



(21) 申请号 201910205555.2

(22) 申请日 2014.03.17

(65) 同一申请的已公布的文献号  
申请公布号 CN 109951714 A

(43) 申请公布日 2019.06.28

(30) 优先权数据

13305453.6 2013.04.08 EP

13306010.3 2013.07.15 EP

14305109.2 2014.01.27 EP

(62) 分案原申请数据

201480026132.X 2014.03.17

(73) 专利权人 杜比国际公司

地址 荷兰阿姆斯特丹

(72) 发明人 P.博兹 P.安德里冯 E.乔利

(74) 专利代理机构 中国贸促会专利商标事务所  
有限公司 11038

专利代理师 宋岩

(51) Int.Cl.

H04N 19/597 (2014.01)

H04N 19/30 (2014.01)

H04N 19/96 (2014.01)

H04N 19/463 (2014.01)

H04N 19/186 (2014.01)

H04N 19/44 (2014.01)

H04N 19/50 (2014.01)

(56) 对比文件

US 2009051624 A1, 2009.02.26

US 6483518 B1, 2002.11.19

US 5953506 A, 1999.09.14

CN 1432969 A, 2003.07.30

US 2003214502 A1, 2003.11.20

US 2010086226 A1, 2010.04.08

CN 102823248 A, 2012.12.12

CN 1681330 A, 2005.10.12

CN 101415120 A, 2009.04.22

US 2007247647 A1, 2007.10.25

审查员 黎媛

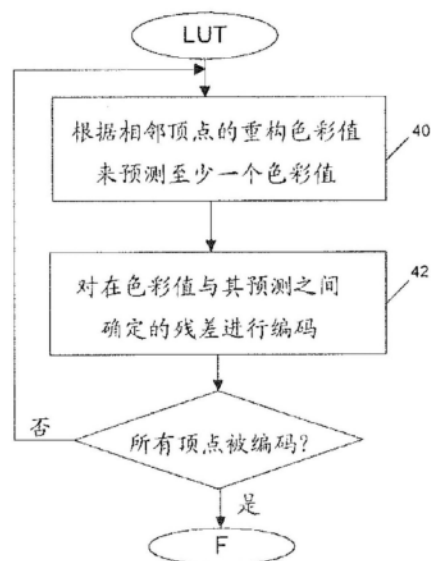
权利要求书3页 说明书19页 附图9页

(54) 发明名称

对LUT进行编码的方法和进行解码的方法以及对应的设备

(57) 摘要

公开了对LUT进行编码的方法和进行解码的方法以及对应的设备。公开一种用于对被定义为顶点的点阵的LUT进行编码的方法。至少一个值与点阵的每个顶点相关联。该方法包含,对于当前顶点:根据例如从与相邻顶点相关联的重构值获得的另外的值来预测与所述当前顶点相关联的至少一个值;以及在比特流中对当前顶点的至少一个值与其在比特流中的预测之间计算出的至少一个残差进行编码。



1. 一种对与八叉树的八分区相关联的3维色彩查找表进行编码的编码方法,包含:

针对当前八分区的、指示是否对其至少一个直接孩子八分区进行递归编码的拆分标志进行编码;

响应于所述拆分标志对所述当前八分区进行编码,其中,对所述当前八分区进行编码包含:根据相邻顶点的至少一个重构值来预测所述当前八分区中的顶点的至少一个值以获得预测;

确定在顶点的至少一个值与所述预测之间计算出的残差;以及

对残差进行熵编码;

其中,该编码方法还包括:

针对所述当前八分区中的顶点,对指示是否针对该顶点对至少一个残差进行编码的标志进行编码。

2. 根据权利要求1所述的编码方法,其中,所述顶点的至少一个值以及所述相邻顶点的至少一个重构值是色彩值。

3. 根据权利要求2所述的编码方法,其中,所述色彩值表示色彩变换。

4. 根据权利要求1至3中的任一项所述的编码方法,其中,所述相邻顶点的至少一个重构值属于当前八分区的父八分区。

5. 一种对八叉树的八分区的3维色彩查找表进行解码的解码方法,包含:

针对当前八分区的、指示是否对其至少一个直接孩子八分区进行递归解码的拆分标志进行解码;以及

响应于所述拆分标志来重构所述当前八分区,其中,重构所述当前八分区包含:

(a) 根据相邻顶点的至少一个重构值来预测所述当前八分区中的顶点的至少一个值以获得预测;

(b) 对接收的数据进行熵解码以获得残差;以及

(c) 根据所述预测以及残差来重构顶点的所述至少一个值;

其中,该解码方法还包括:

针对所述当前八分区中的顶点,对指示是否针对该顶点对至少一个残差进行解码的标志进行解码。

6. 根据权利要求5所述的解码方法,其中,所述顶点的至少一个值以及所述相邻顶点的至少一个重构值是色彩值。

7. 根据权利要求6所述的解码方法,其中,所述色彩值表示色彩变换。

8. 根据权利要求5至7中的任一项所述的解码方法,其中,所述相邻顶点的至少一个重构值属于当前八分区的父八分区。

9. 一种传送装置,包含:

传送指示是否对当前八分区的至少一个直接孩子八分区进行递归编码的拆分标志的部件;

传送针对所述当前八分区、响应于所述拆分标志而编码的数据从而提供表示八分区的3维色彩查找表的比特流的部件,

其中,响应于所述拆分标志而编码的数据表示在所述当前八分区的顶点的至少一个值与根据相邻顶点的至少一个重构值获得的预测之间计算出的熵编码的残差,并且

其中,该传送装置还包含:

传送针对所述当前八分区中的顶点、指示是否针对该顶点对至少一个残差进行编码的标志的部件。

10.一种对与八叉树的八分区相关联的3维色彩查找表进行编码的编码装置,包含:

针对当前八分区的、指示是否对其至少一个直接孩子八分区进行递归编码的拆分标志进行编码的部件;

响应于所述拆分标志对所述当前八分区进行编码的部件,其中,对所述当前八分区进行编码的部件被配置为:根据相邻顶点的至少一个重构值来预测所述当前八分区中的顶点的至少一个值以获得预测;

确定在顶点的至少一个值与所述预测之间计算出的残差的部件;以及

对残差进行熵编码的部件;

其中,该编码装置还包括:

针对所述当前八分区中的顶点、对指示是否针对该顶点对至少一个残差进行编码的标志进行编码的部件。

11.一种对八叉树的八分区的3维色彩查找表进行解码的解码装置,包含:

针对当前八分区的、指示是否对其至少一个直接孩子八分区进行递归解码的拆分标志进行解码的部件;以及

响应于所述拆分标志来重构所述当前八分区的部件,其中,重构所述当前八分区的部件被配置为:

(a) 根据相邻顶点的至少一个重构值来预测所述当前八分区中的顶点的至少一个值以获得预测;

(b) 对接收的数据进行熵解码以获得残差;以及

(c) 根据所述预测以及残差来重构顶点的所述至少一个值;

其中,该解码装置还包括:

针对所述当前八分区中的顶点、对指示是否针对该顶点对至少一个残差进行解码的标志进行解码的部件。

12.一种计算机可读存储介质,其上存储有计算机程序指令,所述计算机程序指令在被处理器执行时使得处理器执行根据权利要求1-4中的任一项所述的编码方法或者根据权利要求5-8中的任一项所述的解码方法。

13.一种用于对与八叉树的八分区相关联的3维色彩查找表进行编码的编码装置,该编码装置包括:

存储器,被配置成存储程序指令,及

耦接到存储器的处理器,被配置成执行程序指令,

其中程序指令在被处理器执行时使得处理器执行根据权利要求1-4中任一项所述的编码方法。

14.一种用于对八叉树的八分区的3维色彩查找表进行解码的解码装置,该解码装置包括:

存储器,被配置成存储程序指令,及

耦接到存储器的处理器,被配置成执行程序指令,

其中程序指令在被处理器执行时使得处理器执行根据权利要求5-8中任一项所述的解码方法。

## 对LUT进行编码的方法和进行解码的方法以及对应的设备

[0001] 本申请是申请号为201480026132.X,申请日为2014年3月17日,题为“对LUT进行编码的方法和进行解码的方法以及对应的设备”的中国发明专利申请的分案申请。

### 技术领域

[0002] 公开一种对被定义为顶点的点阵的查找表进行编码的方法,其中至少一个值与点阵的每个顶点相关联。示例性地,值为色彩值。公开一种对应的解码方法、编码设备和解码设备。

### 背景技术

[0003] 如图1所示,可缩放的视频解码在于对基础层(BL)比特流和至少一个增强层(EL)比特流进行解码(相应地,编码)。通常,根据(可能经向上采样的)解码的BL画面来预测EL画面。然而,当EL画面和BL画面用不同的色彩空间来表示和/或被进行了不同的色彩评级,则预测不太高效。为了改进预测,已知对解码的BL画面应用色彩变换。更准确地,色彩变换使用色彩信息将BL色彩空间(第一色彩空间)的色彩映射为EL色彩空间(第二色彩空间)的色彩。

[0004] 如图2所示,在视频内容分发中,通常对解码画面应用色彩变换,使得经变换的解码画面适合于终端设备呈现能力。

[0005] 这种色彩变换也被称为色彩映射函数(CMF)。CMF例如通过加上偏移的 $3 \times 3$ 增益矩阵(增益偏移模型)来近似。在该情况下,CMF由12个参数来定义。然而,这样的CMF的近似不是非常精确,因为其假设线性变换模型。因此,使用3D查找表(也被称为3D LUT)来描述这样的CMF,而不用在CMF模型上的任何先验。该3D LUT精确得多,因为其大小可以根据所需要的准确度来增加。然而,该3D LUT可能因此表示巨大的数据集。因此,将3D LUT传送给接收器需要对LUT进行编码。

[0006] 近似CMF的LUT将第二色彩空间中的另外的色彩值与第一色彩空间中的至少一个色彩值相关联。LUT允许将第一色彩空间划分成由LUT的顶点限定的一组区域。示例性地,3D LUT将一组色彩值与第一色彩空间中的色彩值的三元组相关联。该组色彩值可以是第二色彩空间中的色彩值的三元组或者表示被用于将第一色彩空间中的色彩值变换为第二色彩空间中的色彩值的色彩变换(例如,局部定义的CMF参数)的一组色彩值。在图3中,方形3D LUT表示为 $N \times N \times N$ 个顶点的点阵。对于3D LUT中的每个顶点 $V(c_1, c_2, c_3)$ ,需要存储色彩值 $(V_{c_1}, V_{c_2}, V_{c_3})$ 的对应的三元组。与3D LUT相关联的数据量为 $N \times N \times N \times K$ ,其中K是被用于存储一个LUT三元组值的比特的量。三元组值例如是(R,G,B)三元组、(Y,U,V)三元组或(Y,Cb,Cr)三元组等。对所有顶点值进行编码是不高效的,因为其表示巨大量的数据。

### 发明内容

[0007] 本发明的目的是克服现有技术中的至少一个不足。公开一种用于对被定义为顶点的点阵的LUT进行编码的方法。至少一个值与点阵的每个顶点相关联。该编码方法包含对当

前顶点：

[0008] -根据另外的值来预测与所述当前顶点相关联的至少一个值；以及

[0009] -在比特流中对当前顶点的至少一个值与其在比特流中的预测之间计算出的至少一个残差进行编码。

[0010] 有利地，所述另外的值根据与相邻顶点相关联的重构值来获得。

[0011] 根据本发明的具体特征，值是色彩值。根据具体实施例，色彩值表示色彩变换。根据本发明的具体特征，色彩值是增益参数或偏移。

[0012] 根据具体实施例，LUT是使用八叉树编码的3D LUT，并且三个值与点阵的每个顶点相关联。在该情况下，用于预测的相邻节点属于当前顶点所属的当前八分区的父八分区。

[0013] 根据本发明的具体特征，根据与相邻顶点相关联的重构值来预测与所述当前顶点相关联的至少一个值包含根据相邻顶点的对应的重构值来内插所述至少一个值。

[0014] 根据具体特征，该方法还包含对比特流中的指示内插的类型的索引进行编码。

[0015] 根据具体实施例，对至少一个残差进行编码包含通过量化器将至少一个残差量化为量化残差以及在比特流中对量化残差进行熵编码，并且其中该方法还包含在比特流中对量化器进行编码。

[0016] 根据本发明的具体特征，针对当前顶点，对指示是否针对顶点对至少一个残差进行编码的标志进行编码。

[0017] 根据变型，针对当前顶点的每个值，对指示是针对该值对残差进行编码还是不对残差进行编码并将其推断为零的标志进行编码。

[0018] 有利地，针对当前八分区对指示是对其直接孩子进行递归编码还是将其所有尚未被编码的孩子的顶点的所有残差推断为零的拆分标志进行编码。

[0019] 还公开一种用于对被定义为顶点的点阵的LUT进行解码的方法。至少一个值与点阵的每个顶点相关联。该解码方法包含对于当前顶点：

[0020] -从比特流中解码出至少一个残差；

[0021] -根据另外的值来预测与当前顶点相关联的至少一个值；以及

[0022] -根据其预测和解码的至少一个残差来重构当前顶点的至少一个值。

[0023] 有利地，所述另外的值根据与相邻顶点相关联的重构值来获得。

[0024] 根据本发明的具体特征，值为色彩值。根据具体实施例，色彩值表示色彩变换。根据本发明的具体特征，色彩值是增益参数或偏移。

[0025] 根据具体实施例，LUT是使用八叉树解码的3D LUT，并且三个值与点阵的每个顶点相关联，相邻顶点属于当前顶点所属的当前八分区的父八分区。

[0026] 根据本发明的具体特征，根据与相邻顶点相关联的重构值来预测与当前顶点相关联的至少一个值包含根据相邻顶点的对应的重构值来内插至少一个值。

[0027] 根据具体特征，该方法还包含从比特流中解码出指示内插的类型的索引。

[0028] 根据具体实施例，解码至少一个残差包含从比特流中熵解码出量化残差并且用量化器将量化残差逆量化成解码的残差，并且其中该方法还包含从比特流中解码出量化器。

[0029] 根据本发明的具体特征，对每个顶点解码出指示是否针对顶点对至少一个残差进行编码的标志。

[0030] 根据变型，针对当前顶点的每个值解码出指示是针对该值解码出残差还是将该残

差推断为零的标志。

[0031] 有利地,针对当前八分区解码出指示是对其直接孩子进行递归解码还是将其所有尚未被解码的孩子的所有的顶点的残差推断为零的拆分标志。

[0032] 一种对被定义为顶点的点阵的LUT进行编码的编码器,其中至少一个值与点阵的每个顶点相关联,包含对于当前顶点:

[0033] -用于根据另外的值来预测与所述当前顶点相关联的至少一个值的部件;以及

[0034] -用于对当前顶点的至少一个值与其在比特流中的预测之间计算出的至少一个残差进行编码的部件。

[0035] 有利地,所述另外的值根据与相邻顶点相关联的重构值来获得。

[0036] 该编码器被配置为执行用于编码的方法的步骤。

[0037] 一种对被定义为顶点的点阵的LUT进行解码的解码器,其中至少一个值与点阵的每个顶点相关联,包含:

[0038] -用于从比特流中解码出与当前顶点相关联的至少一个残差的部件;

[0039] -用于根据另外的值来预测与当前顶点相关联的至少一个值的部件;以及

[0040] -用于根据其预测和解码的至少一个残差来重构当前顶点的至少一个值的部件。

[0041] 有利地,所述另外的值根据与相邻顶点相关联的重构值来获得。

[0042] 该编码器被配置为执行用于解码的方法的步骤。

[0043] 公开一种至少对被定义为顶点的点阵的LUT进行编码的比特流。至少一个值与点阵的每个顶点相关联,并且该比特流包含在其中被编码的、在当前顶点的至少一个值与其预测之间计算出的至少一个残差。

## 附图说明

[0044] 通过下面对本发明的一些实施例的描述,将呈现本发明的其他特征和优点,该描述结合附图进行,附图中:

[0045] -图1示出根据现有技术的使用全范围可缩放性(gamut scalability)的可缩放视频解码器的架构;

[0046] -图2示出根据现有技术的包含色彩变换以适于呈现显示特征的视频解码器的架构;

[0047] -图3概略地示出根据现有技术的被表示为 $N \times N \times N$ 个顶点的点阵的方形3D LUT;

[0048] -图4图示根据本发明的示例性实施例的编码方法的流程图;

[0049] -图5概略地示出2D LUT;

[0050] -图6呈现根据本发明的另一个示例性实施例的编码方法的流程图;

[0051] -图7A和图7B概略地在左边示出将立方体递归地细分成八分区,并且在右边示出对应的八叉树;

[0052] -图8示出根据本发明的顶点的色彩值的内插;

[0053] -图9A概略地示出被表示为不均匀点阵的3D LUT;

[0054] -图9B示出根据具体且非限制性的实施例的与顶点相关联的值的预测;

[0055] -图9C表示8个顶点中的仅4个被编码的八分区;

[0056] -图9D表示共享顶点的八分区;

- [0057] -图10表示根据本发明的示例性实施例的解码方法的流程图；
- [0058] -图11呈现根据本发明的另一个示例性实施例的解码方法的流程图；
- [0059] -图12概略地示出根据本发明的用于对LUT进行编码的编码器；
- [0060] -图13概略地示出根据本发明的用于对LUT进行解码的解码器；以及
- [0061] -图14图示八分区的8个顶点的位置。

## 具体实施方式

[0062] 图4呈现根据本发明的示例性实施例的编码方法的流程图。该方法用于对被定义为顶点的点阵的LUT进行编码,其中至少一个值(例如色彩值)与点阵的每个顶点相关联。在提到解码时,术语“重构”和“解码”往往被用作同义词。下文中,术语色彩值包含诸如RGB、YUV或Y,Cb,Cr值这样的给定色彩空间的色彩值,并且还包含诸如CMF参数(亦即矩阵参数和偏移值)这样的表示色彩变换的值。

[0063] 在步骤40中,当前顶点的至少一个色彩值根据与相邻顶点相关联的重构色彩值来预测。

[0064] 作为示例,在图5中图示与顶点 $V_0(c_{1_0}, c_{2_0})$ 即例如色彩值的值 $(V_{0_{c_1}}, V_{0_{c_2}})$ 的对应相关联的2D LUT。与顶点 $V_0$ 相关联的值例如根据空间相邻的顶点 $V_1(c_{1_1}, c_{2_1})$ 、 $V_2(c_{1_2}, c_{2_2})$ 、 $V_3(c_{1_3}, c_{2_3})$ 和 $V_4(c_{1_4}, c_{2_4})$ 以及值 $(V_{i_{c_1}}, V_{i_{c_2}})$   $i=1, \dots, 4$ 的对应来预测。预测 $P(P_{c_1}, P_{c_2})$ 例如使用如下内插来计算:

$$[0065] \quad P_{c_1} = 0.25 * (V_{1_{c_1}} + V_{2_{c_1}} + V_{3_{c_1}} + V_{4_{c_1}})$$

$$[0066] \quad P_{c_2} = 0.25 * (V_{1_{c_2}} + V_{2_{c_2}} + V_{3_{c_2}} + V_{4_{c_2}})$$

[0067] 根据变型,2D LUT将一组参数 $(m_{11}, m_{12}, m_{21}, m_{22}, o_1, o_2)$ 而不是值 $(V_{0_{c_1}}, V_{0_{c_2}})$ 的对与顶点 $V_0(c_{1_0}, c_{2_0})$ 相关联。该组参数被用于根据 $V_0$ 的值 $(c_{1_0}, c_{2_0})$ 来重构值 $(V_{0_{c_1}}, V_{0_{c_2}})$ 如下:

$$[0068] \quad V_{0_{c_1}} = m_{11} * c_{1_0} + m_{12} * c_{2_0} + o_1$$

$$[0069] \quad V_{0_{c_2}} = m_{21} * c_{1_0} + m_{22} * c_{2_0} + o_2$$

[0070] 与顶点 $V_0$ 相关联的参数例如根据与空间相邻的顶点 $V_1(c_{1_1}, c_{2_1})$ 、 $V_2(c_{1_2}, c_{2_2})$ 、 $V_3(c_{1_3}, c_{2_3})$ 和 $V_4(c_{1_4}, c_{2_4})$ 相关联的重构参数来预测。预测例如使用内插针对当前顶点的参数来计算。

[0071] 在步骤42中,在当前顶点的至少一个色彩值与其预测之间确定至少一个残差,并且还将其编码到比特流F中。通过从当前顶点的至少一个色彩值中减去其预测来确定残差。编码通常包含熵编码。根据变型,编码包含用量化器 $q$ 对残差进行量化,并且对量化残差进行熵编码。

[0072] 作为示例,在2D LUT的情况下,针对顶点 $V_0$ 计算了第一和第二残差。第一残差等于 $(V_{0_{c_1}} - P_{c_1})$ 并且第二残差等于 $(V_{0_{c_2}} - P_{c_2})$ 。然后,将残差或量化残差 $(V_{0_{c_1}} - P_{c_1})/q$ 和 $(V_{0_{c_2}} - P_{c_2})/q$ 熵编码到比特流F中。熵编码使用诸如指数Golomb、Huffman、CABAC(“上下文自适应二进制算术编码”的英文首字母缩写)这样的传统的二进制编码技术。

[0073] 步骤40和42迭代以对LUT的另外的顶点进行编码,直至LUT的所有顶点被编码为止。

[0074] 可选地,编码方法包含将量化器值 $q$ 编码到比特流F中。当前顶点的至少一个色彩



值被重构以用于其他顶点的预测。如果未使用量化器,亦即 $q=1$ ,则每个重构的至少一个色彩值等于对应的原始色彩值。否则,通过对残差去量化并且将去量化的残差添加到预测来确定至少一个重构色彩值。

[0075] 根据变型,编码方法还包含将用于预测顶点的至少一个色彩值的内插类型编码到比特流F中。更精确地,从比特流中解码出标识内插类型的索引。作为示例,索引0标识双线性内插,索引1标识最近近邻内插,索引2标识具有2个最近近邻的线性内插。

[0076] 根据变型,并非将LUT的所有顶点都编码到比特流F中。例如,如果顶点的所有残差或所有量化残差的绝对值低于阈值TH,则对于该顶点不对残差进行编码,例如在TH=0或TH=1的情况下。因此,在比特流中,针对每个顶点,在比特流中对指示是针对该顶点对至少一个残差进行编码还是对所有的残差都不编码并且因此而推断为零的二元标志进行编码。

[0077] 根据变型,针对每个顶点的每个色彩值,对指示是针对该色彩值对残差进行编码还是不对残差进行编码并且推断为零的二元标志进行编码。

[0078] 可选地,还将LUT的大小编码到比特流中。

[0079] 图6呈现根据本发明的另一个示例性实施例的编码方法的流程图。该方法用于对使用八叉树被定义为顶点的点阵的3D LUT进行编码,其中诸如色彩值的三元组这样的一组n个色彩值与点阵的每个顶点相关联,其中n是 $\geq 1$ 的整数。八叉树用于划分3D色彩空间,其通过递归地将其细分成八个八分区,如图7A和7B所示。在图7A中,划分是对称的,而在图7B中,划分是不对称的。级别N的八分区具有其级别为N-1的父八分区。八叉树是四叉树的3D模拟。公开用于对属于当前八分区的点阵的当前顶点V进行编码的该编码方法。在下文中, $n=3$ 。

[0080] 在步骤50中,与坐标 $(r, g, b)$ 的当前顶点V相关联的三个色彩值 $(V_r, V_g, V_b)$ 中的每一个根据与相邻顶点亦即属于当前八分区的父八分区的顶点相关联的重构色彩值来预测。使用 $(r, g, b)$ 而不是 $(c_1, c_2, c_3)$ ,以便简化标记。但是本发明不限于 $(R, G, B)$ 色彩空间。其可以应用于 $(Y, U, V)$ 、 $(Y, Cb, Cr)$ 等色彩空间表示。

[0081] 于是,针对每个色彩值,例如使用三线内插,如图8所示出的那样,来确定预测:

$$[0082] \quad \bar{V}_r = K \times \sum_{i=0,1} \sum_{j=0,1} \sum_{k=0,1} s_i(r) \times s_j(g) \times s_k(b) \times \text{LUT}[r_i][g_j][b_k] \cdot r$$

[0083] 其中:具有 $i=0,1$ 、 $j=0,1$ 和 $k=0,1$ 的 $(r_i, g_j, b_k)$ 是3D色彩空间中的父八分区的顶点的坐标;

[0084]  $(r, g, b)$ 是当前顶点的坐标;

[0085]  $\text{LUT}[r_i][g_j][b_k] \cdot r$ 是与顶点 $(r_i, g_j, b_k)$ 相关联的第一色彩值;

$$[0086] \quad K = \frac{1}{(r_1 - r_0) \times (g_1 - g_0) \times (b_1 - b_0)}$$

[0087]  $s_0(t) = t_1 - t$ 并且 $s_1(t) = t - t_0$ ,其中 $t=r, g$ 或 $b$ 。

[0088] 相同的等式用于 $g$ 和 $b$ 。可以使用诸如棱柱、锥体或四面体等其他类型的内插。

[0089] 点阵不必是如图9A所示那样均匀的。

[0090] 在步骤50的变型中,与索引j的当前八分区的坐标 $(r, g, b)$ 的当前顶点V相关联的三个色彩值 $(V_r, V_g, V_b)$ 中的每一个根据与坐标 $(r_p, g_p, b_p)$ 的一个相邻顶点 $V_p$ ,亦即属于索引 $(j-1)$ 的相邻八分区的顶点相关联的至少一个重构色彩值预测。相邻八分区是在编码八分

区列表中与当前八分区共享至少一个顶点和/或在当前八分区之前的八分区。编码八分区列表是指定八分区的编码的次序的列表。在具体实施例中,所有八分区属于同一个级别。这样,针对每个色彩值,对分量 $r$ :  $\bar{V}_r = \text{PredA}_r + \text{PredB}_r$ , 如图9B所示那样地确定预测, 其中 $\text{PredA}_r$ 是取决于3D LUT中的顶点的位置的值, 并且 $\text{PredB}_r$ 是取决于与相邻顶点 $V_p$ 相关联的重构色彩值的值。 $\text{PredA}_r$ 例如等于 $r_p$ 。示例性地,  $\text{PredB}_r = V'_r - \text{PredA}'_r$ , 其中 $V'_r$ 是相邻八分区的顶点 $V_p$ 的重构色彩值, 并且 $\text{PredA}'_r$ 是取决于3D LUT中的顶点 $V_p$ 的位置的值。 $\text{PredA}'_r$ 例如等于 $r_p$ 。对 $g$ 和 $b$ 使用相同的等式。

[0091] 在步骤52中, 针对当前顶点计算三个残差, 对每个色彩分量一个:  $\text{res}_r = (V_r - \bar{V}_r)$ 、 $\text{res}_g = (V_g - \bar{V}_g)$  和  $\text{res}_b = (V_b - \bar{V}_b)$ 。

[0092] 然后, 将残差熵编码到比特流中或者在进行熵编码之前进行量化。熵编码使用诸如指数Golomb、Huffman、CABAC (“上下文自适应二进制算术编码”的英文首字母缩写) 这样的传统的二进制编码技术。

[0093] 级别0的八分区没有父八分区。同样, 编码八分区列表中的第一八分区没有在前面的八分区。因此, 与该八分区的当前顶点 $V$ 相关联的三个色彩值 ( $V_r, V_g, V_b$ ) 中的每一个根据已知的色彩值 (例如值128) 来预测。根据变型, 针对级别0的八分区的不同顶点使用不同的已知值。根据另外的变型, 根据已知的色彩值进行预测还对级别0的八分区之外的其他八分区进行。根据变型, 与该八分区的当前顶点 $V$ 相关联的三个色彩值 ( $V_r, V_g, V_b$ ) 中的每一个根据与属于相同八分区的相邻顶点相关联的已经重构的色彩值来预测。

[0094] 可选地, 该编码方法包含将量化器值 $q$ 编码到比特流中。还重构当前顶点的三个色彩值, 并且可用于其他顶点的预测。如果未使用量化器 ( $q=1$ ), 则每个重构色彩值等于原始色彩值。否则, 每个重构色彩值通过将对应的残差去量化并且将去量化的残差添加到对应的预测来确定。

[0095] 根据变型, 该编码方法还包含将用于预测顶点的至少一个色彩值编码的内插类型编码到比特流中。更精确地, 将标识内插类型的索引编码到比特流中。作为示例, 索引0标识三线性内插, 索引1标识棱柱内插, 索引2标识锥体内插, 并且索引3标识四面体内插。

[0096] 根据变型, 并非将LUT的所有顶点都编码到比特流中。例如, 如果顶点的所有残差或所有量化残差的绝对值低于阈值 $TH$ , 则针对该顶点, 不对残差进行编码, 例如 $TH=0$ 或 $TH=1$ 。因此, 针对每个顶点, 将指示是否针对该顶点对至少一个残差进行编码的二元标志编码到比特流中。根据变型, 针对每个顶点的每个色彩值, 对指示是针对该色彩值对残差进行编码还是不对残差进行编码并将其推断为零的二元标志进行编码。在另外的变型中, 如图9C所示, 仅对特定的顶点进行编码。在该图中, 针对每个八分区, 8个顶点中仅4个被编码。

[0097] 通常将LUT编码到比特流中, 以便用于变换诸如呈现显示色彩适配或色彩全范围可缩放性这样的应用中的视频的画面。可以将LUT与视频一起编码和传送。如果编码方法知道3D LUT的一些部分将不被应用所使用, 则属于3D LUT的该部分的顶点不被编码。同样, 如果编码方法知道3D LUT的一些部分对视频的最终呈现影响很小, 则不对属于3D LUT的该部分的顶点进行编码。

[0098] 递归地应用该方法以便对整个3D LUT进行编码。在该情况下, 对八叉树的所有八分区进行编码。当当前八分区的所有顶点被编码时, 当前八分区的孩子八分区的顶点被编

码。

[0099] 在变型中,3D LUT在被编码之前被预处理。在该情况下,拆分标志与八叉树中的每个八分区相关联,并且最初被设置为假。在预处理步骤期间,针对每个八分区来确定拆分标志。如果当前八分区要被编码的至少一个顶点具有至少一个大于TH的可能经量化的残差,则其父八分区的拆分标志被设置为“真”。因此,级别N的当前八分区的拆分标志指示是对其直接孩子(亦即,级别N+1的孩子)递归地进行编码还是其所有尚未被编码的孩子(级别N+k的孩子,其中k>0)的顶点的所有残差被推断为零。

[0100] 在编码步骤期间,将拆分标志和残差编码到比特流中。当当前八分区的所有顶点被编码时,如果当前八分区拆分标志为真,则当前八分区的孩子八分区的顶点被编码。优选地,对属于两个八分区的顶点仅编码一次。在变型中,对在若干八分区之间共享的顶点编码一次以上。具体地,用不同的值将顶点编码若干次,其中针对其所属的每个八分区用一个值。参照图9D,在八分区(j)和八分区(j-1)之间共享顶点V1。因此,可以对V1编码两次,其中针对八分区(j)用一个值,并且针对八分区(j-1)用另外的值。每一值和另外的值可以不同。同样,可以对在4个八分区之间共享的顶点V2编码4次。

[0101] 3D LUT例如被编码为VPS(“视频参数集”)、SPS(“序列参数集”)、PPS(“画面参数集”)或者如在AVC、HEVC、SVC或SHVC视频编码标准中定义的一个SEI消息(“补充增强信息”)。3D LUT例如被编码为诸如在下面定义的SEI消息。可选地,还将LUT的大小S编码到比特流S中。S是一个方向上的顶点的数量。

[0102] 根据变型,对包含具有1个色彩值的顶点的n个3D LUT进行编码,而不是对包含如上所公开的具有n个相关联的色彩值(例如n=3)的顶点的一个3D LUT进行编码。

[0103] 根据变型,当3D LUT大小为S=2时,对向解码器指示根据解码的3DLUT来计算加上偏移的3x3增益矩阵并将其用作CMF而不是3D LUT的标志进行编码。

[0104] 根据另外的变型,包含具有n个相关联的色彩值的顶点的一个3D LUT被用于对局部定义的色彩变换的参数进行编码。示例性地,3D LUT的每个顶点与表示色彩变换的12个色彩值相关联。代替将3个色彩值( $V_r, V_g, V_b$ )与当前顶点V相关联,表示色彩变换的12个色彩值与顶点V(r,g,b)相关联,其中在i=0、1或2的情况下的12个色彩值[ai,bi,ci,oi]被定义为满足下面的等式:

$$[0105] \quad \begin{pmatrix} V_r \\ V_g \\ V_b \end{pmatrix} = \begin{pmatrix} a_0 & b_0 & c_0 \\ a_1 & b_1 & c_1 \\ a_2 & b_2 & c_2 \end{pmatrix} \begin{pmatrix} r \\ g \\ b \end{pmatrix} + \begin{pmatrix} o_0 \\ o_1 \\ o_2 \end{pmatrix}$$

[0106] 根据变型,在i=0、1或2的情况下的参数[ai,bi,ci]与顶点相关联。

[0107] 实际上,在大小为2(具有8个顶点的一个单一八分区)的3D LUT的情况下,可以选择顶点的三个色彩值,使得3D LUT等同于由下面的等式定义的增益偏移模型:

$$[0108] \quad \begin{pmatrix} y' \\ u' \\ v' \end{pmatrix} = \begin{pmatrix} a_0 & b_0 & c_0 \\ a_1 & b_1 & c_1 \\ a_2 & b_2 & c_2 \end{pmatrix} \begin{pmatrix} y \\ u \\ v \end{pmatrix} + \begin{pmatrix} o_0 \\ o_1 \\ o_2 \end{pmatrix}$$

[0109] 为此,必须将8个顶点A至H的3个色彩值( $y_x, u_x, v_x$ ) $\times$ A,...H设置为下面的值:

$$\begin{aligned}
 [0110] \quad & \left\{ \begin{array}{l} y_A = 255 \cdot a_0 \\ y_B = 255 \cdot a_0 + o_0 \\ y_C = 255 \cdot b_0 + o_0 \\ y_D = 255 \cdot (a_0 + b_0) + o_0 \\ y_E = 255 \cdot c_0 + o_0 \\ y_F = 255 \cdot (a_0 + c_0) + o_0 \\ y_G = 255 \cdot (b_0 + c_0) + o_0 \\ y_H = 255 \cdot (a_0 + b_0 + c_0) + o_0 \end{array} \right. \quad \left\{ \begin{array}{l} u_A = 255 \cdot a_1 \\ u_B = 255 \cdot a_1 + o_1 \\ u_C = 255 \cdot b_1 + o_1 \\ u_D = 255 \cdot (a_1 + b_1) + o_1 \\ u_E = 255 \cdot c_1 + o_1 \\ u_F = 255 \cdot (a_1 + c_1) + o_1 \\ u_G = 255 \cdot (b_1 + c_1) + o_1 \\ u_H = 255 \cdot (a_1 + b_1 + c_1) + o_1 \end{array} \right. \\
 [0111] \quad & \left\{ \begin{array}{l} v_A = 255 \cdot a_2 \\ v_B = 255 \cdot a_2 + o_2 \\ v_C = 255 \cdot b_2 + o_2 \\ v_D = 255 \cdot (a_2 + b_2) + o_2 \\ v_E = 255 \cdot c_2 + o_2 \\ v_F = 255 \cdot (a_2 + c_2) + o_2 \\ v_G = 255 \cdot (b_2 + c_2) + o_2 \\ v_H = 255 \cdot (a_2 + b_2 + c_2) + o_2 \end{array} \right.
 \end{aligned}$$

[0112] K是取决于被用于表示每个色彩值的比特的数量的常量。如果色彩值以8比特来表示,则K=255,如果色彩值以10比特来表示,则K=1023,等等。于是,三线性内插等同于增益偏移模型。

[0113] 图10表示根据本发明的示例性实施例的解码方法的流程图。

[0114] 在步骤140中,从比特流F中解码出至少一个残差。解码通常包含熵解码。根据变型,解码包含对量化残差进行熵解码以及用量化器q对量化残差进行逆量化。熵解码使用诸如指数Golomb、Huffman、CABAC(“上下文自适应二进制算术编码”的英文首字母缩写)这样的传统的二进制解码技术。

[0115] 可选地,解码方法包含从比特流F中解码出量化器值q。

[0116] 根据变型,并非将LUT的所有顶点都编码到比特流F中。可选地,针对每个顶点,从比特流中解码出指示是否针对该顶点对至少一个残差进行编码的二元标志。如果没有残差被编码,则针对该顶点,将残差假设为零。

[0117] 在步骤142中,根据与相邻顶点相关联的重构色彩值来预测当前顶点的至少一个色彩值。作为与顶点V0(c1,c2)相关联的2D LUT的示例,在图5中图示色彩值(V0<sub>c1</sub>,V0<sub>c2</sub>)的对应对。根据空间相邻的顶点V1、V2、V3和V4来预测与顶点V0相关联的色彩值。例如,使用内插来计算出预测器P(Pc1,Pc2),如下:

$$[0118] \quad Pc1 = 0.25 * (V1_{c1} + V2_{c1} + V3_{c1} + V4_{c1})$$

$$[0119] \quad Pc2 = 0.25 * (V1_{c2} + V2_{c2} + V3_{c2} + V4_{c2})$$

[0120] 根据变型,2D LUT将一组参数(m11,m12,m21,m22,o1,o2)而不是值(V0<sub>c1</sub>,V0<sub>c2</sub>)的对与顶点V0(c1<sub>0</sub>,c2<sub>0</sub>)相关联。该组参数被用于根据V0的值(c1<sub>0</sub>,c2<sub>0</sub>)来重构值(V0<sub>c1</sub>,V0<sub>c2</sub>),如下:

$$[0121] \quad V0_{c1} = m11 * c1_0 + m12 * c2_0 + o1$$

$$[0122] \quad V0_{c2} = m21 * c1_0 + m22 * c2_0 + o2$$

[0123] 例如根据与空间相邻的顶点V1(c1<sub>1</sub>,c2<sub>1</sub>)、V2(c1<sub>2</sub>,c2<sub>2</sub>)、V3(c1<sub>3</sub>,c2<sub>3</sub>)和V4(c1<sub>4</sub>,c2<sub>4</sub>)相关联的重构参数来预测与顶点V0相关联的参数。预测例如针对当前顶点的参数使用内插

来计算。

[0124] 在步骤144中,重构顶点。更精确地,当前顶点的至少一个色彩值根据其预测和解码的至少一个残差来重构。

[0125] 例如,在2D LUT的情况下,针对当前顶点V0对两个残差 $R_{c1}$ 和 $R_{c2}$ 进行解码。从而,当前顶点通过如下计算其两个色彩值来重构: $(R_{c1}+P_{c1})$ 和 $(R_{c2}+P_{c2})$ 。

[0126] 根据变型,解码方法还包含从比特流F中解码出用于预测顶点的至少一个色彩值的内插类型。更精确地,从比特流中解码出标识内插类型的索引。例如,索引0标识双线性内插,并且索引1标识最近顶点值内插。

[0127] 可选地,还从比特流中解码出LUT的大小。

[0128] 图11表示根据本发明的另一个示例性实施例的解码方法的流程图。该方法用于对使用八叉树被定义为顶点的点阵的3D LUT进行解码,其中色彩值的三元组与点阵的每个顶点相关联。公开该解码方法用于对属于当前八分区的点阵的当前顶点V进行解码。

[0129] 在步骤150中,从比特流F中解码出三个残差 $res_r$ 、 $res_g$ 、 $res_b$ 。解码通常包含熵解码。根据变型,解码包含对量化残差进行熵解码以及使用量化器q对量化残差进行逆量化。熵解码使用诸如指数Golomb、Huffman、CABAC(“上下文自适应二进制算术编码”的英文首字母缩写)这样的传统的二进制解码技术。

[0130] 可选地,该解码方法包含从比特流F中解码出量化器值q。

[0131] 根据变型,并非LUT的所有顶点都被编码到比特流F中。可选地,针对每个顶点,从比特流中解码出指示是否针对该顶点对至少一个残差进行编码的二元标志。如果没有残差被编码,则针对该顶点,将残差假设为零。在另外的变型中,如图9C所示,仅对特定顶点进行解码。在该图中,针对每个八分区,仅对8个顶点中的4个进行解码。

[0132] 在步骤152中,根据与相邻顶点亦即属于当前八分区的父八分区的顶点相关联的重构色彩值来预测与坐标 $(r, g, b)$ 的当前顶点V相关联的三个色彩值 $(V_r, V_g, V_b)$ 中的每一个。使用 $(r, g, b)$ 而不是 $(c1, c2, c3)$ 以便简化标记。但本发明不限于 $(R, G, B)$ 色彩空间。其可以应用于 $(Y, U, V)$ 、 $(Y, Cb, Cr)$ 等色彩空间表示。

[0133] 因此,针对每个色彩值确定出预测。

[0134] 因此,针对每个色彩值,例如使用如图8所示的三线性内插来确定出预测:

$$[0135] \quad \bar{V}_r = K \times \sum_{i=0,1} \sum_{j=0,1} \sum_{k=0,1} s_i(r) \times s_j(g) \times s_k(b) \times LUT[r_i][g_j][b_k] \cdot r$$

[0136] 其中:具有 $i=0,1$ ,  $j=0,1$ 和 $k=0,1$ 的 $(r_i, g_j, b_k)$ 是3D色彩空间中的父八分区的顶点的坐标;

[0137]  $(r, g, b)$ 是当前顶点的坐标;

[0138]  $LUT[r_i][g_j][b_k] \cdot r$ 是与顶点 $(r_i, g_j, b_k)$ 相关联的第一色彩值;

$$[0139] \quad K = \frac{1}{(r_1 - r_0) \times (g_1 - g_0) \times (b_1 - b_0)}$$

[0140]  $s_0(t) = t_1 - t$ 并且 $s_1(t) = t - t_0$ ,其中 $t=r, g$ 或 $b$ 。

[0141] 对 $g$ 和 $b$ 使用相同的等式。可以使用诸如棱柱、锥体或四面体这样的其他类型的内插。点阵不必如图9A所示那样地均匀。

[0142] 在步骤152的变型中,与索引j的当前八分区的坐标 $(r, g, b)$ 的当前顶点V相关联的

三个色彩值( $V_r, V_g, V_b$ )中的每一个根据与坐标( $r_p, g_p, b_p$ )的相邻顶点 $V_p$ 亦即属于索引( $j-1$ )的相邻八分区的顶点相关联的至少一个重构色彩值来预测。相邻八分区是与当前八分区共享至少一个顶点和/或在编码八分区列表中在当前八分区前面的八分区。编码八分区列表是指定八分区的编码的次序的列表。在具体实施例中,所有八分区属于同一个级别。从而,针对分量 $r$ ,如图9B所示那样,针对每个色彩值来确定预测: $\bar{V}_r = \text{PredA}_r + \text{PredB}_r$ ,其中 $\text{PredA}_r$ 是取决于3D LUT中的顶点的位置的值,并且 $\text{PredB}_r$ 是取决于与相邻顶点 $V_p$ 相关联的重构色彩值的值。 $\text{PredA}_r$ 例如等于 $r_p$ 。示例性地, $\text{PredB}_r = V'_r - \text{PredA}'_r$ ,其中 $V'_r$ 是相邻八分区的顶点 $V_p$ 的重构色彩值,并且 $\text{PredA}'_r$ 是取决于3D LUT中的顶点 $V_p$ 的位置的值。 $\text{PredA}'_r$ 例如等于 $r_p$ 。对 $g$ 和 $b$ 使用相同的等式。

[0143] 在步骤154中,针对当前顶点计算出并因而重构的三个色彩值,其中根据它们的预测和对应的解码的残差( $\text{res}_r, \text{res}_g, \text{res}_b$ )。于是,当前顶点通过如下计算三个色彩值而被重构: $(\text{res}_r + \bar{V}_r), (\text{res}_g + \bar{V}_g)$  并且  $(\text{res}_b + \bar{V}_b)$ 。

[0144] 级别0的八分区没有父八分区。同样,编码八分区列表中的第一八分区没有前面的八分区。因此,与该八分区的当前顶点 $V$ 相关联的三个色彩值( $V_r, V_g, V_b$ )中的每一个根据已知的色彩值(例如值128)来预测。根据变型,与该八分区的当前顶点 $V$ 相关联的三个色彩值( $V_r, V_g, V_b$ )中的每一个根据与属于相同八分区的相邻顶点相关联的已经重构的色彩值来预测。

[0145] 递归地应用该方法,以便对整个3D LUT进行解码。优选地,对属于两个八分区的顶点仅解码一次。在变型中,对在若干八分区之间共享的顶点解码一次以上。具体地,用不同的值对顶点解码若干次,针对其所属的每个八分区使用一个值。参照图9D,在八分区( $j$ )和八分区( $j-1$ )之间共享顶点 $V1$ 。因此,可以对 $V1$ 解码两次,针对八分区( $j$ )用一个值,针对八分区( $j-1$ )使用另外的值。第一个值和另外的值可以不同。同样,在4个八分区之间共享的顶点 $V2$ 可以被解码4次。

[0146] 当当前八分区(级别 $N$ )的所有顶点被解码时,当前八分区的孩子(级别 $N+1$ )八分区的顶点被解码。

[0147] 根据变型,针对级别 $N$ 的当前八分区解码出指示是对其直接孩子(亦即级别 $N+1$ 的孩子)递归地进行解码还是将尚未被解码的所有孩子(亦即级别 $N+k$ 的孩子,其中 $k>0$ )的顶点的所有残差推断为零的分拆标志。

[0148] 根据变型,该解码方法还包含从比特流 $F$ 中解码出用于预测顶点的至少一个色彩值的内插类型。更精确地,从比特流中解码出标识内插类型的索引。例如,索引0标识三线性内插,索引1标识棱柱内插,索引2标识锥体内插,并且索引3标识四面体内插。

[0149] 3D LUT例如从如在AVC、HEVC、SVC或SHVC视频编码标准中所定义的一个SEI消息中或VPS、SPS、PPS中解码出。可选地,还从比特流中解码出LUT的大小。例如在诸如在下面定义的SEI消息中解码出3D LUT。

[0150] 根据变型,对包含具有1个色彩值的顶点的 $n$ 个3D LUT进行解码,而不是对包含具有 $n$ 个色彩值(例如 $n=3$ )的顶点的一个3D LUT进行解码。

[0151] 根据变型,当3D LUT大小为 $S=2$ 时,解码出指示解码器根据解码的3DLUT计算加上偏移的 $3 \times 3$ 增益矩阵并将其用作CMF而不是3D LUT的标志。

[0152] 关于根据图4和6的编码方法公开的相同变型可以应用于解码器侧。

[0153] 在图4、6、10和11中,所表示的框纯粹是功能实体,其未必对应于物理上分开的实体。如本领域中的技术人员将意识到的那样,本原理的方法可以实施为系统、方法或计算机可读介质。因此,本原理的方法可以采用完全硬件实施例、完全软件实施例(包括固件、常驻软件、微代码等)或者结合软件和硬件方面的实施例(在本文中,一般可以将其称为“电路”、“模块”或“系统”)的形式。而且,本原理的方面可以采用计算机可读存储介质的形式。可以利用一个或多个计算机可读存储介质的任何组合。

[0154] 附图中的流程图和/或框图示出根据本发明的各个实施例的系统、方法和计算机程序产品的可能的实现方式的配置、操作和功能。就此而言,流程图或框图中的每个框可以表示代码的模块、片断或部分,其包含用于实现所指定的逻辑功能的一个或多个可执行的指令。应当注意到,在一些替代的实现方式中,在框中标记的功能可以不以在图中标记的次序出现。例如,相继示出的两个框实际上可以基本并行地执行,或者各个框有时可以以相反的次序来执行,或者各个框可以以替代性的次序来执行,取决于所涉及的功能。还要注意的,框图和/或流程图图示的每个框以及框图和/或流程图图示中的框的组合可以由执行特定功能或动作的专用的基于硬件的系统或者利用的硬件和计算机指令的组合来实现。虽然未明确描述,但是所呈现的实施例可以以任何组合或子组合来使用。

[0155] 还公开了对诸如3D LUT这样的LUT进行编码的比特流。由图中的编码方法所产生的比特流对被定义为顶点的点阵的至少一个LUT进行编码,其中至少一个色彩值与点阵的每个顶点相关联,其中所述比特流包含在其中编码的在当前顶点的至少一个色彩值与其预测之间计算出的至少一个残差。

[0156] 在ITU-T SG16WP3和ISO/IEC JTC1/SC29/WG11的视频编码联合协作组(JCT-VC)的文献JCTVC-L1003中定义的HEVC编码标准或者作为在ITU-T SG16WP3和ISO/IEC JTC1/SC29/WG11的视频编码联合协作组(JCT-VC)的文献JCTVC-L1008中定义的HEVC编码标准的可缩放的扩展的SHVC编码标准的框架内提出示例性的实施例。该标准定义编码数据的任何流必须遵守以兼容该标准的语法。该语法具体地定义如何对不同项的信息进行编码(例如包括在序列中的画面有关的数据、运动矢量等)。在SHVC编码标准的背景下,LUT可以被编码为PPS或VPS。语法元素use\_color\_prediction被用于指示当前画面中的色彩预测的使用,如表1所示。

[0157]	<b>use_color_prediction</b>
	if(use_color_prediction)
	<b>3D_LUT_color_data ( )</b>
	rbsp_trailing_bits( )

[0158] 表1:发出预测参数的信号

[0159] 如果use\_color\_prediction标志等于“1”,则调用3D\_LUT\_color\_data函数以发出3D LUT数据的信号,如表2所示。

[0160]

3D_LUT_color_data() {	<b>Descriptor</b>
<b>nbpCode</b>	u(3)
coding_octant(0, 0,0,0)	
}	

[0161] 表2:3D LUT色彩数据的编码

[0162] nbpCode指示针对nbpCode的给定值的如在表4中列出的3D LUT大小。量化器值可以通过3D\_LUT\_color\_data()函数来编码。

[0163] 根据变型,3D\_LUT\_color\_data()在表3中被定义如下。

[0164]

3D_LUT_color_data() {	<b>Descriptor</b>
<b>nbpCode</b>	u(3)
NbitsPerSample	u(5)
coding_octant(0, 0,0,0)	
}	

[0165] 表3:3D LUT色彩数据的编码

[0166] nbpCode指示针对nbpCode的给定值的如在表4中列出的3D LUT大小。量化器值可以通过3D\_LUT\_color\_data()函数来编码。

[0167] NbitsPerSample指示用于表示色彩值的比特的数量。

[0168]

<b>nbpCode</b>	<b>3D LUT 大小</b>
----------------	------------------

[0169]

0	2
1	3
2	5
3	9
4	17
5	33

[0170] 表4:nbpCode的解释

[0171] 对octant(layer,y,u,v)的解码是如表4所示的递归函数。每个八分区包含与指示是对残差色彩值进行编码还是全部推断为零的标志(encoded\_flag[i])相关联的8个顶点。通过将残差添加到色彩值的预测来重构色彩值。例如使用layer\_id-1的8个相邻顶点的三线性内插来计算色彩值的预测。



[0172]	coding_octant ( layer_id, y,u,v) {		<b>Descriptor</b>
	for( i = 0; i < 8 ; i++ ) {		
	<b>encoded_flag[i]</b>		u(1)
	if ( encoded_flag[i] ) {		
	<b>resY[j]</b>		ue(v)
	<b>resU[j]</b>		ue(v)
	<b>resV[j]</b>		ue(v)
	}		
	}		
	<b>split_flag</b>		u(1)
	if ( split_flag ) {		
	for( i = 0; i < 8 ; i++ ) {		
	coding_octant ( layer_id+1, y+dy[i],u+du[i],v+dv[i])		
	}		
	}		

[0173] 表5: coding\_octant() 的语法元素

[0174] 根据另外的有利的实施例, 将LUT编码为SEI消息 (SEI代表“补充增强信息”)。HEVC标准在其附录D中定义了对被称为SEI的附加信息进行编码的方式。该附加信息在语法中通过被称为payloadType的字段来引用。SEI消息例如有助于与显示有关的处理。注意, 如果解码设备不具有对于其使用所必要的功能, 则忽略该信息。根据本发明的具体实施例, 定义一种新的类型的SEI消息, 以便对与3D LUT有关的附加信息进行编码。为此, 从尚未使用的值之中定义字段payloadType的新值 (例如payloadType等于24)。

[0175] SEI数据的语法 (亦即sei\_payload) 以下面的方式扩展:

[0176]	color_mapping_info( payloadSize ) {		<b>Descriptor</b>
	<b>color_map_id</b>		ue(v)
	<b>color_map_cancel_flag</b>		u(1)
	if( !color_map_cancel_flag ) {		
	<b>color_map_repetition_period</b>		ue(v)
	<b>color_description_present_flag</b>		u(1)
	If (colour_description_present_flag) {		
	<b>color_primaries_input_id</b>		u(8)
	<b>color_primaries_output_id</b>		u(8)
	}		
	<b>color_output_rgb</b>		u(1)
	<b>lut_bit_depth_minus8</b>		u(4)
	3D_ LUT_ color_data ( )		
	}		
	}		

[0177] 表6: 色彩映射SEI消息

[0178] 根据变型, SEI消息还包含指示符color\_interpolator\_id, 例如在color\_description\_present\_flag之后, 其值指示如表7中所规定的内插的类型。

[0179]	color_interpolator_id	3D LUT大小
	0	三线形
	1	四面体

[0180] 表7:内插的类型

[0181] 该SEI消息提供使得能够针对具体显示环境的定制来重新映射输出的解码画面的色彩样本。该重新映射过程将RGB色彩空间中的编码的样本值映射为目标样本值。映射被表示在亮度或RGB色彩空间域中,并且应当被应用于亮度分量或者由解码画面的色彩空间转换相应地产生的每个RGB分量。

[0182] 3D\_LUT\_color\_data()在表2或3中定义。

[0183] 解码的3D LUT被应用于属于例如通过NAL单元首部的索引nuh\_layer\_id标识的层的解码画面(参见SEI消息的ITU-T SG16 WP3和ISO/IEC JTC1/SC29/WG11的视频编码联合工作组(JCT-VC)的文献JCTVC-L1003中定义的文献HEVC编码标准的7.3.1.2节)。

[0184] color\_map\_id包含可以被用于标识色彩映射模型的目的的标识号。color\_map\_id的值可以被用作由应用所定义的。color\_map\_id可以被用于支持适合于不同显示场景的色彩映射操作。例如,color\_map\_id的不同的值可以对应于不同的显示位深度。

[0185] color\_map\_cancel\_flag等于1指示色彩映射信息SEI消息取消输出次序中的任何先前色彩映射信息SEI消息的留存(persistence)。color\_map\_cancel\_flag等于0指示色彩映射信息跟随。

[0186] color\_map\_repetition\_period指定色彩映射信息SEI消息的留存并且可以指定在其之中具有相同值的color\_map\_id的另外的色彩映射信息SEI消息或编码视频序列的末尾将存在于比特流中的画面次序计数间隔。color\_map\_repetition\_period等于0指定色彩映射信息仅适用于当前解码画面。

[0187] color\_map\_repetition\_period等1指定色彩映射信息以输出次序留存,直至以下条件中的任何一个为真为止:

[0188] -新的编码视频序列开始;

[0189] -输出具有大于被表示为PicOrderCnt(CurrPic)的当前解码画面的POC的画面次序计数(被称为POC)的包含具有相同值的color\_map\_id的色彩映射信息SEI消息的访问单位中的画面。

[0190] color\_map\_repetition\_period等于0或等于1指示具有相同值的color\_map\_id的另外的色彩映射信息SEI消息可以存在或不可以在。

[0191] color\_map\_repetition\_period大于1指示色彩映射信息留存直至以下条件中的任何一个为真为止:

[0192] -新的编码视频序列开始;

[0193] -输出具有大于PicOrderCnt(CurrPic)并且小于或等于PicOrderCnt(CurrPic)+color\_map\_repetition\_period的POC的包含具有相同值的color\_map\_id的色彩映射信息SEI消息的访问单位中的画面。

[0194] color\_map\_repetition\_period大于1指示对于被输出的具有大于PicOrderCnt(CurrPic)并且小于或等于PicOrderCnt(CurrPic)+color\_map\_repetition\_period的POC的访问单位中的画面,存在具有相同值的color\_map\_id的另外的色彩映射信息SEI消息;除非比特流结束或者新的编码视频序列开始而不输出这样的画面。

[0195] color\_description\_present\_flag等于1指定colour\_primaries\_input\_id和colour\_primaries\_output\_id存在。

[0196] color\_description\_present\_flag等于0指定colour\_primaries\_input\_id和colour\_primaries\_output\_id不存在。

[0197] color\_primaries\_input\_id指示按照由ISO 11664-1所规定的x和y的CIE1931定义的在表8中指定的来源主色的色度坐标。

[0198] color\_primaries\_output\_id指示当应用3D色彩lut时,按照由ISO11664-1所规定的x和y的CIE 1931定义的在表8中指定的经色彩映射的主色的色度坐标。

[0199] color\_output\_rgb等于1指定输出色彩样本为亮度和色度信号。color\_output\_rgb等于0指定输出色彩样本为绿色彩、红色、蓝色值。

[0200] lut\_bit\_depth\_minus8指定3D LUT样本的位深度。

[0201] nbp\_code指示针对给定值的nbp\_code的在表4中所列出的3D LUT大小nbp。

[0202] 3D LUT解码的输出是nbp x nbp x nbp大小的三维阵列LUT。每个LUT阵列元素被称为顶点,并且与等于(lut\_bit\_depth\_minus8+8)的位深度的3个重构样本值(recSamplesY,recSamplesU,recSamplesV)相关联。如果 $i \% (nbp >> layer\_id)$ 、 $j \% (nbp >> layer\_id)$ 、 $k \% (nbp >> layer\_id)$ 的值等于零,则说顶点lut[i][j][k]属于层layer\_id。一个顶点可能属于若干层。层layer\_id的八分区包含属于layer\_id的8个相邻顶点(图14)。

[0203] octant(layer\_id,y,u,v)的解码是递归函数。每个八分区包含与指示是对残差分量值(resY[i],resU[i],resV[i])进行编码还是全部推断为零的标志(encoded\_flag[i])相关联的8个顶点( $i=0, \dots, 7$ )。分量值通过将残差添加到分量值的预测来重构。分量值的预测使用layer\_id-1的8个相邻顶点的三线性内插来计算。一旦被重构,就将顶点标记为重构。

[0204] 在 $(y+dy[i])$ 、 $(u+du[i])$ 和 $(v+dv[i])$ 是当前八分区(具有(y,u,v)作为第一顶点( $i=0$ )坐标)的8个孩子八分区坐标(第一3D色彩顶点的坐标)。针对给定层的值dy[i],du[i]和dv[i]在表9中示出。

[0205]	值	主色	信息备注
	0	保留	由 ITU-T ISO/IEC 将来使用
	1	主色                      x                      y	ITU-R Rec. BT.709-5

[0206]

	绿色	0.300	0.600	ITU-R Rec. BT.1361 传统的色彩全范围系统和扩展的色彩全范围系统
	蓝色	0.150	0.060	
	红色	0.640	0.330	
	白色 D65	0.3127	0.3290	IEC 61966-2-1 ( sRGB 或 sYCC ) IEC 61966-2-4 电影电视工程师协会 RP 177 ( 1993 ) 附录 B
2	未指定			图像特征未知或者由应用来确定
3	保留			由 ITU-T ISO/IEC 将来使用
4	主色	x	y	ITU-R Rec. BT.470-6 系统 M ( 历史 )
	绿色	0.21	0.71	美国国家电视系统委员会 1953 年关于彩色电视的传送标准的建议
	蓝色	0.14	0.08	美国联邦通信委员会标题 47 编码联邦规定 ( 2003 ) 73.682 ( a ) ( 20 )
	红色	0.67	0.33	
	白色 C	0.310	0.316	
5	主色	x	y	ITU-R Rec. BT.470-6txx B,G ( 历史 )
	绿色	0.29	0.60	ITU-R Rec. BT.610-6 625
	蓝色	0.15	0.06	ITU-R Rec. BT.1358 625
	红色	0.64	0.33	ITU-R Rec. BT.1700 625 PAL 和 625SECAM
	白色 D65	0.3127	0.3290	
6	主色	x	y	ITU-R Rec. BT.601-6 525
	绿色	0.310	0.595	ITU-R Rec. BT.1358 525
	蓝色	0.155	0.070	ITU-R Rec. BT.1700 NTSC
	红色	0.630	0.340	电影电视工程师协会 170M ( 2004 )
	白色 D65	0.3127	0.3290	( 在功能上与值 7 相同 )

[0207]

7	主色	x	y	电影电视工程师协会 240M (1999) (在功能上与值 6 相同)
	绿色	0.310	0.595	
	蓝色	0.155	0.070	
	红色	0.630	0.340	
	白色 D65	0.3127	0.3290	
8	主色	x	y	一般电影 (使用照明 C 的色彩 滤波器)
	绿色	0.243	0.692	
	( Wratten 58 )			
	蓝色	0.145	0.049	
	( Wratten 47 )			
9	主色	x	y	Rec. ITU-R BT.2020
	绿色	0.170	0.797	
	蓝色	0.131	0.046	
	红色	0.708	0.292	
	白色 D65	0.3127	0.3290	
10..255	保留			由 ITU-T ISO/IEC 将来使用

[0208] 表8:色彩主色 (color\_primaries\_input\_id和color\_primaries\_output\_id)

[0209]

i	dy[i]	du[i]	dv[i]
0	0	0	0
1	0	0	nbp>>layer_id
2	0	nbp>>layer_id	0
3	0	nbp>>layer_id	nbp>>layer_id
4	nbp>>layer_id	0	0
5	nbp>>layer_id	0	nbp>>layer_id
6	nbp>>layer_id	nbp>>layer_id	0
7	nbp>>layer_id	nbp>>layer_id	nbp>>layer_id

[0210] 表9:属于层=layer\_id的顶点的以索引i的函数的形式的值dy[i]、du[i]和dv[i]

[0211] 属于层=layer\_id的八分区的顶点的((y+dy[i]), (u+du[i]), (v+dv[i]))重构的3D色彩LUT样本(recSamplesY[i]、recSamplesU[i]、recSamplesV[i])由下式给出:

[0212]  $\text{recSamplesY}[i] = \text{resY}[i] + \text{predSamplesY}[i]$ 

[0213] 其中,使用关于包含当前八分区的层=layer\_id-1的八分区的顶点的三线性内插来推出predSampleY[i]的值。

[0214] 图12表示编码器1的示例性架构。编码器被配置为执行编码方法的步骤。编码器1

包含由数据和地址总线64链接到一起的以下元件：

- [0215] -微处理器61 (或CPU) ,其例如是DSP (或数字信号处理器) ;
- [0216] -ROM (或只读存储器) 62;
- [0217] -RAM (或随机存取存储器) 63;
- [0218] -一个或若干I/O (输入/输出) 设备65,诸如例如键盘、鼠标、网络摄像机;以及
- [0219] -电源66。

[0220] 根据变型,电源66在编码器的外部。图12中的这些元件中的每一个是本领域技术人员公知的,并且将不做进一步公开。在每个所提及的存储器中,在说明书中使用的措词“寄存器”在每个所提及的存储器中指低容量(一些二进制数据)的存储器区域以及大容量(使得能够存储整个程序或者表示所计算出的或要显示的数据的数据的全部或一部分)的存储器区域这两者。ROM 62包含程序和编码参数(诸如阈值TH)。根据本发明的编码方法的算法存储在ROM 62中。当接通时,CPU 61将程序620上载到RAM中并且执行对应的指令。

[0221] RAM 63包含寄存器中的由CPU 61执行并且在接通编码器1之后上载的程序、寄存器中的输入数据、寄存器中的编码方法的不同状态下的编码的数据以及寄存器中的用于编码的其他变量。

[0222] 图13表示解码器2的示例性架构。解码器被配置为执行解码方法的步骤。解码器2包含由数据和地址总线74链接到一起的以下元件：

- [0223] -微处理器71 (或CPU) ,其例如是DSP (或数字信号处理器) ;
- [0224] -ROM (或只读存储器) 72;
- [0225] -RAM (或随机存取存储器) 73;
- [0226] -I/O接口75,用于从应用接收要传送的数据;以及
- [0227] -电池76。

[0228] 根据变型,电池76在编码器的外部。图13中的这些元件中的每个被本领域技术人员所公知,并且将不做进一步公开。在每个所提及的存储器中,在说明书中使用的措词“寄存器”可以对应于小容量(一些比特)的区域或者非常大的区域(例如整个程序或大量的接收或解码的数据)。ROM 72包含至少一个程序和解码器参数。根据本发明的解码方法的算法存储在ROM 72中。当接通时,CPU 71将程序720上载到RAM中,并且执行对应的指令。

[0229] RAM 73包含寄存器中的由CPU 71执行并且在解码器2的接通之后上载的程序、寄存器中的输入数据、寄存器中的解码方法的不同状态下的解码的数据以及寄存器中的用于解码的其他变量。

[0230] 在本文中描述的实现方式可以实现为例如方法或过程、装置、软件程序、数据流或信号。尽管仅在单一形式的实现方式的背景下进行了讨论(例如,仅作为方法或设备进行了讨论),但是所讨论的特征的实现方式也可以实现为其他形式(例如程序)。装置可以实现为例如适当的硬件、软件和固件。所述方法可以实现在诸如例如通常被称为处理设备(例如包括计算机、微处理器、集成电路或可编程的逻辑设备)的处理器这样的装置中。处理器还包括通信设备,诸如例如计算机、手机、便携/个人数字助理(“PDA”)以及便于与终端用户进行信息通信的其他设备。

[0231] 在本文中所述的各种处理和特征的实现方式可以实施为各种不同的装备或应用,具体地例如装备或应用。这样的装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、网络服务器、机顶盒、膝上型电脑、个人计算机、手机、PDA以及其他通信设备。应当清楚的是,装备可以是移动的,甚至可以安装在移动交通工具中。

[0232] 另外,方法可以通过由处理器执行的指令来实现,并且这样的指令(和/或由实现方式产生的数据值)可以存储在处理器可读取的介质上,诸如例如集成电路、软件载体或诸如例如硬盘、压缩盘(“CD”)、光盘(诸如例如,往往被称为数多功能盘或数字视频盘的DVD)、随机存取存储器(“RAM”)或只读存储器(“ROM”)这样的其他存储设备。指令可以形成有形地实现在处理器可读取的介质上的应用程序。指令例如可以在硬件、固件、软件或组合中。指令例如可以在操作系统、单独的应用或两者的结合中。因此,可以将处理器表征为例如被配置为执行处理的设备以及包括具有用于执行处理的指令的处理器可读取的介质(诸如存储设备)的设备这两者。另外,除了指令之外或者替代指令,处理器可读取的介质还可以存储由实现方式产生的数据值。

[0233] 对于本领域的技术人员将是显而易见的是,实现方式可以产生被格式化以携带例如可以被存储或传送的信息的各种信号。信息可以包括例如用于执行方法的指令或者由所述实现方式之一产生的数据。例如,信号可以被格式化为携带用于写入或读取所述实施例的语法的规则作为数据或者携带由所述实施例写入的实际语法值作为数据。例如,这样的信号可以被格式化为电磁波(例如,使用频谱的射频部分)或者基带信号。格式化可以包括例如对数据流进行编码以及用编码的数据流来调制载波。信号携带的信息可以是例如模拟或数字信息。信号可以通过各种不同的有线或无线链路来传送,如已知的那样。信号可以存储在处理器可读取的介质上。

[0234] 已经描述了许多实现方式。然而,将理解的是,可以进行各种修改。例如,可以组合、补充、修改或去除不同的实现方式的元件以产生其他实现方式。另外,本领域的技术人员将理解到,其他过程可以替换所公开的那些,并且得到的实现方式将以与所公开的实现方式至少基本相同的方式来执行至少基本相同的功能以获得至少基本相同的结果。因此,本申请想到这些以及其他实现方式。具体地,根据本发明的用于对LUT进行编码的方法可以用于对任何类型的LUT进行编码,无论与每个顶点相关联的值的类型如何。

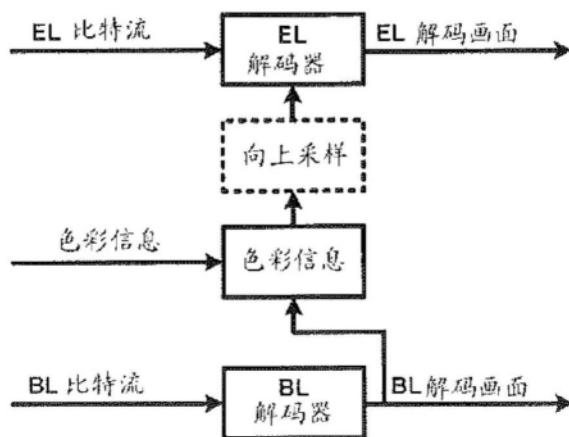


图1

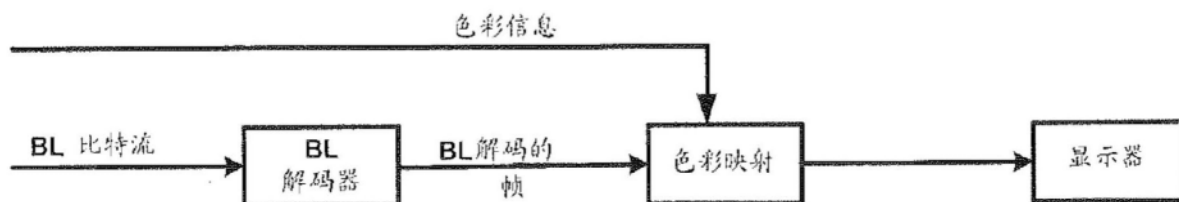


图2

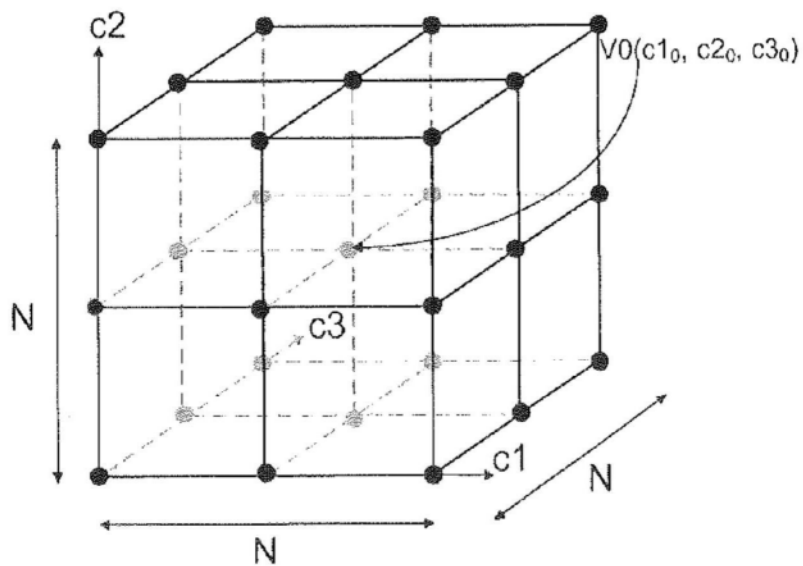


图3



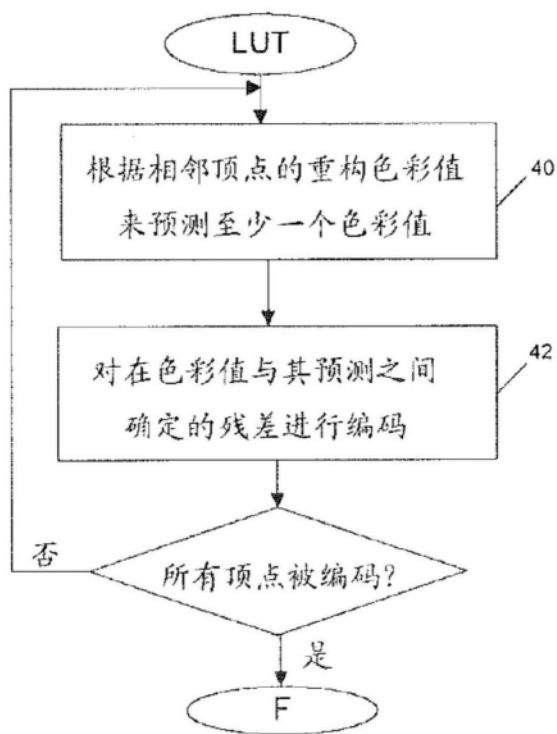


图4

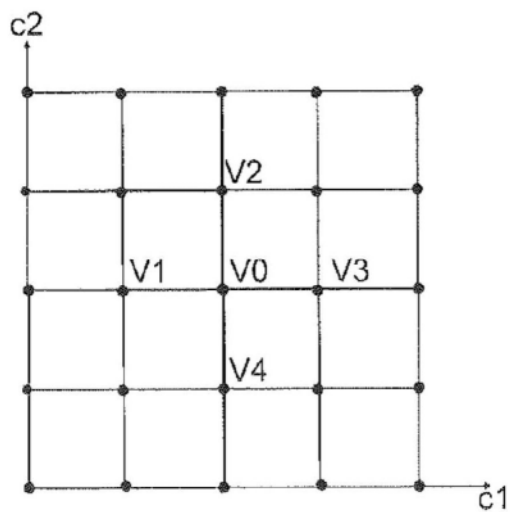


图5

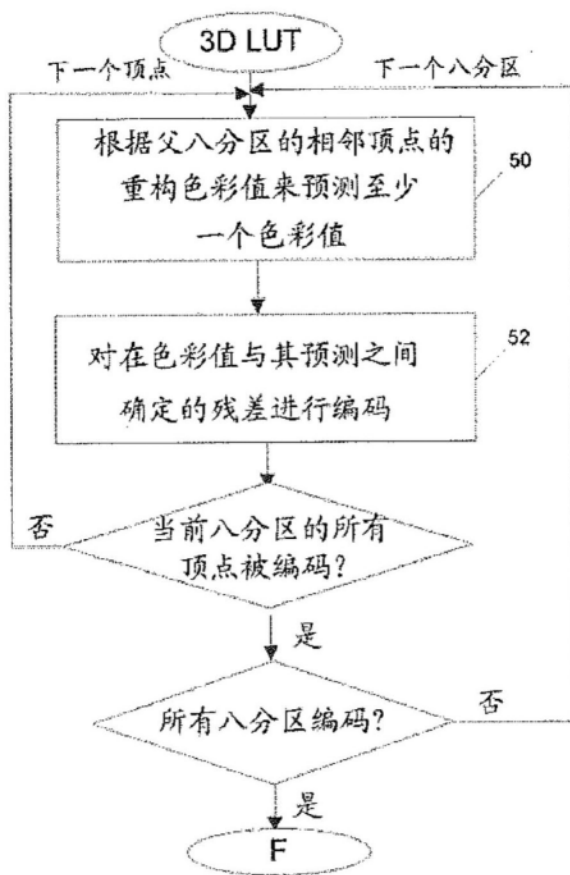


图6

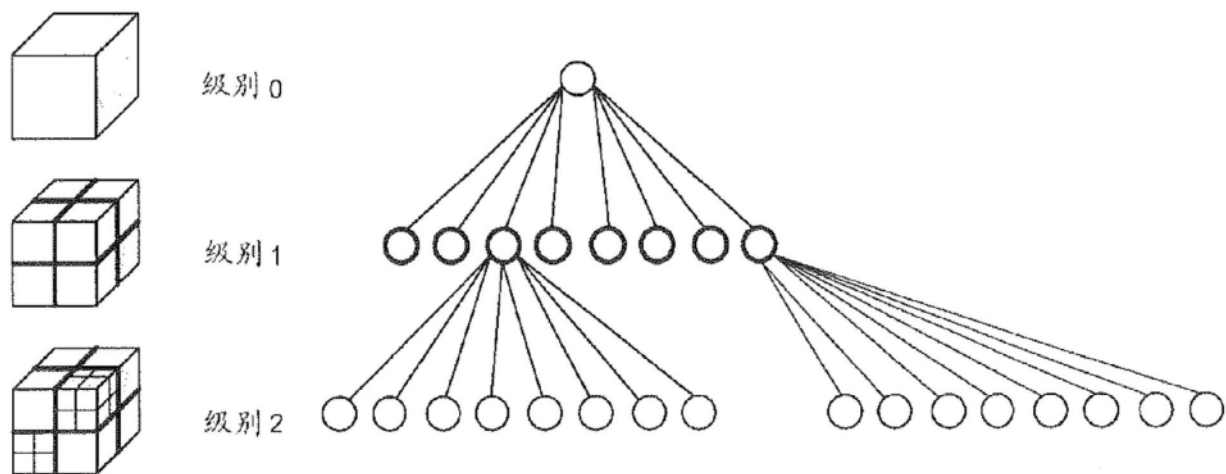


图7A

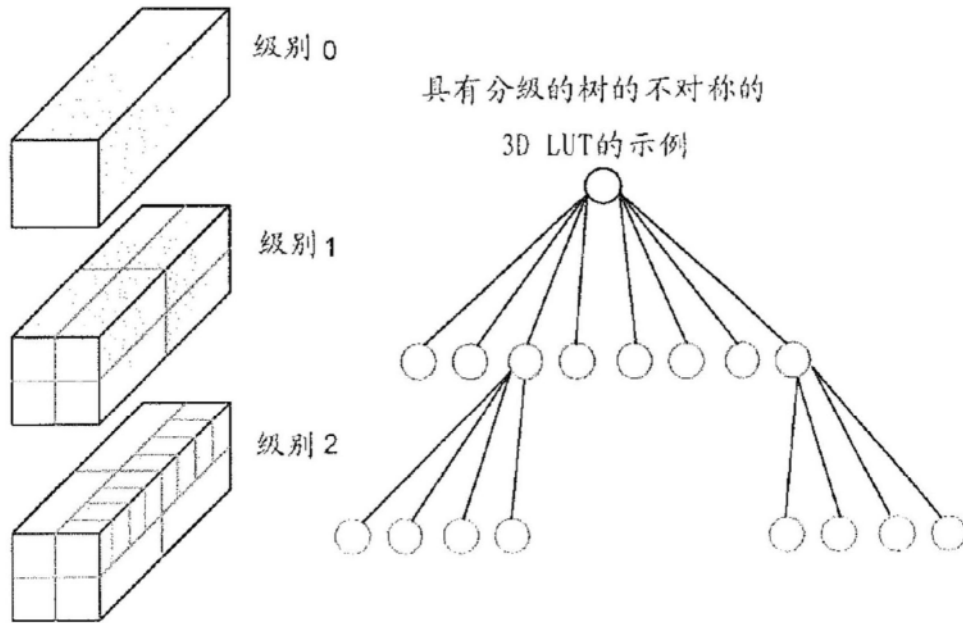


图7B

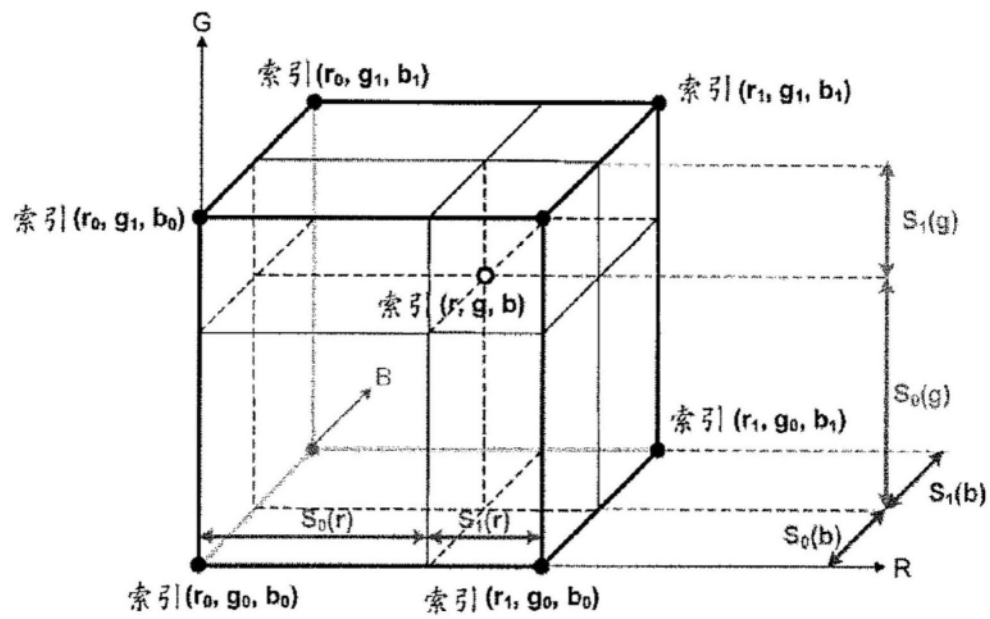


图8

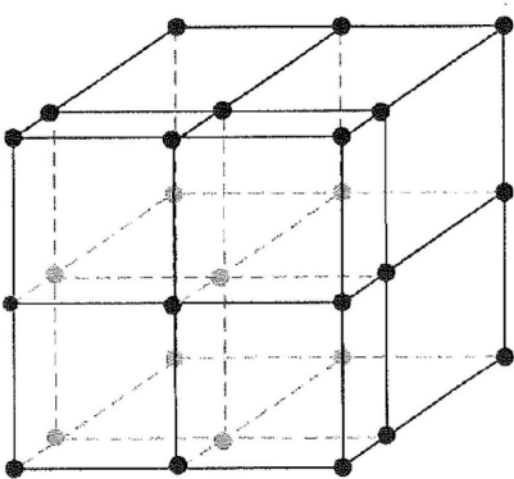


图9A

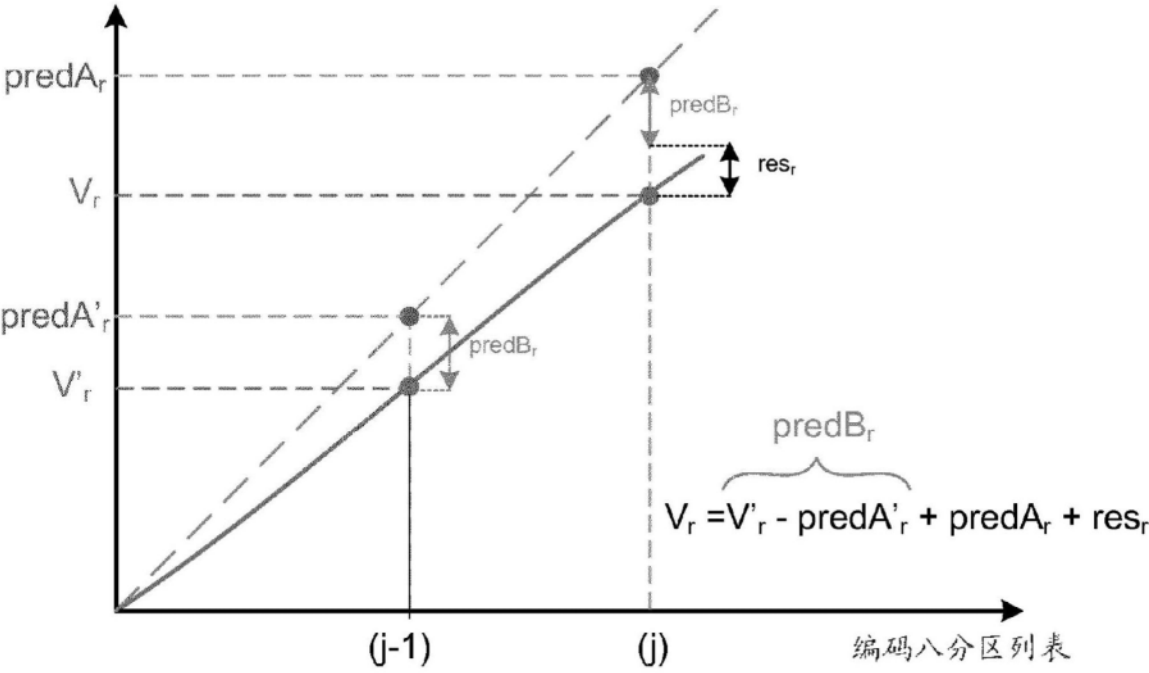


图9B

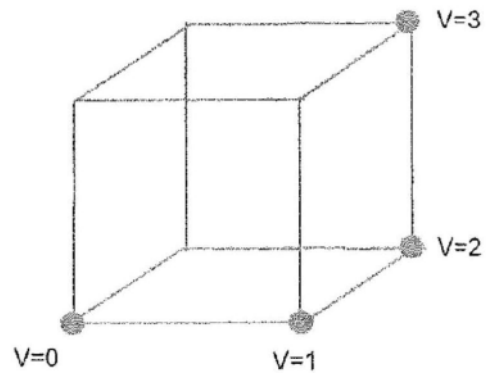


图9C

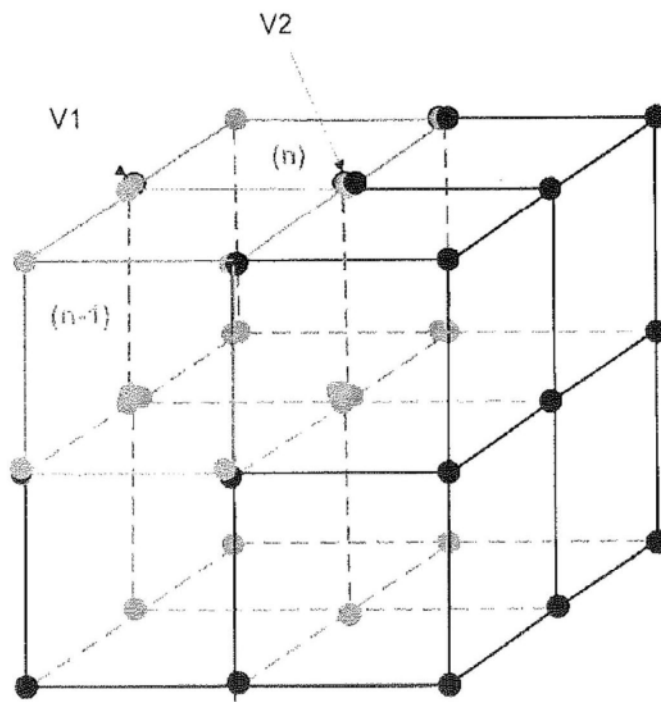


图9D

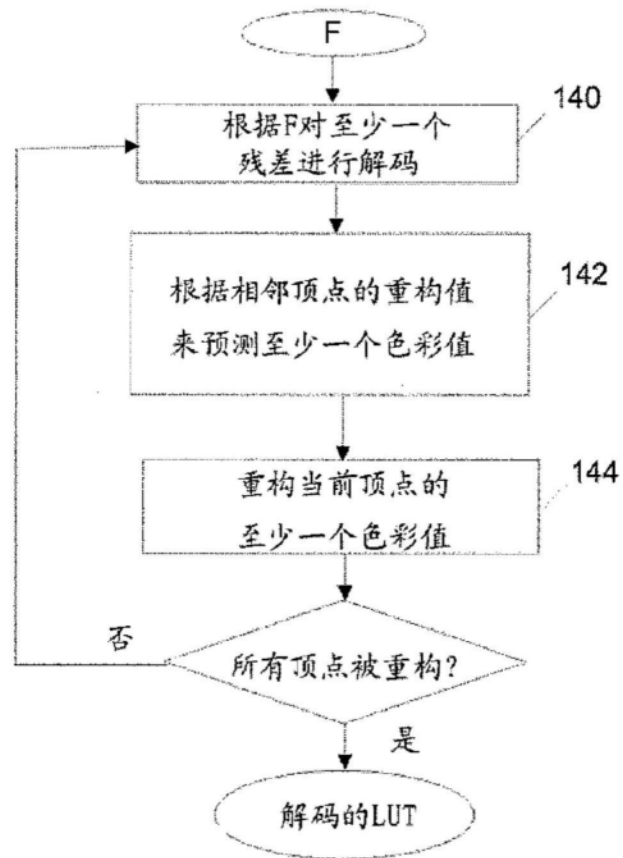


图10

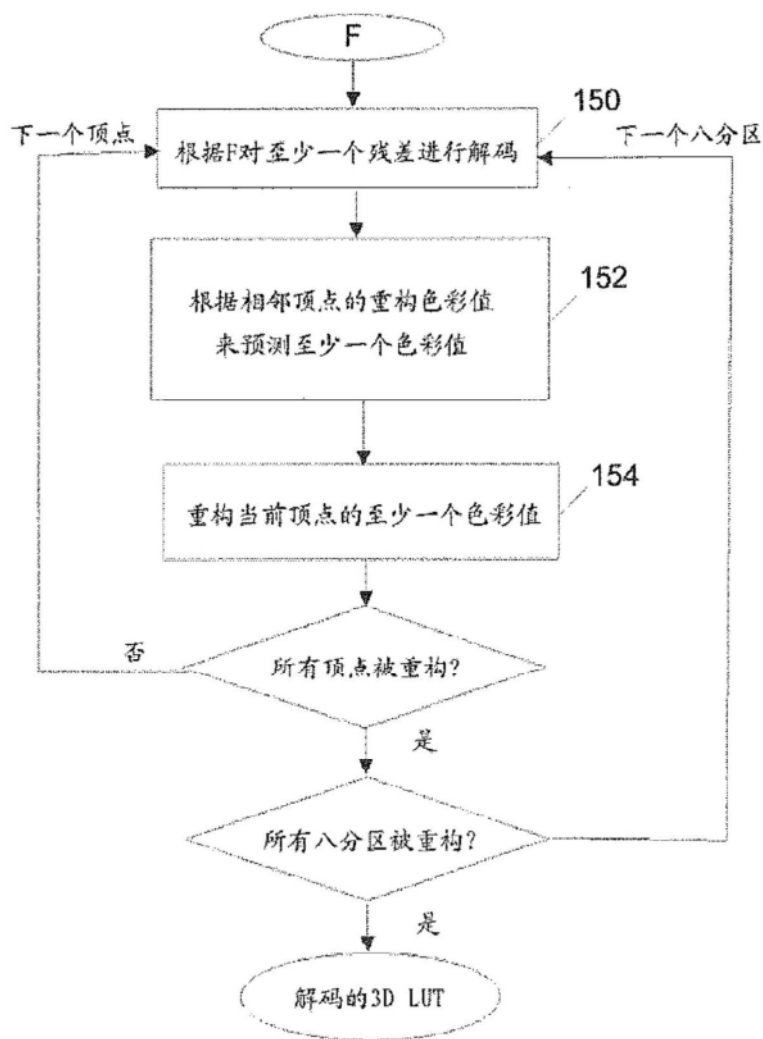


图11

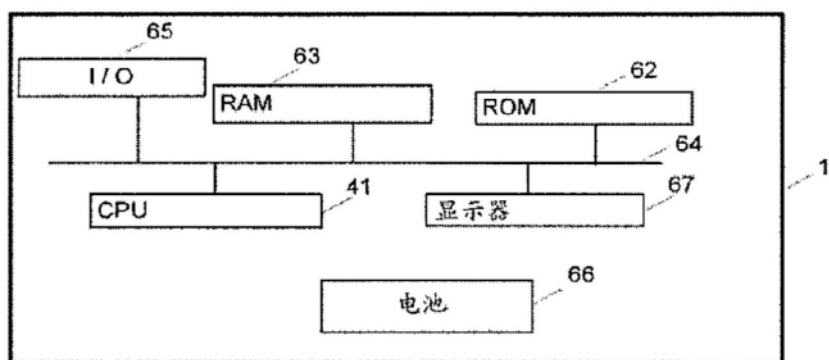


图12

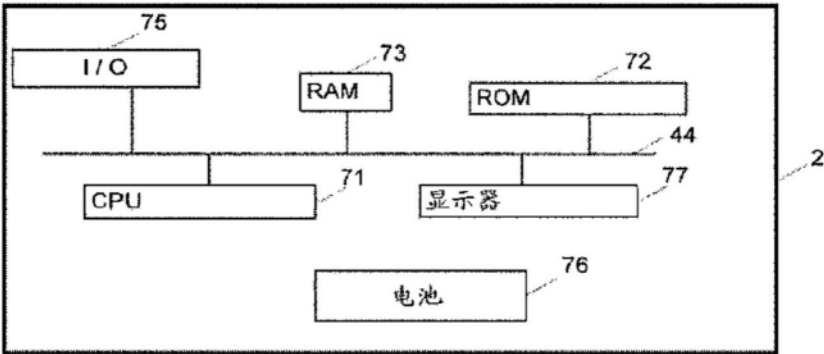


图13

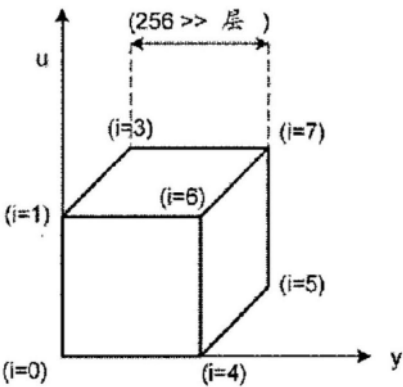


图14