



(12)发明专利

(10)授权公告号 CN 106797468 B

(45)授权公告日 2018.10.30

(21)申请号 201580045958.5

H04N 19/593(2014.01)

(22)申请日 2015.10.02

H04N 19/103(2014.01)

(65)同一申请的已公布的文献号
申请公布号 CN 106797468 A

H04N 19/463(2014.01)

H04N 19/91(2014.01)

H04N 19/186(2014.01)

(43)申请公布日 2017.05.31

H04N 19/182(2014.01)

(30)优先权数据

62/060,450 2014.10.06 US

(56)对比文件

WO 2013177975 A1,2013.12.05,

CN 101340587 A,2009.01.07,

(85)PCT国际申请进入国家阶段日
2017.02.27

Wei Pu et al..AHG10: Simplification of Palette Based Coding.《Joint

(86)PCT国际申请的申请数据

PCT/US2015/053683 2015.10.02

Collaborative Team on Video Coding (JCT-VC) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11 17th Meeting, 27 March - 4 April 2014, JCTVC-Q0074》.2014,

(87)PCT国际申请的公布数据

W02016/057323 EN 2016.04.14

Jianqing Zhu et al..AHG10: Modified copy above mode for palette based coding.《Joint Collaborative Team on Video Coding (JCT-VC) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11 17th Meeting, 27 March - 4 April 2014, JCTVC-Q0174》.2014,

(73)专利权人 华为技术有限公司

地址 518129 广东省深圳市龙岗区坂田华为总部办公楼

(72)发明人 于浩平 王炜 马展 徐猛

审查员 陟爽

(51)Int.Cl.

H04N 19/176(2014.01)

H04N 19/70(2014.01)

H04N 19/13(2014.01)

权利要求书3页 说明书14页 附图7页

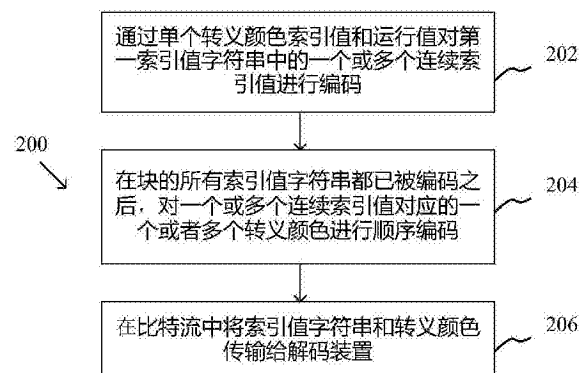
(54)发明名称

高效视频编码(HEVC)屏幕内容编码(SCC)中改进的调色板模式

义颜色传输给解码装置。

(57)摘要

编码装置、解码装置和编码方法。所述编码装置包括:处理器,用于:接收包括屏幕内容的视频帧;生成包含所述视频帧中的屏幕内容的颜色的索引图的块,其中所述块包括共享相同转义颜色索引值的索引值字符串,所述转义颜色索引值表示转义颜色;对所述块中的每个所述索引值字符串进行编码;以及在所述块中的每个所述索引值字符串都已编码之后,对所述转义颜色进行顺序编码。可操作地耦合至所述处理器的发射器用于在比特流中将所述索引值字符串和所述转



1. 一种编码方法,其特征在于,包括:
接收包括屏幕内容的视频帧;
生成包含所述视频帧中的屏幕内容的颜色的索引图的块,其中所述块包括索引值字符串;
处理器通过单个转义颜色索引值和标识有多少连续转义颜色索引值已被编码的运行值对第一索引值字符串中的一个或多个连续索引值进行编码;
在所述块的所有索引值字符串都已被编码之后,所述处理器对所述一个或多个连续索引值对应的转义颜色进行顺序编码;以及
可操作地耦合至所述处理器的发射器在比特流中将所述索引值字符串和所述转义颜色传输给解码装置。
2. 如权利要求1所述的方法,其特征在于,COPY_INDEX_MODE用于对所述一个或多个连续索引值和所述运行值进行编码。
3. 如权利要求1所述的方法,其特征在于,至少两个所述转义颜色是互不相同的。
4. 如权利要求1所述的方法,其特征在于,每个所述转义颜色都是相同的颜色。
5. 如权利要求1所述的方法,其特征在于,还包括:通过从所述第一索引值字符串中复制所述单个转义颜色索引值,对包括两个或以上所述连续索引值的第二索引值字符串进行编码,尽管所述第一索引值字符串中和所述第二索引值字符串中的至少一些转义颜色互不相同。
6. 如权利要求4所述的方法,其特征在于,COPY_ABOVE_MODE用于从所述第一索引值字符串中复制所述单个转义颜色索引值。
7. 如权利要求4所述的方法,其特征在于,所述第二索引值字符串紧接在所述第一索引值字符串的下面。
8. 如权利要求1所述的方法,其特征在于,所述屏幕内容为文本和计算机图形内容之一。
9. 如权利要求1所述的方法,其特征在于,所述屏幕内容不包括任何摄像机捕获的视频。
10. 如权利要求1所述的方法,其特征在于,所述第一索引值字符串是所述块内的顶部字符串。
11. 如权利要求1所述的方法,其特征在于,所述索引值字符串中的每个索引值都是颜色的数值表示。
12. 如权利要求1所述的方法,其特征在于,所述编码和所述顺序编码为无损编码。
13. 一种编码方法,其特征在于,包括:
接收包括屏幕内容的视频帧;
生成包含所述视频帧中的屏幕内容的颜色的索引图的块,其中所述块包括索引值字符串;
处理器对包括转义颜色索引值的第一索引值字符串进行编码;
所述处理器通过从所述第一索引值字符串中复制所述转义颜色索引值对包括所述转义颜色索引值的第二索引值字符串进行编码;
在所述块的所有索引值字符串都已被编码之后,所述处理器对所述第一索引值字符串

中和所述第二索引值字符串中的所述转义颜色索引值对应的转义颜色进行顺序编码;以及可操作地耦合至所述处理器的发射器在比特流中将所述索引值字符串和所述转义颜色传输给解码装置。

14. 如权利要求13所述的方法,其特征在于,所述第一字符串的所述转义颜色索引值对应的转义颜色不同于所述第二字符串的所述转义颜色索引值对应的转义颜色。

15. 如权利要求13所述的方法,其特征在于,所述第一字符串的所述转义颜色索引值对应的转义颜色与所述第二字符串的所述转义颜色索引值对应的转义颜色相同。

16. 如权利要求13所述的方法,其特征在于,COPY_ABOVE_MODE用于从所述第一索引值字符串中复制所述转义颜色索引值。

17. 如权利要求13所述的方法,其特征在于,所述转义颜色索引值是从所述第一索引值字符串中复制的,尽管所述第一索引值字符串的所述转义颜色索引值表示第一转义颜色,所述第二索引值字符串表示不同于所述第一转义颜色的第二转义颜色。

18. 如权利要求13所述的方法,其特征在于,所述第一字符串中的所述转义颜色索引值是多个连续转义颜色索引值中的一个,所述第二字符串中的所述转义颜色索引值是多个连续转义颜色索引值中的一个。

19. 如权利要求13所述的方法,其特征在于,所述索引值字符串表示屏幕内容,所述屏幕内容由非摄像机捕获的图像组成。

20. 一种编码装置,其特征在于,包括:

处理器,用于:

接收包括屏幕内容的视频帧;

生成包含所述视频帧中的屏幕内容的颜色的索引图的块,其中所述块包括共享相同转义颜色索引值的索引值字符串,所述转义颜色索引值表示转义颜色;

对所述块中的每个所述索引值字符串进行编码;以及

在所述块中的每个所述索引值字符串都已被编码之后,对所述转义颜色进行顺序编码;以及

发射器,可操作地耦合至所述处理器,并用于在比特流中将所述索引值字符串和所述转义颜色传输给解码装置。

21. 如权利要求20所述的编码装置,其特征在于,来自第二索引值字符串的转义颜色索引值是对来自第一索引值字符串的转义颜色索引值的精确复制。

22. 如权利要求20所述的编码装置,其特征在于,单个索引值字符串中的转义颜色是连续的。

23. 一种解码方法,其特征在于,包括:

使用解码器扫描块中的索引值字符串,其中,一个或多个所述索引值字符串包含相同转义颜色索引值,所述转义颜色索引值表示转义颜色;以及

使用所述解码器顺序扫描在所述块中的所述索引值字符串之后被编码的转义颜色,以确定与所述一个或多个索引值字符串中的每个所述相同转义颜色索引值相对应的转义颜色。

24. 如权利要求23所述的方法,其特征在于,还包括:所述解码器接收编码器传输的比特流中的所述块。

25. 如权利要求24所述的方法,其特征在于,还包括:解析比特流以确定所述索引值字符串和在所述索引值字符串之后被顺序编码的所述转义颜色。

26. 如权利要求23所述的方法,其特征在于,来自所述块中第二字符串的转义颜色索引值是对来自所述块中第一字符串的转义颜色索引值的精确复制。

27. 如权利要求23所述的方法,其特征在于,至少一个所述索引值字符串包括使用单个转义颜色索引值的多个连续索引值。

28. 一种解码装置,其特征在于,包括:

接收器,用于接收包含块中的索引值字符串的比特流,其中,一个或多个所述索引值字符串包含相同转义颜色索引值,所述转义颜色索引值表示转义颜色;以及

处理器,可操作地耦合至所述接收器,并用于在所述块中的每个所述索引值字符串都被解码之后,对所述转义颜色进行顺序解码。

29. 如权利要求28所述的解码装置,其特征在于,来自第二字符串的转义颜色索引值是对来自第一字符串的转义颜色索引值的精确复制。

30. 如权利要求28所述的解码装置,其特征在于,一个或多个所述索引值字符串中的所述转义颜色索引值为相同的数值。

高效视频编码 (HEVC) 屏幕内容编码 (SCC) 中改进的调色板模式

[0001] 关于由联邦政府赞助研究或开发的声明

[0002] 本申请要求于2014年10月6日Haoping Yu等人递交的第62/060,450号、发明名称为“HEVC SCC中改进的调色板模式”的美国临时专利申请案的在先申请优先权,该在先申请案以引入的方式并入本文。

[0003] 参考缩微胶片附录

[0004] 不适用

背景技术

[0005] 即使是描述相对较短的电影也需要大量的视频数据量,这可能导致在流式传输或通过具有有限带宽容量的通信网络传送数据时出现困难。因此,视频数据通常在通过现代电信网络传送之前被压缩。视频压缩设备通常在源头使用软件和/或硬件于传输之前对视频数据进行编码,从而减少表示数字视频图像所需的数据量。然后由解码视频数据的视频解压设备在目的地接收压缩数据。随着有限的网络资源和对更高视频质量日益增长的需求,在不增加比特率的情况下提高图像质量的改进的压缩和解压缩技术合乎所需。

发明内容

[0006] 在一实施例中,本发明包括一种编码方法,包括:接收包括屏幕内容的视频帧;生成包含所述视频帧中的屏幕内容的颜色的索引图的块,其中所述块包括索引值字符串;处理器通过单个转义颜色索引值和标识有多少连续转义颜色索引值已被编码的运行值对第一索引值字符串中的一个或多个连续索引值进行编码;在所述块的所有索引值字符串都已被编码之后,所述处理器对所述一个或多个连续索引值对应的转义颜色进行顺序编码;以及可操作地耦合至所述处理器的发射器在比特流中将所述索引值字符串和所述转义颜色传输给解码装置。

[0007] 在另一实施例中,本发明包括一种编码方法,包括:接收包括屏幕内容的视频帧;生成包含所述视频帧中的屏幕内容的颜色的索引图的块,其中所述块包括索引值字符串;处理器对包括转义颜色索引值的第一索引值字符串进行编码;所述处理器通过从所述第一索引值字符串中复制所述转义颜色索引值对包括所述转义颜色索引值的第二索引值字符串进行编码;在所述块的所有索引值字符串都已被编码之后,所述处理器对所述第一索引值字符串中和所述第二索引值字符串中的所述转义颜色索引值对应的转义颜色进行顺序编码;以及可操作地耦合至所述处理器的发射器在比特流中将所述索引值字符串和所述转义颜色传输给解码装置。

[0008] 在又一实施例中,本发明包括一种编码装置,包括:处理器,用于:接收包括屏幕内容的视频帧;生成包含所述视频帧中的屏幕内容的颜色的索引图的块,其中所述块包括共享相同转义颜色索引值的索引值字符串,所述转义颜色索引值表示转义颜色;对所述块中的每个所述索引值字符串进行编码;以及在所述块中的每个所述索引值字符串都已被编码

之后,对所述转义颜色进行顺序编码;以及发射器,可操作地耦合至所述处理器,并用于在比特流中将所述索引值字符串和所述转义颜色传输给解码装置。

[0009] 通过以下结合附图和权利要求的详细描述,这些以及其它特征将会被更清楚地理解。

附图说明

[0010] 为了更透彻地理解本发明,现参阅结合附图和具体实施方式而描述的以下简要说明,其中的相同参考标号表示相同部分。

[0011] 图1是用于使用调色板模式说明索引图编码的块中的索引图颜色的图;

[0012] 图2A是使用简化上下文模型进行编码的方法的一实施例的流程图;

[0013] 图2B是使用简化上下文模型进行编码的方法的一实施例的流程图;

[0014] 图2C是使用简化上下文模型进行编码的方法的一实施例的流程图;

[0015] 图3是分组索引编码方案的一实施例的图;

[0016] 图4是大规模并行处理的一实施例的图;

[0017] 图5是视频编码器的一实施例的示意图;

[0018] 图6是视频解码器的一实施例的示意图;

[0019] 图7是可以包括编码器和解码器的网络单元的一实施例的示意图;

[0020] 图8是典型的通用网络组件或计算机系统的示意图。

具体实施方式

[0021] 首先应理解,尽管下文提供一项或多项实施例的说明性实施方案,但所公开的系统 and/或方法可使用任何数目的技术来实施,无论该技术是当前已知还是现有的。本发明决不应限于下文所说明的说明性实施方案、附图和技术,包括本文所说明并描述的示例性设计和实施方案,而是可在所附权利要求书的范围以及其等效物的完整范围内修改。

[0022] 通常,视频媒体涉及相对快速连续地显示静止图像或帧的序列,从而使观看者感知运动。每个帧可包括多个图像元素或像素,每个图像元素或像素可表示帧中的单个参考点。在数字处理期间,可以给每个像素分配整数值(例如,0、1、……、或255),该整数值表示在相应参考点处的图像质量或特征,如亮度或色度。在使用中,图像或视频帧可包括大量像素(例如,1920×1080帧中的2,073,600个像素)。因此,独立地对每个像素进行编码和解码(以下简称为编码)可能繁琐又低效。为了提高编码效率,通常会将视频帧分成多个矩形块或宏块,其可以作为诸如预测、变换和量化等处理的基本单元。例如,典型的NxN块可包括N²个像素,其中N是大于1的整数且通常是4的倍数。

[0023] 在国际电信联盟(International Telecommunications Union,简称ITU)电信标准化部门(International Telecommunications Union Telecommunications Standardization Sector,简称ITU-T)和国际标准化组织(International Organization for Standardization,简称ISO)/国际电工技术委员会(International Electrotechnical Commission,简称IEC)中,引入了用于高效视频编码(High Efficiency Video Coding,简称HEVC)的新的块概念。例如,编码单元(Coding Unit,简称CU)可以指将视频帧子分割成的具有相等或可变尺寸的矩形块。在HEVC中,CU可以替换以前的标准的宏

块结构。取决于帧间或帧内预测的模式, CU可包括一个或多个预测单元 (Prediction Unit, 简称PU), 每个预测单元可作为预测的基本单位。例如, 对于帧内预测, 8x8CU可以对称地分成四个4x4PU。再如, 对于帧间预测, 64x64CU可以不对称地分成16x64PU和48x64PU。类似地, PU可包括一个或多个变换单元 (Transform Unit, 简称TU), 每个变换单元可作为变换和/或量化的基本单元。例如, 32x32PU可以对称地分成四个16x16TU。一个PU的多个TU可以共享相同的预测模式, 但是却可以单独变换。此处, 术语块通常可以指宏块、CU、PU或TU中的任意一个。

[0024] 取决于本申请, 可以以无损模式 (例如, 无失真或信息丢失) 或有损模式 (例如, 具有失真) 对块进行编码。在使用中, 可以使用无损模式对高质量视频 (例如, 具有4:4:4的YUV二次采样) 进行编码, 而可以使用有损模式对低质量视频 (例如, 具有4:2:0的YUV二次采样) 进行编码。如此处所使用的, YUV中的Y分量指的是颜色的亮度 (亮度), 而U和V分量指的是颜色本身 (色度)。有时, 单个视频帧或切片 (例如, 具有4:4:4或4:2:0的YUV二次采样) 可以同时使用无损和有损模式对多个区域进行编码, 所述多个区域可以是矩形或不规则形状的。每个区域可包括多个块。例如, 复合视频可包括诸如文本和计算机图形内容 (例如, 非摄像机捕获的图像) 和自然视图内容 (例如, 摄像机捕获的视频) 等不同类型的内容的组合。在复合帧中, 可以以无损模式对文本和图形的区域进行编码, 而可以以有损模式对自然视图内容的区域进行编码。在例如计算机屏幕共享应用中可能需要文本和图形的无损编码, 因为有损编码可能会导致文本和图形的质量或保真差, 这可能会导致眼睛疲劳。

[0025] 随着半导体、组网、通信、显示器、计算机以及诸如平板电脑和智能手机等设备的快速且持续的进步, 许多应用需要基于HEVC的压缩/编码解决方案, 其能够以高视觉质量有效地压缩非摄像机捕获的视频内容。该非摄像机捕获的视频内容, 此处可称为屏幕内容, 可包括计算机生成的图形、具有应用中常见的如窗口切换和移动、文本滚动等典型动作的文本。在许多情况下, 所述非摄像机捕获的视频内容以高对比度提供具有不同颜色的清晰纹理和锐利边缘, 并且可以具有4:4:4颜色采样格式。

[0026] 当前HEVC屏幕内容编码引入了调色板模式以更有效地表示计算机屏幕。在以引用方式并入文本, R. Joshi和J. Xu的2014年10月 (HEVC SCC), 法国斯特拉斯堡, HEVC屏幕内容编码的工作草案2, MPEG-N14969/JCTVC-S1005中, 描述了调色板模式。所述调色板模式也在屏幕内容编码测试模型 (Screen Content Coding Test Model, 简称SCM) 2.0参考软件中使用。

[0027] 尽管所述调色板模式在当前HEVC框架内提供了效率, 但仍然还有改进的空间。此处公开了用于改进的视频编码的系统和方法。本发明提供了简化的熵 (例如, 无损) 编码方案。为了降低整体复杂度, 编码方案会在索引图处理结束时对转义值进行编码。通过这样做, 即使索引图包含转义颜色相对应的索引值, 编码方案在使用例如COPY_ABOVE_MODE时也会增加索引匹配的数量。另外, 由于转义值的交织编码和索引的其它语法, HEVC SCC中的当前解决方案不支持索引图的并行处理, 而与HEVCSCC中的当前解决方案相反, 该编码方案支持索引图和转义颜色的并行处理。此外, 该编码方案简化了索引图编码的设计, 在不牺牲性能的情况下提高了吞吐量, 并且使得编码更易于描述和实现。

[0028] 在可能特别适合于屏幕生成内容编码的基于调色板的编码中, 视频编码器 (例如, 视频编码器或视频解码器) 形成了表示给定块的视频数据的颜色的“调色板”。调色板可包

括给定块中的最主要(例如,经常使用)的颜色。给定块的视频数据中不经常或很少表现的颜色不包括在调色板中。不包括在调色板中的颜色称为转义颜色。

[0029] 在调色板模式编码期间对与给定块相对应的索引图进行编码时,给调色板中包括的每个颜色分配索引值。例如,如果调色板中包括黑色和白色,则白色可以具有索引值0,黑色可以具有索引值1。另外,不包括在调色板中的每个颜色也被分配了例如单个或共同索引值。例如,如果调色板中不包括蓝色、绿色和红色,则这些颜色都会具有索引值3。不包括在调色板中的颜色的索引值可称为转义颜色索引值。

[0030] 当前HEVC SCC草案使用基于运行的一维(One-Dimensional,简称1-D)字符串复制。尽管如此,在以引用方式并入文本,由W.Wang、Z.Ma、M.Xu、H.Yu的2014年10月,法国斯特拉斯堡,第JCTVC-S0151号,“非CE6:HEVC SCC中调色板模式的2-D索引图编码”以及于2014年10月递交的第62/060,450号,名称为“HEVC SCC中的调色板模式”的美国临时专利申请案中,已经提出了二维(Two-Dimensional,简称2-D)字符串复制方法。虽然为了简明起见在本文中完全没有描述,但是本领域技术人员将理解,在某些情况下,2-D字符串复制方法可以默认为基于运行的1-D字符串复制方法。

[0031] 当使用1-D字符串方法进行调色板模式的索引模式编码时,对于每个CU,涉及两个主要部分。这两个部分是颜色表处理和索引图编码。举例来说,1-D字符串方法的索引图编码可以使用COPY_ABOVE模式,其中COPY_ABOVE_MODE应用于指示当前字符串中的索引是否与来自当前字符串正上方的字符串中的索引相同。

[0032] 图1示出了将用于提供使用调色板模式的索引图编码的示例的索引图颜色的8x8块100。如图所示,块100中的顶部字符串102具有索引值1、1、1、1、2、2、2和2(从左到右),紧接在顶部字符串102下面的字符串104具有索引值1、1、1、1、1、1、1和1,块100中的下一个字符串106具有索引值1、1、1、1、1、1、1和1,块100中的下一个字符串108具有索引值1、1、3(其中3表示蓝色)、3(其中3表示绿色)、2、2、2和2,块100中的下一个字符串110具有索引值1、1、3(其中3表示绿色)、(其中3表示蓝色)、2、2、2和2,块100中的下一个字符串112具有索引值1、1、3(其中3表示红色)、2、2、2、2和2,块100中的下一个字符串114具有索引值2、2、2、2、2、2、2和2,块100中的底部字符串116具有索引值2、2、2、2、2、2、2和2。为了在比特流中对第一字符串102进行编码,使用了palette_run_type标志(例如,1位palette_run_type_flag)、索引值以及运行值。

[0033] 该palette_run_type标志指示当前字符串上面的字符串中的任意索引值是否已被复制。如果上面的字符串的一部分已被复制,则将该palette_run_type标志设置为表示COPY_ABOVE_MODE的第一二进制数(例如,1)。如果上面的字符串尚未被复制,则将该palette_run_type标志设置为表示COPY_INDEX_MODE的第二二进制数(例如,0)。对顶部字符串102进行编码时,将palette_run_type标志默认设置为0,因为顶部字符串102的上面没有设置字符串。索引值是在块100中的字符串内表示的特定数值(例如,1或2)。运行值指可以复制多少个连续索引值。例如,如果运行值设置为1,则复制单个索引值。如果运行值设置为2,则复制两个连续索引值。如果运行值设置为3,则复制三个连续索引值,等等。所以,为了对具有索引值1、1、1、1、2、2、2、2的顶部字符串102进行编码,使用以下语法:palette_run_type_flag=0,索引值=1,运行值=4,palette_run_type_flag=0,索引值=2,运行值=4。

[0034] 为了对具有索引值1、1、1、1、1、1、1的下一个字符串104进行编码,可以使用以下语法:palette_run_type_flag=1,运行值=4,palette_run_type_flag=0,索引值=1,运行值=4。或者,也可以使用以下语法:palette_run_type_flag=0,运行值=8。为了对具有索引值1、1、1、1、1、1、1的下一个字符串106进行编码,使用以下语法:palette_run_type_flag=1,运行值=8。用于编码的特定语法可以由例如速率失真优化模块或其它合适的编码器组件来确定。速率失真优化模块可以基于例如哪一个语法对编码最有效来选择特定语法。例如,字符串104和106中的所有索引可以被编码为palette_run_type_flag=0,索引值=1,运行值=16。

[0035] 下一个字符串108具有索引值1、1、3、3、2、2、2、2,其中3是对应于转义颜色蓝色或转义颜色绿色的转义颜色索引值。使用现有的传统编码方案,使用以下语法对下一个字符串108进行编码:palette_run_type_flag=1,运行值=2,palette_run_type_flag=0,索引值=3,转义颜色=蓝色,palette_run_type_flag=0,索引值=3,转义颜色=绿色,palette_run_type_flag=0,索引值=2,运行值=4。同样地,使用现有的传统编码方案,使用以下语法对下一个字符串110进行编码:palette_run_type_flag=1,运行值=2,palette_run_type_flag=0,索引值=3,转义颜色=绿色,palette_run_type_flag=0,索引值=3,转义颜色=蓝色,palette_run_type_flag=1,运行值=4。使用现有的传统编码方案,使用以下语法对下一个字符串112进行编码:palette_run_type_flag=1,运行值=2,palette_run_type_flag=0,索引值=3,转义颜色=红色,palette_run_type_flag=1,运行值=5。

[0036] 所以,如上所示,当使用传统编码方案对下一个字符串108、下一个字符串110和下一个字符串112进行编码时,紧接在块100中的转义颜色索引值之后对特定转义颜色的标识进行编码。遗憾的是,以这种方式进行编码有缺点。例如,当同一字符串或相邻字符串中的转义颜色互不相同,充分利用COPY_ABOVE_MODE和COPY_INDEX_MODE的能力会受到限制。例如,当下一个字符串108中的转义颜色索引值3对应的转义颜色是蓝色,且下一个字符串110中的转义颜色索引值3对应的转义颜色是绿色时,会因为每个字符串的转义颜色不同而丢失使用COPY_ABOVE_MODE的能力。类似地,需要使用两个palette_run_type_flag=0的语法元素来单独地对两个连续转义颜色进行编码。

[0037] 然而,已经发现存在对块100中的字符串(例如,字符串108-112)进行编码的方法,使得即使在相邻字符串的转义颜色互不相同,也可以使用COPY_ABOVE_MODE和COPY_INDEX_MODE功能。在这方面,提出了一种编码方案,其中,将转义颜色索引值编码为对字符串进行编码的一部分,而转义颜色本身则是在索引图编码结束时被顺序编码而不是紧接在转义颜色索引值之后就被编码。再次参阅图1,所提出的新编码方案的示例介绍如下。

[0038] 为了对下一个字符串108进行编码,使用以下语法:palette_run_type_flag=1,运行值=2,palette_run_type_flag=0,索引值=3,运行值=2,palette_run_type_flag=0,索引值=2,运行值=4。下一个字符串110被编码为:palette_run_type_flag=1,运行值=8。下一个字符串112被编码为:palette_run_type_flag=0,索引值=2,运行值=2,palette_run_type=0,索引值=3,运行值=1,palette_run_type_flag=0,索引值=2,运行值=5。因此,即使这些字符串中的转义颜色是不同的,也能使用具有运行值的COPY_ABOVE_MODE或COPY_INDEX_MODE对字符串108、110和112中的转义颜色进行编码。

[0039] 继续进行对块100的编码,下一个字符串114被编码为:palette_run_type_flag=0,索引值=2,运行值=8。底部字符串116被编码为palette_run_type_flag=1,运行值=8。在一实施例中,下一个字符串110、112、114和116可以被编码为:palette_run_type_flag=1,运行值=11,palette_run_type_flag=0,索引值=2,运行值=21。

[0040] 在块110中的字符串102-116中的每一个都已如上所述被编码之后,在索引图编码结束时对转义颜色进行顺序编码。因此,在底部字符串116的索引图编码之后,对以下语法进行编码以按照它们在块100中遇到的顺序标识每个转义颜色:转义颜色=蓝色,绿色,绿色,蓝色,红色。因此,块中(例如,在字符串108中)遇到的第一个索引值3对应的第一个转义颜色被标识为蓝色,块中(例如,在字符串108中)遇到的第二个索引值3对应的下一个转义颜色被标识为绿色,块中(例如,在字符串110中)遇到的第二个索引值3对应的下一个转义颜色被标识为绿色,块中(例如,在字符串110中)遇到的第二个索引值3对应的下一个转义颜色被标识为蓝色,块中(例如,在字符串112中)遇到的第三个索引值3对应的最后一个转义颜色被标识为红色。通过以这种方式进行编码,实现了种种益处。例如,即使在同一字符串或相邻字符串中的转义颜色互不相同,也可使用COPY_ABOVE_MODE功能。另外,COPY_INDEX_MODE可以用于通过单个转义颜色索引值连同对应于连续转义颜色数量的运行值对连续转义颜色一起进行编码。实际上,在一实施例中,是以相同的方法和/或类似的方式来对转义颜色索引值和其它索引值进行编码的。另外,新的编码方案允许增加的匹配索引,而不必停止编码,允许索引图和索引图颜色的并行处理,并简化了索引图编码的设计。

[0041] 图2A是转义颜色编码方法200的一实施例的流程图。该方法200可以在例如针对将要传输给解码装置的比特流对已接收到视频帧且需要对转义颜色进行有效地编码时实现。在框块202中,通过单个转义颜色索引值(例如,图1中的值3)和标识有多少个连续转义颜色索引值被编码的运行值来对第一索引值字符串(例如,图1的下一个字符串108)中的一个或多个连续索引值进行编码。在框块204中,在块(例如,图1的块100)的所有索引值字符串都被编码之后,对多个连续索引值对应的转义颜色进行编码。在一实施例中,转义颜色互不相同。例如,第一转义颜色可以是蓝色,第二转义颜色可以是绿色,第三转义颜色可以是红色,等等。本领域技术人员在阅读本发明后将理解,也可以以相同的方式对具有转义颜色的附加字符串(例如,顶部字符串102、下一个字符串104、下一个字符串106)进行编码。任意附加字符串中的转义颜色索引值将具有与其它字符串一起使用的相同转义颜色索引值(例如,图1中的值3)。另外,由附加转义颜色索引值中的一个所表示的任意转义颜色相对于第一和第二转义颜色被顺序编码。在框块206中,在比特流中将索引值字符串和转义颜色传输给解码装置。

[0042] 图2B是转义颜色编码方法250的一实施例的流程图。该方法250可以在例如针对将要传输给解码装置的比特流对已接收到视频帧且需要对转义颜色进行有效地编码时实现。在框块252中,对包括转义颜色索引值(例如,图1中的值3)的第一索引值字符串(例如,图1的下一个字符串108)进行编码。在框块254中,通过从第一索引值字符串中复制转义颜色索引值,对包括所述转义颜色索引值(例如,图1中的相同值3)的第二索引值字符串(例如,图1的下一个字符串110)进行编码。在一实施例中,尽管第一索引值字符串的转义颜色索引值表示第一转义颜色(例如,蓝色),第二索引值字符串表示不同于第一转义颜色的第二转义颜色(例如,绿色),该复制也会发生。在一实施例中,第一索引值字符串中的转义颜色索引

值是第一个字符串(例如,第一个字符串1 1 2 3 2 3 1 1)中的多个转义颜色索引值之一,第二个索引值字符串中的转义颜色索引值是第二个字符串(例如,第二个字符串2 1 2 3 2 3 1 1)中的多个转义颜色索引值之一。在一实施例中,第一个索引值字符串中的转义颜色索引值是第一个字符串(例如,第一个字符串1 1 2 3 3 3 1 1)中的多个连续转义颜色索引值之一,第二个索引值字符串中的转义颜色索引值是第二个字符串(例如,第二个字符串2 1 2 3 3 3 1 1)中的多个连续转义颜色索引值之一。在这两种情况下,将通过从第一个字符串中复制最后7个索引值对第二个字符串的最后7个索引值,包括转义颜色索引值,一起进行编码。

[0043] 在框块256中,在块(例如,图1的块100)的所有索引值字符串都已被编码之后,对第一转义颜色(例如,蓝色)和第二转义颜色(例如,绿色)进行顺序编码。本领域技术人员在阅读本发明后将理解,在框块252中可以在第一个索引值字符串之前对附加字符串(例如,顶部字符串102、下一个字符串104、下一个字符串106)进行编码。另外,在框块254中,可以在第二个索引值字符串之后对附加字符串(例如,下一个字符串112、下一个字符串114和底部字符串116)进行编码。任意附加字符串中的附加转义颜色索引值将具有与其它字符串一起使用的相同转义颜色索引值(例如,图1中的值3)。另外,由附加转义颜色索引值中的一个所表示的任意转义颜色相对于第一和第二转义颜色被顺序编码。在框块258中,在比特流中将索引值字符串和转义颜色传输给解码装置。

[0044] 图2C是转义颜色编码方法270的一实施例的流程图。该方法270可以在例如比特流已被解码装置接收且需要对转义颜色进行解码以便显示例如屏幕内容时实现。在框块272中,接收比特流的解码器扫描块中的索引值字符串。一个或多个索引值字符串包含表示转义颜色(例如,蓝色、红色、绿色等)的相同转义颜色索引值(例如,值3)。在框块274中,解码器顺序扫描在块中的索引值字符串之后被编码的转义颜色,以确定与一个或多个索引值字符串中的每个相同转义颜色索引值相对应的转义颜色。此后,例如可以为用户显示与比特流中的信息相对应的屏幕内容。

[0045] 此处所描述的用于改进的转义编码的语法的示例如表1所示:

[0046] 表1——改进的转义编码的语法示例

[0047]

palette_escape_val_present_flag	ae(v)
--	-------

[0048]

if(palette_escape_val_present_flag)	
indexMax = palette_size	
else	
indexMax = palette_size - 1	
scanPos = 0	
while(scanPos < nCbS * nCbS) {	
xC = x0 + travScan[scanPos][0]	
yC = y0 + travScan[scanPos][1]	
if(scanPos > 0) {	
xC_prev = x0 + travScan[scanPos - 1][0]	
yC_prev = y0 + travScan[scanPos - 1][1]	
}	
if(scanPos >= nCbS && palette_mode[xC_prev][yC_prev] != COPY_ABOVE)	
palette_mode[xC][yC]	ae(v)
if(palette_mode[xC][yC] != COPY_ABOVE) {	
adjustedIndexMax = indexMax	
adjustedRefIndex = indexMax + 1	
}	
if(scanPos > 0) {	
if(palette_mode[xC_prev][yC_prev] == INDEX) {	
adjustedIndexMax--	
if(palette_index[xC_prev][yC_prev] == palette_size)	
adjustedRefIndex = adjustedIndexMax	
else	
adjustedRefIndex =	=
paletteMap[xC_prev][yC_prev]	
}	

[0049]

if(scanPos>=nCbS&&	
(palette_mode[xC_prev][yC_prev]==	COPY_ABOVE	
)) {		
adjustedIndexMax--=1		
if(palette_index [xC_prev][yC_Prev] ==		
palette_size)		
adjustedRefIndex = adjustedIndexMax		
else		
adjustedRefIndex = paletteMap[xC][yC		
- 1]		
}		
}		
if(palette_mode[xC][yC]!=COPY_ABOVE) {		
if(adjustedIndexMax > 0)		
palette_index	ae(v)	
if(palette_index>=adjustedRefIndex)		
palette_index++		
}		
palette_run	ae(v)	
runPos = 0		
runMode = palette_mode[xC][yC]		
while (runPos<=palette_run) {		
xC = x0 + travScan[scanPos][0]		
yC = y0 + travScan[scanPos][1]		
if(palette_mode[xC][yC]==INDEX) {		
paletteMap[xC][yC] = palette_index		
} else {		
paletteMap[xC][yC] = paletteMap[xC][y		
- 1]		

[0050]

}	
runPos++	
scanPos++	
}	
}	
scanPos = 0	
while(scanPos < nCbS * nCbS) {	
if(palette_index[scanPos]==palette_size) {	
for(cIdx = 0; cIdx < 3; cIdx++) {	
palette_escape_val	
paletteEscapeVal[cIdx][xC][yC] =	
palette_escape_val	
}	
scanPos++	
}	
}	
previousPaletteSize = palette_size	
current_size = palette_size	
for(i = 0; i < palette_size; i++)	
for (cIdx = 0; cIdx < 3; cIdx++)	
tempPaletteEntries[cIdx][i] = palette_entries[cIdx][i]	
for(i = 0; i < previousPaletteStuffingSize && current_size <	
max_palette_predictor_size;	
i++)	
if(previous_palette_entry_flag[i]==0) {	
for (cIdx = 0; cIdx < 3; cIdx++)	
tempPaletteEntries[cIdx][current_size] =	
previousPaletteEntries[cIdx][i]	

[0051]

<code>current_size++</code>	
<code>}</code>	
<code>previousPaletteStuffingSize = current_size</code>	
<code>previousPaletteEntries = tempPaletteEntries</code>	
<code>}</code>	

[0052] 除了上面所提到的转义编码方案之外,可以执行分组索引编码方案以改进编码。在当前HEVC SCC草案中,在CU级别对索引图进行处理。例如,对于2Nx2N CU,根据如图3所示的预定义扫描顺序,从第一个位置到最后一个位置对索引图进行处理。

[0053] 显然,吞吐量受限于要处理的位置的数量。因此,分组索引(或索引组)可以用于实现图4所示的大规模并行处理。分组方式可以实现为:

[0054] 针对所有CU尺寸的固定4x4尺寸

[0055] 针对任意2Nx2N CU的NxN尺寸

[0056] 针对任意2Nx2N CU的N/2x2N或2NxN/2尺寸

[0057] 对于每个索引组,可以重新初始化基于上下文的自适应二进制算术编码(Context-Adaptive Binary Arithmetic Coding,简称CABAC)状态,以便实现并行熵编码。

[0058] 为了进一步提高其编码效率,可以引入附加标志来指示当前索引组是相同颜色还是可以从上面的行完全复制。利用该标志,编码效率显著提高。

[0059] 除了上面所提到的分组索引编码方案之外,可以执行CU尺寸相关的颜色表尺寸方案以改进编码。使用CU相关的自适应颜色表尺寸,而不是针对所有CU尺寸应用固定表尺寸。基本上,对于较小的CU,使用较小的颜色表尺寸(对于当前表或参考表或甚二者)。这减少了对较小CU的资源需求,并且还显著改善了吞吐量,特别是考虑到小CU处理呈现整个处理流水线的瓶颈的事实。

[0060] 例如,32可以用作64x64和32x32CU的参考表尺寸和当前表尺寸,16可以用于16x16和8x8CU。另外,可以将进一步限制应用于“重用”颜色选择过程中,其中,参考颜色仅在参考表的一部分中进行评估和选择,而不是在当前工作草案中的整个参考表中。使用此方法,将参考表设置为大于当前表尺寸以更好地跟踪参考颜色的历史。另外,参考颜色仅从尺寸等于当前表的参考表的子集中选择,以提高吞吐量。对于上述示例,使用该方案,参考表尺寸增加到64,且如上所述,当前表尺寸保持不变。然而,对于64x64和32x32CU的前32个条目以及对于16x16和8x8CU的仅前16个条目,重用标志评估受限。这可以进一步提高编码吞吐量。此外,只需要与当前表的尺寸相同的数量而不是统一64位向量来用信号传递重用向量。

[0061] 除了上面所提到的CU尺寸相关的颜色表尺寸方案之外,可以执行对COPY_ABOVE_MODE的扩展搜索以改进编码。当前HEVC SCC草案中的调色板(Palette,简称PLT)模式的索引图编码具有两种模式: COPY_INDEX_MODE和COPY_ABOVE_MODE。COPY_ABOVE_MODE仅使用相邻的上一行作为参考。可以扩展COPY_ABOVE_MODE以引用当前CU中更上面的附加行。使用一元编码和截断二进制编码或任何其它二进制化方案来对当前行和参考行之间的差进行编码。搜索距离受限于当前CU。

[0062] 除了上面所提到的对COPY_ABOVE_MODE方案的扩展搜索之外,可以执行具有动态宽度和高度的二维(Two-Dimensional,简称2-D)匹配以改进编码。当前HEVC SCC草案中的PLT模式的索引图编码使用基于运行的一维(One-Dimensional,简称1-D)匹配。对于任何给定位置,当执行基于运行的1-D搜索时,也执行具有变化的宽度和高度的2-D搜索。如果由2-D搜索方法所覆盖的区域大于1-D运行值,则选择2-D字符串模式,并将其块向量(例如,bvx、bvy)以及高度和宽度编码到比特流中。否则,选择1-D匹配模式,并将其运行值编码到比特流中。

[0063] 在语法中添加1位标志2D_flag以指示当前字符串模式是2-D还是1-D。如果2D_flag=1,则对其块向量(例如,bvx、bvy)进行编码。添加1位bvx0标志以指示bvx是否等于0,添加1位bvy0标志以指示bvy是否等于0。如果相应的标志(例如,bvx0或bvy0)等于0,则添加1位bvx_sign标志以指示bvx是否大于0,并添加1位bvy_sign标志以指示bvy是否大于0。如果相应的标志(例如,bvx0或bvy0)等于0,则使用例如具有上下文的固定长度或具有上下文的截断二进制代码等对bvx或bvy的绝对值进行编码。也使用基于上下文的或非基于上下文的编码方法来对2-D宽度和高度的值进行编码。当块向量bvx和bvy都等于0,但2D_flag=1时,这意味着2-D块实际上是单色块,将其颜色索引编码到比特流中。例如,假定编码器和解码器都已知bvx、bvy、宽度和高度的最大值,则可以应用截断二进制编码方法来对它们进行编码。

[0064] 图5示出了视频编码器500的一实施例。视频编码器500可以包括如图5中所布置的速率失真优化(Rate-Distortion Optimization,简称RDO)模块510、预测模块520、变换模块530、量化模块540、熵编码器550、解量化模块560、逆变换模块570、重建模块580以及调色板创建和索引图处理模块590。在操作中,视频编码器500可接收包括视频帧(或切片)序列的输入视频。这里,帧可以指预测帧(P帧)、帧内编码帧(I帧)或双向预测帧(B帧)中的任一个。同样地,切片可以指P切片、I切片或B切片中的任一个。

[0065] RDO模块510可用于针对一个或多个其它模块协调或做出逻辑决策。例如,基于一个或多个之前已编码的帧,RDO模块510可以确定如何将正被编码的当前帧(或切片)分成多个CU,以及如何将CU分成一个或多个PU和TU。如上所述,CU、PU和TU是在HEVC中使用的各种类型的块。另外,RDO模块510可以确定如何去预测当前帧。可以通过帧间和/或帧内预测来预测当前帧。对于帧内预测,在HEVC中有多个可用预测模式或方向(例如,针对Y分量的34个模式和针对U或V分量的六个模式(包括线性模式(Linear Mode,简称LM))),而最佳模式可以由RDO模块510确定。例如,RDO模块510可以针对每个预测模式计算绝对误差的和(Sum of Absolute Error,简称SAE),并选择SAE最小的预测模式。

[0066] 在一实施例中,预测模块520用于从输入视频中生成当前块的预测块。预测模块520可以采用用于帧间预测的参考帧或用于帧内预测的当前帧中的参考像素。预测块包括多个预测像素样本,其中每个可基于位于对应重建亮度块中的多个重建亮度样本和位于对应重建色度块中的多个重建色度样本来生成。

[0067] 在生成当前块的预测块时,预测块可以减去当前块,或者反之亦然,以生成残余块。残余块可以反馈到变换模块530中,变换模块530可以通过诸如离散余弦变换(Discrete Cosine Transform,简称DCT)等2-D正交变换将残余样本转换成变换系数矩阵。然后,变换系数矩阵可以在反馈到熵编码器550之前由量化模块540进行量化。量化模块540可以改变

变换系数的范围并将其舍入为整数,这可以减少非零变换系数的数量。因此,可以增加压缩比。在一实施例中,熵编码器550用于实现此处公开的发明概念。

[0068] 量化的变换系数可以由熵编码器550进行扫描并编码到编码的比特流中。进一步地,为了促进块的连续编码,量化的变换系数也可以反馈到解量化模块560中以恢复变换系数的原始范围。然后,逆变换模块570可以执行变换模块530的逆操作,并生成原始残余块的噪声版本。然后,可以将有损残余块反馈到重建模块580中,重建模块580可以生成用于将来的块的帧内预测的重建样本。如果需要,可以在重建样本用于帧内预测之前对其进行滤波。在一实施例中,图5的编码器500和/或调色板创建和索引图处理模块590用于实现图2A和图2B的方法200。

[0069] 应当注意,图5可为视频编码器的简化说明,因此其可仅包括存在于视频编码器中的模块的一部分。尽管在图5中未示出,也可以包括其它模块(例如,滤波器、扫描仪和发射器)以方便于本领域技术人员所理解的视频编码。此外,取决于该编码方案,可以略过视频编码器中的一些模块。例如,在某些视频内容的无损编码中,可能不允许信息丢失,因此可以略过量化模块540和解量化模块560。再如,如果残余块是被直接编码而不是转换为变换系数,则可以略过变换模块530和逆变换模块570。此外,在从编码器传输之前,编码的比特流可以用于包括其它信息,例如视频分辨率、帧速率、块分割信息(尺寸、坐标)、预测模式等,使得编码的视频帧序列可以由视频解码器恰当地进行解码。

[0070] 图6示出了视频解码器600的一实施例。视频解码器600可与图5的视频编码器500相对应,且可包括如图6中所布置的熵解码器610、解量化模块620、逆变换模块630、预测模块640、重建模块650以及调色板恢复和索引图解码模块690。在操作中,包含视频帧序列的信息的编码的比特流可以由熵解码器610接收,熵解码器610可将比特流解码为未经压缩的格式。可以生成量化的变换系数的矩阵,其随后可反馈到解量化模块620中,解量化模块620可与图5中的解量化模块560相同或类似。然后,解量化模块620的输出可反馈到逆变换模块630,逆变换模块630可将变换系数转换为残余块的残余值。另外,包含当前块的预测模式的信息也可以由熵解码器610进行解码。预测模块640可基于此处公开的发明概念生成当前块的预测块。在一实施例中,熵解码器610和/或调色板恢复和索引图解码模块690用于实现此处公开的发明概念。

[0071] 图7示出了网络单元700的一实施例,网络单元700可以包括例如在网络或系统内处理如上所描述的视频帧的编码器(例如,图5的编码器500)和解码器(例如,图6的解码器600)。网络单元700可以包括用于从其它网络单元或组件、逻辑单元或处理器720中接收数据以处理数据并确定向哪个网络单元发送数据的多个入端口710和/或接收器单元(Rx)712,以及用于向其它网络单元传输数据的多个出端口730和/或发射器单元(Tx)732。所述逻辑单元或处理器720可以用于实现此处所描述的任何方案,例如使用转义作为如上所描述的匹配概念的编码及解码和/或图2A-2C的方法。可以使用硬件、软件或二者实现逻辑单元720。

[0072] 以上所描述的方案可在任意一个通用网络组件中实现,例如具有足够的处理能力、存储资源和网络吞吐能力以处理置于其上的必要工作负载的计算机或网络组件。图8示出了适于实现此处公开的诸如图2A-2C的编码和解码方法200、250和270等方法的一个或多个实施例的典型的通用网络组件或计算机系统800的示意图。通用网络组件或计算机系统

800包括处理器802(其可被称为中央处理器或CPU),处理器802与存储设备通信,存储设备包括辅助存储器804、只读存储器(Read Only Memory,简称ROM)806、随机存取存储器(Random Access Memory,简称RAM)808、输入/输出(input/output,简称I/O)设备810和网络连通设备812。虽然示出的是单个处理器,但处理器802不限于此,并且可包括多个处理器。处理器802可用作一个或多个CPU芯片、核芯(例如,多核处理器)、现场可编程门阵列(Field-Programmable Gate Array,简称FPGA)、专用集成电路(Application Specific Integrated Circuit,简称ASIC)和/或数字信号处理器(Digital Signal Processor,简称DSP)予以实现,和/或可以为一个或多个ASIC的一部分。处理器802可以用于实现此处所描述的任何方案,例如使用转义作为匹配概念的编码及解码和/或图2A-2C的方法。可以使用硬件、软件或这二者实现处理器802。

[0073] 辅助存储器804通常包括一个或多个磁盘驱动器或磁带驱动器,用于数据的非易失性存储,而且如果RAM 808的容量不足以存储所有工作数据,辅助存储器则用作溢出数据存储设备。辅助存储器804可以用于存储程序,当选择执行这些程序时,所述程序将加载到RAM 808中。ROM 806用于存储在程序执行期间读取的指令以及可能读取的数据。ROM 806为非易失性存储设备,其存储容量相对于辅助存储器804的较大存储容量而言通常较小。RAM 808用于存储易失性数据,还可能用于存储指令。访问ROM 806和RAM 808通常要比访问辅助存储器804的速度更快。此处公开的一个或多个存储设备(例如,RAM 808等)可以存储当其由逻辑单元720和/或处理器802执行时实现图2A-2C的方法200、250和270的软件、编程和/或指令。

[0074] 术语网络“元件”、“节点”、“组件”、“模块”和/或类似术语可互换地用于一般性地描述网络设备,且不具有特定或特殊意义,除非在本发明内另有具体说明和/或要求。

[0075] 虽然本发明提供了多个具体实施例,但应当理解,所公开的系统和方法也可通过其它多种具体形式体现,而不会脱离本发明的精神或范围。本发明的实例应被视为说明性而非限制性的,且本发明并不限于本文本所给出的细节。例如,各种元件或组件可以在另一系统中组合或集成,或者某些特征可以省略或不实施。

[0076] 此外,在不脱离本发明的范围的情况下,各种实施例中描述和说明为离散或单独的技术、系统、子系统和方法可以与其它系统、模块、技术或方法进行组合或集成。展示或论述为彼此耦合或直接耦合或通信的其它项也可以采用电方式、机械方式或其它方式通过某一接口、设备或中间组件间接地耦合或通信。其它变更、替换、更替示例对本领域技术人员而言是显而易见的,均不脱离此处公开的精神和范围。

100

1	1	1	1	1	1	2	2
1	1	1	1	1	1	2	2
1	1	1	1	1	1	2	2
1	3 (蓝色)	3 (绿色)	2	2	2	2	2
1	3 (绿色)	3 (蓝色)	2	2	2	2	2
2	2	3 (红色)	2	2	2	2	2
2	2	2	2	2	2	2	2
2	2	2	2	2	2	2	2

102 104 106 108 110 112 114 116

图1

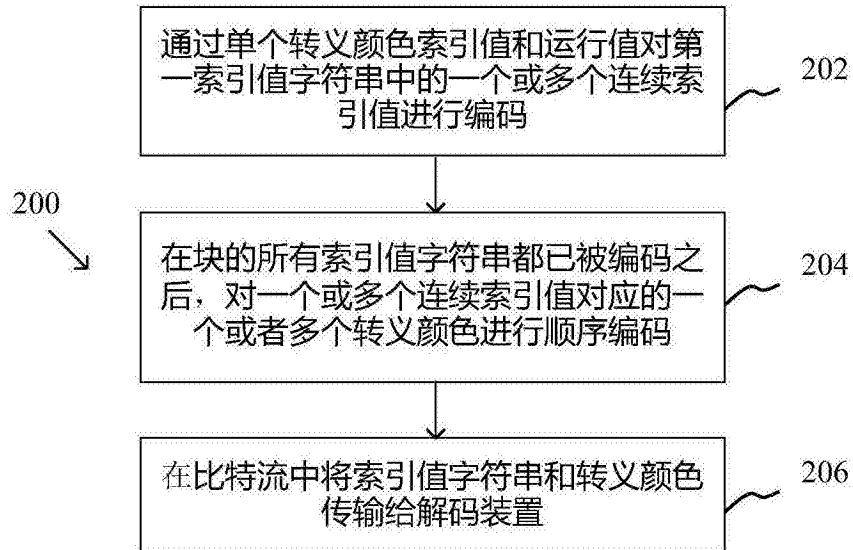


图2A

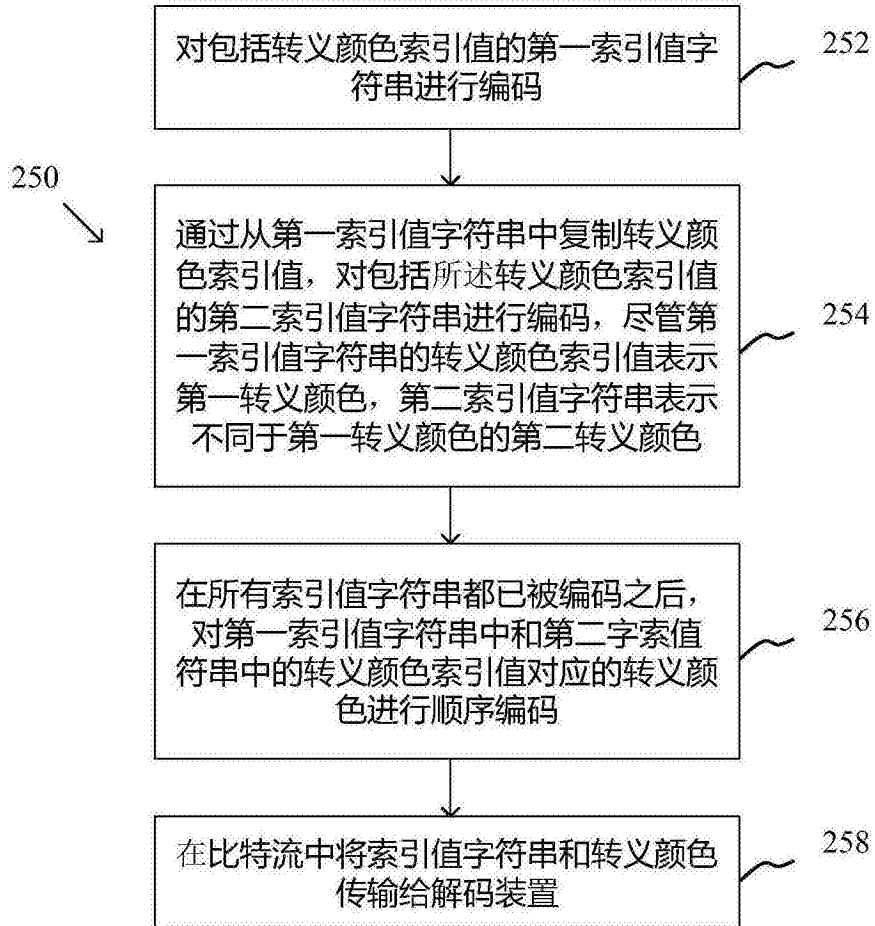


图2B

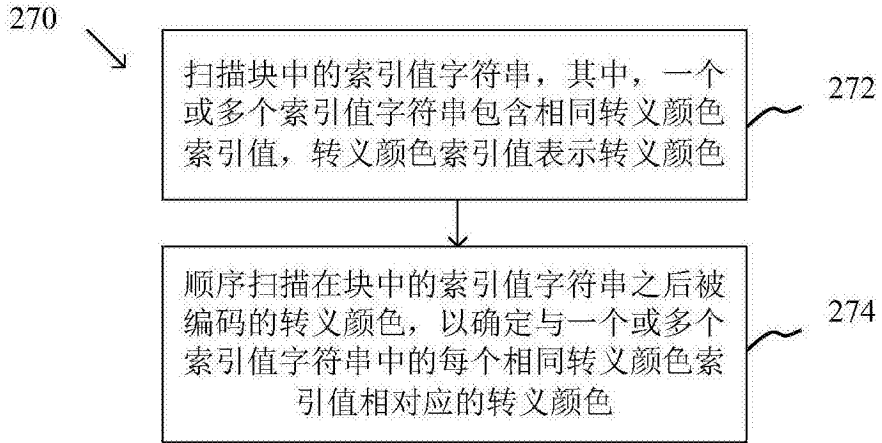


图2C

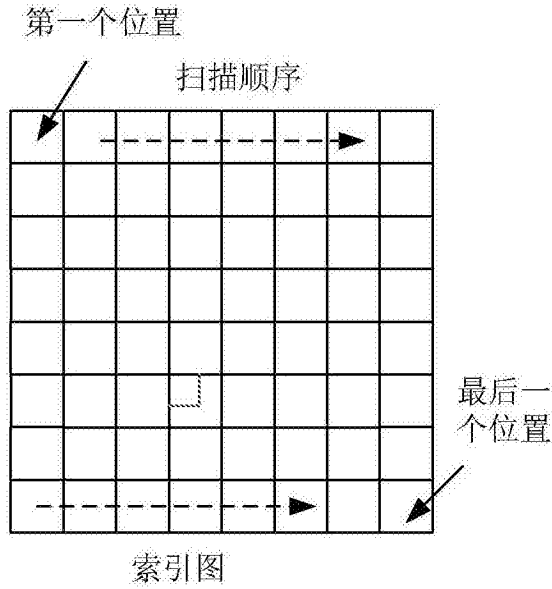


图3

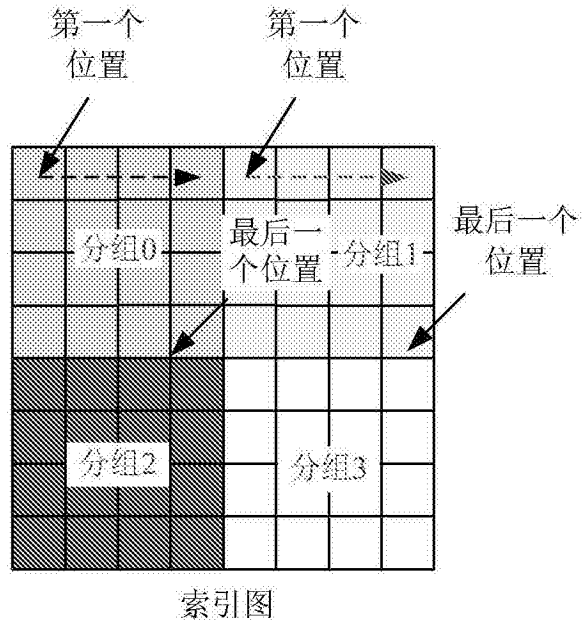


图4

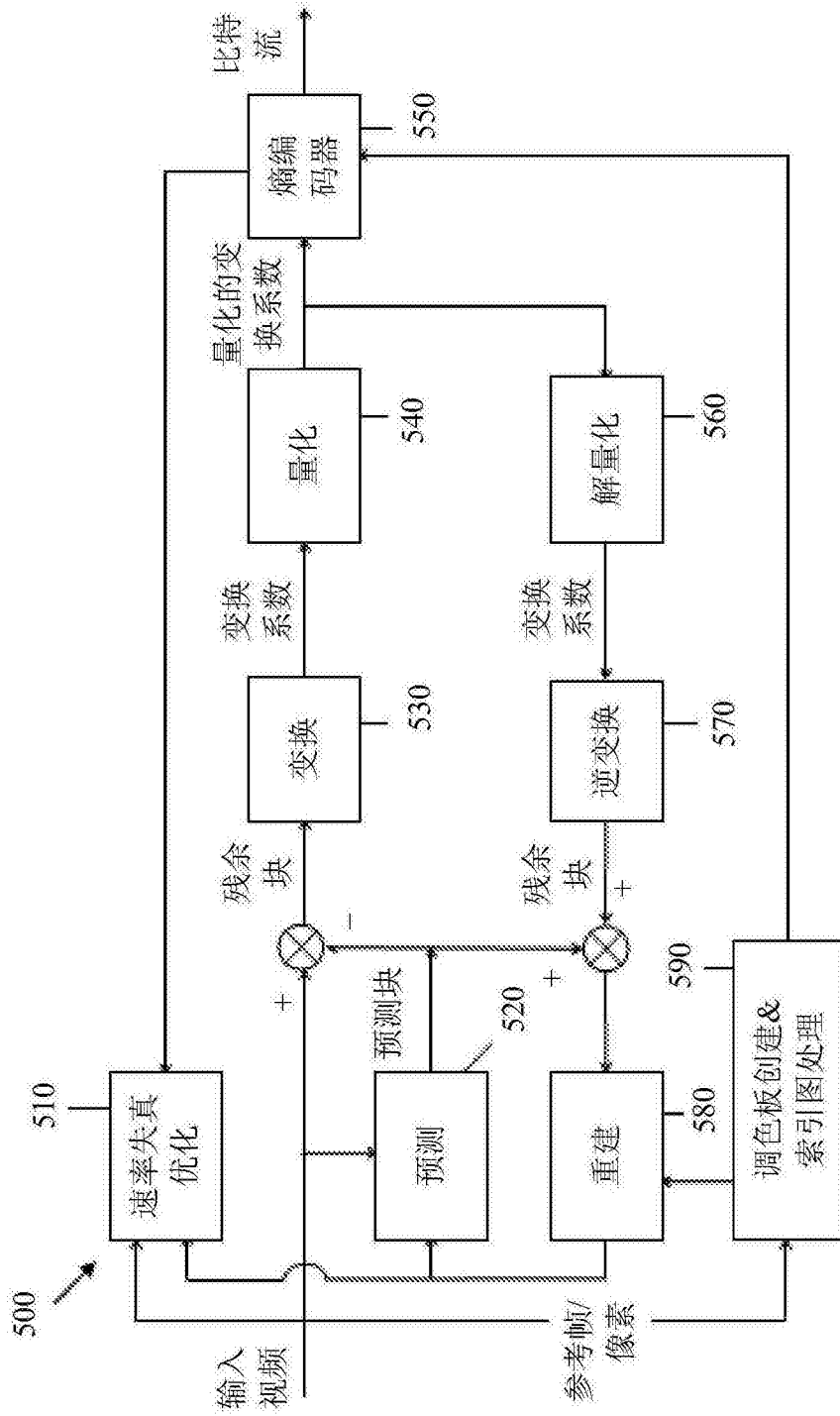


图5

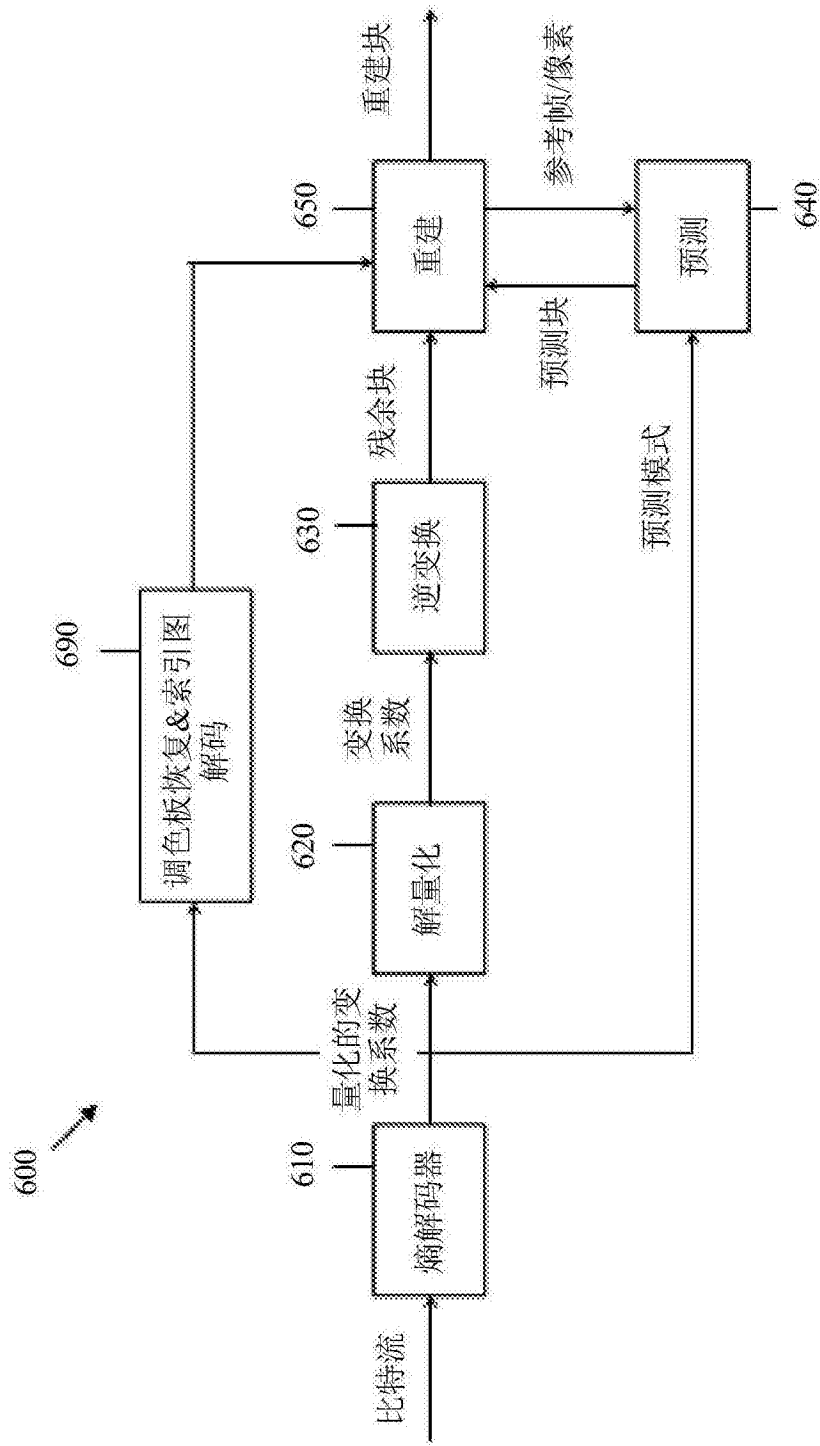


图6

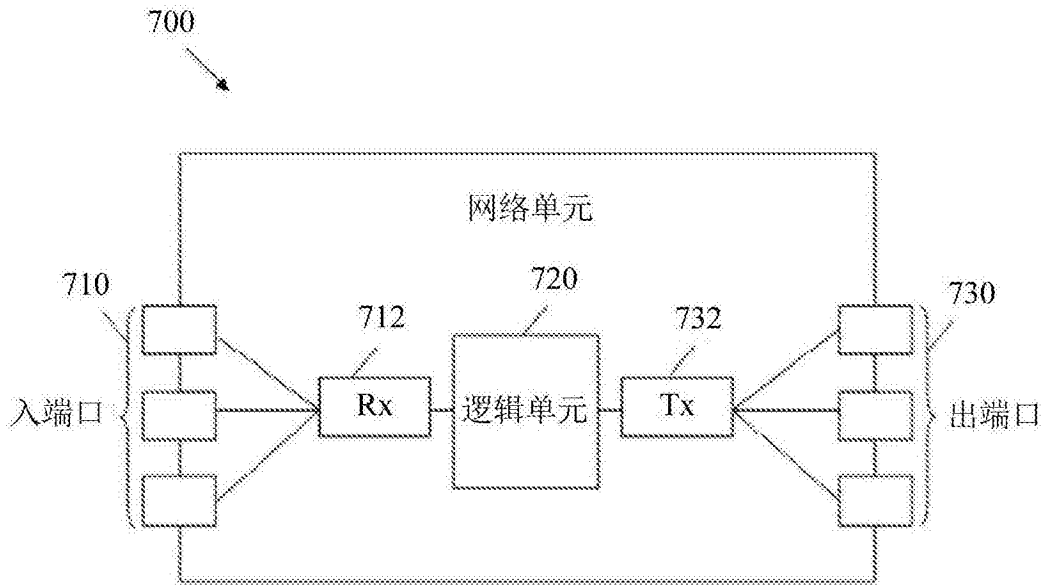


图7

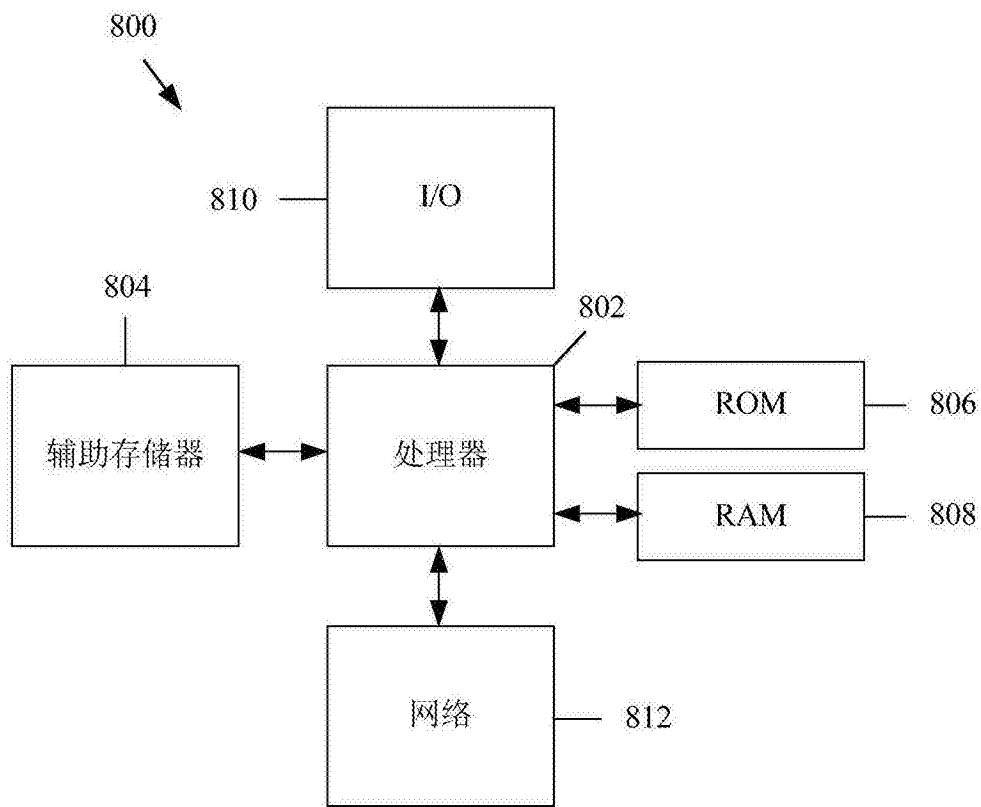


图8