



(12) 发明专利

(10) 授权公告号 CN 109155859 B

(45) 授权公告日 2021.09.21

(21) 申请号 201780029657.2

(22) 申请日 2017.05.05

(65) 同一申请的已公布的文献号  
申请公布号 CN 109155859 A

(43) 申请公布日 2019.01.04

(30) 优先权数据  
16305562.7 2016.05.13 EP

(85) PCT国际申请进入国家阶段日  
2018.11.13

(86) PCT国际申请的申请数据  
PCT/EP2017/060766 2017.05.05

(87) PCT国际申请的公布数据  
W02017/194405 EN 2017.11.16

(73) 专利权人 交互数字麦迪逊专利控股公司  
地址 法国巴黎

(72) 发明人 弗兰克·伊龙 爱德华·弗朗索瓦  
克里斯托夫·舍旺斯  
皮埃尔·安德里翁

(74) 专利代理机构 中科专利商标代理有限责任  
公司 11021

代理人 陈慧

(51) Int.Cl.  
H04N 19/70 (2006.01)  
H04N 19/85 (2006.01)

审查员 田亚平

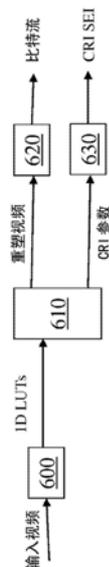
权利要求书2页 说明书13页 附图10页

(54) 发明名称

编码包括亮度分量和两个色度分量的输入视频的方法

(57) 摘要

本原理涉及用于对包括亮度分量和两个色度分量在内的输入视频进行编码的方法和设备。所述方法包括：- 通过将分段线性函数应用(610)于亮度分量和色度分量来重塑输入视频；- 将重塑视频编码(620)为比特流；以及- 将表示分段线性函数的参数编码(630)为辅助信息；并且该方法的特征在于，输入视频的重塑还取决于：解码的重塑视频的目标范围、以及在对解码的重塑视频应用逆重塑之后的重构视频的范围。



1. 一种用于对包括亮度分量和两个色度分量在内的输入视频进行编码的方法,所述方法包括:

通过将分段线性函数应用于亮度分量和色度分量来重塑所述输入视频;

将重塑视频编码为比特流;

编码第一信息数据和第二信息数据作为第一辅助信息,所述第一信息数据确定解码的重塑视频的目标范围,所述第二信息数据确定重构视频的范围;

将表示所述分段线性函数的参数编码为第二辅助信息;

其中,所述输入视频的重塑还取决于解码的重塑视频的目标范围、以及对解码的重塑视频应用逆重塑之后的重构视频的范围。

2. 一种用于根据比特流对重构视频进行解码的方法,包括:

通过对比特流进行解码来获得解码的重塑视频;

根据第一辅助信息获得第一信息数据和第二信息数据,所述第一信息数据确定解码的重塑视频的目标范围,所述第二信息数据确定重构视频的范围;以及

通过对解码的重塑视频进行逆重塑来获得所述重构视频,

其中,对所述解码的重塑视频的所述逆重塑取决于:所述解码的重塑视频的目标范围、以及所述逆重塑应用于所述解码的重塑视频之后的重构视频的范围。

3. 根据权利要求1或2所述的方法,其中,所述方法还包括:通过1D查找表来表示每个分段线性函数。

4. 根据权利要求1或2所述的方法,其中,所述方法还包括:检查所述第一信息数据的状态和所述第二信息数据的状态,以便确定是否达到预期状态;

然后,如果达到第一预期状态,则将所述分段线性函数乘以小于1的缩放因子,并且将非零偏移值添加到所述分段线性函数;

然后,如果达到第二预期状态,则将所述分段线性函数乘以小于1的缩放因子;

然后,如果达到第三预期状态,则将所述分段线性函数乘以等于1的缩放因子。

5. 根据权利要求4所述的方法,其中,使用由HEVC标准版本2指定的颜色重映射信息补充增强信息消息来承载第一辅助信息。

6. 一种用于对包括亮度分量和两个色度分量在内的输入视频进行编码的设备,所述设备包括处理器,所述处理器被配置为:

通过将分段线性函数应用于所述亮度分量以及将两个另外的分段线性函数应用于所述色度分量来重塑所述输入视频;

将重塑视频编码为比特流;

编码第一信息数据和第二信息数据作为第一辅助信息,所述第一信息数据确定解码的重塑视频的目标范围,所述第二信息数据确定重构视频的范围;

将表示所述分段线性函数的参数编码为第二辅助信息;

其中,用于对所述输入视频进行重塑的装置还取决于解码的重塑视频的目标范围、以及对解码的重塑视频应用逆重塑之后的重构视频的范围。

7. 一种用于根据比特流对重构视频进行解码的设备,包括处理器,所述处理器被配置为:用于通过对比特流进行解码来获得解码的重塑视频;用于根据第一辅助信息获得第一信息数据和第二信息数据,所述第一信息数据确定解码的重塑视频的目标范围,所述第二

信息数据确定重构视频的范围;以及用于通过对解码的重塑视频进行逆重塑来获得所述重构视频,其中,对所述解码的重塑视频的逆重塑还取决于:所述解码的重塑视频的目标范围、以及所述逆重塑应用于所述解码的重塑视频之后的重构视频的范围。

8. 根据权利要求6或7所述的设备,其中,所述处理器还被配置为通过1D查找表来表示每个分段线性函数。

9. 根据权利要求6或7所述的设备,其中,所述处理器还被配置为:检查所述第一信息数据的状态和所述第二信息数据的状态,以便确定是否达到预期状态;

然后,如果达到第一预期状态,则将所述分段线性函数乘以小于1的缩放因子,并且将非零偏移值添加到所述分段线性函数;

然后,如果达到第二预期状态,则将所述分段线性函数乘以小于1的缩放因子;

然后,如果达到第三预期状态,则将所述分段线性函数乘以等于1的缩放因子。

10. 根据权利要求9所述的设备,其中,使用由HEVC标准版本2指定的颜色重映射信息补充增强信息消息来承载第一辅助信息。

## 编码包括亮度分量和两个色度分量的输入视频的方法

### 技术领域

[0001] 本原理总体涉及使用由HEVC标准版本2指定的颜色重映射信息(CRI)补充增强信息(SEI)消息的视频编码,以在两个给定颜色体积(colour volume)之间执行颜色体积映射(通常对应于不同的色域和动态范围)。

### 背景技术

[0002] 本部分是旨在向读者介绍可能与下文所述的和/或所要求保护的本原理的各个方面相关的各技术方面。相信该讨论有助于向读者提供背景信息,以使得更好地理解本发明原理的各个方面。因此,应当理解:这些陈述应按这种方式解读,而不是作为对现有技术的承认。

[0003] 在下文中,图片包含特定图片/视频格式的一个或若干样本(像素值)阵列,所述特定图片/视频格式规定了相对于图片(或视频)的像素值的所有信息以及可被显示器和/或任意其他设备用于例如显现或解码图片(或视频)的所有信息。图片包括:第一样本阵列形式的至少一个分量(通常是亮度(luma)分量)、以及可能是至少一个其他样本阵列形式的至少一个其他分量(通常是颜色分量)。或者等价地,相同的信息还可以通过一组颜色样本阵列来表示,例如传统的三原色RGB表示。

[0004] 像素值通过C个值的矢量来表示,其中C是分量的数目。矢量的每个值由多个比特表示,其限定了像素值的最大动态范围。

[0005] 图1示出了根据现有技术对输入视频进行编码/解码的方法(HDR CE2:CE2.a-2、CE2.c、CE2.d和CE2.e-3,Taoran Lu、Fangjun Pu、Peng Yin、Tao,JCTVC-W0084)。该方法描述了使用HEVC实现改进的HDR压缩效率的解决方案(J.Boyce、J.Chen等,“Draft high efficiency video coding(HEVC) version 2,combined format range extensions(RExt),scalability(SHVC),and multi-view(MV-HEVC) extensions,”JCTVC-R1013,第18次会议:日本,札幌,2014年6月30日至7月9日),通过应用输入HDR信号的重塑以将其转换为更容易压缩的重塑视频信号。为此,在比特流中插入元数据(通过在SPS/PPS中添加新的句法元素,如文档“JCTVC-W0092\_TestModel\_specification.docx”中所述,包括在无标题的“Description of the Exploratory Test Model (ETM) for HDR/WCGextension of HEVC”的JCTVC贡献JCTVC-W0092中,作者是K.Minoo、T.Lu、P.Yin、L.Kerofsky、D.Rusanovskyy、E.Francois)。

[0006] 在编码器侧,在步骤100中,模块对输入视频进行重塑以生成重塑视频,并生成与重塑视频相关联的元数据。该重塑可以被认为是在视频信号的编码之前应用的预处理。

[0007] 在步骤110中,重塑视频及其相关联的元数据在比特流F中被编码。

[0008] 在解码侧,在步骤120中,模块对比特流F进行解码并生成解码的重塑视频和解码的元数据。

[0009] 在步骤130中,模块使用解码的元数据对解码的重塑视频进行逆重塑,以生成符合输入视频(除了压缩伪影之外)的重构视频。该逆重塑可以被认为是后处理。

[0010] 图2示意性地示出了图1的方法的编码侧的细节。

[0011] 在步骤100中,预处理基于三个重塑函数 $f_{rk}()$ ,其中 $k=0,1$ 或 $2$ :一个重塑函数针对亮度 $f_{r0}()$ ,基于分段多项式模型(PWP),有8个段;此外,两个重塑函数针对色度 $f_{r1}()$ 和 $f_{r2}()$ ,基于分段线性模型(PWL),具有单个段。实际上,建模的内容是这些函数的逆, $invf_{rk}()$ ,其中 $k=0,1$ 或 $2$ (使得对于任何 $x$ , $invf_{rk}(f_{rk}(x))=x$ ),因为这是在后处理中必须在解码器侧应用的。根据这些建模的逆重塑函数,实际应用于预处理的重塑函数 $g_{rk}()$ 是重塑函数 $invf_{rk}()$ 的逆。逆重塑函数通常以1D查找表(LUT)的形式实现。

[0012] 在步骤110中,模块在比特流中对重塑视频进行编码,并将这些模型的参数(称为重塑参数)作为元数据嵌入PPS中。

[0013] 简言之,在预处理期间,根据输入视频导出亮度重塑参数(亮度PWP参数)和色度重塑参数(色度PWL参数)。然后,这些重塑参数用于对输入视频进行重塑。然后,在比特流中对得到的重塑视频进行编码,然后将重塑参数作为元数据编码在比特流中。

[0014] 图3示意性地示出了图1的方法的解码细节。

[0015] 在步骤120中,通过对比特流进行解码来获得解码的重塑视频和元数据(重塑参数),根据解码的重塑参数导出亮度和色度逆重塑函数或LUT,并且基于这些亮度和色度逆重塑函数或LUT对解码的重塑视频应用逆重塑。

[0016] 逆重塑包括以下特定信号范围管理。

[0017] 为了真正受益于用于表示视频的全范围的码字,在全范围中生成重塑视频(意味着,在10位信号的情况下,使用从0到1023的所有值)。然而,在实际渲染设备中,使用的是有限(或合法)范围(通常从64到960的值)。因此,在图1的方法中,重塑和逆重塑实现特定的信号范围管理,使得有限范围中的输入视频被转换为全范围中的重塑视频,并且全范围解码的重塑视频被转换为有限范围的重构视频。

[0018] 这种特定的信号范围管理应该是规范性的。

[0019] 该重塑机制的缺点是必须修改HEVC编解码器和规范以将其与HEVC组合。

## 发明内容

[0020] 以下呈现了本原理的简单概括以便提供对本原理的一些方面的基本理解。本发明内容不是本原理的扩展概述。并不旨在标识本原理的关键或重要元素。下面的概述只是以简化形式介绍本原理的一些方面,作为稍后提供的更详细描述的前言。

[0021] 本原理旨在通过适用在HEVC中定义的颜色重映射信息(CRI)SEI消息的使用来改进与图1至图3描述的重塑过程的缺点,以便执行等同于上面结合图1至图3所述机制的重塑机制(就编码效率而言)。

[0022] 更准确地说,本原理涉及直接在CRI参数中考虑输入视频(要由实现重塑的逆CRI机制来转换)和输出视频(由实现逆重塑的CRI机制转换得到)两者的范围的管理。

[0023] 使用CRI机制来管理重塑过程的输入和输出视频的范围是CRI的一种不寻常的使用,其避免了对HEVC编码器和解码器的任何修改。

[0024] 根据它们的方面,本原理涉及用于对包括亮度分量和两个色度分量的输入视频进行编码的方法。所述方法包括:

[0025] -通过将分段线性函数应用(610)于亮度和色度分量来重塑所述输入视频;

[0026] -将重塑视频编码为比特流;以及

[0027] -将代表所述分段线性函数的参数编码为辅助信息;

[0028] 并且该方法的特征在于,所述输入视频的重塑还取决于解码的重塑视频的目标范围、以及在对解码的重塑视频应用逆重塑之后的重构视频的范围。

[0029] 根据实施例,该方法还包括:将第一信息数据和第二信息数据编码为辅助信息,第一信息数据确定解码的重塑视频的目标范围,第二信息数据确定重构视频的范围。

[0030] 根据本原理的另一方面,本原理涉及一种用于根据比特流对重构视频进行解码的方法。该方法包括:通过对比特流进行解码来获得解码的重塑视频、以及通过对解码的重塑视频进行逆重塑来获得重构视频。该方法的特征在于,对解码的重塑视频的逆重塑取决于:解码的重塑视频的目标范围、以及在逆重塑应用于解码的重塑视频之后的重构视频的范围。

[0031] 根据实施例,解码方法还包括:根据辅助信息获得第一信息数据和第二信息数据,第一信息数据确定解码的重塑视频的目标范围,第二信息数据确定重构视频的范围。

[0032] 根据实施例,上述方法之一还包括:通过1D查找表来表示每个分段线性函数。

[0033] 根据实施例,上述方法之一还包括:检查第一信息数据的状态和第二信息数据的状态,以便确定是否达到预期状态;

[0034] 然后,如果达到第一预期状态,则将分段线性函数乘以小于1的缩放因子,并且将非零偏移值添加到分段线性函数;

[0035] 然后,如果达到第二预期状态,则将分段线性函数乘以小于1的缩放因子;

[0036] 然后,如果达到第三预期状态,则将分段线性函数乘以等于1的缩放因子。

[0037] 根据上述方法之一的变型,使用由HEVC标准版本2规定的颜色重映射信息补充增强信息消息来承载辅助信息。

[0038] 根据本原理的另一方面,本原理涉及一种用于对包括亮度分量和两个色度分量在内的输入视频进行编码的设备。该设备包括执行如下操作的装置:

[0039] -通过将分段线性函数应用于亮度分量并将两个另外的分段线性函数应用于色度分量来对输入视频进行重塑;

[0040] -将重塑视频编码为比特流;以及

[0041] -将代表分段线性函数的参数编码为辅助信息。

[0042] 该设备的特征在于,用于对输入视频的重塑的装置还取决于解码的重塑视频的目标范围、以及在对解码的重塑视频应用逆重塑之后的重构视频的范围。

[0043] 根据本原理的另一方面,本原理涉及一种用于根据比特流对重构视频进行解码的设备,包括:用于通过对比特流进行解码来获得解码的重塑视频的装置、以及用于通过对解码的重塑视频进行逆重塑来获得重构视频的装置。该设备的特征在于,对解码的重塑视频的逆重塑还取决于:解码的重塑视频的目标范围、以及在逆重塑应用于解码的重塑视频之后的重构视频的范围。

[0044] 根据本原理的另一方面,本原理涉及承载编码的重塑视频的信号。该信号的特征在于,所述信号还承载第一信息数据和第二信息数据,第一信息数据确定解码的重塑视频的目标范围,第二信息数据确定在对解码的重塑视频应用逆重塑之后的重构视频的范围。

[0045] 根据本原理的其他方面,本原理涉及一种包括配置为实施上述方法的处理器在内

的设备、包括程序代码指令以在计算机上执行该程序时执行上述方法的步骤的计算机程序产品、以及承载用于在计算设备上执行程序时执行上述方法的步骤的程序代码的指令的非瞬时存储介质。

[0046] 本原理的特定性质以及本原理的其他目的、优势、特征和用途将根据以下结合附图的示例描述变得清楚明白。

### 附图说明

- [0047] 在附图中，示出了本原理的示例，其示出了：
- [0048] -图1示出了用于根据现有技术对输入视频进行编码/解码的方法；
  - [0049] -图2示意性地示出了图1的方法的编码细节；
  - [0050] -图3示意性地示出了图1的方法的解码细节；
  - [0051] -图4示出了在HEVC中定义的颜色重映射模型；
  - [0052] -图5示出了根据本原理的示例的颜色重映射模型；
  - [0053] -图6示意性地示出了根据本原理的示例的编码细节；
  - [0054] -图7示意性地示出了根据本原理的示例的解码步骤的细节；
  - [0055] -图8示出了根据步骤610的变型的范围管理的示例；
  - [0056] -图9示出了根据本原理的示例的设备的架构的示例；以及
  - [0057] -图10示出了根据本原理的示例的通过通信网络进行通信的两个远程设备；
  - [0058] -图11示出了根据本原理的示例的信号的句法；以及
  - [0059] -图12示出了根据本原理的示例的在HEVC中定义的CRI SEI消息的句法的示例。
- [0060] 用相同的参考标号表示类似或相同的元件。

### 具体实施方式

[0061] 下文将参照附图更完全地对本原理进行描述，其中示出了本原理的实施例。然而，本原理可以以许多备选的形式来体现，并且不应被解释为受限于本文所阐述的示例。因此，尽管本原理允许各种改进和替代形式，但以示例方式在附图中示出其特定示例，并将在下文中对其进行详细描述。然而，应该理解，并不旨在将本原理局限于公开的具体形式，相反地本公开涵盖落入由权利要求限定的本原理的精神和范围内的所有修改、等同物和/或备选。

[0062] 本文所用的术语仅是为了描述特定示例，而不意在限制本原理。如本文中使用的，单数形式“一”、“一个”和“所述”意在还包括复数形式，除非上下文明确地给出相反的指示。还应理解，术语“包括”、“具有”、“含有”和/或“包含”在本说明书中使用时表示存在所陈述的特征、整数、步骤、操作、元件和/或部件，但并不排除存在或添加一个或多个其他特征、整数、步骤、操作、元件、部件和/或其组合。此外，当提及元件“响应于”或“连接到”另一元件时，它可以直接响应于或连接到另一元素，或者可以存在中间元件。相反，当提及元素“直接响应于”或“直接连接到”另一元素时，不存在中间元素。如本文中所使用的，术语“和/或”包括一个或多个相关联列出的项目的任意和所有组合，并且可以简写为“/”。

[0063] 应当理解，虽然术语第一、第二等可以在本文用于描述各种元件，但是这些元件不应该受这些术语的限制。这些术语仅用来将元件彼此区分。例如，在不背离本原理教导的情

况下,第一元件可以被称为第二元件,并且类似地,第二元件也可以被称为第一元件。

[0064] 尽管一些图包括通信路径上的箭头来指示通信的主要方向,将理解通信可以在与所指示的箭头的相反方向上发生。

[0065] 相对于方框图 and 操作流程图来描述一些示例,其中每一个方框表示电路元件、模块或代码的一部分,其包括用于实现指定逻辑功能的一个或多个可执行指令。还应当注意,在其他替代实现方式中,框中标注的功能可以不按照标注的次序进行。例如,连续示出的两个框可能在执行时实质上是同时执行的,或者这些框在某些时候会以相反的顺序执行,这将取决于所涉及到的功能。

[0066] 本文中“根据示例”或“在一个示例中”的提及意味着结合示例所描述的特定特征、结构或特性可以包含在本原理的至少一个实施方式中。根据说明书中各处出现短语“在一个示例中”或“根据示例”不一定都指代相同的示例,也不必是与其他示例互斥地分离的或备选的示例。

[0067] 权利要求中出现的附图标记仅仅是说明性的,并且对权利要求的范围没有限制作用。

[0068] 虽然没有明确描述,但是本示例及其变体可以被实施为任意组合或子组合。

[0069] 本原理对n个图片编码/解码进行了描述,但其扩展到对图片序列(视频)的编码/解码,因为如下所述,序列的每个图像被顺序编码/解码。

[0070] 本原理提出使用CRI参数代替在引入部分中描述的重塑过程的元数据。

[0071] 其中一个优点是HEVC规范不需要改变,同时保留了由引入部分中描述的重塑机制所提供的压缩改进。本公开的要点是在重塑之后或者在逆重塑之后适当地管理输入和输出视频信号的范围。

[0072] 在HEVC中定义的颜色重映射信息(CRI) SEI消息传达用于将图片从一个颜色体积映射到另一颜色体积的信息。CRI SEI消息承载CRI参数,该CRI参数是根据图4所示的颜色重映射模型获得的。

[0073] 这种颜色重映射模型包括三个部分:应用于每个颜色分量(Pre-LUT)的第一分段线性函数,接着是应用于三个结果颜色分量的三乘三矩阵,接着是应用于每个结果颜色分量(Post-LUT)的第二分段线性函数。

[0074] 这些数据集的每一个都是可选的(例如,只有第一1D Pre-LUT可以应用),从而导致仅将一个传递函数应用于输入信号的每个颜色分量。对最多33个点进行编码以指定分段线性函数。

[0075] 以下等式说明了该模型在输入颜色样本的值( $R_{in}, G_{in}, B_{in}$ )中的应用(这也可以应用于其他颜色表示,例如YCbCr):

$$[0076] \quad \begin{bmatrix} R_1 \\ G_1 \\ B_1 \end{bmatrix} = \begin{bmatrix} PreLUT[R_{in}] \\ PreLUT[G_{in}] \\ PreLUT[B_{in}] \end{bmatrix}$$

$$[0077] \quad \begin{bmatrix} R_2 \\ G_2 \\ B_2 \end{bmatrix} = M_{3 \times 3} \begin{bmatrix} R_1 \\ G_1 \\ B_1 \end{bmatrix}$$

$$[0078] \quad \begin{bmatrix} R_{out} \\ G_{out} \\ B_{out} \end{bmatrix} = \begin{bmatrix} PostLUT[R_2] \\ PostLUT[G_2] \\ PostLUT[B_2] \end{bmatrix}$$

[0079] 由于输入和输出比特深度可以不同,因此该模型可以支持标准动态范围 (SDR) 与高动态范围 (HDR) 之间的范围转换和/或不同颜色体积 (通常对应于不同的色域和动态范围) 之间的范围转换。

[0080] 在下文中,如上所述的颜色重映射模型限于三个1D LUT (表示每个分段线性函数),如图5所示。这里不使用 $3 \times 3$ 矩阵和三个后1DLUT,它们是CRI参数的常用附加数据。在图5和图6中,CRI适用于视频信号的R、G和B三个分量。CRI还可以应用于以其他颜色表示 (例如YCbCr) 所表示的颜色样本。这是本发明的推荐。通过禁用 $3 \times 3$ 矩阵和三个后1D LUT,有利地,CRI可以直接应用于4:2:0YCbCr信号,其中色度分量Cb和Cr具有亮度分量的分辨率的一半。如果应用 $3 \times 3$ 矩阵,则需要对色度进行上采样。

[0081] 图6示意性地示出了根据本原理的示例的编码步骤的细节。

[0082] 在步骤600中,模块确定CRI模型的三个1D LUT,表示每个分段线性逆重塑函数:一个1D LUT用于亮度,并且两个1D LUT用于色度,例如单个段。

[0083] 在步骤610中,模块通过将用于亮度的1D LUT的逆应用于输入视频的亮度分量以及将用于色度的两个1D LUT的逆应用于输入视频的色度分量,来获得重塑视频。

[0084] 在步骤620中,模块在比特流中对重塑视频进行编码,并且在步骤630中,模块将CRI参数 (相对于三个1D LUT) 嵌入在CRI SEI消息中,如HEVC标准中定义的。

[0085] 针对每个1D LUT  $c, c=0, 1$  或  $2$ ,CRI参数由一组点 ( $pre\_lut\_coded\_value[c][i]$ ,  $pre\_lut\_target\_value[c][i]$ ) 组成,其中  $i=0$  到  $pre\_lut\_num\_val\_minus1[c]$ ,如文档 JCTVC-R1013 (J.Boyce、J.Chen等,“Draft high efficiency video coding (HEVC) version 2,combined format range extensions (RExt),scalability (SHVC),and multi-view (MV-HEVC) extensions”的部分D.2.33和D.3.33节中说明,JCTVC-R1013,第18次会议:日本,札幌,2014年6月30日至7月9日)。这些点对分段线性函数  $invf_{Rc}(x)$  (用于分量  $c$  的逆重塑函数) 建模。例如,针对  $pre\_lut\_coded\_value[c][i]$  和  $pre\_lut\_coded\_value[c][i+1]$  中的  $x$ ,函数  $invf_{Rc}(x)$  对应于连接点 ( $pre\_lut\_coded\_value[c][i]$ ,  $pre\_lut\_target\_value[c][i]$ ) 和 ( $pre\_lut\_coded\_value[c][i+1]$ ,  $pre\_lut\_target\_value[c][i+1]$ ) 的段。实现  $invf_{Rc}(x)$  的LUT  $LUT_{Rc}$  通常如下创建:

[0086]  $LUT_{Rc}[x] = Round(inv f_{Rc}(x))$ ,其中  $x = [0, 1023]$ ,针对10位信号,其中  $Round(x)$  是给出  $x$  的最接近整数值的函数。

[0087] 图7示意性地示出了根据本原理的示例的解码步骤的细节。

[0088] 在步骤700中,模块通过解码比特流来获得解码的重塑视频。

[0089] 在步骤710中,模块根据接收的CRI SEI消息获得CRI参数。

[0090] 在步骤720中,根据CRI参数获得三个1D LUT:一个1D LUT用于亮度,两个1D LUT用于色度。

[0091] 在步骤730中,模块通过基于这三个1D LUT对解码的重塑视频进行逆重塑来获得重构视频。

[0092] 结合图6和图7描述的编码/解码方案避免了对将重塑机制与HEVC编码/解码过程

组合的HEVC标准的任何修改。

[0093] 根据步骤600的实施例,确定三个1D LUT以便更好地利用码字范围,即以便获得改进的压缩。

[0094] 表示逆重塑函数的三个1D LUT是根据输入视频的图片确定的。

[0095] 可以为输入视频的每个图片确定三个1D LUT,或者备选地,相同的三个LUT可以用于输入视频的若干图片或所有图片。

[0096] 例如,表示用于亮度的重塑函数F(L)的1D LUT根据输入视频的图片确定如下:

[0097] 首先,估计图片的平均线性光亮度 $L_{\text{mean}}$ 。仅考虑高于最小阈值 $L_{\text{min}}$ 的亮度样本。通常, $L_{\text{min}}$ 设定为0.1尼特。

[0098] 亮度的重塑函数F(L)定义如下:

$$[0099] \quad F(L) = \log(1. + (L/S)^g) / \log(1. + (P/S)^g) \quad (1)$$

[0100] 其中L是图片的线性光输入亮度,P是主要的(mastering)显示峰值亮度,S和g是两个控制参数。通常g是固定的并设置为2.8。

[0101] 控制参数S与图片有关,并且根据 $L_{\text{mean}}$ 导出如下:

$$[0102] \quad S = a * L_{\text{mean}}^2 + b * L_{\text{mean}} + c$$

[0103] 基于各个内容集合,启发式地确定参数a、b和c,其中 $a = -0.0003$ ,  $b = 0.558$ ,  $c = 4.6767$ 。

[0104] 可以执行控制参数S值的时间平滑以避免强时间变化。

[0105] 然后,获得表示亮度的重塑函数F(L)的1D LUT,以便输入视频的每个值与通过将重塑函数F(L)应用于输入视频的所述值所获得的码值匹配。

[0106] 接下来,通过对表示亮度的重塑函数F(L)的所述1D LUT求逆,获得表示亮度的重塑函数F(L)的所述1D LUT的逆。所述逆1D LUT表示逆重塑函数的分段线性模型。例如,该模型有9、17或33个点。分段线性模型的参数是嵌入在CRI SEI消息中的CRI参数。

[0107] 例如,表示色度的重塑函数的1D LUT根据输入视频的图片确定为使用缩放因子和偏移的简单缩放。

[0108] 这是基于这样的事实:通常,内容的真实色域小于用于编码它的容器的色域。例如,预计不久的将来UHD视频将在BT.2020容器中表示,但内容实际上将限于P3颜色。

[0109] 为了补偿编码器侧的BT.709/P3和BT.2020颜色表示之间的颜色体积差,色度重塑函数是单个分段线性模型,如下所示:

$$[0110] \quad y_{\text{Cb}} = S_{\text{cb}} * (x_{\text{cb}} - 0_{\text{cb}})$$

$$[0111] \quad y_{\text{Cr}} = S_{\text{cr}} * (x_{\text{cr}} - 0_{\text{cr}}) \quad (2)$$

[0112] 其中 $x_{\text{cr}}$ 和 $x_{\text{cb}}$ 是输入色度样本, $y_{\text{cb}}$ 和 $y_{\text{cr}}$ 是前向色度重塑器的输出,并且 $S_{\text{cr}}$ 、 $S_{\text{cb}}$ 是缩放因子, $0_{\text{cr}}$ 、 $0_{\text{cb}}$ 通常设置为10b比特深度色度样本的偏移512。

[0113] 例如,缩放因子 $S_{\text{cr}}$ 、 $S_{\text{cb}}$ 和偏移 $0_{\text{cr}}$ 、 $0_{\text{cb}}$ 可以源自输入视频的原生(native)色域的原色(colour primary)和目标色容器的原色的对应关系,如下表所示。

色彩空间	白点		原色					
	$x_W$	$y_W$	$x_R$	$y_R$	$x_G$	$y_G$	$x_B$	$y_B$
[0114] DCI-P3	0.314	0.351	0.680	0.320	0.265	0.690	0.150	0.060
[0115] ITU-R BT.709	0.3127	0.3290	0.64	0.33	0.30	0.60	0.15	0.06
[0115] ITU-R BT.2020	0.3127	0.3290	0.708	0.292	0.170	0.797	0.131	0.046

[0116] 例如,缩放因子 $S_{cr}$ 、 $S_{cb}$ 的计算如下:

$$[0117] \quad S_{cb} = \sqrt{\left(\frac{tD_B}{nD_B}\right)^2 + \left(\frac{tD_G}{2 * nD_G}\right)^2}$$

$$[0118] \quad S_{cr} = \sqrt{\left(\frac{tD_R}{nD_R}\right)^2 + \left(\frac{tD_G}{nD_G}\right)^2}$$

[0119] 其中 $tD_R$ 、 $tD_G$ 和 $tD_B$ 是根据原色 $(x_R, x_G, x_B)$ 和白点 $(x_W, y_W)$ 定义的目标色域得出的变量:

$$[0120] \quad tD_R = \sqrt{(x_R - x_W)^2 + (y_R - y_W)^2}$$

$$[0121] \quad tD_G = \sqrt{(x_G - x_W)^2 + (y_G - y_W)^2}$$

$$[0122] \quad tD_B = \sqrt{(x_B - x_W)^2 + (y_B - y_W)^2}$$

[0123]  $nD_R$ 、 $nD_G$ 和 $nD_B$ 是根据原色 $(x_R, x_G, x_B)$ 和白点 $(x_W, y_W)$ 定义的原色域得出的变量:

$$[0124] \quad nD_R = \sqrt{(x_R - x_W)^2 + (y_R - y_W)^2}$$

$$[0125] \quad nD_G = \sqrt{(x_G - x_W)^2 + (y_G - y_W)^2}$$

$$[0126] \quad nD_B = \sqrt{(x_B - x_W)^2 + (y_B - y_W)^2}$$

[0127] 根据步骤610的变型,模块根据称为VFR的第一信息数据和称为CRFR的第二信息数据获得重塑视频,第一信息数据确定解码的重塑视频(步骤700的输出)的目标范围,第二信息数据确定重构视频(步骤730的输出)的范围。

[0128] 根据编码/解码方案的这种变型,在解码侧不需要像现有技术那样明确的范围管理(步骤130,图3),因为在计算三个1D LUT(步骤610)期间考虑范围管理。

[0129] 图8示出了根据步骤610的变型的范围管理的示例。

[0130] 根据该示例,第一信息数据和第二信息数据可以具有状态“真”或“假”。

[0131] 首先,模块检查第一信息VFR的状态和第二CRFR的状态,当比特流中存在(或意图存在)时。

[0132] 然后,如果第一信息VFR的状态等于假并且第二信息数据CRFR的状态等于假,则根据以下应用10比特信号的等式(3)和(4)(其中偏移等于64)确定三个1D LUT:

[0133] 针对亮度:

$$[0134] \quad LUT_{R0}[i] = Clip\left(\frac{876}{1023} * invf_{R0}(i) + offset + 0.5, 0, 1023\right) \quad (3)$$

[0135] 针对色度,其中k=1或2:

$$[0136] \quad LUT_{Rk}[i] = Clip\left(\frac{896}{1023} * (invf_{Rk}(i) - 512) + 512 + 0.5, 0, 1023\right) \quad (4)$$

[0137] 如果第一信息VFR的状态等于真并且第二信息数据CRFR的状态等于假,则根据等式(3)和等式(4)(其中偏移等于0)确定三个1D LUT。

[0138] 针对第一信息状态和第二信息状态的其他组合,不应用缩放或偏移。

[0139] 针对亮度:

$$[0140] \quad LUT_{R0}[i] = Clip(invf_{R0}(i) + 0.5, 0, 1023) \quad (5)$$

[0141] 针对色度,其中k=1或2:

$$[0142] \quad LUT_{Rk}[i] = Clip(invf_{Rk}(i) + 0.5, 0, 1023) \quad (6)$$

[0143] 例如,色度分量的LUT<sub>Rk</sub>对由等式1和等式2实现的重塑函数的逆进行建模。

[0144] 根据如图12所示的实施例,第一信息数据是HEVC信令的句法元素。

[0145] 优选地,句法元素是VUI(视频可用性信息)的标志“video\_full\_range\_flag”,其指示解码的重塑视频的范围。根据HEVC标准版本2,句法元素video\_full\_range\_flag指示根据E'Y、E'PB和E'PR或E'R、E'G和E'B实值分量信号导出的亮度和色度信号的黑电平和范围。当video\_full\_range\_flag句法元素不存在时,推断video\_full\_range\_flag的值等于0。

[0146] 注意,不必在比特流中设置VUI参数,此外,当VUI在比特流中时,不必设置“video\_full\_range\_flag”。

[0147] 根据该示例,第二信息数据是HEVC信令的句法元素。

[0148] 优选地,所述句法元素是VUI的“colour\_remap\_full\_range\_flag”,其指示重构视频的范围。根据HEVC标准版本2,当使用CRI机制时,在比特流中存在句法元素“colour\_remap\_full\_range\_flag”以指示颜色重映射视频的范围。该句法元素(colour\_remap\_full\_range\_flag)可以嵌入在CRI SEI消息中。

[0149] 语法元素“colour\_remap\_full\_range\_flag”具有与条款E.3.1中针对video\_full\_range\_flag句法元素指定的语义相同的语义,除了colour\_remap\_full\_range\_flag指定重构视频的图片的色彩空间,而不是用于CLVS的色彩空间。CLVS代表编码的分层视频序列,并且对应于图片和相关数据的序列,与所描述的视频序列(例如图片的大小、视频的帧速率、图像样本的比特深度)所需的图片无关。

[0150] 当不存在时,推断“colour\_remap\_full\_range\_flag”句法元素的值等于“video\_full\_range\_flag”句法元素的值。

[0151] 例如,可以在全范围中生成重塑视频以及解码的重塑视频,而出于应用目的,重构视频以及输入视频在有限范围中。

[0152] 要编码的信号由码字形成,通常是从10比特信号的0到1023。这些码字来自所谓的“传递函数”的应用,其能够将线性光信号转换成适合于编码器和解码器接口的码字值(或

者从码字反向到线性光信号)。由于重塑信号基于HEVC规范的VUI传递函数(在文档JCTVC-R1013的E.2.1节中)中未指定的自适应传递函数,因此建议将VUI句法元素“transfer\_characteristics”设置为“未指定”。在CRI消息中,colour\_remap\_transfer\_function句法元素应设置为输入的视频信号的传递函数的索引,如文档JCTVC-R1013的表E.4中所指定的。例如,如果输入信号是用ST2084传递函数表示的HDR视频,在文档JCTVC-R1013的表E.4中ST2084传递函数的索引是16,则colour\_remap\_transfer\_function语法元素应该被设置为16。

[0153] 解码器(步骤700、710)被配置为解码已由编码器编码(分别为步骤620、630)的数据。

[0154] 编码器(和解码器)不限于能够对视频和元数据编码/解码的特定编码器,例如HEVC编码器(解码器)。

[0155] 用于生成在预处理中应用的重塑函数的步骤总结如下:

[0156] -重塑函数 $f_{Rc}$ 的推导,其中 $c=0,1,2$

[0157] -逆重塑函数 $invf_{Rc}$ 的推导,其中 $c=0,1,2$

[0158] -通过PWP模型(PreLUTc)用适当的范围管理来对逆重塑函数 $invf_{Rc}$ 的建模,其中 $c=0,1,2$

[0159] -根据PWP模型(PreLUTc)推导出的逆函数 $g_{Rc}$ 的推导,其中 $c=0,1,2$

[0160] -使用函数 $g_{Rc}$ 的重塑,其中 $c=0,1,2$

[0161] -对PreLUTc的PWP模型的编码,其中 $c=0,1,2$

[0162] 在后处理中,步骤总结如下:

[0163] -对PWP模型的解码,并生成PreLUTc,其中 $c=0,1,2$

[0164] -通过应用PreLUTc进行逆重塑,其中 $c=0,1,2$

[0165] 在图1至图8上,模块是功能单元,所述功能单元可以或不可以与可区分的物理单元相关。例如,这些模块或其中的一些模块被一起引入到单个组件或电路中,或者对软件的功能有贡献。相反地,一些模块可能由分离的物理实体组成。与本原理相兼容的装置是使用纯硬件(比如,使用诸如《Application Specific Integrated Circuit》、《Field-Programmable Gate Array》、《Very Large Scale Integration》中分别记载的ASIC或FPGA或VLSI之类的专用硬件)来实现的,或者使用来自设备中所嵌入的若干集成电子组件或来自硬件组件与软件组件的混合物的组件来实现。

[0166] 图9表示设备90的示范性体系结构,所述设备可以配置为实现参考图1至图8描述的方法。

[0167] 设备90包括通过数据和地址总线91连接在一起的以下元件:

[0168] -微处理器92(或CPU),其是例如DSP(或数字信号处理器);

[0169] -ROM(或只读存储器)93;

[0170] -RAM(或随机存取存储器)94;

[0171] -I/O接口95,用于从应用接收要发送的数据;以及

[0172] -电池96。

[0173] 根据示例,电池96在设备外部。在所提及的每个存储器中,说明书中使用的词语“寄存器”可以对应于小容量的区域(一些比特)或非常大的区域(例如整个程序或大量的接

收或解码的数据)。ROM 93至少包括程序和参数。ROM 93可以存储算法和指令来执行根据本原理的技术。当接通时,CPU 92将程序上传到RAM中并执行对应的指令。

[0174] RAM 94在寄存器中包括由CPU 92执行并在设备90接通之后上传的程序、寄存器中的输入数据、寄存器中的方法的不同状态的中间数据以及用于执行寄存器中的方法的其它变量。

[0175] 可以例如用方法或过程、装置、软件程序、数据流或信号来实现本文所描述的实施方式。虽然仅在单一形式的实现的上下文中进行讨论(例如,仅作为方法或设备进行讨论),但是所讨论的特征的实现还可以以其他形式(如程序)来实现。装置可以以例如适当的硬件、软件和固件来实现。所述方法可以被实现于诸如处理器的装置中,所述处理器一般地是指处理设备,包括例如计算机、微处理器、集成电路或可编程逻辑器件。处理器还包括通信设备(例如计算机、蜂窝电话、便携/个人数字助理(“PDA”))以及促进终端用户之间的信息通信的其他设备。

[0176] 根据编码或编码器的示例,输入视频是从源获得的。例如,源属于包括以下各项的集合:

[0177] -本地存储器(93或94),例如,视频存储器或RAM(或随机存取存储器)、闪存、ROM(或只读存储器)、硬盘;

[0178] -存储接口(95),例如,具有大容量存储器、RAM、闪存、ROM、光盘或磁性支撑体的接口;

[0179] -通信接口(95),例如有线接口(例如,总线接口、广域网接口、局域网接口)或无线接口(例如,IEEE 802.11接口或Bluetooth®接口);以及

[0180] -图片捕捉电路(例如,传感器,如CCD(电荷耦合器件)或CMOS(互补金属氧化物半导体))。

[0181] 根据解码或解码器的示例,解码图片 $\hat{I}$ 被发送到目的地;具体地,目的地属于包括以下各项的集合:

[0182] -本地存储器(93或94),例如视频存储器或RAM、闪存、硬盘;

[0183] -存储接口(95),例如,具有大容量存储器、RAM、闪存、ROM、光盘或磁性支撑体的接口;

[0184] -通信接口(95),例如有线接口(例如,总线接口(例如,USB(或通用串行总线))、广域网接口、局域网接口、HDMI(高清晰度多媒体接口)接口)或无线接口(例如,IEEE 802.11接口、WiFi®或Bluetooth®接口);以及

[0185] -显示器。

[0186] 根据编码或编码器的示例,比特流和CRI SEI消息被发送到目的地。作为示例,比特流和CRI SEI消息之一存储在本地或远程存储器(例如,视频存储器(94)或RAM(94)、硬盘(93))中。在变型中,将一个或者两者比特流发送到存储接口(95),例如,与大容量存储器、闪存、ROM、光盘或磁性支撑体的接口,和/或经过通信接口(95)(例如,到点对点链路、通信总线、点对多点链路或广播网络的接口)进行传输。

[0187] 根据解码或解码器的示例,从源获得比特流和CRI SEI消息。示例性地,从本地存储器(例如视频存储器(94)、RAM(94)、ROM(93)、闪存(93)或硬盘(93))读取比特流。在变型

中,从存储接口(95)(例如与大容量存储器、RAM、ROM、闪存、光盘或磁性支撑体的接口)接收比特流,和/或从通信接口(95)(例如,到点对点链路、总线、点对多点链路或广播网络的接口)接收比特流。

[0188] 根据示例,被配置为实现关于图6描述的编码方法的设备90属于包括以下各项的集合:

[0189] -移动设备;

[0190] -通信设备;

[0191] -游戏设备;

[0192] -平板电脑(或平板计算机);

[0193] -膝上型笔记本电脑;

[0194] -静止图片相机;

[0195] -视频相机;

[0196] -编码芯片;

[0197] -静态图片服务器;以及

[0198] -视频服务器(例如,广播服务器、视频点播服务器或web服务器);

[0199] 根据示例,被配置为实现关于图7描述的解码方法的设备90属于包括以下各项的集合:

[0200] -移动设备;

[0201] -通信设备;

[0202] -游戏设备;

[0203] -机顶盒;

[0204] -电视机;

[0205] -平板电脑(或平板计算机);

[0206] -膝上型笔记本电脑;

[0207] -显示器;以及

[0208] -解码芯片。

[0209] 根据图10所示的本原理的示例,在两个远程设备A和设备B之间通过通信网络NET的传输环境中,设备A包括与被配置为实现如关于图6所述的用于对图片编码的方法的存储器RAM和ROM相关的处理器,并且设备B包括与被配置为实现如关于图7所述的用于解码的方法的存储器RAM和ROM相关的处理器。

[0210] 根据示例,网络是广播网络,适于将来自设备A的静止图片或视频图像广播到包括设备B的解码设备。

[0211] 意图由设备A发送的信号承载比特流和/或CRI SEI消息。比特流包括如前所述的编码的重塑视频。该信号还可以包括CRI参数,表示意图应用于解码的重塑视频的逆重塑。

[0212] 图11示出了当通过基于分组的传输协议发送数据时这种信号的语法的示例。每个发送的分组P包括报头H和有效载荷PAYLOAD。例如,有效载荷PAYLOAD的至少一个比特专用于表示CRI参数,该CRI参数表示意图应用于解码的重塑视频的逆重塑。

[0213] 这里所述的多种处理和特征的实现方案可以实现为多种不同设备或应用。这种设备的示例包括编码器、解码器、处理来自解码器的输出的后处理器、向编码器提供输入的预

处理器、视频编码器、视频解码器、视频编解码器、web服务器、机顶盒、膝上型笔记本电脑、个人电脑、蜂窝电话、PDA以及处理图片或视频的任意其他设备或其他通信设备。应当清楚，设备可以是移动的，甚至安装在移动交通工具中。

[0214] 此外，所述方法可以通过由处理器执行的指令来实施，并且可以将这些指令（和/或由这种实施产生的数据值）存储在计算机可读存储介质中。计算机可读存储介质可以采用计算机可读程序产品的形式，所述计算机可读程序产品体现在一个或多个计算机可读介质中，并且其中体现有可由计算机执行的计算机可读程序代码。如本文使用的计算机可读存储介质被认为是非暂时性存储介质，提供有用于在其中存储信息的固有能力以及用于从中获取信息的固有能力。计算机可读存储介质可以是例如但不限于：电、磁、光、电磁、红外或半导体系统、装置或设备或前述任意合适组合。应当认识到，以下内容尽管提供了可以应用本发明原理的更特定的计算机可读存储介质的示例，但如本领域普通技术人员更容易认识到的，仅是说明性而非穷尽地列出为：便携式计算机磁盘；硬盘；只读存储器（ROM）；可擦除可编程只读存储器（EPROM或闪存）；便携式紧凑盘只读存储器（CD-ROM）；光存储设备；磁存储设备；或以上的任意合适组合。

[0215] 指令可以形成在处理器可读介质上有形实施的应用程序。

[0216] 指令可以位于例如硬件、固件、软件或其组合中。可以在例如操作系统、单独应用或这两者的组合中发现指令。因此，处理器可以被表征为例如配置用于执行处理的设备和包括具有用于执行处理的指令的处理器可读介质的设备（诸如，存储设备）。此外，处理器可读介质可以存储实现所产生的数据值，作为对指令的添加或替代。

[0217] 本领域技术人员应清楚，实施方式可以产生被格式化为承载例如所存储的或所传输的信息的多种信号。信息可以包括例如用于执行方法的指令、或通过描述的实施方式之一所产生的数据。例如，信号可以被格式化为将用于写入或读取本原理所描述示例的语法的规则作为数据进行承载，或将由本原理所描述示例写入的真实语法值作为数据进行承载。这种信号可被格式化为例如电磁波（例如使用频谱的射频部分）或基带信号。格式化可以包括例如对数据流进行编码或者使用已编码数据流对载波进行调制。信号承载的信息可以是例如模拟或数字信息。可以通过公知的多种不同的有线或无线链路来传输该信号。信号可以存储在处理器可读介质上。

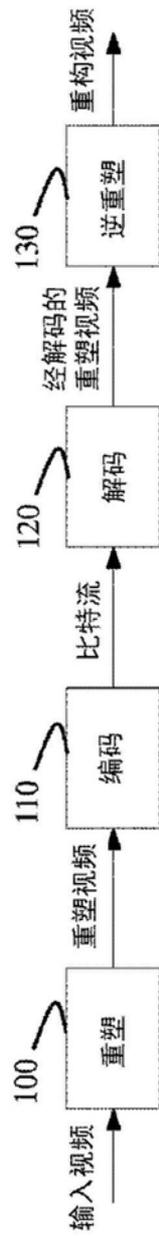


图1

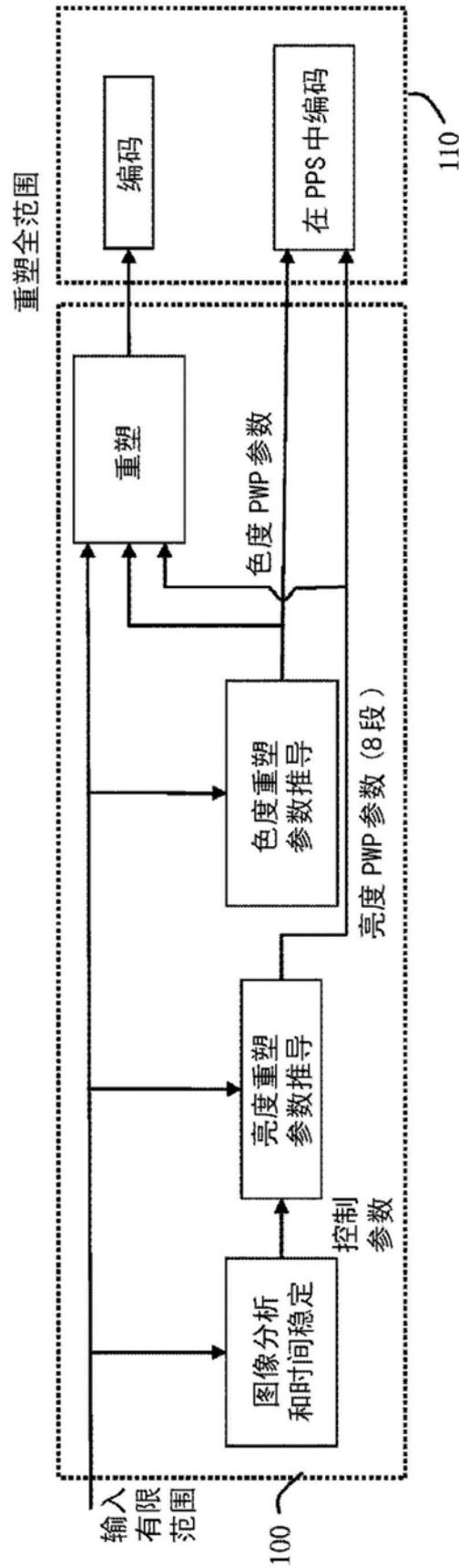


图2

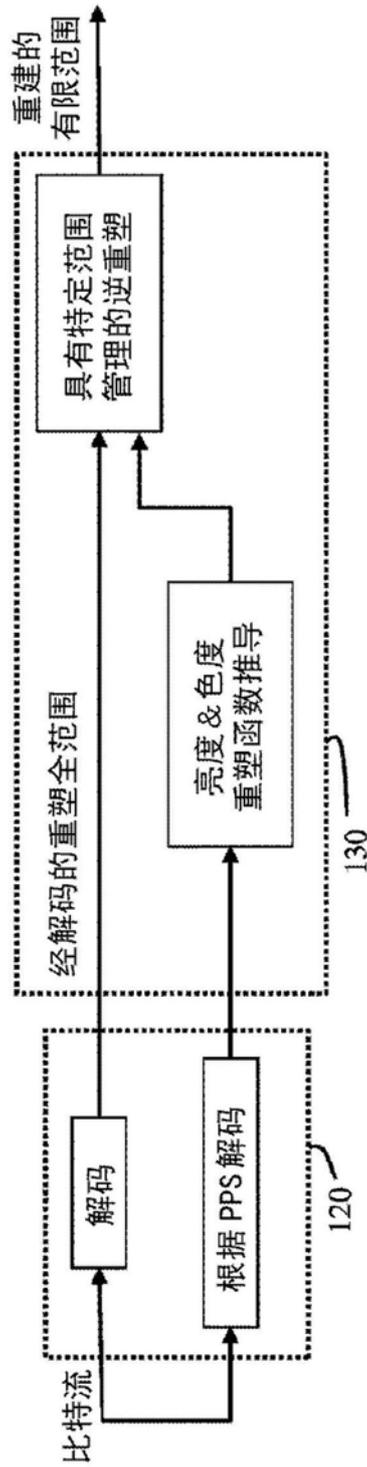


图3

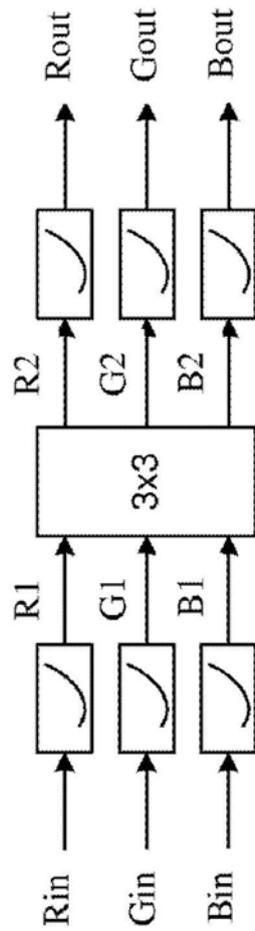


图4

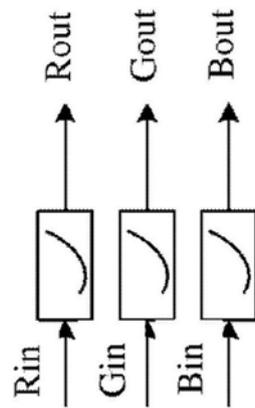


图5

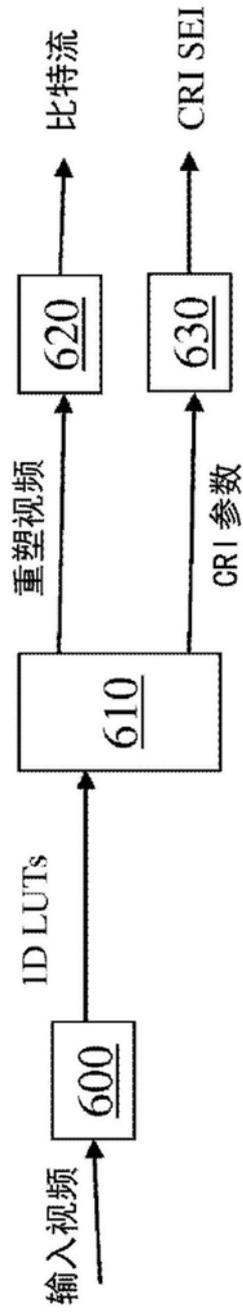


图6

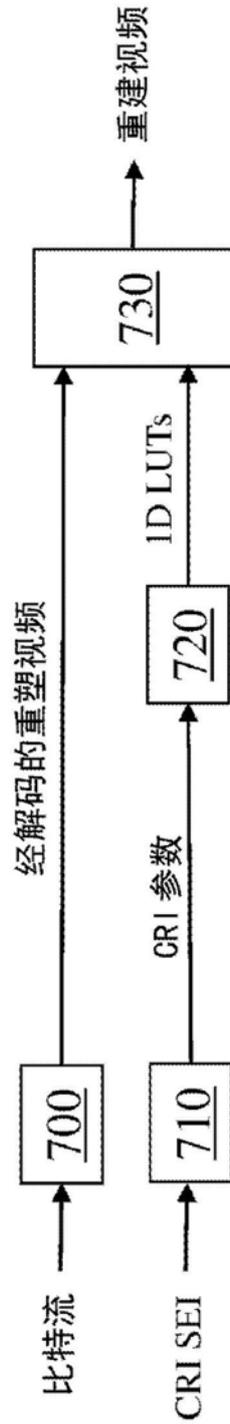


图7

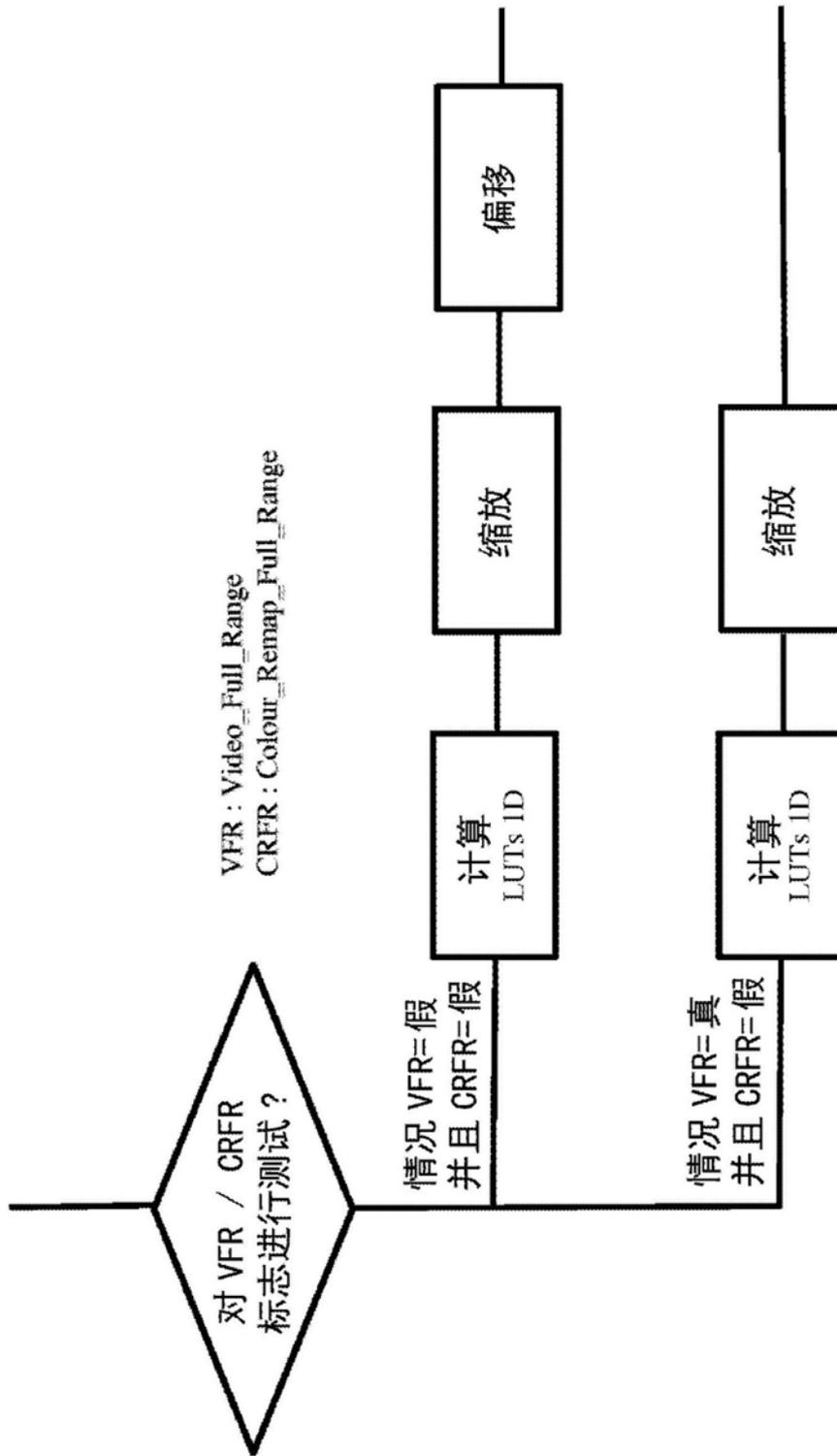


图8

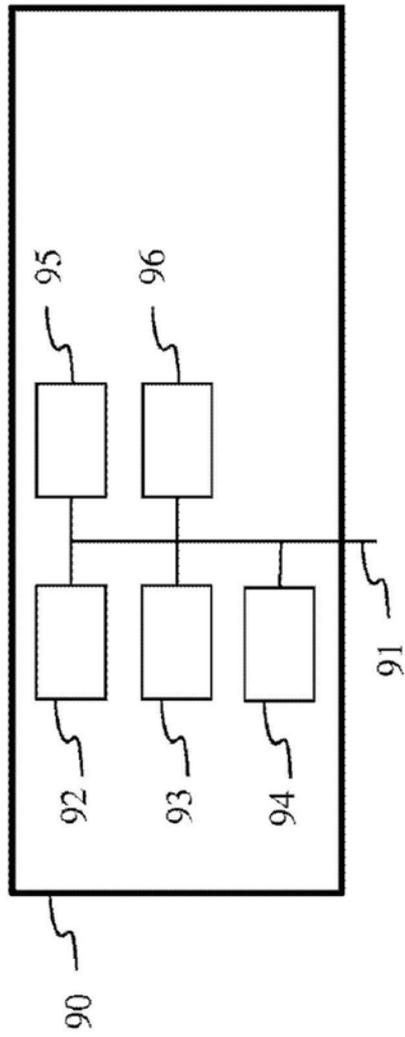


图9



图10



图11

```

colour_remapping_info(payloadSize) {
    colour_remap_id
    colour_remap_cancel_flag
    if(!colour_remap_cancel_flag) {
        colour_remap_persistence_flag
        colour_remap_video_signal_info_present_flag
        if(colour_remap_video_signal_info_present_flag) {
            colour_remap_full_range_flag
            colour_remap_primaries
            colour_remap_transfer_function
            colour_remap_matrix_coefficients
        }
        colour_remap_input_bit_depth
        colour_remap_bit_depth
        for( c = 0; c < 3; c++ ) (
            pre_lut_num_val_minus1[ c ]
            if( pre_lut_num_val_minus1[ c ] > 0 )
                for( j = 0; j <= pre_lut_num_val_minus1[ c ]; j++ ) {
                    pre_lut_coded_value[ c ][ j ]
                    pre_lut_target_value[ c ][ j ]
                }
        )
        colour_remap_matrix_present_flag
        if( colour_remap_matrix_present_flag ) {
            log2_matrix_denom
            for( c = 0; c < 3; c++ )
                for( i = 0; i < 3; i++ )
                    colour_remap_coeffs[ c ][ i ]
        }
        for( c = 0; c < 3; c++ ) {
            post_lut_num_val_minus1[ c ]
            if( post_lut_num_val_minus1[ c ] > 0 )
                for( i = 0; i <= post_lut_num_val_minus1[ c ]; i++ ) {
                    post_lut_coded_value[ c ][ i ]
                    post_lut_target_value[ c ][ i ]
                }
        }
    }
}
    
```

Descriptor  
 ue(v)  
 ur(1)  
 u(1)  
 ur(1)  
 u(1)  
 ur(8)  
 ur(8)  
 ur(8)  
 u(8)  
 u(8)

每个分量的分段线性模型  
 + 视频率范围管理  
 (1D LUT)

3x3 矩阵  
 未使用 (根据实施例)  
 Colour\_remap\_matrix\_present\_flag = 0

每个分量的分段线性模型 (1D LUT)  
 未使用 (根据实施例)  
 Pos\_lut\_num\_val\_minus1[ c ] = 0

图12