块 440 读取视角 i 的比特流限制参数,并将控制传递到判决块 445。判决块 445 判定变量 i 是否等于变量 M。如果是,则控制被传递到结束块 499。否则,控制被传递到功能块 450。

[0099] 功能块 450 设定变量 i 等于 i 加 1, 并将控制返回到功能块 425。

[0100] 为每个视角的每个时间级别规定比特流限制参数

[0101] 可以为每个视角的每个时间级别规定比特流限制参数。我们提出了 `mvc_vui_parameters_extension` 的语法, 其是 `subset_sequence_parameter_set` 的一部分。表 3 示出了 `mvc_vui_parameters_extension` 的语法。

[0102] 表 3

[0103]

<code>mvc_vui_parameters_extension() {</code>	C	描述符
<code>num_views_minus1</code>	0	$ue(v)$
<code>for(i = 0; i <= num_views_minus1; i++) {</code>		
<code>view_id[i]</code>	0	$u(3)$
<code>num_temporal_layers_in_view_minus1[i]</code>	0	$ue(v)$
<code>for(i = 0; j <= num_temporal_level_in_view_minus1; j++) {</code>		
<code>temporal_id[i][j]</code>		
<code>bitstream_restriction_flag[i][j]</code>	0	$u(1)$
<code>if(bitstream_restriction_flag[i][j]) {</code>		
<code>motion_vectors_over_pic_boundaries_flag[i][j]</code>	0	$u(1)$
<code>max_bytes_per_pic_denom[i][j]</code>	0	$ue(v)$
<code>max_bits_per_mb_denom[i][j]</code>	0	$ue(v)$
<code>log2_max_mv_length_horizontal[i][j]</code>	0	$ue(v)$
<code>log2_max_mv_length_vertical[i][j]</code>	0	$ue(v)$
<code>num_reorder_frames[i][j]</code>	0	$ue(v)$
<code>max_dec_frame_buffering[i][j]</code>	0	$ue(v)$
<code>}</code>		

}		
}		
}		

[0104] 比特流限制语法元素的语义如下：

[0105] `bitstream_restriction_flag[i][j]` 规定在具有等于 `view_id` 的 `view_id[i]` 的视角中具有等于 `temporal_id` 的 `temporal_id[i][j]` 的时间级别的 `bitstream_restriction_flag` 的值。

[0106] `motion_vectors_over_pic_boundaries_flag[i][j]` 规定在具有等于 `view_id` 的 `view_id[i]` 的视角中具有等于 `temporal_id` 的 `temporal_id[i][j]` 的时间级别的 `motion_vectors_over_pic_boundaries_flag` 的值。当 `motion_vectors_over_pic_boundaries_flag[i]` 语法元素不存在时,在具有等于 `view_id` 的 `view_id[i]` 的视角中具有等于 `temporal_id` 的 `temporal_id[i][j]` 的时间级别的 `motion_vectors_over_pic_boundaries_flag` 应当被推断为等于 1。

[0107] `max_bytes_per_pic_denom[i][j]` 规定在具有等于 `view_id` 的 `view_id[i]` 的视角中具有等于 `temporal_id` 的 `temporal_id[i][j]` 的时间级别的 `max_bytes_per_pic_denom` 值。当 `max_bytes_per_pic_denom[i]` 语法元素不存在时,在具有等于 `view_id` 的 `view_id[i]` 的视角中具有等于 `temporal_id` 的 `temporal_id[i][j]` 的时间级别的 `max_bytes_per_pic_denom` 的值应当被推断为等于 2。

[0108] `max_bits_per_mb_denom[i][j]` 规定在具有等于 `view_id` 的 `view_id[i]` 的视角中具有等于 `temporal_id` 的 `temporal_id[i][j]` 的时间级别的 `max_bits_per_mb_denom` 值。当 `max_bits_per_mb_denom[i]` 不存在时,在具有等于 `view_id` 的 `view_id[i]` 的视角中具有等于 `temporal_id` 的 `temporal_id[i][j]` 的时间级别的 `max_bits_per_mb_denom` 的值应当被推断为等于 1。

[0109] `log2_max_mv_length_horizontal[i][j]` 和 `log2_max_mv_length_vertical[i][j]` 分别规定在具有等于 `view_id` 的 `view_id[i]` 的视角中具有等于 `temporal_id` 的 `temporal_id[i][j]` 的时间级别的 `log2_max_mv_length_horizontal` 和 `log2_max_mv_length_vertical` 的值。当 `log2_max_mv_length_horizontal[i]` 不存在时,在具有等于 `view_id` 的 `view_id[i]` 的视角中具有等于 `temporal_id` 的 `temporal_id[i][j]` 的时间级别的 `log2_max_mv_length_horizontal` 和 `log2_max_mv_length_vertical` 的值应当被推断为等于 16。

[0110] `num_reorder_frames[i][j]` 规定在具有等于 `view_id` 的 `view_id[i]` 的视角中具有等于 `temporal_id` 的 `temporal_id[i][j]` 的时间级别的 `num_reorder_frames` 的值。`num_reorder_frames[i]` 的值应当在 0 至 `max_dec_frame_buffering` 的范围中,包括 0 和 `max_dec_frame_buffering`。当 `num_reorder_frames[i]` 语法元素不存在时,在具有等于 `view_id` 的 `view_id[i]` 的视角中具有等于 `temporal_id` 的 `temporal_id[i][j]` 的时间级别的 `num_reorder_frames` 的值应当被推断为等于 `max_dec_frame_buffering`。

[0111] `max_dec_frame_buffering[i][j]` 规定在具有等于 `view_id` 的 `view_id[i]` 的视角中具有等于 `temporal_id` 的 `temporal_id[i][j]` 的时间级别的 `max_dec_frame_buffering` 的值。`max_dec_frame_buffering[i]` 的值应当在 `num_ref_frames[i]` 至 `MaxDpbSize` (在 MPEG-4 AVC 标准中的子条款 A. 3.1 或 A. 3.2 中规定) 的范围中,包括 `num_ref_frames[i]` 和 `MaxDpbSize`。当 `max_dec_frame_buffering[i]` 语法元素不存在时,在具有等于 `view_id` 的 `view_id[i]` 的视角中具有等于 `temporal_id` 的 `temporal_id[i][j]` 的时间级别的 `max_dec_frame_buffering` 的值应当被推断为等于 `MaxDpbSize`。

[0112] 在 `mvc_vui_parameters_extension()` 中,执行两个循环。外循环是在与 `subset_sequence_parameter_set` 相关联的所有视角上循环。在外循环中规定针对每个视角的时间级别的数目的 `view_id`。内循环在一视角的所有时间级别上循环。在内循环中规定比特流限制信息。

[0113] 参考图 5,用于利用 `mvc_vui_parameters_extension()` 语法元素来对每个视角中的每个时间级别的比特流限制参数编码的示例性方法由标号 500 总地指示。

[0114] 方法 500 包括开始块 505,该开始块将控制传递到功能块 510。功能块 510 设定变量 `M` 等于视角数目减 1,并将控制传递到功能块 515。功能块 515 将该变量 `M` 写入到比特流,并将控制传递到功能块 520。功能块 520 设定变量 `i` 等于零,并将控制传递到功能块 525。功能块 525 写入 `view_id[i]` 语法元素,并将控制传递到功能块 530。功能块 530 设定变量 `N` 等于视角 `i` 中的时间级别的数目减 1,并将控制传递到功能块 535。功能块 535 将该变量 `N` 写入到比特流,并将控制传递到功能块 540。功能块 540 设定变量 `j` 等于零,并将控制传递到功能块 545。功能块 545 写入 `temporal_id[i][j]` 语法元素,并将控制传递到功能块 550。功能块 550 写入 `bitstream_restriction_flag[i][j]` 语法元素,并将控制传递到判决块 555。判决块 555 判定 `bitstream_restriction_flag[i][j]` 语法元素是否等于零。如果是,则控制被传递到判决块 565。否则,控制被传递到功能块 560。

[0115] 功能块 560 写入视角 `i` 中的时间级别 `j` 的比特流限制参数,并将控制传递到判决块 565。判决块 565 判定变量 `j` 是否等于变量 `N`。如果是,则控制被传递到判决块 570。否则控制被传递到功能块 575。

[0116] 判决块 570 判定变量 `i` 是否等于变量 `M`。如果是,则控制被传递到结束块 599。否则,控制被传递到功能块 580。

[0117] 功能块 580 设定变量 `i` 等于 `i` 加 1,并将控制返回到功能块 525。

[0118] 功能块 575 设定变量 `j` 等于 `j` 加 1,并将控制返回到功能块 545。

[0119] 参考图 6,用于利用 `mvc_vui_parameters_extension()` 语法元素来对每个视角中的每个时间级别的比特流限制参数解码的示例性方法由标号 600 总地指示。

[0120] 方法 600 包括开始块 605,该开始块将控制传递到功能块 607。功能块 607 从比特流中读取变量 `M`,并将控制传递到功能块 610。功能块 610 设定视角数目等于变量 `M` 加 1,并将控制传递到功能块 620。功能块 620 设定变量 `i` 等于零,并将控制传递到功能块 625。功能块 625 读取 `view_id[i]` 语法元素,并将控制传递到功能块 627。功能块 627 从比特流中读取变量 `N`,并将控制传递到功能块 630。功能块 630 设定视角 `i` 中的时间级别的数目等于 `N` 加 1,并将控制传递到功能块 640。功能块 640 设定变量 `j` 等于零,并将控制传递到功能块 645。功能块 645 读取 `temporal_id[i][j]` 语法元素,并将控制传递到功能块 650。功能块

650 读取 bitstream_restriction_flag[i][j] 语法元素, 并将控制传递到判决块 655。判决块 655 判定 bitstream_restriction_flag[i][j] 语法元素是否等于零。如果是, 则控制被传递到判决块 665。否则, 控制被传递到功能块 660。

[0121] 功能块 660 读取视角 i 中的时间级别 j 的比特流限制参数, 并将控制传递到判决块 665。判决块 665 判定变量 j 是否等于变量 N。如果是, 则控制被传递到判决块 670。否则, 控制被传递到功能块 675。

[0122] 判决块 670 判定变量 i 是否等于变量 M。如果是, 则控制被传递到结束块 699。否则, 控制被传递到功能块 680。

[0123] 功能块 680 设定变量 i 等于 i 加 1, 并将控制返回到功能块 625。

[0124] 功能块 675 设定变量 j 等于 j 加 1, 并将控制返回到功能块 645。

[0125] 为每个操作点规定比特流限制参数

[0126] 可以为每个操作点规定比特流限制参数。我们提出了在视角可缩放性信息 SEI 消息中传达每个操作点的比特流限制参数。视角可缩放性信息 SEI 消息的语法可如表 4 那样被修改。比特流限制信息的语法被插入在一个在所有操作点上循环的循环中。

[0127] 表 4

[0128]

view_scalability_info(payloadSize) {	C	描述符
num_operation_points_minus1	5	ue(v)
for(i = 0; i <= num_operation_points_minus1; i++) {		
operation_point_id[i]	5	ue(v)
priority_id[i]	5	u(5)
temporal_id[i]	5	u(3)
num_active_views_minus1[i]	5	ue(v)
for(j = 0; j <= num_active_views_minus1[i]; j++)		
view_id[i][j]	5	ue(v)
profile_level_info_present_flag[i]	5	u(1)
bitrate_info_present_flag[i]	5	u(1)
frm_rate_info_present_flag[i]	5	u(1)
op_dependency_info_present_flag[i]	5	u(1)

init_parameter_sets_info_present_flag[i]	5	u(1)
bitstream_restriction_flag[i]		
if(profile_level_info_present_flag[i]){		
op_profile_idc[i]	5	u(8)
op_constraint_set0_flag[i]	5	u(1)
op_constraint_set1_flag[i]	5	u(1)
op_constraint_set2_flag[i]	5	u(1)
op_constraint_set3_flag[i]	5	u(1)
reserved_zero_4bits/*equal to 0*/	5	u(4)
op_level_idc[i]	5	u(8)
}else		
profile_level_info_src_op_id_delta[i]		ue(v)
if(bitrate_info_present_flag[i]){		
avg_bitrate[i]	5	u(16)
max_bitrate[i]	5	u(16)
max_bitrate_calc_window[i]	5	u(16)
}		
if(frm_rate_info_present_flag[i]){		
constant_frm_rate_idc[i]	5	u(2)
avg_frm_rate[i]	5	u(16)
}else		
frm_rate_info_src_op_id_delta[i]	5	ue(v)
if(op_dependency_info_present_flag[i]){		

num_directly_dependent_ops[i]	5	ue(v)
for(j = 0 ; j < num_directly_dependent_ops[i] ; j++) {		
directly_dependent_op_id_delta_minus1[i][j]	5	ue(v)
}else		
op_dependency_info_src_op_id_delta[i]	5	ue(v)
if(init_parameter_sets_info_present_flag[i]) {		
num_init_seq_parameter_set_minus1[i]	5	ue(v)
for(j = 0 ; j <= num_init_seq_parameter_set_minus1[i] ; j++)		
init_seq_parameter_set_id_delta[i][j]	5	ue(v)
num_init_pic_parameter_set_minus1[i]	5	ue(v)
for(j = 0 ; j <= num_init_pic_parameter_set_minus1[i] ; j++)		
init_pic_parameter_set_id_delta[i][j]	5	ue(v)
}else		
init_parameter_sets_info_src_op_id_delta[i]	5	ue(v)
if(bitstream_restriction_flag[i]) {		
motion_vectors_over_pic_boundaries_flag[i]	0	u(1)
max_bytes_per_pic_denom[i]	0	ue(v)
max_bits_per_mb_denom[i]	0	ue(v)
log2_max_mv_tength_horizontal[i]	0	ue(v)
log2_max_mv_length_vertical[i]	0	ue(v)
num_reorder_frames[i]	0	ue(v)
max_dec_frame_buffering[i]	0	ue(v)
}		

}		
}		

[0129]

[0130] 比特流限制语法元素的语义如下：

[0131] `bitstream_restriction_flag[i]` 规定具有等于 `operation_point_id` 的 `operation_point_id[i]` 的操作点的 `bitstream_restriction_flag` 的值。

[0132] `motion_vectors_over_pic_boundaries_flag[i]` 规定具有等于 `operation_point_id` 的 `operation_point_id[i]` 的操作点的 `motion_vectors_over_pic_boundaries_flag` 的值。当 `motion_vectors_over_pic_boundaries_flag[i]` 语法元素不存在时，具有等于 `operation_point_id` 的 `operation_point_id[i]` 的操作点的 `motion_vectors_over_pic_boundaries_flag` 应当被推断为等于 1。

[0133] `max_bytes_per_pic_denom[i]` 规定具有等于 `operation_point_id` 的 `operation_point_id[i]` 的操作点的 `max_bytes_per_pic_denom` 值。当 `max_bytes_per_pic_denom[i]` 语法元素不存在时，具有等于 `operation_point_id` 的 `operation_point_id[i]` 的操作点的 `max_bytes_per_pic_denom` 的值应当被推断为等于 2。

[0134] `max_bits_per_mb_denom[i]` 规定具有等于 `operation_point_id` 的 `operation_point_id[i]` 的操作点的 `max_bits_per_mb_denom` 值。当 `max_bits_per_mb_denom[i]` 不存在时，具有等于 `operation_point_id` 的 `operation_point_id[i]` 的操作点的 `max_bits_per_mb_denom` 的值应当被推断为等于 1。

[0135] `log2_max_mv_length_horizontal[i]` 和 `log2_max_mv_length_vertical[i]` 分别规定具有等于 `operation_point_id` 的 `operation_point_id[i]` 的操作点的 `log2_max_mv_length_horizontal` 和 `log2_max_mv_length_vertical` 的值。当 `log2_max_mv_length_horizontal[i]` 不存在时，具有等于 `operation_point_id` 的 `operation_point_id[i]` 的操作点的 `log2_max_mv_length_horizontal` 和 `log2_max_mv_length_vertical` 的值应当被推断为等于 16。

[0136] `num_reorder_frames[i]` 规定具有等于 `operation_point_id` 的 `operation_point_id[i]` 的操作点的 `num_reorder_frames` 的值。`num_reorder_frames[i]` 的值应当在 0 至 `max_dec_frame_buffering` 的范围中，包括 0 和 `max_dec_frame_buffering`。当 `num_reorder_frames[i]` 语法元素不存在时，具有等于 `operation_point_id` 的 `operation_point_id[i]` 的操作点的 `num_reorder_frames` 的值应当被推断为等于 `max_dec_frame_buffering`。

[0137] `max_dec_frame_buffering[i]` 规定具有等于 `operation_point_id` 的 `operation_point_id[i]` 的操作点的 `max_dec_frame_buffering` 的值。`max_dec_frame_buffering[i]` 的值应当在 `num_ref_frames[i]` 至 `MaxDpbSize`（在 MPEG-4 AVC 标准中的子条款 A.3.1 或 A.3.2 中规定）的范围中，包括 `num_ref_frames[i]` 和 `MaxDpbSize`。当 `max_dec_frame_buffering[i]` 语法元素不存在时，具有等于 `operation_point_id` 的 `operation_point_id[i]` 的操作点的 `max_dec_frame_buffering` 的值应当被推断为等于 `MaxDpbSize`。

[0138] 参考图 7,用于利用 `view_scalability_parameters_extension()` 语法元素来对每个操作点的比特流限制参数编码的示例性方法由标号 700 总地指示。

[0139] 方法 700 包括开始块 705,该开始块将控制传递到功能块 710。功能块 710 设定变量 M 等于操作点数目减 1,并将控制传递到功能块 715。功能块 715 将该变量 M 写入到比特流,并将控制传递到功能块 720。功能块 720 设定变量 i 等于零,并将控制传递到功能块 725。功能块 725 写入 `operation_point_id[i]` 语法元素,并将控制传递到功能块 730。功能块 730 写入 `bitstream_restriction_flag[i]` 语法元素,并将控制传递到判决块 735。判决块 735 判定 `bitstream_restriction_flag[i]` 语法元素是否等于零。如果是,则控制被传递到判决块 745。否则,控制被传递到功能块 740。

[0140] 功能块 740 写入操作点 i 的比特流限制参数,并将控制传递到判决块 745。判决块 745 判定变量 i 是否等于变量 M 。如果是,则控制被传递到结束块 799。否则,控制被传递到功能块 750。

[0141] 功能块 750 设定变量 i 等于 i 加 1,并将控制返回到功能块 725。参考图 8,用于利用 `view_scalability_parameters_extension()` 语法元素来对每个操作点的比特流限制参数解码的示例性方法由标号 800 总地指示。

[0142] 方法 800 包括开始块 805,该开始块将控制传递到功能块 807。功能块 807 从比特流中读取变量 M ,并将控制传递到功能块 810。功能块 810 设定操作点数目等于变量 M 加 1,并将控制传递到功能块 820。功能块 820 设定变量 i 等于零,并将控制传递到功能块 825。功能块 825 读取 `operation_point_id[i]` 语法元素,并将控制传递到功能块 830。功能块 830 读取 `bitstream_restriction_flag[i]` 语法元素,并将控制传递到判决块 835。判决块 835 判定 `bitstream_restriction_flag[i]` 语法元素是否等于零。如果是,则控制被传递到判决块 845。否则,控制被传递到功能块 840。

[0143] 功能块 840 读取操作点 i 的比特流限制参数,并将控制传递到判决块 845。判决块 845 判定变量 i 是否等于变量 M 。如果是,则控制被传递到结束块 899。否则,控制被传递到功能块 850。

[0144] 功能块 850 设定变量 i 等于 i 加 1,并将控制返回到功能块 825。

[0145] 现在将描述本发明的许多伴随优点 / 特征中的一些,其中一些已经在上文中提及了。例如,一个优点 / 特征是一种装置,该装置包括一编码器,用于通过为各个视角、一视角中的各个时间级别和各个操作点中的至少一种规定视频可用性信息来对多视角视频内容编码。

[0146] 另一个优点 / 特征在于如上所述的具有编码器的装置,其中参数是在至少一个高级别语法元素中规定的。

[0147] 另外,另一个优点 / 特征在于如上所述的具有编码器的装置,其中至少高级别语法元素包括 `Mvc_vui_parameters_extension()` 语法元素、`Mvc_scalability_info` 补充增强信息语法消息、序列参数集、图片参数集以及补充增强信息的至少一部分中的至少一种。

[0148] 此外,另一个优点 / 特征在于如上所述的具有编码器的装置,其中所述视频可用性信息的至少一部分包括比特流限制参数。

[0149] 有关领域的普通技术人员基于这里的教导可以很容易确定本发明的这些和其他特征和优点。应当理解,本发明的教导可以用各种形式的硬件、软件、固件、专用处理器或其

组合来实现。

[0150] 最优选的是本发明的教导被实现为硬件和软件的组合。另外,软件可以实现为有形地包含在程序存储单元上的应用程序。应用程序可以被上载到包括任何适当的体系结构的机器并被该机器所执行。优选地,该机器是在具有诸如以下硬件的计算机平台上实现的:一个或多个中央处理单元(“CPU”)、随机存取存储器(“RAM”)以及输入/输出(“I/O”)接口。该计算机平台还可包括操作系统和微指令代码。这里描述的各种过程和功能可以是微指令代码的一部分或者应用程序的一部分,或者是其任何组合(可由CPU执行)。此外,各种其他的外围单元可连接到该计算机平台,例如附加的数据存储单元和打印单元。

[0151] 还应当理解,因为附图中示出的构成系统组件和方法中的一些优选地是用软件实现的,因此系统组件或过程功能块之间的实际连接可能根据对本发明编程的方式而有所不同。有了这里的教导,有关领域的普通技术人员将能够想出本发明的这些和类似实现方式或配置。

[0152] 虽然这里已经参考附图描述了示例性的实施例,但是应当理解,本发明并不限于这些确切的实施例,有关领域的普通技术人员可在其中实现各种变化和修改,而不脱离本发明的精神或范围。所有这种变化和修改都意图被包括在权利要求所限定的本发明的范围之内。

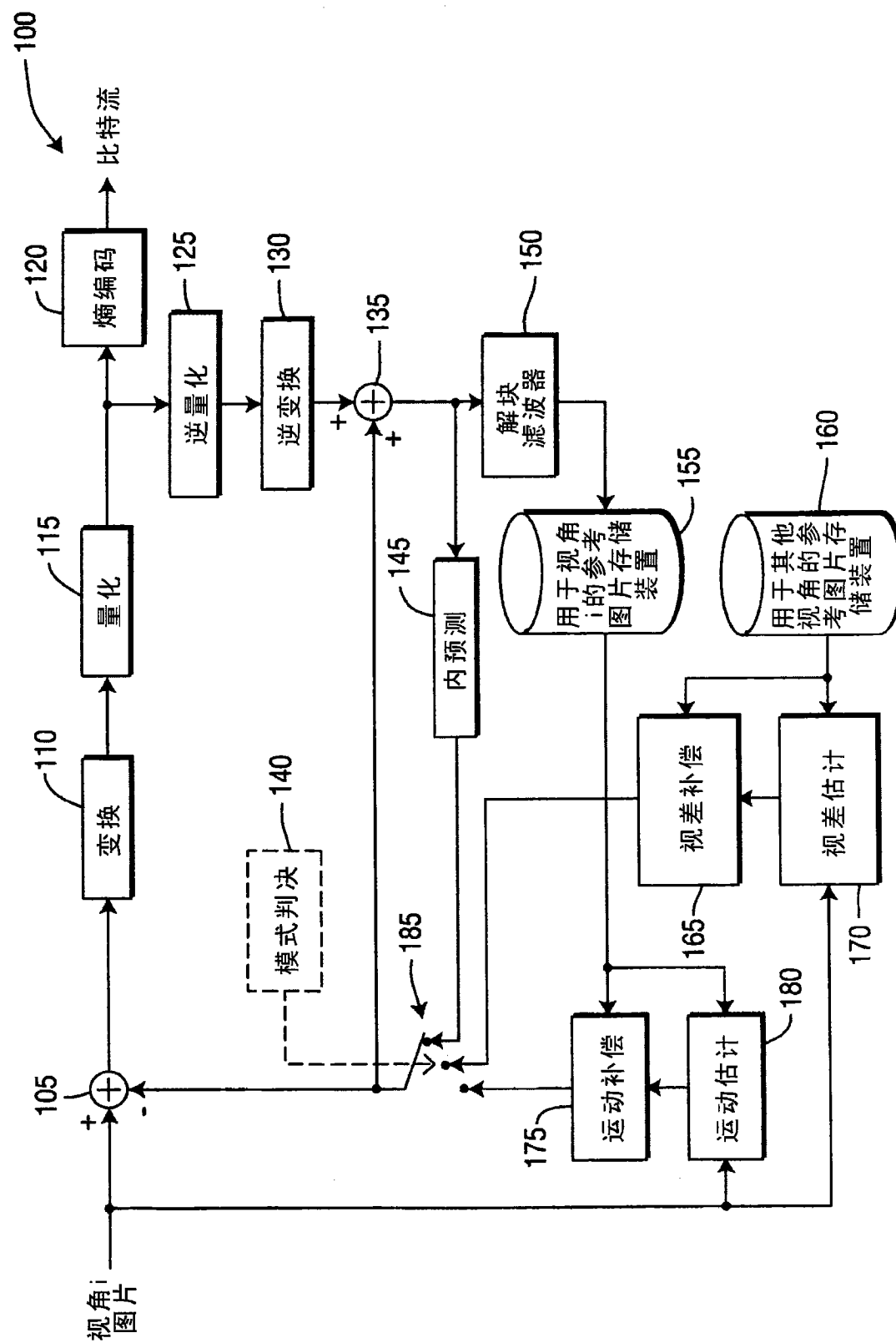


图 1

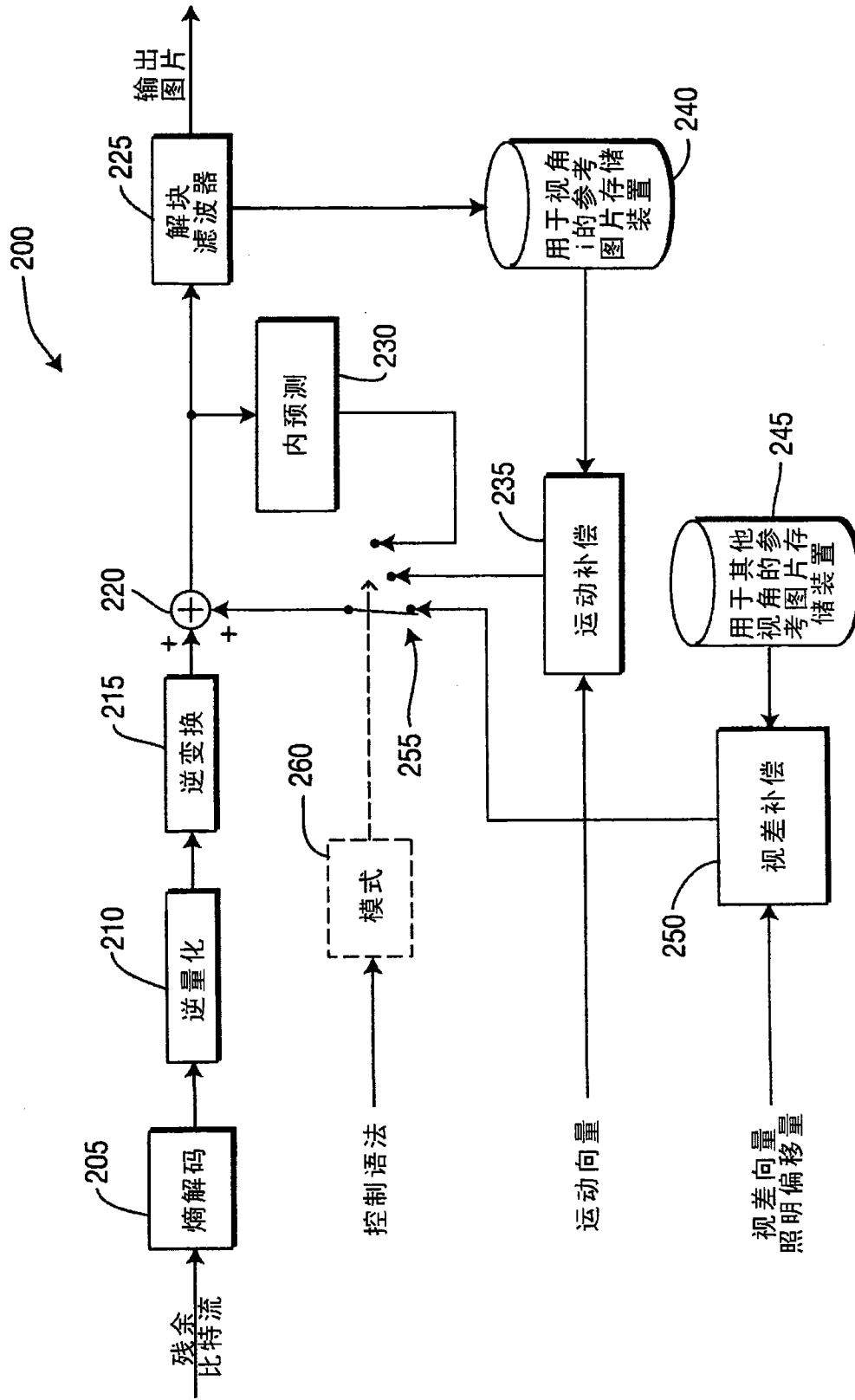


图 2

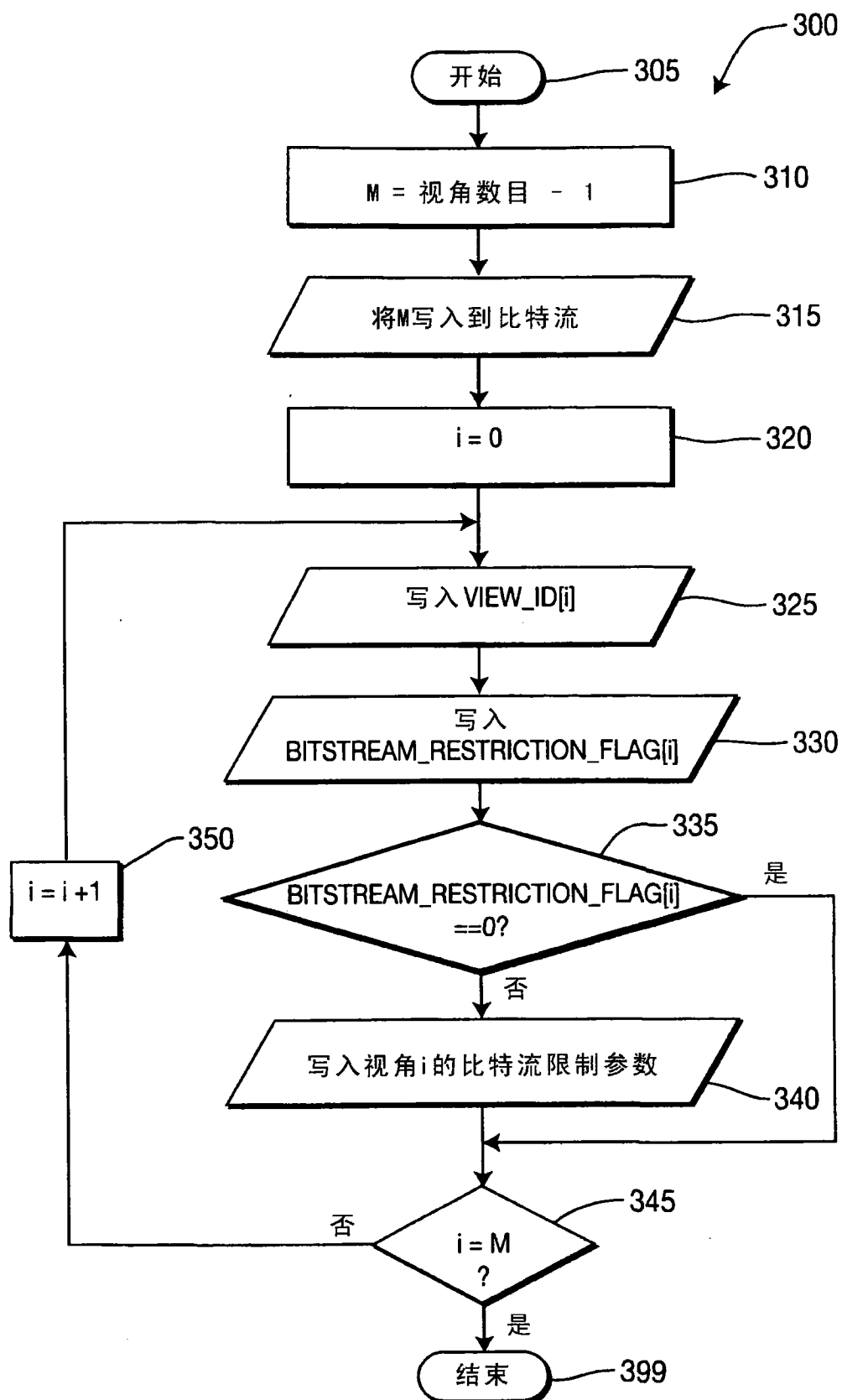


图 3

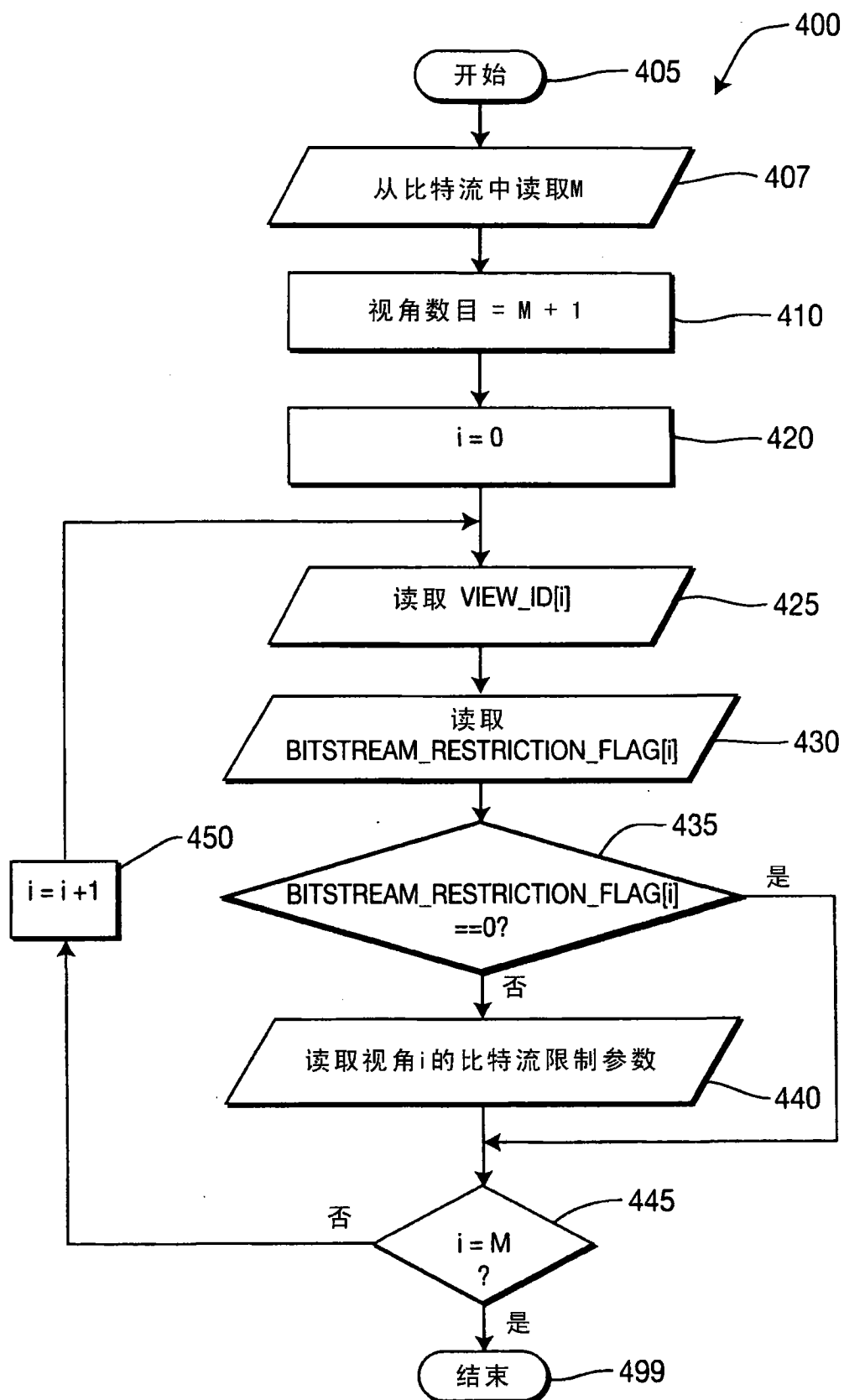


图 4

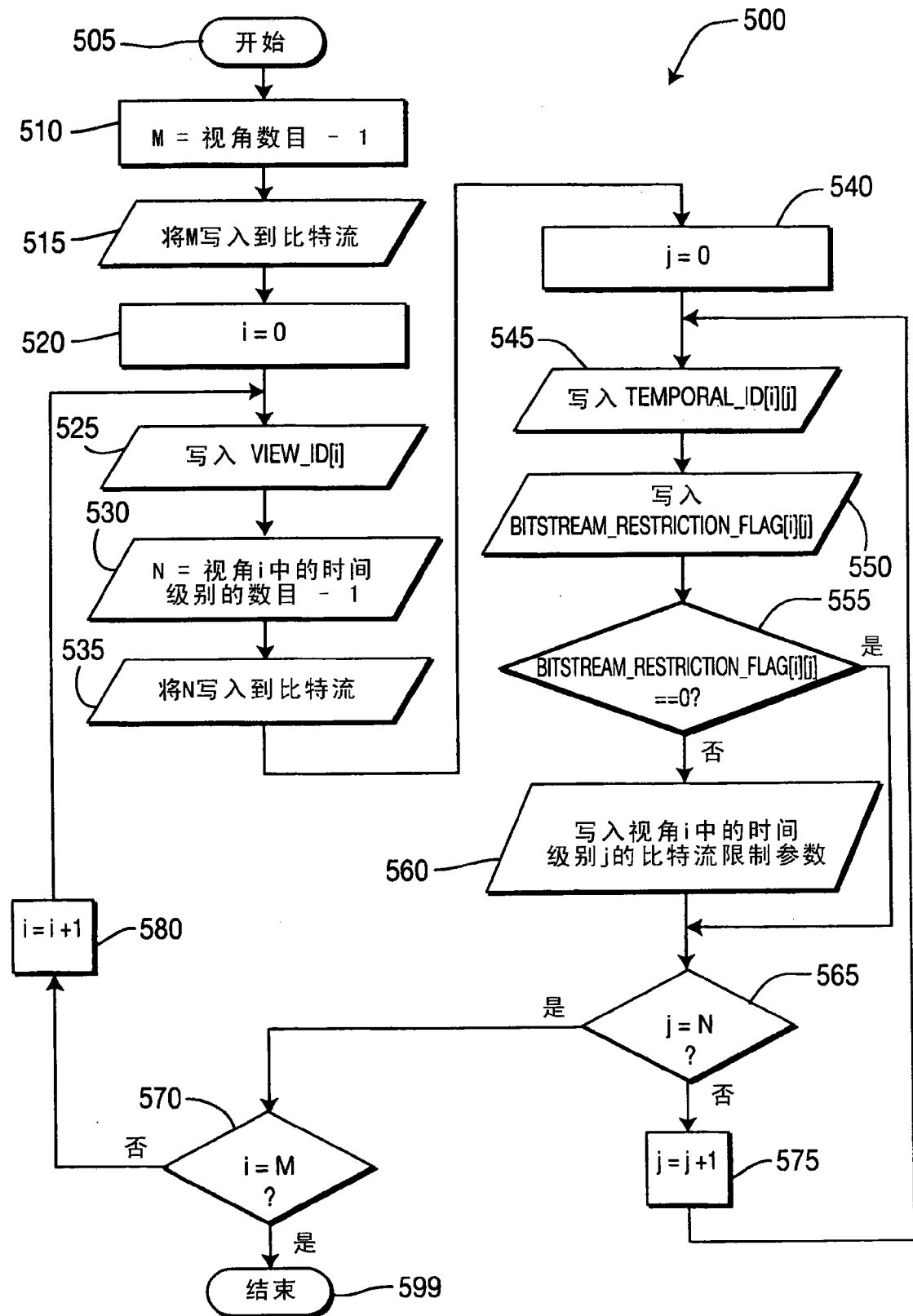


图 5

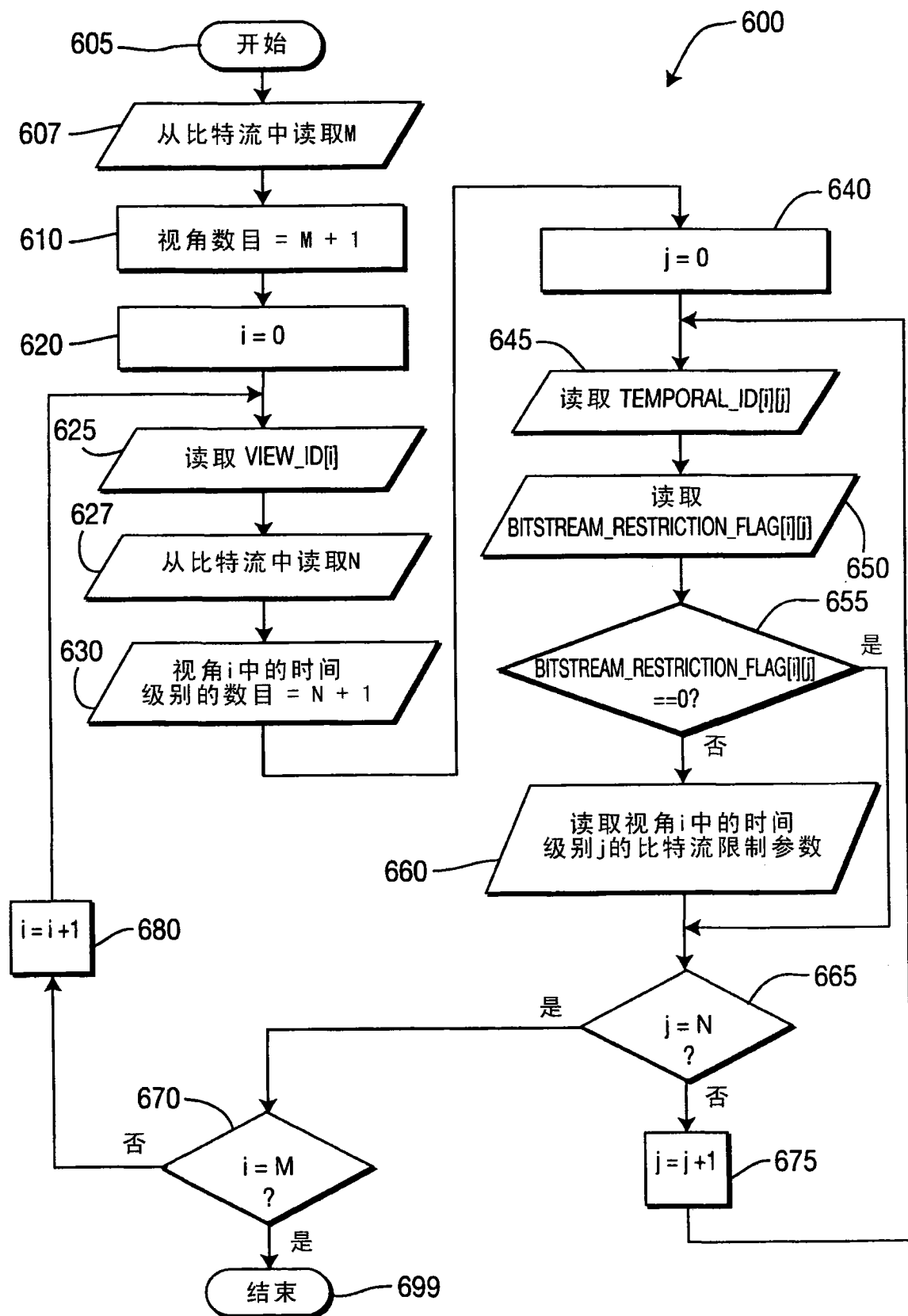


图 6

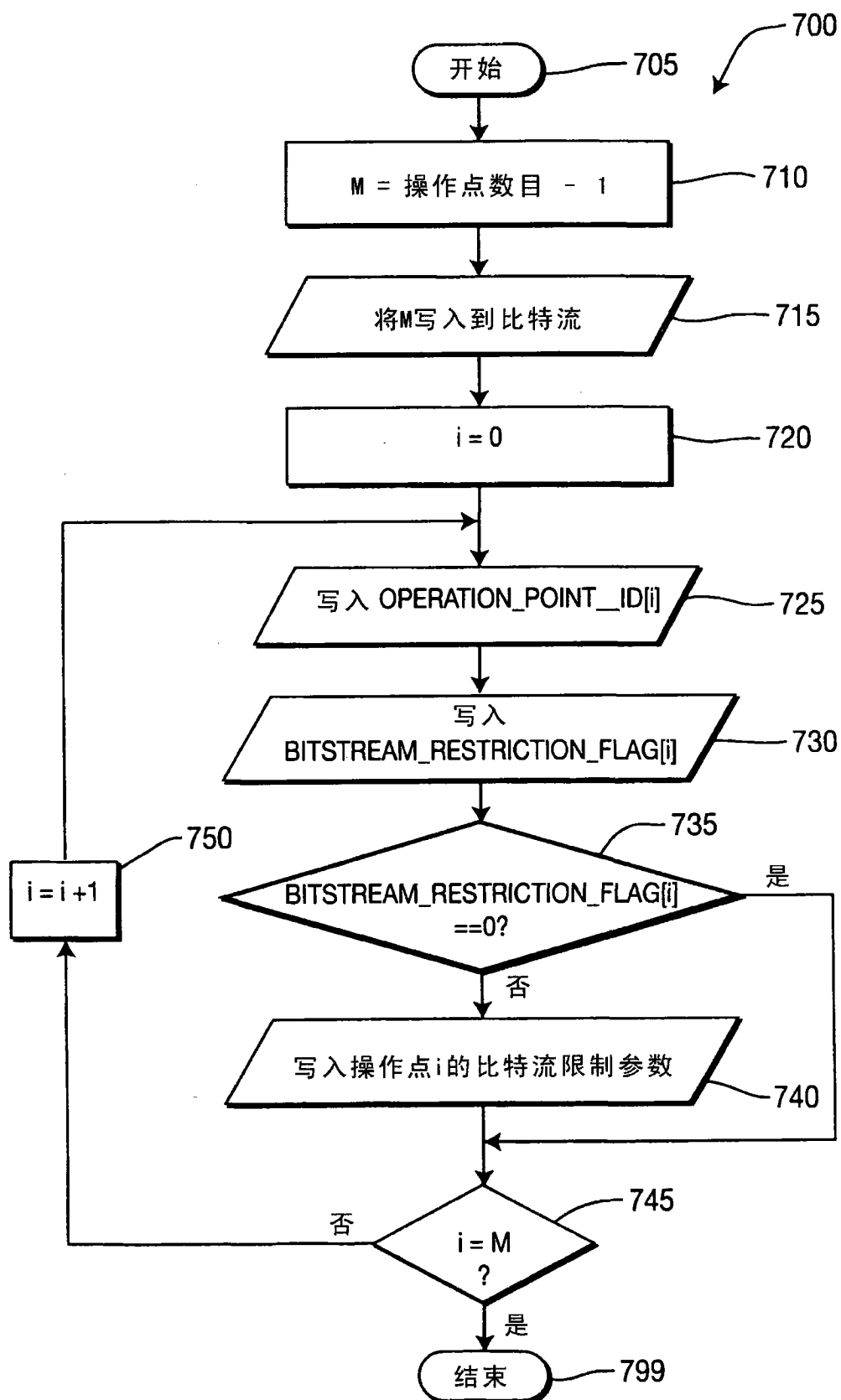


图 7

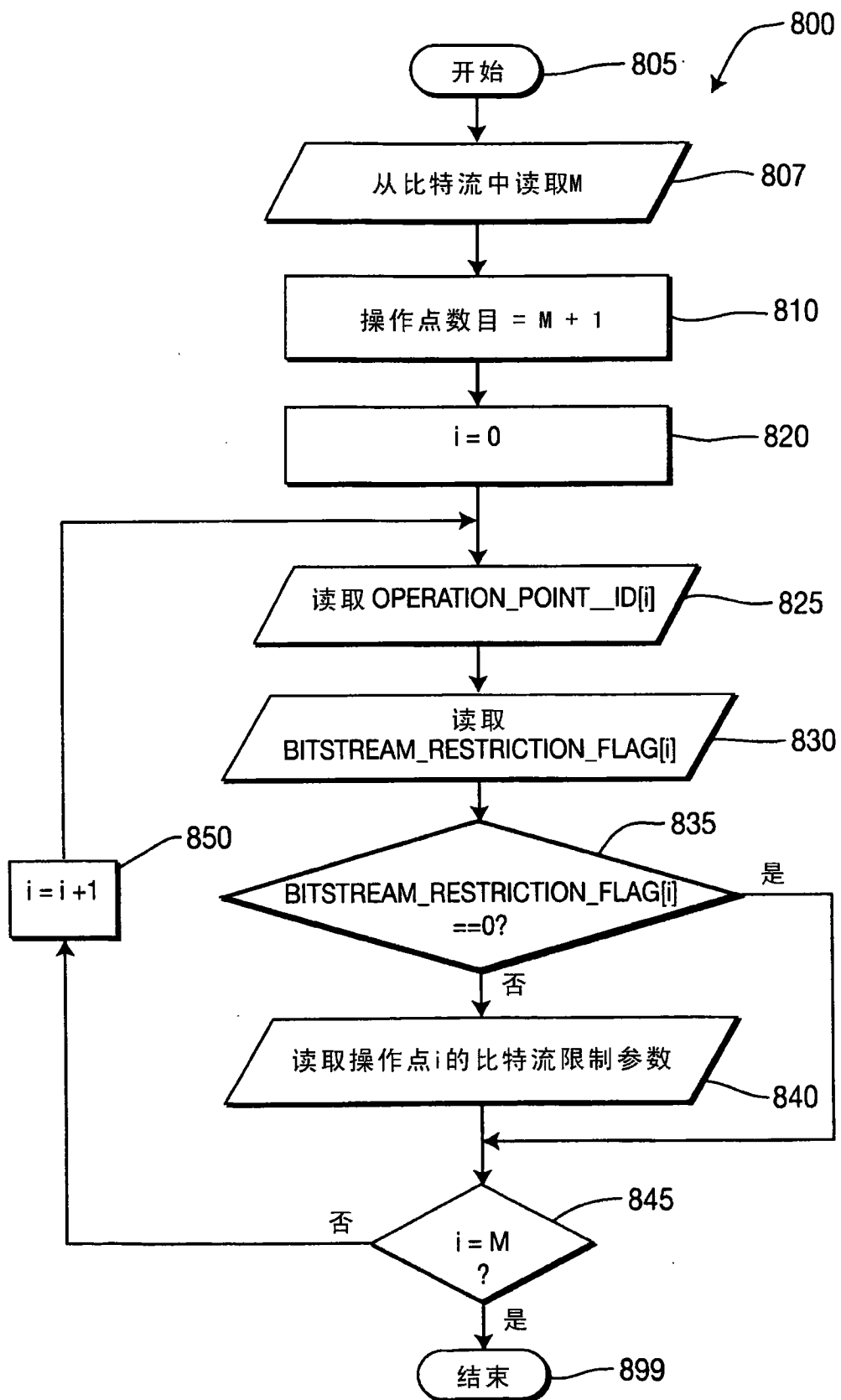


图 8