



(12)发明专利

(10)授权公告号 CN 106464885 B

(45)授权公告日 2020.03.13

(21)申请号 201580032098.1

(22)申请日 2015.06.19

(65)同一申请的已公布的文献号
申请公布号 CN 106464885 A

(43)申请公布日 2017.02.22

(30)优先权数据
62/015,347 2014.06.20 US
62/062,797 2014.10.10 US
14/743,776 2015.06.18 US

(85)PCT国际申请进入国家阶段日
2016.12.15

(86)PCT国际申请的申请数据
PCT/US2015/036769 2015.06.19

(87)PCT国际申请的公布数据
WO2015/196126 EN 2015.12.23

(73)专利权人 高通股份有限公司
地址 美国加利福尼亚州

(72)发明人 张莉 陈建乐
霍埃尔·索赖·罗哈斯
马尔塔·卡切维奇

(74)专利代理机构 北京律盟知识产权代理有限公司 11287

代理人 宋献涛

(51)Int.Cl.
H04N 19/176(2014.01)
H04N 19/70(2014.01)
H04N 19/124(2014.01)
H04N 19/186(2014.01)
H04N 19/157(2014.01)

(56)对比文件
CN 102595130 A,2012.07.18,
CN 103329539 A,2013.09.25,
CN 102257806 A,2011.11.23,
US 2005259730 A1,2005.11.24,
US 2010215097 A1,2010.08.26,
ZHANG L ET AL.AhG8:In-loop color-
space transform.《JOINT COLLABORATIVE TEAM
ON VIDEO CODING OF ISO/IEC JTC1/SC29/WG11
AND ITU-T SG.16》.2014,

审查员 黄驰

权利要求书7页 说明书59页 附图9页

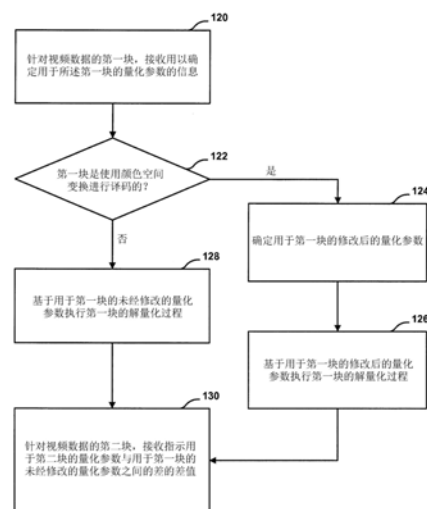
(54)发明名称

块自适应颜色空间转换译码

(57)摘要

一种用于对视频数据进行解码的装置,其包含:存储器,其经配置以存储视频数据;一或多个处理器,其经配置以:接收所述视频数据的第一块;确定用于所述第一块的量化参数;响应于确定所述第一块是使用用于所述第一块的残余数据的颜色空间变换模式进行译码的,修改用于所述第一块的所述量化参数;基于用于所述第一块的所述修改后的量化参数对所述第一块执行解量化过程;接收所述视频数据的第二块;接收指示用于所述第二块的量化参数与用于所述第一块的所述量化参数之间的差的差值;基于所述接收到的差值及用于所述第一块的所述量化参数

确定用于所述第二块的所述量化参数;以及基于所述所确定的量化参数对所述第二块进行解码。



1. 一种对视频数据进行解码的方法,所述方法包括:

接收所述视频数据的第一译码单元,其中所述第一译码单元包括第一亮度分量和第一色度分量;

接收用以确定用于所述第一译码单元的所述第一亮度分量的亮度-色度YCbCr颜色空间量化参数的信息;

响应于确定所述第一译码单元是使用用于所述第一译码单元的残余数据的颜色空间变换模式进行译码的,修改用于所述第一译码单元的所述第一亮度分量的所述YCbCr颜色空间量化参数以确定用于所述第一译码单元的红、绿和蓝RGB颜色空间量化参数;

基于用于所述第一译码单元的所述RGB颜色空间量化参数对所述第一译码单元执行解量化过程;

接收所述视频数据的第二译码单元,其中所述第二译码单元包括第二亮度分量和第二色度分量,其中所述第二译码单元在解码次序上在所述第一译码单元之后,且其中所述第二译码单元是不同于所述第一译码单元的译码单元;

针对所述第二译码单元的所述第二亮度分量,接收指示用于所述第二译码单元的所述第二亮度分量的YCbCr颜色空间量化参数与用于所述第一译码单元的所述第一亮度分量的未经修改的所述YCbCr颜色空间量化参数之间的差的差值;

基于所述接收到的差值及用于所述第一译码单元的所述第一亮度分量的未经修改的所述YCbCr颜色空间量化参数确定用于所述第二译码单元的所述第二亮度分量的所述YCbCr颜色空间量化参数;

基于所述所确定的用于所述第二译码单元的所述第二亮度分量的YCbCr颜色空间量化参数对所述第二译码单元进行解码;以及

输出包含所述第一译码单元和所述第二译码单元的经解码视频数据。

2. 根据权利要求1所述的方法,其进一步包括:

响应于确定对所述第二译码单元启用所述颜色空间变换模式,修改用于所述第二译码单元的所述第二亮度分量的所述YCbCr颜色空间量化参数以确定用于所述第二译码单元的RGB颜色空间量化参数,

其中基于用于所述第二译码单元的所述第二亮度分量的所述YCbCr颜色空间量化参数对所述第二译码单元进行解码包括:

基于用于所述第二译码单元的所述RGB颜色空间量化参数对所述第二译码单元执行解量化过程。

3. 根据权利要求1所述的方法,其中基于用于所述第二译码单元的所述第二亮度分量的所述YCbCr颜色空间量化参数对所述第二译码单元进行解码包括:

响应于确定对所述第二译码单元停用所述颜色空间变换模式,基于用于所述第二译码单元的所述第二亮度分量的所述YCbCr颜色空间量化参数对所述第二译码单元执行解量化过程。

4. 根据权利要求1所述的方法,其进一步包括:

接收用于所述第一译码单元的旗标以确定所述视频数据的第一译码单元是使用用于所述第一译码单元的残余数据的所述颜色空间变换模式进行译码的。

5. 根据权利要求1所述的方法,其中接收用以确定用于所述第一译码单元的所述第一

亮度分量的所述YCbCr颜色空间量化参数的信息包括接收用于所述第一译码单元的所述第一亮度分量的所述YCbCr颜色空间量化参数的初始值。

6. 根据权利要求5所述的方法, 其中接收用于所述YCbCr颜色空间量化参数的所述初始值包括:

在切片级接收所述YCbCr颜色空间量化参数的所述初始值。

7. 根据权利要求1所述的方法, 其进一步包括:

在经译码单元级接收指示用于所述第二译码单元的所述第二亮度分量的所述YCbCr颜色空间量化参数与用于所述第一译码单元的所述第一亮度分量的未经修改的所述YCbCr颜色空间量化参数之间的所述差的所述差值。

8. 根据权利要求1所述的方法, 其中接收指示用于所述第二译码单元的所述第二亮度分量的所述YCbCr颜色空间量化参数与用于所述第一译码单元的所述第一亮度分量的未经修改的所述YCbCr颜色空间量化参数之间的所述差的所述差值包括接收指示所述差的绝对值的语法元素以及接收指示所述差的正负号的语法元素。

9. 根据权利要求1所述的方法, 其进一步包括:

基于用于所述第一译码单元的所述RGB颜色空间量化参数确定解块滤波过程的边界强度参数; 以及

对所述第一译码单元执行所述解块滤波过程。

10. 一种对视频数据进行编码的方法, 所述方法包括:

确定用于所述视频数据的第一译码单元的第一亮度分量的亮度-色度YCbCr颜色空间量化参数, 其中所述第一译码单元包括所述第一亮度分量和第一色度分量;

响应于确定所述视频数据的第一译码单元是使用用于所述第一译码单元的残余数据的颜色空间变换模式进行译码的, 修改用于所述第一译码单元的所述第一亮度分量所述YCbCr颜色空间量化参数以确定用于所述第一译码单元的红、绿和蓝RGB颜色空间量化参数;

基于用于所述第一译码单元的所述RGB颜色空间量化参数执行所述第一译码单元的量化过程;

确定用于所述视频数据的第二译码单元的第二亮度分量的YCbCr颜色空间量化参数, 其中所述第二译码单元包括第二亮度分量和第二色度分量, 其中所述第二译码单元在解码次序上在所述第一译码单元之后, 且其中所述第二译码单元是不同于所述第一译码单元的译码单元;

用信号在经编码视频数据的位流中发送用于所述第一译码单元的所述第一亮度分量的未经修改的所述YCbCr颜色空间量化参数与用于所述第二译码单元的所述第二亮度分量的所述YCbCr颜色空间量化参数之间的差值; 以及

输出所述经编码视频数据的位流。

11. 根据权利要求10所述的方法, 其进一步包括:

响应于确定对所述视频数据的第二译码单元启用颜色空间变换模式, 修改用于所述第二译码单元的所述第二亮度分量的所述YCbCr颜色空间量化参数以确定用于所述第二译码单元的RGB颜色空间量化参数;

基于用于所述第二译码单元的所述RGB量化参数对所述第二译码单元执行量化过程。

12. 根据权利要求10所述的方法,其进一步包括:

响应于确定对所述视频数据的第二译码单元停用颜色空间变换模式,基于用于所述第二译码单元的所述第二亮度分量的所述YCbCr颜色空间量化参数对所述第二译码单元执行量化过程。

13. 根据权利要求10所述的方法,其进一步包括:

产生用于所述第一译码单元的旗标以指示所述视频数据的第一译码单元是否是使用用于所述第一译码单元的残余数据的所述颜色空间变换模式进行译码的。

14. 根据权利要求10所述的方法,其中确定用于所述视频数据的第一译码单元的所述第一亮度分量的所述YCbCr颜色空间量化参数包括确定用于所述第一译码单元的所述第一亮度分量的所述YCbCr颜色空间量化参数的初始值,所述方法进一步包括:

在包括所述第一译码单元的切片的切片标头中用信号发送用于所述YCbCr颜色空间量化参数的所述初始值。

15. 根据权利要求10所述的方法,其进一步包括:

在经译码单元级用信号发送指示用于所述第二译码单元的所述第二亮度分量的所述YCbCr颜色空间量化参数与用于所述第一译码单元的所述第一亮度分量的未经修改的所述YCbCr颜色空间量化参数之间的所述差的所述差值。

16. 根据权利要求10所述的方法,其中用信号发送指示用于所述第二译码单元的所述第二亮度分量的所述YCbCr颜色空间量化参数与用于所述第一译码单元的所述第一亮度分量的未经修改的所述YCbCr颜色空间量化参数之间的所述差的所述差值包括产生指示所述差的绝对值的语法元素以及产生指示所述差的正负号的语法元素。

17. 根据权利要求10所述的方法,其进一步包括:

基于用于所述第一译码单元的所述RGB颜色空间量化参数确定解块滤波过程的边界强度参数;以及

对所述第一译码单元执行所述解块滤波过程。

18. 一种用于对视频数据进行解码的装置,所述装置包括:

存储器,其经配置以存储视频数据;

一或多个处理器,其经配置以:

接收所述视频数据的第一译码单元,其中所述第一译码单元包括第一亮度分量和第一色度分量;

接收用以确定用于所述第一译码单元的所述第一亮度分量的亮度-色度YCbCr颜色空间量化参数的信息;

响应于确定所述第一译码单元是使用用于所述第一译码单元的残余数据的颜色空间变换模式进行译码的,修改用于所述第一译码单元的所述第一亮度分量的所述YCbCr颜色空间量化参数以确定用于所述第一译码单元的红、绿和蓝RGB颜色空间量化参数;

基于用于所述第一译码单元的所述RGB颜色空间量化参数对所述第一译码单元执行解量化过程;

接收所述视频数据的第二译码单元,其中所述第二译码单元包括第二亮度分量和第二色度分量,其中所述第二译码单元在解码次序上在所述第一译码单元之后,且其中所述第二译码单元是不同于所述第一译码单元的译码单元;

针对所述第二译码单元的所述第二亮度分量,接收指示用于所述第二译码单元的所述第二亮度分量的YCbCr颜色空间量化参数与用于所述第一译码单元的所述第一亮度分量的未经修改的所述YCbCr颜色空间量化参数之间的差的差值;

基于所述接收到的差值及用于所述第一译码单元的所述第一亮度分量的未经修改的所述YCbCr颜色空间量化参数确定用于所述第二译码单元的所述第二亮度分量的所述YCbCr颜色空间量化参数;

基于所述所确定的用于所述第二译码单元的所述第二亮度分量的YCbCr颜色空间量化参数对所述第二译码单元进行解码;以及

输出包含所述第一译码单元和所述第二译码单元的经解码视频数据。

19. 根据权利要求18所述的装置,其中所述一或多个处理器进一步经配置以:

响应于确定对所述视频数据的第二译码单元启用所述颜色空间变换模式,修改用于所述第二译码单元的所述第二亮度分量的所述YCbCr颜色空间量化参数以确定用于所述第二译码单元的RGB颜色空间量化参数,

其中为了基于用于所述第二译码单元的所述第二亮度分量的所述YCbCr颜色空间量化参数对所述第二译码单元进行解码,所述一或多个处理器基于用于所述第二译码单元的所述RGB颜色空间量化参数对所述第二译码单元执行解量化过程。

20. 根据权利要求18所述的装置,其中为了基于用于所述第二译码单元的所述第二亮度分量的所述YCbCr颜色空间量化参数对所述第二译码单元进行解码,所述一或多个处理器经配置以:

响应于确定对所述第二译码单元停用所述颜色空间变换模式,基于用于所述第二译码单元的所述第二亮度分量的所述YCbCr颜色空间量化参数对所述第二译码单元执行解量化过程。

21. 根据权利要求18所述的装置,其中所述一或多个处理器进一步经配置以:

接收用于所述第一译码单元的旗标以确定所述视频数据的第一译码单元是使用用于所述第一译码单元的残余数据的所述颜色空间变换模式进行译码的。

22. 根据权利要求18所述的装置,其中为了接收用以确定用于所述第一译码单元的所述第一亮度分量的所述YCbCr颜色空间量化参数的信息,所述一或多个处理器接收用于所述第一译码单元的所述第一亮度分量的所述YCbCr颜色空间量化参数的初始值。

23. 根据权利要求22所述的装置,其中为了接收用于所述YCbCr颜色空间量化参数的所述初始值,所述一或多个处理器在切片级接收所述YCbCr颜色空间量化参数的所述初始值。

24. 根据权利要求18所述的装置,其中所述一或多个处理器进一步经配置以:

在经译码单元级接收指示用于所述第二译码单元的所述第二亮度分量的所述YCbCr颜色空间量化参数与用于所述第一译码单元的所述第一亮度分量的未经修改的所述YCbCr颜色空间量化参数之间的所述差的所述差值。

25. 根据权利要求18所述的装置,其中为了接收指示用于所述第二译码单元的所述第二亮度分量的所述YCbCr颜色空间量化参数与用于所述第一译码单元的所述第一亮度分量的未经修改的所述YCbCr颜色空间量化参数之间的所述差的所述差值,所述一或多个处理器接收指示所述差的绝对值的语法元素以及接收指示所述差的正负号的语法元素。

26. 根据权利要求18所述的装置,其中所述一或多个处理器进一步经配置以:

基于用于所述第一译码单元的所述RGB颜色空间量化参数确定解块滤波过程的边界强度参数;以及

对所述第一译码单元执行所述解块滤波过程。

27. 根据权利要求18所述的装置,其中所述装置包括以下各项中的一者:

微处理器;

集成电路IC;以及

无线通信装置,其包括视频解码器。

28. 一种用于对视频数据进行编码的装置,所述装置包括:

存储器,其经配置以存储视频数据;

一或多个处理器,其经配置以:

确定用于所述视频数据的第一译码单元的第一亮度分量的亮度-色度YCbCr颜色空间量化参数,其中所述第一译码单元包括所述第一亮度分量和第一色度分量;

响应于确定所述视频数据的第一译码单元是使用用于所述第一译码单元的残余数据的颜色空间变换模式进行译码的,修改用于所述第一译码单元的所述第一亮度分量所述YCbCr颜色空间量化参数以确定用于所述第一译码单元的红、绿和蓝RGB颜色空间量化参数;

基于用于所述第一译码单元的所述RGB颜色空间量化参数执行所述第一译码单元的量化过程;

确定用于所述视频数据的第二译码单元的第二亮度分量的YCbCr颜色空间量化参数,其中所述第二译码单元包括第二亮度分量和第二色度分量,其中所述第二译码单元在解码次序上在所述第一译码单元之后,且其中所述第二译码单元是不同于所述第一译码单元的译码单元;

用信号在经编码视频数据的位流中发送用于所述第一译码单元的所述第一亮度分量的未经修改的所述YCbCr颜色空间量化参数与用于所述第二译码单元的所述第二亮度分量的所述YCbCr颜色空间量化参数之间的差值;以及

输出所述经编码视频数据的位流。

29. 根据权利要求28所述的装置,其中所述一或多个处理器进一步经配置以:

响应于确定对所述视频数据的第二译码单元启用颜色空间变换模式,修改用于所述第二译码单元的所述第二亮度分量的所述YCbCr颜色空间量化参数以确定用于所述第二译码单元的RGB颜色空间量化参数;

基于用于所述第二译码单元的所述RGB量化参数对所述第二译码单元执行量化过程。

30. 根据权利要求28所述的装置,其中所述一或多个处理器进一步经配置以

响应于确定对所述视频数据的第二译码单元停用颜色空间变换模式,基于用于所述第二译码单元的所述第二亮度分量的所述YCbCr颜色空间量化参数对所述第二译码单元执行量化过程。

31. 根据权利要求28所述的装置,其中所述一或多个处理器进一步经配置以:

产生用于所述第一译码单元的旗标以指示所述视频数据的第一译码单元是否是使用用于所述第一译码单元的残余数据的所述颜色空间变换模式进行译码的。

32. 根据权利要求28所述的装置,其中确定用于所述视频数据的第一译码单元的所述

第一亮度分量的所述YCbCr颜色空间量化参数包括确定用于所述第一译码单元的所述第一亮度分量的所述YCbCr颜色空间量化参数的初始值,其中所述一或多个处理器进一步经配置以在包括所述第一译码单元的切片的切片标头中用信号发送用于所述YCbCr颜色空间量化参数的所述初始值。

33. 根据权利要求28所述的装置,其中所述一或多个处理器进一步经配置以

在经译码单元级用信号发送指示用于所述第二译码单元的所述第二亮度分量的所述YCbCr颜色空间量化参数与用于所述第一译码单元的所述第一亮度分量的未经修改的所述YCbCr颜色空间量化参数之间的所述差的所述差值。

34. 根据权利要求28所述的装置,其中为了用信号发送指示用于所述第二译码单元的所述第二亮度分量的所述YCbCr颜色空间量化参数与用于所述第一译码单元的所述第一亮度分量的未经修改的所述YCbCr颜色空间量化参数之间的所述差的所述差值,所述一或多个处理器进一步经配置以产生指示所述差的绝对值的语法元素以及产生指示所述差的正负号的语法元素。

35. 根据权利要求28所述的装置,其中所述一或多个处理器进一步经配置以:

基于用于所述第一译码单元的所述RGB颜色空间量化参数确定解块滤波过程的边界强度参数;以及

对所述第一译码单元执行所述解块滤波过程。

36. 根据权利要求28所述的装置,其中所述装置包括以下各项中的至少一者:

微处理器;

集成电路IC;或

无线通信装置,其包括视频编码器。

37. 一种用于对视频进行解码的设备,所述设备包括:

用于接收视频数据的第一译码单元的装置,其中所述第一译码单元包括第一亮度分量和第一色度分量;

用于接收用以确定用于所述第一译码单元的所述第一亮度分量的亮度-色度YCbCr颜色空间量化参数的信息的装置;

用于响应于确定所述第一译码单元是使用用于所述第一译码单元的残余数据的颜色空间变换模式进行译码的而修改用于所述第一译码单元的所述第一亮度分量的所述YCbCr颜色空间量化参数以确定用于所述第一译码单元的红、绿和蓝RGB颜色空间量化参数的装置;

用于基于用于所述第一译码单元的所述RGB颜色空间量化参数对所述第一译码单元执行解量化过程的装置;

用于接收所述视频数据的第二译码单元的装置,其中所述第二译码单元包括第二亮度分量和第二色度分量,其中所述第二译码单元在解码次序上在所述第一译码单元之后,且其中所述第二译码单元是不同于所述第一译码单元的译码单元;

用于针对所述第二译码单元的所述第二亮度分量而接收指示用于所述第二译码单元的所述第二亮度分量的YCbCr颜色空间量化参数与用于所述第一译码单元的所述第一亮度分量的未经修改的所述YCbCr颜色空间量化参数之间的差的差值的装置;

用于基于所述接收到的差值及用于所述第一译码单元的所述第一亮度分量的未经修

改的所述YCbCr颜色空间量化参数确定用于所述第二译码单元的所述第二亮度分量的所述YCbCr颜色空间量化参数的装置；

用于基于所述所确定的用于所述第二译码单元的所述第二亮度分量的YCbCr颜色空间量化参数对所述第二译码单元进行解码的装置；以及

用于输出包含所述第一译码单元和所述第二译码单元的经解码视频数据的装置。

38. 一种存储指令的非暂时性计算机可读存储媒体，所述指令在由一或多个处理器执行时致使所述一或多个处理器：

接收视频数据的第一译码单元，其中所述第一译码单元包括第一亮度分量和第一色度分量；

接收用以确定用于所述第一译码单元的所述第一亮度分量的亮度-色度YCbCr颜色空间量化参数的信息；

响应于确定所述第一译码单元是使用用于所述第一译码单元的残余数据的颜色空间变换模式进行译码的，修改用于所述第一译码单元的所述第一亮度分量的所述YCbCr颜色空间量化参数以确定用于所述第一译码单元的红、绿和蓝RGB颜色空间量化参数；

基于用于所述第一译码单元的所述RGB颜色空间量化参数对所述第一译码单元执行解量化过程；

接收所述视频数据的第二译码单元，其中所述第二译码单元包括第二亮度分量和第二色度分量，其中所述第二译码单元在解码次序上在所述第一译码单元之后，且其中所述第二译码单元是不同于所述第一译码单元的译码单元；

针对所述第二译码单元的所述第二亮度分量，接收指示用于所述第二译码单元的所述第二亮度分量的YCbCr颜色空间量化参数与用于所述第一译码单元的所述第一亮度分量的未经修改的所述YCbCr颜色空间所述量化参数之间的差的差值；

基于所述接收到的差值及用于所述第一译码单元的所述第一亮度分量的未经修改的所述YCbCr颜色空间量化参数确定用于所述第二译码单元的所述第二亮度分量的所述YCbCr颜色空间量化参数；

基于所述所确定的用于所述第二译码单元的所述第二亮度分量的所述YCbCr颜色空间量化参数对所述第二译码单元进行解码；以及

输出包含所述第一译码单元和所述第二译码单元的经解码视频数据。

块自适应颜色空间转换译码

[0001] 本申请案主张以下各者的权益：

[0002] 2014年6月20日提交的第62/015,347号美国临时申请案

[0003] 2014年10月10日提交的第62/062,797号美国临时申请案，所述美国临时申请案的全部内容以引入的方式并入本文中。

技术领域

[0004] 本发明涉及视频编码及解码。

背景技术

[0005] 数字视频能力可并入到多种多样的装置中，包含数字电视、数字直播系统、无线广播系统、个人数字助理 (PDA)、膝上型或桌上型计算机、平板计算机、电子图书阅读器、数码相机、数字记录装置、数字媒体播放器、视频游戏装置、视频游戏控制台、蜂窝式或卫星无线电电话 (所谓的“智能电话”)、视频电话会议装置、视频串流装置及其类似者。数字视频装置实施视频译码技术，例如，由MPEG-2、MPEG-4、ITU-T H.263、ITU-T H.264/MPEG-4第10部分、高级视频译码 (AVC) 定义的标准、高效率视频译码 (HEVC) 标准及目前正在开发的此类标准的扩展中所描述的视频译码技术。视频装置可通过实施这些视频译码技术而更有效率地传输、接收、编码、解码及/或存储数字视频信息。

[0006] 视频译码技术包含空间 (图片内) 预测及/或时间 (图片间) 预测以减少或移除视频序列中固有的冗余。对于基于块的视频译码，可将视频切片 (例如，视频帧或视频帧的一部分) 分割成视频块 (其也可被称作树块)、译码单元 (CU) 及/或译码节点。使用关于同一图片中的相邻块中的参考样本的空间预测对图片的经帧内译码 (I) 切片中的视频块进行编码。图片的经帧间编码 (P或B) 切片中的视频块可使用相对于同一图片中的相邻块中的参考样本的空间预测或相对于其它参考图片中的参考样本的时间预测。图片可被称为帧，且参考图片可被称为参考帧。

[0007] 空间或时间预测产生待译码块的预测性块。残余数据表示待译码原始块与预测性块之间的像素差。根据指向形成预测性块的参考样本块的运动向量及指示经译码块与预测性块之间的差的残余数据对经帧间译码块进行编码。根据帧内译码模式及残差数据对经帧内译码块进行编码。为了进一步压缩，可将残余数据从像素域变换到变换域，从而产生残余变换系数，可接着量化所述残余变换系数。可扫描一开始按二维阵列排列的经量化变换系数，以便产生变换系数的一维向量，且可应用熵译码以实现更多压缩。

发明内容

[0008] 本发明描述与确定在使用颜色空间转换译码时的量化参数相关的技术，并且此外，描述用于将在使用颜色空间转换译码时的量化参数从编码器用信号发送到解码器的技术。

[0009] 在一个实例中，一种对视频数据进行解码的方法包含：接收所述视频数据的第一

块;接收用以确定用于所述第一块的量化参数的信息;响应于确定所述第一块是使用用于所述第一块的残余数据的颜色空间变换模式进行译码的,修改用于所述第一块的所述量化参数;基于用于所述第一块的所述修改后的量化参数对所述第一块执行解量化过程;接收所述视频数据的第二块;针对所述第二块,接收指示用于所述第二块的量化参数与用于所述第一块的所述量化参数之间的差的差值;基于所述接收到的差值及用于所述第一块的所述量化参数确定用于所述第二块的所述量化参数;以及基于所述所确定的用于所述第二块的量化参数对所述第二块进行解码。

[0010] 在另一实例中,一种对视频数据进行编码的方法包含:确定用于视频数据的第一块的量化参数;响应于确定所述视频数据的第一块是使用用于所述第一块的残余数据的颜色空间变换模式进行译码的,修改用于所述第一块的所述量化参数;基于用于所述第一块的所述修改后的量化参数对所述第一块执行量化过程;确定用于视频数据的第二块的量化参数;以及用信号发送用于所述第一块的所述量化参数与用于所述第二块的所述量化参数之间的差值。

[0011] 在另一实例中,一种用于对视频数据进行解码的装置包含:存储器,其经配置以存储视频数据;一或多个处理器,其经配置以:接收所述视频数据的第一块;接收用以确定用于所述第一块的量化参数的信息;响应于确定所述第一块是使用用于所述第一块的残余数据的颜色空间变换模式进行译码的,修改用于所述第一块的所述量化参数;基于用于所述第一块的所述修改后的量化参数对所述第一块执行解量化过程;接收所述视频数据的第二块;针对所述第二块,接收指示用于所述第二块的量化参数与用于所述第一块的所述量化参数之间的差的差值;基于所述接收到的差值及用于所述第一块的所述量化参数确定用于所述第二块的所述量化参数;以及基于所述所确定的用于所述第二块的量化参数对所述第二块进行解码。

[0012] 在另一实例中,一种用于对视频数据进行编码的装置包含:存储器,其经配置以存储视频数据;一或多个处理器,其经配置以:确定用于视频数据的第一块的量化参数;响应于确定所述视频数据的第一块是使用用于所述第一块的残余数据的颜色空间变换模式进行译码的,修改用于所述第一块的所述量化参数;基于用于所述第一块的所述修改后的量化参数对所述第一块执行量化过程;确定用于视频数据的第二块的量化参数;以及用信号发送用于所述第一块的所述量化参数与用于所述第二块的所述量化参数之间的差值。

[0013] 在另一实例中,一种用于对视频进行解码的设备,所述设备包括:用于接收所述视频数据的第一块的装置;用于接收用以确定用于所述第一块的量化参数的信息的装置;用于响应于确定所述第一块是使用用于所述第一块的残余数据的颜色空间变换模式进行译码的而修改用于所述第一块的所述量化参数的装置;用于基于用于所述第一块的所述修改后的量化参数对所述第一块执行解量化过程的装置;用于接收所述视频数据的第二块的装置;用于针对所述第二块而接收指示用于所述第二块的量化参数与用于所述第一块的所述量化参数之间的差的差值的装置;用于基于所述接收到的差值及用于所述第一块的所述量化参数确定用于所述第二块的所述量化参数的装置;以及用于基于所述所确定的用于所述第二块的量化参数对所述第二块进行解码的装置。

[0014] 一种存储指令的计算机可读存储媒体,所述指令在由一或多个处理器执行时致使所述一或多个处理器:接收所述视频数据的第一块;接收用以确定用于所述第一块的量化

参数的信息;响应于确定所述第一块是使用用于所述第一块的残余数据的颜色空间变换模式进行译码的,修改用于所述第一块的所述量化参数;基于用于所述第一块的所述修改后的量化参数对所述第一块执行解量化过程;接收所述视频数据的第二块;针对所述第二块,接收指示用于所述第二块的量化参数与用于所述第一块的所述量化参数之间的差的差值;基于所述接收到的差值及用于所述第一块的所述量化参数确定用于所述第二块的所述量化参数;以及基于所述所确定的用于所述第二块的量化参数对所述第二块进行解码。

[0015] 在附图及下文描述中陈述本发明的一或多个实例的细节。本发明的其它特征、目标及优点将从所述描述及图式以及权利要求书而显而易见。

附图说明

[0016] 图1是说明可利用本发明中描述的技术的实例视频编码及解码系统的框图。

[0017] 图2是说明高效率视频译码 (HEVC) 帧内预测模式的概念图。

[0018] 图3A及图3B是说明根据本发明的一或多种技术的空间相邻运动向量合并候选者及高级运动向量预测 (AMVP) 模式的概念图。

[0019] 图4是说明根据本发明的一或多种技术的帧内块复制 (BC) 实例的概念图。

[0020] 图5是说明根据本发明的一或多种技术用于帧内 8×8 块的目标块及参考样本的实例的概念图。

[0021] 图6是说明可实施本发明中描述的技术的实例视频编码器的框图。

[0022] 图7是说明可实施本发明中描述的技术的实例视频解码器的框图。

[0023] 图8是说明根据本发明的技术的实例视频解码方法的流程图。

[0024] 图9是说明根据本发明的技术的实例视频解码方法的流程图。

具体实施方式

[0025] 本发明描述视频译码技术,包含与最近定稿的高效率视频译码 (HEVC) 标准的新兴屏幕内容译码 (SCC) 扩展及范围扩展 (RCEX) 相关的技术。SCC及范围扩展设计为潜在地支持高位深度 (例如,超过8位) 及/或高色度采样格式,并且因此设计为包含基本HEVC标准中未含有的新的译码工具。

[0026] 一种此类译码工具是颜色空间转换译码。在颜色空间转换译码中,视频编码器可将残余数据从第一颜色空间 (例如,YCbCr) 转换为第二颜色空间 (例如,RGB),以便获得更佳的译码质量 (例如,更佳的速率失真权衡)。无论残余数据的颜色空间,视频编码器通常将残余数据变换为变换系数并量化所述变换系数。视频解码器执行解量化所述变换系数并逆变换所述变换系数以重构残余数据的互逆过程。视频编码器向视频解码器用信号发送指示在量化所述变换系数时所使用的缩放量的量化参数。所述量化参数还可以由其它视频译码过程 (例如解块滤波) 使用。

[0027] 本发明描述与确定在使用颜色空间转换译码时的量化参数相关的技术,并且此外,描述用于将在使用颜色空间转换译码时的量化参数从编码器用信号发送到解码器的技术。例如,在颜色空间转换译码中,视频译码器 (例如,视频编码器或视频解码器) 可修改用于第一块的量化参数。对于用于第二块的量化参数,视频译码器可对针对差值的信息进行译码 (例如,对其进行编码或对其进行解码)。在本发明中描述的技术中,差值是用于第一块

的量化参数(即,未经修改的量化参数)与用于第二块的量化参数之间的差。

[0028] 图1是说明可利用针对屏幕内容译码的技术的实例视频编码及解码系统10的框图。如图1中所示,系统10包含源装置12,其提供稍后待由目的地装置14解码的经编码视频数据。具体来说,源装置12经由计算机可读媒体16将视频数据提供到目的地装置14。源装置12及目的地装置14可包括广泛范围的装置中的任一者,包含台式计算机、笔记型(即,膝上型)计算机、平板计算机、机顶盒、例如所谓的“智能”电话的电话手持机、所谓的“智能”板、电视、相机、显示装置、数字媒体播放器、视频游戏控制台、视频流式传输装置或其类似者。在一些情况下,可装备源装置12及目的地装置14以用于无线通信。

[0029] 目的地装置14可经由计算机可读媒体16接收待解码的经编码视频数据。计算机可读媒体16可包括能够将经编码视频数据从源装置12移动到目的地装置14的任一类型的媒体或装置。在一个实例中,计算机可读媒体16可包括通信媒体以使源装置12能够实时地将经编码视频数据直接传输到目的地装置14。可根据通信标准(例如,无线通信协议)调制经编码视频数据,并将其传输到目的地装置14。通信媒体可包括任何无线或有线通信媒体,例如射频(RF)频谱或一或多个物理传输线。通信媒体可形成分组网络(例如,局域网、广域网或全球网络,例如因特网)的部分。通信媒体可包含路由器、交换器、基站或任何其它可用于促进从源装置12到目的地装置14的通信的设备。

[0030] 在一些实例中,源装置12可输出经编码数据,其可被输出到存储装置。类似地,输入接口可从存储装置存取经编码数据。存储装置可包含多种分布式或本地存取的数据存储媒体中的任一者,例如硬盘驱动器、蓝光光盘、DVD、CD-ROM、快闪存储器、易失性或非易失性存储器,或任何其它用于存储经编码视频数据的合适的数字存储媒体。在另一实例中,存储装置可对应于可存储由源装置12产生的经编码视频的文件服务器或另一中间存储装置。目的地装置14可从所述存储装置经由流式传输或下载来存取所存储的视频数据。文件服务器可为能够存储经编码视频数据且将所述经编码视频数据传输到目的地装置14的任何类型的服务器。实例文件服务器包含Web服务器(例如,用于网站)、FTP服务器、网络附接存储(NAS)装置或本地磁盘驱动器。目的地装置14可通过任何标准数据连接(包含因特网连接)来存取经编码视频数据。所述数据连接可包含无线信道(例如,Wi-Fi连接)、有线连接(例如,DSL、电缆调制解调器等),或适合于存取存储在文件服务器上的经编码视频数据的两者的组合。经编码视频数据从存储装置的传输可能是流式传输、下载传输或其组合。

[0031] 本发明的技术未必限于无线应用或设定。所述技术可应用于支持各种多媒体应用中的任一者的视频译码,例如空中电视广播、有线电视传输、卫星电视传输、因特网流视频传输(例如,HTTP动态自适应流式传输(DASH))、被编码到数据存储媒体上的数字视频,对存储在数据存储媒体上的数字视频的解码,或其它应用。在一些实例中,系统10可经配置以支持单向或双向视频传输,从而支持例如视频流式传输、视频重放、视频广播及/或视频电话等应用。

[0032] 在图1的实例中,源装置12包含视频源18、视频编码器20及输出接口22。目的地装置14包含输入接口28、视频解码器30及显示装置32。根据本发明,源装置12的视频编码器20可经配置以使用颜色空间转换过程应用用于对视频块进行编码的技术。在其它实例中,源装置及目的地装置可包含其它组件或布置。例如,源装置12可从外部视频源18(例如外部相机)接收视频数据。同样,目的地装置14可与外部显示装置介接,而不是包含集成式显示装

置。

[0033] 图1的所说明的系统10只是一个实例。可通过任何数字视频编码及/或解码装置来执行用于使用颜色空间转换过程对视频块进行译码的技术。尽管本发明的技术总体上由视频译码装置执行,但是所述技术还可通过视频编码器/解码器(通常被称作“编解码器”)执行。源装置12及目的地装置14仅为源装置12产生用于传输到目的地装置14的经译码视频数据的此类译码装置的实例。在一些实例中,装置12、14可以基本上对称的方式操作,使得装置12、14中的每一者包含视频编码及解码组件。因此,系统10可支持视频装置12、14之间的单向或双向视视频传播,以(例如)用于视频流式传输、视频重放、视频广播或视频电话。

[0034] 源装置12的视频源18可包含视频捕捉装置,例如摄像机、含有先前所捕捉视频的视频存档,及/或用于从视频内容提供者接收视频的视频馈送接口。作为另一替代方案,视频源18可产生基于计算机图形的数据作为源视频,或实况视频、所存档视频与计算机产生的视频的组合。在一些情况下,如果视频源18是摄像机,那么源装置12及目的地装置14可形成所谓的相机电话或视频电话。然而,如上文所提及,本发明中所描述的技术可大体上适用于视频译码,且可应用于无线及/或有线应用。在每一情况下,经捕捉、经预先捕捉或经计算机产生的视频可由视频编码器20编码。经编码视频信息可接着由输出接口22输出到计算机可读媒体16上。

[0035] 计算机可读媒体16可包含瞬时媒体,例如无线广播或有线网络传输,或存储媒体(即,非暂时性存储媒体),例如硬盘、快闪驱动器、压缩光盘、数字视频光盘、蓝光光盘或其它计算机可读媒体。在一些实例中,网络服务器(未示出)可从源装置12接收经编码视频数据,且例如经由网络传输将经编码视频数据提供到目的地装置14。类似地,媒体生产设施(例如,光盘冲压设施)的计算装置可从源装置12接收经编码视频数据且生产含有经编码视频数据的光盘。因此,在各种实例中,计算机可读媒体16可理解为包含各种形式的一或多个计算机可读媒体。

[0036] 目的地装置14的输入接口28从计算机可读媒体16接收信息。计算机可读媒体16的信息可包含由视频编码器20定义的语法信息,所述语法信息也由视频解码器30使用,其包含描述块及其它经译码单元(例如,GOP)的特性及/或处理的语法元素。显示装置32 将经解码视频数据显示给用户,且可包括多种显示装置中的任一者,例如阴极射线管(CRT)、液晶显示器(LCD)、等离子显示器、有机发光二极管(OLED)显示器或另一类型的显示装置。

[0037] 视频编码器20及视频解码器30各自可实施为多种合适的编码器电路中的任一者,例如一或多个微处理器、数字信号处理器(DSP)、专用集成电路(ASIC)、现场可编程门阵列(FPGA)、离散逻辑、软件、硬件、固件或其任何组合。当所述技术部分地在软件中实施时,装置可将用于所述软件的指令存储于合适的非暂时性计算机可读媒体中并使用一或多个处理器执行硬件中的指令以执行本发明的技术。视频编码器20及视频解码器 30中的每一者可包含在一或多个编码器或解码器中,所述编码器或解码器中的任一者可集成为装置中的组合编码器/解码器(CODEC)的部分。包含视频编码器20及/或视频解码器30的装置可包括集成电路、微处理器及/或无线通信装置,例如,蜂窝式电话。

[0038] 视频译码标准包含ITU-T H.261、ISO/IEC MPEG-1Visual、ITU-T H.262或ISO/IEC MPEG-2Visual、ITU-T H.263、ISO/IEC MPEG-4Visual及ITU-T H.264(也称为ISO/IEC MPEG-4AVC),包含其可缩放视频译码(SVC)及多视图视频译码(MVC)扩展。已由ITU-T 视频

译码专家组 (VCEG) 及 ISO/IEC 动画专家组 (MPEG) 的视频译码联合协作小组 (JCT-VC) 完成新的视频译码标准 (即, 高效率视频译码 (HEVC)) 的设计。视频编码器 20 及视频解码器 30 可根据例如 HEVC 等视频译码标准操作, 并且可符合 HEVC 测试模型 (HM)。替代地, 视频编码器 20 及视频解码器 30 可根据其它专属或业界标准来操作, 所述标准例如 ITU-T H.264 标准, 替代地被称为 MPEG-4 第 10 部分高级视频译码 (AVC), 或此类标准的扩展。但是, 本发明的技术不限于任何特定译码标准。视频译码标准的其它实例包含 MPEG-2 及 ITU-T H.263。

[0039] ITU-T H.264/MPEG-4 (AVC) 标准由 ITU-T 视频译码专家组 (VCEG) 连同 ISO/IEC 动画专家组 (MPEG) 制定为被称为联合视频小组 (JVT) 的集体合作伙伴的产品。在一些方面中, 本发明中描述的技术可应用到大体符合 H.264 标准的装置。H.264 标准描述于 ITU-T 研究组的日期为 2005 年 3 月的“ITU-T 推荐 H.264, 用于通用视听服务的高级视频译码” (ITU-T Recommendation H.264, Advanced Video Coding for generic audiovisual services) 中, 所述标准在本文中可被称作 H.264 标准或 H.264 规范或 H.264/AVC 标准或规范。联合视频小组 (JVT) 继续致力于对 H.264/MPEG-4 AVC 的扩展。

[0040] JCT-VC 开发 HEVC 标准。所述 HEVC 标准化努力是基于被称作 HEVC 测试模型 (HM) 的视频译码器的进化的模型。HM 假定视频译码器相对于根据例如 ITU-T H.264/AVC 的现有装置的若干额外能力。例如, 虽然 H.264 提供九个帧内预测编码模式, 但 HM 可提供多达三十三个帧内预测编码模式。

[0041] 一般来说, HM 的工作模型描述视频帧或图片可被分成一连串的译码树单元 (CTU)。CTU 也可被称作树块或最大译码单元 (LCU)。CTU 中的每一者可包括亮度样本的译码树块、色度样本的两个对应的译码树块, 以及用以对译码树块的样本进行译码的语法结构。在单色图片或具有三个单独颜色平面的图片中, CTU 可包括单个译码树块及用于对所述译码树块的样本进行译码的语法结构。译码树块可为样本的 $N \times N$ 块。位流内的语法数据可定义 LCU (其是就像素数目来说的最大译码单元) 的大小。

[0042] 在 HEVC 中, 切片中的最大译码单元称为译码树块 (CTB)。CTB 含有四叉树, 其节点被称为译码单元 (CU)。在 HEVC 主规范中, CTB 的大小可在 16×16 到 64×64 的范围内, 但是也可支持更小的大小, 例如 8×8 的 CTB 大小, 及更大的大小。

[0043] 本发明可使用术语“视频单元”或“视频块”或“块”来指代一或多个样本块以及用于对所述一或多个样本块的样本进行译码的语法结构。实例类型的视频单元可包含 CTU、CU、PU、变换单元 (TU)、宏块、宏块分区等。在一些情形中, PU 的论述可与宏块或宏块分区的论述互换。

[0044] 切片包含按译码次序的若干连续树块。视频帧或图片可被分割成一或多个切片。每一树块可根据四叉树分裂成译码单元 (CU)。一般来说, 四叉树数据结构包含每个 CU 一个节点, 其中一个根节点对应于所述树块。如果一个 CU 分裂成四个子 CU, 那么对应于 CU 的节点包含四个叶节点, 其中的每一者对应于所述子 CU 中的一者。

[0045] 四叉树数据结构的每一节点可提供用于对应 CU 的语法数据。例如, 四叉树中的节点可包含分裂旗标, 其表明对应于所述节点的所述 CU 是否分裂成子 CU。用于 CU 的语法元素可递归地定义, 且可取决于 CU 是否分裂成子 CU。如果 CU 未进一步分裂, 那么将其称作叶 CU。在本发明中, 叶 CU 的四个子 CU 也将被称作叶 CU, 即使不存在原始叶 CU 的明确分裂时也是如此。例如, 如果 16×16 大小的 CU 不进一步分裂, 那么四个 8×8 子 CU 也将被称作叶 CU, 尽管 16

×16CU从未分裂。

[0046] CU可具有CTB的相同大小并且可小至 8×8 。每一CU以一个预测模式译码。当使用帧间预测模式对CU进行译码时(即,当CU经帧间译码时),CU可进一步分割成两个或更多个预测单元(PU)。在其它实例中,当不应用进一步分割时CU可包含仅一个PU。在其中CU被分割成两个PU的实例中,每一PU可为具有等于CU的一半的大小的矩形,或具有CU大小的1/4或3/4的两个矩形。在HEVC中,最小PU大小为 8×4 及 4×8 。

[0047] CU具有与H.264标准的宏块类似的目的,除了CU不具有大小区别。例如,树块可分裂成四个子节点(也称为子CU),且每一子节点又可为父节点且可分裂成另外四个子节点。最后未分裂的子节点(被称作二叉树的叶节点)包括译码节点,也被称作叶CU。与经译码位流相关联的语法数据可定义树块可分裂的最大次数,被称作最大CU深度,且还可定义译码节点的最小大小。因此,位流还可定义最小译码单元(SCU)。本发明使用术语“块”指HEVC的情况下的CU、PU或TU中的任一者,或其它标准的情况下的类似数据结构(例如,其在H.264/AVC中的宏块及子块)。

[0048] CU包含译码节点及与所述译码节点相关联的预测单元(PU)及变换单元(TU)。CU的大小对应于译码节点的大小并且形状必须是正方形。CU的大小范围可从 8×8 像素到具有 64×64 像素或更大像素的最大值的树块的大小。每一CU可含有一或多个PU及一或多个TU。例如,与CU相关联的语法数据可描述CU分割成一或多个PU。分割模式可在CU被跳过或经直接模式编码、帧内预测模式编码或帧间预测模式编码之间有区别。PU可分割成非正方形形状。例如,与CU相关联的语法数据还可描述CU根据二叉树分割成一或多个TU。TU可为正方形或非正方形(例如,矩形)形状。

[0049] HEVC标准允许根据TU变换,TU可针对不同CU而有所不同。TU的大小通常是基于针对经分割LCU定义的给定CU内的PU的大小而确定,但是情况可能并不总是如此。TU通常与PU大小相同或小于PU。在一些实例中,对应于CU的残余样本可使用被称为“残余二叉树”(RQT)的二叉树结构细分成较小单元。RQT的叶节点可被称为变换单元(TU)。可以变换与TU相关联的像素差值以产生变换系数,所述变换系数可经量化。

[0050] 叶CU可包含一或多个预测单元(PU)。一般来说,PU表示对应于相对应的CU的全部或一部分的空间区域,并且可包含用于检索PU的参考样本的数据。此外,PU包含与预测有关的数据。例如,当PU经帧内模式编码时,用于PU的数据可包含在残余二叉树(RQT)中,残余二叉树可包含描述用于对应于PU的TU的帧内预测模式的数据。作为另一实例,当PU经帧间模式编码时,PU可包含定义PU的一或多个运动向量的数据。界定PU的运动向量的数据可描述(例如)运动向量的水平分量、运动向量的垂直分量、运动向量的分辨率(例如,四分之一像素精度或八分之一像素精度)、运动向量所指向的参考图片,及/或运动向量的参考图片列表(例如,列表0、列表1或列表C)。

[0051] 作为实例,HM支持各种PU大小的预测。假设特定CU的大小是 $2N \times 2N$,那么HM支持 $2N \times 2N$ 或 $N \times N$ 的PU大小的帧内预测,及 $2N \times 2N$ 、 $2N \times N$ 、 $N \times 2N$ 或 $N \times N$ 的对称PU大小的帧间预测。HM还支持针对 $2N \times nU$ 、 $2N \times nD$ 、 $nL \times 2N$ 及 $nR \times 2N$ 的PU大小的帧间预测的不对称分割。在不对称分割中,不分割CU的一个方向,但是将另一方向分割成25%及75%。CU的对应于25%分区的部分通过“n”、后面接续“上”、“下”、“左”或“右”的指示来指示。因此,例如,“ $2N \times nU$ ”是指水平地分割的 $2N \times 2N$ CU,其中顶部为 $2N \times 0.5N$ PU,而底部为 $2N \times 1.5N$ PU。

[0052] 在本发明中,“ $N \times N$ ”与“ N 乘 N ”可互换使用来指在垂直及水平维度方面的视频块的像素尺寸,例如, 16×16 像素或16乘16像素。大体来说, 16×16 块将在垂直方向上具有16个像素($y=16$),且在水平方向上具有16个像素($x=16$)。同样, $N \times N$ 块总体上在垂直方向上具有 N 个像素,并且在水平方向上具有 N 个像素,其中 N 表示非负整数值。块中的像素可布置成行及列。此外,块未必需要在水平方向上与在竖直方向上具有相同数目个像素。例如,块可包括 $N \times M$ 个像素,其中 M 未必等于 N 。

[0053] 具有一或多个PU的叶CU还可包含一或多个变换单元(TU)。变换单元可使用RQT(也称为TU四叉树结构)来指定,如上文所论述。例如,分裂旗标可指示叶CU是否分裂成四个变换单元。接着,每一变换单元可进一步分裂成更多个子TU。当TU未进一步分裂时,其可被称作叶TU。一般来说,对于帧内译码,所有属于一个叶CU的叶TU共享相同的帧内预测模式。也就是说,通常应用相同的帧内预测模式来计算一个叶CU的所有TU的经预测值。对于帧内译码,视频编码器可使用帧内预测模式将每一叶TU的残余值计算为CU的对应于TU的部分与原始块之间的差。TU不一定限于PU的大小。因此,TU可大于或小于PU。对于帧内译码,PU可与相同CU的对应叶TU处于相同位置。在一些实例中,叶TU的最大大小可以对应于对应的叶CU的大小。

[0054] HEVC指定具有 4×4 、 8×8 、 16×16 及 32×32 的大小的四个变换单元(TU)以对预测残余进行译码。CU可递归地分割成4个或更多个TU。TU可使用类似于离散余弦变换(DCT)的基底函数。此外,在一些实例中,可使用从离散正弦变换(DST)导出的整数变换来变换属于经帧内译码区的 4×4 亮度变换块。色度变换块可使用与亮度变换块相同的TU大小。

[0055] 此外,叶CU的TU还可与四叉树数据结构(被称作残余四叉树(RQT))相关联。即,叶CU可包含指示叶CU如何分割成TU的四叉树。TU四叉树的根节点一般对应于叶CU,而CU四叉树的根节点一般对应于树块(或LCU)。未经分裂的RQT的TU被称作叶TU。一般来说,本发明分别使用术语CU及TU指代叶CU及叶TU,除非另有指出。

[0056] 当CU经帧间译码时,针对每一PU可存在运动信息的一个集合。在一些实例中,例如当PU位于B切片中时,针对每一PU可存在运动信息的两个集合。此外,可以唯一帧间预测模式对每一PU进行译码以导出针对每一PU的运动信息的集合。

[0057] 视频序列通常包含一系列视频帧或图片。图片群组(GOP)一般包括一系列的一或多个视频图片。GOP可包含GOP的标头、一或多个图片的标头或其它地方中的语法数据,其描述GOP中包含的多个图片。图片的每一切片可包含描述用于所述切片的编码模式的切片语法数据。视频编码器通常对个别视频切片内的视频块进行操作以便对视频数据进行编码。视频块可与CU内的译码节点相对应。视频块可具有固定或变化的大小,并且根据指定译码标准可在大小上有所不同。

[0058] 图2是说明HEVC帧内预测模式的概念图250。针对每一PU的亮度分量,通过33个有角度的帧内预测模式(编索引为从2到34)、DC模式(编索引为1)及平面模式(编索引为0)来利用帧内预测方法,如关于图2所描述。

[0059] 除了以上35个帧内预测模式外,还通过HEVC采用再多一个帧内预测模式,名为帧内脉码调制(I-PCM)。在I-PCM模式中,绕过预测、变换、量化及熵译码,同时通过预定义数目的位来对预测样本进行译码。I-PCM模式的主要目的是处理当无法通过其它帧内预测模式有效地对信号进行译码时的情况。

[0060] 在使用CU的PU的帧内预测性或帧间预测性译码之后,视频编码器20可计算CU 的TU的残余数据。PU可包括描述在空间域(也被称作像素域)中产生预测性像素数据的方法或模式的语法数据,并且TU可包括在对残余视频数据应用变换(例如离散余弦变换(DCT)、整数变换、小波变换或概念上类似的变换)之后变换域中的系数。所述残余数据可对应于未经编码图片的像素与对应于PU的预测值之间的像素差。视频编码器20可形成包含CU的残余数据的TU,并且接着变换TU以产生CU的变换系数。

[0061] 在用以产生变换系数的任何变换后,视频编码器20可执行变换系数的量化。量化一般指代对变换系数进行量化以可能减少用以表示系数的数据的量从而提供进一步压缩的过程。量化过程可减少与系数中的一些或全部相关联的位深度。例如,n位值可在量化期间被下舍入到m位值,其中n大于m。

[0062] 在量化之后,视频编码器20可扫描变换系数,从包含经量化变换系数的二维矩阵产生一维向量。扫描可经设计以将较高能量(并且因此较低频率)的系数放置在阵列的前面,并且将较低能量(并且因此较高频率)的系数放置在阵列的背面。在一些实例中,视频编码器20可使用预定义扫描次序来扫描经量化变换系数以产生可经熵编码的串行化向量。在其它实例中,视频编码器20可执行自适应扫描。在扫描经量化变换系数以形成一维向量之后,视频编码器20可例如根据上下文自适应可变长度译码(CAVLC)、上下文自适应二进制算术译码(CABAC)、基于语法的上下文自适应二进制算术译码(SBAC)、概率区间分割熵(PIPE)译码或另一熵编码方法,对表示一维向量中的变换系数的语法元素进行熵编码。视频编码器还20也可对与经编码视频数据相关联的语法元素进行熵编码以供视频解码器30在对视频数据进行解码时使用。

[0063] 视频编码器20可输出包含形成经译码图片及相关联数据的表示的位序列的位流。因此,位流包括视频数据的经编码表示。位流可包括一连串网络抽象层(NAL)单元。NAL单元是含有NAL单元中的数据类型的指示及含有所述数据的呈按需要穿插有模拟阻止位的原始字节序列有效负载(RBSP)的形式的字节的语法结构。NAL单元中的每一者包含NAL单元标头,且包封RBSP。NAL单元标头可包含指示NAL单元类型码的语法元素。由NAL单元的NAL单元标头指定的NAL单元类型码指示NAL单元的类型。RBSP可为含有包封在NAL单元内的整数数目个字节的语法结构。在一些情况下,RBSP包含零个位。

[0064] 不同类型的NAL单元可包封不同类型的RBSP。例如,不同类型的NAL单元可包封用于视频参数集(VPS)、序列参数集(PS)、图片参数集(PPS)、经译码切片、补充增强信息(SEI)等的不同RBSP。封装视频译码数据的RBSP(与参数集及SEI消息的RBSP相对)的NAL单元可被称为视频译码层(VCL)NAL单元。在HEVC(即,非多层HEVC)中,存取单元可为按解码次序连续且含有确切一个经译码图片的NAL单元的集合。除经译码图片的经译码切片NAL单元之外,存取单元还可含有不含经译码图片的切片的其它NAL单元。在一些实例中,对存取单元的解码始终产生经解码图片。补充增强信息(SEI)含有并非对来自VCLNAL单元的经译码图片的样本进行解码所必需的信息。SEIRBSP含有一或多个SEI消息。

[0065] 如上文所简要地指出,NAL单元可包封VPS、PS及PPS的RBSP。VPS是包括应用于零个或多个完整经译码视频序列(CVS)的语法元素的语法结构。PS也是包括应用于零个或多个完整CVS的语法元素的语法结构。PS可包含识别当PS在作用中时在作用中的VPS的语法元素。因此,VPS的语法元素可比PS的语法元素更一般地适用。PPS是包括应用于零个

或更多个经译码图片的语法元素的语法结构。PPS可包含识别当PPS在作用中时在作用中的SPS的语法元素。切片的切片标头可包含指示当切片正经译码时在作用中的PPS的语法元素。

[0066] 视频解码器30可接收由视频编码器20产生的位流。另外,视频解码器30可剖析位流以从位流获得语法元素。视频解码器30可至少部分地基于从位流获得的语法元素重构视频数据的图片。用于重构视频数据的过程通常可与由视频编码器20执行的过程互逆。举例来说,视频解码器30可使用PU的运动向量来确定用于当前CU的PU的预测性块。另外,视频解码器30可逆量化当前CU的TU的系数块。视频解码器30可对系数块执行逆变换以重构当前CU的TU的变换块。通过将用于当前CU的PU的预测性块的样本增加到当前CU的TU的变换块的对应样本上,视频解码器30可重构当前CU的译码块。通过重构图片的每一CU的译码块,视频解码器30可重构图片。

[0067] 在HEVC标准中,存在两个帧间预测模式。针对预测单元(PU),这些帧间预测模式分别为合并模式(应注意,跳过模式被视为合并模式的特殊情况)及高级运动向量预测(AMVP)模式。在AMVP或合并模式中,可为多个运动向量预测符维持运动向量(MV)候选者列表。可通过从MV候选者列表获取一个候选者来产生当前PU的一或多个运动向量以及合并模式中的参考索引。

[0068] 在一些情况下,MV候选者列表可含有针对合并模式的多达5个候选者及针对AMVP模式的仅两个候选者。合并候选者可含有运动信息集合,例如对应于两个参考图片列表(例如,列表0及列表1)的运动向量及参考索引。如果通过合并索引来识别合并候选者,那么使用参考图片用于当前块的预测,以及确定相关联的运动向量。然而,在AMVP模式下,对于来自列表0或列表1的每一潜在预测方向,需要明确地将参考索引连同对MV候选者列表的MVP索引一起用信号发送,这是因为AMVP候选者可仅含有运动向量。在AMVP模式中,可进一步优化经预测运动向量。

[0069] 合并候选者可对应于运动信息的完整集合,而AMVP候选者可含有用于特定预测方向及参考索引的仅一个运动向量。可类似地从相同空间及时间相邻块导出用于两种模式的候选者。

[0070] 图3A及图3B是说明根据本发明的一或多种技术的空间相邻运动向量合并候选者及高级运动向量预测(AMVP)模式的概念图。如关于图3A及图3B所描述,针对特定PU(PU0)从图3A及图3B中示出的相邻块导出空间MV候选者,但是对于合并模式及AMVP模式,从所述块产生候选者的方法不同。

[0071] 在合并模式中,可以图3A中示出的次序导出多达四个空间MV候选者,其中编号及次序以下:左(0)、上(1)、右上(2)、左下(3)及左上(4),如图3A中所示。

[0072] 在AMVP模式中,相邻块分成两个群组:由块0及1组成的左群组310以及由块2、3及4组成的上群组320,如图3B中所示。针对左群组310及上群组320中的每一者,参考如由经信号传送的参考索引所指示的同一参考图片的相邻块中的潜在候选者具有待选择以形成所述群组的最终候选者的最高优先权。有可能所有相邻块均不含有指向同一参考图片的运动向量。因此,如果无法找到此类候选者,那么可按比例缩放第一可用候选者以形成最终候选者,因此可补偿时间距离差异。

[0073] 例如远程桌面、远程游戏、无线显示器、车用信息娱乐、云计算等多个应用在日常

生活中正变得常规。这些应用中的视频内容通常是自然内容、文本、人工图形等的组合。在文本及人工图形区中,通常存在重复的图案(例如字符、图标、符号等)。帧内块复制(帧内BC)是可使视频译码器能够去除此种冗余且改进图片内译码效率的技术。在一些情况下,帧内BC替代地可被称为帧内运动补偿(MC)。

[0074] 根据一些帧内BC技术,视频译码器可将同一图片内的经先前译码视频数据块用作当前视频数据块,所述经先前译码视频数据块在同一图片中的当前视频数据块(待译码)的正上方或直接与所述当前视频数据块水平成一直线,以用于当前块的预测。换句话说,如果视频数据的图片强加于2-D栅格,那么每一视频数据块将占据x值及y值的唯一范围。因此,一些视频译码器可基于仅共享相同x值集合(即,与当前块垂直成一直线)或相同y值集合(即,与当前块水平成一直线)的经先前译码视频数据块来预测当前视频数据块。

[0075] 图4是说明根据本发明的一或多种技术的帧内块复制(BC)实例的概念图。如关于图4所描述,帧内BC已包含于RExt中。帧内BC的实例如图4中所示,其中当前CU 402 是从当前图片/切片的已经解码块404预测的。当前帧内BC块大小可与CU大小一样大,范围为 8×8 至 64×64 ,但在一些应用中,可另外应用其它约束。

[0076] 在传统的视频译码中,可假设图像为连续色调的及在空间上平滑的。基于这些假设,已经开发了例如基于块的变换、滤波等各种工具,并且它们已经显示出针对具有自然内容的视频的良好性能。然而,在某些应用中,例如远程桌面、协作式工作及无线显示器,计算机所产生的屏幕内容可为待压缩的主要内容。此类型的内容往往会是离散色调的并且特征为具有高对比度物体边界的清晰线。然而,连续色调及平滑的假设可能不再适用。因而,传统的视频译码技术可能无法有效地工作。

[0077] 为了矫正此效率损失,视频译码器可使用调色板模式译码。2013年4月10日提交的第61/810,649号美国临时申请案描述了调色板译码技术的实例。针对每一CU,可导出调色板,其包含当前CU中的最主要像素值。首先传输调色板的大小及元素。接着根据特定扫描次序对CU中的像素进行编码。针对每一位置,视频编码器20可首先传输语法元素,例如旗标palette_flag,以指示像素值是否在调色板中(“行程模式”)或不在调色板中(“像素模式”)。

[0078] 在“行程模式”中,视频编码器20可用信号发送跟随有“行程”的调色板索引。所述行程为指示扫描次序中具有与当前经译码的像素相同的调色板索引值的连续像素的数目的语法元素。如果扫描次序中即时连续中的多个像素具有相同调色板索引值,那么“行程模式”可由语法元素(例如,palette_flag)指示。可确定计数器值,其等于具有与当前像素相同的调色板索引值的接续当前像素的像素的数目,且行程设定成等于所述计数器值。视频编码器20不需要针对“行程”所覆盖的后续位置传输palette_flag或调色板索引,因为当前像素后续的像素中的每一者具有相同像素值。在解码器侧,仅将对当前像素的第一调色板索引值进行解码,且将针对“行程”语法元素中指示的像素的“行程”中的每一像素复制结果。在“像素模式”中,视频编码器20传输针对此位置的像素样本值。如果例如palette_flag的语法元素指示“像素模式”,那么仅针对经解码的当前像素确定调色板索引值。

[0079] 根据本发明中描述的技术,针对呈4:4:4色度格式的序列提出了针对残余信号(即,残余块)的环路内颜色空间变换;然而,所述技术不限于4:4:4格式。环路内颜色空间变换过程将呈RGB/YUV色度格式的预测误差信号(即,残余信号)变换为次佳颜色空间中的那

些信号。环路内颜色空间变换可进一步减少颜色分量之间的相关性。可通过奇异值分解 (SVD) 从每一CU的像素样本值导出变换矩阵。颜色空间变换可应用于帧内模式及帧间模式两者的预测误差。

[0080] 当将颜色空间变换应用于帧间模式时, 首先将残余转换成具有所导出变换矩阵的不同域。在颜色空间转换之后, 按次序执行译码步骤, 例如DCT/DST、量化及熵译码。

[0081] 当将颜色空间变换应用于使用帧内模式译码的CU时, 首先分别将预测及当前块转换成具有所导出变换矩阵的不同域。在颜色空间转换之后, 进一步通过DCT/DST变换当前块与当前块的预测符之间的残余、量化所述残余及对所述残余进行熵译码。

[0082] 例如视频编码器20的视频编码装置执行正向运算, 其中将包括转换值a、b、c、d、e、f、g、h及i的颜色空间变换矩阵应用于三个平面G、B及R以导出针对颜色分量P、Q及S的值, 如下:

$$[0083] \quad \begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix} \begin{bmatrix} G \\ B \\ R \end{bmatrix} = \begin{bmatrix} P \\ Q \\ S \end{bmatrix}$$

[0084] 所得值可削减在HEVC规范的范围, 因为在最差的情况中值可放大高达 $\sqrt{3}$ 倍。例如视频解码器30的视频解码装置执行反向运算, 其中将包括转换值 a^t 、 b^t 、 c^t 、 d^t 、 e^t 、 f^t 、 g^t 、 h^t 及 i^t 的颜色空间变换矩阵应用于三个颜色分量P'、Q'及R'以导出三个平面G'、B'及R', 如下:

$$[0085] \quad \begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix}^t \begin{bmatrix} P' \\ Q' \\ S' \end{bmatrix} = \begin{bmatrix} G' \\ B' \\ R' \end{bmatrix}$$

[0086] 图5是说明根据本发明的一或多种技术用于帧内 8×8 块的目标块及参考样本的实例的概念图。可使用奇异值分解 (SVD) 从参考样本值导出变换矩阵。针对帧内情况及帧间情况, 视频译码装置 (例如, 视频编码器20或视频解码器30) 可使用不同参考样本。对于经帧内译码块的情况, 目标块及参考样本可如图5中所示。在图5中, 目标块由 8×8 交叉影线样本构成, 而参考样本为条纹及加点的样本。

[0087] 针对经帧间译码块的情况, 用于矩阵导出的参考样本可与用于运动补偿的参考样本相同。可对高级运动预测 (AMP) 块中的参考样本进行二次采样使得减少参考样本的数目。例如, 使 12×16 块中的参考样本的数目减少2/3。

[0088] 在以上实例中的一些中, 可始终应用颜色空间变换过程。因此, 可能不需要用信号发送是否调用颜色空间变换过程。另外, 视频编码器20及视频解码器30均可使用相同方法来导出变换矩阵以避免用于用信号发送变换矩阵的开销。

[0089] 视频编码器20及视频解码器30可使用各种颜色空间变换矩阵。例如, 视频编码器20及视频解码器30可针对不同颜色空间应用不同颜色空间变换矩阵。举例来说, 视频编码器20及视频解码器30可使用一对YCbCr变换矩阵来将样本值从RGB颜色空间转换为YCbCr颜色空间以及返回。以下方程式显示YCbCr变换矩阵的一个实例集合:

$$[0090] \quad \text{正向: } \begin{bmatrix} Y \\ Cb \\ Cr \end{bmatrix} = \begin{bmatrix} 0.2126 & 0.7152 & 0.0722 \\ -0.1172 & -0.3942 & 0.5114 \\ 0.5114 & -0.4645 & -0.0469 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

$$[0091] \quad \text{反向:} \begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 1 & 0 & 1.5397 \\ 1 & -0.1831 & -0.4577 \\ 1 & 1.8142 & 0 \end{bmatrix} \begin{bmatrix} Y \\ Cb \\ Cr \end{bmatrix}$$

[0092] 在另一实例中,视频编码器20及视频解码器30可使用一对YCoCg变换矩阵来将样本值从RGB颜色空间转换为YCoCg颜色空间以及返回。以下方程式显示YCoCg变换矩阵的一个实例集合:

$$[0093] \quad \text{正向:} \begin{bmatrix} Y \\ Co \\ Cg \end{bmatrix} = \begin{bmatrix} 1/4 & 1/2 & 1/4 \\ 1/2 & 0 & -1/2 \\ -1/4 & 1/2 & -1/4 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

$$[0094] \quad \text{反向:} \begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 1 & 1 & -1 \\ 1 & 0 & 1 \\ 1 & -1 & -1 \end{bmatrix} \begin{bmatrix} Y \\ Co \\ Cg \end{bmatrix}$$

[0095] 另一此类矩阵可为YCoCg-R矩阵,其为以因数2按比例缩放Co及Cg分量的YCoCg 矩阵的经修正版本。通过使用提升技术,视频编码器20及视频解码器30可通过以下方程式获得正向及反向变换:

$$[0096] \quad \begin{aligned} & Co = R - B \\ & t = B + [Co/2] \\ & Cg = G - t \\ & Y = t + [Cg/2] \end{aligned}$$

$$[0097] \quad \begin{aligned} & t = Y - [Cg/2] \\ & G = Cg + t \\ & B = t - [Co/2] \\ & R = B + Co \end{aligned}$$

[0098] 在以上方程式及矩阵中,可在编码过程(例如,通过视频编码器)之前执行正向变换。相反,可在解码过程(例如,通过视频解码器)之后执行反向变换。

[0099] 切片的切片标头含有关于所述切片的信息。举例来说,切片的切片标头可含有语法元素,视频解码器30可从所述语法元素导出用于所述切片的定量参数。在HEVC中,切片段标头语法结构对应于切片标头。下表示出了如所定义的切片段标头的一部分:

[0100]	slice_segment_header() {	描述符
	...	
	slice_qp_delta	se(v)
	if(pps_slice_chroma_qp_offsets_present_flag) {	
	slice_cb_qp_offset	se(v)
	slice_cr_qp_offset	se(v)
	}	
	if(chroma_qp_offset_list_enabled_flag)	
	cu_chroma_qp_offset_enabled_flag	u(1)
	...	
	}	

[0101] 在以上实例及本发明的其它语法表中,具有形式u(n)的描述符的语法元素(其中n是非负整数)是长度n的无正负号值。此外,描述符se(v)指示其中左位第一的带正负号的整数的0阶经指数-哥伦布译码的语法元素。

[0102] 在上表中,slice_qp_delta语法元素指定Q_{py}的初始值将用于切片中的译码块直

到由 CU层中的CuQpDeltaVal的值修改。用于切片的 Q_{PY} 是用于切片的块的亮度分量的QP。可导出用于切片的 Q_{PY} 量化参数的初始值Slice Q_{PY} ,如下:

[0103] $\text{Slice}Q_{PY} = 26 + \text{init_qp_minus26} + \text{slice_qp_delta}$

[0104] 在以上方程式中,init_qp_minus26是在PPS中用信号发送的语法元素。init_qp_minus26语法元素指定初始值减去用于每一切片的Slice Q_{PY} 的26。Slice Q_{PY} 的值可在 $-Q_{pBdOffset_Y}$ 到+51(包含端点)的范围内。 $Q_{pBdOffset_Y}$ 是等于bit_depth_luma_minus8语法元素乘以6的变量。bit_depth_luma_minus8语法元素指定亮度阵列的样本的位深度及亮度量化参数范围偏移 $Q_{pBdOffset_Y}$ 的值。视频编码器20可在SPS中用信号发送bit_depth_luma_minus8语法元素。

[0105] 另一语法元素slice_cb_qp_offset指定当确定 $Q_{p'}_{cb}$ 量化参数的值时待加入 pps_cb_qp_offset(或亮度量化参数偏移)的值的差。slice_cb_qp_offset的值可在-12到+12(包含端点)的范围内。当不存在slice_cb_qp_offset时,推断slice_cb_qp_offset等于0。pps_cb_qp_offset+slice_cb_qp_offset的值可在-12到+12(包含端点)范围内。

[0106] 语法元素slice_cr_qp_offset指定当确定 $Q_{p'}_{cb}$ 量化参数的值时待加入 pps_cr_qp_offset(或亮度量化参数偏移)的值的差。slice_cr_qp_offset的值可在-12到+12(包含端点)的范围内。当不存在slice_cr_qp_offset时,推断slice_cr_qp_offset等于0。pps_cr_qp_offset+slice_cr_qp_offset的值可在-12到+12(包含端点)范围内。

[0107] 当语法元素cu_chroma_qp_offset_enabled_flag等于1时,在变换单元语法中可存在 cu_chroma_qp_offset_flag。当cu_chroma_qp_offset_enabled_flag等于0时,在变换单元语法中可不存在cu_chroma_qp_offset_flag。当不存在时,推断 cu_chroma_qp_offset_enabled_flag的值等于0。

[0108] 变换单元可具有以下语法:

	transform_unit(x0, y0, xBase, yBase, log2TrafoSize, trafoDepth, blkIdx) {	描述符
	log2TrafoSizeC = Max(2, log2TrafoSize - (ChromaArrayType == 3 ? 0 : 1))	
	cbfDepthC = trafoDepth - (ChromaArrayType != 3 && log2TrafoSize == 2 ? 1 : 0) [Ed. Check case with smaller max depth.]	
	xC = (ChromaArrayType != 3 && log2TrafoSize == 2) ? xBase : x0	
	yC = (ChromaArrayType != 3 && log2TrafoSize == 2) ? yBase : y0	
	cbfLuma = cbf_luma[x0][y0][trafoDepth]	
	cbfChroma = cbf_cb[xC][yC][cbfDepthC] cbf_cr[xC][yC][cbfDepthC] (ChromaArrayType == 2 && (cbf_cb[xC][yC + (1 << log2TrafoSizeC)][cbfDepthC] cbf_cr[xC][yC + (1 << log2TrafoSizeC)][cbfDepthC]))	
[0109]	if(cbfLuma cbfChroma) {	
	if(cu_qp_delta_enabled_flag && !IsCuQpDeltaCoded) {	
	cu_qp_delta_abs	ae(v)
	if(cu_qp_delta_abs)	
	cu_qp_delta_sign_flag	ae(v)
	}	
	if(cu_chroma_qp_offset_enabled_flag && cbfChroma && !cu_transquant_bypass_flag && !IsCuChromaQpOffsetCoded) {	
	cu_chroma_qp_offset_flag	ae(v)
	if(cu_chroma_qp_offset_flag && chroma_qp_offset_list_len_minus1 > 0)	
	cu_chroma_qp_offset_idx	ae(v)
	}	
	if(cbfLuma)	
[0110]	residual_coding(x0, y0, log2TrafoSize, 0)	
	...	
	}	
	}	

[0111] 语法元素cu_qp_delta_abs指定当前译码单元的亮度量化参数与其预测之间的差CuQpDeltaVal的绝对值。在上表中,描述符ae(v)指示经上下文自适应算术熵译码的语法元素。

[0112] 语法元素cu_qp_delta_sign_flag指定CuQpDeltaVal的正负号。如果 cu_qp_delta_sign_flag等于0,那么相应CuQpDeltaVal具有正值。否则 (cu_qp_delta_sign_flag等于1),相应CuQpDeltaVal具有负值。当不存在 cu_qp_delta_sign_flag时,推断cu_qp_delta_sign_flag等于0。

[0113] 当存在cu_qp_delta_abs时,可导出变量IsCuQpDeltaCoded及CuQpDeltaVal如下。

[0114] IsCuQpDeltaCoded=1

[0115] CuQpDeltaVal=cu_qp_delta_abs*(1-2*cu_qp_delta_sign_flag)

[0116] CuQpDeltaVal的值可在-(26+QpBdOffsetY/2)到+(25+QpBdOffsetY/2)的范围内(包含端点)。

[0117] 当语法元素cu_chroma_qp_offset_flag存在且等于1时,其指定cb_qp_offset_list[]中的条目用以确定CuQp0ffsetCb的值,并且cr_qp_offset_list[]中的相应条目用以确定 CuQp0ffsetCr的值。当变量cu_chroma_qp_offset_flag等于0时,不使用这些列表来确定CuQp0ffsetCb及CuQp0ffsetCr的值。

[0118] 当存在语法元素cu_chroma_qp_offset_idx时,其指定用以确定CuQpOffsetCb及CuQpOffsetCr的值的对cb_qp_offset_list[]及cr_qp_offset_list[]的索引。当存在时,cu_chroma_qp_offset_idx的值将在0到chroma_qp_offset_list_len_minus1(包含端点)的范围内。当不存在时,推断cu_chroma_qp_offset_idx的值等于0。

[0119] 可检查其中因为cu_chroma_qp_offset_flag已经存在于同一群组的一些其它CU中所以不存在cu_chroma_qp_offset_flag的情况,以及其中因为列表含有仅一个条目所以旗标等于1但是不存在索引的情况。当存在cu_chroma_qp_offset_flag时,变量IsCuChromaQpOffsetCoded设定成等于1。接着导出变量CuQpOffsetCb及CuQpOffsetCr。如果cu_chroma_qp_offset_flag等于1,那么CuQpOffsetCb=cb_qp_offset_list[cu_chroma_qp_offset_idx],且CuQpOffsetCr=cr_qp_offset_list[cu_chroma_qp_offset_idx]。否则(cu_chroma_qp_offset_flag等于0),CuQpOffsetCb及CuQpOffsetCr均设定成等于0。

[0120] 在解码过程中,对于针对量化参数的导出过程,对此过程的输入是指定当前亮度译码块的左上方样本相对于当前图片的左上方亮度样本的亮度位置(xCb,yCb)。在此过程中,导出变量QpY、亮度量化参数Qp'Y及色度量化参数Qp'Cb及Qp'Cr。

[0121] 根据本发明的技术,量化群组是CU的TU的集合,其中TU中的每一者共享相同QP值。亮度位置(xQg,yQg)指定当前量化群组的左上方亮度样本相对于当前图片的左上方亮度样本。水平及垂直位置xQg及yQg分别设定成等于xCb-(xCb&((1<<Log2MinCuQpDeltaSize)-1))及yCb-(yCb&((1<<Log2MinCuQpDeltaSize)-1))。量化群组的亮度大小Log2MinCuQpDeltaSize确定共享相同qPY_PRED的译码树块内的最小区域的亮度大小。

[0122] 视频译码器可通过以下有序的步骤导出经预测亮度量化参数qPY_PRED:

[0123] 1) 可导出变量qPY_PREV。如果以下条件中的一或多者成立,那么视频译码器设定qPY_PREV等于SliceQpY:当前量化群组是切片中的第一量化群组、当前量化群组是瓦片中的第一量化群组、或当前量化群组是译码树块行中的第一量化群组,且entropy_coding_sync_enabled_flag等于1。否则,qPY_PREV设定成等于按解码次序在先前量化群组中的最后一个译码单元的亮度量化参数QpY。

[0124] 2) 以设定成等于(xCb,yCb)的位置(xCurr,yCurr)及设定成等于(xQg-1,yQg)的相邻位置(xNbY,yNbY)作为输入而调用z扫描次序中的块的可用性导出过程,且将输出指派给availableA。导出变量qPY_A如下:如果以下条件中的一或多者成立,那么qPY_A设定成等于qPY_PREV:availableA等于FALSE,或含有覆盖亮度位置(xQg-1,yQg)的亮度译码块的译码树块的译码树块地址ctbAddrA不等于CtbAddrInTs,其中ctbAddrA导出如下:

[0125] xTmp=(xQg-1)>>Log2MinTrafoSize

[0126] yTmp=yQg>>Log2MinTrafoSize

[0127] minTbAddrA=MinTbAddrZs[xTmp][yTmp]

[0128] ctbAddrA=(minTbAddrA>>2)*(CtbLog2SizeY-Log2MinTrafoSize)

[0129] 否则,qPY_A设定成等于含有覆盖(xQg-1,yQg)的亮度译码块的译码单元的亮度量化参数QpY。

[0130] 3) 以设定成等于(xCb,yCb)的位置(xCurr,yCurr)及设定成等于(xQg,yQg-1)的相邻位置(xNbY,yNbY)作为输入而调用z扫描次序中的块的可用性导出过程。将输出指派给

availableB。导出变量 q_{PY_B} 。如果以下条件中的一或多者成立,那么 q_{PY_B} 设定成等于 q_{PY_PREV} :availableB等于FALSE,或含有覆盖亮度位置 $(xQg, yQg-1)$ 的亮度译码块的译码树块的译码树块地址 $ctbAddrB$ 不等于 $CtbAddrInTs$,其中 $ctbAddrB$ 导出如下:

[0131] $xTmp = xQg \gg \text{Log2MinTrafoSize}$

[0132] $yTmp = (yQg-1) \gg \text{Log2MinTrafoSize}$

[0133] $\text{minTbAddrB} = \text{MinTbAddrZs}[xTmp][yTmp]$

[0134] $ctbAddrB = (\text{minTbAddrB} \gg 2) * (\text{CtbLog2SizeY} - \text{Log2MinTrafoSize})$

[0135] 否则, q_{PY_B} 设定成等于含有覆盖 $(xQg, yQg-1)$ 的亮度译码块的CU的亮度量化参数 Q_{PY} 。

[0136] 4) 经预测亮度量化参数 q_{PY_PRED} 可导出如下:

[0137] $q_{PY_PRED} = (q_{PY_A} + q_{PY_B} + 1) \gg 1$

[0138] 变量 Q_{PY} 可导出如下:

[0139] $Q_{PY} = ((q_{PY_PRED} + \text{CuQpDeltaVal} + 52 + 2 * \text{QpBdOffsetY}) \% (52 + \text{QpBdOffsetY})) - \text{QpBdOffsetY}$

[0140] 亮度量化参数 $Q_{p'Y}$ 可导出如下:

[0141] $Q_{p'Y} = Q_{PY} + \text{QpBdOffsetY}$

[0142] 当ChromaArrayType不等于0时,变量 q_{PiCb} 及 q_{PiCr} 导出如下:

[0143] $q_{PiCb} = \text{Clip3}(-\text{QpBdOffsetC}, 57, Q_{PY} + \text{pps_cb_qp_offset} + \text{slice_cb_qp_offset} + \text{CuQpOffsetCb})$

[0144] $q_{PiCr} = \text{Clip3}(-\text{QpBdOffsetC}, 57, Q_{PY} + \text{pps_cr_qp_offset} + \text{slice_cr_qp_offset} + \text{CuQpOffsetCr})$

[0145] 如果ChromaArrayType等于1,那么基于索引 q_{Pi} 等于 q_{PiCb} 及 q_{PiCr} 而分别将变量 q_{Pcb} 及 q_{Pcr} 设定成等于 Q_{pc} 的值。否则,基于索引 q_{Pi} 等于 q_{PiCb} 及 q_{PiCr} 而分别将变量 q_{Pcb} 及 q_{Pcr} 设定成等于 $\text{Min}(q_{Pi}, 51)$ 。

[0146] 用于Cb及Cr分量的色度量化参数 $Q_{p'Cb}$ 及 $Q_{p'Cr}$ 导出如下:

[0147] $Q_{p'Cb} = q_{Pcb} + \text{QpBdOffsetC}$

[0148] $Q_{p'Cr} = q_{Pcr} + \text{QpBdOffsetC}$

[0149] 对于ChromaArrayType等于1,取决于 q_{Pi} 的 Q_{pc} 的规范如下所示:

[0150]	q_{Pi}	<30	30	31	32	33	34	35	36	37	38	39	40	41	42	43	>43
	Q_{pc}	= q_{Pi}	29	30	31	32	33	33	34	34	35	35	36	36	37	37	= $q_{Pi}-6$

[0151] 在解量化过程中,导出用于每一分量索引($cIdx$)的量化参数 q_P 。如果 $cIdx$ 等于0,则 $q_P = Q_{p'Y}$ 。否则,如果 $cIdx$ 等于1,则 $q_P = Q_{p'Cb}$ 。否则($cIdx$ 等于2), $q_P = Q_{p'Cr}$ 。在解块滤波过程中,首先确定取决于 Q_{PY} 的亮度/色度边缘。HEVC的子条款8.7.2.5.3及8.7.2.5.5提供了解块滤波过程的详细内容。

[0152] 2014年4月18日提交的美国临时专利申请案61/981,645定义了环路内颜色变换公式,例如具有位深度递增的归一化YCgCo变换及YCgCo变换。另外,美国临时专利申请案61/981,645描述了颜色变换可在预测过程之后在常规变换/量化过程之前应用于帧内模式的残余域。此外,美国临时专利申请案61/981,645指出不同颜色分量可将不同增量QP用于以基于变换标准的颜色变换译码的块。

[0153] 美国临时专利申请案61/981,645的视频译码技术可能在若干方面得以改进。举例来说,用于三个颜色分量的固定增量QP设定可能并非对所有情况(例如,所有帧内/随机存取/低延迟)都最佳。此外,当使用具有位深度递增的非归一化YCgCo变换时,变换导致标准变换的位宽增加,这会增加硬件实施方案的成本。相反,如果标准变换保持无变化,那么对于一些情况其可归因于输入残余数据的精确度增加而导致溢出。

[0154] 本发明的技术提供解决方案以相较于先前设计改进环路内颜色空间变换的译码性能并减少解码器复杂度。例如视频编码器20或视频解码器30的视频译码器可执行关于图1到9描述的技术中的任一者。

[0155] 在一些实例中,用于经解码的三个颜色分量的QP的增量值集合表示为(ΔQP_{C0} , ΔQP_{C1} , ΔQP_{C2}),其指示用于启用颜色变换的块的QP相较于以不启用颜色变换而确定的QP的偏移。对于以启用颜色变换而译码的块,对于的三个颜色分量,在解量化过程中使用的最终QP设定为 $qP + \Delta QP_{C0}$ 、 $qP + \Delta QP_{C1}$ 、 $qP + \Delta QP_{C2}$,其中分量索引cIdx分别等于0、1、2。 qP 是常规QP导出过程的输出。在一些实例中, ΔQP_{C0} 等于 ΔQP_{C1} ,而 ΔQP_{C0} 及 ΔQP_{C1} 两者均小于 ΔQP_{C2} 。

[0156] 举例来说,视频编码器20可对视频数据的CU进行编码。在对视频数据进行编码时,视频编码器20可确定使用颜色空间转换对所述CU进行编码。对于颜色分量,视频编码器20可确定所述颜色分量的初始QP,并基于使用颜色空间转换经编码的CU设定所述颜色分量的最终QP,使得所述颜色分量的最终QP等于所述颜色分量的初始QP与所述颜色分量的非零QP偏移的总和。视频编码器20可基于所述颜色分量的最终QP量化所述CU的系数块,所述CU的系数块是基于所述颜色分量的样本值。一旦已经量化每一系数,视频编码器20可进一步在经编码位流中输出基于所述CU的经量化系数块的经编码CU。

[0157] 在另一实例中,视频解码器30可对视频数据的CU进行解码。在对视频数据进行解码时,视频解码器30可确定所述CU已使用颜色空间转换进行解码。对于颜色分量,视频解码器30可确定所述颜色分量的初始QP,并基于使用颜色空间转换经编码的CU确定所述颜色分量的最终QP,使得所述颜色分量的最终QP等于所述颜色分量的初始QP与所述颜色分量的非零QP偏移的总和。视频解码器30可基于所述颜色分量的最终QP逆量化所述CU的系数块,所述CU的系数块是基于所述颜色分量的样本值。一旦已经逆量化每一系数,视频解码器30可基于所述CU的经逆量化系数块重构所述CU。

[0158] 在一些实例中,对于一或多个颜色分量中的颜色分量,可在PPS、SPS或切片标头中的一者中用信号发送所述颜色分量的QP偏移。在一些进一步的实例中,多个颜色分量可包括三个颜色分量。在此类实例中,用于第一颜色分量的第一量化参数的第一QP偏移等于用于第二颜色分量的第二QP的第二QP偏移,第一QP偏移(及第二量化参数偏移)小于用于第三颜色分量的第三QP的第三QP偏移。

[0159] 因此,在一些实例中,CU是第一CU。在此类实例中,视频编码器20可对第二CU进行编码。在对第二CU进行编码时,视频编码器20可针对颜色分量确定所述颜色分量的QP;基于未经使用颜色空间转换编码的第二CU设定所述颜色分量的最终QP值,使得所述颜色分量的最终QP值等于所述颜色分量的初始QP值;并且基于所述颜色分量的最终QP量化第二CU的系数块,第二CU的系数块是基于所述颜色分量的样本值。视频编码器20可进一步输出视频数据位流,包括表示经量化第二系数块中的每一者的一或多个经熵编码语法元素的第二集

合。

[0160] 在此实例的解码中,视频解码器30可对第二CU进行解码。在对第二CU进行解码时,视频解码器30可针对多个颜色分量中的颜色分量确定所述颜色分量的QP;基于未经使用颜色空间转换编码的第二CU确定所述颜色分量的最终QP值,使得所述颜色分量的最终QP值等于所述颜色分量的初始QP值;并且基于所述颜色分量的最终QP逆量化第二CU的系数块,第二CU的系数块是基于所述颜色分量的样本值。视频解码器30 可基于第二CU的一或多个经逆量化系数块中的每一者重构第二CU。

[0161] 用于三个颜色分量的增量QP的设定可取决于模式,而不是使用用于所有模式的增量QP的一个固定集合。在一个实例中,帧内及帧内BC模式可共享($\Delta QP_{C0}, \Delta QP_{C1}, \Delta QP_{C2}$)的相同集合,而帧间模式可共享($\Delta QP_{C0}, \Delta QP_{C1}, \Delta QP_{C2}$)的另一集合,所述另一集合不同于帧内及帧内BC模式所使用的集合。在另一实例中,帧内模式可共享($\Delta QP_{C0}, \Delta QP_{C1}, \Delta QP_{C2}$)的相同集合,而帧内BC模式及帧间模式可共享($\Delta QP_{C0}, \Delta QP_{C1}, \Delta QP_{C2}$)的另一集合,所述另一集合不同于帧内模式所使用的集合。在一些实例中,增量QP的集合($\Delta QP_{C0}, \Delta QP_{C1}, \Delta QP_{C2}$)可为 $(-4+6*BitInc, -4+6*BitInc, -3+6*BitInc)$ 、 $(-4+6*BitInc, -4+6*BitInc, -2+6*BitInc)$ 、 $(-5+6*BitInc, -5+6*BitInc, -3+6*BitInc)$ 或 $(-5+6*BitInc, -5+6*BitInc, -2+6*BitInc)$,其中BitInc可为0、1、2。

[0162] 换句话说,在一些实例中,多个颜色分量包括三个颜色分量。在此类实例中,量化参数偏移可等于 $(-5+6*BitInc, -5+6*BitInc, -3+6*BitInc)$ 。在其它此类实例中,量化参数偏移可等于其它值,例如 $(-4+6*BitInc, -4+6*BitInc, -3+6*BitInc)$ 、 $(-4+6*BitInc, -4+6*BitInc, -2+6*BitInc)$ 或 $(-5+6*BitInc, -5+6*BitInc, -2+6*BitInc)$ 。在任何情况下,BitInc 可等于0、1或2。

[0163] I切片是可含有仅经帧内译码块或经帧内BC译码块的切片。P切片是可含有仅经帧内译码及经单向帧间预测块的切片。B切片是可含有经帧内预测块、经单向帧间预测块及经双向帧间预测块的切片。在一些实例中,用于三个颜色分量的增量QP的设定可取决于切片类型,而不是使用用于所有模式的增量QP的一个固定集合。在一个实例中,I 切片可共享相同集合,而P/B切片共享相同集合。在另一实例中,不同集合可应用于I/P/B 切片。此外,在一些实例中,可在SPS、PPS或切片标头中用信号发送增量QP的集合。

[0164] 换句话说,在一些实例中,对于多个颜色分量中的颜色分量,所述颜色分量的QP偏移可取决于CU的切片类型是I切片类型、P切片类型还是B切片类型。在此类实例中,对于颜色分量,视频编码器20可确定当CU的切片类型是I切片类型时所述颜色分量的QP偏移等于第一值,以及当CU的切片类型是P切片类型或B切片类型时所述QP 偏移等于第二值,所述第一值不同于所述第二值。在其它此类实例中,对于颜色分量,视频编码器20可确定当CU的切片类型是I切片类型时所述颜色分量的QP偏移等于第一值,当CU的切片类型是P切片类型时所述QP偏移等于第二值,以及当CU的切片类型是B切片类型时所述QP偏移等于第三值,所述第一值不同于所述第二值,所述第二值不同于所述第三值,且所述第一值不同于所述第三值。

[0165] 在一些实例中,对于多个颜色分量中的颜色分量,所述颜色分量的QP偏移可取决于CU的切片类型是I切片类型、P切片类型还是B切片类型。在此类实例中,对于颜色分量,视

频解码器30可确定当CU的切片类型是I切片类型时所述颜色分量的QP偏移等于第一值,以及当CU的切片类型是P切片类型或B切片类型时所述QP偏移等于第二值,所述第一值不同于所述第二值。在其它此类实例中,对于颜色分量,视频解码器30可确定当CU的切片类型是I切片类型时所述颜色分量的QP偏移等于第一值,当CU的切片类型是P切片类型时所述QP偏移等于第二值,以及当CU的切片类型是B切片类型时所述QP偏移等于第三值,所述第一值不同于所述第二值,所述第二值不同于所述第三值,且所述第一值不同于所述第三值。

[0166] 在一些实例中,当数据动态范围归因于颜色变换而增加时,视频译码器可将经变换残余削减到与颜色变换之前的那些残余相同的范围中。例如,如果输入数据是N位精确度,那么在帧内/帧间预测之后的残余可在 $[-2^N, 2^N-1]$ 范围内(或更精确地,在 $[-2^N-1, 2^N-1]$ 范围内)。在应用颜色变换之后,还可以将经变换残余削减到相同范围。在一些实例中,当三个颜色分量的经译码块旗标全部等于0时,可跳过逆颜色变换。

[0167] 在一些实例中,当应用颜色变换时,常规地导出的 Q_{pY} 可进一步修改为 $(Q_{pY} + \text{delta}Q_{pC0})$ 。因此,在解块滤波过程中,可首先确定取决于修改后的 Q_{pY} 的亮度/色度边缘的边界强度。替代地,可在解块滤波过程的亮度/色度边缘的边界强度中使用未经修改的 Q_{pY} 。

[0168] 换句话说,在一些实例中,多个颜色分量包含亮度分量及一或多个色度分量。在此类实例中,视频编码器20可至少部分地基于亮度分量的最终QP进一步确定亮度边缘的边界强度。视频编码器20可至少部分地基于色度分量的最终QP进一步确定色度边缘的边界强度。响应于确定亮度边缘的边界强度不符合第一阈值,视频编码器20可对亮度边缘执行解块滤波过程。此外,响应于确定色度边缘的边界强度不符合第二阈值,视频编码器20可对色度边缘执行解块滤波过程。

[0169] 在其它实例中,多个颜色分量包含亮度分量及一或多个色度分量。在此类实例中,视频解码器30可至少部分地基于亮度分量的最终QP进一步确定亮度边缘的边界强度。视频解码器30可至少部分地基于色度分量的最终QP进一步确定色度边缘的边界强度。响应于确定亮度边缘的边界强度不符合第一阈值,视频解码器30可对亮度边缘执行解块滤波过程。此外,响应于确定色度边缘的边界强度不符合第二阈值,视频解码器30可对色度边缘执行解块滤波过程。

[0170] 在一些实例中,可在规范中增加约束,当针对一个CU启用颜色变换且所述CU是以帧内模式译码时,所述CU内的所有PU将使用直接模式(DM)。当使用直接模式对PU进行编码时,视频编码器20不用信号发送运动信息语法元素,但是可用信号发送用于表示残余数据的语法元素。换句话说,色度预测模式可与亮度预测模式相同。替代地,此外,当针对一个CU启用颜色变换时,pcm_flag将等于0。

[0171] 换句话说,在一些实例中,颜色空间转换的输入数据具有N位精确度。在此类实例中,在帧内/帧间预测之后的CU的残余数据可在 $[-2^N, 2^N-1]$ 范围内。在一些其它实例中,响应于确定所述CU是以帧内译码模式进行译码的,视频编码器20可使用同一色度预测模式进一步预测所述CU的所有色度块。在此类实例中,视频编码器20可使用同一亮度预测模式进一步预测所述CU的所有亮度块。所述同一亮度预测模式可与所述同一色度预测模式相同。在另一实例中,一个CU可含有四个亮度块。在此类实例中,每一亮度块可以其自身的亮度预测模式进行译码,并且所述CU内的左上方亮度块的亮度预测模式可与所述同一色度预测模式相同。

[0172] 在其它实例中,颜色空间转换的输入数据具有N位精确度。在此类实例中,在帧内/帧间预测之后的CU的残余数据可在 $[-2^N, 2^N-1]$ 范围内。在一些其它实例中,响应于确定所述CU是以帧内译码模式进行译码的,视频解码器30可使用同一色度预测模式进一步预测所述CU的所有色度块。在此类实例中,视频解码器30可使用同一亮度预测模式进一步预测所述CU的所有亮度块。所述同一亮度预测模式可与所述同一色度预测模式相同。在另一实例中,一个CU可含有四个亮度块。在此类实例中,每一亮度块可以其自身的亮度预测模式进行译码,并且所述CU内的左上方亮度块的亮度预测模式可与所述同一色度预测模式相同。

[0173] 视频编码器20可进一步例如在帧标头、块标头、切片标头或GOP标头中将例如基于块的语法数据、基于帧的语法数据及基于GOP的语法数据等语法数据发送到视频解码器30。GOP语法数据可描述GOP中的帧的数目,且帧语法数据可指示用以对相对应的帧进行编码的编码/预测模式。

[0174] 本发明还包含用以解决与用信号发送量化参数以及量化参数在解块滤波过程中结合自适应颜色空间转换使用相关的另外问题的技术。例如,本发明包含用于配置视频编码器20及视频解码器30以在其中不存在针对块的颜色变换旗标的译码情境中(例如,在其中 rqt_root_cbf 等于0且变换块中不存在非零系数的情境中)操作的技术。另外,本发明包含在其中针对一个序列或图片或切片启用颜色变换的译码情境中将增量QP值从视频编码器20用信号发送到视频解码器30的技术。因此,本发明提供用以改进与环路内颜色空间变换相关联的译码性能的一些潜在解决方案。

[0175] 出于解释的目的,本发明将用于三个颜色分量(即R、G及B)的量化参数(QP)的增量值集合标示为 ΔQP_{C0} 、 ΔQP_{C1} 及 ΔQP_{C2} ,其指示启用颜色变换时块的QP的偏移。如以上文所描述的方式确定的,对于亮度及色度量化的参数,将使用缩写qp。

[0176] 根据本发明的技术,对于以启用颜色变换而译码的块,在解量化过程及/或解块滤波过程中使用的最终QP针对三个颜色分量修改为 $qp_0 + \Delta QP_{C0}$ 、 $qp_1 + \Delta QP_{C1}$ 、 $qp_2 + \Delta QP_{C2}$ 。 qp_{cIdx} 是常规QP导出过程的输出,如HEVC范围扩展工作草案7(RExt WD7)的子条款8.6.1中所定义的,其中 $cIdx$ 等于如输入的0、1、2。

[0177] 根据本发明的一个技术,如果不存在针对块的颜色变换旗标,那么视频编码器20及/或视频解码器30在解块滤波过程期间如同停用颜色变换一般处理所述块。当对于帧间或帧内BC模式 rqt_root_cbf 等于0时,或当所述块是以常规帧内模式译码但色度模式不等于DM模式时,颜色变换旗标可(例如)不存在。当使用字典模式对色度块进行译码时,所述色度块在分区大小等于 $2N \times 2N$ 时具有与亮度预测单元相同的预测模式,或在分区大小等于 $N \times N$ 时具有与左上方亮度预测单元相同的预测模式。应注意,一个经帧内译码CU可具有一个亮度预测模式(分区大小等于 $2N \times 2N$)或四个亮度预测模式(分区大小等于 $N \times N$),但经帧内译码CU可仅具有一个色度预测模式,无论分区大小如何。在HEVC屏幕内容译码中,仅当色度模式等于字典模式时用信号发送颜色变换旗标。

[0178] 根据本发明的另一技术,当不存在针对块颜色变换旗标时,视频编码器20及视频解码器30可经配置以在解块滤波过程期间如同启用颜色变换一般处理所述块。在此类情况下,视频编码器20及视频解码器30可经配置以在解块滤波过程的边界强度确定期间使用修改后的QP(包含增量QP)。视频编码器20及/或视频解码器30可首先确定亮度/色度边缘的边界强度,其可取决于修改后的QP。

[0179] 在一个实例中,如果视频编码器20及/或视频解码器30在启用颜色变换的情况下处理一个块,那么视频编码器20及/或视频解码器30可在确定亮度/色度边缘的边界强度时使用($qP+\Delta QP_{C0}$, $qP+\Delta QP_{C1}$, $qP+\Delta QP_{C2}$),其可与在解量化过程中所使用的相同。

[0180] 替代地,对于以启用颜色变换而译码的块,视频编码器20及/或视频解码器30可在解量化及解块滤波器过程中使用不同量化参数。例如,视频编码器20及/或视频解码器30可在确定亮度/色度边缘的边界强度时使用($qP_0+\Delta QP_{C0}$, $qP_1+\Delta QP_{C0}$, $qP_2+\Delta QP_{C0}$),而在解量化过程中使用($qP_0+\Delta QP_{C0}$, $qP_1+\Delta QP_{C1}$, $qP_2+\Delta QP_{C2}$)。替代地,视频编码器20及/或视频解码器30可对仅一个分量使用修改后的量化参数。例如,如果一个块是在使用颜色变换的情况下进行译码/处理的,那么视频编码器20及/或视频解码器30可使用($qP_0+\Delta QP_{C0}$, qP_1 , qP_2)。

[0181] 在另一实例中,当块是使用帧内模式而不是DM模式进行译码时,视频编码器20及/或视频解码器30可在解块过程期间处理在使用颜色变换情况下处理的所述块。也就是说,在确定边界强度期间使用修改后的QP(即,包含增量QP)。在另一实例中,可取决于序列类型决定在解块滤波过程中是否在使用颜色变换的情况下处理所述块。例如,对于RGB译码,视频编码器20及/或视频解码器30可使用在使用颜色变换的情况下对块进行译码,因为其优选地用于更多块选择颜色变换,而对于YCbCr译码,视频编码器20及/或视频解码器30在具有颜色变换的情况下处理所述块。

[0182] 根据本发明的技术的另一实例,当不存在针对一个块的颜色变换旗标时,可取决于模式决定视频编码器20及/或视频解码器30在解块滤波过程中是否在使用颜色变换的情况下处理所述块。在一个实例中,对于以帧内BC及帧间模式译码的块,当对于帧间或帧内BC模式 rqt_root_cbf 等于0时,即,不存在非零系数时,视频编码器20及/或视频解码器30在使用颜色变换的情况下处理所述块。而对于帧内模式,如果不存在颜色变换旗标,那么视频编码器20及/或视频解码器30在停用颜色变换的情况下处理所述块。

[0183] 在另一实例中,当不存在针对一个块的颜色变换旗标时,如取决于输入序列的格式,视频编码器20及/或视频解码器30针对解块滤波过程确定是在启用颜色变换还是停用颜色变换的情况下处理所述块。在一个实例中,对于呈RGB格式的序列,视频编码器20及/或视频解码器30在启用颜色变换的情况下处理所述块。在另一实例中,对于呈YUV/YCbCr格式的序列,视频编码器20及/或视频解码器30在停用颜色变换的情况下处理所述块。

[0184] 根据本发明的另一技术,即使当在解量化及/或解块滤波器过程中使用修改后的QP时,视频编码器20及/或视频解码器30仍可使用未经修改的QP用作针对后续经译码CU的预测符。例如,不论是否使用颜色空间变换模式对第一块进行编码,以及因此不论是否使用修改后的量化参数解量化第一块的变换系数,视频编码器20都将用于第二块的量化参数作为用于第一块的未经修改的量化参数与用于第二块的未经修改的量化参数之间的差用信号发送到视频解码器30。

[0185] 替代地,视频编码器20及/或视频解码器30可使用修改后的QP作为用于后续经译码CU的预测符。

[0186] 图6是说明可使用颜色空间转换过程实施用于对视频块进行编码的技术的视频编码器20的实例的框图。视频编码器20可执行视频切片内的视频块的帧内及帧间译码。帧内译码依靠空间预测来减少或去除给定视频帧或图片内的视频中的空间冗余。帧间译码依靠

时间预测来减少或去除视频序列的相邻帧或图片内的视频中的时间冗余。帧内模式 (I模式) 可指若干基于空间的译码模式中的任一者。例如单向预测 (P模式) 或双向预测 (B 模式) 等帧间模式可指若干基于时间的译码模式中的任一者。

[0187] 如图6中所示, 视频编码器20接收待编码视频帧内的当前视频块。在图6的实例中, 视频编码器20包含模式选择单元40、参考图片存储器64、求和器50、变换处理单元52、量化单元54及熵编码单元56。模式选择单元40又包含运动补偿单元44、运动估计单元42、帧内预测单元46及分割单元48。基于所选模式, 模式选择单元40还可包含其它单元, 例如帧内BC模式模块。为了视频块重构, 视频编码器20还包含逆量化单元58、逆变换单元60及求和器62。还可包含解块滤波器(图6中未示出) 以对块边界进行滤波从而从经重构视频去除成块效应假影。在典型实例中, 求和器62接收解块滤波器的输出。除了解块滤波器之外, 还可使用额外滤波器(环路内或环路后)。为简洁起见未示出此类滤波器, 但是必要时, 此类滤波器可对求和器50的输出进行滤波(作为环路内滤波器)。

[0188] 在编码过程期间, 视频编码器20接收待编码的视频帧或切片。所述帧或切片可划分成多个视频块。运动估计单元42及运动补偿单元44基于一或多个参考帧中的一或多个块执行视频块的帧间预测性译码以提供时间预测。视频编码器20可执行多个译码遍次, (例如) 从而为每一视频数据块选择适当的译码模式。

[0189] 运动估计单元42及运动补偿单元44可高度集成, 但出于概念的目的分别加以说明。由运动估计单元42执行的运动估计是产生运动向量的过程, 所述过程估计视频块的运动。例如, 运动向量可指示当前视频帧或图片内的视频块的PU相对于参考帧(或其它经译码单元) 内的预测性块相对于当前帧(或其它经译码单元) 内经译码的当前块的位移。预测性块是被发现在像素差方面与待译码块紧密匹配的块, 像素差可通过绝对差总和 (SAD)、平方差和及 (SSD) 或其它差度量来确定。在一些实例中, 视频编码器20可计算存储于参考图片存储器64中的参考图片的次整数像素位置的值。例如, 视频编码器20 可内插参考图片的四分之一像素位置、八分之一像素位置或其它分数像素位置的值。因此, 运动估计单元42可相对于全像素位置及分数像素位置执行运动搜索并且输出具有分数像素精确度的运动向量。

[0190] 运动估计单元42通过比较PU的位置与参考图片的预测性块的位置来计算经帧间译码切片中的视频块的PU的运动向量。参考图片可选自第一参考图片列表(列表0) 或第二参考图片列表(列表1), 所述参考图片列表中的每一者识别存储在参考帧存储器64中的一或多个参考图片。运动估计单元42向熵编码单元56及运动补偿单元44发送计算出的运动向量。

[0191] 由运动补偿单元44执行的运动补偿可涉及基于由运动估计单元42确定的运动向量来提取或产生预测性区块。再者, 在一些实例中, 运动估计单元42与运动补偿单元44 可在功能上整合。在接收到当前视频块的PU的运动向量后, 运动补偿单元44可在参考图片列表中的一者中定位所述运动向量指向的预测块。求和器50可通过从经译码当前视频块的像素值减去预测性块的像素值从而形成像素差值来形成残余视频块, 如下文所论述。一般来说, 运动估计单元42相对于亮度分量执行运动估计, 并且运动补偿单元 44对于色度分量及亮度分量两者使用基于亮度分量计算的运动向量。模式选择单元40 还可产生与视频块及视频切片相关联的语法元素以供视频解码器30在对视频切片的视频块进行解码时使用。

[0192] 作为如上文所描述由运动估计单元42及运动补偿单元44执行的帧间预测的替代

方案,帧内预测单元46可对当前块进行帧内预测。具体来说,帧内预测单元46可确定用来对当前块进行编码的帧内预测模式。在一些实例中,帧内预测单元46可例如在单独编码遍次期间使用各种帧内预测模式对当前块进行编码,且帧内预测单元46(或在一些实例中为模式选择单元40)可从测试模式中选择适当帧内预测模式来使用。

[0193] 例如,帧内预测单元46可使用速率-失真分析计算各种经测试帧内预测模式的速率-失真值,且从经测试模式当中选择具有最佳速率-失真特性的帧内预测模式。速率失真分析一般确定经编码块与经编码以产生经编码块的原始的未经编码块之间的失真(或误差)的量,以及用于产生经编码块的位速率(也就是说,位数目)。帧内预测单元46可根据各种经编码块的失真及速率计算比率,以确定哪种帧内预测模式对于所述块展现最佳速率失真值。

[0194] 在选择用于块的帧内预测模式后,帧内预测单元46可将指示用于块的选定帧内预测模式的信息提供到熵编码单元56。熵编码单元56可对指示选定帧内预测模式的信息进行编码。视频编码器20可在所传输的位流中包含配置数据,其可包含以下各者:多个帧内预测模式索引表及多个修改后的帧内预测模式索引表(还被称作码字映射表);各种块的编码上下文的定义;及待用于上下文中的每一者的最有可能的帧内预测模式、帧内预测模式索引表及修改后的帧内预测模式索引表的指示。

[0195] 帧内预测单元46可基于与待译码块相同帧或切片中的一或多个相邻块执行视频块的帧内预测性译码以提供空间预测。此外,分割单元48可基于前述译码遍次中的前述分割方案的评估将视频数据块分割成子块。例如,分割单元48可起初将帧或切片分割成LCU,并且基于速率失真分析(例如,速率失真优化)将LCU中的每一者分割成子CU。模式选择单元40可进一步产生指示将LCU分割成子CU的四叉树数据结构。四叉树的叶节点CU可包含一或多个PU及一或多个TU。

[0196] 模式选择单元40可例如基于误差结果选择译码模式中的一者(帧内或帧间),并且可将所得经帧内或帧间译码块提供到求和器50以产生残余块数据,以及提供到求和器62以重构经编码块用作参考帧。模式选择单元40还将语法元素(例如帧内模式指示符、分割信息及其它此类语法信息)提供到熵编码单元56。

[0197] 视频编码器20可通过从经译码的原始视频块减去来自模式选择单元40的预测数据而形成残余视频块。求和器50表示执行此减法运算的一或多个组件。变换处理单元52将例如离散余弦变换(DCT)或概念上类似的变换等变换应用于残余块,从而产生包括残余变换系数值的视频块。变换处理单元52可执行概念上类似于DCT的其它变换。也可使用子波变换、整数变换、子带变换或其它类型的变换。在任何状况下,变换处理单元52向残余块应用所述变换,从而产生残余变换系数的块。变换可将残余信息从像素值域转换到变换域(例如,频域)。变换处理单元52可将所得变换系数发送到量化单元54。量化单元54可将所述变换系数量化以进一步减小位速率。量化过程可减少与系数中的一些或全部相关联的位深度。可通过调节量化参数来修改量化的程度。在一些实例中,量化单元54可接着执行对包含经量化变换系数的矩阵的扫描。替代地,熵编码单元56可执行扫描。

[0198] 在量化后,熵编码单元56对经量化变换系数进行熵译码。例如,熵编码单元56可执行上下文自适应可变长度译码(CAVLC)、上下文自适应二进制算术译码(CABAC)、基于语法的上下文自适应二进制算术译码(SBAC)、概率区间分割熵(PIPE)译码或另一熵译码技术。

就基于上下文的熵译码而论,上下文可基于相邻块。在由熵编码单元56进行熵译码之后,可将经编码位流传输到另一装置(例如,视频解码器30),或者将所述经编码位流存档以用于稍后传输或检索。

[0199] 根据本发明的技术,视频编码器20的熵编码单元56可执行本发明的一或多种技术。例如,视频编码器20的熵编码单元56可对视频数据的CU进行编码。在对视频数据进行编码时,颜色空间转换单元51可确定是否使用颜色空间转换对CU进行编码。对于颜色分量,量化单元54可确定所述颜色分量的初始QP,并基于使用颜色空间转换经编码的CU设定所述颜色分量的最终QP,使得所述颜色分量的最终QP等于所述颜色分量的初始QP与所述颜色分量的非零QP偏移的总和。量化单元54可基于所述颜色分量的最终QP量化所述CU的系数块,所述CU的系数块是基于所述颜色分量的样本值。一旦已经量化每一系数,熵编码单元56可进一步输出包括表示经量化系数块中的每一者的一或多个经熵编码语法元素的视频数据位流。

[0200] 逆量化单元58及逆变换单元60分别应用逆量化和逆变换以在像素域中重构残余块,例如以供稍后用作参考块。运动补偿单元44可通过将残余块添加到参考图片存储器64的一个帧的预测性区块来计算参考块。运动补偿单元44还可将一或多个内插滤波器应用于经重构残余块来计算用于在运动估计中使用的次整数像素值。求和器62将经重构残余块添加到由运动补偿单元44产生的经运动补偿的预测块,从而产生经重构视频块以用于存储于参考图片存储器64中。经重构视频块可由运动估计单元42及运动补偿单元44用作参考块以对后续视频帧中的块进行帧间译码。

[0201] 视频编码器20表示视频编码器的实例,其经配置以:确定用于视频数据的第一块的量化参数;响应于确定视频数据的第一块是使用用于第一块的残余数据的颜色空间变换模式进行译码的,修改用于第一块的量化参数;基于用于第一块的修改后的量化参数对第一块执行量化过程;确定用于视频数据的第二块的量化参数;以及用信号发送用于第一块的量化参数与用于第二块的量化参数之间的差值。

[0202] 图7是说明可实施用于对视频块进行解码的技术的视频解码器30的实例的框图,所述视频块中的一些是使用颜色空间转换过程经编码的。在图7的实例中,视频解码器30包含熵解码单元70、运动补偿单元72、帧内预测单元74、逆量化单元76、逆变换单元78、参考图片存储器82及求和器80。视频解码器30还可包含其它单元,例如帧内BC单元。在一些实例中,视频解码器30可执行总体上与关于视频编码器20(图6)描述的编码遍次互逆的解码遍次。运动补偿单元72可基于由从熵解码单元70接收到的语法元素确定的运动向量产生预测数据,而帧内预测单元74可基于从熵解码单元70接收到的帧内预测模式指示符产生预测数据。在一些实例中,帧内预测单元74可推断一些帧内预测模式指示符。

[0203] 在解码过程期间,视频解码器30接收表示经编码视频切片的视频块及相关联语法元素的经编码视频位流。视频解码器30的熵解码单元70对位流进行熵解码以产生经量化系数、运动向量或帧内预测模式指示符及其它语法元素。熵解码单元70将语法元素转发到运动补偿单元72。

[0204] 根据本发明的技术,视频解码器30可执行本发明的一或多种技术。例如,视频解码器30的熵解码单元70可对视频数据的译码单元(CU)进行解码。在对视频数据进行解码时,视频解码器30的逆颜色空间转换单元79可确定所述CU是使用颜色空间转换经编码的。对于

颜色分量,视频解码器30的逆量化单元76可确定所述颜色分量的初始量化参数(QP),并基于使用颜色空间转换经编码的CU确定所述颜色分量的最终QP,使得所述颜色分量的最终QP等于所述颜色分量的初始QP与所述颜色分量的非零QP偏移的总和。视频解码器30的逆量化单元76可基于所述颜色分量的最终QP逆量化所述CU的系数块,所述CU的系数块是基于所述颜色分量的样本值。一旦已经逆量化每一系数,视频解码器30的求和器80可基于所述CU的经逆量化系数块重构所述译码单元。

[0205] 当切片是I切片、P切片或B切片时,帧内预测单元74可使用帧内预测模式来产生预测性块。换句话说,可在切片中具有允许单向或双向帧间预测的经帧内预测块。当视频帧经译码为经帧间译码(即,B、P或GPB)切片时,运动补偿单元72基于从熵解码单元70接收到的运动向量及其它语法元素产生用于当前视频切片的视频块的预测性块。可从参考图片列表中的一者内的参考图片中的一者产生所述预测性块。视频解码器30可基于参考图片存储器82中存储的参考图片使用默认构造技术构造参考图片列表:列表0及列表1。运动补偿单元72通过剖析运动向量及其它语法元素来确定用于当前视频切片的视频块的预测信息,并使用所述预测信息以产生用于经解码的当前视频块的预测性块。例如,运动补偿单元72使用一些接收到的语法元素确定用于对视频切片的视频块进行译码的预测模式(例如,帧内预测或帧间预测)、帧间预测切片类型(例如,B切片、P切片或GPB切片)、用于切片的参考图片列表中的一或多者的构建信息、切片的每一经帧间编码的视频块的运动向量、切片的每一经帧间译码的视频块的帧间预测状态,以及用以对当前视频切片中的视频块进行解码的其它信息。

[0206] 逆量化单元76使在位流中提供且由熵解码单元70解码的经量化变换系数逆量化,即解量化。逆量化过程可包含使用由视频解码器30针对视频切片中的每一视频块计算出的量化参数 QP_Y 来确定应该应用的量化程度和同样的逆量化程度。

[0207] 视频解码器30表示视频解码器的实例,其可经配置以:接收视频数据的第一块;接收用以确定用于第一块的量化参数的信息;响应于确定第一块是使用用于第一块的残余数据的颜色空间变换模式进行译码的,修改用于第一块的量化参数;基于用于第一块的修改后的量化参数对第一块执行解量化过程;接收视频数据的第二块;针对第二块,接收指示用于第二块的量化参数与用于第一块的量化参数之间的差的差值;基于接收到的差值及用于第一块的量化参数确定用于第二块的量化参数;以及基于所确定的用于第二块的量化参数对第二块进行解码。视频解码器30还可基于用于第一块的修改后的量化参数确定用于解块滤波过程的边界强度参数并对第一块执行解块滤波过程。

[0208] 响应于确定针对视频数据的第二块启用颜色空间变换模式,视频解码器30可修改所确定的用于第二块的量化参数,并通过基于用于第二块的修改后的量化参数对第二块执行解量化过程而基于所确定的用于第二块的量化参数对第二块进行解码。视频解码器30还可通过响应于确定针对第二块停用颜色空间变换模式而基于所确定的用于第二块的量化参数对第二块执行解量化过程来基于所确定的用于第二块的量化参数对第二块进行解码。

[0209] 视频解码器30可接收用于第一块的旗标以确定视频数据的第一块是使用用于第一块的残余数据的颜色空间变换模式进行译码的。视频解码器30可接收信息以通过接收用于第一块的量化参数的初始值来确定用于第一块的量化参数。视频解码器30可在切片级接

收所述初始值。视频解码器30可在经译码单元级接收指示用于第二块的量化参数与用于第一块的量化参数之间的差的差值。为了接收指示用于第二块的量化参数与用于第一块的量化参数之间的差的差值,视频解码器30可接收指示所述差的绝对值的语法元素以及接收指示所述差的正负号的语法元素。

[0210] 在以上技术的一些实例实施方案中,可在序列参数集中找到必需的语法元素。在下表中,斜体字表示相对于HEVC标准的现行草案的添加部分。粗体字表示语法元素。在一些实例中,序列参数集RBSP可具有以下语法:

[0211]	seq_parameter_set_rbsp() {	描述符
	sps_video_parameter_set_id	u(4)
	sps_max_sub_layers_minus1	u(3)
	sps_temporal_id_nesting_flag	u(1)
	profile_tier_level(sps_max_sub_layers_minus1)	
	...	
	vui_parameters_present_flag	u(1)
	if(vui_parameters_present_flag)	
	vui_parameters()	
	sps_extension_present_flag	u(1)
	if(sps_extension_present_flag) {	
	for(i = 0; i < 1; i++)	
	sps_extension_flag[i]	u(1)
	sps_extension_7bits	u(7)
	if(sps_extension_flag[0]) {	
	transform_skip_rotation_enabled_flag	u(1)
	transform_skip_context_enabled_flag	u(1)
	intra_block_copy_enabled_flag	u(1)
	implicit_rdpem_enabled_flag	u(1)
	explicit_rdpem_enabled_flag	u(1)
	extended_precision_processing_flag	u(1)
	intra_smoothing_disabled_flag	u(1)
	high_precision_offsets_enabled_flag	u(1)
	fast_rice_adaptation_enabled_flag	u(1)
	cabac_bypass_alignment_enabled_flag	u(1)
	<i>color_transform_enabled_flag</i>	<i>u(1)</i>
	<i>lossless_enable_flag</i>	<i>u(1)</i>
	}	
	if(sps_extension_7bits)	
	while(more_rbsp_data())	
	sps_extension_data_flag	u(1)
	}	
	rbbsp_trailing_bits()	
	}	

[0212] 在此实例中,color_transform_enabled_flag等于1指示启用颜色变换。当语法元素 color_transform_enabled_flag等于0时,不启用颜色变换。当语法元素lossless_enable_flag 等于1时,应用无损译码。另外,当color_transform_enabled_flag等于1时,使用原始 YCoCg-R变换。当语法元素lossless_enable_flag等于0时,应用有损译码。另外,当 color_transform_enabled_flag等于1时,使用原始YCoCg变换。

[0213] 替代地,可仅当chroma_format_idc等于3时用信号发送最新引入的旗标。

[0214]

seq_parameter_set_rbsp() {	描述符
sps_video_parameter_set_id	u(4)
sps_max_sub_layers_minus1	u(3)
sps_temporal_id_nesting_flag	u(1)
profile_tier_level(sps_max_sub_layers_minus1)	
...	
vui_parameters_present_flag	u(1)
if(vui_parameters_present_flag)	
vui_parameters()	
sps_extension_present_flag	u(1)
if(sps_extension_present_flag) {	
for(i = 0; i < 1; i++)	
sps_extension_flag[i]	u(1)
sps_extension_7bits	u(7)
if(sps_extension_flag[0]) {	
transform_skip_rotation_enabled_flag	u(1)
transform_skip_context_enabled_flag	u(1)
intra_block_copy_enabled_flag	u(1)
implicit_rdpem_enabled_flag	u(1)
explicit_rdpem_enabled_flag	u(1)
extended_precision_processing_flag	u(1)
intra_smoothing_disabled_flag	u(1)
high_precision_offsets_enabled_flag	u(1)
fast_rice_adaptation_enabled_flag	u(1)
cabac_bypass_alignment_enabled_flag	u(1)
if(<i>chroma_format_idc</i> == 3) {	
color_transform_enabled_flag	u(1)
lossless_enable_flag	u(1)
}	
}	
if(sps_extension_7bits)	
while(more_rbsp_data())	
sps_extension_data_flag	u(1)
}	
rbbsp_trailing_bits()	
}	

[0215] 替代地,可仅当chroma_format_idc等于3且4:4:4色度格式的三个颜色分量未经分别译码时用信号发送最新引入的旗标。因此,以上条件‘if(chroma_format_idc==3)’可由‘if(chroma_format_idc==3&&!separate_colour_plane_flag)’替代。

[0216] 此外,当color_transform_enabled_flag等于1时可应用约束,chroma_format_idc可等于3。替代地,此外,当color_transform_enabled_flag等于1时,separate_colour_plane_flag 可等于0。

[0217] 在一些实例中,译码单元可具有以下语法:

[0218]

coding_unit(x0, y0, log2CbSize) {	描述符
if(transquant_bypass_enabled_flag)	
cu_transquant_bypass_flag	ae(v)
if(slice_type != I)	
cu_skip_flag[x0][y0]	ae(v)
nCbS = (1 << log2CbSize)	
if(cu_skip_flag[x0][y0])	
prediction_unit(x0, y0, nCbS, nCbS)	
else {	
if(slice_type != I)	
pred_mode_flag	ae(v)
if(CuPredMode[x0][y0] != MODE_INTRA log2CbSize == MinCbLog2SizeY)	
part_mode	ae(v)
if(CuPredMode[x0][y0] == MODE_INTRA) {	
if(PartMode == PART_2Nx2N && pcm_enabled_flag && log2CbSize >= Log2MinIpcmCbSizeY && log2CbSize <= Log2MaxIpcmCbSizeY)	
pcm_flag[x0][y0]	ae(v)
if(pcm_flag[x0][y0]) {	
while(!byte_aligned())	
pcm_alignment_zero_bit	f(1)
pcm_sample(x0, y0, log2CbSize)	
} else {	
if(color_transform_enabled_flag) {	
color_transform_flag[x0][y0]	ae(v)
}	
pbOffset = (PartMode == PART_NxN) ? (nCbS / 2) : nCbS	
for(j = 0; j < nCbS; j = j + pbOffset)	
for(i = 0; i < nCbS; i = i + pbOffset)	
prev_intra_luma_pred_flag[x0 + i][y0 + j]	ae(v)
for(j = 0; j < nCbS; j = j + pbOffset)	
for(i = 0; i < nCbS; i = i + pbOffset)	
if(prev_intra_luma_pred_flag[x0 + i][y0 + j])	
mpm_idx[x0 + i][y0 + j]	ae(v)
else	

[0219]

rem_intra_luma_pred_mode[x0 + i][y0 + j]	ae(v)
if(ChromaArrayType == 3 && !color_transform_flag[x0][y0])	
for(j = 0; j < nCbS; j = j + pbOffset)	
for(i = 0; i < nCbS; i = i + pbOffset)	
intra_chroma_pred_mode[x0 + i][y0 + j]	ae(v)
else if(ChromaArrayType != 0)	
intra_chroma_pred_mode[x0][y0]	ae(v)
}	

	} else {	
	...	
	}	
	}	
	if(!pcm_flag[x0][y0]) {	
	if(CuPredMode[x0][y0] != MODE_INTRA && !(PartMode == PART_2Nx2N && merge_flag[x0][y0]))	
	rqt_root_cbf	ae(v)
	if(rqt_root_cbf) {	
	if(color_transform_enabled_flag && CuPredMode[x0][y0] != MODE_INTRA) {	
[0220]	color_transform_flag[x0][y0]	ae(v)
	}	
	MaxTrafoDepth = (CuPredMode[x0][y0] == MODE_INTRA ? (max_transform_hierarchy_depth_intra + IntraSplitFlag) : max_transform_hierarchy_depth_inter)	
	transform_tree(x0, y0, x0, y0, log2CbSize, 0, 0)	
	}	
	}	
	}	
	}	

[0221] 在以上实例中,对于帧内模式,首先用信号发送颜色变换旗标。当此旗标等于1时,可跳过用信号发送intra_chroma_pred_mode,其中色度分量共享与亮度相同的模式。

[0222] 替代地,在一些实例中,译码单元可具有以下语法:

	coding_unit(x0, y0, log2CbSize) {	描述符
	if(transquant_bypass_enabled_flag)	
	cu_transquant_bypass_flag	ae(v)
	if(slice_type != I)	
	cu_skip_flag[x0][y0]	ae(v)
	nCbS = (1 << log2CbSize)	
	if(cu_skip_flag[x0][y0])	
	prediction_unit(x0, y0, nCbS, nCbS)	
	else {	
	if(slice_type != I)	
	pred_mode_flag	ae(v)
	if(CuPredMode[x0][y0] != MODE_INTRA log2CbSize == MinCbLog2SizeY)	
	part_mode	ae(v)
	if(CuPredMode[x0][y0] == MODE_INTRA) {	
	if(PartMode == PART_2Nx2N && pcm_enabled_flag && log2CbSize >= Log2MinIpcmCbSizeY && log2CbSize <= Log2MaxIpcmCbSizeY)	
	pcm_flag[x0][y0]	ae(v)
	if(pcm_flag[x0][y0]) {	
[0223]		

[0224]

while(!byte_aligned())	
pcm_alignment_zero_bit	f(1)
pcm_sample(x0, y0, log2CbSize)	
} else {	
pbOffset = (PartMode == PART_NxN) ? (nCbS / 2) : nCbS	
for(j = 0; j < nCbS; j = j + pbOffset)	
for(i = 0; i < nCbS; i = i + pbOffset)	
prev_intra_luma_pred_flag [x0 + i][y0 + j]	ae(v)
for(j = 0; j < nCbS; j = j + pbOffset)	
for(i = 0; i < nCbS; i = i + pbOffset)	
if(prev_intra_luma_pred_flag[x0 + i][y0 + j])	
mpm_idx [x0 + i][y0 + j]	ae(v)
else	
rem_intra_luma_pred_mode [x0 + i][y0 + j]	ae(v)
if(ChromaArrayType == 3)	
for(j = 0; j < nCbS; j = j + pbOffset)	
for(i = 0; i < nCbS; i = i + pbOffset)	
intra_chroma_pred_mode [x0 + i][y0 + j]	ae(v)
else if(ChromaArrayType != 0)	
intra_chroma_pred_mode [x0][y0]	ae(v)
}	
} else {	
...	
}	
}	
if(!pcm_flag[x0][y0]) {	
if(CuPredMode[x0][y0] != MODE_INTRA && !(PartMode == PART_2Nx2N && merge_flag[x0][y0]))	
rqt_root_cbf	ae(v)
if(rqt_root_cbf) {	
if(color_transform_enabled_flag &&(CuPredMode[x0][y0] == MODE_INTER !intra_chroma_pred_mode[x0][y0])) {	
color_transform_flag [x0][y0]	ae(v)
}	
MaxTrafoDepth = (CuPredMode[x0][y0] == MODE_INTRA ? (max_transform_hierarchy_depth_intra + IntraSplitFlag) : max_transform_hierarchy_depth_inter)	
transform_tree(x0, y0, x0, y0, log2CbSize, 0, 0)	
}	
}	
}	
}	

[0225] 替代地,当帧内BC模式被视为帧内模式时,也就是说,相应的CuPredMode[x0][y0]等于MODE_INTRA时,以上突出显示的条件‘if(color_transform_enabled_flag && (CuPredMode[x0][y0] == MODE_INTER || !intra_chroma_pred_mode[x0][y0]))’可由‘if(color_transform_enabled_flag && (CuPredMode[x0][y0] == MODE_INTRA || intra_bc_flag[x0][y0] || !intra_chroma_pred_mode[x0][y0]))’替代。替代地,在所有以上实例中,CuPredMode[x0][y0] == MODE_INTRA可由CuPredMode[x0][y0] != MODE_INTRA替代。

[0226] 替代地,以上条件‘if(color_transform_enabled_flag&&(CuPredMode[x0][y0]==MODE_INTER||!intra_chroma_pred_mode[x0][y0]))’可简单地由‘if(color_transform_enabled_flag)’替代。在这种情况下,可符合当color_transform_enabled_flag等于1且当前CU经帧内译码时色度及亮度模式相同的约束。

[0227] 在当前CU/PU/TU未经无损译码时(即,当cu_transquant_bypass_flag等于0时)可调用以下变化。在一个实例中,当应用颜色变换时,解量化过程中使用的QP可变化。然而,解块过程中使用的 Q_{PY} 可无变化,即,不具有要考虑的增量QP(ΔQ_{PCO})。

[0228] 在解码过程中,对于针对量化参数的导出过程,对此过程的输入是指定当前亮度译码块的左上方样本相对于当前图片的左上方亮度样本的亮度位置(x_{Cb}, y_{Cb})。在此过程中,导出变量 Q_{PY} 、亮度量化参数 $Q_{P'Y}$ 及色度量化参数 $Q_{P'Cb}$ 及 $Q_{P'Cr}$ 。

[0229] 亮度位置(x_{Qg}, y_{Qg})指定当前量化分组的左上方亮度样本相对于当前图片的左上方亮度样本。水平及垂直位置 x_{Qg} 及 y_{Qg} 分别设定成等于 $x_{Cb} - (x_{Cb} \& ((1 < \log_2 \text{MinCuQpDeltaSize}) - 1))$ 及 $y_{Cb} - (y_{Cb} \& ((1 < \log_2 \text{MinCuQpDeltaSize}) - 1))$ 。量化分组的亮度大小 $\log_2 \text{MinCuQpDeltaSize}$ 确定共享相同 q_{PY_PRED} 的译码树块内的最小区域的亮度大小。

[0230] 可通过以下有序的步骤导出经预测亮度量化参数 q_{PY_PRED} :1)可导出变量 q_{PY_PREV} 。如果以下条件中的一或多者成立,那么 q_{PY_PREV} 设定成等于Slice Q_{PY} :当前量化分组是切片中的第一量化分组、当前量化分组是瓦片中的第一量化分组、或当前量化分组是译码树块行中的第一量化分组,且entropy_coding_sync_enabled_flag等于1。否则, q_{PY_PREV} 设定成等于按解码次序在先前量化分组中的最后一个译码单元的亮度量化参数 Q_{PY} 。

[0231] 2)以设定成等于(x_{Cb}, y_{Cb})的位置(x_{Curr}, y_{Curr})及设定成等于($x_{Qg}-1, y_{Qg}$)的相邻位置(x_{NbY}, y_{NbY})作为输入而调用z扫描次序中的块的可用性导出过程,并且将输出指派给availableA。导出变量 q_{PY_A} 如下:如果以下条件中的一或多者成立,那么 q_{PY_A} 设定成等于 q_{PY_PREV} :availableA等于FALSE,或含有覆盖亮度位置($x_{Qg}-1, y_{Qg}$)的亮度译码块的译码树块的译码树块地址ctbAddrA不等于CtbAddrInTs,其中ctbAddrA导出如下:

[0232] $x_{Tmp} = (x_{Qg}-1) \gg \log_2 \text{MinTrafoSize}$

[0233] $y_{Tmp} = y_{Qg} \gg \log_2 \text{MinTrafoSize}$

[0234] $\text{minTbAddrA} = \text{MinTbAddrZs}[x_{Tmp}][y_{Tmp}]$

[0235] $\text{ctbAddrA} = (\text{minTbAddrA} \gg 2) * (\text{CtbLog2SizeY} - \log_2 \text{MinTrafoSize})$

[0236] 否则, q_{PY_A} 设定成等于含有覆盖($x_{Qg}-1, y_{Qg}$)的亮度译码块的译码单元的亮度量化参数 Q_{PY} 。

[0237] 3)以设定成等于(x_{Cb}, y_{Cb})的位置(x_{Curr}, y_{Curr})及设定成等于($x_{Qg}, y_{Qg}-1$)的相邻位置(x_{NbY}, y_{NbY})作为输入而调用z扫描次序中的块的可用性导出过程。将输出指派给availableB。导出变量 q_{PY_B} 。如果以下条件中的一或多者成立,那么 q_{PY_B} 设定成等于 q_{PY_PREV} :availableB等于FALSE,或含有覆盖亮度位置($x_{Qg}, y_{Qg}-1$)的亮度译码块的译码树块的译码树块地址ctbAddrB不等于CtbAddrInTs,其中ctbAddrB导出如下:

[0238] $x_{Tmp} = x_{Qg} \gg \log_2 \text{MinTrafoSize}$

[0239] $y_{Tmp} = (y_{Qg}-1) \gg \log_2 \text{MinTrafoSize}$

[0240] $\text{minTbAddrB} = \text{MinTbAddrZs}[x_{Tmp}][y_{Tmp}]$

[0241] $\text{ctbAddrB} = (\text{minTbAddrB} \gg 2) * (\text{CtbLog2SizeY} - \text{Log2MinTrafoSize})$

[0242] 否则, qPY_B 设定成等于含有覆盖 $(\text{xQg}, \text{yQg}-1)$ 的亮度译码块的译码单元的亮度量化参数 QpY 。

[0243] 经预测亮度量化参数 qPY_PRED 可导出如下:

[0244] $\text{qPY_PRED} = (\text{qPY_A} + \text{qPY_B} + 1) \gg 1$

[0245] 变量 QpY 可导出如下:

[0246] $\text{QpY} = ((\text{qPY_PRED} + \text{CuQpDeltaVal} + 52 + 2 * \text{QpBdOffsetY}) \% (52 + \text{QpBdOffsetY})) - \text{QpBdOffsetY}$

[0247] 亮度量化参数 Qp'_Y 可导出如下:

[0248] $\text{Qp}'_Y = \text{QpY} + \text{QpBdOffsetY}$

[0249] 当 ChromaArrayType 不等于 0 时, 变量 qPicb 及 qPicr 导出如下:

[0250] $\text{qPicb} = \text{Clip3}(-\text{QpBdOffsetc}, 57, \text{QpY} + \text{pps_cb_qp_offset} + \text{slice_cb_qp_offset} + \text{CuQpOffsetcb})$

[0251] $\text{qPicr} = \text{Clip3}(-\text{QpBdOffsetC}, 57, \text{QpY} + \text{pps_cr_qp_offset} + \text{slice_cr_qp_offset} + \text{CuQpOffsetCr})$

[0252] 如果 ChromaArrayType 等于 1, 那么基于索引 qPi 等于 qPicb 及 qPicr 而分别将变量 qPcb 及 qPcr 设定成等于 Qpc 的值。否则, 基于索引 qPi 等于 qPicb 及 qPicr 而分别将变量 qPcb 及 qPcr 设定成等于 $\text{Min}(\text{qPi}, 51)$ 。

[0253] 用于 Cb 及 Cr 分量的色度量化参数 Qp'_{Cb} 及 Qp'_{Cr} 导出如下:

[0254] $\text{Qp}'_{\text{Cb}} = \text{qPcb} + \text{QpBdOffsetc}$

[0255] $\text{Qp}'_{\text{Cr}} = \text{qPcr} + \text{QpBdOffsetc}$

[0256] 对于 ChromaArrayType 等于 1, 取决于 qPi 的 Qpc 的规范如下所示:

[0257]	qPi	<30	30	31	32	33	34	35	36	37	38	39	40	41	42	43	>43
	Qpc	= qPi	29	30	31	32	33	33	34	34	35	35	36	36	37	37	= $\text{qPi}-6$

[0258] 在解量化过程中, 可导出用于每一分量索引 (cIdx) 的量化参数 qP 。对此过程的输入可为指定当前亮度变换块的左上方样本相对于当前图片的左上方亮度样本的亮度位置 (xTbY, yTbY)、指定当前块相对于译码块的层次深度的变量 trafoDepth 、指定当前块的颜色分量的变量 cIdx , 以及指定当前变换块的大小的变量 nTbS 。此过程的输出可为具有元素 $\text{r}[\text{x}][\text{y}]$ 的残余样本 r 的 $(\text{nTbS}) \times (\text{nTbS})$ 阵列。

[0259] 可导出量化参数 qP 。如果 cIdx 等于 0, 则

[0260] $\text{qP} = \text{Qp}'_Y + (\text{color_transform_flag}[\text{xTbY}][\text{yTbY}] ? \text{deltaQPc0} : 0)$

[0261] 否则, 如果 cIdx 等于 1, 则

[0262] $\text{qP} = \text{Qp}'_{\text{Cb}} + (\text{color_transform_flag}[\text{xTbY}][\text{yTbY}] ? \text{deltaQPc1} : 0)$

[0263] 否则 (cIdx 等于 2), 则

[0264] $\text{qP} = \text{Qp}'_{\text{Cr}} + (\text{color_transform_flag}[\text{xTbY}][\text{yTbY}] ? \text{deltaQPc2} : 0)$

[0265] 在一个实例中, 可分别将 deltaQPc0 、 deltaQPc1 及 deltaQPc2 设定成 -5、-5 及 -3。在另一实例中, 解块过程中使用的 QpY 无变化, 即, 具有要考虑的增量 $\text{QP}(\text{deltaQPc0})$ 。在解码过程中, 对于针对量化参数的导出过程, 对此过程的输入可为指定当前亮度译码块的左上方样本相对于当前图片的左上方亮度样本的亮度位置 (xCb, yCb)。在此过程中, 可导出变量 QpY 、亮

度量化参数 Qp'_Y 及色度量化参数 Qp'_{Cb} 及 Qp'_{Cr} 。

[0266] 亮度位置 (xQg, yQg) 指定当前量化分组的左上方亮度样本相对于当前图片的左上方亮度样本。水平及垂直位置 xQg 及 yQg 分别设定成等于 $xCb - (xCb \& ((1 < \log_2 \text{MinCuQpDeltaSize}) - 1))$ 及 $yCb - (yCb \& ((1 < \log_2 \text{MinCuQpDeltaSize}) - 1))$ 。量化分组的亮度大小 $\log_2 \text{MinCuQpDeltaSize}$ 确定共享相同 qPY_{PRED} 的译码树块内的最小区域的亮度大小。

[0267] 可通过以下有序的步骤导出经预测亮度量化参数 $qPY_{\text{PRED}}:1$)可导出变量 qPY_{PREV} 。如果以下条件中的一或多者成立,那么 qPY_{PREV} 设定成等于 SliceQpY :当前量化分组是切片中的第一量化分组、当前量化分组是瓦片中的第一量化分组、或当前量化分组是译码树块行中的第一量化分组,且 $\text{entropy_coding_sync_enabled_flag}$ 等于1。否则, qPY_{PREV} 设定成等于按解码次序在先前量化分组中的最后一个译码单元的亮度量化参数 QpY 。

[0268] 2) 以设定成等于 (xCb, yCb) 的位置 $(xCurr, yCurr)$ 及设定成等于 $(xQg-1, yQg)$ 的相邻位置 $(xNbY, yNbY)$ 作为输入而调用z扫描次序中的块的可用性导出过程,并且将输出指派给 availableA 。导出变量 qPY_A 如下:如果以下条件中的一或多者成立,那么 qPY_A 设定成等于 qPY_{PREV} : availableA 等于FALSE,或含有覆盖亮度位置 $(xQg-1, yQg)$ 的亮度译码块的译码树块的译码树块地址 ctbAddrA 不等于 CtbAddrInTs ,其中 ctbAddrA 导出如下:

[0269] $xTmp = (xQg-1) \gg \log_2 \text{MinTrafoSize}$

[0270] $yTmp = yQg \gg \log_2 \text{MinTrafoSize}$

[0271] $\text{minTbAddrA} = \text{MinTbAddrZs}[xTmp][yTmp]$

[0272] $\text{ctbAddrA} = (\text{minTbAddrA} \gg 2) * (\text{CtbLog2SizeY} - \log_2 \text{MinTrafoSize})$

[0273] 否则, qPY_A 设定成等于含有覆盖 $(xQg-1, yQg)$ 的亮度译码块的译码单元的亮度量化参数 QpY 。

[0274] 3) 以设定成等于 (xCb, yCb) 的位置 $(xCurr, yCurr)$ 及设定成等于 $(xQg, yQg-1)$ 的相邻位置 $(xNbY, yNbY)$ 作为输入而调用z扫描次序中的块的可用性导出过程。将输出指派给 availableB 。导出变量 qPY_B 。如果以下条件中的一或多者成立,那么 qPY_B 设定成等于 qPY_{PREV} : availableB 等于FALSE,或含有覆盖亮度位置 $(xQg, yQg-1)$ 的亮度译码块的译码树块的译码树块地址 ctbAddrB 不等于 CtbAddrInTs ,其中 ctbAddrB 导出如下:

[0275] $xTmp = xQg \gg \log_2 \text{MinTrafoSize}$

[0276] $yTmp = (yQg-1) \gg \log_2 \text{MinTrafoSize}$

[0277] $\text{minTbAddrB} = \text{MinTbAddrZs}[xTmp][yTmp]$

[0278] $\text{ctbAddrB} = (\text{minTbAddrB} \gg 2) * (\text{CtbLog2SizeY} - \log_2 \text{MinTrafoSize})$

[0279] 否则, qPY_B 设定成等于含有覆盖 $(xQg, yQg-1)$ 的亮度译码块的译码单元的亮度量化参数 QpY 。

[0280] 经预测亮度量化参数 qPY_{PRED} 可导出如下:

[0281] $qPY_{\text{PRED}} = (qPY_A + qPY_B + 1) \gg 1$

[0282] 变量 QpY 可导出如下:

[0283] $QpY = ((qPY_{\text{PRED}} + \text{CuQpDeltaVal} + 52 + 2 * \text{QpBdOffsetY}) \% (52 + \text{QpBdOffsetY})) - \text{QpBdOffsetY}$

[0284] $QpY = QpY + (\text{color_transform_flag}[xCb][yCb] ? \text{deltaQPc0} : 0)$

[0285] 亮度量化参数 Qp'_Y 可导出如下:

[0286] $Qp'_Y = Qp_Y + QpBdOffset_Y$

[0287] 当ChromaArrayType不等于0时,变量 qPi_{Cb} 及 qPi_{Cr} 可导出如下:

[0288] $qPi_{Cb} = Clip3(-QpBdOffset_C, 57, Qp_Y + pps_cb_qp_offset + slice_cb_qp_offset + CuQpOffset_{Cb})$

[0289] $qPi_{Cr} = Clip3(-QpBdOffset_C, 57, Qp_Y + pps_cr_qp_offset + slice_cr_qp_offset + CuQpOffset_{Cr})$

[0290] 如果ChromaArrayType等于1,那么可基于索引 qPi 等于 qPi_{Cb} 及 qPi_{Cr} 而分别将变量 qP_{Cb} 及 qP_{Cr} 设定成等于 Qp_C 的值。否则,可基于索引 qPi 等于 qPi_{Cb} 及 qPi_{Cr} 而分别将变量 qP_{Cb} 及 qP_{Cr} 设定成等于 $\min(qPi, 51)$ 。用于Cb及Cr分量的色度量化参数 Qp'_{Cb} 及 Qp'_{Cr} 可导出如下:

[0291] $Qp'_{Cb} = qP_{Cb} + QpBdOffset_C$

[0292] $Qp'_{Cr} = qP_{Cr} + QpBdOffset_C$

[0293] 对于ChromaArrayType等于1,取决于 qPi 的 Qp_C 的规范可如下所示:

[0294]

qPi	<30	30	31	32	33	34	35	36	37	38	39	40	41	42	43	>43
Qp_C	= qPi	29	30	31	32	33	33	34	34	35	35	36	36	37	37	= $qPi-6$

[0295] $Qp'_{Cb} = Qp'_{Cb} + (\text{color_transform_flag}[x_{Cb}][y_{Cb}]?deltaQP_{C1}:0)$

[0296] $Qp'_{Cr} = Qp'_{Cr} + (\text{color_transform_flag}[x_{Cb}][y_{Cb}]?deltaQP_{C2}:0)$

[0297] 在解量化过程中,可导出用于每一分量索引($cIdx$)的量化参数 qP 如下。如果 $cIdx$ 等于0,则

[0298] $qP = Qp'_Y + (\text{color_transform_flag}[x_{TbY}][y_{TbY}]?deltaQP_{C0}:0)$

[0299] 否则,如果 $cIdx$ 等于1,则

[0300] $qP = Qp'_{Cb} + (\text{color_transform_flag}[x_{TbY}][y_{TbY}]?deltaQP_{C1}:0)$

[0301] 否则($cIdx$ 等于2),

[0302] $qP = Qp'_{Cr} + (\text{color_transform_flag}[x_{TbY}][y_{TbY}]?deltaQP_{C2}:0)$

[0303] 在一个实例中,可分别将 $deltaQP_{C0}$ 、 $deltaQP_{C1}$ 及 $deltaQP_{C2}$ 设定成-5、-5及-3。

[0304] 现将描述一些实例实施方案详情。下文相较于大卫·弗林(David Flynn)等人的JCTVC-Q1005_v4(“高效率视频译码(HEVC)范围扩展文字规范:草案7(High Efficiency Video Coding(HEVC) Range Extensions text specification:Draft 7)”,JCTVC-Q1005_v4,关于ITU-T SG 16WP 3及ISO/IEC JTC 1/SC 29/WG 11的视频译码的联合合作小组(JCT-VC),第17次会议:巴伦西亚,ES,2014年3月27日到4月4日)中的语法元素及语义来描述语法元素及语义变化,JCTVC-Q1005_v4可从以下网址获得:

[0305] http://phenix.int-evry.fr/jct/doc_end_user/documents/17_Valencia/wg11/JCTVC-Q1005-v4.zip

[0306] 新添加的部分以粗体或斜体(即,粗体或斜体)突出显示,并且删除的部分标记为方括号中的粗体,例如,[[删除的文字]]。斜体部分与本发明中提出的技术有关。

语法元素及语义

[0307]

7.3.2.2 序列参数集 RBSP 语法

[0308]

seq_parameter_set_rbsp() {	描述符
sps_video_parameter_set_id	u(4)
sps_max_sub_layers_minus1	u(3)
sps_temporal_id_nesting_flag	u(1)
profile_tier_level(sps_max_sub_layers_minus1)	
...	
vui_parameters_present_flag	u(1)
if(vui_parameters_present_flag)	
vui_parameters()	
sps_extension_present_flag	u(1)
if(sps_extension_present_flag) {	
for(i = 0; i < 1; i++)	
sps_extension_flag[i]	u(1)
sps_extension_7bits	u(7)
if(sps_extension_flag[0]) {	
transform_skip_rotation_enabled_flag	u(1)
transform_skip_context_enabled_flag	u(1)
intra_block_copy_enabled_flag	u(1)
implicit_rdpcm_enabled_flag	u(1)
explicit_rdpcm_enabled_flag	u(1)
extended_precision_processing_flag	u(1)
intra_smoothing_disabled_flag	u(1)
high_precision_offsets_enabled_flag	u(1)
fast_rice_adaptation_enabled_flag	u(1)
cabac_bypass_alignment_enabled_flag	u(1)
color_transform_enabled_flag	u(1)
lossless_enable_flag	u(1)
}	
if(sps_extension_7bits)	
while(more_rbsp_data())	
sps_extension_data_flag	u(1)
}	
rbsp_trailing_bits()	
}	

[0309] color_transform_enabled_flag等于1指示启用颜色变换。color_transform_enabled_flag 等于0指示不启用颜色变换。

[0310] lossless_enable_flag等于1指示应用无损译码。另外,当color_transform_enabled_flag 等于1时,使用原始YCoCg-R变换。

[0311] lossless_enable_flag等于0指示应用有损译码。另外,当color_transform_enabled_flag 等于1时,使用原始YCoCg变换。

[0312] **7.3.5.8 译码单元语法**

[0313]

coding_unit(x0, y0, log2CbSize) {	描述符
if(transquant_bypass_enabled_flag)	
cu_transquant_bypass_flag	ae(v)
if(slice_type != I)	
cu_skip_flag[x0][y0]	ae(v)
nCbs = (1 << log2CbSize)	
if(cu_skip_flag[x0][y0])	
prediction_unit(x0, y0, nCbs, nCbs)	
else {	
if(slice_type != I)	
pred_mode_flag	ae(v)
if(CuPredMode[x0][y0] != MODE_INTRA	
log2CbSize == MinCbLog2SizeY)	
part_mode	ae(v)
if(CuPredMode[x0][y0] == MODE_INTRA) {	
if(PartMode == PART_2Nx2N && pcm_enabled_flag &&	
log2CbSize >= Log2MinIpcmCbSizeY &&	
log2CbSize <= Log2MaxIpcmCbSizeY)	
pcm_flag[x0][y0]	ae(v)
if(pcm_flag[x0][y0]) {	
while(!byte_aligned())	
pcm_alignment_zero_bit	f(1)
pcm_sample(x0, y0, log2CbSize)	
} else {	
pbOffset = (PartMode == PART_NxN) ? (nCbs / 2) : nCbs	
for(j = 0; j < nCbs; j = j + pbOffset)	
for(i = 0; i < nCbs; i = i + pbOffset)	
prev_intra_luma_pred_flag[x0 + i][y0 + j]	ae(v)
for(j = 0; j < nCbs; j = j + pbOffset)	
for(i = 0; i < nCbs; i = i + pbOffset)	
if(prev_intra_luma_pred_flag[x0 + i][y0 + j])	
mpm_idx[x0 + i][y0 + j]	ae(v)
else	
rem_intra_luma_pred_mode[x0 + i][y0 + j]	ae(v)
if(ChromaArrayType == 3)	
for(j = 0; j < nCbs; j = j + pbOffset)	
for(i = 0; i < nCbs; i = i + pbOffset)	
intra_chroma_pred_mode[x0 + i][y0 + j]	ae(v)
else if(ChromaArrayType != 0)	
intra_chroma_pred_mode[x0][y0]	ae(v)
}	
} else {	
...	
}	

	}	
	if(!pcm_flag[x0][y0]) {	
	if(CuPredMode[x0][y0] != MODE_INTRA && !(PartMode == PART_2Nx2N && merge_flag[x0][y0]))	
	rqt_root_cbf	ae(v)
	if(rqt_root_cbf) {	
	if(color_transform_enabled_flag	
	&&(CuPredMode[x0][y0] == MODE_INTER	
	 !intra_chroma_pred_mode[x0][y0]) {	
	color_transform_flag[x0][y0]	ae(v)
	}	
[0314]	else {	
	color_transform_flag[x0][y0] = defaultVal	
	}	
	MaxTrafoDepth = (CuPredMode[x0][y0] == MODE_INTRA ? (max_transform_hierarchy_depth_intra + IntraSplitFlag) : max_transform_hierarchy_depth_inter)	
	transform_tree(x0, y0, x0, y0, log2CbSize, 0, 0)	
	}	
	}	
	}	
	}	

应注意，仅使用一个经帧内译码 CU 内的左上方 PU 的色度模式
 [0315] (即，!intra_chroma_pred_mode[x0][y0]) 以确定是否应用信号发送
color_transform_flag。

[0316] 在一个实例中，常量defaultVal始终设定成等于0。

[0317] 在另一实例中，常量defaultVal始终设定成等于SPS/PPS/切片标头中的颜色变换
 旗标的启用，例如，color_transform_enabled_flag。

[0318] 现将描述第一实例，下文称为实例#1。如上文所指示，在此实例#1中新添加的部分
 以粗体或斜体(即，粗体或斜体)突出显示，并且删除的部分标记为方括号中的粗体，例如，
 [[删除的文字]]。斜体部分与本发明中提出的技术有关。

[0319] 在此实例中，defaultVal等于0。

[0320] 8.4.1以帧内预测模式译码的译码单元的一般解码过程

[0321] 对此过程的输入是：

[0322] -...

[0323] 此过程的输出是解决滤波之前的修改后的经重构图片。

[0324] 以亮度位置(xCb,yCb)作为输入而调用如子条款8.6.1中指定的量化参数的导出
 过程。

[0325] 变量nCbs设定成等于1<<log2CbSize。

[0326] 取决于pcm_flag[xCb][yCb]及IntraSplitFlag的值，亮度样本的解码过程指定如
 下：

[0327] -如果pcm_flag[xCb][yCb]等于1，那么经重构图片修改如下：

[0328] ...

[0329] -否则 (pcm_flag[xCb][yCb]等于0),如果IntraSplitFlag等于0,那么适用以下有序的步骤:

[0330] 1.当intra_bc_flag[xCb][yCb]等于0时,以亮度位置(xCb,yCb)作为输入而调用如子条款8.4.2中指定的帧内预测模式的导出过程。

[0331] 2.当intra_bc_flag[xCb][yCb]等于1时,以亮度位置(xCb,yCb)及变量log2CbSize作为输入而调用如子条款8.4.4中指定的帧内块复制预测模式中的块向量分量的导出过程,且输出为bvIntra。

3. 如果 color_transform_flag[xCb][yCb]等于 1, 那么适用以下:

- 对于以值 0..2 继续进行的变量 cIdx, 适用以下有序的步骤:

- 设定变量 comp 等于(!cIdx ? L : (cIdx == 1 ? Cb : Cr))。

[0332] - 以位置(xCb, yCb), 设定成等于 log2CbSize 的变量 log2TrafoSize、设定成等于 0 的变量 trafoDepth、设定成等于 IntraPredModeY[xCb][yCb] 的变量 predModeIntra、设定成等于 intra_bc_flag[xCb][yCb] 的变量 predModeIntraBc、变量 bvIntra、变量 cIdx 及等于 1 的变量 controlPara 作为输入而调用如子条款 8.4.4.1 中指定的帧内块的一般解码过程, 且输出为残余样本阵列 resSamples_{comp}。

- 以设定成等于 nCbS 的变量 blkSize、设定成等于 resSamples_L 的 (nCbS)×(nCbS)阵列 r_Y、设定成等于 resSamples_{Cb} 的 (nCbS)×(nCbS)阵列 r_{Cb} 及设定成等于 resSamples_{Cr} 的 (nCbS)×(nCbS)阵列 r_{Cr} 作为输入而调用如子条款 8.6.7 中指定的使用颜色空间转换的残余块的残余修改过程, 且输出为(nCbS)×(nCbS)阵列 resSamples_L、resSamples_{Cb} 及 resSamples_{Cr} 的修改后的型式。

[0333] 4.以亮度位置(xCb,yCb)、设定成等于log2CbSize的变量log2TrafoSize、设定成等于0的变量trafoDepth、设定成等于IntraPredModeY[xCb][yCb]的变量 predModeIntra、设定成等于intra_bc_flag[xCb][yCb]的变量predModeIntraBc、变量bvIntra、**|| 及 ||** 设定成等于0的变量cIdx、**及 等 于**

(color_transform_flag[xCb][yCb] ? 2 : 3)的变量 controlPara 作为输入而调用如子条款8.4.4.1中指定的帧内块的一般解码过程,且输出为解决块滤波之前的修改后的经重构图片。

[0334] -否则 (pcm_flag[xCb][yCb]等于0且IntraSplitFlag等于1),对于以值0..3继续进行的变量blkIdx,适用以下有序的步骤:

[0335] 1.变量xPb设定成等于xCb+ (nCbS>>1)*(blkIdx%2)。

[0336] 2.变量yPb设定成等于yCb+ (nCbS>>1)*(blkIdx/2)。

[0337] 3.以亮度位置(xPb,yPb)作为输入而调用如子条款8.4.2中指定的帧内预测模式

的导出过程。

4. 如果 **color_transform_flag[xCb][yCb]** 等于 1, 那么适用以下:

- 对于以值 0..2 继续进行的变量 **cIdx**, 适用以下有序的步骤:

- 设定变量 **comp** 等于 **(!cIdx ? L : (cIdx == 1 ? Cb : Cr))**。

- 以亮度位置位置(**xPb, yPb**)、设定成等于 **log2CbSize - 1** 的变量 **log2TrafoSize**、设定成等于 1 的变量 **trafoDepth**、设定成等于 **IntraPredModeY[xPb][yPb]** 的变量 **predModeIntra**、设定成等于 0 的变量 **predModeIntraBc**、变量 **cIdx** 及设定成等于 1 的变量 **controlPara** 作为输入而调用如子条款 8.4.4.1 中指定的帧内块的一般解码过程, 且输出为残余样本阵列 **resSamples_{comp}**。

[0338]

- 设定变量 **nSubCbS** 等于 **(nCbS >> 1)**, 且以设定成等于 **nSubCbS** 的变量 **blkSize**、设定成等于 **resSamples_L** 的 **(nSubCbS)×(nSubCbS)** 阵列 **r_Y**、设定成等于 **resSamples_{Cb}** 的 **(nSubCbS)×(nSubCbS)** 阵列 **r_{Cb}**、及设定成等于 **resSamples_{Cr}** 的 **(nSubCbS)×(nSubCbS)** 阵列 **r_{Cr}** 作为输入而调用如子条款 8.6.7 中指定的使用颜色空间转换的残余块的残余修改过程, 且输出为 **(nSubCbS)×(nSubCbS)** 阵列 **resSamples_L**、**resSamples_{Cb}** 及 **resSamples_{Cr}** 的修改后的型式。

[0339] 5. 以亮度位置(**xPb, yPb**)、设定成等于 **log2CbSize-1** 的变量 **log2TrafoSize**、设定成等于 1 的变量 **trafoDepth**、设定成等于 **IntraPredModeY[xPb][yPb]** 的变量 **predModeIntra**、设定成等于 0 的变量 **predModeIntraBc**、**及** 设定成等于 0 的变量 **cIdx** **及** 设定成等于 **(color_transform_flag[xCb][yCb] ? 2 : 3)** 的变量 **controlPara** 作为输入而调用如子条款 8.4.4.1 中指定的帧内块的一般解码过程, 且输出为解决块滤波之前的修改后的经重构图片。

[0340] 当 **ChromaArrayType** 不等于 0 时, 适用以下。

[0341] 变量 **log2CbSizeC** 设定成等于

[0342] **log2CbSize - (ChromaArrayType == 3 ? 0 : 1)**。

[0343] 取决于 **pcm_flag[xCb][yCb]** 及 **IntraSplitFlag** 的值, 色度样本的解码过程指定如下:

[0344] - 如果 **pcm_flag[xCb][yCb]** 等于 1, 那么经重构图片修改如下:

[0345] ...

[0346] - 否则 (**pcm_flag[xCb][yCb]** 等于 0), 如果 **IntraSplitFlag** 等于 0 或 **ChromaArrayType** 不等于 3, 那么适用以下有序的步骤:

[0347] 1. 当 **intra_bc_flag[xCb][yCb]** 等于 0 时, 以亮度位置(**xCb, yCb**) 作为输入而调用如 8.4.3 中指定的色度帧内预测模式的导出过程, 且输出为变量 **IntraPredModeC**。

[0348] 2.以色度位置($x_{Cb}/SubWidthC, y_{Cb}/SubHeightC$)、设定成等于 $\log2CbSizeC$ 的变量 $\log2TrafoSize$ 、设定成等于0的变量 $trafoDepth$ 、设定成等于 $IntraPredModeC$ 的变量 $predModeIntra$ 、设定成等于 $intra_bc_flag[x_{Cb}][y_{Cb}]$ 的变量 $predModeIntraBc$ 、变量 $bvIntra$ 、**[[及]]**设定成等于1的变量 $cIdx$ 、**及设定成等于**

(color_transform_flag[xCb][yCb] ? 2 : 3)的变量 controlPara作为输入而调用如子条款8.4.4.1中指定的帧内块的一般解码过程,且输出为解块滤波之前的修改后的经重构图片。

[0349] 3.以色度位置($x_{Cb}/SubWidthC, y_{Cb}/SubHeightC$)、设定成等于 $\log2CbSizeC$ 的变量 $\log2TrafoSize$ 、设定成等于0的变量 $trafoDepth$ 、设定成等于 $IntraPredModeC$ 的变量 $predModeIntra$ 、设定成等于 $intra_bc_flag[x_{Cb}][y_{Cb}]$ 的变量 $predModeIntraBc$ 、变量 $bvIntra$ 、**[[及]]**设定成等于2的变量 $cIdx$ 、**及设定成等于**

(color_transform_flag[xCb][yCb] ? 2 : 3)的变量 controlPara作为输入而调用如子条款8.4.4.1中指定的帧内块的一般解码过程,且输出为解块滤波之前的修改后的经重构图片。

[0350] -否则($pcm_flag[x_{Cb}][y_{Cb}]$ 等于0、 $IntraSplitFlag$ 等于1且 $ChromaArrayType$ 等于3),对于以值0..3继续进行的变量 $blkIdx$,适用以下有序的步骤:

[0351] 1.变量 xPb 设定成等于 $x_{Cb} + (nCbS >> 1) * (blkIdx \% 2)$ 。

[0352] 2.变量 yPb 设定成等于 $y_{Cb} + (nCbS >> 1) * (blkIdx / 2)$ 。

[0353] 3.以亮度位置(xPb, yPb)作为输入而调用如8.4.3中指定的色度帧内预测模式的导出过程,且输出为变量 $IntraPredModeC$ 。

[0354] 4.以色度位置(xPb, yPb)、设定成等于 $\log2CbSizeC-1$ 的变量 $\log2TrafoSize$ 、设定成等于1的变量 $trafoDepth$ 、设定成等于 $IntraPredModeC$ 的变量 $predModeIntra$ 、设定成等于0的变量 $predModeIntraBc$ 、**[[及]]**设定成等于1的变量 $cIdx$ 、**及设定成等于(color_transform_flag[xCb][yCb] ? 2 : 3)的变量 controlPara**作为输入而调用如子条款8.4.4.1中指定的帧内块的一般解码过程,且输出为解块滤波之前的修改后的经重构图片。

[0355] 5.以色度位置(xPb, yPb)、设定成等于 $\log2CbSizeC-1$ 的变量 $\log2TrafoSize$ 、设定成等于1的变量 $trafoDepth$ 、设定成等于 $IntraPredModeC$ 的变量 $predModeIntra$ 、设定成等于0的变量 $predModeIntraBc$ 、**[[及]]**设定成等于2的变量 $cIdx$ 、**及设定成等于(color_transform_flag[xCb][yCb] ? 2 : 3)的变量 controlPara**作为输入而调用如子条款8.4.4.1中指定的帧内块的一般解码过程,且输出为解块滤波之前的修改后的经重构图片。

[0356] 8.4.4.1 帧内块的一般解码过程

[0357] 对此过程的输入是:

[0358] -...

[0359] -指定当前块的颜色分量的变量 $cIdx$ 。

[0360] - **变量 controlPara 指定适用过程。**

[0361] 此过程的输出**当 controlPara 不等于 1 时**为解决块滤波之前的修改后的经重构图片, **或当 controlPara 等于 1 时**为残余样本阵列。

[0362] 指定当前亮度变换块的左上方样本相对于当前图片的左上方亮度样本的亮度样本位置 (xTbY, yTbY) 导出如下:

[0363] $(xTbY, yTbY) = (cIdx == 0) ? (xTb0, yTb0) : (xTb0 * SubWidthC, yTb0 * SubHeightC)$ (8-26)

[0364] 变量splitFlag导出如下:

[0365] -如果cIdx等于0,那么splitFlag设定成等于 split_transform_flag[xTbY][yTbY][trafoDepth]。

[0366] -否则,如果以下所有条件成立,那么splitFlag设定成等于1。

[0367] -cIdx大于0

[0368] -split_transform_flag[xTbY][yTbY][trafoDepth]等于1

[0369] -log2TrafoSize大于2

[0370] -否则,splitFlag设定成等于0。

[0371] 取决于splitFlag的值,适用以下:

[0372] -如果splitFlag等于1,适用以下有序的步骤:

[0373] ...

[0374] -否则(splitFlag等于0),对于以值0..(cIdx>0&&ChromaArrayType== 2?1:0)继续进行的变量blkIdx,适用以下有序的步骤:

[0375] 1. 变量nTbS设定成等于 $1 \ll \log_2 \text{TrafoSize}$ 。

[0376] 2. 变量yTbOffset设定成等于blkIdx*nTbS。

[0377] 3. 变量yTbOffsetY设定成等于yTbOffset*SubHeightC。

[0378] 4. **当 controlPara 不等于 2 时**,变量residualDpcm导出如下:

[0379] -如果以下所有条件成立,那么residualDpcm设定成等于1。

[0380] -implicit_rdpdm_enabled_flag等于1。

[0381] -或transform_skip_flag[xTbY][yTbY+yTbOffsetY][cIdx] 等于1,或cu_transquant_bypass_flag等于1。

[0382] -或predModeIntra等于10,或predModeIntra等于26。

[0383] -否则,residualDpcm设定成等于explicit_rdpdm_flag[xTbY][yTbY+yTbOffsetY][cIdx]。

[0384] 5. **当 controlPara 不等于 1 时**,取决于predModeIntraBc的值,适用以下:

[0385] -当predModeIntraBc等于0时,以变换块位置 (xTb0, yTb0+ yTbOffset)、帧内预测模式predModeIntra、变换块大小nTbS及变量cIdx作为输入而调用如子条款8.4.4.2.1中指定的一般帧内采样预测过程,且输出为 (nTbS) × (nTbS) 阵列predSamples。

[0386] -否则(predModeIntraBc等于1),以变换块位置 (xTb0, yTb0+ yTbOffset)、变换块大小nTbS、变量trafoDepth、变量bvIntra及变量cIdx作为输入而调用如子条款8.4.4.2.7中指定的帧内块复制过程,且输出为 (nTbS) × (nTbS) 阵列predSamples。

[0387] 6. 当 **controlPara 不等于 2 时**, 以亮度位置 (xTbY, yTbY+yTbOffsetY)、变量 trafoDepth、变量 cIdx 及设定成等于 nTbS 的变换大小 trafoSize 作为输入而调用如子条款 8.6.2 中指定的缩放及变换过程, 且输出为 (nTbS) × (nTbS) 阵列 resSamples。

[0388] 7. 当 **controlPara 不等于 2 且 residualDpcm 等于 1 时**, 取决于 predModeIntraBc 的值, 适用以下:

[0389] - 当 predModeIntraBc 等于 0 时, 以设定成等于 predModeIntra/26 的变量 mDir、变量 nTbS、及设定成等于阵列 resSamples 的 (nTbS) × (nTbS) 阵列 r 作为输入而调用如子条款 8.6.5 中指定的使用绕过变换的用于块的定向残余修改过程, 且输出为修改后的 (nTbS) × (nTbS) 阵列 resSamples。

[0390] - 否则, (predModeIntraBc 等于 1), 以设定成等于 explicit_rdp_pcm_dir_flag [xTbY] [yTbY+yTbOffsetY] [cIdx] 的变量 mDir、变量 nTbS 及设定成等于阵列 resSamples 的 (nTbS) × (nTbS) 阵列 r 作为输入而调用如子条款 8.6.5 中指定的使用绕过变换的用于块的定向残余修改过程, 且输出为修改后的 (nTbS) × (nTbS) 阵列 resSamples。

[0391] 8. 当 **controlPara 不等于 2 且 cross_component_prediction_enabled_flag 等于 1、ChromaArrayType 等于 3 且 cIdx 不等于 0 时**, 以当前亮度变换块位置 (xTbY, yTbY)、变量 nTbS、变量 cIdx、设定成等于当前变换块的相应亮度残余样本阵列 resSamples 的 (nTbS) × (nTbS) 阵列 r_Y, 及设定成等于阵列 resSamples 的 (nTbS) × (nTbS) 阵列 r 作为输入而调用如子条款 8.6.6 中指定的使用跨分量预测的变换块的残余修改过程, 且输出为修改后的 (nTbS) × (nTbS) 阵列 resSamples。

[0392] 9. 当 **controlPara 不等于 1 时**, 以变换块位置 (xTb0, yTb0+yTbOffset)、均设定成等于 nTbS 的变量 nCurrSw 及 nCurrSh、变量 cIdx、(nTbS) × (nTbS) 阵列 predSamples 及 (nTbS) × (nTbS) 阵列 resSamples 作为输入而调用如子条款 8.6.6 中指定的颜色分量的环路内滤波之前的图片重构过程。

8.5.4 以帧间预测模式译码的译码单元的残余信号的解码过程

[0393]

8.5.4.1 综述

[0394] 对此过程的输入是:

[0395] -...

[0396] 此过程的输出是:

[0397] ...

[0398] 取决于 rqt_root_cbf 的值, 适用以下:

[0399] - 如果 rqt_root_cbf 等于 0 或 skip_flag [xCb] [yCb] 等于 1, 那么 (nC_{bSL}) × (nC_{bSL}) 阵列 resSamples_L 的所有样本设定成等于 0, 并且当 ChromaArrayType 不等于 0 时, 两个 (nC_{bSwc}) × (nC_{bShc}) 阵列 resSamples_{Cb} 及 resSamples_{Cr} 的所有样本设定成等于 0。

[0400] - 否则 (rqt_root_cbf 等于 1), 适用以下有序的步骤:

[0401] 1....

[0402] 2....

[0403] 3. 当 ChromaArrayType 不等于 0 时, 以亮度位置 (xCb, yCb)、设定成等于 (0, 0) 的亮

度位置 (xB0, yB0)、设定成等于log2CbSize的变量log2TrafoSize、设定成等于0的变量trafoDepth、设定成等于2的变量cIdx、设定成等于nCbSwC的变量 nCbSw、设定成等于nCbShC的变量nCbSh,及 (nCbSwC) × (nCbShC) 阵列resSamples_{Cr}作为输入而调用如下文子条款8.5.4.3中指定的色度残余块的解码过程,且输出为 (nCbSwC) × (nCbShC) 阵列resSamples_{Cr}的修改后的版本。

4. 当 color_transform_flag[xCb][yCb]等于 1 时, 以设定成等于 nCbS_L 的变量 blkSize、设定成等于 resSamples_L 的(nCbS_L)×(nCbS_L)阵列 r_Y、设定成等于 resSamples_{Cb} 的 (nCbS_L)×(nCbS_L) 阵列 r_{Cb}、及设定成等于 resSamples_{Cr} 的 (nCbS_L)×(nCbS_L)阵列 r_{Cr}作为输入而调用如子条款 8.6.7 中指定的使用颜色空间转换的残余块的残余修改过程, 且修改后的阵列 resSamples_L、resSamples_{Cb} 及 resSamples_{Cr}作为输出。

[0405] 8.6.1量化参数的导出过程

[0406] 对此过程的输入是指定当前亮度译码块的左上方样本相对于当前图片的左上方亮度样本的亮度位置 (xCb, yCb)。

[0407] 在此过程中,导出变量Q_{PY}、亮度量化参数Q_{p'}_Y及色度量化参数Q_{p'}_{Cb}及Q_{p'}_{Cr}。

[0408] 亮度位置 (xQg, yQg) 指定当前量化分组的左上方亮度样本相对于当前图片的左上方亮度样本。水平及垂直位置 xQg 及 yQg 分别设定成等于 xCb - (xCb & ((1 < Log2MinCuQpDeltaSize) - 1)) 及 yCb - (yCb & ((1 < Log2MinCuQpDeltaSize) - 1))。量化分组的亮度大小Log2MinCuQpDeltaSize确定共享相同q_{PY_PRED}的译码树块内的最小区域的亮度大小。

[0409] 经预测亮度量化参数q_{PY_PRED}由以下有序的步骤导出:

[0410] ...

[0411] 变量Q_{PY}导出如下:

[0412]
$$Q_{PY} = ((q_{PY_PRED} + CuQpDeltaVal + 52 + 2 * QpBdOffset_Y) \% (52 + QpBdOffset_Y)) - QpBdOffset_Y \quad (8-261)$$

[0413]
$$Q_{PY} = Q_{PY} + (color_transform_flag[xCb][yCb] ? deltaQP_{C0} : 0)$$

[0414] 亮度量化参数Q_{p'}_Y导出如下:

[0415]
$$Q_{p'}_Y = Q_{PY} + QpBdOffset_Y \quad (8-262)$$

[0416] 当ChromaArrayType不等于0时,适用以下。

[0417] -变量q_{PiCb}及q_{PiCr}导出如下:

[0418]
$$q_{PiCb} = Clip3(-QpBdOffset_C, 57, Q_{PY} + pps_cb_qp_offset + slice_cb_qp_offset + CuQpOffset_{Cb}) \quad (8-263)$$

[0419]
$$q_{PiCr} = Clip3(-QpBdOffset_C, 57, Q_{PY} + pps_cr_qp_offset + slice_cr_qp_offset + CuQpOffset_{Cr}) \quad (8-264)$$

[0420] -如果ChromaArrayType等于1,那么基于索引q_{Pi}等于q_{PiCb}及q_{PiCr}而分别将变量q_{Pcb}及q_{Pcr}设定成等于如表8-10中指定的Q_{pc}的值。

[0421] -否则,基于索引q_{Pi}等于q_{PiCb}及q_{PiCr}而分别将变量q_{Pcb}及q_{Pcr}设定成等于 Min(q_{Pi}, 51)。

[0422] -用于Cb及Cr分量的色度量化参数 Qp'_{Cb} 及 Qp'_{Cr} 导出如下:

[0423] $Qp'_{Cb} = qP_{Cb} + QpBdOffset_c$ (8-265)

[0424] $Qp'_{Cr} = qP_{Cr} + QpBdOffset_c$ (8-266)

[0425] 表8-10-对于ChromaArrayType等于1,取决于qPi的 Qp_c 的规范

[0426]	qPi	<30	30	31	32	33	34	35	36	37	38	39	40	41	42	43	>43
	Qp_c	=qPi	29	30	31	32	33	33	34	34	35	35	36	36	37	37	=qPi-6

$Qp'_{Cb} = Qp'_{Cb} + (color_transform_flag[xCb][yCb] ? deltaQP_{C1} : 0)$

[0427] $Qp'_{Cr} = Qp'_{Cr} + (color_transform_flag[xCb][yCb] ? deltaQP_{C2} : 0)$

[0428] 在一个实例中,分别将 $deltaQP_{C0}$ 、 $deltaQP_{C1}$ 及 $deltaQP_{C2}$ 设定成-5、-5及-3。在另一实例中,分别将 $deltaQP_{C0}$ 、 $deltaQP_{C1}$ 及 $deltaQP_{C2}$ 设定成-5、-5及-5。

[0429] 在解量化过程中,用于每一分量索引(cIdx)的量化参数qP导出如下:

[0430] 8.6.2缩放及变换过程

[0431] 对此过程的输入是:

[0432] -指定当前亮度变换块的左上方样本相对于当前图片的左上方亮度样本的亮度位置(xTbY,yTbY),

[0433] -指定当前块相对于译码块的层次深度的变量trafoDepth,

[0434] -指定当前块的颜色分量的变量cIdx,

[0435] -指定当前变换块的大小的变量nTbS。

[0436] 此过程的输出为具有元素 $r[x][y]$ 的残余样本r的 $(nTbS) \times (nTbS)$ 阵列。

[0437] 量化参数qP导出如下:

[0438] -如果cIdx等于0,则

[0439] $qP = Qp'_Y[[] + (color_transform_flag[xTbY][yTbY] ? -5 : 0)]$ (8-267)

[0440] -否则,如果cIdx等于1,则

[0441] $qP = Qp'_{Cb}[[] + (color_transform_flag[xTbY][yTbY] ? -5 : 0)]$ (8-268)

[0442] -否则(cIdx等于2),

$$qP = Qp'_{Cr}[] + (\text{color_transform_flag} [xTbY][yTbY] ? -3 : 0)[] \quad (8-269)$$

8.6.7 使用颜色空间转换的变换块的残余修改过程

仅当 ChromaArrayType 等于 3 时调用此过程。

对此过程的输入是：

- 指定块大小的变量 blkSize,
- 具有元素 $r_Y[x][y]$ 的亮度残余样本 r_Y 的 (blkSize)×(blkSize) 阵列,
- 具有元素 $r_{Cb}[x][y]$ 的色度残余样本 r_{Cb} 的 (blkSize)×(blkSize) 阵列,
- 具有元素 $r_{Cr}[x][y]$ 的色度残余样本 r_{Cr} 的 (blkSize)×(blkSize) 阵列。

此过程的输出是：

- 亮度残余样本的修改后的 (blkSize)×(blkSize) 阵列 r_Y ,
- 色度残余样本的修改后的 (blkSize)×(blkSize) 阵列 r_{Cb} ,
- 色度残余样本的修改后的 (blkSize)×(blkSize) 阵列 r_{Cr} 。

[0443]

残余样本 r_Y 、 r_{Cb} 及 r_{Cr} 的 (blkSize)×(blkSize) 阵列修改如下：

- 如果 cu_transquant_bypass_flag 等于 1, 那么残余样本 r_Y 、 r_{Cb} 及 r_{Cr} 的 (blkSize)×(blkSize) 阵列(其中 $x = 0..blkSize - 1$ 、 $y = 0..blkSize - 1$)修改如下：

```
tmp = rY[x][y] - ( rCb[x][y] >> 1 )
rY[x][y] = tmp + rCb[x][y]
rCb[x][y] = tmp - ( rCr[x][y] >> 1 )
rCr[x][y] = rCb[x][y] + rCr[x][y]
```

- 否则 (cu_transquant_bypass_flag 等于 0), 残余样本 r_Y 、 r_{Cb} 及 r_{Cr} 的 (blkSize)×(blkSize) 阵列(其中 $x = 0..blkSize - 1$ 、 $y = 0..blkSize - 1$)修改如下：

```
tmp = rY[x][y] - rCb[x][y]
rY[x][y] = rY[x][y] + rCb[x][y]
rCb[x][y] = tmp - rCr[x][y]
rCr[x][y] = tmp + rCr[x][y]
```

[0444]

表9-4-用于初始化过程中的每一initializationType的ctxIdx与语法元素的关联

语法结构	语法元素	ctxTable	initType		
			0	1	2
coding_unit()	cu_transquant_bypass_flag	表 9 8	0	1	2
	cu_skip_flag	表 9-9		0..2	3..5
	intra_bc_flag[][]	表 9-33	0	1	2
	pred_mode_flag	表 9-10		0	1

[0445]

[0446]	part_mode	表 9-11	0 9..11	1..4	5..8
	prev_intra_luma_pred_flag[][]	表 9-12	0	1	2
	intra_chroma_pred_mode[][]	表 9-13	0	1	2
	rqt_root_cbf	表 9-14		0	1
	color_transform_flag	表 9-XX	0	1	2

[0447] 表 9-XX-color_transform_flag 的 ctxIdx 的 initValue 值

[0448]	初始化变量	color_transform_flag 的 ctxIdx		
		0	1	2
	initValue	154	154	154

[0449] 表9-34-语法元素及相关联的二进制化

语法结构	语法元素	二进制化	
		过程	输入参数
[0450] coding_unit()	cu_transquant_bypass_flag	FL	cMax = 1
	cu_skip_flag	FL	cMax = 1
	intra_bc_flag	FL	cMax = 1
	pred_mode_flag	FL	cMax = 1
	part_mode	9.3.3.5	(xCb, yCb) = (x0, y0), log2CbSize
	pcm_flag[][]	FL	cMax = 1
	prev_intra_luma_pred_flag[][]	FL	cMax = 1
	mpm_idx[][]	TR	cMax = 2, cRiceParam = 0
	rem_intra_luma_pred_mode[][]	FL	cMax = 31
	intra_chroma_pred_mode[][]	9.3.3.6	-
	rqt_root_cbf	FL	cMax = 1
	color_transform_flag	FL	cMax = 1

[0451] 现将描述第二实例,下文称为实例#2。如上文所指示,在此实例#2中新添加的部分以粗体或斜体(即,粗体或斜体)突出显示,并且删除的部分标记为方括号中的粗体,例如,[[删除的文字]]。斜体部分与本发明中提出的技术有关。粗体带下划线的(即,“粗体带下划线的”)部分突出实例#2与实例#1之间的区别。

[0452] 此实例#2给出其中defaultVal等于1的情况的实例。在一个实例中,可通过以上章节中对于实例#1所定义的相同的方式进行处理。

[0453] 替代地,可应用以下以避免在当不存在color_transform_flag但是 color_transform_flag的值重设为1时对一个块进行帧内译码的情况下错误使用颜色变换:

[0454] 8.4.1以帧内预测模式译码的译码单元的一般解码过程

[0455] 对此过程的输入是:

[0456] -...

[0457] 此过程的输出是解决滤波之前的修改后的经重构图片。

[0458] 以亮度位置(xCb,yCb)作为输入而调用如子条款8.6.1中指定的量化参数的导出过程。

[0459] 变量nCbs设定成等于 $1 < \log_2 \text{CbSize}$ 。

[0460] 取决于pcm_flag[xCb][yCb]及IntraSplitFlag的值,亮度样本的解码过程指定如下:

[0461] -如果pcm_flag[xCb][yCb]等于1,那么经重构图片修改如下:

[0462] ...

[0463] -否则 (pcm_flag[xCb][yCb]等于0),如果IntraSplitFlag等于0,那么适用以下有序的步骤:

[0464] 1. 设定变量 bModified 等于(intra_chroma_pred_mode[x0][y0] == 4 && color_transform_flag[xCb][yCb])。

[0465] 2.当intra_bc_flag[xCb][yCb]等于0时,以亮度位置(xCb,yCb)作为输入而调用如子条款8.4.2中指定的帧内预测模式的导出过程。

[0466] 3.当intra_bc_flag[xCb][yCb]等于1时,以亮度位置(xCb,yCb)及变量log2CbSize作为输入而调用如子条款8.4.4中指定的帧内块复制预测模式中的块向量分量的导出过程,且输出为bvIntra。

4. 如果[[color_transform_flag[xCb][yCb]]] bModified 等于 1, 那么适用以下:

- 对于以值 0..2 继续进行的变量 cIdx, 适用以下有序的步骤:

- 设定变量 comp 等于(!cIdx ? L : (cIdx == 1 ? Cb : Cr))。

[0467] - 以位置 (xCb, yCb)、设定成等于 log2CbSize 的变量 log2TrafoSize、设定成等于 0 的变量 trafoDepth、设定成等于 IntraPredModeY[xCb][yCb]的变量 predModeIntra、设定成等于 intra_bc_flag[xCb][yCb]的变量 predModeIntraBc、变量 bvIntra、变量 cIdx 及等于 1 的变量 controlPara 作为输入而调用如子条款 8.4.4.1 中指定的帧内块的一般解码过程, 且输出为残余样本阵列 resSamples_{comp}。

- 以设定成等于 nCbS 的变量 blkSize、设定成等于 resSamples_L 的 (nCbS)×(nCbS)阵列 r_Y、设定成等于 resSamples_{Cb} 的 (nCbS)×(nCbS)阵列 r_{Cb} 及设定成等于 resSamples_{Cr} 的 (nCbS)×(nCbS)阵列 r_{Cr} 作为输入而调用如子条款 8.6.7 中指定的使用颜色空间转换的残余块的残余修改过程, 且输出为 (nCbS)×(nCbS)阵列 resSamples_L、resSamples_{Cb} 及 resSamples_{Cr} 的修改后的型式。

[0468] 5.以亮度位置(xCb,yCb)、设定成等于log2CbSize的变量log2TrafoSize、设定成等于0的变量trafoDepth、设定成等于IntraPredModeY[xCb][yCb]的变量 predModeIntra、设定成等于intra_bc_flag[xCb][yCb]的变量predModeIntraBc、变量bvIntra、[[及]]设定成等于0的变量cIdx、及 等 于

([[color_transform_flag[xCb][yCb]]] bModified ? 2 : 3)的变量 controlPara 作为输入而调用如子条款8.4.4.1中指定的帧内块的一般解码过程,且输出为解决滤波之前的修改后的经重构图片。

[0469] -否则 (pcm_flag[xCb][yCb]等于0且IntraSplitFlag等于1),对于以值0..3继续

进行的变量blkIdx,适用以下有序的步骤:

[0470] 6. 变量xPb设定成等于 $x_{Cb} + (n_{CbS} \gg 1) * (blkIdx \% 2)$ 。

[0471] 7. 变量yPb设定成等于 $y_{Cb} + (n_{CbS} \gg 1) * (blkIdx / 2)$ 。

[0472] 8. 以亮度位置 (xPb, yPb) 作为输入而调用如子条款8.4.2中指定的帧内预测模式的导出过程。

[0473] 9. 如果 $[[color_transform_flag[xCb][yCb]]]$ **bModified** 等于1, 那么适用以下:

- 对于以值 0..2 继续进行的变量 cIdx, 适用以下有序的步骤:

- 设定变量 comp 等于 $(!cIdx ? L : (cIdx = 1 ? Cb : Cr))$ 。

- 以亮度位置位置 (xPb, yPb)、设定成等于 $\log_2 CbSize - 1$ 的变量 $\log_2 TrafoSize$ 、设定成等于 1 的变量 $trafoDepth$ 、设定成等于 $IntraPredModeY[xPb][yPb]$ 的变量 $predModeIntra$ 、设定成等于 0 的变量 $predModeIntraBc$ 、变量 cIdx 及设定成等于 1 的变量 $controlPara$ 作为输入而调用如子条款 8.4.4.1 中指定的帧内块的一般解码过程, 且输出为残余样本阵列 $resSamples_{comp}$ 。

[0474]

- 设定变量 $nSubCbS$ 等于 $(n_{CbS} \gg 1)$, 且以设定成等于 $nSubCbS$ 的变量 $blkSize$ 、设定成等于 $resSamples_L$ 的 $(nSubCbS) \times (nSubCbS)$ 阵列 r_Y 、设定成等于 $resSamples_{Cb}$ 的 $(nSubCbS) \times (nSubCbS)$ 阵列 r_{Cb} 、及设定成等于 $resSamples_{Cr}$ 的 $(nSubCbS) \times (nSubCbS)$ 阵列 r_{Cr} 作为输入而调用如子条款 8.6.7 中指定的使用颜色空间转换的残余块的残余修改过程, 且输出为 $(nSubCbS) \times (nSubCbS)$ 阵列 $resSamples_L$ 、 $resSamples_{Cb}$ 及 $resSamples_{Cr}$ 的修改后的型式。

修改后的型式。

[0475] 10. 以亮度位置 (xPb, yPb)、设定成等于 $\log_2 CbSize - 1$ 的变量 $\log_2 TrafoSize$ 、设定成等于 1 的变量 $trafoDepth$ 、设定成等于 $IntraPredModeY[xPb][yPb]$ 的变量 $predModeIntra$ 、设定成等于 0 的变量 $predModeIntraBc$ 、**[[及]]** 设定成等于 0 的变量 cIdx 及设定成等于 $([[color_transform_flag[xCb][yCb]]] \text{ bModified} ? 2 : 3)$ 的变量

controlPara 作为输入而调用如子条款8.4.4.1中指定的帧内块的一般解码过程, 且输出为解块滤波之前的修改后的经重构图片。

[0476] 当ChromaArrayType不等于0时, 适用以下。

[0477] 变量 $\log_2 CbSizeC$ 设定成等于 $\log_2 CbSize - (ChromaArrayType == 3 ? 0 : 1)$ 。

[0478] 取决于 $pcm_flag[xCb][yCb]$ 及 $IntraSplitFlag$ 的值, 色度样本的解码过程指定如下:

[0479] -如果 $pcm_flag[xCb][yCb]$ 等于 1, 那么经重构图片修改如下:

[0480] ...

[0481] - 否则 ($\text{pcm_flag}[\text{xCb}][\text{yCb}]$ 等于 0), 如果 IntraSplitFlag 等于 0 或 ChromaArrayType 不等于 3, 那么适用以下有序的步骤:

[0482] 4. 当 $\text{intra_bc_flag}[\text{xCb}][\text{yCb}]$ 等于 0 时, 以亮度位置 (xCb, yCb) 作为输入而调用如 8.4.3 中指定的色度帧内预测模式的导出过程, 且输出为变量 IntraPredModeC 。

[0483] 5. 以色度位置 ($\text{xCb}/\text{SubWidthC}, \text{yCb}/\text{SubHeightC}$)、设定成等于 $\log2\text{CbSizeC}$ 的变量 $\log2\text{TrafoSize}$ 、设定成等于 0 的变量 trafoDepth 、设定成等于 IntraPredModeC 的变量 predModeIntra 、设定成等于 $\text{intra_bc_flag}[\text{xCb}][\text{yCb}]$ 的变量 predModeIntraBc 、变量 bvIntra 、及设定成等于 1 的变量 cIdx 、**及设定成等于**

($[[\text{color_transform_flag}[\text{xCb}][\text{yCb}]] \text{ bModified} ? 2 : 3]$) 的变量 controlPara 作为输入而调用如子条款 8.4.4.1 中指定的帧内块的一般解码过程, 且输出为解决滤波之前的修改后的经重构图片。

[0484] 6. 以色度位置 ($\text{xCb}/\text{SubWidthC}, \text{yCb}/\text{SubHeightC}$)、设定成等于 $\log2\text{CbSizeC}$ 的变量 $\log2\text{TrafoSize}$ 、设定成等于 0 的变量 trafoDepth 、设定成等于 IntraPredModeC 的变量 predModeIntra 、设定成等于 $\text{intra_bc_flag}[\text{xCb}][\text{yCb}]$ 的变量 predModeIntraBc 、变量 bvIntra 、**及设定成等于 2 的变量 cIdx 、及设定成等于**

($[[\text{color_transform_flag}[\text{xCb}][\text{yCb}]] \text{ bModified} ? 2 : 3]$) 的变量 controlPara 作为输入而调用如子条款 8.4.4.1 中指定的帧内块的一般解码过程, 且输出为解决滤波之前的修改后的经重构图片。

[0485] - 否则 ($\text{pcm_flag}[\text{xCb}][\text{yCb}]$ 等于 0、 IntraSplitFlag 等于 1 且 ChromaArrayType 等于 3), 对于以值 0..3 继续进行的变量 blkIdx , 适用以下有序的步骤:

[0486] 6. 变量 xPb 设定成等于 $\text{xCb} + (\text{nCbS} >> 1) * (\text{blkIdx} \% 2)$ 。

[0487] 7. 变量 yPb 设定成等于 $\text{yCb} + (\text{nCbS} >> 1) * (\text{blkIdx} / 2)$ 。

[0488] 8. 以亮度位置 (xPb, yPb) 作为输入而调用如 8.4.3 中指定的色度帧内预测模式的导出过程, 且输出为变量 IntraPredModeC 。

[0489] 9. 以色度位置 (xPb, yPb)、设定成等于 $\log2\text{CbSizeC}-1$ 的变量 $\log2\text{TrafoSize}$ 、设定成等于 1 的变量 trafoDepth 、设定成等于 IntraPredModeC 的变量 predModeIntra 、设定成等于 0 的变量 predModeIntraBc 、**[[及]] 设定成等于 1 的变量 cIdx 、及设定成**

等于($[[\text{color_transform_flag}[\text{xCb}][\text{yCb}]] \text{ bModified} ? 2 : 3]$) 的变量 controlPara 作为输入而调用如子条款 8.4.4.1 中指定的帧内块的一般解码过程, 且输出为解决滤波之前的修改后的经重构图片。

[0490] 10. 以色度位置 (xPb, yPb)、设定成等于 $\log2\text{CbSizeC}-1$ 的变量 $\log2\text{TrafoSize}$ 、设定成等于 1 的变量 trafoDepth 、设定成等于 IntraPredModeC 的变量 predModeIntra 、设定成等于 0 的变量 predModeIntraBc 、**[[及]] 设定成等于 2 的变量 cIdx 、及设定成**

等于($[[\text{color_transform_flag}[\text{xCb}][\text{yCb}]] \text{ bModified} ? 2 : 3]$) 的变量 controlPara 作为输入而调用如子条款 8.4.4.1 中指定的帧内块的一般解码过程, 且输出为解决滤波之前的修改后的经重构图片。

[0491] 8.6.1 量化参数的导出过程

[0492] 对此过程的输入是指定当前亮度译码块的左上方样本相对于当前图片的左上方亮度样本的亮度位置 (xCb, yCb)。

[0493] 在此过程中, 导出变量 Q_{PY} 、亮度量化参数 $Q_{p'Y}$ 及色度量化参数 $Q_{p'Cb}$ 及 $Q_{p'Cr}$ 。

[0494] 亮度位置 (xQg, yQg) 指定当前量化群组的左上方亮度样本相对于当前图片的左上方亮度样本。水平及垂直位置 xQg 及 yQg 分别设定成等于 $xCb - (xCb \& ((1 < \log_2 \text{MinCuQpDeltaSize}) - 1))$ 及 $yCb - (yCb \& ((1 < \log_2 \text{MinCuQpDeltaSize}) - 1))$ 。量化群组的亮度大小 $\log_2 \text{MinCuQpDeltaSize}$ 确定共享相同 q_{PY_PRED} 的译码树块内的最小区域的亮度大小。

[0495] 经预测亮度量化参数 q_{PY_PRED} 由以下有序的步骤导出:

[0496] ...

[0497] **变量 bModified 定义为**(color transform flag[xCb][yCb] && (CuPredMode [xCb][yCb] == MODE_INTER || intra chroma pred mode[xCb][yCb] == 4))。

[0498] 变量 Q_{PY} 导出如下:

[0499] $Q_{PY} = ((q_{PY_PRED} + \text{CuQpDeltaVal} + 52 + 2 * Q_{pBdOffsetY}) \% (52 + Q_{pBdOffsetY})) - Q_{pBdOffsetY} (8-261)$

[0500] $Q_{pY} = Q_{PY} + ([\text{color_transform_flag}[xCb][yCb]] \text{ bModified} ? \text{deltaQP}_{C0} : 0)$

[0501] 亮度量化参数 $Q_{p'Y}$ 导出如下:

[0502] $Q_{p'Y} = Q_{pY} + Q_{pBdOffsetY} (8-262)$

[0503] 当ChromaArrayType不等于0时, 适用以下。

[0504] ...

[0505] -用于Cb及Cr分量的色度量化参数 $Q_{p'Cb}$ 及 $Q_{p'Cr}$ 导出如下:

[0506] $Q_{p'Cb} = q_{PCb} + Q_{pBdOffsetC} (8-265)$

[0507] $Q_{p'Cr} = q_{PCr} + Q_{pBdOffsetC} (8-266)$

[0508] $Q_{p'Cb} = Q_{p'Cb} + ([\text{color_transform_flag}[xCb][yCb]] \text{ bModified} ? \text{deltaQP}_{C1} : 0)$

$Q_{p'Cr} = Q_{p'Cr} + ([\text{color_transform_flag}[xCb][yCb]] \text{ bModified} ? \text{deltaQP}_{C2} : 0)$

[0509] 替代地, 可适用以下:

[0510] **变量 bModified 定义为**(color transform flag[xCb][yCb] && (CuPredMode [xCb][yCb] != MODE_INTRA || intra chroma pred mode[xCb][yCb] == 4))。

[0511] 替代地, 可适用以下:

[0512] **变量 bModified 定义为**(color transform flag[xCb][yCb] && (CuPredMode [xCb][yCb] != MODE_INTRA || intra bc flag[xCb][yCb] || intra chroma pred mode[xCb][yCb] == 4))。

[0513] 现将描述第三实例, 下文称为实例#3。此实例#3与如上文所描述的实例#2之间的差别以粗体(即, “粗体”)突出显示。在此实例3中, $Q_{p'Y}$ 、 $Q_{p'Cb}$ 及 $Q_{p'Cr}$ 保持无变化。然而, 解量化及解块滤波过程将检查颜色变换的使用情况并修改QP。

[0514] 8.6.1 量化参数的导出过程

[0515] 对此过程的输入是指定当前亮度译码块的左上方样本相对于当前图片的左上方

亮度样本的亮度位置(xCb,yCb)。

[0516] 在此过程中,导出变量 Q_{PY} 、亮度量化参数 $Q_{p'Y}$ 及色度量化参数 $Q_{p'Cb}$ 及 $Q_{p'Cr}$ 。

[0517] 亮度位置(xQg,yQg)指定当前量化分组的左上方亮度样本相对于当前图片的左上方亮度样本。水平及垂直位置xQg及yQg分别设定成等于xCb-(xCb&((1<<Log2MinCuQpDeltaSize)-1))及yCb-(yCb&((1<<Log2MinCuQpDeltaSize)-1))。量化分组的亮度大小Log2MinCuQpDeltaSize确定共享相同 q_{PY_PRED} 的译码树块内的最小区域的亮度大小。

[0518] 经预测亮度量化参数 q_{PY_PRED} 由以下有序的步骤导出:

[0519] ...

[0520] 变量 Q_{PY} 导出如下:

[0521] $Q_{PY} = ((q_{PY_PRED} + CuQpDeltaVal + 52 + 2 * QpBdOffsetY) \% (52 + QpBdOffsetY)) - QpBdOffsetY$ (8-261)

[0522] **$[[Q_{PY} = Q_{PY} + (color_transform_flag[xCb][yCb] \&\& bModified ? deltaQP_{C0} : 0)]]$**

[0523] 亮度量化参数 $Q_{p'Y}$ 导出如下:

[0524] $Q_{p'Y} = Q_{PY} + QpBdOffsetY$ (8-262)

[0525] 当ChromaArrayType不等于0时,适用以下。

[0526] -....

[0527] -用于Cb及Cr分量的色度量化参数 $Q_{p'Cb}$ 及 $Q_{p'Cr}$ 导出如下:

[0528] $Q_{p'Cb} = q_{Pcb} + QpBdOffsetC$ (8-265)

[0529] $Q_{p'Cr} = q_{PCr} + QpBdOffsetC$ (8-266)

$[[Q_{p'Cb} = Q_{p'Cb} + (color_transform_flag[xCb][yCb] \&\& bModified ? deltaQPC1 : 0)]]$

[0530] **$[[Q_{p'Cr} = Q_{p'Cr} + (color_transform_flag[xCb][yCb] \&\& bModified ? deltaQPC2 : 0)]]$**

[0531] 替代地,可适用以下:

[0532] **$[[变量\ bModified\ 定义为(color_transform_flag[xCb][yCb] \&\& (CuPredMode[xCb][yCb] \neq MODE_INTRA \parallel intra_chroma_pred_mode[xCb][yCb] == 4)).]]$**

[0533] 替代地,可适用以下:

$[[变量\ bModified\ 定义为(color_transform_flag[xCb][yCb] \&\& (CuPredMode[xCb][yCb] \neq MODE_INTRA \parallel intra_bc_flag[xCb][yCb] \parallel intra_chroma_pred_mode[xCb][yCb] == 4)).]]$

[0535] 8.6.2缩放及变换过程

[0536] 对此过程的输入是:

[0537] -指定当前亮度变换块的左上方样本相对于当前图片的左上方亮度样本的亮度位置(xTbY,yTbY),

[0538] -指定当前块相对于译码块的层次深度的变量trafoDepth,

[0539] -指定当前块的颜色分量的变量cIdx,

- [0540] -指定当前变换块的大小的变量nTbS。
- [0541] 此过程的输出为具有元素 $r[x][y]$ 的残余样本 r 的 $(nTbS) \times (nTbS)$ 阵列。
- [0542] **将变量 xCb 及 yCb 设定为在覆盖当前亮度变换块的译码单元的左上方位置。**
- [0543] 量化参数 qP 导出如下：
- [0544] -如果cIdx等于0,则
- [0545] $qP = Qp'_Y + (\text{color_transform_flag}[xCb][yCb] ? \text{deltaQP}_{C0} : 0)$ (8-267)
- [0546] -否则,如果cIdx等于1,则
- [0547] $qP = Qp'_{Cb} + (\text{color_transform_flag}[xCb][yCb] ? \text{deltaQP}_{C1} : 0)$ (8-268)
- [0548] -否则(cIdx等于2),
- [0549] $qP = Qp'_c + (\text{color_transform_flag}[xCb][yCb] ? \text{deltaQP}_{C2} : 0)$ (8-269)
- [0550] 8.7.2.5.3亮度块边缘的决策过程
- [0551] 对此过程的输入是：
- [0552] -亮度图片样本阵列recPictureL,
- [0553] -对此过程的输入是指定当前亮度译码块的左上方样本相对于当前图片的左上方亮度样本的亮度位置(xCb,yCb),
- [0554] -指定当前亮度块的左上方样本相对于当前亮度译码块的左上方样本的亮度位置(xBl,yBl),
- [0555] -指定垂直(EDGE_VER)或水平(EDGE_HOR)边缘是否经滤波的变量edgeType,
- [0556] -指定边界滤波强度的变量bS。
- [0557] 此过程的输出是：
- [0558] -含有决策的变量dE、dEp及dEq,
- [0559] -变量 β 及 t_c 。
- [0560] 如果edgeType等于EDGE_VER,那么样本值 $p_{i,k}$ 及 $q_{i,k}$ (其中 $i=0..3$ 且 $k=0$ 及3)导出如下：
- [0561] $q_{i,k} = \text{recPictureL}[xCb+xBl+i][yCb+yBl+k]$ (8-300)
- [0562] $p_{i,k} = \text{recPictureL}[xCb+xBl-i-1][yCb+yBl+k]$ (8-301)
- [0563] 否则(edgeType等于EDGE_HOR),样本值 $p_{i,k}$ 及 $q_{i,k}$ (其中 $i=0..3$ 且 $k=0$ 及3)导出如下：
- [0564] $q_{i,k} = \text{recPictureL}[xCb+xBl+k][yCb+yBl+i]$ (8-302)
- [0565] $p_{i,k} = \text{recPictureL}[xCb+xBl+k][yCb+yBl-i-1]$ (8-303)
- [0566] 变量 Qp_Q 及 Qp_P 分别设定成等于包含含有样本 $q_{0,0}$ 及 $p_{0,0}$ 的译码块的译码单元的 Qp_Y 值。
- [0567] **当含有样本 $q_{0,0}$ 的译码单元的 color_transform_flag[xCb][yCb]等于 1 时, Qp_Q 重设成等于($Qp_Q + \text{deltaQP}_{C0}$)。**
- 当含有样本 $p_{0,0}$ 的译码单元的 color_transform_flag[xCb][yCb]等于 1 时, Qp_P 重设成等于($Qp_P + \text{deltaQP}_{C0}$)。**
- [0568] 变量 qP_L 导出如下：

- [0569] $q_{PL} = ((Q_{pQ} + Q_{pP} + 1) \gg 1)$ (8-304)
- [0570] 8.7.2.5.5色度块边缘的滤波过程
- [0571] 仅当ChromaArrayType不等于0时调用此过程。
- [0572] 对此过程的输入是：
- [0573] -色度图片样本阵列 s' ，
- [0574] -指定当前色度译码块的左上方样本相对于当前图片的左上方色度样本的色度位置 (x_{Cb}, y_{Cb}) ，
- [0575] -指定当前色度块的左上方样本相对于当前色度译码块的左上方样本的色度位置 (x_{Bl}, y_{Bl}) ，
- [0576] -指定垂直(EDGE_VER)或水平(EDGE_HOR)边缘是否经滤波的变量edgeType，
- [0577] -指定图片级色度量化参数偏移的变量cQpPicOffset。
- [0578] 此过程的输出是修改后的色度图片样本阵列 s' 。
- [0579] 如果edgeType等于EDGE_VER,那么值 p_i 及 q_i (其中 $i=0..1$ 且 $k=0..3$)导出如下：
- [0580] $q_{i,k} = s'[x_{Cb} + x_{Bl} + i][y_{Cb} + y_{Bl} + k]$ (8-335)
- [0581] $p_{i,k} = s'[x_{Cb} + x_{Bl} - i - 1][y_{Cb} + y_{Bl} + k]$ (8-336)
- [0582] 否则(edgeType等于EDGE_HOR),样本值 p_i 及 q_i (其中 $i=0..1$ 且 $k=0..3$)导出如下：
- [0583] $q_{i,k} = s'[x_{Cb} + x_{Bl} + k][y_{Cb} + y_{Bl} + i]$ (8-337)
- [0584] $p_{i,k} = s'[x_{Cb} + x_{Bl} + k][y_{Cb} + y_{Bl} - i - 1]$ (8-338)
- [0585] 变量 Q_{pQ} 及 Q_{pP} 分别设定成等于包含含有样本 $q_{0,0}$ 及 $p_{0,0}$ 的译码块的译码单元的 Q_{pY} 值。

当含有样本 $q_{0,0}$ 的译码单元的 **color_transform_flag[xCb][yCb]** 等于 1 时, Q_{pQ} 重设成等于 $(Q_{pQ} + \text{deltaQP}_{C1})$ 。

[0586]

当含有样本 $p_{0,0}$ 的译码单元的 **color_transform_flag[xCb][yCb]** 等于 1 时, Q_{pP} 重设成等于 $(Q_{pP} + \text{deltaQP}_{C1})$ 。

[0587] 如果ChromaArrayType等于1,那么基于如下导出的索引 q_{Pi} 如表8-10中指定的确定变量 Q_{pC} :

[0588] $q_{Pi} = ((Q_{pQ} + Q_{pP} + 1) \gg 1) + cQpPicOffset$ (8-339)

[0589] 在一个实例中,分别将 deltaQP_{C0} 、 deltaQP_{C1} 及 deltaQP_{C2} 设定成-5、-5及-3。

[0590] 在另一实例中,分别将 deltaQP_{C0} 、 deltaQP_{C1} 及 deltaQP_{C2} 设定成-5、-5及-5。

[0591] 替代地,如以上实例#2的子条款8.6.1中所使用,color_transform_flag[xCb][yCb]可由bModified替代。

[0592] 现将描述第四实例,下文称为实例#4。在实例#1中,在章节8.6.1中修改 Q_{pY} 、 $Q_{p'Cb}$ 、 $Q_{p'Cr}$ 。在这种情况下,解块滤波过程及解量化过程中使用的QP保持相同。然而,QP预测符导出过程无变化(即,子条款8.6.1中的 q_{PY_PRED})。在此实例中,当针对当前切片启用颜色变换时修改QP预测符的导出过程。除了用于导出QP预测符的当前条件以外,还进一步包含相关相邻块/最后一个经译码块的颜色变换旗标。这对应于章节4的项目编号5。此外,导出的值限制为不小于0。相较于实例#1的变化以粗体突出显示。

[0593] 8.6.1量化参数的导出过程

[0594] 对此过程的输入是指定当前亮度译码块的左上方样本相对于当前图片的左上方亮度样本的亮度位置 (xCb, yCb)。

[0595] 在此过程中, 导出变量 q_{PY} 、亮度量化参数 $q_{p'Y}$ 及色度量化参数 $q_{p'Cb}$ 及 $q_{p'Cr}$ 。

[0596] 亮度位置 (xQg, yQg) 指定当前量化群组的左上方亮度样本相对于当前图片的左上方亮度样本。水平及垂直位置 xQg 及 yQg 分别设定成等于 $xCb - (xCb \& ((1 < \log_2 \text{MinCuQpDeltaSize}) - 1))$ 及 $yCb - (yCb \& ((1 < \log_2 \text{MinCuQpDeltaSize}) - 1))$ 。量化群组的亮度大小 $\log_2 \text{MinCuQpDeltaSize}$ 确定共享相同 q_{PY_PRED} 的译码树块内的最小区域的亮度大小。

[0597] 经预测亮度量化参数 q_{PY_PRED} 由以下有序的步骤导出:

[0598] 1. **设定变量 b_{Y_PREV} 等于 false, 并且变量 q_{PY_PREV} 导出如下:**

[0599] -如果以下条件中的一或多者成立, 那么 q_{PY_PREV} 设定为等于 SliceQP_Y :

[0600] -当前量化群组是切片中的第一量化群组。

[0601] -当前量化群组是瓦片中的第一量化群组。

[0602] -当前量化群组是译码树块行中的第一量化群组且 $\text{entropy_coding_sync_enabled_flag}$ 等于 1。

[0603] -否则, q_{PY_PREV} 及 b_{Y_PREV} 分别设定成等于按解码次序在先前量化群组中的最后一个译码单元的亮度量化参数 Q_{PY} 及 $\text{color_transform_flag}$ 。

[0604] 2. 以设定成等于 (xCb, yCb) 的位置 (xCurr, yCurr) 及设定成等于 (xQg-1, yQg) 的相邻位置 (xNbY, yNbY) 作为输入而调用如子条款 6.4.1 中指定的 z 扫描次序中的块的可用性导出过程, 且将输出指派给 availableA。变量 q_{PY_A} 及 b_{Y_A} 导出如下:

[0605] -如果以下条件中的一或多者成立, 那么 q_{PY_A} 设定成等于 q_{PY_PREV} 且 b_{Y_A} 设定成等于 b_{Y_PREV} :

[0606] -availableA 等于 FALSE。

[0607] -含有覆盖亮度位置 (xQg-1, yQg) 的亮度译码块的译码树块的译码树块地址 ctbAddrA 不等于 CtbAddrInTs , 其中 ctbAddrA 导出如下:

[0608] $x_{\text{Tmp}} = (xQg - 1) \gg \text{MinTbLog2SizeY}$

[0609] $y_{\text{Tmp}} = yQg \gg \text{MinTbLog2SizeY}$

[0610] $\text{minTbAddrA} = \text{MinTbAddrZs}[x_{\text{Tmp}}][y_{\text{Tmp}}]$

[0611] $\text{ctbAddrA} = \text{minTbAddrA} \gg (2 * (\text{CtbLog2SizeY} - \text{MinTbLog2SizeY}))$ (8-252)

[0612] -否则, q_{PY_A} 及 b_{Y_A} 分别设定成等于含有覆盖 (xQg-1, yQg) 的亮度译码块的译码单元的亮度量化参数 Q_{PY} 及 $\text{color_transform_flag}$ 。

[0613] 3. 以设定成等于 (xCb, yCb) 的位置 (xCurr, yCurr) 及设定成等于 (xQg, yQg-1) 的相邻位置 (xNbY, yNbY) 作为输入而调用如子条款 6.4.1 中指定的 z 扫描次序中的块的可用性导出过程, 且将输出指派给 availableB。变量 q_{PY_B} 及 b_{Y_B} 导出如下:

[0614] -如果以下条件中的一或多者成立, 那么 q_{PY_B} 设定成等于 q_{PY_PREV} 且 b_{Y_B} 设定成等于 b_{Y_PREV} :

[0615] -availableB等于FALSE。

[0616] -含有覆盖亮度位置(xQg,yQg-1)的亮度译码块的译码树块的译码树块地址ctbAddrB不等于CtbAddrInTs,其中ctbAddrB导出如下:

[0617] xTmp=xQg>>MinTbLog2SizeY

[0618] yTmp=(yQg-1)>>MinTbLog2SizeY

[0619] minTbAddrB=MinTbAddrZs[xTmp][yTmp]

[0620] ctbAddrB=minTbAddrB>>(2*(CtbLog2SizeY- MinTbLog2SizeY)) (8-253)

[0621] -否则,qPY_B及bY_B分别设定成等于含有覆盖(xQg,yQg-1)的亮度译码块的译码单元的亮度量化参数QPY及color_transform_flag。

[0622] 4.经预测亮度量化参数qPY_PRED导出如下:

[0623] 如果bY_A及bY_B相等,那么

[0624]
$$qPY_PRED = ((qPY_A + qPY_B + 1) >> 1) + (\text{color_transform_flag}[xCb][yCb] == bY_A ? 0 : (bY_A ? -\text{deltaQP}_{c0} : \text{deltaQP}_{c0}))$$
 (8-254)

如果bY_A及bY_B不同,那么

[0625]
$$qPY_PRED = ((qPY_A + qPY_B - \text{deltaQP}_{c0} + 1) >> 1) + (\text{color_transform_flag}[xCb][yCb] ? \text{deltaQP}_{c0} : 0)$$
 (8-254)

替代地,当bY_A及bY_B相等时,公式(8-254)可由以下公式替代:

$$qPY_PRED = ((qPY_A + qPY_B + 1) >> 1) + (\text{color_transform_flag}[xCb][yCb] == bY_A ? 0 : (\text{color_transform_flag}[xCb][yCb] ? \text{deltaQP}_{c0} : -\text{deltaQP}_{c0}))$$
 (8-254)

[0626] 变量QPY导出如下:

[0627]
$$QPY = ((qPY_PRED + CuQpDeltaVal + 52 + 2 * QpBdOffsetY) \% (52 + QpBdOffsetY)) - QpBdOffsetY$$
 (8-261)

[0628]
$$QPY = \max(-QpBdOffsetY, QPY + (\text{color_transform_flag}[xCb][yCb] ? \text{deltaQP}_{c0} : 0))$$

替代地,以上两个方程式可由一个方程式替代:

[0629]
$$QPY = ((qPY_PRED + CuQpDeltaVal + 52 + (\text{color_transform_flag}[xCb][yCb] ? \text{deltaQP}_{c0} : 0) + 2 * QpBdOffsetY) \% (52 + QpBdOffsetY)) - QpBdOffsetY$$
 (8-261)

[0630] 亮度量化参数Qp'Y导出如下:

[0631]
$$Qp'Y = QPY + QpBdOffsetY$$
 (8-262)

[0632] 当ChromaArrayType不等于0时,适用以下。

[0633] -变量qPicb及qPicr导出如下:

[0634]
$$qPicb = \text{Clip3}(-QpBdOffsetc, 57, QPY + \text{pps_cb_qp_offset} + \text{slice_cb_qp_offset} +$$

$CuQpOffset_{Cb}$) (8-263)

[0635] $qPi_{Cr} = Clip3(-QpBdOffset_C, 57, Qp_Y + pps_cr_qp_offset + slice_cr_qp_offset + CuQpOffset_{Cr})$ (8-264)

[0636] -如果ChromaArrayType等于1,那么基于索引qPi等于qPiCb及qPiCr而分别将变量qPcb及qPcr设定成等于如表8-10中指定的Qpc的值。

[0637] -否则,基于索引qPi等于qPiCb及qPiCr而分别将变量qPcb及qPcr设定成等于 $\min(qPi, 51)$ 。

[0638] -用于Cb及Cr分量的色度量化参数 Qp'_{Cb} 及 Qp'_{Cr} 导出如下:

[0639] $Qp'_{Cb} = qP_{Cb} + QpBdOffset_C$ (8-265)

[0640] $Qp'_{Cr} = qP_{Cr} + QpBdOffset_C$ (8-266)

[0641] 表8-10-对于ChromaArrayType等于1,取决于qPi的Qpc的规范

qPi	<30	30	31	32	33	34	35	36	37	38	39	40	41	42	43	>43
Qpc	=qPi	29	30	31	32	33	33	34	34	35	35	36	36	37	37	=qPi-6

[0643] $Qp'_{Cb} = \max(0, Qp'_{Cb} + (\text{color_transform_flag}[xCb][yCb] ? \text{deltaQP}_{C1} : 0))$

$Qp'_{Cr} = \max(0, Qp'_{Cr} + (\text{color_transform_flag}[xCb][yCb] ? \text{deltaQP}_{C2} : 0))$

[0644] 替代地,适用以下以避免用于色度QP导出最大值函数:

[0645] 当ChromaArrayType不等于0时,适用以下。

[0646] -变量qPiCb及qPiCr导出如下:

$[[qPi_{Cb} = Clip3(-QpBdOffset_C, 57, Qp_Y + pps_cb_qp_offset + slice_cb_qp_offset + CuQpOffset_{Cb})$ (8-263)

$qPi_{Cr} = Clip3(-QpBdOffset_C, 57, Qp_Y + pps_cr_qp_offset + slice_cr_qp_offset + CuQpOffset_{Cr})$ (8-264)]]

[0647] $qPi_{Cb} = Clip3(-QpBdOffset_C, 57, Qp_Y + pps_cb_qp_offset + slice_cb_qp_offset + (\text{color_transform_flag}[xCb][yCb] ? (\text{deltaQP}_{C1} - \text{deltaQP}_{C0}) : 0) + CuQpOffset_{Cb})$ (8-263)

$qPi_{Cr} = Clip3(-QpBdOffset_C, 57, Qp_Y + pps_cr_qp_offset + slice_cr_qp_offset + (\text{color_transform_flag}[xCb][yCb] ? \text{deltaQP}_{C2} - \text{deltaQP}_{C0} : 0) + CuQpOffset_{Cr})$ (8-264)

[0648] -如果ChromaArrayType等于1,那么基于索引qPi等于qPiCb及qPiCr而分别将变量qPcb及qPcr设定成等于如表8-10中指定的Qpc的值。

[0649] -否则,基于索引qPi等于qPiCb及qPiCr而分别将变量qPcb及qPcr设定成等于 $\min(qPi, 51)$ 。

[0650] -用于Cb及Cr分量的色度量化参数 Qp'_{Cb} 及 Qp'_{Cr} 导出如下:

[0651] $Qp'_{Cb} = qP_{Cb} + QpBdOffset_C$ (8-265)

[0652] $Qp'_{Cr} = qP_{Cr} + QpBdOffset_C$ (8-266)

[0653] 表8-10-对于ChromaArrayType等于1,取决于qPi的Qpc的规范

qPi	<30	30	31	32	33	34	35	36	37	38	39	40	41	42	43	>43
Qpc	=qPi	29	30	31	32	33	33	34	34	35	35	36	36	37	37	=qPi-6

[0655]
$$\begin{aligned} & \ll \text{Qp}'_{\text{Cb}} = \max(0, \text{Qp}'_{\text{Cb}} + (\text{color_transform_flag}[\text{xCb}][\text{yCb}] ? \text{deltaQP}_{\text{C1}} : 0)) \\ & \text{Qp}'_{\text{Cr}} = \max(0, \text{Qp}'_{\text{Cr}} + (\text{color_transform_flag}[\text{xCb}][\text{yCb}] ? \text{deltaQP}_{\text{C2}} : 0)) \end{aligned}$$

在一个实例中, 分别将 $\text{deltaQP}_{\text{C0}}$ 、 $\text{deltaQP}_{\text{C1}}$ 及 $\text{deltaQP}_{\text{C2}}$ 设定成-5、-5 及-3。在另一实例中, 分别将 $\text{deltaQP}_{\text{C0}}$ 、 $\text{deltaQP}_{\text{C1}}$ 及 $\text{deltaQP}_{\text{C2}}$ 设定成-5、-5 及-5。

可在规格中增加 $\text{qP}_{\text{Y_PRED}}$ 不应小于 0 的限制。

[0656] 在解量化过程中, 用于每一分量索引 (cIdx) 的量化参数 qP 导出如下:

[0657] 8.6.2 缩放及变换过程

[0658] 对此过程的输入是:

[0659] -指定当前亮度变换块的左上方样本相对于当前图片的左上方亮度样本的亮度位置 (xTbY , yTbY),

[0660] -指定当前块相对于译码块的层次深度的变量 trafoDepth ,

[0661] -指定当前块的颜色分量的变量 cIdx ,

[0662] -指定当前变换块的大小的变量 nTbS 。

[0663] 此过程的输出为具有元素 $r[x][y]$ 的残余样本 r 的 $(\text{nTbS}) \times (\text{nTbS})$ 阵列。

[0664] 量化参数 qP 导出如下:

[0665] -如果 cIdx 等于 0, 则

[0666]
$$\text{qP} = \text{Qp}'_{\text{Y}} \ll + (\text{color_transform_flag}[\text{xTbY}][\text{yTbY}] ? -5 : 0) \quad (8-267)$$

[0667] -否则, 如果 cIdx 等于 1, 则

[0668]
$$\text{qP} = \text{Qp}'_{\text{Cb}} \ll + (\text{color_transform_flag}[\text{xTbY}][\text{yTbY}] ? -5 : 0) \quad (8-268)$$

[0669] -否则 (cIdx 等于 2),

[0670]
$$\text{qP} = \text{Qp}'_{\text{Cr}} \ll + (\text{color_transform_flag}[\text{xTbY}][\text{yTbY}] ? -3 : 0) \quad (8-269)$$

[0671] 图8是说明根据本发明的技术的实例视频解码方法的流程图。将关于视频编码器 20 描述图8的技术, 但是应理解, 图8的技术不限于任何特定类型的视频编码器。视频编码器 20 确定用于第一块的量化参数 (100)。响应于确定视频数据的第一块是使用用于第一块的残余数据的颜色空间变换模式进行译码的 (102, 是), 视频编码器 20 基于用于第一块的修改后的量化参数对第一块执行量化过程 (104)。响应于确定视频数据的第一块不是使用用于第一块的残余数据的颜色空间变换模式进行译码的 (102, 否), 视频编码器 20 基于用于第一块的未经修改的量化参数对第一块执行量化过程 (106)。针对视频数据的第二块, 视频编码器 20 用信号发送指示用于第二块的量化参数与用于第一块的未经修改的量化参数之间的差的差值 (108)。

[0672] 图9是说明根据本发明的技术的实例视频解码方法的流程图。将关于视频解码器 30 描述图9的技术, 但是应理解, 图9的技术不限于任何特定类型的视频解码器。针对视频数据的第一块, 视频解码器 30 接收用以确定用于所述第一块的量化参数的信息 (120)。响应于确定视频数据的第一块是使用用于第一块的残余数据的颜色空间变换模式进行译码的 (122, 是), 视频解码器 30 确定修改后的量化参数 (124) 并基于用于第一块的修改后的量化参数对第一块执行解量化过程 (126)。响应于确定视频数据的第一块不是使用用于第一块的残余数据的颜色空间变换模式进行译码的 (122, 否), 视频解码器 30 基于用于第一块的未

经修改的量化参数对第一块执行解量化过程(128)。针对视频数据的第二块,视频解码器30接收指示用于第二块的量化参数与用于第一块的未经修改的量化参数之间的差的差值(130)。

[0673] 应认识到,取决于实例,本文中所描述的技术中的任一者的某些动作或事件可用不同顺序来执行,可添加、合并或全部省略所述动作或事件(例如,实践所述技术未必需要所有所描述动作或事件)。此外,在某些实例中,可(例如)通过多线程处理、中断处理或多个处理器同时而非顺序地执行动作或事件。

[0674] 在一或多个实例中,所描述功能可以硬件、软件、固件或其任何组合来实施。如果以软件实施,那么所述功能可作为一或多个指令或代码在计算机可读媒体上存储或传输,并且由基于硬件的处理单元来执行。计算机可读媒体可包含计算机可读存储媒体,其对应于有形媒体,例如数据存储媒体,或包括任何促进将计算机程序从一处传送到另一处的任何媒体(例如,根据通信协议)的通信媒体。以此方式,计算机可读媒体一般可对应于(1)有形计算机可读存储媒体,其为非暂时性的,或(2)通信媒体,例如信号或载波。数据存储媒体可为可由一或多个计算机或一或多个处理器存取以检索用于实施本发明中描述的技术的指令、代码及/或数据结构的任何可用媒体。计算机程序产品可包含计算机可读媒体。

[0675] 借助于实例而非限制,此类计算机可读存储媒体可包括RAM、ROM、EEPROM、CD-ROM或其它光盘存储装置、磁盘存储装置或其它磁性存储装置、快闪存储器或可用来存储指令或数据结构的形式的期望程序代码并且可由计算机存取的任何其它媒体。并且,任何连接被恰当地称作计算机可读媒体。例如,如果使用同轴缆线、光纤缆线、双绞线、数字订户线(DSL)或例如红外线、无线电及微波等无线技术从网站、服务器或其它远程源传输指令,那么同轴缆线、光纤缆线、双绞线、DSL或例如红外线、无线电及微波等无线技术包含在媒体的定义中。然而,应理解,所述计算机可读存储媒体及数据存储媒体并不包含连接、载波、信号或其它暂时性媒体,而是实际上针对非暂时性的有形存储媒体。如本文中所使用,磁盘及光盘包含压缩光盘(CD)、激光光盘、光学光盘、数字多功能光盘(DVD)、软性磁盘及蓝光光盘,其中磁盘通常以磁性方式再现数据,而光盘利用激光以光学方式再现数据。以上各项的组合也应包含在计算机可读媒体的范围内。

[0676] 可由例如一或多个数字信号处理器(DSP)、通用微处理器、专用集成电路(ASIC)、现场可编程逻辑阵列(FPGA)或其它等效集成或离散逻辑电路等一或多个处理器来执行指令。因此,如本文中所使用的术语“处理器”可指前述结构或适合于实施本文中所描述的技术的任一其它结构中的任一者。另外,在一些方面中,本文中所描述的功能性可在经配置用于编码及解码的专用硬件及/或软件模块内提供,或者并入在组合编解码器中。而且,所述技术可完全实施于一或多个电路或逻辑元件中。

[0677] 本发明的技术可在各种各样的装置或设备中实施,包含无线手持机、集成电路(IC)或一组IC(例如,芯片组)。本发明中描述各种组件、模块或单元是为了强调经配置以执行所公开的技术的装置的功能方面,但未必需要由不同硬件单元实现。实际上,如上文所描述,各种单元可结合合适的软件及/或固件组合在编解码器硬件单元中,或者通过互操作硬件单元的集合来提供,所述硬件单元包含如上文所描述的一或多个处理器。

[0678] 已描述本发明的各种实例。涵盖所描述的系统、操作或功能的任何组合。这些及其它实例在所附权利要求书的范围内。

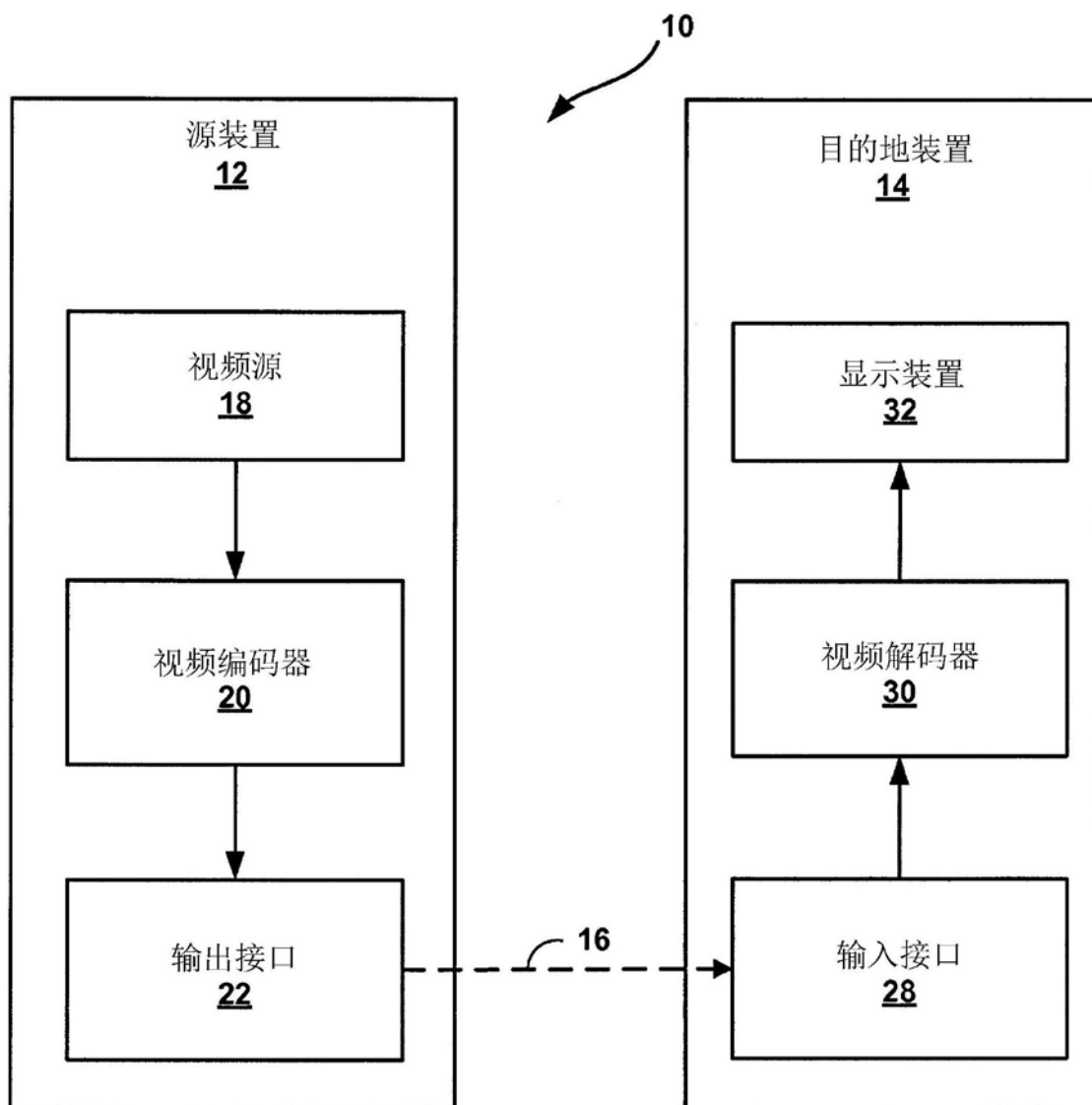


图1

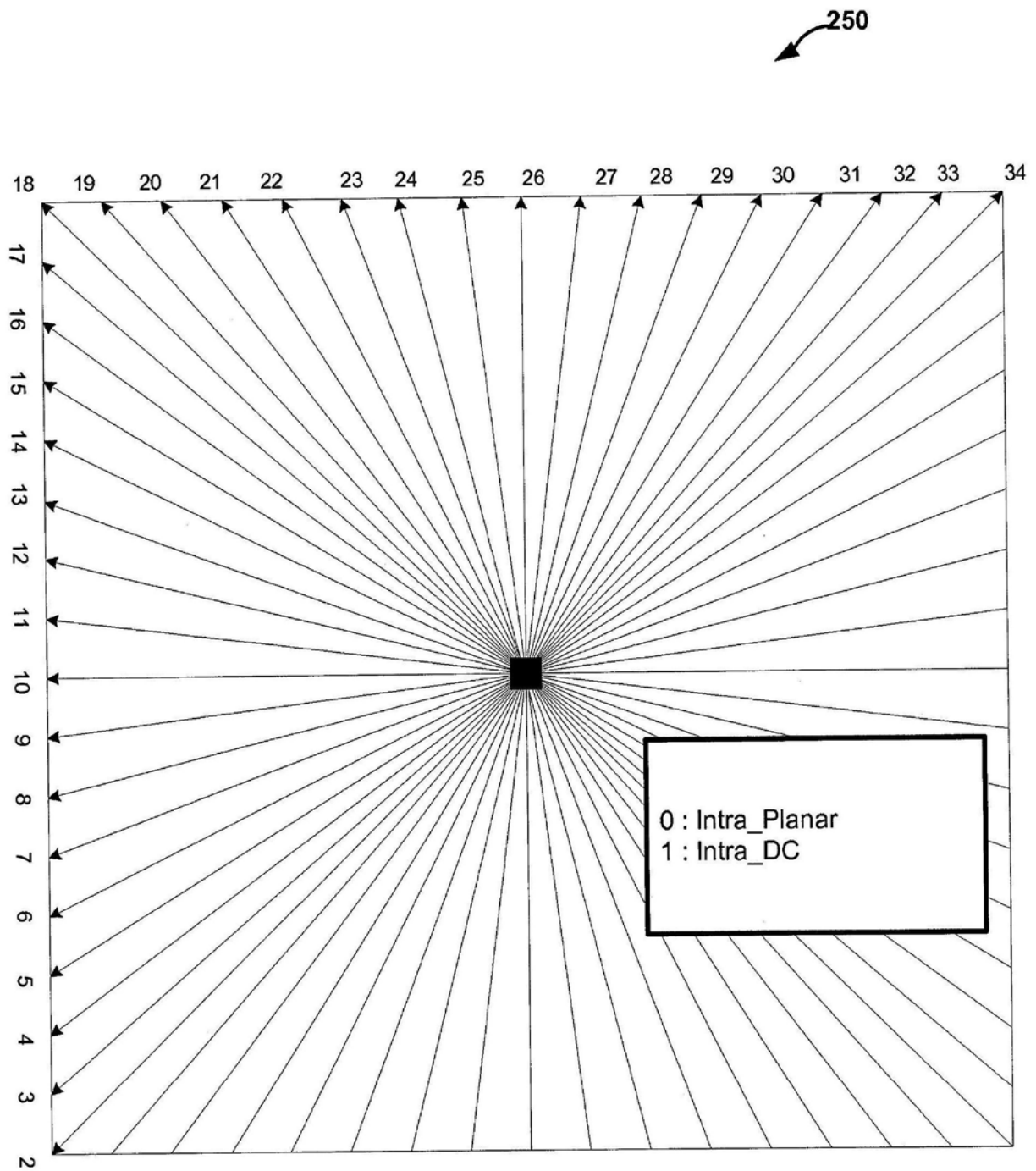


图2

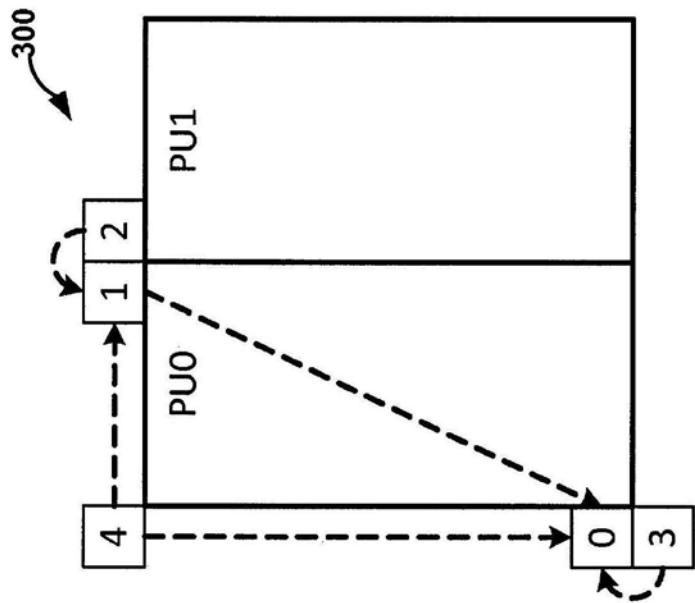


图3A

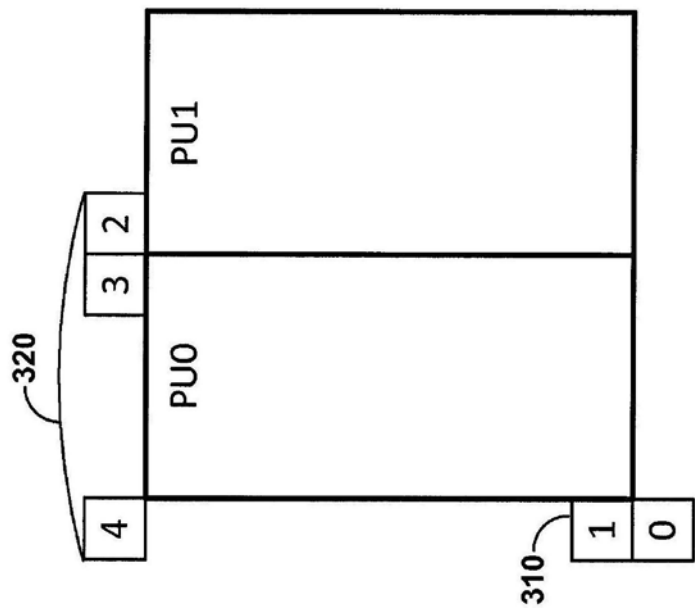


图3B

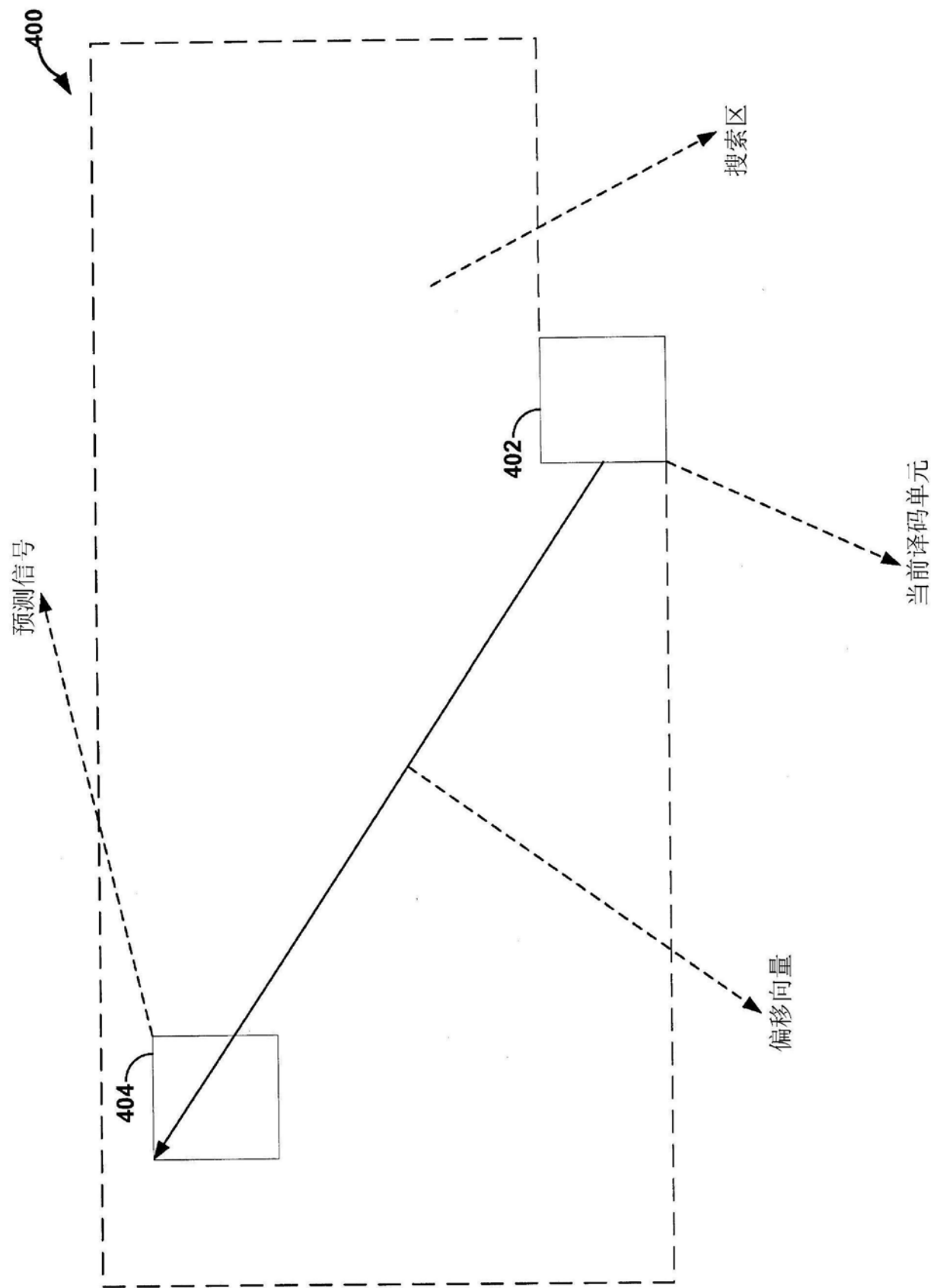


图4

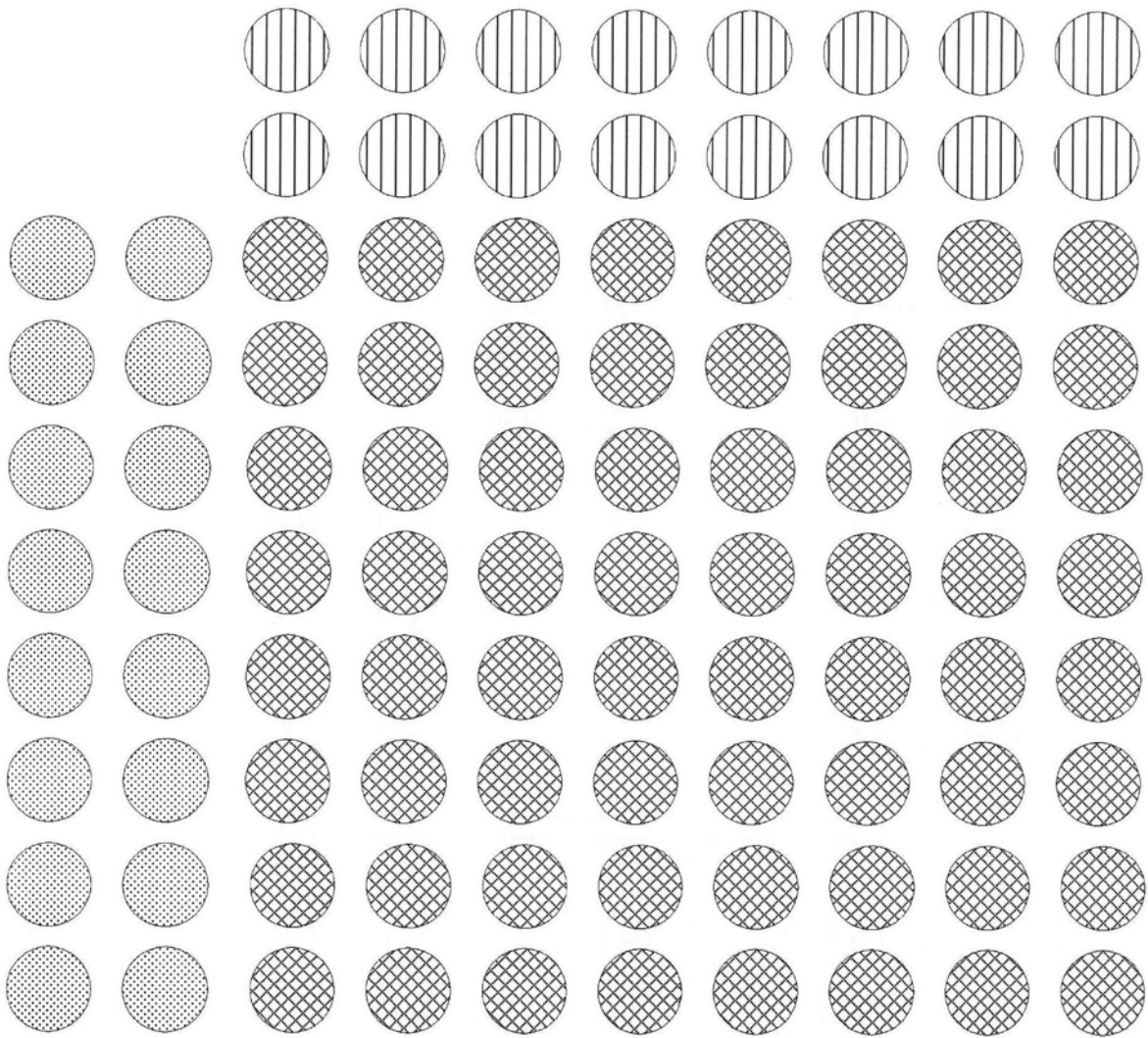


图5

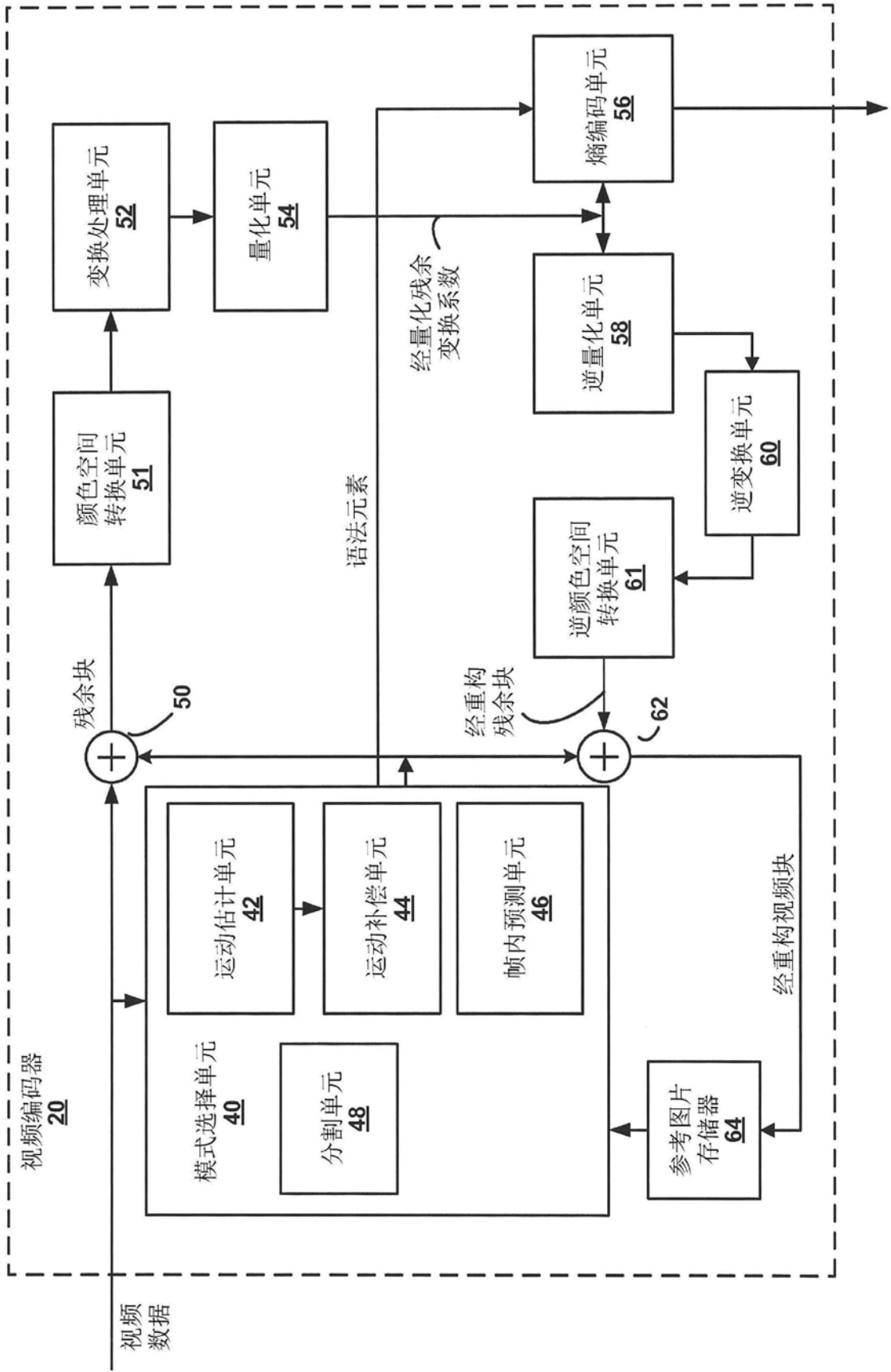


图6

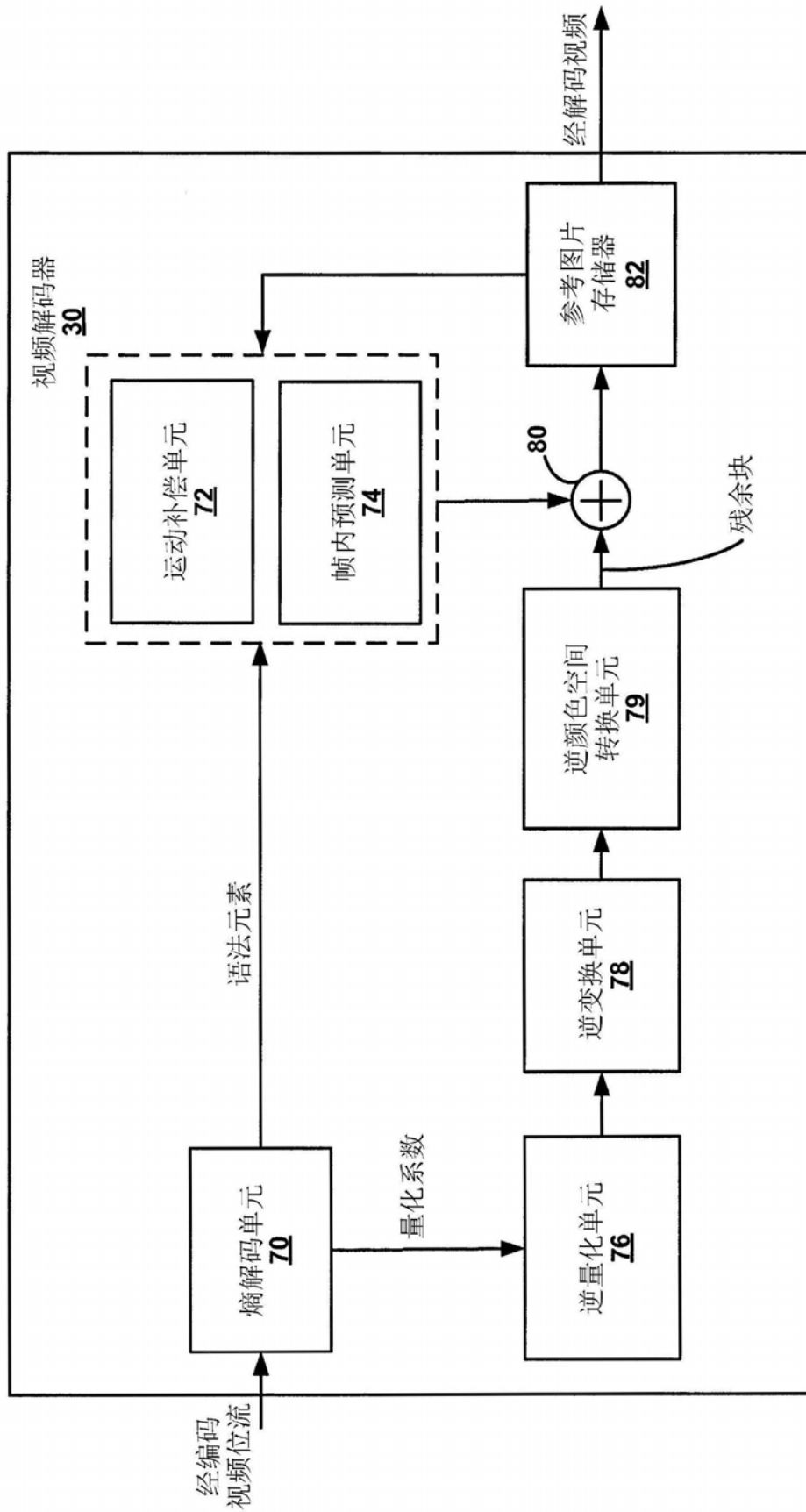


图7

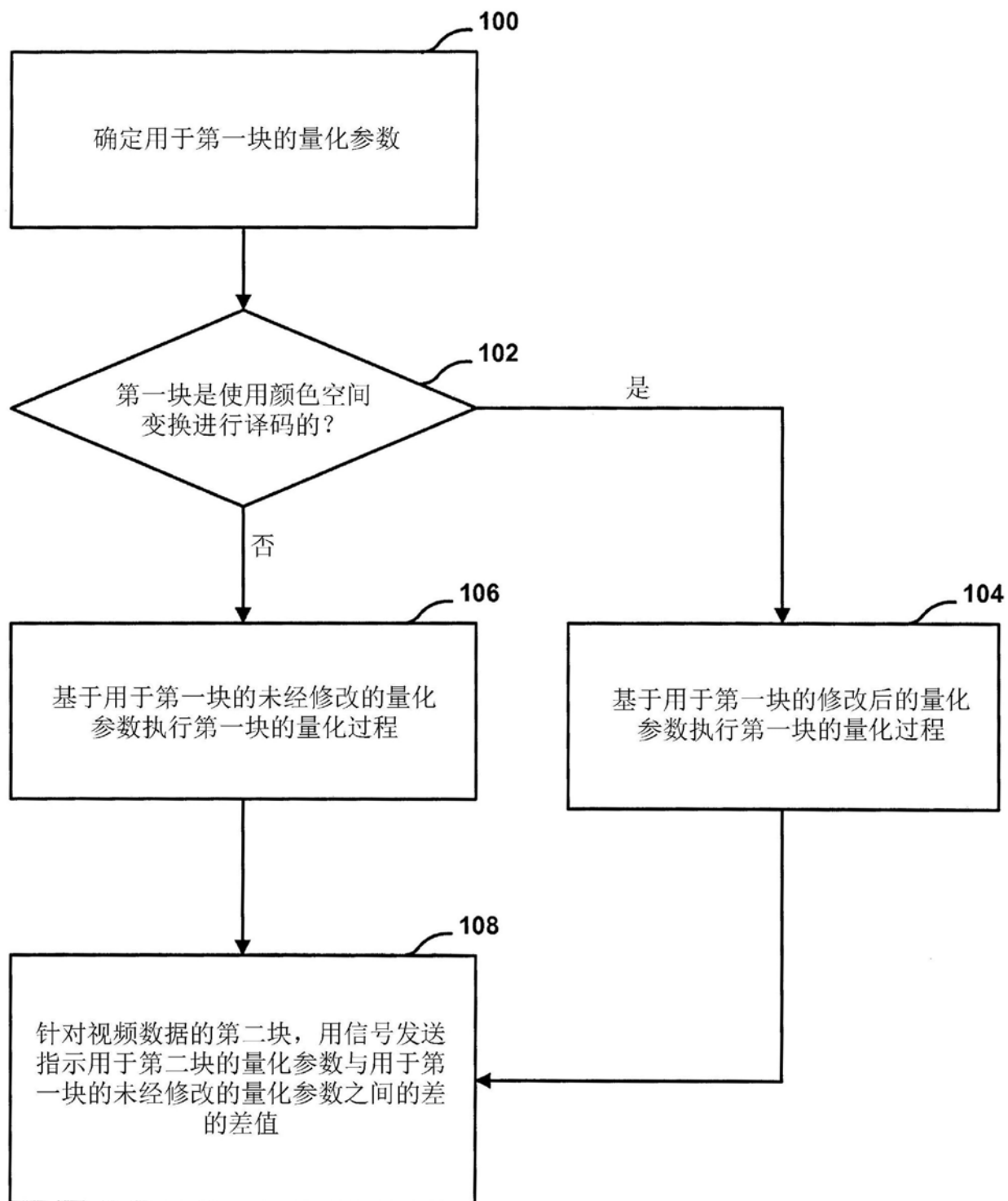


图8

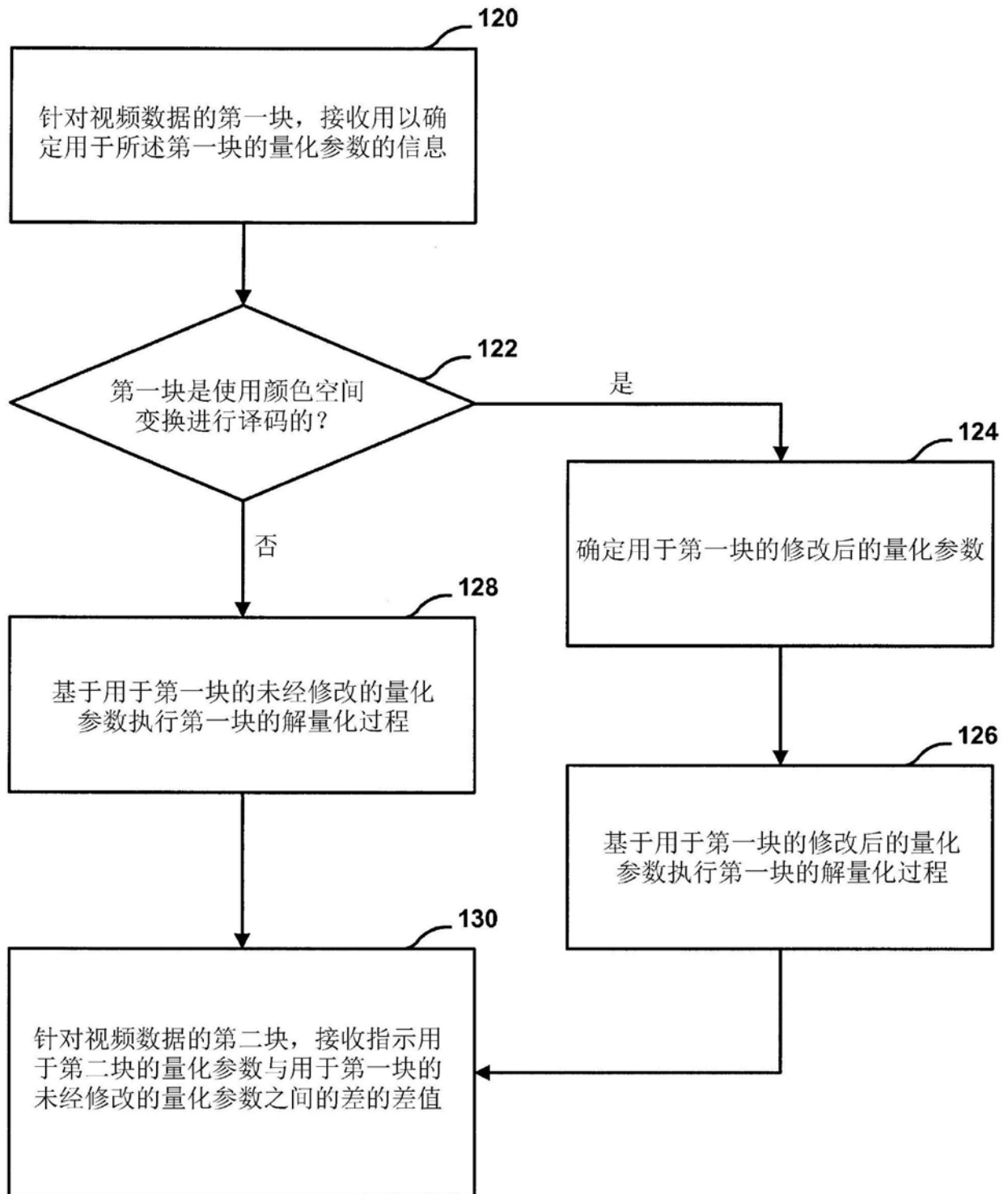


图9