



(12)发明专利

(10)授权公告号 CN 103458243 B

(45)授权公告日 2017.03.01

(21)申请号 201310369407.7

(22)申请日 2006.03.16

(65)同一申请的已公布的文献号

申请公布号 CN 103458243 A

(43)申请公布日 2013.12.18

(30)优先权数据

60/671,255 2005.04.13 US

60/700,834 2005.07.20 US

(62)分案原申请数据

200680011968.8 2006.03.16

(73)专利权人 汤姆逊许可公司

地址 法国伊西莱穆利诺

(72)发明人 于浩平

(74)专利代理机构 北京市柳沈律师事务所

11105

代理人 叶齐峰

(51)Int.Cl.

H04N 19/105(2014.01)

H04N 19/176(2014.01)

H04N 19/70(2014.01)

H04N 19/46(2014.01)

H04N 19/61(2014.01)

H04N 19/103(2014.01)

H04N 19/186(2014.01)

H04N 19/157(2014.01)

H04N 19/85(2014.01)

(56)对比文件

US 5557332 A,1996.09.17,

CN 1095286 C,2002.11.27,

CN 1112810 C,2003.06.25,

Gary Sullivan et al.Advanced Video Coding Amendment 1 Fidelity Range Extensions.《Joint Video Team(JVT)of ISO/IEC MPEG&ITU-T VCEG》.2004,第1-115页.

Woo-Shik Kim.Proposal for the unsolved issues in Professional Extensions II.《Joint Video Team(JVT)or ISO/IEC MPEG &ITU-T VCEG)》.2003,第1-11页.

审查员 韩盼

权利要求书1页 说明书14页 附图18页

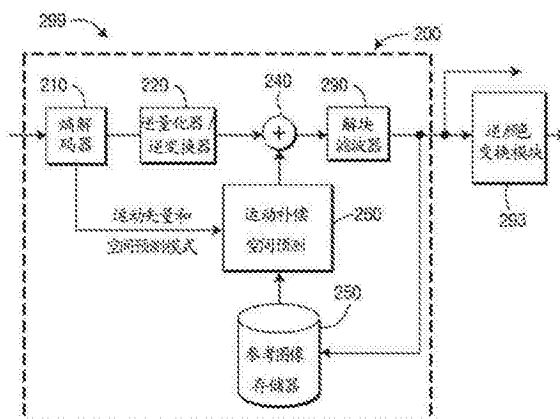
(54)发明名称

视频解码方法和设备

(57)摘要

提供了视频编码器、视频解码器和对应方法。一种用于对图像块的视频信号数据进行编码的视频编码器包括编码器(100),该编码器(100)用于利用公共预测器(315)对视频信号数据的所有颜色分量进行编码。一种用于对图像块的视频信号数据进行解码的视频解码器包括解码器(200),该解码器(200)用于利用公共预测器(430)对视频信号数据的所有颜色分量进行解码。另外,用于对图像块的信号数据进行编码和解码的设备和方法包括编码器和解码器,该编码器和解码器用于对视频信号数据的颜色分量进行编码/解码而不对它们应用残差颜色转换。而且,用于对图像块的视频信号数据进行编码/解

码的视频编码器和解码器包括编码器和解码器,该编码器和解码器用于利用对于视频信号数据每个颜色分量唯一的预测器,来对视频信号数据进行编码/解码。



1. 一种用于对图像块的视频信号数据进行解码的视频解码器, 所述视频解码器包括利用公共预测器对所述视频信号数据的所有颜色分量进行解码的解码器, 其中, 所述解码器将公共插值滤波器用于所述视频信号数据的所有颜色分量的B和P帧, 并且在不对所述视频信号数据的所有颜色分量应用残差颜色转换的情况下对所述视频信号数据的所有颜色分量进行解码。

2. 根据权利要求1所述的视频解码器, 其中所述公共预测器是用于所述视频信号数据的亮度分量和色度分量的亮度预测器。

3. 根据权利要求1所述的视频解码器, 其中所述解码器将公共空间预测模式用于所述视频信号数据的所有颜色分量。

4. 根据权利要求3所述的视频解码器, 其中通过国际电信联盟电信部门H.264标准的prev_intra8×8_pred_mode_flag、rem_intra8×8_pred_mode、prev_intra4×4_pred_mode_flag和rem_intra4×4_pred_mode参数, 来设置所述公共空间预测模式。

5. 根据权利要求1所述的视频解码器, 其中所述视频信号数据的采样对应于国际电信联盟电信部门H.264标准的4:4:4、4:2:2和4:2:0格式中的任何格式。

6. 一种用于对图像块的视频信号数据进行解码的方法, 所述方法包括利用公共预测器对所述视频信号数据的所有颜色分量进行解码的步骤, 其中, 使用公共插值滤波器用于所述视频信号数据的所有颜色分量的B和P帧, 并且其中所述解码的步骤在不对所述视频信号数据的所有颜色分量应用残差颜色转换的情况下对所述视频信号数据的所有颜色分量进行解码。

7. 根据权利要求6所述的方法, 其中所述公共预测器是用于所述视频信号数据的亮度分量和色度分量的亮度预测器。

8. 根据权利要求6所述的方法, 其中将公共空间预测模式用于所述视频信号数据的所有颜色分量。

9. 根据权利要求8所述的方法, 其中通过国际电信联盟电信部门H.264标准的prev_intra8×8_pred_mode_flag、rem_intra8×8_pred_mode、prev_intra4×4_pred_mode_flag和rem_intra4×4_pred_mode参数, 来设置所述公共空间预测模式。

10. 根据权利要求6所述的方法, 还包括在解码步骤之后的后处理步骤中对所述视频信号数据执行逆颜色变换。

11. 根据权利要求6所述的方法, 其中所述视频信号数据的采样对应于国际电信联盟电信部门H.264标准的4:4:4、4:2:2和4:2:0格式中的任何格式。

视频解码方法和设备

[0001] 本申请是申请日为2006年3月16日、申请号为200680011968.8、发明名称为“视频解码方法和设备”的发明专利申请的分案申请。

[0002] 交叉引用相关申请

[0003] 本申请要求2005年4月13日提交的美国临时申请序号60/671,255和2005年7月20日提交的美国临时申请序号60/700,834的优先权,它们在此都被整个引入作为参考。而且,本申请涉及题为“METHOD AND APPARATUS FOR VIDEO ENCODING(视频编码方法和设备)”的美国专利申请律师签号(attorney docket no.)PU050051、PU060023;题为“METHOD AND APPARATUS FOR VIDEO DECODING(视频解码方法和设备)”的美国专利申请律师签号PU060030;以及题为“METHOD AND APPARATUS FOR VIDEO ENCODING AND DECODING(视频编码与解码方法和设备)”的美国专利申请律师签号PU050159,它们每一个都同此同时提交。

技术领域

[0004] 本发明一般涉及视频编码器和解码器,并且尤其涉及视频编码与解码方法和设备。

背景技术

[0005] 目前,国际电信联盟电信标准化部门(ITU-T)H.264标准(以下称为“H.264标准”)的4:4:4格式只将三个通道之一编码为亮度,而利用较低效工具将另外两个通道编码为色度。当编解码器的输入是每一个输入分量都具有全分辨率的4:4:4格式时,利用较低效色度编码算法对三个输入分量中的两个分量进行编码将导致在这两个通道中使用更多的位。该特殊问题在帧内帧中更为显著。例如,对于40dB(PSNR)及以上的整体压缩质量,以Intra-Only(仅仅帧内)模式运行的H.264标准比JPEG2k效率低。

[0006] 因此,所希望、且非常有利的是获得克服上述现有技术缺点的视频编码与解码方法和设备。

发明内容

[0007] 本发明致力于解决现有技术的这些及其它缺点和不利条件,其涉及视频编码和解码方法及设备。

[0008] 根据本发明一方面,提供一种用于对图像块的视频信号数据进行编码的视频编码器。该视频编码器包括编码器,该编码器用于利用公共预测器对视频信号数据的所有颜色分量进行编码。

[0009] 根据本发明另一方面,提供一种用于对图像块的视频信号数据进行编码的方法。该方法包括,利用公共预测器对视频信号数据的所有颜色分量进行编码。

[0010] 根据本发明又一方面,提供一种用于对图像块的视频信号数据进行解码的视频解码器。该视频解码器包括解码器,该解码器用于利用公共预测器对视频信号数据的所有颜色分量进行解码。

[0011] 根据本发明又一方面,提供一种用于对图像块的视频信号数据进行解码的方法。该方法包括利用公共预测器对视频信号数据的所有颜色分量进行解码。

[0012] 以下将要结合附图对示范性实施例进行详细描述,本发明的这些及其它方面、特征和有益效果将变得很清楚。

附图说明

[0013] 根据以下示例图可以更好地理解本发明,其中:

[0014] 图1是示出可以应用本发明原理的示范性视频编码设备的框图;

[0015] 图2是示出可以应用本发明原理的示范性视频解码设备的框图;

[0016] 图3是示出根据本发明原理具有预编码颜色变换块的示范性视频编码过程的流程图;

[0017] 图4是示出根据本发明原理具有后解码逆颜色变换块的示范性视频解码过程的流程图;

[0018] 图5是示出残差颜色转换(RCT)简化模型的框图;

[0019] 图6A和6B是根据本发明原理的、关于ATV intra-only的平均PSNR对位速率的曲线;

[0020] 图7A和7B是根据本发明原理的、关于CT intra-only的平均PSNR对位速率的曲线;

[0021] 图8A和8B是根据本发明原理的、关于DT intra-only的平均PSNR对位速率的曲线;

[0022] 图9A和9B是根据本发明原理的、关于MIR_HD intra-only的平均PSNR对位速率的曲线;

[0023] 图10A和10B是根据本发明原理的、关于RT intra-only的平均PSNR对位速率的曲线;

[0024] 图11A和11B是根据本发明原理的、关于STB_HD intra-only的平均PSNR对位速率的曲线;

[0025] 图12是示出根据本发明原理的H.264序列参数句法的表;

[0026] 图13A-13D是示出根据本发明原理的H.264残差数据句法的表;

[0027] 图14是示出根据本发明原理具有预编码颜色变换块的示范性视频编码过程的流程图;

[0028] 图15是示出根据本发明原理具有后解码逆颜色变换步骤块的示范性视频解码过程的流程图;以及

[0029] 图16A-16B是示出根据本发明原理的H.264宏块预测句法的表。

具体实施方式

[0030] 本发明针对用于对视频信号数据进行视频编码和解码的方法和设备。应该理解,虽然主要对于利用ITU-T H.264标准的4:4:4格式采样的视频信号数据来描述本发明,但是在保持本发明范围的同时,本发明也可应用于利用H.264标准的其它格式(例如4:2:2和/或4:2:0格式)以及其它视频压缩标准采样的视频信号数据。

[0031] 应该理解,根据本发明原理的方法和设备不需要为亮度或色度压缩算法使用任何新工具。而是可以使用现有亮度编码工具。因此,由此的一个有利结果是,可以在保持向后

兼容性、并使现有H.264标准(或其它适用标准)的任何变化最小化的同时,使4:4:4格式的编码性能最大化。

[0032] 根据如实施例中所配置的本发明原理,利用亮度编码算法来对例如4:4:4内容的所有三个分量通道进行编码。该实施例的优点包括,相对于现有技术提高了4:4:4内容压缩的整体编码性能。目前,在现有H.264标准中,只将三个通道之一编码为亮度,而利用较低效工具将另外两个通道编码为色度。

[0033] 进一步,根据如实施例中所配置的本发明原理,执行颜色变换作为预处理步骤。因而,根据该实施例,在压缩循环中不执行残差颜色转换(RCT)。该实施例的优点包括,在所有颜色格式之间提供一致的编码器/解码器结构。

[0034] 而且,根据如实施例中所配置的本发明原理,将相同的运动/空间预测模式用于所有三个分量。该实施例的优点包括减小的编解码器复杂性以及向后兼容性。

[0035] 而且,根据另一实施例,不是为所有三个分量使用相同预测器,而是可以为三个分量使用3个约束空间预测器的集合(或子集)。该实施例的优点包括,相对于现有技术提高了4:4:4内容压缩的整体编码性能。

[0036] 应该理解,可以将以上及随后在此描述的各实施例实施为单独实施例,或者可以按照如该领域和相关领域的普通技术人员所容易理解的任何方式进行组合。因而,例如在第一组合实施例中,有利地利用亮度编码算法对所有三个分量通道进行编码,执行颜色变换作为预处理步骤,并将单个预测器用于所有三个分量通道。在第二组合实施例中,有利地利用亮度编码算法对所有三个分量通道进行编码,执行颜色变换作为预处理步骤,并可以将3个约束空间预测器的集合(或子集)用于三个分量通道。当然,如上所述,给定在此提供的本发明原理的教导,在保持本发明范围的同时,也可实施各实施例的其它组合。

[0037] 本说明书举例说明本发明的原理。因而,应该理解,本领域技术人员将能够设计出虽然在此未明确描述或示出、但是能具体实施本发明原理且被包括在本发明精神和范围内的各种配置。

[0038] 在此列举的所有例子和条件性语言都用于教导目的以帮助读者理解本发明人所提出的、有助于促进本技术领域发展的本发明原理和概念,并且应该认为本发明不限于这些具体列举的例子和条件。

[0039] 而且,在此列举本发明原理、方面和实施例及本发明特定例子的所有语句,都预定包括其结构和功能等效物。另外,这种等效物预定包括当前已知的等效物以及将来开发的等效物,即与结构无关的执行相同功能的任何被开发元件。

[0040] 因而,例如本领域技术人员应该理解,在此给出的框图代表具体实施本发明原理的说明性电路的概念图。类似应该理解,任何流程表、流程图、状态转移图、伪码等,都代表实质上可以在计算机可读介质中表示、从而可通过计算机或处理器来执行的各种过程,而不管这种计算机或处理器是否明确示出。

[0041] 利用专用硬件以及能够与适当软件关联地执行软件的硬件,可以提供图中所示的各种元件的功能。当通过处理器来提供这些功能时,可以通过单个专用处理器、或通过单个共享处理器、或通过其中一些可以被共享的多个单独处理器,来提供这些功能。而且,术语“处理器”或“控制器”的显式使用不应被认为只指能够执行软件的硬件,而是可以隐含地包括但不限于数字信号处理器(DSP)硬件、用于存储软件的只读存储器(ROM)、随机存储器

(RAM)和非易失性存储器。

[0042] 也可以包括其它常规和/或定制的硬件。类似,图中所示的任何开关都只是概念性的。如从上下文更具体地理解的,可以通过程序逻辑操作、通过专用逻辑、通过程序控制和专用逻辑的交互、乃至手动操作,来实现它们的功能,可以由实施者来选择特殊技术。

[0043] 在本发明权利要求中,被表示为用于执行指定功能的装置的任何元件都预定包括执行该功能的任何方式,例如包括:a)执行该功能的电路元件的组合;或b)任何形式的软件,因而包括和用于执行该软件的适当电路相结合、以执行该功能的固件、微码等。如这些权利要求所定义的本发明在于,以权利要求所要求的方式来组合并集合由列举的各种装置所提供的功能性。因而,认为能够提供这些功能性的任何装置都等效于在此所示出的那些装置。

[0044] 转到图1,一般用附图标记199来表示示范性视频编码设备。视频编码设备199包括视频编码器100和预编码颜色变换模块105。

[0045] 预编码颜色变换模块105用于,在视频信号输入到视频编码器100之前对视频信号执行颜色预处理。以下进一步描述预编码颜色变换模块105所执行的颜色预处理。应该理解,在某些实施例中可省略预编码颜色变换模块105。

[0046] 预编码颜色变换模块105的输入和视频编码器100的输入可用作视频编码设备199的输入。

[0047] 预编码颜色变换模块105的输出以信号通信方式与视频编码器100的输入相连。

[0048] 视频编码器100的输入以信号通信方式与求和点110的非反相输入相连。求和点110的输出以信号通信方式与变换器/量化器120相连。变换器/量化器120的输出以信号通信方式与熵编码器140相连。熵编码器140的输出可用作视频编码器100的输出,并且也可用作视频编码设备199的输出。

[0049] 变换器/量化器120的输出进一步以信号通信方式与逆变换器/逆量化器150相连。逆变换器/逆量化器150的输出以信号通信方式与解块滤波器160的输入相连。解块滤波器160的输出以信号通信方式与参考图像存储器170相连。参考图像存储器170的第一输出以信号通信方式与运动和空间预测估计器180的第一输入相连。视频编码器100的输入进一步以信号通信方式与运动和空间预测估计器180的第二输入相连。运动和空间预测估计器180的输出以信号通信方式与运动和空间预测补偿器190的第一输入相连。参考图像存储器170的第二输出以信号通信方式与运动和空间预测补偿器190的第二输入相连。运动和空间预测补偿器190的输出以信号通信方式与求和点110的反相输入相连。

[0050] 转到图2,一般用附图标记299来表示示范性视频解码设备。视频解码设备299包括视频解码器200和后解码器逆颜色变换模块293。

[0051] 视频解码器200的输入可用作视频解码设备299的输入。视频解码器200的输入以信号通信方式与熵解码器210的输入相连。熵解码器210的第一输出以信号通信方式与逆量化器/逆变换器220的输入相连。逆量化器/逆变换器220的输出以信号通信方式与求和点240的第一输入相连。

[0052] 求和点240的输出以信号通信方式与解块滤波器290相连。解块滤波器290的输出以信号通信方式与参考图像存储器250相连。参考图像存储器250以信号通信方式与运动和空间预测补偿器260的第一输入相连。运动和空间预测补偿器260的输出以信号通信方式与

求和点240的第二输入相连。熵解码器210的第二输出以信号通信方式与运动和空间预测补偿器260的第二输入相连。解块滤波器290的输出可用作视频解码器200的输出,并且也可用作视频解码设备299的输出。

[0053] 而且,后解码逆颜色变换模块293的输出可用作视频解码设备299的输出。在这种情况下,视频解码器200的输出可以以信号通信方式与后解码逆颜色变换模块293的输入相连,后解码逆颜色变换模块293是关于视频解码器200的后处理模块。后解码逆颜色变换模块293的输出提供相对于视频解码器200输出的后处理逆颜色变换信号。应该理解,后解码逆颜色变换模块293的使用是可选的。

[0054] 现在描述根据本发明原理的增强4:4:4编码。首先描述的实施例是一种组合实施例,其中将亮度编码算法用于所有颜色分量,将相同空间预测模式用于所有颜色分量,且从压缩循环中省略残差颜色转换(RCT)。也提供该组合实施例的测试结果。随后,描述第二组合实施例,其中将亮度编码算法用于所有颜色分量,将约束空间预测器的集合(或子集)(而不是单一空间预测模式)用于所有颜色分量,并从压缩循环中省略RCT。因而,第一和第二组合实施例之间的不同在于,在第一组合实施例中将单一空间预测模式用于所有颜色分量,而在第二组合实施例中将约束空间预测器的集合(或子集)用于所有颜色分量。当然,如上所述,可以将在此描述的各实施例实施为单独实施例,或者可以按照如该领域和相关领域的普通技术人员所容易理解的任何方式进行组合。例如,根据如实施例中所配置的本发明原理,只使用单一空间预测模式,而不和其它实施例组合,如从压缩循环中省略RCT。应该理解,给定在此提供的本发明原理的教导,在保持本发明范围的同时,该领域和相关领域的普通技术人员将能够容易弄清本发明的实施例的这些和其它变化、实施和组合。

[0055] 转到图3,一般分别用附图标记300和301来表示具有预编码颜色变换块的示范性视频编码过程。

[0056] 应该理解,预编码颜色变换块301包括块306、308和310。而且,应该理解,预编码颜色变换块301是可选的,因而在本发明某些实施例中可以省略。

[0057] 预编码颜色变换块301包括循环限制块306,循环限制块306对图像中的每一个块开始循环,并将控制传递给功能块308。功能块308对当前图像块的视频信号数据执行颜色预处理,并将控制传递给循环限制块310。循环限制块310结束循环。而且,循环限制块310将控制传递给被包括在视频编码过程300中的循环限制块312。

[0058] 循环限制块312对图像中每一个块开始循环,并将控制传递给功能块315。功能块315利用一个用于当前图像块每个颜色分量的公共预测器,来形成当前图像块的运动补偿或空间预测,并将控制传递给功能块320。功能块320从当前图像块减去运动补偿或空间预测,以形成预测残差,并将控制传递给功能块330。功能块330对预测残差进行变换和量化,并将控制传递给功能块335。功能块335对预测残差进行逆变换和逆量化,以形成编码预测残差,并将控制传递给功能块345。功能块345将编码残差加到预测上,以形成编码图像块,并将控制传递给结束循环块350。结束循环块350结束循环,并将控制传递给结束块355。

[0059] 转到图4,一般分别用附图标记400和460来表示具有后解码逆颜色变换块的示范性视频解码过程。

[0060] 应该理解,后解码逆颜色变换块460包括块462、464、466和468。而且,应该理解,后解码逆颜色变换块460是可选的,因而在本发明某些实施例中可以省略。

[0061] 解码过程400包括循环限制块410,循环限制块410对图像中的当前块开始循环,并将控制传递给功能块415。功能块415对编码残差进行熵解码,并将控制传递给功能块420。功能块420对解码的残差进行逆变换和逆量化,以形成编码残差,并将控制传递给功能块430。功能块430将编码残差加到由应用于每个颜色分量的公共预测器所形成的预测上,以形成编码图像块,并将控制传递给循环限制块435。循环限制块435结束循环,并将控制传递给结束块440。

[0062] 在某些实施例中,循环限制块435可选地将控制传递给后解码逆颜色变换块460,尤其是后解码逆颜色变换块460中所包括的循环限制块462。循环限制块462对图像中每个块开始循环,并将控制传递给功能块464。功能块464对当前图像块的视频信号数据执行逆颜色后处理,并将控制传递给循环限制块466。循环限制块466结束循环,并将控制传递给结束块468。

[0063] 在H.2644:4:4格式中,每个分量通道都具有全分辨率。因而,根据上述第一组合实施例,对每个颜色分量都使用亮度编码算法,以获得最大整体压缩效率。因此,在该实施例中,对于帧内帧,例如可以利用2004年7月的文档N6540,ISO)/IEC1449610高级视频编码第三版(ITU-T Rec.H.264)、ISO/IEC JTC1/SC29/WG11和ITU-T SG16Q.6中的表8-2、表8-3和表8-4中所列出的那些预测模式,来压缩每种颜色分量。

[0064] 另外,在实施例中,将相同空间预测模式用于所有三个像素分量,以进一步减小编解码器的复杂性以及提高性能。例如,所有三个分量都可以使用通过宏块预测头中的亮度的prev_intra4×4_pred_mode_flag、rem_intra4×4_pred_mode、prev_intra8×8_pred_mode_flag和rem_intra8×8_pred_mode参数所设置的预测模式。因此,不需要额外的比特位和句法元素。对于B和P(预测)帧,可以通过H.264标准的8.4.2.2.1节中描述的插值方法,对所有三个通道计算小数像素位置处的参考像素。以下进一步讨论当前H.264标准的详细句法和语义变化。

[0065] 在高4:4:4档次(Hign4:4:4profile)中,将RCT加到编码器/解码器上。结果,4:4:4格式的压缩结构不同于H.264标准的所有其它档次中当前为4:2:0和4:2:2格式使用的压缩结构。这导致了某些额外的实施复杂性。而且,类似于其它任何颜色变换,YCOCG不一定提高整体压缩性能。YCOCG的效率是与内容高度相关的。因而,为提高整体压缩性能和鲁棒性,在实施例中,将颜色变换放在预测循环外,作为预处理块一部分。通过这样做,为特定压缩任务选择最佳颜色变换是操作问题,并且可以在许多选项中找到特殊输入序列的最佳的颜色变换。根据其中所有三个分量都将相同空间预测器用于帧内帧、并将相同插值滤波器用于B和P(预测或帧间编码)帧的实施例,当忽略舍入/截断误差时,对预测残差执行颜色变换等同于在编解码器外对源图像执行颜色变换。以下将进一步对此进行讨论。因而,从编码结构去除RCT块,以使编码结构在所有颜色格式之间一致。

[0066] 转到图5,一般用附图标记500来表示RCT简化模型。RCT模型500包括参考像素发生器510、求和点520和线性变换模块530。参考像素发生器510的输入被配置成接收运动/边缘信息和矢量 $[X_1], [X_2], \dots, [X_n]$ 。参考像素发生器510的输出以信号通信方式与求和点520的反相输入相连,参考像素发生器510向求和点520提供预测矢量 $[X_p]$ 。求和点520的非反相输入被配置成接收输入矢量 $[X_{in}]$ 。求和点520的输出以信号通信方式与线性变换模块530的输入相连,求和点520向线性变换模块530提供矢量 $[X_d]$ 。线性变换模块530的输出被配置成

提供矢量 $[Y_d]$ 。

[0067] 在RCT500的简化模型中,由 3×3 矩阵 $[A]$ 表示的颜色变换(线性变换)被定义如下:

$$[0068] \quad \begin{bmatrix} Y \\ u \\ v \end{bmatrix} = [A] \begin{bmatrix} R \\ G \\ B \end{bmatrix} \quad (1)$$

[0069] $[X_{in}], [X_d], [X_p], [X_1], [X_2], \dots, [X_n]$ 是代表RGB域中像素的 3×1 矢量。 $[Y_d]$ 是代表颜色变换结果的 3×1 矢量。因此

$$[0070] \quad [Y_d] = [A][X_d] = [A][X_{in}] - [A][X_p] \quad (2)$$

[0071] 因为在实施例中,根据如实施例中所配置的本发明原理,将相同的空间预测器和插值滤波器用于宏块中的所有三个分量,所以可以将参考像素 $[X_p]$ 表示如下:

$$[0072] \quad [X_p] = [X_1 X_2 X_3, \dots, X_n] \begin{bmatrix} c_1 \\ c_2 \\ c_3 \\ \vdots \\ c_n \end{bmatrix} = \begin{bmatrix} R_1 & R_2 & R_3, \dots, R_n \\ G_1 & G_2 & G_3, \dots, G_n \\ B_1 & B_2 & B_3, \dots, B_n \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \\ c_3 \\ \vdots \\ c_n \end{bmatrix}, \quad (3)$$

[0073] 其中 $n \times 1$ 矢量 $[C]$ 代表H.264标准中定义的空间预测器和插值滤波器中涉及的线性运算。在此,假定利用总共 n 个相邻像素 $[X_1], [X_2], \dots, [X_n]$ 来计算参考像素。

[0074] 将式(3)中的 $[X_p]$ 代入式(2)得到以下结果:

$$[0075] \quad [Y_d] = [A][X_{in}] - [A] \left(\begin{bmatrix} R_1 & R_2 & R_3, \dots, R_n \\ G_1 & G_2 & G_3, \dots, G_n \\ B_1 & B_2 & B_3, \dots, B_n \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \\ c_3 \\ \vdots \\ c_n \end{bmatrix} \right). \quad (4)$$

[0076] 忽略舍入/截断误差并假定在RGB或Y域选择相同预测模式,得到以下结果:

$$[0077] \quad [Y_d] = [A][X_{in}] - \left([A] \begin{bmatrix} R_1 & R_2 & R_3, \dots, R_n \\ G_1 & G_2 & G_3, \dots, G_n \\ B_1 & B_2 & B_3, \dots, B_n \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \\ c_3 \\ \vdots \\ c_n \end{bmatrix} \right) = [Y_{in}] - \begin{bmatrix} Y_1 & Y_2 & Y_3, \dots, Y_n \\ u_1 & u_2 & u_3, \dots, u_n \\ v_1 & v_2 & v_3, \dots, v_n \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \\ c_3 \\ \vdots \\ c_n \end{bmatrix}. \quad (5)$$

[0078] 因此,

$$[0079] \quad [Y_d] = [Y_{in}] - [Y_1 Y_2 Y_3 \dots Y_n] \begin{bmatrix} c_1 \\ c_2 \\ c_3 \\ \vdots \\ c_n \end{bmatrix} \quad (6)$$

[0080] 因而,式(6)清楚地显示出,把YUV用作根据如该实施例中所配置的本发明原理的编码器/解码器的输入,等同于执行RCT。

[0081] 而且,根据如实施例中所配置的本发明原理,将新的4:4:4档次(4:4:4profile)添加到H.264标准中,在此被称为“profile_idc=166的高级4:4:4档次”。可以将该新的profile_idc加在序列参数头中,并且可以将其用于宏块层头以及残差数据头中。

[0082] 为支持利用亮度算法对所有三种颜色分量进行编码,可以对残差数据句法作出某些改变。另外,也可以对宏块头、残差数据头等之中的一些元素的语义作出改变。一般,H.264规范中的现有亮度句法将保持不变,并将用于对三个分量之一进行编码。变化是向后兼容的。在此,以下描述详细的句法和语义变化。

[0083] 现在,将描述根据如各实施例中配置的本发明原理所执行的模拟结果。

[0084] 转到图6A和6B,一般分别用附图标记600和650来表示关于ATV intra-only的平均PSNR(峰值信噪比)对位速率的曲线。

[0085] 转到图7A和7B,一般分别用附图标记700和750来表示关于CT intra-only的平均PSNR对位速率的曲线。

[0086] 转到图8A和8B,一般分别用附图标记800和850来表示关于DT intra-only的平均PSNR对位速率的曲线。

[0087] 转到图9A和9B,一般分别用附图标记900和950来表示关于MIR_HD intra-only的平均PSNR对位速率的曲线。

[0088] 转到图10A和10B,一般分别用附图标记1000和1050来表示关于RT intra-only的平均PSNR对位速率的曲线。

[0089] 转到图11A和11B,一般用附图标记1100和1150来表示关于STB_HD intra-only的平均PSNR对位速率的曲线。

[0090] 尤其是,图6A、7A、8A、9A、10A和11A示出了所建议的高级4:4:4档次(用术语“新”来表示,并在它前面加上“新”)对与之相应的近似结果的测试结果。而且,图6B、7B、8B、9B、10B和11B示出了所建议的高级4:4:4档次(用术语“新”来表示,并在它前面加上“新”)对JPEK2k的测试结果。

[0091] 在图6A、6B至11A、11B的所有图中,用分贝(dB)来表示PSNR,并用每秒位数(bps)来表示位速率。ATV、CT、DT、MIR、RT、STB是测试剪辑的名称。

[0092] 在测试中,使用JVT-J042电影起源测试序列中所描述的所有JVT/FRExt测试序列。它们都是4:4:410位电影素材,并且每个剪辑有58帧。

[0093] 在JVT参考软件JM9.6中实施所建议的高级4:4:4档次。Intra-only和IBBP编码结构都用于测试中。对于每一条R-D曲线,量化参数都被设为6、12、18、24、30和42。使用RD优化模式选择。

[0094] 也将所建议的高级4:4:4档次和通过对每个单独输入分量运行YUVFormat=0(4:0:0)的参考软件所得到的结果进行比较。把三个分开的单独压缩位计数简单相加,以得到用于计算压缩位速率的总压缩位数。

[0095] 关于JPEG2k,在测试中使用KaKadu V2.2.3软件。利用具有9/7双正交小波滤波器的5级小波解压缩,来产生测试结果。每帧只有一个瓦片(tile),并且对于给定目标位速率也使用RD优化。

[0096] 所有PSNR测量都在RGB域中执行。利用被定义为 $(\text{PSNR}(\text{红})+\text{PSNR}(\text{绿})+\text{PSNR}(\text{蓝}))/3$ 的平均PSNR,来比较整体压缩质量。这主要是因为利用软件所提供的未知速率控制算法来计算JPEG2k压缩数据。对于某些情况,RGB PSNR值相互之间相差很大,尤其是当使用JPEG2k颜色变换时。

[0097] 如下执行压缩比较:

[0098] • New1:具有单一预测模式的建议高级4:4:4档次。

[0099] • New3:具有三种预测模式的建议高级4:4:4档次。

[0100] • RCT-OFF:RCT=关闭的RGB输入。

[0101] • RCT-ON:RCT=开启的RGB输入。

[0102] • YCOCG:在编解码器外面执行RGB到YCOCG转换。然后把转换的YCOCG用作JVT软件的输入。

[0103] • R+G+B:通过分别压缩R、G和B信号而近似的建议方法。

[0104] • Y+CO+CG:通过分别压缩所转换的Y、CO和CG信号而近似的建议方法。

[0105] • J2k_RGB:在RGB域执行JPEG2k压缩。关掉JPEG2k颜色变换。

[0106] • J2k_YUV:在YUV域执行JPEG2k压缩。使用JPEG2k颜色变换。

[0107] 根据测试结果,根据如实施例中所配置的本发明原理的实施在整体压缩效率方面一般非常类似于JPEG2k。在某些情况下,甚至稍好一点。

[0108] 此外,对于40dB(PSNR)以上的质量,根据如实施例中所配置的本发明原理的实施提供了比当前高4:4:4档次(High4:4:4Profile)高得多的性能(压缩)。具体地说,New1-YCOCG或New3-YCOCG好于YCOCG和RCT-ON;New1-RGB或New3-RGB好于RCT-OFF。在等于和大于45dB(PSNR)的PSNR下,平均PSNR的平均提高大于1.5dB。在最后一个例子中,可以将该提高转变为45dB PSNR下25%以上的位节省。

[0109] 根据测试结果,好像当内容具有更饱和颜色时,如TP、RT,颜色变换将有助于编码性能。即,如果颜色是中性的、且不饱和,则在RGB域的编码或许是正确选择。以上观察与使用什么颜色变换无关。

[0110] 比较New1-YCOCG或New3-YCOCG和JPEG-2k_YUV的结果,可以观察到,特定颜色变换的性能在提高编码效率方面是与内容非常相关的。没有一种颜色变换总是最好的。因此,我们的数据确认了,在编码(或解码)循环内执行颜色变换如RCT可能不是一个好主意。相反,如果必要,在编码器/解码器外执行颜色变换可能使整个压缩系统提供更好、更鲁棒的性能。

[0111] 比较YCOCG和RCT-ON,测试结果没有显示出由RCT引起的任何编码效率提高。另外,应该主意,运行RCT开启的参考软件大大增加了编码时间。运行时间长了2.5倍多。

[0112] 现在,将描述根据如实施例中所配置的本发明原理的句法和语义变化。

[0113] 转到图12,一般用附图标记1200来表示H.264序列参数句法表。用斜体字来表示根据如实施例中所配置的本发明原理的句法变化。

[0114] 转到图13,一般用附图标记1300来表示H.264残差数据句法表。用斜体字来表示根据如实施例中所配置的本发明原理的句法添加/变化。在表1300中,残差数据头中的亮度部分和某些必须文本修改一起重复两次,以分别支持luma1和luma2。

[0115] 如上所述,通过在JVT参考软件JM9.6中实施本发明原理,来评价和测试上述第一组合实施例。用New1-RGB或New1-YCOG标记的测试结果代表第一组合实施例。

[0116] 如上所述,根据如实施例中所配置的本发明原理,将3个约束空间预测器的集合(或子集)而不是单一空间预测模式,用于分量通道(例如RGB、YUV、YCrCb格式等)。而且,如上所述,可以将该实施例和在此描述的其它实施例组合起来,例如,只利用亮度编码算法来对内容的所有三个分量通道进行编码、以及/或者使用颜色变换作为预处理步骤。

[0117] 现在,将描述上述第二组合实施例,它涉及将3个约束空间预测器的集合(或子集)用于颜色分量,只使用亮度编码算法对所有三个颜色分量进行编码,以及使用颜色变换作为预处理步骤(即,在压缩循环内没有RCT)。与此同时也将描述该实施例的某些变形方案。

[0118] 转到图14,一般分别用附图标记1400和1401来表示具有预编码颜色变换步骤的示范性视频编码过程。

[0119] 应该理解,预编码颜色变换块1401包括块1406、1408和1410。而且,应该理解,预编码颜色变换块1401是可选的,因而在本发明某些实施例中可以省略。

[0120] 预编码颜色变换块1401包括循环限制块1406,循环限制块1406对图像中的每个块开始循环,并将控制传递给功能块1408。功能块1408对当前图像块的视频信号数据执行颜色预处理,并将控制传递给循环限制块1410。循环限制块1410结束循环。而且,循环限制块1410将控制传递给被包括在视频编码过程1400中的循环限制块1412。

[0121] 循环限制块1412对图像中的每个块开始循环,并将控制传递给功能块1415。功能块1415利用一个用于当前图像块的每个颜色分量的公共预测器,来形成当前图像块的运动补偿或空间预测,并将控制传递给功能块1420。功能块1420从当前图像块减去运动补偿或空间预测,以形成预测残差,并将控制传递给功能块1430。功能块1430对预测残差进行变换和量化,并将控制传递给功能块1435。功能块1435对预测残差进行逆变换和逆量化,以形成编码预测残差,并将控制传递给功能块1445。功能块1445将编码残差加到预测上,以形成编码图像块,并将控制传递给结束循环块1450。结束循环块1450结束循环,并将控制传递给结束块1455。

[0122] 转到图15,一般分别用附图标记1500和1560来表示具有后解码逆颜色变换步骤的示范性视频解码过程。

[0123] 应该理解后解码逆颜色变换块1560包括块1562、1564、1566和1568。而且,应该理解,后解码逆颜色变换块1560是可选的,因而在本发明某些实施例中可以省略。

[0124] 解码过程1500包括循环限制块1510,循环限制块1510对图像中的当前块开始循环,并将控制传递给功能块1515。功能块1515对编码残差进行熵解码,并将控制传递给功能块1520。功能块1520对解码残差进行逆变换和逆量化,以形成编码残差,并将控制传递给功能块1530。功能块1530将编码残差加到由用于每个颜色分量上的公共预测器形成的预测上,以形成编码图像块,并将控制传递给循环限制块1535。循环限制块1535结束循环,并将

控制传递给结束块1540。

[0125] 在某些实施例中,循环限制块1535可选地将控制传递给后解码逆颜色变换块1560,尤其是后解码逆颜色变换块1560中所包括的循环限制块1562。循环限制块1562对图像中的每个块开始循环,并将控制传递给功能块1564。功能块1564对当前图像块的视频信号数据执行逆颜色后处理,并将控制传递给循环限制块1566。循环限制块1566结束循环,并将控制传递给结束块1568。

[0126] 如上所述,披露了高级4:4:4档次的新档次(profile_idc=166)。该新档次也可用于第二组合实施例,对应的语义和句法变化如下所述。该新profile_idc被加在序列参数集合中,并将主要用于随后的头中,以指示输入格式为4:4:4、以及所有三个输入通道都类似地被编码为亮度。

[0127] 为使H.264标准所需的改变减到最少,不为高级4:4:4档次披露新的宏块类型。而是所有宏块类型与H.264标准的表7-11、表7-13和表7-14中列出的关联编码参数一起仍然有效。对于帧内宏块的情况,所有三个输入通道,亮度、Cr和Cb,都将基于H.264标准的表7-11中定义的MbPartPredMode来进行编码。例如,高级4:4:4档次中的Intra_4×4宏块意味着,可以利用H.264标准的表8-2中给出的所有9种可能预测模式来对每个输入分量通道进行编码。仅供参考,在当前高4:4:4档次(Hign4:4:4Profile)中,将把Intra_4×4宏块的通道中的两个处理为色度,并且将只使用H.264标准的表8-5中的4种可能帧内预测模式之一。对于B和P宏块,对高级4:4:4档次所作的变化发生在小数像素位置处的参考像素值的插值计算处理中。在此,H.264标准的8.4.2.2.1节中描述的过程,亮度样本插值处理,将适用于亮度、Cr和Cb。再次仅供参考,当前高4:4:4档次(Hign4:4:4Profile)将H.264标准的8.4.2.2.2节的色度样本插值处理用于输入通道中的两个。

[0128] 在CABAC被选为熵编码模式的情况下,将为Cr和Cb生成和当前为亮度定义的上下文模型相同的两组分开上下文模型。在编码期间,也将独立地更新它们。

[0129] 最后,在实施例中,因为在编码循环中没有RCT块,所以从高级4:4:4档次的序列参数集合中去除ResidueColorTransformFlag。

[0130] 到现在为止,如图13所示,大多数句法变化都出现在残差数据中,其中原始亮度句法被重复两次,以支持所建议的高级4:4:4档次中的Cr和Cb。

[0131] 关于H.264宏块层表(未示出),对应句法的语义变化包括以下。

[0132] coded_block_pattern(添加)。当chroma_format_idc等于3且coded_block_pattern存在时,应该将CodedBlockPatternChroma设为0。另外,对于宏块的12个8×8亮度、Cb和Cr块的每一个,CodedBlockPatternLuma都规定以下情况之一:(1)8×8亮度、8×8Cb和8×8Cr块中的12个4×4亮度块的所有变换系数电平都等于0;(2)8×8亮度、8×8Cb和8×8Cr块中的一个或多个4×4亮度块的一个或多个变换系数电平应该为非零值。

[0133] 现在,将描述根据第二组合实施例(或关于三个约束空间预测器集合(或子集)的使用的单独实施例)的帧内块空间预测模式选择。

[0134] 为了使每个分量独立选择其最佳MbPartPredMode和随后的最佳空间预测模式,和分别对每个输入通道进行编码的情况一样,可以将某些新帧内块类型加到H.264标准的表7-11中。结果,将对H.264标准作出大量改变。在关于第二组合实施例的实施例中,当前mb_types保持不变,并且提供替换解决方案。在该实施例中,限制用相同的MbPartPredMode或

宏块类型对三个输入通道进行编码。然后,将少量的新元素添加到宏块预测句法中,以支持三种不同预测模式。因此,每个分量理论上仍然能够独立选择其最佳空间预测模式,以便使每个分量通道的预测误差最小化。例如,假定将Intra_4×4宏块选为mb_type,则亮度、Cr或Cb仍然可以在H.264标准的8.3.1.1节的表8-2中找到它自己的最佳空间预测模式,例如用于亮度的Intra_4×4_Vertical、用于Cr的Intra_4×4_Horizontal以及用于Cb的Intra_4×4_Diagonal_Down_Left。

[0135] 关于上述第一组合实施例的另一种方法是,约束所有三个输入通道都共享相同预测模式。这可以利用现有句法元素当前所携带的预测信息来实现,如宏块预测句法中的prev_intra4×4_pred_mode_flag、rem_intra4×4_pred_mode、pred_intra8×8_pred_mode_flag和rem_intra8×8_pred_mode。该选项将导致H.264的少量变化以及编码效率稍微有些损失。

[0136] 基于测试结果,与第一组合实施例相比,使用三种预测模式可以将总编码性能提高大约0.2dB。

[0137] 转到图16,一般用附图标记1600来表示H.264宏块预测句法表。仅供参考,以下列出了用于支持使用三种预测模式的修改宏块预测句法,其中:

[0138] prev_intra4×4_pred_mode_flag0和rem_intra4×4_pred_mode0用于亮度;

[0139] prev_intra4×4_pred_mode_flag1和rem_intra4×4_pred_mode1用于Cr;

[0140] prev_intra4×4_pred_mode_flag2和rem_intra4×4_pred_mode2用于Cb。

[0141] 现在,将描述根据如关于第二组合实施例的实施例所配置的本发明原理而执行的模拟结果。

[0142] 使用JVT-J042,电影起源测试序列,JVT-J039(Viper)中所描述的所有JVT/FRExt测试序列。它们都是4:4:410位素材,并且每个剪辑有58帧。

[0143] 在JVT参考软件JM9.6中实施所建议的算法,并在测试中使用修改的软件。对Intra-only和IBRrBP都进行测试。在此,“Br”表示所记录的B图像。Intra-only情况适合于量化参数等于6、12、18、24、30、36和42的所有序列。由于模拟中涉及大量的时间,IBRrBP图像组(GOP)结构只适合于量化参数等于12、18、24、30和36的电影剪辑。根据4:4:4AHG中的论述,在测试中使用以下关键参数:

[0144] SymbolMode=1

[0145] RDOptimization=1

[0146] ScalingMatrixPresentFlag=0

[0147] OffsetMatrixPresentFlag=1

[0148] QoffsetMatrixFile="q_offset.cfg"

[0149] AdaptiveRounding=1

[0150] AdaptRndPeriod=1

[0151] AdaptRndChroma=1

[0152] AdaptRndWFactorX=8

[0153] SearchRange=64

[0154] UseFME=1

[0155] 关于JPEG2k,在测试中使用KaKadu V2.2.3软件。利用具有9/7双正交小波滤波器

的5级小波解压缩,来产生测试结果。每帧只有一个瓦片(tile),并且对于给定目标位速率也使用RD优化。

[0156] 主要在源内容的原始色域中计算PSNR度量,对于上述剪辑,源内容的原始色域是RGB。利用被定义为 $(\text{PSNR}(\text{红})+\text{PSNR}(\text{绿})+\text{PSNR}(\text{蓝}))/3$ 的平均PSNR,来比较整体压缩质量。

[0157] 如下执行压缩比较:

[0158] • New1:具有单一预测模式的建议高级4:4:4档次。

[0159] • New3:具有三种预测模式的建议高级4:4:4档次。

[0160] • RCT-OFF:RCT=关闭的RGB输入。

[0161] • RCT-ON:RCT=开启的RGB输入。

[0162] • YCOCG:在编解码器外执行RGB到YCOCG转换。然后把转换的YCOCG用作JVT软件的输入。

[0163] • R+G+B:通过分别压缩R、G和B信号而近似的建议方法。

[0164] • Y+CO+CG:通过分别压缩所转换的Y、CO和CG信号而近似的建议方法。

[0165] • JPEG2k_RGB:在RGB域执行JPEG2k压缩。关掉JPEG2k颜色变换。

[0166] • JPEG2k_YUV:在YUV域执行JPEG2k压缩。使用JPEG2k颜色变换。

[0167] 对于Intra-only情况,根据本发明原理的建议高级4:4:4档次在整体压缩效率方面非常类似于JPEG2k。在某些情况下,甚至更好一些。

[0168] 根据本发明原理的方法明显好于当前高4:4:4档次(Hign4:4:4Profile)。在等于和大于45dB(PSNR)的PSNR下,平均PSNR的平均提高超过1.5dB。在某些情况下,可以将该提高转换为45dB PSNR下25%以上的位节省。

[0169] 即使对于相同块类型,使用三种预测模式也稍好于单一预测模式。然而,可以使用更多句法和语义变化。

[0170] 现在,将描述本发明实施例的原理所提供的许多附带优点/特征中的一些。

[0171] 测试结果表明,与当前高4:4:4档次(Hign4:4:4Profile)相比,利用与本发明原理对应的改进的建议高级4:4:4档次提供了改善的性能。并且性能提高的很显著。另外,将颜色变换移到编解码器外面将使编解码器的结构在所有颜色格式之间一致。结果,它将使实施更容易,且降低成本。在选择最佳颜色变换以获得更好编码效率的方面,它也将使编解码器更鲁棒。而且,建议的方法不添加任何新的编码工具,而只需对句法和语义稍作改变。

[0172] 因而,根据如实施例中配置的本发明原理,提供一种视频编码和解码方法及设备。提供对现有H.264标准的修改,这些修改将性能提高到当前可获得的性能之上。而且,对于高质量应用,甚至将性能提高到超过JPEG-2000。根据如实施例中配置的本发明原理,可以通过利用亮度编码算法对4:4:4内容的所有三个颜色分量进行编码,来实现H.264标准的4:4:4编码性能重大提高。即,不需要新工具用于亮度(或没有使用的色度)压缩算法。而是使用现有亮度编码工具。进一步,可以根据本发明原理来实施当前4:4:4档次的句法和语义变化,以支持所有三个分量通道的亮度编码。在根据本发明实施例进行的测试中,当源内容有许多空间纹理和边缘时,亮度中使用的空间预测工具明显显示出它们的性能优于色度中所使用的空间预测工具。对于某些测试序列,当每个颜色分量都被编码为亮度时,在大于或等于45dB(平均PSNR)的压缩质量下可观察到超过30%的位减少。

[0173] 应该理解,虽然在此主要相对于利用H.264标准的4:4:4格式采样的视频信号数据

描述了本发明,但是也可容易相对于利用H.264标准的其它格式(如4:2:0格式和/或4:2:2格式)以及其它视频压缩标准采样的视频信号数据,来实施本发明。给定在此提供的本发明教导,在保持本发明范围的同时,该领域及相关领域普通技术人员也可容易实施本发明的这些及其它变化。

[0174] 相关领域的普通技术人员根据在此给出的教导可以容易地弄清本发明的这些及其它特征和优点。应该理解,可以用各种形式的硬件、软件、固件、专用处理器或它们的组合,来实施本发明的教导。

[0175] 更为优选地,把本发明的教导实施为硬件和软件的组合。而且,软件可为在程序存储单元上具体实施的应用程序。该应用程序可以被上载到包括任何合适架构的机器上,并通过该机器来执行。优选地,在具有硬件如一个或多个中央处理器(CPU)、RAM和输入/输出(I/O)接口的计算机平台上,实现该机器。该计算机平台也可包括操作系统和微指令码。在此描述的各种过程和功能可以是,可通过CPU执行的微指令码一部分、或应用程序一部分、或它们的任何组合。另外,其它各种外围单元如附加数据存储单元和打印单元,可以连接到该计算机平台。

[0176] 进一步应该理解,因为附图中所描述的一些组成系统部件和方法优选地是用软件来实现的,所以系统部件或过程功能块之间的实际连接可以随本发明的编程方式而有所不同。给定在此的教导,相关领域普通技术人员将能够预期本发明的这些及类似实施或配置。

[0177] 虽然在此参考附图描述了说明性实施例,但是应该理解,本发明不限于这些具体实施例,并且相关领域普通技术人员可以在不脱离本发明范围或精神的情况下实现各种变动和修改。所有这种变动和修改都预定被包括在如所附的权利要求所阐述的本发明范围内。

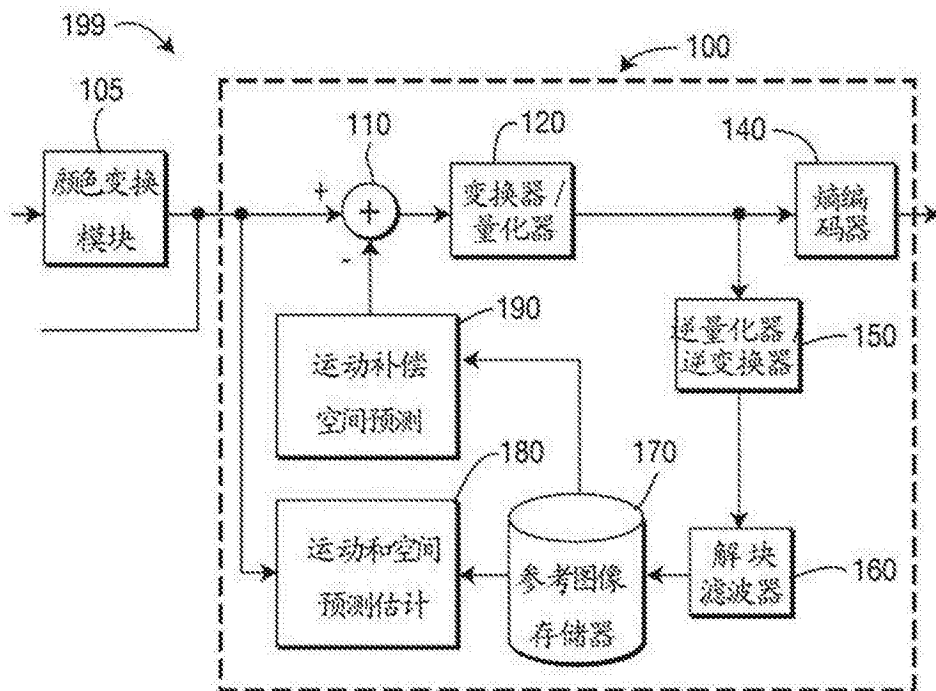


图1

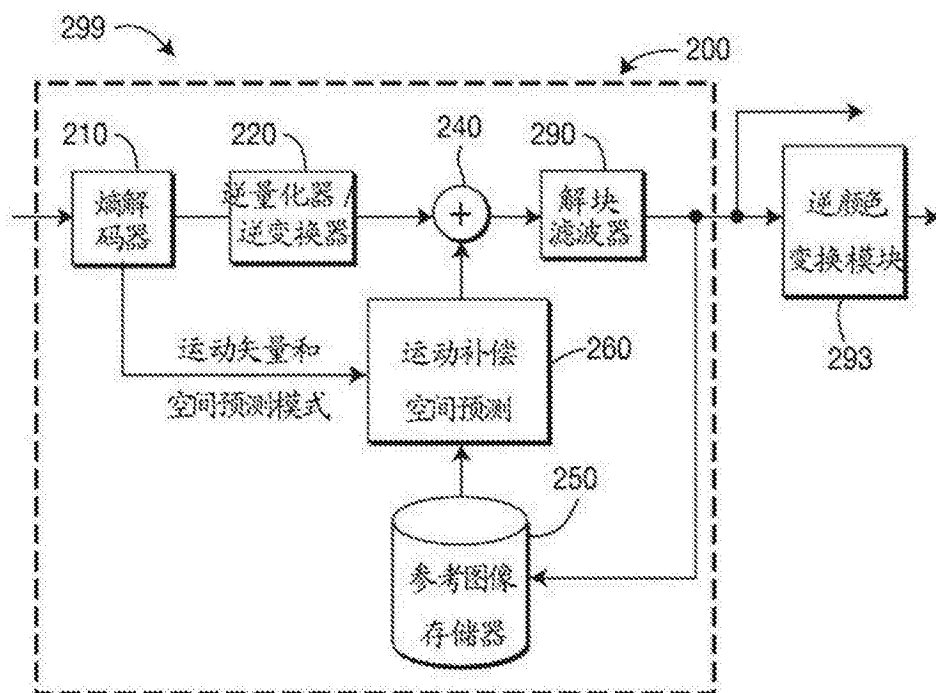


图2

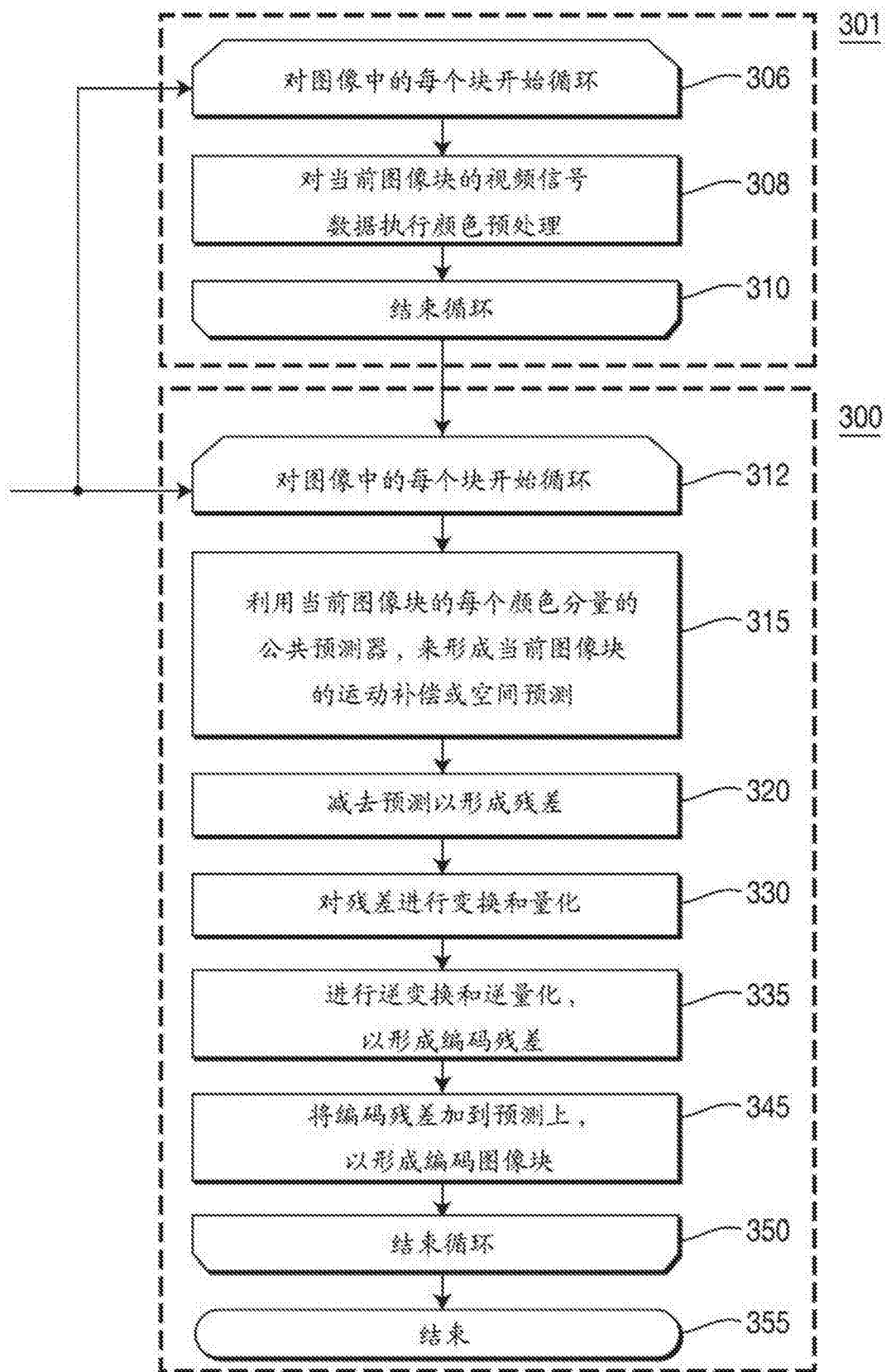


图3

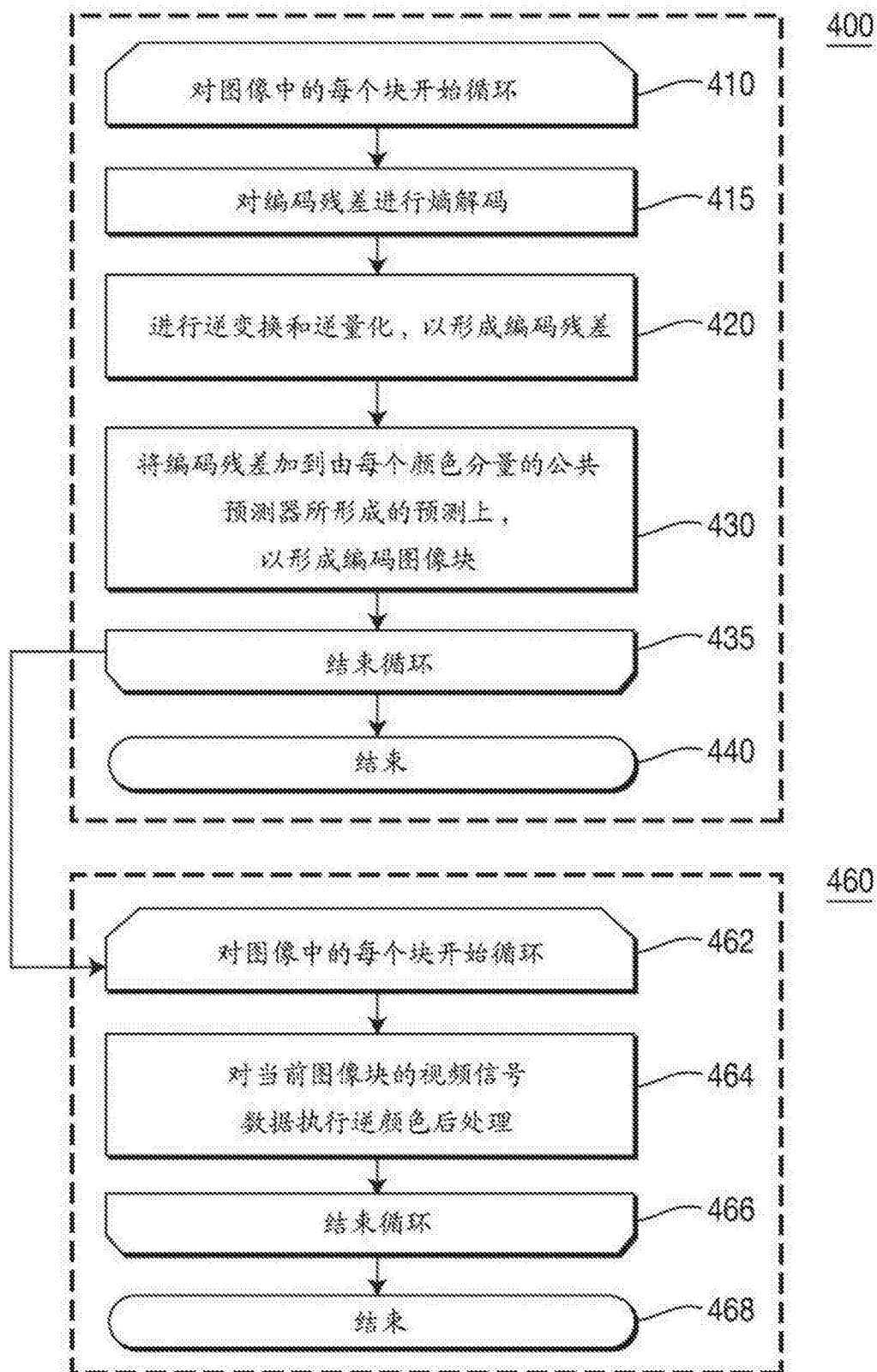


图4

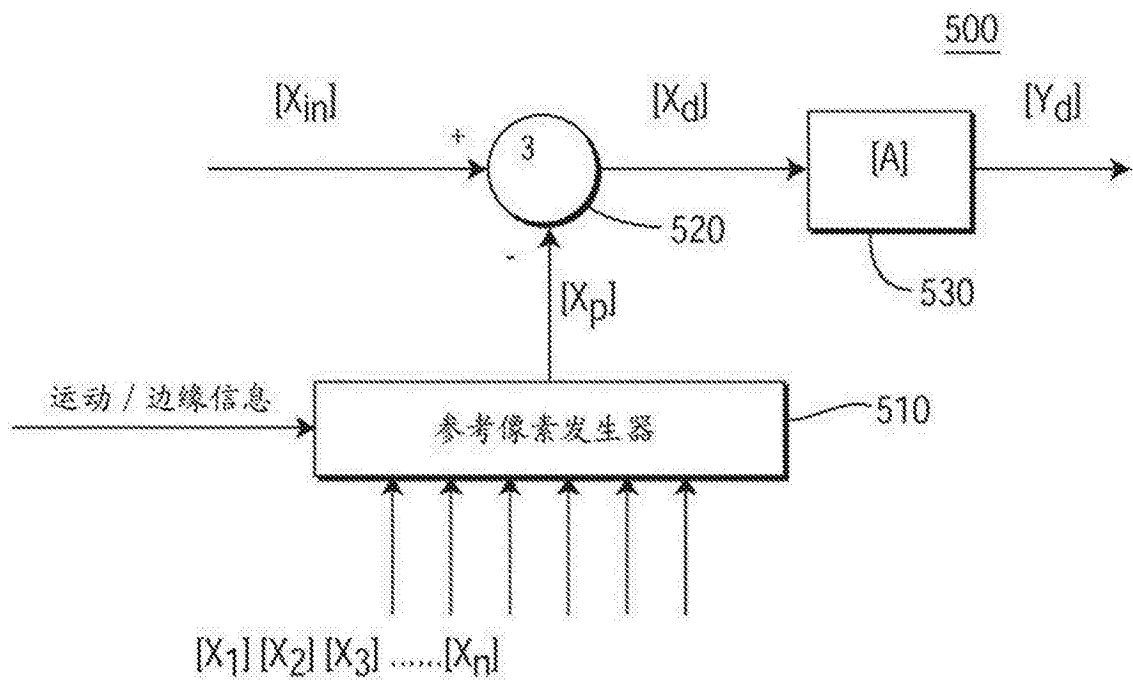


图5

1200

seq_parameter_set_rbsp(){	C	Descriptor
profile_idc	0	u(8)
constraint_set0_flag	0	u(1)
constraint_set1_flag	0	u(1)
constraint_set2_flag	0	u(1)
constraint_set3_flag	0	u(1)
reserved_zero_4bits /* equal to 0 */	0	u(4)
level_idc	0	u(8)
seq_parameter_set_id	0	ue(v)
if(profile_idc == 100 profile_idc == 110 profile_idc == 122 profile_idc == 166 profile_idc == 188){		
chroma_format_idc	0	ue(v)
bit_depth_luma_minus8	0	ue(v)

图12

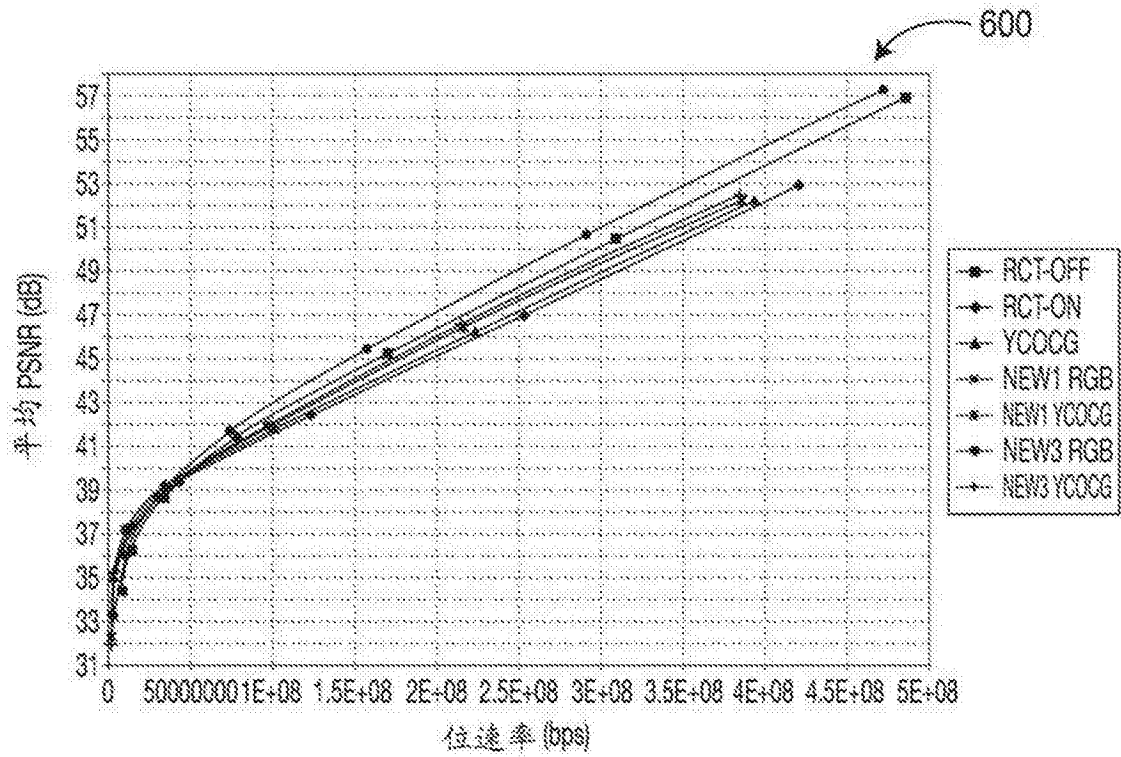


图6A

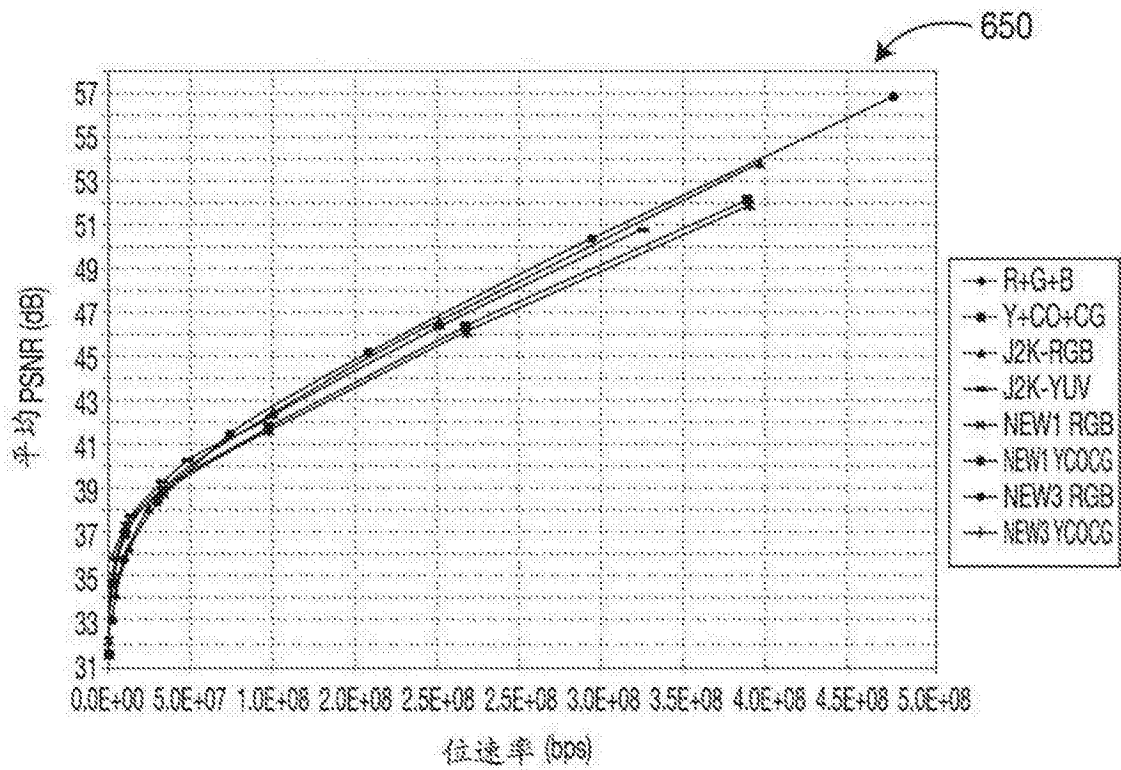


图6B

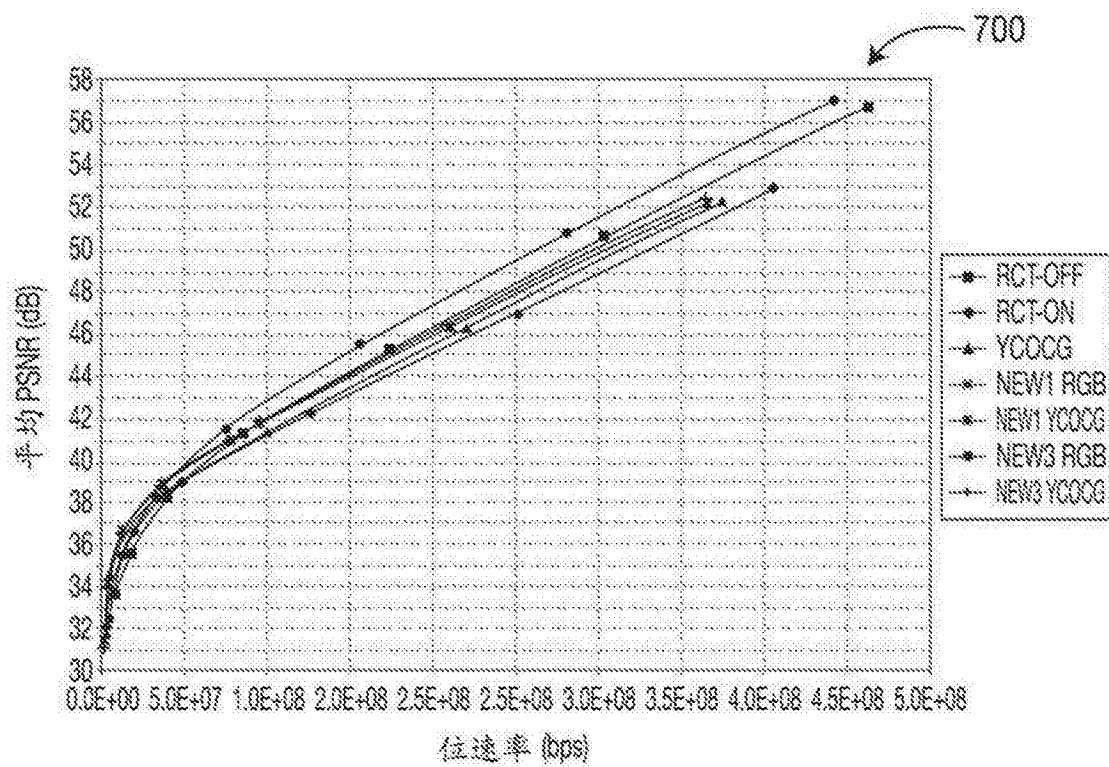


图7A

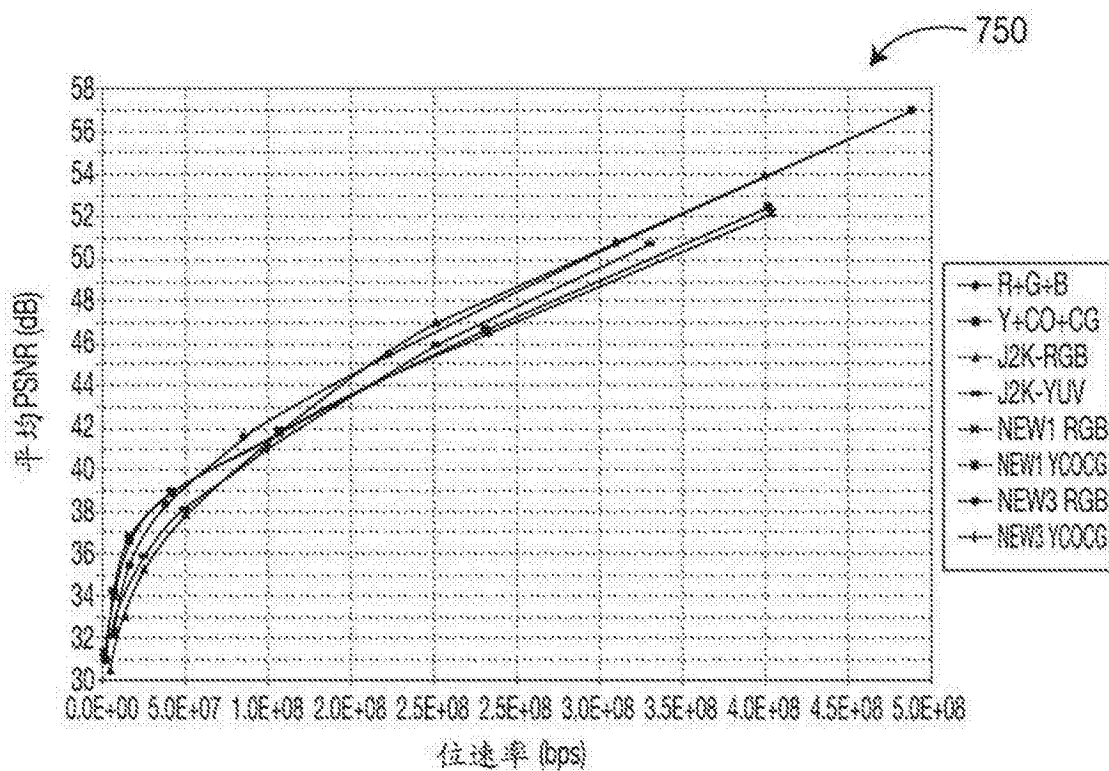


图7B

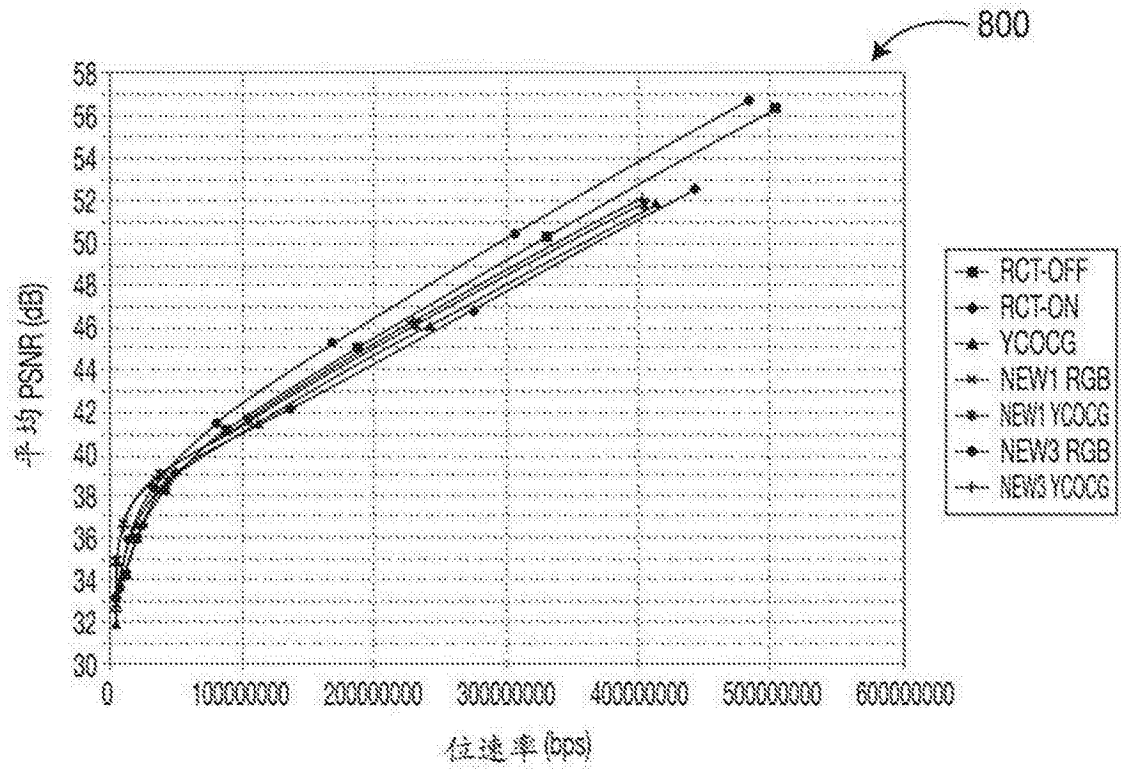


图8A

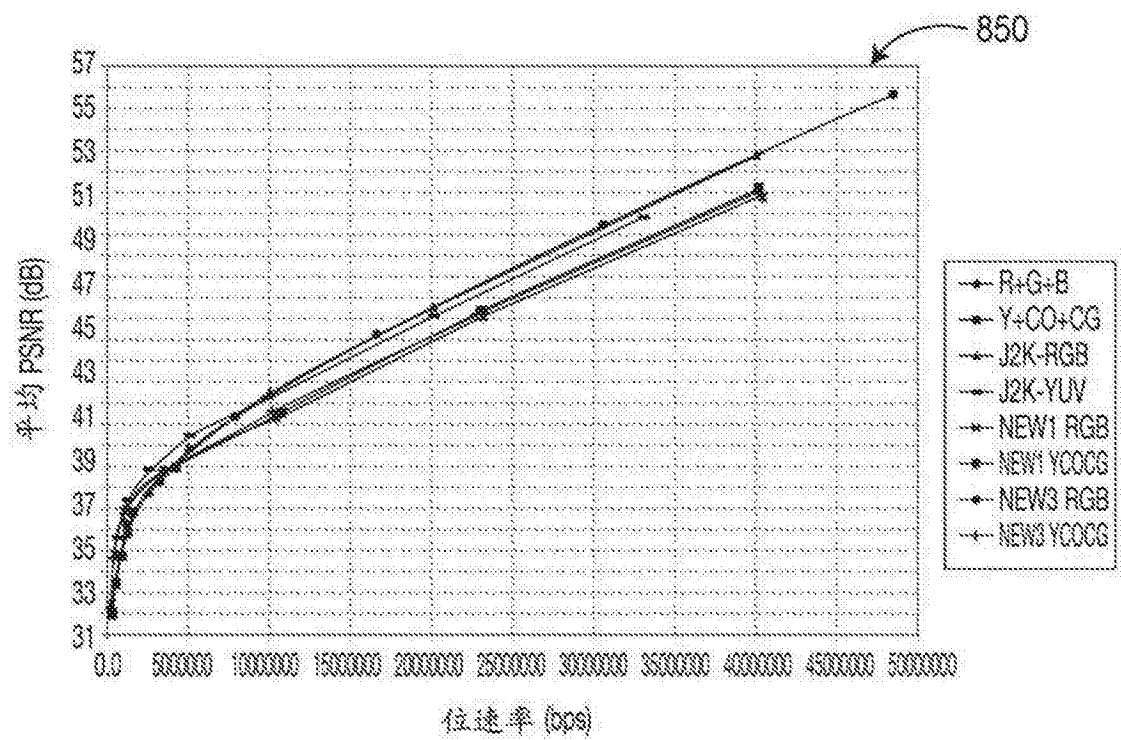


图8B

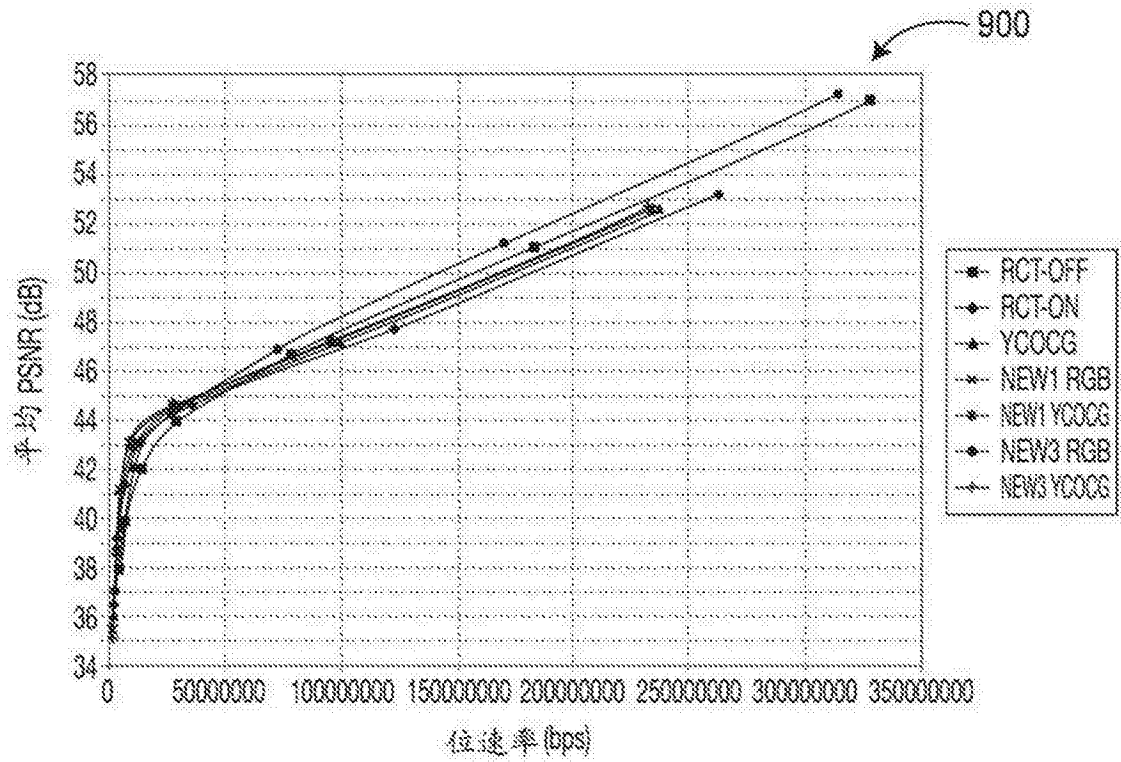


图9A

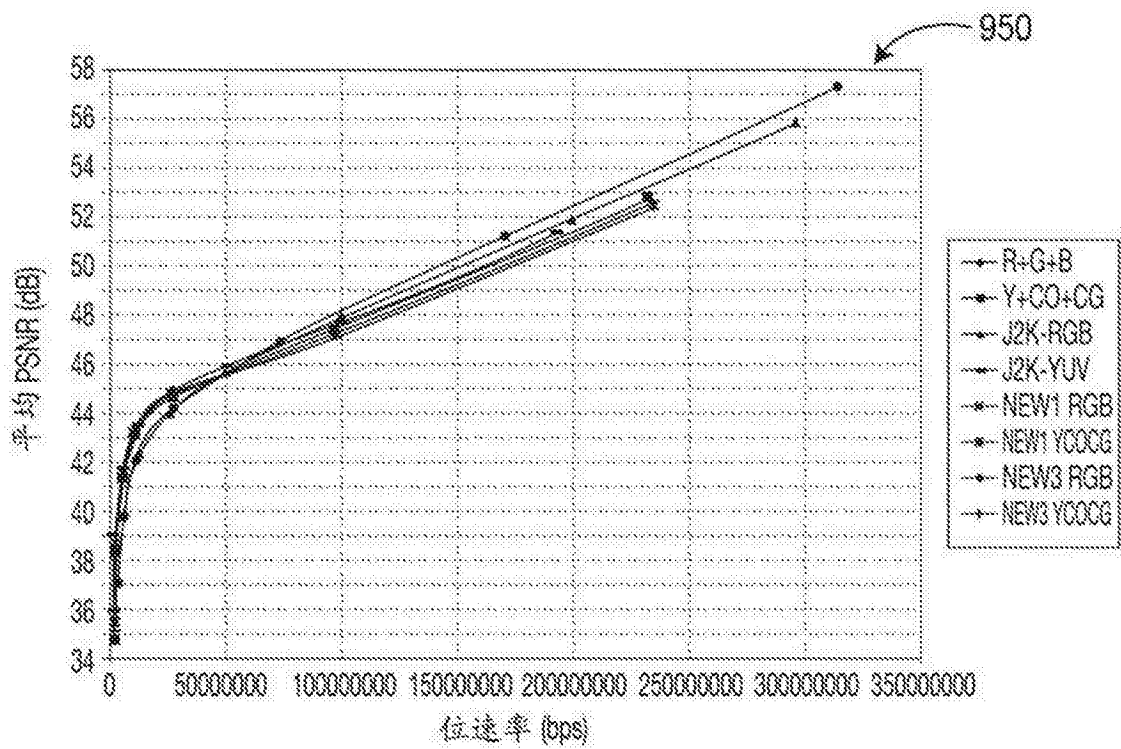


图9B

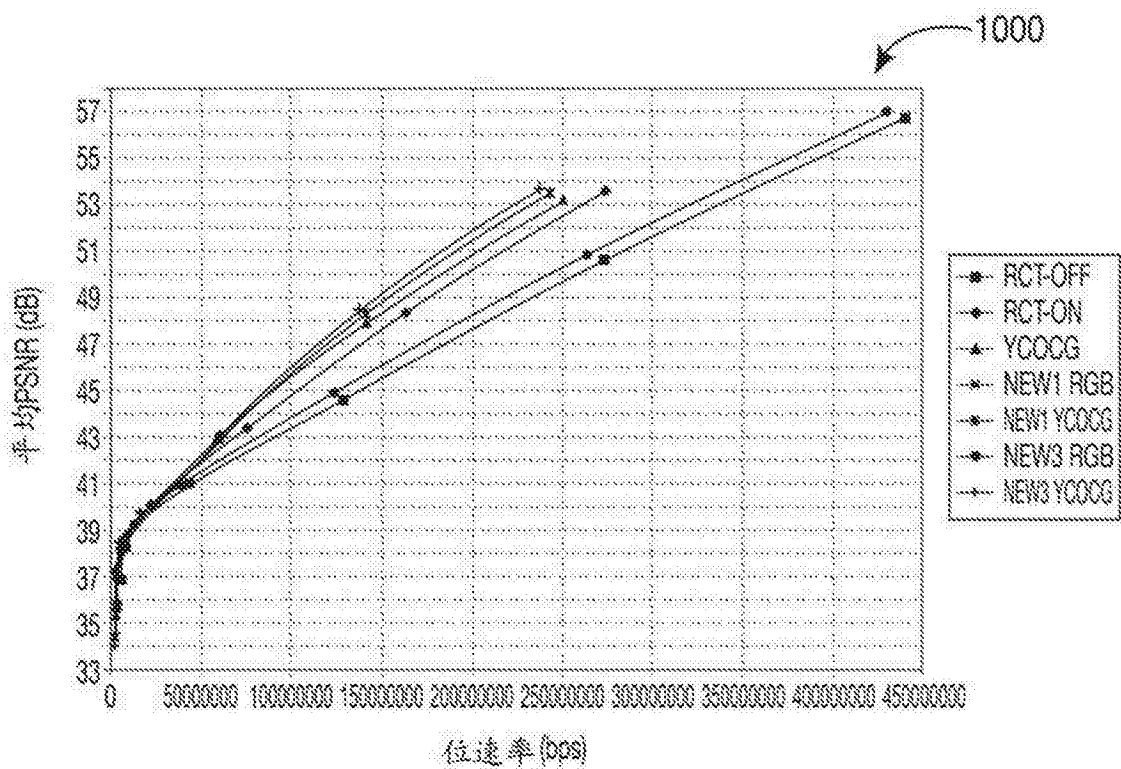


图10A

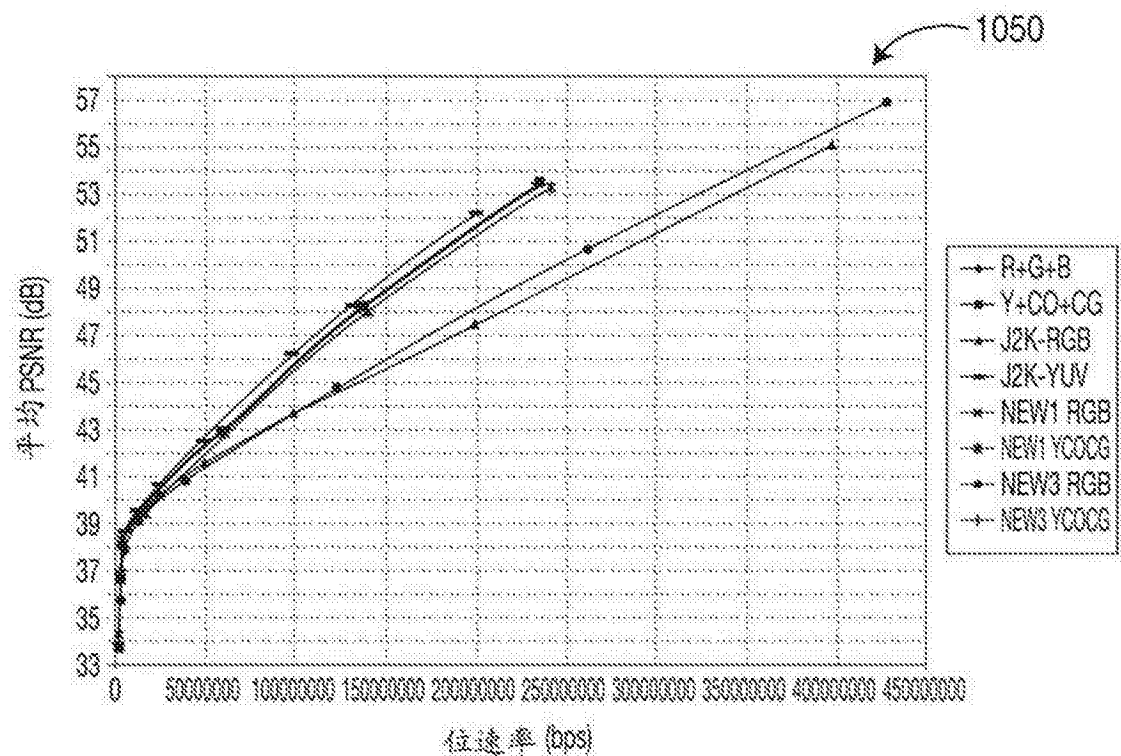


图10B

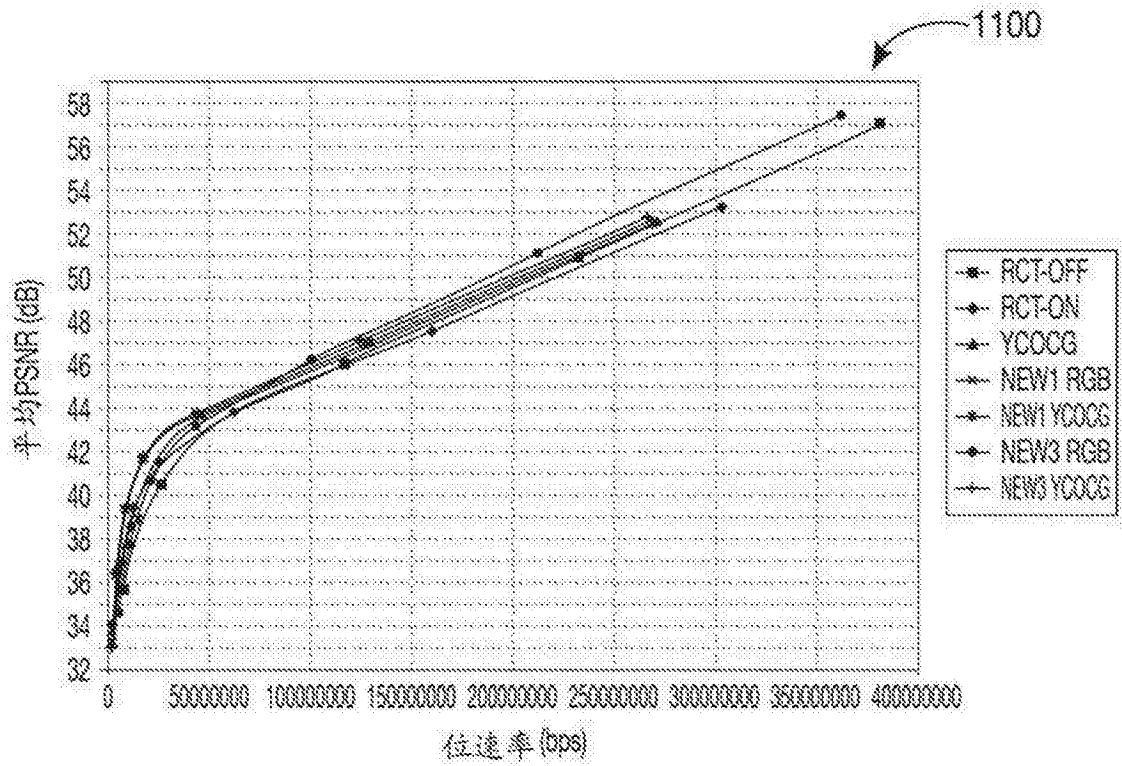


图11A

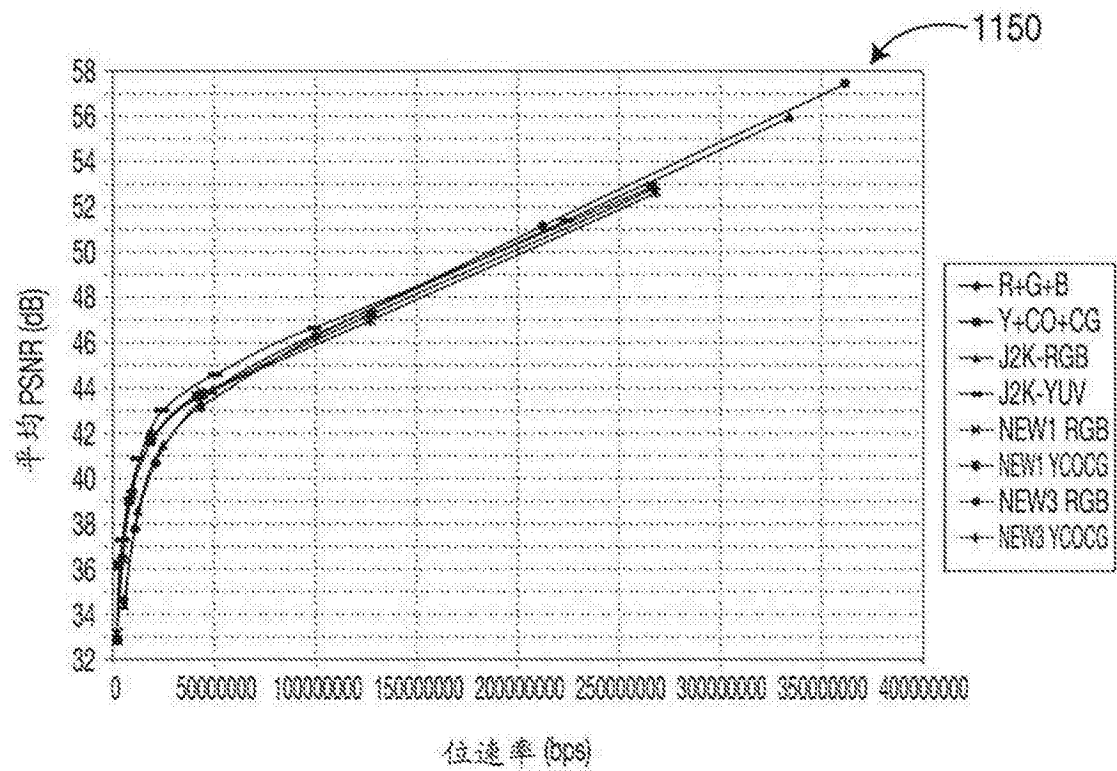


图11B

Residual() {	C	Descriptor
if(!entropy_coding_mode_flag)		
residual_block = residual_block_cavlc		
else		
residual_block = residual_block_cabac		
if(MbPartPredMode(mb_type, 0) == Intra_16x16)		
residual_block(Intra16x16DCLevel, 16)	3	
for(i8x8 = 0; i8x8 < 4; i8x8++) /* 每个 8x8 亮度块 */		
if(!transform_size_8x8_flag !entropy_coding_mode_flag)		
for(i4x4 = 0; i4x4 < 4; i4x4++) { /* 块的每个 4x4 子块 */		
if(CodedBlockPatternLuma & (1 << i8x8))		
if(MbPartPredMode(mb_type, 0) == Intra_16x16)		
residual_block(Intra16x16ACLevel[i8x8 * 4 + i4x4], 15)	3	
else		
residual_block(LumaLevel[i8x8 * 4 + i4x4], 16)	3 4	
else if(MbPartPredMode(mb_type, 0) == Intra_16x16)		
for(i = 0; i < 15; i++)		
Intra16x16ACLevel[i8x8 * 4 + i4x4][i] = 0		
else		
for(i = 0; i < 16; i++)		
LumaLevel[i8x8 * 4 + i4x4][i] = 0		
if(!entropy_coding_mode_flag && transform_size_8x8_flag)		
for(i = 0; i < 16; i++)		
LumaLevel8x8[i8x8][4 * i + i4x4] =		
LumaLevel[i8x8 * 4 + i4x4][i]		
}		

图13A

else if(CodedBlockPatternLuma & (1 << i8x8))		
residual_block(LumaLevel8x8[i8x8], 64)	3 4	
else		
for(i = 0; i < 64; i++)		
LumaLevel8x8[i8x8][i] = 0		
if(chroma_format_idc == 1 chroma_format_idc == 2) {		
NumC8x8 = 4 / (SubWidthC * SubHeightC)		
for(iCbCr = 0; iCbCr < 2; iCbCr++)		
if(CodedBlockPatternChroma & 3) /* 存在色度直流残差 */		
residual_block(ChromaDCLevel[iCbCr], 4 * NumC8x8)	3 4	
else		
for(i = 0; i < 4 * NumC8x8; i++)		
ChromaDCLevel[iCbCr][i] = 0		
for(iCbCr = 0; iCbCr < 2; iCbCr++)		
for(i8x8 = 0; i8x8 < NumC8x8; i8x8++)		
for(i4x4 = 0; i4x4 < 4; i4x4++)		
if(CodedBlockPatternChroma & 2)		
/* 存在色度交流残差 */		
residual_block(ChromaACLevel[iCbCr][i8x8*4+i4x4], 15)	3 4	
else		
for(i = 0; i < 15; i++)		
ChromaACLevel[iCbCr][i8x8*4+i4x4][i] = 0		
} else if(chroma_format_idc == 3) {		
if(MbPartPredMode(mb_type, 0) == Intra_16x16) /* Cb for 4:4:4 */		
residual_block(CbIntra16x16DCLevel, 16)	3	
for(i8x8 = 0; i8x8 < 4; i8x8++) /* 每个 8x8 Cb 块 */		
if(!transform_size_8x8_flag !entropy_coding_mode_flag)		
for(i4x4 = 0; i4x4 < 4; i4x4++) { /* 块的每个 4x4 子块		
*/		
if(CodedBlockPatternLuma & (1 << i8x8))		

图13B

if(MbPartPredMode(mb_type, 0) == Intra_16x16)		
residual_block(CbIntra16x16ACLevel[i8x8*4 + i4x4], 15)	3	
else		
residual_block(CbLevel[i8x8 * 4 + i4x4], 16)	3 4	
else if(MbPartPredMode(mb_type, 0) == Intra_16x16)		
for(i = 0; i < 15; i++)		
CbIntra16x16ACLevel[i8x8 * 4 + i4x4][i] = 0		
else		
for(i = 0; i < 16; i++)		
CbLevel[i8x8 * 4 + i4x4][i] = 0		
if(!entropy_coding_mode_flag && transform_size_8x8_flag)		
for(i = 0; i < 16; i++)		
CbLevel8x8[i8x8][4 * i + i4x4] =		
CbLevel[i8x8 * 4 + i4x4][i]		
}		
else if(CodedBlockPatternLuma & (1 << i8x8))		
residual_block(CbLevel8x8[i8x8], 64)	3 4	
else		
for(i = 0; i < 64; i++)		
CbLevel8x8[i8x8][i] = 0		
if(MbPartPredMode(mb_type, 0) == Intra_16x16) /*Cr for 4:4:4 */		
residual_block(CrIntra16x16DCLevel, 16)	3	
for(i8x8 = 0; i8x8 < 4; i8x8++) /*每个 8x8 Cr 块*/		
if(!transform_size_8x8_flag !entropy_coding_mode_flag)		
for(i4x4 = 0; i4x4 < 4; i4x4++) { /* 块的每个 4x4 子块		
*/		
if(CodedBlockPatternLuma & (1 << i8x8))		
if(MbPartPredMode(mb_type, 0) == Intra_16x16)		
residual_block(CrIntra16x16ACLevel[i8x8*4 + i4x4], 15)	3	
else		

图13C

residual_block(CrLevel[i8x8 * 4 + i4x4], 16)	3 4	
else if(MbPartPredMode(mb_type, 0) == Intra_16x16)		
for(i = 0; i < 15; i++)		
CrIntra16x16ACLevel[i8x8 * 4 + i4x4][i] = 0		
else		
for(i = 0; i < 16; i++)		
CrLevel[i8x8 * 4 + i4x4][i] = 0		
if(lentropy_coding_mode_flag && transform_size_8x8_flag)		
for(i = 0; i < 16; i++)		
CrLevel8x8[i8x8][4 * i + i4x4] = CrLevel[i8x8 * 4 + i4x4][i]		
}		
else if(CodedBlockPatternLuma & (1 << i8x8))		
residual_block(CrLevel8x8[i8x8], 64)	3 4	
else		
for(i = 0; i < 64; i++)		
CrLevel8x8[i8x8][i] = 0		
}		

图13D

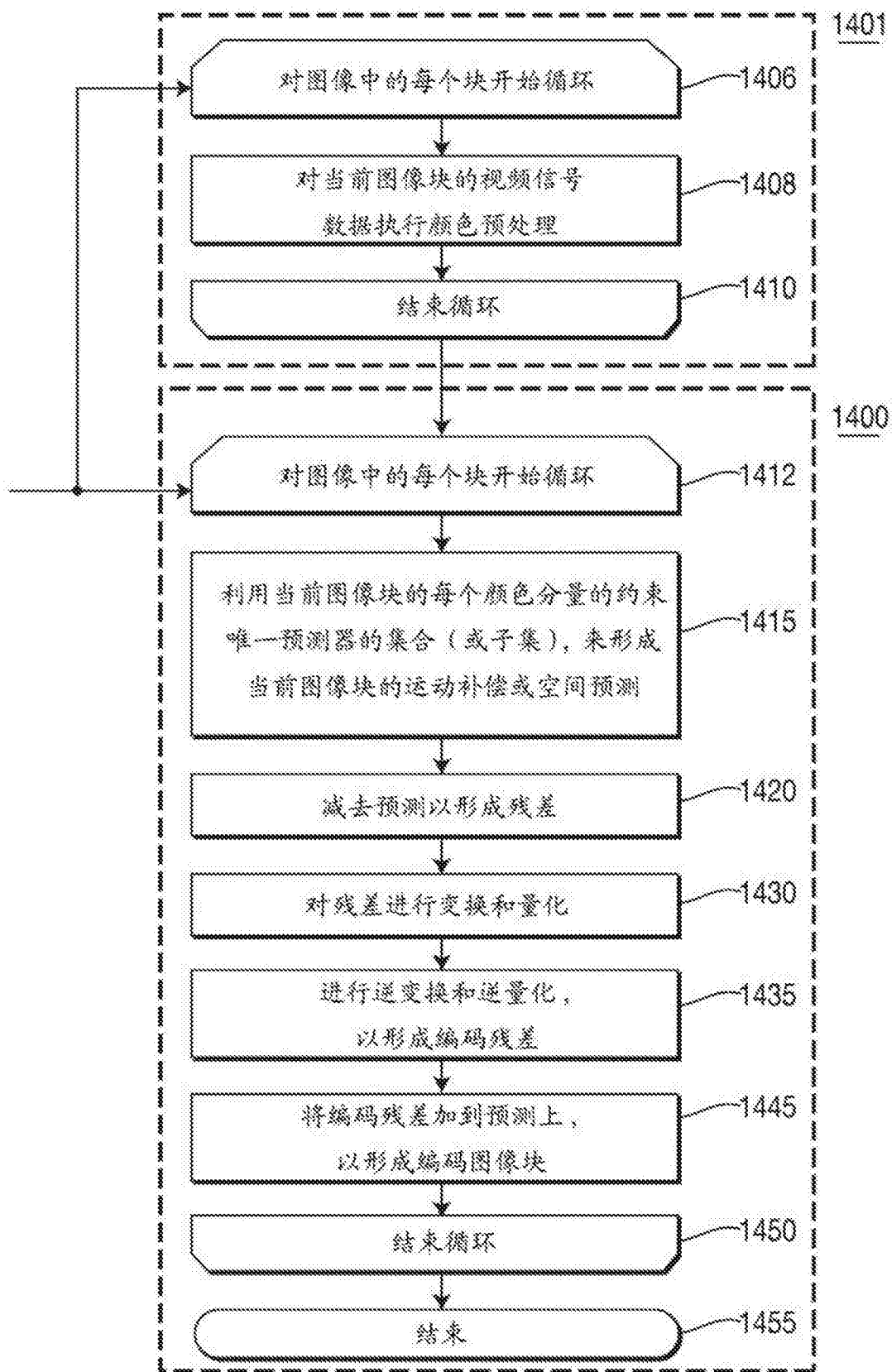


图14

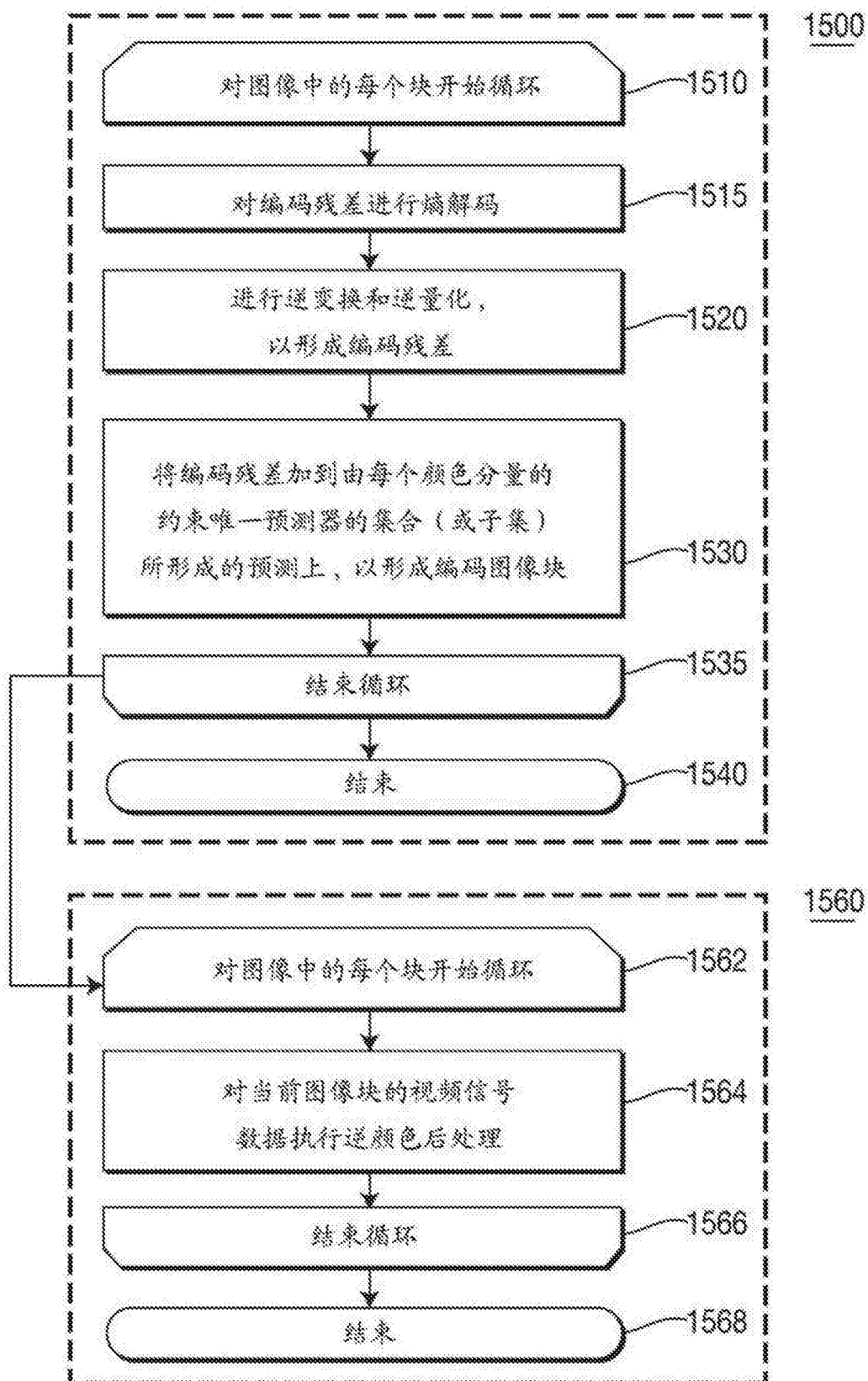


图15

1600

mb_pred(mb_type){	C	Descriptor
if(MbPartPredMode(mb_type, 0) == Intra_4x4		
MbPartPredMode(mb_type, 0) == Intra_8x8		
MbPartPredMode(mb_type, 0) == Intra_16x16){		
if(profile_idc != 166){		
if(MbPartPredMode(mb_type, 0) == Intra_4x4)		
for(luma4x4BlkIdx=0; luma4x4BlkIdx<16; luma4x4BlkIdx++){		
prev_intra4x4_pred_mode_flag[luma4x4BlkIdx]	2	u(1) ae(v)
if(!prev_intra4x4_pred_mode_flag[luma4x4BlkIdx])		
rem_intra4x4_pred_mode[luma4x4BlkIdx]	2	u(3) ae(v)
}		
if(MbPartPredMode(mb_type, 0) == Intra_8x8)		
for(luma8x8BlkIdx=0; luma8x8BlkIdx<4; luma8x8BlkIdx++){		
prev_intra8x8_pred_mode_flag[luma8x8BlkIdx]	2	u(1) ae(v)
if(!prev_intra8x8_pred_mode_flag[luma8x8BlkIdx])		
rem_intra8x8_pred_mode[luma8x8BlkIdx]	2	u(3) ae(v)
}		
if(chroma_format_idc != 0)		
intra_chroma_pred_mode	2	u(v) ae(v)
} else {		
if(MbPartPredMode(mb_type, 0) == Intra_4x4)		
for(luma4x4BlkIdx=0; luma4x4BlkIdx<16; luma4x4BlkIdx++){		
prev_intra4x4_pred_mode_flag0[luma4x4BlkIdx]	2	u(1) ae(v)
if(!prev_intra4x4_pred_mode_flag0[luma4x4BlkIdx])		
rem_intra4x4_pred_mode0[luma4x4BlkIdx]	2	u(3) ae(v)
prev_intra4x4_pred_mode_flag1[luma4x4BlkIdx]	2	u(1) ae(v)

图16A

if(!prev_intra4x4_pred_mode_flag[luma4x8BIdx])		
rem_intra4x4_pred_mode1[luma4x8BIdx]	2	u(3) ae(v)
prev_intra4x4_pred_mode_flag2[luma4x8BIdx]	2	u(1) ae(v)
if(!prev_intra4x4_pred_mode_flag2[luma4x8BIdx])		
rem_intra4x4_pred_mode2[luma4x8BIdx]	2	u(3) ae(v)
}		
if(MbPartPredMode(mb_type, 0) == Intra_8x8)		
for(luma8x8BIdx=0; luma8x8BIdx<4; luma8x8BIdx++) {		
prev_intra8x8_pred_mode_flag0[luma8x8BIdx]	2	u(1) ae(v)
if(!prev_intra8x8_pred_mode_flag0[luma8x8BIdx])		
rem_intra8x8_pred_mode0[luma8x8BIdx]	2	u(3) ae(v)
prev_intra8x8_pred_mode_flag1[luma8x8BIdx]	2	u(1) ae(v)
if(!prev_intra8x8_pred_mode_flag1[luma8x8BIdx])		
rem_intra8x8_pred_mode1[luma8x8BIdx]	2	u(3) ae(v)
prev_intra8x8_pred_mode_flag2[luma8x8BIdx]	2	u(1) ae(v)
if(!prev_intra8x8_pred_mode_flag2[luma8x8BIdx])		
rem_intra8x8_pred_mode2[luma8x8BIdx]	2	u(3) ae(v)
}		
}		
} else if(MbPartPredMode(mb_type, 0) != Direct) {		

图16B