

[19] 中华人民共和国国家知识产权局

[51] Int. Cl.
H04N 7/50 (2006.01)



[12] 发明专利申请公开说明书

[21] 申请号 200510109820.5

[43] 公开日 2006年2月22日

[11] 公开号 CN 1738434A

[22] 申请日 2005.7.12

[21] 申请号 200510109820.5

[30] 优先权

[32] 2004. 7. 12 [33] US [31] 60/587,797

[32] 2005. 5. 10 [33] US [31] 11/126,650

[71] 申请人 微软公司

地址 美国华盛顿州

[72] 发明人 吴 枫 J·徐 季向阳

[74] 专利代理机构 上海专利商标事务所有限公司

代理人 李 玲

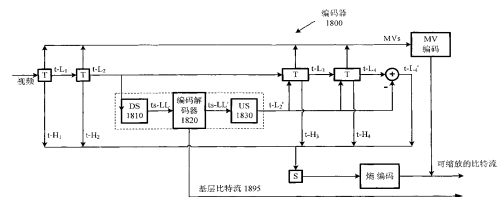
权利要求书 3 页 说明书 47 页 附图 21 页

[54] 发明名称

用于 3D 子带编码的嵌入式基层编解码器

[57] 摘要

描述了用于可缩放视频编码和解码的技术和工具。例如：包括嵌入式基层编解码器以及时间子带转换和空间子带转换的 3D 子带视频编解码器。基层编解码器在子带转换之间的位置和可在可缩放视频编码中基层编解码器的作用随实施而变化。通常，基层编解码器提供在低比特率的有效压缩并生成与现有解码器兼容的基层压缩视频比特流。同时，3D 子带视频编码器在较高比特率提供空间和时间缩放性选择，来改进基层视频。相应的 3D 子带视频编解码器包括嵌入式基层编解码器。



- 1、一种方法包括：
用其中嵌有基层编解码器的3D子带视频编码器对视频进行编码，和
5 输出编码结果。
- 2、如权利要求1的方法，其中基层编解码器被嵌在3D子带视频编码器的
复数个时间子带转换中间，以便使基层编解码器的编码/解码跟踪一个或多个复
数时间子带转换，并先于一个或多个复数时间子带转换进行。
- 3、如权利要求2的方法，其中复数个时间子带转换包括动作补偿时间滤波。
- 10 4、如权利要求1的方法，其中基层编解码器是具有二维频率转换的以块为
基础的动作补偿预测编解码器。
- 5、如权利要求1的方法，其中编码包括：
对输入到基层编解码器中的视频进行降抽样；
对输入到基层编解码器中的降抽样视频进行编码和解码，来生成基层比特
15 流和重建的输出视频；以及
对重建的输出视频进行升抽样。
- 6、如权利要求1的方法，其中编码包括：
执行空间子带转换，来生成空间低通子带以及一个或多个空间高通子带。
用基层编解码器对空间低通子带进行编码和解码，来生成基层比特流和重
20 建空间低通子带；
计算残余子带并作为空间低通子带和重建空间低通子带之间的差；以及
在残余子带和一个或多个空间高通子带上执行反向空间子带转换。
- 7、如权利要求1的方法，其中编码过程包括在残余子带上执行一个或多个
时间子带转换，其中残余子带至少部分基于输入到基层编解码器的视频和从基
25 层编解码器输出的重建视频之间的差。
- 8、如权利要求1的方法，其中编码包括在基层编解码器编码/解码之后在
一个或多个时间子带转换中执行动作补偿时间滤波，并且其中至少一些动作补
偿时间滤波是和从基层编解码器输出的重建视频中的一个或多个参考图像有
关。
- 30 9、如权利要求8的方法，其中动作补偿时间滤波包括预测阶段和更新阶段，

并且其中对于一个或多个大块中的每一个的动作补偿时间滤波包括:

当来自较低级时间低通子带的参考图像被用在预测阶段时, 对该参考图像执行更新阶段; 以及

5 当重建输出视频中的一个或多个参考图像被用在预测阶段时, 对该参考图像跳过更新阶段。

10、如权利要求 1 的方法, 其中编码包括在随后的使用 3D 子带视频编码器的编码中再次利用来自基层编解码器的编码/解码中的动作信息。

11、如权利要求 1 的方法, 其中编码包括在随后的使用 3D 子带视频编码器的编码中再次利用来自基层编解码器的编码/解码中的内部模式信息。

10 12、一种方法包括:

用其中嵌有基层解码器的 3D 子带视频解码器对视频进行解码; 输出解码结果。

13、如权利要求 12 的方法, 其中基层解码器被嵌在 3D 子带视频解码器的复数个反向时间子带转换中间, 以便使基层解码器的解码跟踪一个或多个复数
15 反向时间子带转换, 并先于一个或多个复数反向时间子带转换进行。

14、如权利要求 12 的方法, 其中基层解码器是具有二维反向频率转换的以块为基础的动作补偿预测解码器。

15、如权利要求 12 的方法, 其中解码包括:

20 用基层解码器对基层比特流进行解码, 来生成重建的输出视频; 以及对重建的输出视频进行升抽样。

16、如权利要求 12 的方法, 其中解码包括:

25 执行空间子带转换, 来生成空间低通子带以及一个或多个空间高通子带。用基层解码器对基层比特流进行解码, 来生成重建的空间低通子带; 合并残余子带和重建的空间低通子带, 来生成合并的低通子带; 以及在合并的低通子带和一个或多个空间高通子带上执行反向空间子带转换。

17、如权利要求 12 的方法, 其中解码包括在残余子带上执行一个或多个反向时间子带转换。

18、如权利要求 12 所述的方法, 其中解码包括在基层编解码器解码之后在一个或多个反向时间子带转换中执行反向动作补偿时间滤波, 并且其中至少一
30 些反向动作补偿时间滤波是和从基层解码器输出的重建视频中的一个或多个参

考图像有关。

19、如权利要求 18 所述的方法，其中反向动作补偿时间滤波包括预测阶段和更新阶段，并且其中对于一个或多个大块中的每一个的动作补偿时间滤波包括：

- 5 当来自较低级时间低通子带的参考图像被用在预测阶段时，对该参考图像执行更新阶段；以及

 当重建输出视频中的一个或多个参考图像被用在预测阶段时，对该参考图像跳过更新阶段。

20、一种系统包括：

- 10 用于利用 3D 子带视频编码器或 3D 子带视频解码器处理可缩放压缩视频比特流的装置；

 用于利用其中嵌有基层视频编解码器的 3D 子带视频编码器，或者其中嵌有基层解码器的 3D 子带视频解码器，来处理基层压缩视频比特流的装置。

用于 3D 子带编码的嵌入式基层编解码器

5 相关申请的交叉参考

包括本申请要求共同享有于 2004 年 7 月 12 日提出的申请号为 60/587,797 的美国临时申请，其公布这里引为参考。

技术领域

描述了用于可缩放的视频编码和解码的相关技术和工具。例如：3D 子带视
10 频编码器包括嵌入式基层编解码器，3D 子带视频解码器包括嵌入式基层解码器。

背景技术

数字视频消耗大量的存储容量和传输容量。典型的原始数字视频序列每秒
包括 15 或 30 帧。每帧可以包括几万或几十万像素（也称象素），其中每一个像
15 素代表图像的一个微小的元素。以原始形式，计算机通常用三个样值共 24 比特
的一组代表一个像素。例如：一个像素可以包括一个定义像素灰度成分的八比
特的亮度样值（也称发光率样值，在此处术语“亮度”与“发光率”是通用的）
和两个定义像素色彩成分的八比特的色度样值（也称为色品样值，在此处术语
“色度”与“色品”是通用的）。这样，典型的原始数字视频序列每秒的比特数
20 目或比特率，可达到每秒 5 百万比特或更多。

许多计算机和计算机网络缺乏资源来处理原始数字视频。由于这一原因，
工程师使用压缩（也称编码或译码）来减少数字视频的比特率。压缩通过将视
频转换成较低比特率的形式降低了存储和传输视频的成本。解压缩（也称解码）
从视频的压缩形式重建视频的最初版本。一个“编解码器”是一个编码/解码系
25 统。压缩可以是无损的，在这种情况下视频质量不会受到损失，但是由于受视
频数据可变性（有时称为熵）的固有数量的制约比特率缩减受到限制。或者，
压缩也可以是有损的，在这种情况下视频质量会受到损失，但获得的比特率的
缩减更为可观。有损压缩和无损压缩通常是同时发生的—有损压缩确定一个信
息近似值，而无损压缩用于表示这一近似值。

30 作为视频压缩的一般规则，质量变化直接与比特率有关。对于给定视频序

列，如果这一序列以较高质量进行编码，那么序列的比特率也就比较高，而如果这一序列以较低质量进行编码，那么这一序列的比特率也就比较低。各种因素都能影响比特率和原始数字视频序列的质量，包括时间分辨率（例如：每秒 7.5、15、30 或 60 视频帧）、空间分辨率（例如：每视频帧 176×144 (QCIF)，
5 352×288 (CIF)，或 704×576 (4CIF) 像素)，和样值分辨率（如：每像素 8，16 或 24 比特）。质量和比特率可能会由于活动时间、空间或者样值分辨率的高低而变化。

质量和比特率也取决于在有损压缩过程中由信息内容的简化或消除产生的失真量。这影响着，例如，重建时视频中的模糊度、块度、颗粒度等的数量。
10 明显不同地，有损压缩降低序列的质量以使编码器得到较低的比特率。

作为另外一个一般规则，质量和比特率取决于视频序列在细节和运动方面的复杂度。对于某些固定的质量水平，复杂序列典型地比简单序列需要更多比特进行编码。反过来的一面是，当以某些固定比特率编码时，复杂序列典型地得到比简单序列更低的质量。

15 在某些情况下，全部要求是以单一比特率/质量水平对视频编码。例如，如果视频是为了用单一类型的设备再现被编码，或者如果视频是为了经由电话线的点对点视频会议中再现被编码，那么以单一比特率/质量水平对视频进行简单编码是合理的。然而，在许多其他情况下，以多种比特率/质量水平对视频进行编码是合理的。例如：在互联网上流动视频时，视频服务器常常不得不向其
20 不同容量的设备提供视频和/或在各种具有不同速度和可靠特性的网络环境中分发视频。

一种寻址不同网络和重现需求的方法是以多种比特率和质量水平对同一视频序列进行编码，这会因为多种独立的压缩视频比特流导致存储和传输的低效率。作为可选择的，子带或子波视频编码提供一种在单一、可缩放的压缩视频
25 比特流中以多分辨率方式对视频序列进行编码的方法。利用子带或子波编码，视频序列将被分解成不同时间和空间的子带。

作为一个简单例子，视频序列被拆分成低分辨率时间子带（大约相当该序列的低帧率版本）和一个高分辨率时间子带（这种子带能与低分辨率时间子带合并来修复原始帧率序列）。单个的视频帧的信息可以同样的被拆分成低分辨率
30 空间子带和多个较高分辨率空间子带。时间和空间的分解可同时使用。每一种

类型的分解可以被重复，例如，就像一个低分辨率的子带可以被进一步分解。通过以不同分辨率为传输或解码选择特定子带，可以实现时间和空间的可缩放性。

5 除此以外，单个子带信息可以用具有多个比特率层的比特平面来表示。对于原始编码信息的保真度可以通过传输一些而不是全部子带的比特选择性地减少（和比特率一起）。或者，保真度可以通过对少于全部的子带比特进行解码选择性地减少（连同处理要求一起）。

10 虽然可缩放视频编码和解码技术使压缩比特流的不同时间、空间和比特保真度的可缩放性更容易，但是在现有可缩放视频编码和解码技术中存在几个缺点。

现有可缩放视频编码和解码技术典型地不能提供在低比特率方面与不可缩放技术相比的性能。在可缩放视频编码和解码技术的性能在较高的比特率和质量方面很好的同时，相比不可缩放视频编码和解码技术，它们在低比特率方面使用太多比特。

15 此外，许多现有硬件和软件工具是根据特定不可缩放视频编码和解码技术设计的。这些工具的使用者不愿购买新的和现存工具不可兼容的可缩放视频编码和解码技术、工具。而且，内容提供者可能不愿生产和安装与多数视频解码工具不兼容的编码内容。

20 有时，解码器以比原始空间分辨率低的空间分辨率播放视频。这可能发生在以下情况，例如，如果解码设备只有一个小屏幕或是通过网络获得较高空间分辨率信息。然而，当在解码过程中以原始空间分辨率发生时间分解时，以较低空间分辨率解码是很成问题的。现有的可缩放视频解码技术不能充分访问这一编码脚本。

25 最后，现有的可缩放视频编码和解码技术在编码和解码期间不能说明在某些判断中的失真的可察觉性。典型地，现有可缩放视频编码技术在某些种类时间分解的低分辨率时间子带中引入了过量可察觉的失真。

举出了数字视频压缩与解压缩的最重要性，对于可缩放视频编码器具有广阔发展领域就不令人惊讶了。无论先前的可缩放视频编码和解码技术如何有益，然而它们并不具有以下技术和工具的长处。

30 发明概述

这里描述的技术和工具改进了可缩放视频编码和解码的性能。例如，3D 子带视频编码器包括嵌入式基层编解码器。该基层编解码器具有几个优点，包括在低比特率下有效压缩以及与现有解码器的基层兼容性。同时，3D 子带视频编码器提供在较高比特率下的时间、空间可缩放性的选项。相应的 3D 子带视频解

5 码器包括嵌入式基层解码器。

根据第一组工具和技术，其中嵌有基层编解码器的 3D 子带视频编码器，对视频编码并输出编码结果。例如：基层编解码器在 3D 子带视频编码器中跟踪一个或更多时间变换并且优先于在编码器中的一个或更多其他的时间变换。可以将基于基层编解码器的输入和基层编解码器的输出的差的余量信息提供给在

10 基层编解码器之后的时间变换。或者，时间转换可以包括相对于基层编解码器输出中的参考图像的动作补偿时间滤波。在一些实施中，3D 子带编码器再次利用动作信息和/或来自基层编解码器的模式信息。

根据第二组技术和工具，其中嵌有基层解码器的 3D 子带视频解码器，对视频解码并输出解码结果。例如，基层解码器在 3D 子带视频解码器中跟踪一个

15 或更多反向时间变换并且优先于在解码器中的一个或更多其他的反向时间变换。可以将余量信息提供给在基层解码器之后的反向时间变换，然后将其和基层解码器的输出合并。或者，反向时间转换可以包括相对于基层编解码器输出中的参考图像的动作补偿时间滤波。在一些实施中，3D 子带解码器再次利用动作信息和/或来自基层解码器的模式信息。

20 本发明的前述和其它的目的、特征和优点将从下面继续进行的参考附图的详细描述中变得更加明显。

附图说明

图 1 是可缩放视频编码和解码的一般的操作环境的框图。

图 2、3、4 和 5 是结合各种描述的可以实现的实施例的一般视频编码器和/

25 或解码器的框图。

图 6 是说明在动作补偿时间滤波 (“MCTF”) 中输出视频帧、时间低通帧和时间高通帧之间关系的图表。

图 7、8 是分别说明了编码器中的 MCTF 和解码器中反向 MCTF (“IMCTF”) 的框图。

30 图 9、10 是分别说明在编码器中具有自适应更新级的 MCTF 和在解码器中

具有自适应更新级的 IMCTF 的框图。

图 11、12、13 是说明自适应更新操作技术的流程图。

图 14、15、16、17、18 和 19 是说明在 3D 子带编码器中嵌入基层编解码器或者在 3D 子带解码器中嵌入基层解码器的不同方案的框图。

5 图 20 是说明当基层编解码器或解码器提供参考候选时的不同参考帧的合并的图。

图 21、22 和 23 是说明用 3D 子带编解码器和嵌入式基层编解码器编码的帧模型的图。

10 图 24 是说明用具有嵌入式基层编码器的 3D 子带编码器进行可缩放编码的技术的流程图。

图 25 是说明用具有嵌入式基层解码器的 3D 子带解码器进行可缩放解码的技术的流程图。

图 26 是说明以较低空间分辨率输出的对 SDMCTF 编码视频进行解码的两种方案的框图。

15 图 27 是说明 SDIMCTF 和 IBIMCTF 中动作补偿预测的框图。

图 28 是说明由于动作移动导致的功率谱泄漏的图解。

图 29 是说明以较低空间分辨率输出的对 SDMCTF 编码视频进行解码的基准方案的图表。

20 图 30 是说明以较低空间分辨率输出对 SDMCTF 编码视频进行解码的一般方案框图。

图 31、32、33 和 34 是说明以较低空间分辨率输出的对 SDMCTF 编码视频进行解码的方案框图。

具体实施方式

25 在三维（“3D”）子波或子带视频编码中，通过几次空间转换和时间转换将视频分解为许多时间-空间子带。这些子带某种程度上是独立的，其中它们中的一些放弃可缩放性。例如：为了解码低空间分辨率视频，空间高通子带被放弃或者解码器只对空间低通子带中的接收数据实施解码。

在这里描述的各种技术和工具可以被独立使用。一些技术和工具可以组合使用（例如，在组合的编码和/或解码过程的不同阶段）。

30 I. 操作环境范例

图 1 说明了适当的计算平台 (100) 的一般例子, 其中几个描述的实施例可以在这里实现。计算平台 (100) 不试图建议对使用范围或功能提供任何限制, 所描述的技术和工具可在各种通常用途或特定用途的计算平台中实现。

参考图 1, 计算平台 (100) 包括了至少一个处理单元 (110) 和存储器 (120)。在图 1 中, 虚线内包括了最基本的配置 (130)。处理单元 (110) 执行计算机可执行的指令并且可以是真实或虚拟的处理器, 在多处理系统中, 多个处理单元通过执行计算机可执行的指令来提高处理能力。存储器 (120) 可能是易失存储器 (例如: 寄存器, 高速缓存, RAM), 非易失存储器 (例如: ROM, EEPROM, 闪存等), 或者是两者之间的一些组合。存储器 (120) 存储利用这里描述的一种或多种技术实现的视频编码器和/或解码器的软件 (180)。

计算平台可以具有附加特征。例如: 计算平台 (100) 包括内存 (140), 一个或多个输入设备 (150), 一个或多个输出设备 (160), 以及一个或多个通信连接 (170)。交互连接结构 (未示出) 诸如总线、控制器或网络, 将计算平台 (100) 中的部件相互连接起来。典型地, 操作系统软件 (未示出) 为其他在计算平台 (100) 中的软件提供操作平台, 并且协调计算平台 (100) 中各部分的运行。

内存 (140) 可以是可删除的或不可删除的, 并且包括磁盘, 磁带或盒式磁带, CD-ROM、DVD 或别的任何能够用于存储信息以及能访问计算平台 (100) 的其他介质。内存 (140) 为软件 (180) 存储指令。

输入设备 (150) 可以是可触摸式输入设备, 像键盘、鼠标、电子笔或跟踪球, 一种声音输出设备、扫描仪或者其他可以向计算平台 (100) 提供输入的设备。因为对音频和视频编码, 输入设备 (150) 可以是声卡、视频卡、TV 调谐卡或能以模拟或数字形式接收音频和视频输入的类似设备, 或者是能将音频和视频抽样读取到计算平台 (100) 中的 CD-ROM 或 CD-RW。输出设备 (160) 可以是显示器、打印机、扬声器、CD 刻录机或其他能从计算平台 (100) 提供输出的设备。

通信连接 (170) 能够通过通信介质和别的计算实体通信。通信介质传送的信息诸如计算可执行指令、音频或视频的输入或输出, 或者其他已调制数据信号形式的数据。已调制数据信号是一种具有一个或多个自身特征组的信号, 或者是一种以类似信号中编码信息的方式变化的信号。通过举例的方式, 而不是

限制，通信介质包括利用电、光、射频（RF）、红外线、声音或其他载波实现的有线或无线技术。

这些技术和工具可以在计算机可读介质的常规上下文中进行描述。计算机可读介质是任何能访问计算机平台的可获得介质。通过举例的方式，而不是限制在计算平台（100）中，计算机可读介质包括存储器（120）、内存（140）、通信介质以及以上任何的组合。

这些技术和工具可以在计算机可执行指令的常规上下文中进行描述，诸如那些包括在程序模块中、以在计算平台中在目标真实或虚拟处理器上来执行的指令。一般来说，程序模块包括能执行特定任务或实现特定抽取数据类型的例程、程序、程序库、目标、类、组件、数据结构等。程序模块的功能在不同实施例中可以按需要在程序模块之间组合或拆分。用于程序模块的计算可执行指令可以在本地或分布式计算平台执行。

出于说明的目的，详细描述用像“信号”、“决定”和“应用”这样的术语来描述计算平台中的计算机操作。这些术语是对计算机执行的操作的高度提炼，并且不应和人类执行的动作混淆。实际的相应于这些术语的计算机操作依赖实施而变化。

II. 视频编码器和解码器的范例

图2是结合各种描述的可以实现的实施例的一般视频编码器系统（200）的框图。图4显示了由一层空间分解跟踪四层时间分解的常规编码器（400）。图3是结合各种描述的可以实现的实施例的一般视频解码器系统（300）的框图。图5显示了具有四层（反向）时间分解和两层（反向）空间分解的组合编码器/解码器系统（500）。

在编码器和解码器内的模块之间显示的关系指出了编码器和解码器中主要的信息流程；为了简明起见没有示出其他关系。特别的是，除了动作信息之外，图2和图3没有表示出指示用于视频序列、帧、宏模块、块等的编码器设置、模式、表格等的侧面信息。这种侧面信息大概在进行侧面信息的熵编码后，在输出比特流中进行发送。

依靠实施和想要的压缩类型，图中显示的编码器或解码器的模块可以被增加、省略、拆分成多个模块、与其他模块组合，和/或被类似的模块替换。在可选实施例中，具有不同模块和/或模块的其它配置的编码器和解码器实现一个或

更多所描述的技术。例如，虽然图 2 中显示了在编码器系统 (200) 中空间分解 (230) 在时间分解 (210) 下游，但编码器系统 (200) 可以替换地或额外在时间分解 (210) 开始前包括一层或多层空间分解。在这种情况下，相应的解码器系统 (300) 可以在反向时间分解 (310) 之后包括一层或多层反向空间分解。

5 3D 子带视频编码对比不可缩放视频编码 (例如，每 H.264 或 VC-1 标准) 的技术等级获得了中等或高比特率的性能。与常规不可缩放方法相比，无论怎样，花费在 3D 子带视频编码中的运动矢量编码的比特量相对较大。因此，3D 子带视频编码在低比特率通常比不可缩放方法执行差，这里编码运动矢量所花费的比特量在总比特率中占较大比重。

10 A 视频编码器范例

3D 子带视频编码是一种使用多级时间/空间子带视频分解来提供多级空间和/或时间缩放能力的一流方法。图 2 说明一般的 3D 子带视频编码方案 (200)。编码器 (200) 接收视频图像 (205) 的序列 (例如：改进的视频帧、隔行扫描视频帧或隔行扫描视频帧的半帧)，并且产生压缩视频信息 (295) 作为输出。

15 视频编码器的特定实施例典型地采用一般编码器 (200) 的变形或增补版本。

编码器 (200) 通过时间子带转换 (210) 来分解视频信号。图 2 显示单一级别时间分解被输出为低分辨率子带和高分辨率子带，但是时间分解可能以包括 4、5 个或一些其它数目级别的分解。

20 在图 2 中，时间分解(210)是动作补偿时间滤波 (“MCTF”)，所以编码器 (200) 执行动作估计 (220) 以计算视频信号动作信息。动作估计 (220) 的确切策略取决于具体实施，并且可以包括，例如，按照子像素提纯进行的整数像素检索、螺旋检索模式、被认为真实的失真率函数最小化或者模仿残余编码消耗、动作信息编码消耗、动作预测模式等。

25 在一些实施例中，编码器 (200) 为在动作估计 (220) 中的可变尺寸区域的动作矢量 (从 4×4 到 16×16) 计算动作矢量，并在 MCTF 中应用这样的动作矢量。例如：对于 16×16 的大区域，编码器 (200) 会为一个 16×16 分区、两个 16×8 分区、两个 8×16 分区或四个 8×8 分区计算动作矢量，这里每个 8×8 分区可以进一步拆分为两个 8×4 分区、两个 4×8 分区或四个 4×4 分区。或者，编码器 (200) 为更大区域 (例如 32×32) 的较高空间分辨率视频 (如
30 CIF) 计算动作矢量以便在只对较低空间分辨率视频 (如 QCIF) 解码时更方便

地利用动作矢量。可替换地，编码器(200)为其他尺度区域的动作矢量计算动作信息。

动作估计 (220) 和 MCTF 的动作精度取决于具体实现。例如：动作估计 (220) 和 MCTF 使用四分之一像素动作矢量。可选择地，动作估计 (220) 和 MCTF 使用一些其他精度的动作信息。例如，为了在只对 QCIF 视频解码时更方便地利用四分之一像素动作矢量，编码器 (200) 为 CIF 视频计算二分之一像素动作矢量和为 4CIF 视频计算整数像素动作矢量。

动作估计 (220) 和 MCTF 的动作矢量方向可能是前向、后向、双向或内部的。在一些实施例中，编码器 (200) 设置动作估计 (220) 和一个大区域每分区的补偿方向为前向、后向、双向或内部中的任何一个。可选择地，编码器 (200) 在一些其他级别设置方向和/或不允许一些方向 (例如内部)。

在一些实施例中，参考帧可以被填充 (例如，通过简单的重复填充) 以便允许参考帧之外的动作估计/补偿。利用双线性滤波器、双三次滤波器、正弦滤波器或一些其他滤波器，参考帧的内插子像素产生抽样值中的分数动作估计或补偿的抽样值。在一些实施例中，编码器使用重叠区域动作补偿 (“OBMC”)。可选择地，动作矢量范围被限制在参考帧边界内和/或不使用 OBMC。

下面将用 5/3 双正多边形子波结构描述几个 MCTF 的例子。可选择地，编码器 (200) 进行其它数字和/或类型的时间子带转换 (210) (例如 Haar) 和/或动作估计(220)。图 2 表示在全部空间分辨率视频被时间转换的的空间域 MCTF。可选择地，编码器 (200) 在时间分解 (210) 之前包括一个或多个空间子带转换，并且 MCTF 是来自原始空间分解的空间子带的带内 MCTF。

在时间转换 (210) 之后，编码器 (200) 提供一个或多个二维 (“2D”) 空间子带转换 (230) 来进一步分解信号。图 2 显示了两级 2D 空间分解的输出，其中具有垂直和水平的低空间分辨率的子带被进一步垂直地和水平地分解。例如：2D 空间子带转换是双值离散子波转换 (“DWT”)。这种子波滤波器可以是，例如，9/7DWT 滤波器或 5/3DWT 滤波器。可选择地，编码器 (200) 执行其它数目 (例如 3 或 4) 的空间子带转换和/或执行其它类型的空间子带转换 (230)。

在时间转换 (210) 和空间转换 (230) 之后，信号被组织成不同的时空子带，其相应于输入视频 (205) 的不同时空分辨率。编码器 (200) 对子带进行熵编码 (240)，例如，在一些实施例中，编码器 (200) 使用具有最佳截断的 3D

嵌入式分组编码 (EBCOT)。JPEG2000 利用 EBCOT 的 2D 模式, 在该实施例中的编码器 (200) 基本上将 2D EBCOT 扩展为 3D 编码。由时间和空间转换产生的每一个子带, 被分成独立编码的 3D 编码块。对于每个编码块, 编码器 (200) 应用比特平面编码和基于上下文的运算编码。可替换地, 编码器 (200) 对子带信息使用某种其它形式的熵编码。

编码器 (200) 也对动作信息 (例如, 动作矢量、动作模式选择信息) 进行编码(250)。例如, 编码器 (200) 使用像简单可变长度码这样的某种形式的熵编码来对动作矢量信息、大块分区和动作估计/补偿的方向进行熵编码。对于动作矢量, 编码器 (200) 对动作矢量与它的预测器之间的差进行编码。动作矢量预测器可以从先前的编码/解码动作矢量中利用中值预测或者某些其它的动作矢量预测技术进行计算。可替换地, 编码器 (200) 利用其它和/或额外的为动作信息的编码 (250) 技术。

拥有子带转换 (210、230), 编码器 (200) 自然能支持空间可缩放性和时间/帧率可缩放性。此外, 具有子带系数比特平面编码, 编码器 (200) 通常也能够支持质量/信噪比 (“SNR”) 的可缩放性。

这些可缩放性对于在有可能出现错误的网络像国际互联网或无线网络上的视频传输是非常有用的。不同的子带可以进行不同纠错或传输错误恢复水平的编码以便传输。或者, 不同子带可以以不同优先级进行传输, 其中网络要支持优先传输。

这些不同的可缩放性也能帮助不同设备分享同一压缩视频流, 甚至不同设备具有不同存储资源、计算能力、显示尺寸、访问带宽等。通过这种方式, 以不同比特率进行编码视频(205)的任务得到简化。

多级别时间分解的一个优点是视频信号的时间冗余能在很大程度上被利用。首先的几个级别的时间子带转换可以利用帧附近中的时间冗余, 最后几个级别的时间子带转换可以利用较长距离交叉的帧间的相关性。

图4显示了具有四层时间分解和一层空间分解的3D 子带视频编码器(400)的例子。实际上, 图4显示了图2中时间子带转换(210)的一种可能的扩展, 并且也显示了不同级别时间分解的数据流。图4也介绍了时间和空间分解的新的标记法。

在图4中, 编码器(400)对具有每秒 f 帧的帧率的视频提供四级时间子带

转换。每级时间子带转换 T_i 把输入到本级的信号分成两个时间子带—时间低通子带和时间高通子带。从而每个子带具有输入到本级视频信号的一半的帧率。在图 4 中, 字母 $t-L_i$ 和 $t-H_i$ 分别代表在第 i 级时间转换 T_i 之后输出的低通子带和高通子带。 $t-L_i$ 和 $t-H_i$ 的帧率均为 $f/2^i$ 。图 4 中, 信号 $t-L_i$ 被用时间子带转换 T_{i+1} 分解成几个迭代以便利用长距离相关和使得较低帧率的视频能被解码器解码。

用简单的几个例子来说明, 假定输入视频是每秒 30 帧 (“fps”) 的 CTF 视频。从最先的时间分解中输出的低通子带 $t-L_1$ 是 15 fps 的视频, 从最先的时间分解中输出的高通子带 $t-H_1$ 与此一样。从第二次时间分解中输出的子带 $t-L_2$ 和 $t-H_2$ 是 7.5 fps 的视频, 从第三次时间分解中输出的子带 $t-L_3$ 和 $t-H_3$ 是 3.75 fps 的视频。最后, 从第四次时间分解中输出的子带 $t-L_4$ 和 $t-H_4$ 是 1.875 fps 的视频。在本例中空间分辨率对于所有输出的低通和高通子带均是 CIF。

帧中的动作信息以不同级别进行估计, 而每一级别的时间转换是根据当前级别的运动轨迹来执行的。这样每个级别的时间分解产生了随后被编码的动作矢量。图 4 中, 相应于四级时间转换, 有四套动作矢量。

输出时间低通子带 $T-L_4$ 和输出时间高通子带 $t-H_1$ 、 $t-H_2$ 、 $t-H_3$ 和 $t-H_4$ 用单一级别的 2D 空间子带转换 S_1 来分解, 这将各自的输入子带分解成低通水平、低通垂直 (“LL”), 低通水平、高通垂直 (“LH”), 高通水平、低通垂直 (“HL”) 和高通水平、高通垂直 (“HH”) 子带。可替换地, 编码器 (400) 在时间分解级别之前和/或之后包括其它和/或附加级别的时间分解, 或者编码器 (400) 使用不同的空间分解模式。

继续以上以图 4 开始的几个例子, 低通子带 $t-L_4$ 具有 CIF 空间分辨率。随着单一级别的 2D 空间子带转换 S_1 , 时间子带 $t-L_4$ 被分解成 LL 子带、LH 子带、HL 子带和 HH 子带。LL、LH、HL 和 HH 子带中的每一个具有 QCIF 分辨率。相似的是, 每个时间高通子带 $t-H_1$ 、 $t-H_2$ 、 $t-H_3$ 和 $t-H_4$ 以 CIF 分辨率开始并被分解成为具有 QCIF 分辨率的 LL、LH、HL 和 HH 子带。

被空间分解的子带接着被熵编码并输出。

B 视频解码器的范例

图 3 说明了一般的 3D 子带视频解码方案 (300)。解码器 (300) 接收压缩视频信息 (395) 并产生视频图像 (305) 序列 (例如, 改进视频帧、隔行扫描

视频帧或者隔行扫描视频半帧)作为输出。视频解码的特定实施例典型地使用一般解码器(300)的变形或增补版本。

解码器(300)对一些或全部空间-时间子带的信息(395)进行熵解码(340)。视频信息(395)仍被组成不同的空间-时间子带,其相应于输入视频(205)的不同的时间分辨率。以这种方法,解码器(300)可以在不同的空间-时间分辨率之间进行选择,忽略压缩比特流(395)的部分,或解码器(300)可以只对解码器(300)实际收到的压缩比特流(395)的某些子集进行解码(340)。解码器(300)实际上只对解码器(300)解码的信息执行在编码器(200)中进行的熵编码的(240)的反向处理。例如:解码器(300)接收并对以3D DBCOT编码的信息进行解码(340)。可替换地,解码器(300)对一些或全部子带信息采用某些其它形式的熵解码。

当解码器(300)执行反向MCTF(“IMCTF”)时,解码器(300)对动作信息(例如,动作矢量、动作模式选择信息)进行解码(350),在许多情况下执行在编码器(200)中执行的编码(250)的反向处理。例如,解码器(300)使用像单一可变长度编码的解码这样的某种形式的熵解码。对动作矢量,解码器(300)利用中值预测或某些其它动作矢量预测技术来计算动作矢量预测,并且之后解码器(300)用动作矢量微分合并动作矢量预测。可替换地,解码器(300)使用其它和/或额外的解码(350)技术。

解码器(300)提供一个或多个2D反向空间子带转换(330)以空间地重建视频。图3显示两级2D反向空间分解的输入和输出。例如:2D反向空间子带转换是双值反向DWT(“IDWT”)。可替换地,解码器(300)执行其它数目和/或类型的反向空间子带转换(330)。

解码器(300)通过一个反向时间子带转换(310)来进一步重建视频信号。图3显示了来自低分辨率子带和高分辨率子带的单一级别的时间重建,但是该时间组合可能包括4、5个或一些其它数目级别的合成。

在图3中,反向时间转换(310)使用IMCTF,而解码器(300)利用已解码(350)动作信息对视频信号进行动作补偿。通常,动作信息和IMCTF的属性(例如,动作补偿分区、动作矢量的精确度、动作补偿方向、参考帧填充的使用、子像素插值、OBMC的使用)是在相应的编码器(200)中并且在压缩视频比特流(395)中发送信号。下面描述了IMCTF的几个例子。可替换地,解码

器(300)执行其它数目和/或类型的反向时间子带转换(310)和/或动作补偿。
图3显示了空间域的IMCTF,其中利用反向时间变换重建了全部空间分辨率视频。可选择地,解码器(300)执行带内MCTF。

回到图4的编码器(400),相应的解码器只能通过 $t-L_i$ 重建 $f/2^i$ 帧率的视频。
5 或者,解码器能通过 $t-L_i$ 和 $t-H_i$ (包括随后的时间高通子带的信息 $t-H_{i+1}$ 等)重建 $f/2^{i-1}$ 帧率的视频。

图5显示编码器中具有四层时间分解和两层空间分解以及在解码器中具有相应数目反向转换的3D子带视频编解码器(“编解码器”)系统(500)的例子。实际上,图5显示了图2和3中的编码器(200)和解码器(300)的其它
10 可能版本。图5还介绍了其它更简明的时间和空间分解以及他们反向操作的表示法。

在图5中,编码器向视频提供四个级别的时间子带转换 T_i (例如, T_1 、 T_2 、 T_3 和 T_4)以及向时间分解的结果提供两个级别的空间子带转换 S_i (例如 S_1 和 S_2)。编码器接着执行熵编码 E 。解码器执行熵解码 E^{-1} 并对熵解码执行两级反向
15 空间子带转换 S_i^{-1} (例如 S_2^{-1} 和 S_1^{-1})。解码器接着向空间重建的结果提供四级反向时间子带转换 T_i^{-1} 的四个级别(如 T_4^{-1} 、 T_3^{-1} 、 T_2^{-1} 和 T_1^{-1})。

III.动作补偿的时间滤波

为了提高编码效率,在时间分解中提供动作调整时间转换(这里可选择地称作动作补偿时间滤波(“MCTF”))以便有效删除时间冗余。MCTF的主要思
20 想是当应用时间分解时,在不同帧中调整像素。动作调整3D编码方案能提供像无补偿一样的可缩放性,或者甚至比其它编码方案更好的编码效率。

在各种MCTF方案中,基于提升的MCTF提供了灵活而有效的实施。图6说明了在给定的5/3正多边形子波的基于提升MCTF方案中输入视频帧、时间低通帧和时间高通帧之间的关系。

图6显示了输入到MCTF的四个原始、索引视频帧的像素栏。(输入帧本身可以是,例如,从MCTF的前一级别中输出的时间低通子带。)图6中的基于提升的子波变换包括两个基本阶段:动作补偿预测(“MCP”)阶段和动作补偿更新(“MCU”)阶段。
25

在预测阶段,对于奇数帧(例如,帧 I_{2i+1}),来自一个或多个相邻偶数帧(例如,帧 I_{2i} 和 I_{2i+2})的动作补偿指示该奇数帧像素的动作补偿预测。高通帧(例
30

如 H_i) 实质上是奇数帧和对奇数帧的动作补偿预测的差。

在更新阶段, 一个或多个可利用的高通帧 (例如帧 H_i 和 H_{i-1}) 的动作补偿指示了动作补偿的更新。低通帧 (例如帧 L_i) 实质上是更新和偶数帧 (例如帧 I_{2i}) 之间的差。

- 5 图 7 说明编码器中具有提升的常规 MCTF, 图 8 说明了解码器中相应的 IMCTF。为保持一致, 在本申请的图 7 和图 8 中以及其它地方, 偶数视频帧作为动作补偿预测的参考, 并被用作低通帧的信息, 以及奇数视频帧是 MCP 的主体并用作高通视频帧的信息。可替换地, 奇数帧和偶数帧的角色可交换。

10 在图 7 的 MCTF (700) 中, 偶数视频帧被用作 MCP (710) 的参考。对于特定奇数帧 I_{2i+1} , 编码器根据一个或多个相邻的偶数帧 I_{2i} 和 I_{2i+2} 计算动作补偿预测。在预测与原始奇数帧 I_{2i+1} , 之间的差就是时间高通帧 H_i 。

高通帧 H_i 在 MCU (720) 中被用作参考。对于特定偶数视频帧 I_{2i} , 编码器根据一个或两个高通帧 H_{i-1} 和 H_i 为相邻的奇数视频帧计算动作补偿更新。更新与原始偶数帧 I_{2i} 的合并就是时间低通帧 L_i 。

- 15 例如, 假设视频帧序列 $I_0, I_1, \dots, I_{2n-1}$ 要用图 7 中的 MCTF (700) 处理。预测模块 (710) 按如下所示根据连续的偶数帧 I_{2i} 和 I_{2i+2} 来计算奇数帧 I_{2i+1} 的预测 $P(I_{2i+1})$ 。

$$P(I_{2i+1}) = \frac{1}{2} (MC(I_{2i}, MV_{2i+1 \rightarrow 2i}) + MC(I_{2i+2}, MV_{2i+1 \rightarrow 2i+2})) \quad (1),$$

- 20 在这里 $MV_{2i+1 \rightarrow 2i}$ 表示从帧 I_{2i+1} 到帧 I_{2i} 的动作矢量, $MV_{2i+1 \rightarrow 2i+2}$ 具有相似意思, 而 $MC()$ 表示来自相邻偶数帧的当前奇数帧的预测的至少一部分的动作补偿。(等式(1)意思是说明用于提供动作矢量的大块、大块分区或者是其它部分的奇数帧的 MCP, 并且这样的 MCP 被用于重复奇数帧其它的大块、大块分区等。) 在一些情况下, 编码器在计算动作补偿预测时 (像在 Haar 转换中) 只考虑相邻偶数帧中的一个 (而不是两个), 相应地改变等式 (1) 中的 MC 属性和权重;
- 25 或者编码器利用内部压缩对部分或全部奇数帧进行编码。结合嵌入式基层编解码器, 编码器可以用除相邻偶数帧以外的帧作为参考帧, 如下面所述。

预测模块(710)接着按如下所示计算高通帧 H_i 。

$$H_i = I_{2i+1} - P(I_{2i+1}) \quad (2).$$

- 30 在更新步骤 (720), 其接着预测步骤 (710) 来完成一级时间转换, 编码器按如下所示根据相邻高通帧为偶数帧 I_{2i} 计算更新信息 $U(I_{2i})$ 。

$$U(I_{2i}) = \frac{1}{4} ((MC(H_{i-1}, MV_{2i \rightarrow 2i-1}) + MC(H_i, MV_{2i \rightarrow 2i+1})) \quad (3)。$$

这里 $MC()$ 表示根据相邻高通帧为偶数帧产生至少部分更新的动作补偿。作为一种简单方法, MCU 的动作矢量可以用相应 MCP 动作矢量的反转得到。(等式 (3) 意思是说明用于提供动作矢量的大块、大块分区或者是其它部分的偶数帧的 MCU , 并且这样的 MCU 被重复用于偶数帧其它的大块、大块分区等。) 5 在一些情况下, 编码器在计算动作补偿更新时只考虑高通帧中的一个(而不是两个), 相应地改变等式 (3) 中的 MC 属性和权重; 或者编码器两个高通帧均不考虑。

编码器接着按如下所示产生低通帧。

$$L_i = I_{2i} + U(I_{2i}) \quad (4)。$$

在图 8 的 $IMCTF(800)$ 中, 高通帧 H_i 在 $MCU(820)$ 中被用作参考。对于特定偶数视频帧 I_{2i} , 解码器根据高通帧 H_{i-1} 和 H_i 中的一个或全部为相邻奇数视频帧计算动作补偿的更新。在更新和相应的偶数帧 I_{2i} 的时间低通帧 L_i 之间的差就是重建的偶数帧 I_{2i} 。

15 在重建时, 偶数视频帧在 $MCP(810)$ 中被作为参考。对于特定奇数视频帧 I_{2i+1} , 解码器根据相邻重建偶数帧 I_{2i} 和 I_{2i+2} 中的一个或者两个计算动作补偿的预测。预测与奇数帧 I_{2i+1} 的相应时间高通帧 H_i 的合并就是重建的奇数帧 I_{2i+1} 。

在编码器中, 预测步骤的通常的目标是和用于预测的动作矢量一起最小化或者减少编码高通帧(在空间分解和熵编码中)所需的比特率。这样, 高通帧 H_i 20 实质上是相应的来自相关偶数索引原始视频帧的奇数索引输入视频帧 I_{2i+1} 的双向 MCP (或者前向 MCP , 或者后向 MCP , 视情况而定)的余量。接着, 原始偶数索引帧 I_{2i} 用预测余量更新为低通帧。

高通帧的能量取决于动作模式的成功, 后者取决于运动的复杂性、动作交叉帧的一致性和动作模式自身。如果动作预测是精确的, 那么预测残余通常具有很小幅度。甚至在动作被真实捕获时, 然而, 更新步骤仍是有用的, 由于沿着运动轨迹的低通滤波他们能减少噪音和失真。通常, 更新步骤有助于低通帧 25 中高频部分的删除, 因此提高了编码效率。

如果动作预测不准确(例如, 由于复杂、不可预测的动作或者动作模式自身的不足), 目标特征可能不被校正, 引发了几个问题。不准确的动作预测可

能导致高通子带帧中的多边沿和能量的增加。这不仅仅损害了高通子带自身的编码压缩性能，而且也在更新步骤中增加了时间低通帧的边沿和轮廓的重影。

更普遍的是，当具有时间可缩放性时更新步骤可能影响解码性能。尤其是，例如，由于有限的信道带宽或设备能力，当一些时间高通帧部分或全部丢失时，5 解码器中可能出现的问题。（通常，IMCTF 中的时间可缩放性的获得是通过保留时间低通子带同时丢弃时间转换域中的高通子带。）当动作预测无效时，在编码的更新步骤期间高通帧将噪声带给低通帧（也就是说，较低帧率的视频）。当视频仅仅根据时间低通帧重建时，高通帧的影响不能被解码器中的 MCU 消除。或者，如果时间高通帧信息被粗劣地量化，那么由于高通帧的影响只能在 MCU 10 中被近似地删除，时间低通帧的质量可能受到损害。因此，在更多比特被分配给低通帧用于编码时，低通帧中的重影也不能被消除。

一种可能的解决方法是在编码器中跳过以提升为基础的MCTF中的更新步骤。这会显著降低编码器的整体性能，因为不管怎样，更新步骤中由于沿运动轨迹的低通滤波能够减少噪音和失真。而且，更新步骤通过在时间高通子带和 15 时间低通子带中分散失真来减少解码器中的整体失真。

IV .在MCTF和IMCTF中的自适应更新

在一些实施例中，可缩放视频编码器在以提升为基础的MCTF中自适应地执行更新步骤和/或可缩放视频解码器在IMCTF中自适应地执行更新步骤。通过利用人工可视系统的特征，自适应更新步骤提高从时间低通子带解码的低帧 20 率视频的可视质量。同时，对比常规更新步骤，自适应更新步骤为从时间低通和时间高通子带解码的较高帧率的视频保持了令人满意的性能。

A 编码器和解码器范例

自适应更新方案在两个相互矛盾的目标：（1）尽可能多地增加原始更新信息和（2）同时减少由更新步骤产生的视觉噪声之间取得了良好的平衡。

25 图9描述了编码器中具有自适应更新操作的MCTF框架(900)，图10描述了解码器中具有自适应更新操作的IMCTF框架(1000)。例如，框架(900, 1000)像在参考图2-5的描述那样被并入编码器和/或解码器之中。可替换地，自适应更新操作被并入到另一种编码器和/或解码器中。

30 参考图9，MCTF框架(900)包括预测(910)和更新(920)阶段。这些阶段基本上像参考图7的描述那样继续进行。

MCTF 框架 (900) 使用人工可视系统 (“HVS”) 模型 (930), 按照视觉质量对来自预测步骤的信号进行估计。例如: 编码器根据 HVS 模型(930)对偶数帧中的信息进行估计。在一些实施例中, HVS 模型 (930) 采用恰巧留意到差异 (JND) 的进制。而在其它实施例中, HVS 模型 (930) 使用其它已知标准。

- 5 HVS 模型 (930) 为阈值 (940) 阶段提供阈值信息, 该阶段考虑阈值信息进行估计并将结果提供给更新 (920) 阶段。例如: HVS 模型(930)给根据这一模型估计的偶数帧的每一像素提供阈值。阈值的各种例子及它们在更新操作的使用将在下面描述。可替换地, HVS 模型 (930) 提供不同的阈值信息。

- 10 阈值 (940) 阶段的结果是更新信息可以被限制在失真特别明显的图像的部分中, 同时更新信息不对图像的其它部分进行限制。因此根据人们的感觉, 在更新信息被提供给偶数帧之前更新信息可以被自适应的删除或限制。

参考图 10, IMCTF 框架(1000)也包括预测 (1010) 和更新 (1020) 阶段。这些阶段基本上像参考图 8 的描述那样进展。

- 15 IMCTF 框架(1000)使用 HVS 模型 (1030) 按照可视质量估计信号。例如: 解码器根据 HVS 模型(1030)估计低通帧中的信息。在一些实施例中, HVS 模型 (1030) 使用恰巧留意到差异 (“JND”) 的进制。在其它实施例中, HVS 模型 (1030) 使用其它已知标准。

- 20 HVS 模型 (1030) 向阈值阶段 (1040) 提供阈值信息, 阈值阶段参考阈值信息进行估计和为更新阶段 (1020) 提供结果。例如, HVS 模型 (1030) 提供按照该模型估计的低通帧的每像素的阈值。阈值的各种范例及它们在更新操作中的运用将在下文描述。可替换地, HVS 模型 (1030) 提供不同的阈值信息。

- 25 比较图 9 和图 10, 编码器方的 HVS 模型 (930) 被提供给偶数帧, 而解码器方的 HVS 模型 (1030) 被提供给低通帧。即使在同样的 HVS 模型 (930, 1030) 被用在编码器方和解码器方, 阈值也会有细微的差别。对于相应的偶数和低通帧来说阈值可能很类似是在可容忍的限度内, 因为编码器和解码器状态间的偏差是帧的局部 (相对于随时间累积和变坏)。

此外, 在图 9 和 10 的框架 (900, 1000) 中, 阈值在每一级 MCTF/IMCTF 进行计算。就本身而言, 阈值即使对于时间对应的帧在不同级的 MCTF/IMCTF 也可能是不同的。

- 30 因此, 在一些实施例中, 编码器方的 MCTF(900)和解码器方的 IMCTF(1000)

在 HVS 模型(930,1030)中应用同样的 JND 进制。由于阈值在编码器和解码器中被独立计算和应用，所以没有对和阈值相联系开销进行编码或阈值操作。可替换地，编码器向解码器发送阈值信息或进行判定，或者解码器提供与编码器不同的 HVS 模型。

5 B HVS 模型和阈值操作的范例

编码器中的自适应更新步骤通常可以按如下所示表示： $L_i = I_{2i} + f(U_{2i})$ 。函数 $f()$ 是为偶数帧 I_{2i} 提供更新 U_{2i} 的自适应函数。 L_i 表示作为结果的低通帧。

自适应函数利用人工可视模型的结果和推展。已经进行深入的研究以便发展基于人工可视系统的计算模型。在各种模型中，明确的标准是基于在 HVS 灵敏度以及在空间和时间域中亮度变化与对比度变化的激励之间的不相容性的观察。

在 HVS 的大量计算模型中，JND 被广泛用在可视编码和图片水印中。JND 使用定义为亮度边沿振幅函数的可视阈值，其中扰动一直增加到人们刚好能看清为止。更多信息参见 A.N.Netravali 与 B.Prasada 所著的“Adaptive Quantization of Picture Signals Using Spatial Masking”，Proc. IEEE, 卷号 65, 页数 536-548 (1977 年 4 月)。JND 与 HVS 的纹理屏蔽特性有着密切关系。根据这一标准，噪音在平面和没有纹理的区域中更明显，而噪音在具有边沿和纹理的区域中就比较少见。

JND 阈值是图像的相关。只要更新信息保持小于 JND 阈值，在低通帧中就可获得“更新残余”的透明度。就本身而言，JND 像在自适应更新方案中的 HVS 模型一样发挥很好的作用。在自适应更新方案的实施例中，JND 模型按如下所示定义。

$$JND_x(i, j) = 1 - \frac{1}{1 + \theta \sigma_x^2(i, j)} \quad (5)$$

其中 $\sigma_x^2(i, j)$ 表示中心在坐标为 (i, j) 的像素的窗口内的图像 x 的局部变量。

25 JND 可以只根据亮度抽样计算来降低计算的复杂性。或者，JND 可以根据亮度抽样和色度抽样来计算。窗口尺寸决定于实施。一个范例窗口的尺寸是 3×3 ，另外一个为 5×5 。 θ 是为特定图像选择的调整参数。这与噪音能见度函数值一样，假定了图像是非静止高斯的。调整参数 θ 在 JND 定义中起对比度调整的作用。为了使调整参数 θ 与图像相关，它可以按照如下所示进行计算：

$$\theta = \frac{D}{\sigma_{x_{\max}}^2} \quad (6),$$

这里 $\sigma_{x_{\max}}^2$ 是给定图像最大局部变量, $D \in [50, 100]$ 是实验决定参数。根据 JND 的定义, 由于 $\sigma_x^2(i, j)$ 很小, JND 的值在平整区域也很小, 反之亦然。

更精确的表示编码器中自适应更新的方法是:

$$L_i = I_{2i} + f(I_{2i}, U_{2i}) \quad (7),$$

其中

$$f(I_{2i}, U_{2i}) = \begin{cases} U_{2i} & |U_{2i}| < JND_{I_{2i}} \cdot S \\ JND_{I_{2i}} \cdot S & U_{2i} \geq JND_{I_{2i}} \cdot S \\ -JND_{I_{2i}} \cdot S & U_{2i} \leq -JND_{I_{2i}} \cdot S \end{cases} \quad (8).$$

$JND_{I_{2i}}$ 是按上面定义的 $JND_x(i, j)$ 的例子, S 表示取决于实施的强度因子, 例如 12.5。JND 函数与可视屏蔽相关的局部帧特征相适应。如果局部亮度变量很大, JND 相对大 (根据等式 (5) 接近 1), 指示为繁忙或复杂的区域。另一方面, 如果局部亮度变量小, JND 相对小 (根据等式 (5) 接近 0), 指示为平整的区域。它是根据要在编码器中更新的偶数索引帧以及解码器中的偶数索引帧 (实际上, 低通图像近似阈值) 计算的。结果, 自适应更新可以有效地减少高通帧导致的重影以及提供时间缩放的编码性能。

像上面提到的, 在一些实施例中, 编码器和解码器使用相同的 JND 进制和阈值 (随等式 (8) 中索引值等而改变), 对于自适应更新操作没有任何开销要进行编码和分发给解码器。虽然编码器和解码器操作不同的图像 (在编码器中的原始帧和解码器中重建的低通版本), 实验结果显示在解码器中更新屏蔽的结果是很接近的编码器中的近似值的。

可替换地, 编码器和/或解码器利用其它定义的 JND、其它 HVS 模型和/或其它阈值操作。例如可以采用其它和/或附加参数。

C 自适应更新技术的范例

图 11 到图 13 说明了自适应更新操作技术 (1100, 1200, 1300)。一个像参考图 9 描述的那样的编码器实现图 11 和 12 中所示的技术 (1100, 1200), 一个像参考图 10 描述的那样的解码器实现图 11 和 13 中所示的技术 (1100, 1300)。可替换地, 其它编码器或解码器都可以实现该技术 (1100, 1200, 1300)。

图 11 显示自适应更新操作的简单、概括的技术(1100)。编码器或解码器为自适应更新计算 (1120) 阈值。解码器或编码器接着执行 (1160) 自适应更新

操作。

图 12 显示在编码器中自适应更新操作的详细技术 (1200)，包括附加特征 MCP。

对于当前为之计算低通帧的偶数帧，编码器计算(1220)阈值。例如；编码器
5 器为像在前面部分描述的当前偶数帧的像素计算 JND。可替换地，编码器采用某些其它进制计算(1220)阈值。

对于 MCP，编码器根据当前偶数帧计算 (1230) 一个或多个动作补偿预测，
例如，前向和/或后向预测相邻奇数帧。编码器可以根据其它的偶数帧为相邻奇
数帧计算 (1230) 一个或多个其它的动作补偿预测（或者这种预测可以根据较
10 早偶数帧提前地进行计算）。编码器 (1240) 接着根据动作补偿预测和他们相应的相邻奇数帧来计算一个或多个高通帧。

接下来，编码器根据一个或多个高通帧为当前偶数帧计算 (1250) 动作补
偿更新。例如，编码器反向应用在 MCP (1230) 期间根据当前偶数帧计算的动
作信息。可替换地，编码器使用不同的机制来计算 (1250) 更新。

编码器接着为当前偶数帧进行 (1260) 自适应更新，并以低通帧作为结果。
15 例如：编码器利用动作补偿更新和前面部分描述的 JND 阈值来进行自适应阈值操作。可替换地，编码器可以利用不同的阈值机制来进行 (1260) 自适应更新。

编码器 (1290) 决定是否继续使用 MCTF，以及如果继续，则以下一个偶
数帧继续。

20 图 13 显示在解码器中的自适应更新操作的详细技术 (1300)，包括不同帧率的分开的解码路径。

解码器为解码确定 (1310) 时间分辨率。时间分辨率可以根据使用者等级
设置或某些其它设置来确定，根据解码器实际接收的信息量的多少（例如，只
有通过易于出错信道的低时间相应息）来确定的，或者由某种其它方法决定。
25 可替换地，解码器可以只有一条解码路径。例如，一个解码器可以包括低帧率
路径同时另一个解码器包括高帧率路径，以适应各自解码器的资源。

在低帧率路径中，解码器根据接收到的低通帧信息（例如：使用低通帧作
为当前帧）来计算 (1312) 当偶数帧。由于高相应息和动作信息被忽略或不能
接收，所以不执行自适应更新操作。即使缺少这种信息，由于编码期间的自适
30 应更新操作，由于在编码期间偶数帧显著失真的增加被限制，所以当前偶数帧

的质量被提高了（和常规 IMCTF 相比）。

在高帧率路径中，对于当前偶数帧，解码器根据被解码器接收的相应低通帧计算（1320）阈值。例如：解码器为前面部分描述的低通帧的像素计算 JND。可替换地，解码器使用某个其它进制计算（1320）阈值。

- 5 解码器根据一个或多个高通帧来为当前偶数帧来计算（1330）动作补偿更新（值）。例如：解码器反向应用在当前偶数帧的 MCP（1230）期间计算的动作信息。可替换地，解码器可使用不同机制来计算（1330）更新。

- 解码器接着对低通帧执行（1360）自适应更新来重建当前偶数帧。例如：解码器利用更新和前面部分中描述的 JND 阈值来执行自适应阈值操作。可替换地，解码器可以利用不同阈值机制来执行（1360）自适应更新。

 对于 MCP，解码器根据重建的当前偶数帧来计算（1370）一个或多个动作补偿预测，例如：前向和/或后向以便预测相邻奇数帧。（其它给相邻奇数帧的动作补偿预测可以由较早偶数帧被提前地计算。）解码器接着根据奇数帧和相应高通帧的动作补偿计算（1380）重建的奇数帧。

- 15 解码器确定（1390）是否继续 IMCTF，以及如果是，则用下一个偶数帧继续。

 在图 12 和图 13 中显示的各种操作都可以被分解、与其它操作组合，或重新排列。例如，虽然图 12 以并行方式显示某些操作，然而这些操作可以代替地以串行方式执行。

20 V. 3D 子带视频编码和解码中的嵌入式基层编解码器

- 在一些实施例中，可缩放视频编码器包括嵌入式基层编解码器和包含在嵌入式基层解码器中的可缩放视频解码器。基本上，由嵌入式基层编解码器产生的比特流是作为低比特率、3D 子带视频编码比特流的基层用的。在一些可缩放视频编码/解码系统的背景中，例如：基层视频以最小空间、时间和 SNR 分辨率为系统提供视频，根据加入系统的附加的增强层为系统提供缩放允许的最大空
- 25 间、时间和 SNR 分辨率。（在编码和解码中基层的确切作用在不同实施例中不同。）例如：3D 子带视频编码器包括嵌入式视频编解码器而 3D 子带视频解码器包括相应的嵌入式视器解码器。

- 对于许多类型的嵌入式编解码器，这提高了低比特率的编码性能，其中这
- 30 比特在常规 3D 子带编码中编码动作信息花费的比特消耗了可用比特率的无效

率的部分, 相比较而言, 嵌入式视频编解码器更有效率。在一些情况下, 嵌入式编解码器提供诸如闭环时间预测、内环滤波和失真率优化而没有与 3D 子带编码相关的制约的优点。同时, 3D 子带在中等和较高比特率编码的优点(例如: 空间和时间的可缩放性) 被保持下来。

5 在 3D 子带视频编码中嵌入视频编解码器的另一个优点是嵌入式编解码器和相应解码器提供兼容。例如: 如果 MPEG-1 编解码是嵌入式, 任何具有 MPEG-1 解码能力的接收机能对压缩视频(即使 MPEG-1 编解码器本身没有改进编码性能) 的粗略版本进行解码。如果 H.264 或 VC-1 编解码器是嵌入式的, 除了兼容性外, 这还在低比特率的情况下提高了编码性能。

10 这里描述的技术和工具并不是对具有性能和/或兼容上的优点的系统的限制。除了性能和兼容性, 在可缩放视频编码器中的嵌入式基层编解码器和在可缩放视频解码器中的嵌入式视频解码器还具有其它优点。

A 编码器和解码器范例

15 图 14 至图 19 说明了在编码器或解码器中分别嵌入基层编解码器或解码器的三种不同方案。在编码器中, 用于基层的编解码器被嵌在时间子带转换中, 空间子带转换跟着时间子带转换之后。附加的空间子带转换可以放在时间子带转换之前和/或之后, 也对解码器中的可能位置作相应地改变。基层编解码器可以被放在各种转换间的不同位置, 相应地改变在解码器中的可能位置。

20 通常, 嵌入式编解码器的位置为基层视频设置了帧率和空间分辨率。为了简化在图 14 至图 19 中显示三个方案之间的比较, 基层分辨率被设置为原始视频的四分之一帧率和二分之一的空间分辨率。这样, 如果原始视频是 30 fps CIF 视频, 例如, 基层视频就为 7.5 fps QCIF 视频。

25 这样, 在图 14、16 和 18 的编码器中基层编解码器是在两个时间子带转换之后, 在图 15、17 和 19 中的解码器中基层解码器在两个反向时间子带转换之前。改变基层编码器和解码器在时间转换中的位置是一种改变基层帧率的方法。例如: 通过图 14、16 和 18 中显示嵌入式基层模块接收 $t-L_2$ 子带作为输入, 可替换地基层模块被放在 $t-L_1$ 子带转换或其它子带转换之后。通常, 基层模块在时间转换中的位置取决于想要的基层视频的帧率。

30 在图 14、16 和 18 中, 基层空间分辨率的改变是通过抽样操作或空间转换操作以及一些随机因素(相邻嵌入式编解码器或解码器) 来完成的。改变抽样/

空间子带转换操作的比率是一种改变基层空间分辨率的方法。

图 14、16 和 18 中包括时间子带转换模块 T、空间子带转换模块 S 和 MV 编码及熵编码模块，还有这些模块的输入和输出。图 15、17 和 19 包括反向时间子带转换模块 T^{-1} 、反向空间子带转换模块 S^{-1} 、MV 解码和熵解码模块，还有
5 这些模块的输入和输出。一般，这些模块的作用和它们的输入/输出如参考图 2 至 5 讨论的那样。

图 14、16 和 18 中所示的基层编解码器,以及 图 15、17 和 19 中相应的基层解码器可以是产生遵从标准的比特流(例如 MPEG-1, MPEG-2, MPEG-4,H.261, H.263, H.264/AVC,或者 VC-1)。或者, 基层编解码器/解码器能成为商品(如:
10 WMV8,WMV9)。更普遍地, 基层编解码器/解码器可以是任何能够获得的具有期望性能和/或兼容特性的编解码器/解码器。

1 第一种方案

图 14 显示了根据第一种方案的 3D 子带编码器 (1400), 编码器 (1400) 包括嵌入式基层编解码器 (1420) 以及在基层编解码器 (1420) 左右的降抽样
15 模块 (1410) 和升抽样模块 (1430)。

降抽样模块 (1410) 和升抽样模块 (1430) 是可选的。它们在嵌入式编解码器 (1420) 进行编码的视频具有比输入到嵌入式编解码器 (1420) 的时间子带低的
空间分辨率时被使用。另外, 降抽样模块(1410)和一个升抽样模块(1430)可以被跳过。

20 在图 14 中虚线框包括了与在两级时间分解之后和基层编解码器 (1420) 相关的操作。对于基层, 输入视频是每秒 $f/4$ 帧, 而分辨率可以被降抽样模块(1410)减少。因此, 基层视频提供低帧率和(潜在的)原始视频低空间分辨率的表示。

降抽样模块 (1410) 使用简单的子抽样、低通滤波器以及其它滤波机制对 $t-L_2$ 时间子带进行降抽样。例如: 降抽样模块 (1410) 利用两个水平或垂直因子
25 减少空间分辨率。在一些实施例中, 降抽样模块 (1410) 采用 MPEG 滤波器, 像在文献由 Bjontegaard 等著文献“H.263 Anchors – Technical Description”中描述的那样, 或者 Daubechies 9/7 DWT 滤波器中被描述。尤其是, 降抽样模块 (1410) 根据任一随机比率改变嵌入式编解码器 (1420) 的输入的空间分辨率来获得想要的空间分辨率和/或匹配基层编解码器 (1420) 允许的输入空间分辨率。
30 率。

嵌入式编解码器 (1420) 利用嵌入式编解码器 (1420) 的常用操作来对降抽样的 $t-L_2$ 时间子带 (所示的 $ts-LL$ 子带) 编码。嵌入式编解码器 (1420) 以基层分辨率生成输出的基层比特流 (1495), 其中基层比特流 (1495) 和相应的基层解码器是兼容的。为了在编码器 (1400) 中后续编码的目的, 嵌入式编解码器 (1420) 还利用嵌入式编解码器 (1420) 的相应解码操作来对 $ts-LL$ 子带进行解码, 以生成重建子带 $ts-LL'$ 。

升抽样模块 (1430) 利用某些滤波机制对从基层编解码器 (1420) 的重建输出的子带 $ts-LL'$ 进行升抽样, 以产生时间子带 $t-L_2$ 的重建版本 $t-L_2'$ 。重建子带 $t-L_2'$ 具有与 $t-L_2$ 相同的分辨率。例如: 升抽样模块 (1430) 通过两个水平或垂直因子来增加空间分辨率。在一些实施例中, 升抽样模块 (1430) 采用简易 6 抽头滤波器 $\{-1, 5, 20, 20, 5, -1\}$ 、像在由 Chen 等著 “Improved MC-EZBC with Quarter-pixel Motion Vectors” 中描述的 8 抽头正弦滤波器或者 Daubechies 9/7 IDWT 滤波器。升抽样滤波器不需要和降抽样滤波器对称—例如: MPEG 滤波器可能被用来降抽样, 而 IDWT 滤波器可以用作升抽样。尤其是, 升抽样模块 (1430) 通过任一随机比率改变嵌入式编解码器 (1420) 的输出的空间分辨率, 以便恢复视频在降抽样模块 (1410) 之前具有的空间分辨率。

编码器 (1400) 从原始时间子带 $t-L_2$ 减去重建版本 $t-L_2'$ 。这产生一个差或者残余子带 $t-L_2''$ 。当基层编解码器 (1420) 是有高效的 (例如: VC-1 或 H.264/AVC) 以及基层比特率是足够的, 基层质量可以相对较好, 因此子带差没有很多能量。另一方面, 当基层编解码器 (1420) 是低效的或基层比特率是不充分的, 基层质量可能相对较差, 因此子带差具有更多能量。

编码器 (1400) 在子带差 $t-L_2''$ 基础上执行两个更深级别的时间分解。最后, 编码器 (1400) 在各自子带 (例如: $t-H_1, t-H_2, t-H_3, t-H_4$ 和 $t-L_4$) 执行空间转换和熵编码以生成可缩放比特流。

图 15 显示了根据第一种方案的相应的 3D 子带解码器 (1500), 解码器 (1500) 包括嵌入式基层解码器 (1520) 以及在基层解码器 (1520) 之后的升抽样模块 (1530)。

升抽样模块 (1530) 是可选的。它被用在当由嵌入式解码器 (1520) 解码的视频具有比输出到下流反向时间转换级别的时间子带 $t-L_2$ 低的空间分辨率时。否则, 抽样模块 (1530) 可以被跳过。

图 15 中的虚线框包括了与在两个附加级别的反向时间分解之前的和基层编解码器 (1520) 相关的操作。对于基层, 视频是每秒 $f/4$ 帧, 而空间分辨率可能已经减少了。因此, 基层视频提供了低帧率和 (潜在地) 低空间分辨率的视频的表示。实际上, 具有低帧率和 (潜在地) 低空间分辨率的视频可以直接被

5 基层编解码器 (1520) 解码并输出。

另外, 编解码器 (1500) 对来自编解码器 (1500) 将用于重建视频的可缩放比特流 (例如: $t-H_1$ 、 $t-H_2$ 、 $t-H_3$ 、 $t-H_4$ 和 $t-L_4$) 的任一个子带执行熵解码和反向空间分解。编解码器 (1500) 根据 $t-H_3$ 、 $t-H_4$ 和 $t-L_4$ 子带和相关的动作信息执行第三和第四级反向时间分解来重建子带差 $t-L_2$ ”。

10 嵌入式编解码器 (1520) 接收以基层分辨率编码的视频的基层比特流 (1595)。使用适当的解码操作, 嵌入式编解码器 (1520) 对在基层比特流 (1595) 中压缩的降抽样的子带 $ts-LL$ 进行解码, 以产生子带的一个重建版本 $ts-LL'$ 。

升抽样模块 (1530) 采用某种滤波机制对来自基层编解码器 (1520) 的重建子带 $ts-LL'$ 进行升抽样, 以产生时间子带 $t-L_2$ 的重建版本 $t-L_2'$ 。重建子带 $t-L_2'$ 与 $t-L_2$ 具有相同的分辨率。例如: 升抽样模块 (1530) 通过两个水平或垂直因子来增加空间分辨率。尤其是, 升抽样模块 (1530) 通过任一随机比率改变从嵌入式编解码器 (1520) 输出的空间分辨率来恢复视频在任何基层降抽样之前的空间分辨率。

15

编解码器 (1500) 将重建子带差 $t-L_2$ ”加到相应的重建子带 $t-L_2'$ 上。这样产生

20 时间低通子带 $t-L_2$ 的重建版本, 这个版本被输入到反向时间分解的剩余部分。

2. 第二种方案

图 16 显示了根据第二种方案的 3D 子带编解码器 (1600)。编解码器 (1600) 包括嵌入式基层编解码器 (1620) 以及在基层编解码器 (1620) 左右的 DWT 模块 (1610) 和 IDWT 模块 (1630)。

25 像在图 14 中显示的降抽样模块 (1410) 和升抽样模块 (1430) 一样, DWT 模块 (1610) 和 IDWT 模块 (1630) 是可选的。它们在由嵌入式编解码器 (1620) 进行编码的视频具有比输入到嵌入式编解码器 (1620) 的时间子带 $t-L_2$ 低的空间分辨率时被使用。否则, DWT 模块 (1610) 和 IDWT 模块 (1630) 可以被跳过。

30 在图 16 中虚线框包括了与两级时间分解之后的基层编解码器 (1620) 的相

关操作。DWT 模块 (1610) 处理 $t-L_2$ 时间子带, 以生成低通子带 $ts-LL$ 。一般地, DWT 模块 (1610) 改变嵌入式编解码器 (1620) 的输入的空间分辨率以便获得想要的基层空间分辨率和/或匹配被基层编解码器 (1620) 允许的输入空间分辨率。DWT 模块 (1610) 也生成路由到 IDWT 模块 (1630) 的空间高通子带

5 $ts-LH$ 、 $ts-HL$ 和 $ts-HH$ 。

嵌入式编解码器 (1620) 利用任何嵌入式编解码器 (1620) 的常用操作来对 $ts-LL$ 子带进行编码。嵌入式编解码器 (1620) 以基层分辨率生成输出的基层比特流 (1695)。嵌入式编解码器 (1620) 也利用嵌入式编解码器 (1620) 的相应解码操作来对 $ts-LL$ 子带进行解码, 以生成重建子带显示 $ts-LL'$ 。

10 解码器 (1600) 从原始 $ts-LL$ 子带减去其重建版本 $ts-LL'$ 。这产生一个残余子带 $ts-LL''$ 。和图 14 的编码器 (1400) 的情况相同, 当基层编码器 (1620) 是有效的和基层比特率是充分的, 残余 $ts-LL''$ 通常没有很多能量。另一方面, 当基层编解码器 (1620) 是低效率的或基层比特率是不充分的时, 残余 $ts-LL''$ 具有更多的能量。

15 从残余子带 $ts-LL''$ 和空间高通子带 $ts-LH$ 、 $ts-HL$ 和 $ts-HH$ 中, IDWT 模块 (1630) 生成新的残余子带 $t-L_2''$ 。新的残余子带 $t-L_2''$ 与被输入到 DWT 模块 (1610) 的子带 $t-L_2$ 具有相同的空间分辨率。

编码器 (1600) 在残余子带 $t-L_2''$ 上执行两个更深级别的时间分解。最后编码器 (1600) 在各自子带 (如: $t-H_1$ 、 $t-H_2$ 、 $t-H_3$ 、 $t-H_4$ 和 $t-L_4$) 执行空间转换和

20 熵编码以生成可缩放比特流。

图 17 显示了根据第二种方案的相应 3D 子带解码器 (1700)。解码器 (1700) 包括嵌入式基层解码器 (1720) 以及在基层解码器 (1720) 左右的 DWT 模块 (1710) 和 IDWT 模块 (1730)。

DWT 模块 (1710) 和 IDWT 模块 (1730) 是可选的。它们被用在当由嵌入式解码器 (1720) 进行编码的视频具有比输出到下流反向时间转换级别的时间子带 $t-L_2$ 的空间分辨率低时。否则, DWT 模块 (1710) 和 IDWT 模块 (1730) 可以被跳过。

25

图 17 中虚线框包括了在两个附加级别的反向时间分解之前和基层解码器 (1720) 相关的操作。基层给出了低帧率和 (潜在地) 低空间分辨率的视频的

30 表示, 并且这样较低空间分辨率的视频能被基层解码器 (1720) 直接解码并输

出。

另外，解码器（1700）对来自解码器（1700）将用于重建视频的可缩放比特流（例如： $t-H_1$ 、 $t-H_2$ 、 $t-H_3$ 、 $t-H_4$ 和 $t-L_4$ ）的任一个子带执行熵解码和反向空间分解。解码器（1700）根据 $t-H_3$ 、 $t-H_4$ 和 $t-L_4$ 子带和相关的动作信息执行第
5 三和第四级反向时间分解来重建残余子带 $t-L_2$ ”。

DWT 模块（1710）处理残余子带 $t-L_2$ ”，生成空间低通残余子带 $ts-LL$ ” 的重建版本以及空间高通子带 $ts-LH$ ”、 $ts-HL$ ”和 $ts-HH$ ”。通常，DWT 模块（1710）改空间分辨率以便低通残余子带 $ts-LL$ ”能在空间分辨率方面匹配嵌入式解码器（1720）的输出。

10 嵌入式解码器（1720）接收以基层分辨率进行编码的视频的基层比特流（1795）。使用适当的解码操作，嵌入式解码器（1720）可以对在基层比特流（1795）中压缩的转换子带 $ts-LL$ 进行解码，生成子带的重建版本 $ts-LL'$ 。

嵌入式解码器（1700）将重建残余子带 $ts-LL$ ”加到从嵌入式解码器（1720）输出的相应重建子带 $ts-LL'$ 。这样生成原始 $ts-LL$ 子带的重建版本。

15 IDWT 模块（1730）接着将从 DWT 模块（1710）中输出的空间高通子带 $ts-LH$ ”、 $ts-HL$ ”和 $ts-HH$ ”和重建 $ts-LL$ 子带合并。这样生成时间低通子带 $t-L_2$ 的重建版本，其被输入到反向时间分解的剩余部分。

3. 第三方案

在先前的两个编码方案中，基层编解码器被嵌在时间转换之中以及残余信号在后来的时间转换之中被进一步分解。与此相反，在第三个编码方案中，后
20 来的时间转换被用到时间低通帧中（与残余帧相反）。

图 18 显示了根据第三个方案的 3D 子带编码器（1800）。编码器（1800）包括嵌入式基层编解码器（1820）以及在基层编解码器（1820）左右的降抽样模块（1810）和升抽样模块（1830）。图 18 中的虚线框包括了在两级时间分解
25 之后的基层编解码器（1820）的相关操作。像图 14 中所示的编码器（1400）的相应模块一样，降抽样模块（1810）和升抽样模块（1830）是可选的。和在第二和第三时间转换之间的可缩放视频一样，基层视频是每秒 $f/4$ 帧，而分辨率可以由下抽样模块（1810）减少。因此，基层视频再次提供了低帧率和（潜在地）低空间分辨率的原始视频
30 的表示。

降抽样模块（1810）使用简单子抽样、低通滤波器或其它滤波机制对 $t-L_2$

时间子带降抽样。通常，降抽样模块（1810）根据任一随机比率改变嵌入式编解码器（1820）的输入的空间分辨率，来获得想要的基层空间分辨率和/或匹配被基层编解码器（1820）允许的输入空间分辨率。

5 嵌入式编解码器（1820）利用嵌入式编解码器（1820）的常用操作对降抽样 $t-L_2$ 子带（图示为 $ts-LL$ 子带）进行编码。嵌入式编解码器（1820）以基层分辨率生成最低比特率的输出基层比特流（1895），其中基层比特流（1895）和相应基层解码器是兼容的。为了编码器（1800）中后面编码的目的，嵌入式编解码器（1820）还利用嵌入式编解码器（1820）的相应解码操作来对 $ts-LL$ 子带解码，以生成重建子带显示 $ts-LL'$ 。

10 升抽样模块（1830）利用一些滤波器机制对基层编解码器（1820）的重建输出子带 $ts-LL'$ 进行升抽样，生成时间子带 $t-L_2$ 的重建版本 $t-L_2'$ 。重建子带 $t-L_2'$ 与 $t-L_2$ 具有相同的分辨率。通常，升抽样模块（1830）利用随机比率改变嵌入式编解码器（1820）的输出的空间分辨率来恢复视频在降抽样模块（1810）之前具有的空间分辨率。

15 编码器（1800）在原始（基层预处理）时间子带 $t-L_2$ 上执行第三和第四级别的时间转换。对于第三和第四级别的时间转换，编码器（1800）为时间高通子带 $t-H_3$ 和 $t-H_4$ 计算动作补偿预测。像期望的那样，时间子带 $t-L_2$ 为第三级别时间转换中的动作补偿提供预选参考帧（例如，偶数帧 $t-L_2$ ），时间子带 $t-L_3$ 为第四级别时间转换中的动作补偿提供预选参考帧（例如，偶数帧 $t-L_3$ ）。根据第
20 三方方案，在经过升抽样的重建子带 $t-L_2'$ 中与时间相应的图像，被作为动作补偿的附加的预选参考帧。例如，交替帧 $t-L_2'$ （具有和奇数索引帧 $t-L_2$ 相同的计时）被提供作为在第三级别时间转换中奇数索引帧 $t-L_2$ 的 MCP 的参考。每第四帧 $t-L_2'$ （具有和奇数索引帧 $t-L_3$ 相同的计时）被提供作为在第四级别时间转换中时间相应的奇数索引帧 $t-L_3$ 的 MCP 的参考。

25 图 20 显示接着嵌入式基层编码器的时间转换中的动作补偿的不同参考的组合情况。预选参考帧包括来自输入时间低通子带的帧（如偶数帧）。预选参考帧也包括来自基层升抽样的输出的时间相应（如相同时间）的重建帧。例如：
在图 20 的组合（1）中，高通子带帧 H 使用来自前面时间子带（相应于低通子带帧 L）的偶数帧的第一动作补偿预测，以及使用来自基层（与高通子带帧 H
30 在时间上协同定位）的升抽样的重建的第二动作补偿预测。相应地，高通子带

帧 H 仅被用于更新用于动作补偿的偶数帧（以便生成相应低通子带帧 L）。如果源自基层的参考被用在预测步骤，对于它相应的更新步骤是不需要的。

在一些实施例中，编码器在大块相连的基础上选择和变换用于动作补偿的参考帧。可替换地，编码器在其它基础（例如：每个当前奇数帧，每个动作矢
5 量和大块分区）上选择和变换用于动作补偿的参考帧。

回到图 18，在最后时间分解之后，基层输出 $t-L_2'$ 的帧率是最终低通子带 $t-L_4$ 的帧率的四倍。编码器（1800）从最终低通子带 $t-L_4$ 中减去基层输出 $t-L_2'$ 的时间相应的升抽样重建帧。这样生成残余子带 $t-L_4'$ 。最后，编码器（1800）在各自子带（例如： $t-H_1$ 、 $t-H_2$ 、 $t-H_3$ 、 $t-H_4$ 和 $t-L_4'$ ）上执行空间转换和熵编码以生成
10 可缩放比特流。

图 19 显示根据第三个方案的相应 3D 子带解码器（1900）。解码器（1900）包括嵌入式基层解码器（1920）和在基层解码器（1920）之后的升抽样模块（1930）。

像图 15 中所示的解码器（1500）一样，升抽样模块（1930）是可选的。它
15 被用在当由嵌入式解码器（1920）解码的视频具有比下游反向时间转换级别使用的时间子带帧少的空间分辨率时。另外，升抽样模块（1930）可以被跳过。

图 19 中的虚线框包括了与在四级反向时间分解之前的基层解码器（1920）的相关操作。像用其它基层解码器那样，具有低帧率和（潜在地）低空间分辨率的视频可以被基层解码器（1920）直接解码和输出。

另外，解码器（1900）对从解码器（1900）将用于重建视频的可缩放比特流（例如： $t-H_1$ 、 $t-H_2$ 、 $t-H_3$ 、 $t-H_4$ 和 $t-L_4'$ ）中的任一个子带执行熵解码和反向空
20 间分解。

嵌入式解码器（1920）接收用于以基层分辨率进行视频解码的基层比特流（1995）。使用适当的解码操作，嵌入式解码器（1920）可以对在基层比特流
25 （1995）中压缩的降抽样子带 $ts-LL$ 进行解码，生成子带的重建版本 $ts-LL'$ 。升抽样模块（1930）利用某种滤波机制对来自基层解码器（1920）的重建子带 $ts-LL'$ 进行升抽样，以生成时间子带 $t-L_2$ 的重建版本 $t-L_2'$ 。

解码器（1900）将残余子带 $t-L_4'$ 加到升抽样基层输出 $t-L_2'$ 的时间相应（相同时间）帧。这样生成时间低通子带 $t-L_4$ 的重建版本，其被输入到第四级反向
30 时间分解中。

编码器（1900）执行第三和第四级别反向时间分解以便根据重建的 $t-H_3$ 、 $t-H_4$ 和 $t-L_4$ 子带以及相关的动作信息来重建时间子带 $t-L_2$ 。对于第三和第四级别时间转换，解码器（1900）为时间高通子带 $t-H_3$ 和 $t-H_4$ 计算动作补偿的预测。时间子带 $t-L_3$ 具有用于第四级别时间转换中的 MCP 的重建参考帧（例如，重建偶数帧 $t-L_3$ ），时间子带 $t-L_2$ 具有为第三级别时间转换中的 MCP 的重建参考帧（例如，重建偶数帧 $t-L_2$ ）。在经过升抽样的重建子带 $t-L_2'$ 中的时间相应（例如，相同时间）图像，可以作为在第三和第四级别反向时间转换中的 MCP 的附加参考帧。

B. 基层信息的附加利用的范例

除了低比特率视频的比率失真检测中提供后向兼容以及提高压缩效率之外，基层的输出还有附加作用。

1. 基层动作矢量的再利用

在具有以提升为基础的 MCTF 的 3D 子带编码中，MCP 步骤的动作矢量是通过动作估计获得的。在一些实施例中，动作估计过程和传统的用于符合现有视频标准例如：MPEG2、H.264/AVC 等的编码器的动作估计是类似的。而且，Haar 和 5/3 时间转换分别对于常规 P-图像和 B-图像的动作补偿过程是类似的。

由于和现有标准例如 MPEG-2、MPEG-4、H.264/AVC 等兼容的基层编解码器，用内部图像（P-图像或 B-图像）内部压缩的块编码典型地具有相关的动作信息。同时，在 3D 子带编码中，在基层编解码器的各级时间分解下游（例如，图 14、16 和 18 中的第三和第四级别）的高通子带可以在基层输出中找到时间相应的图像。在许多情况下（特别是根据第三方案），在这种高通子带的动作信息和基层中与时间相应的内部图像的动作信息之间存在很高的相似性。基于这种相关性，编码器和/或解码器因为几个目的重新利用基层编解码器的动作矢量。

在基层动作矢量附加应用的一个例子中，3D 子带编码器和 3D 子带解码器仅仅将基层动作矢量作为时间相应高通子带帧的块的真实动作矢量使用。（当基层帧和高通子带帧具有不同的空间分辨率时，3D 子带编码器和 3D 子带解码器相应地能估算基层动作矢量。）尤其是，3D 子带编码器和 3D 子带解码器在低比特率下为这一目的使用基层动作矢量。这样减少了动作矢量的开销，当然，也由于基层动作矢量已经作为基层比特流的部分被发送。在许多情况下，在动作预测精度损失方面的补偿是轻微的。

在底层动作矢量附加应用的其它例子中，3D 子带编码器和 3D 子带解码器应用动作矢量预测中来自底层编解码器的动作矢量。作为常规动作矢量编码和解码的部分，动作矢量预测为当前动作矢量值提供预测。当前动作矢量接着可能用相对预测动作矢量的差表示并被传送。编码器和解码器都根据相邻块的
5 在前重建的动作矢量为当前动作矢量选择或导出预测动作矢量。

在一些实施例中，当为时间高通子带块的当前动作矢量选择或导出预测动作矢量时，3D 子带编码器和 3D 子带解码器根据时间相应基层帧的空间相应块使用动作矢量。例如：3D 子带编码器和 3D 子带解码器使用基层动作矢量作为预测动作矢量。或者，3D 子带编码器和 3D 子带解码器使用基层动作矢量作为
10 选择预测动作矢量的候选（例如：作为候选的中间值）。

在底层动作矢量的附加应用的另一个例子中，3D 子带编码器使用基层动作矢量加快动作估计。例如：子带编码器考虑早期动作估计中的相应基层动作矢量以便更快找到合适的当前动作矢量。

类似地，3D 子带编码器和 3D 子带解码器能够再次利用基层的动作模式（例
15 如，块分区、方向）信息。在一些例子中，基层模式信息可以被用作时间相应高通子带帧的块的真实动作信息。或者，基层动作模式被用于动作信息的预测或用来加快动作估计。

图 21 到 23 显示了不同的编码结构（帧类型模式），在这种情况下，基层动作信息对于基层之后的时间分解中的时间高通子带有着显著的作用。在图 21 中，
20 基层的编码结构被调整以适应普通 3D 子带编码结构。三个 B-图像被插在两个 I-或 P-图像之间，而第二张 B-图像（像用 H.264/AVC 存储的 B-图像）被用作第一张和第三张 B-图像的参考图像。通过这种方法，在基层中的 B-图像的动作矢量能够很自然的被再次用在 3D 子带编码中时间相应的高通子带帧。

为了与用于基层编解码器的某些现存标准兼容，图 22 的帧结构采用常规基
25 层编码结构但是调整了用于 3D 子带编码的帧类型的模式。在基层中 3 个 B-图像被插在两个 I/P-图像之间，但是 3 个 B-图像仅把 I/P-图像作为参考（没有 B-图像作为动作补偿参考）。在 3D 子带编码方案中，该模式包括在两个低通子带间产生的三个高通子带。

图 23 说明一种兼容模式。在基层中，在 3D 子带编码中时间上相应于高通
30 子带的图像被用 B-或 P-图像进行编码。反之，在图 21 和 22 中所示的模式适合

5/3 时间分解，图 23 所示的模式可以自适应地用于在 Haar 和 5/3 时间分解（分别相应于 P-图像、B-图像的关系）。

2. 预测来自基层的内部域

当视频序列存在封闭和不规则动作时，动作估计在时间高通子带中只能发现和一些块缺少匹配的块。当没有内部编码模式可用时，为这些情况的强制内部预测可以使得编码功能恶化并使视觉质量受到坏影响。

在 H.264/AVC 标准中，内部预测技术对 MCP 失败的情况提出警告。在内部预测中，当前块的预测值是由邻近重建像素的空间预测得到的。在具有开环 MCTF 结构的 3D 子带编码中，无论怎样，这样的空间预测趋向指出不匹配和误差。

在上文描述的第一和第二嵌入式基层编码方案中，在完成基层编码和解码后，对差信息进一步执行 3D 子带编码的时间分解。（例如：在图 14 中，差信息是输入时间子带 $t-L_2$ 和从基层编解码器输出 $ts-LL'$ 的升抽样版本 $t-L_2'$ 间的残余。）典型地，差只有高频能量以及它的空间冗余很低。然而，当一个块在基层编解码器中被以内部模式编码时，在 3D 子带编码后来的时间分解中内部预测通常是不必要提出的。这有助于避免与开环 3D 子带编码中内部预测有关的不匹配的问题，同时提高了编码性能和视觉质量。

另一方面，在上文描述的第三种嵌入式基层编码方案中，不对差信息执行 3D 子带编码的进一步时间分解。第三方案中，无论怎样，基层编解码器的输出可以提供预选参考图像，而预选参考图像可以包括使用空间内部预测编码的块。执行相对于来自于基层编解码器输出的参考图像的内部编码块的动作补偿可以有助于指出在 MCTF 中的 MCP 的情况，否则只能发现匹配不佳的块。来自基层编解码器输出的参考图像也可以被用在后面的 3D 子带编码中低通子带的预测值。3D 子带编码器可以利用来自基层的内部信息来确定何时为后来的 MCP 或其它预测使用来自基层的参考图像。

3. 为进一步时间分解的动作估计

在纯 3D 子带编码中，以及在具有根据上文描述的第三个方案的嵌入式基层编解码器的 3D 子带编码中，动作估计是在原始低通子带上完成的以便获得较高级别的高通子带和动作信息。这种动作估计有效地结合许多的常规动作估计参数，例如，对编码残余信息的消耗和编码动作信息的消耗之比加权的“ λ ”参

数。

根据上文描述的第一和第二嵌入式基层编解码方案中的具有嵌入式编解码器 3D 子带编码, 无论怎样, 为进一步时间分解执行差信息的动作估计。这样, 动作估计匹配常规动作估计参数以在进一步时间分解中利用相邻帧间的时间冗余。例如: 编码器协调 λ 参数来对输入到时间分解的差作出解释。

C. 使用基层编码器或解码器的技术的范例

图 24 显示具有嵌入式基层编解码器可缩放编码的技术 (2400)。图 25 显示具有嵌入式基层解码器的可缩放解码的技术 (2500)。诸如参考图 14、16 或 18 所描述的 3D 子带编码器执行在图 24 所示的技术 (2400)。诸如参考图 15、17 或 19 所描述的 3D 子带解码器执行在图 25 所示的技术 (2500)。可选择地, 另一种编码器或解码器执行技术 (2400, 2500)。

参考图 24, 3D 子带编码器利用子带编码对一层或多层视频进行编码 (2410), 这样产生的输入到基层编解码器。例如: 3D 子带编码器执行一层或多层 MCTF, 找到时间高通子带的动作信息并生成用作输入到基层编解码器中的低帧率视频的时间低通子带。可替换地, 3D 子带编码器在这一阶段执行其它和/或者附加类型的编码。

3D 子带编码器用基层编解码器对基层视频进行编码 (2430)。例如: 基层编解码器是一个 H.264/AVC、MPEG-2、VC-1 或 WMV9 编解码器并生成相应的压缩视频比特流和重建的输出视频。动作信息和内部模式信息是压缩视频比特流的一部分, 或这一信息可以以一些其它形式被独立地提供给 3D 子带编码器。

3D 子带编码器接着用基层编解码器的结果对一层或多层视频进行编码 (2450)。例如: 3D 子带编码器对根据基层编码或解码的结果计算出来的差信息执行一级或多级时间分解。或者, 3D 子带编码器利用来自基层编解码器的重建视频中的参考图像对时间低通子带执行一级或多级时间分解。或者, 3D 子带编码器将动作信息和/或内部模式信息用在编码或时间分解的判决中。可选地, 3D 子带编码器在随后的子带编码中以某种其它方法利用基层编解码器的结果, 或跳过随后的编码 (2450)。

编码器确定 (2490) 是否继续编码过程, 并且, 如果是, 继续下一帧。
(参考图 25, 3D 子带解码器用基层解码器对基层视频进行编码(2530))。例

如：基层解码器是 H.264/AVC、MPEG-2、VC-1 或 WMV9 解码器，它解析和解码相应的压缩视频比特流并生成重建输出视频。

3D 子带解码器确定 (2540) 解码分辨率。该分辨率可以根据使用者水平设置或某种其它设置来确定，可以根据解码器实际接收多少信息（例如，正好是
5 基层视频或者一个或多个附加层）来确定，或者用某种其它方法来确定。可替换地，3D 子带解码器可以只有一条解码路径。例如，一个解码器可以包括基层路径同时另一个解码器包括基层+可缩放层路径，以适应各自解码器的资源。

如果没有较高分辨率视频要解码（在判断 2540 之外的“无”路径），3D 子带解码器在阶段 (2590) 继续。

10 否则，3D 子带解码器用来自基层解码的结果对视频进行一个或多个附加层的解码 (2550)。例如：3D 子带解码器对差信息执行一级或多级反向时间分解然后把重建的差信息加到基层解码的结果中去。或者，3D 子带解码器利用来自基层解码器的重建视频的参考图像执行一级或多级反向时间分解来重建时间低通子带。可替换地，3D 子带解码器在随后的子带解码中以某种其它方式利用来自
15 基层解码器的结果。

最后，编码器确定 (2590) 是否继续解码过程，并且，如果是，则继续下一帧。

在图 24 和 25 中所示的各种操作是可以分开、和其它操作组合或被重新编排。

20 VI. 对 SDMCTF-编码的视频进行解码的空间缩放性

通常，根据 MCTF 在 3D 子带视频解码何时执行有两类 MCTF。在第一类中，被称作带内 MCTF (“IBMCTF”)，编码器首先对视频进行空间压缩然后在子带/子波域中执行 MCTF。编码器接着可以执行进一步空间分解。在第二类中，被称作空域 MCTF (“SDMCTF”)，编码器在空间分解之前，直接在空间域内
25 执行 MCTF。类似地，在解码期间，IMCTF 可以在空间域 (“SDIMCTF”) 或带内 (“IBMCTF”) 执行。

这样，在具有 SDMCTF 的 3D 子带编码中，整个视频在空域和随后的空间转换中被几次动作调整时间转换分解。图 5 显示了具有四级（反向）时间转换和两级（反向）空间转换的完整的编码和解码方案，而在解码中没有空间缩放
30 性。解码过程正好是图 5 中编码的反向。

通常，在特定比特率质量方面 IBMCTF 方案的性能没有 SDMCTF 方案的性能好。由于这个原因，许多 3D 子带编码器使用 SDMCTF。然而，在一些申请中，解码器生成视频输出的较低空间分辨率版本。例如：对于 CIF 分辨率（352 × 288 像素）的原始视频，对于具有一级空间缩放性的解码，3D 子带解码器生成解码视频的 QCIF 版本（176 × 144 像素）。这可以作为以下情况：解码器和较小屏幕设备（如：蜂窝式可视电话）相连或者解码器通过不可靠连接只接收较低空间分辨率子带信息。这些方案在视频的 MCTF 操作是在空域中被执行时是有问题的，因为不管怎样，动作信息、参考图像等是按照完整的空间分辨率表示的。

10 这部分描述了当对以低空间分辨率输出的 SDMCTF-编码视频进行解码时，各种提高解码性能的机制。在一些实施例中，例如：3D 子带视频解码器接收以某种分辨率（如 CIF）使用 SDMCTF 进行编码的视频但是对以更低分辨率（如 QCIF）输出的视频进行解码。3D 子带解码器使用几种不同机制中的任何一个来提高解码性能。

15 A. 对 SDMCTF 视频解码中的简单空间可缩放方案

图 26 说明从 SDMCTF 编码视频的压缩视频比特流解码和输出较低空间分辨率视频的两个简单方案（2600，2650）。低空间分辨率视频不同于图 26 中通过一级空间缩放的完整空间分辨率。

20 第一种方案（2600）提供了直接解决方案。3D 子带解码器首先恢复完整空间分辨率视频（所示的 CIF 视频）。（假定第二反向空间转换的空间高通（“SH”）子带丢失的情况下，解码器在第二反向空间转换中把 SH 子带作为空白（SH=0）对待。）在反向时间分解之后，3D 子带解码器用滤波器（所示的空间转换）对完整空间分辨率视频进行降抽样以达到到输出分辨率（所示的 QCIF）。解码器执行 SDIMCTF，并且当在中间步骤对 CIF 视频进行解码时，解码器使用许多解
25 码资源（例如，计算资源、参考图像缓冲器）。

30 另一方面，在第二种方案（2650）中，3D 子带解码器执行 IBIMCTF。第二种方案（2650）说明了一种对低分辨率视频进行解码的渐进的方法。像第一种方案（2600）一样，第二种方案（2650）包括熵解码和四级反向时间转换。第二种方案（2650）实际上包括了一种比第一种方案（2600）少的反向空间转换。（在第二种方案（2650）中所示的第二反向空间转换和随后的空间转换彼此

抵消并且可以被省略。)

在第二种方案(2650)中, 3D子带解码器在以SL子带分辨率执行反向时间转换过程中, 必须把空间低通(“SL”)子带(第一反向空间转换的输出)作为原始视频的全部对待。这近似没有空间缩放性的解码过程, 并具有低计算
5 复杂性和低缓冲器要求(对于参考图像, 等等)的优点。更进一步说, 视频原始分辨率和空间缩放性对于解码器来说可以是完全透明的。第二种方案(2650)的性能在低比特率方面是令人满意的但在高比特率方面不能令人满意。尤其是, 随着比特率的增加它对于重建高质量低空间分辨率视频变得困难和不可能, 即使比特率非常高。

10 图26所示的两个解码方案(2600, 2650)间的不同在于在IMCTF期间MCP和MCU的步骤如何被执行。图27说明了在一级IMCTF中两种方案(2600, 2650)的MCP。尤其是, 图27显示了根据参考图像中的可用SL子带信息(而不是SH子带信息)在预测步骤的SL提升信号的产生。

对于SDIMCTF, SL提升信号按如下所示产生:

$$15 \quad \xrightarrow{\text{SL}} \text{IDWT} \xrightarrow{\text{SD}} \text{INTP}_{\text{SD}} \rightarrow \text{MCP}_{\text{SD}} \xrightarrow{\text{SD}} \text{DWT} \quad (9),$$

这里 INTP_{SD} 表示参考图像信息在空间域中像素的插值, 而 MCP_{SD} 表示在空间域的MCP。DWT是线性操作, 并且:

$$\text{DWT}(F_1+F_2)=\text{DWT}(F_1)+\text{DWT}(F_2) \quad (10).$$

为使用SL缓冲器来保留中间结果, DWT对于在SL域中的预测减少了空
20 间域预测。

对于IBIMCTF, SL提升信号按如下所示产生:

$$\xrightarrow{\text{SL}} \text{INTP}_{\text{SL}} \rightarrow \text{MCP}_{\text{SL}} \xrightarrow{\text{SL}} \quad (11),$$

这里 INTP_{SL} 表示参考图像信息在SL域中像素的插值, 而 MCP_{SL} 表示在SL域的MCP。为了简便, 同样的插值滤波器被用于 INTP_{SD} 和 INTP_{SL} 。

25 B. 理论分析

各种因素导致在对SDMCTF编码视频进行3D子带解码过程中的空间缩放性出现问题。一个因素是SL域中插入值和动作补偿的无效性。另一因素是IMCTF的位置(与在编码中的MCTF的位置相比)。还有另外一个因素由动作补偿引起的跨越子带的能量谱泄漏。

30 除了这些因素外, 各种编码/解码特征能导致编码器中SDMCTF和解码器

中 IBIMCTF 之间的不匹配。两个这样的特征是动作矢量缩放比例和 OBMC。在这部分中，为了简便，假设在 IBIMCTF 中动作矢量的精度是不变的，并且在 IBIMCTF 期间子像素插入值被用作动作补偿是足够好的以便能充分利用动作信息。也是为了简便的目的，OBMC 是无作用的，MCP 操作假定是只根据可适用
5 动作矢量的简单像素取出操作。甚至作为一些简化假设，在前面段落中列举的因素可以降低 IBIMCTF 的功能。

1. 插入值

甚至当同样的插值滤波器（像双线性的滤波器或 8 抽头正弦滤波器）被用于空域插值和 SL 域插值时，给定的一帧中相同位置的局部像素是从不同批像素
10 中被插入的，这是由于两个操作的不同范围。这导致由被插入子像素的参考图像间的不匹配。

由于另一个原因在 SL 域的插入可能不是最佳的。由于子波转换的抽取方面，只有空间 LL 带不是原始视频的整个低通信号的完全代表。也就是说，不考虑高通信号或当高通信号缺失时，在二分之一相位位置部分低通信号丢失。抽取的一个结果是空间 LL 带偏移变量性质。由于 LL 带的不完整性，当获得插值
15 低通帧时，在 LL 带（如在操作（12）显示）中进行直接插入可能不是最佳的。

$$\xrightarrow{SL} \text{INTP}_{SL} \xrightarrow{mxm SL} \text{MCP}_{SL} \xrightarrow{SL} \quad (12),$$

$$\xrightarrow{SL} \text{IDWT} \rightarrow \text{OCDWT} \rightarrow \text{INTP}_{OC-LL} \xrightarrow{mxm SL} \text{MCP}_{SL} \xrightarrow{SL} \quad (13), \text{ 和}$$

$$\xrightarrow{SL} \text{IDWT} \xrightarrow{SD} \text{INTP}_{SD} \xrightarrow{mxm SD} \text{DSDWT} \xrightarrow{mxm SL} \text{MCP}_{SL} \xrightarrow{SL} \quad (14).$$

一种可供选择的方法是通过变换过完整子波表示执行二分之一像素插入。来自 SL 的过完整表示可以由完全到过完整的 DWT(未示出)或由反向 DWT 和过
20 完整的 DWT(“OCDWT”)（在操作 13 中显示)产生。这种 IDWT 加上 OCDWT 的和作为一级插入。余下级别的插入可以用过完整子波域的常规插入完成，用操作 13 中的 INTP_{OC-LL} 表示。

变换到常规过完整子波表示只要提供在 SL 域的二分之一像素插入。为了支持在 SL 域中四分之一像素动作矢量精度或更出色动作，解码器可以利用连续
25 的相位过完整转换(“CPOCWT”)。为了在 SL 子带中四分之一像素位置，解码器在空间域中执行插入，如操作(14)所示。接着，插入帧的低通子带通过 DWT 获得。DWT 是比例敏感操作。只要在编码器中与 DWT 匹配，编码器在插入帧中以 m-抽样像素执行 OCDWT，这里 mxm 是在 OCDWT 之前空间域插入的因

子。在操作 (14) 中, DSDWT 表示对将空间域插入帧降抽样为一些子帧, 在各自子帧执行 OCDWT 以及向后地隔行扫描获得的系数。象这样, DSDWT 是一种 OCDWT。

根据 (12) 至 (14) 的每一操作, 解码器在 SL 域中根据 mxm 参考图像执行动作补偿预测。

2. DWT 和 MCP 的位置变换

对于 IBIMCTF, 即使在像 (14) 中带内插入被用空域插入代替的情况下, MCP 的参考图像仍是低通帧—MCP 发生在子波域中。与编码器相比较, DWT 和 MCP 的预期排序被交换 (这里, 在编码器中, MCP 先于 DWT)。然而, 由于 DWT 的偏移变量属性, DWT 和 MCP 操作是不可交换的:

$$DWT(MCP_{SD}(F_{SD})) \neq MCP_{SL}(DWT(F_{SD})) \quad (15) \text{和}$$

$$IDWT(MCP_{SL}(F_{SL})) \neq MCP_{SD}(IDWT(F_{SL})) \quad (16)。$$

假设为了简化目的 MCP 包括简单的以块为基础的动作变换操作, 等式 (15) 按如下解释, 假设当前帧被分解成块 $B = \{B_i | i=1, \dots, m\}$, 并且这些块的动作矢量 $MV = \{mv_i | i=1, \dots, m\}$ 。 F_{ref}^i 是参考帧 F_{ref} 的新版本, 这里只有作为块 B_i 参考的像素被保留, 其它像素被定义为零。

$$F_{pred}(x) = \sum_{i=1}^m F_{ref}^i(x + mv_i) \quad (17),$$

$$\begin{aligned} DWT(MCP(F_{ref})) &= DWT\left(\sum_{i=1}^m F_{ref}^i(x + mv_i)\right) \\ &= \sum_{i=1}^m DWT(F_{ref}^i(x + mv_i)) \neq \sum_{i=1}^m (DWT F_{ref}^i(x + mv_i)) \end{aligned} \quad (18)。$$

$$\begin{aligned} &= MCP\left(\sum_{i=1}^m DWT(F_{ref}^i)\right) \neq MCP\left(DWT\left(\sum_{i=1}^m F_{ref}^i\right)\right) \\ &\neq MCP(DWT(F_{ref})) \end{aligned}$$

在 (18) 中的第一个不等式是由于 DWT 的偏移变量属性。在 (18) 中的第二个不等式是由于当动作非常复杂时 MCP 中动作变换期间的重叠和泄漏。

一种可能的解决方案是将 MCP 移到 DWT 之前的空间域, 像操作 (19) 中所示的那样。这种情况下, MCP 是 SDIMCTF 的一部分。

$$\xrightarrow{SL} IDWT \xrightarrow{SD} INTP_{SD} \rightarrow MCP_{SD} \xrightarrow{SD} DWT \quad (19).$$

3. 动作移动导致的功率谱泄漏

在 3D 子波编码中, 时间空间子带通常假定是独立的。在许多申请中, 时

间空间子带根据希望被丢弃，特别是高通子带。然而，在使用 MCTF 的 3D 子带编码中，由于动作移动引入的功率谱泄漏，相邻帧的低通子带和高通子带具有类似之处。

当只包含一个空间子带信号的帧根据动作变换时，部分该信号被转移到其它的空间子带。图 28 说明简单球形动作现象。在图 28 的第一行中，原始帧 Org 被分成两部分：SL 信号 A 和 SH 信号 B。在第二行中，只包括 SL 信号 A 的帧被用简单图像域动作变换（“IMS”）变换到空间域中，而变换后的帧现在包括 SH 信号。相似地，在图 28 的第三行中，只包括 SH 信号 B 的帧被切换到空间域中，而变换后的帧包括 SL 信号。当运动复杂时频谱泄漏问题变得更为严重。

因此，在编码器中具有 SDMCTF 而不是使用 SL 信息的 IMCTF，参考帧的 SL 部分预测该帧的 SH 部分。因此，即使在解码器在开始阶段只有 SL 子带信息的情况下，解码器也能在每级 IMCTF 的中间效果中获得有意义的 SH 子带信息。无论如何，当解码器使用 SL 分辨率缓冲器来保留各级 IMCTF 之间的中间结果时，会丢弃 SH 子带信息，也许这对于下一级 IMCTF 中的 MCP 是有用的。

此外，参考帧的 SH 部分预测参考帧的 SL 部分。为了对 SL 编码，编码器可以在参考帧中使用 SH 子带信息。当解码器不能访问这些 SH 部分时，SH 子带的缺少导致一种漂移。（例如：当编码器根据完整分辨率帧(包括 SH 信息)执行 MCP，而解码器仅有 SL 信息。）这种结果是 PSNR 的极限—在相对低的比特率下 PSNR 曲线变成水平的。

C. 对具有空间缩放性的 SDMCTF-编码的视频进行解码的改进方案范例

在一些实施例中，3D 子带解码器当对 SDMCTF-编码的视频进行解码并以较低空间分辨率输出时，使用几项技术中的任一种来提高解码性能。这一技术一步步地提高解码性能，并且主要地不需要改变编码器以及发送的比特流。在这一集合中，这些技术以不同复杂级别的不同选项提供复杂的解码方案。

在一些方案中，解码器分配额外的资源用于解码。通常，当额外的计算资源致力于解码时解码器能重建更好的视频。而且，当额外缓冲器资源致力于存储参考图像信息时解码器重建更好的视频。

或者，编码器或解码器因为解码 SL 视频强调 SH 子带信息。例如：为将 SH 子带信息合并到 IMCTF 判决中，即使在解码器只重建 SL 分辨率视频时，

比特流抽取器给 SH 系数分配带宽，而不是完全丢弃它们。

这部分的各个图（也就是，图 29 到 34）显示了一级空间缩放。可替换地，3D 子带解码器以多级空间缩放输出视频。在相同的列中，这部分的各个图显示了为用 MCTF 以 CIF 分辨率编码的原始视频输出 QCIF 视频的 3D 子带解码器。

5 这里描述的技术和工具不限于用 CIF 和 QCIF 分辨率。

对这部分的许多图来说，动作矢量具有原始分辨率二分之一的像素精度和 SL 分辨率四分之一的像素精度。这影响了象子像素插值这样的操作。可替换地，动作矢量具有其它精度，相应地解码器执行更多或更少的子像素插值。而且，通过这部分的几个图中为了插值显示了用于插值的正弦滤波器，可替换地，解
10 码器使用其它类型的滤波器（如，双线性的，双三次的）。

最后，这部分的各个图显示了二元 DWT 和 IDWT 的操作。可替换地，解码器为空间转换和反向空间转换使用其它的操作。

1. 方案 A: IBIMCTF(基线)

为了与后面的方案比较，图 29 说明在解码期间没有额外的计算和缓冲器资源
15 源的 IBIMCTF。图 29 中的方案相应于图 26 的 IBIMCTF 解码器（2650）中的一级 IMCTF。在图 29 所示的操作存在于子波域中。

根据图 29，3D 子带解码器使用 SL 缓冲器（2910）存储 SL 子带信息作为参考图像信息。SL 子带信息的有效分辨率是 QCIF。

解码器采用 4×4 正弦滤波器执行子像素插值（2920）。（在 SL 域中 MCP 的动作矢量精度是四分之一像素的精度。）这生成子像素插值参考帧（2930）。
20 插值参考帧的有效分辨率是 4CIF（ 704×576 像素）。

解码器在 SL 域中执行 MCP（2940），使用参考图像信息的插值参考帧（2930）。 MCP_{SL} （2940）在 SL 域中生成预测值（2950）。预测值（2950）的有效分辨率也是 QCIF。

25 2. 方案 B: 最优化 IBIMCTF

在第二种方案中，3D 子带解码器通过将插值操作和/或 MCP 操作移动到空间域中来提高 IMCTF。解码器依然使用用于参考图像信息的 SL 分辨率缓冲器。图 30 显示了第二种方案的概括框架，其中反向时间转换被用最优化反向时间转换 OT^1 表示。图 31、32 和 33 显示根据第二种方案的不同变形的一级 IMCTF 的范
30 例。

a. 方案 B1

根据图 31 显示的方案（方案 B1），3D 子带解码器使用 SL 缓冲器（3110）来存储作为参考图像信息使用的 SL 子带信息。解码器利用缓存的 SL 子带信息执行 IDWT 操作（3120）。SH 子带在 IDWT 操作（3120）时被认为是空的（SH=0）。
5 这样 IDWT 操作（3120）根据 SL 子带信息生成参考帧（3130）。当 SL 子带的有效分辨率是 QCIF 时，参考帧（3130）的有效分辨率是 CIF。

解码器接着对参考帧（3130）执行 OCDWT（3140）操作，生成过完整表示（3150）。可替换地，解码器执行完全到过完整的 DWT 而不是 IDWT（3120）和 OCDWT（3140）。任一种方式，过完整表示（3150）包括适合二分之一像素
10 MCP 的相移值。

在给四分之一像素 MCP 预备中，解码器接着利用 2×2 正弦滤波器执行子像素插值（3160）。这生成子像素插值参考帧（3170）。该插值参考帧（3170）的有效分辨率是 4CIF。

解码器利用参考图像信息的插值参考帧（3170）在 SL 域中执行 MCP
15 （3180）。 MCP_{SL} （3180）在 SL 域中生成预测值（3190）。预测值（3190）的有效分辨率也是 QCIF。

b. 方案 B2

根据图 32 所示的方案（方案 B2），3D 子带解码器使用 SL 缓冲器（3210）来存储用作参考图像信息的 SL 子带信息。解码器利用缓存的 SL 子带信息来执行 IDWT 操作（3220）。SH 子带在 IDWT 操作（3220）中被认为是空的（SH=0）。
20 这样 IDWT 操作（3220）根据 SL 子带信息生成参考帧（3230）。当 SL 子带的有效分辨率是 QCIF 时，参考帧（3230）的有效分辨率是 CIF。

解码器利用 2×2 正弦滤波器执行子像素插值（3240）。这产生了子像素插值参考帧（3250），其具有由两个水平和垂直因子增加的分辨率。插值参考帧
25 （3250）的有效分辨率是 4CIF。

解码器接着在插值参考帧（3250）上执行 DSDWT（3260）操作。这样生成参考帧信息的过完整表示（3270）。过完整表示（3270）包括适合四分之一像素 MCP 的相移值。

解码器利用参考图像信息的插值参考帧（3270），在 SL 域中执行 MCP
30 （3280）。 MCP_{SL} （3280）在 SL 域中生成预测值（3290）。预测值（3290）的有

效分辨率也是 QCIF。

c. 方案 B3

根据图 33 所示的方案（方案 B3），3D 子带解码器使用 SL 缓冲器（3310）来存储用作参考图像信息的 SL 子带信息。解码器利用 SL 子带信息执行 IDWT 操作（3320）。SH 子带在 IDWT 操作（3320）中被认为是空的（SH=0）。这样 IDWT 操作（3320）根据 SL 子带信息生成参考帧（3330）。当 SL 子带的有效分辨率是 QCIF 时，参考帧(3330)的有效分辨率是 CIF。

解码器利用 2×2 正弦滤波器执行子像素插值（3340）。这产生了子像素插值参考帧（3350），其具有由两个水平和垂直因子增加的分辨率。插值参考帧（3350）的有效分辨率是 4CIF。

解码器利用参考图像信息的插值参考帧（3350），在空间域中执行 MCP（3360）。 MCP_{SD} （3360）（具有二分之一像素精度的动作矢量）在空间域中生成预测值（3370）。在这种例子中，预测值（3370）的有效分辨率是 CIF。

解码器根据预测（3370）执行 DWT 操作（3380），生成预测（3370）的低空间分辨率版本（3390）。低空间分辨率预测（3390）的有效分辨率也是 QCIF。

3. 方案 C: SDIMCTF

图 34 说明具有专用于解码期间的附加计算和缓冲资源的 SDIMCTF。图 34 中的方案相应于图 26 的 SDIMCTF 解码器（2600）中的一级 IMCTF。图 34 中的操作存在于空域中。与方案 B3 相比，在方案 C 中参考图像缓冲器是 SD 缓冲器，解码器不执行每一级 IMCTF 的 IDWT 和 DWT 操作。以这种方式，中间结果中的 SH 信息被保存用作后来的 SDIMCTF 中。

根据方案 C，3D 子带解码器使用 SD 缓冲器（3410）存储空间域信息并用作参考图像信息。对于第一个反向时间转换，最后的反向空间转换从 SL 子带信息产生 SD 缓冲器（3410）中的信息。对于随后的反向时间转换，从较早的反向时间转换中 SD 缓冲器（3410）的信息被保存，并且 SD 缓冲器（3410）的信息可以包括由于功率谱泄漏产生的 SH 信息。SD 缓冲器（3410）中信息的有效分辨率是 CIF。

解码器利用 2×2 正弦(sinc)滤波器执行子像素插值（3420）。这产生了子像素插值参考帧（3430），其具有由两个水平和垂直因子增加的分辨率。插值参考帧（3430）的有效分辨率是 4CIF。

解码器利用参考图像信息的插值参考帧 (3430)，在空间域中执行 MCP (3440)。MCP_{SD} (3440) (具有二分之一像素精度的动作矢量) 在空间域中生成预测值 (3450)。在这种例子中，预测值 (3450) 的有效分辨率是 CIF。

4. 方案 D: 具有 SH 系数的 SDIMCTF

5 为了处理 SH 漂移问题，带宽被分配给比特流抽取器中的 SH 系数。这不像常规空间缩放性，按照常规方法当为输出 SL 分辨率视频而解码时 SH 子带信息被简单地丢弃。

为了确定 SH 系数的最佳比率，测量 SH 系数减少低空间分辨率视频的失真的作用。(相反，按照常规空间缩放性的实践，测量 SH 系数减少全部原始分辨率视频的失真的作用) 换句话说，即使在只有低空间分辨率视频输出的时候，也考虑 SH 子带的增益因子以便考虑 SH 子带在减少 SL 视频失真方面的优势(例如，提高的动作补偿预测精度)。以这种方式，SH 子带信息很可能被向前带进 SL 视频解码中。

10 在方案 D 中，当解码 SL 视频时使用 SH 子带的当前增益因子，仿佛解码器正提取原始分辨率视频一样。SL 和 SH 的系数被允许包括在最后的比特流中。可替换地，可以为用于原始空间分辨率解码和 SL 解码的 SH 子带计算不同增益因子。

在方案 D 中，当解码 SL 视频时，3D 子带解码器为一级 IMCTF 采用图 34 所示的解码框架。参考图像缓冲器 (3410) 是 SD 缓冲器。对于第一反向时间转换，SD 缓冲器包括用 SL 子带信息和 (至少一些) SH 子带信息重建的参考图像信息，即使最后输出视频只会具有有效的 SL 分辨率。中间结果中的 SH 信息也被保存用在后来的 SDIMCTF 中。

20 解码器利用 2×2 正弦滤波器执行子像素插值 (3420)。这产生了子像素插值参考帧 (3430)，其具有由两个水平和垂直因子增加的分辨率。解码器利用参考图像信息的插值参考帧 (3430)，在空间域中执行 MCP (3440)。MCP_{SD} (3440) (具有二分之一像素精度的动作矢量) 在空间域中生成预测值 (3450)。

5. 比较

下表比较了方案 A、B1、B2、B3、C 和 D 的特征。

方案	只有 SL 系数	只有 SL 缓存	带内 INTP	带内 MCP
A	Yes	Yes	Yes	Yes
B1	Yes	Yes	Yes	Yes
B2	Yes	Yes	No	Yes
B3	Yes	Yes	No	No
C	Yes	No	No	No
D	No	No	No	No

可替换地，当为以较低空间分辨率输出对 SDMCTF 编码视频解码时，解码器使用另外一些空间域插值、空间域动作补偿、空域参考图像缓冲器的组合和/或使用 SH 信息来提高解码性能。

VI. 连续相位的过完整子带转换

IBMCTF 和 IBIMCTF 的一个问题是怎样减少子波转换偏移变量在动作估计和补偿方面的影响—在图像域中的偏移/动作不能直接映射到转换域中。一些技术将整数像素移动映射到转换域中。（可参见，例如，Y.A. Andreopoulos 等作者的“A New Method for Complete-to-Overcomplete Discrete Wavelet Transforms”，Proc. IEEE Conf. Digital Signal Proc. (2002)；以及，X.Li、L. Kerofsky 和 S.Lei 的“All-phase Motion Compensated Prediction for High Performance Video Coding”，Proc. ICIP, vol. 3, pp.538-541 (2001)）这部分描述了将分像素移动映射到转换域中的相位变化的技术。通过这样处理，动作估计和补偿能够完成得更精确，也能明显影响编码性能。

这部分用一级一维子波转换的例子说明了连续相位过完整子带转换 (“CPOST”)。p-像素 CPOST 的分解过程有四个连续阶段：(1)p-像素插值，(2)连续移位，(3)降抽样，和(4)子波转换。

把连续 p-像素相位过完整转换认为是 ($p=1, 1/2, 1/4, 1/8, \dots$)。当 $p=1$ 时，CPOST 是整数像素相位过完整的例子，其它 p 值表示子像素相位过完整的例子。假设输入信号是离散时域内的序列 $\{x(n)\}$ ， $n=0, 1, 2, 3, \dots$ 。编码器或解码器首先执行 $\{x(n)\}$ 的 p-像素插值，然后继续位移 p-像素插值序列 $\frac{2}{p} - 1$ 次并对位移的序列

降抽样。这样生成一组 $\{x(n)\}$ 的连续相移序列，用 $\{x_p^i(n)\}$ ， $i=0,1,\dots,\frac{2}{p}-1$ 表示。最后，编码器或解码器对 $\{x_p^i(n)\}$ 执行子波分解转换。

对于 $p=2$ 的例子，连续相移序列按如下所示。 $\{y(n)\}$ 是由 $\{x(n)\}$ 的 $\frac{1}{2}$ -像素插值点组成的序列。

$$\begin{array}{l}
 5 \quad \dots x(n) \quad y(n) \quad x(n+1) \quad y(n+1) \quad x(n+2) \quad y(n+2) \quad x(n+3) \quad y(n+3) \dots \\
 \quad 0\text{-相位} \quad \dots x(n) \quad x(n+1) \quad x(n+2) \dots \\
 \quad \frac{1}{2}\text{-相位} \quad \dots y(n) \quad y(n+1) \quad y(n+2) \dots \\
 \quad 1\text{-相位} \quad \dots x(n+1) \quad x(n+2) \quad x(n+3) \dots \\
 \quad \frac{3}{2}\text{-相位} \quad \dots y(n+1) \quad y(n+2) \quad y(n+3) \dots
 \end{array}$$

10 2D CPOST 能通过分别在水平和垂直方向上执行两个序列的 1D CPOST 实现。

在子波域中，连续 p -像素相移序列可以提供使用一组预测滤波器 $F_p^i(z)$ $i=0, 1, \dots, \frac{2}{p}-1$ 。这些滤波器等价于连续位移子像素插值序列和降抽样这些位移的序列。 p -像素 CPOST 的分解过程可以表示为 $\{x(n)\}$ 的滤波器应用，接着为子波

15 分解转换目的分别用低通滤波 $G_L(z)$ 和高通滤波器 $G_H(z)$ 处理结果，然后降抽样。 $(i=0,1,\dots,\frac{2}{p}-1)$ 的 $L_p^i(n)$ 和 $h_p^i(n)$ 分别是 CPOST 的低通和高通系数。在 z 域中， $X(z)$ 和 $X_p^i(z)$ 分别是 $\{x(n)\}$ 和 $\{x_p^i(n)\}$ 的 z 转换。 $L_p^i(z)$ 和 $H_p^i(z)$ 分别是 $L_p^i(n)$ 和 $h_p^i(n)$ 的 z 转换。

$$X_p^i(z) = F_p^i(z)X(z) \quad i=0,1,2,\dots,\frac{2}{p}-1 \quad (20),$$

$$\begin{array}{l}
 20 \quad L_p^i(z) = \frac{1}{2} [G_L(z^{\frac{1}{2}})X_p^i(z^{\frac{1}{2}}) + G_L(-z^{\frac{1}{2}})X_p^i(-z^{\frac{1}{2}})] \\
 \quad H_p^i(z) = \frac{1}{2} [G_H(z^{\frac{1}{2}})X_p^i(z^{\frac{1}{2}}) + G_H(-z^{\frac{1}{2}})X_p^i(-z^{\frac{1}{2}})] \quad (21). \\
 \quad i=0,1,2,3,\dots,\frac{2}{p}-1
 \end{array}$$

p -像素 CPOST 分解过程符合矩阵表达式：

$$\begin{bmatrix} L_p^0(z) \\ H_p^0(z) \end{bmatrix} = \frac{1}{2} \begin{bmatrix} G_L(z^{\frac{1}{2}})G_L(-z^{\frac{1}{2}}) \\ G_H(z^{\frac{1}{2}})G_H(-z^{\frac{1}{2}}) \end{bmatrix} \begin{bmatrix} X(z^{\frac{1}{2}}) \\ X(-z^{\frac{1}{2}}) \end{bmatrix} \quad (22),$$

$$\begin{array}{l}
 25 \quad \begin{bmatrix} L_p^i(z) \\ H_p^i(z) \end{bmatrix} = \frac{1}{2} \begin{bmatrix} G_L(z^{\frac{1}{2}})G_L(-z^{\frac{1}{2}}) \\ G_H(z^{\frac{1}{2}})G_H(-z^{\frac{1}{2}}) \end{bmatrix} \begin{bmatrix} X_p^i(z^{\frac{1}{2}}) \\ X_p^i(-z^{\frac{1}{2}}) \end{bmatrix} \quad (23). \\
 \quad i=1,2,3,\dots,\frac{2}{p}-1
 \end{array}$$

表达式 (22) 表示没有位移的原始序列子波分解转换。表达式 (23) 表示连续子相位移序列的子波分解转换。表达式 (23) 和表达式 (24) 组成 CPOST。对于 $L_p^i(z)$ ， $H_p^i(z)$ 和 $L_p^0(z)$ 和 $H_p^0(z)$ 的关系，从 (20) 到 (23)：

$$\begin{bmatrix} L'_p(z) \\ H'_p(z) \end{bmatrix} = \frac{1}{2} \begin{bmatrix} G_L(z^{1/2})G_L(-z^{1/2}) \\ G_H(z^{1/2})G_H(-z^{1/2}) \end{bmatrix} \begin{bmatrix} F'_p(z^{1/2}) & 0 \\ 0 & F'_p(-z^{1/2}) \end{bmatrix} \begin{bmatrix} X(z^{1/2}) \\ X(-z^{1/2}) \end{bmatrix} \quad (24),$$

从 (22) 到 (24):

$$\begin{bmatrix} L'_p(z) \\ H'_p(z) \end{bmatrix} = G(z) \begin{bmatrix} F'_p(z^{1/2}) & 0 \\ 0 & F'_p(-z^{1/2}) \end{bmatrix} G^I(z) \begin{bmatrix} L^0(z) \\ H^0(z) \end{bmatrix} \quad (25),$$

其中,

$$5 \quad G(z) = \begin{bmatrix} G_L(z^{1/2})G_L(-z^{1/2}) \\ G_H(z^{1/2})G_H(-z^{1/2}) \end{bmatrix}$$

并且其中 $G^I(z)$ 是 $G(z)$ 的逆矩阵。

根据 (25), 变换矩阵 $T'_p(z)$ 和逆变换矩阵 $IT'_p(z)$ 能从原始序列 (例如 0-相位移动序列) 的子波变换获得以便得到子像素相移序列的子波转换。

$$T'_p(z) = G(z) \begin{bmatrix} F'_p(z^{1/2}) & 0 \\ 0 & F'_p(-z^{1/2}) \end{bmatrix} G^I(z) \quad (26),$$

$$10 \quad IT'_p(z) = [T'_p(z)]^{-1} = G^I(z) \begin{bmatrix} F'_p(z^{1/2}) & 0 \\ 0 & F'_p(-z^{1/2}) \end{bmatrix}^{-1} G(z) \quad (27)。$$

根据 (26), 如果存在表达式 (28), 在离散时域中过完整子波转换之后的子像素插值的结果和子像素插值之后的过完整子波转换的结果是相等的。然而, 在离散时域中过完整子波转换之后子像素的结果和子像素插值之后的过完整子波转换的结果是不相等的。

$$15 \quad \begin{bmatrix} F'_p(z^{1/2}) & 0 \\ 0 & F'_p(-z^{1/2}) \end{bmatrix} G^I(z) = G^I(z) \begin{bmatrix} F'_p(z^{1/2}) & 0 \\ 0 & F'_p(-z^{1/2}) \end{bmatrix} \quad (28)。$$

通常, 表达式 (28) 不存在, 这就意味不能交换带内视频编码的子像素插值和过完整子波转换的顺序。然而, 根据 (25), 原始序列的 CPOST 系数可以从它的子波转换系数中获得。

CPOST 方法不仅能实现整数像素相位过完整子波转换而且也能实现子像素相位过完整子波转换。对于 $p=1/2$, 具有 5-3 子波转换和 6-抽头插值滤波器的连续相位过完整子波转换离散系数随后产生。转换矩阵 $T_{1/2}^1(z)$ 、 $T_{1/2}^2(z)$ 和 $T_{1/2}^3(z)$ 每一部分的系数按如下所示:

$$\begin{aligned} T_{1/2}^1(0,0) &= [0.0098 \ -0.0957 \ 0.8867 \ 0.2227 \ -0.0215 \ -0.0020] \\ T_{1/2}^1(0,1) &= [-0.0024 \ 0.0488 \ -0.2290 \ 0.3242 \ -0.1440 \ 0.0020 \ 0.0005] \\ 25 \quad T_{1/2}^1(1,0) &= [0.0391 \ -0.2344 \ 0.3438 \ -0.1406 \ -0.0078] \end{aligned}$$

$$T_{\frac{1}{2}}^1(1,1)=[-0.0098 \ 0.1582 \ 0.3633 \ -0.5352 \ 0.0215 \ 0.0020]$$

$$T_{\frac{1}{2}}^2(0,0)=[-0.0625 \ 0.5625 \ 0.5625 \ -0.0625]$$

$$T_{\frac{1}{2}}^2(0,1)=[0.015625 \ -0.25 \ 0.46875 \ -0.25 \ 0.015625]$$

$$T_{\frac{1}{2}}^2(1,0)=[-0.25 \ 0.5 \ -0.25]$$

$$5 \quad T_{\frac{1}{2}}^2(1,1)=[0.0625 \ -0.5625 \ -0.5625 \ 0.0625]$$

$$T_{\frac{1}{2}}^3(0,0)=[-0.0020 \ -0.0215 \ 0.2227 \ 0.8867 \ -0.0957 \ 0.0098]$$

$$T_{\frac{1}{2}}^3(0,1)=[0.0005 \ 0.0020 \ -0.1140 \ 0.3242 \ -0.2290 \ 0.0488 \ -0.0024]$$

$$T_{\frac{1}{2}}^3(1,0)=[-0.0078 \ -0.1406 \ 0.3438 \ -0.2344 \ 0.0391]$$

$$T_{\frac{1}{2}}^3(1,1)=[0.0020 \ 0.02148 \ -0.5352 \ 0.3633 \ 0.1582 \ -0.0098]$$

- 10 考虑到许多可能应用本发明公开的原理的实施例，应当认识到所说明的实施例只是本发明的首选例子，不应该将其作为对发明范围的限制。更确切地，本发明的范围通过下面的权利要求书进行限定。为此我们要求在这些权利要求的范围和精神内的全部作为我们的发明。

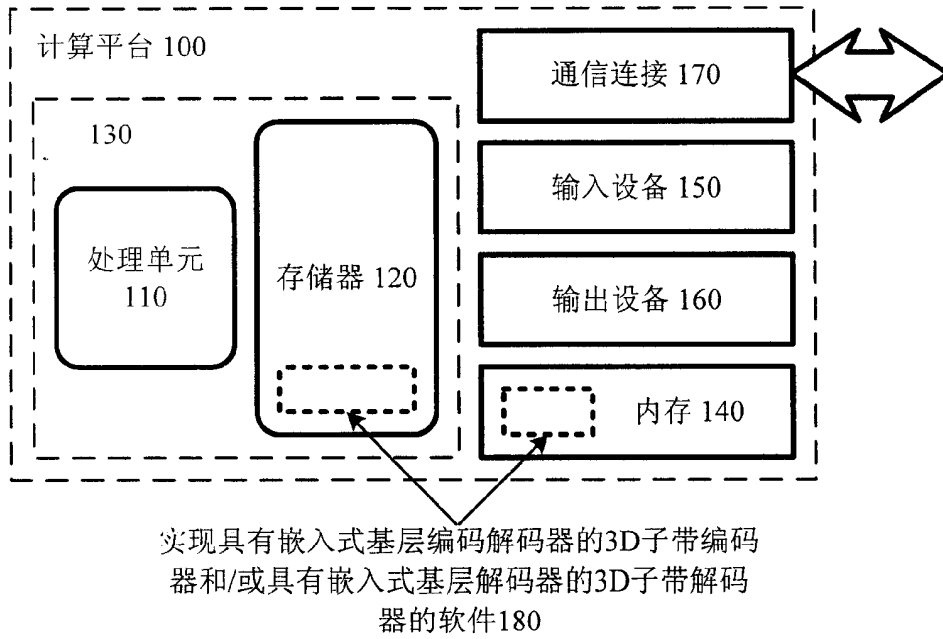


图 1

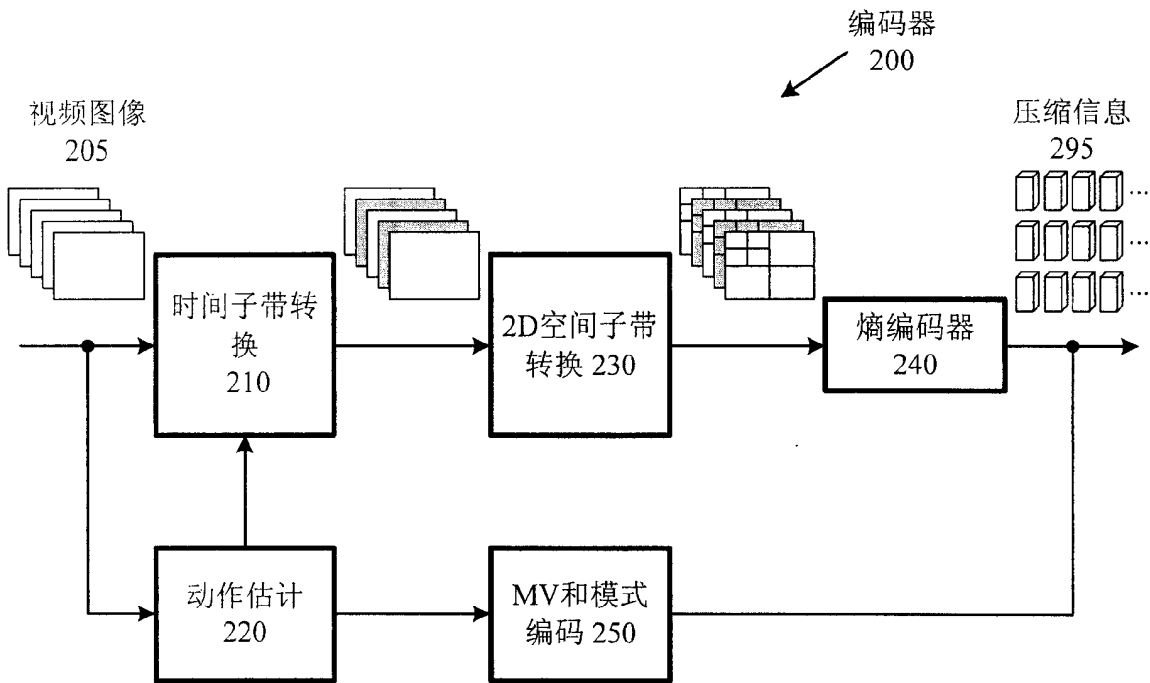


图 2

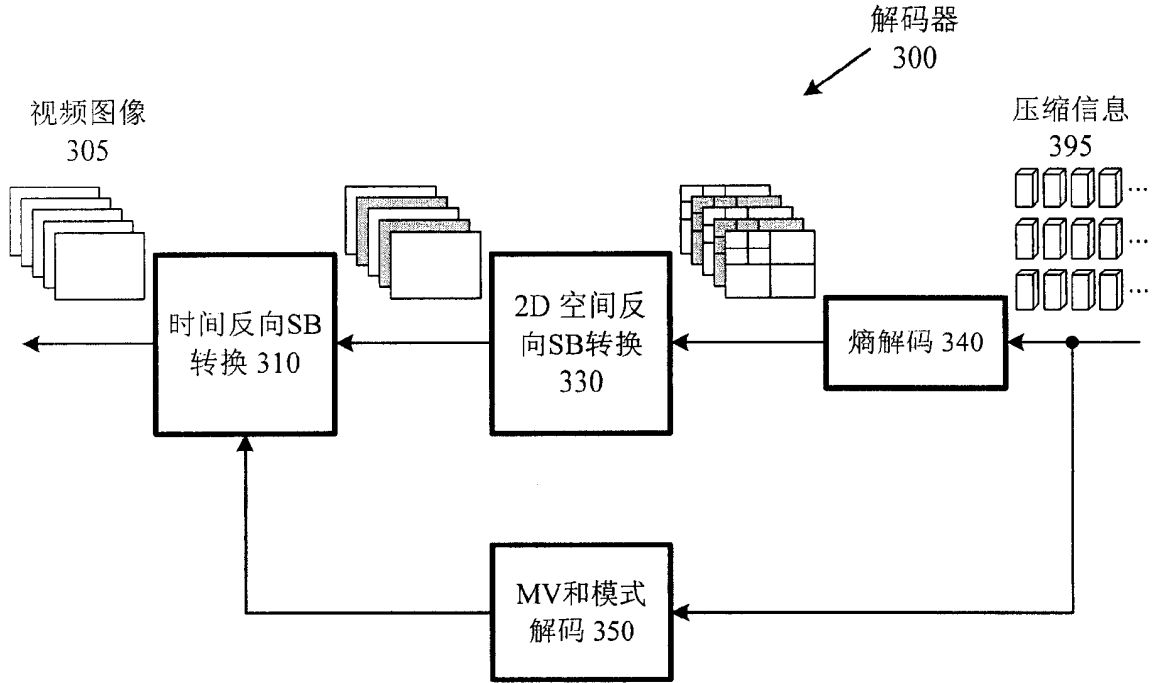


图 3

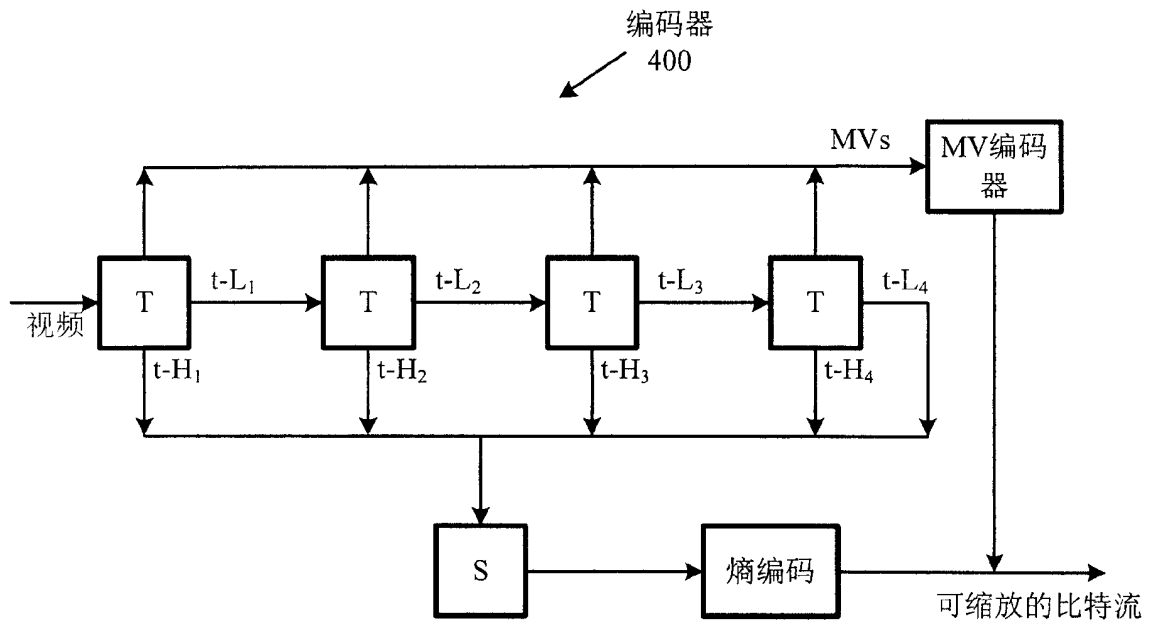


图 4

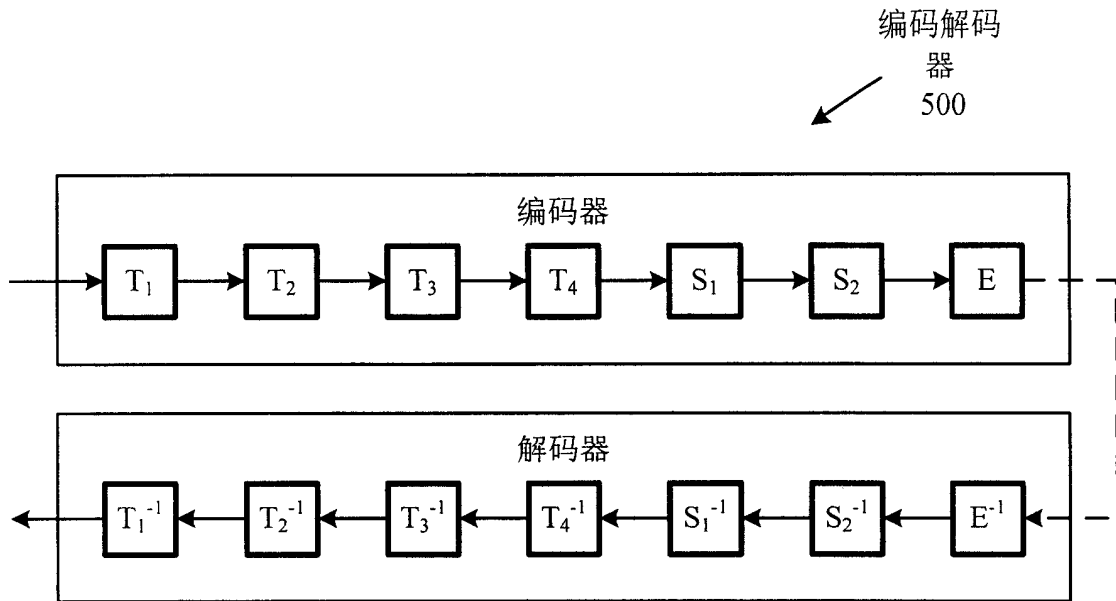


图 5

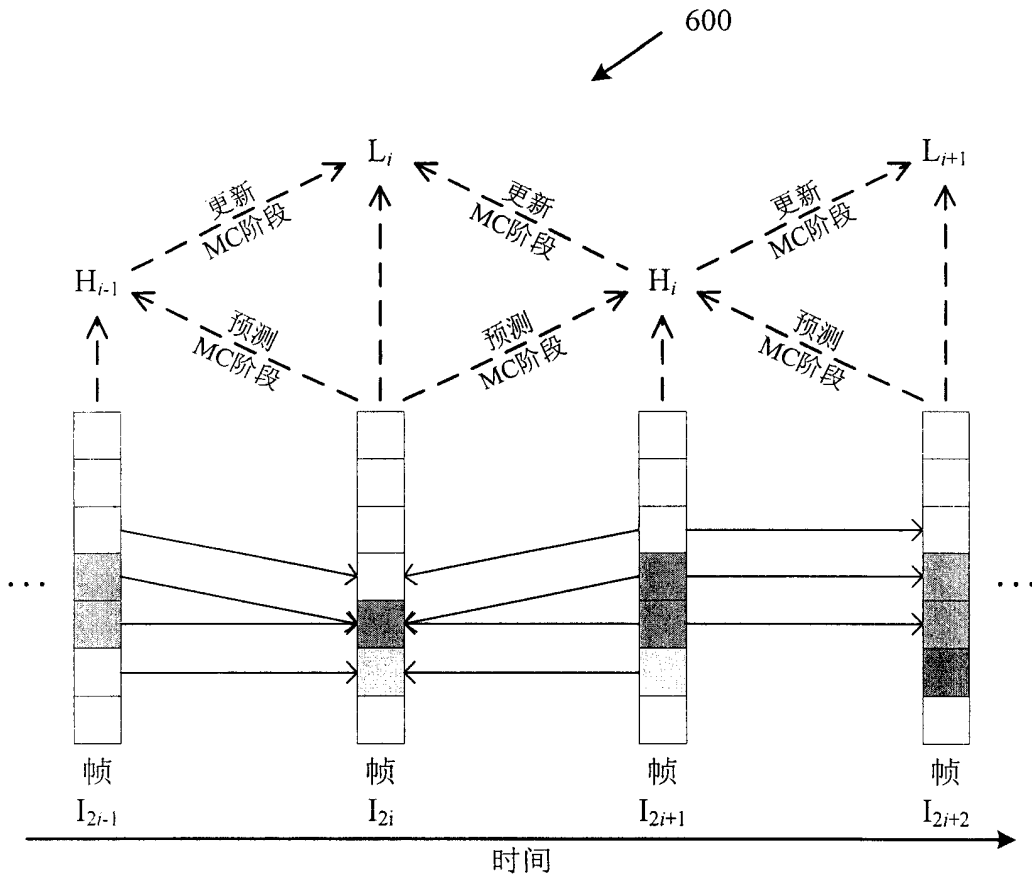


图 6

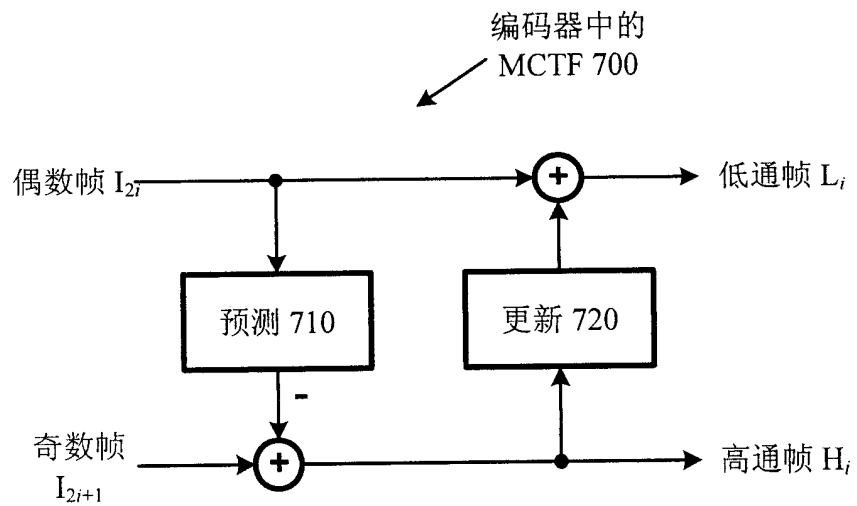


图 7

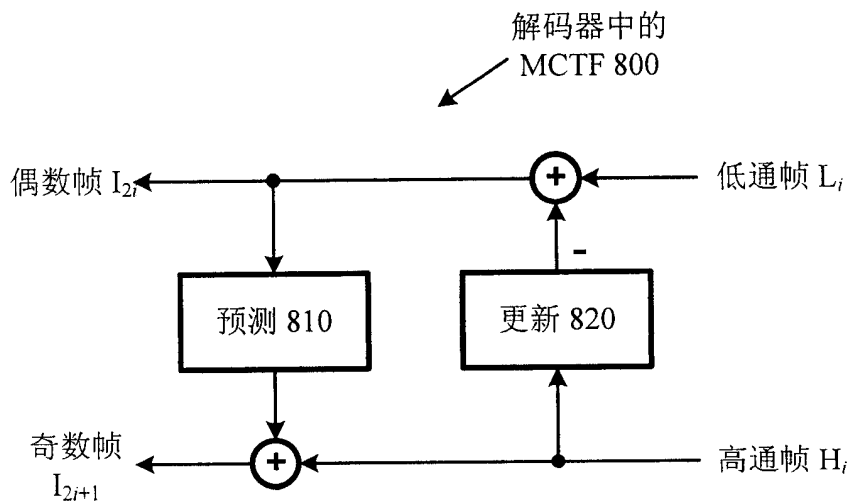


图 8

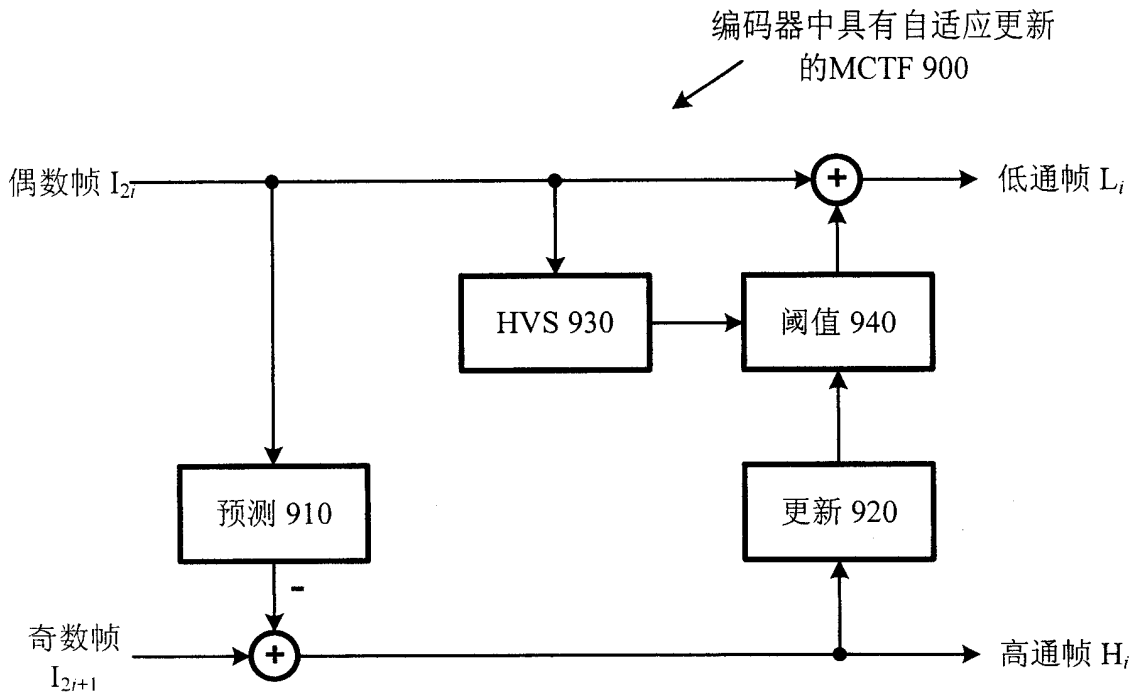


图 9

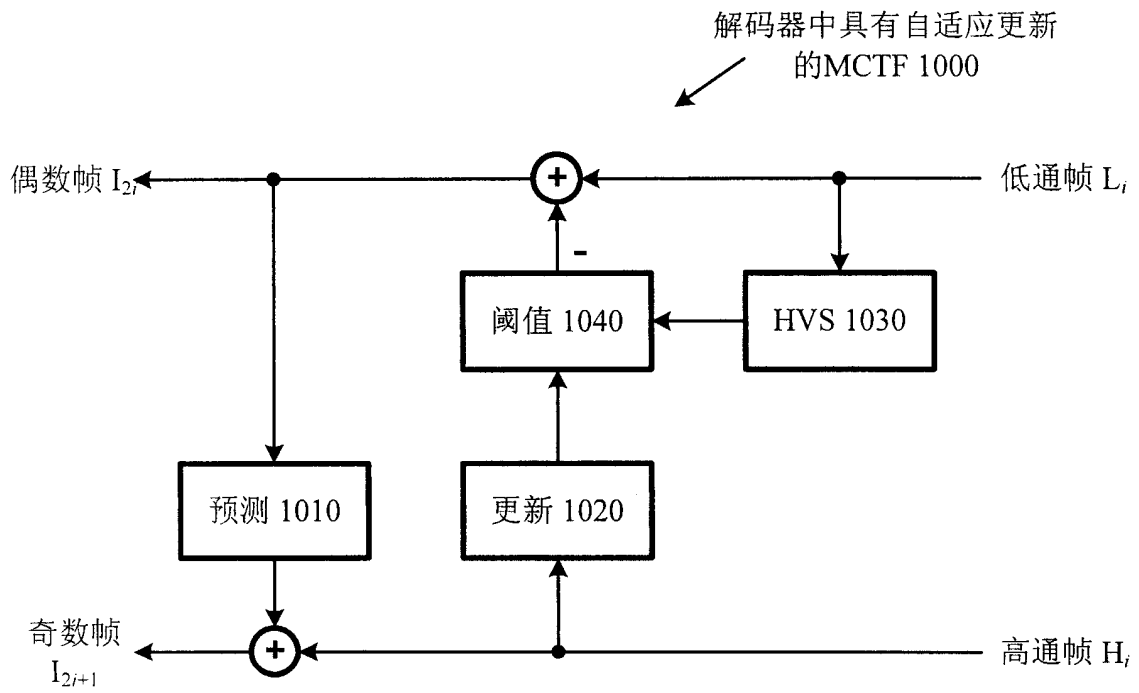


图 10

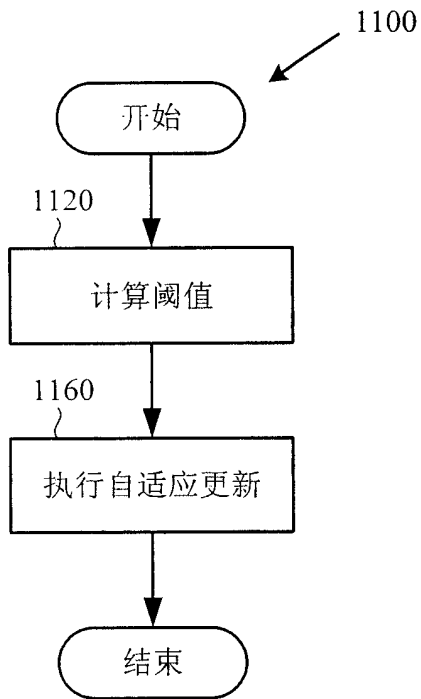


图 11

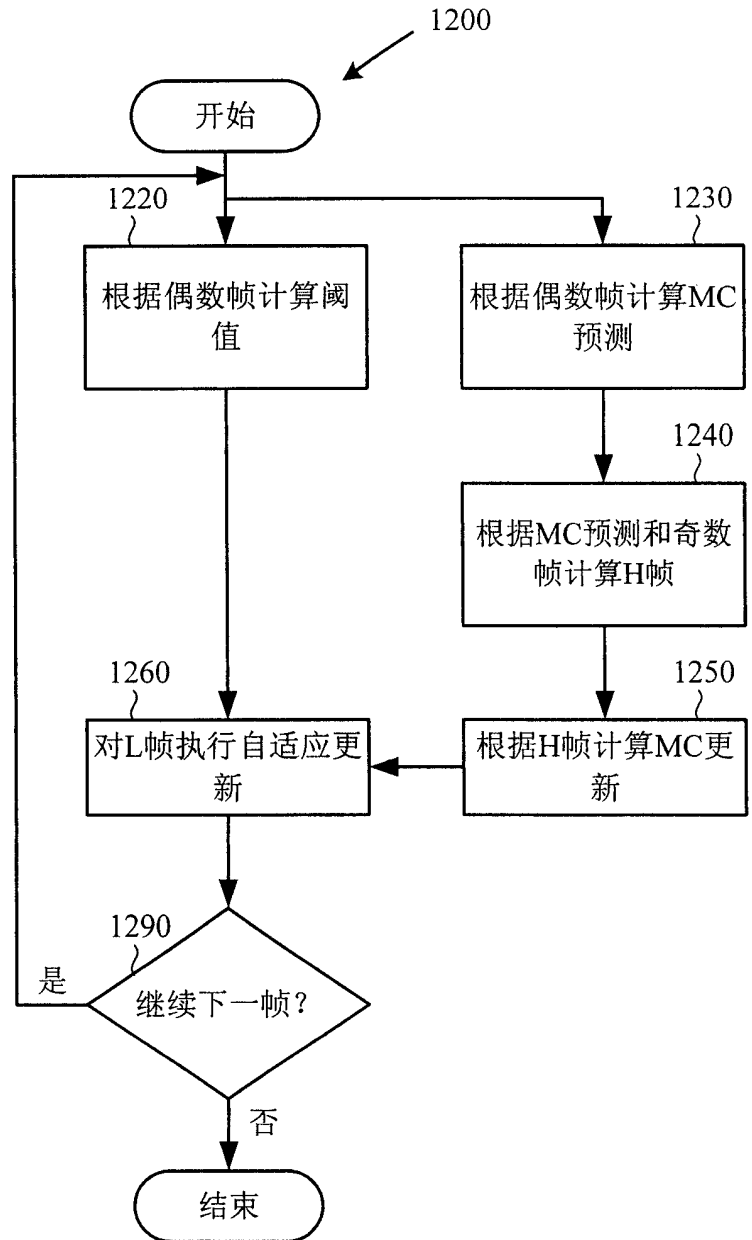


图 12

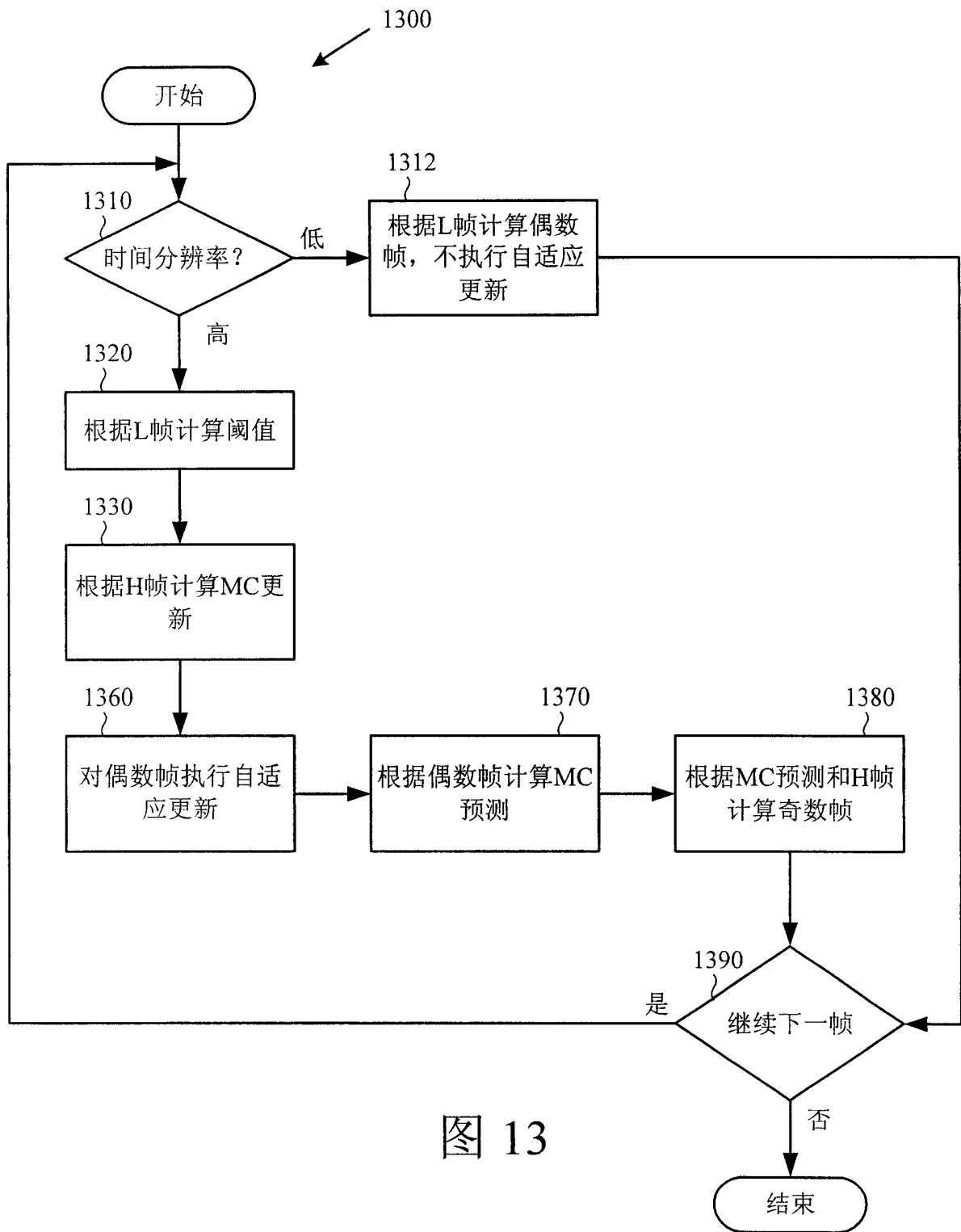


图 13

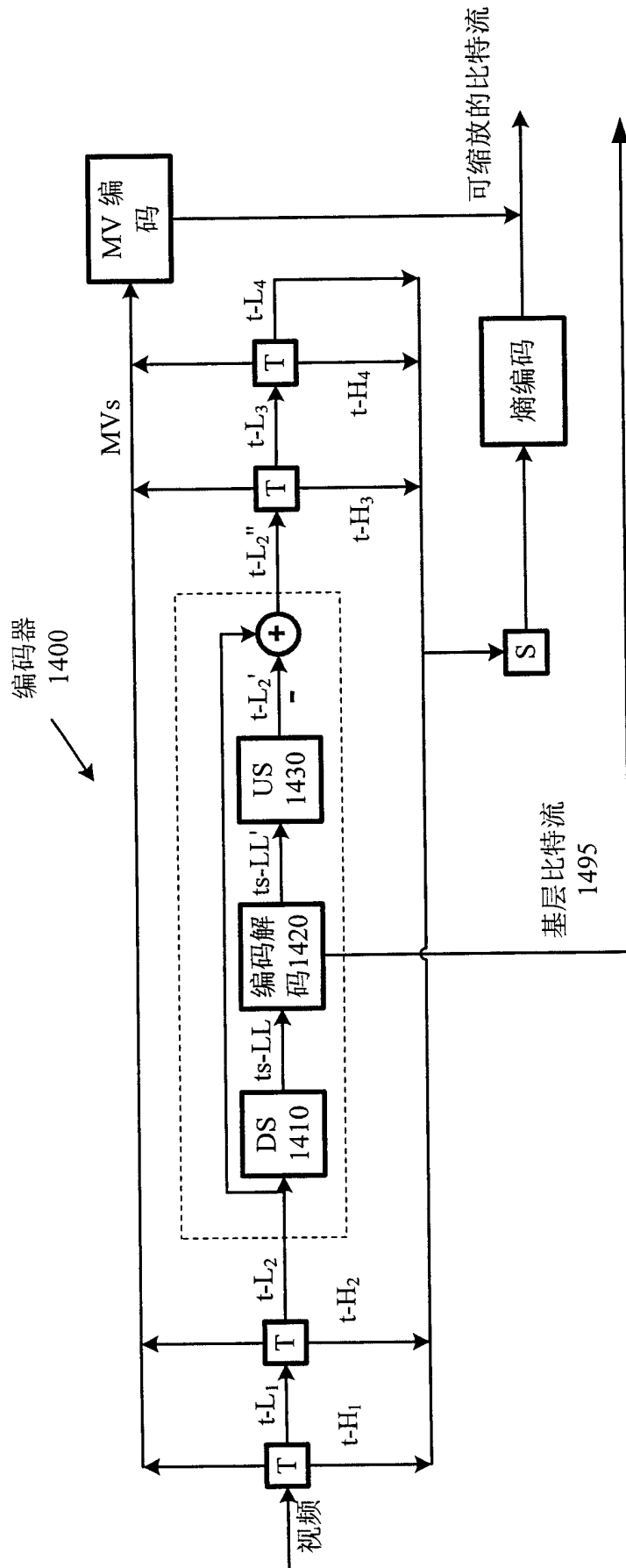


图 14

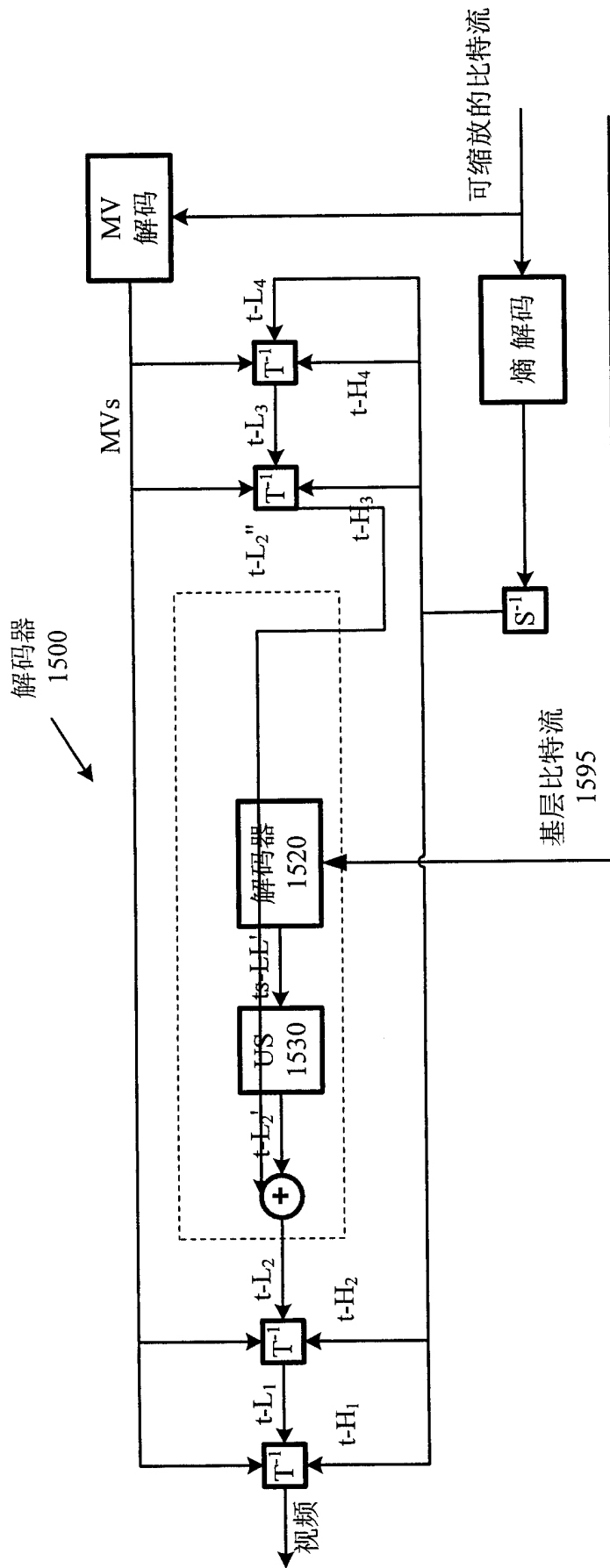


图 15

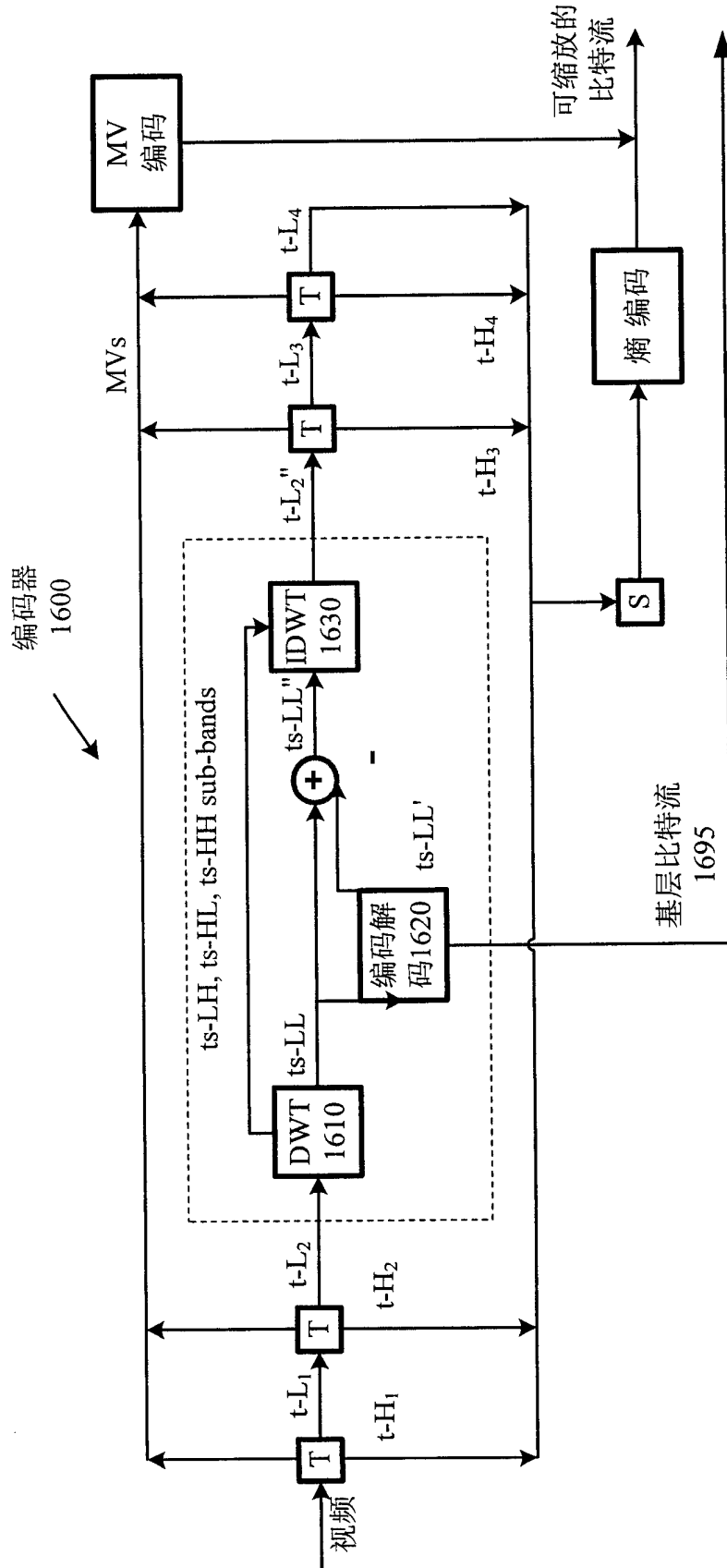


图 16

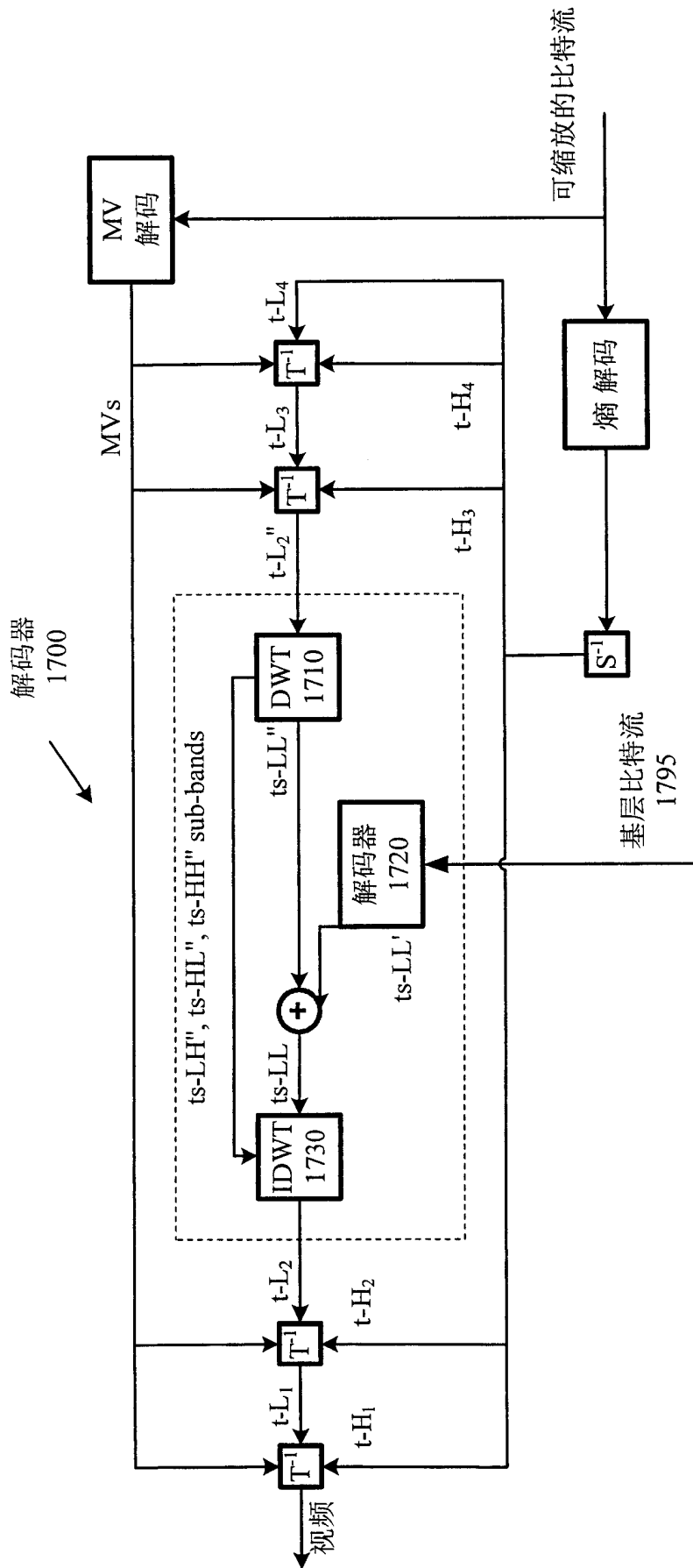


图 17

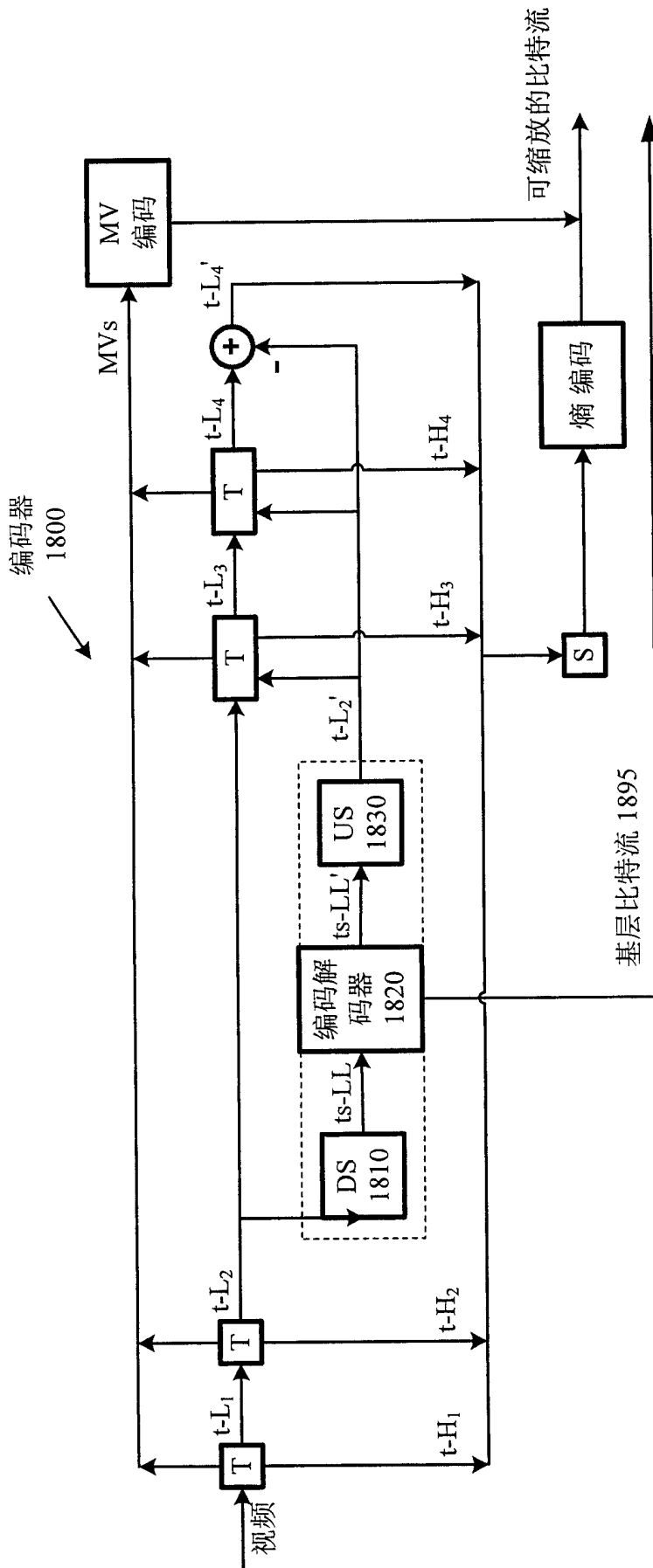


图 18

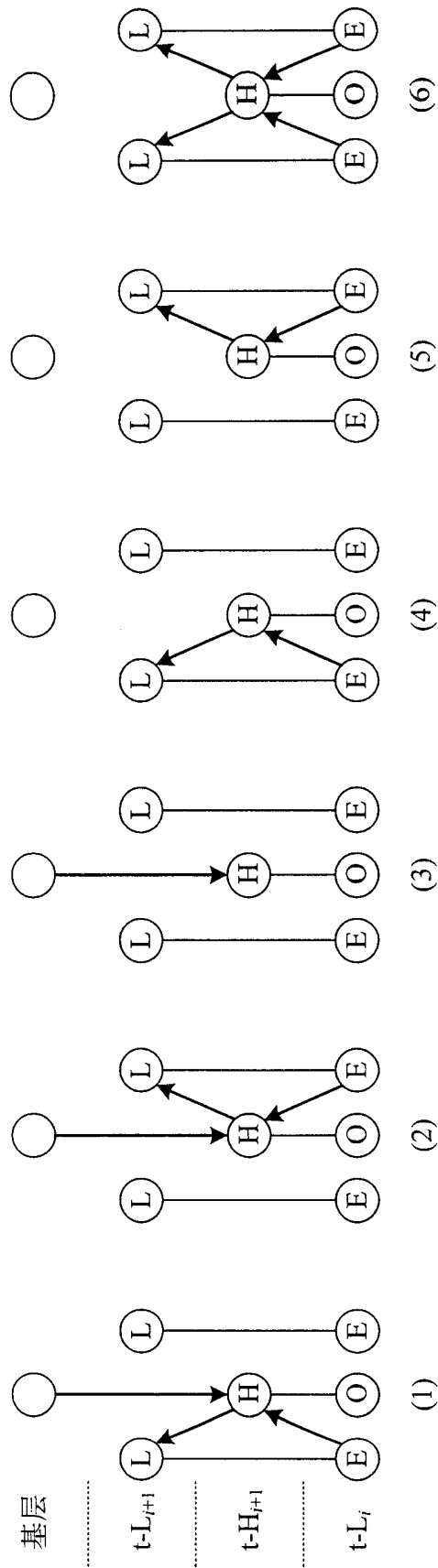


图 20

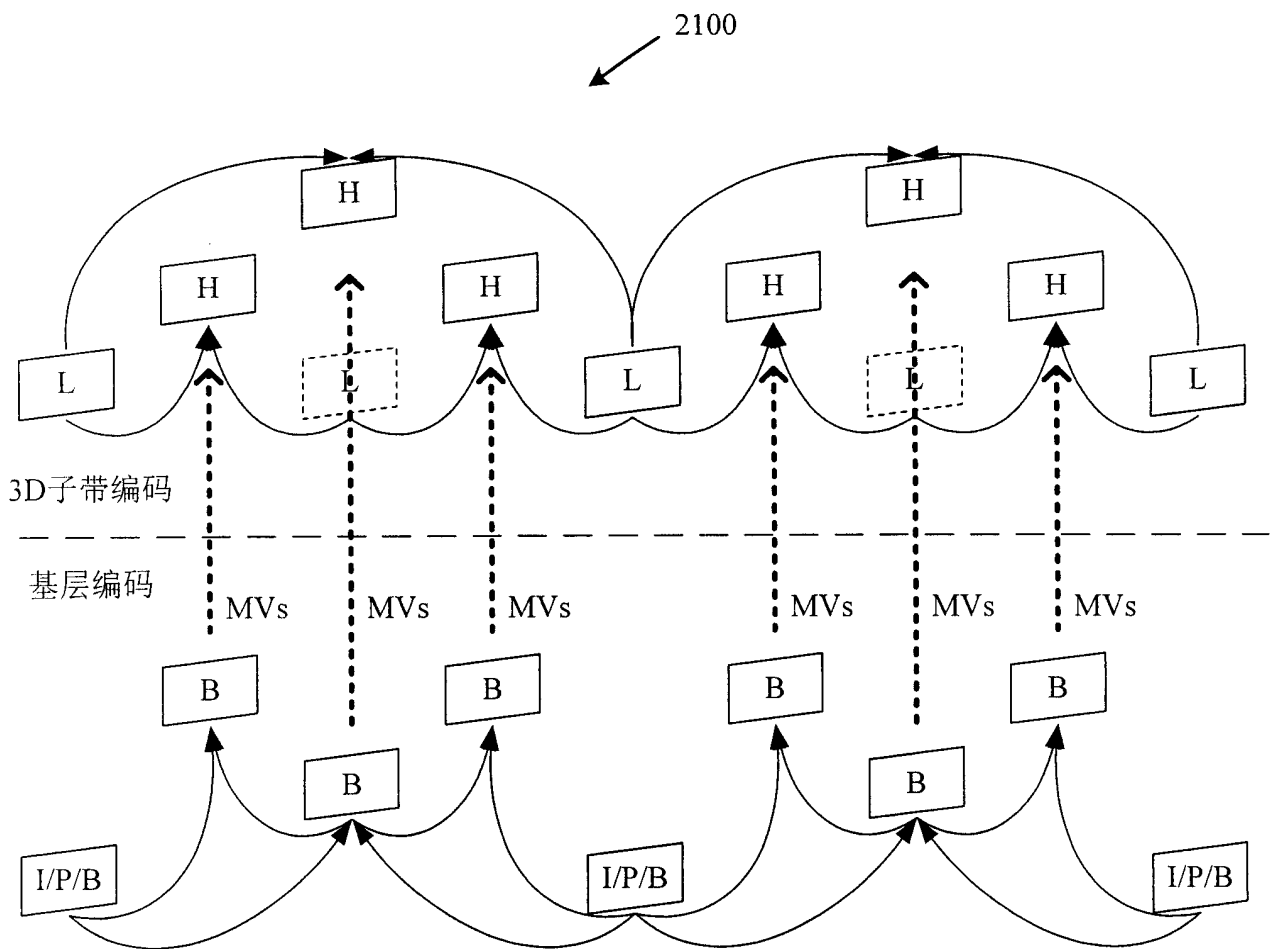


图 21

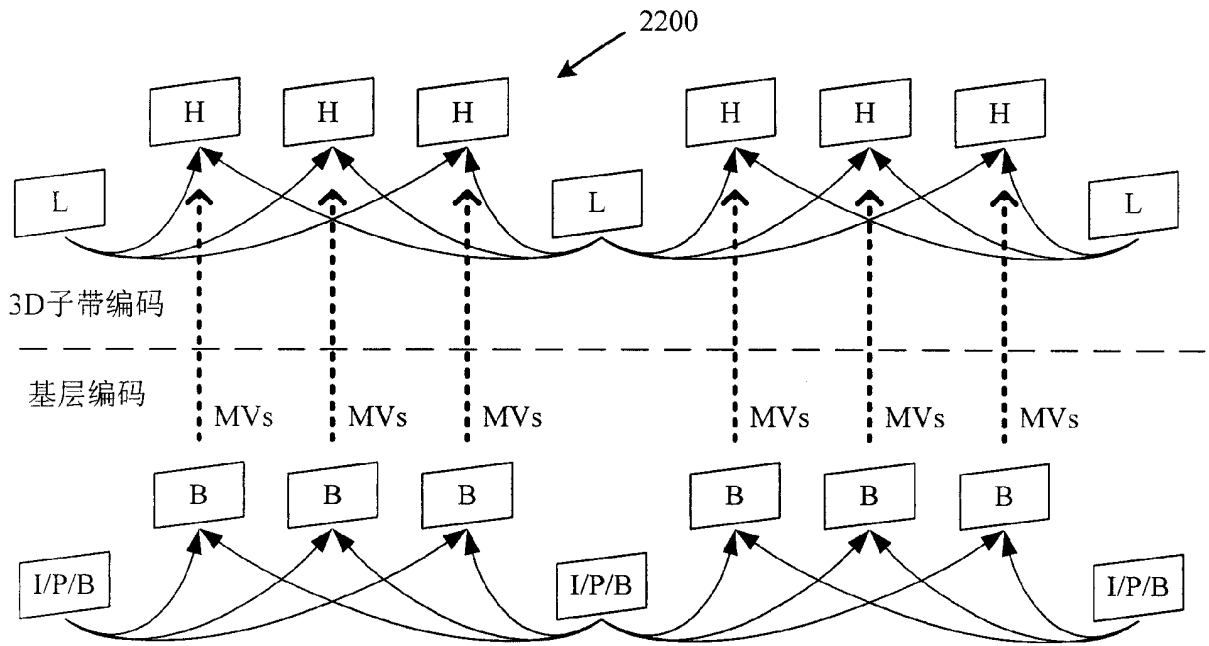


图 22

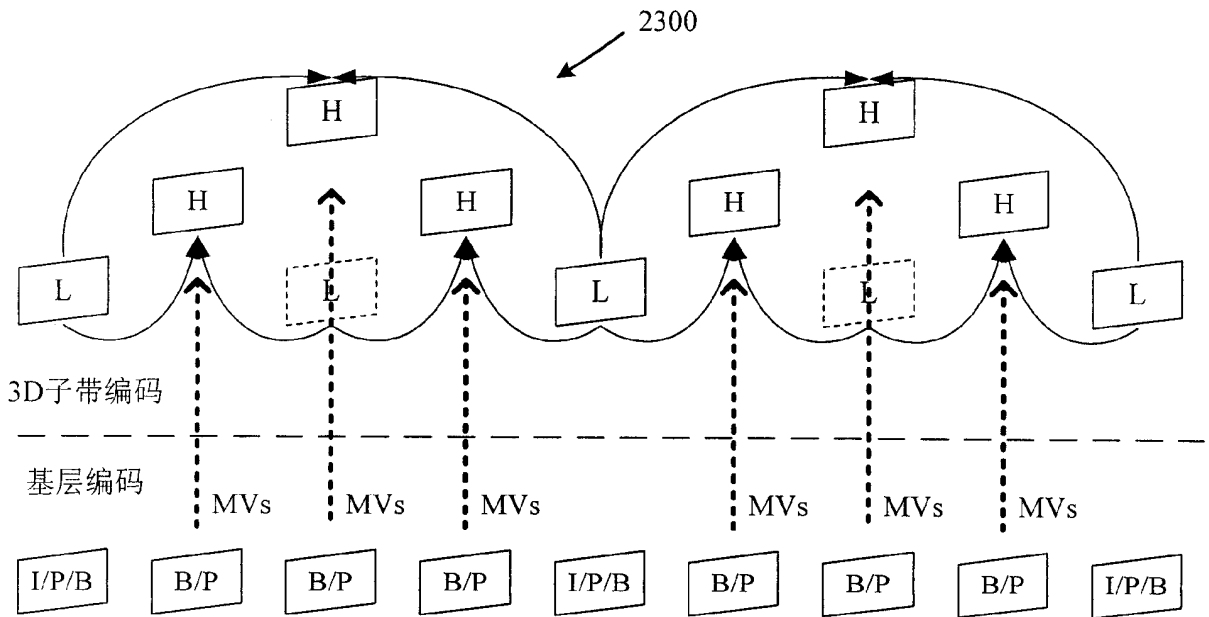


图 23

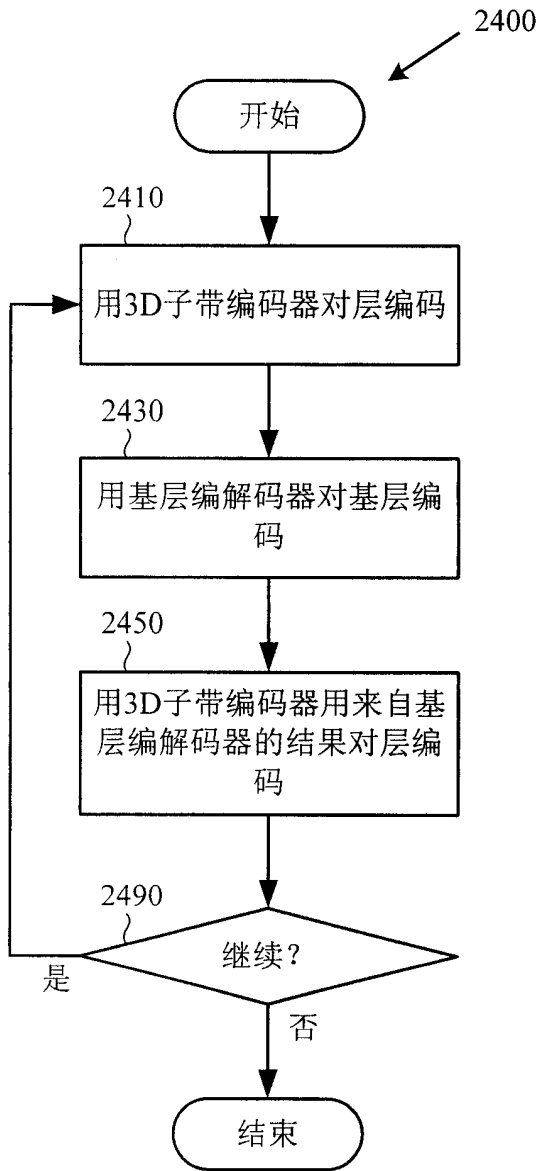


图 24

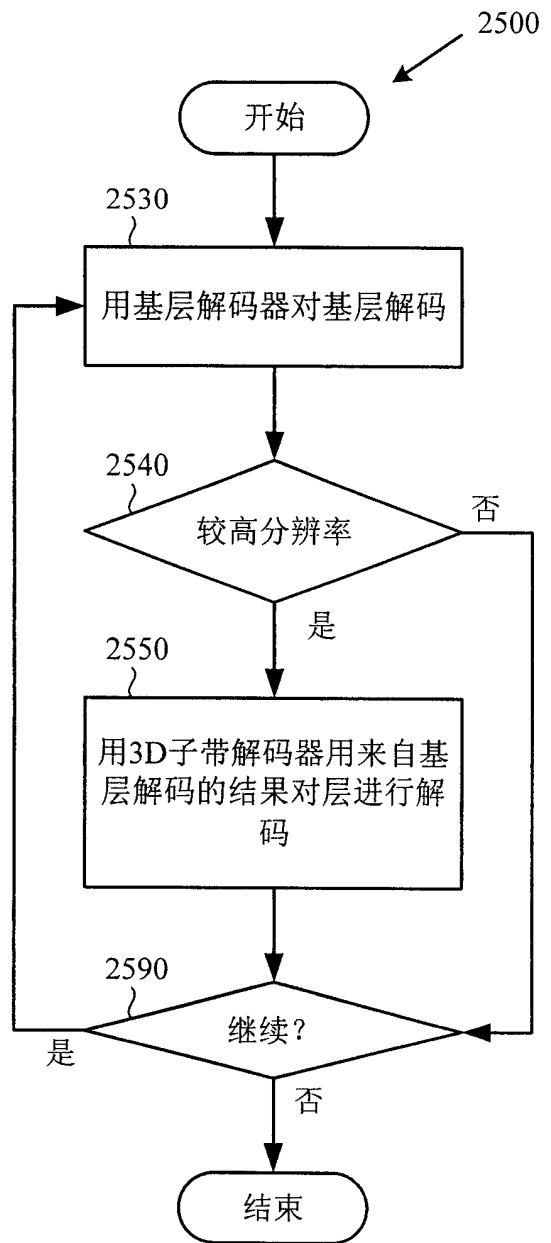


图 25

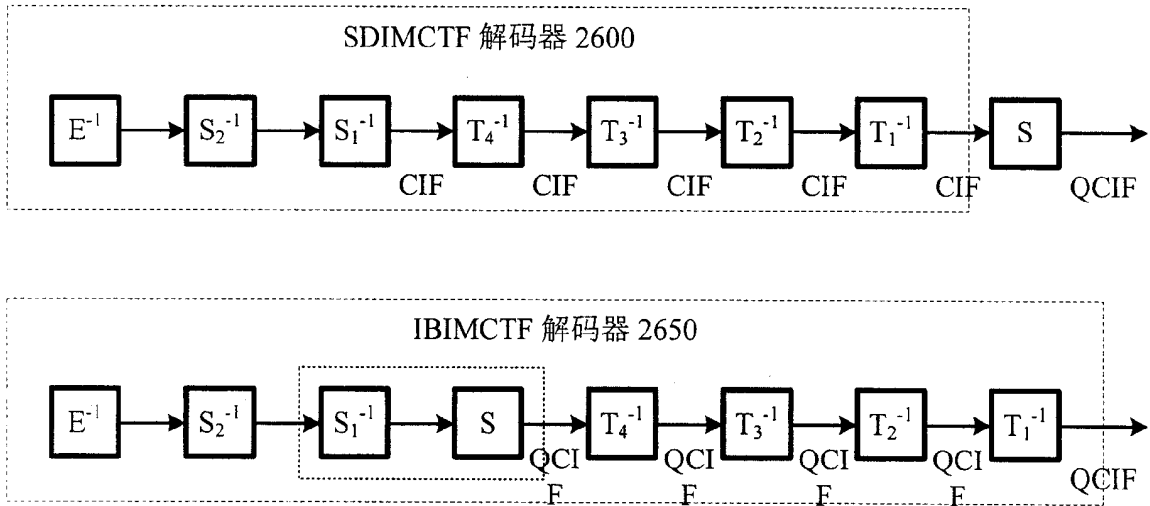


图 26

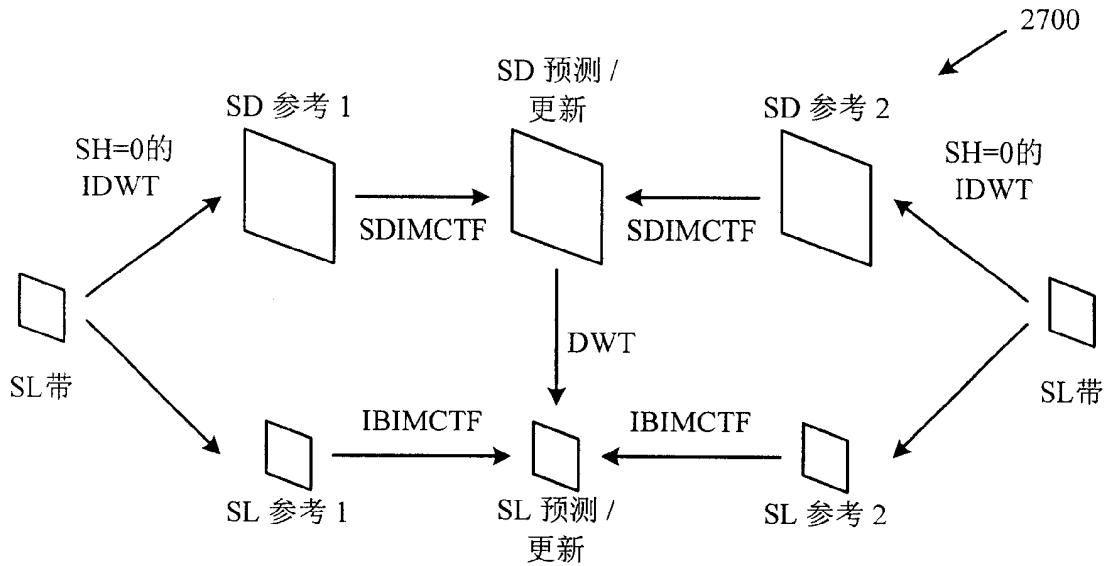


图 27

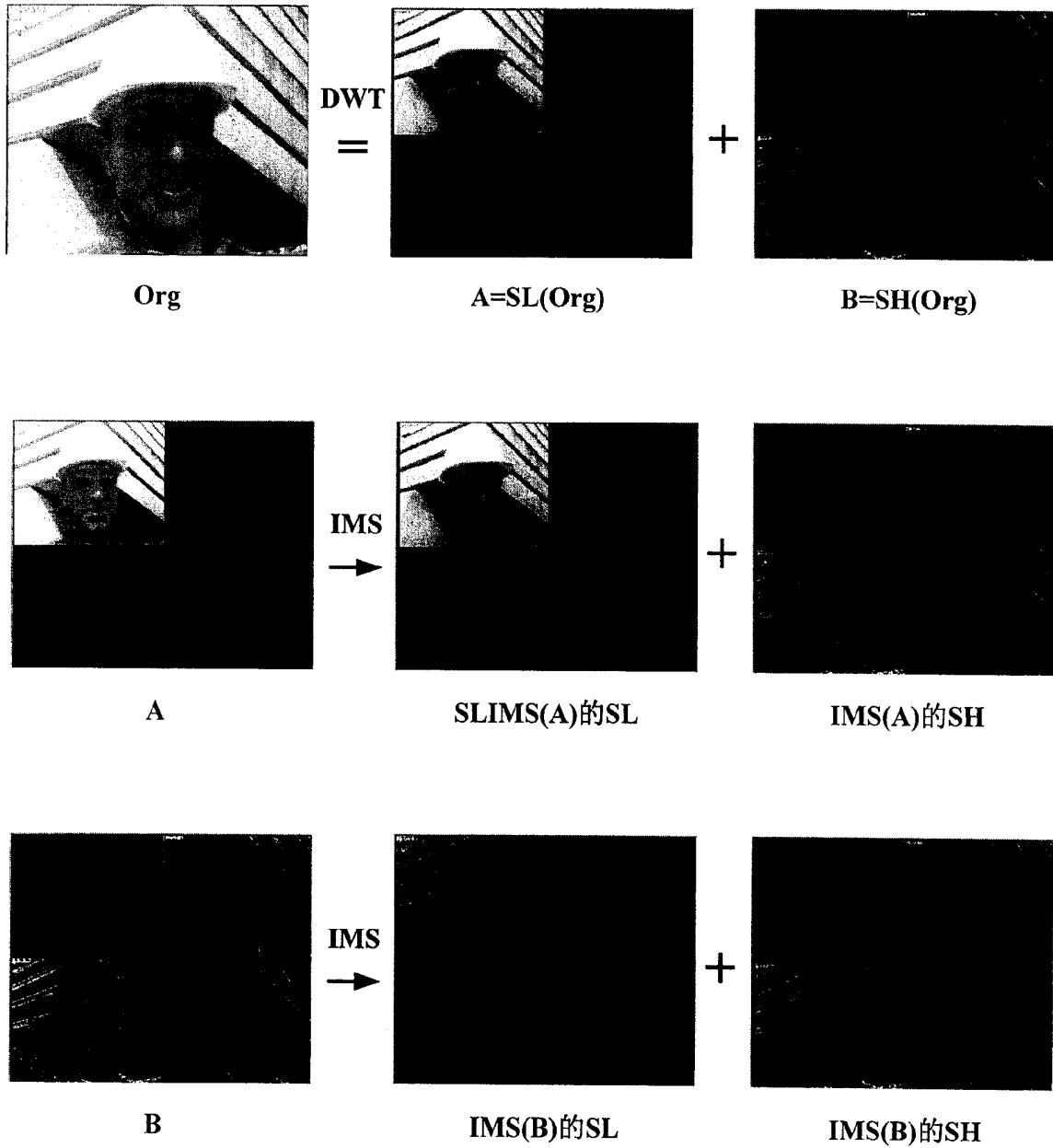


图 28

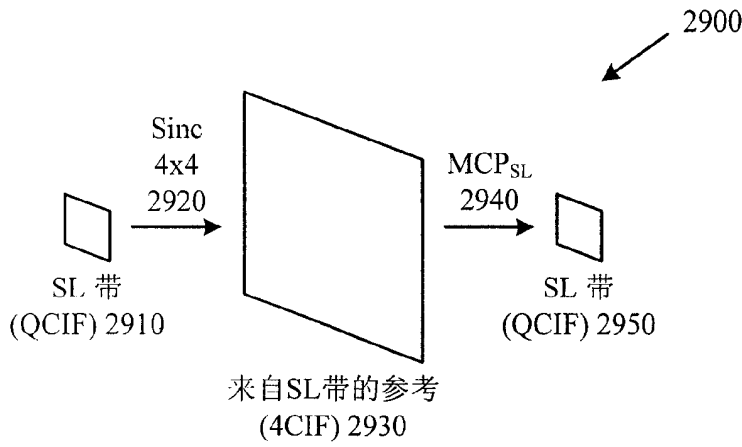


图 29

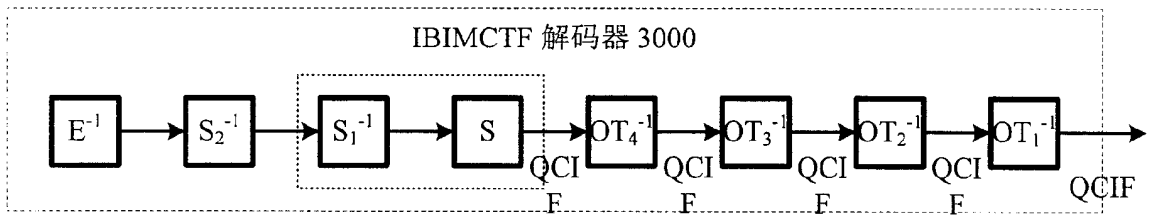


图 30

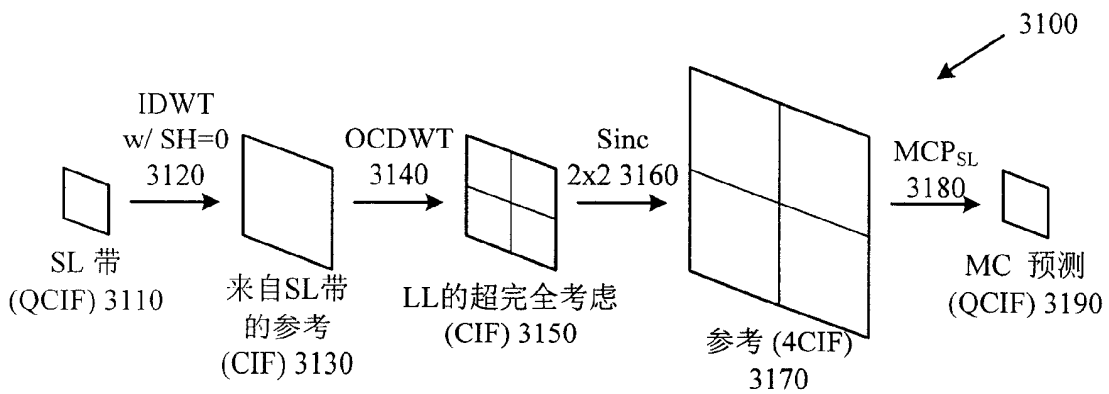


图 31

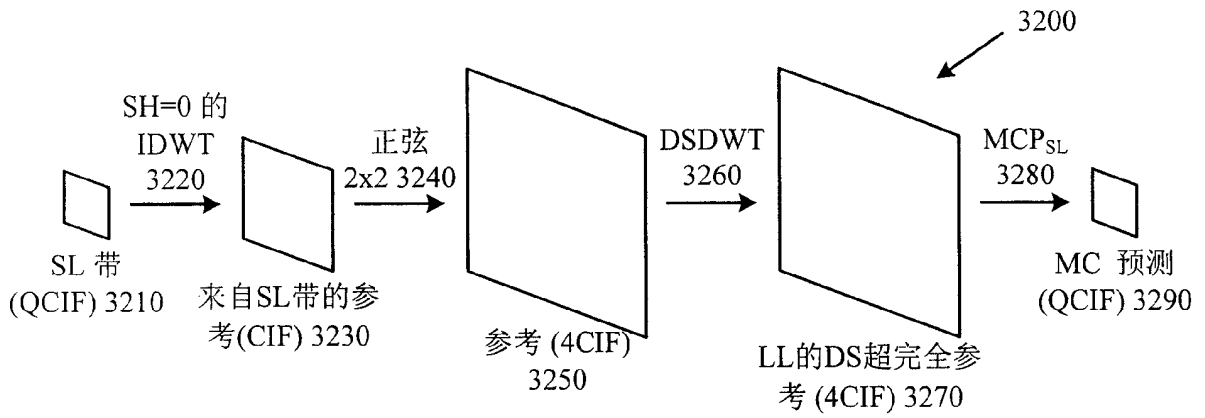


图 32

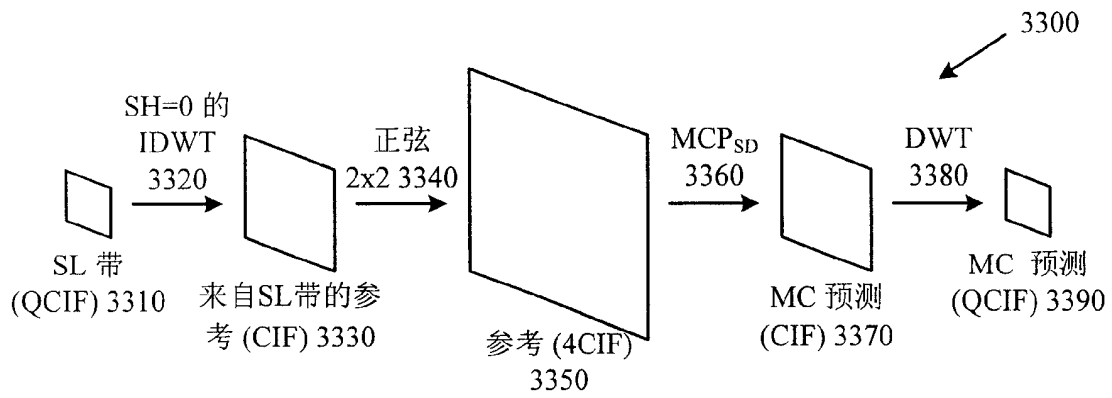


图 33

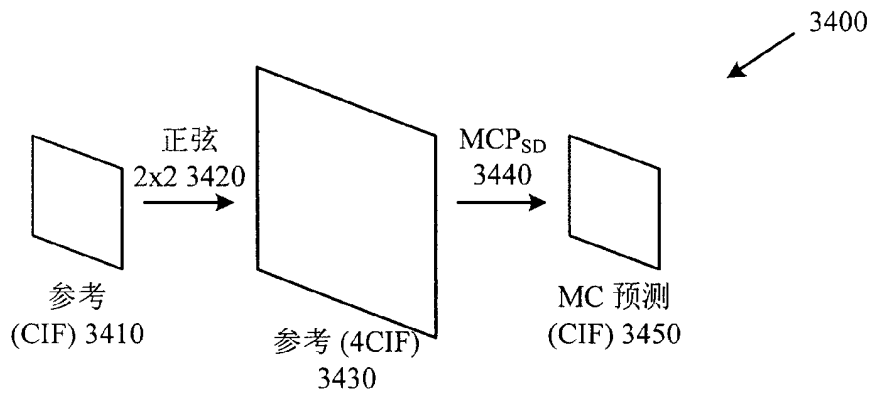


图 34