



(10) 授权公告号 CN 113632487 B

(45) 授权公告日 2024. 07. 19

(21) 申请号 202080024077.6

(22) 申请日 2020.09.22

(65) 同一申请的已公布的文献号
申请公布号 CN 113632487 A

(43) 申请公布日 2021.11.09

(30) 优先权数据

62/905,339 2019.09.24 US

17/024,246 2020.09.17 US

(85) PCT国际申请进入国家阶段日
2021.09.24(86) PCT国际申请的申请数据
PCT/US2020/051975 2020.09.22(87) PCT国际申请的公布数据
W02021/061631 EN 2021.04.01

(73) 专利权人 腾讯美国有限责任公司

地址 美国加利福尼亚州帕洛阿尔托公园大道2747号

(72) 发明人 杜祎鑫 崔秉斗 李翔 赵欣
史蒂芬·文格尔 刘杉

(74) 专利代理机构 北京德琦知识产权代理有限公司 11018

专利代理师 徐文静 陈世华

(51) Int.Cl.
H04N 19/61 (2006.01)

(56) 对比文件

CN 103988518 A, 2014.08.13

CN 104768005 A, 2015.07.08

CN 107079160 A, 2017.08.18

审查员 何美伶

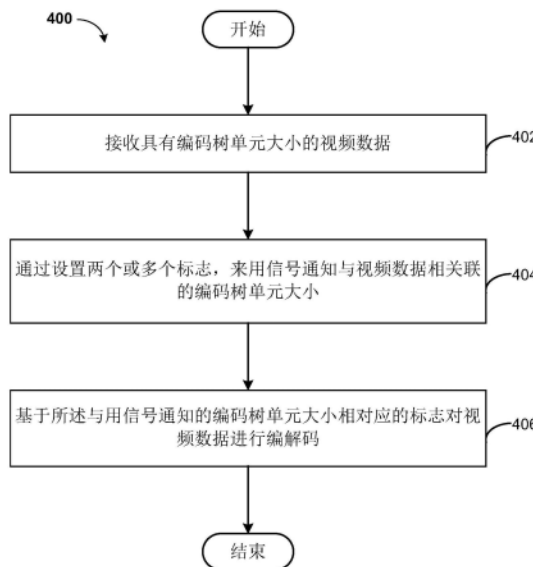
权利要求书2页 说明书13页 附图7页

(54) 发明名称

一种对视频数据进行编解码的方法、系统以及存储介质

(57) 摘要

提供一种对视频数据进行编解码的方法、计算机程序和计算机系统。接收具有编码树单元大小的视频数据。通过设置两个或多个标志,来用信号通知与视频数据相关联的编码树单元大小。基于与用信号通知的编码树单元大小相对应的标志来编码/解码视频数据。



1. 一种对视频数据进行编码的方法,其特征在于,所述方法包括:

接收具有编码树单元大小的视频数据;

通过设置第一标志,来用信号通知与所述视频数据相关联的所述编码树单元大小,所述第一标志包括32像素编码树单元大小标志、64像素编码树单元大小标志和128像素编码树单元大小标志中的至少一个;以及

基于与用信号通知的所述编码树单元大小相对应的所述第一标志来对所述视频数据进行编码;

所述方法进一步包括:

基于所述编码树单元大小大于或等于64*64像素,用信号通知第二标志sps_max_luma_transform_size_64_flag,所述第二标志用于指示64像素最大亮度变换大小;

其中,用于指示64像素最大亮度变换大小的所述第二标志等于1用于指示亮度样本中的最大变换大小等于64;用于指示64像素最大亮度变换大小的所述第二标志等于0用于指示所述亮度样本中的最大变换大小等于32。

2. 根据权利要求1所述的方法,其特征在于,所述通过设置第一标志,来用信号通知与所述视频数据相关联的所述编码树单元大小,包括:

用信号通知所述32像素编码树单元大小标志,并且,当所述32像素编码树单元大小标志不等于1时,用信号通知所述64像素编码树单元大小标志。

3. 根据权利要求1所述的方法,其特征在于,所述通过设置第一标志,来用信号通知与所述视频数据相关联的所述编码树单元大小,包括:

用信号通知所述64像素编码树单元大小标志,并且,当所述64像素编码树单元大小标志不等于1时,用信号通知所述32像素编码树单元大小标志。

4. 根据权利要求1所述的方法,其特征在于,所述通过设置第一标志,来用信号通知与所述视频数据相关联的所述编码树单元大小,包括:

用信号通知所述32像素编码树单元大小标志,并且,当所述32像素编码树单元大小标志不等于1时,用信号通知所述128像素编码树单元大小标志。

5. 根据权利要求1所述的方法,其特征在于,所述通过设置第一标志,来用信号通知与所述视频数据相关联的所述编码树单元大小,包括:

用信号通知所述128像素编码树单元大小标志,并且,当所述128像素编码树单元大小标志不等于1时,用信号通知所述32像素编码树单元大小标志。

6. 根据权利要求1所述的方法,其特征在于,所述通过设置第一标志,来用信号通知与所述视频数据相关联的所述编码树单元大小,包括:

用信号通知所述64像素编码树单元大小标志,并且,当所述64像素编码树单元大小标志不等于1时,用信号通知所述128像素编码树单元大小标志。

7. 根据权利要求1所述的方法,其特征在于,所述通过设置第一标志,来用信号通知与所述视频数据相关联的所述编码树单元大小,包括:

用信号通知所述128像素编码树单元大小标志,并且,当所述128像素编码树单元大小标志不等于1时,用信号通知所述64像素编码树单元大小标志。

8. 根据权利要求1所述的方法,其特征在于,所述方法进一步包括:

基于用信号通知所述32像素编码树单元大小标志并且所述32像素编码树单元大小标

志等于1,不用信号通知所述64像素最大亮度变换大小标志。

9.根据权利要求1所述的方法,其特征在于,所述方法进一步包括:

基于用信号通知所述64像素编码树单元大小标志并且所述64像素编码树单元大小标志等于0,用信号通知所述32像素编码树单元大小标志并将其设置为等于1,并且不用信号通知所述64像素最大亮度变换大小标志。

10.根据权利要求2所述的方法,其特征在于,所述方法进一步包括:

基于用信号通知所述128像素编码树单元大小标志并且所述128像素编码树单元大小标志等于0,用信号通知所述32像素编码树单元大小标志并且将其设置为等于1,并且不用信号通知所述64像素最大亮度变换大小标志。

11.根据权利要求1-10任一项所述的方法,其特征在于,所述方法进一步包括:设置与序列参数集相对应的第三标志和第四标志,所述序列参数集与所述视频数据相关联,所述第三标志用于指示是否应用最小编码树单元大小,并且所述第四标志指示是否应用最大编码树单元大小。

12.根据权利要求11所述的方法,其特征在于,所述方法进一步包括:用信号通知所述第三标志,并且,基于所述最小编码树单元大小未被应用,用信号通知所述第四标志。

13.根据权利要求11所述的方法,其特征在于,所述方法进一步包括:用信号通知所述第四标志,并且,基于所述最大编码树单元大小未被应用,用信号通知所述第三标志。

14.根据权利要求11所述的方法,其特征在于,所述方法进一步包括:基于所述最小编码树单元大小未被应用,用信号通知64像素最大亮度变换大小标志。

15.一种对视频数据进行解码的方法,其特征在于,所述方法包括:

接收具有编码树单元大小的视频数据;

通过解析第一标志来确定所述视频数据相关联的所述编码树单元大小,所述第一标志包括32像素编码树单元大小标志、64像素编码树单元大小标志和128像素编码树单元大小标志中的至少一个;以及

基于与所述编码树单元大小相对应的所述第一标志,来对所述视频数据进行解码;

所述方法进一步包括:

接收用信号通知的第二标志spsmax_luma_transform_size_64_flag并确定取值,所述第二标志用于指示64像素最大亮度变换大小,其中,用于指示64像素最大亮度变换大小的所述第二标志等于1用于指示亮度样本中的最大变换大小等于64;用于指示64像素最大亮度变换大小的所述第二标志等于0用于指示亮度样本中的最大变换大小等于32。

16.一种对视频数据进行编码的计算机系统,其特征在于,所述计算机系统包括:

一个或多个计算机可读非易失性存储介质,被配置为存储计算机程序代码;以及

一个或多个计算机处理器,被配置为访问所述计算机程序代码并且如所述计算机程序代码所指示执行如权利要求1-15任一项所述的方法。

17.一种非易失性计算机可读介质,其特征在于,其上存储有对视频数据进行编解码的计算机程序,所述计算机程序被配置为使一个或多个计算机处理器如权利要求1-15中任一项所述的方法。

18.一种服务器计算机,包括:处理器和存储器;所述存储器存储有计算机程序,所述计算机程序被所述处理器执行时,使得所述处理器执行如权利要求1-15中任一项所述的方法。

一种对视频数据进行编解码的方法、系统以及存储介质

[0001] 交叉引用

[0002] 本申请要求于2019年9月24日在美国专利和商标局提交的第62/905,339号美国临时专利申请和于2020年9月17日在美国专利和商标局提交的第17/024,246号美国专利申请的优先权,这两个专利申请的全部内容通过引用结合在本申请中。

技术领域

[0003] 本公开总体上涉及数据处理领域,并且更具体地涉及视频编码和解码。

背景技术

[0004] ITU-T VCEG (Q6/16) 和ISO/IEC MPEG (JTC 1/SC 29/WG 11) 在2013年(版本1)、2014年(版本2)、2015年(版本3)和2016年(版本4)出版了H.265/HEVC(高效率视频编码)标准。2015年,这两个标准组织共同形成了联合视频探索小组(JVET, Joint Video Exploration Team),以探索开发超越HEVC的下一视频编码标准的潜力。2017年10月,他们发布了关于具有超越HEVC能力的视频压缩联合提案(CfP, Call for Proposals)。到2018年2月15日,分别提交了标准动态范围(SDR, standard dynamic range)上的总共22个CfP响应、高动态范围(HDR, high dynamic range)上的12个CfP响应和360个视频类别上的12个CfP响应。2018年4月,所有收到的CfP响应都在122MPEG/第十次JVET会议中进行了评估。作为此次会议的成果,JVET正式启动了超越HEVC的下一代视频编码标准化过程。新标准被命名为下一代视频编码(VVC, Versatile Video Coding),并且JVET被重命名为联合视频专家组。VTM(VVC测试模型)的当前版本,即VTM 6。

发明内容

[0005] 实施例涉及对视频数据进行编解码的方法、系统以及计算机可读存储介质。根据一个方面,提供了一种对视频数据进行编解码的方法。所述方法包括:接收具有编码树单元大小的视频数据;通过设置两个或多个标志,来用信号通知与所述视频数据相关联的所述编码树单元大小;以及基于与所述用信号通知的编码树单元大小相对应的标志来对所述视频数据进行编解码。

[0006] 根据另一方面,提供了对视频数据进行编解码的计算机系统。所述计算机系统包括一个或多个处理器,一个或多个计算机可读存储器,一个或多个计算机可读有形存储设备,以及存储在所述一个或多个存储设备中的至少一个存储设备中的程序指令,所述程序指令由所述一个或多个处理器中的至少一个处理器通过一个或多个存储器中的至少一个存储器执行,其中,所述计算机系统能够执行一个方法。所述方法包括接收具有编码树单元大小的视频数据;通过设置两个或多个标志,来用信号通知与所述视频数据相关联的所述编码树单元大小;以及基于与所述用信号通知的编码树单元大小相对应的标志来对所述视频数据进行编解码。

[0007] 根据又一方面,提供了一种对视频数据进行编解码的计算机可读介质。所述计算

机可读存储介质可以包括一个或多个计算机可读存储设备以及存储在所述一个或多个有形存储设备中的至少一个有形存储设备上的程序指令,所述程序指令可由处理器执行。所述程序指令由处理器执行以执行方法,包括:接收具有编码树单元大小的视频数据;通过设置两个或多个标志,来用信号通知与所述视频数据相关联的所述编码树单元大小;以及基于与所述用信号通知的编码树单元大小相对应的标志来对所述视频数据进行编解码。

附图说明

[0008] 这些和其它目的、特征和优点将从以下结合附图阅读的说明性实施例的详细描述中变得显而易见。由于附图是用于方便本领域技术人员结合详细描述进行清楚的理解,因此附图的各种特征并非按比例绘制。在附图中:

[0009] 图1示出了根据至少一个实施例的联网计算机环境;

[0010] 图2是根据至少一个实施例的四叉树/二叉树(QTBT,quad-tree/binary-tree)块结构的示意图;

[0011] 图3A-3D示出了根据至少一个实施例的示例性语法元素;

[0012] 图4示出了根据至少一个实施例的由视频编解码的程序执行的步骤的操作流程图;

[0013] 图5是根据至少一个实施例的图1中描绘的计算机和服务器的内部部件和外部部件的框图;

[0014] 图6是根据至少一个实施例的包括图1中描绘的计算机系统的示例性云计算环境的框图;以及

[0015] 图7是根据至少一个实施例的图6的示例性云计算环境的功能层的框图。

具体实施方式

[0016] 本文公开了所要求保护的结构和方法的详细实施例;然而,可理解的是,所公开的实施例仅仅是图示了可以以各种形式实施的所要求保护的结构和方法。然而,这些结构和方法可以以许多不同的形式来实施,并且不应当被解释为限于本文阐述的示例性实施例。相反,提供这些示例性实施例是为了使本公开清楚和完整,并且将范围完整地传达给本领域技术人员。在描述中,可以省略公知特征和技术的细节,以避免不必要地模糊所呈现的实施例。

[0017] 实施例总体上涉及数据处理领域,并且更具体地涉及数据编码和解码。下面描述的示例性实施例提供了一种系统、方法和计算机程序,使用单独的标志来代替编码树单元大小语法来编码视频数据等。因此,一些实施例具有通过标志发送编码树单元大小来节省比特位以使用更少的存储器来改进计算领域的能力。

[0018] ITU-T VCEG(Q6/16)和ISO/IEC MPEG(JTC 1/SC 29/WG 11)在2013年(版本1)、2014年(版本2)、2015年(版本3)和2016年(版本4)出版了H.265/HEVC(高效率视频编码)标准。2015年,这两个标准组织共同形成了联合视频探索小组(JVET,Joint Video Exploration Team),以探索开发超越HEVC的下一视频编码标准的潜力。2017年10月,他们发布了关于具有超越HEVC能力的视频压缩联合提案(CfP,Call for Proposals)。到2018年2月15日,分别提交了标准动态范围(SDR,standard dynamic range)上的总共22个CfP响

应、高动态范围(HDR,high dynamic range)上的12个CfP响应和360个视频类别上的12个CfP响应。2018年4月,所有收到的CfP响应都在122MPEG/第十次JVET会议中进行了评估。作为此次会议的成果,JVET正式启动了超越HEVC的下一代视频编码标准化过程。新标准被命名为下一代视频编码(VVC,Versatile Video Coding),并且JVET被重命名为联合视频专家组。VTM(VVC测试模型)的当前版本,即VTM 6。

[0019] 在HEVC中,通过使用表示为编码树的四叉树结构将编码树单元分割成编码单元以适应各种局部特性。在编码单元级做出关于是否使用图片间(时间)预测还是图片内(空间)预测来编码图片区域的决定。可以根据预测单元分割类型将每个编码单元进一步分割为一个、两个或四个预测单元。在一个预测单元内部,应用相同的预测过程,并且在预测单元的基础上将相关信息发送到解码器。在通过应用基于预测单元分割类型的预测过程来获得残差块之后,可以根据另一四叉树结构(如用于编码单元的编码树)将编码单元分区成变换单元。HEVC结构的关键特征中的一个它是它具有多个分区概念,所述多个分区概念包括编码单元、预测单元和变换单元。然而,使用固定长度编码 $u(2)$ 来描述语法 $\log_2 \text{ctu_size_minus5}$ 可能浪费一个比特,因为可能仅三个数要分别进行编码:0、1或2,如果编码数是0或1,则 $u(2)$ 可能会浪费一个比特。因此,用单独的标志代替原始编码树单元大小语法对节省序列参数集中的比特可能是有利的。

[0020] 本文参考根据各个实施例的方法、装置(系统)和计算机可读介质的流程图和/或框图来描述各方面。应当理解,流程图图示和/或框图的每个框,以及流程图图示和/或框图中的框的组合可由计算机可读程序指令实现。

[0021] 参考图1,联网计算机环境的功能框图示出了用于使用单独的标志来代替编码树单元大小语法对视频数据进行编解码的视频编解码系统100(以下被称为“系统”)。应当理解,图1仅提供一种实施方案的图示,并不意味着对可实现不同实施例的环境的任何限制。可基于设计和实现需求对所描绘的环境进行许多修改。

[0022] 系统100可以包括计算机102和服务器计算机114。计算机102可通过通信网络110(以下被称为“网络”)与服务器计算机114进行通信。计算机102包括处理器104以及存储在数据存储设备106上的软件程序108,并且能够与用户对接以及与服务服务器计算机114通信。如下面将参考图4讨论的,计算机102可以分别包括内部组件800A和外部组件900A,并且服务器计算机114可以分别包括内部组件800B和外部组件900B。计算机102可为例如移动设备、电话、个人数字助理、上网本、膝上型计算机、平板计算机、台式计算机或者能够运行程序、访问网络和访问数据库的任何类型的计算设备。

[0023] 如以下结合图6和图7所讨论的,服务器计算机114还可以在云计算服务模型中操作,诸如软件即服务(Software as a Service,SaaS)、平台即服务(Platform as a Service,PaaS)或基础设施即服务(Infrastructure as a Service,IaaS)。服务器计算机114还可位于云计算部署模型中,诸如私有云、社区云、公共云或混合云中。

[0024] 用于对视频数据进行编解码的服务器计算机114能够运行与数据库112交互的视频编解码程序116(以下被称为“程序”)。下面将结合图4更详细地解释视频编解码程序的方法。在一个实施例中,计算机102可作为包括用户接口的输入设备操作,而程序116可主要在服务器计算机114上运行。在可替代实施例中,程序116可主要在至少一个计算机102上运行,而服务器计算机114可用于处理和存储程序116所使用的数据。应注意,程序116可以是

独立的程序,或者可以集成到更大的视频编解码程序中。

[0025] 然而,应当注意,在一些实例中,可以以任何比例在计算机102和服务器计算机114之间共享对程序116的处理。在另一个实施例中,程序116可在一个以上计算机、服务器计算机或者计算机和服务器计算机的某一组合上操作,例如,通过网络110与单个服务器计算机114进行通信的多个计算机102。在另一个实施例中,例如,程序116可在多个服务器计算机114上操作,所述多个服务器计算机114通过网络110与多个客户端计算机进行通信。可替代地,程序可在网络服务器上运行,该网络服务器通过网络与服务器和多个客户端计算机进行通信。

[0026] 网络110可以包括有线连接、无线连接、光纤连接或其某一组合。通常,网络110可为支持计算机102和服务器计算机114之间的通信的连接和协议的任何组合。网络110可以包括各种类型的网络,诸如,例如局域网(Local Area Network, LAN)、广域网(Wide Area Network, WAN) (例如因特网)、电信网络(例如公共交换电话网络(Public Switched Telephone Network, PSTN))、无线网络、公共交换网络、卫星网络、蜂窝网络(例如,第五代(the fifth generation, 5G)网络、长期演进(Long-Term Evolution LTE)网络、第三代(the third generation, 3G)网络、码分多址(Code Division Multiple Access, CDMA)网络等)、公共陆地移动网络(Public Land Mobile Network, PLMN)、城域网(Metropolitan Area Network, MAN)、专用网络、自组织网络、内部网、基于光纤的网络等,以及/或者这些或其它类型网络的组合。

[0027] 图1所示的设备和网络的数量和布置是作为示例提供的。实际上,可存在附加的设备和/或网络、更少的设备和/或网络、不同的设备和/或网络、或者与图1所示的设备和/或网络具有不同地布置的设备和/或网络。此外,图1所示的两个或更多个设备可在单个设备中实现,或者图1所示的单个设备可以实现为多个分布式设备。此外或可替代地,系统100的一组设备(例如,至少一个设备)可执行至少一个函数,所述至少一个函数被描述为由系统100的另一组设备执行。

[0028] 现在参考图2,描绘了示例性QTBT块结构200。QTBT块结构200可以包括使用QTBT的块分区。也可以描绘对应的树表示(tree representation)。实线可以指示四叉树分割,并且虚线可以指示二叉树分割。在二叉树的每个分割(即,非叶)节点中,可以发信号通知一个标志以指示可以使用哪个分割类型(即,水平或垂直),其中,0可以指示水平分割并且1可以指示垂直分割。对于四叉树分割,可能不需要指示分割类型,因为四叉树分割总是对块进行水平地和垂直地分割以产生具有相等大小的4个子块。

[0029] QTBT可以去除多个分区类型的概念。例如,QTBT可去除编码单元、预测单元和变换单元概念的分离,并且支持编码单元分区形状的更多灵活性。在QTBT块结构200中,编码单元可以具有正方形形状或矩形形状。编码树单元(CTU, coding tree unit)可以由四叉树结构进行分区。四叉树叶节点可由二叉树结构进一步分区。二叉树分割中可能存在两种分割类型:对称的水平分割和对称的垂直分割。

[0030] 二叉树叶节点可以称为编码单元(CU, coding unit),并且该分段可用于预测和变换处理而无需任何进一步分区。这意味着编码单元、预测单元和变换单元在QTBT编码块结构200中可以具有相同的块大小。编码单元可以包括不同色彩分量的编码块(CB, coding block) (例如,在4:2:0色度格式的P条带和B条带的情况下,一个编码单元可以包含一个亮

度CB和两个色度CB)或可以包括单个分量的CB(例如,在I条带的情况下,一个编码单元可以包含一个亮度CB或两个色度CB)。

[0031] 可以为QTBT分区方案定义以下参数:

[0032] 编码树单元大小可以是四叉树的根节点大小,与HEVC中的概念相同;

[0033] MinQTSIZE可以是最小允许四叉树叶节点大小;

[0034] MaxBTSIZE可以是最大允许二叉树根节点大小;

[0035] MaxBTDepth可以是最大允许二叉树深度;以及

[0036] MinBTSIZE可以是最小允许二叉树叶节点大小。

[0037] 在QTBT分区结构200的一个示例中,可以将编码树单元大小设置为 128×128 亮度样本,所述亮度样本具有两个对应的 64×64 色度样本块。可以将MinQTSIZE设置为 16×16 。可以将MaxBTSIZE设置为 64×64 。可以将MinBTSIZE(宽度和高度)设置为 4×4 。可以将MaxBTDepth设置为4。可以首先将四叉树分区应用于编码树单元以生成四叉树叶节点。四叉树叶节点可以具有从 16×16 (即,MinQTSIZE)到 128×128 (即,编码树单元大小)的大小。如果叶四叉树节点是 128×128 ,则它可能不会被二叉树进一步分割,因为其大小可能超过MaxBTSIZE(即, 64×64)。否则,叶四叉树节点可由二叉树进一步分区。因此,四叉树叶节点也可以是二叉树的根节点,并且它具有二叉树深度0。

[0038] 当二叉树深度达到MaxBTDepth(即,4)时,可以不考虑进一步的分割。当二叉树节点具有等于MinBTSIZE(即,4)的宽度时,可以不考虑进一步的水平分割。类似地,当二叉树节点具有等于MinBTSIZE的高度时,可以不考虑进一步的垂直分割。通过预测和变换处理来进一步处理二叉树的叶节点,而无需任何进一步的分区。在一个示例中,最大编码树单元大小可以是 256×256 亮度样本。

[0039] 另外,QTBT方案可以支持亮度和色度具有单独的QTBT结构的灵活性。当前,对于P条带和B条带(slice),一个编码树单元中的亮度CTB和色度CTB可以共享相同的QTBT结构。然而,对于I条带,可以通过QTBT结构将亮度CTB分区成CU,并且通过另一QTBT结构将色度CTB分区成色度编码单元。这意味着I条带中的CU可以包括亮度分量的编码块或两个色度分量的编码块,并且P条带或B条带中的编码单元由所有三个色彩分量的编码块组成。

[0040] 现在参考图3A至图3D,描绘了示例性语法元素300A-300D。语法元素300A-300D可用于用信号通知编码树单元大小以节省比特。

[0041] 根据一个或多个实施例,三个标志中的两个可用于用信号通知编码树单元大小,即,use_32_ctu_size_flag、use_64_ctu_size_flag和use_128_ctu_size_flag。在一个实施例中,可以首先用信号通知use_32_ctu_size_flag。如果use_32_ctu_size_flag可等于1,则可结束编码树单元大小信令。否则,可以用信号通知use_64_ctu_size_flag。在一个实施例中,可首先用信号通知use_64_ctu_size_flag。如果use_64_ctu_size_flag可等于1,则可结束编码树单元大小信令。否则,可以用信号通知use_32_ctu_size_flag。在一个实施例中,可以首先用信号通知use_32_ctu_size_flag。如果use_32_ctu_size_flag可等于1,则可结束编码树单元大小信令。否则,可以用信号通知use_128_ctu_size_flag。在一个实施例中,可首先用信号通知use_128_ctu_size_flag。如果use_128_ctu_size_flag可等于1,则可结束编码树单元大小信令。否则,可以用信号通知use_32_ctu_size_flag。在一个实施例中,可首先用信号通知use_64_ctu_size_flag。如果use_64_ctu_size_flag可等于1,

则可结束编码树单元大小信令。否则,可以用信号通知use_128_ctu_size_flag。在一个实施例中,可首先用信号通知use_128_ctu_size_flag。如果use_128_ctu_size_flag可等于1,则可结束编码树单元大小信令。否则,可以用信号通知use_64_ctu_size_flag。

[0042] 现在参考图3A至图3B,根据一个或多个实施例,序列参数集中的各个标志指示是否可以应用最小编码树单元大小(use_smallest_ctu_size_flag)以及是否可以应用最大编码树单元大小(use_largest_ctu_size_flag)。在一个实施例中,可以首先用信号通知用于指示是否可以应用最小编码树单元大小的序列参数集标志,如果不应用最小编码树单元大小,则可以用信号通知用于指示是否可以应用最大编码树单元大小的另一序列参数集标志。在另一实施例中,可以首先用信号通知用于指示是否可以应用最大编码树单元大小的序列参数集标志,如果不应用最大编码树单元大小,则可以用信号通知用于指示是否可以应用最小编码树单元大小的另一序列参数集标志。

[0043] use_smallest_ctu_size_flag等于1可以指示每个编码树单元的亮度编码树块大小可等于 32×32 。use_smallest_ctu_size_flag等于0可以指示use_largest_ctu_size_flag可存在。

[0044] use_largest_ctu_size_flag等于1可以指示每个编码树单元的亮度编码树块大小可等于 128×128 。use_largest_ctu_size_flag等于0可以指示每个编码树单元的亮度编码树块大小可等于 64×64 。

[0045] log2_min_luma_coding_block_size_minus2加2可指定最小亮度编码块大小。

[0046] 变量CtbLog2SizeY、CtbSizeY、MinCbLog2SizeY、MinCbSizeY、MinTbLog2SizeY、MaxTbLog2SizeY、MinTbSizeY、MaxTbSizeY、PicWidthInCtbsY、PicHeightInCtbsY、PicSizeInCtbsY、PicWidthInMinCbsY、PicHeightInMinCbsY、PicSizeInMinCbsY、PicSizeInSamplesY、PicWidthInSamplesC和PicHeightInSamplesC可以被推导为:if (use_smallest_ctu_size_flag)

[0047] CtbLog2SizeY=0

[0048] elseif (use_largest_ctu_size_flag)

[0049] CtbLog2SizeY=2

[0050] else

[0051] CtbLog2SizeY=1

[0052] CtbSizeY=1<<CtbLog2SizeY

[0053] 现在参考图3C,根据一个或多个实施例,仅当编码树单元大小大于或等于 64×64 时,才可以用信号通知sps_max_luma_transform_size_64_flag。在一个实施例中,如果use_32_ctu_size_flag可以首先用信号通知并且等于1,则可以不用信号通知sps_max_luma_transform_size_64_flag。在一个实施例中,如果use_64_ctu_size_flag可以首先用信号通知并且等于0,则可接着用信号通知use_32_ctu_size_flag并且use_32_ctu_size_flag可等于1,可以不用信号通知sps_max_luma_transform_size_64_flag。在一个实施例中,如果use_128_ctu_size_flag可以首先用信号通知并且等于0,则可接着用信号通知use_32_ctu_size_flag并且use_32_ctu_size_flag可等于1,可以不用信号通知sps_max_luma_transform_size_64_flag。

[0054] use_32_ctu_size_flag等于1可以指示每个编码树单元的亮度编码树块大小可等

于 32×32 。use_32_ctu_size_flag等于0可以指示use_128_ctu_size_flag可存在。use_128_ctu_size_flag等于1可以指示每个编码树单元的亮度编码树块大小可等于 128×128 。use_128_ctu_size_flag等于0可以指示每个编码树单元的亮度编码树块大小可等于 64×64 。

[0055] log2_min_luma_coding_block_size_minus2加2可以指示最小亮度编码块大小。

[0056] sps_max_luma_transform_size_64_flag等于1可以指示亮度样本中的最大变换大小可以等于64。sps_max_luma_transform_size_64_flag等于0可以指示亮度样本中的最大变换大小可以等于32。当不存在时,可以推断sps_max_luma_transform_size_64_flag的值等于0。

[0057] 变量CtbLog2SizeY、CtbSizeY、MinCbLog2SizeY、MinCbSizeY、MinTbLog2SizeY、MaxTbLog2SizeY、MinTbSizeY、MaxTbSizeY、PicWidthInCtbsY、PicHeightInCtbsY、PicSizeInCtbsY、PicWidthInMinCbsY、PicHeightInMinCbsY、PicSizeInMinCbsY、PicSizeInSamplesY、PicWidthInSamplesC和PicHeightInSamplesC可以被推导为:if (use_32_ctu_size_flag)

[0058] CtbLog2SizeY=0

[0059] elseif (use_128_ctu_size_flag)

[0060] CtbLog2SizeY=2

[0061] else

[0062] CtbLog2SizeY=1

[0063] CtbSizeY= $1 \ll \text{CtbLog2SizeY}$

[0064] 现在参考图3D,根据一个或多个实施例,仅当编码树单元大小可能不是最小编码树单元大小时,才可以用信号通知sps_max_luma_transform_size_64_flag。

[0065] use_smallest_ctu_size_flag等于1可以指示每个编码树单元的亮度编码树块大小可以等于 32×32 。use_smallest_ctu_size_flag等于0可以指示use_largest_ctu_size_flag存在。

[0066] use_largest_ctu_size_flag等于1可以指示每个编码树单元的亮度编码树块大小可等于 128×128 。use_largest_ctu_size_flag等于0可以指示每个编码树单元的亮度编码树块大小可等于 64×64 。

[0067] log2_min_luma_coding_block_size_minus2加2可以指示最小亮度编码块大小。

[0068] sps_max_luma_transform_size_64_flag等于1可以指示亮度样本中的最大变换大小可以等于64。sps_max_luma_transform_size_64_flag等于0可以指示亮度样本中的最大变换大小可以等于32。当不存在时,可以推断sps_max_luma_transform_size_64_flag的值等于0。

[0069] 变量CtbLog2SizeY、CtbSizeY、MinCbLog2SizeY、MinCbSizeY、MinTbLog2SizeY、MaxTbLog2SizeY、MinTbSizeY、MaxTbSizeY、PicWidthInCtbsY、PicHeightInCtbsY、PicSizeInCtbsY、PicWidthInMinCbsY、PicHeightInMinCbsY、PicSizeInMinCbsY、PicSizeInSamplesY、PicWidthInSamplesC和PicHeightInSamplesC可以被推导为:if (use_smallest_ctu_size_flag)

[0070] CtbLog2SizeY=0

[0071] elseif (use_largest_ctu_size_flag)

[0072] CtbLog2SizeY=2

[0073] else

[0074] CtbLog2SizeY=1

[0075] CtbSizeY=1<<CtbLog2SizeY

[0076] 现在参考图4,描绘了图示对视频数据进行编解码的方法400的步骤的操作流程图。在一些实施方式中,图4的一个或多个过程块可以由计算机102(图1)和服务器计算机114(图1)来执行。在一些实施方式中,图4的一个或多个过程块可以由另一个设备或一组设备来执行,所述另一个设备或一组设备与计算机102和服务器计算机114分离,或者包括计算机102和服务器计算机114。

[0077] 在402处,方法400包括接收具有编码树单元大小的视频数据。

[0078] 在404处,方法400包括通过设置两个或多个标志,来用信号通知与视频数据相关联的编码树单元大小。

[0079] 在一些实施例中,所述两个或多个标志包括32像素编码树单元大小标志、64像素编码树单元大小标志和128像素编码树单元大小标志当中的两个或多个。

[0080] 在406处,方法400包括基于所述与用信号通知的编码树单元大小相对应的标志对视频数据进行编解码。

[0081] 可以理解,图4仅提供了一种实施方案的图示,并不意味着对如何实现不同实施例的任何限制。可基于设计和实现需求对所描绘的环境进行许多修改。

[0082] 图5是根据示例性实施例的图1中描绘的计算机的内部组件和外部组件的框图500。应当理解,图5仅提供一种实施方案的图示,并不意味着对可实现不同实施例的环境的任何限制。可基于设计和实现需求对所描绘的环境进行许多修改。

[0083] 计算机102(图1)和服务器计算机114(图1)可以包括图4所示的相应的各组内部组件800A、800B和外部组件900A、900B。各组内部组件800中的每一个包括至少一个总线826上、至少一个操作系统828上以及至少一个计算机可读有形存储设备830上的至少一个处理器820、至少一个计算机可读RAM822以及至少一个计算机可读ROM824。

[0084] 处理器820以硬件、固件、或者硬件和软件的组合来实现。处理器820是中央处理单元(Central Processing Unit,CPU)、图形处理单元(Graphics Processing Unit,GPU)、加速处理单元(Accelerated Processing Unit,APU)、微处理器、微控制器、数字信号处理器(Digital Signal Processor,DSP)、现场可编程门阵列(Field-Programmable Gate Array,FPGA)、专用集成电路(Application-Specific Integrated Circuit ASIC)或另一种类型的处理部件。在一些实施方案中,处理器820包括能够被编程以执行函数的至少一个处理器。总线826包括允许在内部组件800A、800B之间进行通信的组件。

[0085] 服务器计算机114(图1)上的至少一个操作系统828、软件程序108(图1)和视频编解码程序116(图1)存储在至少一个相应的计算机可读有形存储设备830中,以由至少一个相应的处理器820经由至少一个相应的RAM822(其一般包括高速缓冲存储器)来执行。在图4所示的实施例中,每一个计算机可读有形存储设备830是内部硬盘驱动器的磁盘存储设备。可替代地,每个计算机可读有形存储设备830是半导体存储设备,诸如ROM824、EPROM、闪存、光盘、磁光盘、固态盘、压缩盘(Compact Disc,CD)、数字多功能盘(Digital Versatile

Disc,DVD)、软盘、盒式磁盘、磁带和/或可存储计算机程序和数字信息的另一种类型的非易失性计算机可读有形存储设备。

[0086] 每组内部组件800A、800B还包括R/W驱动器或接口832,以从至少一个便携式计算机可读有形存储设备936(例如CD-ROM、DVD、记忆棒、磁带、磁盘、光盘或半导体存储设备)读取或向其写入。软件程序(诸如软件程序108(图1)和视频编解码程序116(图1))可存储在至少一个相应的便携式计算机可读有形存储设备936上、经由相应的R/W驱动器或接口832进行读取并且被加载到相应的硬盘驱动器830中。

[0087] 每组内部组件800A、800B还包括网络适配器或接口836(例如TCP/IP适配卡)、无线Wi-Fi接口卡;或者3G、4G或5G无线接口卡或其它有线或无线通信链路。可经由网络(例如,因特网、局域网或其它广域网)和相应的网络适配器或接口836从外部计算机将服务器计算机114(图1)上的软件程序108(图1)和视频编解码程序116(图1)下载到计算机102(图1)。从网络适配器或接口836,服务器计算机114上的软件程序108和视频编解码程序116被加载到相应的硬盘驱动器830中。网络可以包括铜线、光纤、无线传输、路由器、防火墙、交换机、网关计算机和/或边缘服务器。

[0088] 每组外部组件900A、900B可以包括计算机显示监视器920、键盘930和计算机鼠标934。外部组件900A、900B还可以包括触摸屏、虚拟键盘、触摸板、定点设备和其它人机接口设备。每组内部组件800A、800B还包括设备驱动器840,以与计算机显示监视器920、键盘930和计算机鼠标934进行接口连接。设备驱动器840、R/W驱动器或接口832、以及网络适配器或接口836包括硬件和软件(存储在存储设备830和/或ROM824中)。

[0089] 应预先理解,尽管本公开包括对云计算的详细描述,但本文所引用的教导的实施方案不限于云计算环境。相反,一些实施例能够结合现在已知的或以后开发的任何其它类型的计算环境来实现。

[0090] 云计算是服务交付模型,该服务交付模型用于实现对可配置计算资源(例如,网络、网络带宽、服务器、处理、存储器、存储、应用、虚拟机和服务)的共享池的方便的、按需的网络访问,该可配置计算资源可以通过最少的管理工作或服务提供者的交互而快速地调配和释放。该云模型可以包括至少五个特性、至少三个服务模型和至少四个部署模型。

[0091] 特性如下:

[0092] 按需自助服务(On-demand self-service):云用户可根据需要自动地单方面提供计算功能,诸如服务器时间和网络存储,而无需与服务提供商进行人工交互。

[0093] 广泛的网络访问(Broad network access):功能可通过网络使用,并通过标准机制进行访问,该标准机制促进了异构瘦客户端平台或厚客户端平台(例如,移动电话、膝上型电脑和PDA)的使用。

[0094] 资源池化(Resource pooling):提供者的计算资源可以通过多租户模型集中起来,以为多个消费者提供服务,并根据需求动态分配和重新分配不同的物理和虚拟资源。一般情况下,消费者不能控制或并不知晓所提供的资源的确切位置,但可以能够指定较高抽象级别(例如,国家、州或数据中心)的位置,因此具有位置独立性。

[0095] 快速弹性(Rapid elasticity):在一些情况下,可以快速且弹性地调配功能,在某些情况下,自动地快速向外扩展和快速释放以快速向内扩展。对于用户来说,可用于调配的功能通常看起来是无限的,并且可在任何时间按任何数量购买。

[0096] 已测量服务 (Measured service): 云系统通过利用适于服务类型 (例如存储、处理、带宽和活跃用户帐号) 的某种抽象级别的计量能力, 自动地控制和优化资源使用。可监视、控制和报告资源使用, 从而为所使用服务的提供者和使用者双方提供透明度。

[0097] 服务模型如下:

[0098] 软件即服务 (Software as a Service, SaaS): 提供给消费者的功能是使用提供商在云基础架构上运行的应用程序。可通过诸如网络浏览器 (例如基于web的电子邮件) 之类的瘦客户端接口从各种客户端设备访问应用程序。除了可能会限制用户特定的应用配置设置外, 用户既不管理也不控制包括网络、服务器、操作系统、存储、或甚至单个应用程序功能的底层云基础架构。

[0099] 平台即服务 (Platform as a Service, PaaS): 提供给消费者的功能是将消费者创建的或获取的应用程序部署到云基础架构上, 这些应用程序是使用提供商支持的编程语言和工具创建的。用户既不管理也不控制包括网络、服务器、操作系统或存储在云的底层云基础架构, 但对已部署的应用程序具有控制权, 并且对应用托管环境配置可能也具有控制权。

[0100] 基础架构即服务 (Infrastructure as a Service, IaaS): 提供给消费者的功能是提供处理, 存储, 网络和其他基础计算资源, 使消费者能够在其中部署和运行任意软件, 所述任意软件包括操作系统和应用程序。用户既不管理也不控制底层云基础架构, 但是对操作系统、存储、已部署的应用程序具有控制权, 并且对选择网络部件 (例如, 主机防火墙) 可能具有有限的控制权。

[0101] 部署模型如下:

[0102] 私有云: 云基础架构单独为某个组织运行。它可由该组织或第三方管理, 并且可存在于内部 (on-premises) 或外部 (off-premises)。

[0103] 社区云: 云基础架构由若干组织共享, 并且支持具有共同关注点 (例如, 任务、安全要求、策略和合规性注意事项 (compliance considerations)) 的特定社区。它可由所述组织或第三方管理, 并且可存在于内部或外部。

[0104] 公共云: 云基础架构可供公众或大型产业群使用, 并由出售云服务的组织拥有。

[0105] 混合云: 云基础架构是由两个或更多个云 (私有、社区或公共) 组成的, 这些云仍然是唯一的实体, 但是通过标准化或专有技术将他们绑定在一起, 该标准化或专有技术实现数据和应用可移植性 (例如, 云间负载均衡的云爆炸)。

[0106] 云计算环境是面向焦点的服务, 该焦点为无状态、低耦合、模块化和语义互操作性。云计算的核心是包括相互连接的节点组成的网络的基础架构。

[0107] 参考图6, 描绘了示例性云计算环境600。如图所示, 云计算环境600包括至少一个云计算节点10, 云用户使用的本地计算设备可以与所述至少一个云计算节点进行通信, 该本地计算设备为诸如, 例如个人数字助理 (Personal Digital Assistant PDA) 或蜂窝电话54A、台式计算机54B、膝上型计算机54C, 和/或汽车计算机系统54N。云计算节点10可彼此通信。它们可在至少一个网络中进行物理地或虚拟地分组 (未示出), 所述网络为诸如如上所述的私有云、社区云、公共云或混合云或其组合。这允许云计算环境600提供基础架构即服务、平台即服务和/或软件即服务, 云消费者不需要维护本地计算设备上的资源。应当理解, 图5中所示的计算设备54A-N的类型仅旨在是示例性的, 并且云计算节点10和云计算环境600可通过任何类型的网络和/或网络可寻址连接 (例如, 使用web浏览器) 与任何类型的计

计算机化设备进行通信。

[0108] 参考图7,示出了由云计算环境700(图6)提供的一组功能抽象层700。应当预先理解,图7中所示的部件、层和功能仅仅是示例性的,并且实施例不限于此。如图所示,提供以下层和对应功能:

[0109] 硬件和软件层60包括硬件部件和软件部件。硬件部件的示例包括:主机61;基于精简指令集计算机(Reduced Instruction Set Computer,RISC)体系结构的服务器62;服务器63;刀片服务器64;存储设备65;以及网络和联网部件66。在一些实施例中,软件部件包括网络应用服务器软件67和数据库软件68。

[0110] 虚拟化层70提供抽象层,从该抽象层可提供虚拟实体的以下示例:虚拟服务器71;虚拟存储器72;虚拟网络73,包括虚拟专用网络;虚拟应用程序和操作系统74;以及虚拟客户端75。

[0111] 在一个示例中,管理层80可提供下面描述的功能。资源供应81提供计算资源和用于在云计算环境内执行任务的其它资源的动态获取。当在云计算环境中利用资源时,计量和定价82对资源的使用进行成本跟踪,并且为这些资源的消耗提供帐单和发票。在一个示例中,这些资源可以包括应用软件许可证。安全性为云用户和任务提供身份验证,以及对数据和其他资源的保护。用户门户83为用户和系统管理员提供对云计算环境的访问。服务级别管理84提供云计算资源分配和管理,从而满足所需的服务级别。服务水平协议(Service Level Agreement,SLA)规划和实现85为根据SLA预期未来需求的云计算资源提供了预安排和采购。

[0112] 工作负载层90提供可以利用云计算环境的功能的示例。可以从该层提供的工作负载和功能的示例包括:映射(mapping)及导航91;软件开发及生命周期管理92;虚拟课堂教学交付93;数据分析处理94;交易处理95;以及视频编解码96。视频编解码96可以使用单独的标志来代替编码树单元大小语法来对视频数据进行编码和解码。

[0113] 一些实施例可涉及在任何可能的技术细节集成水平的系统、方法和/或计算机可读介质。计算机可读介质可以包括计算机可读非易失性存储介质,计算机可读非易失性存储介质具有用于使处理器执行操作的计算机可读程序指令。

[0114] 计算机可读存储介质可为有形设备,有形设备可保存和存储由指令执行设备使用的指令。计算机可读存储介质可为,例如但不限于,电子存储设备、磁存储设备、光存储设备、电磁存储设备、半导体存储设备或前述的任何合适的组合。计算机可读存储介质的更具体示例的非穷尽列表包括以下内容:便携式计算机软盘、硬盘、随机存取存储器(Random Access Memory,RAM)、只读存储器(Read-Only Memory,ROM)、可擦除可编程只读存储器(Erasable Programmable Read-Only Memory,EPR0M或闪存)、静态随机存取存储器(Static Random Access Memory,SRAM)、便携式光盘只读存储器(Compact Disc Read-Only Memory,CD-ROM)、数字通用盘(Digital Versatile Disk,DVD)、记忆棒、软盘、机械编码装置,例如其上记录有指令的凹槽中的打孔卡片或凸起结构,以及前述的任何合适的组合。如在本文使用的,计算机可读存储介质不被解释为是瞬时信号本身,例如无线电波或其它自由传播的电磁波、通过波导或其它传输介质传播的电磁波(例如,通过光缆的光脉冲),或通过导线传输的电信号。

[0115] 可将本文描述的计算机可读程序指令从计算机可读存储介质下载到相应的计算/

处理设备,或者经由网络(例如,因特网、局域网、广域网和/或无线网络)下载到外部计算机或外部存储设备。网络可以包括铜传输缆线、光传输纤维、无线传输、路由器、防火墙、交换机、网关计算机和/或边缘服务器。每个计算/处理设备中的网络适配卡或网络接口从网络接收计算机可读程序指令,并且转发计算机可读程序指令以存储在相应的计算/处理设备内的计算机可读存储介质中。

[0116] 执行操作的计算机可读程序代码/指令可为汇编指令、指令集体系结构(Instruction-Set-Architecture, ISA)指令、机器指令、机器相关指令、微代码、固件指令、状态设置数据、集成电路的配置数据,或者以一种或多种编程语言的任何组合编写的源代码或目标代码,该编程语言包括面向对象的编程语言(诸如SmallTalk、C++等)以及过程编程语言(诸如“C”编程语言或类似的编程语言)。计算机可读程序指令可完全在用户的计算机上执行、部分地在用户的计算机上执行、作为独立软件包部分地在用户的计算机上并且部分地在远程计算机上、或者完全在远程计算机或服务器上执行。在后一种情况下,远程计算机可通过任何类型的网络(包括局域网(Local Area Network, LAN)或广域网(Wide Area Network, WAN))连接到用户的计算机,或者可连接到外部计算机(例如,通过使用因特网服务提供商的因特网)。在一些实施例中,包括(例如)可编程逻辑电路、现场可编程门阵列(Field-Programmable Gate Arrays, FPGA)或可编程逻辑阵列(Programmable Logic Arrays, PLA)的电子电路可通过利用计算机可读程序指令的状态信息来执行计算机可读程序指令以使电子电路个性化,进而执行本公开的各方面或操作。

[0117] 可以将这些计算机可读程序指令提供给通用计算机、专用计算机或其它可编程数据处理装置的处理器以产生机器,使得经由计算机或其它可编程数据处理装置的处理器执行的指令,创建用于实现流程图和/或框图的框或多个框中指定的功能/动作的方法。这些计算机可读程序指令还可被存储在计算机可读存储介质中,该计算机可读存储介质可指导计算机、可编程数据处理装置和/或其它设备以特定方式运行,以使存储有指令的计算机可读存储介质包括这样的制造品,该制造品包括实现在流程图和/或框图的框或多个框中指定的功能/操作的各方面的指令。

[0118] 还可以将计算机可读程序指令加载到计算机、其他可编程数据处理装置或其他设备上,以在计算机、其他可编程装置或其他设备上执行一系列操作步骤,以产生计算机实现的处理,使得在计算机、其他可编程装置或其他设备上执行的指令实现在流程图和/或框图的框或多个框中指定的功能/操作。

[0119] 附图中的流程图和框图图示了根据各种实施例的系统、方法和计算机可读介质的可能实现的体系结构、功能和操作。在这点上,流程图或框图中的每个框可表示指令的模块、片段或部分,该指令包括用于实现指定的逻辑功能的至少一个可执行指令。方法、计算机系统和计算机可读介质可以包括附加框、更少的框、不同的框,或者与附图中所描绘的那些框不同布置的框。在一些可替换实施方案中,在框中指出的功能可以不按照附图中指出的顺序执行。例如,连续示出的两个框实际上可以同时或基本上同时执行,或者这些框有时可以以相反的顺序执行,这取决于所涉及的功能。还应注意,框图和/或流程图图示中的每个框,以及框图和/或流程图图示中的框的组合可通过基于硬件的专用系统来实现,所述基于硬件的专用系统执行指定功能或操作、或者执行专用硬件和计算机指令的组合。

[0120] 显然,本文描述的系统和/或方法可以以不同形式的硬件、固件、或硬件和软件的

组合来实现。实现这些系统和/或方法的实际专用控制硬件或软件代码不限于这些实施方案。因此,在不参考特定软件代码的情况下在本文描述了系统和/或方法的操作和行为——应当理解,可基于在本文的描述来设计软件和硬件以实现系统和/或方法。

[0121] 本文中使用的元件、动作或指令都不应被解释为关键的或必要的,除非明确地如此描述。此外,如本文所用,冠词“一种(a)”和“一种(an)”旨在包括至少一个项目,并且可与“至少一个”互换使用。此外,如本文所用,术语“组”旨在包括至少一个项目(例如,相关项目、不相关项目、相关项目和不相关项目的组合等),并且可与“至少一个”互换使用。当仅意指一个项目时,使用术语“一个”或类似语言。此外,如本文所用,术语“具有(has)”、“具有(have)”、“具有(having)”等旨在是开放式术语。进一步地,短语“根据”旨在表示“至少部分地根据”,除非另外明确声明。

[0122] 已出于图示的目的呈现了对各个方面和实施例的描述,但该描述并非旨在是详尽的或限于所公开的实施例。虽然特征的组合在权利要求中引用和/或在说明书中公开,但是这些组合并不旨在对可能的实施方案的公开进行限制。事实上,这些特征中的许多特征可以以权利要求中未具体列举和/或说明书中未公开的方式组合。虽然下面列出的每个从属权利要求可直接从属于仅一个权利要求,但是可能的实施方式的公开包括与权利要求组中的每个其它权利要求进行组合的每个从属权利要求。在不脱离所描述的实施例的范围的情况下,许多修改和变化对于本领域普通技术人员是显而易见的。本文使用的术语被选择为最佳地解释实施例的原理、实际应用或相对于市场中发现的技术的改进,或者使得本领域普通技术人员能够理解本文公开的实施例。

100

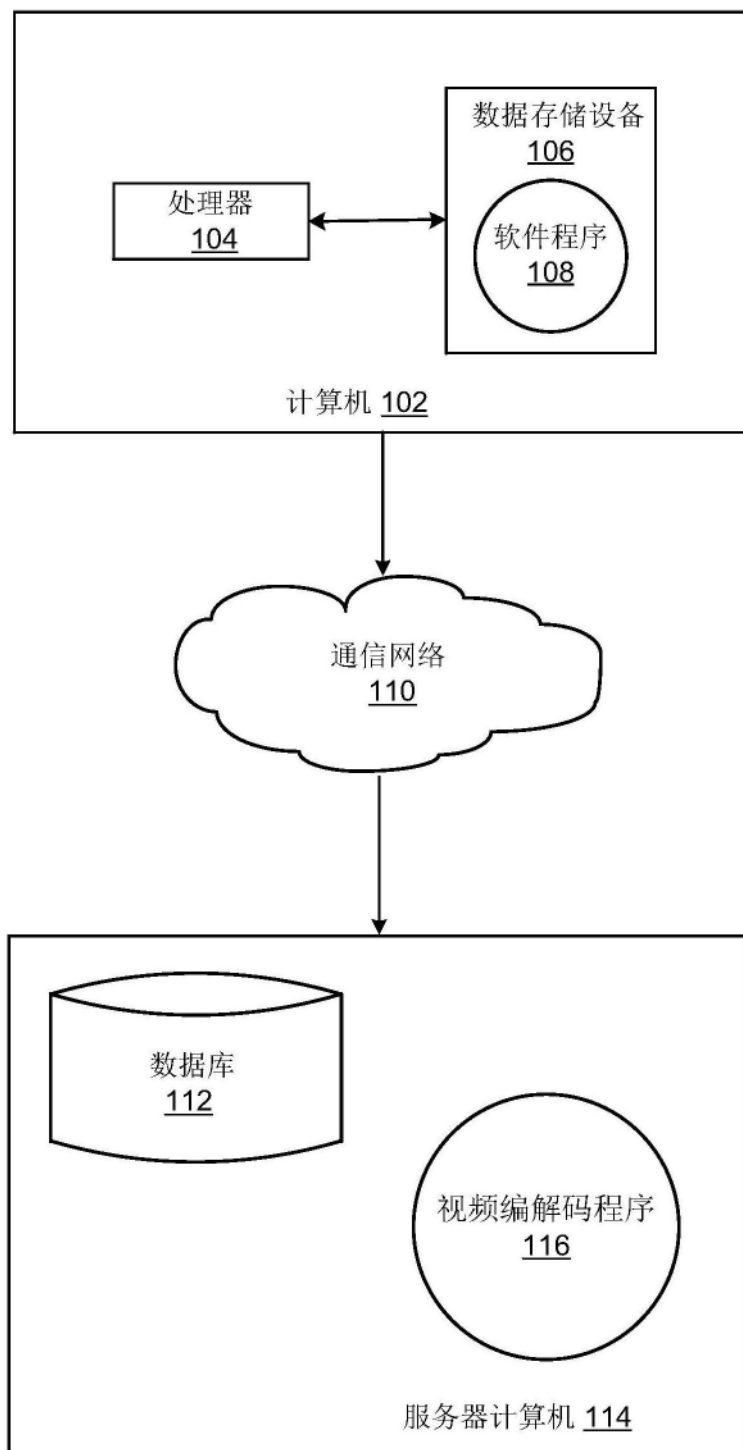


图1

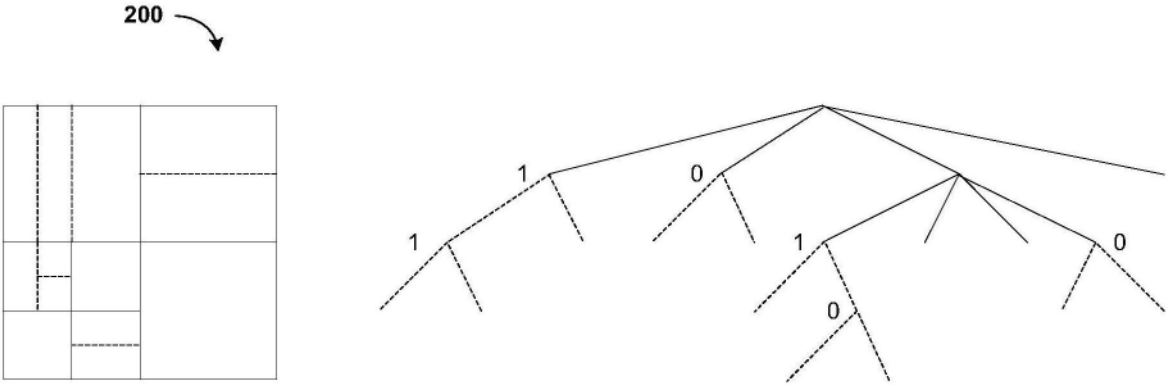


图2

300A

qtbtt_dual_tree_intra_flag	u(1)
use_smallest_ctu_size_flag	u(1)
if (!use_smallest_ctu_size_flag)	
use_largest_ctu_size_flag	u(1)
log2_min_luma_coding_block_size_minus2	ue(v)
...	

图3A

300B

qtbtt_dual_tree_intra_flag	u(1)
use_largest_ctu_size_flag	u(1)
if (!use_largest_ctu_size_flag)	
use_smallest_ctu_size_flag	u(1)
log2_min_luma_coding_block_size_minus2	ue(v)
...	

图3B

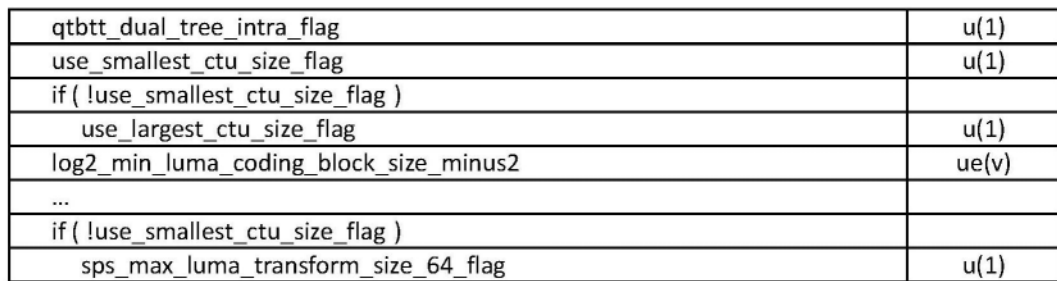
300C



qtbtt_dual_tree_intra_flag	u(1)
use_32_ctu_size_flag	u(1)
if (!use_32_ctu_size_flag)	
use_128_ctu_size_flag	u(1)
log2_min_luma_coding_block_size_minus2	ue(v)
...	
if (!use_32_ctu_size_flag)	
sps_max_luma_transform_size_64_flag	u(1)

图3C

300D



qtbtt_dual_tree_intra_flag	u(1)
use_smallest_ctu_size_flag	u(1)
if (!use_smallest_ctu_size_flag)	
use_largest_ctu_size_flag	u(1)
log2_min_luma_coding_block_size_minus2	ue(v)
...	
if (!use_smallest_ctu_size_flag)	
sps_max_luma_transform_size_64_flag	u(1)

图3D

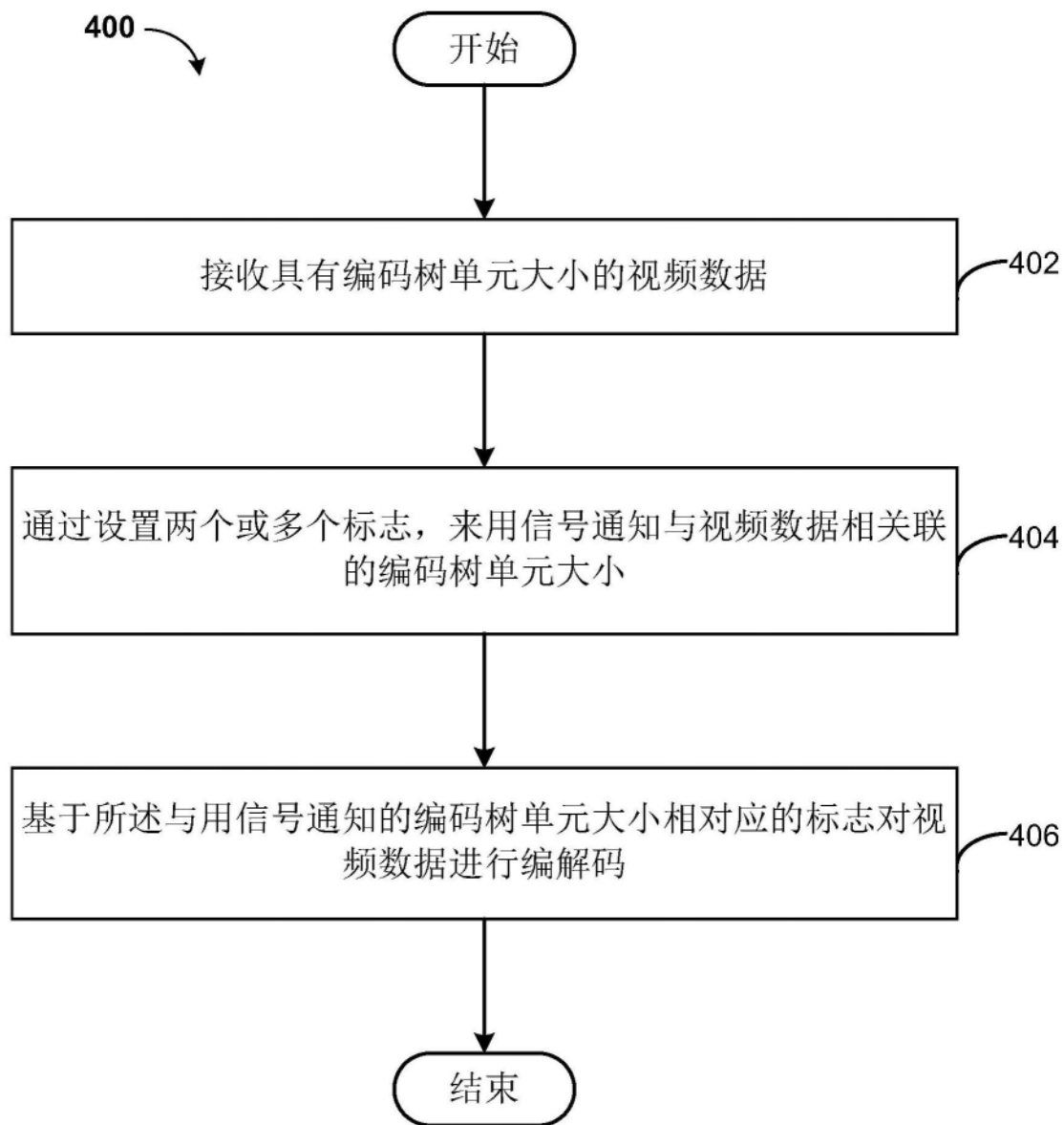


图4

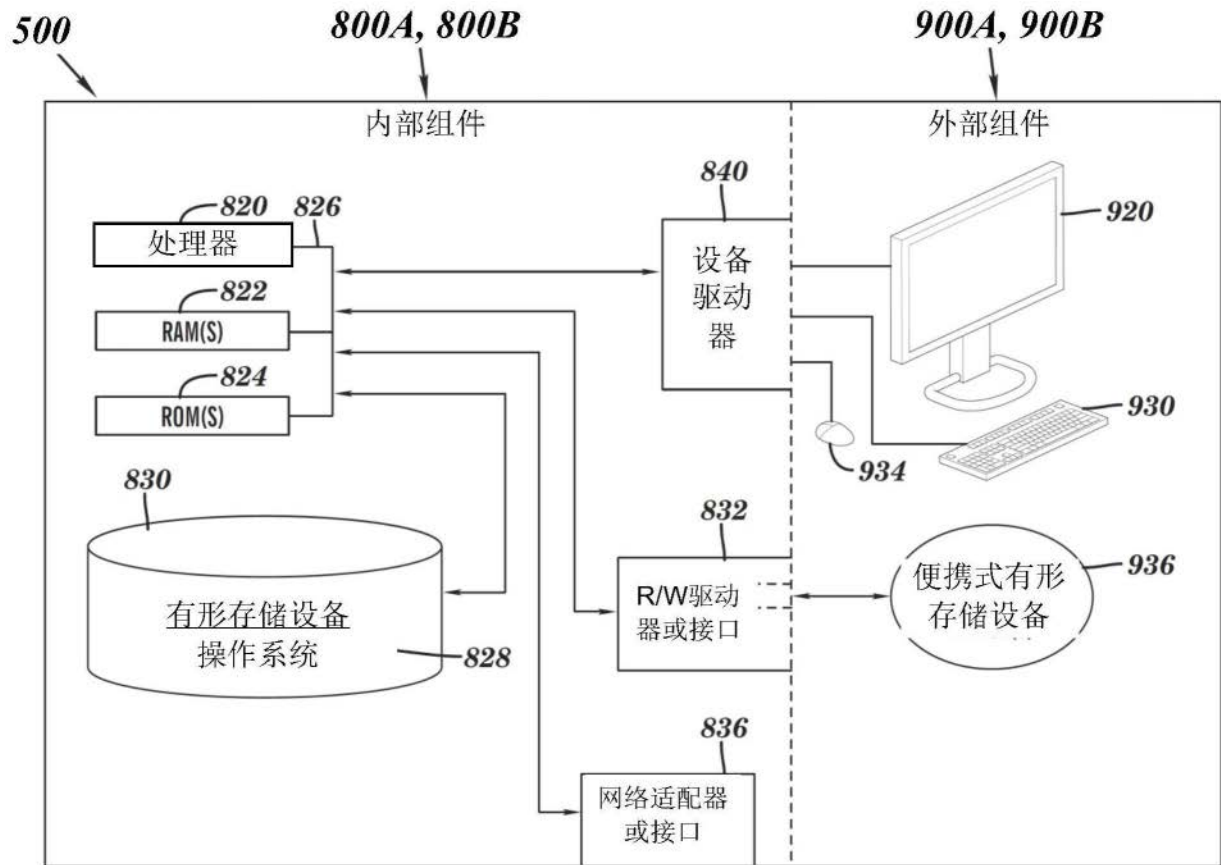


图5

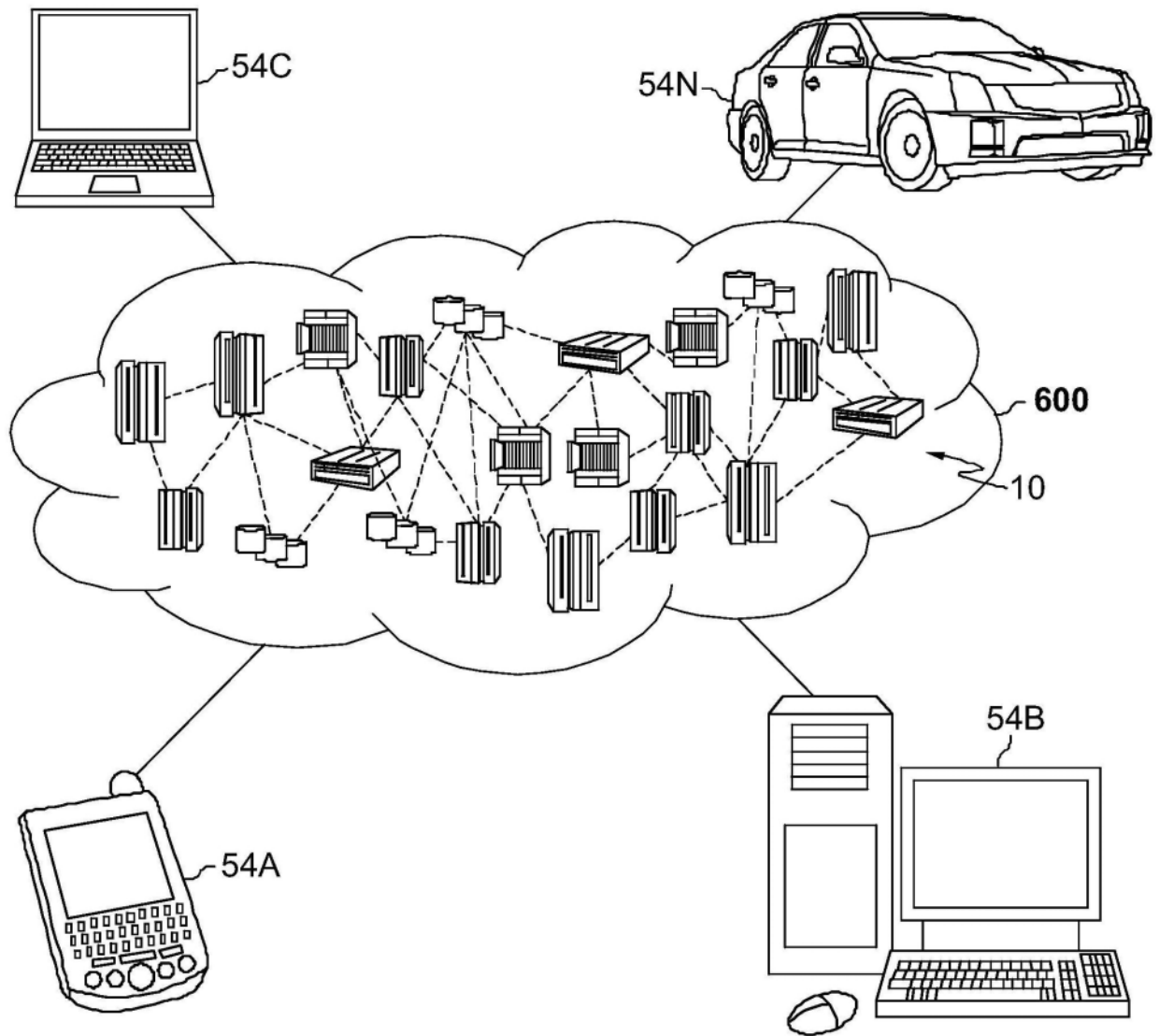


图6

700

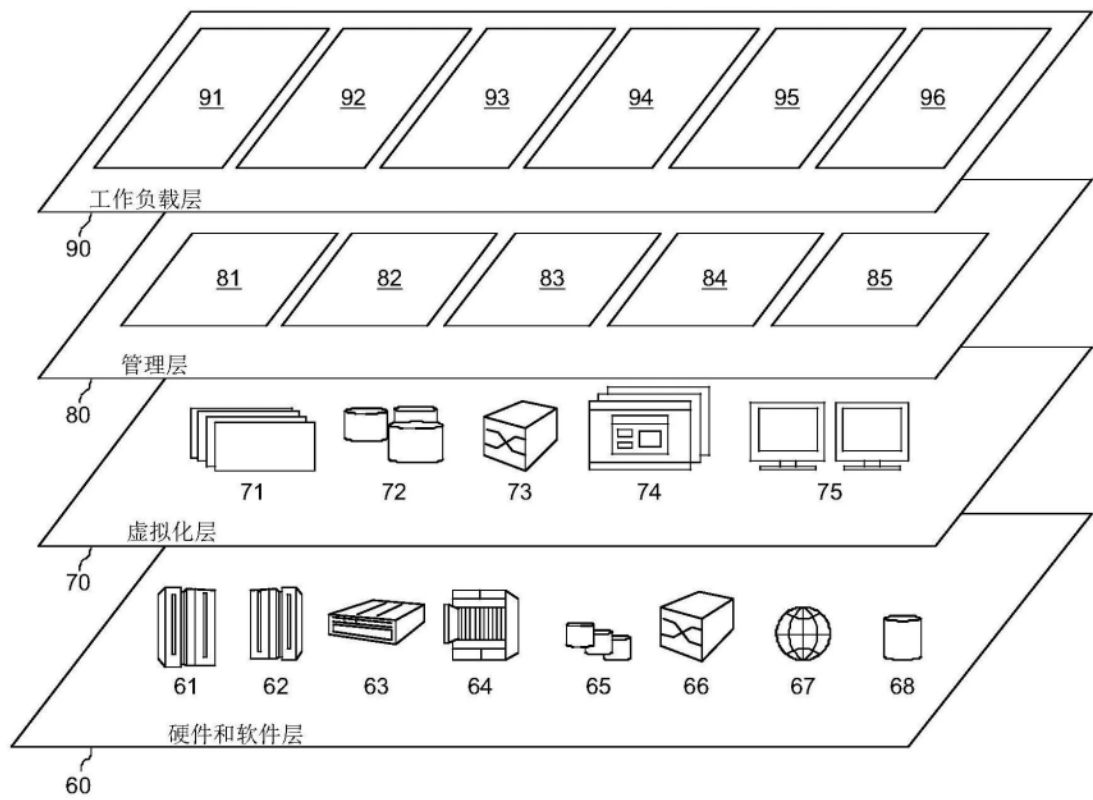


图7