



(12) 发明专利

(10) 授权公告号 CN 1735208 B

(45) 授权公告日 2011. 12. 14

(21) 申请号 200510084714. 6

(22) 申请日 2005. 07. 12

(30) 优先权数据

60/587, 923 2004. 07. 12 US

11/126, 997 2005. 05. 10 US

(73) 专利权人 微软公司

地址 美国华盛顿州

(72) 发明人 F·吴 J·徐 L·宋

(74) 专利代理机构 上海专利商标事务所有限公  
司 31100

代理人 张政权

(51) Int. Cl.

H04N 7/26 (2006. 01)

(56) 对比文件

WO 00/74385 A2, 2000. 12. 07, 说明书摘要.

JP 特开平 9-74561 A, 1997. 03. 18, 全文.

US 5495292 A, 1996. 02. 27, 全文.

CN 1378384 A, 2002. 11. 06, 说明书第 1 页第

22 行 - 第 4 页第 25 行、第 12 页第 2 行 - 17 页第 2 行.

US 6259631 B1, 2003. 03. 04,

A Sultan, HA Latchman. Adaptive quantization scheme for MPEG video coders based on HVS (Human Visual System).

Proceedings of SPIE Vol. 2668, 1996, Vol. 2668 1 81-188.

杨柳 徐维朴. 一种新型基于运动补偿的自适应去隔行算法. 《微电子学与计算机》. 2002, (第 9 期), 4-6.

审查员 吴永兴

权利要求书 3 页 说明书 33 页 附图 22 页

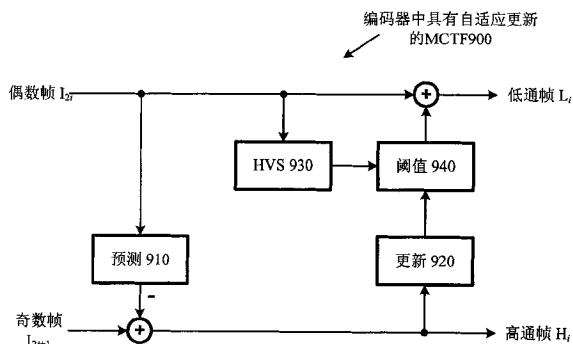
(54) 发明名称

在运动补偿的时间滤波中的自适应更新

(57) 摘要

描述用于可缩放视频编码和解码的技术和工具。例如，视频编码器和解码器分别在基于提升的运动补偿时间滤波和逆运动补偿的时间滤波中，执行自适应更新操作。在预测阶段，编码器计算用于奇数图像的运动补偿预测和将高通图像计算为奇数图像和预测间的差值。对更新阶段，编码器在高通图像上执行运动补偿（以及可能另一高通图像）以便计算运动补偿更新。基于感知准则，编码器将更新和偶数图像自适应组合以形成低通图像。对更新阶段，解码器在一个或多个高通图像上执行运动补偿以便计算运动补偿更新。基于感知准则，编码器将更新和低通图像组合以重构偶数帧。

CN 1735208 B



1. 一种运动补偿时间滤波或逆运动补偿时间滤波的方法,包括运动补偿预测阶段和运动补偿更新阶段,所述方法包括:

自适应来自运动补偿更新阶段的更新信息,其中,所述自适应至少部分地基于人类视觉系统的模型,所述自适应包括从要应用所述更新信息的图像的样本计算阈值,所述阈值包括一个或多个图像相关阈值,用于对尤其能感知失真的图像部分限制更新信息而对其它部分的图像不限制更新信息;以及

将所述自适应的结果应用在编码或解码中。

2. 如权利要求 1 所述的方法,其特征在于,所述模型是最小可察觉差模型。

3. 如权利要求 1 所述的方法,其特征在于,编码器和解码器将相同的度量用于所述自适应。

4. 如权利要求 1 所述的方法,其特征在于,所述一个或多个图像相关阈值根据所述采样值在所述图像的空间邻域中的方差改变。

5. 如权利要求 1 所述的方法,其特征在于,所述一个或多个图像相关阈值对所述图像内的平坦区域较低,而对所述图像内的噪声区较高。

6. 如权利要求 1 所述的方法,其特征在于,所述一个或多个图像相关阈值根据对比度调整调谐参数改变。

7. 如权利要求 1 所述的方法,其特征在于,所述自适应包括,对于要应用所述更新信息的所述图像:

使绝对值低于适用阈值的一部分更新信息通过;

否则将该部分更新信息限制到按照由该部分更新信息的符号修改的适用阈值,这包括:

如果该部分更新信息的符号是正的,则将该部分更新信息限制到所述适用阈值;

如果该部分更新信息的符号是负的,则将该部分更新信息限制到负的所述适用阈值。

8. 如权利要求 1 所述的方法,其特征在于,所述要应用所述更新信息的图像是第二图像,所述方法还包括,在编码期间:

在运动补偿预测阶段,对第一图像,

参考所述第二图像,计算用于所述第一图像的运动补偿预测;以及

计算所述第一图像和用于所述第一图像的所述运动补偿预测间的差值作为高通图像;以及

在所述高通图像上执行运动补偿,其中,经运动补偿的高通图像对所述更新信息有贡献,其中,所述应用包括将所述更新信息与所述第二图像组合以形成低通图像,其中,所述方法还包括后续的空间分解和/或熵编码,

其中,所述第二图像是偶视频帧,所述第一图像是奇视频帧。

9. 如权利要求 1 所述的方法,其特征在于,还包括在解码期间,在一个或多个高通图像上执行运动补偿,其中,经运动补偿的一个或多个高通图像对所述更新信息有贡献,其中,所述自适应包括将所述更新信息与低通图像组合,以及其中,所述应用包括后续的运动补偿预测和/或显示。

10. 如权利要求 1 所述的方法,其特征在于,在编码期间,所述要应用所述更新信息的图像是偶数帧,而在解码期间,所述要应用所述更新信息的图像是低通帧,所述运动补偿

时间滤波是空间域运动补偿时间滤波或带内运动补偿时间滤波,所述逆运动补偿时间滤波是空间域逆运动补偿时间滤波或带内逆运动补偿时间滤波。

11. 一种用于可缩放的视频编码或解码方法,包括:

至少部分地基于一个或多个感知准则,计算用于基于提升的运动补偿时间滤波或逆运动补偿时间滤波的一个或多个更新阈值,其中,所述计算一个或多个更新阈值使用要应用所述更新信息的图像的样本;以及

使用所述一个或多个更新阈值执行自适应更新操作,包括根据所述一个或多个更新阈值,对来自运动补偿更新阶段的更新信息进行阈值处理,所述一个或多个更新阈值包括一个或多个图像相关阈值,用于对尤其能感知失真的图像部分限制更新信息而对其它部分的图像不限制更新信息。

12. 如权利要求 11 所述的方法,其特征在于,对要被更新的多个图像的每一个计算每个图像的一个或多个更新阈值。

13. 如权利要求 11 所述的方法,其特征在于,所述一个或多个更新阈值根据所述采样值在所述图像内的空间邻域中的方差改变。

14. 如权利要求 11 所述的方法,其特征在于,所述一个或多个更新阈值对所述图像内的平坦区域较低,而对所述图像内的噪声区较高。

15. 如权利要求 11 所述的方法,其特征在于,所述一个或多个更新阈值根据对比度调整调谐参数改变。

16. 如权利要求 11 所述的方法,其特征在于,所述阈值处理包括,对于所述要应用所述更新信息的图像:

使绝对值低于所述一个或多个更新阈值的适用更新阈值的一部分更新信息通过;

否则,将该部分更新信息限制到按照由该部分更新信息的符号修改的所述适用更新阈值,这包括:

如果该部分更新信息的符号是正的,则将该部分更新信息限制到所述适用更新阈值;

如果该部分更新信息的符号是负的,则将该部分更新信息限制到负的所述适用更新阈值。

17. 如权利要求 11 所述的方法,其特征在于,在编码期间,所述要应用所述更新信息的图像是偶数帧,而在解码期间,所述要应用所述更新信息的图像是低通帧。

18. 一种视频编码器,包括:

至少部分地基于一个或多个感知准则,计算用于基于提升的运动补偿时间滤波或逆运动补偿时间滤波的一个或多个更新阈值的装置,其中,所述计算一个或多个更新阈值使用要应用所述更新信息的图像的样本;以及

使用所述一个或多个更新阈值执行自适应更新操作的装置,包括根据所述一个或多个更新阈值,对来自运动补偿更新阶段的更新信息进行阈值处理,所述一个或多个更新阈值包括一个或多个图像相关阈值,用于对尤其能感知失真的图像部分限制更新信息而对其它部分的图像不限制更新信息。

19. 一种视频解码器,包括:

至少部分地基于一个或多个感知准则,计算用于基于提升的运动补偿时间滤波或逆运动补偿时间滤波的一个或多个更新阈值的装置,其中,所述计算一个或多个更新阈值使用

要应用所述更新信息的图像的样本;以及

使用所述一个或多个更新阈值执行自适应更新操作的装置,包括根据所述一个或多个更新阈值,对来自运动补偿更新阶段的更新信息进行阈值处理,所述一个或多个更新阈值包括一个或多个图像相关阈值,用于对尤其能感知失真的图像部分限制更新信息而对其它部分的图像不限制更新信息。

## 在运动补偿的时间滤波中的自适应更新

[0001] 相关申请的交叉引用

[0002] 本发明要求 2004 年 7 月 12 日提交的 U. S. 临时申请 No. 30/587, 923 的优先权, 其内容在此引入以供参考。

### 技术领域

[0003] 描述用于可缩放的视频编码和解码的技术和工具。例如, 视频编码器和解码器在基于提升的运动补偿时间滤波中执行自适应更新操作。

### 背景技术

[0004] 数字视频消耗大量存储和传输容量。典型的原始数字视频序列每秒包括 15 或 30 帧。每帧能包括几万或几十万像素 (也称为象素), 其中每个像素表示图像的极小单元 (tiny element)。在原始形式中, 计算机通常将像素表示为三个样本的集合, 总共 24 比特。例如, 像素可以包括定义像素的灰度级分量的八比特亮度样本 (也称为 luma 样本, 因此, 在此可互换使用术语“亮度”和“luma”), 以及定义像素的颜色分量的两个八比特色度样本 (也称为 chroma 样本, 因此, 在此可互换使用术语“色度”和“chroma”)。因此, 典型的原始数字视频序列的每秒的比特数, 或比特率可以是每秒 5 百万比特或更高。

[0005] 许多计算机和计算机网络缺少资源来处理原始数字视频。为此, 工程师使用压缩 (也称为编码) 来降低数字视频的比特率。压缩通过将视频转换成比特率形式, 减少存储和传输视频的成本。解压缩 (也称为解码) 由压缩形式重构初始视频的版本。“编解码器”是编码器 / 解码器系统。压缩能是无损的, 其中, 视频的质量不变坏, 但由于视频数据的固有可变量 (有时称为熵), 限制比特率降低。或者, 压缩能是有损的, 其中, 视频的质量变坏, 但可实现的比特率减少更显著。通常结合无损压缩使用有损压缩 - 有损压缩确定近似信息, 以及无损压缩应用于表示近似。

[0006] 作为视频压缩的一般规则, 质量直接与比特率有关改变。对于给定视频序列, 如果以更高质量编码序列, 用于序列的比特率将更高, 以及如果以较低质量编码序列, 用于序列的比特率将更低。各种影响会影响原始视频序列的比特率和质量, 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), 352×288 (CIF) 或 704×576 (4CIF) 像素), 以及采样分辨率 (例如每像素 8, 16 或 24 比特)。质量和比特率可以通过向上或向下移动时间、空间和 / 或采样分辨率来改变。

[0007] 质量和比特率也可以由通过有损压缩期间, 简化或去除信息引入的失真量而定。当重构时, 这影响例如视频中的模糊度 (blurriness) 量、分块效应、粒度等等。表述不同, 有损压缩降低序列的质量以便允许编码器实现更低比特率。

[0008] 作为另一通用规则, 在细节和运动方面, 质量和比特率由视频序列的复杂度而定。对一些固定的质量级, 复杂序列通常比简单序列要求编码更多的位。其另一方面是, 当以一些固定比特率编码时, 复杂序列通常具有比简单序列更低的质量。

[0009] 在一些情况下, 所需的是以单一比特率 / 质量级编码视频。例如, 如果编码用于通

过单一类型的设备重放的视频,或如果编码用于在电话线上,点对点视频会议中重放的编码,期望以单一比特率/质量级简单地编码视频。在许多其他情况下,然而,期望以多个比特率和质量级编码视频。例如,当在 Internet 上流动视频时,视频服务器通常必须向具有不同性能的设备提供视频和/或在具有不同速度和可靠性特性的各种网络环境上递送视频。

[0010] 解决不同网络和重放需求的一种方法是以多个比特率和质量级编码相同视频序列,其能导致用于多个独立压缩视频比特流的存储和传输低效。作为替代方法,子带或小波视频编码提供在单个、可缩放压缩视频比特流中,以多个分辨率方法编码视频序列。通过子带或小波编码,将视频序列解压缩成不同时间和空间子带。

[0011] 作为简单的例子,将视频序列分成低分辨率时间子带(粗略地对应于该序列的低帧率版本)和高分辨率时间子带(能与低分辨率时间子带结合以重构初始帧率序列)。用于各个视频帧的信息可以类似地划分成低分辨率空间子带和多个较高分辨率空间子带。时间和空间分解可以一起使用。可以重复任一分解类型,例如,以便进一步分解低分辨率子带。通过选择特定子带,以便以不同分辨率传输或解码,可以实现时间和空间可缩放性。

[0012] 另外,用于单个子带的信息可以表示为具有多个位分辨率层的位平面。通过传输一些,而不是全部用于子带的位,能有选择地降低初始编码信息的保真度(以及比特率)。或者,通过解码小于用于子带的所有位,能有选择地降低保真度(以及处理需求)。

[0013] 尽管可缩放视频编码和解码技术便于压缩比特流的各种空间、时间和位保真度可缩放性,现有的可缩放视频编码和解码技术存在几个缺点。

[0014] 现有的可缩放视频编码和解码技术通常不提供与以低速率的不可缩放技术竞争的性能。尽管可缩放视频编码和解码技术的性能在较高比特率和质量很好,与不可缩放视频编码和解码技术相比,在低比特率时,它们使用太多位。

[0015] 此外,根据专用不可缩放视频编码和解码技术,设计了许多现有的硬件和软件工具。这些工具的用户不愿意购买新的可缩放视频编码和解码技术以及不与现有工具兼容的工具。此外,内容提供者不愿意生产不与视频解码工具的主流安装基础兼容的编码内容。

[0016] 有时,解码器以低于初始空间分辨率的空间分辨率重放。如果解码器设备仅具有小的屏幕或如果由网络丢弃较高空间分辨率信息,这可能会发生。然而,当在编码期间,时间分解以初始空间分辨率发生时,以较低空间分辨率解码是有问题的。现有的可缩放视频解码技术不能充分地解决这一解码情况。

[0017] 最后,现有的可缩放视频编码和解码技术未考虑在编码和解码期间,某些判决中的失真可感知性。特别地,现有的可缩放视频编码技术在某些类型的时间分解中,在低分辨率时间子带中,引入过量可感知失真。

[0018] 假定对数字视频来说,压缩和解压缩最重要,可缩放视频编码和解码是充分开发的领域不奇怪。不论先前可缩放视频编码和解码技术的优点是什么,然而,它们不具有下述技术和工具的优点。

## 发明内容

[0019] 在此描述的技术和工具提供可缩放视频编码和解码的性能。特别地,视频编码器和/或解码器在编码和解码期间,解决某些判决中的失真的感知性。例如,这些判决与运动

补偿时间滤波中的更新操作有关。自适应更新操作提高由压缩的视频比特流重构的低帧速率视频的可感知质量。同时,对由压缩视频比特流重构的高帧速率视频来说,自适应更新操作仍然有效。

[0020] 根据工具和技术的第二集合,编码器,诸如 3D 子带视频编码器执行包括预测阶段和更新阶段的运动补偿时间滤波。编码器自适应至少部分基于人类视觉的模型(例如最小可觉差模型)的更新信息。例如,编码器设置一个或多个图像相关阈值,其中,图像相关阈值对平面区域来说为低,而对噪声区域来说为高。编码器然后在编码中使用该自适应结果。

[0021] 根据工具和技术的第三集合,解码器,诸如 3D 子带视频解码器执行包括预测阶段和更新阶段的逆运动补偿时间滤波。解码器自适应至少部分基于人类视觉系统的模型的更新信息,然后在解码中使用该自适应的结果。

[0022] 根据工具和技术的第四集合,编码器,诸如 3D 子带视频编码器计算用于基于至少部分基于感性标准的提升的运动补偿时间滤波的一个或多个更新阈值。例如,编码器对将更新的多个图像的每一个,计算每个图像的更新阈值。然后,编码器根据一个或多个更新阈值,确定更新信息阈值。

[0023] 根据工具和技术的第五集合,解码器,诸如 3D 子带视频解码器至少部分基于感性标准,计算用于运动补偿的时间滤波的一个或多个更新阈值。然后,解码器根据一个或多个更新阈值,确定更新信息阈值。

[0024] 从下述结合附图进行的详细描述,上述技术和工具的上述和其他目的、特征和优点将变得更显而易见。

## 附图说明

[0025] 图 1 是用于可缩放视频编码和解码的通用操作环境的框图。

[0026] 图 2,3,4 和 5 是结合可以实现各个所述实施例的通用视频编码器和 / 或解码器的框图。

[0027] 图 6 是示例说明在运动补偿时间滤波 (“MCTF”) 中,输入视频帧、时间低通帧和时间高通帧间的关系的图。

[0028] 图 7 和 8 分别是示例说明编码器中的 MCTF 和解码器中的逆 MCTF (“IMCTF”) 中的框图。

[0029] 图 9 和 10 分别是示例说明在编码器中,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。

[0030] 图 11、12 和 13 是示例说明用于自适应更新操作的技术的流程图。

[0031] 图 14、15、16、17、18 和 19 是示例说明用于将基层编解码器嵌入 3D 子带编码器中或将基层解码器嵌入 3D 子带解码器中的不同方案的框图。

[0032] 图 20 是示例说明当基层编解码器或解码器提供基准候选的不同基准组合的图。

[0033] 图 21、22 和 23 是示例说明用于通过 3D 子带编解码器和嵌入基层编解码器编码的帧图形的图。

[0034] 图 24 是示例说明通过具有嵌入基层编解码器的 3D 子带编解码器可缩放编码的技术的流程图。

[0035] 图 25 是示例说明用于通过具有嵌入基层解码器的 3D 子带解码器可缩放解码的技

术的流程图。

[0036] 图 26 是示例说明用于解码在低空间分辨率输出的 SDMCTF 编码的视频的两种方案的框图。

[0037] 图 27 是示例说明在 SDIMCTF 和 IBIMCTF 中的运动补偿预测的图。

[0038] 图 28 是示例说明由于运动偏移的功率谱泄漏的图。

[0039] 图 29 是示例说明用于解码用于在低空间分辨率输出的 SDMCTF 编码的视频的基线方案的图。

[0040] 图 30 是示例说明用于解码用于在低空间分辨率输出的 SDMCTF 编码的视频的通用方案的框图。

[0041] 图 31, 32, 33 和 34 是示例说明用于解码用于在低空间分辨率输出的 SDMCTF 编码的视频的方案的图。

### 具体实施方式

[0042] 在三维 (“3D”) 小波或子带视频编码中, 通过多次时间变换和空间变换, 视频分解成许多时间 - 空间子带。子带稍微独立, 在于因为可缩放性, 可丢失它们中的一些。例如, 为解码低空间分辨率视频, 丢失空间高通子带以及解码器仅通过空间低通子带中的所接收的数据, 执行解码。

[0043] I. 示例性操作环境

[0044] 图 1 示例说明可以实现几个所述实施例的适当的计算环境 (100) 的通用例子。计算环境 (100) 不打算建议对有关使用和功能性的范围作任何限制, 因为在各种通用或专用计算环境中可以实现所述技术和工具。

[0045] 参考图 1, 计算环境 (100) 包括至少一个处理单元 (110) 和存储器 (120)。在图 1 中, 该最基本的配置 (130) 包括在虚线内。处理单元 (110) 执行计算机可执行指令以及可以是实时或虚拟处理器。在多处理系统中, 多处理单元执行计算机可执行指令来增加处理能力。存储器 (120) 可以是易失性存储器 (例如寄存器、超高速缓存、RAM)、非易失性存储器 (例如 ROM、EEPROM、闪存等等), 或两者的一些组合。存储器 (120) 存储实现使用上述一种或多种技术的视频编码器和 / 或解码器的软件 (180)。

[0046] 计算环境可以具有另外的特征。例如, 计算环境 (100) 包括存储器 (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)。互连机制 (未示出), 诸如总线、控制器、或网络互连计算环境 (100) 的部件。典型地, 操作系统软件 (未示出) 提供用于在计算环境 (100) 中执行的其他软件的操作环境, 以及协调计算环境 (100) 的部件的活动性。

[0047] 存储器 (140) 可以是可移动或不可移动, 以及包括磁盘、磁带或磁带盒、CD-ROMs、DVDs 或能用来存储信息和能在计算环境 (100) 中存取的任何其他介质。存储器 (140) 存储用于软件 (180) 的指令。

[0048] 输入设备 (150) 可以是触摸输入设备, 诸如键盘、鼠标、笔或跟踪球、语音输入设备、扫描设备或向计算环境 (100) 提供输入的另一设备。对音频或视频编码, 输入设备 (150) 可以是声卡、视频卡、TV 调谐器卡或接收以模拟或数字形式的音频或视频输入的类似的设备, 或将音频或视频样本读入计算环境 (100) 的 CD-ROM 或 CD-RW。输入单元 (160)



可以是显示器、打印机、扬声器、CD 记录器,或从计算环境 (100) 提供输出的另一设备。

[0049] 通信连接 (170) 允许通过通信介质,与另一计算实体的通信。通信介质传送信息,诸如计算机可执行指令、音频或视频输入或输出,或以调制数据信号的其他数据。调制数据信号是具有以将信息编码在信号中的方式,设置或改变其一个或多个特性信号。举例来说,且不是限制,通信介质包括用电、光、RF、红外、声或其他载体实现的有线或无线技术。

[0050] 能在计算机可读介质的一般上下文中,描述技术和工具。计算机可读介质是能在计算环境内存取的任何可用介质。举例来说,且不是限制,通过计算环境 (100),计算机可读介质包括存储器 (120)、存储装置 (140)、通信介质和上述的任何组合。

[0051] 能在计算机可执行指令的一般上下文中,描述技术和方法,诸如包括在程序模块中、在目标实际或虚拟处理器上的计算环境中执行的那些指令。通常,程序模块包括执行特定任务或实现特定抽象数据类型的例程、程序、库、对象、类、部件、数据结构等等。可以如在各个实施例中所需的程序模块那样,组合或分离程序模块的功能性。可以在局部或分布式计算环境内,执行用于程序模块的计算机可执行指令。

[0052] 为了说明,详细的描述使用术语,类似“信号”、“确定”和“应用”来描述计算环境中的计算机操作。这些术语是用于由计算机执行的操作的高级抽象,以及不应当与由人执行的动作混淆。对应于这些术语的实际计算机操作根据实现改变。

[0053] II. 示例性视频编码器和视频解码器

[0054] 图 2 是通用视频编码器系统 (200) 的框图,结合该系统,可以实现各个所述实施例。图 4 表示在一层空间分解后,具有四层时间分解的通用编码器 (400)。图 3 是通用视频解码器系统 (300) 的框图,结合该系统,可以实现各个所述实施例。图 5 表示具有四层 (反) 时间分解和两层 (反) 空间分解的组合编码器 / 解码器系统 (500)。

[0055] 在编码器和解码器内的模块间所示的关系表示编码器和解码器中的信息的主流动,为简化起见,未示出其他关系。特别地,除运动信息外,图 2 和 3 不表示用于视频序列、帧、宏块、块等等的编码器设定值、模式、表等等的辅助信息。可能在辅助信息的熵编码后,在输出比特流中发送这些辅助信息。

[0056] 根据所需的实现和压缩类型,能增加、省略在图中所示的编码器或解码器的模块,分成多个模块,与其他模块组合,和 / 或替换类似模块。在另一实施例中,具有不同编码器和 / 或模块的其他配置的编码器或解码器执行一个或多个所述技术。例如,尽管图 2 表示在编码器系统 (200) 中,空间分解 (230) 在时间分解 (210) 下游,编码器系统 (200) 可以替代或另外在开始时间分解 (210) 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) 可以包括在逆时间分解 (310) 后的一层或多层逆空间分解。

[0057] 3D 子带视频编码实现可与以中间或高比特率的现有技术不可缩放视频编码 (例如,每 H. 264 或 VC-1 标准) 相比的性能。然而,与传统的不可缩放方法相比,花费在 3D 子带视频编码中的编码运动矢量上的比特量相当大。因此,3D 子带视频编码在低比特率时,通常比不可缩放方法执行更糟,其中,花费在编码运动矢量上的比特占了较大百分比的总比特率。

[0058] A. 示例性视频编码器

[0059] 3D 子带视频编码是通过采用视频的多级时间 / 空间子带分解的提供多级空间和 / 或时间可缩放性的巧妙方法。图 2 示例说明一般 3D 子带视频编码方案 (200)。编码器

(200) 接收一系列视频图像 (205) (例如逐行视频帧、隔行视频帧或隔行视频帧场), 以及产生压缩视频信息 (295) 作为输出。视频编码器的特定实施例通常使用通用编码器 (200) 的变化或补充版本。

[0060] 编码器 (200) 通过时间子带变换 (210), 分解视频信号。图 2 表示用于单级时间分解成低分辨率子带和高分辨率子带的输出, 但时间分解可以包括 4, 5 或一些其他级数的分解。

[0061] 在图 2 中, 时间分解 (210) 是运动补偿时间滤波 (“MCTF”), 以便编码器 (200) 执行运动估计 (220) 以便计算用于视频信号的运动信息。运动估计 (220) 的精确策略取决于实现, 以及能包括例如子像素细化、空间检索模式、考虑实际或模拟剩余编码成本的最小化码率失真函数、运动信息编码成本、运动预测模式等等后的整数像素搜索。

[0062] 在一些实现中, 编码器 (200) 在运动估计 (220) 中, 计算用于运动矢量的可变大小块 (从  $4 \times 4$  到  $16 \times 16$ ) 的运动矢量以及在 MCTF 中, 应用这些运动矢量。例如, 对  $16 \times 16$  宏块, 编码器 (200) 计算用于一个  $16 \times 16$  分区、两个  $16 \times 8$  分区、两个  $8 \times 16$  分区, 或四个  $8 \times 8$  分区的运动矢量, 其中, 每个  $8 \times 8$  可以进一步划分成两个  $8 \times 4$  分区、两个  $4 \times 8$  分区, 或四个  $4 \times 4$  分区。或者, 编码器 (200) 计算用于更高空间分辨率视频 (例如 CIF) 的大块 (例如  $32 \times 32$ ) 的运动矢量, 以便当仅解码更低空间分辨率视频 (例如 QCIF), 便于使用运动矢量。另外, 编码器 (200) 计算用于其他大小块的运动信息。

[0063] 运动估计 (220) 和 MCTF 的运动精度由实现而定。例如, 运动估计 (220) 和 MCTF 使用四分之一像素运动矢量。另外, 运动估计 (220) 和 MCTF 使用具有一些其他精度的运动信息。例如, 当仅解码 QCIF 视频时, 为便于使用四分之一像素运动矢量, 编码器 (200) 计算用于 CIF 视频的半像素运动矢量和 / 或用于 4CIF 视频的整数像素运动矢量。

[0064] 运动估计 (220) 和 MCTF 的方向可以向前、向后、双向或内部。在一些实现中, 编码器 (200) 将每宏块的分区的运动估计 (220) 和补偿方向设置成向前、向后、双向或内部。另外, 编码器 (200) 在一些其他级设置方向和 / 或不允许一些方向 (例如内部)。

[0065] 在一些实现中, 可以填补基准帧 (例如通过简单重复填补) 以允许除基准帧外的运动估计 / 补偿。基准帧上的子像素内插使用双线性滤波器、双三次滤波器、Sinc 滤波器或一些其他滤波器, 生成用于局部运动估计 / 补偿的样本值。在一些实现中, 编码器使用重叠块运动补偿 (“OBMC”)。另外, 将运动矢量范围限制到基准帧边界内和 / 或不使用 OBMC。

[0066] 下面, 使用 5/3 双正交小波结构, 描述 MCTF 的几个例子。另外, 编码器 (200) 执行另一数字和 / 或时间子带变换类型 (210) (例如 Haar) 和 / 或运动估计 (220)。图 2 表示时间变换全空间分辨率视频的空间域 MCTF。另外, 编码器 (200) 包括在时间分解 (210) 后的一个或多个空间子带变换, 以及 MCTF 是来自初始空间分解的空间子带上的带内 MCTF,

[0067] 在时间变换 (210) 后, 编码器 (200) 应用一个或多个二维 (“2D”) 空间子带变换 (230) 以便进一步分解信号。图 2 表示用于 2D 空间分解的二级输出, 其中, 进一步水平和垂直分解具有水平和垂直的低空间分辨率的子带。例如, 2D 空间子带变换是二重离散小波变换 (“DWT”)。小波滤波器可以是例如 9/7DWT 滤波器或 5/3DWT 滤波器。另外, 编码器 (200) 执行另一数字 (例如 3 或 4) 空间子带变换和 / 或执行另一类型的空间子带变换 (230)。

[0068] 在时间变换 (210) 和空间变换 (230) 后, 信号组织成不同空间 - 时间子带, 对应

于输入视频 (205) 的不同空间 - 时间分辨率。编码器 (200) 熵编码 (240) 子带。例如, 编码器 (200) 在一些实现中, 使用具有最佳截断的 3D 嵌入块编码 (“EBCOT”)。JPEG200 使用 EBCOT 的 2D 形式, 以及在那些实现中, 编码器 (200) 基本上将 2D EBCOT 扩展成 3D 编码。由时间和空间变换生成的每个子带划分成独立地编码的 3D 编码块。对每个编码块, 编码器 (200) 使用位平面编码和基于上下文的算术编码。另外, 编码器 (200) 使用用于子带信息的熵编码的一些其他形式。

[0069] 编码器 (200) 还编码 (250) 运动信息 (例如运动矢量、运动模式选择信息)。例如, 编码器 (200) 使用一些形式的熵编码, 诸如简单可变长度编码以便熵编码运动矢量信息、宏块的分区, 以及运动估计 / 补偿的方向。对运动矢量, 编码器 (200) 编码运动矢量和其预测值间的差。使用来自先前编码 / 解码的运动矢量或使用一些其他运动矢量预测技术, 可以计算运动矢量预测值另外, 编码器 (200) 使用用于运动信息的其他和 / 或另外的编码 (250) 技术。

[0070] 通过子带变换 (210, 230) 的属性, 编码器 (200) 固有地支持空间可缩放性和时间 / 帧速率可缩放性。此外, 通过子带系数的位平面编码, 编码器 (200) 通常还支持质量 / 信号噪声比 (“SNR”) 可缩放性。

[0071] 这些可缩放性还用于易出错网络, 诸如 Internet 或无线网络上的视频传输。可以编码不同子带, 用于通过不同级误差校正或传输错误恢复能力。或者, 可以以不同优先级传送不同子带, 其中网络支持优先传输。

[0072] 各种可缩放性还帮助不同设备共享相同压缩视频流, 即使不同设备具有不同存储器资源、计算功能、显示大小、可存取带宽等等。用这种方式, 简化了以不同比特率, 编码视频的任务 (205)。

[0073] 多级时间分解的一个优点是在大的范围中, 采用视频信号内的时间冗余。前几级时间子带变换能采用相近帧中的时间冗余, 以及最后几级时间子带变换能利用更长距离上的帧中的相关性。

[0074] 图 4 表示具有四层时间分解和一层空间分解的示例性 3D 子带视频编码器 (400)。基本上, 图 4 表示图 2 中时间子带变换 (210) 的一个可能扩展以及还表示用于不同级时间分解的数据流。图 4 还介绍用于时间和空间分解的新符号。

[0075] 在图 4 中, 编码器 (400) 将四级时间子带变换应用于具有每秒  $f$  帧的帧速率的视频。时间子带变换的每级  $T_i$  将输入到那一级的信号划分成两个时间子带 - 时间低通子带和时间高通子带。然后, 每个子带具有那一级的输入视频信号的帧速率的一半。在图 4 中, 术语  $t-L_i$  和  $t-H_i$  分别表示在第  $i$  级时间变换  $T_i$  后, 输出低通子带和高通子带。 $t-L_i$  和  $t-H_i$  具有帧速率  $f/2^i$ 。在图 4 中, 通过时间子带变换  $T_{i+1}$ , 几次迭代分解信号  $t-L_i$  来采用远程相关性和允许能由解码器解码更低速率视频。

[0076] 为通过简单数值示例说明, 假定输入视频是每秒 30 帧 (“fps”) 的 CIF 视频。来自第一时间分解的输出低通子带  $t-L_1$  是 15fps 视频, 与来自第一时间分解的输出高通子带  $t-H_1$  一样。来自第二时间分解的输出子带  $t-L_2$  和  $t-H_2$  为 7.5fps 视频, 以及来自第三时间分解的输出子带  $t-L_3$  和  $t-H_3$  为 3.75fps 视频。最后, 来自第四时间分解的输出子带  $t-L_4$  和  $t-H_4$  为 1.875fps 视频。在该例子中, 空间分辨率是用于所有输出低通和高通子带的 CIF。

[0077] 以不同级估计帧中的运动信息, 以及沿当前级的运动轨迹, 执行每级的时间变换。

由此,每级时间分解产生随后编码的运动矢量。在图 4 中,存在对应于四级时间变换的四个运动矢量集。

[0078] 然后,通过将各个输入子带划分成低通水平、低通垂直 (“LL”)、低通水平、高通垂直 (“LH”)、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”) 子带的单级 2D 空间子带变换  $S_1$ , 分解输出时间低通子带  $t-L_4$  和输出时间高通子带  $t-H_1$ ,  $t-H_2$ ,  $t-H_3$  和  $t-H_4$ 。另外,编码器 (400) 包括时间分解级前后的空间分解的其他和 / 或另外级,或编码器 (400) 使用不同空间分解图。

[0079] 续用于图 4 的数例,低通子带  $t-L_4$  具有 CIF 空间分辨率。通过单级 2D 空间子带变换  $S_1$ , 将时间子带  $t-L_4$  分解成 LL 子带、LH 子带、HL 子带和 HH 子带。每个 LL、LH、HL 和 HH 子带的每一个具有 QCIF 分辨率。类似地,时间高通子带  $t-H_1$ ,  $t-H_2$ ,  $t-H_3$  和  $t-H_4$  的每一个从 CIF 分辨率开始以及分解成 QCIF 分辨率 LL、LG、HL 和 HH 子带。

[0080] 然后,熵编码和输出空间分解的子带。

[0081] B. 示例性视频解码器

[0082] 图 3 示例说明通用 3D 子带视频解码方案 (300)。解码器 (300) 接收压缩的视频信息 (395) 和将视频图像序列 (305) (例如逐行视频帧、隔行视频帧或隔行视频帧场) 产生为输出。视频解码器的特定实施例通常使用通用解码器 (300) 的变化或互补版本。

[0083] 解码器 (300) 熵解码 (340) 用于一些或全部空间 - 时间子带的信息 (395)。视频信息 (395) 仍然组织成不同空间 - 时间子带,其对应于输入视频 (205) 的不同空间 - 时间分辨率。用这种方式,解码器 (300) 可以在不同空间 - 时间分辨率间选择,忽视压缩比特流 (395) 部分,或解码器 (300) 可以简单地解码 (340) 解码器 (300) 实际上接收的压缩比特流 (395) 的子集。解码器 (300) 基本上执行在用于解码器 (300) 解码的信息的编码器 (200) 中执行熵编码 (240) 的逆操作。例如,解码器 (300) 接收和解码 (340) 使用 3D EBCOT 编码的信息。另外,解码器 (300) 使用用于一些或所有子带信息的一些其他形式的熵解码。

[0084] 当解码器 (300) 执行逆 MCTF (“IMCTF”) 时,在执行在编码器 (200) 中执行的编码 (250) 的逆操作的许多情况下,解码器 (300) 解码 (350) 运动信息 (例如运动矢量、运动模式选择信息)。例如,解码器 (300) 使用一些形式的熵解码,诸如解码简单可变长度编码。对运动矢量,解码器 (300) 使用中值预测或一些其他运动矢量预测技术,计算运动矢量预测,然后,解码器 (300) 将运动矢量预测和运动矢量微分结合。另外,解码器 (300) 使用其他和 / 或另外的解码 (350) 技术。

[0085] 解码器 (300) 应用一个或多个 2D 逆空间子带变换 (330) 以便空间重构视频。图 3 表示用于两级 2D 逆空间分解的输入和输出。例如,2D 逆空间子带变换是二元逆 DWT (“IDWT”)。另外,解码器 (300) 执行另一数字或类型的逆空间子带变换 (330)。

[0086] 解码器 (300) 通过逆时间子带变换 (310),进一步重构该视频信号。图 3 表示由低分辨率子带和高分辨率子带,用于单级时间重构的输出,但时间合成可以包括 4,5 或一些其他的综合。

[0087] 在图 3 中,逆空间变换 (310) 使用 IMCTF,以及解码器 (300) 使用用于视频信号的解码 (350) 的运动信息,执行运动补偿。通常,运动信息和 IMCTF 的属性 (例如用于运动补偿的分区、运动矢量精度、运动补偿的方向、使用基准帧填补、子像素内插、使用 OBMC) 与在相应的编码器 (200) 中相同以及在压缩视频比特流 (395) 中发信号。IMCTF 的几个例子如

下所述。另外,解码器(300)执行其他数字和/或类型的逆时间子带变换(310)和/或运动补偿。图3表示空间域IMCF,其中使用逆时间变换,重构全空间分辨率视频。另外,解码器(300)执行带内MCTF。

[0088] 返回到图4的编码器(400),相应的解码器能仅从 $t-L_i$ 重构 $f/2^i$ 帧速率视频。或者,解码器能从 $t-L_i$ 和 $t-H_i$ 两者重构 $f/2^{i-1}$ 帧速率视频(包括后续时间高通子带信息 $t-H_{i+1}$ 等等)。

[0089] 图5表示在编码器中,具有四层时间分解和两层空间分解,以及解码器中相应数目的逆变换的示例性3D子带视频编码器-解码器(“编解码器”)系统(500)。基本上,图5表示图2和3的编码器(200)和解码器(300)的其他可能版本。图5还介绍用于时间和空间分解以及它们的逆操作的另一、更简单的符号。

[0090] 在图5中,编码器将四级时间子带变换 $T_i$ (即 $T_1, T_2, T_3$ 和 $T_4$ )应用于视频以及将两级空间子带变换 $S_i$ (即 $S_1$ 和 $S_2$ )应用于时间分解的结果。然后,编码器执行熵编码E。解码器执行熵解码 $E^{-1}$ 和对熵解码的结果执行两级逆空间子带变换 $S_i^{-1}$ (即 $S_2^{-1}$ 和 $S_1^{-1}$ )。然后,解码器将四级逆时间子带变换 $T_i^{-1}$ (即 $T_4^{-1}, T_3^{-1}, T_2^{-1}$ 和 $T_1^{-1}$ )应用于空间重构的结果。

[0091] III. 运动补偿时间滤波

[0092] 为提高编码效率,在时间分解中应用运动对准的时间变换(在此另外称为运动补偿的时间滤波(“MCTF”))以便有效地去除时间冗余。MCTF的主要概念是对准不同帧中的像素同时应用时间分解。与其他编码方案相比,运动对准的3D编码方案能提供可缩放性特性和非受损的,或甚至更好编码效率。

[0093] 在各个MCTF方案中,基于提升的MCTF提供灵活和高效的实现。图6示例说明在用于5/3双正交小波的基于提升的MCTF方案的输入视频帧、时间低通帧和时间高通帧间的关系。

[0094] 图6表示输入到MCTF中的四个初始的、索引视频帧的像素列。(输入帧本身可以是例如来自MCTF的先前级的时间低通子带)。图6中基于提升的小波变换包含两个基本级:运动补偿预测(“MCP”)级和运动补偿更新(“MCU”)级。

[0095] 在预测级中,对奇数帧(例如帧 $I_{2i+1}$ ),来自一个或多个相邻偶数(例如 $I_{2i}$ 和 $I_{2i+2}$ )的运动补偿表示用于奇数帧的像素的运动补偿的预测。高通帧(例如 $H_i$ )基本是奇数帧和奇数帧运动补偿预测间的差值。

[0096] 图7示例说明在编码器中,具有提升的传统MCTF,以及图8示例说明解码器中相应的IMCTF。为了一致性,在图7和8以及应用的其他地方,偶视频帧是用于运动补偿预测的基准和用于低通帧的形成,以及奇视频帧是MCP的主体和用于形成高通视频帧。另外,奇数和偶数帧的角色可以切换。

[0097] 在图7的MCTF(700)中,偶数视频帧用作MCP(710)中的基准。对特定奇数视频帧 $I_{2i+1}$ ,编码器由相邻的偶数帧 $I_{2i}$ 和 $I_{2i+2}$ 的一个或两个,计算运动补偿预测。预测和初始奇数帧 $I_{2i+1}$ 间的差值是时间高通帧 $H_i$ 。

[0098] 高通帧 $H_i$ 用作MCU(720)中的参考。对特定偶数视频帧 $I_{2i}$ ,编码器由用于相邻奇数视频帧的 $H_{i+1}$ 和 $H_i$ 中的一个或两个,计算运动补偿的更新。更新和初始偶数帧 $I_{2i}$ 的组合是时间低通帧 $L_i$ 。

[0099] 例如,假定帧 $I_0, I_1, \dots, I_{2n-1}$ 的视频序列将由图7的MCTF(700)处理。预测模块

(710) 由连续偶数帧  $I_{2i}$  和  $I_{2i+2}$ , 计算用于奇数帧  $I_{2i+1}$  的预测  $P(I_{2i+1})$  如下。

$$[0100] \quad P(I_{2i+1}) = \frac{1}{2}(\text{MC}(I_{2i}, \text{MV}_{2i+1 \rightarrow 2i}) + \text{MC}(I_{2i+2}, \text{MV}_{2i+1 \rightarrow 2i+2})) \quad (1)$$

[0101] 其中,  $\text{MV}_{2i+1 \rightarrow 2i}$  是指从帧  $I_{2i+1}$  到帧  $I_{2i}$  的运动矢量,  $\text{MV}_{2i+1 \rightarrow 2i+2}$  具有类似含义, 以及  $\text{MC}()$  是指由相邻偶数帧生成至少部分当前奇数帧的预测的运动补偿。(方程式 (1) 是指示例说明用于宏块的 MCP、宏块分区或应用运动矢量的奇数帧的其他部分, 以及对奇数帧的其他宏块、宏块分区等等, 重复这种 MCP)。在一些情况下, 当计算运动补偿预测 (如在 Haar 变换中) 时, 编码器仅考虑相邻偶数帧的一个 (不是两个), 以及方程式 (1) 中 MC 贡献和加权的相应改变, 或编码器使用内压缩, 编码部分或所有奇数帧。结合嵌入的基层编解码器, 编码器可以将除相邻奇数帧的帧用作基准帧, 如下所述。

[0102] 然后, 预测模块 (710) 计算高通帧  $H_i$  如下。

$$[0103] \quad H_i = I_{2i+1} - P(I_{2i+1}) \quad (2)$$

[0104] 在更新步骤 (720) 中, 遵循预测步骤 (710) 以完成一级时间变换, 编码器由相邻高通帧, 计算用于奇数帧  $I_{2i}$  的更新  $U(I_{2i})$  如下。

$$[0105] \quad U(I_{2i}) = \frac{1}{4}((\text{MC}(H_{i-1}, \text{MV}_{2i \rightarrow 2i-1}) + \text{MC}(H_i, \text{MV}_{2i \rightarrow 2i+1})) \quad (3)$$

[0106] 其中,  $\text{MC}()$  是指由相邻高通帧, 生成用于偶数帧的更新的至少一部分的运动补偿。作为简单近似, 用于 MCU 的运动矢量被导出作为用于 MCP 的相应运动矢量的逆。(方程式 (3) 是指表示用于宏块、宏块分区, 或应用运动矢量的其他部分的偶数帧, 以及对其他宏块、宏块分区等等, 重复这种 MCU)。在一些情况下, 通过方程式 (3) 中的 MC 贡献和加权的相应变化, 当计算运动补偿更新时, 编码器仅考虑高通帧的一个 (不是两个), 或编码器不考虑高通帧的任何一个。

[0107] 然后, 编码器生成低通帧如下。

$$[0108] \quad L_i = I_{2i} + U(I_{2i}) \quad (4)$$

[0109] 在图 8 的 IMCTF (800) 中, 将高通帧  $H_i$  用作 MCU (820) 中的基准。对特定偶数视频帧  $I_{2i}$ , 解码器由用于相邻奇数视频帧的高通帧  $H_{i-1}$  和  $H_i$  的一个或两个, 计算运动补偿的更新。用于偶数帧  $I_{2i}$  的更新和相邻时间低通帧  $L_i$  间的差值是重构的偶数帧  $I_{2i}$ 。

[0110] 当重构时, 将偶数视频帧用作 MCP (810) 中的基准。对特定奇数视频帧  $I_{2i+1}$  解码器由相邻重构的偶数帧  $I_{2i}$  和  $I_{2i+2}$  的一个或两个, 计算运动补偿预测。用于奇数帧  $I_{2i+1}$  的预测和相应时间高通帧  $H_i$  的组合是重构的奇数帧  $I_{2i+1}$ 。

[0111] 在编码器中, 预测步骤通常尝试最小化或降低编码高通帧 (在时间分解和熵编码中) 连同用于预测的运动矢量所需的比特率。因此, 高通帧  $H_i$  基本上是来自从相关的偶数索引的初始视频帧的相应的奇数索引的输入视频  $I_{2i+1}$  的双向 MCP (或向前 MCP, 或向后 MCP, 视具体情况而定) 的剩余量。然后, 用预测剩余量, 更新初始偶数索引帧  $I_{2i}$  作为低通帧。

[0112] 高通帧的能量由运动模型的成功而定, 运动模型由运动的复杂度、帧间的运动均匀性, 以及运动模型本身而定。如果运动预测准确, 预测余量通常具有小的量值。即使当真正俘获运动时, 然而, 更新步骤仍然有用, 因为它们能降低由于沿运动轨迹的低通滤波的噪声和混叠。通常, 更新步骤便于移除低通帧中的高频部分, 从而提高编码效率。

[0113] 如果运动预测不精确 (例如由于复杂度、不可预测运动或运动模型本身的缺点),

对象特征可以未对准,引起几个问题。不精确运动预测会导致多个边缘以及增加高通子带帧中的能量。这不仅损害用于编码高通子带本身的压缩性能,而且在更新步骤期间,还将寄生边缘和轮廓增加到时间低通帧上。

[0114] 一般地说,当允许时间可缩放性时,更新步骤能影响解码性能。特别地,由于有限信道带宽和设备能力,当部分或完全丢失一些时间高通帧时,解码器的问题是可能的。(通常,通过保留时间低通子带同时丢弃时间变换域中的高通子带,实现 IMCF 中的时间可扩展性。)当运动预测不有效时,高通帧在编码中的更新步骤期间,对低通帧(即低帧速率视频)产生伪像。当仅由时间低通帧重构视频时,通过解码器中的 MCU,未去除高通贡献中的任何一个。或者,如果粗糙地量化时间高通帧信息,时间低通帧的质量会受损,因为在解码器中的 MCU,仅近似地去除来自高通帧的贡献。从而不能去除低通帧中的重影,即使当将更多位分配到低通帧的编码。

[0115] 一种可能的解决方案是在编码器中的基于提升的 MCTF 中,跳过更新步骤。这能显著地降低整个编码性能,然而,因为更新步骤能降低由于沿运动轨迹的低通滤波的噪声和混叠。因此,更新步骤能通过时间高通子带和时间低通子带中扩散失真,减少整个失真。

[0116] IV. MCTF 和 IMCTF 中的自适应更新阶段

[0117] 在一些实施例中,可缩放视频编码器自适应地执行基于提升的 MCTF 中的更新步骤和/或可缩放视频解码器自适应地执行 IMCTF 中的更新步骤。通过利用人的视觉系统的属性,自适应更新步骤提高由时间低通子带解码的低帧速率视频的可感知视觉质量。同时,与传统的更新步骤相比,自适应更新步骤维持由时间低通和时间高通子带解码的更高帧速率的令人满意的性能。

[0118] A. 示例性编码器和解码器

[0119] 自适应更新方案实现 (1) 尽可能增加初始更新信息,以及 (2) 同时,减少由更新步骤引入的视觉伪像的两种矛盾目标。

[0120] 图 9 描述在编码器中,具有自适应更新操作的 MCTF 框架 (900),以及图 10 描述在解码器中,具有自适应更新操作的 IMCTF 框架 (1000)。例如,框架 (900,1000) 包含在编码器和/或解码器中,诸如参考图 2-5 所述。另一种方式,自适应更新操作包含在另一编码器和/或解码器中。

[0121] 参考图 9, MCTF 框架 (900) 包括预测 (910) 和更新 (920) 阶段。这些阶段基本上参考图 7 所述处理。

[0122] MCTF 框架中 (900) 使用人的视觉系统 (“HVS”) 模型 (930) 来在视觉质量方面,估计来自预测步骤的信号。例如,编码器根据 HVS 模型 (930),估计偶数帧中的信息。在一些实现中,HVS 模型 (930) 使用最小可察觉差 (“JND”) 度量。在其他实现中,HVS 模型 (930) 使用其他可感知标准。

[0123] HVS 模型 (930) 向根据阈值信息,估计和应用更新 (920) 阶段的结果的阈值 (940) 阶段提供阈值信息。例如,HVS 模型 (930) 提供根据模型估计的每个偶数帧的像素的阈值。在更新操作中阈值和它们的用途的各个例子描述如下。另外,HVS 模型 (930) 提供不同阈值信息。

[0124] 阈值 (940) 阶段的结果是在尤其可感知失真的图像部分中,限制更新信息,而对其他部分的图像,不限制更新信息。从而根据人的感觉,在将更新信息应用于偶数帧前,自

适应移除或限制更新信息。

[0125] 参考图 10, IMCTF 框架 (1000) 也包括预测 (1010) 和更新 (1020) 阶段。这些阶段基本上如参考图 8 所述处理。

[0126] IMCTF 框架 (1000) 使用 HVS 模型 (1030) 来根据视觉质量估计信号。例如, 解码器根据 HVS 模型 (1030), 估计低通帧中的信息。在一些实现中, HVS 模型 (1030) 使用最小可察觉差 (“JND”) 度量。在其他实现中, HVS 模型 (1030) 使用其他感知标准。

[0127] HVS 模型 (1030) 将阈值信息提供到阈值 (1040) 阶段, 其根据阈值信息, 估计和应用更新 (1020) 阶段的结果。例如, HVS 模型 (1030) 提供根据模型估计的、每低通帧的像素的阈值。阈值和它们的用途的各个例子如下所述。另外, HVS 模型 (1030) 提供不同阈值信息。

[0128] 比较图 9 和 10, 将编码器端的 HVS 模型 (930) 应用于偶数帧, 而将解码器端的 HVS 模型 (1030) 应用于低通帧。即使在编码器端和解码器端使用相同的 HVS 模型 (930, 1030), 阈值可以稍微不同。由于对相应的偶数和低通帧, 阈值很可能非常类似, 以及由于编码器和解码器间的偏差对帧是局部的 (与随时间累积和变得更糟相比), 这是可容忍的。

[0129] 此外, 在图 9 和 10 的框架 (900, 1000) 中, 在每级 MCTF/IMCTF, 计算阈值。同样地, 即使对时间相应帧, 在不同 MCTF/IMCTF 级, 阈值可以不同。

[0130] 因此, 在一些实现中, 编码器端 MCTF (900) 和解码器端 IMCTF (1000) 在 HVS 模型 (930, 1030) 中应用相同的 JND 度量。没有与阈值或阈值操作有关的编码开销, 因为在编码器和解码器中, 独立地计算和应用阈值。另外, 编码器信号阈值信息或对解码器的判决应用了与编码器不同的 HVS 模型。

[0131] B. 示例性 HVS 模型和阈值操作

[0132] 编码器中的自适应更新步骤可以通常表示如下:  $L_i = I_{2i} + f(U_{2i})$ 。函数  $f()$  是将用于偶数帧  $I_{2i}$  的更新  $U_{2i}$  的自适应函数。  $L_i$  是最终所得的低通帧。

[0133] 自适应函数利用人的视觉模型的结束和开发。已经实施广泛研究来开发基于人的视觉系统的计算模型。在各个模型中, 感知标准基于 HVS 的灵敏度与空间和时间域中的可变对比度和亮度变化不一致的观察。

[0134] 在 HVS 的许多计算模型中, JND 广泛用在感觉编码和图像水印中。JND 使用定义为增加干扰直到对人类来说, 它变得恰可辨别为止的亮度边缘的幅度的函数的可见度阈值。对于更多信息, 见 A. N. Netravali 和 B. Prasada, “Adaptive Quantization of Picture Signals Using Spatial Masking”, Proc. IEEE. vol. 65, pp. 536-548 (1997 年 4 月)。JND 与 HVS 的纹理掩模属性密切相关。根据该准则, 在平坦或缺少纹理的区域, 噪声更可见, 以及在具有边缘和纹理的区域, 噪声不太可见。

[0135] JND 阈值是图像相关的。只要更新信息仍然在 JND 阈值下, 在低通帧中实现“更新剩余量”透明度。同样地, JND 与在自适应更新方案中的 HVS 模型一样工作良好。

[0136] 在自适应更新方案的一种实现中, JND 模型定义如下。

$$[0137] \quad \text{JND}_x(i, j) = 1 - \frac{1}{1 + \theta \sigma_x^2(i, j)} \quad (5)$$

[0138] 其中,  $\sigma_x^2(i, j)$  表示在具有坐标  $(i, j)$  的像素上集中的窗口中的图像  $x$  的局部方差。JND 可以由仅亮度样本计算以便降低计算复杂度。或者, 可以由亮度和色度样本计算



JND。窗口的大小由实现而定。一种示例性窗口大小为  $3 \times 3$ ，以及另一个是  $5 \times 5$ 。 $\theta$  是能为特定图像选择的调谐参数。这与噪声可见度函数值是相同的，假定图像是非平稳 Gaussian。调谐参数  $\theta$  在 JND 定义中，起对比度调整的作用。为使调谐参数  $\theta$  与图像相关，可以计算如下

$$[0139] \quad \theta = \frac{D}{\sigma_{x_{\max}}^2} \quad (6)$$

[0140] 其中  $\sigma_{x_{\max}}^2$  是用于指定图像的最大局部方差，以及  $D \in [50, 100]$  是实验上确定的参数。根据该 JND 定义，由于  $\sigma_x^2(i, j)$  很小，在平坦区域中，JND 值很小，以及反之亦然。

[0141] 表示编码器中的自适应更新的更精确方法是：

$$[0142] \quad L_i = I_{2i} + f(I_{2i}, U_{2i}) \quad (7)$$

[0143] 其中

$$[0144] \quad f(I_{2i}, U_{2i}) = \begin{cases} U_{2i} & |U_{2i}| < \text{JND}_{I_{2i}} \cdot S \\ \text{JND}_{I_{2i}} \cdot S & U_{2i} \geq \text{JND}_{I_{2i}} \cdot S \\ -\text{JND}_{I_{2i}} \cdot S & U_{2i} \leq -\text{JND}_{I_{2i}} \cdot S \end{cases} \quad (8)$$

[0145]  $\text{JND}_{I_2}$  是如上定义的  $\text{JND}_x(i, j)$  的例子，以及  $S$  表示实现相关的强度因素，例如 12.5。JND 函数与可视掩模相关的局部帧特性自适应。如果局部强度方差大，JND 相对大（根据方程式 (5) 接近 1），表示繁变或细节区域。另一方面，如果局部强度方差小，JND 相对小（根据方程式 (5) 接近 0），表示平坦区域。由将在编码器中更新的偶数指数帧和解码器中的偶数指数帧（实际上，其低通图像近似）来计算它。因此，由于高通帧，自适应更新能有效地避免重影伪像以及提高用于时间可缩放性的编码性能。

[0146] 如上所述，在一些实现中，编码器和解码器使用相同的 JND 度量和阈值（在方程式 (8) 中，随指数值变化等等），以及没有编码和递送到用于自适应更新操作的开销。尽管编码器和解码器在不同图像上操作（编码器上的初始帧和解码器上的重构低通版本），经验结果已经显示解码器上的最终更新掩码非常接近编码器。

[0147] 另外，编码器和 / 或解码器使用 JND 的另一定义、另一 HVS 模型和 / 或其他阈值操作。例如，可以使用其他和 / 或另外的参数。

[0148] C. 示例性自适应更新技术

[0149] 图 11 至 13 示例说明用于自适应更新操作的技术 (1100, 1200, 1300)。编码器，诸如参考图 9 所述，执行图 11 和 12 所述的 (1100, 1200)，以及解码器，诸如参考图 10 所述，执行如图 11 和 13 所示的技术 (1100, 1300)。另外，另一编码器或解码器执行技术 (1100, 1200, 1300)。

[0150] 图 11 表示用于自适应更新操作的简化、通用的技术 (1100)。编码器或解码器计算 (1120) 用于自适应更新的阈值。然后，编码器或解码器执行 (1160) 自适应更新操作。

[0151] 图 12 表示用于编码器中的自适应更新操作的详细技术 (1200)，包括用于 MCP 的另外的细节。

[0152] 对将计算低通帧的当前偶数帧，编码器计算 (1220) 阈值。例如，编码器计算用于当前偶数帧的像素的 JND，如在前一章节中所述。另外，编码器使用一些其他度量计算 (1220) 阈值。

[0153] 对 MCP，编码器由当前偶数帧计算 (1230) 一个或多个运动补偿预测，例如向前和 /

或向后预测相邻奇数帧。编码器可以由其他偶数帧,计算用于相邻奇数帧的一个或多个其他运动补偿预测(或可以由较早的偶数帧先前计算的一个预测)。然后,编码器由运动补偿预测和它们的相应的相邻奇数帧,计算(1240)一个或多个高通帧。

[0154] 接着,编码器由一个或多个高通帧,计算(1250)用于当前偶数帧的运动补偿更新。例如,编码器相反地应用在MCP(1230)期间由当前偶数帧计算的运动信息。另外,编码器使用不同机制来计算(1250)更新。

[0155] 然后,编码器对当前偶数帧执行(1260)自适应更新,导致低通帧。例如,如在先前章节中,编码器使用运动补偿更新和JND阈值,执行自适应阈值操作。另外,编码器使用不同阈值机制,执行(1260)自适应更新。

[0156] 编码器确定(1290)是否继续MCTF,以及如果是,继续下一偶数帧。

[0157] 图13表示用于解码器中的自适应更新操作的详细技术(1300),包括用于以不同帧速率解码的单独的路径。

[0158] 解码器确定(1310)用于解码的时间分辨率。可以根据由解码器实际上接收多少信息(例如在易于出错信道上的时间低通信息)确定的,或用一些其他方式确定的用户级设定值或一些其他设定值,确定时间分辨率。另外,解码器可以具有仅一个解码路径。例如,一个解码器可以包括低帧速率路径,而另一解码器包括高帧速率路径,适合各个解码器的资源。

[0159] 在低帧速率路径,解码器由所接收的低通帧信息,(例如将低通帧用作当前帧),计算(1312)当前偶数帧。不执行自适应更新操作,因为忽略或不接收高通信息和运动信息。即使在缺少这些信息的情况下,由于在编码期间自适应更新操作提高了当前偶数帧的质量(与传统的IMCTF相比),因为在编码期间限制了将显著的失真添加到偶数帧。

[0160] 在高帧速率路径中,对当前偶数帧,解码器由解码器接收的相应的低通帧,计算(1320)阈值。例如,解码器计算用于低通帧的像素的JND,如在前章节中所述。另外,解码器使用一些其他度量,计算(1320)阈值。

[0161] 解码器从一个或多个高通帧,计算(1330)用于当前偶数帧的运动补偿更新。例如,解码器相反地应用由当前偶数帧,在MCP(1230)期间计算的运动信息。另外,解码器使用不同机制来计算(1330)更新。

[0162] 然后,解码器对低通帧执行(1360)自适应更新以便重构当前偶数帧。例如,解码器使用更新和JND阈值,执行自适应阈值操作,如在前章节中所述。另外,解码器使用不同阈值机制,执行(1360)自适应更新。

[0163] 对MCP,解码器从重构的当前偶数帧,计算(1370)一个或多个运动补偿预测,例如向前和/或向后预测相邻奇数帧。(已经由从早先的偶数帧先前计算了用于相邻奇数帧的其他运动补偿预测)。然后,解码器从用于奇数帧和相应的高通帧的运动补偿的预测,计算(1380)重构的奇数帧。

[0164] 解码器确定(1390)是否继续IMTF,以及如果是,继续下一偶数帧。

[0165] 图12和13所示的各个操作可以拆分、与其他操作组合或记录。例如,尽管图12表示并行的某些操作,相反,可以串行地执行这些操作。

[0166] V. 3D子带视频编码和解码中嵌入的基层编解码器

[0167] 在一些实施例中,可缩放视频编码器包括嵌入的基层编解码器以及可缩放视频解

码器包括嵌入的基层解码器。基本上,由嵌入的基层编解码器产生的比特流用作 3D 子带视频编码比特流的低比特率基层。在一些可缩放视频编码 / 解码系统的上下文中,例如,基层以用于系统的最小空间、时间和 SNR 分辨率提供视频,在此之后,增加另外的增强层以便提供用于系统的达最大空间、时间和 SNR 分辨率的可缩放性。(编码和解码中的基层的精确作用在不同实施例中是不同的)。例如,3D 子带视频编码器包括嵌入的视频编解码器以及 3D 子带视频解码器包括相应的嵌入视频解码器。

[0168] 对许多类型的嵌入编解码器,这改进了以低比特率的编码性能,其中在传统的 3D 子带编码中编码运动信息所花的比特消耗了可用比特率的无效部分,以及在比较中,嵌入视频编解码器更有效。在一些情况下,嵌入的基层编解码器提供诸如闭环时间预测、环内滤波和码率失真优化,而没有与 3D 子带编码有关联的约束的优点。同时,保持以中间和较高比特率的 3D 子带编码的优点(例如空间和时间可缩放性)。

[0169] 将编解码器嵌入 3D 子带视频编码中的另一优点是它提供与嵌入的编解码器和相应的解码器兼容。例如,如果嵌入 MPEG-1 编解码器,具有 MPEG-1 解码能力的任何接收机能解码压缩视频的粗略版本(即使 MPEG-1 编解码器本身不提高编码性能)。如果嵌入 H. 264 或 VC-1 编解码器,除兼容性外,这提高了以低比特率的编码性能。

[0170] 在此所述的技术和工具不限于具有在性能和 / 或兼容性方面优点的系统。除性能和兼容性外,将基层编解码器嵌入可缩放视频编码器和将基层解码器嵌入可缩放视频解码器可以具有其他优点。

[0171] A. 示例性编码器和解码器

[0172] 图 14 至 19 分别示例说明用于将基层编码器或解码器嵌入编码器或解码器中的三个不同方案。在编码器中,用于基层的编解码器嵌入时间子带变换中,以及时间子带变换后接空间子带变换。通过解码器中的相应的变化,另外的子带变换可以放在时间子带变换前和 / 或后。通过解码器中可能的相应变化,基层编解码器可以位于各种变换中的不同位置中。

[0173] 通常,嵌入编解码器的位置设置用于基层视频的帧速率和空间分辨率。为了简化图 14 对 19 所示的三个方案间的比较,将基层分辨率设置成帧速率的四分之一以及初始视频的空间分辨率的一半。因此,如果初始视频为 30fps CIF 视频,例如,基层视频为 7.5fps QCIF 视频。

[0174] 同样地,在图 14, 16 和 18 的编码器中,基层编解码器在两个时间子带变换后,以及在解码器 15, 17 和 19 中,基层解码器在最后两个相反的时间子带变换前。在时间变换中改变基层编解码器和解码器的位置是改变基层帧速率的一种方法。例如,尽管图 14, 16 和 18 表示嵌入基层模块以便接受  $t-L_2$  子带作为输入,或者,基层模块放在  $t-L_1$  子带变换或另一子带变换后。通常,时间变换中的位置由基层视频所需的帧速率而定。

[0175] 在图 14, 16 和 18 中,通过按一些任意因素,采样操作或空间子带变换操作(与嵌入编解码器或解码器相邻),实现用于基层的空间分辨率变化。改变采样 / 空间子带变换操作的比率是改变基层空间分辨率的一种方法。

[0176] 图 14, 16 和 18 包括时间子带变换模块 T、空间子带变换模块 S 和 MV 编码和熵编码模块,以及来自模块的输入和输出。图 15, 17 和 19 包括逆时间子带变换模块  $T^{-1}$ 、逆空间子带变换模块  $S^{-1}$ , 以及 MV 解码和熵解码模块,以及来自模块的输入和输出。通常,这些模块

和它们的输入 / 输出的作用如参考图 2 至 5 所述。

[0177] 图 14, 16 和 18 所述的基层编解码器可以是产生符合标准的比特流 (例如 MPEG-1, MPEG-2、MPEG-4、H. 261、H. 263、H. 264/AVC 或 VC-1), 以及图 15, 17 和 19 中的相应的基层编解码器。或者, 基层编解码器 / 解码器能是商业产品 (例如 WMV8, WMV9)。一般来说, 基层编解码器 / 解码器能是具有所需性能和 / 或兼容性属性的任何可用编解码器 / 解码器。

[0178] 1. 第一方案

[0179] 图 14 表示根据第一方案的 3D 子带编码器 (1400)。编码器 (1400) 包括嵌入的基层编解码器 (1420) 和基层编解码器 (1420) 周围的下采样模块 (1410) 和上采样模块 (1430)。

[0180] 下采样模块 (1410) 和上采样模块 (1430) 是可任选的。当由嵌入编解码器 (1420) 编码的视频具有比输入到嵌入编解码器 (1420) 的空间子带更低的空间分辨率时, 使用它们。否则, 可以省略下采样模块 (1410) 和上采样模块 (1430)。

[0181] 图 14 中的虚线矩形包含在两级时间分解后, 与基层编解码器 (1420) 有关的操作。对基层, 输入视频为每秒  $f/4$  帧, 以及通过下采样模块 (1410), 降低分辨率。因此, 基层视频提供低帧速率和初始视频的 (可能地) 低空间分辨率表示。

[0182] 下采样模块 (1410) 使用简单的子采样、低通滤波器或一些其他滤波机制, 下采样  $t-L_2$  时间子带。例如, 下采样模块 (1410) 按两个水平或垂直因素, 降低空间分辨率。在一些实现中, 下采样模块 (1410) 应用“MPEG”滤波器, 如在 Bjontegaard et al. “H. 263 Anchors-Technical Description”或 Daubechies 9/7DWT 滤波器的文献中所述。一般来说, 下采样模块 (1410) 按任何任意比率, 改变输入到嵌入编解码器 (1420) 的时间分辨率, 以便实现所需基层时间分辨率和 / 或匹配基层编解码器 (1420) 中所允许的输入空间分辨率。

[0183] 嵌入编解码器 (1420) 使用用于嵌入编解码器 (1420) 的通常操作的任何一个, 编码下采样的  $t-L_2$  子带 (示为  $t_{s-LL}$  子带)。嵌入的编解码器 (1420) 以基层分辨率产生输出基层比特流 (1495), 其中, 基层比特流 (1495) 与相应的基层解码器兼容。为了编码器 (1400) 中的稍后编码的目的, 嵌入编解码器 (1420) 还使用用于嵌入编解码器 (1420) 的相应的解码操作, 还解码  $t_{s-LL}$  子带, 产生重构的子带  $t_{s-LL}'$ 。

[0184] 上采样模块 (1430) 使用一些滤波机制, 从基层编解码器 (1420) 上采样重构的输出子带  $t_{s-LL}'$ , 生成时间子带  $t-L_2$  的重构版本  $t-L_2'$ 。重构的子带  $t-L_2'$  具有与  $t-L_2$  相同的分辨率。例如, 上采样模块 (1430) 按 2 的倍数, 水平和垂直增加空间分辨率。在一些实现中, 上采样模块 (1430) 应用简单的 6-抽头滤波器  $\{-1, 5, 20, 20, 5, -1\}$ 、8-抽头 Sinc 滤波器, 如在 Chen et al., “Improved MC-EZBC with Quarter-pixel Motion Vectors”, 或 Daubechies 9/7 IDWT 滤波器。上采样滤波器不需要镜像下采样滤波器 - 例如, 可以将 MPEG 滤波器用于下采样和 IDWT 滤波器用于上采样。一般来说, 上采样模块 (1430) 按任意比改变从嵌入编解码器 (1420) 输出的空间分辨率, 以便将视频恢复成在下采样模块 (1410) 前具有的空间分辨率。

[0185] 从初始时间子带  $t-L_2$ , 编码器 (1400) 减去重构的版本  $t-L_2'$ 。这产生差值或剩余子带  $t-L_2''$ 。当基层编解码器 (1420) 有效时 (例如 VC-1 或 H. 264/AVC) 以及基层比特率适当时, 基层质量会相对好, 以便差值子带不具有更多能量。另一方面, 当基层编解码器 (1420) 低效或基层比特率不适当时, 基层质量会相对差, 使得差值子带具有更多能量。

[0186] 编解码器 (1400) 在差值子带  $t-L_2$  上执行另外两级时间分解。最后, 编码器 (1400) 在各个子带 (例如  $t-H_1$ ,  $t-H_2$ ,  $t-H_3$ ,  $t-H_4$  和  $t-L_4$ ) 上执行时间变换和熵编码以便生成可扩展比特流。

[0187] 图 15 表示根据第一方案的相应的 3D 子带解码器 (1500)。解码器 (1500) 包括嵌入基层解码器 (1520) 和基层解码器 (1520) 后的上采样模块 (1530)。

[0188] 上采样模块 (1530) 是可任选的。当由嵌入解码器 (1520) 解码的视频具有比输出到下游逆时间变换级的时间子带  $t-L_2$  更低的空间分辨率时, 使用它。否则, 可以省略上采样模块 (1530)。

[0189] 图 15 中的虚线矩形包含与在两个附加的逆时间分解级前的基层解码器 (1520) 有关的操作。对基层, 视频是每秒  $f/4$  帧, 以及可以降低空间分辨率。因此, 基层提供视频的低帧速率以及 (可能的) 低空间分辨率表示。实际上, 能通过基层解码器 (1520) 直接解码具有低帧速率和 (可能地) 低空间分辨率的视频并输出。

[0190] 或者, 解码器 (1500) 从解码器 (1500) 将用来重构视频的可缩放比特流 (例如  $t-H_1$ ,  $t-H_2$ ,  $t-H_3$ ,  $t-H_4$  和  $t-L_4$ ) 的子带的任何一个上执行熵解码和逆空间分解。解码器 (1500) 执行第三和第四级逆时间分解以便由  $t-H_3$ ,  $t-H_4$  和  $t-L_4$  子带和相关运动信息重构差分子带  $t-L_2'$ 。

[0191] 嵌入解码器 (1520) 接收用于在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, 嵌入解码器 (1520) 解码基层比特流 (1595) 中的压缩的、下采样的子带  $t_s-L_L$ , 产生子带的重构的版本  $t_s-L_L'$ 。

[0192] 上采样模块 (1530) 使用一些滤波机制, 上采样来自基层解码器 (1520) 的重构的子带  $t_s-L_L'$ , 生成时间子带  $t-L_2$  的重构版本  $t-L_2'$ 。重构子带  $t-L_2'$  具有与  $t-L_2$  相同的分辨率。例如, 上采样模块 (1530) 按 2 的倍数, 水平或垂直增加空间分辨率。一般来说, 上采样模块 (1530) 按任意比, 改变从嵌入解码器 (1520) 输出的时间分辨率, 以便使视频恢复到任何基层下采样前具有的空间分辨率。

[0193] 解码器 (1500) 将重构的差分子带  $t-L_2''$  增加到相应的重构子带  $t-L_2'$ 。这产生输入到逆时间分解的剩余部分的时间低通子带  $t-L_2$  的重构版本。

## [0194] 2. 第二方案

[0195] 图 16 表示根据第二方案的 3D 子带编码器 (1600)。编码器 (1600) 包括嵌入基层编解码器 (1620) 和基层编解码器 (1620) 周围的 DWT 模块 (1610) 和 IDWT 模块 (1630)。

[0196] 与图 14 所示的下采样模块 (1410) 和上采样模块 (1430) 类似, DWT 模块 (1610) 和 IDWT 模块 (1630) 是可任选的。当由嵌入编解码器 (1620) 编码的视频具有比输入到嵌入编解码器 (1620) 中的时间子带  $t-L_2$  更低的空间分辨率时, 使用它们。否则, 可以省略 DWT 模块 (1610) 和 IDWT 模块 (1630)。

[0197] 图 16 中的虚线矩形包含在两级时间分解后, 与基层编解码器 (1620) 有关的操作。DWT 模块 (1610) 处理  $t-L_2$  时间子带, 生成低通子带  $t_s-L_L$ 。通常, DWT 模块 (1610) 改变输入到嵌入编解码器 (1620) 的空间分辨率以便实现所需基层空间分辨率和 / 或匹配基层编解码器 (1620) 所允许的输入空间分辨率。DWT 模块 (1610) 还产生发送到 IDWT 模块 (1630) 的空间高通子带  $t_s-L_H$ 、 $t_s-H$  和  $t_s-H_H$ 。

[0198] 嵌入编解码器 (1620) 使用用于嵌入编解码器 (1620) 的通常操作的任何一个, 编

码  $t_s$ -LL 子带。嵌入编解码器 (1620) 以基层分辨率产生输出基层比特流 (1695)。嵌入编解码器 (1620) 还使用用于嵌入编解码器 (1620) 的相应的解码操作, 解码  $t_s$ -LL 子带, 产生重构子带  $t_s$ -LL'。

[0199] 从初始  $t_s$ -LL 子带, 编码器 (1600) 减去其重构的版本  $t_s$ -LL'。这产生剩余子带  $t_s$ -LL''。作为用于图 14 的编码器 (1400) 的情形, 当基层编码器 (1620) 有效且基层比特率适当时, 剩余  $t_s$ -LL'' 通常不具有太多能量。另一方面, 当基层编解码器 (1620) 低效或基层比特率不适当时, 剩余  $t_s$ -LL'' 可以具有更多能量。

[0200] 从剩余子带  $t_s$ -LL'' 和空间高通子带  $t_s$ -LH、 $t_s$ -HL 和  $t_s$ -HH, IDWT 模块 (1630) 生成新的剩余子带  $t$ -L<sub>2</sub>''。新剩余子带  $t$ -L<sub>2</sub>'' 具有与输入到 DWT 模块 (1610) 的子带  $t$ -L<sub>2</sub> 相同的空间分辨率。

[0201] 编码器 1600) 在剩余子带  $t$ -L<sub>2</sub>'' 上执行另外两级时间分解。最后, 编码器 (1600) 在各个子带 (例如  $t$ -H<sub>1</sub>,  $t$ -H<sub>2</sub>,  $t$ -H<sub>3</sub>,  $t$ -H<sub>4</sub> 和  $t$ -L<sub>4</sub>) 上执行空间变换和熵编码以便生成可缩放比特流。

[0202] 图 17 表示根据第二方案的相应的 3D 子带解码器 (1700)。解码器 (1700) 包括嵌入基层解码器 (1720) 和基层解码器 (1720) 周围的 DWT 模块 (1710) 和 IDWT 模块 (1730)。

[0203] DWT 模块 (1710) 和 IDWT 模块 (1730) 是可任选的。当由嵌入解码器 (1720) 解码的视频具有比输出到下游逆时间变换级的时间子带  $t$ -L<sub>2</sub> 更低的空间分辨率时, 使用它们。否则, 可以省略 DWT 模块 (1710) 和 IDWT 模块 (1730)。

[0204] 图 17 中的虚线矩形包含在两级另外的逆时间分解前, 与基层解码器 (1720) 有关的操作。基层提供低帧速率和视频的 (可能的) 低空间分辨率表示, 以及通过基层解码器 (1720) 直接解码这种更低质量视频并输出。

[0205] 或者, 解码器 (1700) 在来自解码器 (1700) 将用来重构视频的可缩放比特流 (例如  $t$ -H<sub>1</sub>,  $t$ -H<sub>2</sub>,  $t$ -H<sub>3</sub>,  $t$ -H<sub>4</sub> 和  $t$ -L<sub>4</sub>) 的子带的任何一个上执行熵解码和逆空间分解。解码器 (1700) 执行第三和第四级逆时间分解以便从  $t$ -H<sub>3</sub>,  $t$ -H<sub>4</sub> 和  $t$ -L<sub>4</sub> 子带和相关运动信息重构剩余子带  $t$ -L<sub>2</sub>''。

[0206] DWT 模块 (1710) 处理剩余子带  $t$ -L<sub>2</sub>'' , 产生空间低通剩余子带  $t_s$ -LL'' 和空间高通子带  $t_s$ -LH''、 $t_s$ -HL'' 和  $t_s$ -HH'' 的重构版本。通常, DWT 模块 (1710) 改变空间分辨率以便低通剩余子带  $t_s$ -LL'' 在空间分辨率方面, 与嵌入解码器 (1720) 的输出匹配。

[0207] 嵌入解码器 (1720) 接收用于以基层分辨率编码的视频的基层比特流 (1795)。使用适当的解码操作, 嵌入解码器 (1720) 解码基层比特流 (1795) 中的压缩、变换的子带  $t_s$ -LL, 产生子带的重构版本  $t_s$ -LL'。

[0208] 解码器 (1700) 将重构的剩余子带  $t_s$ -LL'' 增加到从嵌入解码器 (1720) 输出的相应的重构子带  $t_s$ -LL' 上。这产生初始  $t_s$ -LL 子带的重构版本。

[0209] 然后, IDWT 模块 (1730) 将重构的  $t_s$ -LL 子带与从 DWT 模块 (1710) 输出的空间高通子带  $t_s$ -LH''、 $t_s$ -HL'' 和  $t_s$ -HH'' 组合。这产生时间低通子带  $t$ -L<sub>2</sub> 的重构版本, 其被输入到逆时间分解的剩余部分。

[0210] 3. 第三方案

[0211] 在前两个编码方案中, 基层编解码器被嵌入在时间变换中以及在稍后的时间变换中, 进一步分解剩余信号。相反, 在第三编码方案中, 稍后的时间变换应用于时间低通帧

(与剩余帧相反)。

[0212] 图 18 表示根据第三方案的 3D 子带编码器 (1800)。编码器 (1800) 包括嵌入基层编解码器 (1820) 和基层编解码器 (1820) 周围的下采样模块 (1810) 和上采样模块 (1830)。图 18 中的虚线矩形包含在两级时间分解后,与基层编解码器 (1820) 有关的操作。关于图 14 所示的编码器 (1400) 的相应模块,下采样模块 (1810) 和上采样模块 (1830) 是可任选的。与第二和第三时间变换间的可缩放视频类似,基层视频为每秒  $f/4$  帧,以及通过下采样模块 (1810),降低分辨率。因此,基层视频再次提供初始视频的低帧速率和(可能的)低时间分辨率表示。

[0213] 下采样模块 (1810) 使用简单的子采样、低通滤波器或一些其他过滤机制,下采样  $t-L_2$  时间子带。通常,下采样模块 (1810) 按任意比率,改变输入到嵌入编解码器 (1820) 的时间分辨率以便实现所需基层空间分辨率和 / 或匹配基层编解码器 (1820) 所允许的输入空间分辨率。

[0214] 嵌入编解码器 (1820) 使用用于嵌入编解码器 (1820) 的通常操作的任何一个,编码下采样的  $t-L_2$  子带(示为  $t_{s-LL}$  子带)。嵌入编解码器 (1820) 产生最低比特率,以基层分辨率输出基层比特流 (1895),其中,基层比特流 (1895) 与相应的基层解码器匹配。为在编码器 (1800) 中稍后编码,嵌入编解码器 (1820) 使用用于嵌入编解码器 (1820) 的相应的解码操作,解码  $t_{s-LL}$  子带,产生重构子带  $t_{s-LL}'$ 。

[0215] 上采样模块 (1830) 使用一些过滤机制,从基层编解码器 (1820) 上采样所重构的输出子带  $t_{s-LL}'$ ,生成时间子带  $t-L_2$  的重构版本  $t-L_2'$ 。所重构的子带  $t-L_2'$  具有与  $t-L_2$  相同的分辨率。通常,上采样模块 (1830) 按任何比率,改变来自嵌入编解码器 (1820) 的输出空间分辨率以便使视频恢复到在下采样模块 (1810) 前所具有的空间分辨率。

[0216] 编码器 (1800) 在初始(预基层处理)时间子频带  $t-L_2$  上执行第三和第四级时间变换。对第三和第四时间变换,编码器 (1800) 计算用于时间高通子带  $t-H_3$  和  $t-H_4$  的运动补偿预测。如所期望的,时间子带  $t-L_2$  提供用于第三级时间变换中,运动补偿的候选基准帧(例如偶数帧  $t-L_2$ ),以及时间子带  $t-L_3$  提供用于第四级时间变换中的运动补偿的候选基准帧(例如偶数帧  $t-L_3$ )。根据第三方案,在上采样、重构的子带  $t-L_2'$  中的时间相应图像充当用于运动补偿的另外的候选基准帧。例如,在第三级时间变换中, $t-L_2'$  的交替帧(具有与  $t-L_2$  的基数下标的帧相同的时间)可用作用于奇数下标帧  $t-L_2$  的 MCP 的基准。在第四级时间变换中, $t-L_2'$  的每第四帧(具有与相应的奇数下标帧  $t-L_3$  相同的时间)可用作用于  $t-L_3$  的时间相应奇数下标帧的 MCP 的基准。

[0217] 图 20 表示在嵌入基层编解码器后,用于时间变换中的运动补偿的不同基准组合情形。候选基准帧包括来自输入时间低通子带的帧(例如偶数帧)。候选基准帧还包括来自基层的上采样输出的时间相应的(例如相同时间)、重构帧。例如,在图 20 的组合 (1) 中,高通子带帧 H 使用来自先前时间子带(对应于低通子带帧 L)的偶数帧的第一运动补偿预测和使用来自基层(与高通子带帧 H 时间上共存)的上采样重构的第二运动补偿预测。相应地,高通子带帧 H 仅用来更新用于运动补偿的偶数帧(以便生成相应的低通子带帧 L)。如果在预测步骤中使用来自基层的基准,对它来说,不需要相应的更新步骤。

[0218] 在一些实现中,编码器基于逐个宏块,选择和切换用于运动补偿的基准帧。另外,编码器在一些其他基础上(例如,每当前奇数帧、每运动矢量和宏块划分),选择和切换用

于运动补偿的基准帧。

[0219] 基准图 18, 在最后一个时间分解后, 基层输出  $t-L_2'$  的帧速率是最终时间低通子带  $t-L_4$  的帧速率的四倍。从最终时间低通子带  $t-L_4$ , 编码器 (1800) 减去基层输出  $t-L_2'$  的时间相应的、上采样重构帧。这生成剩余子带  $t-L_4'$ 。最终, 编码器 (1800) 在各个子带 (例如  $t-H_1, t-H_2, t-H_3, t-H_4$  和  $t-L_4'$ ) 上执行空间变换和熵编码以便生成可缩放比特流。

[0220] 图 19 表示根据第三方案, 相应的 3D 子带解码器 (1900)。解码器 (1900) 包括嵌入基层解码器 (1920) 和基层解码器 (1920) 后的上采样模块 (1930)。

[0221] 关于图 15 所示的解码器 (1500), 上采样模块 (1930) 是可任选的。当由嵌入解码器 (1920) 解码的视频具有比由下游逆时间变换级使用的时间子带帧更低的时间分辨率时使用它。否则, 可以省略上采样模块 (1930)。

[0222] 图 19 中的虚线矩形包含与在四级逆时间分解前的基层解码器 (1930) 有关的操作。关于其他基层解码器, 能通过基层解码器 (1920) 直接解码具有低、基层帧速率和 (可能的) 低空间分辨率的视频并输出。

[0223] 否则, 解码器 (1900) 在来自可缩放比特流的子带的任何一个 (例如  $t-H_1, t-H_2, t-H_3, t-H_4$  和  $t-L_4'$ ) 上执行熵解码和逆空间分解, 解码器 (1900) 将用来重构视频。

[0224] 嵌入解码器 (1920) 接收用于以基层分辨率编码的视频的基层比特流 (1995)。使用适当的解码操作, 嵌入解码器 (1920) 解码基层比特流 (1995) 中的压缩、下采样子带  $t_s-LL$ , 产生子带的重构版本  $t_s-LL'$ 。上采样模块 (1930) 使用一些过滤机制, 从基层解码器 (1920) 上采样重构的子带  $t_s-LL'$ , 生成时间子带  $t-L_2$  的重构版本  $t-L_2'$ 。

[0225] 解码器 (1900) 将剩余的子带  $t-L_4'$  增加到上采样的基层输出  $t-L_2'$  的时间相应 (例如相同时间) 帧上。这产生时间低通子带  $t-L_4$  的重构版本, 其被输入到第四级逆时间分解。

[0226] 解码器 (1900) 执行第三和第四级逆时间分解以便由重构  $t-H_3, t-H_4$  和  $t-L_4$  子带及相关的运动信息, 重构时间子带  $t-L_2$ 。对第三和第四逆时间变换, 解码器 (1900) 计算用于时间高通子带  $t-H_3$  和  $t-H_4$  的运动补偿预测。时间子带  $t-H_3$  具有用于第四级时间变换中的 MCP 的重构的基准帧 (例如, 重构的偶数帧  $t-L_3$ ), 以及时间子带  $t-L_2$  具有第三级时间变换中的 MCP 的重构基准帧 (例如重构偶数帧  $t-L_2$ )。上采样的、重构子带  $t-L_2'$  中的时间相应 (例如相同时间) 的图像可以充当用于第三和第四级逆时间变换中的另外的基准帧。

[0227] B. 基层信息的示例性另外用途

[0228] 除提供向后兼容性和在用于低比特率视频的码率 - 失真意义上, 提高压缩效率外, 来自基层的输出能具有另外的用途。

[0229] 1. 重新使用来自基层的运动矢量

[0230] 在具有基于提升的 MCTF 的 3D 子带编码中, 通过运动估计, 获得用于 MCP 步骤的运动矢量。在一些实现中, 运动估计过程与用于符合现有的视频标准, 诸如 MPEG2、H. 264/AVC 等等的传统的运动估计类似。此外, Haar 和 5/3 时间变换分别与传统的 P- 图像和 B- 图像的运动补偿过程类似。

[0231] 对与现有的标准, 诸如 MPEG-2、MPEG-4、H. 264/AVC 等等兼容的基层编解码器, 使用用于图像内 (P- 图像或 B- 图像) 的内部压缩编码的块通常具有相关联的运动信息。同时, 在 3D 子带编码中, 处于来自基层编码器的时间分解下游级 (例如处于图 14、16 和 18 的



第三和第四级)的高通子带能在基层输出中找出时间相应的图片。在许多情况下(特别是根据第三方案),在用于这些高通子带的运动信息和基层中的时间相应的图片内的运动信息间,存在高度类似。基于该相关,为几个目的,编码器和/或解码器能重新使用来自基层编解码器的运动矢量。

[0232] 在基层运动矢量的另外的用途的一个例子中,3D子带编码器和3D子带解码器简单地使用基层运动矢量,作为用于时间相应的高通子带帧的块的实际运动矢量。(当基层帧和高通子带帧具有不同空间分辨率时,3D子带编码器和3D子带解码器能相应地伸缩基层运动矢量)。特别地,在低比特率,为此目的,3D子带编码器和3D子带解码器使用基层运动矢量。这降低了用于运动信息的开销,当然,因为作为基层比特流的一部分,已经发出基层运动矢量信号。在许多情况下,在运动预测精度损失方面的惩罚很轻微。

[0233] 在基层运动矢量的另外的用途的另一例子中,3D子带编码器和3D子带解码器在运动矢量预测中,使用来自基层编解码器的运动矢量。作为传统的运动编码和解码的一部分,运动矢量预测提供用于当前运动矢量的值的预测。然后,当前运动矢量可以被表示和被发信号为相对于预测运动矢量的差值。编码器和解码器从相邻块的先前重构运动矢量,选择或导出用于当前运动矢量的预测运动矢量。

[0234] 在一些实现中,当选择和导出用于时间高通子带的块的当前运动矢量的预测运动矢量时,3D子带编码器和3D子带解码器使用来自时间相应的基层帧的空间相应块的运动矢量。例如,3D子带编码器和3D子带解码器将来自基层的运动矢量用作预测的运动矢量。或者,3D子带编码器和3D子带解码器使用来自基层的运动矢量作为选择预测运动矢量的候选(例如作为候选的中值)。

[0235] 在基层运动矢量的另外的用途的另一例子中,3D子带编码器使用基层运动矢量来加速运动估计。例如,子带编码器在运动估计中,先考虑相应的基层运动矢量以便更快速地找出适当的当前运动矢量。

[0236] 类似地,3D子带编码器和3D子带解码器能重新使用来自基层的运动模式(例如块划分、方向)信息。在一些情况下,基层模式信息可以用作用于时间相应的高通子带帧的块的实际模式信息。或者,基层模式信息可以用于预测模式信息或加速运动估计。

[0237] 图21至23表示不同编码结构(帧类型的模式),在这些结构下,来自基层的运动信息特别有用于在基层后的时间分解中的时间高通子带。在图21中,调整基层的编码结构以便适合于用于3D子带编码的公共结构。在两个I或P图像间插入三个B图像,以及第二B图像(就像用在H.264/AVC中的所存储的B图像)用作用于第一和第三B图像的基准。用这种方式,基层中的B-图像的运动矢量能自然地用于3D子带编码中的时间相应的高通子带帧。

[0238] 为了与用于基层编解码器的某些现有标准兼容,图22的框架将传统的编码结构用于基层,但调整用于3D子带编码的帧类型的模式。在基层中的两个I/P图像间插入三个B图像,但三个B-图像仅将I/P图像用作基准(无B图像用作运动补偿基准)。在3D子带编码方案中,模型包括在两个低通子带间生成的三个高通子带。

[0239] 图23示例说明类属模式。在基层中,用B-或P-图像编码时间上对应于3D子带编码中的高通子带的图像。而图21和22中所示的模式适合于5/3时间分解,图23中所示的模式能自适应于Haar和5/3时间分解(分别对应于P-图像和B-图像关系)。

## [0240] 2. 从基层预测内区域

[0241] 当在视频序列中存在遮挡或不规则运动时,运动估计仅可找出用于时间高通子带中的一些块的不良匹配块。当使能无内编码模式时,用于这些情形的强迫内部预测会产生编码性能恶化和对可视质量具有不良影响。

[0242] 在 H.264/AVC 标准中,内部预测技术解决 MCP 失败的情形。在内部预测中,通过来自相邻重构像素的空间预测,获得用于当前块的预测器。在具有开环 MCTF 结构的 3D 子带编码中,然而,这种空间预测易于引入不匹配和错误漂移。

[0243] 在如上所述的第一和第二嵌入基层编码方案中,在完成基层编码和解码后,在不同信息上执行用于 3D 子带编码的进一步时间分解。(在图 14 中,例如,差值信息是输入时间子带  $t-L_2$  和来自基层编解码器的输出  $t_s-LL'$  的上采样版本  $t-L_2'$  间的剩余量)。典型地,差值仅具有高频能量及其空间冗余为低。因此,当在基层编解码器中的内部模式中,编码块时,3D 子带编码中的稍后的时间分解中的内部预测通常表现为不必要。这有助于避免与开环 3D 子带编码中的内部预测相关联的不匹配问题,同时提高编码性能和可视质量。

[0244] 另一方面,在如上所述的第三嵌入基层编码方案中,在差值信息上不执行用于 3D 子带编码的进一步时间分解。然而,在第三方案中,基层编解码器的输出可以提供候选基准图片,以及候选基准图片可以包括使用空间内部预测编码的块。相对于来自基层输出的基准图像的内部编码块,执行运动补偿有助于解决 MCTF 中的 MCP 仅找出不良匹配块的情形。来自基层输出的基准图片也能作用于稍后的 3D 子带编码中的低通子带的预测器。3D 子带编码器能使用来自基层的内部模式信息来由用于稍后的 MCP 或其他预测的基层输出,确定何时使用基准图像。

## [0245] 3. 用于进一步时间分解的运动估计

[0246] 在纯 3D 子带编码中,以及在根据如上所述的第三方案,具有嵌入基层编解码器的 3D 子带编码中,在原始低通子带上进行运动估计以便获得更高级高通子带和运动信息。这些运动估计有效地包含许多传统的运动估计参数,诸如“ $\lambda$  参数”,该参数相对编码运动信息的成本加权编码剩余信息的成本。

[0247] 对具有根据如上所述的第一和第二方案的嵌入基层编解码器的 3D 子带编码,然而,在差值信息上执行用于进一步时间分解的运动估计。同样地,运动估计采用传统的运动估计参数来在进一步时间分解中,利用相邻帧中的时间冗余。例如,编码器调整  $\lambda$  参数来考虑到时间分解的不同输入。

## [0248] C. 用于使用基层编解码器或解码器的示例性技术

[0249] 图 24 表示通过嵌入基层编解码器,用于可缩放编码的技术 (2400)。图 25 表示通过嵌入基层解码器,用于可缩放解码的技术 (2500)。3D 子带编码器,诸如基准图 14、16 或 18 所述的编码器执行图 24 中所示的技术 (2400),以及 3D 子带解码器,诸如基准图 15、17 或 19 所述的解码器执行图 25 中所示的技术 (2500)。另外,另一编码器或解码器执行技术 (2400, 2500)。

[0250] 基准图 24,3D 子带编码器使用子带编码,编码 (2410) 视频的一层或多层。这产生基层编解码器的输入。例如,3D 子带编码器执行一级或多级 MCTF,查找用于时间高通子带的运动信息以及产生时间低通子带以使用作基层编解码器的基础帧速率视频输入。另外,3D 子带编码器在该级执行其他和 / 或另外类型的编码。

[0251] 3D 子带编码器用基层编解码器, 编码 (2430) 基层视频。例如, 基层编解码器是 H. 264/AVC、MPEG-2、VC-1 或 WMV9 并产生相应的压缩视频比特流和重构的输出视频。运动信息和内部模式信息是压缩的视频比特流的一部分, 或该信息可以以一些其他形式分开提供给 3D 子带编码器。

[0252] 然后, 3D 子带编码器使用来自基层编解码器的结果, 编码 (2450) 一层或多层视频。例如, 3D 子带编码器在从基层编码和解码的结果计算的差值信息上, 执行一级或多级时间分解。或者, 3D 子带编码器在来自基层编解码器的重构视频中, 使用基准图片, 在时间低通子带上执行一级或多级时间分解。或者, 3D 子带编码器在时间分解的编码或判定中, 使用运动信息和 / 或内部模式信息。另外, 3D 子带编码器在后续子带编码中, 以一些其他方式使用来自基层编解码器的结果, 或跳过后续编码 (2450)。

[0253] 编码器确定 (2490) 是否继续编码过程, 如果是, 继续下一帧。

[0254] 基准图 25, 3D 子带解码器用基层解码器, 解码 (2530) 基层视频。例如, 基层解码器是 H. 264/AVC、MPEG-2、VC-1 或 WMV9 解码, 以及解析和解码相应的压缩视频比特流和产生重构的输出视频。

[0255] 3D 子带解码器确定 (2540) 用于解码的分辨率。根据用户级设定值或根据解码器实际上接收多少信息 (例如仅基层视频或一个或多个另外的层) 确定的, 或用一些其他方式确定的一些其他设定值, 可以确定分辨率。另外, 3D 子带解码器仅具有一个解码路径。例如, 一个解码器可以包括基层路径, 而另一解码器包括基层 + 可缩放层路径, 适合于各个解码器的资源。

[0256] 如果没有要被解码的较高分辨率视频 (在判决 2540 外的“否”路径), 3D 子带解码器在级 (2590) 继续。

[0257] 否则, 3D 子带解码器使用来自基层解码的结果, 解码 (2550) 用于视频的一个或多个另外的层。例如, 3D 子带解码器在差值信息上执行逆时间分解的一级或多级, 然后将重构的差值信息增加到基层解码的结果上。或者, 3D 子带解码器重构使用来自基层解码器的重构视频中的基准图片, 执行一级或多级逆时间分解以便重构时间低通子带。另外, 3D 子带解码器在后续子带解码中, 以一些其他方式, 使用来自基层解码器的结果。

[0258] 最后, 解码器确定 (2590) 是否继续解码过程, 以及如果是, 继续下一帧。

[0259] 可以分开图 24 和 25 中所示的各个操作, 与其他操作结合并记录。

[0260] VI. 解码 SDMCTF 编码视频中的空间可缩放性

[0261] 通常, 根据何时在 3D 子带视频编码中执行 MCTF, 存在两种类别 MCTF。在第一种类中, 称为带内 MCTF (“IBMCTF”) 中, 编码器首先空间上分解视频, 然后在子带 / 小波域中执行 MCTF。在第二种类中, 称为空间域 MCTF (“SDMCTF”), 在空间分解前, 编码器在空间域中直接执行 MCTF。类似地, 在解码期间, 在空间域 (“SDIMCTF”) 或带内 (“IBIMCTF”) 中可以执行 IMCTF。

[0262] 因此, 在具有 SDMCTF 的 3D 子带编码中, 通过时间域中的多次运动对准时间变换和后续的空间变换, 分解整个视频。图 5 表示具有四级 (逆) 时间变换和两级 (逆) 空间变换的整个编码和解码方案, 且在解码中没有空间可缩放性。在图 5 中, 解码过程正好是编码的逆过程。

[0263] 通常, 在用于特定比特率的质量方面, IBMCTF 方案的性能不与 SDMCTF 方案的性能

一样好。为此,许多 3D 子带编码器使用 SDMCTF。在一些应用中,然而,解码器产生视频输出的低空间分辨率版本。例如,对以 CIF 分辨率 (352×288 像素) 的初始视频,对于用一级空间可缩放性解码,3D 子带解码器产生解码视频的 QCIF 版本 (176×144 像素)。这可以是如果解码器与更小屏幕设备 (例如便携式视频电话) 相关联,或如果解码器仅接收不可靠连接上的低空间分辨率子带信息的情形。当在空间域中,执行视频上的 MCTF 操作时,然而,因为根据全空间分辨率表示的运动信息、基准图片等等,这些情况是成问题的。

[0264] 该节描述当以低空间分辨率解码用于输出的 SDMCTF 编码的视频时,提高解码性能的各种机制。在一些实施例中,例如,3D 子带视频解码器接收使用以某个分辨率 (例如 CIF) 的 SDMCTF 编码的视频,但以低分辨率 (例如 QCIF) 解码视频以便输出。3D 子带解码器使用几个不同机制的任何一个来提高解码性能。

[0265] A. 在解码 SDMCTF 视频中的简单的空间可缩放性方案

[0266] 图 26 示例说明两个简单的方案 (2600, 2650), 用于解码和输出来自用于 SDMCTF 编码的视频的压缩视频比特流的低空间分辨率视频。在图 26 中,低空间分辨率视频与全空间分辨率相差一级空间可缩放性。

[0267] 第一方案 (2600) 提供直接解决方案。3D 子带解码器首先恢复全空间分辨率视频 (示为 CIF 视频)。(假定用于第二逆空间变换的空间高通 (“SH”) 子带丢失,在第二逆空间变换中,解码器将 SH 子带处理为空 (SH = 0)。在逆空间分解中,3D 子带解码器通过滤波器 (示为空间变换),下采样整个空间分辨率视频为输出分辨率 (示为 QCIF)。解码器执行 SDIMCTF,以及当在中间步骤中,解码 CIF 视频时,解码器使用许多解码资源 (例如计算资源、基准图像缓冲器)。

[0268] 另一方面,在第二方案 (2650) 中,3D 子带解码器执行 IBIMCTF。第二方案 (2650) 示例说明解码低分辨率视频的启发式方式。与第一方案 (2600) 相同,第二方案 (2650) 包括熵解码和四级逆时间变换。第二方案 (2650) 有效地包括比第一方案 (2600) 更少的逆空间变换。(对于第二方案 (2650) 所示的第二逆空间变换和后续空间变换彼此抵消并可以省略。)

[0269] 在第二方案 (2650) 中,3D 子带解码器基本上将空间低通 (“SL”) 子带 (第一逆空间变换的输出) 处理为整个初始视频,以 SL 子带分辨率执行逆时间变换。这近似没有空间可缩放性的解码过程,以及其具有低计算复杂性和低缓冲要求 (用于基准图片等等) 的优点。此外,视频的初始分辨率和空间可缩放性对解码器来说总体上是透明的。在低比特率,方案 (2650) 的性能是令人满意的,但在高比特率不令人满意。特别地,当比特率增加时,它变得难以或不可能重构高质量低空间分辨率视频,即使比特率非常高。

[0270] 图 26 所示的两个解码方案 (2600, 2650) 间的差别在于在 IMCTF 期间,如何执行 MCP 和 MCU 步骤。图 27 示例说明在一级 IMCTF 中,用于两个方案 (2600, 2650) 的 MCP。特别地,图 27 表示在基准图像中,由可用 SL 子带信息 (不是 SH 子带信息),在预测步骤中,生成 SL 提升信号。

[0271] 对 SDIMCTF, SL 提升信号生成如下:

[0272]

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

[0273] 其中,INTP<sub>SD</sub> 表示在用于基准图像信息的空间域中,像素上的内插,以及 MCP<sub>SD</sub> 表示

空间域中的 MCP。DWT 是线性操作,以及:

$$[0274] \quad \text{DWT}(F_1+F_2) = \text{DWT}(F_1) + \text{DWT}(F_2) \quad (10)$$

[0275] 为使用 SL 分辨率缓冲器来保存中间结果, DWT 将空间域预测降低到 SL 域中的预测。

[0276] 对于 IBIMCTF, SL 提升信号生成如下:

[0277]

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

[0278] 其中,  $\text{INTP}_{SL}$  表示用于基准图片信号的 SL 域中的像素上的内插, 以及  $\text{MCP}_{SL}$  表示 SL 域中的 MCP。为简化起见, 相同的内插滤波器用于  $\text{INTP}_{SD}$  和  $\text{INTP}_{SL}$ 。

[0279] B. 理论分析

[0280] 各种因素导致 SDMCTF 编码的视频的 3D 子带解码中的空间可缩放性的问题。一个因素是 SL 域中的内插和运动补偿低效。另一因素是 IMCTF 的位置 (与编码中的 MCTF 的位置相比)。另一因素是由运动补偿引起的子带上的功率谱泄漏。

[0281] 除这些因素外, 各种编码/解码特征能导致编码器中的 SDMCTF 和解码器中的 IBIMCTF 间的不匹配。两个这种特征是运动矢量缩放和 OBMC。在该章节中, 为简化起见, 假定在 IBIMCTF 中, 运动矢量的精度不改变, 以及用于运动补偿的子像素内插足够好在 IBIMCTF 期间, 利用运动信息。同样为了简化起见, 禁用 OBMC, 以及假定 MCP 操作是仅根据可用运动矢量的简单的像素提取操作。即使通过这些简化的假定, 在先前的段落中列出的因素会降低 IBIMCTF 的性能。

[0282] 1. 内插

[0283] 即使当相同的内插滤波器 (诸如双线性滤波器或 8-抽头 Sinc 滤波器) 用于空间域内插和 SL 域内插, 由于两个操作的不同规模, 在一帧中相同位置处的给定部分像素被从不同像素集内插。这导致在子像素内插的基准图片间的不匹配。

[0284] 由于另一原因, SL 域中的内插可能不是最佳的。由于小波变换的抽取方面, 只有空间 LL 频带不是初始视频的整个低通信号的完全表示。即, 不考虑高通信号或当缺少高通信号时, 在相位位置的一半处, 丢失低通信号部分。抽取的一个结果是空间 LL 频带的偏移变量属性。由于 LL 频带的不完整性, 当获得内插低通帧时, 在 LL 频带中直接执行内插 (如方程式 (12) 所示) 可能不是最佳的。

[0285]

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

$$[0286] \quad \xrightarrow{SL} \text{IDWT} \rightarrow \text{OCDWT} \rightarrow \text{INTP}_{OC-LL} \xrightarrow{mxm SL} \text{MCP}_{SL} \xrightarrow{SL} \quad (13), \text{以及}$$

[0287]

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

[0288] 备选方案是通过转换成超完整小波表示, 执行半像素内插。通过完整-超完整 DWT (未示出) 或通过逆 DWT 和超完整 DWT (“OCDWT”) (如方程式 13 中所示), 可以产生来自 SL 子带的超完整表示。该 IDWT 加 OCDW 组合充当一级内插。在超完整 (overcomplete) 小波域中, 通过传统的内插, 执行剩余级内插, 在方程式 (13) 中, 表示为  $\text{INTP}_{OC-LL}$ 。

[0289] 到传统的超完整小波表示的转换仅在 SL 域中提供半像素内插。为支持 SL 域中的四分之一像素运动矢量精度或更细微运动, 解码器可以使用连续脉冲超完整小波变换

(“CPOCWT”)。对 SL 子带中的四分之一像素位置,解码器在空间域中执行内插,如方程式 (14) 所示。然后,通过 DWT,获得内插帧的低通子频带。DWT 是缩放灵敏的操作。为匹配编码器中的 DWT,解码器在内插帧中的  $m$  采样像素上执行 OCDWT,其中  $m \times m$  是 OCDWT 前的空间域内插的因子。在方操作 (14) 中, DSDWT 表示将空间域内插帧下采样成许多子帧,在各个子帧上执行 OCDWT,以及向后交错所获得的系数。同样地, DSDWT 是一种 OCDWT。

[0290] 根据方程式 (12) 至 (14) 的每一个,解码器在  $m \times m$  基准图像上,在 SL 域中,执行运动补偿预测。

[0291] 2. DWT 和 MCP 的位置交换

[0292] 对 IBIMCTF,即使用空间域内插代替带内内插,如在 (14) 中,用于 MCP 的基准图片仍然是低通帧 -MCP 出现在小波域中。与编码器相比,交换 DWT 和 MCP 的期望顺序(其中,在编码器中,MCP 在 DWT 前)。然而,由于 DWT 的偏移变化属性,DWT 和 MCP 操作是不可交换的:

$$[0293] \quad \text{DWT}(\text{MCP}_{\text{SD}}(F_{\text{SD}})) \neq \text{MCP}_{\text{SL}}(\text{DWT}(F_{\text{SD}})) \quad (15) \text{ 和}$$

$$[0294] \quad \text{IDWT}(\text{MCP}_{\text{SL}}(F_{\text{SL}})) \neq \text{MCP}_{\text{SD}}(\text{IDWT}(F_{\text{SL}})) \quad (16)$$

[0295] 假定为简化起见,MCP 包含简单的、基于块的运动偏移操作,方程式 (15) 能表示如下。假定将当前帧划分成块  $B = \{B_i | i = 1, \dots, M\}$ ,以及用于这些块的运动矢量为  $MV = \{mv_i | i = 1, \dots, M\}$ 。  $F_{\text{ref}}^i$  是基准帧  $F_{\text{ref}}$  的新版本,其中保留仅由块  $B_i$  引用的像素以及将其其他像素设置成零。

$$[0296] \quad F_{\text{pred}}(x) = \sum_{i=1}^m F_{\text{ref}}^i(x + mv_i) \quad (17)$$

$$[0297] \quad \text{DWT}(\text{MCP}(F_{\text{ref}})) = \text{DWT}\left(\sum_{i=1}^m F_{\text{ref}}^i(x + mv_i)\right)$$

$$[0298] \quad = \sum_{i=1}^m \text{DWT}(F_{\text{ref}}^i(x + mv_i)) \neq \sum_{i=1}^m (\text{DWT}F_{\text{ref}}^i(x + mv_i)) \quad (18)$$

$$[0299] \quad = \text{MCP}\left(\sum_{i=1}^m \text{DWT}(F_{\text{ref}}^i)\right) = \text{MCP}(\text{DWT}\left(\sum_{i=1}^m F_{\text{ref}}^i\right))$$

$$[0300] \quad \neq \text{MCP}(\text{DWT}(F_{\text{ref}}))$$

[0301] (18) 中的第一不等式是由于 DWT 的偏移变化属性。(18) 中的第二不等式是由于当运动非常复杂时,在 MCP 中的运动偏移期间重叠和未覆盖。

[0302] 一个可能的解决方案是在 DWT 前,将 MCP 移动到空间域,如方程式 (19) 所示。在这种情况下,MCP 是 SDIMCTF 的一部分。

[0303]

$$\xrightarrow{\text{SL}} \text{IDWT} \xrightarrow{\text{SD}} \text{INTP}_{\text{SD}} \rightarrow \text{MCP}_{\text{SD}} \xrightarrow{\text{SD}} \text{DWT} \quad (19)$$

[0304] 3. 由于运动偏移的功率谱泄漏

[0305] 在 3D 小波编码中,通常假定空间 - 时间子带为独立。在许多应用中,随意丢失空间 - 时间子带,特别是高通子带。在具有 MCTF 的 3D 子带编码中,然而,由于由运动偏移引起的功率谱泄漏,相邻帧的低通子带和高通子带具有类似。

[0306] 当根据运动,仅将信号包含在一个空间子带的帧偏移时,部分信号将传送到其他空间子带。图 28 示例说明用于简单全局运动的现象。在图 28 的第一行中,将初始帧 Org

分成两部分:SL 信号 A 和 SH 信号 B。在第二行中,通过简单图像域运动偏移 (“IMS”),在空间域中偏移仅包含 SL 信号 A 的帧,以及偏移帧现在包含 SH 信号。类似地,在图 28 的第三行中,在空间域中偏移仅包含 SH 信号 B 的帧,以及偏移的帧现在包含一 SL 信号。当运动复杂时,频谱泄漏问题变得更严重。

[0307] 因此,通过编码器中的 SDMCTF,但使用 SL 信息的 IMCTF,基准帧的 SL 分量预测帧的 SH 分量部分。因此,即使解码器在开始时,仅具有 SL 子带信息,在每级 IMCTF 的中间结果中,解码器获得有意义的 SH 子带信息。当解码器使用 SL 分辨率缓冲器来保持 IMCTF 间的中间结果时,然而,这丢失 SH 子带信息,在下一 IMCTF 级中,这些信息或许对 MCP 是有用的。

[0308] 此外,基准帧的 SH 分量预测帧的 SL 分量部分。为编码 SL 分量,编码器可用利用基准帧中的 SH 子带信息。当解码器不能存取这些 SH 分量时,缺少 SH 子带导致一种漂移。(例如,当编码器从全分辨率帧(包括 SH 信息),执行 MCP 时,但解码器仅具有 SL 信息)。结果是 PSNR 最高限度-PSNR 曲线以相对低比特率水平旋转。

[0309] C. 用于解码 SDMCTF 的改进方案的例子-具有空间可缩放性的编码视频

[0310] 在一些实施例中,当解码用于低空间分辨率输出的 SDMCTF 编码的视频时,3D 子带解码器使用几个技术的任何一个来提高解码性能。这些技术逐步提高解码性能,以及大部分不要求改变编码器和传送比特流。总的来说,这些技术以不同级复杂性,提供具有不同选项的灵活解码方案。

[0311] 在一些方案中,解码器将额外资源分配给解码。通常,当将额外的资源专用于解码,解码器重构更好的视频。此外,当额外缓冲器资源专用于存储基准图片信息时,解码器重构更好的视频。

[0312] 或者,编码器或解码器强调用于解码 SL 视频的 SH 子带信息。例如,为将 SH 子带信息包含在 IMCTF 判决中,即使当解码器仅重构 SL 分辨率视频,比特流抽取器将带宽分配给 SH 系数,而不完全丢失它们。

[0313] 该节中的各个图(即图 29 至 34)表示一级空间可缩放性。另外,3D 子带解码器输出用于多级空间可缩放性的视频。沿相同行,该节中的各个图表示输出用于以 CIF 分辨率,通过用 MCTF 编码的初始视频的 QCIF 视频的 3D 子带解码器。在此描述的技术和工具不限于使用 CIF 和 QCIF 分辨率。

[0314] 对该节中的许多图,运动矢量具有以初始分辨率的半像素精度和 SL 分辨率的四分之一像素精度。这影响诸如子像素内插的操作。另外,运动矢量具有其他精度,以及解码器相应地执行或多或少子像素内插。此外,尽管该节中的几个图表示用于内插的 Sinc 滤波器,另外解码器使用另一类型的滤波器(例如双直线、双三次)。

[0315] 最后,本节中的各附图示出了二重 DWT 和 IDWT 操作。或者,解码器适用其它操作用于空间变换和逆空间变换。

[0316] 1. 方案 A:IBIMCTF(基线)

[0317] 为了与稍后方案比较,图 29 示例说明 IBIMCTF,而没有在解码期间专用的另外的计算或缓冲器资源。图 29 中的方案对应于图 26 的 IBIMCTF 解码器(2650)的一级 IMCTF。图 29 中所示的操作出现在小波域中。

[0318] 根据图 29,3D 子带解码器使用 SL 缓冲器(2910)存储 SL 子带信息,用于用作基准

图片信息。SL 子带信息的有效分辨率是 QCIF。

[0319] 解码器使用  $4 \times 4$  Sinc 滤波器, 执行子像素内插 (2920)。(用于 SL 域中的 MCP 的运动矢量精度是四分之一像素精度)。这产生子像素内插基准帧 (2930)。内插基准帧的有效分辨率为 4CIF (704  $\times$  576 像素)。

[0320] 解码器将内插的基准帧 (2930) 用于基准图片信息, 在 SL 域中执行 MCP (2940)。MCP<sub>SL</sub> (2940) 在 SL 域中产生预测 (2950)。同样地, 预测 (2950) 的有效分辨率为 QCIF。

[0321] 2. 方案 B: 最佳 IBIMCTF

[0322] 在第二类型的方案中, 3D 子带解码器通过将内插操作和 / 或 MCP 操作移动到空间中, 提高 IMCTF。解码器仍然将 SL 分辨率缓冲器用于基准图像信息。图 30 表示用于第二类型方案的通用框架, 其中将逆时间变换表示为最佳逆时间变换  $OT_i^{-1}$ 。图 31, 32 和 33 表示根据第二类型的不同变化的一级 IMCTF 的例子。

[0323] a. 方案 B1

[0324] 根据如图 31 所示的方案 (方案 B1), 3D 子带解码器使用 SL 缓冲器 (3110) 来存储 SL 子带信息, 用作基准图片信息。解码器使用缓冲的 SL 子带信息, 执行 IDWT 操作 (3120)。在 IDWT 操作 (3120) 中, 将 SH 子带视为空 ( $SH = 0$ )。从而 IDWT 操作 (3120) 从 SL 子带信息产生基准帧 (3130)。当 SL 子带的有效分辨率是 QCIF 时, 基准帧 (3130) 的有效分辨率为 CIF。

[0325] 然后, 解码器在基准帧 (3130) 上执行 OCDWT (3140) 操作, 产生超完整表示 (3150)。另外, 解码器执行完整 - 超完整 DWT, 而不是 IDWT (3120) 和 OCDWT (3140)。任一方法, 超完整表示 (3150) 包括适合于半像素 MCP 的相移值。

[0326] 在准备四分之一像素 MCP 中, 然后, 解码器使用  $2 \times 2$  Sinc 滤波器, 执行子像素内插 (3160)。这产生子像素内插基准 (3170)。内插基准帧 (3170) 的有效分辨率是 4CIF。

[0327] 解码器将内插的基准帧 (3170) 用于基准图像信息, 在 SL 域中执行 MCP (3180)。MCP<sub>SL</sub> (3180) 在 SL 域中产生预测 (3190)。同样地, 预测 (3190) 的有效分辨率为 QCIF。

[0328] b. 方案 B2

[0329] 根据图 32 中所示的方案 (方案 B2), 3D 子带解码器使用 SL 缓冲器 (3210) 来存储 SL 子带信息, 用作基准图片信息。解码器使用缓冲的 SL 子带信息, 执行 IDWT 操作 (3220)。在 IDWT 操作 (3220) 中, 将 SH 子带视为空 ( $SH = 0$ )。IDWT 操作 (3220) 从而从 SL 子带信息产生基准帧 (3230)。当 SL 子带的有效分辨率为 QCIF 时, 基准帧 (3230) 的有效分辨率为 CIF。

[0330] 解码器使用  $2 \times 2$  Sinc 滤波器, 执行子像素内插 (3240)。通过按 2 的倍数, 水平和垂直增加的分辨率, 这产生子像素内插基准帧 (3250)。内插基准帧 (3250) 的有效分辨率为 4CIF。

[0331] 然后, 解码器对内插基准帧 (3250) 执行 DSDWT (3260) 操作。这产生基准帧信息的超完整表示 (3270)。超完整表示 (3270) 包括适合于四分之一像素 MCP 的相移值。

[0332] 解码器将内插基准帧 (3270) 用于基准图片信息, 在 SL 域中执行 MCP (3280)。MCP<sub>SL</sub> (3280) 在 SL 域中产生预测 (3290)。同样地, 预测 (3290) 的有效分辨率为 QCIF。

[0333] c. 方案 B3

[0334] 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 SL



子带信息,用作基准图片信息。解码器使用缓冲的 SL 子带信息,执行 IDWT 操作 (3320)。在 IDWT 操作 (3320) 中,将 SH 子带视为空 ( $SH = 0$ )。从而 IDWT 操作 (3330) 由 SL 子带信息产生基准帧 (3330)。当 SL 子带的有效分辨率为 QCIF,基准帧 (3330) 的有效分辨率为 CIF。

[0335] 解码器使用  $2 \times 2$ Sinc 滤波器,执行子像素内插 (3340)。通过按 2 的倍数水平和垂直增加的分辨率,这产生子像素内插基准帧 (3350)。内插基准帧 (3350) 的有效分辨率为 4CIF。

[0336] 使用用于基准图片信息的内插基准帧 (3350),解码器在空间域中执行 MCP (3360)。MCP<sub>SD</sub> (3360) (具有半像素精度的运动矢量) 在空间域中产生预测 (3370)。在这种情况下,预测 (3370) 的有效分辨率为 CIF。

[0337] 解码器在预测 (3370) 上执行 DWT 操作 (3380),产生预测 (3370) 的低空间分辨率版本 (3390)。低空间分辨率预测 (3390) 的有效分辨率为 QCIF。

[0338] 3. 方案 C :SDIMCTF

[0339] 图 34 示例说明具有在译码期间专用的另外的计算和缓冲资源的 SDIMCTF。图 34 中的方案对应于图 26 的 SDIMCTF 解码器 (2600) 的一级 IMCTF。图 34 所示的操作出现在空间域中。与方案 B3 相比,方案 C 中的基准图片缓冲器为 SD 缓冲器,以及每级 IMCTF,解码器不执行 IDWT 和 DWT 操作。用这种方式,保留中间结果中的 SH 信息,用在稍后的 SDIMCTF 中。

[0340] 根据方案 C,3D 子带解码器使用 SD 缓冲器 (3410) 来存储空间域信息,用作基准图片信息。对第一逆时间变换,最终逆时间变换从 SL 子带信息产生 SD 缓冲器 (3410) 中的信息。对稍后的逆时间变换,从稍前逆时间变换,保留 SD 缓冲器 (3410) 中的信息,以及 SD 缓冲器 (3410) 中的信息可以包括由于功率谱泄漏生成的 SH 信息。SD 缓冲器 (3410) 中的信息的有效分辨率为 CIF。

[0341] 解码器使用  $2 \times 2$ Sinc 滤波器,执行子像素内插 (3420)。通过按 2 的倍数水平和垂直增加的分辨率,这产生子像素内插基准帧 (3430)。内插基准帧 (3430) 的有效分辨率为 4CIF。

[0342] 解码器将内插基准帧 (3430) 用于基准图片信息,在空间域中执行 MCP (3440)。MCP<sub>SD</sub> (3440) (通过半像素精度运动矢量) 在空间域中产生预测 (3450)。在这种情况下,预测 (3450) 的有效分辨率为 CIF。

[0343] 4. 方案 D :具有 SH 系数的 SDIMCTF

[0344] 为处理 SH 漂移问题,将带宽分配到比特流抽取器中的 SH 系数。这与当对 SL 分辨率视频输出解码时,简单地丢去 SH 子带信息的传统的空间可缩放性不同。

[0345] 为确定用于 SH 系数的最佳速率,测量用于降低低空间分辨率视频的失真的 SH 系数贡献。(相反,通过传统的空间可缩放实践,测量用于降低整个、初始空间分辨率视频的失真的 SH 系数贡献)。换句话说,即使仅当输出低空间分辨率视频时,考虑用于 SH 子带的增益因子,以便说明在降低 SL 视频中的失真中,SH 子带的用处(例如提高运动补偿预测精度)。用这种方式,SH 子带信息很可能向前推进到解码 SL 视频中。

[0346] 在方案 D 中,当解码 SL 视频时,使用用于 SH 子带的当前增益因子,就象解码器正抽取初始分辨率视频一样。允许 SL 和 SH 子带的系数包括在最后比特流中。另外,对于初始空间分辨率解码和 SL 解码的 SH 子带,计算不同增益因子。

[0347] 在方案 D 中,当解码 SL 视频时,3D 子带解码器使用图 34 中所示的解码框架,用于一级 IMCTF。基准图片缓冲器 (3410) 为 SD 缓冲器。对第一逆时间变换,SD 缓冲器包括使用 SL 子带信息和 (至少一些)SH 子带信息重构的基准图片信息,即使最终输出视频将有效地仅具有 SL 分辨率。也保留中间结果中的 SH 信息,用在稍后的 SDIMCTF 中。

[0348] 解码器使用  $2 \times 2$  Sinc 滤波器,执行子像素内插 (3420)。通过按 2 的倍数水平和垂直增加的分辨率,产生子像素内插基准帧 (3430)。解码器将内插基准帧 (3430) 用于基准图片信息,在空间域中执行 MCP (3440)。MCP<sub>sp</sub> (3440) (具有半像素精度运动矢量) 在空间域中产生预测 (3450)。

[0349] 5. 比较

[0350] 下表比较方案 A、B1、B2、B3、C 和 D 的特性。

[0351]

方案	仅 SL 系数	仅 SL 缓冲器	带内 INTP	带内 MCP
A	是	是	是	是
B1	是	是	是	是
B2	是	是	否	是
B3	是	否	否	否
C	是	否	否	否
D	否	否	否	否

[0352] 另外,当解码用于低空间分辨率输出的 SDMCTF 编码的视频时,解码器使用空间域内插、时间域运动补偿、空间域基准图片缓冲器的一些其他组合和 / 或使用 SH 信息来提高解码性能。

[0353] VI. 连续相位超完整子带传送

[0354] IBMCTF 和 IBIMCTF 的一个问题是如何降低小波变换的偏移方差运动估计和补偿的影响 - 图像域中的偏移 / 运动不直接映射到变换域。一些技术将整数像素运动映射到变换域 (见例如 Y. A. Andreopoulos et al., “A New Method for Complete-to-Overcomplete Discrete Wavelet Transforms”, Proc. IEEE Conf. Digital Signal Proc. (2002) 和 X. Li, L. Kerofsky, S. Lei “All-phase Motion Compensated Prediction for High Performance Video Coding”, Proc. ICIP. vol. 3, pp. 538-541 (2001))。该节描述用于将部分像素运动映射到变换域中的相位变化的技术。通过这样做,能更高精度地执行运动估计和补偿,共能显著地影响编码性能。

[0355] 该节通过一级、一维小波变换的例子,示例说明连续相位超完整子带变换 (“CPOST”)。p- 像素 CPOST 的分解过程具有四个顺序级:(1) p- 像素内插,(2) 连续偏移,(3) 下采样,以及 (4) 小波变换。

[0356] 假定用于 ( $p = 1, 1/2, 1/4, 1/8, \dots$ ) 的连续 p 像素相位超完整变换。当  $p = 1$  时,CPOST 是整数像素相位超完整情形。其他 p 值提供子像素相位超完整情形。假定输入

信号是离散时间域序列  $\{x(n)\}$ ,  $n = 0, 1, 2, 3, \dots$ 。编码器或解码器首先执行  $\{x(n)\}$  的  $p$ -像素内插, 然后连续偏移  $p$ -像素内插序列  $\frac{2}{p}-1$  次以及下采样偏移的序列。这产生  $\{x(n)\}$  的连续相位偏移序列集, 如由  $\{x_p^i(n)\}$ ,  $i = 0, 1, \dots, \frac{2}{p}-1$  所述。最后, 编码器或解码器在  $x_p^i(n)$  上执行小波分解变换。

[0357] 对  $p = 1/2$  的情形, 连续相位偏移序列如下。  $\{y(n)\}$  是由  $\{x(n)\}$  的  $1/2$  像素内插点组成的序列。

[0358]  $\dots x(n)y(n)x(n+1)y(n+1)x(n+2)y(n+2)x(n+3)y(n+3)\dots$

[0359] 0-phase  $\dots x(n+1) \quad x(n+1) \quad x(n+2)\dots$

[0360]  $1/2$ -phase  $\dots y(n+1) \quad y(n+1) \quad y(n+2) \dots$

[0361] 1-phase  $\dots x(n+1) \quad x(n+2) \quad x(n+3)\dots$

[0362]  $3/2$ -phase  $\dots y(n+1) \quad y(n+2) \quad y(n+3)\dots$

[0363] 通过分别沿水平和垂直方向执行两个顺序的 1D CPOSTs, 能实现 2D CPOST。在小波域中, 使用预测滤波器集  $F_p^i(z)$   $i = 0, 1, \dots, \frac{2}{p}-1$ , 能获得连续  $p$ -像素相位偏移序列。这些滤波器等效于连续地偏移子像素内插序列和下采样这些偏移的序列。  $p$ -像素 CPOST 的分解过程能作为滤光器的应用提供给  $\{x(n)\}$ , 后跟用低通滤波器  $G_L(z)$  处理结果以及通过高通滤波器  $G_H(z)$  单独地用于小波分解变换, 然后下采样。对于  $(i = 0, 1, 2, \dots, \frac{2}{p}-1)$  的  $l_p^i$  和  $h_p^i$  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  $z$  变换,  $L_p^i(z)$  和  $H_p^i(z)$  分别是  $l_p^i(n)$  和  $h_p^i(n)$  的  $z$  变换。

$$[0364] \quad X_p^i(z) = F_p^i(z)X(z) \quad i = 0, 1, 2, \dots, \frac{2}{p}-1 \quad (20)$$

$$[0365] \quad L_p^i(z) = \frac{1}{2} \left[ G_L(z^{1/2})X_p^i(z^{1/2}) + G_L(-z^{1/2})X_p^i(-z^{1/2}) \right]$$

$$[0366] \quad H_p^i(z) = \frac{1}{2} \left[ G_H(z^{1/2})X_p^i(z^{1/2}) + G_H(-z^{1/2})X_p^i(-z^{1/2}) \right] \quad (21)$$

$$[0367] \quad i = 0, 1, 2, 3, \dots, \frac{2}{p}-1$$

[0368]  $p$ -像素 CPOST 的分解过程对应于矩阵表达式:

$$[0369] \quad \begin{bmatrix} L_p^0(z) \\ H_p^0(z) \end{bmatrix} = \frac{1}{2} \begin{bmatrix} G_L(z^{1/2}) & G_L(-z^{1/2}) \\ G_H(z^{1/2}) & G_H(-z^{1/2}) \end{bmatrix} \begin{bmatrix} X(z^{1/2}) \\ X(-z^{1/2}) \end{bmatrix} \quad (22)$$

$$[0370] \quad \begin{bmatrix} L_p^i(z) \\ H_p^i(z) \end{bmatrix} = \frac{1}{2} \begin{bmatrix} G_L(z^{1/2}) & G_L(-z^{1/2}) \\ G_H(z^{1/2}) & G_H(-z^{1/2}) \end{bmatrix} \begin{bmatrix} X_p^i(z^{1/2}) \\ X_p^i(-z^{1/2}) \end{bmatrix} \quad (23)$$

$$[0371] \quad i = 1, 2, 3, \dots, \frac{2}{p}-1$$

[0372] 表达式 (22) 提供初始序列的小波分解变换, 而没有偏移。表达式 (23) 提供连续子相位偏移序列的小波分解变换。表达式 (23) 和 (24) 组成 CPOST。关于  $L_p^i(z)$ ,  $H_p^i(z)$  和  $L_p^0(z)$ ,  $H_p^0(z)$  间的关系。从 (20) 至 (23):

$$[0373] \quad \begin{bmatrix} L_p^i(z) \\ H_p^i(z) \end{bmatrix} = \frac{1}{2} \begin{bmatrix} G_L(z^{1/2}) & G_L(-z^{1/2}) \\ G_H(z^{1/2}) & G_H(-z^{1/2}) \end{bmatrix} \begin{bmatrix} F_p^i(z^{1/2}) & 0 \\ 0 & F_p^i(-z^{1/2}) \end{bmatrix} \begin{bmatrix} X(z^{1/2}) \\ X(-z^{1/2}) \end{bmatrix} \quad (24)$$

[0374] 从 (220 和 (24) :

$$[0375] \begin{bmatrix} L_p^i(z) \\ H_p^i(z) \end{bmatrix} = G(z) \begin{bmatrix} F_p^i(z^{1/2}) & 0 \\ 0 & F_p^i(-z^{1/2}) \end{bmatrix} G^{-1}(z) \begin{bmatrix} L^0(z) \\ H^0(z) \end{bmatrix} \quad (25)$$

[0376] 其中

$$[0377] G(z) = \begin{bmatrix} G_L(z^{1/2}) & G_L(-z^{1/2}) \\ G_H(z^{1/2}) & G_H(-z^{1/2}) \end{bmatrix}$$

[0378] 以及其中,  $G^{-1}(z)$  是  $G(z)$  的逆矩阵。

[0379] 根据 (25), 从初始序列 (即 0 相位偏移序列) 的小波变换, 能获得变换矩阵  $T_p^i(z)$  和逆变换矩阵  $IT_p^i(z)$  以便获得子像素移相序列的小波变换。

$$[0380] T_p^i(z) = G(z) \begin{bmatrix} F_p^i(z^{1/2}) & 0 \\ 0 & F_p^i(-z^{1/2}) \end{bmatrix} G^{-1}(z) \quad (26)$$

$$[0381] IT_p^i(z) = [T_p^i(z)]^{-1} = G^{-1}(z) \begin{bmatrix} F_p^i(z^{1/2}) & 0 \\ 0 & F_p^i(-z^{1/2}) \end{bmatrix}^{-1} G(z) \quad (27)$$

[0382] 从 (26), 如果存在表达式 (28), 在超完整小波变换后的子像素内插的结果等于在离散时间域中, 子像素内插后的超完整小波变换的结果。否则, 在超完整小波变换后的子像素内插的结果不等于在离散时间域中, 子像素内插后的超完整小波变换的结果。

$$[0383] \begin{bmatrix} F_p^i(z^{1/2}) & 0 \\ 0 & F_p^i(-z^{1/2}) \end{bmatrix} G^{-1}(z) = G^{-1}(z) \begin{bmatrix} F_p^i(z^{1/2}) & 0 \\ 0 & F_p^i(-z^{1/2}) \end{bmatrix} \quad (28)$$

[0384] 通常, 表达式 (28) 不存在, 这建议不交换用于带内视频编码的子像素内插和超完整小波变换的顺序。根据 (25), 然而, 能由其滤波变换系数导出初始序列的 CPOST 系数。

[0385] CPOST 方法不仅实现整数像素相位超完整小波变换, 而且实现子像素相位超完整小波变换。用于具有 5-3 小波变换的  $p = 1/2$  和 6- 抽头内插滤波器的连续相位超完整变换矩阵系数如下。用于变换矩阵  $T_{1/2}^1(z)$ ,  $T_{1/2}^2(z)$  和  $T_{1/2}^3(z)$  的每个元素的系数如下:

$$[0386] T_{1/2}^1(0, 0) = [0.0098-0.09570.88670.2227-0.0215-0.0020]$$

$$[0387] T_{1/2}^1(0, 1) = [-0.00240.0488-0.22900.3242-0.14400.00200.0005]$$

$$[0388] T_{1/2}^1(1, 0) = [0.0391-0.23440.3438-0.1406-0.0078]$$

$$[0389] T_{1/2}^1(1, 1) = [-0.00980.15820.3633-0.53520.02150.0020]$$

$$[0390] T_{1/2}^2(0, 0) = [-0.06250.56250.5625-0.0625]$$

$$[0391] T_{1/2}^2(0, 1) = [0.015625-0.250.46875-0.250.015625]$$

$$[0392] T_{1/2}^2(1, 0) = [-0.250.5-0.25]$$

$$[0393] T_{1/2}^2(1, 1) = [0.0625-0.5625-0.56250.0625]$$

$$[0394] T_{1/2}^3(0, 0) = [-0.0020-0.02150.22270.8867-0.09570.0098]$$

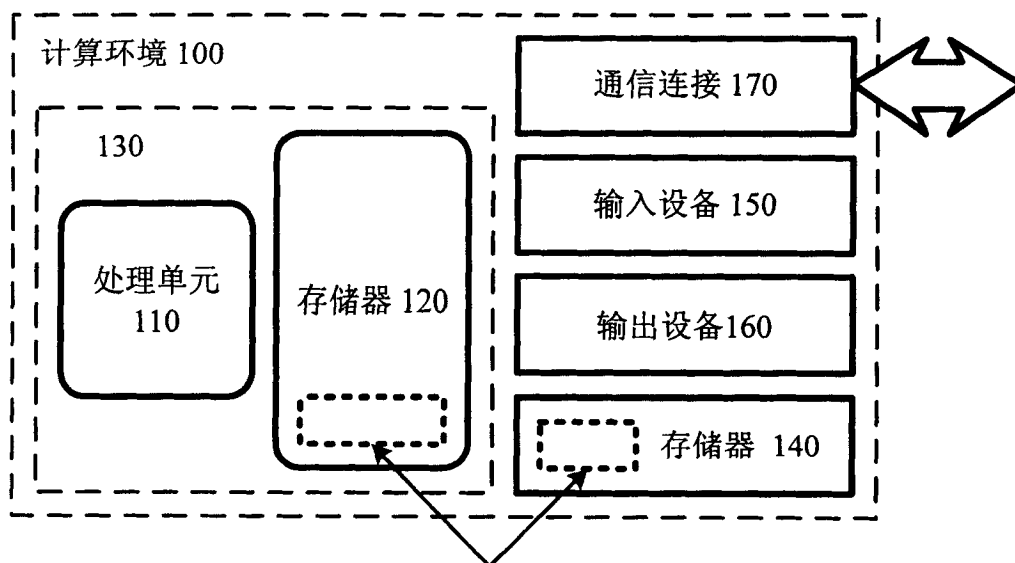
$$[0395] T_{1/2}^3(0, 1) = [0.00050.0020-0.14400.3242-0.22900.0488-0.0024]$$

$$[0396] T_{1/2}^3(1, 0) = [-0.0078-0.14060.3438-0.23440.0391]$$

$$[0397] T_{1/2}^3(1, 1) = [0.00200.02148-0.53520.36330.1582-0.0098]$$

[0398] 鉴于可以应用所公开的发明的原理的许多可能的实施例, 应意识到所示的实施例

仅是本发明的优选例子, 不应当视为限制本发明的原理。相反, 本发明的原理由下述权利要求定义。因此, 要求所有落在这些权利要求的范围和精神内均视为本发明。



通过自适应运动补偿更新，实现视频编码器和/或解码器的软件180

图 1

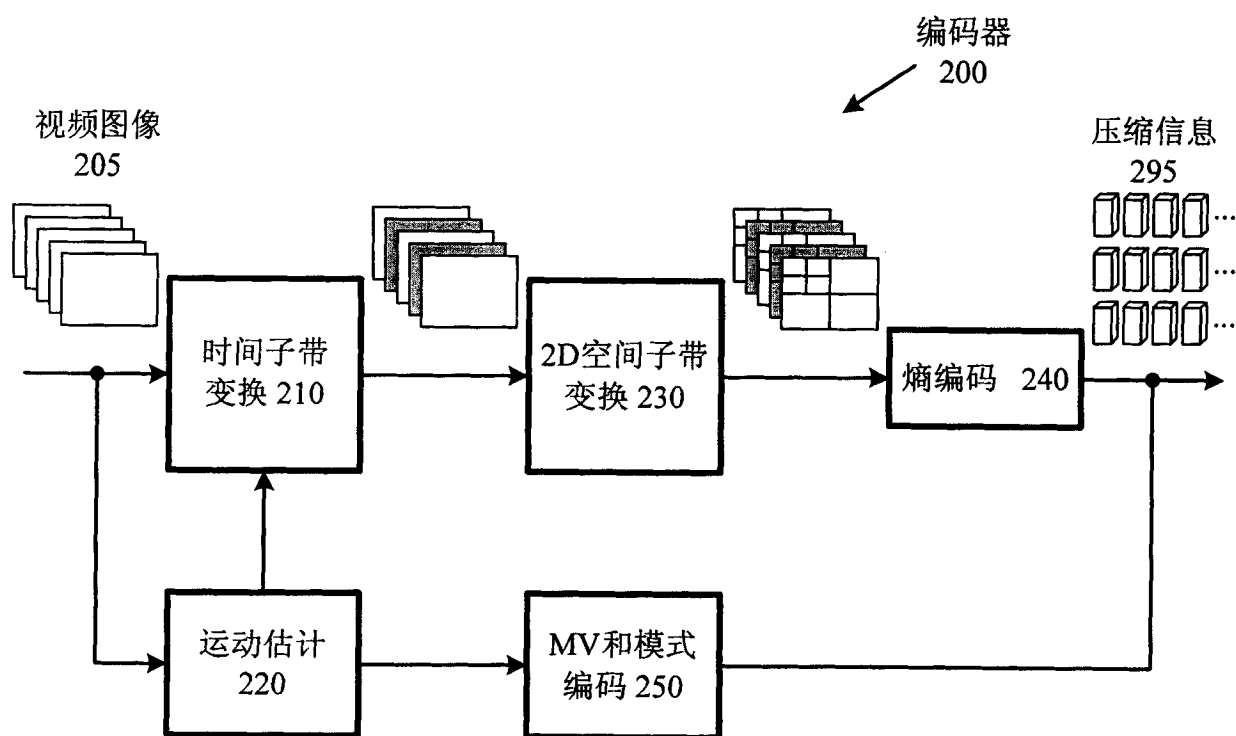


图 2

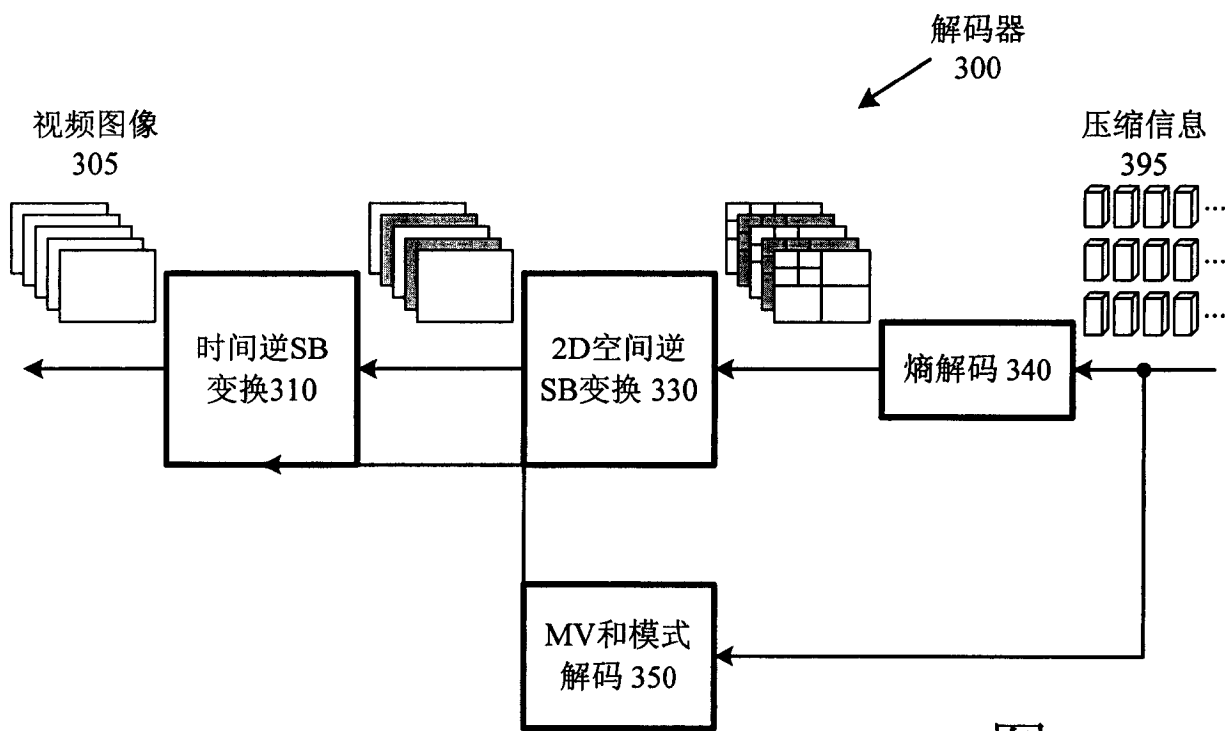


图 3

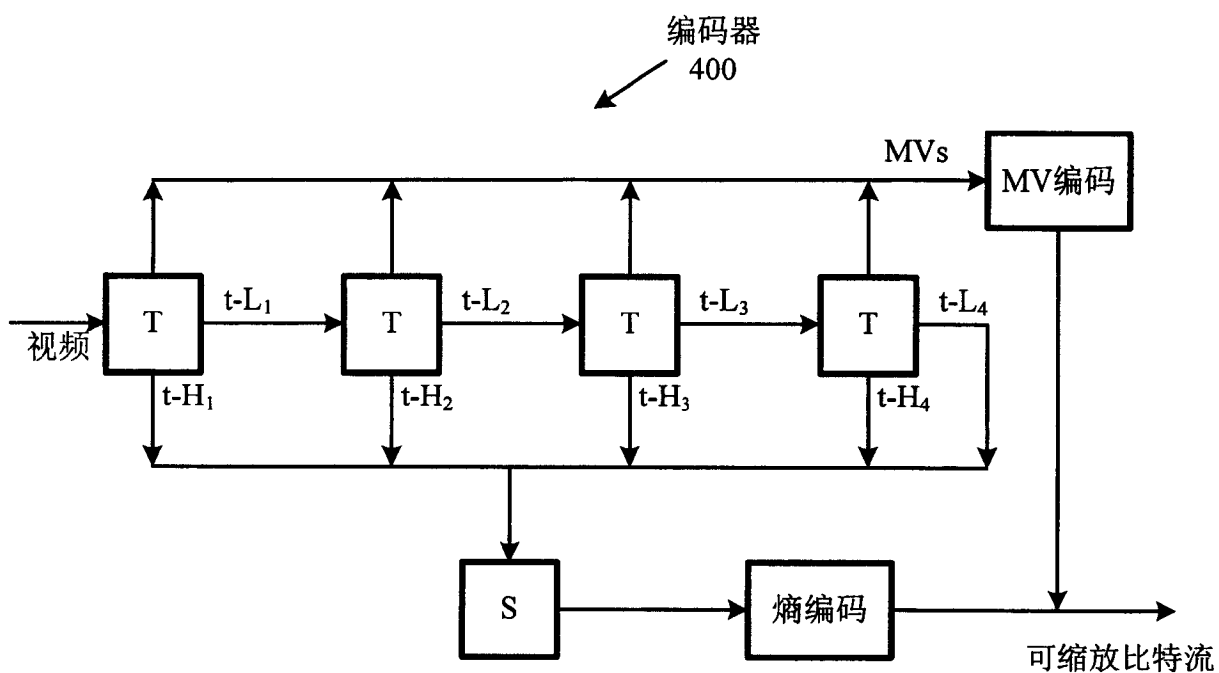


图 4

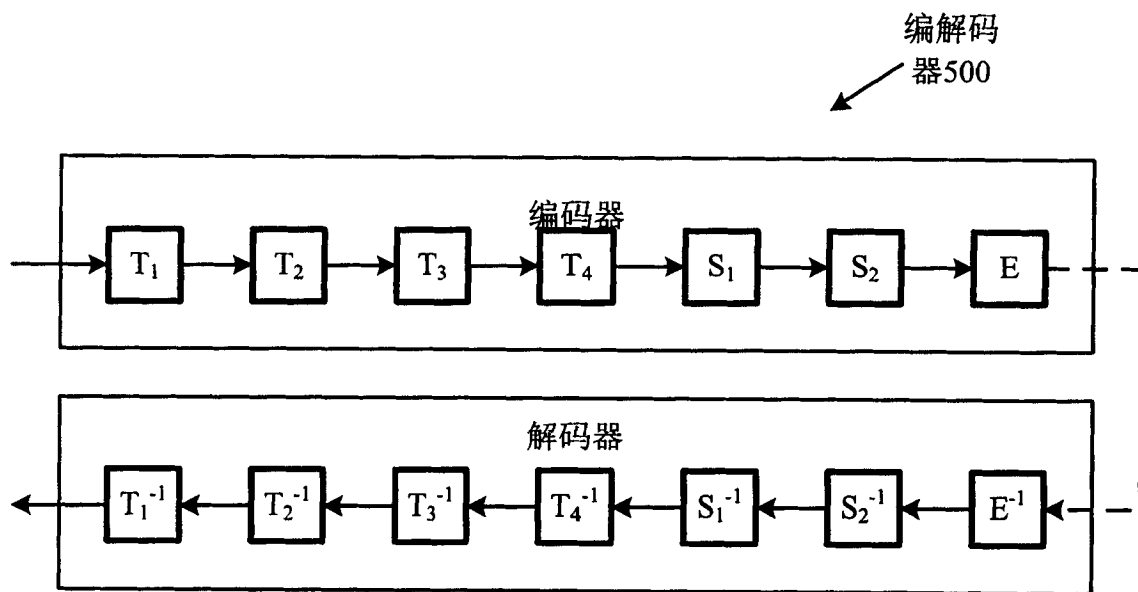


图 5

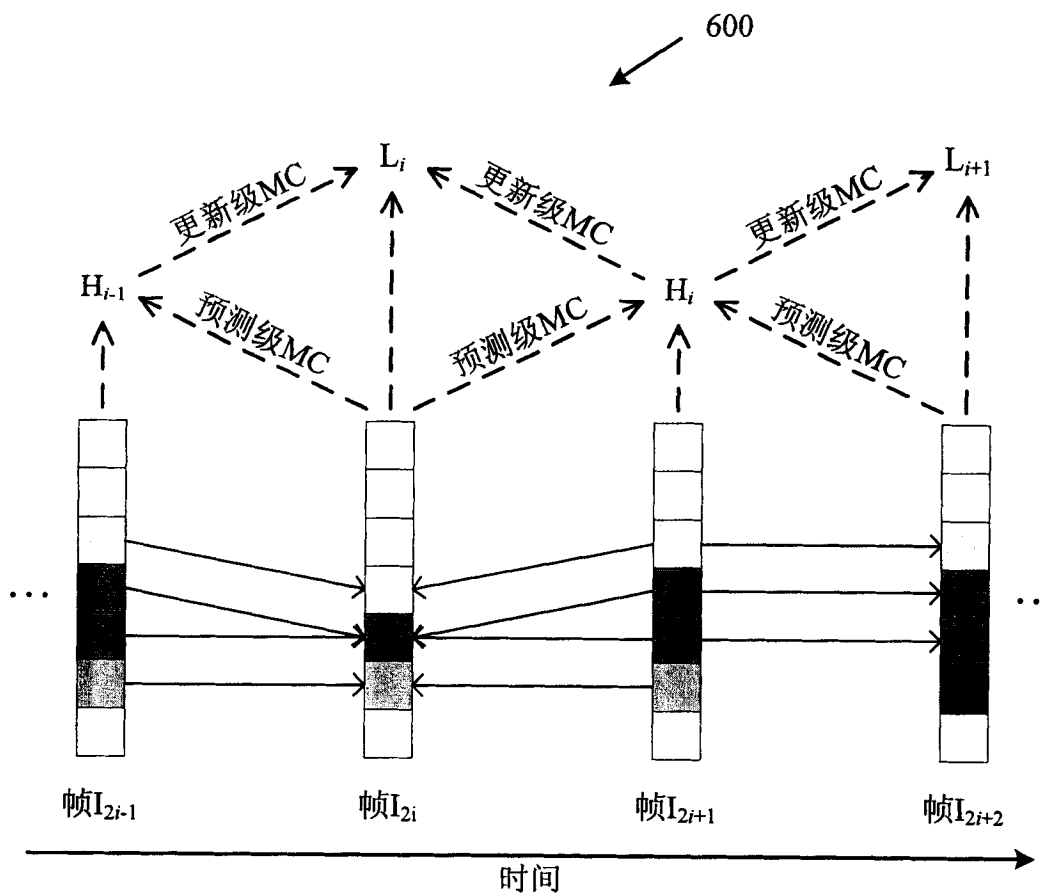


图 6



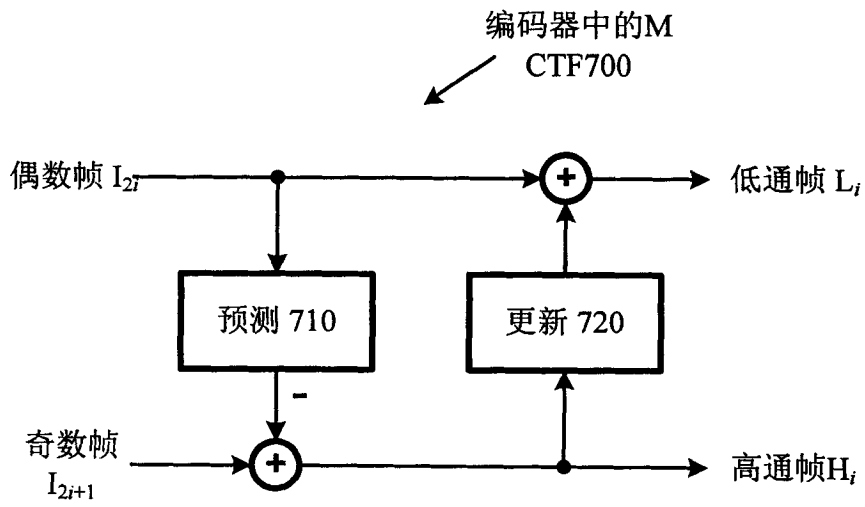


图 7

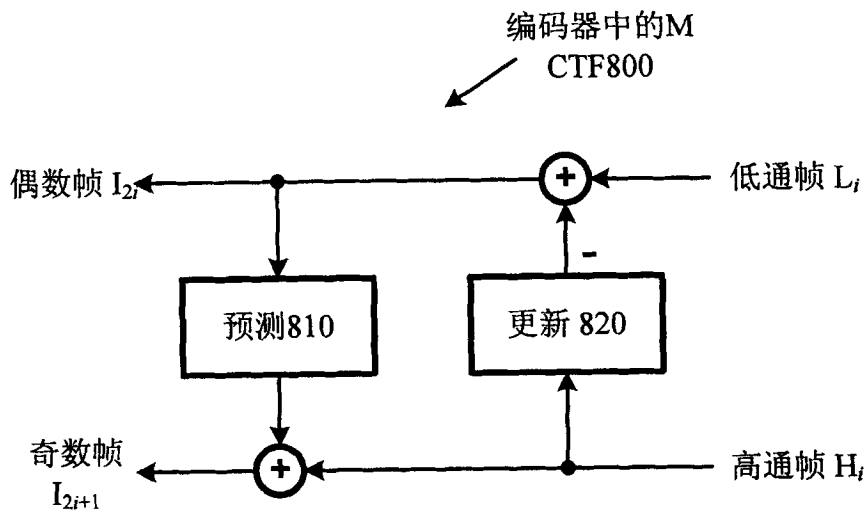


图 8

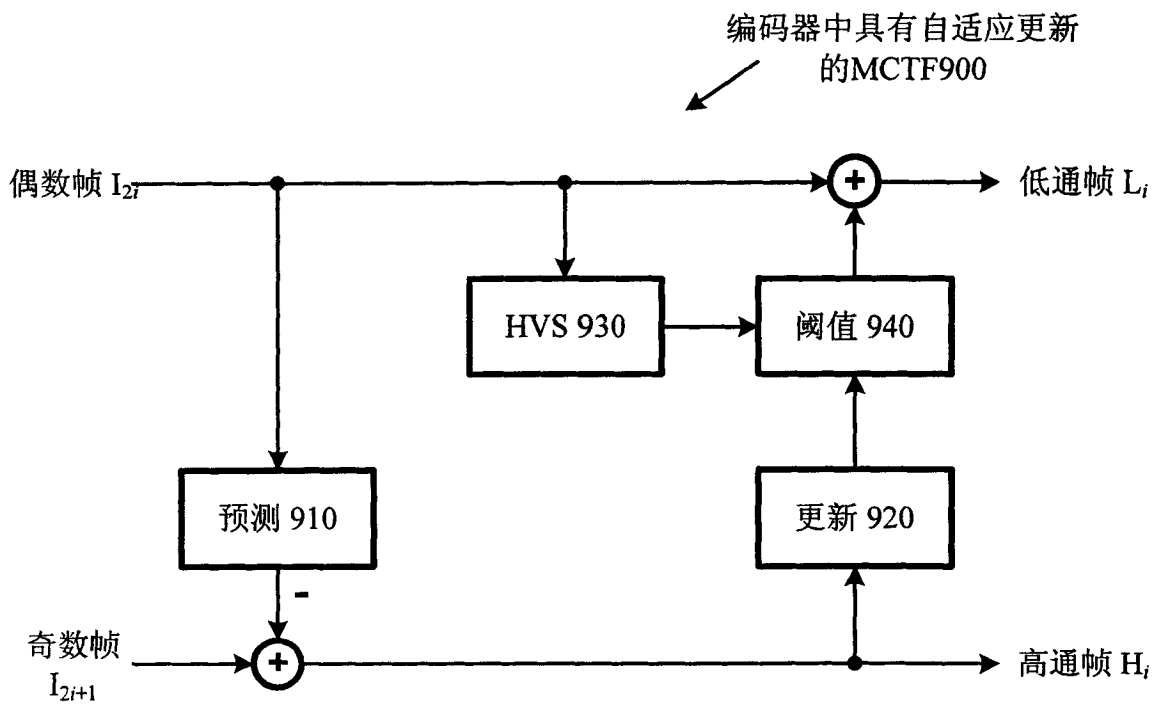


图 9

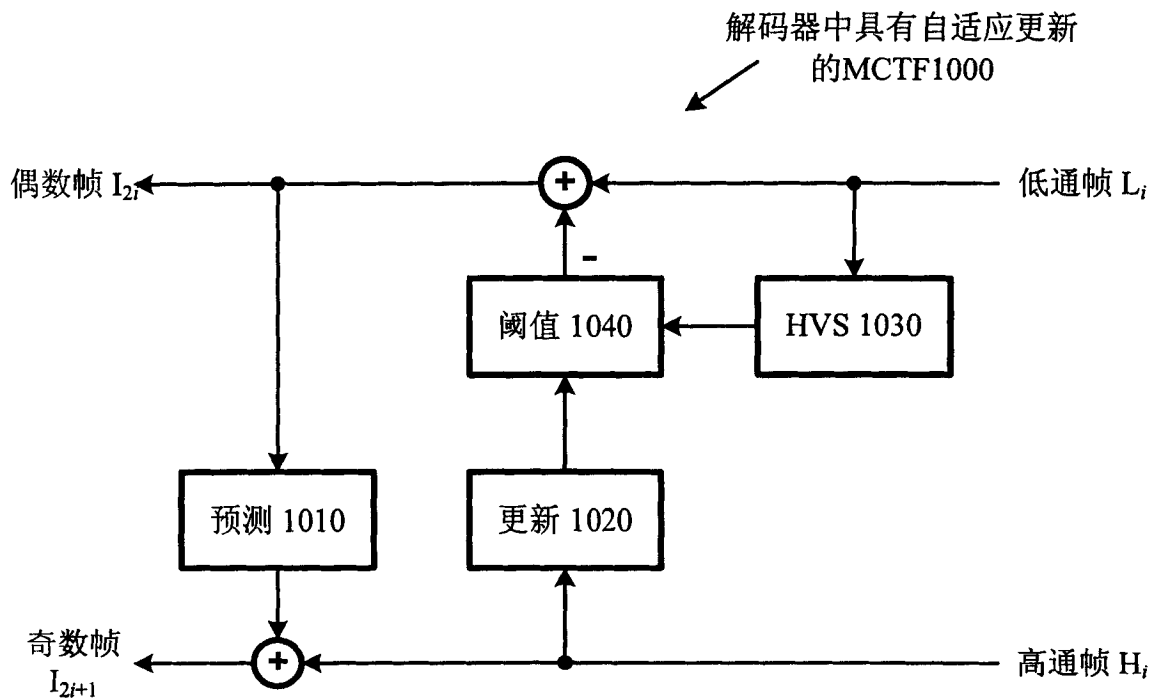


图 10

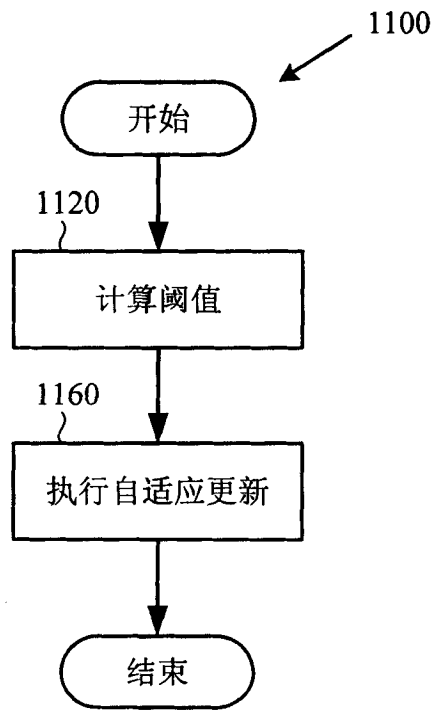


图 11

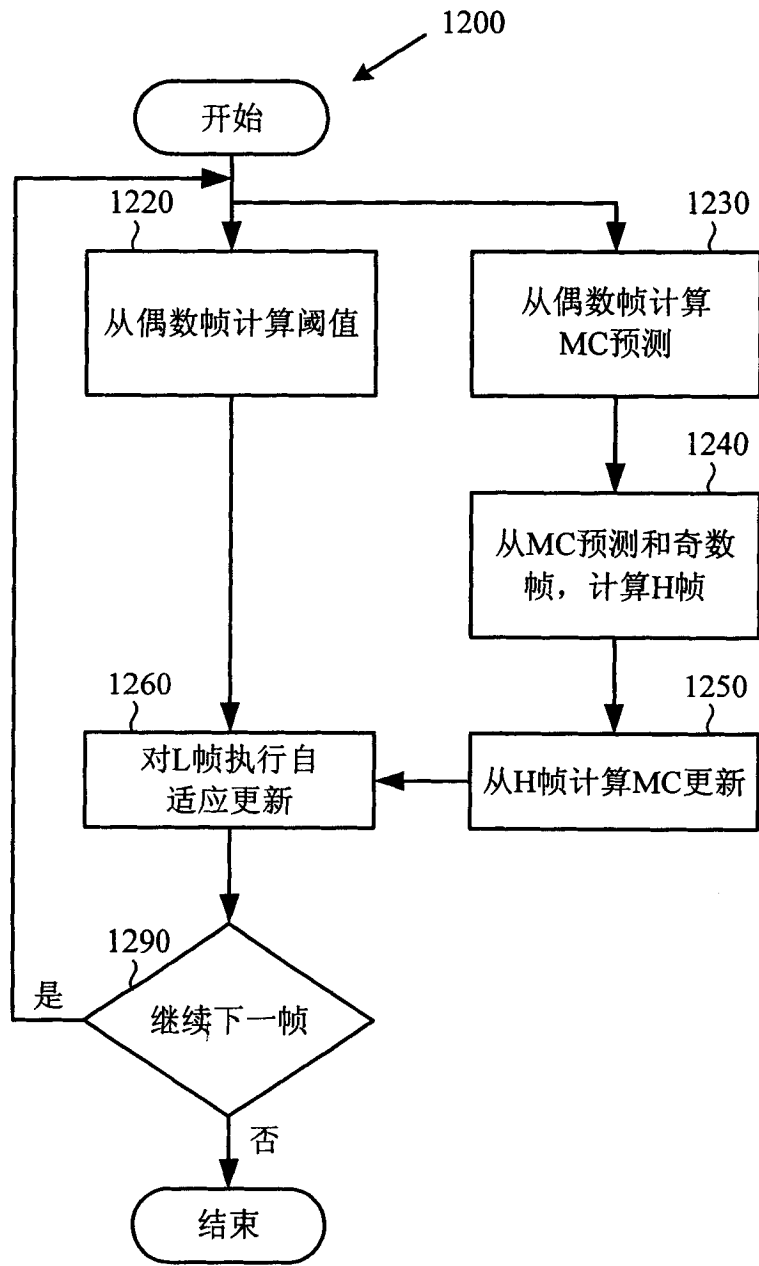


图 12

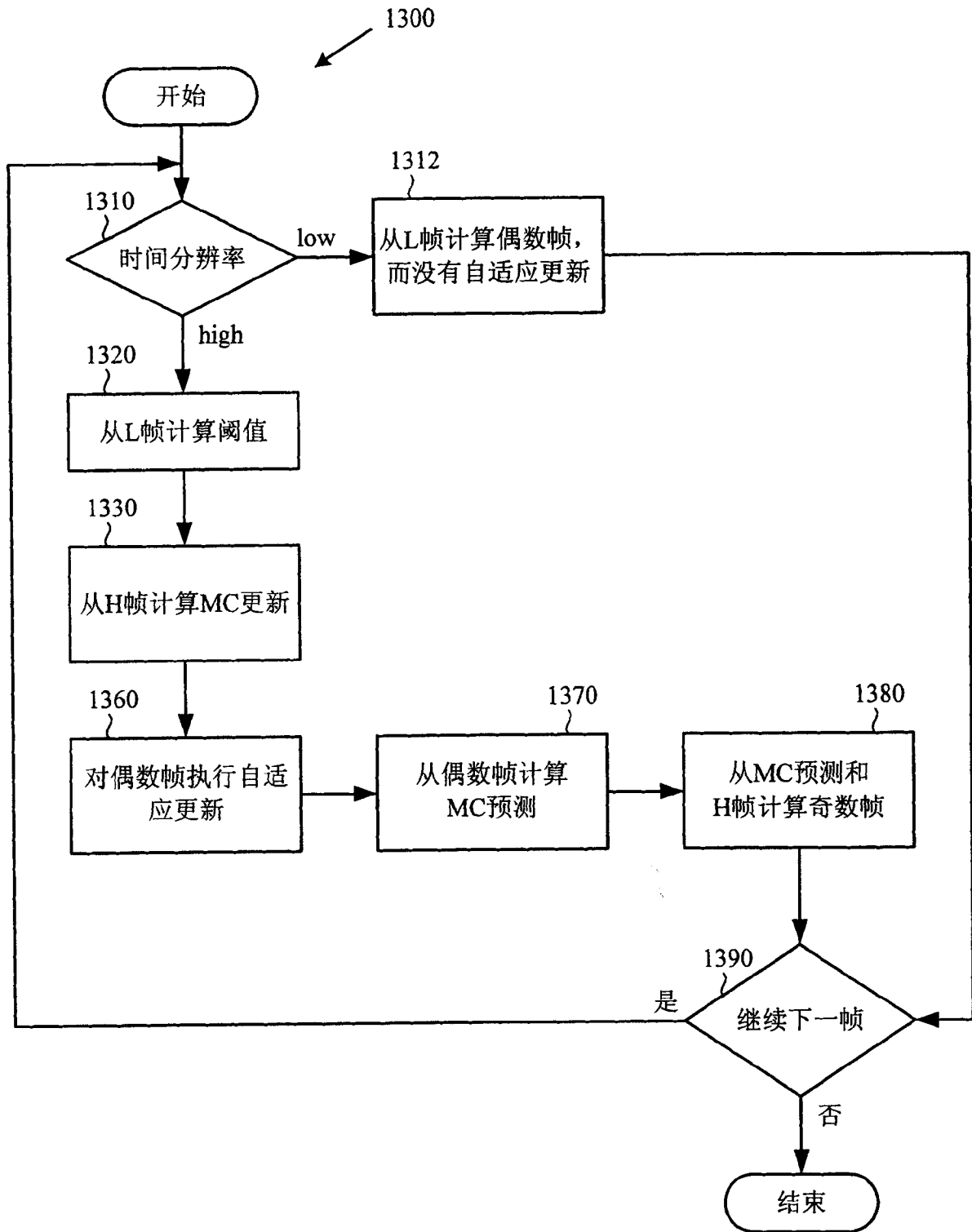


图 13

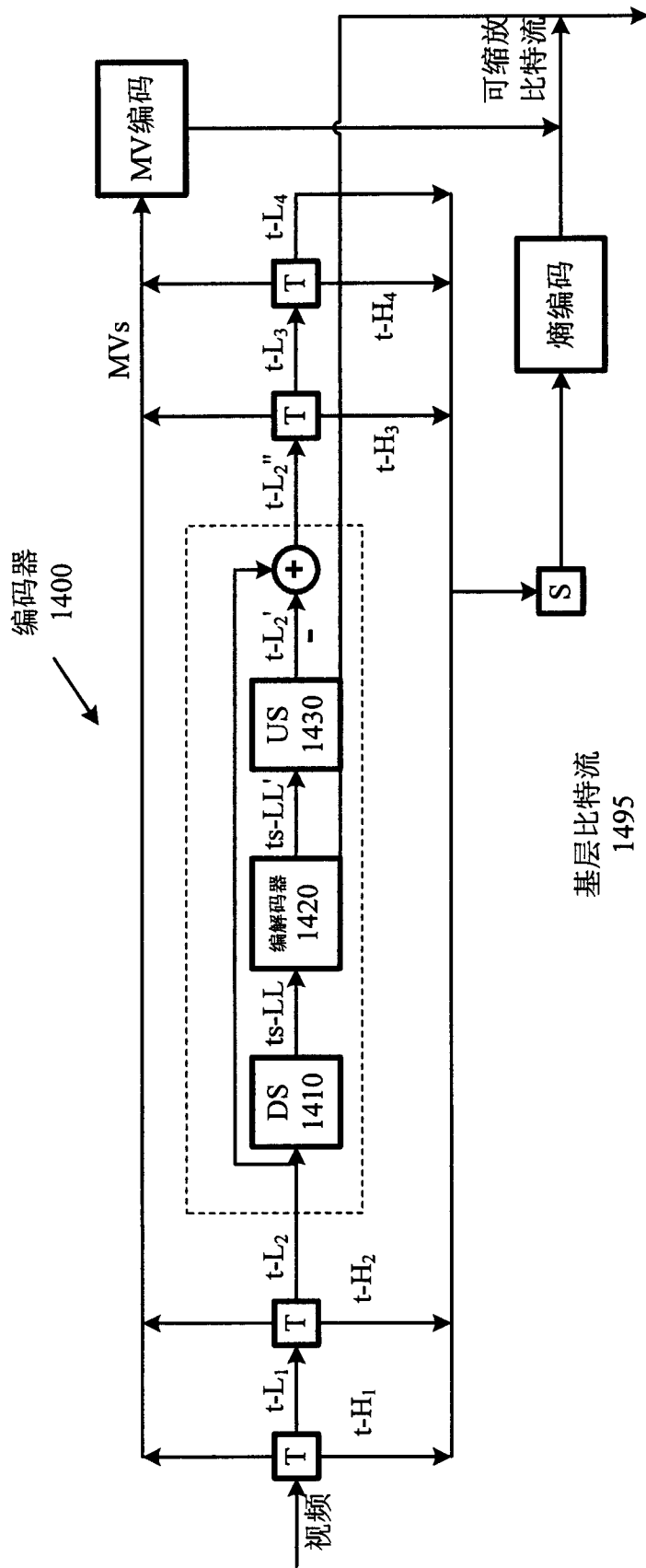


图 14

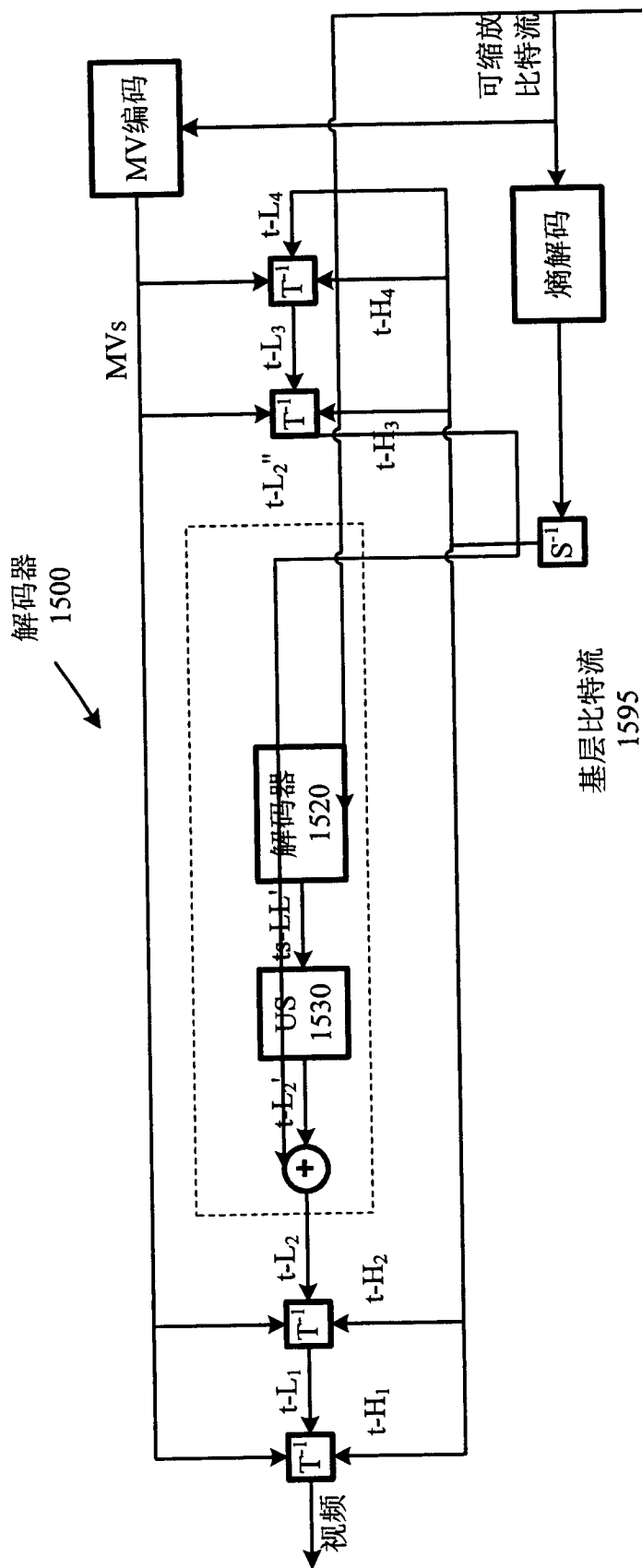


图 15

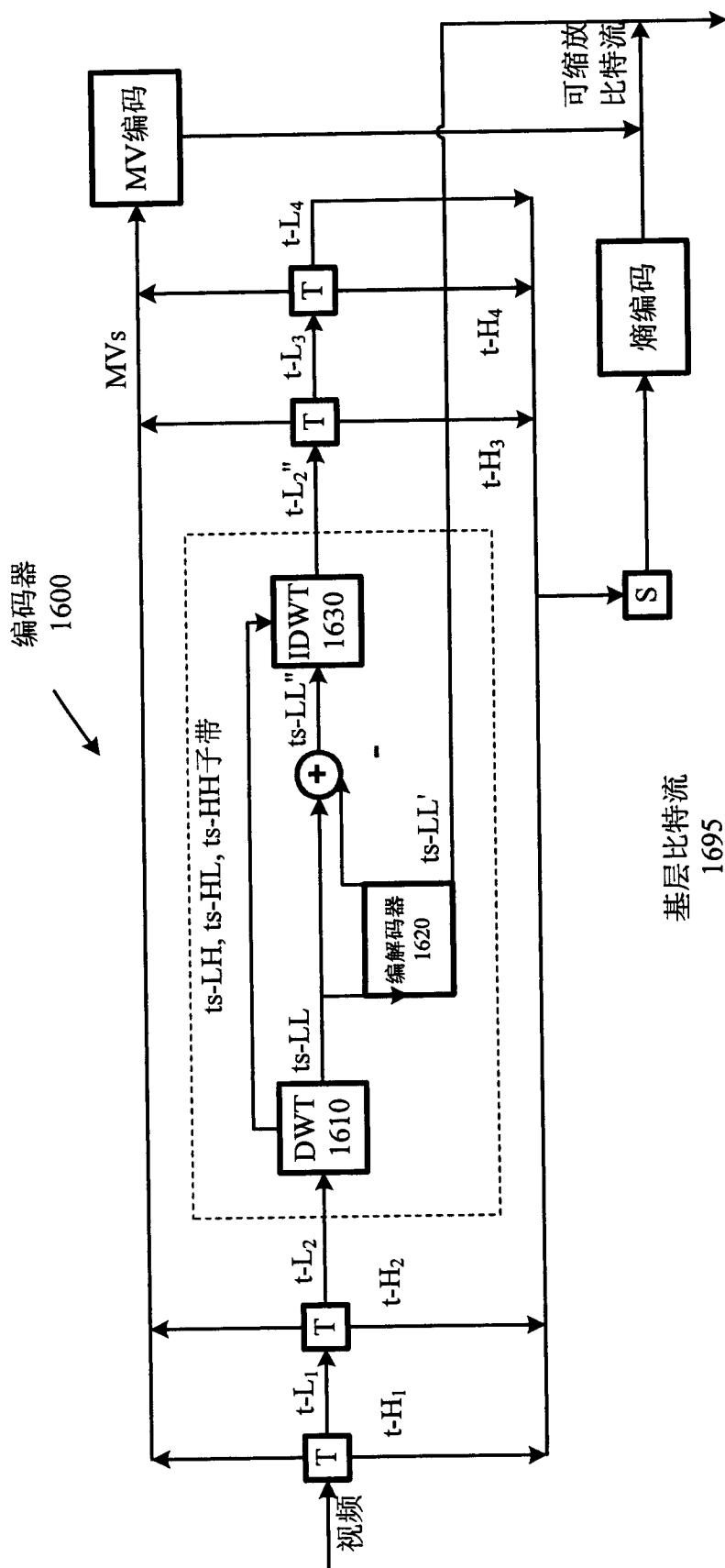


图 16



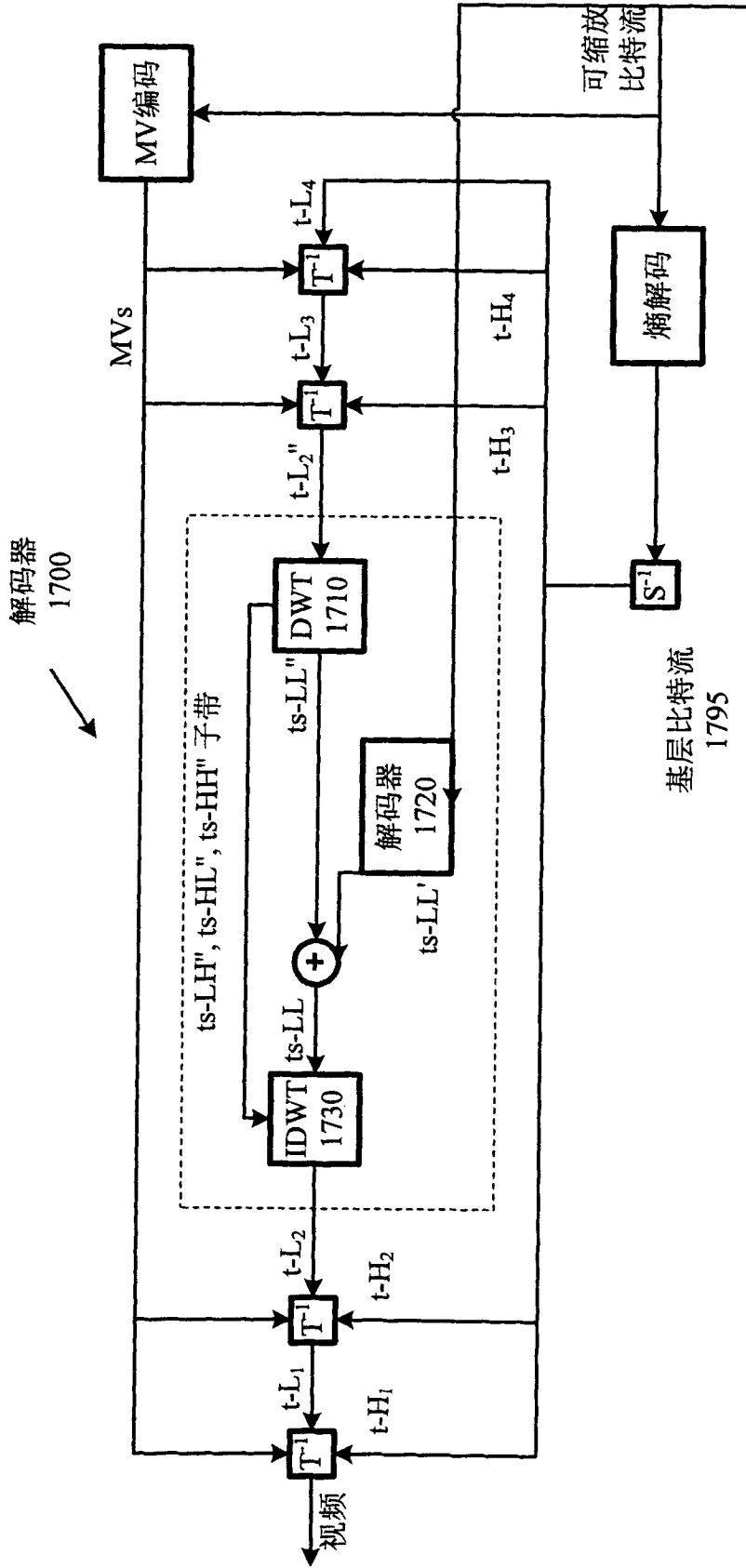


图 17

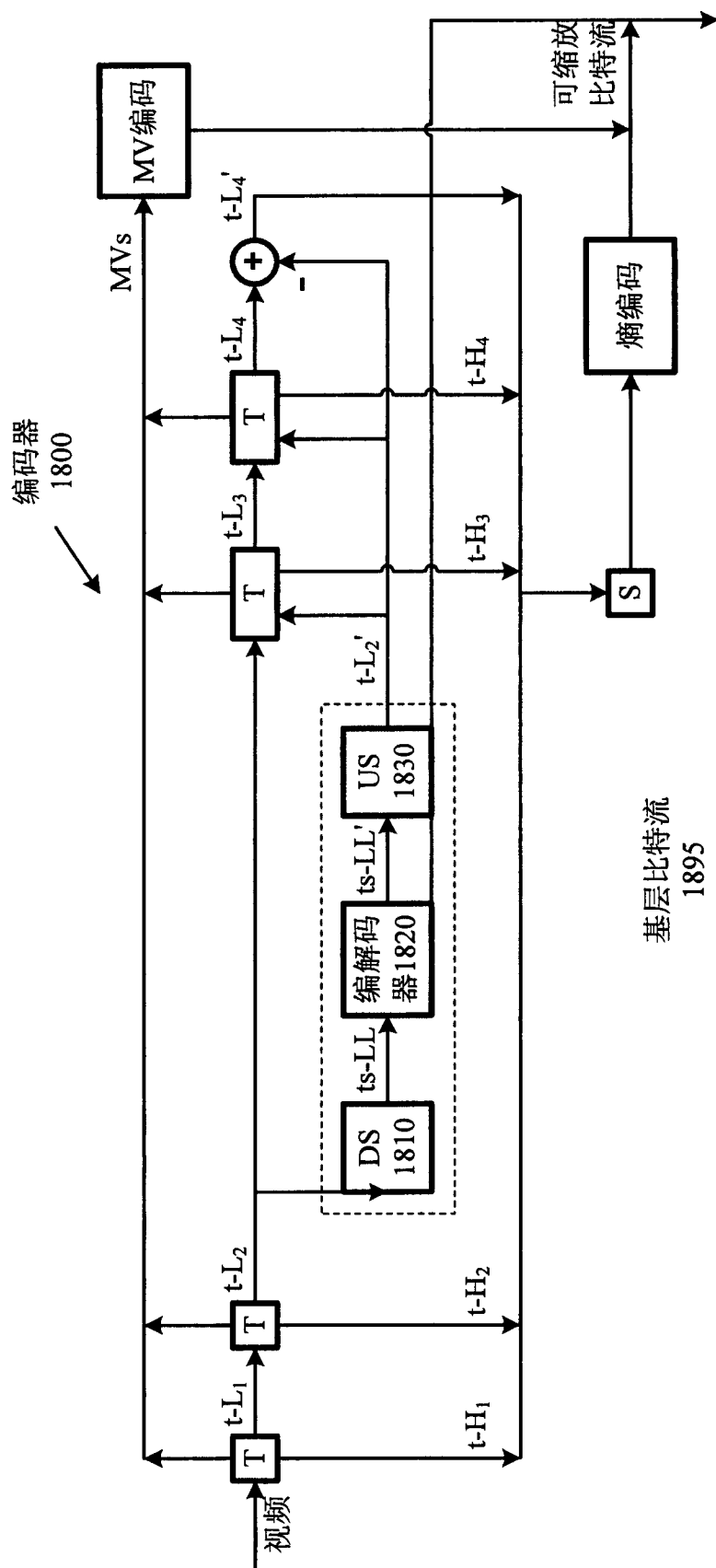


图 18

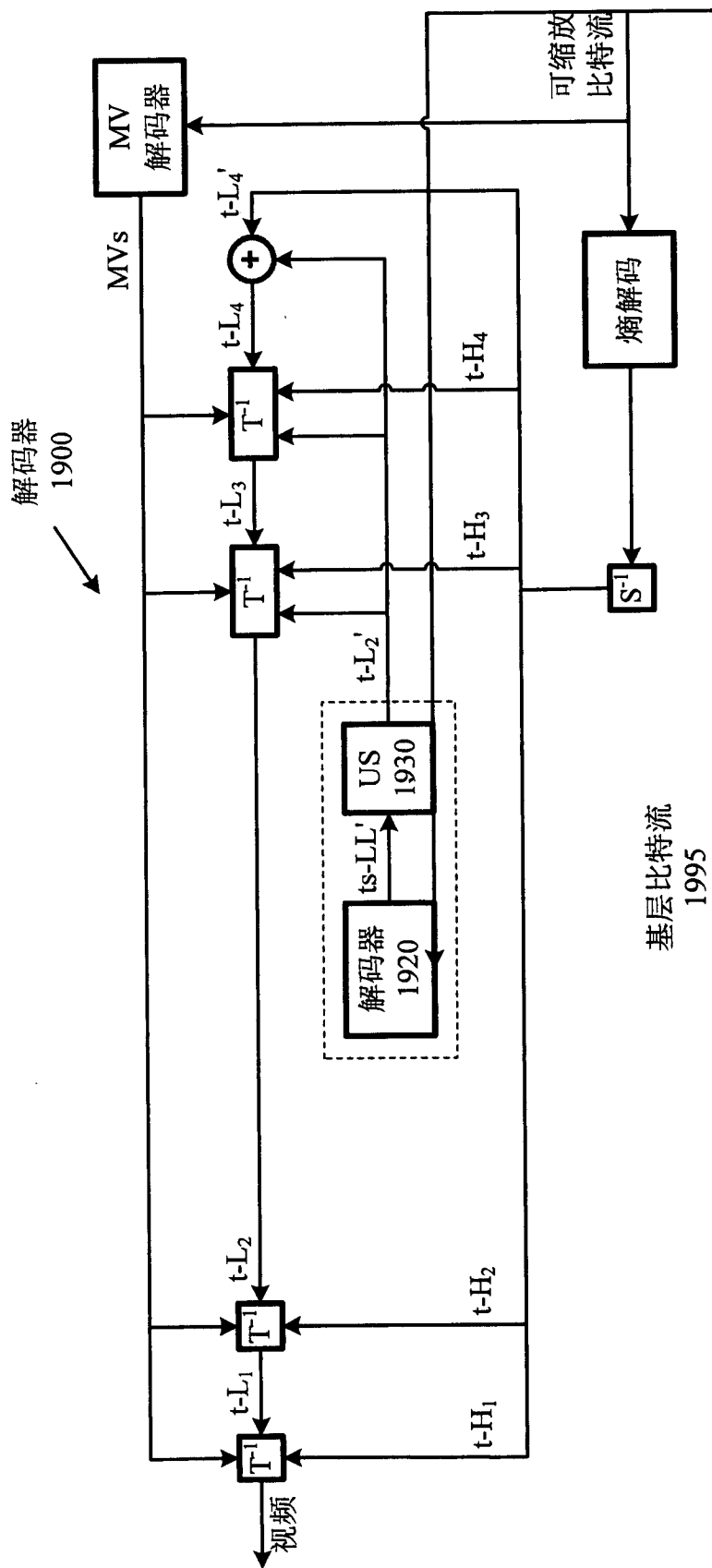


图 19

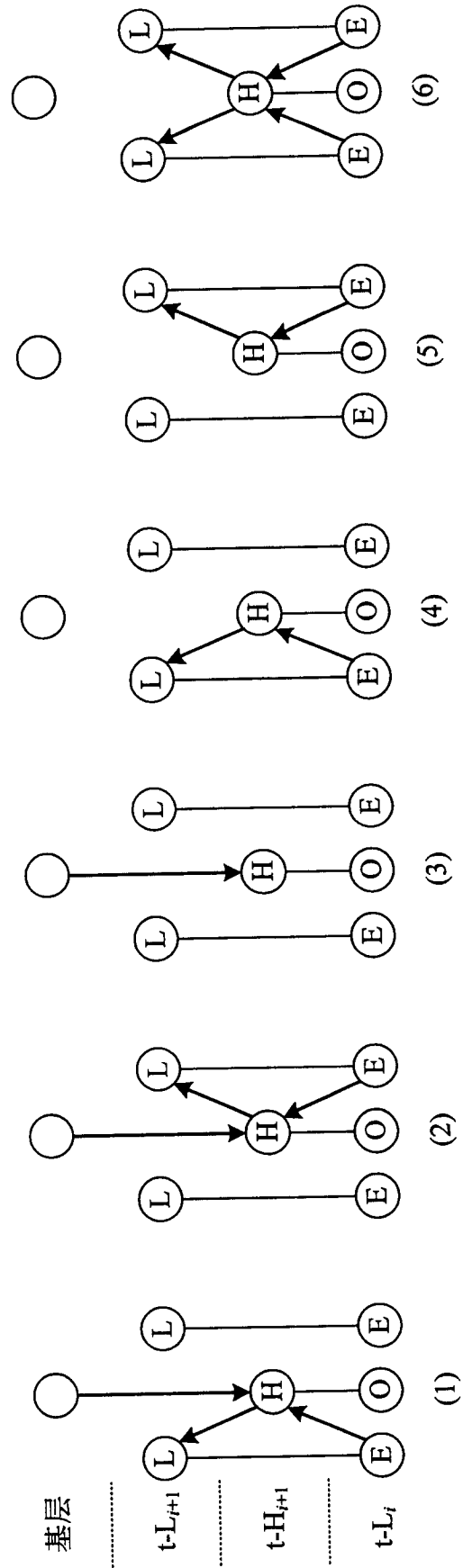


图 20

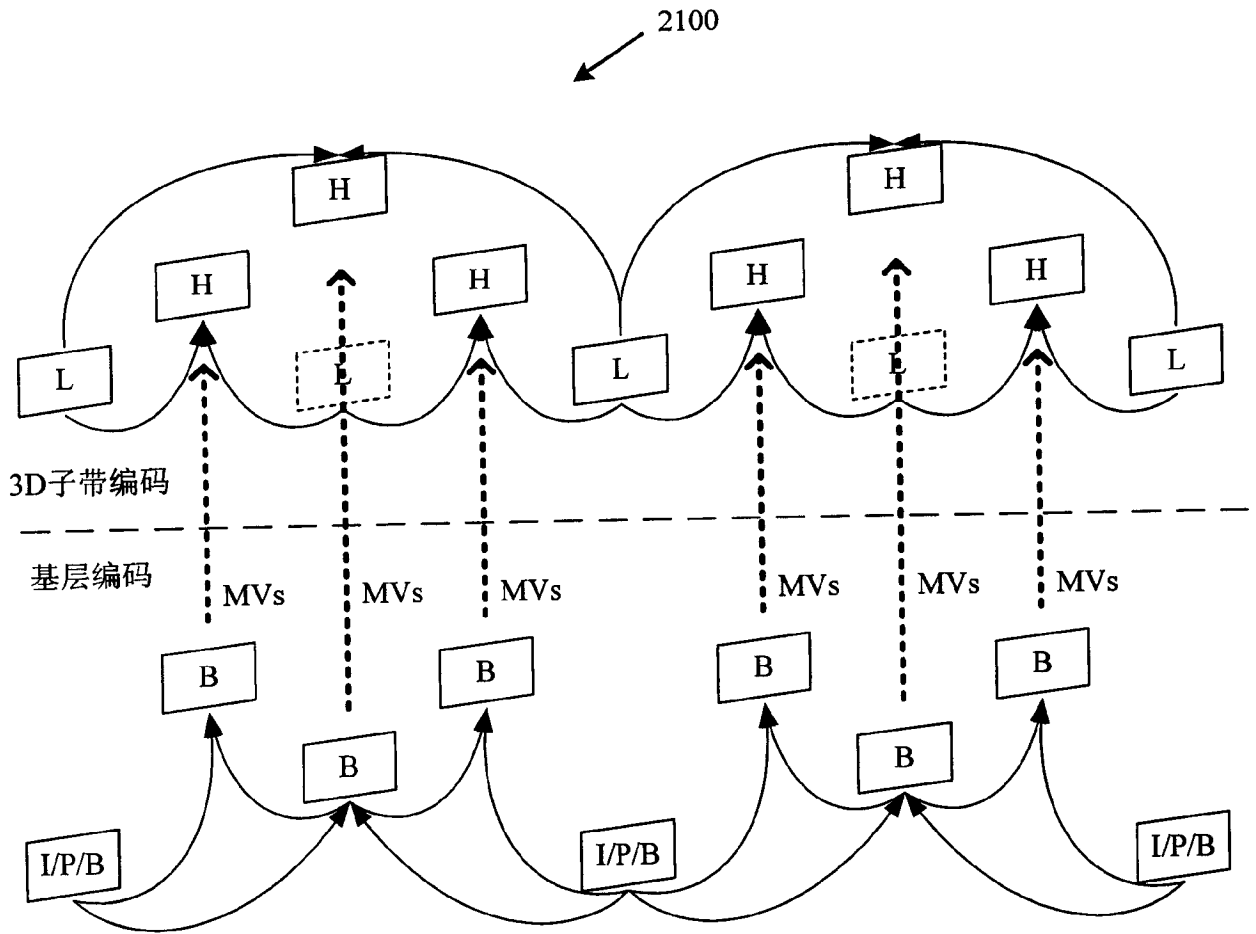


图 21

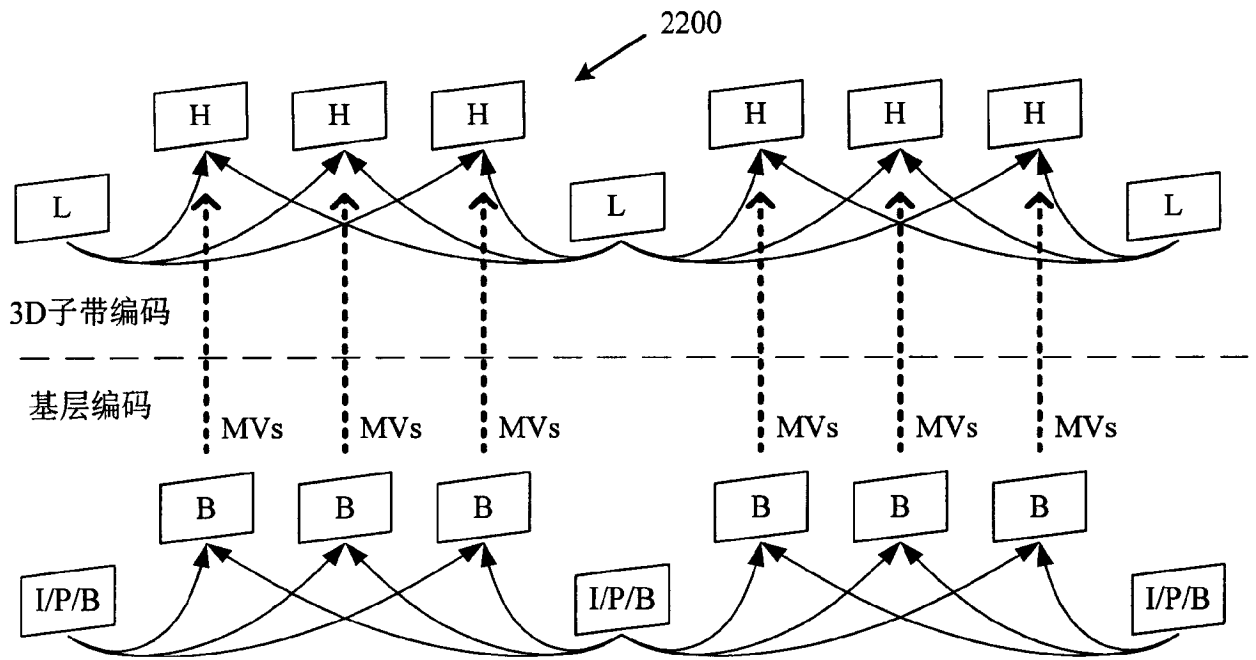


图 22

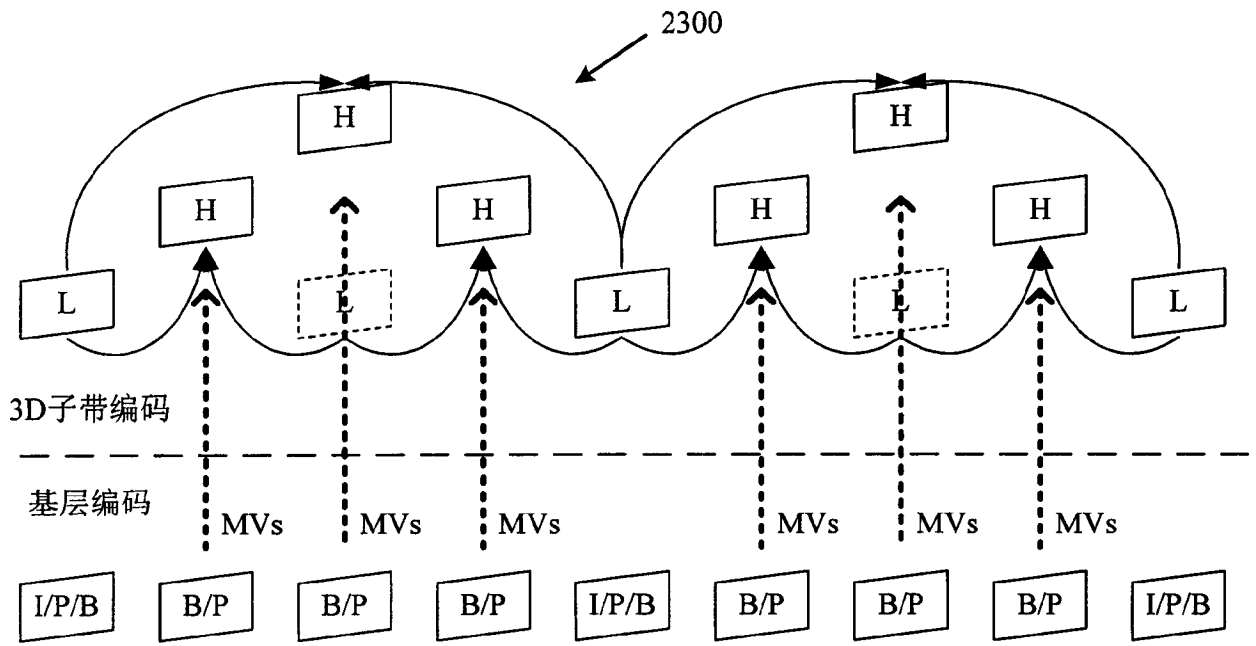


图 23

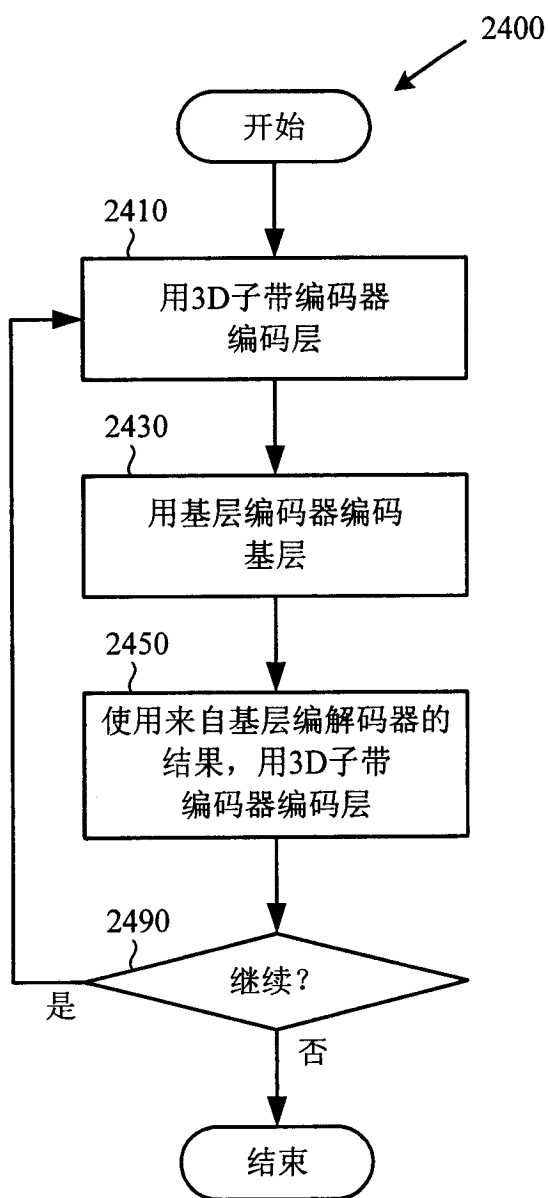


图 24

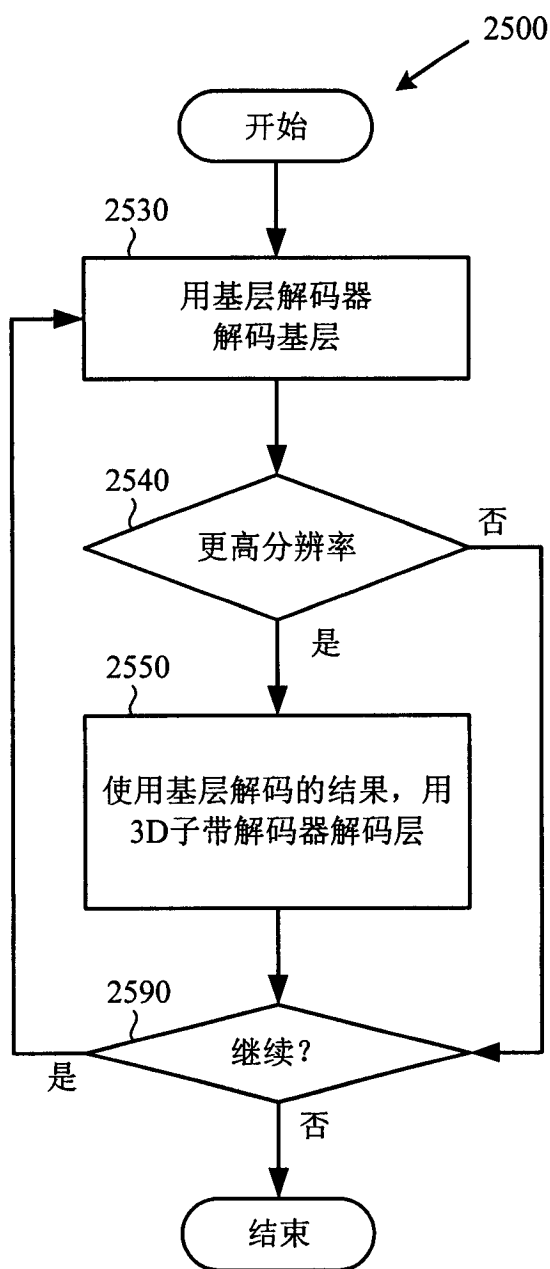


图 25

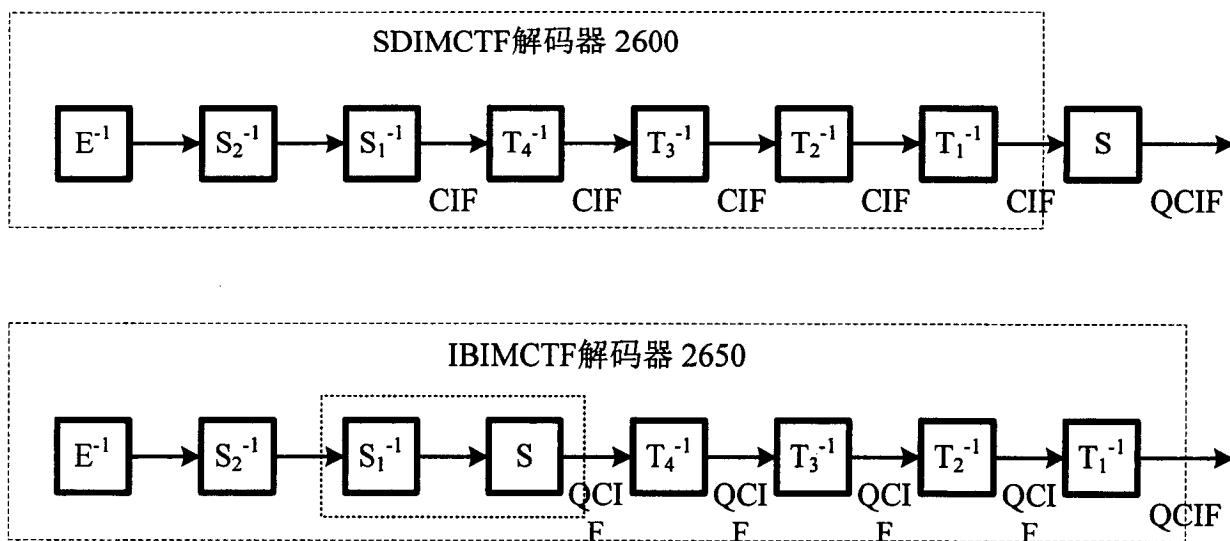


图 26

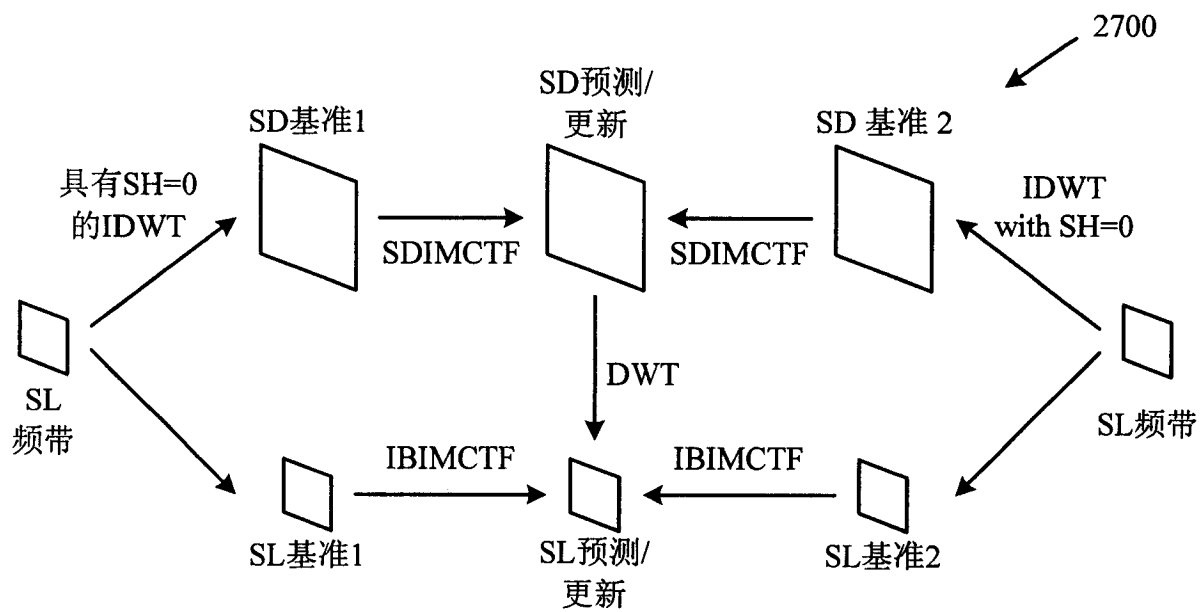


图 27



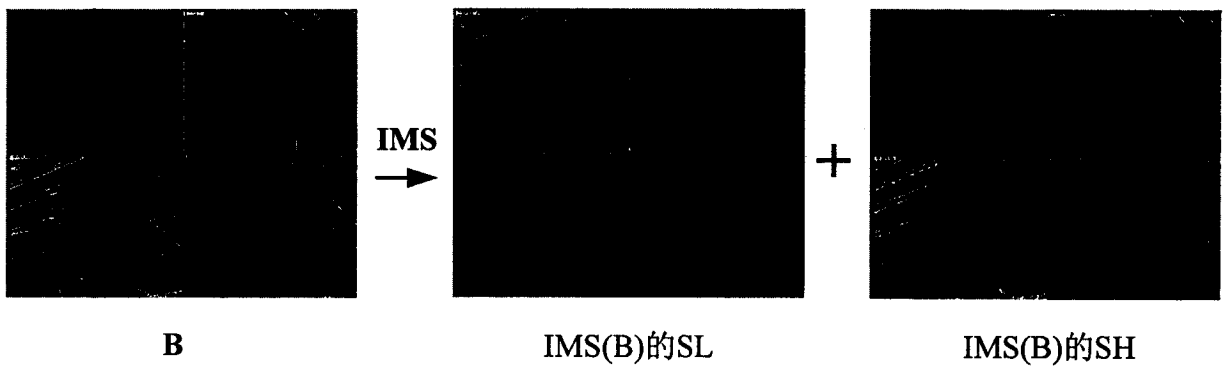
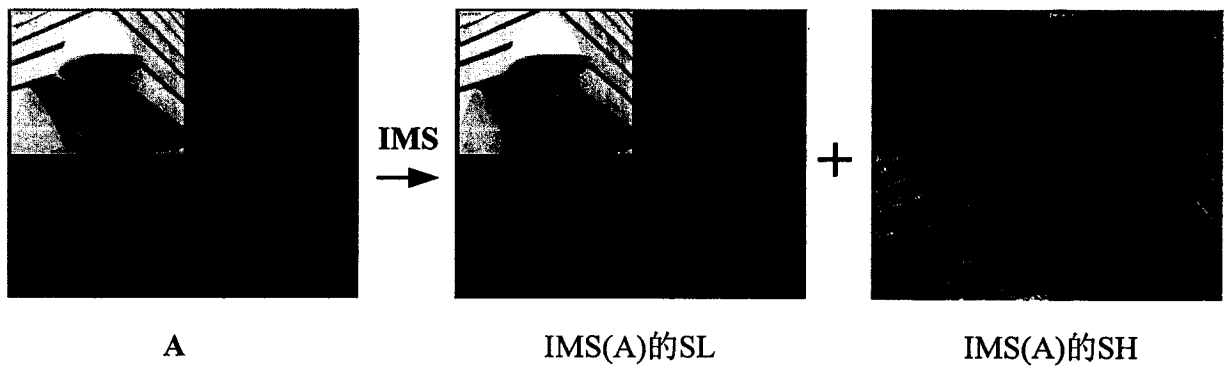
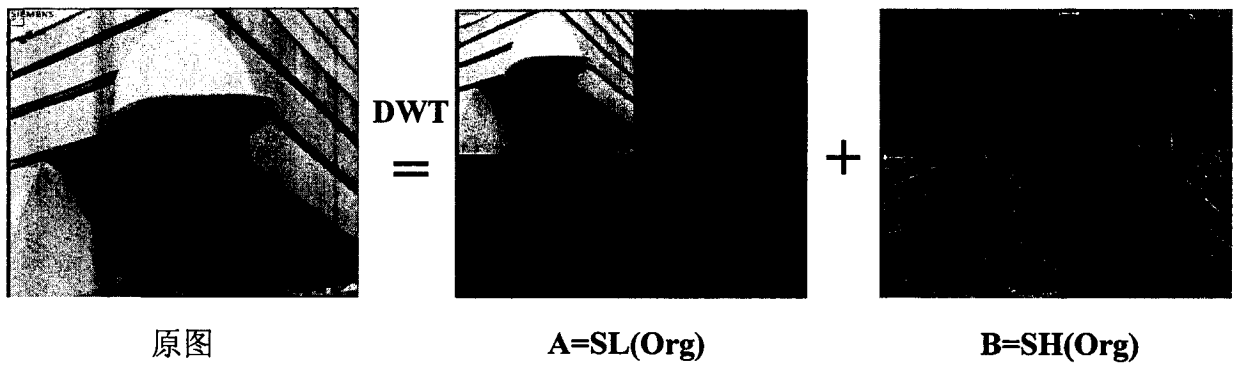


图 28

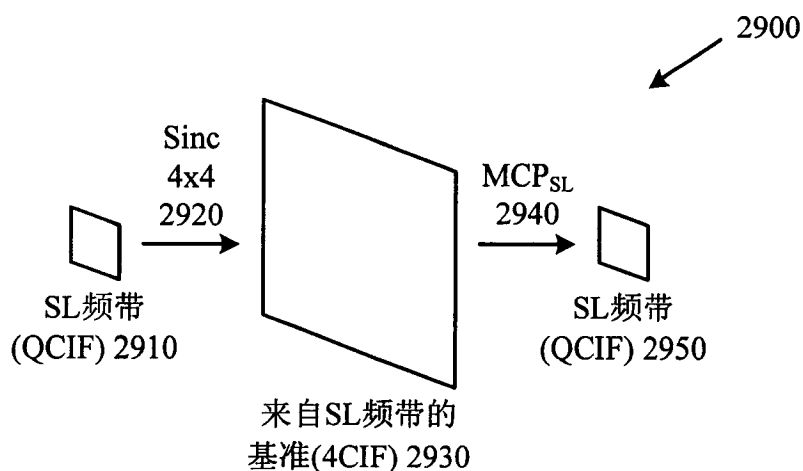


图 29

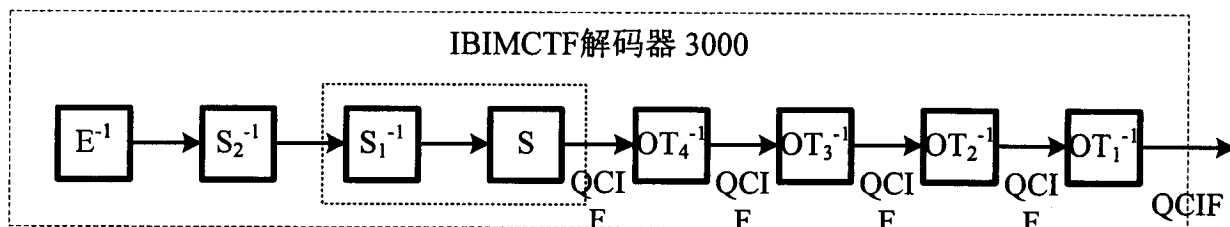


图 30

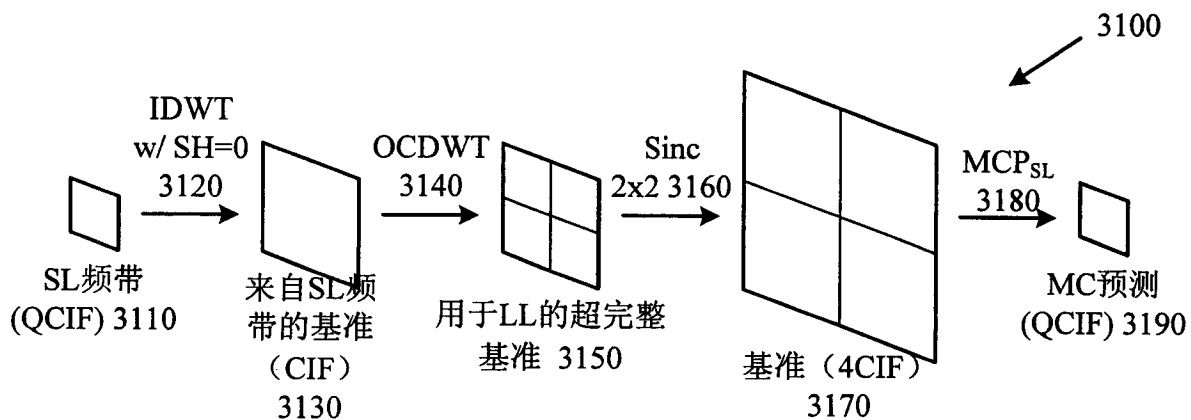


图 31

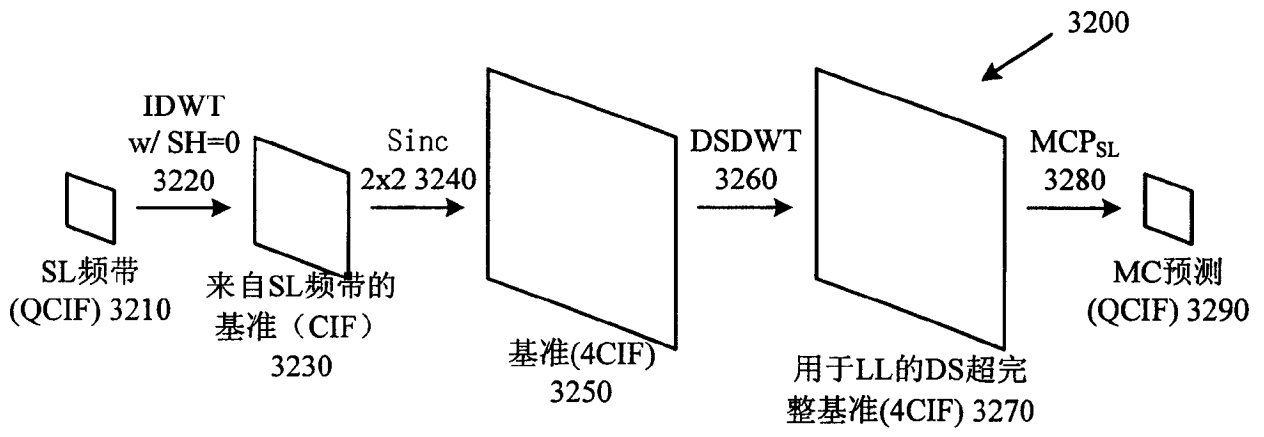


图 32

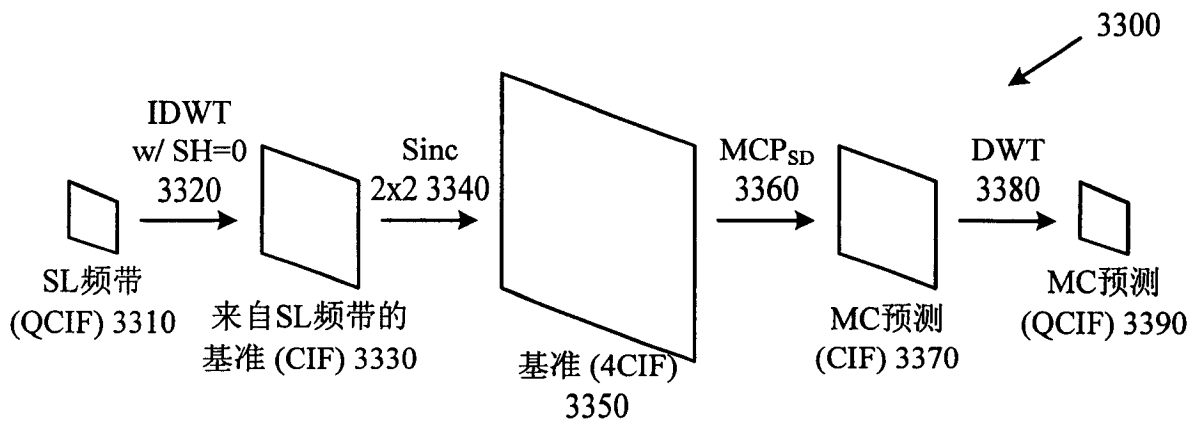


图 33

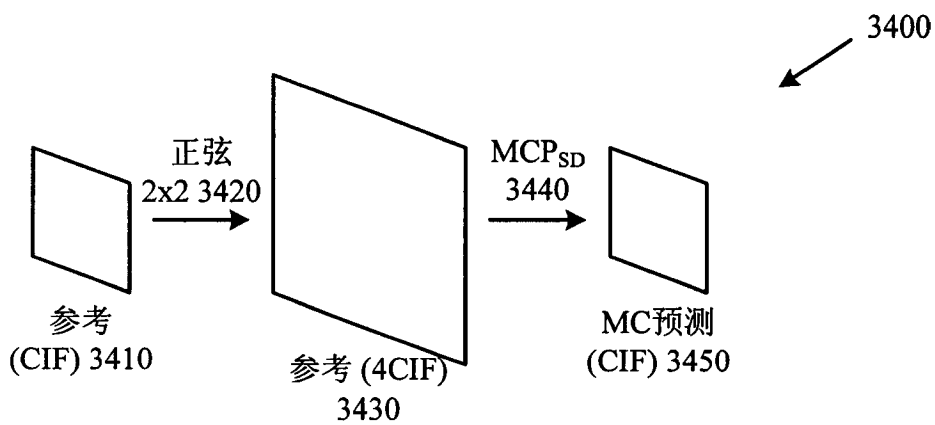


图 34