

[19] 中华人民共和国国家知识产权局

[51] Int. Cl.

H04N 7/26 (2006.01)

H04N 7/50 (2006.01)



[12] 发明专利申请公布说明书

[21] 申请号 200880022424.0

[43] 公开日 2010 年 3 月 31 日

[11] 公开号 CN 101690230A

[22] 申请日 2008.6.24

[21] 申请号 200880022424.0

[30] 优先权

[32] 2007.6.28 [33] US [31] 60/946,932

[86] 国际申请 PCT/US2008/007894 2008.6.24

[87] 国际公布 WO2009/005658 英 2009.1.8

[85] 进入国家阶段日期 2009.12.28

[71] 申请人 汤姆森特许公司

地址 法国布洛涅-比扬库尔

[72] 发明人 珀文·B·潘迪特 尹 澎

[74] 专利代理机构 北京市柳沈律师事务所

代理人 吕晓章

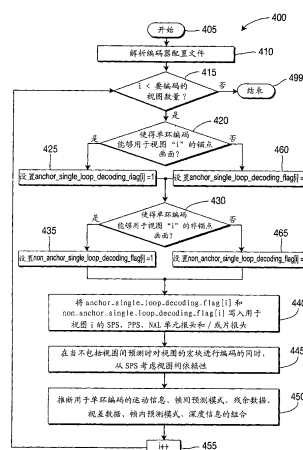
权利要求书 3 页 说明书 16 页 附图 7 页

[54] 发明名称

多视图编码视频的单环解码

[57] 摘要

提供了在编码器和解码器处用于支持多视图编码视频的单环解码的方法和装置。装置包括：编码器(100)，用于对多视图视频内容进行编码，以使得当使用视图间预测对多视图视频内容进行编码时能够进行多视图视频内容的单环解码。类似地，还描述了用于对多视图视频内容进行编码、以在使用视图间预测对多视图视频内容进行编码时支持多视图视频内容的单环解码的方法(400)。还描述了对应的解码器(200)装置和方法(500)。



1.一种装置，包括：

编码器（100），用于对多视图视频内容进行编码，以使得当使用视图间预测对所述多视图视频内容进行编码时能够进行所述多视图视频内容的单环解码。

2.如权利要求1所述的装置，其中，所述多视图视频内容包括参考视图和其它视图，所述其它视图能够被重构而无需所述参考视图的完全重构。

3.如权利要求1所述的装置，其中，所述视图间预测包括：从所述多视图视频内容的参考视图推断运动信息、帧间预测模式、帧内预测模式、参考索引、残余数据、深度信息、照度补偿偏移、去块强度和视差数据中的至少一个。

4.如权利要求1所述的装置，其中，所述视图间预测包括：从如下这样的特性推断用于所述多视图内容的给定视图的信息，所述特性与来自关于所述给定视图的所述多视图视频内容的参考视图的至少一个画面的至少一部分中的至少一个有关；并且对与所述至少一个画面的至少一部分有关的信息进行解码。

5.如权利要求1所述的装置，其中，使用高级语法元素来指示使得所述单环解码能够用于所述多视图视频内容。

6.如权利要求5所述的装置，其中，所述高级语法元素分别进行以下之一：指示是否使得所述单环解码能够用于所述多视图视频内容中的锚点画面和非锚点画面，基于视图指示是否使得能够进行所述单环解码，基于序列指示是否使得能够进行所述单环解码，指示仅对于所述多视图视频内容中的非锚点画面使得能够进行所述单环解码。

7.一种方法，包括：

对多视图视频内容进行编码，以当使用视图间预测对所述多视图视频内容进行编码时，支持所述多视图视频内容的单环解码（420）。

8.如权利要求7所述的方法，其中，所述多视图视频内容包括参考视图和其它视图，所述其它视图能够被重构而无需所述参考视图的完全重构（450）。

9.如权利要求7所述的方法，其中，所述视图间预测包括：从多视图视频内容的参考视图推断运动信息、帧间预测模式、帧内预测模式、参考索引、

残余数据、深度信息、照度补偿偏移、去块强度和视差数据中的至少一个(450)。

10.如权利要求7所述的方法,其中,所述视图间预测包括:从如下这样的特性推断用于所述多视图内容的给定视图的信息,所述特性与来自关于所述给定视图的所述多视图视频内容的参考视图的至少一个画面的至少一部分中的至少一个有关;并且对与所述至少一个画面的所述至少一部分有关的信息进行解码。

11.如权利要求7所述的方法,其中,使用高级语法元素来指示使得所述单环解码能够用于所述多视图视频内容(425,460)。

12.如权利要求11所述的方法,其中,所述高级语法元素分别进行以下之一:指示是否使得所述单环解码能够用于所述多视图视频内容中的锚点画面和非锚点画面(425,435,460,465),基于视图指示是否使得能够进行所述单环解码(420,425,435,460,465),基于序列指示是否使得能够进行所述单环解码(615,620,625,630,665,660),并且指示仅对于所述多视图视频内容中的非锚点画面使得能够进行所述单环解码。

13.一种在其上编码有视频信号数据的存储介质,包括:

多视图视频内容,其被编码以当使用视图间预测对所述多视图视频内容进行编码时,支持所述多视图视频内容的单环解码。

14.如权利要求13所述的存储介质,其中,所述多视图视频内容包括参考视图和其它视图,所述其它视图能够被重构而无需所述参考视图的完全重构。

15.权利要求13所述的存储介质,其中,所述视图间预测包括:从所述多视图视频内容的参考视图推断运动信息、帧间预测模式、帧内预测模式、参考索引、残余数据、深度信息、照度补偿偏移、去块强度和视差数据中的至少一个。

16.如权利要求13所述的存储介质,其中,所述视图间预测包括:从如下这样的特性推断用于所述多视图内容的给定视图的信息,所述特性与来自关于所述给定视图的所述多视图视频内容的参考视图的至少一个画面的至少一部分中的至少一个有关;并且对与所述至少一个画面的所述至少一部分有关的信息进行解码。

17.如权利要求13所述的存储介质,其中,使用高级语法元素来指示使得所述单环解码能够用于所述多视图视频内容。

18.如权利要求 17 所述的存储介质，其中，所述高级语法元素分别进行以下之一：指示是否使得所述单环解码能够用于所述多视图视频内容中的锚点画面和非锚点画面，基于视图指示是否使得能够进行所述单环解码，基于序列指示是否使得能够进行所述单环解码，并且指示仅对于所述多视图视频内容中的非锚点画面使得能够进行所述单环解码。

多视图编码视频的单环解码

相关申请的交叉引用

本申请要求于 2007 年 6 月 28 日提交的序列号为 60/946,932 的美国临时申请的权益,通过引用将其整体合并于此。此外,本申请涉及题为“METHODS AND APPARATUS AT AN ENCODER AND DECODER FOR SUPPORTING SINGLE LOOP DECODING OF MULTI-VIEW CODED VIDEO”的代理案号 PU080067 的非临时申请,其是共同受让的,在此通过引用而被并入,并且与本申请同时提交。

技术领域

本原理通常涉及视频编码和解码,更具体地,涉及在编码器和解码器处用于支持多视图编码视频的单环解码的方法和装置。

背景技术

多视图视频编码(MVC)服务于广泛的各種应用,包括自由视点和三维(3D)视频应用、家庭娱乐以及监视。在这些多视图应用中,所涉及的视频数据量是巨大的。

由于多视图视频源包括相同或类似的场景的多个视图,因此在多视图图像之间存在高度的相关性。因此,除了时间冗余之外,还可以利用视图冗余,并且通过跨相同或类似的场景的不同视图执行视图预测来实现视图冗余。

在第一现有技术途径中,提出了运动跳跃模式以改进用于 MVC 的编码效率。该第一现有技术途径源于在两个相邻视图之间关于运动存在相似性的构思。

运动跳跃模式直接从相同时间瞬时处的相邻视图中的对应宏块推断运动信息(如宏块类型、运动向量和参考索引)。该方法分解为下述两个阶段:(1)搜索对应的宏块;和(2)获得(derivation)运动信息。在第一阶段,使用全局视差向量(GDV)来指示相邻视图的画面中的对应位置(宏块)。通过当前画面与相邻视图的画面之间的单元的宏块大小来测量该全局视差向量。可以周期性地、例如每锚点(anchor)画面地对全局视差向量进行估计并且对其进行解

码。在此情况下,使用来自锚点画面的新近的全局视差向量来插入非锚点画面的全局视差向量。在第二阶段,从相邻视图的画面中的对应宏块获得运动信息,并且将该运动信息应用到当前宏块。由于第一现有技术途径的所提出的方法利用来自相邻视图的画面来表示用于视图间预测处理的另一方式,因此在当前宏块处于基本视图的画面中或处于在联合多视图视频模型(JMVM)中定义的锚点画面中的情况下,禁用运动跳跃模式。

为了向解码器通知运动跳跃模式的使用,在用于多视图视频编码的宏块层语法元素的报头中包括 motion_skip_flag。如果使得能够进行 motion_skip_flag,则当前宏块从相邻视图中的对应宏块获得宏块类型、运动向量和参考索引。

然而,在实际情形中,将使用不同类的照相机(camera)或尚未完全校准的照相机来建立包括大量照相机的多视图视频系统。在如此多的照相机的情况下,解码器的存储器需求以及复杂度可能显著地增加。此外,某些应用可以仅要求对来自一组视图的一些视图进行解码。结果,可以无需完全地重构输出不需要的视图。

发明内容

针对在编码器和解码器处用于支持多视图编码视频的单环解码的方法和装置的本原理处理现有技术的这些和其它缺点和劣势。

根据本原理的一方面,提供一种装置。该装置包括:编码器,用于对多视图视频内容进行编码,以使得当使用视图间预测对多视图视频内容进行编码时,能够进行多视图视频内容的单环解码。

根据本原理的另一方面,提供一种方法。该方法包括:对多视图视频内容进行编码以使得当使用视图间预测对多视图视频内容进行编码时,支持多视图视频内容的单环解码。

根据本原理的又一方面,提供一种装置。该装置包括:解码器,用于当使用视图间预测对多视图视频内容进行编码时,使用单环解码对多视图视频内容进行解码。

根据本原理的再一方面,提供一种方法。该方法包括:当使用视图间预测对多视图视频内容进行编码时,使用单环解码对多视图视频内容进行解码。

从将结合附图阅读的示例性实施例的下述详细的描述中,本原理的这些和其它方面、特点和优点将变得明显。

附图说明

根据以下示例性附图可以更好地理解本原理，其中：

图 1 是根据本原理实施例的可以应用本原理的示例性多视图视频编码（MVC）编码器的方框图；

图 2 是根据本原理实施例的可以应用本原理的示例性多视图视频编码（MVC）解码器的方框图；

图 3 是根据本原理实施例的可以应用本原理的具有 8 视图的示例性 MVC 系统的编码结构的图；

图 4 是根据本原理实施例的用于对多视图视频内容进行编码以支持单环解码的示例性方法的流程图；

图 5 是根据本原理实施例的用于多视图视频内容的单环解码的示例性方法的流程图；

图 6 是根据本原理实施例的用于对多视图视频内容进行编码以支持单环解码的另一示例性方法的流程图；和

图 7 是根据本原理实施例的用于多视图视频内容的单环解码的另一示例性方法的流程图。

具体实施方式

本原理针对在编码器和解码器处用于支持多视图编码视频的单环解码的方法和装置。

本描述例示本原理。因此应理解本领域技术人员将能够设计出虽然未在这里明确描述或示出、但是体现本原理并且包括在本原理的精神和范围内的各种布置。

这里所叙述的所有例子和条件语句意在教学目的，以帮助读者理解由发明人贡献以促进本领域技术的本原理和构思，并被解释为不限制为这样具体叙述的例子和条件。

此外，这里叙述本原理的原理、方面、以及实施例的所有陈述、以及本原理的具体例子意在包含本原理的结构和功能的等价物。另外，意在这样的等价物包括当前已知的等价物以及将来发展的等价物，即，无论结构如何、执行相同功能的所开发的任何元件。

因此，例如，本领域技术人员将理解：这里呈现的框图表示体现本原理的示例电路的概念性视图。类似地，将理解：任何流程图示、流程图、状态转换图、伪代码等等表示各种处理，所述各种处理可以基本上在计算机可读介质中表示，并因此由计算机或处理器执行，无论这样的计算机或处理器是否被明确

示出。

附图中所示的各种元件的功能可以通过使用专用硬件以及能够与适合的软件相关联而执行软件的硬件来提供。当由处理器提供时，所述功能可以由单个专用处理器、单个共享处理器、或其中一些可以被共享的多个独立处理器提供。此外，术语“处理器”或“控制器”的明确使用不应被解释为唯一地代表能够执行软件的硬件，其也可以隐含地、不受限制地包括数字信号处理器（“DSP”）硬件、用于存储软件的只读存储器（“ROM”）、随机存取存储器（“RAM”）、以及非易失性存储装置。

也可以包括其它的传统的和/或定制的硬件。类似地，附图中所示的任何开关只是概念性的。它们的功能可以通过程序逻辑的操作、通过专用逻辑、通过程序控制和专用逻辑间的交互作用、或甚至手动地执行，具体技术可由实施者选择，如从上下文中被更具体地理解的。

在本权利要求书中，表示为执行指定功能的部件的任何元件意在包含执行该功能的任何方式，所述方式包括：例如，a）执行该功能的电路元件的组合，或者 b）任何形式的软件，因此包括与用于执行该软件的适合的电路结合以执行该功能的固件、微代码等等。由这样的权利要求书限定的本原理归于这样的事实：由各种所叙述的部件提供的功能以权利要求书要求的方式组合并集合。因此认为：能够提供那些功能的任何部件等价于这里所示的那些部件。

说明书中引用的本原理的“一个实施例”或者“实施例”指结合所述实施例描述的特定特征、结构、特性等等包含在本原理的至少一个实施例中。由此，在贯穿说明书的各处出现的短语“在一个实施例中”或者“在实施例中”的出现不一定都指同一实施例。此外，短语“在另一个实施例中”不排除使所描述的实施例的主题与另一实施例的整体或部分上的组合。

另外，应该理解，术语“和/或”和“至少一个”的使用，例如在“A 和/或 B”和“A 和 B 中的至少一个”的情况下，意在涵盖仅选择所列的第一选项（A）、或者仅选择所列的第二选项（B）、或者选择两个选项（A 与 B）。作为进一步的例子，在“A、B、和/或 C”和“A、B、C 中的至少一个”的情况下，此类短语意在涵盖仅选择所列的第一选项（A）、或者仅选择所列的第二选项（B）、或者仅选择所列的第三选项（C）、或者仅选择所列的第一与第二选项（A 与 B）、或者仅选择所列的第一与第三选项（A 与 C）、或者仅选择所列的第二与第三选项（B 与 C）、或者选择全部三个选项（A 与 B 与 C）。本领域与相关领域的普通技术人员容易理解，可以将此扩展到所列的诸多项目。

如在此使用的那样，“多视图视频序列”指从不同的视点捕捉相同场景的

一组两个或更多个视频序列。

进一步，如在此可互换使用的那样，“交叉视图”和“视图间”两者都指属于除了当前视图之外的某一视图的画面。

另外，如在此使用的那样，短语“无需完全重构”指当没有在编码环或解码环中执行运动补偿时的情况。

此外，应理解，虽然在此关于 MPEG 4 AVC 标准的多视图视频编码扩展来描述本原理，但本原理不唯一地限于此标准和对应的扩展，因此在保持本原理精神的同时，可以对与多视图视频编码有关的其它视频编码标准、建议及其扩展来利用本原理。

转至图 1，附图标记 100 总地指示示例性的多视图视频编码（MVC）编码器。编码器 100 包括组合器 105，其具有以信号通信方式而与变换器 110 的输入连接的输出。变换器 110 的输出以信号通信的方式而与量化器 115 的输入连接。量化器 115 的输出以信号通信的方式而与熵编码器 120 的输入和逆量化器 125 的输入连接。逆量化器 125 的输出以信号通信的方式而与逆变换器 130 的输入连接。逆变换器 130 的输出以信号通信的方式而与组合器 135 的第一同相输入连接。组合器 135 的输出以信号通信的方式而与帧内预测器 145 的输入和去块滤波器 150 的输入连接。去块滤波器 150 的输出以信号通信的方式而与参考画面存储 155（用于视图 i）的输入连接。参考画面存储 155 的输出以信号通信的方式而与运动补偿器 175 的第一输入和运动估计器 180 的第一输入连接。运动估计器 180 的输出以信号通信的方式而与运动补偿器 175 的第二输入连接。

参考画面存储 160（用于其它视图）的输出以信号通信的方式而与视差/照度估计器 170 的第一输入和视差/照度补偿器 165 的第一输入连接。视差/照度估计器 170 的输出以信号通信的方式而与视差/照度补偿器 165 的第二输入连接。

熵解码器 120 的输出可用作编码器 100 的输出。组合器 105 的正相输入可用作编码器 100 的输入，并且以信号通信的方式而与视差/照度估计器 170 的第二输入和运动估计器 180 的第二输入连接。开关 185 的输出以信号通信的方式而与组合器 135 的第二同相输入连接并且与组合器 105 的反相输入连接。开关 185 包括以信号通信的方式而与运动补偿器 175 的输出连接的第一输入、以信号通信的方式而与视差/照度补偿器 165 的输出连接的第二输入、以及以信号通信的方式而与帧内预测器 145 的输出连接的第三输入。

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。

转至图 2, 附图标记 200 总地指示示例性的多视图视频编码 (MVC) 解码器。解码器 200 包括熵解码器 205, 其具有以信号通信的方式而与逆量化器 210 的输入连接的输出。逆量化器的输出以信号通信的方式而与逆变换器 215 的输入连接。逆变换器 215 的输出以信号通信的方式而与组合器 220 的第一正相输入连接。组合器 220 的输出以信号通信的方式而与去块滤波器 225 的输入和帧内预测器 230 的输入连接。去块滤波器 225 的输出以信号通信的方式而与参考画面存储 240 (用于视图 i) 的输入连接。参考画面存储 240 的输出以信号通信的方式而与运动补偿器 235 的第一输入连接。

参考画面存储 245 (用于其它视图) 的输出以信号通信的方式而与视差/照度补偿器 250 的第一输入连接。

熵编码器 205 的输入可用作进入解码器 200 的输入, 用于接收残余比特流。此外, 模式模块 260 的输入也可用作进入解码器 200 的输入, 用于接收控制语法以控制开关 255 选择哪个输入。而且, 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。此外, 视差/照度补偿器 250 的第二输入可用作进入解码器 200 的输入, 用于接收视差向量和照度补偿语法。

开关 255 的输出以信号通信的方式而与组合器 220 的第二正相输入连接。开关 255 的第一输入以信号通信的方式而与视差/照度补偿器 250 的输出连接。开关 255 的第二输入以信号通信的方式而与运动补偿器 235 的输出连接。开关 255 的第三输入以信号通信的方式而与帧内预测器 230 的输出连接。模式模块 260 的输出以信号通信的方式而与开关 255 连接, 用于控制开关 255 选择哪个输入。去块滤波器 225 的输出可用作解码器的输出。

如上所述, 本原理针对在编码器和解码器处用于支持多视图编码视频的单环解码的方法和装置。

本原理特别地适合于当仅将要对多视图视频内容的某些视图进行解码时的情况。这些应用不包括完全地重构参考视图 (即像素数据)。在实施例中, 可以推断来自那些视图的某些元素, 并且可以将其用于其它视图, 因此节约了存储器和时间。

当前多视图视频编码规范要求完全重构所有视图。重构的视图然后可以用作视图间参考。转至图 3, 附图标记 300 总地指示用于具有 8 视图的示例性 MVC 系统的编码结构。

作为重构的视图可以用作视图间参考这一事实的结果, 即使可以不输出各个视图, 也必须对每一视图进行完全地解码并且将其存储在存储器中。由于人

们将需要花费处理器时间来对这些非输出视图进行解码,以及花费存储器来为这种非输出视图存储已解码画面,因此从存储器和处理器利用而言,这并不是非常有效率的。

因此,根据本原理,我们提出了用于支持用于多视图编码序列的单环解码的方法和装置。如上所述,虽然主要关于 MPEG-4 AVC 标准的多视图视频编码扩展来描述在此提供的示例,但是,给出了对在此提供的本原理的教学,容易理解,在保持本原理的精神的同时,本领域或相关领域普通技术人员可以容易地将本原理应用到任何多视图视频编码系统。

在单环解码的一个实施例中,仅锚点画面将使用完全重构的画面作为参考,而非锚点画面不使用完全重构的画面作为参考。为了改进用于非锚点画面的编码效率,我们提出了使用视图间预测,从而视图间预测从相邻视图推断某些数据而不需要完全重构相邻视图。表 1 中示出的序列参数集语法指示相邻参考视图。表 1 示出了根据本原理实施例的用于 MPEG-4 AVC 标准的多视图视频编码扩展的序列参数集 (SPS) 语法。

seq_parameter_set_mvc_extension() {	C	描述符
num_views_minus_1		ue(v)
for(i = 0; i <= num_views_minus_1; i++)		
view_id[i]		ue(v)
for(i = 0; i <= num_views_minus_1; i++) {		
num_anchor_refs_l0[i]		ue(v)
for(j = 0; j < num_anchor_refs_l0[i]; j++)		
anchor_ref_l0[i][j]		ue(v)
num_anchor_refs_l1[i]		ue(v)
for(j = 0; j < num_anchor_refs_l1[i]; j++)		
anchor_ref_l1[i][j]		ue(v)
}		
for(i = 0; i <= num_views_minus_1; i++) {		
num_non_anchor_refs_l0[i]		ue(v)
for(j = 0; j < num_non_anchor_refs_l0[i]; j++)		
non_anchor_ref_l0[i][j]		ue(v)
num_non_anchor_refs_l1[i]		ue(v)
for(j = 0; j < num_non_anchor_refs_l1[i]; j++)		
non_anchor_ref_l1[i][j]		ue(v)
}		
}		

表 1

可以从相邻参考视图推断而无需完全重构的信息可以是下述一个或多个的组合: (1) 运动和模式信息; (2) 残余预测; (3) 帧内预测模式; (4) 照度补偿偏移; (5) 深度信息; 以及 (6) 去块强度。应理解, 前面类型的信息仅

仅是示例性的，并且本原理不唯一地限于关于可以从相邻视图推断而无需完全重构的信息的前面类型的信息。例如，应理解，在保持本原理的精神的同时，根据本原理可以使用与来自相邻视图的至少一部分画面的特性有关的任何类型的信息，包括与对这些画面或画面部分进行编码和/或解码有关的任何类型的信息。此外，在保持本原理的精神的同时，可以从语法和/或其它源推断这种信息。

关于运动和模式信息，这与当前多视图视频编码规范中的运动跳跃模式类似，其中从相邻视图推断运动向量、模式和参考索引信息。此外，可以通过发送额外的数据来提炼(refine)所推断的运动信息。此外，还可以推断视差信息。

关于残余预测，这里将来自相邻视图的残余数据作用于当前宏块的残留的预测数据。可以通过发送用于当前宏块的额外数据来进一步提炼该残余数据。

关于帧内预测模式，也可以推断这些模式。可以将重构的帧内宏块直接用作预测数据，或可以将帧内预测模式直接用于当前宏块。

关于照度补偿偏移，可以推断以及此外进一步提炼照度补偿偏移值。

关于深度信息，也可以推断深度信息。

为了确定多视图视频编码序列是否支持单环解码，可以以下述中的一个或多个来表示高级语法：序列参数集 (SPS)；画面参数集 (PPS)；网络抽象层 (NAL) 单元报头；片报头(slice header)；以及补充增强信息 (SEI) 消息。也可以将单环多视图视频解码指定为简档(profile)。

表 2 示出了所提出的根据实施例的用于 MPEG-4 AVC 标准的多视图视频编码扩展的序列参数集 (SPS) 语法，其包括 non_anchor_single_loop_decoding_flag 语法元素。non_anchor_single_loop_decoding_flag 是在以信号通知非锚点画面参考的环中添加的额外语法元素。添加 non_anchor_single_loop_decoding_flag 语法元素，以便以信号通知是否应该对用于视图 “i” 的非锚点画面的参考进行完全解码以对视图 “i” 进行解码。non_anchor_single_loop_decoding_flag 语法元素具有下述语义：

non_anchor_single_loop_decoding_flag[i] 等于 1 指示不需要对用于视图 id 等于 view_id[i] 的视图的非锚点画面的参考视图进行完全重构以对该视图进行解码。non_anchor_single_loop_decoding_flag[i] 等于 0 指示应该对用于视图 id

等于 view-id[i]的视图的非锚点画面的参考视图进行完全重构以对该视图进行解码。

表 2

seq_parameter_set_mvc_extension() {	C	描述符
num_views_minus_1		ue(v)
for(i = 0; i <= num_views_minus_1; i++)		
view_id[i]		ue(v)
for(i = 0; i <= num_views_minus_1; i++) {		
num_anchor_refs_l0[i]		ue(v)
for(j = 0; j < num_anchor_refs_l0[i]; j++)		
anchor_ref_l0[i][j]		ue(v)
num_anchor_refs_l1[i]		ue(v)
for(j = 0; j < num_anchor_refs_l1[i]; j++)		
anchor_ref_l1[i][j]		ue(v)
}		
for(i = 0; i <= num_views_minus_1; i++) {		
num_non_anchor_refs_l0[i]		ue(v)
non_anchor_single_loop_decoding_flag[i]		u(1)
for(j = 0; j < num_non_anchor_refs_l0[i]; j++)		
non_anchor_ref_l0[i][j]		ue(v)
num_non_anchor_refs_l1[i]		ue(v)
for(j = 0; j < num_non_anchor_refs_l1[i]; j++)		
non_anchor_ref_l1[i][j]		ue(v)
}		
}		

表 3 示出了所提出的根据另一实施例的用于 MPEG-4 AVC 标准的多视图视频编码扩展的序列参数集 (SPS) 语法，其包括 non_anchor_single_loop_decoding_flag 语法元素。non_anchor_single_loop_decoding_flag 语法元素用来指示对于整个序列，可以对所有的非锚点画面进行解码而无需完全重构参考视图。non_anchor_single_loop_decoding_flag 语法元素具有下述语义：

non_anchor_single_loop_decoding_flag 等于 1 指示可以对所有视图的所有非锚点画面进行解码而无需完全重构对应的参考视图的画面。

表 3

seq_parameter_set_mvc_extension() {	C	描述符
num_views_minus_1		ue(v)
non_anchor_single_loop_decoding_flag		u(1)
for(i = 0; i <= num_views_minus_1; i++)		
view_id[i]		ue(v)
for(i = 0; i <= num_views_minus_1; i++) {		
num_anchor_refs_l0[i]		ue(v)
for(j = 0; j < num_anchor_refs_l0[i]; j++)		
anchor_ref_l0[i][j]		ue(v)
num_anchor_refs_l1[i]		ue(v)
for(j = 0; j < num_anchor_refs_l1[i]; j++)		
anchor_ref_l1[i][j]		ue(v)
}		
for(i = 0; i <= num_views_minus_1; i++) {		
num_non_anchor_refs_l0[i]		ue(v)
for(j = 0; j < num_non_anchor_refs_l0[i]; j++)		
non_anchor_ref_l0[i][j]		ue(v)
num_non_anchor_refs_l1[i]		ue(v)
for(j = 0; j < num_non_anchor_refs_l1[i]; j++)		
non_anchor_ref_l1[i][j]		ue(v)
}		
}		

在单环解码的另一实施例中，甚至关于单环解码使能锚点画面。表 4 示出了所提出的根据另一实施例的用于 MPEG-4 AVC 标准的多视图视频编码扩展的序列参数集 (SPS) 语法，其包括 anchor_single_loop_decoding_flag 语法元素。anchor_single_loop_decoding_flag 语法元素可以存在，用于序列参数集中的锚点画面依赖性环。anchor_single_loop_decoding_flag 语法元素具有下述语义：

anchor_single_loop_decoding_flag[i] 等于 1 指示不需要对用于视图 id 等于 view_id[i] 的视图的锚点画面的参考视图进行完全重构以对该视图进行解码。anchor_single_loop_decoding_flag[i] 等于 0 指示应该对用于视图 id 等于 view_id[i] 的视图的锚点画面的参考视图进行完全重构以对该视图进行解码。

表 4

seq_parameter_set_mvc_extension() {	C	描述符
num_views_minus_1		ue(v)
for(i = 0; i <= num_views_minus_1; i++)		
view_id[i]		ue(v)
for(i = 0; i <= num_views_minus_1; i++) {		
num_anchor_refs_l0[i]		ue(v)
anchor_single_loop_decoding_flag[i]		u(1)
for(j = 0; j < num_anchor_refs_l0[i]; j++)		
anchor_ref_l0[i][j]		ue(v)
num_anchor_refs_l1[i]		ue(v)
for(j = 0; j < num_anchor_refs_l1[i]; j++)		
anchor_ref_l1[i][j]		ue(v)
}		
for(i = 0; i <= num_views_minus_1; i++) {		
num_non_anchor_refs_l0[i]		ue(v)
non_anchor_single_loop_decoding_flag[i]		u(1)
for(j = 0; j < num_non_anchor_refs_l0[i]; j++)		
non_anchor_ref_l0[i][j]		ue(v)
num_non_anchor_refs_l1[i]		ue(v)
for(j = 0; j < num_non_anchor_refs_l1[i]; j++)		
non_anchor_ref_l1[i][j]		ue(v)
}		
}		

表 5 示出了所提出的根据另一实施例的用于 MPEG-4 AVC 标准的多视图视频编码扩展的序列参数集 (SPS) 语法, 其包括 anchor_single_loop_decoding_flag 语法元素。anchor_single_loop_decoding_flag 语法元素具有下述语义:

anchor_single_loop_decoding_flag 等于 1 指示可以对所有视图的所有锚点画面进行解码而无需完全重构对应的参考视图的画面。

表 5

seq_parameter_set_mvc_extension() {	C	描述符
num_views_minus_1		ue(v)
anchor_single_loop_decoding_flag		u(1)
non_anchor_single_loop_decoding_flag		u(1)
for(i = 0; i <= num_views_minus_1; i++)		
view_id[i]		ue(v)
for(i = 0; i <= num_views_minus_1; i++) {		
num_anchor_refs_l0[i]		ue(v)
for(j = 0; j < num_anchor_refs_l0[i]; j++)		
anchor_ref_l0[i][j]		ue(v)
num_anchor_refs_l1[i]		ue(v)
for(j = 0; j < num_anchor_refs_l1[i]; j++)		
anchor_ref_l1[i][j]		ue(v)
}		
for(i = 0; i <= num_views_minus_1; i++) {		
num_non_anchor_refs_l0[i]		ue(v)
for(j = 0; j < num_non_anchor_refs_l0[i]; j++)		
non_anchor_ref_l0[i][j]		ue(v)
num_non_anchor_refs_l1[i]		ue(v)
for(j = 0; j < num_non_anchor_refs_l1[i]; j++)		
non_anchor_ref_l1[i][j]		ue(v)
}		
}		

转至图 4, 附图标记 400 总地指示用于对多视图视频内容进行编码以支持单环解码的示例性方法。

方法 400 包括将控制传递到功能块 410 的开始块 405。功能块 405 解析编码器配置文件, 并且将控制传递到判决块 415。判决块 415 确定变量 i 是否比要编码的视图的数量少。如果是, 则将控制传递到判决块 420。否则, 将控制传递到结束块 499。

判决块 420 确定是否使得单环编码能够用于视图 i 的锚点画面。如果是, 则将控制传递到功能块 425。否则, 将控制传递到功能块 460。

功能块 425 设置 `anchor_single_loop_decoding_flag[i]` 等于 1, 并且将控制传递到判决块 430。判决块 430 确定是否使得单环编码能够用于视图 i 的非锚点画面。如果是, 则将控制传递到功能块 435。否则, 将控制传递到功能块 465。

功能块 435 将 `non_anchor_single_loop_decoding_flag[i]` 设置为 1, 并且将控制传递到功能块 440。

功能块 440 将 `anchor_single_loop_decoding_flag[i]` 和 `non_anchor_single_loop_decoding_flag[i]` 写入用于视图 i 的序列参数集 (SPS)、画面参数集 (PPS)、网络抽象层 (NAL) 单元报头和/或片报头, 并且将控制传递到功能块 445。功能块 445 从 SPS 考虑视图间依赖性, 同时当不包括视图间预测时对视图的宏块进行编码, 并且将控制传递到功能块 450。功能块 450 推断用于单环编码的运动信息、帧间预测模式、残余数据、视差数据、帧内预测模式和深度信息的组合, 并且将控制传递到功能块 455。功能块 455 将变量 i 增加 1, 并且将控制返回到判决块 415。

功能块 460 将 `anchor_single_loop_decoding_flag[i]` 设置为等于 0, 并且将控制传递到判决块 430。

功能块 465 将 `non_anchor_single_loop_decoding_flag[i]` 设置为等于 0, 并且将控制传递到功能块 440。

转至图 5, 附图标记 500 总地指示用于多视图视频内容的单环解码的示例性方法。

方法 500 包括将控制传递到功能块 510 的起始块 505。功能块 510 从用于视图 i 的序列参数集 (SPS)、画面参数集 (PPS)、网络抽象层 (NAL) 单元报头或片报头中读取 `anchor_single_loop_decoding_flag[i]` 和 `non_anchor_single_loop_decoding_flag[i]`, 并且将控制传递到判决块 515。判决块 515 确定变量 i 是否小于要解码的视图的数量。如果是, 则将控制传递到判

决块 520。否则，将控制传递到结束块 599。

判决块 520 确定当前画面是否为锚点画面。如果是，则将控制传递到判决块 525。否则，将控制传递到判决块 575。

判决块 525 确定 `anchor_single_loop_decoding_flag[i]` 是否为 1。如果是，则将控制传递到功能块 530。否则，将控制传递到功能块 540。

在当不包括视图间预测时对视图 *i* 的宏块进行解码的时候，功能块 530 从序列参数集（SPS）中考虑视图间依赖性，并且将控制传递到功能块 535。功能块 535 推断用于运动跳跃宏块的运动信息、帧间预测模式、残余数据、视差数据、帧内预测模式、深度信息的组合，并且将控制传递到功能块 570。

功能块 570 将变量 *i* 增加 1，并且将控制返回到判决块 515。

在当包括视图间预测时对视图 *i* 的宏块进行解码的同时，功能块 540 从序列参数集（SPS）中考虑视图间依赖性，并且将控制传递到功能块 545。功能块 545 推断运动信息、帧间预测模式、残余数据、视差数据、帧内预测模式和深度信息的组合，并且将控制传递到功能块 570。

判决块 575 确定 `non_anchor_single_loop_decoding_flag[i]` 是否为 1。如果是，则将控制传递到功能块 550。否则，将控制传递到功能块 560。

在当不包括视图间预测时对视图 *i* 的宏块进行解码的同时，功能块 550 从序列参数集（SPS）中考虑视图间依赖性，并且将控制传递到功能块 555。功能块 555 推断用于运动跳跃宏块的运动信息、帧间预测模式、残余数据、视差数据、视图内模式和深度信息的组合，并且将控制传递到功能块 570。

在当包括视图间预测时对视图 *i* 的宏块进行解码的同时，功能块 560 从序列参数集（SPS）中考虑视图间依赖性，并且将控制传递到功能块 565。功能块 565 推断运动信息、帧间预测模式、残余数据、视差数据、帧内预测模式和深度信息的组合，并且将控制传递到功能块 570。

转至图 6，附图标记 600 总地指示用于对多视图视频内容进行编码以支持单环解码的另一示例性方法。

方法 600 包括用于将控制传递到功能块 610 的起始块 605。功能块 610 解析编码器配置文件，并且将控制传递到判决块 615。判决块 615 确定是否使得单环编码能够用于每一视图的所有锚点画面。如果是，则将控制传递到功能块 620。否则，将控制传递到功能块 665。

功能块 620 将 `anchor_single_loop_decoding_flag` 设置为等于 1，并且将控制传递到判决块 625。判决块 625 确定是否使得单环编码能够用于每一视图的

所有非锚点画面。如果是，则将控制传递到功能块 630。否则，将控制传递到功能块 660。

功能块 630 将 `non_anchor_single_loop_decoding_flag` 设置为等于 1，并且将控制传递到功能块 635。功能块 635 将 `anchor_single_loop_decoding_flag` 写入序列参数集 (SPS)、画面参数集 (PPS)、网络抽象层 (NAL) 单元报头和/或片报头，并且将控制传递到判决块 640。判决块 640 确定变量 *i* 是否小于要编码的视图的数量。如果是，则将控制传递到功能块 645。否则，将控制传递到结束块 699。

在当不包括视图间预测时对视图的宏块进行编码的同时，功能块 645 从 SPS 中考虑视图间依赖性，并且将控制传递到功能块 650。功能块 650 推断用于单环编码的运动信息、帧间预测模式、残余数据、视差数据、帧内预测模式、深度信息的组合，并且将控制传递到功能块 655。功能块 655 将变量 *i* 增加 1，并且将控制返回到判决块 640。

功能块 665 将 `anchor_single_loop_decoding_flag` 设置为等于 0，并且将控制传递到判决块 625。

功能块 660 将 `non_anchor_single_loop_decoding_flag` 设置为等于 0，并且将控制传递到功能块 635。

转至图 7，附图标记 700 总地指示用于多视图视频内容的单环解码的另一示例性方法。

方法 700 包括将控制传递到功能块 710 的起始块 705。功能块 710 从用于视图 *i* 的序列参数集 (SPS)、画面参数集 (PPS)、网络抽象层 (NAL) 单元报头或片报头中读取 `anchor_single_loop_decoding_flag` 和 `non_anchor_single_loop_decoding_flag`，并且将控制传递到判决块 715。判决块 715 确定变量 *i* 是否小于要解码的视图的数量。如果是，则将控制传递到判决块 720。否则，将控制传递到结束块 799。

判决块 720 确定当前画面是否是锚点画面。如果是，则将控制传递到判决块 725。否则，将控制传递到判决块 775。

判决块 725 确定 `anchor_single_loop_decoding_flag` 是否等于 1。如果是，则将控制传递到功能块 730。否则，将控制传递到功能块 740。

在当不包括视图间预测时对视图 *i* 的宏块进行解码的时候，功能块 730 从序列参数集 (SPS) 中考虑视图间依赖性，并且将控制传递到功能块 735。功能块 735 推断用于运动跳跃宏块的运动信息、帧间预测模式、残余数据、视差

数据、帧内预测模式、深度信息的组合，并且将控制传递到功能块 770。

功能块 770 将变量 i 增加 1，并且将控制返回到判决块 715。

在当包括视图间预测时对视图 i 的宏块进行解码的同时，功能块 740 从序列参数集（SPS）中考虑视图间依赖性，并且将控制传递到功能块 745。功能块 745 推断运动信息、帧间预测模式、残余数据、视差数据、帧内预测模式和深度信息的组合，并且将控制传递到功能块 770。

判决块 775 确定 `non_anchor_single_loop_decoding` 是否等于 1。如果是，则将控制传递到功能块 750。否则，将控制传递到功能块 760。

在当不包括视图间预测时对视图 i 的宏块进行解码的同时，功能块 750 从序列参数集（SPS）中考虑视图间依赖性，并且将控制传递到功能块 755。功能块 755 推断用于运动跳跃宏块的运动信息、帧间预测模式、残余数据、视差数据、帧内预测模式和深度信息的组合，并且将控制传递到功能块 770。

在当包括视图间预测时对视图 i 的宏块进行解码的同时，功能块 760 从序列参数集（SPS）中考虑视图间依赖性，并且将控制传递到功能块 765。功能块 765 推断运动信息、帧间预测模式、残余数据、视差数据、帧内预测模式和深度信息的组合，并且将控制传递到功能块 770。

现在将给出对于本原理的许多伴随优点/特点中的一些伴随优点/特点的描述，其中一些已经在上面提到。例如，一个优点/特点是具有如下这样的编码器的装置，所述编码器用于当使用视图间预测对多视图视频内容进行编码时，对多视图视频内容进行编码以使得能够进行多视图视频内容的单环解码。

另一优点/特点是具有如上所描述的编码器的装置，其中所述多视图视频内容包括参考视图和其它视图。所述其它视图能够被重构而无需所述参考视图的完全重构。

又一优点/特点是具有如上所描述的编码器的装置，其中视图间预测包括：从多视图视频内容的参考视图推断运动信息、帧间预测模式、帧内预测模式、参考索引、残余数据、深度信息、照度补偿偏移、去块强度和视差数据中的至少一个。

再一优点/特点是具有如上所描述的编码器的装置，其中视图间预测包括：从如下特性推断用于多视图内容的给定视图的信息，所述特性与来自关于所述给定视图的所述多视图视频内容的参考视图中的至少一个画面的至少一部分的至少一个有关；并且对与所述至少一个画面的至少一部分有关的信息进行解码。

此外，另一优点/特点是具有如上所描述的编码器的装置，其中使用高级语法元素来指示使得单环解码能够用于多视图视频内容。

此外，另一优点/特点是具有使用所描述的高级语法的编码器的装置，其中所述高级语法元素分别进行以下之一：指示是否使得所述单环解码能够用于多视图视频内容中的锚点画面和非锚点画面，基于视图指示是否使得能够进行所述单环解码，基于序列指示是否使得能够进行所述单环解码，并且指示仅对于多视图视频内容中的非锚点画面使得能够进行所述单环解码。

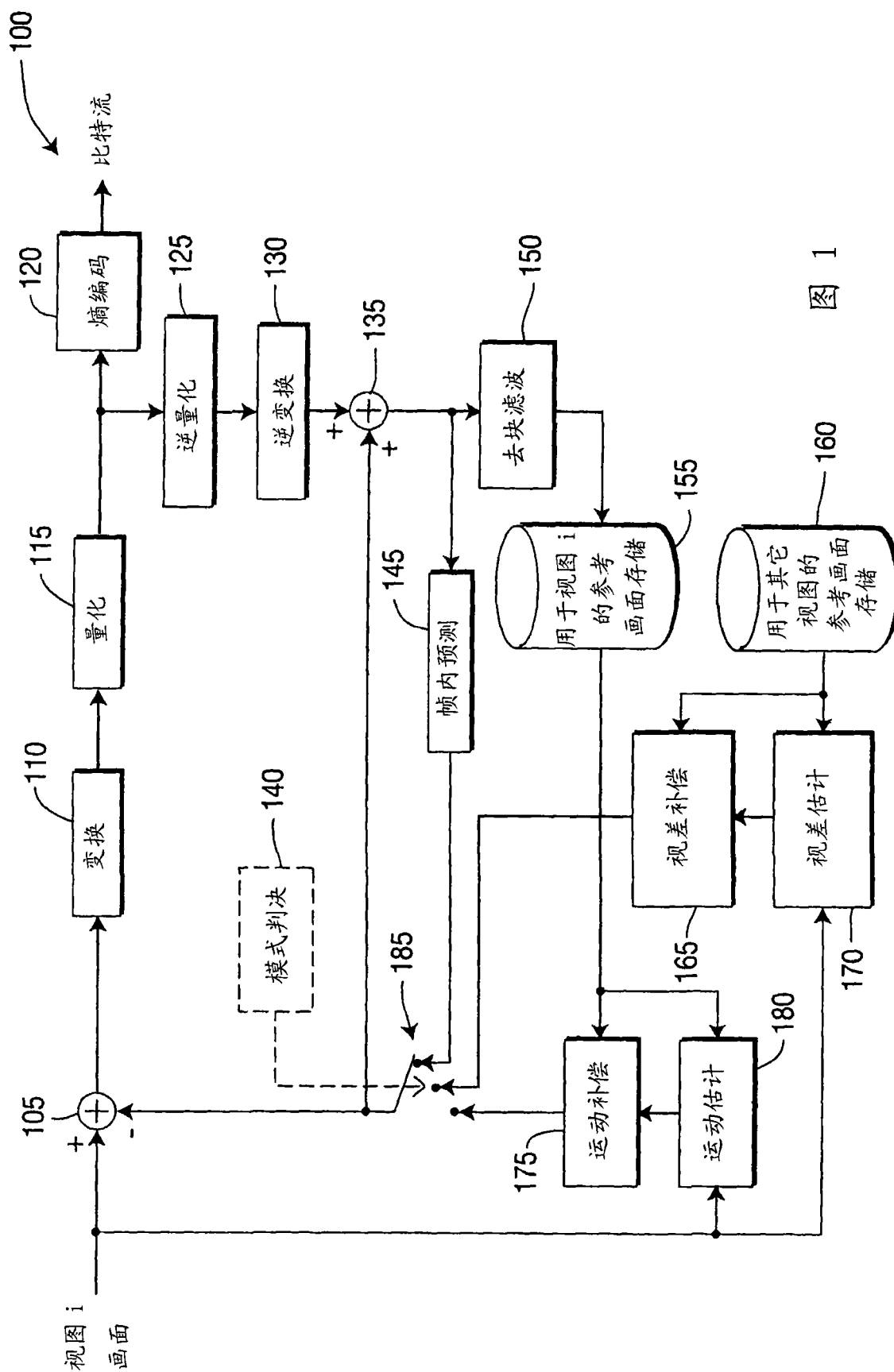
基于这里的教学，相关领域中的普通技术人员可以容易地确定本原理的这些和其他特点和优势。应当理解，可以在各种形式的硬件、软件、固件、专用处理器或它们的组合中实施本原理的教学。

最优选地，本原理的教导被实现为硬件和软件的组合。另外，软件可以实现为以有形方式实现在程序存储单元上的应用程序。该应用程序可以被上载到包含任何适当架构的机器并且由其执行。优选地，所述机器被实现在计算机平台之上，其具有诸如一个或者多个中央处理单元（“CPU”）、随机存取存储器（“RAM”）、以及输入/输出（“I/O”）接口之类的硬件。计算机平台还可以包括操作系统和微指令代码。此处描述的各种处理和功能可以为微指令代码的一部分或者为应用程序的一部分，或者其任何组合，其可以由 CPU 执行。另外，例如附加的数据存储单元以及打印单元的各种其他外设单元可以连接到计算机平台。

此外，要理解的是，对于在其上编码有视频信号数据的存储介质的引用，无论在说明书还是权利要求中对其进行引用，都包括在其上记录了这种数据的任何类型的计算机可读存储介质。

还应该理解，因为在附图中所示的某些系统构成组件与方法优选地以软件实现，所以系统组件或者处理功能块之间的实际连接可能依赖于本原理的编排方式而不同。给出此处的教学，相关领域的普通技术人员将能够设想本原理的这些以及类似的实现或者配置。

虽然此处参照附图描述了说明性实施例，但是应该理解本原理不限于这些确切的实施例，并且相关领域的普通技术人员在不脱离本原理的范围与精神的情况下，可以在其中进行各种改变和修改。所有这些改变和修改都意在包含在权利要求书所提出的本原理的范围之内。



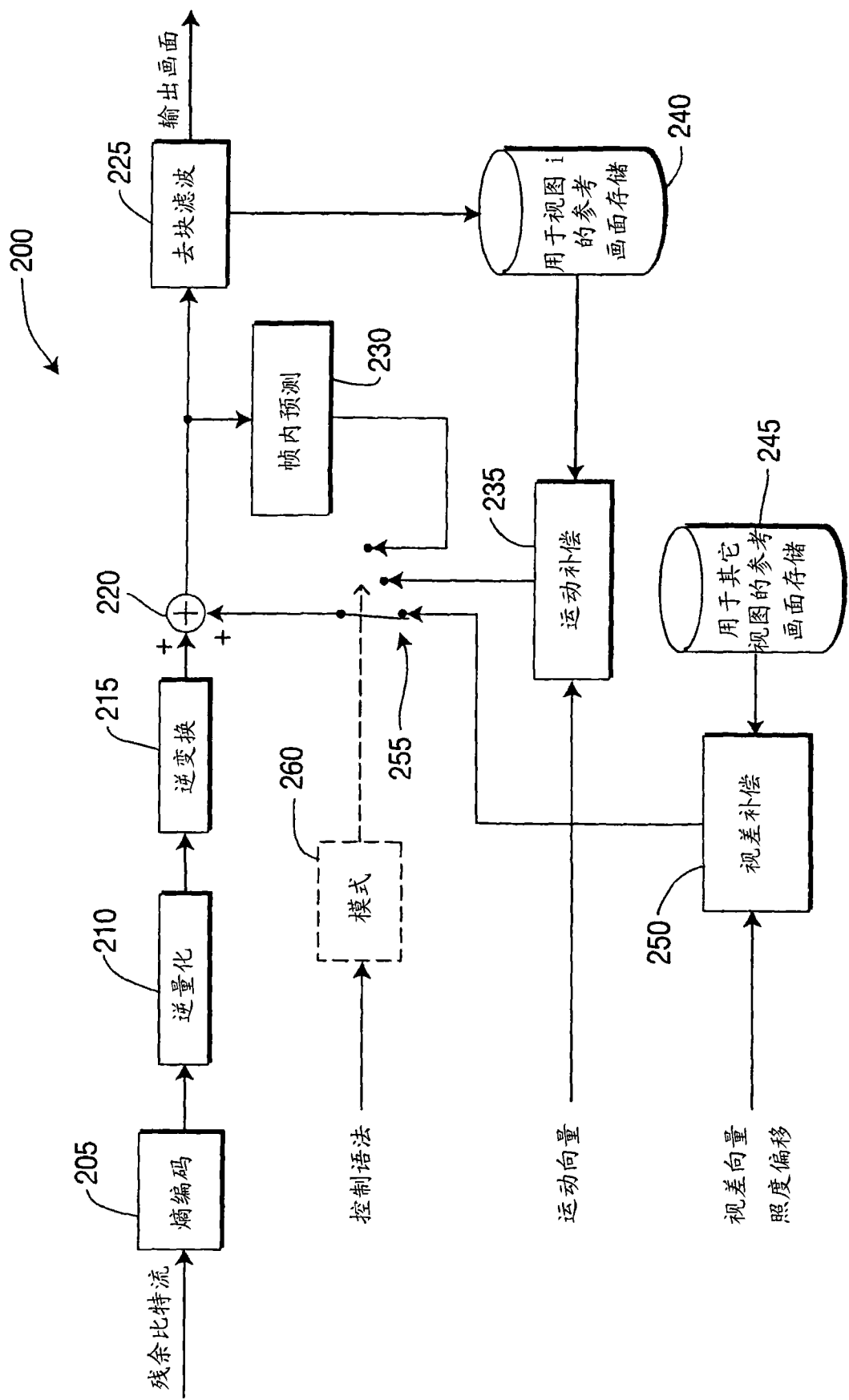


图 2

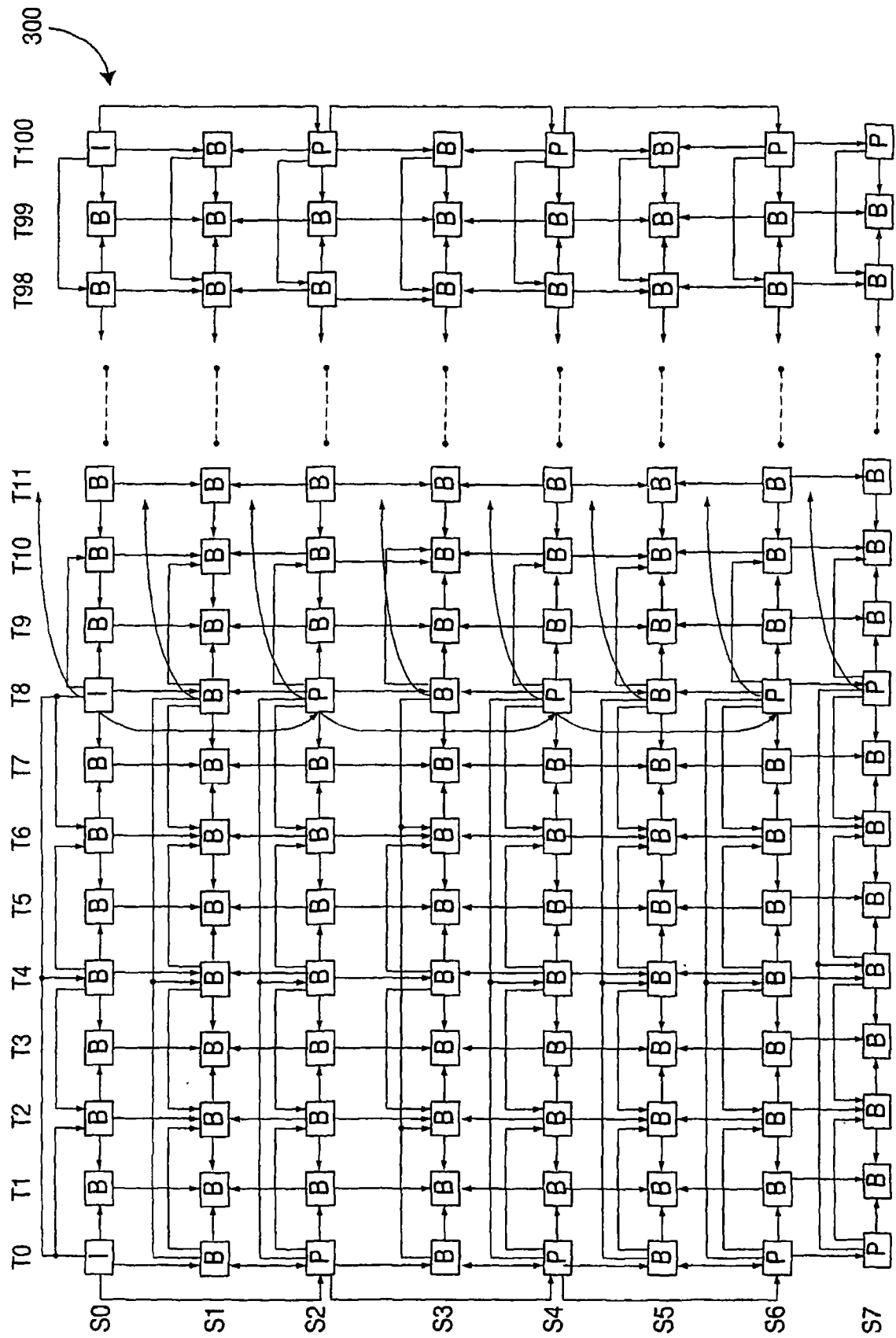


图 3

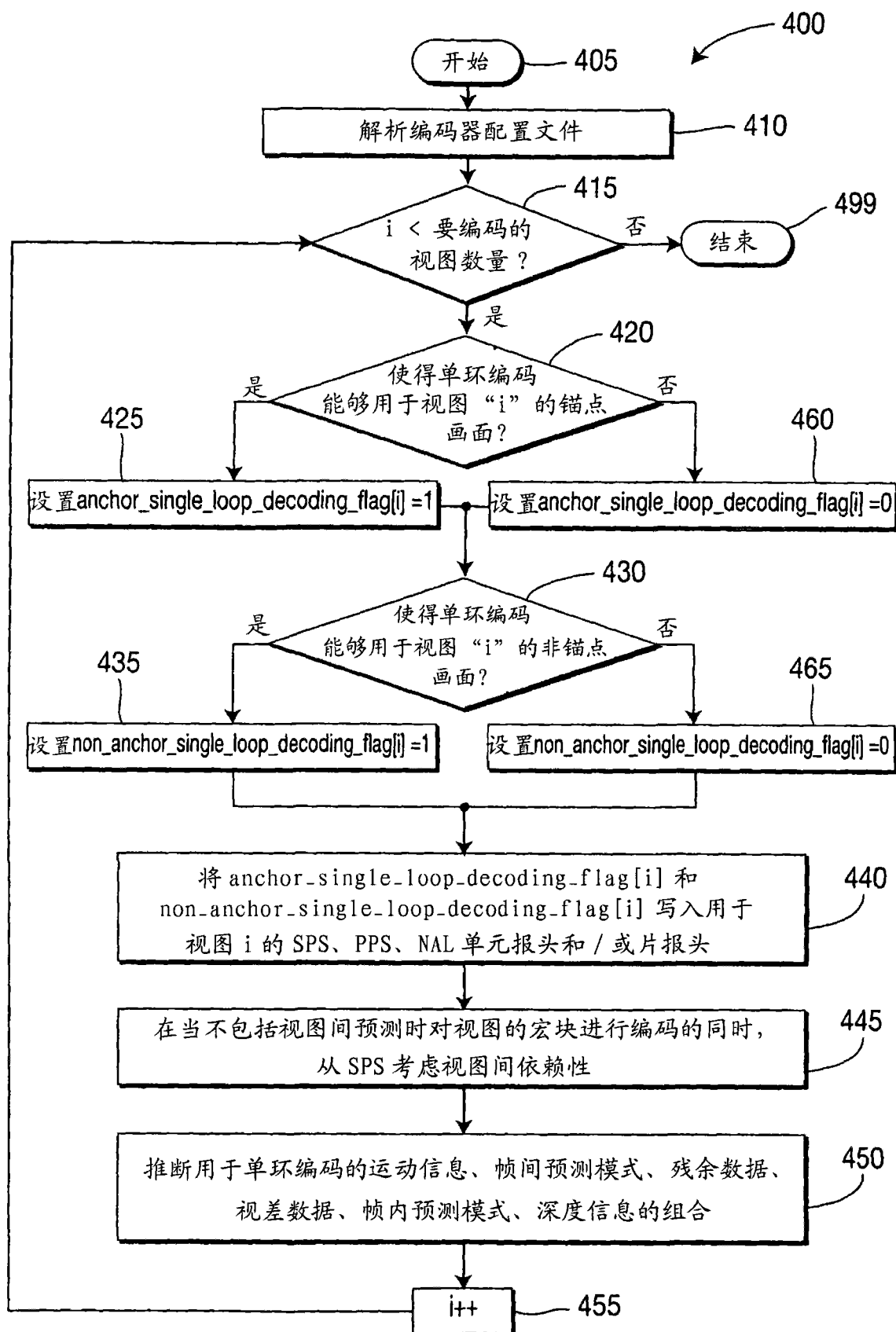


图 4

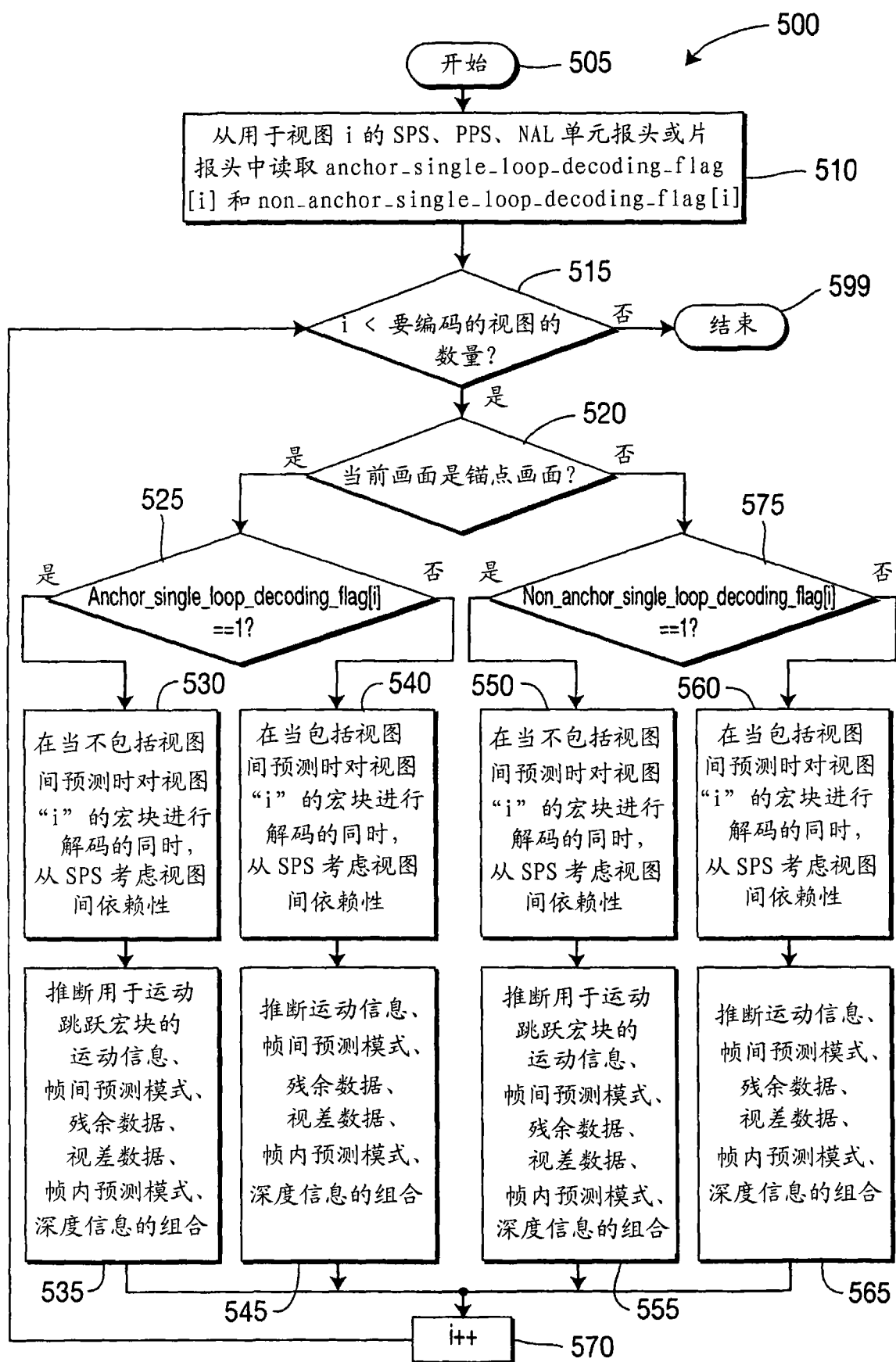


图 5

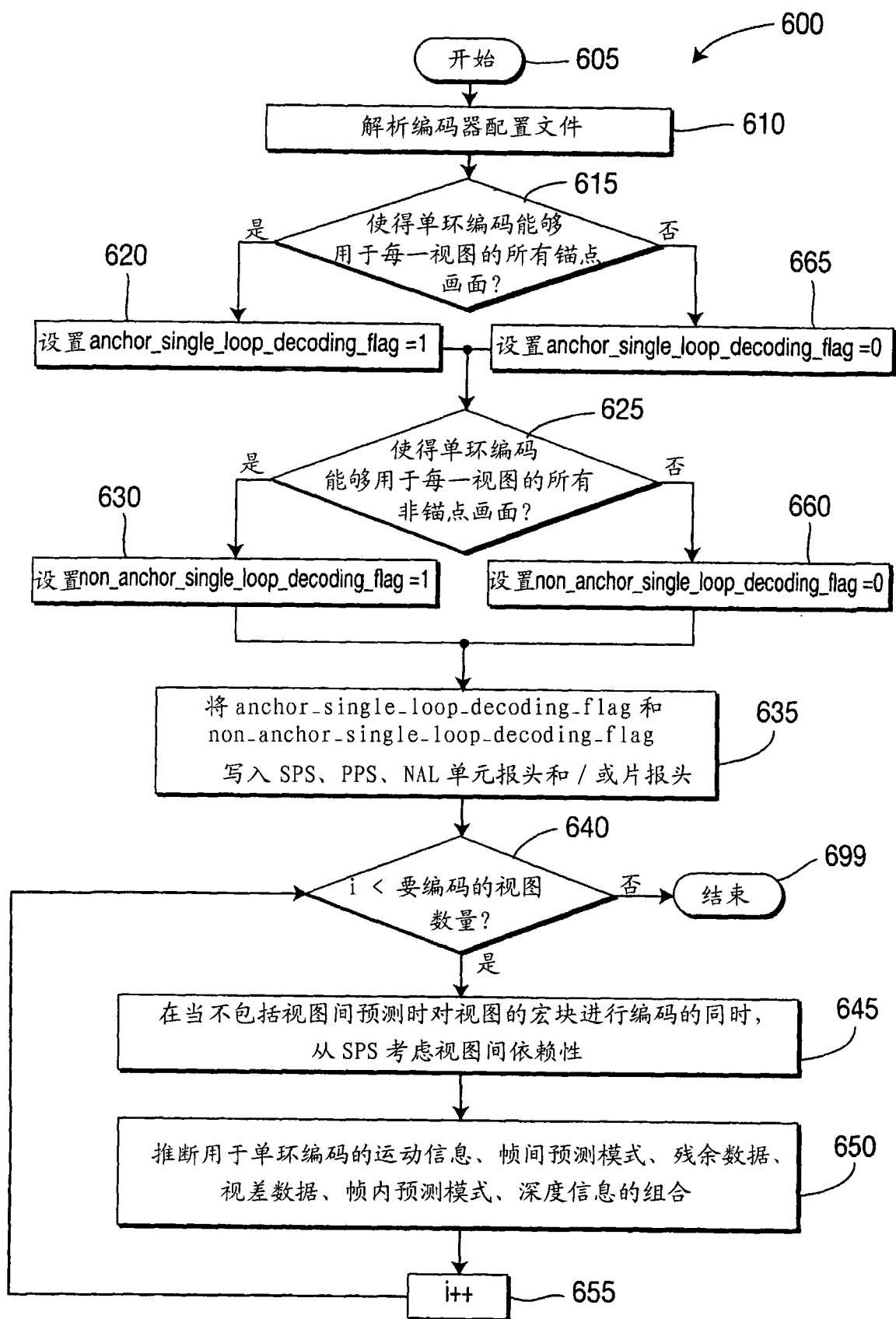


图 6

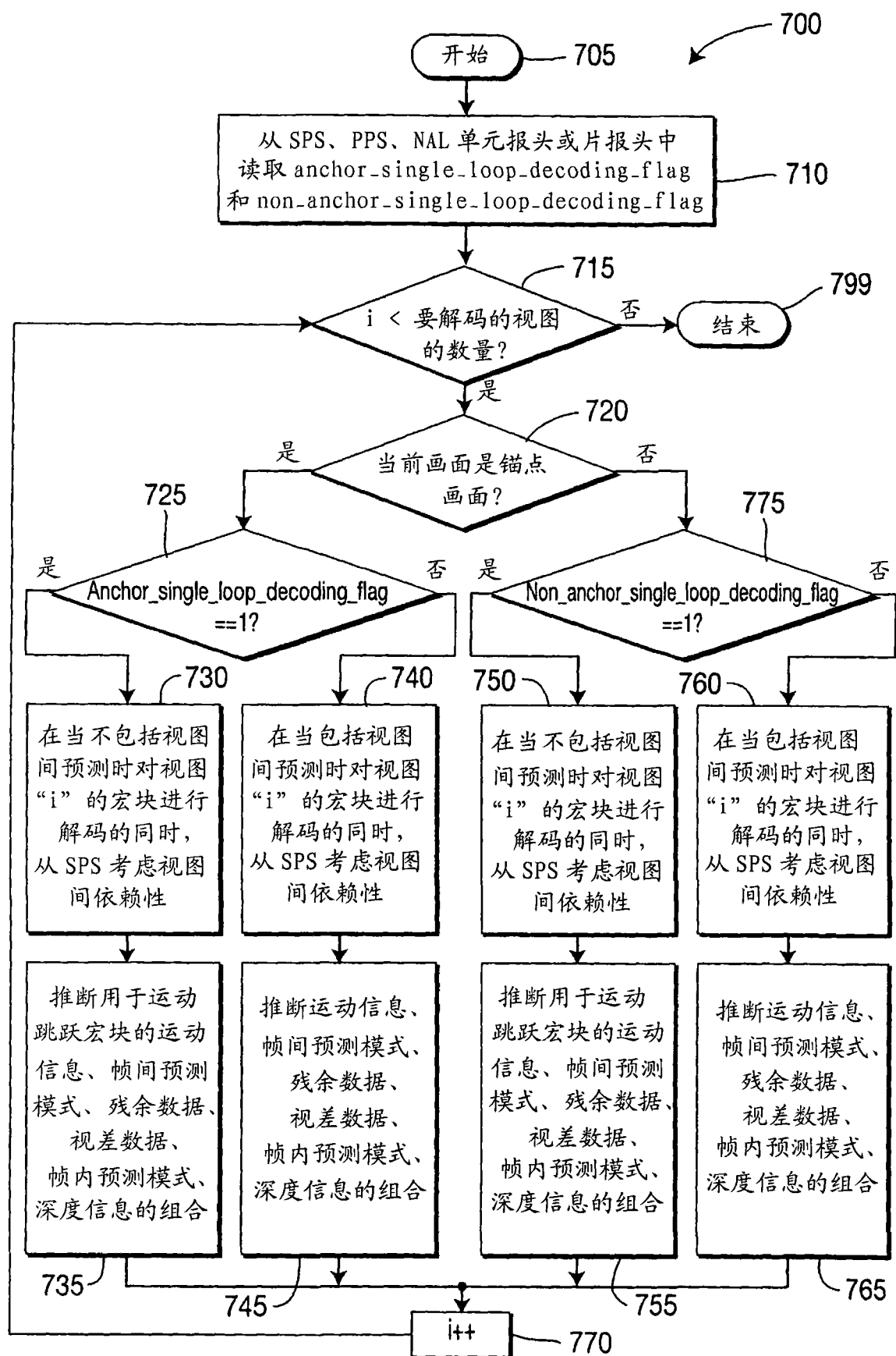


图 7